



Politecnico
di Bari

Repository Istituzionale dei Prodotti della Ricerca del Politecnico di Bari

Advanced service orchestration for intent-based networks: architectures, security, and algorithms

This is a PhD Thesis

Original Citation:

Advanced service orchestration for intent-based networks: architectures, security, and algorithms / De Trizio, F.. - ELETTRONICO. - (2026).

Availability:

This version is available at <http://hdl.handle.net/11589/304500> since: 2026-07-03

Published version

DOI:

Publisher: Politecnico di Bari

Terms of use:

(Article begins on next page)



Politecnico
di Bari

Department of Electrical and Information Engineering
ELECTRICAL AND INFORMATION ENGINEERING

Ph.D. Program

SSD: IINF-03/A–Telecommunications

Final Dissertation

Advanced Service Orchestration for Intent-Based Networks: Architectures, Security, and Algorithms

by
de Trizio Federica

Supervisors:

Prof. Giuseppe Piro

Prof. Gennaro Boggia

Coordinator of Ph.D. Program:

Prof. Nicola Giaquinto

Course n°38, 01/04/2023-31/03/2026



LIBERATORIA PER L'ARCHIVIAZIONE DELLA TESI DI DOTTORATO

Al Magnifico Rettore
del Politecnico di Bari

La sottoscritta Federica de Trizio nata a Bisceglie (BT) il 30/06/1998 residente a Molfetta (BA) in via Leonardo Mezzina, 4, e-mail federicadetrizio@gmail.com, iscritta al 3° anno di Corso di Dottorato di Ricerca in Ingegneria Elettrica e dell'Informazione ciclo 38esimo ed essendo stata ammessa a sostenere l'esame finale con la prevista discussione della tesi dal titolo:

Advanced Service Orchestration for Intent-Based Networks: Architectures, Security, and Algorithms

DICHIARA

- 1) di essere consapevole che, ai sensi del D.P.R. n. 445 del 28.12.2000, le dichiarazioni mendaci, la falsità negli atti e l'uso di atti falsi sono puniti ai sensi del codice penale e delle Leggi speciali in materia, e che nel caso ricorressero dette ipotesi, decade fin dall'inizio e senza necessità di nessuna formalità dai benefici conseguenti al provvedimento emanato sulla base di tali dichiarazioni;
- 2) di essere iscritto al Corso di Dottorato di ricerca in Ingegneria Elettrica e dell'Informazione ciclo 38esimo, corso attivato ai sensi del "Regolamento dei Corsi di Dottorato di ricerca del Politecnico di Bari", emanato con D.R. n.286 del 01.07.2013;
- 3) di essere pienamente a conoscenza delle disposizioni contenute nel predetto Regolamento in merito alla procedura di deposito, pubblicazione e autoarchiviazione della tesi di dottorato nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica;
- 4) di essere consapevole che attraverso l'autoarchiviazione delle tesi nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica del Politecnico di Bari (IRIS-POLIBA), l'Ateneo archiverà e renderà consultabile in rete (nel rispetto della Policy di Ateneo di cui al D.R. 642 del 13.11.2015) il testo completo della tesi di dottorato, fatta salva la possibilità di sottoscrizione di apposite licenze per le relative condizioni di utilizzo (di cui al sito <http://www.creativecommons.it/Licenze>), e fatte salve, altresì, le eventuali esigenze di "embargo", legate a strette considerazioni sulla tutelabilità e sfruttamento industriale/commerciale dei contenuti della tesi, da rappresentarsi mediante compilazione e sottoscrizione del modulo in calce (Richiesta di embargo);
- 5) che la tesi da depositare in IRIS-POLIBA, in formato digitale (PDF/A) sarà del tutto identica a quelle **consegnate**/inviata/da inviarsi ai componenti della commissione per l'esame finale e a qualsiasi altra copia depositata presso gli Uffici del Politecnico di Bari in forma cartacea o digitale, ovvero a quella da discutere in sede di esame finale, a quella da depositare, a cura dell'Ateneo, presso le Biblioteche Nazionali Centrali di Roma e Firenze e presso tutti gli Uffici competenti per legge al momento del deposito stesso, e che di conseguenza va esclusa qualsiasi responsabilità del Politecnico di Bari per quanto riguarda eventuali errori, imprecisioni o omissioni nei contenuti della tesi;
- 6) che il contenuto e l'organizzazione della tesi è opera originale realizzata dal sottoscritto e non compromette in alcun modo i diritti di terzi, ivi compresi quelli relativi alla sicurezza dei dati personali; che pertanto il Politecnico di Bari ed i suoi funzionari sono in ogni caso esenti da responsabilità di qualsivoglia natura: civile, amministrativa e penale e saranno dal sottoscritto tenuti indenni da qualsiasi richiesta o rivendicazione da parte di terzi;
- 7) che il contenuto della tesi non infrange in alcun modo il diritto d'Autore né gli obblighi connessi alla salvaguardia di diritti morali od economici di altri autori o di altri aventi diritto, sia per testi, immagini, foto, tabelle, o altre parti di cui la tesi è composta.

Luolo e data Molfetta, 31/03/2026

Firma

Federica de Trizio

La sottoscritta, con l'autoarchiviazione della propria tesi di dottorato nell'Archivio Istituzionale ad accesso aperto del Politecnico di Bari (POLIBA-IRIS), pur mantenendo su di essa tutti i diritti d'autore, morali ed economici, ai sensi della normativa vigente (Legge 633/1941 e ss.mm.ii.),

CONCEDE

- al Politecnico di Bari il permesso di trasferire l'opera su qualsiasi supporto e di convertirla in qualsiasi formato al fine di una corretta conservazione nel tempo. Il Politecnico di Bari garantisce che non verrà effettuata alcuna modifica al contenuto e alla struttura dell'opera.
- al Politecnico di Bari la possibilità di riprodurre l'opera in più di una copia per fini di sicurezza, back-up e conservazione.

Luolo e data Molfetta, 31/03/2026

Firma

Federica de Trizio





Politecnico
di Bari

Department of Electrical and Information Engineering
ELECTRICAL AND INFORMATION ENGINEERING

Ph.D. Program

SSD: IINF-03/A–Telecommunications

Final Dissertation

Advanced Service Orchestration for Intent-Based Networks: Architectures, Security, and Algorithms

by

de Trizio Federica

Federica de Trizio

Referees:

Prof. Giacomo Morabito

Prof. Antonio Iera

Supervisors:

Prof. Giuseppe Piro

Prof. Gennaro Boggia

Coordinator of Ph.D. Program:

Prof. Nicola Giaquinto

«I'm not sure what I'll do, but – well, I want to go places and see people, I want my mind to grow. I want to live where things happen on a big scale.»

F. Scott Fitzgerald

POLITECNICO DI BARI

Abstract

Department of Electrical and Information Engineering

Doctor of Philosophy

Advanced Service Orchestration for Intent-Based Networks: Architectures, Security, and Algorithms

by Federica de Trizio

The deployment of 5G and the evolution toward 6G are transforming communication networks into dynamic, programmable ecosystems where conventional performance-driven management is no longer sufficient. Future networks must simultaneously satisfy technical requirements and broader value-oriented objectives formalized as Key Value Indicators (KVIs) alongside traditional Key Performance Indicators (KPIs). Against this background, three challenges remain open: securely translating high-level human intents into executable orchestration actions that jointly meet KPIs and KVIs; endowing Digital Twins (DTs) with the algorithmic intelligence to turn network state into context-aware decisions; and extending orchestration beyond centralized assumptions toward secure, autonomous control in decentralized environments. In response, this thesis advances service orchestration for Intent-Based Networks (IBNs) along three interconnected dimensions. First, it proposes a secure intent translation framework and a bi-objective optimization approach for joint KPI and KVI satisfaction. Second, it develops a DT-assisted algorithmic intelligence perspective grounded in Multi-Criteria Decision Making (MCDM) and game-theoretic methods. Third, it explores how agentic AI paradigms can enable autonomous, trust-aware orchestration at the edge of future Internet-of-Everything (IoE) environments.

Acknowledgements

I've heard it for the first time when I was fourteen, and I've never stopped remembering it ever since: *πάθει μάθος*, learning through suffering. In the *Antigone*, this would've been a non-written law of the chthonic gods, and the common sense, to abide by; for the Federica I have come to know, it is a pillar to defy. Today, I'd like to plead for two of my most beloved ancient Greek sentences, *κτῆμα ἐς αἰεί* ("possession for ever"), and *δίκη πάντως ἔλθε* ("Justice always comes"). The former refers to history, from which to learn not to repeat the same mistakes. The latter, instead, doesn't pray for vengeance, but paci es; the stride of justice is short but inevitable: all in due time. From my shyly romantic perspective, I want this to translate into *every orts will always be repaid*.

The biggest and most sincere and heartfelt *thank you* goes to my favorite PhD student, and post-doc, and researcher, and person to learn from, who let me make all the unknowable mistakes in the world. My *most precious* one, Giancarlo: I am Nick but, in every life as Daisy, I would always choose Jay. Thank you for the irresistible prejudice in my favor, and for drawing out the best in me. I always felt safe knowing you were my team. You are a wonderful teacher; I hope you never stop being one.

The second thank you goes to Ingrid, the best comrade I could have asked for: thank you for the countless tries. And the talks. And the walks. It wouldn't have been this special without you. Thanks to my Telematics, begging for me to talk, shut up, and finally stop crying. Thanks to my coddling DIANA duo, for making me feel at home when I was kilometres away from it. Thanks to my mum, my dad, my sister: I love coming back home. And lastly, my PhD advisors, who seeded this path by asking me the question I had been desperately waiting for ("*Have you ever considered a PhD?*"), without whom I wouldn't have met all these people, nor lived all these lives.

For once, yes, I could have done more – I should have done so much more – but I am nonetheless terribly satisfied.

[ ]All contributions are the result of research activities carried out by different academic and industrial partners, collaborating as a partnership in the context of the project COHERENT, supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 – program "RESTART").

Contents

Acknowledgements	v
List of Figures	xii
List of Tables	xiii
List of Acronyms	xv
Scientific Contributions	xxi
Introduction	1
1 Secure Intent Translation for KPI and KVI Satisfaction	5
1.1 Background and Related Works	5
1.1.1 Foundations of Intent-Based Networking	6
1.1.2 Intent expression, translation, and lifecycle management . .	7
1.1.3 Security in intent-driven systems	9
1.1.4 From KPI-centric orchestration to KVI-aware network man- agement	9
1.2 Architecture for Secure Intent Translation	10
1.2.1 Secure intent admission	12
1.2.1.1 Rasa-based Intent Profiling chatbot	14
1.2.1.2 Malicious Intent Detection module	15
1.2.1.3 Performance Evaluation	17
1.2.2 Value-driven intent translation and orchestration	20
1.2.2.1 System Model	21
1.2.2.2 Social and Ethical Values	22
1.2.2.3 Problem Formulation and Solution	26
1.2.2.4 Optimization Support Through TETREES	31
1.2.2.5 Performance Evaluation	35
1.3 Case study: Sustainable Video Streaming	47
1.3.1 GreenTent Framework	50
1.3.2 Sustainability-aware Orchestration Logic	51
1.3.3 Performance Evaluation	56
2 Digital Twin-Driven Intelligent Orchestration	61
2.1 Background and Related Works	62
2.1.1 Service orchestration in advanced network environments .	62
2.1.2 Digital-twin-assisted orchestration	62
2.1.3 Intent-driven orchestration	63

2.1.4	Task offloading and disaster-response scenarios	64
2.2	Surviving disaster events via dynamic in-network processing as- sisted by NDTs	65
2.2.1	The Reference Scenario and the Proposed MCDM Solution	65
2.2.2	Network Digital Twins Parameters	67
2.2.3	The Conceived Algorithm Based on TOPSIS Methodology	68
2.2.4	Performance Evaluation	69
2.2.5	Simulation Results	70
2.3	Optimizing KVI through DT-aided service orchestration mechanism	74
2.3.1	The Reference Architecture and System Model Definition	74
2.3.2	Parameters Computation	78
2.3.3	Problem Formulation	80
2.3.4	The Proposed Solution	83
2.3.5	Simulation Results	90
2.4	From theory to practice: a COHERENT perspective	102
2.4.1	Reference Architecture for Digital Twin Networks	103
2.4.2	Cloud-to-Edge Management via the Asset Administration Shell	104
2.4.3	COHERENT Architecture Implementation	106
2.4.4	PoC Evaluation: Energy Efficiency	107
2.4.4.1	PoC 1 – Energy Efficiency in 5G Radio Access Net- works	107
2.4.4.2	PoC 2 – Energy Optimization in Fixed Access Broadband Networks	108
3	Distributed and Trustworthy Orchestration for Future Internet-of- Everything Environments	111
3.1	Agentic AI for Autonomous Internet-of-Everything	112
3.1.1	From Foundation Models to Distributed Agentic Orchestration	112
3.1.1.1	Language-Model-Powered Agents	113
3.1.1.2	LLM- and SLM-Powered Agents	113
3.1.1.3	TinyML-Powered Agents	114
3.1.1.4	Multi-Agent Systems and Interoperability	115
3.1.2	Distributed Agentic Intelligence for Autonomous IoE	115
3.1.2.1	Digital Twins as Predictive Substrates for Au- tonomous 6G Networks	116
3.1.2.2	Enhancing Intent-Based Networking with Agen- tic AI	116
3.1.2.3	A Three-Tier Intelligence Architecture for 6G IoE Management	117
3.1.3	Case Studies: Agentic Cross-Domain Intent Assurance	119
3.1.3.1	V2I Cooperative Perception Scenario	119
3.1.3.2	Cross-Domain Hospital Smart-Alarm Scenario	120
3.1.3.3	Performance Evaluation	120
3.1.4	Challenges and Research Ahead	121
3.1.4.1	Design and Deployment	125
3.1.4.2	Performance Metrics	125

3.1.4.3	Evaluation Strategies and Benchmarks	126
3.1.4.4	Sustainable Intelligence: Green Challenges	127
3.1.4.5	Explainability and Trustworthiness	127
3.1.4.6	Ethical, Privacy, and Security Implications	128
3.2	Trustworthiness Dissemination in Social IoE Architectures	128
3.2.1	Problem Statement	129
3.2.2	Proposed Framework	129
3.2.3	Preliminary Evaluation	131
	Conclusions and Future Works	133
	Bibliography	135

List of Figures

1.1	Proposed intent-based architecture [60].	11
1.2	The considered enterprise network environment [62].	12
1.3	The proposed Malicious Intent Detection mechanism within the enterprise intent-processing pipeline [62].	13
1.4	Confusion matrices of entities and intentions [62].	18
1.5	Confidence distribution of predicted intentions.	19
1.6	Overview of the Trade-off Evaluation Through Refined Exact Epsilon-Constraint Solver (TETREES) framework architecture [61].	31
1.7	Pareto fronts for increasing numbers of service providers (N) considering 300 service requests [61].	34
1.8	Pareto fronts considering different service requests (J) and network resources (N).	38
1.9	Pareto fronts with varying service requests J and δ [60].	39
1.10	Pareto fronts obtained for varying amounts of network resources, with the number of service requests fixed at $J = 100$ [60].	40
1.11	Network quality performance and social and ethical value considering neither quality nor social and ethical prioritization [60]. . . .	42
1.12	Network quality performance and social and ethical value considering an environmental sustainability focus [60].	45
1.13	Network quality performance and social and ethical value considering a trustworthiness focus [60].	46
1.14	Network quality performance and social and ethical value considering an inclusiveness focus [60].	48
1.15	The proposed intent-based orchestration framework [80].	50
1.16	Perceived video quality for the first service request under (a) Green-Tent, (b) throughput-oriented orchestration (Throughput-Oriented (TO)), and (c) energy-aware application-level approach (Energy-Aware (EA)) [80].	56
1.17	Average delivered bitrate, carbon efficiency, and CO ₂ emissions under increasing load [80].	57
2.1	Reference scenario and Intent-Based Networking (IBN)-based network architecture [116].	66
2.2	Overview of the drone flight and domain selection [116].	71
2.3	Overview of the drone flight and domain selection [116].	72
2.4	Overloaded data, missed acquisitions and overall service availability for CPU Service Level Agreement (SLA) focus [116].	73
2.5	Overloaded data, missed acquisitions and overall service availability for trustworthiness SLA focus [116].	73

2.6	The proposed architecture inspired by the framework of IBN for Network 2030 [117].	75
2.7	Overall orchestration procedure [69].	84
2.8	SLA missed adherence [69].	94
2.9	Average matching satisfaction per request [69].	96
2.10	Key Performance Indicator (KPI) and Key Value Indicator (KVI) analysis assessed in S2 with 1500 requests/30 min [69].	97
2.11	KPI and KVI analysis assessed in S2 with 2500 requests/30 min [69].	98
2.12	KPI and KVI analysis assessed in S4 with 1500 requests/30 min [69].	99
2.13	KPI and KVI analysis assessed in S4 with 2500 requests/30 min [69].	100
2.14	Reference architecture for the proposed Digital Twin Network (DTN) [134].	103
2.15	YAML Ain't Markup Language (YAML) file of the Digital Twin (DT) deployment [134].	105
2.16	COHERENT implementation modules [134].	106
2.17	Simulated RAN scenario with next-generation NodeBs (gNBs) and User Equipments (UEs) [134].	108
2.18	RAN cluster energy consumption with Large Language Model (LLM) Energy Saving enabled [134].	108
2.19	Gigabit Passive Optical Network (GPON) power consumption with Broadband Network Gateway (BNG) consolidation [134].	109
3.1	Architecture of a tool-using language-model agent [139].	114
3.2	Three-tier agentic architecture integrating IBN and DTs for closed-loop Internet of Everything (IoE) autonomy [139].	117
3.3	Per-class bandwidth allocation across all controllers and stress scenarios [139].	122
3.4	Mean Time-to-Action (TTA) across all controllers and stress scenarios [139].	122
3.5	Operational carbon footprint per decision [139].	122
3.6	The proposed double-clustered architecture [154].	130
3.7	Entities' reputation values propagation [154].	131

List of Tables

1.1	Precision, Recall, F1-score for the entity extraction task.	19
1.2	Precision, Recall, F1-score for the intentions classification.	20
1.3	Main Symbols Description.	23
1.4	Simulation results considering 300 service requests and 110 service providers under different solver configurations.	35
1.5	Simulation settings.	36
1.6	Simulation run times with varying δ	38
1.7	Simulation run times with varying weights of KVI.	40
1.8	Role of KPIs and KVIs across the proposed framework.	59
2.1	Review of related works on disaster-oriented task offloading and DT-assisted domain selection.	65
2.2	Main symbols description.	76
2.3	Intent categories definition.	92
2.4	Simulation settings.	92
3.1	V2I scenario: per-scenario intent configuration and decision outcomes across all baselines (mean $\pm$ std over three repetitions). Decision latency t_{dec} is the control latency; t_{val} is the DT validation share within it.	123
3.2	Hospital smart-alarm scenario: per-scenario intent configuration and decision outcomes across all baselines (mean $\pm$ std over three repetitions). t_{dec} is the control latency; t_{val} is the DT validation share within it.	124

List of Acronyms

3GPP	3rd Generation Partnership Project
AAS	Asset Administration Shell
ACT	Attack Countermeasure Tree
AI	Artificial Intelligence
API	Application Programming Interface
B5G	Beyond 5G
BILP	Binary Integer Linear Programming
BLEU	Bilingual Evaluation Understudy
BNG	Broadband Network Gateway
CCL	Closed-Control Loop
DAA	Deferred Acceptance Algorithm
DAE	Dependency Arc Entailment
DCC	Decentralized Congestion Control
DH	Digital Hat
DIET	Dual Intent and Entity Transformer
DT	Digital Twin
DTD	Digital Twin Definition Language
DTH	Digital Hub
DTN	Digital Twin Network
EA	Energy-Aware
ENI	Experienced Networked Intelligence
ETSI	European Telecommunications Standards Institute
EWM	Entropy-Weighting Method
FLAN	Finetuned Language Net

- GAN** Generative Adversarial Network
- GAP** Generalized Assignment Problem
- GHG** greenhouse gas
- GM** Greedy Matching
- gNB** next-generation NodeB
- GPON** Gigabit Passive Optical Network
- GPU** Graphics Processing Unit
- GUI** Graphical User Interface
- HTTP** Hypertext Transfer Protocol
- IBN** Intent-Based Networking
- IDS** Intrusion Detection System
- IEC** International Electrotechnical Commission
- IETF** Internet Engineering Task Force
- IoE** Internet of Everything
- IoT** Internet of Things
- IDTA** Industrial Digital Twin Association
- ITU** International Telecommunication Union
- ITU-T** ITU Telecommunication Standardization Sector
- JDM** Joint Decision Making
- K8S** Kubernetes
- KPI** Key Performance Indicator
- KVI** Key Value Indicator
- LIME** Local Interpretable Model-agnostic Explanations
- LLM** Large Language Model
- M2M** Machine-to-Machine
- MAS** Multi-Agent System
- MCDM** Multi-Criteria Decision Making
- MEC** Multi-access Edge Computing

MID Malicious Intent Detection

MIP Mixed-Integer Programming

ML Machine Learning

mMTC massive Machine-Type Communications

MQTT Message Queuing Telemetry Transport

MTTF Mean Time To Failure

NDT Network Digital Twin

NER Named Entity Recognition

NFV Network Function Virtualization

NGN Next-Generation Network

NILE Network Intent Language

NIS Negative Ideal Solution

NLP Natural Language Processing

NTN Non-Terrestrial Network

OAM Operations, Administration, and Management

OLT Optical Line Terminal

ONOS Open Network Operating System

ORAN Open Radio Access Network

PGM Performance-oriented Greedy Matching

PIS Positive Ideal Solution

PoC Proof of Concept

PoW Proof of Work

PRB Physical Resource Block

PUE Power Usage Effectiveness

QA Question Answering

QoS Quality of Service

RAG Retrieval-Augmented Generation

RAMI Reference Architectural Model for Industry

- RAN** Radio Access Network
- RBAC** Role-Based Access Control
- RC** Relative Closeness
- REST** Representational State Transfer
- RIC** RAN Intelligent Controller
- RLHF** Reinforcement Learning from Human Feedback
- RM** Random Matching
- ROUGE** Recall-Oriented Understudy for Gisting Evaluation
- RSU** Road Side Unit
- SDG** Sustainable Development Goal
- SDSN** Software-Defined Systems and Networks
- SDN** Software-Defined Networking
- SFC** Service Function Chain
- SHAP** SHapley Additive exPlanations
- SloE** Social Internet of Everything
- SINR** Signal-to-Interference-plus-Noise Ratio
- SLA** Service Level Agreement
- SLM** Small Language Model
- SMS** Semantic Matching Score
- SOA** Service-Oriented Architecture
- SOS** Sustainability of Service
- SQL** Structured Query Language
- STIX** Structured Threat Information eXpression
- SVM** Support Vector Machine
- SDK** Software Development Kit
- TETREES** Trade-off Evaluation Through Refined Exact Epsilon-Constraint Solver
- TinyML** Tiny Machine Learning
- TMS** Trust Management System

TN-NTN Terrestrial Network-Non Terrestrial Network

TO Throughput-Oriented

TPU Tensor Processing Unit

TTA Time-to-Action

UAV Unmanned Aerial Vehicle

UE User Equipment

UN United Nations

URLLC Ultra-Reliable Low-latency Communications

V2I Vehicle-to-Infrastructure

VAB Virtual Automation Bus

VGM Value-oriented Greedy Matching

VNF Virtual Network Function

XAI Explainable Artificial Intelligence

xApp xApplication

YAML YAML Ain't Markup Language

YANG Yet Another Next Generation

ZSM Zero-touch network and Service Management

CQI Channel Quality Indicator

SINR Signal-to-Interference-plus-Noise Ratio

RL Reinforcement Learning

Scientific Contributions

All the scientific contributions produced during the doctoral course are listed below.

International Journals:

- F. de Trizio, G. Sciddurlo, I. Ciani, G. Piro, and G. Boggia, "Optimizing Key Value Indicators in Intent-Based Networks through Digital Twins aided service orchestration mechanisms", *Computer Communications* (Elsevier), 228 (2024), 107977.
- G. Sciddurlo, F. de Trizio, G. Piro, G. Boggia, "A Value-Driven System Design Framework for Sustainable 6G Networks", *Computer Networks* (Elsevier), 269 (2025), 111477.
- F. de Trizio, G. Sciddurlo, G. Piro, and G. Boggia, "Tetrees: Trade-off evaluation through refined exact epsilon-constraint solver," *Software Impacts* (Elsevier), 28 (2026), 100828.

International Conferences:

- F. de Trizio, G. Sciddurlo, G. Piro, D. Striccoli, I. Ciani, and G. Boggia, "Surviving Disaster Events Via Dynamic In-Network Processing Assisted by Network Digital Twin", in *2023 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, Cosenza, Italy, 2023.
- F. de Trizio, G. Sciddurlo, D. Rutigliano, G. Piro, and G. Boggia, "A Novel Malicious Intent Detection Approach in Intent-Based Enterprise Networks", in *2024 20th International Conference on Network and Service Management (CNSM)*, Prague, Czech Republic, 2024.
- F. de Trizio, G. Sciddurlo, A. Petrosino, G. Piro, G. Boggia, "A Scalable Framework for Responsive Trustworthiness Dissemination in Social IoA", *Proceedings of the ACM CoNEXT Student Workshop 2024 (CoNEXT-SW '24)*, Los Angeles, California, 2024.
- M. Marchese, F. de Trizio, G. Sciddurlo, N. Cordeschi, G. Piro, G. Boggia, "GreenTent: an Intent-based Framework for Sustainable Network Orchestration", in *2026 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2026.

Book Chapters:

- F. de Trizio, G. Sciddurlo, S. Rossitto, G. di Natale, M. Becattini, G. Lia, C. Suraci, and L. Pintor, "Sustainable Digital Twin Networks: Architecture,

Intent-Based Orchestration, and Optimization", RESTART Telecommunications of the Future: A book of contributions from the RESTART program, Nicola Blefari Melazzi and Antonio Capone, (Eds.), Chapter 1, Fondazione RESTART, ISBN: 9788832170061, 2026.

Under Review:

- F. de Trizio, G. Sciddurlo, G. Piro, G. Boggia, "Agentic AI for Autonomous Internet-of-Everything: A Distributed Three-Tier Intelligence Ecosystem", submitted to IEEE Internet of Things Magazine.

Awards and Recognitions:

- Best Poster Runner-up, RESTART Plenary Dissemination Workshop, Bologna, Italy, January 31, 2024, for the poster "Effective Intent-based Service Provisioning via Network Digital Twins".
- Best Poster Winner, RESTART Plenary Dissemination Workshop, Catania, Italy, July 5, 2024, for the poster "Coherent: introducing a reference architecture for digital twin networks".
- Best Poster Runner-up, RESTART Plenary Dissemination Workshop, Catania, Italy, July 5, 2024, for the poster "Optimizing Key Value Indicators in Intent-Based Networks through Digital Twins aided service orchestration mechanisms".
- Best Poster Winner, RESTART Plenary Dissemination Workshop, Palermo, Italy, January 30, 2025, for the poster "Towards Communication Network Virtualization: Asset Administration Shell and AI-Powered Cloud-to-Edge Integration for Realizing a Digital Twin of a Coherent Network".
- Best Poster Runner-up, RESTART Plenary Dissemination Workshop, Palermo, Italy, January 31, 2025, for the poster "Integrating Digital Twin Federation and Intent-Based Networking for Optimized Performance in Ethical Next-Generation Networks".
- Best Project Contribution Winner, RESTART Plenary Dissemination Workshop, Napoli, Italy, June 30, 2025, for the demonstration of a complete Digital Twin Network developed and integrated within the COHERENT project.
- Recipient of the Francesco Carassa Award 2025, conferred by the Gruppo Telecomunicazioni e Tecnologie dell'Informazione (GTTI) during the Annual Meeting of GTTI, Bologna, Italy, September 15–17, 2025, for the work entitled "Digital Twin Networks and Secure and Autonomous Orchestration of Beyond 5G Systems through Intent-Aware and Value-Driven Optimization".

Introduction

The ongoing deployment of 5G systems and the evolution toward Next-Generation Networks (NGNs) are pushing communication infrastructures to unprecedented levels of complexity [1]. The convergence of softwarization, virtualization, cloud-native architectures, Artificial Intelligence (AI)-enabled control, service disaggregation, and multi-domain orchestration is transforming networks from static service platforms into dynamic, programmable, and highly interdependent ecosystems. As devices, data, services, and users become increasingly intertwined, coordination demands intensify, resource heterogeneity grows, and management becomes substantially more challenging [2]. In this context, conventional approaches based on static configurations, isolated monitoring, centralized control, and manual intervention are no longer sufficient to guarantee efficient, trustworthy, and adaptive network operation.

At the same time, mobile and wireless networks have evolved far beyond their traditional role as communication infrastructures. They have become essential enablers of social participation, economic activity, and access to critical services, making connectivity pervasive and continuously available. Their impact is especially evident in applications such as telepresence, remote collaboration, telemedicine, and online education, where connectivity reduces the need for permanent physical co-location, improves accessibility, and fosters inclusion, particularly in rural and underserved areas. Telecommunications therefore contribute not only to ubiquitous service delivery, but also to broader social and environmental objectives, making their design and operation increasingly relevant within the wider sustainability agenda [3].

This broader perspective naturally connects the evolution of communication networks to the framework established by the United Nations (UN) Sustainable Development Goals (SDGs) [4]. Since the definition of the 17 SDGs as a shared blueprint for peace and prosperity for people and the planet, all sectors have been called to contribute to long-term environmental, social, and economic sustainability, and telecommunications have progressively emerged as one of the enabling domains through which such goals can be supported. Within this context, networking exhibits an inherently dual role [5]. On the one hand, networks enable sustainability in other sectors by supporting digital services that reduce travel, optimize industrial processes, improve access to healthcare and education, and promote more efficient use of resources. On the other hand, networks themselves must become more sustainable by reducing energy consumption, greenhouse gas (GHG) emissions, waste, and overall environmental footprint across their life cycle [6]. This distinction is reflected in the sustainability terminology proposed in the Internet Engineering Task Force (IETF), which separates the sustainability of a technology from the sustainability outcomes it enables in external domains, and stresses the need to evaluate

both rigorously [7]. For future wireless systems, this means that sustainability cannot remain a generic aspiration, but must be grounded in measurable objectives and actionable decision mechanisms [8].

This evolution also implies that NGNs can no longer be assessed solely through conventional performance-oriented metrics. Traditional KPIs remain essential for characterizing technical behaviour, service quality, and operational efficiency, including throughput, latency, reliability, availability, and resource utilization. However, they are not sufficient to capture broader environmental, ethical, and social objectives when networks are also expected to be sustainable, inclusive, and trustworthy. For this reason, recent Beyond-5G research introduces KVIs as complementary criteria that quantify value-oriented dimensions and connect technical behaviour with broader societal outcomes [9]. In this sense, KVIs do not replace KPIs; rather, they extend the evaluation space of telecommunication systems and provide a bridge between sustainability principles, SDG-oriented aspirations, and the actual governance of advanced communication networks [10]. As a result, future communication systems are increasingly expected not only to optimize performance, but also to embed value-aware reasoning into their management and orchestration processes.

If KPIs and KVIs define what should be measured and optimized, a further challenge concerns how these objectives can be operationalized in increasingly complex network environments. In this regard, DTs have emerged as a particularly promising enabling technology [11]. A DT maintains a virtual counterpart of a physical system, continuously connected to it and enriched with the information needed to represent its relevant properties, behaviours, and state evolution. Over time, this concept has expanded from a product-centric representation into a broader paradigm for simulation, prediction, and control of complex systems, with the goal of anticipating failures, supporting safer experimentation, and improving decision making through a sufficiently faithful digital representation of the physical world. These capabilities make DTs especially relevant for next-generation communication networks. By combining real-time observations with historical data, a DT can support monitoring, diagnosis, fault localization, prediction, and optimization, enabling operators to anticipate issues before they materialize in the physical infrastructure. This can improve resource allocation, service assurance, and operational efficiency, while reducing the cost and risk of direct experimentation on production systems. The networking literature further extends this vision through the notion of Network Digital Twins (NDTs), where the twin becomes part of a layered architecture for modelling, monitoring, and controlling communication infrastructures [12]. From a sustainability perspective, DTs are particularly relevant because they support more efficient operation, better trade-off analysis among performance, resilience, and energy-related objectives, and more informed management decisions.

However, DTs alone do not determine which goals should be pursued, which constraints should prevail, or which system behaviours are most relevant from a service perspective. Their full value emerges when they are embedded within a wider intent-driven control process. Intents provide the semantic directive that gives purpose to the twin: they define what should be achieved, what constraints must be respected, what should be validated before enforcement, and what should be continuously monitored during operation. From this perspective, IBN offers the

abstraction layer through which high-level requirements are expressed, while DTs provide the decision-support environment in which those requirements can be interpreted, evaluated, and validated before being translated into executable actions.

Despite the progress enabled by DTs and IBN, several open challenges still hinder the realization of fully trustworthy, adaptive, and scalable orchestration frameworks for NGNs. First, the secure translation of high-level intents into executable orchestration actions remains difficult, especially when multiple requirements, constraints, and stakeholders coexist, and when orchestration must jointly account for both performance-oriented and value-oriented objectives. Second, although DTs provide a powerful environment for monitoring, simulation, prediction, and validation, their effectiveness depends on the availability of reasoning mechanisms able to transform data into actionable decisions. This calls for orchestration frameworks in which intelligence is not an external add-on, but an integral component of the control process. Third, future network environments will be increasingly heterogeneous, decentralized, and pervasive, requiring orchestration mechanisms that move beyond centralized assumptions and support cooperation among distributed entities while preserving security, trust, and operational autonomy. This is particularly relevant for edge, last-mile, and future IoE-oriented scenarios, where local intelligence and dynamic interactions will play a more active role in the control loop.

Against this background, this thesis investigates advanced service orchestration for intent-based networks through three tightly connected dimensions: architectural design, secure intent translation, and algorithmic decision support. These dimensions are not addressed in isolation. Rather, they define a coherent orchestration perspective in which architectural design provides the structural foundations of the system, secure intent translation ensures the trustworthy operationalization of high-level objectives, and algorithmic decision-support mechanisms enable adaptive and context-aware control. More specifically, the thesis addresses the following research questions:

(RQ1) How can high-level intents be securely translated into orchestrable network actions that jointly satisfy both KPIs and KQIs?

This question is addressed in Chapter 1, which introduces a high-level architecture and a methodological framework for the secure translation of performance- and value-oriented requirements into intents. The chapter also defines a bi-objective optimization framework for service orchestration and presents an implementation-oriented case study, showing how technical metrics and broader value dimensions can be jointly handled within the same orchestration process.

(RQ2) How can DTs be endowed with intelligence to provide trustworthy, context-aware, and effective decision support for network orchestration?

Chapter 2 addresses this question by developing a mathematical-reasoning perspective for intelligent orchestration through Multi-Criteria Decision Making (MCDM) methods and game-theoretic algorithms, connecting theory and practice within the broader vision of the COHERENT project.

(RQ3) *How can secure, autonomous, and trustworthy distributed orchestration be achieved across last-mile and future network environments?*

Chapter 3 discusses this question by exploring the evolution of orchestration toward distributed and trustworthy intelligence, with particular attention to security, autonomy, and future-facing architectural directions. It adopts a prospective view, examining how emerging agentic AI paradigms may contribute to the orchestration of highly dynamic IoE ecosystems.

Finally, the conclusions summarize the main findings of the thesis, discuss their implications, and outline possible future research directions.

Chapter 1

Secure Intent Translation for KPI and KVI Satisfaction

IBN has emerged as a promising paradigm to bridge the gap between high-level service objectives and low-level network operations. By enabling users and operators to express desired outcomes in a declarative form, it shifts network management from device-centric configuration to intent-driven service control.

However, the practical effectiveness of IBN depends on a critical capability: the reliable translation of high-level intents into actionable and verifiable orchestration decisions. This step is far from trivial, since intents may be incomplete, ambiguous, context-dependent, or expressed through natural language, and therefore cannot be translated through a purely syntactic mapping. The problem becomes even more demanding in value-aware networks, where orchestration is expected to account not only for conventional KPIs, but also for broader value-oriented dimensions captured through KVIs. In this case, intent translation is no longer limited to deriving technical configurations from service demands, but must support the joint formalization of performance targets, value objectives, and enforcement constraints within a common decision framework. Moreover, the translation pipeline itself cannot be assumed to be neutral or risk-free. When intents are accepted as inputs to automated orchestration, harmful, adversarial, or otherwise untrustworthy formulations may propagate into the operational layer unless they are detected and handled before policy generation and deployment.

Against this background, this chapter develops an architectural and methodological perspective for secure intent translation in value-aware networks. It first reviews the main foundations and related research on intent-based networking, secure intent handling, and value-aware orchestration. It then introduces the reference architecture for secure intent admission and value-driven orchestration, formalizing a bi-objective optimization framework for the joint satisfaction of KPIs and KVIs, and presenting an implementation-oriented case study on sustainable network orchestration.

1.1 Background and Related Works

Future NGNs are expected to satisfy not only technical requirements such as throughput, latency, and scalability, but also broader objectives related to sustainability, resilience, inclusiveness, and trustworthiness [4], [9], [13], [14], [15], [16]. As a result, communication networks are increasingly viewed as socio-technical

systems whose design and operation affect not only service performance, but also energy consumption, environmental impact, and the ability to support wider societal goals [17], [18], [19].

In this context, IBN has emerged as a promising paradigm to manage network complexity by allowing users and operators to express high-level service goals, while delegating the derivation of configurations, policies, and orchestration decisions to the underlying management system [20], [21], [22]. This abstraction is particularly relevant in Beyond 5G (B5G) environments, where automation must jointly account for heterogeneous infrastructures, slicing, cloud-edge integration, service dynamism, and requirements spanning performance, security, and sustainability.

Despite the progress of the literature, three limitations remain particularly relevant for this thesis. First, many IBN solutions focus on semantic interpretation and deployment while implicitly assuming that incoming intents are legitimate and trustworthy. Second, most orchestration approaches are still centered on traditional KPIs, with limited support for broader value-oriented objectives. Third, secure intent processing, semantic translation, and value-aware orchestration are often addressed separately rather than within a unified end-to-end framework. These gaps motivate the background discussion developed in the following subsections and, more importantly, the architectural choices introduced later in this chapter.

1.1.1 Foundations of Intent-Based Networking

The conceptual foundations of IBN are grounded in the notion of *intent* as a declarative expression of operational goals and expected outcomes. According to RFC 9315, an intent is defined as a set of goals that a network should achieve, specified without describing the low-level procedures required to implement them [20]. This definition is important because it distinguishes intent from traditional device-centric configuration and from purely imperative policy specification. In other words, intent does not prescribe how the network must behave internally at each step, but rather states what outcome should be achieved and leaves the realization strategy to the IBN system [21].

This abstraction principle is one of the main reasons why IBN is regarded as a suitable paradigm for highly dynamic and large-scale communication systems. As discussed in recent surveys, IBN reduces operational complexity by raising the level of interaction between human stakeholders and the network, thus enabling management processes that are more closely aligned with service objectives than with individual device settings [22]. In practice, this means that network control is progressively shifting from low-level parameter provisioning toward intent-driven orchestration and assurance workflows, in which the network continuously interprets, enforces, monitors, and refines the desired state.

This view is also consistent with standardization efforts related to zero-touch and autonomous management. In particular, intent-based control is increasingly recognized as a key enabler for automation in network slicing and future mobile systems, where the network must react quickly to changing conditions while remaining aligned with operator objectives and service expectations [23]. Therefore, IBN

should not be interpreted as a single interface technology, but rather as a lifecycle-oriented management paradigm.

A useful way to make this perspective more explicit is to consider the intent lifecycle. Rather than being a one-shot request, an intent is expected to remain in effect over time and to be continuously evaluated against the observed network behavior [20]. From this viewpoint, an IBN system operates as a closed-loop process in which intents are acquired and interpreted, translated into operational policies and configurations, enforced through orchestration and provisioning mechanisms, and then continuously monitored to verify whether the resulting network state remains compliant with the original objectives [20], [23]. When deviations, conflicts, or changing conditions arise, the same lifecycle supports refinement, retranslation, optimization, or corrective actions, thereby limiting intent drift and preserving alignment between service goals and network operation over time [20].

From a system perspective, this lifecycle-oriented interpretation is particularly relevant for this thesis. It implies that the value of IBN lies not only in simplifying user-network interaction, but also in enabling a closed-loop management process in which high-level goals can be translated into operational behavior and continuously verified over time. Such a property is essential in NGNs, where service demands, resource availability, and environmental conditions may vary rapidly and require adaptive responses that remain aligned with the original intent.

1.1.2 Intent expression, translation, and lifecycle management

One of the central research problems in IBN concerns the semantic gap between the way users express their objectives and the way the network internally represents, interprets, and enforces them. To bridge this gap, the literature has proposed multiple approaches for intent acquisition and translation, ranging from template-based interaction and domain-specific abstractions to Natural Language Processing (NLP)-driven interfaces and conversational agents [24], [25], [26], [27]. Although these approaches differ in design and operational assumptions, they share a common goal: enabling users to formulate requests in a human-friendly manner while still producing structured representations suitable for automated orchestration.

Early work on intent refinement had already highlighted that high-level and often underspecified user objectives cannot be directly mapped into low-level configurations without intermediate processing steps [24]. This insight remains highly relevant, since the value of IBN depends not only on abstraction itself, but also on the ability of the system to progressively transform abstract service expectations into operationally meaningful representations. In this sense, intent handling is not a single translation act, but a structured process in which semantic interpretation, refinement, validation, and enforcement are tightly coupled.

Building on these premises, more recent studies have explored chatbot-based and natural-language-based interfaces for enterprise networks and intent-driven service management. Systems such as Lumi and Rasa-based intent translators show

that user requests can be acquired through conversational interactions and progressively mapped into structured intents by extracting entities, classes, contextual constraints, and service-related semantics from free text [25], [26], [28]. These developments are especially important because they demonstrate that the usability of IBN strongly depends on the front-end interaction layer and on the robustness of the semantic extraction mechanisms that connect human expressions with machine-processable intent representations.

Beyond user-friendly interaction, another major research direction concerns the translation of high-level intents into policies, configurations, and orchestration actions. Several works have addressed this problem in specific operational contexts, including Quality of Service (QoS) control, adaptive routing, network slicing, and multi-domain infrastructure management [29], [30], [31]. In these solutions, the translation block plays a crucial mediating role between the abstract language of user goals and the concrete language of network operations. As a result, ambiguities, incomplete interpretations, or modeling errors introduced at this stage may propagate downstream and compromise deployment correctness, resource allocation, and service assurance.

The emergence of LLMs has further expanded the design space of intent translation. Recent studies have investigated LLMs for policy generation, configuration assistance, semantic interpretation of open-ended service requests, and automated generation of network actions from high-level descriptions [32], [33], [34], [35], [36]. Their appeal lies in the ability to capture richer semantic information than traditional rule-based pipelines, potentially including contextual preferences, latent constraints, and higher-level service expectations. This is particularly relevant in advanced orchestration scenarios, where requests may involve heterogeneous objectives and may not fit rigid intent templates.

At the same time, the same literature makes clear that LLMs do not solve the intent translation problem by themselves. Their outputs may still suffer from ambiguity, inconsistency, hallucination, limited controllability, and partial misalignment with the actual capabilities and constraints of the underlying network [33], [34], [37]. Consequently, while LLMs represent a major opportunity for more expressive and flexible intent-driven management, they also reinforce the need for validation, controllability, and assurance mechanisms that can constrain and verify the generated outputs before operational enforcement.

This perspective naturally connects intent translation with the broader lifecycle-oriented view of IBN. Once acquired and interpreted, intents must be translated into executable policies, resolved against resource constraints and possible conflicts, activated through orchestration mechanisms, and continuously monitored to verify that the observed network behavior remains aligned with the desired objectives [20], [23]. Therefore, intent expression and translation should not be regarded as isolated front-end tasks, but as foundational phases of a broader closed-loop management process in which semantic understanding, execution, and assurance remain tightly interconnected over time.

1.1.3 Security in intent-driven systems

Security is increasingly recognized as a key requirement of IBN, particularly because the paradigm introduces a new human-to-network interaction layer and pushes a larger portion of control logic into software-driven and AI-assisted components. Existing research has shown that intents can be effectively used to express security-oriented requirements and support the deployment of secure network services. Representative examples include intent-based cloud security services, intent-driven security provisioning for network slicing, data protection intents, conformance checking for security policies, and optimization-based placement of security-oriented network functions [38], [39], [40], [41], [42], [43]. These contributions clearly demonstrate that IBN is well suited to encode and automate security objectives at the operational level.

Another line of work investigates trust-preserving mechanisms for secure policy realization, including blockchain-based approaches and accountability-oriented frameworks for policy enforcement [44]. Such studies reinforce the idea that IBN can improve operational security by reducing configuration errors, centralizing policy interpretation, and accelerating the deployment of preventive and reactive countermeasures. In this sense, security has often been treated as a property that can be specified through intents and automatically enforced by the orchestration logic.

However, the same architectural features that make IBN attractive also introduce new vulnerabilities. Because intents are expressed at a high level and then translated into concrete operational actions, errors or manipulations at the entry point of the lifecycle may propagate across the entire control process. This issue becomes even more relevant when AI-assisted or LLMs-assisted components are involved in semantic interpretation, since the flexibility of natural-language interaction may increase ambiguity, reduce predictability, and expose the system to malformed, deceptive, or adversarial requests. More broadly, recent research and standardization efforts highlight that centralized control logic, semantic misinterpretation, interface manipulation, and intent conflict propagation are all relevant security concerns in intent-driven systems.

A crucial limitation of the current state of the art is that most security-oriented IBN works focus on translating and deploying security requirements, rather than on validating the legitimacy of the incoming intent. In many cases, the input request is implicitly assumed to be well-formed, benign, and trustworthy, while the research focus is placed on how to deploy it correctly once it has already been accepted. This assumption is problematic in realistic enterprise, multi-tenant, and automated service environments, where malicious or deceptive requests may intentionally attempt to trigger privilege escalation, service disruption, unsafe reconfiguration, or unauthorized access to sensitive resources.

1.1.4 From KPI-centric orchestration to KVI-aware network management

While security represents one major research challenge, another equally important limitation of current IBN and orchestration frameworks concerns their predominantly KPI-centric nature. In most service management and orchestration solutions,

network decisions are still driven by indicators such as throughput, latency, packet loss, utilization, acceptance ratio, or SLA compliance. These metrics remain essential because they quantify technical efficiency and service quality. However, they are not sufficient to characterize the broader value that communication systems are expected to generate in NGNs. As a consequence, the literature has progressively introduced KVIs as complementary indicators that capture dimensions not adequately represented by conventional KPIs. In this perspective, KVIs extend the objective space of network design and management by making it possible to reason more explicitly about societal, ethical, and environmental outcomes.

A first research direction has investigated sustainability-aware optimization in communication networks through energy-efficient routing, resource allocation, service placement, and green orchestration strategies. Representative examples can be found in cellular, edge, cloud, and Internet of Things (IoT) scenarios, where optimization models aim at reducing energy consumption, carbon emissions, or related operational costs [45], [46], [47], [48], [49], [50], [51]. These studies are important because they show that environmental concerns can be explicitly incorporated into network decision-making, rather than treated only as ex-post reporting dimensions.

A similar trend can be observed in multimedia and video-delivery systems, where the environmental impact of data-intensive services has motivated research on energy-aware streaming, caching, transcoding, and cloud-network optimization [52], [53], [54], [55], [56], [57], [58]. In these works, sustainability is typically addressed through application-level adaptation, path selection, or compute-resource optimization. Nevertheless, in most cases it is modeled as a narrow optimization target, often limited to energy efficiency or carbon reduction, without being embedded into an end-to-end intent lifecycle and without being jointly handled with richer service semantics and broader societal objectives [8], [59].

Taken together, these studies confirm that the field is still fragmented. On the one hand, most existing solutions focus on one value dimension at a time, especially energy efficiency. On the other hand, secure intent processing, semantic translation, and value-aware orchestration are still often addressed as separate problems rather than as tightly integrated components of a unified end-to-end framework. This gap motivates the need for orchestration architectures capable of jointly handling technical performance, societal value, and trustworthy intent operationalization.

1.2 Architecture for Secure Intent Translation

Figure 1.1 illustrates the general intent-based architecture proposed in this thesis and highlights the functional blocks that connect secure request processing, value-aware intent translation, multi-objective optimization, and runtime orchestration support. The proposed architecture builds upon the intent lifecycle consolidated in the literature, whereby a high-level service requirement or operational objective expressed by a requester is progressively processed through intent profiling, translation, resolution, activation, and assurance functions. Within this perspective, requests may originate from heterogeneous actors, including end users, network operators, and administrators, who interact with the system through high-level interfaces such as applications, templates, chatbots, Graphical User Interfaces (GUIs),

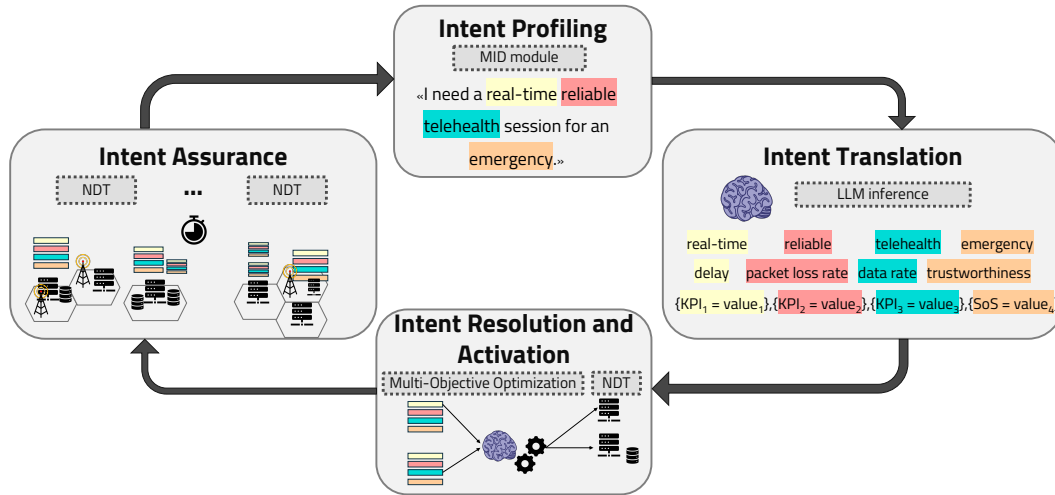


Figure 1.1: Proposed intent-based architecture [60].

or Application Programming Interfaces (APIs), rather than by directly configuring low-level network mechanisms.

More specifically, once a request is submitted, the architecture performs an initial profiling and information extraction step aimed at identifying the relevant entities, contextual elements, constraints, and service objectives expressed in natural language. This processing stage prepares the semantic content required for the subsequent formulation of the intent and makes explicit the transition from user-oriented expressions to machine-processable representations. When needed, this interpretation process may also include an intent confirmation step, through which the requester is asked to validate or refine the inferred objective before it is forwarded to deployment-oriented functions. In this way, the architecture does not treat the request as a simple input string to be translated, but as a progressively refined specification that becomes increasingly precise along the intent lifecycle.

Building on the research gaps identified in the previous section, the architecture proposed in this thesis extends this conventional lifecycle by enriching the processing and enforcement chain with security-, value-, and optimization-oriented capabilities. A first extension concerns the request-processing phase, where a Malicious Intent Detection (MID) step is introduced before full intent translation in order to analyze whether the submitted request is harmful, deceptive, or otherwise unsuitable for admission into the orchestration pipeline. This design choice addresses a critical risk of highly automated IBN systems, namely that malicious natural-language requests may otherwise be translated into disruptive configurations or policies and subsequently enforced by the infrastructure itself.

A second extension concerns the intent-translation stage, whose output is no longer limited to a purely technical description of the requested service, but is instead represented as a structured intent enriched with both conventional KPIs and broader KVIs. More specifically, the value-driven formulation adopted in this thesis enables the extraction of performance-oriented parameters, such as delay, data rate, and packet loss, together with value-oriented dimensions related to trustworthiness, inclusiveness, and environmental sustainability. In this phase, an LLM can support the interpretation of natural-language requests by extracting the relevant

semantic elements and mapping them into explicit parameters suitable for orchestration and optimization.

After translation, the architecture enters the intent resolution and activation phases, where the system derives feasible decisions under both performance and value constraints. At this stage, the thesis adopts a multi-objective optimization perspective, and the software support for this phase is provided by TETREES [61], a modular Python framework built on Gurobi that makes it possible to compute deterministic non-dominated solutions and to analyze the trade-offs between conflicting objectives in a reproducible way.

When the target scenario requires awareness of dynamic topology, residual capacities, energy conditions, or emission profiles, the same orchestration loop can be further supported by a NDT. In this case, the NDT provides an updated virtual representation of the infrastructure and assists both orchestration and assurance by exposing the state information needed to enforce adaptive and context-aware decisions. This is particularly relevant in sustainability-oriented scenarios, where the intent may include carbon-efficiency requirements that depend on time-varying operational and environmental conditions. Finally, the activated intent is monitored through an assurance loop that verifies whether the deployed configuration remains aligned with the original objectives and triggers corrective or adaptive actions when deviations emerge.

In this sense, the proposed architecture addresses RQ1 introduced in the thesis introduction by defining a general intent-processing framework in which high-level requests are progressively transformed into enforceable orchestration decisions through secure admission, semantic translation, multi-objective resolution, and runtime adaptation. Rather than viewing intent handling as a simple translation-and-deployment pipeline, the architecture formalizes it as an integrated process. The remainder of the chapter develops this answer through the specific contributions of the thesis, namely the MID module for secure request admission, the value-driven KPI/KVI translation framework, the *TETREES* optimization engine, and the NDT-aided sustainability-oriented orchestration case study.

