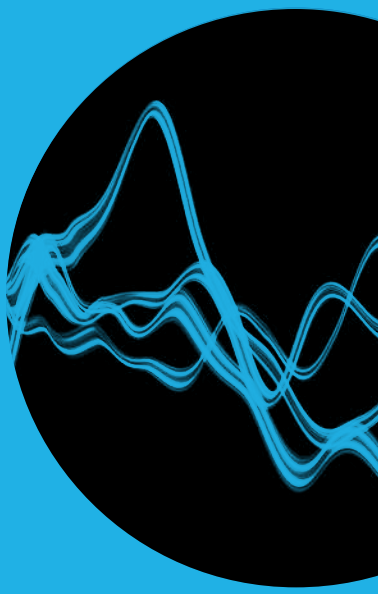
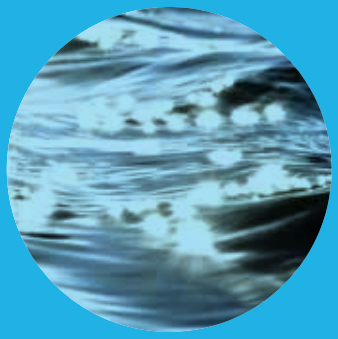
CHARTING THE FUTURE OF INNOVATION | #3·2022

Ericsson Technology Review



ARCHITECTING 5G FOR HYBRID AND MULTI-CLOUD



ERICSSON

5G architecture for hybrid and multi-cloud environments

A unified approach to developing, deploying and operating 5G services – including 5G RAN, Core, OSS and BSS applications – in public and private cloud environments is a key enabler for communication service providers to successfully adopt a hybrid and multi-cloud strategy. The main benefits are faster time to market and lower total cost of ownership.

ANTONIO ALONSO, HENRIK SAAVEDRA PERSSON, HOSSEIN KASSAEI

The amazing growth of cloud infrastructure that we see today is the result of many years of open-source innovation combined with the efforts of hyperscale cloud providers such as Amazon Web Services, Microsoft Azure and Google Cloud Platform, as well as large and medium-size communication service providers that have created their own private clouds in-house.

■ In addition to capacity, cloud infrastructure is growing geographically from national and regional levels all the way to edge and far edge locations, bringing the infrastructure to within a few milliseconds of where the developers want to deploy their applications, which is as close to the consumers as possible. 5G adds ultra-low latency, high capacity and programmable connectivity to such

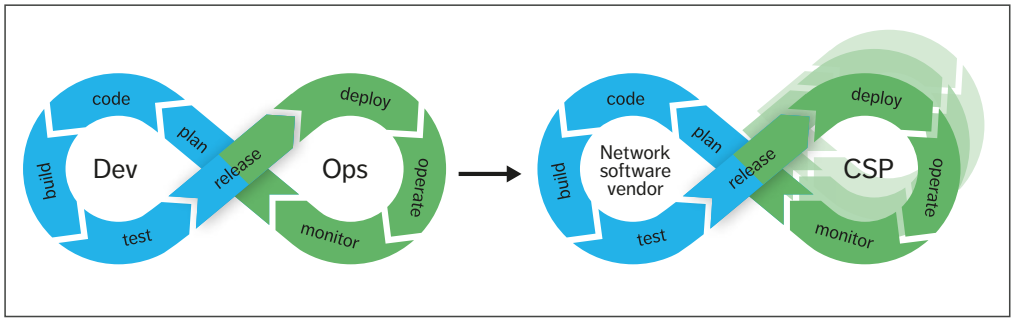


Figure 1 The DevOps and aaP business models

massive-scale infrastructure. The combination of ubiquitous infrastructure and superior connectivity provides application developers with a unique and unprecedented opportunity for innovation.

DevOps in the telecom context

DevOps – a set of practices that brings together software development and IT operations with the goal of shortening the development and delivery cycle and increasing software quality – is often thought of and discussed in the context of a single company or organization. The company usually develops the software, operates it and provides it as a service to customers, according to the software-as-a-service (SaaS) model. Within this context, it is easier to have full control over the entire flow, including full knowledge of the target deployment environment.

