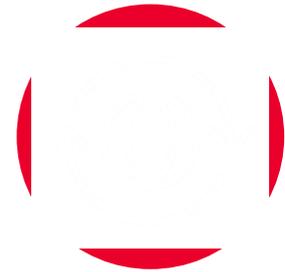




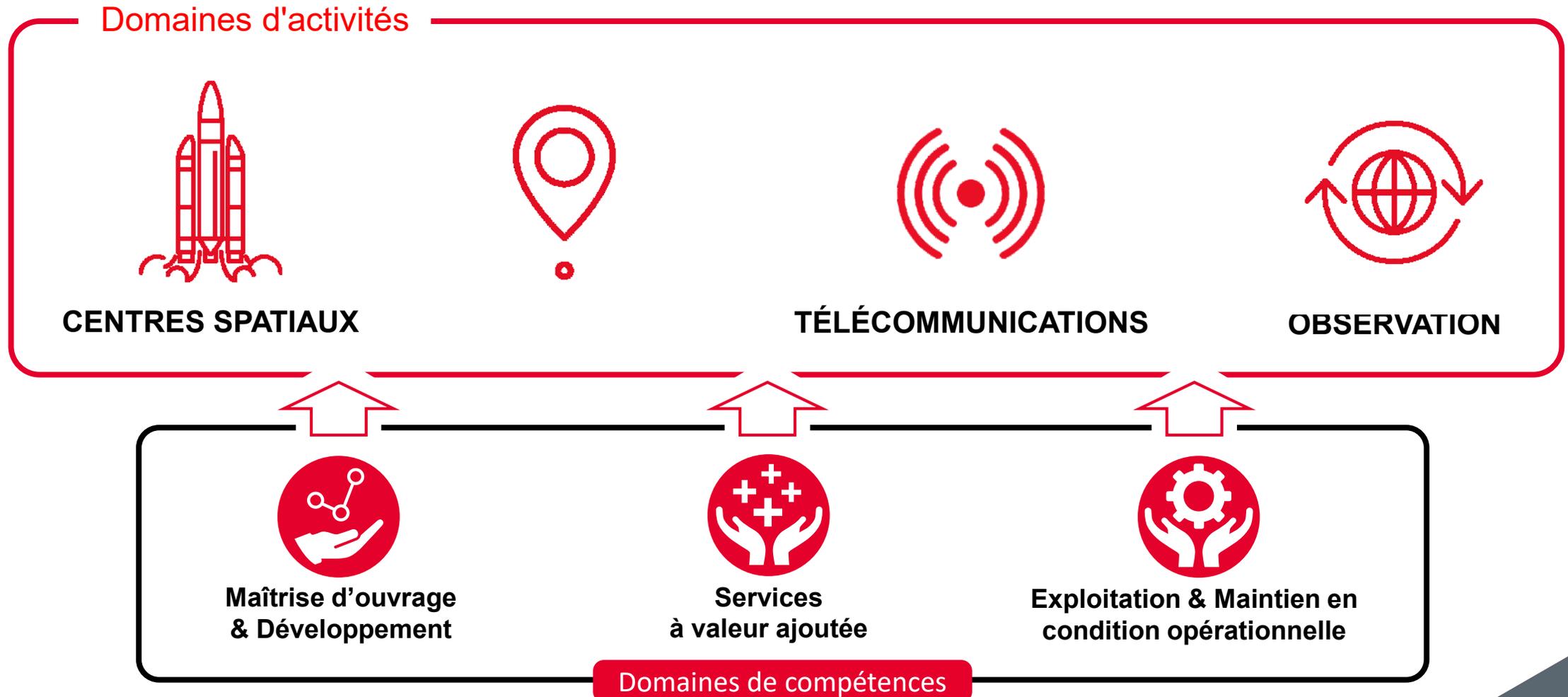
“ **Orchestration de
Traitements Distribués** ”

STORM – REX Chorégraphie



Nos métiers

Telespazio est présent sur les principaux domaines d'activités du spatial et son expertise se déploie à travers quatre grandes familles de services.





