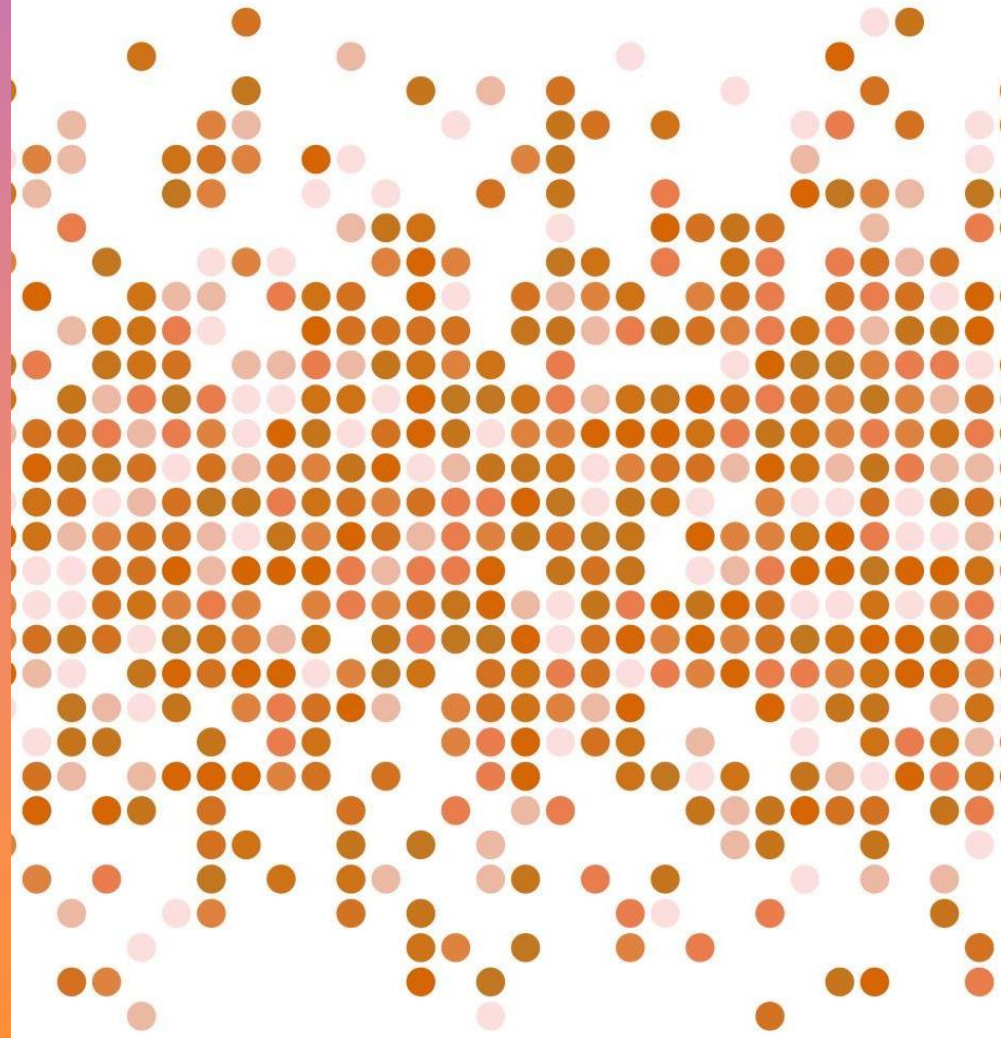
SMALL BUSINESS TELEMETRCIS

Technical Approval Design – Phase 1 Deployment

January 2022



Purpose & Context

Initial pass and insight to the TDA

Purpose

- To give initial pass and insight to the TDA of the work within SBT. This not for final approval. Will come back to the TDA if changes and for final approval.
- The outlined principles have already been discussed with Head of Architecture, Head of Data Science and Head of Ops .

Context

- Phase 1 capabilities
- The phase 1 over laid the reference architecture and is only a sub part of the MVP this time

Phase 1 – Run containers in Kubernetes

The practical case will be for pre-processing journey data

Use Case Summary

The high level goal is to move more to a data meshing and micro service architecture. This won't happen over night, so will need to this is in strategic and tactical phases

So the first phase is to have a pattern and process to expose simple aggregated data for a journey for a particular policy. This is basically building a strangle pattern around the existing MongoDB and then allowing us to change that persistence/source at later date.

