Datalab

Bilan Sprints #0

Samuel Barbas

GIE-IRIS

FONCTIONS U GIE IRIS SUPPORT

Bilan Sprint #0

- ★ Objectifs
- ★ Solutions
- ★ Bilan
- ★ Préparation Sprint #1

Objectifs

- ★ Chercher une alternative à la BI traditionnelle **OK : Bilan étude Micropole - IRIS**
  - Open Source
  - Innovante
  - Economique
  - Rapide
- ★ Sprint #0 : Monter une architecture sur une infrastructure pilote
- ★ Sprint #1 : Réaliser un prototype sur des uses cases concrets
  - Issus de l'étude Datamart Isilog

Equipe Sprint #0

Pilote IRIS

Product Owner
CPU IRIS

Sc Master
CPI IRIS

MICROPOLE
Driving Distinction

Spécialis taViz
Architecte sionnel
Consultant Se

- ★ Objectifs
- ★ Solutions
- ★ Bilan
- ★ Préparation Sprint #1

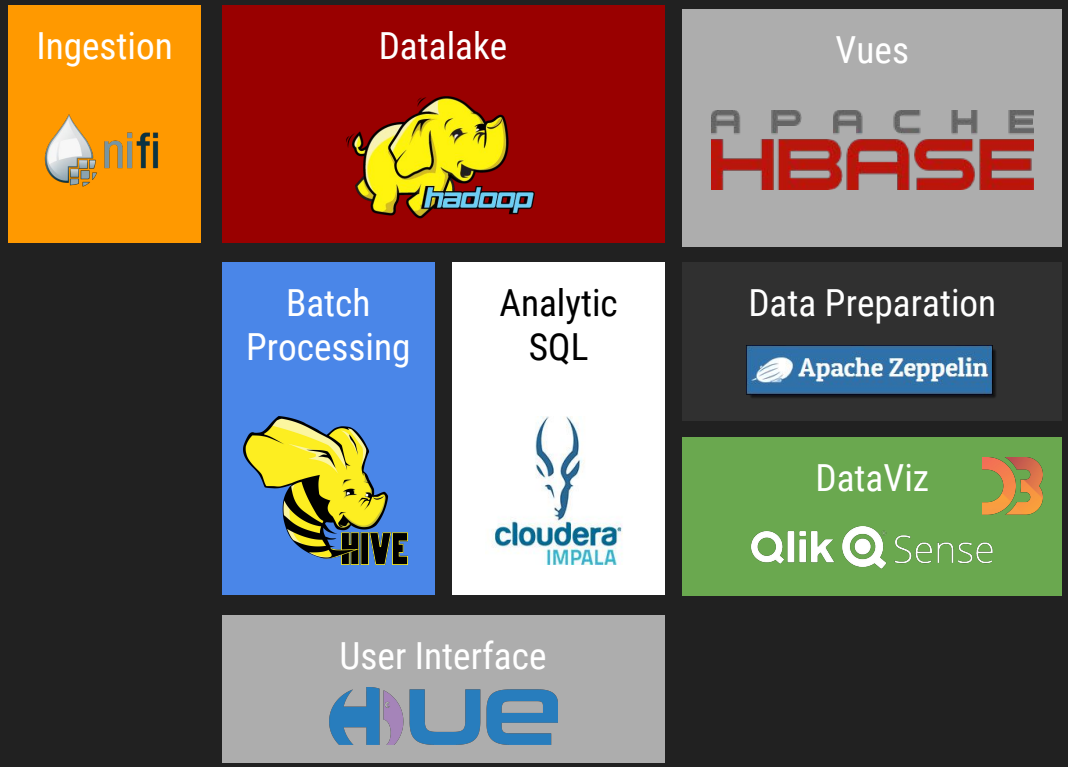
IaaS

OS

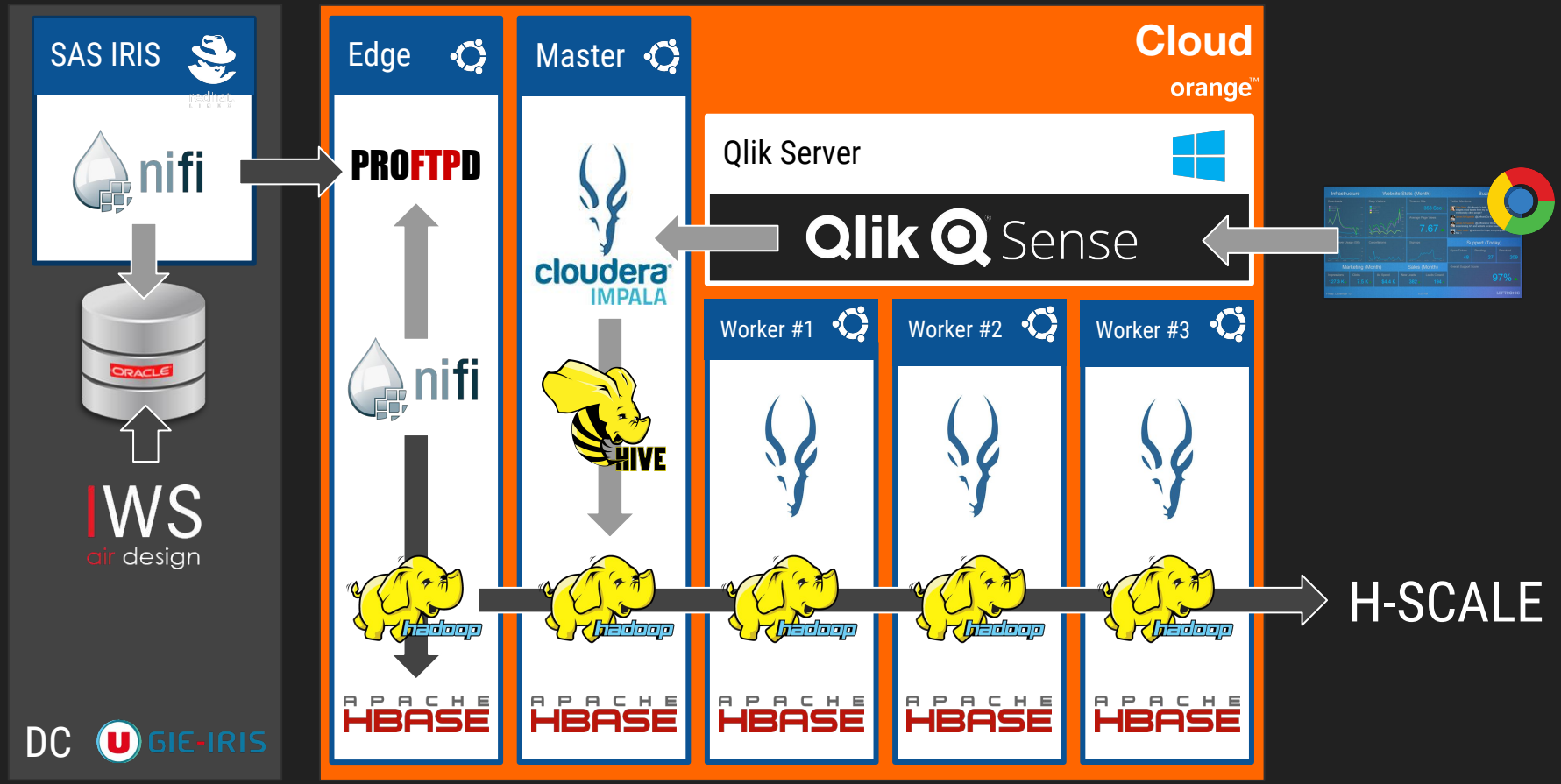
Soft

Client

Solutions


Ingestion
nifi
Datalake
hadoop
Vues
APACHE
HBASE
Batch
Processing
HIVE
Analytic
SQL
cloudera
IMPALA
Data Preparation
Apache Zeppelin
DataViz
Qlik Sense
User Interface
HUE

