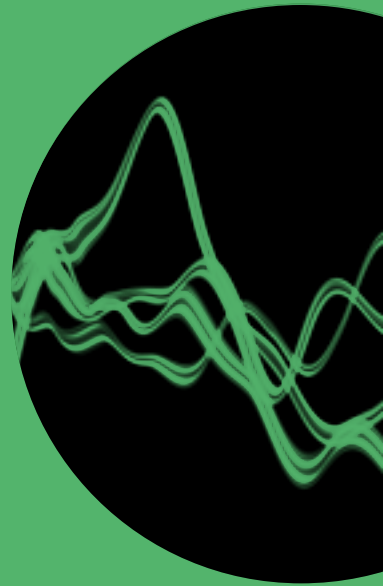


Review

ERICSSON
TECHNOLOGY



MULTI-LAYER,
INTENT-BASED
**AUTONOMOUS
NETWORKS**



Autonomous networks with multi-layer, intent-based operation

A truly autonomous network has the ability to change requirements dynamically without human involvement. Creating a network that can do this requires the use of intents between multiple operational domains of the network architecture, including the business, service and resource layers.

.....
JÖRG NIEMÖLLER,
JOHAN SILVANDER,
PAUL STJERNHOLM,
LARS ANGELIN,
ULF ERIKSSON
.....

As consumer expectations continue to rise and networks grow ever more complex, communication service providers (CSPs) around the world are becoming increasingly aware of the need for intent-based autonomous networks that significantly reduce the need for human intervention.

■ Intent is the expression of declarative requirements that stem from the specific business needs of a CSP and its customers. When operations are based on intents as opposed to tasks, the

●● THE RECEIVING SYSTEM HAS A MUCH HIGHER DEGREE OF FREEDOM IN CHOOSING WHICH SOLUTION STRATEGIES AND ACTIONS TO APPLY ●●

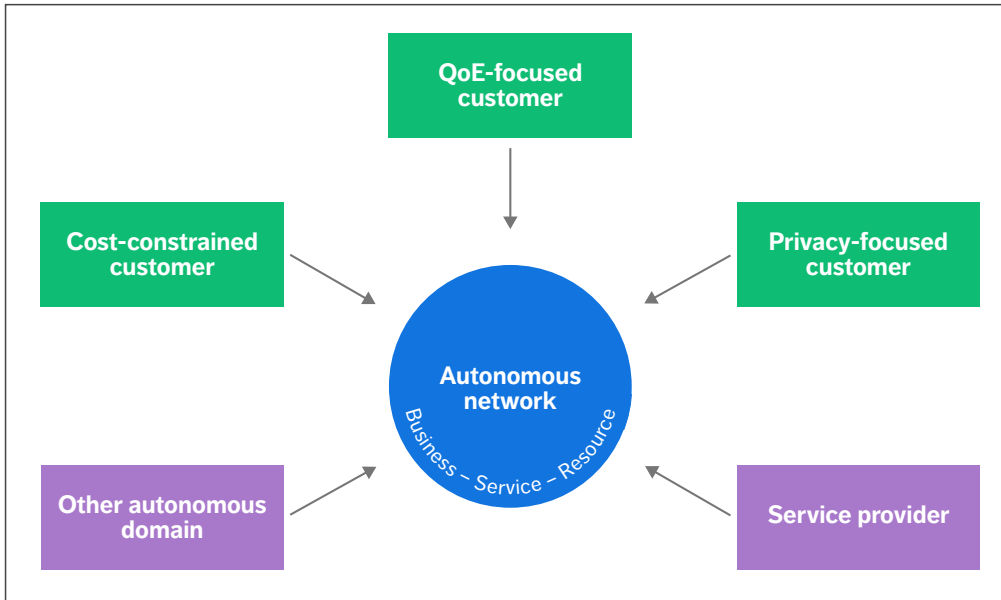


Figure 1 Intents are derived from the business needs of multiple parties within an autonomous network

receiving system has a much higher degree of freedom in choosing which solution strategies and actions to apply. Requirement-centric interaction is essential to achieve a clear separation of concerns, which ultimately enables higher levels of autonomy.

