Ericsson White Paper
BCSS-26:032863 Uen
November 2026

ERICSSON

Serviceability

Content

What serviceability is and why it is important **3**

Solution: What good serviceability is and how to achieve it **4**

Serviceability and automation 4

Serviceability and observability 5

Serviceability and feedback loops 6

Serviceability and artificial intelligence 8

Conclusion **11**

References and further reading **12**

Authors 13

What serviceability is and why it is important

Serviceability refers to the ability of products and solutions to support efficient deployment, operation, and maintenance. However, maintaining high serviceability in complex telecommunications systems is becoming increasingly challenging. As networks become more software-driven and automated, operators and vendors face pressure to keep systems on the latest software releases that deliver improved quality, security, and features without causing service disruptions. Communication service providers (CSPs) and infrastructure suppliers often struggle to ensure smooth, risk-free upgrades and deployments, manage configuration complexity, and gain visibility into system health while minimizing manual effort.

The most common challenges are coordinating multiple interdependent network functions and tools, extended outages or degraded performance during updates, and obtaining reliable, actionable feedback from both machine-generated metrics and human insights. For example, automation tools may malfunction or lack the stability needed for remediation. CSPs and suppliers also lack standardized benchmarks to objectively compare serviceability performance across vendors, networks, and releases, hindering continuous improvement and trust-building.

In telecom networks, where agility and reliability are crucial, these issues create a need for techniques to anticipate issues and measure serviceability early and continuously, reducing costly post-release fixes. Without addressing these concerns, CSPs risk inefficiencies, increased downtime, and degraded customer experience in a highly competitive environment.

Solution: What good serviceability is and how to achieve it

Serviceability and automation

A system with good serviceability stays on the latest software release, providing the best quality, security, and features. It can be upgraded at any time, without offloading traffic and with no or limited human involvement. Deployments and upgrades follow a digital, automated procedure that requires minimal time and effort for planning, preparation, and execution, with a success rate close to 100 percent.

Software and configuration state are version-controlled, while the system state can be easily updated or rolled back to any previous state. Configuration is designed to be as simple as possible for the user, without sacrificing flexibility. Configuration changes are automatically uplifted as needed, such as during an upgrade. Different deployment options are flexibly available, without sacrificing simplicity. Dependencies between network functions, their modular blocks, and tools are tracked to ensure compatibility at any time, guiding the process of upgrading complex networks while retaining compatibility.

Lifecycle management (LCM) and configuration work seamlessly at any scale, from large sites to many smaller sites. All infrastructure requirements and prerequisites are managed in a machine-readable format and automatically validated or realized. All LCM capabilities are available on demand for use by higher-order orchestration and closed-loop automation. In case of automation failures, good troubleshooting tools are available to enable quick root-cause analysis and issue resolution. During LCM, the system automatically checks network health and functionality and takes preventive action when needed.

To achieve this, several coordinated capabilities are needed. Procedures and system requirements must be fully digitized as machine-readable, customizable assets, removing the need for document preparation and review. LCM procedures take the form of orchestration pipelines that handle all required steps. Configuration is also governed by machine-readable assets, facilitating automated validation, uplifts, and processing. Tools are available in every step of the flow to prepare, validate, or transform the data used in the procedure.

All data is validated before execution against schemas, rules, and dry-run checks to reduce risk and support the first-time-right principle, which is key to good serviceability. The deployment and upgrade procedure is designed to be fully autonomous and executable at any time, with no impact on traffic. All system states are maintained in a version-controlled repository, with the network automatically reconciling against that as the source of truth, such as in a GitOps paradigm.

The software, automation, and configuration follow a modular approach, enabling flexible deployments. Automation tools guide and support in keeping systems in a compatible state throughout the upgrade process. To support distributed installations, inventoried data is used alongside highly automated, pre-verified procedures.

All LCM and configuration capabilities are exposed through application programming interfaces (APIs), such as GitAPIs and automation/orchestration APIs, to facilitate integration with northbound systems. LCM tools integrate with observability to facilitate quick root cause analysis and issue fixing.

Serviceability and observability

To improve serviceability, it is important to obtain near real-time data that reflects the system's current state. Observability of system applications, as well as their underlying platforms and infrastructure, enables insights for monitoring and analysis. Telemetry data is then scraped from sources, collected, and exposed for external persistence and consumption [2].

