

Projet Majeure : Show Room Manager

*Côme BURGUBURU
Hugues CHARBON
Benjamin JORNET*

Plan

- ① Présentation fonctionnelle du projet
- ② Choix architecture et technologique
- ③ Choix organisationnels
- ④ Démonstration
- ⑤ Conclusions



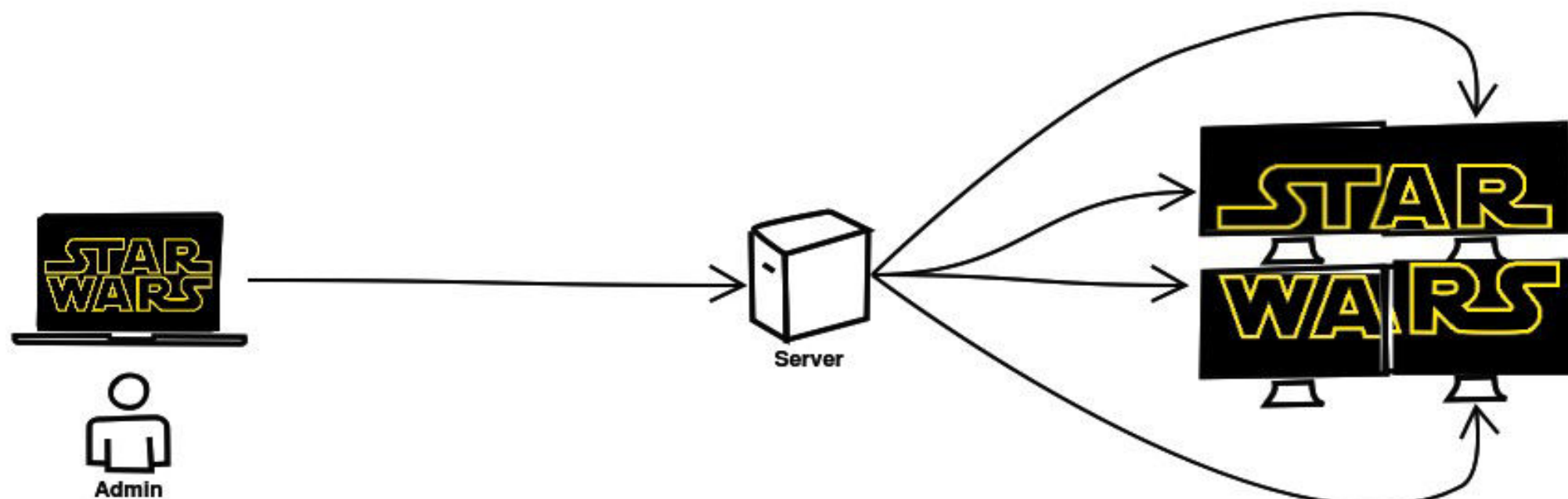
1. Présentation fonctionnelle du projet

Description

- ▶ Show Room Manager permet de visualiser sur différents écrans distants des flux.
- ▶ Types de flux : image, vidéo, facebook, twitter.
- ▶ Communication et identification des écrans via sockets.
- ▶ Application Web.
- ▶ Possibilité d'upload des fichiers via l'écran d'admin.
- ▶ Disponible en application mobile.

