Evolución de una arquitectura

Arquitectura de referencia

- ¿Quién es quién?
  - Edge
  - On-premise / Cloud
- ¿Es importante el quién es quién?
- ¿Es importante saber qué va en la nube y qué no va en la nube?
- Telemetría, del IT a la planta o de la planta al IT
  - mqtt, mqtt-sn, CoAP, ¿http?
  - tpc/ip vs udp

Arquitectura de referencia

- Gobernanza del dato
  - Quién puede ver qué, pero también, como lo puede ver y para qué lo puede ver
  - Seguridad del dato (comunicación y almacenamiento)
  - Auditoría del datos
- Arquitectura lambda
- Productos que la componen
- Proyectos reales

¿Quién es quién?

- Edge (los nodos o las zonas dónde se produce la información RAW)
  - En las primeras fases de la captura se habla de edge sin capacidad de computo (captura de datos sin más)
  - En las fases subsiguientes es importante entender que las zonas periféricas deben estar gestionadas y deben tener capacidad de computo (captura de datos, etl, alarmas tempranas)
- Cluster (Sistema en el que se almacena y coordinan los procesos relacionados con los datos)
  - En primera fase son sistemas de almacenamiento y procesado
  - En segunda fase se vuelven sistemas de cómputo y almacenamiento federados con capacidad elástica

¿Es importante quién es quién?

- Es importante saber
  - Quién produce datos (qué vamos a medir)
  - Cómo van a llegar al sistema (cómo los vamos a mover)
  - Cómo se van a almacenar (cómo los vamos a procesar) ¿textos al revés?
  - Cómo se van a procesar (cómo los vamos a almacenar)
- En las primeras fases del despliegue no es critico saber si el cluster va a ser Cloud o va a ser on-premis
- Es importante diseñar las piezas que se van a desarrollar pensando en la posibilidad de que el sistema evolucione de cloud → on premise o viceversa

Telemetría / sensorización / captura en planta

- No soy informático … pero vengo del mundo IT (no soy experto en telemetría, ni en sus protocolos)
- Tengo la sensación de que el mundo del IT y el mundo de “la planta” no se parecen mucho, y menos en el entorno industrial
- Creo que hay dos formas de llegar a un proyecto que implica uso de datos de planta
  - Desde el IT o desde la Planta
- Si se llega desde el IT, se olvidan las necesidades de planta (sobre todo para los consultores de IT)
- Si se llega desde Planta … se olvida que que al otro lado hay perfiles de IT

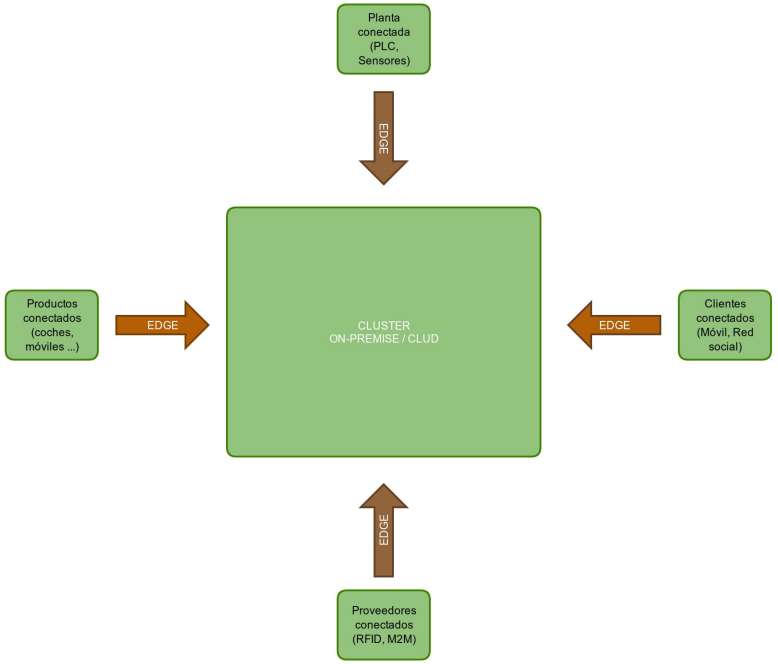
Telemetría / sensorización / captura en planta

