



DATA ROUTING

ESILV A5 IBO IOT

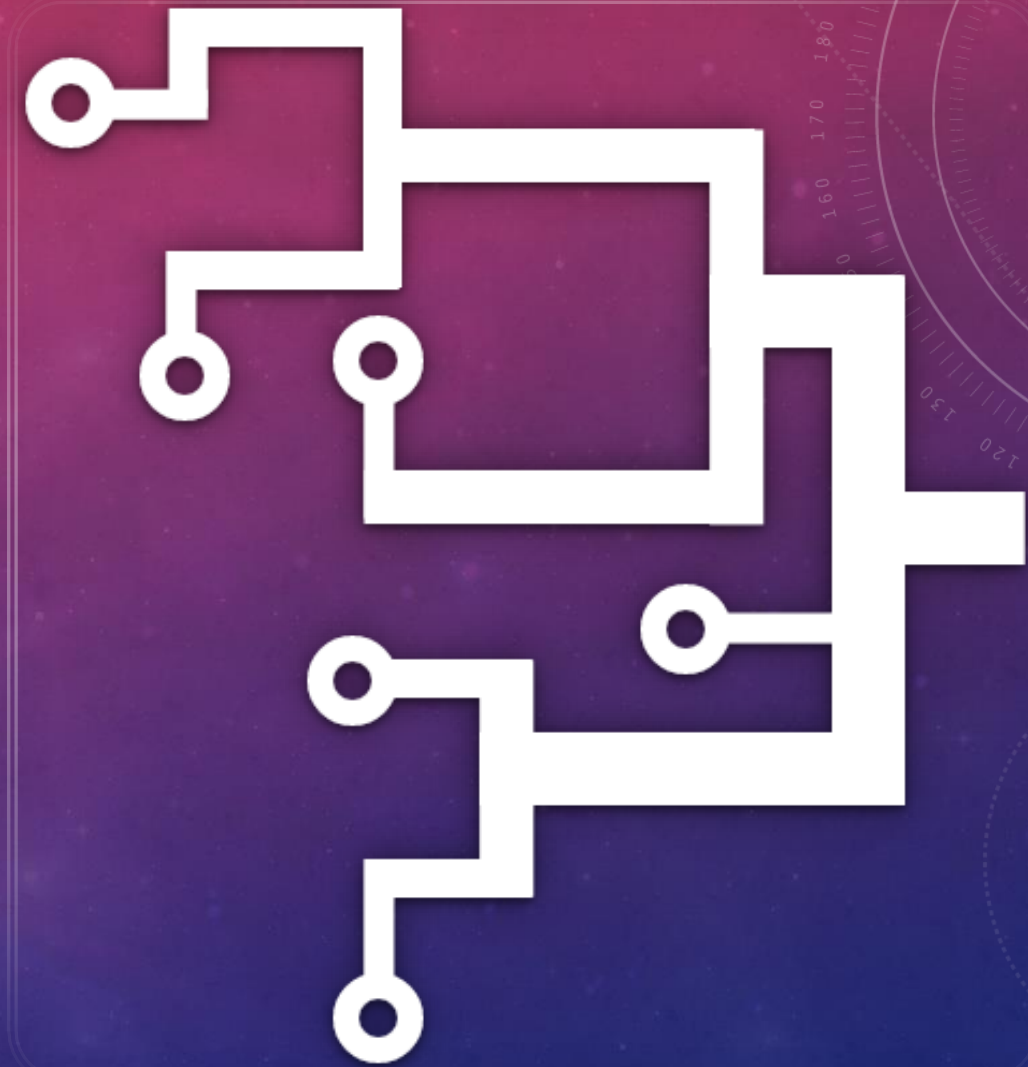
BOUCAUD STEPHANE – GAILLET LOUIS – CHHEANG THOMAS

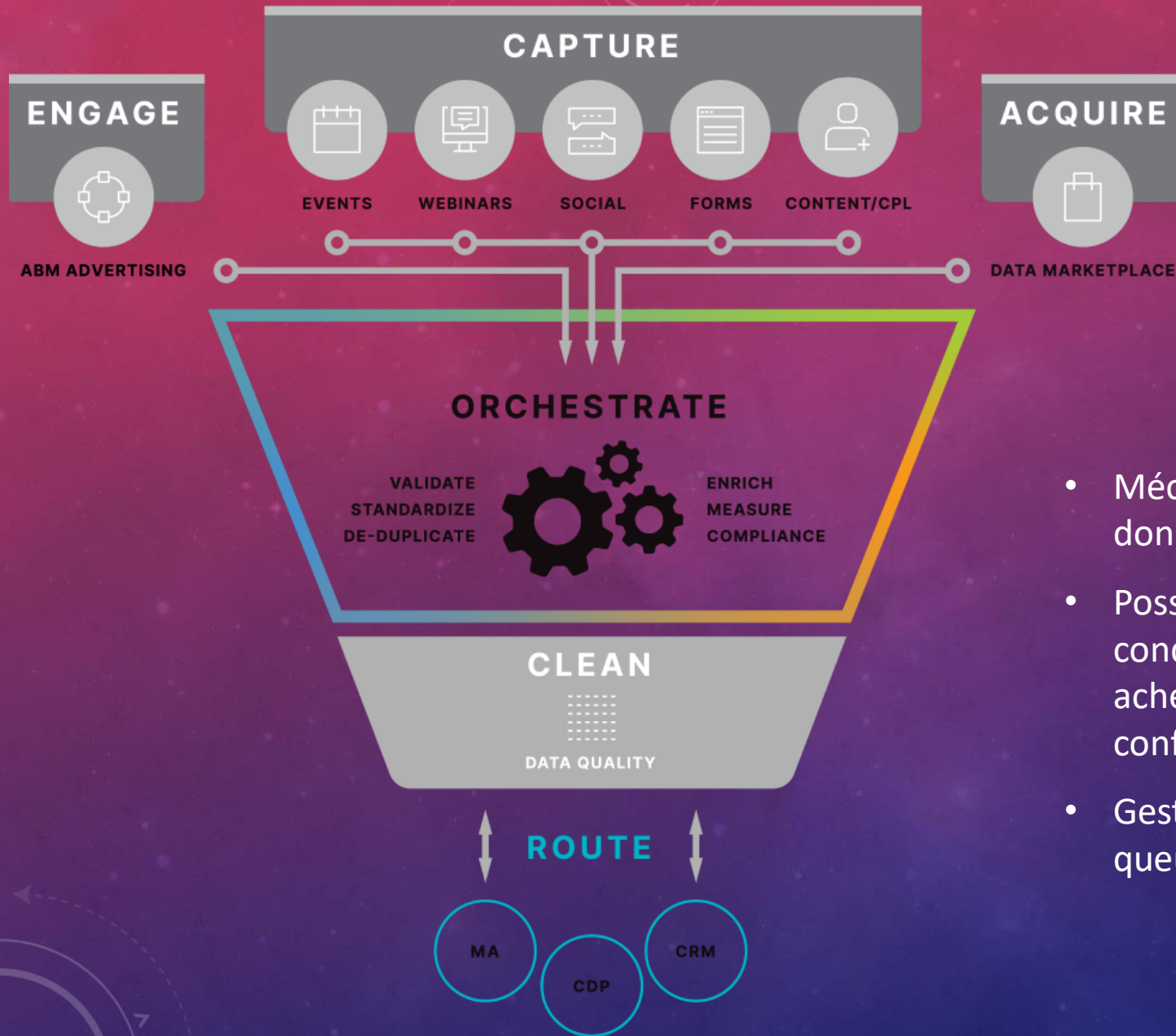
SOMMAIRE

- DATA ROUTING - Introduction
- QUELQUES DEFINITIONS – Concepts à comprendre
- APACHE NIFI – Présentation de l'outil
- DIFFERENCE AVEC MiNifi
- DÉPLOIEMENT D'APACHE NIFI
- DEMONSTRATION SOUS NIFI

DATA ROUTING

- Definition
- Exemple
- Solutions





DATA ROUTING DÉFINITION

- Mécanisme par lequel on oriente les flux de données entre émetteurs et destinataires
- Possibilité de modification ou mise sous condition des données au cours de leurs acheminements afin de les valider, standardiser, conformer, enrichir, dupliquer.
- Gestion des flux en temps réel / différé par des queues dans un cadre d'IoT

Reçu

Receipt

Il s'agit de certifier que JAL a reçu la somme suivante.

