Ericsson Review
The communications technology journal since 1924
2014 • 4

Data without borders: an information architecture for enterprises

April 24, 2014

ERICSSON

Data without borders: an information architecture for enterprises

Enterprise information needs to be relevant, available and all parts of an organization need to share a common understanding of it. As the significance of enterprise information and business agility rises, an information architecture that can capitalize on the changing nature of information, how it is generated, and how it is consumed, is an important enabler for business evolution and growth.

LARS ANGELIN, JATIN SURI, MUNISH AGARWAL AND AKSHAY MEHRA

For most enterprises and industries, information is an asset. It is the raw material used to develop product strategies, roadmaps and marketing campaigns – a key element of modern business practices.

The way people and businesses create and share information has changed considerably. The mass-market adoption of the internet, social media, apps and gaming – as well as the devices that have made non-stop information sharing and gathering possible on a global scale – are just a few of the factors that have contributed to this change.

Consumer insight, for example, is no longer limited to information gathered by a business in its interactions with customers. Today, knowledge about customers, their needs and their preferences can be assimilated from a wide variety of data sources, including networks, social media platforms, subscription databases, and application interfaces. The analysis that creates useful information from this raw data can be monetized and offered as a service to other enterprises, but, for most businesses, good information management is an essential component of customer care and daily service provision.

The growth in machine-to-machine (M2M) communication is yet another contributor to the richness of information. M2M communication provides enterprises with remote access to real-time information accumulated by a range of mobile and static devices and sensors – information that can be used to optimize existing offerings or identify service gaps.

The availability of real-time location data has been one of the more significant game changers. This type of information offers developers a basis for building sophisticated applications with a higher level of subscriber interest. Map applications, for example, deliver services that provide answers to basic questions such as "Where am I?" to more complex queries, such as "Where is the nearest gas station that sells fresh sandwiches?"

Information virtualization

Traditional approaches to IT and systems architecture are also changing – moving toward greater flexibility to enable business evolution. This change is in turn shifting the direction of system design toward greater virtualization and higher levels of abstraction.

For telco operators, service virtualization of networks and network functions is a key part of ongoing network transformation. Much of the work on software-defined networking (SDN) and Network Functions Virtualization (NFV) relates to the functional perspective of virtualization. Somewhat less discussed, but just as important, is the management and virtualization of information. Similar to the way that SDN, through abstraction, separates control (management) from the data plane, information virtualization separates functionality from information.

Nature of information

Information is diverse. Many different types of systems exist for storing, using and modifying information, and many different semantic models and formats are used to describe it. Even within the same enterprise, information is not necessarily structured in a harmonized way – there are for example, several ways to refer to payroll, which could be defined as wages, payroll or salaries.

BOX A Terms and abbreviations

BSS	business support systems
CDN	content delivery network
CDR	Call Detail Record
CEP	complex event processing
CIME	canonical information model of the enterprise
CRM	customer relationship management
M2M	machine-to-machine
NFV	Network Functions Virtualization
NoSQL	not only Structured Query Language
OSS	operations support systems
RAM	random access memory
RDBMS	relational database management system
SDN	software-defined networking
SOA	service-oriented architecture
TCO	total cost of ownership
TTM	time to market

The significance of this lack of coherence is most apparent when it comes to sharing information – when, for example, the same information is used by multiple applications, or across different technologies, between different vendor solutions, or across the internal boundaries of an enterprise. The complex protocols developed by the telecom industry are testament to the constant need for information transformation.

Traditional ICT architectures tend to be function-centric. As shown in Figure 1A, they are made up of discrete applications that perform specific network functions – each one with its own semantics, formats and storage capabilities. This architecture can be visualized as many small information islands that have little concern for each other. Unfortunately, one of the consequences of this approach is cost; integrating applications that need to share information and separating suboptimal storage utilization is time-consuming and can be quite complicated. Here are some general observations related to evolution of systems architecture:

- **the growing significance of information is not a good fit with the function-centric architecture of traditional systems;**
- **the greater the need for information sharing, the higher the integration costs;**
- **to make information reusable and sharable across the borders of a system, it needs to be part of a common and shared information model;**
- **information transformation is undesirable but unavoidable – it will not disappear completely;**
- **to be of value, information must be trustworthy and correct.**

From a software architecture perspective, if information is already available, known and in a readable format, it is a fairly straightforward process to build an application that manipulates or analyzes the data in some way if it is already accessible. However, if the starting point for application development is to fulfill a specific function without much knowledge about what information is available, then the process of accessing the right information and integrating it with the application is likely to be difficult and costly. With an information architecture in place, whatever information is available is known, and so no translation is required.

