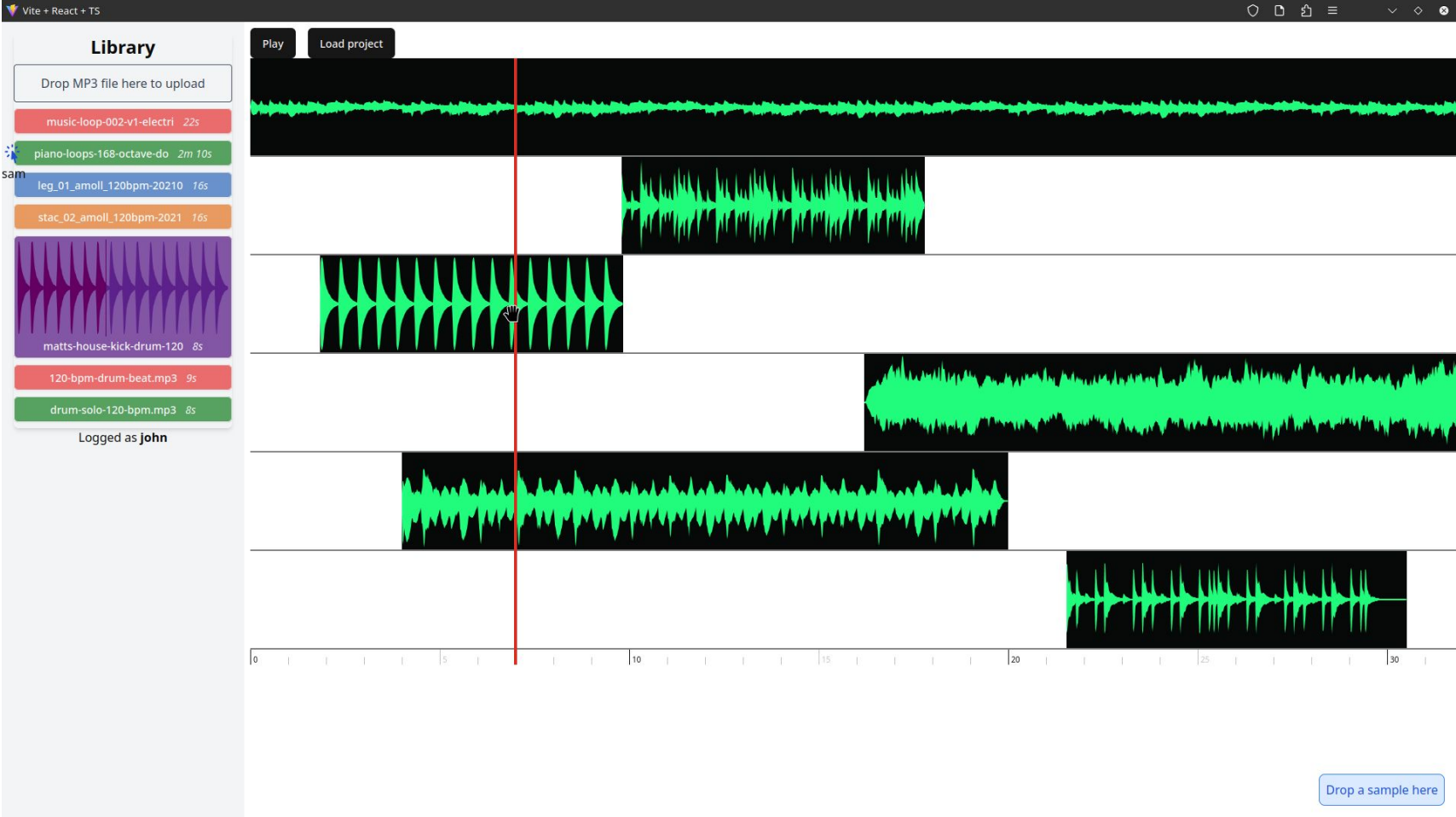




**JamCraft**

COLLABORATIVE  
EDITOR

# JamCraft - Demo



# Technologie gRPC

google Remote Procedure Call

```
package mouse;

service MouseService {
    // Server-side streaming to broadcast mouse positions to all clients
    rpc GetMouseUpdates(MouseSubscription) returns (stream MousePosition);
    // Unary sending to let clients send their mouse positions when they mo
    rpc SendMousePosition(MousePosition) returns (google.protobuf.Empty);
}

message MousePosition {
    uint32 userId = 1;
    uint32 x = 2;
    uint32 y = 3;
}

message MouseSubscription {
    uint32 userId = 1;
}
```