This is to certify that JAL has received the following amount.

Japan Airlines Co., Ltd.



JAPAN AIRLINES

DATA ROUTING EXEMPLE

PROVENANT DE

RECEIVED FROM

JAL

TOTAL

TOTAL

147,79 € (EUR)

EN PAIEMENT DE

IN PAYMENT OF

TARIF · TAXE · FRAIS · CHARGE

MODE DE PAIEMENT

FORM OF PAYMENT

REMARQUES

REMARKS

DATE D'ÉMISSION DU BILLET

TICKET ISSUED DATE

15/09/2017

DATE AFFICHÉE SUR LE REÇU EN LIGNE

RECEIPT ISSUE DATE

05/03/2018

Il s'agit de l'affichage électronique des données reçues.

This is an electronic display of receipt data.

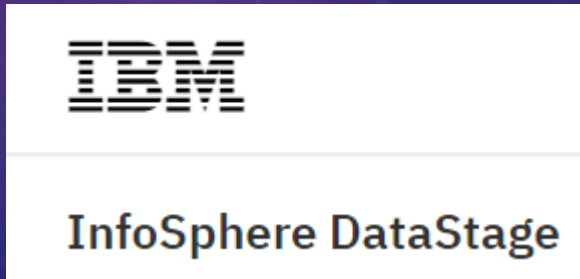
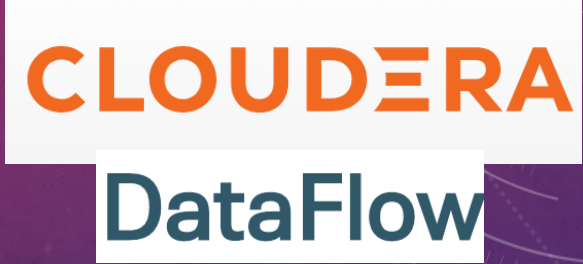
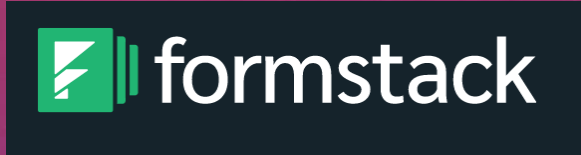
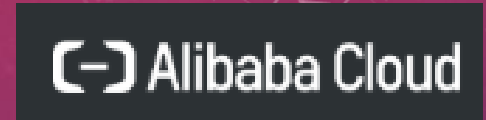
Par exemple, on peut créer un reçu pour le client et un légèrement différent pour le vendeur pour une seule commande. Les deux auront la même data en entrée mais seront des documents différents en sortie. Par ailleurs, on peut gérer la création des documents avec des conditions logiques. Par exemple : si la commande est incomplète : ne pas envoyer un reçu mais plutôt un document en lien avec l'avancée de la préparation de la commande.

DATE D'ÉMISSION DU BILLET (M) : 147,79 € (EUR)

1312130133014

DATA ROUTING

SOLUTIONS





QUELQUES DEFINITIONS

- Paradigme de programmation
- Programmation orientée composants (POC)
- Dataflow programming
- Flow-based programming
- Tolérance aux fautes (fault tolerance)

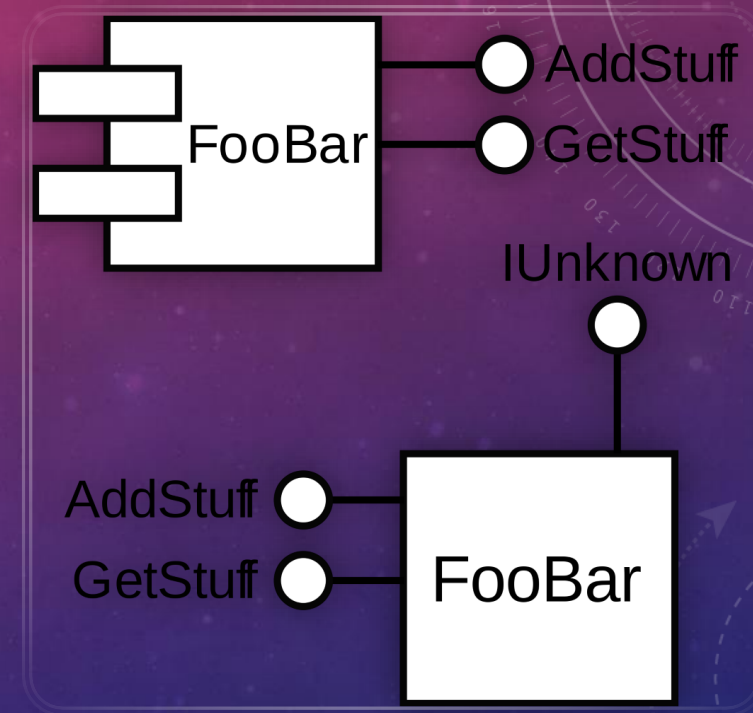
QUELQUES DEFINITIONS

PARADIGME DE PROGRAMMATION

- C'est la manière dont on perçoit, interprète, pense la solution à un problème donné. Cela permet de produire la solution à un problème en utilisant un langage de programmation adapté. Un langage de programmation répond en général à un paradigme en particulier.
- Par exemple, la Programmation Orientée Objet (POO) est un paradigme de programmation, qui conçoit la solution sous forme d'objets qui interagissent entre eux. Le java, lui, est un langage qui répond au paradigme de POO.
- Conception de la solution <-- Langage de programmation
- En savoir plus sur les paradigmes de programmation:
- [https://fr.wikipedia.org/wiki/Paradigme_\(programmation\)#Liste_de_paradigmes](https://fr.wikipedia.org/wiki/Paradigme_(programmation)#Liste_de_paradigmes)

QUELQUES DEFINITIONS PROGRAMMATION ORIENTÉE COMPOSANTS (POC)

- La programmation orientée composant (POC) consiste à utiliser une approche modulaire de l'architecture d'un projet informatique, ce qui permet d'assurer au logiciel une meilleure lisibilité et une meilleure maintenance. Les développeurs, au lieu de créer un exécutable monolithique, se servent de briques réutilisables.
- En savoir plus à propos de la programmation orientée composants:
- https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_composant



QUELQUES DÉFINITIONS

DATAFLOW PROGRAMMING & FLOW BASED PROGRAMMING

DataFlow Programming	Flow-based Programming
Paradigme de programmation Définition sous forme de graphe orienté Modelise le flux de données entre opérations Variété des langages de programmation y répondant	Paradigme de programmation Utilise des connections prédéfinies à des applications Spécifie un réseau entre des applications sous forme de "black boxes" Hérite du concept de POC Cas particulier de DataFlow programming

