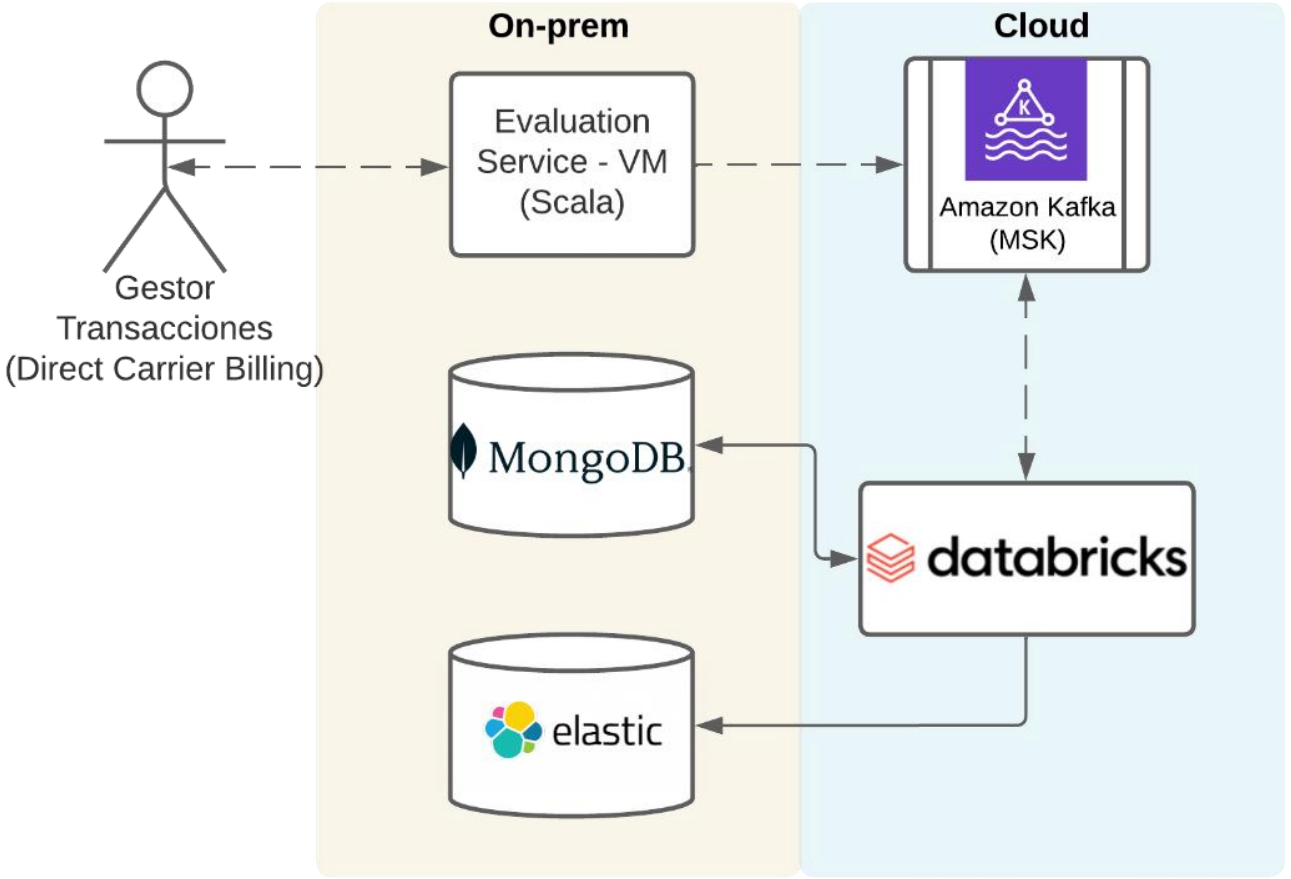
COMO APLICAR NoSQL EN CASOS REALES

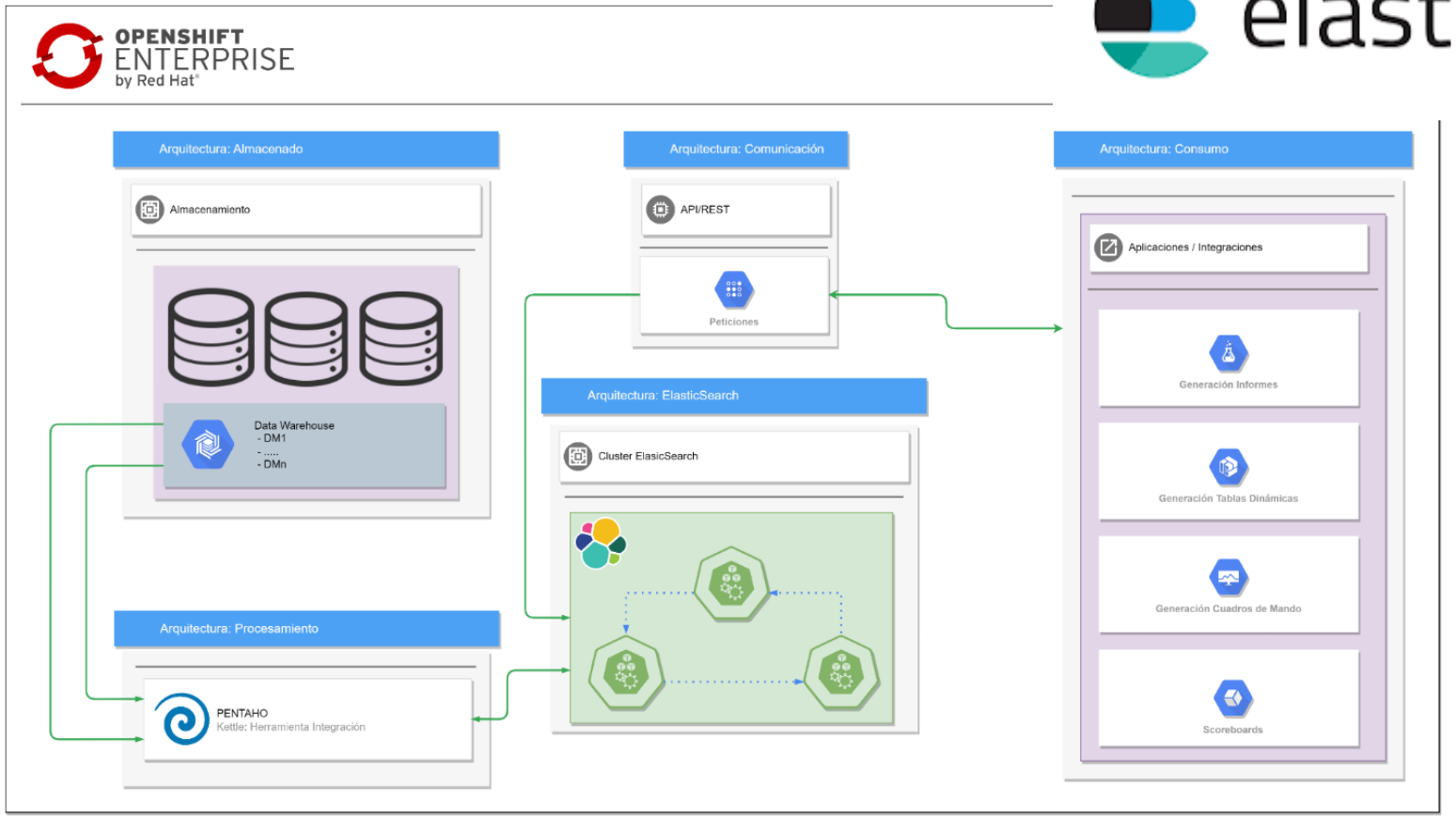
- 1. Predicción de impagos

en e-commerce



COMO APLICAR NoSQL EN CASOS REALES

- 2. Búsqueda de documentos optimizada en las AAPP

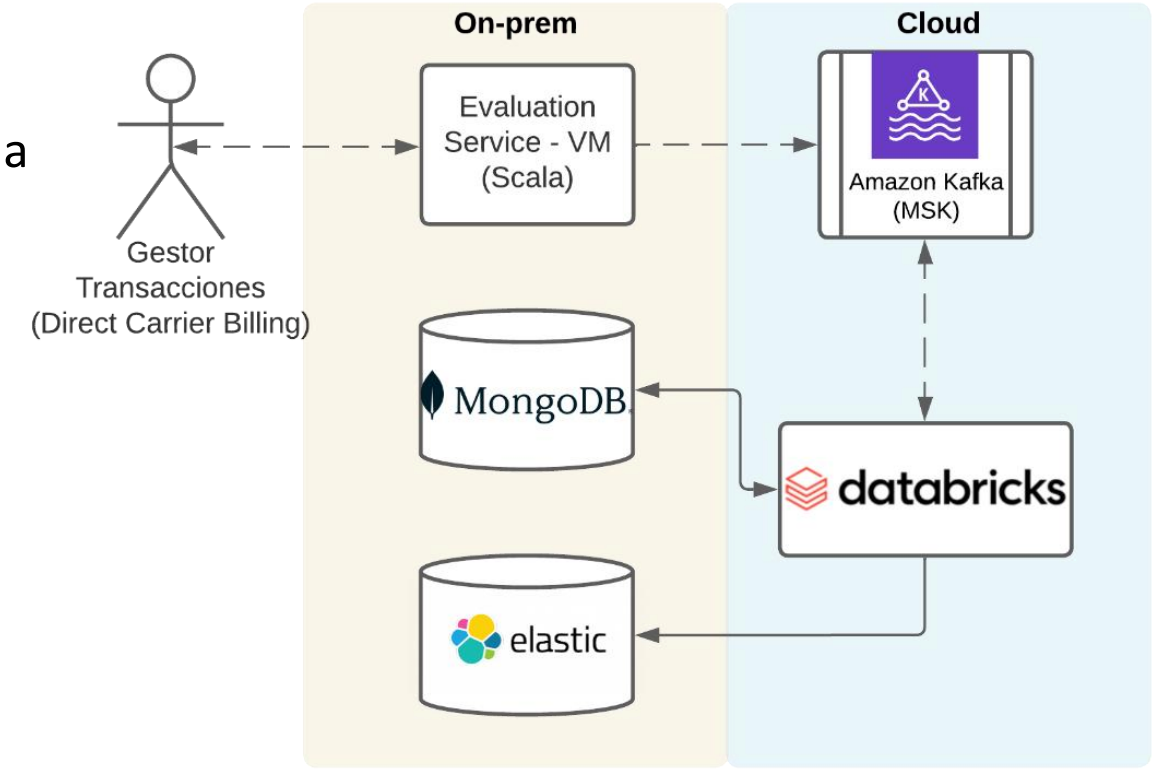


1. Predicción de impagos en e-commerce

- **Objetivo: Rechazar las transacciones con alta probabilidad de impago**
  - Ej. Apple Store, Google Play, Play Station,…
  - Facturación única a través del operador de telefonía (Direct Carrier Billing)
- **Escenario de datos**
  - **Clientes**: id, alta en DCB, tipo de contrato, tipo de cliente, perfil de riesgo…
  - **Transacciones:** id de cliente, site (ej. Google, Apple,…) , operator (Vodafone, Másmóvil, Movistar…), precio, status, …
    - Se reciben en **tiempo real**. Actualmente unos **100 millones de transacciones por cada operador**
  - **Bad-Debts:** Información real de facturas impagadas, proporcionada mensualmente por cada operador