“ **Produit STORM et les
services de Surveillance
Maritime** ”



La surveillance Maritime



NOS SERVICES

De l'analyse contextuelle jusqu'au soutien aux opérations

Des services évolutifs selon les besoins et les contraintes

Facile à utiliser grâce à un accès en ligne

- **Détection de navires**
Détection automatique des navires sur l'imagerie spatiale open source
- **Services AIS**
Identification d'enjeux maritimes grâce à l'analyse de l'AIS
- **Connaissance de zones**
Compréhension des problématiques d'une zone d'intérêt
- **Compte rendu d'activité maritime**
Détection, classification et identification
- **Analyse de site d'intérêt maritime**
Image renseignée des sites stratégiques

- Le service est accessible via un simple **portail web** et s'articule autour des trois grandes fonctions qui sont :
 - Service **24/7**
 - Gestion de communautés et utilisateurs
 - Support technique et opérationnel pour l'accès aux données spatiales
- Le service inclut des **services** de détections **automatiques** basé sur de l'IA

- Les clients





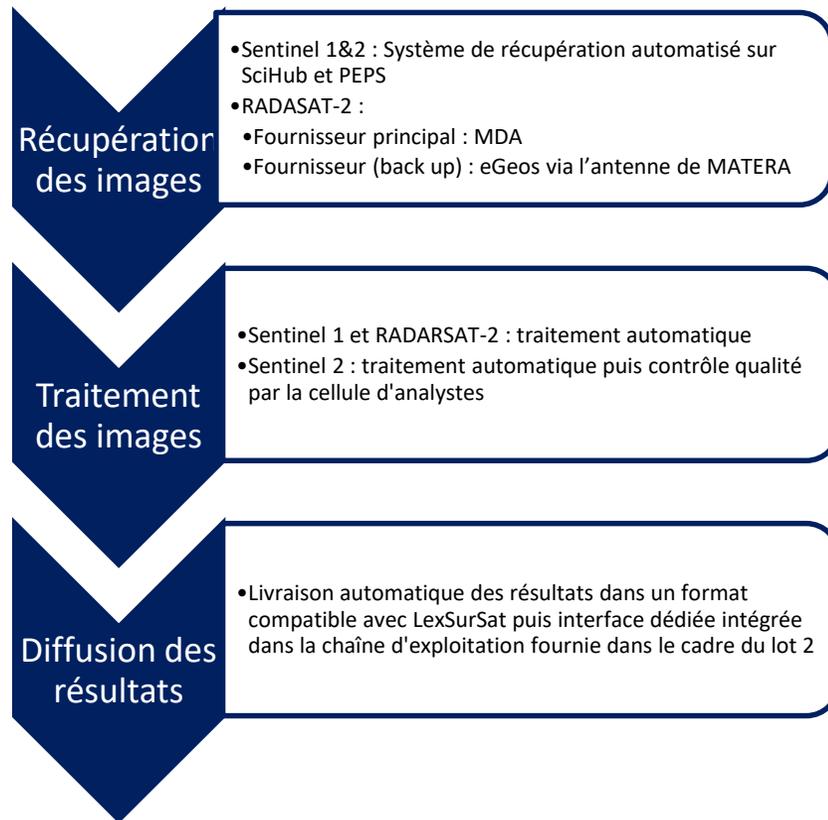
Les critères et choix de conception

- Point de départ en 2017
 - réutilisation et migration d'une application monolithe conçue en 2014 mais avec un volonté
 - d'évoluer vers **une architecture en micro services**
 - de rajouter des capacités de **traitements automatiques**
- Justification des Choix d'architecture :
 - **24 / 7** -> **Résilience** du service pour tenir le SLA client
 - **Détection Automatique**
 - Pouvoir « **scaler** » facilement sur une infra de type Cloud
 - Traiter un gros volume d'image
 - Tenir une « **timeliness** » sur les exigences de Near Real Time
 - Un service amené à **évoluer** avec de nouvelles fonctionnalités métier
 - Le micro service pour faciliter les évolutions et la maintenance.



