



## Thème 3 – Data Routing

Carole DUMOULIN, Vincent BONNET, Théophile BORNON, Xavier BOUVET

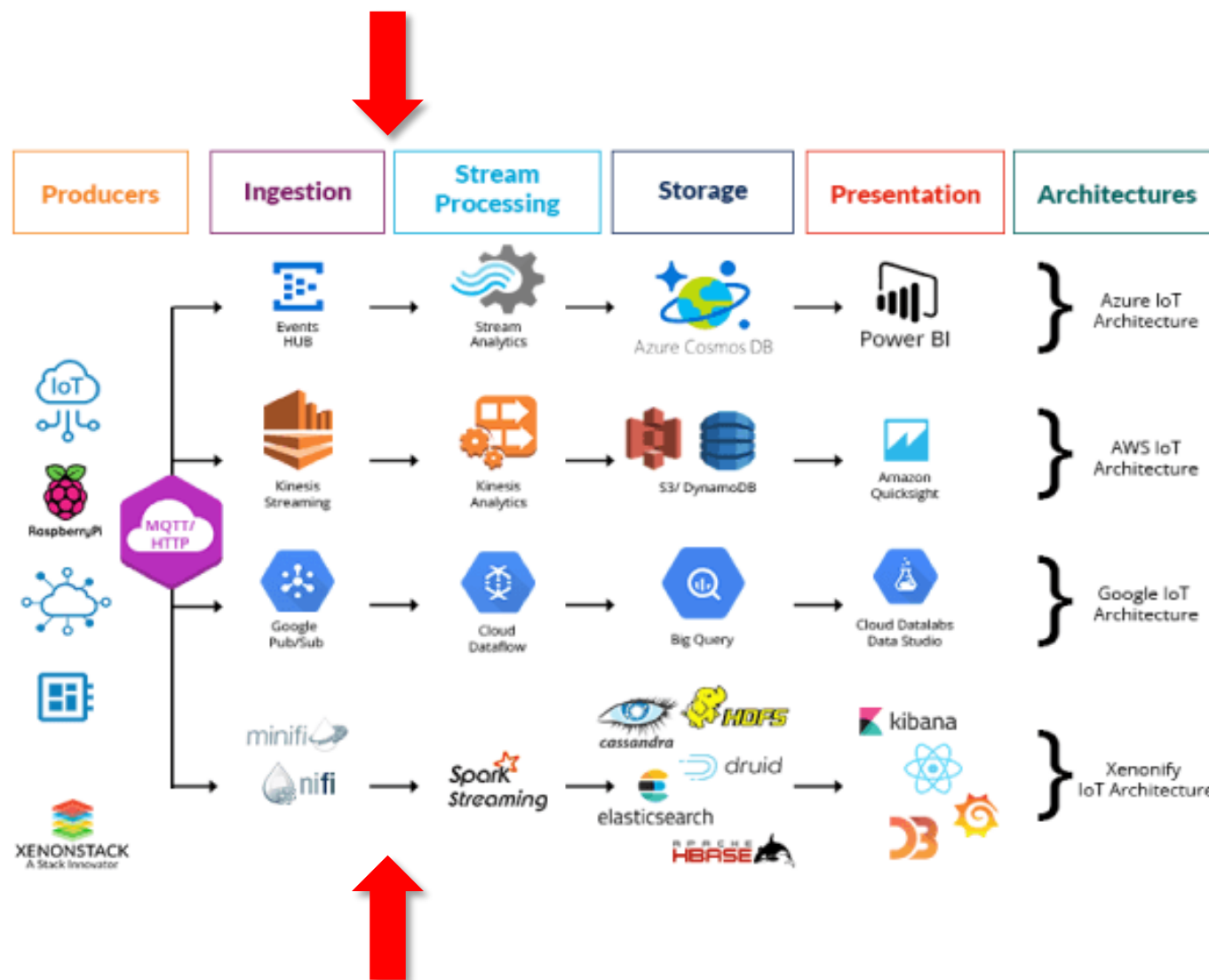






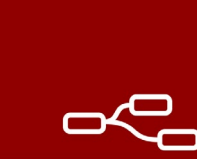
# Plan du cours

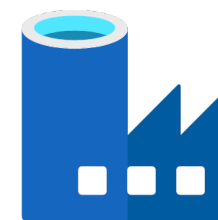
1. Introduction
2. Apache NiFi
  1. NiFi
  2. Histoire
  3. Présentation de l'UI
  4. Caractéristiques
  5. Déploiement
  6. Architecture
  7. Démonstrations avec des exemples concrets
  8. Limites
  9. Différences avec MiNiFi
3. Annexe
4. Références



**Concurrents de NiFi**  
Datafactory, Dataflow, NodeRed

  
Google Cloud Dataflow

  
Node-RED







# Apache NiFi

*(... des idées de génie )*

1. NiFi
2. Histoire
3. Présentation de l'UI
4. Caractéristiques
5. Architecture
6. Démonstrations avec des exemples concrets
7. Déploiement
8. Limites
9. Différence avec MiNiFi