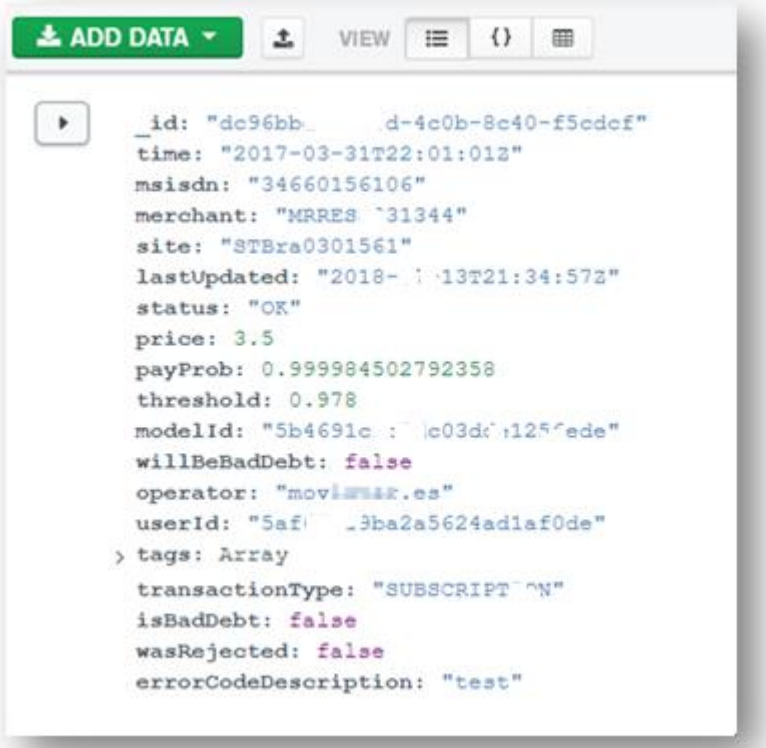
1. Predicción de impagos en e-commerce

- **Arquitectura híbrida Big Data cloud/on-premise**
  - Cloud: Amazon AWS Databricks, Kafka MSK
  - On-premise: **Mongo DB, Elasticsearch**, MySQL
- **Mongo DB** como almacenamiento principal del sistema
  - Clúster de 3 nodos
  - **Carga** de transacciones y usuarios en **tiempo real**
  - **Lectura masiva** de esos datos para el entrenamiento de modelos de clasificación
  - **Almacenamiento de los modelos** entrenados