Managing requirements with intents

Most CSPs today have a broad customer base comprised of organizations whose priorities, expectations and network requirements may vary widely. Figure 1 illustrates that one customer may be particularly constrained by cost, for example, while another prioritizes quality of experience (QoE)

above all else. For yet another customer, privacy may be of the utmost importance. The CSP serving these customers will also have its own distinct expectations. As the owner of the autonomous network, the CSP expects it to generate revenue while also being secure, energy efficient and compliant to regulation. The purpose of intents in this scenario is to serve as information objects within the operational processes that carry knowledge about the individual requirements of all parties.

The intent receiver is an automated software system. An autonomous network consists of multiple automated systems arranged in an architecture with

Terms and abbreviations

API – Application Programming Interface | **BSS** – Business Support Systems | **CSP** – Communication Service Provider | **DT** – Digital Twin | **E2E** – End-to-End | **IMF** – Intent Management Function | **KPI** – Key Performance Indicator | **ML** – Machine Learning | **NF** – Network Function | **OSS** – Operations Support Systems | **QoE** – Quality of Experience | **RAN** – Radio-Access Network | **SKPI** – Service KPI | **SLA** – Service Level Agreement | **SMO** – Service Management and Orchestration

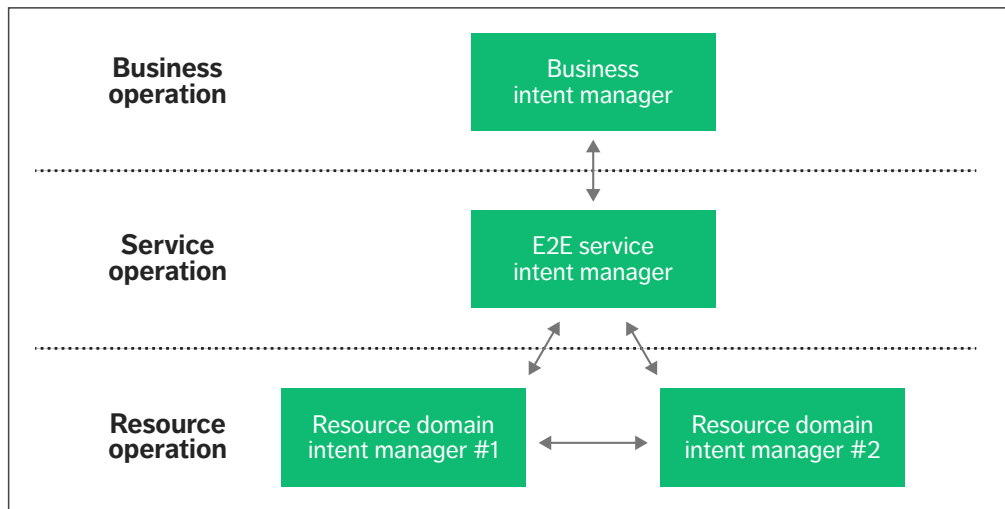


Figure 2 Ericsson's multi-layer, intent-based operation concept

business, service and resource layers, as shown in [Figure 2](#). Each layer contains various autonomous domains [1]. For example, in the resource layer, a radio-access network (RAN) management domain controls RAN resources and services next to other domains, such as transport and cloud management. In this architectural structure, an intent can be used between any operational layers and individual domains. Intent is therefore not only used for expressing the requirements of users and businesses directly, but also for distributing derived individual requirements between operational domains.

A system domain that is able to assure and optimize its operation autonomously within its defined scope of responsibility is referred to as an “autonomous domain.” Within an autonomous domain, an intent manager – also known as an Intent Management Function (IMF) [2] – is the entity that receives and sends intent. It controls the assurance loop for domain-specific requirements within the domain and coordinates with other domain-specific functions regarding which actions to take for provisioning, optimizing and healing. One example of a possible action that an autonomous domain can take in an intent-driven network is sending further intents to other autonomous domains.

The originator of the intent typically has its own requirements to fulfill. To be able to do this successfully, it defines instrumental requirements to which it needs other domains to comply. It uses intent to express these requirements and an intent application programming interface (API) [3] to communicate them and manage their life cycle. The intent receiver needs to assure that its domain complies with the requirements and reports its success back to the originator.

