



# Projet innovant RICM4 E-cave

Jonathan Lemoine  
Nicolas Husson

# Plan

## I. Recherche :

- 1) Découverte du matériel et de la technologie
- 2) Format NDEF
- 3) Environnement de développement

## II. Développement :

- 1) Diagramme
- 2) Lire/Écrire tag
- 3) Serveur base de donnée

## III. Application :

- 1) Interface
- 2) Autres fonctionnalités

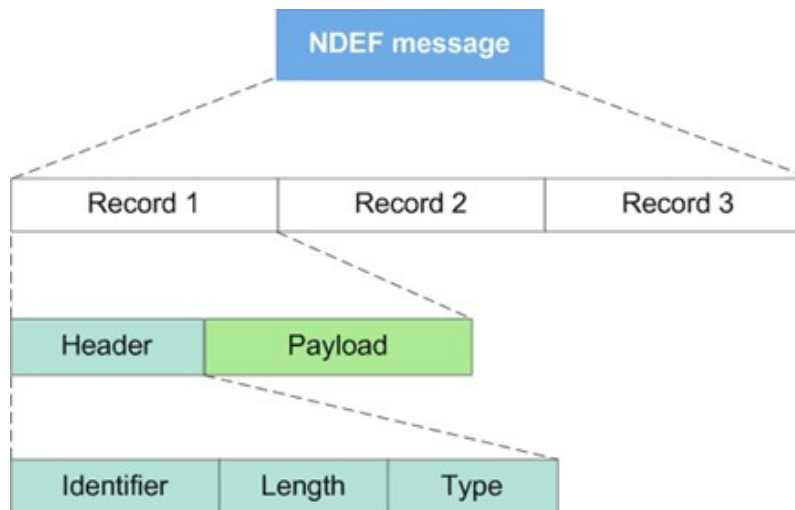
# Découverte du matériel et de la technologie

- NFC : extension de RFID
  - Communication sans-fil
  - Courte portée haute fréquence
- Matériel :
  - Portable android NFC
  - Clé usb / TouchaTag
  - Tablette tactile

# Format NDEF

- NDEF : NFC Data Exchange Format
- Tableau de records :
  - Header + Payload (data)
- Header :
  - Identifier ( type d'enregistrement )
  - Length
  - Type

7	6	5	4	3	2	1	0
MB	ME	CF	1	IL	TNF		
TYPE LENGTH							
PAYLOAD LENGTH							
TYPE							
ID							
PAYLOAD							



# JNI/JNA

- JNI : Utilisation d'un langage natif pour effectuer des appels aux fonctions natives.
- JNA : accéder à du code natif sans faire appel à la couche de programmation JNI
- Utilisation pour la librairie Libnfc
- Révisions des priorités (smartphone)

