

# SemiHackathon ASOV 2016

Présentation du cahier des charges d'un nouveau projet visant à piloter différents logiciels de traitement depuis un client lourd Java en s'appuyant sur UWS (UWSlib), partage conseils, retours d'expériences, idées...

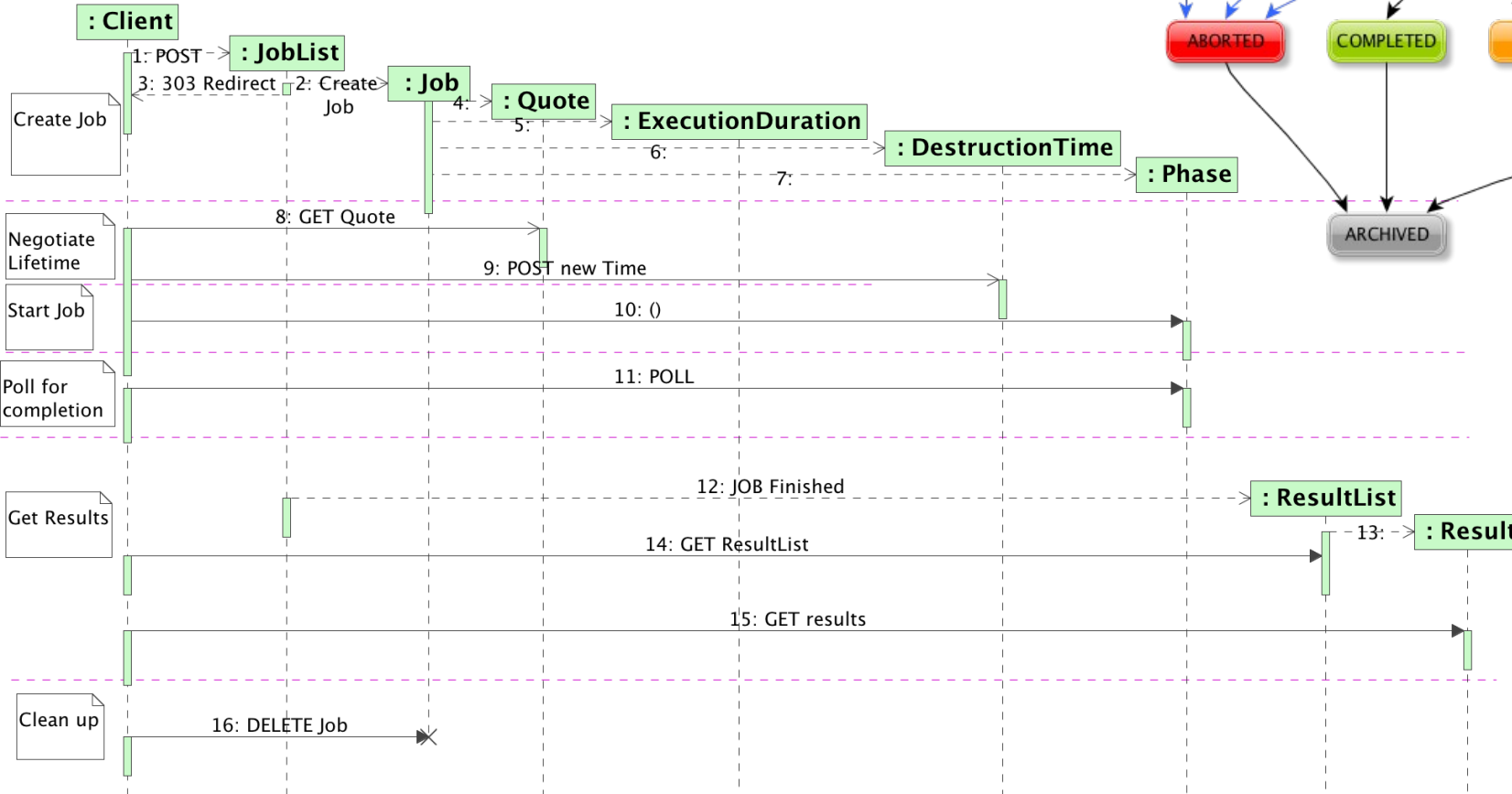