- Si al equipo de IT le tienen que llegar datos... y los datos los tiene el equipo de planta... habrá que acordar un modelo de intercambio en el que ambos estén a gusto.
- Entiendo que lo importante es por tanto ... decidir la arquitectura de captación y de transporte hasta el cluster los datos. Protocolos de IoT y de IIoT, capa de transporte, arquitectura
  - Protocolos
    - mqtt
    - mqtt-sn
    - CoAP
    - http?
  - TCP – UDP ..
  - Arquitectura OPC-UA

Telemetría / sensorización / captura en planta

- Lo importante para mi modo de verlo
  - UDP vs TCP (MQTT vs MQTT-NS)
  - Seguridad (SSL/TLS vs DTLS vs IPSEC)
  - Protocolo completo vs mensajería (OPC-UA vs MQTT)
  - Consumo de datos basado en eventos o en consultas (OPC-UA vs MQTT)

Telemetría / sensorización / captura en planta

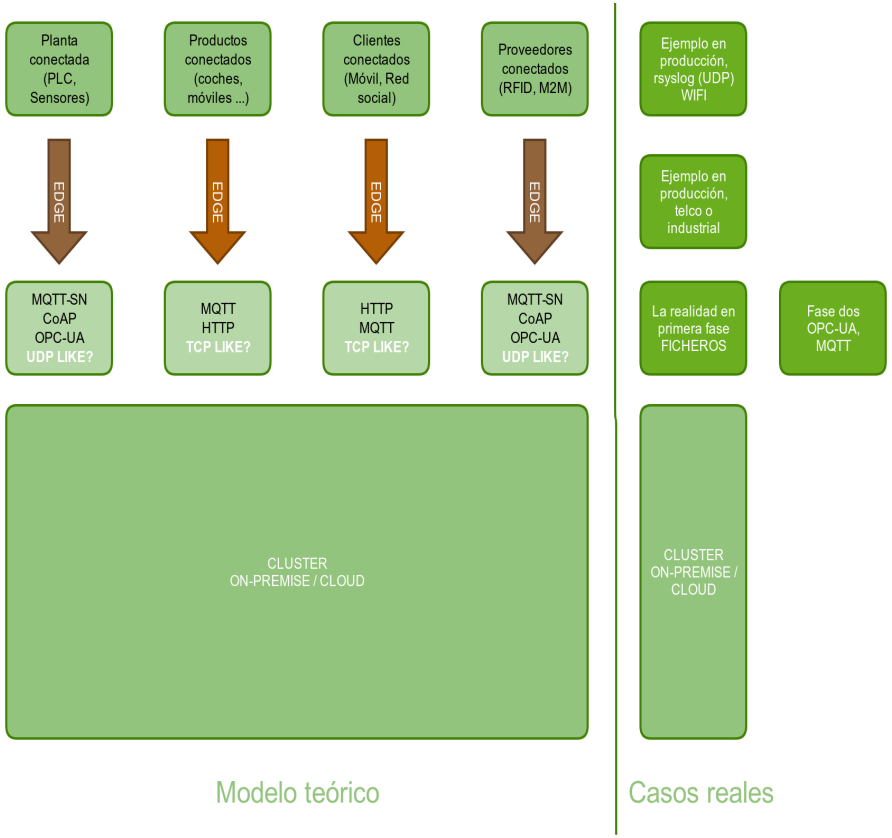


- Planta
- Clientes conectados
- Cadena de proveedores
- Productos conectados

Telemetría / sensorización / captura en planta

- Cómo se procesa esta información que llega desde estas 4 zonas
  - La realidad es que en primera fase …
  - … ficheros en un sftp, en el 90% de los proyectos que hemos realizado
  - ¿Por qué?,
    - El funcionamiento de la planta es más importante que ninguna otra cosa
    - Ya existen mucha información intercambiada entre sistemas
    - Comunicaciones y seguridad

