Implémentation d'une Plate-forme d'exploration interactive de données complexes et massives (POC)

(Extrait de Mathrice ANF 2018 - Module BD4)

https://indico.math.cnrs.fr/event/3550/timetable/#b-880-bd-bd

Christophe Cancé

Univ. Grenoble Alpes

UMS GRICAD

PLAN

I. Approche interactive
- les couches : collecte, requête, visualisation, couche élaborée d'informations

II. Calibrage
- Ordres de grandeurs et enseignements de l'Analyse de logs avec ELK
- Introduction aux ETL

III. Collecte et organisation de l'information
Acheminement de l'information vers un modèle cible

IV. Bases graphes
- Choix d'une solution
- structure, propriétés
- exploitation, intégration

V. IHM de navigation dans l'information
Apports des SIG et webmapping avec Openlayers.

VI. Moteur de recherche et Visualisation

VII : production d'information métier.
Exemple : trajectoire

I.Contexte, objectifs et moyens

Problématique :

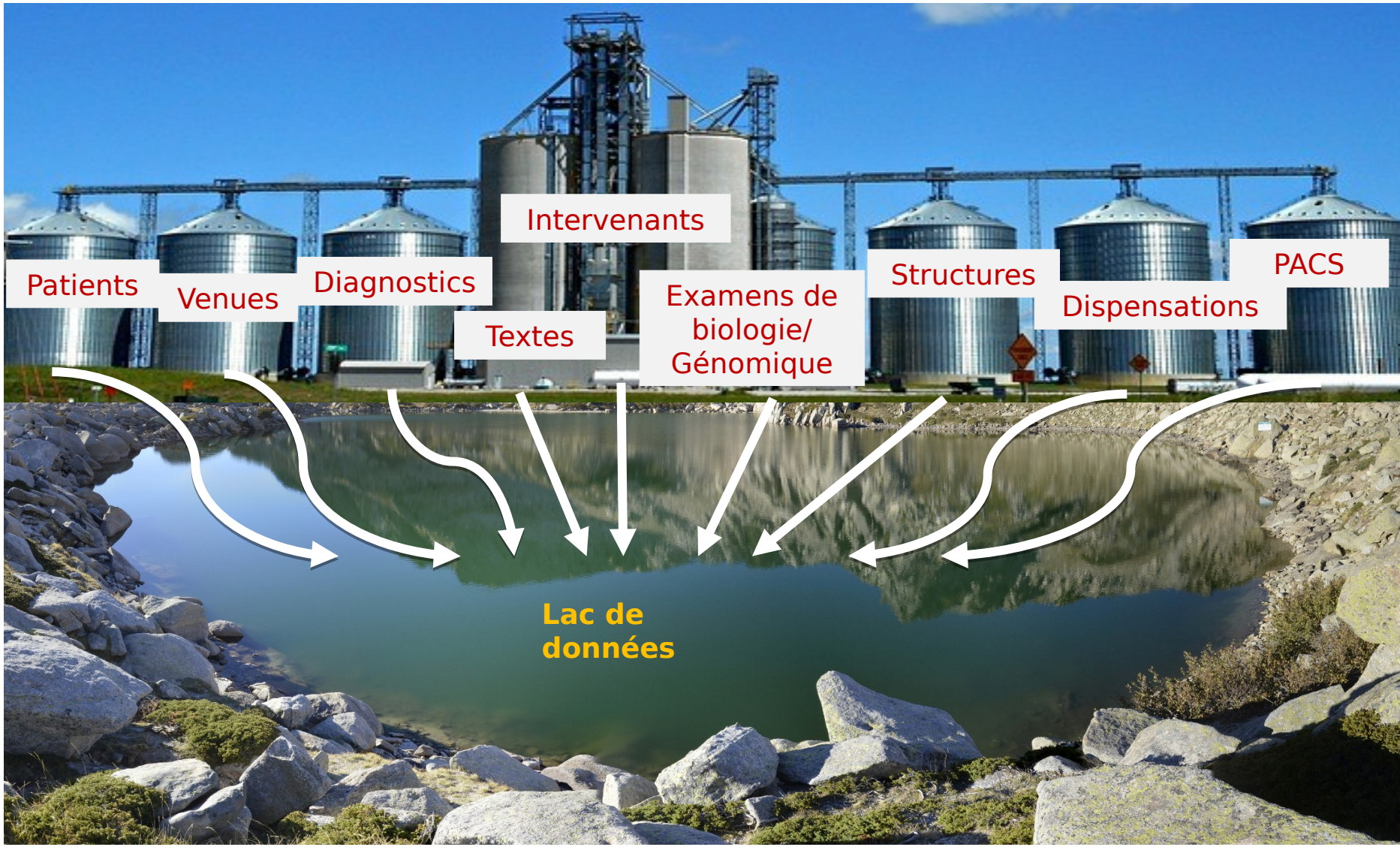
- Réunir l'information pertinente sous une forme organisée et interopérable.
- L'interroger de façon interactive sur des volumes significatifs
- Observer les résultats intermédiaires
- Permettre d'élaborer de nouvelles couches d'information métier

Contexte : économie de code, briques interchangeables

Moyens : intégration, généricité du code et cohérence de la donnée

- Approche graphes : intérêt et points d'attention
- Focus collecte de données ETL
- Lac de données et requêtes: base graphe
- IHM
- Moteur de recherche et data-visualisation

Le projet PREDIMED



Conception d'un lac de données: préalable à l'intégration des données

- Contexte de données potentiellement hétérogènes, complexes et massives stockées dans des bases de données disparates (cas de silos multiples)
- Objectif: rassembler les informations collectées par la couche ETL sous une forme organisée et exploitable

  => concevoir le schéma cible de données, au plus près du sens intrinsèque du phénomène qui doit être observé.

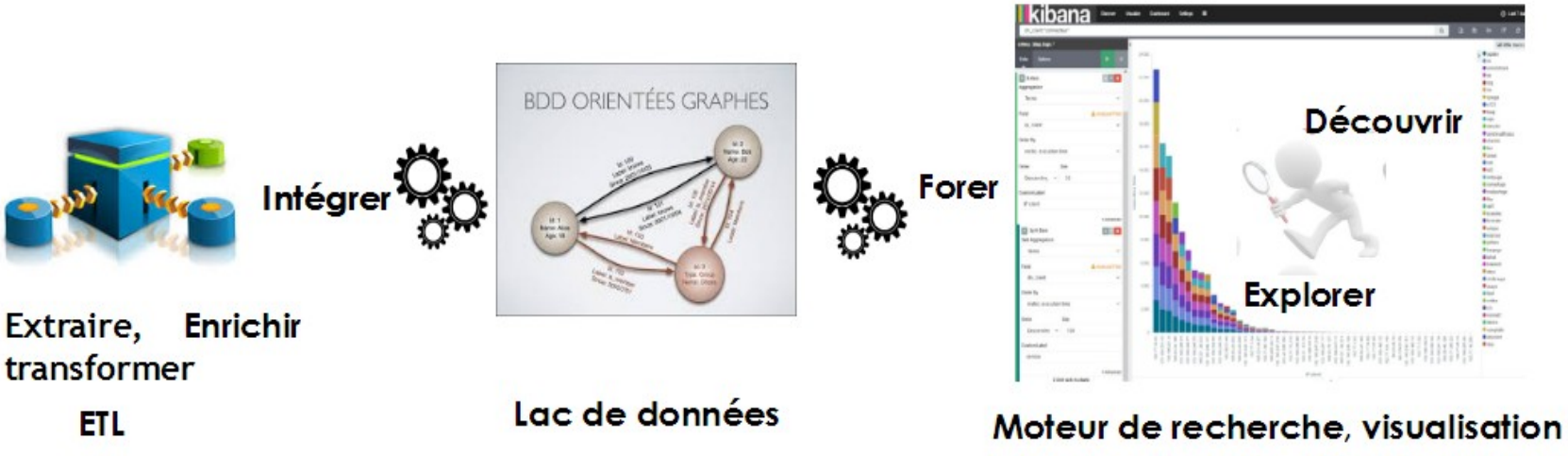
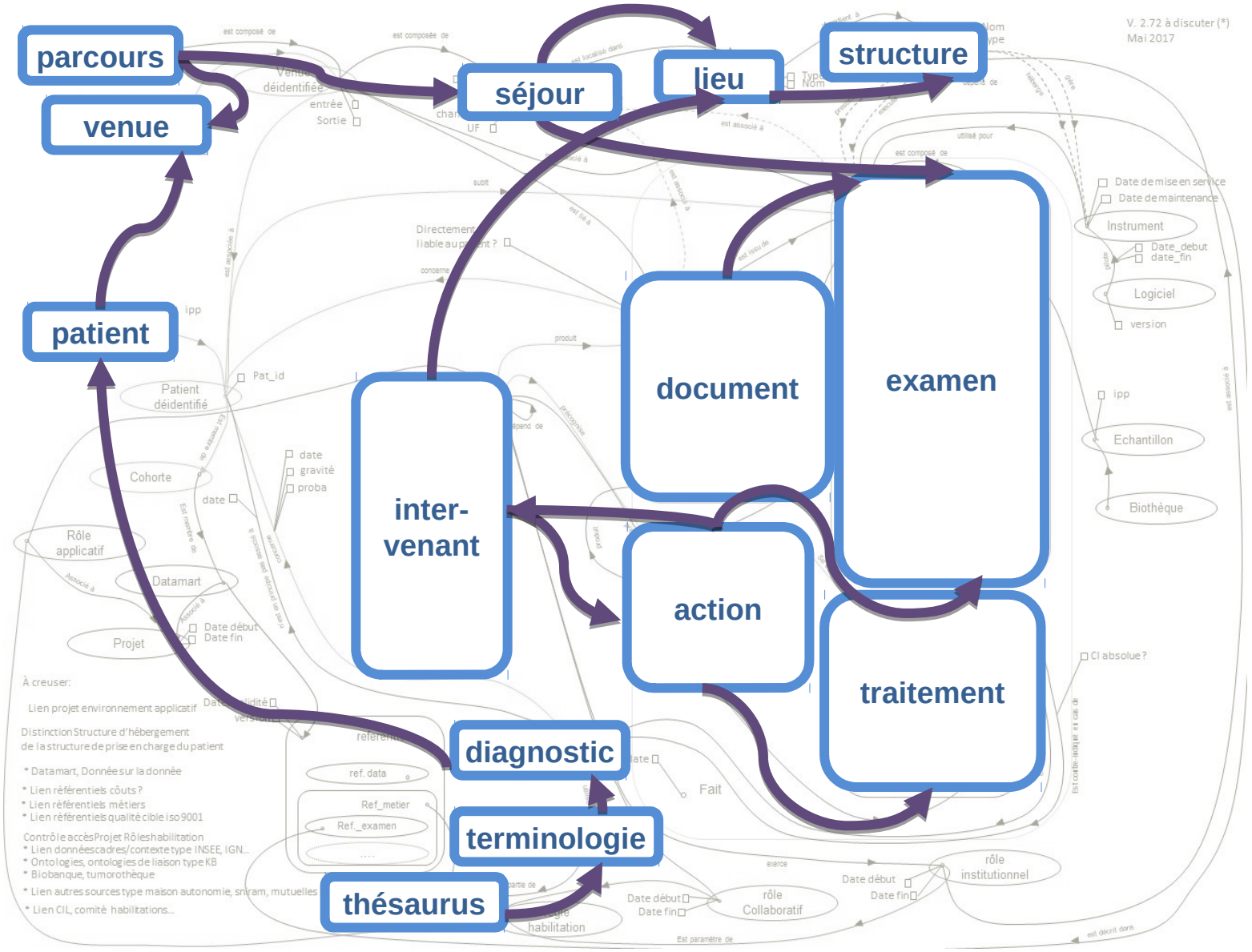


