

Best practices and lessons learnt from Running NiFi at Renault

Kamelia Benchekroun- Big Data Architect
Renault

Abdelkrim Hadjidj - Solution Engineer
Hortonworks

GROUPE RENAULT



Agenda

- ◆ The NiFi journey at Renault
- ◆ Best practices for running NiFi in production
- ◆ Lessons learnt at Renault
- ◆ Questions & answers

The NiFi journey at Renault

GROUPE RENAULT

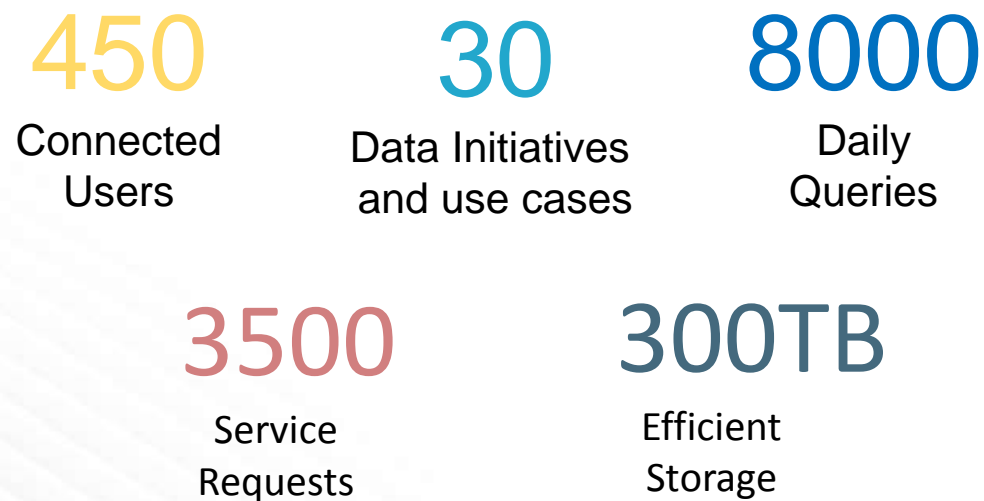


About Us

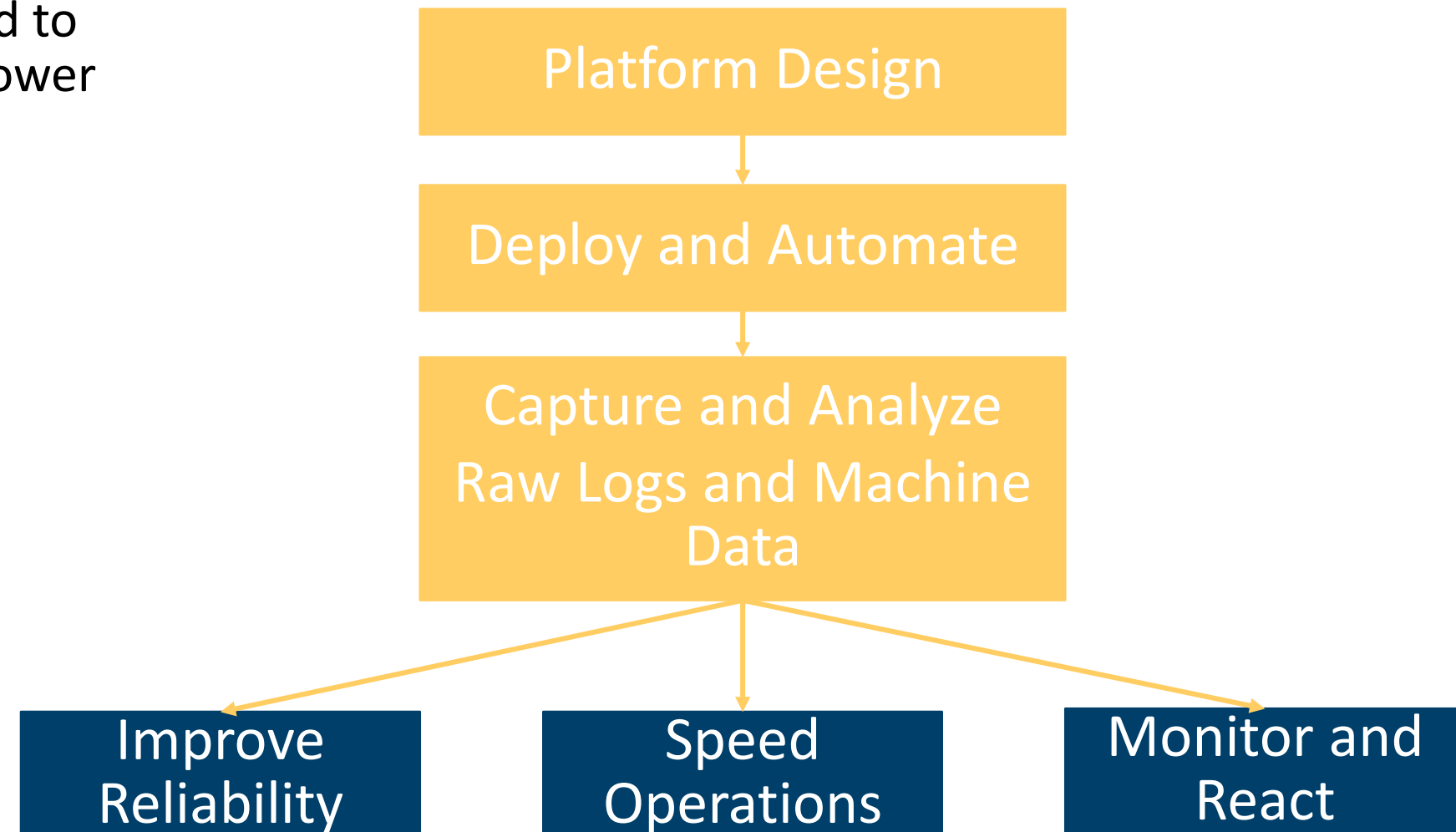
The Datalake Squad:

- A passionate Team who work really hard to deliver solutions and services and empower Data Initiatives at Renault

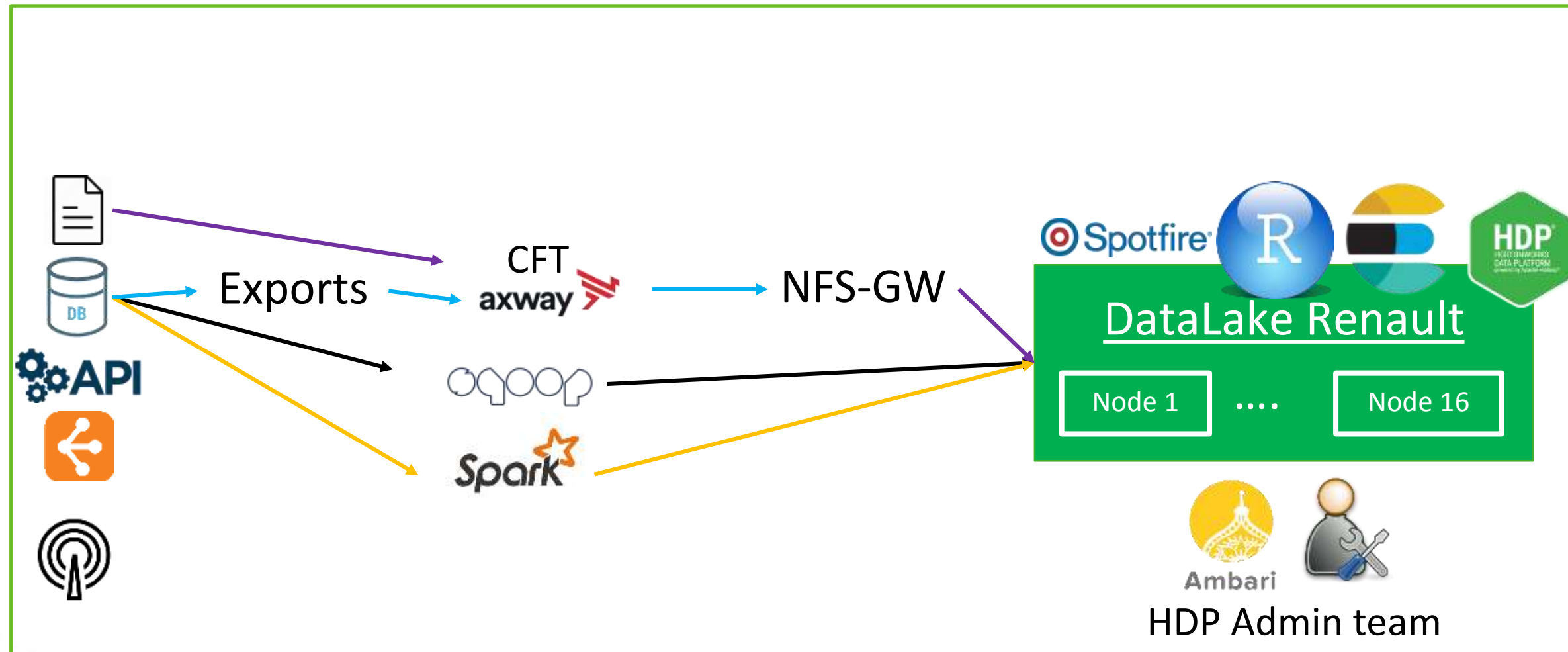
Datalake Platform Metrics:



Datalake Squad Activities



Data Lake at Renault – the beginning of the story



June 2016

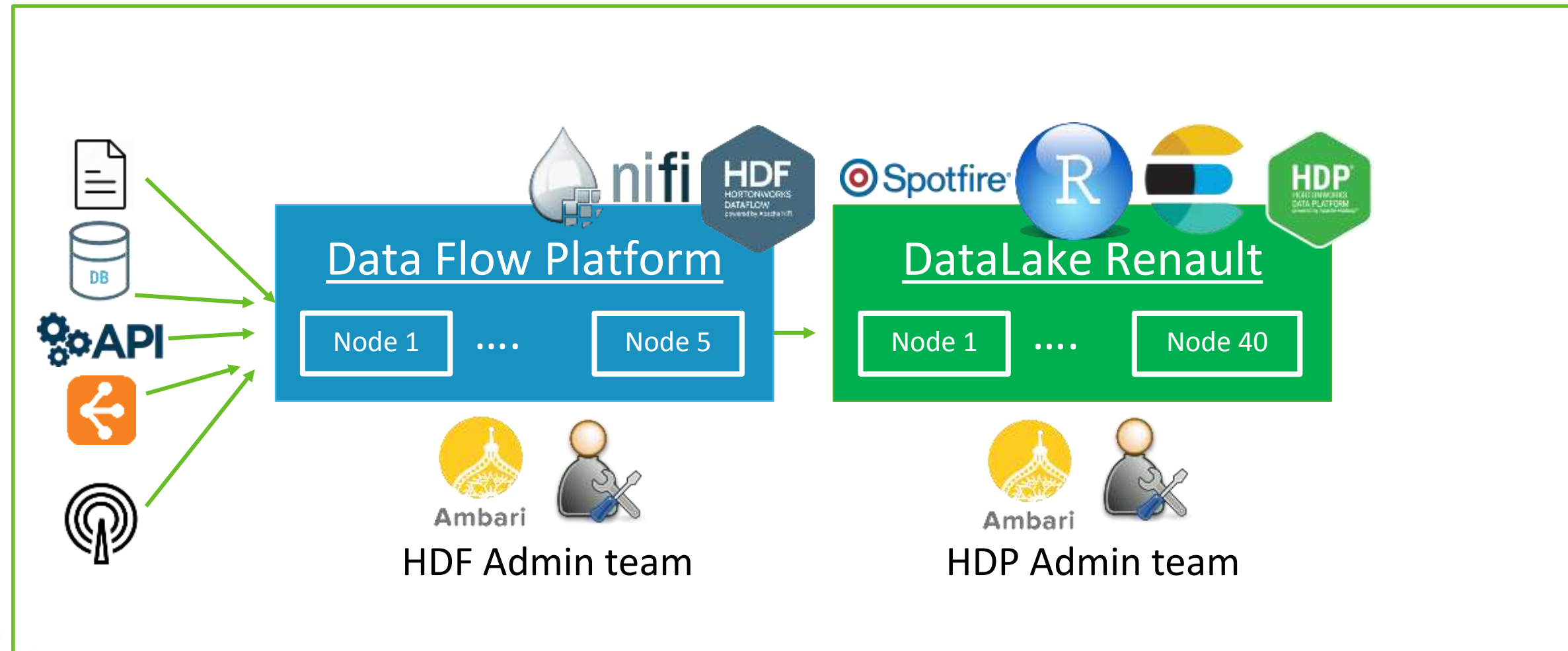
- 10 Users
- Requests by emails, 1 Admin
- Manual provisioning

Big Data Ingestion

GROUPE RENAULT



HDF came in (Q1 2017)