FIGURE 1A **The enterprise information challenge**

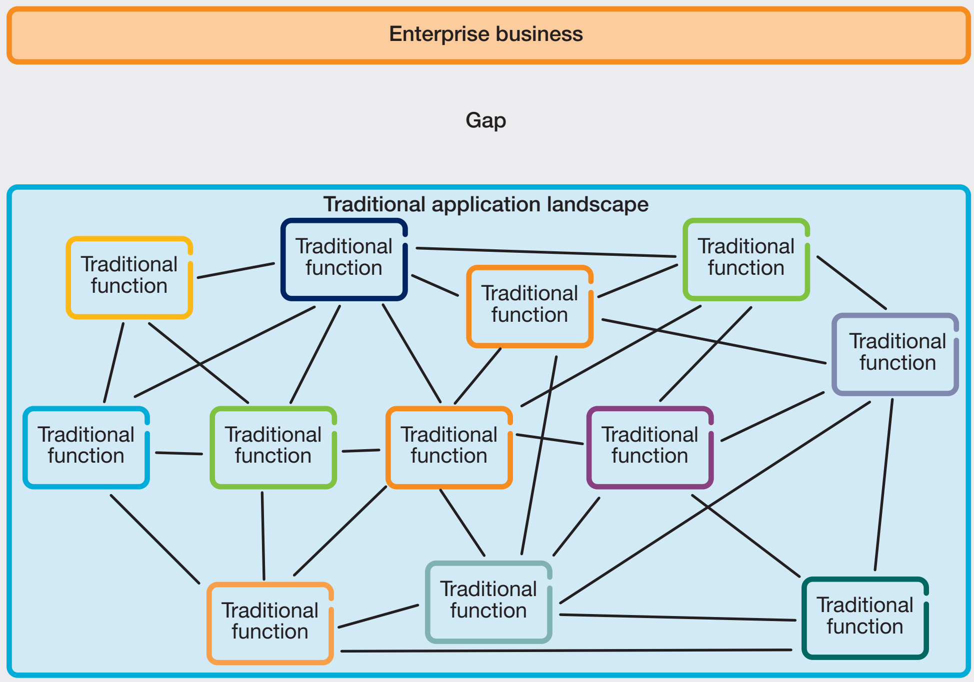


FIGURE 1B **Overcoming the enterprise information challenge**

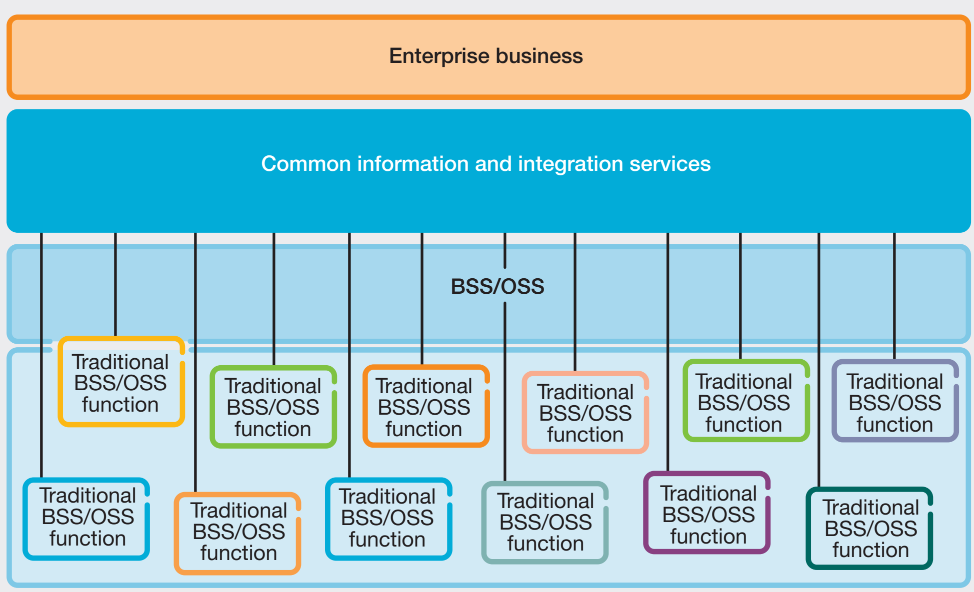
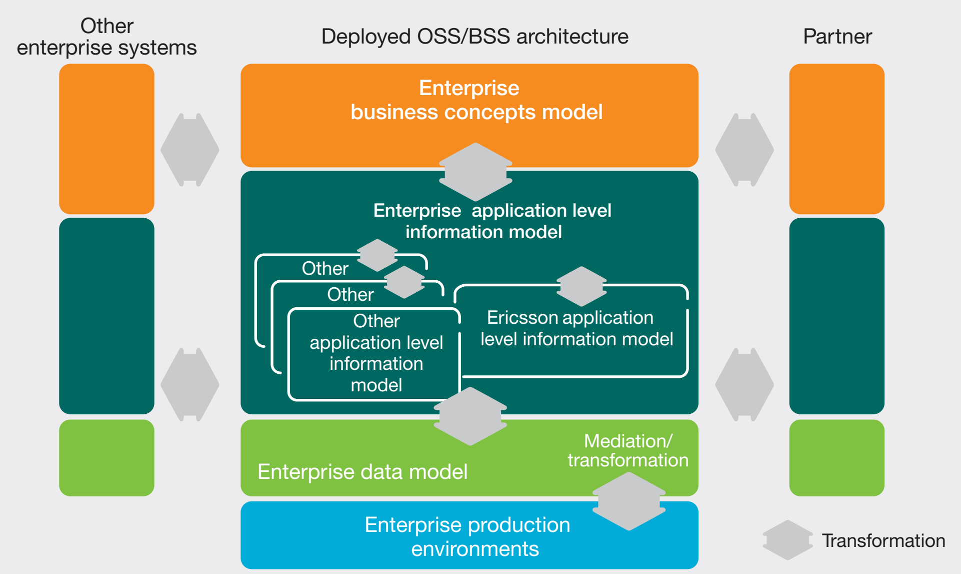