1ère utilisation de l'application

- Découpage d'un flux sur plusieurs écrans

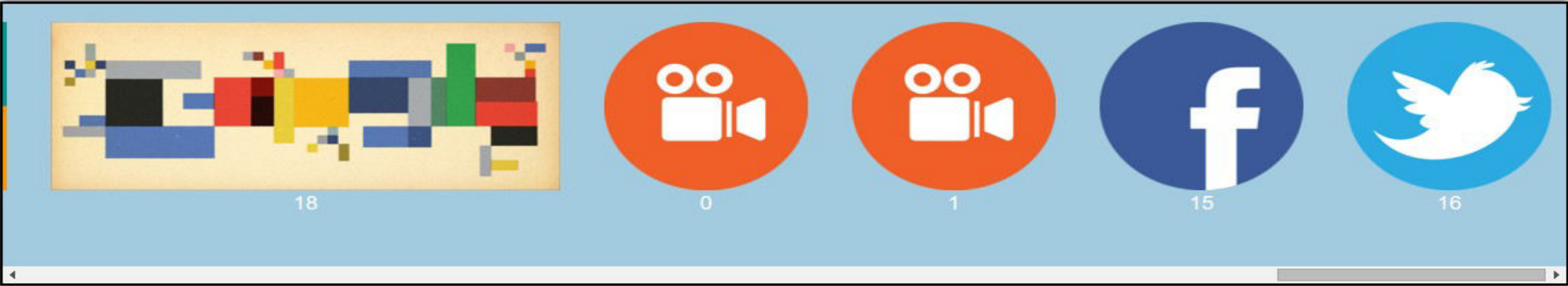


1ère utilisation de l'application

localhost:1337/index.html#/

Ambiance Watcher 0 Pause Activer grid ☒


Liste d'images



18 0 1 15 16

Composition d'écrans

