



# Architecture de la plateforme Lola

Architecture et fonctionnement

**Auteur :** Philippe Noel

**Nom de l'organisation :** UL/Loria/CNRS

**Date de création :** 15-02-2022

**Date de modification :** 15-02-2022

**Mots-clés :** Architecture, serveurs, Docker



## Table des matières

<b>1</b>	<b>Liste des abbréviations</b>	<b>3</b>
<b>2</b>	<b>Lexique</b>	<b>3</b>
<b>3</b>	<b>Architecture globale</b>	<b>3</b>
<b>4</b>	<b>Dépôt des données au format Experience Application Programming Interface (xAPI) sur la plateforme</b>	<b>4</b>
4.1	Format des données à déposer sur la plateforme . . . . .	4
4.2	Serveur de dépôt de données . . . . .	5
4.3	Validation et intégration des données xAPI par le serveur « Backend » . . . . .	5
<b>5</b>	<b>Application web sur le serveur « Frontend »</b>	<b>5</b>
<b>6</b>	<b>Serveur de « Backend »</b>	<b>5</b>
6.1	Lien avec le « Frontend » . . . . .	7
6.2	Lien avec le serveur de calcul . . . . .	7
<b>7</b>	<b>Serveur de calcul</b>	<b>7</b>
<b>8</b>	<b>Serveur Harbor</b>	<b>8</b>

## 1 Liste des abbréviations

GUI	Graphical User Interface.
HTTP	Hyper Text Transfer Protocol.
LHS	Laboratoire de Haute Sécurité.
LRS	Learning Record Store.
SFTP	Secure File Transfer Protocol.
SQL	Structured Query Language.
xAPI	Experience Application Programming Interface.

## 2 Lexique

### Algorithme

Un algorithme est un programme utilisé pour faire une seule tâche. C'est une des briques d'un scénario de la plateforme Lola. Voir [ici](#).

Faire lien  
vers la doc  
n°2: Notion  
de scénario

### Backend

Serveur dont la charge est d'orchestrer les différents services de la plateforme. À l'inverse du « Frontend », il n'est pas accessible à l'extérieur.

### DMZ

Demilitarized Zone (Zone demilitarisée en français) est une zone tampon entre internet et un réseau interne. Ici, la zone DMZ permet de faire l'interface entre les utilisateurs et le réseau interne de la plateforme Lola par le biais de l'application web Lola.

### Docker

Solution open-source de conteneurisation. Permet d'isoler un programme du reste du système afin de limiter l'utilisation de ses ressources. Cette solution simplifie également l'installation des programmes car les conteneurs contiennent toutes les dépendances nécessaires à son utilisation. Le programme et ses dépendances forment ainsi une image Docker.

### Frontend

Serveur qui héberge l'application web utilisable par les utilisateurs.

### Lolapy

Programme installé sur le serveur « Backend » et qui orchestre les différents services.

### Slurm

Solution open source d'ordonnancement de tâches informatiques. Slurm permet de gérer les files d'attente de programmes, d'allouer les ressources (processeurs et mémoire) spécifique à l'exécution d'un programme.

### xAPI

Format de données utilisé pour les données d'E-Education. Voir [ici](#).

Faire lien  
vers la doc  
n°3: xAPI

## 3 Architecture globale

L'architecture de Lola est hébergée au LHS et comprend plusieurs serveurs :

- Un serveur SFTP qui permet de déposer des jeux de données au format xAPI;
- Un serveur « Frontend » qui héberge l'application web;
- Un serveur Harbor public, qui permet de stocker les images Docker utilisées pour lancer les scripts des utilisateurs;

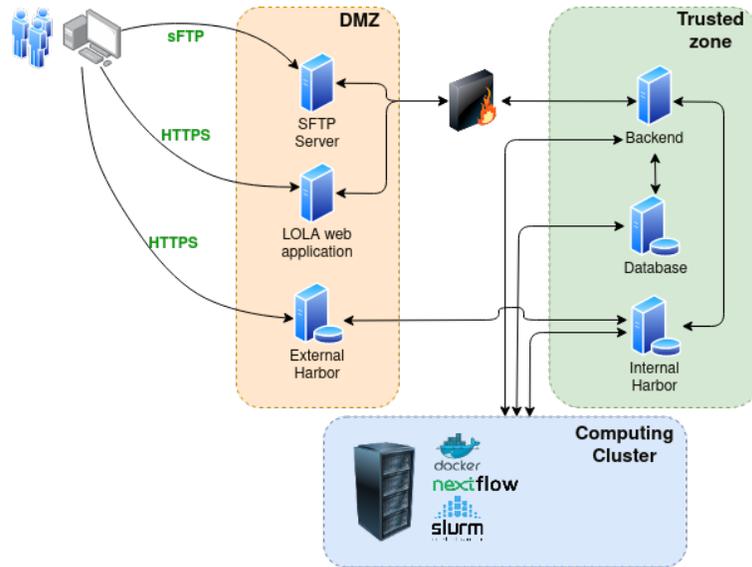


FIGURE 1 – Schéma de l'architecture Lola. En orange, la zone démilitarisée (DMZ), qui contient l'application web de Lola (LOLA web application), le serveur serveur SFTP (SFTP server) et le harbor public (External Harbor). En vert, la zone de confiance (Trusted Zone), qui contient le serveur de « Backend », le serveur de base de données (Database) et le serveur harbor interne (Internal Harbor). Enfin en bleu, le serveur de calcul.

- Un serveur Harbor interne, réplique exacte du serveur Harbor public. Il est accessible uniquement aux machines internes ;
- Un serveur de « Backend » qui gère les jeux de données, l'installation des algorithmes/scénarios, les exécutions de scénario, etc ;
- Un serveur de calcul qui permet de d'exécuter les scénarios sur plusieurs processeurs ;
- Un serveur de base de données qui permet de stocker les jeux de données au format xAPI.

Les serveurs SFTP, Harbor et le « Frontend » sont les seules accessibles à l'extérieur. Les serveurs de « Backend », de base de données et de calcul sont quant à eux protégés sur le réseau sécurisé du Laboratoire de Haute Sécurité (LHS). Le serveur de « Backend » est ainsi le point central de l'architecture, il permet de piloter les différents acteurs et de faire la liaison entre l'application web et les différents services de la plateforme (base de données, calcul, etc.). Le serveur de « Backend » ne contient aucune donnée, c'est pourquoi il est possible de le lier au serveur de « Frontend » externe. Les serveurs de base de données et de calcul contiennent des données critiques et ne doivent donc pas avoir de lien vers la DMZ.

## 4 Dépôt des données au format xAPI sur la plateforme

### 4.1 Format des données à déposer sur la plateforme

Les données xAPI déposées sur la plateforme Lola sont soumises à un format particulier. Plusieurs étapes sont à effectuer avant de déposer les données sur la plateforme :

1. Génération d'un fichier bordereau doit être généré dans l'interface web. Ce bordereau permet l'identification du jeu de données et de son propriétaire via un numéro unique.



2. Compression du fichier bordereau et du fichier au format xAPI afin de limiter l'espace disque utilisé par le jeu de données et la bande passante lors du transfert entre l'utilisateur et la plateforme
3. Chiffrement de l'archive à l'aide d'une clef de chiffrement publique, permettant de garantir la sécurité des données lors du transfert de l'utilisateur vers le serveur de dépôt de données.

## 4.2 Serveur de dépôt de données

Le serveur de dépôt de données est un serveur SFTP. Les utilisateurs peuvent y déposer leurs jeux de données une fois celui-ci compressé et chiffré selon le format présenté 4.1. Pour simplifier ces différentes étapes, ainsi que le transfert des données, un outil appelé « Lola Toolbox » disponible à l'adresse suivante : [https://gitlab.inria.fr/lola/lola\\_toolbox/-/releases](https://gitlab.inria.fr/lola/lola_toolbox/-/releases).

Cet outil Graphical User Interface (GUI) permet d'effectuer les étapes présentées précédemment ainsi que le transfert vers la plateforme Lola. Les données sont stockées sur le serveur de dépôt dans « une zone tampon » avant d'être transférées dans un espace de stockage accessible au serveur « Backend ».

## 4.3 Validation et intégration des données xAPI par le serveur « Backend »

Une fois accessible au serveur « Backend », les données sont traitées selon les étapes suivantes :

1. Déchiffrement à l'aide d'une clef privée, accessible uniquement sur le serveur « Backend » ;
2. Décompression pour accéder au bordereau et au fichier xAPI
3. Validation des deux fichiers. Le « Backend » vérifie que les deux fichiers sont au format correct.
4. Intégration des données.

L'intégration des données se fait au travers d'un Learning Record Store (LRS). Le choix s'est porté sur Trax (<https://traxlrs.com/>). Intégrer des données dans Trax se fait à travers des requêtes Hyper Text Transfer Protocol (HTTP). Trax stock ensuite ses données dans une base de données Structured Query Language (SQL).

Faire lien vers la doc n°3 : Données xAPI

## 5 Application web sur le serveur « Frontend »

L'application web est l'interface que l'utilisateur utilise pour la majorité des opérations de la plateforme Lola. Elle est disponible à l'adresse <https://lola.lhs.loria.fr>.

L'onglet Dataset de l'application web Lola est visible sur la figure 2.

## 6 Serveur de « Backend »

Le serveur de « Backend » est l'orchestrateur de la plateforme Lola. C'est lui qui gère la majorité des connexions et des services. Les principaux services que le « Backend » gère sont :

- Lien avec le « Frontend » pour les actions utilisateurs. Voir 6.1 ;
- Lien avec le serveur de calcul ;
- Lien avec le serveur de base de données. Pour vérifier les accès, créer ou supprimer des jeux de données xAPI.

La gestion de ces services est réalisée par un programme appelé Lolapy qui est une API interne, c'est-à-dire que d'autres services peuvent appeler Lolapy pour effectuer des actions, par exemple le « Frontend ».

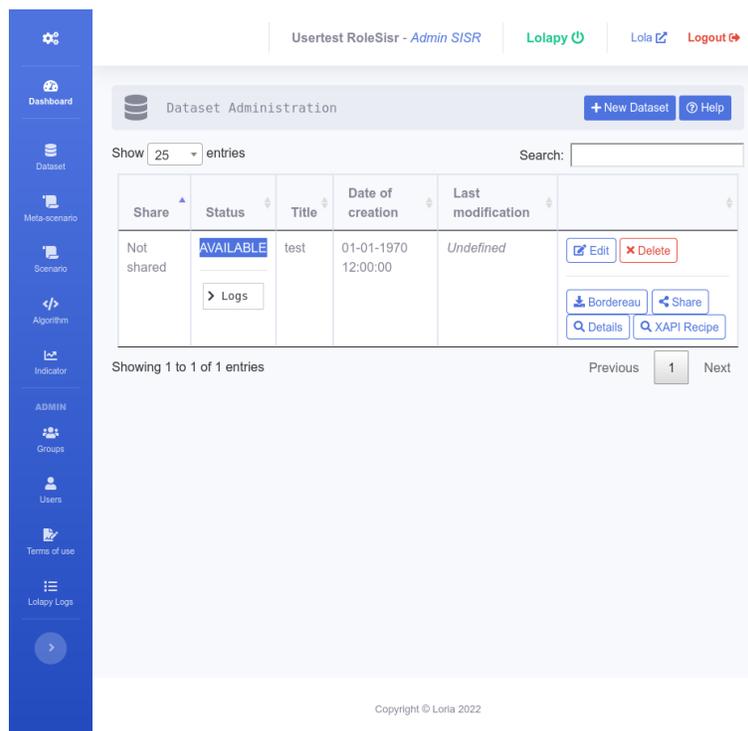


FIGURE 2 – Interface de l'application web Lola



## 6.1 Lien avec le « Frontend »

Le lien entre Lolapy et le « Frontend » est réalisé au travers d'une connexion sécurisée. Le « Frontend » communique avec lolapy principalement pour les actions des utilisateurs. Par exemple un utilisateur veut avoir des informations (nombre de statements, nombre de verbes uniques, etc.) sur un jeu de données spécifique, le « Frontend » envoie alors une requête au serveur « Backend » pour obtenir ces informations. Le « Backend » répond ensuite avec ces informations et le « Frontend » peut alors les afficher à l'utilisateur.

## 6.2 Lien avec le serveur de calcul

Lorsqu'un utilisateur veut exécuter un scénario via l'application web, plusieurs étapes sont effectuées :

1. Le « Frontend » envoie à Lolapy les informations saisies par l'utilisateur (jeu de données, nom du scénario à exécuter, paramètres des algorithmes, etc.)
2. Lolapy prépare l'environnement de travail, les dossiers qui vont contenir les données, etc.
3. Lolapy envoie une demande de réservation pour une instance Trax sur le serveur de calcul via Slurm. Les données nécessaires au scénario seront accessibles sur cette instance spécifiquement.
4. Une fois la réservation pour trax disponible, Lolapy exécute le scénario avec Nextflow. Nextflow est le gestionnaire de workflow utilisé pour gérer les scénarios (voir [pour plus d'information sur Nextflow.](#))
5. Nextflow lance les différents algorithmes du scénario à travers Slurm (Voir 7)
6. Lolapy analyse les fichiers de sorties de Nextflow et Slurm tout au long de l'exécution. Il envoie une partie des informations au « Frontend » à afficher dans l'application web. Cela permet à l'utilisateur de suivre l'état d'avancement du scénario. Il surveille aussi le bon déroulement du scénario afin de gérer les erreurs possibles.
7. Une fois le scénario terminé, Lolapy stoppe le serveur Trax, nettoie les fichiers temporaires et envoie un signal au « Frontend » indiquant la fin du scénario. L'utilisateur peut alors télécharger ses fichiers de résultats à travers l'application web.

Faire un lien vers la doc n°2 : Notion de scénario

## 7 Serveur de calcul

Le serveur de calcul est le serveur sur lequel est effectué tous les calculs des utilisateurs de la plateforme. Plus particulièrement, toutes les opérations nécessitant un temps de traitement long sont effectuées sur le serveur de calcul. Celui-ci dispose d'un grand nombre de processeurs et de mémoires permettant de réaliser les calculs en parallèle.

Les calculs sont transmis du serveur « Backend » au serveur de calcul via l'intermédiaire de Slurm. Slurm est une solution open source d'ordonnancement de tâches informatiques. Ce logiciel permet de gérer les réservations ou les files d'attente de programmes, d'allouer les ressources (processeurs et mémoire) spécifique à l'exécution d'un programme, et fournit un environnement pour le suivi des programmes en cours d'utilisation. De cette façon, si plusieurs utilisateurs lancent des scénarios en même temps, slurm ordonnera au mieux leur exécution et gèrera les files d'attente afin d'éviter une surutilisation du serveur.

## 8 Serveur Harbor

Harbor est une solution open source de stockage d'images Docker. Les algorithmes utilisés sur la plateforme sont utilisés sous la forme d'images Docker. Cela permet de simplifier leurs installations et évite de polluer le serveur avec les dépendances des utilisateurs. Le serveur Harbor est accessible à l'extérieur. Tout utilisateur qui possède un compte sur l'application web Lola peut déposer des images Docker sur le serveur Harbor.

Le serveur Harbor public est répliqué en interne afin de faciliter son accès au serveur de calcul. En effet, lors de l'installation d'un scénario, le « Backend » récupère la liste des images Docker nécessaires à son bon fonctionnement. Il envoie alors un signal au serveur de calcul (où seront exécutés les algorithmes) pour télécharger les images Docker sur le serveur Harbor interne.

Changer le terme polluer pour un moins agressif