Phase Capabilities

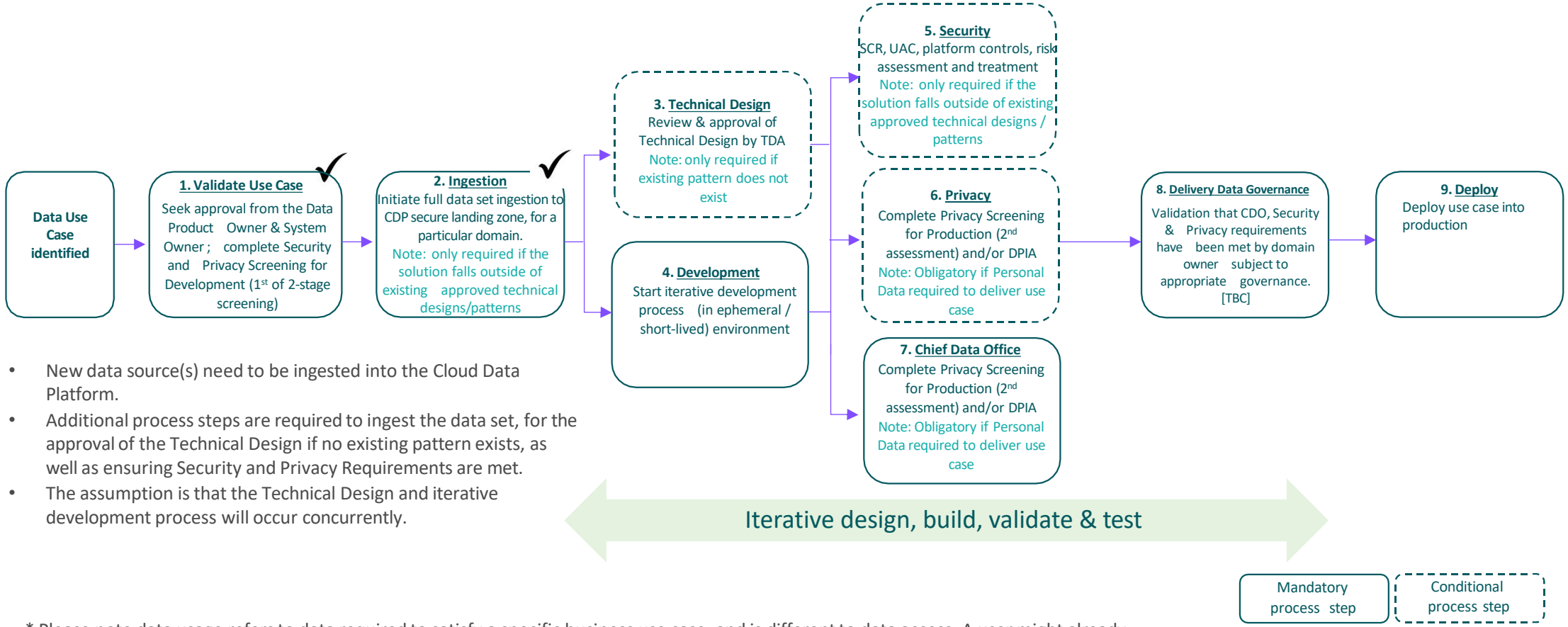
- A pattern/process of deploying and running a docker container in Kubernetes
- It's not restricted to a particular language (Python, .Net Core , Java, Nodejs, etc), CI/CD tool and git repository
- It's not necessarily restricted to AWS
- Look at GitOps continuous delivery tool for Kubernetes, instead of traditional Dev Ops process
- Revisit E2E testing within GitOps

Data Privacy and Security Considerations

- At this stage NO one will have access to the data at this stage.
- Will work closely with security and privacy teams to define an appropriate treatment method to allow for the safe storage and use of former customer data (e.g. in a depersonalised way) if there is any. Though for this phase will try to avoid any sensitive or GDPR data.

Onboarding Data to the Cloud

Data use journey process (Working Progress)

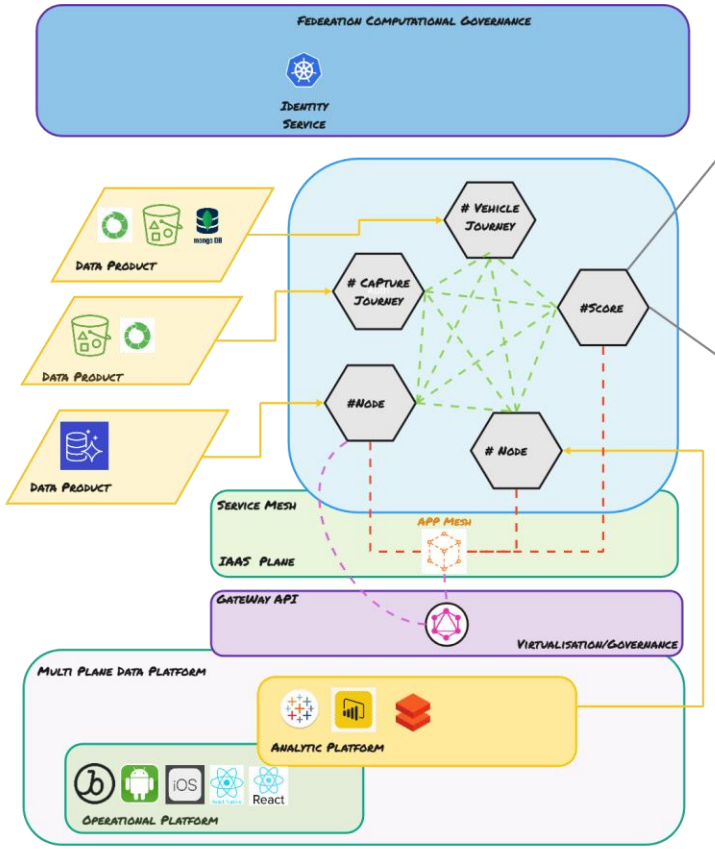


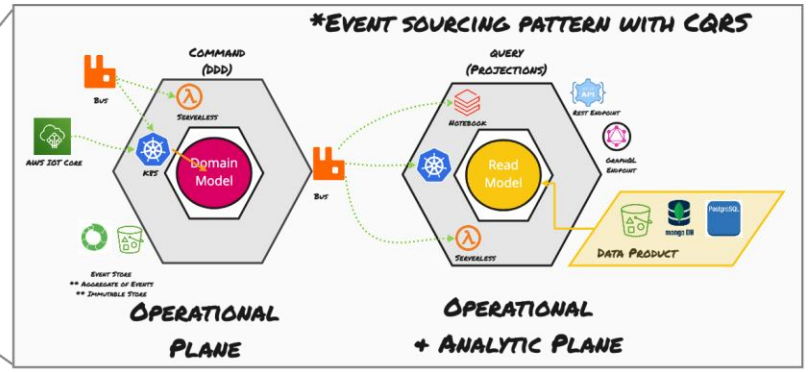
- New data source(s) need to be ingested into the Cloud Data Platform.
- Additional process steps are required to ingest the data set, for the approval of the Technical Design if no existing pattern exists, as well as ensuring Security and Privacy Requirements are met.
- The assumption is that the Technical Design and iterative development process will occur concurrently.

* Please note data usage refers to data required to satisfy a specific business use case; and is different to data access. A user might already have access to data, but not necessarily the approval to use the data for a new or different purpose.

Architecture Holistic Overview

A holistic view of the future architecture





AWS Event sourcing pattern Example

The four data mesh principals:

1. Domain-oriented decentralised data ownership and architecture.
2. Data as a product.
3. Self-serve data infrastructure as a platform.
4. Federated computational governance.

