openMINTED

Robert Bossy
Mouhamadou Ba
INRA

OpenMinTeD

Tutorial CORIA/TALN, Rennes, 2018-05-15

Qu’est-ce que c’est

Infrastructure ouverte de *text-mining*

où les chercheurs peuvent découvrir,

créer, partager et réutiliser

des logiciels, des documents et des

ressources pour le TM, TAL, EI, etc.

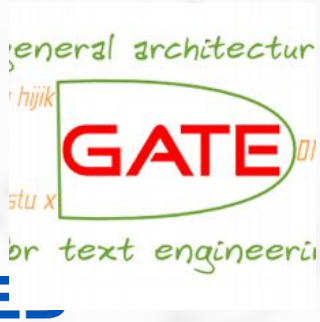
à partir de source scientifiques.

Projet européen E-INFRA

Fin: mai 2018.

U d'Athènes
Athena Research Center
TU Darmstadt
U Sheffield
U Manchester - NaCTeM
Barcelona Supercomputing Center
Polytechnique Lausanne
INRA

- Athena Research Center opère l'infrastructure pendant un an.
- Projet VisaTM: étude d'exploitation par INIST en France.



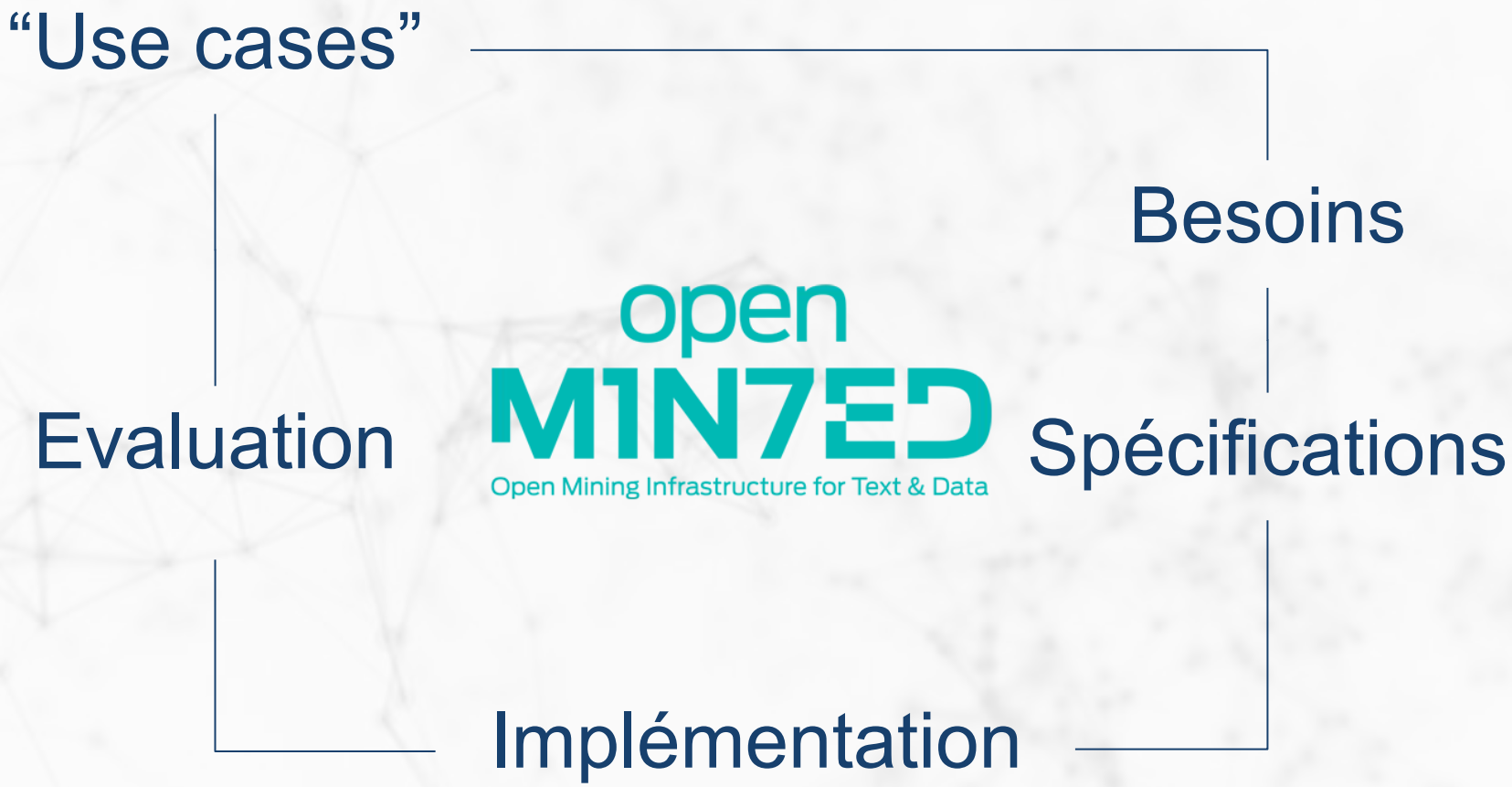
openMINTED

Constat en 2016

Défaillance d'interopérabilité-s

- description hétérogène des outils, des corpus et des ressources
- diversité des environnements d'exécution
- difficulté d'interpréter les licences et leur agrégation

Méthodologie du projet



“Use cases”

