FMT Reversi Android

by

Francesco Benincasa

Roadmap

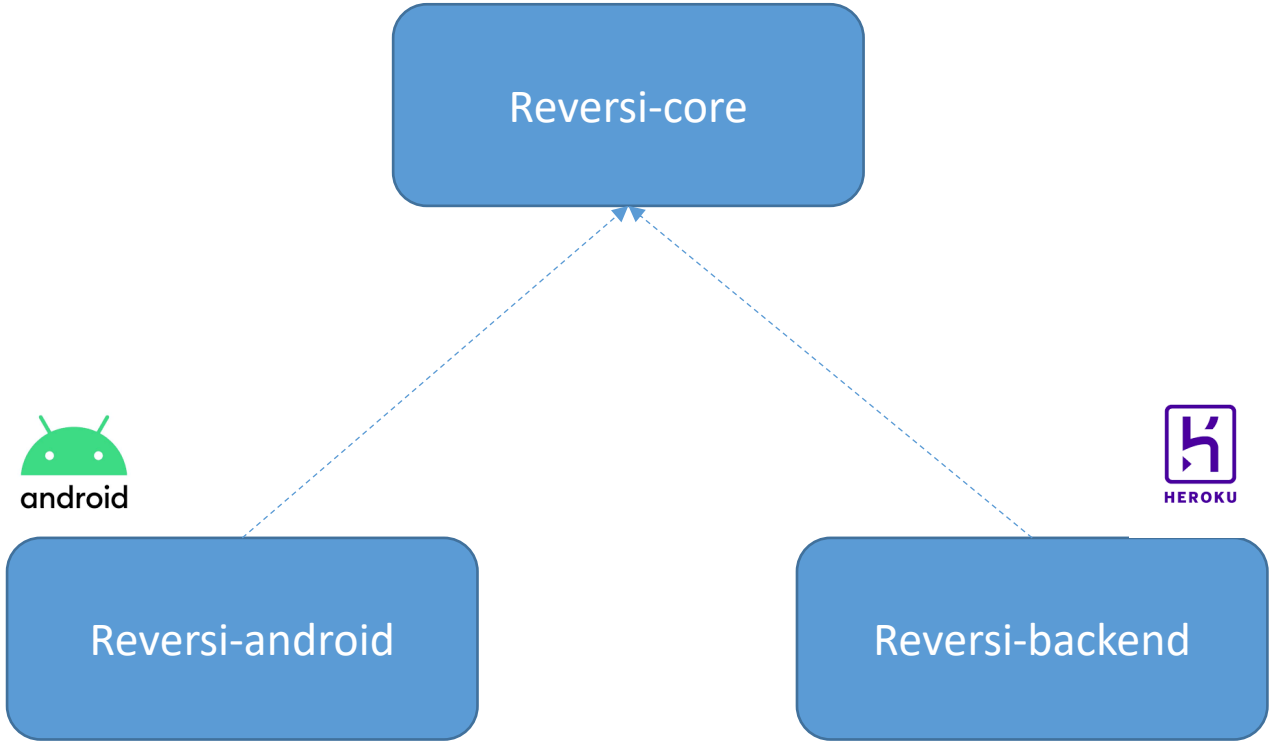
- Intro
- The project
- Demo
- Let’s dive into the code
- Conclusions

Introduction

- The aim of the project is to realize a Reversi implementation on Android platform:
  - Match on the same device
  - Match between two devices (each player on his own device)
- The game rules are based on Reversi on Wikipedia and Federazione Nazionale Gioco Othello

The project - what did we use?

