Xaana.Ai

TURIUM

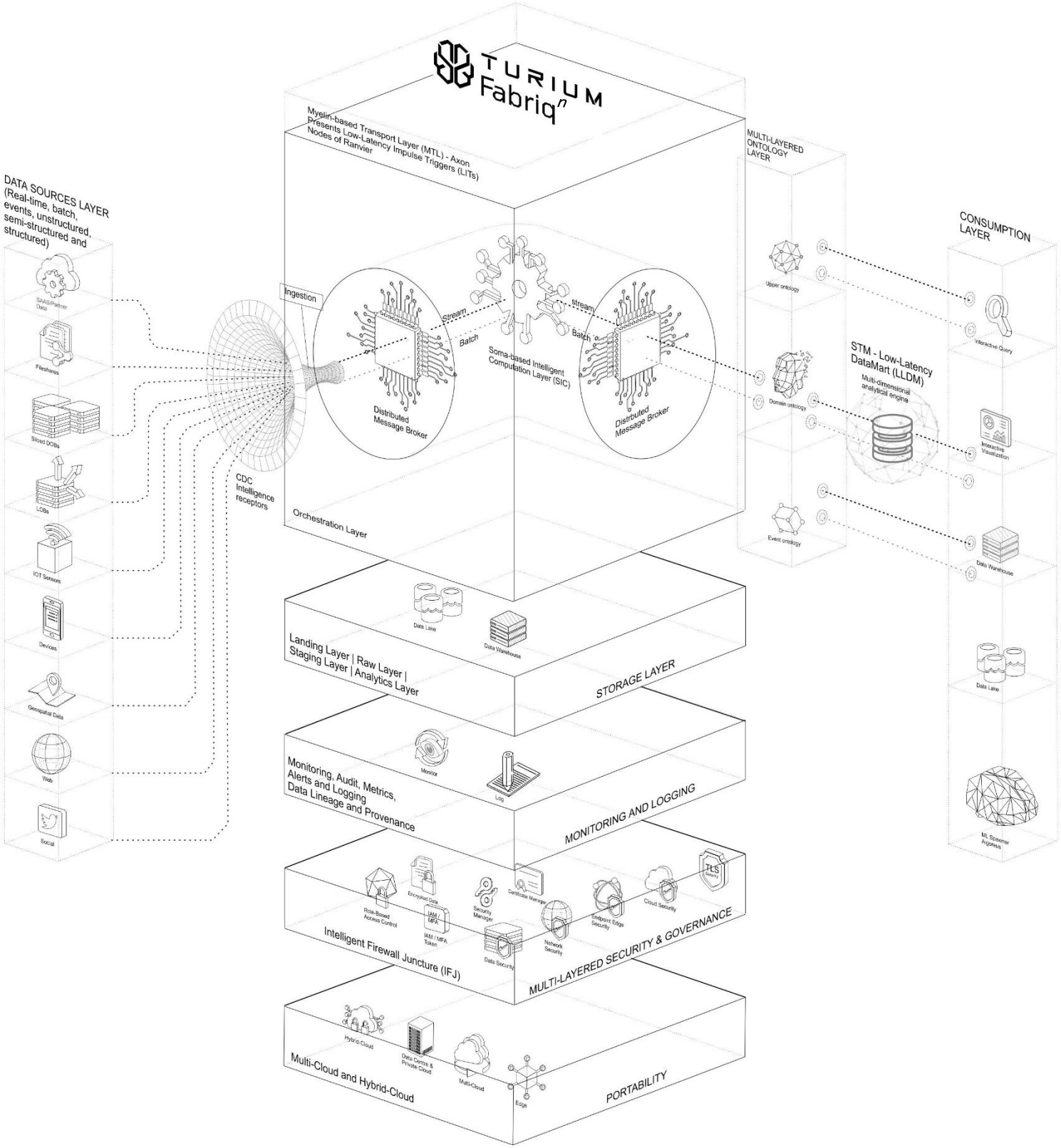
Fabriq^{n} + Algoreus

Technical Specifications

January 2023

Contents

Deploy **FABRIQ**^{n}, a data supremacy platform, for cognitive integration



I. Introduction of FABRIQ^n

Data supremacy is a comprehensively transformative process that produces the most qualitative, well-curated, highly contextual data to enable data literacy and real-time decision precision across the enterprise. FABRIQ^n specialises in systems integration and interoperability solutions, with a focus on knitting disparate databases, IOT systems, apps, and platforms together, so each component works together harmoniously, producing insights in real-time and delivering a capability greater than just the sum of its elements. With one cognitive platform, users can discover previously unseen links across their entire universe of data.

It is designed to reduce the cost of data integration over time through a rich suite of capabilities that act as a force multiplier for enterprise teams. FABRIQ^n is developed to serve as the data integration intelligence for the most complex environments in the world. It consists of hundreds of distinct services that cover a wide range of functionality. Together, these services combine to form a modular, end-to-end operations platform with low configuration.

FABRIQ^n's novel attempt at the multi-layered Ontology assimilates data and models into a comprehensive semantic model of the business. Capabilities include structured mechanisms for capturing data from end users back into the semantic foundation; out-of-the-box applications for exploring the ontology in structured, unstructured, geospatial, temporal, simulated, and other paradigms; and rich APIs for leveraging the Ontology as an "operational picture" throughout all parts of the enterprise.

FABRIQ^n application frameworks evolve analytics into operational workflows that enable user action, alerting, and other frontline functions. Capabilities include low-code and no-code application building, which automates the management of underlying storage, compute, data, and security enforcement; an application development framework with live preview; and APIs, webhooks, and other interfaces that allow for full-spectrum integration within the enterprise.

FABRIQ^n provides analytical capabilities for every type of user, both technical and non-technical. Capabilities include both point-and-click and code-based tools that enable table-based analysis, top-down visual analysis, geospatial analysis, time series analysis, scenario simulation, and more.

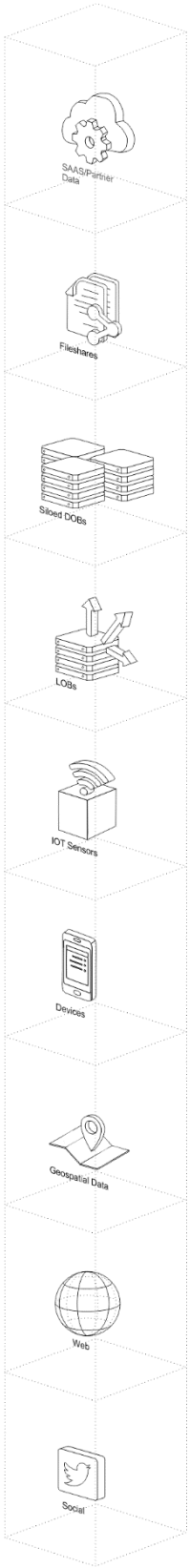
Analytics in FABRIQ^n goes beyond conventional "read-only" paradigms to write data back into the Ontology, producing valuable new insights within unified security, lineage, and governance models.

It features the best-in-class security model that propagates across the entire platform and remains with data wherever it travels by default. Capabilities include encryption of all data, both in transit and at rest; authentication and identity protection controls; authorisation controls; robust security audit logging; and highly extensible information governance, management, and privacy controls.

The first step to getting value from FABRIQ^n is to connect it to your Organisation's sources of data. FABRIQ^n's tools for connecting to data support the full range of standard enterprise data sources, ranging from cloud-based object stores, file systems, data lakes, and data warehouses.

II. FABRIQ^n ARCHITECTURE

A. *Data Sources*



FABRIQ^n enables you to ingest and analyse data from a variety of sources, including all the popular ones. Many of these sources, such as line of business (LOB) applications and siloed databases generate highly structured batches of data at fixed intervals. In addition to internal structured sources, you can receive data from modern sources that are third-party applications such as through the web, devices, sensors, video streams, and social media. These modern sources typically generate semi-structured and unstructured data, often as continuous streams.

1. Operational database sources

Typically, organisations store their operational data in various relational and NoSQL databases. FABRIQ^n connects to a variety of operational RDBMS and NoSQL databases and ingests their data into the storage layer landing zone. Our Ingestion layer performs a one-time import of the source data, replicates ongoing changes in the CDC, and encrypts incoming data using keys. It provides a scalable feature which is a predefined template that generates a data ingestion workflow based on input parameters such as source database, target location, target dataset format, target dataset partitioning columns, and schedule. A template-generated workflow implements an optimised and parallelised data ingestion pipeline consisting of crawlers, multiple parallel jobs, and triggers connecting them based on conditions.

2. Streaming data sources

The ingestion receives streaming data from internal and external sources. With a few clicks, you can configure an API endpoint where sources can send streaming data such as clickstreams, application and infrastructure logs and monitoring metrics, and IoT data such as devices telemetry and sensor readings. It does the following:

- Buffers incoming streams
- Batches, compresses, transforms, and encrypts the streams
- Stores the streams as distributed file system objects in the landing zone in the data lake

It natively integrates with the security and storage layers and can deliver data to the data landing zone. It automatically scales to adjust to the volume and throughput of incoming data.

3. File sources

Many applications store structured and unstructured data in files that are hosted on Network Attached Storage (NAS) arrays. Organisations also receive data files from partners and third-party vendors.

3.1 Internal file shares

FABRIQ^n can ingest hundreds of terabytes, and millions of files from NFS and SMB enabled NAS devices into the storage layer landing zone. It automatically handles scripting of copy jobs, scheduling, and monitoring transfers, validating data integrity, and optimising network utilisation. Also, it can perform one-time file transfers and monitor and sync changed files through CDC into the data storage layer.

3.2 Partner data files

FTP is the most common method for exchanging data files with partners. FABRIQ^n is a highly available and scalable service that supports secure FTP endpoints and natively integrates with the data storage layer. It supports encryption using keys and common authentication methods, including IAM.

4 Data APIs

Organisations today use SaaS and partner applications to support their business operations. Analysing SaaS and partner data in combination with internal operational application data is critical to gaining 360-degree business insights.

4.1 SaaS APIs

With a few clicks, your flows can connect to SaaS applications (such as SAP, Salesforce, and ServiceNow), ingest data, and store it in the data lake in the storage layer. With FABRIQ^n, you can schedule data ingestion flows or trigger them by events in the SaaS application. Ingested data can be validated, filtered, mapped, and masked before storing in the data lake. It natively integrates with authentication, authorisation, and encryption services in the security and governance layer.

4.2 Partner APIs

To ingest data from partner and third-party APIs, organisations build or purchase custom applications that connect to APIs, and fetch data to the landing zone. FABRIQ^n provides out-of-the-box capabilities to schedule singular API call or include them as part of a more complex data ingestion workflow built.