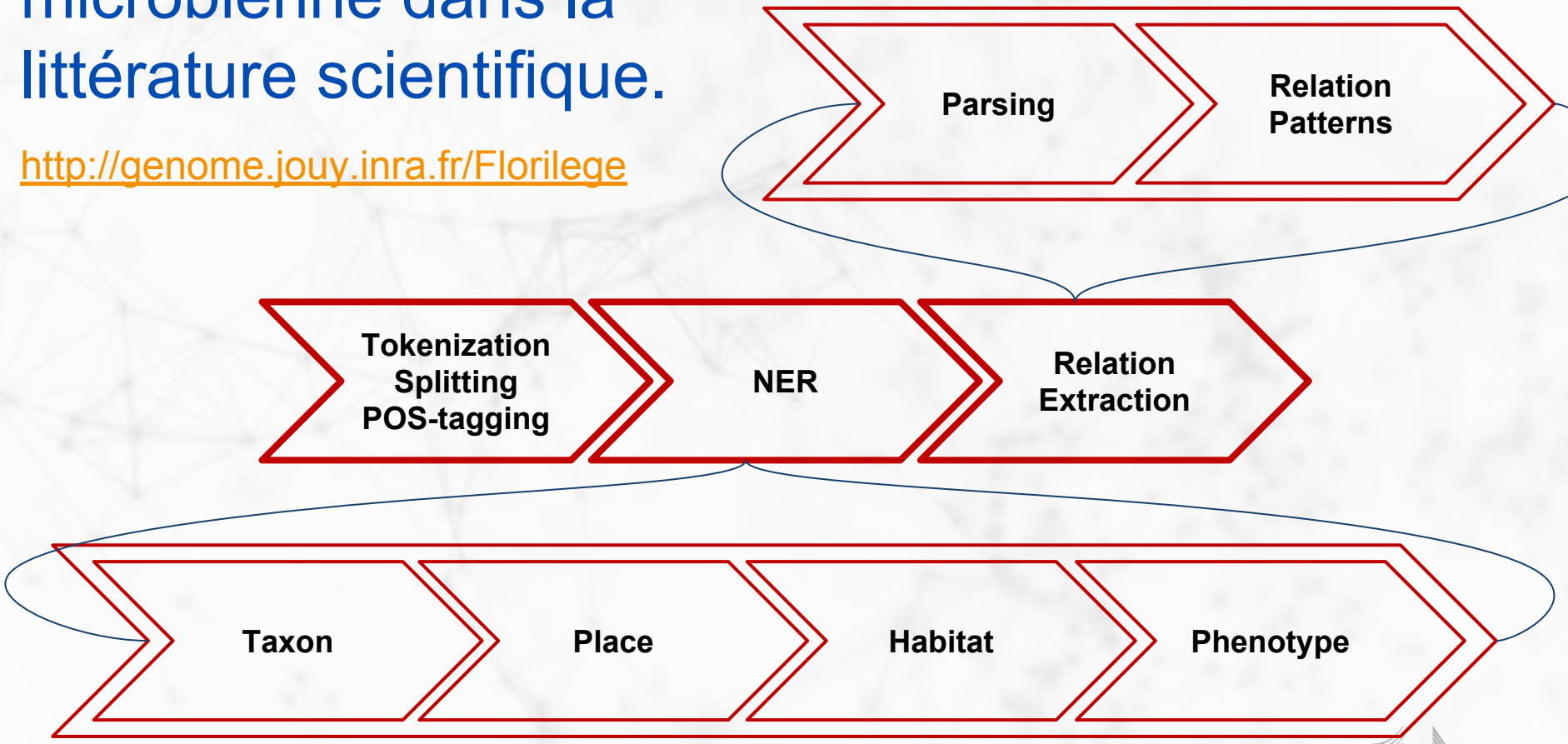
8 applications dans 4 domaines scientifiques:

- Bibliothèque numériques
- Sciences de la vie
- Agriculture & Biodiversité
- Sciences sociales

Exemple : Florilège

Biodiversité microbienne dans la littérature scientifique.