Figure 3: Alimentation du lac de données, gisement d'information

Organisation cible des données (PREDIMED)



Calibrage – Analyse de logs LDAP avec Elastic - Bilan

- 1 VM « standard » avec 4 coeurs 16Go RAM, 1 To Disque, QoS 2000 IOps
- 66 millions de lignes par jour en moyenne sur la COMUE GA
- Maximum admissible observé ~1000/lignes/sec en lecture sur un port UDP
- Trop juste pour le temps réel => lecture depuis un fichier => dérive 8h/j
- Agrégations concentriques sur 500 000 000 d'entrées en ~1 seconde
- **Pile Elastic search Kibana adaptée pour forage interactif dans les données**

  Organisation de la redondance

  Projection dans des index

  Usage en lecture
- https://www.jres.org/fr/presentation?id=90

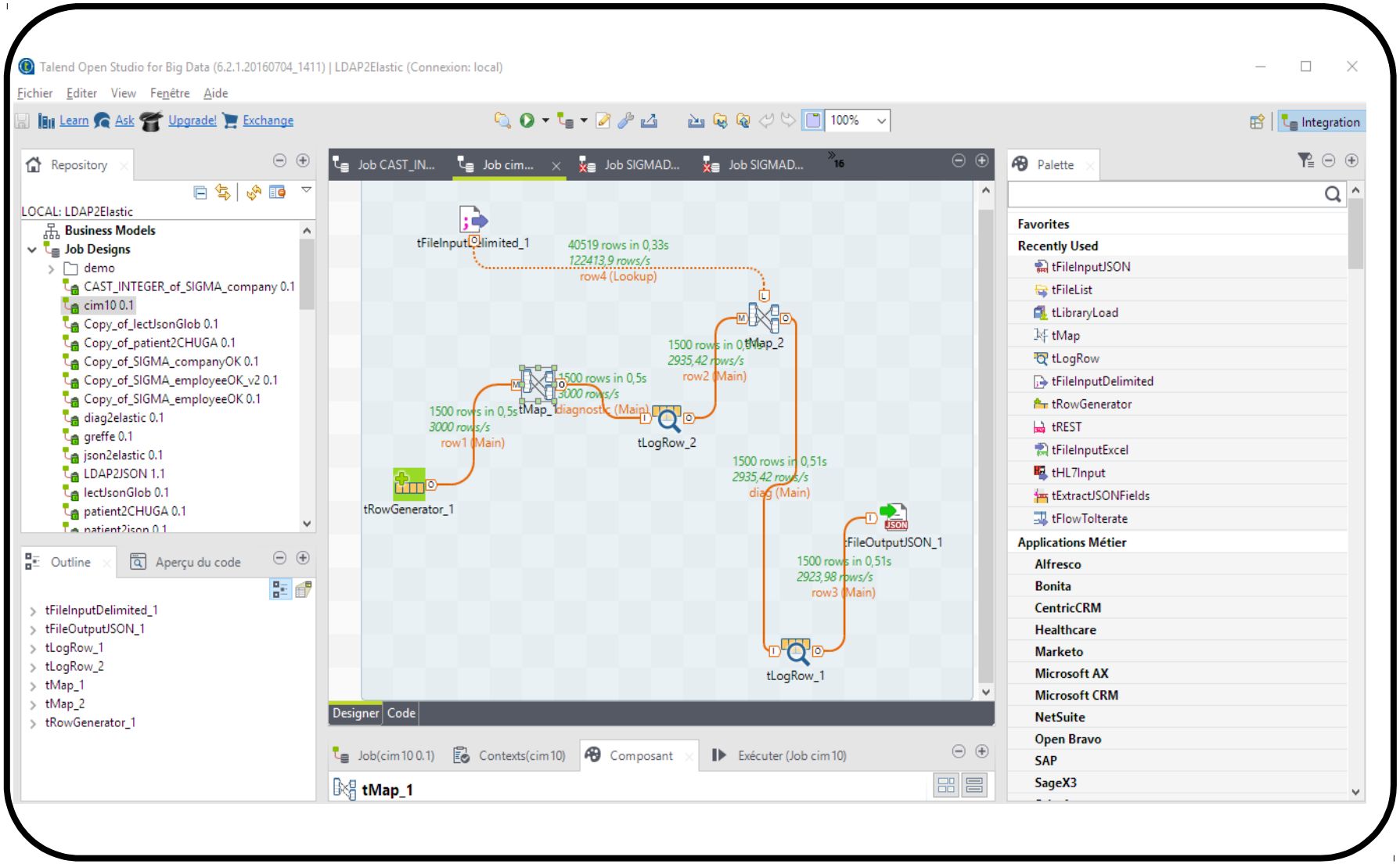
III. Collecte et organisation de l'information - Talend

- Extract Transform Load – Talend - https://fr.talend.com/
- Logiciel d'intégration de données **open source**

  2006 : https://fr.wikipedia.org/w/index.php?title=Talend_Open_Studio_for_Data_Integration&action=edit&redlink=1
- Talend membre de la fondation Apache de 2010
- Générateur de code Java
- Business model open Core : fonctionnalités supplémentaires payantes

  support, outils de développement enrichis, ...
- **900 composants et connecteurs utilisables dans des jobs**
- Sequenceur de jobs
- A un projet correspond un espace de travail, des jobs

Job : graphe d'exécution de traitements choisis parmi 900 composants



Organiser l’information

Conception d’un lac de données - Méthode

- Concevoir avec qui ? Les ingénieurs SI et experts métiers du domaine
- Pour produire quoi ? Identification des concepts pertinents, leurs propriétés, leurs relations.

  Phase cognitive : reconnaissance des ensembles, éléments, propriétés, relations.
- Comment ? Interview experts métiers, modélisation collaborative par itération

  Définition des éléments de la structure => thésaurus

  Définir une représentation partagée de l'organisation de l'information

  CHUGA : choix d’une méthode de type entité association fondée sur les hypergraphes

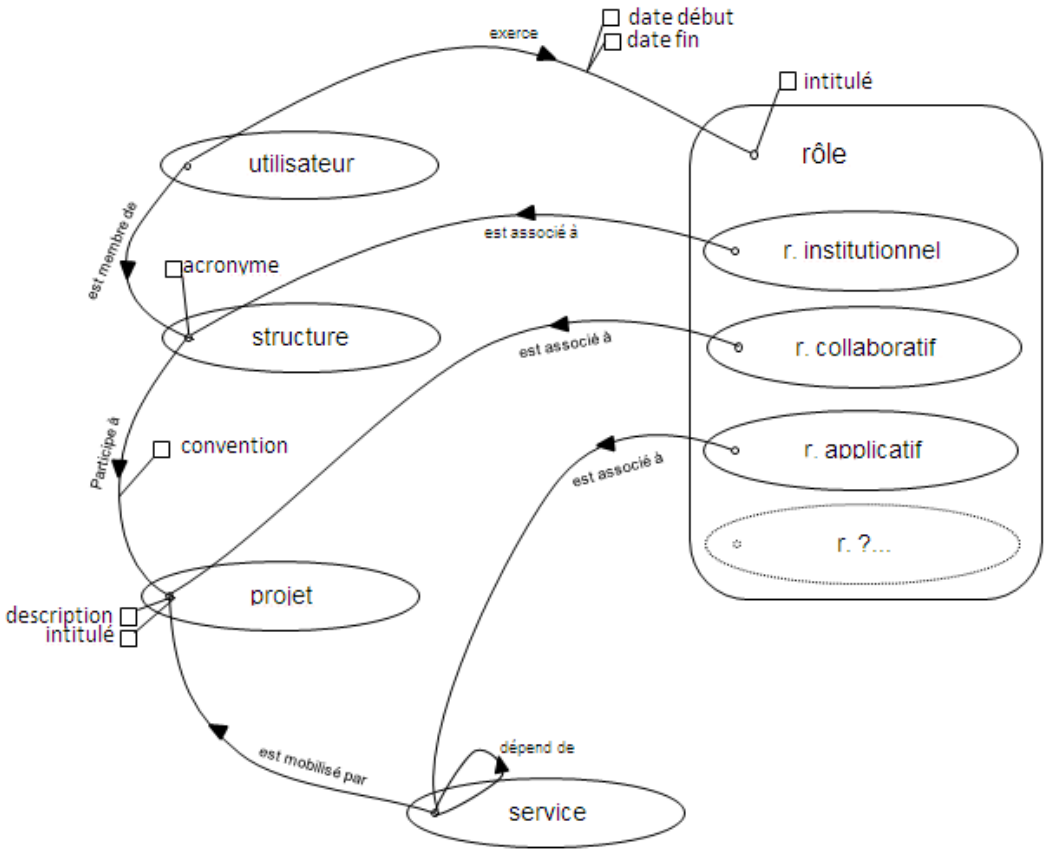
  [Hypergraph Based Data Structure, F. Bouillé – 77]

  http://cours-fad-public.ensg.eu/course/view.php?id=44
- Vérification de la capacité de la structure à supporter les algorithmes nécessaires aux traitements des use-cases exprimés par les experts métiers.

Exemple de structuration « entité-association » de type graphe – (HBDS)

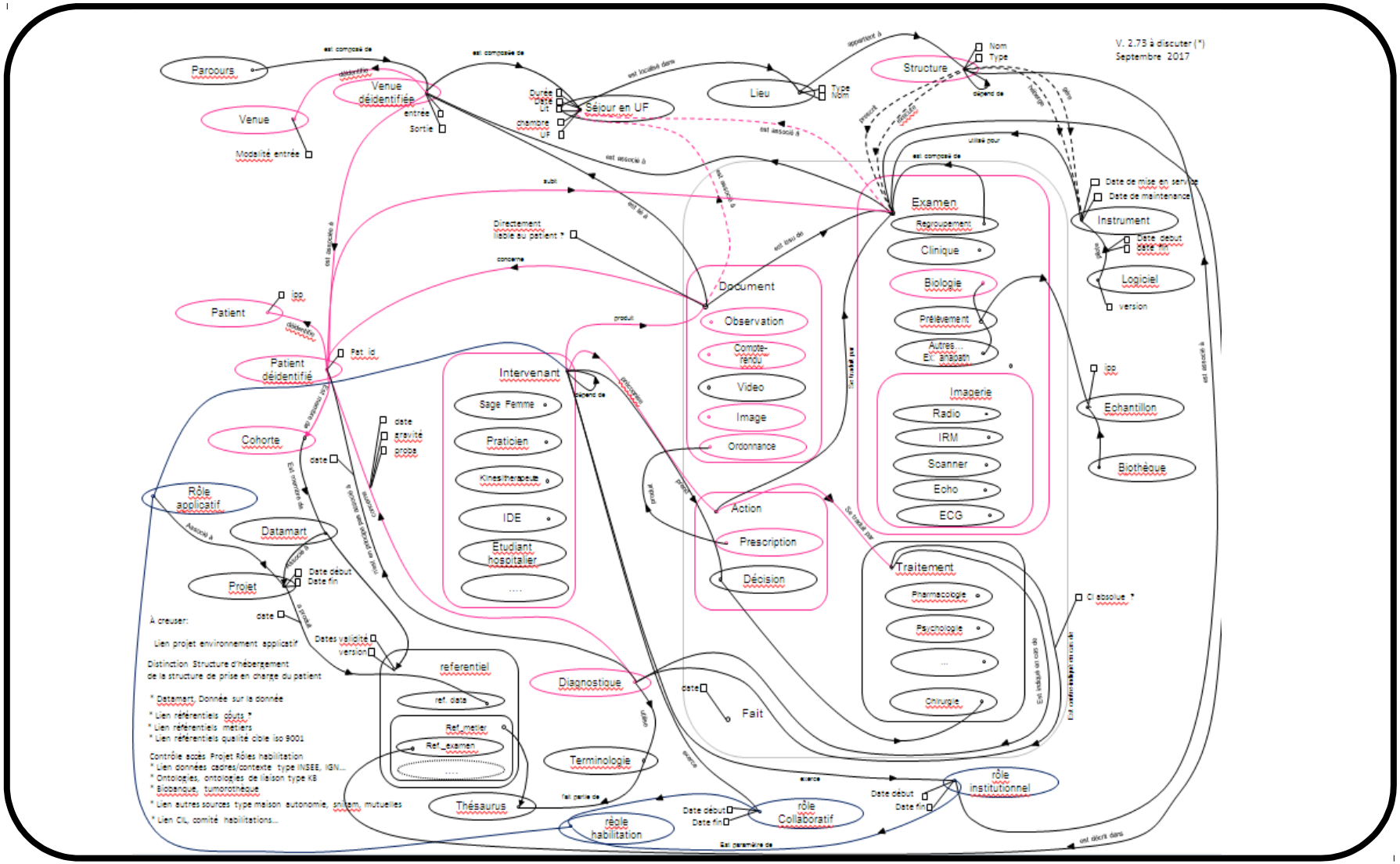
- Sommets (classes, d'objets) : concepts majeurs
- Arcs orientés, nommés : liens entre les concepts
- Attributs (de lien, classe, hyperclasse)

