



Isabelle Delignière

Soutenance du 21/09/2021

GLG204 Architectures Logicielles Java(2)

Pascal Graffion, Serge Rosmorduc

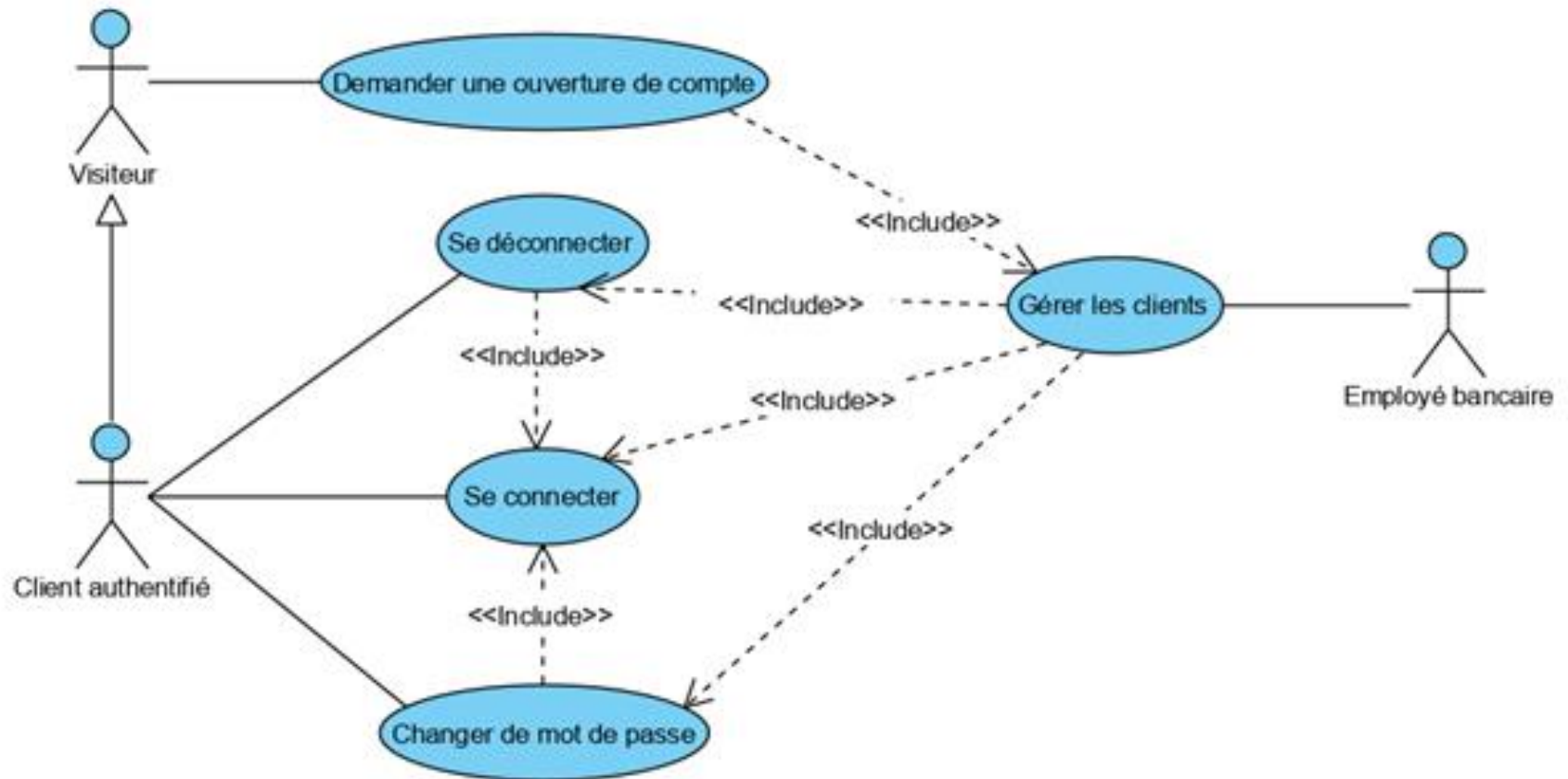
le cnam  
Paris

# Présentation du projet

- Application Web bancaire
- Banque fictive créée pour UE GLG204
- Utilisateurs :
  - Employés bancaires
    - Gestion des clients et des comptes bancaires
  - Clients
    - Gestion des comptes bancaires Mocha Bank et agrégation de comptes externes
- Méthode du projet : **Arrington**

