

Annexes

Conception et développement d'une plateforme d'acquisition et de traitement de données relationnelles



Table des matières

Note : Ce document adopte la même structure que le rapport de stage afin de s'y retrouver plus facilement et pour la cohérence de la lecture.

1 Introduction

2 Contexte de recherche dans un laboratoire de sociologie

3 Analyse et conception

3.1 Objectifs du projet

3.1.1 Module d'optimisation réseau

3.1.2 Charte graphique et conception graphique

4 Développement et organisation

5 Conclusion

6 Bilan technique et personnel

Bibliographie

Figure 2.1-1 Organisation du LISST

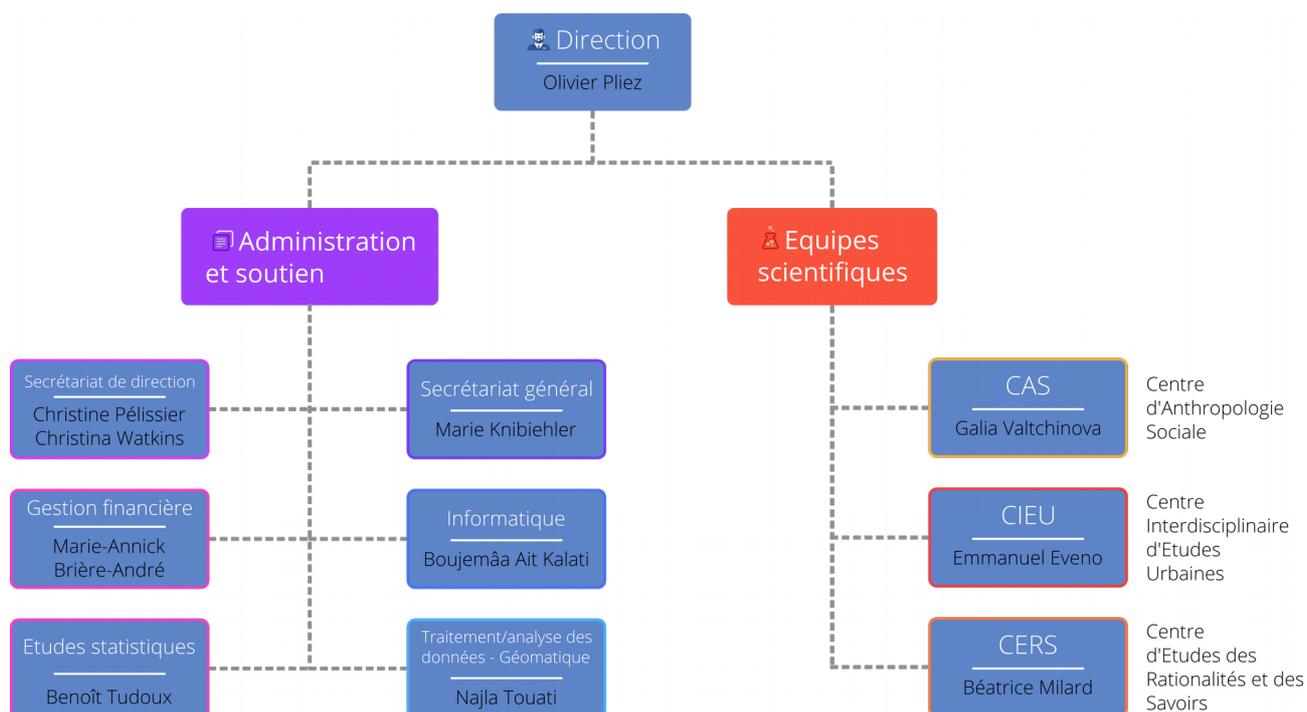


Figure 2.1-2 Description officielle de l'équipe LISST-CERS

Le CERS constitue l'équipe de sociologie fondamentale du LISST. Au-delà de tous les projets et activités dans lesquels les sociologues de l'équipe sont engagés aux côtés de leurs collègues du laboratoire (voir les axes du LISST), le CERS effectue un travail spécifique de sociologie qui se fonde sur des choix théoriques et épistémologiques : une sociologie soucieuse d'éviter toutes les formes de naturalisation du monde social, attentive à la pluralité des logiques d'action, et capable de mobiliser de façon cohérente des courants théoriques distincts mais rendus partiellement compatibles par leur insertion dans un cadre épistémologique plus général fondé sur l'articulation des échelles d'analyse.

Ce travail s'effectue dans le cadre d'activités de séminaire (« journée du Cers », « journées des doctorants », « séminaire du Cers ») et de recherches sur des thèmes émergents ou sur des aspects théoriques et méthodologiques liés à la discipline.

Figure 2.2-1 Organigramme des services communs de l'IRIT

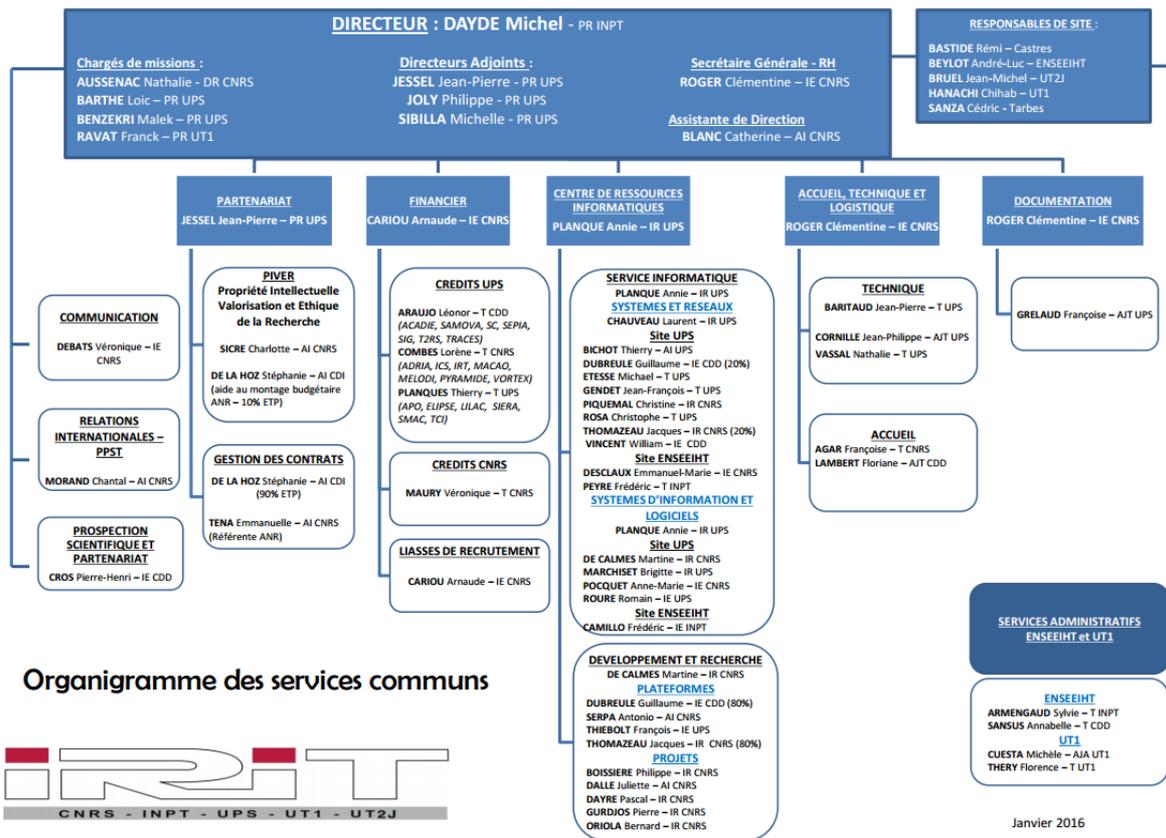
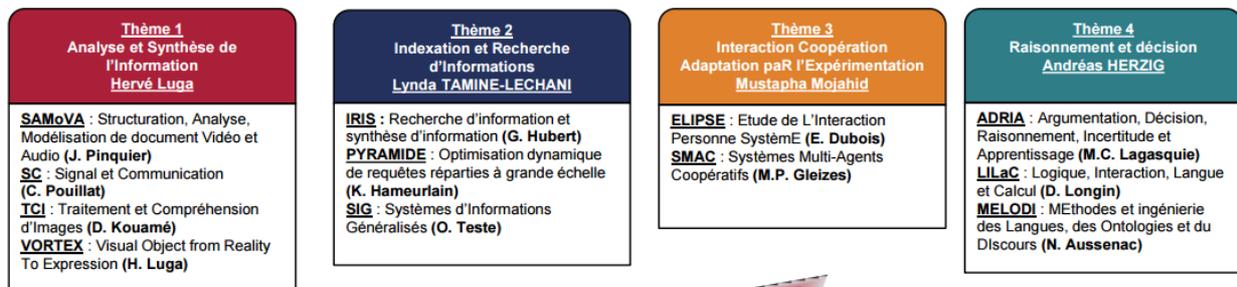


Figure 2.2-2 Organigramme de la Recherche à l'IRIT (équipes et axes)