QUELQUES DÉFINITIONS

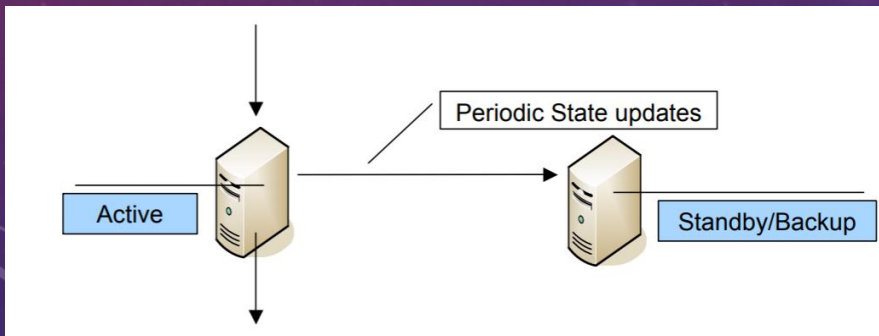
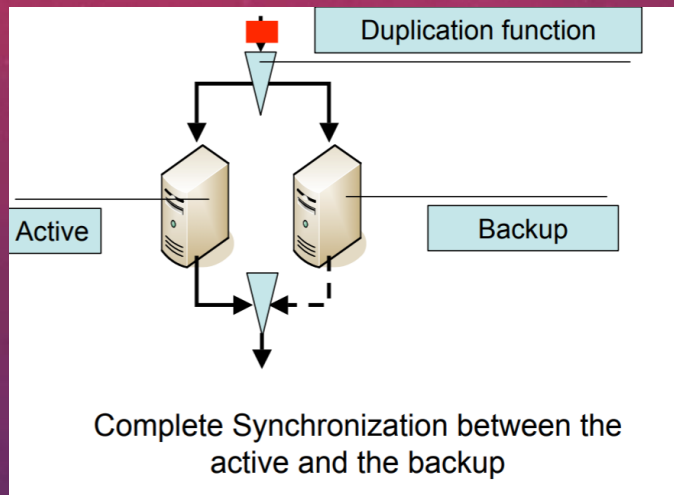
TOLÉRANCE AUX FAUTES (FAULT TOLERANCE)

- Capacité d'un système de délivrer un service :
 - Même en présence d'erreurs imprévues
 - Et Restaurer après une erreur
- Prérequis opérationnels: scalabilité, Interopérabilité, pas de perte de données, transparence
- Afin d'assurer la tolérance aux fautes, 3 configurations existent:
 - Hot Standby
 - Cold Standby
 - Warm Standby
- En savoir plus: <https://www.sanog.org/resources/sanog8/sanog8-fault-tolerant-sana-tariq.pdf>

QUELQUES DÉFINITIONS

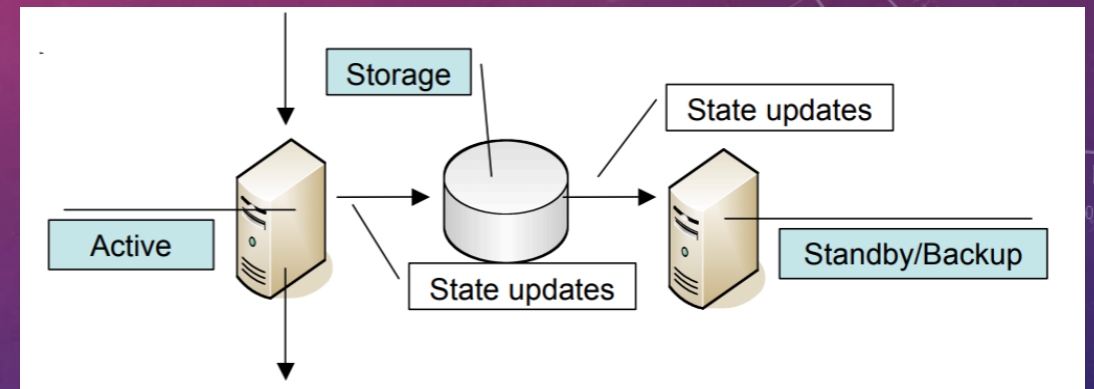
TOLÉRANCE AUX FAUTES (FAULT TOLERANCE)

Hot standby (synchro complète mais gâchis ressources)



Cold Standby

(différé vers un stockage fiable - m à j état, mais impacte l'utilisateur lors de la récup: préparation du serveur de secours à partir de la sauvegarde)



Warm standby (checkpoints réguliers vers un serveur de secours, un peu plus temps réel que cold standby)

PRÉSENTATION D'APACHE NIFI

- Principe général
- Historique
- Architecture
- Limites



APACHE NIFI

PRINCIPE GÉNÉRAL

- Automatisation de flux de données entre systèmes
- Flow-based programming: nous évite d'écrire des tonnes de lignes de code
- Contrôle en temps réel, sécurisation de données
- Tolère les pannes, scalable, gros volumes de données
- Projet open source d'Apache, supporté par Hortonworks
- Apache NiFi est compatible avec Kerberos qui assure l'authentification, avec Apache Ranger qui permet la sécurité des autorisations d'accès et avec Apache Knox qui gère la sécurité au niveau authentication et celle des appels REST and HTTP.
- Beaucoup de possibilités (293 processeurs dans Nifi 1.9.2)

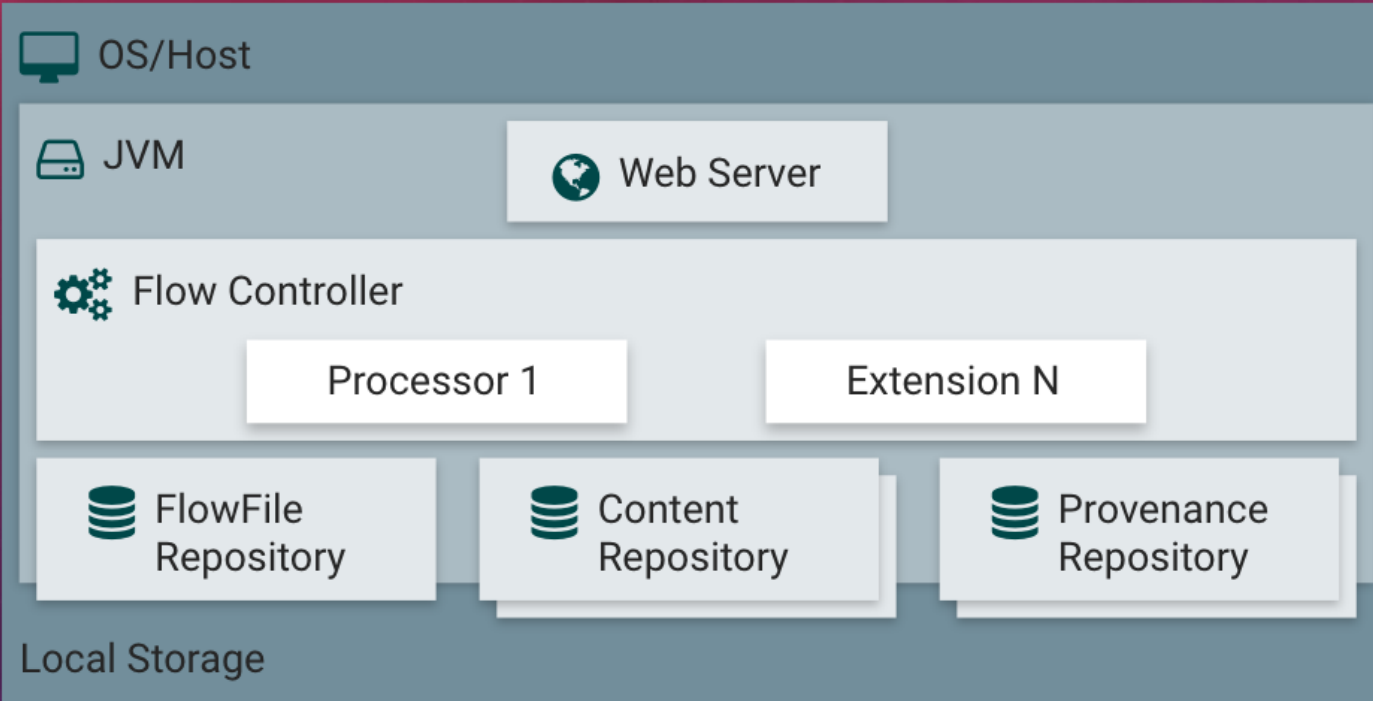


APACHE NIFI HISTORIQUE

- Le logiciel est initialement un projet interne de la National Security Agency (NSA), débuté en 2006 et nommé Niagarafiles. Son développement est alors assuré par l'entreprise Onyara.
- En novembre 2014, la NSA libère le projet dans le cadre de son programme de transfert de technologies et le confie à l'incubateur de la fondation Apache.
- En juillet 2015, NiFi devient un des projets de premier niveau de la fondation Apache.
- En 2016, NiFi est en développement actif au sein de la fondation Apache et plusieurs versions du logiciel ont été publiées.



