

# GLASS

## Hackathon SG connected Hack

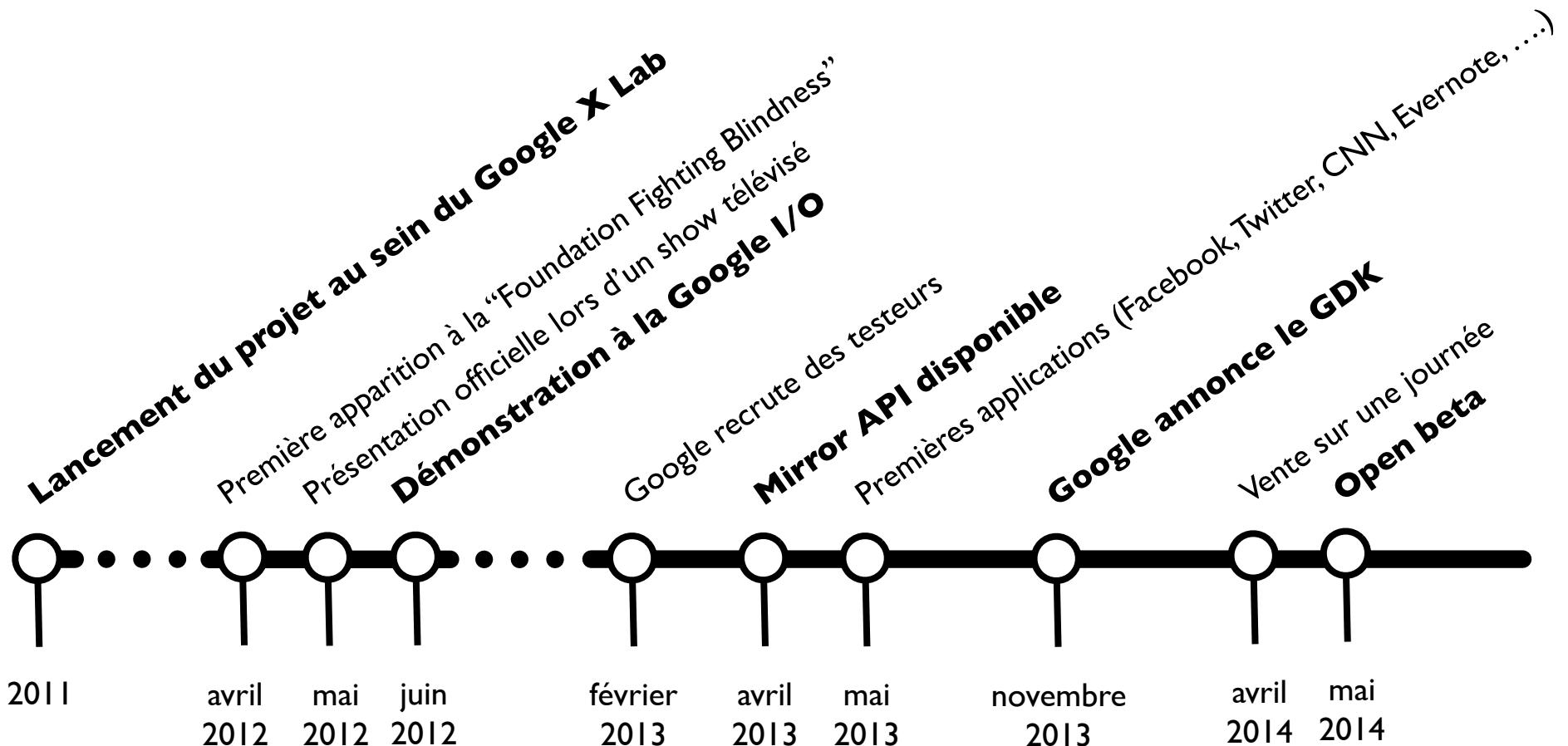
Mathieu Hausherr



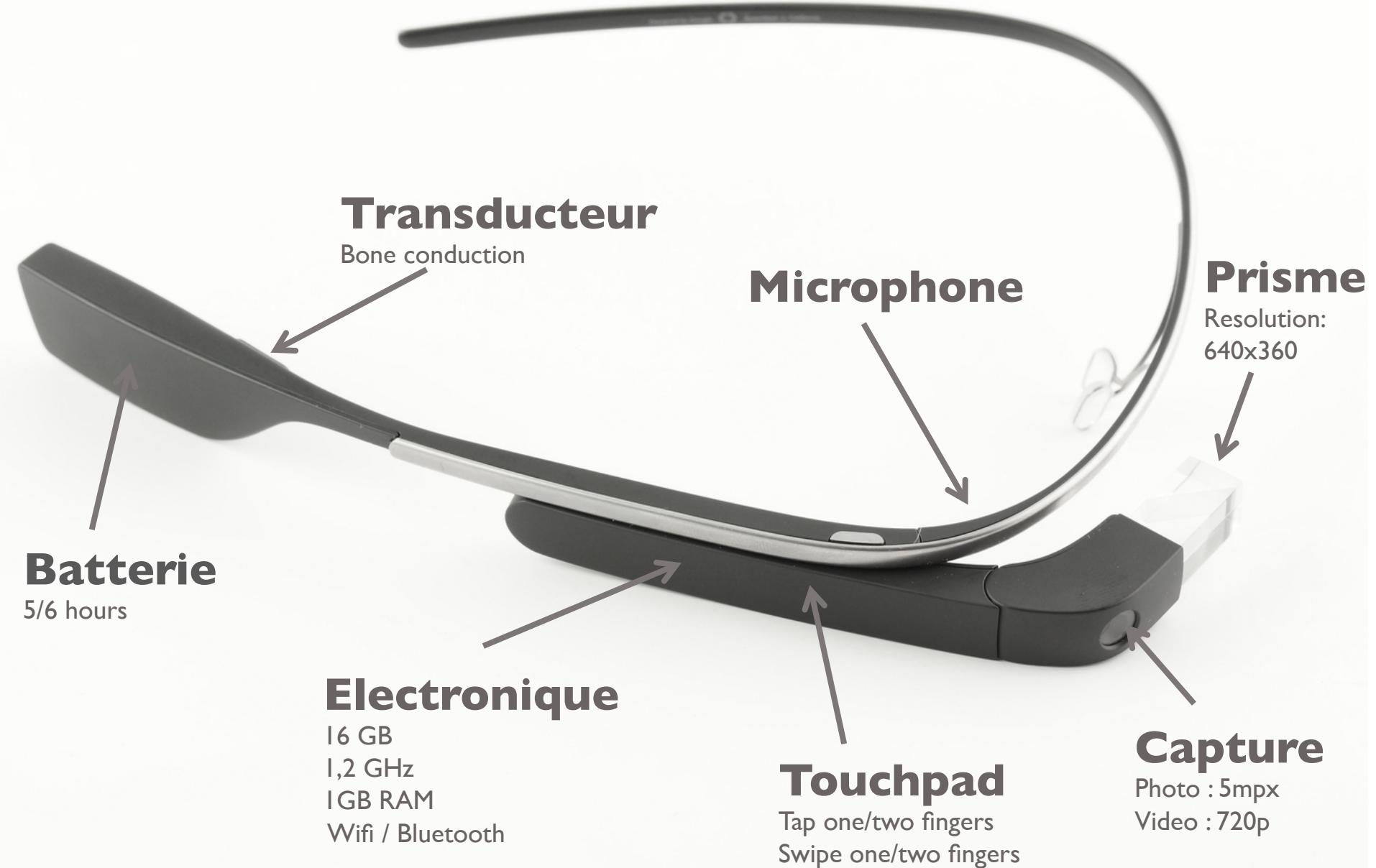
# **Les Google Glass et ses contraintes**

**Comment développer sur les Google Glass ?**

**Votre première application**



Un petit bijou technologique ...



... ayant les mêmes fonctionnalités qu'un smartphone ...