# Alors Jamy, qu'est ce que NiFi ?

Apache NiFi est une plate-forme d'**ingestion de données** open source

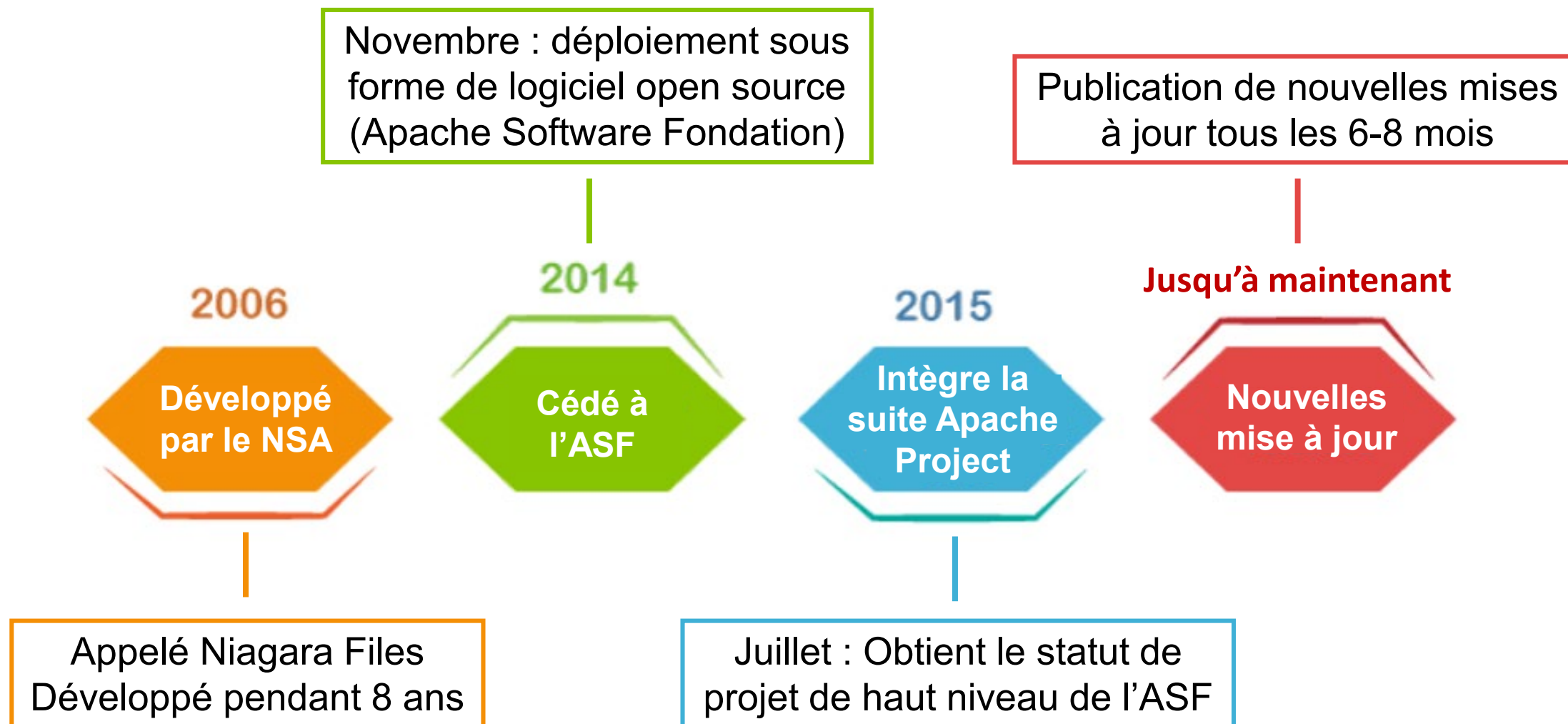


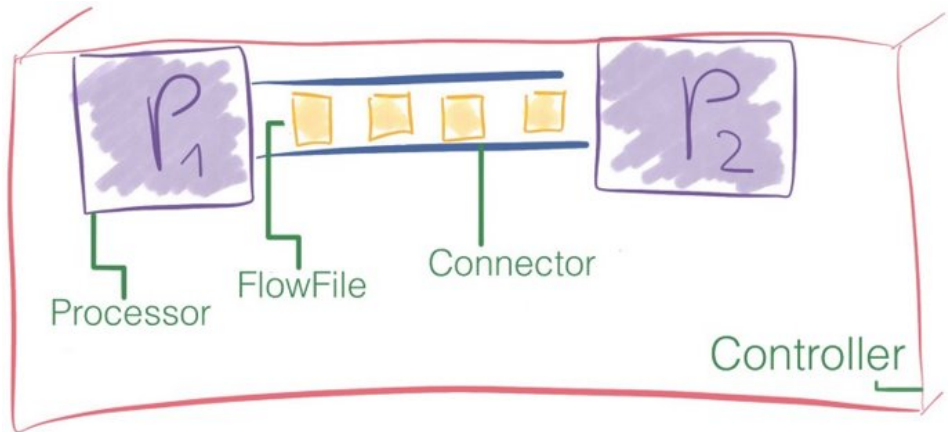
- Puissant et fiable utilisé pour **traiter** et **distribuer** des données entre différents système
- Aide à **gérer** et à **automatiser** le flux de données entre les systems
- Il fournit une **interface web** utilisateur, qui utilise le protocole HTTPS pour exécuter NiFi sur un navigateur web qui rend l'interaction sécurisée