- Modèle évolutif
- Représentation propice au dialogue
- Rassemble les concepts proches



IV.Approche « graphes » et bases graphes

Modèle pivot du lac CHUGA: décloisonnement de l'information (« dé-silo-tage ») via l'ETL



Du modèle à la base orientée graphe (ArangoDB)

- Similitude des concepts utilisés en modélisation (HBDS)
- infrastructures de type base orientée graphe
- Objets représentés sous forme d'ensemble de **couples de clefs-valeurs constituant des documents**
- Simplicité et souplesse des structures, parfois inexistantes dans les bases graphes
- Classes et Liens forment des **collections d'objets documents** (Arangodb), correspondantes à des unités d'indexation

  Classes : l'ensemble des sommets correspondants à une classe constitue une collection de documents de type « vertex »

  Liens : l'ensemble des liens correspondants à une relation est rassemblé dans une collection documents de type « edge » dont 2 couples de clef-valeur indiquent les sommets initial et final.
- Format natif **JSON** (ex : tableaux, listes vecteurs, matrices, complexes…)
- Exploitation : requêtes SQL-LIKE spécifiques, fonctions de propagation dans un graphes, chemin le + court

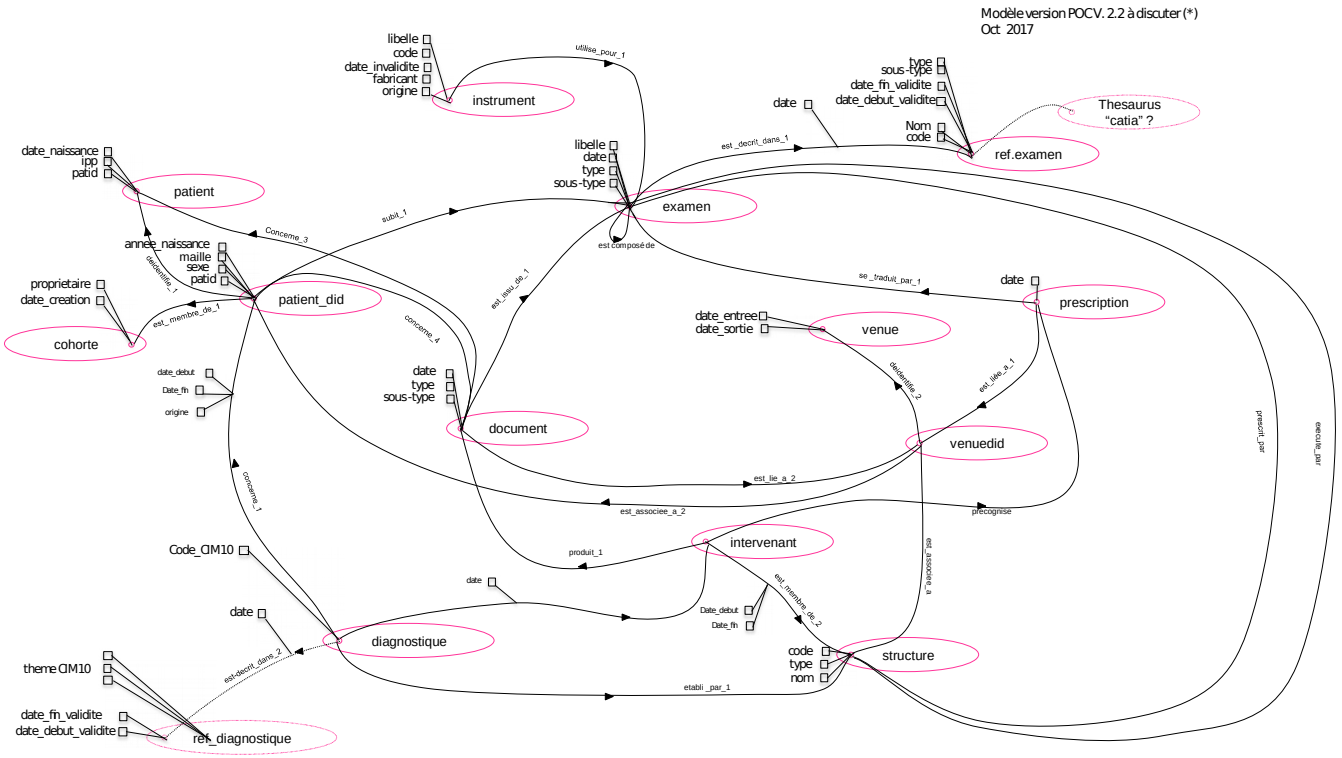
  Des foncteurs permettent de **naviguer par bonds** d'ensemble d'objets à d'autres à mesure que l'on se propage le long des arcs en appliquant d'éventuels filtres sur les propriétés et orientations. http://cours-fad-public.ensg.eu/pluginfile.php/1270/mod_imscp/content/1/res/10_2_-_Les_foncteurs_de_base.pdf
- Implémentation de l'héritage : jouer sur la souplesse des structures
- => possibilité de requête sur les champs communs aux objets de différentes classes d'une hyperclasse
  => économie d'écriture de code

Fonctionnalités cibles nécessaires à la plate-forme

- Compatibilité d'échange web services **HTTP REST JSON**
- intégrer une **API customisable** permettant de faire abstraction des spécificité du langage de requête interne, fréquemment spécifique.

  => développer un moyen graphique d'interroger le lac.
- Capacité à interroger de l'ordre de $x.10^7$ à $x.10^8$ objets et leurs relations de façon interactive.

Du modèle conceptuel au POC : choix d'implémentation



Réunion des objets dans les ensembles du plus haut niveau du modèle.
Distinction des classes (sous ensembles) avec un attribut commun « Type »
=> simplification de la structure, généricité nécessaire du code, évolutivité

Solutions open source envisagées

- Neo4J : https://neo4j.com

  + lisibilité du langage de requête « cypher » : https://www.opencypher.org/

  + Compatibilité Apache's TinkerPop API

  -structuration de l'information ?

  - coûts potentiels de la montée en charge

```
MATCH (me:User),(me)-[rating:RATED]->(movie)
WHERE me.name = 'Me'
RETURN movie.title, rating.stars, rating.comment;
```

- Orientdb : https://orientdb.com/

  base document multi-modèles (document, graphe, cle/valeur,,), sous licence Apache

  + Compatibilité Apache's TinkerPop API

  + clustering, sharding

  +scalabilité

  - documentation API, personnalisation

- **Arangodb : https://www.arangodb.com/**

  base multimodèle (document, graphe, clé/valeur), JSON natif (nested any depht, arrays) , sous licence Apache

  + simplicité de la structure et lisibilité des informations

  + API HTTP REST JSON personnalisable microservices foxx

  + documentation+++

  +- AQL puissant mais spécifique

  + Graphql (facebook 2015)

Focus sur ArangoDB

- Base orientée graphe « in memory » clusterisable (sharding)
- Base graphe, clef-valeur, document (les nœuds et arcs du graphe peuvent avoir des attributs)
- Format natif JSON
- Unités d'indexation sous forme de collections (sommets ou arcs)

- Accès depuis le service HTTP intégré http://localhost:8529

depuis la console : unix> arangosh

shell de commandes javascript

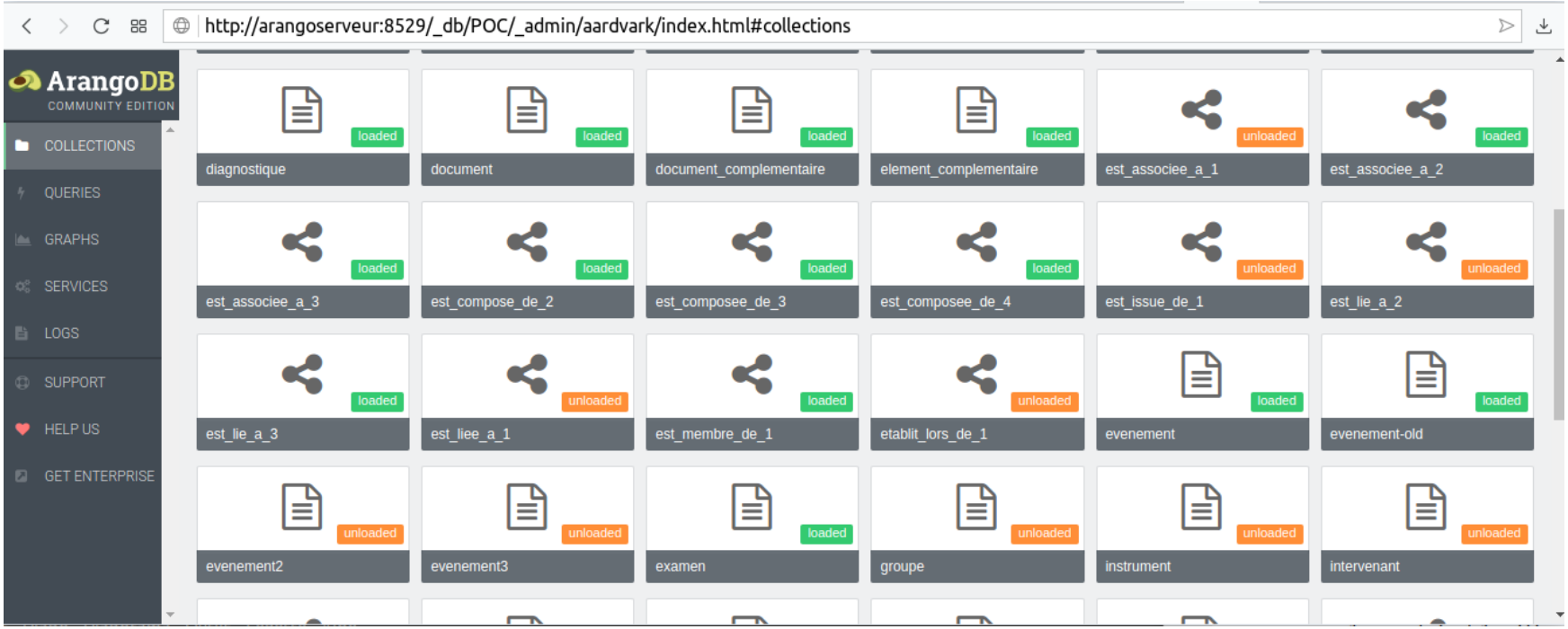
- https://www.arangodb.com/documentation/