2016

- 10 Users
- Requests by emails, 1 Admin
- Manual provisioning



2017

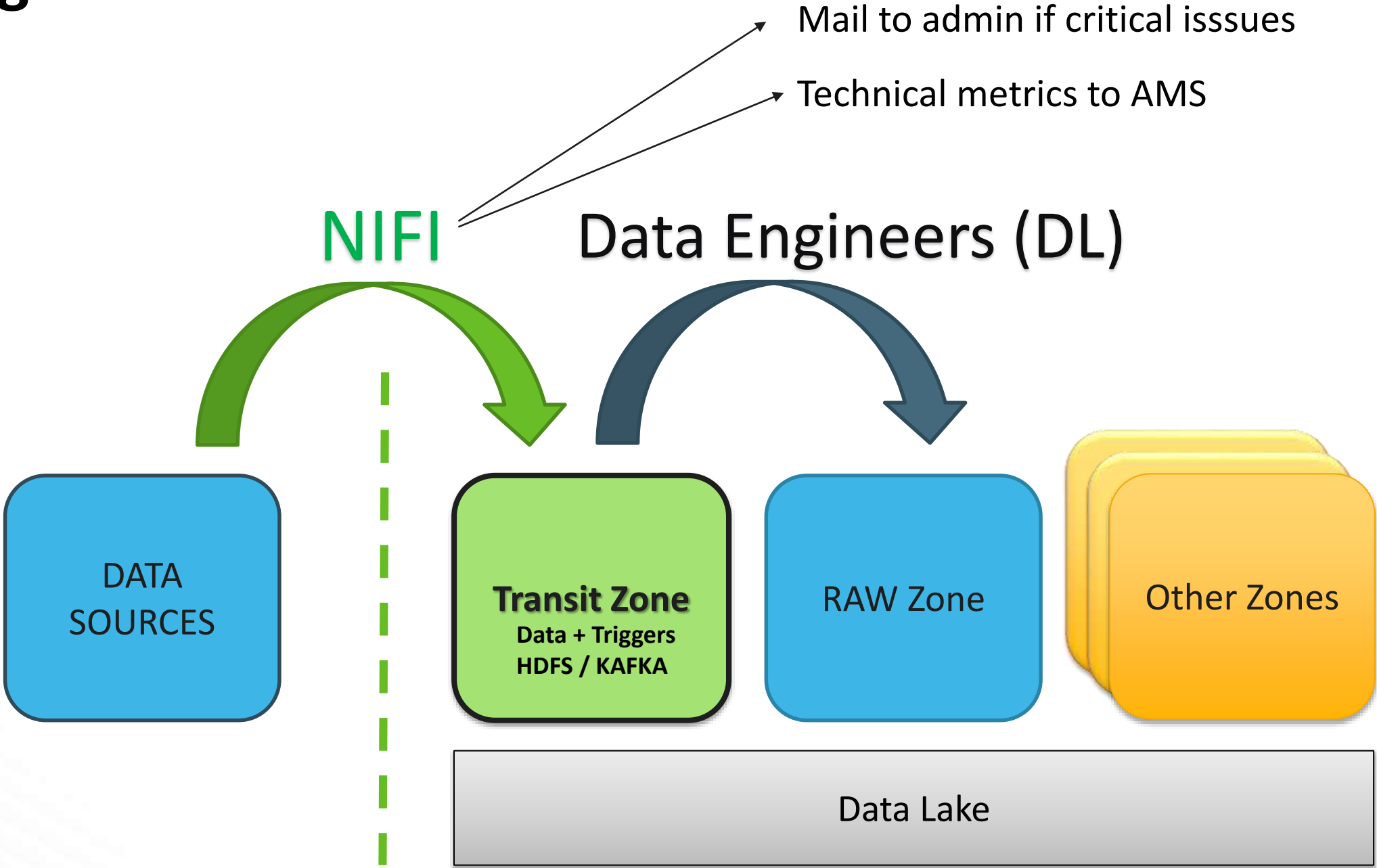
- 200 users
- Jira portal, Admins/devops
- NiFi, Automation (Jenkins for HDP)



Today

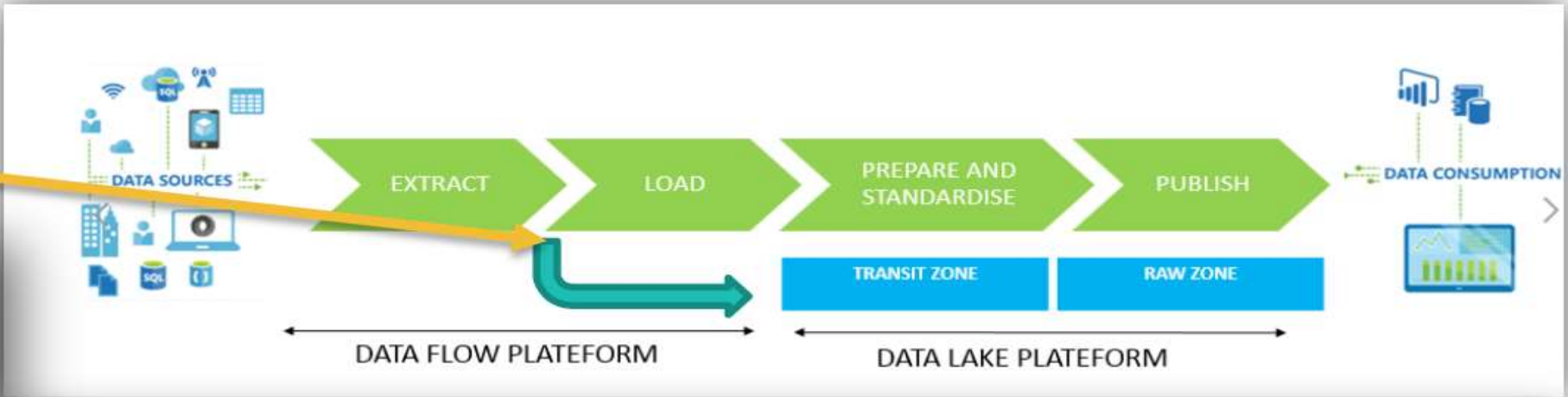
- 300 – 500 active users
- 7000 query per day
- 40 data source in production

Data Ingestion Process



Data ingestion workflow

NIFI



Datalake Core Services

DLK Platform Support

Welcome To The Datalake Platform Services! You can raise a request or report an incident from the options provided.

1. Data Access Requests

2. Provisioning Requests

3. Data Flow Requests

4. Common Requests

5. Reporting an incident

6. Platform usage

7. Logins and Accounts

Export data from DLK
Request data export from datalake

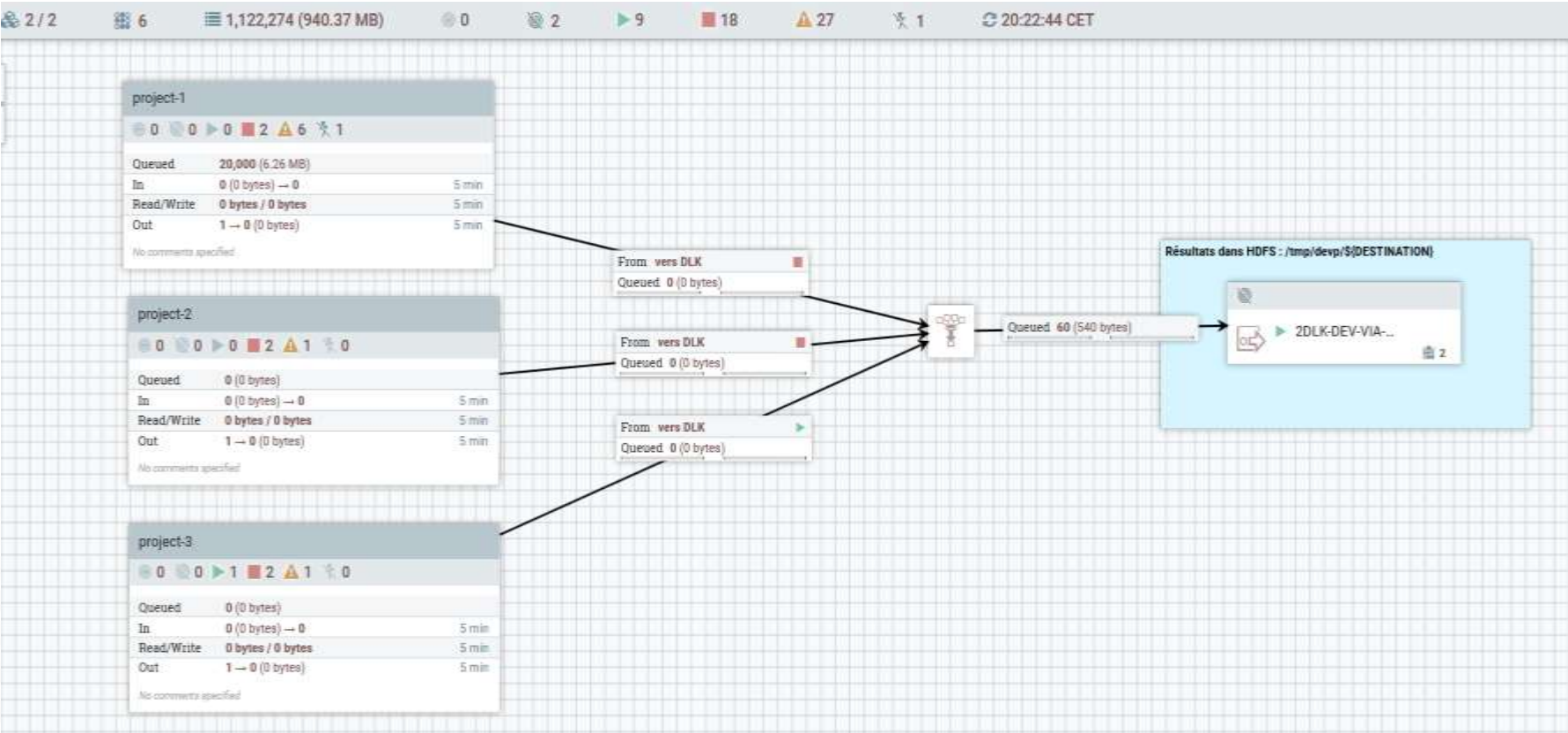
Import data to DLK
Request a new data source import

New CFT connection
Request for a new CFT connection

What	How ?	Who ?	When ?
Extract and Load Data Sources	By submitting a request to the Big Data Squad : <div></div> /servicedesk/customer/portal/1/create/18 So that Extraction can be implemented in the Data Flow Platform (DFP)	The Data Flow is scheduled by the DFP Admin at Squad Team	Once the data sources prerequisite are ready. A One shot data offload is launched, Then, when the customer is satisfied with the result. The Frequently import is set.
Publish Data Sets in the Raw Zone	By releasing the Data Pipelines into Production	The Data Engineer is responsible for the Go Live The Datalake Admin is responsible for Auditing Data Format and Usage	Once the Business or Data Analyst confirm Data accuracy
Prepare and Standardize	By developing Data Preparation and Standardization Pipelines. Using HQL queries and Spark applications	The Data Engineer is responsible for developing Data Pipelines	Once Data is loaded by the Data Flow Platform

Dev to prod testing

Dev



S2S



Prod



/tmp/devp/\${DESTINATION}

NiFi Value for Renault

90% of data
ingestion since 2016

One Platform for
all data sources : Files,
DBs, API, Brokers, etc.
Offload CFT

100 active data
flows, + 2000
processors in
production,.

Accelerate time to
insights, use case
development and
improve monitoring and
governance