1.2.1 Secure intent admission

The first specific contribution developed within the proposed architecture concerns secure request admission. Since high-level requests may contain harmful, deceptive, or adversarial formulations, they cannot be assumed to be trustworthy before entering the orchestration pipeline. To address this issue, this thesis introduces the MID module as a protection mechanism operating at the beginning of the intent lifecycle.

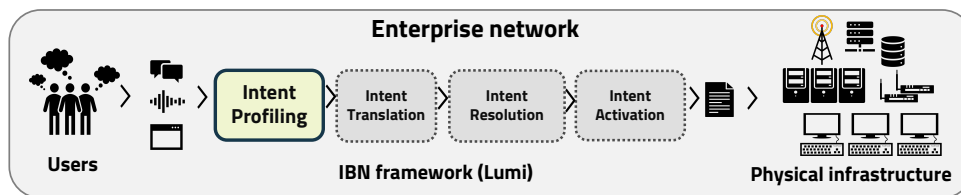


Figure 1.2: The considered enterprise network environment [62].

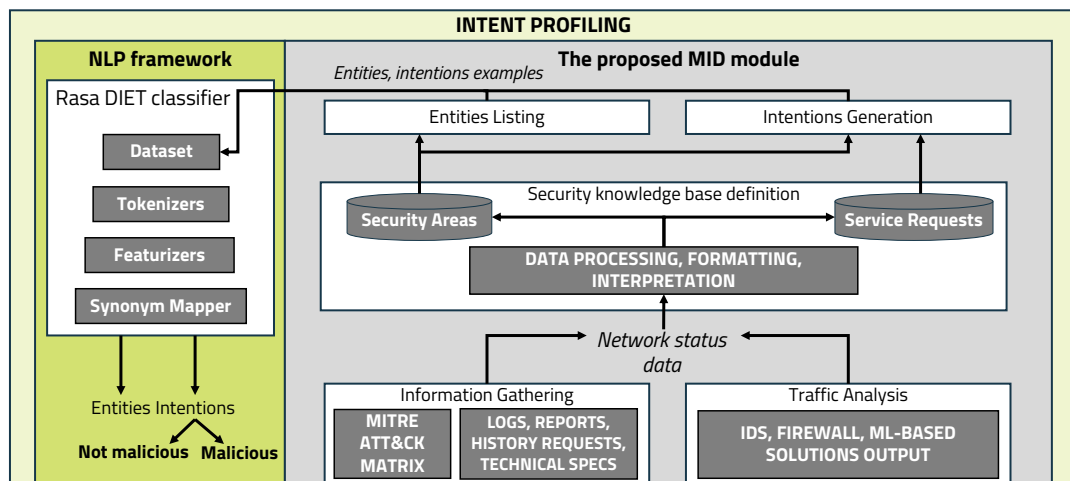


Figure 1.3: The proposed Malicious Intent Detection mechanism within the enterprise intent-processing pipeline [62].

Fig. 1.2 illustrates the enterprise network environment considered for the design and validation of the MID, where users or network operators express requests to access, configure, or protect enterprise resources and services [62]. Such requests are formulated through high-level interfaces and expressed in natural language rather than through low-level commands. This interaction paradigm allows users to specify the desired network behavior, possibly including quality, temporal, or spatial constraints, while delegating the derivation of deployable actions to the IBN system.

In the considered scenario, the enterprise infrastructure is modeled as a controlled operational environment, such as a corporate LAN protected by a gateway and a firewall and exposing a regulated interface toward external networks. Within this setting, the intent-processing chain comprises four main functions, namely *Intent Profiling*, *Intent Translation*, *Intent Resolution*, and *Intent Activation*.

Upon submission of a request, *Intent Profiling* acquires the user utterance and extracts the relevant entities and contextual information through an NLP framework, thus preparing the request for structured interpretation. The resulting information is then forwarded to *Intent Translation*, where the request is reformulated into an IBN language. In the considered implementation, the selected representation is Network Intent Language (NILE), which converts the profiled content into a structured intent characterized by a name and a set of tags defining its scope. Subsequently, *Intent Resolution* verifies whether the translated intent can be consistently admitted with respect to the operational conditions of the enterprise network, possibly including an interaction loop for user confirmation or refinement. Finally, *Intent Activation* compiles the resolved intent into network configurations or policies to be enforced in the physical infrastructure. Within this flow, the proposed MID mechanism operates during *Intent Profiling* so as to assess request trustworthiness before translation. In this way, harmful or deceptive requests can be intercepted at an early stage, and only trusted inputs are propagated to the subsequent orchestration functions. In the considered implementation, Lumi[®] is adopted as the reference

[®]<https://lumichatbot.github.io/>

IBN framework, while a Rasa-based conversational interface is used to acquire and interpret user requests.

1.2.1.1 Rasa-based Intent Profiling chatbot

Rasa is an open-source framework tailored for the creation of adaptable and scalable chatbots, employing advanced NLP and machine learning techniques. In the considered solution, it implements the conversational front-end through which enterprise users articulate their requests in natural language and supports the profiling of such requests before they are transformed into formal intents. The choice of Rasa is motivated by its flexibility and by the possibility of customizing its components to reflect the needs of the protected enterprise environment.

Within the proposed architecture, the Rasa-based chatbot supports the early stages of the intent lifecycle by interpreting and categorizing user expressions. To this end, Rasa provides models and components specifically designed to extract pertinent information from user inputs, including entities, synonyms, and intention classes. These intention classes describe the meaning or goal of an utterance and may capture conversation initiation or termination, request confirmation or rejection, operational service requests, and anomalous utterances that do not belong to the expected classes.

A relevant feature is the use of the *out-of-scope* class. In standard conversational systems, this class is used to collect utterances that do not match the defined intent classes. In the present work, this class is deliberately repurposed so as to capture requests that are classified as malicious or harmful for the enterprise network, thereby turning a conversational fallback mechanism into a security-oriented classification outcome. The proposed implementation also incorporates the Dual Intent and Entity Transformer (DIET) classifier, a state-of-the-art multi-task model capable of jointly performing entity recognition and intent classification. By relying on advanced tokenization techniques, the DIET classifier improves the extraction of features that are relevant to both tasks. The classifier further exploits the Synonym Mapper component, which supports direct associations between alternative expressions referring to operations, services, locations, protocols, or constraints and their predefined entities or intents. To accurately recognize entities and classify user utterances, the adopted classifier must be trained with a dataset containing examples of enterprise requests grouped according to the corresponding intention classes. In the considered implementation, examples of user expressions include statements such as “Set up an anti-phishing filter on the email server”, “Assign network administrator rights to the Operations, Administration, and Management (OAM) group”, and “Enable two-factor authentication for all network management accounts.” These examples show that the conversational layer must capture not only the requested action, but also the involved service, target, user group, or security control.

The flexibility and customizability of Rasa make it particularly suitable for extending the profiling stage of the IBN lifecycle with security-aware capabilities. In particular, this work exploits such flexibility to support the recognition and explicit classification of malicious user requests. As a consequence, the chatbot does

not merely extract semantic information for downstream translation, but also contributes to deciding whether the request should be admitted into the subsequent lifecycle stages.

1.2.1.2 Malicious Intent Detection module

To detect malicious behavior during the early stages of the lifecycle, this work customizes *Intent Processing* by placing the MID mechanism alongside the NLP framework used by the chatbot. As shown in Fig. 1.3, the proposed module is organized into five main functions: (i) Information Gathering, (ii) Traffic Analysis, (iii) Security Knowledge Base Definition, (iv) Entities Listing, and (v) Intentions Generation. Together, these functions create the knowledge and training artifacts required to distinguish legitimate operational requests from harmful ones.

The *Information Gathering* and *Traffic Analysis* functions are tightly coupled with the status of the enterprise infrastructure. Their goal is to collect information that reflects the security posture of the monitored environment, including assets, vulnerabilities, historical service requests, and potential attack evidence. This makes the malicious intent detection process aware of the concrete security context of the protected enterprise network rather than purely dependent on generic natural-language patterns. More specifically, *Information Gathering* acquires internal data from the enterprise environment, including logs, reports, service request histories, and technical specifications obtained from servers, devices, and virtual machines. These data sources provide a structured view of the enterprise infrastructure and of the requests that have already been observed within the system. In addition, external security-support tools can be leveraged to help identify weaknesses and attack patterns relevant to the considered deployment scenario. Among such tools, the MITRE ATT&CK Enterprise Matrix plays a particularly important role. It provides a structured view of attacker tactics and techniques and supports enterprises in understanding, analyzing, and mitigating security threats. Each matrix cell is associated with a specific technique, and the matrix adopts color-coded indicators to represent the frequency and severity of the corresponding attack behaviors. This makes it suitable as a reference source for contextualizing how malicious users or attackers may attempt to formulate disruptive requests. The *Traffic Analysis* function complements internal information by processing network-level observations obtained from Intrusion Detection Systems (IDSs), firewalls, and machine-learning-based anomaly detectors. In the considered implementation, these detectors include Naïve Bayes classifiers, decision trees, and Support Vector Machine (SVM), selected because of their interpretability and fast training behavior. Their outputs are used to identify anomalies in traffic traces and flows and to enrich the knowledge used for malicious intent detection with evidence derived from actual network behavior. The information collected from the previous functions is then processed by the *Security Knowledge Base Definition* function. Its role is to combine heterogeneous data gathered from multiple sources into a unified and structured representation that can support the classifier in taking informed decisions. To this end, the collected data undergo refinement operations such as filtering, normalization, and aggregation before being encoded through descriptive formats and standards including XML, JSON, and Structured Threat Information eXpression (STIX). The

adoption of STIX-compatible representations is particularly useful because it enables structured handling of threat-related information, such as indicators of compromise, tactics, techniques, procedures, and other security observables. On top of this normalized data pool, clustering techniques such as k-means and HDBSCAN are used to identify patterns, interpret the collected knowledge, and detect new observations with respect to the existing data. This supports the assessment and quantification of risks and vulnerabilities and helps characterize the mechanisms through which adversaries may gain unauthorized access to enterprise resources. The resulting knowledge base is organized into two distinct repositories. The first repository stores security areas together with their associated threats and targets, thereby representing a structured model of the enterprise security domain. The second repository stores service requests previously observed in the enterprise environment. This dual-database organization makes it possible to jointly reason on what should be protected and on how users tend to formulate benign or malicious requests. On top of this knowledge base, the *Entities Listing* function generates the domain-specific artifacts needed to specialize the chatbot to the enterprise environment. By exploiting the information stored in the Security Areas repository and leveraging Named Entity Recognition (NER) algorithms, this function produces entities and synonyms that populate or refine the chatbot configuration files. In practical terms, the output of this function is used to enrich the Rasa configuration so that the conversational layer can recognize enterprise-relevant concepts with the level of granularity needed by the downstream classifier. The *Intentions Generation* function instead focuses on the creation of request examples to support intent classification. In the proposed implementation, Generative Adversarial Networks (GANs) are applied to the data stored in the two repositories to generate conceivable malicious requests that can be used as training examples. This choice is meant to expand the diversity of harmful utterances seen during training and to improve the ability of the classifier to identify malicious expressions that undermine the integrity of the enterprise network. A crucial implementation choice concerns the way in which intention classes are modeled inside the chatbot. Since the Rasa-based flow does not natively include a dedicated malicious-intent class, this work defines a legitimate operational request class denoted as *build-intention* and reuses the *out-of-scope-intention* class to collect all user expressions labeled as malicious. In this way, benign service requests and harmful requests are clearly separated at classifier level, while still remaining compatible with the standard operation of the conversational platform.

The artifacts generated by the MID module are then used to train and refine the DIET classifier that supports the profiling stage. Once a request is submitted by a user, the chatbot extracts entities and classifies the corresponding intention. If the request is mapped to the malicious class, the chatbot returns the predefined response associated with the *out-of-scope-intention* category, and the request is blocked and discarded. Conversely, only the contextual information extracted from legitimate requests is forwarded to the subsequent lifecycle stages, starting from *Intent Translation*. In this way, the proposed mechanism operates as a security gate that prevents harmful requests from being translated into deployable intents and propagated toward resolution and activation.

1.2.1.3 Performance Evaluation

The performance of the proposed approach in detecting harmful requests within an enterprise network is investigated herein. More specifically, the evaluation focuses on the ability of the MID-enhanced profiling stage to correctly extract entities, classify user intentions, and intercept malicious requests before they are propagated.

A comprehensive definition of the MID module's Security Knowledge Base has been synthesized to elucidate entities and malicious intentions. The resulting knowledge captures security areas relevant to the physical infrastructure of the enterprise network, including antispam, antivirus, authentication, authorization, access control, malware and data loss prevention, encryption, intrusion detection, network segmentation, patching, hardening, and backup. Each security area is correlated with its associated threats and targets and contributes to shaping the semantic space used by the classifier.

Starting from this knowledge base, 12 types of entities have been delineated, namely traffic, locations, middleboxes, operations, groups, services, QoS constraints, QoS values, QoS units, hours, protocols, and values. Examples of these entities, together with their potential synonyms, have been cataloged in the `domain.yml` and `lookups.yml` configuration files of the Rasa framework. In addition, the `domain.yml` file specifies the entities that must be detected and classified by the DIET classifier during the profiling stage. Potential malicious intentions have then been inferred from the delineated threats and encoded as training expressions for the intent-classification task. These malicious expressions are stored in the `intents.yml` configuration file of the Rasa framework and are used to populate the malicious class represented through *out-of-scope*. In total, 196 malicious requests were defined and complemented with examples from the other intention classes, including those related to initiating, modifying, or concluding a conversation. Apart from the synthesized dataset used to train the DIET classifier embedded in the profiling stage, the evaluation also considers 755 user requests sourced from interactions with the enterprise system chatbot. These requests are processed through the proposed MID-enhanced conversational layer in order to assess whether harmful intentions can be reliably detected and filtered before the requests are translated into enforceable intents.

The influence of the MID module on the DIET classifier was examined using k -fold cross-validation. The dataset was partitioned into K subsets of comparable size, referred to as folds. The classifier was then trained iteratively K times, each time using one fold as validation data and the remaining $K - 1$ folds as training data, and the final performance was obtained by averaging the results over all iterations. Due to the limited size of the produced dataset, the number of folds was set to 3.

To describe the classification efficacy for both entities and intentions, Fig. 1.4 reports the confusion matrices obtained from the DIET classifier. These matrices show the number of detected instances, including false positives and false negatives, across all entity types and intention classes. In this representation, rows correspond to predicted classes, whereas columns correspond to actual entities and user intentions. The reported results show that 234 instances of the *build* intention were correctly classified out of 246 instances. In addition, the classifier successfully identified 265 malicious intentions out of 284 by assigning them to the *out-of-scope*

group	131		4							14		
hour		20										
location			445							17		
middlebox				321	12					10		
operation				4	716							
protocol						180						
qos_constraint						2	82			4		
qos_unit								52	19			
qos_value									13			
service					8					542		
traffic							6			11	277	
value												13

True Class

group hour location middlebox operation protocol qos_constraint qos_unit qos_value service traffic value

Predicted Class

(a) Entities confusion matrix.

welcome	27								
cancel		20				2	1		
thanks			24			1			
goodbye				18	2	2			
reject					44				
confirm			2			59			5
build-feedback							18		
build								234	12
out_of_scope							1	18	265

True Class

welcome cancel thanks goodbye reject confirm build-feedback build out_of_scope

Predicted Class

(b) Intentions confusion matrix.

Figure 1.4: Confusion matrices of entities and intentions [62].

class. Consequently, only 19 malicious intentions were not intercepted by the proposed MID mechanism and therefore escaped malicious classification.

To accurately translate users' expressions, the conversational profiling layer must first extract and recognize the involved entities. In this respect, the classifier achieves an accuracy of 95%, with an average precision of 94%, an average recall of 92%, and an average F1-score of 92%. While the extraction performance is weaker for a few entity types, such as *Value*, the classifier consistently captures the information needed to support request understanding and downstream intent processing.

To provide further insight into the reliability of the classification outcomes, Fig. 1.5 depicts the distribution of the intent prediction confidence score. This score ranges from 0 to 1, with higher values indicating a stronger alignment between the

Table 1.1: Precision, Recall, F1-score for the entity extraction task.

Entity	Precision	Recall	F1-score
Traffic	0.96	0.97	0.96
Location	0.96	0.94	0.95
Middlebox	0.92	0.90	0.91
Operation	0.95	0.96	0.95
Group	0.97	0.88	0.92
Service	0.94	0.91	0.92
Hour	0.94	0.80	0.86
Value	1	0.08	0.14
QoS constraint	0.77	0.86	0.81
QoS value	0.84	1	0.91
QoS unit	1	0.95	0.98
Protocol	0.98	0.96	0.97

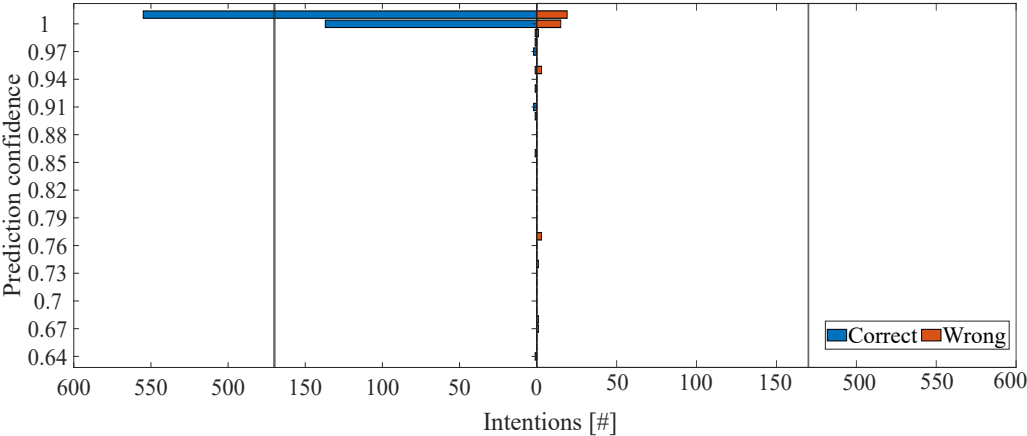


Figure 1.5: Confidence distribution of predicted intentions.

predicted class and the declared intention. The reported results show a confidence level above 0.98 for correctly classified intentions, with 555 instances correctly identified at a confidence score equal to 1. By contrast, only 19 intentions reached the highest confidence score among the incorrectly classified instances.

The outcomes of the intention-classification task were evaluated through precision, recall, and F1-score, as detailed in Table 1.2. Precision measures the proportion of true positive predictions among all positive predictions made by the classifier. Recall measures the proportion of true positive instances that were correctly identified out of all actual positive instances. F1-score measures the harmonic mean of precision and recall and provides a balanced assessment of the classifier performance.

The *out-of-scope* class consistently exhibits high-performance metrics, which confirms the efficacy of the proposed MID mechanism in detecting malicious intentions. Specifically, the malicious class achieves a precision of 94%, a recall of 95%, and an F1-score of 94%. These values indicate that the proposed profiling-stage extension can effectively discriminate harmful requests from legitimate operational demands. The Rasa framework also provides the overall accuracy of the

Table 1.2: Precision, Recall, F1-score for the intentions classification.

Intent	Precision	Recall	F1-score
Welcome	0.96	1	0.98
Cancel	1	0.82	0.90
Thanks	0.92	0.96	0.94
Goodbye	1	0.77	0.87
Reject	0.96	1	0.98
Confirm	0.92	0.86	0.89
Build feedback	0.90	1	0.95
Build	0.92	0.95	0.93
Out-of-scope	0.94	0.95	0.94

intention-classification task, computed as the ratio between correct predictions and total predictions. Across the full set of 755 user requests, the observed results yield an average accuracy of 93% in correctly classifying intentions. This accuracy, together with the high metrics achieved by the *out-of-scope* class, substantiates the capability of the proposed approach to detect malicious intentions and discard them before they are propagated to the subsequent lifecycle stages, where the focus shifts to determining how the intent should be formally represented for value-aware orchestration.

1.2.2 Value-driven intent translation and orchestration

Once a request has been admitted as trustworthy, the next step is to translate it into a structured representation that can guide orchestration decisions. To this end, this thesis adopts a value-driven intent translation framework in which natural-language service requests are mapped into both conventional KPIs and broader KVIs [60]. In particular, the translation process identifies performance requirements such as service delay, data rate, and packet loss rate, together with value-oriented dimensions such as trustworthiness, inclusiveness, and environmental sustainability. In this way, intent translation is extended beyond purely technical targets and becomes able to capture a richer set of service objectives within a unified representation.

To bridge the semantic gap between user expressions and network-level decision variables, the framework relies on a general-purpose LLM that is instruction-tuned for the considered task. Given a natural-language request, the model extracts the relevant contextual information, infers the associated KPIs and KVIs, and produces a structured intent representation suitable for subsequent optimization. The generated output includes desired performance values and acceptable tolerances for the considered KPIs, while the value-oriented component captures the broader societal and ethical dimensions embedded in the request.

More specifically, the extracted performance-related parameters are mapped to traditional KPIs, including service delay, data rate, and packet loss rate, and are expressed through target values together with admissible tolerance thresholds. In parallel, trustworthiness, inclusiveness, and environmental sustainability are modeled as KVIs, thereby extending the service specification beyond purely technical

performance aspects and aligning intent translation with wider societal objectives reflected in the UN SDGs [9], [63]. In particular:

- the trustworthiness dimension is intended to capture network security, robustness, and privacy, thus contributing to the support of resilient technological infrastructures;
- the inclusiveness dimension addresses fair access to digital technologies, the mitigation of the digital divide, and the empowerment of heterogeneous communities;
- the sustainability dimension reflects the ecological perspective by promoting energy efficiency, supporting economic development, and reducing environmental impact.

The resulting structured intent is then used as input to the subsequent orchestration stage, where resource allocation is determined by jointly accounting for network quality and social and ethical value. The adopted system model and the corresponding optimization problem are introduced below.

1.2.2.1 System Model

Building on the intent-based system design introduced above, this work considers a communication infrastructure in which services are delivered by a set of network resources denoted as $\mathcal{R} = \{r_1, r_2, \dots, r_n, \dots, r_N\}$. Each resource $r_n \in \mathcal{R}$ is associated with a capability profile $(C_{r_n}, y_{1_{r_n}}, y_{2_{r_n}}, \dots, y_{i_{r_n}}, \dots, y_{I_{r_n}})$, where C_{r_n} represents the number of cores of the r_n -th resource available for service execution, while $y_{1_{r_n}}, y_{2_{r_n}}, \dots, y_{i_{r_n}}, \dots, y_{I_{r_n}}$ denote the exposed performance attributes used to evaluate the considered KPIs, such as computation delay, data rate, and packet loss rate.

Users generate service requests of different types, represented by the set $\mathcal{S} = \{s_1, s_2, \dots, s_j, \dots, s_J\}$. Each requested service $s_j \in \mathcal{S}$ is described by a tuple of KPIs, $(D_{s_j}, y_{1_{s_j}}, y_{2_{s_j}}, \dots, y_{i_{s_j}}, \dots, y_{I_{s_j}}, y'_{1_{s_j}}, y'_{2_{s_j}}, \dots, y'_{i_{s_j}}, \dots, y'_{I_{s_j}})$, where D_{s_j} denotes the amount of processing resources required to execute the s_j -th service, under the assumption that each resource corresponds to a single service task and each task requires one processing core. The parameters $y_{1_{s_j}}, y_{2_{s_j}}, \dots, y_{i_{s_j}}, \dots, y_{I_{s_j}}$ denote the desired performance levels, whereas $y'_{1_{s_j}}, y'_{2_{s_j}}, \dots, y'_{i_{s_j}}, \dots, y'_{I_{s_j}}$ represent the minimum acceptable thresholds for the corresponding KPIs.

Starting from these thresholds, the model introduces $Q_{tol_{s_j}}$ as the overall quality tolerance associated with a service request. This parameter quantifies the admissible deviation from the ideal requested performance and takes values in the interval $0 \leq Q_{tol_{s_j}} \leq 1$. It is computed as

$$Q_{tol_{s_j}} = \sum_{i=1}^I w_i \cdot \hat{y}_{i_{s_j}}, \quad (1.1)$$

where w_i is the weight associated with the i -th KPI, with $\sum_{i=1}^I w_i = 1$ and $0 \leq w_i \leq 1$. In addition, $\hat{y}_{i_{s_j}}$ denotes the normalized value of the i -th KPI for service request s_j , obtained through min-max normalization as follows:

$$\hat{y}_{i,s_j} = \frac{y'_{i,s_j} - \min(y'_i)}{\max(y'_i) - \min(y'_i)}$$

where $\max(y'_i)$ and $\min(y'_i)$ denote the maximum and minimum observed values of the i -th KPI over all service requests.

Since different performance attributes may be measured according to heterogeneous standards, the model defines the normalized performance quality associated with the i -th attribute of resource r_n when serving request s_j as $\hat{q}_{i,r_n,s_j}$, constrained in the interval $[0, 1]$. This quantity is evaluated according to the standard normalization model in [64] as follows:

$$\hat{q}_{i,r_n,s_j} = \begin{cases} 1 - \frac{\max(y_{i,r_n}) - y_{i,s_j}}{\max(y_{i,r_n}) - y'_{i,s_j}}, & \text{if the KPI is a benefit,} \\ 1 - \frac{y_{i,s_j} - \min(y_{i,r_n})}{y'_{i,s_j} - \min(y_{i,r_n})}, & \text{if the KPI is a cost,} \\ 0, & \text{otherwise.} \end{cases} \quad (1.2)$$

The overall KPI value, denoted by Q_{r_n,s_j} , represents the aggregate performance quality delivered by resource r_n to service s_j and is computed as

$$Q_{r_n,s_j} = \sum_{i=1}^I w_i \cdot \hat{q}_{i,r_n,s_j}. \quad (1.3)$$

Without loss of generality, Table 1.3 reports the main symbols used throughout the formulation.

1.2.2.2 Social and Ethical Values

The main advancement of the proposed approach lies in its capability to provide services that are aligned with social and ethical KVIs. In particular, three indicators, namely environmental sustainability, trustworthiness, and inclusiveness, are introduced to represent intangible yet fundamental societal values and to assess the contribution of communication networks to the SDGs.

The *environmental sustainability indicator* σ_{r_n,s_j} measures the climate impact of service provisioning, expressed in grams of carbon dioxide equivalent. This KVI reflects the global warming potential associated with the greenhouse gas emissions generated during the provisioning interval. The indicator also incorporates the greenhouse gas emission compensation associated with the r_n -th resource, achieved through the acquisition of carbon credits or offsets, where each unit corresponds to a tonne of gCO₂e reduced or removed from the atmosphere. Such compensation may be enabled by initiatives including reforestation, renewable energy deployment, or carbon capture projects [65]. This mechanism enables individuals, enterprises, and public bodies to mitigate their environmental footprint and move toward carbon neutrality, which is becoming increasingly important. Environmental sustainability, however, goes beyond carbon reduction alone and also includes metrics at the

Table 1.3: Main Symbols Description.

Symbol	Meaning
$\mathcal{R}$	Set of network resources
$\mathcal{S}$	Set of requested services
N	Number of network resources [#]
J	Number of services [#]
C_{r_n}	Cores of the r_n -th resource
D_{s_j}	Resource demand of the j -th service [#]
$y_{i_{r_n}}$	Exposed performance attribute of the r_n -th resource
$y_{i_{s_j}}$	Desired performance parameter of the s_j -th service
$y_{i_{s_j}}^{\min}$	Minimum tolerable value for $y_{i_{s_j}}$ parameter
$q_{i_{r_n}, s_j}$	Normalized performance quality of i -th attribute for r_n facing s_j
$Q_{tol_{s_j}}$	Overall quality tolerance of the s_j -th service request
Q_{r_n, s_j}	Overall quality KPI for r_n providing the s_j -th service
V_{r_n, s_j}	Overall social and ethical KVI of r_n providing s_j
w_1, w_2, w_3	Weights for the KPIs relevance
w'_1, w'_2, w'_3	Weights for the KVIs relevance
σ_{r_n, s_j}	Environmental sustainability indicator
$t_{comp}^{r_n, s_j}$	Computing delay of the r_n -th resource facing the s_j -th service [s]
B_{s_j}	Input size of the data to be processed for the s_j -th service request [Mb]
Clk_{r_n}	Processing capability of the r_n -th resource [Megacycles/s]
λ_{r_n, s_j}	Average number of service requests that r_n can handle per day
co_{r_n}	Carbon offset associated with the r_n -th resource [gCO ₂ e per day]
$u_c^{r_n}$	Core usage factor of the r_n -th resource
$P_c^{r_n}$	Power consumption of a single core of the r_n -th resource [kW]
$P_m^{r_n}$	Power consumption of memory of the r_n -th resource [kW]
$PowerUsageEffectiveness(PUE)$	Power usage effectiveness coefficient
CI	Carbon intensity factor [gCO ₂ e/kWh]
θ_{r_n, s_j}	Trustworthiness indicator
ρ_{r_n, s_j}	Cyber risk of r_n providing s_j
κ_{r_n, s_j}	Cyber confidence associated with r_n facing s_j -th service
l_{r_n, s_j}	Inclusiveness indicator
$MTTF_{r_n}$	Mean time to failure of the r_n -th resource [h]
p_{r_n}	Probability of failure of the r_n -th resource within a day
x_{s_j, r_n}	Binary variable indicating if resource r_n provides service s_j
$Q(X)$	Network quality performance function
$V(X)$	Network social and ethical function

device, equipment, flow, path, and domain levels, as well as the prevention of different forms of environmental degradation [66]. In this setting, the adopted environmental sustainability indicator does not preclude the integration of complementary metrics capturing other ecological dimensions.

Formally, the environmental sustainability indicator is defined as the difference between the carbon offset co_{r_n} of the r_n -th resource, characterized by a carbon intensity factor CI expressed in gCO₂e/kWh, and the carbon footprint associated with provisioning service s_j .

For each $r_n \in \mathcal{R}$, the time required by the r_n -th resource to process service s_j , denoted as computing delay $t_{comp}^{r_n, s_j}$, is calculated as follows:

$$t_{comp}^{r_n, s_j} = \frac{\Phi \cdot B_{s_j}}{Clk_{r_n}}, \quad (1.4)$$

where Φ denotes the number of CPU cycles needed to process one bit. According to [67], this value is set to 1000 cycles per bit. Moreover, B_{s_j} denotes the input data size to be processed for the requested service, while Clk_{r_n} represents the processing

capability of the r_n -th resource, expressed in $[cycles/s]$.

Given λ_{r_n, s_j} , namely the average number of service requests s_j that the r_n -th resource can serve per day, the environmental sustainability indicator can be computed as follows [68]:

$$\sigma_{r_n, s_j} = co_{r_n} - [t_{comp}^{r_n, s_j} \cdot \lambda_{r_n, s_j} \cdot C_{r_n} \cdot (P_c^{r_n} \cdot u_c^{r_n} + P_m^{r_n}) \cdot PUE \cdot CI], \quad (1.5)$$

where $P_c^{r_n}$ denotes the power consumption of a single processing core, and $P_m^{r_n}$ denotes the power consumption of memory, both expressed in kilowatts (kW). The core usage factor $u_c^{r_n}$, ranging from 0 to 1, expresses the fraction of the total execution time during which a processor core is actively involved in computation. A value equal to 1 indicates full and continuous utilization throughout the entire execution interval, whereas lower values account for idle periods or reduced workload. This parameter makes it possible to correct theoretical power consumption values, often derived from processor thermal design power, so as to better reflect actual operating conditions. When detailed usage information is unavailable or hard to obtain, setting $u_c^{r_n} = 1$ provides a conservative estimate that is likely to overestimate the actual energy consumption. Moreover, PUE denotes the power usage effectiveness coefficient, set to 1.67 as the globally averaged measured value [68], while CI represents the carbon intensity factor, which quantifies the equivalent carbon emissions associated with the consumed energy. The value of CI varies significantly across countries, reflecting differences in national energy mixes, with lower values generally associated with renewable-based systems and higher values with fossil-fuel-dependent contexts.

Secondly, the *trustworthiness indicator* θ_{r_n, s_j} measures the security level of the service provisioning offered by the r_n -th resource to the s_j -th service. As discussed in our previous work [69], successful cyber-attacks can severely affect network performance, with major consequences for both infrastructure operation and service reliability. It is therefore important to quantify the cyber risk threatening the confidentiality, integrity, and availability of the data and services associated with each r_n -th resource. Following the methodology presented in [69], the likelihood of a successful cyber attack, denoted by L_{r_n} , can be computed for each resource. Accordingly, the associated cyber risk ρ_{r_n, s_j} is defined as the product of L_{r_n} and the impact Δ_{s_j} of an attack on the provisioning of service s_j . The impact parameter Δ_{s_j} captures the potential consequences and disruptions caused by a successful cyber attack. Hence, the cyber confidence κ_{r_n, s_j} , expressing the confidence of the r_n -th resource in serving the s_j -th service, is formally computed as follows:

$$\kappa_{r_n, s_j} = 1 - \rho_{r_n, s_j}. \quad (1.6)$$

The indicator θ_{r_n, s_j} is modeled through a generalized logistic function that increases with cyber confidence. This formulation ensures that more resilient resources, characterized by stronger security mechanisms and lower cyber risk, attain higher values of θ_{r_n, s_j} . The use of a logistic function effectively captures the nonlinear relationship between cyber confidence and trustworthiness, as discussed in [70], and is expressed as follows:

$$\theta^{r_n, s_j} = L_\theta + \frac{U_\theta - L_\theta}{1 + e^{-B(\kappa_{r_n, s_j} - x_0)}}. \quad (1.7)$$

Here, L_θ denotes the lower bound of the trustworthiness attainable by a resource lacking strong security guarantees. This value may vary across providers, since even resources without a strong security profile may still exhibit a baseline level of trust due to third-party management, secure default settings, or compliance with regulatory requirements. Conversely, U_θ denotes the upper bound of achievable trust, acknowledging that even highly secure systems remain exposed to residual risks, including zero-day vulnerabilities and insider threats. The parameter B controls the steepness of the curve, determining how rapidly trust increases as the security posture improves. Higher absolute values of B produce a sharper transition around the inflection point, whereas lower values yield a smoother increase that may be more appropriate in settings where trust evolves over longer organizational or cultural timescales. The inflection point x_0 identifies the security level at which a resource starts to be perceived as trustworthy. This threshold may vary according to the complexity of the organization, since more complex infrastructures are usually exposed to a broader range of systemic threats. Importantly, highly vulnerable resources may show limited risk reduction after initial countermeasures, whereas mature and secure systems often experience diminishing returns from additional security investments. This behavior reflects realistic cybersecurity dynamics, where the effectiveness of interventions strongly depends on the initial resilience of the system and risk mitigation follows a nonlinear trajectory.

Furthermore, the *inclusiveness indicator*, denoted by ι_{r_n, s_j} , measures the capability of the network to provide adequate service to users, thus ensuring operational continuity and resilience for uninterrupted service functionality. In this work, inclusiveness is approximated by using availability as a proxy. This property is evaluated through the time-to-failure TF_{r_n} , modeled through an exponential distribution under the assumption of a constant failure rate. The parameter $MTTF_{r_n}$ denotes the Mean Time To Failure (MTTF), namely the average time between consecutive failures. This metric is essential for evaluating the reliability of the r_n -th provider. The probability that the r_n -th provider experiences failure or downtime within a given time period t is expressed as follows:

$$P(TF_{r_n} \leq t) = 1 - e^{-\frac{t}{MTTF_{r_n}}} \quad (1.8)$$

Let p_{r_n} denote the probability of failure of the r_n -th resource within one day. Assuming that the C_{r_n} available cores of the r_n -th resource fail independently with probability p_{r_n} , the probability of observing k failures within an hour follows a binomial distribution and is given by:

$$P(F_{r_n} = k) = \binom{C_{r_n}}{k} (p_{r_n})^k (1 - p_{r_n})^{C_{r_n} - k}, \quad k = 0, 1, \dots, C_{r_n} \quad (1.9)$$

Hence, the probability that no failures occur within an hour, denoted by $P(F_{r_n} = 0)$, is given by:

$$P(F_{r_n} = 0) = (1 - p_{r_n})^{C_{r_n}} \quad (1.10)$$

The inclusiveness indicator can therefore be expressed as follows:

$$\iota_{r_n, s_j} = t_{comp}^{r_n, s_j} \cdot \lambda_{r_n, s_j} \cdot (1 - p_{r_n})^{C_{r_n}} \quad (1.11)$$

Finally, the overall social and ethical KVI associated with the provisioning of service s_j by network resource r_n is defined as:

$$V_{r_n, s_j} = w'_1 \cdot \sigma_{r_n, s_j} + w'_2 \cdot \theta_{r_n, s_j} + w'_3 \cdot \iota_{r_n, s_j} \quad (1.12)$$

where w'_1 , w'_2 , and w'_3 are weighting coefficients in the interval $[0, 1]$, whose sum is equal to 1 and which express the relative importance of each indicator.

1.2.2.3 Problem Formulation and Solution

Let x_{r_n, s_j} be a binary variable indicating whether service request $s_j \in \mathcal{S}$ is assigned to network resource r_n , and let X denote the set of all binary indicator variables:

$$x_{r_n, s_j} = \begin{cases} 1 & \text{if } r_n \text{ provides service for } s_j \\ 0 & \text{otherwise.} \end{cases} \quad (1.13)$$

The network capability to provide services can be expressed through the overall quality KPI resulting from the matching between service requests and available resources, as follows:

$$Q(X) = \sum_{s_j \in \mathcal{S}} \sum_{r_n \in \mathcal{R}} Q_{r_n, s_j} \cdot x_{r_n, s_j}. \quad (1.14)$$

Similarly, the network ability to provide value-oriented and ethical services can be expressed through the total KVI resulting from the matching between service requests and available resources:

$$V(X) = \sum_{s_j \in \mathcal{S}} \sum_{r_n \in \mathcal{R}} V_{r_n, s_j} \cdot x_{r_n, s_j}. \quad (1.15)$$

For users requesting services, the benefit lies in obtaining satisfactory provisioning by jointly ensuring performance quality through KPIs and social and ethical value through KVIs. Accordingly, the following multi-objective optimization problem is formulated:

$$\max_X (Q(X), V(X)), \quad (1.16)$$

$$\text{s.t.} \quad (Q_{r_n, s_j} - Q_{tol_{s_j}}) \cdot x_{r_n, s_j} \geq 0, \forall s_j \in \mathcal{S}, \forall r_n \in \mathcal{R}, \quad (1.17)$$

$$V_{r_n, s_j} \cdot x_{r_n, s_j} \geq 0, \forall s_j \in \mathcal{S}, \forall r_n \in \mathcal{R}, \quad (1.18)$$

$$\sum_{s_j \in \mathcal{S}} D_{s_j} \cdot x_{r_n, s_j} \leq C_{r_n}, \forall r_n \in \mathcal{R}, \quad (1.19)$$

$$\sum_{r_n \in \mathcal{R}} x_{r_n, s_j} = 1, \forall s_j \in \mathcal{S} \quad (1.20)$$

$$x_{r_n, s_j} \in \{0, 1\}, \forall s_j \in \mathcal{S}, \forall r_n \in \mathcal{R} \quad (1.21)$$

Constraint (1.17) ensures that the service assigned to the network resources satisfies the overall quality tolerance of the service request. Similarly, constraint (1.18) imposes that the social and ethical value provided by the selected network resources is greater than zero. Constraint (1.19) ensures that the total amount of requested services does not exceed the available capacity of the network resources. Constraint (1.20) imposes that each service is assigned to exactly one network resource, whereas constraint (1.21) enforces the binary nature of the decision variables.

The problem defined in (1.16) is a generalization of the maximum Generalized Assignment Problem (GAP), which is known to be NP-hard [71]. Moreover, attempts to improve service quality and performance may also increase system complexity or enlarge the attack surface, thereby potentially introducing additional security issues. Since the problem involves two conflicting objective functions, a solution that simultaneously optimizes both objectives may not always exist. In the context of multi-objective optimization, the notion of Pareto optimality can therefore be used to evaluate the quality of solutions. A solution X^* is Pareto optimal, or dominant, if there exists no other solution X such that $Q(X) \geq Q(X^*)$ and $V(X) \geq V(X^*)$, where X is distinct from X^* and belongs to the feasible region.

To compute the set of dominant solutions, the exact ε -constraint method can be adopted [72], [73]. Generally, multi-objective optimization problems involve two or more conflicting objective functions that must be optimized jointly. A widely adopted solution strategy is the weighted sum method, in which the original multi-objective formulation is transformed into a single-objective problem by assigning a weight to each objective. However, the effectiveness of this approach strongly depends on the choice of such weights, which is often based on empirical assumptions. As a result, the method may introduce a degree of subjectivity and may also increase the effort required to calibrate the optimization process.

A more structured alternative is indeed provided by the ε -constraint method [74]. In this approach, one objective is selected as the primary objective function, whereas the remaining objectives are converted into constraints bounded by suitable ε -values. In this way, the original multi-objective problem is reformulated into a sequence of single-objective subproblems, commonly referred to as ε -constraint problems. By progressively varying these bounds, the method allows the reconstruction of the set of non-dominated solutions, that is, the Pareto front.

Compared with the weighted sum method, the ε -constraint technique overcomes several of its main limitations. In particular, it is less sensitive to the scaling of the objective functions and provides finer control over the distribution of Pareto-optimal solutions through the adjustment of the ε -bounds. In practical terms, the optimization workflow adopted in this work is implemented through TETREES, while its software architecture and internal components are discussed in detail in the next subsection.

Without loss of generality, the social and ethical KVIs objective is treated as the primary objective in this work. Accordingly, the original multi-objective problem in Eq. (1.16) is reformulated as the following single-objective optimization problem:

$$\max_X V(X), \quad (1.22)$$

$$\text{s.t. } Q(X) \geq \varepsilon, \quad (1.23)$$

$$(1.17), (1.18), (1.19), (1.20), (1.21). \quad (1.24)$$

In constraint (1.23), the parameter ε defines the lower bound imposed on the objective function $Q(X)$. More precisely, ε varies between the lower and upper bounds attained by $Q(X)$ over the Pareto-optimal set, which correspond to the Nadir and ideal points, respectively.

In its traditional form, the ε -constraint method relies on a uniform partition of the admissible ε -interval. According to this scheme, the interval is divided into equally spaced sub-intervals, and each boundary value is used as a candidate ε -level. Although straightforward to implement, this strategy does not guarantee the identification of all non-dominated solutions, since some Pareto-optimal points may be missed because of the fixed discretization.

To overcome this limitation, the exact ε -constraint method can be adopted. This variant adaptively refines the selection of the ε -values and thus supports the generation of the complete Pareto front. The detailed procedure is summarized in Algorithm 1.

Algorithm 1 Exact ε -Constraint Algorithm

Input: $\mathcal{R}, \mathcal{S}$

Output: Pareto-optimal solutions X^*

- 1: Compute the ideal points (V_I, Q_I) with **Algorithm 2**.
 - 2: Compute the Nadir points (V_N, Q_N) , by solving problems 1.25 and 1.28 with **Algorithm 2**.
 - 3: Set $\varepsilon_{\min} = Q_N$, $\varepsilon_{\max} = Q_I$, and step size equal to δ .
 - 4: **for** $\varepsilon = \varepsilon_{\min}$ **to** $\varepsilon_{\max}$ **with step** δ **do**
 - 5: Solve each problem 1.22 with **Algorithm 2**.
 - 6: Return the best-known solution X^* .
 - 7: Store the optimal solution if non-dominated.
 - 8: **end for**
 - 9: Stop when $\varepsilon \geq \varepsilon_{\max}$.
-

The purpose of Algorithm 1 is to identify the set of Pareto-optimal solutions X^* , namely the solutions for which one objective cannot be improved without deteriorating the other. To this end, the algorithm first determines the admissible range of ε , which acts as the bound of the constrained objective. This range is derived by solving two independent single-objective optimization problems.

The first optimization problem considers only $V(X)$ as objective function, without imposing any bound on $Q(X)$. The second one optimizes $Q(X)$ alone, without constraining $V(X)$. These two optimizations yield the ideal points. More specifically, the solution of the first subproblem is denoted as V_I , which represents the ideal point of $V(X)$, whereas the solution of the second subproblem is denoted as Q_I , corresponding to the ideal point of $Q(X)$. Together, these values identify the best achievable outcome for each objective when optimized independently.

Subsequently, two further optimization subproblems are solved to determine the Nadir values of the two objective functions, denoted by V_N and Q_N , respectively. These values are obtained by optimizing one objective while constraining the other to its ideal value. More specifically, V_N is computed by maximizing $V(X)$ under the constraint $Q(X) = Q_I$, whereas Q_N is obtained by maximizing $Q(X)$ under the constraint $V(X) = V_I$, as shown below:

$$\max_X V(X), \quad (1.25)$$

$$\text{s.t. } Q(X) = Q_I, \quad (1.26)$$

$$(1.17), (1.18), (1.19), (1.20), (1.21). \quad (1.27)$$

$$\max_X Q(X), \quad (1.28)$$

$$\text{s.t. } V(X) = V_I, \quad (1.29)$$

$$(1.17), (1.18), (1.19), (1.20), (1.21). \quad (1.30)$$

Once these auxiliary optimization problems have been solved, the lower bound of ε is set equal to the Nadir value of the constrained objective, whereas the upper bound is set equal to its ideal value. In order to explore the Pareto front systematically, a step size δ is introduced to discretize the admissible range of ε . The choice of δ must balance computational efficiency and solution accuracy, so as to capture the non-dominated solutions while limiting the computational burden.

The algorithm then proceeds iteratively by increasing ε from its lower bound to its upper bound. At each iteration, a new optimization problem is formulated and solved by keeping one objective as the primary function and by constraining the other through the current value of ε . The resulting solution is then checked for dominance. If it is non-dominated, it is added to the Pareto-optimal set. The iterative process continues until ε exceeds its upper bound, which indicates that the whole Pareto front has been explored. At termination, the algorithm returns the set of Pareto-optimal solutions representing the trade-offs between the conflicting objectives.

Algorithm 2 Branch and Cut Algorithm

Input: Problem 1.22
Output: Optimal solution.

- 1: Solve the relaxed version of the problem, without integrality constraints.
- 2: Set the initial solution space as the relaxed problem's feasible region.
- 3: **while** solution space is not empty **do**
- 4: Partition the solution space into subproblems by branching.
- 5: For each subproblem, compute a lower bound of the objective function.
- 6: **if** a subproblem has a lower bound greater than or equal to the best-known solution **then**
- 7: Prune the subproblem.
- 8: **end if**
- 9: Apply cutting planes to tighten the feasible region.
- 10: Solve each remaining subproblem and update the best-known solution.
- 11: **end while**
- 12: Return the best-known solution.

Each of the optimization problems introduced above can be solved iteratively through algorithmic schemes such as Branch and Bound, which can be further strengthened by integrating pruning mechanisms and cutting planes [75]. In general, Branch and Bound is a combinatorial optimization technique that explores the solution space through a tree structure, where each node corresponds to a subproblem derived from the original formulation.

The procedure starts by solving a relaxed version of the problem, typically obtained by removing the integrality constraints, so as to derive an upper bound in the case of maximization problems. If the relaxed solution is already feasible, then it is optimal. Otherwise, the problem is partitioned into smaller subproblems by introducing additional constraints, for example by fixing a variable to 0 or 1. In order to limit the computational burden, pruning is used to discard those subproblems whose bounds indicate that they cannot lead to an optimal solution [76].

This procedure can be further improved by incorporating cutting planes, namely additional constraints that tighten the feasible region without excluding optimal integer solutions. The resulting enhanced strategy is known as Branch and Cut [77]. More specifically, when the current solution is fractional, additional cuts are introduced to remove that fractional point while preserving all feasible integer solutions. If the solution remains fractional after the addition of cuts, the standard branching process of Branch and Bound is applied.

This search strategy reduces the number of explored branches and therefore improves computational efficiency. Its implementation is summarized in Algorithm 2. In the present work, these optimization procedures are implemented within TETREES, which acts as the software support for the exact exploration of the Pareto front and is described in detail in the next subsection.

In general, the worst-case time complexity of the pure Branch and Bound algorithm is exponential. However, by exploiting integrality relaxations, pruning, and cutting planes, the average-case complexity may be significantly reduced and is often expressed as $O(n^k)$, where k is a small positive integer that depends on the

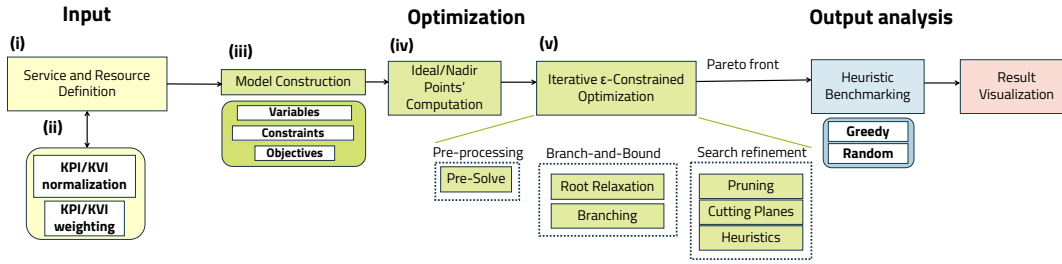


Figure 1.6: Overview of the TETREES framework architecture [61].

specific problem instance and on the effectiveness of the adopted pruning and cutting strategies. For commercial solvers such as Gurobi, k may in practice take small values, for example 1 or 2.

In the case of Algorithm 1, the time complexity is given by $O(4 \cdot (J + N)^k + (Q_I - (Q_N - \delta)) \cdot (J + N)^k)$, which simplifies to $O((Q_I - (Q_N - \delta)) \cdot (J + N)^k)$. Here, $O((J + N)^k)$ represents the time complexity associated with computing the ideal and Nadir points, as well as solving each subproblem for a fixed ε . The term $(Q_I - (Q_N - \delta))$ corresponds to the number of iterations of the inner for-loop.

The selected parameter ranges reflect plausible operating conditions for distributed orchestration scenarios, in which each orchestrator manages service requests over a limited subset of network resources, typically associated with a cluster or with a bounded geographical area. Preliminary scale-up experiments on instances one order of magnitude larger suggested solver convergence within acceptable tolerances. Due to space limitations and to the methodological focus of the paper, a comprehensive large-scale benchmarking campaign is left for future work.

1.2.2.4 Optimization Support Through TETREES

Within the optimization perspective introduced above, this thesis relies on TETREES [61], as the software framework supporting the exact exploration of the feasible trade-offs between conflicting objectives. Rather than acting as a simple implementation layer, TETREES provides a reproducible computational workflow for the bi-objective orchestration problem considered in this chapter, thereby enabling the systematic reconstruction of the Pareto front associated with the adopted KPI/KVI model.

As illustrated in Fig. 1.6, the framework is organized into five main components, namely i) service and resource definition, ii) KPI/KVI normalization and weighting, iii) model construction and objective-specific optimization, iv) ideal and Nadir point computation, and v) iterative ε -constrained optimization. This decomposition is consistent with the logic adopted throughout the chapter, where high-level service requirements are progressively translated into structured variables, constraints, and optimization criteria.

The underlying optimization problem is formulated as a Binary Integer Linear Programming (BILP) in which each service request must be assigned to exactly one candidate resource. Within this setting, TETREES jointly handles heterogeneous KPIs, such as throughput, latency, and packet loss, together with broader KVIs, such

as sustainability, trustworthiness, and inclusiveness. As a consequence, the framework supports the exact exploration of orchestration decisions in which technical efficiency and broader societal or ethical dimensions are treated within the same decision process, rather than being separated into distinct optimization stages.

A relevant feature of the framework is the explicit representation of both service-side and resource-side information. More specifically, services are characterized by demand profiles, quality thresholds, and target KPI/KVI values, whereas resources are described through attributes such as capacity, reliability, energy efficiency, and computational performance. Starting from these inputs, TETREES constructs the optimization model by combining assignment constraints, service-level requirements, resource-capacity limitations, and ε -constraints enforcing Pareto-efficient trade-offs. The admissible range of ε is derived from the ideal and Nadir values of the constrained objective and is then explored through iterative exact optimization. In practical terms, this means that the framework first computes the extremal values needed to initialize the ε -constraint method and then repeatedly solves the resulting constrained subproblems so as to reconstruct the corresponding set of non-dominated solutions. This design makes the overall procedure reproducible and relieves the analyst from manually orchestrating the successive optimization runs required for Pareto-front construction.

The exact optimization stage is delegated to Gurobi as Mixed-Integer Programming (MIP) backend. In practice, the solver addresses integrality and linearity constraints through a branch-and-bound process enriched by presolve reductions, pruning, cutting planes, and heuristic routines. The solution process starts from the relaxed formulation at the root node, branches on fractional variables, prunes subproblems that cannot improve the incumbent solution, and injects valid inequalities to tighten the feasible region. During presolve, redundant constraints may be removed and structurally similar model components may be aggregated, while additional heuristics may accelerate the discovery of high-quality feasible solutions.

A further important aspect concerns solver configurability. In particular, *MIP-Gap* specifies the acceptable optimality gap between the incumbent solution and the best available bound, thus allowing the user to control the trade-off between convergence precision and runtime. *MIPFocus* biases the internal search toward different priorities, such as finding feasible solutions more rapidly, strengthening bounds, or proving optimality more aggressively. *VarBranch* defines the branching strategy adopted during the search, for example by favoring pseudo-reduced-cost decisions or stronger branching policies. Finally, *Heuristics* regulates the amount of effort allocated to heuristic procedures, thereby affecting how aggressively the solver searches for good feasible solutions before full proof of optimality is achieved.