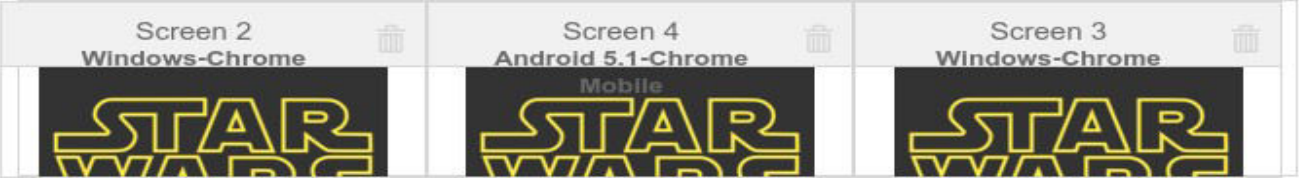
Ecran



Screen 2
Windows-Chrome

Screen 4
Android 5.1-Chrome
Mobile

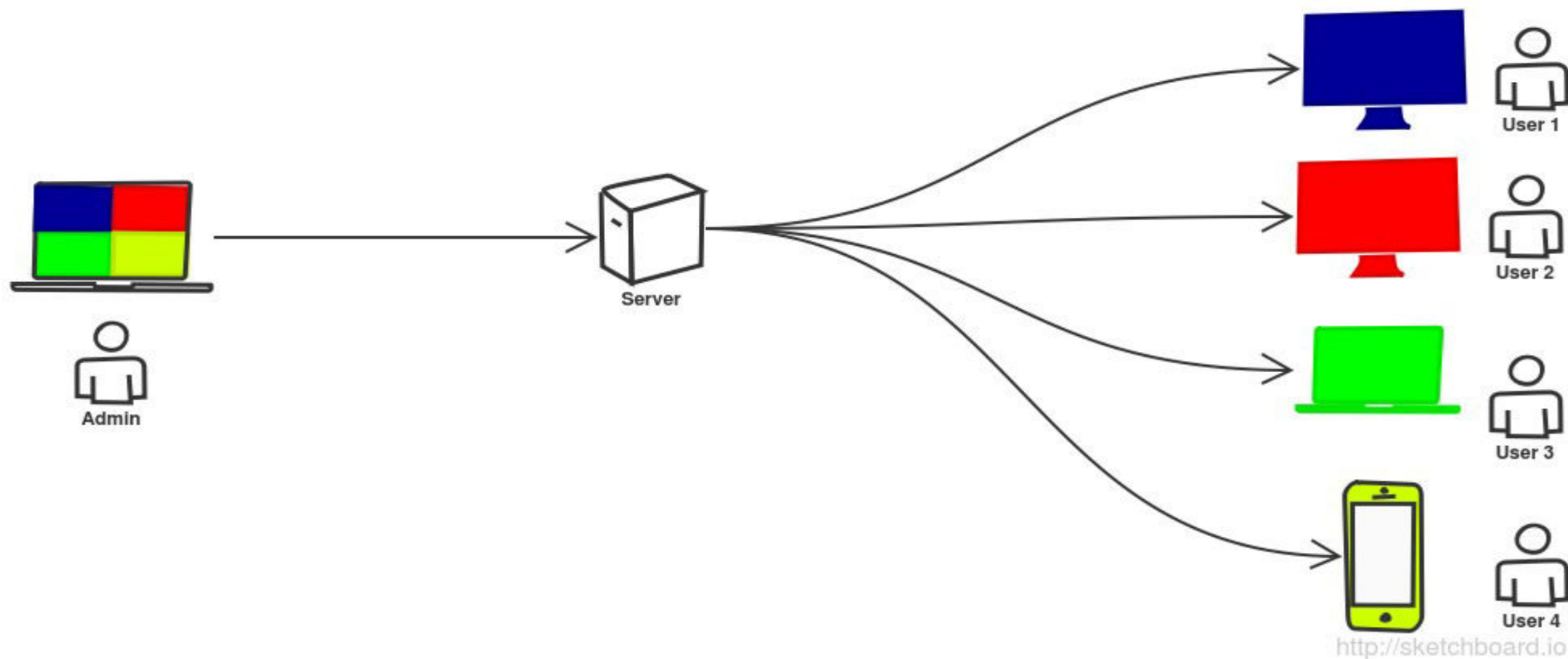
Screen 3
Windows-Chrome



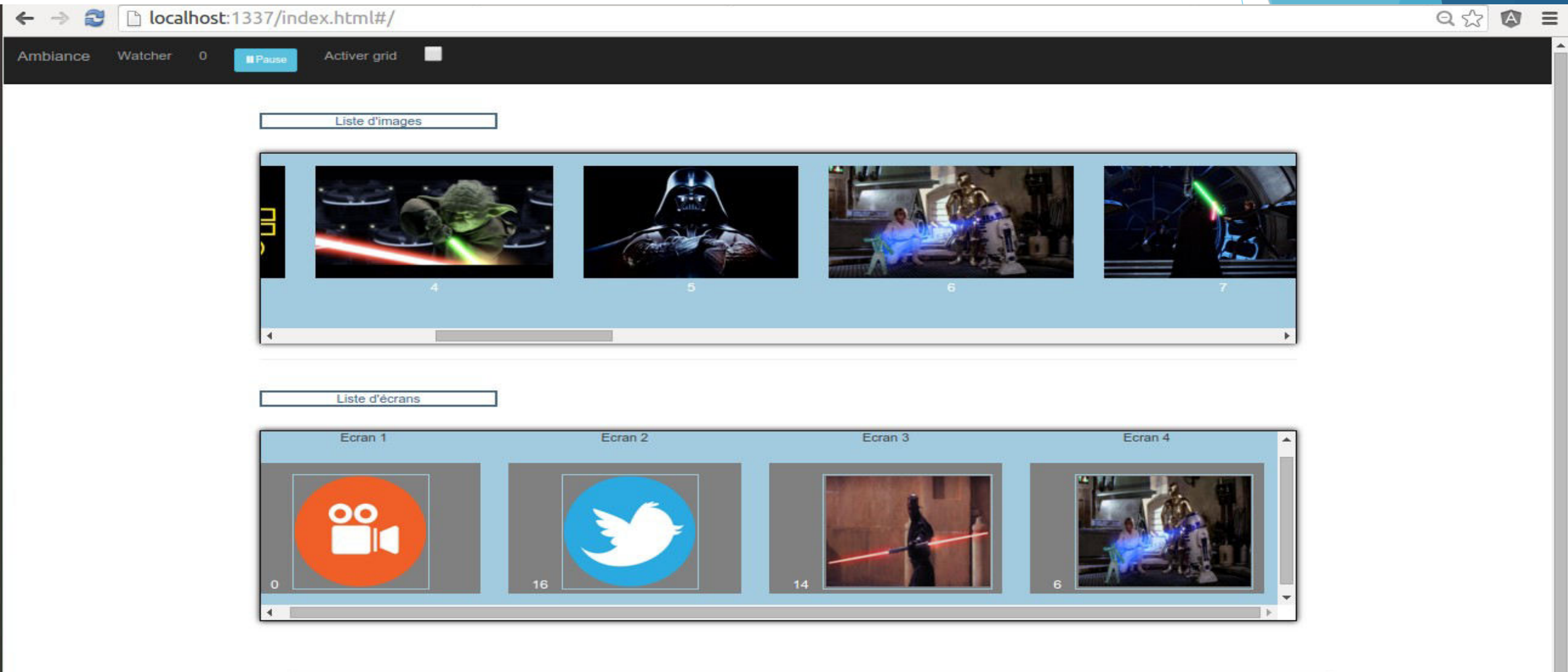
Ambiance Show Room - BURGUBURU Côte , CHARBON Hugues , JORNET Benjamin