Telemetría / sensorización / captura en planta

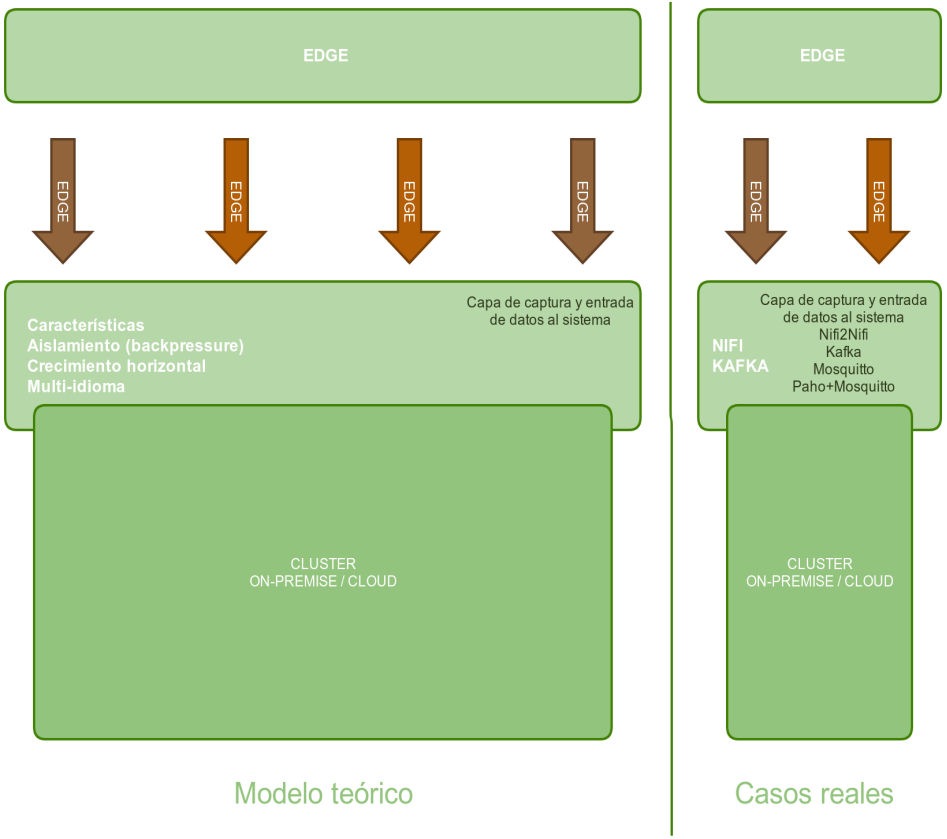


- Idealmente
  - MQTT
  - MQTT-SN
  - OPC-UA
- En realidad
  - 90% intercambio de fichero
  - r-syslog por UDP
- NIFI + MINIFI nos ayuda en la transición

Telemetría / sensorización / captura en planta

- ¿Qué características tiene esa capa de entrada?
  - No debe producir efecto bola de nieve sobre los clientes que la consumen (aislamiento de las bodegas en los barcos)
  - Debe gestionar la congestión de manera transparente a los clientes que la consumen, tanto hacía el cluster como desde los sensores (backpressure)
  - Debe hablar muchos idiomas y protocolos
  - Crecimiento horizontal
  - Gestión del estado

Telemetría / sensorización / captura en planta



- Idealmente
  - Back pressure
  - Reintentos
  - HA, crecimiento horizontal (elástico)
- En realidad
  - Se cumple
- NIFI, KAFKA Paho, mosquitto
- AVRO

Telemetría / sensorización / captura en planta

- Un apunte que viene desde el IT
  - ¿qué mensaje intercambiarías?
    - XML
    - JSON
  - ¿Qué tienen de bueno? ...
  - pero más importante ... ¿Qué tienen de malo?
- Un apunte que viene desde todas partes
  - Seguridad (transporte y en el dato)

Telemetría / sensorización / captura en planta

- Resumen
  - Mosquitto, RabbitMQ, paho
  - Server OPC-UA
  - NIFI + Kafka (Avro)
  - Minifi + NIFI + Kafka (Avro)

Explotación y Gobernanza del dato

- Ya tenemos el dato en el cluster, ¿qué hacemos con él?,
  - desde una visión clásica (bigdata 1.0)
    - Procesarlo
    - Almacenarlo
    - Explotarlo (EDW, ML etc..)
  - desde una visión actual (bigdata 2.0)
    - Procesarlo (enriquecerlo msrv)
    - Liberarlo
      - Explotarlo (Strems, ML, Alarmas, msrv)
      - Explotarlo almacenarlo → Explotarlo (EDW, ML etc..msrv)