Taken together, these parameters allow the analyst to study how the internal search strategy influences runtime, convergence behavior, and Pareto-front granularity even when the optimization model itself remains unchanged. This is particularly relevant in orchestration scenarios, where the quality of the final Pareto set must often be balanced against operational timing constraints and available computational budget.

Another relevant aspect of TETREES concerns the explicit separation between optimization and post-processing. Once the sequence of ε -constrained subproblems

has been solved, the framework filters dominated solutions and exports the resulting allocations in structured CSV form. In addition to Pareto-optimal allocations, the exported outputs may include ε -constrained solutions and heuristic benchmarks, which can then be further inspected through dedicated post-processing scripts and visualization tools. Alongside exact optimization, TETREES also supports heuristic benchmarking through greedy and randomized allocation strategies under the same capacity constraints. These baselines provide a useful complement to the exact procedure, since they make it possible to compare Pareto-optimal solutions against faster approximate strategies and therefore to better interpret the trade-off between solution quality and computational effort. Within the broader architecture proposed in this thesis, TETREES therefore acts as the optimization-oriented counterpart of the secure and value-aware intent translation pipeline discussed in the previous sections. After high-level requests have been validated and translated into structured service requirements, the framework enables the exact exploration of the feasible non-dominated solutions satisfying both operational constraints and value-oriented objectives.

By making the exploration of multi-objective orchestration decisions transparent, reproducible, and extensible, the framework supports the systematic inspection of how different allocations balance network quality and social-ethical value, while also clarifying how solver configurations affect the density and granularity of the resulting Pareto front.

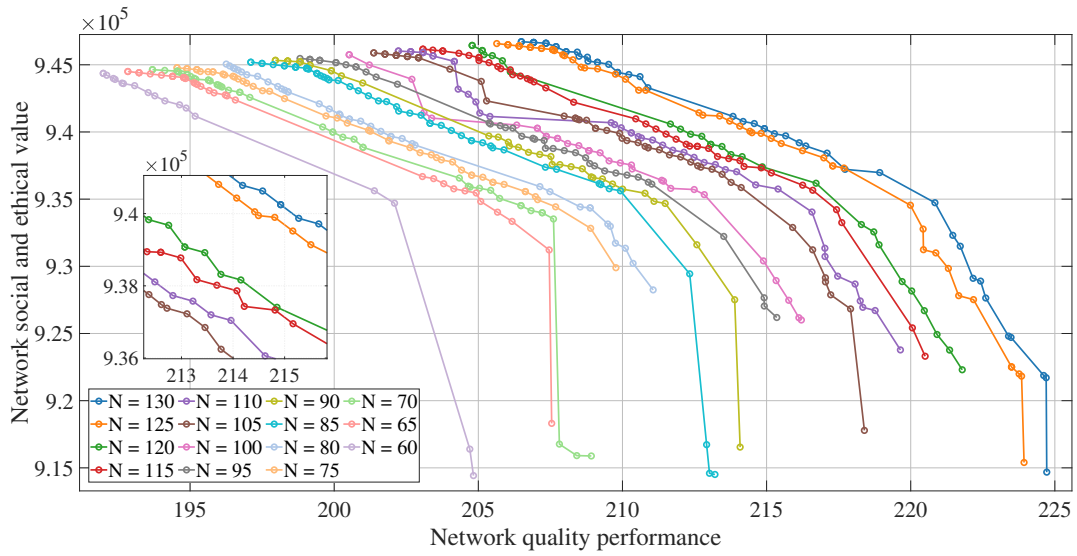
The applicability of the framework is demonstrated in a network orchestration scenario; in this setting, a network orchestrator dynamically assigns incoming service requests to the available resources of one or more providers, selecting the allocation that best satisfies SLAs while balancing potentially conflicting KPI and KVI objectives. For the KPI objective, we consider the metric defined in Eq. 1.15, computed as a weighted and normalized aggregation of the data rate, latency, and packet loss associated with serving each request. For the KVI objective, we adopt the social and ethical value introduced in Eq. 1.14, which combines trustworthiness, inclusiveness, and environmental sustainability into a single composite measure. The examples rely on the parameters defined in the *main.py* script, namely *num_resources* (number of providers), *num_services_list* (number of service requests), and the corresponding service and resource characteristics. In all experiments, the number of resources varies from 60 to 130, while the number of service requests is fixed at 300.

Figure 1.7 reports the resulting Pareto fronts. Each curve corresponds to the set of non-dominated allocations obtained for a particular number of providers, and each point represents a feasible service-resource assignment that achieves an optimal trade-off between performance and social/ethical value. As the number of providers increases, the solution space expands and becomes denser, generating progressively richer and more diverse sets of non-dominated solutions.

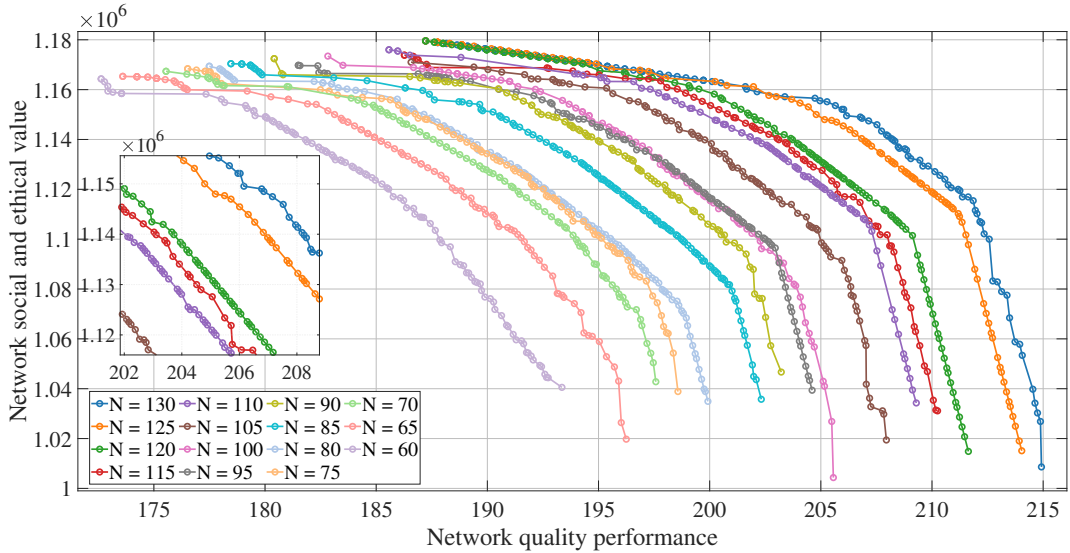
In the first scenario (Fig. 1.7a), service requests prioritize stringent security, privacy, and resilience requirements. The extreme points of each frontier correspond to the maximum attainable value of each individual objective, obtained by optimizing solely the social/ethical value or solely the performance metric. Intermediate points represent trade-offs between them, with the central region of each curve identifying the most balanced compromise solutions. The zoomed-in region highlights that even small increases in resource availability yield measurable improvements in the

attainable trade-offs, enabling stricter SLA compliance and more efficient service allocation.

In the second scenario (Fig. 1.7b), service requests emphasize reliability, physical safety, and energy efficiency. The resulting fronts are more uniformly populated and exhibit smoother progressions compared to the previous case, indicating that this distribution of service requirements induces a more regular and homogeneous solution space. These observations are particularly relevant for infrastructure planning, as they help operators assess the resource availability needed to concurrently meet demanding performance constraints and stringent value-driven objectives.



(a) Security-, privacy-, and resilience-oriented services.



(b) Reliability-, safety-, and energy-efficiency-oriented services.

Figure 1.7: Pareto fronts for increasing numbers of service providers (N) considering 300 service requests [61].

Beyond adjusting the number of providers, TETREES also supports the reconfiguration of the solver's internal search strategy within the same operational context.

By tuning parameters that control heuristic intensity, branching depth, and global search priorities, users can obtain Pareto fronts with comparable densities but significantly different computational costs. To illustrate this, Table 1.4 compares five solver configurations for 110 providers and 300 service requests: Adaptive, Cut-and-Solve, Strong Branching, Sub-Optimal, and Infeasibility-Driven. These configurations modify the Gurobi parameters (e.g., *VarBranch*, *MIPFocus*, and *Heuristics*).

While all configurations eventually produce high-quality Pareto sets, the density and shape of the frontiers differ. The Adaptive and Cut-and-Solve strategies achieve similar runtimes (221–237 s) and produce comparable numbers of Pareto points, though the latter relies on a significantly larger heuristic budget (80%). Strong Branching, despite its computationally intensive branching rule, yields fewer Pareto points (36), confirming that aggressive bound tightening reduces search breadth. The Infeasibility-Driven configuration generates the largest number of non-dominated points (94), but at a runtime more than twice that of the other strategies, illustrating the trade-off between frontier granularity and computational effort. These results show that solver-parameter choices affect not only runtime but also the structure, density, and granularity of the resulting Pareto front. Selecting an appropriate configuration therefore becomes an application-aware design decision, particularly important when timely computation is required for next-generation network services.

Table 1.4: Simulation results considering 300 service requests and 110 service providers under different solver configurations.

Parameter	Adaptive	Cut-and-Solve	Strong-Branching	Sub-Optimal	Infeasibility-Driven
Branch Strategy	Pseudo-Reduced Cost	Pseudo-Reduced Cost	Strong	Pseudo-Shadow Price	Max. Infeasibility
Global Search	Balanced	Feasible Solutions	Best Bound	Best and Fast Bound	Balanced
Heuristics (%)	10	80	0	5	10
Run time (s)	221.98	237.05	248.18	280.38	624.79
Pareto Points (#)	58	59	36	36	94

1.2.2.5 Performance Evaluation

This section outlines the considered experimental setup and discusses the outcomes of the simulation campaigns performed through TETREES. The goal of the evaluation is to analyze the effectiveness of the proposed service orchestration model in jointly optimizing KPIs and KVIs, thereby broadening the assessment beyond traditional performance-related metrics.

The examined environment includes N network resources, with N ranging from [50, 800]. Each resource is characterized by a number of available cores within [10, 50] and by processing capabilities spanning [40, 150] megacycles per second. The MTTF is assumed to lie in the interval [8760, 45000] hours, consistently with [78], whereas the average number of services that each resource can serve per day varies within [150, 250].

The power consumption, expressed in kilowatts, is assumed to fall in the range $[0.01, 0.2]$. Moreover, the carbon credit offsets vary between $[4109, 6849.31]$ gCO₂e per day for each resource. The CI exhibits large differences across countries, ranging from 12 gCO₂e/kWh in regions mainly powered by renewable energy sources, such as Switzerland or Norway, to values exceeding 800 gCO₂e/kWh in countries with strong dependence on coal or gas, such as Australia or South Africa. Although marginal CI would provide a more precise estimate of the environmental impact of computation migration, it is frequently unavailable. Therefore, the average CI is commonly adopted as a practical lower bound and, in the following analysis, it is fixed to the global average value of 475 gCO₂e/kWh.

The probability of a successful cyber attack, denoted by L_{r_n} , is assumed to vary in $[0.25, 1]$. Finally, the lower and upper bounds of the trustworthiness indicator are set to $L_\theta = 1500$ and $U_\theta = 5000$, respectively, representing the minimum and maximum trust levels that a resource can realistically reach. The inflection point x_0 and the growth parameter B are fixed to 0.5 and 0.6, respectively, in accordance with a normalized confidence domain, so that the trust level increases more rapidly around the midpoint of the security assurance scale. The considered parameters are summarized in Table 1.5.

Table 1.5: Simulation settings.

Parameter	Setting
J	$[80, 1200]$
N	$[50, 800]$
C_{r_n}	$[10, 50]$
D_{s_j}	$[2, 5]$
cor_{r_n}	$[4109, 6849.31]$ gCO ₂ e per day
CI	$[475]$ gCO ₂ e/kWh
Clk_{r_n}	$[40, 150]$ Megacycles/s
$P_c^{r_n}, P_m^{r_n}$	$[0.01, 0.2]$ kW
$MTTF_{r_n}$	$[8760, 45000]$ hours
λ_{r_n, s_j}	$[150, 250]$ per day
B_{s_j}	$[600, 1200]$ Mb
Δ_{s_j}	$[0.25, 1]$
L_{r_n}	$[0.25, 1]$
U_θ	5000
L_θ	1500
B	0.6
x_0	0.5

The environment also comprises an unspecified number of consumers generating service requests, which are then translated into intents. The number of service requests under consideration ranges from $[80, 1200]$.

To define the considered intent categories, four classes of 6G services are taken into account, namely *Immersive Experience*, *Collaborative Robots*, *Physical Awareness*, and *Trusted Environments*, as described in [79]:

1. **Immersive Experience** includes multimedia and extended reality services, such as immersive telepresence, education, and gaming, and requires ultra-low latency, high data rates, and accurate positioning. In this case, the Sustainability of Service (SOS) mainly concerns digital divide mitigation and the reduction of energy consumption through more efficient computing and transmission.
2. **Collaborative Robots** refer to autonomous robotic systems operating in industrial and public contexts, which require ultra-reliable low-latency communications while giving relevance to sustainability through the reduction of waste and environmental impact.
3. **Physical Awareness** services rely on sensing technologies for applications such as network-assisted mobility and smart environments. They require high-resolution sensing, accurate positioning, and high reliability, while also supporting environmental monitoring, energy-efficient connectivity, and improved safety in smart cities.
4. **Trusted Environments** concern secure and privacy-oriented applications, including telemedicine and public safety, and therefore require strong security guarantees and very high reliability, together with resilient infrastructures capable of supporting societal well-being.

To instantiate the system model, the relevant parameters are derived from 6G service requirements and from the features of the considered service classes. The service delay, expressed in seconds, ranges within $[0.002, 50]$. The data rate, expressed in bits per second, varies within $[70, 250]$, while the packet loss rate, expressed as the number of packets lost per second, lies in $[20, 50]$. The service demand ranges from $[2, 5]$, whereas the input data size B_{s_j} is sampled in $[600, 1200]$ Mb for each service. In addition, the impact Δ_{s_j} of an attack on service provisioning varies in $[0.25, 1]$. All these intervals are selected according to the deployment type and the characteristics of the involved entities, as discussed in [79].

The performance of the proposed solution is compared against three baseline approaches, each adopting a different orchestration policy:

- **Random Matching (RM)**: service requests are randomly assigned to network resources by selecting the first available provider, without taking into account either the KPIs or the KVI offered by the resource.
- **Performance-oriented Greedy Matching (PGM)**: service requests are allocated through a greedy strategy that iteratively selects the most favorable available option by considering only the provided KPIs.
- **Value-oriented Greedy Matching (VGM)**: service requests are allocated through a value-oriented greedy strategy that iteratively selects the most favorable available option by considering only the provided KVI.

Figure 1.8 shows the Pareto fronts obtained by solving the proposed optimization problem under different combinations of J service requests and N available network resources.

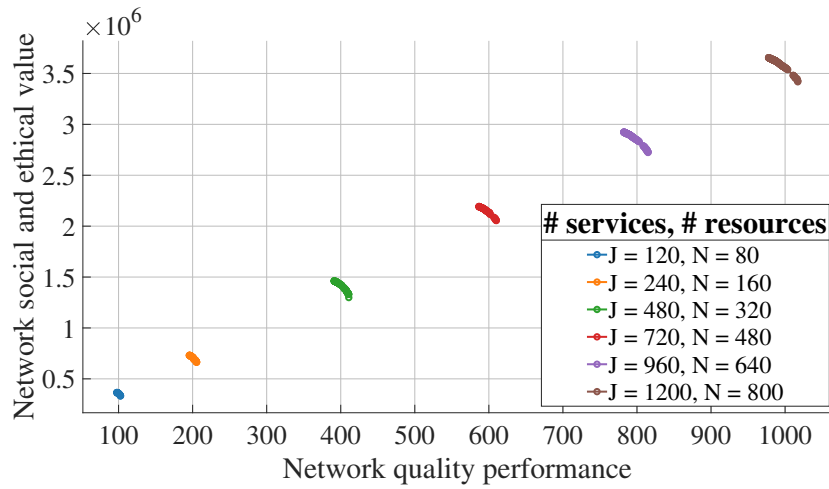


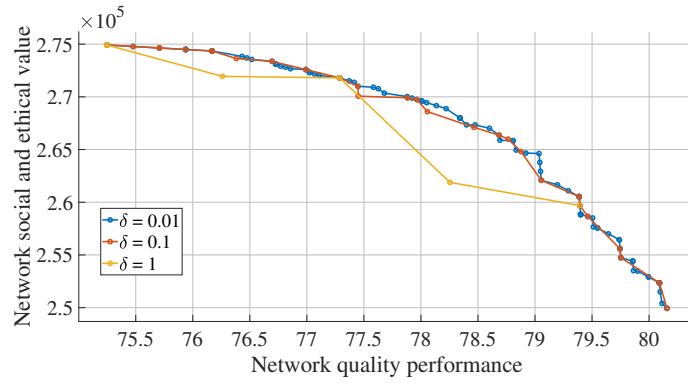
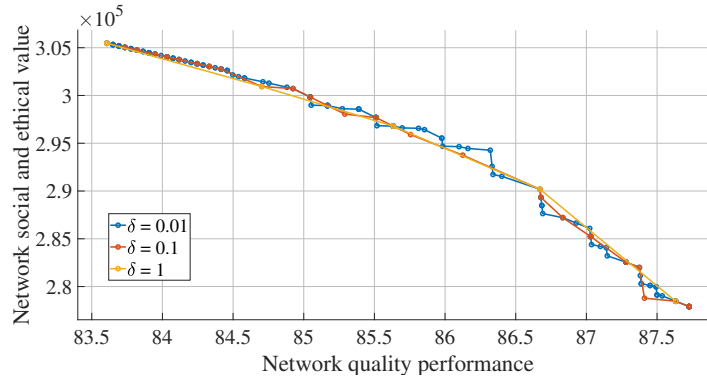
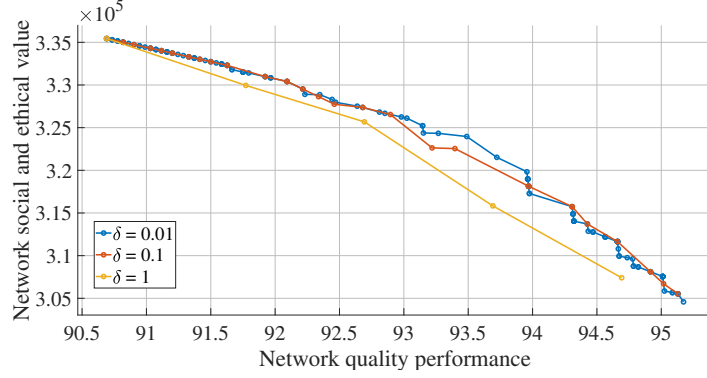
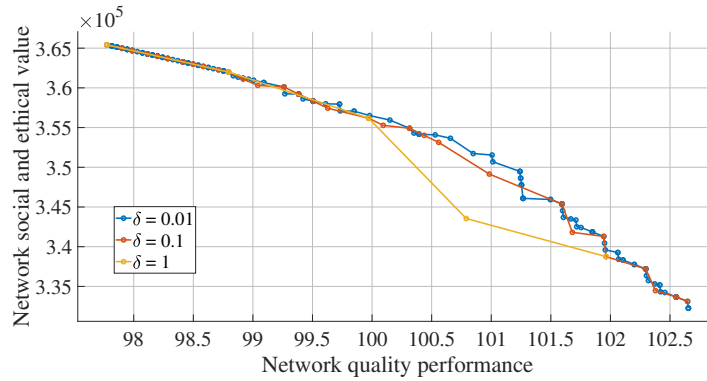
Figure 1.8: Pareto fronts considering different service requests (J) and network resources (N).

In this analysis, the discretization parameter δ is fixed to 0.1 in order to preserve consistency while maintaining a reasonable computational cost during Pareto-front construction. As discussed in Section 2.3.4, this choice makes it possible to identify a sufficiently rich set of Pareto-optimal solutions without the excessive computational burden associated with finer discretization values.

In the considered setting, neither quality performance nor social and ethical value is explicitly prioritized; accordingly, KPIs and KVIs are assigned the same weight. Similarly, the three KVIs are assumed to be equally relevant. Further discussion on the behavior of the proposed approach under different weight configurations and use cases is provided in the following subsections.

Table 1.6: Simulation run times with varying δ .

N	J	δ	Run time (s)
80	90	1	3.27
80	100	1	3.04
80	110	1	4.08
80	120	1	3.45
80	90	0.1	16.59
80	100	0.1	16.75
80	110	0.1	20.89
80	120	0.1	20.48
80	90	0.01	153.36
80	100	0.01	237.82
80	110	0.01	260.43
80	120	0.01	214.93

(a) $N = 80, J = 90$ (b) $N = 80, J = 100$ (c) $N = 80, J = 110$ (d) $N = 80, J = 120$ Figure 1.9: Pareto fronts with varying service requests J and δ [60].

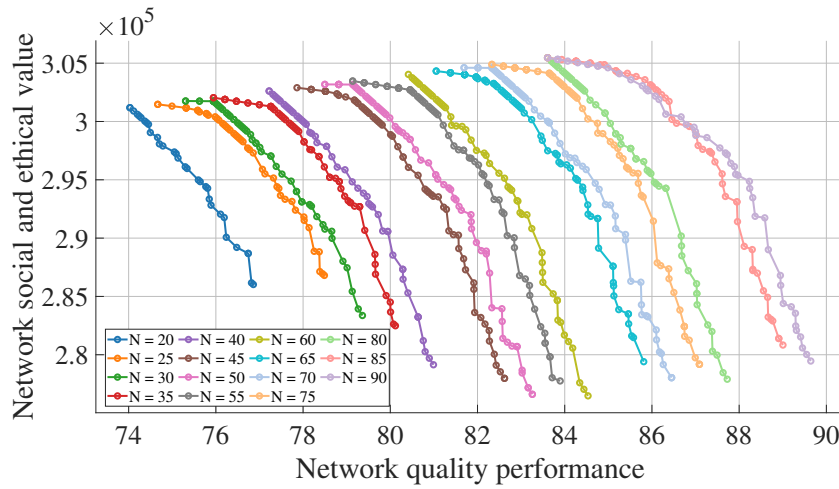


Figure 1.10: Pareto fronts obtained for varying amounts of network resources, with the number of service requests fixed at $J = 100$ [60].

Table 1.7: Simulation run times with varying weights of KVI.

N	J	Pref. KVI	Run time (s)
80	90	σ_{r_n, s_j}	27.57
80	90	θ_{r_n, s_j}	16.00
80	90	ι_{r_n, s_j}	53.98
80	100	σ_{r_n, s_j}	29.36
80	100	θ_{r_n, s_j}	16.64
80	100	ι_{r_n, s_j}	56.25
80	110	σ_{r_n, s_j}	27.63
80	110	θ_{r_n, s_j}	21.24
80	110	ι_{r_n, s_j}	55.18
80	120	σ_{r_n, s_j}	37.36
80	120	θ_{r_n, s_j}	30.43
80	120	ι_{r_n, s_j}	32.13

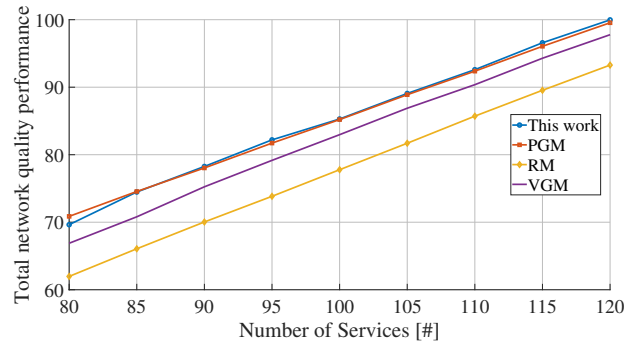
The resulting curves clearly emphasize the trade-off between network quality performance and social and ethical value, showing how the number of available resources and requested services affects the set of optimal solutions. In particular, when both the number of network resources and the number of service requests increase, the Pareto front expands, enabling better trade-offs between the two conflicting objectives. For instance, in small-scale scenarios such as $N = 80$ and $J = 120$, the optimal solutions are confined to the range $(0.4 \cdot 10^6, 105)$. Conversely, in larger-scale scenarios such as $N = 800$ and $J = 1200$, the Pareto front becomes denser, thus reflecting greater flexibility in the decision space. Under these conditions, the network attains a maximum social and ethical value of $3.6 \cdot 10^6$ while reaching a maximum network quality performance equal to 1100. More generally, the results confirm the inherent conflict between the two objectives, since improving one dimension typically causes a reduction in the other.

In addition, it is possible to examine the execution times obtained with the Gurobi commercial solver, focusing in particular on how Pareto-front construction varies under different combinations of services and network resources. The analysis considers variations in the parameters affecting both the ε -constraint method and the system model, including the discretization step δ and the weights associated with the three KVIs, namely environmental sustainability, trustworthiness, and inclusiveness.

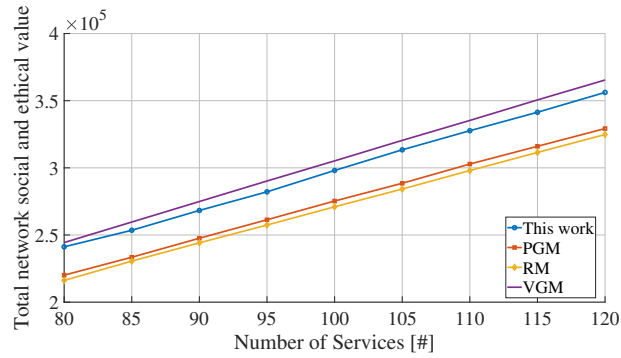
Figure 1.9 analyzes the impact of varying the number of services J and the discretization parameter δ , while keeping the number of network resources fixed at $N = 80$. From the perspective of solution granularity, a finer discretization, namely $\delta = 0.01$, increases the number of optimal trade-offs from 7 to 164 with respect to the coarser choice $\delta = 1$, thus yielding a more detailed representation of the solution space when $J = 120$.

However, the higher granularity entails a significant computational cost, as shown in Table 1.6. More specifically, the execution time increases from 3.45 s for $\delta = 1$ and $J = 120$, to 20.48 s for $\delta = 0.1$ and $J = 120$, and then to 214.93 s for $\delta = 0.01$ and $J = 120$. The most critical case is observed for $\delta = 0.01$ and $J = 110$, where the execution time reaches 260.43 s, further confirming the steep growth in computational complexity induced by finer discretization. By contrast, a coarser discretization such as $\delta = 1$ considerably reduces the runtime, producing values between 3.04 s and 4.08 s. This reduction, however, comes at the cost of a lower number of optimal solutions and may therefore miss relevant trade-offs between quality performance and social and ethical value. On the other hand, smaller values of δ improve the optimal social and ethical network value by 12%, thus enabling a more accurate optimization of environmental sustainability, trustworthiness, and inclusiveness.

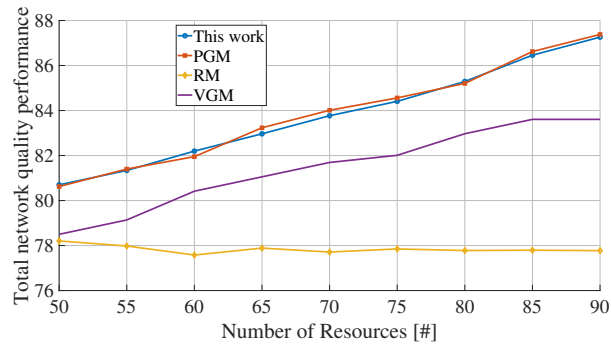
Table 1.7 reports the execution times obtained under different assignments of the KVI weights. This analysis is useful to evaluate the computational impact of prioritizing specific social and ethical dimensions within the optimization process.



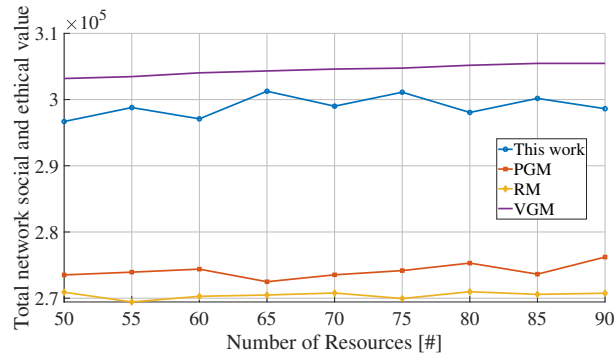
(a) Network quality performance with fixed amount of resources
 $N = 80$



(b) Network social and ethical value with fixed amount of resources
 $N = 80$



(c) Network quality performance with fixed service requests
 $J = 100$



(d) Network social and ethical value with fixed service requests
 $J = 100$

Figure 1.11: Network quality performance and social and ethical value considering neither quality nor social and ethical prioritization [60].

Overall, the results show that the execution times remain largely consistent across the different KVI weight configurations, with only limited variations due to the specific application scenarios. For example, when $J = 100$ and $N = 80$, runtimes range from 16 s to 57 s, with the highest value observed when inclusiveness is prioritized. For a slightly higher service load, such as $J = 120$, runtimes range from 30.43 s to 37.36 s, with the maximum value obtained when environmental sustainability is prioritized. Therefore, execution time is mainly driven by the service load and by the number of available resources, whereas the relative weights assigned to the KVIs influence the optimization result without significantly altering the computational complexity.

Figures 1.10 and 1.11 further analyze the trade-off between network quality performance and social and ethical value under different service–resource combinations, while keeping the parameter δ fixed and considering a scenario in which neither quality performance nor social and ethical value is explicitly prioritized. The curves compare the proposed optimization-based orchestration framework against the baseline approaches PGM and RM. The results are obtained over 500 different seeds in order to ensure robustness with respect to different distributions of service request types.

More specifically, Figure 1.10 provides an empirical validation of the theoretical framework introduced in Section 4, showing the capability of the system to identify and explore Pareto-optimal solutions as the availability of resources increases. With a fixed number of service requests, namely $J = 100$, and an increasing number of available resources, ranging from 20 to 90, the Pareto front progressively shifts, reflecting improvements in both service quality and social and ethical value. This confirms that greater resource diversity facilitates the identification of a broader set of non-dominated configurations and enables more favorable trade-offs between conflicting objectives. Such a behavior is fully aligned with the bi-objective optimization perspective adopted in this work, which jointly considers service demands and resource capabilities in order to promote the use of ethically valuable resources while still satisfying user requirements.

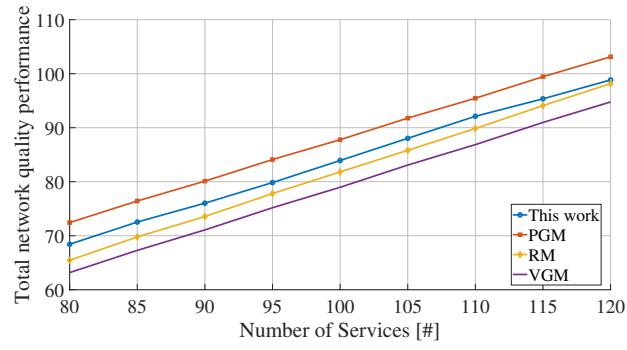
In addition, Figures 1.11a and 1.11b compare network quality performance and social and ethical value in scenarios where the number of network resources is fixed to $N = 80$, while the number of service requests J varies in $[80, 120]$. In terms of network quality performance, the proposed approach exhibits a behavior consistent with that of PGM, preserving the same overall trend as the number of services increases. At the same time, a 10% improvement is observed in the total social and ethical value, demonstrating the effectiveness of the proposed framework in incorporating social and ethical considerations into service orchestration. Compared with RM, the proposed strategy improves by 13% and 10% in the two evaluations, respectively. Moreover, it exhibits a performance gap of 6% with respect to VGM in terms of network quality, while showing a reduction of up to 5% in the delivered social and ethical value. Overall, these results indicate that the proposed strategy outperforms the baseline approaches by allocating resources more efficiently and by balancing performance with broader value-oriented objectives.

By contrast, the greedy baselines achieve the best outcomes only with respect to the specific criterion they explicitly optimize, while remaining suboptimal with

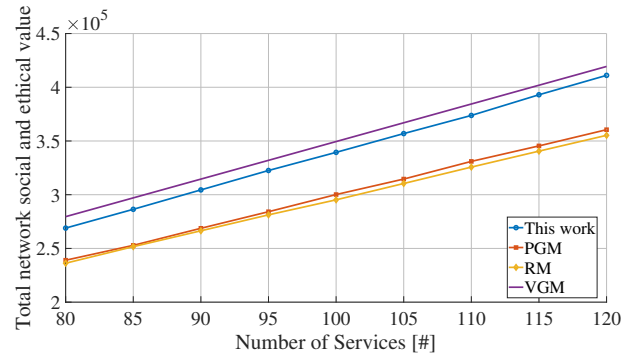
respect to the opposite objective. In particular, PGM follows a purely performance-driven allocation strategy and therefore achieves higher network quality than random allocation, but it does not take into account environmental sustainability, inclusiveness, or other social and ethical aspects. Conversely, VGM attains the highest social and ethical values, but exhibits significantly worse behavior when network quality is considered as the main evaluation criterion. Finally, RM remains the weakest strategy because of its uninformed allocation process, which results in inefficient service–resource matching and lower overall effectiveness.

Figures 1.11c and 1.11d further validate the impact of the proposed optimization strategy on both quality performance and social and ethical value. In this case, the number of services is fixed at $J = 100$, while the number of network resources varies within $[50, 90]$. Consistently with the previous analysis, the proposed approach maintains a quality performance comparable to PGM across all the considered resource settings. At the same time, it achieves improvements of up to 13% over RM and up to 6% over VGM. When only social and ethical value is considered, the proposed framework outperforms both RM and PGM, achieving gains of up to 20% with respect to both baselines, while incurring a performance decrease of at most 1% compared with VGM. These results further confirm that informed strategies are able to exploit a larger provider pool more effectively, whereas RM fails to do so because of its uninformed nature.

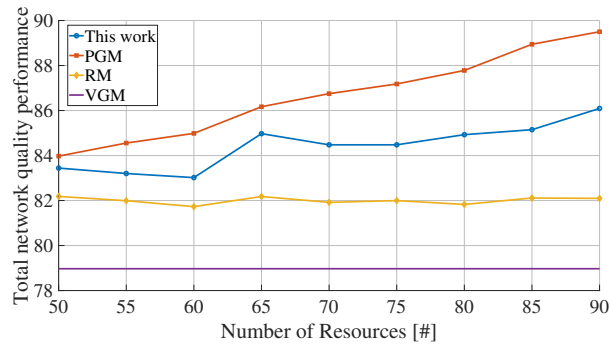
Figures 1.12, 1.13, and 1.14 then move from the general trade-off analysis to specific scenarios, each characterized by realistic KVI and KPI prioritizations that reflect the needs of different 6G service classes. In particular, Figure 1.12 refers to the *Collaborative Robots* and *Physical Awareness* scenarios, where environmental sustainability is prioritized because of the importance of minimizing waste and energy consumption, together with reliability and thus low packet loss. Figure 1.13 instead refers to the *Trusted Environments* scenario, which places stronger emphasis on trustworthiness because of the relevance of security, resilience, and privacy, together with high data rates and minimal packet loss. Finally, Figure 1.14 refers to the *Immersive Experience* scenario, where inclusiveness is assigned greater importance together with low computation delay and high data rate, in order to support equitable access and reduce the digital divide.



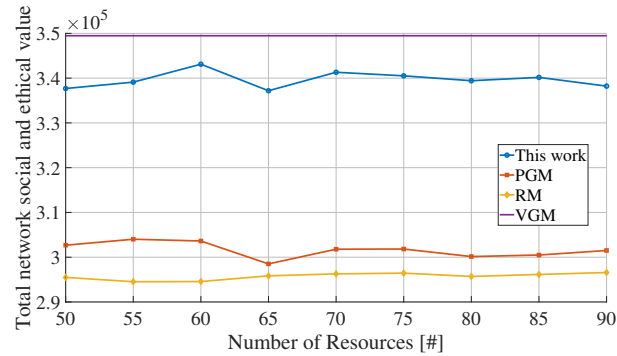
(a) Network quality performance with fixed amount of resources $N = 80$



(b) Network social and ethical value with fixed amount of resources $N = 80$

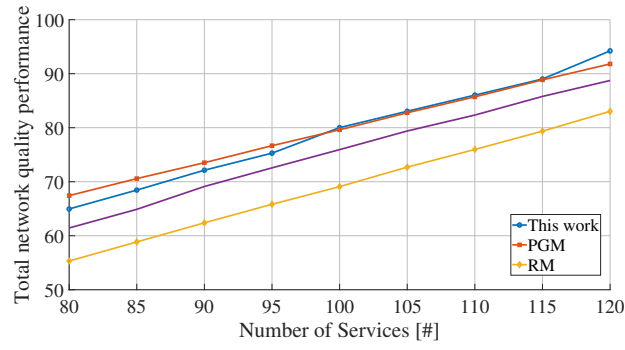


(c) Network quality performance with fixed service requests $J = 100$

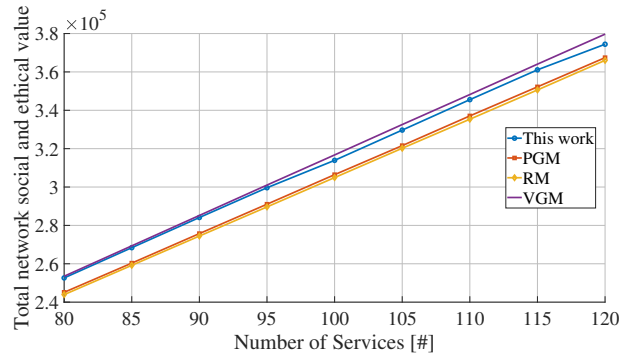


(d) Network social and ethical value with fixed service requests $J = 100$

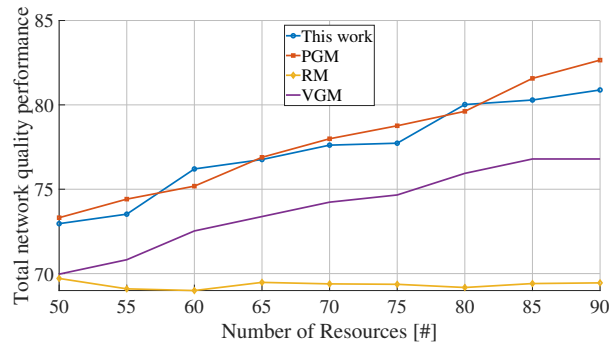
Figure 1.12: Network quality performance and social and ethical value considering an environmental sustainability focus [60].



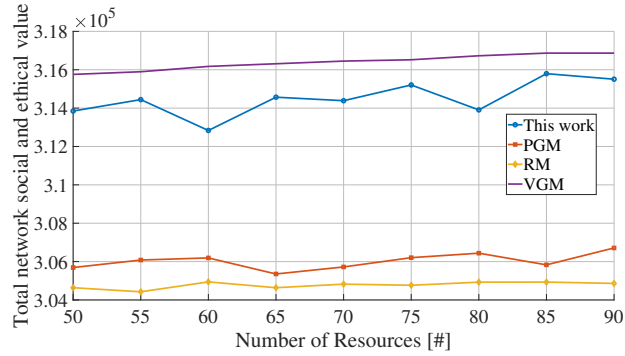
(a) Network quality performance with fixed amount of resources
 $N = 80$



(b) Network social and ethical value with fixed amount of resources
 $N = 80$



(c) Network quality performance with fixed service requests
 $J = 100$



(d) Network social and ethical value with fixed service requests
 $J = 100$

Figure 1.13: Network quality performance and social and ethical value considering a trustworthiness focus [60].

In the first scenario, the results show that the proposed method achieves a network quality performance up to 4% lower than PGM, but up to 9% higher than VGM across different numbers of service requests and network resources. At the same time, it substantially improves the social and ethical value, outperforming PGM by up to 13% and RM by up to 15% under the same configurations. It is worth noting that the achieved social and ethical value remains about 5% lower than that of a purely value-oriented strategy. This indicates that resources providing consistently high KPIs or KVIs at the individual level do not necessarily lead to a globally optimal allocation when both classes of indicators are jointly maximized.

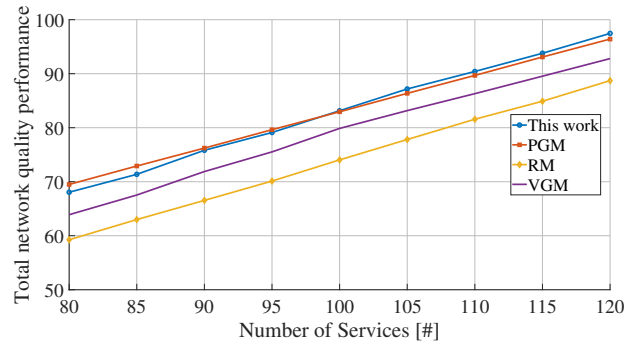
When trustworthiness is prioritized, the proposed orchestration framework maintains a network quality performance closely aligned with PGM, while outperforming VGM by up to 5% and RM by up to 18%. In addition, it surpasses both the performance-oriented and the random baselines in terms of social and ethical value. More specifically, for $J = 100$, it improves over both RM and PGM by about 4%, while consistently matching the performance of VGM. This confirms its capability to incorporate trustworthiness into resource allocation without compromising overall network performance, unlike the purely value-oriented baseline.

Finally, when inclusiveness is prioritized, the proposed method exhibits only a limited degradation in network quality performance across the considered ranges of service requests and network resources, while achieving a substantial increase in social and ethical value. In particular, for $J = 100$, the gain exceeds 70% with respect to both PGM and RM, thus highlighting the effectiveness of the framework in integrating social considerations into network management. In this case, the proposed approach shows a network quality performance gap of up to 12% relative to VGM. Nevertheless, both approaches remain closely aligned in terms of social and ethical value under the same operating conditions, with differences generally remaining within 10%.

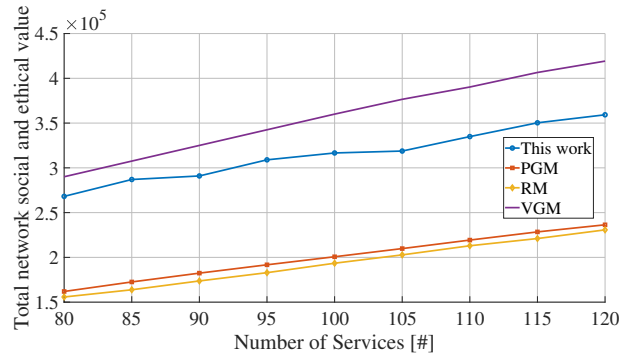
Overall, the proposed framework achieves substantial improvements in the social and ethical value delivered by the network during service orchestration across all the evaluated scenarios. This differs from purely value-driven allocation strategies, which select the resources with the highest KVIs for each request, but often fail to identify globally optimal trade-offs. Any occasional reduction in quality performance with respect to PGM is systematically compensated by significant gains in social and ethical value. These findings confirm the effectiveness of the proposed approach in managing complex service–resource assignments while ensuring a balanced trade-off between technical performance and key societal objectives.

1.3 Case study: Sustainable Video Streaming

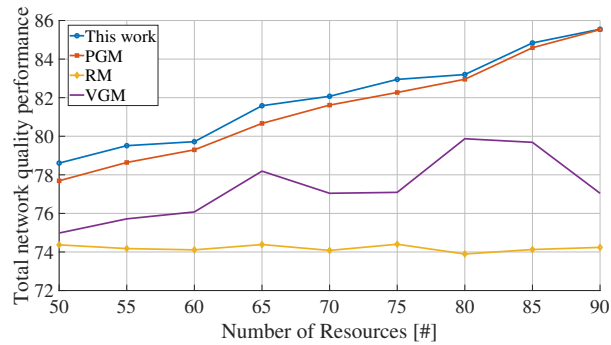
The architectural and methodological perspective developed throughout this chapter can be instantiated in a representative video-streaming scenario. To this end, this section presents a case study on sustainable service orchestration, where intent-driven control is extended beyond conventional performance-oriented decision making so as to jointly account for service quality, infrastructure energy consumption, and equivalent carbon emissions [80].



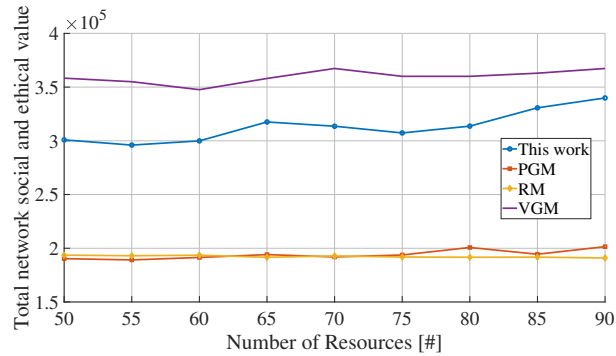
(a) Network quality performance with fixed amount of resources
 $N = 80$



(b) Network social and ethical value with fixed amount of resources
 $N = 80$



(c) Network quality performance with fixed service requests
 $J = 100$



(d) Network social and ethical value with fixed service requests
 $J = 100$

Figure 1.14: Network quality performance and social and ethical value considering an inclusiveness focus [60].

Video streaming provides a particularly relevant use case for this analysis. It already accounts for a large share of global Internet traffic and is associated with a non-negligible environmental footprint, making it a meaningful domain in which to study how orchestration decisions affect both service performance and sustainability outcomes [52]. At the same time, video delivery is strongly sensitive to user-perceived quality, bitrate selection, routing choices, and server allocation, and therefore offers a suitable setting in which to observe the interaction between traditional KPIs and sustainability-oriented objectives.

Within this perspective, the present case study introduces GreenTent, an intent-based orchestration framework designed to jointly account for service performance and environmental sustainability. More specifically, GreenTent extends conventional orchestration by embedding carbon-efficiency requirements as value-oriented objectives alongside traditional service-quality targets. In this way, sustainability is not treated as an external reporting dimension or an ex-post assessment metric, but as an integral part of the orchestration process itself.

From a system perspective, GreenTent instantiates the broader intent-driven workflow discussed in this chapter through a closed-loop architecture comprising intent profiling and translation, orchestration over a virtualized infrastructure view, and runtime assurance. In the considered implementation, this workflow is supported by an Software-Defined Networking (SDN)-enabled control plane based on Open Network Operating System (ONOS) and evaluated over a Mininet emulation environment, thus providing a concrete prototype in which sustainability-aware decisions can be tested under controlled yet realistic operating conditions.

This implementation-oriented perspective is particularly important because it makes it possible to analyze sustainability-aware orchestration not only at the level of abstract optimization objectives, but also in terms of actual routing, server selection, residual link capacity, and service delivery behavior. As a consequence, the case study provides a concrete bridge between the value-aware orchestration principles introduced earlier in the chapter and their practical realization in a dynamic service-management setting.

The relevance of this use case extends beyond video delivery alone. The motivation for considering this scenario lies in the broader dual role played by communication networks within the sustainability landscape. On the one hand, networking infrastructures act as key enablers for social and economic progress by supporting remote work, telemedicine, smart agriculture, digital inclusion, and other services that contribute to the achievement of the UN SDGs. On the other hand, the same infrastructures may generate significant environmental and social externalities, including greenhouse-gas emissions associated with non-renewable electricity consumption, impacts related to equipment manufacturing and resource extraction, and persistent barriers to equitable digital access.

This tension is particularly relevant in light of SDGs 7, 12, and 13, which respectively concern affordable and clean energy, responsible consumption and production, and climate action. In the networking domain, these objectives call for orchestration logics that do not merely maximize throughput or minimize delay, but also explicitly account for energy efficiency, carbon impact, and the broader sustainability profile of service-delivery decisions.

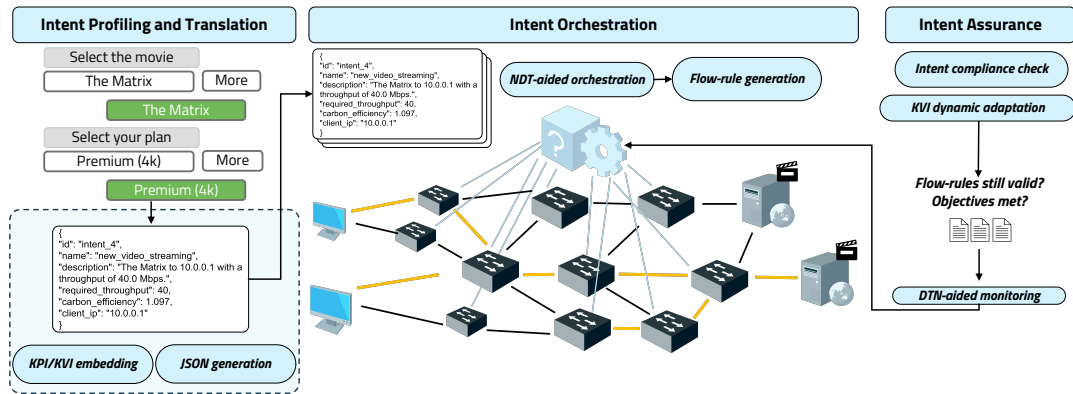


Figure 1.15: The proposed intent-based orchestration framework [80].

A further challenge stems from the fact that sustainability is inherently heterogeneous across both space and time. Carbon intensity may fluctuate according to the varying contribution of renewable generation, while alternative network technologies and routing options may exhibit markedly different energy-efficiency profiles depending on link type, transmission distance, and equipment characteristics. As a consequence, sustainability-aware orchestration cannot be reduced to a static energy-minimization rule, but requires an adaptive decision process capable of incorporating environmental conditions and operational constraints into the service-allocation logic.

1.3.1 GreenTent Framework

This subsection presents the GreenTent framework and outlines the functional workflow through which sustainability-aware service orchestration is performed. As shown in Fig. 1.15, the framework is designed as a closed-loop intent-based control system comprising three main functional blocks, namely *Intent Profiling and Translation*, *Intent Orchestration*, and *Intent Assurance*. This organization is consistent with the broader architectural view developed in this chapter, where high-level requests are progressively translated into enforceable decisions and continuously monitored over time.

From a user perspective, service requests are expressed through a high-level interaction interface, such as a conversational front-end, by specifying only the desired service quality. In the considered video-streaming scenario, this means that users select a content item and an associated subscription plan, thereby expressing service-level requirements in terms of conventional KPIs without explicitly providing sustainability-related parameters. Sustainability requirements are instead inferred by the framework from the operational and energy state of the infrastructure and are incorporated as implicit KVIs.

More specifically, each request is associated with a minimum carbon-efficiency requirement, which captures the minimum amount of delivered data per unit of emitted kgCO_2e . The resulting structured intent therefore combines performance-related service targets with a sustainability-oriented value constraint, so that the

subsequent orchestration process can jointly consider service feasibility, user quality expectations, and environmental impact within a unified decision model.

At the orchestration stage, GreenTent operates on a NDT-based representation of the infrastructure, which maintains an updated view of network topology, residual link capacities, node states, server availability, and energy profiles. This virtual representation combines network-state information with emission-related data, thereby enabling server-selection and routing decisions that account not only for service quality and resource availability, but also for the carbon implications of the selected end-to-end configuration.

Based on this infrastructure view, GreenTent determines the server–path allocation to be applied for each active request. The orchestration logic does not simply identify a feasible route, but searches for configurations that satisfy user-defined bitrate constraints while enforcing a minimum sustainability target expressed through carbon efficiency. In this way, performance and sustainability are jointly embedded into the same control loop rather than being evaluated in separate stages.

The final step is represented by the Intent Assurance function, which continuously monitors whether deployed configurations remain aligned with the original intent specification. If repeated sustainability violations are observed, the framework updates the minimum carbon-efficiency thresholds used in subsequent orchestration cycles. This closes the loop between request interpretation, orchestration, and runtime adaptation, and makes GreenTent suitable for dynamic environments in which network conditions and energy profiles evolve over time.

1.3.2 Sustainability-aware Orchestration Logic

This subsection introduces the system model adopted for the GreenTent case study. The objective is to formalize the infrastructure, the service requests, and the sustainability-related quantities required to support joint orchestration decisions over performance and environmental impact.

The considered infrastructure is represented through a NDT and modeled as a directed graph

$$\mathbb{G} = (\mathbb{V}, \mathbb{E}),$$

where $\mathbb{V}$ denotes the set of network nodes and $\mathbb{E}$ the set of directed communication links. The node set is partitioned into three subsets, namely the set of user devices

$$\mathbb{U} = \{u_1, \dots, u_k, \dots, u_K\},$$

the set of candidate video servers

$$\mathbb{S} = \{s_1, \dots, s_j, \dots, s_J\},$$

and the set of intermediate network nodes

$$\mathbb{N} = \{n_1, \dots, n_w, \dots, n_W\}.$$

Each link $e \in \mathbb{E}$ is characterized by a finite transmission capacity C_e (in Mbps) and a baseline power consumption P_e (in W), representing the power required to keep the link active. Similarly, each node $v \in \mathbb{V}$ is associated with a baseline power consumption P_v (in W), accounting for processing, forwarding, and terminal operations. In line with [81], communication links are assumed to exhibit traffic-independent power consumption, as the power required to process traffic flows varies only marginally over a wide range of traffic loads (e.g., 100-200 mW).

To account for the environmental dimension, the infrastructure is associated with a grid emission factor EF (in kgCO₂e/kWh), representing the carbon intensity of the electricity mix powering the network. Accordingly, the carbon-weighted baseload quantities of links and nodes are defined as

$$\Phi_e = P_e \cdot EF, \quad (1.31)$$

where Φ_e and Φ_v express the equivalent carbon impact of keeping the corresponding element active.

Each user $u_k \in \mathbb{U}$ may generate a video-streaming request r_k , described by the tuple

$$r_k = (b(q_k), \eta_k^{\min}),$$

where $b(q_k)$ denotes the target bitrate associated with the requested video quality level q_k , while $\eta_k^{\min}$ denotes the minimum carbon-efficiency threshold that must be enforced during orchestration. In the considered framework, $b(q_k)$ captures the user-side service requirement, whereas $\eta_k^{\min}$ represents the sustainability-oriented constraint associated with the request.