FIGURE 2 **Enterprise architecture**



Information on the inside

Enterprises typically have many integration points:

- with partners, customers, suppliers and regulators on an advisory, commercial or technical level;
- among enterprise systems; for example, back-office financial systems need to share information with supply, delivery and ordering, as well as with HR and payroll; and
- with production systems – such as technology- or vendor-specific networks and content delivery networks (CDNs).

Generally speaking, an enterprise defines the information it needs to run a profitable business as well as the characteristics of that information (static, structured, event-driven, real-time, transactional or streamed). The spread of information across any given enterprise is often extensive, spanning many different functional areas including marketing, ordering, strategy, HR, production, and finance. Similarly, the range of applications that use information is diverse; including for example, CRM, back-office support, payroll, and resource planning.

The challenge comes in connecting business processes to support applications and production or delivery systems – a challenge that is rooted in the fact that rules, characteristics, information and procedures are not usually harmonized with business processes.

The consequences of this gap are reduced or no agility, lengthy TTM, long time to completion, and high TCO, all of which reduce the ability of a business to evolve. A step toward overcoming this challenge, as illustrated in **Figure 1B**, is to use a single function to serve all enterprise applications with information – making that vital connection between information and processes.

Enterprises tend to be made up of a number of domains, each with their own set of information semantics, goals and characteristics defined at both the business level and application level. The business level, at the top of the semantic information model, includes the enterprise vocabulary and concepts that describe the interactions between the people and systems involved in business processes. The application level is part of the internal value chain of an enterprise. Activities at this level tend to be disjointed and are as such regarded as a set of independent domains, in which each application manages its information, tailored to its particular needs.

To help make the connection between business processes and supporting applications, the information at both the business and application levels needs to be aligned and connected. However, achieving this is not a simple task, with a number of factors contributing to its complexity, including:

- typical telco operators manage several hundred, even thousands of, applications and systems that vary in nature – including OSS/BSS, networks, data centers, content systems and cabling;
- applications, when viewed from an information perspective, are islands that are likely to have been delivered by different vendors and often support different technologies;
- integration is needed to get applications to cooperate and perform business tasks – the more applications involved in a task, the more integration connections are needed, the greater the cost and the longer the time to completion;
- many applications provide similar but not exactly the same functionality, they may use the same or related information – this situation creates synchronization and information mastership issues;
- building new functions on top of multiple applications that use different information models is a challenge; first a common information model is needed to translate information from one format to another so that it can then be shared;
- applications are created and terminated in line with evolving business needs;
- information and functions change independently of each other; and
- the trustworthiness, correctness and relevance aspects of information always need to be maintained.

