

ZIS

UFSP – Universal Financial Solution Platform

BY

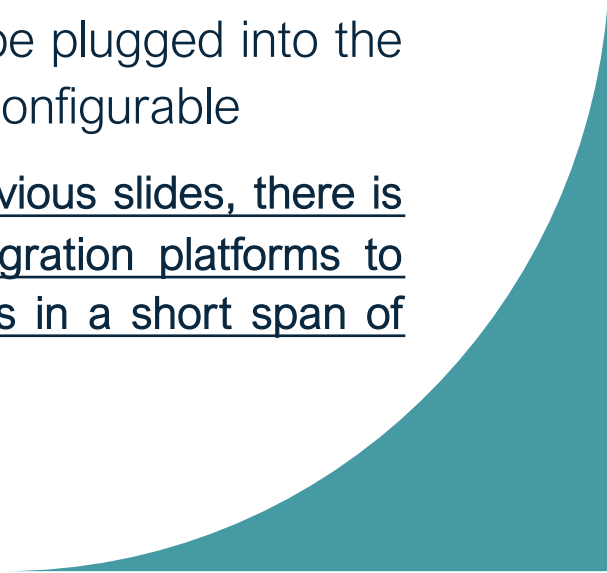
SAMBIT RATH, CTO ZIKZUK



What is the Need

- ✓ Financial Products & Solutions comprise use-cases which are built over multiple data sources over different categories e.g. Accounting, Payments, MCA, Taxation etc. using API Integrations which need to be exposed over these data-sources and need to be modelled as Product or Service offerings
- ✓ There is also a need to Configure or Extend these use-cases / business flows based on specific customer / business needs
- ✓ Other 3rd Party API integrations like CRM capabilities also need to be plugged into the use-cases on customer demand and these integrations need to be configurable

Although there are multiple API Platforms available as mentioned in my previous slides, there is no single Solutions Platform which can work on top of these API / Integration platforms to configure the business use cases to offer scalable and extensible solutions in a short span of time.


A teal-colored decorative curve starts from the bottom right corner and extends upwards and to the left, ending near the middle of the right edge of the slide.

ZIS UFSP - Overview

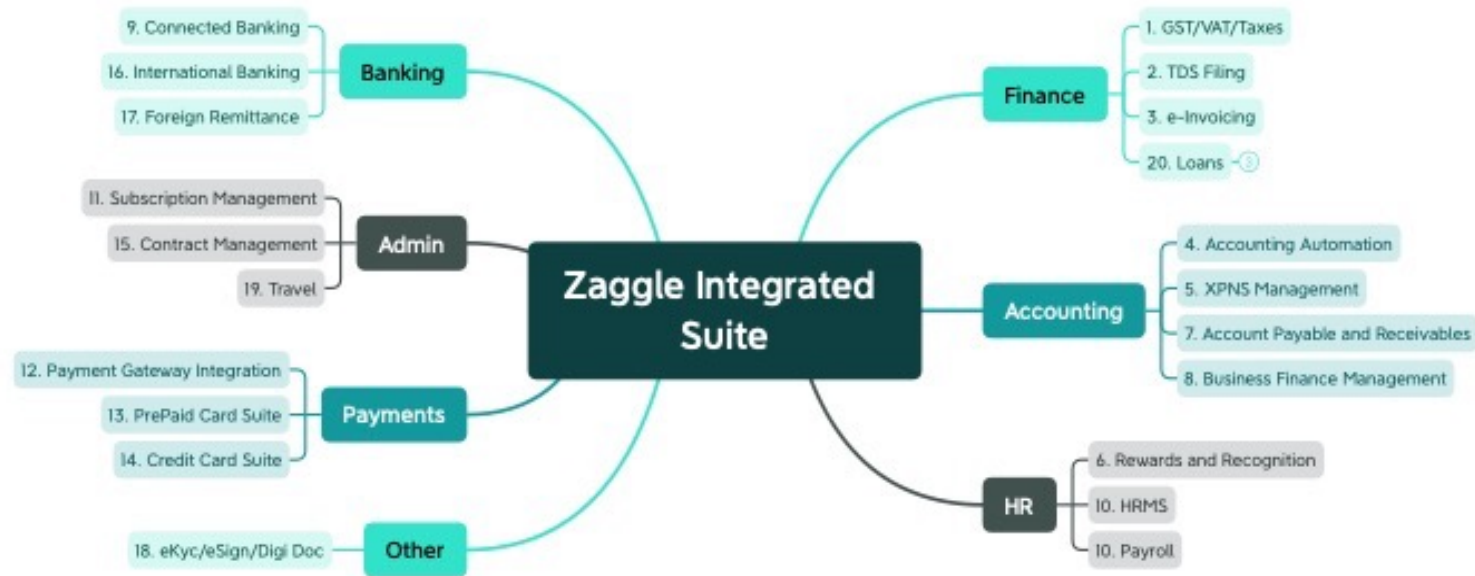
“ZIS” (Zaggle Integrated Suite) or “UFSP” – (Universal Financial Solution Platform) aims to provide a suite of financial solutions through a powerful, scalable and extensible modelling, configuration and integration platform which can cater to the needs of the Fintech industry and support the overall financial ecosystem by enabling common services and API backed by a configurable business process modeler (BPM)

