

A thick vertical purple bar on the left side of the page. A horizontal purple arrow points to the right from the bar, containing the text 'P\_DEV'.

P\_DEV

# Concevoir une médiathèque audio/vidéo partagée avec un protocole P2P

**Chef de projet :** CARREL Xavier

VELICKOVIC Mateja

ÉCOLE DES MÉTIERS TECHNIQUES LAUSANNE



# MQTT

---

Table des matières

1	Analyse préliminaire .....	3
1.1	Introduction .....	3
1.2	Objectifs .....	3
1.3	Gestion de projet .....	4
2	Analyse / Conception .....	4
2.1	Domaine .....	4
2.2	Concepts .....	4
2.3	Analyse fonctionnelle .....	5
2.4	Stratégie de test .....	8
3	Réalisation .....	8
3.1	Points de design spécifiques .....	8
3.1.1	.....	9
3.1.2	.....	9
3.1.3	.....	9
3.2	Déroulement .....	9
3.3	Mise en place de l'environnement de travail .....	9
3.4	Description des tests effectués .....	9
3.5	Bilan .....	9
3.5.1	Erreurs restantes .....	9
3.5.2	Stories .....	9
3.5.3	Dette technique .....	9
3.6	Recours à l'intelligence artificielle .....	10
4	Conclusions .....	10
5	Annexes .....	11
5.1	Journal de travail .....	11

## **1 Analyse préliminaire**

### **1.1 Introduction**

Le projet BitRuisseau a pour but de consulter et de partager ses fichiers audio et vidéo sur une médiathèque (locale ou non) en utilisant un protocole défini avant le démarrage du projet, chaque utilisateur pourra se connecter à un topic (sujet) et pourra consulter les différentes médiathèques présentes dans la communauté.

### **1.2 Objectifs**

Chaque utilisateur pourra :

- Demander sur le réseau quels sont les médiathèques opérationnelles.
- S'annoncer opérationnelle.
- Demander le catalogue à une médiathèque.
- Publier son catalogue.

Fonctionnalités avancées :

- Demander un fragment de média à une médiathèque.
- Fournir un fragment de média à une médiathèque.

### 1.3 Gestion de projet

Pour la gestion de ce projet, j'ai principalement utilisé l'outil IceScrum pour la création de tâches et l'écriture des tests les concernant, GitHub m'a permis de versionner de mon projet afin de garder un suivi de ce dernier et de noter le temps passé sur une tâche en la référençant dans le titre de chaque commit.

## 2 Analyse / Conception

### 2.1 Domaine

- *Quel est le domaine auquel appartiennent vos données ?*
- *Que représentent-elles ?*
- *Quelles échelles de temps sont pertinentes pour ces données ?*
- *A quel public sont-elles destinées ?*

### 2.2 Concepts

#### Protocole MQTT 1.0

```
Version du protocole : 1.0

Protocole : MQTT

Topic : mediaplayer

Message type :

MEDIA_STATUS_REQUEST // Transporte rien du tout

MEDIA_STATUS          // Transporte un objet de type Mediatheque

Class media et mediatheque :

class Media
{
    string Filename
    int Size
    enum Mediatype {
        MP3,
        MP4,
        MOV,
        GIF,
        PNG,
        JPEG,
        JPG,
        WAV,
    }
}

class Mediatheque
{
    bool IsAvailable // Je suis disponible et non connecté.
    string DisplayName
    List<Media> Medias
}

}
```

La définition du protocole 1.0 a été faite en date du 13 novembre 2024 en début d'après-midi, cette dernière se basait principalement sur quel protocole nous allions travailler ainsi que les messages qui vont être envoyés sur le sujet (topic) prédéfini, qui dans notre cas est : mediaplayer.

Nous avons aussi défini différentes classes pour les médias et les médiathèques afin de tous avoir la même classe et de ne pas se retrouver avec des informations différentes.

En ce qui concerne le type de message nous en avons défini deux nouveau à ajouter, un premier pour demander le statut d'une médiathèque et le second qui va transporter un objet de type médiathèque qui va donc contenir toutes les informations de cette dernière.