Le cas d'utilisation nominal

Processus à implémenter

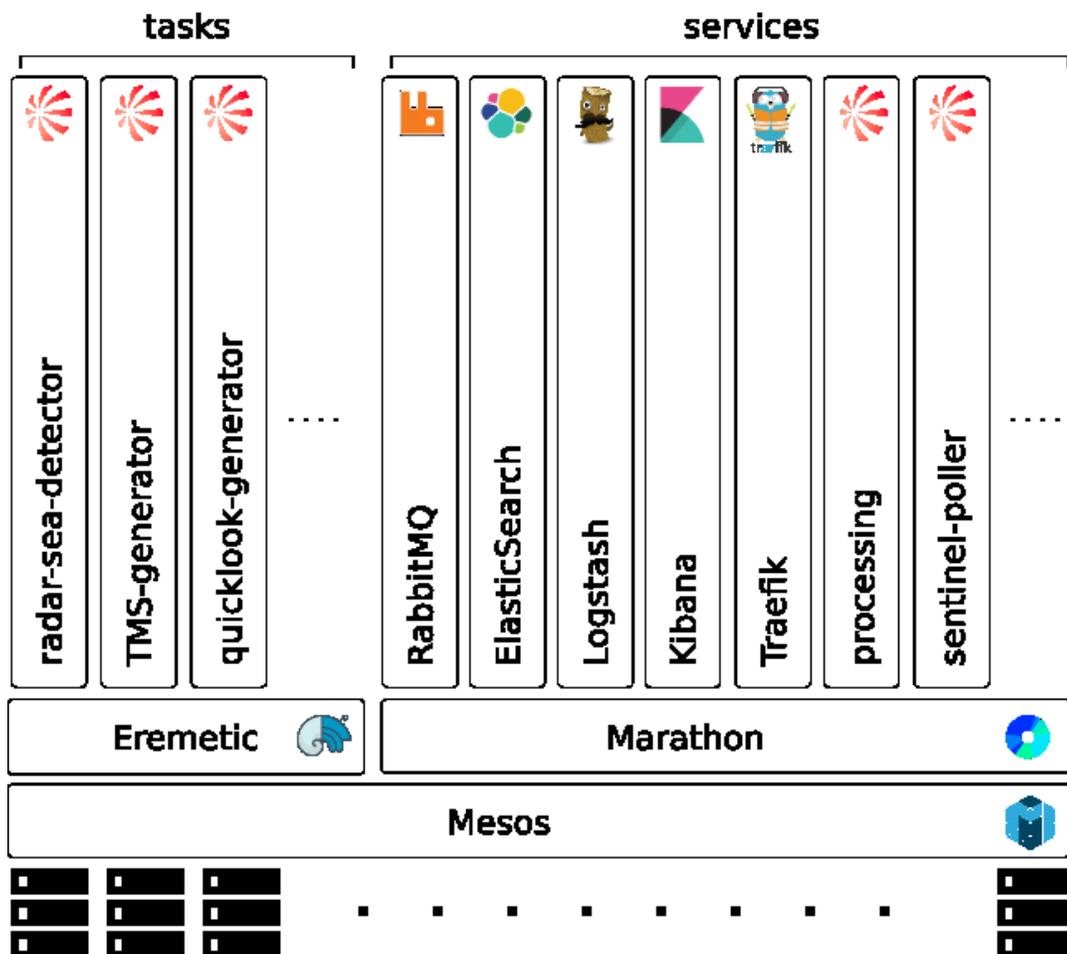


Doit être :

- Evolutif pour embarquer plusieurs processus métiers
-
- Ingérer des données sur plusieurs sources (AIS, images satellite, météo)

