

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

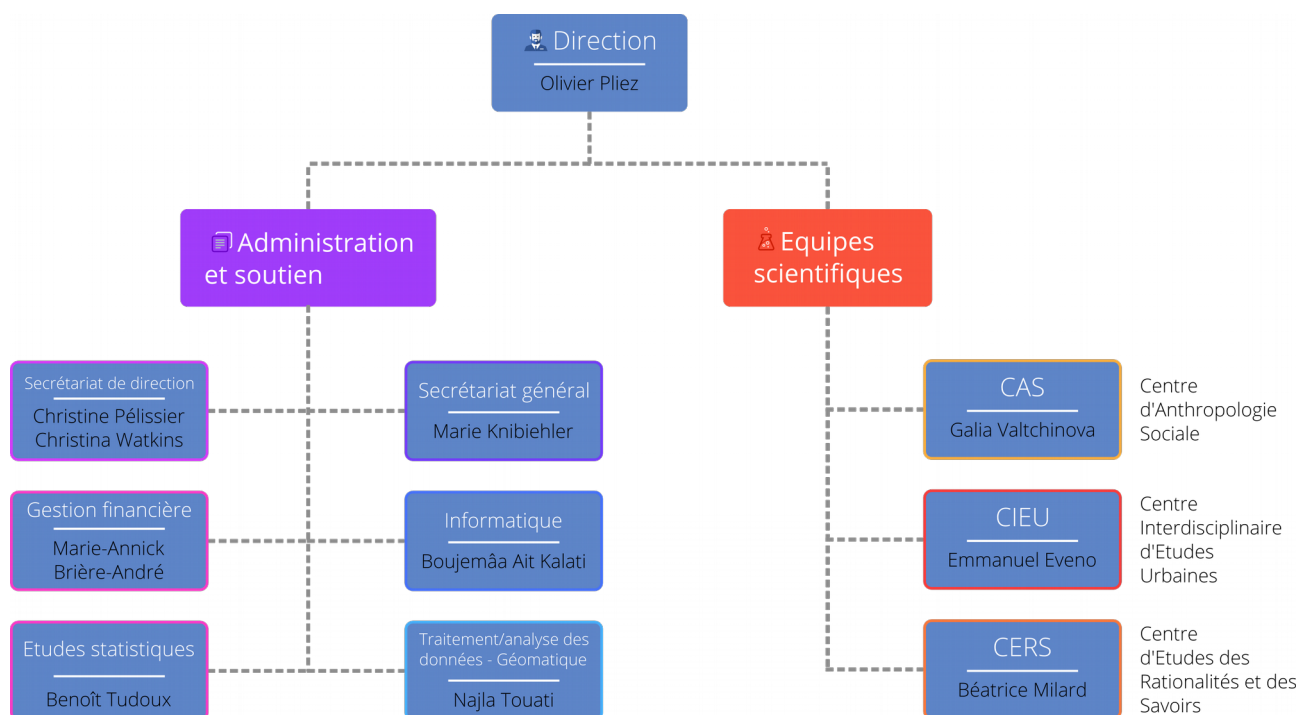
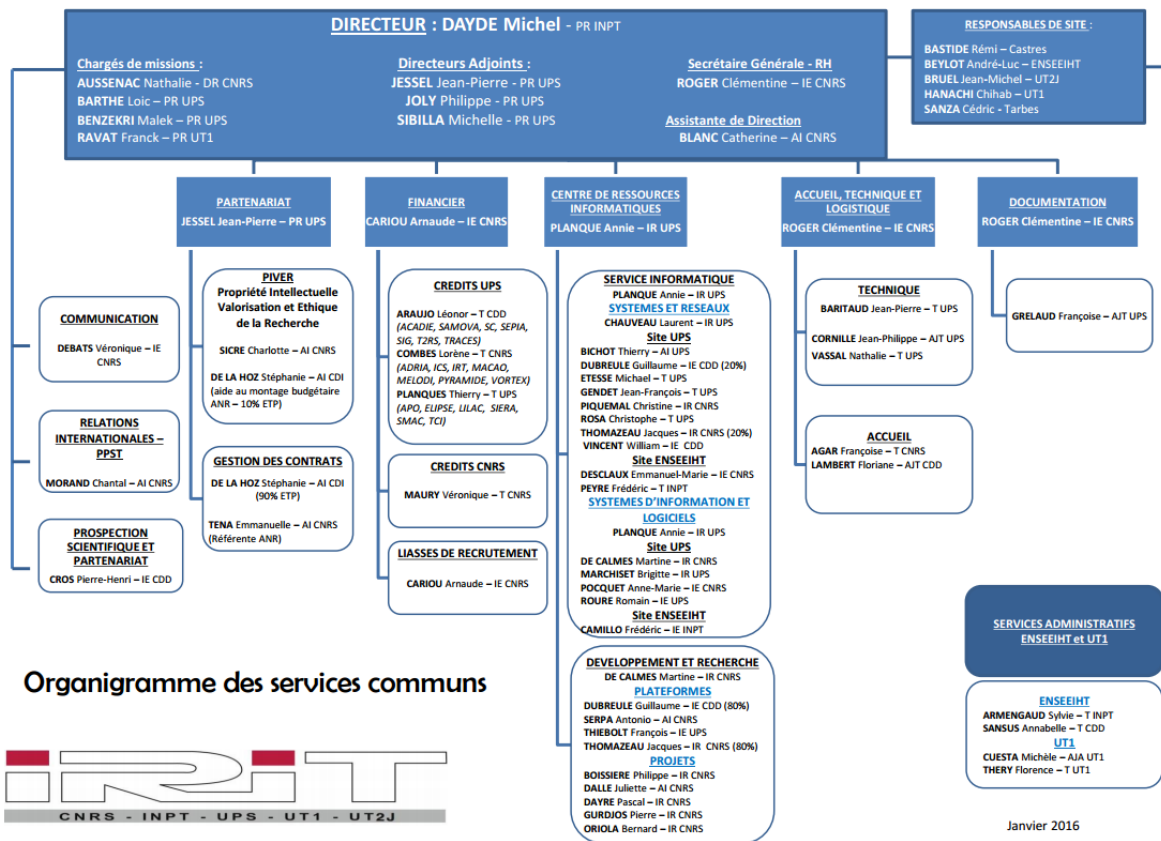


Figure 2.1-2 Description officielle de l'équipe LISST-CERS

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.