To trigger decisions and actions in automated control loops, such telemetry data is exposed via streaming mechanisms to ensure timely processing and actuation on the current state. The first stage of this processing aggregates and correlates telemetry data from different layers and sources. The resulting views are stored in a data store to support multiple consumption use cases.

Externalized data processing has certain requirements for format and content. Telemetry data must be in a structured format with an available schema and enriched with source metadata using a semantic convention. If the system comprises components from multiple vendors, it is even more beneficial to agree on or standardize the interfaces and formats for data exposure.

One important use case for telemetry data is preventive maintenance. Having the system's current state available in near real time and correlated across different sources enables analytical processes to predict the system's expected future state. If the predicted state indicates, for example, service degradation or resource shortages, corresponding countermeasures could be initiated to prevent those unwanted situations. Other algorithms could predict potential outages and failures of the whole system or its components before they occur, allowing mitigation before they materialize.

Finally, comprehensive, correlated telemetry data also enables easier, more consistent root-cause analysis of observed events and fault situations. Because aligned metadata is attached to data objects, teams can identify relationships between multiple events and cause chains faster and more easily.

Serviceability and feedback loops

Serviceability is not a static attribute but an evolving capability that develops iteratively and improves over time. This iterative nature requires structured feedback loops supported by solid, reliable serviceability metrics to guide decisions and evaluate change.

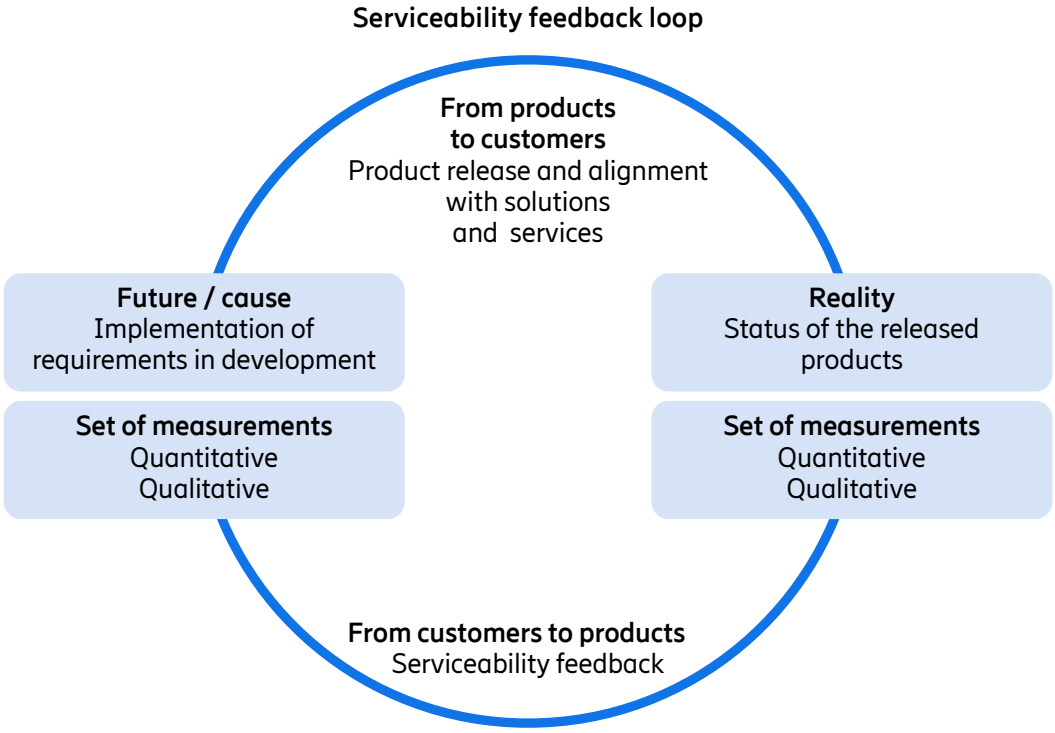
Basically, the feedback loop is a mechanism that feeds operational insights back into the development and support life cycle. Beyond enabling continuous learning and improvement, a well-designed feedback loop can connect product behavior with operational insight, creating a closed cycle from users, including CSPs and support/operational teams, to development, such as software vendors. This way, the effects of system changes, from software and configuration to workflows such as ways of working or processes, can be measured, understood, and acted upon.