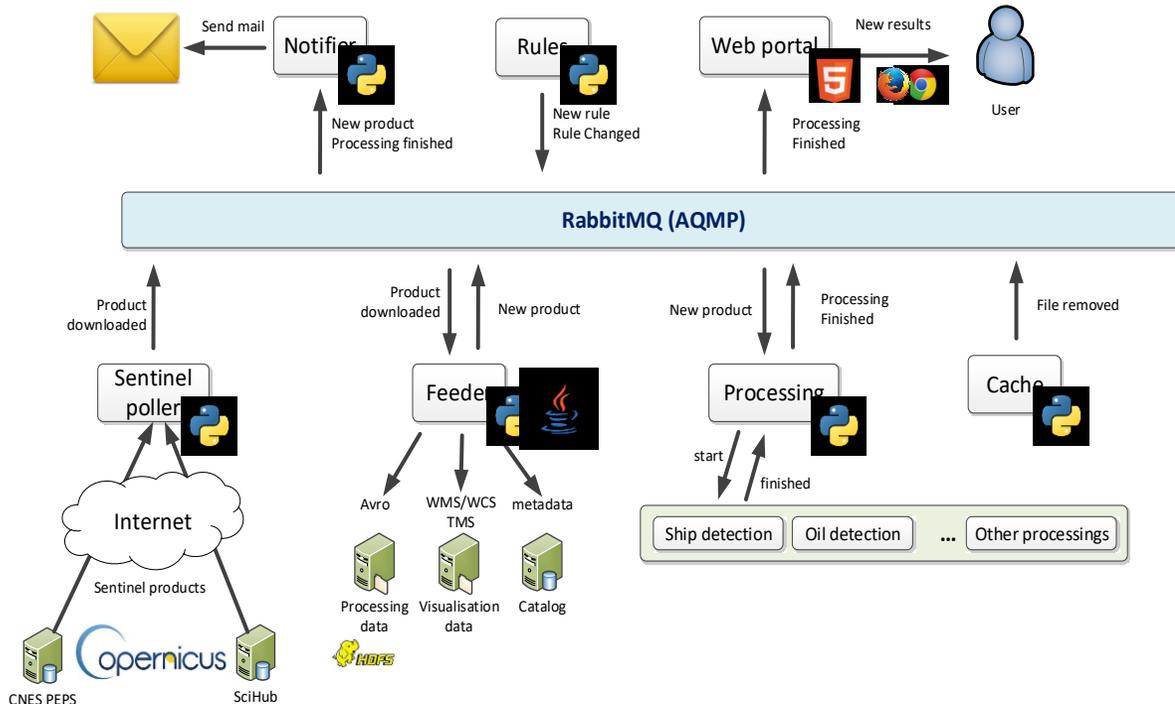


ET VOILA - l'Architecture STORM des services automatiques



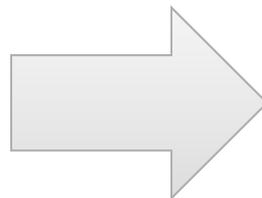
- Basé sur la techno MESOS / MARATHON
- Eremetic en complément de marathon pour l'orchestration des tâches (Docker) notamment pour gérer les fins de tâches avec un mécanisme de callback
- Le système comprend **plus de 20 tâches**

La chorégraphie des services automatiques



Une chorégraphie (asynchrone) à l'aide du BUS Rabbit MQ

- Les services sont **notifiés** par abonnement à un événement et/ou notifiant d'un événement. C'est ainsi que la chorégraphie est implémentée.
- Le service **Processing** permet l'exécution de n'importe quel algorithme métier au travers d'Eremitic
- le processus métier (orchestré) est implémenté en dur dans la tâche dite de « Processing »



• L'Euphorie de la réalisation !!!!



- On innove, on prend des risques
- **Mais** on a testé toutes les technos dans des POC
- On est **confiant** dans le planning

Et puis la Désillusion

- Une Version 1 pas au niveau des attentes :
 - Plateforme on premise
complexité de mise en œuvre et exploitation
 - Problèmes de stabilité du système
Constat d'un manque d'expertise sur Rabbit
Un service discovery « fait maison » qui n'est pas satisfaisant.
Performances pas au niveau attendu du SLA.
Manque de monitoring de la plateforme
- Processus chorégraphié instable et peu maintenable
La surveillance du processus est difficile à suivre (les logs ne sont plus séquentiels dans un seul fichier mais répartis sur plusieurs noeuds)
Mauvaise gestion des erreurs et reprise - Le workflow se relance bien sur erreur mais de manière erratique (la bonne exécution n'est pas forcément la dernière ☹).