7 thèmes – 21 équipes – 4 axes

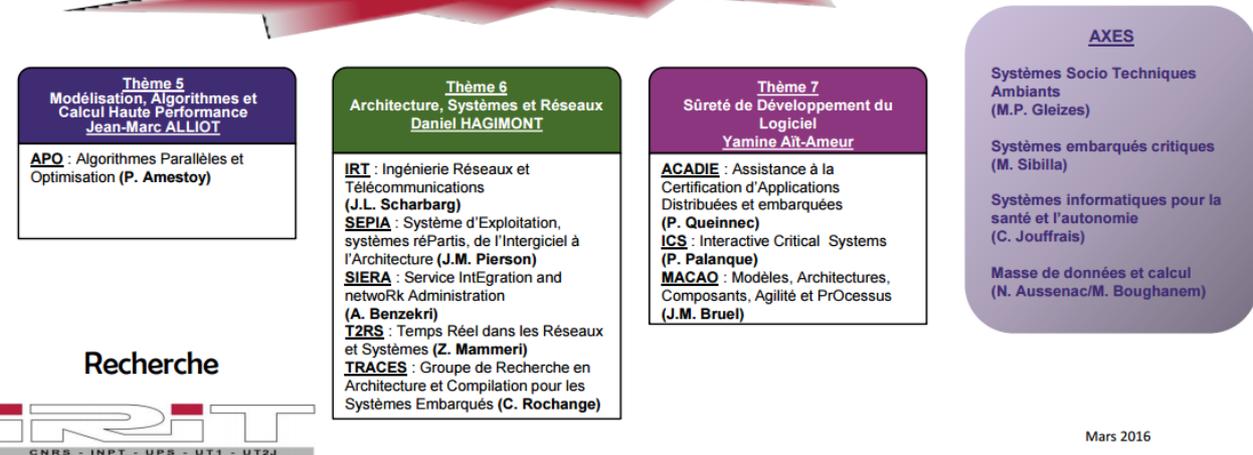


Figure 2.5-3 Tâches

Tâches					
TPE	Nom	Démaré	Terminé	Travail	Etat d'avancement
1	Analyse et Conception	Apr 4	Apr 11	18d 6h	
1.1	Analyse du système	Apr 4	Apr 7	5d	100%
1.2	Solutions techniques	Apr 5	Apr 11	7d 4h	100%
1.3	Conception base de données	Apr 6	Apr 11	6d 2h	100%
2	Version 0.1	Apr 8	Apr 15	14d	
2.1	Version 0.1.1 - Structure de base	Apr 8	Apr 13	12d 2h	
2.1.1	Choix charte graphique	Apr 8	Apr 12	4d 4h	100%
2.1.2	Structure de base du site	Apr 9	Apr 12	4d	100%
2.1.3	Choix réseau et optimisation	Apr 11	Apr 13	3d 6h	100%
2.2	Version 0.1.2 - Utilisateurs, Sécurité	Apr 14	Apr 15	1d 6h	
2.2.1	Gestion base de données	Apr 14	Apr 14	4h	100%
2.2.2	Prototype interface	Apr 14	Apr 14	4h	100%
2.2.3	Implémentation du système complet	Apr 14	Apr 15	6h	100%
3	Version 0.2	Apr 15	May 6	24d	
3.1	Analyse des journaux d'appels	Apr 15	Apr 16	1d 2h	100%
3.2	Passage 0.1	Apr 16	Apr 18	1d 2h	100%
3.3	Passage 1.0	Apr 18	Apr 19	1d 2h	100%
3.4	Prototype interface + choix ergonomiques	Apr 19	Apr 22	3d 6h	100%
3.5	Conception du formulaire dynamique	Apr 22	Apr 24	2d 4h	100%
3.6	Implémentation du formulaire dynamique 100%	Apr 25	May 2	9d	100%
3.7	Prologations : modifications	May 3	May 6	5d	100%
4	Version 0.3	May 7	May 19	17d 2h	
4.1	Conception format de sortie	May 7	May 9	2d 4h	100%
4.2	Choix du format de fichier compatible	May 7	May 9	2d 4h	100%
4.3	Implémentation de l'export	May 9	May 12	4d	100%
4.4	Conception d'un système d'accès optimisé	May 13	May 14	1d 5h	
4.4.1	Conception système d'accès	May 13	May 13	5h	100%
4.4.2	Implémentation système d'accès	May 13	May 13	3h	100%
4.4.3	Benchmark système d'accès	May 13	May 14	5h	100%
4.5	Adaptation au nouveau système	May 14	May 16	1d 2h	100%
4.6	Modification suite à entrevue	May 16	May 19	5d 3h	
4.6.1	Conception	May 16	May 17	1d 2h	100%
4.6.2	Prototype interface	May 17	May 18	1d 2h	100%
4.6.3	Modification interne	May 17	May 19	2d 7h	100%
5	Version 0.4	May 19	May 26	8d 2h	
5.1	Conception	May 19	May 20	1d 2h	100%
5.2	Prototype+Implémentation graphique #1	May 20	May 21	7h	100%
5.3	Prototype+Implémentation graphique #2	May 21	May 22	7h	100%
5.4	Prototype+Implémentation graphique #3	May 22	May 23	7h	100%
5.5	Prototype+Implémentation graphique #4	May 23	May 23	7h	100%
5.6	Prototype+Implémentation graphique #5	May 24	May 24	7h	100%
5.7	Prototype+Implémentation graphique #6	May 24	May 25	7h	100%
5.8	Prototype+Implémentation graphique #7	May 25	May 26	7h	100%
5.9	Prototype+Implémentation graphique #8	May 26	May 26	7h	100%
6	Version 0.4.1	May 26	Jun 1	7d 4h	
6.1	Conception réseau	May 26	May 27	1d 2h	100%
6.2	Prototype svg réseau	May 27	May 29	1d 2h	20%
6.3	Etude force atlas	May 29	May 30	1d 2h	0%
6.4	Développement	May 29	Jun 1	3d 6h	5%
7	Version 0.4.2	Jun 1	Jun 9	8d 6h	
7.1	Conception du système d'interaction	Jun 1	Jun 3	2d 4h	0%
7.2	Implémentation du système	Jun 3	Jun 8	5d	0%
7.3	Adaptation aux graphiques existants	Jun 6	Jun 9	1d 2h	0%
8	Version 0.5	Jun 9	Jun 13	3d 6h	
8.1	Création automatique de compte	Jun 9	Jun 10	1d 2h	0%
8.2	Système de modifications (limitation)	Jun 10	Jun 11	1d 2h	0%
8.3	Système de non-écrasement des données	Jun 11	Jun 13	1d 2h	0%
9	Version 0.6 - Debug+itération -1	Jun 13	Jun 23	12d 4h	0%

Figure 3.1.1-1 Schéma de fonctionnement du module d'optimisation réseau

Ci-dessous le schéma du fonctionnement du module d'optimisation réseau.

