PLANTUML

METTEZ LA DOCUMENTATION DANS VOTRE CODE

Presentation créée par Nicolas Bossard (SOFT) pour le dev/test day
Slides
09 sept 2015

DISCLAIMER

A L'ORIGINE

La séquence de démarrage de MaLivebox

une procédure complexe avec de nombreuses étapes (interactions SI, utilisateur, Livebox...) évoluant en permanence

Javadoc insuffisante, description textuelle longue imposible à maintenir... nécessité de faire un schéma

Utilisation de yuml
(http://www.yuml.me/diagram/scruffy/activity/draw)

un échec :

- un outil en ligne extrèmement instable,
- un schéma moche,
- pas d'adhésion des autres développeurs

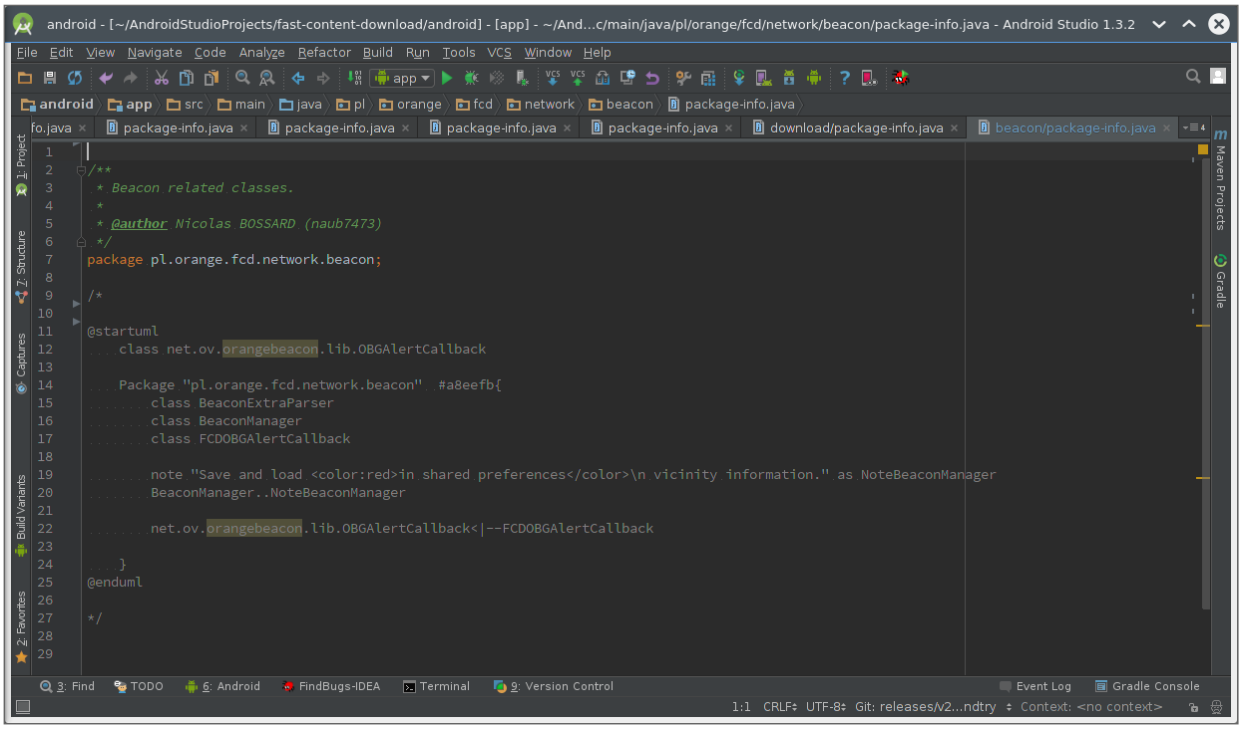
La solution... quitter le projet Ma Livebox
mais un an plus tard...plantUML

PLANTUML POUR QUOI FAIRE

générer des diagrammes UML à partir de descriptions textuelles que vous pouvez inclure dans votre code
éviter de s'appuyer sur des logiciels payants (visio, entreprise architect), peu répandus (dia), ou vieillissant (bouml),mais surtout extenes au source, et que donc on ouvre rarement ou jamais

TARIFS...

open source GPL, développement continu (actuellemer
v8031... sic)


android - [~/AndroidStudioProjects/fast-content-download/android] - [app] - ~/And...c/main/java/pl/orange/fcd/network/beacon/package-info.java - Android Studio 1.3.2
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app
android app src main java pl orange fcd network beacon package-info.java
package-info.java
package-info.java
package-info.java
package-info.java
download/package-info.java
beacon/package-info.java
/**
* Beacon related classes.
*
* @author Nicolas BOSSARD (naub7473)
*/
package pl.orange.fcd.network.beacon;
/*
@startuml
class net.ov.orangebeacon.lib.OBGAlertCallback
Package "pl.orange.fcd.network.beacon" #a8eefb{
class BeaconExtraParser
class BeaconManager
class FCDOBGAlertCallback
note "Save and load <color:red>in shared preferences</color>\n vicinity information." as NoteBeaconManager
BeaconManager..NoteBeaconManager
net.ov.orangebeacon.lib.OBGAlertCallback<|--FCDOBGAlertCallback
}
@enduml
*/
3: Find
TODO
6: Android
FindBugs-IDEA
Terminal
9: Version Control
Event Log
Gradle Console
1:1 CRLF UTF-8 Git: releases/v2...ndtry Context: <no context>

CAPACITÉS DE PLANTUML

DIAGRAMME DE SÉQUENCE

```
@startuml

Bob->Alice : hello

@enduml
```

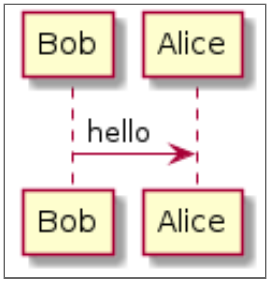


DIAGRAMME DE CLASSE

```
@startuml

        Campaign "1" *-- "many" Room
        Room "1" *--  "many" Measure
        Measure "1" *--  "many" OneWifiMeasure

@enduml
```

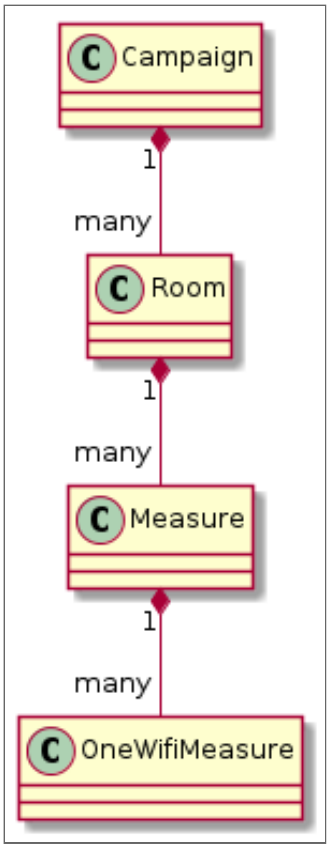


DIAGRAMME DE CAS D'UTILISATION

```
@startuml
developpeur --> (doc dans le code)
: écrit et maintient
po --> (doc dans le code) : écrit
et relit
(doc dans le code) --> testeur :
utilise une doc à jour

@enduml
```

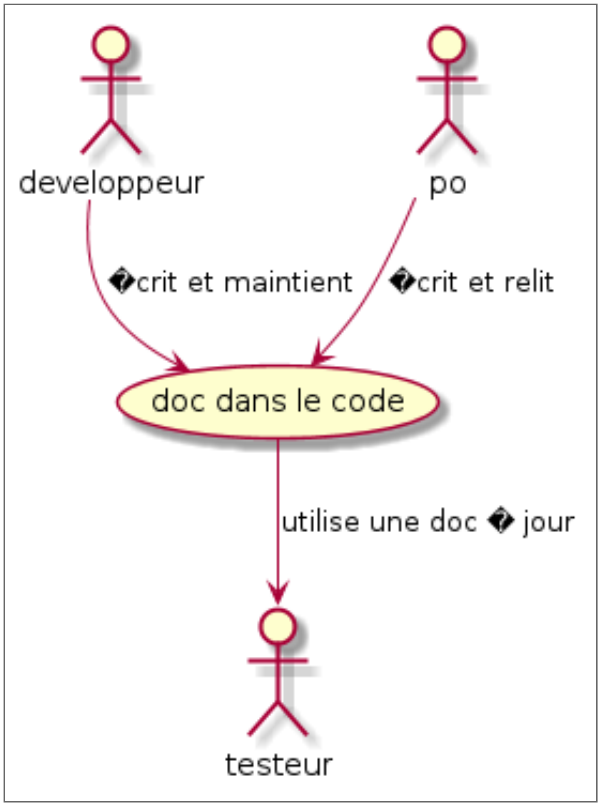
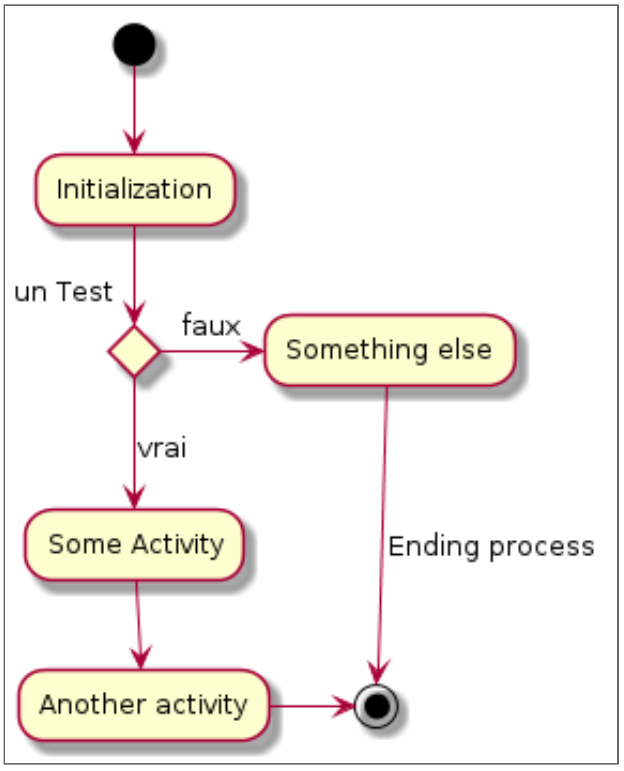


DIAGRAMME D'ACTIVITÉ

```
@startuml
(*) --> "Initialization"

if "un Test" then
  -->[vrai] "Some Activity"
  --> "Another activity"
  -right-> (*)
else
  ->[faux] "Something else"
  -->[Ending process] (*)
endif

@enduml
```



MAIS AUSSI...