2ème utilisation de l'application

- Découpage de différents flux sur plusieurs écrans



2ème utilisation de l'application

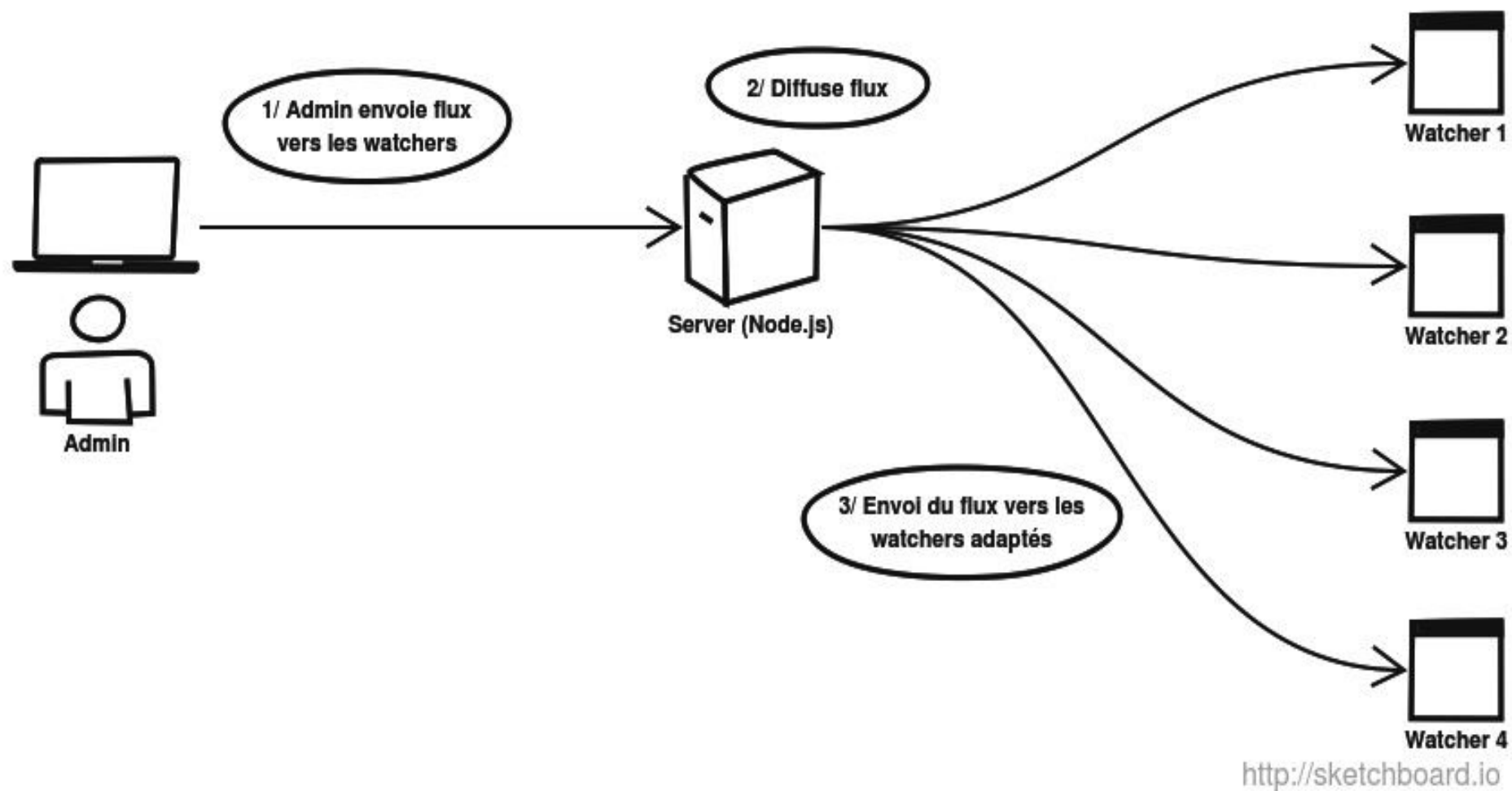


2. Choix architecture et technologique

Choix d'architecture

- ▶ Show Room Manager

Fonctionnement de l'application



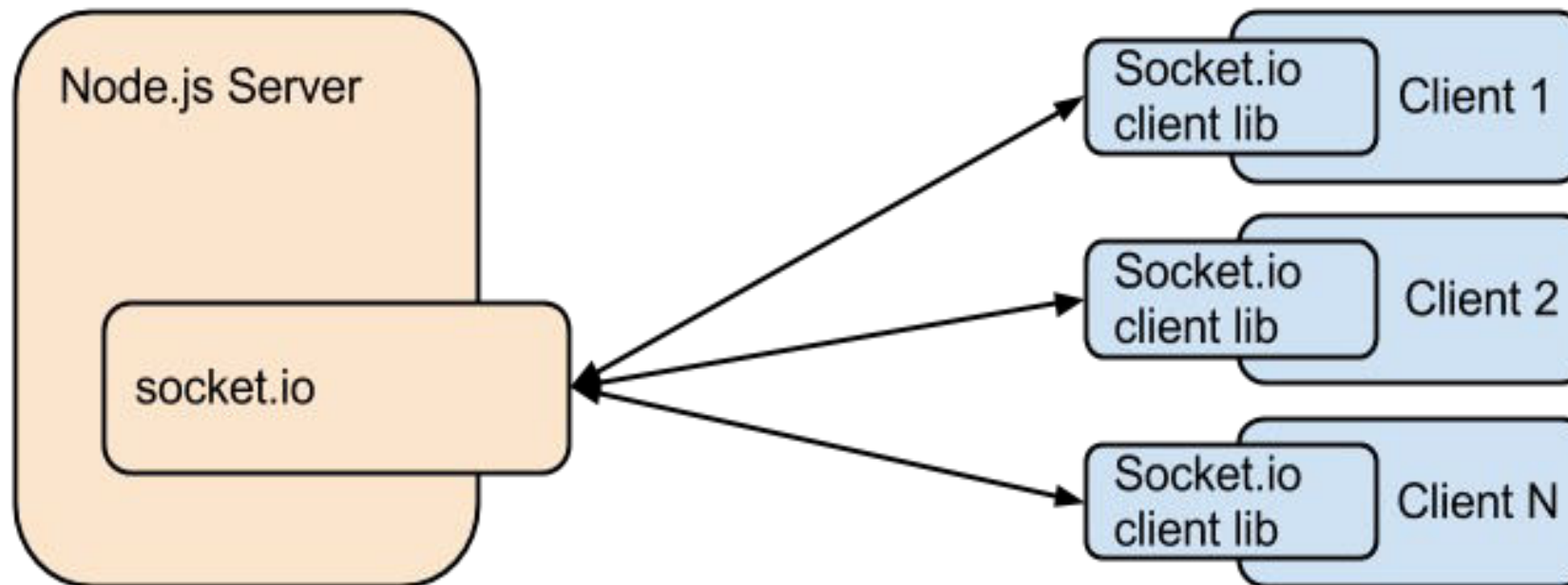
Choix technologiques

- ▶ NODE.JS → Côté serveur
- ▶ ANGULAR.JS → Front-end
- ▶ SOCKET.IO → Communication entre Admin et watchers
- ▶ HTML5-JAVASCRIPT → Affichage côté client
- ▶ MOBILE CORDOVA / ANDROID → Compatibilité mobile
- ▶ Outil GIT → Synchronisation du code