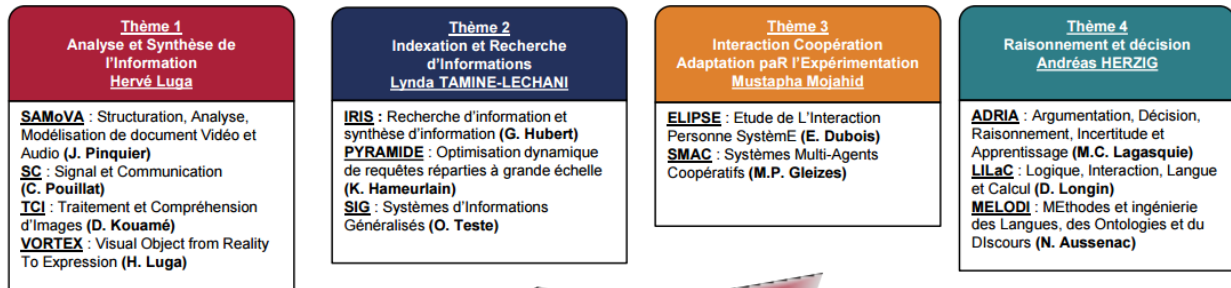
Figure 2.2-1 Organigramme des services communs de l'IRIT



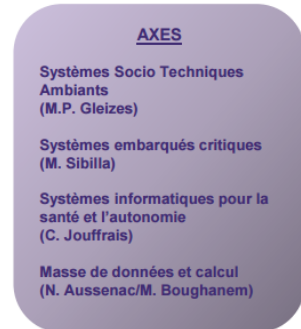
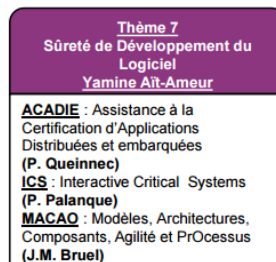
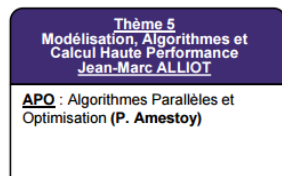
Organigramme des services communs



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



7 thèmes – 21 équipes – 4 axes



Recherche



Mars 2016

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.