Phase 1 – Architecture for Phase 1

Strangle pattern around the database

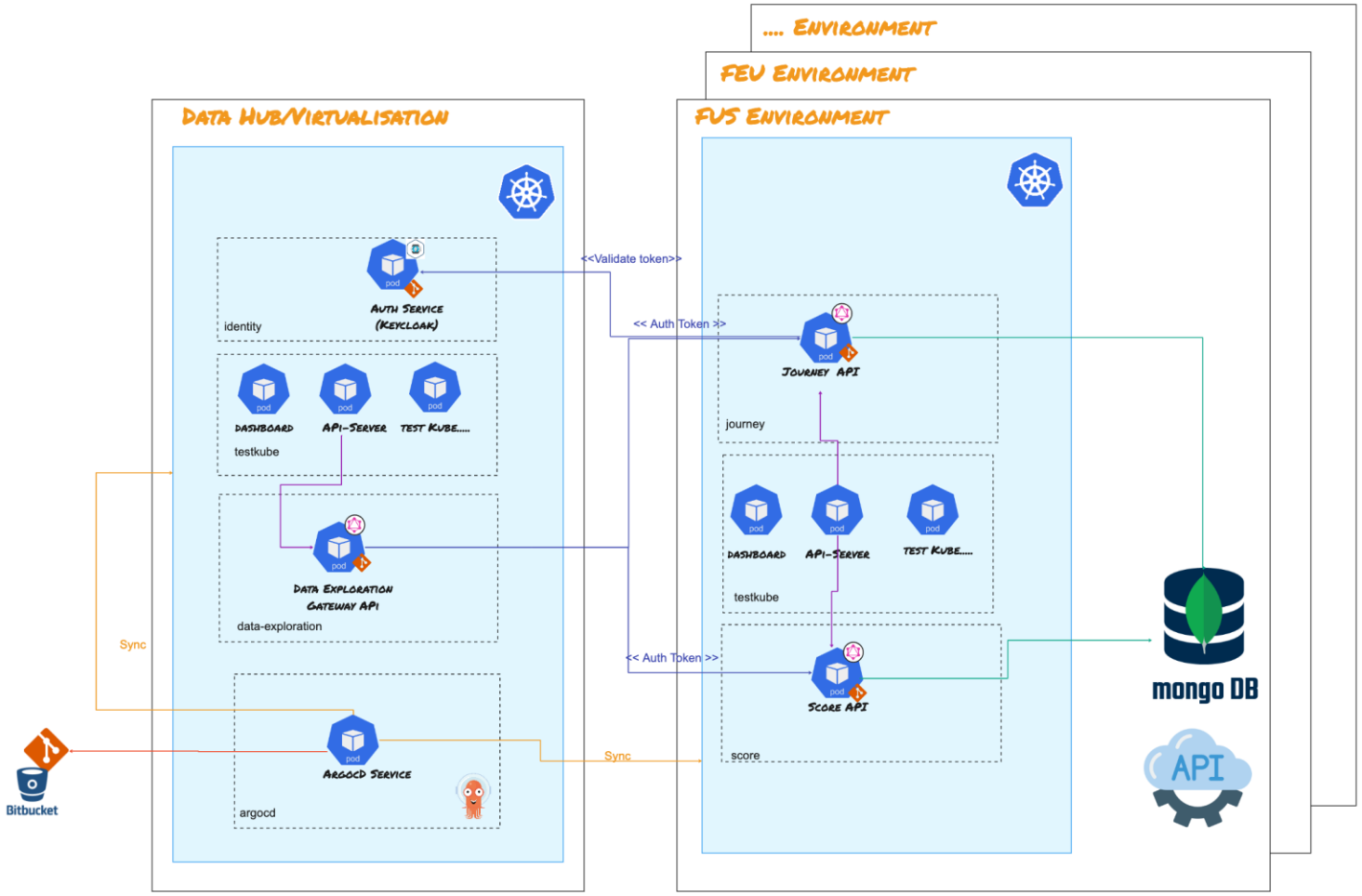
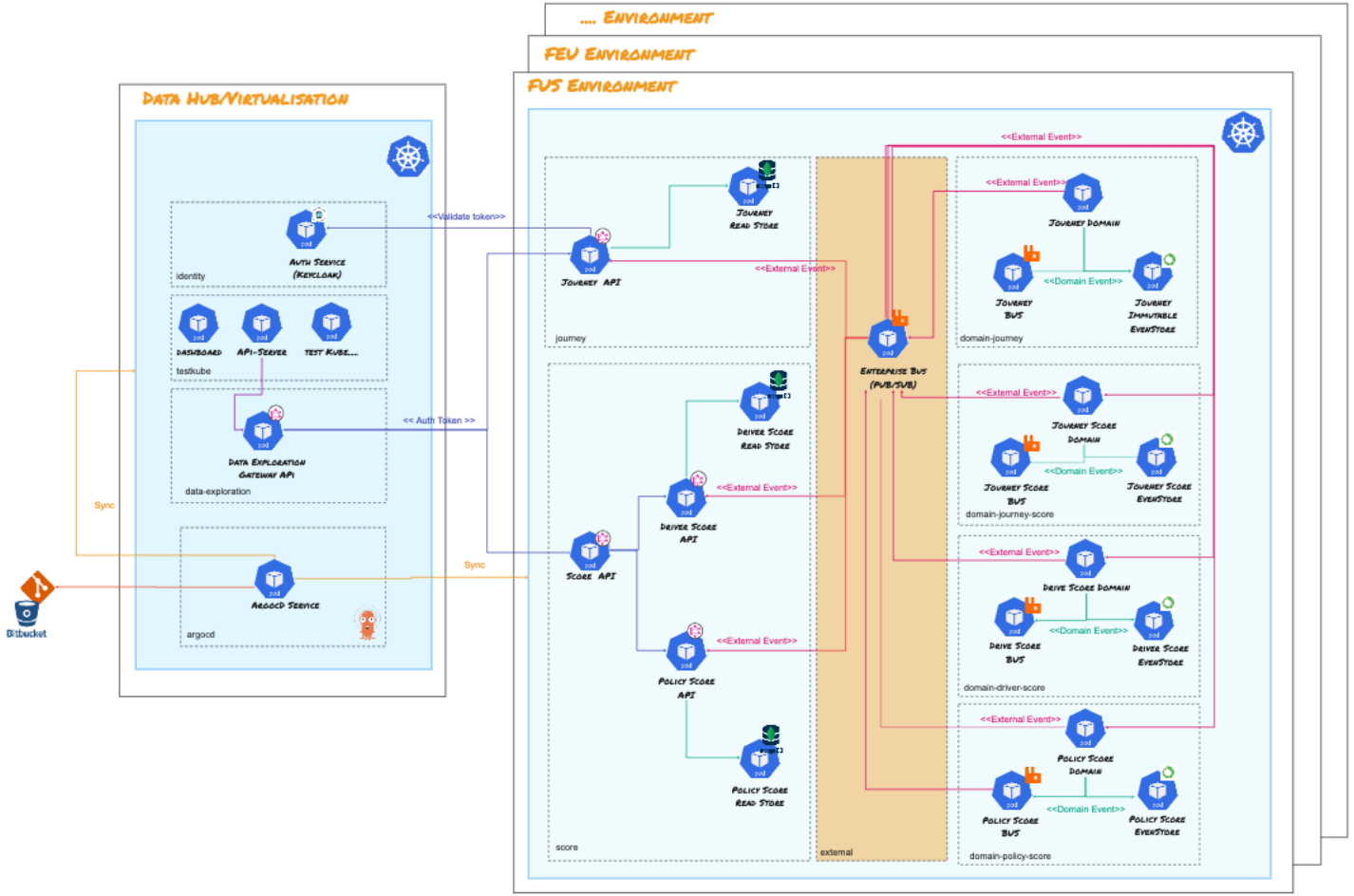