In the telecom space, by contrast, we typically follow the as-a-product (aaP) business model, in which software is developed by network software vendors such as Ericsson and provided to communication service providers (CSPs) that deploy and operate it within their network. This business model requires the consideration of additional aspects.

As shown in *Figure 1*, the most important contrasts between the standard DevOps SaaS model and the telecom aaP model are the multiplicity of deployment environments and the fact the network software vendor development teams cannot know upfront exactly what the target environment looks like. Although a SaaS company is likely to deploy and manage its software on two or more different cloud environments, this is inevitable within telco, as each CSP creates and/or selects its own cloud infrastructure.

Implications of a hybrid and multi-cloud strategy

CSPs are expanding their cloud infrastructures through partnerships with hyperscale cloud providers (HCPs) and are increasingly adopting a hybrid and multi-cloud strategy. A recent Ericsson

Terms and abbreviations

aaP – as a Product | **ACK**– AWS Controllers for Kubernetes | **API** – Application Programming Interface | **ASO** – Azure Service Operator | **AWS** – Amazon Web Services | **CNCF** – Cloud Native Computing Foundation | **CSP** – Communication Service Provider | **GCP** – Google Cloud Platform | **HCP** – Hyperscale Cloud Provider | **IaC** – Infrastructure as Code | **IAM** – Identity and Access Management | **SaaS** – Software as a Service | **TCO** – Total Cost of Ownership | **TTM** – Time to Market

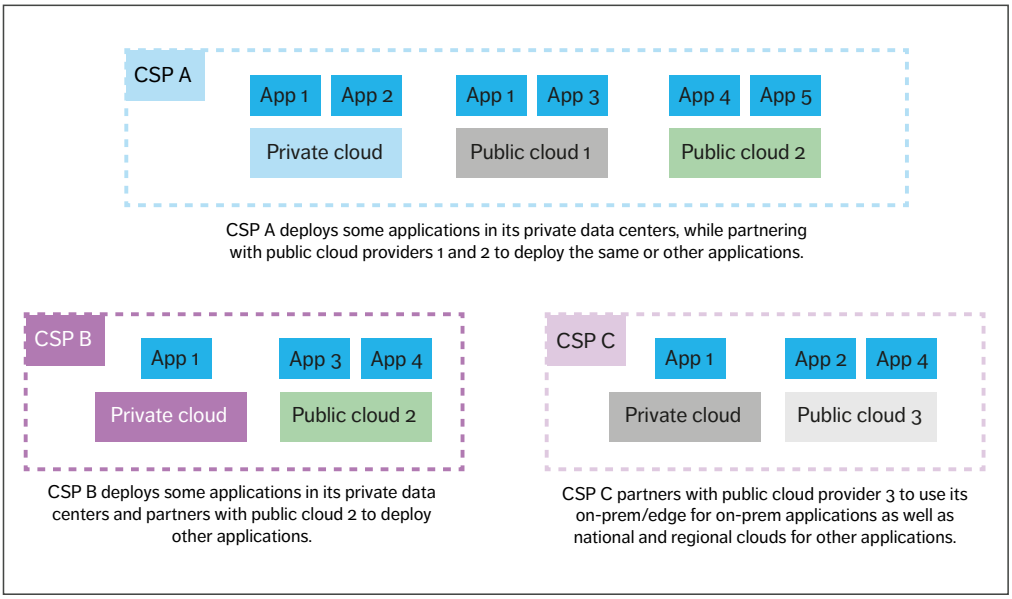


Figure 2 Examples of hybrid and multi-cloud deployment scenarios that applications must be able to support

Technology Review article presented two architecture scenarios for the deployment of telco workloads, showing how telco workloads can be deployed in a dual-stack architecture in a CSP's cloud infrastructure or deployed together with enterprise and consumer applications in an integrated-stack architecture on public clouds [1]. As *Figure 2* illustrates, this approach could lead to a lot of diversity and heterogeneity in the deployment targets for network software vendors.