The project - modules



Build process

Android development

Backend development

GitHub

ngrok

HEROKU

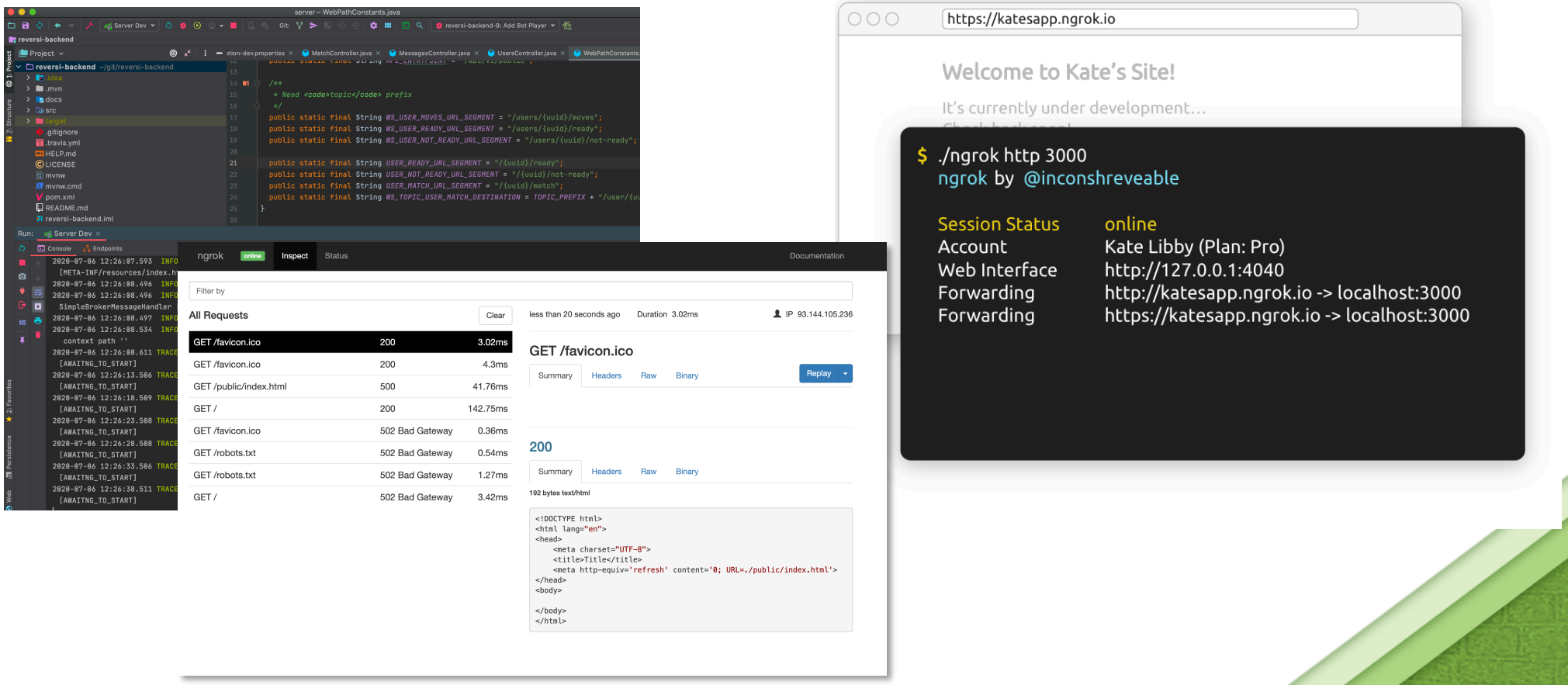
android

https://836c360b507d.ngrok.io

https://fmt-reversi.herokuapp.com/

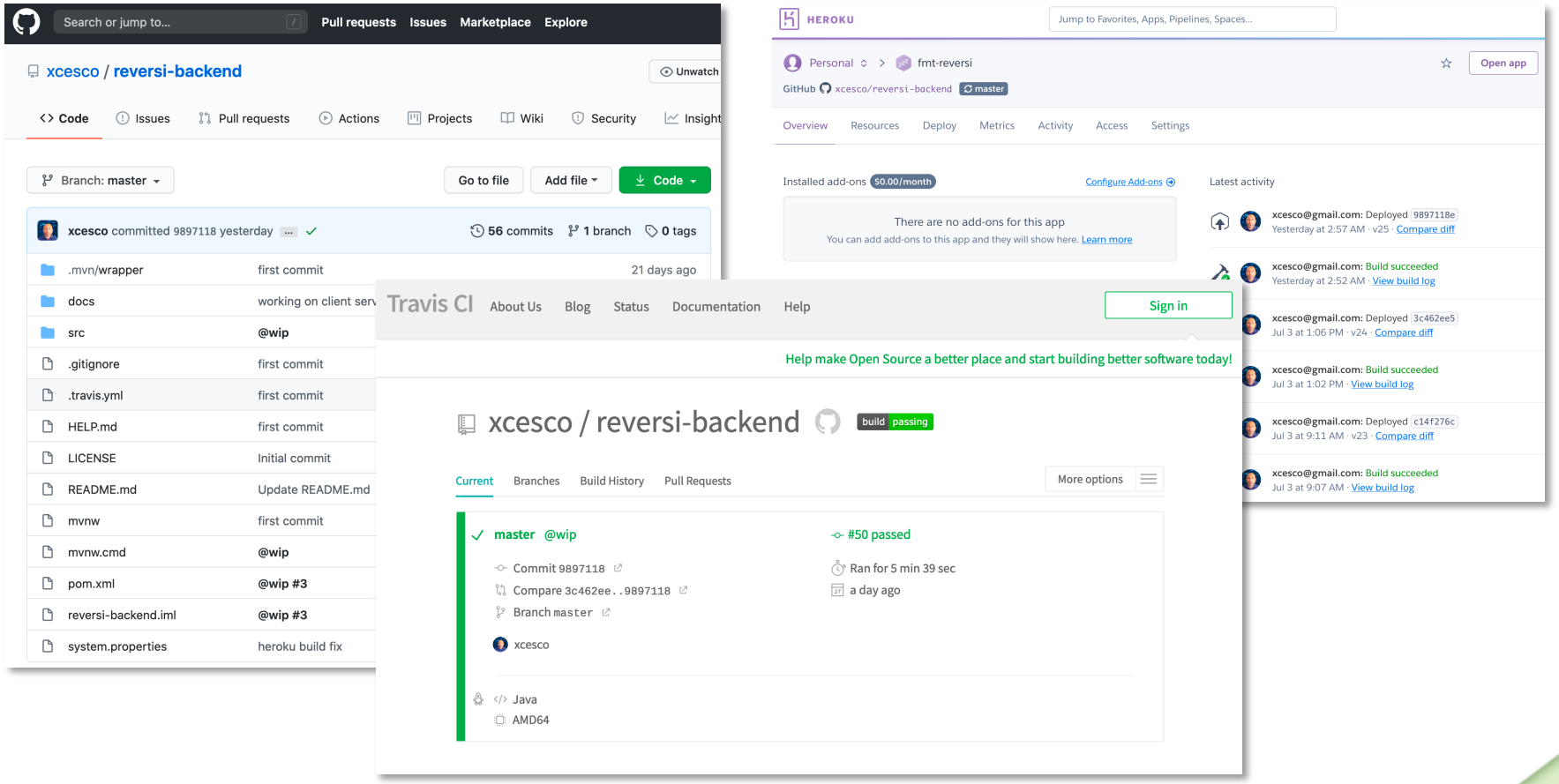
Backend development

NGROK was used to expose local server on the web during development



Backend development

To deploy backend server, github, travis and heroku were used



Android development

- View model & live data
- Shared preferences
- Navigation
- Intent
- Retrofit
- Websocket & STOMP
- Recycler View
- Jackson
- RX java
- Timber
- Firebase
- Dagger2

About Android development

- Retrofit: https://square.github.io/retrofit/
- Okhttp: https://square.github.io/okhttp/
- Websocket & STOMP protocol: https://github.com/NaikSoftware/StompProtocolAndroid
- Jackson: https://github.com/FasterXML/jackson
- ViewBinding: https://developer.android.com/topic/libraries/view-binding#java
- Daggger: https://github.com/google/dagger
- Navigation component: https://developer.android.com/guide/navigation/navigation-pass-data
- ModelView & Live data components: https://developer.android.com/topic/libraries/architecture/livedata
- Timber: https://github.com/JakeWharton/timber
- Firebase https://firebase.google.com/docs/crashlytics/get-started?authuser=0&platform=android

Why WebSocket?

