

# MOI JE CODE M'ADOC

Mehdi Rebiai

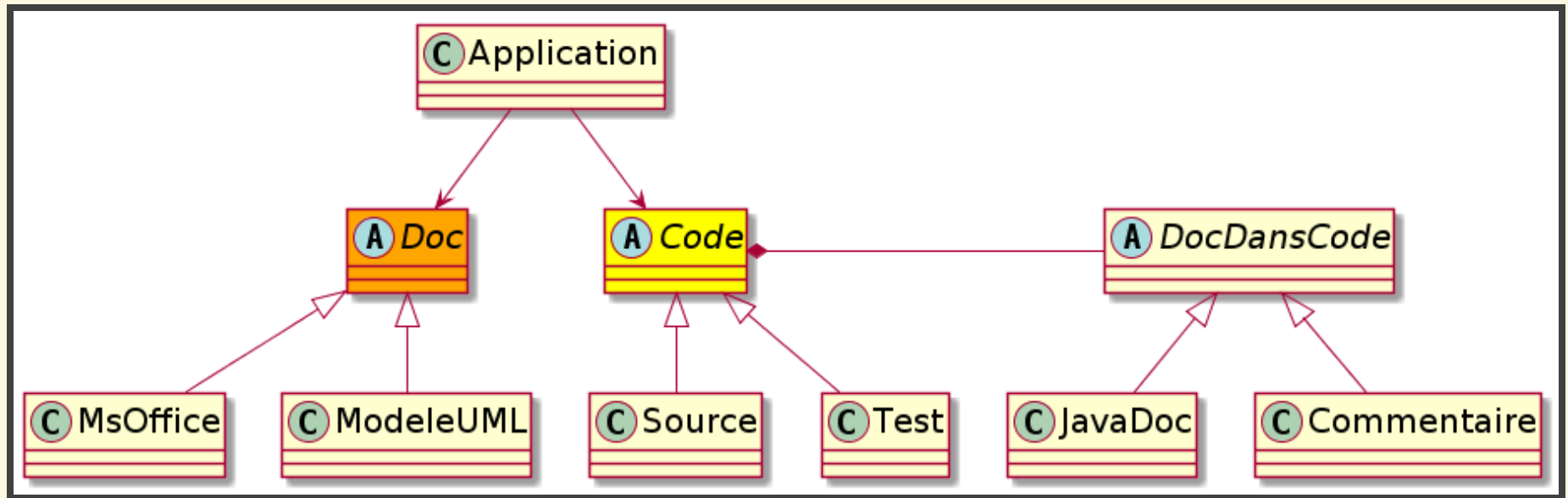
# A PROPOS DE MOI

- Je suis Mehdi Rebiai
- Architecte technique SQLI
- Plus de 10 ans sur Java, JavaEE, UML, ...
- ... et maintenant JS :)
- Utilisateur AsciiDoctor & PlantUML depuis 1 an

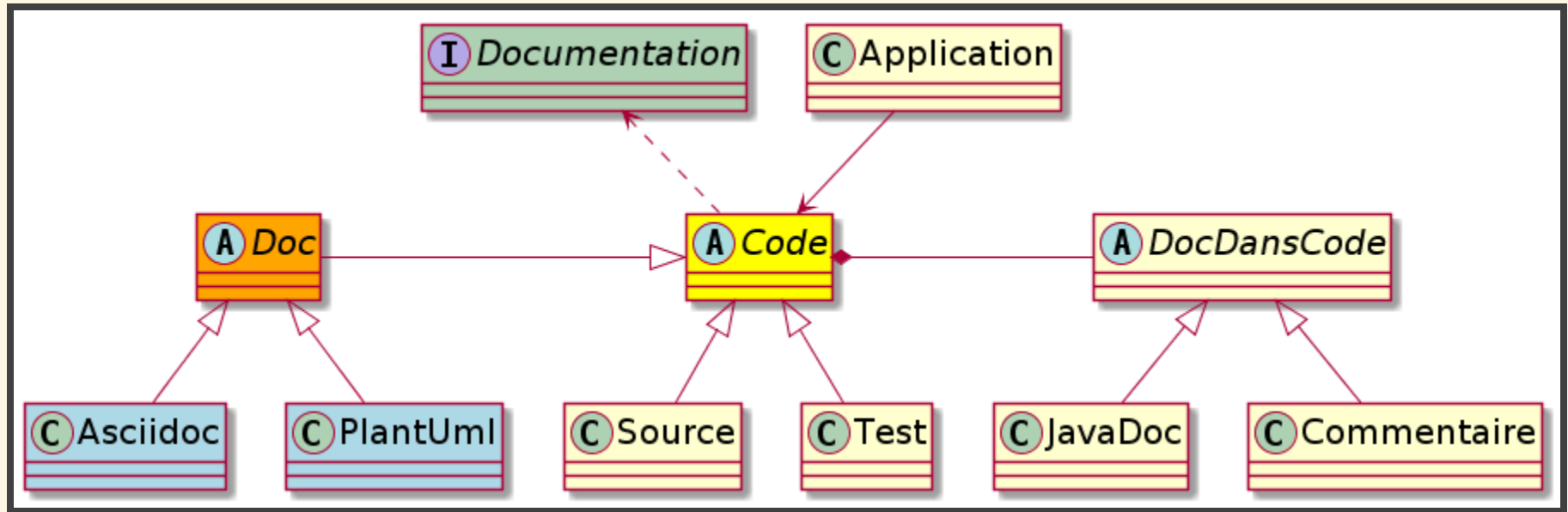
# DOC?