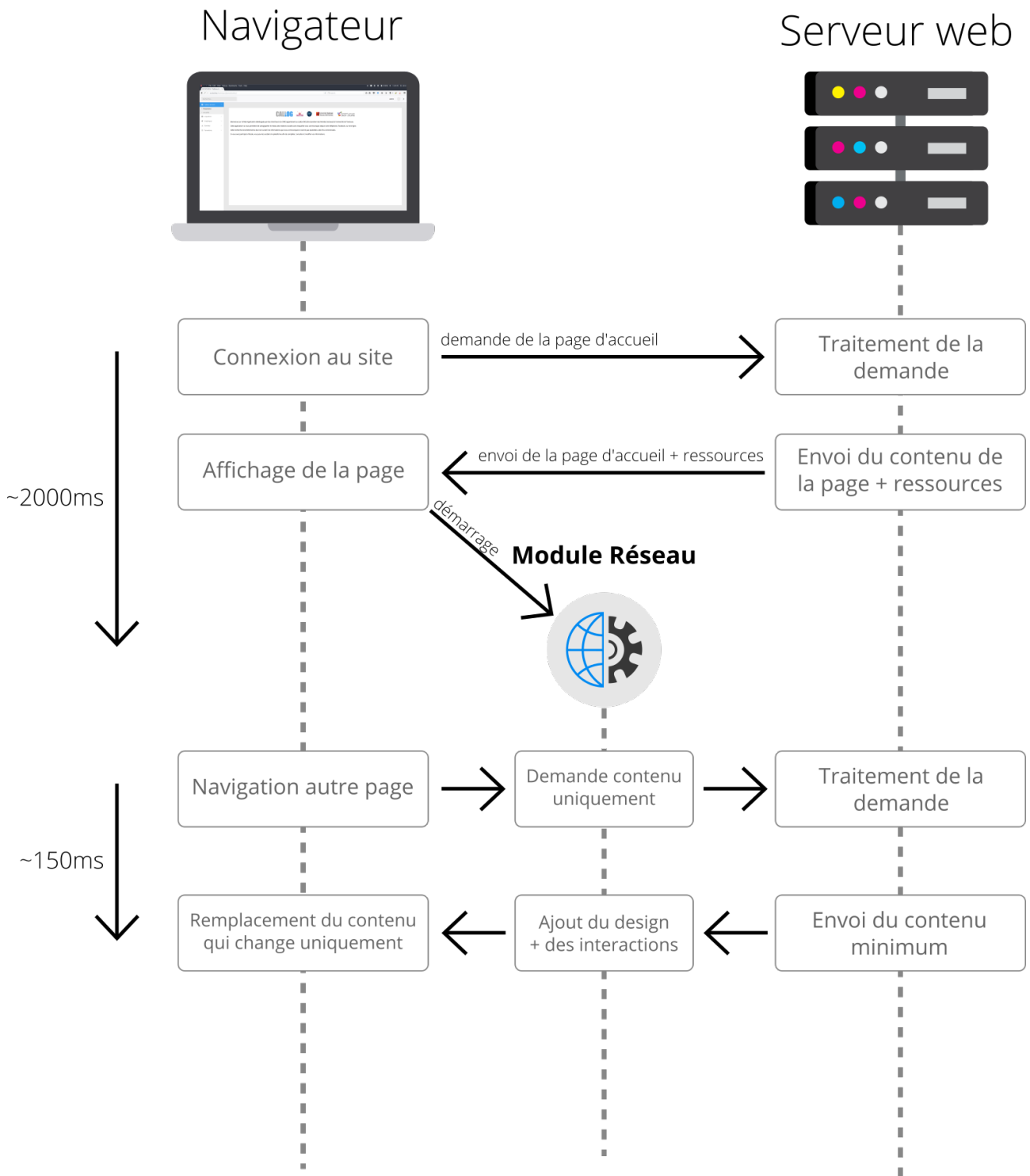
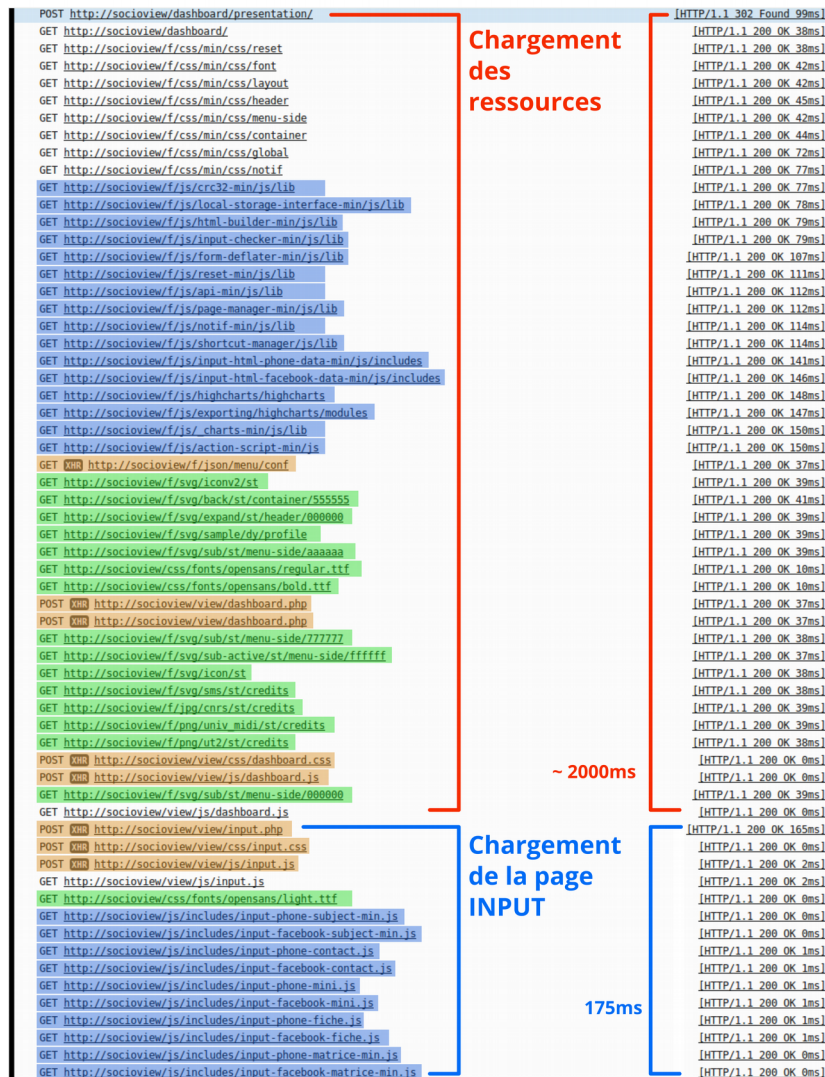