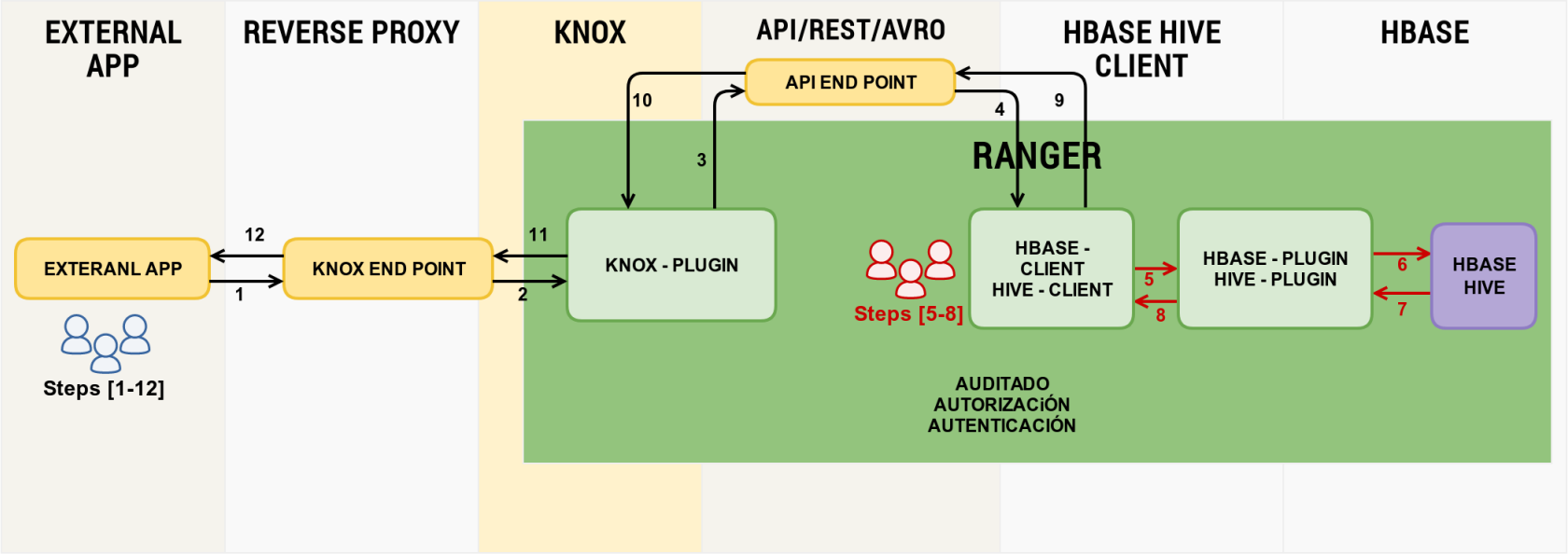
Explotación y Gobernanza del dato

- Ya tenemos el dato en el cluster, ¿cómo lo gobernamos?,
  - Quién puede ver qué, gobernanza
    - Sistema de tags que permite crear reglas
      - El grupo de Científicos de datos ven los datos de venta pero a los datos de usuario se les aplica un hash md5
      - El departamento de calidad no puede ver los datos relativos a I+D
      - Los usuarios con perfil comercial no ven los datos con el tag de expiración
    - Quién ha visto qué, auditoría. Para poder pasar las auditorías de seguridad, las ISO-XXX y cumplir con las leyes
    - El dato a bajo nivel está cifrado .. HTE

Explotación y Gobernanza del dato

- Todo eso se hace con
  - Ranger / Atlas
  - Knox
  - Kerberos
  - Integración con un Directorio Actvio
  - SSL / TLS
- Esto no implica que la seguridad perimetral clásica esté obsoleta
- Sistemas mixtos ... cloud / on-premise
  - Federación de identidades
  - Reglas de seguridad en la nube (firewall)
  - Datos almacenados con cifrado

