Concevoir et partager des workflows d'analyse de données. Application aux traitements intensifs en bioinformatique

François Moreews - GENSCALE

INRA / IRISA

11 décembre, 2015

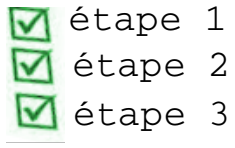
sous la direction de Dominique Lavenier

Introduction
Contexte

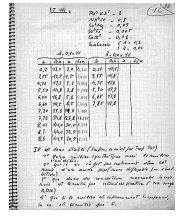
- Open-science
- Analyse de données à haut-débit
- Elaboration de répertoires de chaînes de traitements
- Usage de plus en plus répandu des systèmes de gestion de workflows chez les bio-analystes

Introduction
Définitions

workflow : **protocole *in silico*** correspondant à une expérience

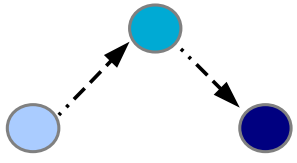


Un répertoire de workflows est alors équivalent à un type de **cahier de laboratoire**



étape du workflow => **acteur/composant**

workflow => **composition, orchestration**



Introduction

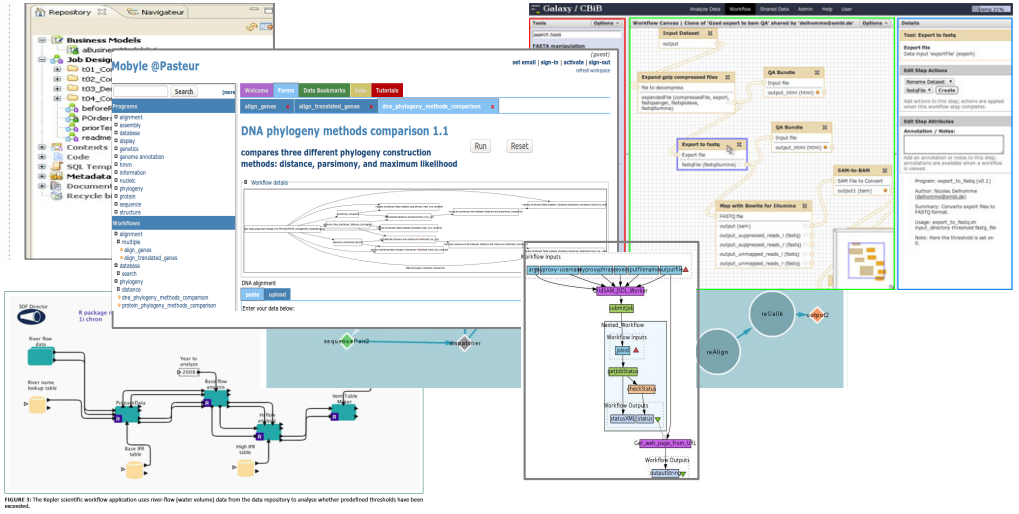
Les promesses de l'approche workflow

Promesses

- accessibilité des méthodes,
- maintenabilité,
- reproductibilité, publication "executable paper",
- échanges facilités, meilleure diffusion des nouvelles méthodes, feed-back des utilisateurs, accélération de la science par une mise en commun.

Introduction
les WfMS Scientifiques.

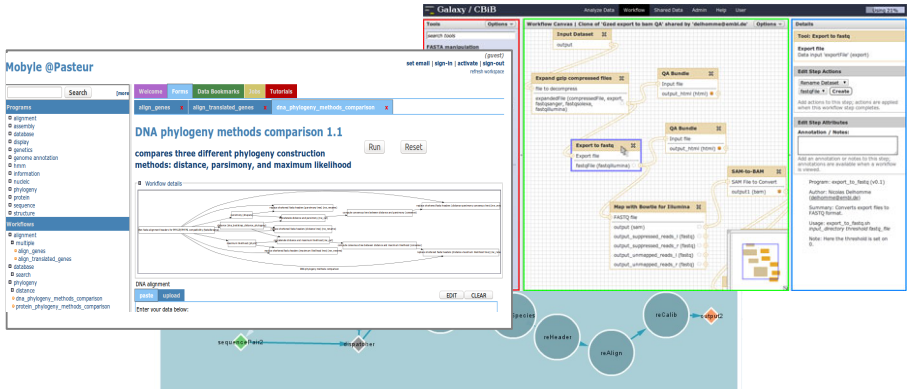
Diversité des WfMS Scientifiques.



Composants consommant et produisant des données, sans état, "boites noires", intégrant des lignes de commandes ou des clients

Introduction
les WfMS Scientifiques Intégratifs.

WfMS Scientifiques intégratifs (Galaxy, Mobyle...)
collections d'outils thématiques et **classification** des méthodes pré-intégrées, gestion de certains **types de données**, fonctionnalités collaboratives



→ **capture une partie de la connaissance d'un domaine** ou d'une discipline (types de données et méthodes).

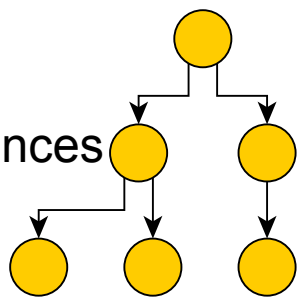
Objectifs

Une plus grande efficacité des moteurs de workflows des WfMS

Efficacité limitée des moteurs de workflows des WfMS scientifiques pour le traitement d'analyse **intensive** de données

- **Le modèle DataFlow : le modèle de calcul le plus courant.**

modèle DataFlow :
- un D.A.G.
- arrêtes: dépendances de données



noeuds=acteurs/composants
- avec des ports d'entrée et de sortie
- peuvent encapsuler des lignes de commandes

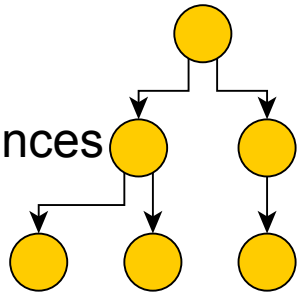
Objectifs

Une plus grande efficacité des moteurs de workflows des WfMS

Efficacité limitée des moteurs de workflows des WfMS scientifiques pour le traitement d'analyse **intensive** de données

- **Le modèle DataFlow : le modèle de calcul le plus courant.**

modèle DataFlow :
- un D.A.G.
- arrêtes: dépendances de données



noeuds=acteurs/composants
- avec des ports d'entrée et de sortie
- peuvent encapsuler des lignes de commandes

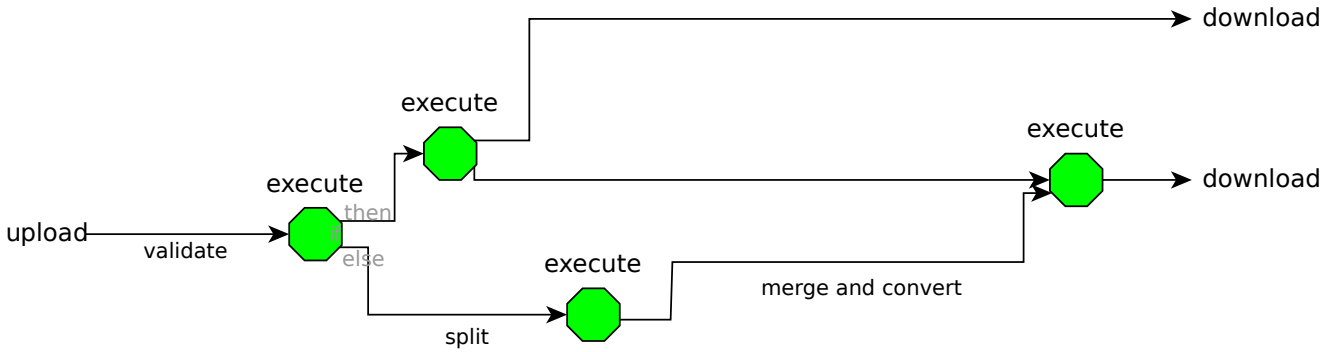
- **Aptitude au parallélisme des modèles DataFlow peu exploitée**

Objectifs

Accélérer et simplifier la conception des workflows

conception de worfklow

- ce que l'on aimerait

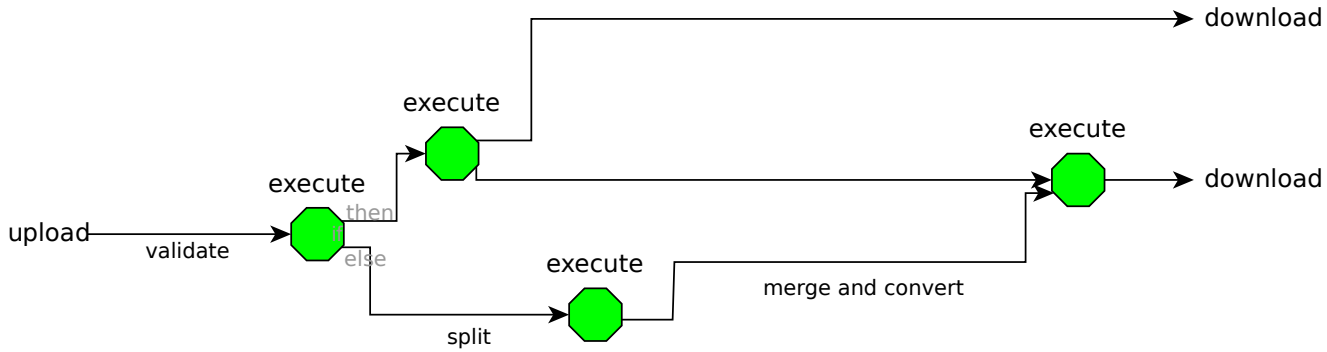


Objectifs

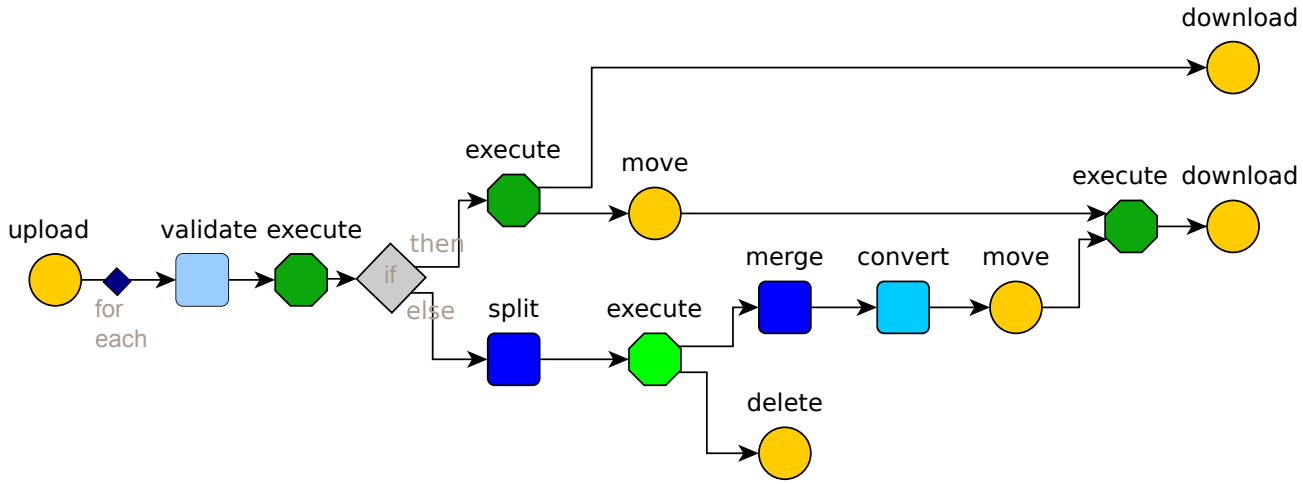
Accélérer et simplifier la conception des workflows

conception de worfklow

- ce que l'on aimerait



- ce que l'on fait



Objectifs

Garantir la pérénité et l'échanges des workflows

Les limitations des WfMS :

- **Complexité du déploiement** des chaînes de traitements $\rightarrow$ **un frein à leur diffusion** et à la reproductibilité.
- **Pas de format d'échange** suffisamment expressif et générique pour la diffusion des chaînes de traitements.

Objectifs

Pallier les limitations des WfMS précédemment identifiées concernant :

- la simplicité de conception des workflows,
  - la **localisation et routage** des données,
  - la **parallélisation**,
  - l'intégration des **composants techniques**.
- la pérénité et les échanges des workflows.
  - le **déploiement**.
  - la **diffusion**.

Nécessité de Formaliser un modèle de calcul dédié à l'analyse de données

Dans les systèmes de gestion de workflows actuels :

- Inadaptation des modèles de calculs aux **tâches longues** ;
- **Expressivité insuffisante** (boucles, conditions, variables, fonctions) ;
- Le **formalisme DataFlow non explicite ou non strict**.

$\Rightarrow$ **Nécessité préalable de concevoir un nouveau modèle de calcul** pour concevoir des moteurs de workflows efficaces adaptés à l'analyse intensive de données dans les WfMS.