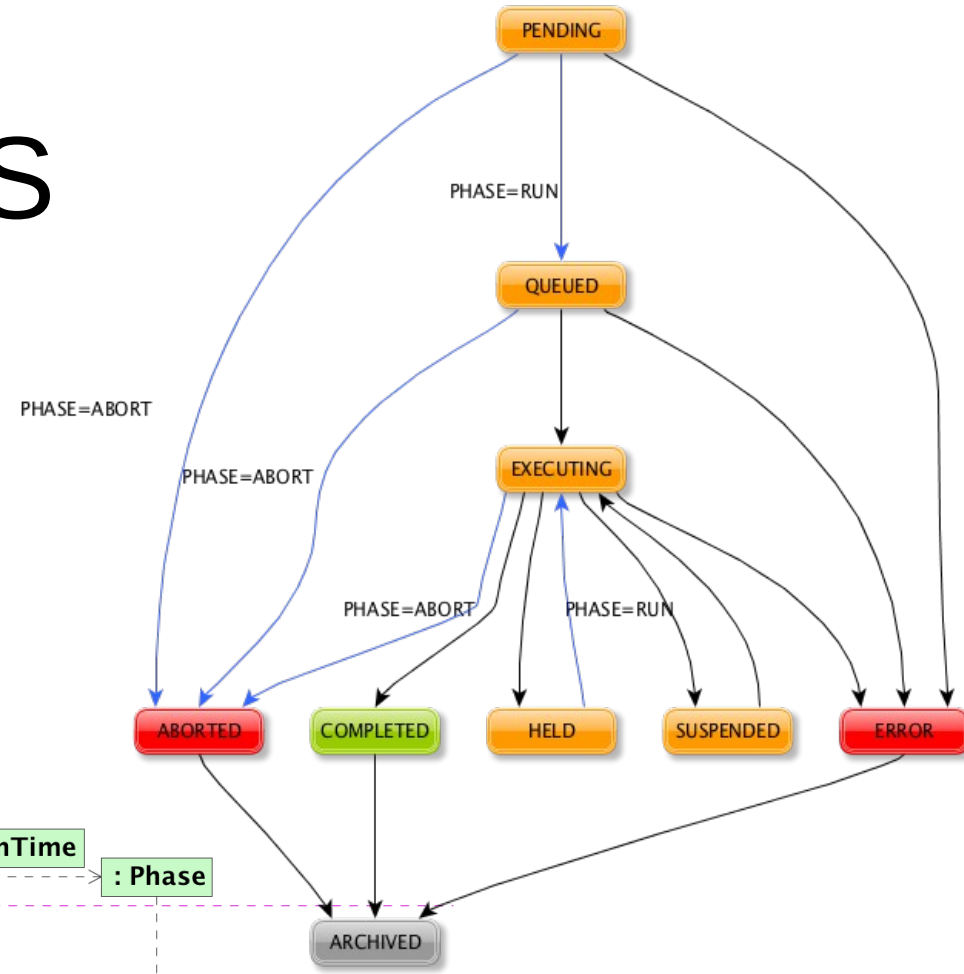
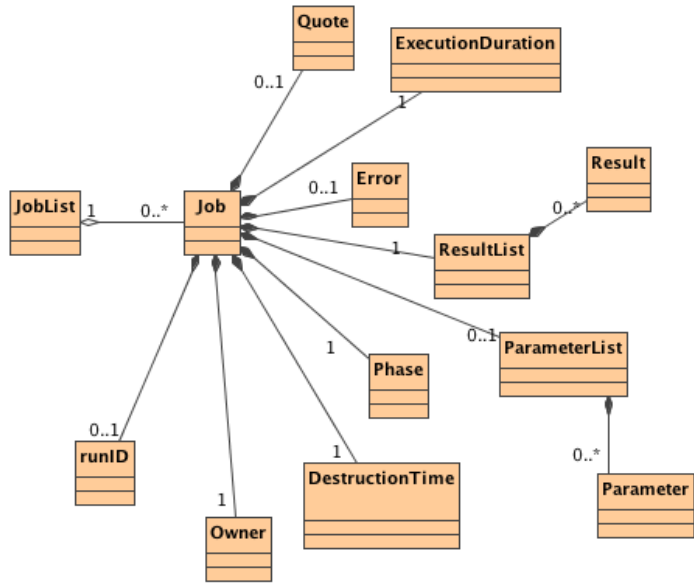
- Mini rappel UWS
- Présentation du contexte scientifique
- Présentation des contraintes techniques
  - Tour de table sur les projets similaires
- Revue des points durs identifiés à priori
  - échanges / conseils

