

HE  
IG

VD

IICT

Institut des  
Technologies de  
l'information et de  
la communication



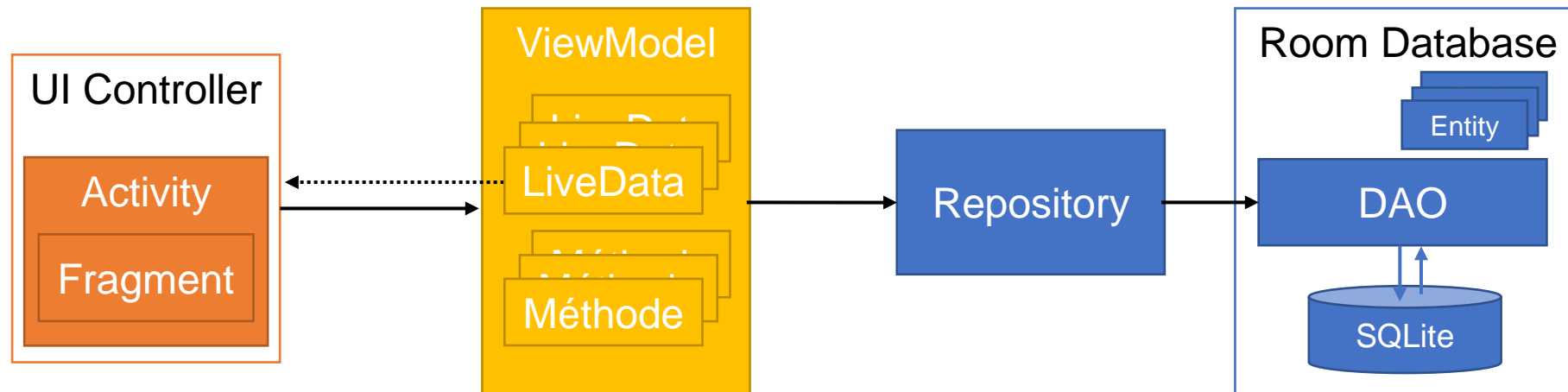
# Développement d'applications *Android*

## Laboratoire 6

Fabien Dutoit

# Introduction

- Réalisation d'une app *MVVM* avec une base de données locale synchronisée avec un serveur REST
- Le squelette de l'app est fourni (≡ état final du laboratoire sur *MVVM*) :
  - Activité
    - Fragment listant les contacts en DB locale
  - Viewmodel, Repository, Dao, Database, Entité
  - Génération de données par défaut (à enlever)



# Le serveur

- Accessible sur Internet : <https://daa.iict.ch/>
- Avec le navigateur on obtient une interface permettant de suivre les requêtes traitées par le serveur

## DAA - Server log

Show 10 entries

Search: 3224f82b-6c28-4f69-aa6f

Recherche par  
UUID conseillée

Date	UUID	Client address	Method	Endpoint	Response Status
2022-11-28T08:29:07.584+00:00	3224f82b-6c28-4f69-aa6f-469286473981	10.192.124.103	GET	/contacts	200 - OK
2022-11-28T08:29:06.352+00:00	3224f82b-6c28-4f69-aa6f-469286473981	10.192.124.103	PUT	/contacts/9086	200 - OK
2022-11-28T08:28:59.589+00:00	3224f82b-6c28-4f69-aa6f-469286473981	10.192.124.103	DELETE	/contacts/9087	204 - NO_CONTENT
2022-11-28T08:28:53.570+00:00	3224f82b-6c28-4f69-aa6f-469286473981	10.192.124.103	POST	/contacts	201 - CREATED
2022-11-28T08:28:27.352+00:00	3224f82b-6c28-4f69-aa6f-469286473981	10.192.124.103	GET	/contacts	200 - OK

Date UUID Client address Method Endpoint Response Status

Showing 1 to 5 of 5 entries (filtered from 6,283 total entries)

Previous 1 Next

# Le serveur – API

- Le serveur met à disposition une API permettant d'interagir avec sa base de données de contacts. Il va gérer une collection de contacts par utilisateur, représentée par un UUID
- Toutes les requêtes vers */contacts* devront être accompagnée de l'UUID (header)
- Les endpoints :
  - GET <https://daa.iict.ch/enroll> Création d'un nouvel utilisateur (UUID) et d'un ensemble de contacts par défaut qui lui sont associés
  - GET <https://daa.iict.ch/contacts> Récupérer tous les contacts associés à un UUID
  - GET <https://daa.iict.ch/contacts/id> Obtention d'un contact particulier par id
  - POST <https://daa.iict.ch/contacts> Création d'un nouveau contact (payload json)
  - PUT <https://daa.iict.ch/contacts/id> Modification d'un contact existant (payload json)
  - DELETE <https://daa.iict.ch/contacts/id> Suppression d'un contact particulier par id

