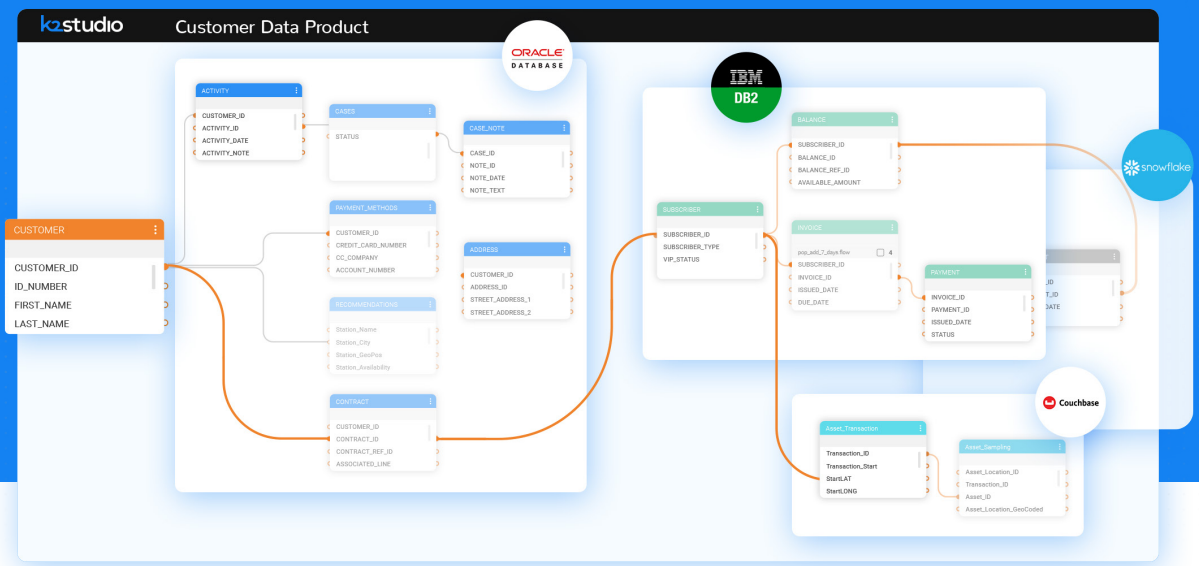
k2view

WHITEPAPER

K2view Data Product Platform



Introduction

Organizations that thrive in today's economic climate move fast, not only to quickly respond to market opportunities and threats, but also to proactively and continually, innovate and differentiate.

To achieve this level of performance, an enterprise needs an agile data foundation, where information can be leveraged for its strategic value – for example, in delivering a proactive and personalized customer experience across all touchpoints.

However, data is typically fragmented across applications and locked in silos and legacy systems. The need to integrate, unify, and cleanse the data – and then make it accessible to authorized data consumers at scale – is a massive burden that hinders agility and innovation.

Best-in-class organizations democratize data access across business units, elevate trust in the data, and make data the foundation for innovation and decision making.

K2view makes this possible through its patented Data Product Platform, which manages data products that create and govern a complete and compliant dataset for every business entity (e.g., customer, device, payment, etc.) – on demand and in real time. The dataset is always in sync with its underlying sources, adapts to changes in the source structures, and is instantly accessible to any authorized data consumer.

Data Product Platform fuels many operational use cases, including test data management, data masking, synthetic data generation, customer 360, data migration, legacy application modernization, and data pipelining – to deliver business outcomes in less than half the time and cost of any other alternative.

The platform inherently supports modern data architectures – data mesh, data fabric, and data hub – and deploys in cloud, on-premise, or hybrid environments.

CHAPTER 1

Data Product Platform Overview

K2view Data Product Platform is the foundation for operational data management in the enterprise.

By managing data products that continually sync, transform, mask, and serve trusted data through millions of Micro-Databases™, the platform fuels both operational and analytical use cases, and deploys in any architecture.

K2view Data Product Platform is built on a distributed, shared, and linearly scalable architecture, with in-memory computer processing, and optimized disk storage.

Beyond flexible on-prem / cloud / hybrid deployment, K2view Data Product Platform addresses the 3 most important challenges of enterprise data management:

1. Provisioning complete, clean, and compliant data for operational and analytical workloads in real time, thanks to its patented data product and Micro-Database™ approach.
2. Implementing the platform in any data architecture, from a highly-federated data mesh to a more centralized data fabric, or multi-domain data hub.
3. Delivering value in weeks, with no-code/low-code tooling, and by integrating to the organization's existing data and analytics technology stack.

This whitepaper will discuss the architecture and key capabilities of Data Product Platform.

Organizing and managing data by business entities

Traditionally, business applications organize and persist data into database tables based on the business entities being managed by the application (e.g., customer data, invoice and payment data, campaign data, product data, etc.).

All the instances of a particular business entity are stored in the same table (e.g., a table containing all the personal information for all customers; another table storing all invoices for all customers; and another table containing all the payments against the invoices).

The result is very large tables that must be queried using complex joins, every time a data consumer wishes to answer a multi-dimensional query on a specific entity (e.g., "What is the total amount that customer "John Smith" was invoiced in the past 6 months?").

> K2view Data Product Platform revolutionizes data management by organizing and managing the dataset for each business entity – e.g., a specific customer – in its own high-performance, always-fresh database, the "Micro-Database".

Organizing data by entities, makes it simple and quick for data consumers to access the entity datasets they need.

K2view Data Product Platform: Schematic diagram

A Micro-Database for each business entity instance

The K2view Data Product Platform integrates, processes, and delivers data via data products.

A data product integrates and organizes a specific entity's data, from all relevant source systems, into its own Micro-Database – one Micro-Database for each individual entity. The data product also ensures that the Micro-Database is always current with fresh data, and instantly accessible, in milliseconds, to authorized data consumers.

> In the case of customers, for example, a Micro-Database is created and managed for each individual customer. The Data Product Platform manages billions of Micro-Databases concurrently.

What's in a data product?

A K2view data product has a single definition (metadata), and as many instances of its data, as there are business entity instances.

Data product metadata

- **Schema**, containing all the tables and fields that capture a business entity's dataset
- **Sync rules**, for how and when entity data is synced with the source systems
- **Logic**, for unifying, transform, and enriching the entity's raw data
- **Governance**, for ensuring data quality and privacy of the entity dataset
- **Accessibility**, defining data ingestion and delivery methods via any combination of JDBC, APIs, Kafka, CDC, and messaging.
- **Access controls**, including authentication and credential validation

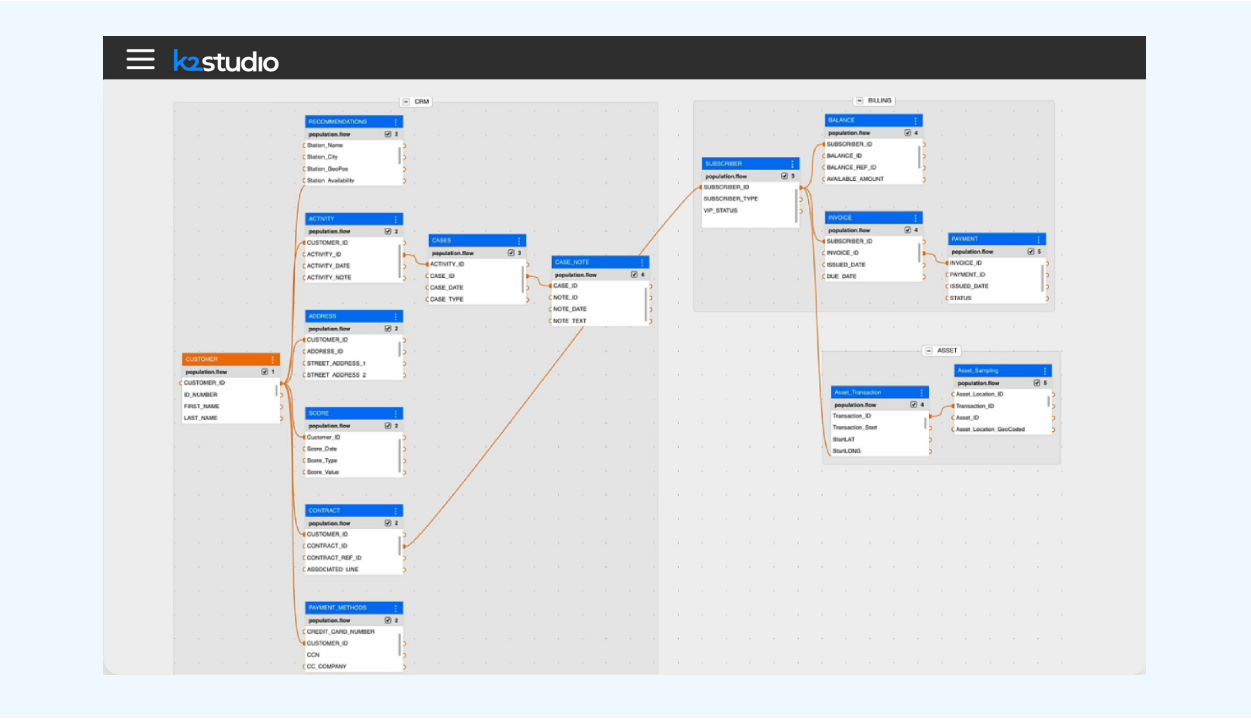
One Micro-Database per business entity instance

- **Unified,** and cleansed
- **Enriched,** with real-time and offline insights
- **Cached,** persisted, or virtualized
- **Secured,** with its own 256-bit encryption key
- **Logged,** so that every data change is tracked
- **Augmented,** with performance and usage active metadata

Defining the data schema for the data product is automated, in large part, via an auto-discovery function, and completed using a graphical data modeling studio – K2view Studio.

The result is a business-oriented data structure that corresponds to a business entity, and which contains the table and object definitions, derived from as many source systems as necessary.

For example, a customer data product may require 9 tables from the CRM system, running on MySQL; 4 tables from the Billing system, running on Oracle; and 2 tables from the Asset system, residing on IBM DB/2.



Multi-source data product schema

The K2view platform integrates, processes, and organizes data via data products, where the data for each instance of a data product is stored in its own Micro-Database. Every Micro-Database is compressed and individually encrypted with a unique key, enabling unparalleled performance, security, availability, and sync capabilities.

