

Laboratoire Ouvert en Learning Analytics

Atelier LOLA

Les intervenants

Chahrazed LABBA

Philippe NOEL

PLAN

1- Aperçu de LOLA

> Présenter le site et l'application web LOLA

2- Vous êtes un Fournisseur de Données

> Convertir vos données en xAPI et les uploader vers un LRS

3- Vous êtes un Scénariste

> Créer un scénario à partir de zéro

4- Vous êtes un Challenger

> présenter la Notion des Algorithmes Interchangeables

Aperçu de LOLA

- Liens utiles:

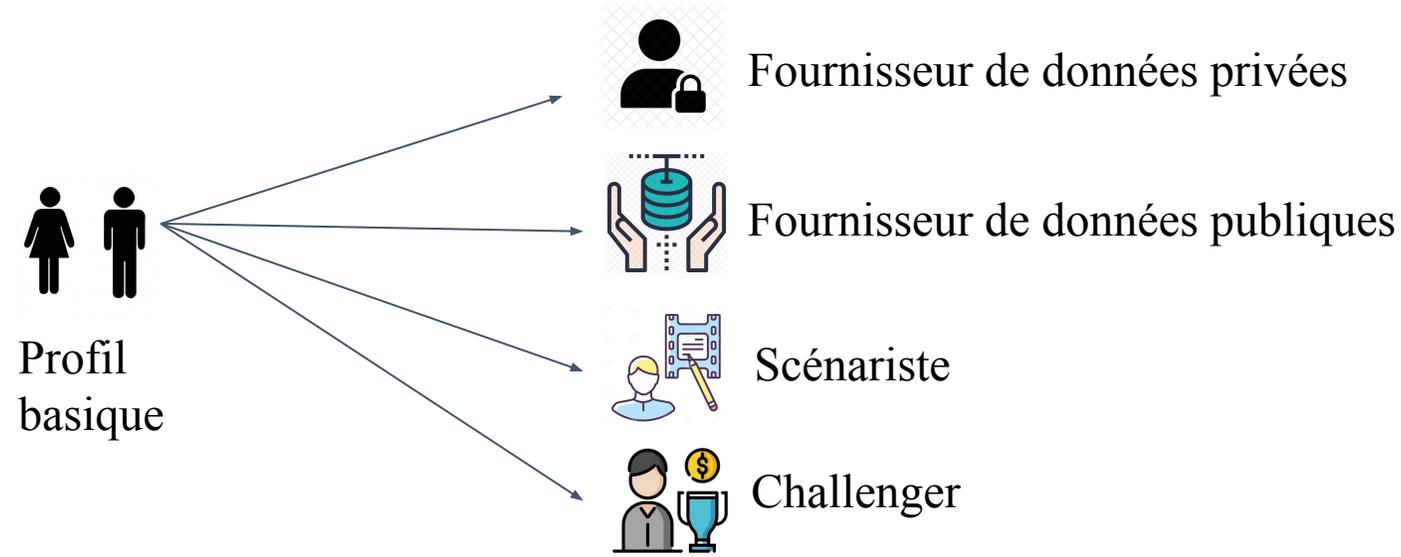
- site Public <https://lola.loria.fr/>
- application Web <https://lola.lhs.loria.fr>
- documentation <lola.loria.fr/ressources>

- Fichiers de l'atelier:

<https://gitlab.inria.fr/lola/atelier-lola.git>

Aperçu de LOLA

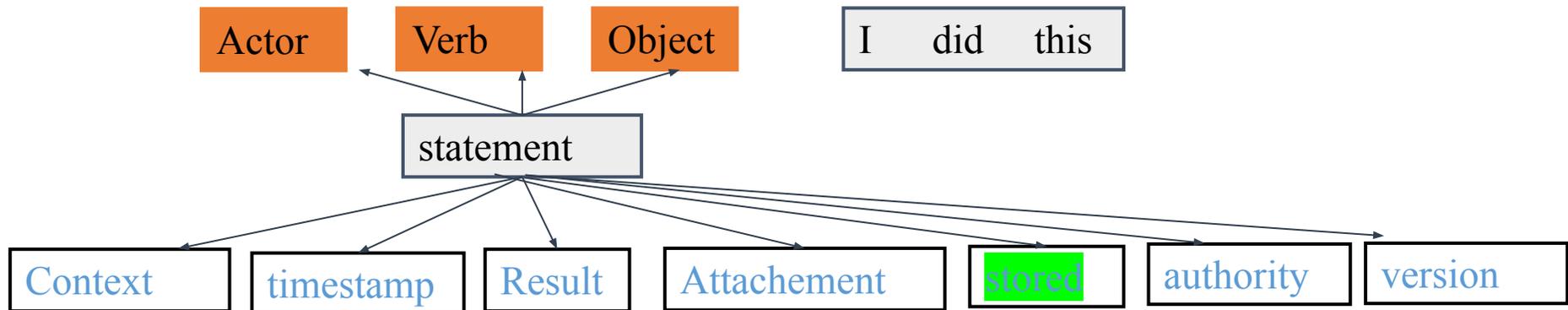
- Les profils utilisateurs



Vous êtes un Fournisseur de Données

- LOLA et xAPI

- ☞ LOLA utilise un « Learning Record Store » pour stocker les données: TraxLRS
- ☞ Experience API= xAPI = Tin Can
- ☞ Statements pour décrire les activités d'apprentissage



Vous êtes un Fournisseur de Données

- Exemple d'un Statement xAPI

👉 Un statement xAPI est un objet JSON: *Bob viewed Java course 2 times on 13 june 2020*

```
{
  "actor": {
  },
  "verb": {
  },
  "object": {
  }
}
```

Vous êtes un Fournisseur de Données

- Exemple d'un Statement xAPI

👉 Un statement xAPI est un objet JSON: **Bob** viewed Java course 2 times on 13 june 2020

```
{
  "actor": {
    "name": "Bob",
    "mbox": "mailto:Bob@gmail.com"
  },
  "verb": {
  },
  "object": {
  }
}
```

Vous êtes un Fournisseur de Données

- Exemple d'un Statement xAPI

👉 Un statement xAPI est un objet JSON: Bob **viewed** Java course 2 times on 13 june 2020

```
{
  "actor": {
    "name": "Bob",
    "mbox": "mailto:Bob@gmail.com"
  },
  "verb": {
    "id": "http://adlnet.gov/exapi/verbs/viewed",
    "display": { "en-US": "viewed" }
  },
  "object": {
  }
}
```

Vous êtes un Fournisseur de Données

- Exemple d'un Statement xAPI

👉 Un statement xAPI est un objet JSON: Bob viewed **Java course** 2 times on 13 june 2020

```
{
  "actor": {
    "name": "Bob",
    "mbox": "mailto:Bob@gmail.com"
  },
  "verb": {
    "id": "http://adlnet.gov/exapi/verbs/viewed",
    "display": { "en-US": "viewed" }
  },
  "object": {
    "id": "http://univ-lorraine/ arche/coursJava"
  }
}
```

Vous êtes un Fournisseur de Données

- Exemple d'un Statement xAPI

👉 Un statement xAPI est un objet JSON: Bob viewed Java course 2 times on 13 june 2020

```
{
  "actor": {
    "name": "Bob",
    "mbox": "mailto:Bob@gmail.com"
  },
  "verb": {
    "id": "http://adlnet.gov/exapi/verbs/viewed",
    "display": { "en-US": "viewed" }
  },
  "object": {
    "id": "http://univ-lorraine/arche/coursJava"
  },
  "context": {
    "extensions": {
      "http://univ-lorraine/arche/click": 2
    }
  }
}
```

Vous êtes un Fournisseur de Données

- Exemple d'un Statement xAPI

👉 Un statement xAPI est un objet JSON: Bob viewed Java course 2 times on 13 June 2020

```
{
  "actor": {
    "name": "Bob",
    "mbox": "mailto:Bob@gmail.com"
  },
  "verb": {
    "id": "http://adlnet.gov/exapi/verbs/viewed",
    "display": { "en-US": "viewed" }
  },
  "object": {
    "id": "http://univ-lorraine/arche/coursJava"
  },
  "context": {
    "extensions": {
      "http://univ-lorraine/arche/click": 2
    }
  },
  "timestamp": "2020-06-13T00:00:00"
}
```

Vous êtes un Fournisseur de Données

Profil xAPI

‘A profile is the human or machine readable documentation of application-specific concepts, extensions, and statement templates used when implementing xAPI in a particular context.’

FromADL



Vous êtes un Fournisseur de Données

Que spécifient le profil xAPI?



Activity

Defines a specific interaction with an object by an actor in a statement. It can be a unit of instruction, a work task — any effort with meaningful purpose that can be tracked with a verb.



Document

Defines information about the data to be stored in the State, Agent Profile and Activity Profile Resources of xAPI. It can be a means to structure data for contextually relevant information that can be abstracted from xAPI statement structure.



Activity Type

Defines the type of activity in a broader context: ie. a course; a video; a book; an assessment; a competency; a job role.



Extension

Defines a specific piece of information to be expressed in the Context, Result and Activity Extensions of a relevant xAPI statement: ie. a confidence score for a systemic assertion on a learner's record; the evidence that resulted in earning a badge.



Attachment Usage Type

Defines the type of usage of an attachment to a statement. For example, a certificate issued upon completion of an eLearning course.



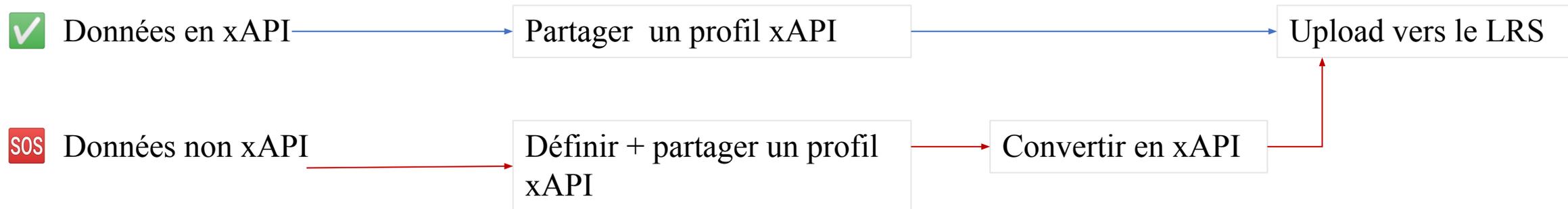
Verb

Defines an identified action performed by the actor in an xAPI statement.

Vous êtes un Fournisseur de Données

profil xAPI?