DIAGRAMME WIREFRAME

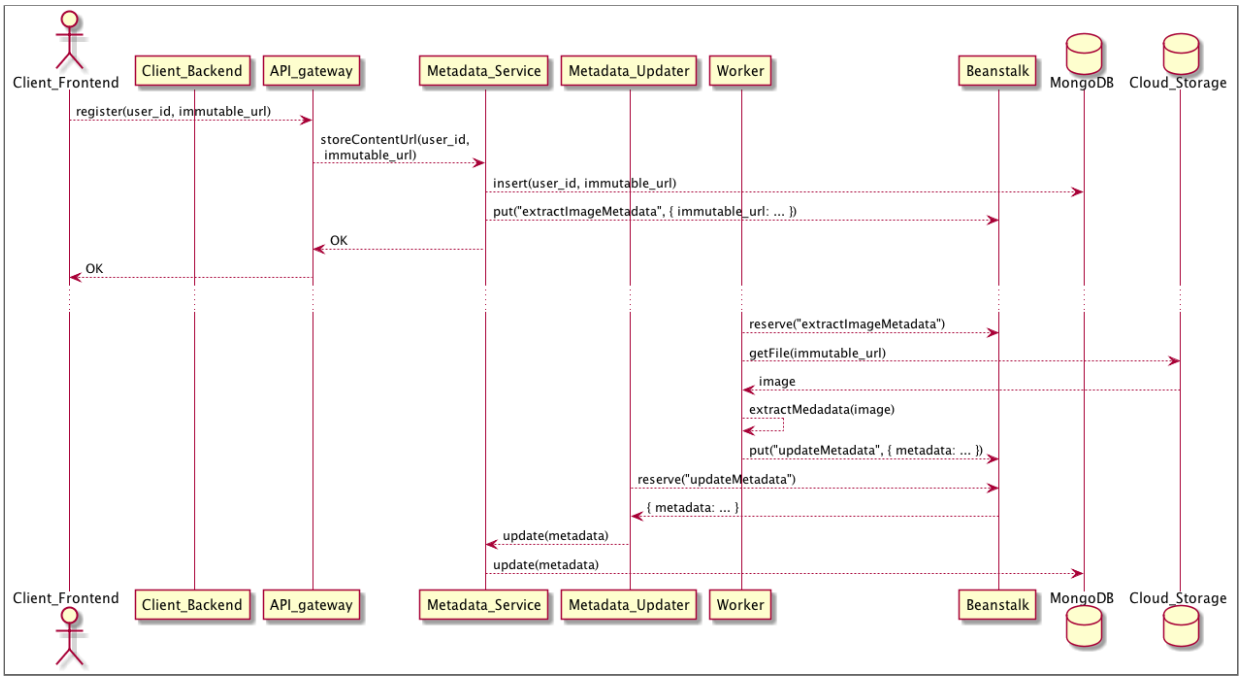
```
@startsalt
{+
  Login    | "nbossard   "
  Password | "****     "
  [Cancel] | [  OK   ]
}
@endsalt
```

Diagrammes de composants
Diagramme de déploiement
Diagramme d'état
Diagramme d'objet

EXEMPLE D'UTILISATION DANS NOS PROJETS

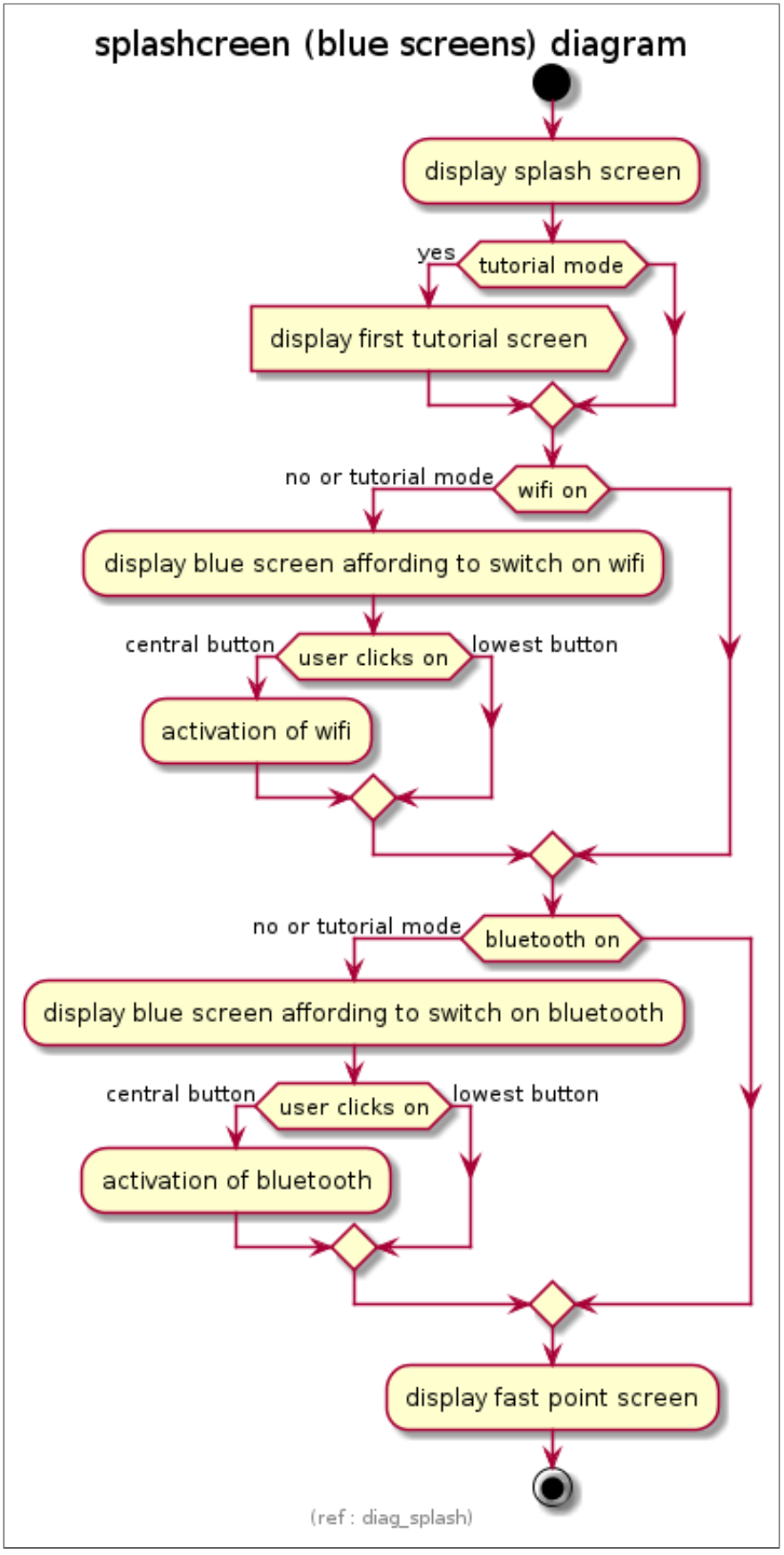
BoxWidgets, Fast content Download, Homelan, Mooc