Sur la plateforme NiFi, nous pouvons définir :

- la source,
- le processeur et la destination pour la collecte des données,
- la transmission des données et le stockage des données

Chaque processeur de NiFi a certaines caractéristiques (succès, échec, réessai, données invalides, etc.), qui sont utilisées lors de la connexion d'un processeur à un autre





Quand 2 processeurs communiquent entre eux, l'échange de données s'effectue via des connecteurs. Ces connecteurs transportent les informations par « paquets » appelé FlowFile



## FlowFile Anatomy

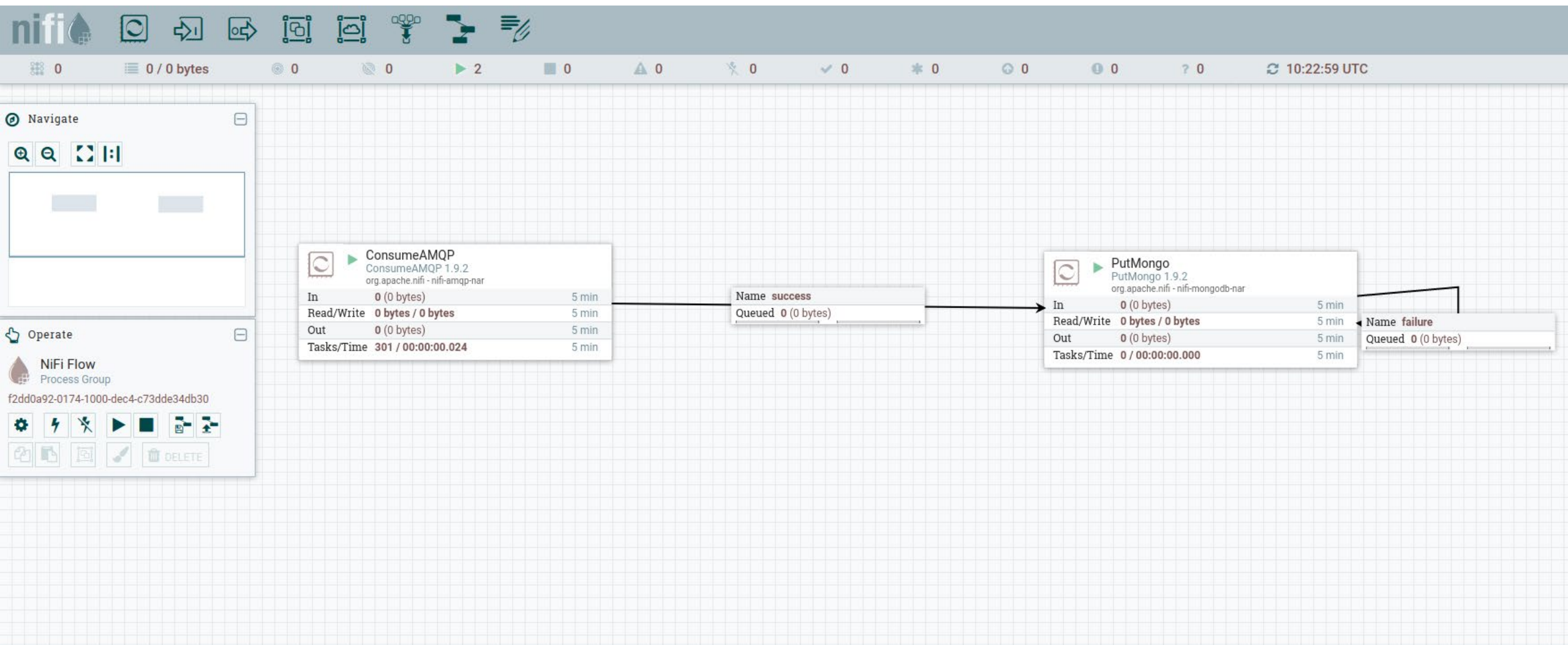
Regardons en détail ce que contient un FlowFile :

- Attributs (métadonnées)
- Contenu (message)





