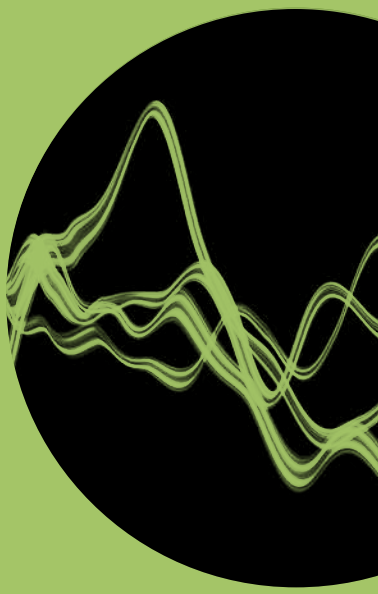
CHARTING THE FUTURE OF INNOVATION | #03·2021

ERICSSON TECHNOLOGY Review



DATA INGESTION ARCHITECTURE FOR TELECOM

ERICSSON

Data ingestion architecture for telecom applications

A fast-growing number of business processes and applications rely on data-driven decision-making and telecom artificial intelligence (AI). As new technologies emerge that make it possible to extract additional value from collected data, the opportunities for enhanced analytics and data-driven, AI-powered decision-making continue to expand. To fully capitalize on collected data, communication service providers need data pipelines that are based on a harmonized data ingestion architecture.

ANNA-KARIN RÖNNBERG,
BO ÅSTRÖM,
BULENT GECER

The purpose of data pipelines is to facilitate access to high-quality data for applications ranging from network management and customer experience management (CEM) to business analytics, product serviceability, artificial intelligence (AI)/machine learning (ML) model training, tailored service offerings and AI for Internet of Things solutions. As many of these applications use the same data sets, a harmonized data ingestion architecture boosts efficiency and makes it possible to focus application development resources on use case realizations rather than on data management.

■ Our harmonized data ingestion architecture is built on the idea that data should be collected once and then shared with any authorized application that needs it. The architecture we propose can be deployed both in customer networks and as a service in application clusters (ACs) external to customer networks. It handles data in accordance with customer contracts using automated procedures and provides security mechanisms with controlled access to original data and de-identified data sets.

Data discovery and the ability to control quality and data life cycle are inherent capabilities of the harmonized data ingestion architecture that build trustworthiness for the benefit of every application that consumes the data. Compliance with standards

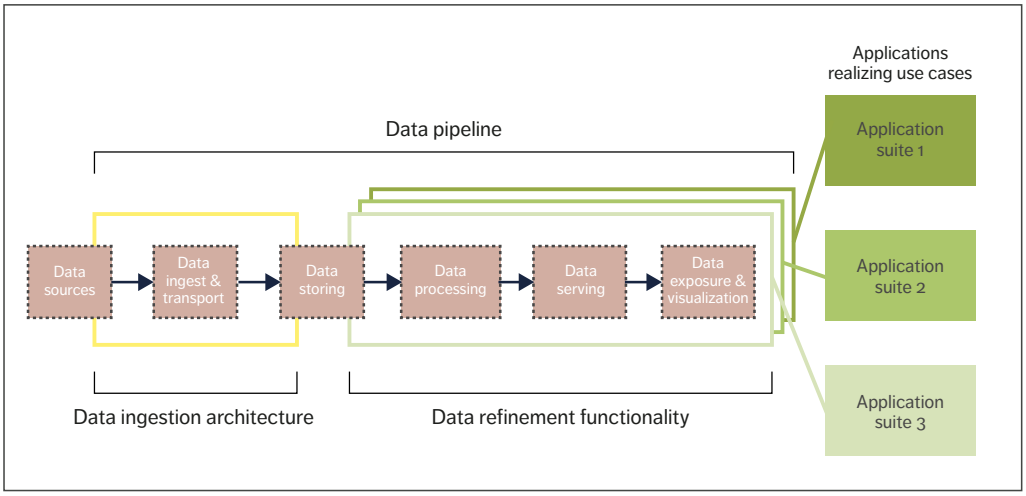


Figure 1 Data pipeline using the harmonized data ingestion architecture

and compatibility with evolving architectures, including ONAP (the Open Network Automation Platform) and ORAN (the Open Radio Access Network), are of crucial importance.

