# **ORION AD Design Document**

# Tan Peng

# 2018/07/23

# Version 1.0

Contents

[Overall ORION AD Design 2](#_Toc520119741)

[1. Requirements 3](#_Toc520119742)

[a. Streaming cluster 3](#_Toc520119743)

[b. Data Source 4](#_Toc520119744)

[c. Filer 5](#_Toc520119745)

[d. MSSQL 5](#_Toc520119746)

[e. Tableau 6](#_Toc520119747)

[2. Mechanism 7](#_Toc520119748)

[a. Receiving Kafka Message (Done in Nifi) 7](#_Toc520119749)

[b. Data Check (Done in Nifi and Python) 7](#_Toc520119750)

[c. Trigger AD (Done in Nifi and Python) 7](#_Toc520119751)

[d. Analysis (Done in Python) 7](#_Toc520119752)

[e. Generate AD Report (Done in Python) 9](#_Toc520119753)

[f. Save Result (Done in Python) 9](#_Toc520119754)

[g. Monitoring (Done in Tableau) 9](#_Toc520119755)

[3. Code Repo 9](#_Toc520119756)

[4. Tableau Template 9](#_Toc520119757)

# Overall ORION AD Design

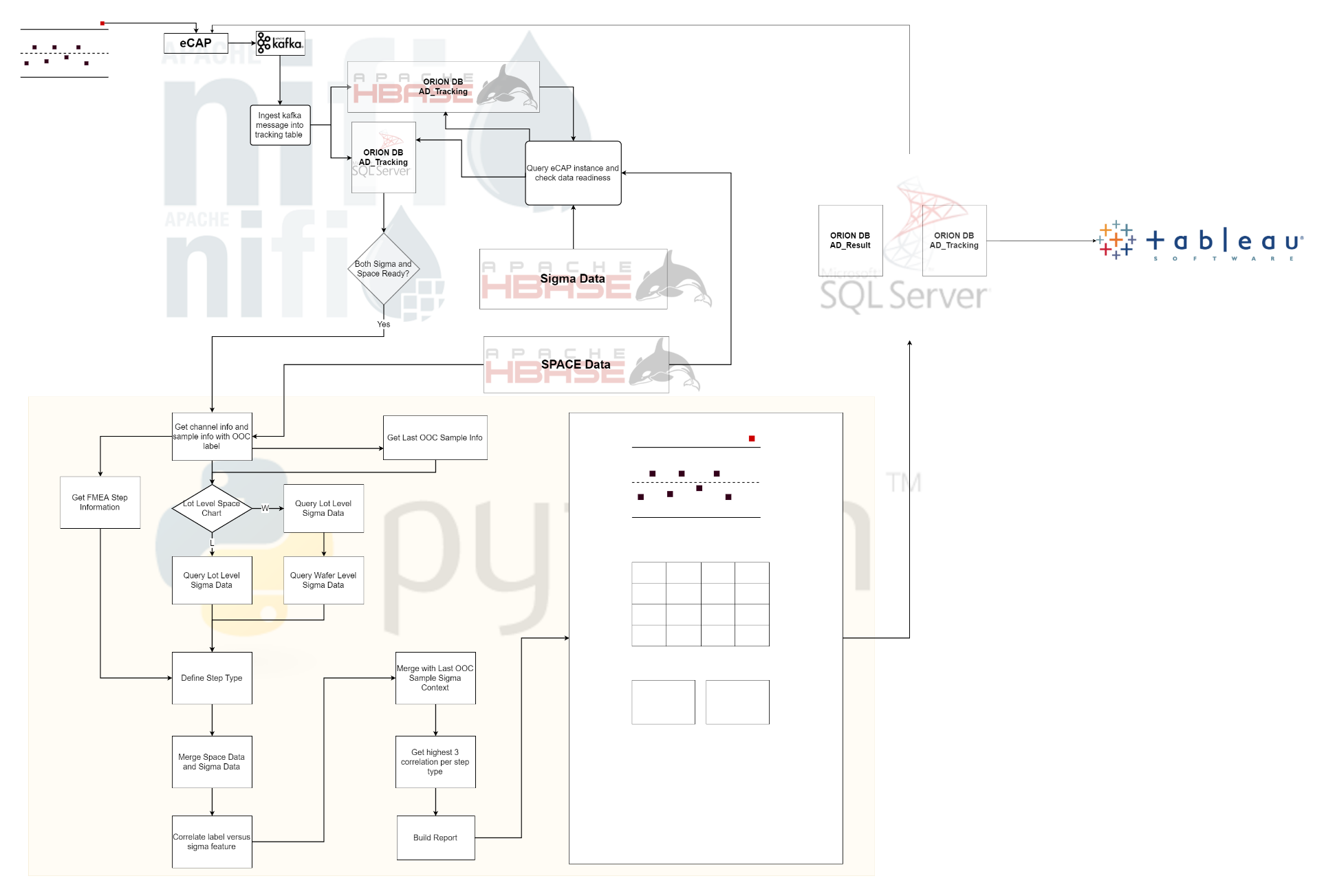


Figure : Overall ORION AD Design

# Requirements

## Streaming cluster

### Python Environment

1. python (version: 3.5)

core python

1. pymssql (version: 2.1.3)

connection library for mssql

1. pandas (version: 0.20.3)

Dataframe manipulate and analysis

1. mu\_thrifthbase (version: 2.0.0)

connection library for HBase

1. seaborn (version: 0.8)

plotting library

1. matplotlib (version: 2.0.2)

plotting library

### Nifi (1.5)

This is the tool to store the data ingestion flow

- Kafka message ingested into HBase/MSSQL

- Data dependency check

- Trigger AD

### Kafka

This is the tool to store eCAP instance messages. Nifi will retrieve eCAP instance details from here.

### Account

1. Hadoop account
   1. This account should be able to run python script from Nifi and save html report into filer
   2. This account should be able to query result from HBase
2. MSSQL account
   1. This account should be able to connect to AD Tracking table in the MSSQL and update row.
   2. This account should be able to connect to AD Result table in the MSSQL and update row.

## Data Source

### eCAP (Kafka)

@TODO Need @Dudy Lim to give more details on this.

### Sigma Data (HBase)

1. sigma\_lot
   1. rowkey schema: ${FULL\_LOT\_ID (first 6 digits reversed)}\_${MFG\_PROCESS\_STEP}