Kubernetes has established itself as the de facto orchestrator for cloud-native applications. As it is offered as a service by all major cloud providers, it can be viewed as a portability layer for application workloads. In addition, in the integrated-stack architecture, CSPs may also want to utilize other managed services that HCPs offer as part of their partnership. These managed services span across a wide range of capabilities from operational and analytics databases to observability and security services.

Aside from avoiding the cost of a DIY approach, one of the other main advantages of using an HCP service is the ability to delegate responsibility for aspects such as performance, availability and fault tolerance, updates and upgrades, security and scalability to the cloud provider. Another benefit is the ability to create alignment between different network software vendors in terms of data management, observability, security, and other non-functional requirements. However, designing and operating an application that is capable of utilizing such a diverse set of HCP managed services also creates several challenges for network software vendors and CSPs alike.

The main challenges to overcome in a hybrid and multi-cloud strategy are:

1. Maintaining portability
2. Controlling the total cost of ownership (TCO)
3. Optimizing productivity and time to market (TTM).

Portability

As network software vendors must have the flexibility to deploy 5G application workloads wherever CSPs decide it makes the most business sense to do so, it is essential that network software vendors support the CSPs' strategy of adopting a hybrid or multi-cloud approach and enable CSPs to avoid HCP vendor lock-in. HCPs continue to innovate and offer great managed services, but in many cases with different and proprietary application programming interfaces (APIs). One challenge to address therefore is to design portable applications in such a way that CSPs can use HCP managed services where desired, while allowing the decision on which managed service to use to be deferred to deployment time.

Total cost of ownership

Three important aspects influence TCO for CSPs: continuous verification and deployment, monitoring and operations, and security and compliance. In a hybrid and multi-cloud environment, these tasks tend to become more complex given the diversity of infrastructure environments and managed services. Unless these tasks are performed with extreme efficiency, the larger number of tracks to verify and increased operational complexities could lead to higher operating expenses, with a negative effect on TCO. Decisions about which managed service(s) to use and which services the CSP itself will deploy could also have a significant impact on TCO. These decisions depend on the use cases, usage patterns and the application's scale.

Optimizing productivity and time to market

If applications for hybrid and multi-cloud deployments are not developed correctly from the beginning, the long-term developer experience may be suboptimal. Take, for example, an application that is built with one cloud environment in mind that utilizes all the relevant managed services from that cloud. If it is later decided that this application will be deployed in a different cloud environment with a different set of managed services (or with similar services but different orchestration APIs), application developers will have to spend considerable time and effort migrating the application. Time and resources that could otherwise be spent on developing new and enhanced features will be spent on porting and adapting the application to a different environment, which reduces productivity and could result in slower TTM.

A unified approach to development, deployment, security and operations

A multi-cloud-native application is a cloud-native application that can be deployed in different clouds with minimal need for refactoring or redesign, while utilizing managed services offered in each cloud. As such, it must be designed and built with multiple clouds in mind from the beginning. As shown in *Figure 3*, this necessitates a holistic approach to the entire application life cycle from design and development to verification, deployment and operations. To the greatest extent possible, this approach also aims to unify all these tasks across multiple cloud environments.

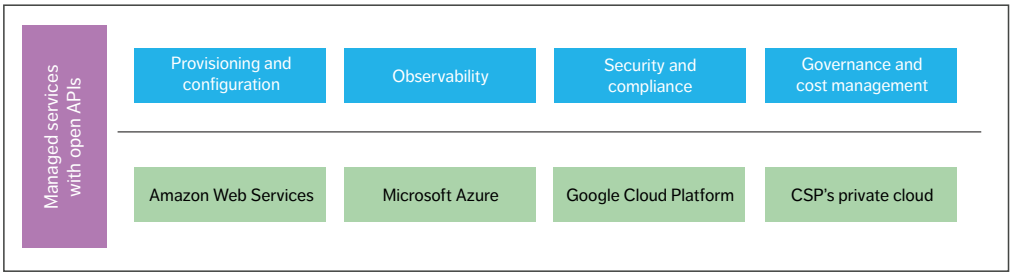


Figure 3 Key enablers for a multi-cloud-native application

> MAJOR CLOUD PROVIDERS ARE TAKING NOTICE OF THE POPULAR OPEN-SOURCE PROJECTS WITHIN THE CNCF

Development based on open APIs

The major cloud providers are taking notice of the popular open-source projects within the Cloud Native Computing Foundation (CNCF) that are being established as de facto standards in their areas and offering them as managed services. The implementations may be proprietary or based on the same open-source projects, but for application developers it is important that they can trust these managed services to support the same open interfaces and protocols. Some examples in the database category are PostgreSQL, MySQL, Cassandra and Redis. Within the observability and metrics-monitoring category, Prometheus and OpenTelemetry are starting to appear as such de facto standards supported by major HCPs.