A key characteristic of intent-driven autonomous networks is therefore a hierarchy of requirements expressed through intent and handled through a hierarchy of intent managers and their autonomous domains. Broader requirements with network-wide, end-to-end (E2E) scope are handled in higher layers. They are typically translated into requirements that are specific to autonomous domains. An autonomous domain is responsible for subsets of resources, considering geographical distinction or branches in the network topology, for example. The central task of each individual domain is to ensure it meets its specific intent requirements and thus contributes to the success of the entire autonomous network in meeting CSP and customer expectations.

Achieving a high degree of autonomy

Three factors are essential to reaching a high degree of system autonomy in intent-based operations: (1) assurance, (2) utility-based decisions and (3) control of the effect and risk of actions.

Assurance

Intent managers assure intent-specified requirements by continuously monitoring the system state and ensuring that all requirements are met, even in cases where there are multiple, potentially conflicting, intents. If an intent manager discovers that a requirement is not being met, it will autonomously take countermeasures and inform the intent owner if it cannot find a solution. If a new intent adds requirements, the intent assurance loop will immediately consider them. In this respect, starting a provisioning process is a countermeasure taken by the intent manager's assurance loop, if it determines that insufficient resources make it impossible to meet intent requirements. If an intent is removed, the intent manager would detect a use of resources that is not justified by requirements. In this case, releasing resources would be a sensible optimization action.

Utility-based decisions

Utility is information about the relative value of an observed state or reached key performance indicator (KPI) value. For example, a typical intent requirement in the service and resource layer may state that a service latency experienced by a user must always be below 100ms. This is a binary requirement that is either met by the system or not. Utility is a gradual expression of the value and importance of any potentially reached latency. If the observed latency is 110ms, the requirement as such is not met, but utility information would quantify if it constituted a slight inconvenience represented by a slightly lower utility value than reaching the 100ms. On the other hand, if the observed latency value represents an important service disruption, the associated utility value would be significantly lower.

Utility is expressed by utility functions and included in the intent expression, which enables the

Key concepts

Intent introduces a structure by which requirements are expressed and managed in a standardized and unified fashion across multiple systems. The requirement-centric approach constitutes a clear separation of concerns between various system layers and domains.

Utility is a measure of business value and relevance of reaching a certain compliance level to a requirement. It will become a key concept for self-adaptive autonomy. The automated system can use utility to make decisions based on a general understanding of value. It has the potential to eliminate the need for use-case-specific logic or human assistance through updating policies. Decisions made through utility maximization ensure that whatever an autonomous domain is doing, it will always be optimized for business value regardless of any new situations or conflicts the intent-based system is facing.

Digital twins are software components that behave like the resources they represent. In the context of intent management, they allow the safe exploration of alternative solutions before they are applied. The combination of exploration with DTs and utility-based evaluation will be a key enabler of autonomy.

originator of intent to not only state requirements, but also express the importance of and preference for an outcome.

The concept can be applied to any requirement on any layer. A utility function for revenue requirements has a direct relationship to business value. A utility function for latency in the resource layer would be shaped with respect to the latency's indirect contribution to revenue. It can furthermore consider customer value and priority through elevated utility for some requirements compared with others.

This makes utility information the base for optimization and prioritization decisions while facing conflicts. For example, optimizing the system for requirements related to energy saving and user experience often requires opposing actions. Utility