Illustration of Architecture going forward

Moving towards a data meshing and micro service architecture

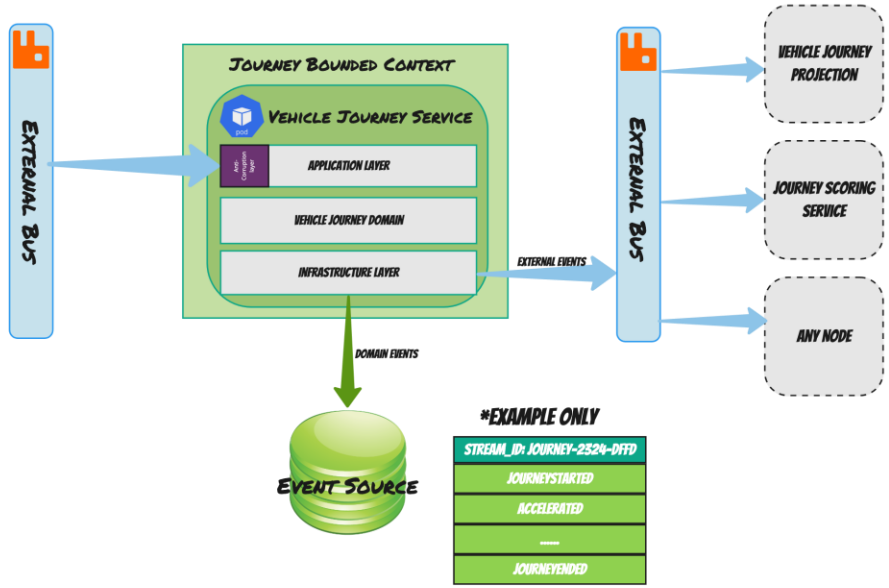


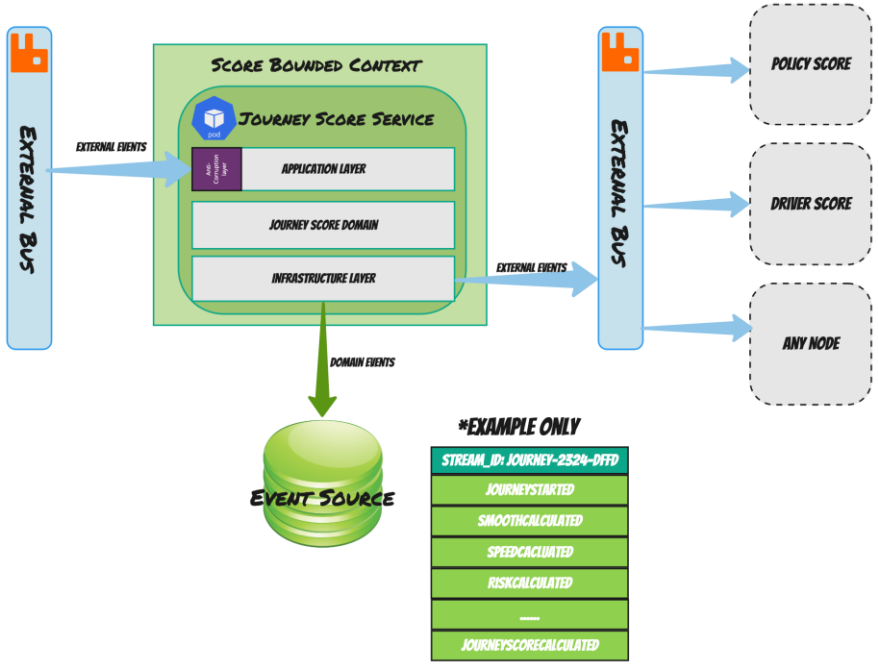
Note: Stack shown is just for illustration purposes (ie Mongo, RabbitMQ, Eventstore)