This positive trend opens the possibility for application developers (typically within network software vendors) to build portable applications that can utilize managed services without locking the application into one cloud environment. This equally benefits operations teams (typically within CSPs) by enabling them to build the knowledge of operating the application once and reusing it in another cloud environment when needed.

In addition, even in scenarios where such managed services may be unavailable – such as on-premises deployments – or not a good fit given the use cases or usage patterns, CSPs can continue to rely on network software vendors to provide the same service as they do now.

Provisioning and configuration of managed services

Careful selection of the managed services that an application will use is only one half of the story. The other half involves harmonizing and unifying the way network software vendor application developers specify the dependency from the application to the managed services, as well as how the CSPs' operations teams provision, configure and manage these services.

HCP managed services have two sets of interfaces:

1. Service interfaces consumed by applications
2. Orchestration interfaces used by orchestrator engines, scripts and so on, to provision and configure the service.

While the first set is based on de facto open APIs, the second set is proprietary to each HCP. In some instances, this can even be the case when a single HCP offers two managed services that provide the same service interface (based on the same open API), but the orchestration of the two is different. In these situations, HCP-managed services that offer the same open-service APIs (such as PostgreSQL, Redis or Cassandra) also provide proprietary and different orchestration APIs for provisioning, configuration and management.

For instance, Amazon has two different managed services that provide the PostgreSQL API – Amazon RDS for PostgreSQL and Aurora PostgreSQL – as well as offering guidance on how to decide which one is the best option based on a list of criteria [2]. It would be an advantage if CSPs were free to choose between the two without requiring the network software vendor to change the application package.

As another example, Google Cloud offers the Cloud SQL service, but it is also developing a new PostgreSQL interface for Cloud Spanner. Providing CSPs with the option of a potential future migration from Google Cloud SQL for PostgreSQL to Cloud Spanner [3] without requiring any change in the application package would be very attractive.

By raising the abstraction level and declaring dependency to a backing service with open APIs (such as PostgreSQL, Redis and Cassandra), application developers can avoid dealing with all the cloud-provider-specific orchestration and

	Terraform	Crossplane
State management	The state is stored in state files and updated by Terraform state command. There is no out-of-the-box and uniform way to handle it in different cloud environments. A typical solution is to use an object store such as S3. Versioning and access control are also closely tied to how the state files are stored.	The state and configuration data are stored in Kubernetes' API server (etcd) as custom objects. A production-grade Kubernetes deployment is needed to properly manage the state for a production environment. Version handling, access control and so on are managed as part of the custom object APIs.
Drift detection and handling	There is no automatic drift detection and reconciliation. This is done manually by invoking the right Terraform commands.	Continuous and automatic state monitoring, drift detection and reconciliation are achieved by utilizing Kubernetes' operator pattern [9].
Abstraction and customizability	Abstraction is achieved by using Terraform modules, usually combining multiple resources from the same cloud provider. Multi-cloud abstractions can also be created. There is no strong separation between platform and developer teams.	Abstraction is achieved by using composite resources and compositions. Multi-cloud abstraction is built into these concepts. The level of abstraction is fully controlled by the platform team that is seen as a separate persona form the developer team.
Developer experience	The developer must be familiar with HashiCorp Configuration Language.	The developer can utilize existing knowledge of Kubernetes YAML Manifest files, or Helm, or other tools already used for deploying the application in a Kubernetes cluster.
Surrounding ecosystem	Terraform has a large and active community that provides a comprehensive set of modules and integrations for many cloud providers.	Since managed service is represented as a Kubernetes custom object, any tool within the ecosystem that integrates with Kubernetes for application workloads can also be used for managed services. This makes for a very rich surrounding ecosystem.