5 Third-party data sources

Your organisation can gain a business edge by combining your internal data with third-party datasets. FABRIQ^n provides a way to find, subscribe to, and ingest third-party data directly into the landing zone. You can ingest a full third-party dataset and then automate detecting and ingesting revisions to that dataset.

B. Ingestion

The overall goal of data ingestion in FABRIQ^n is to provide a digital view of the objective reality within your Organisation. Achieving this goal typically requires syncing data from many source systems, imposing a common schema, combining datasets together, and enabling teams to build use cases off a common data foundation.

The ingestion layer in FABRIQ^n's architecture enables data ingestion from a variety of sources into the storage layer landing zone. It can ingest and deliver batch as well as real-time streaming data into a data lake as well as data warehouse components of the storage layer. It provides the ability to connect to internal and external data sources over a variety of protocols by matching the unique connectivity, data format, data structure, and data velocity requirements of operational database sources, streaming data sources, and file sources.

Data extraction in FABRIQ^n is done in several ways:

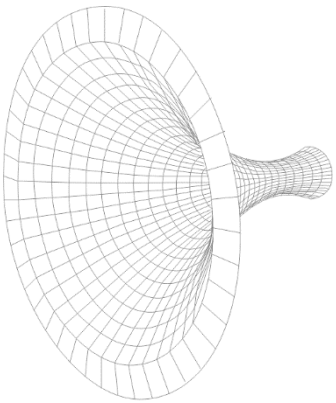
- Full extraction: All data from the table, object, or application is extracted at once. This is the simplest method to use as one does not need to know which data has been altered. Full extraction is ideal for small data sources, but not recommended for larger batches.

- Incremental extraction: Only the changed records are extracted. For this method, the source system connects with CDC to capture which records have been modified, so only those that have been changed are extracted.

- Change notification: The source system sends a notification that contains which changes have been made or will direct you to the altered records in the database. Change notification is a continuous process and therefore requires immediate capture and processing of the messages, unless you buffer them so that they can be processed in batches. This is made possible by the Change Data Capture (CDC) in FABRIQ^n.

Another task of FABRIQ^n's Ingestion Layer is to filter and check the data quality. This confirms that all data is extracted and adding logic-based business rules with Bayesian Inferencing defines how the data should look like. If the data quality is subpar, it can trigger one or multiple events: the data can be rejected, an alert can be raised for manual intervention, or automatic data quality improvement processes can be executed.

C. *Intelligent Receptors*



FABRIQ^n uses Intelligent Receptors such as change data capture (often called CDC), plus snapshots which enable it to capture everything already in the database (master data), following with new changes made to the data.

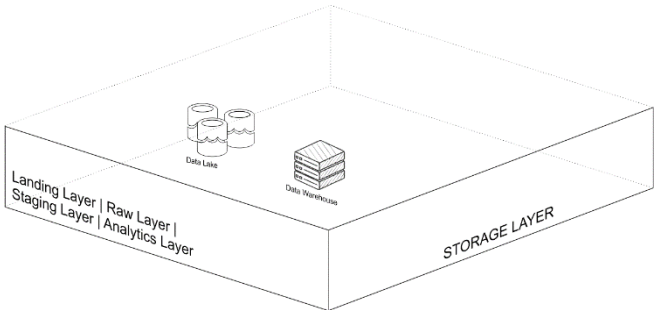
Query-based CDC uses a database query to pull new data from the database. The query will include a predicate to identify what has changed. This will be based on a timestamp field or an incrementing identifier column (or both).

Log-based CDC uses the database's transaction log to extract details of every change made. The particular transaction log implementation and specifics will vary by database, but all are based on similar business logic rules. Every change made in the database is written to its transaction log. The changes written to the transaction log include inserts, updates, and even deletes. So, when two rows are written to the database, two entries are added to the transaction log. Those two entries from the transaction log are decoded, and the actual data from the database row is written to two new events in the distributed message broker. One of the several benefits of log-based CDC is that it can capture not just what the table rows look like now, but also what they looked like before they were changed.

Change data capture in FABRIQ^n refers to the process of identifying and capturing changes as they are made in a database or data sources, then delivering those changes in real-time to a downstream process, system, or target database. Targets include data lakes and data warehouses in FABRIQ^n's storage layer. By detecting changed records in data sources in real time and propagating those changes holistically into the entire landscape, CDC can sharply reduce the need for bulk-load updating of the warehouse. Since it is not only replicating the data, but it also identifies and sends only the most relevant data, putting less of a burden on the system and dramatically speeding up data processing for mission-critical use cases.

FABRIQ^n's CDC ensures that you always have an accurate backup in case of a catastrophe, hardware failure, or a system breach. And having a local copy of key datasets can cut down on latency and lag when global teams are working from the same source data in. Essentially, enabling the eradication of data silos.

D. *Storage Layer*



Data is organised and flows from first coming to the landing zone, then raw, staged, and finally curated for using in the consumption layer. All throughout, the storage layer encrypts data using keys, and IAM policies control granular zone-level and dataset-level access to various users and roles. Data of any structure (including unstructured data) and any format can be stored without needing to predefine any schema. This enables services in the ingestion layer to quickly land a variety of source data into the data lake in its original source format. After the data is ingested into the data lake, components in the processing layer can define schema on top of the datasets and register them in the catalog. Services in the processing and consumption layers can then use schema-on-read to apply the required structure to data read. Datasets stored are often partitioned to enable efficient filtering by the CDC. Data warehouse and lake provide a unified, natively integrated view of the incoming data.

1. Structured data storage in the data warehouse

The data warehouse stores conformed, highly trusted data, structured into the data vault, or highly denormalised schemas. Data stored in a warehouse is typically sourced from highly structured internal and external sources such as transactional systems, relational databases, and other structured operational sources, typically on a regular cadence. FABRIQ^n's warehouses can typically store petabytes scale data in built-in high-performance

storage volumes in a compressed, columnar format. Through MPP engines and fast attached storage, a modern cloud-native data warehouse provides low latency turnaround of complex SQL queries. All changes to data and schemas are tightly governed and validated to provide a highly trusted single source of truth datasets across business domains.

2. Structured and unstructured data storage in a data lake

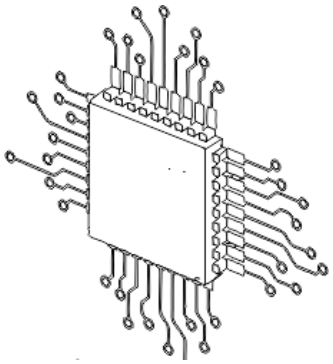
A data lake is the centralised data repository that stores all of an organisation's data. It supports the storage of data in structured, semi-structured, and unstructured formats. It can automatically scale to store petabytes of data. Typically, data is ingested and stored as is in the data lake (without having to first define schema) to accelerate real-time ingestion and reduce the time needed for preparation before data can be explored. Native integration between a data lake and data warehouse also reduces storage costs by allowing you to offload a large quantity of colder historical data from warehouse storage.

3. Catalog

A central Data Catalog manages metadata for all the datasets in the storage layer and is crucial to enabling self-service discovery of data in the interactive query. Additionally, separating metadata from data into a central schema enables schema-on-read for the processing and consumption layer components. It manages both technical metadata (such as versioned table schemas, partitioning information, physical data location, and update timestamps) and business attributes (such as data owner, data steward, column business definition, and column information sensitivity) of all their datasets.

Additionally, it provides APIs to enable metadata registration and management using custom scripts and third-party products. In the monitoring layer, it can track evolving schemas and newly added partitions of datasets in the data lake. It provides the data lake administrator a central place to set up granular table- and column-level permissions for databases and tables hosted in the data lake. Permissions are set up; users and groups can access only authorised tables and columns. In FABRIQ^n, the catalog is shared by both the data lake and data warehouse, and enables writing queries that incorporate data stored in the data lake as well as the data warehouse in the same SQL. As the number of datasets grows, this layer makes datasets discoverable by providing search capabilities in the Interactive Query Consumption Layer.

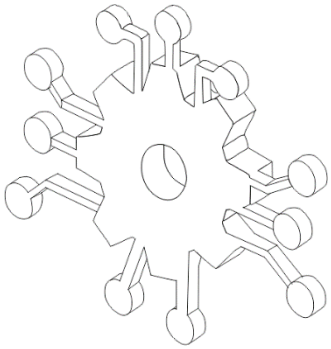
E. Distributed Message Brokers



Distributed Message Brokers are data brokers that exist on both sides of the system that reliably receive raw and deliver computed data between many disparate systems or applications. It comprises two types of components: Components used to create multi-step data processing brokers and components to orchestrate data processing brokers on schedule or in response to event triggers (such as ingestion of new data into the landing zone). It provides components to build, orchestrate, and run a broker that can easily fit to large data volumes. The distributed message brokers accelerate not only the loading process, but also support data transformations.

Additionally, it also provides triggers and workflow capabilities that you can use to build multi-step end-to-end data brokers that include job dependencies by running parallel steps. You can schedule the workflows or run them on demand. It natively integrates to the storage, and security layers. FABRIQ^n provides a visual representation of complex workflows and their running state to make them easy to understand. It manages state, checkpoints, and restarts the workflow for you to make sure that the steps in your data run in order and as expected. Built-in try/catch, retry, and rollback capabilities deal with errors and exceptions automatically.

F. Soma-based Intelligent Computation (SIC) Layer



The computation layer contains multi-step workflows that can catalog, validate, clean, transform, and enrich datasets to advance them from raw to curated zones in the storage layer. The layer provides the quickest time to market by providing purpose-built components that match the right dataset characteristics (size, format, schema, speed), and the processing task at hand. It provides components to support schema-on-write, schema-on-read, partitioned datasets, and diverse data formats. Each component can read and write data to both the data lake and warehouse (collectively, the storage layer).

Components in the computation layer of FABRIQ^n are responsible for transforming data into a consumable state through data validation, clean-up, normalisation, transformation, and enrichment. It performs a variety of transformations, including data warehouse-style SQL, big data processing, batch, and near-real-time. The computation layer can access the storage interfaces and common catalog, thereby accessing all the data and metadata, which avoids data redundancies, unnecessary data movement, and duplication of code that may result when dealing with a data lake and data warehouse separately.