Plan

Formalisation d'un modèle de calcul dédié

un modèle DataFlow

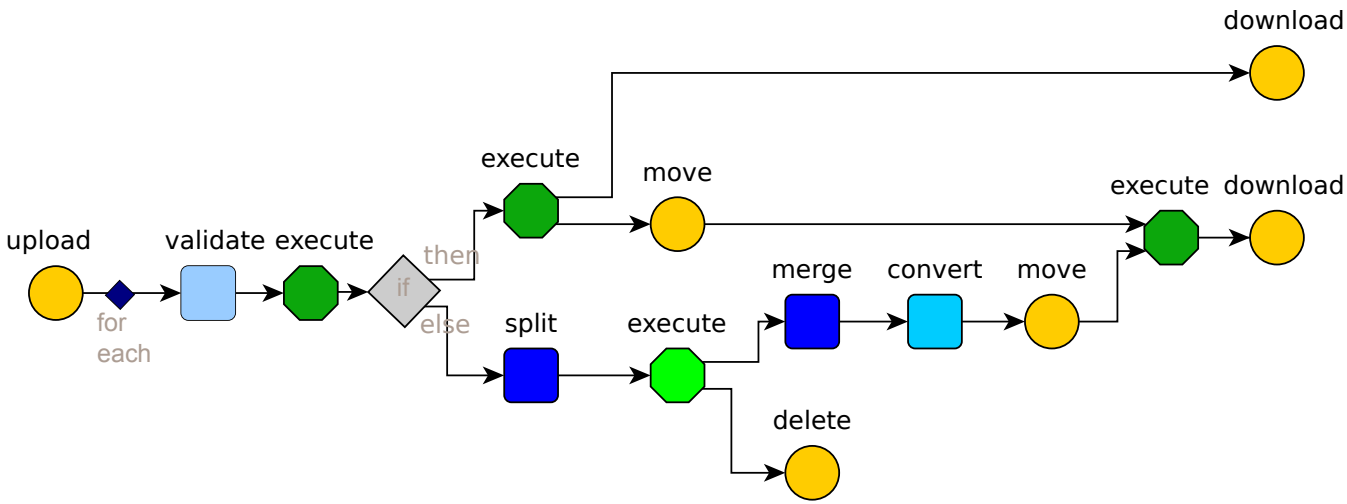
UDFAS (URI based DataFlow for Asynchronous Services).

- **DataFlow**
  D.A.G, arrêtes= dépendances de données

  noeuds=acteurs/composants (encapsulant des lignes de commandes)
- **Synchronous Data Flow** (SDF)
  nombre des jetons de données consommés et produits connu au moment de l'exécution.
  $\rightarrow$ détermination statique la séquence d'exécutions des acteurs (simplicité de l'ordonnancement)
- **UDFAS**
  1. Propriétes du SDF $\rightarrow$ **règle du jeton unique de sortie**.
  2. Ligne de commande : nombre de fichiers de sorties indéterminé.
     $\rightarrow$ **jetons abstraits** (containers et références)

Formalisation d'un modèle de calcul dédié

Expressivité du modèle



Formalisation d'un modèle de calcul dédié

Expressivité du modèle

- La **propagation des valeurs NULL** : employé pour intégrer le conditionnement (if-then-else) au DataFlow.

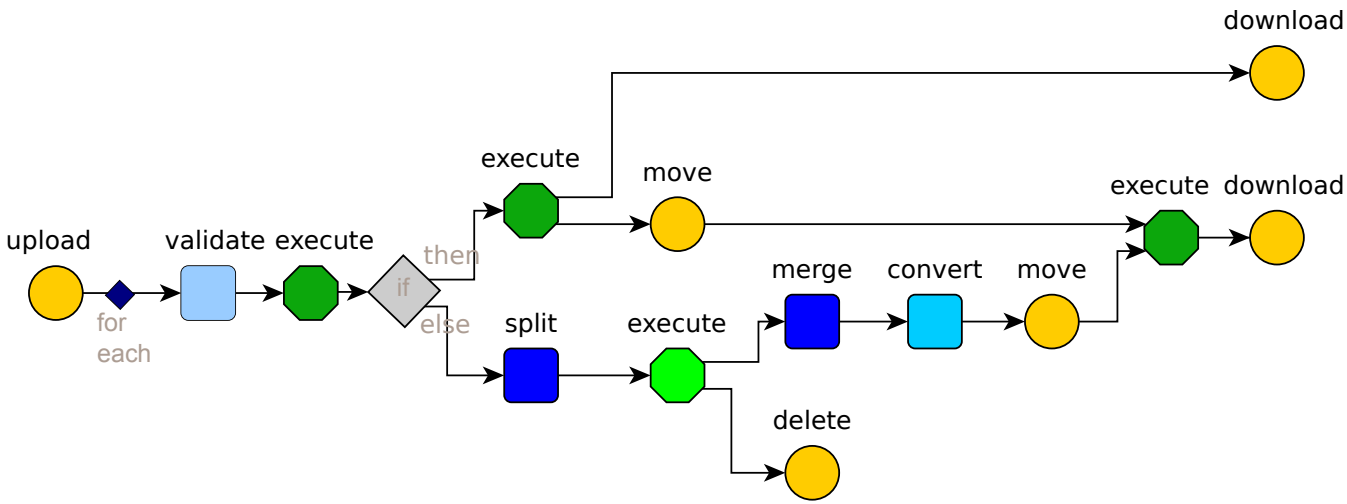
jeton d'entrée A	jeton d'entrée B	jeton de sortie C
NULL	NULL	**NULL**
NULL	jeton B	**NULL**
jeton A	jeton B	**jeton C**

- **Modèle hiérarchique**
  Mécanisme d'**acteur composite** : **convertir un workflow en un acteur**.
  $\rightarrow$ **référence un workflow fils**.

$\Rightarrow$ Patrons de conception garantissant l'**expressivité** dans un contexte **puremment DataFlow**

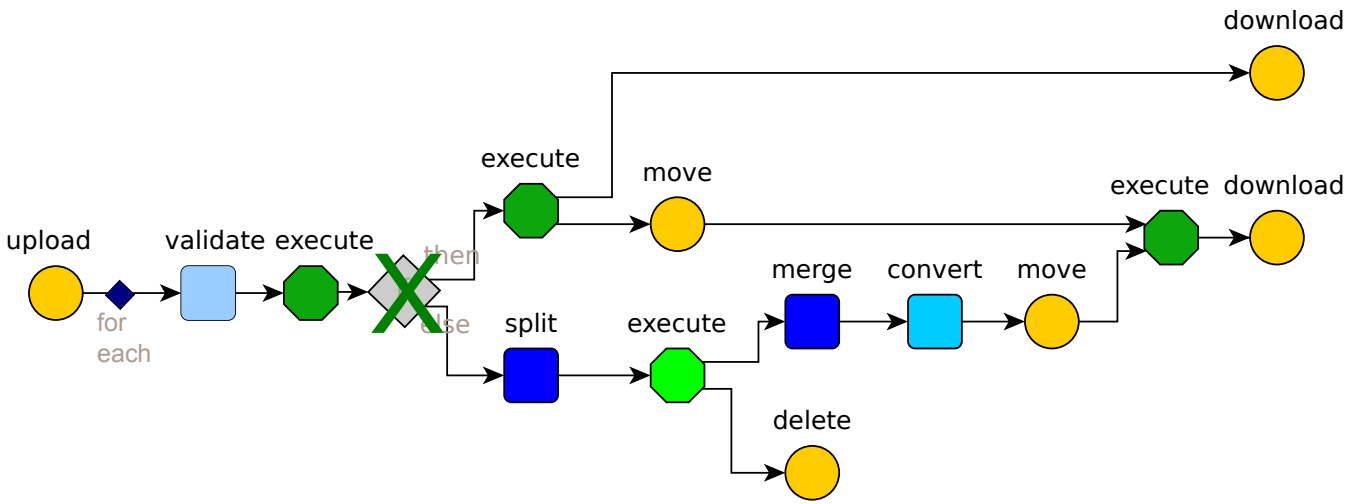
Formalisation d'un modèle de calcul dédié

Expressivité du modèle



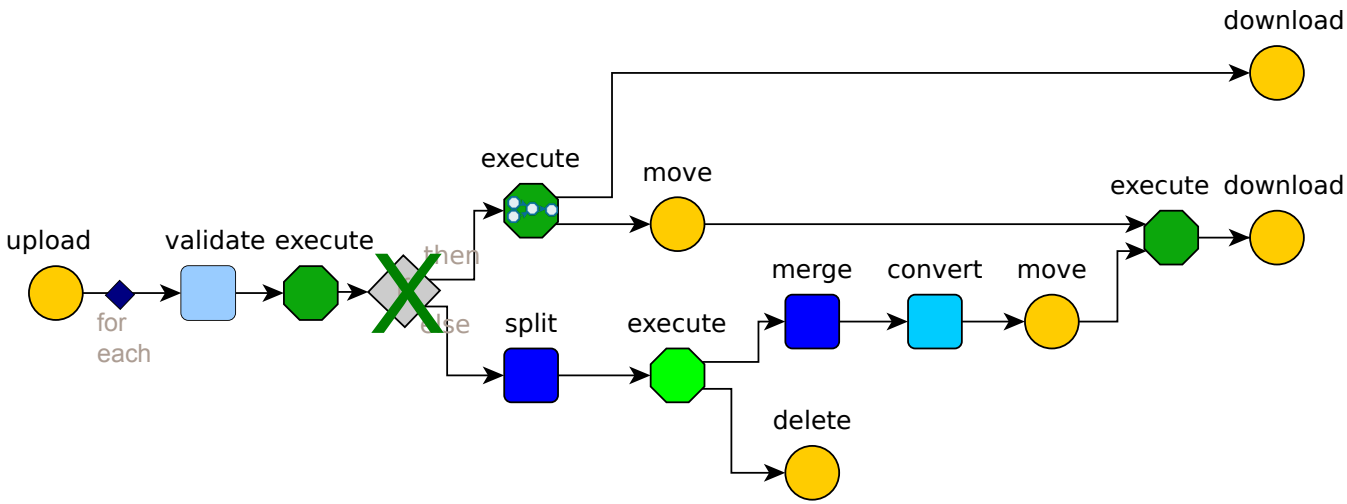
Formalisation d'un modèle de calcul dédié

Expressivité du modèle



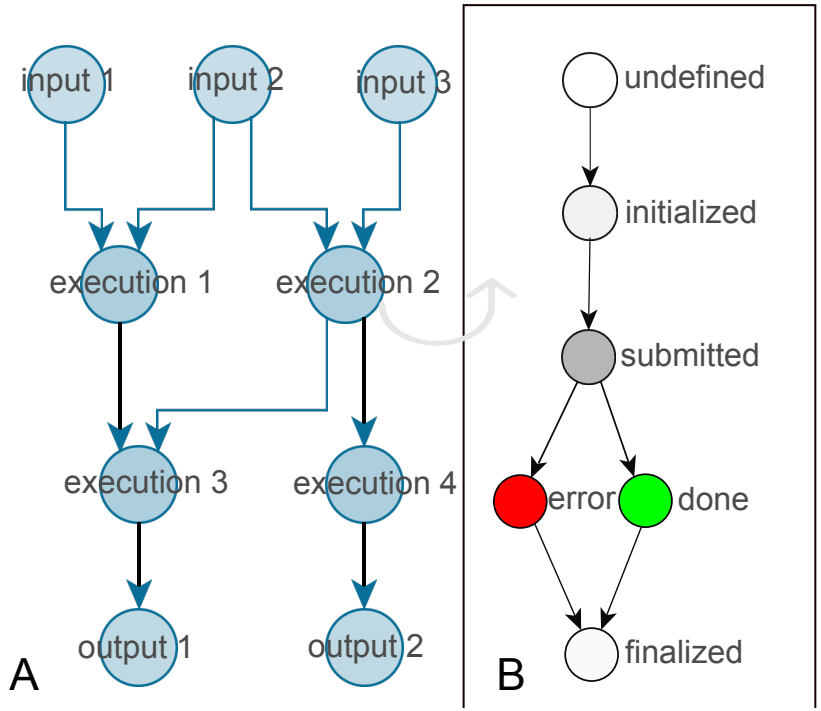
Formalisation d'un modèle de calcul dédié

