Ericsson Technology Review

#13, December 2023

rApps: Transforming network management with intelligent automation apps

Charting the future of innovation

Ericsson Technology Review — Charting the future of innovation | #13, December 2023

rApps: Transforming network management with intelligent automation apps

Authors:
Ryan Fitzgerald, Ciaran Johnston

Intelligent automation using rApps has the potential to revolutionize the world of network management and orchestration, augmenting or replacing existing manual processes in order to improve network performance and reduce costs.

ISSN 0014-0171 284 23-3402 | Uen

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

Open radio-access network (O-RAN) architecture is maturing, while the network management paradigm is shifting emphasis from vendor-specific, hands-on network management and operations toward an open ecosystem of intelligent automation provided by vendors and communication service providers (CSPs) alike.

Today, on top of having access to more network data than ever before, CSPs also have fine-grained controls to manipulate the network topology and a high degree of configuration flexibility. These powerful tools drive complexity and limit the CSP's ability to optimize performance and reduce operational costs without a significant increase in the use of automation focused on the CSP's specific business needs. A benefit of this type of increased automation in network management is the enabling of higher levels of abstraction and simplification through intents, supporting a more dynamic business with rapidly evolving goals.

The network automation landscape

Mobile networks are rapidly evolving toward increased levels of heterogeneity and sophistication. Equipment and technologies from different vendors must deliver better performing and more differentiated services than ever before. Cloud technologies and software-defined networks promise new opportunities for CSPs to utilize their commodity hardware and services to achieve their business goals. In addition, network services continue to become more central to the day-to-day workings of modern societies, and even transient service degradations are therefore becoming less acceptable. Banking, e-commerce, entertainment, transportation, logistics and emergency services are growing ever more reliant on guaranteed, high-quality network services. These realities, coupled with increased legislative, environmental and energy-efficiency considerations pose an increasing challenge for CSPs.

Higher levels of heterogeneity and sophistication in networks are leading to greater complexity. Infrastructure can change independently of the software running on it, which means that performance can change over time. A higher number of open interfaces and network functions creates a more

WHAT IS THE O-RAN ALLIANCE?

The O-RAN Alliance defines specifications in areas of radio-access network (RAN) automation, cloudification and disaggregation. The ambition of the O-RAN Alliance is to enable an open RAN by creating a multi-supplier RAN solution that allows for the separation – or disaggregation – of hardware and software with open interfaces and virtualization, hosting software that controls and updates networks in the cloud [1].

In O-RAN, the term rApp refers to an app that has been designed to work on the non-real-time RAN intelligent controller targeted toward the open RAN. The rApp concept can, however, be applied to other domains as well.

Terms and abbreviations

AI – Artificial Intelligence | **API** – Application Programming Interface | **CNF** – Cloud-Native Network Function | **CSP** – Communication Service Provider | **ML** – Machine Learning | **Non-RT RIC** – Non-Real-Time RAN Intelligent Controller | **O&M** – Operations and Maintenance | **O-RAN** – Open Radio-Access Network | **PNF** – Physical Network Function | **RAN** – Radio-Access Network | **rApp** – Non-RT RIC Application | **SMO** – Service Management and Orchestration