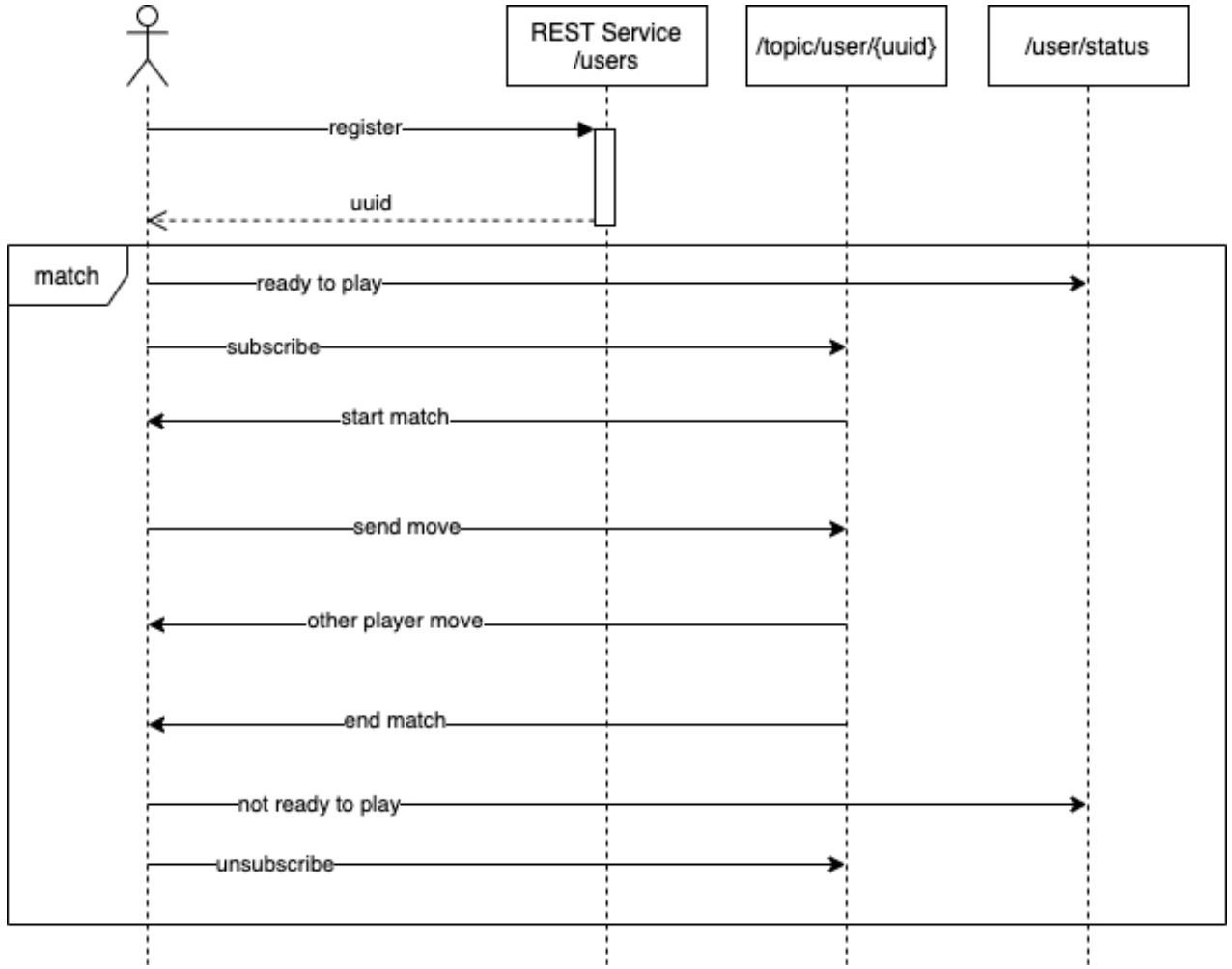
- Game server contains match status
- Network matches required in some case server start comunication with clients.
- The choice was Simple (or Streaming) Text Oriented Message Protocol (STOMP) on WebSocket

```
D/AbstractConnectionProvider: Receive STOMP message: MESSAGE
    type:MATCH_START
    destination:/topic/user/c0056fbe-9cbc-44f7-9c82-44930b477f52
    content-type:application/json
    subscription:066fcd93-cfa5-49c8-b2f9-690a8b6c4b1c
    message-id:246aa132-e4b5-d2fb-b1fc-56a7111962a9-1
    content-length:257

    {"player1Type":"NETWORK_PLAYER","player1Name":"BOT","player2Type":"HUMAN_P
```