Information mediation

As **Figure 2** shows, information originates from many different sources and is consumed by a wide variety of users (people, applications, domains and systems) that can be either internal or external. If information at the application level is federated from many different sources it tends to be more useful.

Inside OSS/BSS

Taking OSS/BSS as an example, the integration points include:

- horizontal integration within the OSS/BSS to support communication among applications of different generations, technologies and vendors;

- northbound integration to support communication with an enterprise's semantic information models; and
- southbound integration to support communication with production domains.

Toward an information architecture

The transition to an information-centric architecture is shown in Figure 3. On the left hand side, mediation plays a strong role and the majority of applications have their own information model. While on the right hand side, the majority of applications satisfy their information needs through a common information model.

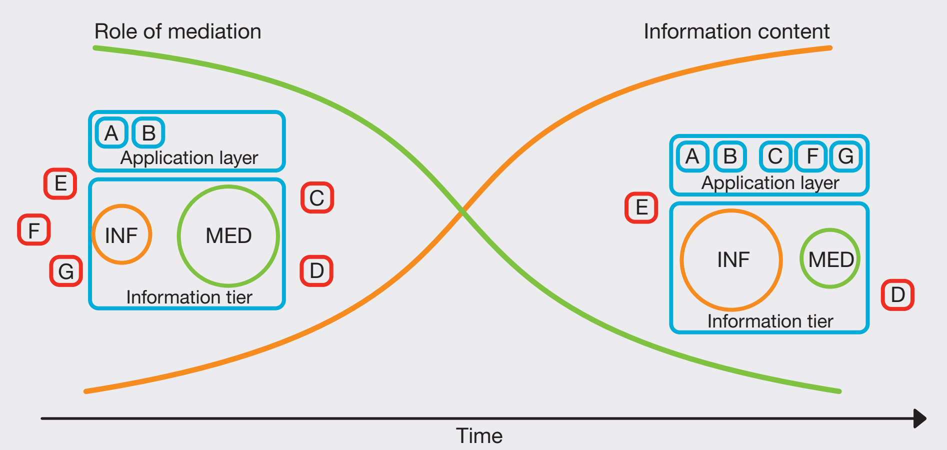
The new model is built using emerging software architectures and makes use of available information. In this way, enterprises can take advantage of new business opportunities as they arise.

The foundation of the information-centric model is business-level information that is fully owned and managed by the enterprise, together with a set of applications that support business processes. In traditional architectures, information is piggybacked and hidden inside applications. In the new model, systems recognize information-related functions as functions in their own right.

In the new model, the set of information functions are located in one information environment – the information tier – that serves the entire enterprise. The available information set, which in effect is the common and shared language used by applications and business users is defined by the enterprise for the application layer. In practice, the information set is a federation of existing information sources and is referred to as the canonical information model of the enterprise (CIME). The CIME includes many different information types, such as static, structured, analytics, real-time data, and streamed.

With full life cycle management, the enterprise can control and maintain governance – from creation to termination – for every information object in the CIME. Full life cycle management supports all information events including creation, publishing, accessing, formatting, and storing, allowing the enterprise to evolve its information set as its business develops.

FIGURE 3 **Transition to information-centric systems**



The information tier offers information to users as a service. How and what information can be accessed is published in an information catalog, including any restrictions that may apply. These restrictions can also be set through the information tier, allowing enterprises to set the appropriate level of quality, integrity and security for individual pieces of information.

As enterprises move deeper into the information-centric model and take full control of their information, applications become reduced to the level of information consumers and delegated creators. There are a number of benefits to building systems with an information tier and CIME:

- lower integration costs;
- smaller applications – as the information architecture delivers many of the information services;
- fewer applications – the same application can be used to perform the same task on different information sets. For example, B2B and B2C invoicing are typically two separate functions today;
- speedier integration with external systems and other information sources;
- faster introduction of new services and business functions;
- flexibility for best business support; and
- information quality control.

Services of the information tier

The information tier is built with software components that offer services in a service-oriented architecture (SOA) manner using a layered responsibility structure. This approach provides scalability, allows for introduction of new functions, and supports partitioning. The information tier is event-driven, and its behavior is controlled by enterprise-defined rules and policies – supporting simplified adaption to changing business needs. All management and governance functions are centralized to gain better overview, common management style, one management console.

Some of the characteristics and services of the information tier include:

Information as a Service – all information objects are offered as a service from a central catalog.