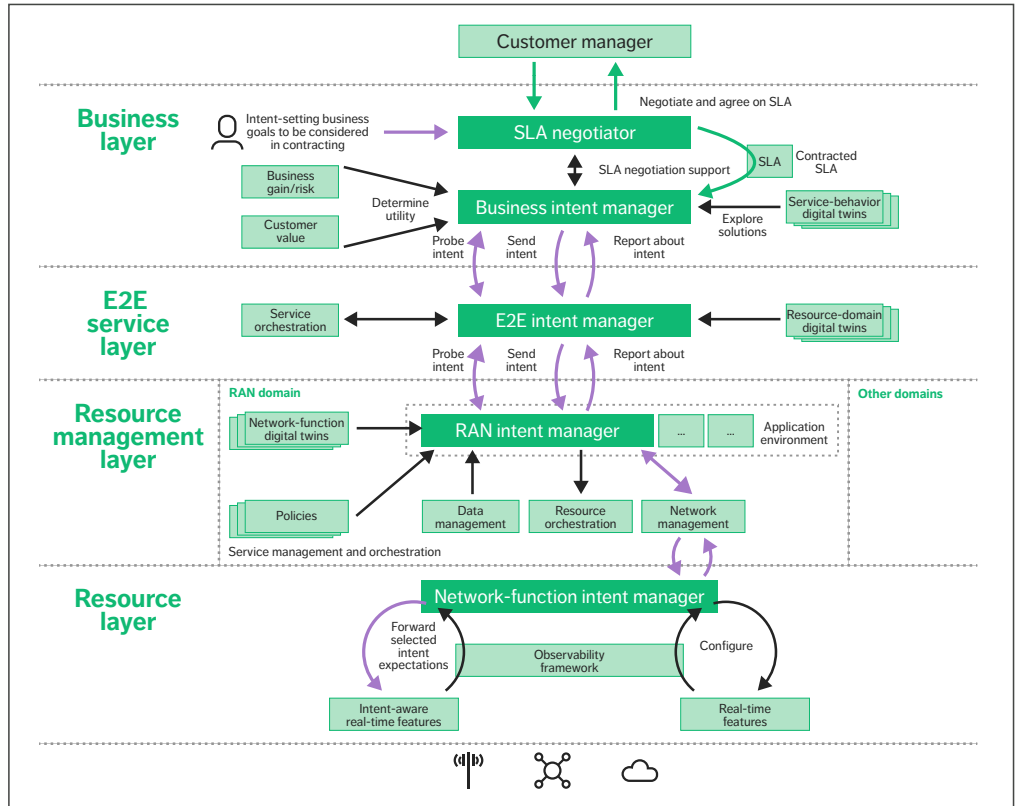


Figure 3 Multi-layer, intent-based operation – detailed view

functions allow for finding the optimal compromise by going for system configurations that maximize accumulated utility across both requirements. The autonomous network therefore reaches the most valuable combined user experience and energy-saving levels possible.

Control of the effect and risk of actions

An intent-based system needs to ensure that its actions not only have the desired outcome, but also have low risk for negative or collateral effects. This can be controlled by employing models that predict possible outcomes and their likelihood. An essential tool would be a digital twin (DT) allowing the

provisional execution of actions without impacting the physical and operational resources. Utility scoring applied to the current system state and possible outcomes of investigated actions would identify and eliminate actions that do not improve the state or bear an unreasonable risk of degradation.

If an intent manager implements intent assurance loops with these capabilities, the result is a highly capable autonomous domain. It would have tight and fully automated control over the domain's compliance to intent requirements, including automated adaptation of intent changes. Through utility-based decisions in combination with

prediction of action effects and risks, it can self-adapt to new situations and always find sensible solutions even for use cases and states that were not explicitly considered at design time. This therefore further reduces the need for human intervention and escalations.

Ericsson's comprehensive, multi-layer approach

The standardization of intent for various autonomous network domains is currently underway in several forums including TM Forum, the 3GPP (3rd Generation Partnership Project) and the O-RAN Alliance. While standards are maturing, Ericsson is exploring the use of intent across our portfolio. **Figure 3** illustrates our approach to intent-based operation use cases in business, service and resource domains, which is based on the cognitive loop presented in a recent Ericsson Technology Review article [4].

Intent in the business layer

The central concern of a business support system (BSS) is to help the CSP capitalize on the investments it makes in its network infrastructure, partner ecosystem, business models and multichannel engagements, while simultaneously maximizing its profitability and delivering a good experience for consumers and enterprises. The decisions taken to realize the desired business outcomes can be seen as reasoning under uncertainty. Intent-based management is well suited to operate in this environment.

In the following examples, we explicitly state the use of the intent-based interfaces [3] between the BSS and the operations support system (OSS). It should be noted, however, that the intent interfaces can also be used between the BSS components.

Part 1: Contract negotiation, operational readiness, and support

In a B2B scenario, every contract between the CSP and a specific customer is a unique instance. Each customer places its requirements on the equipment to obtain a desired business outcome. The customer

WHILE STANDARDS ARE MATURING, ERICSSON IS EXPLORING THE USE OF INTENT ACROSS OUR PORTFOLIO

also has a solid business understanding of how to prioritize the different parts of its business if a resource shortage appears. The choice can be between different equipment using the same service or between different services and the equipment they use. This type of business is driven by Service Level Agreements (SLAs). The SLAs and their related KPIs are business requirements that must be fulfilled with the help of resources in the network.