In high-scale telco-grade environments, manual processes and human-entered data are insufficient. To be effective, a feedback loop must be continuous, automated, data-driven, and integrated into workflows.

At the heart of a strong feedback loop are quantitative and qualitative metrics. Quantitative metrics are objective, numeric indicators generated by systems such as observability platforms, log analytics, continuous integration and continuous delivery (CI/CD) telemetry, and support case management tools. Metrics in this category include mean time to detect (MTTD), mean time to resolve (MTTR), auto-resolution rates, and incident rates after software releases. These quantitative metrics provide consistent measurements and enable benchmarking over time.

Complementing these are qualitative metrics, which are subjective insights derived from human input. These may come from post-incident reviews, usability testing, customer feedback, or support engineer observations. Qualitative data helps explain why certain outcomes occur, offering vital context machines cannot always capture.

Together, these two types of metrics provide a complete view of serviceability performance, effectively explaining what is happening and why it matters.



Different types of feedback loops are necessary to address the full spectrum of serviceability challenges:

- Quality and performance: This can be measured by evaluating how the product changes, such as new features, patches, or architectural updates, and the impact of these changes on overall serviceability.
- Stability and reliability of automation machinery: This can be observed by focusing on the behavior of automation systems, such as auto-remediation scripts, diagnostic tools, and CI/CD pipelines.
- Efficiency of processes and automation: The best way to capture this is by learning how users interact with tools and processes.

Solid serviceability metrics are more meaningful when supported by benchmarking across vendors and CSPs. Understanding what is good, what is average, and what needs immediate attention can create a shared standard for diagnostics, objective comparison, and continuous improvement. Through metrics, vendors and CSPs can continue building and reinforcing trust and transparency.

A serviceability maturity model and framework can further guide development. Such a model can outline a progression from reactive, manual operations to predictive, self-optimizing environments. This method provides a structured framework for assessing current capabilities, prioritizing serviceability initiatives, and aligning technological evolution with strategic goals.

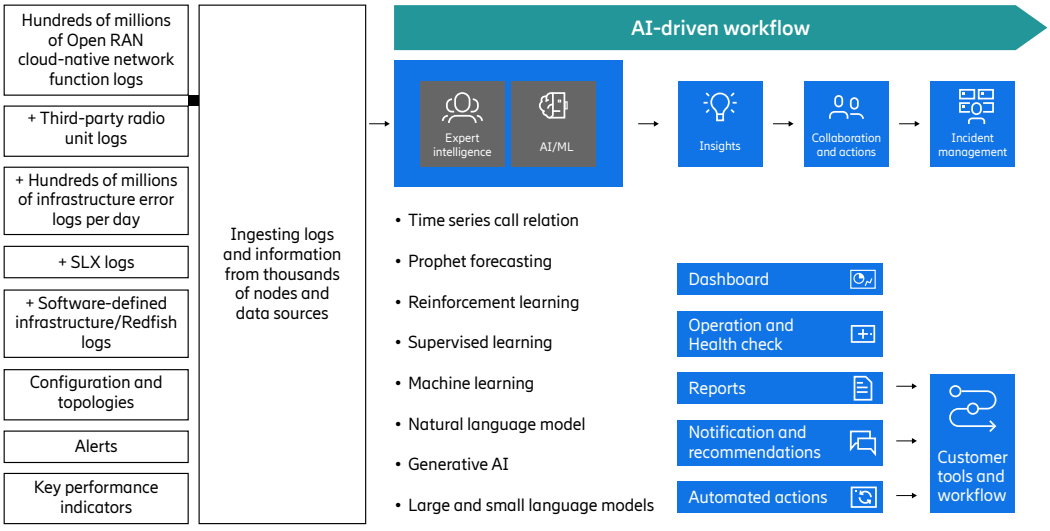
A key element of this framework is the shift-left approach, which assesses serviceability early, usually during the study or opportunity analysis phase, before development begins. By analyzing potential serviceability risks and requirements during design, organizations can ensure that new features include needed LCM capability, observability, diagnostic readiness, and operational safeguards from the start. This reduces the need for costly post-release fixes. When organizations consider serviceability early, release time becomes a checkpoint, allowing them to visualize current readiness, assess risks in a structured way, and empower stakeholders to make informed decisions.