Immutable objects – when information changes, instead of overwriting the existing stored object, a new instance is created with its distinct time stamp.

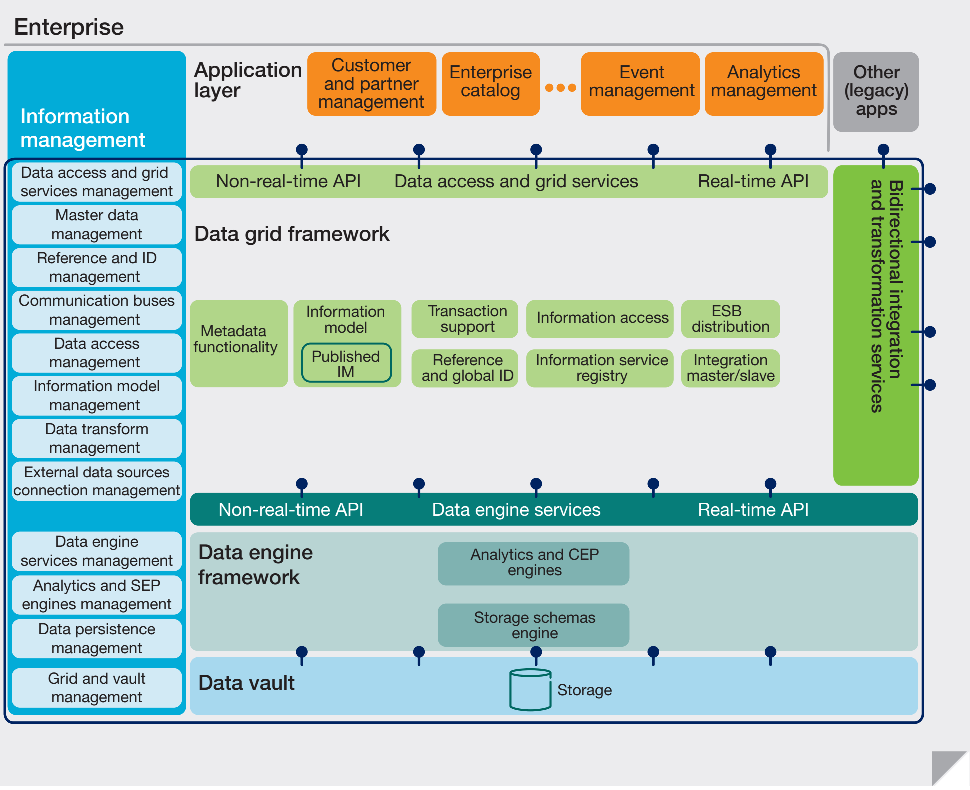
Analytics – as all data passes through the information tier, it is the natural place for analytics and event processing functions.

Two-way connection – to simplify the integration of information coming from, say, an external data source or a legacy application, the information tier includes a bidirectional mediation and connection handling service.

Partitioning – is sometimes needed for legal, business, tenancy or performance reasons. The information tier can be partitioned by information model, by storage, geographically or logically.

Storage – is a generic function that serves the entire CIME. Such as ≫

FIGURE 4 Information architecture



storage and persistence services for applications that require local data.

Information architecture

The proposed information architecture comprises three layers – a data vault, data engine and a data grid – that are governed by an information management function. The detailed architecture proposal is shown in Figure 4.

The data vault offers different storage types, including disk, random access memory (RAM) or virtual resources. The data engine contains a set of components that provide access to the data vault (storage), using a variety of techniques such as RDBMS, graph databases, or NoSQL. The information owner decides which type of database best fits their data access and persistency requirements. The data engine is responsible for executing analytics and complex event processing (CEP).

The data grid is responsible for information creation, mediation and ensuring that the appropriate storage requirements are met. The data grid exposes information services to information architecture users. The applications that consume and produce information sit on the top of the architecture, or to the side of it. Each application places requirements on the grid layer for the information it needs, with the right characteristics and the appropriate level of accessibility.

The information management function is responsible for management, control and governance of the information architecture. It includes functions such as definition, creation, registration, discovery, usage, archiving and decommissioning. This function is also responsible for the definition and management of analytics and CEP algorithms in conjunction with analytics and event applications.

As data is no longer hard-coded by each application, but instead shared among multiple applications, the management function is a vital element of the information architecture to ensure that data is consistent, available and robust at all times.

The information architecture is based on components to facilitate decoupling and late binding, which in turn facilitates configurable run-time characteristics and reconfiguration.