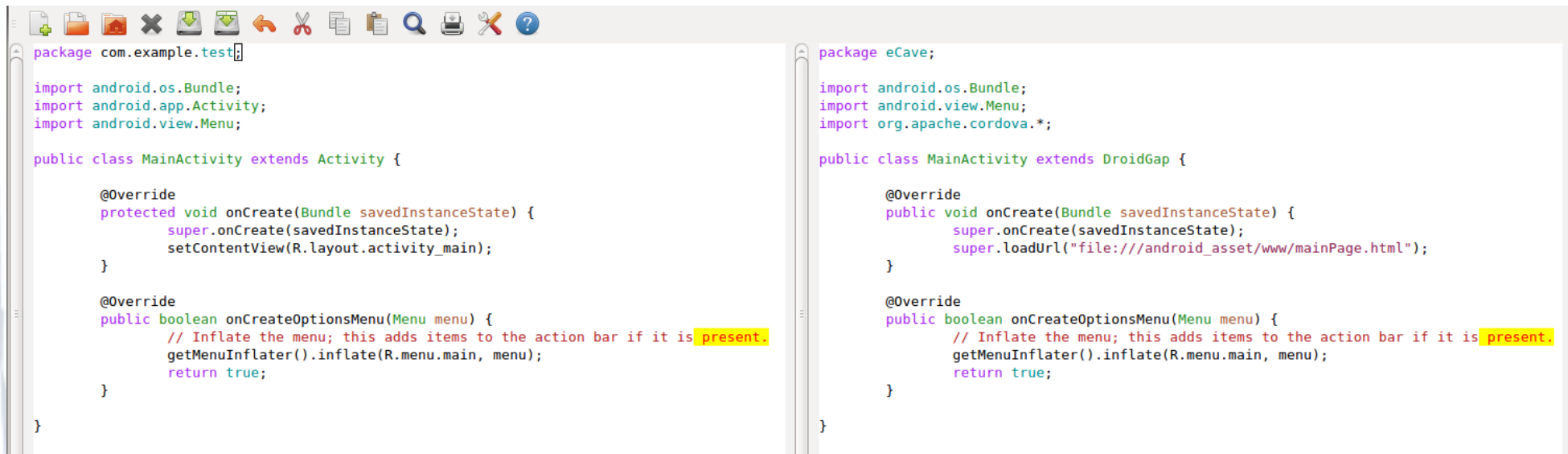
# Androïd VS Phonegap

## Androïd

- Beaucoup de sources d'exemples
- Trop restrictif pour les entreprises

## Phonegap

- Multiplateforme
- Moins de librairies
- Abandon du code natif



The image shows a side-by-side comparison of Java code for an Android application and a Phonegap application. The left editor shows standard Android code using `Activity` and `setContentView`. The right editor shows Phonegap code using `DroidGap` and `super.loadUrl` to load a web page. Both code snippets include imports for `Bundle`, `Activity`, and `Menu`, and implement `onCreate` and `onCreateOptionsMenu` methods. The word "present." is highlighted in yellow in both code blocks.

```
package com.example.test;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

```
package eCave;

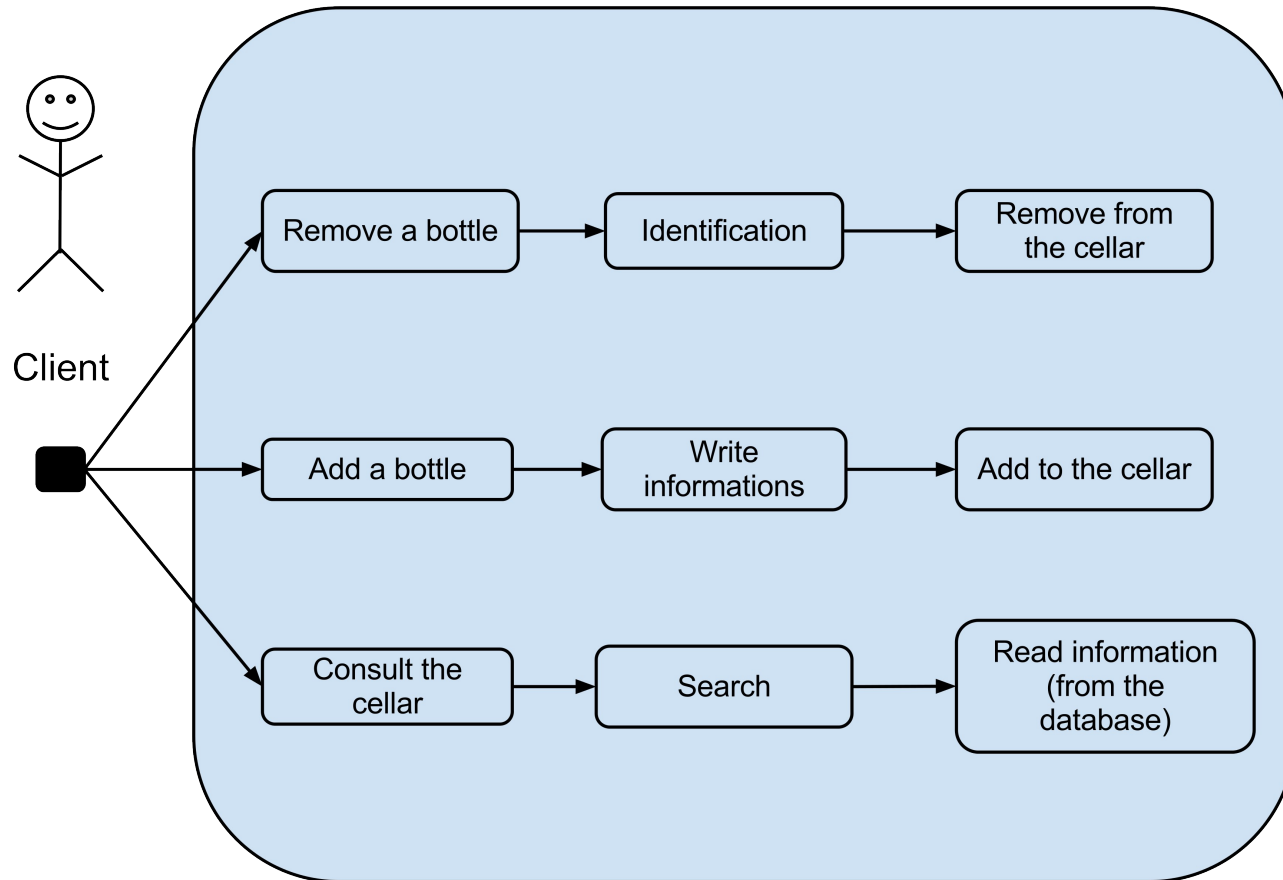
import android.os.Bundle;
import android.view.Menu;
import org.apache.cordova.*;