Also used **export**
data from Data Lake
to other
systems/sites/Cloud

Enables **new use**
cases connected
plants, package
tracking, IoT

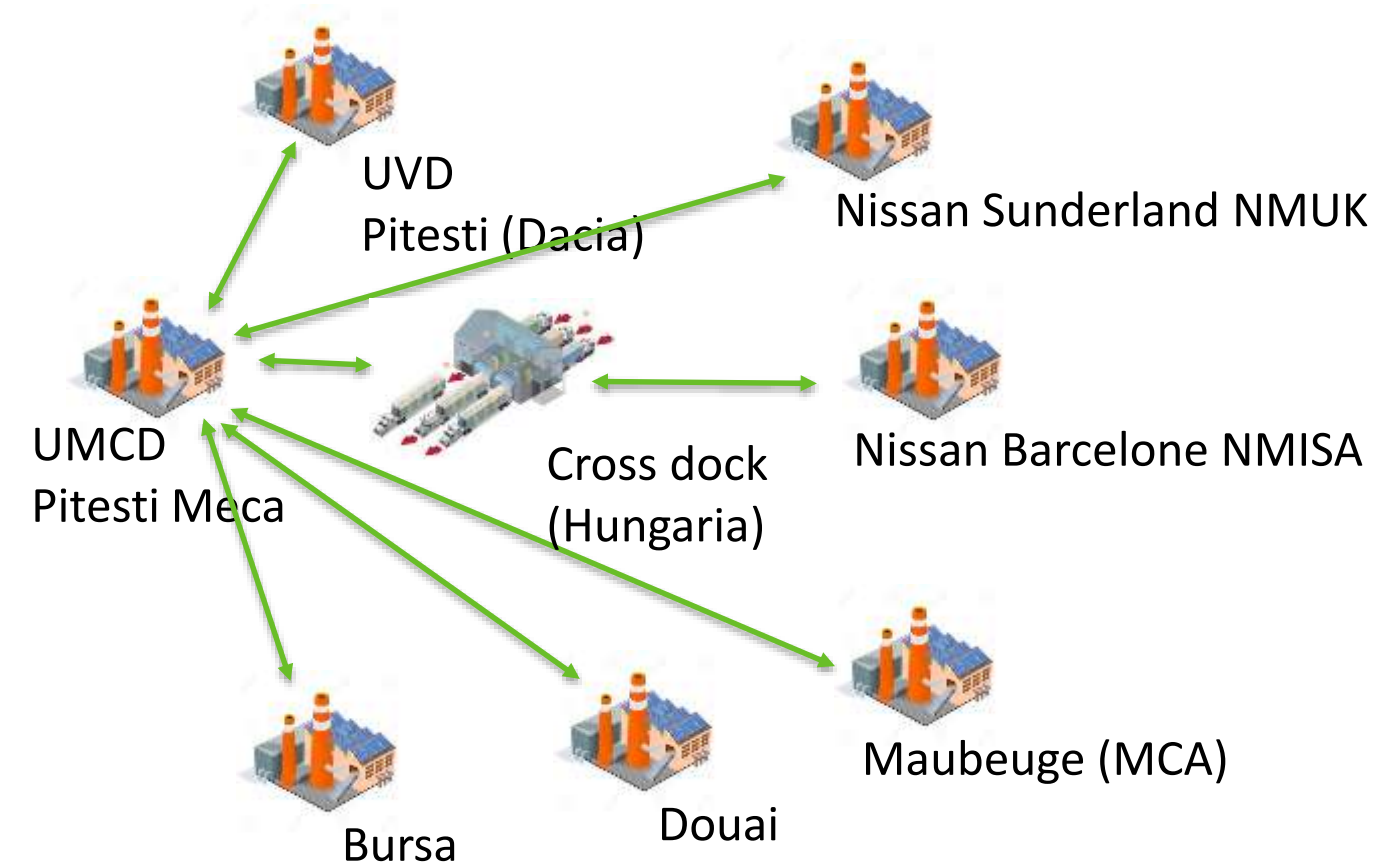
Packaging Traceability

GROUPE RENAULT



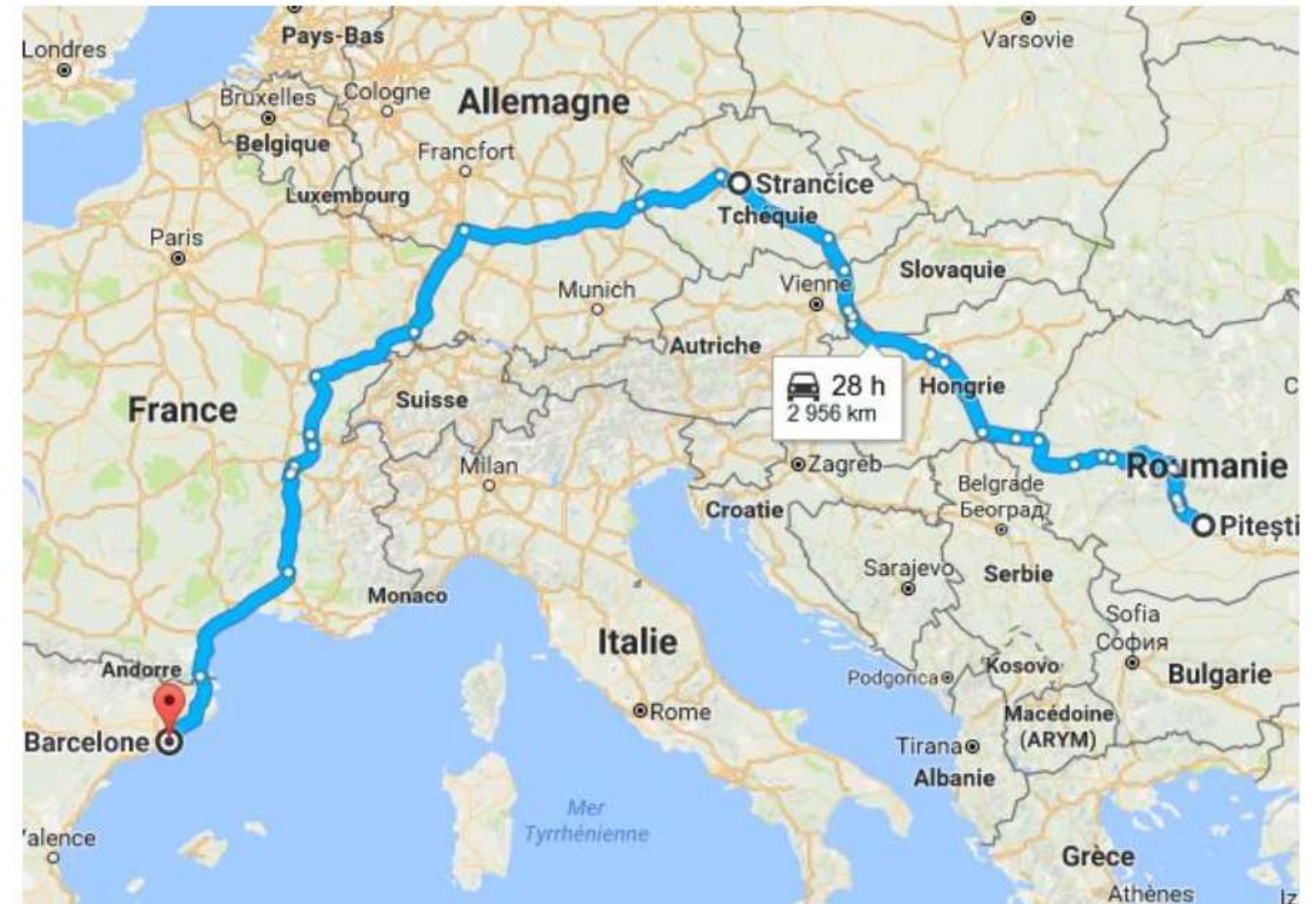
Packaging Traceability

- ◆ POC: 2500 Packages running in the Loop
 - 1 package lost = 800 € (= iPhone)
- ◆ If the solution is generalized : 600k package
- ◆ 2016 Status
 - 400 K€ packaging re-investment due to packaging losses
- ◆ 2017 Status
 - 100 K€ cardboard in January
- ◆ Test Expectations:
 - Reduce cardboard costs and packaging losses
 - Test LoRa technology in an industrial context and Renault activities
 - Validate operational added Value of LoRa technology

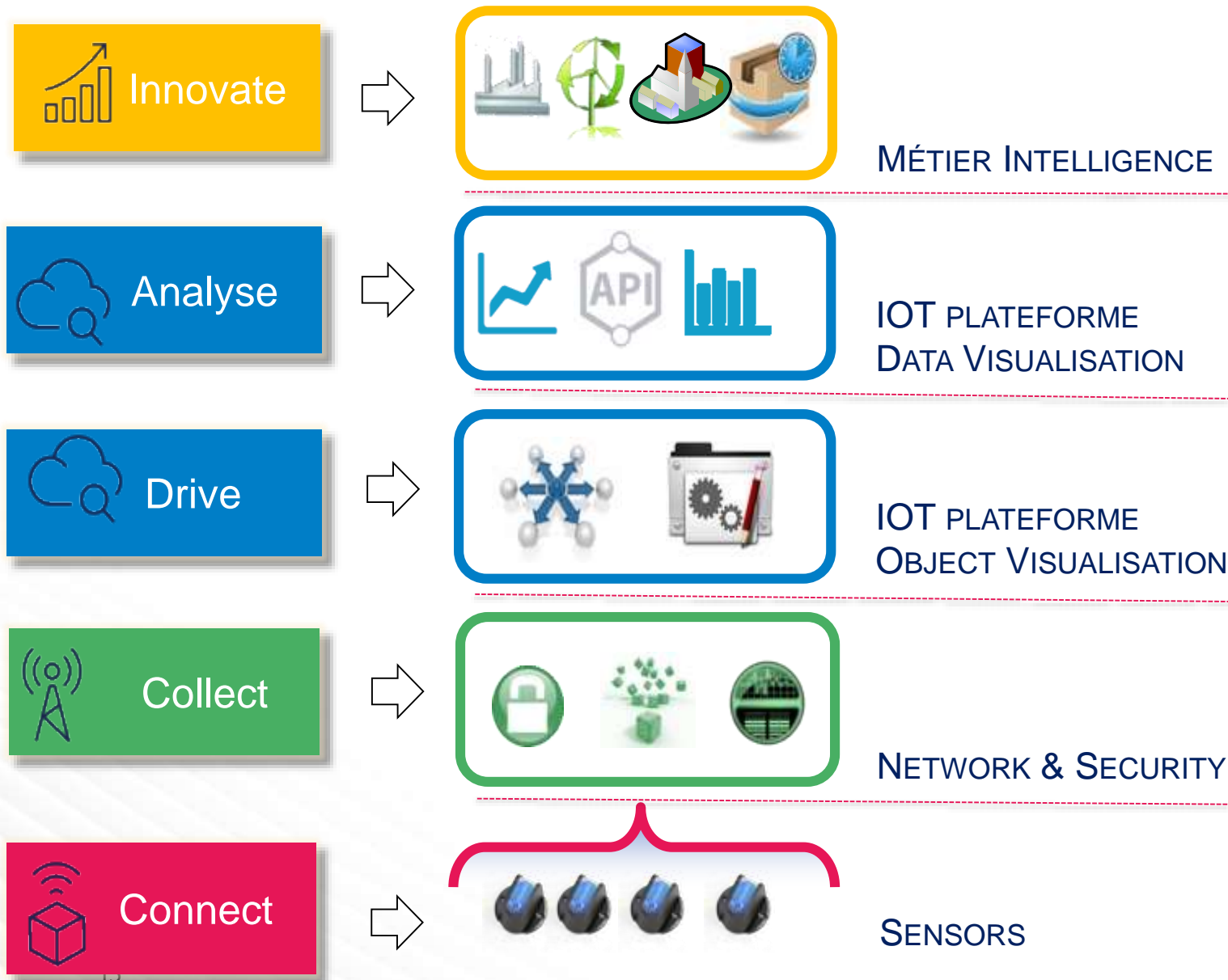


Example flow

- ◆ Pitesti -> Barcelona (full)
 - 3 transports/week
 - 1 truck each time
 - Lead Time : 6 days
- ◆ NMUK -> Pitesti (Empty)
 - once a week
 - 1 truck each time
 - Lead Time : 6 days
 - Cross Dock in Strančice (Tchek Rep.)