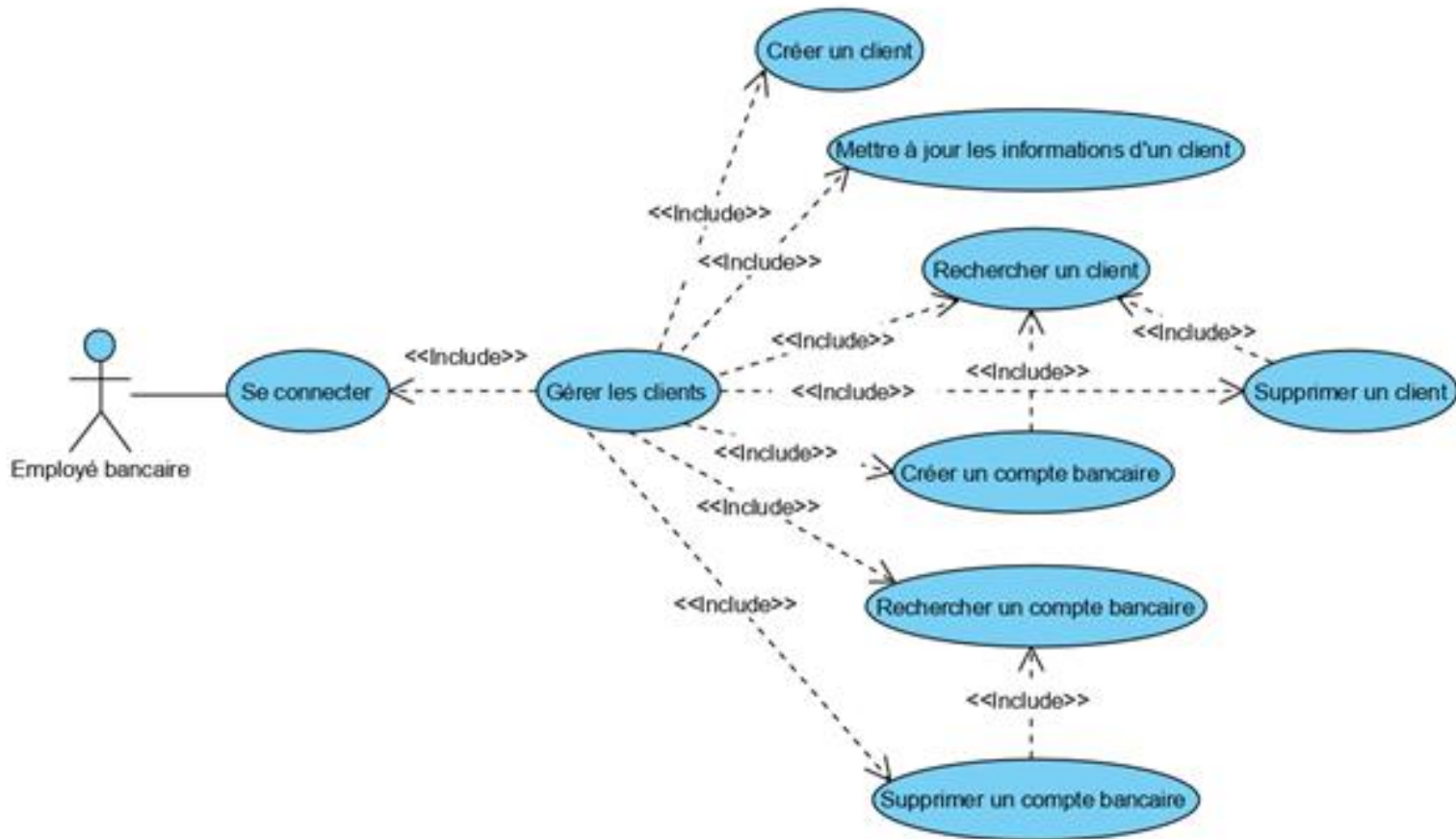
# Analyse

## 1<sup>er</sup> Sous-système « opérations de base »



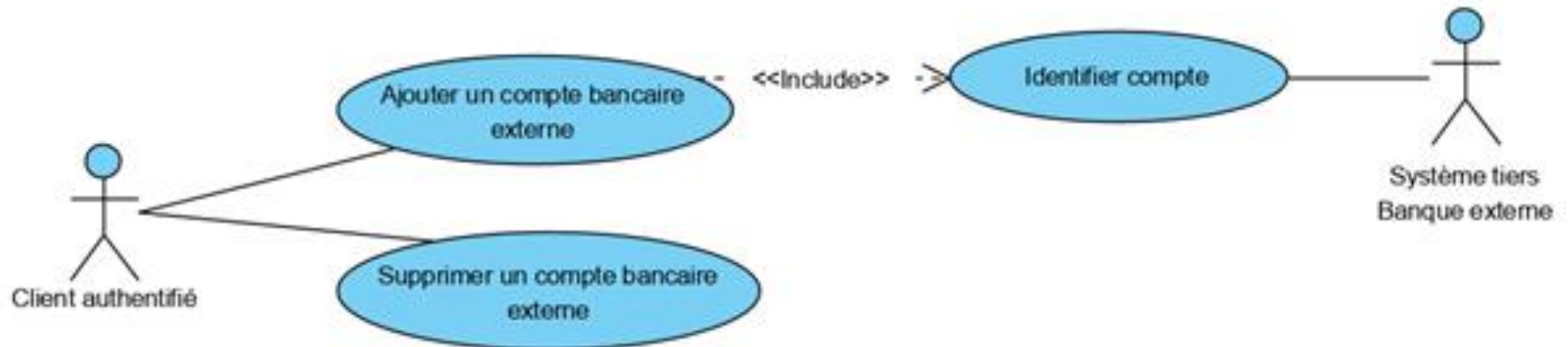
# Analyse

## 2<sup>ème</sup> Sous-système « gérer les clients »



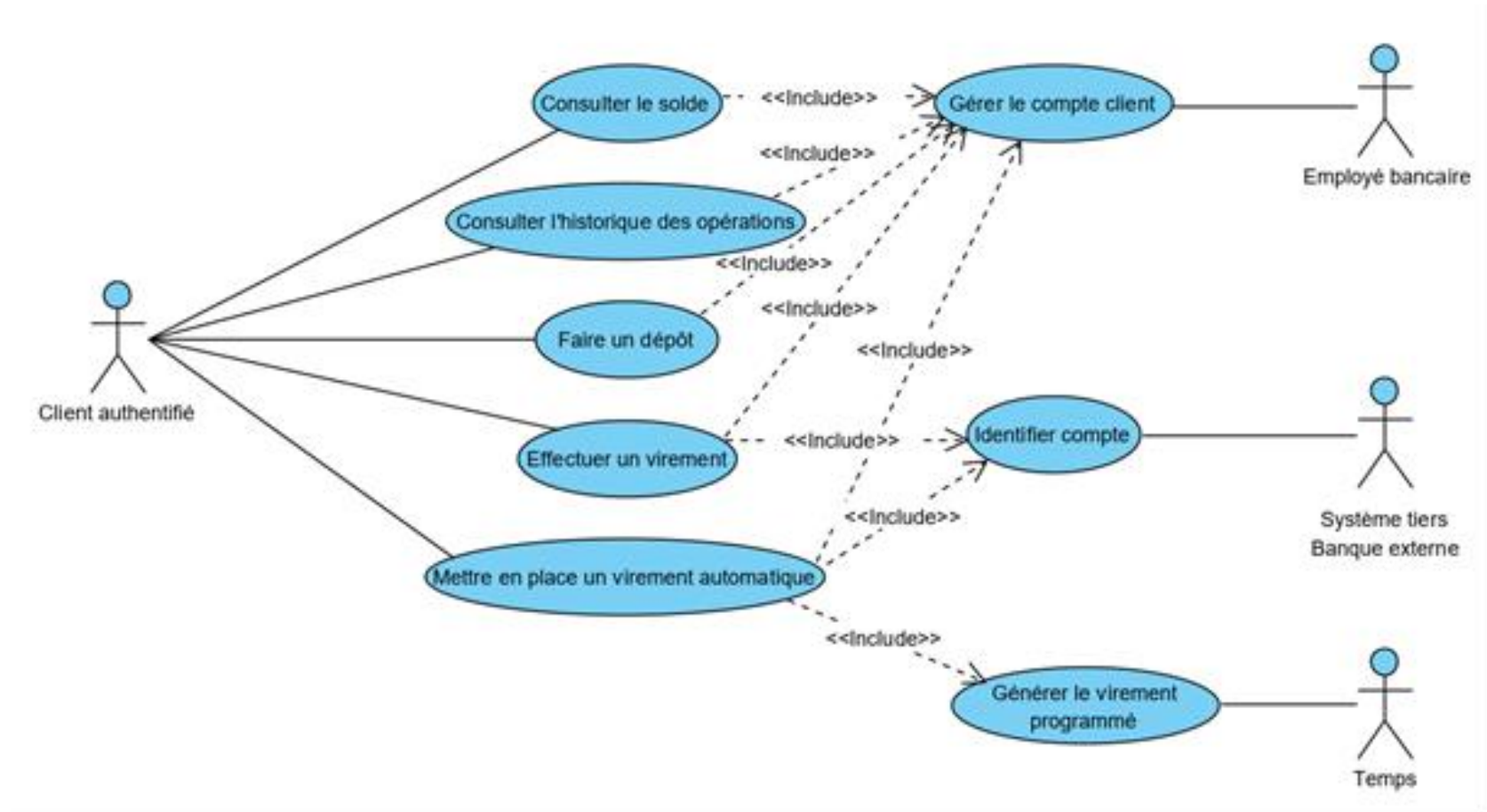
# Analyse

## 3<sup>ème</sup> Sous-système « agréger un compte bancaire »



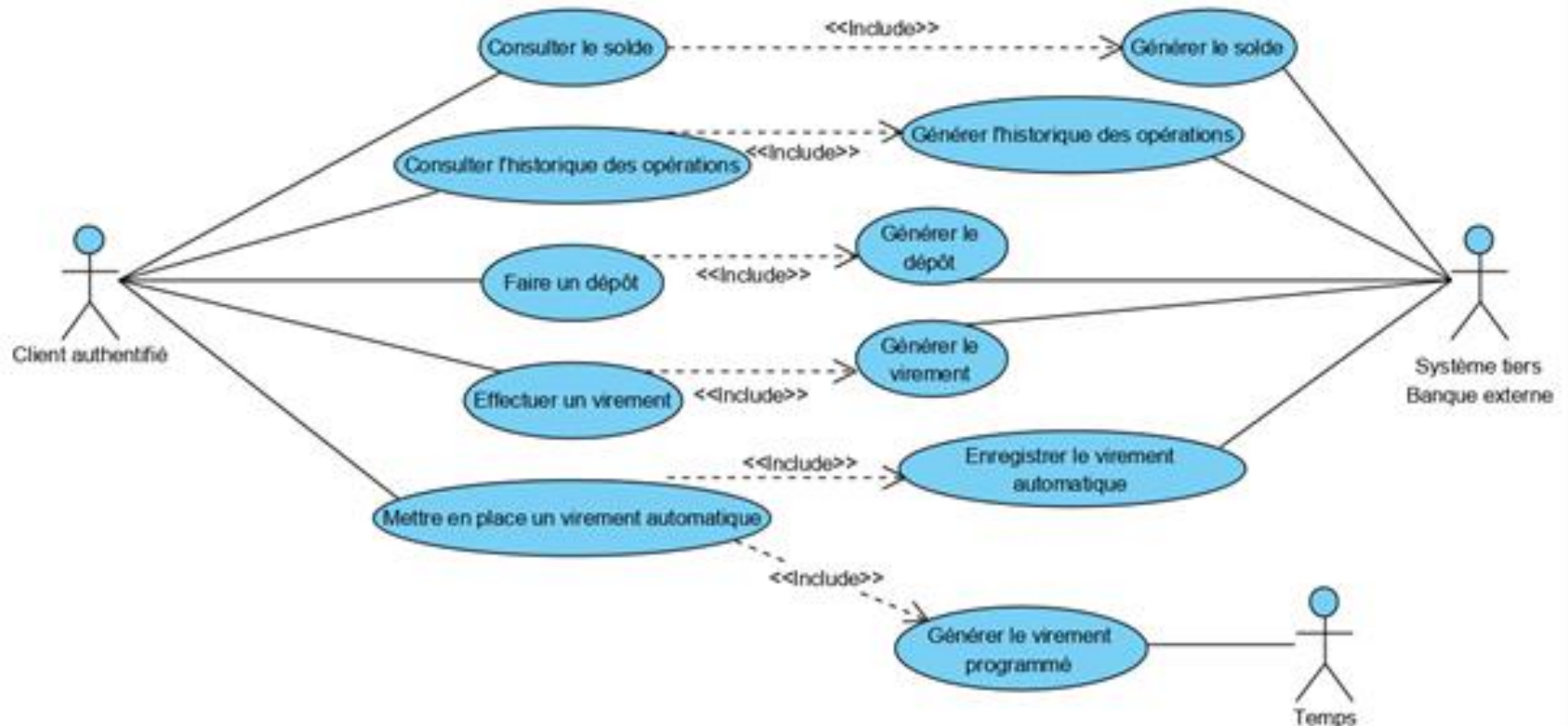
# Analyse

## 4<sup>ème</sup> Sous-système « gérer le compte bancaire » - Cas de la gestion d'un compte bancaire interne



# Analyse

## 5<sup>ème</sup> Sous-système « gérer le compte bancaire » - Cas de la gestion d'un compte bancaire externe agrégé



# Architecture – conception

## Architecture

### ➤ Packages et dépendances :

- `com.anotherbank.mochabank.api`
- `com.anotherbank.mochabank.authentication.domain`  
(api, dao, dto, model, service)
- `com.anotherbank.mochabank.domain`  
(dao, dto, model, service)
- `com.anotherbank.mochabank.configuration`
- `com.anotherbank.mochabank.exception`  
(api)
- `com.anotherbank.mochabank.logging`