Scalabilité et Résilience pas au rendez-vous



Les leçons apprises

- POC de 6 mois sur l'orchestration Mesos /Marathon / Eremetic ont permis de comprendre
 - son utilisation
 - mais pas son administration

- Le Bus Rabbit n'est pas utilisé correctement – Exemple : message en mode « Fire and Forget »

- Le Bus n'est pas redondé donc problème de résilience

- La base Elastic est mal employée et peu performante
 - Big Data certes mais il faut structurer plus finement les données pour ne stocker que le nécessaire.

- Il faut plus de monitoring et des logs plus explicites
 - La logique métier se dilue sur tous les nœuds,
 - on ne comprend plus où et quand s'exécutent les tâches



Les solutions pour obtenir une Version 2 stable

- Des **formations et support d'experts** (Rabbit , Elastic Search, ...)
- Amélioration de la **résilience** sur le bus Rabbit avec :
 - Un **Acquittement** : contrôle de l'acquittement du routage des messages
on s'assure que Rabbit a distribué le message vers au moins une file où le message doit être distribué puis le message est ré émis si nécessaire
 - **Le nommage et durabilité des files** pour identifier facilement les files et s'assurer du maintien de la topologie.
le que on s'assure que la file est toujours alimentée de messages et dans le cas d'un redémarrage d'un service consommateur que celui est toujours alimenté en messages.
 - Un Déploiement sur plusieurs nœud Rabbit
 - Implémentation de la persistance des messages
- Plus de logs avec des identifiants métier (id image/date) pour tracer l'avancement des processus métier.

La scalabilité et la robustesse de la chorégraphie sont acquises



“ **Produit Maas**
Monitoring des segments
sols Copernicus

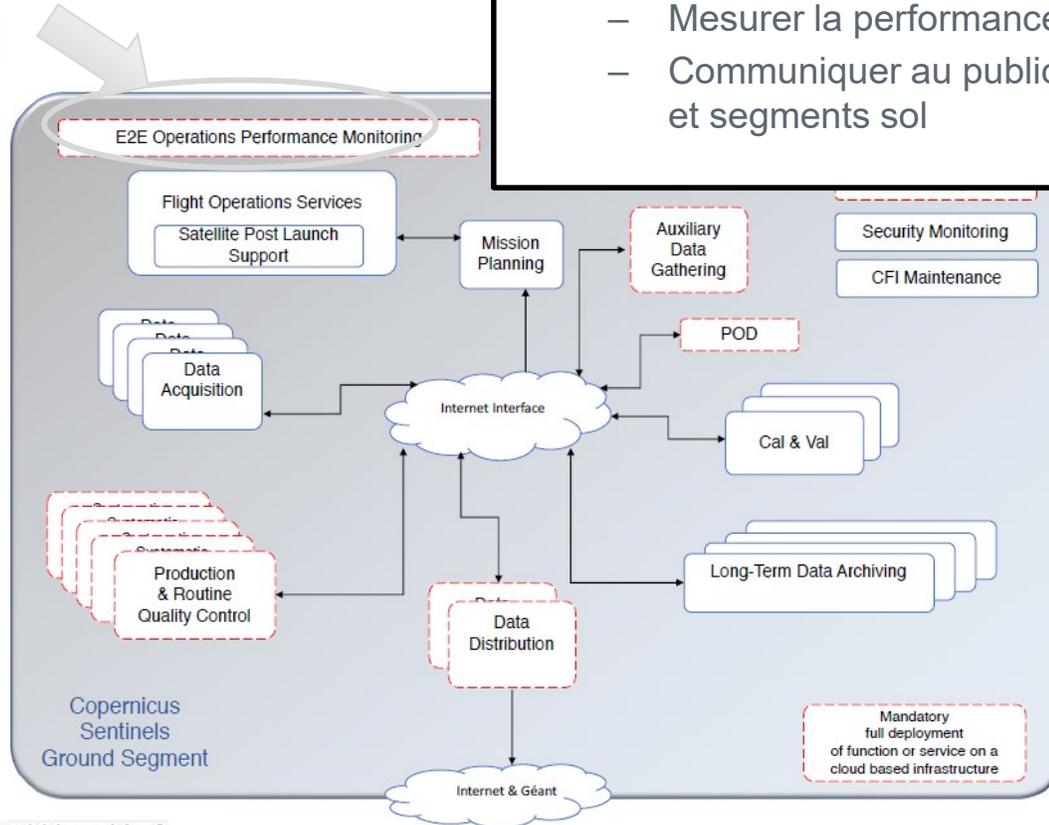
”