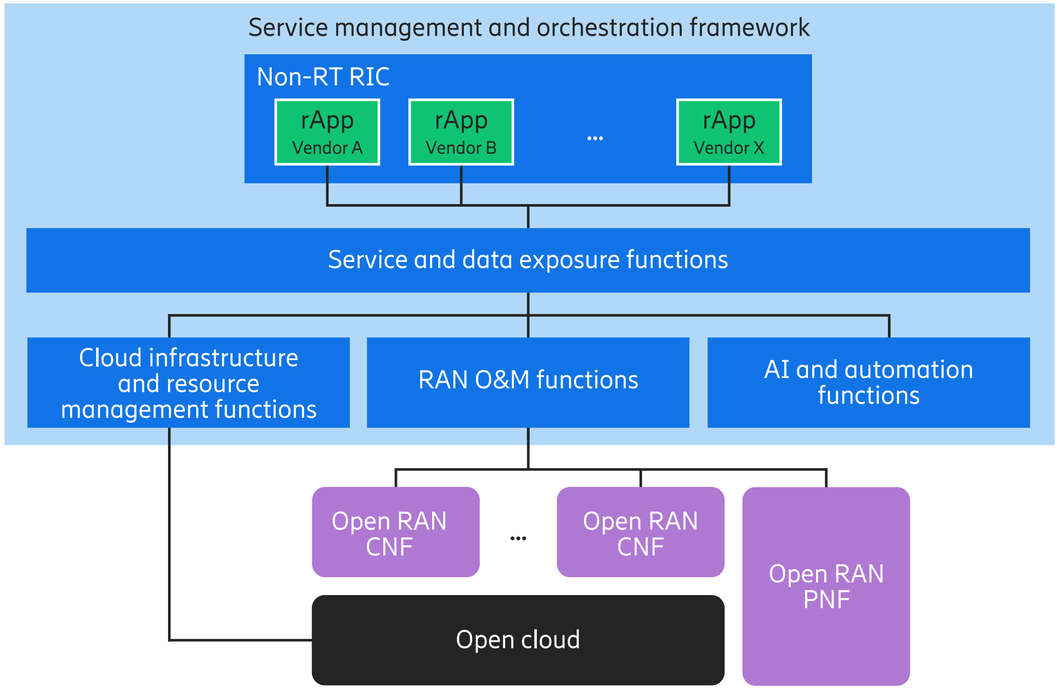

Figure 1: Simplified view of the SMO in the O-RAN context

complex network topology. These network functions are themselves comprised of many small microservices that can be deployed and scaled in different ways.

An increase in the number of network features needed to enable high-quality communication services results in more parameters for vendors and CSPs to configure based on radio network infrastructure performance. More cells and user equipment, together with an increase in dynamic weather conditions and extreme environments (military, industrial and disaster recovery) all lead to an increase in the dynamic nature of the network environment.

Intelligent automation will be required to manage all the complexity and meet expectations while maintaining or decreasing operating costs and capital expenditure. Traditional ways of managing network optimization through static configuration planning and human-centric rollout are no longer sufficient to deliver the network performance required to meet more demanding Service Level Agreements.

To date, the industry has partially addressed these challenges by introducing proprietary SON (self-organizing network) systems performing centralized and closed-loop assurance on top of traditional management systems, with some degree of success.

Intelligent automation will be required to manage all the complexity and meet expectations.

However, such systems are typically heavily reliant on extensive system integration, and expensive to maintain and evolve, as well as often being difficult to manage in coexistence with manual processes or scripted automation. This is especially true given that individual CSPs have different operational needs and are more or less willing to adopt varying levels of automation based on their trust in the technology and operational requirements. CSPs want to be in control of – and in many cases even the creators of – the automation running their network, without the responsibility of managing connectivity, security, conflicts and interoperability issues between vendors.

Modern IT industry best practices have defined a set of architectural patterns and principles to accelerate the development of advanced automation. Standardized security through the OAuth 2.0 protocol and related standards, common deployment patterns using containers, application programming interface (API)-first development using the RESTful (representational state transfer) web service and event-driven APIs with well-defined contracts can all be applied to the creation of well-specified automation enablers. Open-source development in the Cloud Native Computing Foundation, Open Network Automation Platform and O-RAN Software Community provides concrete working code to realize those functions.

rApps and the service management and orchestration framework

The O-RAN Alliance has defined the service management and orchestration (SMO) framework as a key function within the O-RAN architecture. The SMO offers capabilities to orchestrate the deployment of network functions into the open cloud infrastructure and perform fault, configuration, performance and security management for them, while the non-real-time RAN Intelligent Controller (non-RT RIC), a sub-function of the SMO, offers capabilities to enable the implementation of intelligent RAN automation and optimization use cases.