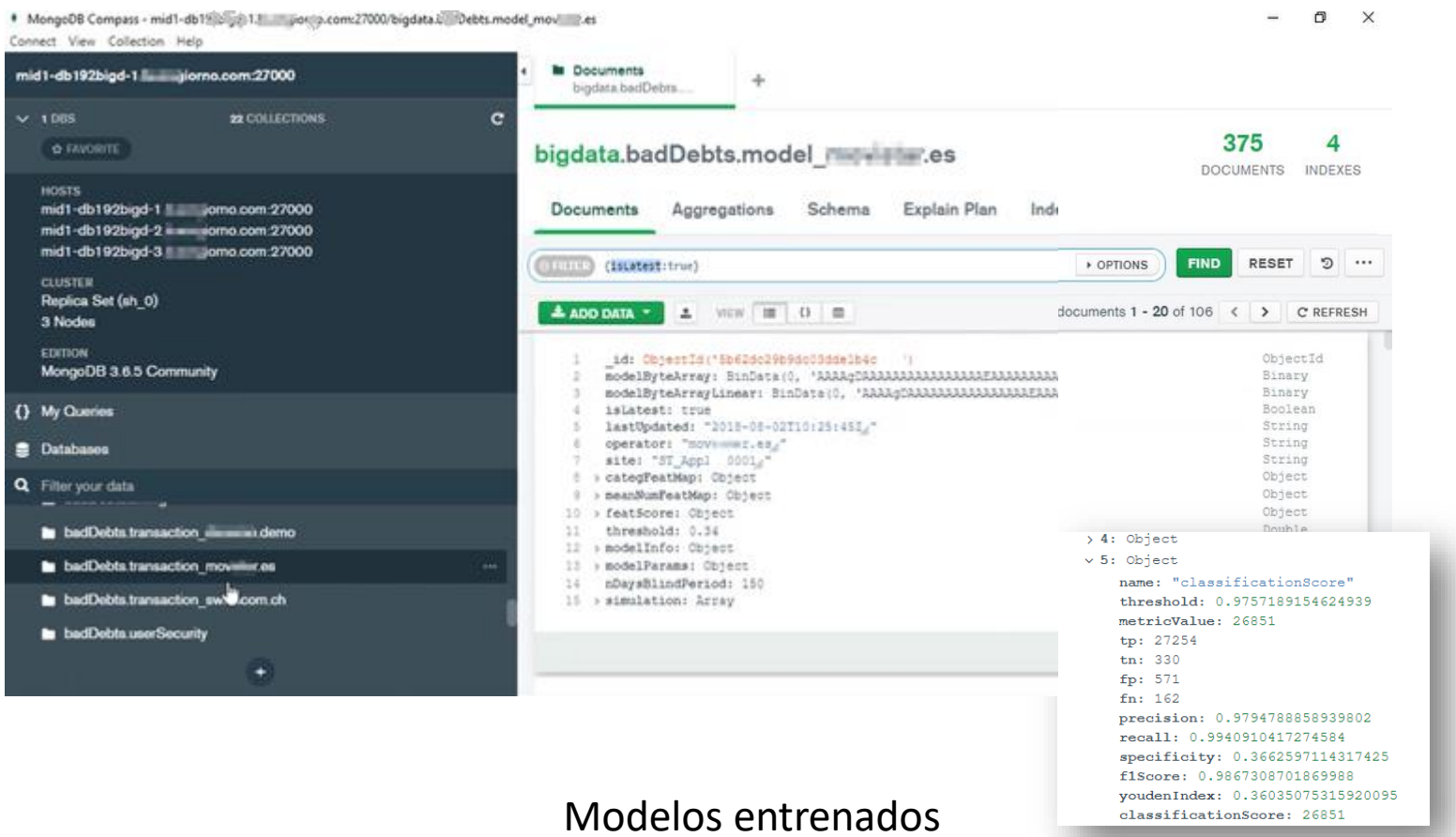


2. Predicción de impagos en e-commerce

- **Mongo DB** como almacenamiento principal del sistema



Transacciones



Modelos entrenados

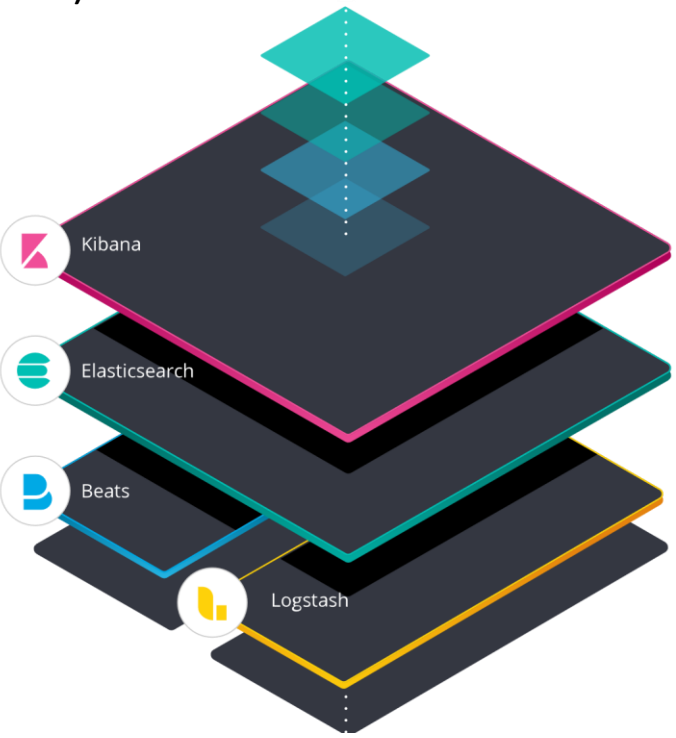
2. Búsqueda de documentos optimizada en la administración pública (Elascticsearch)