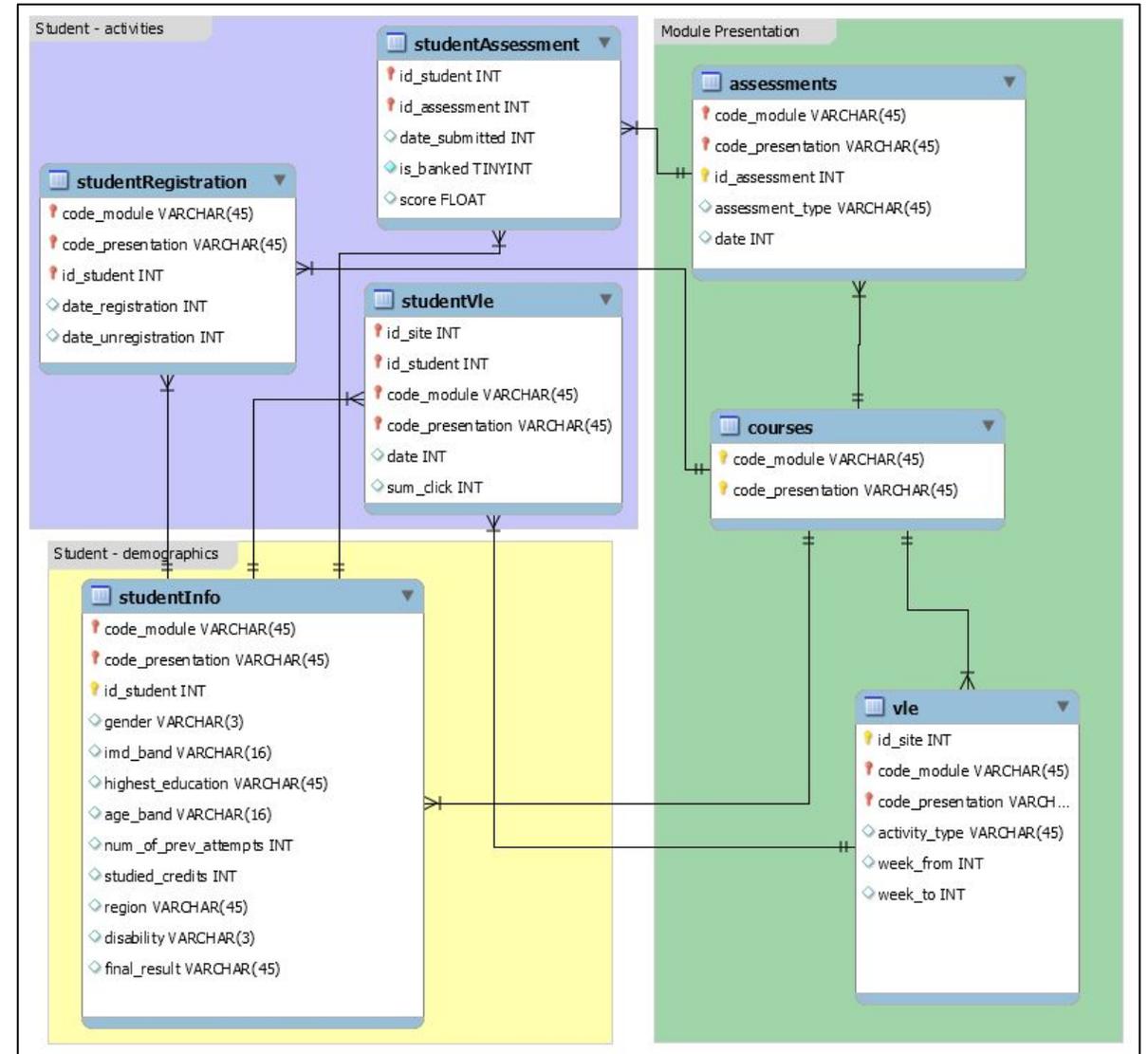
- xAPI Profiles Server (2020): <https://profiles.adlnet.gov>



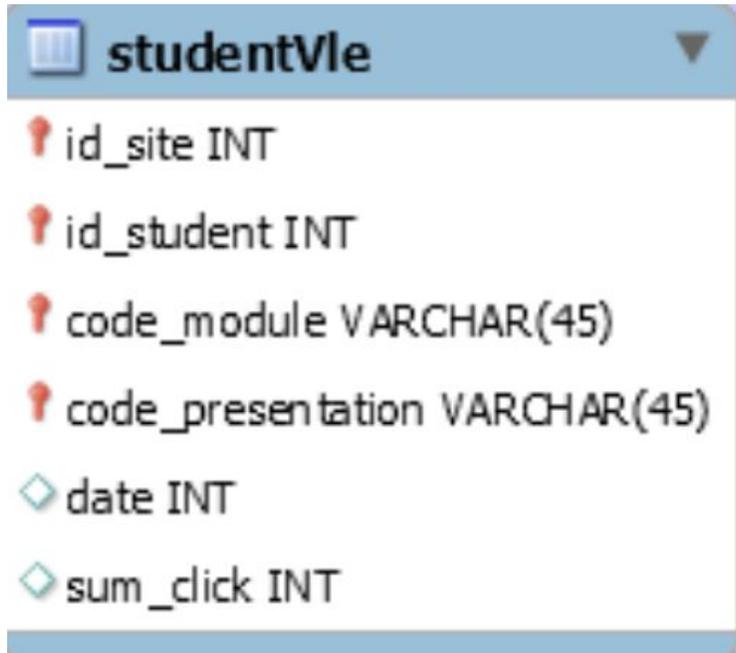
Vous êtes un Fournisseur de Données

- Les données OULAD

- [Open University Learning Analytics Dataset](#)
- Open source
- Format csv
- Il contient des données sur les cours, les étudiants et leurs interactions avec le Virtual Learning Environment (VLE).

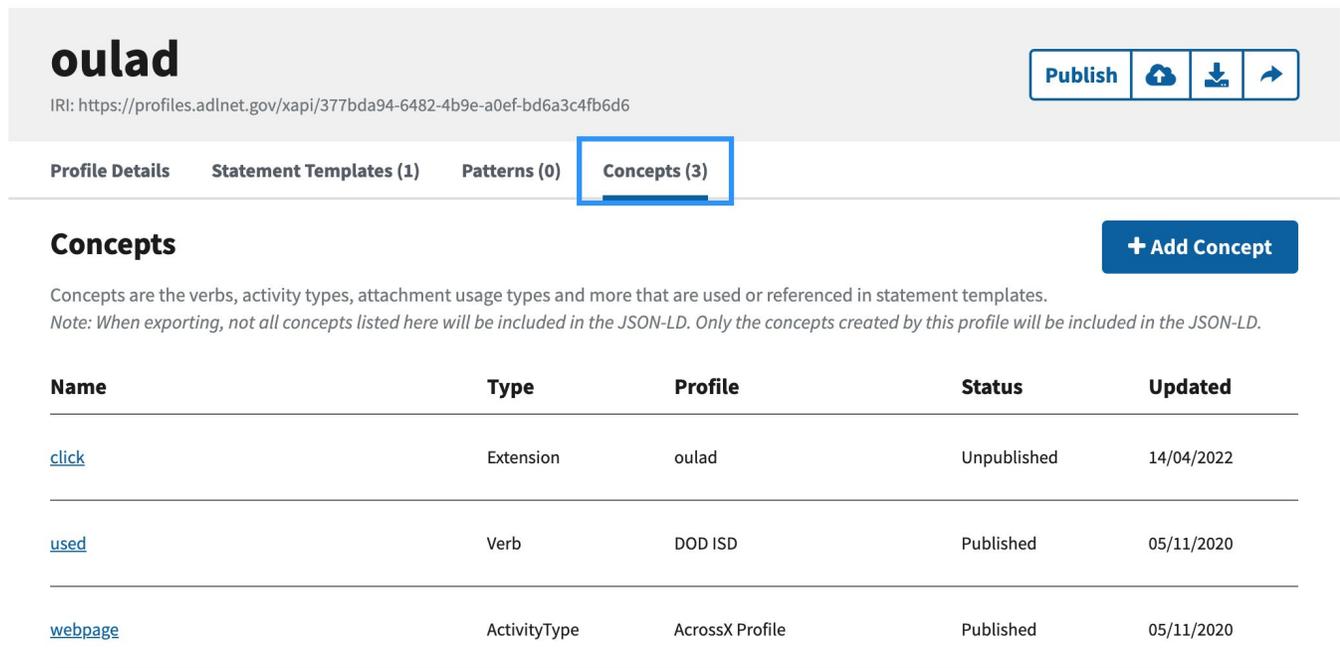


Vous êtes un Fournisseur de Données



The screenshot shows a configuration window for a data source named 'studentVle'. It lists several fields with their data types and status icons:

- id_site INT (red pin icon)
- id_student INT (red pin icon)
- code_module VARCHAR(45) (red pin icon)
- code_presentation VARCHAR(45) (red pin icon)
- date INT (blue diamond icon)
- sum_click INT (blue diamond icon)



The screenshot shows the 'oulad' profile management interface. It includes a header with the profile name, an IRI, and action buttons (Publish, Refresh, Download, Share). Below the header are tabs for Profile Details, Statement Templates (1), Patterns (0), and Concepts (3). The Concepts tab is active, showing a list of concepts with columns for Name, Type, Profile, Status, and Updated. A '+ Add Concept' button is also visible.

oulad
IRI: <https://profiles.adlnet.gov/xapi/377bda94-6482-4b9e-a0ef-bd6a3c4fb6d6>

Profile Details Statement Templates (1) Patterns (0) **Concepts (3)**

Concepts [+ Add Concept](#)

Concepts are the verbs, activity types, attachment usage types and more that are used or referenced in statement templates.
Note: When exporting, not all concepts listed here will be included in the JSON-LD. Only the concepts created by this profile will be included in the JSON-LD.

Name	Type	Profile	Status	Updated
click	Extension	oulad	Unpublished	14/04/2022
used	Verb	DOD ISD	Published	05/11/2020
webpage	ActivityType	AcrossX Profile	Published	05/11/2020

Profile xAPI OULAD

Vous êtes Fournisseur de Données

- OULAD en xAPI

StudentVle.csv

code_module	code_presentiel	d_student	id_site	date	sum_click
AAA	2013J	28400	546652	-10	4
AAA	2013J	28400	546652	-10	1
AAA	2013J	28400	546652	-10	1
AAA	2013J	28400	546614	-10	11

Actor Object Timestamp context
Verb: used

Statement xAPI

```
{  
  "actor": {  
    "mbox": "mailto:28400@open.ac.uk"  
  },  
  "verb": {  
    "id": "https://w3id.org/xapi/dod-isd/verbs/used"  
  },  
  "object": {  
    "id": "http://open.ac.uk/546652",  
  },  
  "context" : {  
    "extensions": {  
      "http://open.ac.uk/click": 4,  
    }  
  },  
  "timestamp": "2013-09-21T00:00:00"  
}
```

Vous êtes Fournisseur de Données

- Convertir Oulad en xAPI



Paramètres:

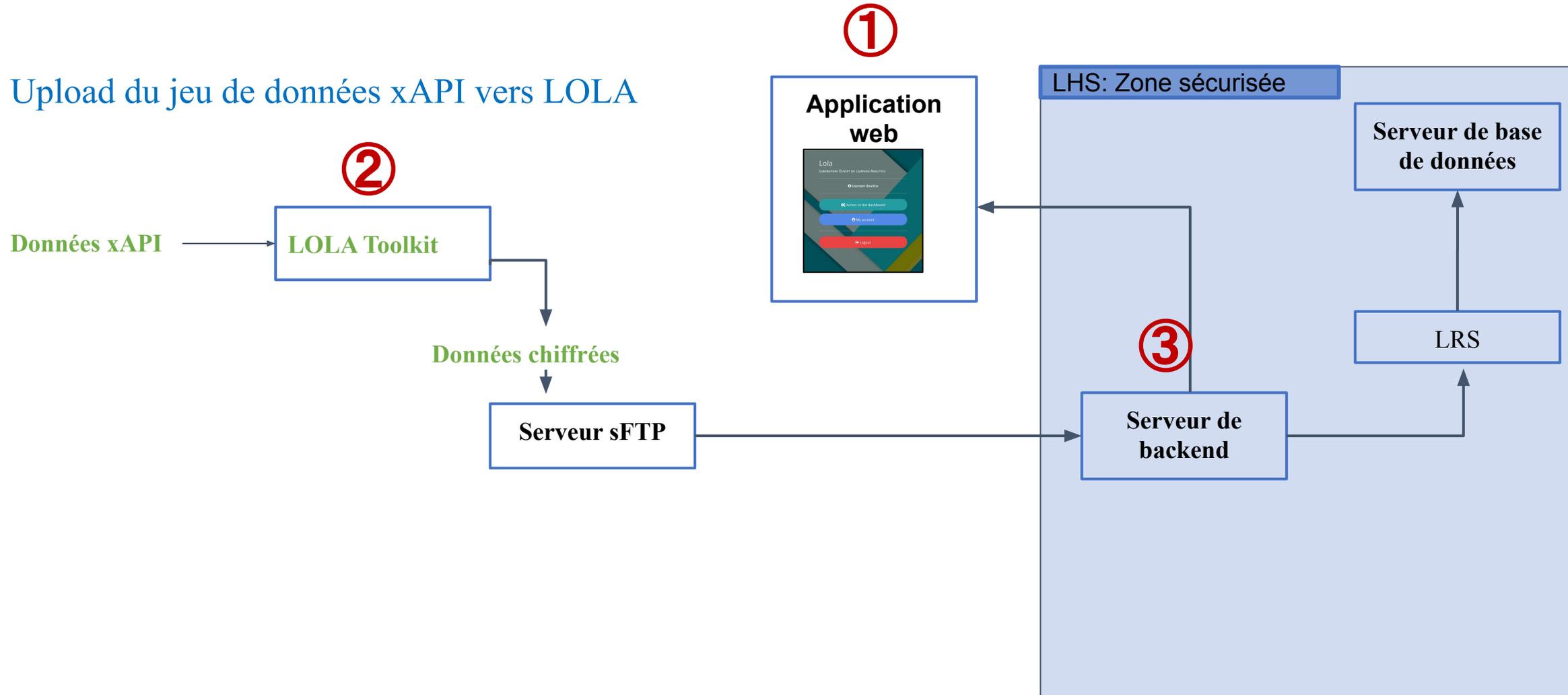
-i, --input <arg>	l'emplacement du fichier .csv
-n, --number <arg>	seuil de partitionnement
-o, --output <arg>	l'emplacement des fichiers .json

1. Aller dans le dossier atelier-lola/csv-to-xapi
2. Utiliser le programme Java pour convertir le fichier CSV avec la commande:

```
java -jar oulad.jar -i train.csv -n 100000 -o .
```

Vous êtes un Fournisseur de Données

Upload du jeu de données xAPI vers LOLA



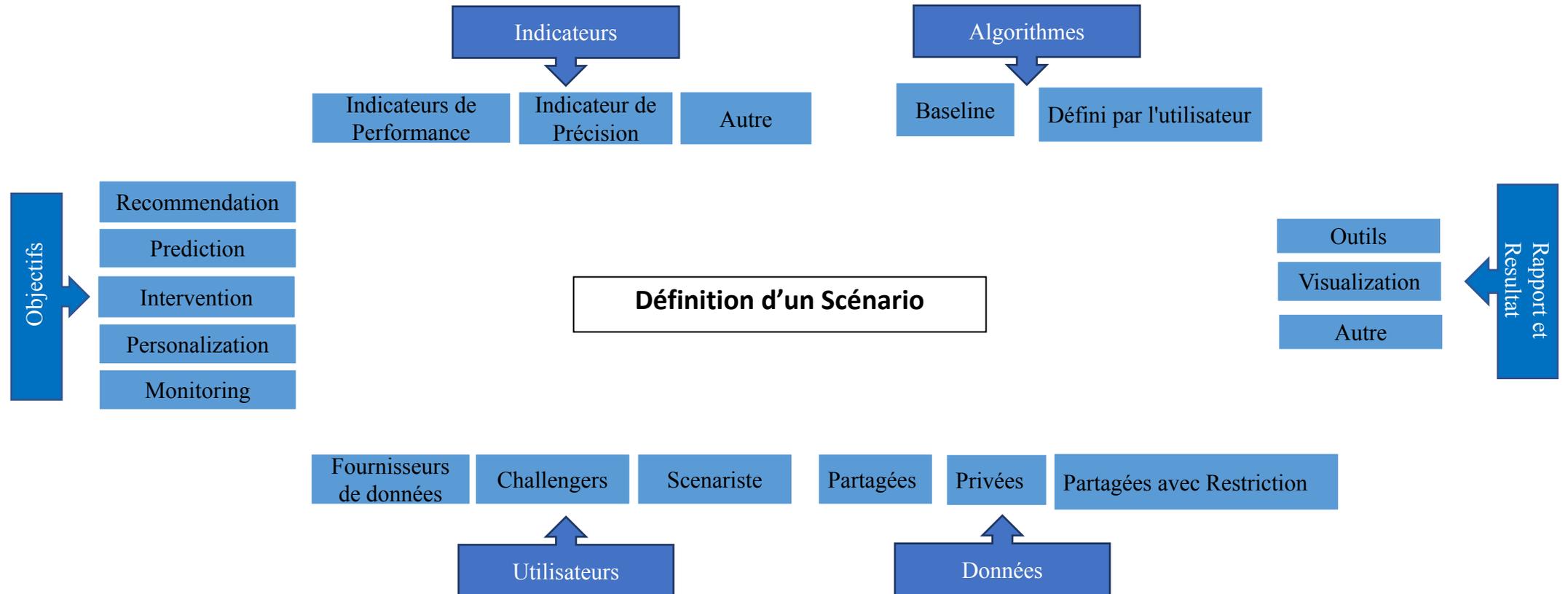
Vous êtes un Fournisseur de Données

- Upload du jeu de données xAPI vers Lola:

1. Déclarer un nouveau dataset dans l'application web Lola: <https://lola.lhs.loria.fr/dashboard/dataset/>
2. Télécharger le fichier bordereau
3. Télécharger l'outil pour l'envoi des données: Lola Toolbox: https://gitlab.inria.fr/lola/lola_toolbox/-/releases
4. Utiliser Lola Toolbox pour téléverser le jeu de données sur la plateforme
 - Compression du bordereau + du fichier xAPI
 - Chiffrement à l'aide de la clef publique
 - Envoie sur le serveur sFTP de la plateforme
5. Voir l'état d'avancement dans l'interface web

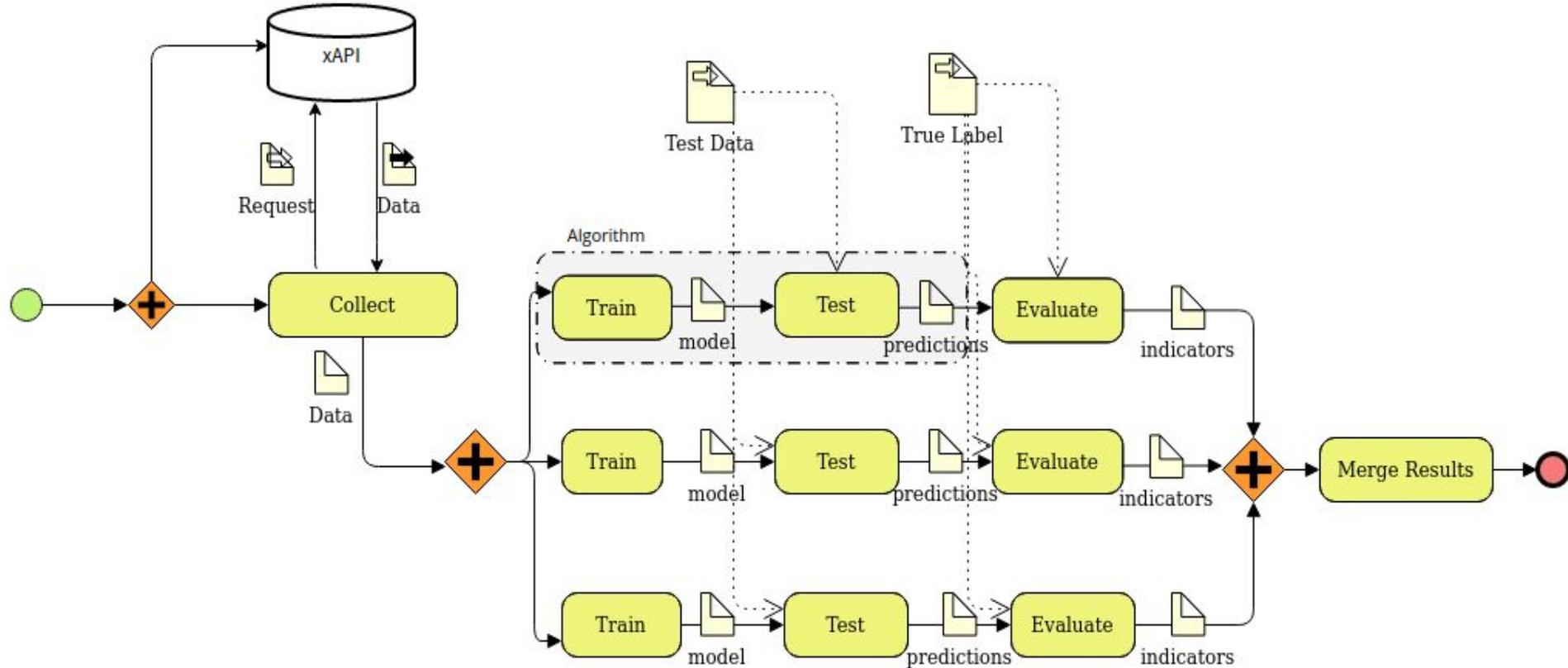
Un Scénario dans LOLA

- C'est quoi un Scénario LOLA?
 - 6 dimensions différentes mais complémentaires.



Un Scénario dans LOLA

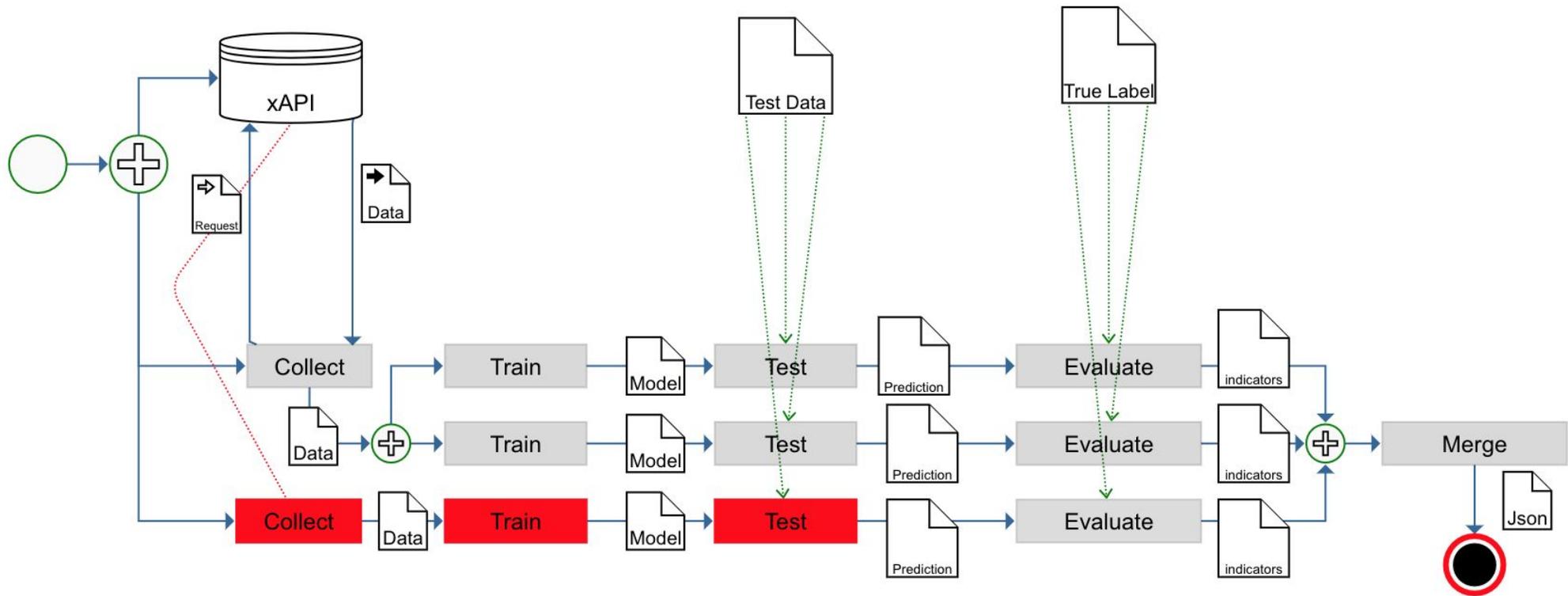
- Un scénario en Pratique
 - Un workflow des algorithm



Un Scénario dans LOLA

Un scénario en Pratique

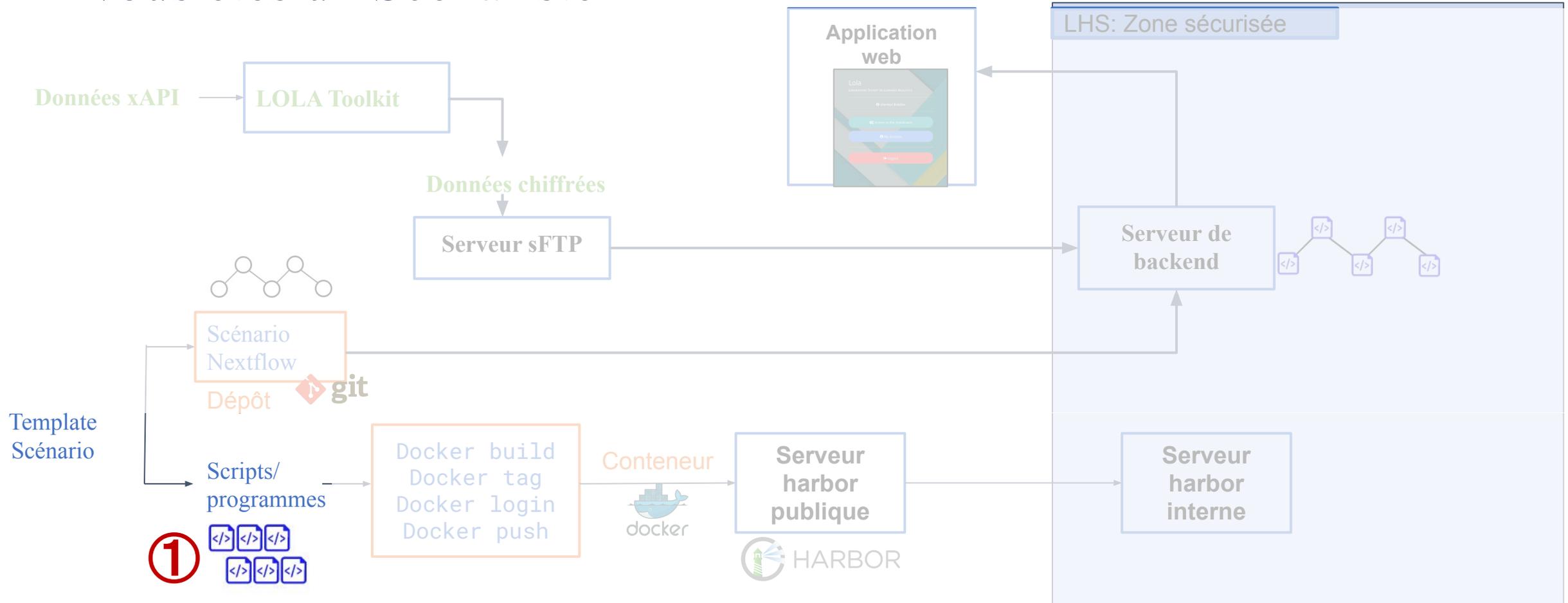
- Template de scénario + Algorithme de Challenger



Vous êtes un Scénariste

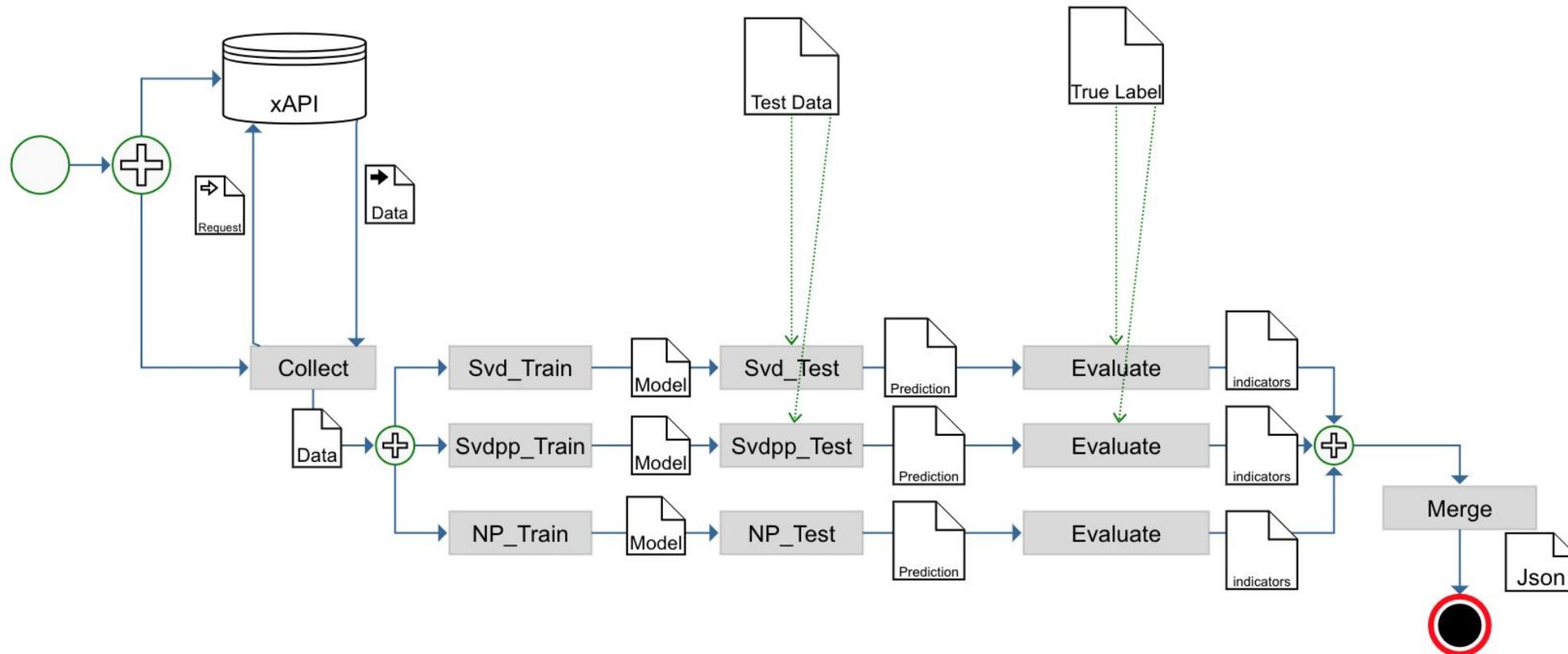
- Créez votre premier template de scénario LOLA
 - **Objectif:** Recommandation des ressources éducatives : Collaborative Filtering
 - **Données:** OULAD (shared)
 - **Algorithmes de Baseline:** SVD, SVD++, Normal Predictor
 - **Indicateurs:** RMSE, MSE, MAE, FCP
 - **Rapport:** Json File
 - **Utilisateurs:** Scénariste

Vous êtes un Scénariste



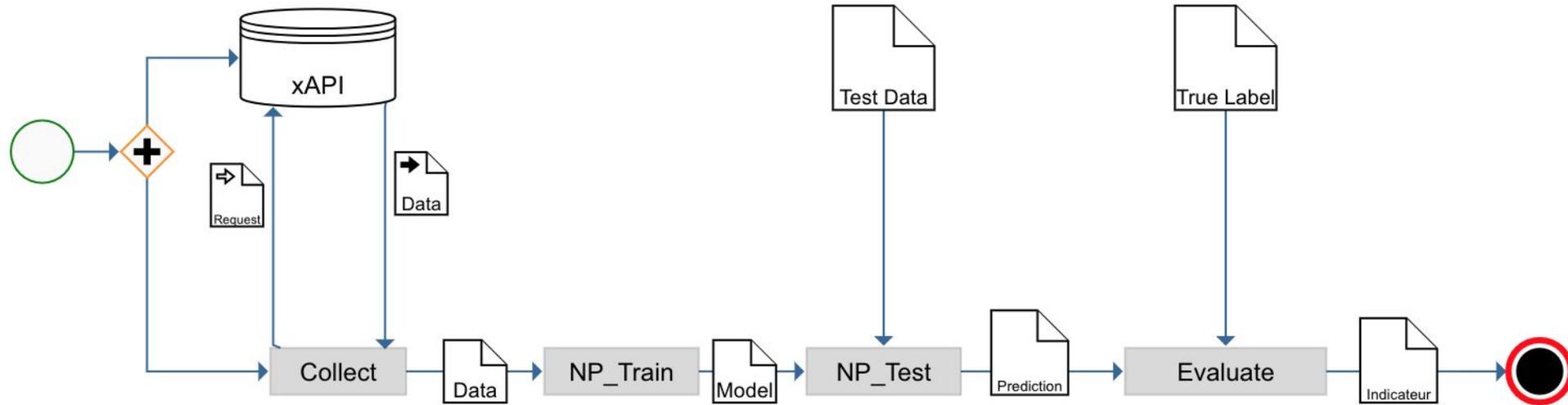
Vous êtes un Scénariste

Le Template de scénario



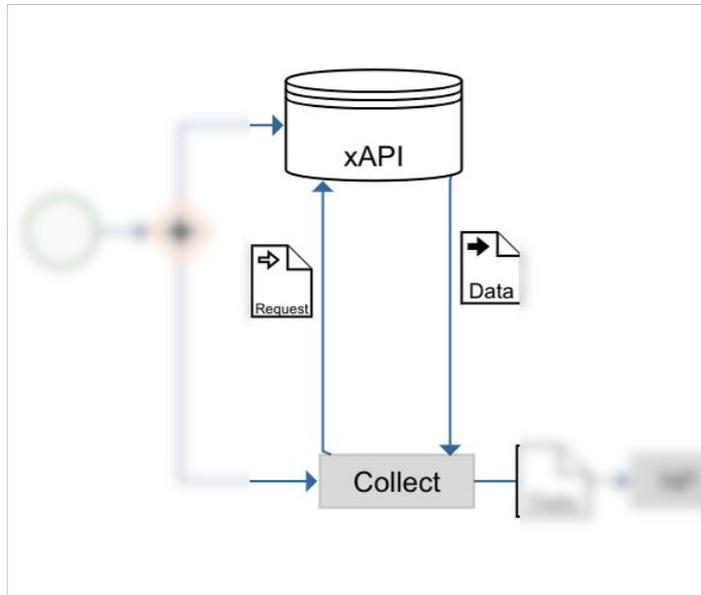
Vous êtes un Scénariste

- Créez votre premier scénario LOLA



Vous êtes un Scénariste

- Collecte de données

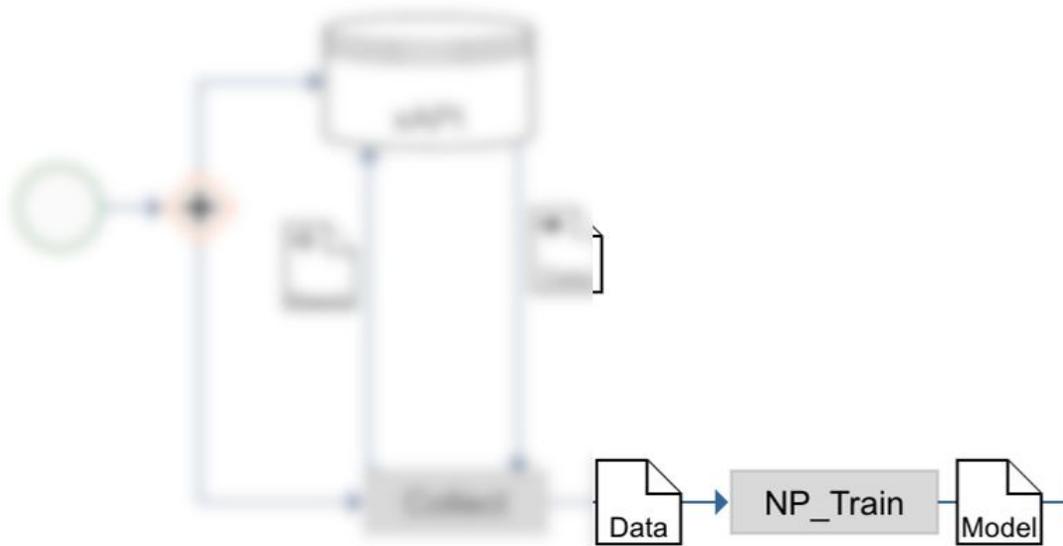


- Un script python: **request.py**
- Input** : Les paramètres de la requête
- Output** : Un fichier csv formaté et normalisé
- Les Paramètres**
 - `--h <arg>` : headers
 - `--p <arg>` : paramètres
 - `--a <arg>` : authentication parameters,
 - `--url <arg>` :URL traxLRS
 - `--port <arg>` : port traxLRS
 - `--c <arg>` :csv file
- Fonctionnement** : Fonction **build_user_request**
Fonction **prepare_data**

```
python request.py --h X-Experience-API-Version=1.0.3 --p since=2013-09-21T00:00:00 --a  
testsuite password --url http://garimpeiro12.loria.fr --port 80 --c train_data.csv
```

Vous êtes un Scénariste

- Entraîner un modèle: Algorithme Normal Predictor

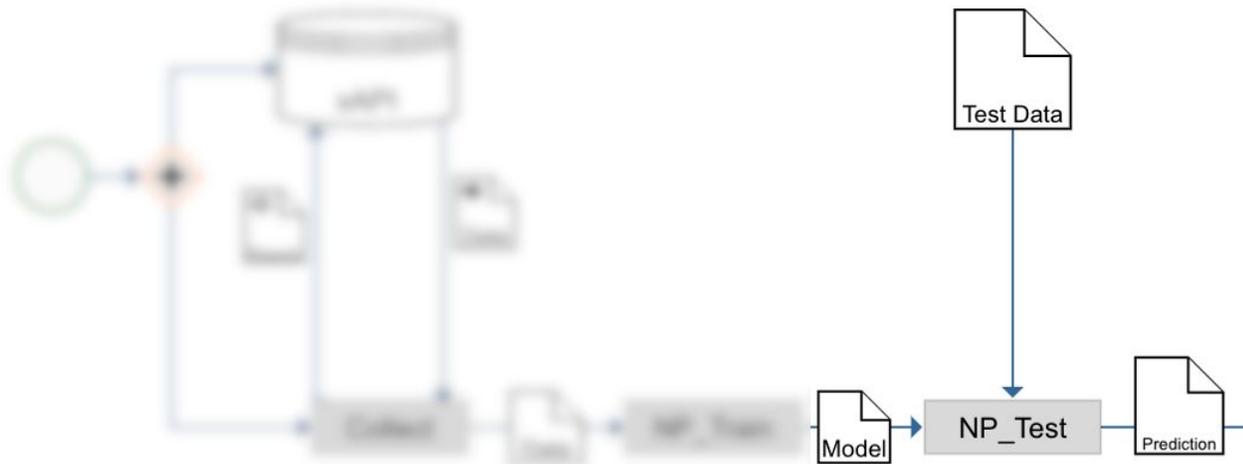


- Un script python: **normalPredictor_train.py**
- **Input** : Données d'entraînement
- **Output** : Un modèle entraîné (.sav)
- **Les Paramètres** :
 - `--a` <arg> : emplacement de données d'entraînement
 - `--m` <arg> : nom du fichier pour sauvegarder le modèle
- **Fonctionnement** : Fonction **train_NormalPredictor**

```
python normalPredictor_train.py --a train_data.csv --m normalPred
```

Vous êtes un Scénariste

- Tester un modèle: Algorithme Normal Predictor



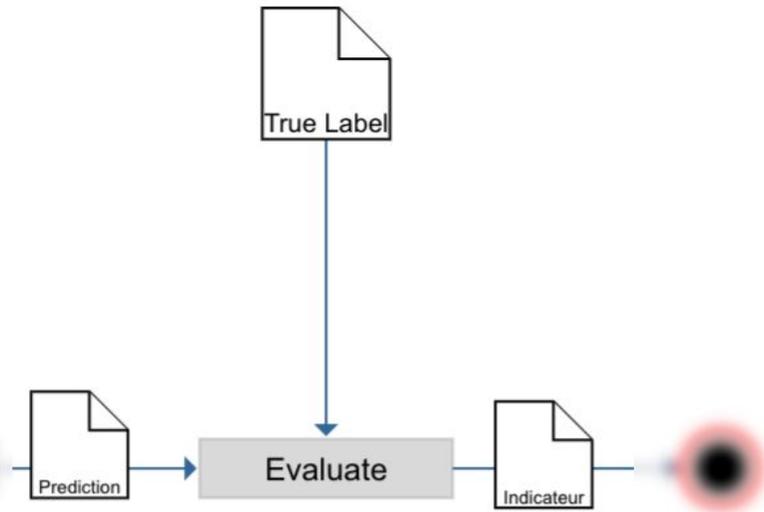
- Un script python: **model_test.py**
- **Input** : modèle entraîné (.sav), données de test sans labels (.csv)
- **Output** : des prédictions (.csv)
- **Les Paramètres** :
 - `--m` <arg> (modèle entraîné) ,
 - `--t` <arg> (emplacement de données de test),
 - `--f` <arg> (nom de fichier pour sauvegarder les prédictions),
- **Fonctionnement** : Fonction **model_test**

```
python model_test.py --m normalPred.sav --t test.csv --f NP_prediction
```

Vous êtes un Scénariste

- Évaluer un modèle

- Un script python: **evaluate.py**
- Input** : les prédictions (csv), données de test labels (.csv)
- Output** : les évaluations (json)
- Les Paramètres** :
 - r (emplacement pour les données de test (true labels))
 - p (emplacement de fichier des prédictions)
 - f (Nom du fichier pour sauvegarder les indicateurs calculé),
 - i (la liste des indicateurs à calculer),
- Fonctionnement** : Fonction `calculate_indicators`
Fonction `calculate_indicator`

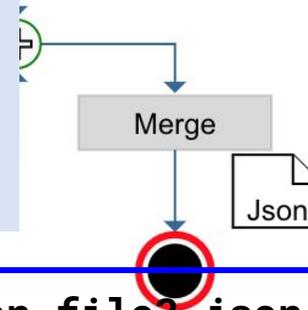


```
python evaluate.py --r TrueLabel.csv --p NP_prediction.csv --f indicators --i rmse mae mse fcp
```

Vous êtes un Scénariste

- Regrouper les Résultats

- Un script python: **mergeResults.py**
- **Input : Les json Files**
- **Output : File json**
- **Les Paramètres :**
 - **--f** (Nom du fichier pour sauvegarder les résultats)
 - **--r** (la liste des json files à regrouper)
- **Fonctionnement :** Fonction **merge_final_results**



```
python evaluate.py --f allIndicators.json --r file1.json file2.json file3.json
```

Vous êtes un Scénariste

- **LOLA: Choix Techniques**

- **Utilisation de Docker pour isoler les algorithmes**

- Technologie de conteneurisation
 - Permet de gérer les dépendances des algorithmes sans «contaminer» les serveurs



- **Nextflow pour l'exécution des scénarios**

- Système de workflow
 - Permet de gérer les ressources de la machine et les files d'attente en cas de calculs longs



Vous êtes un Scénariste

- Les bonnes pratiques de programmation

- **Limitations technique :**

- Pas d'interfaces graphique → uniquement en ligne de commandes
 - Pas d'accès à internet → Si besoin de ressources, elles doivent être dans le conteneur
 - Ressources matérielles sur le cluster de calcul → 96 CPU + 256 Go RAM → Pas de GPU
 - Hébergement des scénarios sur un répertoire Git

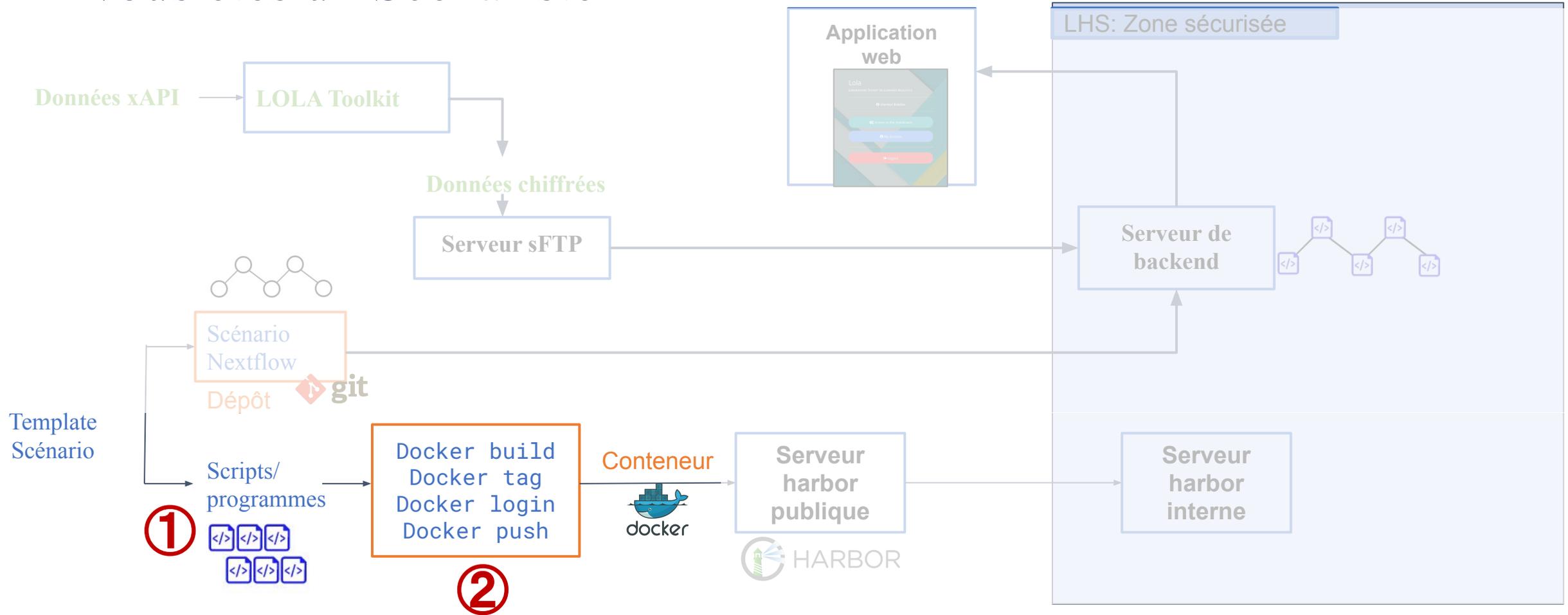
- **Nommer ses paramètres :**

- Mauvaise pratique `$ my_script input.json output.json 5.2`
 - Nommer les paramètres `$ my_script --input input.json \
--output output.json \
--distance 5.2`

Vous êtes un Scénariste

- **Travailler avec Docker**
 - **Docker = solution de conteneurisation**
 - Embarquement des programmes
 - Limitation des ressources utilisées par le conteneur (CPU, mémoire)
 - Limitation des accès aux données des autres utilisateurs
 - **Hébergement des images sur une registry Harbor**
 - Instance publique avec même accès que sur l'application web
 - Répliquée en interne sur une instance Harbor miroir → Le serveur de calcul n'est pas accessible à l'extérieur
 - **Atelier**
 - Faire une image Docker à partir de ses scripts
 - Envoyer son image sur le harbor

Vous êtes un Scénariste



Vous êtes un Scénariste

- Faire une image Docker

- Documentation : [Build Docker Images](#)
- Ouvrir le dossier atelier-lola/**scenario-atelier**
 - Scripts Python dans **le dossier /bin**
- Créer un fichier Dockerfile contenant la recette de construction de l'image Docker (fichier **Dockerfile**)

```
FROM continuumio/miniconda3 ①
COPY bin/ /recommandation-bin/bin ②
COPY requirements.txt /recommandation-bin/bin

WORKDIR /recommandation-bin/bin

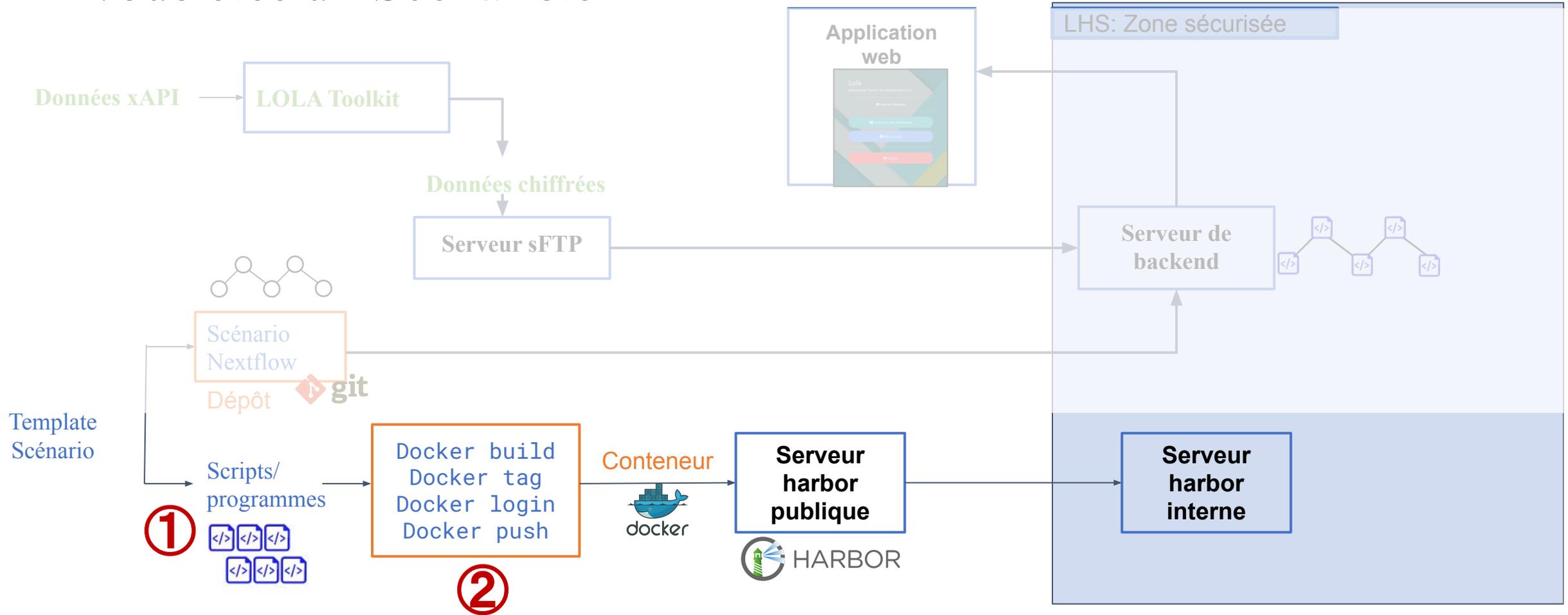
RUN apt update \ && ③
    apt install procps build-essential -y && \ ④
    pip install -r requirements.txt && \ ⑤
    chmod +x /recommandation-bin/bin/*

# Fix to avoid github.com/NicolasHug/Surprise/issues/36
ENV SURPRISE_DATA_FOLDER /tmp ⑥
ENV PATH="/recommandation-bin/bin:${PATH}"
```

- 1) Docker image de base pour avoir Python 3.8
- 2) Copie des scripts python et du fichier contenant les dépendances
- 3) Installation des dépendances pour nextflow (procps, build-essential)
- 4) Installation des dépendances Python
- 5) Rendre exécutable les scripts Python
- 6) Ajout des scripts dans le PATH

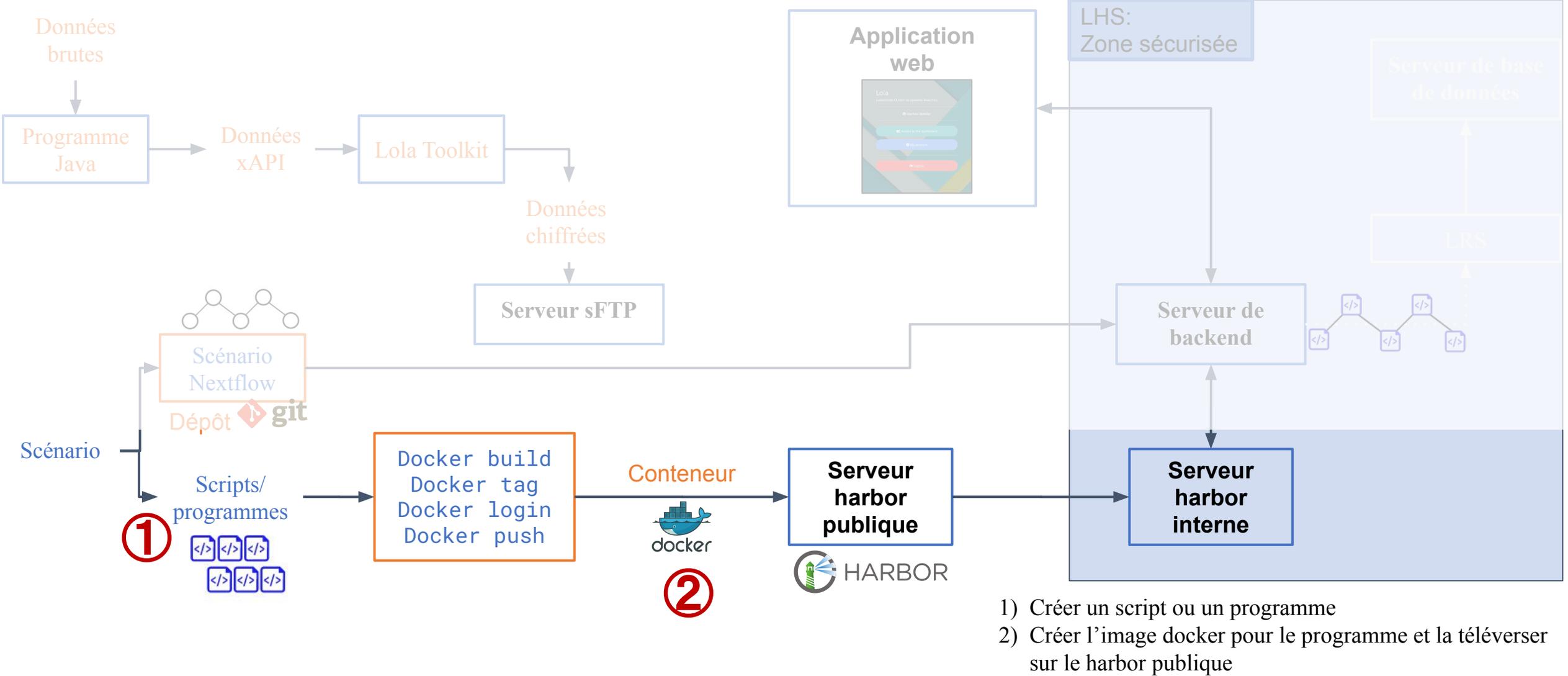
- Constuction de l'image: **\$ docker build -t recommandation .**
- Test de l'image : **\$ docker run recommandation model_test.py**

Vous êtes un Scénariste

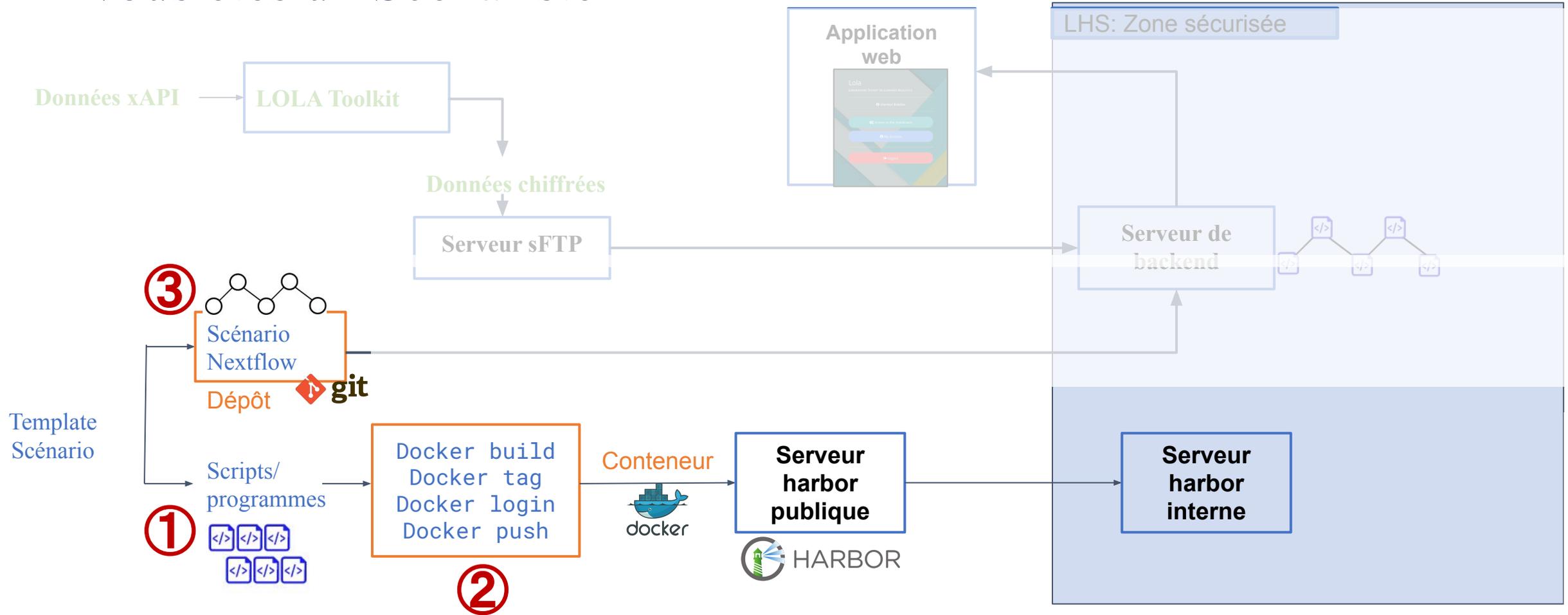


Vous êtes un Scénariste

- Upload de l'image Docker sur le harbor du LHS
 - Se connecter à l'interface web <https://lola.lhs.loria.fr:4443>
 - Cliquer sur **New project** pour créer un dossier/projet:
 - Les noms de dossier sont communs à tous les utilisateurs. Il doit être unique
 - Dans un terminal, donner un nom à son image docker
 - **\$ docker tag recommandation lola.lhs.loria.fr:4443/NOM_PROJET/recommandation:latest**
 - Se connecter avec ses identifiants
 - **\$ docker login lola.lhs.loria.fr:4443**
 - Téléverser l'image Docker
 - **\$ docker push lola.lhs.loria.fr:4443/NOM_PROJET/recommandation:latest**
 - Vérifier dans l'interface web



Vous êtes un Scénariste



Vous êtes un Scénariste

- Faire un scénario avec Nextflow

- Documentation : [How to create a scenario for Lola](#)
- Créer un dossier avec la structure minimale suivante:

①  modules

②  README.md

③  main.nf

④  nextflow.config

⑤  params.json

1) Dossier contenant les objets nextflow (fonctions, processus, etc.)

2) Fichier Markdown donnant des informations sur le scénario (objectifs, références bibliographiques, dépendances, etc.)

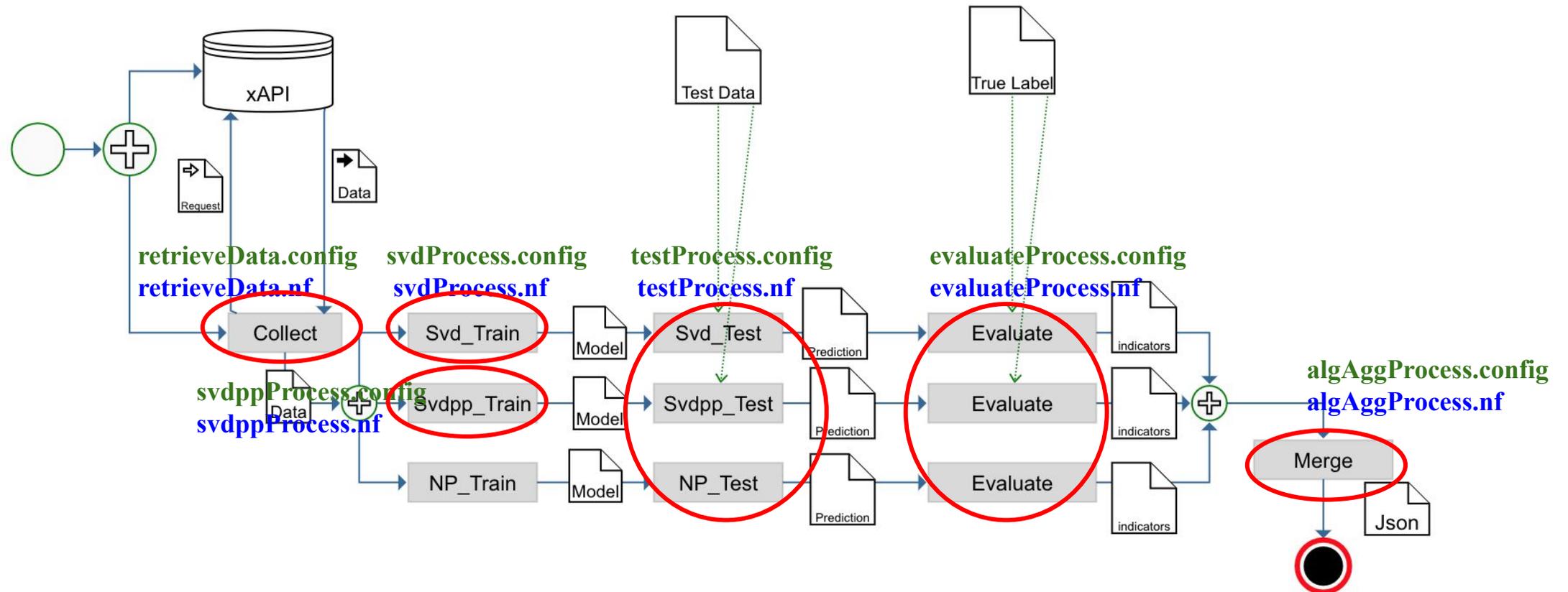
3) Script principale de Nextflow. C'est ce script qui est lancé par nextflow

4) Contient les paramètres nextflow du scénario

5) Fichier json de configuration du scénario. Permet au serveur d'identifier comment configurer le scénario, avoir la liste des paramètres, etc.

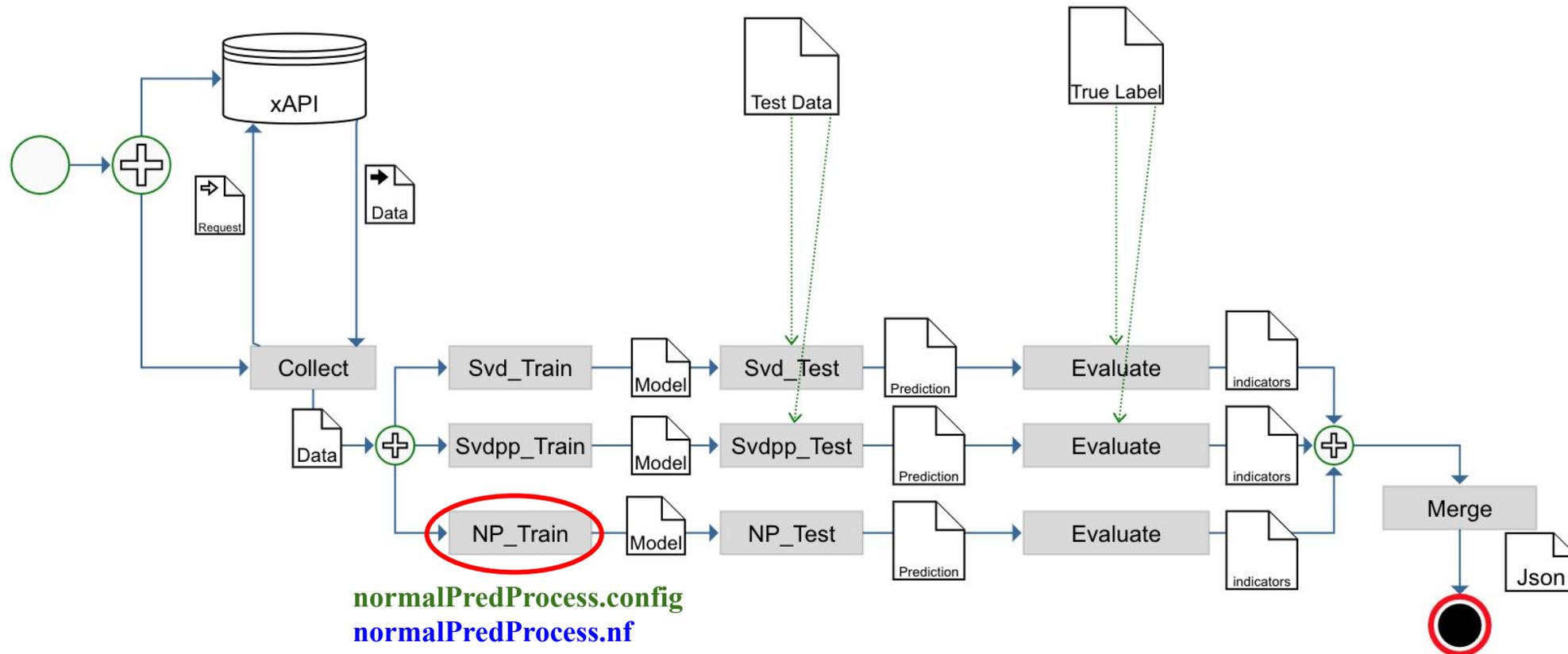
Vous êtes un Scénariste

Le Template de scénario



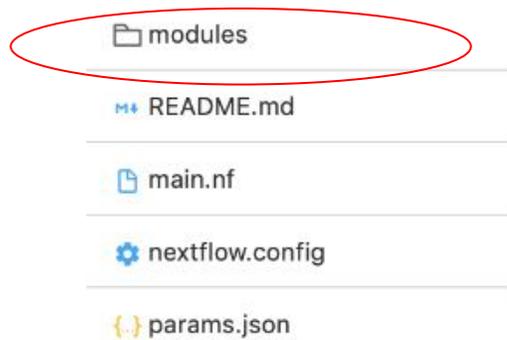
Vous êtes un Scénariste

Le Template de scénario



Vous êtes un Scénariste

. Normal Predictor: Process Pour le Train



NormalPredProcess.config

```
params{
normalP_trained_model= 'normalP_Trained'
normalpredictorflag=true
}
```

NormalPredProcess.nf

```
#!/usr/bin/env nextflow
/*
 * Process to train Normal Predictor on data collected from the LRS
 */
process normalP_train{
    cpus 8
    memory '6G'
    container = "$params.main_container"

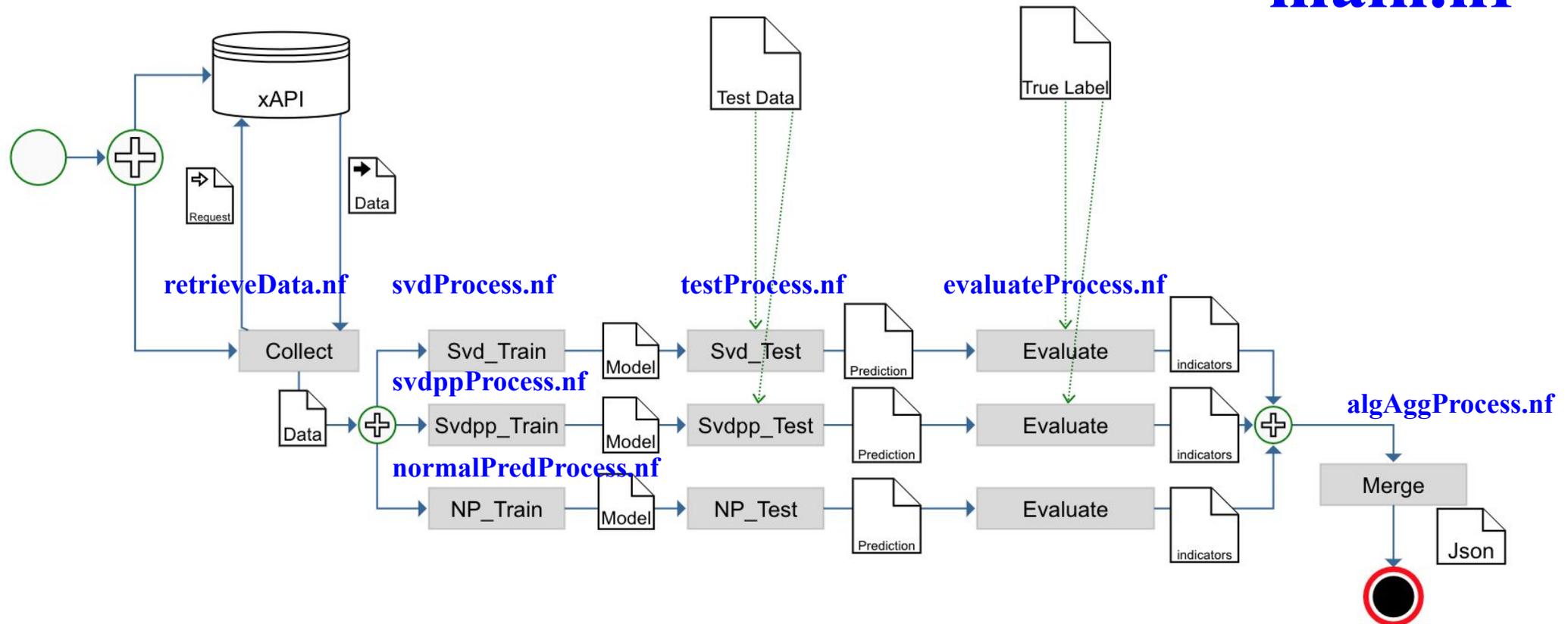
    input:
    path traindata
    val filename
    output:
    path "$filename*" , emit: normalPtrainedmodel

    """
    normalPredictor_train.py --a $traindata --m $filename
    """
}
```

Vous êtes un Scénariste

Le Template de scénario

main.nf



Vous êtes un Scénariste

- Faire un scénario avec Nextflow
 - Test du scénario en local (nécessite d'être sur le réseau du Loria)
 - Être dans le dossier atelier-lola/scenario-atelier

```
nextflow -c nextflow_docker.config run main.nf --lrsHost=http://garimpeiro12.loria.fr --lrsPort=81
```

Vous êtes un Scénariste

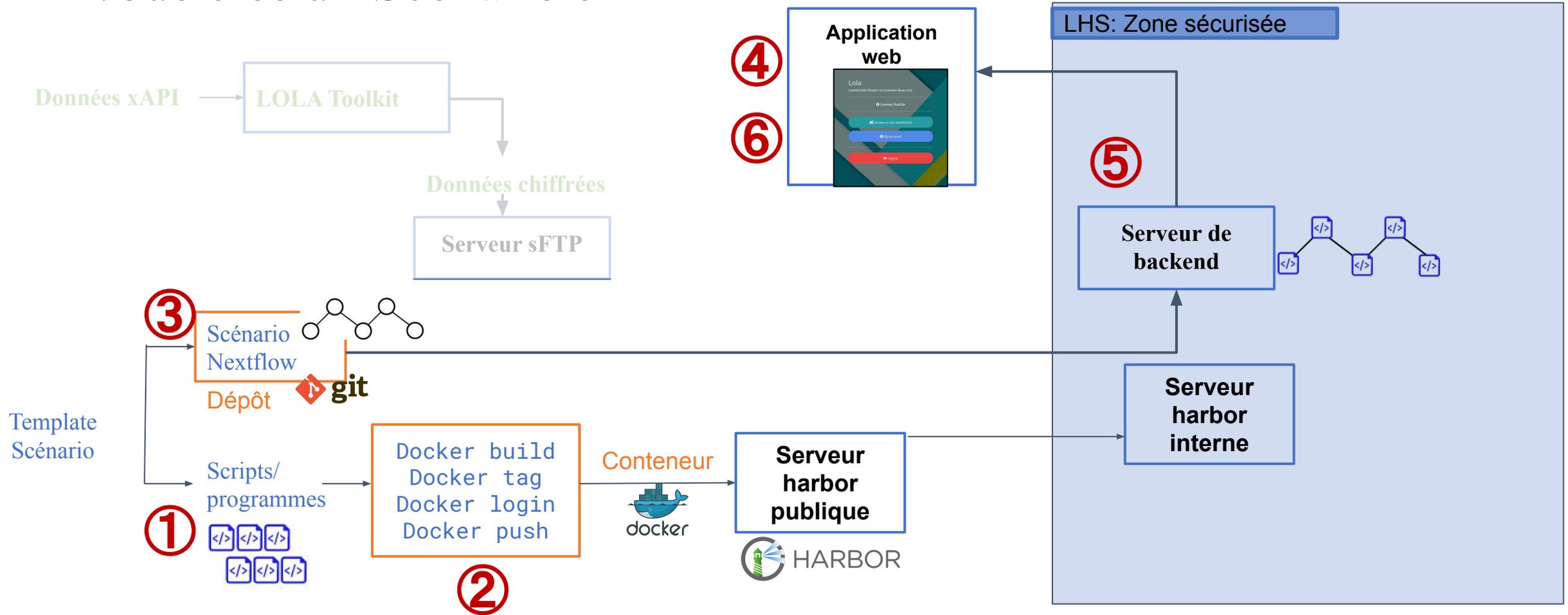
- Faire un scénario avec Nextflow

- [Documentation params.json](#)
- Utiliser le fichier params.json.sample pour créer le votre

- 1) Metadata du scénario
- 2) Liste des fichiers générés par le scénario
- 3) Liste des images docker utilisées par le scénario
- 4) Liste des paramètres nextflow du scénario

```
{  
  1 "scenario_name": "Recommandation",  
  2 "description": "This scenario compute recommandation",  
  3 "output": "all_results.json",  
  "docker_images": [  
    {  
      "full_name": "recommandation:1.0.0",  
      "harbor_url": "http://lola.lhs.loria.fr:  
4443/lola/recommandation:1.0.0"  
    }  
  ],  
  4 "parameters": [  
    {  
      "name": "limit",  
      "description": "limite number of element to compute ",  
      "type": "int",  
      "default": "5"  
    }  
  ],  
  ...  
}
```

Vous êtes un Scénariste



Vous êtes un Scénariste

- **Installation d'un scénario sur la plateforme:**

- Avoir son scénario nextflow sur un dépôt Git et ajouter un numéro de version (git tag)
- Dans l'application web Lola, ajouter un nouveau scénario dans l'onglet **Template scénario** en renseignant:
 - Utiliser l'url
`https://gitlab+deploy-token-420:YP8oviXNzYiT7DjbaZ_p@gitlab.inria.fr/lola/scenarios/scenario-atelier`
 - son nom
- Ajouter un git tag (numéro de version) pour le scénario : 1 . 1 . 6
- Suivre l'évolution dans l'interface

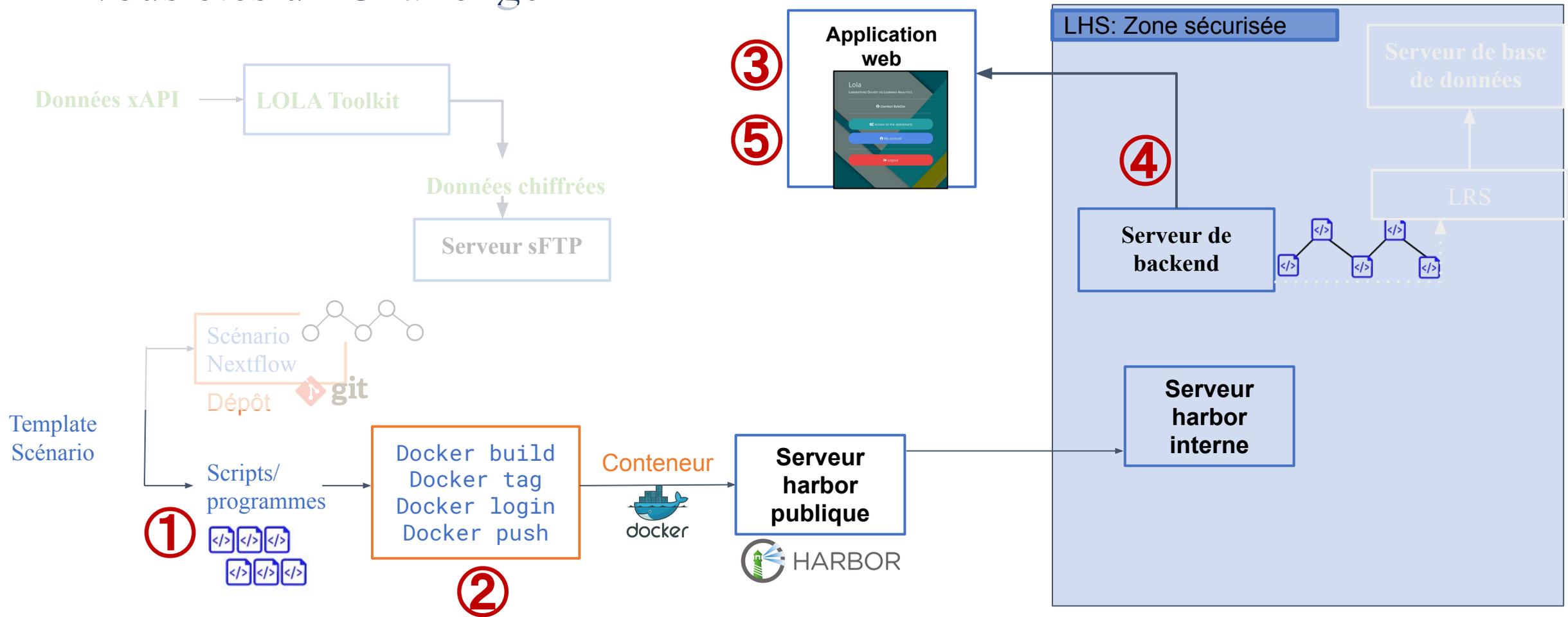
- **Utilisation d'un scénario**

- Dans l'onglet **Template scénario**, cliquer sur **Exécuter**
- Renseigner les paramètres du scénario
- Suivre son exécution dans l'interface

Vous êtes un Challenger

- Comparer son algorithme à ceux du scénario
 - Work in progress solution
 - Requis pour rendre compatible son algorithme à la plateforme
 - Avoir son algorithme dans un conteneur Docker
 - Algorithme uniquement en CLI (Command Line Interface), pas d'interface graphique et pas de connexion à internet
 - Capable de se connecter à un LRS pour récupérer les données
 - Sorties compatibles avec l'algorithme Evaluate
 - Disposer d'un algo_recipe.json (similaire au params.json du scénario) permettant de décrire l'algorithme ainsi que les paramètres d'entrées/sorties/connections à Trax
 - Exemple: https://gitlab.inria.fr/lola/algorithms/example_algorithme

Vous êtes un Challenger



Merci Pour votre Attention