Ericsson acknowledges TM Forum's valuable work on the maturity model for autonomous networks, which provides a solid foundation for automation-related evolution [1]. However, serviceability is not the same as automation. Instead, it is a distinct capability and enabler that requires its own metrics and maturity model. Current industry frameworks do not address this aspect and must be further developed to guide improvements in operational efficiency and customer experience.

Serviceability and artificial intelligence

Good serviceability can be defined as efficiently managing large amounts of network data, configurations, and performance metrics across multi-vendor systems while ensuring seamless integration, quick troubleshooting, and consistent operational control.

Artificial intelligence (AI) and machine learning (ML) are powerful tools that enhance serviceability from the product level to the entire network. They enable the network to proactively detect, predict, and respond to issues and operational changes. By integrating AI/ML-driven workflows into operations and building an AI operations (AIOps) framework, it is possible to establish a more systematic, intelligent architecture that improves network serviceability while strengthening AI tool efficiency and data management practices.

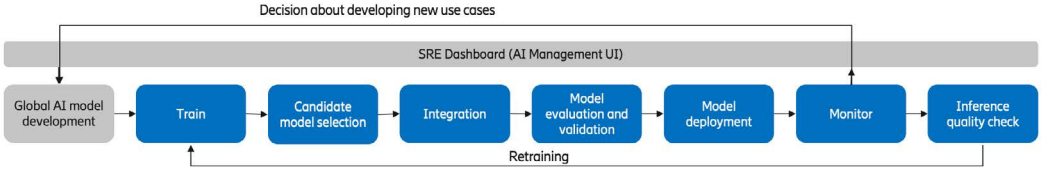


As shown in the image above, the AIOps infrastructure and observability framework ingests massive volumes of logs, alerts, key performance indicators, and configuration data from thousands of nodes and diverse sources across different solution components. It combines expert intelligence with advanced AI/ML techniques, such as time series correlation, reinforcement learning, generative AI, and large language models, to deliver actionable insights. These insights drive full visibility through dashboards, health checks, and automated recommendations. Integrated collaboration enables fast-reacting incident management and closed-loop operations, with seamless handoffs to customer tools and workflows for end-to-end observability and resolution.

External AI agents and partner APIs enhance fault identification and isolation by providing vendor-neutral intelligence and cross-domain correlation in multi-vendor environments. They improve operational visibility and reduce MTTR. To protect data security and intellectual property, and to adopt encrypted data exchange, anonymization techniques, federated learning, and strict access controls, insights are shared without exposing proprietary information or breaching vendor confidentiality.

For initial AIOps adoption that builds confidence and delivers quick wins, simple fault-isolation use cases should be prioritized in reactive mode for well-understood, high-frequency alarms. As AIOps matures, it can progress to proactive stages, focusing on predicting impending failures in critical issues and alarms through anomaly detection and correlation. This can be achieved by reaching high accuracy in recommended root causes and suggested actions from the AI model, with continuous improvement through machine learning operations (MLOps) and daily input from experts' feedback. A closed loop can be implemented for high-confidence use cases, using AI model predictions and detection results.

Online retraining could further improve AI model accuracy. To achieve this, a retraining pipeline is required. The diagram below outlines the AI model life cycle managed via the site reliability engineering (SRE) dashboard, which serves as the AI management user interface. The process requires multiple functions, including training, candidate model selection, integration, model evaluation and validation, model deployment, monitoring, and inference quality checks. It requires not only a general processing unit (GPU) pool but also a data pipeline to ensure data quality, integrity, and coverage.



It is worth adding that not all AI models require online retraining. This usually depends on the model's properties and retraining costs. For example, static models, including small form-factor pluggable fault prediction, that are independent of dynamic factors, such as traffic patterns, may not need online retraining. Model selection impacts retraining costs, as supervised learning demands large, labeled datasets, reducing cost-efficiency.

Basic network data may insufficiently drive innovations such as energy efficiency or closed-loop network healing. External data and AI agents are essential, requiring careful selection of accurate, relevant sources. For energy models, it is best to include mobile and site equipment consumption, air-conditioner usage, and electricity tariffs from power exchanges.

To establish AIOps and MLOps for better serviceability, CSPs and vendors should prioritize high-impact use cases, robust data pipelines, and secure data practices. AI model LCM, seamless workflow integration, and team training are also critical. A good starting point is reactive fault isolation, followed by progressing to proactive prediction, while ensuring scalability with GPU and storage resources.