Expressivité du modèle



Formalisation d'un modèle de calcul dédié

Adaptation aux architectures orientées service pour le calcul intensif



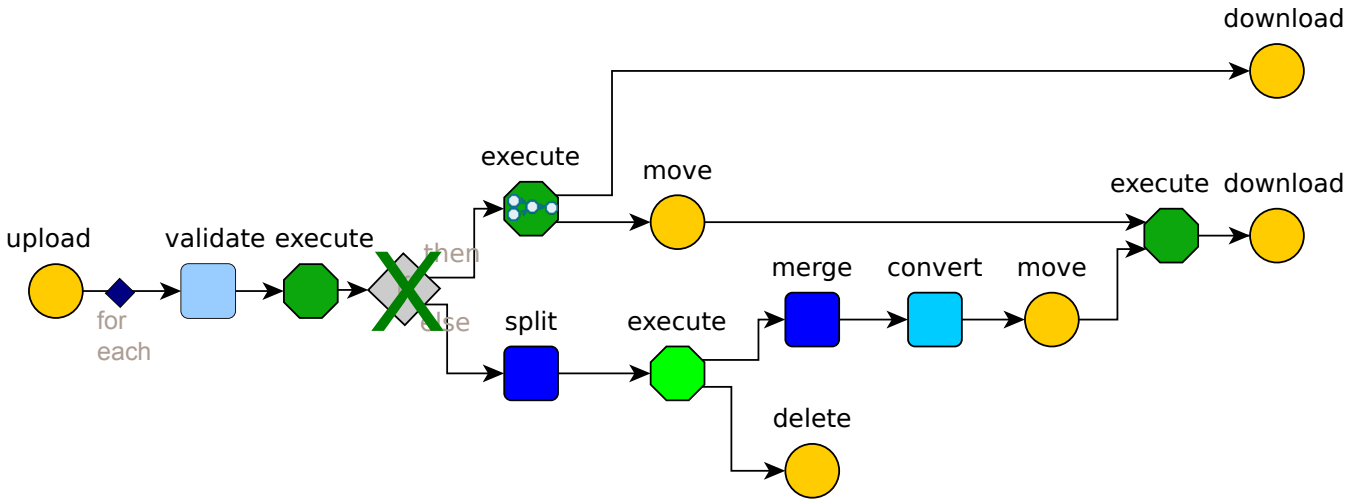
Les deux couches dans le MoC UDFAS :

- la couche DataFlow (A) $\rightarrow$ **orchestration** à partir du flot de jetons DSURI (abstraits, références) ;
- la couche Acteur (B) $\rightarrow$ états internes aux acteurs pour **le contrôle d'appels distants asynchrones (services).**

Plan

Simplifier la conception des workflow

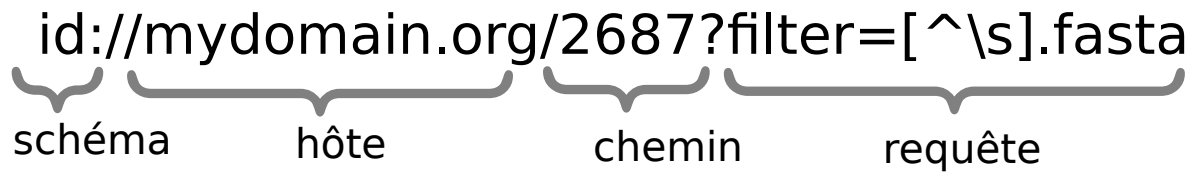
La gestion du routage et du déplacement des données



Simplifier la conception des workflow

La gestion du routage et du déplacement des données

Une classe unique de jetons abstraits : DSURI
(DataSet Uniform Resource Identifier)

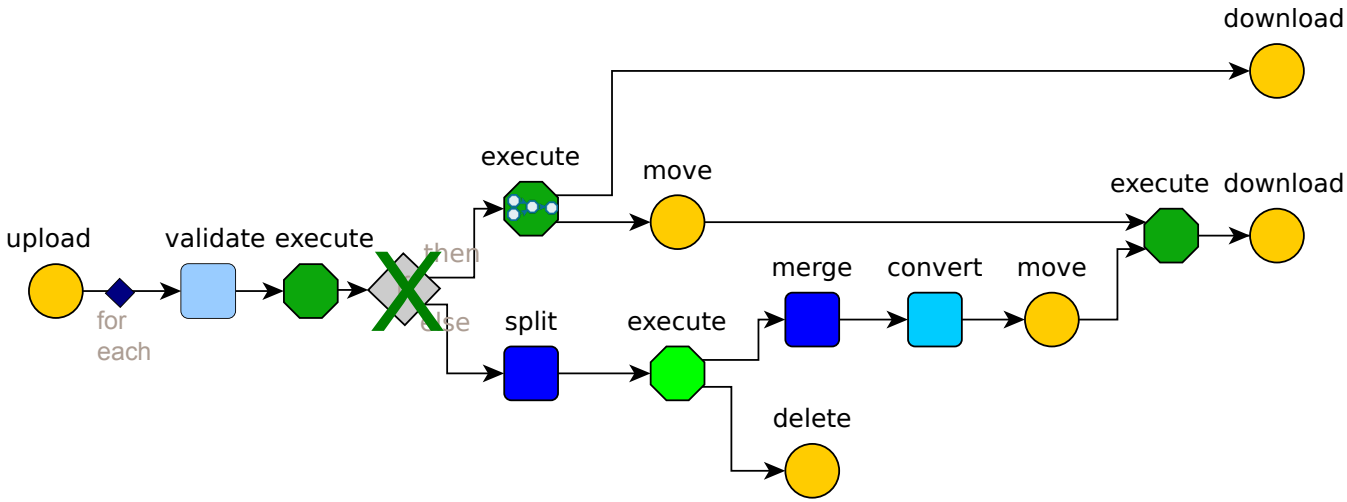


ftp://ftp.ensembl.org/pub/shared/bank/chr8/ref.fa
scp://cluster.bioinfo.org:22/home/user5/hs/seq1.fa
file:///home/user5/hs/seq1.fa

- référence des groupes de fichiers$\rightarrow$ **routage, grands jeux de données**
- protocole et hôte $\rightarrow$ **distribution, localisation**
- requête $\rightarrow$ **extraction de sous-ensembles**.

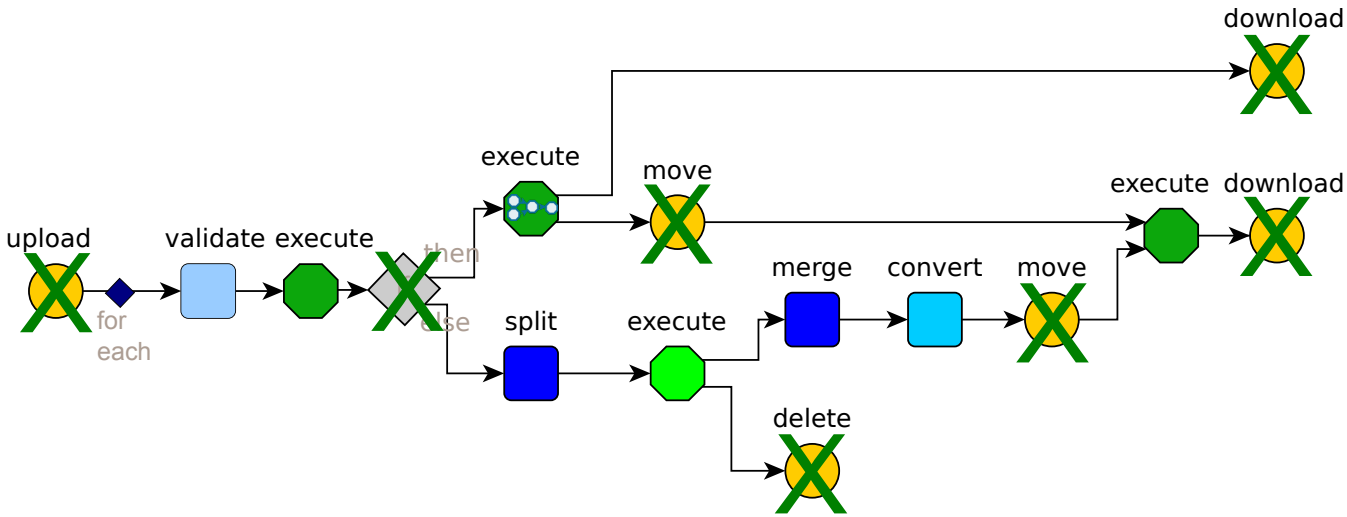
Simplifier la conception des workflow

La gestion du routage et du déplacement des données



Simplifier la conception des workflow

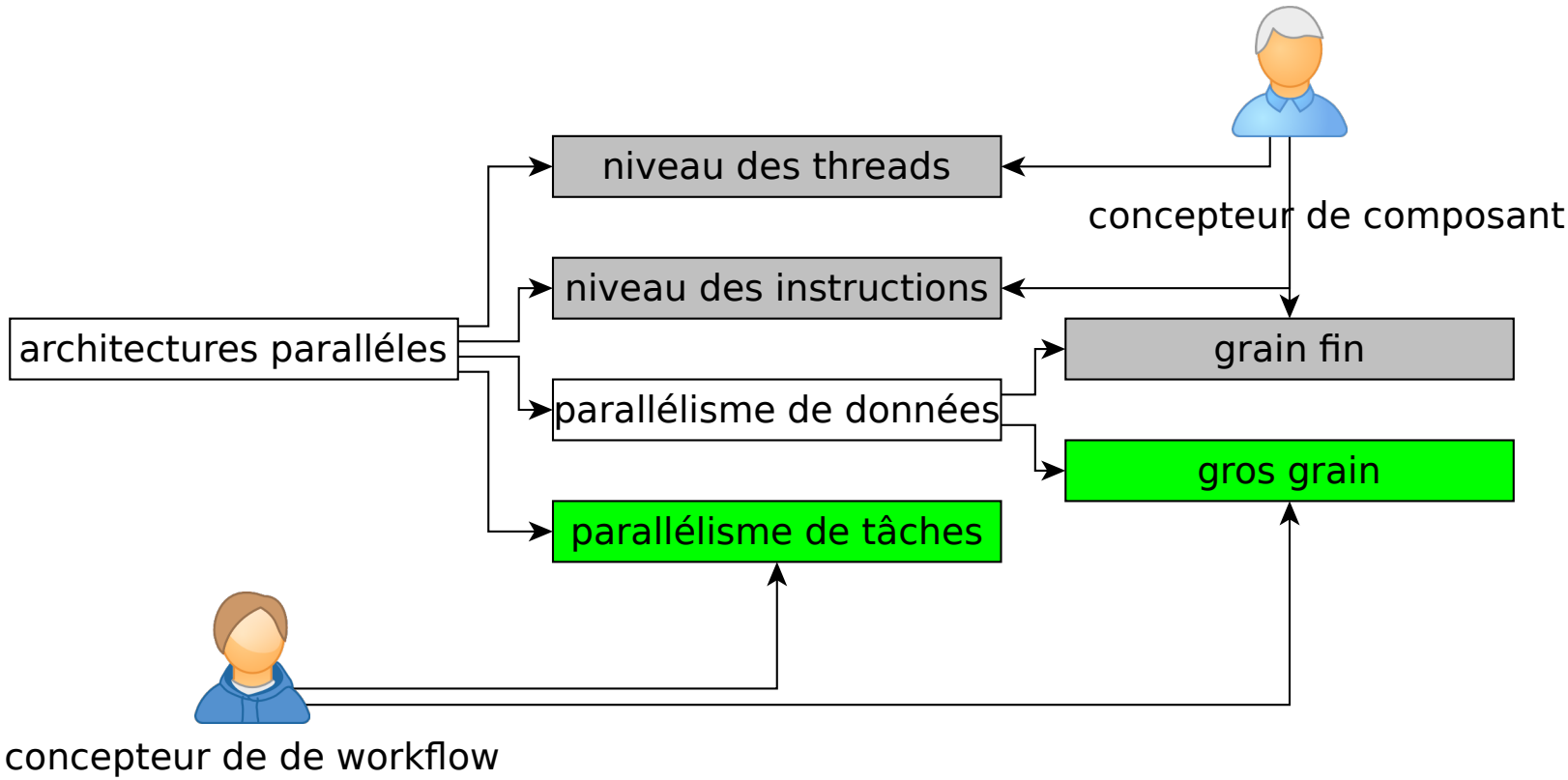
La gestion du routage et du déplacement des données