ArangoDB – structure graphe des données

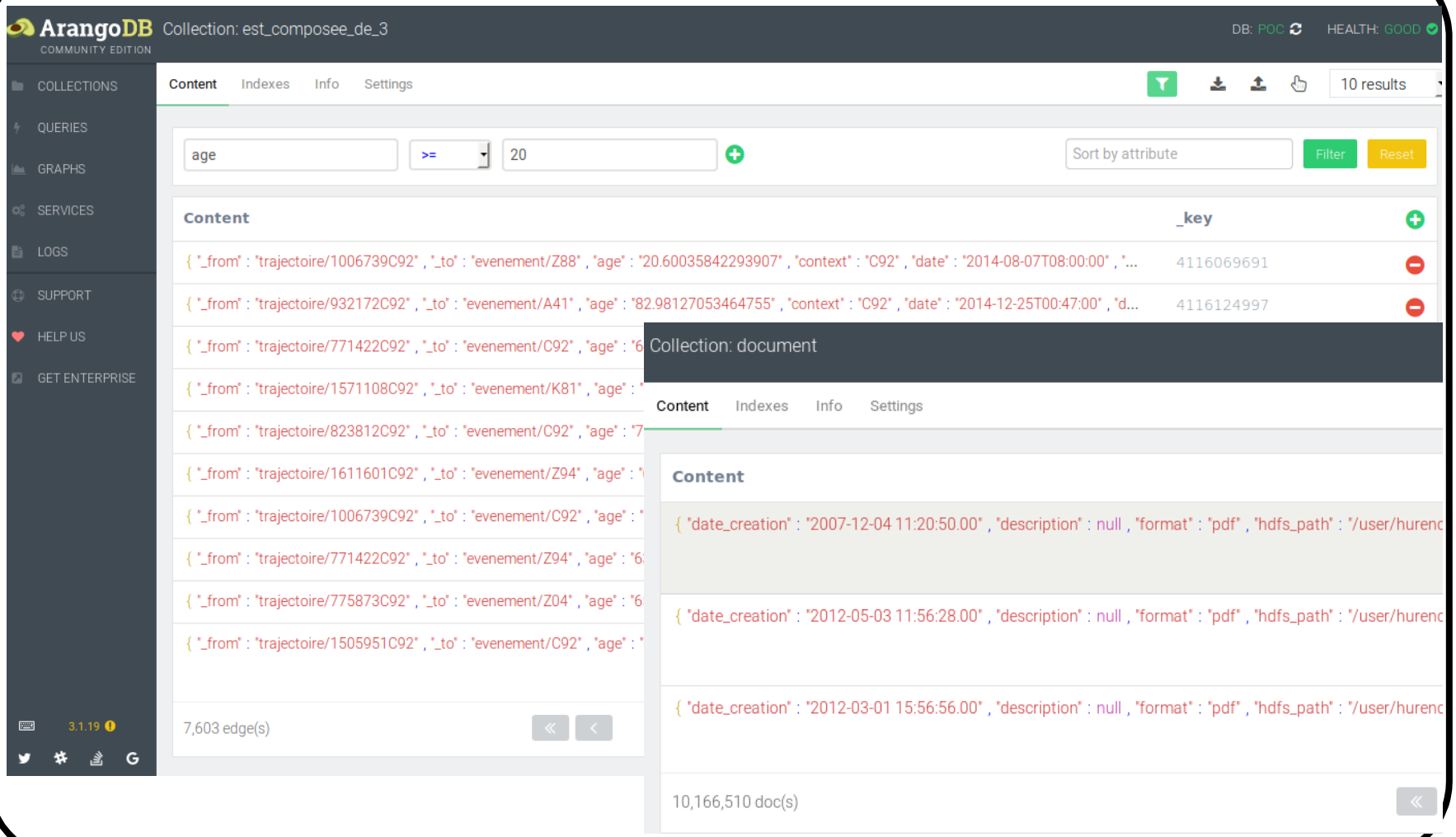
- Collections rassemblent les objets de types : « Document » (Sommet)

  « Edge » (arc)

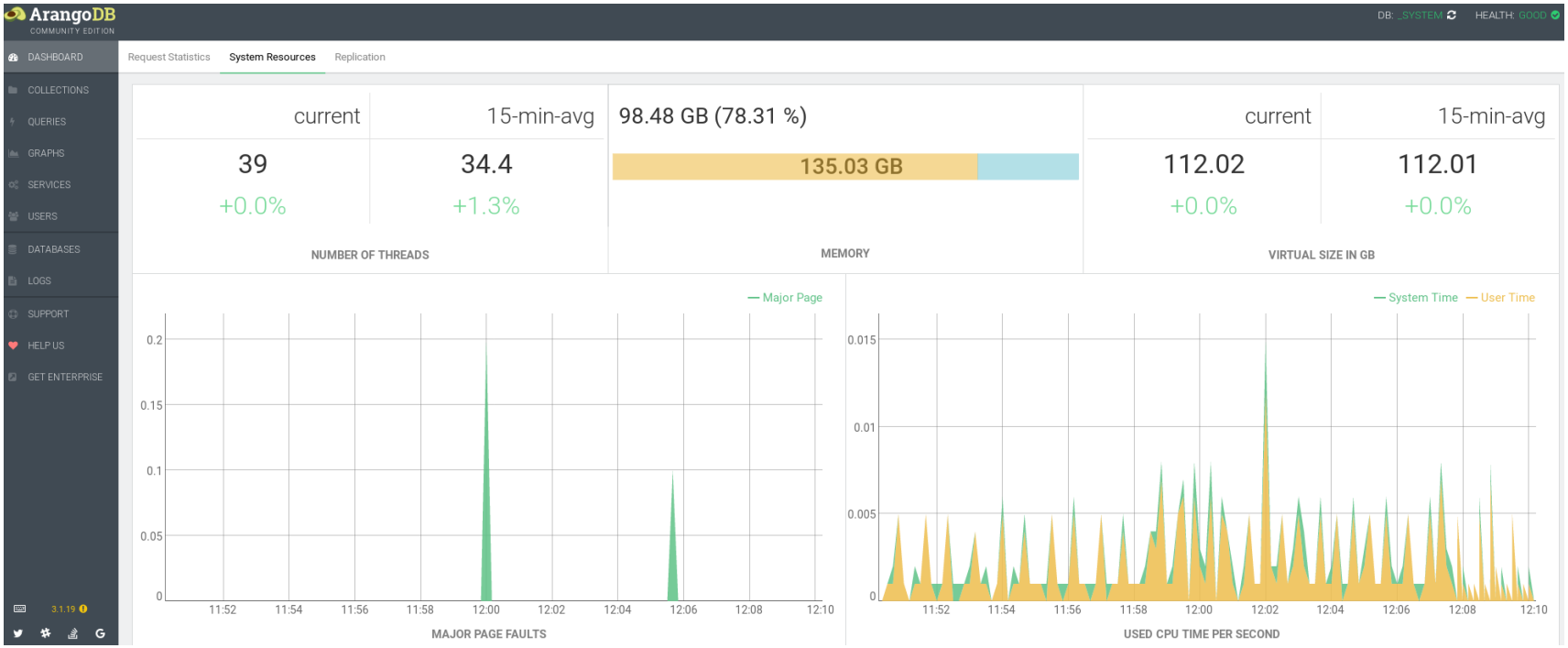
  Un index par collection



Contenu d’une collection : documents JSON

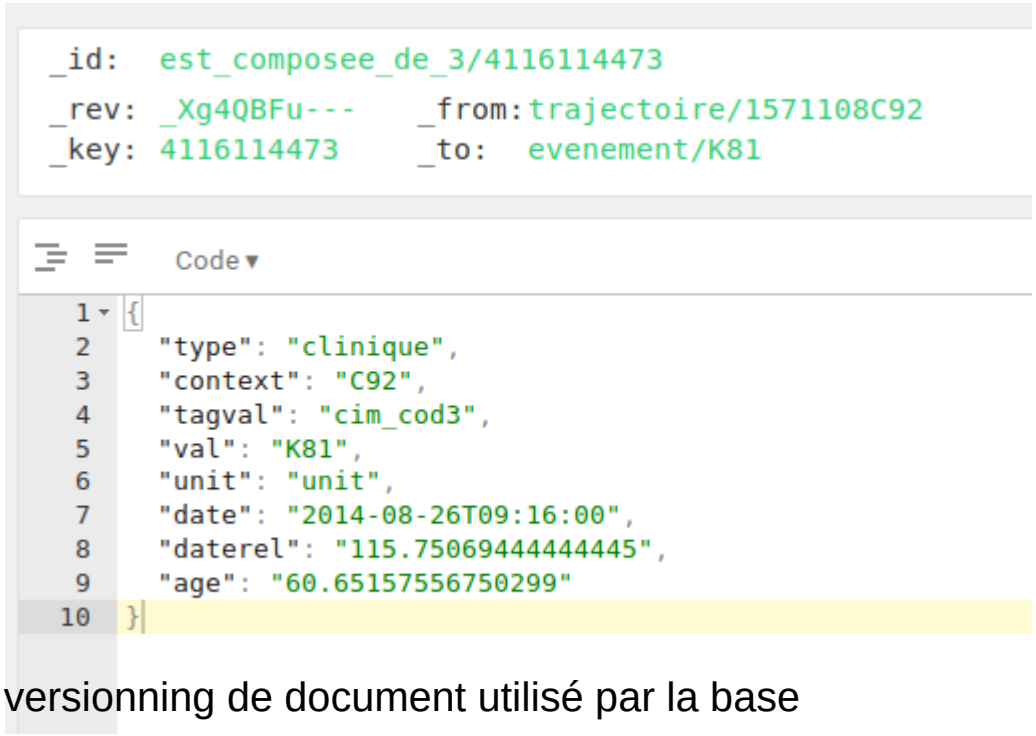


Monitoring


ArangoDB
COMMUNITY EDITION
DB: _SYSTEM
HEALTH: GOOD
DASHBOARD
COLLECTIONS
QUERIES
GRAPHS
SERVICES
USERS
DATABASES
LOGS
SUPPORT
HELP US
GET ENTERPRISE
3.1.19
Request Statistics
System Resources
Replication
current
15-min-avg
39
34.4
+0.0%
+1.3%
NUMBER OF THREADS
98.48 GB (78.31 %)
135.03 GB
MEMORY
current
15-min-avg
112.02
112.01
+0.0%
+0.0%
VIRTUAL SIZE IN GB
Major Page
0.2
0.15
0.1
0.05
0
11:52
11:54
11:56
11:58
12:00
12:02
12:04
12:06
12:08
12:10
MAJOR PAGE FAULTS
System Time
User Time
0.015
0.01
0.005
0
USED CPU TIME PER SECOND

Document JSON d'une collection Arangodb

- Formulaire d'un edge, modifiable, sauf les attributs built-in (préfixés par « _ »)



- _rev : versionning de document utilisé par la base

  => Choix des documents à répliquer entre des serveurs.
  => Fournir au client la dernière version d'un document à modifier

- Amélioration : prévoir un intitulé dans le sens inverse du lien

ArangoDB Query Language : AQL

- Langage propriétaire comparable à SQL, avec des fonctions supplémentaires de traitement des graphes https://docs.arangodb.com/3.3/AQL/

  Utilisation depuis l'interface web Arangodb, le dhell arangosh, l'API HTTP ou encore les microservices Foxx

- Jeu d'instructions pour les Requêtes « SQL-LIKE » https://www.arangodb.com/why-arangodb/sql-aql-comparison/

FOR: Iterate over all elements of an array.