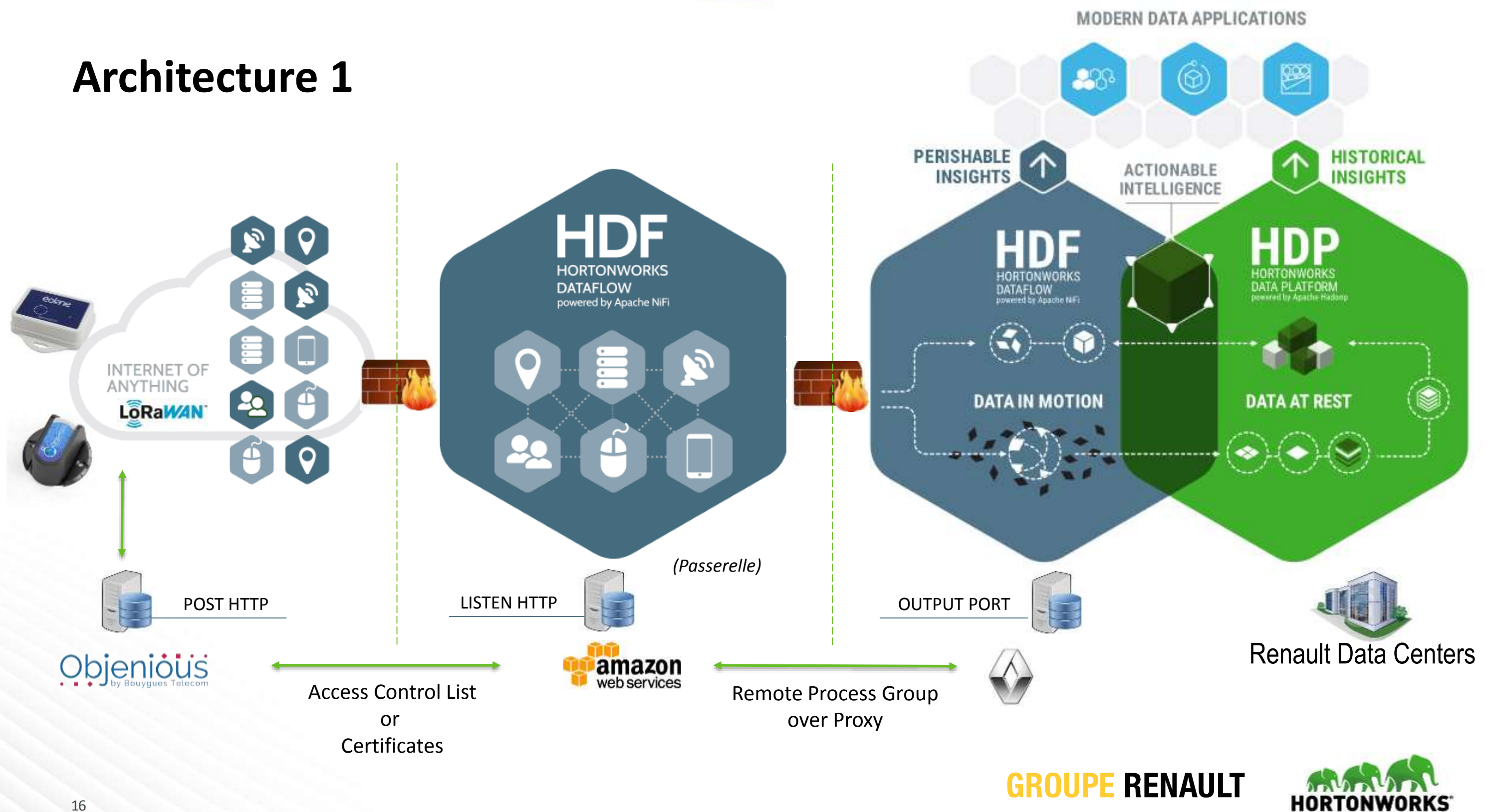


Use case scope

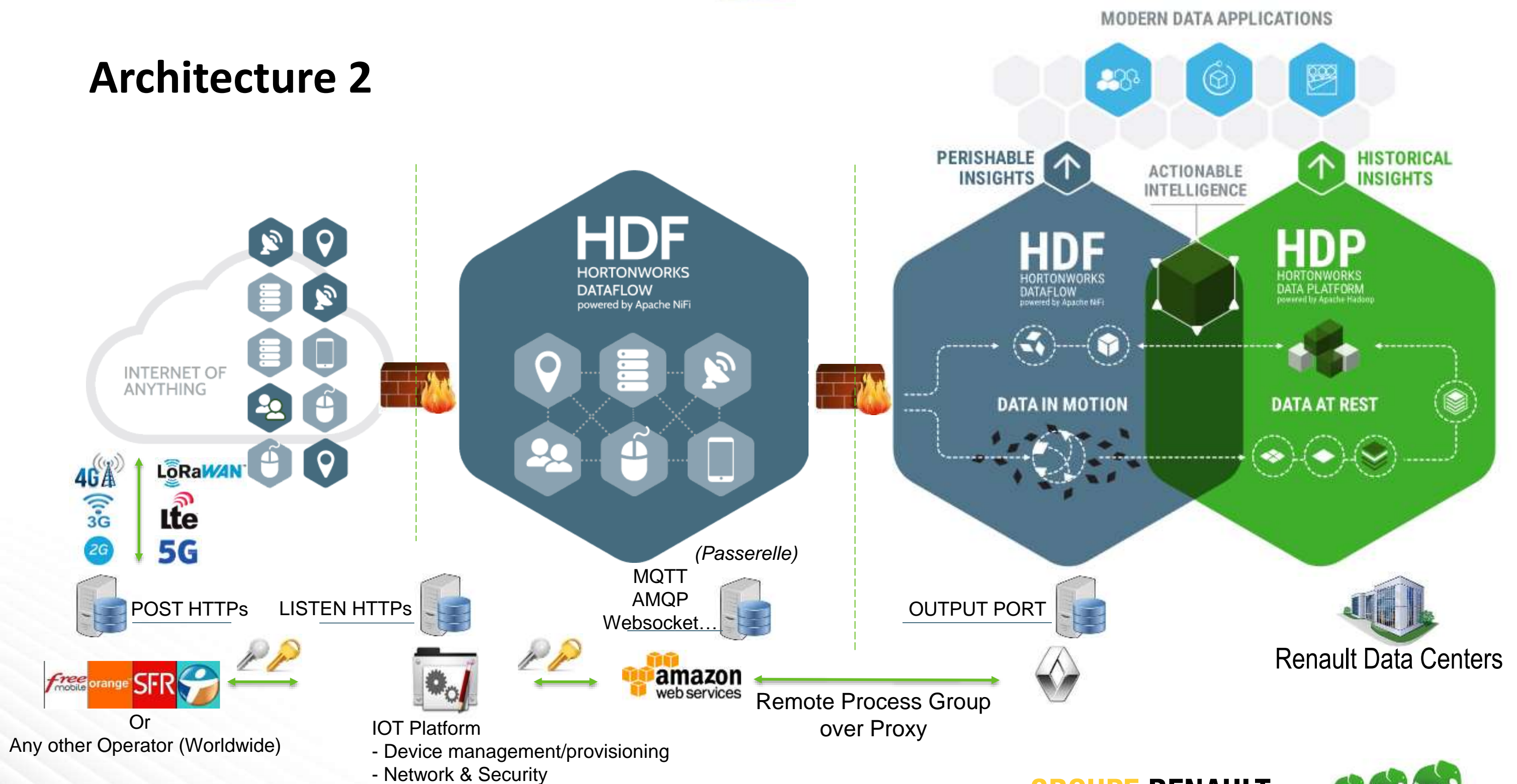


	Not in Scope of Objenious offer			
	2017	S1 2018	S2 2018	WTB
Objenious	Manual Treatment	Manual Treatment	Renault IT (DLK + DFP + IS integration + Analytics)	Renault IT (DLK + DFP + IS integration + Analytics)
		Renault IT (DLK + DFP)		
		Objenious	Objenious	Telecom
				Sensor

Architecture 1



Architecture 2



Reduce cloud communication cost by 50%

Tag Data Decoding with NiFi ExecuteScript Processor

```
var StreamCallback = Java.type("org.apache.nifi.processor.io.StreamCallback");
var IOUtils = Java.type("org.apache.commons.io.IOUtils");
var StandardCharsets = Java.type("java.nio.charset.StandardCharsets");

var flowFile = session.get();

if (flowFile != null) {
  try {
    // Something that might throw an exception here
    // Create a new StreamCallback, passing in a function to define the interface method
    flowFile = session.write(flowFile,
      new StreamCallback(function (inputStream, outputStream) {
        var json = JSON.parse(IOUtils.toString(inputStream, StandardCharsets.UTF_8));

        json["payload"] = [];
        json.payload.push({});
        json.payload[0].timestamp = json.timestamp;

        var splitHex = function (hexValue) {
          //Split each value by 2
          var arr = hexValue.match(/.{1,2}/g);

          var result = {};

          var constant = ((parseFloat(3.6) - parseFloat(2.8)) / 255).toFixed(5);

          for (var key in arr) {
            if (key == "0") {
              result.battery = parseFloat((parseFloat(constant) * parseInt(arr[key], 16) + parseFloat(2.8)).toFixed(4));
            }

            if (key == "1") {
              result.temperature = parseInt(arr[key], 16);
            }
          }
        }
      })
    );
  }
}
```

☐ Set empty string

Configure Processor

SETTINGS | SCHEDULING | PROPERTIES | COMMENTS

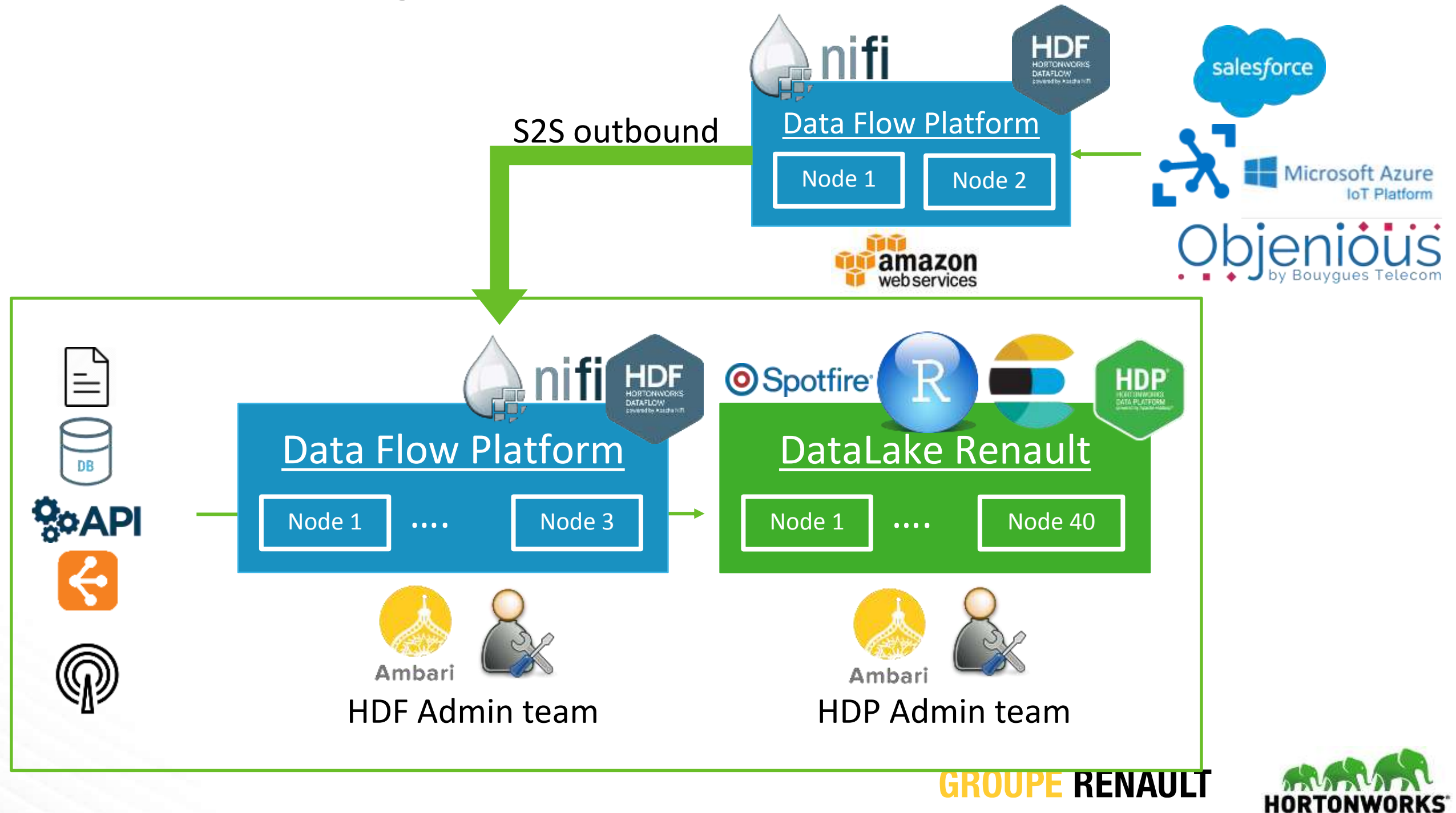
Required field +

Property		Value
Script Engine	?	ECMAScript
Script File	?	No value set
Script Body	?	var StreamCallback = Java.type("org.apache.nifi.process...
Module Directory	?	No value set

CANCEL | APPLY

Raw Data out of sensor : ff1401aa
Decoded Data out of sensor : [{"data":{"battery":3.6007,"temperature":20}}]