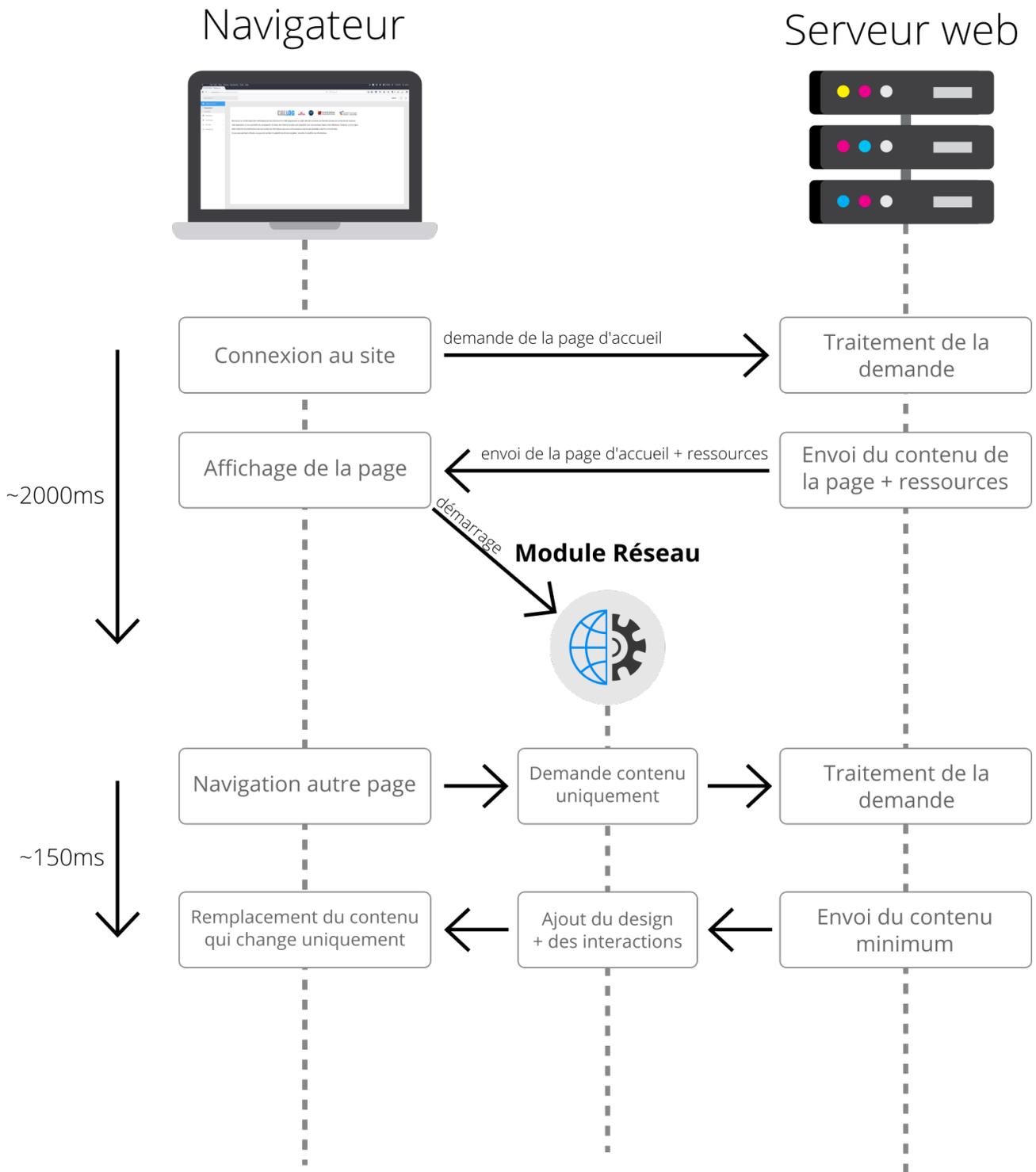
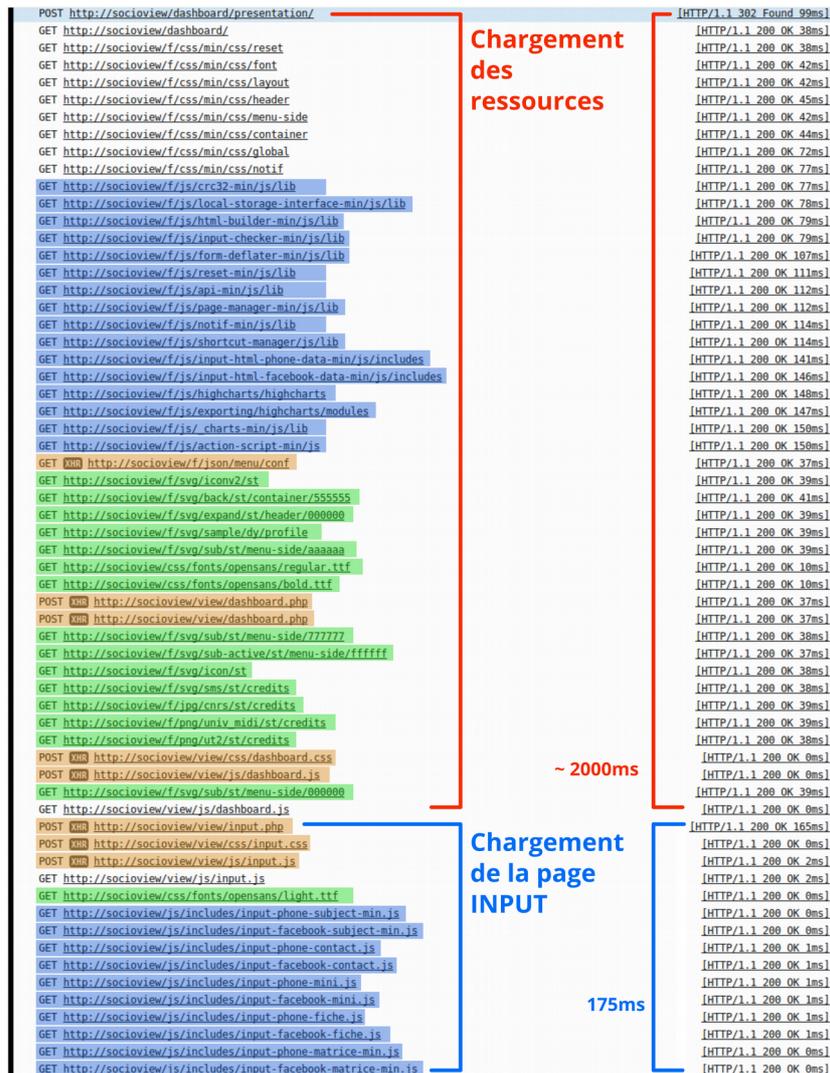


Figure 3.1.1-2 Analyse réseau détaillée

Ci-dessous la liste des fichiers chargés lors du chargement du site (en rouge), puis, ceux chargés lors de la navigation sur la page acquisition (en bleu).

- Image
- Chargement invisible (Ajax)
- Librairie

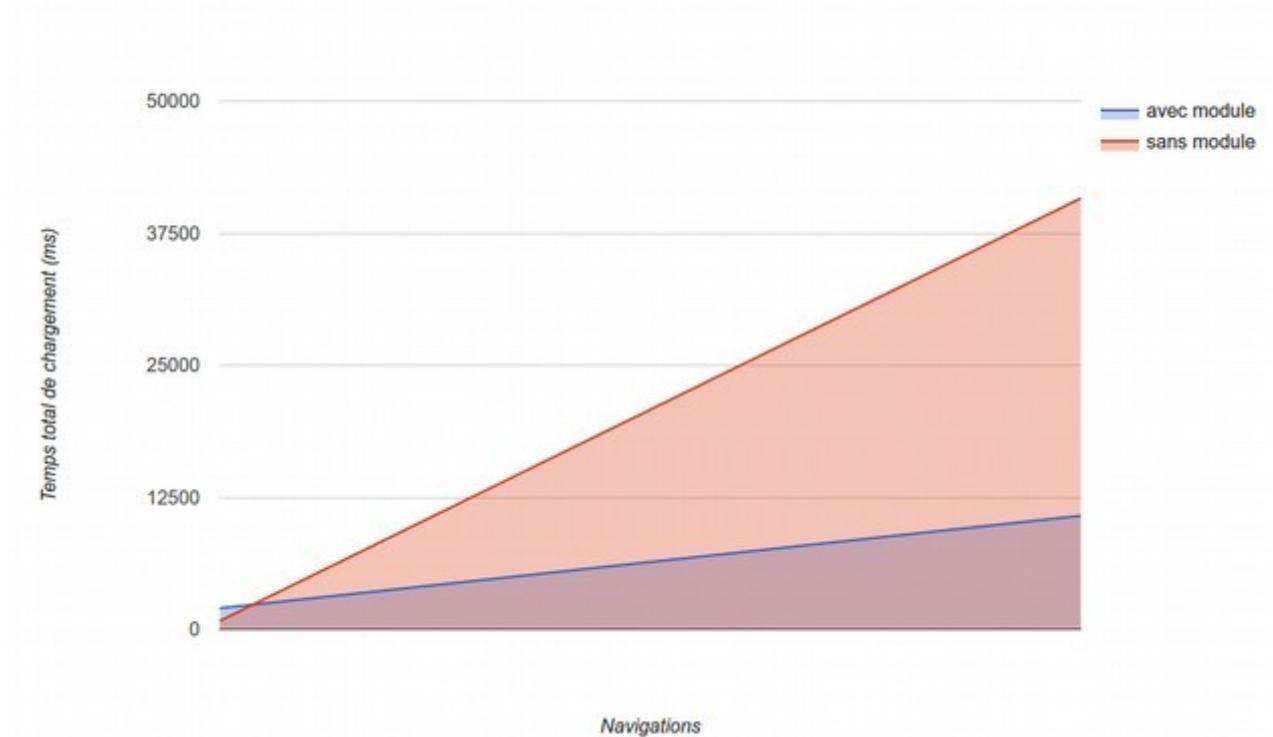


On remarque que le chargement du système complet prends aux alentours de 2 secondes, puis le chargement de chaque page est infime, de l'ordre de 0.175 secondes.

Note : La page acquisition (INPUT) est la plus conséquente du site car elle charge des librairies très lourdes (surlignées en bleu) pour gérer le nombre important d'interactions disponibles sur cette page. La page la plus lourde du site permet ainsi d'avoir une analyse avec approximation majorée.

Figure 3.1.1-3 Comparatif de vitesses de chargement

On peut voir sur le graphique ci-dessous l'efficacité de ce système (en bleu), comparé à un système normal (en rouge). Pour cela, nous avons supposé 0.8 secondes pour le chargement d'une page, bien que la moyenne en France pour le chargement d'une page soit de 6 secondes (selon l'Observatoire GESTE). Cette approximation majorée permet de vérifier – même dans de très mauvaises conditions – que le système est efficace.



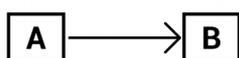
On remarque qu'à partir de 2 navigations (premier chargement inclus), le gain de temps est significativement supérieur à une utilisation normale.

3.1.2-1 Interactions et interconnexion du questionnaire

Vocabulaire du schéma	
Intitulé	Description
sujet	Formulaire permettant de sélectionner le sujet de l'enquête
contact	Champ de saisie d'un contact (pseudo et numéro de téléphone)
mini fiche	Fiche relation rapide à propos d'un contact (15 questions)
fiche	Fiche relation complète à propos d'un contact important (30 questions)
matrice	Matrice permettant de dire quel contact connaît quel(s) contact(s)

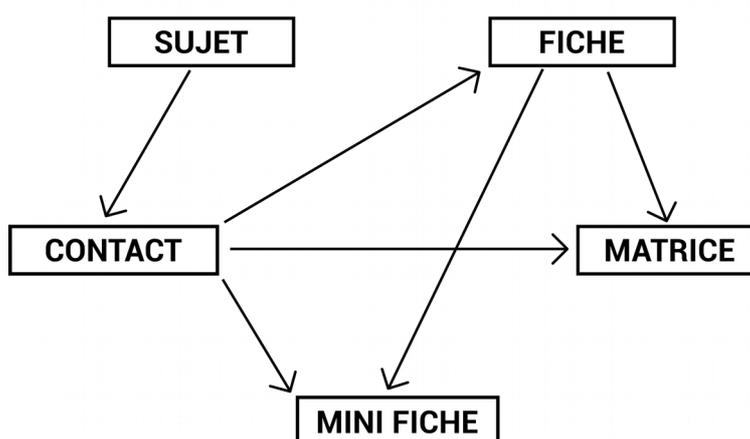
Ci-dessous le schéma d'inter-dépendance entre les entités.