http://genome.jouy.inra.fr/Florilege

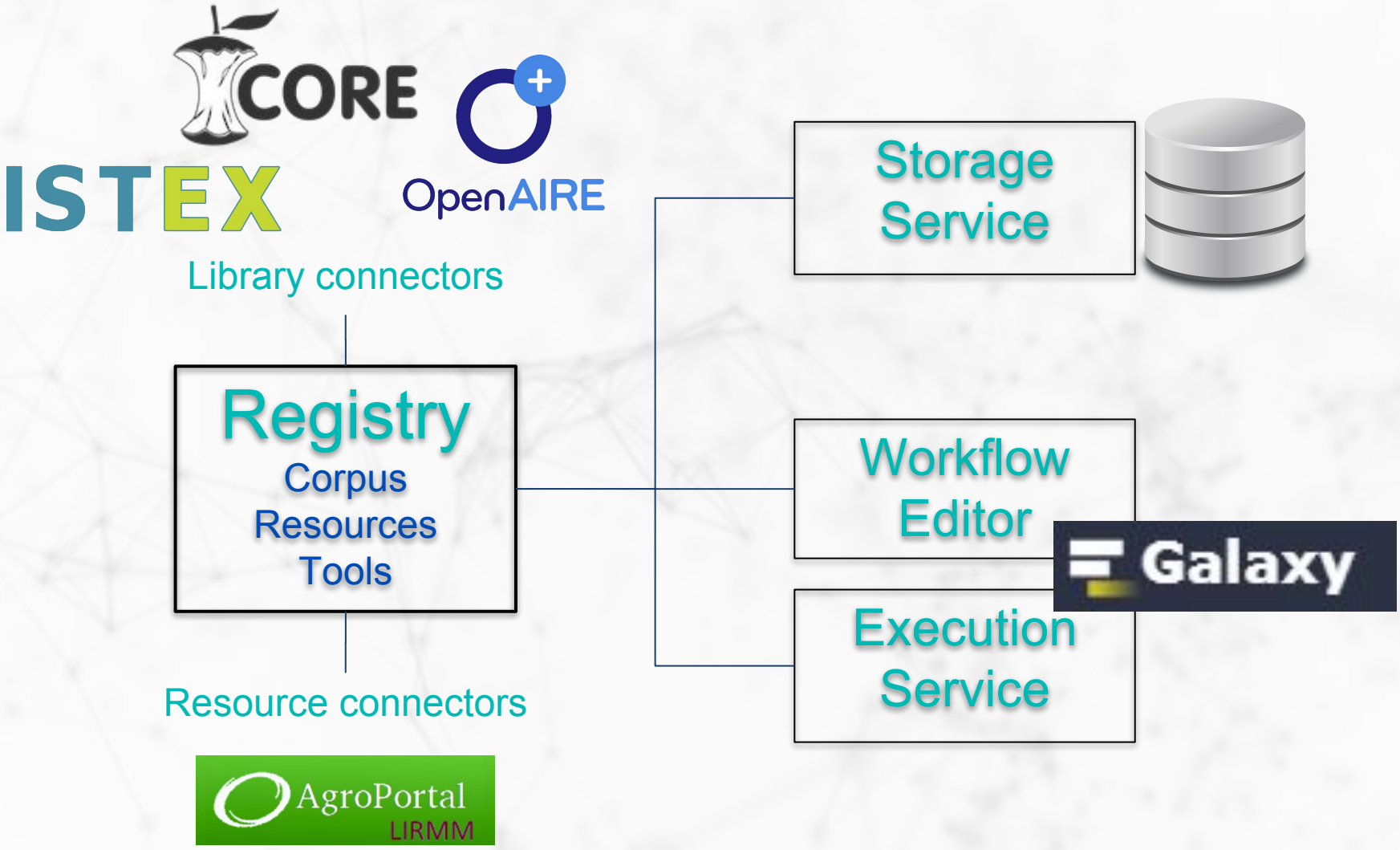


Bénéfices

- Démocratisation des technologies du text-mining, TAL, extraction d'information.
- Valorisation des outils et ressources.
- Environnement de test pour les outils.
- Sécurité juridique.

http://openminted.eu
http://test.openminted.eu

Architecture



OMTD-SHARE

Schéma de description pour toutes les ressources: corpus, documents, composants, ressources (lexiques, modèles, ontologies).

Schéma très complet avec un noyau de propriétés obligatoires.

Saisie par formulaire ou en chargeant un fichier XML conforme au schéma.

Modèle d'exécution

L’exécution est prise en charge par Galaxy.

Les composants sont encapsulés dans une image Docker.

Trois modes possibles:

- **natif**: le fournisseur du composant met lui-même l’image à disposition.
- **web-service**: le composant fait appel à une API externe.
- **maven**: l’image est automatiquement construite à partir d’un POM.

E/S des composants

OpenMinTeD spécifie l'entrée et la sortie des composants.

Documents et annotations:

- sérialisation XMI d'un CAS UIMA
- sérialisation du/des typesystem utilisés

Un format libre est toléré, mais ne garantit pas la possibilité d'enchaîner les composants.

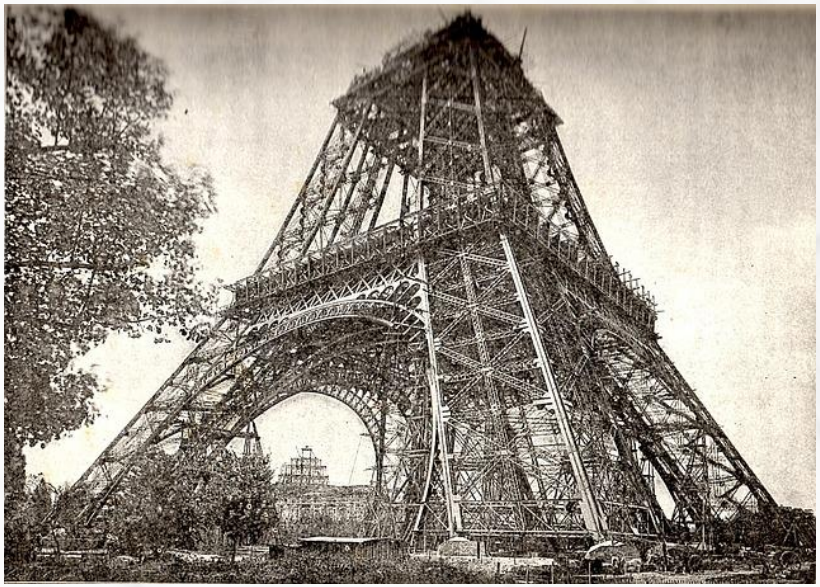
Authentification

OpenMinTeD est une infrastructure dédiée aux chercheurs et utilise la fédération d'identités **eduGAIN**.

La fédération Education-Recherche fait partie de eduGAIN.

https://services.renater.fr/federation/en/index

Perspectives & objectifs



Poursuite du développement.
Accroitre la communauté:

- intermédiaires (infrastructures métier)
- développeurs TM & TAL

Programme de la journée

1. Introduction (10h-11h)
   a. Survol de la plateforme OpenMinTeD
   b. Présentation du schéma OMTD-SHARE
2. Premiers pas avec OpenMinTeD (11h-12h)
   a. Constitution et déclaration d'un corpus
   b. Constitution et déclaration d'une ressource
3. Travailler avec les workflows (14h-15h30)
   a. Exécution d'un workflow-composant
   b. Création d'un workflow
4. Ajouter un composant (16h-18h)
   a. Préparation d'une image Docker
   b. Déclaration d'un composant

openMINTED

Robert Bossy
Mouhamadou Ba
INRA

OpenMinTeD

Metadata

Tutorial CORIA/TALN, Rennes, 2018-05-15

Metadata

contains descriptive, contextual and provenance assertions about the properties of a Digital Object [RDA - DFT Core terms]

Metadata

OpenMinTeD is building an infrastructure based on rich metadata for the resources in the research environment, that support their optimal re-use

The OMTD-SHARE, A Metadata Schema (XSD)

https://openminted.github.io/releases/omtd-share/3.0.2/

Metadata Schema

overview of relevant metadata schemas (e.g. OpenAIRE, CORE, RIOXX guidelines, CrossRef, MetaShare, DataCite, DCAT, CMDI relevant metadata profiles etc.)