APACHE NIFI ARCHITECTURE



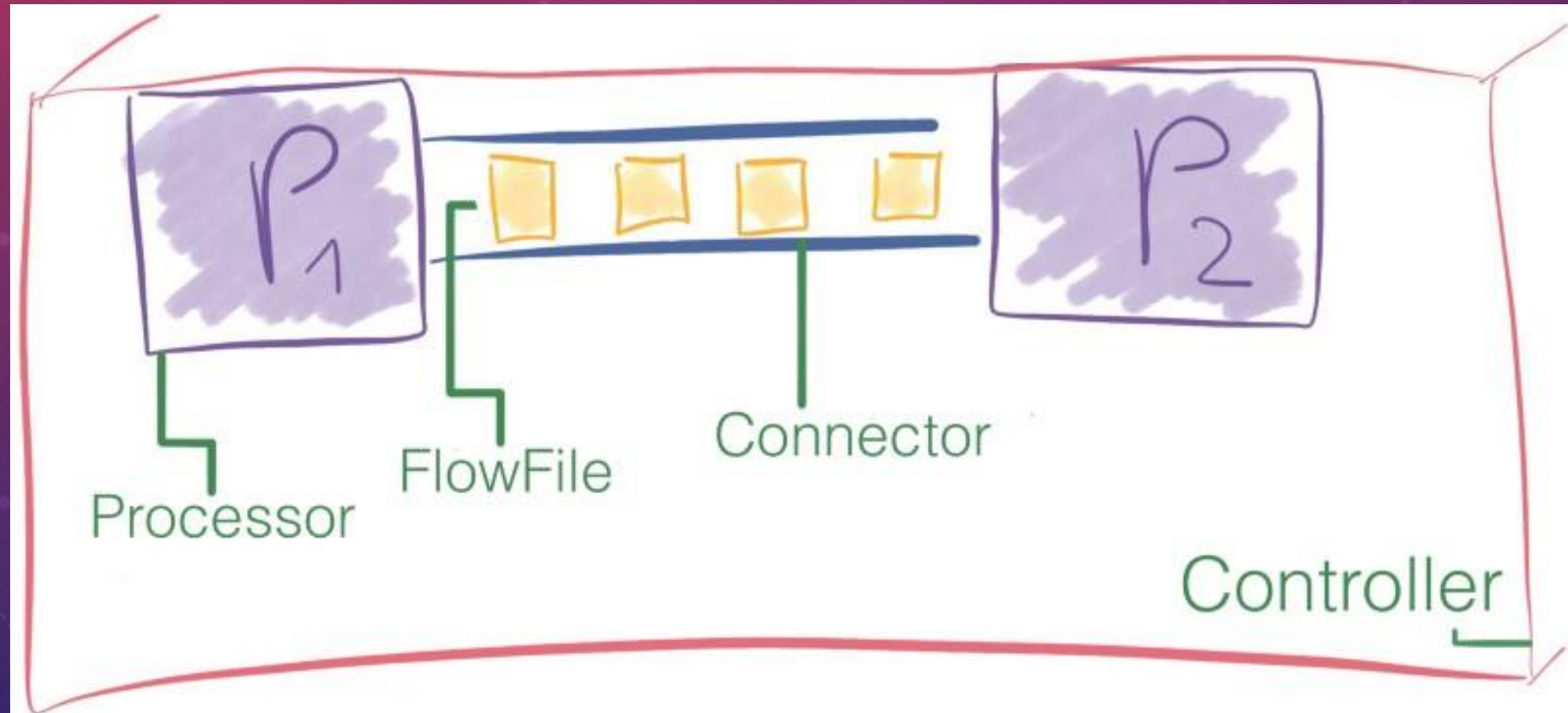
- Documentation d'origine Nifi:
- <https://nifi.apache.org/docs/nifi-docs/html/user-guide.html>
- **Encore plus simple:**
- <https://www.freecodecamp.org/news/nifi-surf-on-your-dataflow-4f3343c50aa2/>

APACHE NIFI

ARCHITECTURE

Simple concepts à connaître:

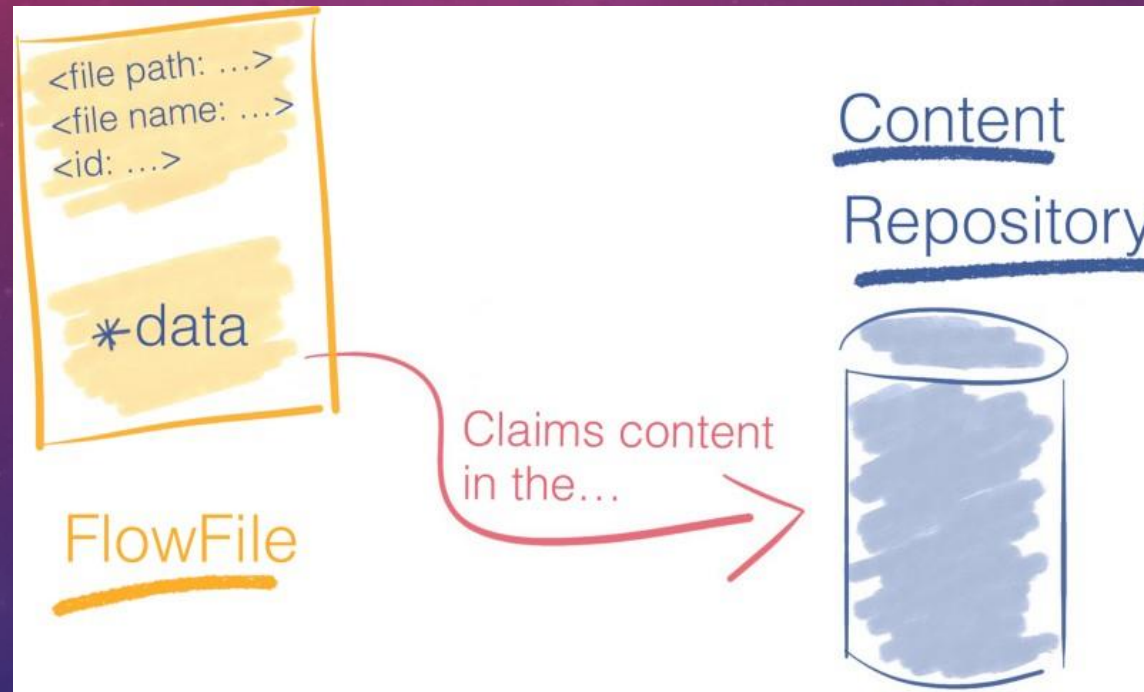
Les processeurs sont reliés entre eux via des connecteurs et forment un réseau à travers lequel circulent les FlowFiles. Le tout étant encapsulé dans un contrôleur.