During the contract negotiation, the customer defines its requirements for the service in the form of service KPIs (SKPIs). With the help of the BSS knowledge and network information, the effects of these SKPIs can be estimated through simulations. Examples of the effects are:

- » How different SKPIs affect each other
- » How SKPIs affect resources
- » How constraints on resources affect the SKPIs
- » How states affect SKPIs.

This additional information can be used to estimate the costs of the contract, the contract implications, and the probability of an impact on the existing resources. The estimates may lead to a variety of possible outcomes including a price modification of the service, a decrease in the amount of user equipment, network buildout or constraints on the stated SLA.

Business metrics are used to calculate the utility of the service(s) covered by the contract, which is defined in terms of the business importance of the service. Examples of metrics include the cost of SLA violations, economical margins, the business state of the customer and the importance of a long-term relationship. The business importance of the service

TOGETHER WITH A SERVICE ORCHESTRATOR, AN E2E INTENT MANAGER IS ONE OF THE KEY COMPONENTS OF AN INTENT-AWARE SERVICE LAYER

can be updated at any time, which triggers a reevaluation of the network capacity and business risks.

The information in the contract proposal, together with the calculated business importance, is used to estimate and simulate the effects of the contract on other customers' services during normal load and during resource shortage. The estimated effects and the probability of their occurrence are used to analyze the business risks of accepting the contract. The estimations and simulations are supported by the BSS knowledge.

After the simulations, requests are sent to the OSS and the SLA monitoring function to get a more accurate estimate of the abilities to honor the SLAs and SKPIs. When detected, issues in the contract are corrected and agreed upon, and requests are sent to the OSS and the SLA monitoring function. Both requests are sent through the IMF interface as intents.

Part 2: Contracts in operation

During the life cycle of an agreed-upon contract, its SLAs must be monitored, and potential SLA violations detected and mitigated. An SLA-monitoring function within the intent manager's control loop predicts and detects violations. Further functions in the loop propose mitigation strategies and the probability of their success, which are analyzed from a business-risk perspective before deciding on actions.

While it is not possible to alter an SLA without engaging in contract negotiations with the affected customer, the intent manager can resolve performance issues without the need for contract negotiations by proposing solutions that are adjusted

for business importance. These adjustments are based on the result of estimations and simulations supported by the BSS knowledge. Adjustments to business importance are expressed through utility information within intent and sent to the service layer through the intent interface.

Intent in the end-to-end service layer

The primary purpose of the service-layer intent manager is the realization and assurance of E2E service in accordance with service requirements received from the business layer. Compliance to intent received from the business layer is the terminal goal of the service layer. Central to achieving this is the translation of service intent into instrumental goals constituting resource-layer requirements for individual resource domains. The idea is that the business intent is met as long as each individual resource domain remains compliant to its assigned intent.

Together with a service orchestrator, an E2E intent manager is one of the key components of an intent-aware service layer. This intent manager analyzes the requirements regarding services to be delivered to customers and decides which services exposed by the network would be suitable. Digital twins of resource domains can help in the evaluation of potential solutions [6]. The intent manager orders the selected services from the orchestrator resulting in various actions toward resource domains. The E2E intent manager also monitors the continued fulfillment of the customer requirements.

With intent-aware resource-layer domains, the action that the service layer takes can be the sending of one or several coordinated intents. This can be in addition to other actions using non-intent-based interfaces for configuration, provisioning and management. While this article focuses on the RAN domain, coordinated actions would also be taken in other domains such as core network or transport. If the respective resource domain is intent aware, such action can involve the sending of resource-specific requirements through intent.

Intent sent into the RAN domain typically involves quality-of-service-related requirements

such as experienced service latency and available throughput. For example, the latency requirements sent to the RAN domain constitute the latency budget that the RAN has to comply with as part of the overall E2E service latency. This breakdown from E2E requirements into partial contributions is a central task of service-layer intent management. The intent can be further contextualized, for example, regarding timeframe of validity, geographical region or user groups.