- `com.anotherbank.mochabank.task`



# Architecture – conception

## Technologies

- Architecture **REST** (REpresentational State Transfer)
- Framework d'application **SPRING**
- Serveur Web **Apache Tomcat**
- Système de gestion de bases de données relationnelles **MySQL**
- API « Java Persistence » (**JPA**)
- Couche métier : Plain Old Java Object (POJO)
- Couche service : Plain Old Java Object (POJO)
- Couche présentation : **Thymeleaf** et **Vue.js**

# Architecture – conception

## Environnement de développement

- Eclipse
- Spring Tools 4 pour Eclipse
- Gradle 7.1.1
- WampServer 3.1.9 (MySQL 5.7.26 - APACHE 2.4.39)
- MySQL connector 5.1.40
- Tomcat 9
- Node.js 14.17.3 - Vue.js 2.6.14
- Bootstrap 4.6
- JUnit vintage-engine-5.7.2
- HTTPUnit 1.7
- GreenMail

# Architecture – conception

## Classes et interfaces réutilisées des TPs de l'unité de GLG203

- Annotation « **@author** Teachers of GLG203 Unit »
- Package logging
- Package exception
- Classes concernant le rôle des utilisateurs
- Le chiffrement : MyCrypt
- La classe UserDetailsServiceImpl
- Les classes de configuration WebSecurityConfig et DefaultViewConfig

# Architecture – conception

## REST (REpresentational State Transfer)

### ➤ Principes respectés :

- Uniformité de l'interface : utilisation de standards et de normes
- Aucune dépendance entre Client / Serveur
- Stateless
- Architecture système en couches

### ➤ Principe prévu :

- Permettre la mise en cache des données

### ➤ Principe non respecté :

- Code sur demande

# Architecture – conception

## Choix d'implémentation

- Simplification actuelle :
  - Un compte par client - Un client par compte
- Format des dates : YYYY/MM/DD (MySQL)
- Mise en place des virements automatiques :
  - Expression Cron Java/Spring
  - Annotation « @EnableScheduling »
  - Classe RunnableTask implémente l'interface Runnable
- Virement externe :
  - Classe HTTPSender
  - Application « Other Bank »

# Architecture – conception

## Communication banque externe



- Clone de Mocha Bank
- Port : 8181
- Acteur « Système tiers Banque externe »
- Permet d'effectuer et de recevoir des virements externes à la banque Mocha Bank

# Architecture – conception

## Sécurité

### ➤ Implémentés :

- Authentification : Spring
- Rôles
- Codage de mot de passe : BCrypt

### ➤ Non implémenté actuellement :

- Authentification forte (minimum deux facteurs) imposée par la directive européenne relative aux services de paiement (DSP2)

# Évolutions

- Correction des erreurs d'implémentation
- Exécution des nouveaux virements automatiques programmés sans restart de l'application
- Implémentation des fonctions manquantes
- Amélioration des tests
- Amélioration de l'architecture REST
- Travail sur la sécurité (ajout d'un code fixe par ex)



# Évolutions

- Travail sur la validation (adresse mail, téléphone...)
- Affinage des offres et types de compte bancaires
- Prise en compte des dates de valeurs et frais bancaires
- Ajout d'alertes selon évènements
- Implémentation du pattern money ?
- Implémentation du pattern de messagerie ?
- Etc...