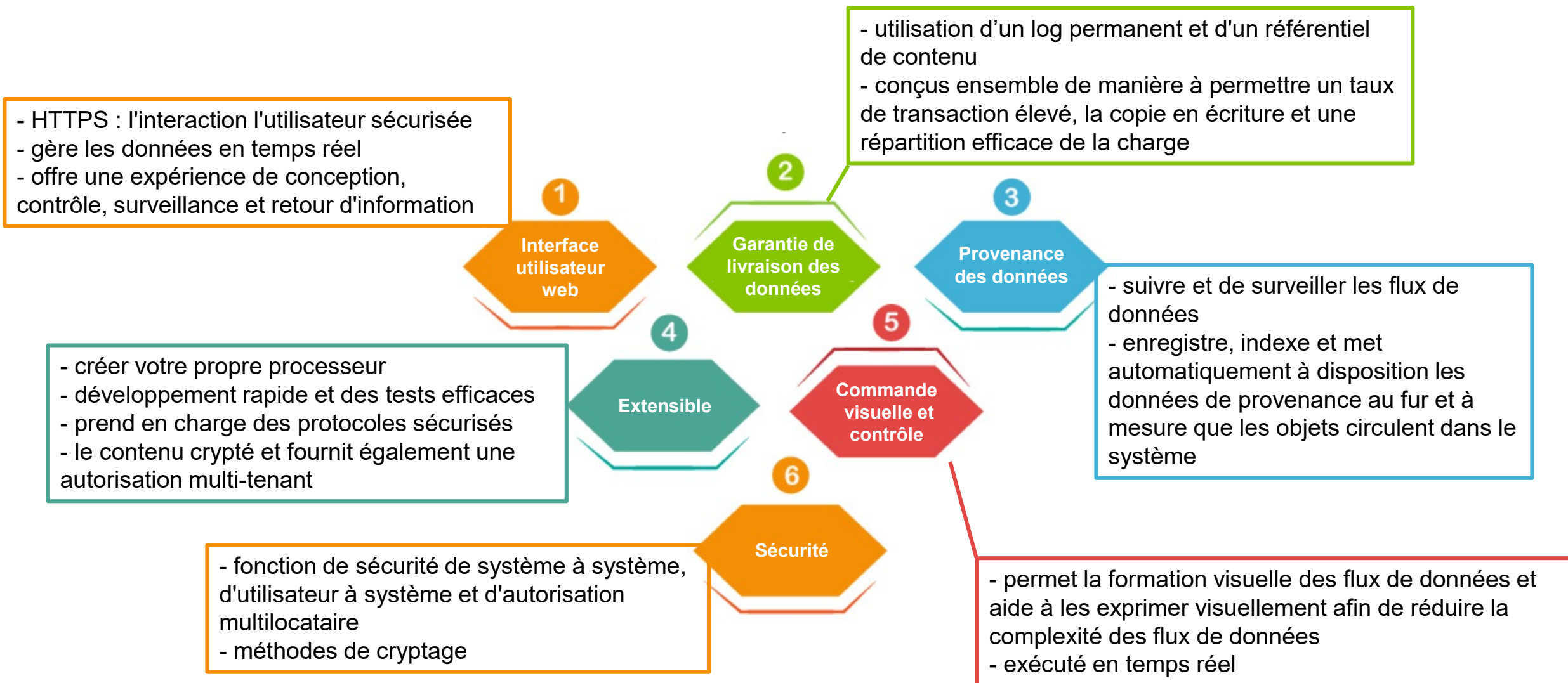
# Présentation de l'interface utilisateur







# Caractéristiques Apache NiFi



Vous pouvez retrouver les détails de toutes les caractéristiques sur ce [lien](#)



- **Local**

NiFi tourne dans une JVM en mode local sur le système d'exploitation hôte (Windows, Linux ou Mac)

- **Mode cluster**

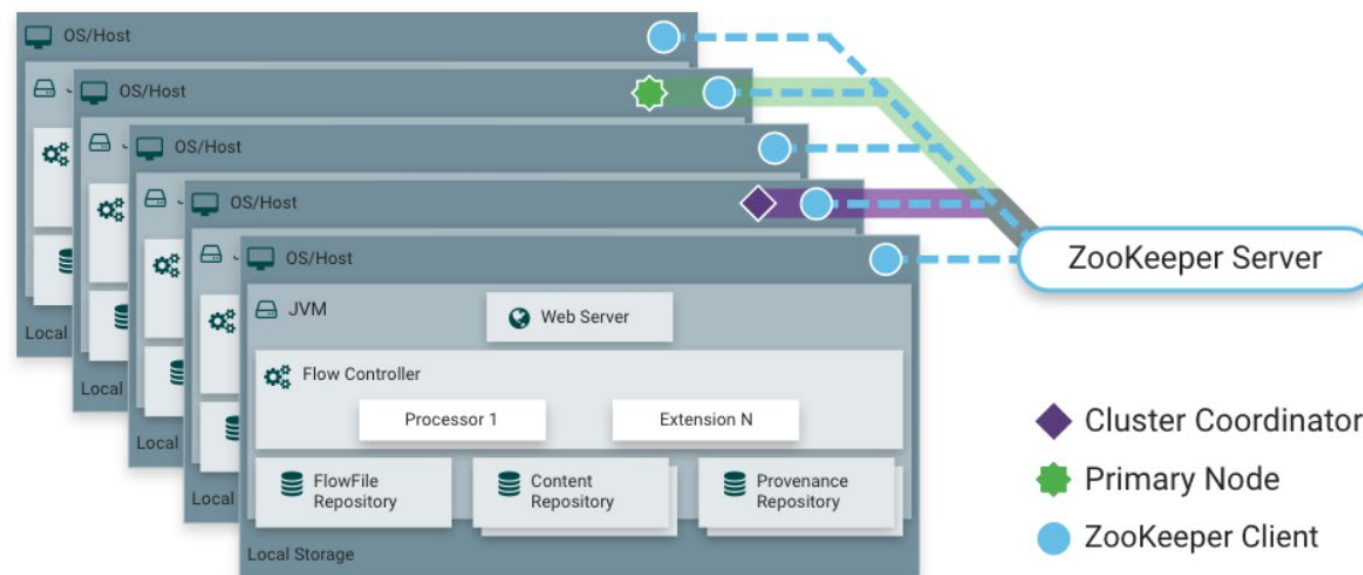
Utilisation de [Apache Zookeeper](#) pour la gestion de la configuration afin d'assurer la haute disponibilité des services