Intent in the RAN domain: RAN management

The central task of RAN management is to assure that the performance of the RAN infrastructure meets the requirements that the E2E service layer has allocated to it. This article considers a RAN resource domain with two layers of intent. The intent coming from the E2E service layer is handled within the RAN management layer. We have chosen to define the RAN resource management layer according to the system architecture of Service Management and Orchestration (SMO) as defined by the O-RAN Alliance [5]. This architecture is characterized by a set of common platform functions including data management, analytics, orchestration and network management. Furthermore, it provides execution and management environments for applications, policies and machine learning (ML) models.

The applications in the automation platform are software components that implement use-case-specific logic and network control, interacting with common platform functions through standardized interfaces. Based on these principles, we have created a RAN intent manager as an application within the automation platform.

As shown in Figure 3, the RAN intent manager receives intent from the E2E service layer. The intent manager verifies compliance based on observations acquired through data management and analytics within the framework of the automation platform. If it identifies a need to change resource provisioning or configuration, for example, the intent manager can act by utilizing orchestration and network management.

Our automation platform utilizes various distinct software components in the assurance loop of the intent-manager application. The components used to do so are referred to as agents. Agents implement various concerns of the assurance loop including:

- » Collecting measurements
- » Identifying issues with respect to intent compliance
- » Proposing healing actions and predicting their outcomes
- » Considering utility and risk in decisions
- » Executing the agreed solutions.

The agents used in our automation platform can be machine-learned models, policies or applications based on other technologies. For example, manually-designed or machine-learned policies can generate solution proposals. Digital twins, which typically include machine-learned models, can be used to safely evaluate the solution proposals and estimate their effects on network behavior. Based on this estimate, an evaluation agent applies utility functions to the current state and the predicted states. A final decision is made by maximizing utility across the available solution alternatives. In this way, the autonomous domain implements a technical solution that is optimized according to customer preference, CSP strategy and overall expected business value. The automation framework would also provide conflict resolution between intent-manager applications and non-intent-aware applications.

Intent in the RAN domain: RAN network functions

RAN network functions (NFs) are characterized by multiple control loops with real-time characteristics. A key challenge of intent-based RAN NFs is therefore calculating how to address real-time characteristics, while facing dynamically changing and potentially conflicting requirements from multiple sources.

To overcome this challenge, Ericsson proposes the introduction of intent-aware RAN, which is best

achieved by allocating intent managers in RAN NFs. This constitutes the second layer of intent within the RAN domain. These intent managers implement intent assurance loops. As such, they monitor RAN performance through an observability framework and identify discrepancies with respect to the intents.

Similar to intent managers in other domains, a RAN NF intent manager collects possible solution strategies from proposing agents, explores their effect using prediction models and makes decisions using utility information. A broad range of implementations is possible for proposing agents and prediction agents, ranging from the reuse of event-condition-action policies to machine-learned models.

If, for example, the resource-controlling features are not intent-aware and the RAN intent manager is responsible for adapting them to the current requirements, it can act by reconfiguring real-time features such as radio resource partitioning or service-adaptive link adaptation. In this way, RAN management can meet real-time requirements by employing real-time optimized features, while a potentially slower intent control loop observes if the real-time features meet intent. If not, configuration changes can be applied to the real-time features.

Alternatively, some RAN features can become intent-aware. For example, an intent-aware scheduler or intent-aware link adaptation can implement self-optimization logic that uses requirements originating from intents directly. In this scenario, the RAN NF intent manager would relay certain requirements directly to the intent-aware feature. The intent manager would ensure that the configuration of intent-aware and non-intent-aware RAN features is consistent and represent an optimal overall operation strategy.

In short, the role of a RAN NF intent manager is predominantly oversight across many real-time control features. The intent manager does not interfere directly with the operation logic, but instead verifies that the RAN features are performing according to the intent requirements. If needed, it takes action through configuration.

Conclusion

Ericsson's intent management solutions enable a high level of autonomy, combining utility evaluation with digital twins and machine-learning-based automated exploration of system behavior. By applying these concepts to multiple architectural layers and system domains with intent as the central mechanism for requirement expression and distribution, we have designed an autonomous network that is highly self-adaptive and reliable. Our concept for multi-layer, intent-based operation enables the network to reliably detect and mitigate conflicts and keep itself optimized, while reducing the need for human assistance, including manual updates of policies and use-case-specific logic.

