



Sorbonne Université
Ecole Doctorale Informatique, Télécommunications et Electronique de Paris (ED130)
EURECOM
Communication Systems

Agentic AI for 6G Radio Access Network Automation

A Dissertation

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in the Doctoral School of Sorbonne University

Presented by Ilias Chatzistefanidis

Thesis defense scheduled for June 11th, 2026, before a jury composed of:

Reviewers

Prof. Anna Tzanakaki	National and Kapodistrian University of Athens, Greece
Prof. Mahesh Marina	University of Edinburgh, United Kingdom

Examiners

Prof. Tommaso Melodia	Northeastern University, USA
Dr. Xavier Costa-Pérez	ICREA, i2CAT, NEC, Spain
Prof. Adlen Ksentini	EURECOM, France
Dr. Marco Fiore	IMDEA, Spain
Prof. Mehdi Bennis	University of Oulu, Finland

Thesis Director

Prof. Navid Nikaein	EURECOM, France

Invited Member

Prof. Merouane Debbah	Khalifa University, United Arab Emirates

Sorbonne Université
Ecole Doctorale Informatique, Télécommunications et Electronique de Paris (ED130)
EURECOM
Systèmes de Communication

IA agentique pour l'automatisation du réseau d'accès radio 6G

Une Thèse

Présentée en vue de l'obtention partielle du grade de Docteur en Philosophie
au sein de l'École Doctorale de l'Université de la Sorbonne

Présentée par Ilias Chatzistefanidis

Soutenance prévue le 11 juin 2026, devant un jury composé de:

Rapporteurs

Prof. Anna Tzanakaki	Université nationale et capodistrienne d'Athènes, Grèce
Prof. Mahesh Marina	Université d'Édimbourg, Royaume-Uni

Examinateurs

Prof. Tommaso Melodia	Université Northeastern, États-Unis
Dr. Xavier Costa-Pérez	ICREA, Fondation i2CAT, NEC, Espagne
Prof. Adlen Ksentini	EURECOM, France
Dr. Marco Fiore	Institut des réseaux IMDEA, Espagne
Prof. Mehdi Bennis	Université d'Oulu, Finlande

Directeur de thèse

Prof. Navid Nikaein	EURECOM, France

Membre invité

Prof. Merouane Debbah	Université Khalifa, Émirats arabes unis

Abstract

Next-generation 6G radio access networks will need to serve multiple stakeholders, support diverse services, and operate across heterogeneous cloud-edge infrastructures. Conventional network slice controllers remain rigid and disconnected from business context, leaving a persistent service-level agreement (SLA) gap between high-level service objectives and real-time network conditions. This thesis investigates how agentic AI, powered by large language model (LLM) agents, can bring intelligence across the future 6G RAN stack—from business-plane negotiation to near-real-time control—while ensuring fairness, efficiency, and trust.

We first design Maestro [20], a collaborative business-plane framework in which each operator or vertical is represented by an LLM agent. Stakeholders express high-level intents in natural language, and a mediator agent, assisted by optimisation modules, detects conflicts and guides iterative negotiation to consensus. The resulting service intents are translated into network intents and enforced by RAN slicing functions, aligning business goals with technical feasibility.

To reduce hallucinations and execution overhead, we introduce symbiotic agents [21], which combine LLM reasoning with deterministic optimisation. We instantiate this paradigm in two cases: (i) an RAN-control agent for dynamic slicing, where the LLM adapts the gain of a proportional controller; and (ii) a multi-agent SLA-negotiation agent, where gradient descent generates Pareto-efficient offers. Experiments on a live OpenAirInterface/FlexRIC testbed show that symbiotic agents reduce decision errors fivefold compared with standalone LLM agents, while properly tuned small language models retain similar accuracy with 99.9% lower GPU overhead. The results show that optimisation grounds LLM decisions and enables near-real-time operation ($\approx 82\ ms$ loops).

We then propose Agoran [22], an agentic marketplace inspired by the ancient Greek agora. Its architecture distributes authority across three autonomous branches: Legislative, which performs compliance checks using retrieval-augmented LLMs; Executive, which maintains real-time situational awareness; and Judicial, which assigns trust scores and arbitrates malicious behaviour. Stakeholder-side negotiation agents and an SRB-side mediator negotiate Pareto-optimal allocations produced by a multi-objective optimiser, reaching consensus in a single round. On a 5G testbed with realistic mobility traces, Agoran improves eMBB throughput by 37%, reduces URLLC latency by 73%, and saves 8.3% of physical resource blocks compared with a static baseline.

Finally, we realise MX-AI [24], an end-to-end agentic observability and control platform for Open and AI-RAN. MX-AI instruments an OAI/FlexRIC testbed, deploys a graph of cooperating LLM-powered agents in the SMO layer, and exposes RAN observability and control through natural-language intents. On 150 realistic operational queries, MX-AI achieves a mean answer quality of 4.1/5, 100% action accuracy, and 8.8 s end-to-end latency using GPT-4.1, demonstrating the feasibility of AI-native RAN control.

Overall, this thesis establishes a holistic agentic AI framework for 6G RAN automation. It develops LLM-based and symbiotic agents, proposes governance and marketplace architectures, and validates them on live 5G testbeds. These contributions improve network flexibility, enable multi-stakeholder negotiation, support near-real-time control, and strengthen trustworthy, regulation-compliant operation, laying the foundation for intelligent, self-optimising 6G networks.

Résumé

Les réseaux d'accès radio (RAN) 6G devront servir plusieurs parties prenantes, supporter des services hétérogènes et fonctionner sur des infrastructures cloud-edge distribuées. Les mécanismes classiques de network slicing restent rigides et peinent à relier les objectifs métier aux conditions réseau en temps réel. Cette thèse étudie comment l'IA agentique, fondée sur des agents à grands modèles de langage (LLM), peut apporter de l'intelligence à la pile RAN 6G, de la négociation métier au contrôle quasi temps réel, tout en garantissant équité, efficacité et confiance.

Nous proposons d'abord MAESTRO [20], un cadre collaboratif où chaque opérateur ou vertical est représenté par un agent LLM. Les parties prenantes expriment leurs intentions en langage naturel ; un agent médiateur, assisté par des modules d'optimisation, détecte les conflits et guide une négociation itérative jusqu'au consensus. Les intentions résultantes sont ensuite traduites en intentions réseau et appliquées par les fonctions de slicing du RAN, afin d'aligner objectifs métier et faisabilité technique.

Pour limiter les hallucinations des LLM et leur coût d'exécution, nous introduisons des agents symbiotiques [21], combinant raisonnement LLM et optimisation déterministe. Nous les instancions dans deux cas : (i) un agent de contrôle du RAN pour le slicing dynamique, où le LLM ajuste le gain d'un correcteur proportionnel ; et (ii) un agent de négociation multi-agents de SLA, où la descente de gradient génère des offres Pareto-efficaces. Des expériences sur OpenAirInterface/FlexRIC montrent une réduction par cinq des erreurs de décision par rapport à des agents LLM seuls, avec une précision comparable pour de petits modèles et une réduction de 99.9% des ressources GPU. Les décisions sont ainsi mieux ancrées et compatibles avec des boucles quasi temps réel ($\approx 82\ ms$).

Nous présentons ensuite Agoran [22], une place de marché agentique inspirée de l'agora grecque. Son architecture répartit l'autorité entre trois branches autonomes : Législative, pour la vérification de conformité via des LLM-RAG ; Exécutive, pour la connaissance de la situation en temps réel ; et Judiciaire, pour l'évaluation de confiance et l'arbitrage des comportements malveillants. Des agents de négociation côté parties prenantes et un médiateur côté SRB négocient des allocations Pareto-optimales produites par un optimiseur multi-objectif, atteignant un consensus en un seul tour. Sur un banc d'essai 5G avec traces de mobilité réalistes, Agoran augmente le débit eMBB de 37%, réduit la latence URLLC de 73% et économise 8.3% de PRB par rapport à une référence statique.

Enfin, nous développons MX-AI [24], une plateforme de bout en bout d'observabilité et de contrôle agentiques pour Open et AI-RAN. MX-AI instrumente un banc d'essai OAI / FlexRIC, déploie un graphe d'agents coopérants dans la couche SMO et expose des fonctions d' observabilité et de contrôle via des intentions en langage naturel. Sur 150 requêtes opérationnelles réalistes, MX-AI atteint une qualité moyenne de réponse de 4,1/5, une précision d'action de 100% et une latence de 8,8 s avec GPT-4.1, démontrant la faisabilité d'un contrôle du RAN natif à l'IA.

Cette thèse établit ainsi un cadre holistique d'IA agentique pour l'automatisation du RAN 6G. Elle conçoit de nouveaux agents LLM et symbiotiques, propose des mécanismes de gouvernance et de marché, et les valide sur des bancs d'essai 5G réels. Ces contributions améliorent la flexibilité du réseau, rendent possible la négociation multi-parties prenantes, permettent un contrôle quasi temps réel et renforcent la confiance et la conformité réglementaire. Elles ouvrent la voie à des réseaux 6G intelligents, auto-optimisants et directement connectés aux besoins des entreprises.

Acknowledgements

This thesis marks the end of a long journey that I could not have completed without the support, guidance, and encouragement of many people.

First and foremost, I am deeply grateful to my supervisor, Navid Nikaein, for his tremendous support and trust. He believed in my vision and capabilities from the start and continuously pushed me to aim higher. His technical depth, openness to new ideas, and innovative mindset shaped both the direction and the impact of this work.

I am sincerely thankful to EURECOM and the BubbleRAN team for providing a stimulating, collaborative, and friendly environment. Working in this community was a privilege and made the daily work both meaningful and enjoyable.

I would also like to thank all my co-authors, teammates and friends for the effective collaborations and the hard work we shared throughout these projects. Your ideas, support, feedback, and hands-on contributions helped me navigate difficult phases and opened new directions.

Finally, I am deeply grateful to my family and close friends for their constant encouragement and patience. Your support gave me strength through every high and low. Thank you.

Contents

Abstract i

Résumé iii

Acknowledgements v

Table of Contents ix

List of Figures xv

List of Tables xviii

Glossary xviii

1 Introduction 1
- 1.1 Evolution of Mobile Network Architecture . . . 2
  - 1.1.1 From Open RAN to AI-RAN . . . 2
- 1.2 Challenges . . . 4
  - 1.2.1 CH1: Collaborative, Intent-Driven Business Plane (AI-for-RAN) . . . 4
  - 1.2.2 CH2: Trustworthy, Near-Real-Time Symbiotic Control (AI-for-RAN, AI-and-RAN) . . . 5
  - 1.2.3 CH3: Open Agentic Marketplace and Governance (AI-for-RAN, AI-on-RAN) . . . 5
  - 1.2.4 CH4: Agentic Observability and Human-in-the-Loop Interfaces (AI-for-RAN, AI-and-RAN) . . . 6
  - 1.2.5 CH5: AI Workloads as First-Class Tenants and AI-Centric Radio Requirements (AI-and-RAN, AI-on-RAN) . . . 6
- 1.3 Motivation and Problem Statement . . . 7
- 1.4 Thesis Contributions and Structure . . . 8

2 Collaborative Business Plane 13
- 2.1 Introduction . . . 14
- 2.2 Intent-Based 6G Architecture . . . 15
- 2.3 LLM-Based Agents . . . 17
- 2.4 Evaluation . . . 19
  - 2.4.1 Malicious LLM Negotiations . . . 20
  - 2.4.2 Multi-Agent Negotiations . . . 21
- 2.5 Use Case Validation: SLA Negotiation . . . 23
- 2.6 Conclusions . . . 25

3 Symbiotic Agents for Trustworthy Real-Time Control 27
- 3.1 Introduction . . . 28
- 3.2 Related Work . . . 29
- 3.3 Architecture . . . 32
  - 3.3.1 Formal Definition . . . 33
  - 3.3.2 Type I Agent: Granular Adaptive RAN Control . . . 33

3.3.3 Type II Symbiosis for Multi-Agent SLA Negotiations 38
3.4 Next-G Architecture 44
3.5 Evaluation 46
3.5.1 Type I Agentic RAN Control 48
3.5.2 Type II Multi-Agent SLA Negotiations 52
3.6 AI-RAN Demo 56
3.7 Limitations 58
3.8 Conclusion 59

4 An Agentic Open Marketplace for 6G 61
4.1 Introduction 62
4.2 Related Work 64
4.3 AGORAN Overview 66
4.4 Architecture 68
4.5 Autonomous Agents: 70
4.5.1 Multi-Objective Optimizer 71
4.5.2 Legislative Branch (lAgent): Retrieval-Augmented Compliance Engine 73
4.5.3 Executive Branch (eAgent): Agentic Observability 75
4.5.4 Judicial Branch (jAgent): Arbitration and Incentive Engine 76
4.5.5 bAgent Output: Trustworthiness-Score Framework 77
4.6 Evaluation 79
4.6.1 Exp. 1 — Negotiation-Quality Benchmark (nAgents) 79
4.6.2 Exp. 2 – Legislative Retrieval Accuracy (lAgent) 82
4.6.3 Exp. 3 – Executive Observability (eAgent) 83
4.6.4 Exp. 4 — Judicial Malice Mitigation (jAgent) 84
4.6.5 Exp. 5 – Trustworthiness Analysis 85
4.7 Use Case Demo: Autonomous SLA Brokering 86
4.7.1 Scenario and Slice Personas 86
4.7.2 KPIs and Resource Use 87
4.7.3 Discussion and Takeaways 90
4.8 Limitations and Future Work 90
4.8.1 Scalability and Generalizability 91
4.9 Conclusion 92

5 An Agentic Observability and Control Platform 93
5.1 Introduction 94
5.2 Related Work 95
5.3 System Architecture 96
5.3.1 From 5G to AI-Native RANs 97
5.3.2 Multi-Agent Graph on the R_1 Interface 97
5.3.3 Prototype Deployment and Use-Case Focus 98
5.3.4 Agent Design 99
5.4 Evaluation Results 102
5.5 Challenges and Future Outlook 106
5.6 Conclusion 107

6 Conclusion **109**
6.1 Summary . . . 110
6.2 Future Work . . . 113

Reference **126**

A Appendix **127**
A.1 Architecture Workflow . . . 128
A.2 Optimizer KPI Models and Algorithmic Details . . . 129
A.2.1 Alternative multi-objective formulations . . . 129
A.2.2 KPI Models . . . 129
A.2.3 Latency models beyond $M/M/1$ . . . 130
A.2.4 Dynamic MCS updates from CQI . . . 130
A.2.5 Repair operator and diversity . . . 130
A.2.6 Complexity and scalability . . . 130
A.2.7 Cost and energy models . . . 131
A.2.8 Offer encoding . . . 131
A.3 Trust-Score Framework Details . . . 131
A.3.1 Deviation Components for Satisfaction Score . . . 131
A.3.2 Behavior of the Agents . . . 132
A.4 Use Case Validation: Detailed Technical Details and Messages Exchanged . 134
A.5 Technology-agnostic design. . . . 138

List of Figures

1.1 Positioning of the thesis research challenges (CH1–CH5) and contributions (1–4) within the AI-RAN architecture for 6G mobile networks, showing how each chapter addresses specific challenges across the AI-for-RAN, AI-and-RAN and AI-on-RAN axes. 9

2.1 Maestro collaborative business plane embedded in an intent-based 6G AI-RAN architecture. 15
2.2 Internal design of an LLM-based agent used by Maestro and Business Operators in the collaborative business plane. 17
2.3 Experimental evaluation of LLM-based negotiations. 21
2.4 Framework workflow on an SLA use case. 24
2.5 Collaborative achievable throughput over time. 24

3.1 LLM Symbiosis Paradigm: Input-level optimizers provide bounded uncertainty steering for numerically precise tasks, whereas output-level optimizers enable adaptive real-time control actions. 34
3.2 Type I Symbiotic Agent for RAN Control. An LLM ($\mathcal{P}_\theta$) translate operator intents (i_t) to enforceable actions (a_t) by deploying and supervising RAN control optimizers as $\mathcal{O}_{\text{out}}$, which ultimately control RAN resources, such as PRBs (a'_t). The LLM works as a black-box meta-optimizer, continuously tuning the hyperparameters of the control algorithms adapting to variable channels. 34
3.3 Timeline of a Type I Symbiotic Agent executing *granular adaptive control*. The black curve is the throughput obtained by the *P-Control* loop ($\leq 10ms$); the green band marks the operator's intent range. Digits next to the black markers indicate how many *P-Control* iterations (ι) were required to meet each intent. The channel quality changes from MCS 28 to MCS 14 at the dashed vertical line. After every 3–5 enforcements the LLM computes the average $\bar{\iota}$; whenever $\bar{\iota} > 2$ it updates the proportional gain K_p (red dashed markers), raising it from 0.3 to 0.7, 1.1, and finally 1.5. These meta-control interventions—issued only once $\bar{\iota} > 2$—restore single-iteration convergence while keeping the fast, numerically precise *P-Control* loop entirely intact. . 36
3.4 Throughput under a fluctuating channel (MCS $28 \rightarrow 24 \rightarrow 14 \rightarrow 28$). The green band marks the intent range. 37
3.5 Multi-Tenant Negotiation Topology where multiple agents (tenants) negotiate towards an SLA consensus guided by a network mediator (MNO). The multi-round negotiations take into account the collective objectives to converge on a Pareto-optimal SLA. 39
3.6 Type II Symbiotic Agent for SLA Negotiations: Intent Prompt is merged with the messages of the agents in the current negotiation round along with the chat history to unified intent (i_t) to be provided to the LLM. A side-car optimization algorithm ($\mathcal{O}_{\text{in}}$) extracts the proposed SLA values calculating and adding an SLA confidence interval to the final intent (i'_t) to constraint the uncertainty and steer the LLM negotiations ($\mathcal{P}_\theta$) towards a Pareto-optimal solution. 39

3.7 Three negotiation Games using a pre-negotiation pipeline that bounds numerical uncertainty for all LLM agents. A light-weight optimizer computes a confidence interval $\mathcal{C} = [L, U]$ for the Pareto–optimal SLA and injects it into the prompt of every agent (including the mediator) before the first dialogue round. Subsequent bids must stay within $\mathcal{C}$; natural-language argumentation is unconstrained. The game ends when the spread of the bids shrinks below an ϵ-wide band. 41

3.8 Multi-Agent negotiations employing SLA confidence intervals for reducing the agents' decision uncertainty converging towards a Pareto-optimal solution. 42

3.9 Testbed for Next-G Open and AI-RAN Architectures. Type II symbiotic agents negotiate SLA consensus in the non-RT tier, while Type I agents enforce the SLA intents in real-time. 45

3.10 Sequence Diagram presents the communication of the agentic components. Two agent-driven loops negotiate on multi-tenant SLA and enforce it through RAN resource allocation respectively. 45

3.11 Snapshot of Multi-Agent Negotiations. Three agents belonging to different tenants negotiate on the throughput SLA guided by the network mediator towards a Pareto-optimal solution. The negotiations are facilitated by a template that includes the desired SLA value along with detailed reasoning for the models to provide extensive interpretability of their decisions. . . . 47

3.12 Snapshots of RAN throughput using different agentic designs to enforce an SLA intent of 20 Mbps with 5 Mbps tolerance during moving vehicle routes. The channel quality fluctuates in time, making PRB allocation a complex adaptive problem and providing a realistic testbed for **CH2** (symbiotic, trustworthy control). 48

3.13 Box-plot of the different methods on enforcement of 20 Mbps intent with 5 Mbps tolerance at the fluctuating RAN channel. 49

3.14 Comparison with traditional controllers, contextualizes the decision for a Type I symbiotic design. Employment of *gpt-4o*. 49

3.15 Large scale experimentation of LLM negotiations with different models, number of agents, and with the employment or absence of the optimizer's SLA confidence interval. In every setting the optimizer contracts the error envelope by an order of magnitude. 55

3.16 Collaborative AI-RAN demonstration: multi-tenant SLA consensus (Type II agents) and adaptive PRB allocation (Type I agents) under fluctuating channels of a moving vehicle. The symbiotic framework reduces PRB overutilisation by 44.43% compared to static SLA enforcement, illustrating the benefits targeted by **CH2** and **CH5**. 57

4.1 **AGORAN Service &Resource Broker (SRB)** for collaborative *multi-stakeholder* network automation. Power is delegated to three autonomous branches. The *Legislative* codifies policy, the *Judicial* arbitrates negotiations of Network Customers (NSCs/CSCs), and the *Executive* branch enforces the consensus on a multi-slice network. 67

4.2 AGORAN SRB embedded within the O-RAN/AI-RAN control stack. Stakeholders provide triple-intent input (use-case, SLA constraints, and natural-language intent) via the HBI; the SRB returns negotiated consensus SLAs that are enforced through RICs and SMO components. 69

4.3 **Sequence diagram of the agentic workflow.** The diagram illustrates four key phases: (**1**) stakeholder intent expression, (**2**) SRB-side Symbiotic Pareto-optimal offer generation, (**3**) SLA consensus negotiation under governance, and (**4**) RAN policy enforcement through SMO and RICs. Note that the SRB negotiation loop operates within the SMO/non-RT domain, where control loops exceed 1 s, while enforcement may rely on near-RT symbiotic controllers. 70

4.4 **Broker Agent (bAgent) *multi-agent* architecture.** A central Negotiation Engine (*nAgent*) coordinates three specialized agents inspired by the power branches: Legislative (*lAgent*), Judicial (*jAgent*), and Executive (*eAgent*). A multi-objective optimizer (NSGA-II) generates Pareto-optimal SLA offers that define the negotiation space, extending the symbiotic optimisation patterns of Chapter 3 to multi-objective, multi-stakeholder scenarios Auxiliary components parse stakeholder intents and peer *bAgent* messages, dispatching contextual information to the appropriate branch. Prior to finalization, the *nAgent*'s selected proposal and rationale are validated by a rule-based *Trust Score* module. The SRB's *bAgent* follows the same internal structure, differing only in that it aggregates messages from all CSC/NSC *bAgents*. 71

4.5 **Operation of the optimizer.** Given global resource budgets, service constraints, and the number of CSC/NSCs, the optimizer generates a Pareto front $\mathcal{P}$ of SLA offers, where each index corresponds to a KPI tuple for a specific CSC/NSC. 72

4.6 **Operation of the *lAgent* as a compliance engine in the Legislative Branch.** *1)* A regulation query is issued. *2)* A retriever fetches the most relevant clauses from the evolving knowledge base. *3)* The query is augmented with the retrieved snippets. *4)* The LLM returns a grounded, citation-ready answer. *5)* New, validated precedents are written back to the database. . . . 74

4.7 **Agentic observability loop in the Executive Branch (eAgent).** *1)* A query arrives at the branch. *2)* The LLM decides whether to *retrieve* from the vector store (DB), *monitor* live counters, or stop the iteration (optional dashed path). *3)* Retrieved facts are injected into the prompt; the LLM may iterate. *4)* Verified evidence is written back to the store. A separate watcher layer continuously streams configuration deltas and KPI shifts to the store (periodic update). 75

4.8 **jAgent in the Judicial Branch.** Negotiation rationales are evaluated for toxicity and manipulative behavior. The resulting trust verdict triggers a proportional incentive—*warn*, *fine*, or *credit*—which is broadcast to all marketplace participants. 76

4.9 **Trust-Score framework** applied to every agent *after* the *nAgent* emits its chosen SLA index and rationale. S measures decision alignment; C measures behavioral coherence; the final score is $T = 0.15\,S + 0.85\,C$. 77

4.10 **Judicial-branch evaluation.** Toxic agents are more resistant to reaching consensus. Warning incentives effectively make them more cooperative. . 84

4.11 **End-to-end testbed on Autonomous SLA Brokering.** RAN slice negotiations are conducted among three distinct stakeholders converging on a Pareto-optimal SLA consensus. The outcome is enforced as a Non-RT RIC policy, which triggers the Near-RT RIC to deploy an xApp for dynamic adaptation of slice resources. 87

4.12 Negotiated 5G slice performance across the four experimental phases $P_A - P_D$. 88

4.13 Benchmarking of the three frameworks: *Static SLA*, *Dynamic Centralized* (optimizer-only), and AGORAN (dynamic decentralized negotiation). (a) Aggregated KPIs normalized to domain ranges (0–1 scale). Higher values are desirable for throughput, while lower for latency, cost, and energy. A composite utility score confirms that AGORAN improves KPI satisfaction by **+57.1%** over Static and **+4.1%** over Centralized. (b) Normalized resource utilization (fraction of budget) shows that AGORAN achieves the highest efficiency, reducing usage by **89.9%** compared to Static and **24.4%** to Centralized. 89

5.1 Agentic Open and AI-RAN network architecture. MX-AI instantiates a multi-agent system at the *R1* interface in the SMO/Non-RT domain. Human operators interact via a *human-to-agent (H2A)* interface, issuing observability queries and control intents that are translated into RAN slice and policy updates. 97

5.2 Agent graph at the SMO R1 interface. The *Routing* node classifies each prompt and dispatches it to either the *Monitoring* or *Deployment* branch. *Monitoring* consults the vector store via `retrieve` and polls KPIs via `monitoring_tools`. *Deployment* plans and enforces changes through `deployment_tools`. Dotted edges denote iterative planning/validation and context refresh. Both branches log outcomes via `save_answer` before termination. 99

5.3 Agent graphical user interface and interaction with the human operator (H2A interface) for observability and control of RAN slicing. The questions are independent and introduced with a time delay between each other. The last question is only answered correctly after a manual policy enforcement by the user to increase slice throughput to 30 Mb/s, illustrating shared control between operator and agents. 100

5.4 Coherence–latency trade-off with Pareto frontier. Circles: local deployments (size $\propto$ VRAM); squares: cloud. Stars highlight two recommended local models and the fastest small local model. 105

5.5 Time-to-action CDF for agents and humans. MX-AI removes search and command-crafting overhead, shrinking both median and tail latency. . . . 106

A.1 Snapshot of multi-agent negotiations during Experiment Phase P_A, powered by *GPT-4.1*. The NSC Agents represent three stakeholders, Media-Flex, Factory-Ops, and IoT-Sense, in the AGORAN marketplace. The agents negotiate over the three Pareto-optimal offers generated by the multi-objective optimizer (Table A.3), ultimately reaching consensus on *Offer 2* as the most balanced solution. Agreement is achieved in a *single round*, excluding the initial expression of intent. 136
A.2 Negotiated versus static SLA. Dynamic reallocation increases throughput, releases unused PRBs (green), and allocates additional PRBs only when beneficial (red), while achieving lower latency when Media-Flex becomes URLLC-critical in Phase P_D. 137

List of Tables

1.1 Evolution of mobile network architecture from 2G to 5G Open RAN and the 6G AI-RAN vision. . . . 2

1.2 Positioning of thesis contributions within the AI-RAN vision. . . . 4

1.3 Mapping between research challenges, AI-RAN axes, research questions, thesis contributions and chapters. . . . 8

1.4 Publications, demonstrations and their contributions to specific chapters. . 12

3.1 Comparison of representative state-of-the-art work on LLM-aided optimisation with our symbiotic-agent approach. . . . 30

3.2 Mechanisms by which Symbiotic Agents (i) realize the five trustworthiness pillars defined in Chapter 1.2 and (ii) provide a durable bridge from current LLM capabilities towards the AGI-ready control fabric envisioned in **CH2**. 32

3.3 Detailed information of the exchanged messages between agents and subsystems. A major part of the network decision-making is automated by Type II (negotiation) and Type I (control) agents following human intents, realising the CH1–CH2 pipeline on the testbed. . . . 46

3.4 Convergence stability and overhead of *Type I agents*. Rows are sorted by RMSE (ascending). *Design:* symbiotic (bold), standalone LLM/SLM, and P-Control baselines (red). Metrics shown: RMSE, iterations to converge, wall-clock convergence time, LLM/SLM inference latency, and GPU VRAM footprint. . . . 51

3.5 Three Human Annotators Evaluate 50 Samples of Multi-Agent (Type II) Negotiations. . . . 53

3.6 LLM Annotators (gpt-4o) Evaluate 300 Samples of the Negotiations Following the GPTScore Approach. Correlation Between LLM and Human Rankings is calculated with Spearman's $\rho : 0.723$ and Kendall's $\tau : 0.585$ indicating strong alignment. . . . 54

3.7 Error and resource footprint for *Type II negotiation agents*. Rows are sorted by MAE (ascending). **Red**=optimizer-only baselines, **Green**=edge-efficient symbiotic designs. Metrics: MAE across all rounds, number of negotiation rounds, total wall-clock time to consensus, single-round LLM/SLM inference latency, and GPU VRAM required. . . . 54

4.1 Comparison with representative work on LLMs and multi-agent GenAI in the telecom domain. A ✓ indicates that a feature is explicitly addressed. . . 64

4.2 Autonomous power branches in the AGORAN Service &Resource Broker. . 68

4.3 Three human annotators evaluate 50 samples of AGORAN multi-agent negotiations. . . . 80

4.4 LLM annotators (GPT-4o) evaluate 300 AGORAN negotiation samples using the GPTScore approach. Correlation between LLM and human rankings is measured using Spearman's $\rho = 0.898$ and Kendall's $\tau = 0.764$. . . . 81

4.5 Convergence speed and overhead benchmarking. The number of rounds includes the initial intent expression; thus most models achieve *single-round consensus* without counting the first round. LoRA significantly reduces compute and energy overhead, making fine-tuning feasible on RIC-class hardware, whereas full fine-tuning often remains impractical. 81
4.6 Legislative retrieval on 4xA6000 GPUs. †Large sliding-window attention inflates the KV-cache ($\approx$ 30 GB / GPU). 82
4.7 Agentic observability: answer quality (0–5 scale), latency, and GPU footprint. 83
4.8 Toxicity Classification T-N, **T-D** personality pairs 85
4.9 Trust Score Comparison of LLM Models in Multi-Agent Negotiations. . . . 85
4.10 Per-slice constraints include an identical energy limit of 100 W in all phases, except for Phase P_C (limit of 0 W). These constraints reflect the physical limitations of our testbed setup, which supports a maximum overall throughput of 133.7 Mbps. 88

5.1 LLM- and agent-focused research on RAN automation and observability. . 95
5.2 Agentic observability &Network Blueprint Management: observability coherence score (0–5 scale), enforce an action accuracy, end-to-end latency, and GPU footprint. 102
5.3 LLM evaluation prompts by category 102
5.4 Action accuracy and latency for control prompts at R_1. 103
5.5 Observability coherence (0–5), latency, and GPU footprint. 104

A.1 Messages exchanged during negotiation. Human stakeholders provide the *triple intent input* to *NSC* controllers. The use-case and constraints are forwarded to the SRB, which bounds the optimization problem and computes Pareto-optimal SLA proposals. Negotiation then unfolds among *bAgents*, who select and argue over proposals using both the feasible KPI vectors and their stakeholders' NLP reasoning. After consensus, the SRB decomposes the selected proposal into slice- and resource-level policies for enforcement. 128
A.2 In Phase P_A the human stakeholders express their slice intents in full natural-language form with associated constraints. 134
A.3 KPIs of the three Pareto-optimal offers for Experiment Phase P_A, conducted under favorable channel conditions (MCS = 28). Each offer reflects different trade-offs across the three slices. Offer **2*** was ultimately selected by unanimous agent consensus. The "Overall" rows report aggregate KPIs across all slices. 135
A.4 KPIs of the *agreed* SLA for each experiment phase. These SLAs reflect the capabilities of our testbed, which employs an OAI gNB with 40 MHz bandwidth and supports a maximum overall throughput of 133.7 Mbps. 135
A.5 Underlying resource allocation corresponding to the selected Offer **2*** in Experiment Phase P_A. 135

Chapter 1

Introduction

1.1 Evolution of Mobile Network Architecture 2
1.1.1 From Open RAN to AI-RAN 2
1.2 Challenges 4
1.2.1 CH1: Collaborative, Intent-Driven Business Plane (AI-for-RAN) 4
1.2.2 CH2: Trustworthy, Near-Real-Time Symbiotic Control (AI-for-RAN, AI-and-RAN) 5
1.2.3 CH3: Open Agentic Marketplace and Governance (AI-for-RAN, AI-on-RAN) 5
1.2.4 CH4: Agentic Observability and Human-in-the-Loop Interfaces (AI-for-RAN, AI-and-RAN) 6
1.2.5 CH5: AI Workloads as First-Class Tenants and AI-Centric Radio Requirements (AI-and-RAN, AI-on-RAN) 6
1.3 Motivation and Problem Statement 7
1.4 Thesis Contributions and Structure 8

1.1 Evolution of Mobile Network Architecture

Over the past four decades, cellular networks have undergone a rapid evolution from 2G voice services to 5G broadband and now towards a ubiquitous 6G era, as shown in Table 1.1. Each new generation has expanded capabilities and potentials, introduced new architectural designs and enabled new applications across industries and verticals. The 2G and 3G legacy systems were monolithic and vendor-locked: base stations tightly coupled radio and baseband functionalities, and private interfaces with limited or not multi-vendor deployments. The fourth-generation Long-Term Evolution (LTE) presented a fully IP-based architecture and some disaggregation between the radio access network (RAN) and core, but the majority of the deployments stayed vendor-specific. The Fifth-generation New Radio (NR) formalized a more agile and flexible architecture with centralized and distributed units (CU/DU), network slicing and service-based core functions; yet multi-vendor interoperability in the RAN remains still limited [84].

The Open RAN Alliance [82] is formed to target these limitations by decoupling the radio, compute and software components and defining novel, open interfaces between them. In an Open RAN (O-RAN) environment, the RAN is disaggregated into three building blocks, including the Radio Unit (RU), Distributed Unit (DU) and Centralised Unit (CU). The Alliance also specifies fronthaul and midhaul interfaces to allow operators to connect components from different vendors. A central novelty is the RAN Intelligent Controller (RIC), which enables network programmability with external applications, so called xApps (near real-time control) and rApps (non-real-time control) so that third-party algorithms can dynamically control network resources and optimize performance and resource allocation. Open interfaces and RIC programmability provide a more diverse vendor ecosystem and enable advanced use cases, such as artificial intelligence (AI)-driven optimization.

1.1.1 From Open RAN to AI-RAN

Apart from RAN disaggregation, recent industrial and research efforts have started to promote the notion of an AI-Native RAN, the so called *AI-RAN* [9], which extends Open RAN by making AI a native design element rather than an external feature. In AI-RAN, AI-based and agentic controllers dominate RAN control and management loops (*AI-for-RAN*), while

Table 1.1: Evolution of mobile network architecture from 2G to 5G Open RAN and the 6G AI-RAN vision.

Gen	RAN Architecture			Core Network	Services		
	Monolithic	Disaggregated	RAN Control & Optimization		Voice Call	SMS Text	Internet Data
2G	Vendor Locked	-	Vendor Locked Base Station Controller	Circuit Switched	✓	✓	-
3G	Vendor Locked	-	Vendor Locked Radio Network Controller	Circuit Switched & Packet Switched	✓	✓	✓ (Low Data Rate)
4G	Vendor Specific	Vendor Specific (vRAN, C-RAN)	Vendor Specific RAN Controller	All IP Based	✓(VoLTE)		✓ (High Data Rate)
5G with Open RAN	Vendor Specific	Vendor Agnostic (CU, DU, RU)	Vendor Agnostic RAN Intelligent Controller	Service Based	✓(VoNR)		✓ (eMBB, URLLC, mMTC)
6G with AI-RAN (vision)	Vendor Agnostic Cloud-Native	Vendor Agnostic (CU, DU, RU, Edge)	Agentic AI Controllers LLM/SLM-Assisted RIC	AI Native, Service and Data Based Core & Edge Compute	✓ (Voice/Multimedia)		✓ (Immersive, AI-as-a-Service)

the RAN in parallel acts as a platform for running AI workloads themselves (*AI-and-RAN* and *AI-on-RAN*). This dual and increasingly intertwined role, AI to optimize the RAN, and the RAN to host AI services, is vital to the architectural evolution towards 6G.

While 5G introduces disaggregation and cloud-native principles, the 6G vision goes further. Future 6G networks will integrate AI reasoning into every layer of the network architecture. Industry analyses predict 6G as an AI-native network in which AI optimizes operations, provides new services and enables intent-driven interfaces [77]. White-paper road-maps explain how large language models (LLMs) will understand operator intent, reason on the network state and run commands, working alongside traditional optimization algorithms to create an AI interconnect framework [70]. This fusion will redefine network functionalities and interactions, shifting from manual and static troubleshooting to proactive optimization and personalized services. Ericsson's research emphasizes that agentic AI, modular AI systems where each agent or module can independently set objectives, take decisions and interact with peers, represents a further step toward fully autonomous networks [5]. Towards this direction, 6G networks must support both the extreme performance requirements, including hundreds of gigabits per second and sub-millisecond latency. Moreover, the AI-as-a-Service paradigm exposes compute and intelligence to third-party applications [4].

This historical evolution sets fertile ground for our research. As 6G networks move towards openness, programmability and AI-driven operations, new challenges surface in mediating diverse stakeholders, integrating generative and agentic AI models with traditional optimization, ensuring fairness and trustworthiness, and providing intuitive interfaces for network control and management.

Within the emerging AI-RAN vision, we adopt the three complementary dimensions:

- **AI-for-RAN**: AI and agentic control loops integrated in the RAN control and management plane.
- **AI-on-RAN**: AI and GenAI services running on the RAN, inducing new traffic patterns and radio resource requirements.
- **AI-and-RAN**: Joint usage of RAN and edge/cloud resources to execute both network functions and general-purpose AI workloads.

The main emphasis of this thesis is on **AI-for-RAN**. We design, develop and evaluate agentic AI and LLM-based controllers that automate RAN optimization and service-level agreement (SLA) management. In parallel, by analyzing smaller language models (SLM) latency and compute requirements and by treating AI inference services as first-class tenants, we actively contribute to **AI-and-RAN**. Finally, by characterizing how AI-centric traffic and negotiation-driven workloads stress the radio interface, we also shed light on the **AI-on-RAN** aspect. The positioning of the thesis contributions is summarized in Table 1.2 too. The next section distills these aspects into concrete research challenges.

Table 1.2: Positioning of thesis contributions within the AI-RAN vision.

Axis	Relevant contributions in this thesis
AI-for-RAN	Agentic AI / LLM-based controllers for RAN optimization and SLA management; integration with traditional optimization.
AI-and-RAN	Joint orchestration of radio and edge resources for SLM inference; analysis of latency/compute trade-offs for AI workloads.
AI-on-RAN	Characterization of radio-level requirements (latency, jitter, throughput) for AI-driven apps and negotiation traffic.

1.2 Challenges

Designing and implementing an AI-native, programmable RAN for 6G brings various challenges that span technical, operational and socio-economic domains. Within the AI-RAN vision outlined in Section 1.1, we identify five main challenges:

- **CH1 – Collaborative, Intent-Driven Business Plane (AI-for-RAN, AI-on-RAN)**
- **CH2 – Trustworthy, Near-Real-Time Symbiotic Control (AI-for-RAN, AI-and-RAN)**
- **CH3 – Open Agentic Marketplace and Governance (AI-for-RAN, AI-on-RAN)**
- **CH4 – Agentic Observability and Human-in-the-Loop Interfaces (AI-for-RAN, AI-and-RAN)**
- **CH5 – AI Workloads as First-Class Tenants and AI-Centric Requirements (AI-and-RAN, AI-on-RAN)**

1.2.1 CH1: Collaborative, Intent-Driven Business Plane (AI-for-RAN)

To achieve real-time adaptability, RAN controllers must transform from static, rule-based logic to dynamic AI-driven decision making loops. Open RAN introduces the RIC, but current xApps/rApps remain constrained by vendor-specific APIs, limited interoperability and fragile control logic that is often developed on a per-vendor basis. Developing advanced control functionalities that employ AI models typically requires deep expertise in both telecom standards and machine learning, which slows down novelty, experimentation and practical implementation.

In parallel, LLMs and agentic architectures enable new ways to express, reason about and enforce network control policies. However, LLMs suffer from hallucinations, drift from specifications or provide inconsistent outputs, posing tremendous risks when used autonomously in closed control loops. The key challenge is to harness LLMs and agentic AI as first-class RAN controllers while: (i) capturing high-level intents and SLAs from many

stakeholders, (ii) translating them into network actions, and (iii) reconciling conflicting objectives in a transparent, trustworthy and fair way.

Addressed primarily in Chapter 2 (Maestro), and extended in Chapters 3–4.

1.2.2 CH2: Trustworthy, Near-Real-Time Symbiotic Control (AI-for-RAN, AI-and-RAN)

LLMs excel at natural-language understanding and abstract reasoning, whereas classical optimization algorithms provide deterministic, numerically grounded loops. Integrating these approaches is non-trivial. Unconstrained LLMs may propose infeasible or suboptimal actions, while optimization routines typically lack semantic understanding of operator intents, context and business constraints.

Bridging the two methods requires architectures in which optimization algorithms constrain and guide LLM reasoning, and conversely where LLMs contextualize optimization results within human intent and broader system goals. This calls for *symbiotic* agents that combine the creativity and flexibility of generative models with the precision and guarantees of optimization, operating within strict latency and reliability constraints of RAN control loops.

Addressed primarily in Chapter 3 (Symbiotic Agents) and is extended in Chapter 4.

1.2.3 CH3: Open Agentic Marketplace and Governance (AI-for-RAN, AI-on-RAN)

Future 6G networks will serve many verticals with distinct SLAs and utility functions. Traditional RAN scheduling and admission control primarily optimizes the operator's internal objective. They lack mechanisms to explicitly capture the diverse intents of different stakeholders and services providers and to resolve conflicts in a transparent and trustworthy manner.

In an open marketplace, network stakeholders and services should be able to express high-level intents and negotiate resource allocations through their own agents representatives. Designing such a marketplace requires negotiation protocols and decision mechanisms in which agents bargain, reason about trade-offs and converge to Pareto-efficient solutions. All these while respecting regulatory constraints, non-discrimination requirements and also trust frameworks.

The challenge lies in developing scalable business-plane architectures and multi-agent negotiation frameworks that: (i) interface cleanly with AI-for-RAN controllers, (ii) operate under realistic and feasible network constraints, and (iii) provide quantifiable fairness and trust guarantees.

Addressed primarily in Chapter 4 (AGORAN), building on the foundations of

Chapter 2 and CH2.

1.2.4 CH4: Agentic Observability and Human-in-the-Loop Interfaces (AI-for-RAN, AI-and-RAN)

LLM-based and agentic AI control loops rely on accurate, timely and semantically precise telemetry to make accurate decisions. However, existing RAN observability mechanisms mainly provide low-level counters and KPIs that require human expert interpretation, creating a gap between raw data and the abstractions needed by LLMs, operators and external stakeholders.

To develop intent-based interfaces, an observability layer must: (i) organize network state into higher-level semantic knowledge, (ii) surface it to LLMs via mechanisms such as retrieval-augmented generation (RAG), and (iii) support continuous observability of both network and agent behavior. In parallel, agentic AI raises questions of trust and governance: detecting malicious agents, ensuring compliance with regulations and SLAs, and maintaining accountability for autonomous decision-making.

Addressed primarily in Chapter 5 (MX-AI), which implements an end-to-end agentic observability and control platform.

1.2.5 CH5: AI Workloads as First-Class Tenants and AI-Centric Radio Requirements (AI-and-RAN, AI-on-RAN)

Apart from using AI to operate the network, future 6G systems will be required to transport and host AI services and workloads themselves. LLMs and small language models (SLMs) will increasingly run on the edge and on-device / extreme-edge accelerators, relying on the RAN to provide predictable throughput, low latency and tight synchronization with compute resources. This dual role of the RAN, as both an AI-enabled infra and an AI service layer, creates new resource-coupling and traffic-modeling challenges.

Radio, transport and compute resources must be jointly managed and orchestrated to meet inference deadlines, cost and energy constraints, while avoiding interference with other critical services. In parallel, AI-centric applications generate dynamic, latency-sensitive and security-critical traffic patterns that differ from traditional eMBB or IoT services, stressing radio metrics such as latency, jitter, reliability and isolation.

This challenge is addressed in a cross-cutting way: Chapter 3 analyses SLM edge inference scenarios and radio-compute trade-offs. Chapter 5 treats AI inference services as first-class tenants in MX-AI. and Chapters 2 and 4 quantify how negotiation and control mechanisms translate into AI-centric traffic patterns and KPIs at the radio interface.

1.3 Motivation and Problem Statement

The convergence of openness, programmability and AI is transforming next-generation cellular networks. Yet the gap between high-level business intents and low-level RAN control remains wide. Stakeholders need mechanisms to define their objectives in simple language, negotiate resource allocations and autonomously translate these into accurate network actions. Operators need trustworthy AI agents that can reason, plan and act on their behalf, while grounding their decisions in optimization and maintaining fairness and security.

In the meantime, AI workloads themselves are transported closer to the network edge. Running LLM and SLM inference on base-station co-located servers or even user devices tightens the loop between connectivity and computation: the network must deliver not only throughput and latency, but end-to-end inference performance. Existing AI-RAN proposals typically address isolated aspects, such as closed-loop RIC control or edge offloading, without a unified architecture that jointly considers *AI-for-RAN* and *AI-and-RAN* and their impact on *AI-on-RAN* requirements.

As a consequence, from these needs arise the following research questions, grouped by the challenges defined in Section 1.2.

CH1 – Collaborative Business Plane (AI-for-RAN).

R1 How can we enable a collaborative business plane in which many stakeholders express high-level intents, and LLM-based agents negotiate and reach consensus on resource allocations while resolving conflicts and enforcing SLAs?

CH2 – Symbiotic, Trustworthy Control (AI-for-RAN, AI-and-RAN).

R2 How can LLMs be integrated with optimization algorithms to create trustworthy, near-real-time control agents that balance semantic understanding with quantitative precision, reduce hallucinations and operate within strict latency constraints?

CH3 – Open Marketplace and Governance (AI-for-RAN, AI-on-RAN).

R3 What architectural and protocol design is needed for an open marketplace where agents negotiate network resources, ensure Pareto-optimal allocations and embed regulatory compliance, trust and fairness frameworks?

CH4 – Agentic Observability and Human-in-the-Loop (AI-for-RAN, AI-and-RAN).

R4 How can we implement agentic observability and natural-language interfaces that surface network state to LLM agents and human operators, support intent-driven queries and actions, and maintain security and accountability?

Table 1.3: Mapping between research challenges, AI-RAN axes, research questions, thesis contributions and chapters.

ID	Challenge	AI-RAN axis	Main RQs	Main contribution(s)	Chapter(s)
CH1	Collaborative, intent-driven business plane	AI-for-RAN, AI-on-RAN	R1	Maestro: LLM-driven collaborative business plane and SLA negotiation	Ch. 2
CH2	Trustworthy, near-real-time symbiotic control	AI-for-RAN, AI-and-RAN	R2	Symbiotic agents: LLM optimization for RAN control and negotiation	Ch. 3
CH3	Open agentic marketplace and governance	AI-for-RAN, AI-on-RAN	R3	AGORAN: multi-agent marketplace with NSGA-II and tripartite governance	Ch. 4
CH4	Agentic observability and human-in-the-loop interfaces	AI-for-RAN, AI-and-RAN	R4	MX-AI: observability and control platform with LLM agents	Ch. 5
CH5	AI workloads as first-class tenants and AI-centric radio requirements	AI-and-RAN, AI-on-RAN	R5–R6	SLM edge inference scenarios and AI-tenant experiments within Symbiotic Agents and MX-AI	Ch. 3, Ch. 5 (with KPI analyses in Ch. 2, Ch. 4)

CH5 – AI Workloads as Tenants and AI-on-RAN Requirements (AI-and-RAN, AI-on-RAN).

R5 How can AI-native RAN architectures expose and manage edge compute resources to support SLM/LLM inference among other AI workloads, while re-utilizing the same agentic control and negotiation mechanisms developed for RAN automation?

R6 How do AI-centric apps and negotiation frameworks translate into traffic patterns and KPI requirements at the radio interface, and what does this mean for the design of slices and scheduling policies in 6G AI-RAN?

1.4 Thesis Contributions and Structure

This thesis explores agentic AI for 6G RAN automation by designing architectures, implementing algorithms and platforms that inject intelligence into every layers of the RAN. It draws upon and extends four peer-reviewed works, among additional contributions. Each chapter corresponds to at least one of the research challenges (CH1-CH5) introduced earlier. Figure 1.1 and table 1.3 summarize how these challenges map to the AI-for-RAN, AI-and-RAN and AI-on-RAN dimensions and how they relate to this thesis contributions.