Figure 4 Comparison of Terraform and Crossplane

> UNIFYING THE ORCHESTRATION OF MANAGED SERVICES FROM MULTIPLE CLOUD PROVIDERS IS THE MAIN GOAL

provisioning details, as well as ensuring that the low-level details of orchestrating managed services do not leak into the application layer. The best approach to doing this is to build a common API layer that is fully controlled by CSPs' internal platform teams. This API layer must fully decouple what an application needs from how the underlying cloud providers deliver it.

When it comes to deciding how to implement this API layer, two broad approaches are conceivable. The first is to use infrastructure as code (IaC) tools and frameworks that predate Kubernetes. HCP proprietary examples are Amazon Web Services (AWS) CloudFormation, Azure Resource Manager and Google Cloud Platform (GCP) Cloud Deployment Manager. The most popular HCP-agnostic example is Terraform [4].

The second option is to utilize a more recent crop of IaC tools that are effectively Kubernetes add-ons that can provision and configure managed services external to the Kubernetes cluster and represent them as custom resources to applications running inside the cluster. AWS Controllers for Kubernetes (ACK) [5], Azure Service Operator (ASO) [6] and GCP Config Connector [7] are HCP proprietary examples that all follow this approach.

Crossplane [8], a CNCF project, aims to provide the same capabilities as ACK, ASO and GCP Config Connector, but also add the capability to compose the underlying managed services into higher-level abstractions that fit the needs of a custom-made platform tailored for the requirements of an organization. This makes it a natural candidate for a common API layer that intends to unify deployment and provisioning of managed services from various cloud providers.

Since unifying the orchestration of managed services from multiple cloud providers is the main goal, we see Terraform and Crossplane as two cloud-agnostic options that have the potential to be used in shaping this API layer. *Figure 4* compares the capabilities of the two from various angles.

Observability of the infrastructure and application

Reliance on selected managed services depends on the ability to observe how they are performing and ensure that they are satisfying the Service Level Agreements. HCPs enable this with the help of dashboards that also make it possible to combine observability data from the infrastructure and the application to create a holistic view, which is essential for the application ops team to gain insights in an efficient manner.

In many cases, the collection of observability data is based on HCP proprietary APIs. However, now that HCPs are starting to make use of de facto standard APIs based on CNCF-provided projects, application development teams can benefit from the observability capabilities that the HCP presents without changing how the data is provided. In most cases, however, the HCP will charge based on the amount of data collected – the number of metric samples or log entries, for example.

In a typical telco application where the number of metrics is high and the scraping interval is low, there are two likely outcomes. Either the observability cost will be high or the observability-related data will need to be filtered or preprocessed before it is provided to the HCP observability solution. This could mean that only metrics that relate to important key performance indicators are forwarded to the HCP solution. In cases where a more detailed view is required, the application would still need to provide its own collecting services for metrics and logs to address additional use cases such as machine learning and general troubleshooting. These decisions would be based on the use case and the scale of the deployment.

In a hybrid or multi-cloud-deployment scenario, the problems within the observability areas are

similar to those within the provisioning and configuration of managed services, with siloed cloud vendor tools and a multi-platform monitoring situation. A unified observability system – often referred to as a single pane of glass – is needed to overcome this challenge. This approach is a win-win for both CSPs and network software vendors, as it enables network software vendor developers to rely on well-defined interfaces that expose application metrics, traces and logs, while CSP ops teams benefit from a consistent way of monitoring the applications, regardless of the cloud in which the application is deployed. Removing the need to train ops staff to work with different observability systems also reduces operating expenses.

Several factors influence the decision about where to host the unified observability system. In the case of a hybrid deployment with most of the workloads deployed in private data centers, it may make sense to host it at the private data center as well (with the caveat that extracting data from the HCP may lead to high data egress costs). In cases where the HCP is used for most workloads, the HCP would probably be better suited for the observability solution. The drawback in this scenario is that data from the private data centers needs to be transferred to the HCP, which could create potential issues relating to sensitive data and other legal aspects.

