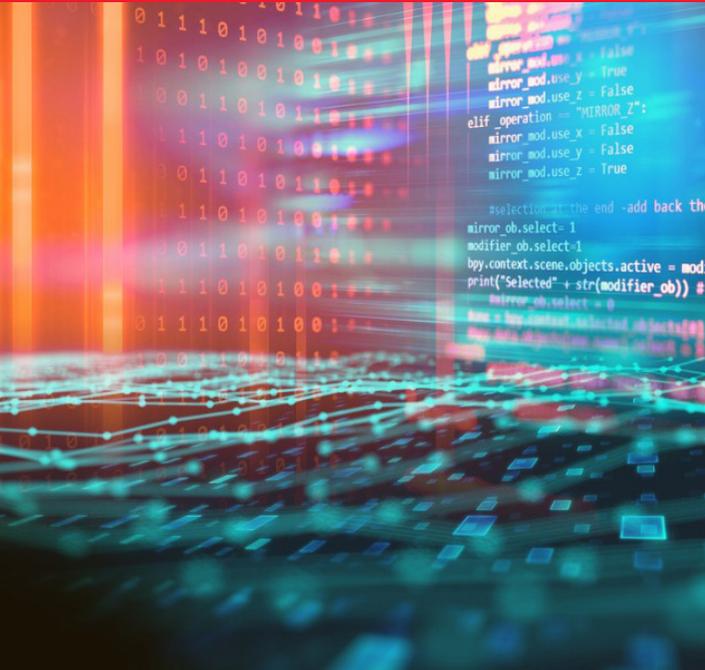


DEVELOPING A DATA FABRIC

Author: Rick F. van der Lans



INTENDA

You can buy an ETL tool, a reporting tool, and a database server, but you can't typically buy a data fabric. When your company needs a data fabric, you must design and develop it yourself. It's just like you can't buy a data warehouse environment or microservices architecture. These need to be designed and implemented and involves the use of many different tools. The same applies for data fabrics.

But what is a data fabric? Conceptually, it's a layer of software that allows any type of data consumer to access data available in one of the many IT systems. In other words, it's about data abstraction, it's about making all the enterprise (and possibly external) data available to all the data consumers, including simple reports, advanced dashboards, apps running on mobile devices, data science tools, real-time applications, and transaction processing applications.

Gartner defines a data fabric as follows: "A data fabric enables frictionless access and sharing of data in a distributed data environment. It enables a single and consistent data management framework, which allows seamless data access and processing by design across otherwise siloed storage."

The key terms here are frictionless data access, sharing of data and a single, and

consistent data management framework.

Frictionless data access means that all the data can be accessed without difficulties, regardless of where and how it is stored. Whether it's stored in a data mart, a transaction database, deeply hidden in a packaged application, or in a simple flat file, all this data should be accessible for those data consumers who need it. Frictionless data access matches the need to democratize data, making the data you own easily available to the entire organization.

A data fabric enables frictionless access and sharing of data in a distributed data environment...

Sharing of data refers to data being made available to many data consumers and for a wide variety of data use. Sharing refers to data consumers sharing the same data and the same metadata describing that data. This differs from many existing data architectures shared by a limited set of users, such as a data lake used only by data scientists and a data warehouse only by dashboard users.

A single and consistent data management framework indicates that the data fabric manages all the data and metadata and delivers a consistent view of the data to all the data consumers.

Technically, a data fabric offers some service interface layer that can be used to retrieve, analyze, insert, update, and delete data. This layer hides the different technologies used by the systems that contain the data, it hides the language or API that is used and the location of the data. If data from different source systems must be integrated, there will be a service that shows that data in an integrated style. The layer is also responsible for data security and data privacy aspects and provides the data consumers with descriptive metadata. As with data, metadata should also be shared by all the data consumers.

A data fabric can be developed in many different ways. For example, companies can use a low-level programming language and develop a large set of services with JSON/REST interfaces that accesses all the data. These services may communicate with applications and with each other through some messaging technology. This is a feasible approach, but leads to a massive software development exercise, because all the aspects need to be covered, including metadata access, data security,

data integration, data cleansing, and so on.

Another approach is by copying all the relevant data from all the source systems to one big data store, a so-called data hub or data lake. A service interface offers access to this data store. Building this service interface on one central data store is easier than the previous approach, but it will still be a gigantic development effort. A drawback of such an approach is that the services can't deliver real-time data.

The third approach for developing a data fabric is by utilizing a platform, such as fraXses. The main advantage of this approach is that the fraXses platform supports the required features needed to develop a data fabric for your organisation. It is built to deliver a data abstraction layer on top of a heterogeneous set of source systems. It can integrate data from source systems without having to store data redundantly, metadata is automatically kept and is made accessible to data consumers, it supports multiple interfaces and languages including SQL and JSON/REST, centralized data security and protection features, and it can utilize the full power of underlying database servers by using query pushdown.

As indicated, you can't typically buy a data fabric, you need to design and develop one that fits within your organisation. If you do, make sure to select an approach that provides all the features you require, and also offers you high productivity and easy maintenance, as offered by fraXses, as that will ultimately determine whether the data fabric dream becomes reality.

The fraXses platform supports all the required features needed to develop a data fabric within your organisation.