public class MainActivity extends DroidGap {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        super.loadUrl("file:///android_asset/www/mainPage.html");
    }

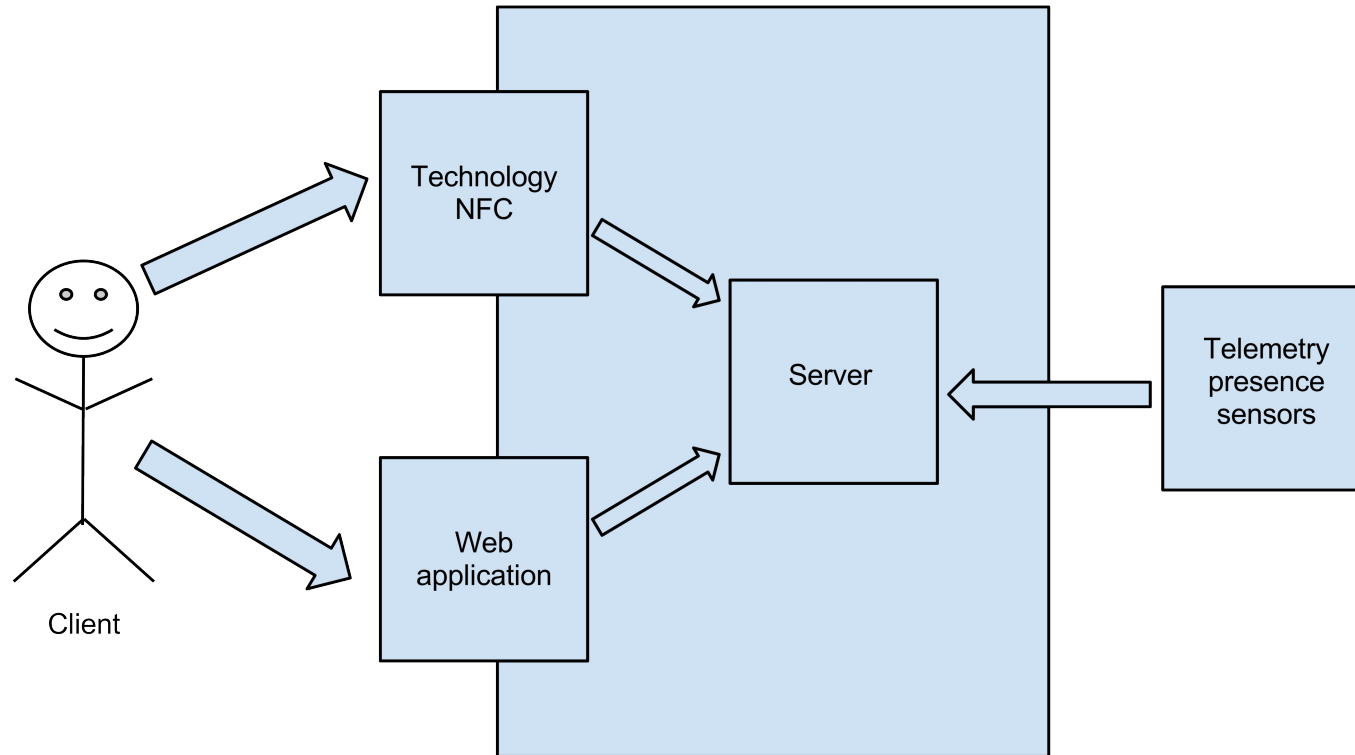
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

