Fybrik

Unlock value from organizational data without sacrificing security and usability

August 01, 2021 | Open Source Software

Modernizing the usage of data is challenging for enterprises

Data is the foundation for business value. All enterprises strive to be as data driven as possible, and to unlock as much value and insight as they can from their data. However, accessing and taking advantage of data is not simple. Enterprise data is stored on a multitude of systems and in a multitude of locations. The use of this data is governed by regulatory requirements and enterprise policies. In addition, accessing data requires a user to tackle a labyrinth of protocols, APIs, location constraints, performance concerns, security requirements, audit mechanisms, lineage updates, and more.

While in reality there are many different personas involved in turning data into value, we'll focus on the challenges faced by three key personas:

The **data user** (e.g., data scientist or analyst) needs approvals to get access to the data, and requires the data to be prepared and made compliant with the enterprise policies. One example of a data user is an application developer who has to be sure their application addresses a wide array of data considerations, such as location-independent data access, performance, obtaining and managing credentials, application security, and compliance with data governance and usage policies.

The **operator** needs to manage the infrastructure resources and make data available to the data users. It is challenging to efficiently manage the resources across hybrid cloud, providing the required performance, managing credentials, enabling access, etc.

The **data governance officer / data steward** sets the policies and guidelines for the data, but is also charged with the enforcement of the data. Tracking the flow of data and copies while understanding the data usage in a dynamic and global regulatory environment is a complex and challenging task.

All of these challenges – which are orthogonal to the actual logic that the business needs to execute in order to get insights from the data – hinder turning data into value. Because of this, organizations suffer from inefficient use of data. The complexity of development, the time it takes to be able to use data, the risk of reputational impact if data is lost, and the risk of fines for regulatory non-compliance all lead to lost opportunities.

The goal is to empower enterprise users to create value

Fybrik is an open-source project whose primary objective is to enable each of these personas to declaratively assert their requirements as high-level policies, freeing up the business to focus on deriving value from data. Based upon these declarations, Fybrik injects system functions into the data path to control the data flows into and out of the workload, controlling the external interactions of the workload. This injected code handles all of the non-business logic, allowing the business to concentrate on creating insights from the data.

Using these control functions, Fybrik can:

1. Abstract the connection to the data, thereby allowing seamless access to data, agnostic to the locations of computation and storage. The injected components transparently connect the application to the data without being aware of the data's actual location or the data source API / protocol. Furthermore, by virtue of being in the data path, Fybrik can schedule compute near the data or initiate data mobility as needed by the computation across multiple clouds.
2. Secure data usage and enforce governance by intermediating all input and output flows, encapsulating and isolating the workload, handling data credentials in the system code without giving the user actual credentials, and delivering data only after enforcing policies which take into account the purpose of the computation, the classification of the data and the running location.
3. Control and optimize the workload by orchestrating computation and data mobility, including creating and using system-managed copies based on requirements and policies, including optimizing performance between compute and data, optimizing communication between computation steps, leveraging caching and more.
4. Provides instrumentation to observe the use of data by enabling the seamless platform-level collection of metrics, logs, audit and data provenance information on all flows for performance, governance and compliance management.

Taken together, these greatly free up time, reduce complexity and process for data users, operators and governance officers. Moreover, it can reduce the risk of data loss, ensure compliance with regulations and accelerate time to value.

Fybrik allows enterprises to service enable their data

Fybrik builds on top of Kubernetes and service mesh concepts and leverages a declarative framework that abstracts away non-functional data functions (e.g., performance, security, governance) and captures the needs of the data user, the availability of data, the restrictions and expectations set by the data steward for using data, and the available compute and storage resources.

Fybrik integrates with native tools for each persona and enables each persona to focus on his or her business goals assisting with automating processes and shifting non-functional features into the platform.

The **data user** develops and runs models, analytics or applications using native tools such as notebooks or container images, deploys the workload in containers, and specifies the needed data sources and business and operational expectations.

The **operator** describes the infrastructure (e.g., cloud resources, data locations, networking) using Kuberentes abstractions and mechanisms.

The **data governance officer / data steward** provides the available data, defines policies and controls through data catalogs and policy management tools, and checks for compliance through logging, audit and lineage mechanism. Fybrik can connect to any 3rd party catalog and policy manager through a pluggable connectors framework

Fybrik consists of a control plane that orchestrates compute and data pipelines, a run time that encapsulates containerized workloads and intermediates the data flows, and an extensible set of pluggable modules in a module repository. The control plane takes the declarative information and builds a deployment plan, called a blueprint, that describes how the workload interacts with Fybrik as illustrated in Figure 1.

Fybrik collects the following information by leveraging a pluggable connector framework to build the blueprint:

1. The expectation of the data user/application and the resources it needs.
2. Information about data sources/assets from a data catalog.
3. Policies that govern the workload and control the use of data from a policy manager.
4. The available infrastructure resources that can be used.

Next Fybrik matches the above with the functions available from the module repository and computes an optimized blueprint which describes the data pipelines for the workload.