FAIR Principles

Findable

F1. (meta)data are assigned a globally unique and eternally persistent identifier.
F2. data are described with rich metadata.
F3. (meta)data are registered or indexed in a searchable resource.
F4. metadata specify the data identifier.

Interoperable

I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
I2. (meta)data use vocabularies that follow FAIR principles.
I3. (meta)data include qualified references to other (meta)data.

Accessible

A1 (meta)data are retrievable by their identifier using a standardized communications protocol.
A1.1 the protocol is open, free, and universally implementable.
A1.2 the protocol allows for an authentication and authorization procedure, where necessary.
A2 metadata are accessible, even when the data are no longer available.

Re-usable

R1. meta(data) have a plurality of accurate and relevant attributes.
R1.1. (meta)data are released with a clear and accessible data usage license.
R1.2. (meta)data are associated with their provenance.

R1.3. (meta)data meet domain-relevant community standards

OMTD-SHARE

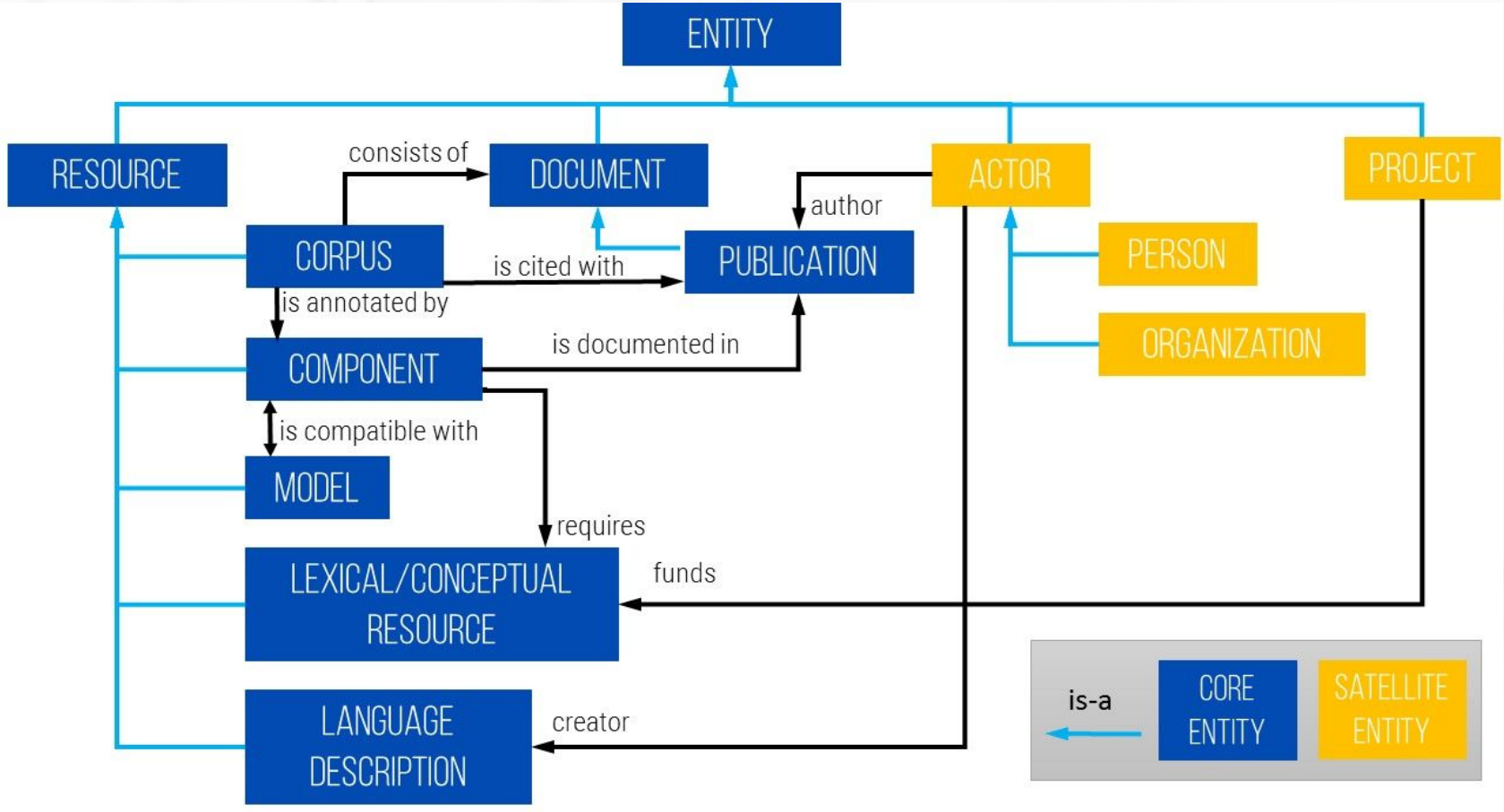
Main Entities

- Applications
- Components
- Corpora
- Annotation Resources
- Models & Grammars

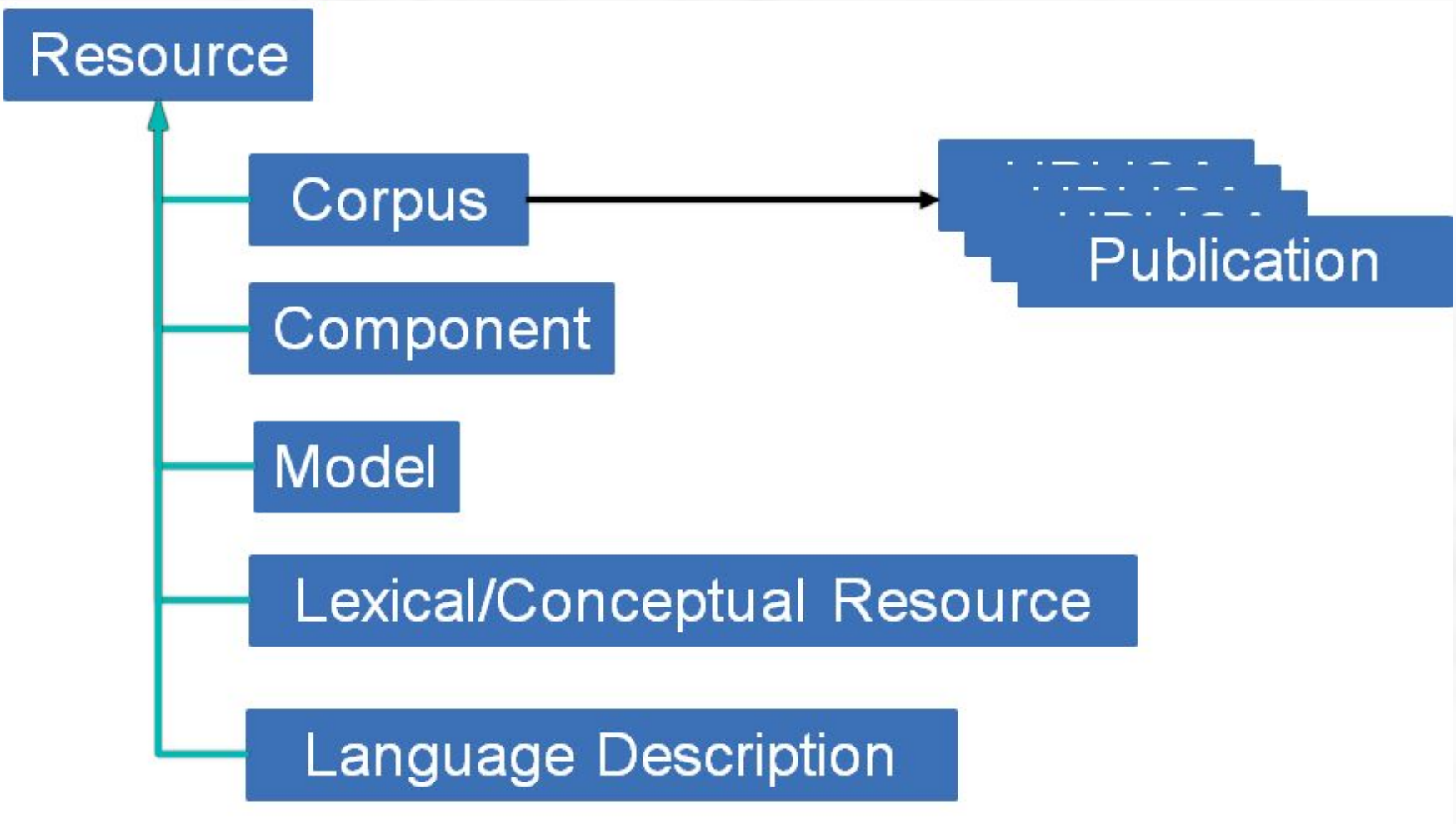
Element & Properties

- Identification
- Distribution
- Rights Info
- Parameters
- Documentation

OMTD-SHARE: Main Entities



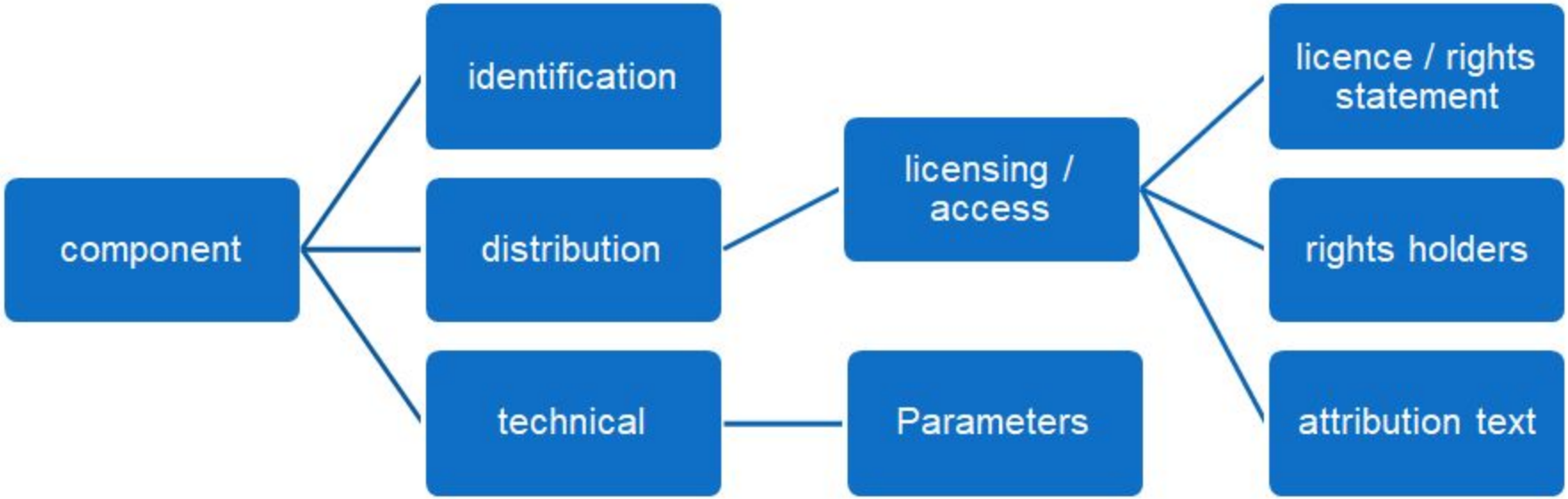
OMTD-SHARE: Entities



OMTD-SHARE: Elements & Properties

- identification & provenance of the metadata record
  - metadata record identifier
  - metadata creation date
- identification of the resource
  - identifiers with identificationScheme (name/URI)
  - title & description (multilingual)
- distribution & licensing/access
  - distribution medium/format (e.g. executable code, downloadable text etc.)
  - licence and/or rightsStatement
  - licence text or URL (provided by system for standard licences)
- contact information
  - either email or landing page
- resource type (& subtype)