Figure 3.1.1-2 Analyse réseau détaillée

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

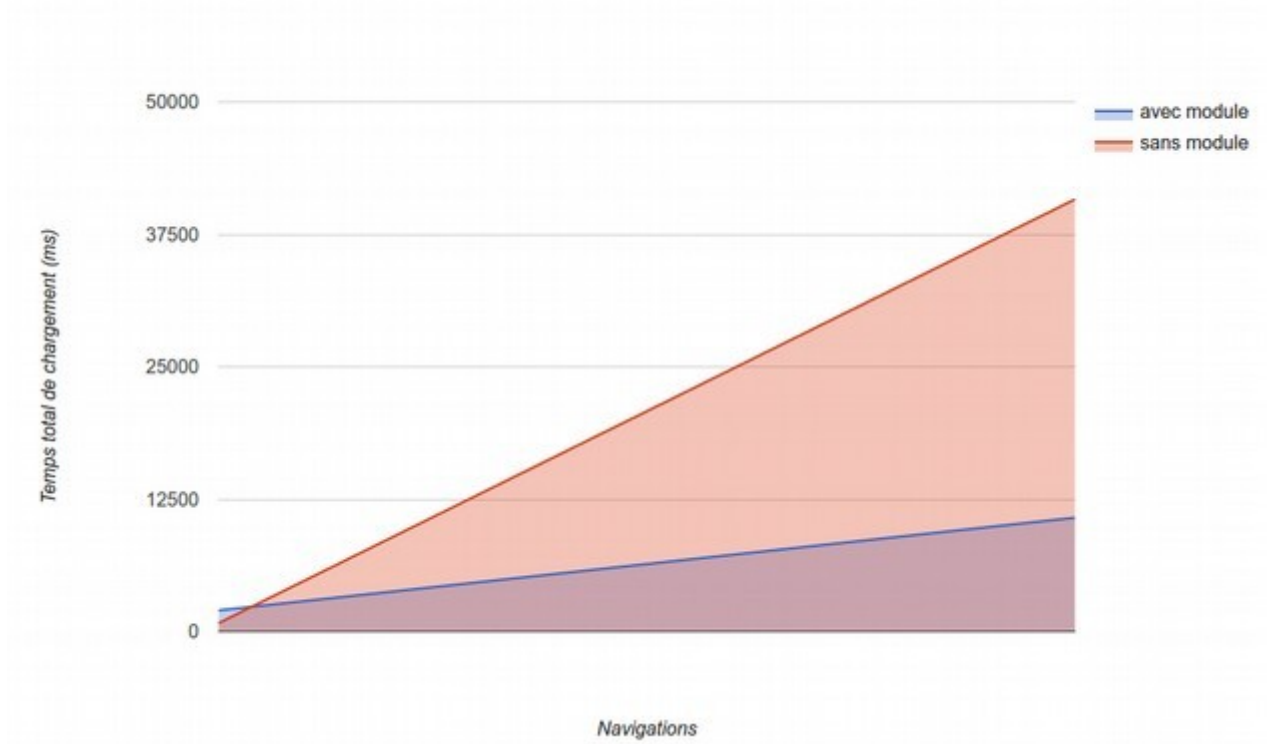


On remarque que le chargement du système complet prends aux alentours de 2 secondes, puis le chargement de chaque page est infime, de l'ordre de 0.175 secondes.

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.



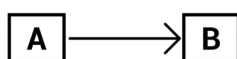
On remarque qu'à partir de 2 navigations (premier chargement inclus), le gain de temps est significativement supérieur à une utilisation normale.

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) |

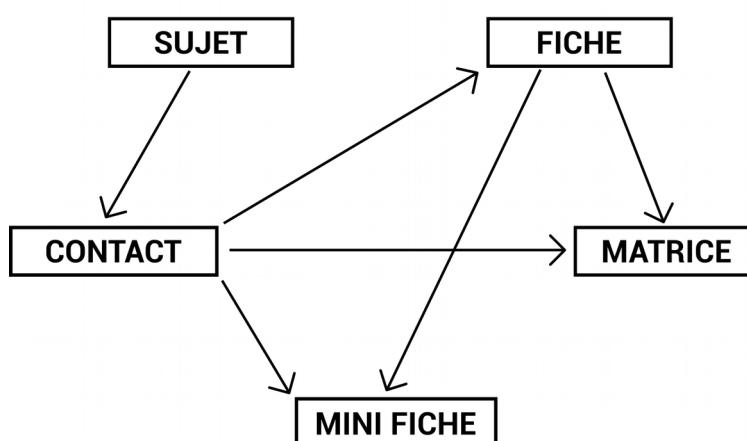
Ci-dessous le schéma d'inter-dépendance entre les entités.

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 a description, permissions, and a 'Try' button. The right sidebar shows the request and response details for the 'search' endpoint, including headers and a JSON schema.

subject

search

Recherche d'un sujet par nom

Permissions: admin

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

subject search

POST <https://soctiview.xdrm.fo/api/subject/search>

Parameters: Le nom du sujet (Verchar(1,50))