Simplifier la conception

Parallèlisme

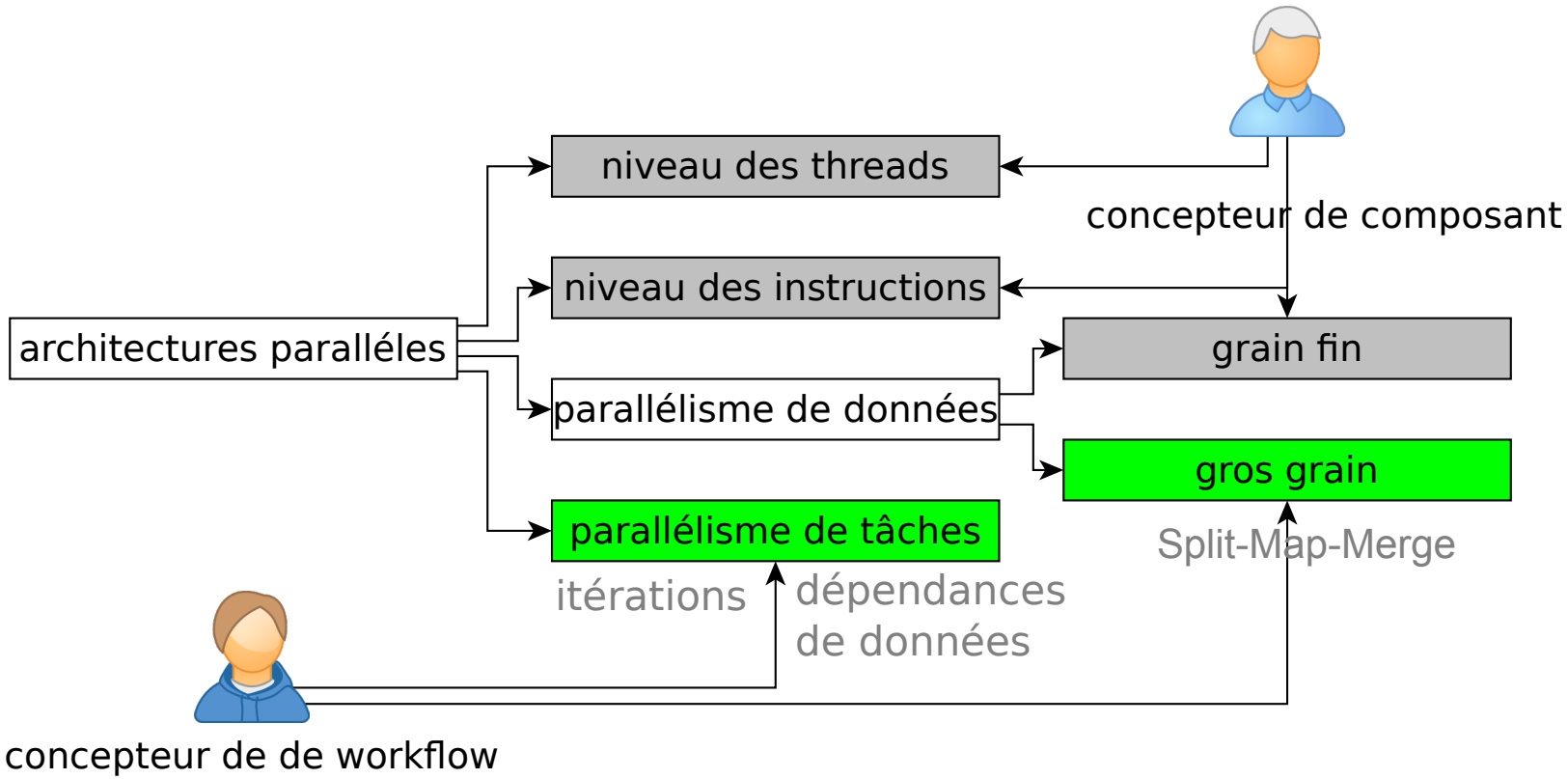
Types de parallelisme accessibles au concepteur de workflow



Simplifier la conception

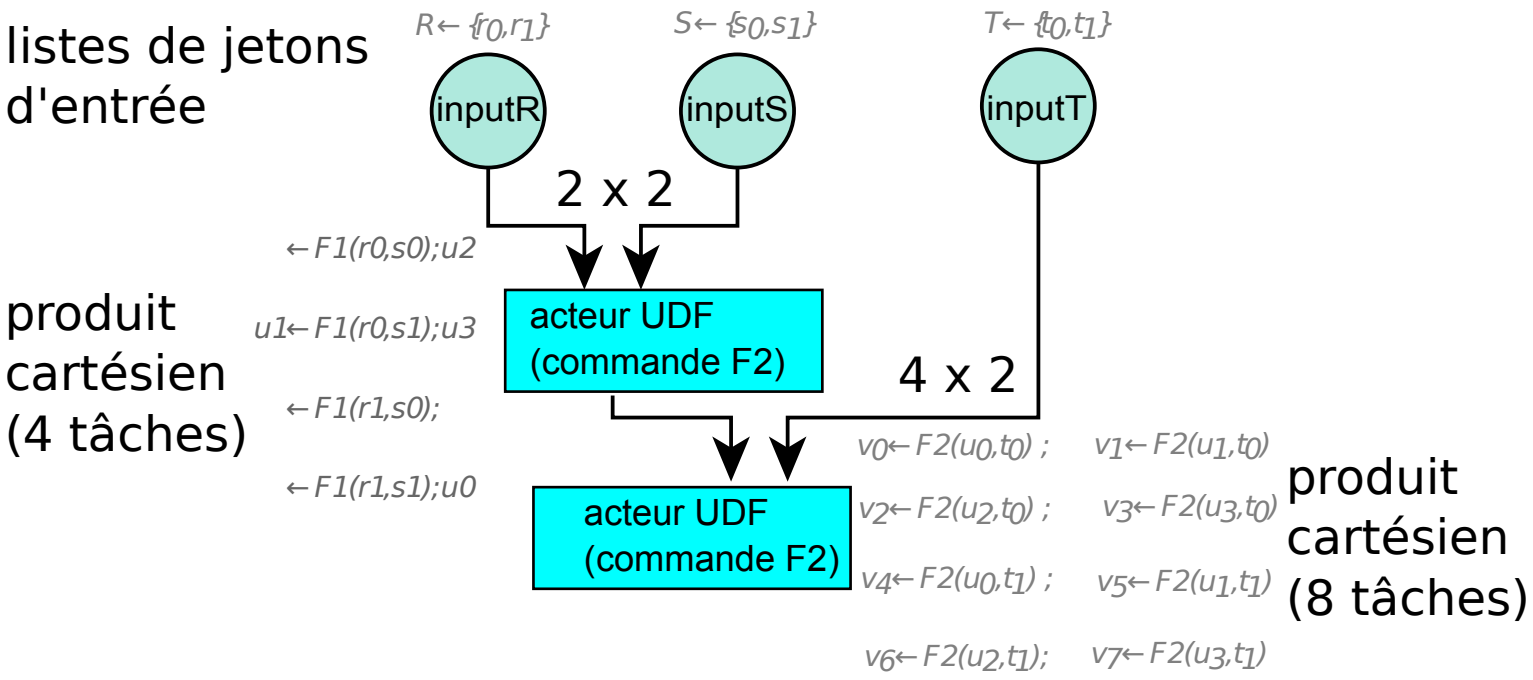
Parallèlisme

Types de parallelisme accessibles au concepteur de workflow



Simplifier la conception

Extraction du parallélisme de tâches : les itérations



Extraction du **parallélisme de tâches issu des itérations**

Simplifier la conception

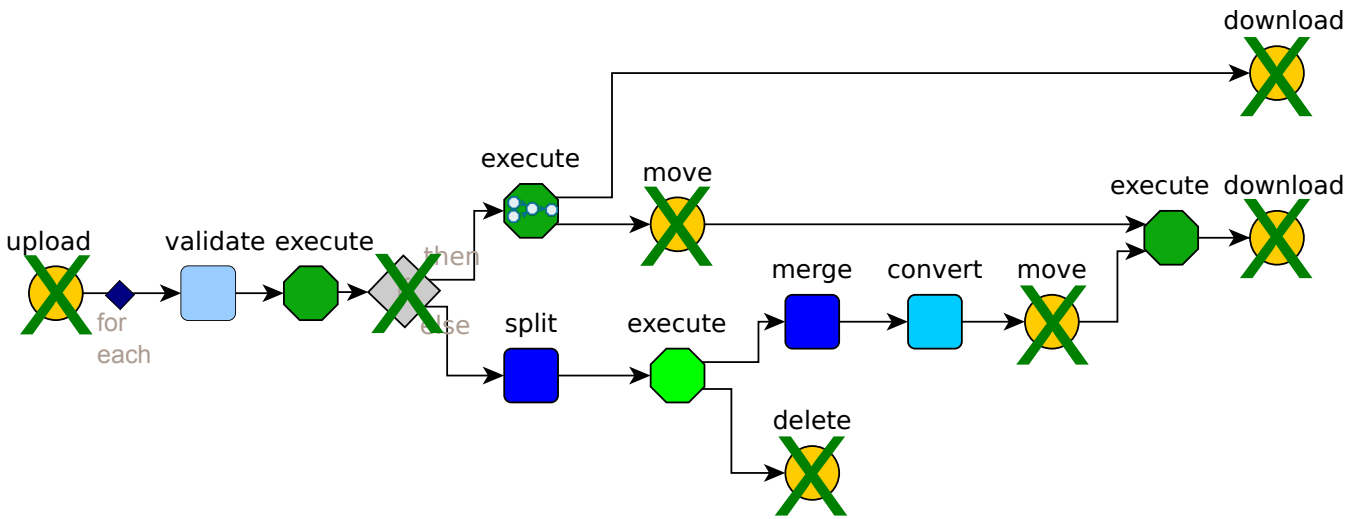
Extraction du parallélisme de tâches : dépendances de données

- **Etats courants** des tâches planifiées $\rightarrow$ stockés dans une structure de données ;
- Pour chaque tâche planifiée $\rightarrow$ lecture des états des **prédécesseurs** $\rightarrow$ définition dynamique de l' **éligibilité** à la soumission ;
- Tâches éligibles$\rightarrow$ **soumission en parallèle** par l'ordonnanceur d'un cluster de calcul, suivant sa stratégie d'ordonnancement.

Analyse **dynamique** des dépendances de données $\rightarrow$ extraction du **parallélisme de tâches issu des dépendances de données**

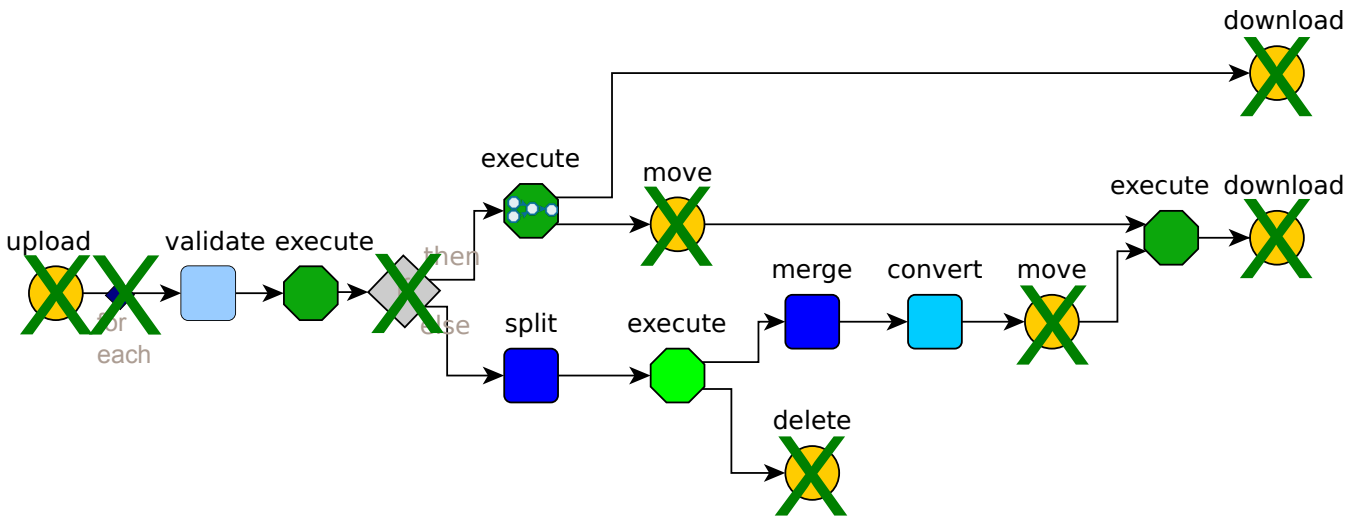
Simplifier la conception

Extraction du parallélisme de tâches : itérations et dépendances de données



Simplifier la conception

Extraction du parallélisme de tâches : itérations et dépendances de données

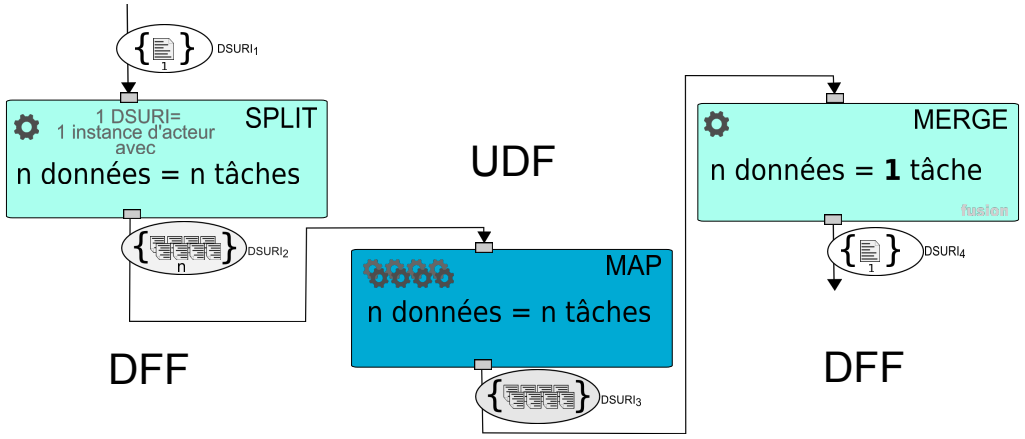


Simplifier la conception des workflows
Intégration du parallèlisme de données gros grain

Gestion du parallèlisme de données gros grain

- Le **parallélisme de données gros grain**
  $\rightarrow$ **parallélisme massif sans modifier les composants** ;

- Parallélisme implémentable suivant un patron conception *Split-Map-Merge* ou *Split-Map-Reduce* $\rightarrow$ facilité par les propriétés des DSURI.