Explotación y Gobernanza del dato

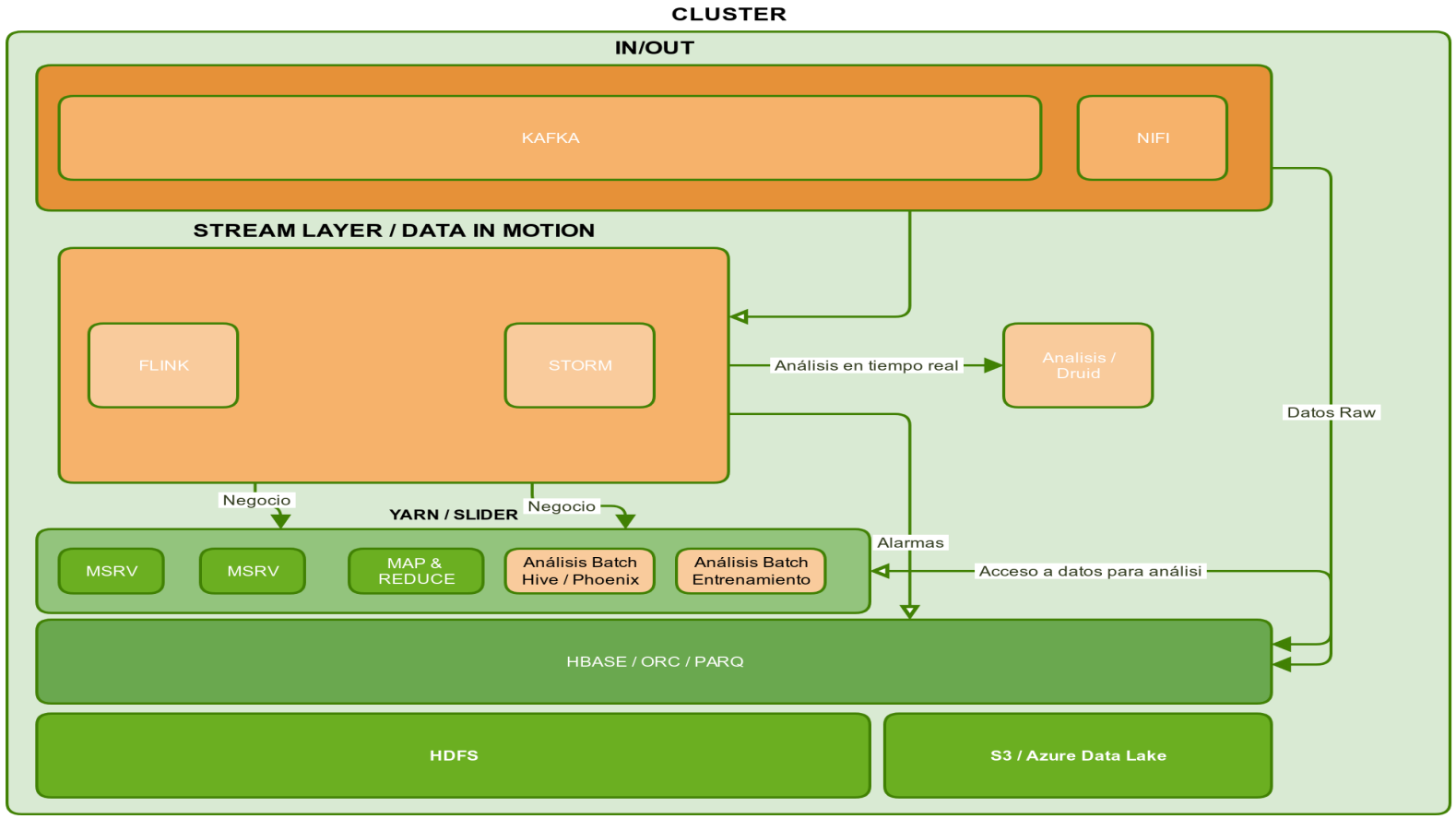


Arquitectura lambda

Lambda architecture is a data-processing architecture designed to handle massive quantities of data by taking advantage of both batch- and stream-processing methods. This approach to architecture attempts to balance latency, throughput, and fault-tolerance by using batch processing to provide comprehensive and accurate views of batch data, while simultaneously using real-time stream processing to provide views of online data. The two view outputs may be joined before presentation. The rise of lambda architecture is correlated with the growth of big data, real-time analytics, and the drive to mitigate the latencies of map-reduce. [Wikipedia]

- La componen las siguientes capas
  - Batch layer (map & reduce / Yarn)
  - Service layer (almacena los datos procesados por las otras dos capas)
  - Speed layer (streams)