1. Big data processing

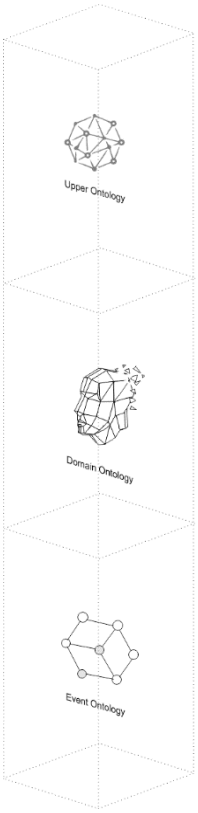
For integrated processing of large volumes of semi-structured, unstructured, or highly structured data hosted on the storage layer, FABRIQ^n can build big data processing jobs. These jobs can use the native as well as open-source connectors to access and combine relational data stored in a data warehouse with complex flat or hierarchical structured data stored in a data lake. These same jobs can store processed datasets back into the data lake, data warehouse, or both in the storage layer. It can process tens of terabytes of data, all without having to manage clusters. Additionally, it provides triggers and workflow capabilities that you can use to build multi-step end-to-end data processing brokers that include job dependencies as well as running parallel steps.

2. Near-real-time

To enable mission-critical use cases, FABRIQ^n can perform the following actions, all in near-real time: Ingest large volumes of high-frequency or streaming data, validate, clean, and enrich it by making it available for consumption. It enables you to build near-real-time data processing message brokers without having to create or manage computation infrastructure. The brokers elastically scale to match the throughput of the source, whereas the streaming jobs can be scaled in minutes by just specifying scaling parameters. Streaming message brokers typically read records from the ingestion layer of the CDC, apply transformations to them, and write processed data to the delivery stream. The delivery stream can deliver processed data to a data lake or data warehouse in the storage layer.

SIC is responsible for in-memory computing, which makes it a crucial component for attaining lightning-fast speed. Additionally, it references datasets from internal to external storage memories, which allows it to seamlessly retrieve the processed data. SIC enables a low-latency Continuous Processing Mode to Structured Streaming, allowing it to handle responses with latencies as low as 1 ms. With SIC, only one-step is needed where data is read into memory, operations performed, and the results are written back—resulting in a much faster execution. It also reuses data by using an in-memory cache to greatly speed up algorithms that repeatedly call a function on the same dataset. Data re-use is accomplished through the creation of Data Frames, an abstraction over Resilient Distributed Dataset (RDD), which is a collection of objects that is cached in memory and reused in multiple operations. This dramatically lowers the latency making SIC multiple times faster, especially for mission-critical enterprises. Also, it allows for Fog Computing in an effortless environment for IoT data sources.

G. *Multi-Layered Ontology*



The ontologies and situational awareness are interwoven in a layered structure to create awareness of events through interactions in FABRIQ^n. The Multi-Layered Ontology works in parallel with the digital assets integrated into FABRIQ^n (datasets and models) and connects them to their real-world counterparts, ranging from physical assets like plants, equipment, and products to concepts like customer orders or financial transactions. It contains both the semantic elements (events, properties, links to domain) and kinetic elements (actions, functions, dynamic security in Upper Ontology) needed to enable use cases of all types. This approach contributes to the integration of heterogeneous data and information from multi-sources and can enhance knowledge formation for decision precision using an explicit representation of the concepts and identification of relations between them.

We identified four properties from multi-sourced information produced in big data to design a novel multi-layered approach: largeness, heterogeneity, dynamism, and complexity. The largeness means a large scale of data produced by all datasets, including the sensor nodes, which is increasing exponentially. The heterogeneity refers to the differences among data models and information systems. Dynamism is the continuous streaming of changed data, schema, and relations among entities. The complexity is due to the rapid increase in the scale of the information involved, which no longer allows any one ontology to cover it.

To this end, a Multi-Layered Ontology in FABRIQ^n allows you to define a robust foundation for end-user workflows, including rich metadata for all fields with granular security and governance for all changes. It is a rich semantic multi-layer that creates a complete picture of an organisation's world by mapping datasets and models to event types, properties, link types in the domain layer, and the action types in the Upper-Ontology.

1. Event type

An event type is the schema definition of a real-world entity or event. An event refers to a single instance of an object type. An event set refers to a collection of multiple instances; that is, a group of real-world entities or events.

2. Property

A property of an event type is the schema definition of a characteristic of a real-world entity or event. A property value refers to the value of a property, or a single instance of that real-world entity or event.

3. Domain

A domain is the schema definition of a relationship between two or more event types. A domain link refers to a single instance of that relationship between two events.

4. Upper-Ontology

The upper-ontology is the schema definition of a set of changes or edits to events, property values, and the action that a user can take at once. It also includes the opportunity cost pattern recognition that occurs with action submission. Once an action is configured in the Upper Ontology, end users can make changes to events by applying actions.

5. Roles

Roles are the central permissioning model in the Multi-Layered Ontology. Similar to role-based control in the FABRIQ^n, Ontology Roles grant access to ontological resources. Roles can either be granted on the Ontology level, which is then inherited by all resources in that Ontology, or can be granted on individual layer or resource level.

6. Functions

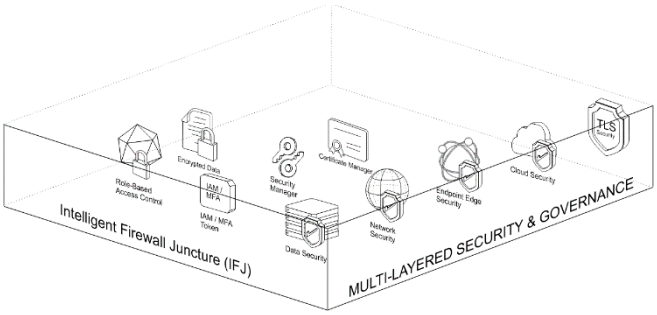
A Function is a piece of code-based logic that takes in input parameters and returns an output. Functions are natively integrated with the Multi-Layered Ontology: they can take event and event sets as input, read property values of events, and be

used across action types and applications that build on the Upper-Ontology.

7. Event Views

Event Views are a central hub for all information and workflows related to a particular event. This includes key information about an event, any linked event (domain), related metrics, actions taken in the Upper-Ontology layer, its analyses, dashboards, and applications.

H. Security and governance layer



With FABRIQ^n, security and governance are thoughtfully built into every last layer and for all components across the architecture. Designed for security-first customers who need the capability to handle financial data, Personally Identifiable Information (PII), Protected Health Information (PHI), Controlled Unclassified Information (CUI), and even classified government data in a secure and compliant manner. FABRIQ^n's strong security enables regulatory requirements across industries and continents by aligning with frameworks like HIPAA, GDPR, and ITAR. Whether you're a small business or a federal agency, you get access to every core enterprise security feature in our standard FABRIQ^n offering; the powerful internal multi-layered defence starts from data to network, cloud, and finally, protecting the endpoint edge security.

1. Embed security by default

Enforce global policies, apply best practices across the API lifecycle, and monitor for compliance. Define gateways that harden over time through feedback loops.

2. Protect sensitive data

Automatically detect and tokenise sensitive data in transit to ensure confidentiality. Get alerts when sensitive information — such as PII, PHI, and credit card data — is in API payloads. Streamline auditing and governance with prebuilt monitoring dashboards.

3. Data Encryption

By default, data is encrypted at rest using encryption keys, and in transit using TLS v1.2. It provides the capability to create and manage symmetric and asymmetric customer-managed encryption keys. In all layers of our architecture, FABRIQ^n encrypts data in the data lake and data warehouse. It supports both creating new keys and importing existing customer keys. Access to the encryption keys is controlled using IAM and is monitored through detailed audit trails.

4. Access control

RBAC-enabled support managing access at a namespace level via Identity and Access Management. Roles limit an authenticated identity's ability to access resources. When building a production application, only grant an identity the permissions it needs to interact with applicable FABRIQ^n's APIs, features, or resources. IAM provides user-, group-, and role-level identity to users and the ability to configure fine-grained access control for resources managed in all layers of our architecture. IAM supports multi-factor authentication and single sign-on through integrations with corporate directories. In data security, you can grant or revoke database-, table-, or column-level access for IAM users, groups, or roles defined in the same account hosting the metadata catalog.

5. Network protection

Our architecture uses Virtual Private Cloud (VPC) to provision a logically isolated section of the Cloud that is isolated from the internet and other customers. It provides the ability to choose your own IP address range, create subnets, and configure route tables and network gateways. It launches resources in this private VPC to protect all traffic to and from these resources. Create a private instance, which can be peered with the respective VPC network. Private instances have a private IP address, and are not exposed to the public internet.

6. Firewall rules

Control ingress and egress by setting the appropriate firewall rules on the customer VPC on which the pipeline is being executed.

7. Edge security

Construct layers of defence with rapidly configured, enterprise-grade Edge gateways. Prevent denial of service (DoS), content, and OWASP Top 10 attacks using a policy-driven architecture that can be deployed in minutes.

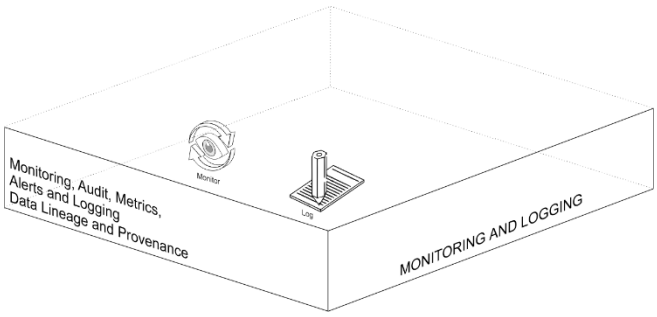
8. Automatic hardening, tokenisation, and policies

Get seamless integration between Edge and API gateways, which automatically detects API attacks, escalates them to the perimeter, and updates protections to eliminate vulnerabilities. Enhance security with a learning system that adapts as new threats emerge. Meet compliance requirements faster with a simple, format-preserving tokenisation service that protects sensitive data while supporting downstream dependencies. Enforce standardised policies across environments, audit deployed policies for compliance, and bridge the gap between security and DevOps teams by empowering API owners to detect out-of-process changes and correct violations. Xaana.Ai also runs a continuous Bug Bounty program.

9. Standardise access

Establish standard API patterns for authentication and authorisation and make patterns available as fragments to promote reuse instead of writing new, potentially insecure code.