Note: TestKube can be each env as well

Illustration of Architecture going forward

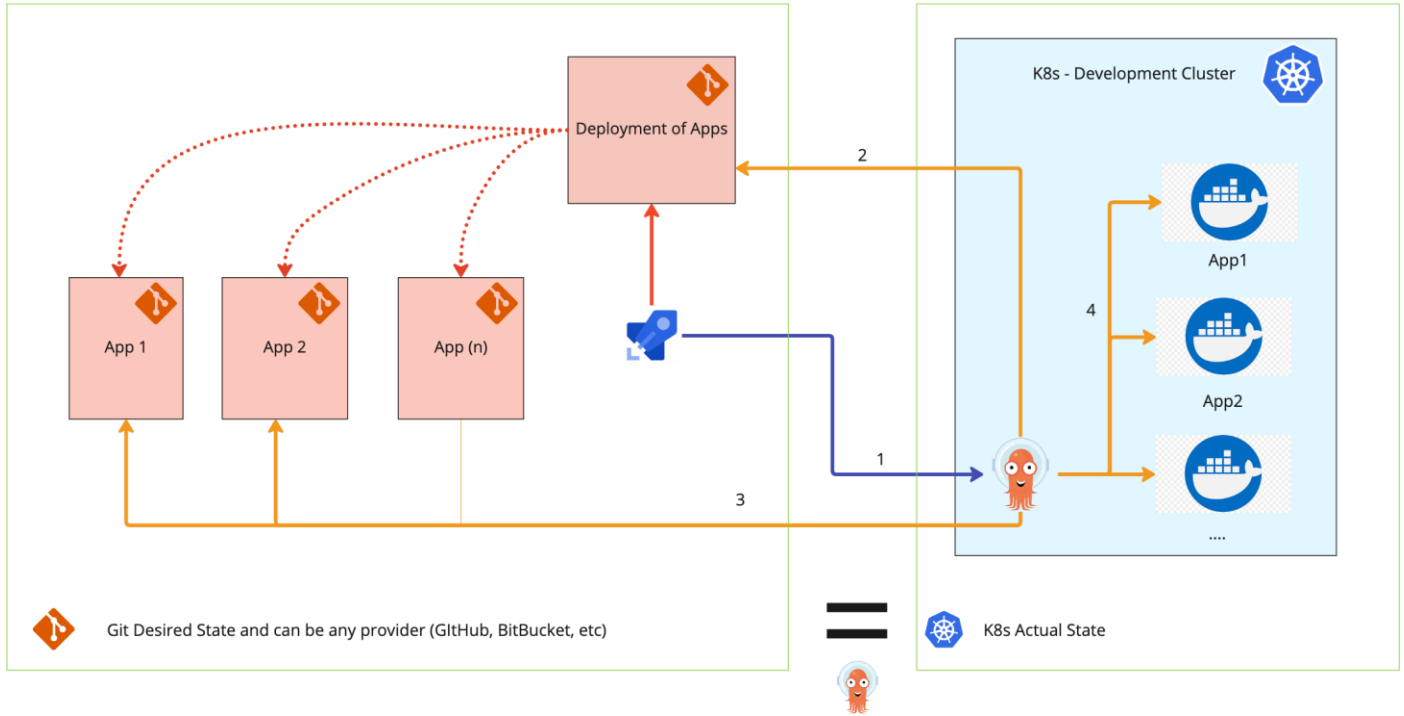
Moving towards a data meshing and micro service architecture