Micro-Databases are easily accessible by any authorized data consumer, in the data access method of their choice:

- **Synchronized** (“pull”): Web services, SQL
- **Asynchronized** (“push”): Messaging, Streaming, CDC

Benefits of the Micro-Database

Managing the data for each specific business entity in its own Micro-Database means that its data, which is scattered across multiple data sources, is now connected, governed, and managed as an integrated unit.

So, instead of scanning through massive tables and complex joins, across different technologies and systems, a query related to a business entity is performed against one Micro-Database. This makes querying a specific business entity possible in milliseconds.

Likewise, data processing for a specific entity can be performed in-flight. For example, cleansing and unifying an entity’s data, augmenting it with real-time and offline insights, and masking the data for compliance with data privacy regulations – are all performed in real time.

Additionally, every Micro-Database can be secured inflight with its own encryption key, achieving unparalleled granular data security to eliminate the possibility of a mass data breach.

Micro-Database versus alternatives

The following summarizes the differences between traditional data representation, and that of data product Micro-Databases in K2view.

Traditional RDBMS

- Data is scattered across multiple systems and databases (e.g., CRM, billing, etc.)
- Each system is independent, with its own technology, database, and data formats
- It is difficult to unify and process the data corresponding to a single business entity (e.g., one customer), and impossible to do so in real time

Data Lake

- Data is stored and distributed in a big data system (e.g., Hadoop, Snowflake)
- There is no business logic in storage
- A single data query is compute-intensive because it requires sifting through massive amounts of data

K2view Micro-Database

- The dataset for each entity is stored, cached, or virtualized as a unit
- The Micro-Database is continually synced with its source systems
- Data integration, processing and delivery are core features

CHAPTER 2

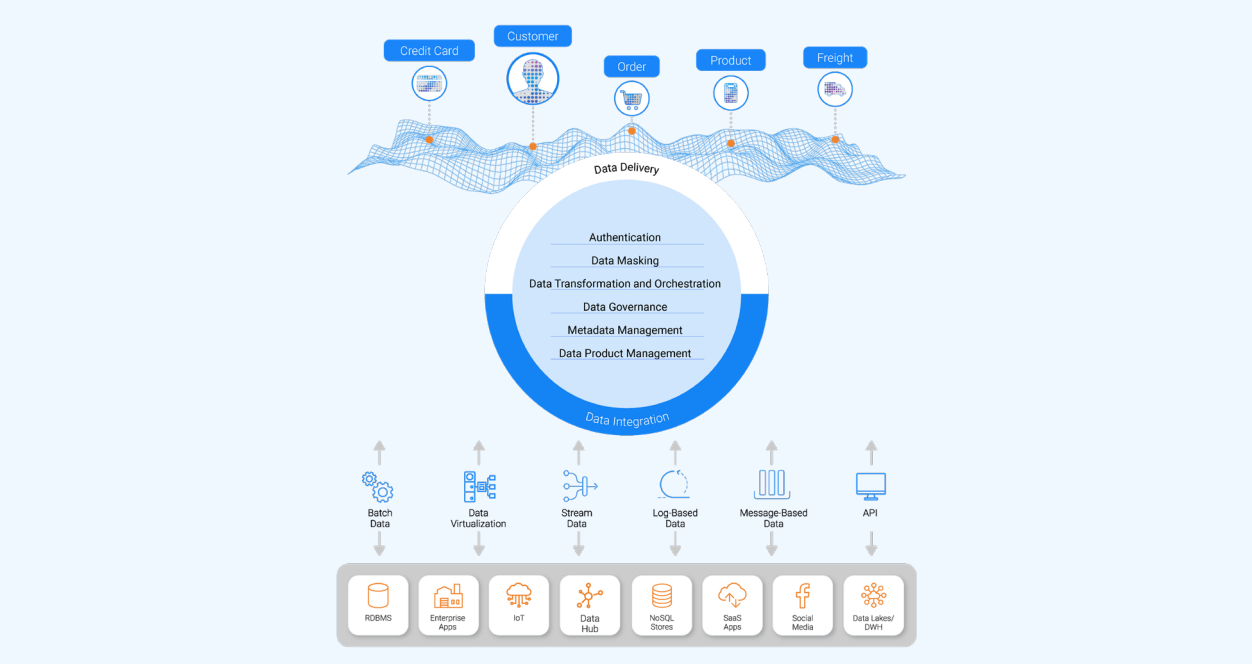
Architecture

Data products in the K2view platform may be defined and managed by data and analytics teams, in a centralized data fabric architecture. Alternatively, data products may be defined and accessed by business domains, in a federated data mesh operational model.

Both approaches are supported by the Data Product Platform, including a hybrid approach, where certain aspects of data management are centralized, while others are federated.

Data Fabric

The diagram below illustrates the K2view platform in a centralized data fabric architecture:



Data Product Platform in a data fabric architecture

Data Mesh

In a data mesh architecture, pictured below, K2view nodes are deployed in clusters aligned with business domains – which manage data products – to create, govern, and access the data products for their respective data consumers.

A central cluster is configured to enable select data management functions that the organization would like to centralize (e.g., data security, common data transformation functions, or shared services).

All nodes are managed and administered as a single integrated cluster, or as multiple independent clusters – depending on security and performance considerations.

Data Product Platform in a data mesh architecture

Platform components and functionality

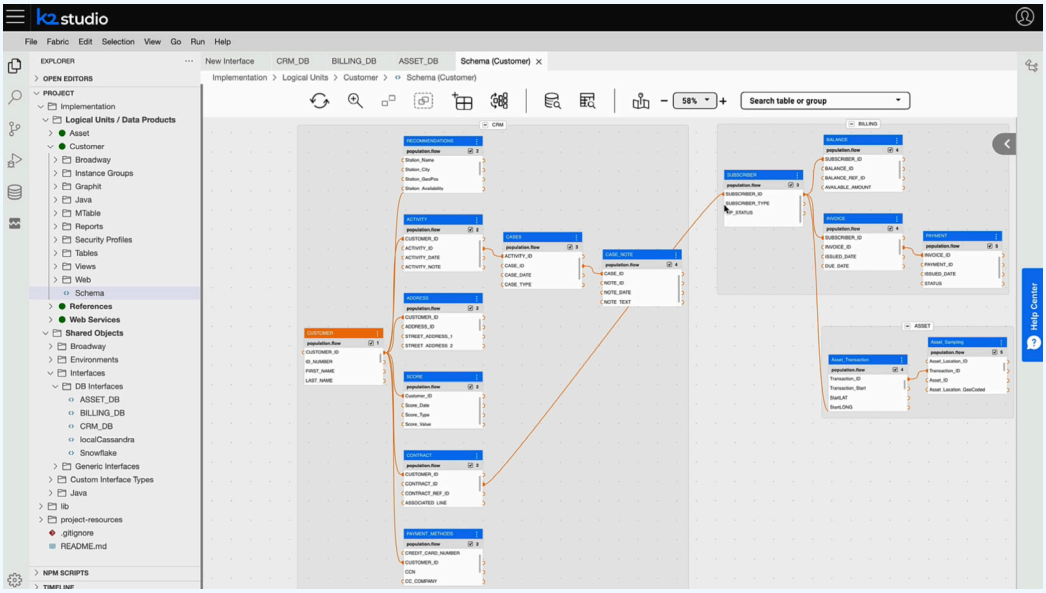
Below is an overview of the main components of the K2view platform. The core functionality of these components is unaffected by the chosen architecture pattern (data fabric / data mesh / data hub).

K2studio

K2studio is a web-based graphical IDE for managing the entire lifecycle of a data product.

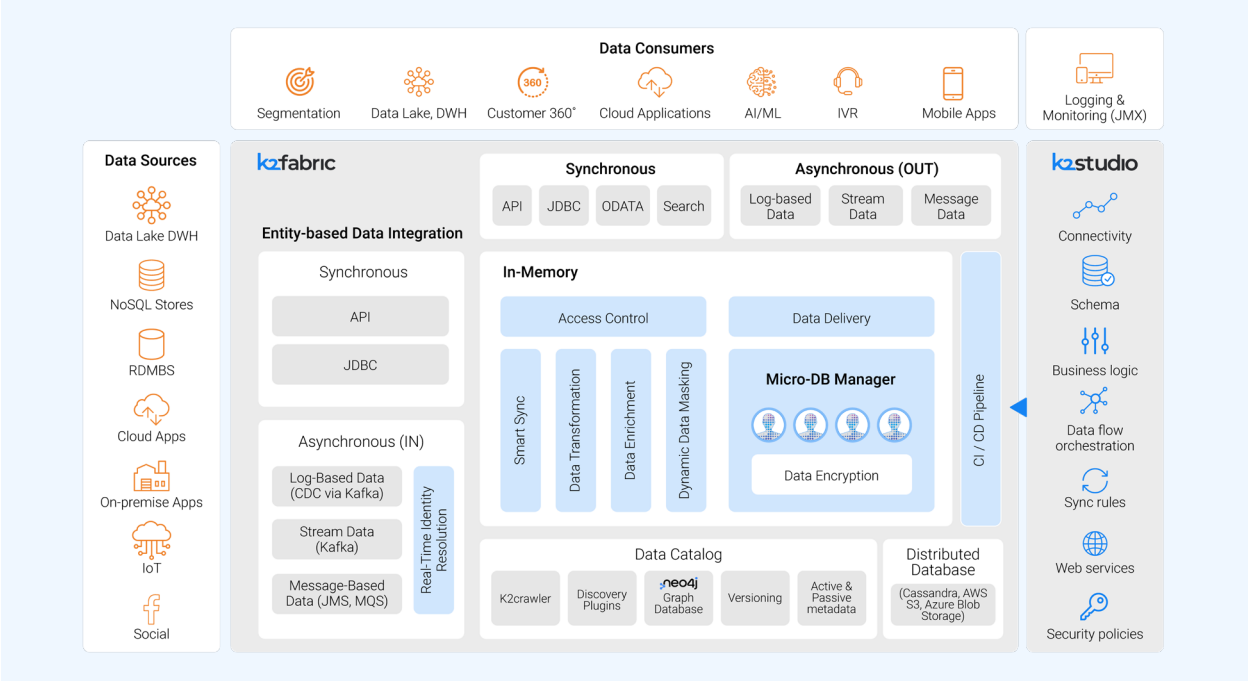
Data teams create the versioned metadata of every data product, including data schema, data integration and delivery methods, synchronization policies, data transformation logic, data privacy and quality policies.

K2studio also features auto-discovery of the data product's schema from underlying source systems and enabling the data engineer to manually adapt and enhance the schema as required.