# Manipulations

- Ajout sur l'app des éléments d'interface graphique permettant de créer, modifier et supprimer un contact (dans un Fragment)
- Une possibilité d'implémentation :

## Modification / Suppression

8:45 3G

← REST Lab

Edit contact

Name Pelletier

Firstname Bernard

E-Mail b.pelletier@gmel.com

Birthday 26.12.2003

Address Avenue des Sports 20

Zip 1400

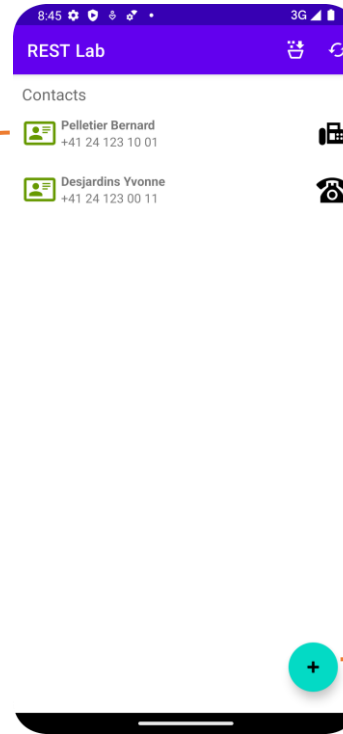
City Yverdon-les-Bains

Phone type

☐ Home ☐ Mobile ☐ Office ☒ Fax

Phone number +41 24 123 10 01

CANCEL DELETE SAVE



## Création

8:46 3G

← REST Lab

New contact

Name

Firstname

E-Mail

Birthday

Address

Zip

City

Phone type

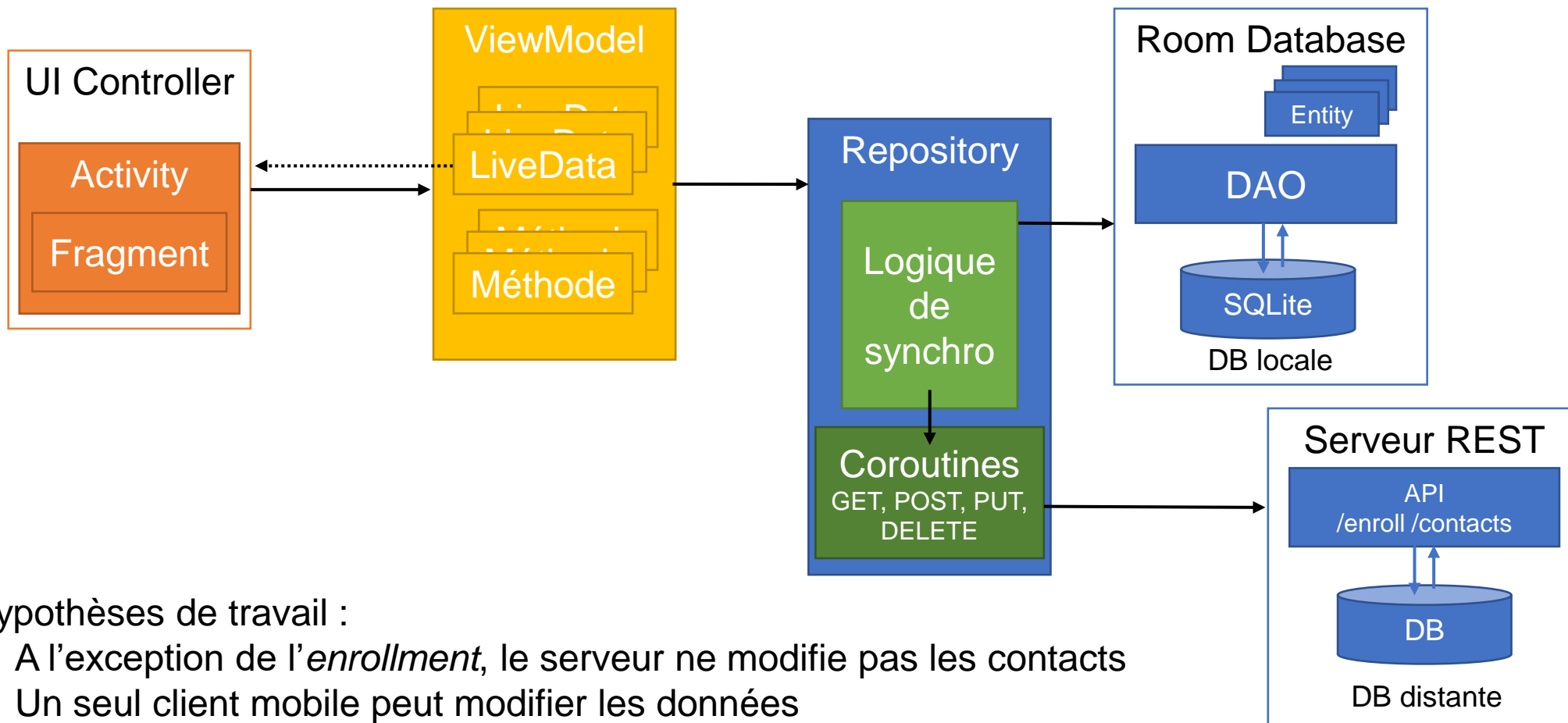
☐ Home ☐ Mobile ☐ Office ☐ Fax

Phone number

CANCEL CREATE

# Manipulations

- Intégration des opérations CRUD et de la logique de synchronisation dans le *Repository* :

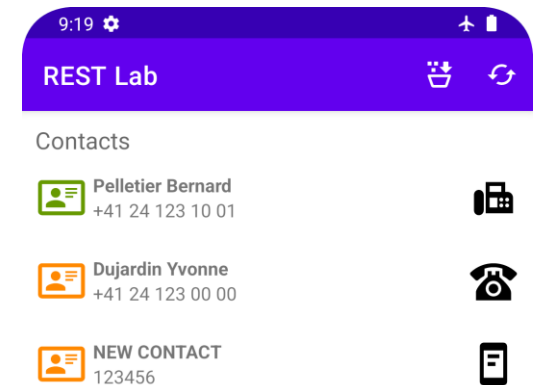


Hypothèses de travail :

- A l'exception de l'*enrollment*, le serveur ne modifie pas les contacts
- Un seul client mobile peut modifier les données

# Hints / Conseils

- N'hésitez pas à mettre votre smartphone / émulateur en mode avion pour simuler l'inaccessibilité du serveur
- Vous pouvez (devez) modifier le schéma de la DB locale pour permettre la mise en œuvre de l'algorithme de synchronisation
- Vous pouvez changer la couleur des entrées «dirty» pour vous aider à mettre en place la synchronisation