Fonctionnement Nodejs



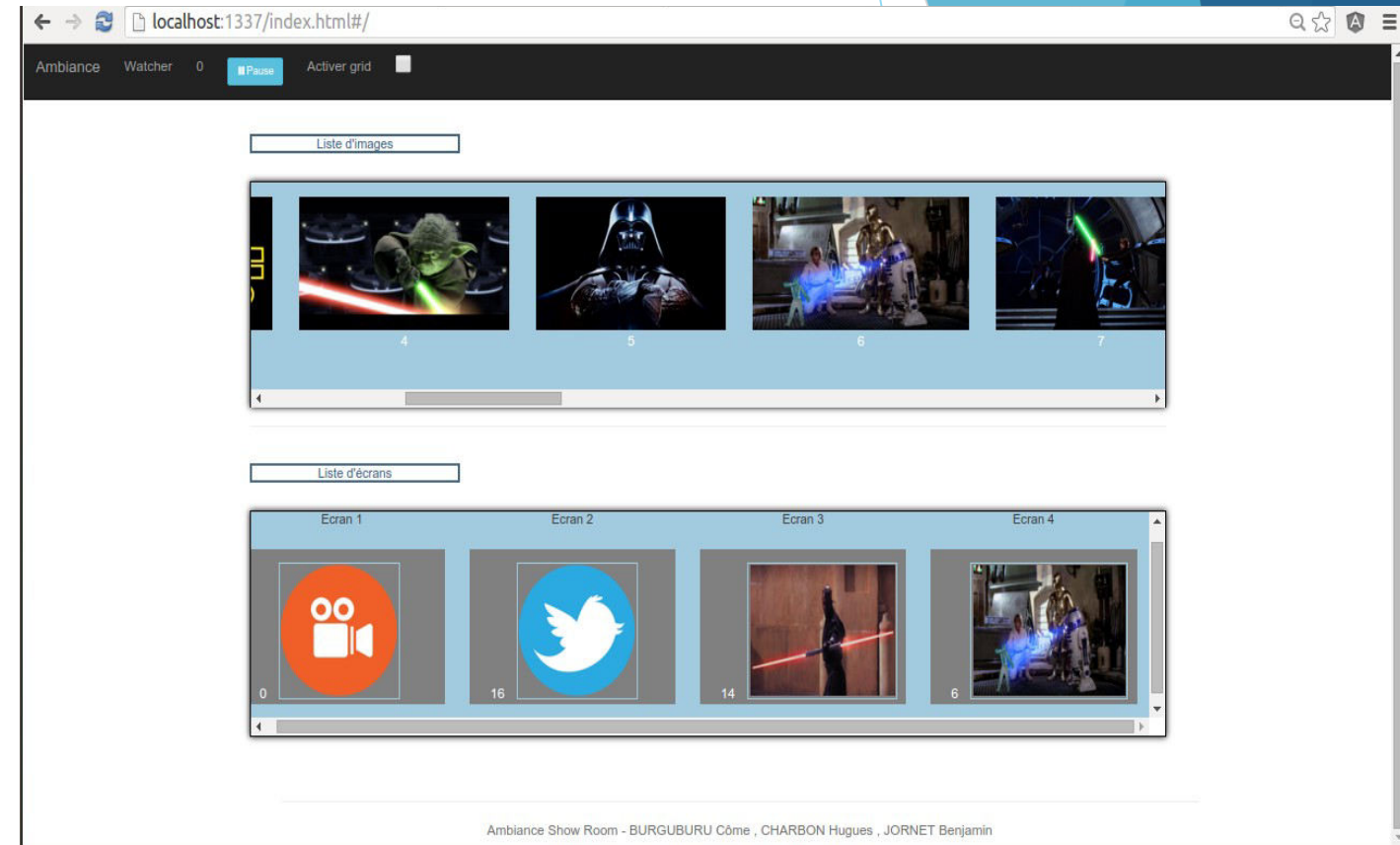
- ▶ Utilisation de socket.io
- ▶ Les clients sont les watchers
- ▶ Diffusion a toutes les sockets



Fonctionnement Angularjs



- ▶ Un controller d'événement utilisateur couplé à un service qui écoute la socket
- ▶ ng-draggable -> pour le drag&drop
- ▶ Controller : gère événement utilisateur
- ▶ Service : gère les sockets



