



PERFECT MEMORY
We Build Knowledge



ENJEUX D'UN SYSTÈME MULTI-AGENTS SÉMANTIQUE

AUDACES – 1^{ER} JUIN 2017



SEMTECH San Francisco
Top 10 Semantic Startup 2013



IBC Innovation
Finalist Award 2013



La France du Numérique
Sélectionné par L'Usine Digitale



2014

BPI FRANCE
Excellence 2014



European Cluste
Excellence 2015



Perfect Memory

| Recherche

- Projet **Mediamap** (2008-2011) en collaboration avec le laboratoire HEUDIASYC de l'UTC
(*HEUristique et DIAgnostic des SYstèmes Complexes*)
UMR CNRS 7253
"Media Management from Acquisition to Publishing"
Développement d'un middleware sémantique
- **Mediamap+** (2012-2014) : Programme de recherche Européen Celtic-Plus
Mise en place d'un écosystème de production audiovisuelle dynamique
Excellence Award (2015)
- Implication dans l'Ontologie **EBUCore**
European **B**roadcasting **U**nion

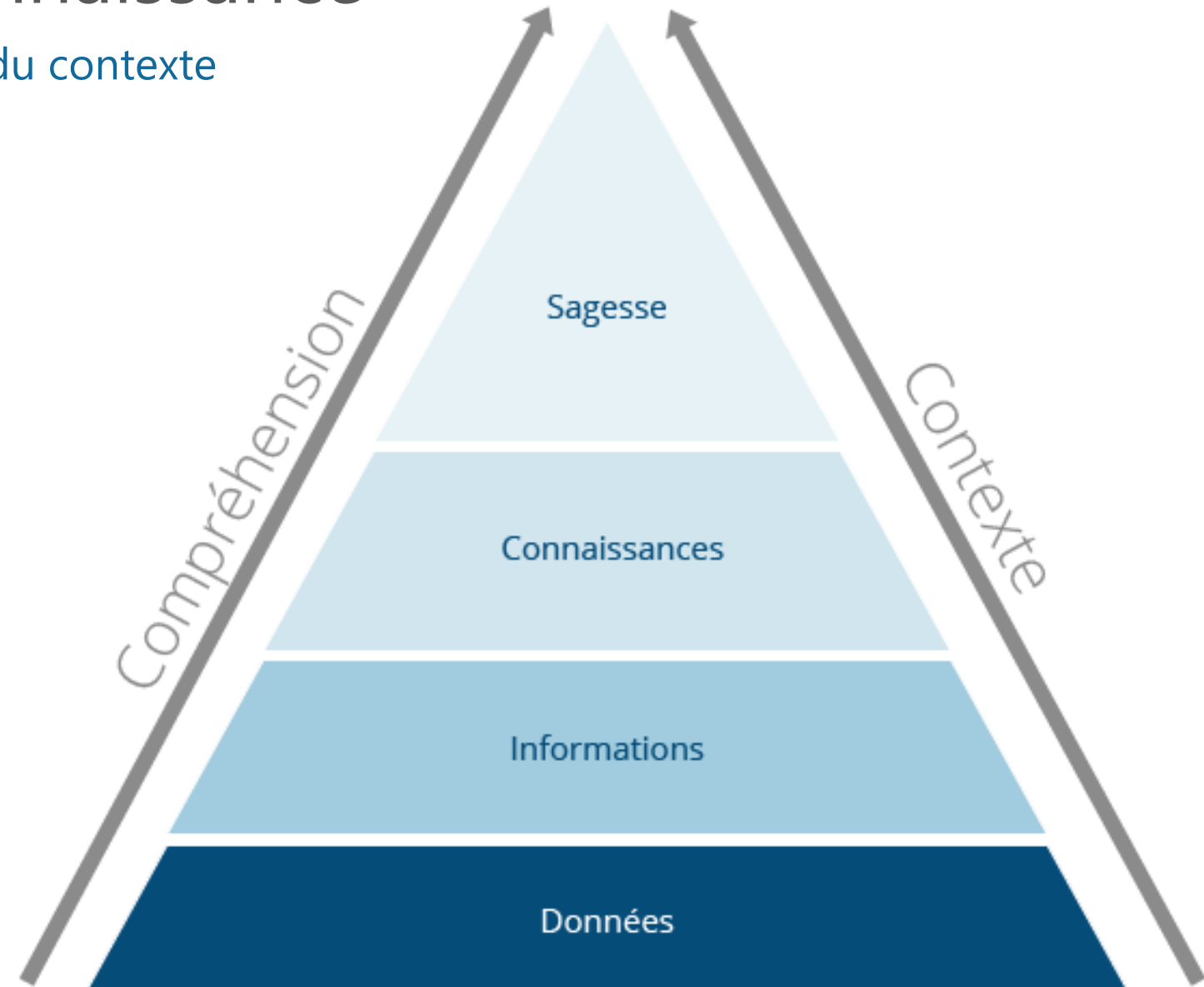


Gestion sémantique

Interopérabilité

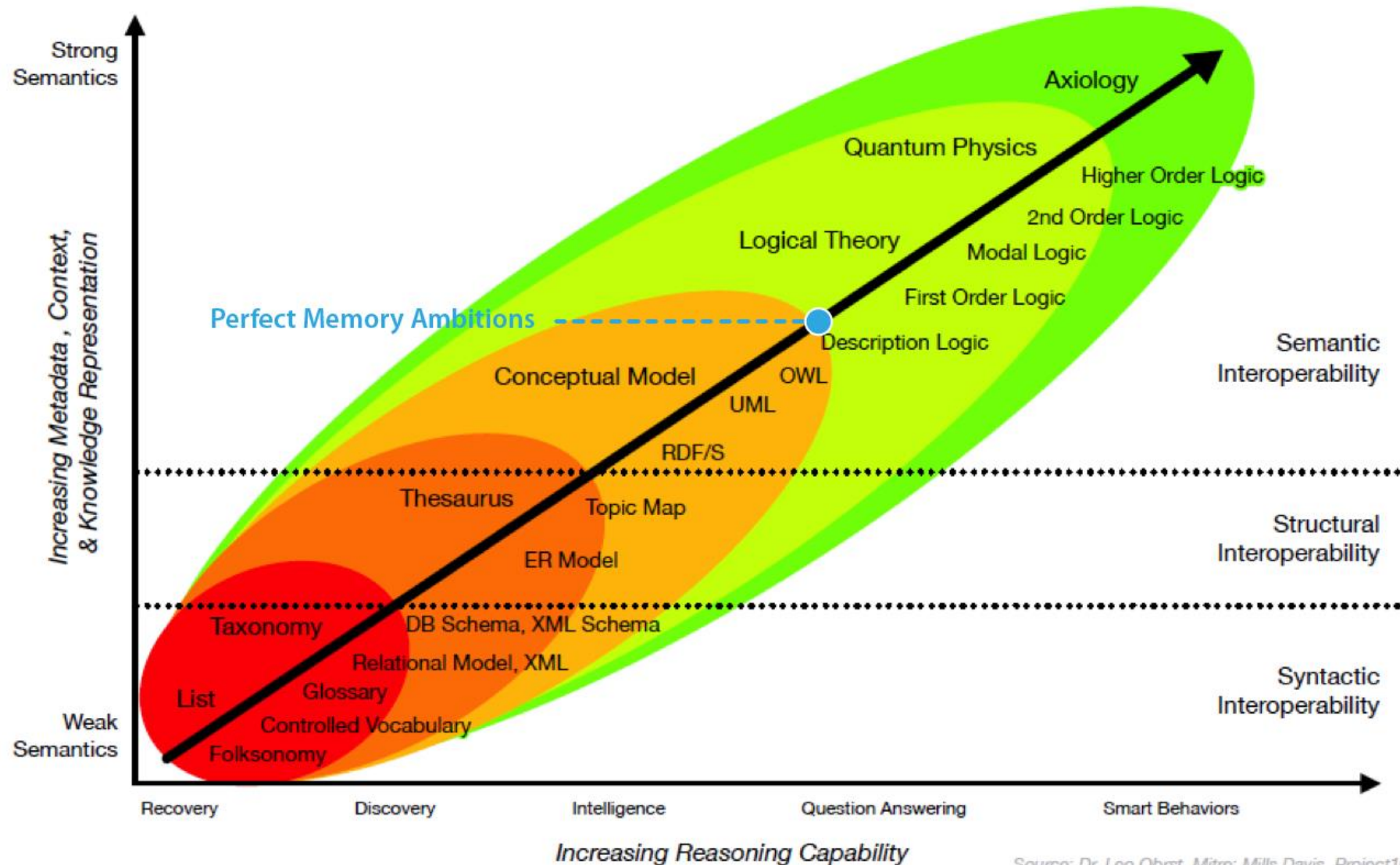
Pyramide de la connaissance

Ajouter de la compréhension et du contexte



Interopérabilité

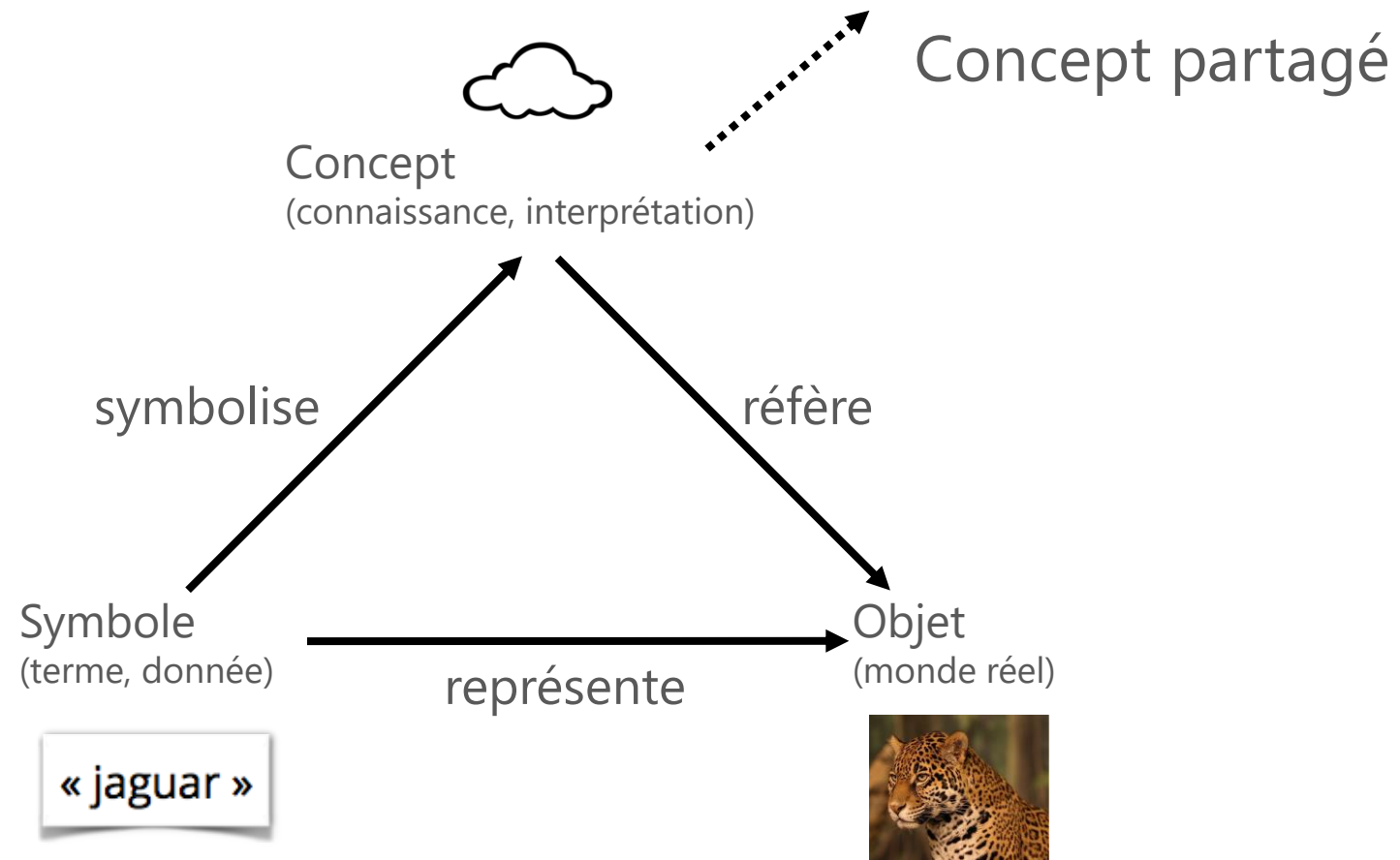
From Searching to Knowing — Spectrum of Knowledge Representation and Reasoning Capabilities