- **Objetivos / Caso de estudio**
  - Proporcionar un **sistema de búsqueda** que permita obtener los **tramites solicitados por el ciudadano** de una forma **ágil**, **accesible** y con unos **tiempos de respuesta ultra rápidos**
  - Como requisito **adicional**, es necesario disponer de un **sistema de recomendación** para el ciudadano **basado en las búsquedas realizadas**

elasticsearch

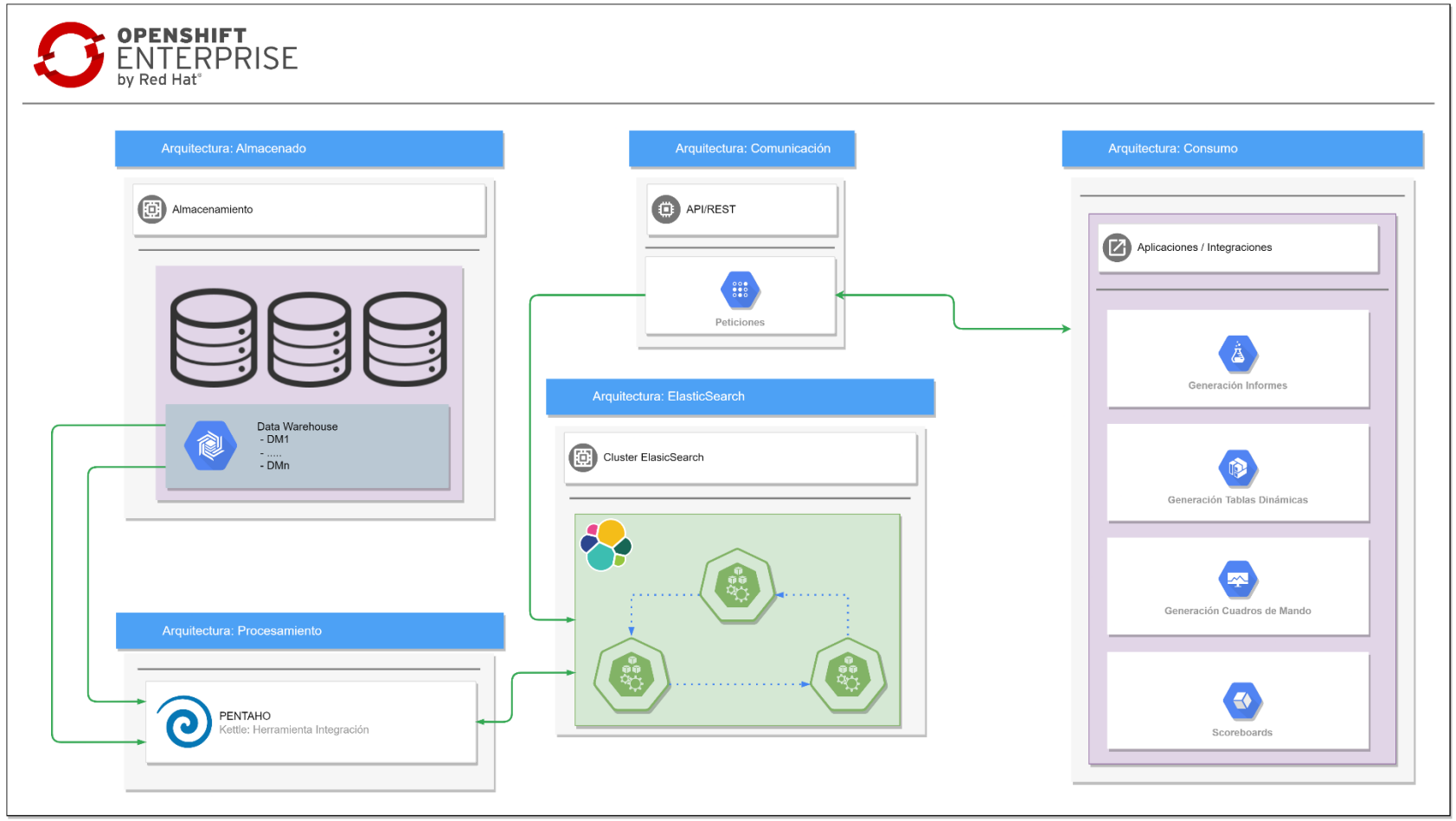
2. Búsqueda de documentos optimizada en la administración pública (Elascticsearch)