# Usecase



Usecase Diagram

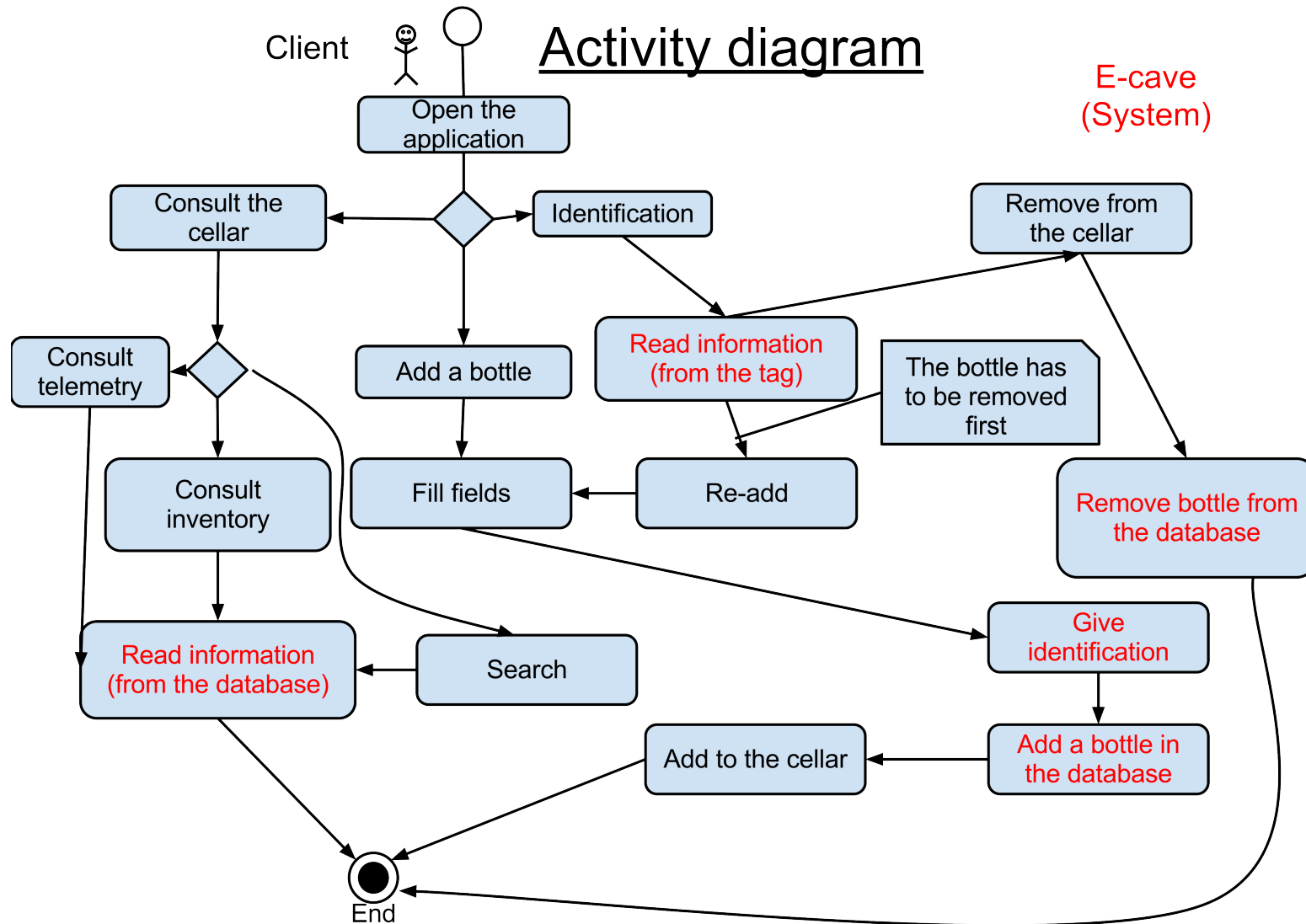
# Context



Context diagram

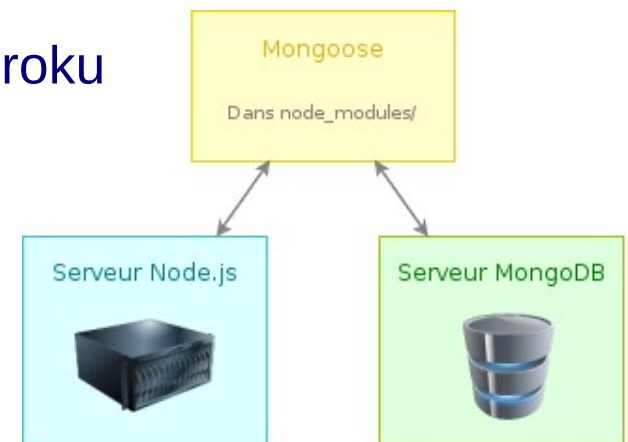
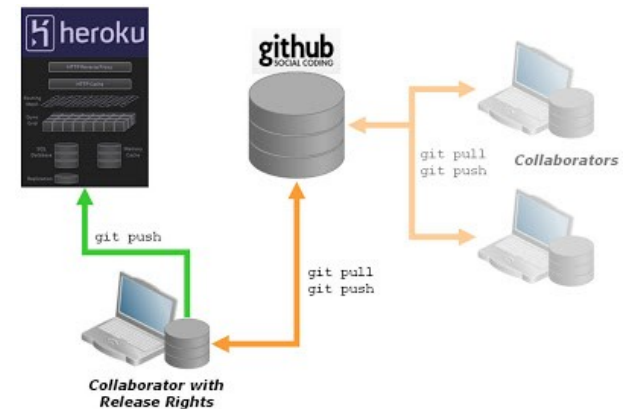


# Activity



# Heroku/Nodejs/MongoDB

- Plateforme Heroku
- Serveur en NodeJs ( uniformité du langage utilisé )
- MongoDB base de donnée en format JSON
  - Format BSON
- Mongolab ( mongoDB-as-a-service )
- Mongoose automatiquement créé et géré par heroku
- Création de schéma mongoose pour créer BD



# Lire/Écrire tag

- Mise en place de NdefListener
- Lire :
  - Récupère payload du record
  - ByteToString
- Écrire :
  - Ndef record de notre texte
  - Écriture sur le tag dès qu'il est présent



# Format JSON

- Taille faible, utilisé pour les applications mobiles (à l'instar de XML)
- Type date, string, number...
- <http://jsonlint.com/>

```
{  
    "typeDeVin" : "Rouge",  
    "annee" : 1990,  
    "domaine" : bordeaux,  
    "dateInput": "2013/04/08",  
    "dateOutput": "2013/04/09",  
    "stocked": 15  
}
```



# Format JSON (coté mongoLab)

```
{  
  "_id": {  
    "$oid": "516402a6e4b02c44a8f02e71"  
  },  
  "typeDeVin": "rouge",  
  "annee": "1990",  
  "domaine": "savoie",  
  "dateInput": "2/3/2013",  
  "dateOutput": "2013/04/09",  
  "stocked": 15  
}
```

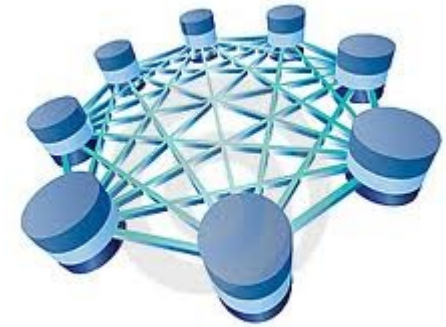


# Pré remplissage des champs

- Abandon du code bar
- Depuis NFC
- Depuis la base de Donnée