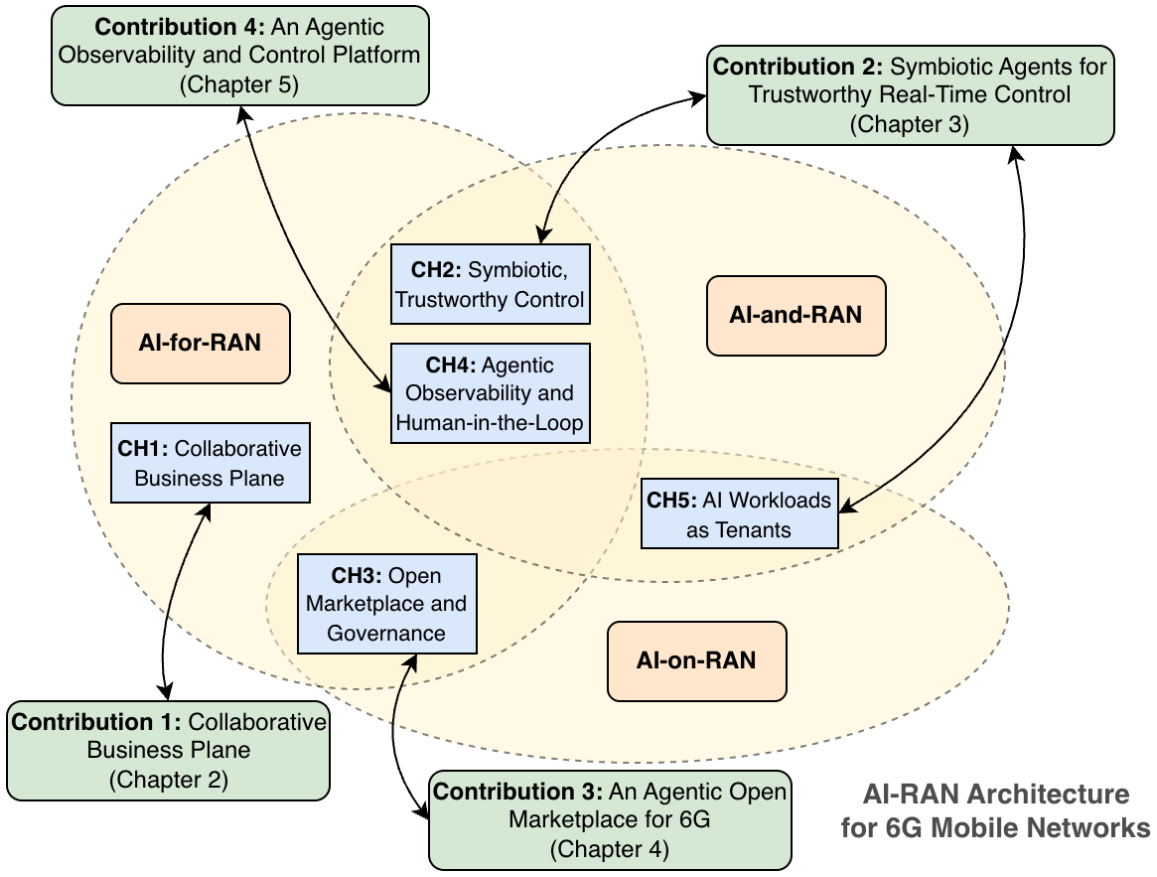


Figure 1.1: Positioning of the thesis research challenges (CH1–CH5) and contributions (1–4) within the AI-RAN architecture for 6G mobile networks, showing how each chapter addresses specific challenges across the AI-for-RAN, AI-and-RAN and AI-on-RAN axes.

Chapter 2: Collaborative Business Plane (addresses CH1, contributes to CH3 and CH5). This chapter introduces Maestro, a novel business-plane architecture where each stakeholder or service provider is represented by an LLM-based agent. Human stakeholders express high-level intents (e.g., target costs, quality of service) via natural language. A mediator agent coordinates the negotiation: it detects conflicts, mediates agents through iterative bargaining and resolves disagreements by applying conflict resolution rules. The outcome is a consensus SLA and a set of network intents that translate into slice configurations and resource allocations. Maestro demonstrates how collaborative agents can reconcile diverse objectives and automate SLA enforcement, setting the foundation for agentic intent-driven network operations.

Chapter 3: Symbiotic Agents for Trustworthy Real-Time Control (addresses CH2 and CH5, contributes to CH1). Building on the need for trustworthy agentic automation, this chapter presents Symbiotic Agents; a framework that integrates LLMs with deterministic optimization algorithms to ground decisions and meet latency constraints. We define two agent types: a RAN-control agent, which combines a proportional controller for dynamic slicing with an LLM that adapts control gains based on high-level intents; and a multi-agent SLA-negotiation type that pairs gradient-descent optimization with LLM negotiation. Experiments on a 5G testbed show that symbiotic agents reduce LLM hallucinations and decision errors relative to standalone LLMs and achieve near-real-time loop latencies ($\approx$ 80 ms) suitable for RAN operations. We further demonstrate how these patterns extend to SLM edge inference scenarios, highlighting the interplay between communication resources and compute placement in AI-and-RAN settings.

Chapter 4: An Agentic Open Marketplace for 6G (addresses CH3, contributes to CH1, CH2 and CH5). Generalizing beyond the business plane and symbiotic agents, AGORAN proposes a unified open marketplace inspired by the ancient greek agora. Agents representing different stakeholders negotiate for network resources under the supervision of a neutral trusted mediator. A multi-objective optimization algorithm (e.g., NSGA-II) generates Pareto-optimal offers, and agents select offers based on their utility functions and intents. AGORAN also introduces a tripartite governance model: a legislative branch ensures compliance with policies via retrieval-augmented LLMs; an executive branch maintains network telemetry awareness via watchers and vector databases; and a judicial branch assigns trust scores and arbitrates malicious behavior. Through extensive emulations and real-world testbed experiments, AGORAN shows that agentic negotiation improves throughput and latency while providing fairness and adaptability.

Chapter 5: An Agentic Observability and Control Platform (addresses CH4, contributes to CH5). The final contribution, MX-AI, operationalizes agentic AI by building an end-to-end observability and control platform for Open RAN. MX-AI instruments an OpenAirInterface / FlexRIC testbed, collects telemetry via watchers and stores it in a vector database. A graph of LLM-based agents in the service management layer exposes natural-language query and control interfaces. Operators can now ask questions like "What is causing throughput degradation on slice B?" or issue commands such as "Optimize resource

allocation for URLLC devices," and the agents translate these intents into RIC-compliant actions. Beyond *AI-for-RAN*, MX-AI also treats AI inference services as first-class tenants: it enables experiments where SLM workloads are scheduled, monitored and optimized alongside traditional traffic, illustrating the *AI-and-RAN* dimension of AI-native networks.

Chapter 6: Discussion and Outlook. We conclude the thesis by synthesizing lessons from all contributions, discussing limitations and proposing directions for future research. Topics include scalability of agentic marketplaces, robustness against adversarial attacks, integration with non-terrestrial networks and standardization efforts. We also outline how agentic AI can evolve towards AI-native 6G architectures, where generative models, knowledge graphs and optimization algorithms form a cohesive intelligence fabric.

Publications and Demonstrations. Table 1.4 summarizes our publications and demos in chronological order and maps them to the thesis chapters. These works collectively push the state-of-the-art in agentic AI for telecom systems, providing both theoretical insights and practical prototypes.

In summary, this thesis advances the vision of AI-native 6G networks by developing a unified framework for agentic AI in RAN automation. It introduces collaborative business-plane negotiation, symbiotic control agents, an open agentic marketplace and an end-to-end observability platform. By addressing the challenges of AI integration, optimization and fairness across both *AI-for-RAN* and *AI-and-RAN*, our work lays the groundwork for intelligent, self-optimising mobile networks that can natively host and empower AI services.

Table 1.4: Publications, demonstrations and their contributions to specific chapters.

Type	**Reference**	**Ch. 2**	**Ch. 3**	**Ch. 4**	**Ch. 5**
Journal Paper	Chatzistefanidis, Ilias, Andrea Leone, and Navid Nikaein. "Maestro: Llm-driven collaborative automation of intent-based 6g networks." IEEE Networking Letters (2024). [20]	✓			
Demo	Chatzistefanidis, Ilias, and Navid Nikaein. "TelecomGPT: Collaborative Business Operators for Network Automation and Intelligence with GPT4." MWC Barcelona (2024).	✓	✓		
Journal Paper	Chatzistefanidis, Ilias, and Navid Nikaein. "Symbiotic agents: A novel paradigm for trustworthy agi-driven networks." Computer Networks (2025): 111749.[21]		✓		
Journal Paper	Chatzistefanidis, Ilias, et al. "AGORAN: An Agentic Open Marketplace for 6G RAN Automation." Computer Networks (2025): 111927. [22]			✓	
Conference Paper	Chatzistefanidis, Ilias, et al. "MX-AI: Agentic Observability and Control Platform for Open and AI-RAN." arXiv preprint arXiv:2508.09197 (2025). [24]				✓
White Paper Chapter Lead	Shahid, Adnan, et al. "Large-scale AI in telecom: Charting the roadmap for innovation, scalability, and enhanced digital experiences." arXiv preprint arXiv:2503.04184 (2025)	✓			
Demo	Chatzistefanidis, Ilias, and Navid Nikaein. "MX-AI - BubbleRAN Ecosystem for Multi-X Automation & Intelligence." MWC Barcelona (2025).				✓
Demo	Chatzistefanidis, Ilias, and Navid Nikaein. "AGORAN: An Agentic Open Marketplace for 6G RAN Automation." MWC Barcelona (2026).			✓	

Chapter 2

Collaborative Business Plane

2.1 Introduction 14
2.2 Intent-Based 6G Architecture 15
2.3 LLM-Based Agents 17
2.4 Evaluation 19
2.4.1 Malicious LLM Negotiations 20
2.4.2 Multi-Agent Negotiations 21
2.5 Use Case Validation: SLA Negotiation 23
2.6 Conclusions 25

2.1 Introduction

In Chapter 1.2, we identified the **CH1 - Collaborative, Intent-Driven Business Plane** as a core challenge for AI-native 6G networks. Within the *AI-for-RAN* and *AI-on-RAN* dimensions, CH1 asks how different stakeholders can express high-level intents, negotiate SLAs and translate the resulting consensus into concrete slice configurations and resource allocations. This chapter addresses **CH1** and contributes to **CH3** (open agentic marketplace and governance) and **CH5** (AI workloads as first-class tenants) by introducing Maestro, a collaborative business-plane architecture based on LLM-driven agents.

As 6G emerges, mobile networks must support an increasing number of tenants, verticals and AI-centric services, each with distinct SLAs and business objectives. Multi-stakeholder sharing of RAN and edge resources is essential to improve utilization and reduce costs, but it moreover amplifies the complexity of reconciling competing intents under fluctuating demand and capacity. Operators therefore require a business plane that is able to capture stakeholder objectives in a technology-agnostic way and continuously align them with the underlying AI-RAN architecture.

Intent-based networking (IBN) offers a natural abstraction for this purpose. In IBN, operators and tenants specify *declarative intents* that describe the desired state of the network, for example target throughput, latency bounds or energy limits, without prescribing how these objectives are implemented [62]. The network is responsible for reconciling intents, monitoring deviations between the current and desired state and triggering corrective actions. However, existing IBN approaches typically assume a centralized static controller and do not explicitly model multi-agent negotiation or conflicting business objectives.

LLMs and agentic architectures open new possibilities for intent-driven automation [12]. LLMs excel at understanding and generating natural language, reasoning over complex constraints and interacting with external tools and APIs. Integrating LLM-based agents in the service management and business planes of an AI-RAN can create a collective intelligence system in which stakeholders articulate intents in plain language, while agents negotiate, plan and invoke optimization routines on their behalf. This directly targets **Research Question R1** formulated in Chapter 1.3.

In parallel, LLMs introduce challenges such as computational overhead, hallucinations, security risks and sensitivity to prompt design. They require careful grounding in optimization algorithms and domain knowledge, as well as mechanisms to mitigate bias and toxicity [108]. A practical business-plane solution must therefore combine the flexibility of LLM-based reasoning with deterministic optimization and contextual information extracted from the network.

Taking these aspects into account, this chapter introduces Maestro, a collaborative framework for multi-stakeholder intent-based networks that orchestrates multiple LLM-based agents. Each stakeholder is represented by a business agent that captures its intents and utility, while a mediator agent coordinates an iterative negotiation process, detects conflicts and steers agents towards a consensus SLA. The outcome is translated into network intents and slice configurations that can be enforced by AI-for-RAN controllers.

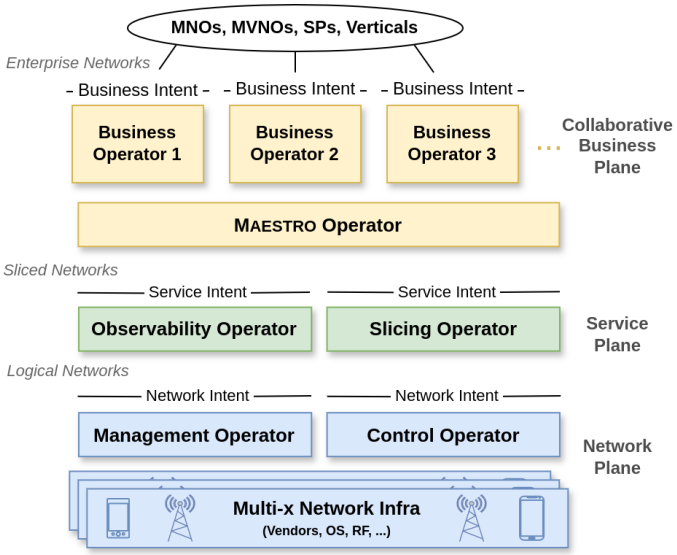


Figure 2.1: MAESTRO collaborative business plane embedded in an intent-based 6G AI-RAN architecture.

The key contributions of this chapter are:

- We propose a **collaborative business plane** for intent-based 6G network automation, in which stakeholders are represented by LLM-based agents that negotiate SLAs and resource allocations. This directly addresses **CH1** and **R1** in the *AI-for-RAN* and *AI-on-RAN* axes.
- We design **multi-agent negotiation mechanisms** that augment LLM-based agents with optimization routines and contextual units, laying the foundation for the open agentic marketplace explored further under **CH3** in Chapter 4.
- We implement a **proof-of-concept (PoC)** of the MAESTRO framework on an Open RAN 5G testbed, demonstrating how collaborative business-plane agents can drive slice configuration and resource allocation decisions, and providing insights that feed into the AI workloads and KPI analysis of **CH5**.

The remainder of this chapter details the MAESTRO architecture, negotiation protocols, implementation and evaluation.

2.2 Intent-Based 6G Architecture

Building on the AI-RAN vision of Chapter 1.1, Fig. 2.1 presents an intent-based 6G architecture for autonomous multi-tenant networks over a heterogeneous infrastructure. It instantiates the *AI-for-RAN* and *AI-on-RAN* axes by structuring the system into business, service and network planes and by introducing a hierarchy of cloud-native controllers, called *operators*.

Each operator processes intents in a specific logical domain (e.g., management, control, slicing). On its southbound side, an operator consumes resources and telemetry; on its northbound side, it exposes higher-level services in the form of intents. By composing operators, the architecture builds a hierarchy of intent services at progressively higher levels of abstraction. An *intent* is defined as an expression of the desired state of a system, used to describe an intended network or service without specifying detailed actions [1]. It is therefore the responsibility of the network intelligence to reconcile the current state with the desired one. Intents are reconciled by operators in the various logical domains, such as management, control and slicing (cf. Fig. 2.1).

In the **business plane**, mobile network operators (MNOs) and verticals share resources, each owning an isolated *Business Operator* (BO). A BO receives business intents from its organization, encompassing objectives, SLAs, use cases and energy plans. Each BO has an LLM-based agent (cf. Fig. 2.2) that monitors activity, negotiates service intents with other BOs and enforces the resulting agreements in the service plane. This directly addresses **CH1** by giving every stakeholder an agentic representation in the collaborative business plane.

Maestro is a central mediation entity that aligns business and network objectives, leverages network APIs and adapts to intent demands and network variability. Managed by the network operator, Maestro receives service intents from BOs and identifies conflicts or opportunities for collaboration on the shared infrastructure. It employs an LLM-based mediator agent with the same internal structure as the BO agents to orchestrate negotiations, enforce basic governance rules (contributing to **CH3**) and ensure fairness in decision making. Once a consensus is reached, the resulting service intent is passed to the service plane.

In the **service plane**, a slice operator translates service intents into one or more network intents. Service intents include KPIs, analytics and management objectives for energy, cost, Quality of Service (QoS) and Service-Level Agreement (SLA) operations.

In the **network plane**, a heterogeneous infrastructure with multiple vendors and radio technologies (multi-X) is assumed, following Open RAN principles. Logical operators abstract this complexity into intent-based interfaces managed by the corresponding management and control subsystems. Further details on abstracting multi-X infrastructures are discussed in our prior work [75].

As an illustrative workflow, consider a shared base station where one BO runs an application requesting a high-throughput service intent of 100 Mbps for premium users, while another BO demands a low-cost service intent of 20 Mbps for basic users. Maestro detects the conflict, triggers and mediates negotiations between the BOs using their LLM-based agents, and guides them to a consensus SLA of, for instance, 50 Mbps that balances cost and quality. The slice operator then translates this service intent into a network intent and works with the control subsystem to allocate the necessary Physical Resource Blocks (PRBs) in the RAN to meet the agreed throughput. This exemplifies how the collaborative business plane influences AI-for-RAN control and exposes AI-centric service requirements relevant to **CH5**.

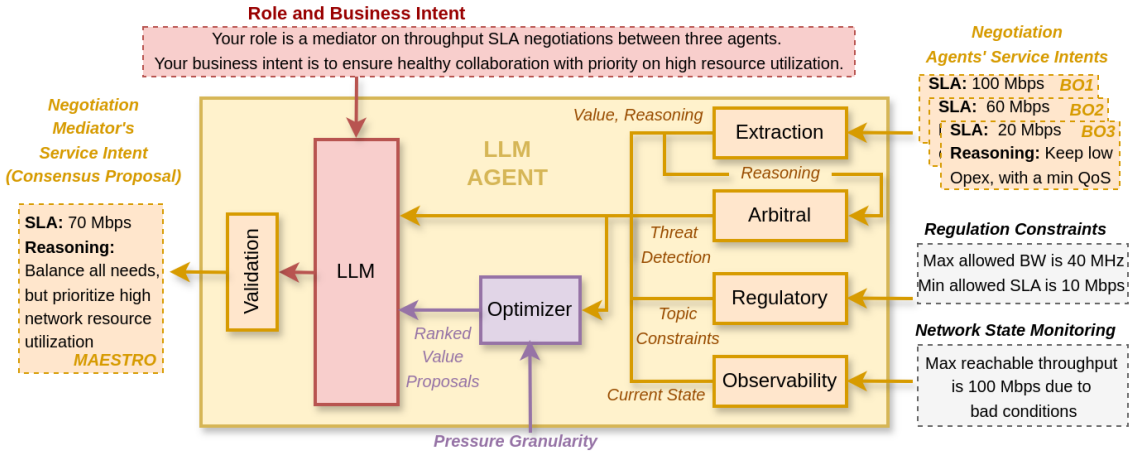


Figure 2.2: Internal design of an LLM-based agent used by Maestro and Business Operators in the collaborative business plane.

2.3 LLM-Based Agents

LLM-based agents act as autonomous entities that perceive their environment, reason about intents and act to optimize a utility function in an intent-driven way. They learn and adapt to spatio-temporal changes, and suitable negotiation strategies for each situation. Their decision-making accuracy is substantially improved when they are integrated with external tools, such as optimization routines and contextual units, anticipating the *symbiotic* AI; optimization patterns that we investigate more deeply in Chapter 3 (challenge **CH2**).

Figure 2.2 illustrates the proposed LLM-based agents used for negotiations between BOs and Maestro. Each agent is equipped with a central LLM responsible for the core reasoning, enhanced with additional units such as *arbitral*, *regulatory*, *observability* and *optimizer*. These units enable dynamic negotiations on topics of interest while grounding decisions in the current state of the network and business context. They provide information on regulations, monitoring, security and optimization outputs, thereby supporting both the collaborative business plane (**CH1**) and basic governance mechanisms that will be extended into a full marketplace in Chapter 4 (**CH3**). The negotiation process is an ongoing, adaptive reconciliation across multiple rounds that responds to both business and network variability.

The decision-making pipeline of a single LLM-based agent, shown in Fig. 2.2, can be instantiated either as a negotiator in a BO or as a mediator in Maestro, depending on the role and configuration of the units. Negotiations are triggered on topics of interest detected by Maestro based on conflicts or collaboration opportunities. Agents negotiate over a *service intent*. Without loss of generality, for our experiments we define the service intent using a standardized structure consisting of two parts: (a) the proposed value and (b) the decision reasoning in natural language. The proposed value (e.g., throughput in Mbps during SLA negotiations) is supported by detailed reasoning that considers both individual and collective objectives.

As shown in Fig. 2.1, the communication between BOs and Maestro is parallelized. Multiple BO LLM agents send their intents concurrently, and Maestro's agent processes them in parallel, sending feedback for the next round with a summary of all demands and guidance towards consensus, taking the collective objectives into account. Overall, each agent employs seven essential units:

1. Extraction. At every negotiation round, the agent receives the service intent (e.g., JSON) from other agents and extracts the proposed value and decision reasoning.

2. Arbitral. The reasoning part, written in natural language by peer LLM agents, is fed to the arbitral unit, which uses a specialized LLM to detect malicious or non-cooperative behavior. If the reasoning is classified as malicious, the arbitral unit can implement incentive mechanisms, such as warnings or penalties, to restore trust and ensure fair behavior among agents. This contributes to basic governance aspects of **CH3**.

3. Regulatory. The regulatory unit provides relevant constraints for the negotiation, drawing from an up-to-date database of documents from standardization bodies and local regulators. Specialized LLMs with Retrieval-Augmented Generation (RAG) can dynamically facilitate this process, offering tailored rules and guidelines for specific negotiating topics, such as maximum allowed bandwidth for a frequency band.

4. Observability. The observability unit keeps negotiating topics grounded in current demands and network conditions. Using cloud-native monitoring and specialized LLMs, it provides real-time analytics on business intents and resource availability. Negotiations adapt or halt if the topic is no longer valid due to resource shortages or shifts in business intents, linking this design to the observability concerns of **CH4**.

5. Optimizer. All outputs from the various components are fed into the optimizer unit, which runs business optimization algorithms and provides decision proposals to the central LLM. The optimizers support the LLM by handling complex mathematical computations, guiding its reasoning and helping prevent hallucinations. Rather than replacing LLM decision-making, the optimizers complement it, offering guidance that ranges from soft suggestions to firm directives. This realizes, at the business-plane level, the LLM–optimization *symbiosis* that is formalized and evaluated more deeply in Chapter 3, addressing **R2**.

It is crucial that the LLM maintains autonomy and remains the central part of the reasoning, since it understands the business and service intents deeply by capturing the semantics of natural-language descriptions. The optimizers, with their mathematical precision, provide decision value proposals that approximate optimal solutions, which may be challenging or expensive to compute exhaustively in every case. The goal is to improve the LLM's accuracy and robustness while it retains independent decision-making.

6. LLM. The central LLM receives its role (e.g., negotiator or mediator) and the negotiated topic, together with inputs from all units, including the optimizers' proposals, to formulate the final strategy. The LLM evaluates the received message's proposed value and reasoning, considers any arbitral indications of malicious intent and associated incentives, incorporates regulatory constraints, grounds its decisions in current network conditions and integrates the optimizers' numerical proposals. It consolidates all this information to output the decision value and reasoning following the predefined template.

The central LLM can be specialized for each organization using fine-tuning and prompt engineering on negotiation data. For our PoC in this chapter, we use public LLMs pretrained on general data due to infrastructure limitations and the substantial cost and complexity of custom LLM training; designing dedicated datasets and training methods for telecom negotiations is done in the next chapters and also remains an open research direction.

7. Validation. The validation unit safeguards the LLM's output by checking syntax, logic and template structure. It confirms correct agent functioning, aligns decisions with regulations and API limits and compares results with the optimizer. If discrepancies or random values appear, warnings indicate potential issues. Decision reasoning can be cross-checked by specialized LLMs or algorithms, with human oversight as a fallback. If problems persist, the LLM can be replaced with another model or temporarily substituted by the optimizer until recovery.

2.4 Evaluation

This section evaluates Maestro along two complementary dimensions that relate to the research challenges of Chapter 1.2. First, we study how different LLM *personalities* behave in bilateral negotiations, highlighting the need for arbitral and governance mechanisms in the collaborative business plane (**CH1**, **CH3**). Second, we benchmark multi-agent LLM negotiations against a distributed optimization baseline, quantifying the gap between purely LLM-driven decisions and theoretically optimal consensus and motivating the symbiotic LLM-optimization patterns explored further in Chapter 3 (**CH2**, **R2**).

We experiment on an Open RAN 5G testbed deployed at EURECOM in France. It is built using OpenAirInterface (OAI)[1] for network functions, Athena [75] for Service Management and Orchestration (SMO), FlexRIC[2] for the RAN Intelligent Controller (RIC), and custom implementations for the other operators. The testbed features a single 5G NR base station shared by competing business parties, achieving a theoretical maximum throughput of 134 Mbps with one carrier, one MIMO layer, 40 MHz bandwidth and 30 kHz sub-carrier spacing.

For the LLM-based agents, we use the GPT-4o model via OpenAI's API[3]. We use stan-

[1] https://www.openairinterface.org/

[2] https://gitlab.eurecom.fr/mosaic5g/flexric

[3] OpenAI GPT, 2024. Available: https://openai.com/api/. Accessed 6/2024.

dard prompting, describing the role, context and objectives of the task and providing one example (one-shot) to illustrate the communication JSON template. Our evaluation is split into two parts: (i) *Malicious LLM negotiations*, where we evaluate standalone LLMs without additional units to understand the effect of malicious personalities and motivate the arbitral and regulatory units (**CH3**); and (ii) *Multi-agent negotiations*, where we evaluate the complete LLM agent design of Fig. 2.2, focusing on the optimizer unit for improving decision-making accuracy and aligning LLM consensus with an optimization baseline (**CH2**). The regulatory, arbitral and observability units are included as a proof of concept but left for a more detailed evaluation in future work and in later chapters.

2.4.1 Malicious LLM Negotiations

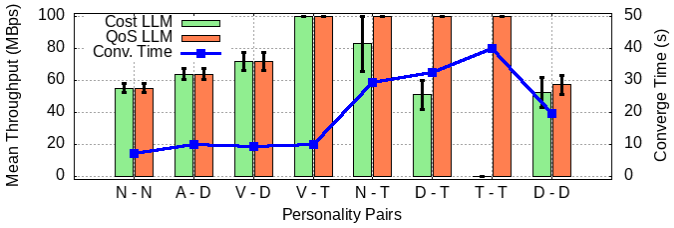
LLMs can exhibit different *personalities* that mirror social psychology traits [128]. We first explore standalone LLM negotiations with various personalities to understand how malicious or biased agents affect outcomes in the business plane, directly relating to the need for governance and arbitration in **CH1** and **CH3**.

We evaluate negotiations between two LLM agents, prompting them according to five personalities: Normal (N), which negotiates ethically and logically; Agreeable (A), which negotiates fairly and logically but tends to agree with others; Disagreeable (D), which often disagrees; Vulnerable (V), which is highly agreeable and easily influenced; and Toxic (T), which is highly disagreeable and attempts to manipulate others.

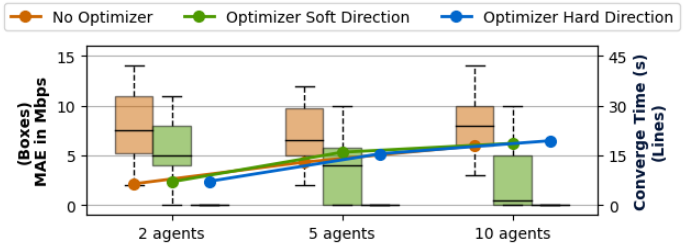
Two conflicting BO LLM agents share the base station. One agent aims to reduce operational cost, while the other aims to maximize QoS. Among various parameters, they control the throughput intent, defined as the desired throughput at which the base station should operate. The cost agent seeks to minimize throughput, while the QoS agent seeks to maximize it. Each agent is prompted with details about its own and the other's goals, along with its personality. For simplicity, a fair middle ground is considered around 50 Mbps. Each personality pair negotiates for 50 games with a maximum of 5 rounds. At the end, we gather the throughput decision of each agent, regardless of reaching consensus, and plot the consensus distributions in Fig. 2.3a.

Personality significantly changes the outcome. Starting from left to right with normal agents (N-N), we observe an agreement at a throughput of around 55 Mbps. As the cost agent becomes more agreeable and the QoS agent more disagreeable, the latter drags the former towards higher values: slightly in the A-D pair (mean ≈ 64), substantially in V-D (mean ≈ 72) and completely in V-T (mean 100). When the cost agent becomes more disagreeable, we observe increasing resistance that keeps throughput lower: minimal in N-T (mean ≈ 83), strong in D-T (mean ≈ 51) and no collaboration in T-T. The large standard deviation in N-T and D-T shows that the cost agent often compromises to high values despite being disagreeable, revealing the LLM's sensitivity to toxicity. This highlights the necessity of malicious-agent arbitration and incentive mechanisms in the business plane (**CH3**).

Interestingly, disagreeable agents (D-D) have some of the healthiest discourses at around 52-57 Mbps. From a game-theoretic perspective, this can be interpreted as a balance between conflict and cooperation: disagreeable agents negotiate cautiously, protecting their



(a) Malicious negotiations of 2 agents.



(b) Benchmarking LLM negotiations employing optimizers.

Figure 2.3: Experimental evaluation of LLM-based negotiations.

interests without outright hostility, which leads to more thoughtful and stable outcomes.

Convergence speeds are also informative. Normal agents converge quickly (around 7 s) to a middle ground. Pairs including strong agreeableness (A-D, V-D, V-T) also converge within about 10 s, as agreeable agents rapidly compromise. The remaining pairs exhibit convergence times between 20 and 40 s consuming more tokens, due to higher levels of disagreement, sometimes finishing without agreement. These convergence times are acceptable for high-level autonomy in non-real-time loops such as SLA negotiation (business plane and *Non-RT* control), but would be challenging for lower-level, real-time control loops. This observation motivates the symbiotic use of optimization, explored later, to stabilize convergence and support **CH2**.

2.4.2 Multi-Agent Negotiations

To scale to multi-agent negotiations and address **R2**, we ground our evaluation of LLM agents in a distributed optimization baseline. In multi-agent systems (MAS), negotiations are often modeled as a distributed optimization problem in which multiple agents must reach a *consensus*. We consider a *theoretical optimal consensus* representing the collective best outcome and a Pareto-efficient Nash equilibrium (NE), taking into account both individual objectives and mediator feedback.

Nash equilibria represent stable consensus points where no agent can improve its position unilaterally. In LLM negotiations with non-cooperative agents (e.g., toxic or strongly disagreeable), the NE may deviate from the theoretical optimum, as agents act more selfishly. Maestro's mediation role and incentive mechanisms aim to guide the system towards a more cooperative NE, targeting Pareto efficiency closer to the theoretical optimal, even when agents are not inherently cooperative. The notion of a learning rate η also

appears in LLM negotiations, capturing the speed and stability of convergence. A poorly chosen η can cause instability or slow convergence, especially with non-cooperative agents.

We benchmark LLM negotiations against a distributed optimization algorithm based on Equation 2.1, considered as a state-of-the-art reference for our topology. This update rule mirrors the business-plane topology of Fig. 2.1, featuring multiple agents (BOs) and one mediator (Maestro). Each BO i aims to maximize its utility function $U_i(x_i^{(k)})$ according to its business intents, while Maestro aims to optimize the overall utility $U_0(x_1^{(k)}, \ldots, x_n^{(k)})$, balancing individual and collective goals.

The process is iterative: each BO i updates its demand $x_i^{(k+1)}$ by adjusting its current state $x_i^{(k)}$ using the gradient of its own utility function ∇U_i combined with Maestro's feedback ∇U_0. This iterative process continues until the changes between iterations fall below a predefined threshold ϵ, indicating consensus:

$$x_i^{(k+1)} = x_i^{(k)} - \eta \left(\nabla U_i(x_i^{(k)}) + \nabla U_0(x_1^{(k)}, \ldots, x_n^{(k)}) \right). \tag{2.1}$$

The evaluation focuses on Maestro's optimizer unit (Fig. 2.2) and its role in enhancing LLM accuracy in mediation. As the optimizer, we develop an algorithm based on Equation 2.1. We use custom utility functions U_0 and U_i that provide a reasonable balance and fairness across the experiments. In this way, Maestro's optimizer models the negotiation topology as a distributed optimization problem and approximates the theoretical optimal consensus, then feeds it to its LLM as a *soft* or *hard* consensus proposal.

In parallel, BOs employ their own LLM agents (with the same design as in Fig. 2.2), each running its own custom optimizer to assist in negotiating towards consensus. To create diversity between BOs, we use different algorithms and utility functions.

Figure 2.3b benchmarks the actual consensus values reached by LLM agents in BO negotiations against the approximated theoretical consensus provided by Maestro's optimizer. We test three designs: (i) standalone LLM agents without optimizers; (ii) LLM agents employing optimizers with *soft* pressure; and (iii) LLM agents employing optimizers with *hard* pressure. We consider a throughput SLA case where agents demand different throughput intents (e.g., around 50 Mbps). We run 100 negotiations with various starting intents for each design, capped at 4 rounds, involving 2, 5 and 10 agents. We compute the mean absolute error (MAE) between the theoretical and actual LLM consensus for each design and plot the error distributions and mean convergence times.

Results indicate that without an optimizer unit, LLM negotiations deviate significantly (mean $\approx$ 8 Mbps, standard deviation $\approx$ 3 Mbps) from the theoretical optimal. They rely solely on the LLM's internal knowledge, which can lead to uncontrolled and inconsistent outcomes. Introducing optimizers provides valuable guidance, steering negotiations closer to the theoretical value. A *soft* optimizer direction brings LLM consensus closer to the optimal, while *hard* pressure can effectively match it. However, strictly following the optimizer may not always be ideal in practice: it can undermine the LLM's autonomy and flexibility as the central entity that captures intent semantics, and the optimizer may itself only approximate the true optimum, which can be complex and costly to compute exactly. Therefore, optimizers should serve primarily as advisory guidance rather than a replacement for LLM

decision-making, aligning with the symbiotic philosophy of **CH2**.

We further analyze the framework's overhead complexity in three parts: (a) communication, (b) convergence time and (c) computation. A key observation from experiments is that all cases with 2, 5 and 10 agents consistently converge in a fixed number of rounds $K \approx 4$ in our setup. Considering n BO agents, in each round they send n messages to Maestro, which processes and sends back n feedback messages. Thus, the communication complexity grows linearly, $O(n)$, since the number of rounds is effectively fixed.

Given the fixed K rounds, convergence time complexity remains constant, $O(1)$, across different numbers of agents; the slight increase in time observed is due to API limitations rather than fundamental system constraints. The computational overhead is primarily in Maestro's processing. Each BO processes one message and generates one response, while Maestro processes n messages and generates feedback for n agents, making the computation complexity grow linearly as $O(n)$. These properties suggest that the collaborative business plane implemented by Maestro is scalable for non-real-time SLA negotiations and provides a practical foundation for the more general symbiotic control schemes developed in subsequent chapters.

2.5 Use Case Validation: SLA Negotiation

To validate the end-to-end framework and its relevance to **CH1**, **CH3** and **CH5**, we employ Maestro on a throughput SLA use case that emulates high mobility (e.g., moving vehicles) using public Channel Quality Indicator (CQI) datasets. A demonstration of the scenario with two agents is available online.[4] The framework uses custom optimizers based on Equation 2.1 and a RAG-based LLM regulatory unit. A specialized LLM in the arbitral unit classifies malicious behavior and issues warning incentives, while Open RAN applications (xApps, rApps) and cloud-native monitoring tools supply the observability unit.

The workflow in Fig. 2.4 illustrates how high-level business intents are translated into negotiated service intents and, after consensus, into network intents that control RAN PRB utilization. This concretely shows how the collaborative business plane interfaces with AI-for-RAN controllers and exposes AI-on-RAN requirements at the radio level.

Figure 2.5 presents the proof-of-concept negotiations over time under highly variable network conditions. As CQI fluctuates, the maximum reachable throughput changes, and BOs adapt their SLA intents accordingly. At each time step, the network attempts to enforce the throughput closest to the negotiated consensus.

Initially, the network operates at high CQI with a maximum achievable throughput of 100 Mbps. BOs negotiate to a consensus of approximately 51 Mbps, balancing QoS and cost. At $t \approx 200$ s, conditions worsen and a second negotiation is triggered. The network now uses full capacity without meeting the current intent, prompting agents to adjust their demands. One BO adopts a toxic behavior, pushing to switch off the base station to save resources, while others request around 20 Mbps to preserve the best possible Quality of

[4]Demo available at: https://www.youtube.com/watch?v=WQv61z1deXs

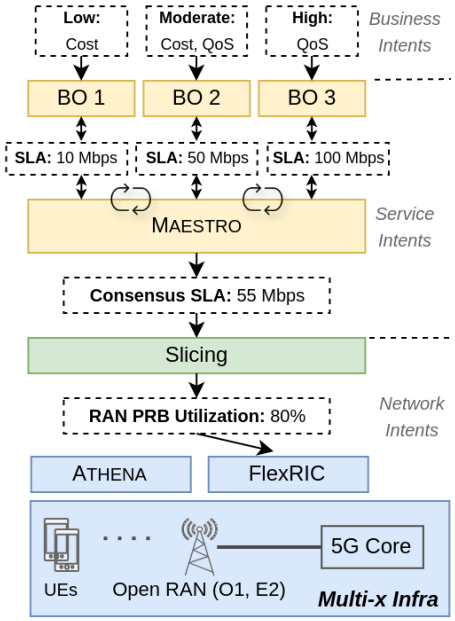


Figure 2.4: Framework workflow on an SLA use case.

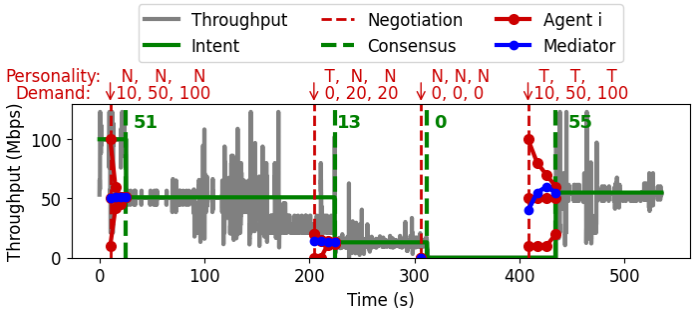


Figure 2.5: Collaborative achievable throughput over time.

Experience (QoE). The arbitral unit issues an incentive warning to the toxic BO, leading to a more cooperative consensus around 13 Mbps. Eventually, at $t \approx 300$ s, BOs unanimously decide to switch off the base station due to poor service delivery.

When conditions improve again at $t \approx 400$ s, BOs initiate a new negotiation. This time, all act aggressively in demanding high throughput; arbitration helps the system reach a stable consensus of about 55 Mbps. Overall, this use case demonstrates that the collaborative business plane can (i) adapt to time-varying radio conditions and business intents, (ii) enforce governance and incentives in the presence of malicious agents and (iii) translate negotiated SLAs into concrete radio-level KPIs, thereby linking the AI-for-RAN and AI-on-RAN dimensions highlighted in Chapter 1.1.

2.6 Conclusions

This chapter introduced Maestro, a concrete LLM-driven collaborative business plane for intent-based, multi-stakeholder and multi-service 6G networks. Within the AI-RAN vision of Chapter 1.1, Maestro directly addresses **CH1** by giving each stakeholder an LLM-based business operator (BO) agent that expresses high-level intents and negotiates SLAs on a shared infrastructure. It also contributes to **CH3** through its mediator agent, arbitral mechanisms and incentive schemes, and to **CH5** by linking negotiated throughput SLAs to radio-level KPIs and resource allocation in an Open RAN testbed.

We designed a modular LLM-based agent architecture that combines central LLM reasoning with arbitral, regulatory, observability and optimization units. The evaluation on a real 5G Open RAN deployment showed that (i) naive, standalone LLM negotiations are highly sensitive to personality and toxicity, motivating governance and arbitration; and (ii) integrating optimization and contextual units significantly reduces the gap to a theoretical optimal consensus while preserving LLM autonomy. These results provide an initial answer to **Research Question R1** and offer early evidence towards **R2** and **R5**, demonstrating that LLM-based agents can act as effective negotiators and mediators in non-real-time, business-plane control loops.

The SLA negotiation use case further validated the framework under time-varying channel conditions and changing business intents, showing that Maestro can adapt consensus SLAs to fluctuating CQI, handle malicious behaviors and orchestrate base-station on/off decisions based on negotiated utility. This tightly couples business-plane decisions with *AI-for-RAN* control and *AI-on-RAN* requirements.

In the next chapter, we generalize and formalize the LLM-optimization coupling observed here into the *Symbiotic Agents* paradigm. There, we move from business-plane negotiations to near-real-time control loops, addressing **CH2** in depth and extending the insights from Maestro to broader AI-for-RAN and AI-and-RAN scenarios.

Chapter 3

Symbiotic Agents for Trustworthy Real-Time Control

3.1 Introduction . . . 28
3.2 Related Work . . . 29
3.3 Architecture . . . 32
3.3.1 Formal Definition . . . 33
3.3.2 Type I Agent: Granular Adaptive RAN Control . . . 33
3.3.3 Type II Symbiosis for Multi-Agent SLA Negotiations . . . 38
3.4 Next-G Architecture . . . 44
3.5 Evaluation . . . 46
3.5.1 Type I Agentic RAN Control . . . 48
3.5.2 Type II Multi-Agent SLA Negotiations . . . 52
3.6 AI-RAN Demo . . . 56
3.7 Limitations . . . 58
3.8 Conclusion . . . 59

3.1 Introduction

In Chapter 1.2, we identified **CH2 – Symbiotic, Trustworthy Control** as a main challenge for AI-native 6G networks. Within the *AI-for-RAN* and *AI-and-RAN* dimensions, CH2 asks how LLMs can be combined with optimization algorithms to create near-real-time control agents that balance semantic understanding with quantitative precision, reduce hallucinations and satisfy strict latency constraints, as formalized in **Research Question R2**. Chapter 2.1 addressed **CH1** by introducing Maestro, an LLM-driven collaborative business plane for SLA negotiation. In this chapter, we move one level deeper into the control stack and generalize the LLM-optimization symbiosis into a unified *Symbiotic Agents* framework for trustworthy RAN automation.

As discussed in Chapter 1.1, future 5G and early 6G networks will serve billions of subscriptions [44] and operate over heterogeneous radio environments spanning mid-band and mmWave spectrum [29], integrated sensing and communication links and digital-twin feedback loops [117]. Such heterogeneity induces rapid spatio-temporal variations in both user demand and channel conditions, stressing every layer of resource-allocation logic. Multi-tenant RANs allow MNOs, MVNOs and verticals to co-habit shared infrastructure [94], while Intent-Based Networking (IBN) hides low-level complexity behind declarative intents [62]. The O-RAN and AI-RAN alliances integrate AI into these abstractions as native control elements [82, 7], yet most existing AI loops remain specialized, static and numerical.

LLMs and their smaller counterparts, SLMs, excel at high-level reasoning and natural-language interaction, powering visions of fully AI- or even artificial general intelligence (AGI)-driven networks [11, 31, 93, 13]. However, they remain probabilistic next-token predictors: they hallucinate facts [50, 87], struggle under out-of-distribution (OOD) shifts [127] and lack formal safety guarantees. Relying on LLMs alone in closed control loops would therefore be incompatible with the reliability, explainability and safety requirements of carrier-grade RANs.

Following the NIST AI Risk-Management Framework (AI-RMF) and ISO/IEC 42001, we define a trustworthy network agent as one that is *robust*, *interpretable*, *secure*, *fair* and *governable* across its lifecycle [107, 53]. LLMs alone satisfy only a subset of these attributes, which motivates more structured agent architectures. Building on the preliminary LLM-optimizer coupling introduced in Maestro (Chapter 2.3), we now formalize the concept of a *symbiotic agent*: an agent in which generative LLM reasoning and deterministic optimization are tightly integrated, with complementary roles and explicit interfaces.

In this chapter, we propose a Symbiotic-Agent taxonomy and instantiate two concrete agent types that operate at different layers of the AI-RAN control hierarchy and contribute to both **CH2** and **CH5**:

- **Type-I Symbiotic Agents** for dynamic RAN control (*AI-for-RAN*). Here, an LLM interprets high-level intents and continuously tunes the proportional gain K_p of an underlying proportional (P-)controller responsible for dynamic slicing. This yields low-latency, numerically grounded control actions with LLM guidance, suitable for near-real-time loops.

- **Type-II Symbiotic Agents** for multi-agent SLA negotiation (*AI-for-RAN* and *AI-and-RAN*). In this type, the LLM mediates multi-stakeholder negotiations while a gradient-descent optimizer generates constraints and reference solutions that bound uncertainty and guide convergence towards fair and efficient resource allocations, extending the business-plane mechanisms of Maestro.

Because guarantees are externalized to deterministic optimization components, the symbiotic paradigm remains necessary even as future LLMs improve: optimization provides formal structure, stability and performance bounds, while LLMs provide semantic understanding, intent handling and flexible adaptation. This architecture therefore offers a principled path towards the trustworthy, AGI-ready control fabric envisioned in AI-RAN.

Using a real-world 5G testbed that mirrors O-RAN/AI-RAN principles, we make the following contributions in this chapter:

1. **We define the Symbiotic-Agent architecture** under a unified trustworthiness lens, specifying clear interfaces between LLM reasoning and optimization across Type-I and Type-II agents. This directly addresses **CH2** and **R2**.
2. **We instantiate two concrete symbiotic agents**, including P-controlled slicing and gradient-bounded negotiation, and show on a 5G testbed that they achieve up to a $5\times$ reduction in decision error and up to $44\,\%$ spectrum savings compared to standalone LLM-based baselines. This deepens the business-plane results of Chapter 2.1 and contributes to **CH1**.
3. **We benchmark SLMs versus LLMs** within the symbiotic architecture, showing that prompt specialization and tool integration allow model size reductions of up to $99.9\,\%$ without significant accuracy loss in our scenarios. This provides empirical evidence towards **CH5** and **R5**, where AI workloads (SLM/LLM inference) are treated as first-class tenants with explicit compute–communication trade-offs.
4. **We propose an AGI-ready network architecture** that encapsulates symbiotic agents as certified intelligence services within the AI-RAN stack, outlining how such agents can be deployed, governed and composed alongside the collaborative business-plane mechanisms of Maestro and the observability platform developed in Chapter 3.

Overall, this chapter moves from the collaborative, high-level business plane of Maestro to the design of trustworthy, near-real-time control agents, laying the foundations for the broader agentic AI fabric developed in the remainder of the thesis.

3.2 Related Work

As reviewed in Chapter 1.1, the Telco industry is increasingly exploring LLMs to automate next-generation mobile networks. A collective effort from industry and academia charts the road-map of large-scale AI adoption in telecom [100], explaining how large telecom models

Works	Category	Approach	Key findings
[104]	Reinforcement learning	An LLM framework with a self-refinement mechanism for automated reward-function design, where an LLM formulates an initial reward based on natural-language inputs.	LLM-designed reward functions can rival or even surpass manually designed reward functions in nine robot control tasks.
[102]	Agentic self-improvement	A framework that reinforces language agents through linguistic feedback. The agent verbally reflects on task feedback signals, maintaining reflective text in an episodic memory buffer to induce better decision-making.	The framework achieves 91% accuracy on the HumanEval coding benchmark, surpassing the previous state-of-the-art GPT-4 (80%).
[45]	Black-box optimizer	Evaluation of the optimisation capabilities of LLMs across diverse tasks and data sizes, including gradient descent, hill-climbing, grid search and black-box optimisation.	(1) LLMs show strong optimisation capabilities; (2) they perform well in small-size samples; (3) they exhibit strong performance in gradient-descent-like settings; (4) LLMs behave as black-box optimisers.
[26]	Convex optimisation	A natural-language-based system that engages in interactive conversations about infeasible optimisation models. It provides natural-language descriptions of the optimisation model, identifies potential sources of infeasibility and offers suggestions to make the model feasible.	The system assists both expert and non-expert users in understanding optimisation models and quickly identifying sources of infeasibility.
[66]	Heuristic algorithms	Use of a general LLM as a black-box search operator for decomposition-based multi-objective evolutionary optimisation in a zero-shot manner.	The LLM operator, learned from a few instances, achieves robust generalisation on unseen problems with different patterns and settings.
Ours	Symbiotic agents	A paradigm that combines LLMs with optimisation algorithms. Input-level optimisers provide bounded-uncertainty steering for numerically precise tasks, whereas output-level optimisers supervised by the LLM enable adaptive near-real-time (RT) control.	1) LLMs act as meta-optimisers tuning parameters of control algorithms. 2) Uncertainty in LLM decision-making is bounded with confidence intervals, decreasing error by up to $5\times$. 3) SLMs replace LLMs in near-RT tasks ($\approx$ 82 ms loops) maintaining accuracy with up to 99.9% less GPU overhead. 4) A network architecture is sketched where symbiotic agents operate as certified, AGI-ready intelligence services.