OMTD-SHARE: Component



OMTD-SHARE: Type System

OpenMinTeD recommends to use UIMA-based type systems to make it **interoperable** the components/applications

Define formats and type systems for (annotated) corpus

Guidelines are provided

OMTD-SHARE: Ontology

OpenMinTeD has defined an ontology to help classify the resources

Main classes

- annotation type (metadata types added to text/data when processed)
- data format (format type of a document/corpus that can be processed by a software)
- TDM method (method implemented by an algorithm)
- componentType/operation (function of the processing software)
- Domain

openMINTED

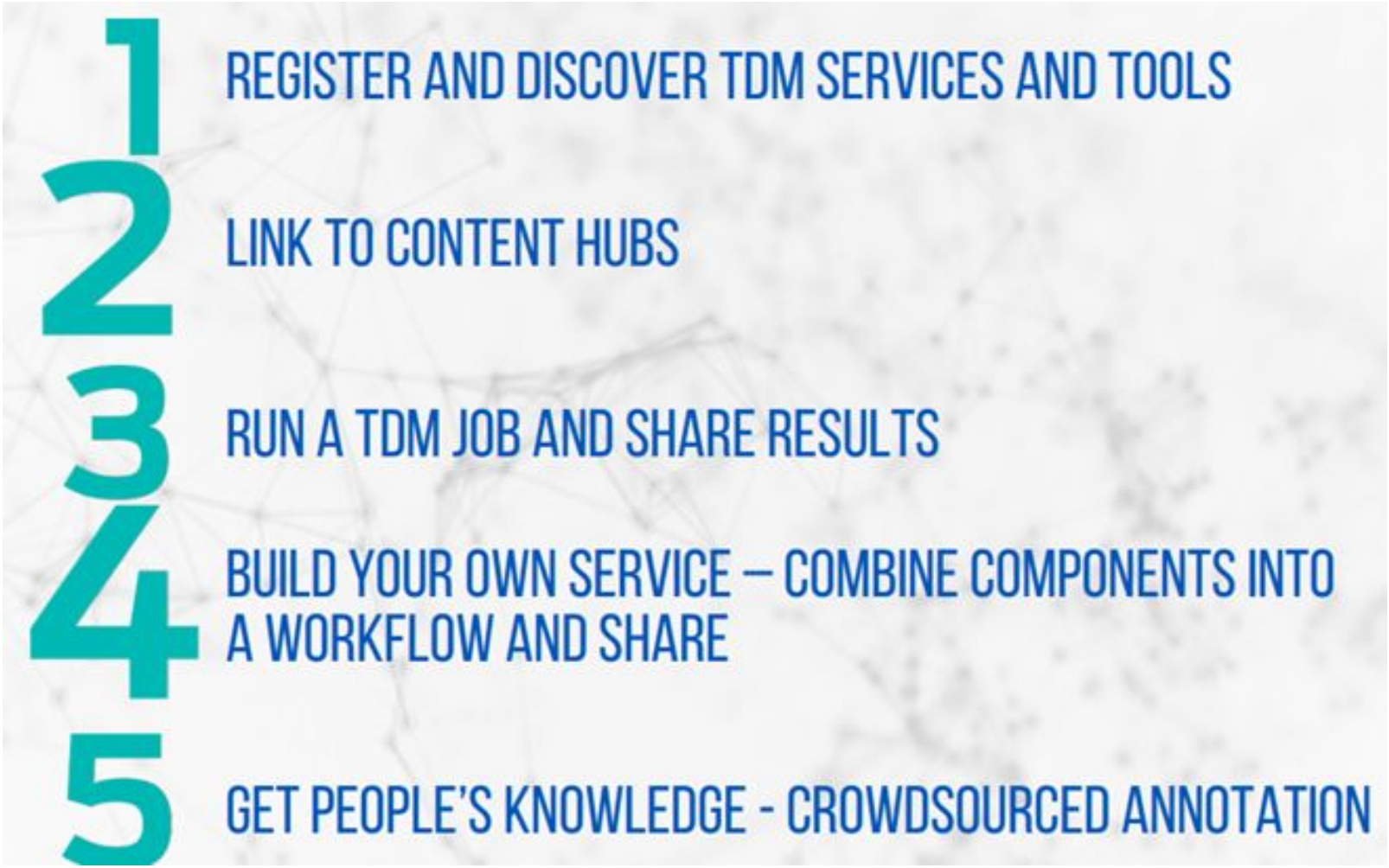
Robert Bossy
Mouhamadou Ba
INRA

OpenMinTeD

Services

Tutorial CORIA/TALN, Rennes, 2018-05-15

OMTD Registry



Search resources

Go to test.openminted.eu

- Facets
- Keywords

Register a corpus (1/2)

Considering a corpus of documents (example of corpora)

1. Create a zip file that contains at least one publication.The zipped file should include a:
   - folder named "**fulltext**", that will contain all publications - mandatory,
   - folder named "**metadata**", that will contain the metadata records of each file - not mandatory
   - folder named "**license**", that contains information about the license of each file - not mandatory
   - folder named "**abstract**", that contains the abstracts of publications (if available) - not mandatory

Register a corpus (2/2)

Considering a corpus of documents (example of corpora)