Architecture

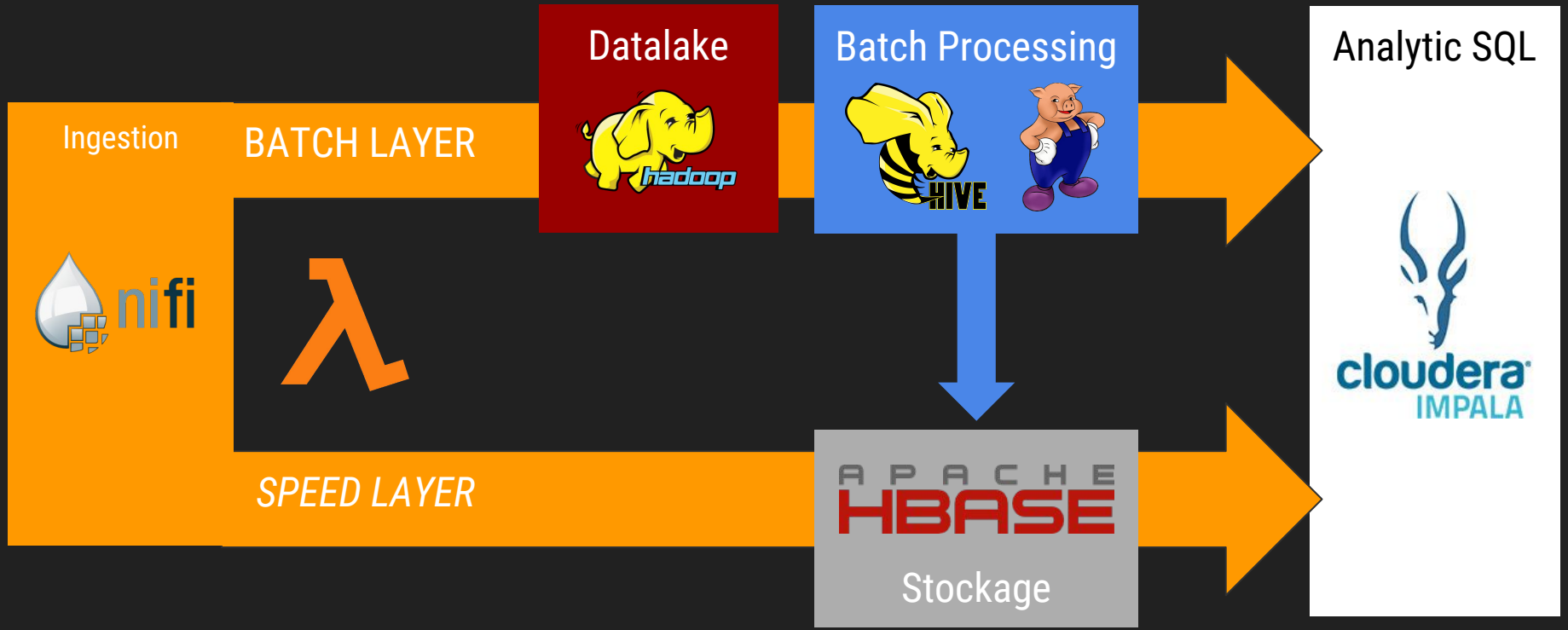


Construction Vues

- ★ Le DataLake contient toutes les mises à jour de données
  - Une même clé peut apparaître plusieurs fois avec une date de mise à jour différente
- ★ Comment avoir une vue “à jour” de nos données depuis la plateforme ?
  - On créé un pseudo - datamart ou “vues métiers”
  - On met à jour chaque enregistrement avec sa dernière valeur
- ★ Comment avoir une vue actualisée / temps réel ?
  - On accepte de faire passer la donnée dans 2 tuyaux : Batch et Speed Layer
  - Batch Layer : Traitement et intégration des données programmée (depuis HDFS)
  - Speed Layer : Chargement brut à l’arrivée des données (sans HDFS)

Une vue ACTIONS chargée (30 Millions d’enregistrements)

Architecture Lambda



- ★ Objectifs
- ★ Solutions
- ★ Bilan
- ★ Préparation Sprint #1

Ingestion

- ★ Développement très rapide : 5 jours pour intégrer 10 tables Isilog
  - Données historique
  - Incrémental
- ★ Ingestion par Apache Nifi
  - Déploiement et développement simple
  - Est-ce industriable ?
    - Simple : possible
    - Complexe : non, préférer une solution type TALEND