Légende



B est lié à **A**

- si **A** change, **B** est mis à jour
- si **A** est créé, **B** est généré en fonction de **A**



Exemple pratique

Si on choisit un *contact* existant dans une autre enquête (soit George avec toutes ses données), on doit mettre à jour la *fiche* et la *mini fiche* pour récupérer les données de George. Ensuite la *matrice* devra se voir afficher le nom de George s'il y apparaît.

3.1.3-1 Documentation de l'API

NxTIC API
Admin-panels • NxTIC

Documentation Inspector

Aplary Powered Documentation
Sign in with Aplary account

Download API Blueprint

INTRODUCTION
Structure et fonctionnement
Types de données

REFERENCE
module
user
call_log
chart
token
subject
input
upload
download

NxTIC API

INTRODUCTION

API de la plateforme d'étude NxTIC, cette documentation présentera toutes les méthodes accessibles depuis la plateforme elle-même et depuis un logiciel tiers. La plateforme NxTIC est une plateforme d'étude sociale développée par Adrien Marqués (*adrm-brackets*) pour un laboratoire de sociologie du CNRS. Elle a pour objectif l'acquisition, la visualisation et l'extraction de données relationnelles.

Cette plateforme est temporairement hébergée sur <https://socioview.xdrmlab/>.

Structure et fonctionnement

Le fonctionnement est basé sur une délégation à 2 niveaux : des **modules** contenant des **méthodes**.

Paramètres

Tous les paramètres doivent être envoyés en `multipart/form-data`.

- Chacun formaté en `json`
- Portant le `nom` défini dans la documentation
- L'ordre n'a pas d'importance
- Respectant le `type` défini dans la documentation (cf. Types)

Note: Les paramètres `URI` ne correspondent pas aux paramètres URL. Ils servent à expliciter les paramètres et leurs types, et correspondent aux variables notées `{nomVar}` dans le corps de la requête.

Réponses

A. Les réponses seront formatées en json et contiendront:

- `ModuleError` - Le code de l'erreur
- `ErrorDescription` - La description de l'erreur

B. Codes `HTTP` et leur signification.

Status	Code HTTP
OK	200 - Success
Erreur	417 - Erreur quelconque

Types de données

Types Simples

Type	Exemple	Description
<code>mixed</code>	<code>[9, "a"]</code> , <code>"a"</code>	Type quelconque (peut être simple ou composé)
<code>id</code>	<code>10</code> , <code>"23"</code>	Nombre entier positif compris entre 0 et 2147483647
<code>text</code>	<code>"Hello!"</code>	Chaîne de caractères de longueur quelconque (peut être vide)
<code>mail</code>	<code>"a.b@c.def"</code>	Adresse mail avec une syntaxe valide
<code>number</code>	<code>0102030405</code>	Numéro de téléphone valide suivant les formats : <code>06</code> , <code>+336</code> , <code>+33 6</code>
<code>array</code>	<code>[1, 2]</code>	Tableau quelconque non vide
<code>boolean</code>	<code>true</code> , <code>false</code>	Booléen
<code>varchar(a,b)</code>	<code>"Hello!"</code>	Chaîne de caractères de taille comprise entre <code>a</code> et <code>b</code> (inclus)

Type composé : array

Type	Sous-Type	Description
<code>array<mixed></code>	<code>mixed</code>	Tableau contenant uniquement des données de type <code>mixed</code>
<code>array<id></code>	<code>id</code>	Tableau contenant uniquement des données de type <code>id</code>
<code>array<text></code>	<code>text</code>	Tableau contenant uniquement des données de type <code>text</code>
<code>array<mail></code>	<code>mail</code>	Tableau contenant uniquement des données de type <code>mail</code>
<code>array<number></code>	<code>number</code>	Tableau contenant uniquement des données de type <code>number</code>
<code>array<array></code>	<code>array</code>	Tableau contenant uniquement des données de type <code>array</code>
<code>array<boolean></code>	<code>boolean</code>	Tableau contenant uniquement des données de type <code>boolean</code>
<code>array<varchar(a,b)></code>	<code>varchar(a,b)</code>	Tableau contenant uniquement des données de type <code>varchar(a,b)</code>

Note: Il est possible de chaîner le type `array` autant de fois que nécessaire.
Ex.: `array<array<id>>` - Soit un tableau contenant des tableaux contenant exclusivement des données de type `id`.

3.1.3-1 Apiary : une documentation interactive

Sur la capture ci-dessous, on peut voir dans le panneau de gauche, un « exemple » d'utilisation. Cela permet de savoir exactement ce que l'on doit envoyer, et ce que l'on doit recevoir. Il est possible en cliquant sur le bouton « Try » de lancer un test sur le site pour vérifier si tout fonctionne bien.

The screenshot displays the Apiary documentation interface for the 'subject' API. The left sidebar contains a navigation menu with 'subject' selected. The main content area shows the 'search' endpoint with the following details:

- search**: Recherche d'un sujet par nom. *Description de la fonctionnalité*
- Permissions: admin. *Permission nécessaire*
- getFriends**: Retourne les informations de tous les contacts renseignés par un sujet d'enquête.
- Permissions: admin
- getAll**: Retourne les informations de tous les sujets.
- Permissions: admin
- create**: Creation d'un nouveau sujet.
- Permissions: admin

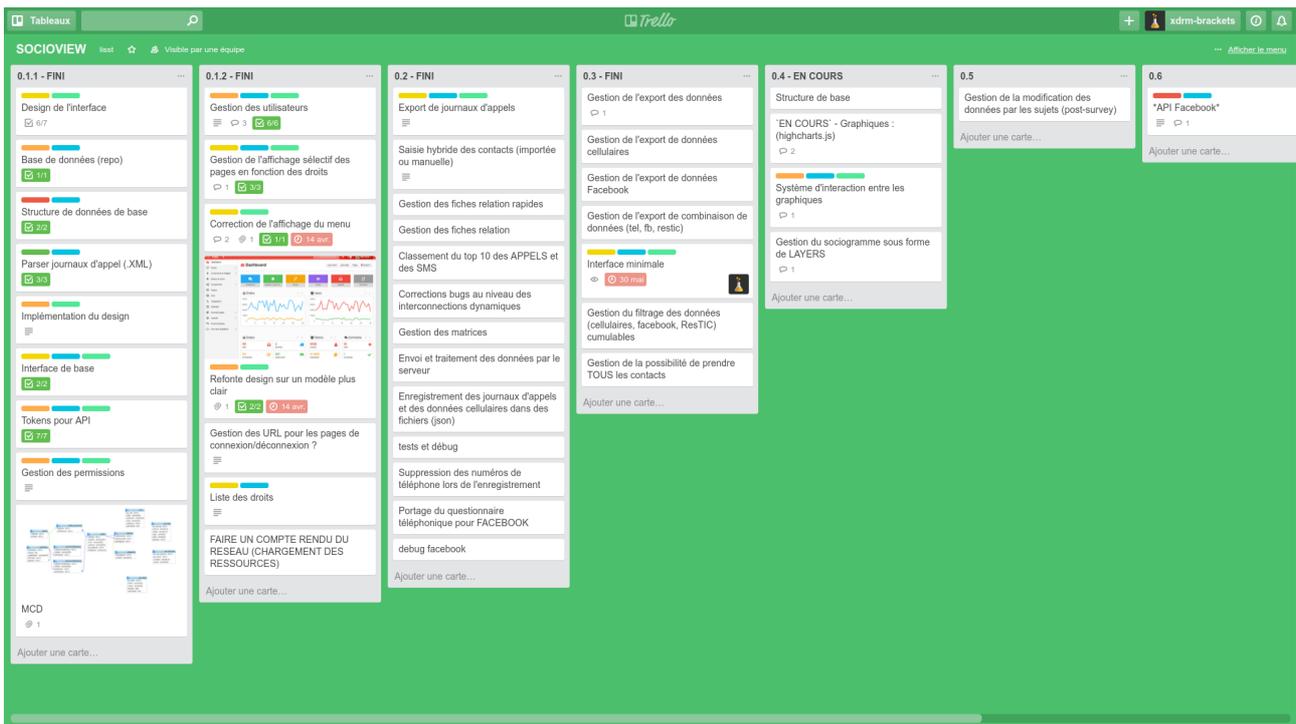
The right sidebar shows the request and response details for the 'search' endpoint:

- Request** (REQUETE):
 - Method: POST
 - URL: <https://soctiview.xdrm.fo/api/subject/search> *Lien ou envoyer les données*
 - Parameters: name (Le nom du sujet, Verchar(1,50))
 - Headers: Content-Type: multipart/form-data; boundary=xxxBOUNDARYxxx, Authorization: Digest {yourAccessToken}, Cache-Control: no-cache
 - Body:

```
01 --xxxBOUNDARYxxx
02 Content-Disposition: form-data; name="name"
03 Content-Type: application/json
04
05 {name}
06 --xxxBOUNDARYxxx--
```
- Response** (REPONSE):
 - Status: 200
 - Headers: Content-Type: application/json
 - Body:

```
JSON SCHEMA
01 {
02   "results": @results
03 }
01 {
02   "results": @results
03 }
```

4.1.3-1 Tableau Trello



On peut voir sur le tableau Trello ci-dessus, qu'il est composé de plusieurs colonnes. Chaque colonne correspond à une version (étape) du développement. Et chaque étape est composée de tâches pouvant être complétées d'images, de commentaires, etc.