Source: Dr. Leo Obrst, Mitre; Mills Davis, Project10X

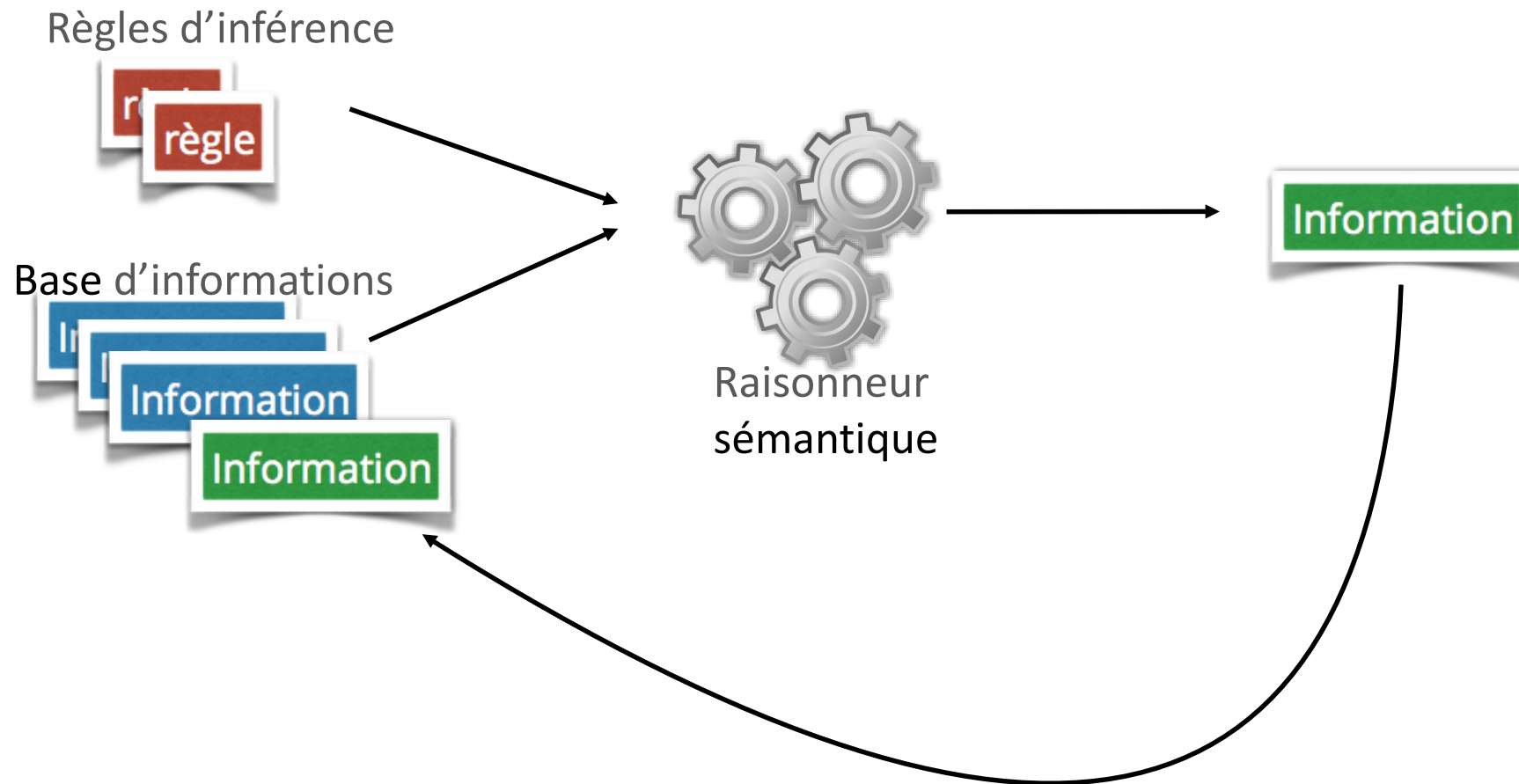
Ontologies

Triangle sémiotique



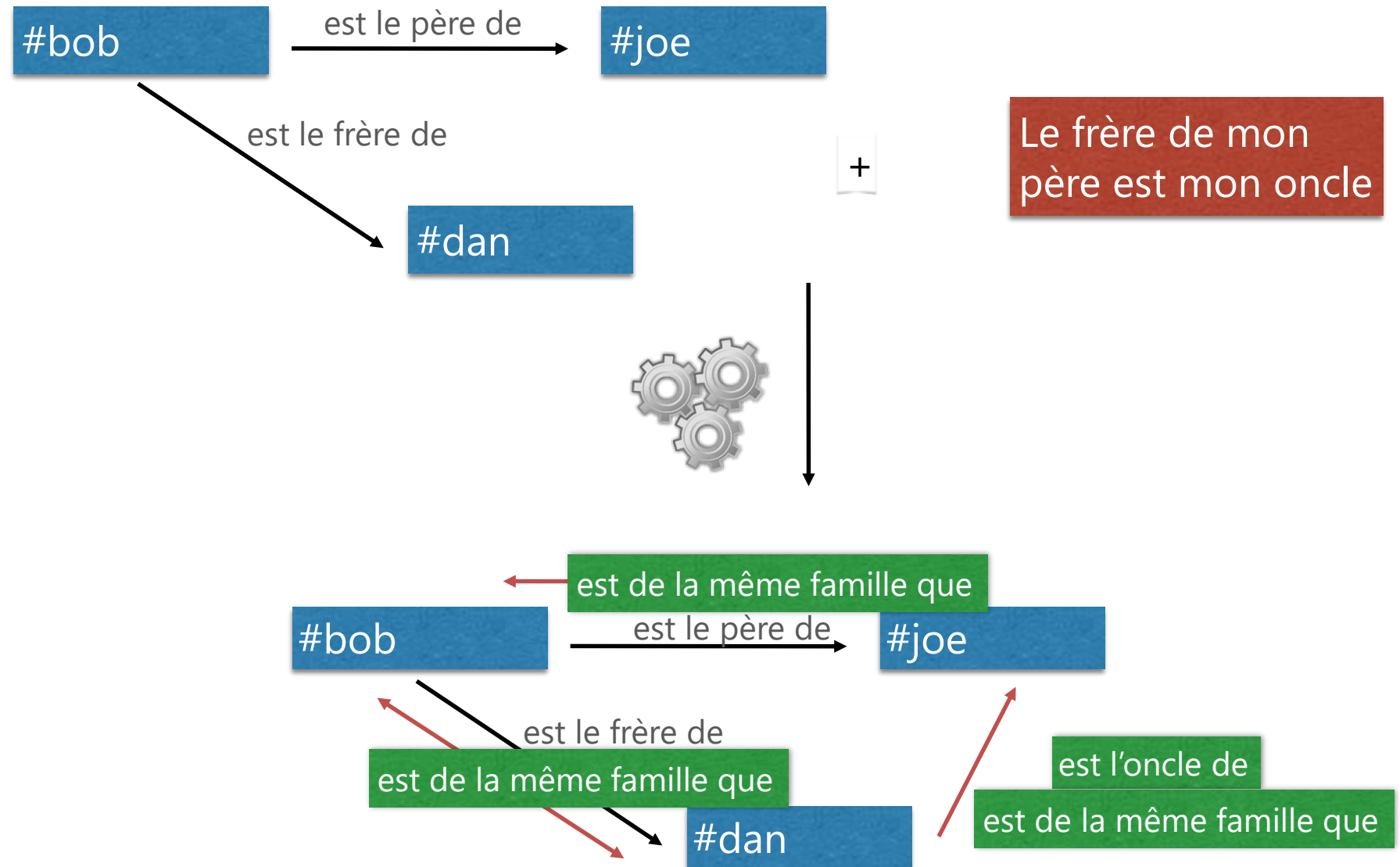
Ontologies

Inférence sémantique



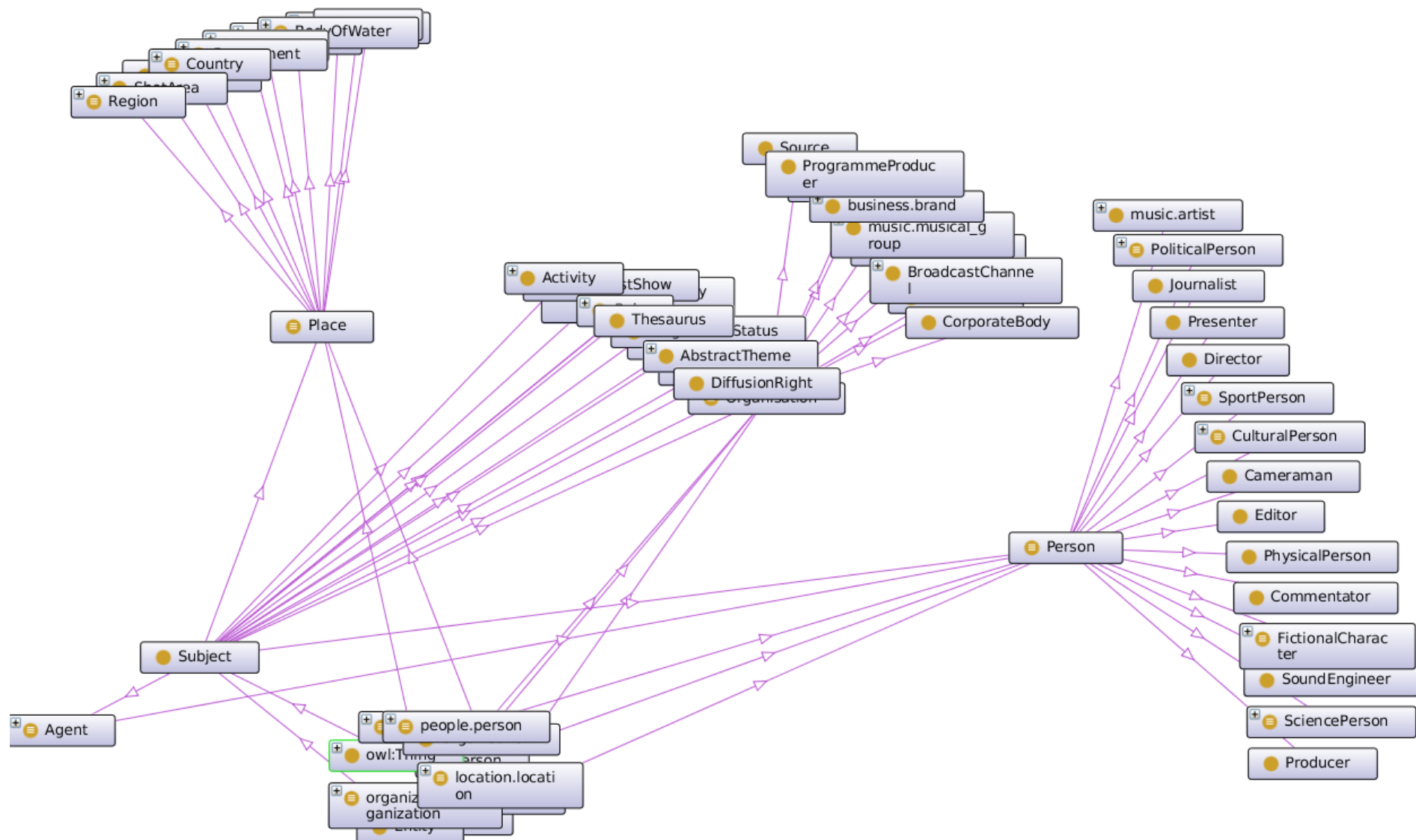
Ontologies

Inférence sémantique



Ontologies

Spécialisations



Workflows

De la modélisation au déploiement

Rôles

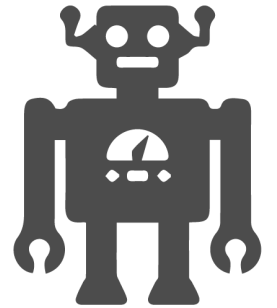
5 rôles clés



Ontologue



Administrateur
Système



Robot



Développeur



Responsable
Plateforme

Ontologie

Gestion de la Connaissance

- Expertise Métier
- Modélisation de la connaissance sous forme de concepts & de relations
- Création des ontologies
- Contact client

Ses outils principaux

- Protégé (*Stanford University*)
- Triplestores



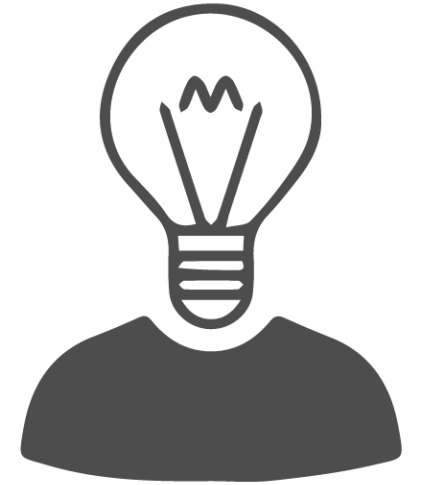
Développeur

Gestion des Services

- Expertise Technique
- Création d'applications (Apps, API, etc.)