Simplifier la conception des workflows

Intégration des méthodes utilitaires DFF : MDA et ontologies

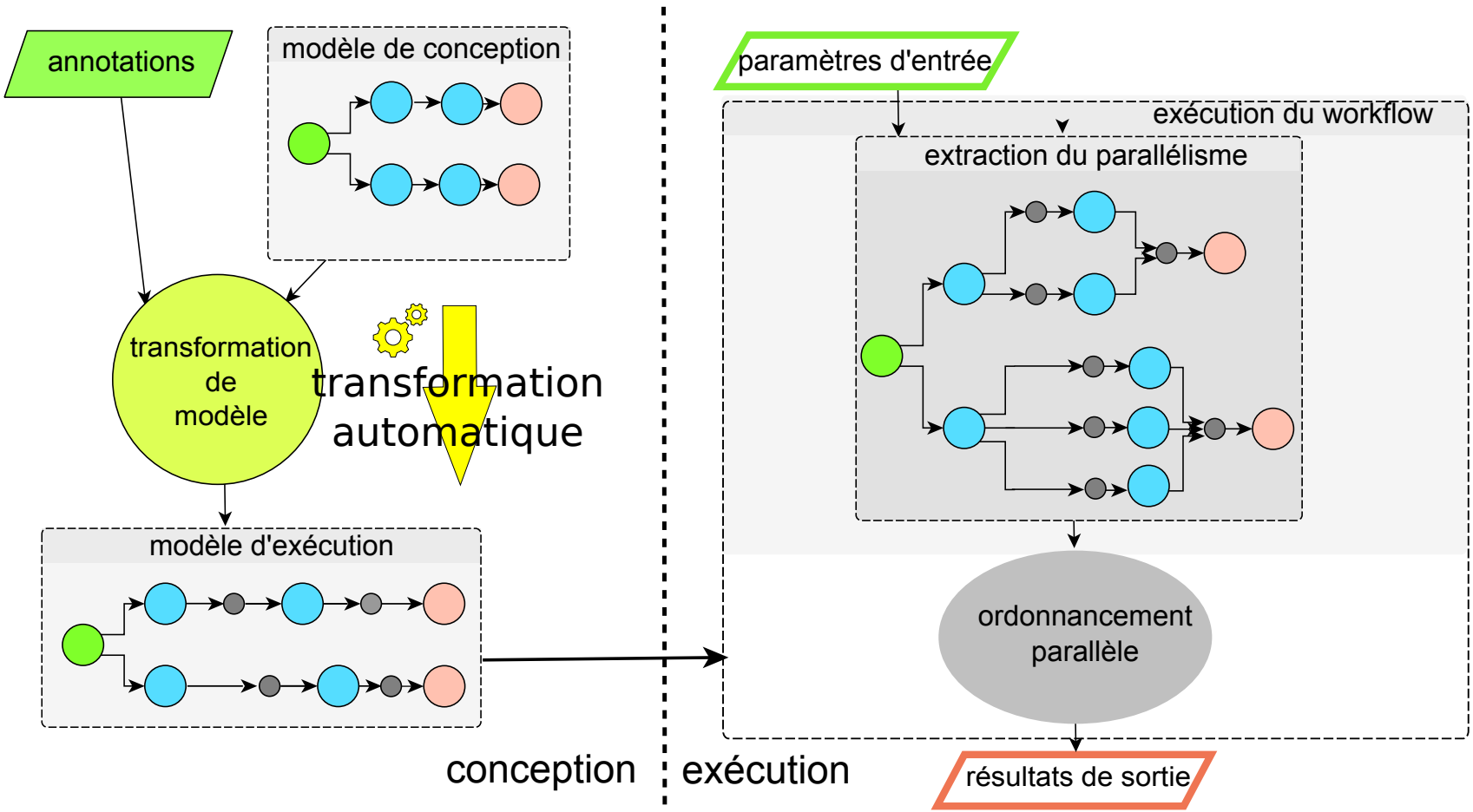
Simplifier la capture des chaînes de traitements d'analyse intensive de données :

- méthodologie de conception
  1. **guidée par l'utilisateur**,
  2. inspirée des architectures dirigées par les modèles ou **MDA** (Model Driven Architecture).
     $\rightarrow$ **raffinage du modèle par étapes successives**, du prototype jusqu'à l'implémentation

Simplifier la conception des workflows

Intégration des méthodes utilitaires DFF : MDA et ontologies

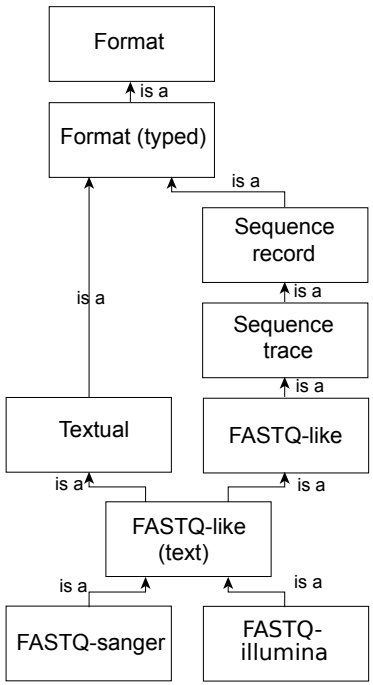
MDA : conception et transformation de modèles de workflow jusqu'à l'exécution.



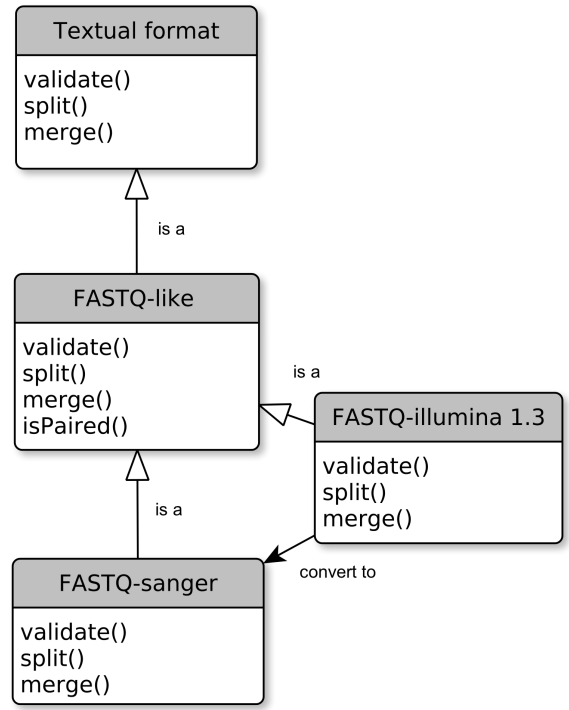
Simplifier la conception des workflows

Intégration des méthodes utilitaires DFF : MDA et ontologies

Exploitation du Méta-Modèle des données pour la parallélisation semi-automatique



Hiérarchie de format dans l'ontologie EDAM (EMBRACE Data And Methods).

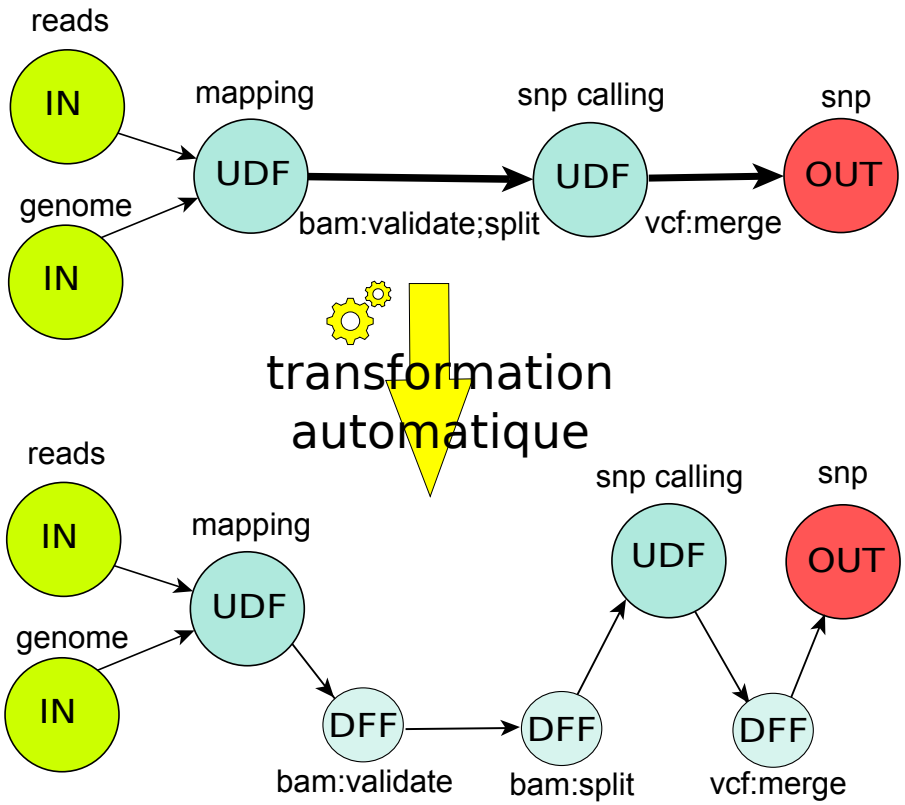


Hiérarchie des formats dérivée (méthodes *Split* et *Merge* sont implémentées pour chaque format).

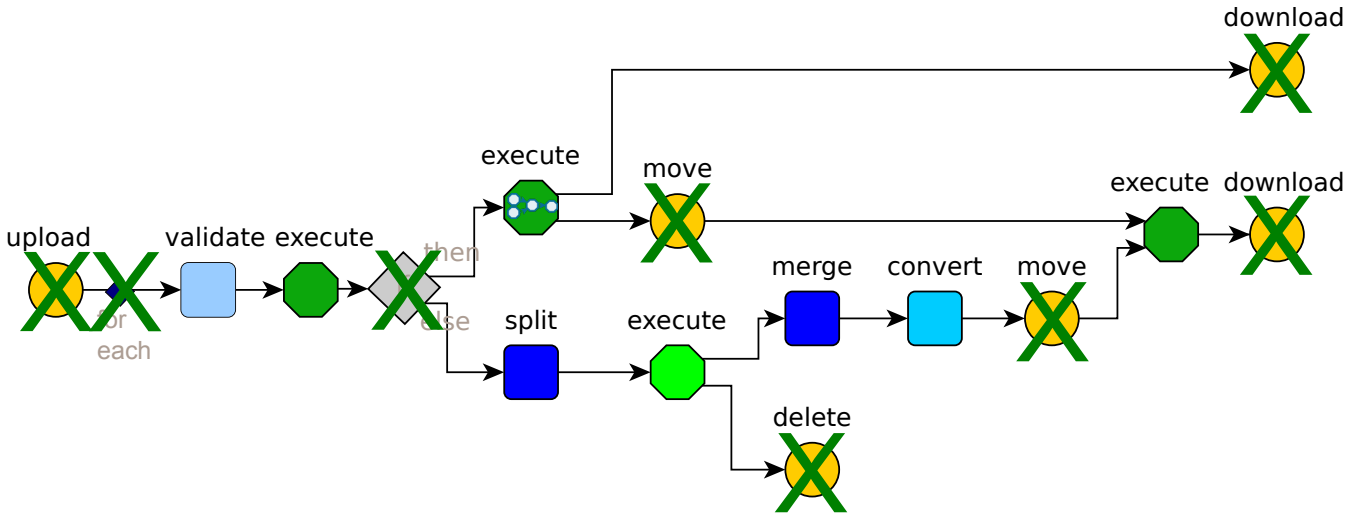
Simplifier la conception des workflows

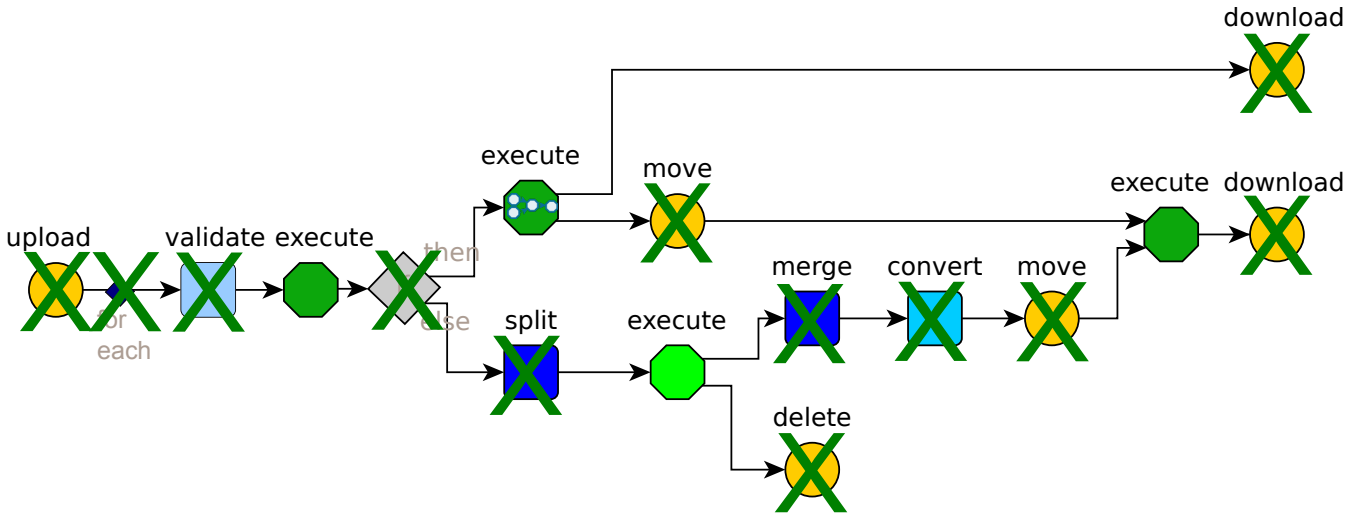
Intégration des méthodes utilitaires DFF : MDA et ontologies