The O-RAN Community and its various working groups are currently developing and agreeing on the specifications that define the SMO in detail. These include the various interfaces and functions exposed within the SMO and the non-RT RIC. **Figure 1** shows a simplified view of the SMO architecture.

The O-RAN Alliance specifies the rApp as a modular application within the non-RT RIC that utilizes the capabilities of the SMO to realize value-adding RAN automation use cases. Because an rApp can be developed, delivered and life-cycled independently of the SMO, and because it utilizes open and standardized interfaces, it can be sourced from different software vendors and interwork with different SMO implementations.

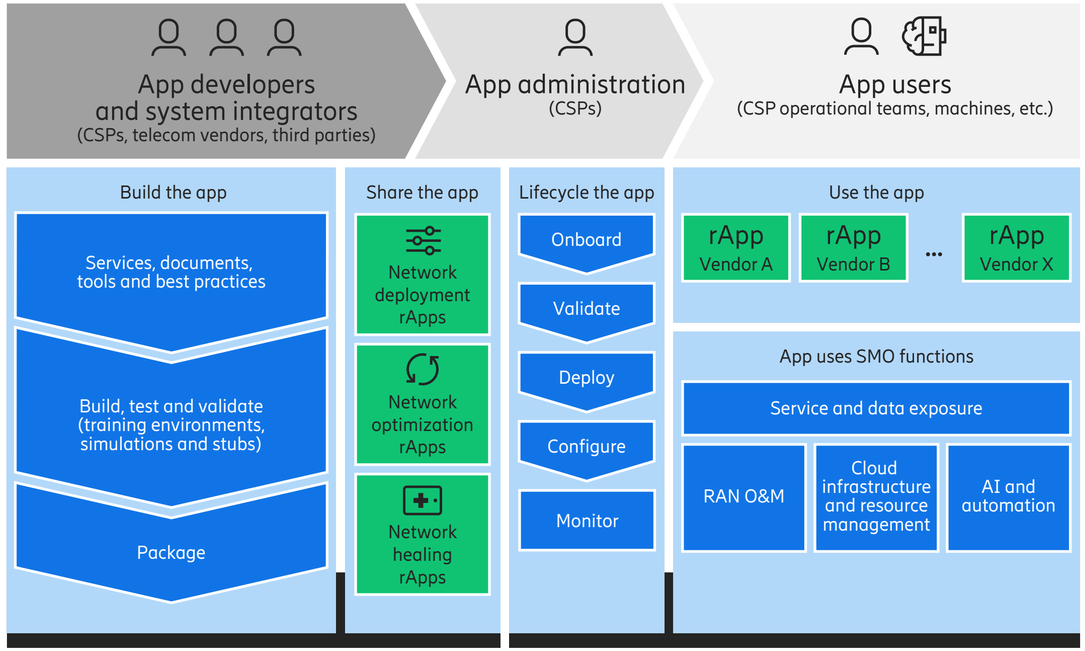

Figure 2: rApp development life cycle

The types of network automation functionality that rApps can implement is extensive. Advanced workload placement and deployment, software canary testing, automated network healing and outage compensation, network performance anomaly detection, configuration validation and optimization are all examples of the valuable automation use cases that have already been proposed.

To achieve these aspirations, rApps will depend on a rich set of SMO capabilities to observe and control the network. Key examples of these capabilities are network inventory and topology discovery, configuration querying and modification, performance and fault data exposure and analysis, and cloud infrastructure and resource management. Furthermore, to help accelerate rApps in delivering smart and intelligent automation use cases, the SMO is likely to provide additional advanced automation supports. Examples of these include artificial intelligence/machine learning (AI/ML), policy, workflow, and intent management frameworks.