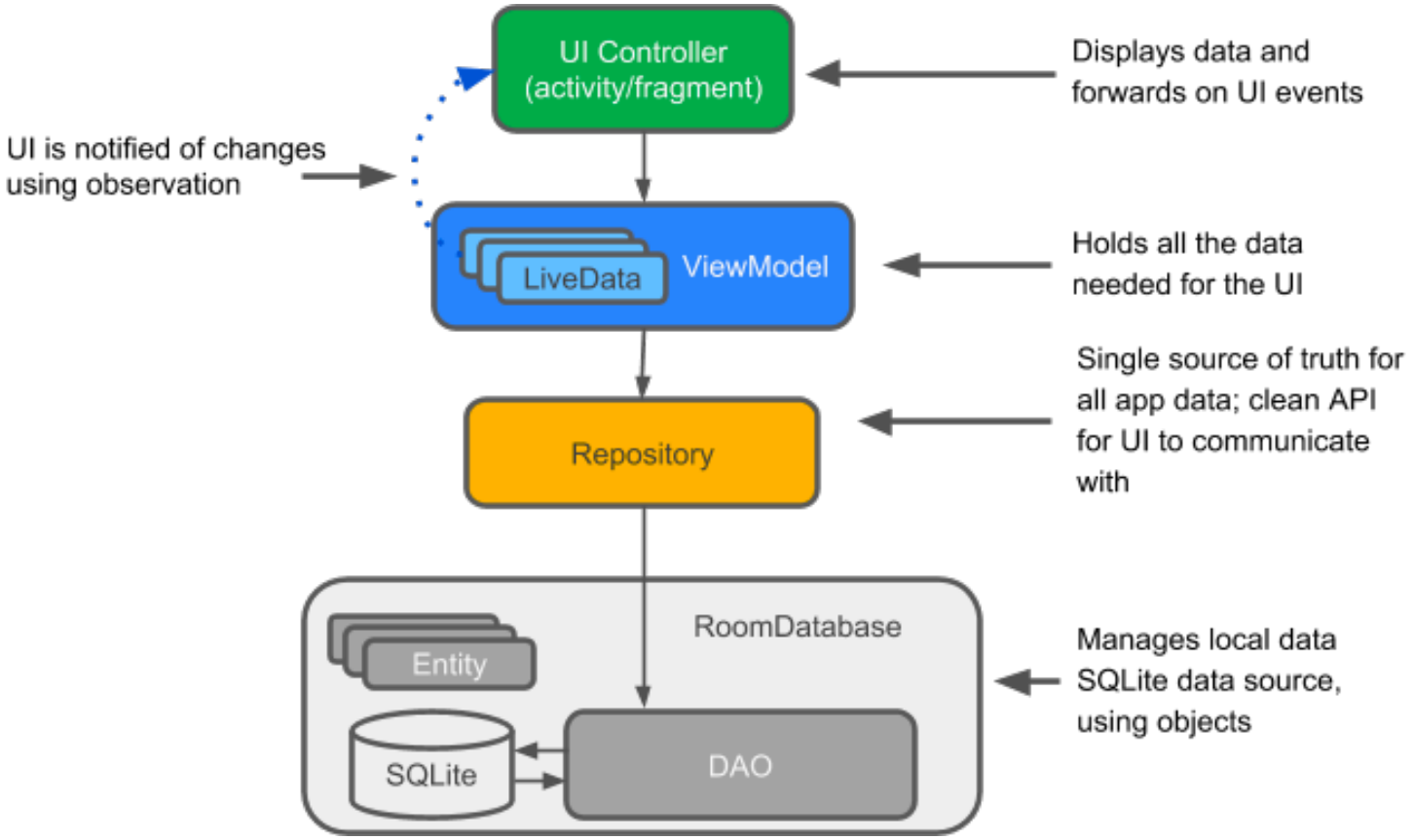
HTTP	WebSocket
Duplex	
Half	Full
Messaging Pattern	
Request-reponse	Bi-directional
Service Push	
Not natively supported. Client polling or streaming download techniques used.	Core feature
Overhead	
Moderate overhead per request/connection.	Moderate overhead to establish & maintain the connection, then minimal overhead per message.
Intermediary/Edge Caching	
Core feature	Not possible
Supported Clients	
Broad support	Modern languages & clients

Why WebSocket?