- README
- Spécification
- Conception technique
- Manuel d'utilisation
- Wiki
- Rapport de stage
- ...

# DOC VS CODE ?

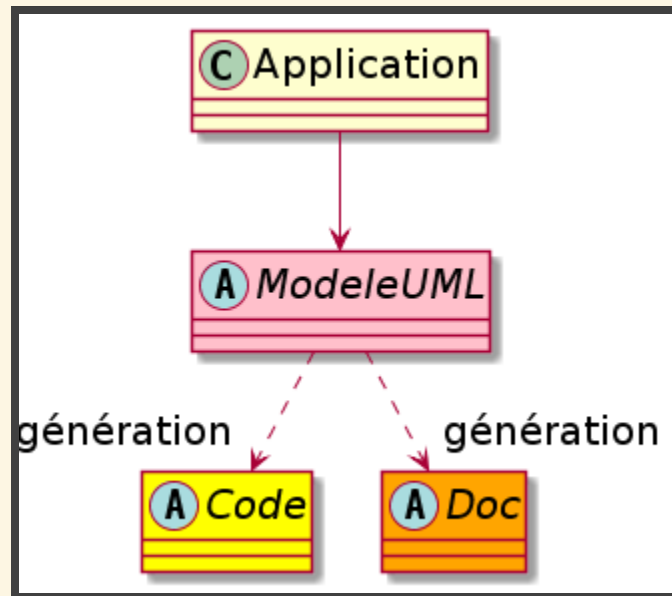


# ET SI DOC = CODE ?



# AVANT... JE CROYAIS AU MDA

- Formateur UML
- Evangéliste MDA



# MAINTENANT... JE CROIS AU FICHER PLAT



- Simple
- Travail collaboratif

# RÉFLEXION CHEZ SQLI

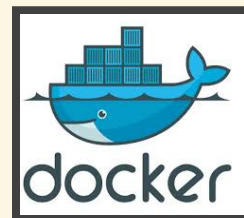
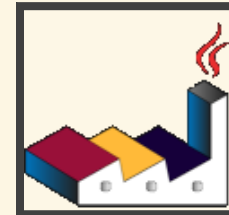
Outils sur le Poste du développeur?



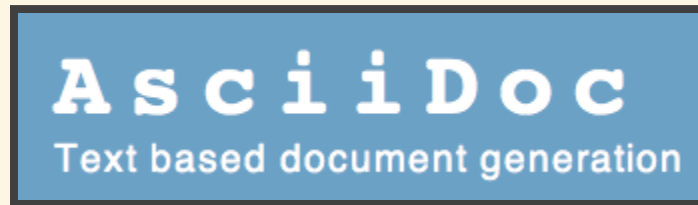
# AVANT CHEZ SQLI



# APRÈS CHEZ SQLI

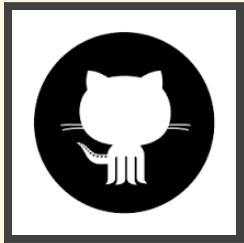


# ASCIIDOC - KEZAKO?



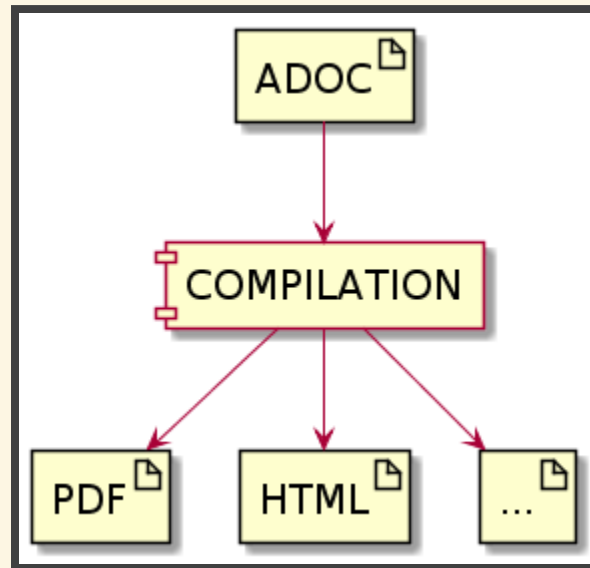
# ASCIIDOC - DÉFINITION

- Language de balisage léger ⇒ générer des documents
- Créé en 2002, processeur en Python
- Format géré par GitHub, GitLab, ...
- Doc SpringFramework en asciidoc!