APACHE NIFI

ARCHITECTURE

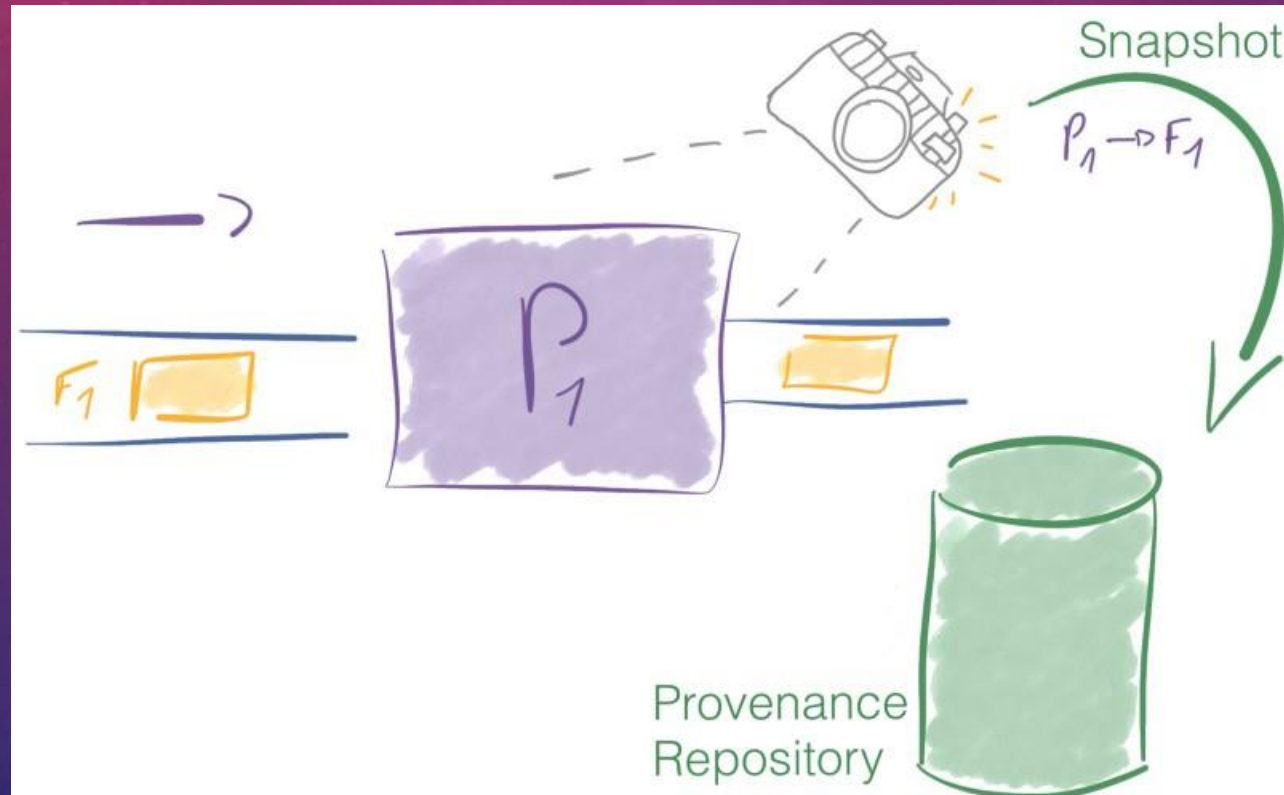
- FlowFile: Packet d'informations qui circule à travers le circuit et les processeurs.
- Composé d'attributs clé valeur et d'un contenu
- Le contenu du message est en réalité stocké dans le Content Repository et pointé depuis le FlowFile (performances)



APACHE NIFI

ARCHITECTURE

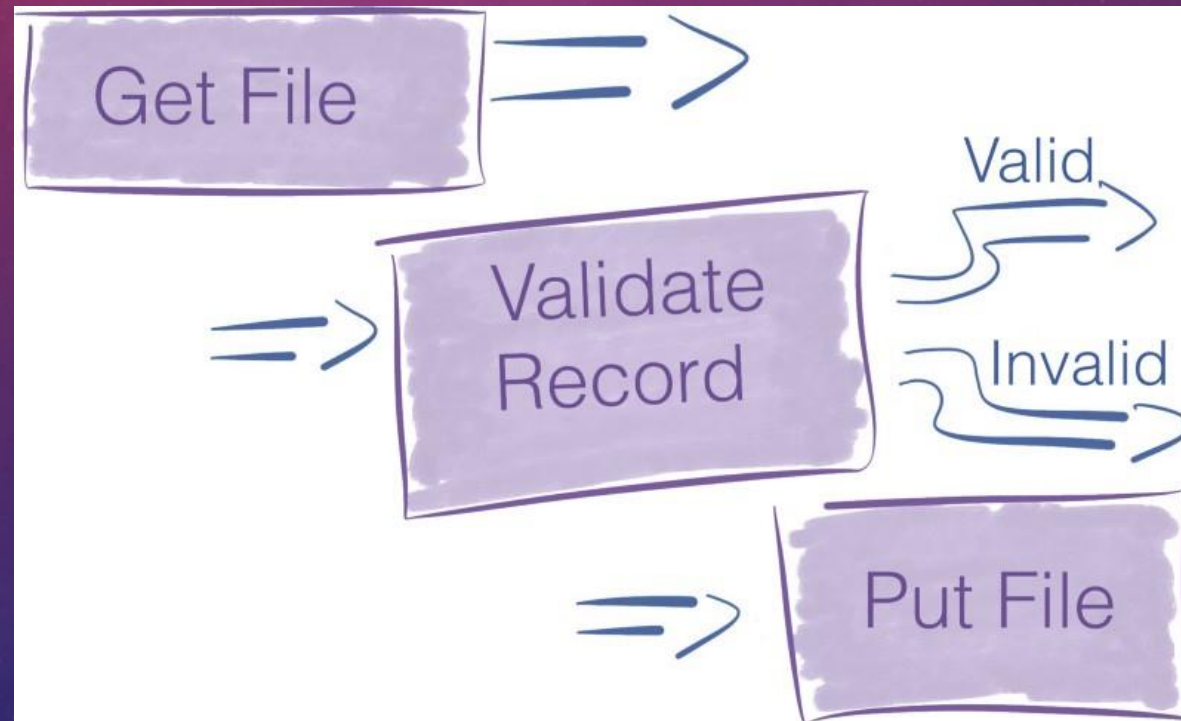
- Content Repository: Enregistre le contenu des FileFlows et ne les supprime jamais
- FlowFile Repository: Contient le dernier état des FlowFiles
- Provenance Repository: Possède un historique complet des modifications des informations



APACHE NIFI

ARCHITECTURE

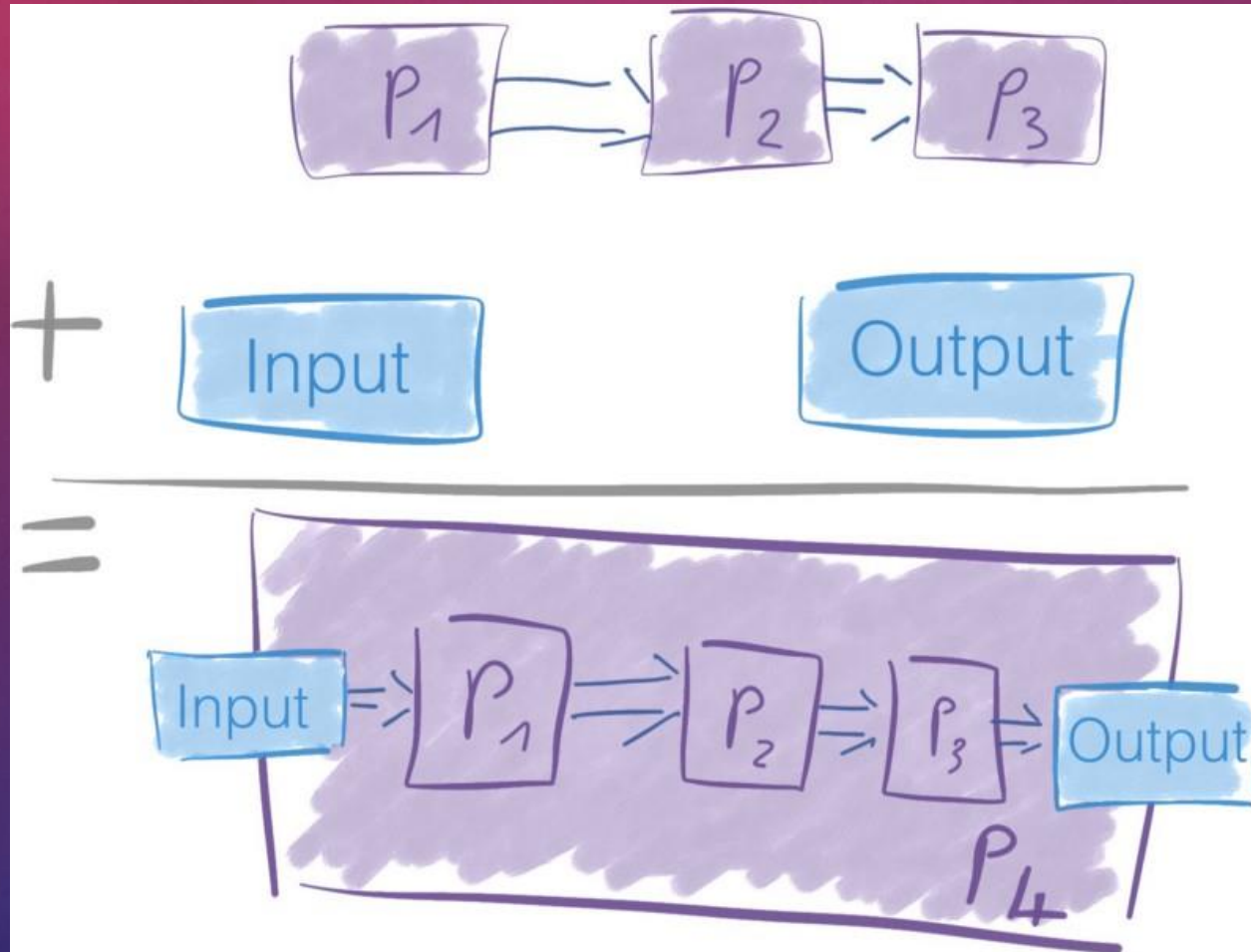
- Processors — Boîtes noires faisant des opérations sur le contenu des FlowFiles
- Operations: entrée, sortie, standardisation, validation, transformation
- Peuvent accéder aux attributs



