

Architecture interfaces import/export

Projet KeopsBridge

Version 1.0 – 24/09/2019

Historique des révisions du document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifications | Auteur |
| 1.0 | 24/09/2019 | Création document | MRO |
| 1.1 | 27/09/2019 |  | OAL |
| 1.2 | 02/10/2019 | Mise à jour des schémas | MRO |
| 1.3 | 22/11/2019 | Ajout de la liste des fonctionnalités à réaliser | MRO |

Sommaire

[Introduction 4](#_Toc20915302)

[Objectif de ce document 4](#_Toc20915303)

[Description fonctionnelle 4](#_Toc20915304)

[Architecture technique 4](#_Toc20915305)

[Organisation de KeopsBridge 4](#_Toc20915306)

[Organisation interne 4](#_Toc20915307)

[Organisation des sources 5](#_Toc20915308)

[Fonctionnement 5](#_Toc20915309)

[Fonctionnalité transverse 7](#_Toc20915310)

[Authentification 7](#_Toc20915311)

# Introduction

## Objectif de ce document

Ce document est un document d’architecture de la solution KeopsBridge ; il vise à donner une vision d’ensemble des problématiques structurantes, donner une vision d’ensemble de la solution (d’un point de vue fonctionnel et technique), ainsi que de présenter les choix structurants de principes et de composants, ainsi que les raisons de ces choix.

Il s’adresse aux personnes suivantes :

* Les développeurs XLM Aero,
* Les sous-traitants de XLM Aero.

# Description fonctionnelle

Les interfaces avec des systèmes tiers vont être réorganisés en tant que système autonome dans Keopsbridge, connecté à Keops.

Les fonctions de KeopsBridge dans le système Keops sont :

1. Extraction de données métier en format électronique
2. Importation de données métier en format électronique
3. Déclencher des traitements internes de Keops à partir de systèmes tiers
4. Monitorer l’état interne de Keops depuis l’exterieur

# Architecture technique

Cette section vise à introduire l’environnement dans lequel s’intègrent, elle se concentre principalement sur les contraintes imposées à cet environnement et les choix techniques.

Le composant est packagé dans une WebApp déployable dans un conteneur de servlets Tomcat.

Il présente des points d’entrée de WebService REST et SOAP. En plus de ces points d’entrée WebService, des traitements peuvent être déclenchés par un scheduleur programmé. Les traitements spécifiques pour chacun des types de déclenchement sont indépendants du traitement métier. Ainsi, la même fonction métier peut être exécutée par différents déclencheurs.

# Organisation de KeopsBridge

## Organisation interne

Deux axes d’organisation sont considérés :

Chaque package est classé, soit dans un des axes ou l’autre en fonction de son caractère dominant. C’est-à-dire, les sources dédiées à des traitements métier, sont classés par l’axe fonctionnel. Les traitements d’interface et déclenchement sont classés par l’axe technique.

## Organisation des sources

Ci-après l’organisation des sources dans KeopsBridge :

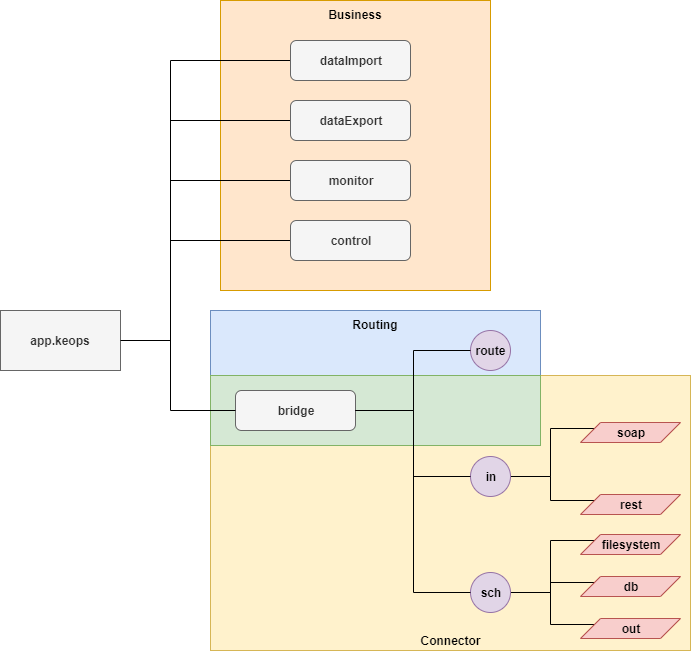


Figure 1 : Organisation interne de KeopsBridge

Exemple d’organisation dans KeopsBridge :

*app.keops.import.flight.ssim.v1.importFlight*

# Fonctionnement

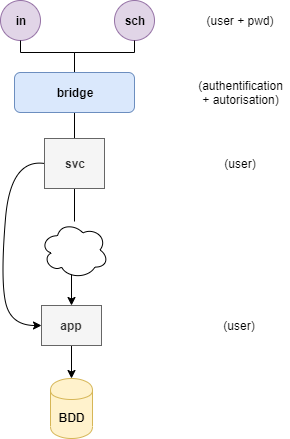


Figure 2 : fonctionnement de KeopsBridge

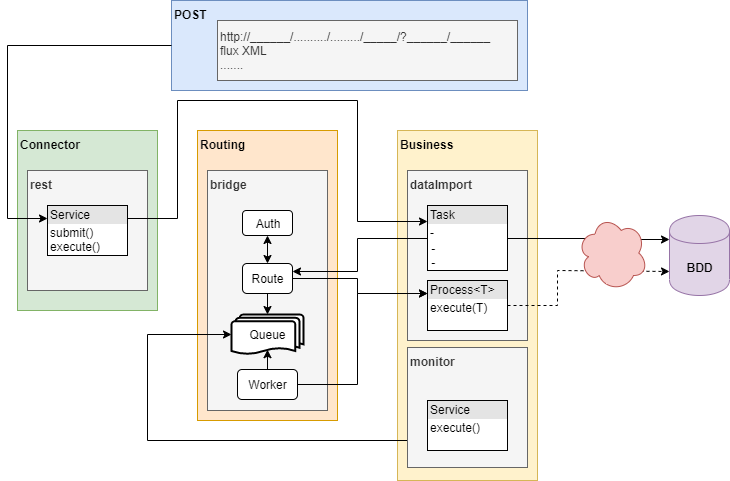


Figure 3: Processus de traitement de KeopsBridge

# Fonctionnalité transverse

## Authentification

Les déclencheurs particuliers obtiennent des crédenciales qui sont traitées dans un composant commun.

Dans tous les cas, les crédenciales consistent en un utilisateur et un mot de passe.

Cet utilisateur et mot-de-passe sont vérifiés dans le contexte de la gestion d’utilisateurs de Keops.

### Eléments à mettre en place pour l’authentification

* Appel à un service (connector)
* Appel du route dans le bridge
* Renvoi si ok ou non
  + Exécution de la queue si autorisation ok