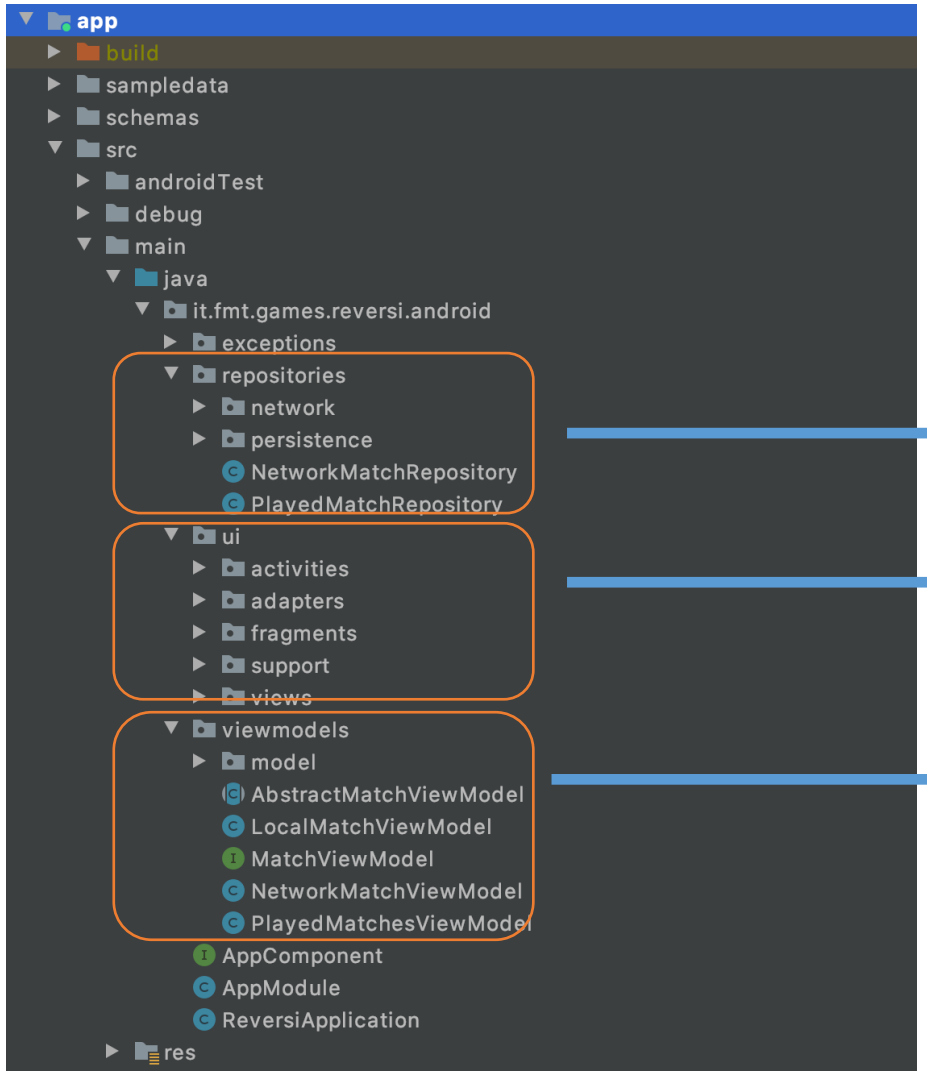
- Client connects to server with a web service (synchronized request-response)
- Client receives an UUID will be used from server to identificate it.
- Client subscribe to match topic to receive server comunication.
- JSON message, based on STOMP over WebSocket.
- Clients send data to server and receive async message from server, until match finish.

Why Live Data and View Model?