RETURN: Produce the result of a query.

FILTER: Restrict the results to filtered elements

SORT: Force a sort of the array

LIMIT: Reduce the number of elements in the result

LET: Assign an arbitrary value to a variable.

COLLECT: Group an array by one or multiple group criteria.

REMOVE: Remove documents from a collection.

UPDATE: Partially update documents in a collection.

REPLACE: Completely replace documents in a collection.

INSERT: Insert new documents into a collection.

UPSERT: Update/replace an existing document, or create it in the case it does not exist.

WITH: Specify collections used in a query (at query begin only).

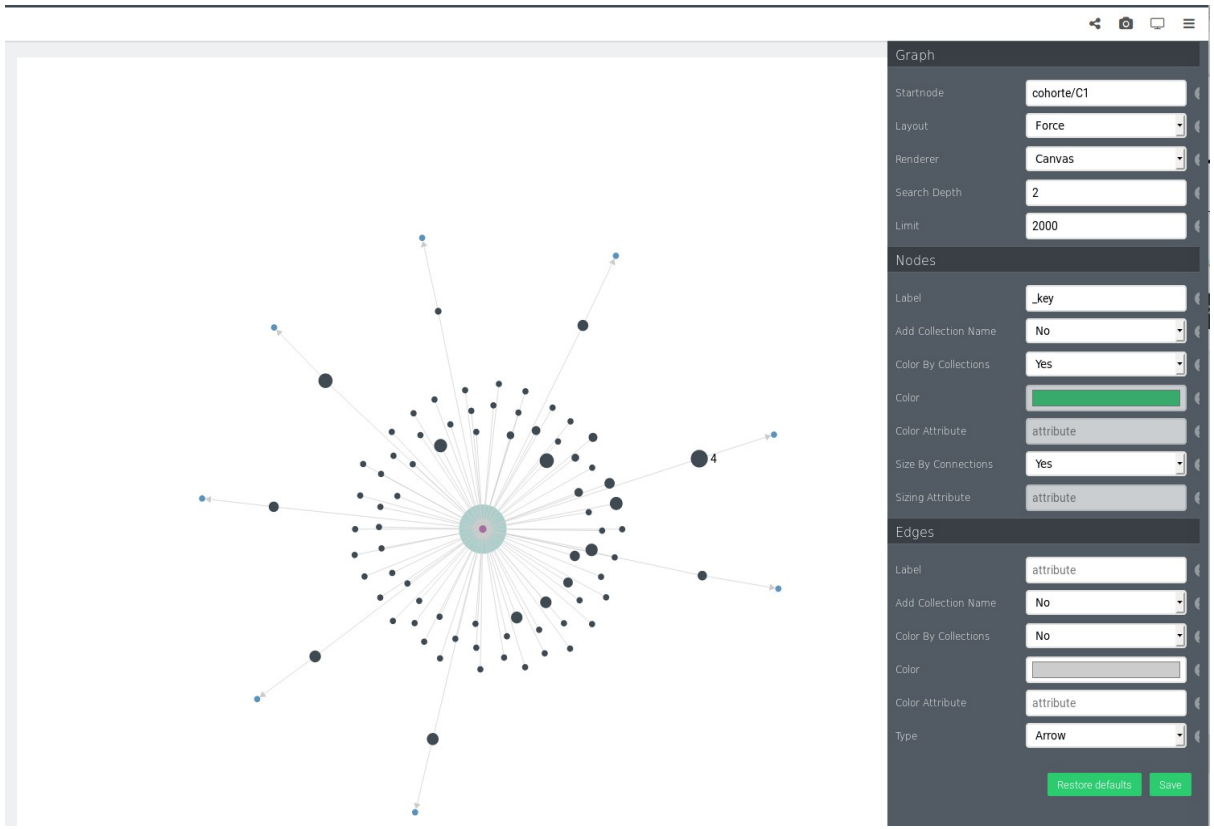
Query: detect_doublon_link

Queries | New | Save | Save as | Previous Query

```
1  FOR lien IN concerne_4
2
3    COLLECT cible=lien._to, source= lien._from WITH COUNT INTO length
4    SORT length
5    RETURN {
6      "source" : source,
7      "cible" : cible,
8      "count" : length
9    }
10
```

- Fonctions courantes chaînes, dates...

Voir graphiquement les données



Viewer quantitativement limité dans arangodb → usage pour vérification locale
Alternative gephi pour voir les graphes massifs

ArangoDB Query Language : AQL

- Liste des objets connectés à un objet de la collection «cohorte», selon les arcs contenus dans la collection « est_membre_de » (ici les membres d'un groupe de patients)

  FOR x IN ANY @cohorte est_membre_de_1 OPTIONS {bfs: true, uniqueVertices: 'global'} RETURN x

- Requête de traversée de graphe multi-chemin depuis un objet (patient) calculant le nombre d'objets de chaque classe à une distance de 1 arc

  FOR v IN 1 ANY 'patient_did/501959' GRAPH "patient_did360"

  COLLECT classe=LEFT(v._id, FIND_FIRST(v._id, '/')) WITH COUNT INTO length

  RETURN {

- "classe" : classe,

  "count" : length}

  réponse en JSON:

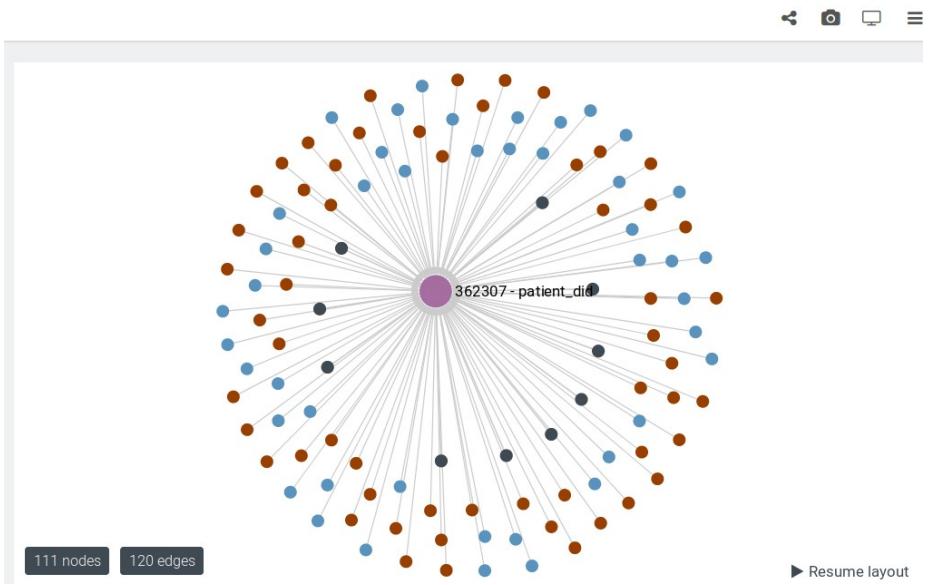
```
[{
    "classe": "diagnostique",
    "count": 148
 },
 {

   "classe": "document",
   "count": 80

 },
 {

 "classe": "venue_did",
 "count": 60
}]
```

réponse graphique :



Traversée de graphe en AQL

- Forme générique de la traversée de graphe

  https://docs.arangodb.com/3.4/AQL/Graphs/Traversals.html

- FOR x IN ANY 'cohorte/C1' regroupe OPTIONS {bfs: true, uniqueVertices: 'global'}

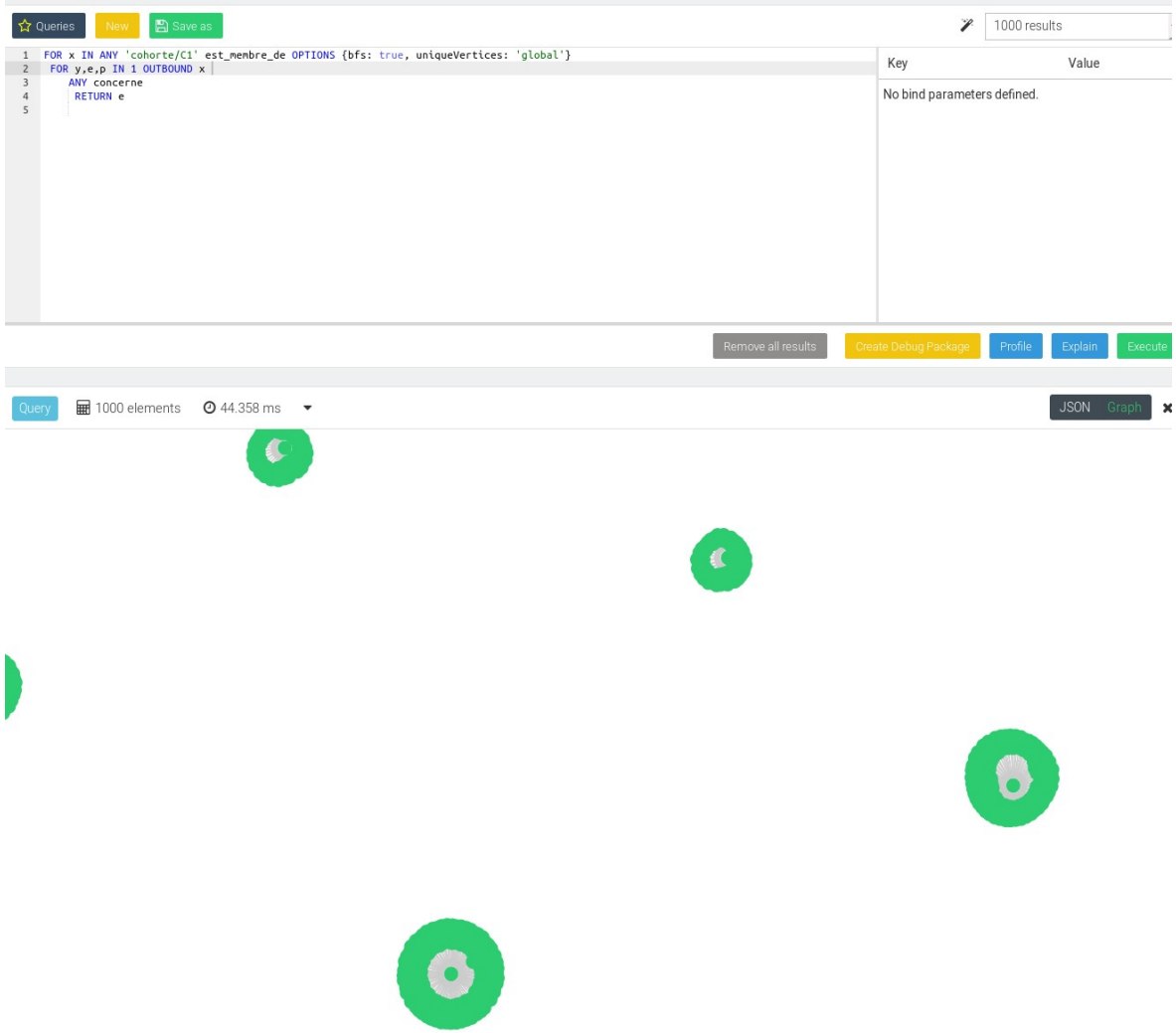
  FOR y,e,p IN 1 OUTBOUND x

  ANY concerne

  RETURN e

- On passe ici de la cohorte aux diagnostiques sans mentionner les patients
- Essayer alternativement avec y, puis y._key, puis p (path)