UFSP

The Key capabilities of UFIP are as follows:

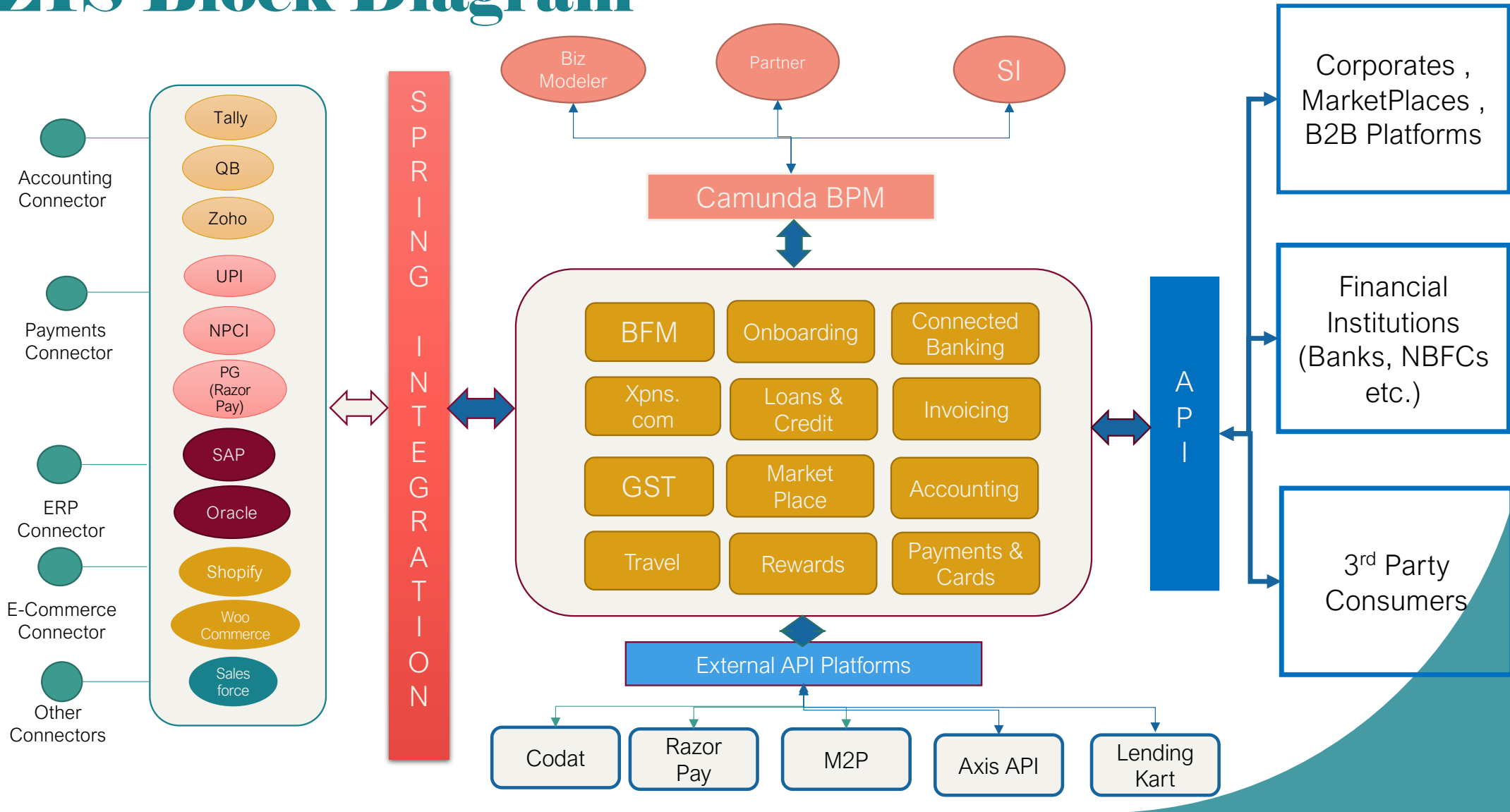
- Configurable and Extensible Use cases based on business needs using a BPM
 - Common suite of API and Product offerings
 - Universal Integration Connectors for various standard financial systems and data providers e.g. Accounting, GST, Bureau, ERP etc.
 - OOTB components and connectors to enable LCNC (Low Code Now Code) configurable business use cases
 - Ability to integrate with any 3rd party system with custom API integrations
 - Decision based Routing and ability to introduce complex approval processes
 - Rule based Decision Support
- 