# ASCIIDOC - OBJECTIFS

Contenu > Forme



# ASCIIDOC - VITE UN EXEMPLE

```
= Vite un exemple en AsciiDoc
Mehdi Rebiai <mrebiai@sqli.com>

Introduction à http://asciidoc.org[AsciiDoc].

== Partie 1

* Point1
* Point2
** Point2.1

== Partie 2

. Item1
. Item2
```

# RENDU HTML5

## Vite un exemple en AsciiDoc

---

Introduction à [AsciiDoc](#).

### Partie 1

- Point1
- Point2
  - Point2.1

---

### Partie 2

1. Item1
2. Item2

# ASCIIDOCTOR - KEZAKO?



- Bye bye Python! Bonjour Ruby!
- <http://asciidoctor.org>
- v1.5.2, novembre 2014
- **Open source**
- Un écosystème, une communauté



# ASCIIDOCTOR - ON PEUT...

```
= Include
image::monImage.png[height=100]
video::monVideo.avi[]

[source, javascript]
----
include::HelloWorld.js[lines=2..4]
----

:val: cas1
include::test.adoc[]
:val: cas2
include::test.adoc[]
```

+ sommaire, tableaux, diagrammes, formules, ...

# LE "JAVAC"

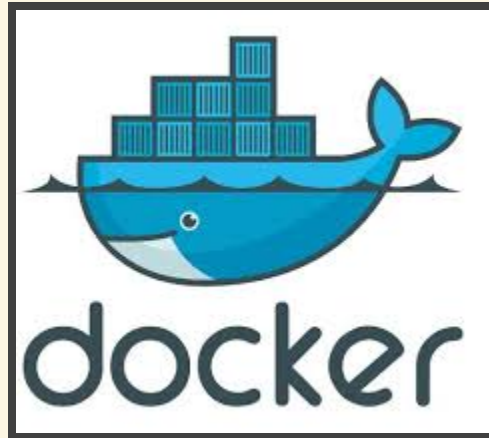
```
$ asciidoctor monfichier.adoc
```

Génération d'un fichier monfichier.html (html5 par défaut)

```
$ asciidoctor -D <output> -T <template> -r <requirements> monfichier.adoc
```

 la gem asciidoctor doit être installée

# INSTALLER RUBY? NON $\Rightarrow$ DOCKER!



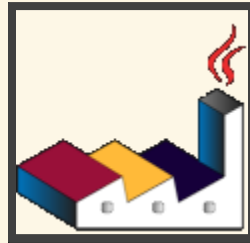
```
$ docker run --rm -v <VOL> asciidoctor/docker-asciidoctor <CMD>
```

# ET ENCORE BEAUCOUP PLUS

<http://asciidoctor.org>

- ❗ Simplicité & Productivité :)

# PLANTUML - KEZAKO?



# PLANTUML

- Le fichier plat pour l'UML!
  - <http://fr.plantuml.com>
  - V8031, 28/09/2015
  - **Open source**
  - Génération de png, svg, ...
- i** besoin de graphviz pour le rendu

# PLANTUML - VITE UN EXEMPLE

```
@startuml
interface Conference
Conference *- Talk
BdxIO .> Conference
@enduml
```



# LE "JAVAC"

Sans asciidoctor

```
$ java -jar plantuml.jar monfichier.puml
```

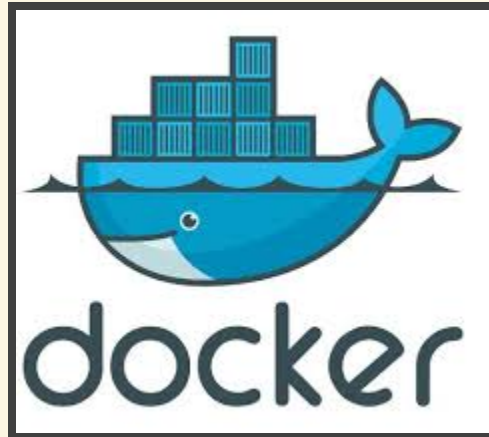
Avec asciidoctor (qui inclue du PlantUml)

```
$ asciidoctor -r asciidoctor-diagram monfichier.adoc
```