Request: REQUETE

Production | Raw | Try

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

ATTRIBUTES

200

HEADERS

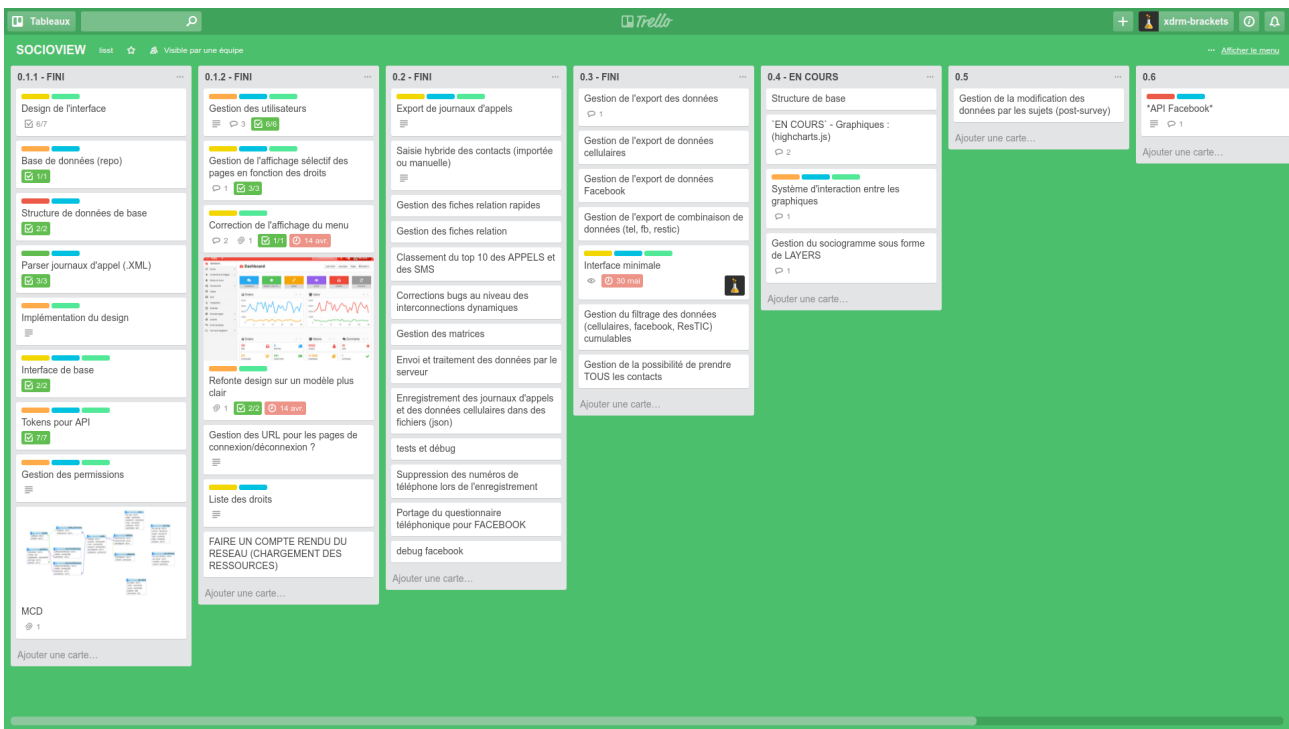
Content-Type: application/json

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