- > Prendre des photos
- > Enregistrer des vidéos
- > Rechercher sur Google
- > Jouer de la musique
- > Envoyer des messages
- > Effectuer des appels
- > Utiliser la navigation
- > Recevoir des notifications
- > Prendre des notes
- > ...

ok glass, google...  
take a picture  
record a video  
get directions to...  
send a message to...  
make a call to

- > Par la voix
  - + « Ok, glass » + commande
- > Par le trackpad
  - + Avec différentes Gestures (taps, swipes, one finger, two fingers)
- > Par des boutons physiques
  - + Pour prendre un photo ou enregistrer une vidéo
  - + Pour allumer/éteindre les Google Glass
- > Avec son corps (tête, yeux, ...)
  - + La détection de mouvement (grâce aux capteurs, accéléromètre, gyroscope, ...)
  - + Le clin d'oeil droit
- > Les applications “My Glass”
  - + Sur smartphone (iOS / Android)
  - + Sur le web (<https://glass.google.com/u/0/myglass>)

ok glass, google...  
take a picture  
record a video  
get directions to...  
send a message to...  
make a call to...

... possédant un petit écran ...



... ainsi qu'une batterie limitée



# Les Google Glass et ses contraintes

**Comment développer sur les Google Glass ?**

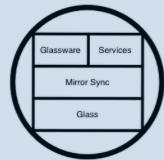
**Votre première application**

Deux options pour vos apps

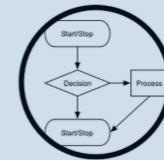
## 1. La Mirror API



Platform independence



Common infrastructure



Built-in functionality

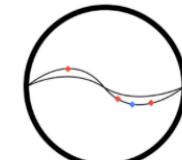
## 2. La Glass Developpement Kit



Real-time User Interaction



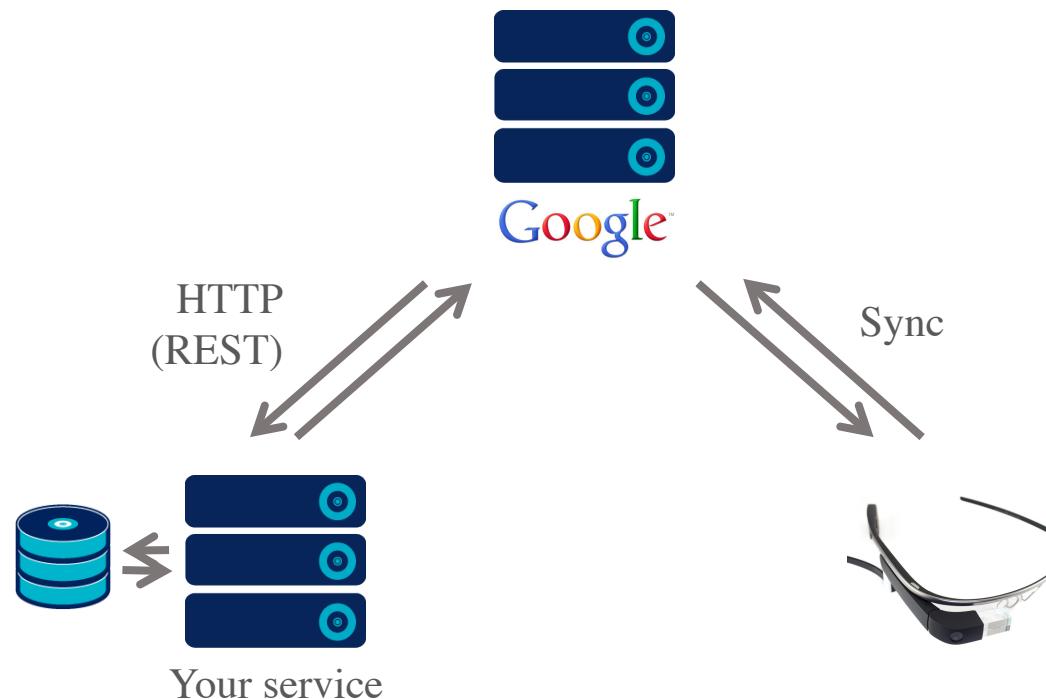
Offline Functionality



Access to Hardware

## > Le code tourne sur le serveur

- + RESTFull
- + Authentification avec OAuth2.0
- + Librairies en Java, PHP, .NET, Ruby, Python ...