DIAGRAMMES DE SEQUENCE

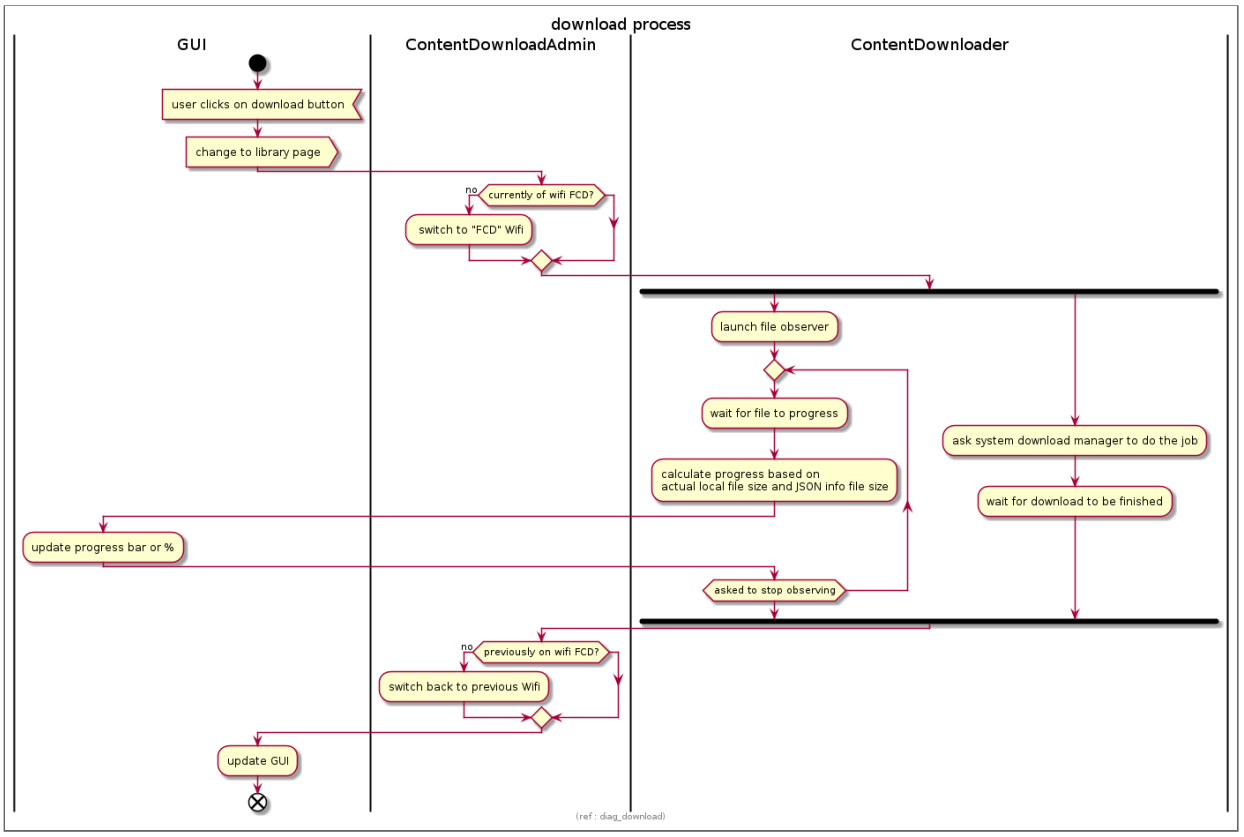


ALGORITHMES

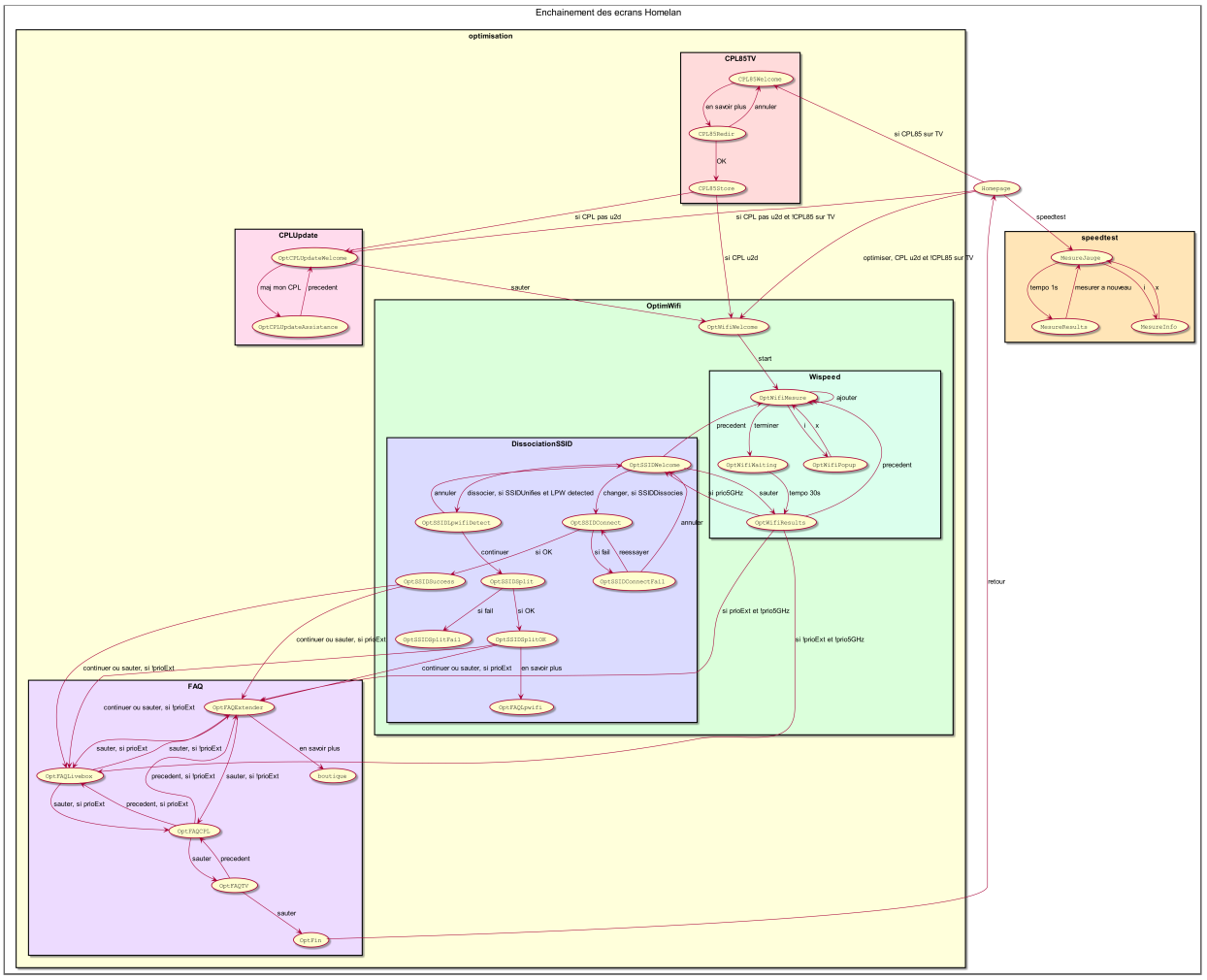
Pour les algos que vous ne cessez de réexpliquer ou pour le reverse engineering


splashcreen (blue screens) diagram
display splash screen
yes
tutorial mode
display first tutorial screen
no or tutorial mode
wifi on
display blue screen affording to switch on wifi
central button
user clicks on
lowest button
activation of wifi
no or tutorial mode
bluetooth on
display blue screen affording to switch on bluetooth
central button
user clicks on
lowest button
activation of bluetooth
display fast point screen
(ref : diag_splash)

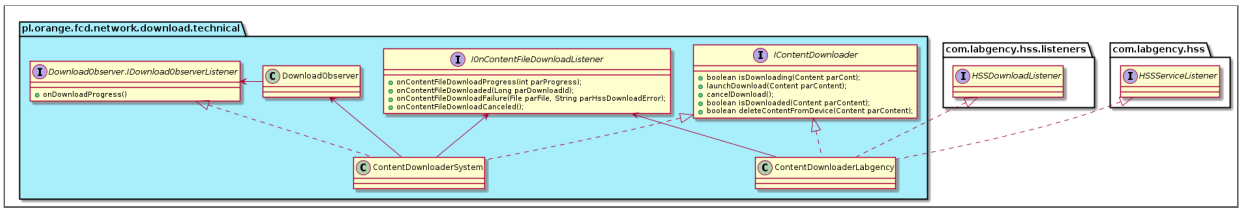
ALGORITHMES (EX. 2)