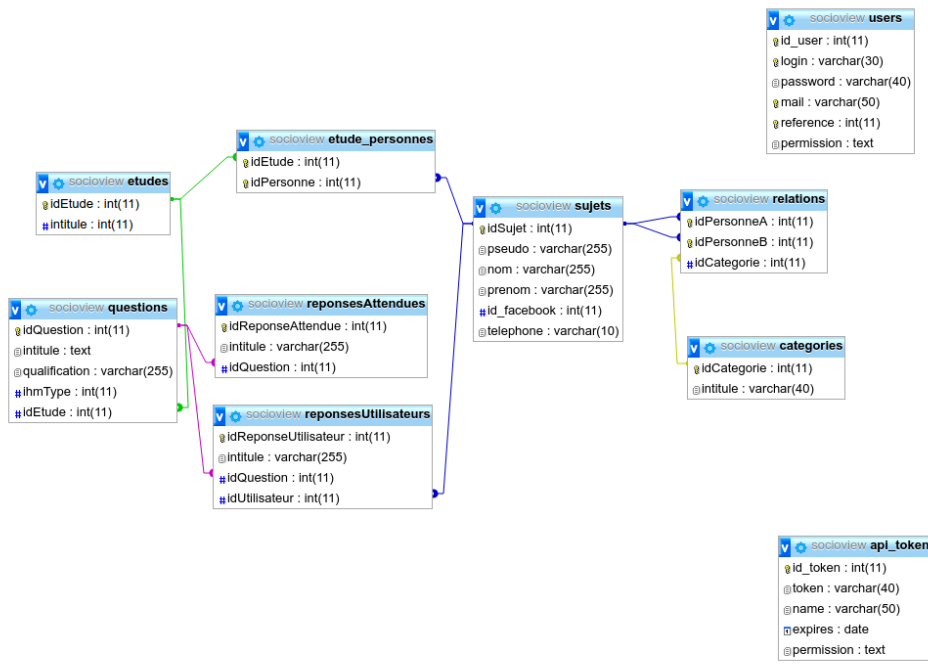


Figure 4.2.1-2 Premier prototype de la charte graphique

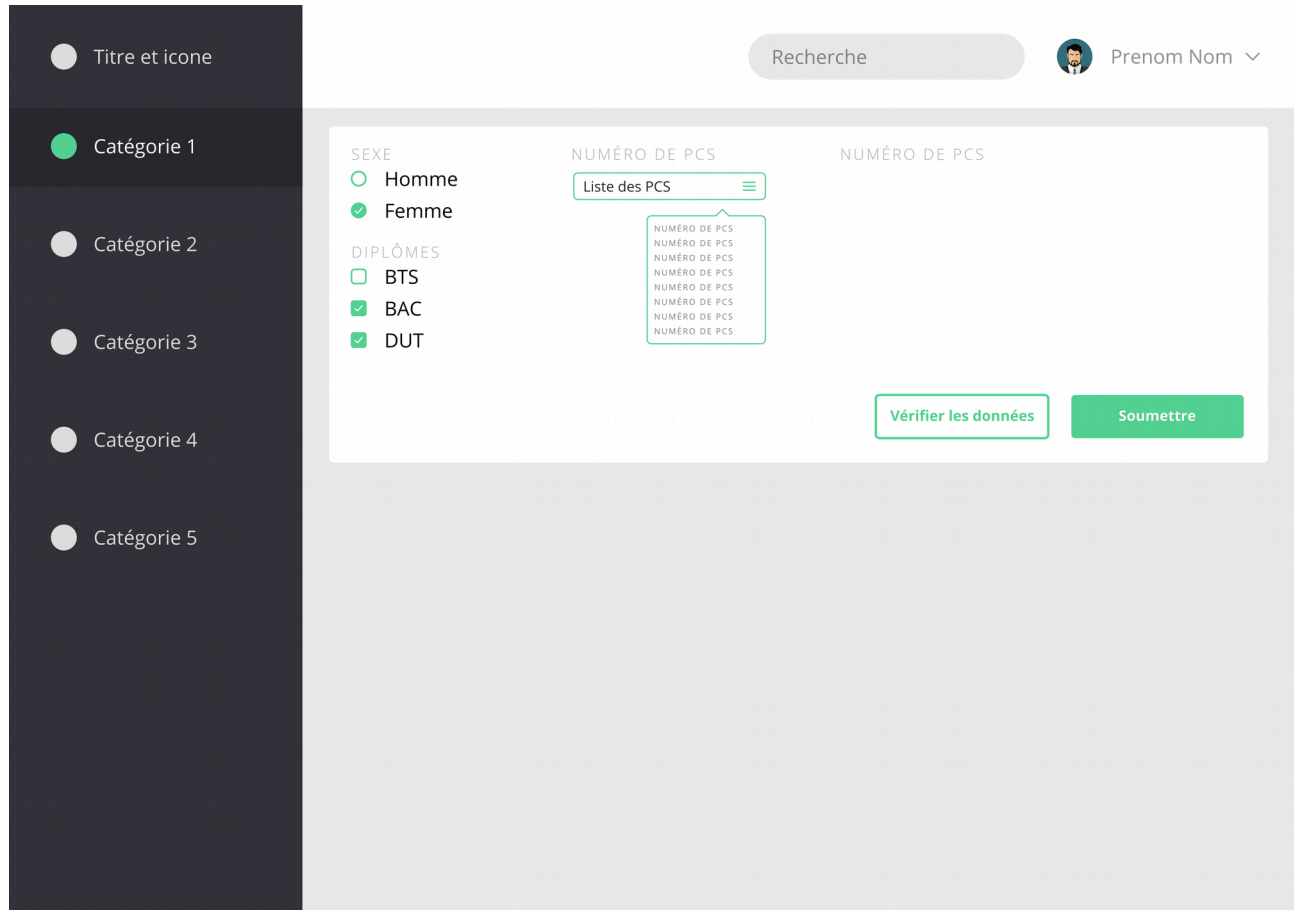


Figure 4.2.1-3 Framework par fichiers

| | |
|--------------------|--|
| 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