Ses outils principaux

- Stack technique : Ruby, Go, Java, etc.
- Bases de données
- Git, Docker



Administrateur Système

Gestion de la Plateforme

- Expertise Technique
- Déploiement de la plateforme et des backends
- Monitoring & Backups
- DBA (ACL, configuration, backups, scalabilité, etc.)

Ses outils principaux

- Outils « PaaS » : OpenShift, Kubernetes, Docker
- Outils de provisioning : Chef
- Bases de données : MariaDB, Elasticsearch, Redis, ...



docker



RabbitMQ



Responsable Plateforme

Gestion des Workflows

- Expertise Métier
- Modélisation des Workflows
- Contact client

Ses outils principaux

- Workflow Manager



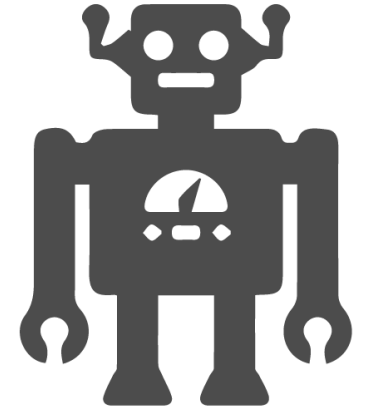
Robot

Gestion des Processus automatiques

- Expertise Binaire
- LifeCycle Application (Installation, Mise à jour, Purge, etc.)
- Simplifier les tâches redondantes & chronophages
- Ami de l'Administrateur Système et du Développeur