   2. rowkey example: **00000N1.001\_8120-DF PST THK PLYAFLMA MEA**
2. sigma\_wafer
   1. rowkey schema: ${LOT\_ID (first 7 digits only, first 6 digits reversed)}\_${MFG\_PROCESS\_STEP}
   2. rowkey example: **00000M1\_0001-01\_8120-CV PRE STR TEOSEXM7 MEA**

### SPACE Data (HBase)

1. samples\_viol
   1. rowkey schema: ${CHANNEL\_ID (reversed)}\_${CKC\_ID}\_${VAL\_TYPE\_ID}\_${SAMPLE\_ID}
   2. rowkey example: **00000N1.001\_8120-DF PST THK PLYAFLMA MEA**
2. samples\_calc
   1. rowkey schema: ${CHANNEL\_ID (reversed)}\_${CKC\_ID}\_${SAMPLE\_ID}
   2. rowkey example: **000006\_0\_2208122632**
3. samples\_lot
   1. rowkey schema: ${SAMPLE\_ID (reversed)}
   2. rowkey example: **0000001142**
4. samples\_wafer
   1. rowkey schema: ${SAMPLE\_ID (reversed)}
   2. rowkey example: **0000000232**
5. t\_channel\_def
   1. rowkey schema: ${CHANNEL\_ID (reversed)}
   2. rowkey example: **000002**

## Filer

### Storage

A filer is required to store generated AD reports. This is the place to store all html reports per instance. Expected size should be calculated based on instance frequency and report size. Currently for F10N, the size is around 500GB.

Example: /nfs/fsstreamapps/orion/orion\_ad/

### Web Server

A web server is required to view generated AD reports. DBExplorer will get the AD report contents via this URL.

An **symbol link** is required under /var/www/html and link to storage folder. (require root access)

An alias is created for this webserver for faster access.

Example: <http://f10norionad/> or <http://fslhdpiclient01.imfs.micron.com/orion_report/>

Full URL Example: <http://fslhdpiclient01.imfs.micron.com/orion_report/2018-07-23/352493_30_Mean_1083767513.html>

## MSSQL

### Tracking

This table is to track all eCAP instances. Various timestamps are available in this table like

* When AD received trigger
* When AD complete the analysis
* Where report is generated

Example:

Server: FSMSSTEST07\FSMSSTEST07

Database: ORION

Table: ORION\_AD\_Tracking

### Result

This table is to store the AD analysis result. DB Explorer will retrieve final AD analysis result from this table.