Fonctionnement communication par sockets

- ▶ Utilisation de socket.io
- ▶ Socket emit "image" id et URL
- ▶ SCHEMA



socket.io

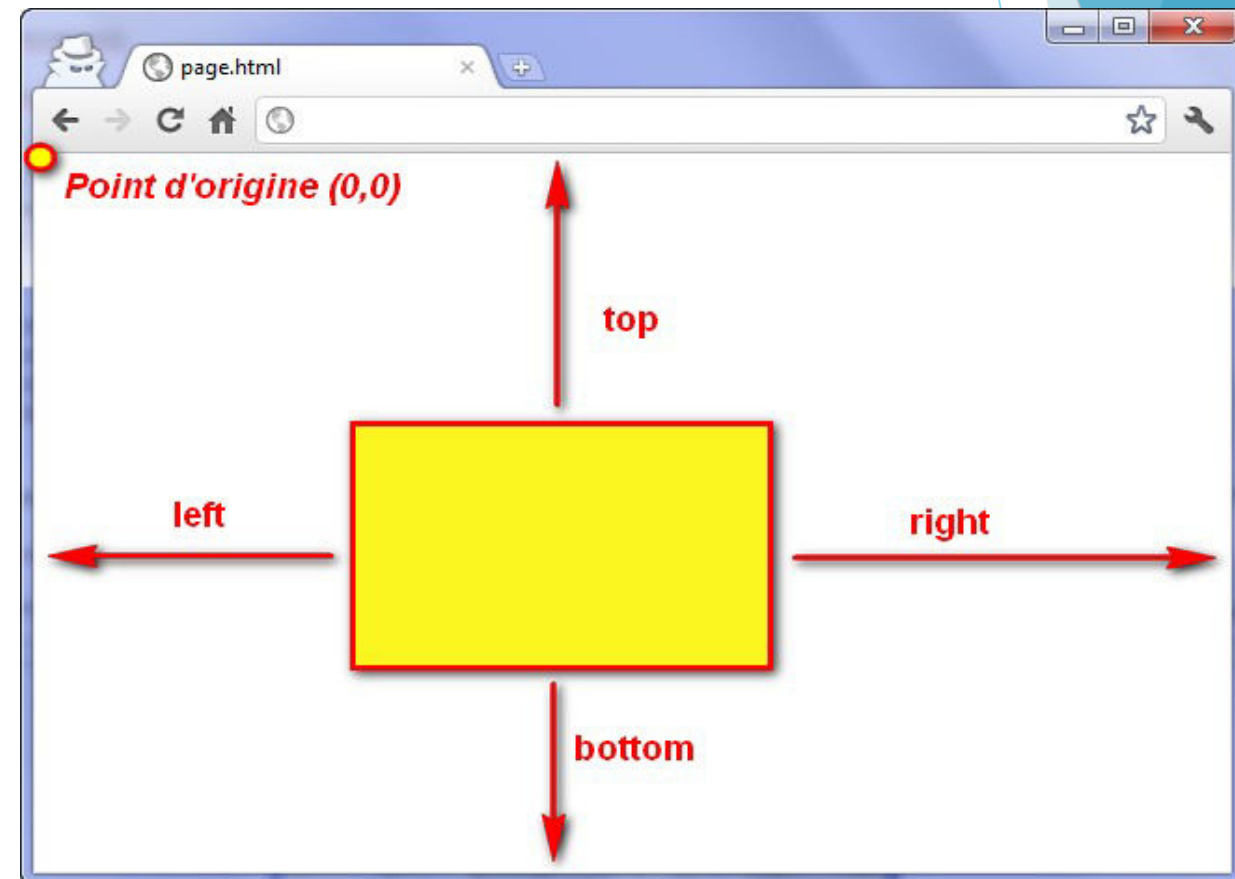
Fonctionnement HTML5-JAVASCRIPT

- ▶ Représente la vue de l'application
- ▶ Utilisation de la librairie gridster.js -> Mode grille
- ▶ ng-gridster (évènement drop) -> mise à jour de la configuration des écrans côté serveur
- ▶ Découpage de l'image en fonction de la grille
- ▶ Attribution de top, left, width, height par fragment d'image