APACHE NIFI

ARCHITECTURE

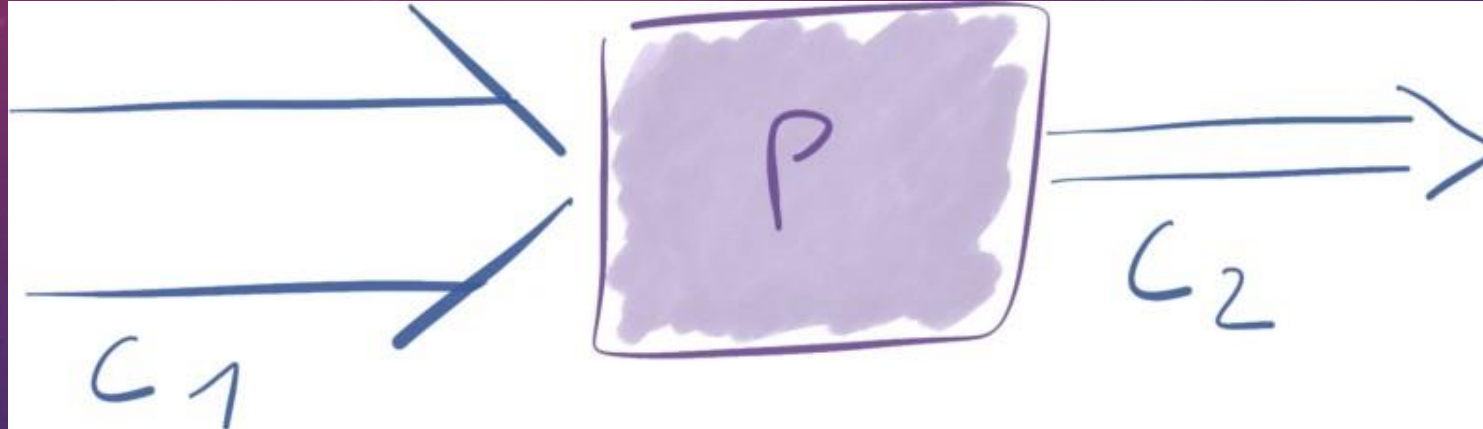
- Groupes (Process Groups): faire des processeurs à partir d'un groupe de processeurs déjà définis



APACHE NIFI

ARCHITECTURE

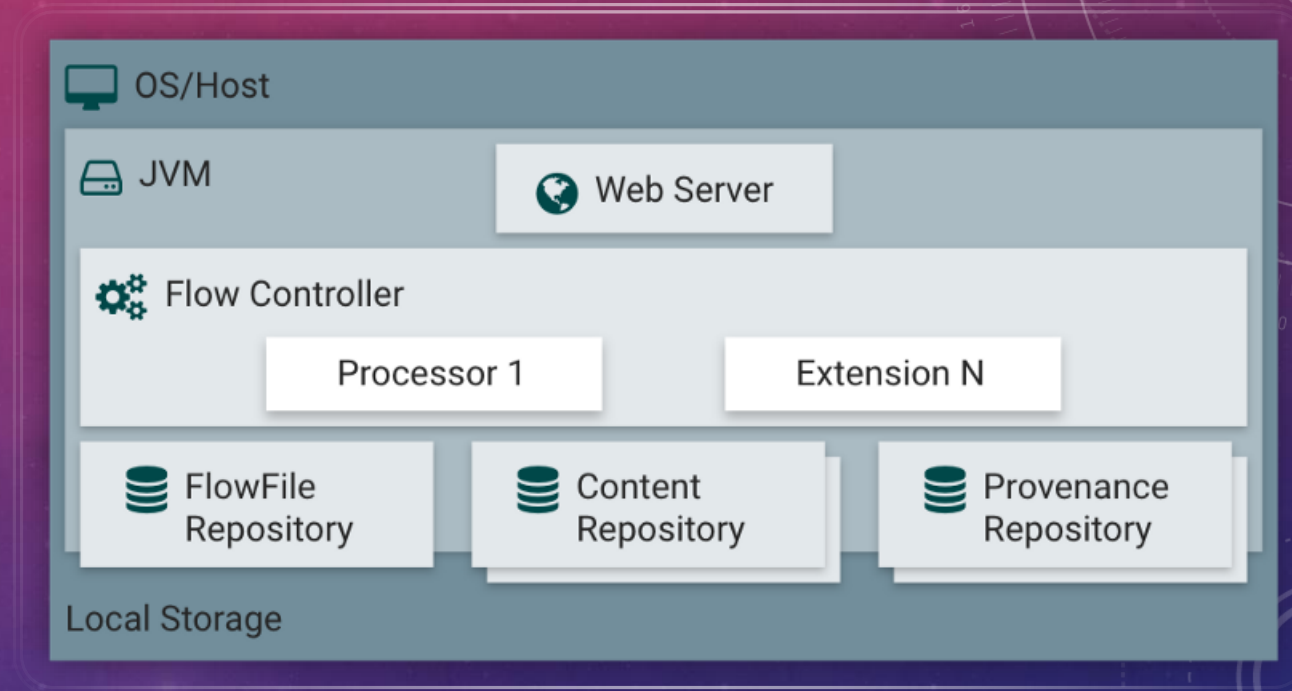
- Connections: queues entre les processeurs
- Lorsque la queue est pleine, le processeur précédent est bloqué



APACHE NIFI ARCHITECTURE

Controlleur (Flow Controller)