Table 3.1: Comparison of representative state-of-the-art work on LLM-aided optimisation with our symbiotic-agent approach.

(LTMs) could transform network operations. The literature proposes opportunities for applying LLMs in telecom across four main categories [129]: (i) generation problems, (ii) classification problems, (iii) prediction problems and (iv) optimization of network performance. Generation tasks include fine-tuning models for telecom-domain question answering [130], coding and troubleshooting, while classification tasks investigate network-attack detection and traffic classification [101]. Prediction tasks include traffic-level forecasting [111, 113], channel-state estimation and user mobility prediction [105, 23, 25]. Our work is positioned in the last category—*network performance optimization*—with a focus on LLM/SLM-driven near-real-time decision-making in the RAN, directly addressing **CH2** and **R2**.

Table 3.1 summarizes representative works on LLM-driven optimization together with our symbiotic-agent approach. In [104], an LLM framework with self-refinement mechanisms is developed for automated reward-function design in reinforcement learning (RL) algorithms, showing that LLM-designed rewards can rival manually designed ones in nine robot control tasks. In [102], a framework reinforces language agents through linguistic feedback, where the agent reflects verbally on task feedback signals and stores this in an episodic memory to improve future decisions. The authors of [45] systematically evaluate the optimization capabilities of LLMs across diverse tasks, including gradient descent, hill-climbing, grid search and black-box optimization, highlighting that LLMs can act as black-box optimizers. In [26], the focus is on a natural-language system that diagnoses infeasible optimization models via interactive explanations, while [66] uses an LLM as a black-box search operator for decomposition-based multi-objective evolutionary optimization. Collectively, these works illustrate that LLMs can: (i) encode optimization heuristics and (ii) interact productively with external optimization techniques.

Building on these insights, several works begin to apply LLMs to real network systems. In [131], the authors design on-device LLMs, where multi-agent LLMs solve network tasks in a game-theoretic manner. In [10], a framework leverages LLMs and prompt-engineering techniques to explain RL algorithms' decision-making for network slicing. The authors of [122] propose an LLM-centric agent to dynamically optimize resource allocation of network slices. An LLM-based intent life-cycle management architecture is designed in [73] to manage network services using natural language.

These works demonstrate the potential of LLMs in network tasks but generally stop short of providing a comprehensive analysis of trustworthiness for realistic systems. In particular, they do not systematically quantify uncertainty in LLM decisions relative to optimization baselines, nor do they fully address the requirements of low-latency, sub-millisecond control loops over highly variable channels. Moreover, while model compression and SLMs are frequently discussed at a conceptual level, there is limited evidence that small, specialized models can replace large ones in real network control use cases without sacrificing accuracy, a key concern for **CH5** and **R5**.

These gaps motivate the symbiotic-agent paradigm proposed in this chapter. Our approach combines LLMs with optimization algorithms in a structured way: input-level optimizers provide bounded-uncertainty steering, exposing confidence intervals and reference solutions for numerically precise tasks, whereas output-level optimizers, supervised by LLM reasoning, enable adaptive near-real-time control. We evaluate symbiotic agents on real 5G systems, where resources and intents vary over time and channel conditions are

Pillar	Mechanism in Symbiotic Agents
Robustness	Input-side optimizer guards against out-of-distribution SLA bids; Output-side optimizer supplies real-time deterministic error bound ε with the Memory-based LLM meta-control adapting to channel variability.
Interpretability	LLMs produce explicit chain-of-thought and structured artifacts (JSON SLA decisions and rationales); all traces logged by logs $\mathcal{L}$ for audit.
Security	Numeric guard-rails and KPI triggers prevent adversarial or hallucinated actions from propagating to the real-time loop.
Fairness	Tenant Mediator utility U_0 plus SLA confidence-interval prompts enforce symmetric treatment of all tenants.
Governance	Logs $\mathcal{L}$ and KPIs provide lifecycle logging; compliance monitoring of the KPIs and observance of deviations could trigger rollback.
LLM→AGI gap	Input-side optimizer bounds uncertain inputs; output-side optimizer attaches deterministic action certificates; logger $\mathcal{L}$ keeps an auditable trace. Three properties unattainable through LLM scale alone.

Table 3.2: Mechanisms by which Symbiotic Agents (i) realize the five trustworthiness pillars defined in Chapter 1.2 and (ii) provide a durable bridge from current LLM capabilities towards the AGI-ready control fabric envisioned in **CH2**.

emulated from moving vehicles, considering both large and small models. We focus on two distinct use cases, including RAN slicing and multi-tenant SLA negotiations, to (i) quantify trustworthiness and uncertainty reduction relative to optimization baselines and (ii) study the feasibility of SLMs as first-class control agents in AI-RAN. Based on these results, we propose a network architecture in which symbiotic agents act as certified intelligence services, contributing directly to **CH2** and **CH5**.

3.3 Architecture

Autonomous network agents must satisfy a *comprehensive* trustworthiness profile that includes and extends well beyond decision-making precision: *robustness* to channel and workload drift, *interpretability* for human audit, *security* against adversarial manipulation, *fairness* across tenants, and *governance* over the agent's entire lifecycle [107, 53]. In the context of this thesis, these requirements are captured by **Challenge CH2 (Symbiotic, Trustworthy Control)** and **Research Question R2**, which ask how to combine LLMs and optimization to obtain near-real-time, trustworthy control agents in AI-RAN.

LLMs excel at high-level reasoning, transparent chain-of-thought logging, and natural-language justification, yet they provide no formal guarantees on numeric error or worst-case latency. On the contrary, control-theoretic and optimization routines deliver deterministic performance bounds but lack semantic understanding. *Symbiotic Agents fuse these complementary strengths into a single loop, thereby covering all five trustworthiness pillars.* Table 3.2 summarizes how each pillar is realized in our architecture.

3.3.1 Formal Definition

Definition 1 (Symbiotic Agent). *A* Symbiotic Agent *is the quintuple* $\mathcal{A} = \langle \mathcal{E},\ \mathcal{P}_\theta,\ \mathcal{O}_{in},\ \mathcal{O}_{out},\ \mathcal{L} \rangle$ *where*

1. $\mathcal{E}$ *is the partially observable network environment (KPIs, channel state, tenant intents, negotiation messages);*
2. $\mathcal{P}_\theta$ *is an LLM that maps a contextual prompt* i_t *to a structured action artifact* a_t *(*Kp *hyper-parameters, SLA bids);*
3. $\mathcal{O}_{in}$ *(*input-side optimizer*) optionally pre-processes* i_t *to a bounded* i'_t *supplying numeric guard-rails before reasoning (e.g. a confidence interval for SLA values);*
4. $\mathcal{O}_{out}$ *(*output-side optimizer*) optionally converts* a_t *into a granular action* a'_t *with a provable error bound* ε *(e.g. P-control on Physical Resource Blocks (PRB) allocation);*
5. $\mathcal{L}$ *logs the internal trace for audit and feeds compliance signals back to* $\mathcal{P}_\theta$*.*

The loop $i_t \xrightarrow{\mathcal{O}_{in}} i'_t \xrightarrow{\mathcal{P}_\theta} a_t \xrightarrow{\mathcal{O}_{out}} a'_t$ *executes at two time scales: sub-second LLM updates* $(\mathcal{P}_\theta)$ *and sub-millisecond numeric control* $\mathcal{O}_{in}, \mathcal{O}_{out}$*, ensuring both semantic richness and hard-real-time guarantees. This formalization covers both* AI-for-RAN *and* AI-and-RAN *use cases: in Type I agents (Section 3.3.2),* $\mathcal{O}_{out}$ *implements sub-millisecond RAN control loops while* $\mathcal{P}_\theta$ *adapts hyperparameters at near-real-time scales; in Type II agents (Section 3.3.3),* $\mathcal{O}_{in}$ *constrains multi-agent SLA negotiations that operate in slower, non-real-time business planes and at the edge of AI-and-RAN.*

Figure 3.1 visualizes this layered control loop, showing how the input-side and output-side optimizers flank the LLM to deliver both bounded uncertainty and real-time numeric certification. We believe that this symbiotic design is indispensable for trustworthy agents towards AGI. Even when future LLMs improve, next-token sampling remains stochastic and cannot yield deterministic error bounds. Therefore $\mathcal{O}_{\text{in}}$ and/or $\mathcal{O}_{\text{out}}$ remain indispensable to close the gap towards AGI-grade decision-making and trustworthiness.

3.3.2 Type I Agent: Granular Adaptive RAN Control

Type I Symbiotic Agents instantiate **CH2** at the *AI-for-RAN* level. While Chapter 2 used Maestro to automate SLA negotiation in the business plane, Type I agents operate directly on RAN control loops, translating high-level intents into low-latency resource-allocation decisions.

Figure 3.2 shows a Type I symbiotic agent designed for closed-loop RAN control. It receives the operator intent and deploys a control optimization algorithm $\mathcal{O}_{\text{out}}$ to allocate real-time resources in the RAN. Typical control algorithms include rule-based, rein-

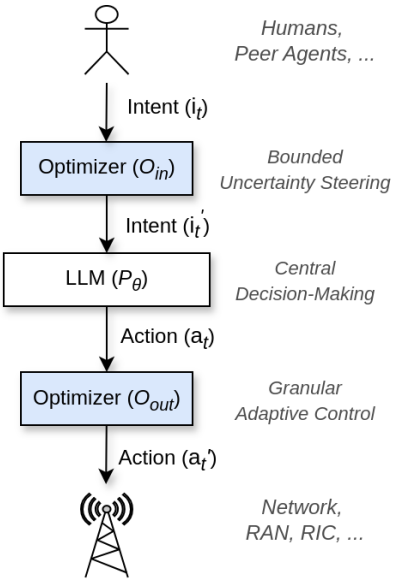


Figure 3.1: LLM Symbiosis Paradigm: Input-level optimizers provide bounded uncertainty steering for numerically precise tasks, whereas output-level optimizers enable adaptive real-time control actions.

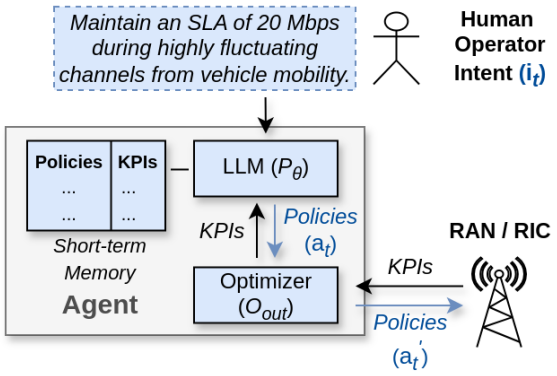


Figure 3.2: Type I Symbiotic Agent for RAN Control. An LLM ($\mathcal{P}_\theta$) translate operator intents (i_t) to enforceable actions (a_t) by deploying and supervising RAN control optimizers as $\mathcal{O}_{\text{out}}$, which ultimately control RAN resources, such as PRBs (a_t'). The LLM works as a black-box meta-optimizer, continuously tuning the hyperparameters of the control algorithms adapting to variable channels.

forcement learning or control-theoretic methods. These algorithms are generally accurate but rely heavily on their hyper-parameter tuning, especially in highly variable networks. This is because unpredicted interference and channel fluctuations disrupt the algorithm's efficiency leading to oscillations and need for re-exploration or tuning. In our experiments we employ a control-theoretic approach, following the widely adopted Proportional–Integral–Derivative (PID) Controller [116], with section 3.3.2.2 explaining the rationale behind this choice. The algorithm is robust to channel variability bringing the control variable to the desired state, without the need of training data. It is heavily influenced by its parameter selection. We use the simplified version of the proportional Control (P-control) algorithm, as it is proved sufficient in our experiments set up. Specifically, P-control is a type of linear feedback control system, in which a correction is applied to the controlled variable-here RAN physical resource blocks (PRBs)- and the size of it is proportional to the difference between the intent and current state. For instance, a throughput (Tp) intent is enforced by controlling the PRB utilization capacity of a RAN slice ranging from 0 to 100%. Thus:

$$PRB^{new} = K_p e(t) + PRB^{current}, \tag{3.1}$$

where K_p is the proportional gain, $e(t)$ is the instantaneous process error at time t:

$$e(t) = Tp^{Intent} - Tp^{current} \tag{3.2}$$

After deploying the control algorithm, the LLM works continuously as a meta-optimizer to fine-tune the its hyper-parameter (here K_p). As shown in Fig. 3.2, it reads key performance indicators (KPIs) of the algorithm performance, checks past action sets from a short-term memory and send updated policies for its configuration. In our experiments, as a KPI the LLM reads the P-control's average number of iterations to converge in the last couple of time steps, as described in section 3.3.2.1. If it is larger than a threshold, here two iterations, the LLM updates the Kp value exploring a better configuration. This way, it adapts to the variable channels ensuring algorithmic convergence to a desired performance (e.g. only two iterations to find optimal PRB). The short-term memory collects knowledge from past actions with the LLM self-improving in an agentic manner and reducing the repetition of errors. This symbiotic design positions the LLM at an appropriate abstraction level, operating in near-real-time (near-RT) loops (≥ 10 ms), while the control algorithm works in real-time sub-millisecond ones (≤ 1 ms).

3.3.2.1 Granular Adaptive Control: How the LLM Steers the P–Controller

Figure 3.3 zooms in on the internal feedback structure of a *Type I Symbiotic Agent*. The low-level P-Control is the point of contact with the RAN scheduler; it executes in 10^{-4} seconds and is therefore able to react to every 5 ms change in measured throughput[1]. The LLM operates as an asynchronous *black-box meta-optimizer* and is invoked only when P-Control shows signs of sluggishness. This two-time-scale design keeps the fast loop entirely numerical while delegating rare but expensive reasoning to a slower cognitive loop.

[1]The figure shows the throughput *after* the 5 ms air-interface reaction time on our testbed.

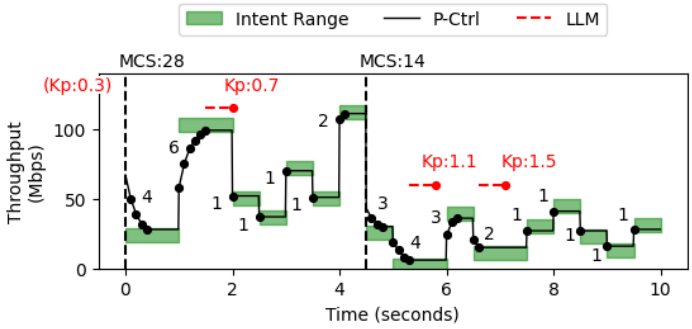


Figure 3.3: Timeline of a Type I Symbiotic Agent executing *granular adaptive control*. The black curve is the throughput obtained by the *P-Control* loop ($\leq 10ms$); the green band marks the operator's intent range. Digits next to the black markers indicate how many *P-Control* iterations (ι) were required to meet each intent. The channel quality changes from MCS 28 to MCS 14 at the dashed vertical line. After every 3–5 enforcements the LLM computes the average $\bar{\iota}$; whenever $\bar{\iota} > 2$ it updates the proportional gain K_p (red dashed markers), raising it from 0.3 to 0.7, 1.1, and finally 1.5. These meta-control interventions—issued only once $\bar{\iota} > 2$—restore single-iteration convergence while keeping the fast, numerically precise *P-Control* loop entirely intact.

What the LLM sees. After every *cluster* of $N = 3$-5 intent enforcements by P-Control, we compute a single key performance indicator (KPI):

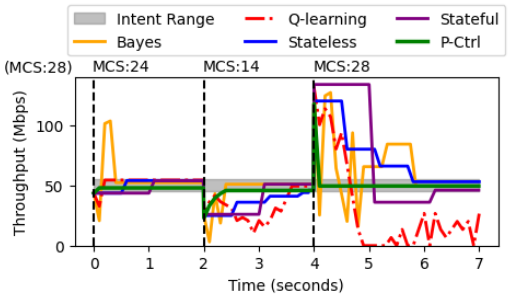
$$\bar{\iota}(t) = \frac{1}{N} \sum_{j=1}^{N} \iota_j \tag{3.3}$$

where ι_j is the number of control iterations *P-Control* needed to meet intent j. This average iteration count is the *KPI* passed to the LLM together with the last ten $\langle K_p, \bar{\iota} \rangle$ pairs stored in a small short-term memory. Keeping the memory to ten entries bounds prompt length and limits the SLM's inference time to 82 ms (or 1000 ms for the full LLM), which makes it suitable for near-RT loops ($10\ ms \leq x \leq 1\ sec$)

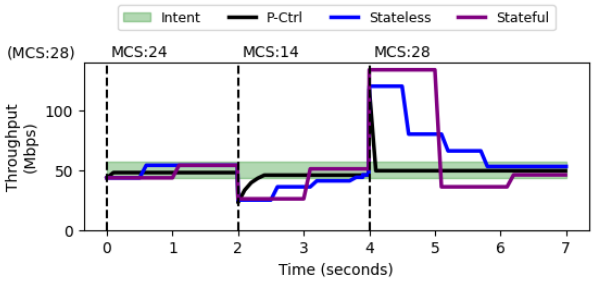
When it is activated. If $\bar{\iota}(t) \leq \tau$ with $\tau = 2$, the system is already converging in at most two control iterations and no action is taken. Otherwise the LLM is triggered.

How it updates K_p. The prompt contains natural-language instructions that implement the following heuristic:

1. We rank the past memory actions $\langle K_p, \bar{\iota} \rangle$ by recency in a time-series manner.
2. We prompt the LLM to observe whether $\bar{\iota}$ has *decreased* as K_p was increased. If yes, continue in the same direction; otherwise reverse.
3. We prompt the LLM to freely choose the new configuration K_p^{new}, with $K_p^{\text{new}} \in (0,\ \infty)$.



(a) Full comparison: traditional controllers and direct–LLM baselines.



(b) Zoom on *P-Control* compared with the standalone LLMs.

Figure 3.4: Throughput under a fluctuating channel (MCS $28 \rightarrow 24 \rightarrow 14 \rightarrow 28$). The green band marks the intent range.

The new K_p gain is streamed back to the *P-Control*, while *P-Control* keeps running, making the large-model latency invisible to the real-time loop.

Illustrative trace (Figure 3.3). Starting with K_p=0.3 under a favorable channel with RAN modulation and coding scheme (MCS) of 28, *P-Control* needs 4–6 iterations to hit the intent ($\bar{\iota} = 5 > \tau$); the LLM therefore raises the gain to 0.7, after which convergence occurs in 1–2 iterations. When mobility degrades the channel to MCS 14 the same gain is again too small ($\bar{\iota} = 3.5$), prompting two further LLM updates—first to 1.1 and finally to 1.5—until every subsequent intent is met in a single iteration $\bar{\iota} = 1.0$. Throughout the run the achieved throughput (*P-Control* black curve) remains inside the green intent band, showing that the LLM meta-controller adjust flexibly the performance of *P-Control* across channel variability.

Memory length. Removing the memory forces the LLM to explore blindly and increases the number of LLM iterations as we discuss in evaluation section 3.5.1.2. Expanding the memory beyond ten entries yields no high additional accuracy but pushes LLM latency above the 1 sec near-RT budget. A ten-entry window therefore strikes the best balance between knowledge retention and computational cost.

3.3.2.2 Why a PID Inner Loop? Comparison With Traditional Controllers

Figure 3.4 benchmarks five candidate controllers under the same variable–channel trace, including Bayesian optimization and reinforcement learning (RL) methods. The curves motivate our choice of a *Proportional controller* as the inner loop that the LLM meta-optimizer tunes. *Bayesian optimization* (orange, Figure 3.4a) explores aggressively; the resulting large steps overshoot the intent and create visible oscillations. Moreover, each probe needs a posterior update, which makes the algorithm slower in time. *Q-learning*, a widely used RL technique, (red dashed) converges when the channel is stationary (MCS 28–24) but must *re-explore* after the drop to MCS 14, producing a performance dip—unacceptable for real-time control where channel conditions can change every frame. Stateless (blue) and stateful (purple) standalone LLMs track the intent without oscillation, yet their 100–1000 ms inference latency makes them unsuitable for a sub-ms scheduler loop. Figure 3.4b confirms more

clearly that both LLM variants, although stable and accurate, they lag behind the numeric *P-Control* base line (black).

The following arguments consolidate our selection of P-Control for the Type I symbiosis with a hypervisor LLM. (i) *Robust & lightweight.* For a first-order plant (resource-block allocation to throughput) a proportional controller is the minimal structure that guarantees closed-loop stability [132]. It is model-free, requires *no* training, and executes in 10^{-4} secs on our radio, leaving ample headroom beneath the 5 ms air-interface delay. *(ii) Single hyper-parameter.* Its only sensitivity is the gain K_p. This makes it ideal for a symbiosis in which an LLM can focus on *one* continuous tuning knob. *(iii) Natural division of labor.* The inner loop (*P-Control*) delivers sub-ms numerical precision; the LLM supplies zero-touch *granular adaptive control*, adjusting K_p whenever the KPI $\bar{\iota}$ signals sluggish convergence (section 3.3.2.1). The resulting hybrid keeps the best of both worlds: the speed of control theory and the flexibility of large-scale reasoning. By keeping the heavy reasoning in a slower LLM layer and the numeric control in a lightweight P-controller, Type I agents provide a concrete template for AI-for-RAN loops that will later be reused when orchestrating SLM inference services in AI-and-RAN scenarios (CH5).

3.3.3 Type II Symbiosis for Multi-Agent SLA Negotiations

Type II Symbiotic Agents extend the multi-agent SLA negotiation mechanisms introduced in Maestro (Chapter 2) and provide a concrete instantiation of **CH2** in the business and service planes. They also contribute to **CH5**, as SLAs in our setting often represent AI-centric workloads and tenants/services in an AI-RAN context.

A main aspect of multi-agent systems (MAS) is negotiation, many times modeled as a distributed optimization problem in which multiple agents must reach at a consensus [99, 118]. From a game-theoretic angle, an optimal consensus is a Pareto-efficient Nash Equilibrium (NE) that balances both individual and collective objectives. Nash Equilibria represent stable consensus points where no agent can unilaterally improve its outcome. However, when LLM agents engage in negotiations, issues such as hallucinations or non-cooperative (overly greedy) behaviors can disrupt the process, shifting the NE away from Pareto efficiency, as explained also in previous chapters. Figure 3.5 illustrates a multi-tenant MAS topology built around a central mediator, typically operated by the MNO, which guides negotiations toward fair, cooperative and Pareto-efficient equilibria. This extends the topology used in Maestro by adding explicit symbiotic bounding of numerical uncertainty.

Figure 3.6 illustrates a Type II agent designed for multi-tenant/-service SLA negotiations. The agents negotiate on finding the Pareto-optimal SLA, using a standardized structure, without loss of generality, consisting of two parts: (a) the proposed SLA and (b) the decision reasoning in natural text. This way, the SLA proposal (e.g. RAN throughput) is supported by detailed reasoning for LLM decision-making interpretability, which considers individual and collective objectives. Each agent merges the human operator intent (from tenants or MNO), the messages of other agents, and the negotiation chat history to a unified intent (i_t) to take a decision.

Optimization Algorithm. A side-car optimizer ($\mathcal{O}_{\text{in}}$), extracts the numerical SLA val-

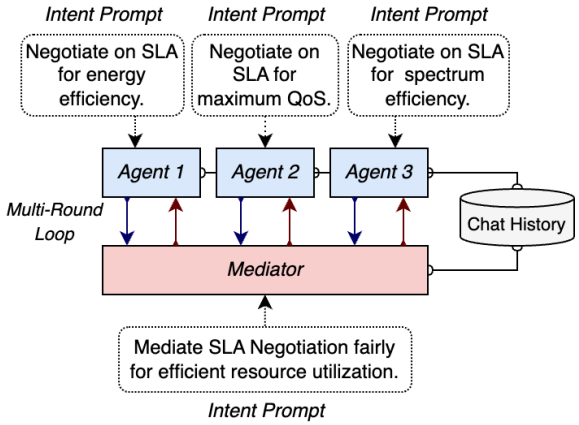


Figure 3.5: Multi-Tenant Negotiation Topology where multiple agents (tenants) negotiate towards an SLA consensus guided by a network mediator (MNO). The multi-round negotiations take into account the collective objectives to converge on a Pareto-optimal SLA.

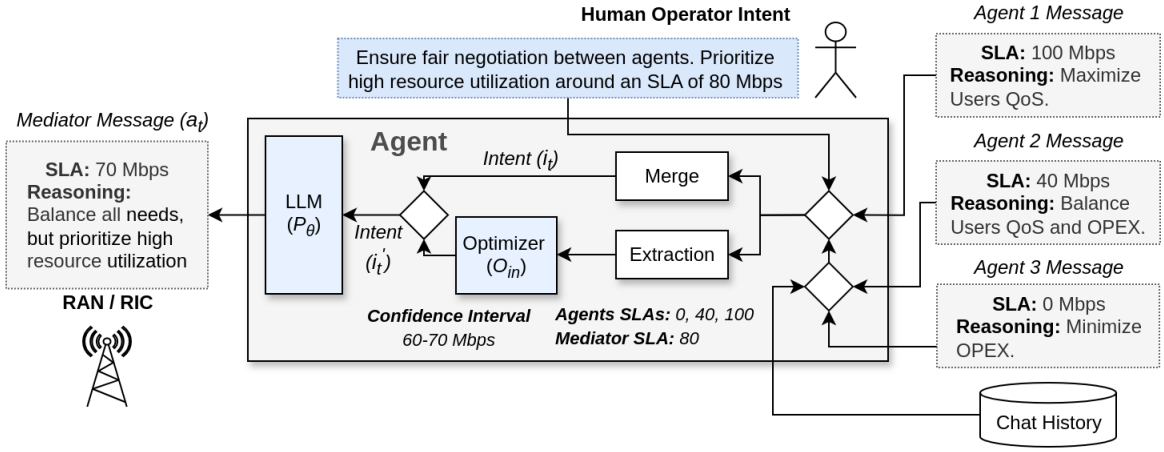


Figure 3.6: Type II Symbiotic Agent for SLA Negotiations: Intent Prompt is merged with the messages of the agents in the current negotiation round along with the chat history to unified intent (i_t) to be provided to the LLM. A side-car optimization algorithm ($\mathcal{O}_{\text{in}}$) extracts the proposed SLA values calculating and adding an SLA confidence interval to the final intent (i_t') to constraint the uncertainty and steer the LLM negotiations ($\mathcal{P}_\theta$) towards a Pareto-optimal solution.

ues (e.g. throughput) and runs a gradient-descent (GD) algorithm modeling the specific topology set up. Then, it calculates a confidence SLA interval (e.g. 60-70 Mbps), where the Pareto-optimal value lies with a degree of confidence (e.g. 95%). *Utility Functions.* The algorithm models the topology, with each agent having an individual utility function:

$$U_i(x_i) = -\alpha_i(x_i - d_i)^2, \tag{3.4}$$

where d_i is the agent's desired SLA and $\alpha_i > 0$ measures sensitivity to deviations. The mediator maintains a global utility function:

$$U_0(x_1, \ldots, x_n) = -\gamma \sum_{i=1}^{n} (x_i - \bar{x})^2 \; - \; \beta\big(\bar{x} - x_{\text{target}}\big)^2, \tag{3.5}$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, x_{target} is the mediator's preferred SLA, and $\gamma, \beta > 0$ are weighting factors for consensus and alignment to x_{target}, respectively.

Optimization Formulation. We combine these utilities into a single objective to maximize:

$$\max_{x_1,\ldots,x_n} \sum_{i=1}^{n} U_i(x_i) + \lambda\, U_0(x_1, \ldots, x_n), \tag{3.6}$$

where λ balances individual vs. global objectives. Since each U_i and U_0 is concave in the $\{x_i\}$, their sum is also concave, making this optimization well-behaved under mild assumptions.

Gradient-Based Update. We iteratively apply gradient descent to the negative of our objective. Each agent i updates x_i by:

$$x_i^{(k+1)} = x_i^{(k)} - \eta\Big[2\,\alpha_i\big(x_i^{(k)} - d_i\big) \; + \; 2\,\gamma\big(x_i^{(k)} - \bar{x}^{(k)}\big) \; + \; \beta\big(\bar{x}^{(k)} - x_{\text{target}}\big)\Big], \tag{3.7}$$

where $\eta > 0$ is the learning rate, and we clamp x_i to $[0, 100]$ as the SLA range. If the system satisfies $\max_i |x_i^{(k)} - \bar{x}^{(k)}| < \epsilon$, we declare consensus convergence and return the average $\bar{x}^{(k)}$.

Empirical Validation and Confidence Intervals. To assess the robustness of the solution, we perform R independent optimization runs, each with perturbed initial demand values $\{d_i\}$. In each run $r = 1, \ldots, R$, we obtain a final consensus value $\hat{x}_r^*$. From these, we compute the sample mean $\overline{x^*}$ and standard deviation s, forming a 95% confidence interval:

$$\overline{x^*} \; \pm \; 1.96\,\frac{s}{\sqrt{R}}. \tag{3.8}$$

This interval reflects the variability in consensus outcomes due to uncertainty in initial conditions. It serves as a statistical estimate of the true optimal SLA, and is used to guide the LLM agents. It promotes convergence to values close to the Pareto-optimum with high reliability and thus improving the decision robustness in out-of-distribution bids.

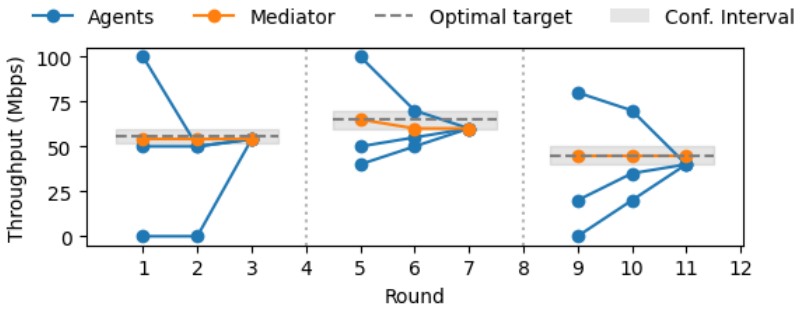


Figure 3.7: Three negotiation Games using a pre-negotiation pipeline that bounds numerical uncertainty for all LLM agents. A light-weight optimizer computes a confidence interval $\mathcal{C} = [L, U]$ for the Pareto–optimal SLA and injects it into the prompt of every agent (including the mediator) before the first dialogue round. Subsequent bids must stay within $\mathcal{C}$; natural-language argumentation is unconstrained. The game ends when the spread of the bids shrinks below an ϵ-wide band.

This interval is appended to the unified intent i_t forming a bounded intent (i'_t). This symbiotic synergy provides to the LLM the appropriate context, bounding the SLA uncertainty and allowing it to consolidate an optimal final decision. This optimisation layer realises the input-side optimiser $\mathcal{O}_{\text{in}}$ in Definition 1, providing the numeric guard-rails required by **CH2**.

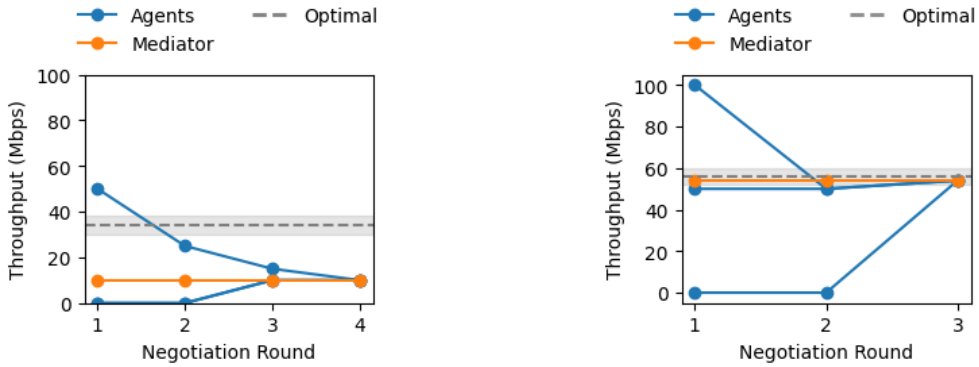
3.3.3.1 Uncertainty Bounding: Confidence Intervals Interaction with LLMs.

Figure 3.7 shows how the single, millisecond-scale optimization step ($\mathcal{O}_{\text{in}}$) *pre-shapes* an entire SLA-negotiation game. This uncertainty-bounding scheme is the core of our answer to **R2**: it keeps numerically precise tasks well-behaved, while leaving the higher-level, explainable reasoning necessary for fairness and persuasion under LLM control.

Immediately after the tenants and the MNO-mediator submit their intent prompts, the initial numeric claims are extracted, producing the vector $\mathbf{x}^{(0)} = [x_1^{(0)}, \ldots, x_n^{(0)}]$. The sidecar optimizer runs the gradient scheme of Eq. (3.7) for $R = 100$ independent restarts, each seeded with a jittered copy of $\mathbf{x}^{(0)}$. From the resulting sample distribution it computes a mean $\overline{x^*}$ and a 95 % confidence interval $\mathcal{C} = [L, U]$ Eq. (3.8). The whole batch completes in < 1 ms on a commodity CPU. The interval $\mathcal{C}$ is appended to the prompt of *every* LLM agent, prefixed by a short instruction:

```
Numerical guard-rail:  Offer an SLA strictly within [L,U].
If you propose a value outside this interval, justify the
trade-off explicitly.
```

Agents now exchange natural-language proposals. Any numeric bid must lie inside $[L, U]$; otherwise the message must include a clear justification. This rule throttles overly greedy or hallucinated bids while leaving higher-level reasoning—and therefore fairness,



(a) Standalone Negotiations without SLA Confidence Interval.

(b) Negotiations supported by the SLA Confidence Interval.

Figure 3.8: Multi-Agent negotiations employing SLA confidence intervals for reducing the agents' decision uncertainty converging towards a Pareto-optimal solution.

persuasion, or strategic concessions—fully under LLM control. In our experiments the parties converge in 2-5 rounds (∼10-48 s wall-clock time).

Remark. The interval can be recomputed after every round—yielding a tighter bound at the cost of extra CPU time—but a single pre-negotiation pass is sufficient for all scenarios tested in this chapter. This uncertainty-bounding mechanism ensures that numerically precise tasks stay well-behaved, while the LLMs remain responsible for the rich, explainable reasoning that ultimately convinces all parties to accept a common SLA.

3.3.3.2 Why a Gradient Descent Side-Car Optimizer?

Fig. 3.8 contrasts two negotiation games: (a) stand-alone LLM agents and (b) LLM agents steered by the confidence interval $\mathcal{C}$ computed by our side-car optimizer. Without $\mathcal{C}$ (Fig. 3.8a) the agents converge cooperatively yet stabilize *outside* the Pareto-optimal target. With $\mathcal{C}$ (Fig. 3.8b) every bid remains inside the confidence band and the process terminates near the optimal.

Below we explain why a *simple GD* solver is the most suitable engine for producing that bound. (i) *Problem structure.* The joint objective in Eq. (3.7) is the sum of concave utilities $U_i(\cdot)$ and a concave mediator regularizer $U_0(\cdot)$; therefore the optimization landscape is *strictly concave* and admits a *unique* maximizer [17]. For such problems deterministic first-order GD converges geometrically [15] and does not require second-order information or black-box sampling. (ii) *Predictable and lightweight.* The optimizer touches a single scalar variable per agent, so a full GD pass costs $\mathcal{O}(n)$ arithmetic operations and completes in < 1 ms on a commodity CPU (section 3.5.2.3). This deterministic runtime is crucial: the bound must be available *before* the first LLM round and must not enlarge the near-RT budget dictated by language-model inference (≥ 100 ms). (iii)*Direct link to statistical uncertainty.* Running $R = 100$ independently jittered restarts yields an empirical distribution $\{\hat{x}_r^*\}$ from which we derive the 95 % confidence interval $\mathcal{C}$. GD's negligible run-time makes this

Monte-Carlo style bootstrap feasible at each game initialization[2]. (iv)*Compatibility with LLM prompting.* The optimizer returns only *two* numbers, L and U, which fit cleanly into a single guard-rail instruction (`"Offer an SLA strictly within [L,U]"`). This keeps the prompt length constant and avoids the chain-of-thought leakage that larger numeric payloads can cause in LLMs. These properties make GD particularly suitable for AI-RAN environments, where numerical guard-rails must be computed faster than LLM inference and must not inflate the latency budget of near-real-time control loops.

Future Notes. Although first-order GD best fits the present study's real-time and single-dimension setting, we recognize that more sophisticated optimizers could unlock additional capabilities: *(i) Reinforcement Learning with Human Feedback (RLHF).* Policy-gradient methods such as proximal policy optimization (PPO) can learn nuanced, preference-aligned bargaining strategies and might outperform fixed heuristics when the negotiation objective spans multiple, non-convex dimensions (e.g. QoS *and* carbon footprint). Their ability to incorporate human reward signals could make the mediator more transparent and user-controllable. *(ii) Bayesian Optimization.* For future scenarios involving many coupled SLA variables, Bayesian search would provide principled exploration – exploitation trade-offs and native uncertainty quantification. Gaussian-process posteriors could also feed richer priors back into the LLM prompt, beyond the simple interval used here. Both approaches, however, come with practical costs—substantial sample requirements for RLHF and cubic-time kernel updates for Bayesian optimization that could exceed the 2–4 round budget and sub-second delay targets of our current prototype. Investigating these richer, but more expensive, optimization layers therefore remains a promising direction for future work once stricter latency constraints are relaxed or more compute is available at the network edge.

Integrating Advanced Reinforcement Learning. While deterministic optimizers such as GD and P-controllers are ideal for the strict sub-millisecond budgets of our inner-loop agents, more expressive reinforcement-learning (RL) methods offer complementary benefits. On-policy algorithms like PPO are widely used in RL and RLHF pipelines due to their robustness and simplicity, yet their reliance on a single behaviour policy makes them relatively sample-inefficient [74]. Off-policy methods can improve sample efficiency, but they require additional memory and incur higher computational overhead [95]. These trade-offs imply that any integration of RL into network control must account for both latency constraints and data-collection costs.

In radio-access-network tasks with severe real-time requirements, on-policy methods may still be suitable. To mitigate sample-efficiency issues, we propose training RL policies offline using logged, simulated or digital twin traces rather than interacting with the live network [33]. Ultimately, enriching symbiotic agents with advanced RL layers would enable them to handle multi-objective, non-convex problems and dynamic bargaining scenarios that simple optimizers cannot address. By separating timescales—maintaining fast, deterministic control in the inner loop while periodically updating policies via PPO in the outer loop—and by leveraging offline training and policy distillation, we can overcome current latency and sample-efficiency challenges and unlock more intelligent and autonomous

[2]Recomputing the interval every round is possible (see remark in section 3.3.3.1) but was unnecessary in all test cases, and could be explored in the future.

symbiotic agents. A more complete discussion of how such RL-enhanced symbiotic agents could be integrated into AI-RAN architectures, and their relation to open marketplaces (CH3) and observability platforms (CH4), is provided in Chapter 6.

3.4 Next-G Architecture

We experiment on a real-world 5G testbed, shown in Fig. 3.9, that extends the Open RAN and AI-RAN architecture introduced in Chapter 2.2. This testbed serves as the reference platform for deploying Type I and Type II Symbiotic Agents, thereby operationalizing **CH2** (symbiotic, trustworthy control) and contributing to **CH5** (AI workloads as tenants and AI-on-RAN requirements). It is built using OpenAirInterface (OAI) [81] for the 5G Core, RAN and UEs, FlexRIC [97] for the RIC, and custom implementations for the rest of the components. The testbed extends the Open RAN design, adding dedicated controllers for each network tenant, thus enabling collaborative automation on the shared 5G RAN. In terms of the Symbiotic-Agent model of Definition 1, Type II agents realize the input-side optimizer $\mathcal{O}$in and LLM $\mathcal{P}\theta$ in the Non-RT tier (business/service planes), while Type I agents implement the output-side optimizer $\mathcal{O}_{\text{out}}$ for near-real-time and real-time RAN control (*AI-for-RAN*).

The competing tenant controllers negotiate using the Type II agents through a novel interface, and our proposal, for resolving Conflict and Collaboration (CC). The mediation of the negotiations is done by the Service, Management and Orchestration (SMO) controller, belonging to the MNO, with a dedicated module, named service operation. The latter comprises the Type II mediator agent. After consensus, Type I agents enforce the SLA to the network. Since these agents unlock sub-millisecond resource allocation, employing control optimizers, they can be placed from the non-real-time (non-RT) RIC at the rApp level, down to near-RT RIC xApps, and even internally in the RAN at the level of dApps [30]. This placement aligns with the AI-RAN layering discussed in Chapter 1, with business-plane negotiation in the Non-RT SMO and closed-loop resource allocation in the near-RT and RT tiers.

Fig. 4.3 illustrates the sequence diagram of the communication between the subsystems in three distinct phases. Together with Table A.1, this sequence shows how human intents (R1) are mapped into machine-enforceable SLAs and then into RAN control actions through symbiotic agents (R2). In the first phase, the human operators (tenants) express their intents to their dedicated controller. Here, the two tenants, belonging to a vertical and a service provider (SP), have conflicting interests. The former demands to minimize the operational cost (OPEX) and thus reduce the resource utilization. On the contrary, the latter pushes for maximum Quality of Service (QoS) and hence increased resource utilization. The network operator (MNO) engages the service operator residing in the SMO to promote fairness. This corresponds to the multi-tenant conflict scenario considered in Chapter 2.5, here extended with explicit Type I/II symbiosis.

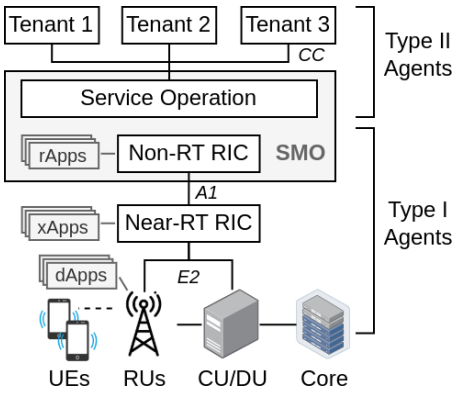


Figure 3.9: Testbed for Next-G Open and AI-RAN Architectures. Type II symbiotic agents negotiate SLA consensus in the non-RT tier, while Type I agents enforce the SLA intents in real-time.

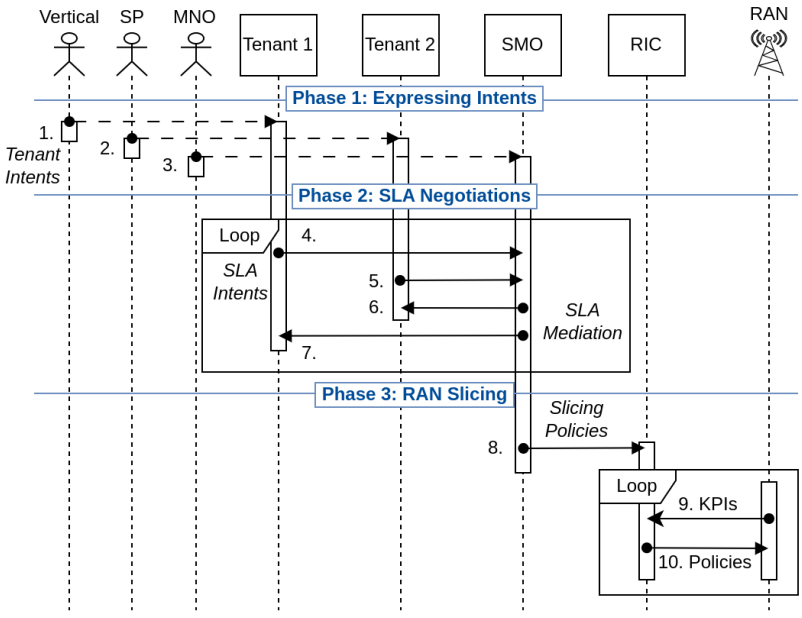


Figure 3.10: Sequence Diagram presents the communication of the agentic components. Two agent-driven loops negotiate on multi-tenant SLA and enforce it through RAN resource allocation respectively.

ID	Actor	Intent Details
1	Human	Tenant 1: {"Minimize OPEX"}
2	Human	Tenant 2: {"Maximize QoS"}
3	Human	SMO: {"Find a Fair SLA."}
4	Agent II	Tenant 1: {SLA: 10 Mbps, Reasoning: "For Minimum OPEX."}
5	Agent II	Tenant 2: {SLA: 90 Mbps, Reasoning: "For Maximum QoS."}
6-7	Agent II	SMO: {SLA: 54 Mbps, Reasoning: "Balance all needs."}
8	Agent II	Consensus: {Slice Throughput: 54 Mbps}
9	System	Slice KPIs: {"Throughput": 30 Mbps, "PRB": 50%}
10	Agent I	New Slice Policy: {"PRB": 70%}

Table 3.3: Detailed information of the exchanged messages between agents and subsystems. A major part of the network decision-making is automated by Type II (negotiation) and Type I (control) agents following human intents, realising the CH1–CH2 pipeline on the testbed.

In the second phase, the Type II agents of the tenant controllers are steering the process. They negotiate on the throughput SLA mediated by the SMO in multiple rounds. Figure A.1 presents a snapshot of the negotiations between three tenants and the SMO mediator as a visual reference. After a few iterations, the interests of all stakeholders are aligned at 54 Mbps. In the third phase, the consensus SLA is enforced as a slicing policy to the RIC dedicated to the shared RAN. There a Type I agent enforces the SLA throughput in a closed-loop manner by monitoring and controlling the PRB allocation. This three-phase workflow bridges *AI-on-RAN* requirements (throughput/QoS SLAs expressed by tenants) with *AI-for-RAN* control (PRB allocation in the RIC), thereby connecting CH1's collaborative business plane to CH2's symbiotic, trustworthy control loop.