Ses outils principaux

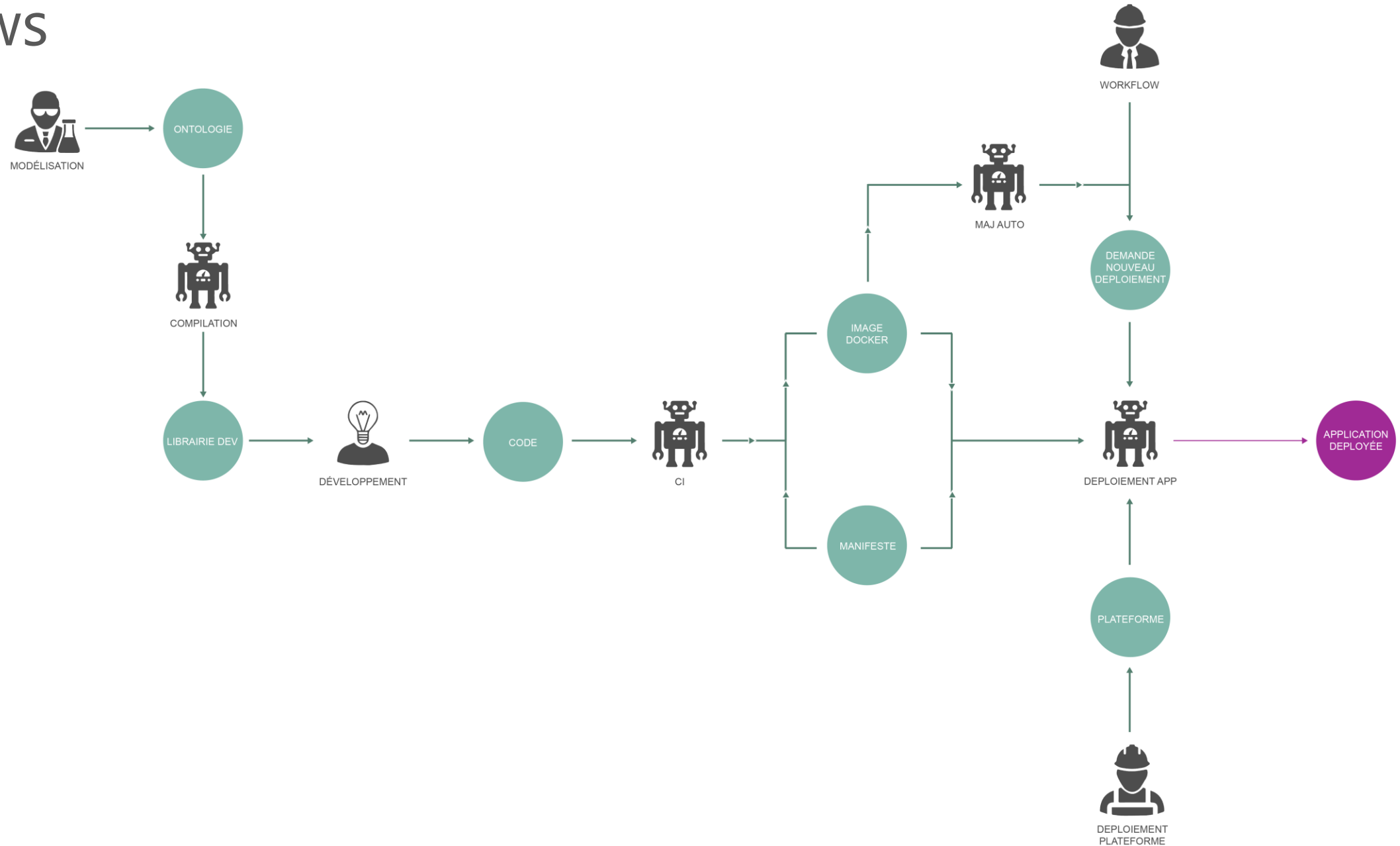
- OpenShift / Docker



OPENSIFT



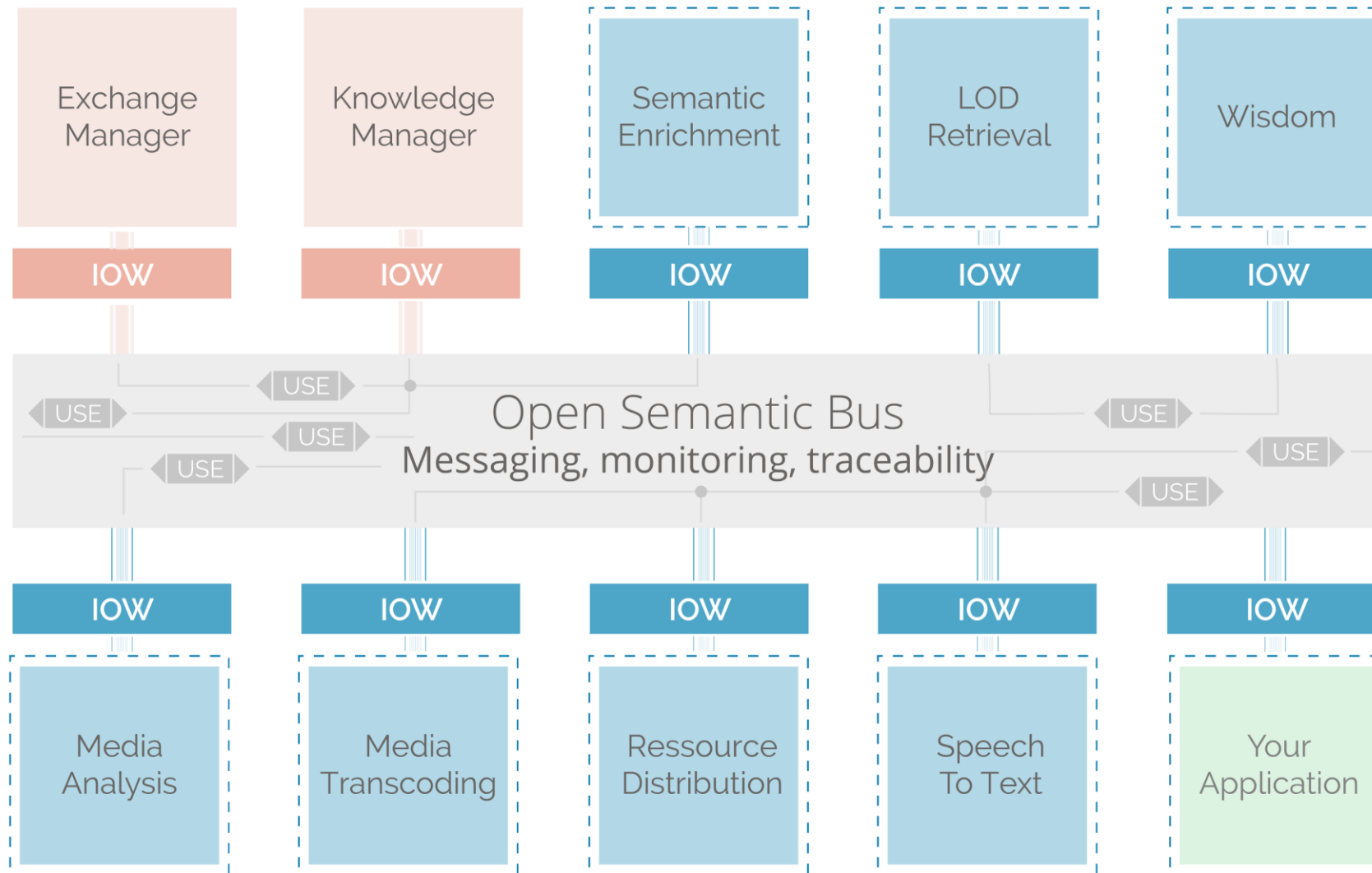
Workflows



Architecture & Framework

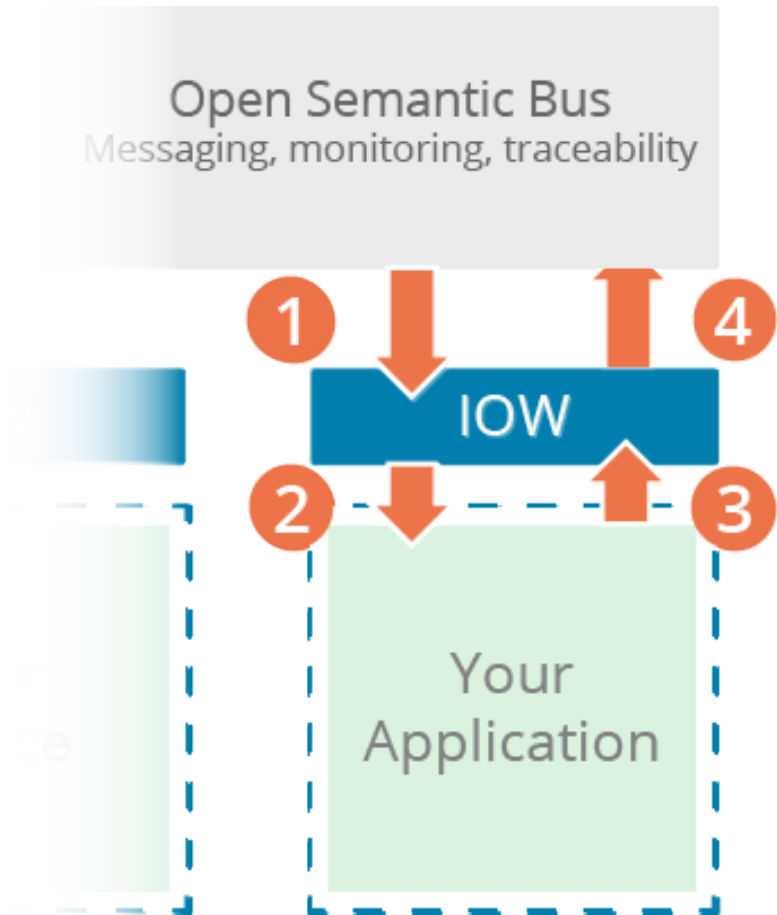
Architecture

Orientée Services sémantiques

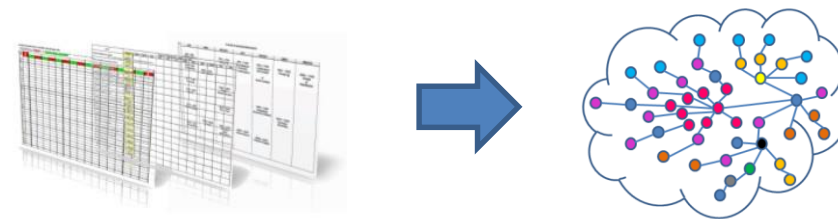


Guichet d'interopérabilité

Principes



- Connecteur (I/O)
- Traducteur (Mapping)