In each of the layers of network operation, intent constitutes a control loop between the source of intent and the system that needs to comply with the intent. Declarative requirements are sent southbound, while intent reports ensure that the originator of the requirements remains informed. The models for expressing intent as well as the application programming interfaces for communicating it are very similar across the system layers. The intent architecture is the same, while different domains mainly constitute different contexts. Within each autonomous domain, an intent manager coordinates the actions needed for complying to the intent, by maximizing the overall utility.

Further reading

- » **Ericsson, Autonomous networks, available at:**
<https://www.ericsson.com/en/future-technologies/autonomous-networks>

References

1. **TM Forum, IG1251 Autonomous Networks – Reference Architecture v1.0.1, 2022, available at:**
<https://www.tmforum.org/resources/how-to-guide/ig1251-autonomous-networks-reference-architecture-v1-0-1/>
2. **TM Forum, IG1253 Intent in Autonomous Networks – v1.3.0, 2022, available at:**
<https://www.tmforum.org/resources/how-to-guide/ig1253-intent-in-autonomous-networks-v1-3-0/>
3. **TM Forum, TMF921 Intent API – v1.1.0, 2022, available at:**
<https://www.tmforum.org/resources/standard/tmf921a-intent-management-api-profile-v1-1-0/>
4. **Ericsson Technology Review, Creating autonomous networks with intent-based closed loops, April 19, 2022, Niemöller, J et al, available at:** <https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/creating-autonomous-networks-with-intent-based-closed-loops>
5. **Ericsson white paper, An intelligent platform: The use of O-RAN's SMO as the enabler for openness and innovation in the RAN domain, available at:** <https://www.ericsson.com/en/reports-and-papers/white-papers/smo-enabling-intelligent-ran-operations>

THE AUTHORS



Jörg Niemöller

◆ is an analytics and customer experience expert in Solution Area Cognitive Network Solutions. He joined Ericsson in 1998. He is currently leading the introduction of machine-reasoning technologies into Ericsson's portfolio to enable solutions for autonomous networks. Niemöller holds a Ph.D. in computer science from Tilburg University, the Netherlands, and a Dipl.-Ing. in electrical engineering from the TU Dortmund University, Germany.



Johan Silvander

◆ is a senior specialist in information management who has worked at Ericsson

for more than 20 years, focusing on OSS and BSS in a variety of roles including serving as a member of core architecture teams, working as a designer, taking technical responsibility for integration and installation projects, and being a test leader. He holds a Ph.D. in software engineering from Blekinge Institute of Technology, Sweden.



Paul Stjernholm

◆ joined Ericsson in 1995 and has been dedicated to the mobile telecom industry since 1G. In his current role as concepts researcher, his work focuses on RAN management strategies, automation and standardization. Stjernholm holds a M.Sc. in applied physics and electrical engineering from Linköping University, Sweden.

Lars Angelin

◆ is an expert in BSS within Business Area Cloud Software & Services

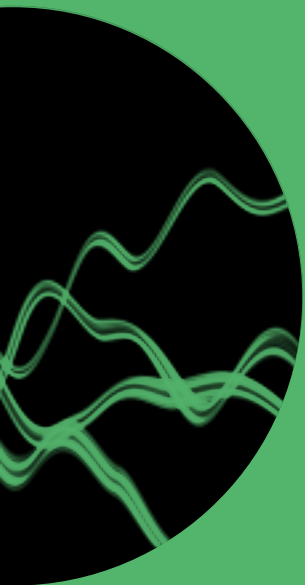


Business Area Digital Services. He joined Ericsson in 1996 as a research engineer. Since 2006 he has focused on BSS – specifically business support, enterprise architectures and the software architectures to implement BSS systems. He holds an M.Sc. in engineering physics and an honorary Ph.D. from Blekinge Institute of Technology.



Ulf Eriksson

◆ joined Ericsson in 1994 and currently serves as a system manager. He holds an M.Sc. in computer science from Linköping University.



ISSN 0014-0171
284 23-3395 | Uen

© Ericsson AB 2023
Ericsson
SE-164 83 Stockholm, Sweden
Phone: +46 10 719 0000