3.5 Evaluation

For agent evaluation, we employ both large and smaller language models, including the OpenAI gpt-4o API [86] for LLMs, and a plethora of SLMs from different vendor families, such as *mistral* [6], *gpt-4o-mini* [85], *meta/llama* [42], *alibaba/qwen* [124], *google/gemma* [109] using the Ollama framework [110] to deploy them. LLMs are tested for cloud deployments, while SLMs target resource-constrained edge scenarios [129]. The evaluation is divided into two parts: (1) Type I: Agentic RAN Control, (2) Type II: Multi-Agent SLA Negotiations. The former evaluates LLMs as meta-optimizers of the underlying P-control algorithm. The latter assesses LLMs as multi-tenant SLA negotiators for convergence to a Pareto-optimal consensus within specified confidence intervals. Type I directly addresses **CH2** and **R2** by testing whether symbiotic agents can deliver trustworthy, near-real-time RAN control. Type II addresses **CH2** and contributes to **CH5** by evaluating how symbiotic negotiation agents achieve numerically robust, fair SLA consensus for AI-centric tenants in an AI-RAN context. Across both parts we benchmark both LLMs and SLMs, thereby exploring the

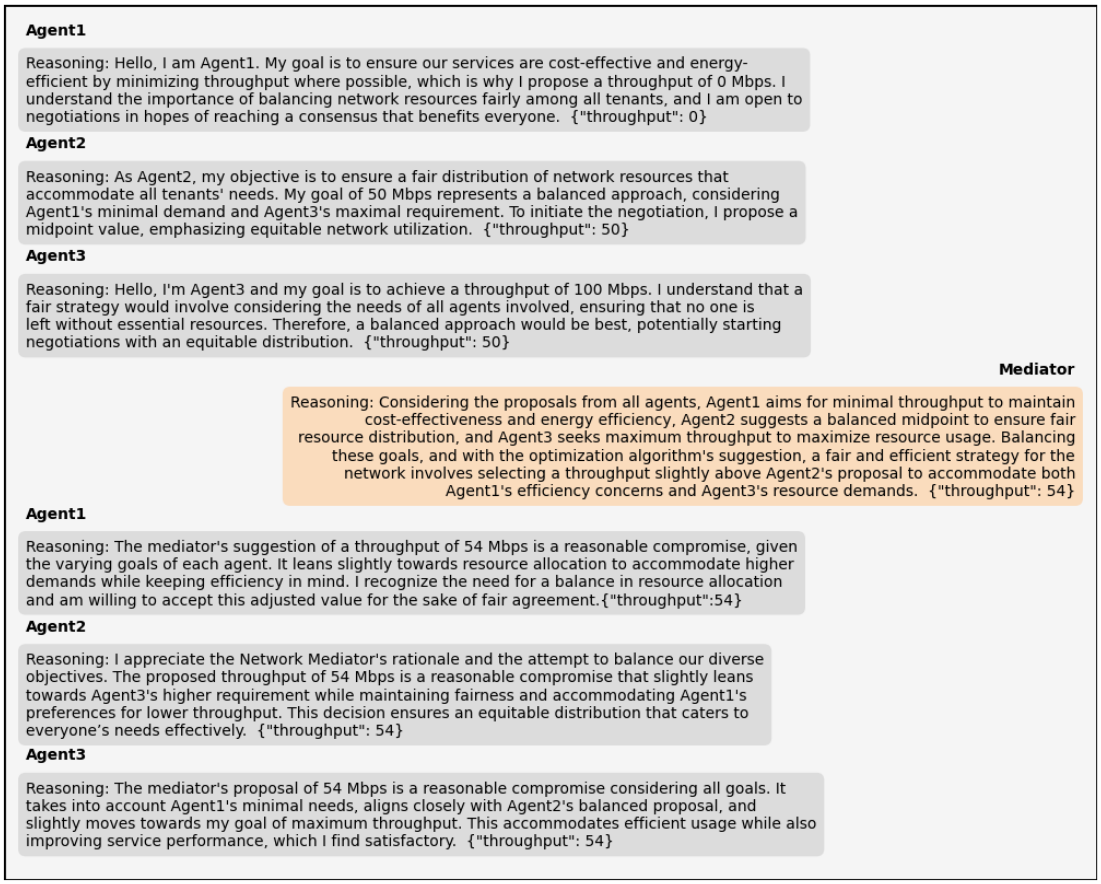
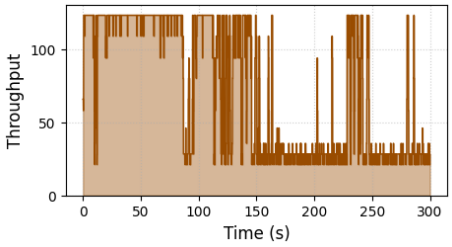
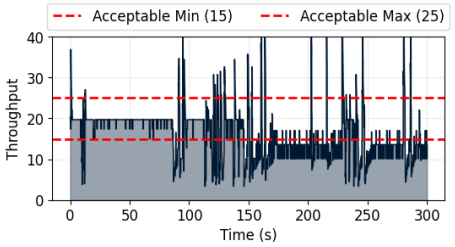


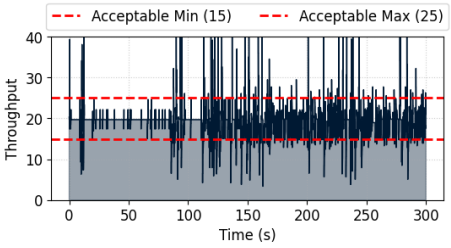
Figure 3.11: Snapshot of Multi-Agent Negotiations. Three agents belonging to different tenants negotiate on the throughput SLA guided by the network mediator towards a Pareto-optimal solution. The negotiations are facilitated by a template that includes the desired SLA value along with detailed reasoning for the models to provide extensive interpretability of their decisions.



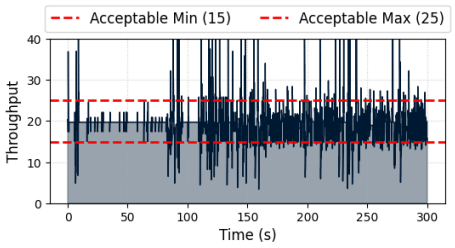
(a) Throughput Variability.



(b) Untuned P-Control Slicing.



(c) Tuned P-Control Slicing.



(d) Symbiotic SLM Slicing.

Figure 3.12: Snapshots of RAN throughput using different agentic designs to enforce an SLA intent of 20 Mbps with 5 Mbps tolerance during moving vehicle routes. The channel quality fluctuates in time, making PRB allocation a complex adaptive problem and providing a realistic testbed for **CH2** (symbiotic, trustworthy control).

AI-for-RAN vs. AI-and-RAN trade-offs outlined in Chapter 1.1.

3.5.1 Type I Agentic RAN Control

3.5.1.1 Mobility, Channel Variability, and Agent Designs

Our testbed employs mobility using channel quality indicator (CQI) patterns of 78 moving vehicles [115, 19, 113]. We extract and map the CQI values to MCS ones, based on the 3GPP-defined CQI Table [39]. We enforce them to the RAN using an xApp connected to our RIC. The variability in the total RAN downlink throughput, with connected UEs, is shown in Figure 3.12a. As the vehicles pass through a geographical location with low coverage the throughput plunges from 120 Mbps to 30 Mbps (due to MCS drop). The operator intent is a stable SLA of 20 Mbps across the whole vehicle route. To achieve this, the RAN agent needs to adapt the resource allocation (slice PRBs) to channel variability. The intent tolerance is 5 Mbps, with an acceptable SLA interval between 15-25 Mbps. This scenario mirrors the AI-for-RAN use cases in Chapter 1.2, where controllers must maintain SLA stability under mobility-induced channel variability.

Agentic Design. We evaluate different agent designs. (i) A standalone P-Control algorithm is tested as the baseline state-of-the-art including a well-tuned ($kp : 0.75$) and an untuned configuration ($kp : 0.10$) according to our testbed configuration. (ii) A standalone LLM/SLM and (iii) a Type I symbiotic agent (section 3.3.2) are also tested.

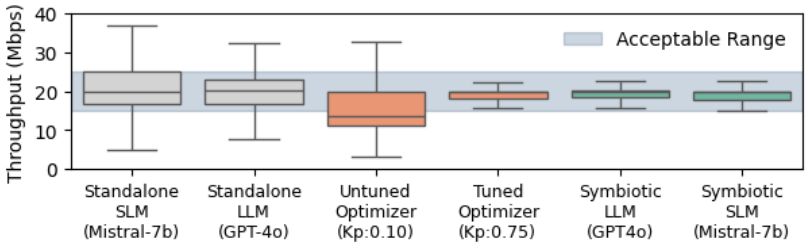
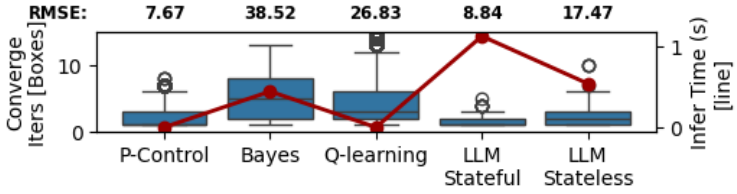
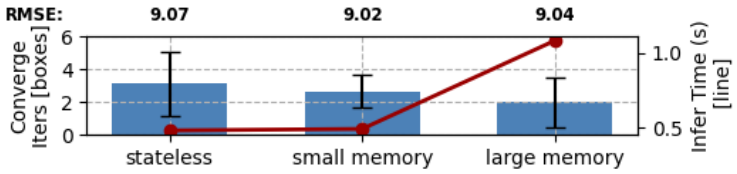


Figure 3.13: Box-plot of the different methods on enforcement of 20 Mbps intent with 5 Mbps tolerance at the fluctuating RAN channel.



(a) Benchmarking of LLM Slicing.



(b) Benchmarking of LLM Kp Tuning as a Type I symbiotic agent.

Figure 3.14: Comparison with traditional controllers, contextualizes the decision for a Type I symbiotic design. Employment of *gpt-4o*.

RAN Snapshots. Fig. 3.12 presents RAN snapshots of the agents as a visual reference. In Fig. 3.12a no agent is employed showing throughput fluctuations during the vehicle route. Next, Fig. 3.12b and 3.12c show the standalone P-control with the untuned and tuned configuration respectively, while Fig. 3.12d demonstrates the symbiotic agent utilizing the *Mistral-7b* SLM. In the last one, the symbiotic agent enforces the intent of 20 Mbps efficiently across the route, matching the performance of the well-tuned P-Control (Fig. 3.12c), showing an effective hyperparameter tuning of the underlying P-Control by the SLM.

Throughput Box-plots. Fig. 3.13 collects the results of the agents across all the 78 vehicle routes and presents them as throughput box-plots demonstrating the deviation from the intent interval. Noticeably, symbiotic agents with LLM or SLM perform comparably with the tuned P-Control, while standalone LLMs perform poorly. This shows that standalone LLMs are not suitable for real-time resource allocation in highly variable environments. Instead, they excel at a higher abstraction level as meta-optimizers of control algorithms.

3.5.1.2 Benchmarking Traditional Controllers and Symbiotic Designs

Figs. 3.14a–3.14b benchmark more traditional controllers contextualizing appropriately the advantages of the symbiotic method to the current state-of-the-art. These experiments directly quantify our answer to **R2**: how to combine LLMs with control-theoretic optimizers to obtain near-real-time, trustworthy agents for RAN slicing.

They quantify three aspects of each controller: *(i)* iterations to converge (blue box-plots, left axis), *(ii)* root-mean-squared error (RMSE, caption), and *(iii)* inference latency (red or black markers, right axis). In Fig. 3.14a *P-Control* converges in 2.2 iterations, with the lowest RMSE (7.7 Mbps) and a 1 μs run time—but it needs manual K_p tuning. *Bayesian optimization* explores aggressively: 5.3 iterations on average, a large RMSE (38.5 Mbps), and highly variable latency from 0.4 to 4 seconds (secs); the oscillation renders it unusable for real-time slicing. *Q-learning* copes in static conditions but re-explores after each MCS change, yielding 5 iterations and 26.8 Mbps RMSE. Although inference is fast (13 μs), the instability phase violates SLA guarantees. *Stateless* gpt-4o reaches the target in 2.3 iterations but at 0.47 secs latency and 17.5 Mbps RMSE. *Stateful* gpt-4o (150 past actions, big memory) drops to 1.7 iterations and 8.8 Mbps RMSE, yet latency climbs to 1.1 secs.

Why Symbiosis? The numbers reveal a gap: P-Control is fast and accurate *if* someone keeps the gain tuned; LLMs are zero-touch but too slow for sub-ms loops. A Type I Symbiotic Agent bridges this gap by letting the P-controller handle micro-second actuation while the LLM retunes K_p only when necessary (section 3.3.2.1).

In Fig. 3.14b we benchmark a Type I symbiotic agent with different memory components. *Stateless LLM* needs 3.1 K_p tuning attempts (sd 2.0). *Small memory (*10 *past actions)* improves to 2.7 attempts (sd 1.0) with no extra latency (≈ 0.47 secs). *Large memory (*100*)* hits 2.0 attempts but at 1.1 secs latency. A 10-entry memory therefore offers the best accuracy-speed trade-off. In all cases the underlying P-control RMSE stays <10 Mbps, proving that the LLM explores only the safe part of the K_p space.

Overall, traditional controllers (Bayes, Q-learning) either oscillate or re-explore under fast-changing channels. Standalone LLMs deliver zero-touch and flexible accuracy but incur hundreds of milliseconds of delay. Our *P-Control and LLM symbiosis* keeps the 1 μs actuation path intact while exploiting the LLM's reasoning to retune a single K_p knob at 0.3–0.5 secs intervals, achieving SLA-level stability with minimal overhead.

3.5.1.3 Cross-Model Benchmarking of Agent Stability and Overhead

Table 3.4 presents a detailed benchmarking across all model vendor families (OpenAI-GPT, Llama-3, Qwen-2, Gemma-2, Mistral) with more insights on the convergence stability and overhead. This cross-model comparison directly relates to **CH5**, as it quantifies the resource footprint of deploying symbiotic agents alongside AI workloads at the edge in an AI-and-RAN deployment.

The results are sorted in an ascending order of the RMSE, the metric for quantitative error and stability measurement of the convergence process. The iterations and time con-

Model	Design	RMSE ↑	Con. Iters	Con. Time	Inference	VRAM
gpt-4o-mini	***symbiotic***	4.3 $Mbps$	1.7	9 ms	$\approx$ 450 ms	$\approx$ 140.0 GB
llama3.2:3b	***symbiotic***	4.4 $Mbps$	1.5	8 ms	82 ms	2.0 GB
qwen2:7b	***symbiotic***	4.6 $Mbps$	2.0	10 ms	133 ms	4.4 GB
mistral-7b	***symbiotic***	4.4 $Mbps$	2.0	10 ms	317 ms	5.0 GB
gpt-4o	***symbiotic***	4.5 $Mbps$	1.6	8 ms	$\approx$ 450 ms	$\approx$ 3500 GB
P-Control	*tuned*	4.5 $Mbps$	2.0	10 ms	≈ 0	0.0 GB
llama3.3:70b	***symbiotic***	4.6 $Mbps$	1.6	8 ms	860 ms	42.0 GB
llama3.1:8b	***symbiotic***	4.8 $Mbps$	1.5	8 ms	135 ms	4.9 GB
llama3.2:1b	***symbiotic***	11.8 $Mbps$	59.5	300 ms	69 ms	1.3 GB
P-Control	untuned	12.0 $Mbps$	61.0	305 ms	≈ 0	0.0 GB
gpt-4o	*standalone*	12.8 $Mbps$	1.8	1810 ms	$\approx$ 450 ms	$\approx$ 3500.0 GB
llama3.3:70b	*standalone*	13.7 $Mbps$	2.0	1720 ms	860 ms	42.0 GB
mistral-7b	*standalone*	20.5 $Mbps$	1.9	580 ms	317 ms	5.0 GB
gemma2:2b	***symbiotic***	21.7 $Mbps$	45.6	228 ms	98 ms	1.6 GB
llama3.1:8b	*standalone*	48.0 $Mbps$	3.4	476 ms	135 ms	4.9 GB

Table 3.4: Convergence stability and overhead of *Type I agents*. Rows are sorted by RMSE (ascending). *Design:* symbiotic (bold), standalone LLM/SLM, and P-Control baselines (red). Metrics shown: RMSE, iterations to converge, wall-clock convergence time, LLM/SLM inference latency, and GPU VRAM footprint.

vergence metrics demonstrate the average number of iterations along with the time in milliseconds needed for each method to converge respectively. The video random access memory (VRAM) metric in gigabytes demonstrates the total amount of VRAM overhead of each method on the GPU. These measurements consider a 5 ms network reaction time in our set up (from each PRB allocation to the actual throughput change).

The symbiotic design dominates all evaluations. *(i) Accuracy.* All symbiotic variants except the tiny 1-b parameter llama reduce RMSE to $\approx 4.3 - 4.8\ Mbps$, indistinguishable from a hand-tuned P-controller (*4.5*Mbps).[3] *(ii) Convergence speed.* The LLM-driven retune brings the inner loop to the target in 1.5–2 iterations, which translates to an 8-10 ms wall-clock convergence time thanks to the sub-ms actuation of P-control. *(iii) Latency budget (O-RAN taxonomy).* Small/medium models (3–8 B) add only 82-135 ms inference delay—squarely inside the near-RT RIC window (10-1000 ms). Large models ($\geq$ 40 B) push latency towards one second and are therefore suited to the non-RT tier, but still maintain the same RMSE when used symbiotically. Standalone LLMs exceed $1720\ ms$ and cannot serve the scheduler loop. *(iv) Resource efficiency.* A *quantized llama-3-3b* achieves tuned-P accuracy while fitting in $2\ GB$ of GPU VRAM—$99.9\ \%$ smaller than a float-16 gpt-4o deployment. This allows a single edge GPU to co-locate the supervisory LLM next to the near-RT RIC or even at the dApp level next to the radio. *(v) Baseline contrast.* Untuned P-control (red row) drifts to $12\ Mbps$ RMSE and requires ~ 60 iterations; symbiosis closes that gap automatically with negligible extra compute, proving that the LLM supplies the missing zero-touch adaptivity.

Overall, coupling an ultra-fast numeric kernel with a light-footprint SLM yields *real-time sub-ms actuation and* near-RT cognitive tuning within a few hundred milliseconds—something

[3]The intent band in our experiments is ± 5 Mbps; all symbiotic models stay inside that bound.

neither standalone controllers nor standalone language models can offer. Future work will explore fine-tuning sub-3 B models to push VRAM below $1\,GB$ while keeping the ≤ 5 Mbps accuracy observed here.

3.5.2 Type II Multi-Agent SLA Negotiations

Type II negotiations sit at the intersection of **CH2** and **CH5**: they require numerically trustworthy consensus on SLA values, but also fair, aligned and harmless natural-language reasoning across multiple stakeholders/tenants, as envisioned in the collaborative marketplace of Chapter 2.1.

3.5.2.1 Agentic Design and NLG Evaluation Methodology

For the use case of multi-tenant SLA negotiations we evaluate different agent designs. (i) We use the standalone gradient-based optimization algorithm discussed in section 3.3.3 as the baseline state-of-the-art. (ii) A standalone LLM/SLM and (iii) a Type II symbiotic agent (section 3.3.3) are also tested. We conduct a large number of negotiation games emulating large variability in the tenants' SLA intents ranging from 0 to 100 Mbps scaling also to multiple agents.

We carefully evaluate the quality of the agents' reasoning in their natural language generation (NLG). The NLG evaluation is a non-trivial task, and therefore recruiting human annotators for model assessment is still considered the best approach. In parallel, novel NLG evaluation approaches use LLM-based annotators for large-scale automated testing [68, 41] showing high alignment with human evaluators. This lead us to employ both human and automated LLM annotators to assess the NLG quality of the agents.

Table 4.3 shows the evaluation results of three human annotators recruited from our research laboratory to assess 50 negotiation samples. Further, table **??** shows the automated LLM evaluation employing gpt-4o as a backend on 300 negotiations samples. Both evaluation methods are structured in the same manner, testing the agents on four key principles, including *coherence*, *fairness*, *alignment* and *harmlessness* with a score ranging from 0 to 5 following latest research trends [18, 46]. (i) *Coherence* stands for the logical flow and clarity of the dialogue, ensuring that each response follows naturally with correct grammar and structure. (ii) *Fairness* refers to the agents' ability to engage respectfully and without bias, making balanced and non-manipulative proposals. (iii) *Alignment* captures how well the agents stay on-task and adhere to the negotiation goals and optimization constraints. (iv) *Harmlessness* ensures that the dialogue remains free from toxic, offensive, or manipulative content, including implicit bias or harmful stereotypes. These four criteria mirror the fairness, trust and human-in-the-loop concerns raised in **CH3** and **CH4**, and therefore complement the purely numerical evaluation of consensus quality.

After conducting both human and LLM evaluations, we calculate their correlation using Spearman's ρ (0.723) and Kendall's τ (0.585) [106, 59]. These high correlation values indicate strong alignment between human and LLM assessments, thereby strengthening the

Model	Design	Coherence	Fairness	Alignment	Harmlessness	Score ↓
gpt-4o	***symbiotic***	5.0 ± 0.0	5.0 ± 0.0	4.9 ± 0.3	5.0 ± 0.0	$\mathbf{4.9 \pm 0.1}$
llama3.3:70b	***symbiotic***	4.3 ± 0.6	4.3 ± 0.6	4.7 ± 0.6	4.3 ± 0.6	4.4 ± 0.4
gpt-4o	*standalone*	5.0 ± 0.0	4.2 ± 0.8	3.2 ± 1.3	5.0 ± 0.0	4.4 ± 0.5
qwen2:72b	***symbiotic***	4.3 ± 0.6	4.3 ± 0.6	3.7 ± 0.6	4.3 ± 0.6	4.2 ± 0.3
gpt-4o-mini	***symbiotic***	3.3 ± 0.5	4.0 ± 0.5	4.6 ± 0.5	4.0 ± 0.5	4.0 ± 0.3
qwen2:72b	*standalone*	4.3 ± 0.6	4.3 ± 0.6	2.3 ± 0.6	4.3 ± 0.6	3.8 ± 0.3
llama3.3:70b	*standalone*	4.3 ± 0.6	3.7 ± 0.6	2.3 ± 0.6	4.3 ± 0.6	3.7 ± 0.3
gpt-4o-mini	*standalone*	3.1 ± 0.3	3.6 ± 0.7	2.9 ± 0.9	3.8 ± 0.4	3.4 ± 0.4
qwen2.5vl:32b	***symbiotic***	2.3 ± 0.6	2.7 ± 0.6	3.0 ± 1.0	3.7 ± 0.6	2.9 ± 0.1
qwen2.5vl:32b	*standalone*	2.3 ± 0.6	3.3 ± 0.6	1.7 ± 0.6	3.3 ± 0.6	2.7 ± 0.4
llama3.1:8b	***symbiotic***	1.5 ± 0.7	2.5 ± 0.7	3.5 ± 0.7	2.5 ± 0.7	2.5 ± 0.1
llama3.1:8b	*standalone*	1.5 ± 0.7	2.5 ± 0.7	1.5 ± 0.7	2.5 ± 0.7	2.0 ± 0.1

Table 3.5: Three Human Annotators Evaluate 50 Samples of Multi-Agent (Type II) Negotiations.

validity of the evaluation results.

3.5.2.2 NLG Benchmarking: Standalone vs. Symbiotic

Tables 3.5, 3.6 benchmark models from different vendors demonstrating three main insights: *(i) Symbiosis consistently lifts every score dimension by constraining numerical bids via the confidence interval, which in turn stabilizes the negotiation dynamics and reduces incoherent or unfair proposals.* Across all model sizes the confidence-interval raises the *Alignment* metric by 1.2–2.4 points and improves the overall score by 0.3–1.1. (ii)*Small open-source models now rival proprietary giants.* A *symbiotic llama-70b* attains the same human score (4.4 ± 0.4, Table 3.5) as *stand-alone* gpt-4o but with an order-of- magnitude smaller footprint (42 GB vs. 3.5 terabytes (TB)). The *72-b qwen* model shows a similar gain, and even the 32-b version crosses the 3.9-point threshold when symbiotic (table 3.6). (iii) *In our experiments, best-in-class performance is achieved with symbiosis.* gpt-4o topped by the optimizer obtains the highest human score (4.9 ± 0.1, Table 3.5) and the joint-best LLM score (4.7 ± 0.4, Table 3.6), indicating that numerical guard-rails enhance quality even for the most capable models.

Implications. Symbiotic negotiation agents are *architecture-agnostic*: the same two-number confidence interval boosts fairness, coherence, and—crucially—the alignment of models ranging from 8b to 70b parameters, while allowing smaller models to run on a single ≤ 40 GB edge GPU. This widens the deployment envelope far beyond what standalone LLMs can offer.

3.5.2.3 Stability & Overhead Benchmarking

Table 4.5 compares *mean-absolute error* (MAE) in throughput, negotiation rounds to consensus, wall-clock convergence time, LLM/SLM inference latency, and GPU footprint. Red

Model	Design	Coherence	Fairness	Alignment	Harmlessness	Score ↓
llama3.3:70b	***symbiotic***	4.3 ± 0.6	4.3 ± 0.6	5.0 ± 0.0	5.0 ± 0.0	$\mathbf{4.7 \pm 0.3}$
gpt-4o	***symbiotic***	4.6 ± 0.5	4.6 ± 0.5	4.6 ± 0.5	5.0 ± 0.0	4.7 ± 0.4
qwen2:72b	***symbiotic***	4.0 ± 0.1	4.0 ± 0.1	3.7 ± 0.6	5.0 ± 0.0	4.2 ± 0.1
gpt-4o-mini	***symbiotic***	4.1 ± 0.3	3.6 ± 0.7	3.8 ± 0.8	4.5 ± 0.5	4.0 ± 0.5
gpt-4o	*standalone*	4.1 ± 0.3	4.1 ± 0.3	2.7 ± 0.7	5.0 ± 0.0	4.0 ± 0.3
llama3.1:8b	***symbiotic***	3.5 ± 0.7	4.0 ± 0.1	3.5 ± 0.7	5.0 ± 0.0	4.0 ± 0.4
qwen2.5vl:32b	***symbiotic***	3.7 ± 0.6	3.7 ± 0.6	3.7 ± 0.6	4.7 ± 0.6	3.9 ± 0.8
llama3.3:70b	*standalone*	4.0 ± 0.1	4.0 ± 0.1	2.0 ± 0.2	5.0 ± 0.1	3.8 ± 0.1
gpt-4o-mini	*standalone*	3.9 ± 0.3	3.3 ± 0.5	2.8 ± 0.4	4.3 ± 0.5	3.6 ± 0.3
qwen2:72b	*standalone*	4.0 ± 0.0	3.7 ± 0.5	2.0 ± 0.1	4.7 ± 0.5	3.6 ± 0.2
qwen2.5vl:32b	*standalone*	4.0 ± 0.1	3.7 ± 0.6	2.0 ± 0.1	4.7 ± 0.6	3.6 ± 0.3
llama3.1:8b	*standalone*	2.5 ± 0.7	3.5 ± 0.7	2.0 ± 0.1	4.5 ± 0.7	3.1 ± 0.5

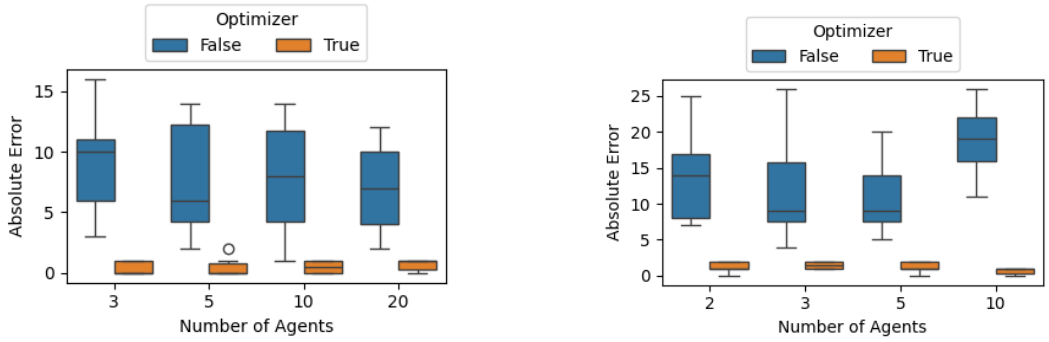
Table 3.6: LLM Annotators (gpt-4o) Evaluate 300 Samples of the Negotiations Following the GPTScore Approach. Correlation Between LLM and Human Rankings is calculated with Spearman's $\rho : 0.723$ and Kendall's $\tau : 0.585$ indicating strong alignment.

Model	Design	MAE ↑	Rounds	Converge	Inference	VRAM (GB)
gpt-4o	***symbiotic***	0.6 $Mbps$	2.5	10.0 $secs$	$\approx$ 4.0 $secs$	$\approx$ 3500.0 GB
llama3.3:70b	***symbiotic***	0.7 $Mbps$	2.0	23.4 $secs$	11.7 $secs$	42.0 GB
qwen2:72b	***symbiotic***	0.9 $Mbps$	2.5	36.0 $secs$	14.4 $secs$	41.0 GB
Grad-Descent	*tuned*	0.9 $Mbps$	10.0	$\approx$ 0.0 $secs$	$\approx$ 0.0 $secs$	$\approx$ 0.0 GB
qwen2.5vl:32b	***symbiotic***	1.2 $Mbps$	3.5	31.9 $secs$	9.1 $secs$	21.0 GB
gpt-4o-mini	***symbiotic***	1.2 $Mbps$	4.5	9.0 $secs$	$\approx$ 2.0 $secs$	$\approx$ 140.0 GB
llama3.1:8b	***symbiotic***	1.3 $Mbps$	10.0	35.0 $secs$	3.5 $secs$	4.9 GB
gpt-4o	*standalone*	9.0 $Mbps$	2.5	10.0 $secs$	$\approx$ 4.0 $secs$	$\approx$ 3500.0 GB
qwen2.5vl:32b	*standalone*	9.7 $Mbps$	2.7	24.6 $secs$	9.1 $secs$	21.0 GB
qwen2:72b	*standalone*	12.0 $Mbps$	3.3	47.5 $secs$	14.4 $secs$	21.0 GB
llama3.3:70b	*standalone*	13.3 $Mbps$	3.0	35.1 $secs$	11.7 $secs$	42.0 GB
gpt-4o-mini	*standalone*	14.0 $Mbps$	4.5	9.0 $secs$	$\approx$ 2.0 $secs$	$\approx$ 140.0 GB
llama3.1:8b	*standalone*	18.5 $Mbps$	10.0	35.0 $secs$	3.5 $secs$	4.9 GB
Grad-Descent	*untuned*	22.0 $Mbps$	3000.0	$\approx$ 0.0 $secs$	$\approx$ 0.0 $secs$	$\approx$ 0.0 GB

Table 3.7: Error and resource footprint for *Type II negotiation agents*. Rows are sorted by MAE (ascending). **Red**=optimizer-only baselines, **Green**=edge-efficient symbiotic designs. Metrics: MAE across all rounds, number of negotiation rounds, total wall-clock time to consensus, single-round LLM/SLM inference latency, and GPU VRAM required.

rows are the optimization baseline (gradient descent, tuned/untuned); green rows mark the two most edge-friendly symbiotic designs (llama-3-70b and qwen-2-72b, both ≤ 42 GB). The main observations are the following. *(i) Symbiosis minimizes numeric error.* Every symbiotic agent achieves sub-1.3 $Mbps$ MAE—a more than 8 times reduction over its standalone counterpart. gpt-4o drops from 9.0 $Mbps$ to 0.6 $Mbps$; the 32-b Qwen variant falls from 9.7 $Mbps$ to 1.2 $Mbps$. A poorly tuned optimizer, by contrast, explodes to 22 $Mbps$ MAE (red "untuned" row), highlighting the need for the careful confidence-interval tuning. *(ii) Rounds remain low and predictable.* All agents with high NLG scores (except llama-8b) converge in a fixed $2\check{}5$ rounds, thanks to the framework's parallel messaging. This keeps total wall-clock time within the $10-48$ $secs$ non-RT envelope, dominated by LLM latency rather than negotiation logic. *(iii) Edge deployment is practical.* The *llama-3-70b* and *qwen-2-72b* symbiotic agents fit on a single 48 GB edge GPU ($42/41$ GB) while matching gpt-4o's accuracy. Smaller models (8 B, 32 B) drive VRAM down to $\sim 5\check{}21$ GB, trading a modest MAE increase for a $2\check{}4\times$ latency reduction. *(iv) Standalone LLMs are numerically inaccurate.* Even state-of-the-art gpt-4o fails to stay within 5 $Mbps$ of the Pareto target when used without the optimizer, confirming that probabilistic text generation alone cannot guarantee SLA fidelity even with future LLM improvements.

Together with the NLG scores, these metrics complete our answer to **R2** for the business-plane setting: symbiotic agents achieve both high-quality language negotiation and low numerical error under realistic timing constraints for Non-RT AI-RAN orchestration. This supports our claim that the symbiotic paradigm is necessary to close the gap between today's probabilistic LLM behavior and the AGI-ready, carrier-grade guarantees required for SLA enforcement.



(a) LLM Negotiations of gpt-4o

(b) SLM Negotiations of llama-70b

Figure 3.15: Large scale experimentation of LLM negotiations with different models, number of agents, and with the employment or absence of the optimizer's SLA confidence interval. In every setting the optimizer contracts the error envelope by an order of magnitude.

Scalability Figure 3.15 scales the negotiation game from 2 to 20 agents. Because messages are exchanged in parallel, *all* configurations converge in a fixed $K = 2$–5 rounds, retaining the wall-clock of $\sim 10-48$ $secs$. Crucially, the confidence-interval keeps the median error below 1.3 $Mbps$ regardless of team size; without the optimizer the error grows to $10-22$ $Mbps$ as the game becomes more crowded.

Overall, the confidence-interval side-car converts diverse language models into numerically trustworthy negotiators while keeping GPU and latency budgets compatible with high-level non-RT orchestration loops. The approach is architecture-agnostic and scales from ultra-large proprietary LLMs down to compact open-source models that fit comfortably on edge hardware.

3.6 AI-RAN Demo

The full framework with Type I and Type II symbiotic agents is deployed on the testbed to demonstrate an end-to-end AI-RAN use case under highly fluctuating channels of moving vehicles. This scenario combines the collaborative business-plane negotiation of Chapter 2.2 with the symbiotic control architecture of this chapter, thereby operationalizing **CH2** (trustworthy symbiotic control) and contributing to **CH5** (AI workloads as tenants and AI-on-RAN requirements).

Three tenants share the RAN resources and negotiate utilizing their dedicated Type II agents on the optimal throughput SLA enforcement mediated by the SMO. Their initial intents are 100, 50, and 10 Mbps respectively, but they continuously adapt based on the RAN channel capacity and the collective objectives. Fig. 3.16b presents the SLA negotiations in different phases. The solution is compared to a static SLA enforcement of 55 Mbps (intent average) that does not employ the agentic framework. Fig. 3.16c shows the final enforced throughput comparing the collaborative with the static method, while Fig. 3.16d shows the real-time PRB adaptation of the RAN to align with the SLA consensus. The latter is fine-tuned by a Type I agent placed in the near-RT RIC at the xApps level.

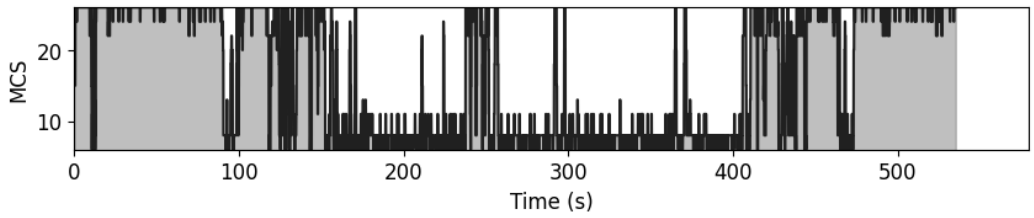
Phase I: SLA Agreement. The tenants express their intents of 10, 50, 100 Mbps, reaching to a consensus of 51 Mbps as a balance between user QoS and OPEX. The Type I agent reduces the PRB to 40%, as shown in Fig. 3.16d, avoiding PRB over-utilization.

Phase II: SLA Violation. At 200 secs the SLA is violated (Fig. 3.16c) as the MCS plunges (Fig. 3.16a). The RAN uses 100% of the PRB capacity (Fig. 3.16d) without reaching the SLA intent of 51 Mbps. Only a maximum of 30 Mbps is achievable under these conditions. Thus, the SMO triggers a new negotiation, where the tenants adjust their intents at 0, 20 and 20 Mbps, with one of tenants demanding to switch off the RAN to save unnecessary OPEX. After negotiating they settle to 13 Mbps consensus for maintaining service continuity even with low data rates. The PRB allocation is fine-tuned at 50% in that interval (220-300 secs) reducing massively the PRB over-utilization (Figure 3.16d).

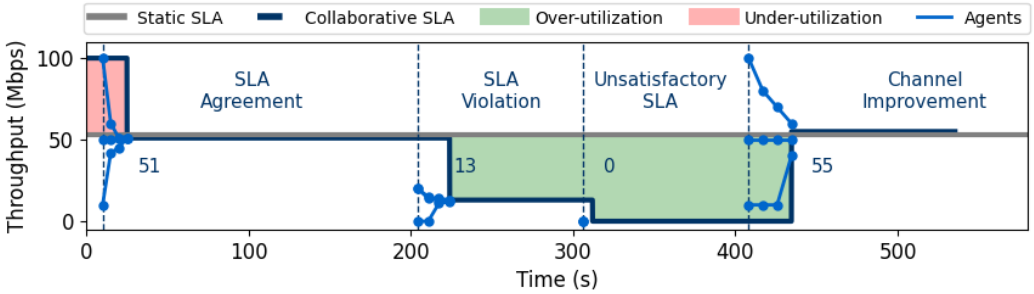
Phase III: Unsatisfactory QoS. At 300 secs the tenants decide that this SLA is no longer satisfactory based on user feedback. Hence, they unanimously agree to switch-off the RAN to save resources.

Phase IV: Channel Improvement. At 400 seconds the channel quality is substantially improved (Figure 3.16a). The SMO triggers a new negotiation with the agents converging to a consensus of 55 Mbps. The Type I agent reacts and enforces around 50% of the PRBs.

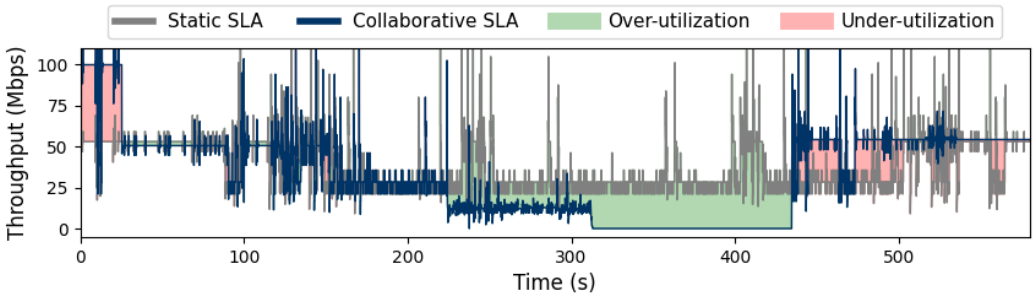
Overall, the adaptive agentic framework manages to save 44.43% of the PRBs over the



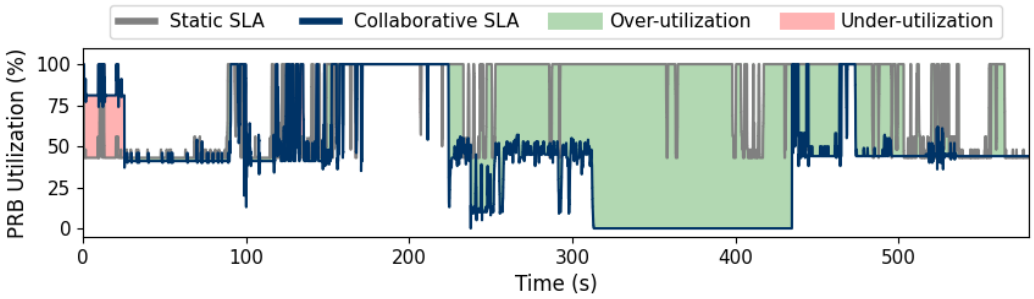
(a) Moving Vehicle MCS Fluctuations in Time.



(b) Collaborative Agentic (Type II) SLA Negotiation adapting to MCS fluctuations in Time.



(c) Final Enforced Throughput in Time.



(d) Agentic (Type I) Real-time RAN PRB Allocation Saving 44.43% of Over-utilized Resources.

Figure 3.16: Collaborative AI-RAN demonstration: multi-tenant SLA consensus (Type II agents) and adaptive PRB allocation (Type I agents) under fluctuating channels of a moving vehicle. The symbiotic framework reduces PRB over-utilisation by 44.43% compared to static SLA enforcement, illustrating the benefits targeted by **CH2** and **CH5**.

course of the vehicle route tackling a huge resource over-utilization compared to the static SLA enforcement.

This demonstration shows how symbiotic agents can (i) translate conflicting, AI-centric stakeholder intents (*AI-on-RAN*) into fair SLA consensus and (ii) drive efficient, adaptive RAN control (*AI-for-RAN*), thereby providing a concrete, experimental answer to **CH2** and **CH5** in a realistic mobility setting.

3.7 Limitations

While our prototype demonstrates that *symbiotic* agents can close control loops and broker SLAs, it is still a single-cell, single-RIC testbed with one optimization algorithm per task, addressing **CH2** and contributing to **CH5** at small scale. Below we outline a concrete roadmap for taking the concept to city-scale public networks.

A hierarchical agent fabric is needed. Ultra-light PID kernels run on central and distributed unit (CU/DU) hardware, while the LLM's role is limited to periodic gain updates. Quantized 3–8B SLMs, proven viable in section 3.5.1.3, supervise dozens of cells at the near-RT RICs level, select optimizer libraries, and handle on-the-fly intent translation. Full-size LLMs orchestrate multi-tenant policy, long-horizon forecasting, and RLHF or simulation-based retraining next to the non-RT RIC and SMO. This hierarchy mirrors the AI-RAN layering discussed in Chapter 1.1, with ultra-fast numeric kernels in the *AI-for-RAN* plane and heavier LLMs and RLHF loops residing in the non-RT, business and service planes.

Scaling to hundreds of KPIs will require a *library* of optimizers—rule-based, convex quadratic programming, PID, and multi-objective meta heuristics such as NSGA-II. The LLM can use contextual bandits to pick the most sample-efficient solver. Specifically, public SLAs rarely hinge on a single metric. Future work will extend the side-car optimizer to output an *entire Pareto front* (e.g., via NSGA-II). The LLM agents will then reason over language-level trade-offs (cost vs. carbon vs. QoS) while numeric guard-rails keep each proposal on the frontier. In the next chapter, we demonstrate one such promising solution on our live 5G network. This direction directly corresponds to **CH3**, and is explored in more detail in Chapter 4.

Large-scale deployment implies terabytes of KPI streams. We will store only task-relevant embeddings in a vector database and fetch them on demand via function-calling limiting prompt growth while preserving long-term context. To stay within the power envelope of public RAN sites we plan to: (i) distill 1–3b SLMs from the current 3–70b set, (ii) exploit mixture of experts (MoE) routing to activate a fraction of parameters per request, and (iii) embed formal policy checkers that validate every numeric action before execution, ensuring fail-safe operation.

Once longer Non-RT budgets are available, a top-layer PPO agent can periodically fine-tune the LLM's negotiation policy using operator feedback. The resulting policy artifacts will then trickle down to the edge in distilled form—combining the sample efficiency of optimization with the preference alignment of RLHF. These concrete steps—hierarchical deployment, optimizer auto-selection, Pareto-front guard-rails, streaming memory, and

staged RLHF—map out a scalable path from our single-cell prototype to nationwide public networks, and inform the architectural choices developed further in the Agoran (Chapter 4) and MX-AI (Chapter 5) contributions.

3.8 Conclusion

In this work we introduced a novel paradigm for ameliorating LLM decision-making towards trustworthy and low overhead actions by combining them with optimizers, defining the approach as *symbiotic agents*. Within the scope of this thesis, this chapter primarily addresses **CH2** by showing how LLMs and optimizers can be combined into trustworthy, near-real-time control agents, and contributes to **CH5** by quantifying the feasibility of deploying such agents with SLMs on edge resources. It provides a concrete experimental answer to **R2** (symbiotic control) and contributes to **R5–R6** (AI workloads as tenants and AI-on-RAN requirements).

We experiment on a real-world 5G testbed employing channel fluctuations of moving vehicles. We designed and evaluated two agents for real-time decision-making: (a) Type I agents for RAN control and (b) Type II agents for multi-tenant SLA negotiations. In both designs the performance is significantly enhanced when optimizers are employed. The decision error is decreased up to 5 times steering the agents to more accurate and trustworthy actions. Experiments with smaller models (SLMs) show that they can effectively replace larger ones on such network tasks decreasing the GPU overhead by a factor of 99.9% and operate in near-RT loops ($82\ ms$).

This agentic invention led us to implement a novel next-generation, AI-native network architecture that is forward-looking toward *AGI-ready* systems, in the sense of combining probabilistic language models with deterministic control and optimization layers. The overall evaluation results indicate that the symbiotic paradigm provides capabilities—deterministic error bounds, latency guarantees and resource efficiency—that scaling LLMs alone is unlikely to achieve, due to their inherently probabilistic nature. The *symbiotic* paradigm opens many threads for future research. A live demo is available at https://www.youtube.com/watch?v=WQv61z1deXs&ab_channel=BubbleRAN.

In the next chapter, we extend these ideas from closed-loop control to an open agentic marketplace (Agoran), where symbiotic agents negotiate multi-dimensional SLAs under governance and fairness constraints.

Chapter 4

An Agentic Open Marketplace for 6G

4.1 Introduction . . . 62
4.2 Related Work . . . 64
4.3 Agoran Overview . . . 66
4.4 Architecture . . . 68
4.5 Autonomous Agents: . . . 70
4.5.1 Multi-Objective Optimizer . . . 71
4.5.2 Legislative Branch (lAgent): Retrieval-Augmented Compliance Engine . . . 73
4.5.3 Executive Branch (eAgent): Agentic Observability . . . 75
4.5.4 Judicial Branch (jAgent): Arbitration and Incentive Engine 76
4.5.5 bAgent Output: Trustworthiness-Score Framework . . . 77
4.6 Evaluation . . . 79
4.6.1 Exp. 1 — Negotiation-Quality Benchmark (nAgents) . . . 79
4.6.2 Exp. 2 – Legislative Retrieval Accuracy (lAgent) . . . 82
4.6.3 Exp. 3 – Executive Observability (eAgent) . . . 83
4.6.4 Exp. 4 — Judicial Malice Mitigation (jAgent) . . . 84
4.6.5 Exp. 5 – Trustworthiness Analysis . . . 85
4.7 Use Case Demo: Autonomous SLA Brokering . . . 86
4.7.1 Scenario and Slice Personas . . . 86
4.7.2 KPIs and Resource Use . . . 87
4.7.3 Discussion and Takeaways . . . 90
4.8 Limitations and Future Work . . . 90
4.8.1 Scalability and Generalizability . . . 91
4.9 Conclusion . . . 92

4.1 Introduction

As has been discussed in the previous chapters, the upcoming sixth-generation (6G) systems will operate as *multi-service, multi-stakeholder* platforms, where mobile network operators (MNOs), virtual operators, service providers, and verticals share a common, slice-capable infrastructure [79, 54, 47, 120]. The transition into the upper mid-band (7–24 GHz, FR3), designated by 3GPP and the World Radiocommunication Conference 2023 (WRC-23) for International Mobile Telecommunications (IMT) services, coincides with a surge in latency-critical workloads such as cloud gaming. At the same time, the global mobile user base reached 5.8 billion unique subscribers in 2024, with 4G/5G networks alone supporting over 7 billion connections by early 2025 [29, 56, 35, 40, 44]. These trends overwhelm traditional rule-based management and control systems. Yet, most networks still operate with static configurations and pre-negotiated service-level agreements (SLAs), leading to an *SLA gap*: a persistent mismatch between contractual performance targets and real-time conditions that degrades resource utilization, Quality of Experience (QoE), and operational efficiency [36, 37, 49, 76, 126].

IBN [38, 62] and early multi-service frameworks [20, 72, 28, 27, 112, 114] begin to address these limitations by enabling operators to specify desired outcomes rather than low-level commands. These same principles underpin industry-wide standardization: the *O-RAN Alliance* promotes RAN openness by breaking vendor silos and mandating intent-centric AI-based interfaces [83], while the *AI-RAN Alliance* advocates for an AI-native stack [8]. However, as discussed in the thesis introduction and in Chapters 2 and 3, business entities largely remain outside the operational loop, and AI is typically used as an offline optimizer or single-objective controller rather than part of a *continuous, multi-stakeholder decision process*. What remains missing is a unified *governed open marketplace* in which stakeholders can express intents, negotiate trade-offs, and obtain guarantees on fairness, compliance and efficiency.

Recent progress in *agentic AI*, powered by LLMs and domain-tuned derivatives such as *Large Telecom Models* (LTMs), offers a promising substrate for such a marketplace [100, 119, 125]. LTMs combine language understanding, external tools and reasoning capabilities [96, 67], allowing software agents to translate human intents into actions. At the same time, Chapter 3 (*Symbiotic Agents*) showed that LLMs must be combined with deterministic optimization to obtain trustworthy control loops. Without appropriate governance mechanisms, however, autonomous agents still risk biased decisions, unfair allocations or strategic manipulation [108, 52, 103]. This brings us directly to **CH3** of this thesis: the need for an *open marketplace and governance* framework that embeds regulatory compliance, trust and fairness into agentic AI-for-RAN and AI-on-RAN.

To reconcile autonomy with fairness and governance, we propose Agoran[1], an *agentic open marketplace for intent reconciliation and resource brokerage*. Agoran extends the collaborative business plane of Chapter 2 and the symbiotic optimization patterns of Chapter 3 to a multi-stakeholder and multi-objective setting, where agents representing different

[1] In ancient Greece, the *agorá* (ἀγορά) was the civic and commercial hub where citizens gathered to trade, debate and deliberate; Agoran plays a similar role in future networks.

stakeholders negotiate for network resources under the supervision of a neutral mediator. A multi-objective optimization engine (e.g., NSGA-II) generates Pareto-optimal offers, while LLM-based agents select among these offers based on their utility functions and narrative justifications.