Arquitectura lambda



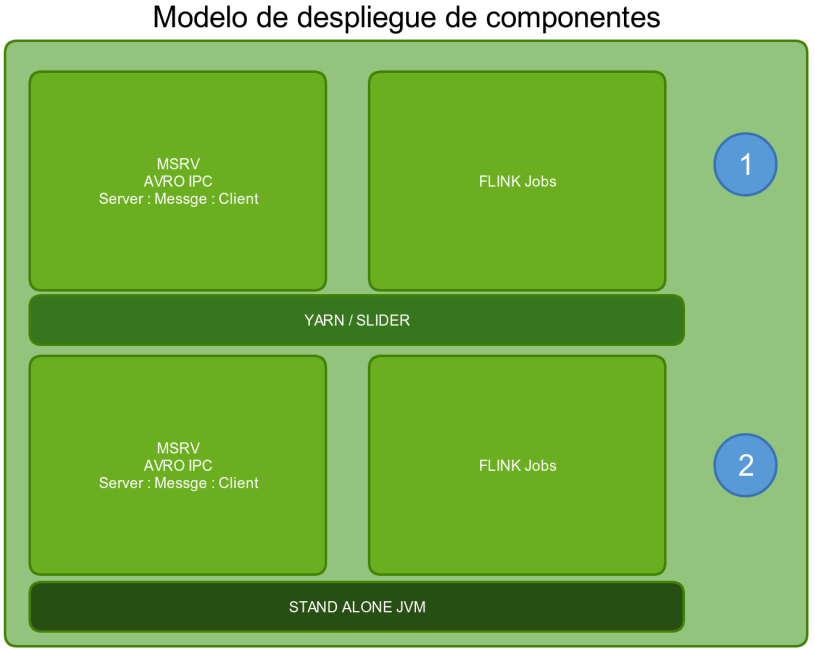
Arquitectura lambda

- Los productos principales que usamos para estas arquitecturas son del ecosistema de hadoop y en concreto de la distribución empresarial hortonworks
  - HDP y HDF
- Están presentes en las dos nubes principales publicas
  - Amazon y Azure
- Y en las privadas como es la de IBM (acuerdo en 2017 entre IBM y Hortonworks)

Arquitectura lambda

- En cualquier caso los productos son los del ecosistema de hadoop por lo que ni a nivel de nube ni de distribución se puede producir un bloqueo de proveedor. (Licencia por servicio en HDP y HDF)
- La lista de los productos que conforman la arquitectura son, entre otros)
  - HDFS
  - YARN
  - HBASE / PHOENIX
  - KAFKA
  - NIFI
  - HIVE
  - FLINK / STORM (SAM)
  - DRUID
  - MAHOUT
  - SPARK
- Los juntamos con el desarrollo/negocio de los microservicios (basados en Avro IPC)

