Ericsson White Paper
GFTL-22:001015 Uen
September 2022

ERICSSON

Cognitive Reasoning for 5G Network Lifecycle Management

Importance of continuous feedback loops for efficient 5G planning, design, fulfillment and assurance

Content

Introduction

With 5G, service providers confront the combined challenges of the huge number of devices, large variety of service requirements (traditional best effort enhanced Mobile Broadband [eMBB] as well as Service Level Agreement [SLA]-based Ultra-Reliable Low-Latency Communication [URLLC] services), and dynamic changes in a complex network environment. Communication Service Providers [CSPs] need to tackle emerging issues within strict deadlines, and proactively introduce new technology to ensure positive business outcomes and market leadership.

The current approach to address these challenges comprises different scopes of lifecycle management such as network, service, and resource/software. These lifecycle scopes consist of different phases that are implemented largely independently by different teams, with several handover points. While the modular approach leads to a separation of concerns and ease of management, it cannot handle the staggering scale and stringent requirements of 5G networks and services.

In this white paper, we specifically address the drawbacks of the phased approach to network lifecycle management by 1) automating the feedback and reconfiguration information among network life cycle phases avoiding unacceptable delays in deployment and assurance of services, 2) continuous monitoring and reconfiguration to address dynamic changes in service requirements and network state, and 3) adaptability to technology evolution. We propose an automated methodology based on a cognitive framework that includes machine reasoning and machine learning based agents. This will enable service providers to handle the diversity, scale, and variability of 5G network and service orchestration.

Phased approach in current network LCM and challenges

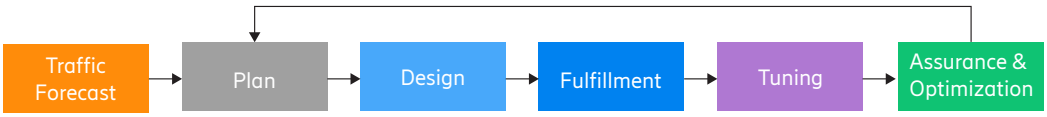

Figure 1. Traditional network lifecycle management

The current phased approach for onboarding a new customer or service with 5G/beyond 5G is typically a lengthy process and involves the following phases (see Figure 1):

- **Requirements gathering (6-12 weeks):** This is a manual process involving detailed discussions with stakeholders to capture the functional and non-functional requirements.

- **Planning (12 weeks):** Network planners use a diverse set of tools for traffic estimation, capacity budgeting, spectrum evaluation, and site surveillance [9].
- **Design and simulation (12 weeks):** The plan is converted to high-level design documents (e.g. cell location, spectrum, edge compute capacity) that may be implemented through low-level orchestration (e.g. physical resource block partitions, Kubernetes pod placement).
- **Fulfillment (12-24 weeks):** The design has to be implemented (preferably) over available sites. In case there are no suitable sites or carriers available, there would be further delays in timelines.
- **Tuning (12 weeks):** Another lengthy process after fulfillment is tuning the network to meet realistic traffic patterns. Extended tuning may negate the plan and design produced in the preceding phases.
- **Assurance and optimization:** The assurance phase involves SLA compliance evaluation, quality of service (QoS) flow management, and network slice life cycle management. The assurance loop continuously monitors, configures and maintains the QoS performance levels as mandated by SLAs. This loop performs root cause analysis of potential problems and resolves them with minimal user impact, with the use of automation tools.

The various tasks in the phases are handled by individual teams within CSPs [8]. The tasks, data and knowledge artifacts, tools, and processes can differ a lot among the phases. With the advent of Artificial Intelligence [AI], it is envisaged that many tasks within the phases would be solved using data-driven, Machine Learning [ML]-based algorithms [2].

Drawbacks

The separation of concerns among the phases leads to easier management and independent evolution. However, there are drawbacks arising because of the phased architecture itself.

First, there is either no feedback from subsequent phases or when it exists, it is manual, sparse, and incurs great delay. For example, feedback from the fulfillment phase could prompt a replanning or redesigning of the network before it is operational. In the absence of this feedback, a replanning or redesign of the network triggered during the assurance phase will lead to disruption/degradation of deployed services and possible violation of SLAs. It is thus necessary to define and implement interphase closed loops.

Second, the architecture is not suited to admitting new requirements on-demand and handling drifts in traffic patterns, which would require continuous replanning, design, tuning, and assurance.

Third, the phased architecture is not agile enough to factor in the technology evolutions such as automation capabilities, virtualization, and interacting closed loop control that can greatly enhance network performance and management.

Proposed changes for network LCM

We propose a number of enhancements to the current network life cycle management (LCM) architecture to rectify the above-mentioned drawbacks.

Interphase feedback loops

We propose multiple new closed loops allowing continuous feedback between phases in the life cycle (see Figure 2). For instance, any deviations in low-level design resource constraints can provide feedback for the planning phase to acquire more resources. Since the fulfillment phase involves the actual implementation of the low-level design, any deviations in implementation must provide quick feedback to the design phase to spin out an alternative design. During the tuning phase, if there is a drift in the traffic pattern, it can be fed back to the traffic forecast phase to consider retraining the forecast models that impact the planning and design phases. Similarly, a deviation in assurance can trigger re-fulfillment with new slice templates. Physical site inspections and changes may be needed when all other alternatives fail; the continuous feedback from the various life cycle phases would minimize the manual operations that are costly and time-consuming.

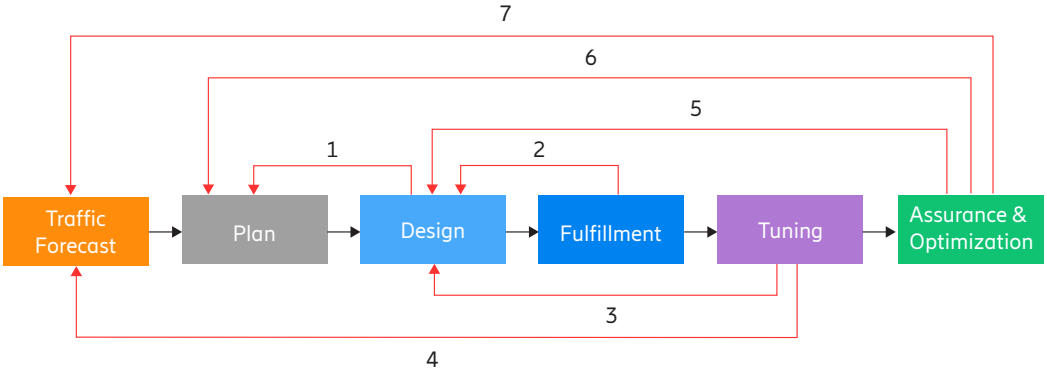

Figure 2. Continuous feedback loops in network design and optimization

Figure 2 consists of the following feedback loops:

1. Deviations in high/low level designs triggering re-planning.
2. Deviations in fulfillment triggering re-design
3. Tuning of the system triggering re-design (rather than significant change during tuning).
4. Tuning triggering re-estimating the traffic pattern forecast.
5. Assurance and optimization triggering a re-design.
6. Assurance and optimization triggering a re-planning phase.
7. Assurance and optimization triggering a change in traffic forecast.