ENCHAINEMENT DES ÉCRANS

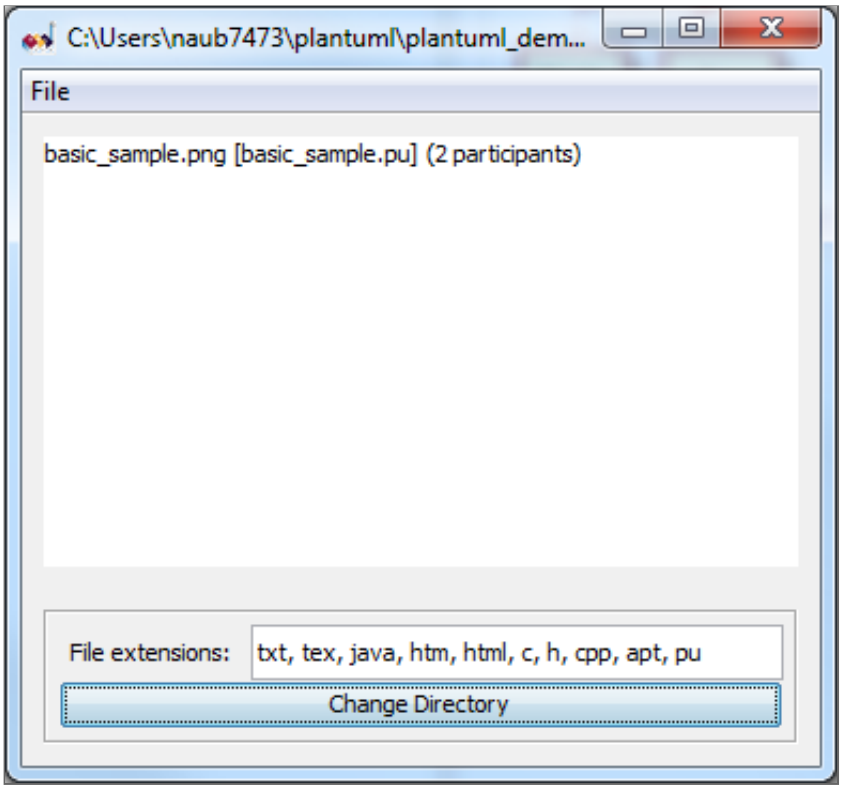


DIAGRAMMES DE CLASSE



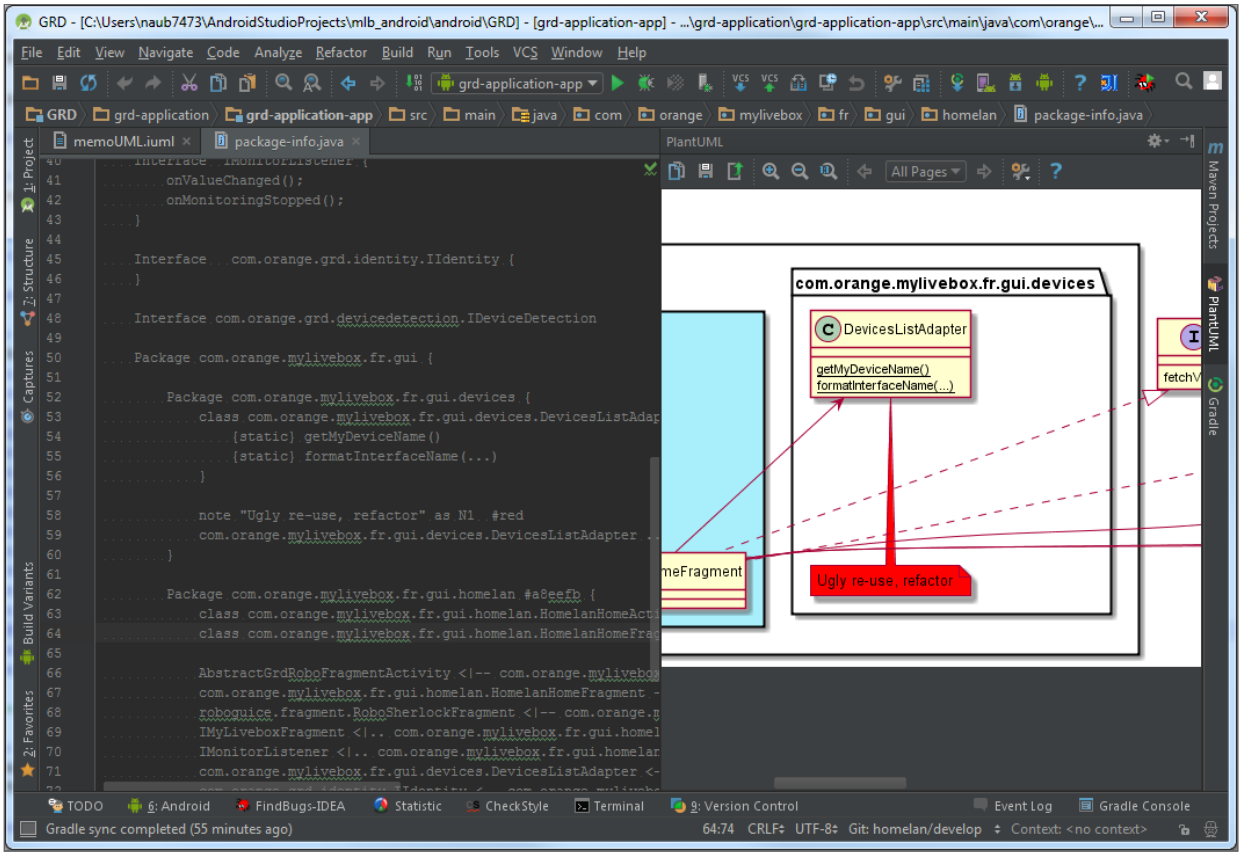
UTILISATION

UTILISATION DU JAR PLANTUML



double click (windows) ou (linux) java -jar plantuml.ja

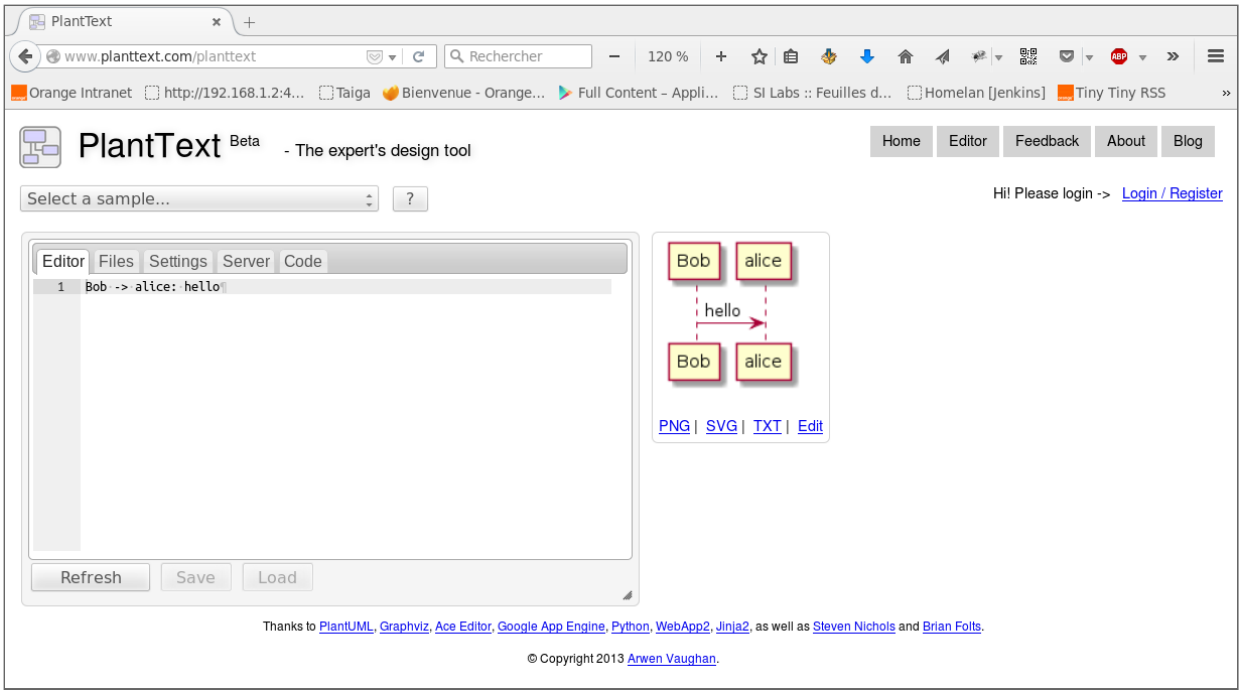
DIRECTEMENT DANS L'IDE (VIA UN PLUGIN)



SUR L'INTERNET

Pour des test rapides

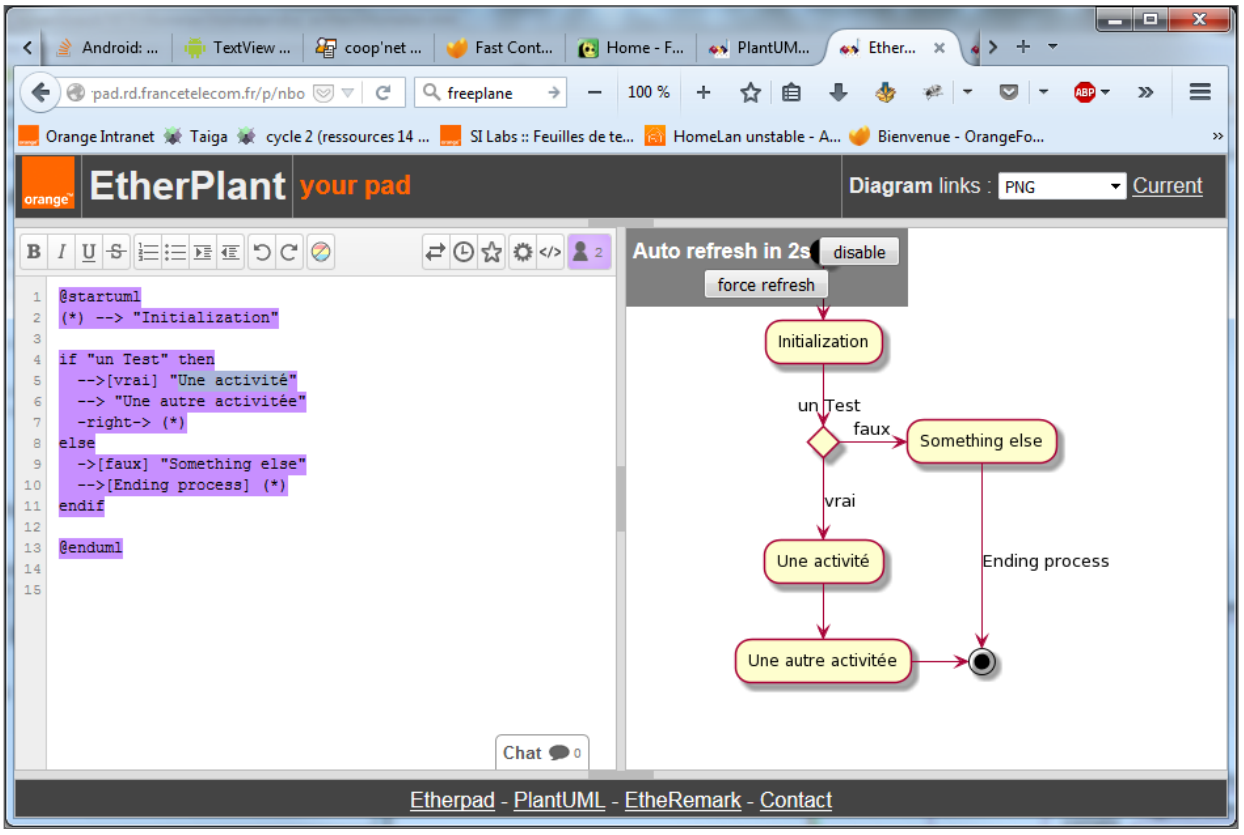
http://www.planttext.com/planttext



SUR L'INTRANET

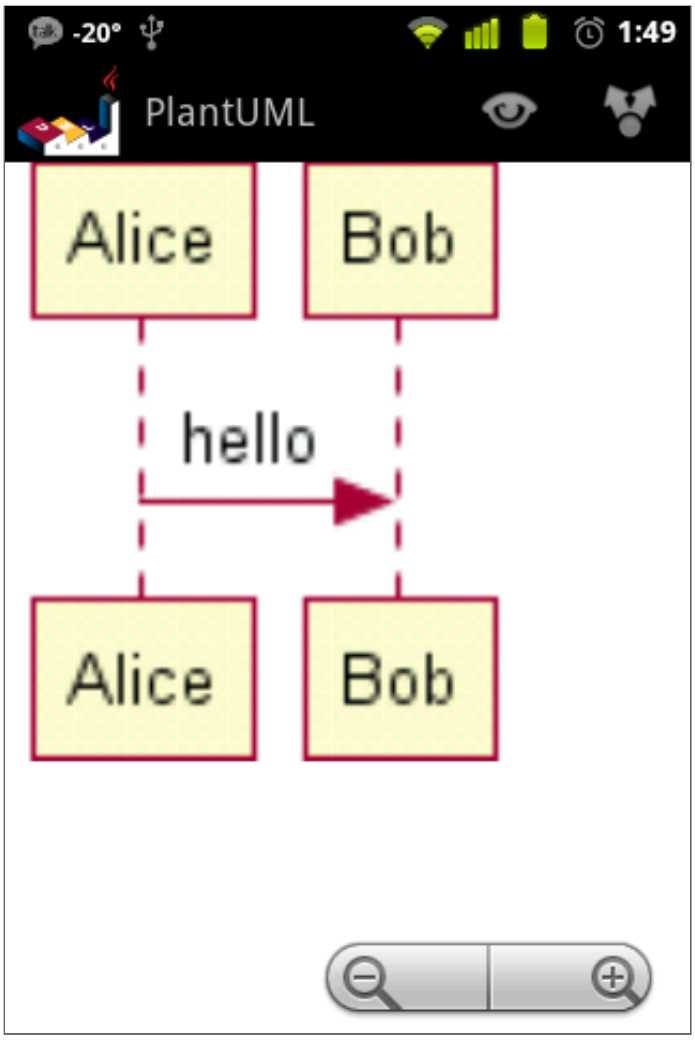
http://plantuml-etherpad.kermit.rd.francetelecom.fr/

Merci à Matthieu SALVAT (présent sur Rennes pour les dev et test days)



DANS LA CHAINE D'INTÉGRATION CONTINUE (JENKINS)

VIA UNE APPLICATION ANDROID...



Bref des dizaines de manière d'utiliser plantuml...
90 à ce jour sur **http://fr.plantuml.com/running.html**

INSTALLATION

Attention le jar repose sur graphviz

http://graphviz.org/Download..php

DEMO

utilisation du plantuml.jar

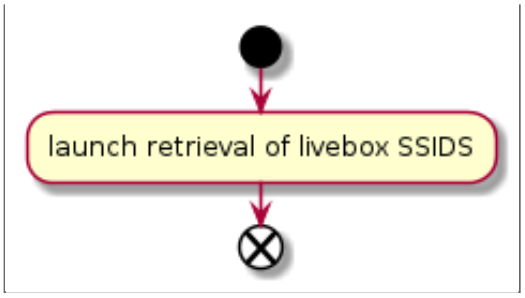
RETOURS D'USAGE EN VRAC

La syntaxe est facile à appréhender et tolérante, on se bat assez peu contre l'outil
Les graphes générés sont lisibles... parce qu'ils sont écrits manuellement et donc orientés pour faire passer un message
La doc sur **http://plantuml.com/** est très complète(bien que d'aspect vieillotte) et les manques éventuels se trouvent facilement dans les forums

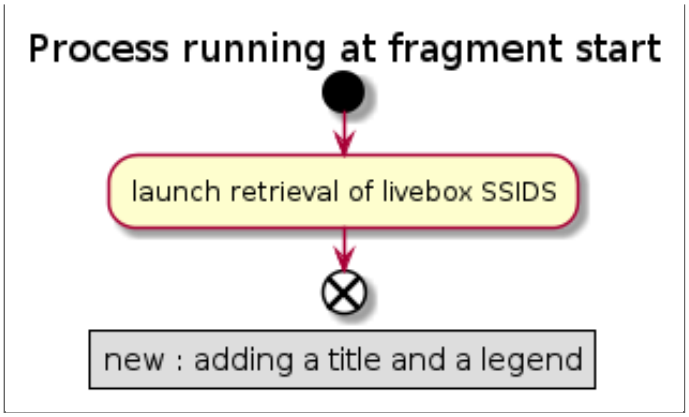
RETOURS D'USAGE EN VRAC (SUITE)