**i** la gem asciidoctor-diagram doit être installée



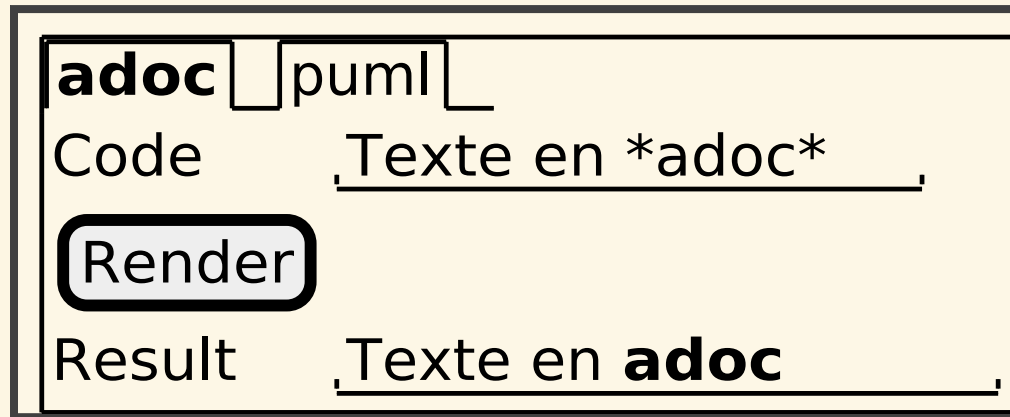
# INSTALLER RUBY? NON $\Rightarrow$ DOCKER!



```
$ docker run --rm -v <VOL> asciidoctor/docker-asciidoctor <CMD>
```

# RÉALISATION D'UNE CONCEPTION UML

Appli Web - "live rendering" ADOC & PUMML



- Front AngularJS
- Back Java - JAX-RS

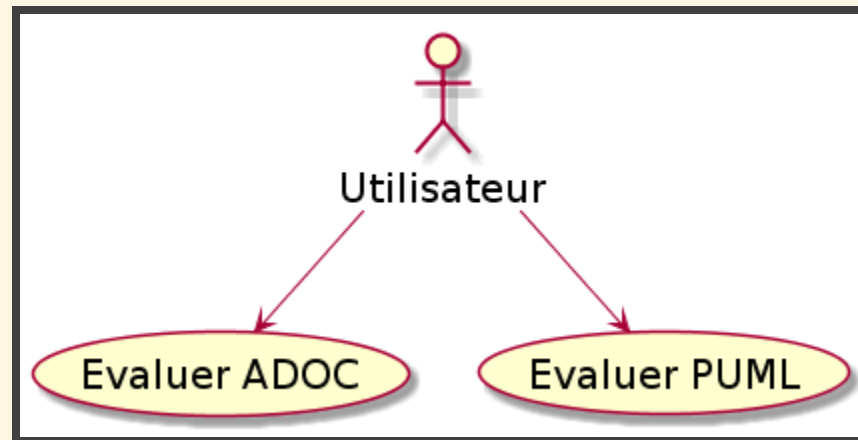
# MA CONCEPTION TECHNIQUE

## Code puml

```
@startuml
:Utilisateur: --> (Evaluer ADOC)
:Utilisateur: --> (Evaluer PUML)
@enduml
```

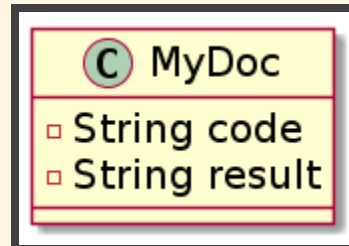
## Inclusion dans un adoc

```
plantuml::puml/Usecase.puml[tmp-Usecase, format="png", align="center"]
```



# MODÈLE DE DONNÉES

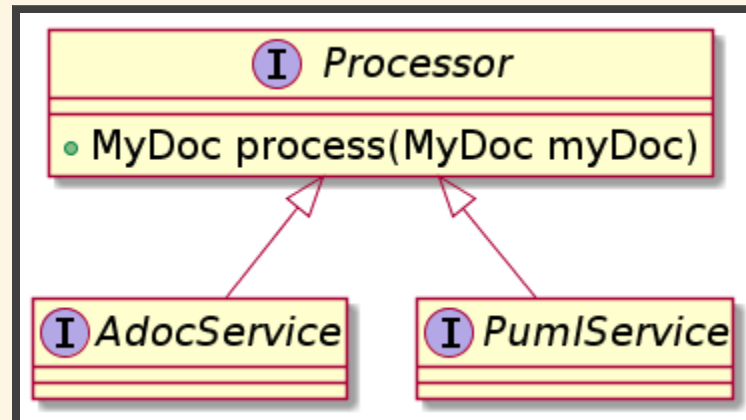
```
@startuml
!include src/main/asciidoc/puml/PumlParams.puml
class MyDoc {
- String code
- String result
}
@enduml
```



# SERVICES

## Code puml

```
@startuml
interface Processor {
  +MyDoc process(MyDoc myDoc)
}
interface AdocService -up-|> Processor
interface PumIService -up-|> Processor
@enduml
```