Note that the implementation of a subset of the loops would depend on the mobile service provider. A few of the phases such as fulfillment, which is a deployment of the design, may not have as much feedback as the other phases.

Continuous reception and processing

The signing-off between phases can benefit from common ontologies and artifacts, which can be implemented as an enhancement of the current network LCM. However, a fast, automated response to any type of change—whether it is the traffic pattern, functional or service requirements, or technology evolution providing different and better solutions—needs a continuous trigger of the phases, which cannot be carried out without a major paradigm shift.

Figure 3 shows the external context (the vertical loops showing interaction with service and infra layers) within which the network LCM must operate. It shows the following additional life cycle loops with different scopes:

1. user experience automation (customer experience management)
2. service automation (service configuration and management)
3. automation tools for orchestration scripts and management of AI/ML-based agents (model update and training)
4. software LCM as part of continuous delivery and deployment (not represented here for simplicity)

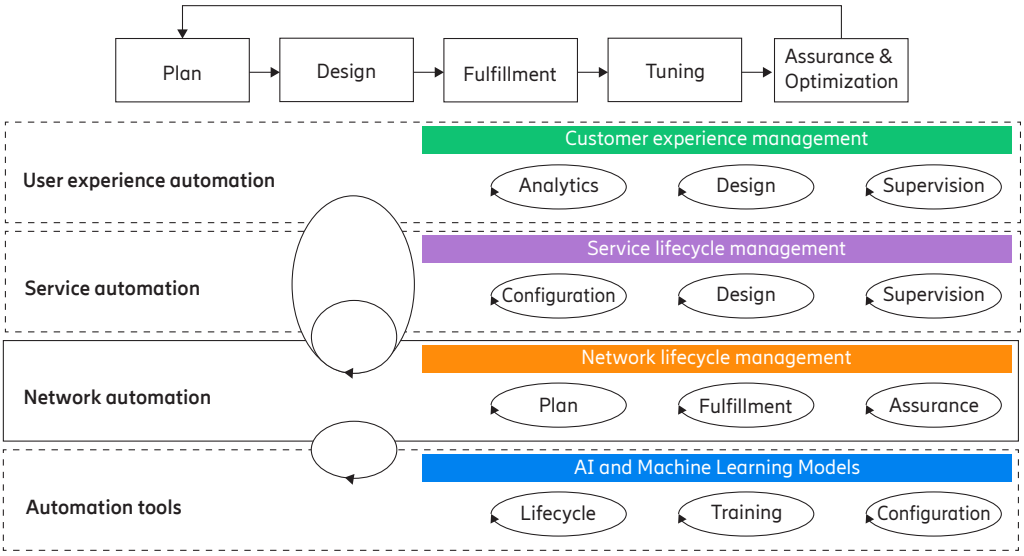

Figure 3. Interacting Control Loops

Interaction of the user experience and service automation loops with the network LCM loop takes place via the planning and assurance phases. The automation tools loop that provides high-quality tools to the entire system interacts with the network LCM loop in all the phases.

As we envision continuous planning, design, assurance, and optimization, multiple phases and automation cycles may be handled simultaneously. A higher level can be inducted into the system where the underlying system can affect phase selection, agent selection for tasks, and causal propagation. This would increase reuse and optimize the use of knowledge across the phases.

Adapting to new scenarios

The proposed LCM will be cognizant of understanding and handling new situations and integrating new technologies seamlessly. For example, virtualization of network function deployment in Radio Access Network [RAN] and core would change the capacity planning and design process. As the aggregated capacity would need to be planned, virtualization would point to the scaling up/down of nodes and function deployment/migration as a result of changes in traffic patterns. This would impact the planning, design, validation, and fulfillment stages.

With the increasing capabilities and complexities of AI agents, the boundaries between these phases could diminish. For instance, the tuning phase could disappear with continuous assurance/optimization and feedback to the design phase through the automated assurance loops. This would lead to the most optimal reuse of knowledge and tools for network LCM.

Cognitive framework-based network LCM

A cognitive framework-based LCM can solve the three challenges of fast feedback among different phases, continuous response and adaptability to new scenarios. Using the cognitive reasoning and intent handling capabilities of the cognitive framework (Cognitive processes for adaptive intent-based networking), as an umbrella we propose to link the different phases of network LCM.

Cognitive Framework

A cognitive framework [6] consists of three essential components: A knowledge base, a reasoning engine, and an agent architecture. As presented in Figure 4, the knowledge base contains the ontology of intents along with domain-specific knowledge such as the current state of the system. The domain-independent reasoning engine serves as the central coordinator function and uses the knowledge graph to orchestrate a number of registered agents for finding solution actions, evaluating their impact, and ordering their execution. Finally, the agent architecture allows any number of models and services to be used. Agents can contain machine-learned models or rule-based policies, or implement services needed in the cognitive reasoning process.