Performances

- ★ Le déploiement HBASE a entrainé une dégradation conséquente des performances et des erreurs de saturation mémoire sur certaines requêtes IMPALA
  - Impala + HBASE = nous sommes aux limites de la configuration minimale requise (HBASE = 8/16 go)
  - Répartition des données par région HBASE pas optimisé suite aux chargements d'initialisation
    - Réagencement manuel
  - IMPALA paramétré sans limite d'utilisation de la mémoire
    - Paramétrage à 8 Go max par noeud
    - Plus d'erreur IMPALA : meilleur contrôle de la consommation mémoire
- ★ Plan d'action à intégrer Sprint #1
  - Tunning Impala / HBase / HDFS
  - Changer format de stockage (Avro vers Parquet) ?
  - Echelonnage vertical
    - RAM : +10go = +77 € par noeud = +231 € par mois
    - Disque rapide = +36 € pour 450 go
  - Ajout de noeuds ? Voir coûts et problématique Sahara / Impalad

Restitution

- ★ Proposition initiale : Zeppelin + D3JS
  - Expérimental / Solution très jeune
  - Zeppelin orienté Data Preparation
  - D3JS n'est pas intégré nativement dans Zeppelin
    - Trop de développements à prévoir
    - Pour notre besoin il fallait implémenter Angular JS
- ★ Solution alternative validée : Qlik Sense
  - Complètement adaptée au premier Use Case (Mesure des escalades en mode tableau de bord)
  - S'intègre parfaitement dans une approche Master Product (Helpdesk)
  - Nous permet d'envisager une solution AD HOC pour nos clients (reporting + Data Exploration)
  - Micropole partenaire expert de la solution (contributeur et partenaire privilégié)
  - Communautaire : Micropole contributeur composant Chord (D3JS)
  - Une problématique de volume de données et coûts hébergement : réglé par la mise en place d'une nouvelle machine sur la plateforme CloudWatt
  - Coûts de licence à estimer / Point à planifier avec Qlik et Micropole en septembre

En résumé

- ★ Cloud : OK
- ★ Scalabité horizontale : A tester Sprint #1 (point performances)
- ★ Performances : A améliorer Sprint #1
- ★ Hadoop : OK !
- ★ Ingestion : OK
- ★ Architecture Lambda : OK
- ★ SQL Low Latency : OK
- ★ CUBE : A tester (Sprint #1 ?)
- ★ Data Preparation / Zeppelin : A faire (Sprint #1 ?)
- ★ Full OpenSource : Tout sauf Qlik
- ★ SCRUM : OK

En résumé

- ★ Organisation DataLab
  - Lieu : Box FB218
  - JIRA Agile
  - Product catalog initialisé
  - Creative Board initialisé
- ★ Architecture EDH - Jean Christophe et Samuel
  - Socle commun en cours de construction
  - Dossier d'architecture initialisé

En résumé

- ★ Cloud : Galère…
  - Sécurisation de la plateforme assez longue, mais indispensable
  - Ouverture des flux complexe : on passe par le hotspot wifi visiteurs !
  - Scalabilité horizontale : A tester Sprint #1
- ★ Ingestion : Rapide (5 jours)
  - 10 tables Isilog chargées / historisées
- ★ Temps de réponse
  - Parfois meilleurs que la base de production Oracle
  - Incomparables sur des grosses masses de données
  - Mais un problème en cours lié au dimensionnement / tunning de la plateforme

Préparation Sprint #1

- ★ Planning : du 05/09/2016 au 23/09/2016
- ★ Accueil Qliker le 05/09/2016 par Bertrand et Samuel
- ★ La plateforme est prête et permet au Qliker de débuter
- ★ Un effort à fournir durant le sprint pour améliorer les performances de la plateforme
- ★ Démo planifiée le 14/09

Equipe Sprint #1

Référents métier

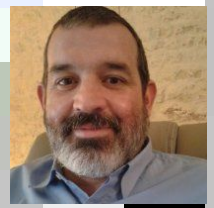
Pilote IRIS

Product Owner
CPU IRIS

Scrum Master

MICROPOLE
Driving Distinction

Spécialiste DataViz
Architecte professionnel
Responsable technique



Consultant
Senior BI

Annexes

Etude Datalake

CR Cadrage architecture du 25/05/2016

Fonctions Supports - Process Outils Qualité

Apache Hadoop EcoSystem

Cadrage architecture du 25/05/2016

Pig

Scripting

Hive ou Impala

SQL Query

Spark

Map Reduce

Distributed Processing Framework

HBase

Columnar Store

Sqoop

Data Exchange

HDFS

Hadoop Distributed File System

Suite échange XEBIA

Hive surclassé par Impala
Pig surclassé par Spark
Spark incontournable
HBase pas indispensable = HDFS directement
Attention à la virtualisation !