Pour les plus (grands) diagrammes de classes, utilisatic
des instruction include et hide
Ne pas trop essayer de gérer la mise en forme, ça résou
des probs temporairement... mais perturbe l'évolution

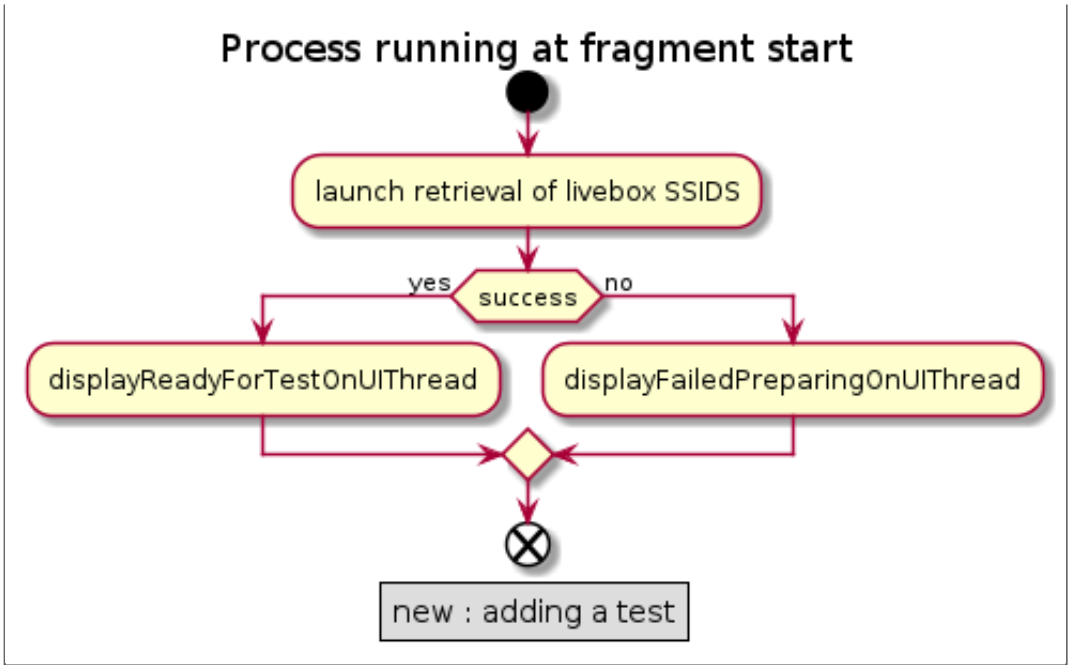
EXEMPLE
ÉCRITURE DIAG D'ACTIVITÉ



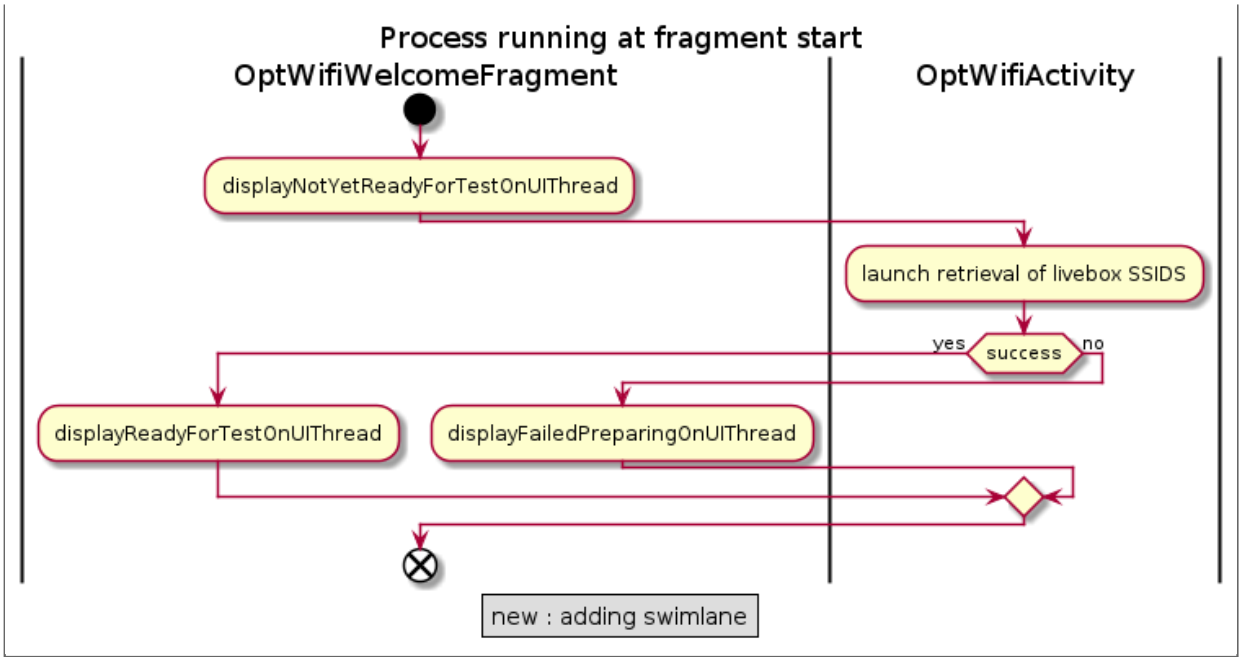

```
@startuml
        start
        :launch retrieval of livebox SSIDS;
        end
@enduml
```



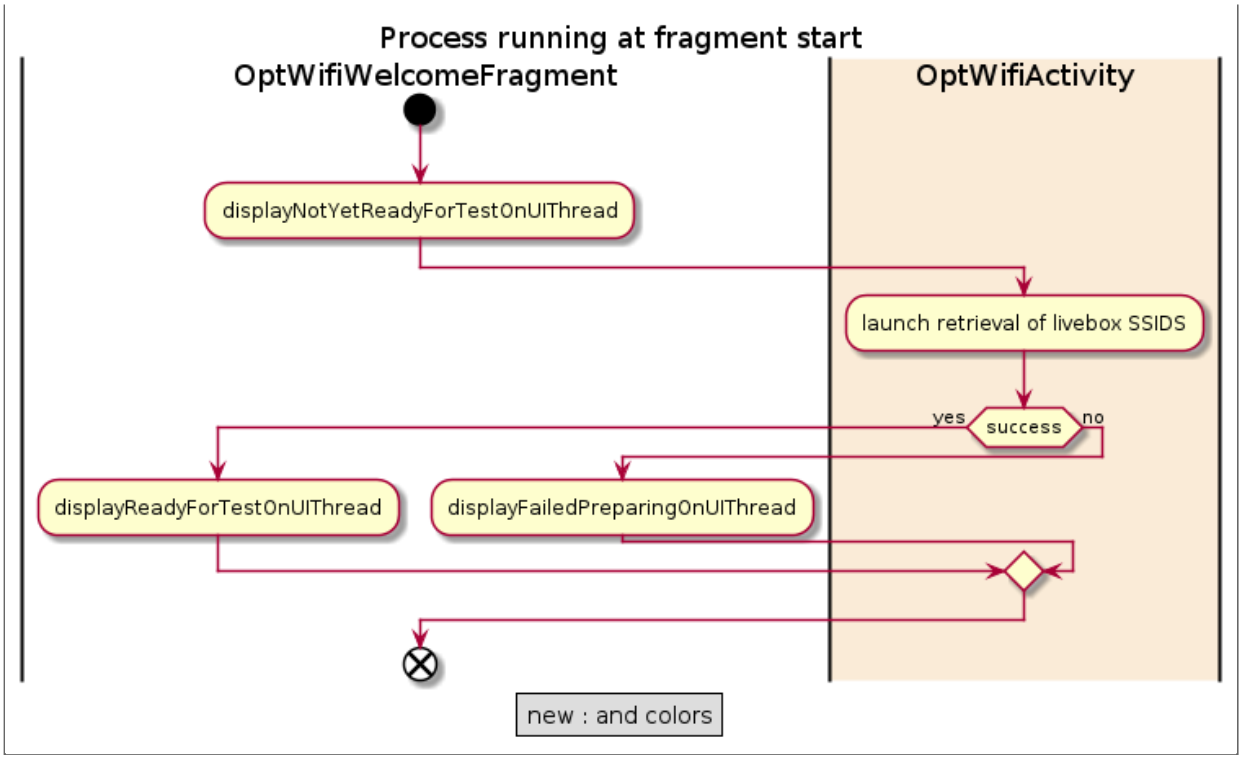

```
@startuml
        title Process running at fragment start

        start
        :launch retrieval of livebox SSIDS;
        end

        legend
                new : adding a title and a legend
        endlegend
@enduml
```




```
@startuml
        title Process running at fragment start

        start
        :launch retrieval of livebox SSIDS;
        if (success) then (yes)
                :displayReadyForTestOnUIThread;
        else (no)

:displayFailedPreparingOnUIThread;
        endif
        end

        legend
                new : adding a test
        endlegend
@enduml
```

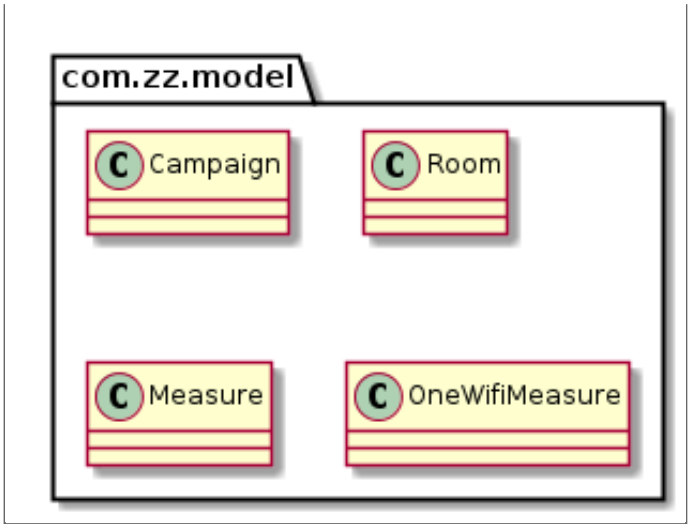



```
@startuml
title Process running at fragment start

|OptWifiWelcomeFragment|
start
:displayNotYetReadyForTestOnUIThread;
|OptWifiActivity|
:launch retrieval of livebox SSIDS;
if (success) then (yes)
        |OptWifiWelcomeFragment|
        :displayReadyForTestOnUIThread;
else (no)
        :displayFailedPreparingOnUIThread;
endif
end

        legend
                new : adding swimlane
        endlegend
@enduml
```