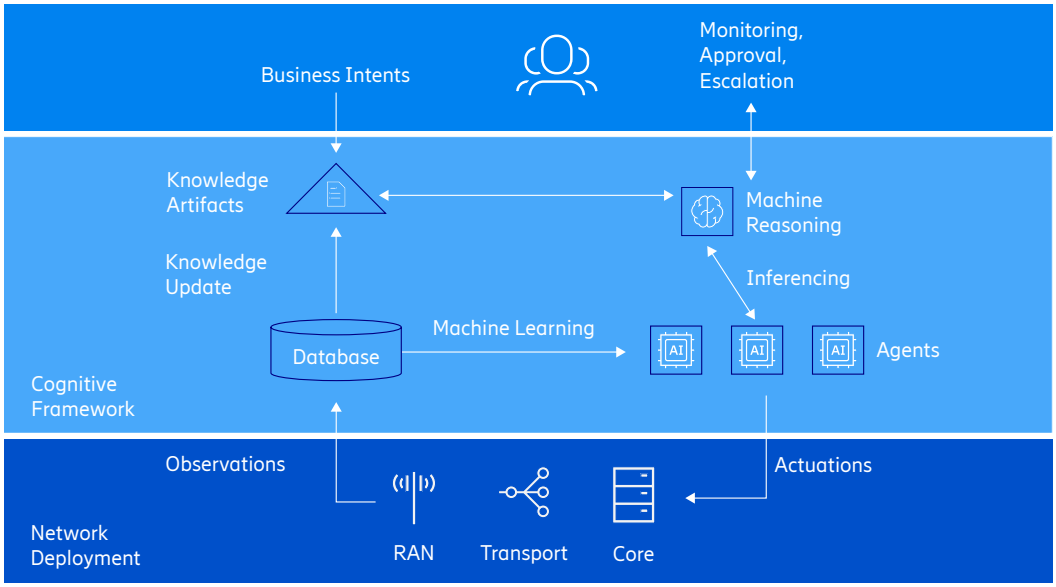

Figure 4. Cognitive Framework

Cognitive solution for network LCM

A cognitive framework allows many possible ways of interacting with the external context (service and infrastructure layers) as well as defining and orchestrating the loops. An approach is taken where each phase (forecasting, design, fulfillment, tuning, assurance) is an agent. The cognitive framework orchestrates these phases (Figure 5). Each of the phases may have an intent handling loop with distinct knowledge entities, timelines, and actions. Note that this is the first approximation of the implementation and may be integrated through a single agent without distinction between phases (described in Section 4.3.7).

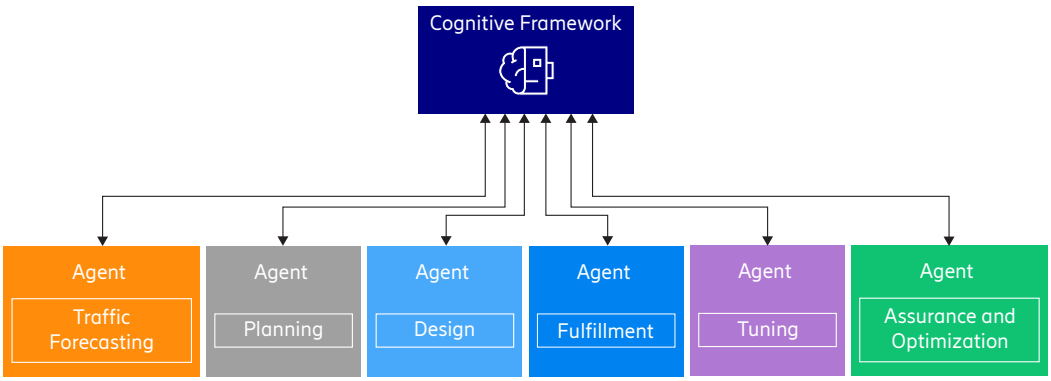

Figure 5. Network LCM agent's interaction with a cognitive framework

Figure 6 provides a view of the interaction between agents, cognitive intent handling, and network design/actuation. Intents can arrive in the system during plan/design time (for example, provision of the network for 5G capacity in a given geographical area), fulfillment (for example, ensuring a set of SLA-based user equipment (UE) receive sufficient QoS) or assurance (for example, provision for a new QoS flow). The intents are multi-objective and can target functional (latency, throughput), business (average revenue per user, customer tiers), and efficiency (Physical Resource Block [PRB], compute utilization) requirements. There are six agents defined in order to handle various facets of this life cycle. These agents rely on the cognitive framework intent handling features such as the knowledge base, reasoning, and state vector management to map intents to expectations.

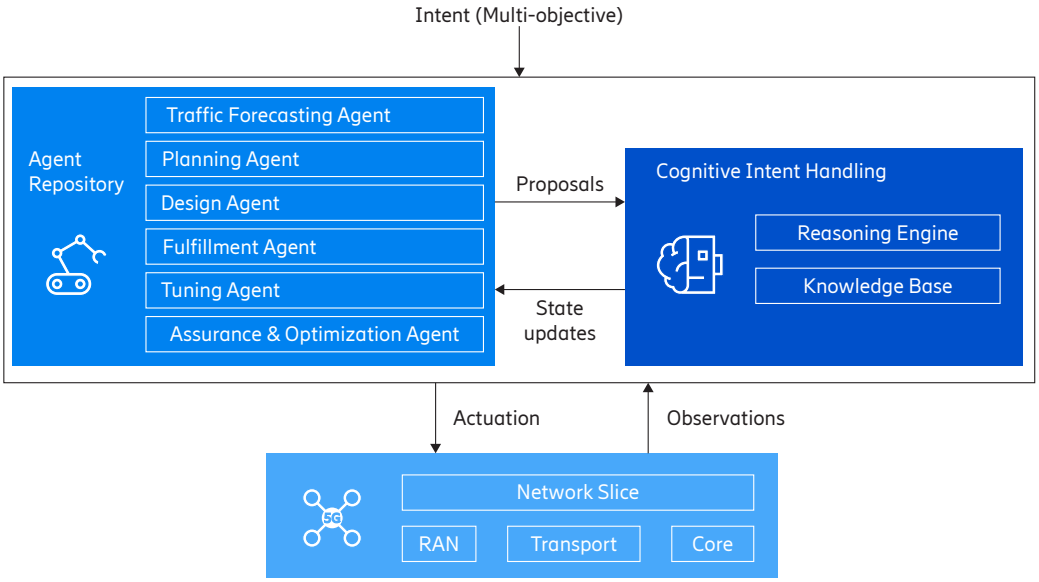

Figure 6. Interaction between cognitive intent handling and intelligent agents

The cognitive framework offering could come with multiple manifestations dependent on 3rd Generation Partnership Project [3GPP] sttandards, Operations Support Systems [OSS]/Business Support Systems [BSS] technologies, and products. In the initial setup, the cognitive framework would interface through Application Programming Interface [API] calls to various modules with OSS/BSS, for instance, policy execution, order management, and product catalog. Going forward, these cognitive features would be integrated natively within OSS/BSS offerings, which is compatible with the AI-native vision of 6G (AI native architectures).