I. Monitoring and Logging Layer

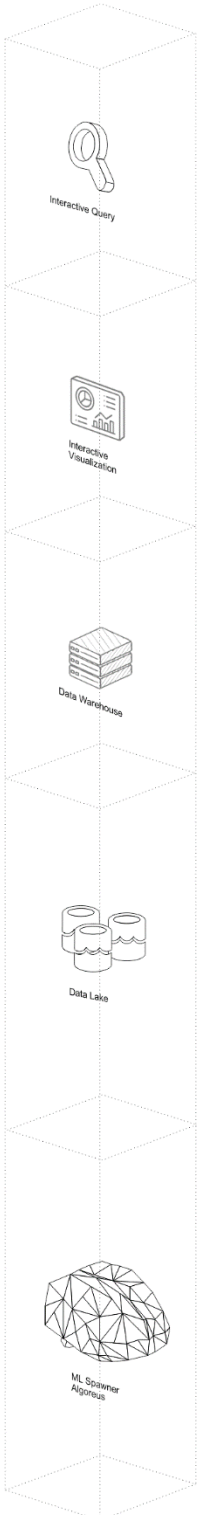


FABRIQ^n in all the layers of the architecture stores detailed logs and monitoring metrics. It provides the ability to analyse logs, visualise monitored metrics, define monitoring thresholds, and send alerts when thresholds are crossed. It also stores extensive audit trails of user and service actions. It provides event history of the account activity, which simplifies security analysis, resource change tracking, and troubleshooting. In addition, it detects unusual activity in your accounts. These capabilities help simplify operational analysis and troubleshooting.

Data Lineage is an interactive tool that facilitates a holistic view of how data flows through the FABRIQ^n platform. With Data Lineage, you can: easily find and discover datasets, and search to find datasets using Project, table, and column names. Also, you can explore workflows through a powerful interface by viewing the attributes of a group of tables at once, visualising your workflow through colouring (e.g., colour out-of-date tables). This allows you to drill into details about your data, such as its schema, when it was last built, and the code that generated the data itself. Lastly, it allows to create snapshots to share with other users.

Monitoring the Health is a FABRIQ^n service that checks and alerts on common issues across datasets. Health comes with pre-built checks for potential issues regarding dataset status, time, size, content, and schema. In the event of a failed check, it sends in-platform notifications and emails to alert you about the failure. It lets you monitor the health of an individual dataset, across a project through catalog, along with a summary of how many checks have passed or failed. Also, it enables monitoring health across a workflow and for the entire platform.

J. *Consumption Layer*



FABRIQ^n's architecture democratises consumption across different persona types by providing purpose-built services that enable a variety of use cases, such as interactive SQL queries, BI through visualisations, and ML through ALGOREUS.

Interactive Query

FABRIQ^n enables you to run complex workflows against terabytes of data stored in the storage layer without needing to load it into a database first. The queries can analyse structured, semi-structured, and columnar data stored in popular formats such as CSV, JSON, XML Avro, Parquet, and ORC. It uses table definitions from the storage layer to apply schema-on-read to data read. It provides faster results and lowers costs by reducing the amount of data it scans by using dataset partitioning information stored in the catalog in the storage layer. It natively integrates with the security and monitoring layer to support authentication, authorisation, encryption, logging, and monitoring. It supports table- and column-level access controls.

The federated query capability enables queries that can join fact data hosted in a data lake with dimension tables hosted in a data warehouse cluster, without having to move data in either direction. You can also include live data in operational databases in the same statement using federated queries. It provides faster results and lowers costs by reducing the amount of data it scans by leveraging dataset partitioning information stored in the catalog. It provides a powerful SQL capability designed for blazing fast online analytical processing (OLAP) of very large datasets that are stored in the storage (across the MPP data warehouse cluster as well as a data lake). The powerful query optimiser can take complex user queries written in PostgreSQL-like syntax and generate high-performance query plans that run on the MPP cluster, as well as a fleet of nodes, to query data in a data lake. This provides results caching capabilities to reduce query runtime for repeat runs of the same query by orders of magnitude. FABRIQ^n's interactive query provides concurrency scaling, which spins up additional transient clusters within seconds, to support a virtually unlimited number of concurrent queries.

Business intelligence (Visualisation)

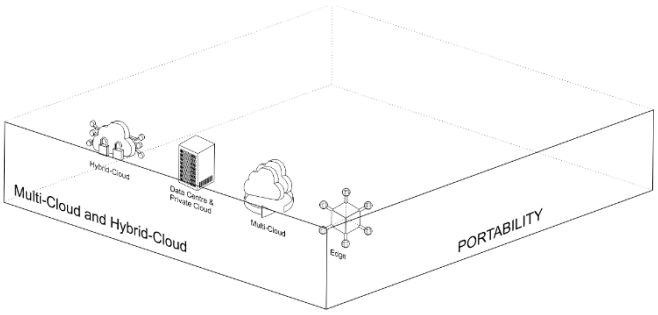
BI capability to easily create and publish rich, interactive dashboards. It enriches dashboards and visuals with out-of-the-box, and narrative highlights. To achieve blazing fast performance for dashboards, LLDM provides an in-memory caching and high-dimensional analytic engine which automatically replicates data for high availability and enables thousands of users to simultaneously perform fast, interactive analysis while shielding your underlying data infrastructure. It allows you to securely manage your users and content via a comprehensive set of security features, including role-based access control, auditing, single sign-on (IAM or third-party), private VPC subnets, and data backup.

You can also enrich dashboards and visuals with automatically generated ML insights such as forecasting, anomaly detection, and narrative highlights. Interactive Visualisation natively integrates with ALGOREUS to enable additional custom ML model-based insights to your BI dashboards. You can access dashboards from any device or embed the dashboards into web applications, portals, and websites.

Predictive analytics and ML

Enterprises typically need to explore, wrangle, and feature engineer a variety of structured and unstructured datasets to prepare for training ML models. For all use cases, including mission-critical ones, enterprises can develop, train, and deploy ML models by connecting ALGOREUS to FABRIQ^n.

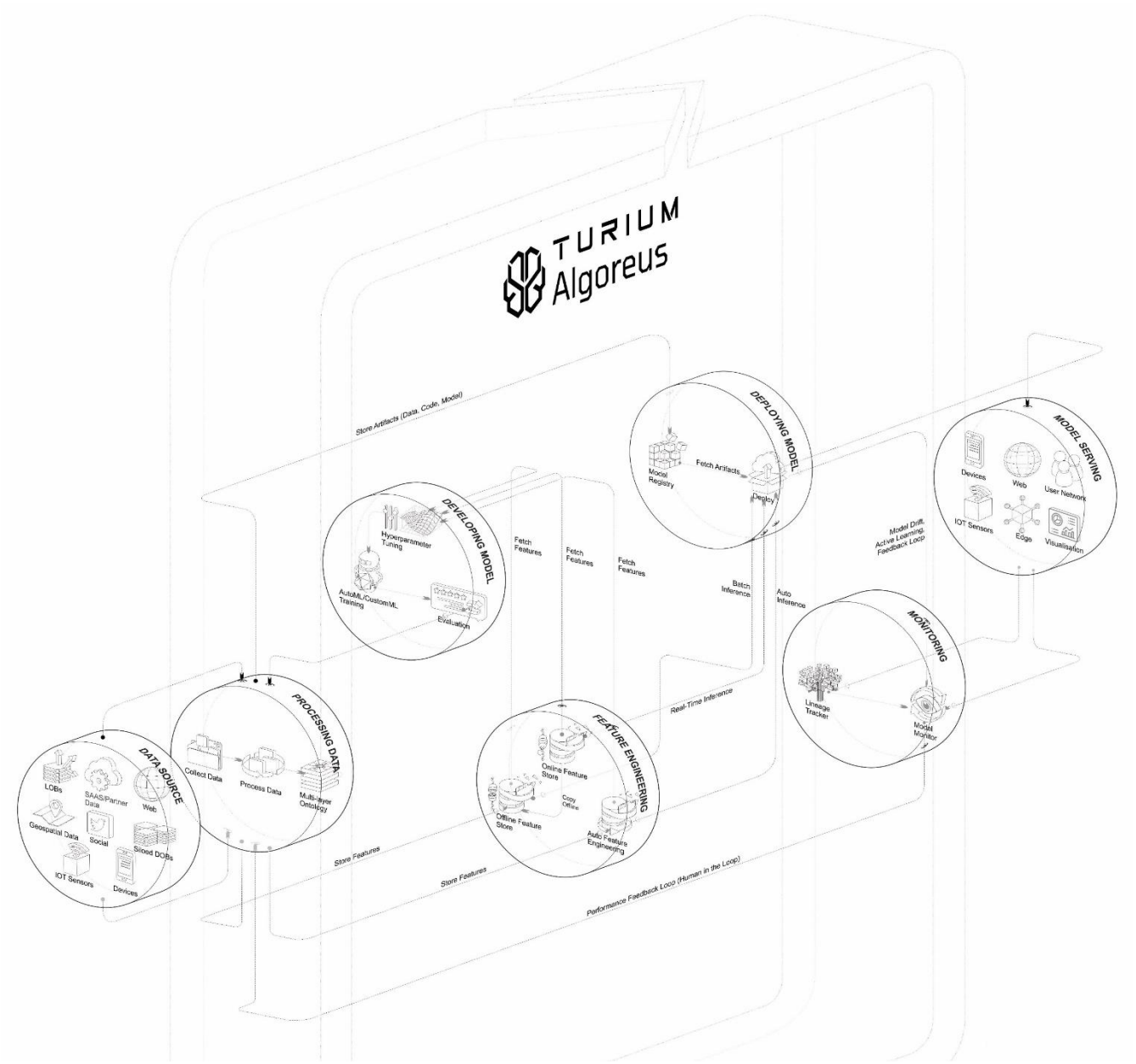
K. Portable Architecture



FABRIQ^n makes cloud data integration simpler and more agile by enabling source-to-target data migration and synchronisation without any manual coding. Enterprise teams can use the intuitive console to configure, execute, and monitor cloud data migration jobs. FABRIQ^n enables IT agility by serving as a single source of truth for all of an enterprise's diverse cloud data integration needs. It enables integration and data movement between nearly any type of source data system – such as relational databases, mainframe systems, data warehouses, and data lakes – and any of the leading cloud data platforms, including Amazon, Azure, and Google. While making it easy and fast to implement bulk data transfers to and among cloud data platforms, it also supports real-time incremental cloud data replication.

With FABRIQ^n's change data capture (CDC) technology, you can easily keep on-premises and cloud data stores in sync, ensuring that your cloud initiatives benefit from the freshest possible data. You can also securely transfer data across Wide Area Networks (WANs) by leveraging AES-256 encrypted multi-pathing. To maximise the utilisation of available bandwidth, large tables can be compressed and split into multiple configurable streams, and small tables or CDC streams can be batched together in the computation layer. To address unpredictable events like network outages, FABRIQ^n offers seamless recovery from transfer interruptions from the exact point of failure. This is accomplished by staging source data in a temporary target directory, then validating it before loading it into the target database. Thus, allowing the system to run on the cloud, on-premises, or even multi-cloud and hybrid-cloud environments at an enterprise scale.