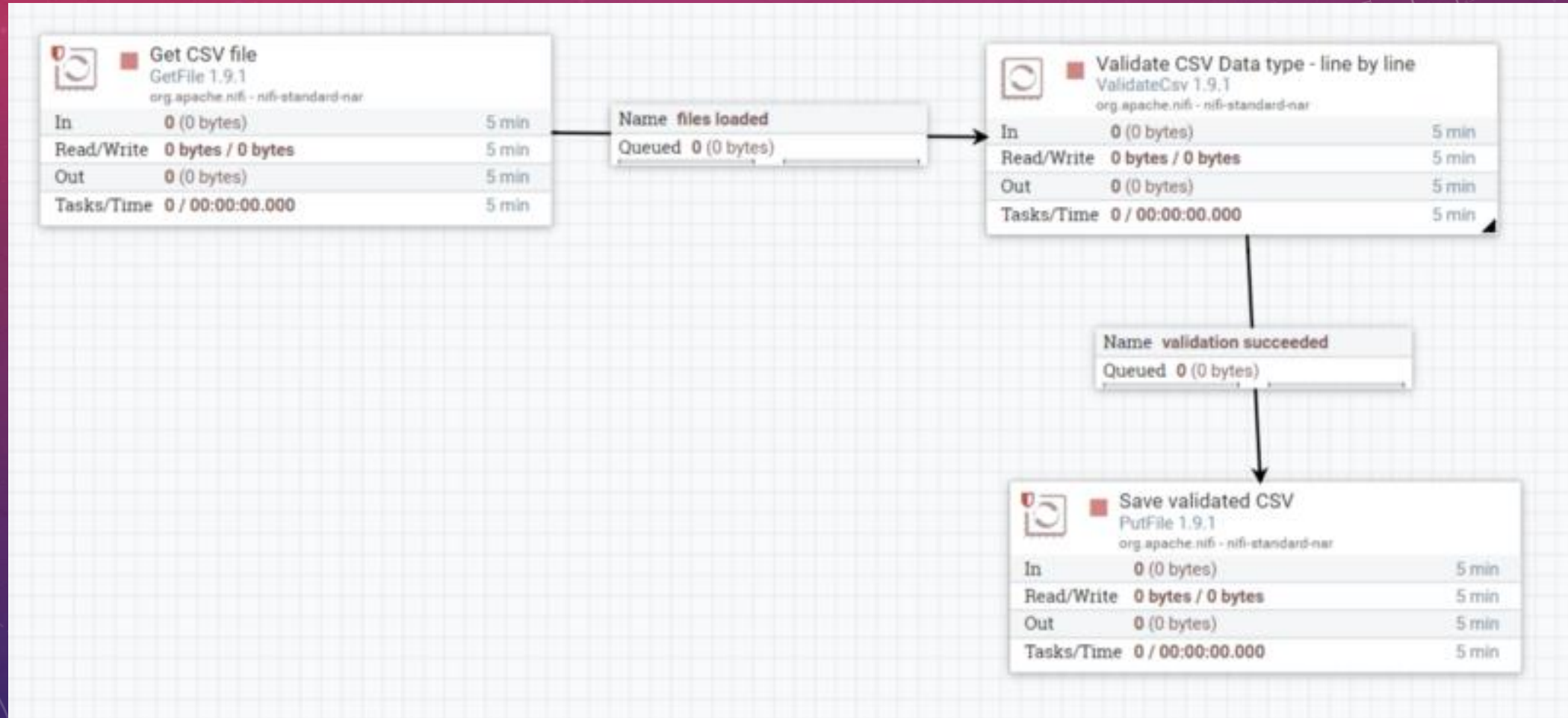
- Permet d'allouer les ressources aux processeurs et queues et manage l'ensemble.



APACHE NIFI

ARCHITECTURE

Exemple simple de circuit de données sous Nifi



DIFFERENCE AVEC MINIFI

SOUS PROJET DE NIFI



DIFFERENCE AVEC MINIFI

- Adapté pour des petites ressources ou petites tailles d'objets
- Entière traçabilité de l'information depuis la source elle-même
- Intégration facile avec Nifi et la poursuite du flux de données
- Management centralisé de l'information proche de la source
- Pas d'interface de développement
- Moins de différents processeurs prédéfinis

--> Sert à l'ingestion de données de manière plus simple et efficace dans un environnement Nifi et IOT

DÉPLOIEMENT D'APACHE NIFI

- Méthode 1 : avec les fichiers sources
- Méthode 2: Docker-compose



DÉPLOIEMENT D'APACHE NIFI

MÉTHODE 1 : AVEC LES FICHIERS SOURCES

- Télécharger les fichiers sources [nifi-1.9.2-bin.zip](https://nifi.apache.org/download.html) sur le site officiel <https://nifi.apache.org/download.html> et décompresser le dossier

bin	26/09/2019 00:27	Dossier de fichiers	
conf	03/10/2019 12:08	Dossier de fichiers	
content_repository	26/09/2019 00:49	Dossier de fichiers	
database_repository	03/10/2019 12:08	Dossier de fichiers	
docs	26/09/2019 00:27	Dossier de fichiers	
extensions	03/04/2019 15:37	Dossier de fichiers	
flowfile_repository	03/10/2019 12:08	Dossier de fichiers	
lib	26/09/2019 00:27	Dossier de fichiers	
logs	03/10/2019 15:00	Dossier de fichiers	
provenance_repository	02/10/2019 11:50	Dossier de fichiers	
run	03/10/2019 12:08	Dossier de fichiers	
state	26/09/2019 00:49	Dossier de fichiers	
work	03/10/2019 12:08	Dossier de fichiers	
LICENSE	26/09/2019 00:27	Fichier	119 Ko
NOTICE	26/09/2019 00:27	Fichier	82 Ko
README	26/09/2019 00:27	Fichier	5 Ko

DÉPLOIEMENT D'APACHE NIFI

MÉTHODE 1 : AVEC LES FICHIERS SOURCES

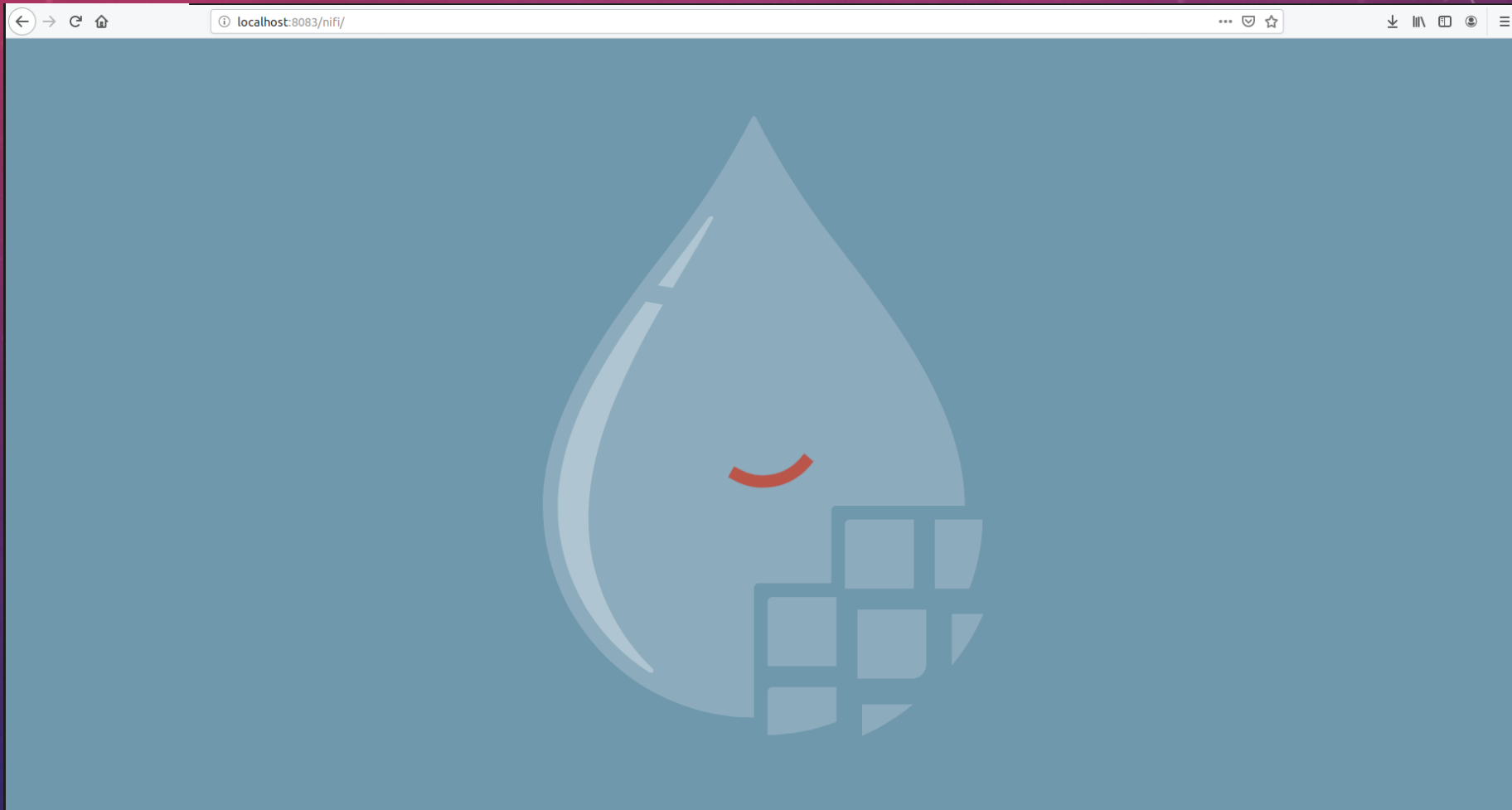
- Ouvrir le fichier /conf/nifi-properties
- Changer le port 8080 par un de disponible (par exemple 9090)

```
# web properties #
nifi.web.war.directory=./lib
nifi.web.http.host=
nifi.web.http.port=9090
nifi.web.http.network.interface.default=
nifi.web.https.host=
nifi.web.https.port=
nifi.web.https.network.interface.default=
nifi.web.jetty.working.directory=./work/jetty
nifi.web.jetty.threads=200
nifi.web.max.header.size=16 KB
nifi.web.proxy.context.path=
nifi.web.proxy.host=
```