K2studio is a web-based IDE for data engineers

Data Product Platform runtime components



Data Product Platform schematic architecture

- **Data integration and delivery** via:
    1. **Synchronous methods:** SQL queries (JDBC), or web service APIs or ODATA
    2. **Asynchronous methods:** Streaming (Kafka), CDC, or messaging (JMS or MQ)
- **Access control,** managing user access and restrictions to data products
- **SmartSync,** of a data product's data with its source systems, based on the sync policies for each data element of the data product
- **Data transformation** performed in real time, based on pre-defined functions that are applied to the entity dataset
- **Data enrichment,** based on custom business logic
- **Data Masking,** performed inflight as required
- **Micro-Database Manager,** that can:
    1. **Retrieve** the appropriate Micro-Database from the distributed database into memory (if it is not already in memory)
    2. **Create** a Micro-Database on the fly (by populating it with the data retrieved from the data integration layer), if one didn't previously exist
    3. **Persist** each Micro-Database data in a distributed database or virtualize it (based on the data product's metadata settings)
    4. **Encrypt** each data product instance using its own unique data encryption key

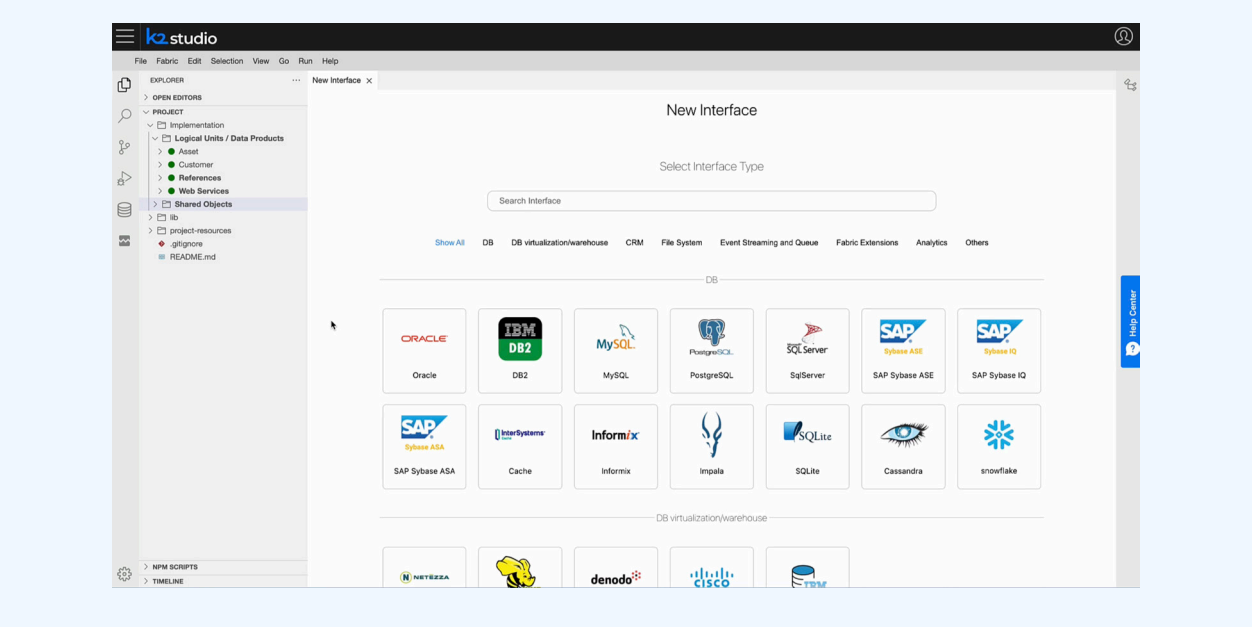
Data consumers

Data consumers are the users and different applications accessing and updating data products across the enterprise. A data consumer is authorized by the data product prior to accessing its data.

Source systems

Source systems refer to an enterprise's current business applications and data stores. K2view integrates data from, and updates to, these source systems – regardless of their structure and technologies.

Source systems include relational and non-relational databases, data lakes and data warehouses, SQL and noSQL data stores, on-premises and on-cloud, legacy and modern cloud-based applications.



K2view integrates to hundreds of SQL and NoSQL DBs,, file systems, and data endpoints

Distributed database

1. Data Product Platform manages and physically stores Micro-Databases in a distributed database, such as Cassandra or Microsoft Blob Storage. Data can be virtualized (without persisting the data), as described in the Data Integration section in the next chapter.
2. By leveraging state-of-the-art, enterprise-grade distributed databases, K2view is one of the world's most efficiently distributed operational data management platforms – without shared resources, or single points of failure.
   For example, the linear scalability of Cassandra has been demonstrated by Netflix and with documented benchmarks available online.
3. K2view is a linearly scalable platform which runs on commodity hardware that ensures the lowest total cost of ownership.
4. K2view leverages the underlying distributed data storage as a base storage layer via a key-value pair, where each Micro-Database is stored with its own unique key. The key is the data product instance ID, and its associated value is a compressed and encrypted database file containing the Micro-Database data. This formula gives K2view a very simple, structured, and efficient way to access distributed data.
5. Communication between K2view and the distributed database is straightforward, and triggered in 2 distinct cases:
   - **Smart Sync retrieves data from a Micro-Database**
     Smart Sync checks if synchronization with source systems is required (based on the defined sync policies). If not, it triggers the retrieval of data from the relevant Micro-Database from the

distributed database. For this type of transaction, K2view simply retrieves the value associated with the corresponding key (data product instance ID), using the native capabilities of the distributed database.

- **The Micro-Database Manager writes data onto disk after compression**
  After a Micro-Database is retrieved and processed in memory, the data may need to be persisted into physical storage (based on the data product's settings). In this case, the data is written into the distributed database key-value storage.

  As both cases show, the interaction between Data Product Platform and underlying distributed datastore is limited to reading data from, and writing data to, distributed database nodes, using the native protocols of distributed databases.

CHAPTER 3

Operational Data Management

Overview

Data Product Platform features a set of embedded data management functions to create and manage data products that ingest, process, and deliver trusted data by business entities. Key data management modules include:

1. Data integration and delivery
2. Data catalog
3. Data governance
4. Data processing
5. Data sync
6. Change management

These data management capabilities differentiate K2view from other data management platforms. They allow data to be retrieved, validated, and augmented – automatically and in real time – without any transformation scripts or third-party tools.

1. Data integration and delivery

K2view takes a data product approach to data integration, allowing enterprises to integrate and deliver data from any data source – on premises or in the cloud – to any data consumer, using any data delivery method.

With a data product approach, data engineers create and manage reusable data pipelines that integrate, process, and deliver data by business entities (customers, orders, loans, etc.).

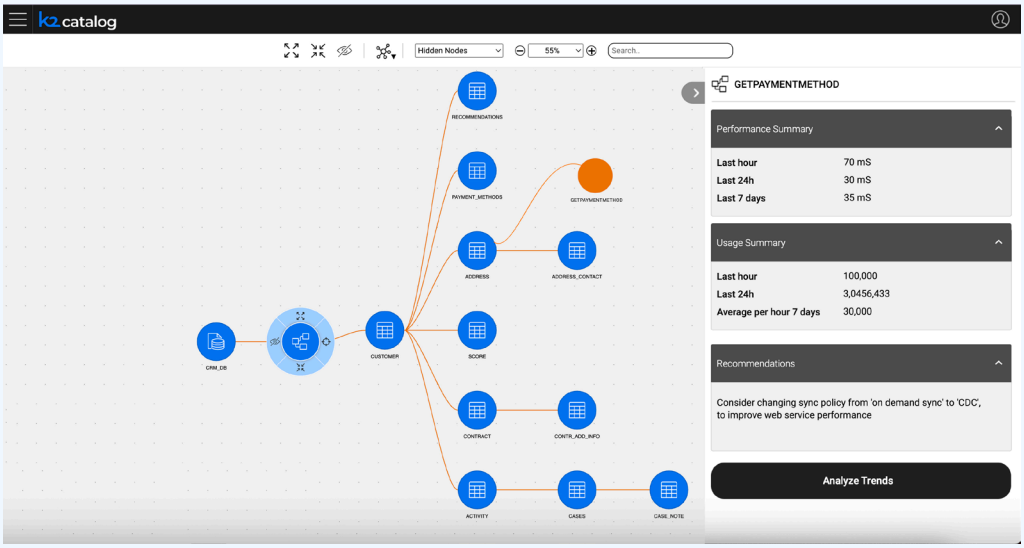
K2view data integration includes:

- No-code and low-code tooling to support varying levels of data engineering expertise
- Built-in connectors to hundreds of data sources and applications, including on-prem and cloud systems
- Support for all data delivery styles: Bulk ETL, data streaming, data virtualization, log-based data changes (CDC), message-based data integration, Reverse ETL, and APIs
- Automated data ingestion based on the data product schema

2. Data catalog

K2view Data Product Platform features a modern graph DB data catalog, which auto-discovers and auto-classifies data assets, enabling rapid understanding of how to use data across the enterprise.

The catalog enables data teams to visualize and analyze data assets and the relationships between them, including passive and active metadata, and to quickly define a data product schema.



Embedded, graph-based data catalog

With K2view Data Catalog, data teams can:

- Auto-discover data assets, regardless of data source
- Automatically classify data assets to enable quick search and understanding
- Keep track of changes, with version management and rollback capabilities
- Visualize both passive (design time) metadata and active (runtime) metadata
- Generate the data product schema directly from the data catalog
- Assure data quality and enforce data privacy via metadata classifications
- Benefit from graph DB architecture, enabling enterprise scale and flexibility
- Share the data catalog metadata with 3rd-party enterprise data catalogs

3. Data governance

K2view applies a data product approach to data governance, which is business-oriented and real-time at its core.

Data products can be created and governed by business domains, in a data mesh architecture and operating model. Alternatively, they can be engineered and governed by centralized data teams in a data fabric architecture.

The data product approach to data governance forges a common language with the business and makes data governance easier to measure.