4.2.1-1 Liste des versions

Version	Sous-version	Libellé	%
0.1	0.1.0	Structure et base de la plateforme	■
	0.1.1	Navigation et charte graphique	
	0.1.2	Sécurité et utilisateurs	
0.2	0.2.0	Acquisition des données	■
	0.2.1	Prototype de l'interface et choix ergonomiques	
	0.2.2	Report du questionnaire ResTIC	
	0.2.3	Contenu 100% dynamique	
	0.2.4	Corrections, optimisation et modifications	
	0.2.5	Supplément modifications	
0.3	0.3.0	Export des données	■
	0.3.1	Format, mise en forme des données	
	0.3.2	Conception d'un système d'accès aux données	
0.4	0.4.0	Visualisation graphique	■
	0.4.1	Réseau	
	0.4.2	Système d'interaction	
0.5	0.5.0	Modification par les sujets	■
	0.5.1	Création automatique de compte	
	0.5.2	Restitution partielle des données et modification	
0.6	0.6.0	Récupération des données LabSurveys	■
0.7	0.7.0	Pré-production	■
	0.7.1	Refonte et nettoyage du design	
	0.7.2	Élaboration des tests unitaires	
	0.7.3	Optimisation du chargement	

Figure 4.2.1-1 Base de données

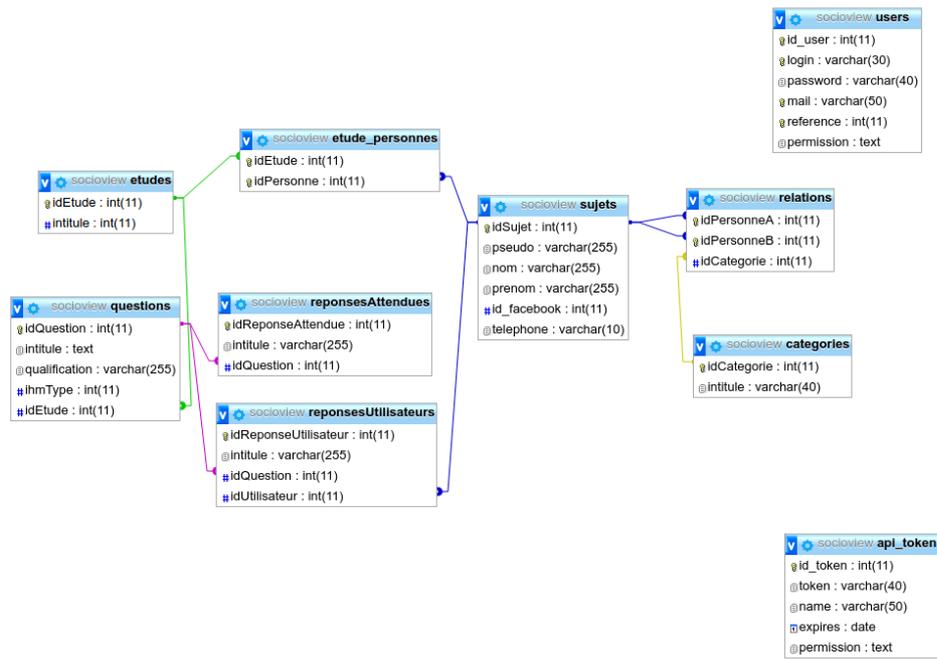


Figure 4.2.1-2 Premier prototype de la charte graphique

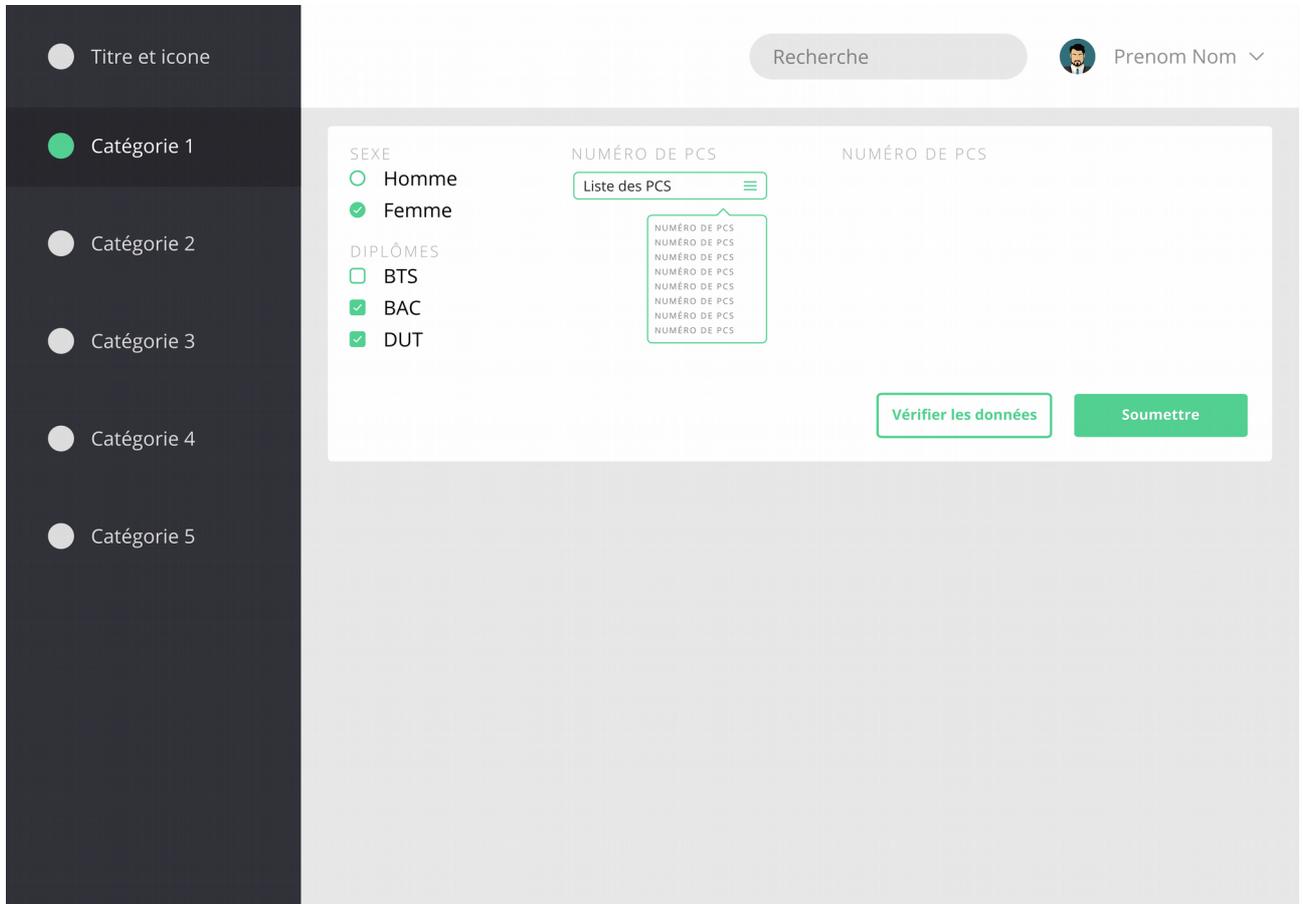


Figure 4.2.1-3 Framework par fichiers

Router	Gestion du chargement des pages en fonction de l'URL
Database	Gestion de la connexion à la base de données ainsi que l'accès
Repo	Gestion des requêtes pour la base de données
ManagerError	Gestion des différentes erreurs
MenuManager	Gestion de la création du menu en fonction des permissions
ModuleRequest	Gestion des requêtes de l'API
ModuleResponse	Gestion des réponses de l'API
ResourceDispatcher	Gestion du chargement des ressources (images, design, librairies, etc)
sessionManager	Gestion de la sécurité des sessions
lightdb	Système d'accès aux données optimisé pour le projet

Figure 4.2.1-4 Page de gestion des utilisateurs

The screenshot shows a web browser window displaying the 'Gestion des utilisateurs' page. The page title is '- GESTION DES UTILISATEURS -'. The main content area is titled 'Gestion des utilisateurs' and includes a sub-header 'Ajouter un utilisateur'. Below this, there is a form for creating a new user with the following fields: 'Identifiant', 'Adresse mail', 'Mot de passe', 'Vérification du mot de passe', 'Droits' (with radio buttons for 'Sujet' and 'Administrateur'), and 'Identifiant d'étude (optionnel)'. A 'Créer un utilisateur' button is located below the form. Below the form, there is a list of existing users, each with a profile icon, a role label (ADMINISTRATEUR or SUJET), and a 'Supprimer' button. The list contains four users: 1. Identifiant: julien.fgeac, Adresse mail: julien.fgeac@gmail.com, Role: ADMINISTRATEUR. 2. Identifiant: Buser, Adresse mail: B@gmail.com, Role: SUJET. 3. Identifiant: Auser, Adresse mail: A@gmail.com, Role: SUJET. 4. Identifiant: Cuser, Adresse mail: C@gmail.com, Role: ADMINISTRATEUR.