Data pipeline functionality areas

As shown in *Figure 1*, a data pipeline consists of two main functionality areas: data ingestion and data refinement. One conventional approach to data pipelines has been for each application suite to have its own data pipeline, with its own data ingestion and data refinement functionality. To overcome the inefficiencies of this approach, our goal has been to create a harmonized solution for data ingestion in which any collected data is made available for further processing in data-refinement functionality and data-consuming applications. In other words, rather than locking data into one data pipeline or application, we want to make it available to any application suite and its associated data pipeline.

The colored lines in Figure 1 show how the common sets of data collected and made available by the data ingestion functionality (marked in yellow) can be refined in several ways (marked in shades of green) to facilitate and tailor the usage of data in different application suites. At Ericsson, we refer to this approach as "democratizing data."

The benefits of our novel approach to data ingestion include significant opex and capex reductions, increased average revenue per user

Terms and abbreviations

AC – Application Cluster | **AI** – Artificial Intelligence | **ARPU** – Average Revenue Per User | **BDR** – Bulk Data Repository | **CEM** – Customer Experience Management | **DCC** – Data Collection Controller | **DDC** – Data Distribution Central | **DR&D** – Data Routing & Distribution | **DRG** – Data Relay Gateway | **DWH** – Data Warehouse | **EDCA** – Extensible Data Collection Architecture | **GDC** – Global Data Catalog | **ML** – Machine Learning | **ONAP** – Open Network Automation Platform

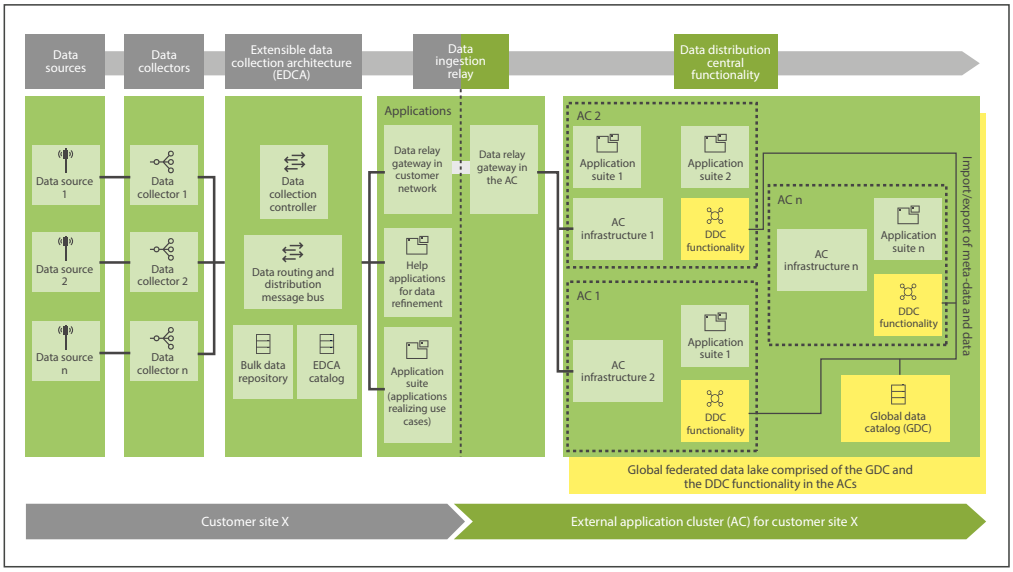


Figure 2 Data ingestion architecture

(ARPU), business process refinement and proactive product support. Opex reductions result from the automation of network assurance, while more efficient network planning and optimization lead to capex savings. The increase in ARPU is due to improvements to network performance and better CEM. Business process refinement comes from increased data quality and faster access to collected data, with the ability to use data collected in near-real time with historical trend analysis. The benefits in terms of proactive product support come from the automation of life-cycle management and serviceability combined with AI-powered decision-making.

Harmonized data ingestion architecture

Put simply, the main objective of our data ingestion architecture is to securely collect data once from the data sources and allow many different consuming applications to use the data in accordance with privacy regulations and customer contracts. These applications may reside in customer networks or be provided as a service and hosted outside the customer networks in application clusters on Ericsson premises or in public clouds, for example.