- Passage de paramètres par l'URL

The screenshot shows an Android application window titled "5554:AndroidPhone". The app's title bar is "Add a bottle". The main content area asks "Do you want to fill the fields ?" and provides two buttons: "nfc" and "Database". Below these is a button labeled "Fill the fields". The form contains four input fields: "Wine type:", "Year:", "Domain:", and "Quantity:". The right side of the screen displays a vertical toolbar with various icons for basic controls, hardware buttons, and a hardware keyboard. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps buttons.

# Consultation de la base de donnée



- Accède via MongoLab's HTTP-based REST API
- Operation SQL HTTP
- Create INSERT POST (en)
- Read (Retrieve) SELECT GET (en)
- Update (Modify) UPDATE (en) PUT (en) / PATCH (en)
- Delete (Destroy) DELETE (en) DELETE (en)
- Exemples de requêtes ajax utilisées :
- Accès à la base de donnée :  
<https://api.mongolab.com/api/1/databases/my-db/collections/my-coll?apiKey=myAPIKey>
- Recherche les entités où typeDeVin vaut « rouge »  
[https://api.mongolab.com/api/1/databases/my-db/collections/my-coll?q={\"typeDeVin\": \"rouge\" }&apiKey=myAPIKey](https://api.mongolab.com/api/1/databases/my-db/collections/my-coll?q={\)



# Suppression d'une bouteille

- Uniquement depuis NFC
- Plusieurs TAG pour un ID
- Présent dans la DB et pas présent dans la cave => stocked 0