Architecture + Cloud ingestion



Connected plants

GROUPE RENAULT



Connected plants

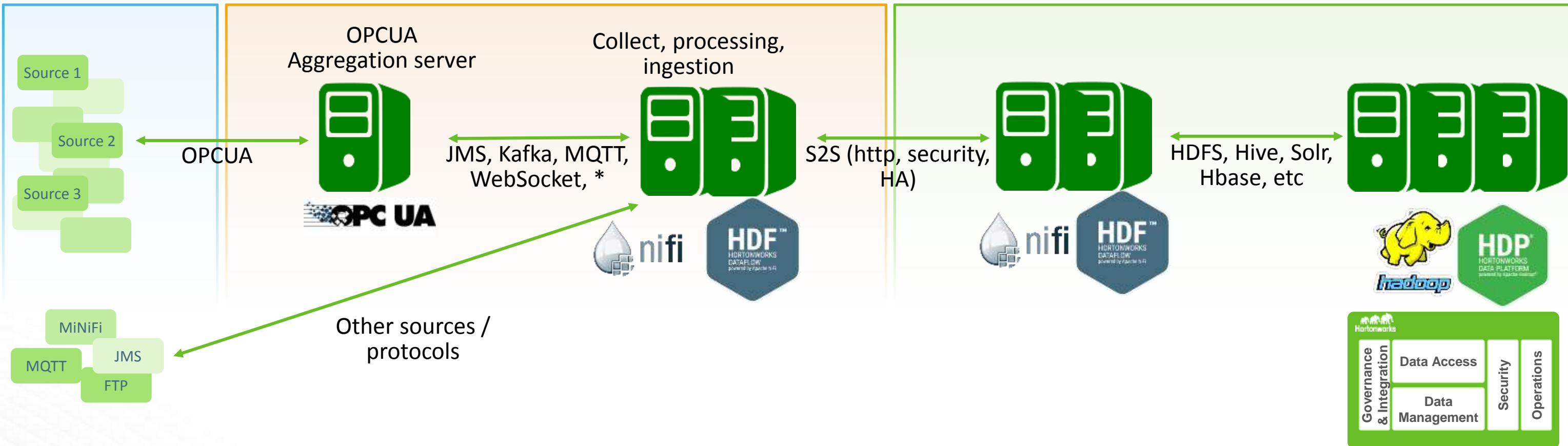
DATA IN MOTION

DATA AT REST

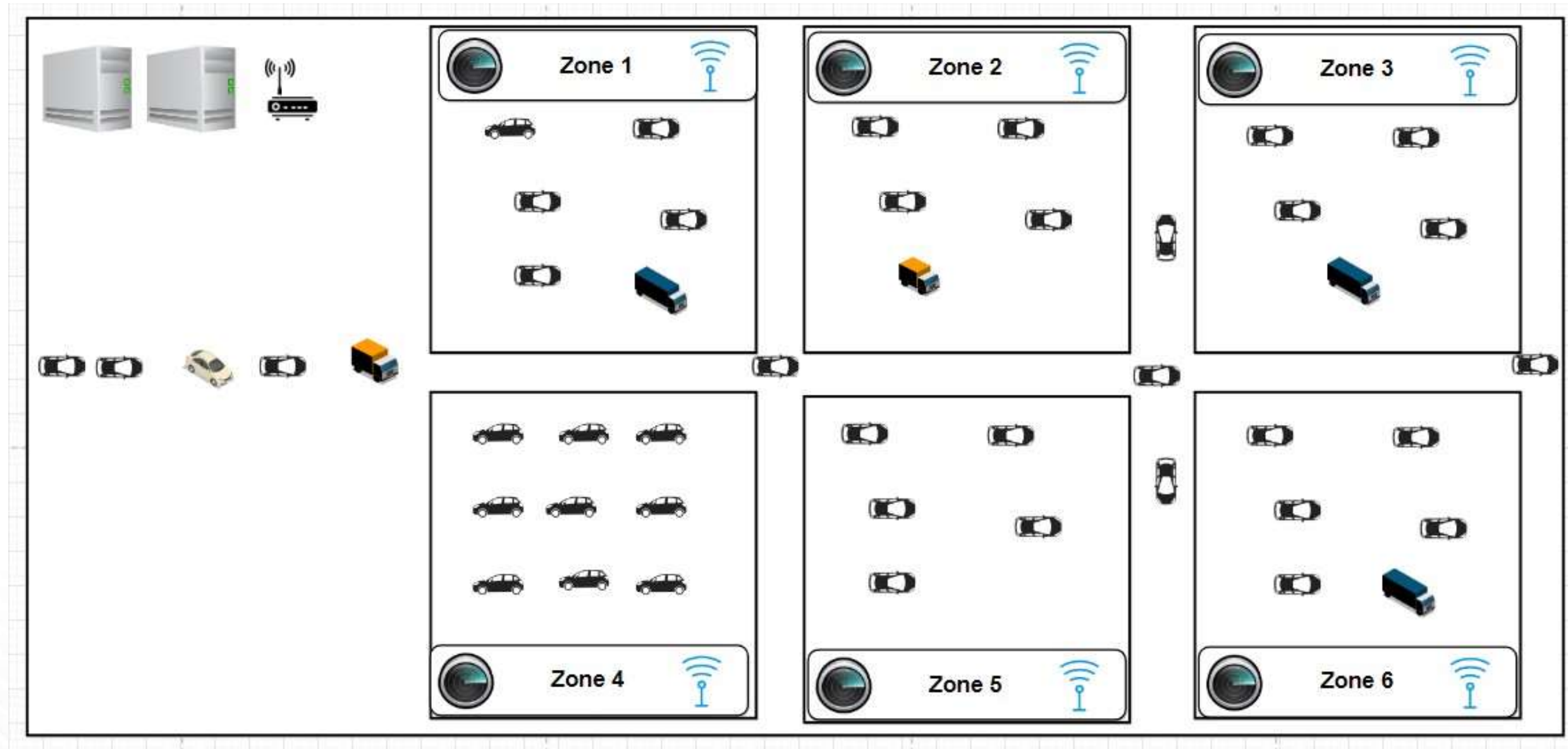
Facility Level

Plant Level

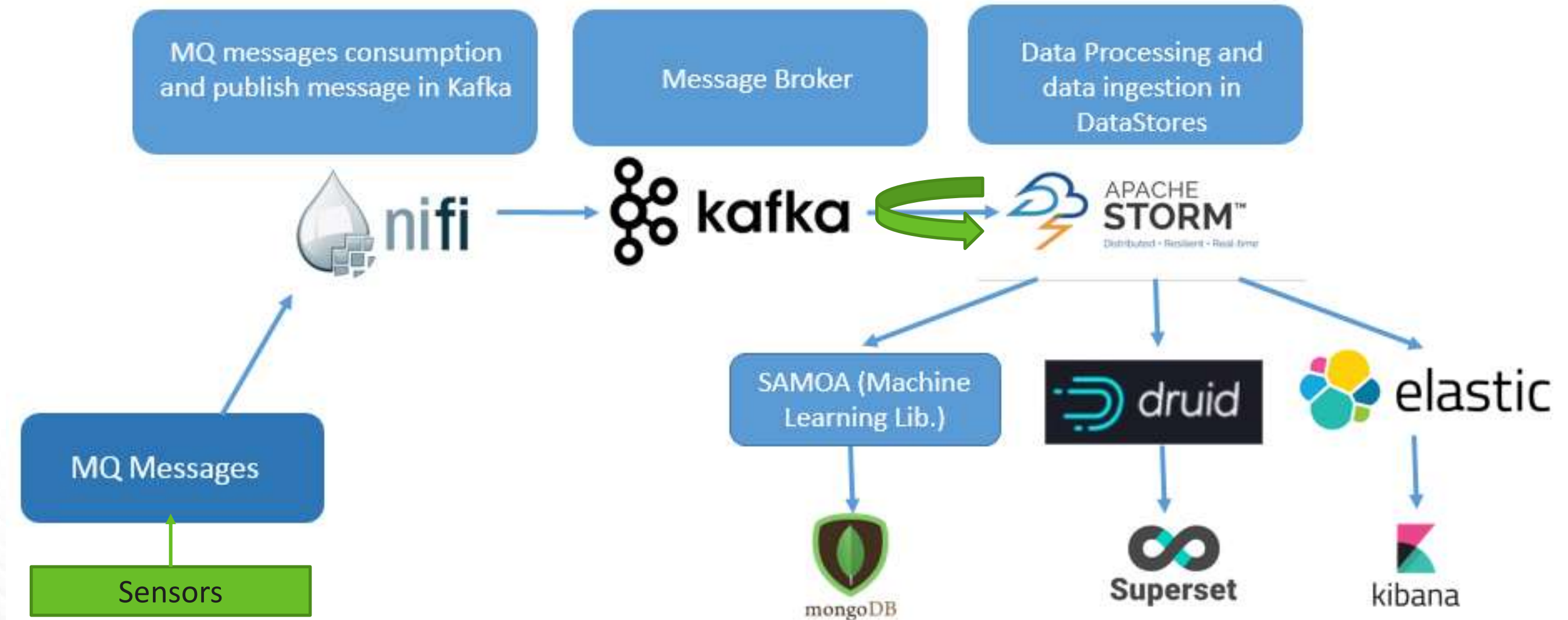
Corporate Level



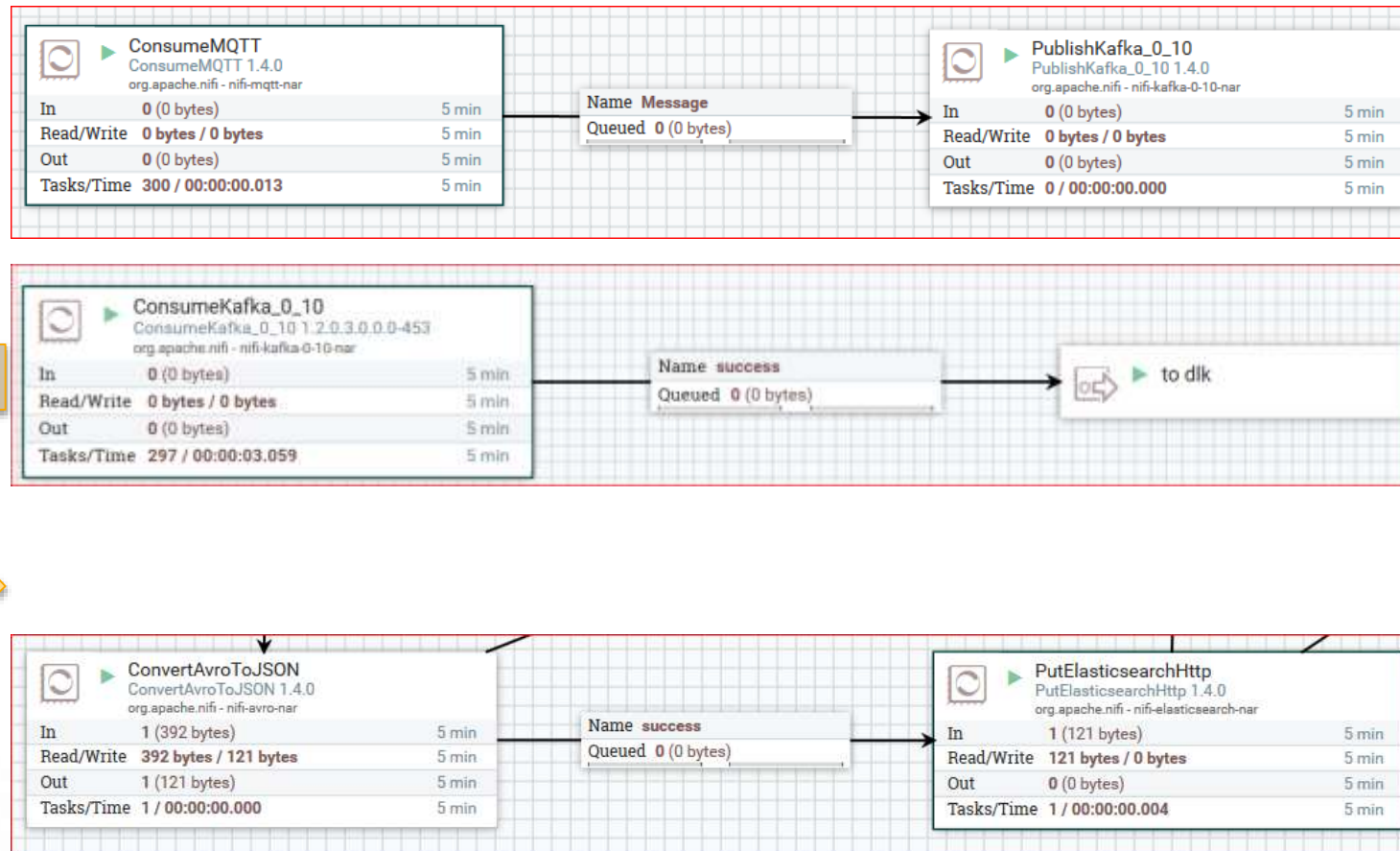
Overview of Plant Assembly Factory UC



Full HDF use case (NiFi, Kafka and Storm)

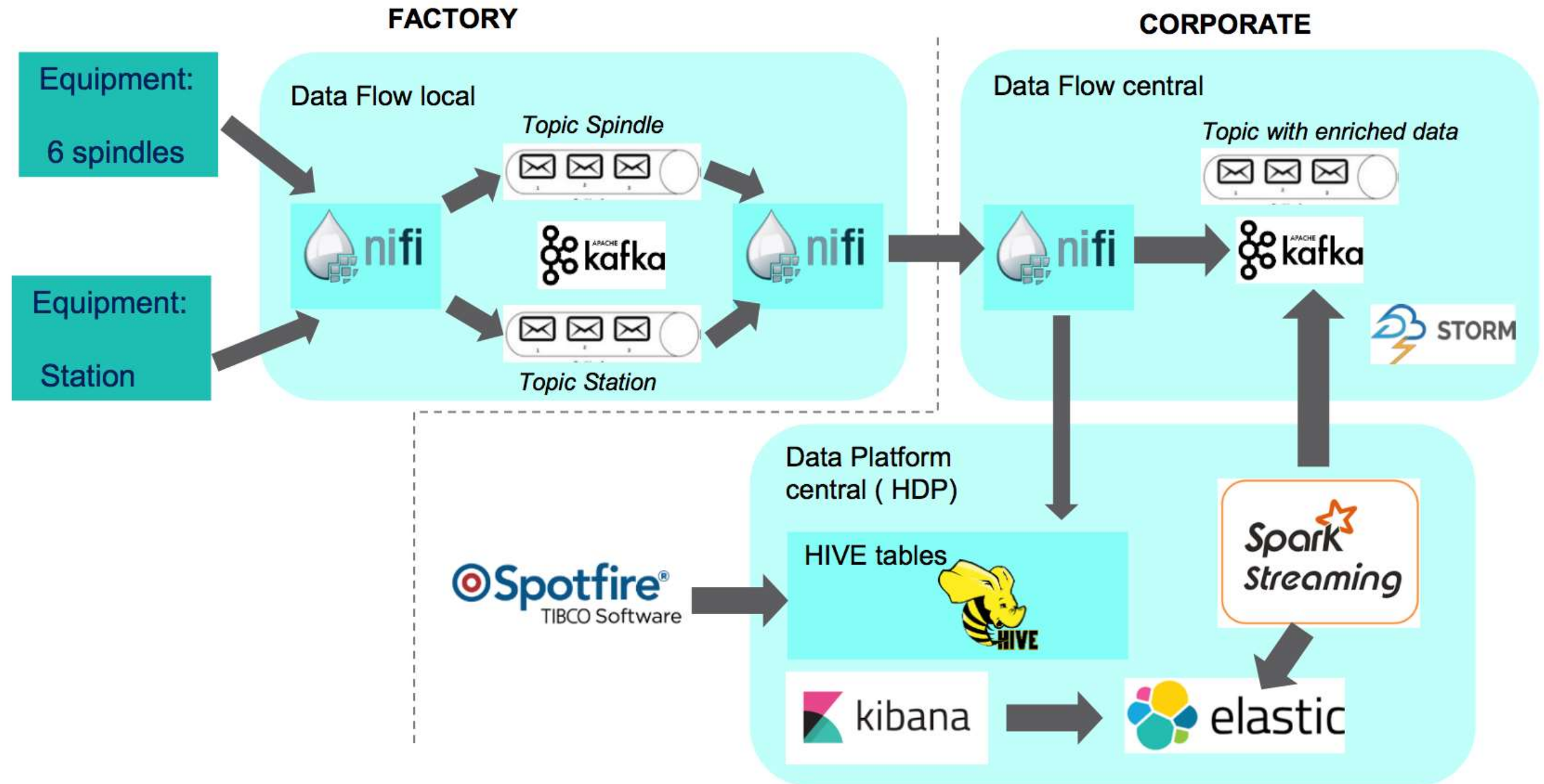


Connected plants

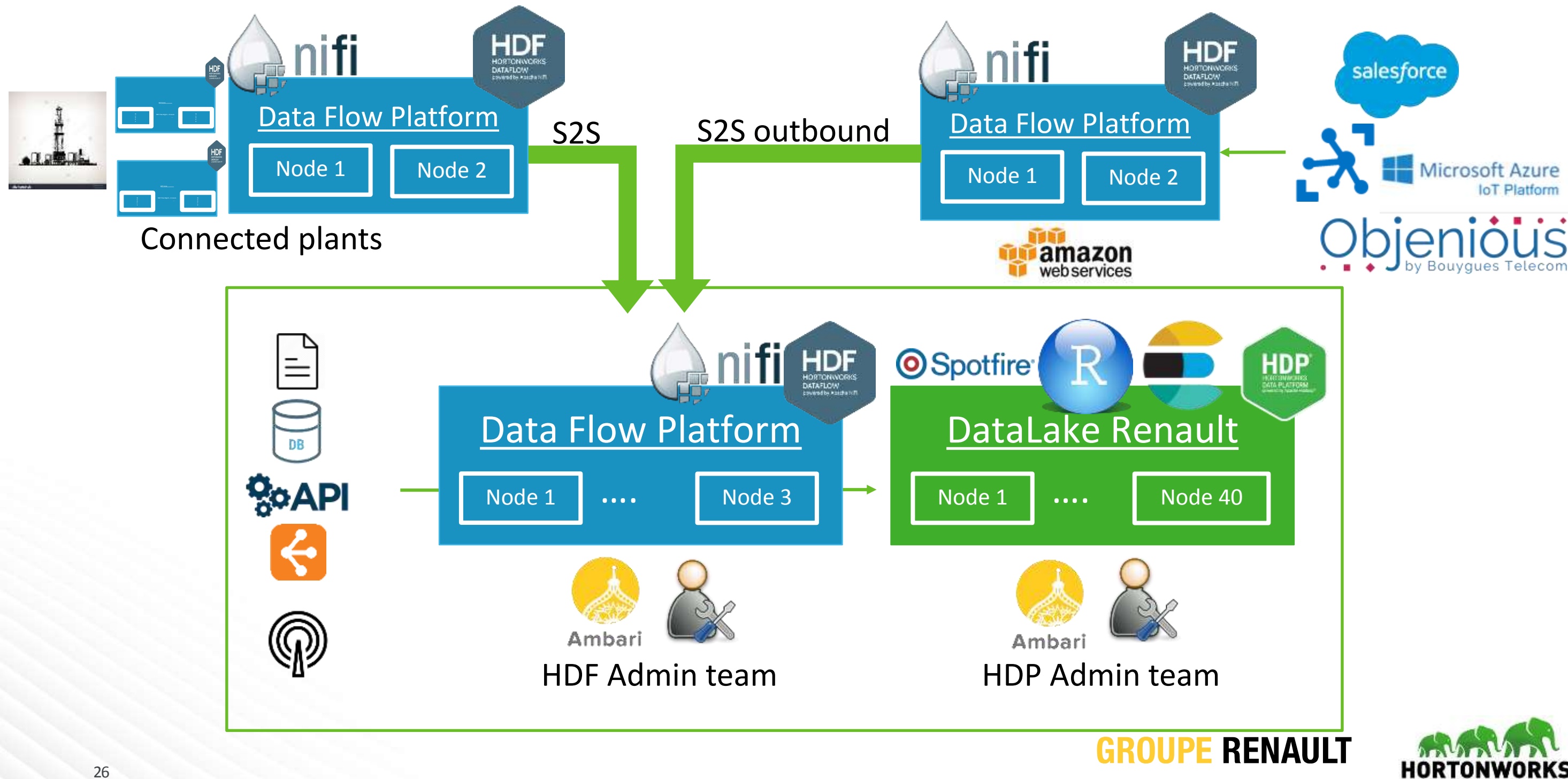


Demo with MQTTTool from
Hand to Data Lake !
Handy, MQTT Broker, NiFi
ConsumeMQTT to Kafka,
NiFi consume Kafka to
ElasticSearch and Hive...