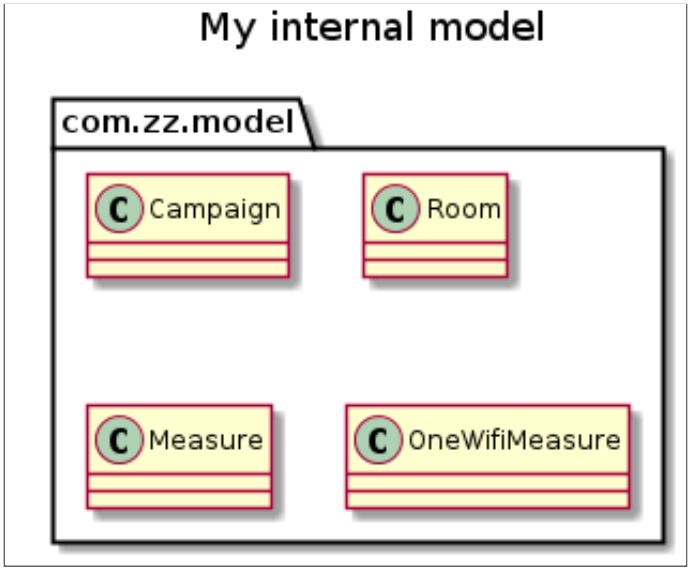



```
@startuml
title Process running at fragment start
|OptWifiWelcomeFragment|
start
:displayNotYetReadyForTestOnUIThread;
|#AntiqueWhite|OptWifiActivity|
:launch retrieval of livebox SSIDS;
if (success) then (yes)
        |OptWifiWelcomeFragment|
        :displayReadyForTestOnUIThread;
else (no)
        :displayFailedPreparingOnUIThread;
endif
end

        legend
                new : and colors
        endlegend
@enduml
```

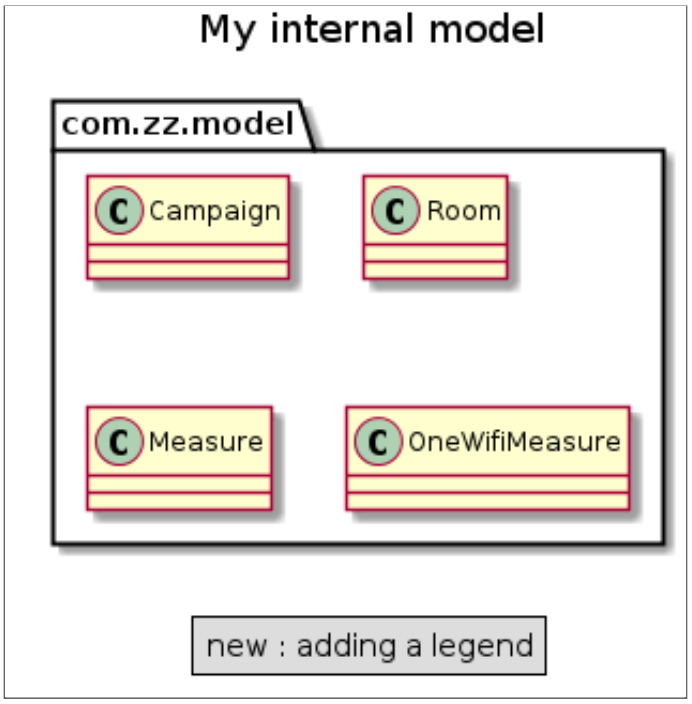
EXEMPLE
ÉCRITURE DIAG DE CLASSE



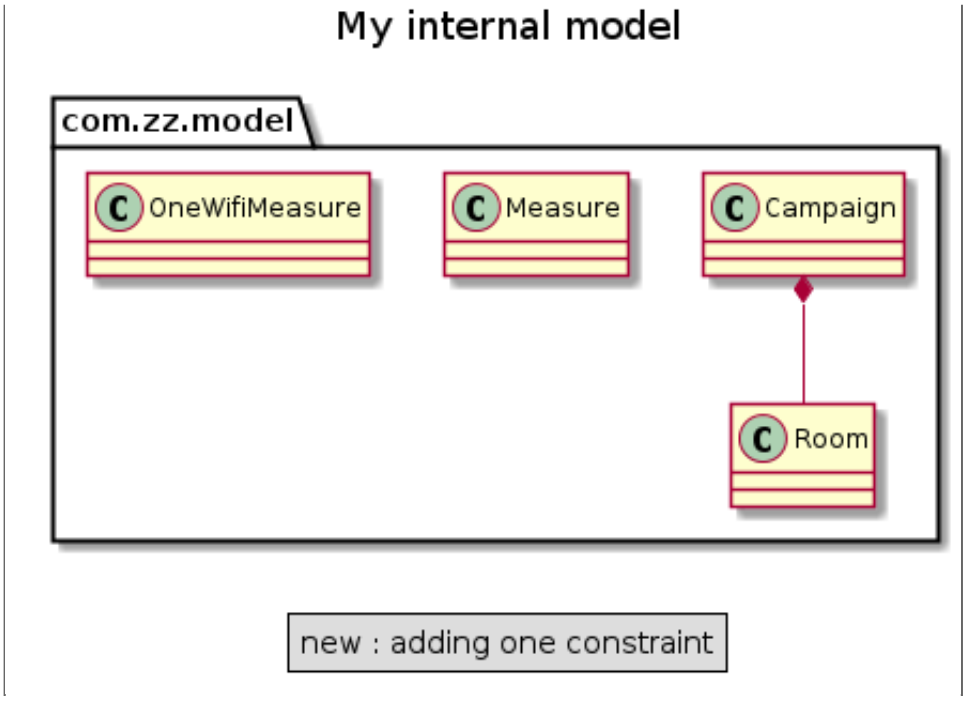

```
@startuml

        class com.zz.model.Campaign
        class com.zz.model.Room
        class com.zz.model.Measure
        class com.zz.model.OneWifiMeasure

@enduml
```



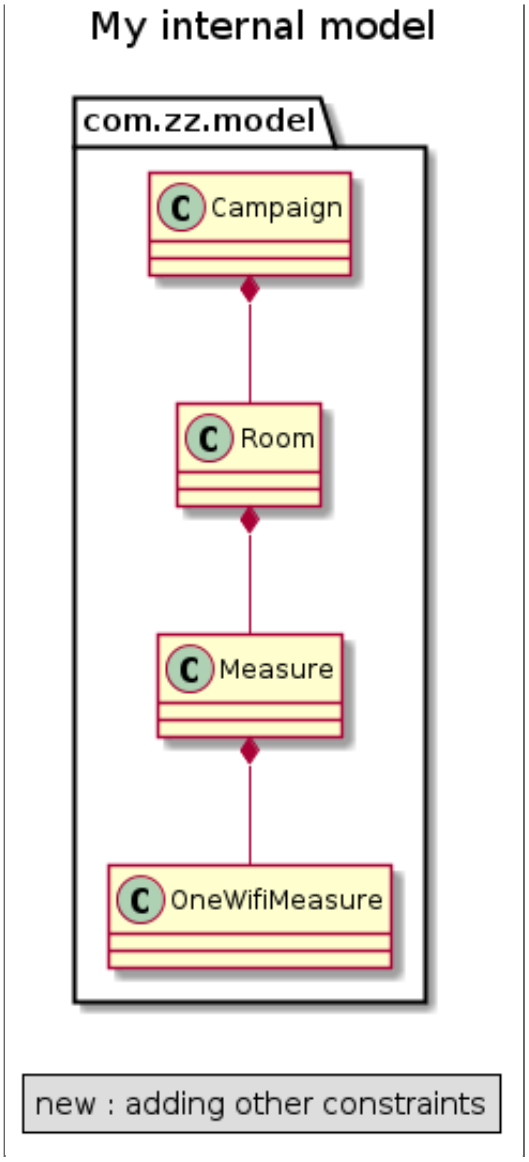

```
@startuml
        title My internal model

        class com.zz.model.Campaign
        class com.zz.model.Room
        class com.zz.model.Measure
        class com.zz.model.OneWifiMeasure

@enduml
```



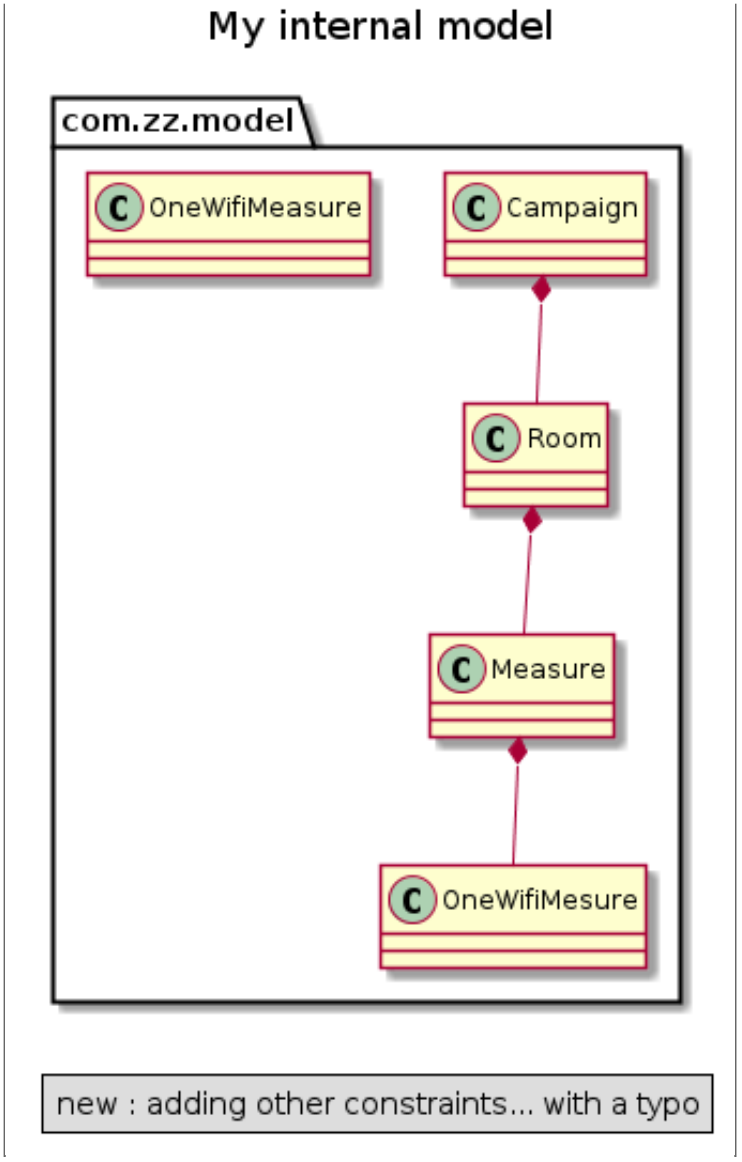

```
@startuml
        title My internal model

        class com.zz.model.Campaign
        class com.zz.model.Room
        class com.zz.model.Measure
        class com.zz.model.OneWifiMeasure

        legend
                new : adding a legend
        endlegend
@enduml
```