Liste des arcs « concerne » liant les diagnostics aux patients d’une cohorte

- 

Focus sur ArangoDB – Intégration

- Intégration - option 1 : Utiliser l'API HTTP standard dans le code

  curl --data @- -X POST --dump - http://localhost:8529/_api/cursor

  { "query" : "FOR u IN users LIMIT 2 RETURN u", "count" : true, "batchSize" : 2

  https://docs.arangodb.com/3.3/HTTP/AqlQueryCursor/AccessingCursors.html

- **Option 2 : Définir sa propre API pour faire abstraction de l'AQL dans les applications clientes.**

  Solution choisie pour implémenter un langage de navigation dans le graphe utilisable par utilisateurs métiers (graphique et cypher-like)

ArangoDB – Interface de développement Foxx

- API Foxx coté server HTTP REST JSON , accès à la data « in-memory »

  cloisonnement par contexte

  JS8 multi-threads

- Exemple

```
 router.get('/patdidsfromcohorte', function (req, res) {
 const stmt = db._createStatement({ "query": "FOR x IN ANY @cohorte est_membre_de_1 OPTIONS {bfs: true, uniqueVertices: 'global'} RETURN DOCUMENT(x._id)" });
stmt.bind('cohorte', req.param('cohorte'));
const c = stmt.execute();
res.json(c.toArray());
  })
  .summary("returns patdids who are member of cohort identifier parameter");
```

- Apport :

  encapsulation de l’API AQL => pas de diffusion dans l’écosysteme applicatif Pyhon, NodeJS, JS

  robustesse aux évolutions (un changement de base graphe par exemple)

  peu de code à maintenir

Tests de l’API

IHM

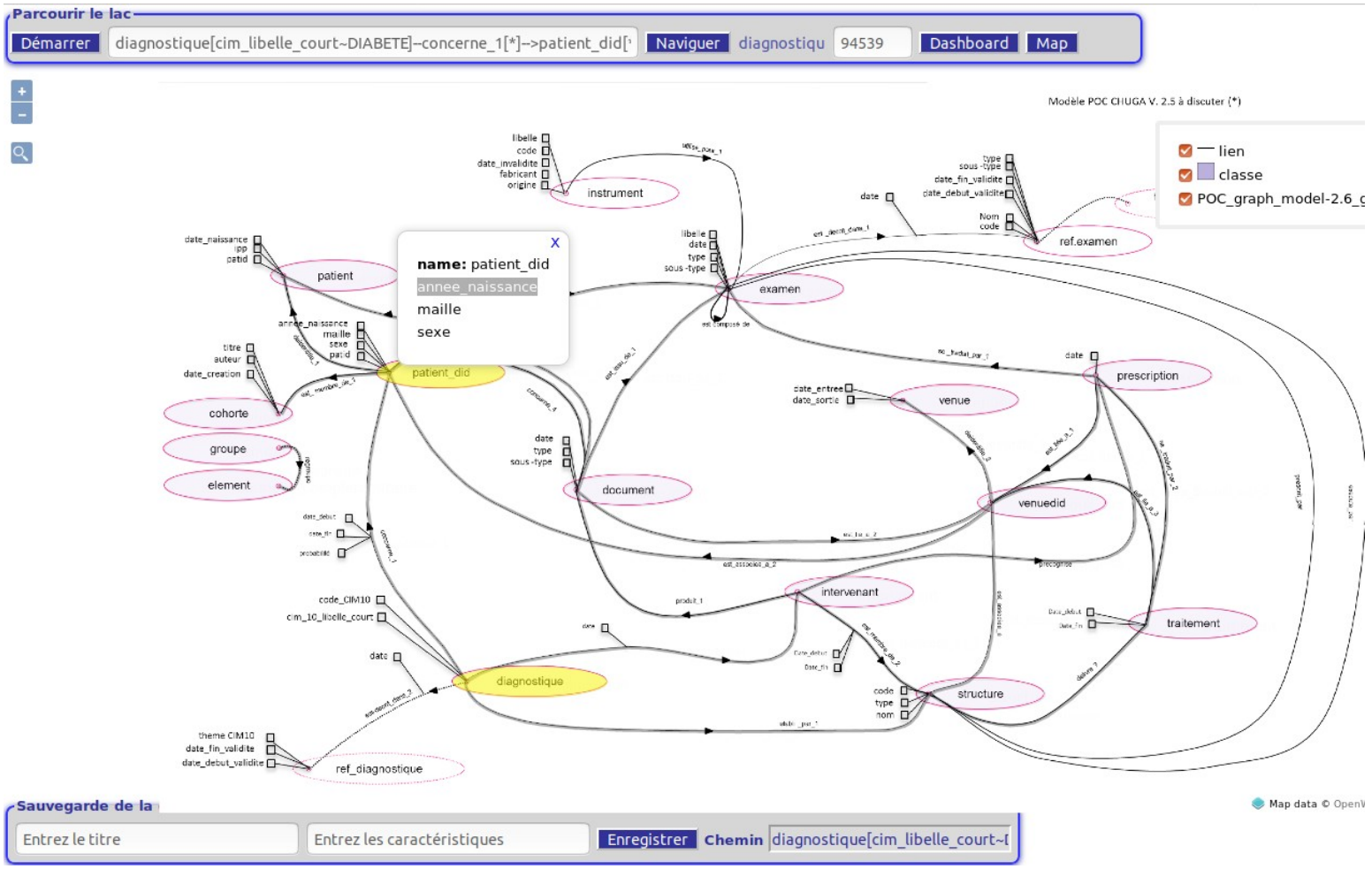
Apport des SIG et du Webmapping

- Intégration d’ Outils de Web mapping (Open Layer) pour générer la souche de l’interface interactive de navigation dans le Lac à partir de sa méta-structure définie dans QGIS (classe des classes, classe des liens, attributs associés).

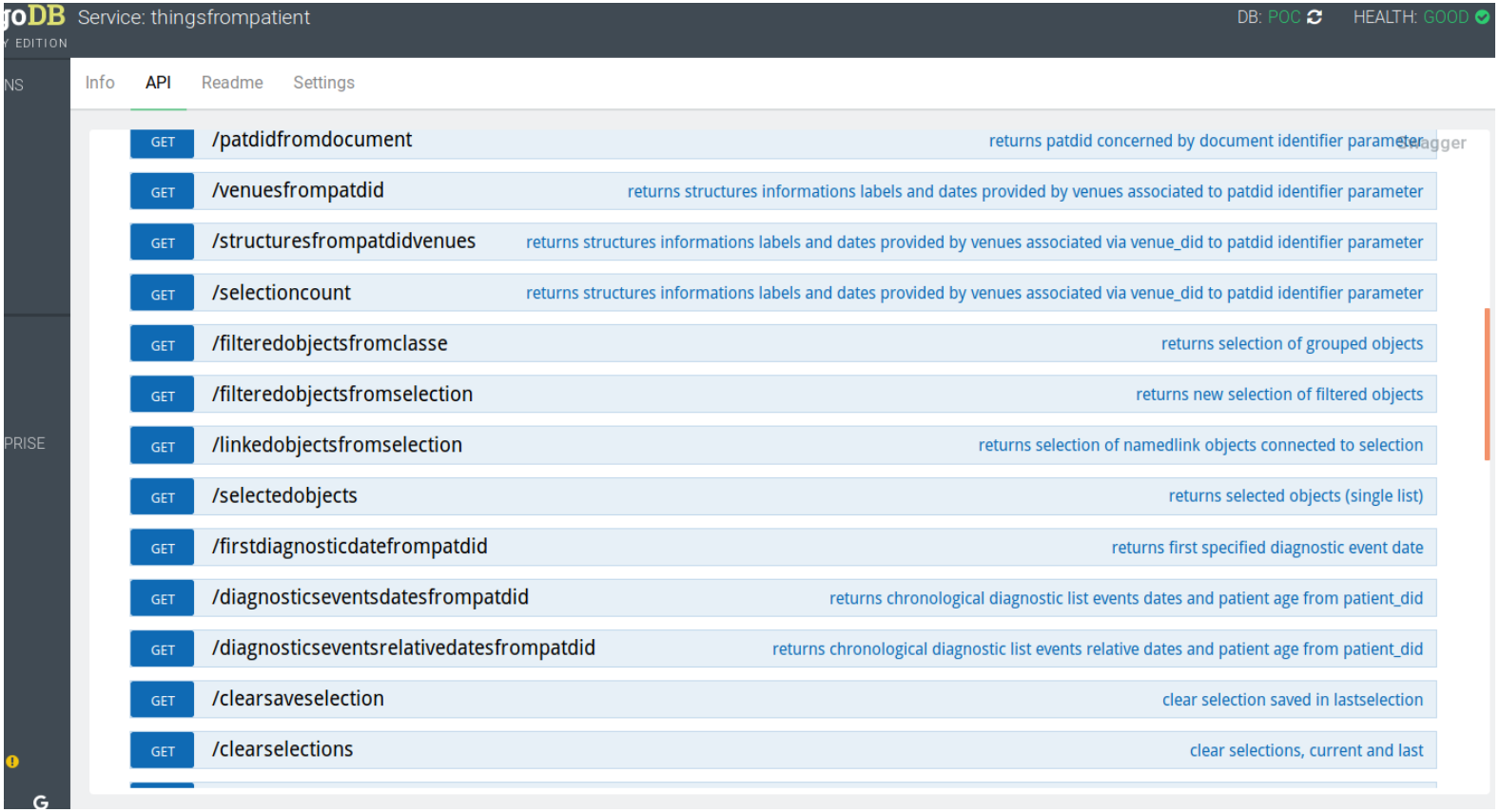
- Même solution pour générer la carte de répartition de cohorte à la volée.