Le Monitoring sur le segment sol Copernicus

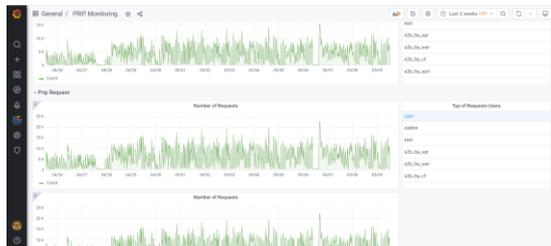


Vous êtes ici



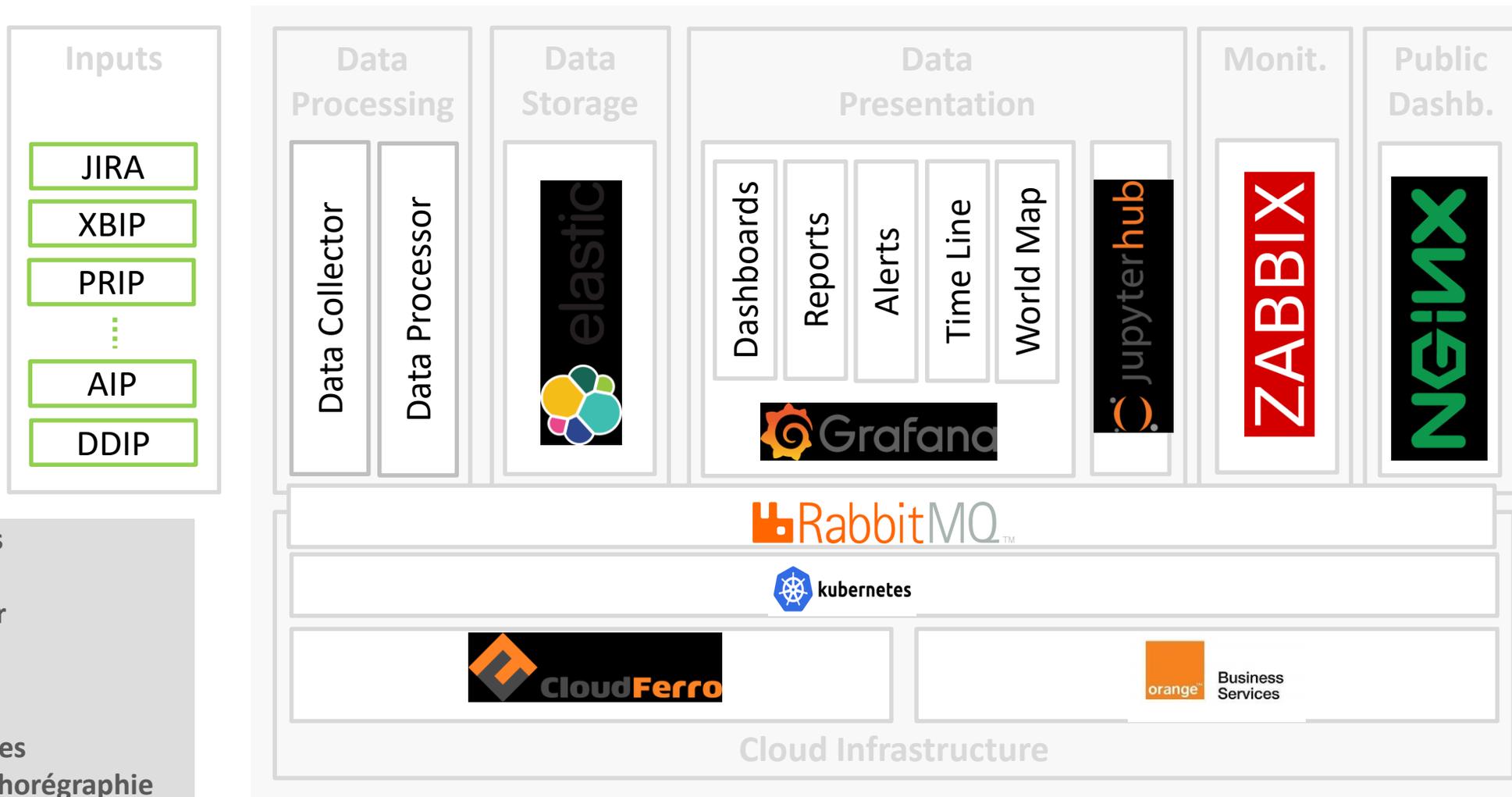
- Un système pour :
 - Mesurer les SLA des différents systèmes du segment sol
 - Mesurer la performance E2E
 - Communiquer au public le statut des satellites et segments sol