ZIS – Platform VISION

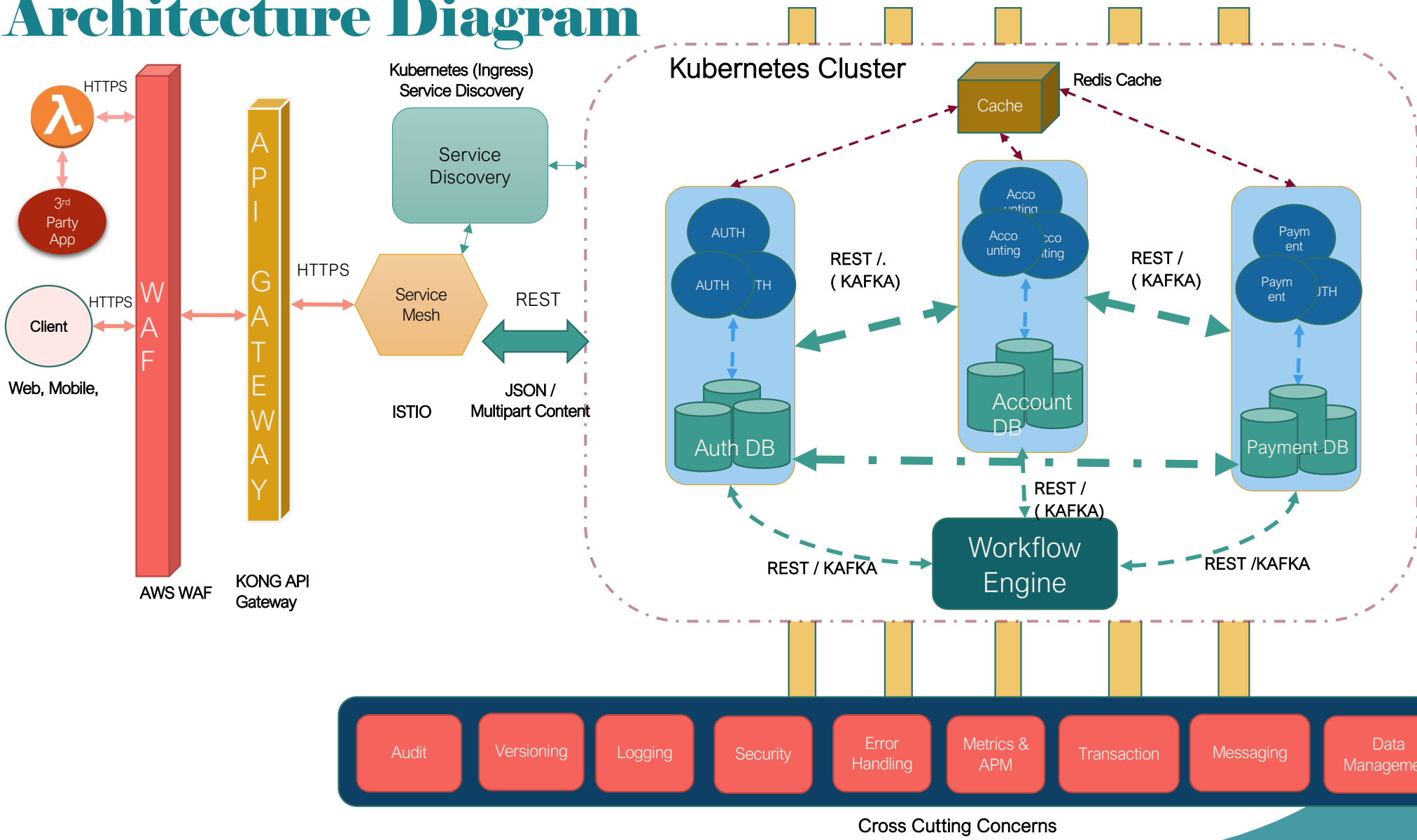


Presented with XMind

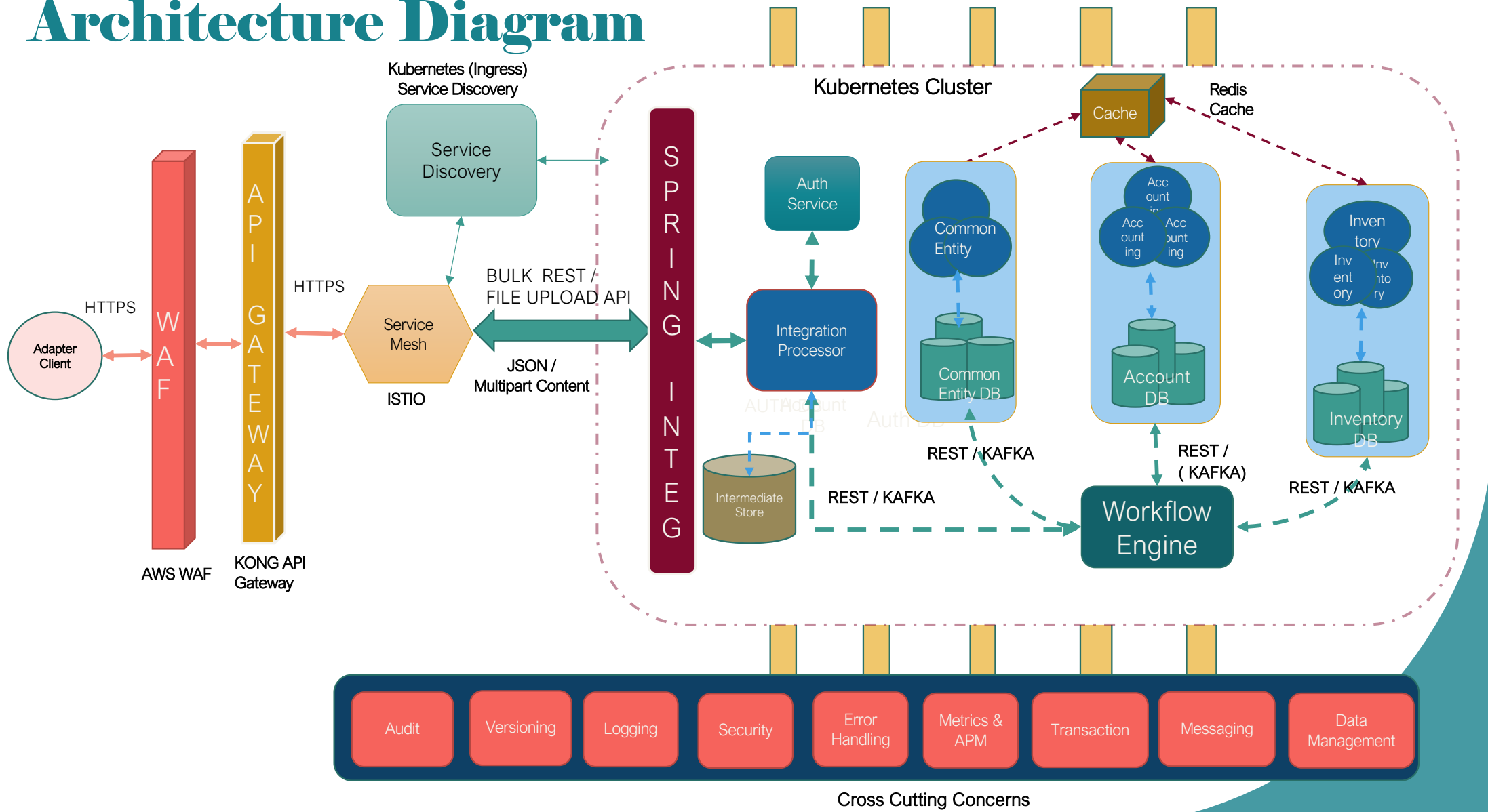
ZIS Block Diagram



Architecture Diagram



Architecture Diagram



The diagram illustrates a KONG API Gateway architecture. It shows the flow of data and services between external clients, the KONG API Gateway, the Kubernetes Cluster, and various Business Services.

External Clients:

- 3rd Party App (Lambd Project):** Connects to the KONG API Gateway via REST (HTTPS) JSON and BULK REST / FILE UPLOAD API (HTTPS) / Streaming.
- Adapter Client:** Connects to the KONG API Gateway via JSON / Multipart Content.