Deploy **ALGOREUS**, an ML Spawner, for Enterprise capability force multiplier



III. Introduction of ALGOREUS

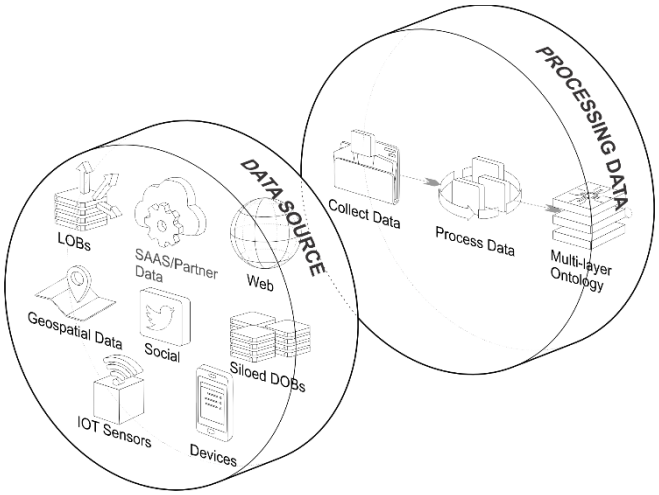
Most enterprise teams have varying levels of machine learning expertise, ranging from novices all the way to experts. To accelerate AI innovation through data-driven decision precision, you need a platform that can bring key decision-makers, operational staff, and data scientists together, offering a seamless yet flexible ML environment. This is where ALGOREUS comes in.

It provides purpose-built tools to help you automate and standardise processes across the Machine Learning (ML) lifecycle. An ML Spawner that lets enterprises: train, test, troubleshoot, deploy, and govern ML models at scale to boost productivity across all chains of command while maintaining model performance in production with confidence. It accelerates time to value with industry-leading machine learning operations and open-source interoperability. The dependable platform is intended for responsible AI applications in machine learning, with built-in fairness and explainability. ALGOREUS also includes built-in governance, security, and compliance for operating machine learning workloads anywhere.

It offers a dedicated solution for training a high-quality model with minimal effort through AutoML or training your case-specific models from scratch with CustomML and even managing those produced by third parties. We empower enterprise teams to build to protect against model drift and retain a competitive advantage by providing a solution to continuously test, iterate, and retest models before pushing them to systems at the edge. By pushing only quality transmissions, our software frees tactical bandwidth while providing greater strategic value for the strategic and operational chain of command.

IV. ALGOREUS Architecture

A. *Prepare ML Data*



You can integrate the FABRIQ^n data preparation flow into your machine learning (ML) lifecycle to simplify and streamline data pre-processing and feature engineering using little to no coding. It provides an end-to-end solution to import, prepare, transform, featurise, and analyse data.

1. Import – Connect to and import data from your data sources or integrate with the storage layer of FABRIQ^n to access the data catalog.

2. Data Flow – Create a data flow to define a series of ML data prep steps. You can use a flow to combine datasets from different data sources, identify the number and types of transformations you want to apply to datasets, and define a data prep workflow that can be integrated into an ML pipeline.

3. Transform – Clean and transform your dataset using soma-based intelligent computation (SIC) like string, vector, and numeric data formatting tools. Featurise your data using transformations like text, date/time embedding, and categorical encoding.

4. Generate Data Insights – Automatically verify data quality and detect abnormalities in your data with Data Insights and Health Report, as well as data analysis tools like target leakage analysis and quick modelling to understand feature correlation.

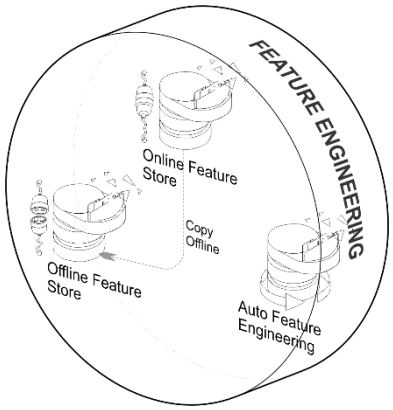
5. Export – Export your data preparation workflow to a different location. The following are example locations:

- Storage Layer in FABRIQ^n

- ALGOREUS Model Building Spawns: Use ALGOREUS Spawns to automate model deployment. You can export the data that you've transformed directly to the pipelines.

- ALGOREUS Feature Store: Store the features and their data in a centralised store.

B. Create Store and Share Features with ALGOREUS feature store



The process of developing machine learning (ML) models frequently begins with extracting data signals, also known as features, from data. The ALGOREUS Feature Store simplifies the creation, sharing, and management of features for machine learning (ML) development for data scientists, machine learning engineers, and general practitioners. Feature Store speeds up this process by minimising the amount of time spent on repeated data processing and curation necessary to turn raw data into features for training an ML model.

Furthermore, the data processing logic is written only once, and the features created are used for both training and inference, decreasing model-serving bias. Feature Store is a centralised repository for features and associated metadata, allowing for easy discovery and reuse of features.

How Feature Store Works

In the Feature Store, features are stored in a collection called a feature group. You can visualise a feature group as a table in which each column is a feature, with a unique identifier for each row. In principle, a feature group is composed of features and values specific to each feature. You can use Feature Store in the following modes:

- Online: In online mode, features are read with low latency (milliseconds) reads and used for high throughput predictions. This mode requires a feature group to be stored in an online store.

- Offline: In offline mode, large streams of data are fed to an offline store, which can be used for training and batch inference. This mode requires a feature group to be stored in an offline store. The offline store uses your data for storage and can also fetch data using queries.

- Online and Offline: This includes both online and offline modes.

Find, Discover, and Share Features

After you create a feature group in Feature Store, other authorised users of the feature store can share and discover it. Users can browse through a list of all feature groups in the Feature Store or discover existing feature groups by searching by feature group name, description, record identifier name, creation date, and tags.

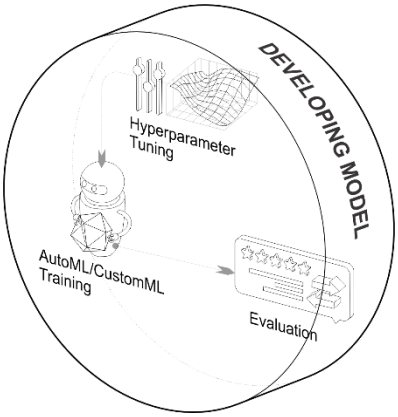
Real-Time Inference for Features Stored in the Online Store

With Feature Store, you can enrich your features stored in the online store in real-time with data from a streaming source (clean stream data from FABRIQ^n) and serve the features with low millisecond latency for real-time inference.

Offline Store for Model Training and Batch Inference

Feature Store provides offline storage for feature values. Your data is stored using a prefixing scheme based on event time. The offline store is an append-only store, enabling Feature Store to maintain a historical record of all feature values. Data is stored in the offline store in Parquet format for optimised storage and query access. Feature Store supports combining data to produce, train, validate, and test data sets, and allows you to extract data at different points in time.

C. Choose an Algorithm



Machine learning can help you accomplish empirical tasks that require some sort of inductive inference. This task involves induction as it uses data to train algorithms to make generalisable inferences. This means that the algorithms can make statistically reliable predictions or decisions, or complete other tasks when applied to new data that was not used to train them.

To help you select the best algorithm for your task, we classify these tasks on various levels of abstraction. At the highest level of abstraction, machine learning attempts to find patterns or relationships between features or less structured items, such as text in a data set. Pattern recognition techniques can be classified into distinct machine learning paradigms, each of which addresses specific problem types. There are currently three basic paradigms for machine learning used to address various problem types:

- Supervised learning
- Unsupervised learning
- Reinforcement learning

Choose an algorithm implementation

After choosing an algorithm, you must decide which implementation of it you want to use. ALGOREUS supports three implementation options that require increasing levels of effort:

- Pre-trained models require the least effort and are models ready to deploy or to fine-tune and deploy in AUTO-ML.

- If there is no built-in solution that works, try to develop one using CUSTOM-ML that uses pre-made images for machine and deep learning frameworks for supported frameworks such as Scikit-Learn, TensorFlow, PyTorch, MXNet, or Chainer.

- If you need to run custom packages or use any code which isn't a part of a supported framework, then you need to build your own custom Docker image that is configured to install the necessary packages or software. The custom image must also be pushed to an online repository like the ALGOREUS Registry.

D. Manage Machine Learning with ALGOREUS experiments

Machine learning is an iterative process. You need to experiment with multiple combinations of data, algorithms, and parameters, all the while observing the impact of incremental changes on model accuracy. Over time this iterative experimentation can result in thousands of model training runs and model versions. This makes it hard to track the best-performing models and their input configurations. It's also difficult to compare active experiments with past experiments to identify opportunities for further incremental improvements.

Model Registry automatically tracks the inputs, parameters, configurations, and results of your iterations as trials. You can assign, group, and organise these trials. Experiments can be integrated to provide a visual interface to browse your active and past experiments, compare trials on key performance metrics, and identify the best-performing models.

Because it enables tracking of all the steps and artefacts that went into creating a model, you can quickly revisit the origins of a model when you are troubleshooting issues in production, or auditing your models for compliance verifications.

Organise Experiments

Experiments offer a structured organisation scheme to help users group and organise their machine learning iterations. The top-level entity, an experiment, is a collection of trials that are observed, compared, and evaluated as a group. A trial is a set of steps called trial components. Each trial component can include a combination of inputs such as datasets, algorithms, and parameters, and produce specific outputs such as models, metrics,

datasets, and checkpoints. Examples of trial components are data pre-processing jobs, training jobs, and batch transform jobs. The goal of an experiment is to determine the trial that produces the best model. Multiple trials are performed, each one isolating and measuring the impact of a change to one or more inputs, while keeping the remaining inputs constant. By analysing the trials, you can determine which features have the most effect on the model.

Track Experiments

1. Automated Tracking

ALGOREUS Experiments automatically tracks AUTO-ML jobs as experiments with their underlying training jobs tracked as trials. Experiments also automatically tracks independently executed training, batch transform, and processing jobs as trial components, whether assigned to a trial or left unassigned. Unassigned trial components can be associated with a trial at a later time. All experiment artefacts, including datasets, algorithms, hyperparameters, and model metrics, are tracked and recorded. This data allows customers to trace a model's complete lineage, which helps with model governance, auditing, and compliance verifications.