For each pair (u_k, s_j) , let

$$\Gamma_{u_k, s_j} = \{\gamma_1, \dots, \gamma_m, \dots, \gamma_M\}$$

denote the set of candidate end-to-end paths connecting user u_k to server s_j . Under a steady-state, per-unit-time model, the total power consumption associated with a candidate path γ_m is computed as

$$P_{\gamma_m} = \sum_{v \in \gamma_m} P_v + \sum_{e \in \gamma_m} P_e. \quad (1.32)$$

Starting from the total path power, the corresponding carbon-emission rate is given by

$$C_{\gamma_m} = \frac{P_{\gamma_m} \cdot EF}{3.6 \cdot 10^6}, \quad (1.33)$$

where the denominator converts power and emission units into a consistent per-second carbon-emission rate.

The carbon efficiency achieved when serving request r_k through server s_j over path γ_m is then defined as

$$\eta_{k, s_j} = \frac{b(q_k)}{C_{\gamma_m}}, \quad (1.34)$$

expressed in bit/kgCO₂e. This quantity captures the amount of delivered data per unit of emitted carbon and acts as the main sustainability-oriented metric adopted

in the case study.

Under this formulation, candidate server–path configurations are evaluated jointly with respect to i) service feasibility in terms of residual link capacities and ii) sustainability compliance in terms of the minimum carbon-efficiency threshold. Therefore, the system model provides the quantitative basis for a service-allocation process in which performance and environmental impact are embedded into the same orchestration logic.

Building on the system model introduced above, GreenTent performs sustainability-aware service orchestration through a closed-loop protocol operating on the NDT representation of the infrastructure. The objective is to determine, for each active request, a server–path configuration that preserves the requested service quality while minimizing the carbon impact of service delivery and enforcing a minimum carbon-efficiency requirement.

The orchestration procedure is articulated into two main phases followed by an Intent Assurance step. The overall workflow is summarized in Algorithm 3.

In Phase 1, GreenTent reduces the baseline carbon footprint of the operational topology by progressively deactivating nodes and links that are not necessary to preserve service reachability. Candidate elements are ranked according to their carbon-weighted baseload terms Φ_v and Φ_e , so that the pruning procedure prioritizes the most carbon-expensive infrastructure elements. At the end of this phase, a reduced graph $\mathbb{G}' = (\mathbb{V}', \mathbb{E}')$ is obtained.

Phase 2 performs request-level orchestration over the reduced topology. For each active request r_k , GreenTent evaluates candidate server–path configurations and selects the one maximizing carbon efficiency, while checking the residual-capacity constraints of the traversed links and the sustainability constraint $\eta_{k,s_j} \geq \eta_k^{\min}$. If both conditions hold, the path is deployed through forwarding rules, and the residual capacities are updated accordingly.

When no fully compliant configuration exists, the request is moved to a congestion list. Once all requests have been processed, an updated graph $\mathbb{G}'' = (\mathbb{V}'', \mathbb{E}'')$ is obtained, reflecting the new residual link capacities. The requests in the congestion list, alongside this graph, is fed back as input to Phase 1, which is re-executed to selectively reactivate nodes and links that may be required after capacity updates, while still guaranteeing connectivity between each requesting user and at least one eligible server.

Then, deferred requests are handled through a refined path-selection strategy, consisting of a modified Dijkstra-based algorithm with incremental penalization. The latter includes a quadratic penalty that reflects potential violations of the minimum carbon-efficiency threshold $\eta_k^{\min}$, thereby favoring paths that are more likely to satisfy sustainability constraints while remaining computationally efficient. In particular, the final cost function is defined as follows:

$$\mathcal{J}(\gamma_m) = -\eta_{k,s_j} + \mu \cdot \left(\max(0, \eta_k^{\min} - \eta_{k,s_j}) \right)^2, \quad (1.35)$$

where μ is a tunable penalty coefficient. The first term favors carbon-efficient paths,

Algorithm 3 GreenTent Sustainability-Aware Orchestration

Require: Network graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$; requests r_k with bitrate $b(q_k)$ and threshold $\eta_k^{\min}$

Ensure: Server selection and routing configuration

Phase 1: Topology Reduction

- 1: Rank all $v \in \mathbb{V}$ and $e \in \mathbb{E}$ by decreasing Φ_v and Φ_e
- 2: **for all** $x \in \mathbb{V} \cup \mathbb{E}$ **do**
- 3: Temporarily deactivate x
- 4: **if** $\exists r_k$ such that no feasible path $u_k \rightarrow s_j$ exists **then**
- 5: Reactivate x
- 6: **end if**
- 7: **end for**
- 8: Obtain reduced graph $\mathbb{G}'$

Phase 2: Request-Level Orchestration (Hard)

- 9: $\mathcal{R}_c \leftarrow \emptyset$
- 10: **for all** r_k **do**
- 11: Select $\gamma_m \in \Gamma_{u_k, s_j}$ on $\mathbb{G}'$ minimizing $-\eta_{k, s_j}$
- 12: **if** $\exists e \in \gamma_m$ such that $b(q_k) > C_e$ **or** $\eta_{k, s_j} < \eta_k^{\min}$ **then**
- 13: $\mathcal{R}_c \leftarrow \mathcal{R}_c \cup \{r_k\}$
- 14: **else**
- 15: Install γ_m and update capacities
- 16: **end if**
- 17: **end for**
- 18: Update graph $\mathbb{G}''$

Phase 2 (Refined): Penalized Selection

- 19: **if** $\mathcal{R}_c \neq \emptyset$ **then**
- 20: Re-run Phase 1 on $\mathbb{G}''$ to obtain $\mathbb{G}'''$
- 21: **for all** $r_k \in \mathcal{R}_c$ **do**
- 22: Select γ_m minimizing $\mathcal{J}(\gamma_m)$
- 23: **if** $\exists e \in \gamma_m$ such that $b(q_k) > C_e$ **then**
- 24: Skip r_k
- 25: **else**
- 26: Install γ_m and update capacities
- 27: **if** $\eta_{k, s_j} < \eta_k^{\min}$ **then**
- 28: Mark r_k as allocated with violation
- 29: **end if**
- 30: **end if**
- 31: **end for**
- 32: **end if**

Intent Assurance

- 33: Update $\eta_k^{\min}$ using Eqs. (1.38)–(1.39)

whereas the second penalizes configurations that violate the minimum carbon-efficiency threshold. Higher values of μ impose stricter adherence to the sustainability constraint, while lower values allow greater flexibility under congested conditions.

If a request can be allocated only through a path violating the carbon-efficiency threshold, the request is still admitted in order to preserve service continuity, but it is explicitly marked as *allocated with violation*. These events are then exploited by the Intent Assurance mechanism to update the sustainability threshold for subsequent orchestration cycles.

To this end, let $x_k \in \{0, 1\}$ denote an indicator variable equal to 1 when request r_k is admitted with at least one violation and 0 otherwise. Over an observation interval of duration Δt , the empirical probability of violation is defined as

$$\mathcal{P}_{\text{emp}}(\Delta t) = \frac{1}{K} \sum_{k=1}^K x_k, \quad (1.36)$$

where K denotes the number of requests handled during the interval.

To complement empirical observations with a model-based estimate, violation events are assumed to follow a Poisson process with expected rate $\lambda(\Delta t)$. Under this assumption, the probability of observing at least one violation during the interval is

$$\mathcal{P}_{\text{mod}}(\Delta t) = 1 - e^{-\lambda(\Delta t)}. \quad (1.37)$$

The effective violation probability is then obtained by combining empirical and model-based information as

$$\hat{\mathcal{P}}(\Delta t) = \alpha \cdot \mathcal{P}_{\text{mod}}(\Delta t) + (1 - \alpha) \cdot \mathcal{P}_{\text{emp}}(\Delta t), \quad (1.38)$$

where $\alpha \in [0, 1]$ controls the balance between analytical estimation and measured evidence.

Finally, the minimum carbon-efficiency threshold is updated according to

$$\eta_k^{\min}(t + \Delta t) = (1 - \hat{\mathcal{P}}(\Delta t)) \cdot \eta_k^{\min}(t), \quad (1.39)$$

thereby aligning the sustainability requirement with the carbon efficiency that is realistically achievable under the current network and energy conditions.

From a computational viewpoint, the overall cost is dominated by the topology-reduction phase, which combines iterative pruning and repeated reachability checks, yielding a worst-case complexity of

$$O(|\mathcal{U}| \cdot (|\mathcal{V}| + |\mathcal{E}|)^2). \quad (1.40)$$

By contrast, the Dijkstra-based request-level orchestration phase has complexity

$$O(|\mathcal{U}| \cdot (|\mathcal{E}| + |\mathcal{V}| \log |\mathcal{V}|)), \quad (1.41)$$

with an additional cost of the same order when the refined penalized-selection stage is triggered for deferred requests.

1.3.3 Performance Evaluation

The effectiveness of GreenTent in adaptive service orchestration is evaluated on a Mininet-emulated testbed managed by the ONOS controller and deployed on an Intel Core i7-10750H host running Ubuntu 22.04.4 LTS. The considered topology comprises $|\mathbb{V}| = 31$ nodes, including 18 intermediate nodes, 10 user devices, and 3 video servers, interconnected by $|\mathbb{E}| = 44$ directed links. Video content is delivered through GStreamer-based servers using RTSP streaming sessions.



(a) Perceived video quality with GreenTent.



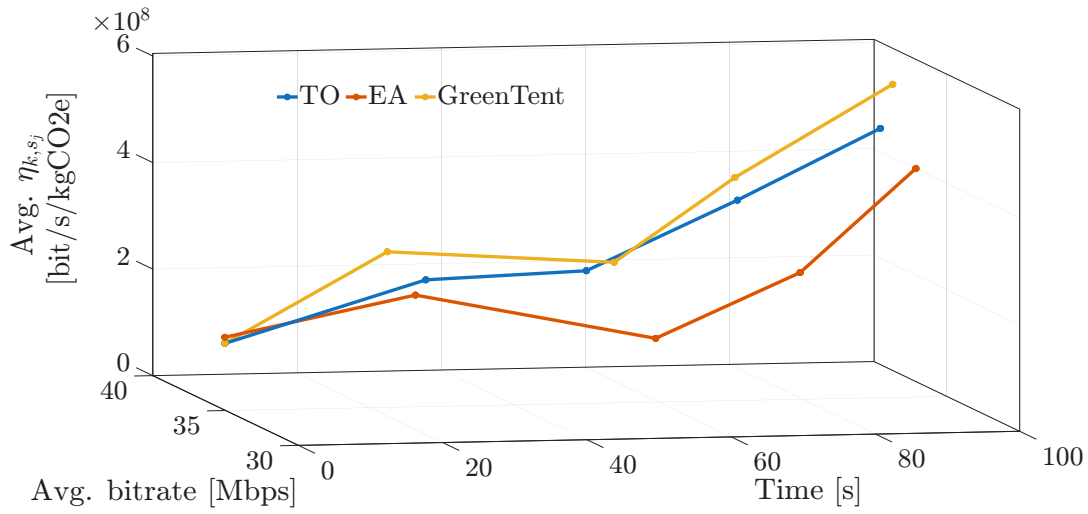
(b) Perceived video quality with TO orchestration (no sustainability criteria).



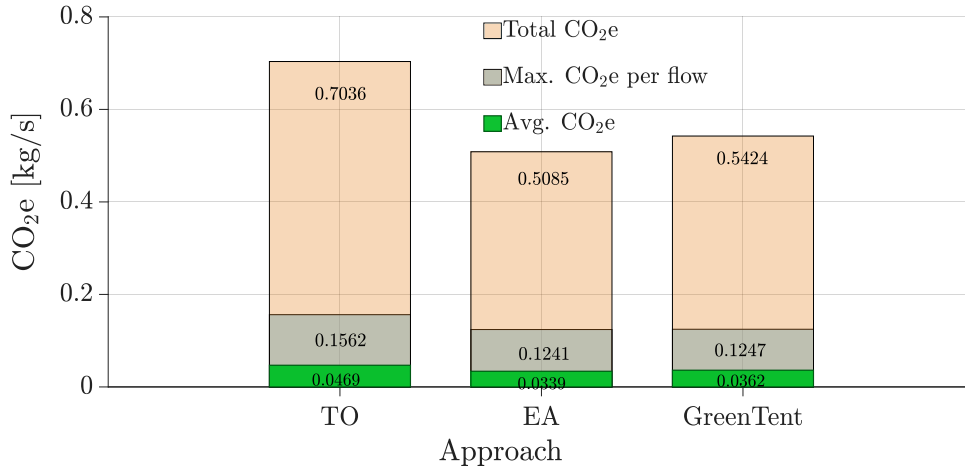
(c) Perceived video quality with the EA approach (no network orchestration).

Figure 1.16: Perceived video quality for the first service request under (a) GreenTent, (b) throughput-oriented orchestration (TO), and (c) energy-aware application-level approach (EA) [80].

Node and link energy profiles are initialized from representative commercial hardware specifications, with link baseload power consumption ranging from 1.2 to 12 W [57], [82]. In accordance with the system model described in Section 1.3.2, all



(a) Average bitrate and carbon efficiency with increasing concurrent service requests.

(b) Total, maximum, and average CO₂ emissions with 15 concurrent service requests.Figure 1.17: Average delivered bitrate, carbon efficiency, and CO₂ emissions under increasing load [80].

energy-related quantities represent baseline, traffic-independent power consumption. The grid emission factor is set to $EF = 475 \text{ gCO}_2\text{e/kWh}$, which, for the sake of simplicity, it is assumed to be equal to the global average, while the initial minimum carbon-efficiency threshold is fixed to $\eta_k^{\min} = 2.68 \cdot 10^6 \text{ bit/kgCO}_2\text{e}$ and dynamically updated through the Intent Assurance mechanism with observation interval $\Delta t = 30 \text{ s}$.

GreenTent is compared against two baseline strategies. The first one is a throughput-oriented orchestration approach, denoted as TO, which optimizes service delivery without considering sustainability-related criteria. The second one is an energy-aware application-level approach, denoted as EA, which aims at reducing energy consumption without performing network-level orchestration of video quality. Video contents are available at three quality levels, namely 720p, 1080p, and 2160p, and service requests are generated through interactions with the Rasa-based front-end, emulating realistic user behavior biased toward cost-efficient subscription plans.

A representative scenario considers concurrent requests for high-resolution video content at 2160p. Figure 1.16 provides a qualitative comparison of the perceived video quality for a single request under the three considered strategies. Both GreenTent and TO satisfy throughput requirements and achieve high-resolution playback. However, GreenTent preserves comparable visual quality while reducing network-level energy consumption and emissions. By contrast, the EA approach minimizes energy usage without enforcing performance constraints, resulting in visibly degraded video quality.

While Fig. 1.16 focuses on a single request, Fig. 1.17 evaluates network behavior under increasing traffic load in order to assess the scalability of the proposed framework. More specifically, Fig. 1.17(a) shows the evolution of average delivered bitrate and network carbon efficiency as the number of concurrent service requests increases, whereas Fig. 1.17(b) reports the corresponding CO₂e emission rates when the network handles 15 concurrent requests. Since flows are progressively injected over time, the figure also captures the dynamics induced by residual-capacity variations during orchestration.

The results show that the higher bitrate achieved by TO does not translate into proportional carbon-efficiency gains because of the associated increase in CO₂e emissions. Conversely, EA attains high carbon efficiency at the expense of service quality. GreenTent instead identifies a more balanced operating point, limiting the average bitrate degradation to 5.47% with respect to TO while improving carbon efficiency by up to 15.54% at $t = 100$ s. Compared with EA, GreenTent provides up to 12.93% higher average bitrate and up to 22.70% higher carbon efficiency.

The stacked-bar representation in Fig. 1.17(b) further shows that TO incurs the highest total CO₂e emission rate, equal to $0.1562 + 0.0469 + 0.7036$ kg/s, whereas EA and GreenTent achieve lower totals, equal to $0.1241 + 0.0339 + 0.5085$ kg/s and $0.1247 + 0.0362 + 0.5424$ kg/s, respectively. These results indicate that per-request optimization without topology-level control leads to higher system-wide carbon costs, whereas GreenTent benefits from the combined action of topology reduction and request-level sustainability-aware orchestration.

Overall, GreenTent achieves a balanced trade-off between sustainability and service performance. In particular, it significantly reduces power consumption and CO₂ emissions with respect to throughput-oriented orchestration while preserving delivered bitrate and user-perceived quality even under high network load.

Concluding Remarks

This chapter proposed an end-to-end architecture for secure and value-aware intent processing, addressing **RQ1** along three interconnected dimensions: trusted request admission, KPI/KVI-aware intent translation, and multi-objective orchestration.

The MID module prevents harmful or adversarial requests from entering the orchestration pipeline by operating at the conversational front-end, before any translation step is performed. The value-driven translation framework enriches the structured intent with both conventional KPIs—namely service delay, data rate, and packet loss—and broader KVIs, capturing environmental sustainability, trustworthiness, and inclusiveness. In this way, orchestration objectives are aligned

with the UN SDGs. The TETREES optimizer leverages this enriched representation to compute exact Pareto-efficient trade-offs between network quality and social/ethical value. This capability is further demonstrated through the sustainable video-streaming case study, where GreenTent achieves a balanced operating point between carbon efficiency and delivered service quality.

To clarify how each dimension propagates from intent translation to the optimization model, Table 1.8 provides a unified overview of the proposed framework, prior to the DT-assisted extensions introduced in Chapter 2.

Table 1.8: Role of KPIs and KVIs across the proposed framework.

Dimension	Role in intent translation	Role in optimization
KPIs	Extracted from natural-language requests by the LLM-assisted translation module and mapped to normalized performance attributes and quality tolerances $Q_{s_j}^{\text{tol}}$.	Aggregated into the network quality performance function $Q(X)$ via per-resource quality scores Q_{r_n, s_j} ; enforced as hard service-level constraints in the binary assignment model; enforced at runtime by GreenTent.
KVI – Sustainability	Captures environment-related requirements such as carbon efficiency and energy constraints embedded in the user request.	Modelled as the difference between carbon offset and emission rate; combined into the social/ethical value function $V(X)$ with dedicated weight w'_1 ; enforced at runtime by GreenTent.
KVI – Trustworthiness	Encodes security, robustness, and privacy expectations beyond conventional SLA constraints.	Derived from cyber-risk and cyber-confidence indicators; contributes to $V(X)$ through weight w'_2 , penalising assignments with elevated security exposure.
KVI – Inclusiveness	Represents fairness and accessibility goals articulated in service requirements.	Expressed via mean-time-to-failure and failure probability of each resource; weighted by w'_3 in $V(X)$, rewarding assignments that improve service continuity across all user groups.

All dimensions are processed through the unified architecture of Fig. 1.1: the LLM-assisted translation layer extracts KPI and KVI targets from natural-language requests, the MID module enforces secure admission, and TETREES resolves the resulting bi-objective optimisation problem.

Chapter 2

Digital Twin-Driven Intelligent Orchestration

As future communication networks grow in complexity, orchestration can no longer be framed as a mere resource-allocation problem, but rather as a broader decision-support challenge in which the system must continuously interpret context, anticipate evolution, and select actions aligned with service objectives over time. In this context, DTs, and more specifically NDTs, have emerged as promising enablers, since they provide synchronized and operationally meaningful abstractions of the underlying infrastructure, allowing the network state to be observed, alternative configurations to be explored, and the impact of orchestration decisions to be estimated before enforcement in the physical system.

Yet, the mere availability of a digital counterpart does not automatically result in intelligent orchestration. While a twin can expose measurements, models, and predictions, it does not by itself determine which objectives should be prioritized, how conflicting constraints should be balanced, or which decision is most appropriate under dynamic conditions. The real challenge, therefore, is not only to build accurate twins, but to endow them with decision-making capabilities that transform data into actionable and trustworthy orchestration choices. In this sense, the proposed DT-assisted orchestration frameworks address **RQ2** by demonstrating how DTs can evolve from descriptive replicas of the infrastructure into intelligent decision-support entities for network orchestration. Rather than being limited to monitoring, synchronization, and prediction, the twin is framed here as an active component of the control loop, able to expose domain knowledge, evaluate alternatives, and support trustworthy decisions under heterogeneous requirements and changing conditions. the network.

Against this background, this chapter develops a mathematical-reasoning perspective for DT-assisted network orchestration. It first reviews related work on DT-assisted orchestration, intent-driven control, and disaster-response task offloading. It then introduces two complementary algorithmic contributions — a MCDM framework based on the TOPSIS methodology for context-aware domain selection, and a game-theoretic matching mechanism grounded in the Deferred Acceptance Algorithm (DAA) for joint KPI and KVI optimization — and consolidates them within a practical platform implementation in the context of the COHERENT project.

2.1 Background and Related Works

As highlighted in Chapter 1, the evolution of future networked systems calls for orchestration approaches able to address not only conventional performance requirements, but also wider objectives associated with resilience, trustworthiness, and sustainability. Against this background, the present chapter narrows the focus to the research directions that are most closely aligned with DT-assisted network orchestration. More specifically, this section discusses four strongly interconnected areas: optimization-oriented service orchestration, NDT-assisted orchestration, intent-driven orchestration, and task offloading in disaster-response scenarios across heterogeneous traversable domains.

2.1.1 Service orchestration in advanced network environments

Service orchestration in B5G networks requires the coordinated selection and activation of distributed resources so as to satisfy heterogeneous service requirements under constrained conditions. Early research in this area mainly addressed conventional KPIs, formulating optimization problems aimed at reducing deployment costs, meeting deadlines, or limiting delay in service execution and workflow scheduling [83], [84], [85], [86], [87]. In parallel, task-allocation strategies grounded in matching theory have also been proposed to improve the efficiency of task offloading in distributed IoT–fog environments [88].

Although these contributions provide valuable algorithmic tools, they are predominantly centered on technical efficiency. As a consequence, they only partially capture the richer decision space that characterizes emerging network environments, where orchestration choices may also need to account for risk, trust, and value-oriented objectives. This limitation becomes particularly relevant in B5G and pre-6G scenarios, where service provisioning increasingly spans heterogeneous domains and must remain aligned with both operational targets and broader stakeholder expectations [9], [13], [63], [89].

2.1.2 Digital-twin-assisted orchestration

A major evolution of the orchestration literature has been enabled by the introduction of NDTs, which provide data-driven virtual abstractions of network domains and allow candidate configurations to be explored before actual deployment [90], [91]. This capability is particularly useful in advanced service orchestration, because it enables the management system to reason on resource availability, predicted behavior, and feasible deployment alternatives through a virtual representation of the infrastructure.

Several recent studies have exploited this opportunity in different ways. In [92], DT function virtualization is coupled with deep reinforcement learning to support adaptive service response in 6G networks. In [93], the authors propose an adaptive service-provisioning mechanism in which the NDT is used to simulate resource aggregation satisfying service requirements, and the resulting problem is addressed through a Boolean-satisfiability formulation with heuristic resolution. In

[94], DTs support Service Function Chain (SFC)-enabled service provisioning in mobile edge computing by predicting reliability and assisting two optimization problems aimed at minimizing costs while maximizing admission capability. A similar orchestration-support role is assigned to the NDT in [95], where the twin assists federated-learning service provisioning over mobile edge networks.

Other works exploit the virtual counterpart more explicitly to expose domain characteristics and drive resource-allocation decisions. In [96], the NDT represents processing and access-related parameters so as to minimize the average processing delay in mobile edge computing. In [97], a DT approach is adopted for joint user association, task offloading, and resource allocation in Ultra-Reliable Low-latency Communications (URLLC) edge networks, with the objective of reducing end-to-end offloading latency. In [98] and [99], DT assistance is used for resource allocation in vehicular and sensing-enabled environments, while [100] extends this line to Unmanned Aerial Vehicle (UAV)-aided Internet of Vehicles scenarios. Taken together, these studies confirm that NDTs can play a central role in exposing the state and capabilities of candidate domains for orchestration purposes.

Nevertheless, most of these solutions still focus on a restricted subset of KPIs, such as latency, throughput, computational efficiency, or acceptance ratio. Even when the NDT effectively supports prediction and simulation, the final orchestration objective is often limited to technical optimization, with weaker attention to higher-level service abstractions and broader KQIs. This leaves open the question of how DTs can be integrated into orchestration frameworks that jointly reason on performance, security, cost, and service value.

2.1.3 Intent-driven orchestration

A complementary research direction concerns IBN, which provides the semantic abstraction through which users and operators can express desired outcomes without directly specifying low-level configurations [20], [21], [22]. This paradigm is particularly suitable for 5G systems, where service demands are heterogeneous and where automation must progressively bridge the gap between high-level objectives and deployable actions.

Recent contributions have explored this potential in several orchestration settings. In [101] and [36], intent-aware orchestration is used to support resource management and proactive control, including reinforcement-learning-based scheduling in Radio Access Network (RAN) slicing scenarios and AI-driven orchestration for consumer-electronics services. In [102], a multi-tenant IBN approach is adopted for Virtual Network Function (VNF) placement, while [103] proposes an intent fulfillment system centered on the translation of high-level goals into enforceable actions. Similarly, [104] develops an intent-driven closed-loop control and management framework for 6G Open RAN environments.

These studies clearly show that IBN can strengthen orchestration by formalizing user demands, enabling automated translation, and supporting closed-loop control. Yet, most available solutions mainly address the semantic and operational handling of intents, whereas the dynamic representation of domain state is often treated separately. In other words, the literature on IBN and the literature on NDTs have both

made significant progress, but their joint use in service orchestration remains insufficiently explored.

This gap is especially relevant when orchestration must consider not only KPIs, but also KVIIs. In emerging network environments, service delivery may need to account for cyber risk, economic sustainability, and trustworthiness in addition to performance constraints. However, current intent-driven orchestration approaches still tend to focus on a relatively narrow set of technical requirements, thus leaving room for more comprehensive frameworks in which intents express heterogeneous demands and NDTs expose the information needed to evaluate them jointly.

2.1.4 Task offloading and disaster-response scenarios

A further body of literature becomes relevant when orchestration is considered under disaster conditions. Natural disasters such as earthquakes, volcanic eruptions, and floods may damage buildings, disrupt communication infrastructures, and cause widespread service unavailability. In such situations, rescue operations often rely on the timely collection, transfer, and processing of data, which in turn requires computationally demanding tasks to be offloaded toward suitable nearby network segments.

On the one hand, a substantial number of studies investigate the use of UAVs in disaster and emergency scenarios [105], [106], [107], [108], [109], [110]. These works confirm the importance of UAVs as flexible and rapidly deployable entities for monitoring and rescue support. However, only a subset of them explicitly focuses on network-processing task offloading, as in [105], [107], [109]. More importantly, these contributions also highlight the difficulty of designing offloading strategies that can react to the dynamic and heterogeneous state of the traversed environment.

On the other hand, several works employ DTs to collect data and model the state of candidate network domains [97], [111], [112], [113], [114]. In [111] and [97], DTs support task offloading and resource allocation in vehicular and edge-network settings. In [112], the twin assists computation offloading and service caching for intelligent transportation systems, while [113] and [114] show how DT models can support disaster-oriented evaluation and smart-city protection scenarios. These studies demonstrate the usefulness of DTs for representing domain state, but they do not fully address dynamic task offloading in post-disaster settings involving traversable heterogeneous domains and in-network processing support.

An additional limitation of the current literature is that trust and social relationships are rarely incorporated into the domain-selection logic. Yet, in disaster scenarios, the suitability of a domain should not depend only on communication and computation parameters, but also on its reliability and trustworthiness. This motivates the integration of a Trust Management System (TMS) within an intent-driven orchestration framework, following the architectural perspective of ITU-T [23] and building on social-trust mechanisms such as those discussed in [115]. In this way, trust can be treated as an explicit support dimension for domain selection during task offloading. To make the positioning of the disaster-oriented contribution more explicit, Table 2.1 summarizes the main features addressed by the most relevant related works, comparing them with one the approach developed in this chapter.

Table 2.1: Review of related works on disaster-oriented task offloading and DT-assisted domain selection.

Work	UAV	Rescue	Offload	Dynamic	Trust/Social	DT
[105]	✓		✓			
[106]	✓	✓				
[107]	✓	✓	✓			
[108]	✓	✓		✓		
[109]	✓	✓	✓			
[110]	✓	✓		✓		
[111]			✓	✓		✓
[97]			✓			✓
[112]			✓	✓		✓
[113]		✓				✓
[114]		✓				✓
This thesis	✓	✓	✓	✓	✓	✓

2.2 Surviving disaster events via dynamic in-network processing assisted by NDTs

The first specific contribution [116] presents a dynamic decision-making framework for post-disaster in-network processing. The proposed approach exploits NDTs to support the selection of the most suitable network segment for task offloading under rapidly changing and disrupted conditions. By continuously abstracting heterogeneous domains through indicators such as bandwidth, computation, storage, and trustworthiness, NDTs enable informed and timely orchestration decisions, thus preserving service continuity during disaster recovery operations. In particular, the framework addresses scenarios in which UAVs, acting as elements of a Non-Terrestrial Network (NTN), capture video data for monitoring affected areas and offload the corresponding processing tasks to available network segments. To this end, and in line with the International Telecommunication Union (ITU)-T ITN 2030 vision, the methodology exploits NDTs to characterize network domains, identify those affected by disaster events, and replace them with available and reliable segments capable of supporting in-network processing. The dynamic selection process is based on TOPSIS, an MCDM technique widely adopted for network selection in heterogeneous wireless environments, and jointly evaluates the status, readiness, and trustworthiness of the candidate domains.

2.2.1 The Reference Scenario and the Proposed MCDM Solution

The considered scenario consists of a conventional multi-domain network in which several network segments cooperate to support end-to-end services. Depending on their capabilities and on the sequence of tasks required to provide the service, including storage and in-network processing, these heterogeneous domains are

assigned different responsibilities. The considered domains span terrestrial networks, NTN segments, and integrated Terrestrial Network-Non Terrestrial Network (TN-NTN) environments.

Within this framework, an UAV traverses the network segments across all domains. First, the UAV performs data acquisition while crossing each domain, collecting information such as videos of industrial assets and critical infrastructures. These data are used for evaluation, inspection, and monitoring purposes, especially in locations that are difficult to access for maintenance or completely unreachable.

Second, the UAV can offload the acquired data to a suitable network segment for processing. In this way, the collected videos can support recovery actions in domains impacted by natural disasters, enabling timely interventions and rescue operations while avoiding the loss of in-network processing capability in conditions where coverage and services are disrupted.

Figure 2.1 depicts the reference scenario together with the IBN-based architecture, which extends the solution proposed in [22] and follows the ITU-T specifications in [117]. The different network segments are grouped into network domains, which represent the physical infrastructures responsible for handling processing and storage requests.

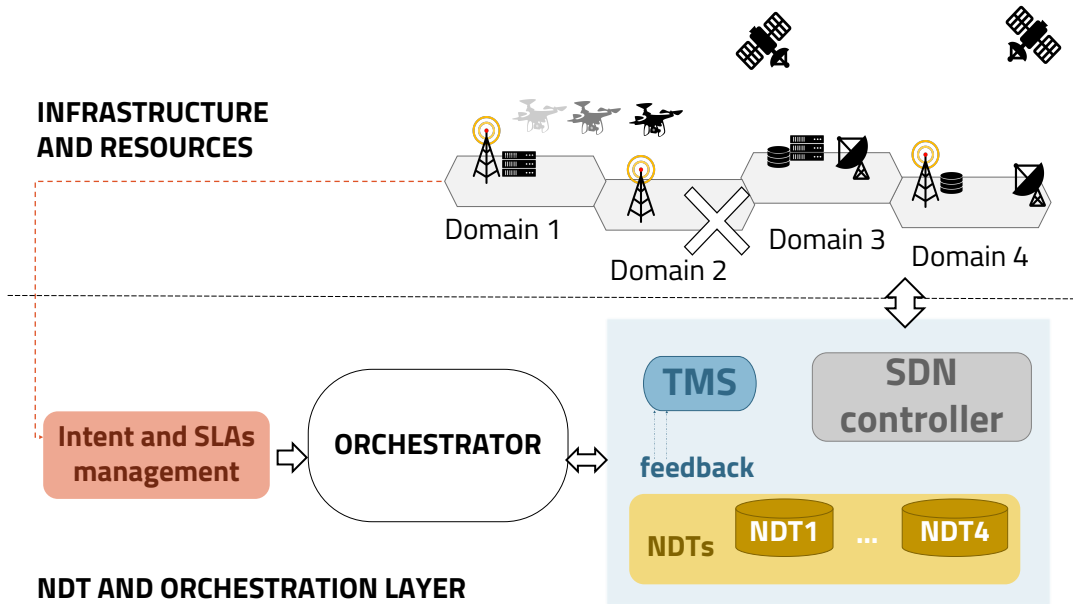


Figure 2.1: Reference scenario and IBN-based network architecture [116].

In this framework, end-to-end service requirements are expressed as an *intent*. This intent requires a target performance level to enable the delivery of the desired service. Such requirements are formalized through the SLA, which ensures that network providers comply with the specified objectives. According to the ITU-T framework described in [117], intents are processed and translated into a set of actions and policies that are then executed by the intent and SLA management module.

Since network segments are heterogeneous and their conditions evolve over time, DTs become essential to represent the characteristics and capabilities of the

corresponding domains. Their use supports network monitoring and maintenance prediction, while also assisting the selection of the most suitable domain for in-network processing through the analysis of the extracted features.

To manage this abstraction of network resources, an SDN controller continuously populates and updates the performance indicators of the DTs through dedicated APIs and modeling languages such as Yet Another Next Generation (YANG)-CBOR, OMA lightweight Machine-to-Machine (M2M), and Digital Twin Definition Language (DTDl). In addition to enabling the configuration of network functionalities, this programmable controller also supports intent implementation.

Moreover, the TMS is responsible for evaluating and acquiring the trustworthiness attribute by analyzing the historical behavior of network segments. More specifically, it relies on an automated feedback-based mechanism to derive a trust parameter, which determines the reputation of each involved network segment. This value is periodically updated after the completion of each end-to-end service, thereby improving the reliability of future intent deployments by identifying trustworthy domains for in-network processing.

Beyond the above modules, the orchestrator plays a central role in the overall management of the multi-domain architecture by handling both service and network orchestration. For each service demand, it evaluates the features contained in the NDTs in order to identify the network segment that is best suited to support the requested service. It then proceeds with task allocation, intent deployment, and monitoring of compliance with the SLA, service models, and service descriptions. The algorithm integrated into the orchestrator for selecting the most appropriate domain for in-network processing is described in subsection 2.2.3.

2.2.2 Network Digital Twins Parameters

The main purpose of the NDT is to build a network model that mirrors the performance indicators of the physical infrastructure. In this way, network metrics and characteristics can be managed through an abstract representation of the actual network state. By relying on this abstraction, the orchestrator can comprehensively monitor network segments, better understand service disruptions, mitigate them, and ultimately support intent fulfillment.

Within the proposed framework, the digital representation of network segments is exploited to identify the most suitable domain for task offloading and in-network processing. The goal is to determine the configuration that best satisfies the predefined service requirements.

The creation of an NDT requires collecting a dataset of relevant network-related information. Accordingly, the proposed method extracts a set of performance indicators from each network segment, capturing radio conditions, network capabilities, and reliability. The selected indicators include bandwidth, CPU performance, available RAM, storage capacity, and trustworthiness.

1. Bandwidth: measured in [MHz], it represents the minimum guaranteed available bandwidth for communications within the considered network segment.
2. CPU: measured in [MHz], it denotes the processing capability of the network resources in the domain.

3. RAM: measured in [KB], it expresses the minimum guaranteed amount of RAM available in the network segment.
4. Storage availability: measured in [GB], it quantifies the available storage space in the domain.
5. Trustworthiness: this is a value in the range [0,1], determined by the TMS, and represents the declared and proven reliability of a network segment. It is computed as the product of two factors, namely the domain reputation and the social tier between domains.

2.2.3 The Conceived Algorithm Based on TOPSIS Methodology

The algorithm devised to identify the most suitable domain for in-network processing is summarized in the pseudo-code reported in **Algorithm 4**.

Algorithm 4 Pseudo code of the conceived decision-making algorithm

```

1: for Every domain do
2:   check_UAV_capacity()
3:    $n \leftarrow n\_traversable\_domains()$ 
4:   offloading_options  $\leftarrow TOPSIS(n)$ 
5:   if !offloading_options then
6:      $domain \leftarrow domain + 1$ 
7:     update_time()
8:   else
9:      $d \leftarrow domain\_bw\_avail(offloading\_options)$ 
10:    if  $decision = d$  and  $id = d$  then
11:       $0 \leftarrow capacity\_UAV$  ▷ Data overload
12:       $domain \leftarrow domain + 1$ 
13:      update_time()
14:    else
15:       $domain \leftarrow domain + 1$ 
16:      update_time()
17:    end if
18:  end if
19: end for

```

The algorithm explicitly accounts for UAV constraints, such as battery capacity during video acquisition along the flight path. Based on these constraints, it estimates how many domains the drone can traverse before battery recharge becomes necessary. In addition, when storage is still available onboard, the drone can record videos while crossing the corresponding domains.

The proposed solution adopts TOPSIS, a widely used MCDM method for network selection in heterogeneous wireless environments. The method identifies the ideal solution as the network segment that provides a cost-effective QoS, characterized by the highest values of CPU, RAM, free storage, and trustworthiness among

the observed domains. It then evaluates how close each candidate domain is to this ideal solution through the Relative Closeness (RC) metric.

Within this approach, the UAV is assumed to spend approximately 10 minutes in each network segment while traversing domains separated by distances between 500 meters and 1 kilometer and moving at speeds between 3 km/h and 6 km/h. Consequently, the parameters of the DTs are updated every 10 minutes.

Considering all traversable domains, the implemented TOPSIS procedure builds a decision matrix in which each row corresponds to a domain described by CPU, RAM, available storage space, and trustworthiness. The decision function then computes the ideal solution and assigns an RC value to each candidate domain. The candidate solutions are ranked in decreasing order of RC, and all domains with an RC below an empirically defined threshold are excluded from the selection process. A communication bandwidth check is then performed to verify whether data can actually be offloaded while the UAV is passing through the selected domain.

The final domain is selected according to the following conditions:

1. The domain allows data offloading given the available bandwidth.
2. The domain is considered both optimal and trustworthy for data offloading according to its RC value.
3. The domain satisfies energy and data-capacity constraints.

If no offloading option is available, the UAV recharges its battery when necessary and then proceeds toward the next network segment. The same procedure is repeated after data acquisition in each subsequent domain. If the previously selected domain no longer matches the ideal solution, the decision is updated. This guarantees a dynamic decision-making process that can quickly react to major changes in network conditions. In particular, when a natural disaster occurs, the framework modules responsible for the digital representation of the network detect and report the event. Consequently, the features of the affected domain exhibit clear anomalies, which indicate a possible disruption and trigger a reevaluation of the most suitable offloading domain.

2.2.4 Performance Evaluation

The performance of the proposed approach is evaluated through computer simulations. A MATLAB script is used to model the DTs representation of the domains traversed by the UAV and to implement the proposed decision-making algorithm described in subsection 2.2.3.

The analyzed scenario includes eight different domains, each characterized by time-varying CPU, RAM, available storage, and trustworthiness values. These parameters change over the day and are updated every 10 minutes, consistently with the time spent by the drone within each domain.

The UAV considered in the study has a storage capacity of 650 MB. While moving across domains, it captures a 130 MB video of the target object for monitoring purposes. As the drone crosses the network segments, the orchestrator runs the proposed algorithm while accounting for both the drone storage availability and

its battery level. The battery model follows the energy consumption behavior described in [118].

Figures 2.2 and 2.3 show an example of dynamic domain selection. During the interval from 00:00 to 00:10, the drone traverses domain 1. After evaluating the feasible domains that can be reached within the drone battery lifetime, the orchestrator applies the proposed algorithm and selects domain 4 as the best candidate for task offloading. This choice is determined by the TOPSIS procedure, which evaluates the relevant features extracted from the DTs. Domain 4 is selected because of its favorable performance indicators, including adequate CPU, RAM, and trustworthiness, as illustrated in Figure 2.2. From 00:10 to 00:20, the drone enters domain 2. The TOPSIS procedure confirms that domain 4 remains the best option for data offloading and in-network processing. In the third phase, at 00:20, a disruption affects domain 4 while the drone is traversing domain 3. The anomaly appears in the feature representation provided by the DTs, which prompts the TOPSIS procedure to recompute the selection of the target domain. In this case, domain 6 becomes the one closest to the ideal solution. Finally, in the interval from 00:50 to 01:00, the drone reaches domain 6 and successfully completes the data offloading operation for in-network processing.

2.2.5 Simulation Results

To assess the effectiveness of the proposed strategy, it is compared against two baseline methods. In the first baseline, data offloading and processing are performed on a randomly selected network segment, and the results are averaged over 15 different seeds. In the second baseline, the orchestrator applies a deterministic strategy, carrying out the selection only when the drone storage is completely full.

Figure 2.4 reports the performance of the proposed solution compared with the baseline approaches. The considered performance metrics are:

- the amount of offloaded data;
- the number of domains in which the drone misses video acquisitions because it cannot offload the stored data and therefore has no remaining storage capacity;
- the overall processing availability, measured as the ratio between the offloaded data and the total amount of data acquired by the drone, expressed as a percentage.

2.2. Surviving disaster events via dynamic in-network processing assisted by NDTs71

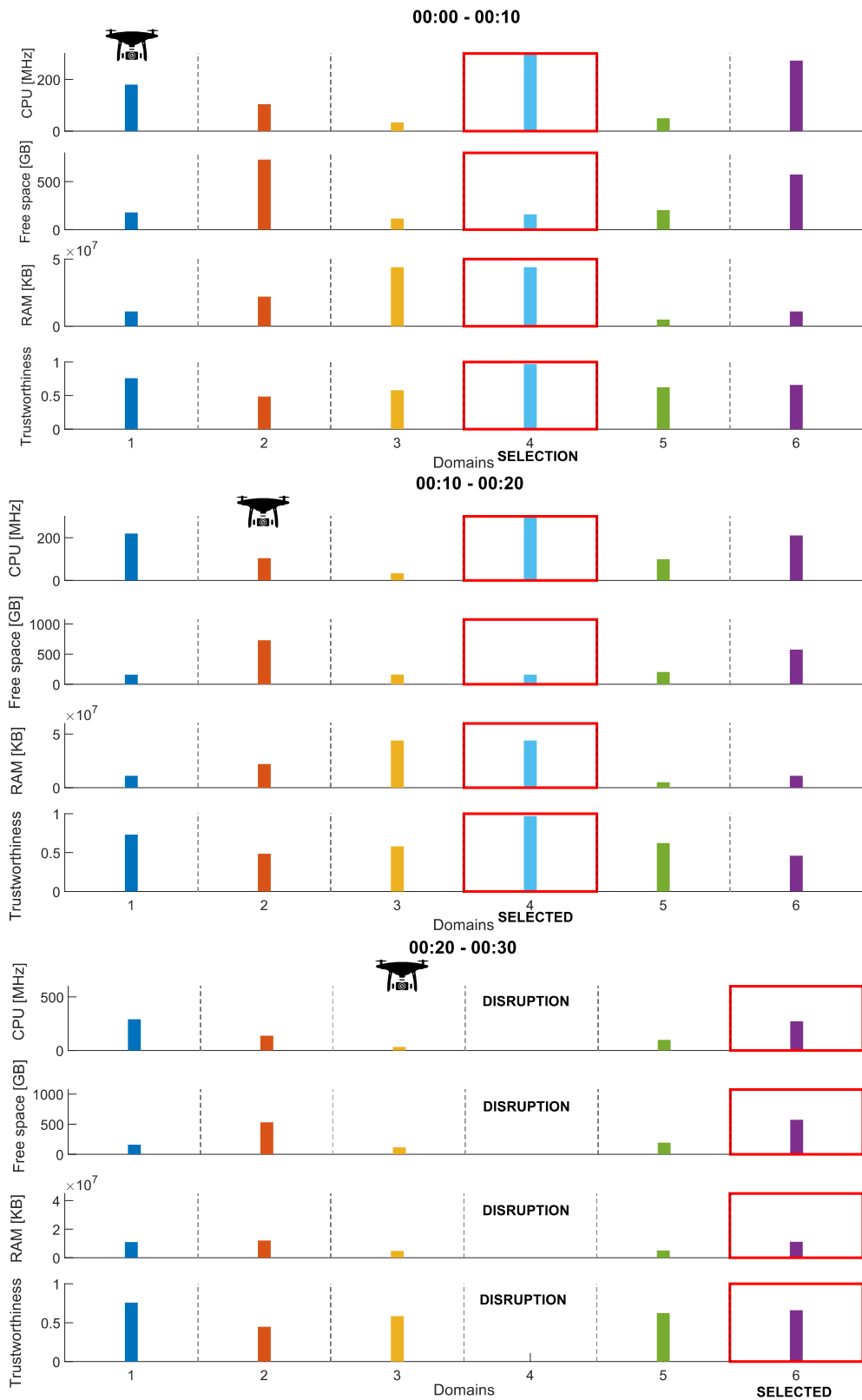


Figure 2.2: Overview of the drone flight and domain selection [116].

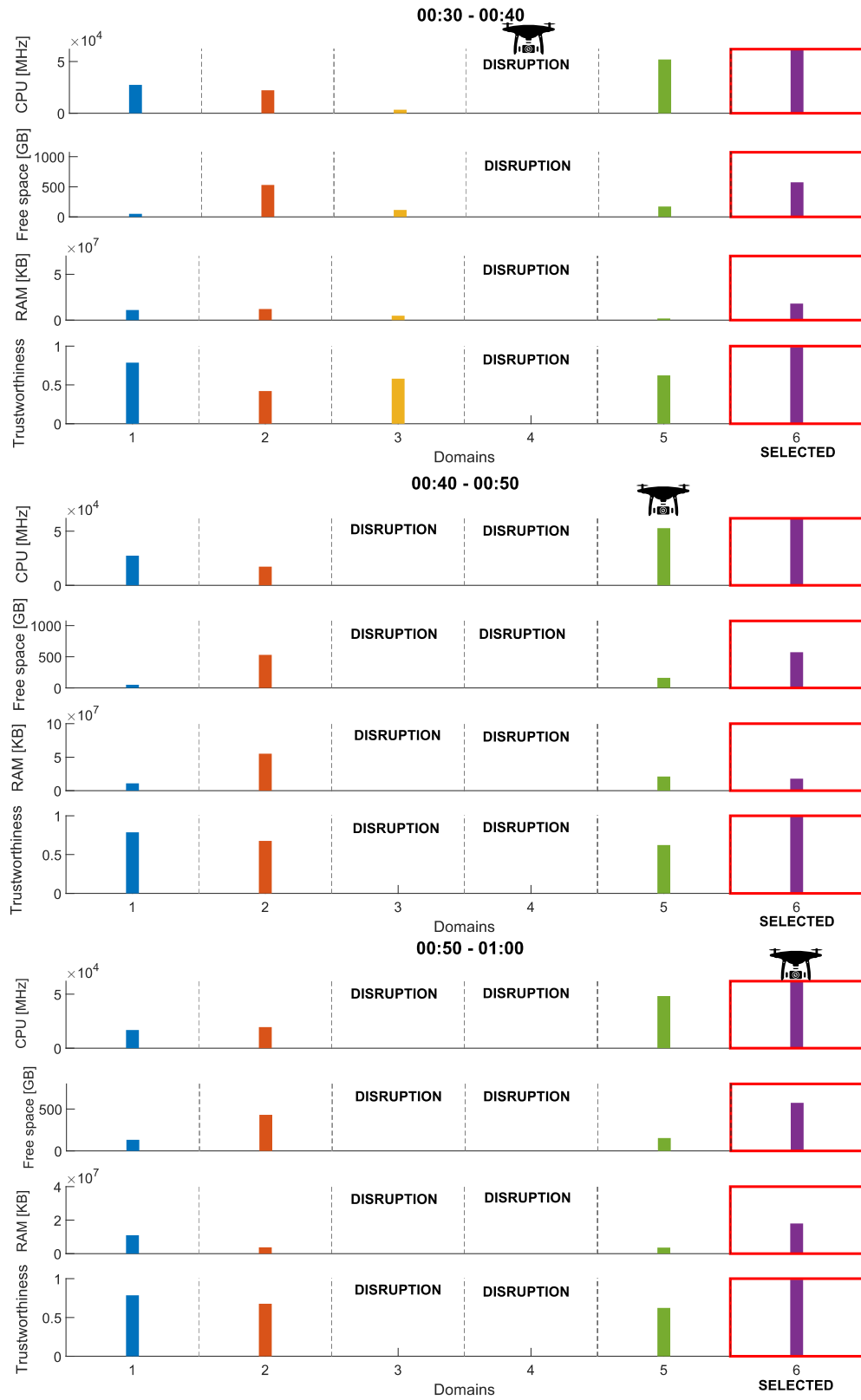


Figure 2.3: Overview of the drone flight and domain selection [116].

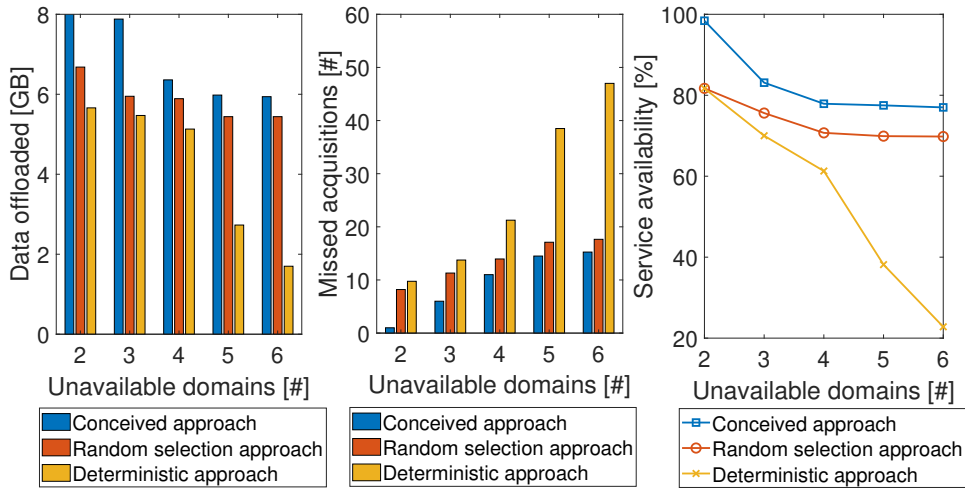


Figure 2.4: Offloaded data, missed acquisitions and overall service availability for CPU SLA focus [116].

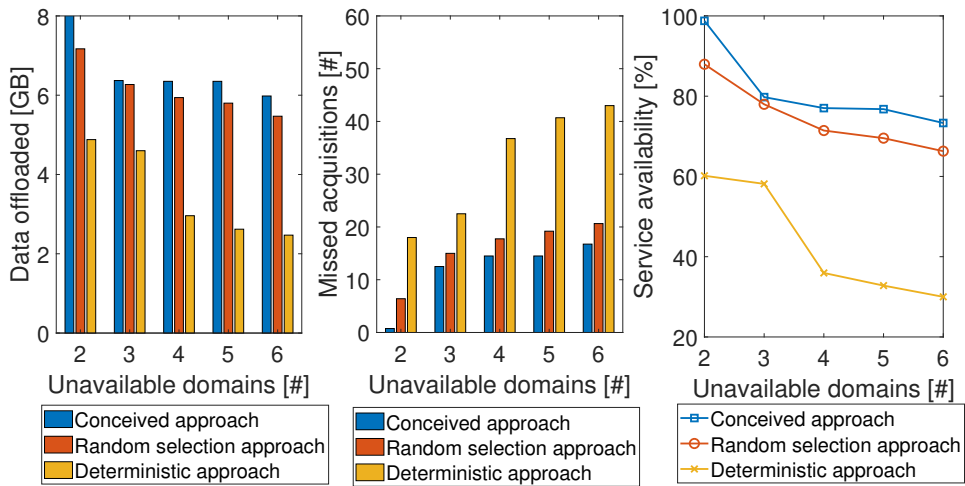


Figure 2.5: Offloaded data, missed acquisitions and overall service availability for trustworthiness SLA focus [116].

In particular, Figure 2.4 refers to a use case in which the SLA requires a high level of CPU performance. The plots show the behavior of the three methods as the number of unavailable domains due to disruptions increases from 2 to 6. In terms of offloaded data, the proposed solution consistently outperforms the baselines in all disruption scenarios. For example, when 3 domains are unavailable, it enables up to 2.41 GB more offloaded data than the random approach, while with 6 unavailable domains it enables up to 4.24 GB more offloaded data than the deterministic approach. With regard to missed video acquisitions, the proposed method reduces the number of missed domains by up to 9 compared with the random approach when 2 domains are unavailable, and by up to 32 compared with the deterministic approach when 6 domains are unavailable. Overall, the method achieves a substantially higher processing service availability, with improvements of up to 28.61% over the random approach and up to 75.61% over the deterministic one.

Figure 2.5 presents the same analysis under SLA requirements focused on trustworthiness. The differences in offloaded data remain comparable, with gains of up to 3.51 GB over the deterministic approach when 6 domains are unavailable and up to 0.93 GB over the random approach when 2 domains are unavailable. In addition, the number of missed video acquisitions is reduced by up to 6 domains compared with the random approach when 2 domains are unavailable, and by up to 27 domains compared with the deterministic approach when 6 domains are unavailable. Finally, the proposed solution provides higher in-network processing availability, with improvements of up to 32.38% compared with the random approach and up to 68.78% compared with the deterministic one.

The larger amount of offloaded data, the lower number of missed video acquisitions, and the higher overall service availability confirm the effectiveness of the proposed solution as a suitable strategy to preserve in-network processing during natural disaster events.