As described in [6], the cognitive framework would also be hierarchical. There can be a central cognitive framework in the case of limited network topology. For larger, more complex networks, this can be implemented as multi-domain RAN, transport, and core cognitive framework implementations with a central coordinator. The deployments of the knowledge base, intent handling, and machine reasoning would be done hierarchically to aid in scale and modularity.

Description of AI agents for 5G phase lifecycle management

This section looks at the various phases of the 5G deployment life cycle and how the interplay between machine reasoning and learning solutions (AI planning, knowledge graphs, reinforcement learning) may be implemented.

Traffic forecast phase

In order to forecast the requirements of the 5G radio network (RAN, transport, core), the forecasting agent must consider multiple factors. These include specification of the service (trend), distribution of users, growth models, and coverage of service (nationwide, region, clusters) [5].

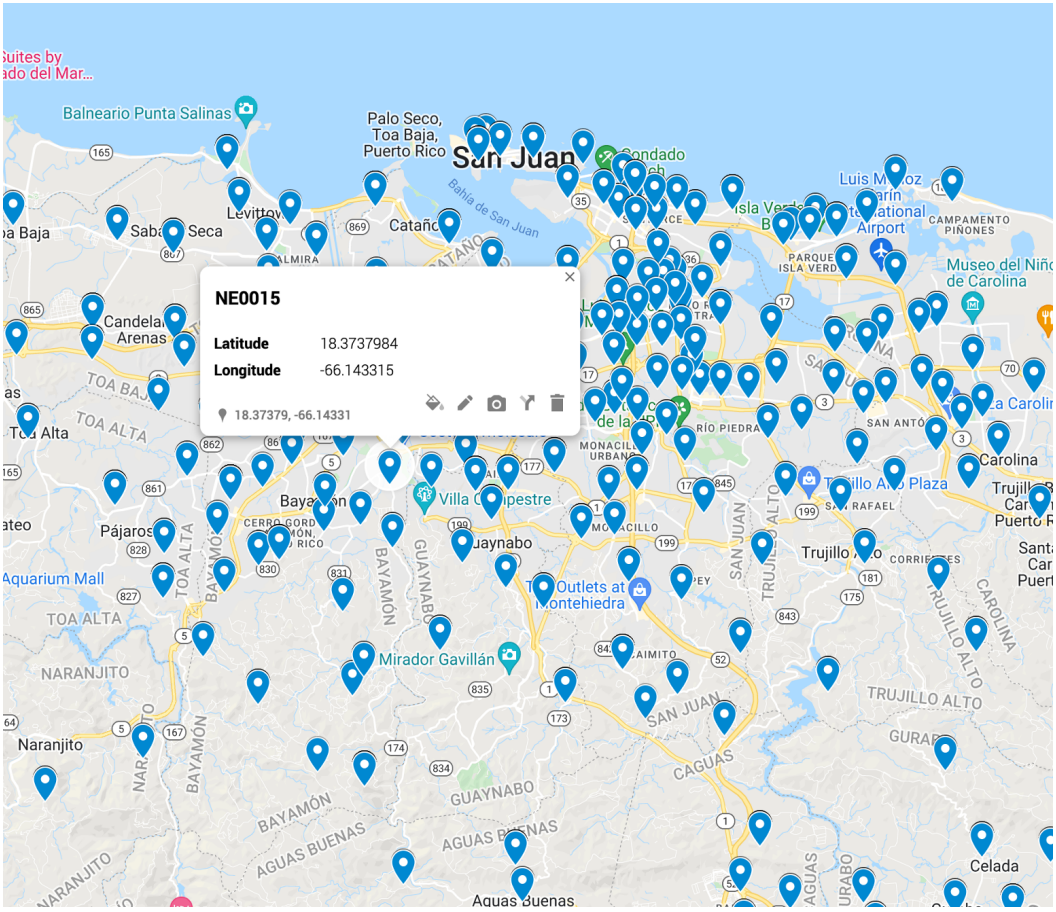

Figure 7. Current 4G deployment locations for a service provider

As most service providers have 4G sites deployed (Figure 7), one strategy would be to reuse datasets from an existing mobile network (month-on-month growth in user traffic, observed throughput). This would then be used to estimate the forecasted growth over a specific time period (for example, 24 months). As the traffic only consists of emBB devices in 4G, the inorganic growth of new service traffic can be estimated through mobile service provider business plans for service deployment.

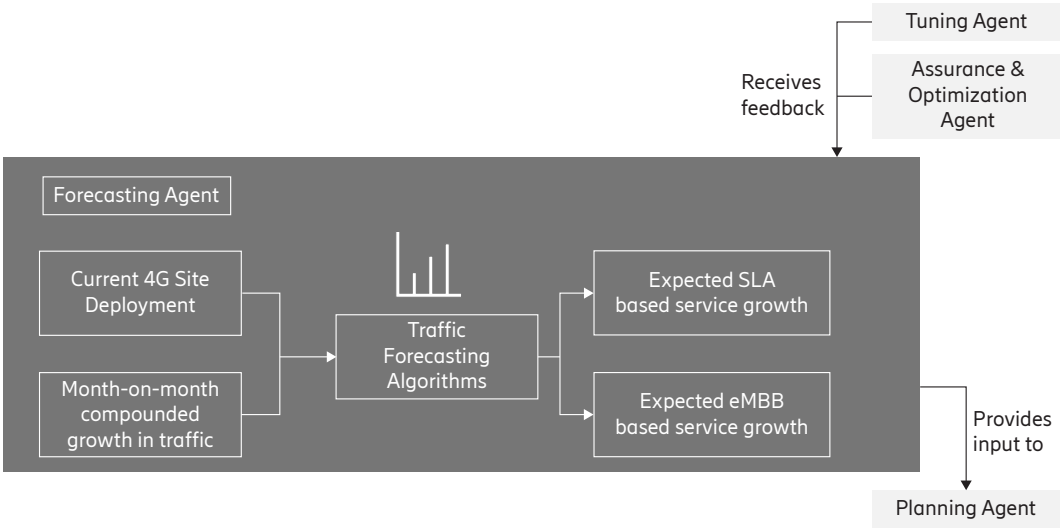

Figure 8. Traffic forecasting agent

Figure 8 represents the actions performed by the forecasting agent. Given historical data on existing services in given geographical areas, business plans for a time horizon, and the live traffic status coming from Tuning and Assurance & Optimization agents, it provides predictions for traffic for different services.