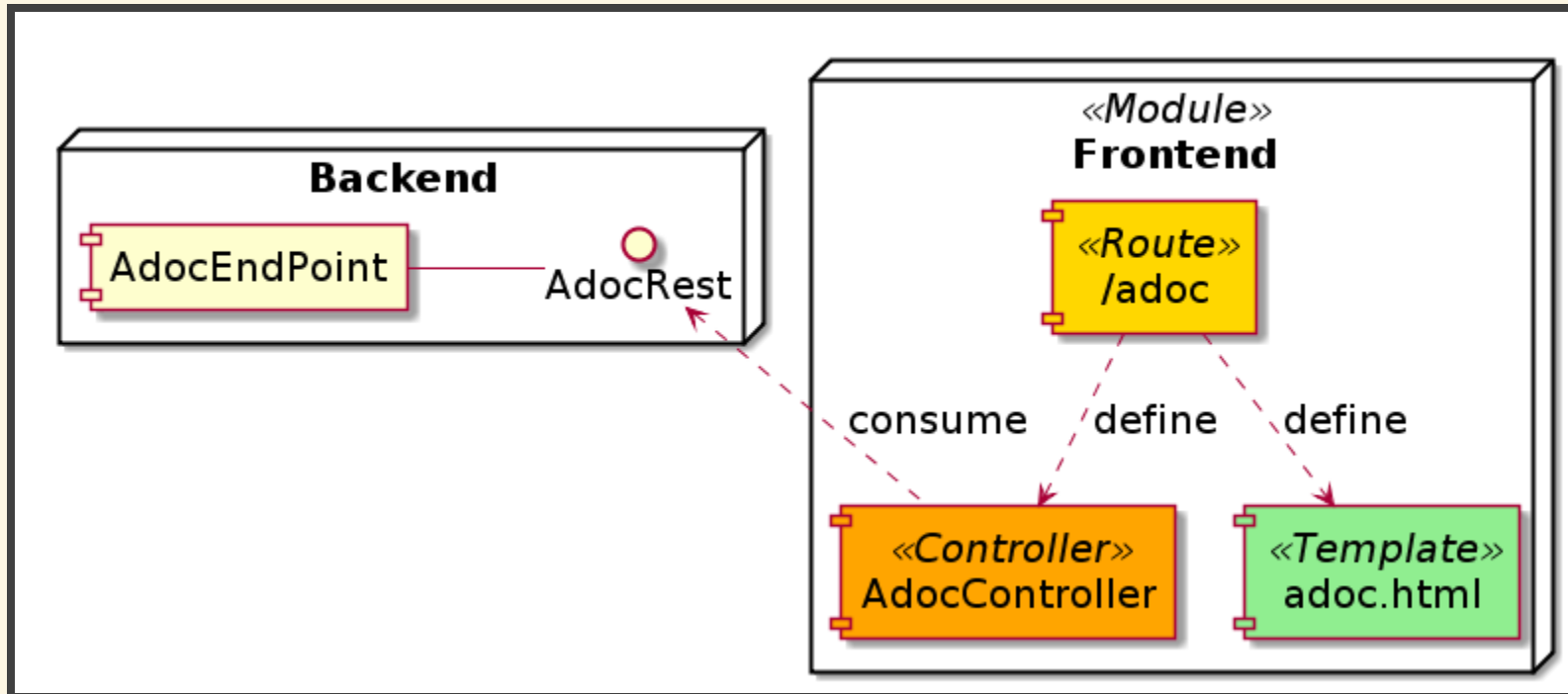
# COMPOSANTS

```
@startuml
node "Frontend" <<Module>> {
  [/adoc] <<Route>> as route_adoc #Gold
  [AdocController] <<Controller>> as contr_adoc #orange
  [adoc.html] <<Template>> as templ_adoc #LightGreen

  route_adoc ..> contr_adoc : define
  route_adoc ..> templ_adoc : define
}

node "Backend" {
  [AdocEndPoint] - AdocRest
}
contr_adoc .up.> AdocRest : consume
@enduml
```

# COMPOSANTS



# SÉQUENCE

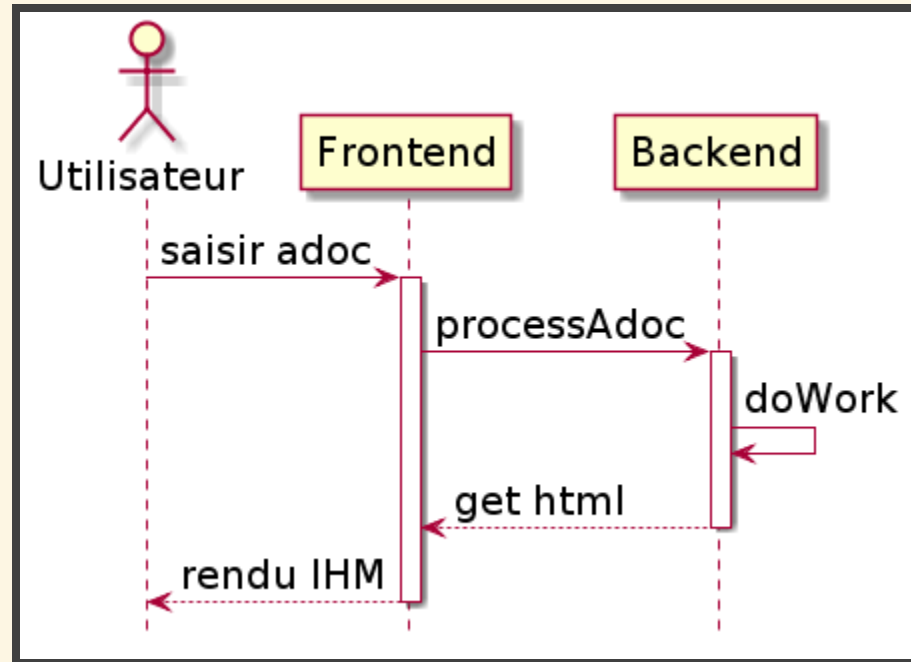
```
@startuml
actor Utilisateur
participant Frontend
participant Backend

Utilisateur -> Frontend : saisir adoc
activate Frontend
Frontend -> Backend : processAdoc
activate Backend
Backend -> Backend : doWork
Backend --> Frontend : get html
deactivate Backend
Frontend -->Utilisateur : rendu IHM
deactivate Frontend

hide footbox
@enduml
```