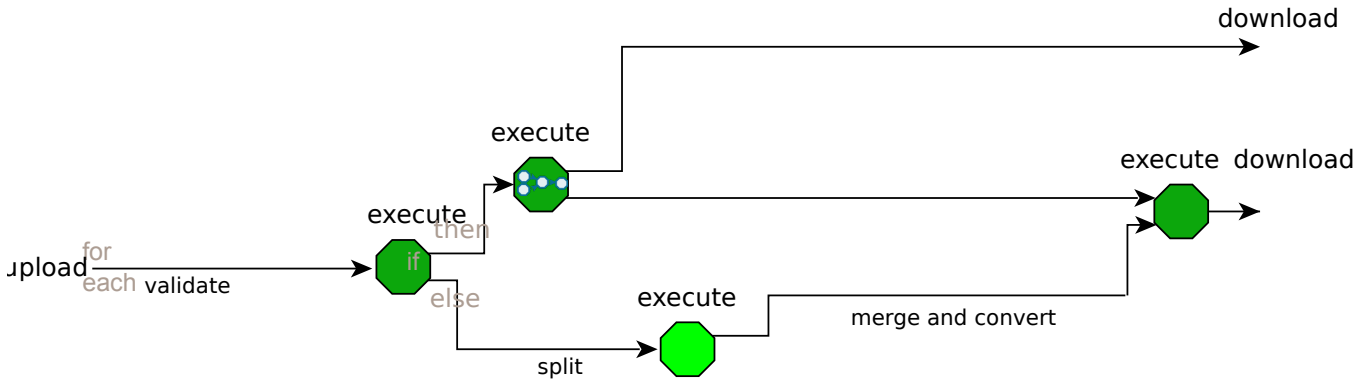
Typologie des acteurs et transformation de modèles



Capture des acteurs UDF + annotations DFF $\rightarrow$ transformation $\rightarrow$ modèle d'exécution.

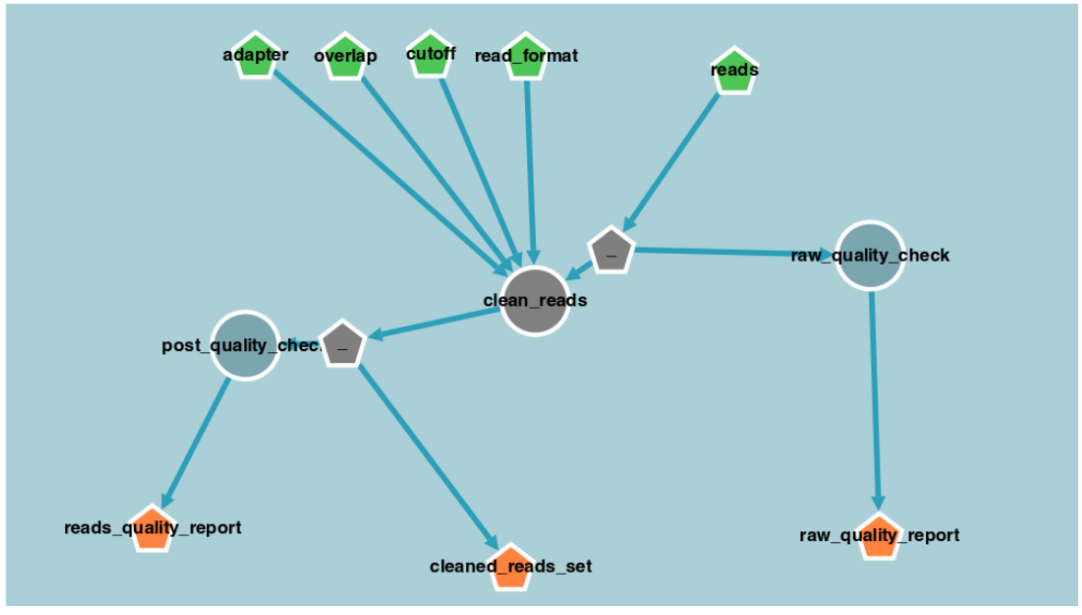

download
execute
move
execute
download
upload
validate
execute
then
merge
convert
move
else
split
execute
delete


download
execute
move
execute
download
upload
validate
execute
then
merge
convert
move
for
each
else
split
execute
delete


download
execute
execute download
execute
then
for
each
validate
if
else
execute
merge and convert
split

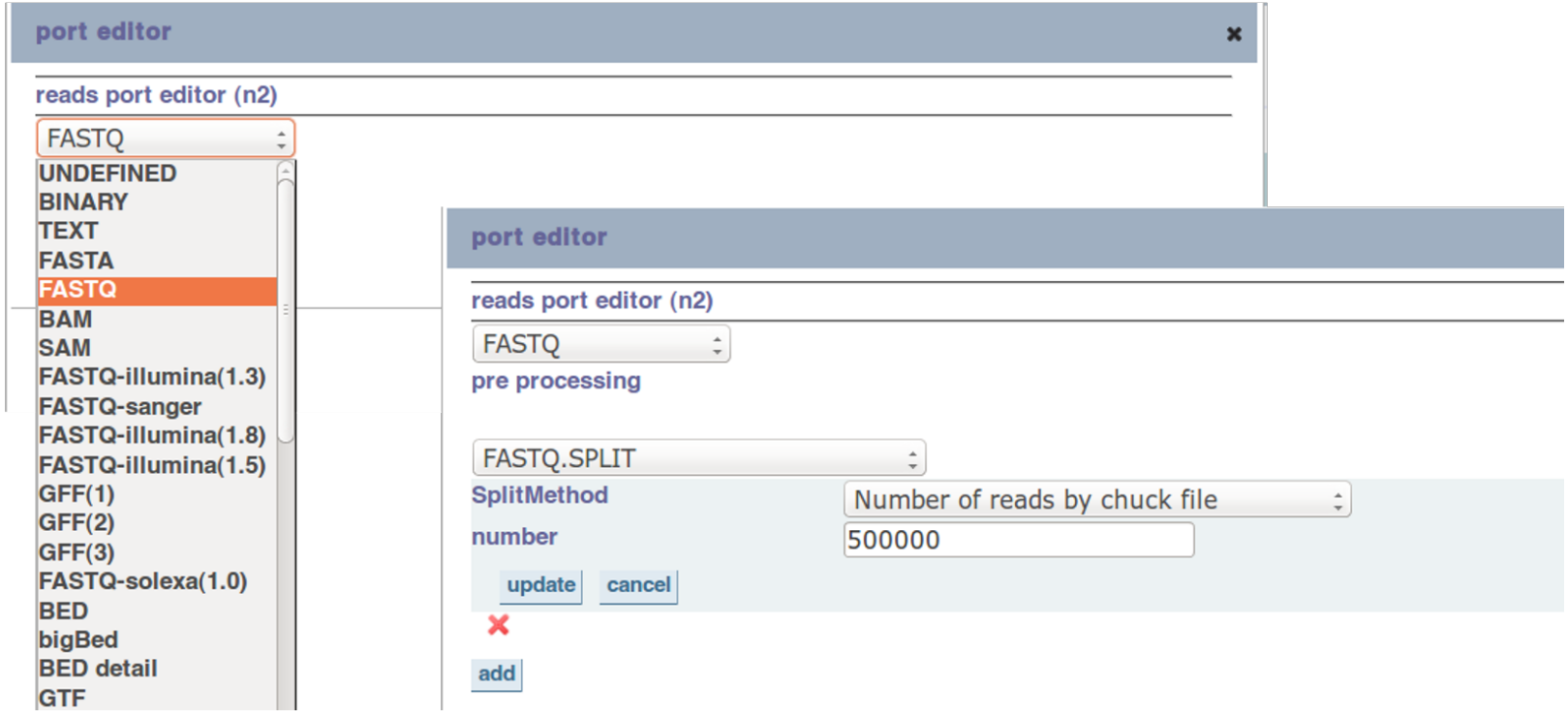
Simplifier la conception des workflows

Implementation : Environnement SaaS de conception graphique et d'exécution de workflow



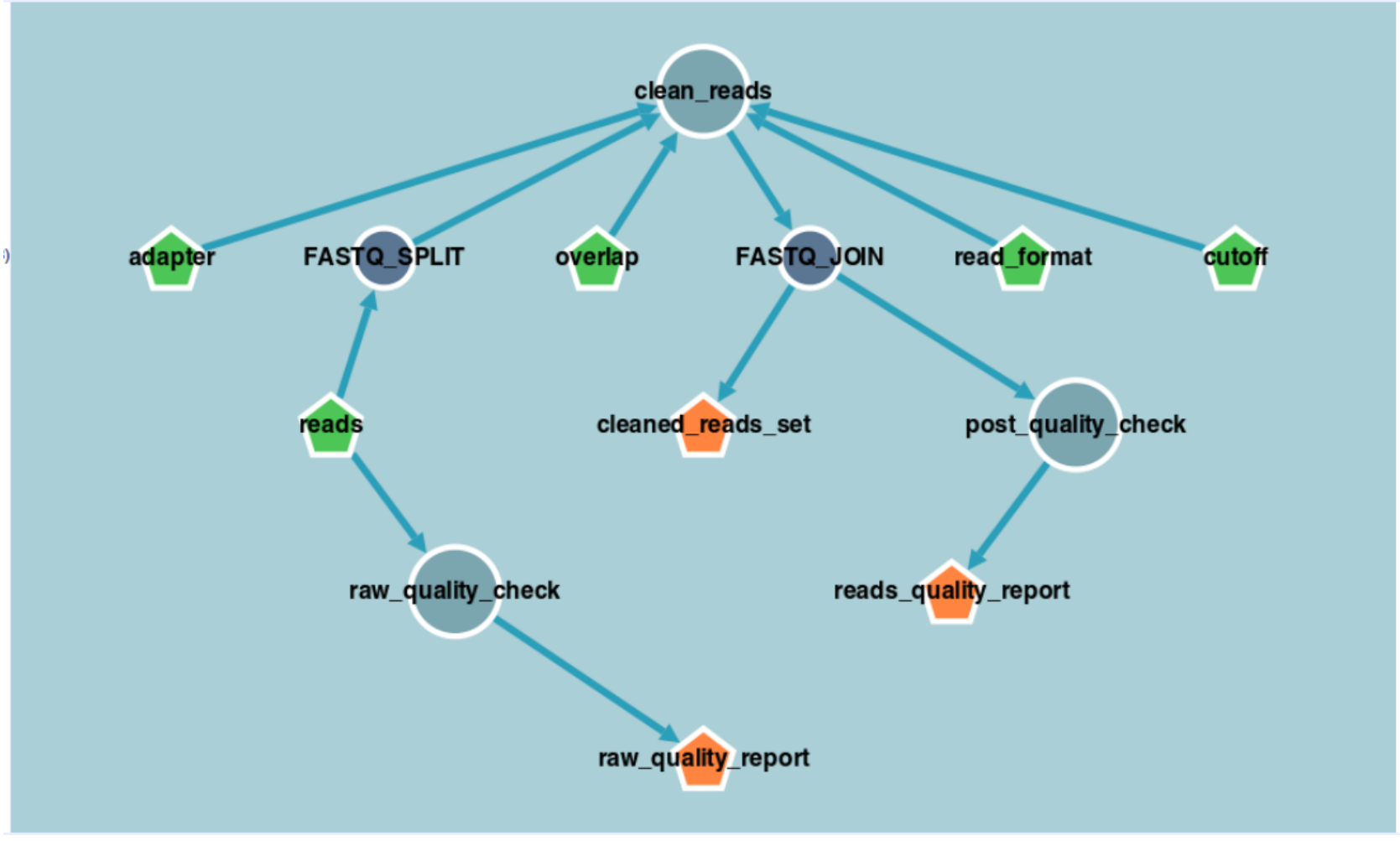
Simplifier la conception des workflows

Implementation : Environnement SaaS de conception graphique et d'exécution de workflow