Conclusion

Good serviceability, along with a continuous, automated, data-driven feedback loop between operational/support teams and development teams to measure, understand, and act on system changes in high-scale telco environments, is the baseline for a robust and cost-effective network. Achieving this requires, among other things, integrating quantitative, machine-generated metrics such as MTTD, MTTR, and incident rates with qualitative human insights, standardizing and structuring telemetry data with metadata enrichment, and using predictive analytics for preventive maintenance to anticipate and mitigate potential system degradations or outages.

In addition, good serviceability advocates for fully digitalized, machine-readable procedures and configuration management governed by automated validation and orchestration pipelines, enabling autonomous deployment and upgrades without traffic impact. Maintaining system state in version-controlled repositories ensures consistency and facilitates continuous improvement. Introducing a serviceability maturity model can guide CSPs from reactive, manual operations to predictive, self-optimizing environments, highlighting the strategic value of a shift-left approach that embeds serviceability considerations early in design to reduce costly post-release fixes.

In conclusion, network operators, vendors, and development teams are encouraged to:

- implement continuous, automated feedback loops integrating both quantitative and qualitative metrics
- apply a shift-left mindset by incorporating serviceability and observability early in the development life cycle to ensure operational readiness and minimize post-release issues
- standardize and enrich telemetry data for real-time system monitoring and predictive maintenance
- automate LCM procedures with machine-readable assets and orchestration to improve serviceability and reduce deployment risks
- adopt version control for system state to support traceability and resilience
- utilize serviceability maturity models to assess current capabilities and prioritize improvements

These actions collectively aim to enhance trust, transparency, and operational excellence while enabling proactive, data-driven management of telco systems.

References and further reading

1. Autonomous networks: in search of best practice – TM Forum benchmark report (December 20, 2024)
2. Cloud-native observability of telecom applications - Ericsson

Authors



Henrik Basilier is an expert in network architecture evolution. He joined the company in 1991 and has worked in a technical leadership and system management role in a diverse set of roles and technical areas. His current focus in on Serviceability where he has a leading role for the strategies and development of automation and related capabilities for the life cycle management for the Ericsson 5G Core Portfolio. Hel holds an M.Sc. in Computer Science and Technology from the University of Linköping, Sweden.



Jörg Aelken is an expert in cloud-native orchestration and life-cycle management whose work focuses on architectures and solutions for the management of telco applications in an autonomous networks context. He joined Ericsson in 1992 and is a member of the BCSS Architecture and Technology organization. He has held several positions in R&D, most recently related to cloud and network automation, including serving as standardization delegate. Aelken holds an M.Sc. (Dipl.-Ing.) in electrical engineering and communication technology from RWTH Aachen University in Germany.



Alla Lokhanko is an expert in serviceability characteristics at Ericsson. She joined the company in 2001 and has over 20 years of technical leadership in telecom product introduction and global deployment. Her current focus is on advancing serviceability across cloud-native systems, ensuring seamless deployment, operation and performance in cloud-native and automated solutions across telecommunications networks. Lokhanko holds a Ph.D. in sociology and a diploma in IT systems engineering from the Kursk State Technical University in Russia.



Peter Wörndle is a senior expert in deployment architectures. Peter is working at Ericsson as a Technology Expert both in Aachen, Germany and Stockholm, Sweden. As a Technology Expert in the area of Deployment Architectures and Serviceability he is responsible for technology evolution, technology strategy and architecture to deliver, deploy and operate Ericsson's products efficiently in various contexts.

Peter has held different positions in R&D and IT within Ericsson working with cloud infrastructure and cloud-native applications in R&D, IT operations and standardization. Currently he focuses on automation and agentic AI for deployment and operations using declarative concepts (GitOps) and pipelines.

Peter has a Master's Degree (Diplom Ingenieur) in Electrical Engineering and Communication from RWTH (Technical University) Aachen, Germany.



King Wai Lee (Andy) is a Strategy Product Manager at Ericsson in Stockholm with over 25 years of telecommunications experience. He leads Cloud and Open RAN service and Healing rApp strategy, focusing on network automation, intent-based operations, and multi-agentic AI frameworks to enable autonomous, self-managing networks across their operational lifecycle.