Figure 1 : Définition d'un protocole

## 2.3 Analyse fonctionnelle

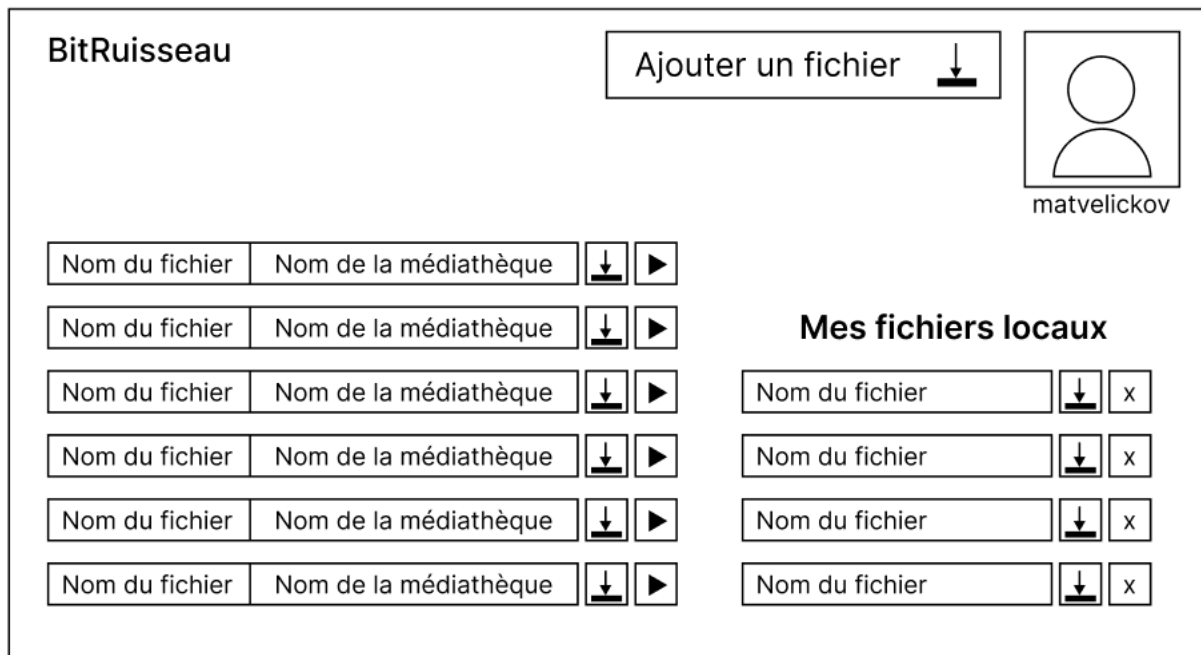


Figure 2 : Maquette Low Fidelity de l'application BitRuisseau

Voici la maquette Low Fidelity qui a été réalisée sur Figma en une dizaine de minutes, l'utilisateur peut voir les différents fichiers publiés sur la communauté ainsi que ses fichiers locaux que lui-même a ajoutés via un bouton situé en haut à droite de son interface.

L'utilisateur peut aussi voir son profil avec son nom d'utilisateur en haut à droite de l'interface aussi.  
(voir figure 2)

### 1. Pouvoir lister le contenu de la médiathèque localement

**En tant qu'utilisateur**, je veux pouvoir visionner les médias qui ont été chargés localement afin de les consulter.

#### Tests d'acceptance :

- **Étant donné** que je veuille lister les médias chargés
- **Lorsque** je suis sur la page d'accueil
- **Alors** je peux voir tous les médias qui ont été chargés.

## 2. Pouvoir visualiser la liste des contenus disponibles dans la communauté

**En tant qu'utilisateur**, je veux pouvoir visionner les médias disponibles dans la communauté afin de consulter les médias de différents utilisateurs.

### Tests d'acceptance :

- **Étant donné** que je veuille lister les médias de la communauté
  - **Lorsque** je suis sur la page de la communauté
  - **Alors** je peux voir tous les médias publiés.
- 

## 3. Pouvoir visualiser les médiathèques opérationnelles

**En tant qu'utilisateur**, je veux pouvoir visualiser les médiathèques opérationnelles afin de les consulter.

### Tests d'acceptance :

- **Étant donné** que je désire consulter les médiathèques opérationnelles
  - **Lorsque** j'arrive sur l'interface des médiathèques
  - **Alors** les médiathèques opérationnelles s'affichent.
- 

## 4. Pouvoir s'annoncer opérationnelle

**En tant qu'utilisateur**, je veux pouvoir m'annoncer opérationnelle auprès du broker afin de partager mon statut.

### Tests d'acceptance :

- **Étant donné** que je désire m'annoncer opérationnelle
  - **Lorsque** je définis mon statut comme *opérationnelle*
  - **Alors** je transmets mon statut au broker.
- 

## 5. Demander le catalogue à une médiathèque

**En tant qu'utilisateur**, je veux demander le catalogue d'une médiathèque afin de consulter son contenu.

### Tests d'acceptance :

- **Étant donné** que je désire consulter le catalogue d'une médiathèque
- **Lorsque** je clique sur une médiathèque
- **Alors** son catalogue s'affiche.

## 6. Publier son catalogue à la vue des autres médiathèques

**En tant qu'utilisateur**, je veux pouvoir publier mon catalogue afin que les autres médiathèques puissent le consulter.