The SMO, along with the rApp-based extensibility of the non-RT RIC, represents a paradigm shift in how network automation is achieved. Up until now, automation was built either in an ad-hoc fashion or as separate and costly add-ons to existing proprietary element management systems. Furthermore, automation was difficult to implement because it required specialized knowledge of proprietary interfaces and because each vendor's network equipment (and management infrastructure) presented different technical challenges to overcome.

With the SMO, and the open ecosystem around it, CSPs have much more power to define and control how their networks are managed and optimized. Automation use cases and automation support capabilities are no longer considered as add-ons but are instead a native, built-in capability of the network management system itself, benefiting from significantly better standardization of the observability and control interfaces to the network. Even in brownfield scenarios, existing equipment can be integrated into an SMO environment by applying a consistent set of management principles and adding adapters for the non-standard interfaces.

The SMO also represents a paradigm shift in network operations. As the automation realized by the rApps and the SMO becomes more advanced, the role of the CSP will transform from directly managing the network to managing the automation that enables and supports the network. The measure of success for a management system will shift from how well it assists the CSP in managing the network to how well it runs the network according to the goals and limits that the CSP sets.

While this approach to automation is being standardized by the O-RAN Alliance, it is not specific to the RAN domain; apps can also be built to deliver automation across all the network domains.

The automation development ecosystem

To fully realize the automation potential, it is essential that the rApp development life cycle – from ideation, implementation, onboarding and operation to eventual retirement – is as efficient as possible. The needs of rApp developers, rApp administrators and operations staff using rApps must all be considered. A high-level view of the rApp development life cycle is illustrated in **Figure 2**.

CSPs have much more power to define and control how their networks are managed and optimized.

A broad range of developer and rApp types must be supported. Some rApps will be produced by full-time software developers and have a relatively long lifespan with multiple feature additions and an extended maintenance period. In contrast, other rApps may be created by part-time or "citizen" developers with limited coding expertise, with a focus on automating a particular operational task. Furthermore, some rApps may be long-running and support scaling to handle large workloads, while others may be quite short-lived and only execute periodically to fulfill a particular task or job.

With such a variety of developer and rApp types, the appropriate developer, administrator and operations supports must be in place. This requires achieving a careful balance between enabling freedom and flexibility for the

advanced developer to create sophisticated rApps and providing ease of use and safe guardrails for the citizen developer, who needs to achieve fast results with a much simpler approach (a script, for example). In all cases, security must be of central concern to thwart nefarious actors who may wish to introduce malicious code into the network.

Developers therefore need a rich set of development resources, ranging from easy-to-follow documentation and tutorials to sample code, sandbox test environments and runnable examples. Build and verification pipelines, as well as secure packaging and a "marketplace" to publish, share and distribute them, are also necessary.

The transition from current processes and tools must be carried out in a stepwise manner.

Once rApps are published in the marketplace, administrators who want to utilize them need to have oversight over which of them are to be deployed in their systems, with clear visibility and control over their requirements and dependencies. Administrators need to be able to monitor and control how rApps use system APIs and resources, as well as to easily observe their performance and output.

Network operations staff need to be able to quickly discover, understand and build trust in the rApp automation applicable to their role. They need to know how to smoothly integrate it into their existing management practices and how to replace their existing practices with the rApp automation when necessary. Further, they need to be able to easily observe how automation impacts their network as well as a way to interact with the automation when required.

Automation and conflict management

As the level of rApp automation grows and more rApps coexist in the same environment, there is an increased risk that the automated network operations they perform will conflict with each other. For example, different rApps may attempt to adjust the cell or antenna configuration based on competing goals (such as throughput versus energy efficiency). While this is not a new problem, the flexibility of the SMO architecture makes this issue more important to address. The consequences can range from some rApps being rendered ineffectual, to race conditions and ping-ponging changes that impact network performance and stability.