- Economie de code

V. IHM : Navigation interactive dans le lac



L'API développée autorise l'interactivité de l'utilisateur



Exemples d'échanges avec d'autres applications

- Echanges avec les briques applicatives de traitement

  usage de web services HTTP REST JSON

  - ETL : ex composant talent « trest »

```
curl -X P0ST --data-binary @- --dump - http://datalaketest-11.imag.fr:8529/_api/document/diagnostic <<EOF
{
"ven_id" : "xxxx4564",
"pat_id" : "yyyy1684",
"type_diagnostic" : "DIAGNOSTIC PRINCIPAL",
"cim_cod : "zz.52"
}
```

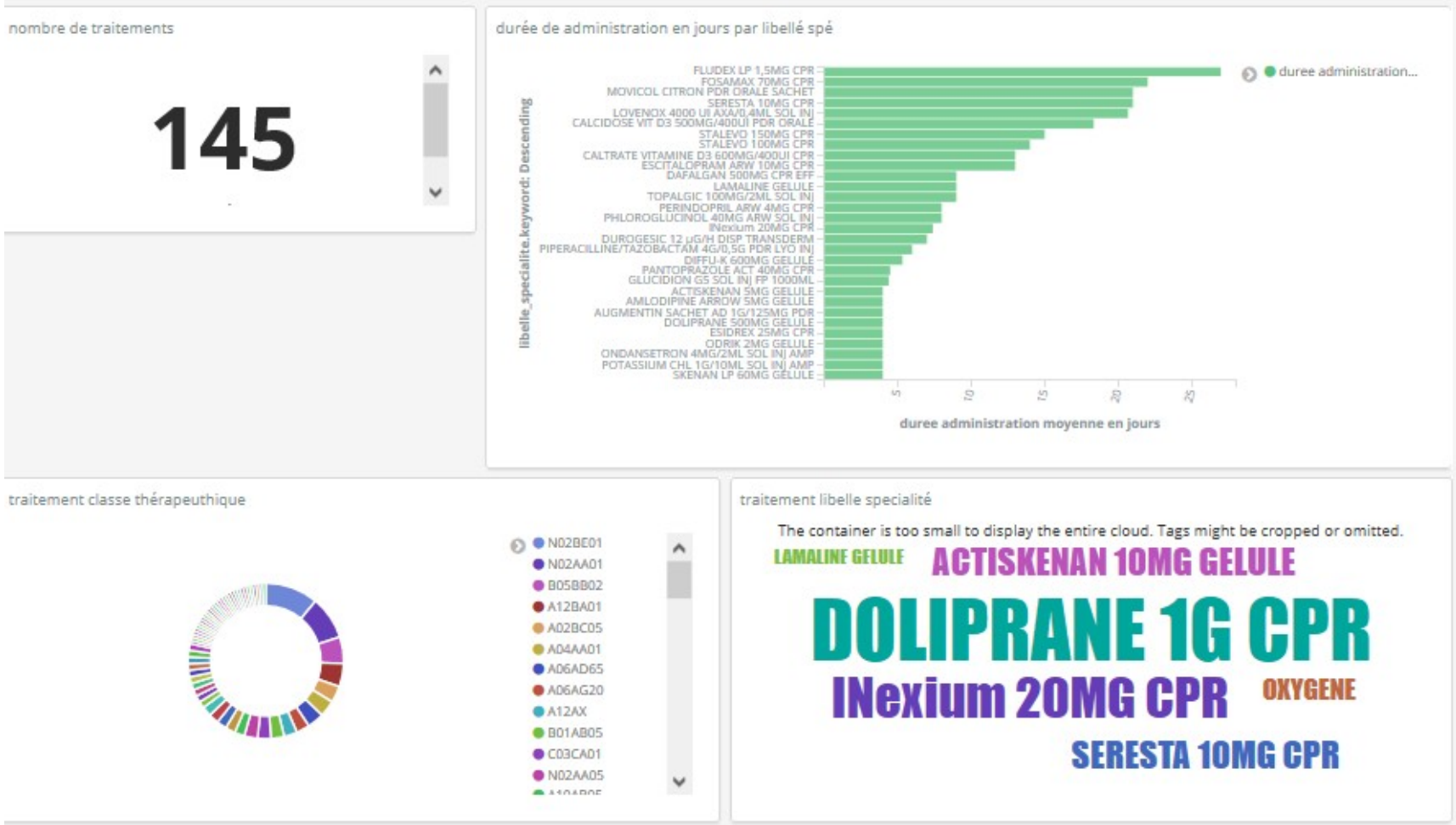
  - Moteur de recherche type Elastic : moteur de recherche, dataviz

```
var url = "http://elastic-13.imag.fr:9200/arango-"+jump.initialclassname+"/"+jump.initialclassname
req1.open("POST", url,false);
req1.setRequestHeader("Access-Control-Allow-Origin","*");
req1.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
req1.setRequestHeader("X-Requested-With","XMLHttpRequest");
req1.send(objtopost);
```

  - Environnement Spark - Hadoop ex : classification

http://160.8.x,x:8529/…./thingsfrompatient/addobjectingroup/?group=groupe/6385&object=patient_did/599571

VI. Projection à la volée dans la pile Elastic visualisation Kibana



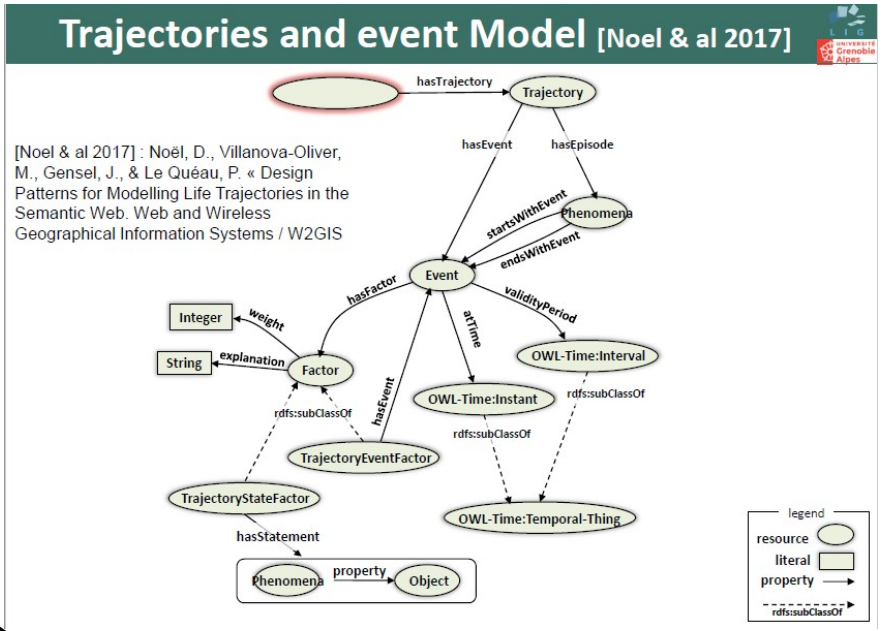
Elaboration de données métier

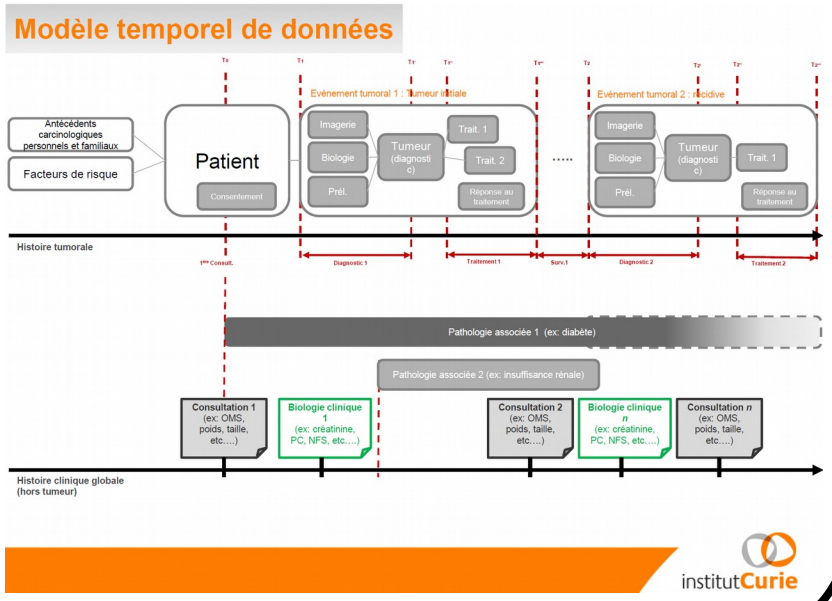
VII. Construction de données métiers élaborées :trajectoires

- Modélisation sous Arangodb de « **trajectoires »** composées **d'évènements** et de **phases.** Concept transversal, en cours d'exploration avec le CHUGA-IAB.

  A un patient est associé une trajectoire d'un point de vue clinique.
  Une trajectoire clinique est composée d'évènements cliniques (d'autres pourraient être pharmacologique, biologique, ou entrelacer des évènements de différents types) et d'éventuelles phases identifiables par les experts métiers.

- références





Elaboration de trajectoires cliniques dans la base graphe

- Développement Python + API foxx
- Phase 1 : Générer 1 trajectoire par patient et la relier aux évènements que l'on peut générer à partir de l'existant clinique dans du lac.
- Phase 2 : Générer des arcs dans une collection « précède » reliant chaque évènement d'une même trajectoire à son successeur → 80 000 Arcs générés. Quel outil de représentation ?
- Projection du résultat vers la pile Elastic

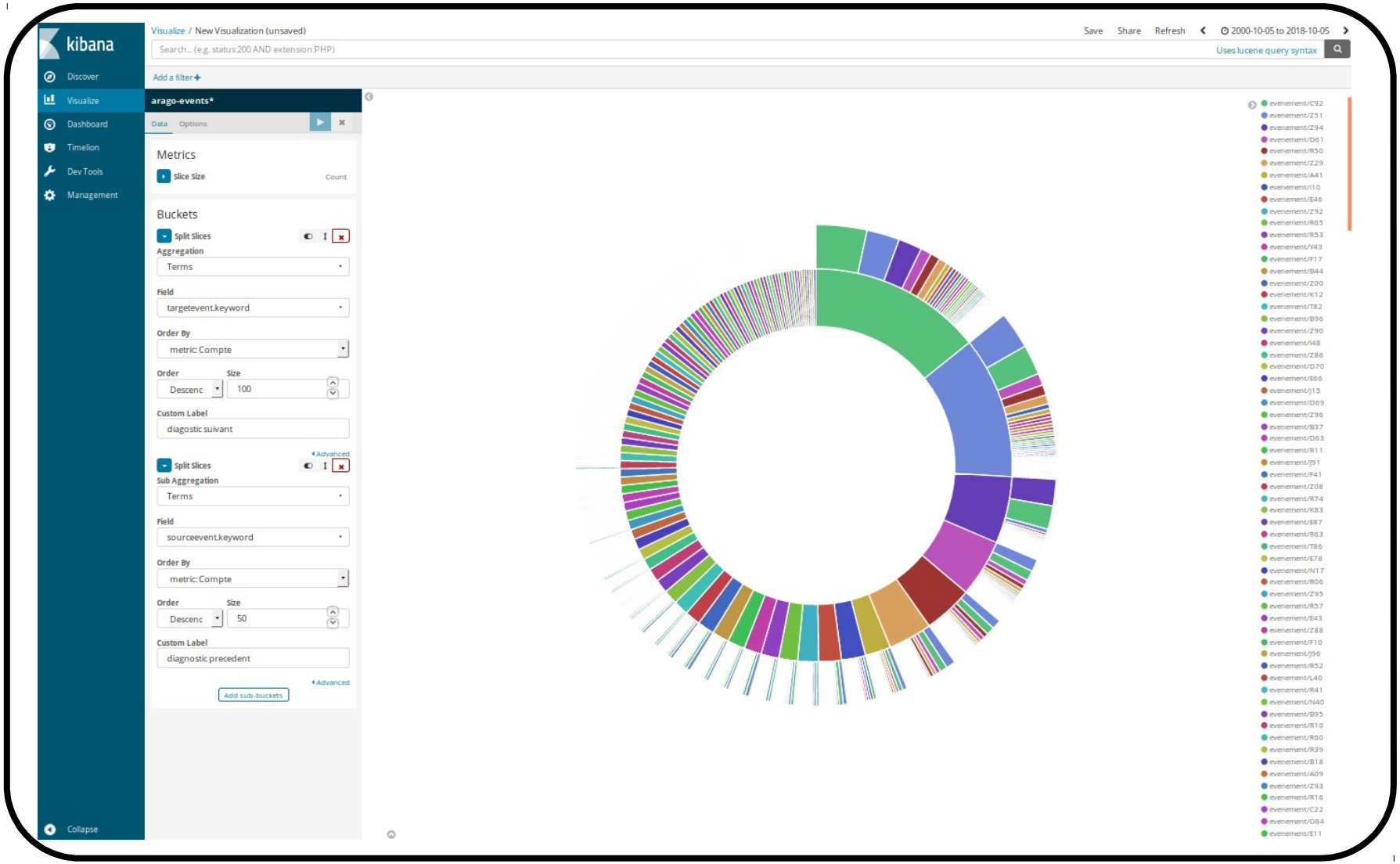
```
// python

def insertEventsLinkInElastic (sindex, svaleur) :

    url="https://simu:9200/"+sindex
    header = {"Access-Control-Allow-Origin" :"*", "X-Requested-With" : "XMLHttpRequest", "Content-Type" :
"application/json"}
    print (svaleur)
    params=svaleur
    req = requests.post (url, headers=header, data=params,auth=auth,verify=False)
    return
```