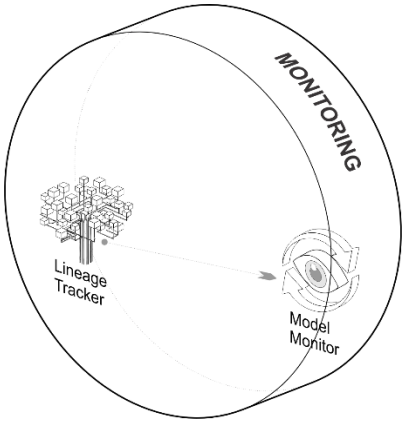
2. Manual Tracking

Experiments provide tracking APIs for recording and tracking machine learning workflows running locally on your notebooks in CUSTOM-ML. These experiments must be part of ALGOREUS training, batch transform, or processing job.

Compare and Evaluate Experiments

Experiments automatically tracks your experiments and trials, and presents visualisations of the tracked data and an interface to search the data. It automatically organises, ranks, and sorts trials based on a chosen metric using the concept of a trial leader board. Also, it produces real-time data visualisations, such as metric charts and graphs, to quickly compare and identify the best-performing models. These are updated in real-time as the experiment progresses.

E. ALGOREUS Monitoring Features



A machine learning (ML) training job can have problems such as system bottlenecks, overfitting, saturated activation functions, and vanishing gradients, which can compromise model performance.

ALGOREUS monitors profiles and debugs training jobs to help resolve such problems and improve your ML model's compute resource utilisation and performance. It sends alerts when training anomalies are found, takes action against the problems, and identifies the root cause by visualising collected metrics and tensors.

After you deploy a model into your production environment, use the ALGOREUS Model Monitor (AMM) to continuously monitor the quality of your machine learning models in real-time. AMM enables you to set up an automated alert triggering system when there are deviations in the model quality, such as data drift and anomalies. It collects log files of monitoring the model status and notifies you when the quality of your model hits certain thresholds that you preset. Early and proactive detection of model deviations through AMM enables you to take prompt actions to maintain and improve the quality of your deployed model.

F. Tune Multiple Algorithms with Hyperparameter Optimisation to find the Best Model

To create a new hyperparameter optimisation (HPO) job with ALGOREUS that tunes multiple algorithms, you can provide job settings that apply to all of the algorithms to be tested and a training definition for each of these algorithms. You must also specify the resources you want to use for the tuning job.

The job settings to configure include warm starting, early stopping, and the tuning strategy. Warm starting and early stopping are available only when tuning a single algorithm.

The training job definition specifies the name, algorithm source, objective metric, and the range of values, when required, to configure the set of hyperparameter values for each training job. It configures the channels for data inputs, data output locations, and any checkpoint storage locations for each training job. The definition also configures the resources to deploy for each training job, including instance types and counts, managed spot training, and stopping conditions.

The tuning job resources: to deploy, including the maximum number of concurrent training jobs that a hyperparameter tuning job can run concurrently and the maximum number of training jobs that the hyperparameter tuning job can run.

Basic Distributed Training Concepts

Distributed Training Solutions:

- Data parallelism: A strategy in distributed training where a training dataset is split up across multiple processing nodes, and each processing node contains a replica of the model. Each node receives different batches of training data, performs a forward and backward pass, and shares weight updates with the other nodes for synchronisation before moving on to the next batch and, ultimately another epoch.

- Model parallelism: A distributed training technique in which the model is partitioned among numerous processing nodes. The model may be complicated and contain a significant number of hidden layers and weights, preventing it from fitting in the memory of a single node. Each node contains a portion of the model, which allows the data flows and transformations to be shared and constructed. In terms of GPU use and training time, the efficiency of model parallelism is strongly reliant on how the model is partitioned and the execution schedule used to conduct forwards and backwards passes.

- Pipeline Execution Schedule (Pipelining): During model training, the pipeline execution schedule dictates the sequence in which calculations (micro-batches) are performed and data is processed across devices. Pipelining is a technique to achieve true parallelisation in model parallelism and overcome the performance loss due to sequential computation by having the GPUs compute simultaneously on different data samples.

When training models, machine learning (ML) practitioners frequently confront two scaling challenges: scaling the model size and scaling training data. While model size and complexity can improve accuracy, there is a limit to how many models can be put into a single CPU or GPU. Furthermore, increasing the size of the model may result in more computations and longer training times.

Parallelising SGD training by spreading the records of a mini-batch over various computing devices is known as data parallel distributed training, and it is the chosen distribution method in ALGOREUS to grow the mini-batch size and process each mini-batch quicker. Furthermore, Pipelining can also be utilised to achieve sequential computation on different data samples over available GPU computations for many chosen models.

G. Validate a Machine Learning Model

After training a model, evaluate it to determine whether its performance and accuracy enable you to achieve your enterprise objectives. You might generate multiple models using different methods and evaluate each. For example, you could apply different business logic rules for each model, and then apply various measures to determine every model's suitability. You can consider whether your model needs to be more sensitive than specific (or vice versa).

You can evaluate your model using historical data (offline) or live data:

1. Offline testing: Use historical, not live, data to send requests to the model for inferences. Deploy your trained model to an alpha endpoint, and use historical data to send inference requests to it.

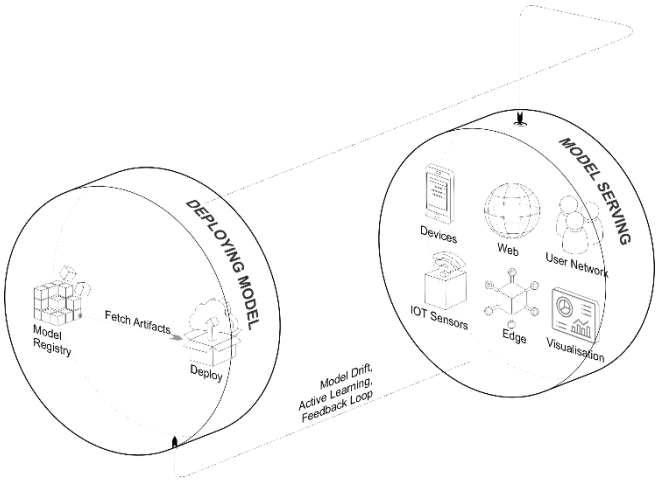
2. Online testing with live data: ALGOREUS supports A/B testing for models in production by using production variants which are models that use the same inference code and are deployed on the same endpoint. You can configure the production variants so that a small portion of the live traffic goes to the model that you want to validate. For example, you can choose to send 10% of the traffic to a model variant for evaluation. After you are satisfied with the model's performance, you can route 100% traffic to the updated model.

H. Detect Pre-training data bias

Algorithmic bias, discrimination, fairness, and related topics have been studied across disciplines such as law, policy, and computer science. An artificial intelligence system might be considered biased if it discriminates against certain individuals or groups of individuals. The machine learning models trained on datasets that exhibit these biases could end up learning them and then reproduce or even exacerbate those biases in their predictions. The field of machine learning provides an opportunity to address biases by detecting them and measuring them at each stage of the ML lifecycle. Use ALGOREUS Interpretability to determine whether data used for training models encodes any bias.

Bias can be measured before training and after training, and monitored against baselines after deploying models to endpoints for inference. Pretraining bias metrics are designed to detect and measure bias in the raw data before it is used to train a model. The metrics used are model-agnostic because they do not depend on any model outputs. However, there are different concepts of fairness that require distinct measures of bias. ALGOREUS Interpretability provides bias metrics to quantify various fairness criteria.

I. Deploy a Model

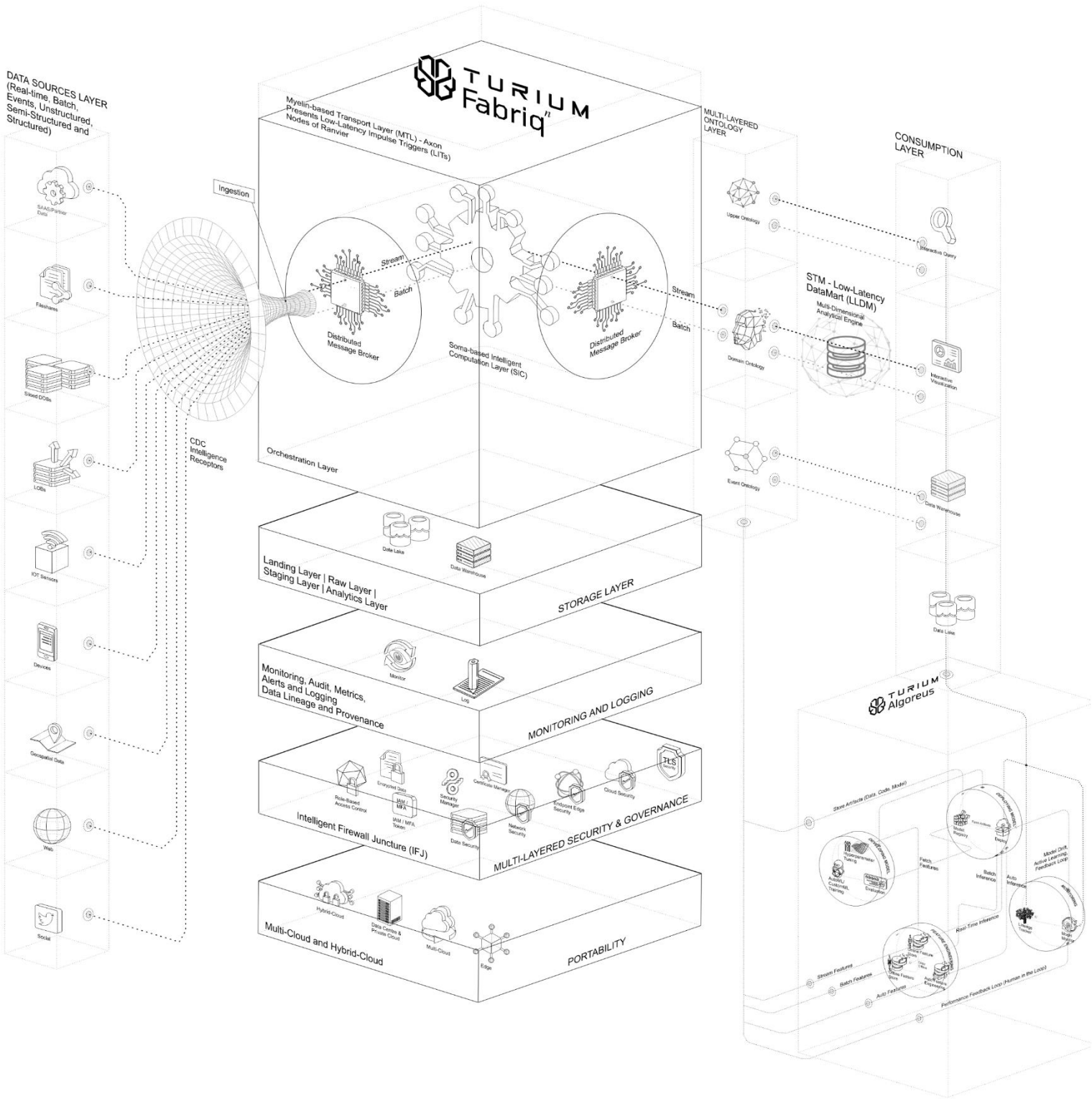


After you train your machine learning model, you can deploy it using ALGOREUS to get predictions in any of the following ways, depending on your use case:

- For persistent, real-time endpoints that make one prediction at a time, use real-time hosting services.
- Requests with large payload sizes, long processing times, and near real-time latency requirements, use Asynchronous Inference.
- To get predictions for an entire dataset, use batch transform.