# SÉQUENCE



# ET ENCORE BEAUCOUP PLUS

<http://fr.plantuml.com>

- ❗ Simplicité & Productivité :)

# WORKFLOW DE GÉNÉRATION

# POUR SOI... "LIVE RENDERING"

- Plugins AsciiDoc & PlantUml
  - IDE : Eclipse, IntelliJ, ...
  - Editeur : Atom, ...
  - Navigateur : Chrome, Firefox, ...
  - Google Doc, Office (pour PlantUml)

# DANS UN PROJET DE DEV

- Maven/Gradle avec JRuby
  - *src/main/asciidoc* ⇒ *target/generated-docs*
  - <https://github.com/asciidoctor/asciidoctor-maven-examples>
  - <https://github.com/asciidoctor/asciidoctor-gradle-examples>
- **asciidoctorjs** via Grunt/Gulp

# ECRIRE UN WIKI EN ASCIIDOC...

- Sources sous Git!!!



- Rendu html???
- apache? asciidoctor-confluence?
- Droit d'accès à la doc?

# SOUS HUBPRESS



Sous Github... Blog :) Doc privée :(

# SOUS {:AWESTRUCT}

- Monde Ruby
- Utilisation d'Asciidoctor
- Conteneur Docker
- Peu familiarisé avec l'outillage Ruby :(



# TEST

```
= Test Adoc et Puml
Mehdi Rebiai <mrebiai@sqli.com>

== Test titre
* Point 1
* Point 2

== Test PlantUml
[plantuml, diagram-classes, png]
....
class BdxIO
class DevoxxFrance
interface Conference

BdxIO ..> Conference
DevoxxFrance ..> Conference
....
```

# SOUS GITLAB

Test Adoc et Puml

Test titre

- Point 1
- Point 2

Test PlantUml

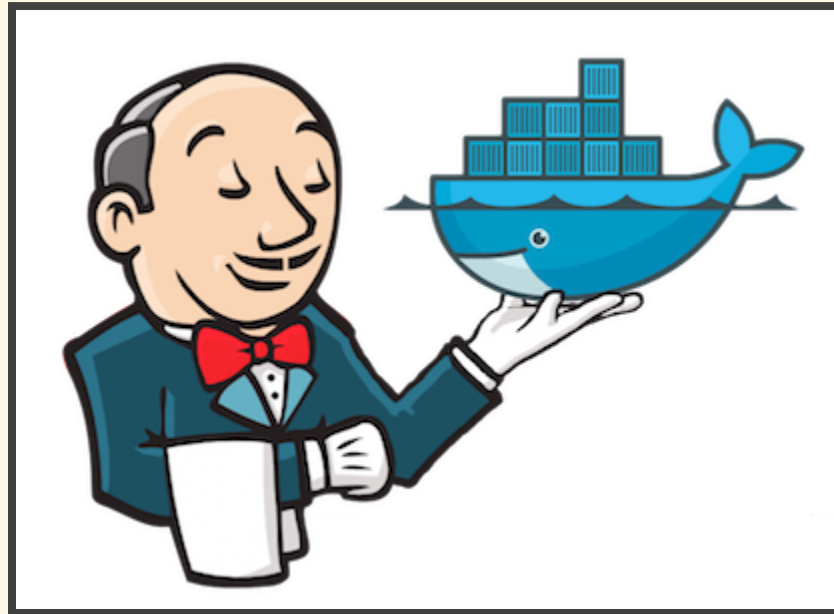
```
class BdxIO
class DevoxxFrance
interface Conference

BdxIO ..> Conference
DevoxxFrance ..> Conference
```