- Client User:** Connects to the KONG API Gateway via REST and JSON.

KONG API Gateway:

- Acts as the central point of entry for all external clients.
- Connects to the **Integration Gateway** within the Kubernetes Cluster.

Kubernetes Cluster:

- Integration Gateway:** Connects to the **Integration Processor**.
- Integration Processor:** Connects to the **Workflow Engine (Camunda)**.
- Workflow Engine (Camunda):** Connects to **Kafka** and the **Batch Processor / Scheduler**.
- Authentication & Authorization Service:** Connects to the **Integration Gateway** and **Kafka**.
- 3rd Party Integrations (Social, LMS, Karza, GST etc.):** Connects to **Kafka**.

Business Services:

- AI/ML:** Connects to the **Cache**.
- Cache (Redis):** Connects to the **Data Warehouse**.
- Data Warehouse:** Connects to **Elastic, Kibana, Data lakes, Tableau**.
- Business Services Grid:**
 - Sales, Payables, Banking, Payments
 - Purchase, Common Entities, Inventory, Analytics Reports Dashboard
 - Receivables, Licensing & Subscriptions, Entries, Notifications
 - Settings, User Mgmt., OCM, Partner Portal

S3

kafka

Kong

Istio

Amazon RDS

mongoDB

redis

elastic

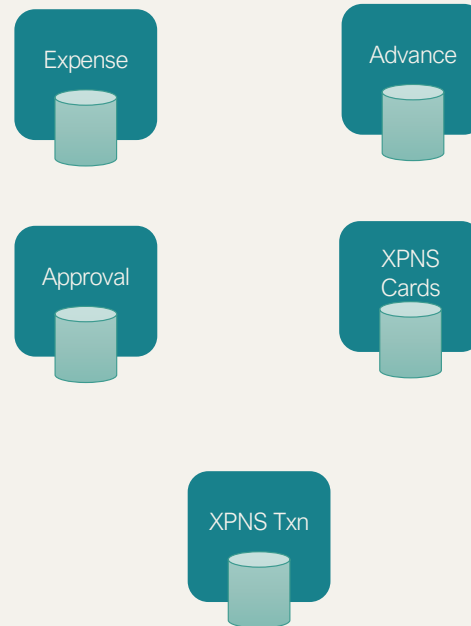
CAMUNDA

Services Distribution

Common



XPNS



BFM



Tech Stack