- Un seul *Fragment* peut s'occuper de la création et de l'édition d'un contact
- Pour simplifier le passage en mode édition, vous pouvez utiliser une *LiveData* «nullable» contenant le contact nouvellement créé ou à éditer. L'affichage du *Fragment* d'édition sera géré en observant cette *LiveData*
- N'hésitez pas à utiliser *Postman* pour tester vos requêtes sur le serveur





## Alternative avec *JetPack Compose*

- Le laboratoire peut également être réalisé en utilisant *Jetpack Compose* pour la réalisation de l'interface graphique
- Vous devez réaliser **une seule version** du laboratoire, **le choix est libre**
- Une autre version du squelette de l'app de base est fournie
- Dans cette version nous n'allons pas utiliser de *Fragment*, nous aurons deux fonctions composables représentant des écrans, en fonction de l'état, l'une (liste des contacts) ou l'autre (édition d'un contact) sera affichée

```
if(editionMode) {  
    ScreenContactEditor(contact = selectedContact, [...])  
}  
else {  
    ScreenContactList(contacts) { selectedContact ->  
        [...]  
    }  
}
```

Etat (mode liste ou mode edition)

A réaliser, équivalent au *Fragment* défini dans la manipulation 2

Fourni

HEIG  
VD  
HAUTE ÉCOLE  
D'INGÉNIEURIE  
ET DE GESTION  
DU CANTON  
DE VAUD

HE<sup>VD</sup>  
IG

HAUTE ÉCOLE  
D'INGÉNIEURIE  
ET DE GESTION  
DU CANTON  
DE VAUD