K2view Data Governance supports:

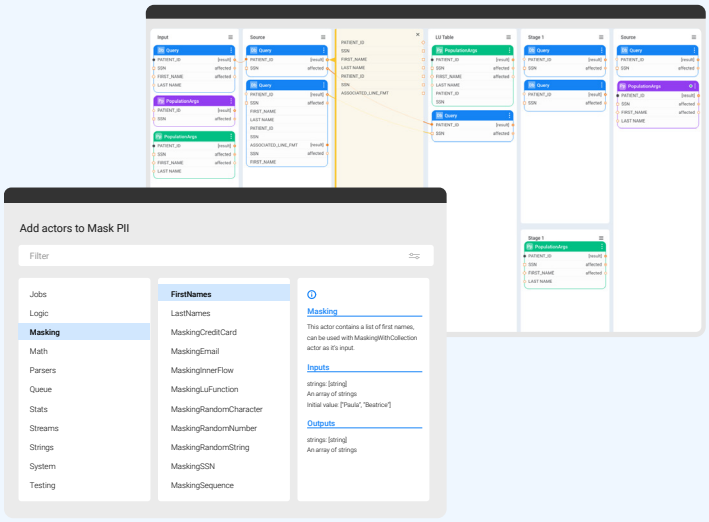
- Rule-based data quality enforcement and exception workflows
- Static and dynamic data masking for sensitive data, PII, and PHI protection
- Data tokenization, for the protection of sensitive Card Holder Data (CHD)
- Data lineage, for source-to-consumer data traceability
- Automation of Data Subject Access Request (DSAR) workflows
- Data usage and active metadata analysis

4. Data Processing

Broadway is the K2view module used to design data movement, its transformation, and the orchestration of data flows. Featuring a powerful user interface for creating and debugging business and data flows, Broadway also provides a high-performance execution engine.

Broadway is used in the K2view platform wherever data movement and orchestration are needed. For example:

- Populating a data product Micro-Database from external databases or REST APIs
- Pipelining data to external systems based on CDC or batch processes
- Subscribing to a message bus and consuming messages
- Orchestrating scheduled activities through the platform's job system
- Transforming data for various use cases (e.g., data masking or enrichment)

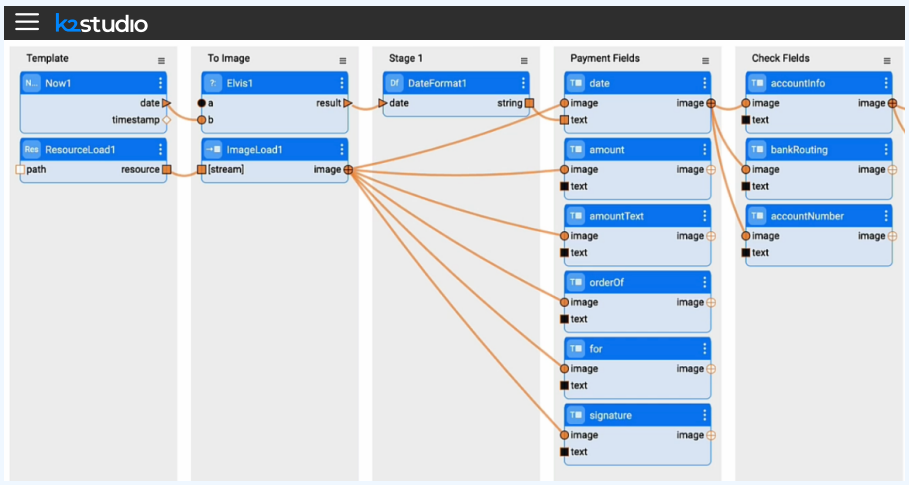


Dozens of out-of-the-box, customizable data processing functions

Broadway is a flexible engine that can be leveraged anywhere in the platform's architecture layers for endless use cases.

Data and business process orchestration

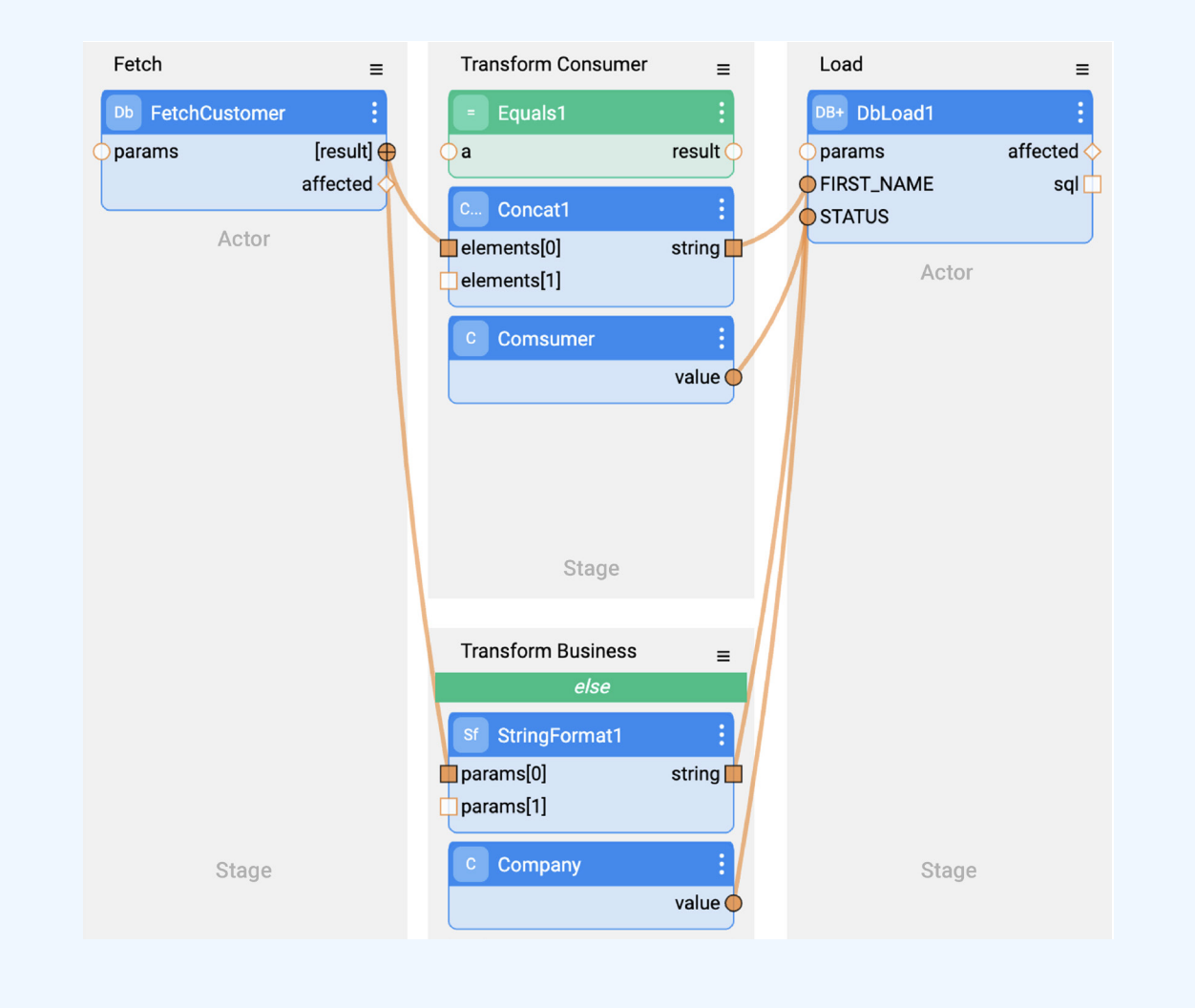
A Broadway flow is built from Stages which are executed from left to right. A flow can be split into different execution paths based on conditions. More than one Stage can be executed at each fork in the path.



Graphical orcchestration of data processing and flows

Data Transformation

Each Stage can contain one or more Actors which are reusable pieces of logic, with input and output arguments, that can be assembled together to create complex logic. Actors are executed by Stages.

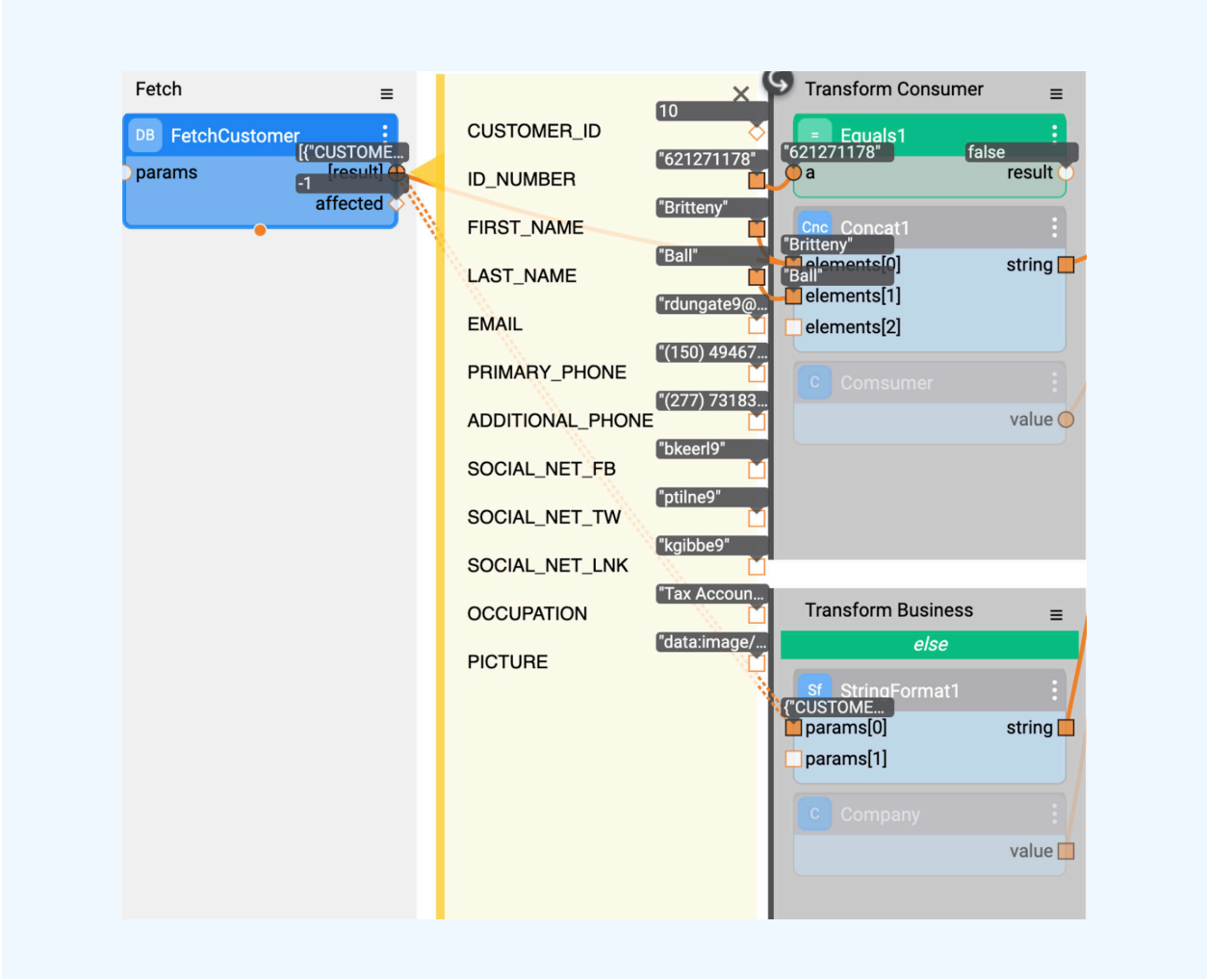


Broadway flows are comprised of stages and actors

An entire Broadway flow can be exported and encapsulated into an Actor and then be reused across flows. This is a powerful tool for reusing logic and working with highly complex flows.