HCPs typically have the capability to ingest observability data from other cloud infrastructures, which allows CSPs to address the multi-cloud deployment scenario. However, moving this type of data across different HCPs typically entails high data egress costs. Initiatives such as the Bandwidth Alliance [10] aim to cut the cost of data transfer between cloud providers and make the multi-could deployment more viable.

There are several external SaaS providers of HCP-agnostic unified observability systems, including Datadog, Dynatrace, Sysdig and Elastic Cloud. Some are mostly based on open-source and CNCF projects, while others are more proprietary in nature. The main advantage of these solutions is that they avoid HCP vendor lock-in even for the observability aspect, making them viable in the hybrid and multi-cloud deployment scenarios. They typically come with small footprint agents to collect different types of data from the infrastructure and the application. By collecting the data as close to the source as possible, some SaaS providers can minimize or even eliminate the data egress cost.

Security and compliance

When utilizing managed services, there is a clearly defined, shared responsibility between the HCP and the application. While the HCP is responsible for protecting the infrastructure so the CSP can offload the burden of operating, managing and controlling the managed service, it remains the responsibility of the application owner to make sure that the managed service complies with applicable laws and regulations. The application owner is also responsible for managing the data, classifying the assets and encryption. This responsibility includes consideration of aspects such as laws and regulations as well as the need for data anonymization related to observability data.

With regard to the integration of HCP-provided security services, the first step is to make use of identity and access management (IAM) capabilities, as this is the way applications gain access to managed services. HCPs have harmonized this by using the Kubernetes concept of a service account that is mapped to the HCP IAM service to provide access to the managed service. From here, deeper integration toward the provided security services can be achieved. The integration would fully utilize and depend on the IAM and key management capabilities that the HCP provides for any authentication and authorization needs, including those between application-specific services. This

> SEVERAL FACTORS INFLUENCE THE DECISION ABOUT WHERE TO HOST THE UNIFIED OBSERVABILITY SYSTEM

may, however, come with some challenges with respect to private keys and root of trust, including the determination of which party is liable for keeping them secure.

Looking further into access control, and especially authorization, different managed and application services typically support a variety of protocols. Different protocols may be required depending on the HCP context in which the application is running. The lack of harmonization within this area forces the applications to be flexible about how the authorization should be addressed. Major alignment is not forecast in this area because several different standards already exist, which means that portability would entail an additional cost for application implementation.

Governance and cost management

No matter which underlying cloud environment is used to deploy the applications, governance and various corporate policies need to be enforced in a consistent way. These policies are typically company specific and could impact various aspects such as selection of services based on cost, availability, performance and other considerations. Each HCP typically offers proprietary services and recommendations for monitoring and optimizing costs, but in a multi-cloud deployment, it is important to harmonize these aspects across different clouds and take the CSP's policies into account. CSPs can either build their own cost management layer or utilize commercial cloud cost management offerings such as Apptio Cloudability and Kubecost.

Conclusion

Communication service providers (CSPs) are partnering with hyperscale cloud providers (HCPs) to deploy 5G services where it makes the most business sense and is mutually beneficial. The result will be a hybrid and multi-cloud future, where 5G services need to be deployed and operated across a wide range of public and private cloud environments. Adopting a uniform strategy in developing, deploying and operating these services

> CSPs CAN CONTROL TOTAL COST OF OWNERSHIP AND ENFORCE UNIFORM GOVERNANCE AND COMPLIANCE

across such diverse target environments is beneficial to both network software vendors and CSPs. By leveraging open source and using HCP managed services that are compatible with their open APIs, network software vendor application developers can build multi-cloud-native applications that are portable across clouds, while utilizing HCP managed services. This, in turn, results in faster time to market for CSPs. On the operations side, CSPs can control total cost of ownership and enforce uniform governance and compliance by unifying important operational aspects such as deployment, security and monitoring, while partnering with multiple cloud providers.