To manage models on edge devices so that you can optimise, secure, monitor, and maintain machine learning models on fleets of edge devices such as smart cameras, robots, personal computers, and mobile devices.

FABRIQ^n + ALGOREUS



V. FABRIQ^n + ALGOREUS ARCHITECTURE

A. Bring your models alongside your data using FABRIQ^n's native platform integrations.

- Import models as code, libraries, or trained model artefacts from ALGOREUS
- Access versioning, branching, reproducibility, and lineage capabilities of FABRIQ^n

B. Tie your models back to the processes that drive your organisation with the FABRIQ^n Ontology.

- Define a robust foundation for AI-powered end-user workflows, with granular security and governance
- Release and inject your models directly into core applications, without adapters or glue-code
- Build feature-rich compound applications in hours instead of months

C. Enrich deployed models with decision data from your organisation's analysts, operators, and decision-makers.

- Facilitate collaboration between AI/ML and operations teams through shared applications
- Enable operators to monitor, retrain, and improve your models with real-time feedback
- Automatically write decision data back to both the Ontology and corresponding systems of action

D. Build Models in ALGOREUS with FABRIQ^n

ALGOREUS implements the complete model lifecycle, spanning problem definition, development of one or more candidate solutions, evaluation of these solutions, deployment, monitoring, and iteration. Modelling Objectives provide the backbone of the model lifecycle for any problem. Core FABRIQ^n functionality extends the traditional model lifecycle upstream (i.e., data enrichment and management) and downstream (i.e., operationalisation and feedback). The combination of end-to-end capability and interoperability means that FABRIQ^n customers can continue leveraging investments already working for them, while using ALGOREUS to augment all machine-learning challenges and provide intelligent decision precision for their enterprise operations.

E. Enrich and manage data

The fuel for any modelling use case is data. This includes not only data ingested from multiple source systems, but also data derived and captured throughout the model and use case lifecycle — business logic, feature sets, labels, model predictions, instance-level outcomes, end-user actions/decisions, and more. FABRIQ^n provides the tools to not just integrate data from anywhere, but also transform, enrich, permission, catalog, quality-control, govern, and maintain it. This is accomplished through an entire platform-wide approach such as: Lineage, which spans across datasets, logic, models, and actions to enable building automation, security, transparency, and downstream attribution. It versions data and code consistently, to simulate how logic changes impact downstream features, metrics, and model-driven decisions. Monitoring of data health, distributions, model feedback, and pipeline health to enable AI delivery at scale.

F. Evaluate and manage models

Model evaluation and management capabilities, rooted in ALGOREUS Model Monitor, are critical to streamlining and assuring successful, ongoing operationalisation of modelling projects, whether by production pipelines, user-facing applications, or other systems. It serves as a searchable catalog for model candidates, capturing models, versions, and

event-specific metadata per-submission. It then enables problem solvers to:

- Explore the set of model submissions along various metadata dimensions and model metrics
- Define model standards, and set up rails and governance around required reviews from various stakeholders, as well as release and deployment processes
- Integrate ALGOREUS' model inputs and outputs to the FABRIQ^n Ontology, enabling connectivity with operational applications and event scenarios
- Implement a systematic testing and evaluation (T&E) plan via software, leveraging managed metrics
- Perform continuous integration and continuous deployment (CI/CD) of models

G. *Operationalise models*

The ultimate goal of a modelling workflow is often deploying models to users and systems that drive decisions and actions. ALGOREUS provides a variety of options for deploying models directly, as well as tools for seamlessly incorporating model deployments into production use. Models can be deployed into managed batch inference pipelines, interactively-queryable "Live" API endpoints, or even external systems (e.g., on-prem enterprise systems, edge hardware, or multi-cloud). Pipeline-based batch deployments are suitable for recurring large-scale processing, and benefit from FABRIQ^n's data enrichment and management capabilities described earlier, such as versioning, orchestration, health checks, and lineage. Their outputs can be consumed by downstream pipelines and applications or synced to external systems. Deployments are especially powerful when models are bound to the FABRIQ^n Ontology.

Integrations enable user-facing applications to:

- Leverage direct and linked properties of input objects, execute Ontology-based scenario analyses using the model, and incorporate the model into broader simulations.
- Perform actions that propose or commit real operational changes.
- Capture actions and feedback as new data via writeback. This provides business and modelling teams a powerful data asset for monitoring, understanding, and improving production performance, as well as identifying and adapting to new circumstances

Collectively, these enable rapid construction and iteration of data-powered workflows and processes that are robust enough for an enterprise's critical path, while closing the loop with model development, evaluation, and management.

VI. Ai-Synthesise

The diversity of types of data and models makes the assessment of the most precise decision into a formidable challenge. While frequentist uncertain inference struggles in aggregating these information signals, the more flexible Bayesian approach was identified to be better suited for this quest. We use the novel Ai-Synthesise, a Bayesian-Epistemology framework which aims to aggregate all the available information, in order to provide a Bayesian probability of the risks caused with making that decision (the opportunity costs) to quantify uncertainty. Our platform solutions do that with information retrieval, machine-learning risk analysis from ALGOREUS (graphical decision-making aids), learning Bayes factors from historical data in FABRIQ, assessing quality of information and determining conditional probabilities for the so-called 'indicators' of causation for Bayesian-Epistemology.

The synthesis of evidence from multiple sources providing different kinds of information, with the aim of evaluating hypotheses and making decisions, plays a fundamental role in many domain industries such as defence. In surveillance, for instance, relevant evidence only becomes available in an unsystematic and motley way, so that evaluating hypotheses is far from the textbook ideal of interpreting a neat result from a randomised controlled trial (RCT).

Ai-Synthesis is a Bayesian framework for evidence aggregation to support timely decision making based on all the available data and models' information. The framework rests on Bayesian epistemology, which unlike Bayesian statistics enables representation of and reasoning with uncertainties attaching to arbitrary propositions. The risk-benefit profile of a decision is assessed and updated throughout the development process: after its hypothesis formula is proposed, during its synthesisation, and in the modelling post-evaluation period. This framework rests on the paradigmatic philosophical account of uncertain inference (Bayesian epistemology) in order to provide a theoretically justified probability of a decision on the basis of all the available evidence. The probability produced by Ai-Synthesis has been developed to be used for making decisions via the maximisation of expected utilities.

Bayesian epistemology is a philosophical theory about (a) what sort of beliefs and strength ('degree') of beliefs can be rational in a particular context and (b) how those beliefs should be revised upon learning new evidence. Bayesianism formalises degrees of beliefs as probabilities; it thereby inherits the formal constraints of probability calculus. It is generally very difficult to calculate conditional probabilities directly or to make a long and complex series of inferences using them. Bayesian networks offer a convenient means for graphically displaying and reasoning with probability functions. The system uses it to specify and read-off conditional independencies from a graph. Technically, a Bayesian network is defined on a set of pairwise different variables by a directed acyclic graph.

To facilitate the task, we employ abstract indicators of causality that are derived from Bradford Hill Guidelines: (a) difference-making, (b) probabilistic dependence, (c) decision-response relationship, (d) rate of growth, (e) temporal precedence, and (f) machine-learning knowledge. Conceptually, indicators of causality are testable (probabilistic) consequences of the causal hypothesis. Ai-Synthesis thus analyses the inferential process from the raw data and models to the hypothesis that a causal link holds between a decision and the uncertainty into two steps: (a) from data and models to causal indicators and (b) from causal indicators to causality.

Its Bayesian basis aids transparency. The Bayesian methodology requires that we define the prior probabilities of possible events and their interrelations within our model, as a precondition of making inferences using Ai-Synthesis. Transparency of an AI's decision-making process widens the scope of users and stakeholders who can understand (directly or indirectly) the system and evaluate the depth of their understanding.

Using machine learning in ALGOREUS, Ai-Synthesis will be enhanced in identifying, extracting, synthesising and interpreting relevant information, converting this into knowledge that can answer complex questions over causal associations.

(a) estimation of conditional probabilities of causal indicators and learning the weighting schemes of the evidential modulators from the data in FABRIQ and

(b) modelling the 'linkage between a direct molecular initiating event and an outcome at a level of organisation relevant to risk assessment'. The latter occurs through an adverse outcome pathway (AOP), that is, a conceptual construct that portrays existing knowledge concerning the linkage between that initiating event at a molecular level and the adverse outcome that can be macroscopically observed. Such 'mechanisms' play an important inferential role.

A. Learning the weighting scheme of the evidential modulators

Machine learning in ALGOREUS is used to estimate frequencies from past data, since we know whether the causal link was present and the values of the modulator variables. Note that, while traditional machine learning can help us to obtain values for the evidential modulators, there's still a problem of 'The Reference Class': the challenge of selecting the set of data from which to infer these frequencies. This is solved by using the Multi-Layered Ontology in FABRIQ. The goal is to estimate the conditional probability of an indicator variable given or its negation (and its other parent variables, if there are any). The predictive power of the causal indicators is inferred from past decisions with a suspected outcome.

B. Information retrieval

At present, most IR systems, use keywords to query and index documents. However, this traditional keyword-based IR model provides little semantic context for the understanding of user information needs. For example, a keyword usually has several meanings and its meaning is ambiguous without context. The push towards integration of semantic context according to the user's information need and the user's understanding of documents in the collection. With respect to Ai-Synthesis, evidence retrieval may boost its performances, by querying databases for all known names for an event, for similar events, domains and similar reaction-decision, as well as disentangling mechanisms of putative causal connections with respect to different events causing the same decision using the Interactive Query in FABRIQ which scans by using partitioning information of semantically rich event information stored in the catalog.

C. AI-powered graphical decision aids

Facing an increasing amount of information puts pressure not only on the way such data must be analysed, but also on the way those data have to be presented for an effective decision making. The use of graphs aids to reduce the adverse effects of information overload on decision quality both in management and communicating risks between strategic and functional users. Ai-Synthesis aids these goals by making it easier to visualise the confirmatory impact of (hypothetical) evidence and the confirmatory impact of indicators. Thus, an interactive graphical representation of strengths of associations leads to better contextual decisions based on Ai-Synthesis.