Guichet d'interopérabilité

Framework & Structure d'une application

- **Common Layer**

Gérées par la core team du Framework

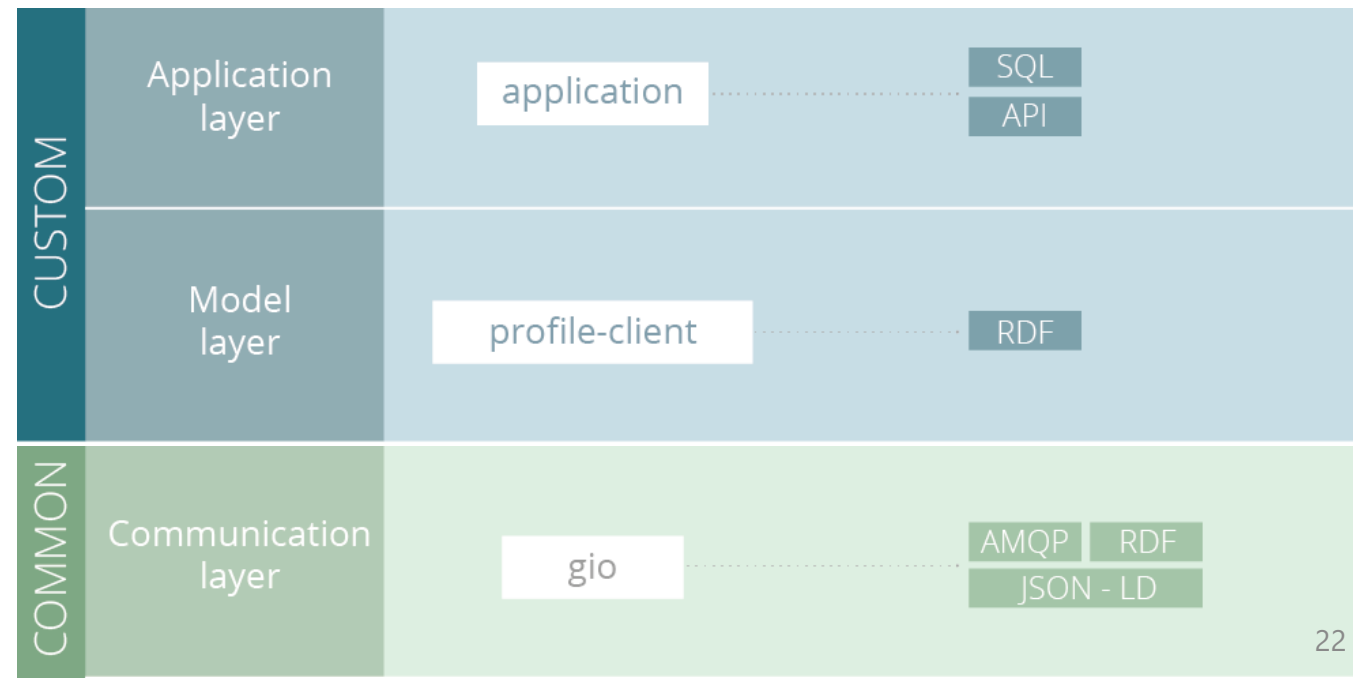
- **Model Layer**

Gérée par l'ontologue

Générée & versionnée automatiquement

- **Application Layer**

Gérée par le développeur du service



Guichet d'interopérabilité

Communication layer « gio »

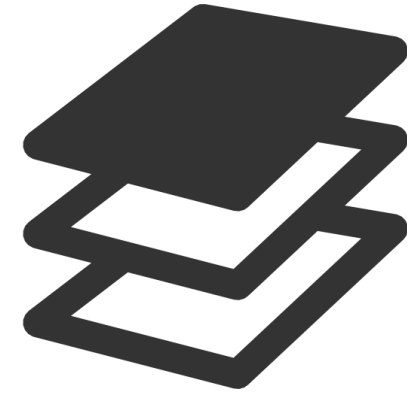
- **Protocole** basé sur **AMQP** (***A**dvanced **M**essage **Q**ueing **P**rotocol*).
- ISO/IEC 19464
- Utilisation du format ouvert **JSON-LD** 1.0
(***J**ava**S**cript **O**bject **N**otation for **L**inked **D**ata*)
- Déclaration et mapping des services de l'application
- Connexion & Authentification
- (Dé)sérialisation des messages et des paramètres
- Support des erreurs & des statuts en temps réel
- Mécanisme de « retry later » exponentiel
- Canal de communication dédié avec chaque instance



Guichet d'interopérabilité

Model layer « profile-client »

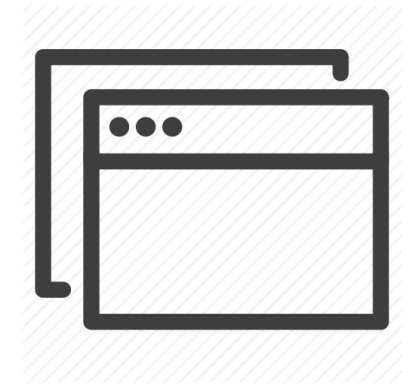
- Générée et versionnée automatiquement sur la base des profiles gérés par l'ontologie
- Object Resource Mapping (**ORM**)
- Abstraction de la couche « data » de sérialisation (JSON-LD)