Figure 2 illustrates the example of a customer network that includes both applications at the customer site (on the left) and applications provided as a service (on the right) as ACs, which are sites where application suites are deployed, typically for a particular purpose such as network operations or product support. Each AC consists of infrastructure and applications where the infrastructure may or may not be cloud based.

Data can be collected from sources for consumption by any application and use case. The collected data is then made available to all applications at the customer site, such as helper applications for data refinement and applications that realize use cases. In accordance with customer contracts, the data ingestion relay transfers data from the customer network to the ACs in Ericsson Cloud using the data relay gateway (DRG). With the help of data distribution central (DDC) functionality and the global data catalog (GDC), the ACs cooperate and form a global federated data lake.

Key functional entities in the data ingestion architecture

The key functional entities in the data ingestion architecture are data sources, data collectors, the extensible data collection architecture (EDCA), the DRG, the DDC functionality, the GDC and the global federated data lake.

Data sources

A data source is a generic term that refers to any type of entity that exports data, such as a node in a telecom network, cloud infrastructure components, a social media site or a weather site. Data sources may export several types of data – including data about performance management, fault management or configuration management – as well as logs and/or statistics from nodes in a telecom network. Data can be exported either by streaming events, files or both.

Data export interfaces will in many cases follow established standards but may also be proprietary. The data can either be pushed from the data source to a data collector or pulled from the data sources by data collectors.

A data source should allow configuration of the data export characteristics, so that it is possible to increase or decrease the data collection frequency, for example. It is not advisable from a load perspective to always collect the maximum number of events from a data source.

From a security point of view, data sources and data collectors should use two-way authentication and encrypt the exported data.

Data collectors

Data collectors facilitate data collection from data sources by using their data export capabilities. They are designed to cause minimal impact to the performance of the data sources. The procedures should be optimized from a data source perspective, and data sets should only be collected once instead of repeatedly. Data can be collected as real-time events or as files in batches depending on the type of data and use cases.

Data collectors make the data available on the EDCA for consumption by multiple applications. It is easy to avoid collecting data that is already available by first checking the EDCA catalog to see which data sets already exist before deploying a new data collector. In this way, data collectors register metadata about their collected data in the EDCA catalog, publish data on data routing and distribution (DR&D) and optionally persist data in the bulk data repository (BDR). Notifications of collected files stored in the BDR are published on DR&D to any consuming application that subscribes to the collected files. The notifications contain references and an address to the BDR that stores the file.

The data collectors collect raw data, and there is an option to apply filtering on the raw data before it is published on DR&D. Further refinement of data can then be tailored for different data pipelines and application suites by EDCA helper applications that implement data-processing and data-serving pipeline functionality, making data ready for visualization and exposure.

Data collectors and EDCA helper applications plug into the EDCA and will evolve according to use case requirements on data pipelines to serve the application suites. It is also possible to integrate data collection from other vendors' data sources with the EDCA.

Extensible data collection architecture

The DR&D function in the EDCA is tasked with receiving data publications from data collectors and delivering these to the consuming applications that subscribe to them. Data on DR&D can be streaming events or notifications of files persisted in the EDCA's BDR. Applications that process the data they receive from DR&D can publish their insights (processed data) back to DR&D, so other applications can benefit from them.

The message bus functionality in DR&D can be realized with one or more message bus technologies, such as Apache Kafka for ONAP compliance, but DR&D is not limited to Kafka. Abstractions are used to shield producers and consumers from the message bus technologies. The DMaaP (Data Movement as a Platform) service wrappers in ONAP are one example of such an abstraction.

> DATA SCHEMAS CAN BE USED TO DYNAMICALLY LOAD MODELS INTO CONSUMING MODEL-DRIVEN APPLICATIONS

DR&D primarily provides the functionality to publish, subscribe to and deliver data, but it can also persist real-time streaming events published on DR&D to enable historical analytics.

DR&D is agnostic to the format of the transferred data. The EDCA catalog stores references to metadata specifications for data sets published on DR&D to facilitate the interpretation and consumption of data by applications. Metadata specifications contain the data schemas (models), and these can be used to dynamically load models into consuming model-driven applications.