1. Go to **https://test.openminted.eu**
   - Click **ADD** and choose **Corpora**
   - then, click **Upload your corpus** and click **Go**
2. Fill in the form fields with information on your corpus metadata.
   - the mandatory fields are marked with an asterisk.
   - You are advised to set an Open Access license (help: https://openminted.github.io/releases/license-matrix/)

Register a semantic resource

Considering an ontology (example of ontology)

1. Create a zip file that contains the ontology
2. Go to https://test.openminted.eu
   - Click **ADD** and choose **Annotation Resources**
   - then, click **OMTD Editor** and click **Go**
3. Fill in the form fields with information on your ontology
   - the mandatory fields are marked with an asterisk

Deploy a docker-based component

Step 1 - Preparing and packaging: build an OMTD-compatible Docker image of component

Step 2 - Adding component information in the OMTD platform: provide the metadata records for the component

Create a workflow

1. go to https://test.openminted.eu
   - click **ADD** and choose **Application**
   - click **Build an application with existing components**
   - click **Build**
2. select components from left sidebar
   - choose component omtdImpoter
   - choose component Alvis ToMap
3. Link the components
   - omtdImpoter → ToMap
4. Click **Save** button at top right
   - fill the metadata

Compatible components

1. The docker image must be standalone, self-contained and executable in command line with params

```
$docker run my_component \
 --input <InFolder>   --output <outFolder> \
 --param:paramName=<paramValue>
```

2. The metadata records must fit the image and contain metadata that will be used to run the docker
   - id/distributionLocation
   - executor/input/output/params

openMINTED

Robert Bossy
Mouhamadou Ba
INRA

OpenMinTeD

Introduction à Docker

Tutorial CORIA/TALN, Rennes, 2018-05-15

Docker ?

Docker est un outil qui peut empaqueter une application et ses dépendances dans un environnement isolé (un conteneur), qui pourra être exécuté sur n'importe quel serveur Linux
une technologie de virtualisation, plateforme de virtualisation par conteneur.

- Lancé en 2013
- Développé en Go
- Open source (Licence apache 2)
- Communauté importante et active (Dockercon)
- Fondateurs (Solomon Hykes)

Les conteneurs

Système de virtualisation, utilisant l'isolation comme méthode de cloisonnement au niveau du système d'exploitation. (Wikipédia)

Fonctionnalités Lunix

- Namespaces, Cgroups, overlayFS, chroot

Technologies de container

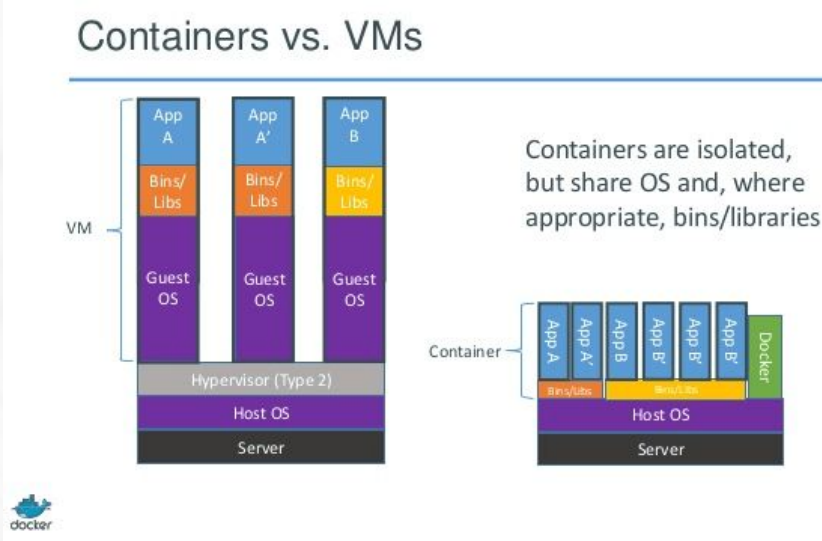
- BSD Jails, Solaris zone, OpenVZ, LXC/LXD, Rkt, systemd-nspawn


Solution: Intermodal Shipping Container
Multiplicity of Goods
A standard container that is loaded with virtually any goods, and stays sealed until it reaches final delivery.
Do I worry about how goods interact (e.g. coffee beans next to spices)
...in between, can be loaded and unloaded, stacked, transported efficiently over long distances, and transferred from one mode of transport to another
Multiplicity of methods for transporting/storing
Can I transport quickly and smoothly (e.g. from boat to train to truck)
docker


Docker is a shipping container system for code
Multiplicity of Stacks
Static website
User DB
Web frontend
Queue
Analytics DB
An engine that enables any payload to be encapsulated as a lightweight, portable, self-sufficient container...
Do services and apps interact appropriately?
...that can be manipulated using standard operations and run consistently on virtually any hardware platform
Multiplicity of hardware environments
Development VM
QA server
Customer Data Center
Public Cloud
Production Cluster
Contributor's laptop
Can I migrate smoothly and quickly
docker

Les conteneurs par rapport aux VMs

plus légers
plus rapides



mais !

- ❏ n'isole pas aussi bien qu'une VM
- ❏ s'installe directement que sur GNU/lunix
  Mac/Windows : boot2docker (VM de ~25M)

docker

facilite le travail des développeurs et administrateurs

- déployer plus souvent
- améliorer la productivité
- migrer facilement des applications
- standardiser la gestion des applications
- Isolation des processus

dans le monde de la recherche… offre des conditions pour la reproductibilité

- Partage de recettes / Partage d'images de *containers*

Utilisation et solutions similaires

intégré aux plateformes des principaux fournisseurs de Cloud, comme AWS, Google Cloud Platform et Microsoft Azure

Plateformes scientifiques :

- Openmined (openminted.eu)
- dockstore (https://dockstore.org/)
- BioShaDock(https://docker-ui.genouest.org/app/#/
- Bioboxes (http://bioboxes.org)
- BioDocker (http://biocontainers.pro)

Utilisation et solutions similaires

Autres technos concurrentes

- Singularity (HPC, Security)
- Shifter
- rkt

Docker en pratique

Un ensemble de concepts ou objets

- Les images
- Les conteneurs
- Le Dockerfiles
- ...