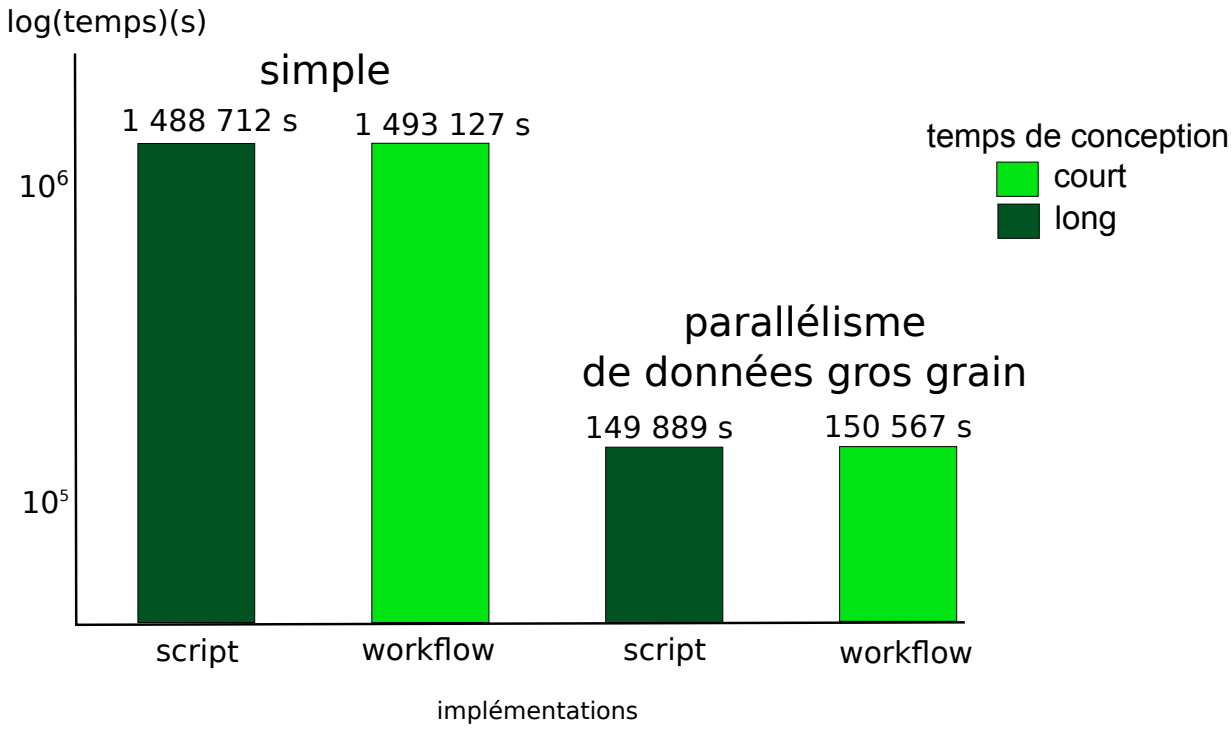
Simplifier la conception des workflows

Implementation : Environnement SaaS de conception graphique et d'exécution de workflow



Simplifier la conception des workflows : Bilan

Implementation : performances



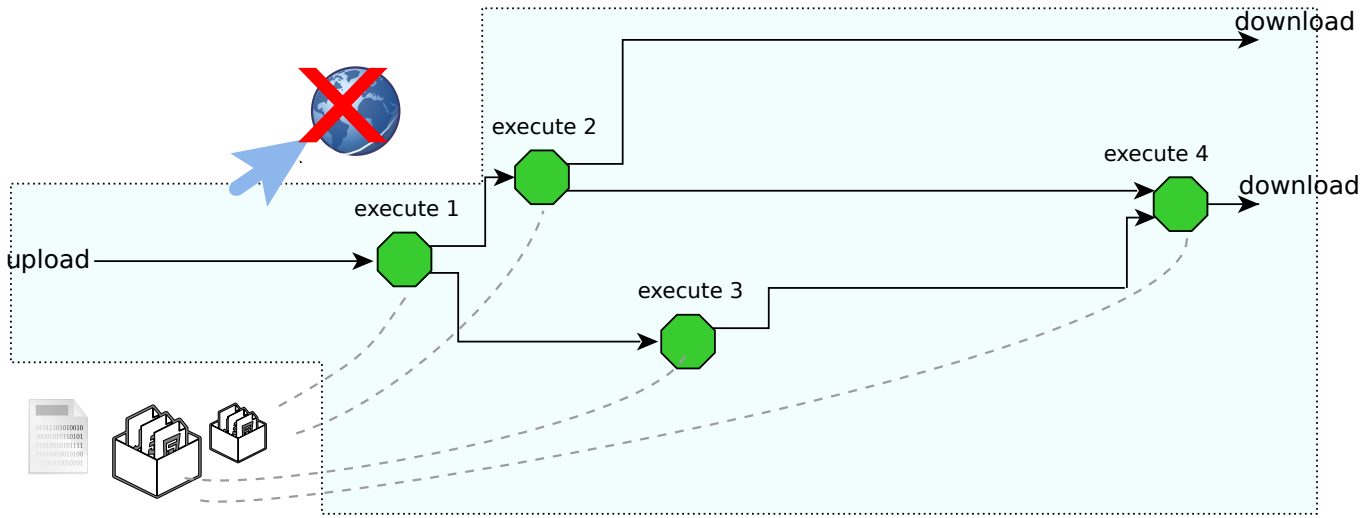
Comparaison des temps d'exécution entre implémentations parallèles et non parallèles d'un même modèle de workflow, avec un jeu de données d'entrée de 10 Go

Plan

Pérénité et échanges des workflows

Déploiement

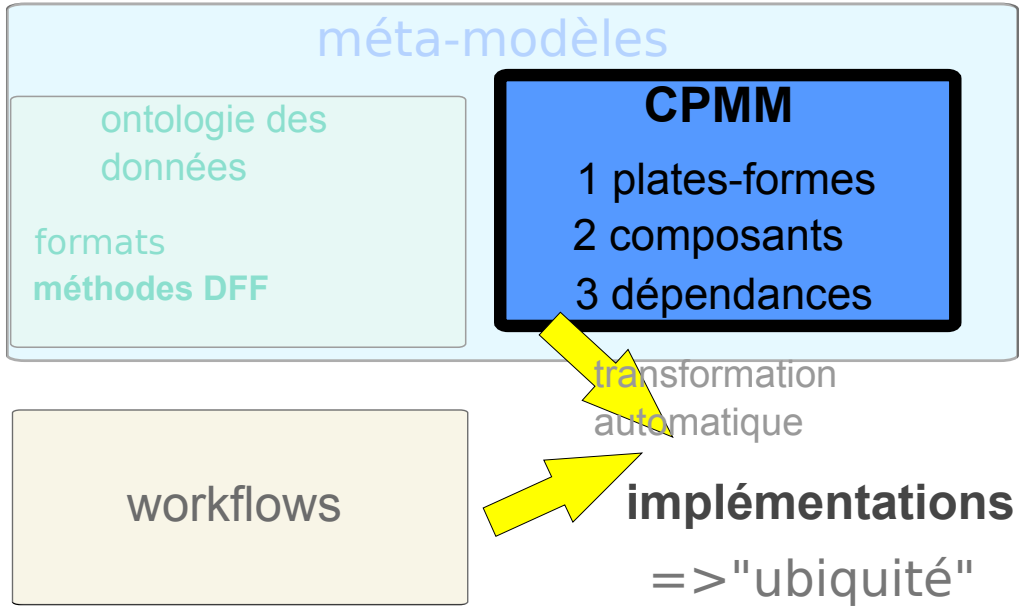
MDA : Adaptation à de multiples environnements d'exécution



Pérénité et échanges des workflows

Déploiement

Méta-Modèle des composants et des plates-formes et adaptation à de multiples environnements d'exécution, **CPMM**
(Components and Platforms Meta-Model)



Pérénité et échanges des workflows
Déploiement

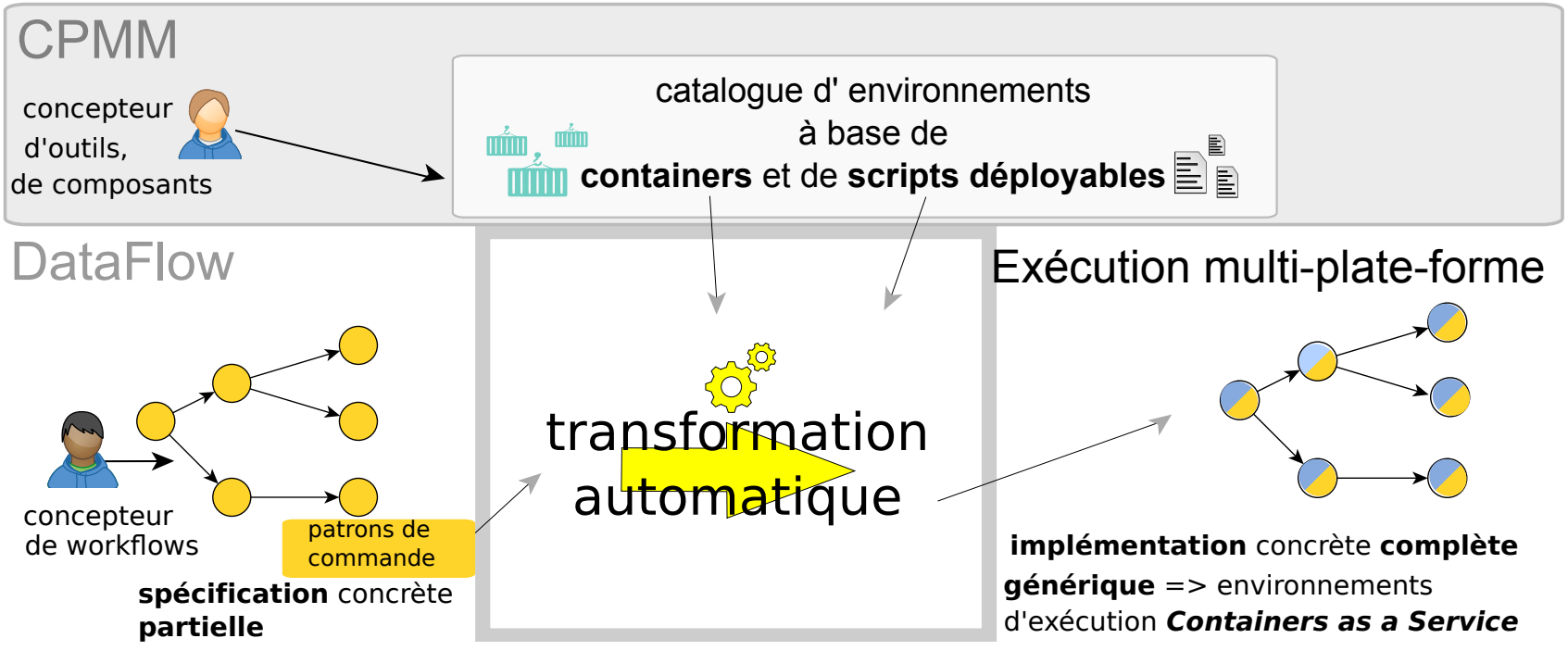
Dépendances de composant générique et plates-formes agnostiques

- **Acteur UDF à code déployable**
  Editer directement le code des composants à partir de l'interface du WfMS.
- **Environnements logiciels à base de containers**
  simplifier la définition rapide des dépendances d'acteur $\rightarrow$ gestion des dépendances basée sur des **containers**(**isolation** ,**encapsulation**) adaptées à la fois au **prototypage** et aux environnements **multi-utilisateurs**.

Pérénité et échanges des workflows

Déploiement

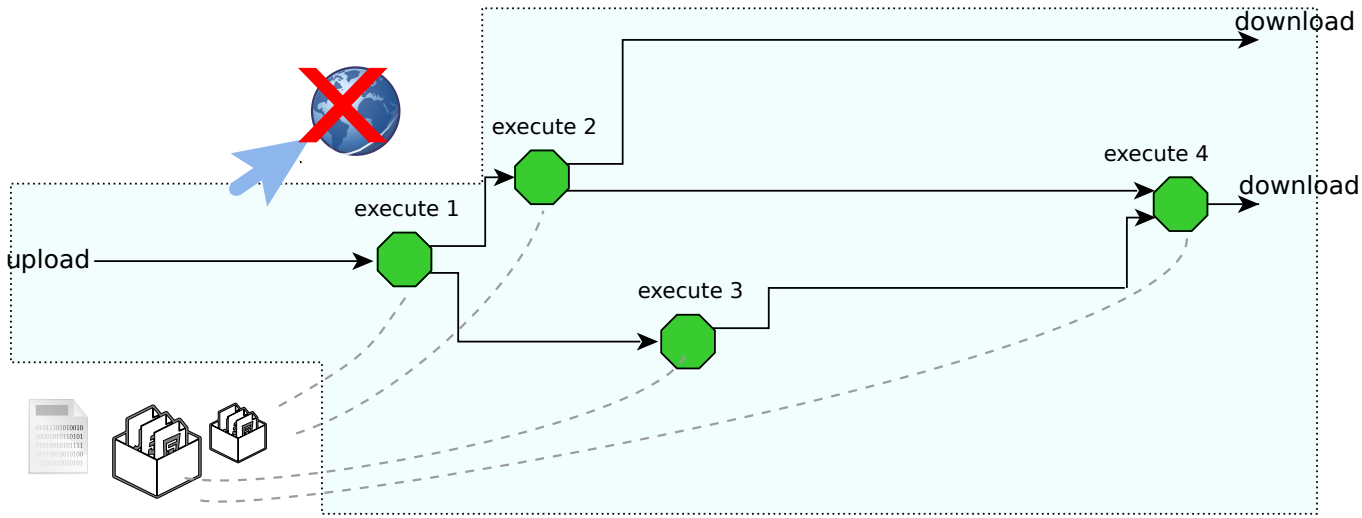
Dépendances de composant générique et plates-formes agnostiques