# Température/Hygrométrie

- Relevé toutes les 3h
- Requête au serveur pour obtenir la dernière valeur
- Format JSON depuis MongoLab :

```
{  
  "_id": {  
    "$oid": "51618463e4b08b993350c365"  
  },  
  "date": "07/04/2013 12",  
  "humidity": 30  
}
```



- Implémentation possible : notification en cas de problème / graphe

# Interface

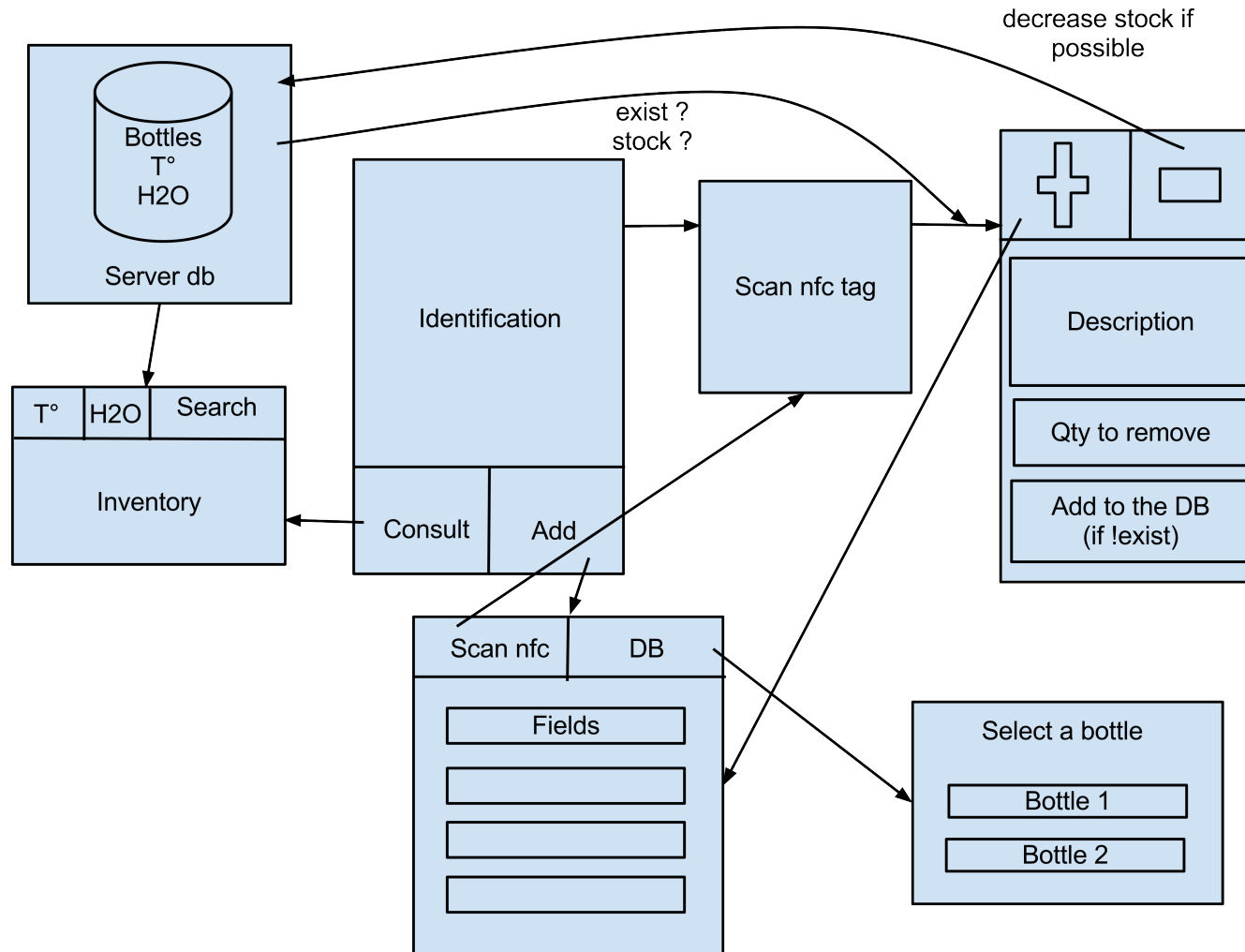


Diagram interface

# Conclusion

- Cahier des charges, travail préliminaire
- Nouvelles technologies, librairies, langages
- Améliorations possibles
- Apport personnel