Fonctionnement watcher

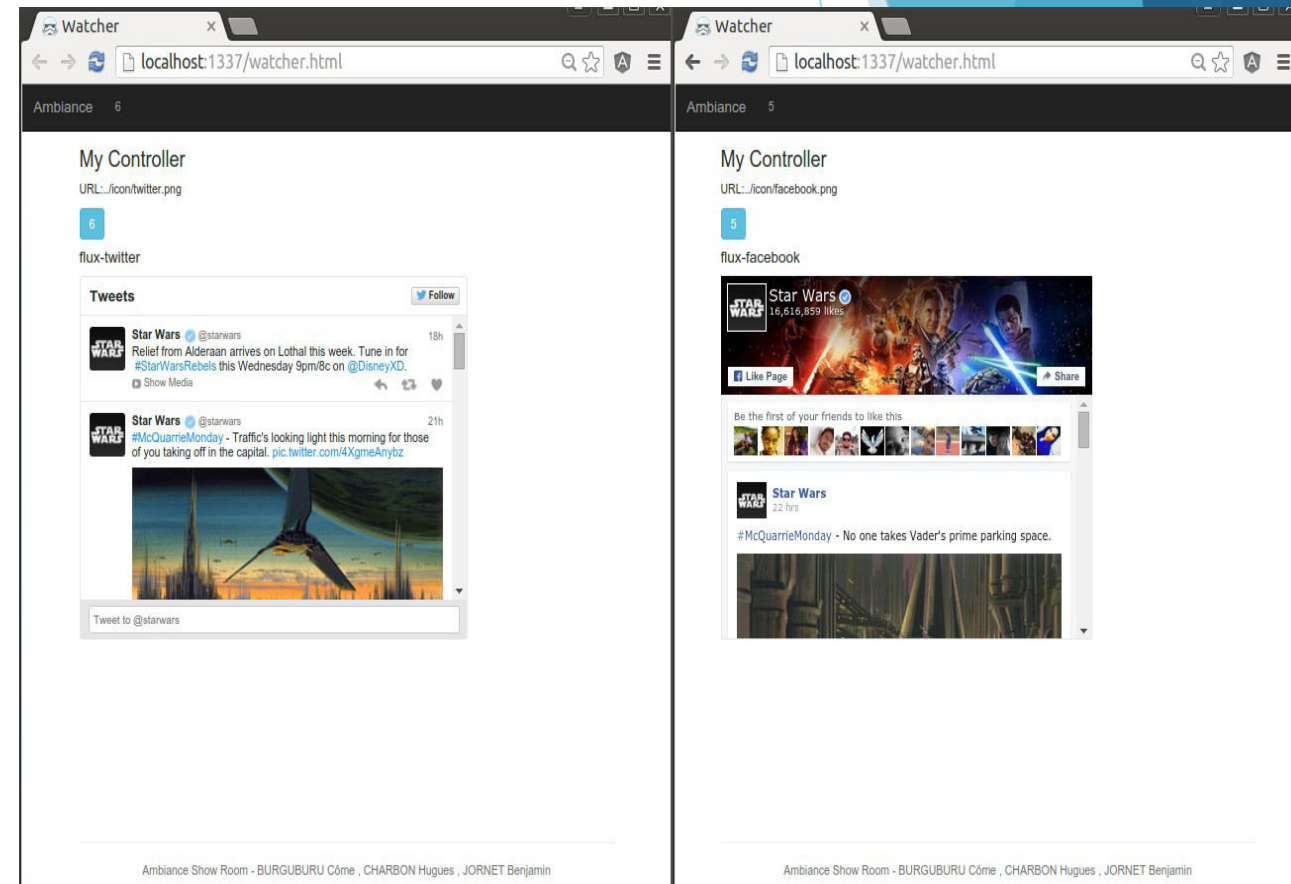
- ▶ Watcher interprète la réponse du serveur
- ▶ Top, left, width, height
- ▶ API Fullscreen



Fonctionnement des flux

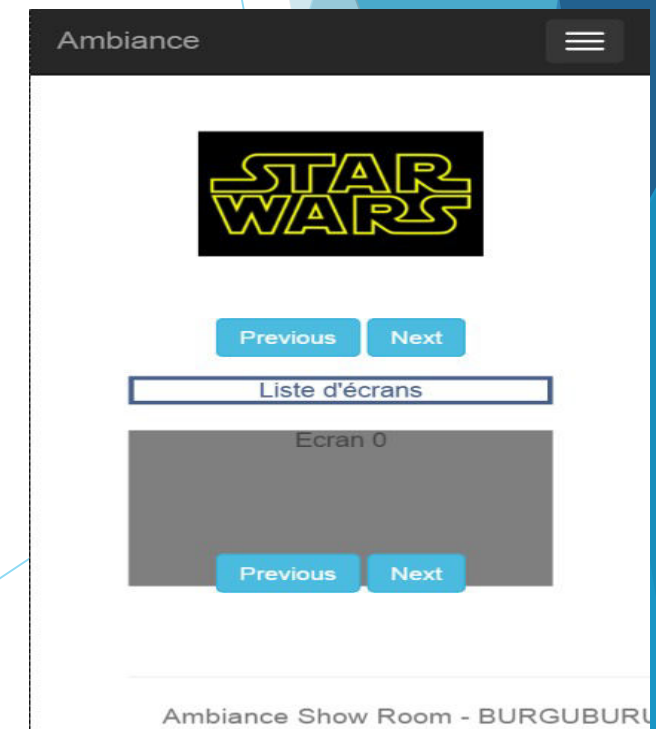
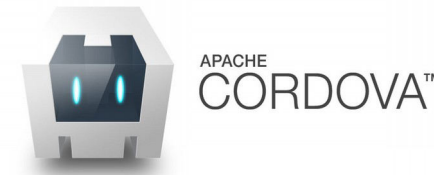


- ▶ Différents types de flux : image, vidéo, Facebook, Twitter
- ▶ Les images sont des représentations visuels d'objet complexe
- ▶ Utilisation d'API Facebook et Twitter
- ▶ Récupération des flux grâce à l'URL et l'ID de la page