Some rApps will be developed to coordinate and communicate with each other, thus avoiding such conflicts, but many will not. Those rApps that cannot coordinate and communicate with each other will rely on the SMO's automation capabilities to mitigate or prevent conflicts. In light of this, the SMO architecture should include three types of conflict management functionality:

- Coordination and control
- Conflict detection
- Conflict intervention and resolution.

The coordination and control capabilities will ensure that system administrators and operators can determine where two rApps may be consuming the same network data (the same performance management counters, for example) or modifying the same configuration (the same configuration management parameter, for example) in the same part or in related parts of the network (the same or adjacent nodes). Some of these capabilities may also be exposed to the rApps themselves, so that developers can code their rApp to make use of this data in runtime to avoid conflicts.

The conflict detection functionality will make it possible for the system to observe the control actions that the rApps propose and determine whether they will (or are likely to) result in a conflict. When a conflict is detected, the operator is alerted and given the opportunity to disable or reconfigure the rApps. Since what constitutes a conflict can be highly context-dependent and differ from one SMO deployment or CSP network to another, the decision-making capabilities of the conflict detection functionality must be highly adaptable. A policy-based approach is therefore essential, and AI/ML techniques show great promise in this area.

The conflict intervention and resolution functionality will enable the system to make decisions about whether requested control actions should be blocked or permitted. It can also be used to determine if restorative or repair actions are required. These decisions can be based on a broader range of factors including the impact of the conflict, the relative priorities of the conflicting rApps, operator preferences on guard periods for sensitive configuration management parameters, current network state and other criteria.

With these three conflict management functionality types in place, CSPs can have confidence in the smooth operation of the automation in their network as they build it out.

Transitioning to intelligent automation

It will take time for CSPs and vendors to fully realize the potential of an open ecosystem of rApp-based automation. The transition from current processes and tools must be carried out in a stepwise manner. Firstly, in defining the scope of an automation platform to run multi-vendor rApps, it is important to focus on a small number of prioritized APIs rather than trying to cover every need immediately. Vendors can build trust by delivering on a small set of important use cases and ensuring that the required capabilities are both robust and performant. Secondly, building up an extensive marketplace of rApps will take time and require strong engagements between standardization delegates in the O-RAN Alliance, open-source communities building reference capabilities, CSPs and network equipment vendors across the industry. Last but by no means least, a change management process to help network operations staff adjust to a higher degree of automation will be required.

Conclusion

The rApp approach to automation is a key component of the service management and orchestration architecture in open radio-access networks. Using rApps, communication service providers will be able to overcome a wide variety of network management challenges, delivering significant benefits in terms of operational cost, network service performance and resource utilization. Realizing the full vision will, however, require a deliberative and stepwise approach. Ericsson's comprehensive strategy for the use of rApps is based on our deep understanding of the challenges posed by heterogeneous automation use cases running simultaneously and on our years at the forefront of the development of an open automation platform architecture.

The authors

Ryan Fitzgerald joined Ericsson in 1998 and currently works as a master engineer focusing on operations support systems (OSS) and network automation. He is currently exploring how OSS and network management products can deliver enhanced automation for Ericsson customers. Fitzgerald holds a B.Eng. in computer engineering from the University of Limerick, Ireland, and an M.Sc. in software engineering from the Athlone Institute of Technology, Ireland.

Ciaran Johnston is a senior expert in OSS and programmable network architecture, and he is the chief architect of Ericsson's network management product portfolio. He joined Ericsson in 2000 and has over 20 years' experience in software development and architecture in the OSS domain. Johnston holds a B.Sc. in pure and applied physics from the University of Manchester Institute of Science and Technology in the UK.

References

1. Ericsson – A leader in the O-RAN Alliance ↗

Further reading

- Ericsson whitepaper, An intelligent platform: The use of O-RAN's SMO as the enabler for openness and innovation in the RAN domain ↗
- Ericsson, rApps ↗
- Ericsson, Intelligent Automation Platform ↗