Figure 4.2.1-5 Page de connexion

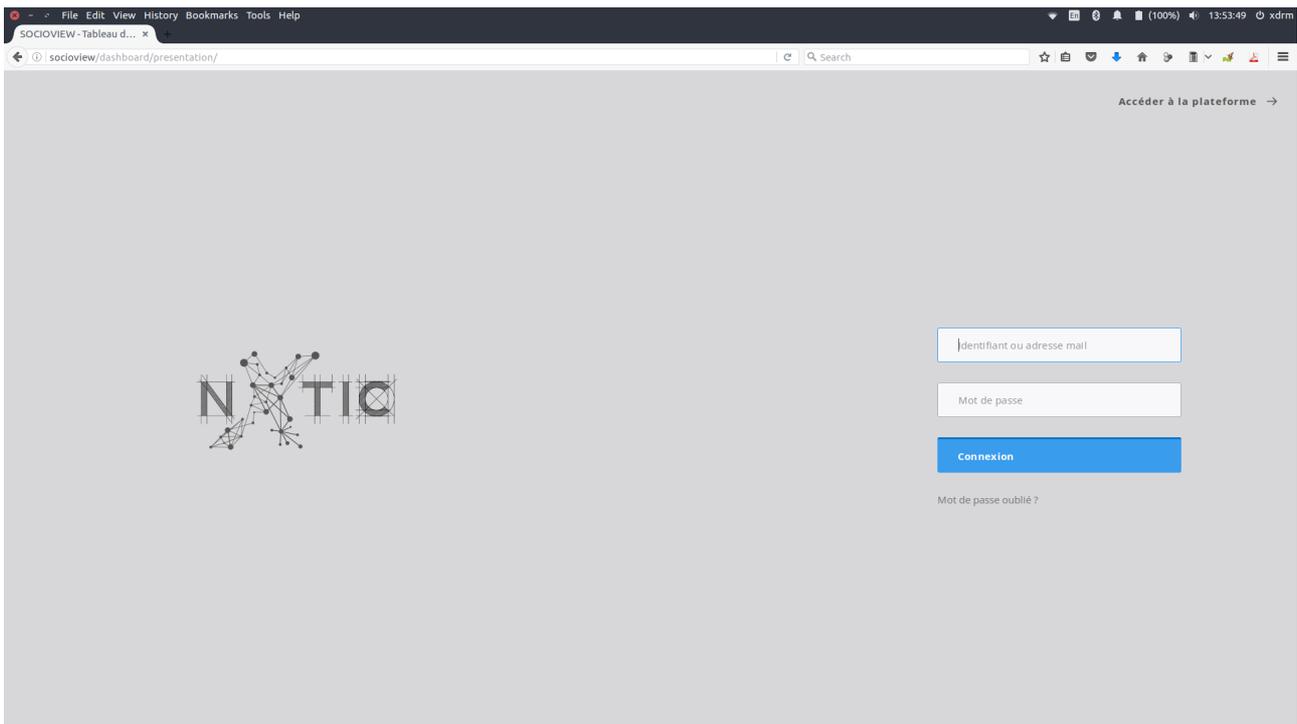


Figure 4.2.3-1 Benchmark de *lightdb*

Analyse de performances - `lightdb` - Introduction

PRESENTATION	
Ce document représente l'étude des performances du système de données développé pendant le stage.	
FEUILLES	
Performances	Résultats des différents tests
Individual Speed	Temps d'exécution de chaque requête
Cluster Speed	Temps d'exécution de chaque requête en regroupant les requêtes en une seule
Cluster Insert	Différence de temps d'exécution entre les requêtes individuelles et groupées d'insertion
Cluster Fetch	Différence de temps d'exécution entre les requêtes individuelles et groupées d'accès
Cluster Delete	Différence de temps d'exécution entre les requêtes individuelles et groupées de suppression

Analyse de performances - 'lightdb' - Performances

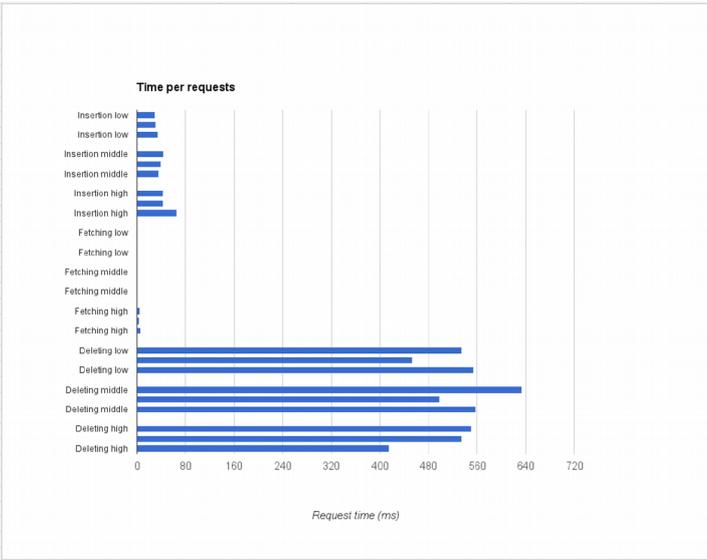
Action	Sample	Length (char)	Total length (char)	Total time (sec)	Unit time (sec)	Character / sec	Unit time (ms)	Character / ms	Remarques	CLUSTER	
INSERTION											
Insertion low	1000	201	201000	29,93966293	0,02993966293	0,0001489525469	29,93966293	0,1489525469	L'écriture étant l'opération la plus longue,	Cluster Insertion low	0,01045703888
Insertion low	1000	201	201000	30,8899501	0,0308899501	0,0001537261199	30,8899501	0,1537261199	dans le projet, j'insère enregistrement de données,	Cluster Insertion low	0,009452104568
Insertion low	1000	201	201000	36,2580249	0,0362580249	0,0001705817895	36,2580249	0,1705817895	mais en une seule requête (un join que je parsie)	Cluster Insertion low	0,008866071701
Cluster Insertion low (x3)	3000	603	1809000	95,1263794	0,03170887931	0,0000525852016	31,70887931	0,0525852016	donc le temps est négligeable		
Insertion middle	1000	2181	2181000	44,36719394	0,04436719394	0,00002034259236	44,36719394	0,02034259236		Cluster Insertion middle	0,0207529066
Insertion middle	1000	2181	2181000	39,02161407	0,03902161407	0,00001789161581	39,02161407	0,01789161581		Cluster Insertion middle	0,02235603333
Insertion middle	1000	2181	2181000	35,12266707	0,03512266707	0,00001610392805	35,12266707	0,01610392805		Cluster Insertion middle	0,02141904831
Cluster Insertion middle (x3)	3000	6543	19629000	118,5114751	0,03958582503	0,00000637570691	39,50382503	0,000637570691			
Insertion high	1000	23781	23781000	42,5663695	0,0425663695	0,000001789931408	42,5663695	0,001789931408		Cluster Insertion High	0,1742489338
Insertion high	1000	23781	23781000	42,88182302	0,04288182302	0,000001803197949	42,88182302	0,001803197949		Cluster Insertion High	0,1706110505
Insertion high	1000	23781	23781000	65,3891108	0,0653891108	0,000002749636719	65,3891108	0,002749636719		Cluster Insertion High	0,176196013
Cluster Insertion high (x3)	3000	71343	214229000	150,8373008	0,05027910026	0,000000704516802	50,27910026	0,000704516802			
INSERTION (CLUSTERIZATION)											
Cluster Insertion low	1000	201	201000	0,01045703888	0,00001645703888	0,0000000520206905	0,01045703888	0,0000520206905	Et donc le même principe en utilisant une requête	Cluster Fetching low	0,05611392784
Cluster Insertion low	1000	201	201000	0,009452104568	0,000009452104568	0,000000025399386	0,009452104568	0,000025399386	d'agrégation (req au lieu de 1000)	Cluster Fetching low	0,0569606636
Cluster Insertion low	1000	201	201000	0,008866071701	0,000008866071701	0,0000000410808046	0,008866071701	0,0000410808046		Cluster Fetching low	0,05898308754
Cluster Insertion low (x3)	3000	603	1809000	0,02875215115	0,000000959173883	0,0000001590669715	0,0959173883	0,00001590669715			
Cluster Insertion middle	1000	2181	2181000	0,0207529068	0,0000207529068	0,00000000951531794	0,0207529068	0,00000951531794		Cluster Fetching middle	0,4761519432
Cluster Insertion middle	1000	2181	2181000	0,02235603333	0,00002235603333	0,0000000102035916	0,02235603333	0,0000102035916		Cluster Fetching middle	0,521548225
Cluster Insertion middle	1000	2181	2181000	0,02141904831	0,00002141904831	0,000000009220746588	0,02141904831	0,000009220746588		Cluster Fetching middle	0,5115008354
Cluster Insertion middle (x3)	3000	6543	19629000	0,06462780843	0,0000215992548	0,000000020780327	0,0215992548	0,0000020780327			
Cluster Insertion high	1000	23781	23781000	0,1742489338	0,0001742489338	0,0000000073233245	0,1742489338	0,0000073233245		Cluster Fetching High	5,31631025
Cluster Insertion high	1000	23781	23781000	0,1706110505	0,0001706110505	0,000000007174513417	0,1706110505	0,000007174513417		Cluster Fetching High	5,5167097
Cluster Insertion high	1000	23781	23781000	0,171696013	0,000171696013	0,00000000719919139	0,171696013	0,00000719919139		Cluster Fetching High	5,59116267
Cluster Insertion high (x3)	3000	71343	214229000	0,5165629387	0,000172187462	0,00000000213581842	0,172187462	0,00000213581842			
FETCHING (basic access)											
Fetching low	1000	201	201000	0,0585610865	0,000585610865	0,00000002913489898	0,0585610865	0,0002913489898		Cluster Deleting low	0,8552341461
Fetching low	1000	201	201000	0,06018710136	0,0006018710136	0,000000029438312	0,06018710136	0,00029438312		Cluster Deleting low	2,361701808
Fetching low	1000	201	201000	0,048530354	0,00048530354	0,0000000241609828	0,048530354	0,000241609828		Cluster Deleting low	1,59364
Fetching low (x3)	3000	603	1809000	0,167311916	0,0005057039719	0,00000009248822087	0,057039719	0,00009248822087			
Fetching middle	1000	2181	2181000	0,2723271983	0,0002723271983	0,0000000124880414	0,2723271983	0,000124880414		Cluster Deleting middle	0,9407751179
Fetching middle	1000	2181	2181000	0,2405027273	0,0002405027273	0,0000000110381033	0,2405027273	0,000110381033		Cluster Deleting middle	0,9947180748
Fetching middle	1000	2181	2181000	0,4875208965	0,0004875208965	0,00000002381067843	0,4875208965	0,0002381067843		Cluster Deleting middle	1,272393018
Fetching middle (x3)	3000	6543	19629000	1,010405302	0,000336018773	0,0000000147912074	0,336018773	0,0000147912074			
Fetching high	1000	23781	23781000	4,162491083	0,0004162491083	0,000000175034166	4,162491083	0,000175034166	L'accès aux données est le point le plus crucial	Cluster Deleting High	0,462346077
Fetching high	1000	23781	23781000	3,72009926	0,000372009926	0,000000156431602	3,72009926	0,000156431602	qui justifie la création de mon système d'accès.	Cluster Deleting High	0,569180556
Fetching high	1000	23781	23781000	5,50944035	0,000550944035	0,0000002316531699	5,50944035	0,0002316531699	L'objectif à ce niveau est donc atteint	Cluster Deleting High	0,4842621478
Cluster Fetching high (x3)	3000	71343	214229000	13,39153504	0,00046384015	0,0000000625487861	4,6384015	0,0000625487861			
FETCHING (CLUSTERIZATION)											
Cluster Fetching low	1000	201	201000	0,06311392784	0,0006311392784	0,000000313999641	0,06311392784	0,000313999641			
Cluster Fetching low	1000	201	201000	0,0569606636	0,000569606636	0,0000002834132655	0,0569606636	0,0002834132655			
Cluster Fetching low	1000	201	201000	0,05898308754	0,0005898308754	0,0000000293481967	0,05898308754	0,000293481967			