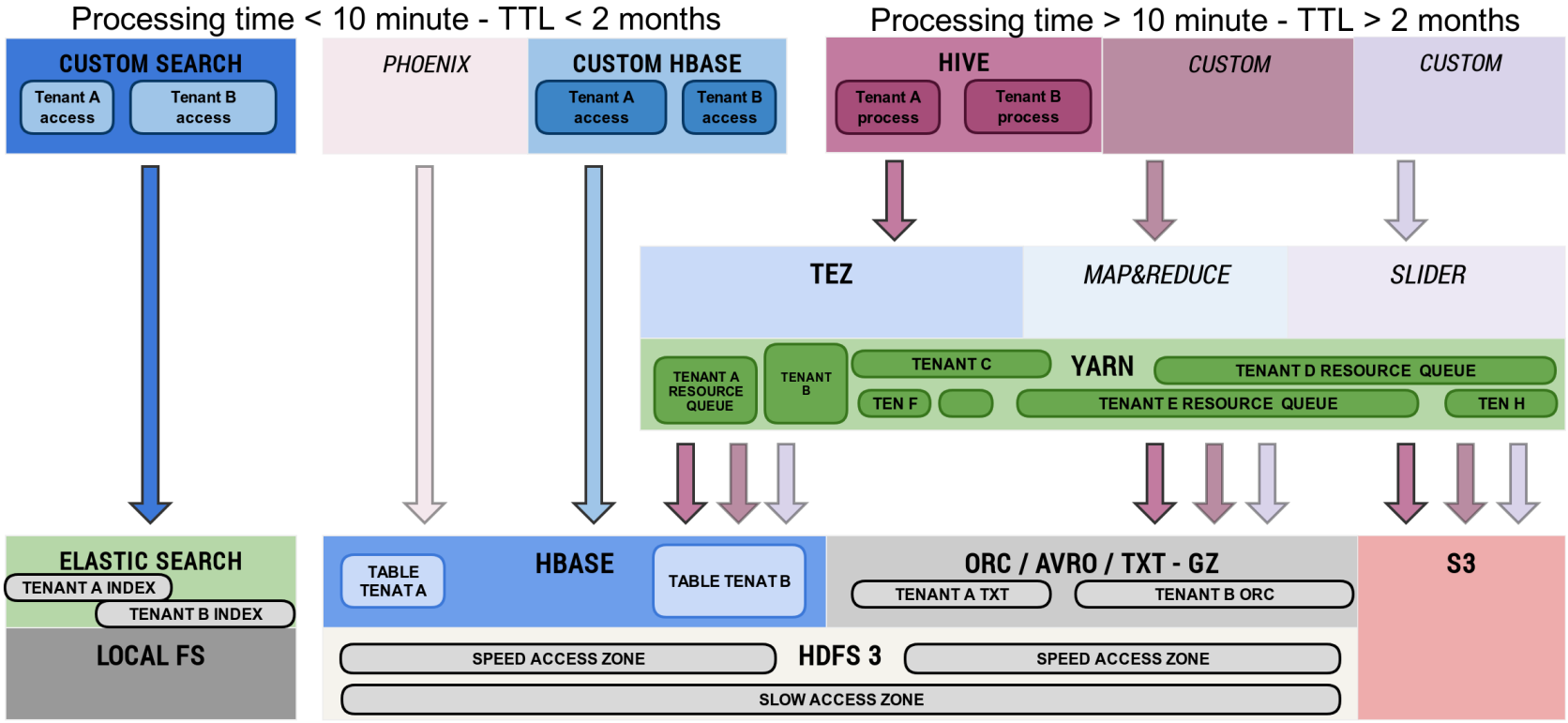
Arquitectura lambda



Micro servicio
Características
Resiliencia
Autodiscover
Multi-idioma
Elásltico

Micro servicio usos
Enriquecimiento
Persistencia
API (Rest)
Carga Data Frames (SPARK)
Basados en HBase

Arquitectura lambda explotación



Conclusión

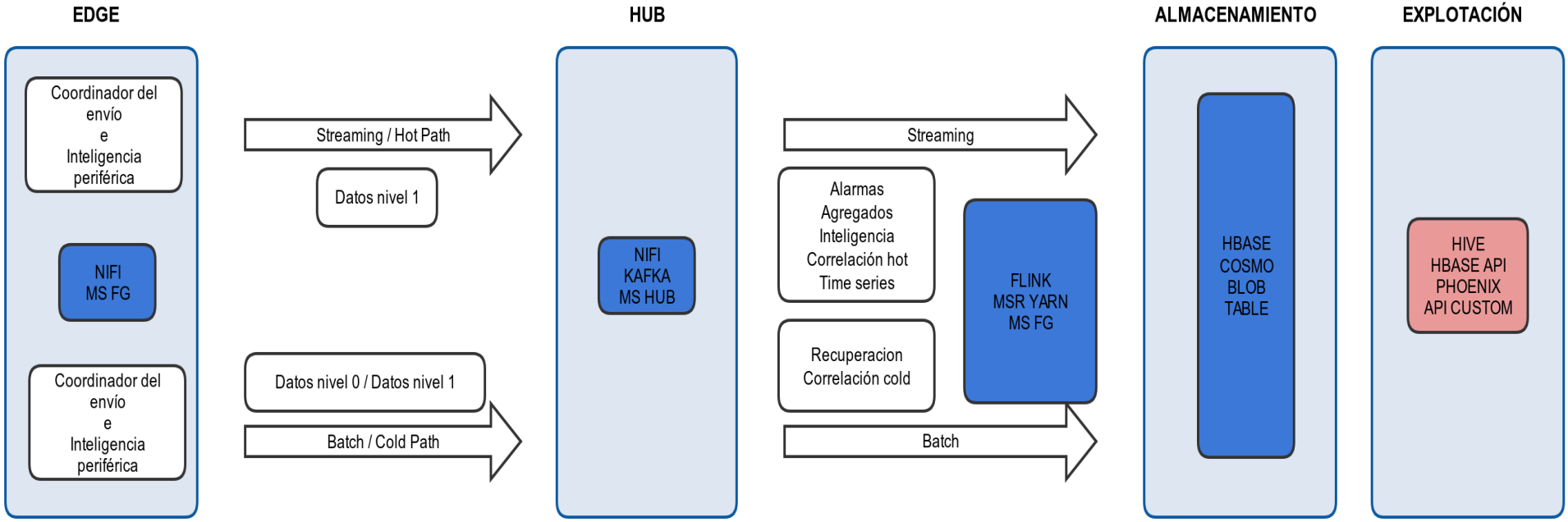
- Ahora que ya sabemos como va todo esto... podemos empezar a pensar en cómo hacer una arquitectura mixta en la que partes estén en la nube y partes en on-premise
- *¿creéis que es algo que estático?*
- ¿qué parte creéis que habría que montar en la nube?
- ¿qué parte en on-premise?
- En qué fijarse
  - Costes
  - Versatilidad de la solución
  - Elasticidad
  - Pero lo primero ... en el caso de negocio que se quiera resolver