2.3 Optimizing KVI through DT-aided service orchestration mechanism

The second specific contribution [69] focuses on a model for orchestrating B5G services while ensuring compliance with established SLAs. The model investigates and identifies the most suitable resources of network domains to manage service requests. In particular, by exploiting the NDT structure, enhanced with intelligence through a many-to-many matching game algorithm, it supports informed decisions that jointly account for both KPIs and KVIs. In this way, the approach minimizes violations in terms of the number of requests that do not satisfy latency, throughput, cost, and security constraints. As already discussed, the service provisioning orchestration problem is formulated as a many-to-many matching game between service requests, mapped into intents, and network domains, whose resources and capabilities are exposed through their NDT representation. The preferences of the players are derived through the TOPSIS decision-making algorithm, which relies on the Entropy-Weighting Method (EWM) to compute the associated weights.

2.3.1 The Reference Architecture and System Model Definition

This section presents the reference architecture and defines the adopted system model. Fig. 2.6 depicts the layered reference architecture inspired by Network 2030 [117], which builds on the IBN paradigm. The considered environment includes consumers, namely users or clients requesting services, and providers, namely network domains delivering such services or content. At the top layer, consumers express service requests through applications, including demands for specific network services or functionalities. The second layer, referred to as the intent layer, plays a central role in enabling intent-driven network management within IBN, thus serving as a key enabler of network automation. Rather than dealing with low-level configurations, users can express their desired outcomes as high-level objectives

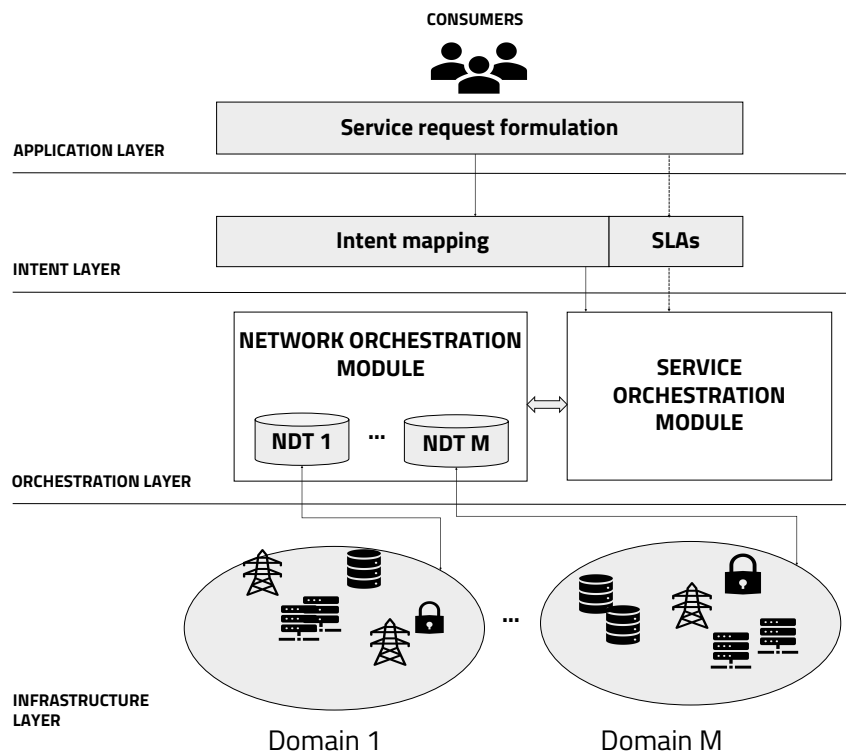


Figure 2.6: The proposed architecture inspired by the framework of IBN for Network 2030 [117].

in a form understandable by the network, which are then transparently translated into concrete configurations and policies. By offering this mechanism, the intent layer maps service requests into intents, thereby increasing agility and ensuring consistency with the intended objectives [119]. Through intent mapping, both the consumer and the service provider can define the level of service expected from the network through SLAs, specifying the quality, availability, and reliability levels that the network must guarantee in order to satisfy user expectations. More specifically, conventional performance-related requirements can be expressed through KPIs, whereas sustainability- and security-related requirements can be described through KVIs. Once high-level service requests are translated into structured and quantifiable intents, they are forwarded to the Orchestration layer. In this layer, the Network Orchestration module and the Service Orchestration module cooperate to determine both how and where these intent requirements should be deployed. More precisely, the former builds NDTs of the domains by analyzing the characteristics and data of the physical Network Infrastructure. In addition, these entities are not merely digital replicas; rather, as dynamic and aggregated representations of domain data, they interact with their physical counterparts to support predictive maintenance and operation. NDTs are exploited to simulate possible service implementations, thus enabling the identification of resource configurations that can be instantiated on demand according to actual resource availability. Conversely, the latter is responsible for determining the most suitable network-domain resources able to satisfy the SLA requirements. The interaction between these two modules allows service providers to simulate multiple resource configurations by leveraging

the abstraction offered by the NDT, and to assess their impact on network performance and behavior before actual deployment in the infrastructure. Finally, the resources of the selected domain that comply with the established SLAs are allocated in the Infrastructure layer in order to serve the request. The details of the overall orchestration procedure are provided in the following sections.

Table 2.2: Main symbols description.

Symbol	Meaning
$\mathbb{I}$	Set of intents.
$\mathbb{D}$	Set of network domains.
N	Number of intent categories.
M	Number of network domains.
$\mathbb{R}^{(n)}$	Set of consumer requests mapped to the n -th intent category.
$\mathbb{T}^{(n)}$	Set of tasks composing a service request.
K	Number of tasks composing a service.
$\tau^{(n)}$	Minimum requested throughput for the $R_s^{(n)}$ -th service request.
$\gamma^{(n)}$	Deadline of the $R_s^{(n)}$ -th service request.
$\Delta^{(n)}$	Impact of a security attack.
$\beta^{(n)}$	Consumer budget for the $R_s^{(n)}$ -th request.
$B^{(n)}$	Input size of the $R_s^{(n)}$ -th service request.
$\theta^{(n)}$	Tolerable risk associated with the $R_s^{(n)}$ -th service request.
$\Omega^{(m)}$	Processing capability of the m -th domain.
$L^{(m)}$	Likelihood of attacks in the m -th domain.
$\sigma^{(m)}$	Cost to pay for resources provided by the m -th domain.
$C_{lf}^{(m)}$	Life cycle cost in the m -th domain.
$C_u^{(m)}$	Usage cost in the m -th domain.
$C_{sec}^{(m)}$	Security cost in the m -th domain.
$q^{(m)}$	Maximum quota of tasks that the m -th domain can handle.
$Bw^{(m)}$	Minimum guaranteed bandwidth for a request by the m -th domain.
$t_{(R_s^{(n)}, D_m)}^{(tran)}$	Transmission delay for the $R_s^{(n)}$ -th request in the m -th domain.
$t_{(R_s^{(n)}, D_m)}^{(proc)}$	Processing delay for the $R_s^{(n)}$ -th request in the m -th domain.
$U_{(R_s^{(n)}, D_m)}$	Transmission rate between the consumer and the m -th domain.
$\rho_{(R_s^{(n)}, D_m)}$	Cyber risk associated with the $R_s^{(n)}$ -th request in the m -th domain.
$Rev^{(m)}$	Total revenue of the m -th domain.
$x_{(T_k(R_s^{(n)}), D_m)}$	Binary indicator variable indicating whether task T_k of $R_s^{(n)}$ is allocated to D_m .
$\mathcal{F}$	Set of tasks with unmet deadline requirements.
$\mathcal{E}$	Set of tasks with unmet throughput requirements.
$\mathcal{C}$	Set of tasks overshooting the user's budget.
$\mathcal{S}$	Set of tasks violating cyber-risk requirements.

Consumers formulate service requests containing specific network-service requirements. The adopted IBN paradigm, supported by appropriate translation mechanisms [119], makes it possible to map those requests into intent categories, each representing a structured definition of service requirements. For clarity, let $\mathbb{I}$ denote the set of N intent categories, such that $\mathbb{I} = \{I_1, I_2, \dots, I_n, \dots, I_N\}$. It is worth observing that multiple service requests may be mapped onto the same n -th intent category. Accordingly, let $\mathbb{R}^{(n)}$ represent the set of all consumer requests mapped to the n -th intent category, i.e., $\mathbb{R}^{(n)} = \{R_1^{(n)}, R_2^{(n)}, \dots, R_s^{(n)}, \dots, R_S^{(n)}\}$.

Hence, the global set of service requests in the system is defined as $\mathbb{R} = \bigcup_{n=1}^N \mathbb{R}^{(n)}$. In general, when a consumer issues a service request $R_s^{(n)} \in \mathbb{R}^{(n)}$, it specifies the following set of parameters: $\langle T^{(n)}, \gamma^{(n)}, \tau^{(n)}, \Delta^{(n)}, \theta^{(n)}, \beta^{(n)}, B^{(n)} \rangle$. Specifically,

- $\mathbb{T}^{(n)}$ denotes the set of K tasks into which the requested service is decomposed. Specifically, $\mathbb{T}^{(n)} = \{T_1^{(n)}, T_2^{(n)}, \dots, T_k^{(n)}, \dots, T_K^{(n)}\}$. Decomposing the service into simpler tasks enables a more systematic and manageable service-delivery process, allowing the network domain to improve efficiency, reliability, and performance throughout the provisioning procedure.
- $\gamma^{(n)}$ denotes the deadline associated with the $R_s^{(n)}$ -th service request, expressed in [s]. Therefore, completing the service within the assigned time constraint represents a conventional performance requirement.
- $\tau^{(n)}$ identifies the minimum throughput required by the consumer for the $R_s^{(n)}$ -th service request.
- $\Delta^{(n)}$ represents the impact of a security attack and captures the effects or consequences caused by an attack resulting from the provisioning of a malicious service. It is a relevant value closely related to the SDGs promoted by the UN, especially with regard to privacy, confidentiality, and trustworthiness [9].
- $\theta^{(n)}$ denotes the acceptable risk associated with the requested service, commonly referred to as risk appetite. In essence, it reflects the consumer's willingness to accept risk.
- $\beta^{(n)}$ is the budget, namely the amount of funds allocated by the consumer for the usage and storage of providers' resources. This parameter reflects a key value related to the UN SDGs on economic growth and sustainability [63].
- $B^{(n)}$ denotes the input data size that must be processed by a service provider, measured in [bits].

On the other hand, let $\mathbb{D}$ be the set of M network domains, namely $\mathbb{D} = \{D_1, D_2, \dots, D_m, \dots, D_M\}$. Each domain $D_m \in \mathbb{D}$ offers storage and computational resources and is characterized by the following tuple of parameters: $\langle \Omega^{(m)}, q^{(m)}, L^{(m)}, \sigma^{(m)}, Bw^{(m)} \rangle$, defined as follows:

- $\Omega^{(m)}$ denotes the processing capability of the m -th domain, expressed in [cycles/s].
- $q^{(m)}$ denotes the maximum number of tasks that the m -th domain can handle simultaneously. This parameter accounts for the computational cores available in the domain, each characterized by a capability equal to $\Omega^{(m)}$.
- $L^{(m)}$ denotes the likelihood of attacks against the security of the m -th domain. It represents the probability that a threat successfully exploits a vulnerability and causes damage within the m -th domain.

- $\sigma^{(m)}$ is described through the tuple $(C_u^{(m)}, C_{lf}^{(m)}, C_{sec}^{(m)})$. In particular, $C_u^{(m)}$ denotes the usage cost that consumers must pay to exploit the resources offered by the m -th network domain, measured in [\$/s]. By contrast, $C_{lf}^{(m)}$ and $C_{sec}^{(m)}$ represent the life-cycle cost and security cost, respectively, both measured in [\$], that the domain must sustain to keep resources operational and security measures effective.
- $Bw^{(m)}$ denotes the minimum guaranteed bandwidth value that the domain can provide to a service request.

Table 2.2 summarizes the main symbols adopted to describe the environment and their meanings.

2.3.2 Parameters Computation

- Delay Computation

The overall end-to-end delay, defined as the time elapsed from the instant a service request is issued to the instant the service is delivered to the consumer, is modeled as the linear combination of two contributions. The first one, denoted by $t_{(R_s^{(n)}, D_m)}^{(tran)}$, represents the transmission delay and captures the time required for the $R_s^{(n)}$ -th service request to reach the m -th network domain. To compute the transmission delay over the communication channel, the transmission rate between the consumer and the provider domain must first be determined. Such rate, associated with the $R_s^{(n)}$ -th request directed to the m -th network domain and denoted by $U_{(R_s^{(n)}, D_m)}$, is computed according to the Shannon formula [120] as follows:

$$U_{(R_s^{(n)}, D_m)} = Bw^{(m)} \cdot \log_2 \left(1 + \frac{P_{tx} \cdot H^{(m)}}{P_{noise}} \right), \quad (2.1)$$

where P_{tx} denotes the transmission power of the consumer requesting the service, $H^{(m)}$ denotes the transmission channel gain, and P_{noise} denotes the noise power over the communication channel. Given the input data size $B^{(n)}$ to be processed for the requested service and the previously derived transmission rate, the transmission delay can be evaluated as shown in eq.2.2.

$$t_{(R_s^{(n)}, D_m)}^{(tran)} = \frac{B^{(n)}}{U_{(R_s^{(n)}, D_m)}}. \quad (2.2)$$

The second contribution, denoted by $t_{(R_s^{(n)}, D_m)}^{(proc)}$, represents instead the processing delay, namely the time required by the m -th provider domain to execute and complete the $R_s^{(n)}$ -th requested service. It can be computed as reported in eq.2.3.

$$t_{(R_s^{(n)}, D_m)}^{(proc)} = \frac{\Phi \cdot B^{(n)}}{\Omega^{(m)}}, \quad (2.3)$$

where Φ denotes the number of CPU cycles required to process a single bit and, according to [67], it is set equal to 1000 cycles per bit.

By combining the previous terms, the total end-to-end delay experienced for serving the $R_s^{(n)}$ -th request at the m -th domain is obtained as in eq. 2.4.

$$t_{(R_s^{(n)}, D_m)} = t_{(R_s^{(n)}, D_m)}^{(tran)} + t_{(R_s^{(n)}, D_m)}^{(proc)}. \quad (2.4)$$

- Cyber Risk Computation

Designed to interconnect a massive number of devices and to integrate technologies such as SDN, network slicing, and edge computing, B5G networks support critical infrastructures and services, including healthcare (e.g., remote surgery), transportation (e.g., autonomous vehicles), energy grids, and public safety. They also promise ultra-high speed and low latency, which are essential for real-time applications. However, the integration of multiple technologies increases the number of potential entry points for cyber attackers, thereby introducing new vulnerabilities and making network security more complex. Each component, such as virtualized network functions or edge devices, may become a potential target of cyber threats. A successful cyber-attack may severely impair network performance, with potentially catastrophic consequences for critical infrastructures and services, including loss of life, major economic disruption, and threats to national security.

Within this context, service providers must also take into account non-negligible risk factors when accepting service requests. Such risks may originate from natural disasters or from human-driven events, including human error, cybercrime (e.g., extortion or fraud), cyberwar, and cyberterrorism [121]. In general, cyber risk affects the confidentiality, availability, or integrity of data and services. Overall, it can be interpreted as a function of three elements: (i) vulnerability, (ii) threat, and (iii) impact [122]. A vulnerability is a weakness in the design, implementation, operation, or management of an asset, and therefore depends on the security level of the network domains. A threat, in turn, is the potential exploitation of a vulnerability. These two factors are often aggregated into the parameter L , which represents the likelihood of a successful attack [123]. The impact of cyber risk, instead, denotes the consequences that security threats and vulnerabilities may generate for individuals, organizations, and systems. It includes reputational, economic, and legal damages that consumers may suffer when they become victims of a threat.

Estimating the security-related parameter affecting network domains is a challenging task. The literature provides several heterogeneous risk-assessment approaches for this purpose. Without loss of generality, this work adopts the model based on Attack Countermeasure Tree (ACT) [124] as the reference framework for likelihood modeling. This model allows attack paths or vulnerabilities to be identified, as well as proper countermeasures to be selected in order to detect and mitigate them. Consequently, the security posture of a domain can be evaluated through the probability of attack success. This is modeled through a “k-out-of-n” probability, where A_S denotes the number of successful attack events out of the total A_{DM} detected and mitigated attacks. The corresponding expression of the attack-success likelihood for the m -th network domain is reported in eq.2.5.

$$L^{(m)} = \sum_{j=A_S}^{A_{DM}} \binom{A_S}{j} Pr_{Atk}^j (1 - Pr_{Atk})^{A_S-j} \quad (2.5)$$

where Pr_{Atk} denotes the probability of an attack event. Hence, for the $R_s^{(n)}$ -th service request processed by the m -th domain, the associated Cyber Risk $\rho_{(R_s^{(n)}, D_m)}$ is defined as follows:

$$\rho_{(R_s^{(n)}, D_m)} = \Delta^{(n)} \cdot L^{(m)}. \quad (2.6)$$

The proposed system model aims to improve orchestration strategies by going beyond traditional KPI-based service-provider selection. To this end, it incorporates the UN SDGs, monitored through KVis, in order to address next-generation network objectives. Although B5G technologies support these goals, they also introduce new vulnerabilities, which makes cyber risk a crucial KVI for preserving B5G security.

- Cost Computation

The service provisioning process also generates costs. In particular, domain providers must manage the complete life cycle of resources and invest in suitable security countermeasures. In this work, $C_{lf}^{(m)}$ denotes the cost associated with the life cycle of the managed resources, and it is expressed as $C_{lf}^{(m)} = C_{act}^{(m)} + (C_{depl}^{(m)} \cdot q^{(m)})$, where $C_{act}^{(m)}$ is the service-activation cost and $C_{depl}^{(m)}$ denotes the deployment and maintenance cost of the physical network. On the other hand, $C_{sec}^{(m)}$ denotes the cost associated with auditing, vulnerability assessment, threat mitigation, the deployment of security tools, and regular updates and patches within the m -th network domain [125]. These costs help maintain the domain secure and compliant with relevant regulations, thus reducing its exposure to cyber attacks. At the same time, consumers are charged according to the duration for which they exploit domain resources. The parameter $C_{ut}^{(m)}$ denotes the cost associated with the use of processing and storage resources. This cost is typically determined by the amount of time for which resources are allocated to a consumer service request, ensuring a proportional charging policy based on actual resource consumption. Overall, the total cost structure of service provisioning includes life-cycle costs ($C_{lf}^{(m)}$), security costs ($C_{sec}^{(m)}$), and usage costs ($C_{ut}^{(m)}$). These cost components are essential to preserve the infrastructure and to guarantee secure and reliable service delivery to consumers. By considering the costs incurred for maintaining network resources throughout their life cycle and for protecting the infrastructure, together with the revenues generated by consumers, the total revenue of the m -th domain, denoted as $Rev^{(m)}$, can be computed as reported in Eq. 2.7.

$$Rev^{(m)} = C_{ut}^{(m)} \cdot t_{(R_s^{(n)}, D_m)}^{(proc)} - (C_{lf}^{(m)} + C_{sec}^{(m)} \cdot L^{(m)}), \quad (2.7)$$

where the resource-usage cost and the security costs are weighted by the resource-utilization time and by the likelihood that cyber attacks succeed in the m -th network domain, respectively. In this way, the revenue computation properly reflects the economic impact of both operational costs and potential security risks.

2.3.3 Problem Formulation

The primary objective of this study is to design a service orchestration strategy able to select available network domains that can satisfy consumer service requests

while minimizing inconsistencies caused by SLA violations. More specifically, the proposed methodology aims to identify resource configurations that jointly achieve the following objectives:

- Minimize the number of tasks whose completion time exceeds the assigned deadline.
- Minimize the number of tasks that do not satisfy the required throughput threshold.
- Minimize the deployment costs that exceed the consumer's budget.
- Minimize the task assignments whose associated cyber risk is higher than the consumer's risk appetite.

Because consumers and their corresponding service requests are inherently heterogeneous, intent mapping represents a crucial initial step for organizing and orchestrating requests effectively. Depending on the type of business or service, intent mapping can automate the categorization process by analyzing the content of consumer requests. This mechanism facilitates the identification of the most suitable network domain for processing and executing tasks in an efficient manner. In addition, the proposed architecture enables the decomposition of requested services into smaller and more manageable tasks that can be executed across different network domains. Ideally, each task corresponds to a specific function or step within the service delivery process. This design improves agility and enhances the overall quality of service delivery. In line with this principle, the proposed service orchestration strategy deploys tasks belonging to different services by identifying a suitable and distributed combination of network domains. Such a combination jointly addresses completion time, throughput, cost, and cyber risk requirements.

Let $x_{(T_k(R_s^{(n)}), D_m)}$ denote a binary indicator variable that specifies whether task T_k associated with service request $R_s^{(n)}$ is allocated to domain D_m :

$$x_{(T_k(R_s^{(n)}), D_m)} = \begin{cases} 1 & \text{if } T_k \text{ for } R_s^{(n)} \text{ is assigned to } D_m, \\ 0 & \text{otherwise.} \end{cases} \quad (2.8)$$

where $n \in [1, N]$ identifies the n -th requested intent category, whereas $m \in [1, M]$ identifies the m -th network domain.

Since each requested service mapped to an intent category is associated with a deadline, it is necessary to define the set $\mathcal{F}$ of tasks whose completion time requirements are not met. Assuming that each task composing service request $R_s^{(n)}$ can be executed simultaneously by the selected network domain, the resulting service time $t_{(R_s^{(n)}, D_m)}$ corresponds to the maximum end-to-end delay among all provider domains involved in executing the assigned tasks. Accordingly, $\mathcal{F}$ can be expressed as follows:

$$\mathcal{F} = \{T_k^{(n)} \in \mathbb{T}^{(n)} \mid x_{(T_k(R_s^{(n)}), D_m)} \cdot t_{(R_s^{(n)}, D_m)}^{(n)} > \gamma^{(n)}; \quad (2.9)$$

$$\forall R_s^{(n)} \in \mathbb{R}^{(n)}, k \in [1, K], n \in [1, N], m \in [1, M]\}.$$

Throughput represents another SLA metric for a requested service, since it refers to the rate at which tasks are processed or completed within a given time interval. In this context, let $\mathcal{E}$ denote the set of tasks whose execution does not guarantee fulfillment of the throughput requirement. Its expression is given as follows:

$$\mathcal{E} = \{T_k^{(n)} \in \mathbb{T}^{(n)} \mid x_{(T_k(R_s^{(n)}), D_m)} \cdot U_{(R_s^{(n)}, D_m)} < \tau^{(n)}; \quad (2.10)$$

$$\forall R_s^{(n)} \in \mathbb{R}^{(n)}, k \in [1, K], n \in [1, N], m \in [1, M]\}.$$

Let $\mathcal{C}$ also denote the set of service tasks whose execution entails costs exceeding the user's budget. Considering that tasks may be distributed across different domains, the processing time $t_{(R_s^{(n)}, D_m)}^{(proc)}$ must account for the maximum processing time among the provider domains involved in handling the assigned tasks. Its expression is reported below:

$$\mathcal{C} = \{T_k^{(n)} \in \mathbb{T}^{(n)} \mid x_{(T_k(R_s^{(n)}), D_m)} \cdot C_{ut}^{(m)} \cdot t_{(R_s^{(n)}, D_m)}^{(proc)} > \beta^{(n)}; \quad (2.11)$$

$$\forall R_s^{(n)} \in \mathbb{R}^{(n)}, k \in [1, K], n \in [1, N], m \in [1, M]\}.$$

Furthermore, if the cyber risk $\rho_{(R_s^{(n)}, D_m)}$ associated with service request $R_s^{(n)}$ handled by domain D_m exceeds the acceptable risk level $\theta^{(n)}$, then a potential misalignment exists between the risk tolerance of the involved parties. As a consequence, cybersecurity requirements are not properly satisfied. Based on this observation, let $\mathcal{S}$ denote the set of service tasks assigned to domains that violate their cyber risk constraints, as defined below:

$$\mathcal{S} = \{T_k^{(n)} \in \mathbb{T}^{(n)} \mid x_{(T_k(R_s^{(n)}), D_m)} \cdot \rho_{(R_s^{(n)}, D_m)} > \theta^{(n)}; \quad (2.12)$$

$$\forall R_s^{(n)} \in \mathbb{R}^{(n)}, k \in [1, K], n \in [1, N], m \in [1, M]\}.$$

Therefore, in order to identify the optimal task-assignment decision within the service orchestration procedure, addressing SLAs violations is essential. This objective can be achieved by jointly minimizing the cardinality of the previously defined sets $\mathcal{F}, \mathcal{E}, \mathcal{C}, \mathcal{S}$, thereby reducing the number of tasks that do not comply with the service requirements. Accordingly, the following multi-objective optimization problem is formulated and reported in eq.2.13:

$$\min_{x_{(T_k(R_s^{(n)}), D_m)}} |\mathcal{F}| + |\mathcal{E}| + |\mathcal{C}| + |\mathcal{S}| \quad (2.13)$$

$$\text{s.t.} \quad \sum_{n=1}^N \sum_{s=1}^S \sum_{k=1}^K x_{(T_k(R_s^{(n)}), D_m)} \leq q^{(m)}, \forall m \in [1, M] \quad (2.14)$$

$$\sum_{m=1}^M \sum_{k=1}^K x_{(T_k(R_s^{(n)}), D_m)} = K, \quad (2.15)$$

$$\forall R_s^{(n)} \in \mathbb{R}^{(n)}, \forall n \in [1, N]$$

$$\sum_{m=1}^M \sum_{s=1}^S \sum_{k=1}^K x_{(T_k(R_s^{(n)}), D_m)} \leq \sum_{m=1}^M q^{(m)} \quad (2.16)$$

$$x_{(T_k(R_s^{(n)}), D_m)} \in \{0, 1\}. \quad (2.17)$$

The proposed multi-objective problem is subject to the constraints expressed in eqs. (2.14)–(2.16). Constraint (2.14) states that the m -th network domain can process at most $q^{(m)}$ tasks simultaneously. Constraint (2.15) imposes that all K tasks composing a requested service must be assigned to a network domain. Finally, constraint (2.16) ensures that the total number of tasks associated with the entire set of requested services does not exceed the global capacity of all network domains. Therefore, all requested services mapped into intents must be assigned and processed by at most one network domain.

The formulation reported in eqs. (2.13)–(2.17) is a combinatorial problem and has been proven NP-hard [126].

2.3.4 The Proposed Solution

In this section, a service orchestration procedure is developed by exploiting a multi-criteria decision-making algorithm within a stable matching game framework. This approach adopts a heuristic method to address the NP-hard problem introduced in Section 2.3.3.

The conceived procedure builds upon matching theory, which enables the design of flexible solutions, especially matching games, for combinatorial problems [127], [128], [129]. In general, a matching game models the interaction between elements belonging to two distinct sets, commonly referred to as players, and returns a matching function that establishes meaningful relations among them. These relations depend on the preferences expressed by each player of one set toward the players of the opposite set, thus indicating the degree of satisfaction associated with a given match according to suitable evaluation criteria [129]. Under these assumptions, the players in the proposed framework are represented by the set of network domains and the set of service requests mapped into intent categories. Their preferences are evaluated through the TOPSIS multi-criteria decision-making algorithm [130]. By defining players and computing preferences in this way, it becomes possible to match intents with domains able to process them in a timely, efficient, convenient, and secure manner. At the same time, domains are matched with intents

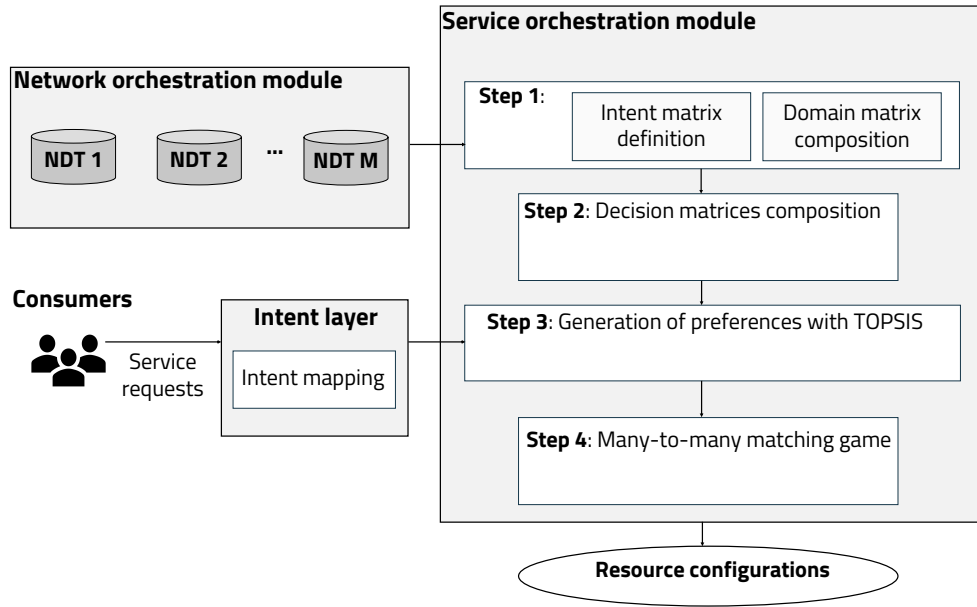


Figure 2.7: Overall orchestration procedure [69].

that provide higher revenues.

The remainder of this section illustrates the methodology followed by the proposed procedure, schematically shown in Fig. 2.7 and fully deployable through the modules of the reference architecture described in Section 2.3.1. More specifically, starting from the NDT representations and the definition of intent categories, the Service Orchestration module constructs the domain and intent matrices. Based on this information, a decision matrix or decision vector is built for each player of the matching game, namely consumers' intents and domains. The values contained in these decision structures, derived from the outcomes of the NDTs, effectively capture the resource configurations that a domain must expose in order to fulfill a given intent. This matrix-based modeling reflects the ability of NDTs to self-simulate their capabilities, support analysis and prediction, and provide decision support to the Service Orchestration module. Once the weights are computed, the TOPSIS multi-criteria decision-making algorithm derives the preference lists for each player from the corresponding decision matrices. The resulting preferences are then exploited in the many-to-many matching game to identify beneficial network resource configurations.

Step 1 - Domains and intent matrix composition

By abstracting network domains, NDTs enable effective monitoring and analysis of complex network systems. This approach supports the construction of detailed behavioral models able to capture performance metrics such as latency and throughput. The proposed work exploits the outputs of the NDTs as the first step of the service orchestration procedure and represents such outputs through a matrix formulation. Specifically, $\mathbf{D}$ denotes the domain matrix, with dimensions $[M \times V]$, where M is the number of considered network domains and V is the set of features captured through the NDTs abstraction. These features include processing capability, the quota of assignable tasks, the likelihood of security-related attacks, cost, and the guaranteed bandwidth of each domain, as detailed in Section 2.3.1. The entries

of the domain matrix, denoted by $d_{m,v}$ with $m \in [1, M]$ and $v \in [1, V]$, represent the value associated with the v -th feature of the m -th domain. Since the parameters captured by the NDTs may vary during the day, the corresponding entries of the domain matrix are updated accordingly. The matrix is defined as follows:

$$\mathbf{D} = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,V} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,V} \\ \vdots & \vdots & \ddots & \vdots \\ d_{M,1} & d_{M,2} & \cdots & d_{M,V} \end{bmatrix}_{M \times V} \quad (2.18)$$

In addition, the proposed procedure requires the definition of intent categories according to the service requirement levels that a consumer may express. These categories are collected in an intent matrix, denoted by $\mathbf{I}$, whose dimensions are $[N \times Z]$ and whose entries are $i_{n,z}$, with $n \in [1, N]$ and $z \in [1, Z]$. Here, N identifies the considered intent categories, whereas Z denotes the set of parameters specified for each service request, as discussed in Section 2.3.1. More precisely, these parameters include the task set, deadline, minimum throughput, impact, risk appetite, budget, and input data size to be processed. The intent matrix is expressed as follows:

$$\mathbf{I} = \begin{bmatrix} i_{1,1} & i_{1,2} & \cdots & i_{1,Z} \\ i_{2,1} & i_{2,2} & \cdots & i_{2,Z} \\ \vdots & \vdots & \ddots & \vdots \\ i_{N,1} & i_{N,2} & \cdots & i_{N,Z} \end{bmatrix}_{N \times Z} \quad (2.19)$$

Step 2 - Decision matrices formulation

The second step consists in constructing the decision matrices, which serve as the input of the subsequent multi-criteria decision-making phase responsible for deriving preferences. Starting from the values contained in matrix $\mathbf{D}$, a domain decision matrix is generated for each of the N intent categories. The purpose of this operation is to capture the state of each domain when serving a request mapped to the n -th intent category. Accordingly, a generic domain decision matrix is denoted by $\mathbf{D}^{(n)}$, with $n \in [1, N]$, and has dimensions $[M \times f]$. Here, M denotes the number of network domains, whereas f corresponds to the set of system parameters considered in the optimization problem formulated in Section 2.3.3, namely completion time, throughput, cost, and cyber risk. A domain decision matrix is represented as follows:

$$\mathbf{D}^{(n)} = \begin{bmatrix} d^{(n)}_{1,1} & d^{(n)}_{1,2} & \cdots & d^{(n)}_{1,4} \\ d^{(n)}_{2,1} & d^{(n)}_{2,2} & \cdots & d^{(n)}_{2,4} \\ \vdots & \vdots & \ddots & \vdots \\ d^{(n)}_{M,1} & d^{(n)}_{M,2} & \cdots & d^{(n)}_{M,4} \end{bmatrix}_{M \times 4} \quad (2.20)$$

Similarly, starting from the values contained in matrix $\mathbf{I}$, an intent decision

vector is built for each of the M domains. This vector is denoted by $\mathbf{I}^{(m)}$, where $m \in [1, M]$, and has dimensions $[N \times 1]$, with $n \in [1, N]$. These vectors represent the revenues obtained from deploying, using, and storing domain resources for each requested intent. Differently from the domain decision matrices, for a given domain the decision to accept a request depends only on the potential revenue generated by allocating and processing the consumer request. As detailed in Section 2.3.1, this quantity captures not only cost aspects but also the potential security risks related to service deployment. An intent decision vector is expressed as follows:

$$\mathbf{I}^{(m)} = \begin{bmatrix} i_{1,1}^{(m)} \\ i_{2,1}^{(m)} \\ \vdots \\ i_{N,1}^{(m)} \end{bmatrix}_{N \times 1} \quad (2.21)$$

Step 3 - Preference lists generation

The third step of the procedure consists in computing the preferences, which serve as the input to the matching game. These preferences quantify the suitability and satisfaction associated with assigning a domain to an intent, and vice versa. To this end, the proposed strategy employs the TOPSIS multi-criteria decision-making algorithm to derive the preference lists by exploiting the features contained in the domain decision matrices and in the intent decision vectors as decision criteria. Within this framework, all the criteria appearing in the domain decision matrices are treated as cost criteria, with the only exception of throughput, which is modeled as a benefit criterion. This choice is consistent with the revenue criterion described by the intent decision vectors, which is also treated as a benefit.

According to TOPSIS, each criterion of every decision matrix must be associated with a weight measuring its importance in the preference-generation process. In this work, the EWM [131] is adopted to evaluate the relevance of each criterion. Specifically, the EWM computes the entropy of each criterion in every column of the decision matrices and assigns higher weights to criteria characterized by greater variability, namely lower entropy. For each domain decision matrix, the resulting weight vector is denoted by $\mathbf{W}^{(n)}$:

$$\mathbf{W}^{(n)} = [w_1^{(n)}, w_2^{(n)}, w_3^{(n)}, w_4^{(n)}]. \quad (2.22)$$

where $\sum_{f=1}^4 w_f^{(n)} = 1$ and $0 \leq w_f^{(n)} \leq 1$.

Once a weight vector has been associated with each decision matrix, TOPSIS proceeds with the derivation of the preference lists. In particular, the adopted multi-criteria decision-making algorithm first normalizes the decision matrices and then multiplies each entry by the corresponding weight obtained through the EWM, as follows:

$$\hat{d}_{m,f}^{(n)} = w_f^{(n)} \cdot \frac{d_{m,f}^{(n)}}{\max_{m \in M} (d_{m,f}^{(n)})} \quad (2.23)$$

Subsequently, TOPSIS identifies the positive ideal solution (Positive Ideal Solution (PIS)), which includes the maximum value for benefit criteria and the minimum value for cost criteria. Likewise, it identifies the negative ideal solution (Negative Ideal Solution (NIS)), which includes the minimum value for benefit criteria and the maximum value for cost criteria. These are defined as follows:

$$\text{PIS}^+ = \left\{ \hat{d}_{m,1}^{(n)+}, \dots, \hat{d}_{m,4}^{(n)+} \right\} \quad (2.24)$$

$$\text{NIS}^- = \left\{ \hat{d}_{m,1}^{(n)-}, \dots, \hat{d}_{m,4}^{(n)-} \right\}. \quad (2.25)$$

At this stage, the procedure computes the Euclidean distance between each row of the normalized domain decision matrices and the corresponding values of the PIS and NIS sets. For clarity, the resulting Euclidean distances are denoted by S_m^+ and S_m^- , respectively, and are expressed as follows:

$$S_m^+ = \sqrt{\sum_{f=1}^4 \left(\hat{d}_{m,f}^{(n)} - \hat{d}_{m,f}^{(n)+} \right)^2}, \quad m = 1, \dots, M, \quad (2.26)$$

$$S_m^- = \sqrt{\sum_{f=1}^4 \left(\hat{d}_{m,f}^{(n)} - \hat{d}_{m,f}^{(n)-} \right)^2}, \quad m = 1, \dots, M. \quad (2.27)$$

By exploiting these distances together with the selected criteria, TOPSIS can determine how close each domain is to the ideal solutions when handling a request mapped to a given intent category. This closeness parameter, which corresponds to the preference score, is defined as the RC and is computed for each domain in the domain decision matrix as follows:

$$RC_m = \frac{S_m^-}{S_m^+ + S_m^-}, \quad m = 1, \dots, M \quad (2.28)$$

The preferences of all domains are then ranked in decreasing order according to their RC values.

This procedure is iteratively repeated for every domain decision matrix, in which the rows describe the behavior of each domain with respect to a service request belonging to the n -th intent category. As a result, the proposed orchestration process generates a prioritized list of provider domains for each service request $R_s^{(n)}$, as shown below:

$$R_s^{(n)}_{\text{pref}} = (D_1, D_2, \dots, D_M). \quad (2.29)$$

The ranking in eq.2.29 implies that service request $R_s^{(n)}$ is better served by domain D_1 than by domain D_2 . This preference relation is denoted as $D_1 \succ_{R_s^{(n)}} D_2$.

Similarly, the orchestration procedure is applied to each intent decision vector, thus generating a preference list for every domain D_m . This can be written as follows:

$$D_{m\text{pref}} = (I_1, I_2, \dots, I_N). \quad (2.30)$$

This ordering indicates that domain D_m prefers handling intent I_1 over intent I_2 . Formally, this is expressed by the relation $I_1 \succ_{D_m} I_2$.

Both the domain and the intent preference lists are then used as inputs to the subsequent matching-game phase.

Step 4 - Many-to-many matching game

The final step consists in executing the many-to-many matching game in order to establish associations between consumer service requests and provider domains, thereby enabling the identification of suitable resource configurations. In particular, by considering the set of network domains $\mathbb{D}$ and the set of service requests $\mathbb{R}$ as two disjoint and finite sets of players, the associations between them are determined according to the preferences defined in the previous step. The game returns a function λ defined as follows:

$$\lambda : \mathbb{D} \cup \mathbb{R} \rightarrow \mathbb{D} \cup \mathbb{R}. \quad (2.31)$$

This function represents a valid matching if the following conditions hold:

- *Condition 1:* each domain $D_m \in \mathbb{D}$ is matched with a subset of service requests mapped to the n -th intent category, i.e., $\lambda(D_m) \subseteq \mathbb{R}^{(n)} \in \mathbb{R}$.
- *Condition 2:* each service request $R_s^{(n)} \in \mathbb{R}^{(n)}$ is matched with a subset of domains, i.e., $\lambda(R_s^{(n)}) \subseteq \mathbb{D}$.
- *Condition 3:* each domain $D_m \in \mathbb{D}$ can process at most $q^{(m)}$ tasks composing service requests, i.e., $|\lambda(D_m)| \leq q^{(m)}$.
- *Condition 4:* a domain $D_m \in \mathbb{D}$ is matched with a service request $R_s^{(n)} \in \mathbb{R}$ if and only if the service request $R_s^{(n)}$ is matched with domain D_m , i.e., $R_s^{(n)} \in \lambda(D_m) \iff D_m \in \lambda(R_s^{(n)})$.

The individual preferences generated through TOPSIS are complete, meaning that each player ranks all the players of the opposite set. Dealing with complete preferences is necessary because constraint (2.16) of the formulated optimization problem requires that all tasks composing service requests be assigned to at most

one network domain, even in cases where individual domains may not fully satisfy the service requirements. In addition, preferences are strict, meaning that each player establishes a strict ordering over the players of the opposite set.

Algorithm 5 Many-to-many matching algorithm

Input: $R_s^{(n)}_{\text{pref}} \forall R_s^{(n)} \in \mathbb{R}^{(n)}, D_m_{\text{pref}} \forall D_m \in \mathbb{D}$
Output: $\lambda : \mathbb{D} \cup \mathbb{R} \rightarrow \mathbb{D} \cup \mathbb{R}$

```

1: while  $\exists R_s^{(n)} \in \mathbb{R}^{(n)}$  not completely assigned do
2:   Send proposal to its most preferred  $D_m$ 
3:   if  $D_m$  busy cores  $< q^{(m)}$  then
4:     if  $D_m$  can accept the entire request then
5:       Assign  $R_s^{(n)}$  to  $D_m$ 
6:     else
7:       Assign  $T_k^{(n)}$  of  $R_s^{(n)}$  to  $D_m$ 
8:     end if
9:     Update  $D_m$  busy cores and  $R_s^{(n)}$  to be assigned
10:  else
11:    Find  $R_{s'}^{(n)}$  in  $\lambda(D_m) \mid R_s^{(n)} \succ_{D_m} R_{s'}^{(n)}$ 
12:    if  $\exists R_{s'}^{(n)}$  then
13:      Remove  $R_{s'}^{(n)}$  from  $\lambda(D_m)$ 
14:      Remove  $D_m$  from  $\lambda(R_{s'}^{(n)})$ 
15:      if  $D_m$  can accept the entire request then
16:        Assign  $R_s^{(n)}$  to  $D_m$ 
17:      else
18:        Assign  $T_k^{(n)}$  of  $R_s^{(n)}$  to  $D_m$ 
19:      end if
20:      Update  $D_m$  busy cores and  $R_s^{(n)}$  to be assigned
21:    else
22:      Not assign  $R_s^{(n)}$  to  $D_m$ 
23:    end if
24:  end if
25:  Delete  $D_m$  from  $R_s^{(n)}_{\text{pref}}$ 
26: end while
  
```

To enable suitable resource configurations, the matching game must converge to a stable outcome. For this purpose, λ must be individually rational, meaning that no player prefers remaining unmatched rather than being matched with another player. Moreover, λ must be pairwise stable, meaning that no blocking pairs exist in the game.

Accordingly, the proposed stable many-to-many matching algorithm is inspired by the well-known DAA [132]. Its details are provided in the pseudocode reported in *Algorithm 5*. The ultimate objective of the designed game is not only to provide a stable matching between service requests and provider domains, but also to explore the possibility of distributing the tasks of the same request across different providers in order to achieve better performance. In *steps 1–2*, each unassigned service request

sends a proposal to its most preferred domain. Every domain receiving at least one proposal performs the following operations:

- If not all of its computational cores are currently occupied, the domain evaluates whether it can process the whole $R_s^{(n)}$ -th service request or only a subset of its tasks, and accepts the request accordingly (steps 3–9).
- If it has reached its quota and it has previously accepted a request that is less preferred than the newly received one (e.g., $R_{s'}^{(n)}$), then the game replaces the less preferred request with the most preferred one (steps 11–22). Otherwise, $R_s^{(n)}$ is not assigned to its D_m -th preferred provider domain.

After the matching procedure is completed (steps 1–24), the first preference is removed from the list $R_s^{(n)}_{\text{pref}}$. The algorithm then iterates until all service requests have been completely matched.

Designing the strategy in this way ensures that service requests are matched with domains according to their ordered preference lists, thus guaranteeing associations that are beneficial to them and yielding an individually rational matching. In the same way, domains accept beneficial proposals according to their own preferences. Moreover, the matching λ prevents the emergence of blocking pairs. Since all preferences are strict, the proposed algorithm returns a matching that is beneficial to the players initiating the procedure and sending proposals first. Therefore, the whole procedure can be regarded as service-request-optimal, meaning that the resulting matching is the best possible stable outcome for all service requests under the considered constraints. Consequently, the outcome of the matching game identifies the combination of domains that provides the most suitable resource configuration by exploiting preferences that capture the service requirements expressed by consumers.

2.3.5 Simulation Results

This section outlines the environmental setup and reviews the results from the simulation campaigns obtained using a MATLAB script. These campaigns evaluate the effectiveness of the proposed service orchestration model in deploying B5G service requests mapped to intents, with the aim of minimizing SLAs violations.

The investigated environment considers 10 network domains that exhibit varying computing capabilities, likelihood of security attacks, countermeasure investments, resource management costs, and available bandwidth, as detailed in Section 2.3.1. These parameters are uniformly sampled from the specified ranges reported in Table 2.4. Specifically, the processing capabilities of domains are expressed in megacycles per second [megacycles/s] and range from 40 to 2000. The maximum quota of tasks that a domain can handle varies from 750 to 2800. The likelihood of attack success is assumed to be in the range [0, 1]. The costs are defined by assessing the prices for accessing and using cloud resources. In this regard, the usage cost $C_{ut}^{(m)}$ is expressed in dollars per second [\$/s] and ranges from 1 to 2. Meanwhile, the lifecycle cost components $C_{lf}^{(m)}$ and the security countermeasure costs $C_{sec}^{(m)}$ are expressed in dollars [\$] and vary in the ranges [0.001, 0.002] and [0.003,

0.005], respectively. Finally, the transmission rate $U_{(R_s^{(n)}, D_m)}$ between consumers and the provider domain is expressed in megabits per second [Mbps] and is set in the range [100, 1000]. The parameters of the aforementioned network domains are captured by their NDTs and updated every 30 minutes.

The environment also includes an indefinite number of consumers who demand service requests mapped through the designed architecture into categories of intents. The considered number of service requests ranges from 750 to 2500 (i.e., 750, 1000, 1500, 2000, 2500) each 30 minutes. Concurrently, five classes of B5G services have been investigated to define the intent categories. These classes include E-Health, Massive Twinning, Mixed-Reality, Industrial, and Smart Cities use cases and services [133], as shown in Table 2.3. These can be mapped to the three common scenarios defined by ITU-T IMT for 2020 and beyond [23], including eMBB, massive Machine-Type Communications (mMTC), and URLLC. Specifically, the E-Health class aligns with the URLLC usage scenario due to its stringent latency requirements, which prioritize real-time communications. Additionally, it demands a high level of reliability, security, and privacy due to the sensitive nature of the critical data being transmitted. The Massive Twinning class also corresponds to the URLLC usage scenario, given its requirements for extremely high reliability, low latency, and precise synchronization in industrial control, decision-making, and operations adaptation. These characteristics allow the management of critical situations, real-time simulations, and dynamic reconfigurations in industrial environments. The Mixed-Reality class is associated with delivering high-quality, immersive experiences that demand high bandwidth, low latency, and high data rates, which are key characteristics of eMBB. The fully immersive experience and real-time context awareness necessitate the high-capacity data transmission that eMBB aims to provide. This class primarily involves outdoor environments with a high density of users, with safety considerations mainly related to the movement of people in urban scenarios. The Industrial class aligns with both the URLLC and eMBB usage scenarios due to the required precise positioning, synchronization, and the need for reliable, low-latency, and high-data-rate communications. Finally, the Smart Cities class pertains to both the eMBB and mMTC scenarios, requiring high data rates to support the collection, transmission, and processing of data from sensors, alongside robust and resilient communication solutions capable of handling a large number of connections. In general, network security and trustworthiness are crucial due to the volume and nature of the collected data. To quantify the parameters and metrics of the system model, the parameters listed in Table 2.3 have been considered, with values defined based on the requirements of B5G services and the characteristics described above in terms of the time deadline, expressed in seconds [s], throughput, expressed in [Mbps], budget, impact, and risk appetite. Besides, the described B5G service classes consider the number of tasks and the size of the request that can be mapped to each category.

In this context, the consumer budget, expressed in dollars [\$], varies in the range [10,120]. Impact and risk appetite are both represented with values in the range [1,3]. Generally, higher impact indicates more severe consequences caused by an attack, while higher risk appetite reflects a consumer's willingness to accept greater risk. The number of tasks ranges from 5 to 10, and the input size, $B^{(n)}$, expressed in megabits [Mb], has been sampled from the range [0.6,1.2] for each category. All

these specified ranges have been set based on the type of deployment and devices involved in the specific service, as noted in [133]. Therefore, 48 test runs are considered for every simulation, spanning from 00:00 to 23:30 at runtime.

To improve the evaluation of the conceived methodology, four different evaluation scenarios have been considered, namely S1, S2, S3, and S4. According to the most common B5G needs, these correspond to specific use cases encompassing the majority of service requests. These needs typically entail stricter deadlines, higher throughput, lower budgets, and higher cyber risk requirements, respectively. For each scenario, the results are obtained by considering 50 different seeds, accounting for varying distributions of service requests. These results are averaged over a 2.5-hour time window, sliding by half an hour, within an 80% confidence interval.

Scenario S1 involves 80% of articulated service requests with stricter deadlines, associated with E-Health and Mixed-Reality B5G services. Conversely, scenario S2 involves 60% of consumer requests requiring high throughput. Scenario S3 considers 80% of generated service requests mapped to intents associated with lower budgets. Lastly, scenario S4 entails 80% of service requests mapped to risky intent categories, associated with higher impacts and lower risk appetite. These refer to the E-Health, Industrial, and Smart Cities services.

Table 2.3: Intent categories definition.

Parameter	E-Health	Massive Twinning	Mixed-Reality	Industrial	Smart Cities
Intent category	1	2	3	4	5
Tasks [#]	5	8	8	10	6
Deadline [s]	0.200	2	0.100	0.200	1
Throughput [Mbps]	0.1	1000	100	0.1	10
Impact	3	1.5	1	2	1.8
Risk appetite	1	2.5	3	2.5	2.8
Budget [\$]	100	90	30	60	10

Table 2.4: Simulation settings.

Parameter	Setting
M	10
N	5
Reqs/30 min	[750, 1000, 1500, 2000, 2500]
$\Omega^{(m)}$	[40, 2000] Megacycles/s
$q^{(m)}$	[750, 2800]
$L^{(m)}$	[0, 1]
$C_u^{(m)}$	[1, 2] \$/s
$C_{lf}^{(m)}$	[0.001, 0.002]\$
$C_{sec}^{(m)}$	[0.003, 0.005]\$
$U_{(R_s^{(n)}, D_m)}$	[100, 1000] Mbps
$B^{(n)}$	[0.6, 1.2] Mb

The performance of the proposed solution has been evaluated against three baseline approaches, each distinguished by its own method of distributing service

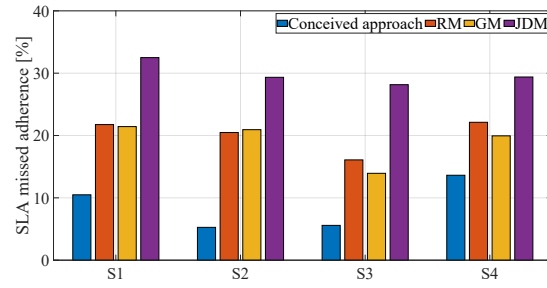
requests across provider domains and by its ability to decompose services into individual tasks.

The considered baseline approaches are described below:

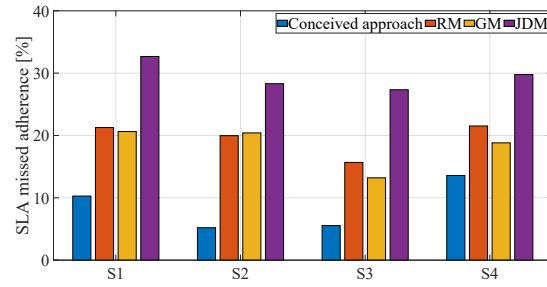
- RM: associations between service requests and available domains are executed randomly, without taking into account any specific preferences.
- Greedy Matching (GM): service requests are greedily assigned to domain providers by iteratively selecting the available and most favorable option based on a given criterion. Specifically, service requests are assigned a priori to the domain that guarantees the lowest end-to-end delay, highest throughput, lowest costs, and lowest cyber risk in the S1-th, S2-th, S3-th, and S4-th scenarios, respectively.
- Joint Decision Making (JDM): service requests are assigned to domains using the algorithm proposed in our previous work [116]. This approach does not include the decomposition of service requests into tasks. Instead, it computes service request preferences through the TOPSIS algorithm and assigns them to the highest-ranked available provider domain without performing a matching procedure.

Firstly, the performance of the proposed approach is evaluated against baseline solutions in terms of SLA missed adherence for each scenario (e.g., S1, S2, S3, S4) and for varying numbers of service requests generated per day (e.g., 750, 1000, 1500, 2000, 2500 requests every 30 minutes). For each generated service request, the average number of inconsistencies related to deadline, throughput, budget, and risk appetite with respect to the established SLAs is calculated. This value is assessed and averaged for all service requests generated every 30 minutes and expressed as a percentage.