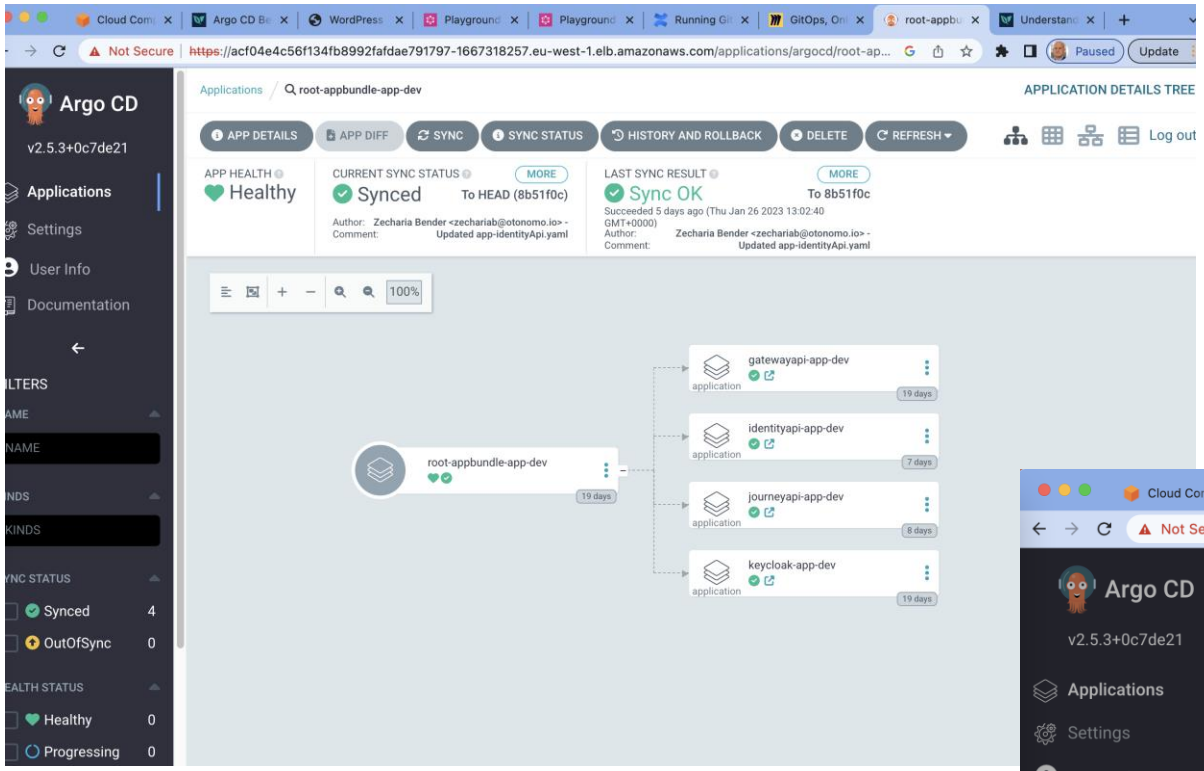
GitOps

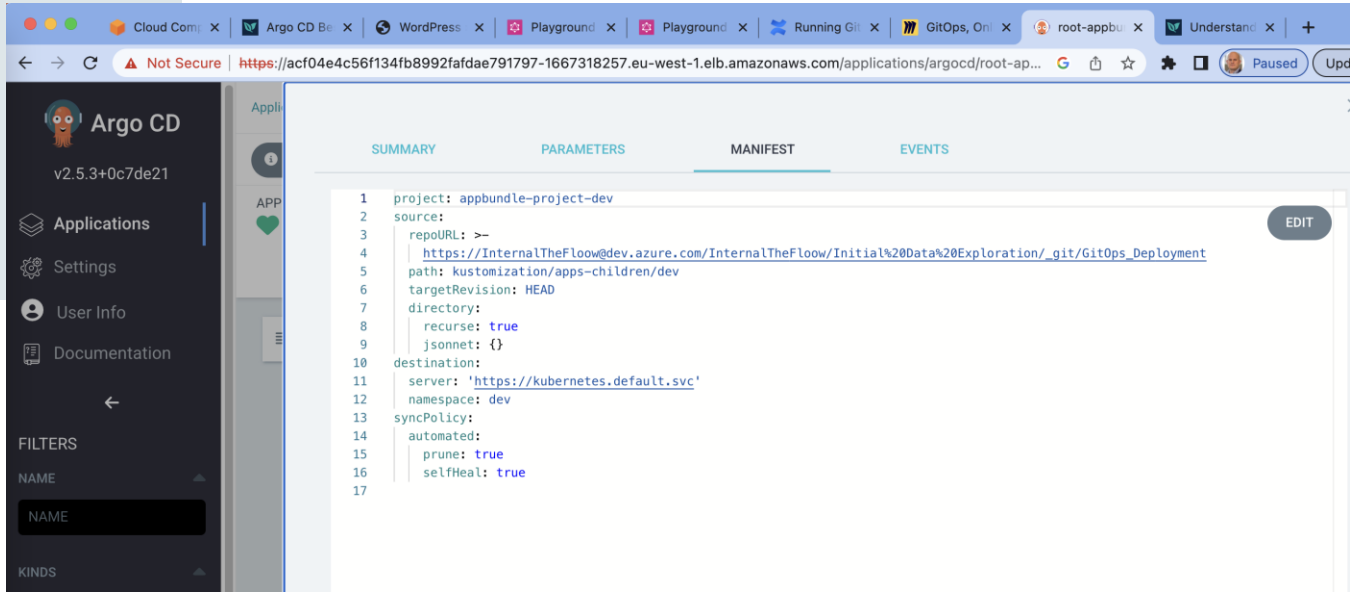
Application of Applications pattern



Argo CD - Example

Application of Applications pattern





Phase 1 – Work in Progress

Completed Capabilities

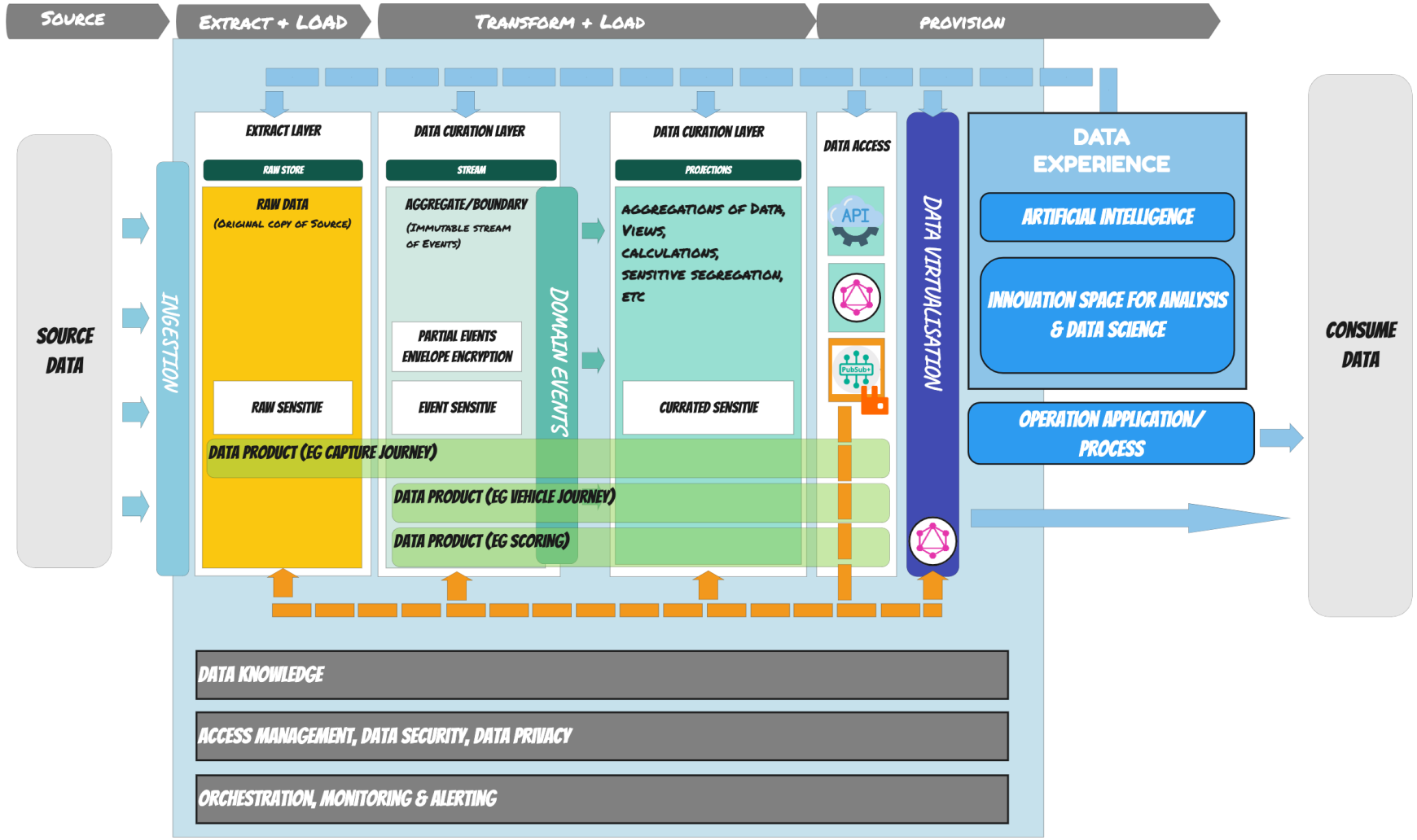
- GitOps continuous deliver for Kubernetes, using Argo CD
- Argo CD - Ability to visualize deployment issues, detect and remediate configuration drift.
- Argo CD – Visualize and monitor Kubernetes cluster and pods, etc
- It's not restricted to a particular language (Python, .Net Core , Java, Nodejs, etc)
- It's not restricted to CI/CD tool and git repository
- It's not restricted to a cloud provider
- Abstracted Security to Central Governance
    - Use Keycloak on K8 cluster - Central Identity Service that can does Authorisation and Authentication