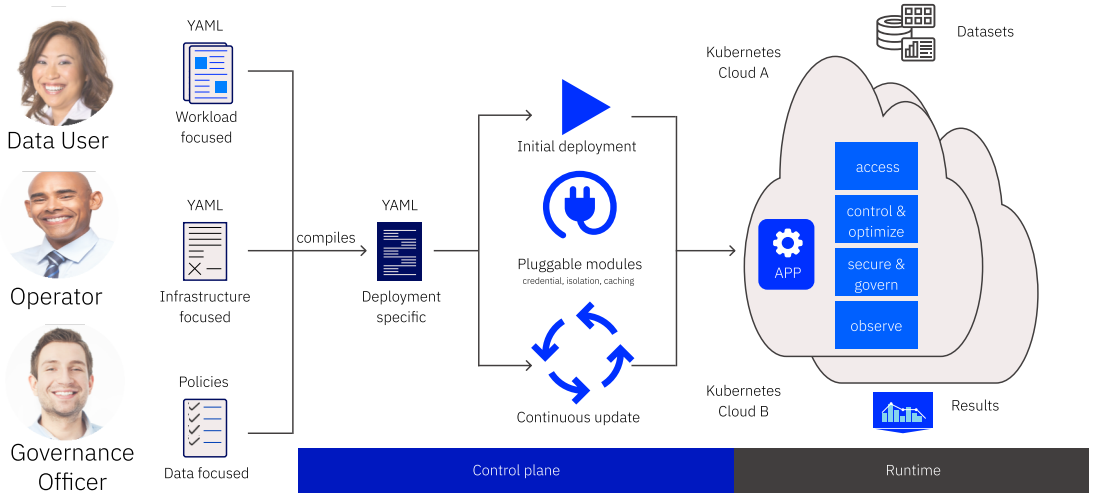

Figure 1: Fybrik control plane - building a blueprint

Modules provide system functions that interact with the workload by intermediating the data pipeline. Modules are triggered to run either (i) before running the workload (e.g., create copies, deploy caches), (ii) after the workload runs (e.g., registering newly created assets, cleanup tasks), or (iii) during the execution of the workload as data pipeline components (e.g., read, write, audit), either in-line or asynchronously.

The modules are deployed from an open repository and new modules can be developed and added. Modules can either be open or proprietary. With this extensible set of modules, Fybrik enables a wide range of non-functional data functions. Examples of functions that modules can perform are injecting credentials automatically on behalf of the user, applying policies in-line to the data pipeline, deploying caching, and abstracting the location of data, evaluating data quality or performing data audit while reading, writing or transferring data.

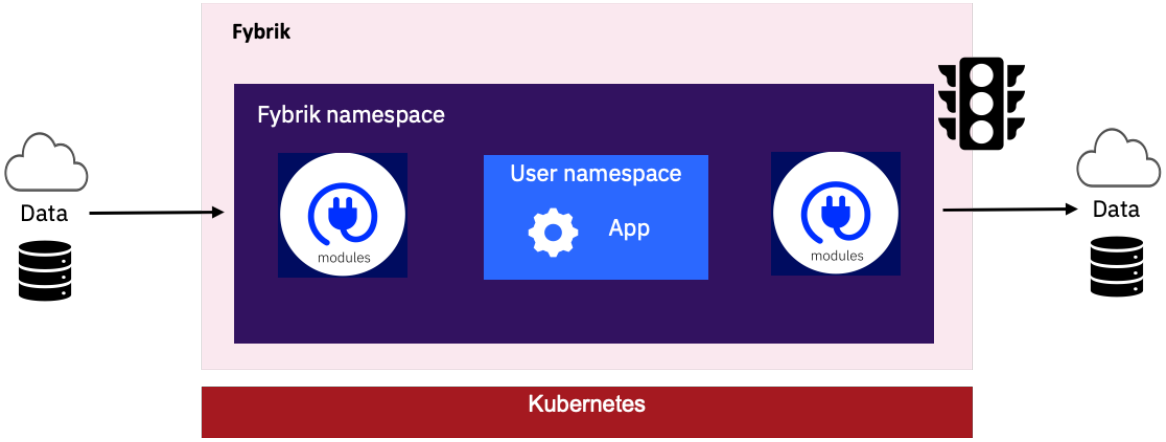

Figure 2: Fybrik runtime encapsulation of a user workload

When the blueprint is deployed, the workload is encapsulated by the Fybrik and the modules are injected into the workload data path, thus intermediating any external interactions. Figure 2 describe a user workload running in a Kubernetes namespace (a mechanism for logically partitioning a cluster to mulitple execution environments), and all external access, either read

or write is redirected through Fybrik modules. The modules run in their own namespace, outside of the user control, and can inspect data coming into or leaving the workload and take actions. For example, filtering incoming data to match policies or audit outgoing data. Modules can be developed using methods such as sidecars, proxies, Kubernetes resources and configurations.

Fybrik is designed for a multi-cloud environment. The control plane can build cross cloud data pipelines and deploy the required modules on multiple clouds to form the needed data pipeline.

More information about Fybrik

More information can be found at the project Web site
https://fybrik.io
and the Git repo at
https://github.com/fybrik

Contacts for the Fybrik

This whitepaper was written by

- Michael Factor factor@il.ibm.com
- Ronen Kat ronenkat@il.ibm.com

This whitepaper was updated on August 1, 2021 to reflect Fybrik as project name