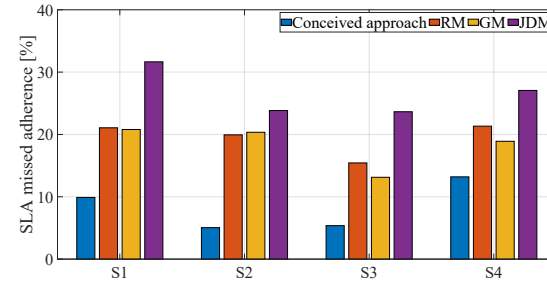
The percentages in Fig. 2.8 illustrate the effectiveness of the proposed model, showcasing a significant improvement in SLA adherence compared to the worst-performing outcomes among the baseline approaches. Low percentages of SLA missed adherence indicate that the proposed approach can effectively identify suitable and available network domain resources to meet all declared service requirements. It consistently achieves a failed compliance rate below 14% across all scenarios and settings, with a minimum of 5.04% in the S2-th scenario with 1500 service requests. In contrast, the baseline approaches frequently report a failed compliance rate exceeding 30%. The JDM approach, in particular, results in the highest failure rate of 32.41% in the S1-th scenario with 750 service requests. Overall, the baseline approaches perform poorly due to their inability to thoroughly evaluate all service requirements articulated by the consumer simultaneously. Specifically, when compared to the worst-performing JDM approach, the proposed approach shows an improvement of up to 22% for 750 service requests every 30 minutes, as reported in Fig. 2.8a. The proposed solution enhances SLA adherence by up to 22.31% with the generation of 1000 service requests every 30 minutes, as illustrated in Fig. 2.8b. With 1500 service requests, the approach achieves an improvement of up to 22.36%, as depicted in Fig. 2.8c. Furthermore, the approach demonstrates improvements of up to 22.44% and 21.14% with 2000 and 2500 service requests every 30 minutes,



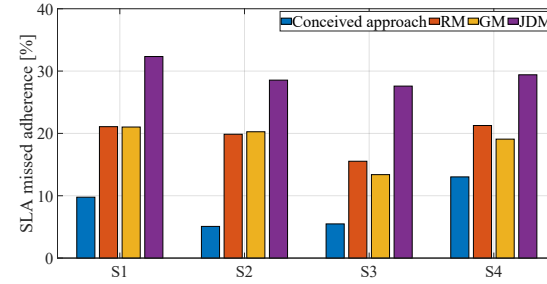
(a) 750 requests/30 min



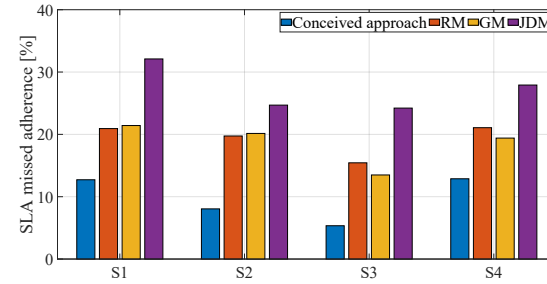
(b) 1000 requests/30 min



(c) 1500 requests/30 min



(d) 2000 requests/30 min



(e) 2500 requests/30 min

Figure 2.8: SLA missed adherence [69].

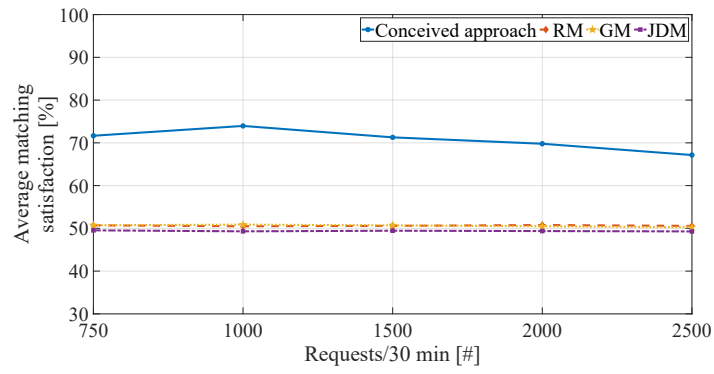
respectively, as shown in Figures 2.8d and 2.8e. These results underscore the approach's ability to identify the service provider that minimizes SLA violations for all intent categories across all reference scenarios.

Fig. 2.9 presents the average matching satisfaction for each reference scenario. It depicts the mean RC assigned to all domains selected through the orchestration procedure to fulfill the tasks of the requested services.

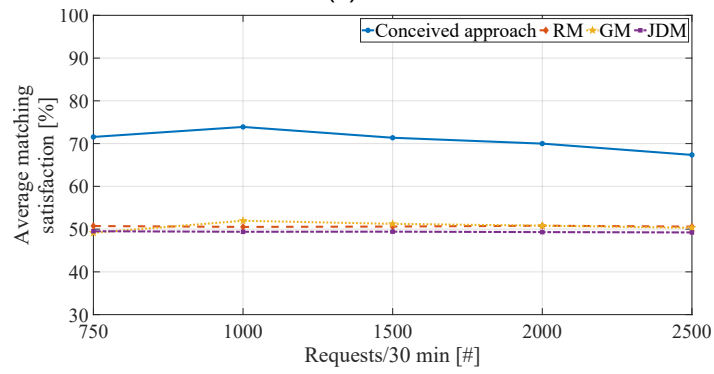
Here, a higher average matching satisfaction indicates the selection of the highest-ranked domain in the computed preference list, which better meets the requirements specified by consumers. This evaluation metric is expressed as a percentage and is reported for an increasing number of generated service requests. Overall, the proposed approach demonstrates superior performance compared to baseline approaches, consistently achieving the highest average matching satisfaction, typically within the 70–80% range across all setups. The baseline approaches, on the other hand, report an average matching satisfaction hovering around 50% (50% for RM and GM, 49% for JDM). Specifically, GM yields substantially lower results compared to the proposed approach. This is because the domain provider selection is based solely on one of the four criteria considered for deriving RC values, without accounting for any trade-off among the evaluation of other requirements. When considering JDM, despite its initial selection of the highest-ranked domain based on relative closeness, the absence of task subdivision means that the most appropriate and available provider choice is not assured. This deficiency arises because JDM does not incorporate any matching game among preferences. Consequently, this approach ranks the lowest in terms of matching satisfaction.

To provide further insights into the effectiveness of the proposed approach, this section presents an analysis of the evaluation criteria for provider selection, traditionally defined as KPIs. Specifically, the average end-to-end delay, expressed in seconds [s], measures the total elapsed time, encompassing both transmission and processing delays, while throughput, expressed in [Mbps], represents the transmission rate between the consumer and the selected network provider. Additionally, novel business needs and security requirements are articulated through KVIs. These include the average cost per service request, expressed in [\$/s], and the average cyber risk per service request. The latter metric quantifies the potential risk a consumer faces when their service request is processed by a matched domain, weighted by the associated risk appetite.

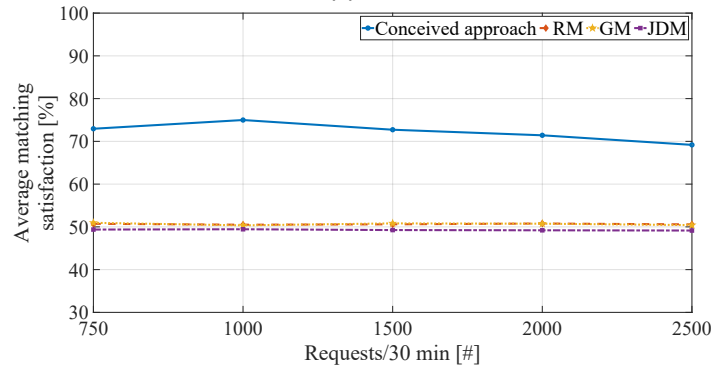
Figures 2.10 and 2.11 illustrate the analysis in the S2-th scenario, characterized by a majority of service requests requiring higher throughput. Specifically, Fig. 2.10 compares the performance of the proposed approach against baseline solutions considering 1500 service requests generated every 30 minutes. Regarding the average end-to-end delay, the novel strategy proposed in this work consistently achieves the lowest values, remaining under 0.01 seconds. In contrast, all the baseline approaches report significantly higher delays. Specifically, the average end-to-end delay with the proposed strategy is reduced by an order of magnitude compared to RM and GM, and by two orders of magnitude compared to JDM. In terms of average throughput, the proposed approach achieves peak performance of 7800 Mbps, which is 15% higher compared to RM, which does not consider any service requirements for provider domain selection. The other baseline approaches, instead, report performance similar to the proposed approach. In this scenario, GM consistently



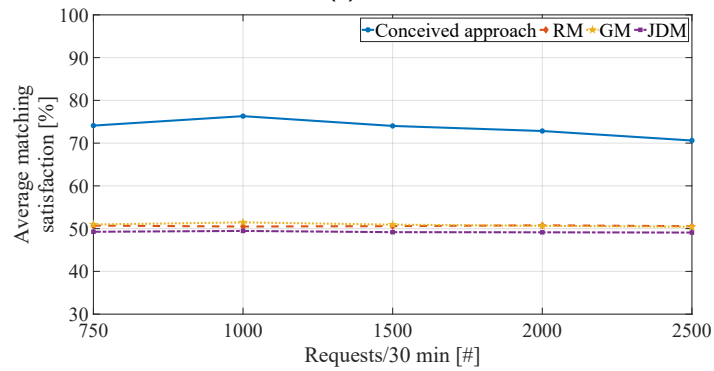
(a) S1



(b) S2



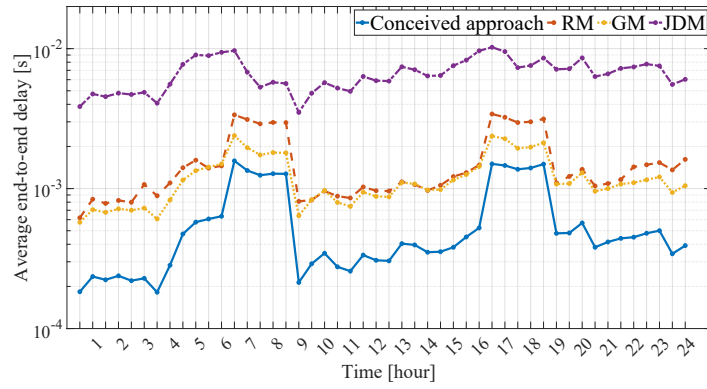
(c) S3



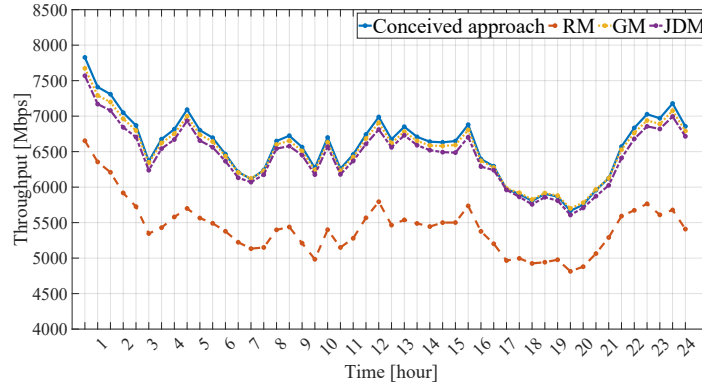
(d) S4

Figure 2.9: Average matching satisfaction per request [69].

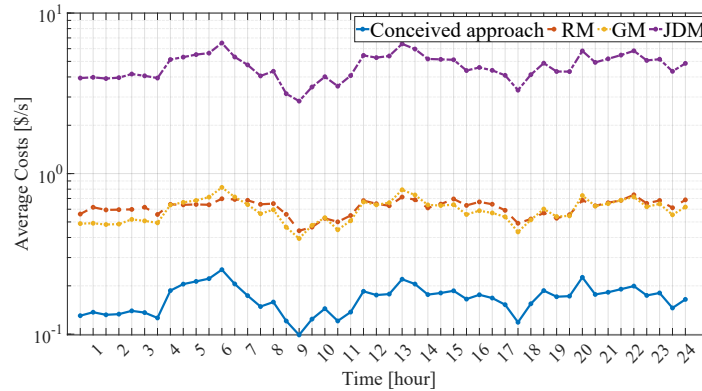
favors provider domains that guarantee higher transmission rates, resulting in a



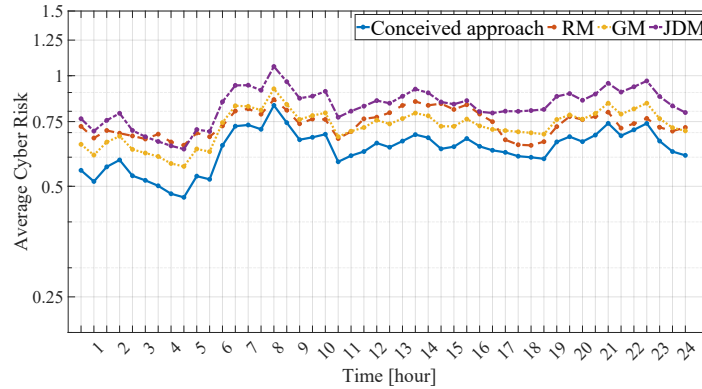
(a) Average end-to-end delay



(b) Average throughput

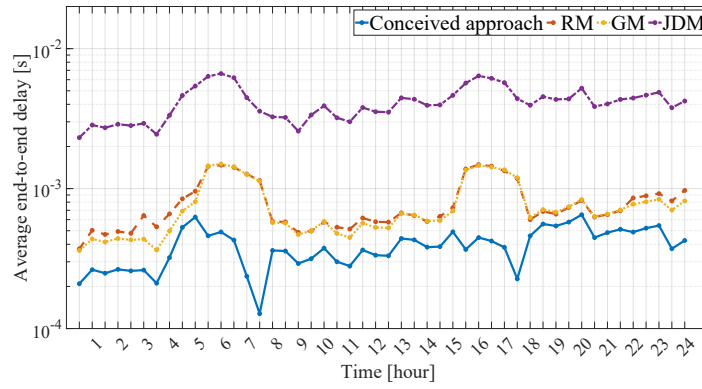


(c) Average costs

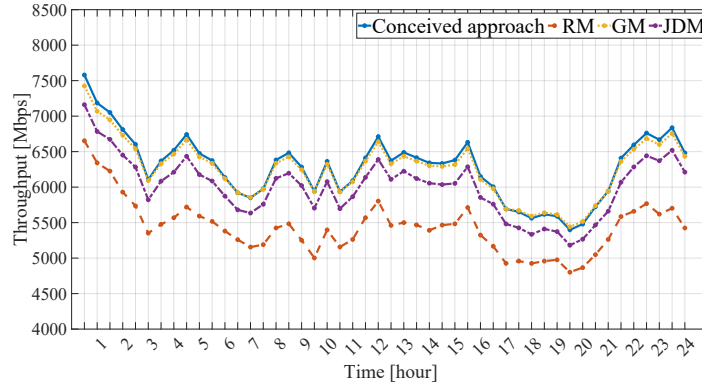


(d) Average cyber risk

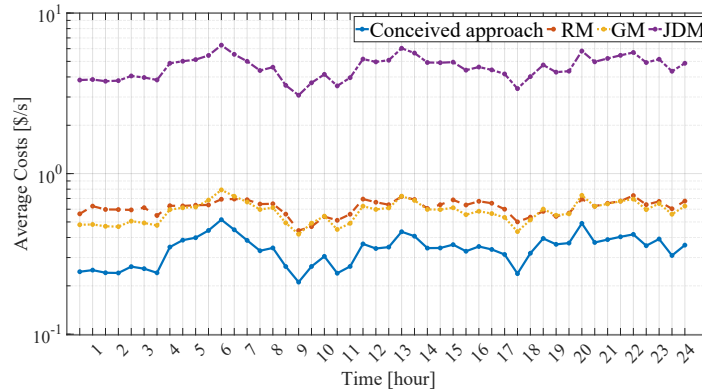
Figure 2.10: KPI and KVI analysis assessed in S2 with 1500 requests/30 min [69].



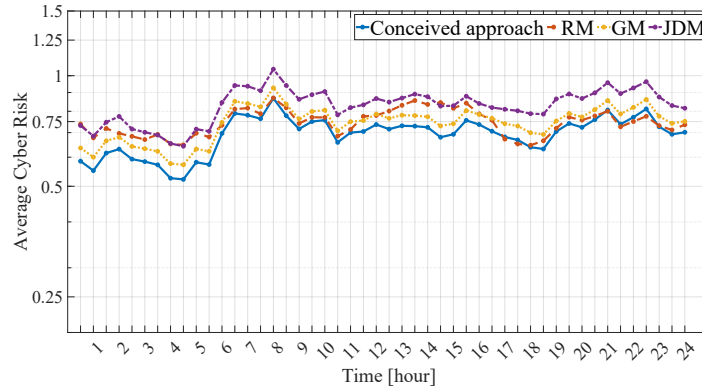
(a) Average end-to-end delay



(b) Average throughput

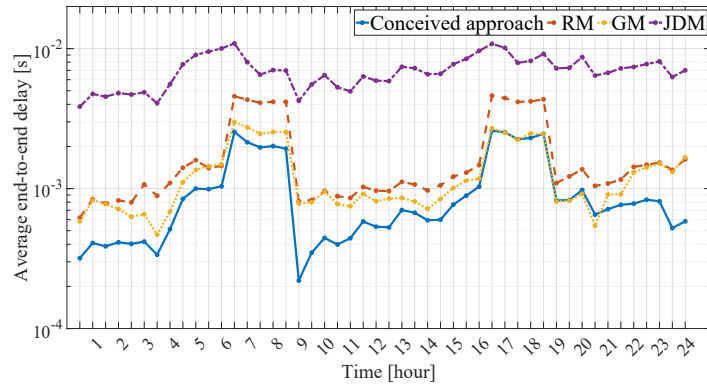


(c) Average costs

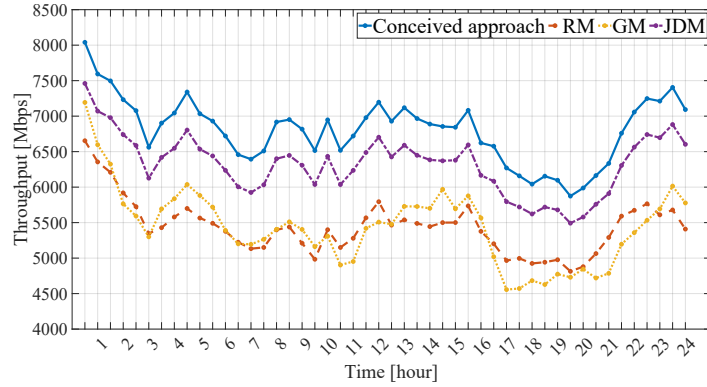


(d) Average cyber risk

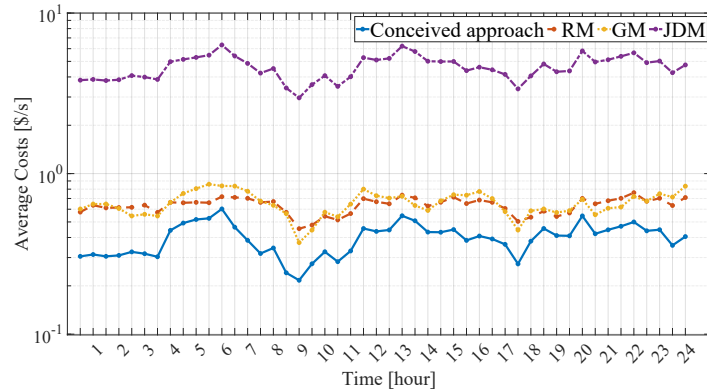
Figure 2.11: KPI and KVI analysis assessed in S2 with 2500 requests/30 min [69].



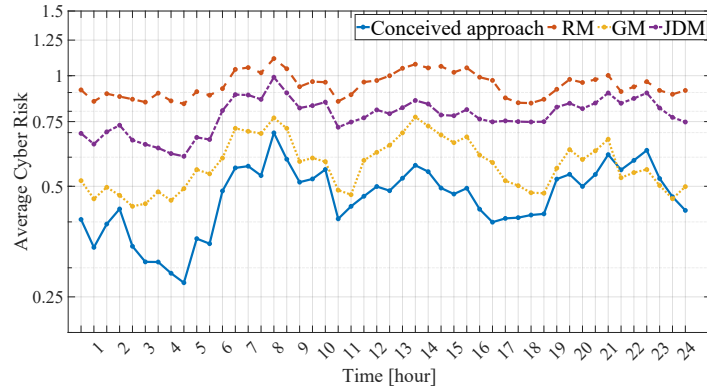
(a) Average end-to-end delay



(b) Average throughput

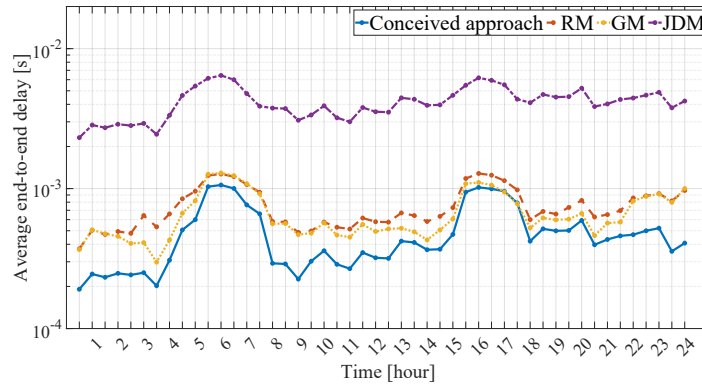


(c) Average costs

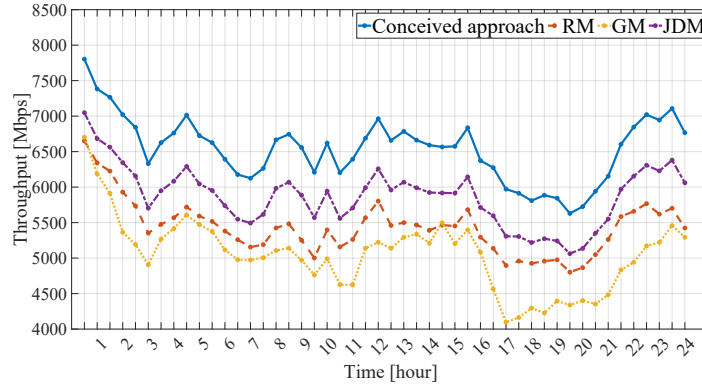


(d) Average cyber risk

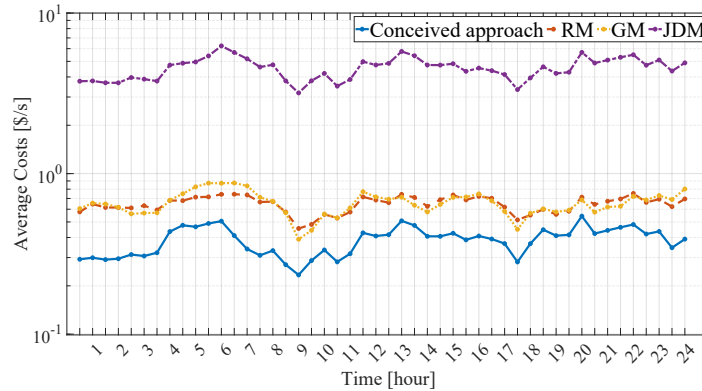
Figure 2.12: KPI and KVI analysis assessed in S4 with 1500 requests/30 min [69].



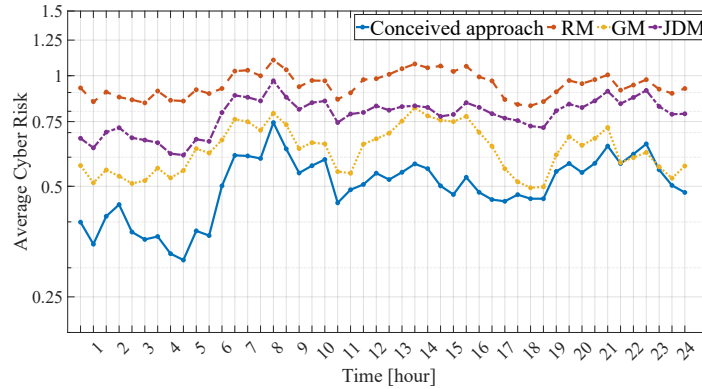
(a) Average end-to-end delay



(b) Average throughput



(c) Average costs



(d) Average cyber risk

Figure 2.13: KPI and KVI analysis assessed in S4 with 2500 requests/30 min [69].

throughput outcome very close to the optimal. Similarly, JDM achieves an average throughput that is 5% lower than the proposed approach, as the JDM methodology assigns higher weights to this KPI in its multi-criteria decision-making process, thereby favoring domains that guarantee higher transmission rates.

Also considering the average costs per service request, the proposed approach registers the most favorable results, consistently remaining under 1 \$/s, whereas RM and GM report costs that are an order of magnitude higher. The JDM approach incurs the highest costs, nearly two orders of magnitude greater than those of the proposed solution. This is attributed to the intensive use of the same network resources by the selected domain, which could be the most expensive. Furthermore, considering the average cyber risk, the proposed approach consistently yields results below those reported by the other approaches, demonstrating the capacity to select the safest domain provider for consumers' requests. Specifically, the baseline approaches report results that are 10–30% higher compared to the proposed approach. In particular, both RM and GM entirely neglect the security requirement. Although the JDM approach employs the same multi-criteria decision-making algorithm, it does not perform any matching procedure. As a result, it replaces the highest-ranked but unavailable domains with lower-ranked and potentially riskier domains.

Fig. 2.11 confirms the efficiency of the proposed approach even with an increasing number of generated service requests. Specifically, when considering 2500 service requests per 30 minutes, the registered average end-to-end delay remains under 0.001 seconds. This is 20% lower than the delays reported by both RM and GM, and 90% lower than the delay reported by JDM. This analysis confirms the validity of the proposed methodology compared to the baselines, demonstrating its ability to identify network providers that concurrently satisfy both KPIs and KVIs, even with an increasing number of generated service requests. To investigate the performance of the proposed methodology under different conditions, the S4-th scenario was evaluated. In this scenario, the majority of service requests are characterized by a lower risk appetite for the demand.

In detail, Fig. 2.12 compares the performance of the proposed solution against baseline approaches with 1500 service requests generated every 30 minutes. In terms of average end-to-end delay, the proposed approach consistently reports values generally under 0.001 seconds, with peaks reaching 0.003 seconds during certain hours. These fluctuations can be attributed to the varying resources exposed by domains through NDTs, which change throughout the day. Both RM and GM exhibit results up to 50% higher than those obtained with the proposed approach. JDM once again registers the highest average end-to-end delays, doubling the results achieved with the proposed strategy. Similar to the S2-th scenario, the proposed approach guarantees the highest average throughput, peaking at 8000 Mbps. It effectively distributes requests among domain providers to ensure high performance for all service requests. In contrast, all baseline approaches report significantly lower performance. Specifically, in this case, GM neglects the throughput requirement of the generated service requests entirely, resulting in an average throughput that is 10% lower than that achieved by the proposed strategy. RM consistently fails to identify provider domains guaranteeing high throughput by not addressing any articulated service requirements. Meanwhile, JDM reports results varying between 7500 and

5500 Mbps, often identifying providers that do not fully meet the requirements specified by consumers.

Concerning the average costs, the reported KVI for the proposed approach is consistently lower than those reported by the baseline approaches, as it is closely related to the transmission delays contributing to the overall end-to-end delay. Consequently, the average costs remain under 0.6 \$/s, whereas other approaches (e.g., RM, GM) reach values 50% higher. The JDM approach, by not performing a matching procedure, selects high-ranked and available domains that may, however, utilize the most expensive resources. Regarding the registered cyber risk, the proposed strategy exhibits provider selections similar to those reported by GM. Indeed, this baseline approach overlooks all service requirements except for security in this scenario, thus favoring network domains that guarantee safer processing of service requests. Conversely, the other baseline techniques identify riskier provider domains either by neglecting the security requirement (e.g., RM) or by assigning requests to available but riskier domains (e.g., JDM). The efficiency and scalability of the conceived solution are evaluated in Fig. 2.13, which reports the performance in the S4-th scenario with an increasing number of generated service requests. In particular, when considering 2500 service requests per 30 minutes, the approach demonstrates even better performance in terms of average end-to-end delay, with peaks reaching 0.001 seconds.

Overall, this analysis underscores both the effectiveness and scalability of the proposed approach in deploying B5G service requests mapped to different intent categories. This results in the joint minimization of SLAs violations related to latency, throughput, costs, and cyber risk requirements.

2.4 From theory to practice: a COHERENT perspective

Given the importance of open, extensible, and practically deployable network architectures, the COHERENT project, developed within the PNRR RESTART program, aims to define a holistic architectural framework that is coherent from both technical and business perspectives. More specifically, COHERENT is conceived to support KVIs beyond traditional KPIs, while enabling a simple and scalable framework for monitoring, optimization, and management across heterogeneous communication segments.

To achieve this objective, the architecture [134] relies on DTs as a unifying abstraction for representing network segments, resources, and their operational conditions. This abstraction allows operators to observe network behavior, correlate infrastructure status with service demands, and support automated decision-making across core, edge, and NTN domains. In this way, COHERENT provides an open architectural basis capable of integrating different technological innovations within a common orchestration framework.

2.4.1 Reference Architecture for Digital Twin Networks

The reference architecture for DTNs, defined in [135], addresses the management challenges emerging in sixth-generation networks as infrastructures evolve toward ubiquitous in-network computing distributed across radio access, core, and edge domains. Orchestration frameworks must therefore optimize both performance and sustainability metrics, with future networks exposing computation-as-a-service capabilities that enable distributed AI applications within the emerging “Network for AI” paradigm.

At its core, this architecture builds upon the DT model—widely adopted in industrial systems for synchronized virtual replicas of physical assets—and applies it to telecommunications. Here, a DTN emerges as an ecosystem of interconnected DTs that continuously monitor network operations, process telemetry data, and predict future states. These capabilities facilitate closed-loop reconfiguration of the physical network, ensuring SLAs compliance while addressing sustainability requirements expressed through KVIs alongside traditional KPIs. Fully aligned with emerging standards from IRTF and ITU-T Y.3090, the architecture supports sustainability-driven IBN by integrating network-level DTs with lightweight prediction mechanisms across three distinct layers, illustrated in Fig. 2.14.

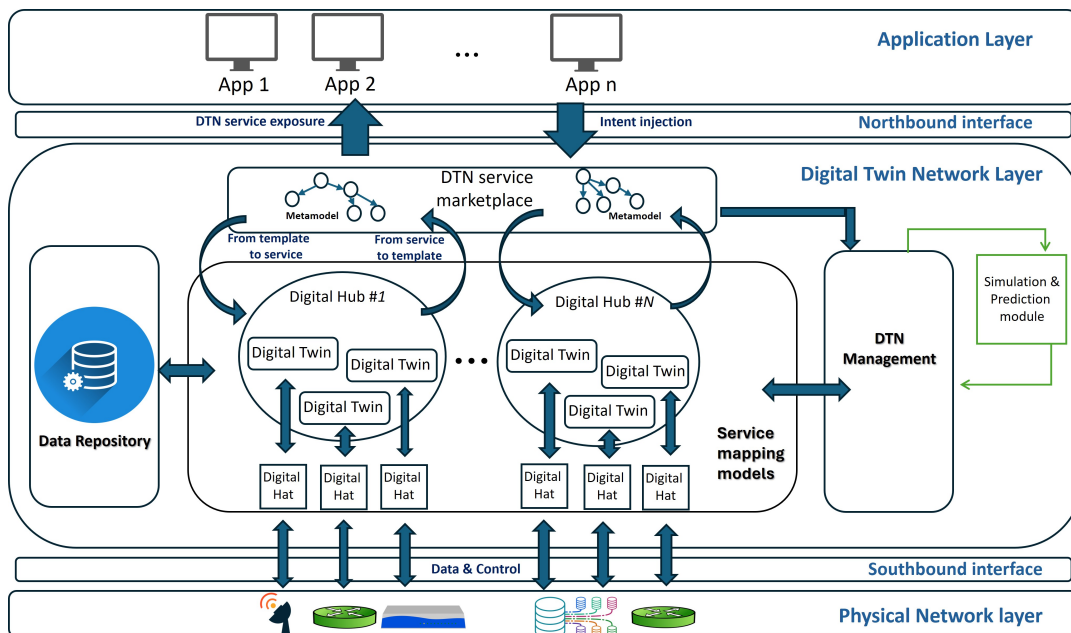


Figure 2.14: Reference architecture for the proposed DTN [134].

The Physical Network Layer encompasses all physical, digital, and phygital network components within programmable 6G environments. Network managers and orchestrators implement Software-Defined Systems and Networks (SDSN)/Network Function Virtualization (NFV) principles, where individual components may be abstracted into DTs or host DTs within containerized runtimes. Above this foundation lies the DTN Layer, which hosts DTNs instantiated, managed, and terminated by a dedicated DTN manager following predefined templates. Several key entities populate this layer: the DTN itself serves as a digital representation of heterogeneous telecommunication networks composed of multiple DTs, supporting both

permanent infrastructure representations and dynamic service-specific instances with self-contained management. Digital Hub (DTH) group related DTs by asset type, function, or operational characteristics, while Digital Hats (DHs) act as adapters, mediators, and wrappers—allowing multiple instances per physical asset across different hubs or DTNs. The Service Marketplace maintains dynamic catalogs of Digital Hat services, metamodels define evolving topology and interaction patterns through peer consensus, and services orchestrate microservices representing individual DT capabilities. DTN Management coordinates these elements while the Simulation and Prediction Module evaluates configuration alternatives to balance user intents with sustainability and efficiency goals.

The Application Layer consumes DTN platform services according to defined policies. Sustainability-aware intents enter through the northbound interface and map to corresponding microservices for execution. Standardized interfaces complete the architecture: the northbound interface exposes APIs for intent submission and service access, while the southbound interface defines telemetry and control protocols between DTs and physical devices. This layered design empowers the DTN to analyze network behavior, predict issues, and trigger timely reconfigurations, seamlessly integrating KVLs into decision-making processes to ensure both SLA compliance and long-term sustainability alongside conventional performance optimization.

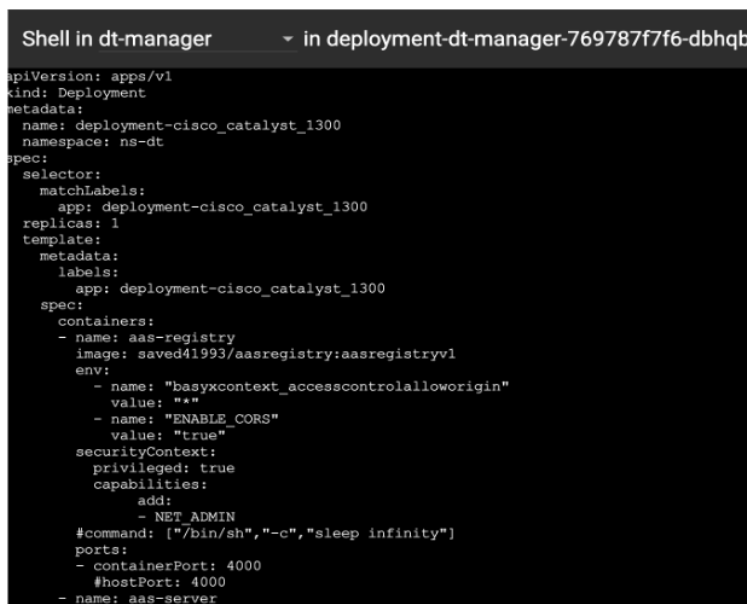
2.4.2 Cloud-to-Edge Management via the Asset Administration Shell

The Asset Administration Shell (AAS) originates from the Reference Architectural Model for Industry (RAMI) reference model, which defines an Industry 4.0 component as the combination of a physical asset and its digital representation. Standardized by International Electrotechnical Commission (IEC) 63278-1:2023 ED1, the AAS provides a uniform mechanism to digitally represent assets throughout their lifecycle, covering both type and instance phases. It encapsulates structural properties, capabilities, operational status, and runtime data, neatly split into a passive part containing static data and history, and an active part with functional logic for control and interaction. This separation elegantly supports scalable cloud-to-edge integration while maintaining a consistent abstraction layer across the entire infrastructure.

Eclipse BaSyx [136] delivers a reference implementation of the AAS through a modular software stack perfectly aligned with Industry 4.0 standards. The framework enforces standardization via the AAS and its extensible submodel ecosystem, while architectural complexity is elegantly managed through a Service-Oriented Architecture (SOA) approach. The Virtual Automation Bus (VAB) effectively abstracts communication heterogeneity, and enterprise-grade operations are supported by comprehensive workflow orchestration, monitoring capabilities, and fine-grained access control based on Role-Based Access Control (RBAC). This combination enables truly interoperable, secure, and scalable digital representations spanning from cloud data centers to edge deployments [137].

BaSyx provides Software Development Kits (SDKs) across multiple languages—Java and C# for full-featured applications, C++ for real-time edge integration, and Python for rapid prototyping. For this implementation, the Java SDK was specifically selected due to its maturity, scalability, and position as the reference implementation ensuring Industrial Digital Twin Association (IDTA) AAS API compliance. This choice guarantees long-term stability and seamless interoperability for core services like the AAS Repository and Registry.

Creating a DT involves instantiating it as an AAS that represents the physical asset. The process unfolds in two main steps: first defining submodels that describe properties, behaviors, and operational metrics, then implementing the AAS application with its execution logic. These can be crafted either through the AASX Package Explorer or programmatically using the Java SDK. Supporting the full DT lifecycle are several core components, all distributed as Docker containers for portable deployment: the AAS Environment provides runtime support with Representational State Transfer (REST) interfaces for CRUD operations on AASs and submodels; the AAS Registry enables discovery and resolution; the AAS WebView offers graphical data interaction; the Submodel Registry handles registration and lookup; and DataBridge facilitates real-time data exchange with external sources via Hypertext Transfer Protocol (HTTP), Message Queuing Telemetry Transport (MQTT), or OPC UA.



```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-cisco_catalyst_1300
  namespace: ns-dt
spec:
  selector:
    matchLabels:
      app: deployment-cisco_catalyst_1300
  replicas: 1
  template:
    metadata:
      labels:
        app: deployment-cisco_catalyst_1300
    spec:
      containers:
        - name: aas-registry
          image: saved41993/aasregistry:aasregistryv1
          env:
            - name: "basyxcontext_accesscontrolalloworigin"
              value: "*"
            - name: "ENABLE_CORS"
              value: "true"
          securityContext:
            privileged: true
            capabilities:
              add:
                - NET_ADMIN
            #command: ["/bin/sh", "-c", "sleep infinity"]
        - name: aas-server
          ports:
            - containerPort: 4000
              #hostPort: 4000

```

Figure 2.15: YAML file of the DT deployment [134].

Figure 2.15 illustrates the Cisco Catalyst YAML configuration file within the DT namespace. This configuration is instantiated on a per-DT basis, as each DT requires a specific deployment profile tailored to its associated physical asset. Consequently, each DT needs its own set of containers for proper operation. The example highlights the container reference for the AAS Registry—a core component essential for the system’s overall functionality. Two complementary HTTP/REST interfaces complete the AAS offering: the AAS Repository API manages complete AASs, while

the AAS API provides fine-grained access to individual DTs, including navigation and manipulation of submodels, properties, and operational data.

For deployment, DTs run on a Kubernetes (K8S) platform where each DT and its services operate within one or more pods. Services ensure stable networking, ConfigMaps and Secrets handle configuration and sensitive data, while Persistent Volumes and Claims maintain state continuity across pod rescheduling or failures. Bidirectional synchronization with physical assets occurs through messaging protocols like MQTT, ensuring continuous alignment between physical systems and their digital counterparts [138].

Finally, the AAS metamodel—governed by IDTA (IDTA-01001-3.0.1)—establishes relationships among assets, AASs, submodels, interfaces, and concept dictionaries, creating a robust semantic foundation for interoperability. Aligned with IEC 62890 and RAMI, it distinguishes between type and instance representations, enabling flexible modeling across diverse industrial scenarios while ensuring consistency and comprehensive lifecycle management from generic asset definitions to concrete operational instances.

2.4.3 COHERENT Architecture Implementation

The COHERENT architecture implements the reference design from ITU Y.3090 as a distributed cloud-edge system, orchestrated by K8S in the cloud and extended to the edge via KubeEdge, as shown in Fig. 2.16. This demo cluster, comprising one cloud node and one edge node, effectively showcases the core operational capabilities.

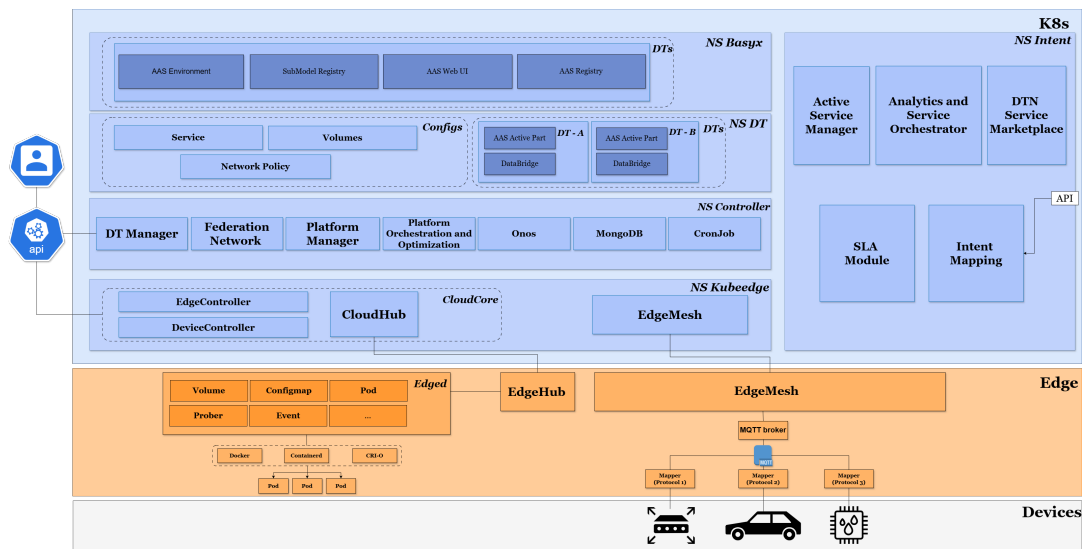


Figure 2.16: COHERENT implementation modules [134].

On the cloud side, five dedicated namespaces organize the components. The DT namespace hosts individual DTs representing physical devices and their configurations, providing connectivity, persistent storage, and synchronization capabilities. For each DT, only the Active Part and DataBridge components are instantiated here, while shared AAS services reside in the dedicated BaSyx namespace to avoid duplication and ensure consistent management.

The BaSyx namespace centralizes common services including the AAS Registry, Submodel Registry, Web UI, and Environment, serving all DTs efficiently. The Controller namespace acts as the orchestration brain, coordinating DT infrastructure through several key components: the DT Manager handles lifecycle operations, the Platform Manager monitors nodes/services/pods, the Federation Network manages inter-DT connectivity, and the ONOS Controller enables IBN alongside video streaming demos. A CronJob periodically pulls updated DT specifications from GitLab, MongoDB provides persistent storage, and the Platform Orchestration & Optimization component leverages K8S for resource efficiency.

The Intent namespace (previously NS SLA) manages service-level requirements and intent processing: the Intent Mapping translates customer requests into structured intents, the DTN Service Marketplace offers KVI-optimized solutions, Analytics & Service Orchestrator performs resource optimization, the SLA Module validates requests, and the Active Service Manager composes high-level services for execution. Finally, the KubeEdge namespace facilitates cloud-edge synchronization through Cloud Hub for communication, Edge Controller for edge orchestration, and Device Controller for device lifecycle management. EdgeMesh ensures reliable bidirectional MQTT communication between DTs and physical devices. At the edge, Edge Hub serves as the local control plane while EdgeMesh, enhanced by Mapper connectors, maintains continuous cloud connectivity across heterogeneous devices.

COHERENT prioritizes interoperability through semantic ontologies and standardized device capability representations. A REST/HTTP architecture simplifies access/control, with secure authentication/encryption and open standards promoting adoption. YANG-to-OMA LwM2M transcoding ensures data model compatibility, while the AAS metamodel—integrating Digital Hub and Digital Hat components—provides the semantic foundation for cross-domain operations.

2.4.4 PoC Evaluation: Energy Efficiency

Two comprehensive Proofs of Concept (PoCs) validate how DTs, intent-based orchestration, analytics, and K8S-native management optimize energy efficiency in production-like scenarios while preserving service quality through automated decision-making.

2.4.4.1 PoC 1 – Energy Efficiency in 5G Radio Access Networks

The first PoC targets 5G RAN energy savings via a cluster of five gNBs controlled by an edge-based RAN Energy Saving Controller. Each gNB is modeled as an AAS-DT exposing configuration and performance metrics including Physical Resource Block (PRB) usage, throughput, serving Signal-to-Interference-plus-Noise Ratio (SINR), and hardware characteristics (Fig. 2.17).

The intent “*Energy efficiency improvements in gNB RAN cluster*” specified KPI constraints (cell capacity $\leq 80\%$, peak throughput ≥ 1 Mbps) alongside KVIs covering Energy Efficiency, Connection Quality, and Service Quality Index. Using the RIC-TaaP simulation platform — integrating FlexRIC, ns-Open Radio Access Network (ORAN), the 5G-LENA NR module, and an Energy Saving xApplication (xApp) — two modes were compared: a baseline in which all gNBs remain always active, and

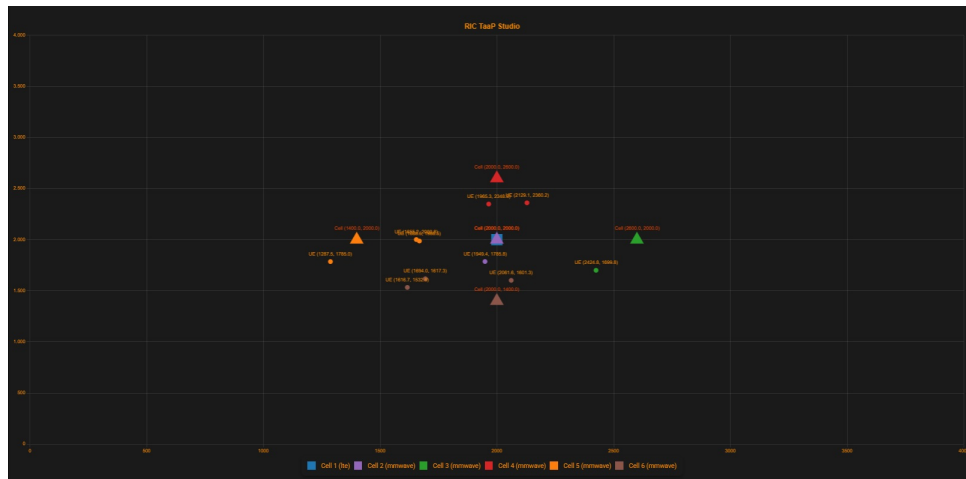


Figure 2.17: Simulated RAN scenario with gNBs and UEs [134].

a DT-assisted mode in which the controller dynamically switches gNBs on/off and offloads traffic according to predicted energy consumption and KPI compliance.

Machine Learning (ML) models — including LLM transformers and a custom XNS algorithm — predicted energy consumption from KPI time series and proposed safe switch-off strategies. Results demonstrated a 6.9% average reduction in cluster energy consumption while maintaining throughput and PRB targets (Fig. 2.18). Transient deviations arising during handovers triggered by gNB deactivation were rapidly corrected by the closed-loop control, with both ML approaches proving viable and the domain-specific model providing higher stability.

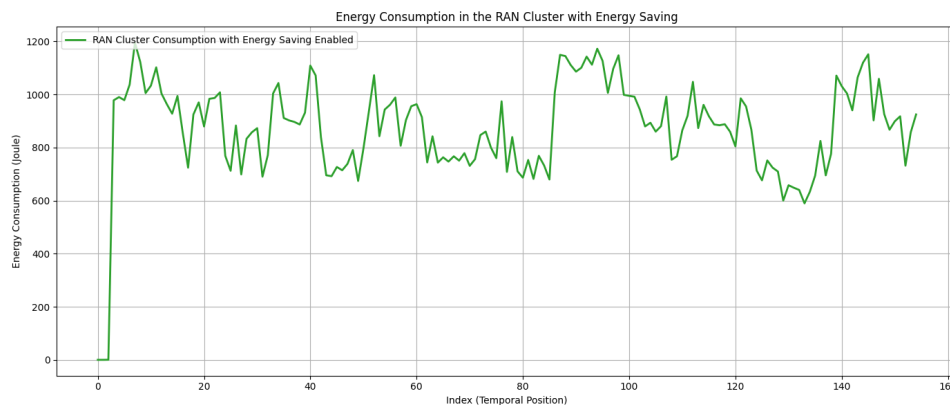


Figure 2.18: RAN cluster energy consumption with LLM Energy Saving enabled [134].

2.4.4.2 PoC 2 – Energy Optimization in Fixed Access Broadband Networks

The second PoC addressed fixed broadband via GPON emulation (1 Optical Line Terminal (OLT), 2 BNGs). High-capacity BNGs contribute significantly to OPEX, particularly during low-traffic periods: while OLTs remain always active, SDN-enabled BNGs can be consolidated or deactivated to reduce energy waste. DTs replicated the

broadband network, monitored traffic patterns and subscriber distribution, and estimated energy footprints under alternative configurations.

An energy-saving intent submitted to the COHERENT platform triggered BNG consolidation during off-peak hours: the DT-assisted control loop determined that one BNG could be safely deactivated, reducing both system-wide and node-level power consumption without service disruption (Fig. 2.19).



Figure 2.19: GPON power consumption with BNG consolidation [134].

Both PoCs confirm that DT-driven, intent-based automation delivers measurable energy savings across heterogeneous network domains — mobile RAN and fixed access — while preserving SLA compliance and supporting fully automated, closed-loop energy management.

Concluding Remarks

This chapter addressed **RQ2** by demonstrating how NDTs can evolve from descriptive infrastructure replicas into active decision-support components within closed-loop orchestration frameworks.

The TOPSIS-based MCDM framework introduced in Section 2.2 enables context-aware domain selection during disaster-response scenarios, where survivability and trustworthiness requirements must be jointly satisfied under degraded network conditions. The game-theoretic matching mechanism presented in Section 2.3 further extends this capability to the joint optimization of KPIs and KVIIs, achieving consistently lower end-to-end delay, higher throughput, reduced operational cost, and lower cyber risk compared to baseline approaches, even under increasing service-request loads. The implementation on the COHERENT platform grounds these contributions within a real cloud-to-edge architecture, where two PoCs demonstrate that DT-driven, intent-based automation can deliver measurable energy savings across heterogeneous network domains—including mobile RAN and fixed-access infrastructures—while preserving SLA compliance.

Overall, these results establish DTs as a fundamental orchestration substrate for future intelligent networks, enabling adaptive, value-aware, and autonomous resource management. This lays the foundation for Chapter 3, where agentic AI paradigms extend autonomous control beyond centralized assumptions toward distributed and trust-aware coordination in future IoE environments.

Chapter 3

Distributed and Trustworthy Orchestration for Future Internet-of-Everything Environments

As network environments evolve toward increasingly distributed, pervasive, and heterogeneous operational conditions, the limitations of centralized orchestration become progressively more evident. The intent-driven control mechanisms and DT-assisted decision support frameworks introduced in the previous chapters provide powerful abstractions for managing complexity; however, they remain insufficient when orchestration must span multiple autonomous domains, edge entities, and dynamically interacting nodes.

In highly decentralized environments, centralized coordination alone can no longer guarantee the responsiveness, trustworthiness, and operational continuity required by next-generation services. Intermittent connectivity, localized decision-making, and cross-domain interactions demand orchestration mechanisms capable of operating effectively even in the absence of a globally available control authority. Consequently, orchestration must progressively incorporate autonomy, secure cooperation, and distributed trust management — particularly at the edge, where local intelligence becomes an active component of the control loop. In this context, resilience extends beyond traditional fault tolerance and increasingly reflects the ability of the orchestration process to remain effective under uncertainty, disruption, and partial system degradation.

Building upon the perspective developed throughout the thesis, **RQ3** is addressed by showing that distributed and trustworthy orchestration does not require abandoning the principles established in the previous chapters, but rather extending them toward agentic and trust-aware forms of control. In this evolution, intent handling, predictive network knowledge, and autonomous decision-making converge into a unified orchestration capability for future IoE environments.

This transition introduces several fundamental research tensions, including the balance between autonomy and controllability, the trade-off between distributed intelligence and coordination overhead, and the need to reconcile reasoning capability with sustainability across the cloud–edge–device continuum. Rather than considering agentic AI as an isolated technological trend, this chapter frames it as the

natural evolution of intent-driven and predictive orchestration for highly dynamic and decentralized IoE systems.

Accordingly, the chapter first discusses the foundations of agentic AI and its applicability to autonomous network management. It then presents a three-tier orchestration architecture integrating LLM-, Small Language Model (SLM)-, and Tiny Machine Learning (TinyML)-based agents according to different operational and reasoning requirements across cloud, edge, and device domains. Finally, it introduces a decentralized trustworthiness dissemination framework for Social Internet of Everything (SloE) systems, highlighting how distributed intelligence and secure cooperation can be jointly supported in large-scale heterogeneous environments.

3.1 Agentic AI for Autonomous Internet-of-Everything

The first contribution [139] addresses the problem of organizing and coordinating intelligence across cloud, edge, and device layers to enable scalable, real-time, and trustworthy decision-making in dynamic IoE environments. In these scenarios, the interaction of devices, people, data, and processes across heterogeneous infrastructures demands adaptive management with minimal human intervention. The telecommunications sector is already transitioning in this direction, with initiatives such as European Telecommunications Standards Institute (ETSI)'s Zero-touch network and Service Management (ZSM) and Experienced Networked Intelligence (ENI), TM Forum's autonomous network architecture, ITU Telecommunication Standardization Sector (ITU-T) frameworks, and ongoing 3rd Generation Partnership Project (3GPP) enhancements converging around closed-loop automation, intent-based management, and DTs [140], [141], [142]. To address the remaining gaps in distributed autonomy and verifiable control, the proposed framework organizes LLM-, SLM-, and TinyML-powered agents within a hierarchical three-tier architecture, integrating DTs for predictive action validation and IBN for semantic intent translation, enabling high-level service objectives to be continuously interpreted, verified, and enforced across the cloud-to-device continuum.