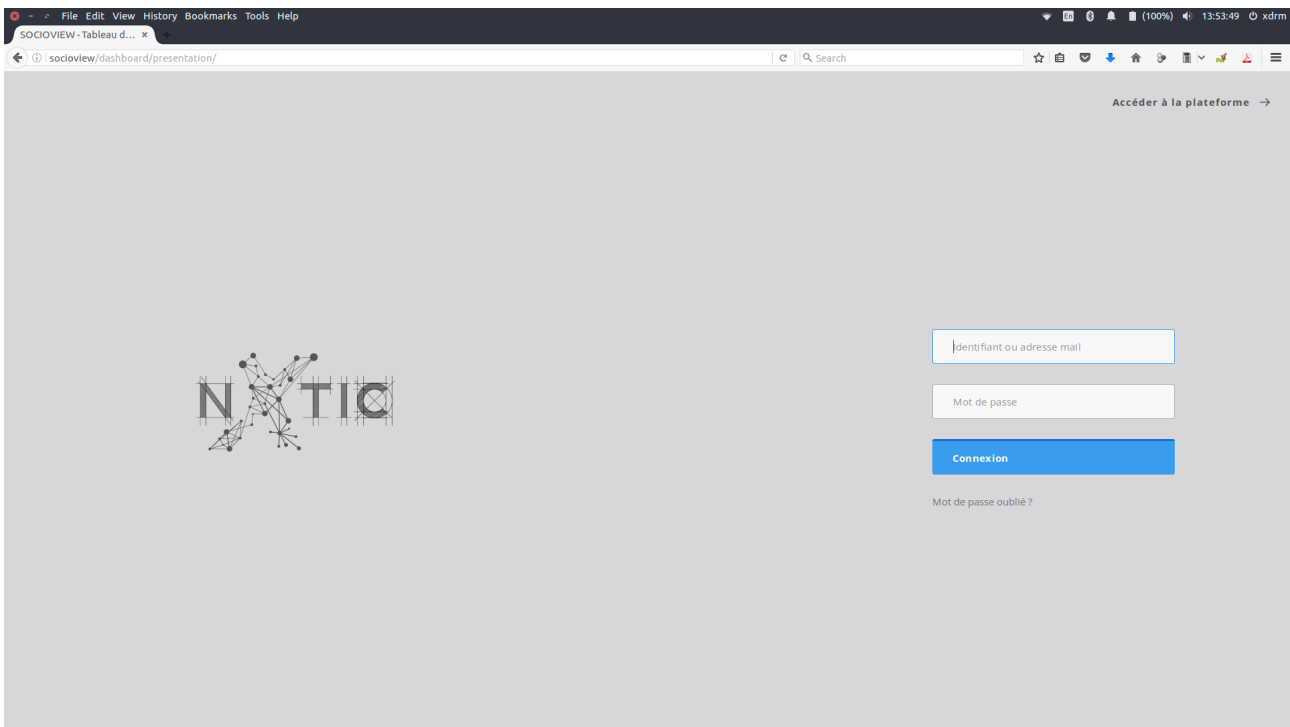


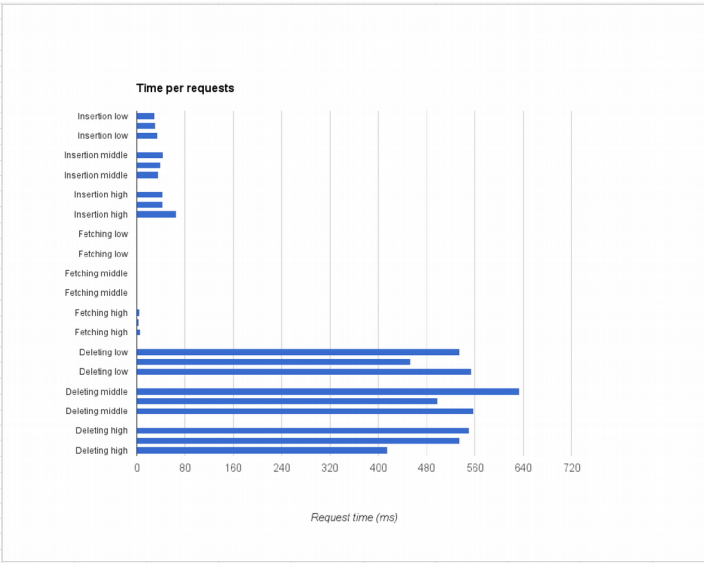
Figure 4.2.3-1 Benchmark de *lightdb*

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 |

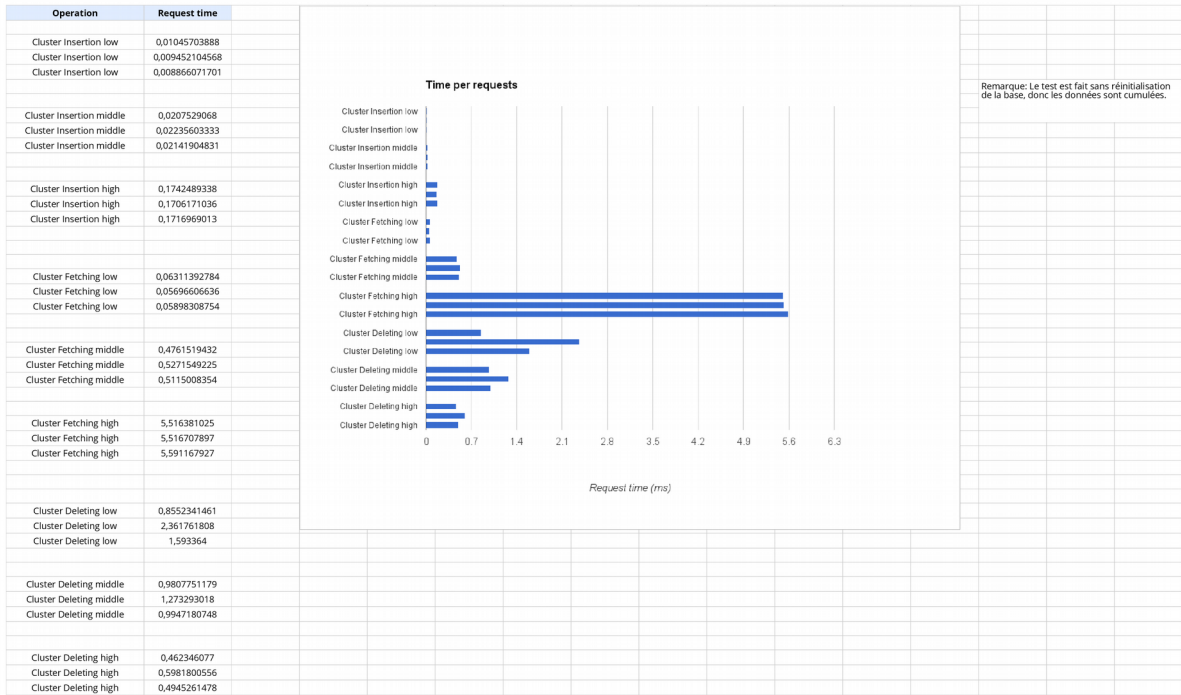
Analyse de performances - `lightdb` - Individual Speed

| 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 |

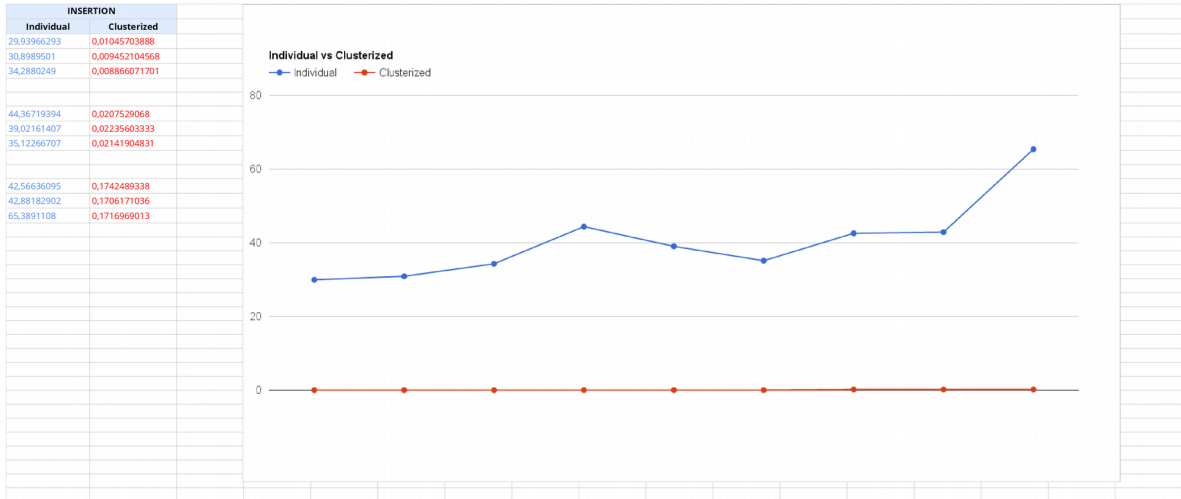


Remarque: Le test est fait sans réinitialisation de la base, donc les données sont cumulées.