*[ZooKeeper](#) est un service centralisé pour la maintenance des informations de configuration, le nommage, la synchronisation distribuée et la fourniture de services de groupe*



# Architecture – Mode cluster

Dans le cluster NiFi, chaque nœud travaille sur un ensemble de données différent, mais il effectue la même tâche sur les données. [Apache Zookeeper](#) choisit un seul nœud comme **coordinateur du cluster** et gère automatiquement l'échec. Chaque nœud du cluster rend compte au coordinateur de l'état actuel. Le coordinateur du cluster est responsable de la connexion ou de la déconnexion des nœuds.

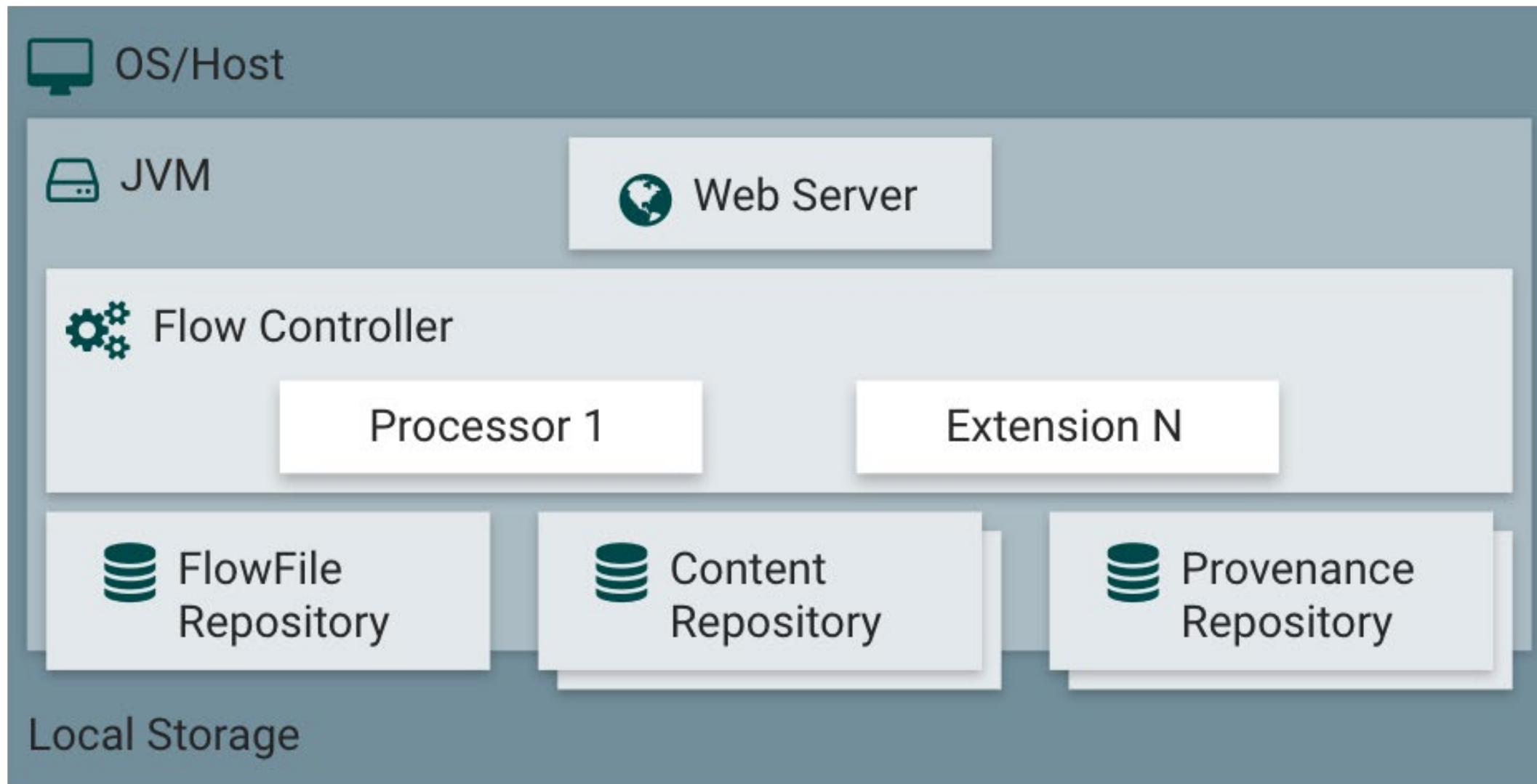


Chaque cluster possède un **nœud primaire**, qui est également sélectionné par Zookeeper. Il est possible d'interagir avec le cluster NiFi en tant que gestionnaire de flux de données ou développeur final en utilisant l'interface utilisateur de n'importe quel nœud. Toutes les modifications apportées par l'utilisateur sont répliquées pour tous les nœuds du cluster, ce qui permet d'avoir plusieurs points d'entrée.