Screwing tools valladolid data analysis



Architecture + Connected plants



Best practices for running NiFi in production

GROUPE RENAULT

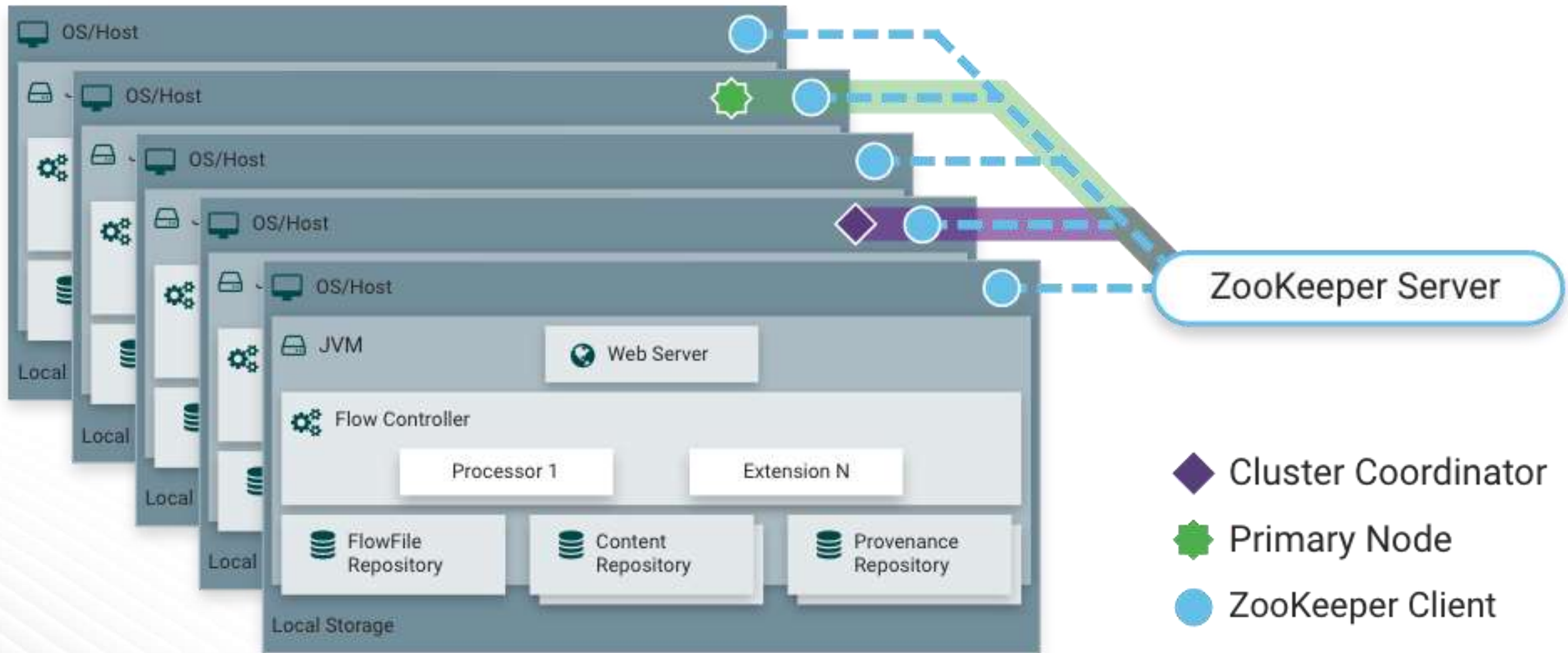


Architecture & sizing

GROUPE RENAULT



Typical NiFi Production Cluster Logical View



Sizing considerations

- ◆ There are baselines for NiFi sizing but NiFi is like a “programming language” !!
- ◆ NiFi resources usage depends on used processors but it’s always IO intensive
 - Use different disks / volumes for the three repositories : flow file, content & provenance
 - For content repository, it’s recommended to have multiple mount points
 - For content and provenance repositories, SSD can provide the best performances
 - Heap sizing depends on the use case and used processors
 - Depending on the workload, we can scale vertically or horizontally
- ◆ Default settings are only for getting started. Tune based on your use case.
 - Thread pool size : Maximum Timer Driven Thread Count
 - Timeouts: `nifi.cluster.node.connection/read.timeout`, `nifi.zookeeper.connect/session.timeout`
 - Pluggable modules: WAL Provenance Repository

Development & deployment

GROUPE RENAULT



Let's consider a simple data flow

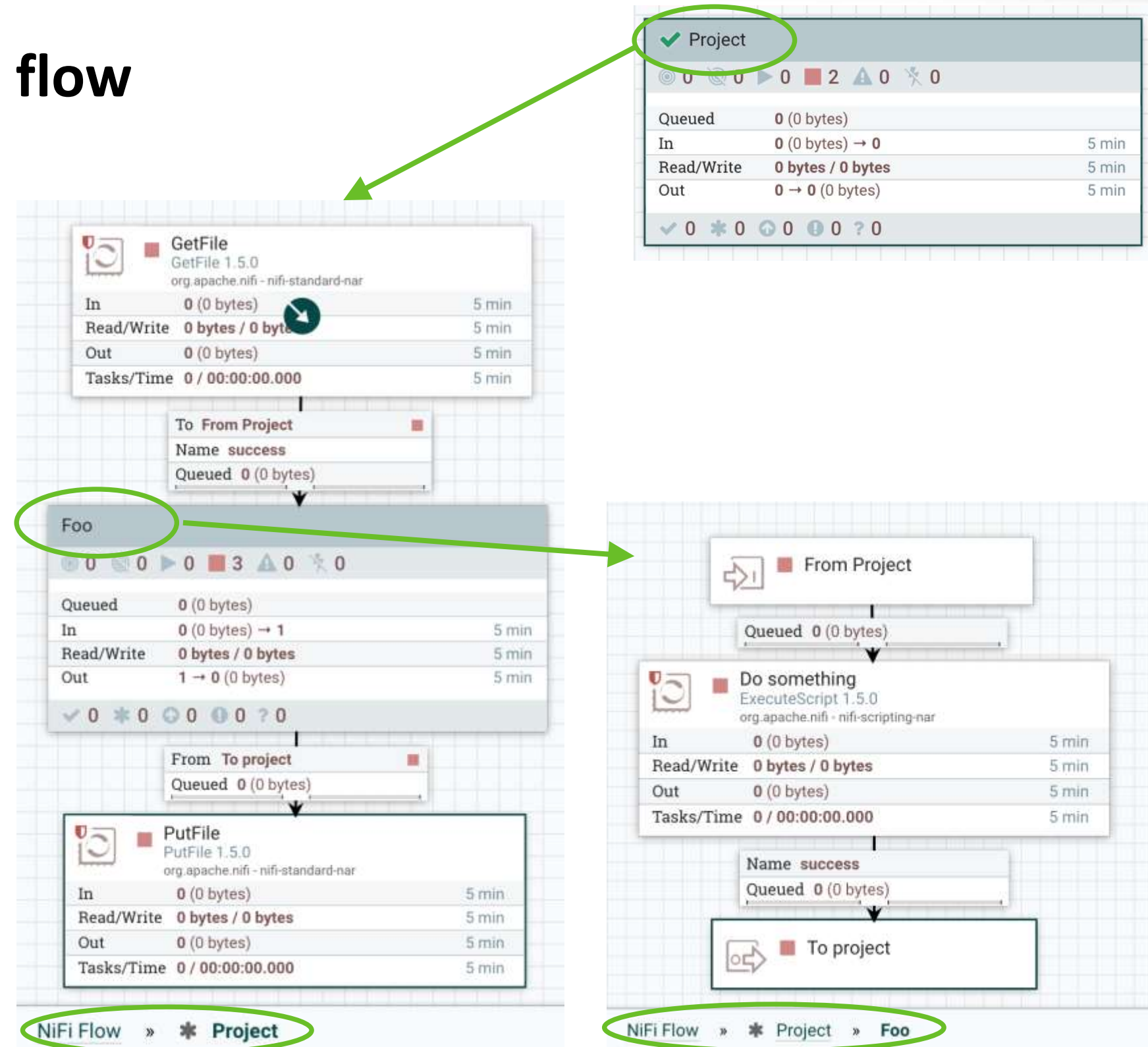
```

Public class Project {

    private int foo(int x) {
        //do something with x -> y
        return y;
    }

    public static void main(String [] args) {
        //some code
        try {
            BufferedReader bufferRead = new
            BufferedReader ...;
            String data =bufferRead.readLine(
            data = foo(data);
            System.out.println(data);
        } catch ...
    }
}

```








Flow development = software development



Programing language

- ◆ Integrated Development Environment
- ◆ Algorithm, code, instructions
- ◆ Functions (arguments, results)
- ◆ Libraries
- ◆ ...

Apache NiFi

- ◆ Apache NiFi UI
- ◆ Flows, processors, funnels 
- ◆ Process groups (input/out ports)   
- ◆ Templates 
- ◆ ...

General guideline

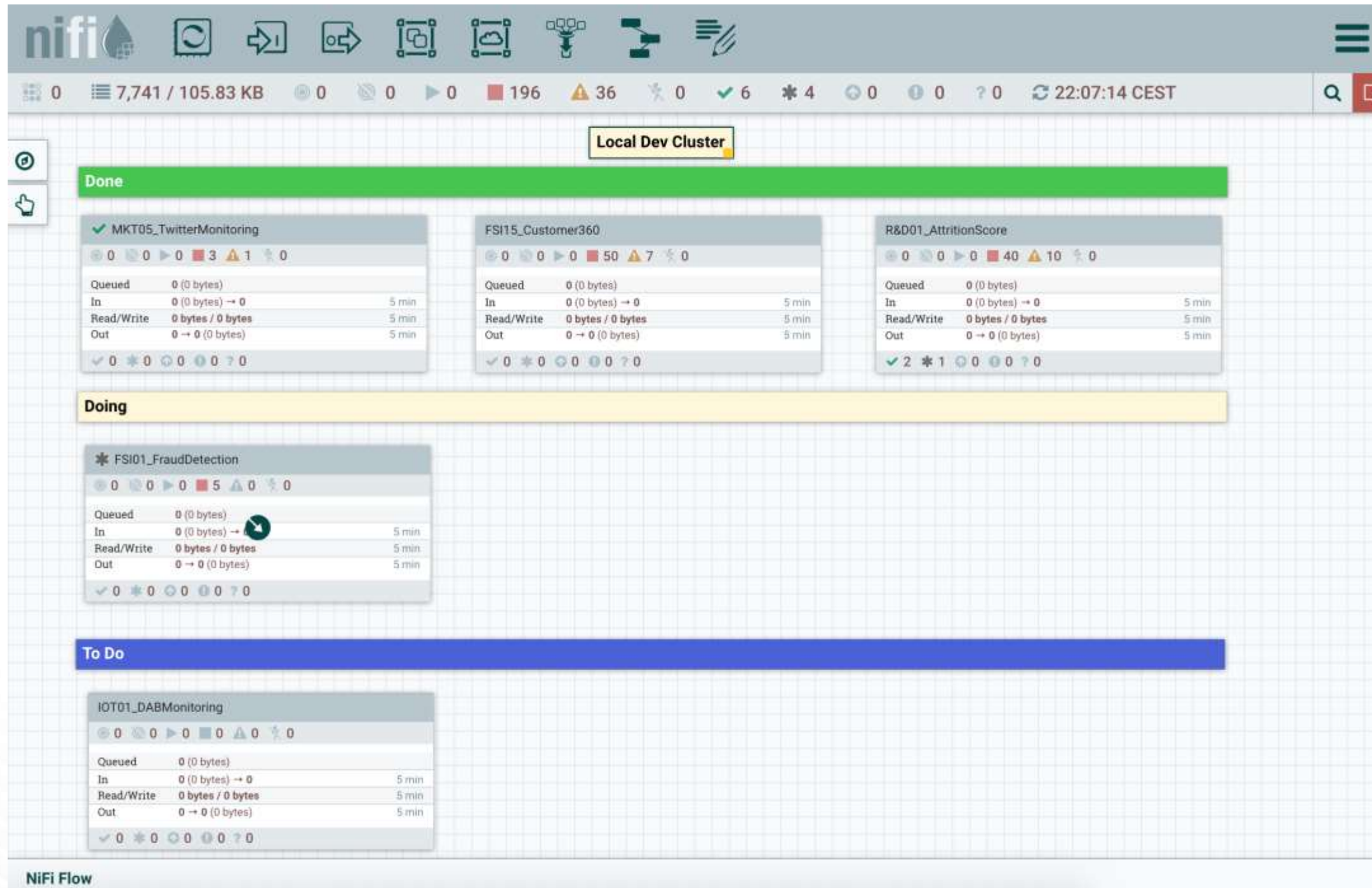
Principle

- ◆ Use separate environments
- ◆ No flows at root level: use a PG per department, BL, project, etc
- ◆ Break your flow into process groups
- ◆ Use a naming convention, use comments (labels, comments)
- ◆ Use variable when possible
- ◆ Organize your projects into three PGs: ingestion, test & monitoring