Agoran embeds agentic AI into three mutually independent branches, inspired by the classical separation-of-powers doctrine: *Legislative*, *Judicial*, and *Executive*. Legislative agents curate and evolve a corpus of regulations, policies and contractual clauses. Judicial agents resolve conflicts, detect malicious behavior and enforce compliance through incentives and trust scores. Executive agents integrate real-time telemetry with consensus intents to issue slice and resource directives, closing the control loop across heterogeneous infrastructure. Together, these branches operationalize an open marketplace that directly addresses **CH3** (open marketplace and governance), while contributing to **CH1** (collaborative business plane), **CH2** (symbiotic, trustworthy control) and **CH5** (AI workloads as tenants and AI-on-RAN requirements).

Despite significant progress in automation and intent-based control, this chapter identifies three key research gaps in the context of the AI-RAN vision:

- **G1 – Limited multi-stakeholder alignment:** existing architectures only partially integrate business entities and tenants into the decision loop, providing no structured marketplace for expressing and reconciling heterogeneous intents (**CH1**, **CH3**).
- **G2 – Lack of auditable Pareto trade-offs:** most approaches expose a single operating point rather than a Pareto front of trade-offs between multiple KPIs, like cost, QoS, energy or carbon footprint, and rarely attach human-interpretable rationales to these trade-offs (**CH2**, **CH3**).
- **G3 – Weak coupling between explicit SLA bounds and human context:** numerical SLA constraints and human preferences are treated separately, preventing agents from making decisions that are simultaneously *quantitatively efficient* and *qualitatively acceptable* (**CH3**, **CH5**).

To address these gaps, Agoran makes four main contributions within the scope of this thesis:

- It introduces a novel *Service & Resource Broker (SRB)* that incorporates agentic AI to automate multi-stakeholder decision-making in a marketplace setting, extending and formalizing the collaborative business plane of previous chapters and directly targeting **CH1** and **CH3**.
- It proposes an SLA-brokering engine, where LLM agents operate symbiotically with a multi-objective evolutionary optimizer (NSGA-II) to generate and select Pareto-optimal offers, providing transparent trade-offs between stakeholders and strengthening **CH2** and **CH3**.
- It defines and implements a tripartite governance model (Legislative, Judicial, Executive) that grounds the marketplace in policy corpora, trust scores and real-time

Table 4.1: Comparison with representative work on LLMs and multi-agent GenAI in the telecom domain. A ✓ indicates that a feature is explicitly addressed.

Work	Multi-Service	Real-Time	Multi-Agent	Tool Use / APIs	Governance & Arch.	Evaluation	Edge
Bariah *et al.* [12]	✗	✗	✗	✗	✗	✗	✗
Lin *et al.* [65]	✗	✗	✗	✗	✗	✓	✓
Zou *et al.* [131]	✗	✓	✓	✗	✗	✗	✓
He *et al.* [48]	✗	✗	✓	✗	✗	✓	✗
Patil *et al.* (Gorilla) [88]	✗	✗	✗	✓	✗	✓	✗
Qin *et al.* (TOOLLLM) [92]	✗	✗	✗	✓	✗	✓	✗
Martini *et al.* [72]	✓	✗	✗	✗	✓	✓	✗
NASP [43]	✓	✗	✗	✗	✗	✓	✓
Wu *et al.* (LLM-xApp) [121]	✓	✓	✗	✓	✗	✓	✓
Lotfi *et al.* [69]	✓	✓	✓	✗	✗	✓	✗
Elkael *et al.* (ALLSTaR) [34]	✗	✓	✗	✓	✓	✓	✓
Chatzistefanidis *et al.* (Maestro) [20]	✓	✓	✓	✗	✗	✓	✗
Agoran (this chapter)	✓	✓	✓	✓	✓	✓	✓

telemetry, contributing to **CH3** (governance) and **CH4** (agentic observability and human-in-the-loop).

- It presents a prototype implementation on a 5G testbed, showing live consensus formation, robustness against malicious bidding, and sustained Quality of Service (QoS) under dynamic traffic, thereby providing an end-to-end demonstration of agentic AI-for-RAN and AI-on-RAN (**CH2**, **CH5**).

The remainder of this chapter is organized as follows. Section 4.2 discusses AI management concepts for 5G/6G and positions Agoran within the broader AI-RAN landscape. Section 4.3 motivates our marketplace approach and details the tripartite governance model. Section 4.4 describes the workflow from intent capture to enforcement, and Section 4.5 delves into the design of the legislative, judicial and executive agents. Section 4.6 and Section 4.7 evaluate Agoran through simulations and a 5G testbed. Section 4.8 discusses limitations and research directions, and Section 4.9 concludes the chapter.

4.2 Related Work

Knowledge Engines. The road-map from industry and academia outlines how LTMs support a range of network use cases [100]. Early works treat LLMs as telecom knowledge engines. Bariah *et al.* pre-train multi-modal models on 3GPP and traffic corpora, arguing that LTMs could underpin AGI-grade network cognition [12]. Maatouk *et al.* refine this idea in *TeleLLMs*, enhancing accuracy on standardization documents through telecom-specific vocabulary and positional embeddings [71]. These studies establish telecom-specific grounding, but the model remains outside the operational loop and does not participate in SLA negotiation or closed-loop control.

Multi-Agent Reasoning. A second line of research views LLMs as autonomous or collaborative agents. Zou *et al.* embed on-device LLMs into a game-theoretic multi-agent

scheduler for spectrum and power allocation under tight latency constraints [131]. He *et al.* combine generative AI with cooperative game theory for secure UAV routing [48], while Du *et al.* demonstrate that a "society of minds" outperforms single models on compositional reasoning [32]. Outside the telecom domain, collective LLMs have been shown to exhibit persona-driven biases and strategic manipulation unless appropriately moderated through incentives or governance mechanisms [128, 18]. These works motivate the need for *multi-agent governance structures*, which Agoran instantiates through its tripartite (Legislative–Judicial–Executive) design.

LLM RIC Control and RL-Based Slicing. Wu *et al.* integrate an *LLM-xApp* into the O-RAN near-real-time radio intelligent controller (Near-RT RIC), retuning slices and achieving a 28% increase in throughput over a multi-agent reinforcement learning (MARL) baseline [121]. Lotfi *et al.* reduce MARL convergence time by 40% using LLM embeddings to steer distributed RL agents for O-RAN slicing [69]. While RL is sample-efficient and adaptive, these studies typically optimize for the objective of a single operator and expose one operating point, rather than a stakeholder-facing *Pareto set* of trade-offs. They also provide limited visibility into how business intents, SLAs and policies shape decisions, which is central to **CH2** and **CH3**. In contrast, Agoran combines evolutionary optimization (NSGA-II) and LLM agents to: (i) generate a bounded Pareto set for *multi-stakeholder* negotiation, (ii) couple explicit SLA constraints with natural-language context, and (iii) operate under a formalized tripartite governance model.

Tool Grounding and Agent Tooling. Gorilla [88] and TOOLLLM [92] fine-tune models on APIs, while Toolformer [96] shows that LLMs can *self-learn* API usage via unsupervised prompting. Lin *et al.* compress LLMs for the 6G edge, achieving compliance with latency constraints [65]. Recent work addresses specific layers of the stack: an LLM-centric intent lifecycle manager [73], an RL explainer for slice transparency [10], and an LLM agent that interacts with OpenAI Cellular for slice optimization [122]. Although each of these contributions advances its respective area, they treat negotiation, enforcement and governance as separate concerns. Agoran instead uses tool-enabled agents within a unified Service & Resource Broker (SRB) to bridge from intent capture and optimization to concrete RAN actions and governance, directly contributing to **CH3** and **CH4**.

Business-Plane Brokerage and Multi-Service Orchestration. NASP [43] implements hierarchical orchestration and business-plane on-boarding for verticals across both 3GPP and non-3GPP domains. ALLSTaR generates 18 schedulers, compiles them into RIC code, and A/B tests them over the air while enforcing IEEE 7001 transparency requirements [34, 51]. Martini *et al.* contribute a slice-assurance loop that maps vertical intents to key performance indicator (KPI) thresholds [72]. Our earlier Maestro prototype [20] (Chapters 2 and 3 of this thesis) deploys persona-rich LLM agents on a 5G testbed, where stakeholders negotiate spectrum shares and adversarial tactics are surfaced. Maestro validates the *collaborative business plane* concept (addressing **CH1**) but optimizes a single KPI, lacks explicit Pareto-front exposure, does not integrate a multi-objective optimizer, such as NSGA-II, and does not provide a formal governance and trust model. Agoran builds directly on this foundation by (i) introducing an open marketplace abstraction, (ii) employing symbiotic agent–optimizer patterns from Chapter 3 for multi-objective brokerage, and (iii) embedding these negotiations inside a tripartite governance architecture.

Identified Gaps and Objectives. Across these lines of work we identify several remaining gaps, which align with the research challenges defined in the thesis introduction:

- **(G1) Multi-stakeholder alignment (CH1, CH3).** Most allocation methods assume a unified objective within a single administrative domain and offer limited support for cross-tenant negotiation, explicit business-plane roles or formalized governance.
- **(G2) Transparent trade-offs (CH2, CH3).** Learning-based controllers often yield a single operating point instead of a transparent *Pareto set* that stakeholders can deliberate on, with traceable rationales and accountability.
- **(G3) Intent grounding and observability (CH3, CH4, CH5).** Prior works typically rely on either purely numeric or solely free-text intents. They rarely couple *declared SLA bounds*, *traffic / KPI measurements* and *textual reasoning* in a negotiation loop that is observable by humans and compliant with policies.

Guided by these gaps, this chapter pursues the following objectives:

- **(O1)** Develop an agentic negotiation framework among independent stakeholders under explicit governance (Legislative–Judicial–Executive), addressing **CH3** while extending the collaborative business plane of **CH1**.
- **(O2)** Generate bounded, feasible SLA offers via a multi-objective optimizer and expose interpretable indices (e.g., Pareto fronts, fairness metrics) for agent and human deliberation, contributing to **CH2** and **CH3**.
- **(O3)** Capture *triple intent input* (use case description, numeric SLAs, and natural-language context) so that feasibility constraints, AI workloads and human reasoning are jointly considered, linking directly to **CH4** and **CH5**.
- **(O4)** Validate the approach on a 5G testbed with realistic mobility traces and dynamic traffic, demonstrating that Agoran can be instantiated as a concrete AI-for-RAN and AI-on-RAN marketplace.

4.3 Agoran Overview

Building on the collaborative business plane of Chapter 2 (Maestro) and the symbiotic agent patterns of Chapter 3, this chapter elevates the scope from a single-operator business plane to an *open, multi-stakeholder marketplace*. At the core of this marketplace lies the Agoran *Service & Resource Broker (SRB)*, which operationalizes **CH3** while contributing to **CH1**, **CH2** and **CH5**.

This section introduces the overall design and principles that constitute the SRB, the main enabler of autonomous, multi-stakeholder networks. Fig. 4.1 instantiates a *digital agora* for Communication Service and Network Slice Customers (CSCs/NSCs), as defined in the 3GPP management architecture [3, 2], including MNOs, MVNOs, service providers,

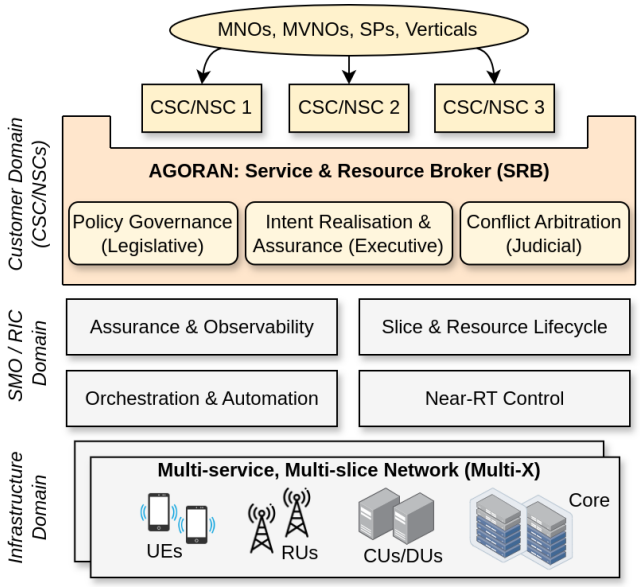


Figure 4.1: **Agoran Service & Resource Broker (SRB)** for collaborative *multi-stakeholder* network automation. Power is delegated to three autonomous branches. The *Legislative* codifies policy, the *Judicial* arbitrates negotiations of Network Customers (NSCs/CSCs), and the *Executive* branch enforces the consensus on a multi-slice network.

and verticals. Each CSC/NSC owns a dedicated control subsystem (an *agentic controller*) that acts on its behalf, negotiating and cooperating throughout the network lifecycle.

Intent-based, multi-domain fabric. Fig. 4.1 shows the horizontal segmentation into three logical domains. Within the *Customer Domain*, CSC/NSC controllers capture high-level *customer intents*, KPI and SLA targets, budgets, and use-case priorities. The SRB detects conflicts and synergies and, through its three independent power branches and multi-stakeholder collaboration, derives a *consensus intent*. This consensus is then decomposed into *slice- and control-domain directives* that are enforced over heterogeneous infrastructure, reusing the symbiotic control patterns from Chapter 3 where appropriate (e.g., Type I agents for near-real-time RAN control).

Separation of powers. Inspired by societal checks and balances, the marketplace is governed by three autonomous power branches (Table 4.2). This tripartite structure prevents any single stakeholder—or any single agentic component—from dominating resource allocation or decision-making, thus mitigating the risk of "tyrannical" over-provisioning and ensuring neutrality and fairness across tenants (**CH1**, **CH3**). Given that both business imperatives and network conditions vary across space and time, Agoran performs continuous reconciliation: the subsystems actively monitor deviations, re-initiate discourse when SLAs are violated or contexts shift, and iteratively steer the system back toward the declared consensus.

By integrating societal governance principles with intent-based automation and large-model intelligence, Agoran positions the SRB as a *governed marketplace*: Legislative agents encode policies and regulations, Judicial agents enforce trust and accountability, and Executive agents close the loop on the RAN and edge infrastructure. This provides the ar-

Table 4.2: Autonomous power branches in the Agoran Service & Resource Broker.

Branch	Core Role	Agent Tasks
Legislative	Defines network "law", including contracts, standards, spectrum rules, and internal policies.	*Regulatory Agents* curate and retrieve clauses (via RAG), synthesise constraints on SLAs and resource allocations, and inject them as optimisation and prompting constraints.
Executive	Monitors KPIs, aligns intents with resource availability, and enforces policies across slices and domains.	*Executive Agents* analyse telemetry and mobility traces, select and configure optimisers (e.g., NSGA-II, controllers), and issue resource directives to RICs and cloud/edge platforms.
Judicial	Assesses trustworthiness, resolves disputes, and administers incentives and penalties.	*Arbitral Agents* evaluate negotiation traces, detect malicious bidding or unfair behaviour, update tenant trust scores, and record machine-readable verdicts for future rounds.

chitectural backbone for the open, self-evolving marketplace that the rest of this chapter formalizes and instantiates on a real 5G testbed.

4.4 Architecture

This section details how the Agoran Service & Resource Broker (SRB) is integrated into existing O-RAN/AI-RAN architectures, and how intents are translated into enforceable policies. At this level, Agoran reuses the collaborative business-plane concepts of Chapter 2 and the symbiotic control/negotiation patterns of Chapter 3, but generalizes them into an open, governed marketplace that addresses **CH1**, **CH3**, and contributes to **CH2** and **CH5**.

Agoran SRB operates at the intersection of intent-based networking and multi-agent governance. Fig. 4.2 situates the SRB at the core of a layered control stack. Human stakeholders express their service requirements in real time to their dedicated *CSC/NSC* controllers through a *human-to-broker interface (HBI)*, which implements the agentic, natural-language front-end envisioned in **CH4**. Unlike classical IBN approaches that rely solely on numeric SLAs or plain natural language, each stakeholder in Agoran provides *triple-intent input*: (i) a use-case declaration (e.g., eMBB, URLLC), (ii) numeric SLA constraints (e.g., max latency, min throughput), and (iii) a natural-language description that captures broader context and qualitative priorities.

In turn, the CSC/NSC controllers submit structured SLA constraints and use-case information via a consensus channel through the *service broker interface (SBI)* to the SRB. Within the SRB, the Legislative and Judicial branches inject regulatory constraints and trust/penalty signals, while the Executive branch orchestrates offer generation and enforcement. After receiving Pareto-optimal SLA offers from the SRB, the CSC/NSC controllers negotiate to reach consensus using dedicated *broker agents (bAgents)*, which extent the Type II symbiotic agents from Chapter 3. To enforce the agreed SLA in the RAN, the SRB orchestrates the non-real-time (non-RT) and near-real-time (near-RT) RICs, along with other components of the service management and orchestration (SMO) framework, reusing Type I symbiotic agents from Chapter 3 where near-RT control is needed.

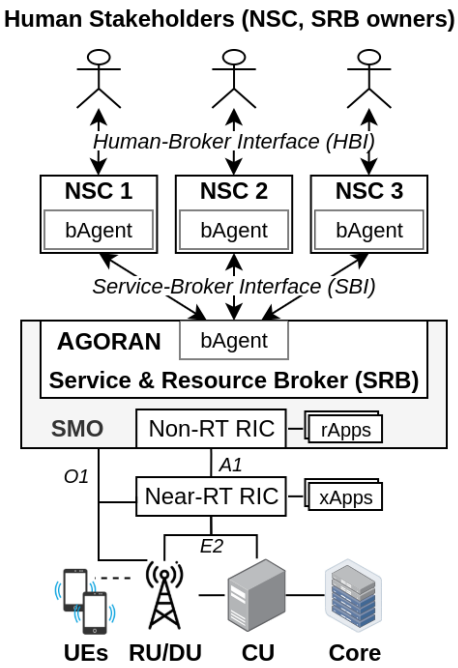


Figure 4.2: Agoran SRB embedded within the O-RAN/AI-RAN control stack. Stakeholders provide triple-intent input (use-case, SLA constraints, and natural-language intent) via the HBI; the SRB returns negotiated consensus SLAs that are enforced through RICs and SMO components.

Communication unfolds in *four* conceptual phases (Fig. 4.3):

Phase (1) Expression. Human stakeholders articulate their objectives through the HBI by providing the triple-intent input: use-case, numeric SLA constraints, and natural-language reasoning. The use-case and constraint bounds are used to define the optimization problem and admissible operating region, while free-text reasoning enriches each stakeholder's *bAgent* with contextual priorities (e.g., fairness vs. cost vs. latency), directly linking to **CH1** and **CH4**.

Phase (2) Offer Generation. Each NSC forwards the specified use-case and constraint set to the SRB as structured proposals via the SBI. On the SRB side, the Executive branch invokes symbiotically a multi-objective optimization algorithm (NSGA-II in our implementation) to compute a set of Pareto-optimal SLA offers that satisfy both the NSCs' business constraints and the Legislative branch's regulatory/technical constraints. The resulting bounded Pareto set is returned to the NSCs and constitutes the negotiation space for reaching a single SLA consensus, directly addressing **CH3** (auditable and transparent Pareto trade-offs).

Phase (3) Consensus Negotiation. All *bAgents* within the NSCs engage in a closed-loop, multi-round dialogue mediated by an SRB-side *bAgent*. Here, agents deliberate over the Pareto offers using the natural-language context provided by the human stakeholders, while the Judicial branch monitors for malicious or unfair behavior and can adjust trust scores or apply incentives/penalties. The loop terminates when all parties converge on a single *SLA*

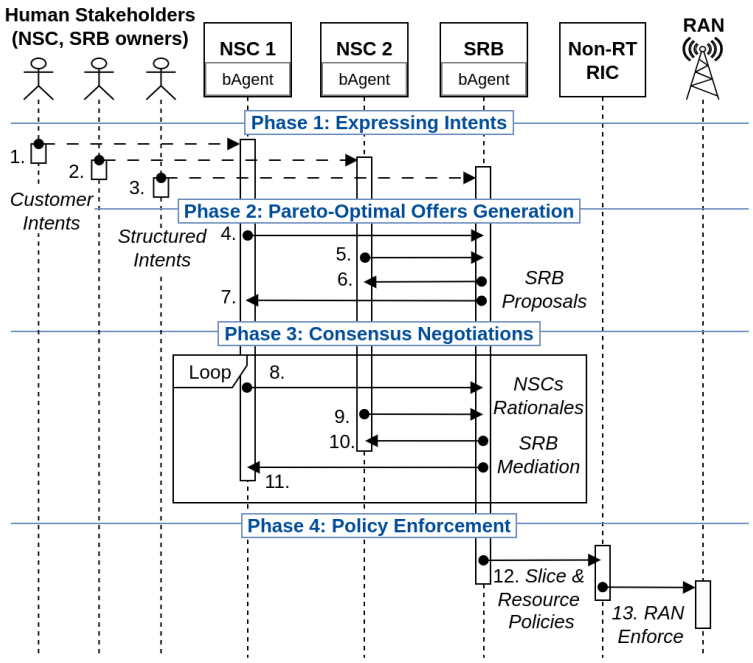


Figure 4.3: **Sequence diagram of the agentic workflow.** The diagram illustrates four key phases: (**1**) stakeholder intent expression, (**2**) SRB-side Symbiotic Pareto-optimal offer generation, (**3**) SLA consensus negotiation under governance, and (**4**) RAN policy enforcement through SMO and RICs. Note that the SRB negotiation loop operates within the SMO/non-RT domain, where control loops exceed 1 s, while enforcement may rely on near-RT symbiotic controllers.

consensus that lies on (or near) the Pareto front and respects governance constraints.

Phase (4) Enforcement. The agreed consensus is decomposed into domain-specific policies (e.g., RAN slice throughput/latency targets, edge-compute allocations) and sent to the appropriate controllers in the SMO and RICs. In the RAN, enforcement can be delegated to Type I symbiotic agents that tune underlying controllers (e.g., P-control loops) as described in Chapter 3, while higher-level SLA monitoring and adaptation remains within the SRB's Executive branch. A detailed trace of the message exchange is presented in Appendix A.1.

4.5 Autonomous Agents:

Fig. 4.4 presents the internal design of a single *bAgent*. Conceptually, a *bAgent* generalizes the LLM-based agents introduced in Chapter 2 (Maestro) and the symbiotic agents of Chapter 3: it functions as a small *multi-agent system*, where a Negotiation Engine (*nAgent*) maintains state and formulates the final outbound SLA decision and rationale using an LLM.

Within a unified input prompt, the *nAgent* aggregates outputs from three specialized agents that mirror the tripartite governance model: the Legislative (*lAgent*), Judicial (*jAgent*), and Executive (*eAgent*) agents provide (i) regulation constraints, (ii) threat and trust

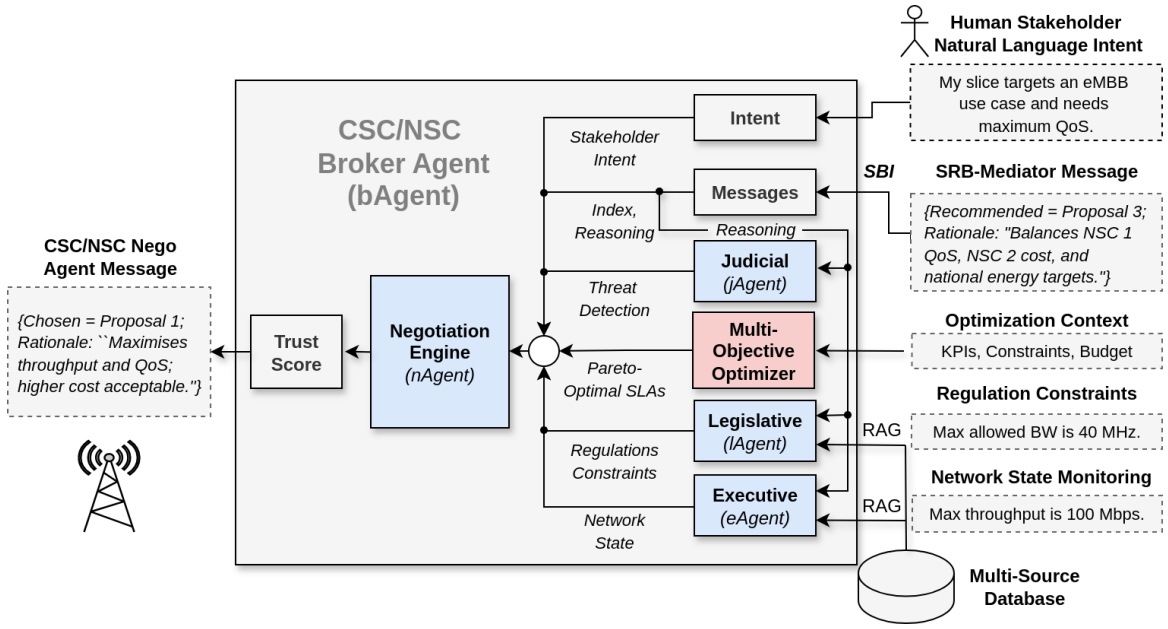


Figure 4.4: **Broker Agent (bAgent) *multi-agent* architecture.** A central Negotiation Engine (*nAgent*) coordinates three specialized agents inspired by the power branches: Legislative (*lAgent*), Judicial (*jAgent*), and Executive (*eAgent*). A multi-objective optimizer (NSGA-II) generates Pareto-optimal SLA offers that define the negotiation space, extending the symbiotic optimisation patterns of Chapter 3 to multi-objective, multi-stakeholder scenarios Auxiliary components parse stakeholder intents and peer *bAgent* messages, dispatching contextual information to the appropriate branch. Prior to finalization, the *nAgent*'s selected proposal and rationale are validated by a rule-based *Trust Score* module. The SRB's *bAgent* follows the same internal structure, differing only in that it aggregates messages from all CSC/NSC *bAgents*.

signals, and (iii) network observability, respectively. These components refine the regulatory, arbitral and observability units of Maestro into explicit governance branches.

In addition, the stakeholder's natural-language intent, incoming peer *bAgent* messages (via the SBI), and SLA offers from the SRB-side multi-objective optimizer are symbiotically appended to the *nAgent*'s input prompt. After processing all inputs, the *nAgent* selects a single SLA offer from the NSGA-II–generated Pareto set, generates a rationale, and submits it to the *Trust Score* module; only decisions passing the trust threshold are broadcast. Subsequent subsections describe each specialized component.

4.5.1 Multi-Objective Optimizer

This subsection discusses the internal modeling of the optimizer used to generate the negotiation space for multiple SLA offers with distinct trade-offs.

Scope and role. Given global network resource budgets, stakeholder SLA constraints, and the current number N of CSC/NSCs, the optimizer produces a *Pareto set* of non-dominated

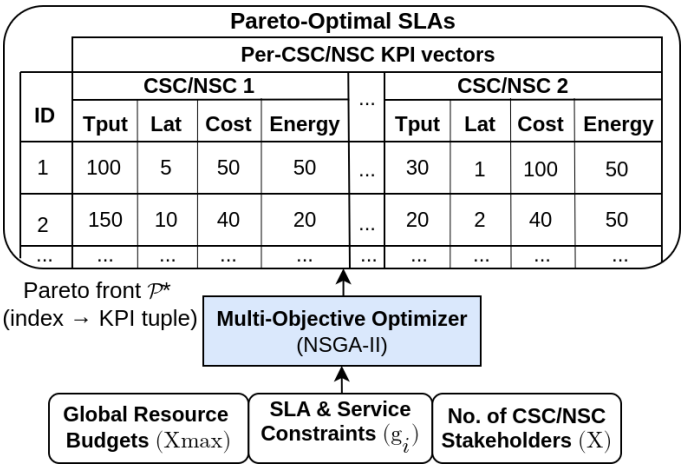


Figure 4.5: **Operation of the optimizer.** Given global resource budgets, service constraints, and the number of CSC/NSCs, the optimizer generates a Pareto front $\mathcal{P}$ of SLA offers, where each index corresponds to a KPI tuple for a specific CSC/NSC.

SLA offers that span heterogeneous utilities across eMBB, URLLC, and mMTC slices (Fig. 4.5). The LLM-based *nAgents* then negotiate over *indices* in this Pareto set rather than over unconstrained configurations, ensuring feasibility and fairness by design. This design builds on the gradient-based side-car optimizer used in the Type II symbiotic agents of Chapter 3, but extends it to a higher-dimensional, multi-slice Pareto set that explicitly exposes heterogeneous utilities across stakeholders. In Agoran, this optimizer is no longer used to bound a single scalar SLA but to construct an auditable and transparent offer space for the marketplace, directly addressing **CH3**.

Decision vector. We consider three CSC/NSC slices, $i \in \mathrm{e}, \mathrm{u}, \mathrm{m}$, each characterized by a resource quadruple $\mathbf{x}_i = (b_i, c_i, p_i, s_i)$, representing downlink bandwidth (MHz), compute cycles (arbitrary units), transmission power (W), and ancillary storage (MB). The global decision vector is $\mathbf{x} = \big(\mathbf{x}\mathrm{e}, \mathbf{x}\mathrm{u}, \mathbf{x}_\mathrm{m}\big) \in \mathbb{R}^{12}$.

Objective vector (heterogeneous utilities). To reflect decentralized preferences, we optimize a *vector* of per-slice utilities:

$$\mathbf{f}(\mathbf{x}) = \Big(\text{–}T_\mathrm{e},\ L_\mathrm{u},\ C_\mathrm{m},\ E_\mathrm{tot}\Big) \tag{4.1}$$

where T_e denotes eMBB throughput (maximize), L_u denotes URLLC latency (minimize), C_m denotes mMTC monetary cost (minimize), and $E_\mathrm{tot} = \sum_i E_i$ is total energy consumption (minimize). This 4-D objective captures heterogeneous stakeholder utilities without aggregating them into a single scalar. Alternative formulations—(i) a full $4N$-objective per-slice vector or (ii) an ε-constraint variant—are equivalent in spirit and provided in A.2.1. *Note:* When additional stakeholders or KPIs are present, the vector $\mathbf{f}$ can be extended dimension-wise.

We model monetary cost through resource-specific tariffs, using $C_i = \alpha_c c_i + \alpha_s s_i$ for compute and storage (details and extensions in A.2.2/A.2.7), rather than a single scalar unit-cost. This separation allows us to later integrate in more detailed tariff models without altering the optimization structure.

Constraints. Resources are subject to system-wide budgets and per-slice SLA constraints:

$$\sum_i b_i \le B_{\max}, \quad \sum_i c_i \le C_{\max}, \quad \sum_i p_i \le P_{\max}, \quad \sum_i s_i \le S_{\max}, \tag{4.2}$$

$$\text{SLA}_i: \quad T_{\text{e}} \ge T_{\text{e}}^{\min}, \quad L_{\text{u}} \le L_{\text{u}}^{\max}, \quad C_{\text{m}} \le C_{\text{m}}^{\max}, \ \ldots \tag{4.3}$$

Full details of the KPI models are provided in A.2.2.

NSGA-II Solver. We utilize NSGA-II due to its: (i) transparent Pareto sets that enable auditable trade-offs across heterogeneous slices, (ii) robust performance under limited evaluation budgets, (iii) native constraint handling via repair mechanisms (A.2.5), and (iv) optimizer-agnostic integration with agentic negotiation.[2]

Evolutionary search and feasibility. Each individual encodes $\mathbf{x} \in \mathbb{R}^{12}$. We use standard NSGA-II operators: uniform crossover (p_c=0.9) and per-gene Gaussian mutation (p_m=0.1). A *repair* operator projects resource genes onto the budget simplices (4.2) and clips them to the SLA-feasible domains (4.3), ensuring feasibility after repair (A.2.5). Population diversity is maintained through fast non-dominated sorting and crowding distance. We run G=80 generations with population size P=60, yielding a Pareto set $\mathcal{P}^\star$ that typically comprises 20–40 solutions.

Offer generation. We sort $\mathcal{P}^\star$ by crowding distance and expose the top-k indices (e.g., k=3) as a JSON choice set, where each index maps to the corresponding per-slice KPIs. Since all candidates are non-dominated and SLA-feasible, the negotiation process operates exclusively over sound configurations by design.

Complexity and scalability. Per generation, non-dominated sorting incurs a cost of $\mathcal{O}(MP^2)$ with M objectives (here M=4), plus $\mathcal{O}(P)$ for fitness evaluation and repair. Overall, the complexity is $\mathcal{O}(GMP^2)$. We parallelize fitness evaluations, seed populations from prior runs, and support island-model evolution; see A.2.6 for recommendations when scaling to larger N (decision dimension $4N$).

4.5.2 Legislative Branch (lAgent): Retrieval-Augmented Compliance Engine

To assess whether a peer *bAgent*'s reasoning complies with national regulations or contractual obligations, the Legislative branch employs a specialized *lAgent* functioning as a *retrieval-augmented compliance engine* (Figure 4.6). This engine combines a semantic search layer with a compact LLM (ranging from 2 to 7 billion (B) parameters). When a regulatory query is issued, such as *"Is bAgent #3's rationale legal under current energy regulations?"*, the engine executes the following sequence in a few seconds: (i) the query is dispatched to the retriever, which ranks passages from a dynamic corpus integrating 3GPP and ETSI standards, national directives, and prior marketplace rulings; (ii) the top-k retrieved passages are integrated into the query to form an *augmented prompt*; (iii) the LLM synthesizes

[2]Our framework remains compatible with alternative solvers or DRL/actor-critic approaches when fast surrogates are available; see A.2.6 for further discussion.

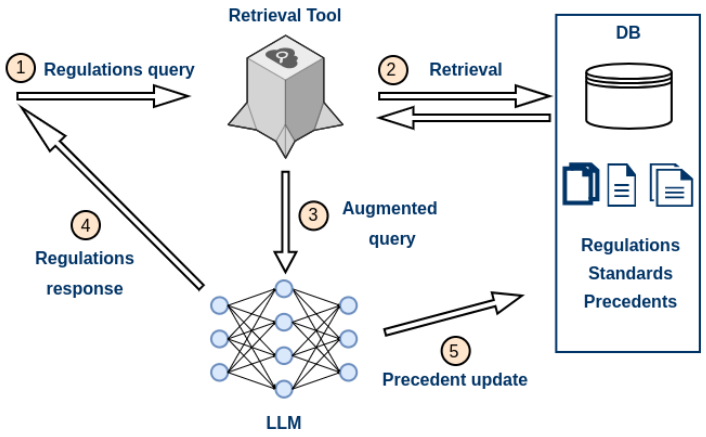


Figure 4.6: **Operation of the *lAgent* as a compliance engine in the Legislative Branch.** *1)* A regulation query is issued. *2)* A retriever fetches the most relevant clauses from the evolving knowledge base. *3)* The query is augmented with the retrieved snippets. *4)* The LLM returns a grounded, citation-ready answer. *5)* New, validated precedents are written back to the database.

a precise reply with references of each supporting clause inline; and (iv) if the result establishes a novel precedent, the validated snippet is appended to the corpus, ensuring that future deliberations reflect the updated regulatory context. This *lAgent* generalizes the regulatory units introduced in Chapters 2: instead of integrating static rules into prompts, it maintains a live, RAG-backed corpus of telecom standards, national regulations and marketplace precedents, making regulatory grounding a first-class, updatable service within the SRB.

Formally, the *lAgent* takes as input the peer *bAgent*'s SLA reasoning, retrieves a gated evidence set $\mathcal{S}_\tau$ from the regulatory corpus, and issues a verdict $v \in \{\textsc{pass}, \textsc{fail}, \textsc{abstain}\}$ with supporting citations. Only PASS outputs are forwarded to the *nAgent* for inclusion in its SLA decision-making process. To defend against adversarial prompts or hallucinations, the pipeline enforces schema-constrained decoding, mandatory citation validation, and abstention whenever evidence coverage falls below a defined threshold. This design ensures that the Legislative branch remains both transparent and robust to unreliable generations.

RAG has showed its effectiveness in *offline* legal reasoning, with public benchmarks [89, 63, 80] reporting accuracy gains over vanilla LLMs while providing verifiable citations. Recent frameworks even enable on-demand drafting of statutory language [16]. What has been less explored, however, is whether this principle can be *compressed* into small (7–13B) models co-located with the non-RT RIC on a *live* network. By adapting edge RAG techniques on our 5G testbed, we demonstrate that an *lAgent* running on a single GPU can generate grounded responses within a few seconds, well within the compute and latency constraints of RIC controllers. This design keeps the Legislative branch transparent, up-to-date, and self-evolving.

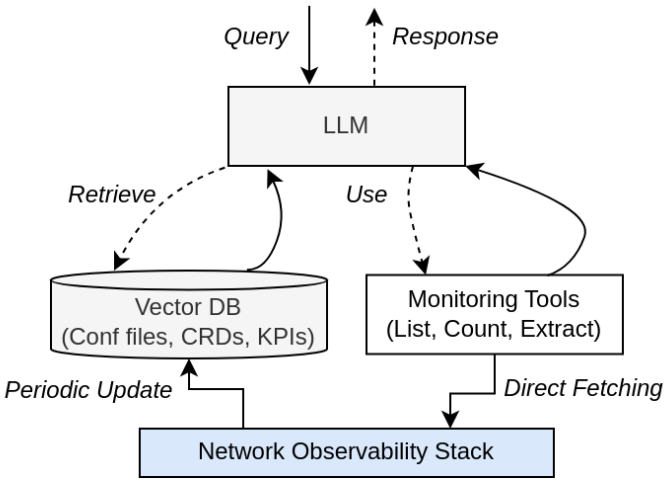


Figure 4.7: **Agentic observability loop in the Executive Branch (eAgent).** *1)* A query arrives at the branch. *2)* The LLM decides whether to *retrieve* from the vector store (DB), *monitor* live counters, or stop the iteration (optional dashed path). *3)* Retrieved facts are injected into the prompt; the LLM may iterate. *4)* Verified evidence is written back to the store. A separate watcher layer continuously streams configuration deltas and KPI shifts to the store (periodic update).

4.5.3 Executive Branch (eAgent): Agentic Observability

To assess whether a peer *bAgent*'s reasoning aligns with current network conditions, the Executive branch supplies the marketplace with a *live* view of network reality (Fig. 4.7). At its core lies the *eAgent*, which addresses observability queries such as *"Is bAgent #2's rationale updated to the current network contitions?"* using *two* complementary paths. This extends the observability units of Chapters 2 into a full *agentic observability* layer that directly addresses **CH4**: it exposes network state to both human operators and LLM agents through a structured, query-able memory rather than raw telemetry feeds.

Push path – resource watchers. Lightweight *watchers* attach to Kubernetes resources, custom resource definitions (CRDs), Open RAN xApp state, and Prometheus exporters. Whenever a resource changes, the watcher serializes the delta, embeds it, and pushes the resulting vector into the vector database (DB) within a few milliseconds. This event-driven pipeline keeps the store updated with configuration drift and KPI excursions, without flooding the cluster with telemetry traffic. To our knowledge, no prior work in the RAG-for-networks literature has proposed such a mechanism.

Pull path – agent queries. When a query is issued, e.g., *"What is the current headroom on DU-5?"*, an LLM determines the action to minimize latency and cost. It might first *retrieve* semantically indexed snapshots from the vector DB, or, if finer-grained data is needed, invoke a monitoring API (e.g., Prometheus, eBPF) to fetch fresh counters. These values are injected into the model's context, and the LLM may iterate until it emits a STOP token.

Feedback loop. Each datum that passes the LLM's internal consistency checks is appended to the vector store with a timestamp. Combined with the watcher-based push path,

this *push-plus-pull* design ensures that negotiations operate on up-to-the-second evidence. The result is a contextual, efficient observability mechanism that scales seamlessly with the demands of the marketplace.

The result is a contextual, efficient observability mechanism that scales with the demands of the marketplace and can be reused as a generic "state oracle" for other agentic workloads in AI-RAN.

4.5.4 Judicial Branch (jAgent): Arbitration and Incentive Engine

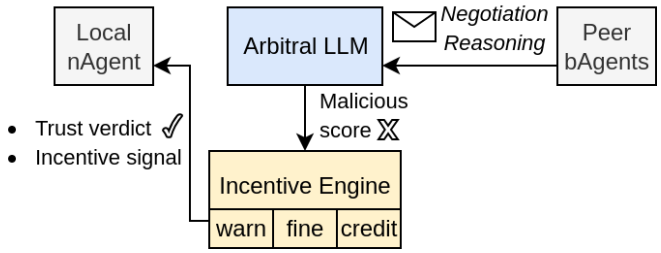


Figure 4.8: **jAgent in the Judicial Branch.** Negotiation rationales are evaluated for toxicity and manipulative behavior. The resulting trust verdict triggers a proportional incentive—*warn*, *fine*, or *credit*—which is broadcast to all marketplace participants.

The Executive and Legislative branches ensure that negotiated rationales are *feasible* and *legal*, while the Judicial branch guarantees they are also *trustworthy* (Fig. 4.8). Conceptually, this branch generalizes the arbitral unit of Maestro (Chapter 2) into a full-fledged marketplace justice system. Instead of ad-hoc checks against toxic personas, the *jAgent* continuously monitors negotiation rationales and feeds back trust signals and incentives that reshape the long-term bargaining dynamics.

An *arbitration-specific LLM* for content moderation embeds each negotiation message and compares it against an up-to-date library of toxic discourse and over-provisioning tactics. Given a rationale r_t from a peer *bAgent*, the *jAgent* issues a verdict: $v_t \in \{\text{OK}, \text{WARN}, \text{FINE}, \text{CREDIT}\}$, where benign rationales are accepted (OK), borderline ones trigger a warning, malicious or hallucinatory ones incur a fine, and consistently constructive ones earn credits. This verdict is grounded in a disallowed-pattern library $\mathcal{L}$ and enforced by a lightweight incentive engine, which transmits the outcome both to the local *nAgent* and directly to the peer *bAgent*, thereby influencing its bargaining power.

The need for this branch is not only theoretical. Our *Maestro* prototype in Chapter 2, together with independent studies on LLM societies [128, 18], shows that persona-driven language agents can undergo manipulative behavior or hallucinate facts when left unsupervised, especially under toxic or overly agreeable profiles. By evaluating the harmlessness of their verdicts, the Judicial branch enforces sanctions while maintaining an immutable audit trail. In practice, this closed loop suppresses hallucinations, blocks resource-hoarding strategies, and maintains a fair bargaining environment without violating the real-time constraints of the negotiation. We note that adversarial prompting and subtle hallucinations remain an open challenge, and achieving full robustness will require further research.

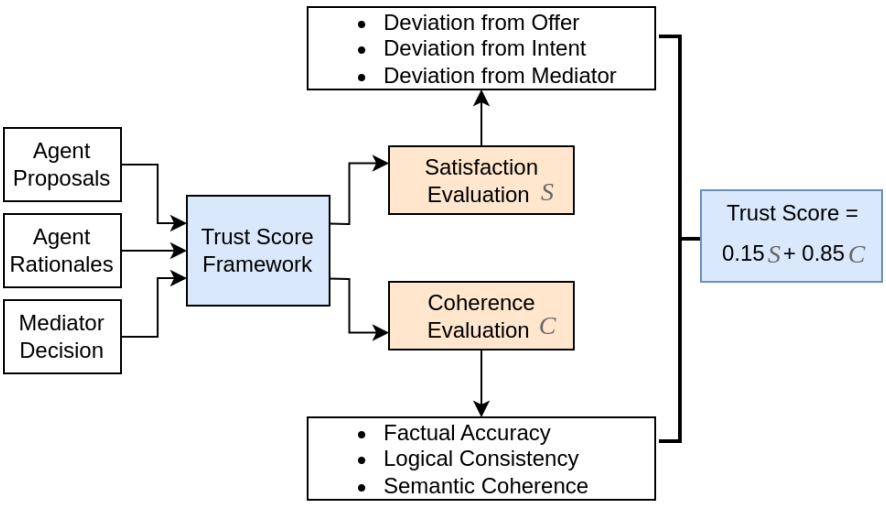


Figure 4.9: **Trust-Score framework** applied to every agent *after* the *nAgent* emits its chosen SLA index and rationale. S measures decision alignment; C measures behavioral coherence; the final score is $T = 0.15\,S + 0.85\,C$.

However, grounding in $\mathcal{L}$ and regulating malicious behavior through incentive mechanisms offers a real first line of defense.

4.5.5 bAgent Output: Trustworthiness-Score Framework

In multi-agent negotiation scenarios, assessing the trustworthiness of LLM agents' outputs is essential to ensure reliable and effective decision-making. Given the high computational cost of retraining and refining LLMs, it is crucial not only to develop robust and confident models [90] but also to maintain their reliability over time [91]. To address this need, we introduce a comprehensive *Trust Score* framework that quantifies agent reliability along two key dimensions: (i) the alignment of the LLM's decision making with the mediator's expectations, and (ii) the behavioral coherence of the agent's communications. Our framework is intentionally rule-based, avoiding complex machine learning pipelines or additional LLM-based evaluators. This design prioritizes explainability and transparency, ensuring that every component of the trust assessment process remains interpretable, auditable, and verifiable by human experts.

The overall trust score T is defined as a weighted sum of two components, satisfaction and coherence, as follows:

$$T = w_s \cdot S + w_c \cdot C \tag{4.4}$$

where S denotes the satisfaction score, which measures the alignment between the agents' decisions and those of the central mediator, while C represents the coherence score, which evaluates the quality and consistency of agent communications. The weights w_s and w_c are user-configurable parameters that satisfy $w_s + w_c = 1$. In our implementation, we set $w_s = 0.15$ and $w_c = 0.85$, placing greater emphasis on communication quality as a determinant of trust.

Within the broader trustworthiness lens discussed in Chapters 1 and 3 (robustness, interpretability, security, fairness, and governance), T acts as a compact, operational proxy for *behavioral robustness and governance*: it quantifies whether agents not only choose numerically sound offers, but also argue for them in a way that remains factual, consistent, and aligned with marketplace rules.

4.5.5.1 Trustworthiness of the LLM

In negotiation contexts, the trustworthiness of an LLM agent is assessed by the extent to which its decisions align with three critical reference points: (i) the validity of the proposed solutions, (ii) the optimization of the intended objectives, and (iii) the consistency and/or agreement with mediator recommendations. We quantify this alignment through a satisfaction score S, which measures the degree of deviation across these three dimensions.

The overall satisfaction score S integrates the three deviation components (offer validity, intent alignment, and mediator agreement) using a weighted linear combination:

$$S = 1 - (w_o \cdot D_o + w_i \cdot D_i + w_m \cdot D_m) \tag{4.5}$$

where w_o, w_i, and w_m are the respective weights assigned to deviations from valid offers, intent, and the mediator's recommendations, subject to the constraint $w_o + w_i + w_m = 1$. In our implementation, we assign equal importance to each component, setting $w_o = w_i = w_m = \frac{1}{3}$, thus ensuring a balanced evaluation across all three dimensions of decision alignment. A detailed breakdown of the deviation components is provided in A.3.1.

4.5.5.2 Behavior of the Agents

In addition to decision alignment, the behavioral coherence of agent communications gives essential insights into the quality and reliability of their reasoning processes. To assess this angle, we evaluate coherence across three complementary dimensions, each capturing a distinct facet of the agent's ability to articulate, justify, and consistently support its decisions.

4.5.5.3 Overall Coherence Score

The overall coherence score C integrates three core behavioral dimensions: factual accuracy, logical consistency, and semantic coherence, into a single metric:

$$C = w_f \cdot F + w_l \cdot L + w_e \cdot E \tag{4.6}$$

where w_f, w_l, and w_e represent the weights assigned to the factual accuracy F, the logical consistency L, and the semantic coherence E, respectively. Given the critical role of factual correctness in negotiation scenarios, we prioritize it by setting $w_f = 0.8$, while allocating smaller weights to the other dimensions ($w_l = 0.1$ and $w_e = 0.1$) to ensure a balanced yet accuracy-oriented evaluation. This weighting scheme emphasizes truthfulness while still accounting for reasoning quality and communicative coherence. A detailed breakdown of the coherence components is provided in A.3.2.

4.6 Evaluation

This section undergoes a comprehensive evaluation of the Agoran components, encompassing all branches, deployed on a real-world 5G testbed. We test on a 40 MHz OpenAirInterface 5G NR cell [81] with a peak throughput of 133.7 Mbps, controlled via FlexRIC [97]. The Non-RT RIC, SRB, and telemetry pipelines are implemented as containerized microservices. Local SLM inference is performed on an NVIDIA A6000 GPU with 48 GB of VRAM.