Guichet d'interopérabilité

Application layer

- Code métier lié au service



Manifeste Applicatif

Dépendances

- **Décrit sémantiquement les dépendances de l'application**
- Supprimer la dépendance du Sys Admin dans le déploiement d'une application
- Nom de l'application
- Paramètres (littéral, chemin, etc.)
- Volumes
- Services (API, Base de données, etc.)
- Variables d'environnement (12 factors)
- Bindings
- Système de packages

```
application:
  metadata:
    name: lod-retrieval
  imports:
    - osb
  parameters:
    google-api-key:
      type: literal
      validates: /\A[a-zA-Z0-9_-]{39}\Z/
      required: true
    configuration-dir:
      type: path
      value: /configuration
      read_only: true
    configuration-vol:
      type: volume
      selector: lod-retrieval-configuration
      required: true
    freebase-blazegraph:
      type: service
      required: true
  env:
    GOOGLE_API_KEY:
      value: ${google-api-key}
    CONFIGURATION_PATH:
      value: ${configuration-dir}
    BLAZEGRAPH_URL:
      value: ${freebase-blazegraph.url}
  volumes:
    configuration:
      mountpoint: ${configuration-dir}
      source: ${configuration-vol}
```

Agents

Application Manager

Agent en charge du cycle de vie applicatif

1. Récupération du manifeste applicatif
2. Extraction des paramètres « utilisateurs »
3. Fusion entre configuration de l'environnement & configuration par défaut
4. Extraction des services / volumes
5. Réconciliation avec les ressources disponibles / affectées précédemment
6. Configuration automatique (génération automatique de credentials)
7. Génération d'un template de déploiement
8. Utilisation de l'API de la plateforme (OpenShift, etc.)

4 modes :

- Installation
- Mise à jour
- Désinstallation
- Purge



Workflow Manager

Agent en charge des processus métiers

- BPM : **B**usiness **P**rocess **M**anagement
- Workflow entièrement dynamique et paramétrable
- Traçabilité de l'ensemble des processus
- Historique de tous les messages échangés (I/O)
- Basé sur la notion d'I/O



Structures :

- Service
- Chaîne
- Condition
- Boucle

Paramètres :

- URI sémantique
- Graphe sémantique
- Littéraux (date, float, string, etc.)
- Données Binaires

Knowledge Manager

Agent en charge de la mémoire

- Gestion des informations (CRUD)
- Persistance des graphes sémantiques
- Inférence
- Moteur de vues sémantiques (SPARQL)

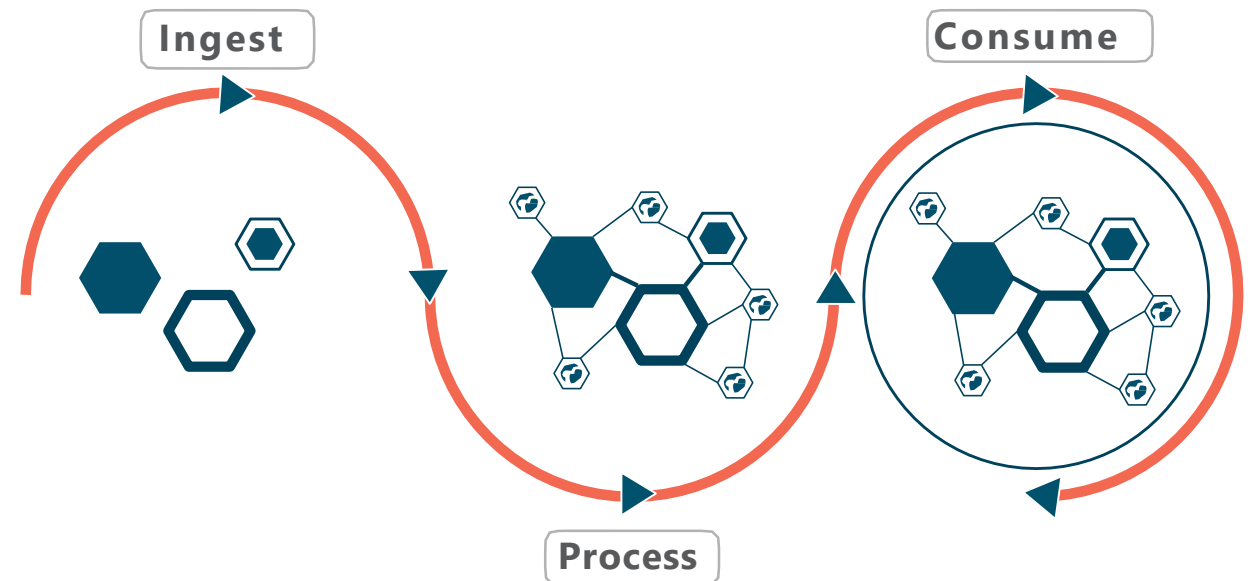


Conclusion

Enjeux

Avantages

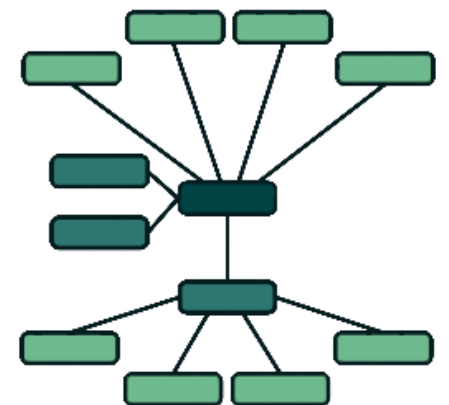
- Modéliser et gérer la **connaissance** ainsi que les **workflows métiers** de n'importe quelle verticalité métier (broadcast, assurance, renseignement, santé, etc.) dans un format ouvert
- **Interopérabilité**
- Brancher facilement de nouvelles briques **applicatives** communiquant avec l'existant
- **Réconcilier** de données (désilotage dans les entreprises)
- Plateforme 100% « Ontology Driven »



Uses cases

Finalité

- Gestion & *internationalisation* de la **connaissance** (thésaurus++)
- Ingestion & réconciliation de n'importe quelle **source de données** (ETL)
- Indexation **automatique** de contenus audiovisuels (vidéo, son, texte) en alliant :
 - Speech To Text
 - Image To Text - Reconnaissance d'image, faciale, etc.
 - Analyse morphosyntaxique
 - Enrichissement avec le Linked Open Data
- **Recherche** intelligente





PERFECT MEMORY
We Build Knowledge

Aymeric BRISSE

CTO

aymeric.brisse@perfect-memory.com

Twitter : @aymericbrisse