

OPENTESTFACTORY

UNE INITIATIVE OPEN SOURCE POUR SIMPLIFIER
L'EXPLOITATION DES TESTS AUTOMATISÉS

Club Qualité Logicielle 19/10/2021



1. LA PROBLÉMATIQUE DE L'AUTOMATISATION DES TESTS
2. LA RÉPONSE D'OPENTESTFACTORY
3. INTÉGRATION AVEC SQUASH
4. USE CASES
5. STATUT COURANT ET ROADMAP

UNE PROBLÉMATIQUE DE L'AUTOMATISATION DES TESTS

Postulat :

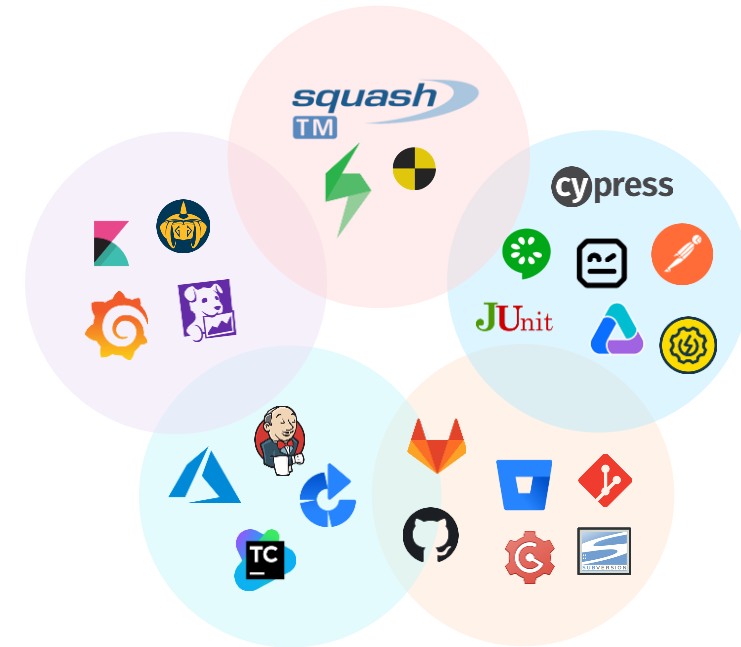
L'automatisation des tests nécessite l'utilisation et l'intégration de divers automates selon le type d'application sous test, la méthodologie et les compétences de l'équipe.

● Build:

- Pour chaque automate, il faut passer du **temps** à mettre en place et **maintenir** le **lancement** et la gestion des **résultats**.
- Ceci requiert des **compétences** que le testeur fonctionnel et l'automaticien n'ont pas nécessairement.
- La **quality gate** doit être mise en place dans le pipeline de CI/CD par un **DevOps**.
- La quality gate **n'est pas** sous le **contrôle** des **testeurs**.

● Run:

- La multiplicité des outils entraîne la nécessité de naviguer plusieurs sources/pages pour avoir
 - une vue d'ensemble de l'état du projet ;
 - une visualisation de son évolution.
- Les données sont en dehors de l'outil du testeur.



UNE PROBLÉMATIQUE DE L'AUTOMATISATION DES TESTS

Postulat :

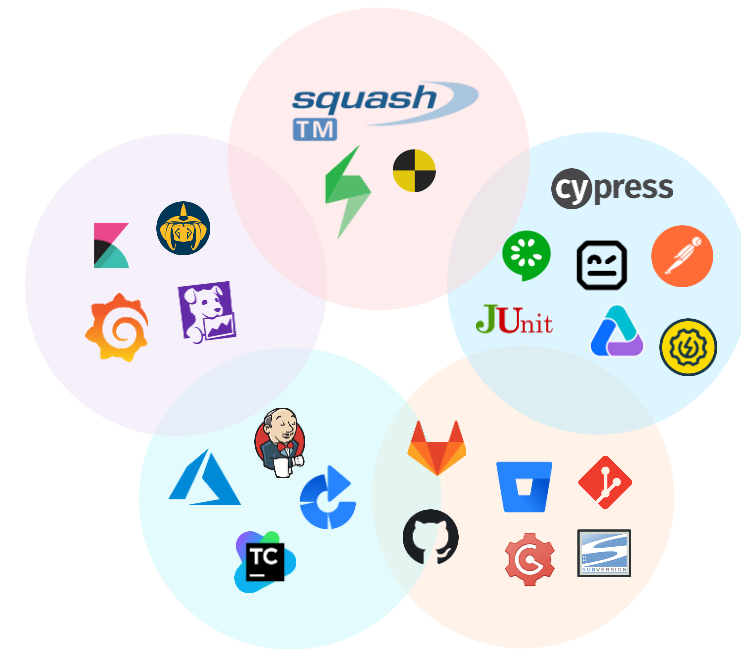
L'automatisation des tests nécessite l'utilisation et l'intégration de divers automates selon le type d'application sous test, la méthodologie et les compétences de l'équipe.

● Build:

- Pour chaque automate, il faut passer du **temps** à mettre en place et **maintenir** le **lancement** et la gestion des **résultats**.
- Ceci requiert des **compétences** que le testeur fonctionnel et l'automaticien n'ont pas nécessairement.
- La **quality gate** doit être mise en place dans le pipeline de CI/CD par un **DevOps**.
- La quality gate **n'est pas** sous le **contrôle** des **testeurs**.

● Run:

- La multiplicité des outils entraîne la nécessité de naviguer plusieurs sources/pages pour avoir
 - une vue d'ensemble de l'état du projet ;
 - une visualisation de son évolution.
- Les données sont en dehors de l'outil du testeur.



LES BESOINS CONCERNANT LA SOLUTION

Besoins fonctionnels

- Une solution pour
 - lancer les tests
 - gérer les résultats
- Une solution intégrant déjà
 - les frameworks / outils de test
 - les forges de CI/CD

les plus populaires et couvrant les principales méthodologies (BDD, KDT...) et types (Web UI, Web Services...) de test
- Une connexion aux référentiels de test afin que
 - les testeurs puissent y gérer le patrimoine de test et les résultats des tests
 - les testeurs aient le contrôle des tests à exécuter
 - les testeurs puissent y définir une quality gate
- Un reporting
 - unifié
 - spécifique à l'utilisateur (manager, PO, ITA...)

Besoins non-fonctionnels

- Une solution scalable
- Une architecture simple et documentée pour
 - susciter les contributions
 - permettre d'intégrer sa propre technologie interne
- Un projet open source pour pérenniser la solution, créer de la confiance et permettre l'enrichissement de la solution

QU'EST CE QU'OPENTESTFACTORY ?

Une **spécification** et une **implémentation de référence** d'un orchestrateur de tests automatiques.

● **Gestion des tests**

- Sélection des tests d'un plan de test automatisé (possiblement défini dans un référentiel de tests)
- Pilotage de l'exécution des tests
- Gestion de leurs résultats

● **Quality gate**

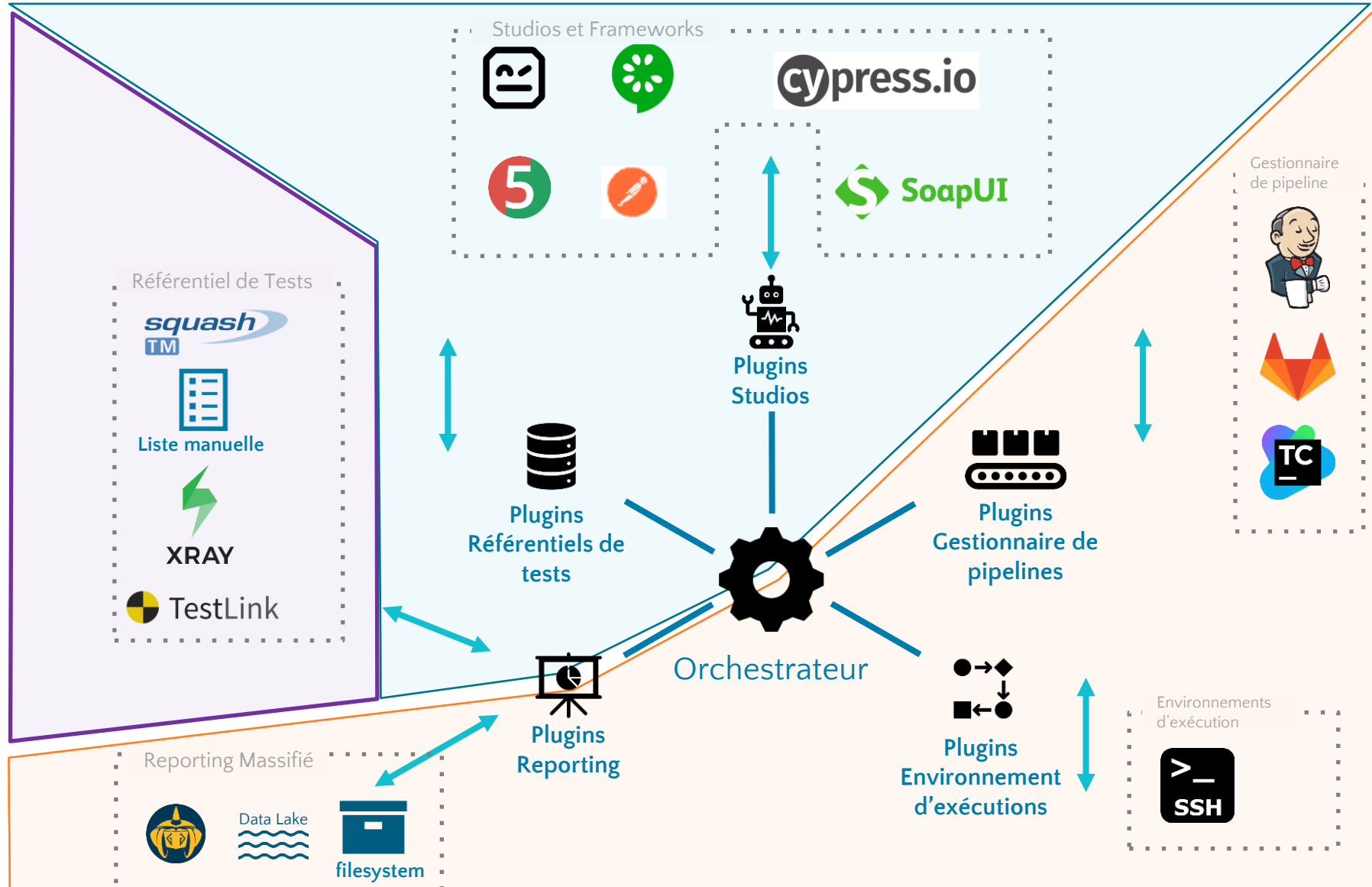
- Définition de critères de succès /échec dans une étape d'un pipeline de CI/CD




● **Reporting**

- Création d'un rapport d'ensemble pour une exécution d'un plan de test
- Consolidation multi-projet de l'historique des exécutions
- Intégration de l'analyse statique de code, la couverture de code, les commits... ?

VUE D'ENSEMBLE

Test Factory



-  Stack test management
-  Stack automatisé des tests
-  Stack DevOps

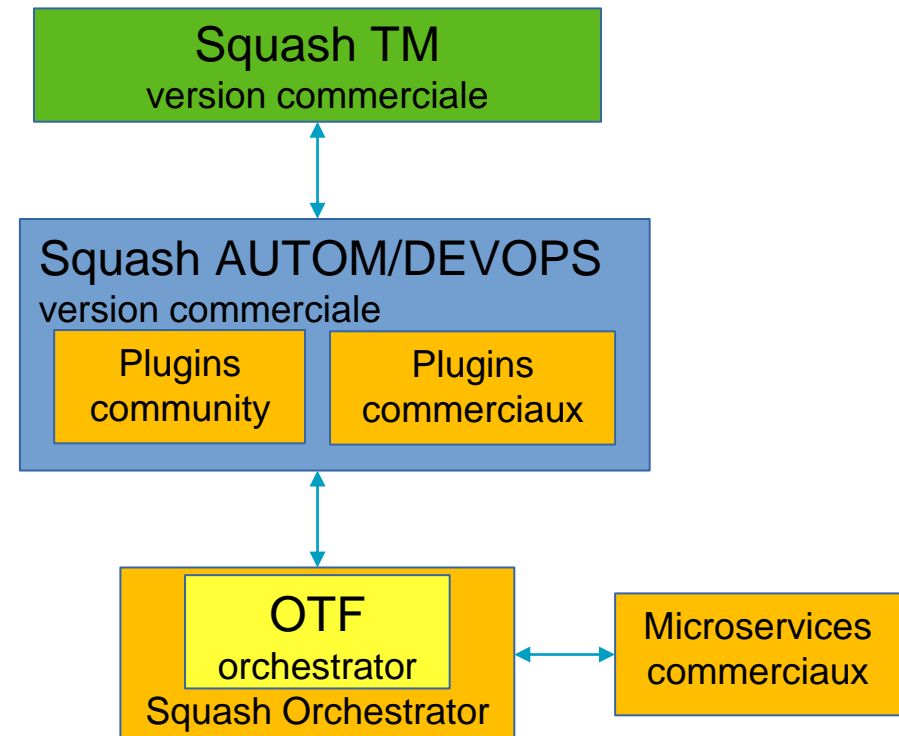
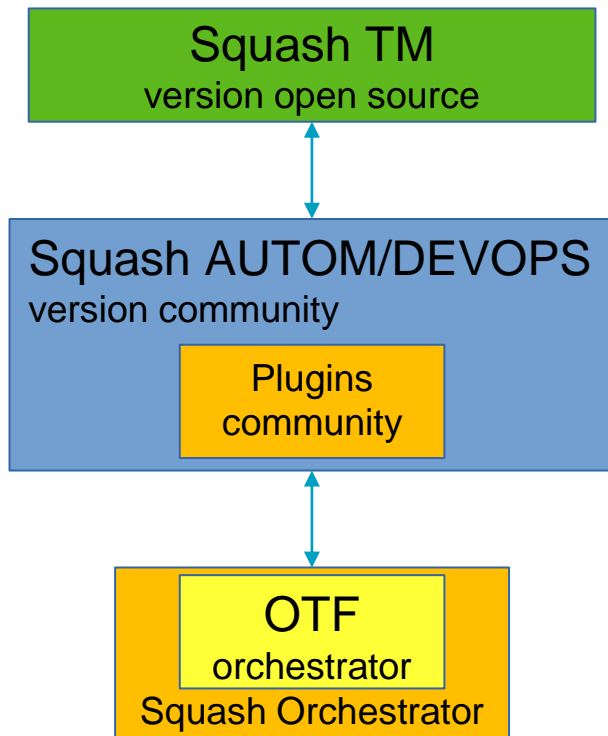
INTÉGRATION AVEC SQUASH

Squash AUTOM

- lancement des tests automatisés depuis Squash TM
- remontée des résultats dans Squash TM

Squash DEVOPS

- récupération dans Squash TM et exécution d'un plan de test depuis une étape du pipeline CI/CD
- remontée des résultats dans Squash TM



USE CASES

UC - 1

☒ Équipe ayant déjà un patrimoine de tests automatisés et voulant bénéficier d'un reporting centralisé

Solution

Intégration d'OTF dans le pipeline de CI/CD

UC - 2

☒ Équipe voulant mettre en place le BDD

Solution autour de Squash

- Écriture des tests en Gherkin et génération des scripts Cucumber dans Squash TM
- Gestion du workflow d'automatisation des tests dans Jira ou Squash TM
- Lancement des tests depuis Squash TM grâce à Squash AUTOM
- Exécution des tests et remontée des résultats dans Squash TM

Solution autour de XRay

- Écriture des tests en Gherkin depuis l'interface Xray
- Liaison des tests Xrays avec les scripts Gherkin de l'automaticien
- Lancement des tests depuis XRay
- Exécution des tests par l'OTF et remontée des résultats dans XRay

USE CASES

UC - 3

Équipe voulant mettre en place le KDT

Solution

Idem solutions UC-2 avec Robot Framework

UC - 4

Équipe voulant définir une quality gate dans le pipeline de CI/CD

Solution

Intégration d'OTF dans le pipeline de CI/CD

UC - 5

Équipe voulant définir une quality gate contrôlée par les testeurs

Solution autour de Squash

- Intégration d'OTF dans le pipeline de CI/CD
- Définition du plan de test dans Squash TM
- Définition des critères de la quality gate dans Squash TM
- Récupération du plan et des critères dans OTF via Squash DEVOPS

USE CASES

UC - 6

Équipe voulant mettre en place l'automatisation avec un outillage (automates et reporting par exemple) spécifique

Solution

- Implémentation des connecteurs spécifiques OTF-Outillage
- Intégration d'OTF dans le pipeline de CI/CD
- (Opt.) Reversement à la communauté des implémentations

UC - 7

Quel est votre besoin ?

Solution

- Venez en discuter à l'atelier OTF cette après-midi

FONCTIONNALITÉS & STATUT COURANTS

- **Gestion des tests**
 - technologies supportées :
 - JUnit
 - Cucumber
 - Cypress
 - Robot Framework
 - SoapUI
 - SKF
 - Postman (2021-11)
 - pilotage de l'environnement d'exécution
 - ssh
 - agent
 - Publication des résultats
 - Amazon S3
 - intégration simplifiée à Jenkins grâce à un plugin
- **Quality gate** → en cours d'implémentation
- **Reporting** → en cours d'analyse

LE PROJET OPEN SOURCE

- **Un site** <https://opentestfactory.org>
 - introduction à OTF
 - spécification
- **Une implémentation de référence** <https://gitlab.com/opentestfactory>
 - Licence Apache 2.0
 - code + tests
 - build + packaging
 - documentation
 - ticketing (user stories, tâches, bugs...)
 - merge requests
- **Un process de contribution**
(<https://opentestfactory.org/contribute/howToContribute.html>)
 - Contributor License Agreements (individu et entité)
 - Soumission d'une proposition
 - Soumission du code / test / doc

ROADMAP DES FONCTIONNALITÉS

Démarrage / arrêt
de l'orchestrateur
dans une étape du
pipeline CI/CD

Quality gate simple
(toujours passante /
arrêt au moindre échec)

Quality gate complète
(définition de critères
selon les types de test)

Oct. 2021

Nov. 2021

Déc. 2021

Q1 2022

Q2 2022

Q3 2022

Q4 2022

Support de
Postman

Première version
du reporting

Amélioration du reporting

ROADMAP OPEN SOURCE

- opentestfactory.org complet
- Code et documentation disponibles dans GitLab

- Build et packaging dans GitLab
 - Ticketing dans GitLab
- OpenTestFactory est un projet OSS à part entière

Oct. 2021

Nov. 2021

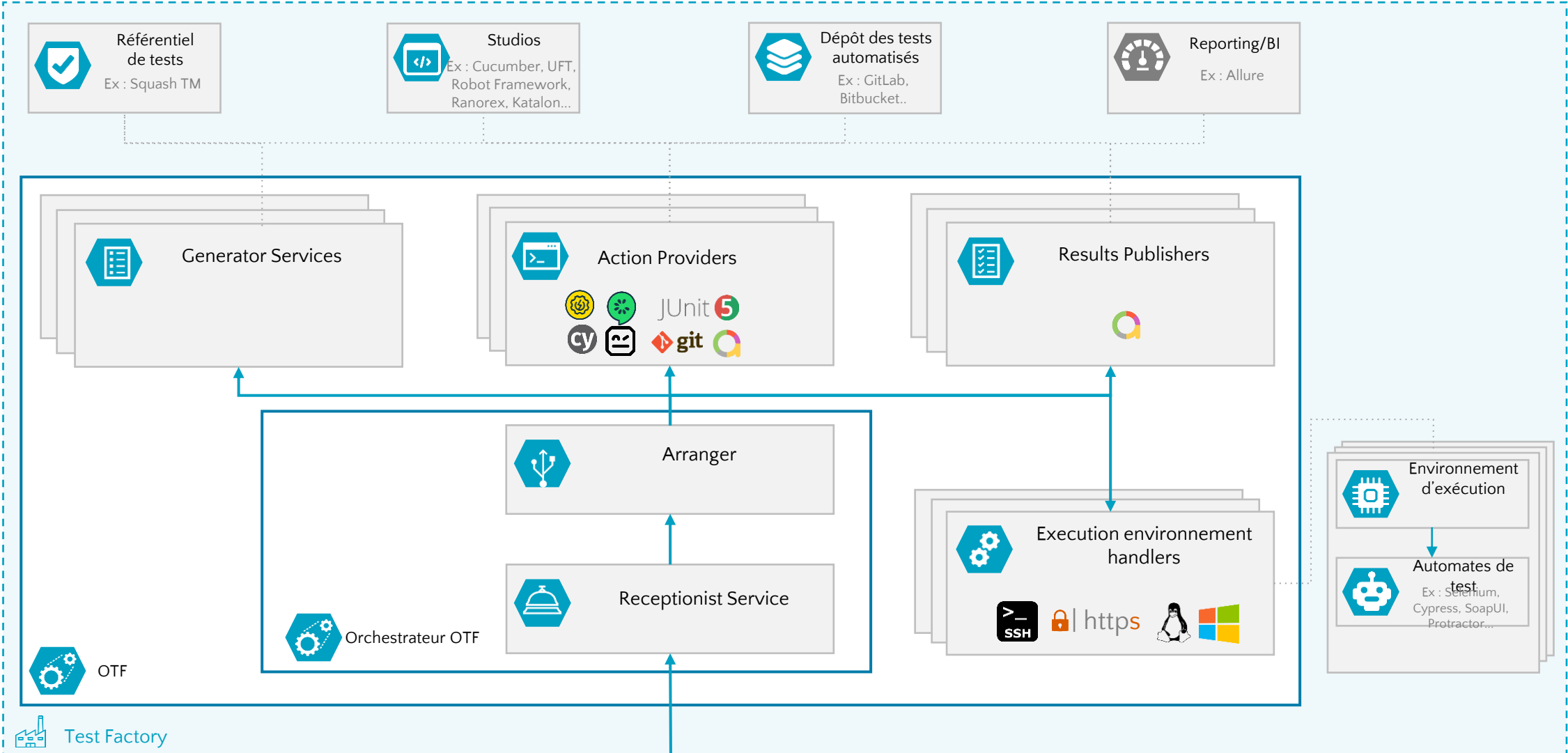
Déc. 2021

- Présentation au Club Qualité

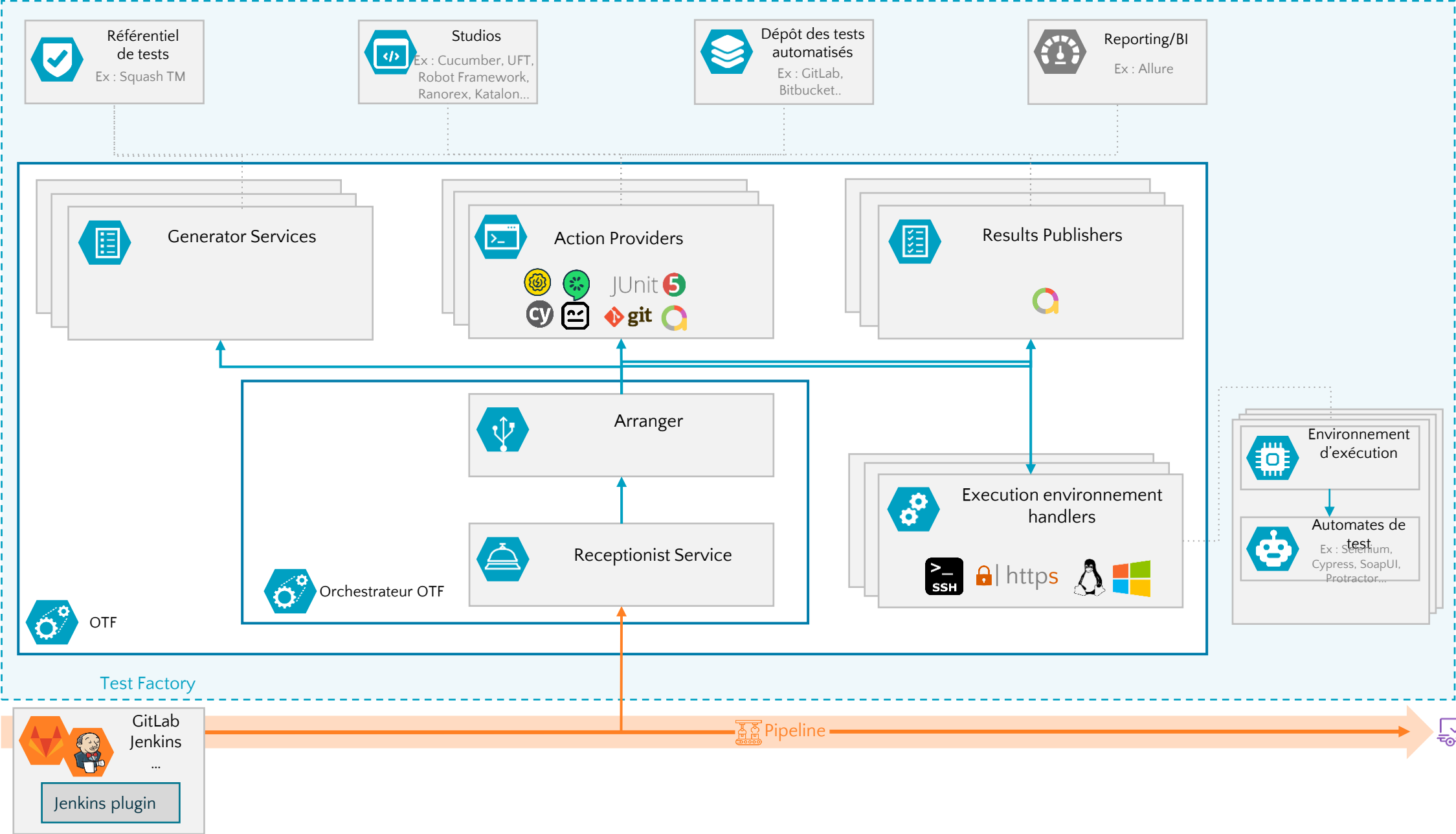
- Présentation à quelques grands comptes et DINUM, appel à contribution
- Inscription au SILL
- Se faire connaître des utilisateurs de GitLab

The end

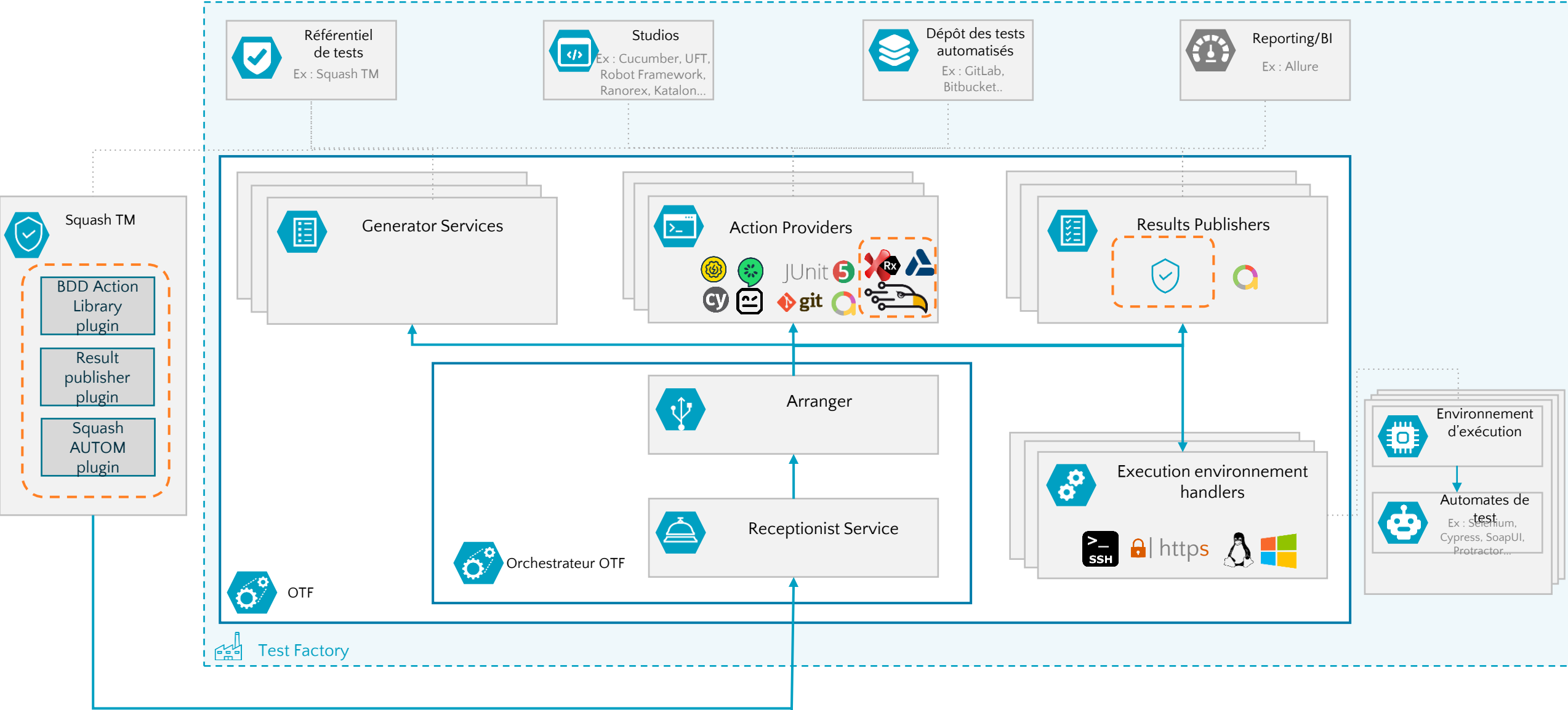
ARCHITECTURE : LES MICROSERVICES D'OTF (STANDALONE)



ARCHITECTURE : LES MICROSERVICES D'OTF (INTÉGRÉ À CI/CD)



SQUASH AUTOM : LES COMPOSANTS DE LA DISTRIBUTION



SQUASH DEVOPS : LES COMPOSANTS DE LA DISTRIBUTION