- Un système devant traiter :
 - Un gros volume de données
 - un SLA à garantir 24/7
 - Des interfaces avec beaucoup de systèmes





Architecture du Maas (Monitoring As A Service)



- Un archi micro services
- Open Source
- Basée sur Grafana pour
 - Dashboards
 - Reporting
 - Alerting
- Ainsi que sur Kubernetes
- Et Rabbit MQ pour la chorégraphie

Rabbit MQ pour la chorégraphie reprenant les bons principes appris auparavant



Les améliorations

- Un travail de conception sur Les **micro services**
 - ils ne sont pas qu'un simple découpage fonctionnel du monolithe
 - Il faut les rendre **résilients**
 - On doit pouvoir tuer les services et ils doivent repartir là où ils en étaient sur n'importe quel noeud
 - Ils doivent être **stateless** et robustes aux **interface externes**
 - les rendre plus **léger**
- Renforcement des **processus de développement**
 - Conception plus poussée sur le génie logiciel
 - Un framework pour cadrer les développements
 - Pair programming
 - Mécanisme de merge request pour éviter les instabilités dans le code
- Les avantages de **Kubernetes** sur Mesos / Marathon
 - Yaml un langage permettant de faciliter la configuration et spécifications d'éléments
 - Un service discovery intégré et robuste
 - Un éco système beaucoup plus complet pour le déploiement et l'orchestration
- Les difficultés
 - Installation /Administration d'un cluster Kub aussi (voire plus) complexe qu'un cluster MESOS/Marathon
 - Privilégier si possible les instancesaaS.



Les apports sur le Maas

Un système Résilient et scalable !

Une capacité de montée à l'échelle

Tenue du SLA 24/7



“ Conclusions

”



Les conclusions du REX

- La **Formation** est primordiale en amont car les concepts sont complexes
 - De l'orchestration des machines et docker – L'administration de ces clusters est complexes
 - De la chorégraphie des processus - Attention au plat de nouilles qui arrive très vite

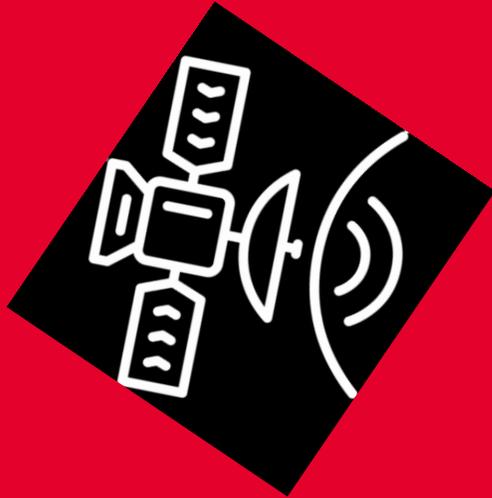
– Il faut une maîtrise parfaite des technologies et de son paramétrage

- La **Résilience** est donc difficile à maîtriser et à obtenir
Il a fallu **2 ans** pour maturer les technologies dans notre équipe

Au-delà de la Hype il faut vraiment se poser la question de l'intérêt d'une l'architecture micro services et de processus chorégraphiés par rapport au besoin.



Avez-vous des questions ?



Donnez-nous une orbite
et un satellite et nous
réinventerons la Terre !