Information models

Meta information

Every piece of information has an associated a set of metadata – time stamps, information architecture functionality, language translations, access rights and storage. Metadata plays a vital role in the correct handling of information services, with its own life cycle and even its own metadata.

Each information object has rules associated with it, governing its ownership and what actions can be performed on it, such as create, publish, expose, access, store, and mediate. These rules are part of the system's meta-information and are verified and approved by the information management.

Application-level information model

The CIME helps to simplify application integration and ensures that application-level support exists. The CIME is used for all inter-application communication and mediation, and defines the common and shared set of information. It is fully controlled, life cycle managed and governed by the enterprise. The information architecture can take the role as the CIME master or just be a supporter, if another master exists.

The CIME is an ontology, meaning that all the relationships of an information object are included in the model and are rules-based. The relationships themselves are also defined as information objects and also adhere to the rules of the information architecture. Building the CIME in this way ensures that the connection between metadata and business processes and the relationships between information types can be maintained, as well as allowing for transformations and supporting different storage formats. Devising a CIME from scratch is both time-consuming and difficult. To simplify the process, a number of blueprint versions exist that fit different types of enterprises.

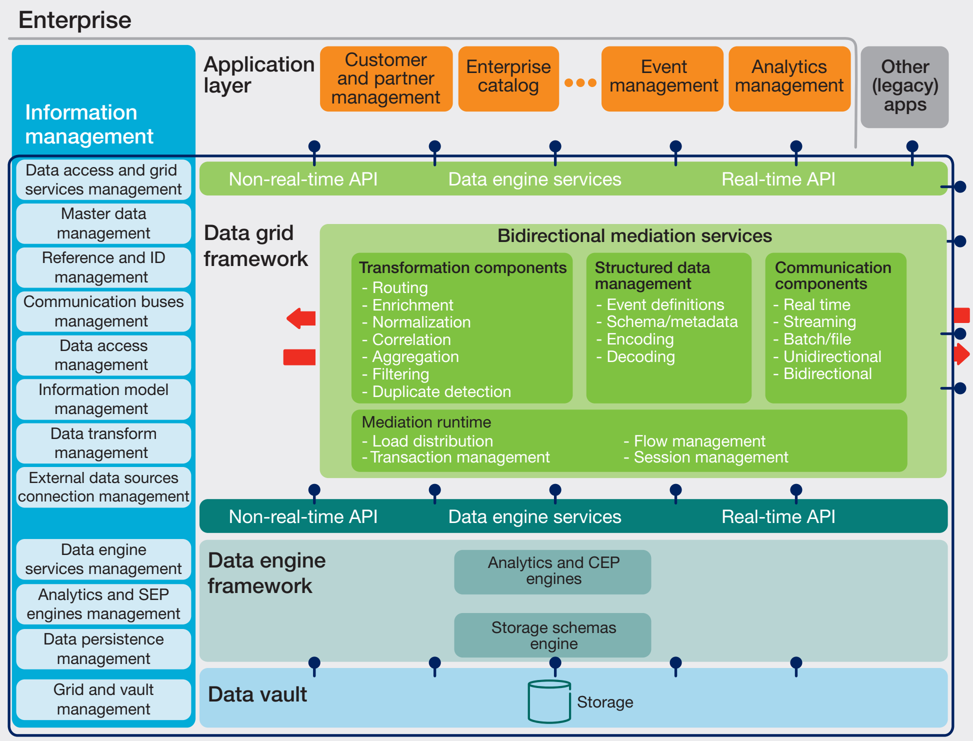
Producers and consumers

The information architecture provides services to producers and consumers of information in an enterprise. Only producers assigned with the role of information owner can create new information objects and instances of objects. This role is assigned by the information management function, which can also adopt the owner role itself.

The information architecture holds the master information objects. Applications – even information owners – that consume specific information objects must use the system for access to that object. All applications connected to the information architecture share common information through the same mechanism; all integration issues regarding external data sources are hidden.

To ensure data consistency, information must be immutable. To meet this requirement, changes are implemented by creating new instances of an information object with a unique time stamp. The history of previous instances is kept, and in this way, the most recent information is always available but past scenarios can also be recreated.

FIGURE 5 **Mediation architecture**



Storage

The storage and access formats for an information object are determined by its owner. To cater for the varying information needs of different users, several storage and access formats can be assigned to the same information object. The information architecture performs the transformation from one storage format to another and also takes care of storage management. There is always a selected storage for the master of each information object.