- **¿Qué es Elasticsearch?**
  - Motor de búsqueda **Open-source**, desarrollado en Java y de tipo **NoSQL Documental**
    - Respuestas instantáneas mediante el uso de **Índices Invertidos** (Inverted Index)
  - El ecosistema completo de Elastic se conoce como **Elastic Stack (ELK):**
    - ElasticSearch: módulo principal de búsqueda y almacenamiento
    - Logstash: modulo de ingesta
    - Kibana: herramienta de visualización y monitorización



2. Búsqueda de documentos optimizada en la administración pública (Elascticsearch)

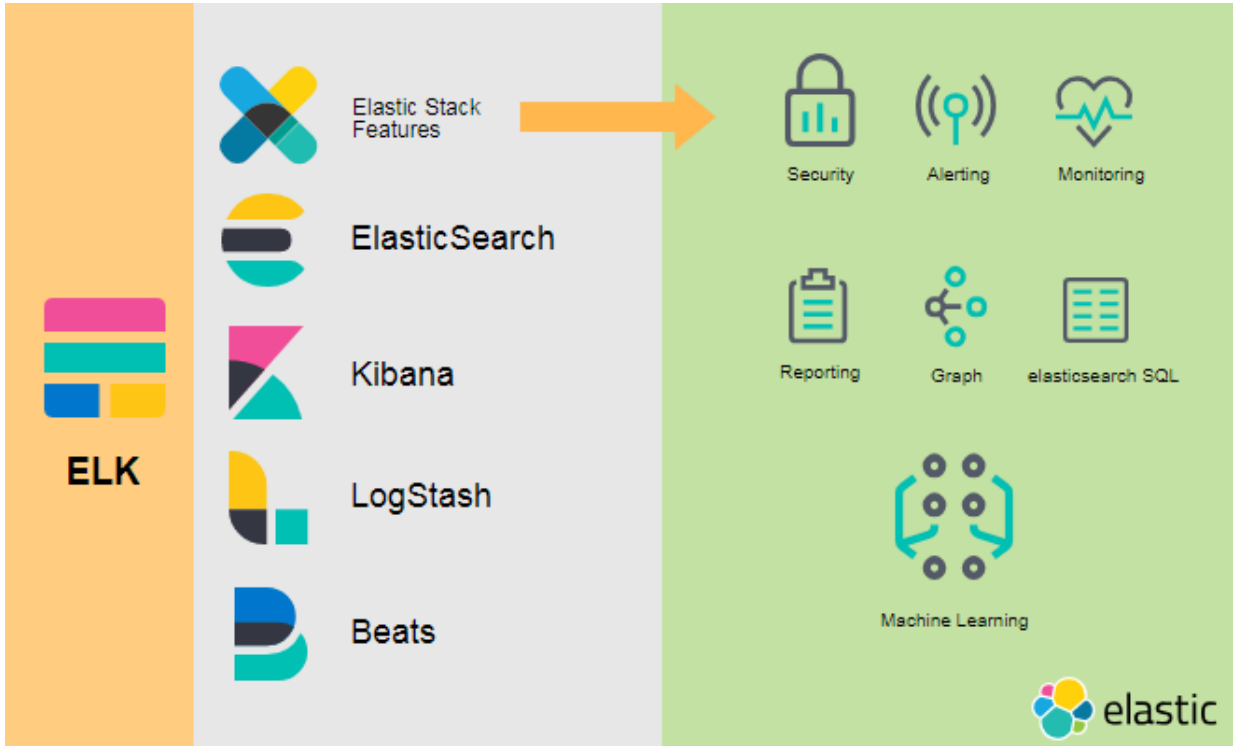
Arquitectura y flujos de datos



2. Búsqueda de documentos optimizada en la administración pública (Elascticsearch)

Aspectos y técnicas destacables

- **Facilidad** y **rendimiento** a la hora de insertar, actualizar y/o borrar documentos
- Propio lenguaje de búsqueda a través de su **API/REST**
- Posibilidad de integración con cualquier aplicativo que haga uso de API/REST