Data Inspection

When Broadway transfers data between Actors, the actual data is displayed in the Broadway flow. Complex data types (objects, arrays) are automatically detected and analyzed, and both metadata and data are visually rendered for easy configuration and debugging.



Broadway data inspection

The example above showcases how Broadway automatically identifies the data structure of the FetchCustomer Actor. This enables selecting specific data elements and passing them to the appropriate Actors.

5. Data Sync

A data product retrieves data from the underlying sources into the relevant Micro-Database based on the sync policy defined for the data product:

- On-demand sync
  Data synchronization is invoked by web service calls, batch scripts, or directly via the K2view Admin.
- Event-based sync
  Triggered using the principles of Change Data Capture (CDC). In this mode, K2view automatically captures changes to data elements in the source systems that are part of its schema.

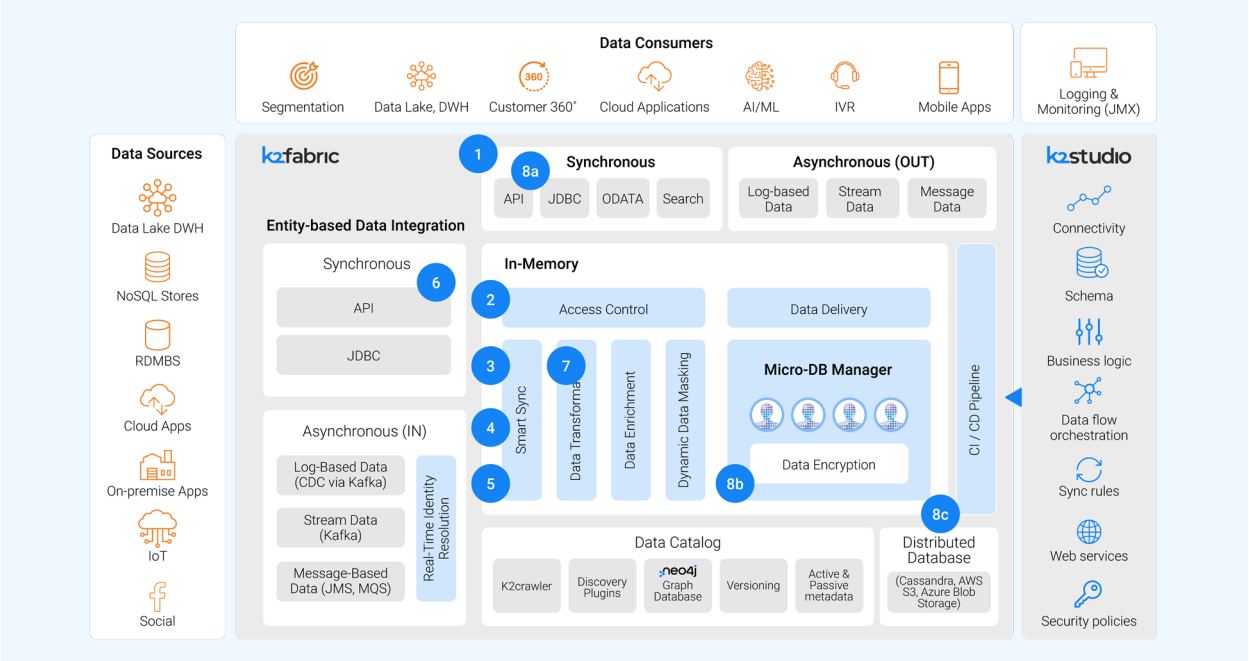
  When a change occurs in a source system, K2view changes only the corresponding data elements in the Micro Database. This function eliminates redundant updates, for optimal performance.

- **AlwaySync**
  This mode allows complete control over the data that needs to be synchronized with source systems

  Using AlwaySync, K2view allows users to configure what data needs to be refreshed automatically, and how frequently. For each element of the data product schema, an AlwaySync timer (that drives K2view synchronization) is set.

  Once configured, K2view synchronizes data with source systems only on data access, to optimize performance. As such, if the timestamp on the retrieved Micro-Database data is older than the timer interval set for this data, K2view triggers synchronization (for that data element only).

AlwaysSyncflow through Data Product Platform



1. The web service (synchronous) API layer relays a data access request from a user application.
2. The user is authenticated and authorized to proceed with data access.
3. Smart Sync retrieves the data from the distributed database storage into memory (if not already there).
4. Smart Sync checks the Micro-Database data timestamp against the AlwaySync timer and discovers that the timestamp is older than the AlwaySync timer.
5. Smart Sync triggers the Data Integration layer for data ingestion.
6. The integration layer synchronizes the data with the source systems and then sends it to the transformation engine.
7. The data transformation engine uses the data to perform necessary operations.
8. Data is delivered to the data consumer by performing the following actions in parallel: data delivery layer sends processed data to the user application. In parallel:
   a. The data delivery layer sends processed data to the user application
   b. The processing engine sends the data for encryption.
   c. The data is compressed and sent to a distributed database for storage.

Micro-Database Resolution in Real Time

K2view Data Product Platform retrieves fresh data from the source systems according to the data sync policies and resolution of the update – from a single field, up to an entire table.

In a data streaming, CDC, or messaging data ingestion modes, the platform must correlate – in real time – data updates to the relevant Micro-Database. To achieve this, the platform provides an elegant, fully integrated solution to ensure that an incoming message from a source system is propagated in real-time and associated with the correct Micro-Database.

The platform receives changes such as inserts/delete/updates while they are performed on the source systems, applies advanced heuristics to determine the instance ID of the Micro-Database impacted by the change, and updates accordingly the corresponding Micro-Database.

For example, in cases where the data changes occurring in the sources do not match the current known parent/child relationship(s) of the data product's data model, the platform will recursively search the previous relationship(s) in the data model hierarchy, identify the newly modified ones, and automatically update the data product instance accordingly.

6. Change Management

Any time a Micro-Database is retrieved from the K2view platform, the Micro-Database Manager component evaluates its current schema against the data product's schema and sync policies, and updates the Micro-Database with fresh source data accordingly.

Consider a scenario in which a change is introduced to the data product schema (e.g., a new table or a new field is added). When the Smart Sync is triggered, it compares the current schema with that of the deployed data product instance, identifies the schema change, and then invokes K2view data integration to retrieve data from the source systems and update the Micro-Database (per the new data schema). Only that data product instance (and Micro-Database) is affected.

As such, K2view makes the process of updating data schemas seamless and transparent to data consumers:

- No downtime is required.
- No changes are required to the consuming applications.
- Data product Micro-Databases are immediately accessible, even after their schema has changed (without going through a complex, costly, and time-consuming migration processes).
- A "time machine" enables data consumers to retrieve Micro-Database data as it evolves through the different schema versions.

CHAPTER 4

Data-as-a-Service Automation

K2view Data Product Platform provides easy access to Micro-Databases via web services – for seamless application integration.

The K2view platform features:

- A query engine supporting full SQL
- An easy-to-configure web service generation layer

Querying Micro-Databases

Data Product Platform uses two query methods, depending on the type of data being queried:

- ANSI SQL, for an operational query on single Micro-Database
- Elasticsearch APIs, for an analytics query spanning multiple Micro-Databases

On top of the standard indexing functionalities provided by full SQL support, K2view provides a unique way to define and utilize indexes, to optimize queries, and enable user access control.

Indexes can be defined for any field of the K2view data product schema. Once a field is defined as an index, it can automatically be used for analytical search queries using Elasticsearch. Indexes are stored as reference data, and can be used to optimize queries, and to define access permissions.

For instance, K2view enables a DBA to define the state field of an address table (contained in the schema of a particular Micro-Database) as an index, thus enhancing the performances of queries selecting all Micro-Databases of a specific state.

Web services

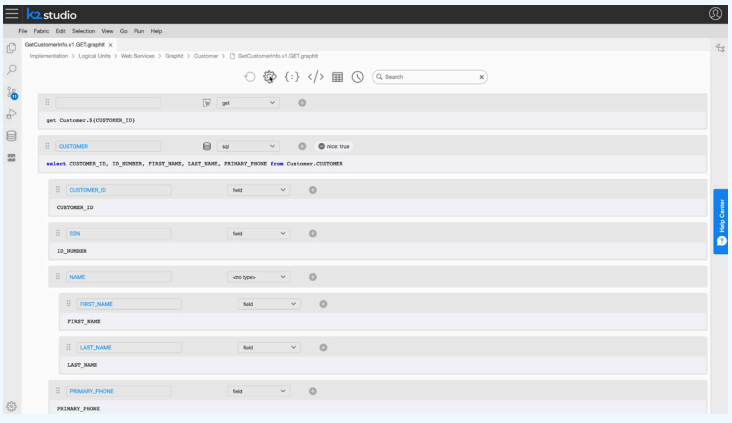
A core feature of K2view Data Product Platform is an embedded web services layer.
Exposing data via web services involves advanced software development. Developers must define:

- Communication protocols with the database management system, and then expose related access methods
- User and security protocols
- Distribution of the web services layer
- Business logic

Such development requires resources, time, and constant maintenance to address database changes, or new functional requests.

K2view Data Product Platform includes a visual low-code/no-code framework to easily and quickly generate web services. Any function (even a query) can be created, and registered, as a web service.

These functions can then be reused and combined with other functions. Simply put, any web service can be easily reused by other web services.



No-code web service generator

Once a web service function is defined, K2view automatically takes care of user access, distribution, updates due to schema changes, and more – saving time and effort.

Each web service can be restricted to a particular Micro-Database, and/or user. Indexes can be used to restrict access to a web service, potentially restricting any field of the data product schema.

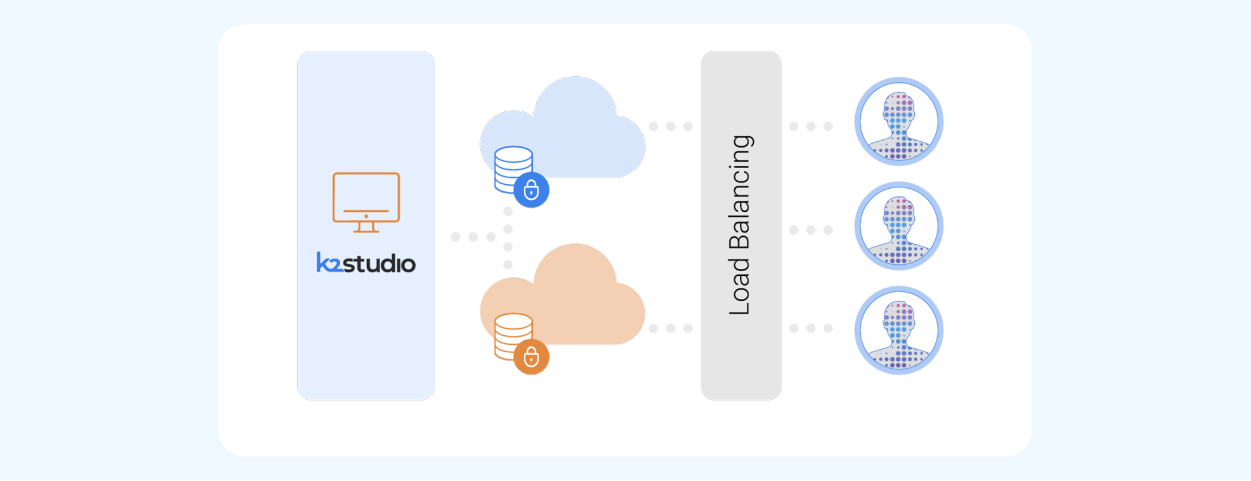
CHAPTER 5

Deployment Modes

K2view Data Product Platform can be deployed in multiple configurations to match an organization's IT and business needs:

- On-premises and private cloud deployment
- Integration Platform as a Service (iPaaS)
- Hybrid

The platform can be configured as a cluster, consisting of multiple nodes, deployed across multiple datacenters, to provide access to any data consumers. This allows for unparalleled flexibility, because the number of nodes can be scaled dynamically to meet demand at any given time, and/or storage capacity constraints.



Multiple-node deployment mode

On-premises and private cloud deployment

The K2view platform can be deployed on premises, installed on a company's local servers.

It can also be deployed to a customer's private cloud (e.g., AWS, GCP, Azure, Oracle) with dedicated servers, resources, and bandwidth provisioned on public cloud infrastructures. In a private cloud deployment, the platform services are consumed for a monthly service fee that varies as a function of the number of nodes and/or clusters required and based on the volume of data processed.

K2view Cloud: Integration Platform as a Service (iPaaS)

The K2view platform can be deployed as a service in a public cloud architecture. K2view iPaaS includes the required cloud hardware, cloud software services, and cloud storage to support the operations of the K2view platform.

Using K2view iPaaS, enterprises can:

- **Simplify integration**
  Application and data integration is seamless, without any delays.
- **Accelerate integration initiatives**
  Integrations are available quickly, to speed your digital transformation initiatives.

- **Spend more time on higher value integration activities**
  There's no longer any need for connectivity code. Built-in connectors, and no-code data orchestration tools, integrate with all application and data sources.
- **Expand integration capacity**
  Integration flows can be used by both business and data teams, to expand the scope of integration.
- **Change integrations rapidly**
  Integration flows are quickly adapted to meet business requirements and digital transformation efforts.

Hybrid deployments

Hybrid deployments are required when enterprise data is spread across multiple locations, system types, and infrastructures.

For example, data sources and targets could be organized according to a mix of cloud-to-cloud, cloud-to-premise, and premise-to-premise configurations, using multiple data processing scenarios and data patterns.

The K2view platform is designed to aggregate data from multiple sources, residing on different systems. Its multi-node configuration capabilities enable all servers to be integrated, regardless of whether each node is running on a virtual machine located on an edge location, or on a physical server situated on premises.

This functionality means that:

- Data products are created and managed closer to their data sources, ensuring that specific data and its metadata are managed close to their source.
- The processing node will process data that is geographically relevant, thus reducing network load and latency.
- Data consumers, running applications either on-premises or in the cloud, will always have access to the local node, thus avoiding scope trespassing.
- Compute resources can be moved between on-prem and cloud, to adjust to processing needs.
- Data engineers and architects can choose the best environment for their projects, and deploy accordingly.

CHAPTER 6

Resilience

Consistency, Durability, and High Availability

K2view Data Product Platform ensures the consistency, durability, and availability of the data it manages.

Consistency

Consistency is ensured by the processing engine. Every time a write into a certain Micro-Database occurs, K2view checks against a transaction table stored in the distributed database, determines if a concurrent transaction is occurring, and if the write should be put on hold. For instance, if 2 or more concurrent transactions are committed to the same Micro-Database, its transaction log is used as a conflict detection/resolution mechanism.

Durability

Durability is inherent to the distributed database layer. K2view provides durability by first appending writes to a commitlog. This means that before performing any write, and using it in memory, it appends the value into a commitlog. This not only ensures full durability – because data is written on disk first – but since it's appending only a small piece of a file, it is basically instantaneous.

High Availability

High availability is maintained by the underlying distributed database architecture, which eliminates single points of failure, by design. With Cassandra, for example, all nodes are equal – without masters or coordinators at the cluster level. It provides for reliable crossover and detection of failures.

Performance

K2view's superior performance is rooted in its patented Micro-Database technology – more specifically, in its ability to run every query on a relatively small amount of data. This makes K2view Data Product Platform the market's fastest, high-scale operational data management system.

Performance is also optimized based on the following principles:

- Every query is executed in memory.
- For search queries that span several Micro-Databases, an integrated Elasticsearch engine is used to scan and search data in multiple Micro-Databases.

In-memory processing

In-memory processing enhances performance because queries are executed in memory, and not on disk. While some queries need to be distributed (e.g., analytic queries), as a rule, K2view does not require complex parallelization for any of its operations.

If a data product schema is correctly designed, most queries are executed against one Micro-Database, on a limited amount of data. Therefore, the amount of data to be retrieved and processed in memory is small enough to provide extremely fast performance – without complex cross-node distribution.

Elasticsearch

Elasticsearch is a distributed, RESTful search and analytics engine, capable of addressing many different use cases. It stores data centrally for high-speed search, fine-tuned relevancy, and powerful analytics.

The K2view platform has native integration with Elasticsearch. It allows users to perform and combine structured and unstructured searches across multiple Micro-Databases, while its aggregations let them zoom out to discover trends and patterns in their data.

CHAPTER 7

Security

A major hurdle for any data management system is ensuring that the data is securely stored. K2view's Micro-Database approach secures data via:

1. **Advanced data encryption**, thanks to the company's patented Hierarchical Encryption Key Schema (HEKS)
2. **User access control**, combining HEKS, data services authorization parameters, and index definitions

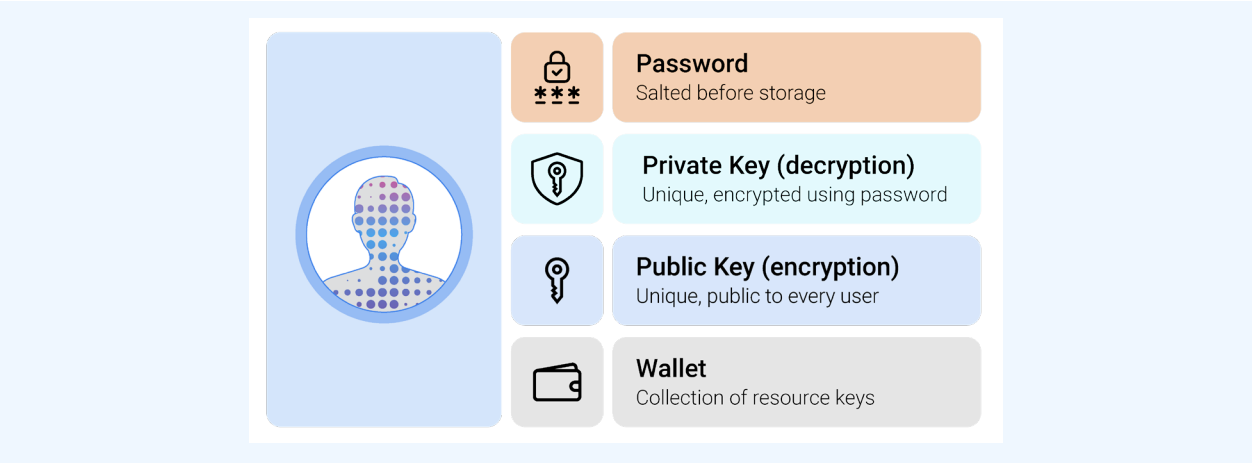
These protocols are used in the authentication and encryption engines. The authentication engine ensures user access control upstream, while the encryption engine encrypts the data downstream, prior to storage.

Encryption engine

Each user of K2view Data Product Platform is created with a set of security attributes, listed in the figure below.

K2view uses an industry-standard, public-key cryptography algorithm to encrypt and decrypt data. The public key encrypts a user's wallet, which stores all resource keys. Data can only be read using a user's password-based private key. The user password, which is salted before being stored in the K2view platform, is never stored anywhere else.

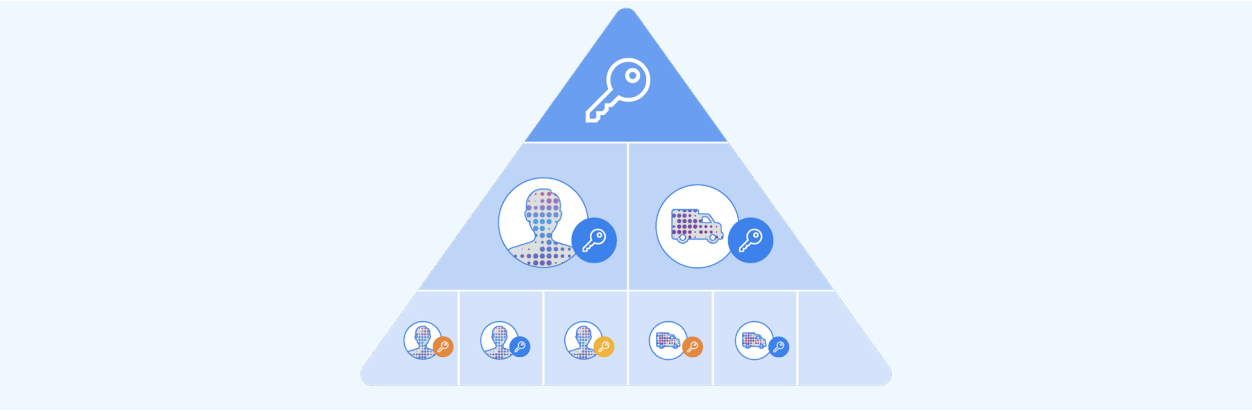
The cryptography grants users access to the resource keys stored in their wallets, enabling certain patented security features.



Security attributes granted per data consumer

K2view's patented algorithm, Hierarchical Encryption-Key Schema (HEKS), relies on the resource keys contained in a user's wallet:

- **Master key**
  The master key, which is automatically generated during K2view Data Product Platform installation, permits access to every resource of the K2view platform. Any user with access to that key, can generate all other key types – and thus encrypt, or decrypt, any resource within the hierarchy.
- **Type keys**
  Type keys restrict access at the data product level (e.g. customers, suppliers, devices, orders), and are a hash of the master key. As such, each user that has access to a type key, can encrypt, or decrypt, data belonging to every type of data product.
- **Instance keys**
  Instance keys restrict access at the data product instance (Micro-Database) level (e.g. specific customer, or specific supplier), and are a hash of their corresponding type keys. As such, a user with access to one data product instance, will not be able to decrypt data from any other instance, even when both instances are part of the same schema.



K2view's patented hierarchical encryption-key schema

The pyramid above shows how HEKS is implemented for data product types, using the following keys:

- 1 master key, allowing full access
- 2 type keys, restricting access to 2 different data product types
- 6 instance keys, 3 for each data product type, restricting access at the Micro-Database level

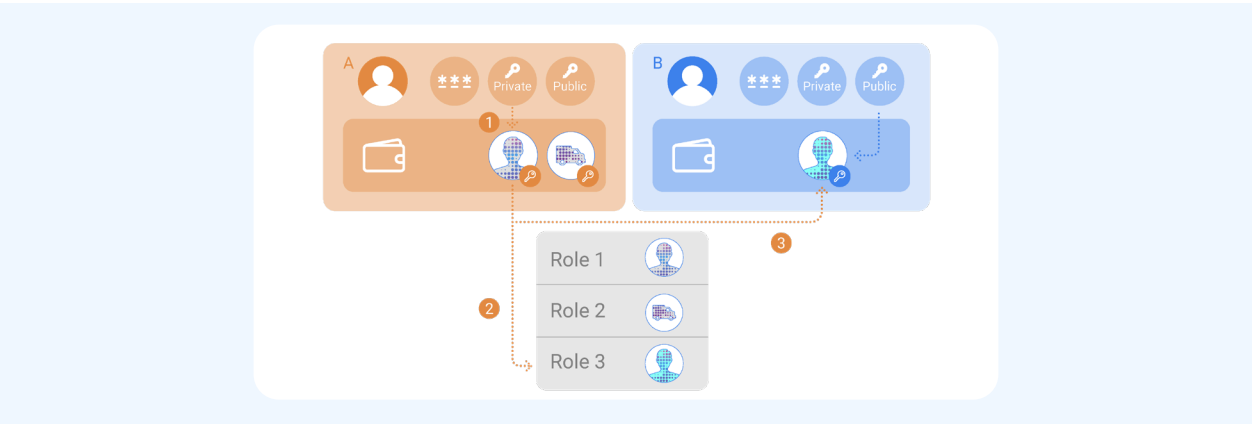
This kind of hierarchical encryption enables complete control over all stored data, and significantly reduces the risk of data breaches.

For example, if 1 Micro-Database key were to be hacked, only the data contained in that single Micro-Database would be breached. All other Micro-Databases would remain safely encrypted. This security feature makes the K2view platform ultra-safe, and makes massive data breaches impossible.

HEKS user access control

Once defined, HEKS resource keys are used for user access control, thus associating roles to resource keys. In the following example, user A is added to a role, and given access to 2 data product types (with corresponding type keys), while user B has no type key:

1. User A uses its private key to decrypt the instance key needed for the grant.
2. Using the instance key, the platform allows the user to create a new role, allowing access to 1 Micro-Database of that data product type.
3. Using user B's public key, user A encrypts the generated instance key (hash of the data product type key), and then adds it to user's B wallet.



HEKS-based user access control

User access control beyond HEKS

Not only does the platform grant access via role, it can also restrict access via role definition:

- At the data product/Micro-Database level, enabling read or write over its structure
- At any other level, defining the method used to access the data (e.g., web services function)

Thus, not only can user access be restricted at the Micro-Database level, it can also be restricted to a single method (e.g., 1 web services reading method).

Administrators can also restrict access, based on indexes defined for each element of the data within the product data schema.

These definitions index cross Micro-Database queries, while allowing for complete granularity in user access control. For example, a country index can be defined, and then used to define a new role that grants access to every Micro-Database from a specific country to a user subset.

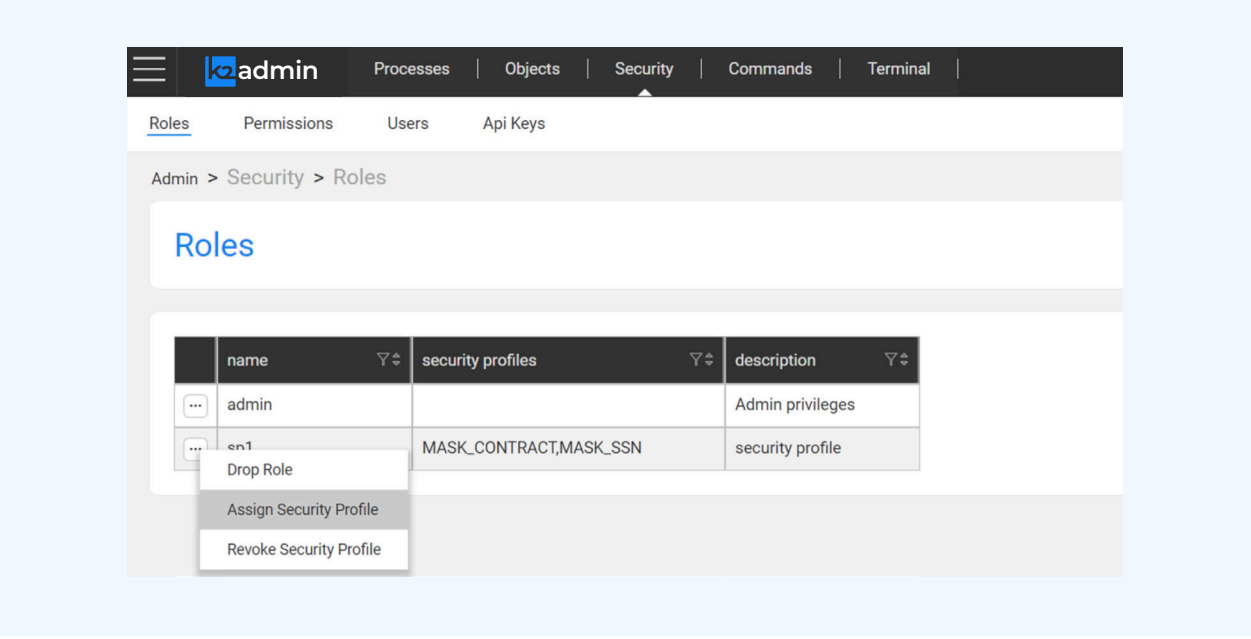
K2view also supports the following access control mechanisms:

- RBAC (Role Based Access Control) - schemes are fully implemented along with Auth&Auth capabilities to ensure K2view environment and data are accessed on a pre-defined, pre-approved per-permission basis.
- ABAC (Attribute based access Control) - masking is applied based on the security profile of the data user and also based on additional attributes, such as the user's location, device, type of access requested and more.

K2fabric also provides user identification and access management (IAM) for web, console, and web-services access, either using K2fabric's local repository or by using the organization's identify provider (IDP) which K2fabric integrated with as Service Provider (SP).

In addition, Data Product Platform supports integration with Secret Manager services, so that secrets - like passwords, used in interfaces that enable communication to external systems - will not be stored in the platform itself.

Data Product Platform supports integration with various external Secret Management providers, in which case it doesn't store the secrets but rather their reference IDs.



Role-based access control (RBAC)

CHAPTER 8

Solutions

K2view provides enterprise solutions that leverage the underlying Data Product Platform, including:

1. **Customer 360**

   K2view Customer 360 is based on an all-in-one customer data hub designed to turn fragmented data into a trusted customer view that's instantly accessible to authorized data consumers. A single point of access to customer data, with built-in Role-Based Access Control (RBAC), the solution requires minimal commodity hardware, on-prem or on-cloud.

   K2view Customer 360 features the following capabilities:

   - Complete unified data, including interactions, transactions, and master data
   - Real-time integration, of structured and unstructured data
   - Data orchestration, to unify and deliver trusted customer data services
   - Golden record, of matched, cleansed, and normalized customer data
   - Data governance, including dynamic data masking and consent management
   - Democratized access, to authorized consuming apps
   - Data services, for quick auto-generation, debugging, and deployment
   - Data encryption, at the customer record level, for maximum security

2. **Test Data Management**

   K2view Test Data Management ingests source data from all systems in a higher environment and organizes it by business entities, while compressing and anonymizing the data into a test data store.

   Data teams can effortlessly perform any operation on the entity data in the test data store, before provisioning it to target environments, on demand.

   With entity-based test data management, testers can:

   - **Provision** structured and unstructured test data from any sources to any targets.
   - **Deliver** multi-source test data subsets in minutes, based on user-defined criteria.
   - **De-identify** PII while ensuring referential integrity.
   - **Generate**, prepare, and deliver realistic and accurate synthetic test data.
   - **Reserve** test data subsets to prevent testers from overriding each other's data.
   - **Version** and roll back test data subsets, to rerun tests with the same data.
   - **Manage** user access, with a multi-layer security portal.
   - **Integrate** test data into your DevOps CI/CD pipelines with web service APIs.

3. **Data Masking**

   K2view Data Masking anonymizes sensitive data at enterprise scale for compliant software testing, data analytics, and B2B data sharing. They feature PII discovery and classification, static and dynamic masking, hundreds of masking functions, as well as support for structured and unstructured data, and all data sources.

   Entity-based data masking provides enterprises with:

   - **Better data protection**, achieved with in-flight and in-context data masking.
   - **Greater productivity**, enabled by no-code tooling and prebuilt masking functions.
   - **Full automation**, with data masking integrated into CI/CD pipelines.
   - **Quick and easy rollout,** implemented in weeks, instead of months.
   - **Flexible deployment,** available in on-prem, cloud, or hybrid environments

- **Future-proof,** by integrating with all data sources.
- **Scalable,** optimized for high-performance at any data volume.
- **Lower cost,** by eliminating the need for per-database software licenses.

4. Synthetic Data Generation

K2view Synthetic Data Generation is a one-stop solution for preparing, generating, and managing compliant, realistic, and accurate data for software testing and ML model training. It generates synthetic data by business entities and ensures that all the required data for each business entity is consistent and contextually accurate.

With k2view entity-based synthetic data generation solution, enterprises benefit from:

- **Generative AI**, for augmentation when data is scarce.
- **Rule-based generation**, for functionality and negative testing.
- **Entity cloning**, for performance and load testing.
- **Data masking**, for all data protection use cases.
- **Auto-discovery**, of all data types, sources, and formats.
- **Entity modeling**, for a blueprint of the data to be generated.
- **Data subsetting**, directly from the source systems.
- **Auto-delivery**, directly into target systems.

5. Data Tokenization

K2view brings a business entity approach to data tokenization, in which sensitive data and corresponding tokens are securely stored and managed as separate entities in a Micro-Database. Each Micro-Database serves as an individually encrypted and compressed "micro" token vault.

K2view Data Tokenization offers enterprises:

- **Better data protection**, due to multiple levels of security and encryption.
- **DevOps-ready solution**, where APIs integrate with CI/CD pipelines.
- **Quick and easy rollout**, with software-only implementation in weeks.
- **Flexible deployment**, in on-prem, cloud, or hybrid environments.
- **No-code intuitive GUI**, for more productivity and less burden on IT.
- **Future-proof solution**, designed to integrate with any and all data sources.
- **Scalable architecture**, supporting an unlimited number of data sources.
- **Support for any data**, in any format, language, or technology.

6. Data Pipelining

The K2view data pipeline solution supports massive-scale, on-prem, cloud, and hybrid deployments – ingesting data from any source system and delivering it to any data lake or data warehouse.

A data product approach to ETL/ELT ensures high-performance data pipelines, data integrity, and business agility. Data engineers can prepare and deliver fresh, trusted data – from all sources to all targets – at enterprise speed and scale.

With K2view Data Pipelining, data teams can:

- **Collect**, process, and serve data by business entity.
- **Ingest** and unify data from all sources while ensuring data integrity.
- **Discover** and visualize data lineage with a built-in data catalog.
- **Transform**, clean, enrich and mask data via reusable functions.
- **Encrypt** data from source until it is served to the data lake.
- **Automate** and productize data preparation flows.
- **Deliver** data to lakes and DWHs in real time, via a schedule, or on demand.
- **Deploy** the solution in hybrid and multi-cloud environments.

CHAPTER 9

Administration

The K2view platform is configured, monitored, and administered using the:

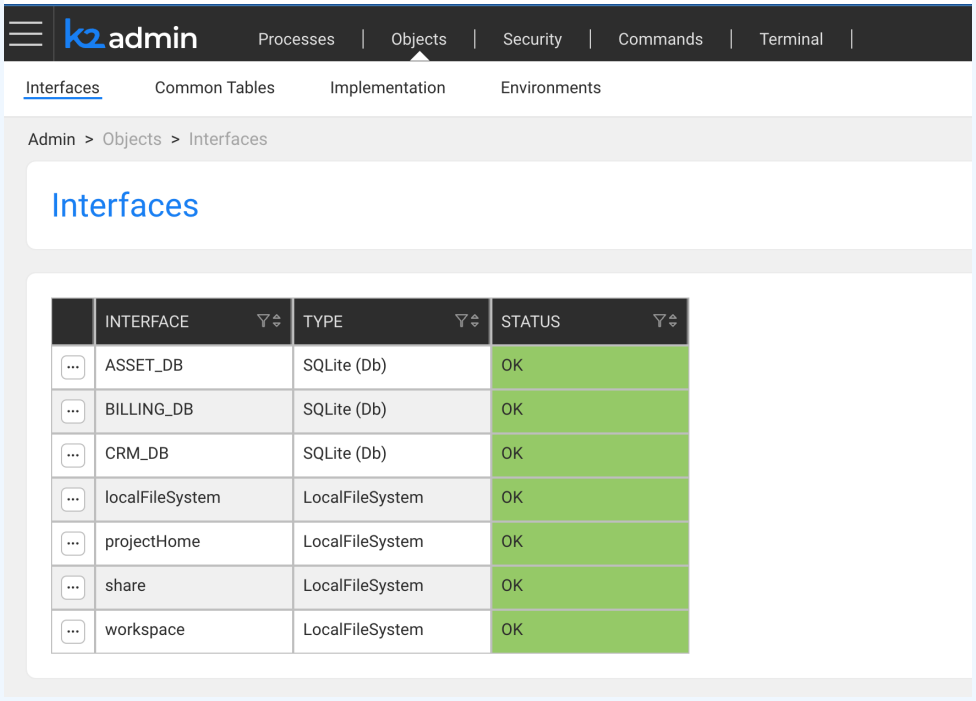
- K2view Admin Manager, and K2view Admin
- Administration capabilities native to distributed databases

K2view Admin Manager

- Version control, defining repositories and developer access
- Password protection, where access can be granular, to any level of the configuration
- Services control, managing start/stop of K2view services

K2view Admin

- Query execution
- Index definition
- User management
- Role and permission management
- Nodes administration



Admin console

CHAPTER 10

Hardware Requirements

K2view Data Product Platform does not require storage of all data in memory, or expensive hardware for scaling performance. Its low total cost of ownership (TCO) is based on 3 concepts:

- In-memory performance, on commodity hardware
- Complete linear scalability
- Risk-free integration

Minimum requirements

The K2view platform is installed on a:

- Linux server, for server node management
- Windows server, for admin/configuration tool operation

The minimum requirements for the Linux cluster are:

- CentOS 6.5 OS, or Redhat 6.5 with latest patches
- Modern Xeon Processor
- 4 nodes x 4 cores
- 32GB RAM
- HDD, either:
    a. 2 Physical HDDs (500GB each) in a RAID0 configuration
    b. 2 500GB SSDs

Note that K2view stores its data on disk, while performing data operations in memory. This makes RAM requirements minimal.

The minimum requirements for the Windows server are:

- Windows Server 2008 r2 64bit Machine, or
- Windows 7 64Bit or Windows 8 64Bit1 CPU
- 4GB RAM
- 100GB available disk space

Linear scalability

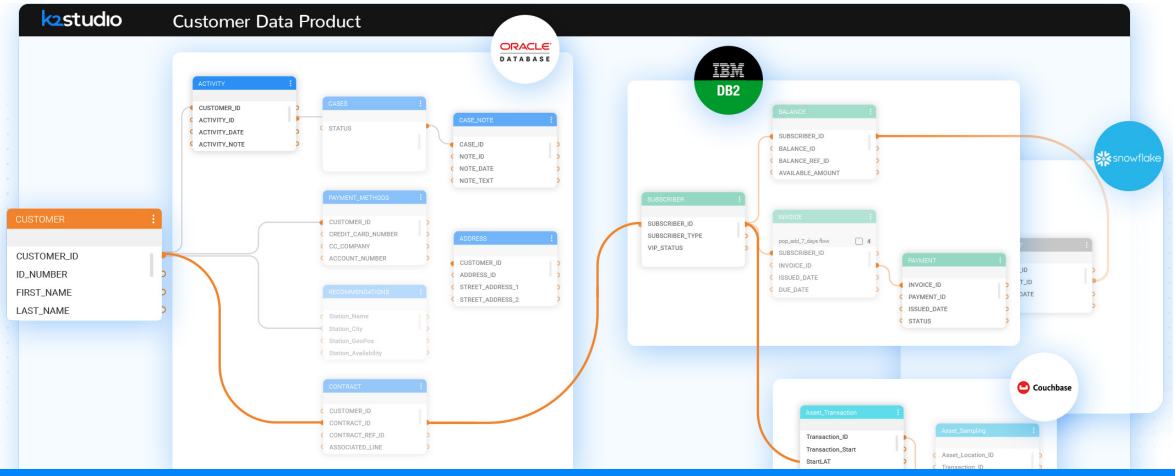
Linear scalability is assured by the underlying distributed database (e.g., Cassandra).

Low-risk automation

Beyond the costs associated with the purchase of a new platform (hardware, software licenses), a major cost component of any data management system is its integration into an existing IT ecosystem, and the risks associated with provisioning data safely to authorized data consumers.

K2view Data Product Platform minimizes integration costs, due to:

- Automated data ingestion, with no impact on source systems
- Flexible synchronization, allowing for incremental legacy systems retirement
- SQL support, with no technical expertise necessary
- Web service generation, automating access to integrated data
- Integration with a customer's data and analytics technology stack



About K2view

At K2view, we believe that every enterprise should be able to use its data to be as disruptive and agile as Google, Amazon, and Netflix.

We make this possible by transforming all your data – wherever it is – into business-driven data products, which are defined and managed by business domains.

Data products could be customers, products, suppliers, orders – or anything else that's important to your business. We manage every individual data product in its own secure Micro-Database™, continuously in sync with all source systems, and instantly accessible to everyone.

This is all made possible by our data product platform, which delivers a trusted, real-time view of any data product. K2view Data Product Platform deploys in weeks, scales linearly, and adapts to change on the fly. It supports modern data architectures, such as data mesh, data hub, and multi-domain MDM – in on-premise, cloud, or hybrid environments.

This one platform drives many use cases, including application modernization, cloud migration, customer 360, data privacy, data testing, and more – to deliver business outcomes in less than half the time, and at half the cost, of any other alternative.