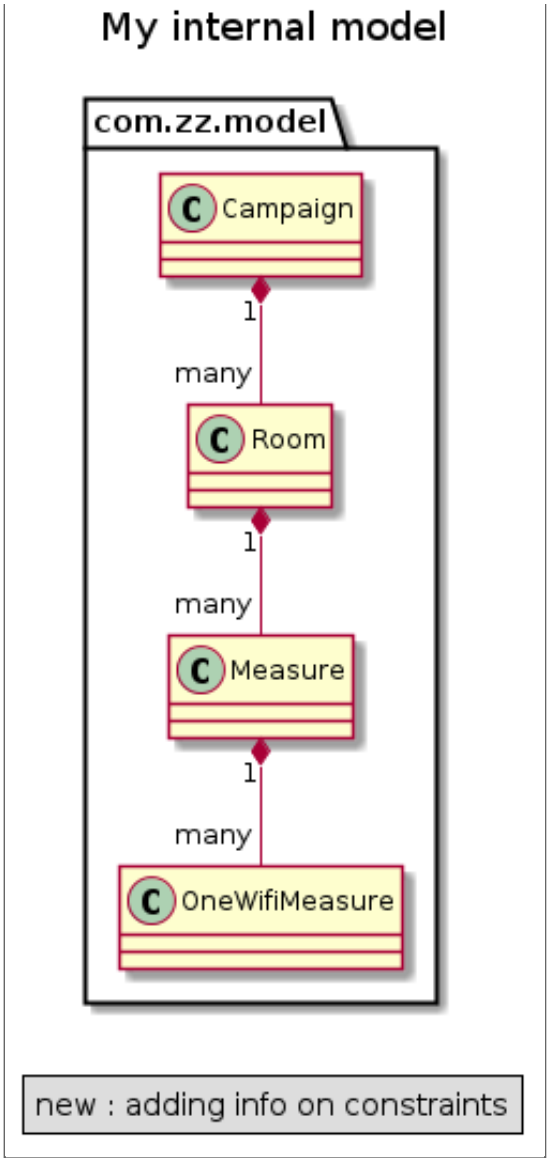

```
@startuml
        title My internal model

        class com.zz.model.Campaign
        class com.zz.model.Room
        class com.zz.model.Measure
        class com.zz.model.OneWifiMeasure

        com.zz.model.Campaign *--
com.zz.model.Room

        legend
                new : adding one constraint
        endlegend
@enduml
```


My internal model
com.zz.model
Campaign
Room
Measure
OneWifiMeasure
new : adding other constraints

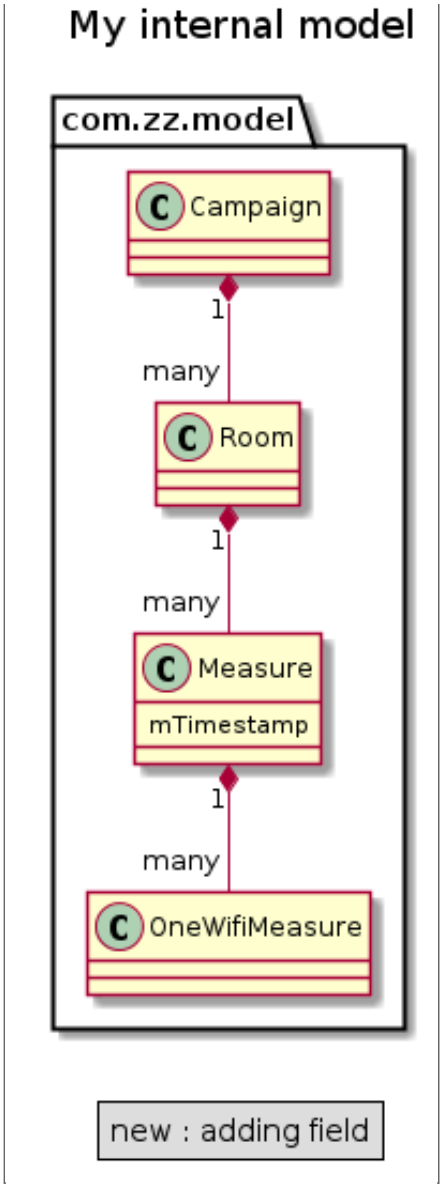
```
@startuml
        title My internal model

        class com.zz.model.Campaign
        class com.zz.model.Room
        class com.zz.model.Measure
        class com.zz.model.OneWifiMeasure

        com.zz.model.Campaign *--
com.zz.model.Room
        com.zz.model.Room *--
com.zz.model.Measure
        com.zz.model.Measure *--
com.zz.model.OneWifiMeasure

        legend
                new : adding other constraints
        endlegend
@enduml
```


My internal model
com.zz.model
OneWifiMeasure
Campaign
Room
Measure
OneWifiMesure
new : adding other constraints... with a typo

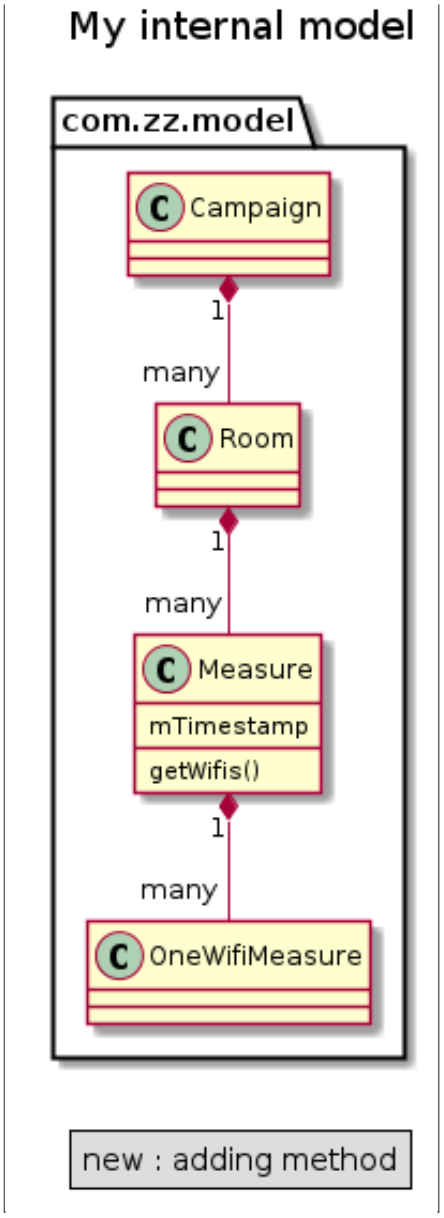
```
@startuml
        title My internal model

        class com.zz.model.Campaign
        class com.zz.model.Room
        class com.zz.model.Measure
        class com.zz.model.OneWifiMeasure

        com.zz.model.Campaign *--
com.zz.model.Room
        com.zz.model.Room *--
com.zz.model.Measure
        com.zz.model.Measure *--
com.zz.model.OneWifiMesure

        legend
                new : adding other constraints...
with a typo
        endlegend
@enduml
```


My internal model
com.zz.model
Campaign
1
many
Room
1
many
Measure
1
many
OneWifiMeasure
new : adding info on constraints

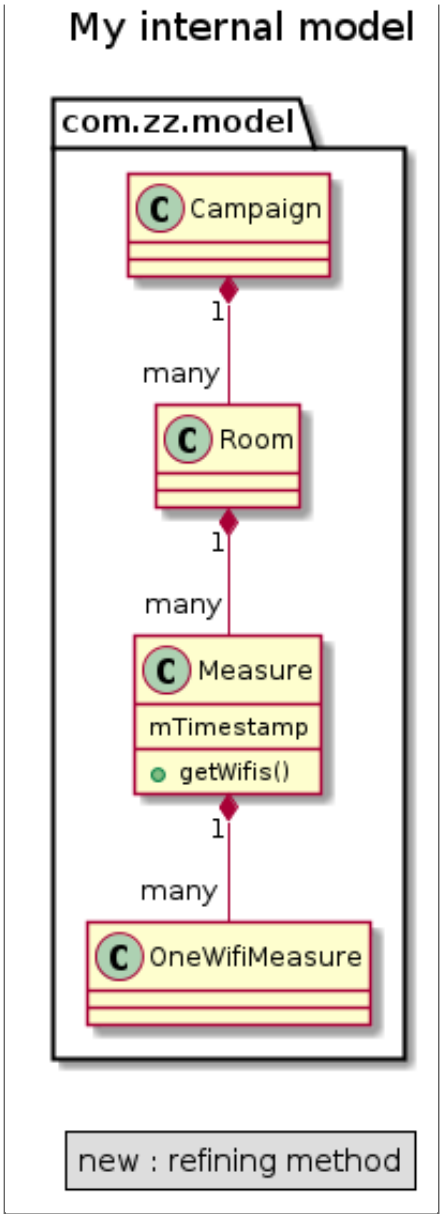
```
@startuml
        title My internal model

        class com.zz.model.Campaign
        class com.zz.model.Room
        class com.zz.model.Measure
        class com.zz.model.OneWifiMeasure

        com.zz.model.Campaign "1" *-- "many"
com.zz.model.Room
        com.zz.model.Room "1" *--  "many"
com.zz.model.Measure
        com.zz.model.Measure "1" *--  "many"
com.zz.model.OneWifiMeasure

        legend
                new : adding info on constraints
        endlegend
@enduml
```


My internal model
com.zz.model
Campaign
1
many
Room
1
many
Measure
mTimestamp
1
many
OneWifiMeasure
new : adding field