# Architecture – Stockage local





# Architecture – Keys component

- **Web Server**

The Web Server hosts the HTTP-based commands and control API of NiFi.

- **Flow Controller**

The flow controller provides threads to execute the extensions. It also schedules the extensions when resources are received to execute. It works as a brain of operations.

- **Extension**

Extensions are various type of plugin that allows Apache NiFi to interact with different systems. Extensions help the process to complete the task. NiFi has several types of extensions. These extensions are executed and operated within the Java Virtual Machine (JVM).

- **FlowFile repository**

The FlowFile repository contains the current state and attribute of each FlowFile that passes through the data flows of NiFi. NiFi keeps track of the state in FlowFile repository, which is currently active in the flow. The root directory is the default location of this repository, it can be changed. The default location of this repository can be changed by changing the property "nifi.flowfile.repository.directory".

- **Content repository**

The content repository stores all the data present in all the flowfiles. Implementation of the content repository is pluggable same as the FlowFile repository. Its default approach is a simple mechanism to store block of data in file system.

The default directory of content repository is in root directory of NiFi and can be changed by changing the "org.apache.nifi.controller.repository.FileSystemRepository" property.

- **Provenance repository**

The provenance repository is the repository that stores all the provenance event data. Event data is indexed and searchable within each location. It allows the user to check information about FlowFile, which means it tracks and stores all the events of all flowfiles that flows in the Apache NiFi. It also enables the troubleshooting if any issue occurs while processing FlowFile

Provenance repository has divided into two types:

- Volatile provenance repository - All provenance data is lost after restart in this repository.
- Persistence provenance repository - The default directory of persistence provenance is in the root directory of Apache NiFi. It can be changed using the "apache.nifi.provenance.PersistanceProvenanceRepository" property.

## Usage de NiFi avec orientation de flux conditionnel et queue de messages



```
version: '3.7'
services:
  nifi:
    image: "apache/nifi:1.9.2"
    hostname: "nifi"
    environment:
      NIFI_WEB_HTTP_PORT: "8080"
    ports:
      - "8080:8080"
    volumes:
      - "./input:/input"
      - "./output:/output"
    networks:
      - iot-labs
    labels:
      NAME: "nifi"
networks:
  iot-labs:
    external: true
```

## Manipulation et transformation de données JSON

## Processors groups et droits d'accès





## Usage d'une l'API

Pour l'API de Nifi c'est par [ici](#)



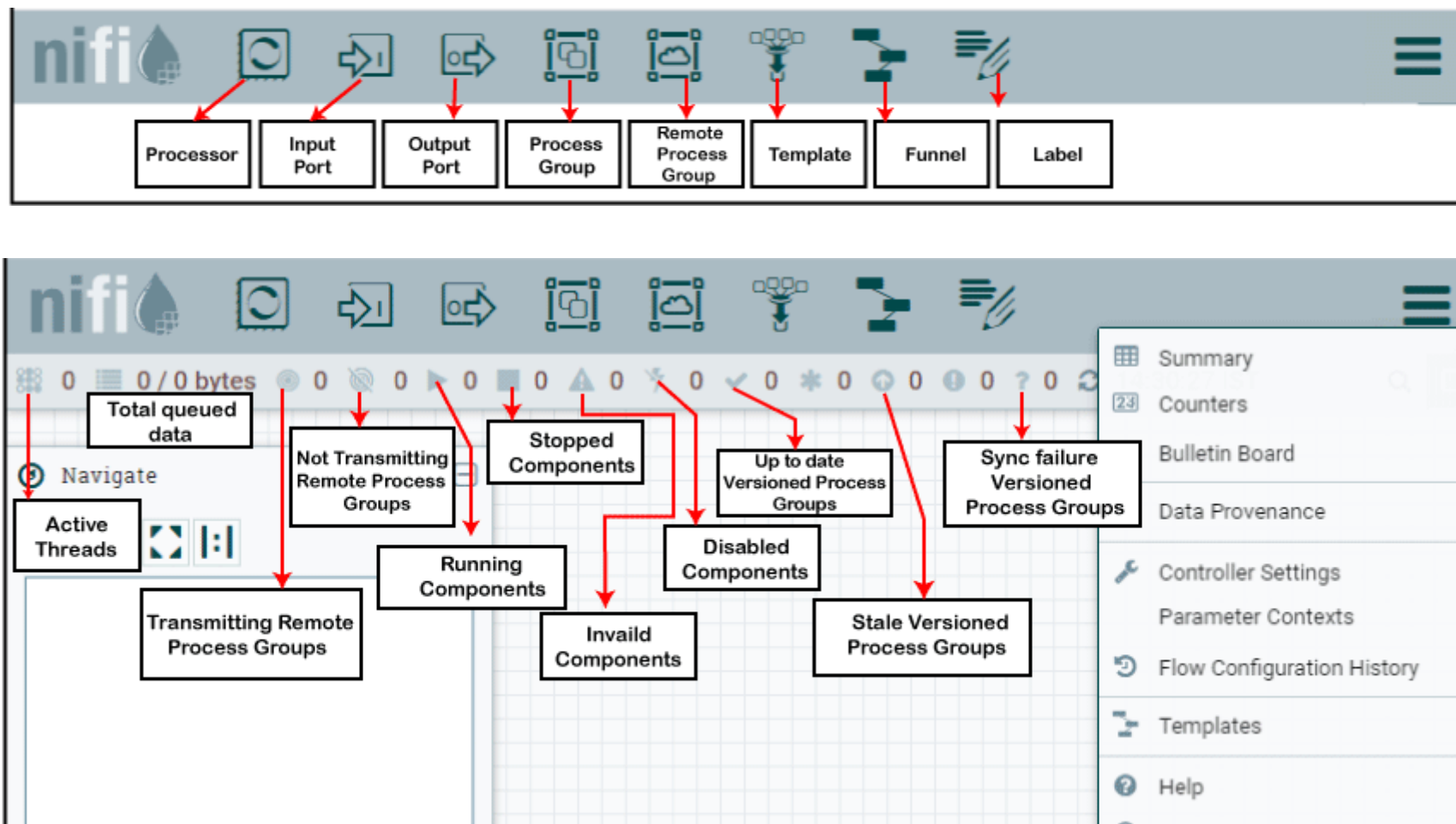
- Lors de toute modification par un nœud utilisateur, celui-ci est **déconnecté du cluster NiFi**, et le fichier flow.xml devient alors **invalide**. Un nœud ne peut pas se reconnecter au cluster tant que l'administrateur n'a pas copié manuellement le fichier xml du nœud connecté
- Toutes les données ne sont pas créées de la même manière
- Faible mémoire disponible
- Usage généralement limité aux DataCenters de par la charge de donnée