3.1.1 From Foundation Models to Distributed Agentic Orchestration

Agentic AI embeds learning-based models within autonomous entities that observe their environment, plan under goals and constraints, act through tools, and adapt over time. In IoE systems, such autonomy is inherently multi-tier, requiring complementary intelligence granularities across cloud, edge, and device layers. Large-scale reasoning and long-horizon coordination are naturally associated with cloud-level orchestration, whereas latency-sensitive adaptation and localized autonomy increasingly depend on lightweight edge and embedded agents. Consequently, the coexistence of LLM-, SLM-, and TinyML-based agents should not be interpreted as a simple taxonomy of models, but rather as an architectural requirement for scalable orchestration across the cloud–edge–device continuum.

This section therefore introduces the main agentic capabilities required for future IoE environments, reviews the complementary roles of LLM-, SLM-, and TinyML-powered agents, and discusses Multi-Agent Systems (MASs) and interoperability mechanisms [143], [144]. Throughout the discussion, particular attention is devoted to the open challenge of coordinating heterogeneous reasoning entities while preserving trustworthiness, responsiveness, and sustainability.

3.1.1.1 Language-Model-Powered Agents

An AI agent is an autonomous software entity that pursues goals and adapts with minimal human intervention [143]. In networking, this entails three closed-loop functions: (i) observing a dynamic system, (ii) selecting actions, often over multiple steps, and (iii) executing them through tools interfacing with operational platforms. Language-model-powered agents realize this loop by using a language model as their cognitive core. Although LLMs and SLMs differ in scale and deployment constraints, they share a common pattern: transforming structured world-state representations into action plans and tool invocations under explicit policies. Fig. 3.1 outlines the reference agent architecture, comprising three main modules, i.e., brain, perception, and action. Specifically,

1. **Brain:** the cognitive core, implemented by a language model and extended with:
 - *Natural Language Interaction:* context-aware dialogue and comprehension;
 - *Knowledge Access:* retrieval from internal or external sources;
 - *Memory Management:* short- and long-term context handling;
 - *Reasoning and Planning:* multi-step problem solving via chain-of-thought, search, or reflective loops;
 - *Generalization:* in-context learning, zero-shot inference, and continual adaptation.
2. **Perception:** interfaces with the environment by converting multimodal inputs (text, audio, visual) into structured representations.
3. **Action:** generates outputs and executes operations, including natural language responses, API calls, code execution, database manipulation, or control of physical and simulated systems, thereby closing the perception-action loop.

Through the coordinated operation of these modules, language-model-powered agents achieve autonomy, adaptability, and situational awareness. Their modular design enables deployment across domains, from network automation to distributed cyber-physical systems.

3.1.1.2 LLM- and SLM-Powered Agents

LLMs are large-scale transformer models trained on broad corpora and aligned via fine-tuning (e.g., supervised tuning, Reinforcement Learning from Human Feedback (RLHF)) to follow instructions and produce structured outputs. Their primary

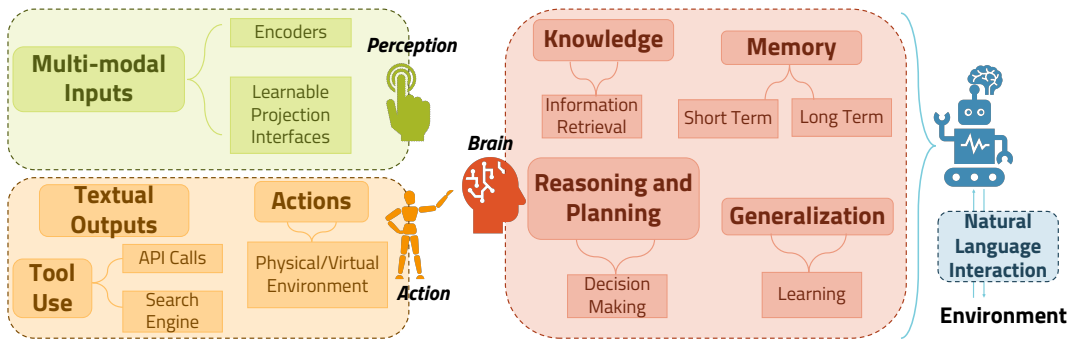


Figure 3.1: Architecture of a tool-using language-model agent [139].

strength lies in semantic and complex reasoning: they interpret ambiguous intents, reason across domains, and generate multi-step strategies under conflicting objectives. They also provide flexible interfaces for multimodal and unstructured inputs, facilitating human-in-the-loop supervision and rapid integration of new intent types. However, in real-time and mission-critical closed loops, LLM-powered agents face constraints in latency, cost, and energy consumption. Their generative nature may also reduce predictability, as small variations in input state can yield qualitatively different actions, potentially leading to instability phenomena such as policy oscillation. In practice, much of agent behavior consists of repetitive and structured tasks—such as parsing, tool invocation, constrained generation, and contextual summarization—that do not require the full generality of an LLM [145].

SLMs address this inefficiency by providing task-specialized reasoning with significantly fewer parameters (typically a few billion) and more predictable execution profiles. A systematic transition from LLM-centric to SLM-centric agents can follow a three-step workflow: first, production LLM-agents log prompts, responses, tool calls, and latency profiles; second, clustering techniques identify recurring task patterns; third, candidate SLMs are selected and fine-tuned on task-specific clusters, enabling rapid specialization with limited computational resources. Together, LLMs and SLMs form complementary substrates for Agentic AI: LLMs support high-level reasoning and cross-domain generalization, while SLMs enable efficient, scalable execution under strict latency, energy, and footprint constraints. This division of labor facilitates the replication of intelligence across cloud and edge tiers in large-scale IoE deployments [145].

3.1.1.3 TinyML-Powered Agents

TinyML extends the agentic paradigm to microcontroller-class devices, enabling always-on perception and local decision-making on sensors and radio nodes [146]. The main deployment bottleneck is not parameter count but peak activation memory during inference. Even compact models may exceed the limited SRAM available on such devices, especially in early high-resolution layers.

Mitigating this memory imbalance requires joint optimization of neural architectures and inference engines. Techniques such as memory-aware neural architecture search and patch-based inference reduce activation footprints by up to an order of magnitude, enabling practical deployment on commodity microcontrollers.

These advances support ultra-low-power perception and control loops at the device tier, minimizing latency, bandwidth, and energy consumption in large-scale IoE systems.

3.1.1.4 Multi-Agent Systems and Interoperability

In a MAS, multiple agents collaborate to achieve objectives beyond the capability of a single entity. In distributed IoE environments, this is not optional but inherent: information is dispersed across devices and domains, decisions must often be taken close to data sources, and actions frequently occur in parallel. Agents therefore specialize (e.g., monitoring, diagnosis, planning, actuation) and coordinate to pursue system-level goals [144]. Interaction can take different forms. Agents may cooperate by sharing observations and intermediate results to optimize a common objective, refine decisions through structured negotiation by proposing and challenging alternative solutions, or—in competitive settings—represent conflicting stakeholders and reason in adversarial or game-theoretic terms. Coordination may rely on a central orchestrator, emerge from peer-to-peer exchanges, or follow hierarchical structures aligned with policy and role separation.

At scale, interoperability becomes critical. Agents must operate across heterogeneous implementations, vendors, and administrative domains, requiring structured mechanisms to exchange context and advertise capabilities. Such mechanisms enable discovery, intent–operation alignment, and task delegation with explicit assumptions on inputs, outputs, and constraints. General-purpose approaches such as the *Model Context Protocol* aim to standardize context sharing among agents and external services, while lightweight domain-specific descriptors support capability discovery and binding. Complementing these abstractions, several emerging protocol stacks address agent-to-agent and agent-to-system interaction. Decentralized approaches such as the *Agent Network Protocol* combine identity, secure communication, and application-layer exchanges; capability-advertisement schemes like *Agent2Agent* employ machine-readable *Agent Cards*; and conversation-centric designs such as the *Agent Interaction & Transaction Protocol* model communication as threaded interactions across transport layers. Together, these efforts lay the groundwork for scalable, interoperable, and policy-aware multi-agent coordination in large-scale IoE systems.

3.1.2 Distributed Agentic Intelligence for Autonomous IoE

Coordinating massive numbers of devices, services, and data flows in IoE ecosystems requires more than incremental automation. Autonomy must be architected by embedding intelligence into the control plane and distributing it across cloud, edge, and device tiers, as monolithic controllers are neither scalable nor sustainable. Agentic AI provides a principled path forward. LLM-, SLM-, and TinyML-powered agents elevate network management from parameter tuning to goal-driven reasoning: interpreting high-level objectives, resolving constraints, planning multi-step actions, and adapting to evolving contexts.

Implementing this paradigm requires two foundations: predictive validation of candidate actions and semantic expression of desired outcomes. We address these

needs by integrating DTs and IBN into a unified three-tier intelligence architecture (Fig. 3.2). Rather than replacing existing orchestration, distributed agents act as cognitive overlays enhancing DT-based validation and intent-driven control loops, forming a scalable closed-loop IoE fabric.

3.1.2.1 Digital Twins as Predictive Substrates for Autonomous 6G Networks

A DT is a high-fidelity virtual replica of a physical system, continuously synchronized through bidirectional data exchange. This coupling enables real-time monitoring, simulation, and predictive analysis, allowing validated configurations to be safely deployed without service disruption. DTs can mirror distinct domains—RAN, transport, and compute—capturing infrastructure state and operational context. In the RAN, they support analysis of channel dynamics, mobility, interference, and user-perceived quality across heterogeneous spectrum bands (FR1–FR3). Beyond physics-based modeling, DTs integrate heterogeneous data sources structured through formal models and knowledge graphs. Hybrid approaches combining analytical and data-driven methods (e.g., deep learning, reinforcement learning) enable predictive optimization under uncertainty.

Language-model-powered agents further enhance DTs by interpreting unstructured inputs, generating synthetic samples under sparse data, and providing semantic explanations of system dynamics [147]. Conversely, DTs provide a controlled sandbox for training, validating, and stress-testing agents, mitigating risks before deployment. DTs therefore enable safe evaluation of candidate actions and help quantify trade-offs across performance, resilience, and sustainability. This view is consistent with ongoing standardization efforts: 3GPP positions DTs as enablers of automation within Closed-Control Loops (CCLs) [148], while ITU-T discusses architectures combining DTs and language models for dynamic controller synthesis and modular adaptation [149].

3.1.2.2 Enhancing Intent-Based Networking with Agentic AI

Language models are key enablers of IBN, where operators specify high-level intents instead of prescriptive configurations. Intents, often expressed in natural language, capture objectives difficult to encode numerically, including KVIs related to sustainability and resilience. Agentic systems interpret, disambiguate, and translate these intents into structured policies for autonomous enforcement. This translation requires semantic disambiguation, contextual reasoning, and conflict detection. By processing multimodal inputs and telemetry, agents generate context-aware policies that reduce ambiguity and configuration errors. Standardization efforts increasingly reflect this shift. ETSI's ZSM and ENI promote policy-based cognition, where formal rules link intents to operational decisions within AI-driven closed-loop automation [140]. In the ENI architecture, cognition models analyze telemetry, evaluate progress against KPIs, and trigger corrective actions, with reinforcement and transfer learning supporting continuous adaptation. TM Forum architectures similarly segment infrastructures into intent-driven domains governed by Intent Management Functions.

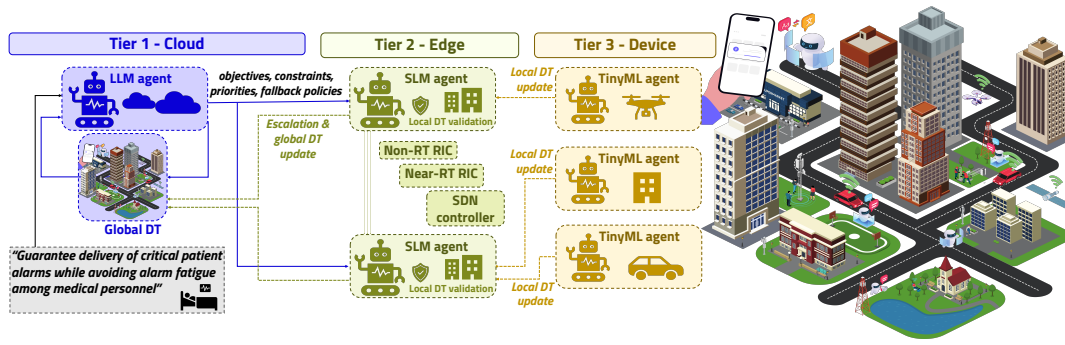


Figure 3.2: Three-tier agentic architecture integrating IBN and DTs for closed-loop IoE autonomy [139].

Beyond translation, language-model-powered agents enable intent assurance by analyzing logs, telemetry, and behavioral traces to verify that deployed configurations satisfy declared objectives, maintaining alignment between expectations and system behavior. Coupled with structured knowledge representations such as knowledge graphs, these agents improve consistency and traceability. In Open-RAN environments, language models translate intents into execution directives for rApps and xApps, enabling real-time, vendor-agnostic orchestration at 6G scale.

3.1.2.3 A Three-Tier Intelligence Architecture for 6G IoE Management

Figure 3.2 summarizes the proposed three-tier architecture, which separates *semantic* (cloud), *operational* (edge), and *perceptual* (device) intelligence, aligning model capacity with control requirements and enabling cross-domain intents to be translated, decomposed, coordinated, validated, and enforced across heterogeneous IoE control domains. The primary novelty of the framework is not the introduction of additional control entities, but the explicit realization of a cross-domain intent-assurance workflow in which high-level objectives are progressively refined into operational actions while maintaining visibility over system-wide objectives, constraints, and trade-offs through DT-assisted validation.

Tier 1 (Cloud): Semantic Intelligence and Global Coordination Tier 1 operates in the cloud and is instantiated by LLM-powered agents responsible for cross-domain reasoning and global coordination. These agents interface with humans and applications by parsing unstructured intents, resolving objective conflicts, and generating structured intent representations (objectives, constraints, priorities, and fallback policies). Intents may originate from human operators, applications, workflow engines, or automated operational procedures and, depending on the source, can be expressed as natural-language requests or as structured events generated by higher-layer systems. Cloud agents translate such inputs into machine-readable objects and, when multiple domains are involved, decompose them into domain-specific sub-intents that can be delegated to specialized operational agents. Tier 1 also maintains a global DT, which performs what-if analyses, validates intent compliance across domains, and recommends configurations that are safe for large-scale deployment; this DT is updated when cross-domain state changes or new policies

are introduced. The global DT provides a common reasoning space where the expected impact of alternative policies can be assessed across communication, sensing, computing, and application domains, enabling the resolution of conflicts that individual domain controllers, operating with only local visibility, cannot address.

Tier 2 (Edge): Operational Intelligence and Domain Actuation Tier 2 consists of domain-specific SLM-powered agents deployed at the edge and Multi-access Edge Computing (MEC), where near-real-time telemetry is available. These agents translate validated intents into coordinated multi-step actions across RAN, transport, and compute domains by interfacing with SDN controllers, RAN Intelligent Controller (RIC) functions, and container orchestration platforms. Depending on the target domain, these actions may correspond to RIC policies, traffic-engineering rules, SDN configurations, resource-allocation decisions, or workload-related policies. To keep the control loop responsive, agents use domain-local DT replicas for low-latency validation and rapid corrective adaptation, and escalate to the global DT only when cross-domain coordination is required. Local decisions are continuously informed by real-time telemetry, including radio indicators (e.g., Channel Quality Indicator (CQI), SINR, spectrum utilization), network measurements (e.g., latency, packet loss, congestion), and compute-resource availability. These measurements update local DT replicas and allow operational agents to evaluate the feasibility and expected impact of candidate actions before deployment; whenever alternative policies generate conflicting outcomes across domains, the corresponding sub-intents are escalated to Tier 1 for global arbitration.

Tier 3 (Device/Physical Domain): Distributed Perception Tier 3 corresponds to the physical IoE domain—people, objects, sensors, and resource-constrained endpoints that continuously shape network dynamics. TinyML-powered agents operate on constrained devices to provide always-on, ultra-low-power inference for anomaly detection, monitoring, and environmental sensing. Rather than continuously forwarding raw measurements, these devices expose compact observations and context information that capture relevant physical-system conditions, supplying ground truth to Tier 2 controllers and local DT replicas while reducing bandwidth usage and enabling scalable perception.

Across the three tiers, intents are structured at Tier 1, assessed via DT-based what-if analyses, enacted by Tier 2 through domain orchestrators, and monitored via Tier 3 sensing. To guarantee safe interaction with operational infrastructures, agent-generated actions are never directly executed by the underlying controllers: candidate actions are first translated into structured policies and validated against controller-specific schemas and DT-based predictions. Policies that violate constraints, fail validation checks, exceed timing requirements, or are predicted to compromise intent satisfaction trigger corrective actions (e.g., fallback policies corresponding to known safe operating conditions) and, when needed, escalation for cross-domain arbitration. This closes the loop over the service lifecycle while respecting latency and energy constraints in dense IoE environments, even when agent outputs are inconsistent, delayed, or unsuitable for direct execution, allowing

otherwise independent controllers and subsystems to cooperate toward common intent-driven objectives.

3.1.3 Case Studies: Agentic Cross-Domain Intent Assurance

To ground the proposed three-tier architecture, we instantiate its roles in two complementary case studies that share the same implementation: an edge-centric Vehicle-to-Infrastructure (V2I) cooperative-perception scenario and a cross-domain hospital smart-alarm scenario. Both follow the same workflow, in which TinyML agents populate local DTs, edge SLM agents perform intent assurance, and cloud LLM agents provide a reference baseline. The goal is not to implement the full architecture, but to validate representative operational slices focused on edge autonomy, sustainability, and cross-domain intent assurance.

All experiments run on macOS 26.3 with Python 3.14. The edge agent is a quantized SLM (qwen2.5:1.5b, Q4_K_M) served locally via Ollama with deterministic decoding (temperature 0, num_predict 150, num_ctx 512, repeat_penalty 1.0) and a fixed JSON-only prompt template, while the cloud baseline uses llama-3.3-70b-versatile via Groq API. All components communicate over an MQTT bus, and a FastAPI Tool Gateway exposes DT services and controller-facing actuation interfaces. The DT implements a closed-form service-level model that maps candidate policies to observable KPIs. Each scenario is executed as an independent closed-loop run of 60 s of simulated time, during which every controller repeatedly observes the current KPI state, decides a candidate policy, and actuates it only if the candidate passes DT validation. To isolate decision quality from actuation frequency, all adaptive agents operate at the same fixed decision cadence, and every metric is averaged over three independent repetitions.

3.1.3.1 V2I Cooperative Perception Scenario

In the first case study, Tier 1 is a cloud-hosted baseline, Tier 2 an Road Side Unit (RSU)-resident SLM controller, and Tier 3 emulated TinyML sensing agents generating lightweight telemetry. Vehicles periodically transmit compact sensing summaries to the RSU over a contention-based ITS-G5-like V2I link [150]. Congestion is handled through distributed mechanisms (e.g., Decentralized Congestion Control (DCC)), making system behavior sensitive to aggregate offered load. This setting exposes the agentic control loop to practical communication constraints, including contention-driven access, variable channel quality, congestion-dependent delay, and protocol-level rate adaptation through DCC-like mechanisms.

At the edge, the SLM agent observes KPIs and selects policies satisfying two intent objectives: (i) a latency target expressed via TTA, and (ii) a bandwidth budget limiting aggregate V2I usage. Policies consist of a reporting period, a message-compression level, and per-class rate limits (safety and bulk traffic), enforced as service-level traffic shaping rather than per-vehicle scheduling. The V2I DT acts as a predictive assurance layer: given the current traffic conditions and a candidate policy, it estimates the expected TTA and aggregate bandwidth, and candidate policies predicted to violate the intent are rejected before enforcement.

We evaluate five stress scenarios: *Moderate Bursts*, *Dense Traffic*, *Flash Crowd*, *Tight Constraints*, and *High Churn*, covering bursty and high-churn traffic conditions. The RSU-resident SLM controller is compared against five alternative mechanisms: the *threshold* controller applies predefined corrective actions when fixed TTA or bandwidth triggers are crossed; the *cloud LLM* returns a complete policy in a single remote inference, embodying maximal reasoning capacity but incurring a round-trip and a large per-decision cost; the *distributed Reinforcement Learning (RL)* controller adapts over the same action space from a reward balancing latency against bandwidth, but without any what-if check; finally, an *O-RAN-style xApp* and a classical *MEC scheduler* provide standards-oriented and heuristic references.

3.1.3.2 Cross-Domain Hospital Smart-Alarm Scenario

The second case study evaluates the architecture in a smart-hospital environment, where intent satisfaction requires coordinating multiple operational domains. Wearable devices monitor patient conditions and generate alarms that must reach healthcare personnel within stringent timing constraints while preserving battery lifetime and avoiding alarm fatigue. Tier 3 is represented by emulated TinyML-enabled wearables, Tier 2 by a ward-level SLM agent, and Tier 1 by a cloud-hosted LLM baseline. The controller must jointly coordinate four interacting domains, adapting wearable sampling frequency, network-priority level, computation placement (edge, hybrid, or cloud), and alarm-escalation policy. A representative intent is: “*Deliver critical patient alarms within the required latency target while minimizing wearable-device energy consumption, limiting alarm fatigue, and maintaining a sensing fidelity of at least 2 Hz.*” For each candidate policy, the hospital DT predicts alarm-delivery latency, battery consumption, alarm fatigue, and sensing fidelity, and rejects any policy predicted to violate a constraint before enforcement.

We evaluate three conditions of increasing stress: *Normal Ward*, *High Acuity*, and *Mass Casualty*. The SLM controller is compared against a *threshold-based multi-domain controller*, a *network-only* baseline, a *distributed RL* baseline without DT coordination, and a *cloud LLM*. The network-only ablation, which can adapt only the message-priority level, is included to expose where single-domain control breaks down.

3.1.3.3 Performance Evaluation

The primary indicators are intent satisfaction (%), i.e., the percentage of intents whose objectives are met over the evaluation window, and the number of reconfigurations, i.e., the total number of policy changes performed by the agent. These are complemented by service-level and sustainability indicators, including TTA (V2I), alarm-delivery latency (hospital), alarm-fatigue rate, wearable battery drain, controller-decision latency t_{dec} , and operational carbon footprint. Controller-decision latency is reported separately from service-level latency because DT validation, whose share is denoted t_{val} , belongs to the control loop rather than to the application data path.

V2I results Across the five scenarios, the edge SLM attains 90–100% intent satisfaction with approximately 10 reconfigurations (Table 3.1). Compared with the threshold controller, the SLM improves intent satisfaction by up to 67.1% under *Tight Constraints* while requiring fewer reconfigurations. It also outperforms the O-RAN xApp and matches or exceeds the distributed RL baseline while using fewer policy changes, highlighting the effect of DT-assisted validation. The MEC scheduler remains competitive on this single-domain task: the SLM matches it in three of five scenarios and trails only under the tightest constraints (90% vs. 96.7%). Compared with the cloud LLM, the edge SLM preserves similar performance in benign conditions while remaining stable under *Flash Crowd* and *Tight Constraints*, where the cloud controller exhibits frequent reconfigurations or complete intent violations. This is visible in the bandwidth allocation (Fig. 3.3): the SLM consistently uses the full budget, allocating more safety bandwidth, which directly explains its higher TTA compliance, and it provides up to 31% more than the O-RAN xApp and 20% more than RL. The resulting mean TTA stays within target across all conditions (Fig. 3.4), unlike the threshold and O-RAN controllers. On overhead, the SLM closes the loop in about 4.6 ms, of which 3.7 ms is DT validation: predictive checking dominates the control latency yet keeps it within a few milliseconds, compatible with near-real-time edge actuation. Finally, reporting the operational footprint only (Fig. 3.5), the edge SLM emits a nearly constant 2.0 mg CO₂e per decision against 42.8 mg for the cloud LLM, a reduction above 90% at comparable decision quality.

Hospital results Under *Normal Ward*, all controllers satisfy the intent, reflecting the relatively relaxed operating conditions (Table 3.2). As stress rises, the network-only ablation collapses from 100% to 48.4% (*High Acuity*) and 51.6% (*Mass Casualty*), because it cannot jointly coordinate sensing, networking, and energy objectives. The distributed RL, although multi-domain, reaches only 66.7–75.0% intent satisfaction despite substantially more reconfigurations. The edge SLM consistently maintains 96.7–100% intent satisfaction with at most one reconfiguration throughout; the gap is largest in *High Acuity*, where it reaches 96.7% against 48.3% (threshold) and 48.4% (network-only). The cloud LLM degrades significantly under higher-stress conditions and emits ≈ 15 mg CO₂e per decision against ≈ 2.0 mg for the edge SLM. On overhead, the SLM closes the loop in 3.8–87.9 ms with only a negligible DT-validation share; the spread tracks the number of policy adaptations actually triggered, and in all cases the decision time stays well below the sub-second alarm targets, confirming compatibility with the clinical workflow. Overall, the two case studies show that edge-resident agentic control matches cloud-level decision quality while substantially improving stability and sustainability, and that DT-assisted cross-domain coordination is essential to sustain intent satisfaction when a single intent spans multiple interacting domains.

3.1.4 Challenges and Research Ahead

While the proposed roadmap highlights the promise of language-model-powered autonomous networks, realizing this vision remains challenging. Key issues span agent architecture and lifecycle management, interoperability across heterogeneous domains, and rigorous validation under realistic operating conditions.

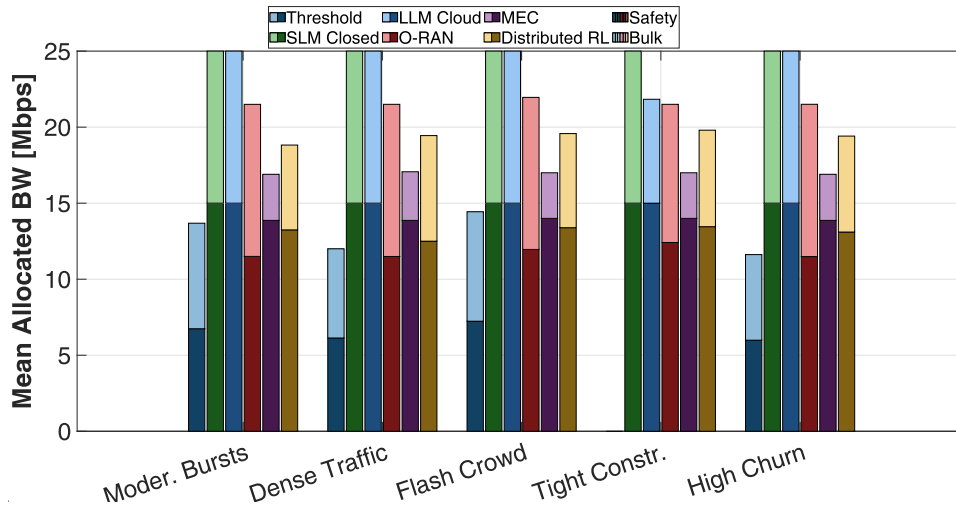


Figure 3.3: Per-class bandwidth allocation across all controllers and stress scenarios [139].

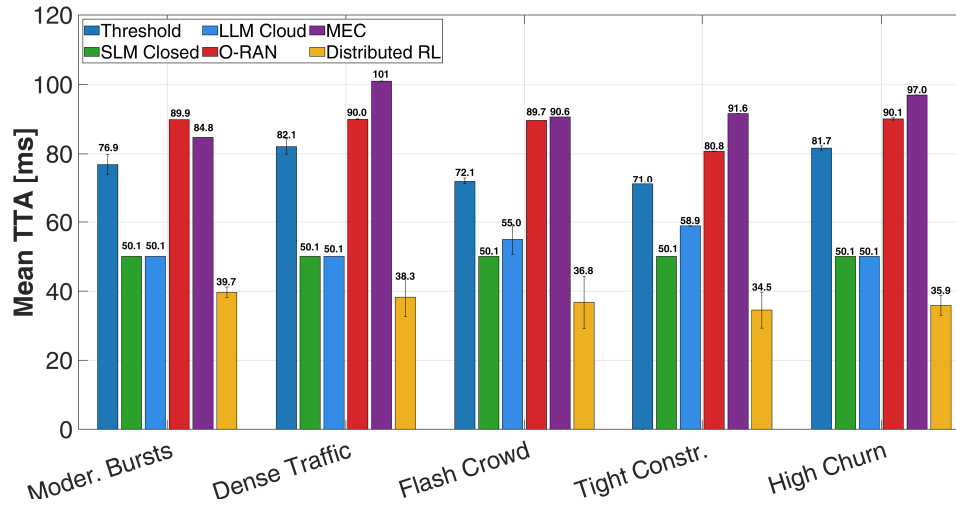


Figure 3.4: Mean TTA across all controllers and stress scenarios [139].

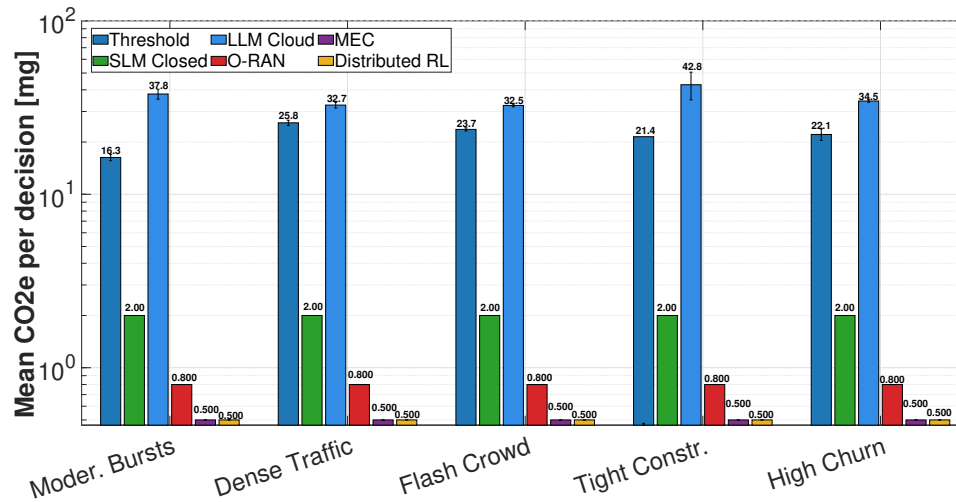


Figure 3.5: Operational carbon footprint per decision [139].

Table 3.1: V2I scenario: per-scenario intent configuration and decision outcomes across all baselines (mean $\pm$ std over three repetitions). Decision latency t_{dec} is the control latency; t_{val} is the DT validation share within it.

Scenario	# Veh.	Intent configuration		Approach	Intent sat. [%]	Reconf. [#]	t_{dec} [ms]	t_{val} [ms]
		TTA [ms]	BW budget [Mbps]					
Moderate Bursts	6	80	30	Threshold	46.1 $\pm$ 5.4	18.0 $\pm$ 3.6	—	—
				SLM (RSU edge)	100.0 $\pm$ 0.0	10.0 $\pm$ 2.0	3.3 $\pm$ 0.2	2.85 $\pm$ 0.27
				LLM (cloud)	100.0 $\pm$ 0.0	20.0 $\pm$ 2.0	—	—
				O-RAN xApp	50.0 $\pm$ 0.0	96.0 $\pm$ 0.0	—	—
				MEC scheduler	100.0 $\pm$ 0.0	2.0 $\pm$ 0.0	—	—
				Distributed RL	84.4 $\pm$ 3.8	15.0 $\pm$ 2.6	—	—
Dense Traffic	12	80	35	Threshold	36.7 $\pm$ 4.4	14.3 $\pm$ 0.6	—	—
				SLM (RSU edge)	100.0 $\pm$ 0.0	10.0 $\pm$ 1.0	4.6 $\pm$ 0.4	3.64 $\pm$ 0.87
				LLM (cloud)	100.0 $\pm$ 0.0	31.0 $\pm$ 5.0	—	—
				O-RAN xApp	50.2 $\pm$ 0.8	95.7 $\pm$ 0.6	—	—
				MEC scheduler	100.0 $\pm$ 0.0	2.0 $\pm$ 0.0	—	—
				Distributed RL	90.0 $\pm$ 5.8	14.0 $\pm$ 2.0	—	—
Flash Crowd	8	70	28	Threshold	52.5 $\pm$ 1.4	30.3 $\pm$ 1.5	—	—
				SLM (RSU edge)	92.0 $\pm$ 0.0	10.0 $\pm$ 1.0	4.1 $\pm$ 0.8	3.72 $\pm$ 0.64
				LLM (cloud)	100.0 $\pm$ 0.0	31.0 $\pm$ 30.5	—	—
				O-RAN xApp	1.0 $\pm$ 0.0	12.0 $\pm$ 0.0	—	—
				MEC scheduler	100.0 $\pm$ 0.0	1.0 $\pm$ 0.0	—	—
				Distributed RL	90.0 $\pm$ 6.7	15.3 $\pm$ 6.7	—	—
Tight Constraints	8	60	22	Threshold	22.9 $\pm$ 1.2	29.0 $\pm$ 0.0	—	—
				SLM (RSU edge)	90.0 $\pm$ 0.0	10.0 $\pm$ 0.0	4.1 $\pm$ 0.2	2.03 $\pm$ 0.31
				LLM (cloud)	0.0 $\pm$ 0.0	91.7 $\pm$ 1.5	—	—
				O-RAN xApp	0.0 $\pm$ 0.0	95.7 $\pm$ 0.6	—	—
				MEC scheduler	96.7 $\pm$ 0.0	1.0 $\pm$ 0.0	—	—
				Distributed RL	88.9 $\pm$ 1.9	11.0 $\pm$ 1.0	—	—
High Churn	10	80	28	Threshold	36.1 $\pm$ 1.6	24.0 $\pm$ 3.6	—	—
				SLM (RSU edge)	100.0 $\pm$ 0.0	10.0 $\pm$ 0.0	4.2 $\pm$ 0.9	3.65 $\pm$ 0.70
				LLM (cloud)	100.0 $\pm$ 0.0	31.0 $\pm$ 2.0	—	—
				O-RAN xApp	50.0 $\pm$ 0.0	94.0 $\pm$ 3.5	—	—
				MEC scheduler	100.0 $\pm$ 0.0	2.0 $\pm$ 0.0	—	—
				Distributed RL	95.2 $\pm$ 1.7	13.3 $\pm$ 2.5	—	—

Table 3.2: Hospital smart-alarm scenario: per-scenario intent configuration and decision outcomes across all baselines (mean $\pm$ std over three repetitions). t_{dec} is the control latency; t_{val} is the DT validation share within it.

Scenario	# Pat.	Intent configuration			Approach	Intent sat. [%]	Reconf. [#]	Alarm [ms]	CO ₂ [mg]	t_{dec} [ms]	t_{val} [ms]
		Alarm [ms]	Bat. [%/h]	Fat. [#h]							
Normal Ward	6	700	6.0	8	Threshold	100.0	1.0	443 $\pm$ 1	0.97 $\pm$ 0.01	—	—
					Network-Only	100.0	1.0	443 $\pm$ 1	0.51	—	—
					Distributed RL	97.8 $\pm$ 3.7	19.0 $\pm$ 1.7	306 $\pm$ 13	1.01	—	—
					SLM (edge)	100.0	0.0	258	2.00	3.8 $\pm$ 0.1	3.0 $\pm$ 0.30
					LLM (cloud)	100.0	3.0	438 $\pm$ 3	15.06 $\pm$ 0.01	—	—
High Acuity	10	350	5.0	10	Threshold	48.3	29.0	315 $\pm$ 7	0.97	—	—
					Network-Only	48.4	31.0	274	0.51	—	—
					Distributed RL	66.7 $\pm$ 6.7	24.3 $\pm$ 2.1	265 $\pm$ 33	1.01	—	—
					SLM (edge)	96.7	1.0	235	2.12	84.8 $\pm$ 0.8	0.10 $\pm$ 0.05
					LLM (cloud)	0.0	3.0	429 $\pm$ 1	15.10	—	—
Mass Casualty	20	450	4.5	16	Threshold	55.2 $\pm$ 6.0	29.0	392 $\pm$ 1	0.96 $\pm$ 0.01	—	—
					Network-Only	51.6 $\pm$ 5.9	30.3 $\pm$ 0.6	339 $\pm$ 2	0.51	—	—
					Distributed RL	75.0 $\pm$ 5.0	25.3 $\pm$ 1.5	318 $\pm$ 15	1.01	—	—
					SLM (edge)	96.7	1.0	283 $\pm$ 1	2.13	87.9 $\pm$ 2.6	0.14 $\pm$ 0.11
					LLM (cloud)	40.9 $\pm$ 7.8	3.0	459 $\pm$ 4	15.13	—	—

Cross-cutting concerns, discussed in this section, including trust, privacy, security, sustainability, and human oversight, further complicate large-scale deployment. Achieving seamless and full-fledged integration of Agentic AI in autonomous IoE networks now requires progress beyond model capability alone. Key research priorities include (i) resource-aware agent deployments that balance reasoning quality with latency and energy constraints, (ii) standardized and interpretable evaluation methods that assess not only outputs but also reasoning robustness and closed-loop behavior, (iii) sustainability-aware orchestration and carbon-efficient inference across tiers, (iv) verifiable and transparent decision-making to support explainability, accountability, and human oversight, and (v) secure and privacy-preserving multi-agent interoperability across heterogeneous domains. Aligning these priorities with ongoing standardization and industry initiatives will accelerate the transition toward intelligent, resilient, and trustworthy autonomous IoE networks.

3.1.4.1 Design and Deployment

A central barrier to deployment lies in the storage, memory, and compute demands of modern models. With billions of parameters, state-of-the-art LLMs are difficult to host on edge nodes or lightweight network elements, particularly in the absence of hardware accelerators (e.g., Graphics Processing Units (GPUs), Tensor Processing Units (TPUs)). Their data-driven nature also makes performance sensitive to training quality, raising concerns about bias, representativeness, and domain generalization.

Architecturally, adopting MASs introduces additional complexity [151]. Agents must securely discover one another, negotiate communication channels, and maintain lifecycle consistency across domains. Designing reusable and verifiable agent workflows remains immature, while risks such as ambiguous task specifications, inefficient communication, lack of verification pipelines, and cascading hallucination errors persist. Mitigation requires structured safeguards. *Verifier agents*, persistent memory, and state-tracking modules help maintain contextual coherence. *Uncertainty quantification* enables adaptive escalation to human or higher-tier agents. Domain-specific verification strategies—ranging from systematic software test coverage to symbolic reasoning—enhance correctness. Reinforcement learning supports adaptive refinement, while *confidence thresholding* restricts execution to sufficiently certain decisions, a crucial safeguard in mission-critical IoE domains such as healthcare, finance, and public safety.

3.1.4.2 Performance Metrics

Evaluating agentic systems is inherently complex: outputs vary in structure, reasoning depth, and operational impact, making manual inspection infeasible at scale. Standardized, automated metrics are therefore essential. Traditional metrics such as Bilingual Evaluation Understudy (BLEU) and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) measure lexical overlap but fail to capture reasoning validity or factual grounding. Semantic similarity metrics (e.g., BERTScore, MoverScore, Semantic Matching Score (SMS)) assess meaning preservation, though with limited

task adaptability. Reference-free approaches (e.g., SUPERT, ROUGE-C) evaluate informativeness without ground truth, while tools such as FactCC and Dependency Arc Entailment (DAE) detect factual inconsistencies. For agentic systems, evaluation must extend beyond final outputs to intermediate reasoning and tool usage. Core indicators include task success rate, *action advancement* across subtasks, and accuracy of tool invocation (e.g., correct API or Structured Query Language (SQL) calls). Robustness metrics assess degradation under noisy, adversarial, or perturbed inputs.

Efficiency and sustainability metrics are equally critical in IoE contexts, capturing compute utilization, tokens-per-second throughput, energy-per-token, and environmental cost. Concepts such as green capacity [152] integrate accuracy, latency, and ecological footprint into unified performance indicators. When integrated with DTs and IBN, evaluation must also measure twin fidelity, synchronization accuracy, and closed-loop assurance effectiveness. Metrics such as twinning fidelity and intent satisfaction rate quantify the alignment between digital replicas, deployed configurations, and declared objectives, providing end-to-end validation of the agentic control loop.

3.1.4.3 Evaluation Strategies and Benchmarks

Beyond static metrics, evaluating agentic systems requires dynamic strategies that capture reasoning, adaptability, and interaction. Two dominant paradigms are commonly adopted:

- *LLM-as-a-judge*, where one model evaluates another (or itself) through preference modeling or relative scoring;
- *Task-based benchmarking*, relying on standardized datasets for translation, summarization, information extraction, and question answering.

However, agentic evaluation must decompose performance into core competencies.

Knowledge Accuracy Hallucination remains a persistent limitation. Evaluation should measure factual consistency against Retrieval-Augmented Generation (RAG)-based baselines or open-domain Question Answering (QA) datasets, ensuring alignment with verifiable knowledge sources.

Reasoning Capabilities Structured inference demands fine-grained assessment across mathematical, logical, and procedural reasoning, as well as code generation and planning. Benchmarks such as GSM8K, MATH, MultiPL-E, BigCodeBench, and composite suites (e.g., Finetuned Language Net (FLAN), AlpacaEval, Arena-Hard-Auto) evaluate reasoning depth under automated or hybrid human–model scoring.

Multi-Modal Understanding With multimodal LLMs, evaluation extends beyond text to visual and structured inputs, including visual QA (e.g., GQA, VQAv2), document interpretation (DocVQA, AI2D), and spatial reasoning (NLVR2). While standardized leaderboards improve reproducibility, static datasets cannot capture

the dynamics of agents leveraging tools, memory, and contextual feedback. Future evaluation frameworks must therefore incorporate interactive environments and simulation testbeds that assess real-time adaptability, reasoning under uncertainty, and closed-loop decision-making—particularly in distributed IoE scenarios.

3.1.4.4 Sustainable Intelligence: Green Challenges

Sustainability is becoming a defining constraint for LLM-powered agents, as inference increasingly dominates lifecycle energy consumption. Unlike training, which is episodic, large-scale inference introduces persistent operational emissions, especially in globally deployed systems.

The environmental footprint stems from both hardware and usage patterns. Accelerators (e.g., GPUs, TPUs) incur significant operational and embodied carbon costs, while token-by-token generation makes LLM inference one to two orders of magnitude more energy-intensive per request than traditional AI or rule-based solutions. Workload relocation to greener grids can reduce emissions but may increase latency, exposing trade-offs between sustainability and performance. Mitigation strategies span hardware, system, and algorithmic levels. Quantization can reduce inference energy and latency by 60–70% for robust tasks, though with accuracy trade-offs in math or code reasoning. Edge-deployed quantized SLMs offer a promising intermediate solution in distributed IoE settings, balancing responsiveness and carbon footprint.

Additional approaches include power capping, carbon-aware scheduling, real-time cost estimation, and early-exit mechanisms that halt inference once confidence thresholds are reached. The previously mentioned green capacity [152] integrates accuracy, latency, and ecological impact into a unified metric, supporting more holistic model comparison. Embedding sustainability into agent design—from model selection to tier placement—will be essential for aligning large-scale Agentic AI adoption with global climate objectives.

3.1.4.5 Explainability and Trustworthiness

As language-model-powered agents extend into sensitive domains such as healthcare, finance, and public policy, explainability becomes foundational to trust. Without transparency, users, operators, and regulators cannot validate agent behavior.

Current approaches rely on verbal rationales, structured prompting (e.g., chain-of-thought, logic-tree generation), and justification outputs. While these improve interpretability, they may not faithfully reflect internal reasoning, limiting their reliability in operational settings. Moreover, added latency can be problematic in real-time control loops.

In multi-agent environments, explainability must also cover inter-agent interactions, negotiations, and dependency chains, where errors can propagate across tiers. Research in Explainable Artificial Intelligence (XAI) offers promising directions: methods such as SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) provide localized explanations, while emerging techniques target distributed and multi-modal systems. Effective frameworks

must balance fidelity (faithfulness to internal reasoning), usability (operator comprehension), and efficiency (minimal overhead). Ultimately, explainability is not only a technical feature but a prerequisite for accountability, regulatory compliance, and ethical deployment in IoE ecosystems.

3.1.4.6 Ethical, Privacy, and Security Implications

Embedding language models into network control planes introduces complex ethical, privacy, and security risks. Prompt injection and jailbreaking can manipulate outputs and bypass safeguards, while data poisoning and backdoors threaten reliability. Training on large corpora also raises concerns about memorization and leakage of sensitive information, exacerbated by membership inference and data reconstruction attacks. Bias amplification remains a critical ethical issue, as outputs may reflect and reinforce demographic or cultural imbalances in training data. Questions of authorship, provenance, and accountability further complicate large-scale deployment.

Mitigation strategies include differential privacy, federated learning with secure aggregation, encrypted inference, and secure execution environments. Guidelines such as the OWASP Top 10 for LLMs recommend secure output handling, fine-grained access control, and sandboxed plugin execution [153]. Ensuring resilience in IoE settings requires integrating adversarial robustness, compliance mechanisms, and continuous monitoring across both architectural and operational layers of agentic systems.

3.2 Trustworthiness Dissemination in Social IoE Architectures

The second contribution [154] addresses trustworthiness dissemination as a foundational enabler of distributed orchestration in large-scale IoE environments. While the previous contribution focused on organizing agentic intelligence across cloud, edge, and device tiers, effective autonomous operation also depends on the ability of interconnected entities to establish, maintain, and propagate reliable trust relationships at scale. Without timely and accurate trustworthiness information, autonomous agents cannot safely delegate tasks, discover services, or coordinate decisions across heterogeneous domains. This contribution directly targets this gap by proposing a scalable, responsive architecture for trust aggregation, updating, and dissemination in SLoE systems.

The SLoE integrates people, processes, data, and IoT devices to enhance the utility of network interactions, enabling novel network experiences and creating significant economic opportunities [155]. This paradigm has progressively influenced a wide range of application domains—including healthcare, smart homes, and smart manufacturing—facilitated by the widespread connectivity provided by 5G networks. By leveraging the concept of social networks, the SLoE has the potential to enhance interactions, particularly in terms of security and trust, thereby addressing prevalent challenges in IoT technology such as user visibility and the effective discovery of resources and services. Research in this field explores trust management

and evaluation techniques aimed at establishing reliable relationships and securing communications, thereby enabling trustworthy information sharing [156].

3.2.1 Problem Statement

In a SLoE environment, interconnected people, processes, and devices—referred to as entities—are endowed with social awareness, enabling them to autonomously establish social relationships. In this context, ensuring the availability and continuous updating of entities' trustworthiness has become increasingly challenging in large-scale networks. The mobility and frequent state transitions of entities, such as sleeping, waking up, or leaving and rejoining networks, further exacerbate this challenge. Therefore, maintaining scalability and a high level of responsiveness is crucial to support billions of entities that interact, share resources, and provide services, while ensuring access to up-to-date information about the most trustworthy service providers [157].

Related works in the scientific literature have proposed blockchain technology as a viable solution for developing effective frameworks to disseminate essential information within the social network of entities [158]. However, blockchain-based solutions are not well-suited to handle large volumes of data and suffer from high latency and limited adaptability to dynamic network changes. These limitations are particularly pronounced in B5G scenarios, where entity density, mobility, and service dynamism place stringent demands on the underlying trust infrastructure. Consequently, scalability and responsiveness in disseminating and maintaining trustworthiness information remain significant open challenges [159].

To address this gap, new strategies are needed for the responsive aggregation, updating, and dissemination of trustworthiness information in SLoE systems—strategies capable of handling the scale and dynamism inherent to future IoE deployments without relying on centralized coordination or high-latency consensus mechanisms.

3.2.2 Proposed Framework

To advance the scientific literature, this work proposes an architecture that leverages edge-fog computing. By combining local processing at the network edge with fog computing resources, this approach aims to optimize performance and scalability for distributed applications and services. Furthermore, the proposed design incorporates two levels of clustering, as illustrated in Fig. 3.6.

At the first level, a social entity can represent the digital counterpart of a physical entity. Depending on their application context, these digital counterparts can join service-based communities, thereby enhancing network navigability and reducing latencies in the service discovery process. Each community is managed by an edge node, which is responsible for computing and storing the trust values of its members and, if necessary, excluding malicious entities through trust management procedures. The second level is instead based on geographical position and is managed by Primary Fog nodes, which handle data collection for a territorial cluster. This two-level structure decouples service-oriented trust management from

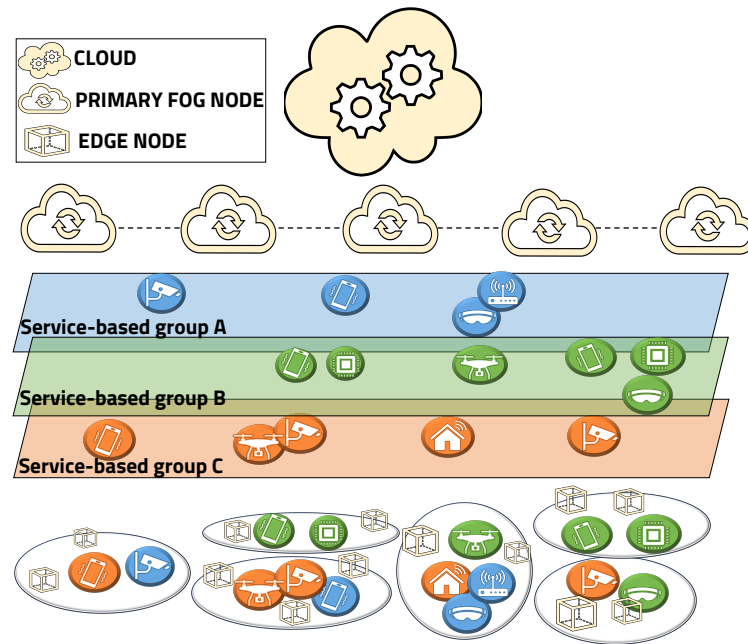


Figure 3.6: The proposed double-clustered architecture [154].

location-aware coordination, enabling each layer to operate independently while remaining synchronized through a shared communication fabric.

Furthermore, this work defines a framework for sharing trustworthiness information between independent territorial clusters, enhancing the system's scalability while improving the responsiveness required by B5G networks. Specifically, the proposed framework eliminates the need for a centralized services orchestrator with comprehensive knowledge of the network topology or the trustworthiness of each entity. Instead, this responsibility is delegated to Primary Fog nodes distributed across different geographical areas, which independently manage trust state for their respective territories and coordinate through a lightweight inter-cluster communication mechanism.

To this end, this work introduces a novel publisher/subscriber design pattern—MQTT-based—to facilitate fog-cloud communication for information sharing among independent clusters. This approach ensures service continuity by accommodating the mobility of entities between territorial clusters, enabling seamless trust propagation without requiring global state consistency or centralized arbitration. The message flow of the designed model unfolds as follows:

- The moving entity attempts to join a service community managed by an edge node in a new territorial cluster.
- To initiate this process, the Primary Fog node sends a “*trust discovery request*” via the cloud to retrieve the entity's trustworthiness information.
- The cloud creates a query for the entire network by contacting the broker and publishing the request.
- All Primary Fog nodes that hold information related to the service community of the searched entity relay the requested trustworthiness data.

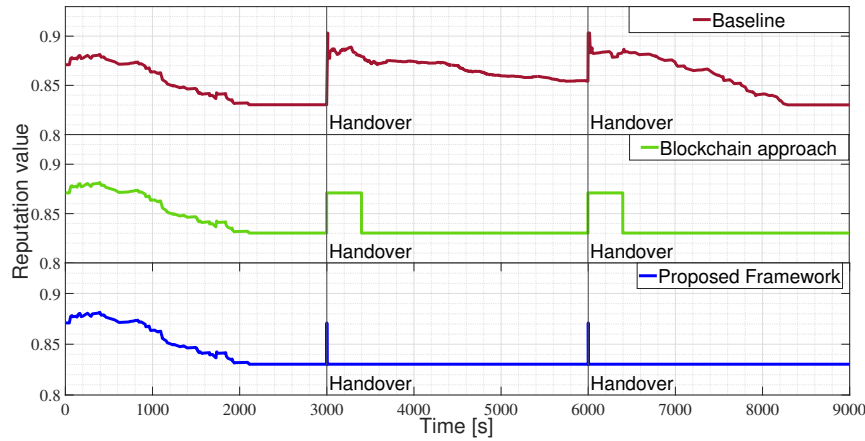


Figure 3.7: Entities' reputation values propagation [154].

- The cloud waits for and collects all data received from edge nodes, as signaled by the appropriate Primary Fog nodes, and aggregates this into synthetic data representing the entity's consolidated trust profile.
- The Primary Fog node that initiated the “*trust discovery request*” receives the updated trust value and assigns a degree of trust to the moving entity, based on the aggregated network knowledge.

This message exchange pattern avoids the need to maintain centralized relational databases with billions of records, where query processing times may return trust values that no longer reflect the current network state. Instead, the proposed approach provides real-time, responsive trust data by querying the distributed fog layer on demand, ensuring that trust decisions are always grounded in up-to-date and network-wide knowledge.

3.2.3 Preliminary Evaluation

Computer simulations are conducted using a C++ simulator to evaluate the system's performance in terms of information-sharing responsiveness. To assess the effectiveness of the proposed methodology, its performance is compared with two benchmark approaches: the baseline and blockchain-based approaches, as shown in Fig. 3.7. In the baseline approach, each cluster operates independently, without any synchronization mechanism across territorial boundaries. In the blockchain-based approach, data sharing is achieved through a transaction process that incurs a mandated delay due to the Proof of Work (PoW) protocol, which limits responsiveness in dynamic and large-scale SIoE environments.