Analyse de performances - 'lightdb' - Performances

Cluster Fetching low (x3)	3000	603	1809000	0,179063017	0,00059688769391	0,0000000989465702	0,05968769391	0,0000989465702			
Cluster Fetching middle	1000	2181	2181000	0,4761519432	0,0004761519432	0,0000000183181766	0,4761519432	0,000183181766			
Cluster Fetching middle	1000	2181	2181000	0,527154025	0,000527154025	0,0000000241703115	0,527154025	0,000241703115			
Cluster Fetching middle	1000	2181	2181000	0,5115008354	0,0005115008354	0,0000000345258301	0,5115008354	0,000345258301			
Cluster Fetching middle (x3)	3000	6543	19629000	1,514807701	0,000540399004	0,0000000771192425	0,540399004	0,0000771192425			
Cluster Fetching High	1000	23781	23781000	5,516381025	0,0005516381025	0,00000002319658982	5,516381025	0,0002319658982	L'accès aux données est le point le plus crucial		
Cluster Fetching high	1000	23781	23781000	5,516707897	0,0005516707897	0,00000002319796433	5,516707897	0,0002319796433	qui justifie la création de mon système d'accès.		
Cluster Fetching high	1000	23781	23781000	5,59116267	0,000559116267	0,0000000251407156	5,59116267	0,000251407156	L'objectif à ce niveau est donc atteint		
Cluster Fetching high (x3)	3000	71343	214229000	16,62425685	0,0005441895	0,0000000770291745	5,5441895	0,0000770291745			
DELETING											
Deleting low	10	201	2010	5,351806164	0,05351806164	0,00225601391	5,351806164	2,62590131	Nécessite la ré-écriture du fichier complet		
Deleting low	10	201	2010	4,536841154	0,04536841154	0,002257134903	4,536841154	2,257134903	sans la ligne à supprimer.		
Deleting low	10	201	2010	5,540421293	0,05540421293	0,00275642483	5,540421293	2,75642483	trouver meilleure solution		
Deleting low (x3)	30	603	18090	15,42906022	0,514200704	0,00082390548	15,4200704	0,852390548	car très très très très très très très très		
Deleting middle	10	2181	21810	6,332051039	0,06332051039	0,0022903278789	6,332051039	2,903278789	mais jamais utilisé dans le projet		
Deleting middle	10	2181	21810	4,98417592	0,0498417592	0,00222857094	4,98417592	2,2857094			
Deleting middle	10	2181	21810	5,572006941	0,05572006941	0,00254794563	5,572006941	2,54794563			
Deleting middle (x3)	30	6543	196290	16,8882339	0,56294113	0,000860371588	16,8882339	0,860371588			
Deleting high	10	23781	237810	5,0778987	0,050778987	0,0002216020006	5,0778987	0,02216020006			
Deleting high	10	23781	237810	5,34885073	0,0534885073	0,000224820509	5,34885073	0,0224820509			
Deleting high	10	23781	237810	4,154047812	0,04154047812	0,000174679402	4,154047812	0,0174679402			
Cluster Deleting high (x3)	30	71343	2142290	15,01068115	0,5005036084	0,000000701338675	5,00336084	0,0000701338675			
DELETING (CLUSTERIZED)											
Cluster Deleting low	1000	201	201000	0,8552341461	0,0008552341461	0,000004254896249	0,8552341461	0,0004254896249	En groupant les requêtes, on obtient un temps tout à		
Cluster Deleting low	1000	201	201000	2,361701808	0,02361701808	0,0001179050875	2,361701808	0,01179050875	fait raisonnable, il faut rajouter/implémenter un système		
Cluster Deleting low	1000	201	201000	1,59364	0,0159364	0,000007037184081	1,59364	0,0007037184081	de cache, puis valider (commit) les données		
Cluster Deleting low (x3)	3000	603	1809000	4,81039955	0,00160343318	0,00000269126564	1,60343318	0,000269126564	et grouper les suppressions		
Cluster Deleting middle	1000	2181	2181000	0,9807751179	0,0009807751179	0,0000044690563	0,9807751179	0,00044690563			
Cluster Deleting middle	1000	2181	2181000	1,272393018	0,001272393018	0,0000050583115627	1,272393018	0,0005			

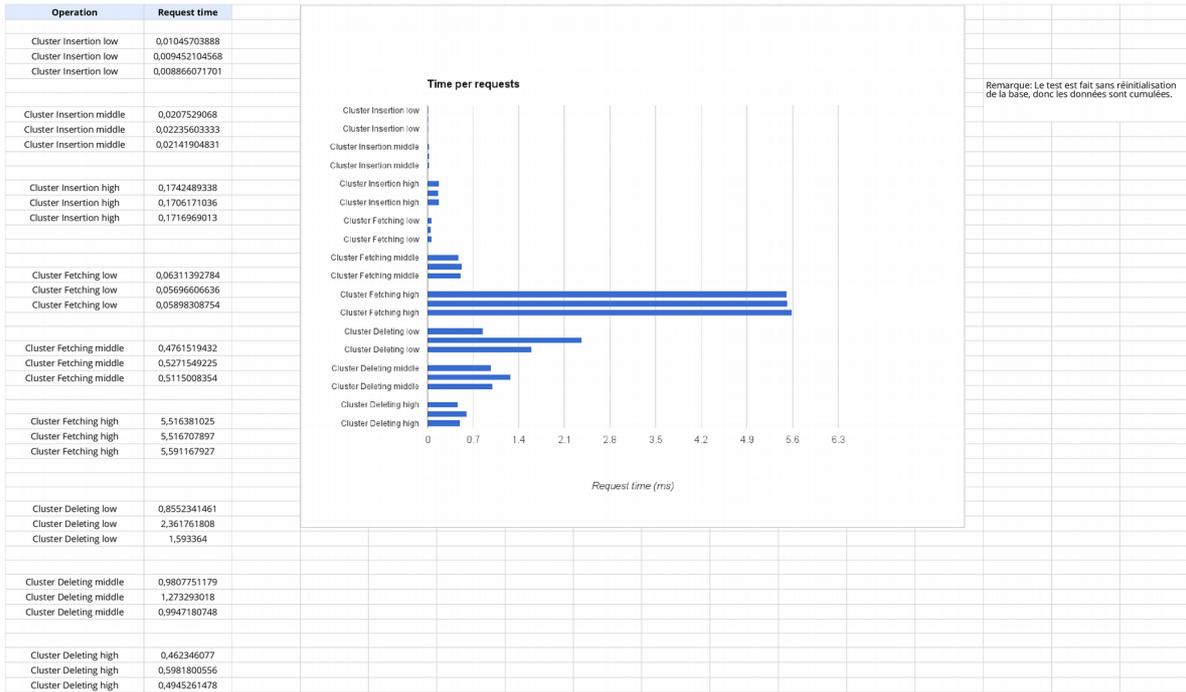
Analyse de performances - `lightdb` - Individual Speed