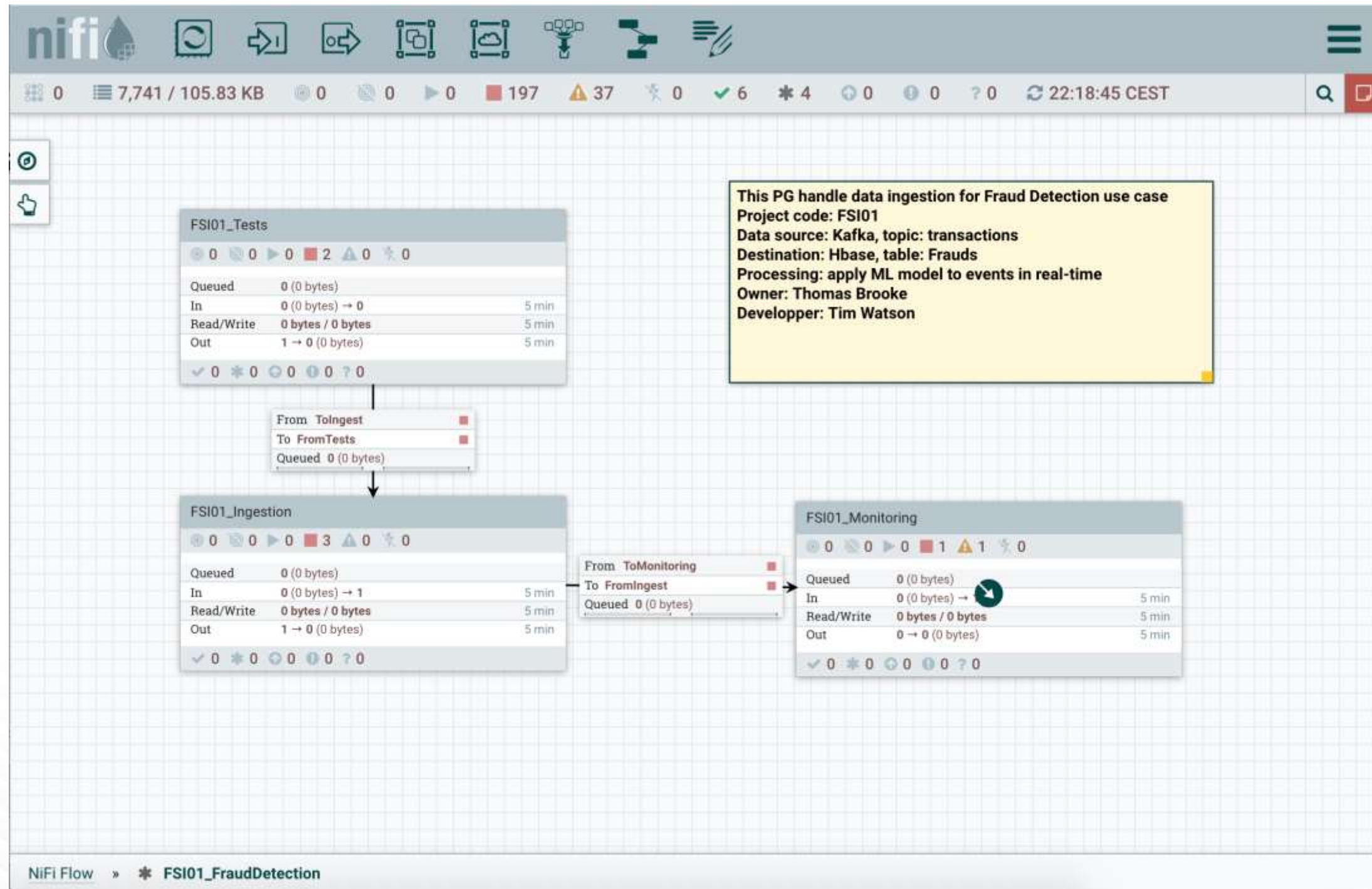
Benefits

- ◆ Performance, security and SLA
- ◆ Easy to secure, everything in NiFi is attached to a PG, heritage
- ◆ Easy to test, update & version
- ◆ Very useful for development/monitoring. Ideally use unique names
- ◆ Promotion (dev > test > prod), update
- ◆ NiFi can generate data for TDD (Test Driven Dev), can collect/parse logs for BAM (Business App Monitoring)

NiFi organization example



NiFi organization example



Know & use NiFi Design Patterns

Fan IN/OUT

List & Fetch

Attributes promotion

Extract, Update Attribute

Throttling

ControlRate, expiration

Funneling

RPG, RouteOnAttribute

Error loops

Relations, Counters

etc

FDLC: Flow Development LifeCycle



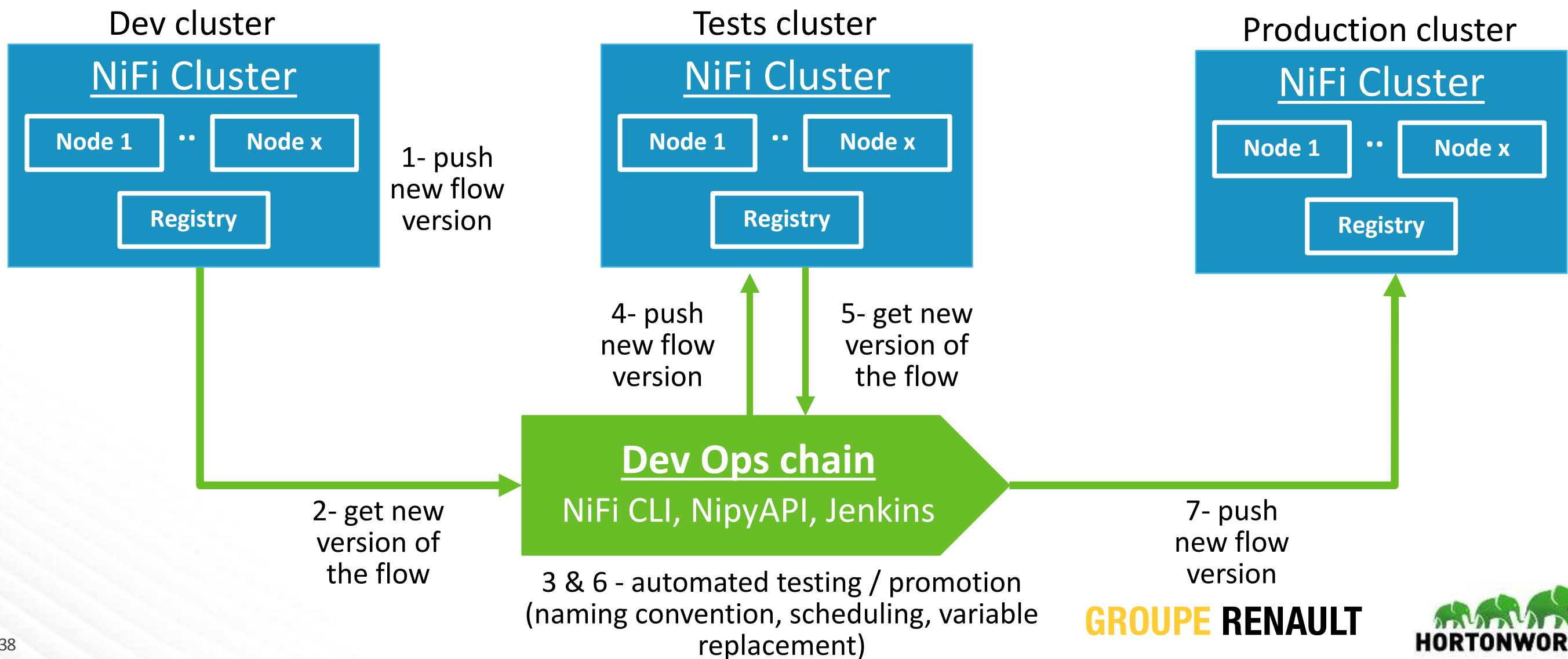
Forget Duplicating Local Changes: Apache NiFi and the Flow Development Lifecycle (FDLC)

TECHNICAL • Data Processing and Warehousing

Thursday, April 19

4:00 PM - 4:40 PM

Room II



Monitoring

GROUPE RENAULT



NiFi Monitoring

Service monitoring

- ◆ Is NiFi service running correctly?
- ◆ Monitor global system metrics such as threads, JVM, disk, etc
- ◆ Monitor global flow metrics such as number of flow files sent, received or queued, processors stopped, etc
- ◆ Solutions
 - NiFi UI
 - Reporting tasks
 - Ambari
 - Grafana

Applications (Flow) monitoring

- ◆ Are a particular flow running correctly?
- ◆ Monitor per application (flow, PG, processor) metrics such as number of flow files, data size, queues, back pressure, etc
- ◆ Solution
 - S2S Reporting tasks
 - Custom flow developments (integrate monitoring and reporting in the application logic)

NiFi UI for service monitoring

Status Indicator

Processor Name

Bulletin Indicator

Processor Type

Active Tasks

5-Minute Statistics

Copy to /review

PutFile 1.2.0

org.apache.nifi - nifi-standard-nar

1

29 (14.16 MB)

5 min

4.88 MB / 4.88 MB

5 min

0 (0 bytes)

5 min

29 / 00:00:00.123

5 min

Name

Active Tasks

Bulletin Indicator

Hover over to see Comments

Version State Counts

5-Minute Statistics

Component Counts

Process Group ABC

2

1 1 2 9 4 0

Queued 26 (12.7 MB)

In 8 (800 KB) → 2

5 min

Read/Write 14.72 MB / 14.8 MB

5 min

Out 3 → 16 (78.57 KB)

5 min

0 * 0 0 0 0 ? 0

Remote Instance Name

Remote Instance URL

5-Minute Statistics

Comments

Last Refresh Time

Secure Indicator

Transmission Status

NiFi Flow

http://localhost:8080/nifi/

Sent 0 (0 bytes) → 0

5 min

Received 0 → 0 (0 bytes)

5 min

No comments specified

05/25/2017 16:50:46 EDT

NiFi Summary

PROCESSORS

INPUT PORTS

OUTPUT PORTS

REMOTE PROCESS GROUPS

CONNECTIONS

PROCESS GROUPS

Component Tabs

Displaying 60 of 60

Filter

by name

View: Single node Cluster

	Name	Type	Run Status	In / Size 5 min	Read / Write 5 min	Out / Size 5 min	Tasks / Time 5 min	
	Base64EncodeCo...	Base64EncodeCo...	Invalid	0 (0 bytes)	0 bytes / 0 bytes	0 (0 bytes)	0 / 00:00:00.000	→ 📊 🔄
	Capture Network ...	ExecuteProcess	Running (5)	0 (0 bytes)	0 bytes / 245.97 MB	2,883 (2...	24.55.968	→ 📊 🔄
	Check if Dataset ...	IdentifyMimeType	Running	0 (0 bytes)	0 bytes / 0 bytes	0 (0 bytes)	0 / 00:00:00.000	→ 📊 🔄
	Decompress	Content	Running	0 (0 bytes)	0 bytes / 0 bytes	0 (0 bytes)	0 / 00:00:00.000	→ 📊 🔄
	Detect	Attribute	Running (2)	11,741,700 (13 GB)	0 bytes / 0 bytes	0 (0 bytes)	00.09....	→ 📊 🔄
	Empty contents	ReplaceText	Running	0 (0 bytes)	0 bytes / 0 bytes	0 (0 bytes)	0 / 00:00:00.000	→ 📊 🔄
	Evaluate.JsonPath	Evaluate.JsonPath	Running	0 (0 bytes)	0 bytes / 0 bytes	0 (0 bytes)	0 / 00:00:00.000	→ 📊 🔄
	Evaluate.JsonPath	Evaluate.JsonPath	Running (4)	11,742,510 (13 GB)	13 GB / 0 bytes	11,742,51...	00:20...	→ 📊 🔄
	Extract File ID	UpdateAttribute	Running	0 (0 bytes)	0 bytes / 0 bytes	0 (0 bytes)	0 / 00:00:00.000	→ 📊 🔄
	Extract File ID	UpdateAttribute	Running	0 (0 bytes)	0 bytes / 0 bytes	0 (0 bytes)	0 / 00:00:00.000	→ 📊 🔄
	ExtractText	ExtractText	Running	1,498,914 (244.45 ...)	244.45 MB / 0 bytes	1,496,995 (244.25 ...)	1,500,692 / 00:02:4...	→ 📊 🔄

Last updated: 17:45:24 UTC

system diagnostics

Status History

Status History

Id

b0a3b095-5408-38b8-b0dd-7b31babb2e04

Group Id

50380170-a353-47ad-8b01-14f3ac4b4551

Name

Twitter Garden Host

Type

GetTwitter

Start

07/28/2016 19:13:47.262

End

07/28/2016 22:47:51.162

NiFi

Min / Max / Mean

0.00 bytes / 76.78 MB / 35.89 MB

Nodes

Min / Max / Mean

0.00 bytes / 76.78 MB / 35.89 MB

nifi-04:8080

Last updated: 22:48:13 UTC

Bytes Written (5 ...