Planning phase

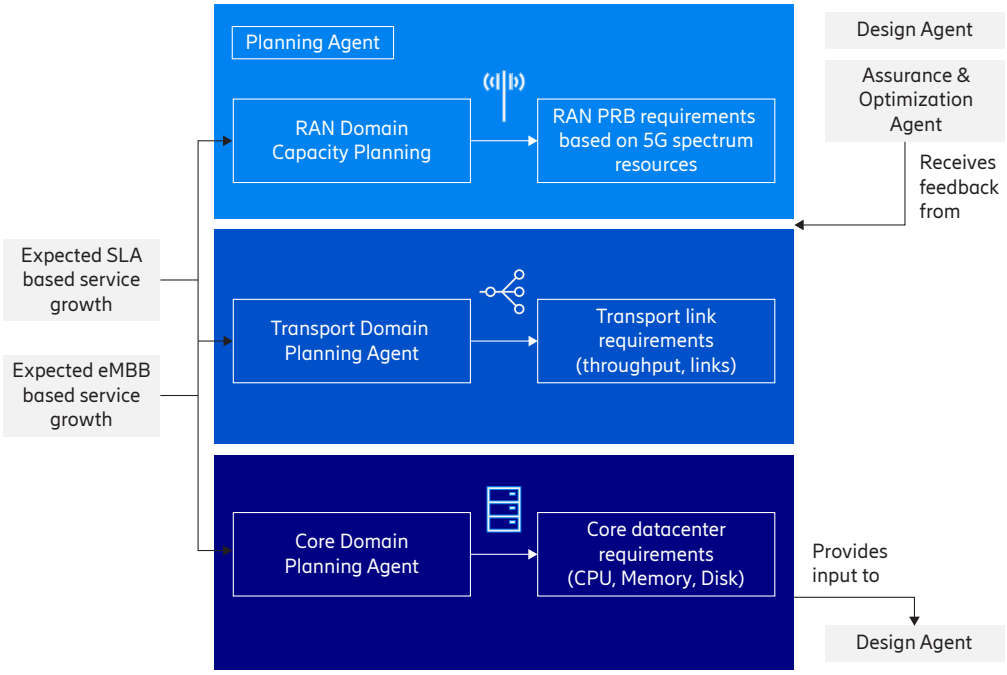

Figure 9. Planning agent

Given the forecasted traffic, the planning phase determines the additional capacity needed at the RAN, transport, and core subnets. Figure 9 provides an overview of the planning agent. Note that this planning process makes use of multi-objective requirements including service Key Performance Indicators [KPIs], resource efficiency, cost, energy, and revenue models.

Design phase

This process involves a high-level design (if the available resources at RAN, transport, and core can meet resources) and a low-level design (host-level information used to configure compute communication nodes).

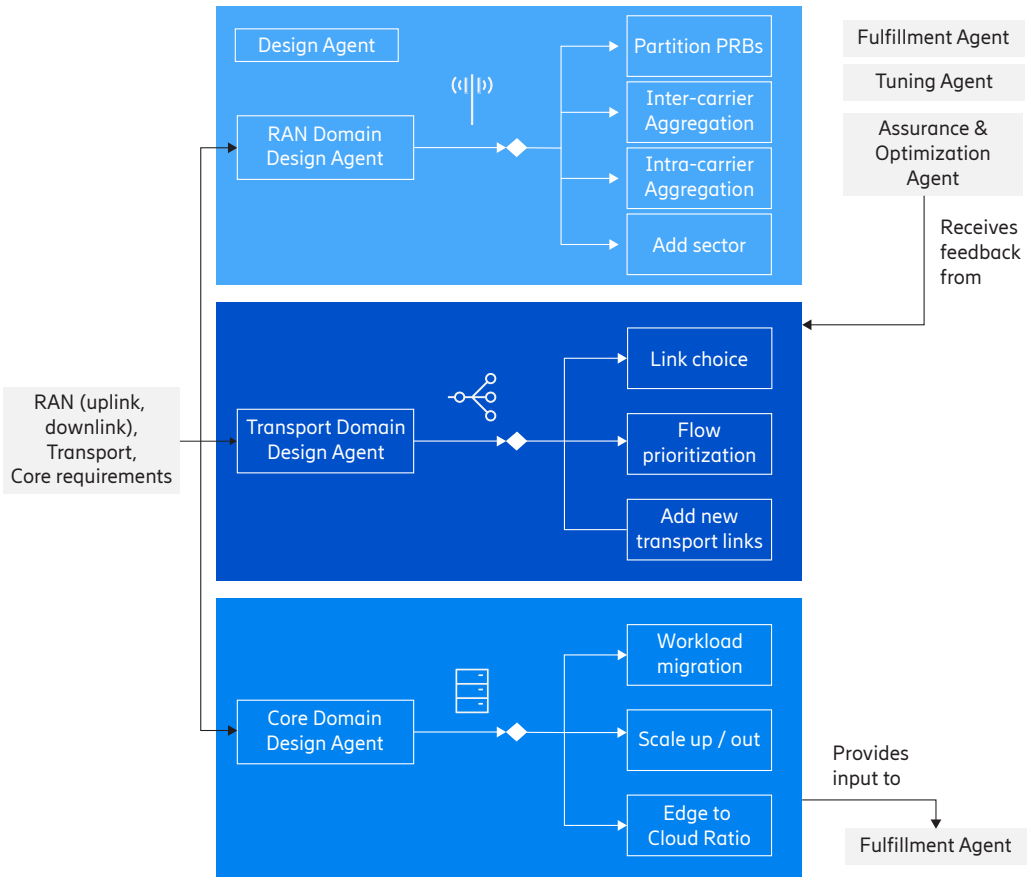

Figure 10. Design agent

Figure 10 describes the design agent functionalities. For instance, in the case of RAN design, there are multiple possible design outputs, which are:

1. In the case of sufficient radio resources, the repartitioning and allocation of resources.
2. Or else, performing intra/inter-band carrier aggregation or load balancing to add additional radio resources.

3. Or else (1 and 2 both not satisfactory), adding an additional antenna sector.
4. Or else (1, 2, 3 not satisfactory), requesting an additional cell site.

Note that some of these operations are lengthy (new cell site deployments) and must be avoided if alternatives exist.