Les images docker

Une image est l'artefact d'un conteneur, l'instance d'une image est ce qu'on appelle un conteneur

https://hub.docker.com/explore/

L'image

- est en lecture seul, ne peut pas être modifiée
- peut s'instancier en plusieurs containers qui tourneront en parallèle
- en général créé à partir d'une recette, un **dockerfile**

Les conteneurs docker

Un conteneur c’est l'environnement isolé faisant qu’une application tourne, on l’obtient en exécuter une image

$docker ps -a

un conteneur

- se déploie assez rapidement, utilise le noyau de la machine hôte
- peut coexister avec plusieurs autres et partager les ressources
- fonctionne de la même manière à toutes les étapes d’un cycle de développement d’un logiciel

Les dockerfiles

Un dockerfile contient la recette (sous forme d'un script) pour construire une image docker

```
FROM ubunbu:14.04
RUN apt-get update
RUN apt-get install -y nginx
CMD  ["nginx", "-g", "daemon off"]
EXPOSE 80
```

C'est un ensemble d'instructions permettant de partir d'une une image de base et d'appliquer une succession d'instruction pour configurer une application.

Docker en pratique

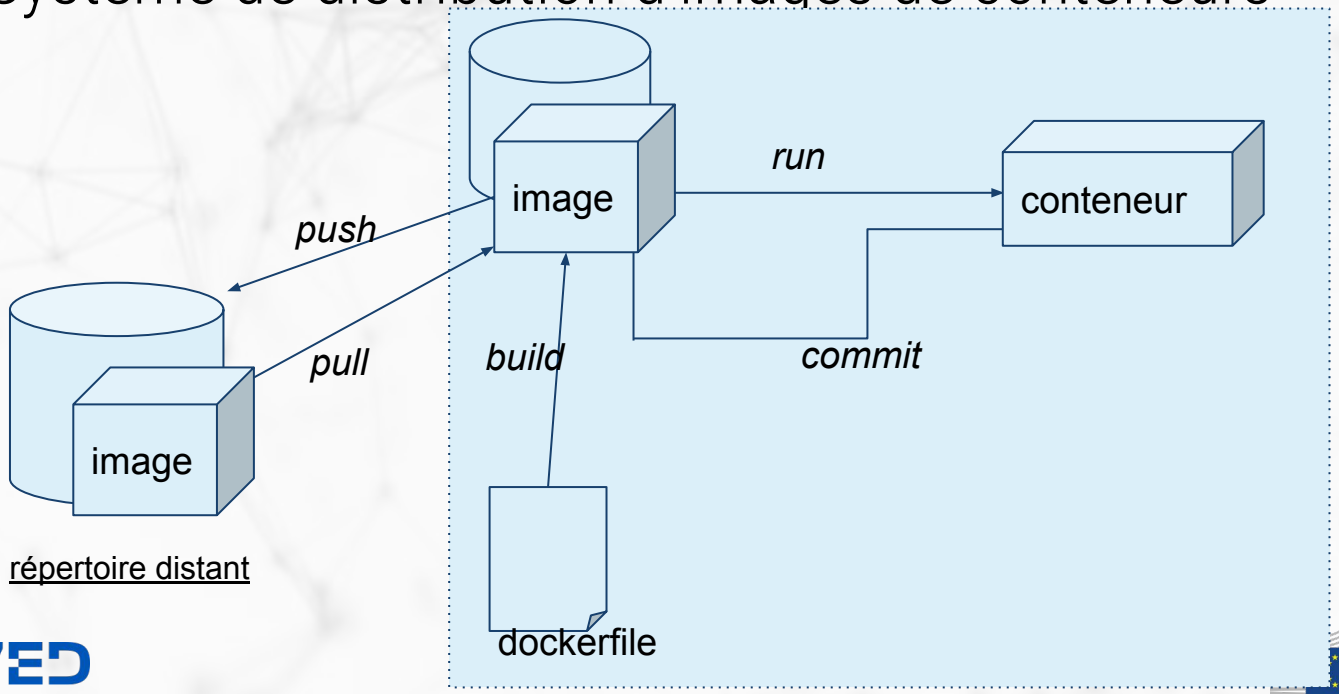
Un environnement de gestion de conteneurs

- Moteur d'exécution de conteneurs
- Système de build d’image (à partir de dockerfile)
- Système de distribution d’images de conteneurs

Docker en pratique

Un environnement de gestion de conteneurs

- Moteur d'exécution de conteneurs
- Système de build d'image (à partir de dockerfile)
- Système de distribution d'images de conteneurs



Docker en pratique

docker pull : récupérer une images depuis un registry privé ou publique.
docker images : lister les images présente sur le docker host.
docker run : démarrer un nouveau conteneur avec une image comme point de départ.
docker ps : lister les conteneurs en utilisation ou qui fut exécutés récemment.
docker stop : arrêter un conteneur en exécution.
docker rm : supprimer un conteneur.
docker rmi : supprimer une image qui fut copier sur le serveur local , ceci ne supprime pas l'image du registry.
docker build : créer une images à partir d'un Dockerfile

Docker en pratique

https://github.com/openminted/alvis-docker/tree/master/openminted-components/tomap

Documentation

https://docs.docker.com/
Training.docker.com
http://brendangregg.com/blog/2017-05-15/container-performance-analysis-dockercon-2017html
https://webcast.in2p3.fr/video/quelle_place_pour_les_containers_dans_le_monde_du_calcul
https://blog.philipphauer.de/discussing-docker-pros-and-cons/
https://github.com/x3rus/training/blob/master/docker/version2/presentation.mkd#whatIsDocker