



# LAL: Bilan DC1

Nicolas Leroy, Serge Du

Montpellier –21- Janvier 2020

# MXT Analyse rapide VHF

- Projet Gitlab mxt/mxt-ql
- fonctions de base pour la production des différents produits issus du traitement rapide des données VHF de MXT
  - timeandflux\_task.py :
    - DT\_MXT et QF\_MXT
  - lightcurve\_task.py:
    - QLC\_MXT quick look light curve
    - QTI\_MXT associated temporal index
  - position\_task.py
    - QPO\_MXT quick look position
- Développement:
  - branches git « develop » et « master » avec des branches perso pour dev. de fonctionnalités. Nous sommes 2 développeurs.
  - Intégration Continue et Contrôle Qualité (serveur sonarqube local)

# Test,Coverage

Name	Stmts	Miss	Coverage
-----			
src/lightvurve.py	52	5	90%
src/lightvurve_task.py	50	4	92%
src/locvhf_2_fits.py	104	0	100%
src/position_task.py	113	17	85%
src/timeandflux_task.py	113	13	88%
-----			
TOTAL	432	39	91%

# Implémentation: mxtql\_pipeline

- Intégration architecture Pipeline-bricks
  - => conformité aux spécifications REST + messaging
  - => orchestrable.
- Automatisée par Pipeline-maker
- Importation des produits dans SDB
  - Utilisation de messaging/Sdblo
- Déploiement via Portainer
  - Fichiers de conf dans svom\_deployment/swarm/mxt-ql
- Validation aussi via Portainer (console):
  - Instructions sur le wiki data-challenge

# Bilan DC1

- Spécifications satisfaites (Polarion-DC1)
  - Pipeline fonctionne et renvoie des produits scientifiques stockés dans SDB (pour 2 d'entre eux)
    - `lightcurve_task.py` => QLC\_MXT and QTI\_MX
    - `position_task.py` => QPO-MXT
    - `timeandflux_task.py` => QF\_MXT and DT\_MXT
- Implémentation facilitée par
  - `pipeline-maker` (`pipeline-bricks`) et `Sdblo`
- Déploiement et Validation facilités par Portainer