The EDCA catalog enables data consumers to discover available data, with instructions on how to access it for consumption. Data must be interpreted to be consumed, and for this purpose the catalog also contains references to metadata specifications for data sets available on the EDCA. Examples of the kind of metadata stored in the catalog include:

- type of data, such as performance management data
- class of data (raw data or insight)
- source of origin
- data quality indication
- data lineage information, including all transformations that have occurred
- location
- data (set) reference
- logical name (URL) for data consumption (DR&D topic)
- schema and serialization method, such as Avro
- metadata specification references, such as references to schemas stored in a schema repository.

The BDR provides the capability to store large amounts of structured, semi-structured and unstructured data – that is, data sometimes referred to as big data. The BDR exposes an (application programming) interface to allow consuming applications that have been notified on DR&D to retrieve a file by using the address and reference included in the notification.

Note that the BDR is not a traditional data warehouse (DWH) intended for long-term data storage. If consuming applications need longer-term storage, subsets of the data should be copied from the BDR and stored in application-specific, long-term DWHs.

The data collection controller (DCC) plays a notable role in the EDCA. Its purpose is to instruct the data sources on how to export data: specifically, it tells them if they should perform basic or extended export. If several applications request extended data collection, it is also the DCC's responsibility to determine when basic data collection can resume and to instruct the data sources to do so. The DCC works in conjunction with data collectors dedicated to specific data sources and implements the configuration of these according to the interfaces and capabilities provided by data sources.

A use case example where the DCC is involved for extended data collection may look something like this:

1. An application interrogates the catalog and discovers that extended data collection from data source 1 is possible but deactivated.
2. The application requires access to the extended set of data and requests that the DCC initiate extended collection through the data collector for data source 1.
3. The DCC supervises the extended reporting.
4. The DCC switches off the extended data collection when no applications require it.

The data relay gateway

The DRG provides functionality to transfer data from a customer network to an external AC in accordance with customer contracts and legislation.

It provides a generic solution for any collected data in a customer network and works in harmony with applications in the customer network that require access to the same data.

The data ingestion relay functionality in the DRG functional entity is placed in the customer network and in an external AC. These are named DRG-CN (customer network) and DRG-AC (application cluster). DRG-CN implements data export policies in accordance with customer contracts and regulations, and it transfers data to DRG-AC, which is the entrance to the ACs. DRG-CN is installed per customer and DRG-AC can be installed per AC or be shared by several ACs.

The solution centralizes the data ingestion relay functionality into one scalable functional entity (DRG-CN) that implements data export policies for any data transferred from customer networks to an external AC (DRG-AC). This avoids duplication of this functionality in applications that collect data in customer networks when the same data should be exported to external ACs.

The DRG-CN connects to the EDCA to access both streaming data and files as soon as these are available. In fact, the DRG behaves like any EDCA application in the customer network but with a single purpose: to transfer data to external ACs where it can be used for many purposes, such as training AI/ML models, serviceability, analytics and proactive product support, and so on. Together, the DRG-CN and DRG-AC form a trusted and secure mechanism to exchange data where, for example, one secure VPN tunnel can be used for all data collected by data collectors in customer networks, and for applications in an AC.

The DRG-CN also connects to a database that stores customer contracts according to a defined information model that expresses the written contracts in a formal language that can be used to execute policies in the DRG. The DRG only transfers what it is allowed to transfer according to the contracts, and when it performs a transfer it does so in the manner specified by the relevant contract(s). This means that some data elements must be de-identified with the help of data refinement functionality before they can leave the customer networks. Examples of such data elements include customer identities such as IMSI (the International Mobile Subscriber Identity) and MSISDN (the Mobile Subscriber Integrated Services Digital Network).

Data distribution central functionality

DDC functionality facilitates cooperation between ACs, enabling them to discover and share data, thereby forming a virtual federated data lake. DDC procedures can be grouped into two sets: one with procedures for exchanging metadata, and the other with procedures for transferring data. The first set of procedures is used to discover data, specify requested data and receive notifications of updated data. The second set is used for data transfer and specifies how data sets can be exchanged with details needed for such interactions (with SFTP (SSH File Transfer Protocol) according to schema X, addresses, ports and so on). The DDC procedures enable ACs to automatically discover and consume data that exists in other ACs, as long as the customer contracts allow it.

The global data catalog and federated data lake

The GDC contains all the data in all of the ACs, including metadata regarding orders, customers, products and customer networks. Together with the DDC functionalities of all the ACs, the GDC creates a globally federated data lake. The GDC also serves as the root for discovering data available in the data lake.