An example design output for a cell NE20605, which proposes carrier aggregation and then partitioning is provided below:

```
3  (NEW_CARRIER_AGGREGATION_N5_50MHZ   NE20605  N5   DL   USER_EMBB   M4T4R)
4  (ALLOCATE_PARTITION_N5 NE20605  UL   USER_EMBB   M4T4R   N5)
5  (ALLOCATE_PARTITION_N5 NE20605  DL   USER_SLA    M4T4R   N5)
6  (ALLOCATE_PARTITION_N5 NE20605  DL   USER_EMBB   M4T4R   N5)
```

Similar granular plans may be done for the transport subnet (links, priority) and core subnet (scaling, workload migration).

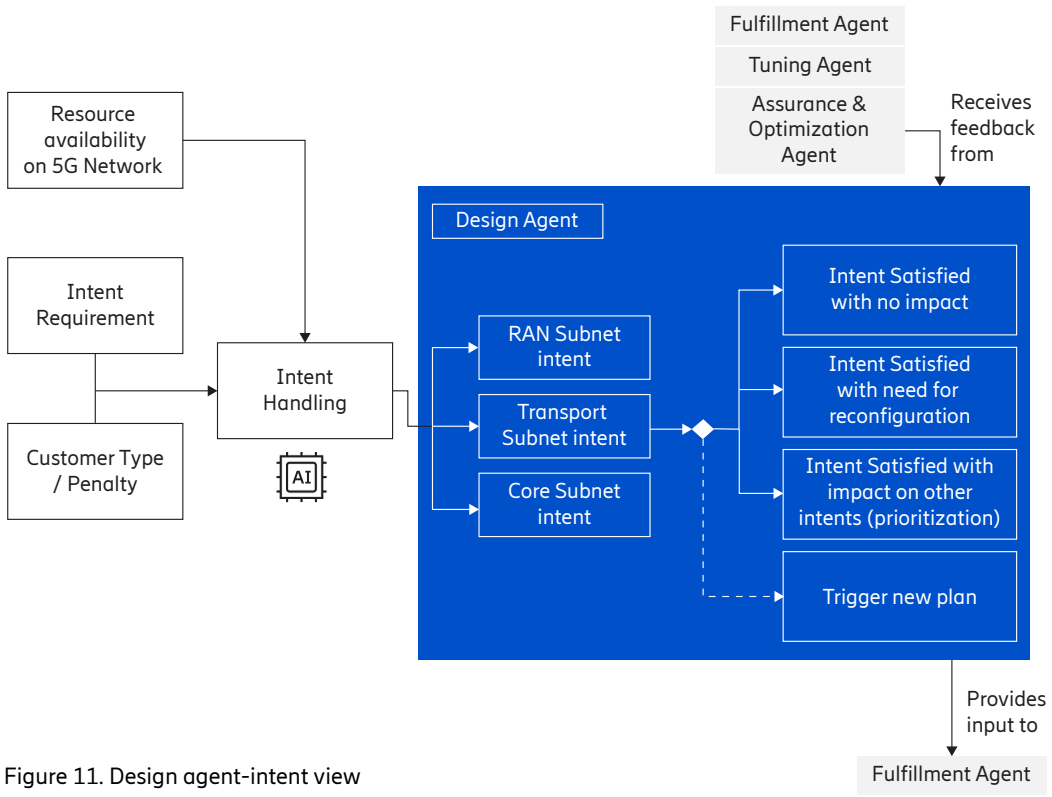

Figure 11. Design agent-intent view

Given pre-deployed services (and associated network slices), the design phase must allocate the deployment of new services without significant impact (penalties, costs) on existing services. Figure 11 provides the intent handling view of the design agent. Given the intent requirement of service and the current resource capacity, the agent can allocate resources with one of the following outcomes for the intent: (i) no side effects, (ii) need for reconfiguration of system, and (iii) impact on other intents (need for prioritization). Under extreme cases, this phase of the life cycle can also trigger a replanning of the deployment.

Fulfillment phase

The design output may be deployed by fulfillment agents to meet the 5G network and service requirements. There may be shorter horizon fulfillment tasks (performing carrier aggregation) or longer horizon tasks (adding a new cell site). The fulfillment phase gives feedback to the design phase.

Tuning phase

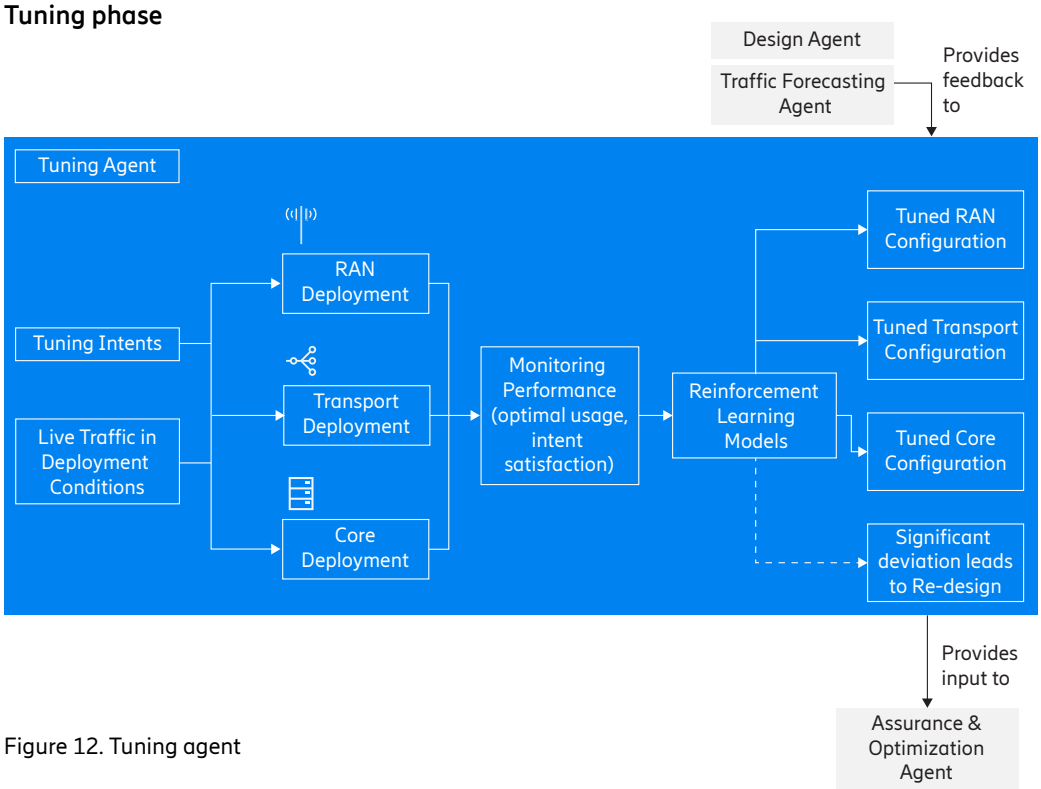

Figure 12. Tuning agent

A tuning agent (Figure 12) can interface most effectively between fulfillment and assurance by monitoring performance over live traffic and optimally configuring the subnets. Note that the intents at this phase could be test intents used to monitor the efficacy of the fulfillment. One technique to be considered would be reinforcement learning, wherein training could