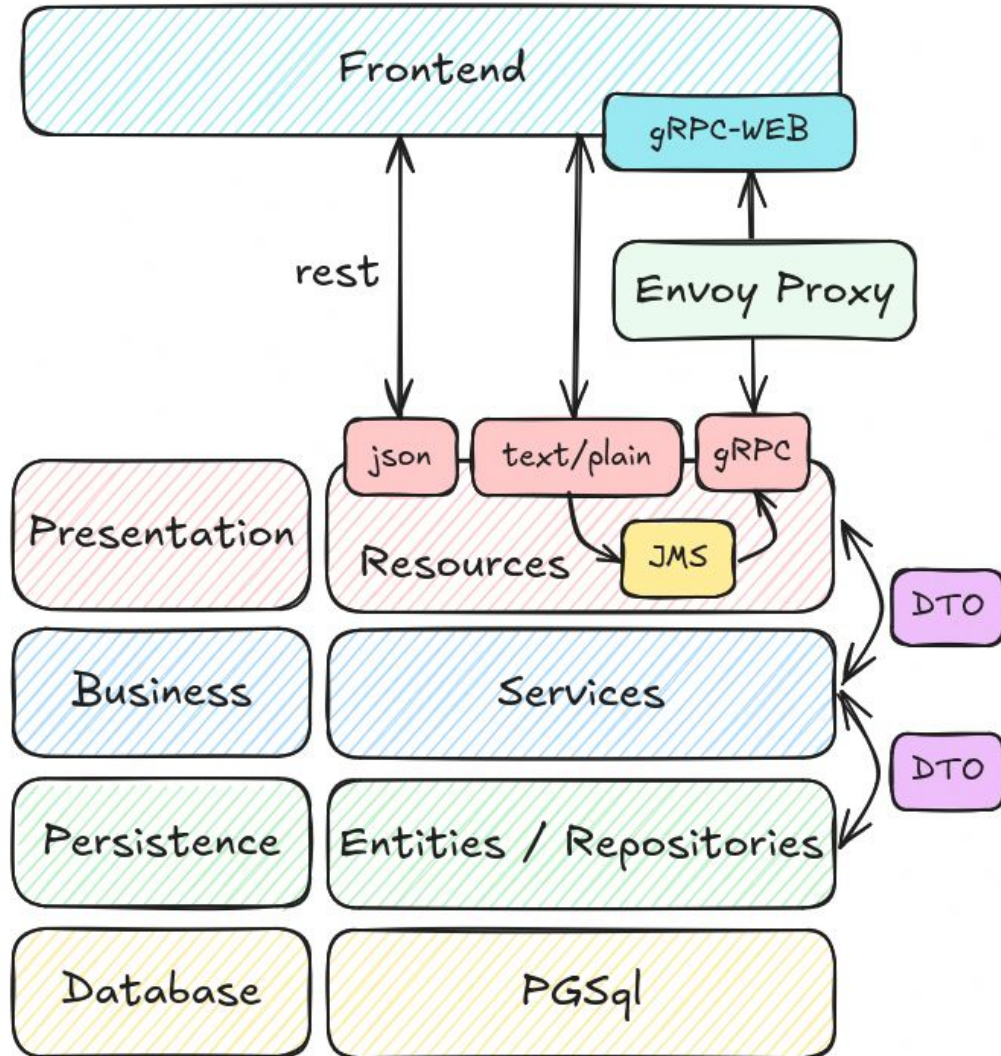
```
// Création du client
const transport = new GrpcWebFetchTransport({
  baseUrl: "http://localhost:8081",
  format: "binary",
});
const mouseClient = new MouseServiceClient(transport);

// Réception sur le stream
mouseClient
  .getMouseUpdates({ userId: selfId }, { abort: signal })
  .responses.onMessage((p) => mousesMap.set(p.userId, p));

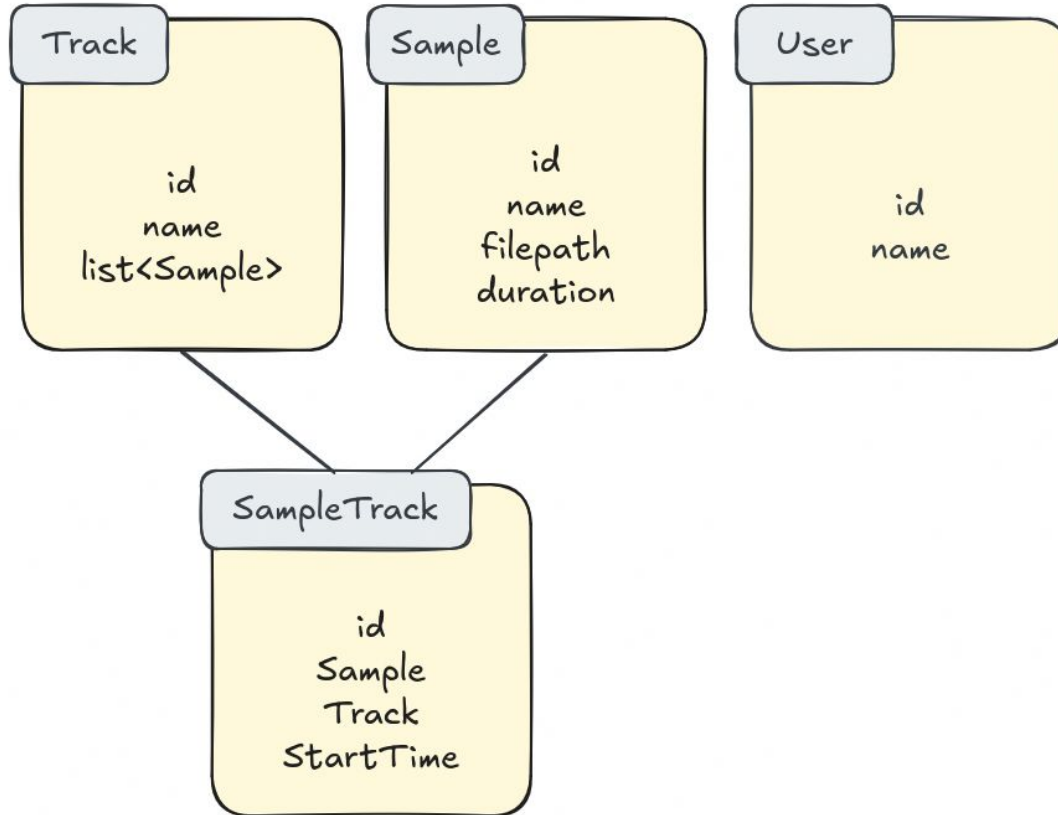
// Envoi
mouseClient.sendMousePosition({ userId: selfId, x: currentX, y: currentY });
```

```
// @generated from protobuf message mouse.MousePosition
export interface MousePosition {
  // @generated from protobuf field: uint32 userId = 1;
  userId: number;
  // @generated from protobuf field: uint32 x = 2;
  x: number;
  // @generated from protobuf field: uint32 y = 3;
  y: number;
}
```