DÉPLOIEMENT D'APACHE NIFI

MÉTHODE 1 : AVEC LES FICHIERS SOURCES

- Exécuter le fichier `bin/run-nifi.bat`
- Ouvrir le navigateur web et entrer l'url `http://localhost:9090/nifi/`



DÉPLOIEMENT D'APACHE NIFI

MÉTHODE 2: DOCKER-COMPOSE

- Utiliser un OS avec Docker installé (Mac ou Linux)
- Créer les fichiers .yaml suivant:

Nifi-docker-compose.yaml

```
version: '3.0'
services:
  nifi:
    image: "apache/nifi:1.9.2"
    hostname: "nifi"
    environment:
      NIFI_WEB_HTTP_PORT: "8080"
    ports:
      - "8083:8080"
    networks:
      - nifi-labs
    labels:
      NAME: "nifi"

networks:
  nifi-labs:
    external: true
```

DÉPLOIEMENT D'APACHE NIFI

MÉTHODE 2: DOCKER-COMPOSE

- Via le terminal, se placer dans le même répertoire que le fichier .yml
- Exécuter la commande:

```
docker-compose -f Nifi-docker-compose.yml up -d
```

```
user@user-vm:~/Documents$ docker-compose -f Nifi-docker-compose.yml up -d
WARNING: Found orphan containers (documents_ftp_1, documents_mongo-express_1, documents_mongo_1) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphans flag to clean it up.
documents_nifi_1 is up-to-date
user@user-vm:~/Documents$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
cca71003dad1 documents_ftp_1	fauria/vsftpd	"/usr/sbin/run-vsftpd..."	2 hours ago	Up 7 minutes	20/tcp, 0.0.0.0:21->21/tcp
9ef776260b4b documents_nifi_1	apache/nifi:1.9.2	"../scripts/start.sh"	2 hours ago	Up 5 minutes	8443/tcp, 10000/tcp, 0.0.0.0:8083->8083/tcp
9cb6fc6f0812 documents_mongo-express_1	mongo-express:0.49.0	"tini -- /docker-entrypoint.sh"	29 hours ago	Up 7 minutes	0.0.0.0:8081->8081/tcp
6a23d97d1f49 documents_mongo_1	mongo:4.2.0-bionic	"docker-entrypoint.sh"	29 hours ago	Up 7 minutes	0.0.0.0:27017->27017/tcp
0032e389bc08 documents_rabbitmq1_1	rabbitmq:3-management	"docker-entrypoint.sh"	7 days ago	Exited (0) 7 days ago	

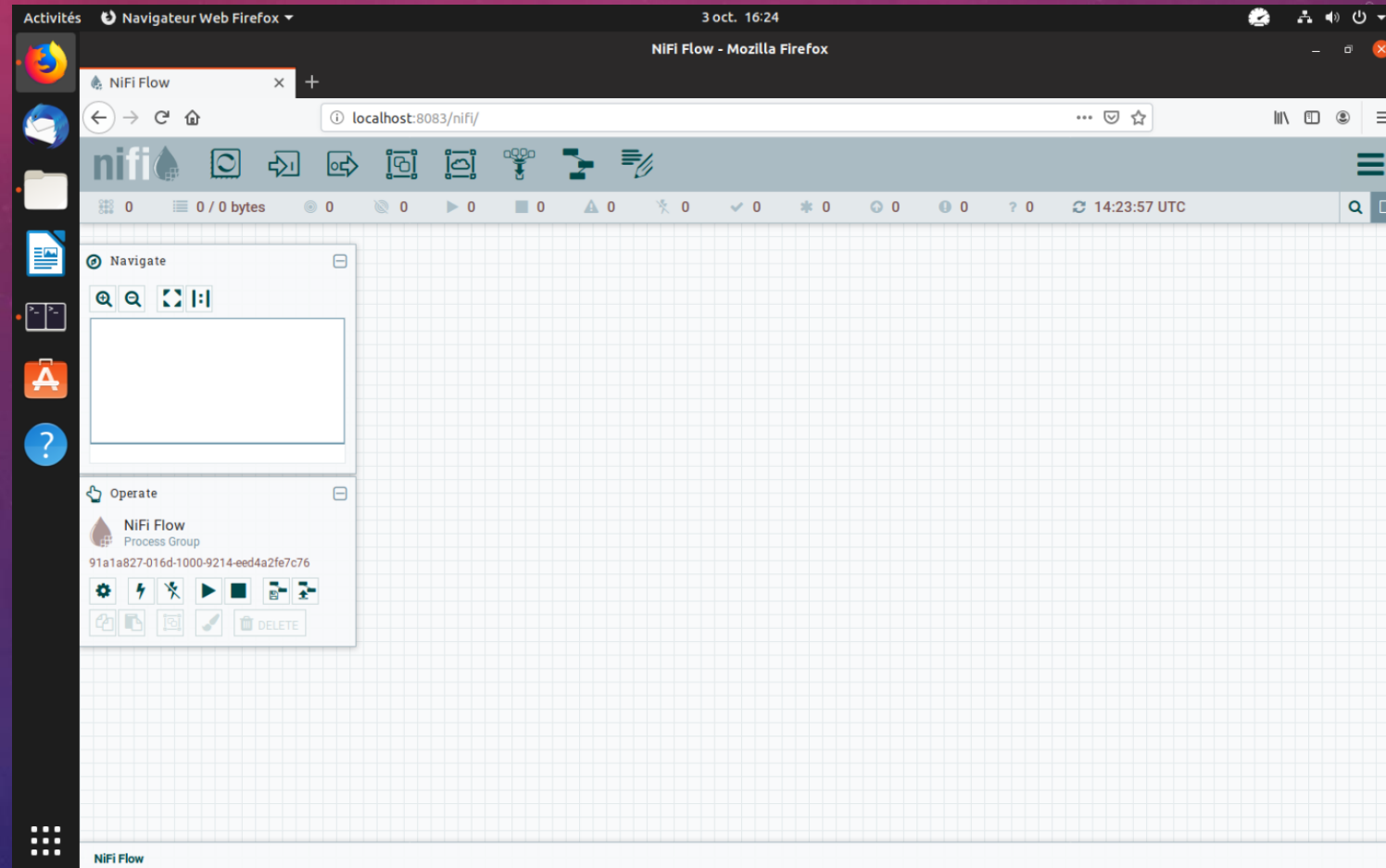
```
user@user-vm:~/Documents$
```

DÉPLOIEMENT D'APACHE NIFI

MÉTHODE 2: DOCKER-COMPOSE

- Accéder à Nifi via le navigateur web avec:

<http://localhost:8083/nifi/>



DEMONSTRATION SOUS NIFI