- Origine : Mini Nifi
- MiNiFi, sous-projet de NiFi qui se concentre sur la collecte de données à la source. Apache NiFi constitue un backbone pour gérer des flux de données complexes dans l'IoT. Apache a livré une mise à jour importante d'Apache NiFi incluant MiNiFi qui apporte des agents très légers qui peuvent être poussés en bout de réseau vers des équipements plus petits. Il s'agit d'une technologie intéressante et très récente.

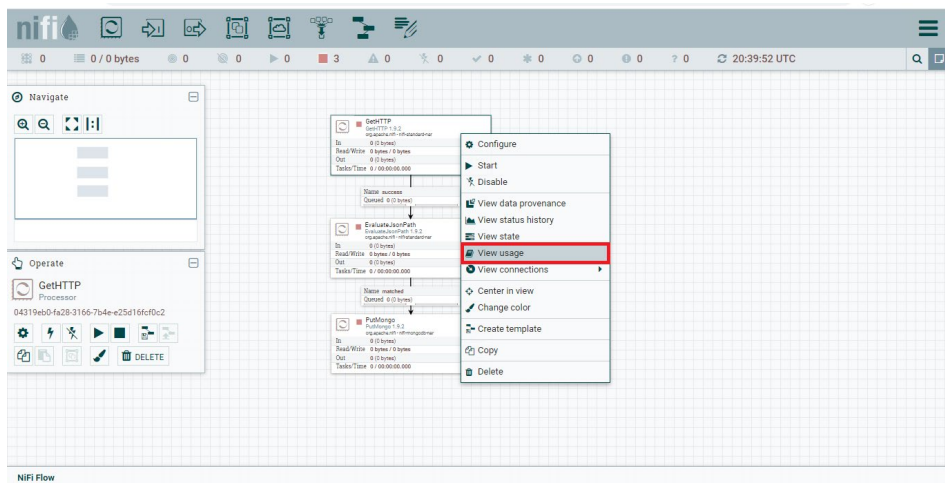


# Annexe - Présentation de l'interface utilisateur

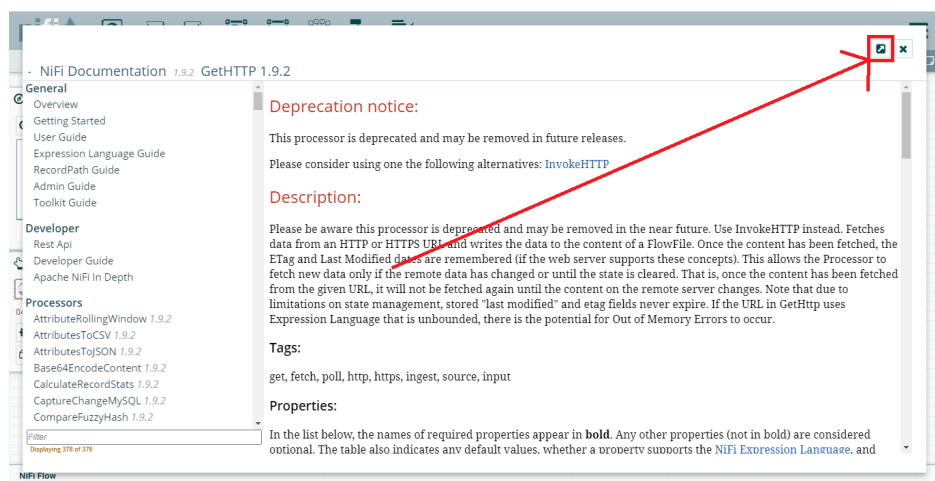


Pour plus de précisions sur l'UI, consultez ce [lien](#)