Ejemplo de arquitectura mixta en producción

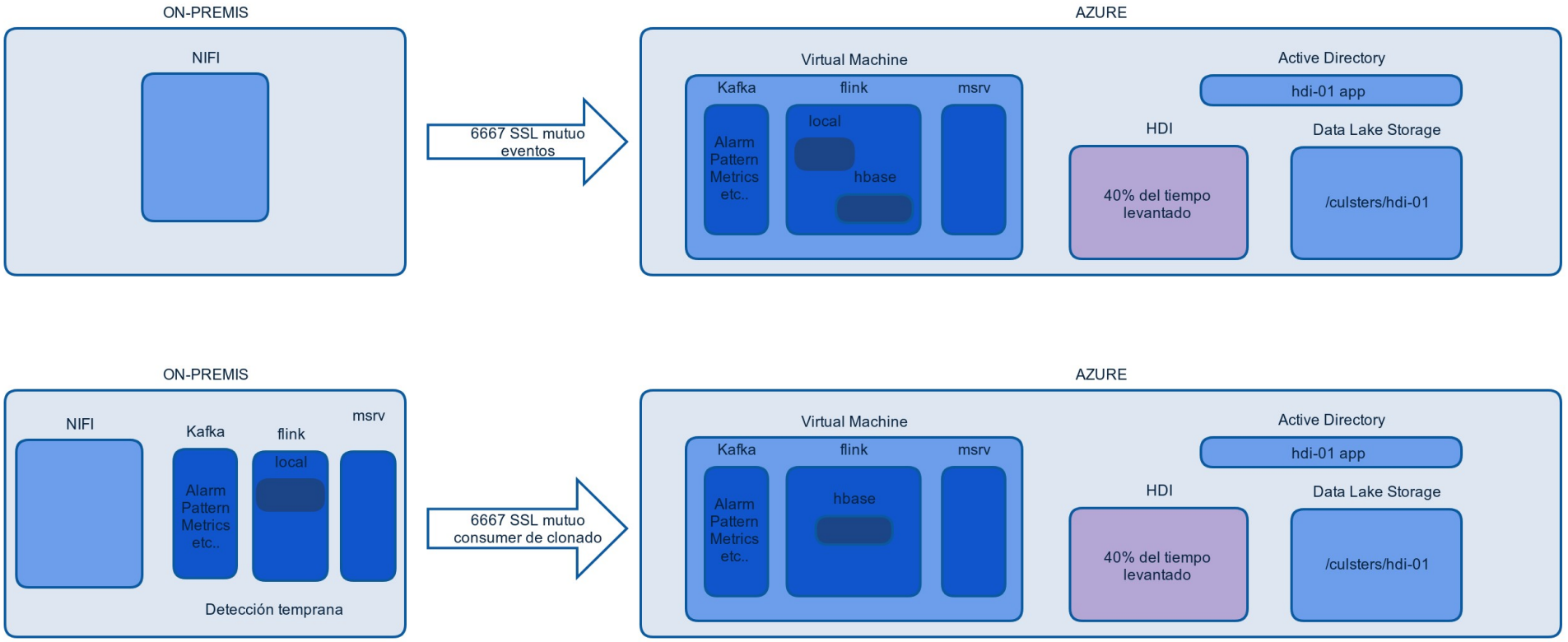
Definición del proyecto

- Captura de datos en planta de varias máquinas de corte
  - MES
  - OPC-UA
  - Ficheros
- Explotación de datos históricos
- Detección temprana de alarmas
  - Datos a los operarios en planta (cambio de pieza)
  - Datos a ingeniería para mejorar los modelos
- Relativamente poca cantidad de datos y relativa alta frecuencia de transmisión de datos

Arquitectura planteada



Arquitectura en producción



Conclusión

- Arquitectura mixta es interesante en cuanto a costes
  - Si el sistema tiene picos de uso
  - Si el sistema puede estar detenido un 30 – 40 % del tiempo
  - Para clusters efímeros (problema de transferencia de datos)
- Peligros de una arquitectura mixta
  - Bloqueo en una nube por el volumen de información a mover
  - Bloqueo por hacer uso de partes privativas de la nube

Gustavo Fernández :: CTO
gus@zylk.net