- Live data and view models allow to manage model changes on UI respecting UI components lifecycle

Source code organization

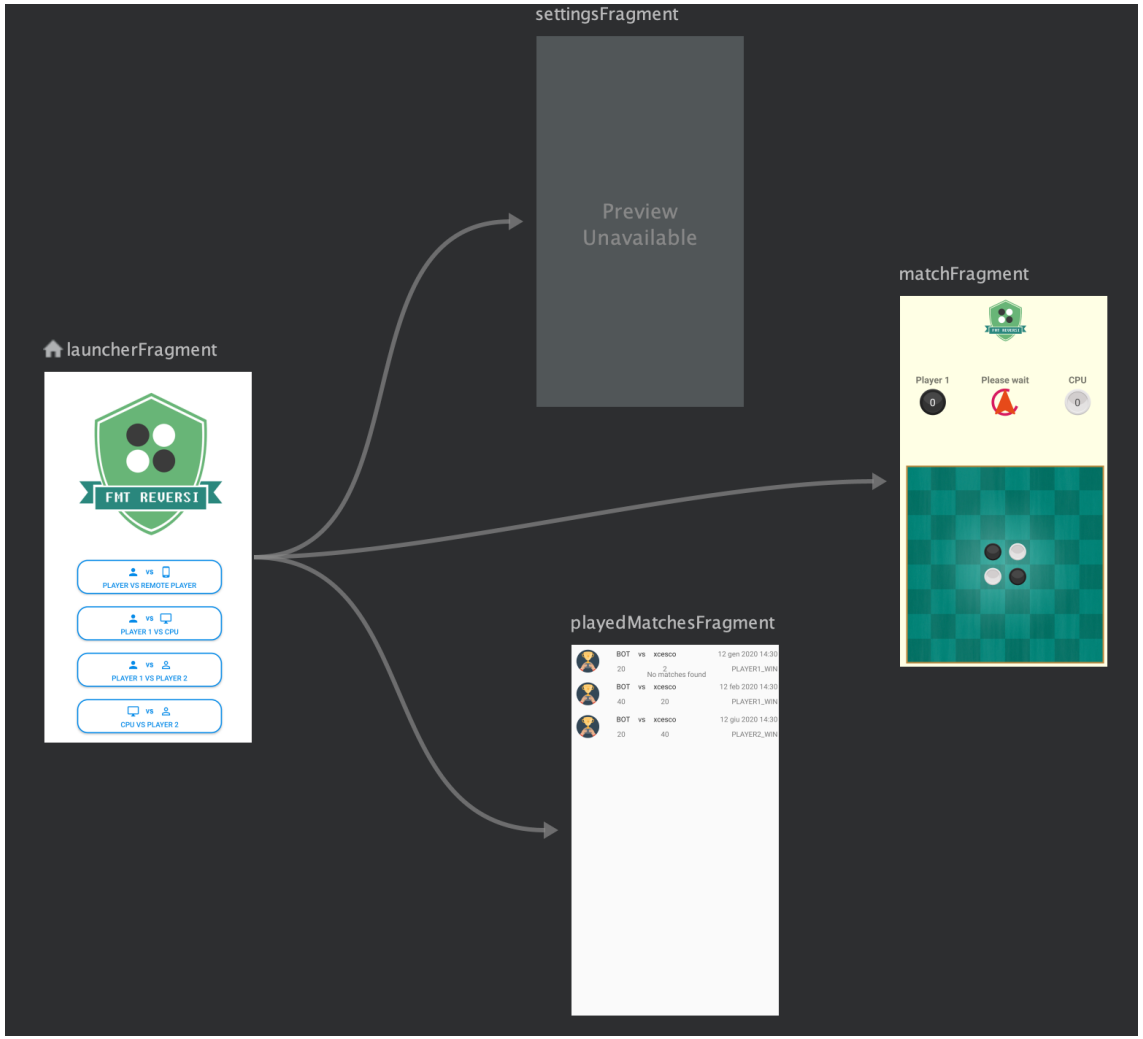


Repositories package

UI (activities, fragment etc)