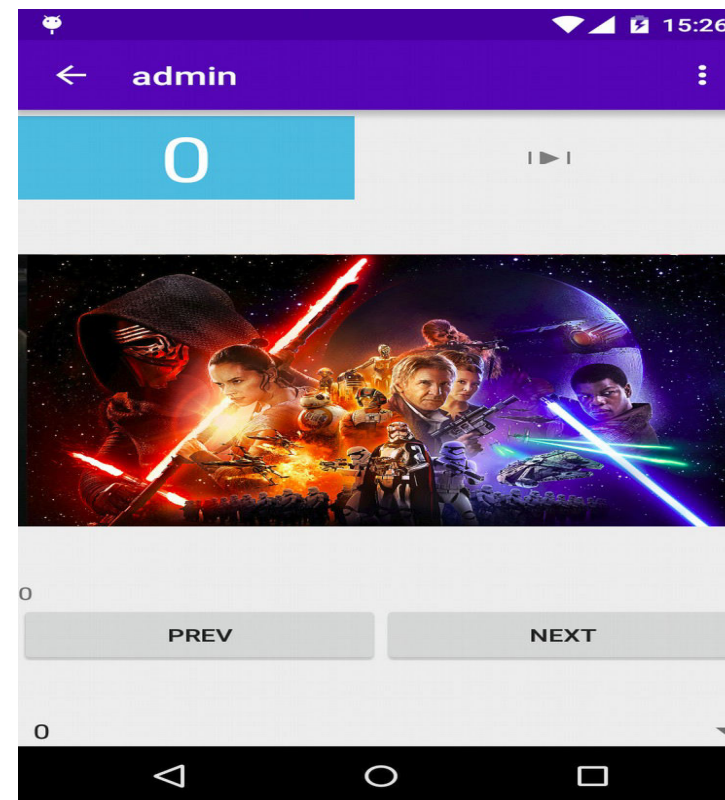
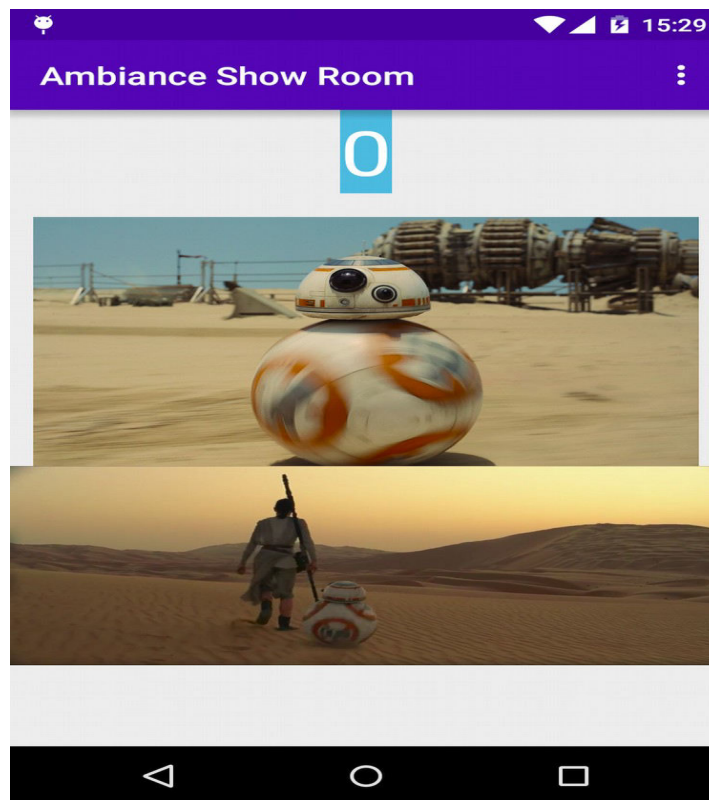
Fonctionnement MOBILE CORDOVA

- ▶ Compatibilité multi-plateforme
- ▶ 2 apk : 1 admin + 1 watcher
- ▶ Interface responsive
- ▶ Utilisation de select à la place de la grille



Fonctionnement MOBILE ANDROID

- ▶ Dépendances : okhttp, picasso, io.socket



3. Choix organisationnels

Répartition en tâches

N° Step	Description de la tâche
Step 1	Identifier + lister des watchers
Step 2	Transférer une image sur le watcher de notre choix
Step 3	Diffuser une image découpée sur plusieurs écrans
Step 4	Configurer la grille des écrans depuis l'administrateur
Step 5	Pouvoir transmettre des flux Facebook et Twitter
Step 6	Pouvoir transférer une vidéo
Step 7	Découper une vidéo
Step 8	Synchroniser la lecture d'une vidéo
Step 9	Avoir une version plateforme mobile

Démonstration

5. Conclusions

➡ Bilan Fonctionnel :

- ▶ Réalisation des 9 steps

➡ Perspectives d'amélioration :

- ▶ Possibilité de déployer des jeux en ligne via les watchers
- ▶ Récupération de flux streaming
- ▶ Optimisation du front-end
- ▶ Identification des utilisateurs via des comptes

Merci pour votre attention
Avez-vous des questions ?