be carried out over the tuning period to suggest optimal configurations. In case the tuning phase does not yield sufficiently optimal outputs, this can trigger a redesign or replan of the deployment. The tuning phase can also provide feedback on the traffic mix assumptions (eMBB vs. SLA) that were used in the traffic forecasting and planning phases.

Assurance and optimization phase

Assurance of QoS flow performance to various tiers of customers involves managing the network slice life cycle (creation, scaling, and deletion), optimal prioritization and resource allocation, QoS deviation monitoring, and intelligent action execution [3]. In this case, the agent (Figure 13) has a view of the slices deployed as well as the granular resource configuration at each sub-net. For instance, it can modify PRB partitions, change the transport router priorities or change affinity rules in Kubernetes pods.

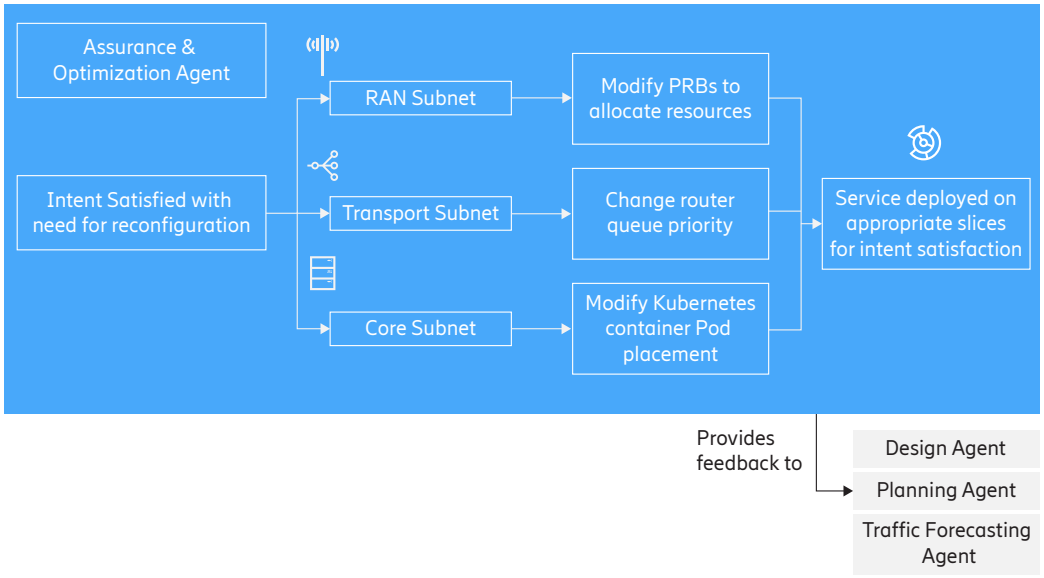

Figure 13. Assurance and optimization Agent

Note that multiple objectives such as prioritizing higher tiers of customers, performing actions with the lowest level of side effects (time/cost), and maximizing resource usage may be input into the system. A principal advantage of interfacing with the cognitive framework is that it can determine if an intent can be handled by the assurance/optimization agent (with configuration changes) or if it requires a coarser redesign or replanning.

Future evolution of the cognitive-based network LCM

While we have sub-divided the agents to target specific phases (as a first approximation as done today) to aid in knowledge management, agent life cycle handling and scale, moving forward to beyond 5G, all processing may be handled through a single end-to-end agent pipeline that receives information from the various phases. The best reuse of underlying

knowledge and agents will be when the phase-agent boundaries are broken, and all the resources are available to the cognitive framework. For example, the same planning or constraint solving agent can provide solution proposals for different tasks handled in different phases.

New challenges such as the scalability of the cognitive framework will eventually surface and these will have to be tackled by distribution. In addition, increased automation and virtualization capabilities could eliminate or effectively merge multiple phases in the life cycle.

Furthermore, the future evolution of RAN and core would have to be integrated within the cognitive life cycle management framework. One evolution of the RAN functionality would be the dynamic split into radio unit [RU], distributed unit [DU] and centralized unit [CU]. The dynamic functional split allows for real-time performance optimization, load balancing and differentiated QoS support (gaming, video). Cognitive framework would have agents for each of the splits with dedicated knowledge sources. Federation of knowledge and learning could be applied here for efficient LCM management.

Requirements for implementing the cognitive framework

There are a few implementation requirements to be considered in future evolution (5G and beyond) of the network LCM, which have been captured by Ericsson's product strategy:

- A common set of ontologies and knowledge graphs would be needed within the cognitive framework for the phases to interact.
- Multiple agent interfaces would be needed to facilitate registration, evaluation, and agent coordination.
- As there would be feedback from multiple subsequent stages (with varying actuation timelines), there could be conflicting actions generated. A system to prioritize and resolve conflicts across phases would be needed.
- The agents may themselves be implemented in independent cognitive-based architectures (as illustrated in Figure 5). Agents may identify issues and propose solutions to the end-to-end LCM without need for explicit loop definitions.

Related work

In [1] Improving Customer Experience and ROI with Mobile Planning, Design, and Optimization, Omdia and Ericsson highlight the need for streamlining network life cycle operations to reduce cost, improve reconfigurability and deploy newer 5G services. As suggested, service providers must improve synergies between various phases to reduce capital expenditure overheads.

In the Ericsson white paper [8] Driving 5G monetization through intent-based network operations, the importance of AI enabled 5G operations is highlighted to ensure superior customer experience, identify new revenue models and improve cost efficiency. Similarly, in the Ericsson white paper [7] Intent-driven Enterprise Service Orchestration, the pillars of intent-based management of enterprise customers is proposed. The result of this is that the ordered services can be delivered within minutes and is reduced to only minutes.