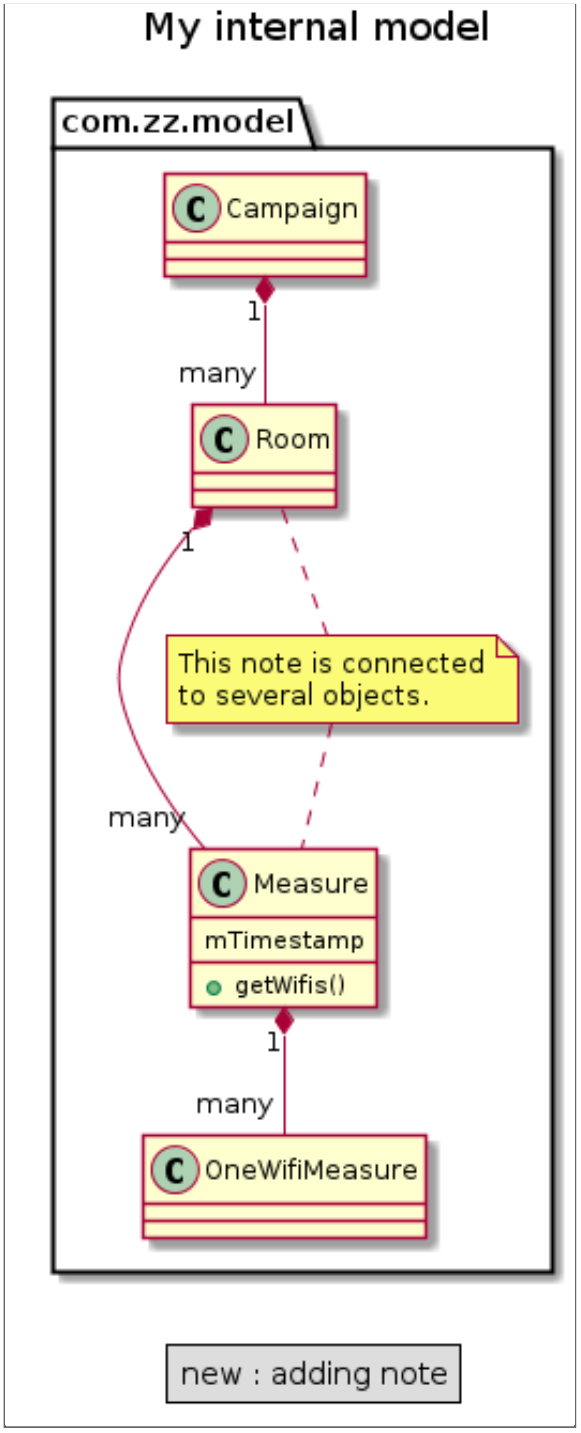
```
@startuml
        title My internal model

        class com.zz.model.Campaign
        class com.zz.model.Room
        class com.zz.model.Measure {
                mTimestamp
        }
        class com.zz.model.OneWifiMeasure

        com.zz.model.Campaign "1" *-- "many"
com.zz.model.Room
        com.zz.model.Room "1" *--  "many"
com.zz.model.Measure
        com.zz.model.Measure "1" *--  "many"
com.zz.model.OneWifiMeasure

        legend
                new : adding field
        endlegend
@enduml
```


My internal model
com.zz.model
Campaign
1
many
Room
1
many
Measure
mTimestamp
getWifis()
1
many
OneWifiMeasure
new : adding method

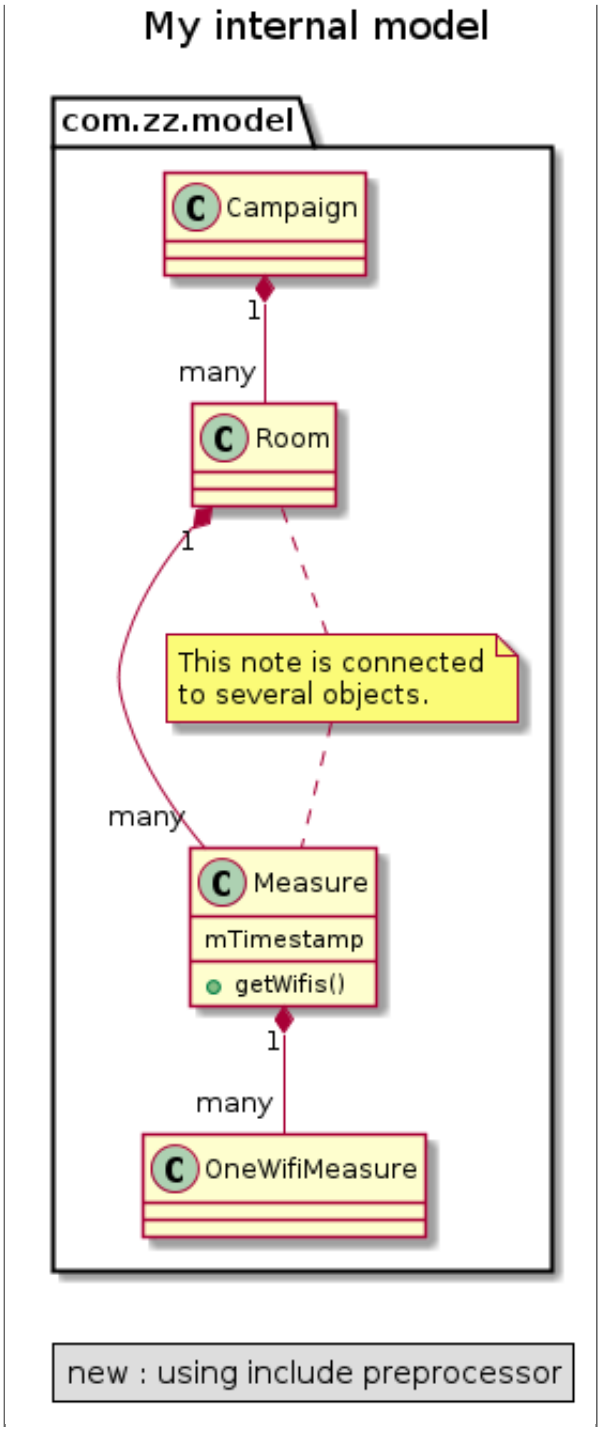
```
@startuml
        title My internal model

        class com.zz.model.Campaign
        class com.zz.model.Room
        class com.zz.model.Measure {
                mTimestamp
                getWifis()
        }
        class com.zz.model.OneWifiMeasure

        com.zz.model.Campaign "1" *-- "many"
com.zz.model.Room
        com.zz.model.Room "1" *--  "many"
com.zz.model.Measure
        com.zz.model.Measure "1" *--  "many"
com.zz.model.OneWifiMeasure

        legend
                new : adding method
        endlegend
@enduml
```


My internal model
com.zz.model
Campaign
1
many
Room
1
many
Measure
mTimestamp
getWifis()
1
many
OneWifiMeasure
new : refining method

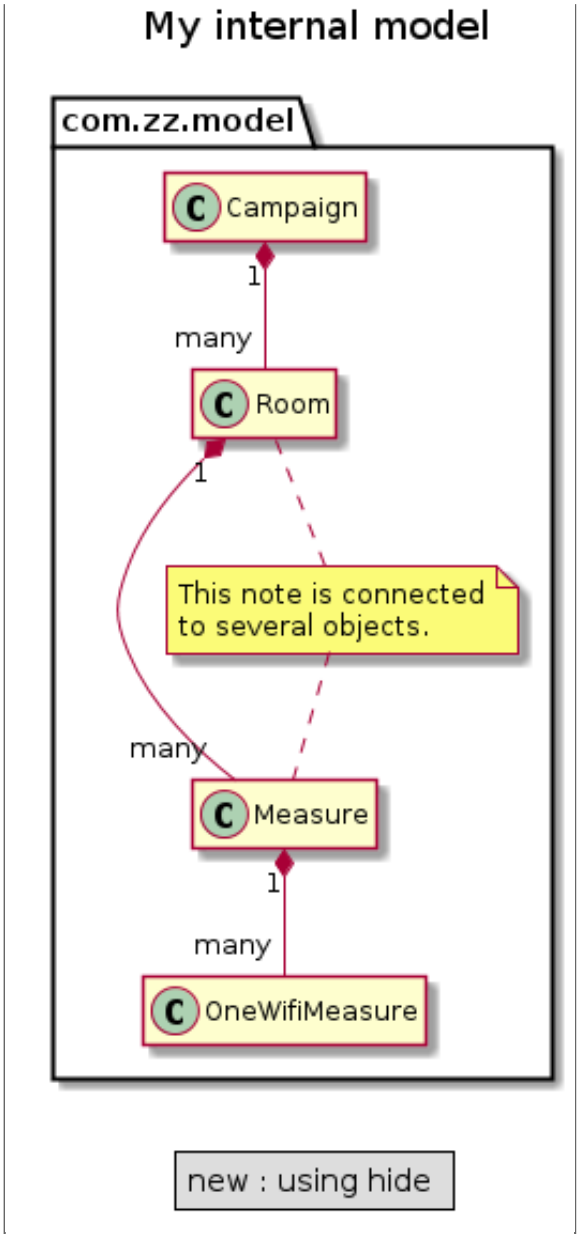
```
@startuml
        title My internal model

        class com.zz.model.Campaign
        class com.zz.model.Room
        class com.zz.model.Measure {
                mTimestamp
                + getWifis()
        }
        class com.zz.model.OneWifiMeasure

        com.zz.model.Campaign "1" *-- "many"
com.zz.model.Room
        com.zz.model.Room "1" *--  "many"
com.zz.model.Measure
        com.zz.model.Measure "1" *--  "many"
com.zz.model.OneWifiMeasure

        legend
                new : refining method
        endlegend
@enduml
```


My internal model
com.zz.model
Campaign
1
many
Room
1
This note is connected
to several objects.
many
Measure
mTimestamp
getWifis()
1
many
OneWifiMeasure
new : adding note

```
@startuml
        title My internal model

        class com.zz.model.Campaign
        class com.zz.model.Room
        class com.zz.model.Measure {
                mTimestamp
                + getWifis()
        }
        class com.zz.model.OneWifiMeasure

        com.zz.model.Campaign "1" *-- "many"
com.zz.model.Room
        com.zz.model.Room "1" *--  "many"
com.zz.model.Measure
        com.zz.model.Measure "1" *--  "many"
com.zz.model.OneWifiMeasure


        note "This note is connected\nto several objects." as
com.zz.model.Note
        com.zz.model.Room .. com.zz.model.Note
        com.zz.model.Note .. com.zz.model.Measure

        legend
                new : adding note
        endlegend
@enduml
```


My internal model
com.zz.model
Campaign
1
many
Room
1
This note is connected
to several objects.
many
Measure
mTimestamp
getWifis()
1
many
OneWifiMeasure
new : using include preprocessor

```
@startuml
        title My internal model

        !include sample_class_diag_11measure.pu

        class com.zz.model.Campaign
        class com.zz.model.Room

        com.zz.model.Campaign "1" *-- "many"
com.zz.model.Room
        com.zz.model.Room "1" *--  "many"
com.zz.model.Measure

        legend
                new : using include preprocessor
        endlegend
@enduml
```


My internal model
com.zz.model
Campaign
1
many
Room
1
This note is connected
to several objects.
many
Measure
1
many
OneWifiMeasure
new : using hide

```
@startuml
        title My internal model

        !include sample_class_diag_11measure.pu
        hide methods
        hide members

        class com.zz.model.Campaign
        class com.zz.model.Room

        com.zz.model.Campaign "1" *-- "many"
com.zz.model.Room
        com.zz.model.Room "1" *--  "many"
com.zz.model.Measure

        legend
                new : using hide
        endlegend
@enduml
```

QUESTIONS

... elles sont les très bienvenues

MERCI

Vous souhaitez récupérer ces slides : Nicolas.Bossard@orange.com