### Tests d'acceptance :

- **Étant donné** que je désire partager mon catalogue
  - **Lorsque** je clique sur **Partager**
  - **Alors** mon catalogue est publié.
- 

## 7. Pouvoir demander un fragment de média à une médiathèque

### Tests d'acceptance :

- **Étant donné** que je désire demander un média
  - **Lorsque** je demande ce média à une médiathèque
  - **Alors** le média m'est envoyé sous forme de fragment.
- 

## 8. Pouvoir transmettre un fragment de média à une médiathèque

### Tests d'acceptance :

- **Étant donné** que je désire transmettre un fragment de mon média
  - **Lorsque** je le transmets à une médiathèque
  - **Alors** le fragment est envoyé.
- 

## 9. Pouvoir ajouter/charger un média depuis un dossier local

**En tant qu'utilisateur**, je veux pouvoir charger un média localement afin de le partager.

### Tests d'acceptance :

- **Sélectionner un média :**
  - **Étant donné** que je veuille sélectionner un média
  - **Lorsque** j'appuie sur *Charger un média*
  - **Alors** une interface s'ouvre pour la sélection.
- **Charger un média :**
  - **Étant donné** que je sois sur l'interface de sélection

- **Lorsque** je sélectionne et confirme un média
- **Alors** il est chargé dans l'application.

## 2.4 Stratégie de test

*Décrire quels sont les **MOYENS** utilisés pour faire les tests, ne pas décrire les tests à effectuer !!!*

*Décrire l'environnement dans lequel se fait la sprint review*

*Décrire la stratégie globale de test :*

- types de des tests et ordre dans lequel ils seront effectués.
- les moyens à mettre en œuvre.
- couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).
- données de test à prévoir (données réelles ?) **et comment elles seront mises en place.**
- les testeurs extérieurs éventuels.

## 3 Réalisation

### 3.1 Points de design spécifiques

*Ce chapitre est constitué de plusieurs sous-chapitre.*

*Chaque sous-chapitre explique un point de design technique particulier, quelque chose que vous avez dû inventer pour répondre au besoin et qui ne peut pas s'expliquer par de simples commentaires dans le code.*

*Il s'agit d'explications techniques sur le fonctionnement du système. Les explications sont appuyées par des diagrammes, ou de très brefs éléments de code.*

*NE PAS mettre ici des pratiques usuelles que tout professionnel de la branche connaît déjà. Par exemple, n'EXPLIQUEZ PAS ICI CE QU'EST LE PATTERN MVC.*

*Exemple (simplifié à l'extrême) : Protection contre des formulaires mal intentionnés ou modifiés*

- **Au moment de générer le formulaire, le script php :**
  - **Concatène les noms de tous les champs contenus dans le formulaire**
  - **Calcule un hash SHA256 de la chaîne obtenue**
  - **Ajoute un input nommé « CSRF » de type hidden dans le form**
- **A la réception du POST du formulaire**
  - **Concatène les noms des indices de \$\_POST**



- *Calcule un hash SHA256 de la chaîne obtenue*
- *Vérifie que la valeur du champ CSRF correspond*

### 3.1.1 ...

### 3.1.2 ...

### 3.1.3 ...

## 3.2 Déroulement

*Résumer comment s'est passé la réalisation de chaque story, ses difficultés, les alternatives envisagées mais rejetées, ses surprises, ...*

## 3.3 Mise en place de l'environnement de travail

- *Comment accéder au code source*
- *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
- *les versions des systèmes d'exploitation et des outils logiciels*
- *la description exacte du matériel*

*Ce chapitre décrit précisément comment un employé qualifié peut recréer l'environnement dans lequel vous avez effectué ce travail*

## 3.4 Description des tests effectués

*Reprendre les tests d'acceptance d'IceScrum au moyen de la feuille ad hoc d'IceTools*

## 3.5 Bilan

### 3.5.1 Erreurs restantes

*S'il reste encore des erreurs :*

- *Description détaillée*
- *Conséquences sur l'utilisation du produit*
- *Actions envisagées ou possibles*

### 3.5.2 Stories

*Ce qu'on pensait faire vs ce qu'on a fait*

### 3.5.3 Dette technique

*Reporter la dette technique connue. S'appuyer sur la pratique des // TODO*

### 3.6 Recours à l'intelligence artificielle

**Comment avez-vous utilisé l'IA dans votre projet.  
Si vous ne l'avez pas utilisée, pourquoi ?**

## 4 Conclusions

*Développez en tous cas les points suivants :*

- *Objectifs atteints / non-atteints*
- *Points positifs / négatifs*
- *Difficultés particulières*
- *Suites possibles pour le projet (évolutions & améliorations)*

## **5 Annexes**

### **5.1 Journal de travail**