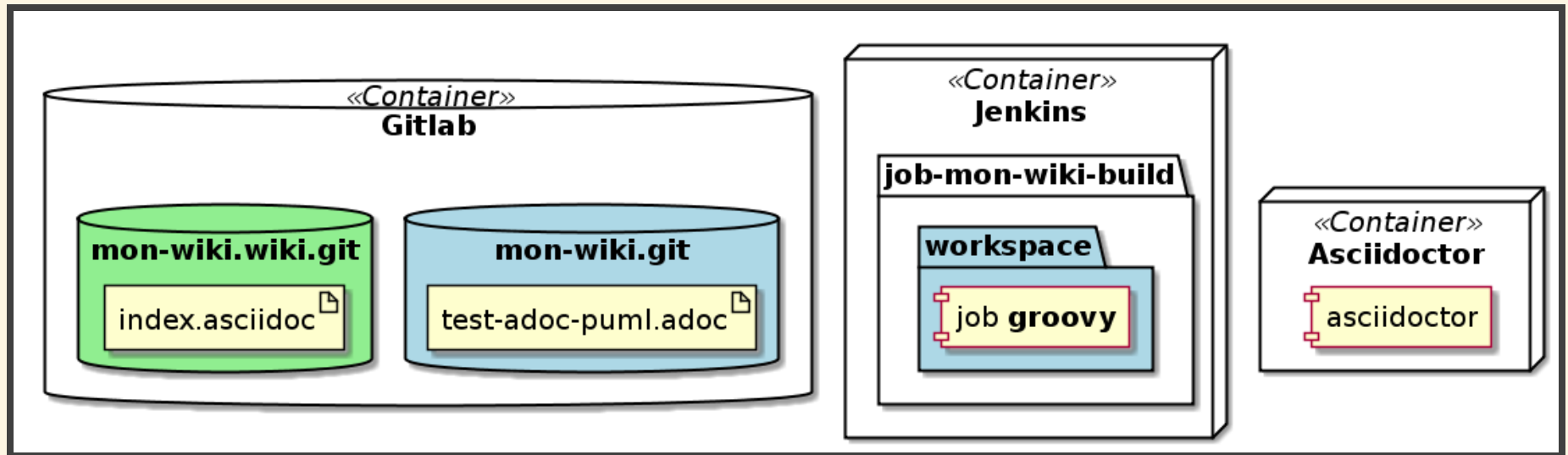
❗ il manque AsciiDoctor!