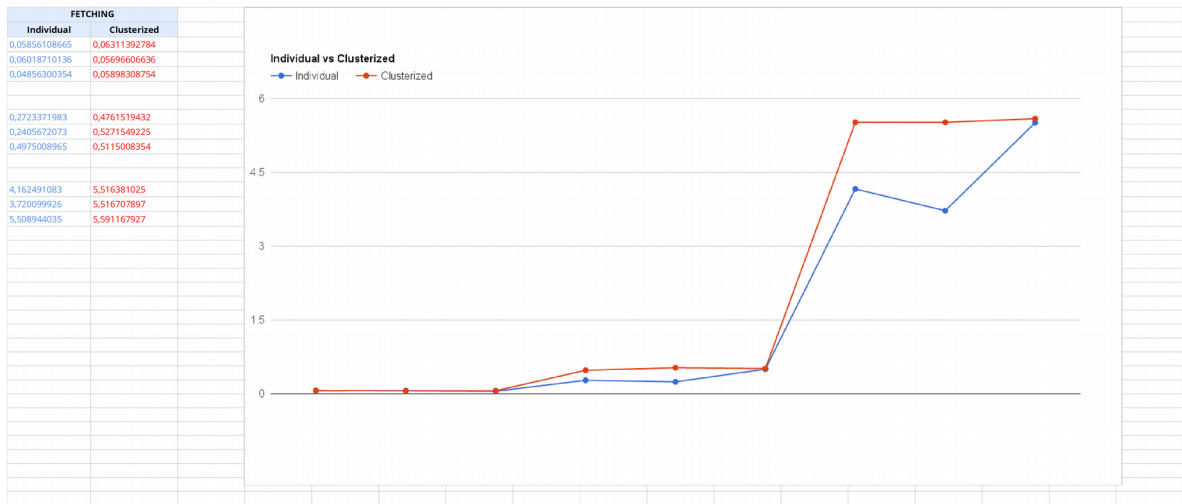
Analyse de performances - 'lightdb' - Cluster Speed



Analyse de performances - 'lightdb' - Cluster Insert



Analyse de performances - 'lightdb' - Cluster Fetch



Analyse de performances - 'lightdb' - Cluster Delete

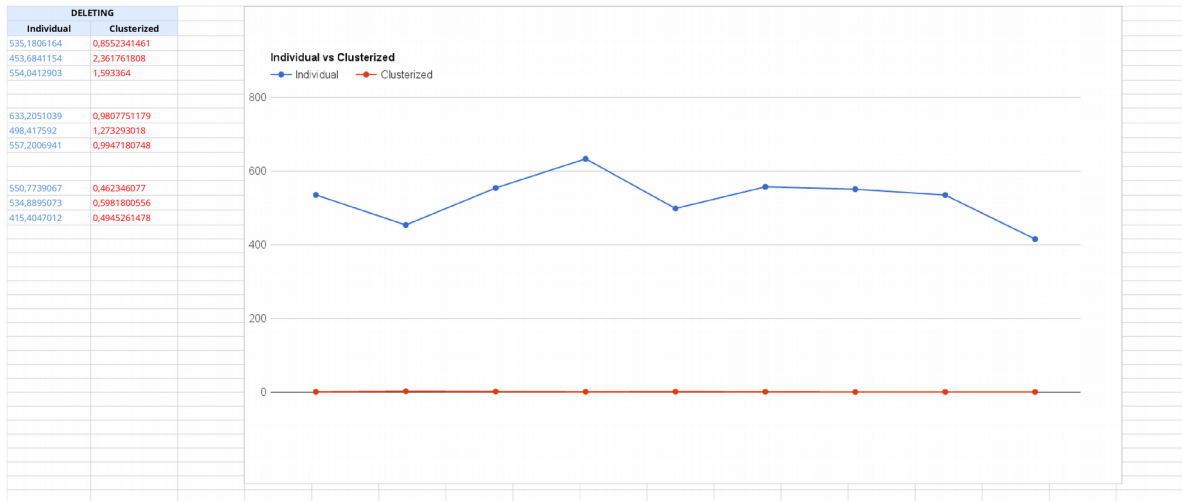
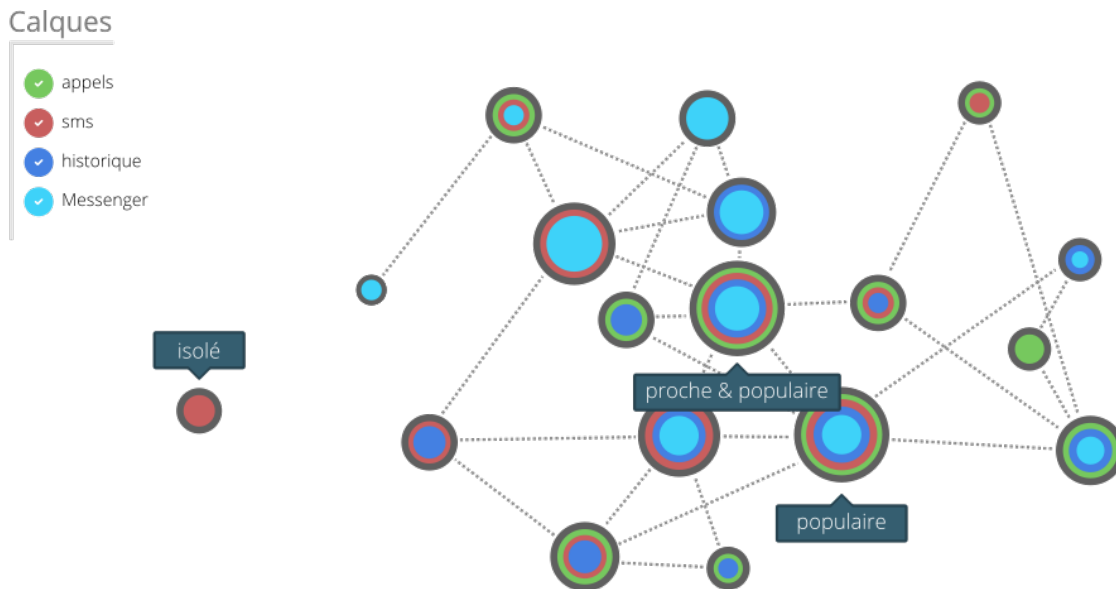


Figure 4.2.4-1 Prototype du graphique de type *réseau*

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