Bytes Written (5 mins)

76.29 MB

66.76 MB

57.22 MB

47.68 MB

38.15 MB

28.61 MB

19.07 MB

9.54 MB

0.00 bytes

20:00

21:00

22:00

Bytes Written (5 mins)

79.85 MB

0.00 bytes

20:00

21:00

22:00

CLOSE

Tools available for service monitoring

- ◆ Bootstrap notifier: send notification when the NiFi starts, stops or died unexpectedly
 - Email/HTTP notification services
- ◆ Use reporting tasks: export metrics to your monitoring solution
 - AmbariReportingTask (global, process group)
 - MonitorDiskUsage (Flowfile, content, provenance repositories)
 - MonitorMemory
- ◆ Also, monitor inactivity
 - NiFi has a built-in MonitorActivity processor
 - To be used with the S2SBulletinReportingTask
 - You can use InvokeHTTP to call the reporting Rest API

Add Reporting Task

Source: all groups | Displaying 11 of 11 | Filter

Source	Type	Version	Tags
ambari	AmbariReportingTask	1.5.0	ambari, metrics, reporting
execute	ControllerStatusReportingTa...	1.5.0	stats, log
garbage collection	DataDogReportingTask	1.5.0	datadog, metrics, reporting
gc	MetricsReportingTask	1.5.0	metrics, reporting
groovy	MonitorDiskUsage	1.5.0	disk, repo, warning, storage, mo...
heap	MonitorMemory	1.5.0	jvm, memory, warning, monitor, ...
js	ScriptedReportingTask	1.5.0	lua, python, groovy, jython, js, lu...
jvm	SiteToSiteBulletinReportingT...	1.5.0	site, restricted, bulletin, site to s...
jython	SiteToSiteProvenanceReport...	1.5.0	lineage, site, provenance, restri...
log	SiteToSiteStatusReportingTa...	1.5.0	site, metrics, history, status, sit...
lua	StandardGangliaReporter	1.5.0	stats, ganglia

AmbariReportingTask 1.5.0 org.apache.nifi - nifi-ambari-nar

Publishes metrics from NiFi to Ambari Metrics Service (AMS). Due to how the Ambari Metrics Service works, this reporting task should be scheduled to run every 60 seconds. Each iteration it will send the metrics from the previous iteration, and calculate the current metrics to be sent on next iteration. Scheduling this reporting task at a frequency other than 60 seconds may produce u...

CANCEL ADD

How to achieve granular monitoring?

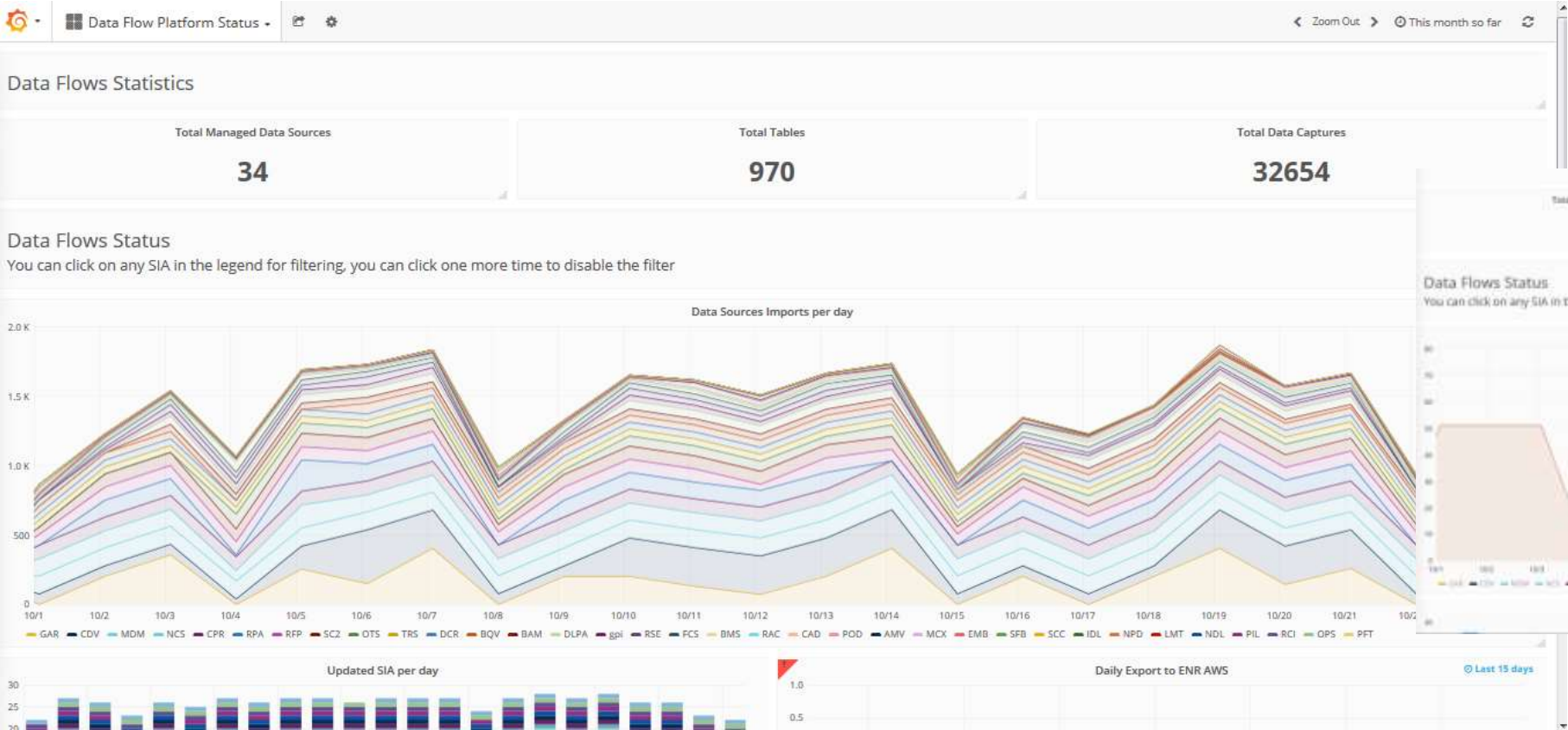
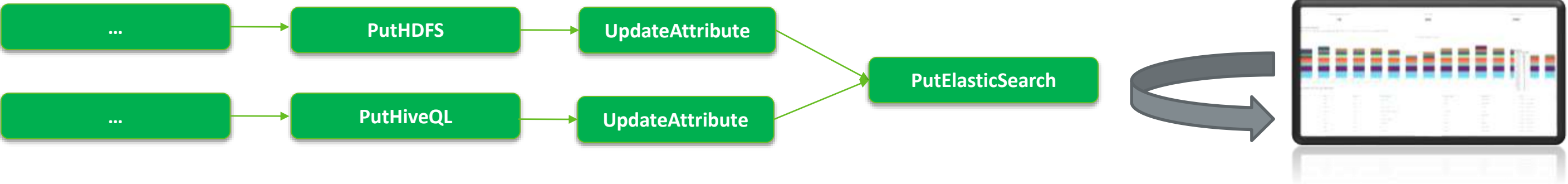
Monitoring Driven Development (MDD)

- ◆ Integrate your monitoring logic in the flow design
 - Count data (lines, tables, events, etc)
 - Extract business metadata from data (table names, project name, source directory, etc)
- ◆ Handle different types of errors (connection, format, schema, etc)
- ◆ Send extracted KPI to brokers, dashboards, API, files, etc

S2S reporting tasks

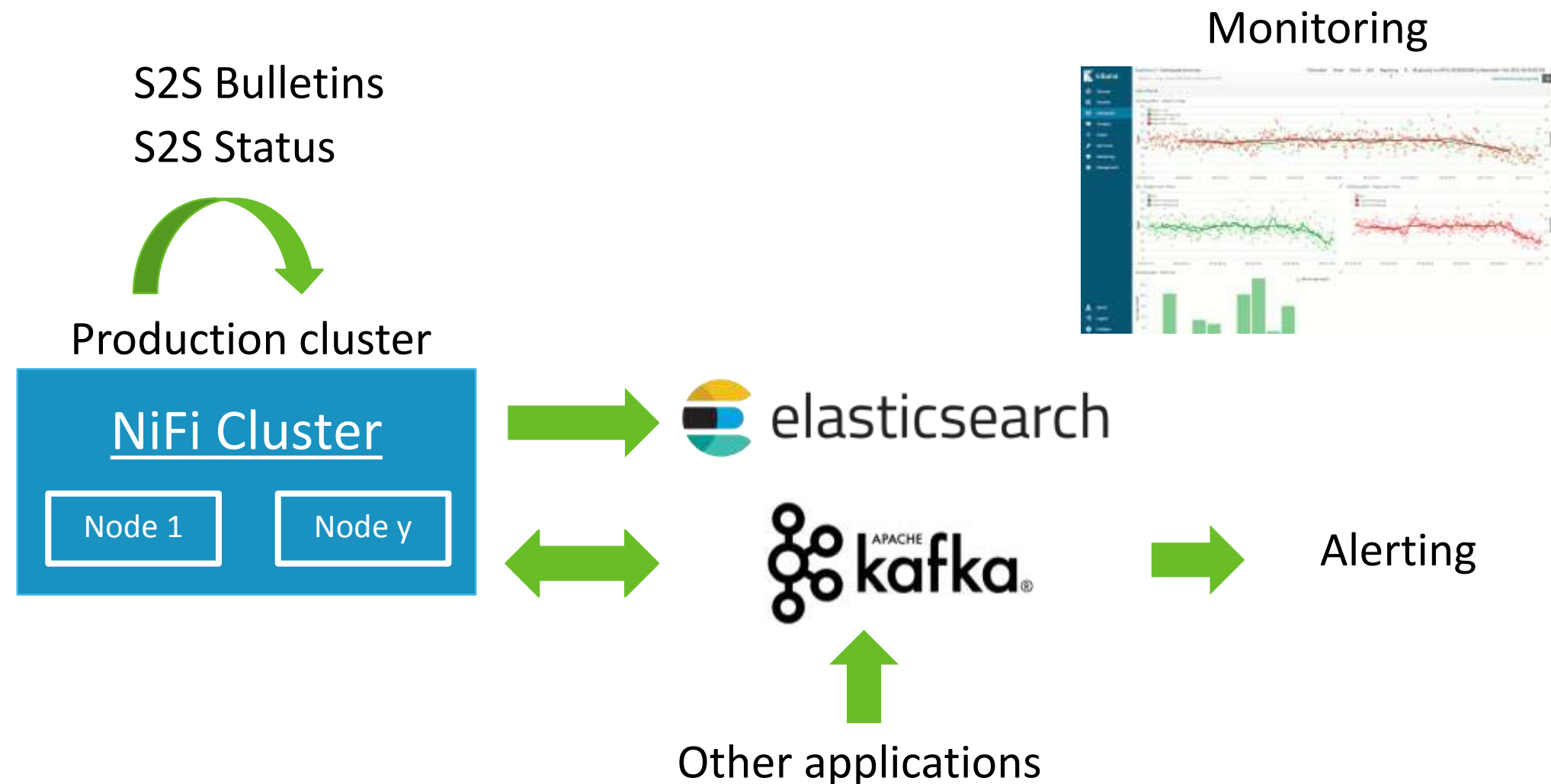
- ◆ NIFInception: NiFi can export metrics to to another cluster or to itself
- ◆ Bulletins provide information on errors
- ◆ Status provide metrics on usage
- ◆ These reports become data and we can use the power of NiFi to extract our KPI
- ◆ **Naming convention is key**

Monitoring Driven Development



NiFiInception architecture

- Generate status and bulletins
- Send them through S2S to a local input port
- Ingest bulletin and status reports
- Parse reports (JSON) and extract useful KPI
- Send KPI to a dashboard tool (AMS, Elastic, etc)
- Use a broker for alerting (Kafka)
- Can ingest logs from other systems or application



NiFiInception architecture 2

- Ingest data
- Generate status and bulletins
- Send them through S2S to a remote cluster

Production cluster



S2S Bulletins
S2S Status

Monitoring cluster



 elasticsearch



 **APACHE** kafka®



Alerting

Monitoring



- Ingest bulletin and status reports
- Parse reports (JSON) and extract useful KPI
- Send KPI to a dashboard tool (AMS, Elastic, etc)
- Use a broker for alerting (Kafka)
- Can ingest logs from other systems or application

Other applications

GROUPE RENAULT



Lessons learnt at Renault

GROUPE RENAULT



Recommendations from day to day life

- ◆ High availability means several weeks to validate before Go live
- ◆ Use Ranger authorizations instead of NIFI build in ACL management
- ◆ Too much « if then do else »: do the data flow and do not offload other processing solutions
- ◆ S2S : Minimize the number of RPG and self-RPG. Use attributes and routing.
- ◆ Put hive via Two redundant processors
- ◆ Discuss with DBA to better handle impacts (Views VS Tables, Open Sockets ..etc).
- ◆ Backup flowfile.xml.gz (users.xml et authorizations.xml) using NiFi itself.

Recommendations from day to day life

- ◆ Success flag can be done through counters instead of InvokHTTP
- ◆ In case of CIFS, do not forget to use AUTO MOUNT for NFS client side on NiFi Servers.
- ◆ Check '0' size before transmitting into HDFS (especially for IoT use case).
- ◆ Configure a TIMER for when we use FAILURE redirections to avoid back-pressure scenario.
- ◆ Build separate clusters for separate use cases (Real Time with SSD, Batch, etc ...)
- ◆ CA PKI very useful for internal communications, no need to wait Security teams answers but need security skills (SSL) .

Questions

GROUPE RENAULT