Fig. 3.7 demonstrates that the proposed methodology outperforms both alternatives in terms of efficiency and responsiveness. Specifically, compared to the baseline approach, the proposed framework prevents malicious entities from resetting their reputation during handover—a critical vulnerability in mobility-prone IoE deployments. Additionally, it achieves lower synchronization latencies between clusters compared to the blockchain-based approach, confirming that the MQTT-based publisher/subscriber design is better suited to the scalability and real-time requirements of B5G networks.

Concluding Remarks

This chapter extended the orchestration perspective developed in the previous chapters toward distributed, trustworthy, and agentic forms of intelligence suitable for future IoE environments. The discussion showed that increasingly heterogeneous and decentralized infrastructures require orchestration mechanisms capable of combining semantic intent interpretation, predictive context awareness, and autonomous reasoning across the cloud–edge–device continuum.

To address this evolution, the chapter first analyzed the foundations of agentic AI, emphasizing the complementary roles of LLM-, SLM-, and TinyML-based agents in enabling hierarchical reasoning and localized autonomy. It then introduced a three-tier orchestration architecture integrating DTs, IBN, and distributed agentic control, highlighting how semantic intent handling and predictive validation can support adaptive yet trustworthy orchestration. Finally, the chapter presented a decentralized trustworthiness dissemination framework for SLoE systems, showing how secure and cooperative interactions among distributed entities can be maintained even under dynamic and partially decentralized operational conditions.

Indeed, the chapter demonstrates that future IoE orchestration cannot rely exclusively on centralized optimization pipelines, but instead requires coordinated ecosystems of reasoning agents capable of balancing autonomy, trustworthiness, scalability, and sustainability within highly dynamic environments.

Conclusions and Future Works

This thesis has investigated advanced service orchestration for intent-based networks along three tightly connected research directions: the secure translation of high-level intents into value-aware orchestration actions, the integration of algorithmic intelligence within DT-assisted control frameworks, and the evolution toward distributed and trustworthy orchestration for future IoE environments. These directions constitute a coherent perspective in which architectural design, algorithmic decision support, and trust-aware autonomy converge into a unified orchestration capability for next-generation networks.

The first research direction addresses how high-level intents can be securely translated into orchestrable network actions that jointly satisfy both KPIs and KVIs. A reference architecture for intent-based networks was introduced, comprising a secure intent admission layer and a value-driven intent translation and orchestration mechanism. The secure admission layer integrates a Rasa-based intent profiling chatbot with a MID module, enabling the detection of malicious or adversarial intent formulations before they propagate into the operational control plane. The value-driven orchestration component formalizes the joint satisfaction of performance objectives and broader social and ethical values through a bi-objective optimization framework. The TETREES solver provides a structured methodology for Pareto-front exploration under varying network conditions, enabling operators to navigate trade-offs between network quality, sustainability, trustworthiness, and inclusiveness. The GreenTent framework further instantiates this logic in a sustainability-oriented case study, demonstrating that carbon efficiency and perceived video quality can be jointly managed through intent-driven orchestration.

The second research direction explores how DTs can be endowed with intelligence to provide trustworthy, context-aware, and effective decision support for network orchestration. Two complementary contributions are presented. The first introduces a MCDM-based algorithm leveraging the TOPSIS methodology, embedded within a NDT framework, for the dynamic selection of network domains in post-disaster scenarios. By combining QoS, social attributes, and trust parameters, the proposed approach consistently outperforms conventional baseline solutions in terms of overloaded data and missed video acquisitions, particularly as domain unavailability increases. The second contribution formalizes a more comprehensive architecture for KVI optimization through DT-aided orchestration, introducing a system model grounded in the IBN framework for Network 2030. A DAA-based matching mechanism enables the joint resolution of KPI and KVI objectives at the service request level, with results confirming superior SLA adherence, matching satisfaction, and value-oriented performance under heterogeneous request configurations. The practical relevance of these contributions is further consolidated through

the COHERENT project implementation, which validates a cloud-to-edge DTN architecture through the AAS and demonstrates measurable energy savings at the RAN level via LLM-assisted energy management.

The third research direction examines how secure, autonomous, and distributed orchestration can be achieved in last-mile and future IoE-oriented environments. Two contributions are developed. The first investigates an Agentic AI architecture in which LLM-, SLM-, and TinyML-powered agents are organized across a hierarchical three-tier structure, integrated with DTs for predictive action validation and IBN for semantic intent translation. A V2I case study instantiates this architecture and demonstrates that edge-resident SLM agents, fed by TinyML-generated telemetry, achieve competitive intent satisfaction with significantly lower per-decision carbon footprints than cloud-centric alternatives. The second contribution addresses trustworthiness dissemination in SloE environments, proposing a double-clustered architecture in which Primary Fog nodes independently manage local trust state and coordinate through an MQTT-based publisher/subscriber inter-cluster protocol. The framework eliminates the dependency on a centralized orchestrator, ensures service continuity under entity mobility, and achieves lower synchronization latencies than blockchain-based alternatives, establishing a scalable foundation for trust-aware autonomous orchestration in B5G networks.

These contributions demonstrate that effective orchestration in intent-based networks requires the concurrent satisfaction of three properties: the ability to securely interpret and formalize high-level intents in value-aware terms, the availability of intelligent decision-support mechanisms grounded in synchronized network representations, and the capacity to coordinate distributed entities in a trustworthy and autonomous manner. The experimental and numerical results consistently confirm that jointly addressing these properties yields measurable improvements over state-of-the-art baselines across diverse network scenarios and application domains.

Future research activities will extend and deepen the contributions presented in this thesis along several directions. First, the secure intent translation pipeline will be further developed by investigating the use of LLMs as semantic front-end for intent acquisition, with dedicated validation and hallucination-mitigation mechanisms to ensure that natural-language-derived intents remain operationally coherent and verifiable before enforcement. Second, the DT-assisted orchestration framework will be extended toward multi-domain and cross-administrative deployments, where DTs must operate under partial observability, synchronization delays, and heterogeneous data models. Third, the Agentic AI framework will be further validated through experimental testbed deployments, exploring the behavior of LLM- and SLM-powered agents under real network conditions, including dynamic radio environments, edge hardware constraints, and adversarial interaction scenarios. Finally, all proposed solutions will be evaluated in the context of emerging 6G standardization activities, with particular focus on zero-touch management, non-terrestrial network integration, and the convergence of communication, computation, and sensing in pervasive IoE deployments.

Bibliography

- [1] L. Pintor, L. Atzori, and A. Iera, *Sustainability in telecommunication networks and key value indicators: A survey*, 2025. arXiv: [2501.04356 \[cs.NI\]](https://arxiv.org/abs/2501.04356). [Online]. Available: <https://arxiv.org/abs/2501.04356>.
- [2] W. Mao et al., “Beyond iot: Agi as a transformative solution for the internet of everything and relationship explosion,” *IEEE Internet of Things Journal*, vol. 13, no. 6, pp. 10 339–10 347, 2026. doi: [10.1109/JIOT.2025.3649804](https://doi.org/10.1109/JIOT.2025.3649804).
- [3] A. Mesodiakaki, A. Rahman, et al., “Towards sustainable networks,” in *Towards Natively Intelligent Networks*, Now Publishers, 2023, ch. 6, pp. 233–242. doi: [10.1561/9781638282396.ch6](https://doi.org/10.1561/9781638282396.ch6).
- [4] N. United, “Sustainable development goals. transforming our world: The 2030 agenda for sustainable development,” United Nations, New York, Tech. Rep., 2015. [Online]. Available: <https://sustainabledevelopment.un.org/topics>.
- [5] C. Pignataro et al., *Environmental sustainability terminology and concepts for networking*, IETF Internet-Draft, draft-pignataro-green-enviro-sust-terminology-02, Work in progress, 2026. [Online]. Available: <https://datatracker.ietf.org/doc/draft-pignataro-green-enviro-sust-terminology/>.
- [6] C. Pignataro, S. Zandi, A. Clemm, and J. T. et al., *Environmental sustainability considerations*, IETF Internet-Draft, draft-pignataro-enviro-sustainability-consid-03, Work in Progress, Nov. 2024.
- [7] C. Pignataro, S. Zandi, and A. C. et al., *Terminology for green and environmental sustainability in networking*, IETF Internet-Draft, draft-pignataro-green-enviro-sust-terminology-03, Work in Progress, Nov. 2024.
- [8] C. Pignataro et al., “Architectural Considerations for Environmental Sustainability,” Internet Engineering Task Force, Internet-Draft draft-pignataro-enviro-sustainability-architecture-03, Nov. 2025, Work in Progress, 15 pp.
- [9] 6. I. Association et al., “What societal values will 6g address? 6g infrastructure association vision and societal challenges working group,” 6G SNS IA, Tech. Rep., 2022.
- [10] Ö. U. Akgül et al., “Sustainable 6g architecture: An organic evolution of 5g networks,” *Computer Networks*, vol. 271, p. 111 629, 2025, issn: 1389-1286. doi: <https://doi.org/10.1016/j.comnet.2025.111629>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1389128625005961>.

- [11] S. Mihai et al., "Digital twins: A survey on enabling technologies, challenges, trends and future prospects," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2255–2291, 2022.
- [12] M. Manalastas, M. Umar Bin Farooq, S. Muhammad Asad Zaidi, H. Naeem Qureshi, Y. Sambo, and A. Imran, "From simulators to digital twins for enabling emerging cellular networks: A tutorial and survey," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 4, pp. 2693–2732, 2025. doi: [10.1109/COMST.2024.3490178](https://doi.org/10.1109/COMST.2024.3490178).
- [13] C.-X. Wang et al., "On the road to 6g: Visions, requirements, key technologies, and testbeds," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 905–974, 2023.
- [14] C. D. Alwis et al., "Survey on 6g frontiers: Trends, applications, requirements, technologies and future research," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 836–886, 2021. doi: [10.1109/OJCOMS.2021.3071496](https://doi.org/10.1109/OJCOMS.2021.3071496).
- [15] K. Ojutkangas, E. Rossi, and M. Matinmikko-Blue, "A deep dive into the birth process of linking 6g and the un sdgs," *Telecommunications Policy*, vol. 46, no. 1, p. 102 283, 2022, issn: 0308-5961. doi: <https://doi.org/10.1016/j.telpol.2021.102283>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0308596121001877>.
- [16] B. Masood Khorsandi et al., "Expanded 6g vision, use cases and societal values – including aspects of sustainability, security and spectrum," A Flagship for B5G6G vision, intelligent fabric of technology enablers connecting human, physical, and digital worlds, Deliverable 1.1, 2021. [Online]. Available: <https://hexa-x.eu/deliverables/>.
- [17] L. Atzori, C. Campolo, A. Iera, and G. Morabito, "Toward the ethicnet: Challenges and enablers for ethics-aware networks," *IEEE Communications Magazine*, vol. 61, no. 11, pp. 192–198, 2023. doi: [10.1109/MCOM.004.2300110](https://doi.org/10.1109/MCOM.004.2300110).
- [18] L. Mohjazi, B. Selim, M. Tatipamula, and M. A. Imran, "The journey toward 6g: A digital and societal revolution in the making," *IEEE Internet of Things Magazine*, vol. 7, no. 2, pp. 119–128, 2024. doi: [10.1109/IOTM.001.2300119](https://doi.org/10.1109/IOTM.001.2300119).
- [19] G. Wikström et al., "Key value indicators: A framework for values-driven next-generation ict solutions," *Telecommunications Policy*, vol. 48, no. 6, p. 102 778, 2024, issn: 0308-5961. doi: <https://doi.org/10.1016/j.telpol.2024.102778>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0308596124000752>.
- [20] A. Clemm, L. Ciavaglia, L. Z. Granville, and J. Tantsura, *Intent-Based Networking - Concepts and Definitions*, RFC 9315, Oct. 2022. doi: [10.17487/RFC9315](https://doi.org/10.17487/RFC9315). [Online]. Available: <https://www.rfc-editor.org/info/rfc9315>.

- [21] A. Leivadeas and M. Falkner, "A survey on intent-based networking," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 625–655, 2023. doi: [10.1109/COMST.2022.3215919](https://doi.org/10.1109/COMST.2022.3215919).
- [22] K. Mehmood, K. Kravlevska, and D. Palma, "Intent-driven autonomous network and service management in future cellular networks: A structured literature review," *Computer Networks*, vol. 220, p. 109 477, 2023, issn: 1389-1286. doi: <https://doi.org/10.1016/j.comnet.2022.109477>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1389128622005114>.
- [23] ITU-T, "Intent-based network management and orchestration for network slicing in IMT-2020 networks and beyond," ITU, Recommendation Y.3161, Dec. 2023.
- [24] A. S. Jacobs, R. J. Pötscher, R. A. Ferreira, and L. Z. Granville, "Refining network intents for self-driving networks," *ACM SIGCOMM Computer Communication Review*, vol. 48, no. 5, pp. 55–63, Jan. 2019. doi: [10.1145/3310165.3310173](https://doi.org/10.1145/3310165.3310173).
- [25] A. S. Jacobs et al., "Hey, Lumi! Using Natural Language for Intent-Based Network Management," in *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, USENIX Association, Jul. 2021, pp. 625–639.
- [26] E. El-Rif, A. Leivadeas, and M. Falkner, "Intent Expression Through Natural Language Processing in an Enterprise Network," in *2023 IEEE 24th International Conference on High Performance Switching and Routing (HPSR)*, 2023, pp. 1–6. doi: [10.1109/HPSR57248.2023.10148046](https://doi.org/10.1109/HPSR57248.2023.10148046).
- [27] R. Caldelli, P. Castoldi, M. Gharbaoui, B. Martini, M. Matarazzo, and F. Sciarone, "On helping users in writing network slice intents through NLP and User Profiling," in *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)*, 2023, pp. 545–550. doi: [10.1109/NetSoft57336.2023.10175400](https://doi.org/10.1109/NetSoft57336.2023.10175400).
- [28] M. Gharbaoui et al., "Integrating rasa chatbot with an intent layer for sdn-based network slice provisioning," in *2024 15th International Conference on Network of the Future (NoF)*, 2024, pp. 219–223. doi: [10.1109/NoF62948.2024.10741478](https://doi.org/10.1109/NoF62948.2024.10741478).
- [29] S. Ji, X. Li, P. Zhang, K. Liu, H. Dong, and Y. Zhang, "Intent-Driven QoS Control in Cloud-Network Integration Environment," in *2022 IEEE International Conference on Satellite Computing (Satellite)*, 2022, pp. 52–53. doi: [10.1109/Satellite55519.2022.00019](https://doi.org/10.1109/Satellite55519.2022.00019).
- [30] J. Lin, K. Dzeparoska, A. Tizghadam, and A. Leon-Garcia, "AppleSeed: Intent-Based Multi-Domain Infrastructure Management via Few-Shot Learning," in *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)*, 2023, pp. 539–544. doi: [10.1109/NetSoft57336.2023.10175410](https://doi.org/10.1109/NetSoft57336.2023.10175410).
- [31] H. Ma et al., "Uint: An intent-based adaptive routing architecture," *Computer Networks*, vol. 257, p. 110 991, 2025, issn: 1389-1286. doi: <https://doi.org/10.1016/j.comnet.2024.110991>.

- [32] K. Dzevaroska, J. Lin, A. Tizghadam, and A. Leon-Garcia, "Llm-based policy generation for intent-based management of applications," in *2023 19th International Conference on Network and Service Management (CNSM)*, vol. 33, IEEE, Oct. 2023, pp. 1–7. doi: [10.23919/cnsm59352.2023.10327837](https://doi.org/10.23919/cnsm59352.2023.10327837). [Online]. Available: <http://dx.doi.org/10.23919/CNSM59352.2023.10327837>.
- [33] C. Wang, M. Scazzariello, A. Farshin, D. Kostic, and M. Chiesa, *Making network configuration human friendly*, 2023. arXiv: [2309.06342](https://arxiv.org/abs/2309.06342) [cs.NI]. [Online]. Available: <https://arxiv.org/abs/2309.06342>.
- [34] C. Wang, M. Scazzariello, A. Farshin, S. Ferlin, D. Kostić, and M. Chiesa, "Netconfeval: Can llms facilitate network configuration?" *Proc. ACM Netw.*, vol. 2, no. CoNEXT2, Jun. 2024. doi: [10.1145/3656296](https://doi.org/10.1145/3656296). [Online]. Available: <https://doi.org/10.1145/3656296>.
- [35] I. Chatzistefanidis, A. Leone, and N. Nikaein, "Maestro: Llm-driven collaborative automation of intent-based 6g networks," *IEEE Networking Letters*, pp. 1–1, 2024. doi: [10.1109/LNET.2024.3503292](https://doi.org/10.1109/LNET.2024.3503292).
- [36] K. Abbas, A. Nauman, M. Bilal, J.-H. Yoo, J. W.-K. Hong, and W.-C. Song, "Ai-driven data analytics and intent-based networking for orchestration and control of b5g consumer electronics services," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 2155–2169, 2024. doi: [10.1109/TCE.2023.3324010](https://doi.org/10.1109/TCE.2023.3324010).
- [37] R. Mondal, A. Tang, R. Beckett, T. Millstein, and G. Varghese, "What do llms need to synthesize correct router configurations?" In *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*, ser. HotNets '23, Cambridge, MA, USA: Association for Computing Machinery, 2023, pp. 189–195, isbn: 9798400704154. doi: [10.1145/3626111.3628194](https://doi.org/10.1145/3626111.3628194). [Online]. Available: <https://doi.org/10.1145/3626111.3628194>.
- [38] J. Kim et al., "IBCS: Intent-Based Cloud Services for Security Applications," *IEEE Communications Magazine*, vol. 58, no. 4, pp. 45–51, 2020. doi: [10.1109/MCOM.001.1900476](https://doi.org/10.1109/MCOM.001.1900476).
- [39] K. Wang, H. Du, and L. Su, "Intent-Driven Network Slicing Security Provision and Management," in *2023 IEEE 23rd International Conference on Communication Technology (ICCT)*, 2023, pp. 1318–1324. doi: [10.1109/ICCT59356.2023.10419581](https://doi.org/10.1109/ICCT59356.2023.10419581).
- [40] B. E. Ujcich and W. H. Sanders, "Data Protection Intents for Software-Defined Networking," in *2019 IEEE Conference on Network Softwarization (NetSoft)*, 2019, pp. 271–275. doi: [10.1109/NETSOFT.2019.8806684](https://doi.org/10.1109/NETSOFT.2019.8806684).
- [41] N. Herbaut, C. Correa, J. Robin, and R. Mazo, "SDN Intent-based conformance checking: application to security policies," in *2021 IEEE 7th International Conference on Network Softwarization (NetSoft)*, 2021, pp. 181–185. doi: [10.1109/NetSoft51509.2021.9492679](https://doi.org/10.1109/NetSoft51509.2021.9492679).

- [42] G. Landeau, M. Avgeris, A. Leivadreas, and I. Lambadaris, "Security-Oriented Network Intent Placement using Particle Swarm Optimization," in *2023 7th Cyber Security in Networking Conference (CSNet)*, 2023, pp. 19–22. doi: [10.1109/CSNet59123.2023.10339783](https://doi.org/10.1109/CSNet59123.2023.10339783).
- [43] E. J. Scheid et al., "INSpIRE: Integrated NFV-based Intent Re $\square$ nement Environment," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2017, pp. 186–194. doi: [10.23919/INM.2017.7987279](https://doi.org/10.23919/INM.2017.7987279).
- [44] J. J. Diaz Rivera, M. Afaq, and W.-C. Song, "Blockchain and Intent-Based Networking: A Novel Approach to Secure and Accurate Network Policy Implementation," in *2023 24th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2023, pp. 77–82.
- [45] A. Balakrishnan, S. De, and L.-C. Wang, "Network operator revenue maximization in dual powered green cellular networks," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 4, pp. 1791–1805, 2021. doi: [10.1109/TGCN.2021.3094164](https://doi.org/10.1109/TGCN.2021.3094164).
- [46] Z. Garroussi, A. W. Badirou, M. D'Amours, A. Girard, and B. Sans $\acute{o}$, "On the economic viability of solar energy when upgrading cellular networks," *IEEE Transactions on Green Communications and Networking*, vol. 7, no. 2, pp. 744–758, 2023. doi: [10.1109/TGCN.2022.3223622](https://doi.org/10.1109/TGCN.2022.3223622).
- [47] K. Chen, Y. Sun, S. Zheng, H. Yang, and P. Yu, "Online collaborative energy-network resource scheduling for wpt-enabled green edge computing," *IEEE Transactions on Green Communications and Networking*, vol. 8, no. 2, pp. 601–618, 2024. doi: [10.1109/TGCN.2023.3339477](https://doi.org/10.1109/TGCN.2023.3339477).
- [48] N. Hu, Z. Tian, X. Du, and M. Guizani, "An energy-e $\square$ cient in-network computing paradigm for 6g," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 4, pp. 1722–1733, 2021. doi: [10.1109/TGCN.2021.3099804](https://doi.org/10.1109/TGCN.2021.3099804).
- [49] Z. Zhou, M. Shojafar, M. Alazab, J. Abawajy, and F. Li, "Afed-ef: An energy-e $\square$ cient vm allocation algorithm for iot applications in a cloud data center," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 2, pp. 658–669, 2021. doi: [10.1109/TGCN.2021.3067309](https://doi.org/10.1109/TGCN.2021.3067309).
- [50] E. Tong et al., "A hierarchical energy-e $\square$ cient service selection approach with qos constraints for internet of things," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 2, pp. 645–657, 2021. doi: [10.1109/TGCN.2021.3069121](https://doi.org/10.1109/TGCN.2021.3069121).
- [51] Y. Huang, H. Xu, H. Gao, X. Ma, and W. Hussain, "Ssur: An approach to optimizing virtual machine allocation strategy based on user requirements for cloud data center," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 2, pp. 670–681, 2021. doi: [10.1109/TGCN.2021.3067374](https://doi.org/10.1109/TGCN.2021.3067374).
- [52] S. Afzal et al., *A survey on energy consumption and environmental impact of video streaming*, 2024. arXiv: [2401.09854](https://arxiv.org/abs/2401.09854) [cs.MM]. [Online]. Available: <https://arxiv.org/abs/2401.09854>.

- [53] O. Ayoub et al., "Energy-efficient video-on-demand content caching and distribution in metro area networks," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 1, pp. 159–169, 2019. doi: [10.1109/TGCN.2018.2878918](https://doi.org/10.1109/TGCN.2018.2878918).
- [54] A. Montazerolghaem et al., "Green cloud multimedia networking: Nfv/sdn based energy-efficient resource allocation," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 3, pp. 873–889, 2020. doi: [10.1109/TGCN.2020.2982821](https://doi.org/10.1109/TGCN.2020.2982821).
- [55] M. Choi et al., "Joint delay-sensitive and power-efficient quality control of dynamic video streaming using adaptive super-resolution," *IEEE Transactions on Green Communications and Networking*, vol. 8, no. 1, pp. 103–117, 2024. doi: [10.1109/TGCN.2023.3319437](https://doi.org/10.1109/TGCN.2023.3319437).
- [56] B. M. Beena et al., "Green video transcoding in cloud environments using kubernetes: A framework with dynamic renewable energy allocation and priority scheduling," *IEEE Access*, vol. 13, pp. 116 362–116 390, 2025. doi: [10.1109/ACCESS.2025.3585615](https://doi.org/10.1109/ACCESS.2025.3585615).
- [57] Carbon Trust, "Carbon impact of video streaming," Carbon Trust, Tech. Rep., Jun. 2021.
- [58] J. Malmudin, "The power consumption of mobile and fixed network data services-the case of streaming video and downloading large files," in *Electronics Goes Green*, vol. 2020, 2020, p. 12.
- [59] C. Pignataro et al., "Sustainability Considerations for Networking Protocols and Applications," Internet Engineering Task Force, Internet-Draft draft-pignataro-enviro-sustainability-consid-03, Nov. 2025, Work in Progress, 10 pp.
- [60] G. Sciddurlo, F. de Trizio, G. Piro, and G. Boggia, "A value-driven system design framework for sustainable 6g networks," *Computer Networks*, vol. 269, p. 111 477, 2025, issn: 1389-1286. doi: <https://doi.org/10.1016/j.comnet.2025.111477>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S138912862500444X>.
- [61] F. de Trizio, G. Sciddurlo, G. Piro, and G. Boggia, "Tetrees: Trade-off evaluation through refined exact epsilon-constraint solver," *Software Impacts*, vol. 28, p. 100 828, 2026, issn: 2665-9638. doi: <https://doi.org/10.1016/j.simpa.2026.100828>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2665963826000187>.
- [62] F. de Trizio, G. Sciddurlo, D. Rutigliano, G. Piro, and G. Boggia, "A novel malicious intent detection approach in Intent-Based enterprise networks," in *1st International workshop on Network Security Operations (NeSecOr) 2024 (NeSecOr 2024)*, Prague, Czech Republic, Oct. 2024, p. 7.
- [63] H. Wymeersch et al., "6G Positioning and Sensing Through the Lens of Sustainability, Inclusiveness, and Trustworthiness," *arXiv e-prints*, 2023. doi: [10.48550/arXiv.2309.13602](https://doi.org/10.48550/arXiv.2309.13602).

- [64] H. Wang, C. Yu, L. Wang, and Q. Yu, "Effective bigdata-space service selection over trust and heterogeneous qos preferences," *IEEE Transactions on Services Computing*, vol. 11, no. 4, pp. 644–657, 2018. doi: [10.1109/TSC.2015.2480393](https://doi.org/10.1109/TSC.2015.2480393).
- [65] I. O. for Standardization, *Greenhouse Gases: Carbon Footprint of Products: Requirements and Guidelines for Quantification*. International Organization for Standardization, 2018.
- [66] A. Clemm et al., "Green Networking Metrics for Environmentally Sustainable Networking," Internet Engineering Task Force, Internet-Draft draft-cx-green-green-metrics-00, Oct. 2024, Work in Progress, 28 pp. [Online]. Available: <https://datatracker.ietf.org/doc/draft-cx-green-green-metrics/00/>.
- [67] G. Zhang, F. Shen, Y. Zhang, R. Yang, Y. Yang, and E. A. Jorswieck, "Delay minimized task scheduling in fog-enabled iot networks," in *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2018, pp. 1–6. doi: [10.1109/WCSP.2018.8555532](https://doi.org/10.1109/WCSP.2018.8555532).
- [68] L. Lannelongue, J. Grealey, and M. Inouye, "Green algorithms: Quantifying the carbon footprint of computation," *Advanced Science*, vol. 8, no. 12, p. 2100707, 2021. doi: <https://doi.org/10.1002/advs.202100707>.
- [69] F. de Trizio, G. Sciddurlo, I. Cianci, G. Piro, and G. Boggia, "Optimizing key value indicators in intent-based networks through digital twins aided service orchestration mechanisms," *Computer Communications*, vol. 228, p. 107977, 2024, issn: 0140-3664. doi: <https://doi.org/10.1016/j.comcom.2024.107977>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0140366424003244>.
- [70] M. Battaglioni, G. Rafaiani, F. Chiaraluce, and M. Baldi, "Magic: A method for assessing cyber incidents occurrence," *IEEE Access*, vol. 10, pp. 73458–73473, 2022. doi: [10.1109/ACCESS.2022.3189777](https://doi.org/10.1109/ACCESS.2022.3189777).
- [71] S. Sahni and T. Gonzalez, "P-complete approximation problems," *J. ACM*, vol. 23, no. 3, pp. 555–565, Jul. 1976, issn: 0004-5411. doi: [10.1145/321958.321975](https://doi.org/10.1145/321958.321975). [Online]. Available: <https://doi.org/10.1145/321958.321975>.
- [72] Y. H. Yv, L. S. Lasdon, and D. Da, "On a bicriterion formulation of the problems of integrated system identification and system optimization," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-1, no. 3, pp. 296–297, 1971. doi: [10.1109/TSMC.1971.4308298](https://doi.org/10.1109/TSMC.1971.4308298).
- [73] G. Mavrotas, "Effective implementation of the ϵ -constraint method in multi-objective mathematical programming problems," *Applied Mathematics and Computation*, vol. 213, no. 2, pp. 455–465, 2009, issn: 0096-3003. doi: <https://doi.org/10.1016/j.amc.2009.03.037>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0096300309002574>.

- [74] J.-F. Bérubé, M. Gendreau, and J.-Y. Potvin, “An exact ε -constraint method for bi-objective combinatorial optimization problems: Application to the traveling salesman problem with profits,” *European Journal of Operational Research*, vol. 194, no. 1, pp. 39–50, 2009, issn: 0377-2217. doi: <https://doi.org/10.1016/j.ejor.2007.12.014>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0377221707012106>.
- [75] T. Denat, A. Harutyunyan, N. Melissinos, and V. T. Paschos, “Average-case complexity of a branch-and-bound algorithm for min dominating set,” *Discrete Applied Mathematics*, vol. 345, pp. 4–8, 2024, issn: 0166-218X. doi: <https://doi.org/10.1016/j.dam.2023.11.021>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0166218X23004353>.
- [76] A. Basu, M. Conforti, M. Di Summa, and H. Jiang, “Complexity of branch-and-bound and cutting planes in mixed-integer optimization,” *Mathematical Programming*, vol. 198, no. 1, pp. 787–810, Mar. 2023, issn: 1436-4646. doi: [10.1007/s10107-022-01789-5](https://doi.org/10.1007/s10107-022-01789-5). [Online]. Available: <https://doi.org/10.1007/s10107-022-01789-5>.
- [77] M.-F. Balcan, S. Prasad, T. Sandholm, and E. Vitercik, *Improved sample complexity bounds for branch-and-cut*, 2022. arXiv: [2111.11207](https://arxiv.org/abs/2111.11207) [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2111.11207>.
- [78] N. Milanovic and B. Milic, “Automatic generation of service availability models,” *IEEE Transactions on Services Computing*, vol. 4, no. 1, pp. 56–69, 2011. doi: [10.1109/TSC.2010.11](https://doi.org/10.1109/TSC.2010.11).
- [79] S. Roessel, C. Drewes, et al., “6g use cases and requirements,” SNS-JU, Horizon, EU, Deliverable 1.2, 2023. [Online]. Available: <https://hexa-x-ii.eu/results/>.
- [80] M. Marchese, F. de Trizio, G. Sciddurlo, N. Cordeschi, G. Piro, and G. Boggia, “GreenTent: an Intent-Based framework for sustainable network orchestration,” in *2026 IEEE International Conference on Communications Workshops (ICC Workshops)*, Glasgow, United Kingdom (Great Britain), May 2026, p. 6.
- [81] A. Petrosino et al., “Dynamic management of forwarding rules in a t-sdn architecture with energy and bandwidth constraints,” in *Ad-Hoc, Mobile, and Wireless Networks: 19th International Conference on Ad-Hoc Networks and Wireless, ADHOC-NOW 2020*, 2020, pp. 3–15, isbn: 978-3-030-61745-5. doi: [10.1007/978-3-030-61746-2_1](https://doi.org/10.1007/978-3-030-61746-2_1).
- [82] Cisco, “Cisco catalyst 2960-x and 2960-xr series switches data sheet,” Cisco Systems, Tech. Rep., 2021, Document ID: c78-728232.
- [83] P. K. Thiruvassagam, A. Chakraborty, A. Mathew, and C. S. R. Murthy, “Reliable placement of service function chains and virtual monitoring functions with minimal cost in softwarized 5g networks,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1491–1507, 2021. doi: [10.1109/TNSM.2021.3056917](https://doi.org/10.1109/TNSM.2021.3056917).

- [84] L. Ji, S. He, W. Wu, C. Gu, J. Bi, and Z. Shi, "Dynamic network slicing orchestration for remote adaptation and configuration in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 4297–4307, 2022. doi: [10.1109/TII.2021.3131355](https://doi.org/10.1109/TII.2021.3131355).
- [85] J.-S. Tsai, I.-H. Chuang, J.-J. Liu, Y.-H. Kuo, and W. Liao, "Qos-aware fog service orchestration for industrial internet of things," *IEEE Transactions on Services Computing*, vol. 15, no. 3, pp. 1265–1279, 2022. doi: [10.1109/TSC.2020.2978472](https://doi.org/10.1109/TSC.2020.2978472).
- [86] J. Sahni and D. P. Vidyarthi, "A cost-effective deadline-constrained dynamic scheduling algorithm for scientific workflows in a cloud environment," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 2–18, 2018. doi: [10.1109/TCC.2015.2451649](https://doi.org/10.1109/TCC.2015.2451649).
- [87] L. Qu, C. Assi, and K. Shaban, "Delay-aware scheduling and resource optimization with network function virtualization," *IEEE Transactions on Communications*, vol. 64, no. 9, pp. 3746–3758, 2016. doi: [10.1109/TCOMM.2016.2580150](https://doi.org/10.1109/TCOMM.2016.2580150).
- [88] C. Swain et al., "Meto: Matching-theory-based efficient task offloading in iot-fog interconnection networks," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12 705–12 715, 2021. doi: [10.1109/JIOT.2020.3025631](https://doi.org/10.1109/JIOT.2020.3025631).
- [89] M. A. Uusitalo et al., "6g vision, value, use cases and technologies from european 6g flagship project hexa-x," *IEEE Access*, vol. 9, pp. 160 004–160 020, 2021. doi: [10.1109/ACCESS.2021.3130030](https://doi.org/10.1109/ACCESS.2021.3130030).
- [90] C. Zhou et al., "Digital twin network concepts and reference architecture," Internet Engineering Task Force, Internet-Draft draft-irtf-nmrg-network-digital-twin-arch-04, Oct. 2023.
- [91] ITU, "Digital twin network - Requirements and architecture," International Telecommunication Union Std, Recommendation ITU-T Y.Y.3090, Feb. 2022.
- [92] Y. Tao, J. Wu, X. Lin, and W. Yang, "Drl-driven digital twin function virtualization for adaptive service response in 6g networks," *IEEE Networking Letters*, vol. 5, no. 2, pp. 125–129, 2023. doi: [10.1109/LNET.2023.3269766](https://doi.org/10.1109/LNET.2023.3269766).
- [93] Y. Ma, F. Song, G. Pau, I. You, and H. Zhang, "Adaptive service provisioning for dynamic resource allocation in network digital twin," *IEEE Network*, pp. 1–1, 2023. doi: [10.1109/MNET.2023.3337245](https://doi.org/10.1109/MNET.2023.3337245).
- [94] J. Li et al., "Digital twin-assisted, sfc-enabled service provisioning in mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 23, no. 1, pp. 393–408, 2024. doi: [10.1109/TMC.2022.3227248](https://doi.org/10.1109/TMC.2022.3227248).
- [95] R. Zhang, Z. Xie, D. Yu, W. Liang, and X. Cheng, "Digital twin-assisted federated learning service provisioning over mobile edge networks," *IEEE Transactions on Computers*, vol. 73, no. 02, pp. 586–598, Feb. 2024, issn: 1557-9956. doi: [10.1109/TC.2023.3337317](https://doi.org/10.1109/TC.2023.3337317).

- [96] N. A. Mitsiou, V. K. Papanikolaou, P. D. Diamantoulakis, T. Q. Duong, and G. K. Karagiannidis, "Digital twin-aided orchestration of mobile edge computing with grant-free access," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 841–853, 2023. doi: [10.1109/OJCOMS.2023.3260165](https://doi.org/10.1109/OJCOMS.2023.3260165).
- [97] D. Van Huynh et al., "Ullc edge networks with joint optimal user association, task offloading and resource allocation: A digital twin approach," *IEEE Transactions on Communications*, vol. 70, no. 11, pp. 7669–7682, 2022. doi: [10.1109/TCOMM.2022.3205692](https://doi.org/10.1109/TCOMM.2022.3205692).
- [98] T. Liu et al., "Resource allocation in dt-assisted internet of vehicles via edge intelligent cooperation," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 17 608–17 626, 2022. doi: [10.1109/JIOT.2022.3156100](https://doi.org/10.1109/JIOT.2022.3156100).
- [99] Y. Gong, Y. Wei, Z. Feng, F. R. Yu, and Y. Zhang, "Resource allocation for integrated sensing and communication in digital twin enabled internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 4, pp. 4510–4524, 2023. doi: [10.1109/TVT.2022.3228583](https://doi.org/10.1109/TVT.2022.3228583).
- [100] B. Hazarika, K. Singh, C.-P. Li, A. Schmeink, and K. F. Tsang, "Radit: Resource allocation in digital twin-driven uav-aided internet of vehicle networks," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3369–3385, 2023. doi: [10.1109/JSAC.2023.3310048](https://doi.org/10.1109/JSAC.2023.3310048).
- [101] C. V. Nahum et al., "Intent-aware radio resource scheduling in a ran slicing scenario using reinforcement learning," *IEEE Transactions on Wireless Communications*, vol. 23, no. 3, pp. 2253–2267, 2024. doi: [10.1109/TWC.2023.3297014](https://doi.org/10.1109/TWC.2023.3297014).
- [102] A. Leivadeas and M. Falkner, "Vnf placement problem: A multi-tenant intent-based networking approach," in *2021 24th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, 2021, pp. 143–150. doi: [10.1109/ICIN51074.2021.9385553](https://doi.org/10.1109/ICIN51074.2021.9385553).
- [103] K. Dzeperoska, A. Tizghadam, and A. Leon-Garcia, "Emergence: An intent fulfillment system," *IEEE Communications Magazine*, vol. 62, no. 6, pp. 36–41, 2024. doi: [10.1109/MCOM.001.2300270](https://doi.org/10.1109/MCOM.001.2300270).
- [104] J. Zhang et al., "Intent-driven closed-loop control and management framework for 6g open ran," *IEEE Internet of Things Journal*, vol. 11, no. 4, pp. 6314–6327, 2024. doi: [10.1109/JIOT.2023.3312795](https://doi.org/10.1109/JIOT.2023.3312795).
- [105] H. Mei and L. Peng, "On multi-robot data collection and offloading for space-aerial-surface computing," *IEEE Wireless Communications*, vol. 30, no. 2, pp. 90–96, 2023. doi: [10.1109/MWC.005.2200400](https://doi.org/10.1109/MWC.005.2200400).
- [106] S. Ganesh, V. Gopalasamy, and N. B. Sai Shibu, "Architecture for drone assisted emergency ad-hoc network for disaster rescue operations," in *2021 International Conference on COMMunication Systems & NETWORKS (COMSNETS)*, 2021, pp. 44–49. doi: [10.1109/COMSNETS51098.2021.9352814](https://doi.org/10.1109/COMSNETS51098.2021.9352814).
- [107] F. Mezghani and N. Mitton, "The potential of cooperative communications to speed up disaster relief operations," in *2019 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, 2019, pp. 1–6. doi: [10.1109/ICT-DM47966.2019.9032963](https://doi.org/10.1109/ICT-DM47966.2019.9032963).

- [108] G. Aiello, F. Hopps, D. Santisi, and M. Venticinque, "The employment of unmanned aerial vehicles for analyzing and mitigating disaster risks in industrial sites," *IEEE Transactions on Engineering Management*, vol. 67, no. 3, pp. 519–530, 2020. doi: [10.1109/TEM.2019.2949479](https://doi.org/10.1109/TEM.2019.2949479).
- [109] Y. Wang et al., "Task offloading for post-disaster rescue in unmanned aerial vehicles networks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1525–1539, 2022. doi: [10.1109/TNET.2022.3140796](https://doi.org/10.1109/TNET.2022.3140796).
- [110] M. Terzi, A. Anastasiou, P. Kolios, C. Panayiotou, and T. Theocharides, "Swifters: A multi-uav platform for disaster management," in *2019 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, 2019, pp. 1–7. doi: [10.1109/ICT-DM47966.2019.9032923](https://doi.org/10.1109/ICT-DM47966.2019.9032923).
- [111] X. Yuan, J. Chen, N. Zhang, J. Ni, F. R. Yu, and V. C. M. Leung, "Digital twin-driven vehicular task offloading and resource configuration in the internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24290–24304, 2022. doi: [10.1109/TITS.2022.3204585](https://doi.org/10.1109/TITS.2022.3204585).
- [112] X. Xu, Z. Liu, M. Bilal, S. Vimal, and H. Song, "Computation offloading and service caching for intelligent transportation systems with digital twin," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 20757–20772, 2022. doi: [10.1109/TITS.2022.3190669](https://doi.org/10.1109/TITS.2022.3190669).
- [113] T. Meuser, L. Baumgrärtner, and B. Becker, "Netskylines: Digital twins for evaluating disaster communication," in *2021 IEEE Global Humanitarian Technology Conference (GHTC)*, 2021, pp. 68–71. doi: [10.1109/GHTC53159.2021.9612413](https://doi.org/10.1109/GHTC53159.2021.9612413).
- [114] M. Khan et al., "Centralized and reliable digital twin models for smart city's buildings protection during disaster," in *2022 International Conference on Computational Modelling, Simulation and Optimization (ICCMO)*, 2022, pp. 226–229. doi: [10.1109/ICCMO58359.2022.00053](https://doi.org/10.1109/ICCMO58359.2022.00053).
- [115] G. Sciddurlo, I. Huso, D. Striccoli, G. Piro, and G. Boggia, "A multi-tiered social iot architecture for scalable and trusted service provisioning," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6. doi: [10.1109/GLOBECOM46510.2021.9685084](https://doi.org/10.1109/GLOBECOM46510.2021.9685084).
- [116] F. De Trizio, G. Sciddurlo, I. Cianci, D. Striccoli, G. Piro, and G. Boggia, "Surviving disaster events via dynamic in-network processing assisted by network digital twins," in *2023 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, 2023, pp. 1–6. doi: [10.1109/ICT-DM58371.2023.10286937](https://doi.org/10.1109/ICT-DM58371.2023.10286937).
- [117] ITU-T, "Network 2030 - Architecture Framework," ITU-T Telecommunication Standardization Sector of ITU, Technical Report (TR), 2020, ITU-T Recommendation FG-NET2030-Arch(2020).

- [118] J. Zhang, J. F. Campbell, D. C. Sweeney II, and A. C. Hupman, "Energy consumption models for delivery drones: A comparison and assessment," *Transportation Research Part D: Transport and Environment*, vol. 90, p. 102 668, 2021, issn: 1361-9209. doi: <https://doi.org/10.1016/j.trd.2020.102668>.
- [119] E. ZSM, "Zero-touch network and service management (zsm); intent-driven autonomous networks; generic aspects," ETSI, ISG ZSM, Tech. Rep., 2023, ETSI GR ZSM 011 V1.1.1 (2023-02), p. 60.
- [120] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.
- [121] G. Strupczewski, "Defining cyber risk," *Safety Science*, vol. 135, p. 105 143, 2021, issn: 0925-7535. doi: <https://doi.org/10.1016/j.ssci.2020.105143>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925753520305397>.
- [122] C. Biener, M. Eling, and J. H. Wirfs, "Insurability of cyber risk: An empirical analysis," *The Geneva Papers on Risk and Insurance-Issues and Practice*, vol. 40, pp. 131–158, 2015.
- [123] G. Stergiopoulos, D. Gritzalis, and V. Kouktzoglou, "Using formal distributions for threat likelihood estimation in cloud-enabled it risk assessment," *Computer Networks*, vol. 134, pp. 23–45, 2018, issn: 1389-1286. doi: <https://doi.org/10.1016/j.comnet.2018.01.033>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1389128618300446>.
- [124] A. Roy, D. S. Kim, and K. S. Trivedi, "Cyber security analysis using attack countermeasure trees," in *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, 2010, pp. 1–4.
- [125] A. Ofori-Yeboah, R. Addo-Quaye, W. Oseni, P. Amorin, and C. Agangmikre, "Cyber supply chain security: A cost benefit analysis using net present value," in *2021 International Conference on Cyber Security and Internet of Things (ICSIoT)*, 2021, pp. 49–54. doi: [10.1109/ICSIoT55070.2021.00018](https://doi.org/10.1109/ICSIoT55070.2021.00018).
- [126] J. Kleinberg and E. Tardos, *Algorithm Design*. USA: Addison-Wesley Longman Publishing Co., Inc., 2005, isbn: 0321295358.
- [127] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching theory for future wireless networks: Fundamentals and applications," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 52–59, 2015. doi: [10.1109/MCOM.2015.7105641](https://doi.org/10.1109/MCOM.2015.7105641).
- [128] J. Ren et al., "Matching algorithms: Fundamentals, applications and challenges," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 3, pp. 332–350, 2021.
- [129] S. Bayat, Y. Li, L. Song, and Z. Han, "Matching theory: Applications in wireless communications," *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 103–122, 2016. doi: [10.1109/MSP.2016.2598848](https://doi.org/10.1109/MSP.2016.2598848).

- [130] E. Roszkowska, "Multi-criteria decision making models by applying the topsis method to crisp and interval data," *Multiple Criteria Decision Making/University of Economics in Katowice*, 2011. [Online]. Available: <https://api.semanticscholar.org/CorpusID:2363622>.
- [131] A. Jahan, F. Mustapha, S. Sapuan, M. Ismail, and M. Bahraminasab, "A framework for weighting of criteria in ranking stage of material selection process," *International Journal of Advanced Manufacturing Technology*, vol. 58, pp. 411–420, Jan. 2012. doi: [10.1007/s00170-011-3366-7](https://doi.org/10.1007/s00170-011-3366-7).
- [132] A. Roth, "Deferred acceptance algorithms: History, theory, practice, and open questions," *International Journal of Game Theory*, vol. 36, pp. 537–569, Feb. 2008. doi: [10.1007/s00182-008-0117-6](https://doi.org/10.1007/s00182-008-0117-6).
- [133] B. Masood Khorsandi et al., "Targets and requirements for 6g - initial e2e architecture," A flagship for B5G6G vision, intelligent fabric of technology enablers connecting human, physical, and digital worlds, Deliverable 1.3, 2022. [Online]. Available: <https://hexa-x.eu/deliverables/>.
- [134] F. de Trizio et al., "Sustainable digital twin networks: Architecture, intent-based orchestration, and optimization," in *RESTART Telecommunications of the Future: A Book of Contributions from the RESTART Program*, N. Blefari Melazzi and A. Capone, Eds. RESTART Foundation, 2026, ch. 1, pp. 3–36, isbn: 9788832170061.
- [135] M. Becattini, A. Iera, L. Paroli, and G. Fontani, "Digital twin networks for sustainable in-network computing in future 6G networks," in *2024 IEEE 35th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, IEEE, 2024, pp. 1–6.
- [136] Eclipse Foundation, *Eclipse basyx: The digital twin infrastructure*, <https://eclipse.dev/basyx/>, 2025.
- [137] G. Cavalieri Di Natale, "Asset administration shell digital twin of 5g communication system," in *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics - Volume 2: ICINCO, INSTICC, SciTePress*, 2024, pp. 395–402, isbn: 978-989-758-717-7. doi: [10.5220/0012914200003822](https://doi.org/10.5220/0012914200003822).
- [138] C. Antona, D. N. Cavalieri, Q. Gambadoro, and S. Rossitto, "Kubernetes-driven digital twin framework using asset administration shell for smart industry," in *Proceedings of the 6th IFAC/INSTICC International Conference on Innovative Intelligent Industrial Production and Logistics*, in press, 2025. [Online]. Available: <https://hdl.handle.net/20.500.11769/694569>.
- [139] F. de Trizio, G. Sciddurlo, G. Piro, and G. Boggia, "Agentic ai for autonomous internet-of-everything: A distributed three-tier intelligence ecosystem," *submitted to the IEEE Internet of Things Magazine*, pp. 1–11, 2026.
- [140] A. Artigiani, C. Bergho, et al., "Unlocking digital transformation with autonomous networks - etsi perspectives and major achievements," European Telecommunications Standards Institute (ETSI), White Paper, 2023.
- [141] K. McDonnell et al., "Autonomous networks technical architecture," TM Forum, Tech. Rep., 2023.

- [142] NGMN Alliance, "Autonomous system and network automation framework," Next Generation Mobile Networks Alliance, Tech. Rep., version 1.0, Sep. 2022. [Online]. Available: <https://www.ngmn.org>.
- [143] Z. Xi et al., *The rise and potential of large language model based agents: A survey*, 2023. arXiv: [2309.07864](https://arxiv.org/abs/2309.07864) [cs.AI]. [Online]. Available: <https://arxiv.org/abs/2309.07864>.
- [144] A. Ehtesham, A. Singh, G. K. Gupta, and S. Kumar, *A survey of agent interoperability protocols: Model context protocol (mcp), agent communication protocol (acp), agent-to-agent protocol (a2a), and agent network protocol (anp)*, 2025. arXiv: [2505.02279](https://arxiv.org/abs/2505.02279) [cs.AI]. [Online]. Available: <https://arxiv.org/abs/2505.02279>.
- [145] P. Belcak et al., *Small language models are the future of agentic ai*, 2025. arXiv: [2506.02153](https://arxiv.org/abs/2506.02153) [cs.AI]. [Online]. Available: <https://arxiv.org/abs/2506.02153>.
- [146] J. Lin, L. Zhu, W.-M. Chen, W.-C. Wang, and S. Han, "Tiny machine learning: Progress and futures [feature]," *IEEE Circuits and Systems Magazine*, vol. 23, no. 3, pp. 8–34, 2023. doi: [10.1109/MCAS.2023.3302182](https://doi.org/10.1109/MCAS.2023.3302182).
- [147] S. Gebreab, A. Musamih, K. Salah, R. Jayaraman, and D. Bosovic, "Accelerating digital twin development with generative ai: A framework for 3d modeling and data integration," *IEEE Access*, vol. 12, pp. 185 918–185 936, 2024. doi: [10.1109/ACCESS.2024.3514175](https://doi.org/10.1109/ACCESS.2024.3514175).
- [148] "Management aspects of closed control loops (release 19)," 3rd Generation Partnership Project (3GPP), Tech. Rep. TS 28.567 V0.2.0, Apr. 2025, Available from 3GPP Organizational Partners' Publications Offices.
- [149] ITU-T Focus Group on Autonomous Networks, "Architecture framework for autonomous networks," International Telecommunication Union, Tech. Rep., Sep. 2022, Technical Specification. [Online]. Available: <https://www.itu.int>.
- [150] ETSI, "Intelligent Transport Systems (ITS); ITS-G5 Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band," European Telecommunications Standards Institute (ETSI), Sophia Antipolis, France, European Standard ETSI EN 302 663, version V1.3.1, Jan. 2020, Accessed: 2026-02-20. [Online]. Available: https://www.etsi.org/deliver/etsi_en/302600_302699/302663/01.03.01_60/en_302663v010301p.pdf.
- [151] M. Cemri et al., *Why do multi-agent llm systems fail?* 2025. arXiv: [2503.13657](https://arxiv.org/abs/2503.13657) [cs.AI]. [Online]. Available: <https://arxiv.org/abs/2503.13657>.
- [152] T. Vartziotis et al., "Learn to code sustainably: An empirical study on green code generation," in *Proceedings of the 1st International Workshop on Large Language Models for Code*, ser. LLM4Code '24, Lisbon, Portugal: Association for Computing Machinery, 2024, pp. 30–37, isbn: 9798400705793. doi: [10.1145/3643795.3648394](https://doi.org/10.1145/3643795.3648394). [Online]. Available: <https://doi.org/10.1145/3643795.3648394>.

- [153] V. Rathod, S. Nabavirazavi, S. Zad, and S. S. Iyengar, "Privacy and security challenges in large language models," in *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*, 2025, pp. 00 746–00 752. doi: [10.1109/CCWC62904.2025.10903912](https://doi.org/10.1109/CCWC62904.2025.10903912).
- [154] F. de Trizio, G. Sciddurlo, A. Petrosino, G. Piro, and G. Boggia, "A scalable framework for responsive trustworthiness dissemination in social ioa," in *Proceedings of the CoNEXT on Student Workshop 2024*, ser. CoNEXT-SW '24, Los Angeles, CA, USA: Association for Computing Machinery, 2024, pp. 5–6, isbn: 9798400712555. doi: [10.1145/3694812.3699925](https://doi.org/10.1145/3694812.3699925). [Online]. Available: <https://doi.org/10.1145/3694812.3699925>.
- [155] C.-H. Hsu, C. E. Montenegro Marin, R. Gonzalez Crespo, and H. F. Mohamed El-sayed, "Guest editorial introduction to the special section on social computing and social internet of things," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 3, pp. 947–949, 2022. doi: [10.1109/TNSE.2022.3167460](https://doi.org/10.1109/TNSE.2022.3167460).
- [156] W. Z. Khan, Q.-A. Arshad, S. Hakak, M. K. Khan, and Saeed-Ur-Rehman, "Trust management in social internet of things: Architectures, recent advancements, and future challenges," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 7768–7788, 2021. doi: [10.1109/JIOT.2020.3039296](https://doi.org/10.1109/JIOT.2020.3039296).
- [157] G. Sciddurlo, A. Petrosino, D. Striccoli, G. Piro, L. A. Grieco, and G. Boggia, "Boosting service provisioning in siot by exploiting trust and capability levels of social objects," in *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2022, pp. 1–6. doi: [10.1109/SMARTCOMP55677.2022.00077](https://doi.org/10.1109/SMARTCOMP55677.2022.00077).
- [158] O. Novo, "Blockchain meets iot: An architecture for scalable access management in iot," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018. doi: [10.1109/JIOT.2018.2812239](https://doi.org/10.1109/JIOT.2018.2812239).
- [159] M. Deng et al., "Lightweight trust management scheme based on blockchain in resource-constrained intelligent iot systems," *IEEE Internet of Things Journal*, vol. 11, no. 15, pp. 25 706–25 719, 2024. doi: [10.1109/JIOT.2024.3380850](https://doi.org/10.1109/JIOT.2024.3380850).