In progress

- Defining a Authorisation process for pattern to use for the Api/Data. Api/UI should respond to capabilities not necessary roles
- Looking at adding other providers like RBAC to our Identity Service
- Testing E2E – Looking at TestKube
- Finalise our Release process using GitOps , as well verifying the Security aspect from release/deployment aspect.

Appendix

Architecture – Future Data Experience



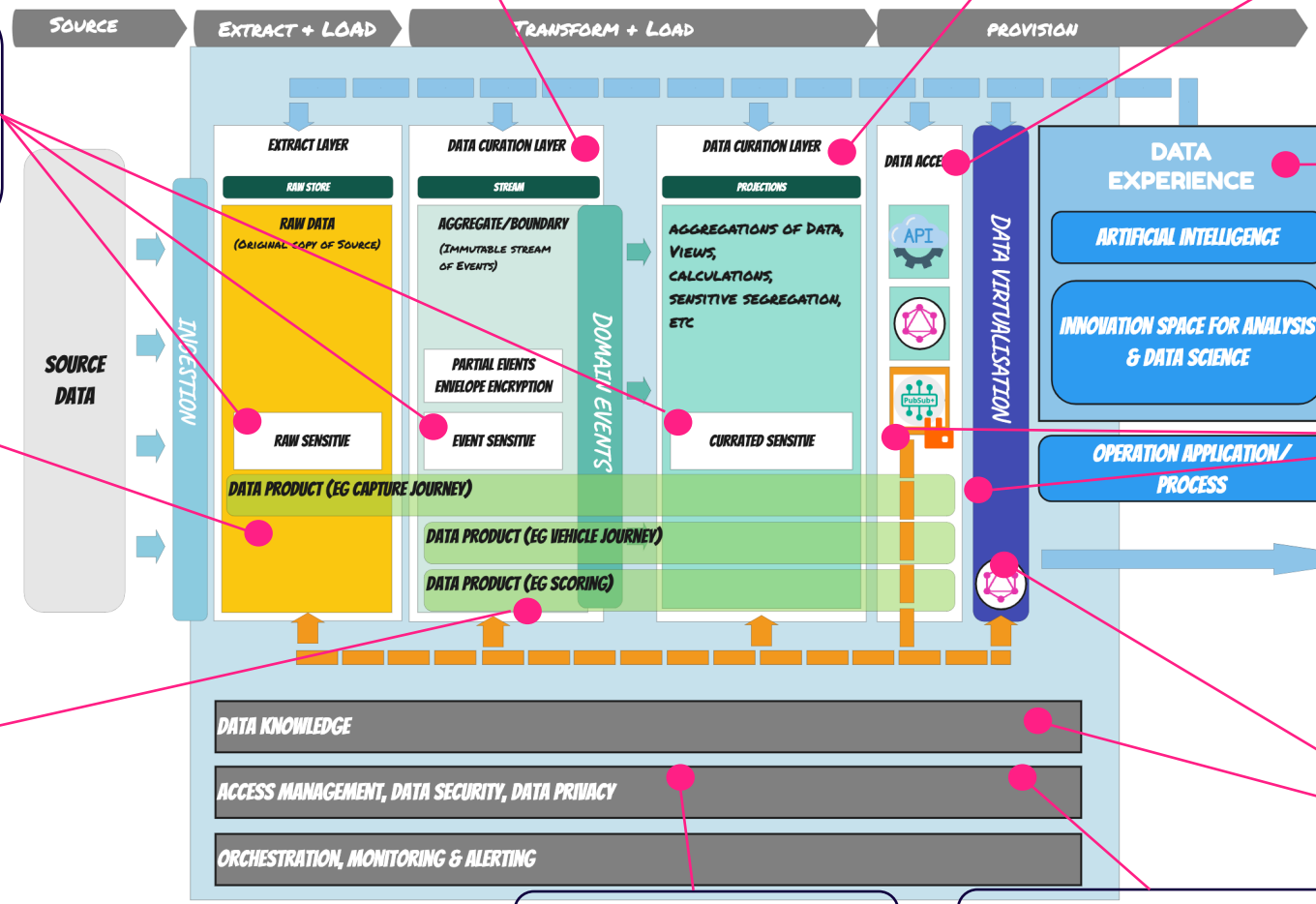
Reference Architecture – Data Product Controls

Store is an immutable and sequential capture of the domains behaviour and NOT current state

Creating views of data by replaying events or/and subscribing to events . Multiple views based on data performance and/or privacy

Privacy Parameters define a permitted use case process for onboarding of new initiatives

"Sensitive" data stores (or streams/topics) implemented to store Strictly Confidential data or data with a higher security criticality, with restricted access. Especially if the **KMS** is not viable.