To achieve success, it is important for CSPs and network software vendors to collaborate and promote the development of open APIs for the platform services that 5G applications need and advocate for the adoption of those open APIs by HCPs. The Cloud Native Computing Foundation has demonstrated that it is one of the best positioned communities for such open collaboration and alignment.

Further reading

- **Ericsson, Cloud native applications, available at:** *https://www.ericsson.com/en/cloud-native*
- **Ericsson Technology Review, Cloud-native application design in the telecom domain, available at:** *https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/cloud-native-application-design-in-the-telecom-domain*

References

1. **Ericsson Technology Review, Service exposure and automated life-cycle management: The key enablers for 5G services, December 16, 2021, Svensson, M; Kovács, B; Mueller, E; Maggiari, M; Szabó, R, available at:** *https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/5g-service-automatation-key-enablers*
2. **Amazon RDS for PostgreSQL vs. Amazon Aurora PostgreSQL, available at:** *https://aws.amazon.com/blogs/database/is-amazon-rds-for-postgresql-or-amazon-aurora-postgresql-a-better-choice-for-me/*
3. **Google, New PostgreSQL interface for Cloud Spanner, available at:** *https://cloud.google.com/blog/topics/developers-practitioners/postgresql-interface-adds-familiarity-and-portability-cloud-spanner*
4. **Terraform, available at:** *https://www.terraform.io/*
5. **Github, AWS controllers for Kubernetes, available at:** *https://github.com/aws-controllers-k8s/community*
6. **Github, Azure Service Operator (for Kubernetes), available at:** *https://github.com/Azure/azure-service-operator*
7. **Google, Config Connector, available at:** *https://cloud.google.com/config-connector/docs/overview*
8. **Crossplane, available at:** *https://crossplane.io/*
9. **Kubernetes, Operator pattern, available at:** *https://kubernetes.io/docs/concepts/extend-kubernetes/operator/*
10. **Cloudflare, Bandwidth Alliance, available at:** *https://www.cloudflare.com/bandwidth-alliance/*

THE AUTHORS

The authors would like to thank Tamas Zsiros for his contribution to this article.

Antonio Alonso

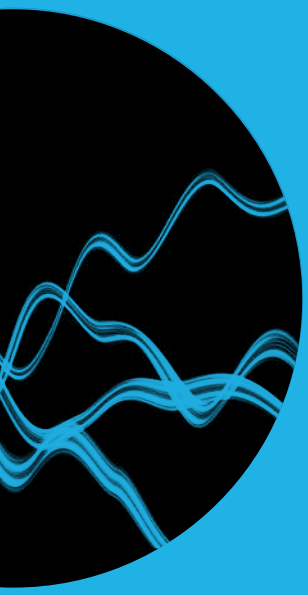
◆ is a technical expert who focuses on cloud-native stateful backing services at Business Area Digital Services. He joined Ericsson in 1995 and has worked on identity and subscription management solutions for different access generations (3G/4G/5G) and domains (circuit switched and packet switched, IMS and identity federation). He joined Ericsson's common development/Application Development Platform (ADP) organization in 2020, where he serves as the technical leader for the data management, backup & restore and life-cycle-management functional areas. Alonso holds an M.Sc. in telecommunications engineering from Universidad Politécnica de Madrid in Spain.

Henrik Saavedra Persson

◆ is a senior principal developer at Business Area Digital Services, where he drives the common cloud-native architecture for the business area's network functions and applications in his role as ADP chief architect. He has been a speaker at KubeCon, NSMCon (Network Service Mesh Conference) and the Open Network Summit, presenting Ericsson's approach to cloud-native architecture and the Cloud Native Computing Foundation. He joined Ericsson in 2004 and has worked in R&D with architecture for both applications and platform products. Saavedra Persson holds an M.Sc. in software engineering from Blekinge Institute of Technology in Sweden.

Hossein Kassaei

◆ is a principal developer and lead platform architect, advocating cloud-native-architecture principles, technologies and best practices in Business Area Digital Services. He joined Ericsson in 2010 and has worked in various software and system design and architecture roles in platform and application teams. Kassaei holds an M. Sc. in computer science from Concordia University in Montreal, Canada.



ISSN 0014-0171
284 23-3375 | Uen

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