Pérénité et échanges des workflows

Déploiement

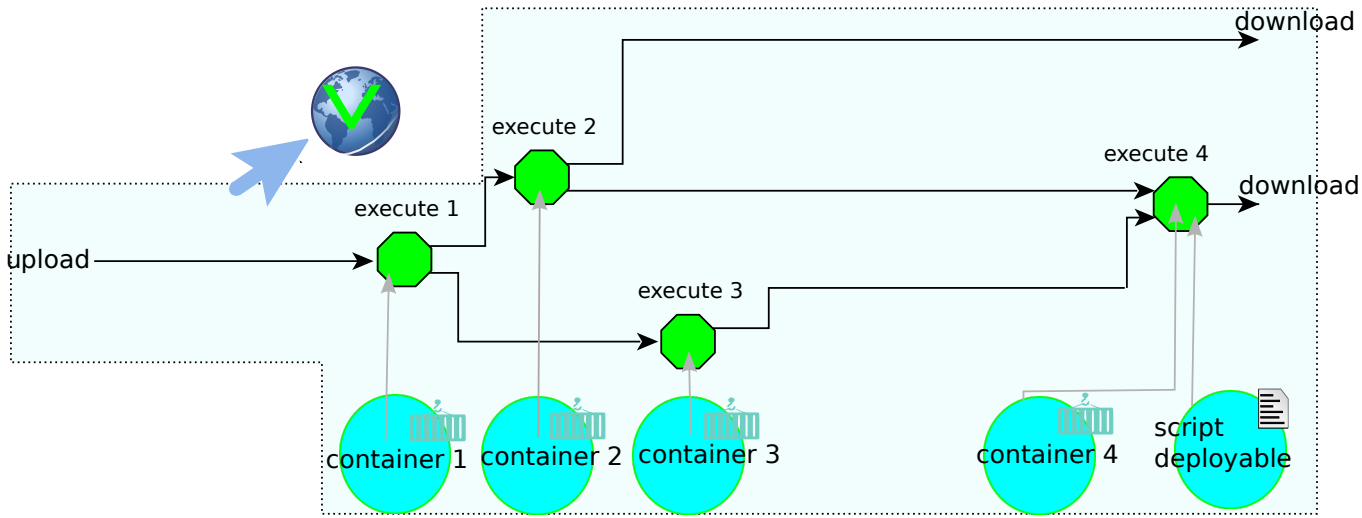
Adaptation à de multiples environnements d'exécution



Pérénité et échanges des workflows

Déploiement

Adaptation à de multiples environnements d'exécution



Spécification DataFlow autonome A-SCDFM (Autonomous Semi-Concrete DataFlow Model).

- Les **URI d'image** de containers et d'URI de jeux de données $\rightarrow$ rapatrier les **dépendances** des acteurs d'exécution, les **scripts** et les **données d'entrée** sur l'environnement d'exécution.
- Spécification **autonome, complete et portable** $\rightarrow$ suffisante pour l'exécution sur **toutes infrastructures CaaS** (Containers as a Service).

Pérénité et échanges des workflows

Echanges

Spécification DataFlow autonome pour l'échange et la reproductibilité (A-SCDFM)

DataFlow + références

issus du méta-modèle des composents et des plate-formes (CPMM)

patron de commande semi-concret

```
<dataflow>[...]
data_uri="http://data.genouest.org/4578?filter=fasta"  [...]
<actor  name="extract_region" nodeId="2"  actor_type="exec">
<commandPattern
  image_uri="http://docker-registry
                .genouest.org/bioperl/latest"
   script_uri="http://data.genouest.org/7895
                  ?filter=extractRegion"
>
  perl extractRegion.pl -i $rawseq
              -o seqout.fasta -l extReg.log
  </commandPattern><inPorts name="rawseq" />
  <outPort name="fasta" filter="*fasta" />
  <outPort name="log" filter="*log"  />
</actor>[...]</dataflow>
```

(données d'entrée + container + script déployable)

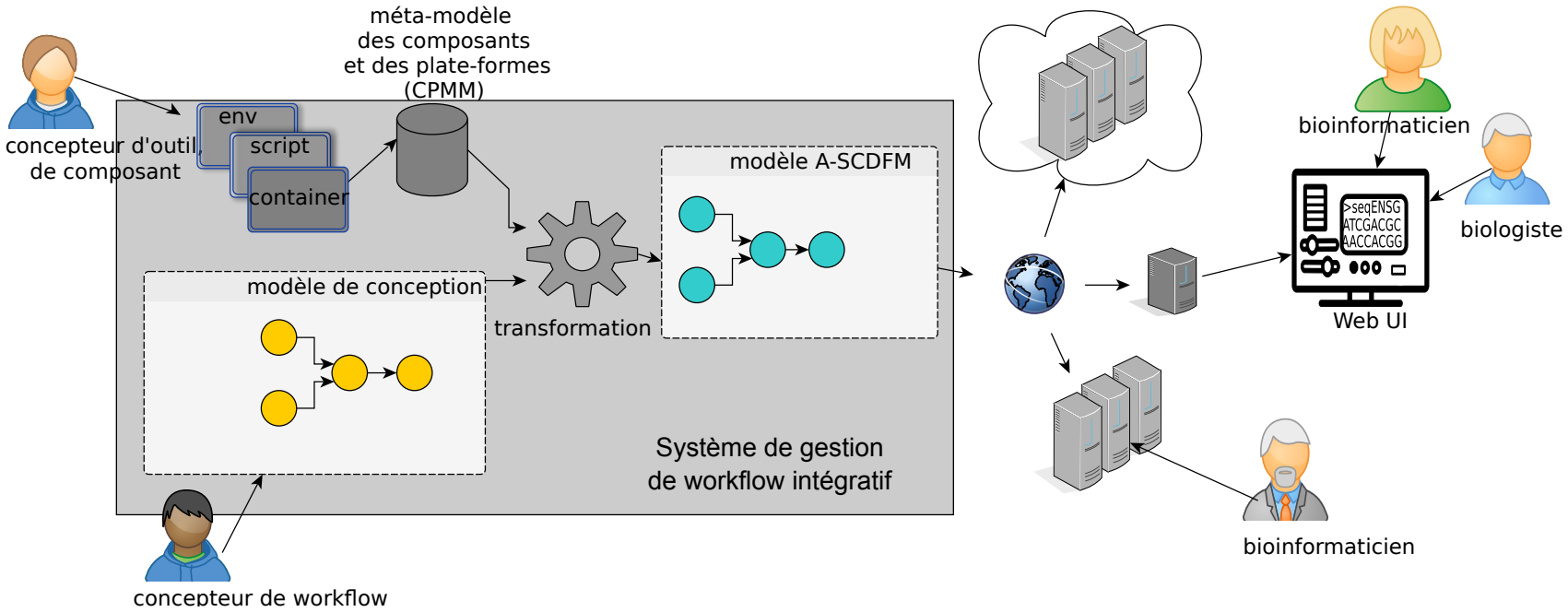
=

spécification portable
adaptée à la diffusion et à l'échange

Pérénité et échanges des workflows

Echanges

Spécification DataFlow autonome pour l'échange et la reproductibilité (A-SCDFM)

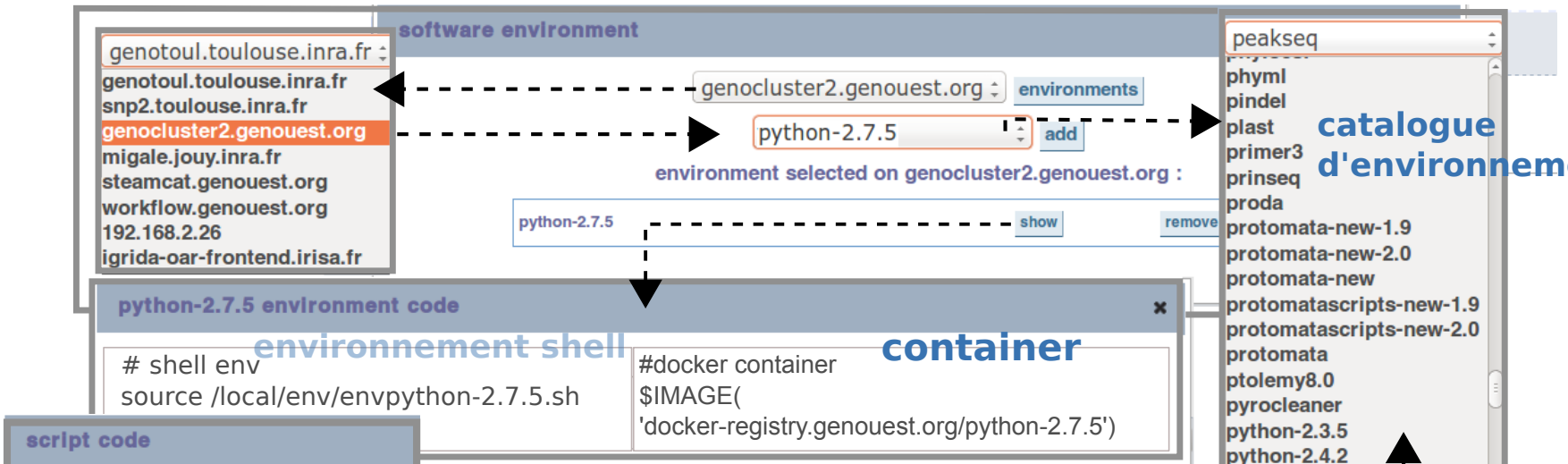


Pérénité et échanges des workflows

Implementation

Intégration du CPMM (variables du shell, script déployables , containers)

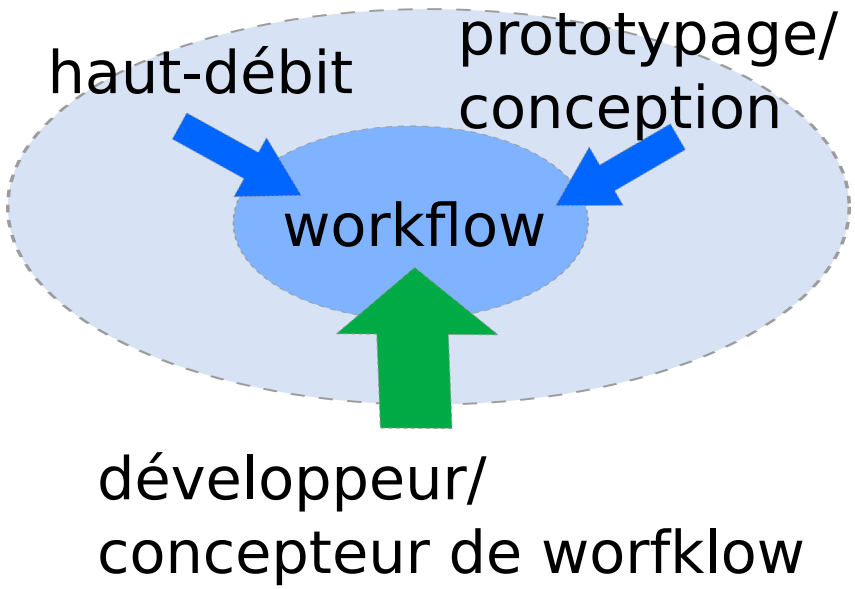
Pour chaque composant :



Plan

Simplifier la conception des workflows

- **un nouveau modèle de calcul UDFAS**
  → cible les niveaux de parallélisme les plus communément extractibles
- **une méthodologie de conception orientée modèle**
  → capture rapide et de **prototypage** des workflows dédié aux concepteurs
  → transformation en implémentation adaptée au haut-débit.



Conclusions et Perspectives

Simplicité de la conception/ intégration du parallélisme

Le **Modèle de calcul UDFAS** .

- la rapidité de l'exécution de notre modèle UDFAS,
- contrôle du **parallélisme interne aux composants**.

Le **Méthodologie de conception** .

- génération d'adaptation DFF (Ba),
- intégration du parallélisme sans annotations

Conclusions et Perspectives

Pérénité et échanges des workflows

- Association de la spécification de workflows avec un **méta-modèle des composants et des plates-formes** (CPMM) $\rightarrow$ génération de **multiples implémentations** ciblant différentes plate-forme.
- **Création d'une spécification de workflows *autonome* (A-SCDFM)**, suffisante pour automatiser le déploiement sur toute plate-forme agnostique CaaS.

Conclusions et Perspectives

Pérénité et échanges des workflows

- Produire A-SCDFM via les systèmes de gestion de workflows intégratifs :
  → offrir à chaque intervenant la possibilité de **contribuer à la capture des workflows et composants**
- Mise en place de **répertoires communautaires de workflows facilement échangeables** basés sur A-SCDFM
  → travailler sur l'**analyse sémantique** et la **découverte de processus métiers** à une échelle suffisamment large pour être pertinente.

→ L'apport d'A-SCDFM à CWL (Common Workflow Language) et à Galaxy doit être envisagé .

Remerciements

- Sandrine Lagarrigue
- Dominique Lavenier
- Aymeric Antoine Lorquin
- Olivier Collin
- Olivier Sallou
- Christophe Blanchet
- Genscale
- Genouest
- IFB-Core
- Dyliss
- Lis