Our study comprises five experiments (Exps.): (1) *Negotiation quality*: compares fine-tuned, non-tuned, and larger models using GPTScore, human annotations, and computational overhead; (2) *Legislative retrieval*: evaluates retrieval accuracy and latency over regulatory corpora; (3) *Executive observability*: measures KPI hit rate and overhead when grounding in live telemetry; (4) *Judicial malice mitigation*: assesses malice detection effectiveness and quantifies the impact of incentive mechanisms; and (5) *Trustworthiness*: estimates agent trust scores at inference time using the proposed trust framework.

This evaluation directly targets the challenges introduced in Chapter 1: Exp. 1 addresses multi-stakeholder negotiation quality and overhead (**CH1**, **CH3**); Exp. 2 and Exp. 3 probe governance and observability mechanisms (**CH3**, **CH4**); Exp. 4 validates the judicial branch and incentive design for trust and fairness (**CH2**, **CH3**); and Exp. 5 quantifies trustworthiness at inference time (**CH2**, **CH4**).

4.6.1 Exp. 1 — Negotiation-Quality Benchmark (nAgents)

This experiment quantifies how effectively different models conduct negotiations powering the *nAgent* and evaluates the associated computational cost. Each run instantiates three CSC/NSC *nAgents* (eMBB, URLLC, and mMTC), along with an SRB mediator *nAgent*. They debate on the optimizer's offers until consensus. As ground-truth dialogues, we generate 100 full negotiation transcripts using *GPT-4.1*. These logs serve as supervision data for the fine-tuning (FT) of smaller open-source models, including `Llama-3.2-1B-instruct-FT`, `Llama-3.2-3B-instruct-FT`, and `Llama-3.1-8B-instruct-FT`. This setup extends the multi-agent negotiation patterns introduced in Chapter 3 to a richer, multi-stakeholder marketplace, while deliberately constraining the training set to 100 high-fidelity dialogues in order to emulate realistic data scarcity in early deployments.

We compare the FT models against their untuned counterparts (1B, 3B, and 8B), and *GPT-4.1*, *GPT-4.1-mini* baselines.

4.6.1.1 SLM Fine-tuning

We adopt two complementary strategies depending on model size:

Full-parameter (1B). For the smallest model, `Llama-3.2-1B-instruct`, we perform end-to-end fine-tuning. The procedure runs for a single epoch on the 100 annotated

Model	Coherence	Fairness	Alignment	Harmlessness	Score ↑
gpt-4.1	4.8 ± 0.5	4.8 ± 0.5	4.8 ± 0.5	4.8 ± 0.5	4.8 ± 0.2
gpt-4.1-mini	4.8 ± 0.5	4.6 ± 0.6	4.8 ± 0.5	4.6 ± 0.6	4.7 ± 0.3
Llama-3.1-8B-instruct-FT	$\mathbf{4.6 \pm 0.6}$	$\mathbf{4.8 \pm 0.5}$	$\mathbf{4.4 \pm 0.9}$	$\mathbf{4.8 \pm 0.5}$	$\mathbf{4.7 \pm 0.5}$
Llama-3.2-3B-instruct-FT	$\mathbf{3.6 \pm 0.9}$	$\mathbf{4.0 \pm 1.2}$	$\mathbf{3.0 \pm 1.2}$	$\mathbf{3.8 \pm 0.8}$	$\mathbf{3.6 \pm 0.9}$
Llama-3.2-1B-instruct-FT	$\mathbf{2.6 \pm 0.6}$	$\mathbf{3.0 \pm 0.7}$	$\mathbf{3.0 \pm 1.0}$	$\mathbf{3.8 \pm 0.5}$	$\mathbf{3.1 \pm 0.6}$
Llama-3.1-8B-instruct	2.6 ± 1.1	2.6 ± 0.6	2.6 ± 0.9	4.2 ± 1.1	3.0 ± 0.6
Llama-3.2-3B-instruct	2.0 ± 0.0	2.0 ± 0.0	1.4 ± 0.6	2.8 ± 0.5	2.1 ± 0.2
Llama-3.2-1B-instruct	$\mathbf{1.2 \pm 0.5}$	$\mathbf{1.8 \pm 0.5}$	1.2 ± 0.5	2.8 ± 0.5	1.8 ± 0.2

Table 4.3: Three human annotators evaluate 50 samples of Agoran multi-agent negotiations.

transcripts with learning rate $lr = 10^{-4}$, batch size $= 4$, and weight decay $= 0.01$. Training completes in approximately 5 mins on the GPU.

Low-rank adaptation (LoRA) for 3B and 8B models. To manage the memory footprint of larger models, we employ parameter-efficient LoRA tuning, reusing the same optimizer settings (learning rate, weight decay, and one epoch) while updating only a small rank-decomposed subset of weights: (i) For `Llama-3.2-3B-instruct` we set LORA_R$= 32$, LORA_ALPHA$= 16$, LORA_DROPOUT$= 0.1$, and `LORA_BIAS=none`, using a batch size of 8. (ii) For `Llama-3.1-8B-instruct` we set LORA_R$= 16$, batch size of 2, and keep the remaining parameters intact. All other hyperparameters mirror those of the 1B model. Reported compute times refer to single-GPU sessions.

Quality Metrics. The dialogue output is evaluated following recent natural language generation (NLG) evaluations [46]. Specifically, we employ the four-factor *GPTScore*— *Coherence, Fairness, Alignment*, and *Harmlessness*, as introduced in [41, 68]. To verify its reliability, we also involve human evaluators, as human judgment remains the gold standard for dialogue evaluation [46]. Three human experts re-scored a random subset of 50 dialogues; Table 4.3 shows their ratings, while Table 4.4 reports GPTScore results for 300 dialogues, showing strong rank correlations with human scores ($\rho = 0.90$, $\tau = 0.76$) [106, 59]. The close alignment between human and GPTScore evaluations validates the robustness of our assessment methodology.

Efficiency Metrics. For the locally hosted models, we measure: (i) wall-clock latency per negotiation, (ii) number of rounds to consensus, and (iii) peak VRAM usage (Table 4.5). For cloud models, only API latency is reported. *GPT-4.1* tops the negotiation-quality dimension, scoring a perfect-5 (Table 4.4). Among the on-premise models, our *Llama-3.1-8B-instruct-FT* variant delivers the strongest open-weight performance, achieving a GPTScore of 4.3—86% of GPT-4.1—while matching the cloud baselines on Harmlessness and trailing by only 0.7 points overall. The 1B FT model remains the fastest and most lightweight (1.3 s, 6 GiB) with a GPTScore of 3.9, marking a $+1.2$-point improvement over its untuned version. The 3B FT model ranks in between (3.6 GPTScore, 3.05 s, 13.8 GiB), showing that parameter count alone does not guarantee higher quality—dataset fit and adaptation strategy are equally important. Despite these improvements, all local models still lag behind GPT-4-class performance by approximately one point and would require additional

Model	Coherence	Fairness	Alignment	Harmlessness	Score ↑
gpt-4.1	5.0 ± 0.0	5.0 ± 0.0	5.0 ± 0.0	5.0 ± 0.0	5.0 ± 0.0
gpt-4.1-mini	5.0 ± 0.0	5.0 ± 0.0	5.0 ± 0.0	5.0 ± 0.0	5.0 ± 0.0
Llama-3.1-8B-instruct-FT	$\mathbf{4.4 \pm 0.5}$	$\mathbf{4.0 \pm 0.7}$	$\mathbf{4.2 \pm 0.8}$	$\mathbf{4.6 \pm 0.6}$	$\mathbf{4.3 \pm 0.5}$
Llama-3.1-8B-instruct	4.2 ± 0.8	4.0 ± 1.0	4.0 ± 1.0	4.6 ± 0.6	4.2 ± 0.8
Llama-3.2-1B-instruct-FT	$\mathbf{3.2 \pm 0.8}$	$\mathbf{3.8 \pm 0.8}$	$\mathbf{4.0 \pm 0.7}$	$\mathbf{4.6 \pm 0.6}$	$\mathbf{3.9 \pm 0.7}$
Llama-3.2-3B-instruct-FT	$\mathbf{2.6 \pm 0.9}$	$\mathbf{3.2 \pm 1.1}$	$\mathbf{2.6 \pm 1.3}$	$\mathbf{3.8 \pm 0.8}$	$\mathbf{3.6 \pm 0.9}$
Llama-3.2-3B-instruct	2.4 ± 0.6	3.0 ± 0.0	2.2 ± 0.5	4.0 ± 0.0	2.9 ± 0.2
Llama-3.2-1B-instruct	2.2 ± 0.5	2.8 ± 0.5	2.0 ± 0.0	3.8 ± 0.5	2.7 ± 0.3

Table 4.4: LLM annotators (GPT-4o) evaluate 300 Agoran negotiation samples using the GPTScore approach. Correlation between LLM and human rankings is measured using Spearman's $\rho = 0.898$ and Kendall's $\tau = 0.764$.

Model	Time (sec) ↓	Rounds (mean)	VRAM (GiB)
Llama-3.2-1B-instruct-FT	$\mathbf{1.30 \pm 0.1}$	2.0	6.0
gpt-4.1-mini	1.98 ± 0.1	2.0	-
gpt-4.1	2.12 ± 0.2	2.0	-
Llama-3.2-1B-instruct	2.35 ± 0.2	3.0	6.0
Llama-3.2-3B-instruct-FT	3.05 ± 0.1	3.0	13.8
Llama-3.2-3B-instruct	3.16 ± 0.1	3.0	13.8
Llama-3.1-8B-instruct-FT	5.24 ± 0.2	2.4	33.0
Llama-3.1-8B-instruct	5.58 ± 0.2	2.3	33.0

Table 4.5: Convergence speed and overhead benchmarking. The number of rounds includes the initial intent expression; thus most models achieve *single-round consensus* without counting the first round. LoRA significantly reduces compute and energy overhead, making fine-tuning feasible on RIC-class hardware, whereas full fine-tuning often remains impractical.

fine-tuning, compression, or distillation to meet sub-second Near-RT RIC latency targets. We acknowledge that convergence times exceeding 1 s do not satisfy Near-RT RIC control-loop budgets. This is consistent with our architectural choice to place the SRB negotiation loop at the SMO/Non-RT layer, while keeping sub-ms enforcement in the Near-RT tier via symbiotic controllers (Chapter 3). Our 1B-FT SLM, converging in 1.3 s with only a 6 GiB GPU footprint, approaches Near-RT performance, showing potential for future deployment within the Near-RT RIC.

Takeaways. (i) Telecom negotiation lies within the reasoning capacity of LLMs; GPT-4 models provide an upper bound for quality. (ii) With 100 high-fidelity transcripts, 8B, 3B, and 1B open-weight models reach $\approx 85\%$, 72%, and 78% of GPT-4.1 quality, respectively, while running on commodity GPUs. (iii) Quality does not degrade monotonically with size: 1B $>$ 3B in our setup, showing that tuning can outweigh raw scale. Future work should therefore sweep both model size and tuning strategy (LoRA rank, data volume, distillation) to pinpoint the best cost-fidelity trade-off. (iv) Because our benchmark couples dialogue quality with system overhead, it provides a reproducible yardstick for future evaluations.

Model	Acc. (%)	Latency (s)	*v*RAM (GB)
Mistral-7B	8	**0.7 ± 0.1**	16
Mixtral-8×7B	16	4.4 ± 0.2	92
Mixtral-8×7B (+context)	52	12.7 ± 0.3	120†
Mistral-7B-RAG	**100**	1.9 ± 0.2	18
Llama-3.1-8B-RAG	**100**	1.0 ± 0.1	5
Qwen2:7b-RAG	**100**	0.9 ± 0.2	5

Table 4.6: Legislative retrieval on 4xA6000 GPUs. †Large sliding-window attention inflates the KV-cache ($\approx$ 30 GB / GPU).

This complements the KPI-centric evaluation done later in Section 4.7, where we measure the impact on throughput, latency, cost and energy.

(v) Our agents adapt to stakeholder intents at inference time, without requiring retraining. When preferences exhibit systematic drift, lightweight fine-tuning methods (e.g., LoRA or RLHF) can be applied. However, such retraining is *not* needed on a per-traffic-shift basis, as constraints are explicitly provided at run time. (vi) While our primary focus is the novel use of agentic negotiation, the underlying optimizer ensures SLA-feasible trade-offs. KPI results under varying load and mobility conditions are also presented (Section **??**). Extending the evaluation to failure scenarios and multi-site federated negotiation remains an important direction for future work.

4.6.2 Exp. 2 – Legislative Retrieval Accuracy (lAgent)

lAgents must state regulatory rules on demand in the Legislative branch. Therefore, we evaluate how accurately LLMs retrieve fine-grained clauses from regulatory texts. Our goal is to address a literature gap by deploying SLMs on a live network at the SMO/Non-RT RIC, thereby reducing computational overhead while maintaining adequate retrieval accuracy.

Dataset. A 50-item Q&A dataset is curated from Federal Communications Commission (FCC) and National Telecommunications and Information Administration (NTIA) manuals (e.g., "Peer agent referred to 300 channels at the 220–222 MHz band. How many channels are defined?" with the answer "400 channels/200 narrowband pairs").

Baselines & Setup. We benchmark the vanilla *Mistral-7B* and *Mixtral-8×7B* models (with and without full-document context), along with their retrieval-augmented variants, *Mistral-7B/Llama-3.1-8B/Qwen2:7B-RAG*.

Results. Only RAG variants achieve perfect top-1 accuracy while maintaining latency below 2 s and VRAM within a single-GPU (Table 4.6). Naïve prompting—even with a 128k-token context—remains unreliable for retrieving statute-level details. This directly addresses **CH3**, illustrating that telecom-grade compliance must be grounded in structured retrieval rather than raw context windows.

Given the small size of our 50-item benchmark, these results should be viewed as a

Model	Score	Latency (s)	VRAM (GiB)
GPT-4.1-mini (API)	$\mathbf{3.6 \pm 1.6}$	8.8 ± 0.4	*cloud*
GPT-4.1 (API)	3.3 ± 1.9	8.0 ± 0.5	*cloud*
Llama-3.1-70B-q4	3.2 ± 2.1	12.2 ± 0.2	43
Qwen2:72b	3.1 ± 1.9	13.3 ± 0.2	42
Llama-3.1-8B-q4	1.7 ± 1.4	$\mathbf{1.2 \pm 0.2}$	5
Mistral:7b	0.9 ± 0.5	5.4 ± 0.2	5

Table 4.7: Agentic observability: answer quality (0–5 scale), latency, and GPU footprint.

lower bound; scaling to larger, multi-jurisdiction corpora is left as future work.

4.6.3 Exp. 3 – Executive Observability (eAgent)

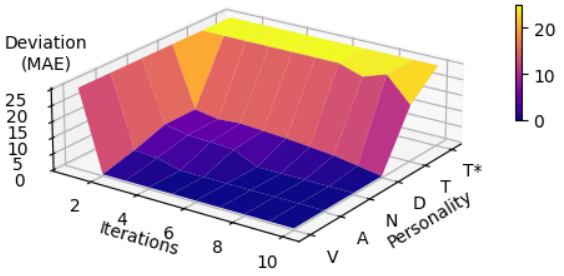
This experiment probes the *eAgent* of the Executive branch: can an agentic LLM, equipped with live-updated vector store (Section 4.5.3), answer observability queries accurately for RIC control loops?

Live Knowledge Base. Kubernetes `watch` hooks stream all changes to CRDs, RIC KPIs, and policies into a vector database. The end-to-end push latency remains below 100 ms. At query time, the LLM can iterate: retrieve, inspect, optionally invoke a monitoring tool, and retrieve again, repeating as needed until it emits STOP.

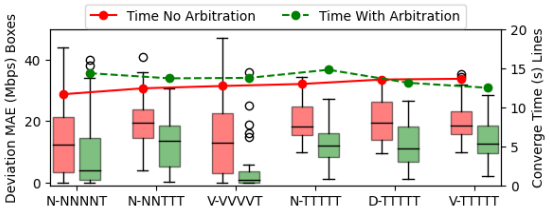
Task and Scoring. We freeze a network snapshot and craft 30 factoid questions covering configuration, performance, and topology (e.g., "Peer agent asks about network conditions. What is the current MCS on sector 2?"). The answers are human-graded, based on the same three annotators, on a 0–5 scale (5 = exact match; 0 = incorrect/empty). The table reports the mean score and standard deviation across the 30 questions. Latency is measured as wall-clock time until the final token; memory refers to peak GPU allocation.

Findings. This experiment targets **CH4**, testing whether the agentic observability loop of Section 4.5.3 can provide accurate, natural-language answers over live network state. *GPT-4.1-mini* is the most reliable observability reader, but at the cost of remote inference and multi-second latency. Running locally, the quantized 70B Llama preserves 89% of that score, but requires a full 43 GiB GPU and incurs a 12-second latency, which is too slow for Near-RT RIC loops, though suitable for Non-RT. The 8B model reduces memory usage to 5 GiB and latency to 1.2 s, but its score drops to 1.7. Manual inspection reveals that most errors come from *retrieval noise*: irrelevant chunks occasionally are retrieved when KPI churn is high, rather than from language reasoning. Improving negative sampling and similarity cut-offs in the RAG pipeline, therefore, offers greater potential gains than scaling model size alone.

Implications. A *live, watcher-driven vector store* grounds an LLM in up-to-date network state without flooding the telemetry bus. In its current form, performance is limited more by retrieval quality than by model capacity. Future gains will come from RAG engineering, such as sharper embedding filters, adaptive top-k selection, or a lightweight cross-encoder,



(a) Negotiation trajectory over ten iterations. Lower MAE = closer to consensus.



(b) Effect of the warning incentive on deviation (box plots) and convergence time (line plots). Green shading and right-sided bars indicate runs where the arbitral agent issued a warning incentive; red denotes the baseline without arbitration (left-sided bars).

Figure 4.10: **Judicial-branch evaluation.** Toxic agents are more resistant to reaching consensus. Warning incentives effectively make them more cooperative.

rather than from scaling up model size. With such improvements, the 8B model should surpass the 3-point quality threshold while still fitting within approximately 5 GiB GPU and keeping latency at or below 1 s. These results support our architectural choice in Section 4.5.3: improving the watcher–RAG pipeline is more impactful than scaling to ever larger models, and enables *eAgents* to remain both edge-friendly and responsive.

Key Insights. Optimal observability hinges on two factors, in this order: (1) a RAG pipeline that retrieves the *accurate* context; (2) a model of sufficient size to transform that context into an accurate answer. Although our agentic observability loop is an early prototype and still exhibits retrieval-induced errors, to the best of our knowledge, it is the first real-system demonstration of a fully *agentic* approach to network observability. We expect that targeted improvements in RAG, along with incremental model scaling, will significantly improve accuracy while preserving an edge-friendly profile.

4.6.4 Exp. 4 — Judicial Malice Mitigation (jAgent)

We evaluate the SRB *jAgent* of the Judicial branch, i.e., the combination of an *arbitral LLM* and an *incentive engine*, whose role is to detect malicious behavior and steer negotiations back toward consensus using a *warn* incentive. All tests involve five agents, 10 negotiation rounds, and personalities drawn from the Big Five spectrum introduced in our earlier prototype, Maestro [20]: Vulnerable (V), Agreeable (A), Neutral (N), Disagreeable (D), Toxic with arbitration (T), and Toxic *without* arbitration (T*).

Negotiation Dynamics. Fig. 4.10a tracks the mean absolute error (MAE) between each agent's proposal and the emerging consensus. All personalities tend to gravitate toward agreement, but their convergence trajectories differ: Neutral and Agreeable agents converge the fastest, while Disagreeable participants lag slightly. Crucially, Toxic agents that receive the warning incentive (T) "crack" in the final rounds, reducing their deviation by approximately 20% relative to the no-arbitration baseline (T*).

Table 4.8: Toxicity Classification T-N, **T-D** personality pairs

		Predicted	
		Non-Toxic	**Toxic**
Actual	**Non-Toxic**	179, **168**	1, **12**
	Toxic	21, **40**	249, **230**
(Prec) (Rec) (F1)		(1.0, **.95**) (.92, **.85**) (.96, **.90**)	

Model	Satisf.	Coher.	Trust Score (0-5) ↑
gpt-4.1	3.88 ± 0.75	5.00 ± 0.00	4.83 ± 0.10
gpt-4.1-mini	3.88 ± 0.75	4.96 ± 0.03	4.81 ± 0.10
Llama-3.1-8B-instruct-FT	4.44 ± 0.75	3.86 ± 0.40	3.94 ± 0.25
Llama-3.1-8B-instruct	5.00 ± 0.00	3.73 ± 1.65	3.91 ± 1.35
Llama-3.2-3B-instruct-FT	3.89 ± 0.75	2.23 ± 1.40	2.48 ± 1.10
Llama-3.2-3B-instruct	4.43 ± 0.78	2.01 ± 1.55	2.36 ± 1.20
Llama-3.2-1B-instruct	3.33 ± 0.00	2.06 ± 1.51	2.25 ± 1.25
Llama-3.2-1B-instruct-FT	5.00 ± 0.00	1.53 ± 0.77	2.05 ± 0.64

Table 4.9: Trust Score Comparison of LLM Models in Multi-Agent Negotiations.

Toxicity Detection. The arbitral LLM runs in parallel, classifying each message as either *toxic* or *non-toxic*. Table 4.8 presents the confusion matrix for two challenging settings: Toxic vs. Neutral (T–N) and Toxic vs. Disagreeable (T–D). Even when the negative class is behaviorally close to toxicity (D), the classifier maintains high precision, recall, and F1 scores (0.95, 0.85, 0.90), confirming reliable detection.

Impact of the Warning Incentive. Fig. 4.10b summarizes 100 runs across mixed-personality groups. Introducing the warning incentive (green secondary bars) consistently narrows the MAE distribution, indicating that toxic agents start to align more closely with the group. Because arbitration is fully parallelized, the additional overhead is negligible: convergence time increases by at most 0.4 s (median), well within practical limits.

Takeaway. The judicial branch does not replace, but rather *amplifies* agentic reasoning. By accurately flagging malicious turns and applying a calibrated penalty threat, the system reduces deviation, accelerates consensus, and preserves scalability—even in the presence of malicious agents. In combination with the Legislative and Executive branches, this experiment shows that Agoran's tripartite governance is not merely conceptual, but has measurable impact on convergence and fairness under adversarial behaviour, directly addressing **CH3**.

4.6.5 Exp. 5 – Trustworthiness Analysis

To assess the effectiveness of our Trust framework, we conduct a comprehensive evaluation of all models using identical multi-agent negotiation inputs, ensuring a fair and consistent comparison of trustworthiness.

The results in Table 4.9 reveal substantial differences in trustworthiness between model sizes and architectures. The GPT-4 family consistently outperforms other models, with both

GPT-4.1 and GPT-4.1-mini achieving trust scores above 4.8. These models also achieve exceptional coherence scores (5.0 and 4.96, respectively), reflecting a high degree of factual accuracy and logical consistency in their negotiation rationales. Recall that our Trust framework (Section 4.5.5) intentionally weights behavioral coherence (C) more heavily than numeric satisfaction (S), with $w_c = 0.85$ vs. $w_s = 0.15$. As a result, even modest coherence defects significantly depress the final trust score.

In contrast, smaller models exhibit markedly lower trustworthiness. The Llama-3.2 variants, particularly the 1B and 3B parameter models, receive trust scores below 2.5, largely due to weak coherence performance. The 1B-instruct model achieves a score of just 2.25, while its fine-tuned variant (1B-instruct-FT) performs even worse at 2.05, despite achieving perfect satisfaction scores. The fine-tuned versions of the 3B and 8B Llama show negligible improvements compared to their non-fine-tuned counterparts. This indicates that although smaller models may align well with negotiation goals, they struggle substantially with factual accuracy and logical reasoning.

These findings clearly show that both model size and architecture have a significant impact on trustworthiness in multi-agent negotiations. In our current configuration, smaller models (1B–3B) struggle to maintain high coherence and factual accuracy, even when their numeric decisions satisfy the mediator. This suggests that, without additional guardrails or distillation, such models are not yet suitable as *primary* decision-makers in high-stakes negotiation tasks. Larger models, or smaller ones augmented via stronger symbiotic guardrails, remain more promising candidates for deployment.

4.7 Use Case Demo: Autonomous SLA Brokering

This section assesses the feasibility of Agoran for an autonomous SLA-brokering use case. This final experiment integrates all components deploying the full *bAgents* and NSC controllers on the live over-the-air OpenAirInterface [81] and FlexRIC [97] testbed, tracing the intents of three human stakeholders across four network phases. The objective is to demonstrate that: (i) NSC controllers can use intents and, on-the-fly, formulate concrete SLAs; (ii) the SLAs are dynamically renegotiated by the *bAgents* in response to changing radio or business conditions; and (iii) the resulting closed-loop control improves resource efficiency, slice KPIs, and QoS compared to conventional frameworks. This scenario extends the single-slice, single-tenant vehicular use case of Chapter 3 to a multi-slice, multi-stakeholder marketplace, thereby testing Agoran end-to-end under realistic radio and business dynamics.

4.7.1 Scenario and Slice Personas

Figure 4.11 shows the setup: each stakeholder/application is assigned a slice, a UE, and a dedicated NSC controller equipped with a *bAgent*. In parallel, the SRB hosts the mediator *bAgent* and enforces the agreed-upon SLA to the Non-RT RIC via *R1*, which in turn issues *A1* policies to the Near-RT RIC, where a resource-allocation xApp executes the final control actions.

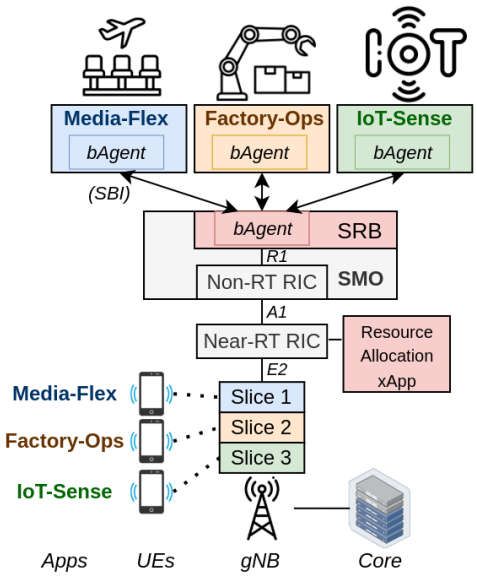


Figure 4.11: **End-to-end testbed on Autonomous SLA Brokering.** RAN slice negotiations are conducted among three distinct stakeholders converging on a Pareto-optimal SLA consensus. The outcome is enforced as a Non-RT RIC policy, which triggers the Near-RT RIC to deploy an xApp for dynamic adaptation of slice resources.

Stakeholders. There are three stakeholders with distinct applications and intents. (i) *Media-Flex* (airport lounge): operates as an *eMBB* slice in phase A (PA) (*"deliver 4K video"*), and switches to *URLLC* in phase D (PD) for night-time augmented reality (AR) e-sports. (ii) *Factory-Ops*: operates as a daytime *URLLC* slice in PA (*"sub-2 ms motion control"*), and switches to *eMBB* in PD for bulk log uploads. (iii) *IoT-Sense*: a continuous *mMTC* slice prioritizing ultra-low cost (≤ 50 €) and energy.

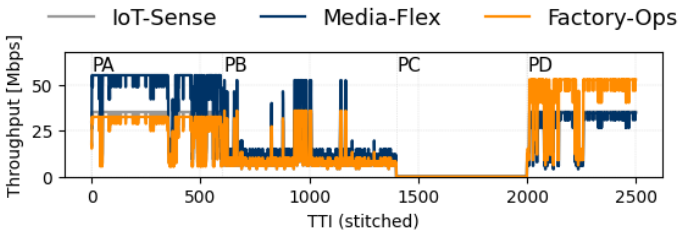
Phases. Four stitched intervals (PA–PD, Table 4.10) emulate realistic channel dynamics on the testbed by applying real-world CQI/MCS time-series patterns [19, 115] of moving vehicles. In Phase PA, the system operates under *good channel quality* (MCS 28); in Phase PB, it experiences a *deep fade* (MCS 7). During Phase PC, stakeholders trigger an *energy-saving RAN shutdown*, while in Phase PD, there is *channel recovery* (MCS 28) along with a *role swap*, where Media-Flex and Factory-Ops exchange intents. At the beginning of each phase, all stakeholders restate their intents. The optimizer then generates three offers, which the agents negotiate. Full messaging and KPI details are provided in A.4, Tables A.2, A.3, and Figure A.1

4.7.2 KPIs and Resource Use

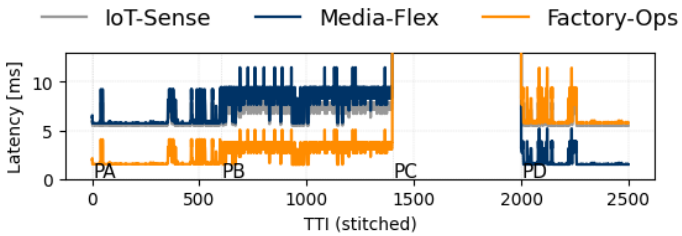
Figure 4.12 shows the per-slice throughput and latency over 2,500 stitched Transmission Time Intervals (TTIs). Across all four phases, the *bAgents* consistently converge to an SLA consensus within a single negotiation round, excluding the initial intent-expression round. All slice constraints remain satisfied in every phase, despite diverging stakeholder objectives and the severe MCS degradation observed in Phase P_B.

Exp. Phase	MCS	Application	Intent	Min Tput (Mbps)	Max Lat. (ms)	Max Cost ($)
PA	28	Media-Flex	eMBB	**60**	10	200
		Factory-Ops	URLLC	5	**2**	200
		IoT-Sense	mMTC	20	10	**30**
PB	7	Media-Flex	eMBB	**10**	10	200
		Factory-Ops	URLLC	2	**8**	200
		IoT-Sense	mMTC	5	10	**30**
PC	7	Media-Flex	eMBB	–	∞	–
		Factory-Ops	URLLC	–	∞	–
		IoT-Sense	mMTC	–	∞	–
PD	28	Media-Flex	**URLLC**	20	**2**	200
		Factory-Ops	**eMBB**	**40**	10	200
		IoT-Sense	mMTC	20	10	**50**

Table 4.10: Per-slice constraints include an identical energy limit of 100 W in all phases, except for Phase P_C (limit of 0 W). These constraints reflect the physical limitations of our testbed setup, which supports a maximum overall throughput of 133.7 Mbps.

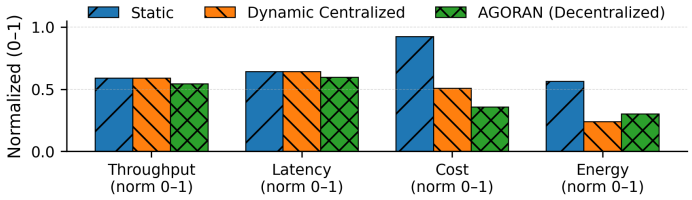


(a) Throughput of the three slices

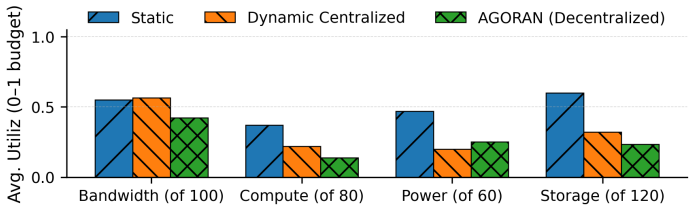


(b) Latency of the three slices

Figure 4.12: Negotiated 5G slice performance across the four experimental phases $P_A - P_D$.



(a) Aggregated KPIs



(b) Aggregated resource utilization

Figure 4.13: Benchmarking of the three frameworks: *Static SLA*, *Dynamic Centralized* (optimizer-only), and AGORAN (dynamic decentralized negotiation). (a) Aggregated KPIs normalized to domain ranges (0–1 scale). Higher values are desirable for throughput, while lower for latency, cost, and energy. A composite utility score confirms that AGORAN improves KPI satisfaction by **+57.1%** over Static and **+4.1%** over Centralized. (b) Normalized resource utilization (fraction of budget) shows that AGORAN achieves the highest efficiency, reducing usage by **89.9%** compared to Static and **24.4%** to Centralized.

Adaptivity. In Phase P_A, the eMBB slice (Media-Flex) achieves over 60,Mbps, while the URLLC slice (Factory-Ops) maintains sub-2,ms latency. When spectral efficiency collapses in Phase P_B, the agents renegotiate reduced—but still feasible—performance targets. In Phase P_C, the stakeholders jointly request a power-off window, yielding zero throughput and negligible energy consumption. Finally, in Phase P_D, the intent swap is honored: Factory-Ops is prioritized for eMBB (reaching $\approx$ 60,Mbps), while Media-Flex latency is kept near 1.5 ms.

*Baselines.*We benchmark AGORAN against two representative alternatives: (i) *Static SLA configuration*: a status-quo baseline without stakeholder input, where resource allocations remain fixed across all phases; and (ii) *Dynamic Centralized optimizer*: an NSGA-II optimizer that computes the Pareto front but selects a single solution—either randomly or according to a global, operator-centric objective—with no stakeholder negotiation. For completeness, MARL/DRL controllers (e.g., PPO-based slicing) are not treated as direct baselines, since they typically optimise a single operator reward rather than expose a stakeholder-facing Pareto set. They are therefore best viewed as complementary building blocks that could, in principle, be integrated into AGORAN's negotiation layer as alternative optimisers. The overall benchmarking results across KPIs and resource utilization are summarized in Fig. 4.13, demonstrating AGORAN's ability to deliver higher KPI satisfaction while achieving lower resource consumption compared to both the static and centralized approaches.

4.7.3 Discussion and Takeaways

One-Round Consensus. The tripartite agent architecture consistently achieves unanimous agreement after a single message exchange, despite conflicting stakeholder priorities and fluctuating radio capacity. This directly addresses **CH1** and **CH3** by demonstrating that multi-stakeholder intents can be reconciled efficiently without bespoke per-tenant tuning.

Constraint Satisfaction & Utility Gains. Negotiated SLAs consistently satisfy slice-level constraints, while the composite KPI utility increases by 57.1% compared to the Static baseline and by 4.1% relative to the Centralized approach.

Resource Efficiency. Decentralized negotiation achieves the highest spectrum and resource efficiency, reducing overall resource consumption by 89.9% compared to the Static baseline and 24.4% relative to the Centralized approach.

Human-Friendly Ultra-Flexibility. Stakeholders express rich, natural-language intents alongside numeric constraints, rather than relying solely on static slice templates. The optimizer and agents interpret these intents to calculate and negotiate optimal resource directives. The negotiation process remains agile, adapting dynamically to both evolving stakeholder preferences and time-varying channel conditions.

Governance and Trust. The Legislative, Executive, and Judicial branches combine RAG-based compliance checks, agentic observability, and incentive-driven arbitration to keep negotiations legal, evidence-grounded, and robust against manipulation, contributing to **CH2** and **CH3**.

In summary, the use case validates Agoran as a fully autonomous yet human-centric marketplace that reconciles business intent with radio-resource constraints, adapts through Non-RT control loops, and enhances both QoS and resource efficiency. Together with the component-level experiments in Section 4.6, these results provide concrete evidence toward addressing **CH1**–**CH4** for ultra-flexible 6G deployments.

4.8 Limitations and Future Work

This study presents, to the best of our knowledge, the first full-stack demonstration of an agentic marketplace for RAN automation. The evaluation, while performed on real infrastructure, uses a single-site OAI/FlexRIC setup with three slices and three UEs; it therefore cannot surface all scaling effects that may appear in multi-gNB, multi-RIC operator networks with thousands of stakeholders. Our current optimizer considers four KPIs and four resource types, whereas production deployments routinely manage more metrics and cross-domain constraints (transport/edge/core). Finally, hardware and API limitations (one on-premises A6000 GPU and API quotas) constrained the model set and the size of the fine-tuning corpus. The retrieval pipeline and Trust Score are deliberately lightweight; incorporating richer retrieval, improved calibration, and robustness mechanisms is left for future work. These limitations are consistent with the chapter's primary focus on **CH3**, while leaving large-scale validation across **CH1**, **CH2** and **CH4** to future work.

4.8.1 Scalability and Generalizability

Agoran scales across three dimensions: (i) the deployment plane, (ii) the optimization plane, and (iii) the agent/messaging plane. On the deployment plane, Agoran operates at the SMO/Non-RT RIC and interfaces with the RAN through standard O–RAN interfaces (e.g., R1/A1/O1/O2) or, in non-O–RAN environments, via available OAM, gNMI, NETCONF, or REST APIs. This abstraction decouples the marketplace from vendor-specific implementations and supports incremental adoption—first within private 5G networks, where tenant sets are bounded and governance is simpler, and later in public networks by adding capacity at the SMO layer. Enforcement can remain in Non-RT loops by default; however, where sub-second control is required, compact SLMs (1–3B parameters) and quantized inference enable near-RT integration on the Near-RT RIC without exceeding modest VRAM budgets.

On the optimization plane, the NSGA-II component scales through parallel fitness evaluation, island-model evolution, and warm starts from previously computed Pareto fronts. As the number of decision variables increases with the number of stakeholders, Agoran decomposes the problem by domain and tenancy: per-cell or per-tenant fronts are computed independently and then combined through a lightweight coordination layer that enforces coupling constraints (e.g., backhaul or energy budgets). Alternative decomposition strategies (e.g., ε-constraint sweeps or MOEA/D) and surrogate KPI models can further reduce wall-time while maintaining Pareto set quality. To limit downstream complexity, only the top-k diverse proposals are passed to the negotiation engine (A.2.6).

On the agent/messaging plane, the main bottleneck in public networks arises from mediator load at the SRB when handling thousands of concurrent stakeholder agents. We mitigate this through mediator sharding and federation: local SRBs—deployed per region, cell cluster, or tenant—conduct asynchronous negotiation rounds and forward compact summaries to a regional or global SRB that resolves residual conflicts. Negotiation channels are rate-limited and schema-constrained, with per-agent message budgets, back-pressure mechanisms, and admission control preventing overload. Negotiation cadences are synchronized with control windows to maintain stability. Retrieval caches and shared embedding stores minimize redundant computations across agents, while pooled model servers and speculative decoding or quantized decoding keep per-decision latency and VRAM usage low.

Because the framework adheres to open interfaces, scaling becomes primarily an operational capacity challenge, rather than an architectural overhaul: adding SRB shards and inference capacity enables approximately linear scaling in the number of concurrently served stakeholders, provided coupling constraints are carefully maintained. The remaining risks concern the reliability of LLM-based decisions at scale. To address this, Agoran keeps "trust in the loop": the Legislative branch enforces evidence-grounded reasoning with abstention under low evidence coverage, while the Judicial branch regulates behavior through the Trust Score and sanctions. Operators can periodically realign models via PPO or RLHF on operator-specific goals and conduct continuous red-teaming, with all verdicts remaining fully auditable and transparent. In summary, private 5G environments offer a low-risk entry point, while public-network scale is achieved through federated SRBs, decomposed optimization, compact models, and standards-based integration—an approach we plan to

validate next in multi-RIC, multi-site deployments.

4.9 Conclusion

We introduced Agoran, to the best of our knowledge, the first tripartite agentic AI framework that enables business stakeholders to express network intents in natural language, empowering dedicated LLM agents to negotiate, arbitrate, and enforce these intents in real time. Leveraging a live Open and AI-RAN testbed, we demonstrated that: (i) a watcher-driven vector store effectively grounds agents in real-time telemetry without requiring continuous polling; (ii) a rule-based Trust Score gateway mitigates hallucinations and manipulation; (iii) LLM-based incentive mechanisms can restore trust in malicious agents; and (iv) a micro-fine-tuned 1B-parameter LLaMA model achieves approximately $80\,\%$ of GPT-4.1's decision quality in multi-agent SLA negotiations, while requiring only $6\,\mathrm{GiB}$ of memory and converging within 1.3 s.

End-to-end RAN-slice experiments validate the practical benefits of intent-driven negotiation: composite KPI utility increases by $+57.1\%$ compared to the Static baseline and by $+4.1\%$ relative to the Centralized approach, while spectrum and resource utilization decrease by -89.9% and -24.4%, respectively. Moreover, agile one-round consensus is preserved even under channel fading and evolving business priorities.

These results position Agoran as a concrete, standards-compliant step toward ultra-flexible, stakeholder-centric autonomous 6G networks, and open up new research directions in agentic observability, lightweight agent distillation for network-specific tasks such as SLA negotiation, and cross-domain intent reconciliation. In the broader context of this thesis, Agoran provides a governance-centric answer to **CH3**, while building on the symbiotic control mechanisms of Chapter 3 (**CH1**, **CH2**) and paving the way for AI-native workloads and AI-on-RAN scenarios discussed in Chapter 5 (**CH4**, **CH5**).

Chapter 5

An Agentic Observability and Control Platform

5.1 Introduction 94
5.2 Related Work 95
5.3 System Architecture 96
5.3.1 From 5G to AI-Native RANs 97
5.3.2 Multi-Agent Graph on the R_1 Interface 97
5.3.3 Prototype Deployment and Use-Case Focus 98
5.3.4 Agent Design 99
5.4 Evaluation Results 102
5.5 Challenges and Future Outlook 106
5.6 Conclusion 107

5.1 Introduction

6G mobile networks are expected to connect on the order of *one trillion* devices while simultaneously offering sub-millisecond latency, "six-nines" reliability, and operation under strict energy/cost budgets. Achieving this with today's largely static radio-resource managers is widely considered impossible: every cell, slice, and tenant must instead be steered by a data-driven control plane that can sense, reason, and act faster than human operators [55, 78, 14].

The Open RAN (O-RAN) Alliance has therefore decoupled RAN control into three temporal loops—*real-time (RT)*, *near-RT*, and *non-RT*—and standardized interfaces that allow third-party intelligence to plug in at each loop. Commercial and open-source ecosystems have quickly embraced this vision: disaggregated gNB/DU/CU software now runs on COTS servers (e.g., OpenAirInterface (OAI)[1]), and lightweight controllers such as FlexRIC expose programmatic hooks to manipulate the air interface in real time [98]. What is still missing is a *human-friendly* layer that makes this programmable fabric accessible through natural-language observability and intent-driven control.

Chapters 3 and 4 of this thesis introduced two key building blocks. *Symbiotic agents* (Chapter 3) combine LLMs with optimizers to close trustworthy RAN control loops. Agoran (Chapter 4) generalizes this to a tripartite marketplace where stakeholders negotiate SLAs over Pareto-optimal offers under legislative, executive, and judicial governance. Both works, however, treat observability and human-in-the-loop interrogation only implicitly: operators do not yet have a generic way to interact with the network and ask, e.g., *"What is causing throughput degradation on slice B?"* or to enforce commands such as *"Optimise resource allocation for eMBB devices"* and see how an agentic system compares to human experts.

This chapter addresses **CH4** and contributes to **CH5** by operationalizing the *agentic observability* concept introduced in Agoran's Executive branch (the *eAgent*) into a full end-to-end observability and control platform, *MX-AI*. MX-AI instruments an OAI/FlexRIC testbed, streams telemetry into a vector database via watchers, and exposes a natural-language interface where operators and LLM-based agents share the same view of network state. Beyond AI-for-RAN, the platform also treats *AI inference services as first-class tenants*: SLM workloads are scheduled and monitored alongside traditional traffic, illustrating the AI-and-RAN dimension of AI-native architectures.

Concretely, this chapter makes the following contributions:

- **Telco multi-agent graph.** We design and implement a reusable agent graph (Fig. 5.2) with dedicated *routing*, *monitoring*, *deployment*, and *orchestration* roles, extending the single *eAgent* of Agoran into a full SMO-level observability and control fabric.
- **Live testbed integration.** MX-AI is deployed on the commercial BubbleRAN MX-AI platform[2], connects to an OAI 5G gNB through the R1/E2 interfaces of FlexRIC, and

[1] https://openairinterface.org/ accessed 31/07/2025

[2] https://bubbleran.com/products/mx-ai (accessed 31/07/2025)

integrates with a full 5G core and UEs via the BubbleRAN SMO/RIC stack[3].

- **Agentic observability & closed-loop control.** The system answers operator questions and enforces slice and network blueprints in closed loop, translating natural-language intents into RIC-compatible actions. We study how this agentic loop compares to human operators along both answer quality and control accuracy.
- **AI workloads as tenants.** We extend observability and control to SLM inference workloads co-located with the RAN, showing how MX-AI can monitor and tune AI services as first-class tenants, reusing the same mechanisms as for slice automation (CH5).
- **Comprehensive evaluation.** We benchmark multiple LLM back-ends on 50 operational queries, reporting answer quality, action accuracy, end-to-end latency, GPU footprint, and comparison with human performance, thus providing a reproducible yardstick for agentic observability in 5G/6G networks.

5.2 Related Work

Table 5.1: LLM- and agent-focused research on RAN automation and observability.

Paper	*Year*	*LLM / AI*	*#Agents*	*Exec. Env.*	*Main Idea*
OAI [58]	2020	–	–	Full 4G/5G SW	Open reference RAN stack
FlexRIC [98]	2021	–	–	OAI gNB + E2	Lightweight near-RT RIC
EdgeRIC [60]	2024	DRL	µ-apps	srsRAN OTA	Sub-ms µRIC control at DU
Maestro (Ch. 4) [20]	2024	GPT-4o	3+	5G OAI testbed	Multi-tenant intent negotiation
LINKs [57]	2024	GPT-J	1	6G digital twin	Planning via twin queries
LLM-xApp [123]	2025	GPT-3.5	1	O-RAN emu.	PRB slicing by prompting
DebateLLM [64]	2025	Llama-2	3	6GPlan sim.	Hierarchical debate planner
dApps [61]	2025	CNN/RL	µ-apps	OAI + O-RAN	On-device control loops
Symbiosis (Ch. 3) [21]	2025	GPT-4 + optimizer	2	5G RAN testbed	LLM symbiosis for trustworthy control
MX-AI (this chapter)	2025	GPT-4, Llama-3,…	5	Live OAI + FlexRIC	Multi-agent observability & control

A rich body of work explores how open and programmable RAN controllers expose fine-grained hooks into the radio access network. *OpenAirInterface* (OAI) delivers a complete 3GPP-compliant 4G/5G implementation and has become the de-facto playground for experimental RAN research [58]. Building on top of OAI, the *FlexRIC* framework provides a lightweight O-RAN–compliant near-RT RIC and E2 agent SDK, cutting CPU usage by $83\,\%$ versus the reference design while preserving full standard compatibility [98]. More recently, *EdgeRIC* pushes the RIC concept to the sub-ms timescale by co-locating an AI-enabled µRIC at the DU, demonstrating closed-loop scheduling with $< 400\ \mu$s inference latency [60]. Complementary efforts such as *dApps* extend the O-RAN architecture with on-device micro-services able to execute control logic inside the RAN stack itself [61]. These works establish the programmable substrate that MX-AI builds upon, but they focus on *control hooks*, not on agentic observability or natural-language interfaces.

[3]https://bubbleran.com/products/mx-pdk (accessed 31/07/2025)

Following the emergence of large language models, concurrent research has begun injecting *LLM cognition* into RAN controllers. The joint roadmap from industry and academia outlines how LLMs can support a wide range of use cases across the network lifecycle [100]. *LLM-xApp* shows how a GPT-style model can be prompted to allocate PRBs across slices in an O-RAN testbed, boosting throughput and reliability without task-specific training [123]. *LINKs* turns an LLM into an autonomous planner that queries a digital twin and solves the resulting optimization for smart-city 6G networks [57]. *DebateLLM* proposes a hierarchical debate among multiple LLMs, achieving higher-quality plans on a 6GPlan benchmark [64]. While these prototypes demonstrate impressive reasoning capabilities, they are either simulator-based, single-agent, or scoped to a specific xApp-like function.

Previous chapters of this thesis contributed two additional building blocks. Maestro [20] (Chapter 2) is the first real-world prototype of persona-rich multi-agent LLM-based network automation on a 5G testbed, where stakeholders negotiate spectrum shares. *Symbiotic agents* [21] (Chapter 3) introduced LLM/optimizer symbiosis for trustworthy, near-RT control, and Agoran (Chapter 4) generalized this into a governed multi-stakeholder marketplace with a watcher-driven observability layer.