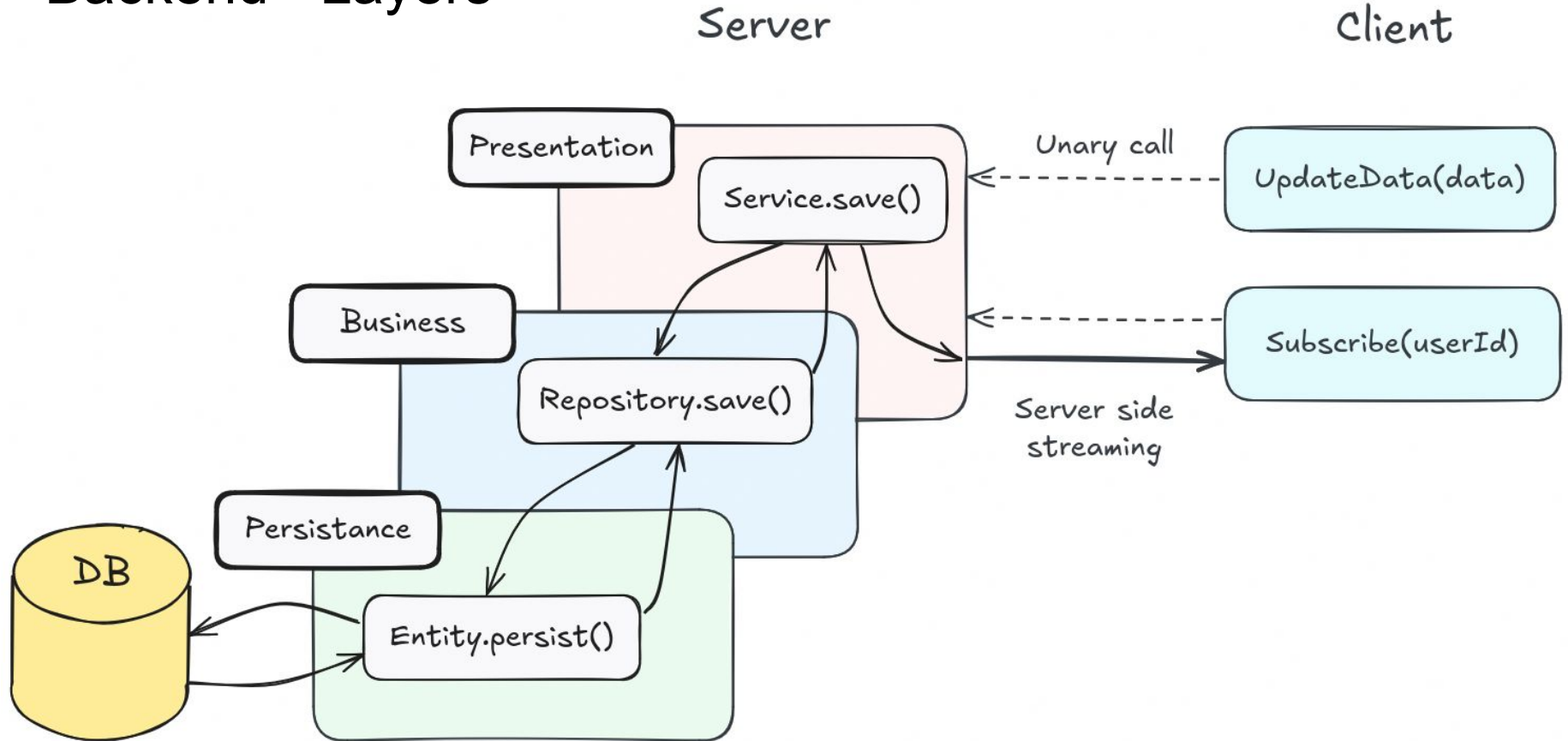
# Architecture



# Backend - Database



# Backend - Layers





# Backend - JMS

Client

/leave

text/plain

UserResource

UserService.delete()

Server

JMS Producer

Other layers

user left

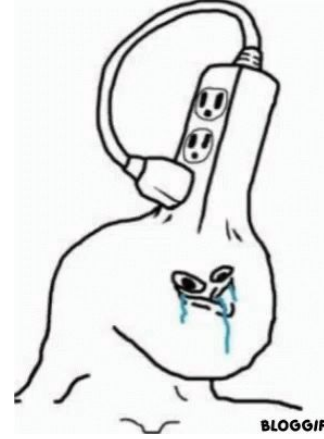
gRPC

Server side streaming

UserGrpcService

broadcast()

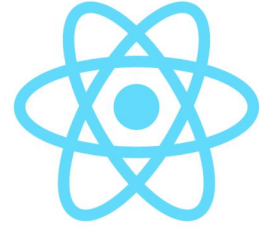
JMS Consumer



BLOGGIF

# Frontend

- Frameworks : **React, Tailwind**
- **Typescript**
- **Composants**
  - App, LoginDialog, Toaster, MouseCursor
  - Library, UploadDropZone, SampleItem
  - TrackList (**WaveSurfer**)





# Problèmes rencontrés - génération des fichiers client

- Installation de **protoc-gen-js**
- Génération des fichiers **xxx\_pb.js** et **xxx\_grpc\_web\_pb.js**
- Traduction CommonJS -> ES Module
- Changement pour **protobuf-ts**
- Ne fonctionne pas sur Windows



# Problèmes rencontrés - gRPC-web

- Limité par le navigateur
- Protocole HTTP 1.1 vs HTTP 2.0
- Manque de certains type de requête
- Nombre de canaux de streaming ouvert limité à 6
- Utilise [EventSource](#) (SSE) derrière les décors
- Nécessite un Proxy

	gRPC	gRPC-web
Protocole	HTTP 2.0	HTTP 1.1
unary call	✓	✓
server side streaming	✓	✓
client side streaming	✓	✗
bidirectional streaming	✓	✗

# Problèmes rencontrés - Déconnexion d'un utilisateur

- **Événements "beforeunload" et "unload"**
  - Pratiques déconseillées et obsolètes pour l'envoi d'analyses.
- **Uniquement des opérations synchrones**
  - Impossible d'exécuter des tâches asynchrones lors de ces événements.
- **Utilisation recommandée : `navigator.sendBeacon`**
  - Permet d'envoyer des données avant la fermeture de la page.
  - Requiert un format de données : `text/plain`.