Should NOT consume from the Event/Stream Store. Should only come from a projection – CQRS Pattern

Raw Store is an immutable and sequential capture of the source data. Though this could be just a landing area and locked down

Granular level, environment specific, Authorization model & service accounts used to provide appropriate levels of operator access;

All services including VMs are secured with end-point protection

Could use the data query pattern GraphQL which is self documenting. Can resolve data based on different Authorization. (ie RBAC model will retrieve different projections but return the same model, or validate before requesting).

All credentials held in secured access management

Use cases approved via governance forums in line with privacy parameters

Reference Architecture – With KMS

PII/SPI data always encrypted. PII/SPI data made unavailable by default

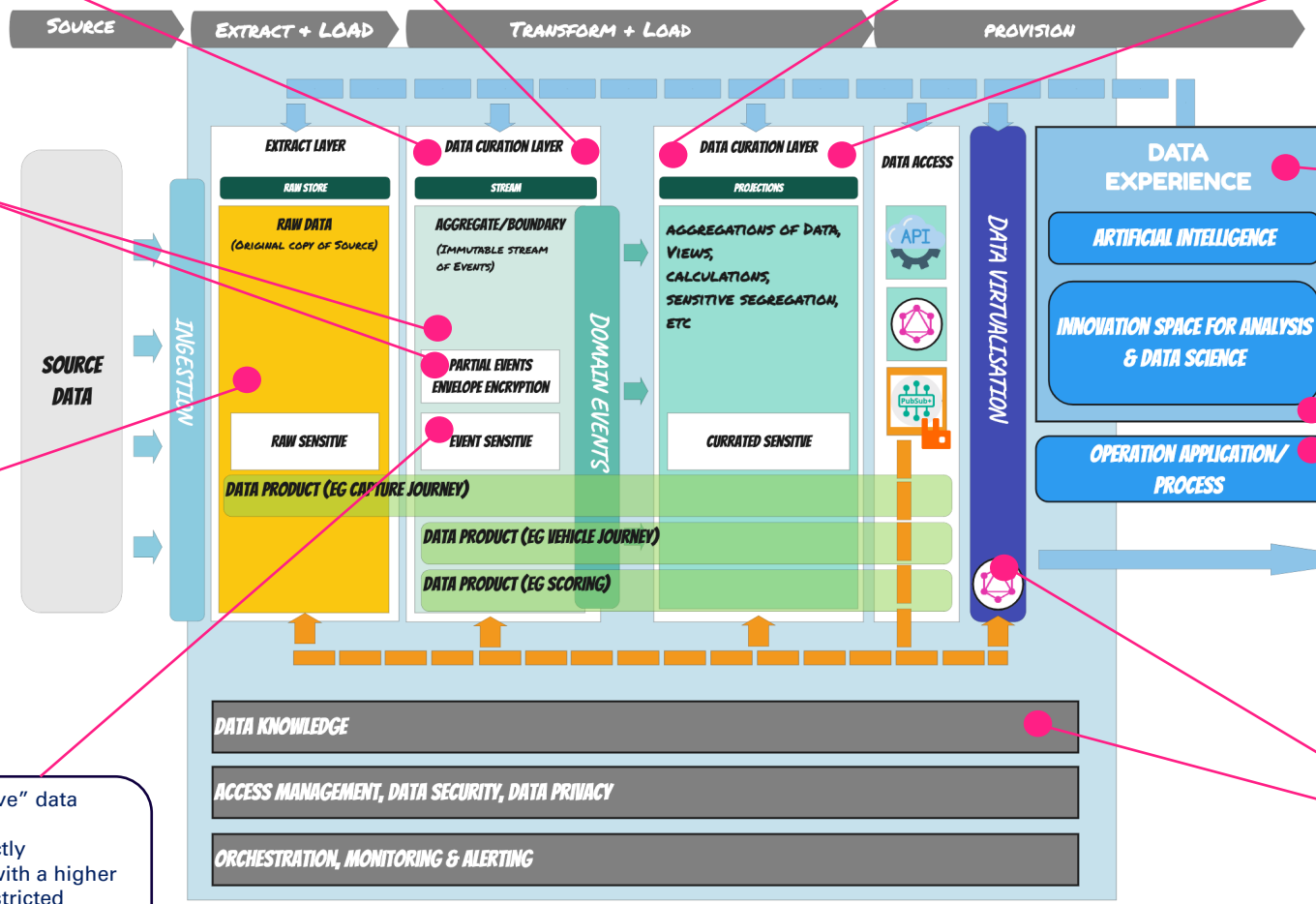
Store is an immutable and sequential capture of the domains behaviour and NOT current state

KMS could be used for views as well instead of "Sensitive" data stores. Schemas defining restricted fields or models via PII or data grouping. Only allowed data will be decrypted.

Creating views of data by replaying events or/and subscribing to events . Multiple views based on data performance and/or privacy

Key Management System (KMS)

1) Encryption key (ie **Customer**) which allow crypto-shredding by deleting the key.
2) Identify PII fields and data grouping and extend the event schema
3) Partially encrypt PII data grouping fields on the event.



Should NOT consume from the Event/Stream Store. Should only come from a projection – CQRS Pattern

KMS will only decrypt data based on appropriate levels of operator access;

Raw Store is an immutable and sequential capture of the source data. Though this could be just a landing area and locked down

Another option is "Sensitive" data stores (or streams/topics) implemented to store Strictly Confidential data or data with a higher security criticality, with restricted access. Especially if the **KMS** is not viable.

Could use the data query pattern GraphQL which is self documenting. Can resolve data based on different Authorization. (ie RBAC model will retrieve different projections but return the same model, or validate before requesting).