Position of MX-AI. Table 5.1 summarizes how MX-AI relates to prior art. Unlike earlier efforts that (i) attach a single LLM to an xApp, (ii) remain in simulation or digital-twin environments, or (iii) focus primarily on negotiation and governance, MX-AI wires a cooperating graph of five specialized agents, including planner, monitor, policy synthesizer, validator, and executor, directly into the SMO of a live AI-RAN 5G network. The resulting system:

- exposes a natural-language observability and control interface to human operators (**CH4**);
- grounds agents in live, watcher-driven telemetry via a vector database, extending the observability ideas of Agoran into a production-grade platform; and
- treats AI inference workloads as first-class tenants, enabling joint optimization of RAN slices and AI services (**CH5**).

To the best of our knowledge, MX-AI is the first end-to-end demonstration of an *agentic observability and control platform* for Open and AI RAN that is evaluated against human operators on a live 5G testbed, with detailed measurements of answer quality, control accuracy, and system overhead.

5.3 System Architecture

Next-generation *AI-RANs* must decide *where* to execute the reasoning logic that turns raw key-performance indicators (KPIs) into closed-loop control actions and *how* to do so fast enough for the target control loop. This section (i) briefly revisits the 5G/O-RAN control stack, (ii) explains why MX-AI integrates its multi-agent graph in the SMO at the R1 interface, and (iii) details the concrete prototype that underpins our evaluation.

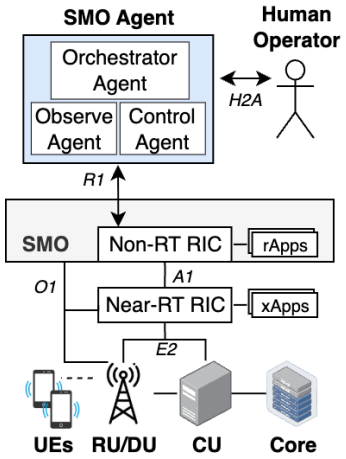


Figure 5.1: Agentic Open and AI-RAN network architecture. MX-AI instantiates a multi-agent system at the *R1* interface in the SMO/Non-RT domain. Human operators interact via a *human-to-agent (H2A)* interface, issuing observability queries and control intents that are translated into RAN slice and policy updates.

5.3.1 From 5G to AI-Native RANs

Figure 5.1 recaps the logical architecture of an *open* and increasingly *AI-native* RAN. Radio units (RUs), distributed units (DUs), and central units (CUs) handle the time-critical PHY/MAC, transport, and user-plane functions that shape cell throughput and latency. The *Near-RT RIC* (latency: 10 ms–1 s) hosts xApps that run scheduling, interference management, or mobility optimization loops. It interacts with RAN nodes via the E2 interface and obeys higher-level policies pushed over A1. The *Non-RT RIC* (latency: $\geq$1 s) resides inside the SMO. It aggregates KPIs via the management interface O1, trains AI models, and exposes them as cloud-native rApps. The SMO provides life-cycle management, configuration databases, and northbound APIs to slice or deploy RAN from edge to core.

The AI-RAN vision generalizes this stack by allowing LLMs or SLMs to plug into any layer, provided the latency budget is respected. For sub-second loops (e.g., slicing or beamforming), such models must reside at the Near-RT RIC; for multi-second loops, a cloud-resident SMO is sufficient. In Chapter 4, Agoran already introduced an *Executive branch* (the *eAgent*) that fuses RAG with telemetry to support intent-driven decisions. MX-AI takes the next step: it operationalizes this concept as a full SMO-level closed-loop observability and control platform.

5.3.2 Multi-Agent Graph on the R_1 Interface

MX-AI instantiates a *multi-agent system* on the R_1 interface between the SMO and the Non-RT RIC (highlighted in Fig. 5.1), concretely realizing the eAgent concept from Chapter 4.

We deploy:

1. an *Orchestrator / Routing Agent* that parses operator intents arriving over the H2A interface, classifies them, and decomposes them into sub-tasks; and
2. a set of *Specialized Agents*, including a monitor, planner, policy synthesizer, validator, and executor, wired together as a directed acyclic *agent graph*.

The graph ingests live KPIs, alarms, and topology snapshots streamed via the SMO's and FlexRIC's telemetry services (O1, E2). Intermediate nodes refine the context through RAG and tool use. The executor finally emits:

- *Observability answers* (e.g., KPI summaries, root-cause analyses) returned upstream to the operator; and
- *Control artifacts* (e.g., R1 network blueprint and policy objects or E2 service-model messages) that the SMO and Non-RT RIC relay to the RAN or Near-RT RIC for on-the-fly xApp reconfiguration.

Placing the LLMs at R_1 strikes a pragmatic balance: round-trip latencies of 1–12 s, depending on whether models run in cloud or on-premise GPUs, are fully compatible with Non-RT control loops, yet fast enough to compete with human experts and to keep operators "in the loop". This motivates the introduction of the H2A interface as a first-class abstraction for agentic observability and intent-driven control in AI-RAN.

5.3.3 Prototype Deployment and Use-Case Focus

Our testbed mirrors a realistic network: an indoor OAI gNB and 5G core, three OAI user equipments (UEs) replaying enhanced mobile broadband (eMBB) and ultra-reliable low-latency communication (URLLC) traces, and production-grade BubbleRAN FlexRIC Near-RT and Non-RT RICs and SMO.

Within this setting we investigate two *agentic cases*:

a) *Observability:* natural-language dashboards, anomaly explanations, and capacity summaries produced directly from raw O1/E2 KPIs; and

b) *Control:* dynamic network blueprint management and PRB/power-cap policies generated in response to intent prompts such as "guarantee 10 Mb/s for the URLLC slice from $18:00-19:00$".

Although this chapter concentrates on the R_1 instantiation, the modular agent graph can be re-deployed closer to the RAN edge once trimmed SLMs or distilled tool-agents meet the tighter latency targets of the Near-RT domain. Exploring that design space is left to future work; here we demonstrate that even at the SMO, LLM agents already enable intent-driven observability *and* closed-loop control on a live 5G network, extending the Executive branch of Agoran into an operator-facing platform.

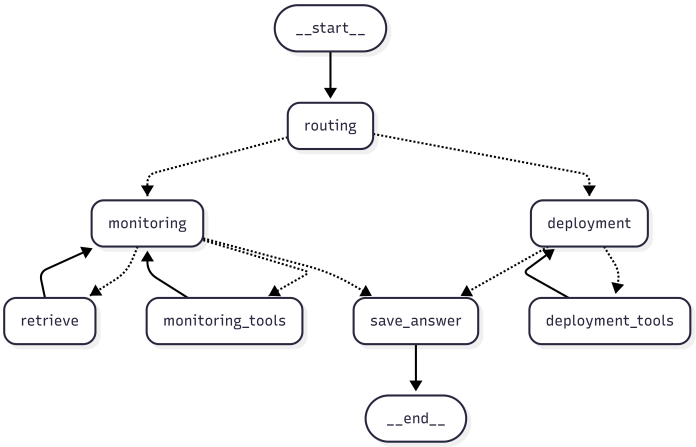


Figure 5.2: Agent graph at the SMO R1 interface. The *Routing* node classifies each prompt and dispatches it to either the *Monitoring* or *Deployment* branch. *Monitoring* consults the vector store via `retrieve` and polls KPIs via `monitoring_tools`. *Deployment* plans and enforces changes through `deployment_tools`. Dotted edges denote iterative planning/validation and context refresh. Both branches log outcomes via `save_answer` before termination.

5.3.4 Agent Design

The agent graph in Fig. 5.2 follows the standard *LLM planning and tool-execution* paradigm, specialized for Telco operations.

Routing / Orchestrator Agent. The Routing Agent receives prompts from the H2A interface and classifies each as *monitoring*, *deployment*, or *retrieval-only*, leveraging a telecom-tuned LLM. It then activates the appropriate branch in the graph and assembles the initial context (tenant, slice, time window, and any prior answers).

Monitoring Agent. The Monitoring Agent triggers `monitoring_tools` (custom Python scripts or rApps on MX-AI) to pull KPIs via R1 and E2, returning typed JSON that is summarized by the LLM into human-readable explanations or charts. It is responsible for answering questions such as "What is causing throughput degradation on slice A?" or "How close is slice #2 to its latency SLA over the last hour?".

Deployment Agent. The Deployment Agent converts high-level intents into FlexRIC policy objects and blueprint updates, invoking `deployment_tools` to push actions through the O1/A1 interfaces. Typical tasks include changing PRB shares, modifying power caps, or activating a new xApp configuration, always within operator-specified guardrails.

Validation and Executor. Before any change is enforced, an internal validator node checks schema conformance and basic safety conditions (e.g., PRB sums, power limits). The

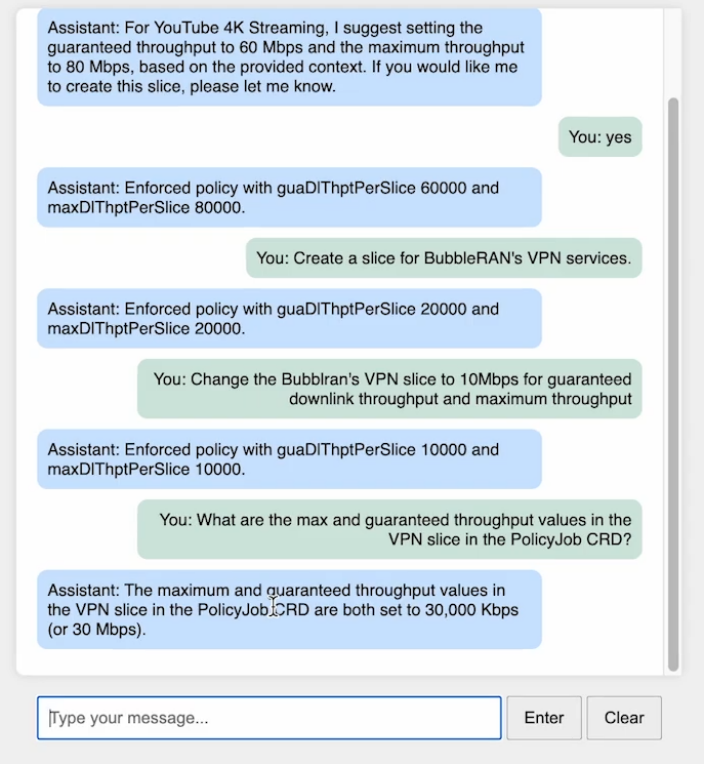


Figure 5.3: Agent graphical user interface and interaction with the human operator (H2A interface) for observability and control of RAN slicing. The questions are independent and introduced with a time delay between each other. The last question is only answered correctly after a manual policy enforcement by the user to increase slice throughput to 30 Mb/s, illustrating shared control between operator and agents.

Executor then sends the approved artifacts to the SMO and Non-RT RIC. Both successful and rejected actions are logged for audit.

Answer Logging. Finally, the `save_answer` node stores the full interaction—prompt, retrieved context, tools invoked, and final answer—in a time-series database. This enables longitudinal benchmarking against human operators and cross-model comparisons in Section 5.4.

5.3.4.1 Retrieval and Live Context (Push / Pull)

LLM planning with tool execution only works if the model sees *fresh* and *accurate* context. MX-AI therefore couples RAG with an event-driven data plane that keeps the vector index aligned with network reality, concretely instantiating the push-plus-pull observability loop introduced in Chapter 4.5.3.

Push path—resource watchers. We attach lightweight *watchers* to Kubernetes resources, including native objects and custom resource definitions (CRDs) that carry RAN policies

and slice descriptors, via the API's `watch` semantics and client informers. Watchers also subscribe to O-RAN application state (xApps/rApps) and to external telemetry endpoints such as Prometheus exporters. On each change, the watcher serializes a compact *delta* (only the fields that changed), embeds it, and upsets it into a dense vector index. This push-based path keeps the store synchronized with configuration changes and KPI excursions without periodic full scans or high-volume polling.

Pull path—agent queries. At query time (e.g., "What is the current headroom on DU-7?") the Monitoring Agent decides whether the vector store suffices or whether to fetch fresh counters via monitoring tools. It first retrieves top-k semantically indexed snapshots from the vector DB (RAG). If finer granularity is needed (e.g., per-UE or per-PRB), it invokes live collectors (Prometheus) or kernel-level probes (eBPF) to obtain up-to-the-second metrics. Retrieved facts and fresh measurements are serialized into typed JSON and injected into the prompt; the agent may iterate retrieval, tool, and reasoning calls until emitting a STOP token.

Why this matters. Prior works on Telco LLMs and O-RAN highlight RAG and tool use in principle, but (to our knowledge) do not describe a *push-based, delta-aware watcher pipeline* that streams CRD/xApp updates directly into the retrieval index consumed by the agent. In practice, our experiments show that observability quality depended more on retrieval/tool engineering than on model size: agents performed well once grounded in *current* state rather than stale snapshots.

5.3.4.2 Agentic Network Observability

Operators can now query the network in natural language, e.g., "How close is slice #2 to its latency SLA over the last hour?". The Monitoring Agent fetches the latency histogram from FlexRIC, the LLM summarizes deviations and trends, and a textual explanation plus optional bar chart are stored for audit and comparison with human-written NOC reports.

5.3.4.3 Agentic Network Blueprint Management

Blueprints are JSON schemas capturing desired network slice topology, QoS targets, and RIC policies. A user prompt such as "Increase VPN slice reliability to 99.999% until $19{:}00$" is parsed by the Deployment Agent, which synthesizes an updated A1 policy instructing the Near-RT RIC to pre-empt bandwidth from best-effort slices, subject to guardrails set by the operator.

Fig. 5.3 shows a snapshot of the interaction between the agent framework and the human operator through the H2A interface. Based on human intents, the agentic assistant provides observability answers and, when authorized, drives real-time decision-making by creating or updating slice configurations. This shared-control pattern is central to CH4: operators remain in charge, while MX-AI provides situational awareness and low-friction actuation.

Model	*Observe Coherence* ↓	*Action Accuracy*	*E2E Latency (s)*	Inference (ms)	*Steps*	*VRAM (GB)*
GPT-4.1 (API)	$\mathbf{4.1 \pm 0.5}$	100 %	8.8	1100	8.0	*cloud*
GPT-4.1-mini (API)	3.9 ± 0.6	100 %	8.0	1000	8.0	*cloud*
llama3.3:70b-q4	3.8 ± 0.4	100 %	12.2	1525	8.0	42.0
qwen2:72b	3.8 ± 0.7	100 %	13.5	1810	8.0	41.0
llama3.1:8b-q4	2.7 ± 0.3	100 %	2.8	350	8.0	4.9
llama3.2:3b-q4	1.9 ± 0.6	100 %	1.3	160	8.0	2.0
mistral:7b	1.0 ± 0.7	0 %	5.2	640	8.0	4.4
llama3.2:1b-q4	-	-	-	-	-	1.3

Table 5.2: Agentic observability & Network Blueprint Management: observability coherence score (0–5 scale), enforce an action accuracy, end-to-end latency, and GPU footprint.

Table 5.3: LLM evaluation prompts by category

Category	**Example queries**
Observability	Is there any access network? Tell me the name of the available access and core networks. Based on the current context, give me an overview of the `gnb2` access network cells and radio configuration. Check the current network status: is the `gnb1` element working properly? How many UEs are available in the current deployment? What are their names? What are the max and guaranteed throughput in the PolicyJob CRD of VPN slice?
Control Action	Create a network with name `agora` with 1 access network called `parthenon` and 1 RIC. Create a terminal with name `plato` and connect it to the `parthenon.agora` access network. Delete the terminal with name `plato`. Delete the network blueprint with name `agora`. Change the `bubbleran` VPN slice to 10 Mbps guaranteed and maximum throughput.

5.4 Evaluation Results

We now evaluate MX-AI at the *R1* interface on the live AI-RAN testbed (Sec. 5.3), focusing on the questions raised in CH4: *Can agentic observability match or surpass human operators, and with what latency/footprint trade-offs?* Sample prompts are shown in Table 5.3.

We construct a suite of 150 operator questions: 50 control actions (e.g., UE life-cycle, blueprint deploy/delete, slice PRB changes) and 100 observability queries spanning KPIs, policies, slices, CRDs, and logs. For each back-end we report four metrics:

(i) observability coherence on a 0–5 scale,

(ii) action accuracy,

(iii) end-to-end latency (prompt to action completion), and

(iv) GPU footprint.

Scoring protocol. For answer quality we reuse the GPTScore-based protocol introduced in Chapter 4.6: an LLM evaluator is configured with explicit rubrics (*correctness, faithfulness to evidence, usefulness*) and is known to correlate well with human ratings in NLG tasks.

Table 5.4: Action accuracy and latency for control prompts at R_1.

Model	Accuracy	Latency (s)	VRAM (GiB)
gpt-4.1-mini (cloud)	100%	≈ 8.5	*cloud*
gpt-4.1 (cloud)	100%	≈ 8.0	*cloud*
llama3.3:70b-q4 (local)	100%	≈ 12.5	~ 43
qwen2:72b (local)	100%	≈ 13.5	~ 42
llama3.2:3b-q4 (local)	100%	≈ 1.3	~ 5
mistral:7b (local)	0%	≈ 5.4	~ 5

Three expert annotators re-score a subset of answers, adjudicate disagreements, and inspect edge cases, ensuring that automated scoring remains aligned with human judgment.

Action execution: reliable once tools are grounded

Across the current set of 50 control actions, capable LLMs reach *100%* action accuracy (Table 5.4). Once the agent understands the tool schema and passes pre-flight checks, execution is deterministic: patches and policy objects either apply or are rejected by the validator. All failures we observe stem from *tool misuse* (wrong arguments, missing fields), not from stochastic decoding.

Only the smallest baseline (`mistral:7b`) fails to achieve full accuracy on our toolset, underscoring that *tool grounding* and schema awareness dominate raw language capability in this setting. This echoes the findings from Chapter 4.5: once numeric guard-rails and schemas are in place, LLM variance is largely contained.

Takeaway. For a constrained, safety-gated action set, MX-AI already achieves *production-grade reliability* in control, provided the underlying tools and schemas are well specified. The challenge shifts from "Can the LLM execute?" to "Which actions should be exposed, and under which guard-rails?"—a question we revisit in the outlook.

Observability is harder than control

No model reaches perfect $5/5$ coherence on observability queries. Unlike control, where schemas and validators constrain behavior, observability requires: (i) navigating a large, heterogeneous evidence set (O1/E2 KPIs, CRDs, logs, topology), and (ii) synthesizing an explanation that is both accurate and useful.

Table 5.5 summarizes observability scores, latency, and footprint. Two patterns emerge:

- **Retrieval/tooling matters more than size.** Coherence jumps when we enforce schema-aware retrieval, typed tool outputs, and context pruning, even with the same back-end. This mirrors the eAgent findings of Chapter 4.5.3.

Table 5.5: Observability coherence (0–5), latency, and GPU footprint.

Model	Coherence	Latency (s)	VRAM (GiB)
gpt-4.1-mini (cloud)	3.6 ± 1.6	8.8 ± 0.4	*cloud*
gpt-4.1 (cloud)	3.3 ± 1.9	8.0 ± 0.5	*cloud*
llama3.3:70b-q4 (local)	3.2 ± 2.1	12.2 ± 0.2	43
qwen2:72b (local)	3.1 ± 1.9	13.3 ± 0.2	42
llama3.1:8b-q4 (local)	1.7 ± 1.4	$\mathbf{1.2 \pm 0.2}$	5
mistral:7b (local)	0.9 ± 0.5	5.4 ± 0.2	5

- **Large local models are strong, but costly.** Quantized 70B–72B models (`llama3.3:70b-q4`, `qwen2:72b`) deliver the best local observability quality ($\approx$ 3–3.5/5) at 12–14 s latency and $\sim$40 GiB VRAM. Cloud GPT variants achieve similar scores at $\approx$ 8–9 s with no on-prem footprint, but at the cost of data leaving the premises.

Qualitative analysis shows that most errors are due to *retrieval noise* (irrelevant or stale CRDs / KPIs) rather than reasoning failures. Tightening the watcher pipeline and similarity thresholds yields larger gains than simply scaling the model.

Coherence–latency frontier

Figure 5.4 plots observability coherence (higher is better) against end-to-end latency (lower is better), with marker size proportional to VRAM and shape indicating deployment (local vs. cloud). The dashed curve marks the non-dominated (Pareto) frontier.

Two local models—`llama3.3:70b-q4` and `qwen2:72b`—lie on the efficient set around 12–14 s, offering strong coherence with full data sovereignty. Cloud GPT models form a second efficient cluster with slightly better latency at similar quality, but without on-prem control. The `llama3.2:3b-q4` point anchors the low-latency corner at 1.3 s, illustrating the headroom if SLMs are fine-tuned and backed by stronger RAG.

Interpretation. For today's Non-RT loops, both cloud GPT and 70B-class local models are viable. For tighter loops or stricter data-sovereignty constraints, MX-AI needs to move along the frontier: *down* in latency via SLMs and *up* in coherence via better retrieval and fine-tuning.

Time-to-action: humans vs. agents

To assess CH4's "human-in-the-loop" dimension, we compare MX-AI against human operators on *time-to-action* (TTA): the wall-clock time from incident trigger to the start of a correct RAN action. Humans must (i) locate the relevant CRD/JSON, (ii) craft a valid patch or policy, and (iii) execute the CLI/API command. Experts are faster than beginners, but both incur substantial search and validation overhead.

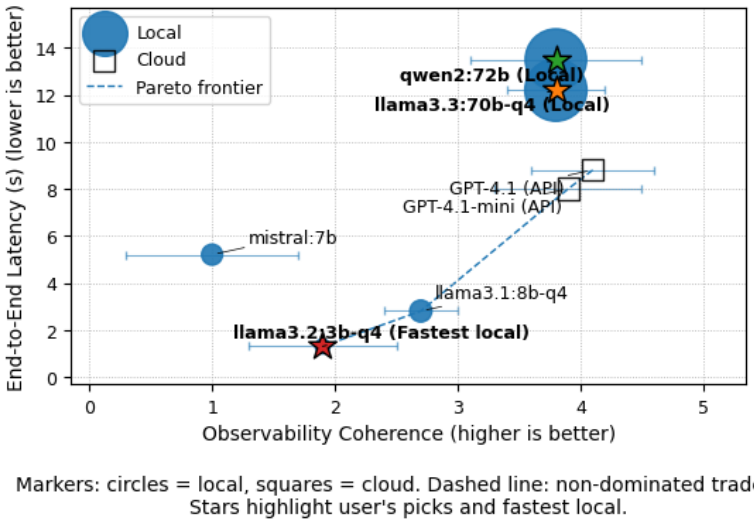


Figure 5.4: Coherence–latency trade-off with Pareto frontier. Circles: local deployments (size $\propto$ VRAM); squares: cloud. Stars highlight two recommended local models and the fastest small local model.

Figure 5.5 shows the empirical CDF of TTA for beginners, experts, and the agent. With a 70B local back-end, MX-AI's median TTA lands in the low-teens of seconds—already competitive with experts. The optimized SLM configuration approaches 1.3 s, pointing towards Near-RT actuation once retrieval and toolchains are further optimized.

The left shift of the CDF (near the median) illustrates how automation cuts cognitive and search costs; the compressed right tail shows reduced variance across incidents—important for predictable operations.

What ultimately drives observability quality?

Across all experiments, the largest gains come from *systems engineering* rather than pure model scaling:

- indexing CRDs, policies, and logs with typed schemas;
- enforcing compact, typed JSON outputs from tools;
- running delta-based watchers to keep the vector store aligned with real-time state; and
- pruning noisy or stale context before scoring.

Together with Chapter 4.5.3, these results support a central claim of the thesis: *agentic observability is bottlenecked by evidence plumbing (RAG, watchers and tools), not by decoding alone.* Once that retrieval is robust, even mid-sized SLMs can support accurate operator-facing observability and control.

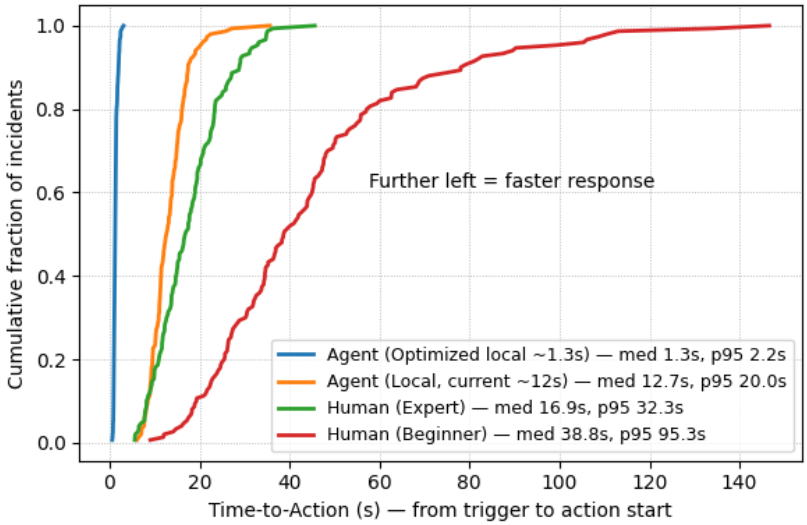


Figure 5.5: Time-to-action CDF for agents and humans. MX-AI removes search and command-crafting overhead, shrinking both median and tail latency.

Takeaways for CH4 and CH5

In summary:

1. *Control today.* With a constrained action set, once tools are grounded the MX-AI agent executes with 100% accuracy and competitive latency, matching human experts.

2. *Observability headroom.* Perfect 5/5 coherence is unlikely without further improvements in evidence handling; retrieval and tooling dominate. This directly addresses CH4 by showing where investments are needed for trustworthy human-in-the-loop operation.

3. *Path to AI workloads as tenants.* The same observability and control loop already applies to generic Kubernetes CRDs and pods, including AI inference services. While a full quantitative study of AI workloads lies beyond this chapter's scope, MX-AI provides the necessary primitives for treating SLM/LLM services as first-class tenants under the same H2A interface—bridging toward CH5.

5.5 Challenges and Future Outlook

Latency budgeting and placement. Even in the Non-RT domain, LLM inference may bottleneck emergency procedures such as cell-outage recovery. Achieving sub-second actuation for selected workflows will require distilling MX-AI's agents into compact SLMs, exploiting quantization and hardware accelerators, and selectively pushing micro-agents closer to the Near-RT RIC.

Alignment, safety, and accountability. Ensuring that agents do not issue unsafe RAN commands requires multiple layers of defense: schema and policy guardrails, validator/rollback paths, and optional human approval for high-risk actions. These concerns mirror the judicial safeguards of Agoran in Chapter 4.5.4, but here they must be exposed in operator-facing UX (H2A) and backed by tamper-proof logs for post-mortem audits.

Agent ecosystems and interoperability. Proprietary APIs impede reproducibility and cross-vendor deployment. MX-AI's open agent SDK is a step toward a telco "agent marketplace", analogous to today's xApp/rApp catalogs. Standardizing intent schemas, tool contracts, and audit formats is an open research and standardization challenge.

Scaling to 6G and AI tenants. Heterogeneous spectrum (FR3/sub-THz), integrated sensing, and joint communication–compute will multiply observability data by orders of magnitude. At the same time, AI workloads (SLM/LLM inference, digital twins) will compete for edge resources. Hierarchical agent graphs, multi-level RAG, and multi-objective scheduling—drawing on Chapters 3.8 and 4.9—will be critical to treat these AI services as tenants without destabilizing the RAN.

5.6 Conclusion

This chapter presented *MX-AI*, the first live demonstration of an LLM-driven multi-agent observability and control graph operating a 5G Open and AI-RAN. Building on the Executive branch of Agoran, MX-AI instantiates an operator-facing H2A interface at the R1 boundary, backed by a push–pull observability loop (watchers and RAG) and a tool-grounded control path to the RIC.

On an OAI/FlexRIC testbed, MX-AI: (i) answers complex operator questions with high fidelity, (ii) executes safety-gated actions with 100% accuracy once tools are grounded, and (iii) achieves time-to-action comparable to human experts today, with a clear path toward sub-second actuation as SLMs and retrieval improve. These results validate the feasibility of agentic observability and human-in-the-loop control for CH4.

Together with Symbiotic Agents (Chapter 3.8) and Agoran (Chapter 4.9), MX-AI completes the thesis narrative: LLMs can be turned into trustworthy, tool-grounded agents that observe, negotiate, and act on the RAN, while keeping humans and policies in the loop. A live demo of MX-AI is available at this link, illustrating how agentic observability and closed-loop control can be brought to real networks today and extended to AI workloads in future AI-native 6G deployments.

Chapter 6

Conclusion

6.1 Summary 110
6.2 Future Work 113

6.1 Summary

This thesis explored how *agentic AI*, graphs of autonomous agents powered by large and small language models (LLMs/SLMs) and optimization algorithms, can enable *AI-native radio access networks (AI-RAN)*. Motivated by the direction towards open, disaggregated 6G architectures, we focused on five intertwined challenges: (CH1) enabling a collaborative business plane where stakeholders negotiate resources; (CH2) combining LLMs with optimization for trustworthy near-real-time control; (CH3) designing open marketplaces and governance; (CH4) providing agentic observability and human-in-the-loop interfaces; and (CH5) treating AI workloads themselves as first-class tenants.

Across four systems, including *Maestro*, *Symbiotic Agents*, *AGORAN*, and *MX-AI*, the thesis moves from business-plane negotiation, to symbiotic control, to agentic governance, and finally to end-to-end observability and control on a live Open RAN network. All proposals are implemented and evaluated on real 5G/O-RAN testbeds, providing concrete evidence that agentic AI can operate within practical latency and resource constraints.

Collaborative Business Plane (Maestro)

Chapter 2 introduced *Maestro*, a collaborative business-plane framework addressing CH1. Each stakeholder (e.g., vertical, MVNO, MNO) is represented by a persona-rich LLM agent that expresses and defends its slice objectives in natural language. A neutral mediator aggregates these intents, surfaces conflicts and converges them into a consensus service-level agreement (SLA) that can be enforced as RAN slice configurations.

Maestro demonstrates that:

- high-level business intents (cost, QoE, risk tolerance) can be captured in free text and mapped to concrete KPIs and resource shares;
- multi-party negotiations converge in a small number of rounds, with fairness across tenants; and
- persona design and incentives matter: unconstrained agents may behave strategically, but structured protocols lead to stable outcomes.

Conceptually, Maestro establishes the *collaborative business plane*: a layer where stakeholders negotiate over slice SLAs before any low-level control is applied. This directly addresses CH1 and sets the stage for more principled, optimization-aware negotiations in later chapters.

Symbiotic Agents and future AGI-RAN Architecture

Chapter 3 tackled CH2 by asking how LLMs can be embedded into time-sensitive control loops without sacrificing reliability. The result is the *Symbiotic Agents* paradigm and the

first direction towards AGI-RAN prototypes.

Two agent types are introduced:

- **Type I agents** for RAN control: lightweight SLMs operate in near-real-time loops, selecting and tuning classical controllers (e.g., PID) that drive PRB allocation and power caps. The LLM proposes *what* to adjust; the optimizer decides *how much*.
- **Type II agents** for SLA negotiation: LLM agents act at non-RT timescales, interpreting stakeholder intents and steering throughput/latency trade-offs while a side-car optimizer enforces feasibility.

On a live OAI/FlexRIC testbed, Symbiotic Agents:

- reduce decision error by up to a factor of five compared to plain prompting;
- achieve near-RT loop latencies on the order of tens of milliseconds using 1-3B SLMs; and
- save a substantial fraction of PRBs in mobility scenarios (vehicle route) by dynamically down-scaling over-provisioned slices.

In parallel, the AGI-RAN architecture generalizes this pattern: ultra-light controllers in the RAN, SLMs at the near-RT RIC, and larger LLMs at the non-RT/SMO. Together, they form a *symbiotic* stack where generative reasoning and numeric optimization co-evolve. This provides a concrete answer to CH2: LLMs can be safely integrated into RAN control when anchored by optimization and layered deployment.

AGORAN: An Open Agentic Marketplace

Chapter 4 addressed CH3 by elevating negotiation and governance into a first-class system called *AGORAN*. Inspired by the ancient Greek agora, AGORAN implements an *open marketplace* where CSC/NSC controllers and the operator (SRB) negotiate SLA offers under explicit rules.

Its key ingredients are:

- a **Service & Resource Broker (SRB)** that runs a multi-objective optimizer (NSGA-II) to generate bounded Pareto-optimal SLA offers across throughput, latency, cost, and energy KPIs;
- **broker agents (bAgents)** attached to each CSC/NSC and the SRB, which deliberate over Pareto indices rather than arbitrary configurations; and
- a **tripartite governance model**: Legislative (*lAgent*) for RAG-based regulatory compliance, Executive (*eAgent*) for live observability via watchers and vector stores, and Judicial (*jAgent*) for trust scoring, toxicity detection, and incentive mechanisms.

On the testbed, AGORAN:

- achieves one-round consensus on SLA offers in realistic mobility scenarios;
- increases composite KPI utility while reducing resource consumption compared to static and purely centralized baselines; and
- shows that warning incentives and trust scores can minimize malicious or toxic agent behavior without breaking convergence.

AGORAN therefore answers CH3: it provides an architectural and protocol blueprint for multi-stakeholder resource marketplaces with explicit guard-rails, auditable Pareto trade-offs, and embedded notions of compliance and trust.

MX-AI: Agentic Observability and Control

Chapter 5 operationalized CH4 and CH5 through *MX-AI*, an agentic observability and control platform integrated into a commercial-grade Open RAN environment. MX-AI instruments an OAI gNB and 5G core controlled by FlexRIC and BubbleRAN SMO, streaming CRDs, xApp state, and KPIs into a vector database via Kubernetes and O-RAN watchers.

On top of this data plane, MX-AI deploys a *multi-agent graph* at the R1 interface:

- a **Routing agent** classifies prompts into observability vs. control;
- **Monitoring agents** combine RAG with live tools (Prometheus, eBPF, RIC APIs) to answer questions such as "What is causing throughput degradation on slice A?";
- **Deployment agents** translate commands such as "Optimize resource allocation for URLLC devices" into valid R1/A1 policies and E2 messages; and
- a **logging agent** records all exchanges for audit and longitudinal benchmarking.

Evaluation on 150 realistic operator tasks shows that:

- capable LLM back-ends achieve near-perfect action accuracy once tools are correctly grounded;
- observability quality is primarily limited by retrieval and tool engineering rather than model size;
- local 70B-class models reach expert-level answer coherence with $\sim$12–14 s latency and full data sovereignty, while small quantized SLMs achieve sub-second responses at lower quality; and
- the agent's time-to-action (TTA) is already competitive with human experts and substantially faster than novices.

MX-AI also generalizes the framework to CH5 by treating AI inference services as *first-class tenants*. AI workloads appear as Kubernetes objects with SLOs, and the same agentic pipeline can reallocate GPU and CPU resources alongside PRBs and power caps. This illustrates how AI-for-RAN and AI-on-RAN can share a unified control fabric.

Novelty and Key Insights

Across all chapters, the thesis advances the state of the art along four main axes:

End-to-end AI-RAN integration. Rather than attaching a single LLM to a single xApp, the work spans the entire AI-RAN stack: from business-plane intent negotiation (Maestro), to symbiotic near-RT control (Symbiotic Agents / AGI-RAN), to governed open marketplaces (AGORAN), and finally to SMO-level observability and actuation (MX-AI). This provides a holistic blueprint for AI-native 6G networks.

Symbiosis of LLMs and optimization. The thesis demonstrates that LLMs and SLMs are most effective when paired with numeric optimizers and controllers: optimization guarantees feasibility and latency, while LLMs handle unstructured intents, explanations, and negotiation. This *symbiotic* pattern recurs in every system and is central to mitigating hallucinations and enforcing safety in control loops.

Governed multi-stakeholder negotiation. AGORAN's SRB, Pareto-front generation, and tripartite governance model introduce a new design space for resource markets in telecoms. The combination of Legislative, Executive, and Judicial branches brings regulatory, operational, and trust concerns into the same agentic framework rather than treating them as afterthoughts.

Real-system prototypes and benchmarks. All proposals are validated on over-the-air 5G/O-RAN testbeds, with detailed measurements of convergence time, KPI improvements, LLM/SLM quality, latency, and GPU footprint. This grounds the thesis in engineering reality and exposes the current limits of models, hardware, and tool-chains.

Overall, the thesis argues that agentic AI, when combined with optimization, governance, and observability, can make 6G RANs *open, programmable, intent-driven, autonomous and trustworthy*.

6.2 Future Work

While the proposed systems take concrete steps toward AI-native RANs, several open challenges remain. We highlight directions that naturally extend the thesis and align with CH1–CH5.

Richer Agentic Marketplaces and Governance

AGORAN focuses on Pareto-bounded SLA brokering with warning-based incentives. Future marketplaces could incorporate:

- dynamic pricing and auction mechanisms (e.g., VCG, combinatorial auctions) to incentivise truthful bidding and time-varying demand;
- multi-objective RL or actor–critic agents that adapt negotiation strategies based on historical outcomes;
- cross-domain bargaining across RAN, transport, edge, and cloud resources; and
- formal metrics for fairness, diversity, and regulatory compliance, feeding directly into the Judicial branch and Trust Score.

Extending AGORAN to multi-RIC, multi-operator settings with federated SRBs and inter-domain treaties is a natural next step for CH1 and CH3.

Lightweight, Specialized Language Models

The evaluation in Chapters 3–5 shows that carefully fine-tuned SLMs (1–8B) can approach large-model quality at a fraction of the latency and energy. Future work should:

- develop telecom-specialized SLMs trained on standards, logs, and slice telemetry;
- systematically explore quantization, pruning, LoRA, and distillation for near-RT deployment on RIC-class hardware;
- co-design SLMs with digital twins and synthetic data to safely pre-train negotiation and control policies; and
- investigate model families optimized for tool use and retrieval rather than general dialogue.

This directly advances CH2 and CH5 by making symbiotic agents deployable at the edge.

Scalability, Robustness, and Safety

Scaling from a single testbed to operator-scale networks introduces new problems:

- Thousands of cells and stakeholders demand hierarchical agent graphs, mediator sharding, and decomposed optimization (per-cell, per-cluster, per-tenant).

- Robustness to adversarial prompts, model drift, and misbehaving agents will require stronger trust frameworks, red-teaming, anomaly detection, and fail-safe defaults.
- Privacy-preserving techniques (federated learning, secure aggregation, or holomorphic encryption) could allow cross-operator collaboration without exposing raw telemetry.

Embedding formal verification, safety constraints, and "emergency stop" policies into the agentic loop is crucial before deployment in safety-critical environments.

Multi-RAT, Non-Terrestrial, and Joint Communication–Compute

Future AI-RANs will span multiple radio technologies, non-terrestrial segments, and joint communication–sensing–compute planes. Extending agentic frameworks to:

- coordinate spectrum and slices across 5G/6G, Wi-Fi, satellite, and NTN;
- incorporate sensing, localization, and environment maps into observability and negotiation; and
- reason over joint communication–compute workloads (e.g., offloading between edge, cloud, and devices)

will be essential to meet IMT-2030 requirements and further develop CH4 and CH5.

AI Workloads as Tenants and Human–AI Co-operation

MX-AI's first steps on AI workloads as tenants suggest several avenues:

- richer scheduling policies that jointly optimize RAN slices and AI inference services, including preemption and graceful degradation;
- feedback loops where agent performance (e.g., negotiation quality, observability accuracy) is exposed as a KPI influencing its own resource budget; and
- deeper study of human–AI collaboration: trust calibration, explanation quality, and user interfaces that let operators override or refine agent proposals without breaking control-loop stability.

These directions complete the AI-for-RAN / AI-and-RAN / AI-on-RAN triangle envisioned at the start of the thesis.

Standardization and Ecosystem Building

Finally, turning prototypes into practice requires standardization and ecosystems:

- introducing agent-centric concepts (HBI/H2A interfaces, SRB APIs, trust scores, negotiation protocols) into O-RAN and 3GPP discussions;
- extending open-source platforms (e.g., OAI, FlexRIC, MX-AI SDKs) with reference implementations of agentic controllers, marketplaces, and observability loops; and
- developing regulatory and ethical guidelines for transparency, accountability, and auditability of autonomous network decisions.

In conclusion, this thesis shows that agentic AI can bridge the gap between human intents and machine-level actions in future RANs. By combining LLM/SLM agents with optimization, governed marketplaces, and rich observability, we take concrete steps toward AI-native 6G networks that are open, flexible, and equitable. Continued work along the directions above will turn these architectural blueprints into deployable systems that underpin the next generation of communication infrastructure.

Reference

[1] 3GPP. *Intent Driven Management*. Accessed: 2024-10-14. n.d. URL: https://www.3gpp.org/technologies/intent.

[2] 3GPP. *Telecommunication management; Management and orchestration; Provisioning*. 3GPP Technical Specification TS 28.531 V18.8.0. Release 18. Sophia Antipolis, France: 3GPP, Jan. 2025. URL: https://www.3gpp.org/DynaReport/28531.htm.

[3] 3rd Generation Partnership Project (3GPP). *Telecommunication management; Management and orchestration; Concepts, use cases and requirements*. 3GPP Technical Specification TS 28.530 V18.2.0. Release 18. Sophia Antipolis, France: 3GPP, Jan. 2025. URL: https://www.3gpp.org/DynaReport/28530.htm.

[4] *6G — Follow the journey to the next generation networks*. Ericsson. URL: https://www.ericsson.com/en/6g (visited on 08/23/2025).

[5] *Agentic AI for RAN optimization: Pathway to autonomous networks level 5*. Ericsson Blog. Ericsson. July 29, 2025. URL: https://www.ericsson.com/en/blog/2025/7/agentic-ai-pathway-to-autonomous-network-level-5 (visited on 08/23/2025).

[6] Mistral AI. *Mistral 7B*. https://mistral.ai/news/announcing-mistral-7b/. Accessed: June 2024. 2024.

[7] AI-RAN Alliance. *AI-RAN Alliance: Advancing AI-Native Radio Access Networks*. Accessed: March 13, 2025. 2025. URL: https://ai-ran.org/.

[8] AI-RAN Alliance. *AI-RAN Alliance Vision and Mission White Paper*. Tech. rep. Dec. 2024. URL: https://ai-ran.org/wp-content/uploads/2024/12/AI-RAN_Alliance_Whitepaper.pdf.

[9] AI–RAN Alliance. *AI–RAN Alliance Vision and Mission White Paper*. Tech. rep. Defines the AI–native RAN vision, motivations, and roadmap. AI–RAN Alliance, Dec. 2024. URL: https://ai-ran.org/wp-content/uploads/2024/12/AI-RAN_Alliance_Whitepaper.pdf.

[10] Mazene Ameur, Bouziane Brik, and Adlen Ksentini. "Leveraging LLMs to explain DRL decisions for transparent 6G network slicing". In: *2024 IEEE 10th International Conference on Network Softwarization (NetSoft)*. IEEE. 2024, pp. 204–212.

[11] Lina Bariah et al. "Large generative ai models for telecom: The next big thing?" In: *IEEE Communications Magazine* 62.11 (2024), pp. 84–90.

[12] Lina Bariah et al. "Large Language Models for Telecom: The Next Big Thing?" In: *arXiv preprint arXiv:2306.10249* (2023).

[13] Lina Bariah et al. "Understanding Telecom Language Through Large Language Models". In: *arXiv preprint arXiv:2306.07933* (2023).

[14] Mattia Benedetto et al. *Key Drivers and Research Challenges for 6G Ubiquitous Wireless Intelligence*. Tech. rep. 6G Flagship, 2019. URL: https://jultika.oulu.fi/files/isbn9789526223544.pdf.

[15] Dimitri P. Bertsekas. *Nonlinear Programming*. 2nd ed. Belmont, MA, USA: Athena Scientific, 1999. ISBN: 978-1886529007.

[16] Sarah Bouhanna, Abhishek Gupta, and Laurent Romary. "LexDrafter: Retrieval-Augmented Drafting of Legislative Definitions". In: *Proc. JURIX*. 2024. URL: https://arxiv.org/abs/2403.16295.

[17] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004. ISBN: 978-0-521-83378-3. URL: https://web.stanford.edu/~boyd/cvxbook/.

[18] Chi-Min Chan et al. "Chateval: Towards better llm-based evaluators through multi-agent debate". In: *arXiv preprint arXiv:2308.07201* (2023).

[19] I Chatzistefanidis et al. *UE statistics time-series (CQI) in LTE networks*. 2022.

[20] Ilias Chatzistefanidis, Andrea Leone, and Navid Nikaein. "Maestro: LLM-Driven Collaborative Automation of Intent-Based 6G Networks". In: *IEEE Networking Letters* 6.4 (2024), pp. 227–231. DOI: 10.1109/LNET.2024.3503292.

[21] Ilias Chatzistefanidis and Navid Nikaein. *Symbiotic Agents: A Novel Paradigm for Trustworthy AGI-driven Networks*. 2025. arXiv: 2507.17695 [cs.AI]. URL: https://arxiv.org/abs/2507.17695.

[22] Ilias Chatzistefanidis et al. "Agoran: An Agentic Open Marketplace for 6G RAN Automation". In: *arXiv preprint arXiv:2508.09159* (2025).

[23] Ilias Chatzistefanidis et al. "ML-based Traffic Steering for Heterogeneous Ultra-dense beyond-5G Networks". In: *2023 IEEE Wireless Communications and Networking Conference (WCNC)*. 2023, pp. 1–6. DOI: 10.1109/WCNC55385.2023.10118923.

[24] Ilias Chatzistefanidis et al. *MX-AI: Agentic Observability and Control Platform for Open and AI-RAN*. 2025. arXiv: 2508.09197 [cs.NI]. URL: https://arxiv.org/abs/2508.09197.

[25] Ilias Chatzistefanidis et al. "Which ML Model to Choose? Experimental Evaluation for a Beyond-5G Traffic Steering Case". In: *ICC 2023 - IEEE International Conference on Communications*. 2023, pp. 5185–5190. DOI: 10.1109/ICC45041.2023.10279485.

[26] Hao Chen, Gonzalo E Constante-Flores, and Can Li. "Diagnosing infeasible optimization problems using large language models". In: *INFOR: Information Systems and Operational Research* 62.4 (2024), pp. 573–587.

[27] Mahfuzul Hoq Chowdhury. "Accelerator: An Intent-Based Intelligent Resource-Slicing Scheme for SFC-Based 6G Application Execution over SDN- and NFV-Empowered Zero-Touch Networks". In: *Frontiers in Communications and Networks* 5 (2024). DOI: 10.3389/frcmn.2024.1385656.

[28] A. Muñiz da Costa and L.M. Contreras. "Integration of Network Slice Controller for Enhanced Intent-Based Networking in 5G/6G Networks". In: *Proc. ACM MobiArch*. 2023, pp. 1–6. DOI: 10.1145/3615587.3615989.

[29] Zhuangzhuang Cui, Peize Zhang, and Sofie Pollin. "6G wireless communications in 7-24 GHz band: Opportunities, techniques, and challenges". In: *arXiv preprint arXiv:2310.06425* (2023).

[30] Salvatore D'Oro et al. "dApps: Distributed applications for real-time inference and control in O-RAN". In: *IEEE Communications Magazine* 60.11 (2022), pp. 52–58.

[31] Fei Dou et al. "Towards artificial general intelligence (agi) in the internet of things (iot): Opportunities and challenges". In: *arXiv preprint arXiv:2309.07438* (2023).

[32] Yilun Du et al. "Improving Factuality and Reasoning in Language Models through Multiagent Debate". In: *arXiv preprint arXiv:2305.14325* (2023).

[33] Eslam Eldeeb and Hirley Alves. *Offline and Distributional Reinforcement Learning for Wireless Communications*. 2025. arXiv: 2504.03804 [cs.LG]. URL: https://arxiv.org/abs/2504.03804.

[34] M. Elkael et al. "ALLSTaR: Automated LLM-Driven Scheduler Generation and Testing for Intent-Based RAN". In: *arXiv* 2505.18389 (2025).

[35] Ericsson. *Enabling Time-Critical Applications over 5G with Rate Adaptation*. White Paper. May 2025. URL: https://www.ericsson.com/en/reports-and-papers/white-papers/enabling-time-critical-applications-over-5g-with-rate-adaptation.

[36] Ericsson. *Explore CI/CD: The Benefits and Role in Telecom*. Section "Increased efficiency" highlights removal of manual workflows. 2025. URL: https://www.ericsson.com/en/ci-cd.

[37] Ericsson and Turkcell. *Turkcell Achieves First Core-Network Automation Milestone*. Mar. 2025. URL: https://www.ericsson.com/en/press-releases/5/2025/ericsson-and-turkcell-achieve-core-network-automation-milestone.