VII. Interoperability

FABRIQ^n is designed to interoperate with the full expanse of data systems. This includes tools and technologies that span traditional data, analytics, governance, and operational domains—including edge devices and rugged environments. It removes the traditional trade-offs often found with full-spectrum platforms; while FABRIQ^n's vast array of capabilities is designed to provide a coherent and complete experience, each consisting of discrete services intended to connect with existing or future technology investments.

A. *Data interoperability*

Each facet of the FABRIQ^n platform maintains a firm commitment to data formats. All data that is integrated into the platform is stored in its original format, and accessible through standard interfaces — such as REST, JDBC, and secure filesystem access. Furthermore, all transformed data is available in open formats such as parquet by default. This allows for deep connectivity with existing data platforms, systems of record, and other services within existing data architectures.

B. *Lineage interoperability*

FABRIQ^n supports comprehensive integration patterns with both necessary (e.g., attribution, lineage) and discretionary (e.g., tags, enrichments) information. FABRIQ^n's lineage services securely expose all metadata attributes that exist across projects, datasets, models, analyses, applications, pipeline orchestrations, resource health, and much more. This enables deep integration with existing data catalogs, metadata management tools, master data management tools, and other services contained within existing governance architectures.

C. *Ontology interoperability*

FABRIQ^{n}Multi-Layered Ontology pushes beyond traditional semantic definitions, and includes granular definitions for the events, event types, links, actions in the Upper Ontology, and functions that drive complex operations. All items in the Ontology layer can be accessible using REST APIs and customised through JSON-driven writing paradigms. This enables bidirectional synchronisation with existing semantic modelling tools, data catalog ontologies, domain-specific modelling tools, and upper-ontology actions.

D. *Logic interoperability*

Logic held in the code repositories in FABRIQ^n are stored within a highly available git service, and can be securely accessed both by UI-driven exports, and API/programmatic interactions for reuse.

E. *Analytical interoperability*

FABRIQ^n comes with a comprehensive set of analytical tools to empower users, but it can also work in tandem with current investments like BI and data science tools. Out-of-the-box connectors are available for common systems such as Power BI®, Tableau, Jupyter, and RStudio®. These connections enable a broad variety of users to access integrated data while using FABRIQ^n's best-in-class data management, model management, and governance.

F. *Security interoperability*

FABRIQ^n delivers comprehensive, transparent controls across all platform resources. FABRIQ^n's security services are intended to leverage existing authentication systems (for example, via SAML) for identity, and existing authorisation systems (for example, Active Directory) for permissions that can span role-based, classification-based, and purpose-based regimes. Dynamic and retroactive access to all security information in FABRIQ^n is accessible through the platform's REST APIs.

The security architecture of FABRIQ^n meets the CORBA security interoperability standards for authentication, delegation, and privileges. In the security architecture, the Security Authentication Service (SAS) protocol is used to exchange tokens in the service contexts of General Inter-ORB Protocol (GIOP) request and reply messages in order to build security contexts. In addition to GIOP, CORBA has

defined Environment-Specific Inter-ORB Protocols (ESIOPs) that can support interoperation over specific networks. The primary transport defined by CORBA for ESIOPs is DCE which includes such features/services as Kerberos, time services, and RPC. SAS requires transport layer security (TCP/IP, SSL/TLS), which adds two further layers for client authentication and delegation.

A client may utilise the attribute layer to push or transport security attributes, such as an identity, to a target server, where they may be used in access control decisions. The SAS protocol is built on the transport layer, which includes message protection, target-to-client authentication, and client authentication. The SAS protocol is divided into two layers: the client authentication layer and the security attribute layer. Identity assertion, privileges authorisation attributes, and proxy endorsement are all part of the security attribute layer.

1. Identity assertion

When a request is made using RMI/IIOP, the attribute layer's identity assertion feature is utilised to assert an identity from a client process to a server process. A statement made by one entity to another to accept an identity on its behalf is referred to as an assertion. A client can assert an identity that represents the subject that was effective at the moment the remote resource was started. In addition to the user's identity token, the client process sends its own identity in either the authentication layer or the transport layer. By completing trust validation, the target server verifies that the client process may assert its identity. If the target server trusts the client, it will utilise the asserted identity to establish a server-side subject that represents the user who was active at the time of invocation at the client process. A user with a Principal Name identity token can be asserted by the client. The format of the primary name is determined by the registry configuration at the client process. The anonymous identity token type is also supported, and when such a token is received, the server utilises the unauthenticated subject.

2. Authentication layer

When a request is made using RMI/IIOP, the Authentication layer is used to transfer authentication information from a client process to a server process. The authentication layer may include a token supplied by the client and used by the server to authenticate the client. The authentication layer supports a variety of token types. For example, the token is used to send the client's username and password, which are validated against the target server's registry. With either token type, the token is used to authenticate the remote user at the server process and create a Subject representation of the Subject that was effective at the client side before the client started the remote object. When identity assertion is enabled, the authentication layer can contain security information representing the client identity, while the identity assertion token reflects the actual remote user at invocation time.

3. Transport layer

The transport layer is used to secure the SAS protocol request message and to provide client certificate authentication from a client process to a server process. The transport layer's primary job is to provide security features for the transfer of SAS protocol messages from a client to a server process. Encryption, signature, or both can be used to protect messages.

When the authentication layer is not used, the transport layer serves as a source of authentication content. If identity assertion is enabled but not the authentication layer, the client process identity is retrieved via the transport's client certificate chain. The client's certificate chain is authenticated by the target server process by mapping it to a user in its registry. The distinguishing name of the certificate issuer is used to assess if the client may be trusted to claim an identity.

When the actual remote method call is started at the target server process, the subject obtained from mapping the client certificate chain is utilised as the caller subject if the identity assertion and authentication layer are not enabled. This also applies when the target server's authentication layer is supported, but not required, and the client did not send an authentication token and an identity token.

VIII. DATA SOVEREIGNTY

Xaana.Ai designs and operates the entire platform stack using best security practices across the software development lifecycle (SDLC). Xaana.Ai is certified with ISO 27001/27002 security controls to protect your data. This includes procedures for monitoring threats, encryption, physical access control, and more. The company also complies with the Information Security Manual (ISM), Protective Security Policy Framework (PSPF), the Privacy Act 1988, GDPR, and HIPAA.

Redundancies are fully managed by us, with hosting and backup stored in Australian data centres with real-time servers, applications, and disaster recovery in place. The facility is designed to run 24x7x365 and employs various measures to help protect operations from power failure, and network outages. The data centres comply with industry standards such as ISO 27001/27002 for physical security and availability. They are managed, monitored, and administered by Xaana.AI operations personnel. Third parties undergo a review process, and an approved vendor list is established and used. These vendors are required to comply with security policies and are audited for compliance.

Starting from our back-end software developers all the way to our partners, Xaana.Ai's staff are continually trained on the most up-to-date information security protocols and procedures to monitor incidents and threats. The data is never used for any other purposes other than the provisioning of services to you and always stays in Australia. It is protected at all times with robust 24/7 infrastructure, servers, and application monitoring systems. Rest assured, we have best-of-breed monitoring software and alerting mechanisms in place.

Xaana.Ai does not use customer data for advertising. Xaana.Ai takes privacy very seriously and never uses enterprise customer data for marketing or advertising purposes.

Xaana.Ai logically segregates storage and processing for different customers through specialised technology engineered to help ensure that customer data is not combined with anyone else's data, and to help prevent one malicious or compromised customer from affecting the service or data of another. Xaana.Ai also takes strong measures to protect customer data from inappropriate use, access, or loss. Additional safeguards include proper controls for administrative access, such as secure user authentication.

When Xaana.Ai engineers or subcontractors need access to customer data, such as when troubleshooting an issue, they have to be explicitly granted access by the customer, and access is revoked when it is no longer necessary. The operational processes and controls that govern access to and use of customer data are protected by strong controls and authentication, such as single-sign-on and multi-factor authentication, which helps limit data access to authorised personnel only.

Xaana.Ai does not provide any third party (including law enforcement, other government entity, or civil litigant) with direct or unfettered access to customer data except as directed by the customer. When a government or law enforcement request for customer data is received, we always attempt to redirect the third party to obtain the requested data from the customer. For valid requests that cannot be redirected to the customer, Xaana.Ai discloses information only when legally compelled to do so, and only to the extent specified in the legal order. We promptly notify customers of any third-party request and provide a copy, unless legally prohibited from doing so. Xaana.Ai never provides any government with its encryption keys or the ability to break its encryption.

Xaana.Ai strives to be transparent in its compliance, security, and privacy practices. This compliance means both government and commercial customers can have confidence knowing they comply with Australian legislative and certification requirements when using our services.

Xaana.Ai recognises the Traditional Custodians of Country throughout Australia. We acknowledge First Nations Peoples as the Traditional Owners, Custodians, and Lore Keepers of the world's oldest living culture and pay respects to their Elders past, present, and emerging.

ABOUT XAANA.Ai

The environment is now more technologically challenging than at any time in history due to knowledge silos and the turf wars they enable within an enterprise. When we looked at the available technology, we saw products that were not complete on their own to provide total solutions, automated methods that fell short in the face of the adaptive situation, and an access rule that compelled enterprises to compromise on collaboration and data security in unacceptable ways. We saw a demand for an integrated AI technology that would require a different kind of company to develop. That's how Xaana came to be.

For years, we've consulted alongside our customers to build our Artificial Intelligence platform backwards, starting from solving the mission-critical enterprise problems. Our goal is to provide the world's most seamless experience for working with real-time data, one that enables people to gain rich insights into their dynamic environment. To achieve this, we create intelligent platform products that unlock the value of big data by layering industry-preferred applications on top of a fully integrated human-augmented and machine-assisted analysis.

We pride ourselves on our core value of Collaborative Intelligence (CINTEL) to move beyond machines replacing people or automating their jobs and instead focus on maximising the benefits of using human capabilities, rich data, and artificial intelligence. With our Collaborative Intelligence (CINTEL) mindset, we believe that institutions today can simplify problems, particularly those of a complex, changeable, and difficult-to-define context, to join us in becoming resourceful for positively impacting the world.

25 + years of combined team experience, deeply committed to assisting you in overcoming your greatest challenges and making your healthy data vision a reality. Over 100+ customers across Australia have chosen Xaana.Ai for an integrated AI experience.

For more information, please visit https://www.xaana.ai/
and
follow us on LinkedIn: @Xaana.Ai

Ushering a New Era in Ai