Varying user needs are typically related to parameters such as semantic abstraction level, latency, integrity, raw formats for real-time performance, batch/volume, and special requests.

Different formats require different storage schemas, and the different technologies are offered as services to information producers.

There are far more applications that use information than there are applications that create it. Many applications are pseudo creators; in other words, they recreate information for internal use. Applications that create information are responsible for storing it, and safeguarding its integrity and access rights. A system without master data is costly to operate and develop, quality can be difficult to maintain and TTM tends to be longer than necessary.

Mediation

Traditionally, the term mediation refers to the area of collecting and processing Call Detail Records (CDRs) from network elements for the purpose of billing. However, in the context of information architecture, mediation refers to the bidirectional transformation and integration services that are essential for for communication with external data sources.

The right-hand side of **Figure 5** shows the set of functions that support the flow of information in and out of the system. This part of the information architecture shields applications on the inside from everything that is on the outside (so that they are agnostic from an information and responsibility perspective). Typically, this is where an enterprise interfaces with legacy systems and external data sources, and communication is achieved through grid components that expose access to information objects. Typical functions include integration, protocol adaptation, handling of services and data streams, as well as transformation, and identity mapping.

Mediation components can be selected, configured and bundled together to create mediation services. The mediation components are arranged so that they model the flow of information in and out of the system. Like all other information in the system, the data carried over communication channels, also needs to be understood and transformed into the CIME.

Canonical representation supports abstraction of an information object into its constituent elements, data types and placement. These generated object definitions are then suitable for further processing through transformation components.

Integration

There is more to integration than simply creating connections to » external data sources. Beyond basic connection, this part of mediation includes consolidation and complete life cycle management of connections to the world outside the information architecture.

The communication components are self-sufficient; they can manage and encapsulate the complete life cycle of communication with an external entity – including the ability to initiate or wait for connections, apply interval-based polling, and retrieve data on-demand.

Communication components encapsulate the integration points and are responsible for ingesting data from external applications or data sources, and for external distribution of data. A variety of communication protocols can be used for this purpose, and they can be real-time or offline, streamed or batch-oriented, unidirectional or bidirectional – all of which needs to be managed by the communication components.

Transformation

Transformation functions operate on the object representation of incoming and outgoing information content. A given transformation component encapsulates programmatic logic to – in a single well-defined step – complete the transformation and deliver the information object to its intended destination and/or map it into its corresponding placeholder (according to the CIME). The primary aim is to be able to transform seemingly non-compliant data into a CIME-compliant framework or to transform data to be compliant with an external process.

Transformation is a two-way process that can be achieved in a number of ways – information mapping, procedural translation and orchestration – depending on the complexity of the transformation needed. Information mapping is not, as might be expected, exactly one-to-one, but can be used to transform information objects that closely resemble the CIME representation. Procedural translation is used when algorithmic logic is required to translate an information object into the CIME representation. Otherwise, orchestration can be used for transformations that are not easily encapsulated into a single step execution, such as:

- handling cardinality differences – in situations where there is a many-to-one or one-to-many relationship between external information objects and objects that are internal to the information architecture;
- achieving temporal alignment or asynchronous communication – so that out-of-band data can be correctly sequenced on the basis of time stamps (or some other criteria);
- correlating information received from multiple information sources to construct single information objects; and
- aggregating information that is split across multiple external information items into a combined information object, using thresholds based on a time window or other criteria, such as summation values.

Event processing and analytics

Analytics routines and CEP are placed in the data engine layer as they perform information services requested by a user.

Typically, services offered include traditional queries, real-time analysis, big data analytics, data streaming, pattern identification and event detection.

Implications for traditional applications

The current approach of information management, and hiding information inside applications, needs to be modernized. Information control and associated business logic need to be externalized from the application and put under the control of the enterprise.

Many data queries and analysis functions need information originating from multiple applications and information sources. To ensure analysis is comparable, it should be performed by an separate analysis service and not internally by applications.

Applications may cache information and are responsible for determining the structure of this information. Cached information is always a replica of the master, and applications are responsible for assuring the relevance of the cached information – providing updates when necessary.

For (real-time) performance reasons, applications may use caching techniques to store data locally, but the data vault is the only persistent storage.

Conclusions

To make the shift from traditional applications acting as isolated islands to a fully life cycle managed information architecture takes time. However, the shift to an information-centric architecture is an incremental one, and so both approaches will exist in parallel. Mediation plays a pivotal role in setting up an information-centric architecture, but once the majority of applications conform to the architecture, its significance will slowly decrease.