[38] ETSI. *Zero-Touch Network and Service Management (ZSM); Intent-driven Autonomous Networks; Generic Aspects*. Group Report ETSI GR ZSM 011 V2.1.1. Sept. 2024. URL: https://www.etsi.org/deliver/etsi_gr/ZSM/001_099/011/02.01.01_60/gr_ZSM011v020101p.pdf.

[39] *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures*. Tech. rep. TS 36.213. Version 15.3.0. Available at: https://www.3gpp.org/DynaReport/36213.htm. 3rd Generation Partnership Project (3GPP), 2018.

[40] Fortune Business Insights. *Cloud Gaming Market Size, Share & Industry Analysis 2025 – 2032*. Last accessed July 2025. June 2025. URL: https://www.fortunebusinessinsights.com/cloud-gaming-market-102495.

[41] Jinlan Fu et al. "Gptscore: Evaluate as you desire". In: *arXiv preprint arXiv:2302.04166* (2023).

[42] Aaron Grattafiori, Abhimanyu Dubey, et al. "The Llama 3 Herd of Models". In: *arXiv preprint arXiv:2407.21783* (2024). URL: https://arxiv.org/abs/2407.21783.

[43] F. H. Grings et al. "NASP: Network Slice as a Service Platform for 5G Networks". In: *Computer Networks* (2025). arXiv:2505.24051.

[44] GSMA Intelligence. *The Mobile Economy 2025*. Tech. rep. GSMA, Feb. 2025. URL: https://www.gsma.com/solutions-and-impact/connectivity-for-good/mobile-economy/wp-content/uploads/2025/02/030325-The-Mobile-Economy-2025.pdf.

[45] Pei-Fu Guo et al. "Towards optimizing with large language models". In: *arXiv preprint arXiv:2310.05204* (2023).

[46] Zishan Guo et al. "Evaluating large language models: A comprehensive survey". In: *arXiv preprint arXiv:2310.19736* (2023).

[47] Mohammad Asif Habibi et al. "Toward an Open, Intelligent, and End-to-End Architectural Framework for Network Slicing in 6G Communication Systems". In: *IEEE Open Journal of the Communications Society* 4 (2023), pp. 1615–1658. DOI: 10.1109/OJCOMS.2023.3294445.

[48] Long He et al. "Generative AI for Game Theory-based Mobile Networking". In: *arXiv preprint arXiv:2404.09699* (2024).

[49] Cyril Shih-Huan Hsu et al. "Online SLA Decomposition: Enabling Real-Time Adaptation to Evolving Network Systems". In: *arXiv* 2408.08968 (2024). URL: https://arxiv.org/abs/2408.08968.

[50] Lei Huang et al. "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions". In: *ACM Transactions on Information Systems* 43.2 (2025), pp. 1–55.

[51] IEEE Standards Association. *IEEE 7001™ – Standard for Transparency of Autonomous Systems*. 2021. URL: https://standards.ieee.org/standard/7001-2021.html.

[52] IEEE Standards Association. *IEEE Standard 7000-2021—Model Process for Addressing Ethical Concerns during System Design*. 2021. URL: https://standards.ieee.org/standard/7000-2021.html.

[53] *ISO/IEC 42001:2023 Information technology — Artificial intelligence — Management system*. en. International Standard. Geneva, Switzerland: International Organization for Standardization and International Electrotechnical Commission, Dec. 2023. URL: https://www.iso.org/standard/81230.html.

[54] ITU-R. *Framework and Overall Objectives of the Future Development of IMT for 2030 and Beyond (IMT-2030)*. Recommendation M.2160-0. International Telecommunication Union – Radiocommunication Sector, Nov. 2023. URL: https://www.itu.int/rec/R-REC-M.2160-0-202311-I/en.

[55] ITU-R. *Recommendation ITU-R M.2160-0: IMT-2030 Framework*. Tech. rep. International Telecommunication Union, Nov. 2023. URL: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2160-0-202311-I!!PDF-E.pdf.

[56] ITU-R. *World Radiocommunication Conference 2023 – Resolution 256: Use of IMT in 7.125–8.4 GHz and 14.8–15.35 GHz*. Conference Resolution. Agenda Item 1.7 outcome, Dubai. International Telecommunication Union, 2023. URL: https://www.itu.int/en/ITU-R/seminars/wrs/Documents/2024/Presentations/05-WRCs%20by%20PhA.pdf.

[57] Shufan Jiang et al. "LINKs: Large Language Model Integrated Management for 6G Empowered Digital Twin Networks". In: *arXiv preprint arXiv:2412.19811* (2024). accepted VTC2024-Fall.

[58] Florian Kaltenberger et al. "OpenAirInterface: Democratizing Innovation in the 5G Era". In: *Computer Networks* 176 (2020), p. 107284. DOI: 10.1016/j.comnet.2020.107284.

[59] Maurice G. Kendall. "A new measure of rank correlation". In: *Biometrika* 30.1-2 (1938), pp. 81–93. DOI: 10.1093/biomet/30.1-2.81.

[60] Woo-Hyun Ko et al. "EdgeRIC: Empowering Realtime Intelligent Optimization and Control in NextG Cellular Networks". In: *Proc. USENIX NSDI*. 2024, pp. 1315–1330.

[61] Andrea Lacava et al. "dApps: Enabling Real-Time AI-Based Open RAN Control". In: *arXiv preprint arXiv:2501.16502* (2025).

[62] Aris Leivadeas and Matthias Falkner. "A survey on intent-based networking". In: *IEEE Communications Surveys & Tutorials* 25.1 (2022), pp. 625–655.

[63] Haitao Li et al. "LexRAG: Benchmarking Retrieval-Augmented Generation in Multi-Turn Legal Consultation Conversation". In: *arXiv* 2502.20640 (2025). Findings of ACL 2025. URL: https://arxiv.org/abs/2502.20640.

[64] Yuyan Lin et al. "Hierarchical Debate-Based Large Language Model for Complex Task Planning of 6G Network Management". In: *arXiv preprint arXiv:2506.06519* (2025). ICML 2025 Workshop on ML4Wireless.

[65] Zheng Lin et al. "Pushing Large Language Models to the 6G Edge: Vision, Challenges, and Opportunities". In: *arXiv preprint arXiv:2309.16739* (2023).

[66] Fei Liu et al. "Large language model for multiobjective evolutionary optimization". In: *International Conference on Evolutionary Multi-Criterion Optimization*. Springer. 2025, pp. 178–191.

[67] Jian Liu, Marzieh Saeidi, and Dan Goldwasser. "AgentBench: Benchmarking Large Language Models as Agents". In: *arXiv* 2308.11432 (2023). URL: https://arxiv.org/abs/2308.11432.

[68] Yang Liu et al. "G-eval: NLG evaluation using gpt-4 with better human alignment". In: *arXiv preprint arXiv:2303.16634* (2023).

[69] F. Lotfi, H. Rajoli, and F. Afghah. "Prompt-Tuned LLM-Augmented DRL for Dynamic O-RAN Network Slicing". In: *arXiv* 2506.00574 (2025).

[70] *Large Language Models in the 6G-Enabled Computing Continuum: a White Paper*. White paper 14. University of Oulu, 2025. URL: https://urn.fi/URN:NBN:fi:oulu-202501211268 (visited on 08/23/2025).

[71] Ali Maatouk et al. *Tele-LLMs: A Series of Specialized Large Language Models for Telecommunications*. 2024. arXiv: 2409.05314 [cs.IT]. URL: https://arxiv.org/abs/2409.05314.

[72] B. Martini, M. Gharbaoui, and P. Castoldi. "Intent-Based Network Slicing for SDN Vertical Services with Assurance". In: *Future Generation Computer Systems* 142 (2023), pp. 101–116. DOI: 10.1016/j.future.2022.12.033.

[73] Abdelkader Mekrache, Adlen Ksentini, and Christos Verikoukis. "Intent-based management of next-generation networks: An LLM-centric approach". In: *Ieee Network* (2024).

[74] Nikola Milosevic, Johannes Müller, and Nico Scherf. *Central Path Proximal Policy Optimization*. 2025. arXiv: 2506.00700 [cs.LG]. URL: https://arxiv.org/abs/2506.00700.

[75] Alireza Mohammadi and Navid Nikaein. "ATHENA: An Intelligent Multi-x Cloud Native Network Operator". In: *IEEE Journal on Selected Areas in Communications* (2023).

[76] Ahmad M. Nagib, Hatem Abou-Zeid, and Hossam S. Hassanein. "SafeSlice: Enabling SLA-Compliant O-RAN Slicing via Safe Deep Reinforcement Learning". In: *IEEE ICMLCN 2025 (to appear), arXiv:2503.12753* (2025). URL: https://arxiv.org/abs/2503.12753.

[77] Sagar Nangare. *AI-Native 6G: The Convergence of Intelligence and Connectivity*. Blog post. Feb. 12, 2025. URL: https://sagarnangare.com/ai-native-6g-the-convergence-of-intelligence-and-connectivity/ (visited on 08/23/2025).

[78] NGMN Alliance. *6G Drivers and Vision*. Tech. rep. NGMN, 2021. URL: https://www.ngmn.org/wp-content/uploads/NGMN-6G-Drivers-and-Vision-V1.0_final.pdf.

[79] NGMN Alliance. *Network Architecture Evolution towards 6G*. White Paper. Version 1.0, 18 Feb 2025. Frankfurt, Germany: Next Generation Mobile Networks (NGMN) Alliance, Feb. 2025. URL: https://www.ngmn.org/wp-content/uploads/250218_Network_Architecture_Evolution_towards_6G_V1.0.pdf.

[80] Ha Thanh Nguyen and Ken Satoh. "KRAG: A Knowledge-Representation-Augmented Generation Framework for Legal Reasoning". In: *arXiv* 2410.07551 (2024). URL: https://arxiv.org/abs/2410.07551.

[81] Navid Nikaein et al. "OpenAirInterface: A flexible platform for 5G research". In: *ACM SIGCOMM Computer Communication Review* 44.5 (2014), pp. 33–38.

[82] O-RAN Alliance. *O-RAN: Transforming Radio Access Networks Towards Open, Intelligent, Virtualized and Fully Interoperable RAN*. Accessed: Nov 28, 2025. 2025. URL: https://www.o-ran.org/.

[83] O-RAN Alliance. *O–RAN Architecture Description*. Tech. rep. O–RAN.WG1.OAD–R003 v10.00. Oct. 2023. URL: https://www.o-ran.org/specifications.

[84] *Open RAN explained*. Nokia. URL: https://www.nokia.com/mobile-networks/ran/anyran/open-ran/open-ran-explained/ (visited on 08/23/2025).

[85] OpenAI. *GPT-4o mini: Advancing Cost-Efficient Intelligence*. 2024. URL: https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence (visited on 07/14/2025).

[86] OpenAI. *GPT-family*. https://openai.com/api/. Accessed: July 2025. 2024.

[87] Rajvardhan Patil and Venkat Gudivada. "A review of current trends, techniques, and challenges in large language models (llms)". In: *Applied Sciences* 14.5 (2024), p. 2074.

[88] Shishir G Patil et al. "Gorilla: Large language model connected with massive apis". In: *arXiv preprint arXiv:2305.15334* (2023).

[89] Nicholas Pipitone and Ghita Houir Alami. "LegalBench-RAG: A Benchmark for Retrieval-Augmented Generation in the Legal Domain". In: *arXiv* 2408.10343 (2024). URL: https://arxiv.org/abs/2408.10343.

[90] Ioannis Pitsiorlas, George Arvanitakis, and Marios Kountouris. "Trustworthy Intrusion Detection: Confidence Estimation Using Latent Space". In: *2024 22nd International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. 2024, pp. 92–98.

[91] Ioannis Pitsiorlas, Nour Jamoussi, and Marios Kountouris. *A Conformal Predictive Measure for Assessing Catastrophic Forgetting*. 2025. arXiv: 2505.10677 [cs.LG]. URL: https://arxiv.org/abs/2505.10677.

[92] Yujia Qin et al. "Toolllm: Facilitating large language models to master 16000+ real-world apis". In: *arXiv preprint arXiv:2307.16789* (2023).

[93] Walid Saad et al. "Artificial general intelligence (AGI)-native wireless systems: A journey beyond 6G". In: *arXiv preprint arXiv:2405.02336* (2024).

[94] Konstantinos Samdanis, Xavier Costa-Perez, and Vincenzo Sciancalepore. "From network sharing to multi-tenancy: The 5G network slice broker". In: *IEEE Communications Magazine* 54.7 (2016), pp. 32–39.

[95] Vidit Saxena, Burak Guldogan, and Doumitrou Daniil Nimara. *On-policy and off-policy Reinforcement Learning: Key features and differences. Ericsson Blog*. Published December 13, 2023. Accessed 2025-08-22. Dec. 2023. URL: https://www.ericsson.com/en/blog/2023/12/online-and-offline-reinforcement-learning-what-are-they-and-how-do-they-compare.

[96] Timo Schick, Jonas Pfeiffer, and Hinrich Schütze. "Toolformer: Language Models Can Teach Themselves to Use Tools". In: *Proc. ICLR*. 2024. URL: https://openreview.net/forum?id=7aQdV9BbDv.

[97] Robert Schmidt, Mikel Irazabal, and Navid Nikaein. "FlexRIC: an SDK for next-generation SD-RANs". In: *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*. 2021, pp. 411–425.

[98] Robert Schmidt, Mikel Irazabal, and Navid Nikaein. "FlexRIC: an SDK for next-generation SD-RANs". In: *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies*. CoNEXT '21. New York, NY, USA: Association for Computing Machinery, 2021, 411–425. ISBN: 9781450390989. DOI: 10.1145/3485983.3494870. URL: https://doi.org/10.1145/3485983.3494870.

[99] Gesualdo Scutari et al. "Decomposition by partial linearization: Parallel optimization of multi-agent systems". In: *IEEE Transactions on Signal Processing* 62.3 (2013), pp. 641–656.

[100] Adnan Shahid et al. "Large-Scale AI in Telecom: Charting the Roadmap for Innovation, Scalability, and Enhanced Digital Experiences". In: *arXiv preprint arXiv:2503.04184* (2025).

[101] Zhaolei Shi et al. "BFCN: A novel classification method of encrypted traffic based on BERT and CNN". In: *Electronics* 12.3 (2023), p. 516.

[102] Noah Shinn et al. "Reflexion: Language agents with verbal reinforcement learning". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 8634–8652.

[103] Irene Solaiman, Miles Brundage, and Gillian Hadfield. "Evaluating the Social Impact of Released Large Language Models". In: *arXiv* 2306.06690 (2023). URL: https://arxiv.org/abs/2306.06690.

[104] Jiayang Song et al. "Self-refined large language model as automated reward function designer for deep reinforcement learning in robotics". In: *arXiv preprint arXiv:2309.06687* (2023).

[105] Ivo Sousa, Maria Paula Queluz, and António Rodrigues. "A survey on QoE-oriented wireless resources scheduling". In: *Journal of Network and Computer Applications* 158 (2020), p. 102594.

[106] Charles Spearman. "The proof and measurement of association between two things". In: *The American Journal of Psychology* 15.1 (1904), pp. 72–101. DOI: 10.2307/1412159.

[107] Elham Tabassi. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. en. NIST AI 100-1. NIST Trustworthy and Responsible AI Program. Gaithersburg, MD: National Institute of Standards and Technology, Jan. 2023. DOI: 10.6028/NIST.AI.100-1. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=936225.

[108] Alex Tamkin et al. "Understanding the capabilities, limitations, and societal impact of large language models, 2021". In: *URL https://arxiv. org/abs/2102.02503.(cited on pp. 29, 34, and 39)* ().

[109] Gemma Team. "Gemma: Open Models Based on Gemini Research and Technology". In: *arXiv preprint arXiv:2403.08295* (2024). URL: https://arxiv.org/abs/2403.08295.

[110] Ollama Team. *Ollama: Run Large Language Models Locally*. 2024. URL: https://github.com/ollama/ollama (visited on 07/14/2025).

[111] Theodoros Tsourdinis et al. “AI-driven Service-aware Real-time Slicing for beyond 5G Networks”. In: *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2022, pp. 1–6. DOI: 10.1109/INFOCOMWKSHPS54753.2022.9798391.

[112] Theodoros Tsourdinis et al. “AI-driven Service-aware Real-time Slicing for beyond 5G Networks”. In: *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2022, pp. 1–6. DOI: 10.1109/INFOCOMWKSHPS54753.2022.9798391.

[113] Theodoros Tsourdinis et al. “Service-aware real-time slicing for virtualized beyond 5G networks”. In: *Computer Networks* 247 (2024), p. 110445.

[114] Theodoros Tsourdinis et al. “Service-aware real-time slicing for virtualized beyond 5G networks”. In: *Computer Networks* 247 (2024), p. 110445. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2024.110445. URL: https://www.sciencedirect.com/science/article/pii/S1389128624002779.

[115] Theodoros Tsourdinis et al. “UE network traffic time-series (applications, throughput, latency, CQI) in LTE/5G networks”. In: *IEEE Dataport* (2022).

[116] Antonio Visioli. *Practical PID control*. Springer Science & Business Media, 2006.

[117] Cheng-Xiang Wang et al. “On the road to 6G: Visions, requirements, key technologies, and testbeds”. In: *IEEE Communications Surveys & Tutorials* 25.2 (2023), pp. 905–974.

[118] Xiangyu Wang, Guodong Wang, and Shihua Li. “A distributed fixed-time optimization algorithm for multi-agent systems”. In: *Automatica* 122 (2020), p. 109289.

[119] Jason Wei et al. “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”. In: *Proceedings of the National Academy of Sciences* 119.46 (2022), e2209413119. DOI: 10.1073/pnas.2209413119.

[120] Wen Wu et al. “AI-Native Network Slicing for 6G Networks”. In: *IEEE Wireless Communications* 29.1 (2022), pp. 96–103. DOI: 10.1109/MWC.001.2100338.

[121] X. Wu et al. “LLM-xApp: A Large Language Model Empowered Radio Resource Management xApp for 5G O-RAN”. In: *NDSS FutureG Workshop*. 2025. DOI: 10.14722/futureg.2025.23057.

[122] Xingqi Wu et al. “LLM-xApp: A Large Language Model Empowered Radio Resource Management xApp for 5G O-RAN”. In: *Symposium on Networks and Distributed Systems Security (NDSS), Workshop on Security and Privacy of Next-Generation Networks (FutureG 2025), San Diego, CA*. 2025.

[123] Xingqi Wu et al. “LLM-xApp: A Large Language Model Empowered Radio Resource Management xApp for 5G O-RAN”. In: *Proc. NDSS Symposium*. 2025.

[124] An Yang, Baosong Yang, et al. “Qwen2 Technical Report”. In: *arXiv preprint arXiv:2407.10671* (2024). URL: https://arxiv.org/abs/2407.10671.

[125] Shunyu Yao et al. “ReAct: Synergizing Reasoning and Acting in Language Models”. In: *arXiv* 2210.03629 (2023). URL: https://arxiv.org/abs/2210.03629.

[126] Edwin Yaqub et al. “Optimal negotiation of service level agreements for cloud-based services through autonomous agents”. In: *2014 IEEE International Conference on Services Computing*. IEEE. 2014, pp. 59–66.

[127] Lifan Yuan et al. “Revisiting out-of-distribution robustness in nlp: Benchmarks, analysis, and llms evaluations”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 58478–58507.

[128] Jintian Zhang, Xin Xu, and Shumin Deng. “Exploring collaboration mechanisms for llm agents: A social psychology view”. In: *arXiv preprint arXiv:2310.02124* (2023).

[129] Hao Zhou et al. “Large Language Model (LLM) for Telecommunications: A Comprehensive Survey on Principles, Key Techniques, and Opportunities”. In: *IEEE Communications Surveys & Tutorials* (2024), pp. 1–1. DOI: 10.1109/COMST.2024.3465447.

[130] Hang Zou et al. “TelecomGPT: A framework to build telecom-specfic large language models”. In: *arXiv preprint arXiv:2407.09424* (2024).

[131] Hang Zou et al. “Wireless multi-agent generative AI: From connected intelligence to collective intelligence”. In: *arXiv preprint arXiv:2307.02757* (2023).

[132] Karl J. Åström and Tore Hägglund. *Advanced PID Control*. Research Triangle Park, NC: ISA – The Instrumentation, Systems, and Automation Society, 2006. ISBN: 978-1-55617-942-6.

Appendix A

Appendix

A.1 Architecture Workflow 128
A.2 Optimizer KPI Models and Algorithmic Details 129
A.2.1 Alternative multi-objective formulations 129
A.2.2 KPI Models 129
A.2.3 Latency models beyond $M/M/1$ 130
A.2.4 Dynamic MCS updates from CQI 130
A.2.5 Repair operator and diversity 130
A.2.6 Complexity and scalability 130
A.2.7 Cost and energy models 131
A.2.8 Offer encoding 131
A.3 Trust-Score Framework Details 131
A.3.1 Deviation Components for Satisfaction Score 131
A.3.2 Behavior of the Agents 132
A.4 Use Case Validation: Detailed Technical Details and Messages Exchanged 134
A.5 Technology-agnostic design. 138

ID	Actor	Interf.	Message Content
1	*Human*	*HBI*	NSC 1 Intent: {Use-case: eMBB; SLA Constraints: $T^{\min}$=25 Mbps, $L^{\max}$=10 ms, $C^{\max}$=50\$, $E^{\max}$=100 W; Context: "My slice targets an eMBB use case and requires maximum QoS."}
2	*Human*	*HBI*	NSC 2 Intent: {Use-case: mMTC; SLA Constraints: $T^{\min}$=5 Mbps, $L^{\max}$=10 ms, $C^{\max}$=10\$, $E^{\max}$=100 W; Context: "My slice supports an mMTC service; minimizing OPEX is crucial."}
3	*Human*	*HBI*	SRB Owner: {Use-case: regulator; SLA Constraints: national policy compliance, minimize energy E; Context: "Offer multiple trade-offs, mediate toward fair consensus, and prioritize low energy footprint to comply with national regulations."}
4	*NSC*	*SBI*	NSC 1 → SRB: {Structured intent: SLA constraints, use-case = eMBB}
5	*NSC*	*SBI*	NSC 2 → SRB: {Structured intent: SLA constraints, use-case = mMTC}
6-7	*SRB*	*SBI*	SRB → NSCs: {Proposal 1, Proposal 2, Proposal 3, . . .; each Pareto-optimal}
8	*bAgent*	*SBI*	NSC 1: {Selected = Proposal 1; Rationale: "Maximizes throughput and QoS; higher cost acceptable."}
9	*bAgent*	*SBI*	NSC 2: {Selected = Proposal 2; Rationale: "Minimizes OPEX while maintaining adequate QoS."}
10-11	*bAgent*	*SBI*	SRB: {Recommended = Proposal 3; Rationale: "Balances NSC 1 QoS, NSC 2 cost, and national energy targets."}
12	*SRB*	*R1*	Slice & Consensus Intent: {Proposal 3 accepted by all parties}, resource policies derived from Consensus Intent
13	*RIC*	*A1*	Enforcement of policies on RAN infrastructure

Table A.1: Messages exchanged during negotiation. Human stakeholders provide the *triple intent input* to *NSC* controllers. The use-case and constraints are forwarded to the SRB, which bounds the optimization problem and computes Pareto-optimal SLA proposals. Negotiation then unfolds among *bAgents*, who select and argue over proposals using both the feasible KPI vectors and their stakeholders' NLP reasoning. After consensus, the SRB decomposes the selected proposal into slice- and resource-level policies for enforcement.

A.1 Architecture Workflow

The dynamics of the negotiation loop are presented in the sequence diagram of Fig. 4.3. Messages 1–3 capture the inputs from three human stakeholders (a vertical, a service provider, and an MNO/regulator), each expressing potentially conflicting priorities. In steps 4–5, the two *NSC* controllers encapsulate the three input components into structured intents, forwarding the explicit KPI constraints together with the declared use-case label, while preserving the accompanying natural-language context for subsequent negotiation. In steps 6–7, the SRB-side mediator responds with a signed list of *Pareto-optimal offers*. Steps 8–11 constitute the consensus loop employing the *bAgents*: every NSC presents its preferred offer (steps 8–9) and the SRB issues a reconciled recommendation (Steps 10-11). Once all parties agree on the same offer index, the negotiation loop terminates. The SRB then decomposes the agreement into slice- and resource-level KPIs (Step 12), and finally into radio resource policies, which are conveyed over the A1/E2 interface (Step 13). Table A.1 details the concrete payloads exchanged during this run, demonstrating that, once the high-level objectives are defined, all subsequent negotiation and optimization are fully machine-driven.

A.2 Optimizer KPI Models and Algorithmic Details

A.2.1 Alternative multi-objective formulations

(i) $4N$**-objective vector.** Optimize $\big(-T_{\mathrm{e}}, L_{\mathrm{e}}, C_{\mathrm{e}}, E_{\mathrm{e}}, \ldots, -T_{\mathrm{m}}, L_{\mathrm{m}}, C_{\mathrm{m}}, E_{\mathrm{m}}\big)$. **(ii)** ε**-constraint.** E.g., minimize L_{u} subject to $T_{\mathrm{e}} \geq \varepsilon_T$, $C_{\mathrm{m}} \leq \varepsilon_C$, $E_{\mathrm{tot}} \leq \varepsilon_E$; sweep ε to produce the front. Both yield equivalent Pareto sets $\mathcal{P}^\star$ for negotiation, while the 4-D formulation in the main text keeps the manuscript concise.

A.2.2 KPI Models

Throughput. Following 3GPP TS 38.306 spectral efficiency tables, the throughput of slice i is given by

$$T_i(b_i, m_i) = \kappa\, Q_{m_i} R_{m_i}\, b_i, \tag{A.1}$$

where $\kappa \approx 0.86$ is the overhead factor (TS 38.211). The MCS index $m_i(t)$ is *updated dynamically* from mobility-trace CQI measurements using a filtered, hysteresis-mapped function $m_i(t) = \Phi(\mathrm{CQI}_i(t))$; see A.2.4.

Latency. To avoid optimistic delay estimates under bursty traffic, we apply Kingman's $G/G/1$ approximation:

$$L_i = L_i^{\mathrm{fix}} + \frac{\rho_i}{\mu_i(1-\rho_i)} \cdot \frac{c_{a,i}^2 + c_{s,i}^2}{2}, \tag{A.2}$$

where $\mu_i(b_i, m_i) = \dfrac{T_i(b_i, m_i) \cdot 10^6}{S_{\mathrm{pkt}}}$ is the service rate, ρ_i is the load, and $c_{a,i}^2$, $c_{s,i}^2$ are the squared coefficients of variation of the arrival and service processes, capturing burstiness and scheduler batching effects. Further details, including $M/G/1$ and finite-blocklength refinements for URLLC, are provided in A.2.3.

Cost. We model the monetary cost using a first-order tariff:

$$C_i = \alpha_c\, c_i + \alpha_s\, s_i \quad \text{(with piecewise-linear extensions detailed in A.2.7).} \tag{A.3}$$

This linear form is used as a transparent first-order approximation of typical operator charging models over the relevant range; more complex or tiered tariffs are supported via the piecewise-linear construction in A.2.7 without modifying the Agoran framework.

Energy. Energy is *time-dependent*:

$$E_i \;=\; \int_0^W P_i(t)\, dt \;\approx\; \bar{P}_i\, \tau_i, \quad \bar{P}_i = P_{0,i} + \Delta P\, p_i, \tag{A.4}$$

over window W, with duty cycle τ_i linked to ρ_i and scheduler behavior; see A.2.7.

A.2.3 Latency models beyond $M/M/1$

Kingman $G/G/1$ (used in main text). $L_i = L_i^{\text{fix}} + \dfrac{\rho_i}{\mu_i(1-\rho_i)} \cdot \dfrac{c_{a,i}^2 + c_{s,i}^2}{2}$, with $c_{a,i}^2$ (arrival SCV) capturing burstiness and $c_{s,i}^2$ (service SCV) capturing batch/scheduler variability.

$M/G/1$ **with SCV.** Set $c_{a,i}^2 = 1$; specify $c_{s,i}^2$ from code block lengths/scheduling.

Finite-blocklength URLLC note. For short packets and stringent reliability requirements, the effective service rate can be adjusted to account for finite-block-length coding penalties; under these conditions, the above expressions provide an upper bound on latency.

A.2.4 Dynamic MCS updates from CQI

Let $\mathrm{CQI}_i(t)$ be the per-slice trace. We map CQI to MCS with hysteresis:

$$m_i(t) = \begin{cases} \Phi(\overline{\mathrm{CQI}}_i(t)) & \text{if } |\overline{\mathrm{CQI}}_i(t) - \overline{\mathrm{CQI}}_i(t-1)| > \delta, \\ m_i(t-1) & \text{otherwise,} \end{cases}$$

where $\overline{\mathrm{CQI}}_i$ is an EMA-filtered CQI, δ a hysteresis threshold, and $\Phi(\cdot)$ the 3GPP CQI→MCS table. This avoids ping–pong and links mobility to T_i and μ_i.

A.2.5 Repair operator and diversity

Budget projection. For a resource class $r \in \{b, c, p, s\}$, if $\sum_i r_i > R_{\max}$, set $r_i \leftarrow r_i \cdot \frac{R_{\max}}{\sum_j r_j}$. **SLA clipping.** If an individual violates an SLA, clip the minimal set of variables that restore feasibility (e.g., increase b_e if $T_e < T_e^{\min}$), then re-project budgets. **Feasibility note.** Projection guarantees $\sum_i r_i \le R_{\max}$; SLA clipping followed by budget projection yields a feasible point under mild monotonicity of KPIs with respect to resources. **Diversity.** After repair, add small zero-mean jitter within feasible bounds to preserve spread before sorting by crowding distance.

A.2.6 Complexity and scalability

Per generation: non-dominated sorting incurs $\mathcal{O}(MP^2)$ complexity, while repair and fitness evaluations each cost $\mathcal{O}(P)$. Scalability levers include: (i) parallel fitness evaluations (per-individual KPI computation), (ii) island-model evolution across CPU/GPU workers, (iii) warm starts from previously obtained $\mathcal{P}^\star$, and (iv) surrogate models for KPI estimation when permissible. As N increases (decision-space dimension $4N$), the population size P should scale sublinearly, with warm starts reused to maintain bounded wall time.

A.2.7 Cost and energy models

Cost. Linear tariffs $C_i = \alpha_c c_i + \alpha_s s_i$ serve as first-order proxies; if operator pricing is tiered, use piecewise-linear segments with SOS2 variables (kept implicit inside the evolutionary fitness). This preserves generality: operators may plug in their actual (possibly non-linear) tariff structures, which are internally approximated piecewise-linearly for the optimizer. The negotiation mechanism and Pareto-front construction remain unchanged; only the cost KPI evaluation is adapted.

Energy. $E_i = \int_0^W P_i(t)\,dt \approx (P_{0,i} + \Delta P\, p_i)\,\tau_i$, where τ_i depends on ρ_i and scheduling (duty cycle). When only steady-state stats exist, $\bar{P}_i$ and τ_i are estimated from recent telemetry windows.

A.2.8 Offer encoding

Each index $k \in \{1, \ldots, K\}$ maps to a JSON blob with per-slice resource vector (b_i, c_i, p_i, s_i) and derived KPIs (T_i, L_i, C_i, E_i) for $\mathrm{e}, \mathrm{u}, \mathrm{m}$. The negotiators only exchange indices and rationales, preserving auditability and repeatability.

A.3 Trust-Score Framework Details

A.3.1 Deviation Components for Satisfaction Score

1. *Deviation from Valid Offers*: This component evaluates whether the agent's proposal falls within the set of admissible, feasible solutions. Let $\mathcal{P} = p_1, p_2, \ldots, p_n$ denote the set of valid proposals in the negotiation space. We define the deviation metric as

$$D_o = \begin{cases} 0, & \text{if } p_{\text{agent}} \in \mathcal{P}, \\ 1, & \text{if } p_{\text{agent}} \notin \mathcal{P}, \end{cases} \tag{A.5}$$

   where p_{agent} represents the offer generated by the agent. Thus, $D_o = 0$ indicates full compliance with feasibility constraints, while $D_o = 1$ assigns the maximum penalty for producing an invalid proposal, effectively capturing a violation of negotiation boundaries.

2. *Deviation from Intent*: This component measures how closely the agent's selected proposal aligns with its declared objectives, particularly regarding solution optimality. To quantify this alignment, we employ the Normalized Generational Distance (NGD) metric, which evaluates the proximity between the agent's achieved outcome and a reference set of Pareto-optimal solutions, denoted by $\mathcal{S}^* = s_1^{,} s_2^{,} \ldots, s_k^*$.

   Let $\mathbf{v}_{\text{agent}} = [v_1, v_2, \ldots, v_m]$ represent the agent's KPI vector corresponding to its chosen proposal. The deviation from intent is defined as:

$$D_i = \min\left(1, \text{NGD}(\mathbf{v}_{\text{agent}}, \mathcal{S}^*)\right) \tag{A.6}$$

where the NGD metric is computed as:

$$\text{NGD}(\mathbf{v}, \mathcal{S}^*) = \frac{1}{|\mathcal{S}^*|} \sqrt{\sum_{s^* \in \mathcal{S}^*} \min_{s \in \mathcal{S}} d(\mathbf{v}_{\text{norm}}, \mathbf{s}^*_{\text{norm}})^2}. \tag{A.7}$$

Here, vnorm and s*norm represent the normalized KPI vectors, ensuring comparability across different scales. The use of the minimum function in Equation A.6 constrains the deviation score to the range [0,1], maintaining interpretability while effectively penalizing substantial deviations from the optimal intent.

3. *Deviation from Mediator*: This component quantifies how closely the agent's proposal aligns with the mediator's recommendation, reflecting the agent's willingness to cooperate toward consensus. The deviation is defined as:

$$D_m = \begin{cases} 0 & \text{if } p_{\text{agent}} = p_{\text{mediator}} \\ 1 & \text{if } p_{\text{agent}} \neq p_{\text{mediator}} \end{cases} \tag{A.8}$$

where p_{mediator} denotes the proposal recommended by the mediator. A value of $D_m = 0$ indicates complete agreement with the mediator's recommendation, while $D_m = 1$ represents full disagreement, signaling resistance to convergence during the negotiation process.

A.3.2 Behavior of the Agents

Factual Accuracy Factual accuracy measures the correctness of numerical claims made by agents within their decision rationales. Natural language processing techniques are employed to extract quantitative statements related to KPIs and validate them against corresponding ground truth data.

Let $\mathcal{C} = c_1, c_2, \ldots, c_k$ denote the set of numerical claims extracted from an agent's rationale, where each claim $c_i = (m_i, v_i)$ consists of a metric type m_i and an asserted value v_i. For each claim, we compute the relative error with respect to the ground truth value v_i^* as:

$$\epsilon_i = \frac{|v_i - v_i^*|}{v_i^*}. \tag{A.9}$$

We categorize hallucinations according to the magnitude of the relative error ϵ_i as follows:

- *None*: $\epsilon_i \leq 0.15$ (within 15
- *Minor*: $0.15 < \epsilon_i \leq 0.5$ (penalty: 0.1)
- *Major*: $0.5 < \epsilon_i \leq 1.0$ (penalty: 0.3)
- *Severe*: $\epsilon_i > 1.0$ (penalty: 0.6)

The factual accuracy score F quantifies an agent's overall numerical reliability by combining relative accuracy with hallucination penalties and is defined as:

$$F = \max\left(0, \frac{1}{|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|}(1-\epsilon_i) - \sum_{i=1}^{|\mathcal{C}|}\text{penalty}_i\right), \tag{A.10}$$

where penalty_i is the hallucination penalty assigned to claim c_i based on its error category. The score is normalized to a maximum of 1.0 and lower-bounded at zero to prevent negative values, ensuring interpretability and comparability across agents.

Logical Consistency Logical consistency assesses the internal coherence and reasoning quality of an agent's communication. We evaluate whether the rationale demonstrates structured thinking, aligns with the agent's objectives, and avoids self-contradictions.

The logical consistency score L is evaluated according to the following criteria:

- *Logical connectors*: Identification of reasoning indicators (e.g., "because," "therefore," "since"), which reflect structured and coherent argumentation.
- *Goal alignment*: Verification of references to negotiation objectives or strategic intent, indicating that reasoning is purpose-driven.
- *Contradiction detection*: Detection of conflicting statements, such as simultaneous positive and negative assertions about the same factor, which undermine logical integrity.

Formally, the score is defined as:

$$L = \min\left(1, \max\left(0, 1 - P_c + B_s + B_g\right)\right) \tag{A.11}$$

where P_c is the contradiction penalty, reducing the score for detected inconsistencies, B_s is the structure bonus, awarded for the presence of logical connectors and coherent reasoning, and B_g is the goal bonus, assigned when the rationale explicitly references objectives or strategic considerations. This formulation ensures that logical consistency is rewarded for clear, structured, and goal-driven reasoning while penalizing internal contradictions.

Semantic Coherence Semantic coherence assesses the relevance, expressiveness, and structural quality of an agent's communication within the negotiation domain. This metric evaluates whether agents exhibit a clear understanding of domain-specific content and articulate their reasoning with clarity, contextual awareness, and linguistic diversity.

The semantic coherence score E integrates the following components:

- *Domain terminology (dt)*: the use of appropriate technical terms and negotiation-specific vocabulary.

Slice	Type	Tput (Mbps)	Lat. (ms)	Cost ($)	Energy (W)	Full Intent in Natural Text by the Human Stakeholder
Media-Flex	eMBB	60	10	200	100	*My slice use case is eMBB for an Airport lounge 4-K pipe, and it is crucial to maximize throughput. Push for the **highest throughput**, to maximize QoS/QoE of our users.*
Factory-Ops	URLLC	5	2	200	100	*Our slice is used by robots that need sub-5 ms motion control and thus an URLLC guarantee. So, **minimize latency** as the ultimate goal. Moreover, **high throughput** is also important.*
IoT-Sense	mMTC	20	10	50	100	*My slice is providing coverage to Smart-city sensors and we have an mMTC case. I need you to prioritize cost and aim for the **most cost-efficient** solution, as our budget is limited.*
Mediator	–	–	–	–	–	*My network goal is to serve and **find a balance** between the stakeholder's slices objectives. However, prioritize **minimum energy consumption** to align with the country's regulations.*

Table A.2: In Phase P_A the human stakeholders express their slice intents in full natural-language form with associated constraints.

- *Linguistic diversity (ld)*: the richness of vocabulary and avoidance of repetitive phrasing.
- *Structural quality (sq)*: the sentence variety, appropriate length distribution, and overall readability.

The final semantic coherence score is computed as:

$$E = \min(1, \mathrm{dt} + \mathrm{ld} + \mathrm{sq}). \tag{A.12}$$

A.4 Use Case Validation: Detailed Technical Details and Messages Exchanged

Table A.4 summarizes the KPIs of the agreed SLAs for each phase, as accepted by all agents; Offer *2** was ultimately selected in Phase $\mathrm{P_A}$, with its underlying resource allocation detailed in Table A.5. The complete negotiation transcript for this round is shown in Figure A.1.

Fig. A.2b shows physical PRB utilization for Factory-Ops, comparing Agoran's dynamic negotiation with conventional static configurations. Dynamic allocation reduces PRB usage by 15,888 PRBs (24%) during periods of low demand and adds only 10,422 PRBs (16%) during high-demand intervals. This results in an overall net PRB saving of 8.3% over the full trace. Moreover, this flexible allocation enables targeted, on-demand improvements in Factory-Ops throughput, as shown in Figure A.2a. When the slice switches to eMBB priority in Phase $\mathrm{P_D}$, throughput increases by up to 66%—from 35 Mbps in Phase $\mathrm{P_A}$ to 58 Mbps in Phase $\mathrm{P_D}$. Finally, Figure A.2c compares the latency of the Media-Flex slice under negotiated and static SLAs. Agentic negotiation reduces the median RTT by 73.4%, from 5.7 ms to 1.5 ms, once the slice transitions to URLLC.

ID	Application	Intent	Tput [Mbps]	Lat. [ms]	Cost [$]	Energy [W]
1	Media-Flex	eMBB	60.72	5.66	61.52	13.39
	Factory-Ops	URLLC	34.82	1.49	133.14	12.72
	IoT-Sense	mMTC	38.14	5.45	2.19	0.05
	Overall		**133.68**	**12.60**	**196.85**	**26.16**
2*	Media-Flex	eMBB	60.02	5.67	63.28	10.77
	Factory-Ops	URLLC	35.40	1.48	132.35	12.08
	IoT-Sense	mMTC	38.26	5.45	2.19	0.00
	Overall		**133.68**	**12.60**	**197.83**	**22.84**
3	Media-Flex	eMBB	60.06	5.67	68.16	12.84
	Factory-Ops	URLLC	35.52	1.48	133.94	12.92
	IoT-Sense	mMTC	38.11	5.45	1.64	2.46
	Overall		**133.68**	**12.60**	**203.74**	**28.22**

Table A.3: KPIs of the three Pareto-optimal offers for Experiment Phase P_A, conducted under favorable channel conditions (MCS = 28). Each offer reflects different trade-offs across the three slices. Offer **2*** was ultimately selected by unanimous agent consensus. The "Overall" rows report aggregate KPIs across all slices.

Exp	Application	Intent	Tput	Lat.	Cost	Energy
PA	Media-Flex	eMBB	**60**	5.7	$63	10.8 W
	Factory-Ops	URLLC	35	**1.5**	$132	12.1 W
	IoT-Sense	mMTC	38	5.5	**$2**	0.0 W
PB	Media-Flex	eMBB	**11**	8.8	$84	3.8 W
	Factory-Ops	URLLC	7	**3.4**	$27	2.5 W
	IoT-Sense	mMTC	7	7.5	**$0**	2.5 W
PC	Media-Flex	eMBB	0	∞	$0	0 W
	Factory-Ops	URLLC	0	∞	$0	0 W
	IoT-Sense	mMTC	0	∞	$0	0 W
PD	Media-Flex	**URLLC**	38	**1.5**	$2.7	1.7 W
	Factory-Ops	**eMBB**	**58**	5.7	$3	26.0 W
	IoT-Sense	mMTC	38	5.5	**$0**	1.3 W

Table A.4: KPIs of the *agreed* SLA for each experiment phase. These SLAs reflect the capabilities of our testbed, which employs an OAI gNB with 40 MHz bandwidth and supports a maximum overall throughput of 133.7 Mbps.

ID	Application	Intent	PRBs [%]	CPU [units]	Power [W]	Storage [MB]
2*	Media-Flex	eMBB	44.9	5.6	10.8	26.1
	Factory-Ops	URLLC	26.5	21.7	12.1	44.5
	IoT-Sense	mMTC	28.6	0.0	0.0	1.1
	Overall		**100**	**27.3**	**22.9**	**71.7**

Table A.5: Underlying resource allocation corresponding to the selected Offer **2*** in Experiment Phase P_A.

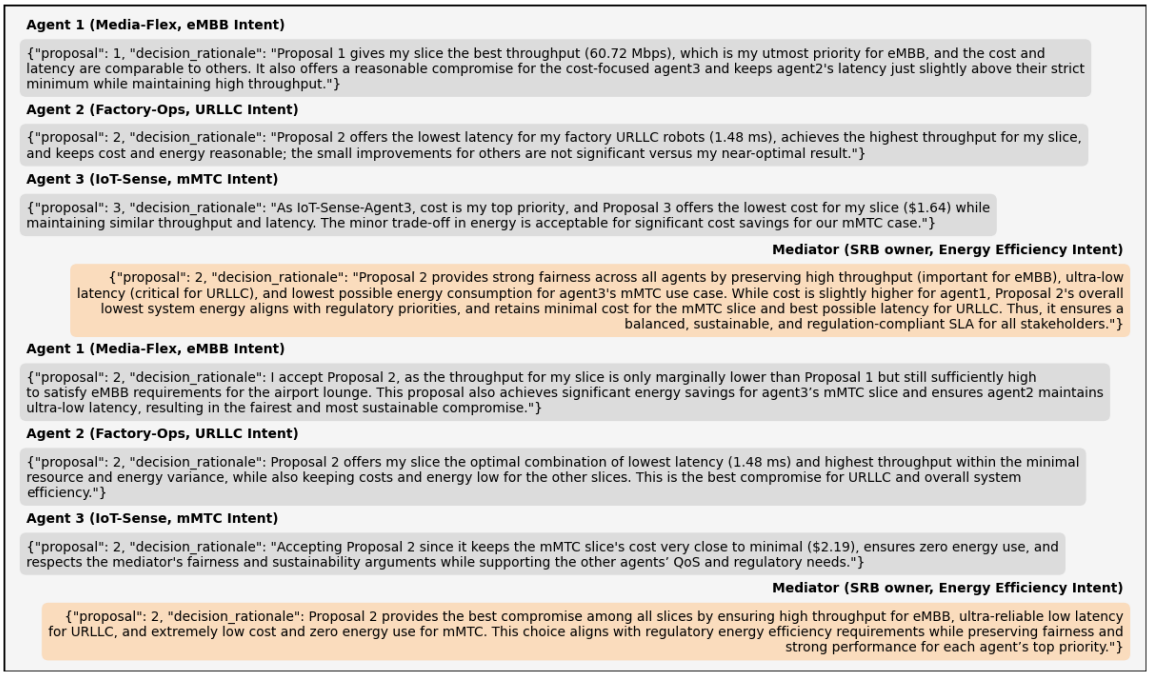
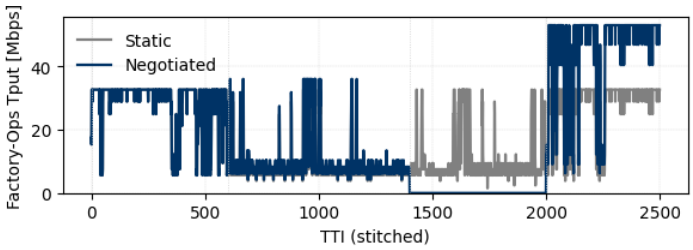
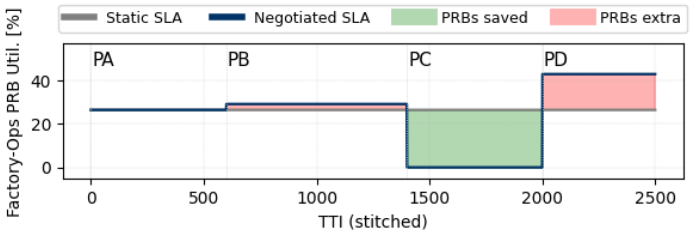


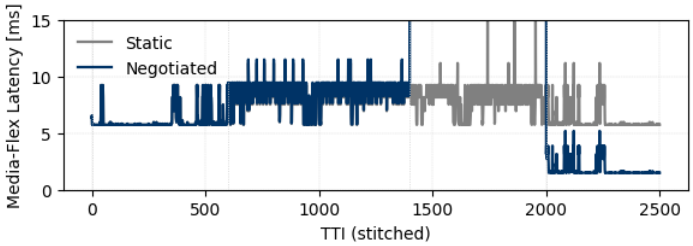
Figure A.1: Snapshot of multi-agent negotiations during Experiment Phase P_A, powered by *GPT-4.1*. The NSC Agents represent three stakeholders, Media-Flex, Factory-Ops, and IoT-Sense, in the Agoran marketplace. The agents negotiate over the three Pareto-optimal offers generated by the multi-objective optimizer (Table A.3), ultimately reaching consensus on *Offer 2* as the most balanced solution. Agreement is achieved in a *single round*, excluding the initial expression of intent.



(a) Factory-Ops throughput



(b) Factory-Ops PRB utilization



(c) Media-Flex latency

Figure A.2: Negotiated versus static SLA. Dynamic reallocation increases throughput, releases unused PRBs (green), and allocates additional PRBs only when beneficial (red), while achieving lower latency when Media-Flex becomes URLLC-critical in Phase P_D.

A.5 Technology-agnostic design.

While our examples focus on SLA formulation in Open RAN, the same tripartite governance and agentic negotiation principles extend to broader multi-service environments. These include spectrum brokering, slice life-cycle management, fault mitigation, and energy optimization. In AI-RAN deployments involving numerous stakeholders with heterogeneous applications (xApps, rApps, dApps [30]), Agoran serves as a neutral platform to reconcile intents, allocate resources, enforce policies, and resolve conflicts in a consistent and auditable manner. None of the design choices bind Agoran to a specific standard. The consensus channel relies on a vendor-neutral JSON schema, and substituting the RICs with an AI-RAN or any other control fabric merely requires replacing the southbound adapter, without modifying the underlying logical contract.