## Ouvrir la documentation depuis NiFi

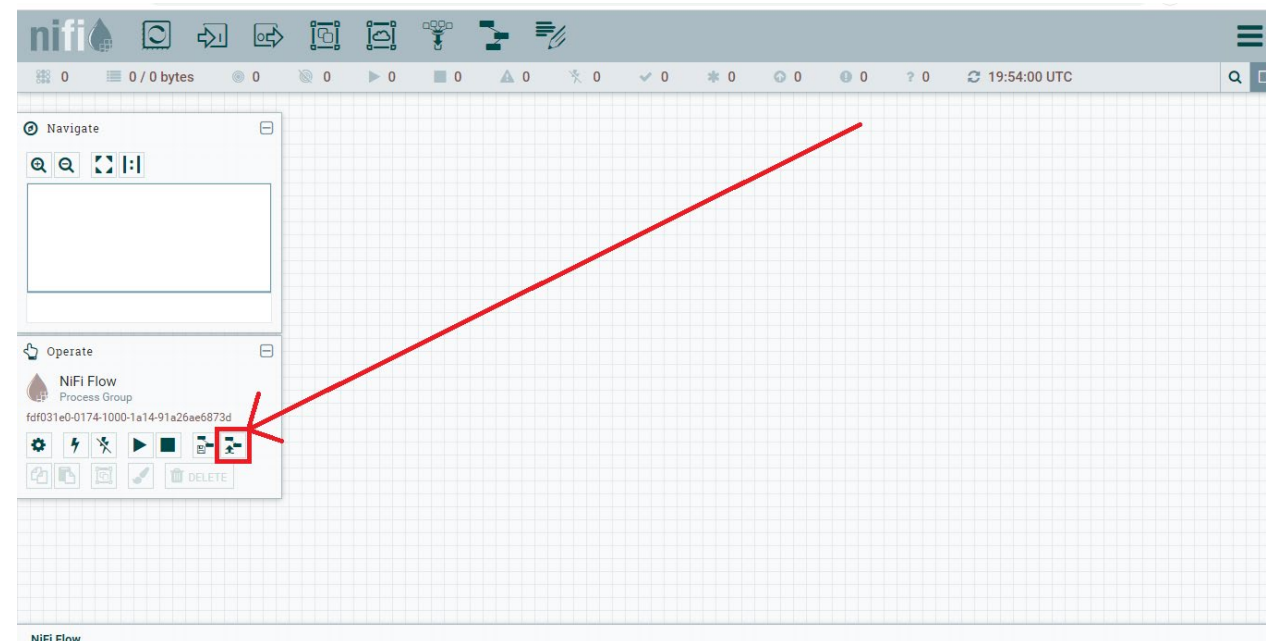


## Dans un nouvel onglet depuis NiFi



## Importation d'un template

Bouton grisé en cas de composant sélectionné



(...sinon, ouvrez la directement en [ligne](#))

## Paramétrer les propriétés d'un processeur

Les paramètres en **gras** sont obligatoires

Certain champ utilise Apache NiFi Expression Language

**Configure Processor**

SETTINGS SCHEDULING **PROPERTIES** COMMENTS

Required field +

| Property                    | Value   |
|-----------------------------|---|
| URL                         | http://api.weatherstack.com/current?access_key=ab9... |
| Filename                    | \$.{uuid}   |
| SSL Context Service         | No value set  |
| Username                    | No value set  |
| Password                    | No value set  |
| Connection Timeout          | 30 sec  |
| Data Timeout                | 30 sec  |
| User Agent                  | No value set  |
| Accept Content-Type         | No value set  |
| Follow Redirects            | false   |
| Redirect Cookie Policy      | default   |
| Proxy Configuration Service | No value set  |
| Proxy Host                  | No value set  |
| Proxy Port                  | No value set  |

CANCEL APPLY

## Paramétrage de la gestion de temps entre deux exécutions d'un processeur

**Configure Processor**

SETTINGS **SCHEDULING** PROPERTIES COMMENTS

Scheduling Strategy ?  
Timer driven

Concurrent Tasks ?  
1

Execution ?  
All nodes

Run Schedule ?  
30 sec

CANCEL APPLY



Certains ports sont **bloqués** par l'administrateur réseau empêchant l'exécution de certain processus ne résultant en aucune erreur.

L'exemple de l'envoi de mail doit être fait sur en partage de connexion.

- <http://nifi.apache.org/docs.html>
- <https://www.javatpoint.com/apache-nifi>
- <https://meritis.fr/challenge/bigdata/travailler-donnee-apache-nifi/>
- <https://zookeeper.apache.org/>
- <https://www.freecodecamp.org/news/nifi-surf-on-your-dataflow-4f3343c50aa2/>



## Merci de votre écoute



Carole DUMOULIN, Vincent BONNET, Théophile BORNON, Xavier BOUVET

