

SVOM Data Challenge-1

Contribution GEPI

Isabelle Jégouzo

Jesse Palmerio

Susanna Vergani

Objectifs 2019 en résumé



- 1) Faire le DC-0
- 2) Faire le DC-1
- 3) Mettre en place le lien avec l'équipe chinoise sur les interactions du pipeline VT

1) DC-0:

- Développement et mise en place d'un service / container sans traitement, capable de communiquer avec les autres services

→ réalisé mi 2019

2) DC-1:

- Tâches réalisées
 - réception d'une liste de sources (filtre R ou filtre V) (sortie d'un pré-traitement du pipeline VT)
 - réception d'une position du MXT (QPO_MXT)
 - test des outils de cross-match, requête http, format de données.
 - gestion des données de sortie (produit QPO_VT): stockage dans la SDB et notification (BA et post-traitement Pipeline VT)
- Tâches en cours ou reportées au DC-2:
 - combinaison des listes en filtre R et V
 - traitement des listes avec la position du MXT
 - gestion des données intermédiaires (base de donnée locale ou utilisation du runtime_param de pipeline-bricks ?)

3) Mettre en place le lien avec l'équipe chinoise sur les interactions du pipeline VT

- Dialogue avec interlocuteurs chinois pour la définition des interfaces et des interactions entre les services de traitement du Pipeline VT (formats, data notification...)

→ réalisé juin - décembre 2019

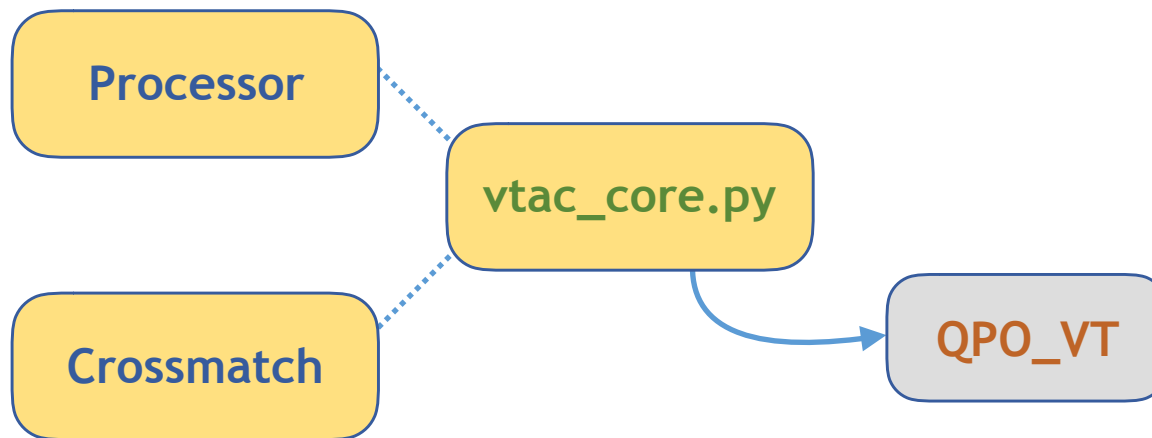
Objectifs 2019 en résumé



- 1) Faire le DC-0
- 2) Faire le DC-1
- 3) Mettre en place le lien avec l'équipe chinoise sur les interactions du pipeline VT

→ Objectifs réalisés

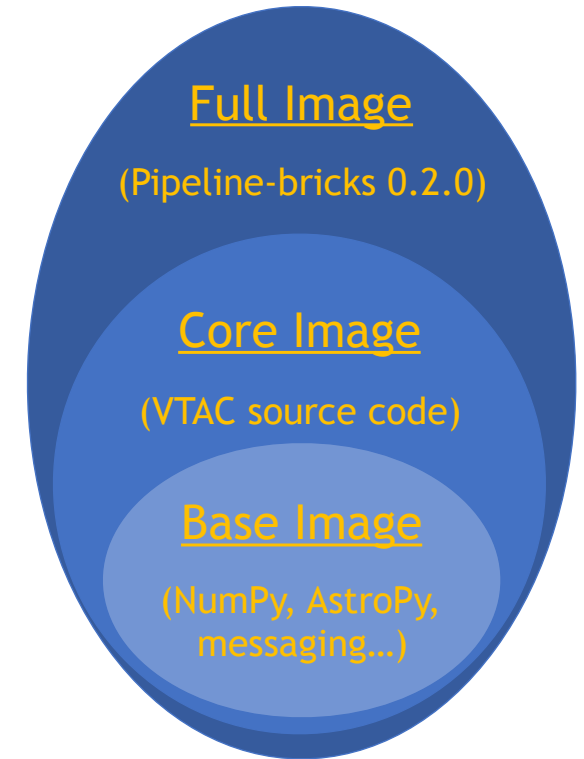
- Service composé de 2 modules python et d'un script python exécutable pour créer le produit QPO_VT



Le service VTAC (0.4) en détail



- Image docker finale construite en plusieurs étapes:
 - base_image contenant les installations stables (NumPy, AstroPy, messaging...)
 - core_image construite à partir de la base_image et contenant en plus les codes sources de VTAC
 - full_image construite à l'aide du pipeline-maker, ajoutant la couche pipeline-bricks (0.2.0) au dessus de la core_image



Le service VTAC (0.4) en détail



- Qualité du code:
 - pylint : 9.85/10
 - couverture de test : 69%
 - Sonarqube
- Mise en place de la CI