Operation	Request time
Insertion low	29,93966293
Insertion low	30,8989501
Insertion low	34,2880249
Insertion middle	44,36719394
Insertion middle	39,02161407
Insertion middle	35,12266707
Insertion high	42,56636095
Insertion high	42,88182902
Insertion high	65,3891108
Fetching low	0,05856108665
Fetching low	0,06018710136
Fetching low	0,04856300354
Fetching middle	0,2723371983
Fetching middle	0,2405672073
Fetching middle	0,4975008965
Fetching high	4,162491083
Fetching high	3,720099926
Fetching high	5,508944035
Deleting low	535,1806164
Deleting low	453,6841154
Deleting low	554,0412903
Deleting middle	633,2051039
Deleting middle	498,417592
Deleting middle	557,2006941
Deleting high	550,7739067
Deleting high	534,8895073
Deleting high	415,4047012

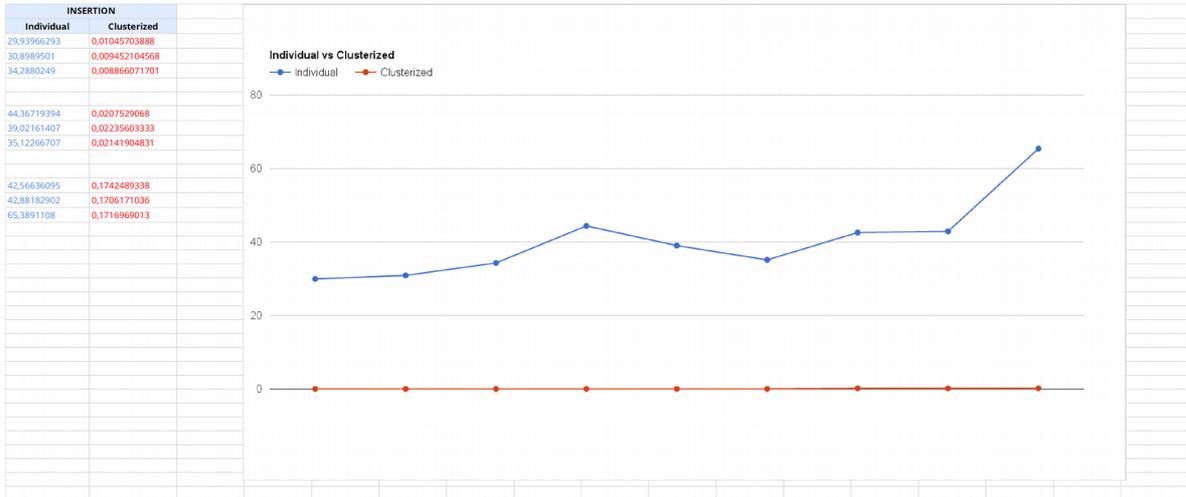


Remarque: Le test est fait sans réinitialisation de la base, donc les données sont cumulées.

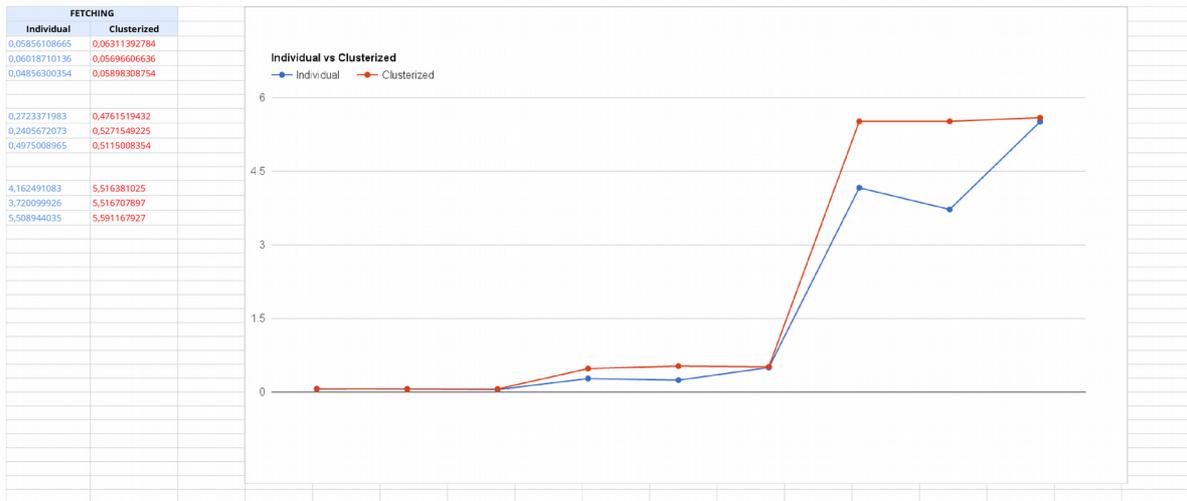
Analyse de performances - 'lightdb' - Cluster Speed



Analyse de performances - 'lightdb' - Cluster Insert



Analyse de performances - 'lightdb' - Cluster Fetch



Analyse de performances - 'lightdb' - Cluster Delete

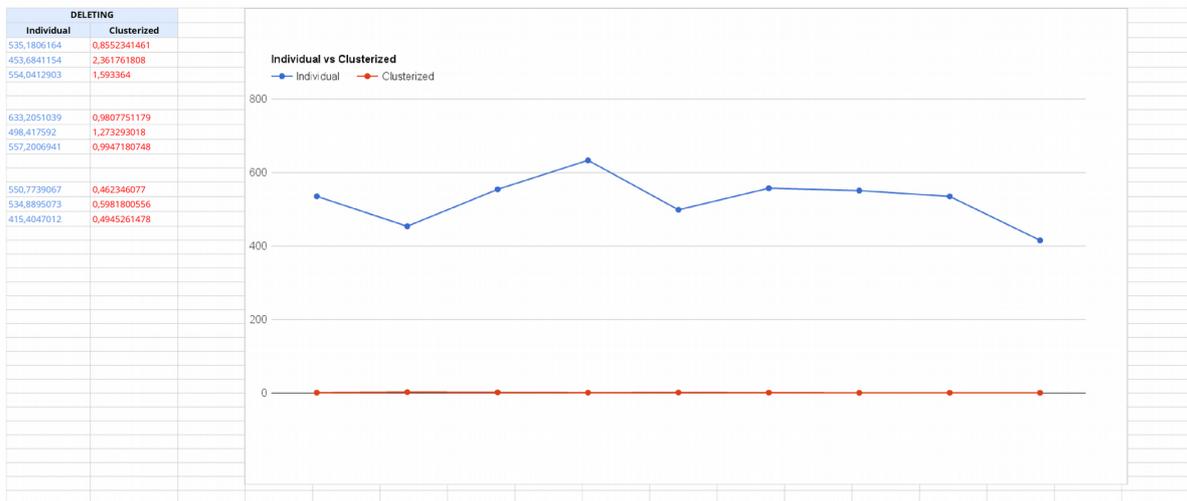
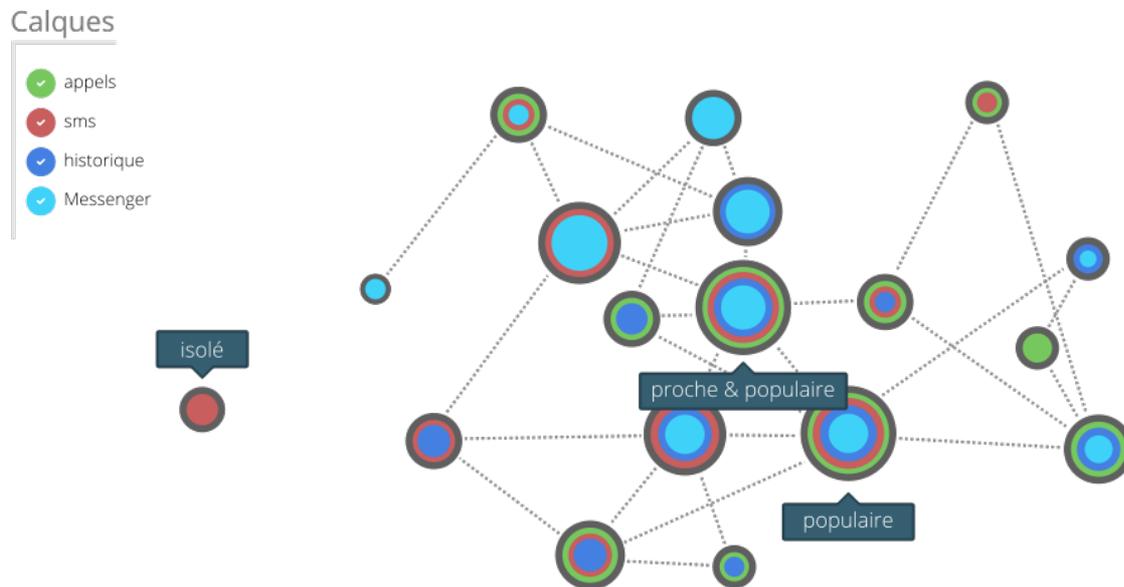


Figure 4.2.4-1 Prototype du graphique de type *réseau*

Ci-dessous le prototype élaboré suite aux discussions avec les chercheurs. Il fonctionne sur le principe de superposition de calques. Il est possible de cocher ou décocher les différents calques afin de filtrer les données.



Chaque **cercle** représente une personne définie par le sujet dans un des quatre « top 10 » de l'enquête. Les **couleurs** dans les **cercles** sont associées aux types de communication, les couleurs sont données dans le panneau "Calques" qui se situe à gauche. Les **pointillés** représente les liens entre les personnes qui se connaissent.

Figure 4.2.4-2 Visualisation des données d'un sujet