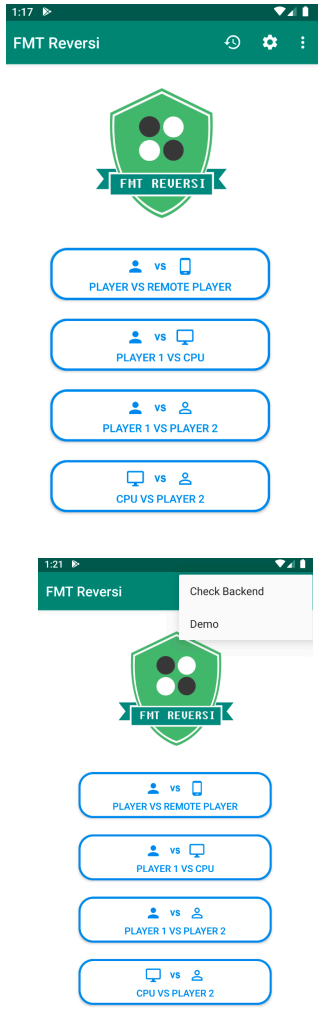
View models

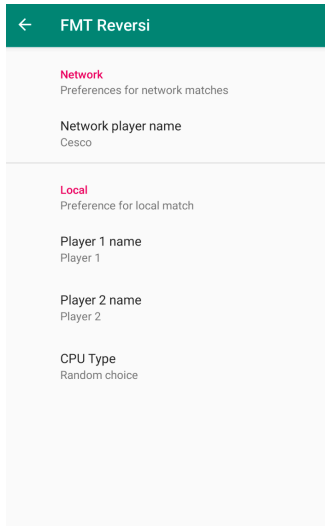
Navigation between fragments

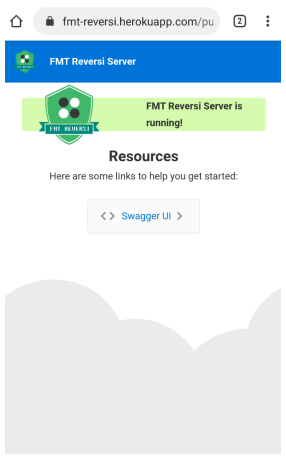


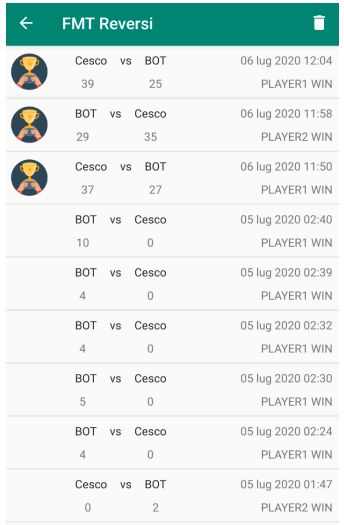
1 activity, 4 fragments, 1 navigation graph

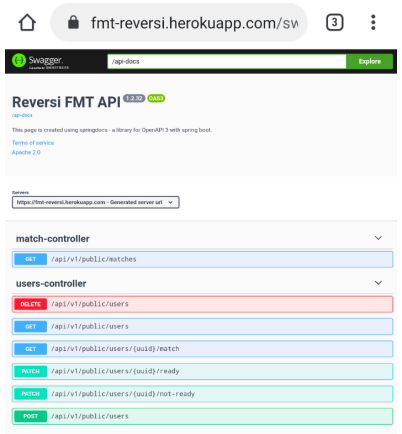
Navigation between fragments

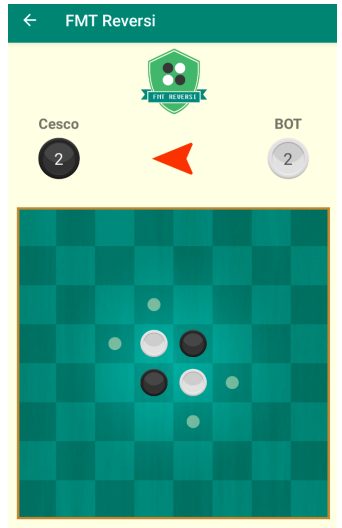




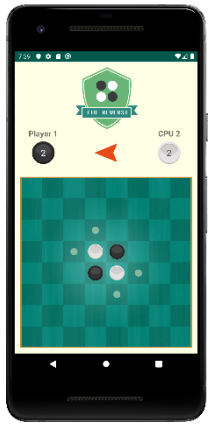








Demo



It’s time to play!

Android version available on Google Play Store

https://play.google.com/store/apps/details?id=it.fmt.games.reversi.android

SCAN ME

Network match demo on https://youtu.be/RUfBwd1IXWg

Source code on GitHub

- FMT Reversi Android source code on Github
- FMT Reversi Backend source code on Github
- FMT Reversi source code on Github

Conclusions

- FMT Reversi can be improved:
  - Support for other platform (web)
  - Improved IA for CPU players
  - Google play game service
  - PS4 version cooming soon!
- Any question?

Thanks!