2. Búsqueda de documentos optimizada en la administración pública (Elascticsearch)

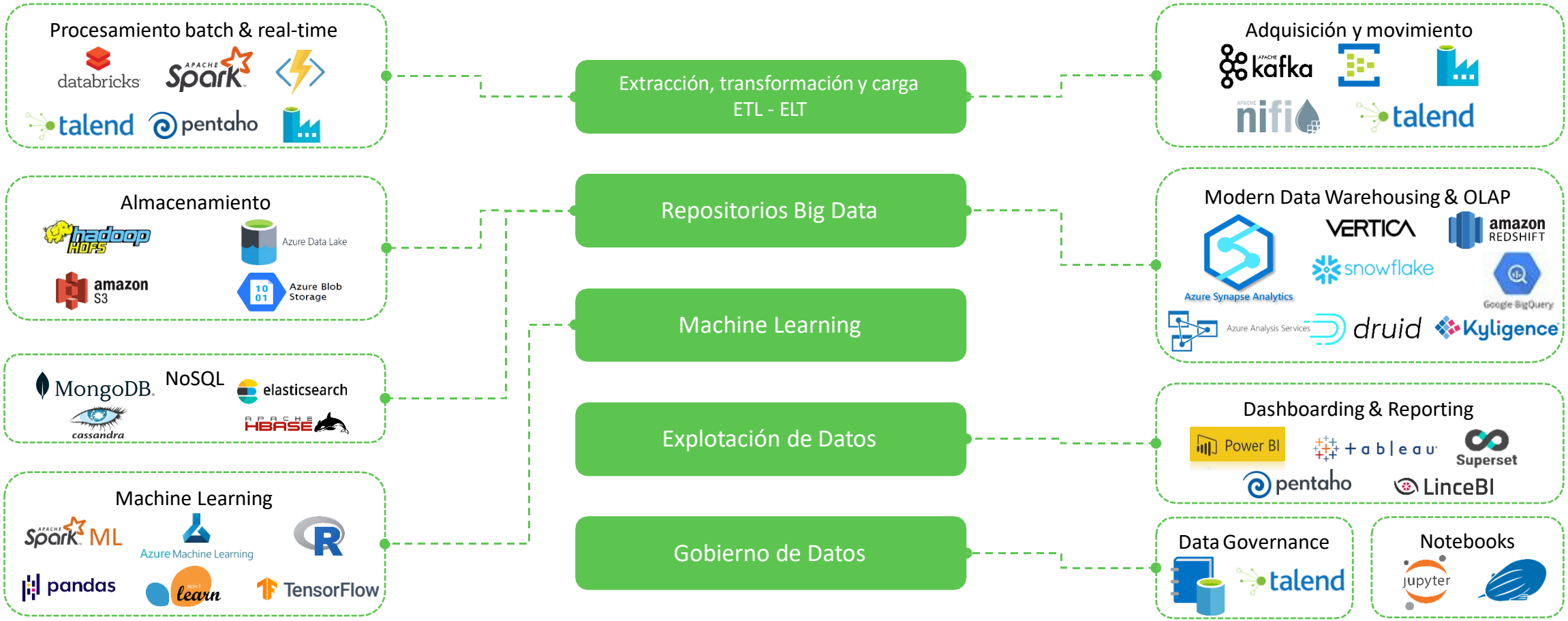
- **Retos y soluciones**
  - Complejidad en el **mantenimiento**
    - Desarrollo de procesos de carga incremental para asegurar la **integridad** del dato sobre los **índices**
  - Opciones de **seguridad** avanzada sujetas a licencias.
    - Uso de integraciones terceras como Keycloak y Azure AD para proveer una mayor seguridad

2. Búsqueda de documentos optimizada en la administración pública (Elascticsearch)

- **Conclusiones**
  - **ElasticSearch** ha conseguido cumplir con todos los requisitos especificados y proporcionar a los ciudadanos un motor de búsqueda ágil con tiempos de respuesta muy rápidos

Técnicas y tecnologías

Principales Aplicaciones y Tecnologías (Comerciales y Open Source):



Uso de Arquitecturas Open Source

Ejemplo de Arquitectura 100% **open-source** para implementación **Data Lakes**: