

# IOT - Exa-M

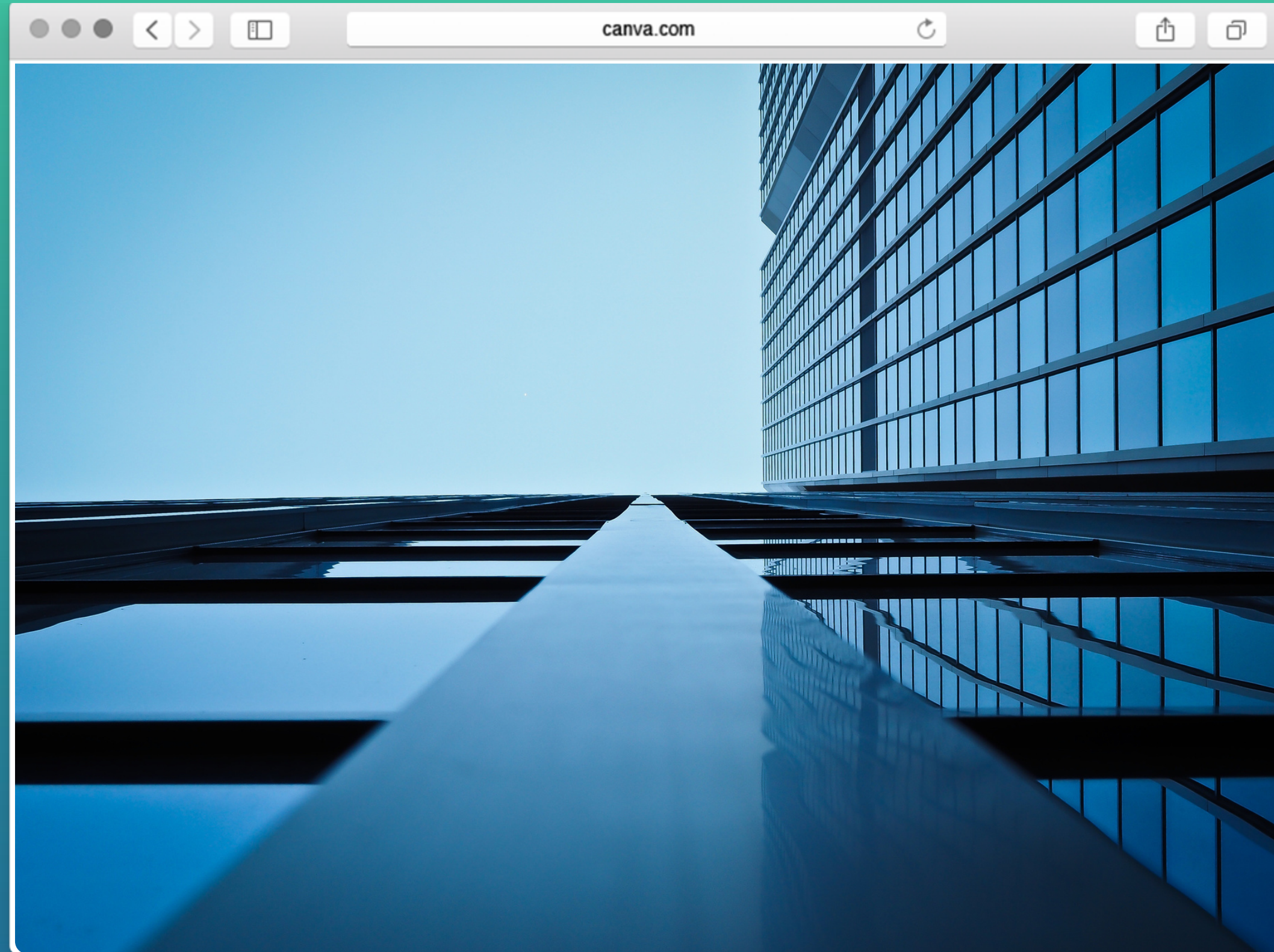
**Alexis CHANDARA**  
**Ismael DKHISSI**

PDG

99 Av. Jean Baptiste Clément,  
93430 Villetaneuse

chandaraalexis@gmail.com  
dkhissi.ismael@gmail.com

01 49 40 30 00



# Problème / Opportunité

Comment détecter une fraude avérée durant un examen en ligne

01

Pouvoir faire une  
détection faciale  
durant l'examen

02

Pouvoir faire une  
détection vocale  
durant l'examen

03

Pouvoir détecter des  
objets durant l'examen

04

Pouvoir détecter les  
mouvements du  
candidat durant  
l'examen



# Exa-M

# Notre produit

Conçu pour lutter contre les fraudes lors d'un examen en ligne.

# Conception du produit

## Connexion

Pouvoir connecter avec fil ou sans fil l'appareil mobile à l'ordinateur

## Serveur

Implémentation d'un serveur permettant de recevoir des données

## Accelerometre

Implémentation d'un accelerometre afin de calculer la vitesse de déplacement du candidat

## Détection facial

Implémentation d'un algorithme d'intelligence artificiel permetant de détecter un ou plusieurs visages

## Détection d'objets

Implémentation d'un algorithme permettant la détection d'objets en temps réel

## Détection vocal

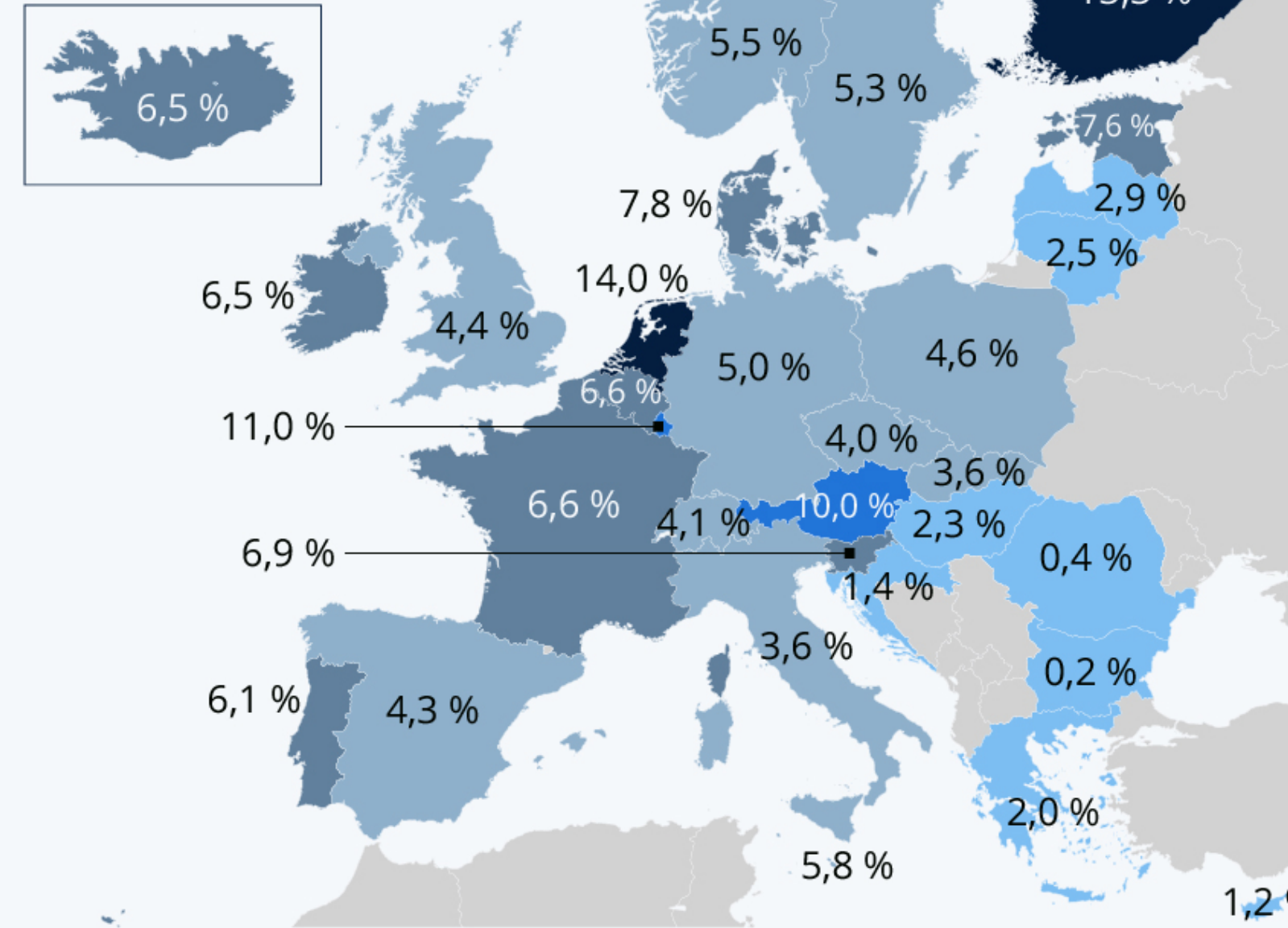
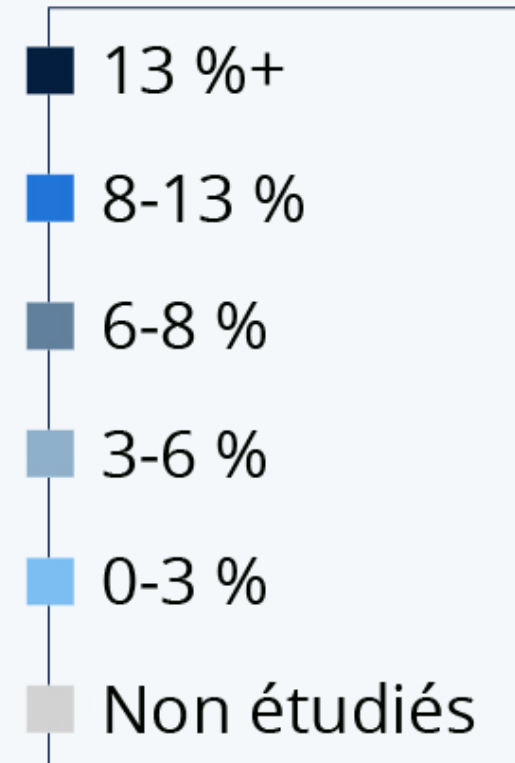
Implémentation d'un algorithme d'intelligence artificiel permettant de détecter si un candidat parle

# La situation actuelle

La pandémie oblige de nombreuses personnes à travailler en distanciel

## Le travail à domicile en Europe

Part des employés qui travaillent habituellement à domicile en 2018 \*

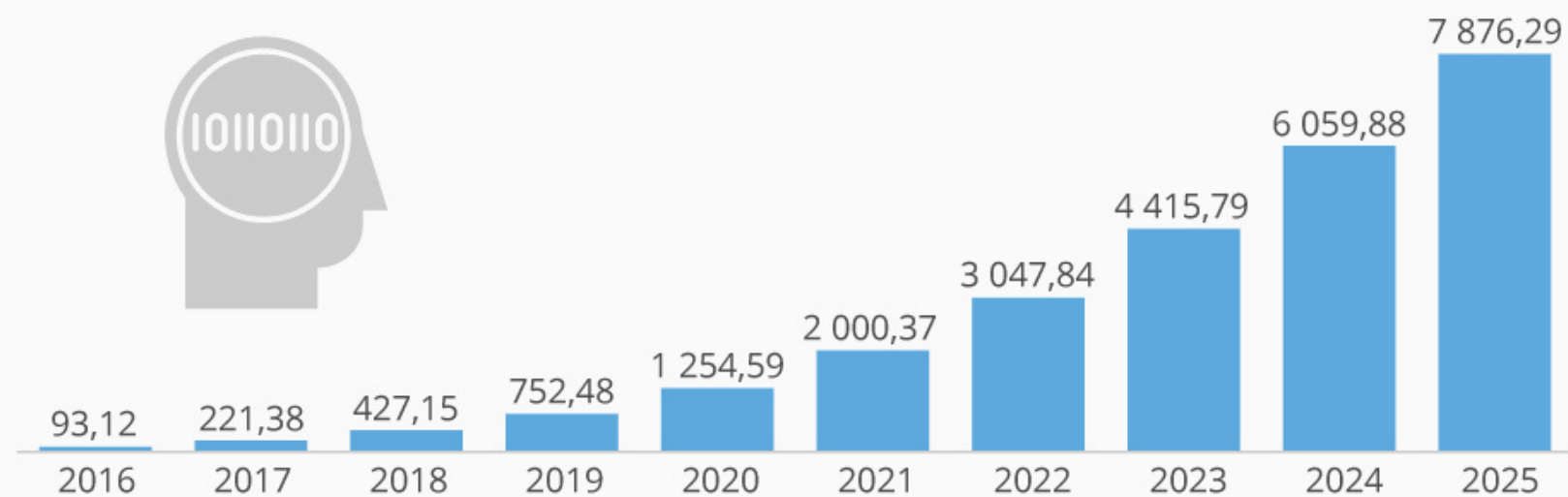


\* dans le total de la population active (15-64 ans).  
Source: Eurostat



### L'intelligence artificielle, un secteur en plein essor en Europe

Prévision du chiffre d'affaires du marché des applications d'IA en Europe (en millions de dollars)



---

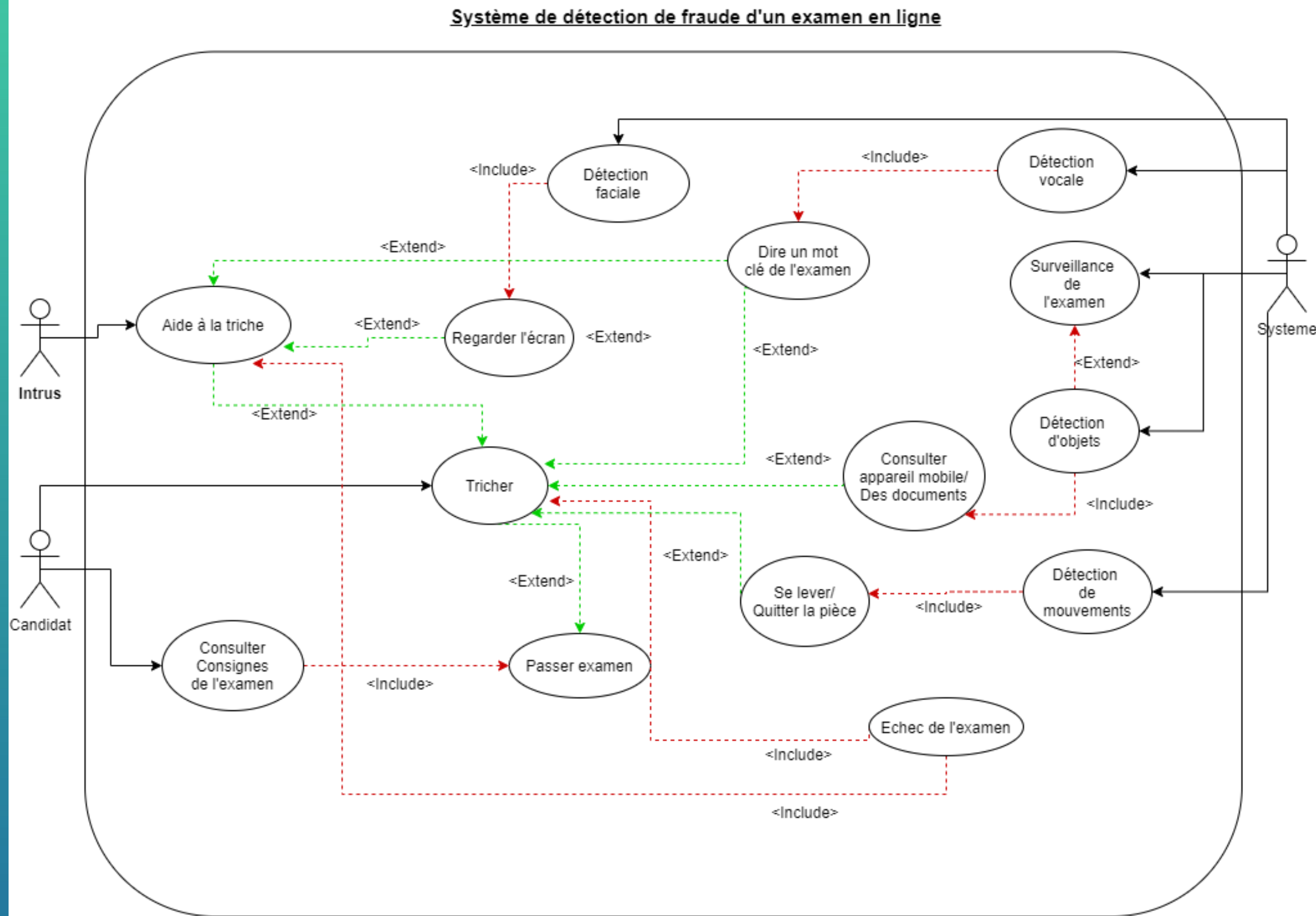
## Les concurrents direct

- Pearson VUE
- ExamShied
- ExamOnline
- eXaminer
- Caveon



# Modèle conceptuel

UseCase de l'utilisation de l'application

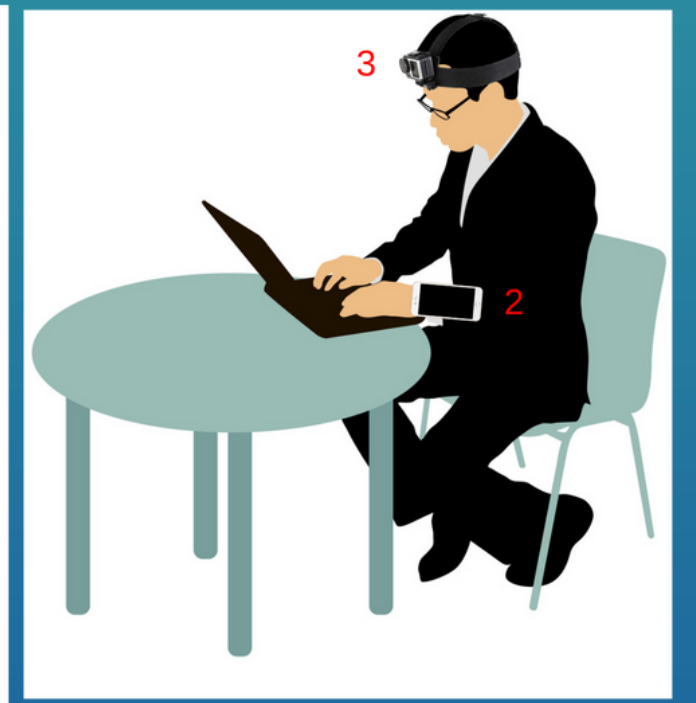


Légende :

- Extend : - - - - ->
- Include : - - - - ->

# Configuration et règles

- 1) Placez vous dans une pièce et ne laissez entrer personnes
- 2) Placez votre téléphone sur votre front
- 3) Personne ne peut entrer dans la pièce pendant la session d'examen. Si cela se produit, l'examen est interrompu.
- 4) Pendant l'épreuve, vous ne pourrez recevoir ni aide ni accompagnement de la part de tiers, et vous ne serez pas autorisé à laisser voir à des tiers l'écran d'ordinateur.
- 5) Vous ne pouvez pas utiliser de ressources
- 6) Vous ne devez pas quitter la zone d'examen
- 7) Votre bureau doit être vide





# Connexion sans fils à l'ordinateur

Le but est de pouvoir connecter le téléphone à l'ordinateur sans le câble.

```
C:\Users\User\AppData\Local\Android\Sdk\platform-tools>adb devices
List of devices attached
ZE222G5SQR      device

C:\Users\User\AppData\Local\Android\Sdk\platform-tools>adb tcpip 5555
restarting in TCP mode port: 5555

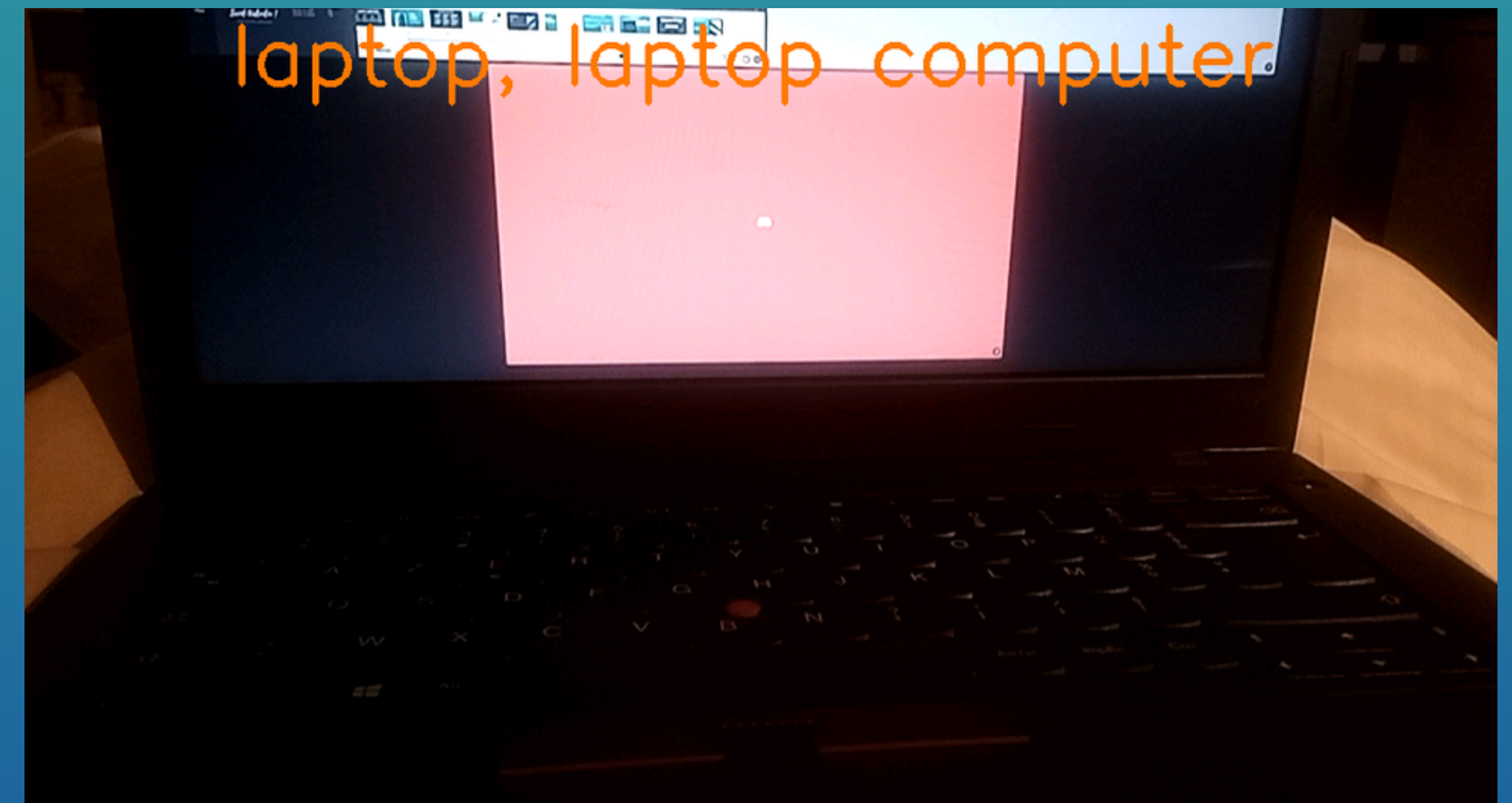
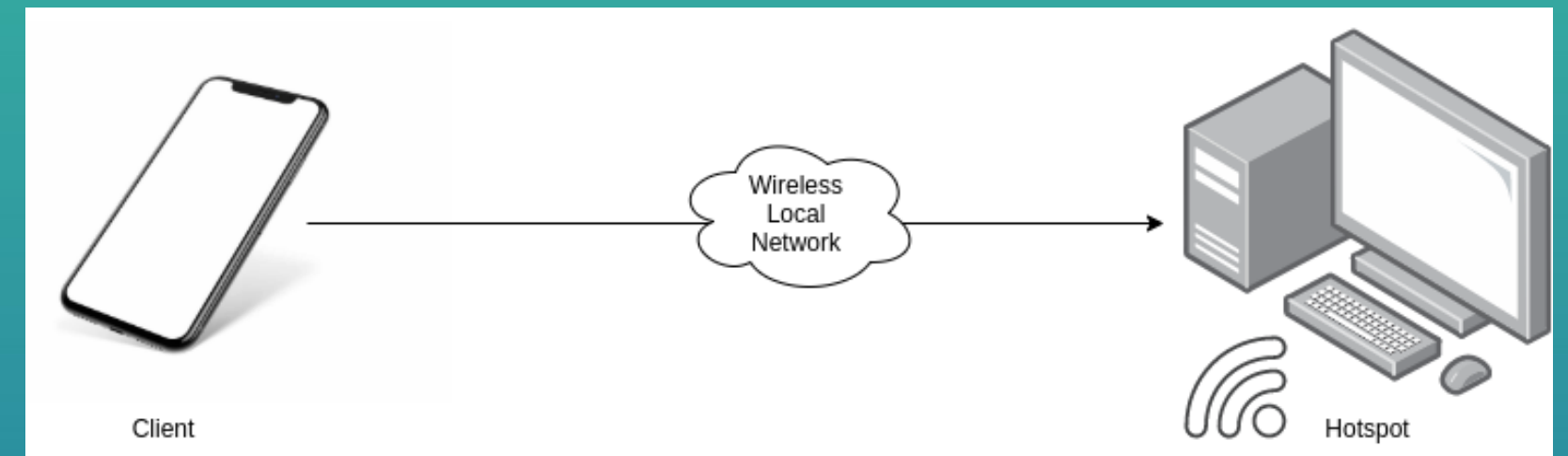
C:\Users\User\AppData\Local\Android\Sdk\platform-tools>adb connect 10.188.175.245
failed to authenticate to 10.188.175.245:5555

C:\Users\User\AppData\Local\Android\Sdk\platform-tools>adb devices
List of devices attached
ZE222G5SQR      device
10.188.175.245:5555  unauthorized

C:\Users\User\AppData\Local\Android\Sdk\platform-tools>adb kill-server

C:\Users\User\AppData\Local\Android\Sdk\platform-tools>adb connect 10.188.175.245
failed to authenticate to 10.188.175.245:5555

C:\Users\User\AppData\Local\Android\Sdk\platform-tools>adb devices
List of devices attached
10.188.175.245:5555  device
```



# Accelerometre

Problème rencontré :

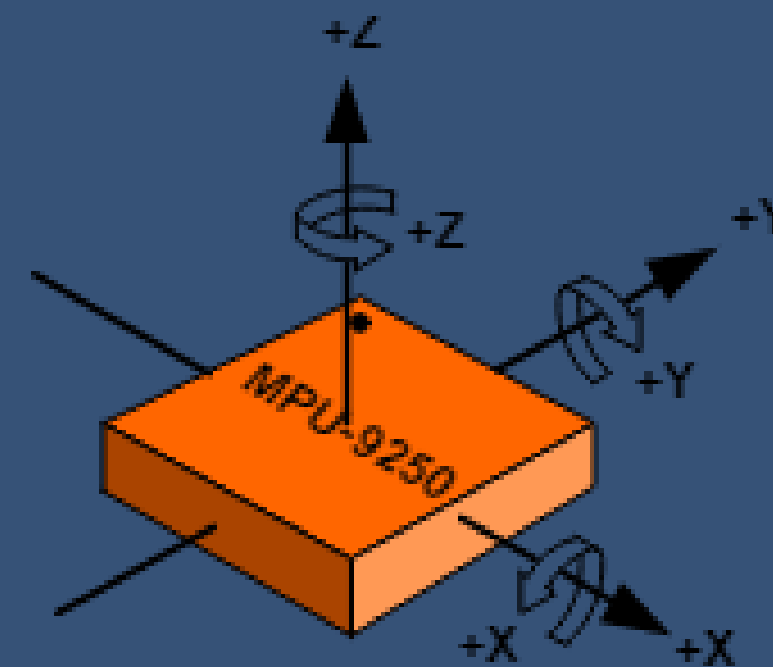
- La force de gravité influence toujours l'accélération

$$A_D = -g - \left(\frac{1}{mass}\right) \sum F_S$$

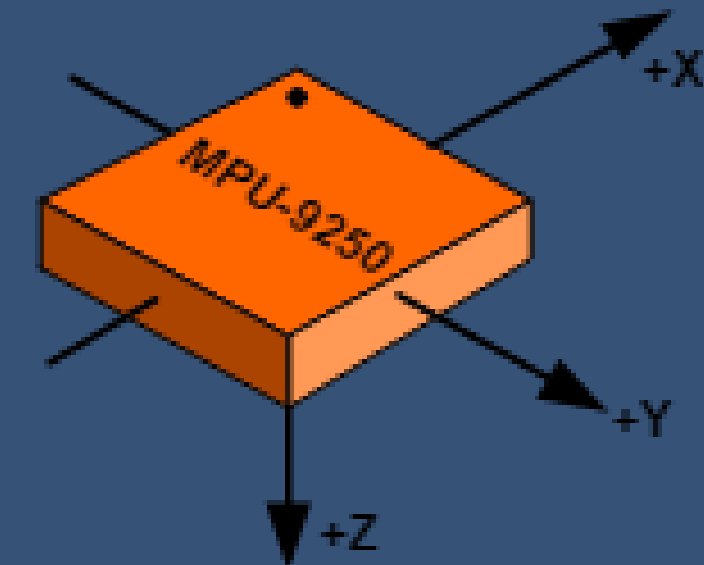
Solution :

- On retire la force de pesanteur

$$\sqrt{x^2 + y^2 + z^2} - g$$



Orientation of Axes of Sensitivity and Polarity of Rotation for Accelerometer and Gyroscope



Orientation of Axes of Sensitivity for Compass

# Envoi des données de l'accelerometre

```
class send extends AsyncTask<Void,Void,Void> {
    Socket s;
    PrintWriter pw;
    PrintWriter pw2;
    @Override
    protected Void doInBackground(Void...params){
        try {
            s = new Socket( host: "10.100.17.96", port: 8000);

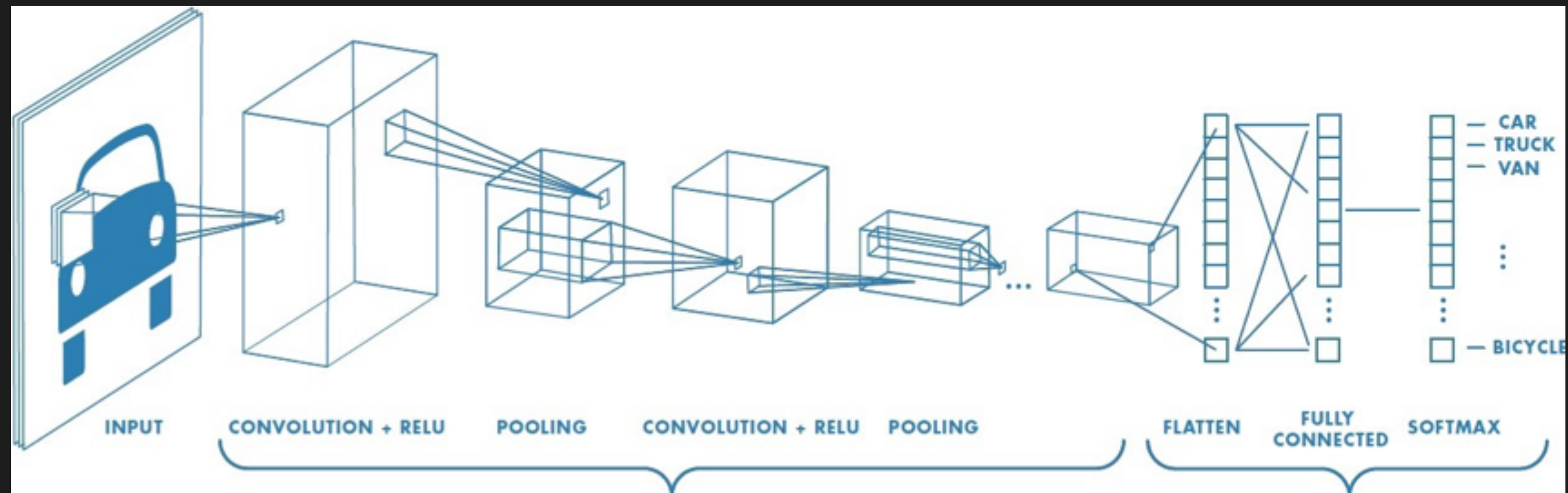
            pw = new PrintWriter(s.getOutputStream(), autoFlush: true);
            //Log.d(TAG, "message1111 : " + message);
            pw.write(message);
            pw.flush();
            pw.close();
            s.close();

        } catch (UnknownHostException e) {
            System.out.println("Fail1");
            e.printStackTrace();
        } catch (IOException e) {
            System.out.println("Fail2");
            e.printStackTrace();
        }
    }
    return null;
}
```

```
#Accepts Incomming Connection
(clientsocket, address) = self.listensocket.accept()
self.message = clientsocket.recv(4096).decode() #Receives Message
```



# Détection d'objets



Image

Feature Learning

Classification

# Envoie des données des labels prédits

---

- Ecriture des labels dans le logcat

```
Log.d(TAG, msg: "getPredictedLabel: "+getPredictedClass(classification, classesPath));
```

- Enregistrement du logcat dans un fichier texte

```
adb logcat -v long time > C:\Users\User\AndroidStudioProjects\logLabel.txt
```

- Lecture du fichier afin de récupérer les labels

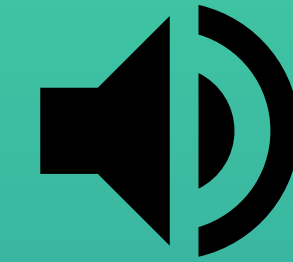
```
#lit le fichier log contenant les labels enregistré par notre application
with open(os.path.join(os.path.dirname(__file__), self.file_name), 'r') as read_label:
    # Read all lines in the file one by one
    for line in read_label:
        # prend le mot apres "getPredictedLabel:" afin de recuperer le label
        if 'getPredictedLabel:' in line:
            self.label = line.split('getPredictedLabel:')[1]
            self.label = self.label.lower().replace(" ", "").strip()
```

# Détection faciale

```
while True:  
    ret, frame = self.cap.read()  
  
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)  
    self.faces = self.face_cascade.detectMultiScale(gray, 1.3, 5)  
  
    for (x, y, w, h) in self.faces:  
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 5)  
  
    cv2.imshow('frame', frame)
```



# Détection vocale



- Tout ce que dit le candidat est enregistré en temps réel
- Lorsqu'un mot clé de l'examen est prononcé, l'examen est arrêté
- Retranscription de la voix en texte
- Historisation dans un fichier texte

```
{'alternative': [{'transcript': 'bye bye', 'confidence': 0.93855584}, {'transcript': 'bye-bye'}, {'transcript': 'salaire bye bye'}, {'transcript': 'itinéraire bye bye'}, {'tra  
{'alternative': [{'transcript': "pourquoi c'est parce qu'il y a trop de stress en cours", 'confidence': 0.90311927}, {'transcript': "pourquoi c'est long parce qu'il y a trop d  
{'alternative': [{'transcript': "ouais on peut pas l'éviter sans rire", 'confidence': 0.9150418}, {'transcript': "horaires on peut pas l'éviter sans rire"}, {'transcript': "on  
[]  
[]  
{'alternative': [{'transcript': 'avant les Fred avant les Fred au tout début ce match de foot sur le toit', 'confidence': 0.89152986}, {'transcript': "avéra les Fred avant les  
[]  
[]  
{'alternative': [{'transcript': "que j'ai bougé vite y a longtemps tu as au moins", 'confidence': 0.82309514}, {'transcript': "que j'ai bougé vu tu y a longtemps tu as au moin
```

# Parallélisation des Threads

```
#lancement du serveur  
serv = Serveur()  
serv.openserver()  
serv.start()
```

```
#lancement de la camera  
camera = Camera()  
camera.start()
```

```
#lancement de la reconnaissance vocale  
voice = Voice_recognizer()  
voice.start()
```



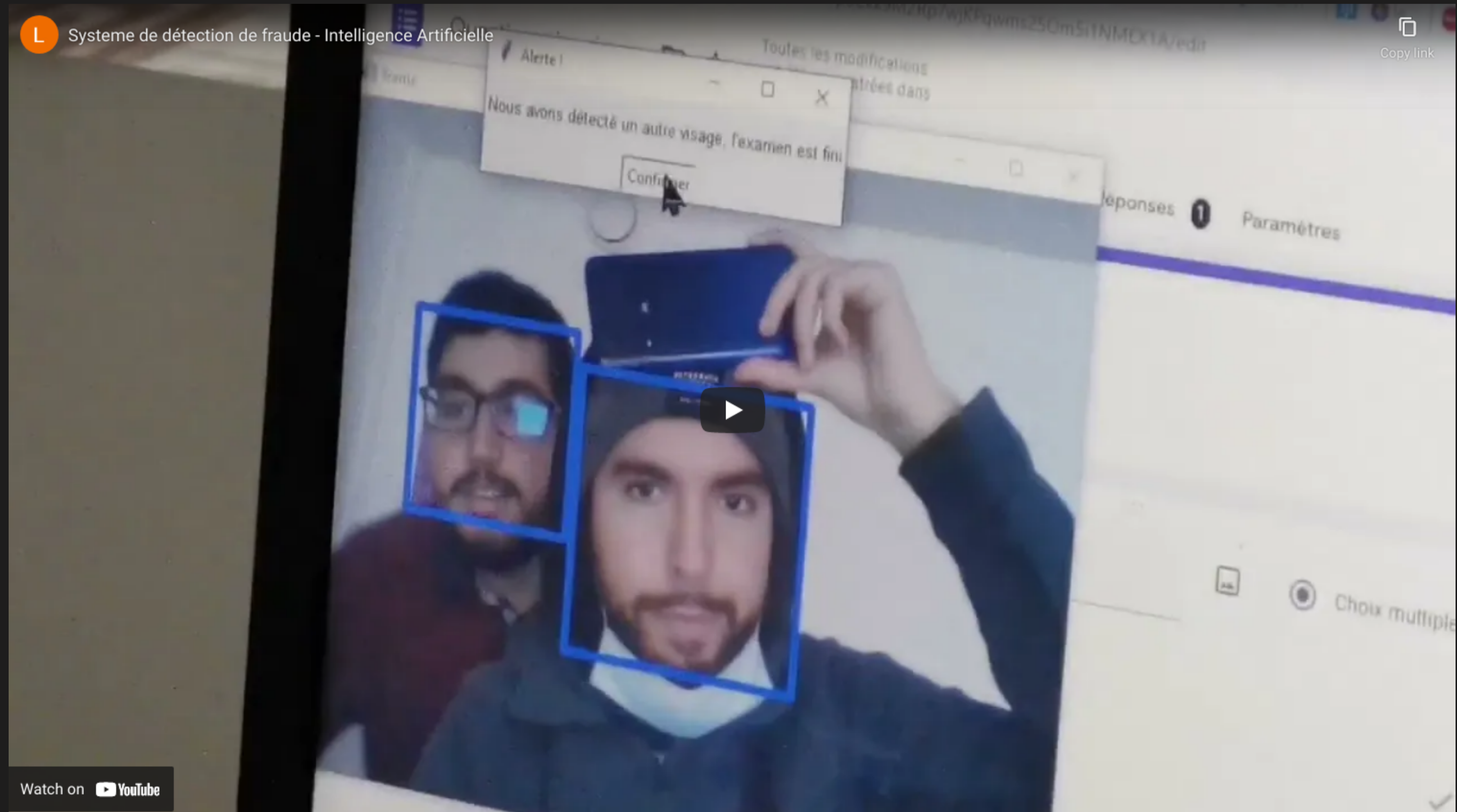
Serveur

Caméra

Voix



# Démonstration de l'application



**Merci pour votre attention**

---

**THANK  
YOU**