In the Ericsson Technology Review [10] End-to-end Network Slicing Orchestration, a transport aware network slicing abstraction is proposed that makes use of AI based traffic admission control and AI-based network parameter tuning to ensure efficient network slicing (demonstrated over an industrial use case). The life cycle management is also in line with Ericsson's view of Intelligent RAN Automation. This involves enhanced network planning and site selection, provisioning of complex networks, and zero touch optimization of networks to match customer intents and minimize operational costs.

The TM Forum Autonomous Networks Project [11] and ETSI Zero touch network and Service Management [ZSM] project [12] specify (intent-driven) closed loops vertically spanning the user (business), service, and resource layers. These loops can be seen as providing requirements to and getting feedback from the resource layer. Note that the deployment phase and the feedback loop that is relevant to this white paper is implemented within the resource layer.

Conclusion

The end-to-end life cycle in cellular network management currently consists of independent planning, design, tuning, fulfillment, assurance, and optimization phases. Going forward, 5G systems will be more complex with newer SLA services vying for resource usage against eMBB customers. Feedback loops from various phases of the network life cycle that enable rapid changes to deployments are required. Envisioned are multiple interphase loops that enable continuous feedback and reconfiguration between various phases, leading to solutions satisfying multi-objective criteria and varying traffic patterns. The intelligent AI agents coordinating actions in each of the phases are orchestrated through a cognitive framework to enable knowledge sharing. Technology evolutions such as automation, virtualization, and life cycle management of other layers (software, services, AI models) will have an impact on the phases and can be handled within the cognitive framework.

Glossary

3GPP	3rd Generation Partnership Project
AI	Artificial Intelligence
API	Application Programming Interface
BSS	Business Support Systems
CSPs	Communication Service Providers
CU	Centralized unit
DU	Distributed unit
eMBB	enhanced Mobile Broadband
KPIs	Key Performance Indicators
LCM	Life Cycle Management
ML	Machine Learning
OSS	Operations Support Systems
PRB	Physical Resource Block
QoS	Quality of Service
RAN	Radio Access Network
RU	Radio unit
SLA	Service Level Agreement
UE	User Equipment
URLLC	Ultra-Reliable Low-Latency Communication
ZSM	Zero touch network and Service Management

References

1. Omdia and Ericsson, Improving Customer Experience and ROI with Mobile Planning, Design, and Optimization, 2021.
2. 5G PPP Technology Board, "AI and ML – Enablers for Beyond 5G Networks", 2021.
3. G. Adachi, A. De Domenico, D. T. Hoang and D. Niyato, "An Artificial Intelligence Framework for Slice Deployment and Orchestration in 5G Networks," in IEEE Transactions on Cognitive Communications and Networking, vol. 6, no. 2, pp. 858-871, June 2020.
4. Ghallab, Malik; Nau, Dana S.; Traverso, Paolo (2004), Automated Planning: Theory and Practice, Morgan Kaufmann.
5. Y. Li, E. Pateromichelakis, N. Vucic, J. Luo, W. Xu and G. Caire, "Radio Resource Management Considerations for 5G Millimeter Wave Backhaul and Access Networks," in IEEE Communications Magazine, vol. 55, no. 6, pp. 86-92, June 2017.
6. Ericsson White paper, Cognitive processes for adaptive intent-based networking - Ericsson, 2020.
7. Ericsson White paper, Intent-driven Enterprise Service Orchestration, 2021.
8. Ericsson White paper, Driving 5G monetization through intent-based network operations, 2021.
9. Capgemini Engineering White paper, 5G Radio Access Network Planning and Optimization, 2020.
10. Ericsson Technology Review, End-to-end Network Slicing Orchestration, 2022.
11. Autonomous Networks: Empowering digital transformation for smart societies and industries, TM Forum white paper, October 2020.
12. Network Transformation; (Orchestration, Network and Service Management Framework), ETSI White Paper No. #32, October 2019.

Other relevant studies

- Huawei 5G Wireless Network Planning Solution White Paper
- Amdocs Network Rollout Solution
- Capgemini 5G radio access Network planning and optimization
- Wipro End-to-End Network Lifecycle Automation in Telecom Networks
- The SOLIDS 6G Mobile Network Architecture: Driving Forces, Features, and Functional Topology

Authors

Ajay Kattepur is a Senior Researcher in the Autonomous Intelligent Systems group at Ericsson Research, with a focus on applying automated planning, reinforcement learning, and verification techniques for 5G networking and robotics applications. Prior to joining Ericsson Research, he was with the Tata Consultancy Services (TCS) Research & Innovation Labs in India. Ajay received his Ph.D. in Computer science from French National Institute for Computer Science and Control (Inria), Rennes, France, and the M.Eng. and B.Eng. degrees in Electrical & Electronic Engineering from Nanyang Technological University (NTU) Singapore.

Sushanth David was formerly a Director of Portfolio Management (Automation & AI) of the Ericsson Managed Services Unit. Sushanth's focus is on the design and construction of reusable frameworks, packages, and components with additional emphasis on applying AI based theories and techniques to the fulfillment and assurance across all domains of the 5G network. His expertise is in SDN/NFVi & O-RAN, time sensitive communications, and high performance distributed compute. Sushanth has served on the Linux Foundation networking board and contributed to ONOS and CORD initiatives. He holds an MS in Computer Science from PennState, USA, and B. Eng in Electronics & Communication from VT University, India.

Swarup Kumar Mohalik is a Principal Researcher at Ericsson Research. His expertise is in the areas of AI and formal methods, and his work primarily focuses on applying them to telecommunications, service automatization, and the Internet of Things (IoT). He has research experience in formal specification and verification of real-time embedded software and AI planning techniques. Swarup holds a Ph.D. in computer science from the Institute of Mathematical Sciences, Chennai, India, and a postdoctoral fellowship at LaBRI, University of Bordeaux, France.

Stephen Terrill is a senior expert in automation and management, with more than 20 years of experience working with telecommunications architecture, implementation and industry engagement. His work has included both architecture definition and posts within standardization organizations such as ETSI, 3GPP, ITU-T (ITU Telecommunication Standardization Sector), and IETF (Internet Engineering Task Force). In recent years, his work has focused on the automation and evolution of OSS, and he has been engaged in open source on ONAP's Technical Steering Committee and as ONAP architecture chair. Terrill holds an M.Sc., a B.E. (Hons.), and a B.Sc. from the University of Melbourne, Australia.