# WORKFLOW CUSTOM

Gitlab + Jenkins Workflow Plugin + Docker

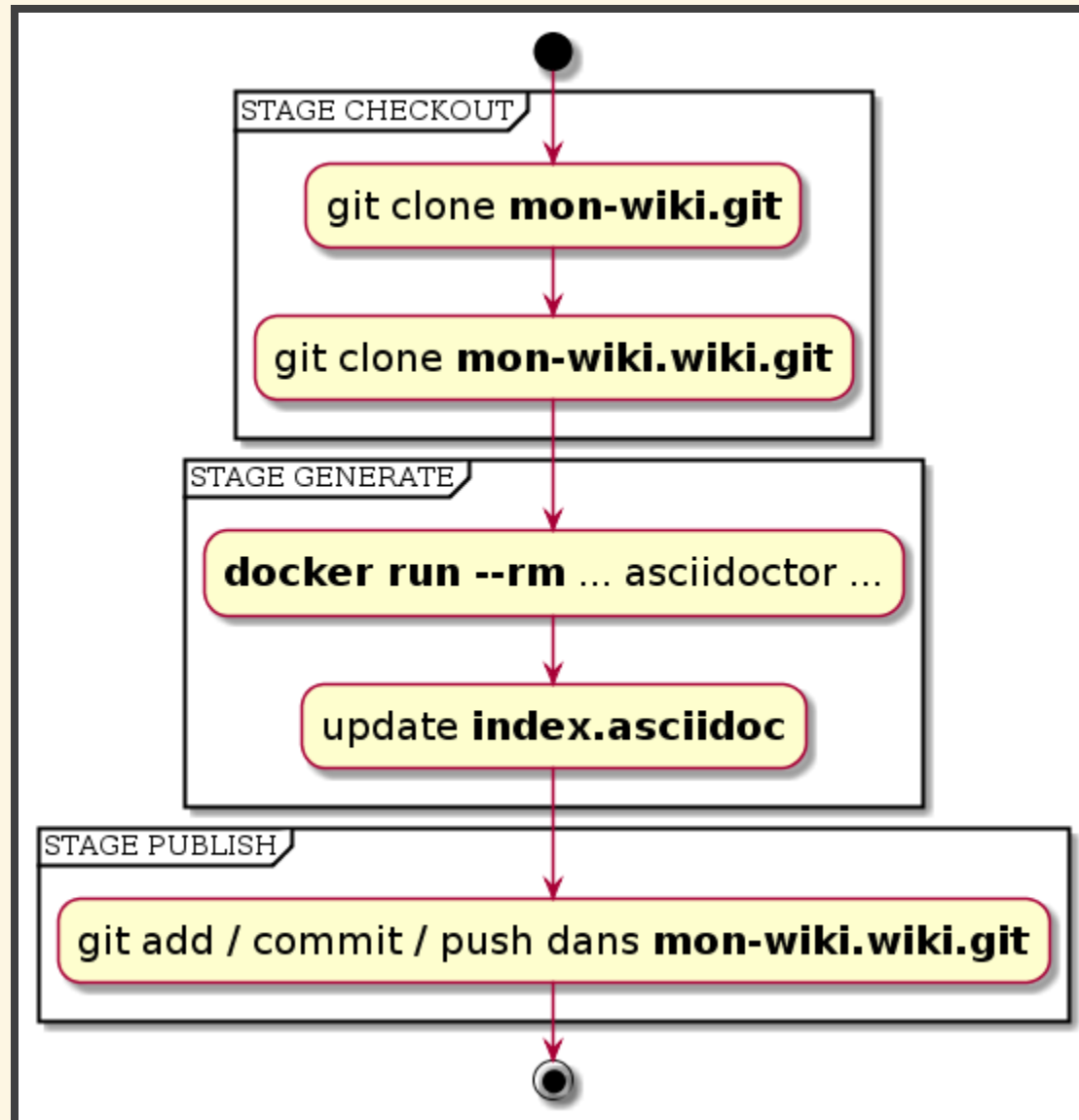


# WORKFLOW CUSTOM



Droits sur `mon-wiki.wiki.git` = Droits sur `mon-wiki.git` :)

# WORKFLOW CUSTOM



# WORKFLOW CUSTOM

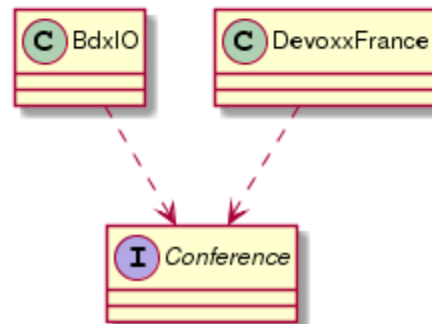
## Test Adoc et Puml

Mehdi Rebiai - [mrebiai@sqli.com](mailto:mrebiai@sqli.com)

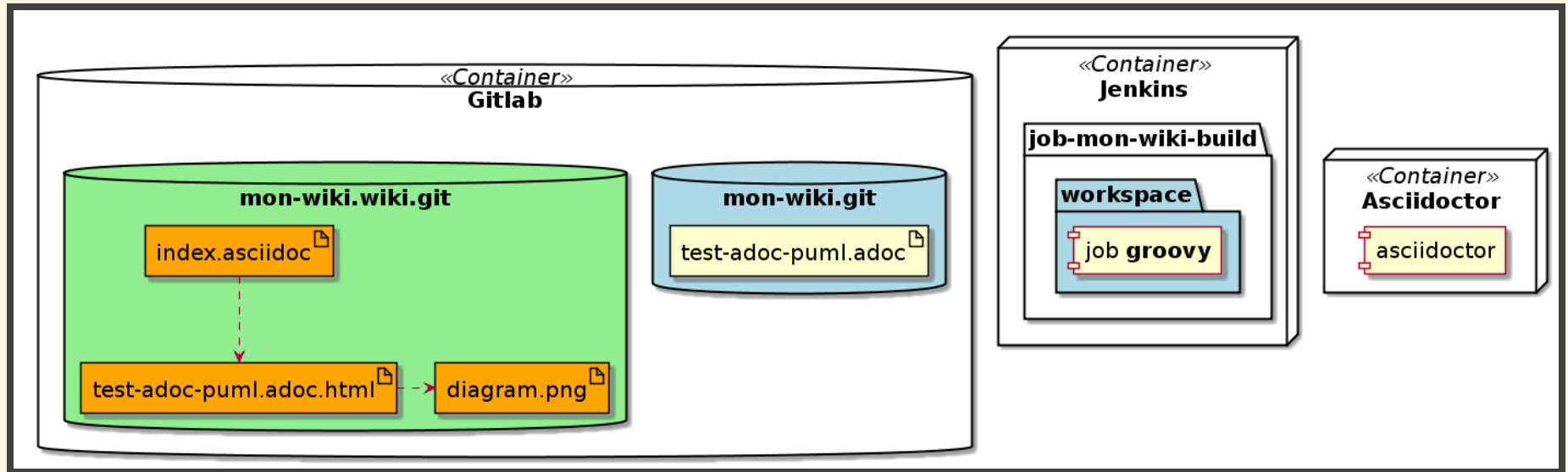
## Test titre

- Point 1
- Point 2

## Test PlantUml



# WORKFLOW CUSTOM



# UNE PRÉSENTATION... COMME CELLE LÀ

- Réalisée en AsciiDoc!
- Backend reveal.js
- Maven plugin
- Diagrammes PlantUml
- Poids des adoc+puml ~ **12 KO**
- <https://github.com/mrebiai/moijecodemadoc>



# VAINCRE C'EST CONVAINCRE

- AsciiDoc ⇒ merci à Benoit Prioux
- Docker ⇒ merci à Christian Chavez
- Bdx.io ⇒ merci à Rémi Goyard
- MERCI A VOUS !

# MOI JE CODE M'ADOC

Questions?

- <https://github.com/mrebiai/moijecodemadoc>
- [mrebiai@sqli.com](mailto:mrebiai@sqli.com)
- [mehdi.rebiai.dev@gmail.com](mailto:mehdi.rebiai.dev@gmail.com)