Example:

Server: FSMSSTEST07\FSMSSTEST07

Database: ORION

Table: ORION\_AD\_Result

## Tableau

### Monitoring

This dashboard is to monitoring ORION AD analysis performance and check whether any abnormality is found. The dashboard focuses on data readiness and analysis duration.

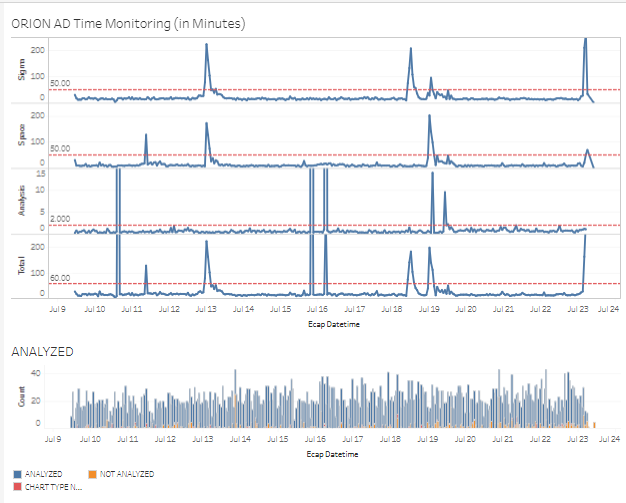


Figure : ORION AD Monitoring Dashboard

Example:

<https://tableau-apac/#/site/F10FEMFG/views/F10N_ORION_AD_Monitoring/ORION_AD_Monitoring?:iid=1>

# Mechanism

## Receiving Kafka Message (Done in Nifi)

* Once eCAP instance information is ingested into Kafka topics, Nifi will help get that information and ingest them into tracking table (both HBase and MSSQL).

## Data Check (Done in Nifi and Python)

* Periodically MSSQL will be queried to get latest records and check whether Sigma data and space data is ready.
  + Sigma and Space data are checked independently.
  + Once data is ready, MSSQL will update corresponding column with timestamp.

## Trigger AD (Done in Nifi and Python)

* Query from MSSQL in Nifi whose Sigma and Space data is both ready and AD is not triggered yet.
* Each instance fire python script to do AD analysis.
* Nifi will control the frequency of parallel triggering.

## Analysis (Done in Python)

* Once AD is triggered, it will start to query SPACE channel information.
  + Sample Specific
    - LOT ID
    - WAFER ID
    - SAMPLE ID
    - UPPER CONTROL LIMIT (UCL)
    - LOWER CONTROL LIMIT (LCL)
    - MEASUREMENT VALUE
    - SAMPLE DATE
    - LABLE OF VIOLATION
  + Channel Specific
    - FAB
    - MODULE (AREA)
    - DESIGN ID
    - CHANNEL ID
    - CKC ID
    - CHART TYPE
    - PARAMETER NAME
    - PROCESS STEP NAME
    - MEASUREMENT STEP NAME
* Identify which sample is last OOC sample
* Identify whether channel is lot level or wafer level
* Query Sigma data:
  + If space chart is lot level, query Sigma lot level data
  + If space chart is wafer level, query sigma lot level plus wafer level data

Lot Level Checked Items:

* + traveler\_id
  + equipment\_id
  + process\_id
  + recipe
  + reticle
  + run\_complete\_datetime
  + mfg\_process\_step
  + part\_type

Wafer Level Checked Items:

* + wafer\_scribe
  + process\_chamber
  + reticle
  + resist
* Merge Sigma data with Space data
* Identify step type (FMEA, process step, measurement step, incoming)
* Pivot data frame into wide format for analysis
* For each column, do the correlation with label.
* Keep features based on last OOC sample running context.
* Find top features (currently is 3 per step type)

## Generate AD Report (Done in Python)

* Query AD Report template, css, js
* Plot SPACE Chart image
* Get FMEA table html
* Get top feature table html
* Plot per top feature
* ADD each section into AD Report template
* Save html into filer
* Get web url link

## Save Result (Done in Python)

* Save URL link and final status back into MSSQL Tracking table
* Save URL into MSSQL Result table.

## Monitoring (Done in Tableau)

* Tableau will connect to MSSQL Tracking table and display ORION AD performance.

# Code Repo

<https://bitbucket.micron.com/bbdc/projects/FDS/repos/orion_ad/browse>

# Tableau Template

<http://collab.micron.com/IS/SGBigDataApp/f10bigdata/Shared%20Documents/ORION%20AD/F10N_ORION_AD_Monitoring_20180723.twb>