Access to all subsets of data will be governed by policies based on contractual rights, with the full set of data visible only to applications and people with the right security classification.

> THE DRG ONLY TRANSFERS WHAT IT IS ALLOWED TO TRANSFER ACCORDING TO THE CONTRACTS

Helper applications for data refinement

Data refinement functionality can be implemented with a set of helper applications that process the raw data available on the EDCA. The purpose of having helper applications is to provide supporting functionality for the applications that realize use cases. The helper applications for data refinement shown in Figure 2 realize data pipeline functionality for data storage, data processing, data serving, data visualization and data exposure.

Other examples of helper applications include:

- security helper applications for de-identification of public data and non-public data
- ETL (extract, transform, load) helper applications for the transformation and preprocessing of raw data into filtered and aggregated data sets
- other repositories to persist structured data for data consumers (applications) with certain needs, such as graph databases for knowledge management.

Helper applications have the ability to offer refined data back to the EDCA for consumers to access it – for example, to applications realizing use cases (CEM, for example) or for exposure in service exposure frameworks. Further, applications that realize use cases and process the refined data to produce insights can also publish those insights in the EDCA for consumption by other applications realizing different use cases. The access rights to the insights and any other data set in the EDCA will be controlled with proper security mechanisms for authentication and policy-based authorization.

Conclusion

Modern intelligent applications depend on the availability of large amounts of high-quality data. In the telecom industry, where many of them rely on the same data sets, gathering the same data for different purposes is a waste of time and resources. In an increasingly data-driven world, it does not make sense to lock data into a single data pipeline or application. On the contrary, collected data should be available for use in the data pipeline of any application suite that needs it. At Ericsson, this is what we mean by democratizing data.

> “COLLECTED DATA SHOULD BE AVAILABLE FOR USE IN THE DATA PIPELINE OF ANY APPLICATION SUITE THAT NEEDS IT”

Our research shows that by harmonizing the data ingestion architecture of data pipelines, it is possible to ensure that the necessary data for all applications, regardless of where they are deployed, is only collected once. We can then make the data available to any application that needs it with the help of a data relay gateway. This secure and efficient solution provides the significant benefit of freeing up application development resources to focus on use-case realizations rather than data management.

THE AUTHORS

Anna-Karin Rönnberg

◆ is an AI and data expert with broad experience ranging from systems management to portfolio management. She joined Ericsson in 1985 and currently serves as a portfolio manager within the CTO office. In recent years, Rönnberg's work has focused on enabling a more data- and AI-driven strategy and portfolio. She holds an M.Sc. in electrical engineering from KTH Royal Institute of Technology in Stockholm, Sweden.

Bo Åström

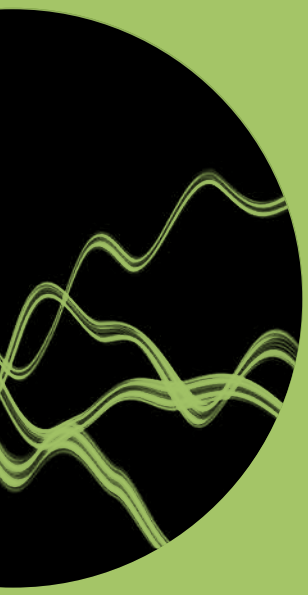
◆ is an expert in system and service architectures within Business Area Networks. He joined Ericsson in 1985 and has extensive experience in radio networks, core networks and enterprise service networks, where he has worked with standardization and product development. Earlier in his career, Åström held technology specialist roles in the areas of interfaces and protocols, messaging architectures and network architectures. He holds more than 70 patents.

Bulent Gecer

◆ joined Ericsson in 2001. He is an expert in data management and incident analytics within Ericsson's CTO office and has played a leading role in developing data-management- and analytics-related architectures at corporate level for the past 10 years. Gecer previously worked as a technology specialist and software developer within core networks. He holds an M.Sc. in engineering physics from Uppsala University in Sweden.

Further reading

- **Ericsson, Autonomous networks, available at:** *https://www.ericsson.com/en/future-technologies/autonomous-networks*
- **Ericsson, Future network security, available at:** *https://www.ericsson.com/en/future-technologies/future-network-security*



ISSN 0014-0171
284 23-3356 | Uen

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