Résultat : agrégations concentriques sur les arcs « précède »

sur l'évènement suivant (sommet final de l'arc) – anneau intérieur -
puis sur l'évènement précédent (sommet initial de l'arc) – anneau extérieur)

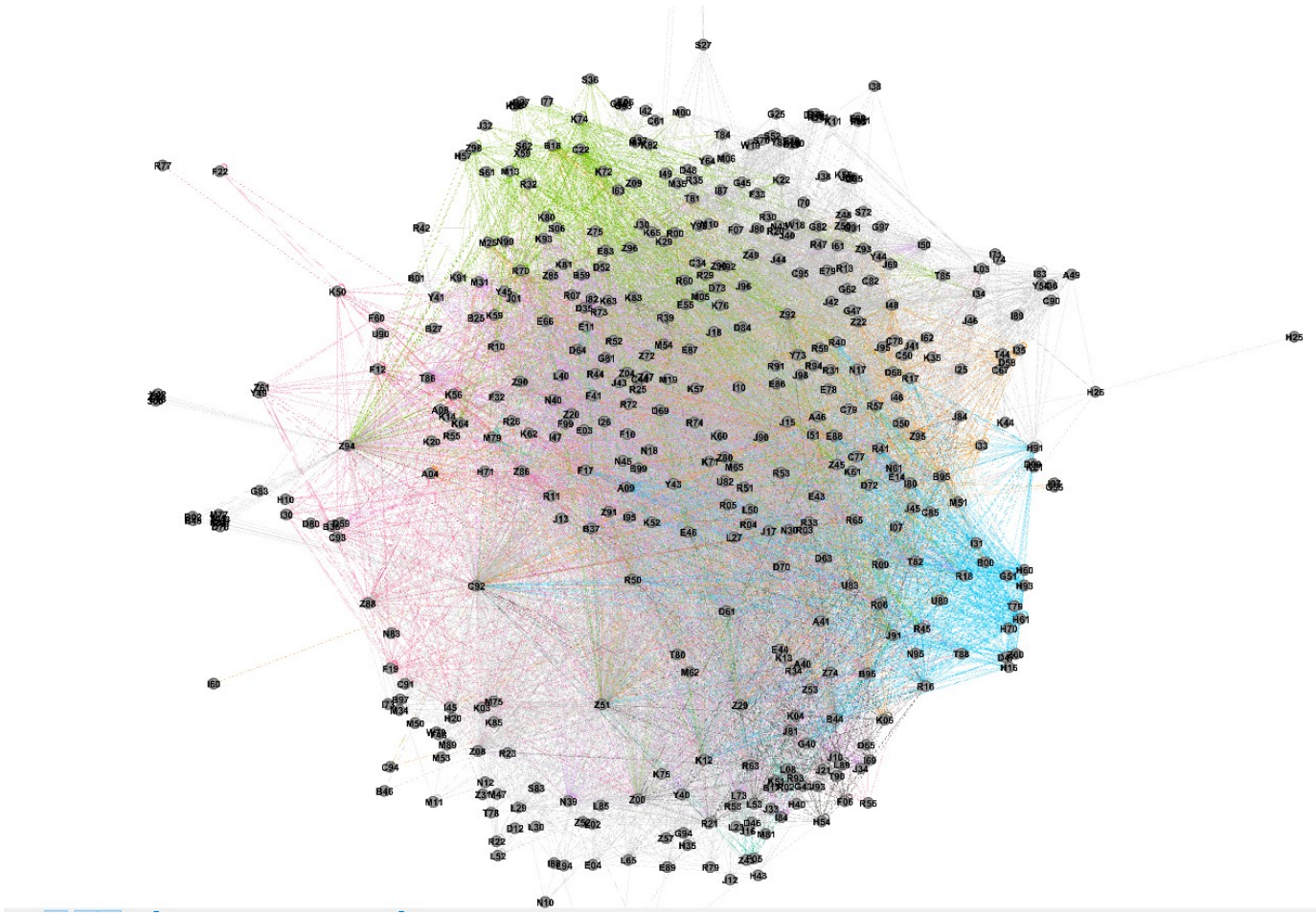


Visualisation et analyse de graphe

première approche

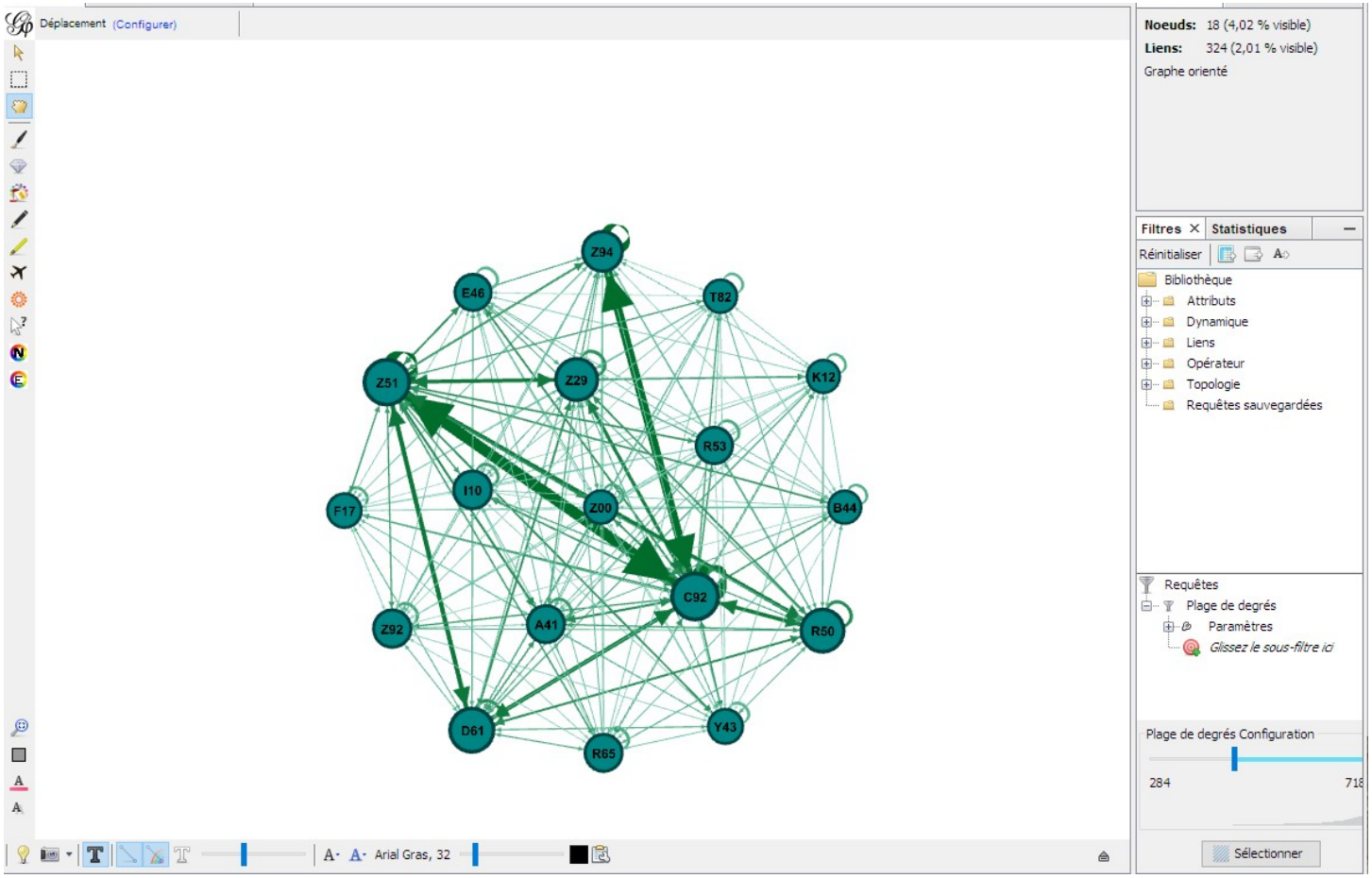
Visualisation de la relation « précède » avec Gephi

- 1 couleur correspond à une trajectoire individuelle



Agrégation des arc reliant les mêmes évènements

- Représentation des relations fréquentes de précédences entre des évènements



• Conclusion Intérêt, points d’attention arangodb (property graph)

- Capacité à rassembler des concepts proches : adaptation à l’exploitation de structures souples, évolutives (possibilités de variantes dans les structures d’un même ensemble).
- Capacité à naviguer rapidement dans des organisations complexes de données massives fortement connectées.
- Intéroperabilité API restful HTTP :capacité à distribuer les traitements sur différents applicatifs au delà du clustering classsique.
- JSON natif, nested: permet de prendre en compte des attributs complexes (listes, matrices,..)
- Robustesse de l’application
- Warnings :

  Fragilité de l’intégrité référentielle => bases graphe davantage adaptée à l’observation qu’à la gestion des données. Contrôles nécessaires (ex : dédoublonnage, topologie…)
  Implémentation de l’héritage : jouer sur la souplesse ou relier les sous ensembles ?
  Distinguer le graphe du modèle (structure du phénomène), du graphe des données (réalisations).

Merci pour votre écoute

... des questions ?

christophe.cance@univ-grenoble-alpes.fr

Bibliographie

- [10] Théorie des graphes, Stéphane Pelle ENSG

  http://cours-fad-public.ensg.eu/pluginfile.php/1525/mod_resource/content/1/Theorie_des_graphes.pdf
- [11] Éléments de théorie des graphes - Alain Bretto, Alain Faisant, François Hennecart
- [12] François Bouillé. Le modèle HBDS. ENSG 2013

  http://cours-fad-public.ensg.eu/mod/imscp/view.php?id=254
- [13] Pelle Stéphane. Documentation HBDS et UML 2006.http://pelle.stephane.free.fr/
- [14] Qwant et le machine learning, JRES 2017 - Sylvain Peyronnet

  https://www.jres.org/fr/videotheque?mode=replay&id=189&resolution=360
- [15] BERGE C, "Graphes et Hypergraphes", Edition Dunod, 1970
- [16] Ph. GENOUD, Web des données: *Les Principes-Les Standards du W3C* –Journée Interopérabilité et Innovation –IGN-BRGM-OGC -7 Octobre 2014 -Paris