# Mini rappel UWS

<http://www.ivoa.net/documents/UWS>

- standard IVOA du groupe Grid&WebService
- 1<sup>er</sup> draft en 2005
- recommandation en V1.0 depuis Oct. 2010
- abstract de la ProposedRecommandation V1.1:
  - « The Universal Worker Service (UWS) pattern defines how to manage asynchronous execution of jobs on a service. Any application of the pattern defines a family of related services with a common service contract. Possible uses of the pattern are also described. »

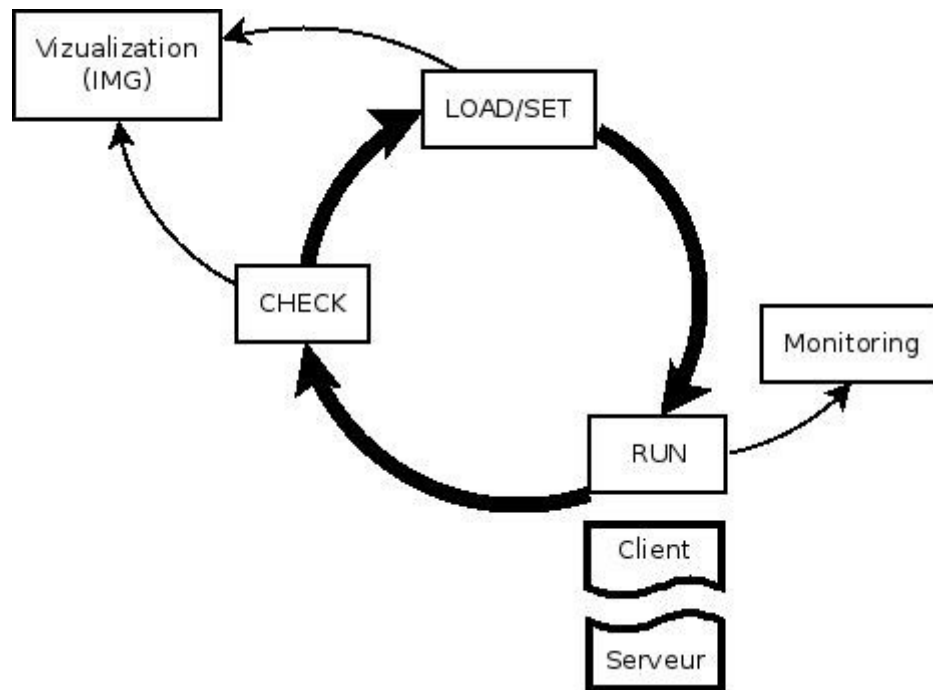
# UWS



# Contexte scientifique

- Faciliter l'utilisation de plusieurs algorithmes de reconstructions d'image
- Fournir une interface de pilotage à l'utilisateur capable de s'interfacer avec plusieurs logiciels de traitement
  - Eviter à l'utilisateur l'installation des logiciels de traitements
- Phases du projets
  - Document de standardisation des interfaces (data, parameters) : première version ok !
  - Mise aux normes : premier logiciel de traitement ok !
  - Réalisation de la version initiale du GUI : ça démarre...

# Scenario simplifié



# Contraintes techniques

- S'appuyer sur un framework existant pour gérer l'asynchrone stateful : UWS
- Rester en accord avec des briques techniques de nos autres services :
  - Client java jMcs
  - Serveur java taplib / uwslib
- Si possible aller plus loin que le mode simple client serveur :
  - Pouvoir mettre à l'échelle sur une grille de calcul
  - Pouvoir être indépendant du réseau si les logiciels sont sur la machine de l'utilisateur

# Echanges

- S'appuyer sur un framework existant pour gérer l'asynchrone stateful

Alternatives ?

- Rester en accord avec des briques techniques de nos autres services :

De manière générale, quelle est la difficulté de mise en œuvre ? Points à surveiller (resources...) ?

Quels clients / services utilisent uwslib ?



# Echanges

- Si possible aller plus loin que le mode simple client serveur :
  - Pouvoir mettre à l'échelle sur une grille de calcul
    - Quels retours d'expériences ?
  - Pouvoir être indépendant du réseau si les logiciels sont sur la machine de l'utilisateur
    - Docker peut certainement aider ?)