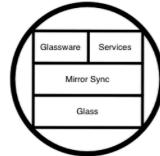


Deux options pour vos apps

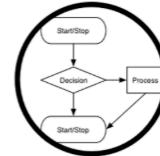
## 1. La Mirror API



Platform independence



Common infrastructure



Built-in functionality

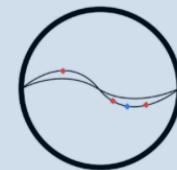
## 2. La Glass Developpement Kit



Real-time User Interaction



Offline Functionality

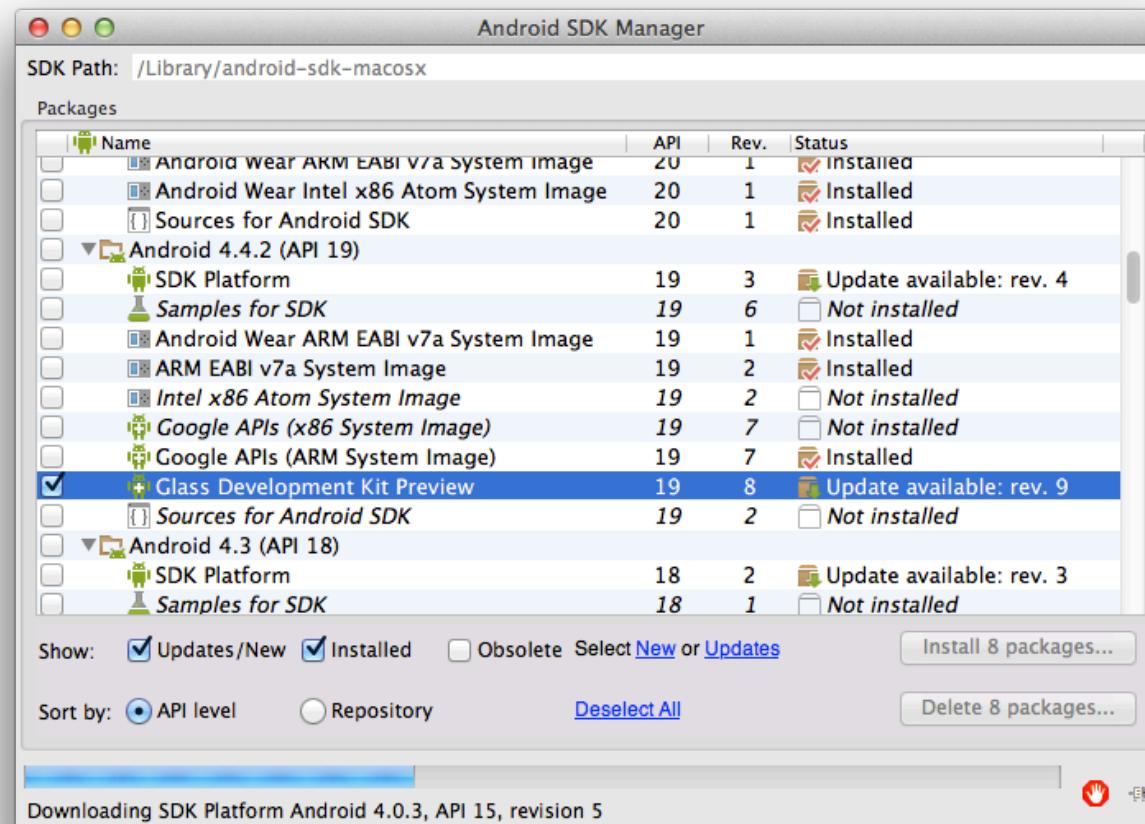


Access to Hardware

## Glass Development Kit



# Installation



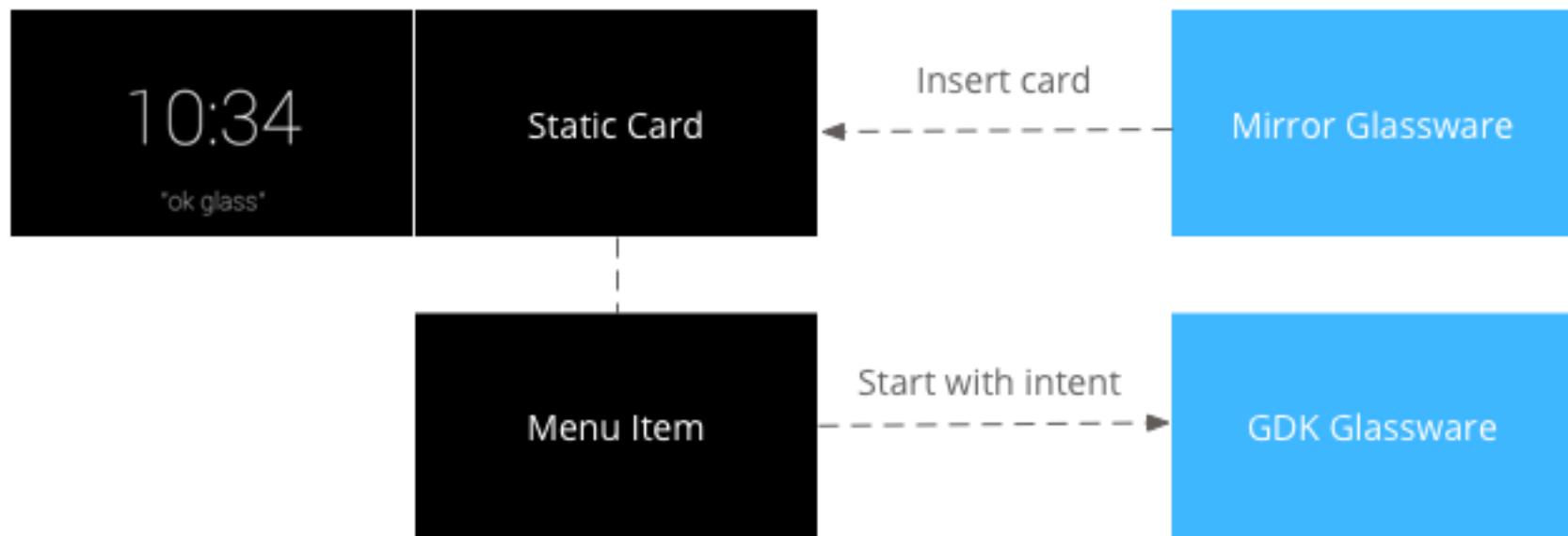
## > Le code Android classique fonctionne sur les Glass

- + Le GDK est en version Preview
  - o Il est disponible directement dans le SDK manager
- + Il ajoute des fonctionnalités dédiées
  - o La gestion de la voie
  - o La gestion des gestes du corps
  - o Un nouveau type d'interface: les Cards



## Vous pouvez utiliser les deux !

Une notification (Card) poussée avec la Mirror API peut lancer une application native (GDK)



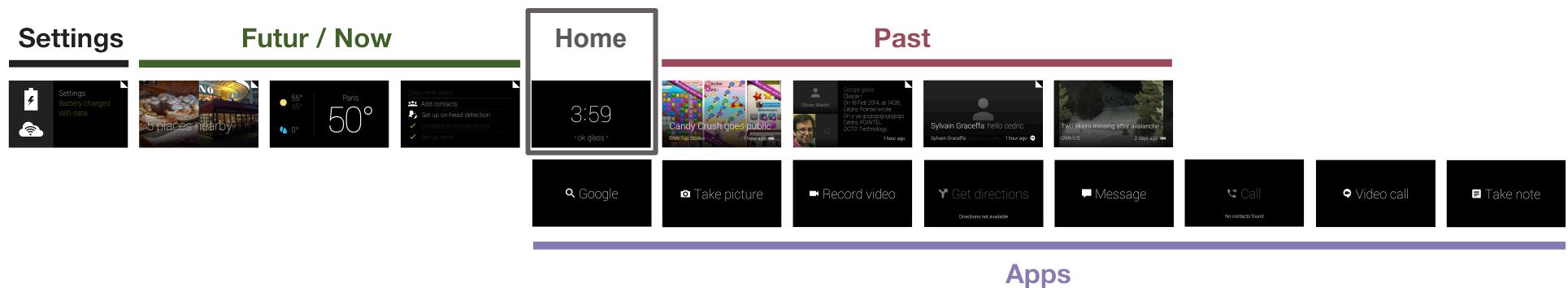
# Les Google Glass et ses contraintes

Comment développer sur les Google Glass ?

**Votre première application**

## > Une vision globale divisée en 4 parties

- > Le passé
- > Le futur / présent
- > Les applications appelées « Glasswares »
- > Les paramètres pour configurer les Google Glass (WIFI, Bluetooth, ...)



- > **Solution** : le GDK de base fournit des types d'écran à utiliser afin d'afficher de l'information

Static cards

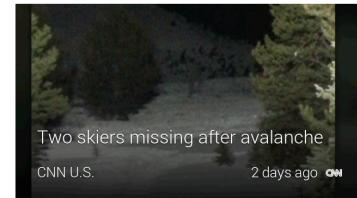
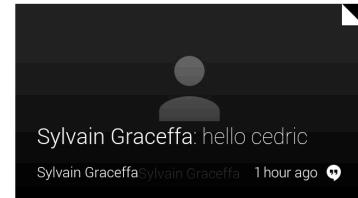
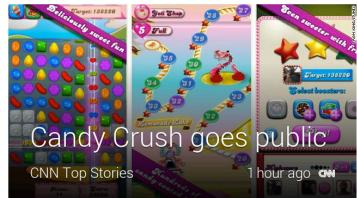
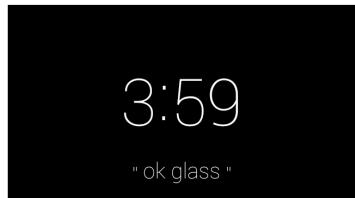
Live cards

Immersion

## Static cards

- > De simple vue
- > Pour afficher du texte, des images, des vidéos ou du HTML
- > Utilisé pour afficher des notifications (→ Mirror API)
- > Possibilité d'y invoquer des LiveCards ou de l'Immersion
- > Ne peut pas être rafraîchie fréquemment

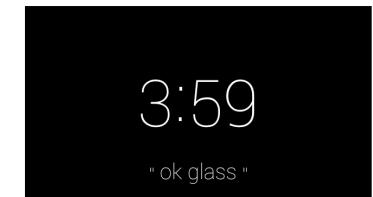
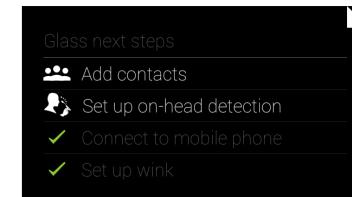
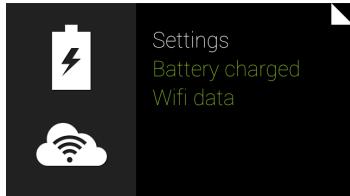
## Past



## Live cards

- > Information Temps-réel
- > Accès à certaines capacités des Google Glass (ex : accéléromètre, gyroscope, camera, ...)
- > S'exécute au sein de la timeline
- > Peut être mis à jour fréquemment
  - > Low Frequency Livecards (de l'ordre de la seconde)
  - > High Frequency Livecards

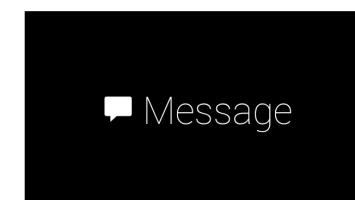
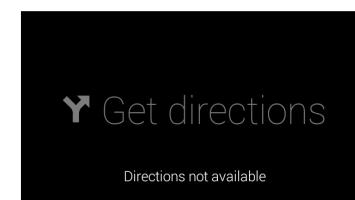
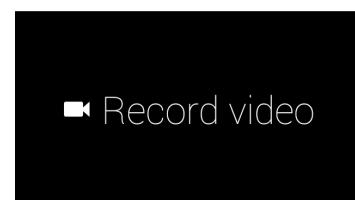
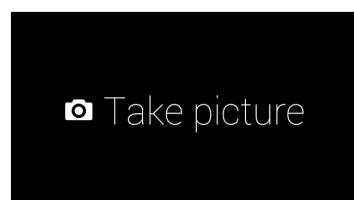
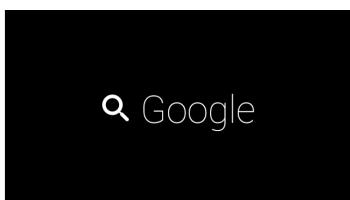
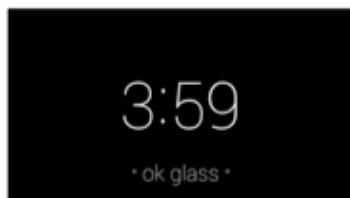
## Settings



## Futur / Now

# Immersion

- > S'exécute en dehors de la TimeLine
- > Développer votre propre interface (→ Activity Android)
- > Accès à l'ensemble des capacités des Google Glass (ex : les capteurs, la voix, les gestes, ...)
- > Attention à respecter les contraintes Google (taille de l'écran, les interactions, gestes)
- > Spécifier dans le Manifest `android:immersive="true"` pour éviter que les Glass se mettent en veille



## Apps

- > Désormais une classe magique : CardBuilder → elle permet de générer un layout selon différent styles intégrés au GDK

```
View view = new CardBuilder(context,  
CardBuilder.Layout.COLUMN)  
    .setText("Bonjour, bienvenue au hackaton connected hack")  
    .setFootnote("Avec la Société Générale")  
    .addImage(R.drawable.image)  
    .getView();
```



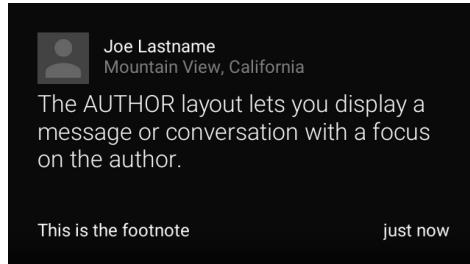
.Layout.CAPTION



.Layout.TITLE



.Layout.ALERT



.Layout.AUTHOR



.Layout.ALERT



.Layout.COLUMN

There is a better way

- > Les commandes vocale via l'écran d'accueil → afin de lancer directement un Glassware depuis l'écran d'accueil
  - + Une ressource XML pour la commande vocale selon une liste définie par Google (→ VoiceTriggers.Command)

```
res/xml/voice_trigger.xml
```

```
<?xml version="1.0" encoding="utf-8"?>
<trigger keyword="SEND_MONEY" />
```

Exemple de commandes :

"SEND_MONEY"	"START_A_RUN"
"FIND_A_PLACE"	"START_A_STOPWATCH"
"FIND_A_HOSPITAL"	"START_A_TIMER"
"FIND_PARKING"	"TAKE_A_PICTURE"
"SHOW_ME_MY_SPEED"	"TAKE_A_NOTE"
"SHOW_ME_THE_NEWS"	"TRANSLATE_THIS"
"SHOW_ME_THE_WEATHER"	"PLAY_A_GAME"

- > Les commandes vocale via l'écran d'accueil → afin de lancer directement un Glassware depuis l'écran d'accueil
  - + Déclarer un Intent Filter au sein du Manifest de l'application avec également la possibilité de fournir un icône à la commande vocale

```
<activity
    android:name=".ui.activity.MainActivity"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:keepScreenOn="true"
    android:immersive="true">
    <intent-filter>
        <action android:name="com.google.android.glass.action.VOICE_TRIGGER" />
    </intent-filter>

    <meta-data
        android:name="com.google.android.glass.VoiceTrigger"
        android:resource="@xml/voice_trigger" />
</activity>
```

- > Les commandes vocale via l'écran d'accueil → afin de lancer directement un Glassware depuis l'écran d'accueil
  - + Possibilité d'ajouter un prompt afin d'offrir la possibilité à l'utilisateur d'ajouter un message supplémentaire

```
<?xml version="1.0" encoding="utf-8"?>
<trigger keyword="SEND_MONEY">
  <input prompt="@string/how_much_prompt" />
</trigger>
```

```
ArrayList<String> voiceResults =
getIntent().getExtras().getStringArrayList(RecognizerIntent.EXTRA_RESULTS);
```

- + Il est également possible de restreindre l'utilisation de la voix selon la disponibilité de certains services au sein des Google Glass (ex : camera, réseau, ...)