An information-architecture model provides enterprises with centralized and shared set of information services and functions, making the vital connection between information and business processes – the key to flexibility.

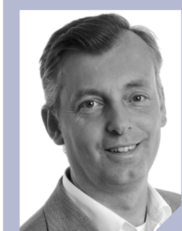
An information architecture is a single, shared and stable information environment that is trustworthy and can be used by all the applications of an enterprise. It separates functionality from information, hides integration aspects, and is responsible for storage and persistence.

The higher level of abstraction that an information architecture provides generates many benefits. For example, as all applications within the system use the same semantics and formats, integration costs are lower. Connections with external systems are simpler, as there is only one internal representation to translate to and from. With only one model and one master, quality measures are easier to enforce, and security is easier to assure. Functionality can be reused on different data sets, which reduces the number of applications needed to conduct business. And applications become smaller as common information services are provided centrally. Ultimately, a single information architecture offers flexibility, allowing an enterprise to make the most of business opportunities that arise.

The primary challenge for the enterprise is to define and manage the CIME that defines its business.

In developing the information architecture concept described in this article, Ericsson has focused on the telco industry. Information services tend to be generic, and so the concept is, in theory, applicable to any industry.

Ericsson
Review



To bring you the best of Ericsson's research world, our employees have been writing articles for Ericsson Review – our communications technology journal – since 1924. Today, Ericsson Review articles have a two-to-five year perspective and our objective is to provide you with up-to-date insights on how things are shaping up for the Networked Society.

Address:
Ericsson
SE-164 83 Stockholm, Sweden
Phone: +46 8 719 0000

Publishing:
Ericsson Review articles and additional material are pub ished on: www ericsson.com/review. Use the RSS feed to stay informed of the latest updates.

Ericsson Technology Insights
All Ericsson Review articles are available on the Ericsson Technology Insights app available for Android and iOS devices. The ink for your device is on the Ericsson Review website:www.ericsson.com/review. If you are viewing this digitally, you can:
download from Google Play or
download from the App Store

Publisher: U f Ewaldsson

Editorial board:
Håkan Andersson, Hans Antvik, Ulrika Bergström, Joakim Cerwall, Deirdre P. Doyle, Dan Fahrman, Anita Frisell, Jonas Högberg, Patrik Jestin,Magnus Karlsson, Cenk Kirbas, Sara Kullman, Börje Lundwall, Hans Mickelsson, U f Olsson, Patrik Regårdh, Patrik Roséen and Gunnar Thrysin

Editor:
Deirdre P. Doyle
deirdre.doyle@jgcommunication se

Subeditors:
Paul Eade, Nathan Hegedus and Ian Nicholson

Art director and layout:
Carola Pilarz

Illustrations:
Claes-Göran Andersson

ISSN: 0014-0171

Volume: 91, 2014

Lars Angelin

is an expert in the technology area multimedia management at BUSS. Lars has more than 28 years of work experience in the areas of concept development, architecture and strategies within the telco and education industries. Lars joined Ericsson in 1996 as a research engineer, and in 2003 he moved to a position as concept developer for telco-near applications, initiating and driving activities, most of them related to M2M or the OSS/BSS area. He holds an M.Sc. in engineering physics and a Tech. Licentiate in tele-traffic theory from Lund Institute.

Jatin Suri

is a senior specialist in mediation solutions and a member of the BUSS architecture team. He has been working at Ericsson since 2005 in the area of BSS/OSS and is currently the deputy chief architect for revenue management (invoicing domain). He holds a B.Eng. in computer science from the University of Pune, India.

Munish Agarwal

is a senior specialist in multimedia architecture and chief implementation architect for DUSS. He has been working at Ericsson since 2004, primarily in the OSS/BSS area. He is currently the driver for the OSS/BSS implementation architecture and is the product owner for Next Generation Execution Environment. He holds a B.Tech.in material science from the Indian Institute of Technology, Kharagpur, India.

Akshay Mehra

is the unit manager for M-Commerce R&D. Prior to holding this position he was the system manager for multi-mediation, supporting the sales team with mediation-based solutions. He has been with Ericsson since 2008, working in the OSS/BSS area. He holds a B.Tech. and an M.Tech. in microelectronics from the Indian Institute of Technology, Bombay, India.

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

ISSN 0014-0171
284 23-3224 | Uen
© Ericsson AB 2014