

ProxiBanque

Une Banque proche de ses Clients

Matthieu CHORRO

Formation et Expérience Professionnelle

- ▶ Formation Universitaire
- ▶ Doctorat de Physique
- ▶ 8 ans d'expérience dans la recherche fondamentale

Reconversion Professionnelle

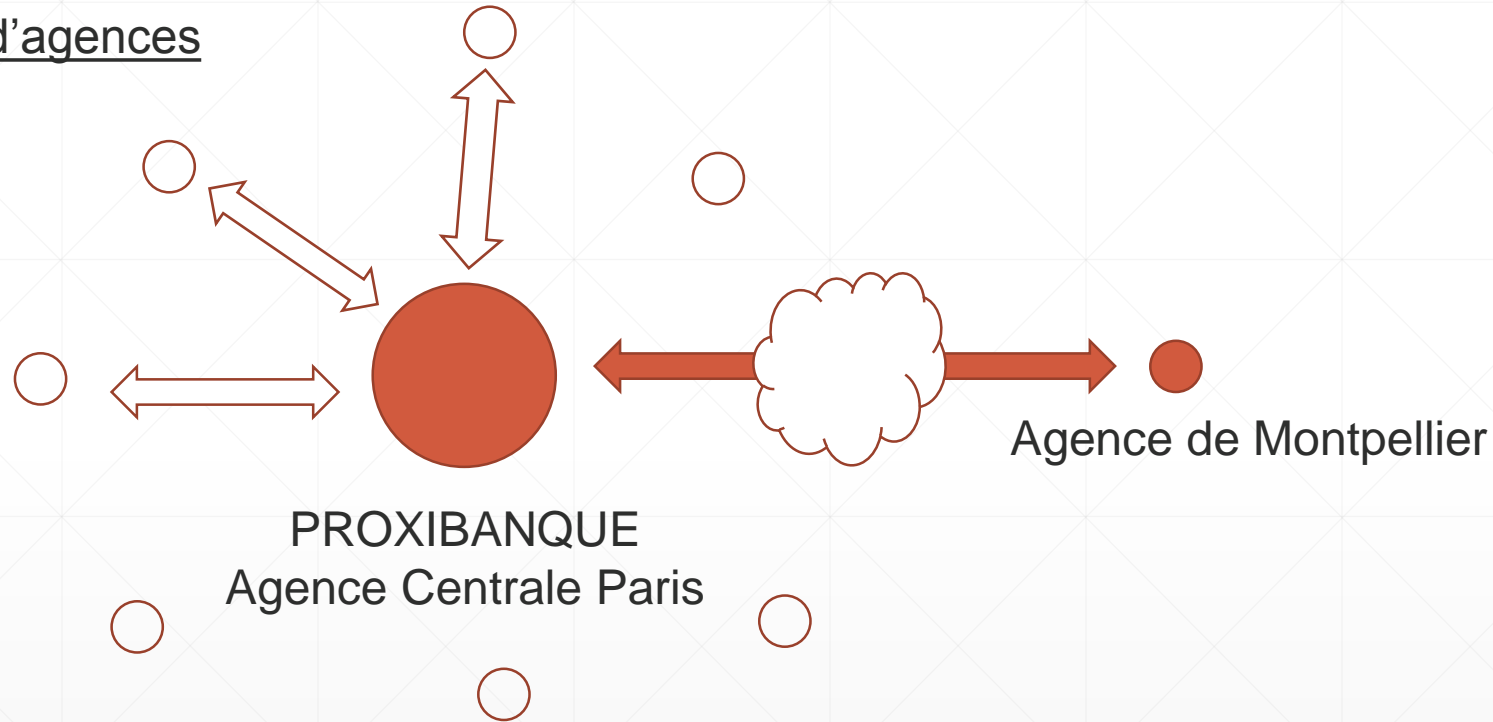
- ▶ Difficulté à trouver un poste fixe en Science
- ▶ Volonté de changer de carrière
- ▶ Intérêt pour le codage informatique
- ▶ Formation GTM : Concepteur Développeur en informatique

Présentation de ProxiBanque

- ProxiBanque est un réseau d'agences bancaires françaises
 - Proxibanque a besoin d'un outil de gestion commerciale à destination de ses conseillers et de ses directeurs d'agence
 - Les services attendus par ProxiBanque sont notamment :
 - Pour un conseiller :
 - Consulter la liste de ses clients, et la modifier
 - Consulter la liste des comptes d'un client
 - Effectuer un virement de compte à compte
 - Pour un directeur d'agence :
 - Consulter la liste des conseillers de l'agence
 - Avoir les mêmes possibilités qu'un conseiller
-

Livraison: Déploiement dans une agence locale

Réseau d'agences



V2 : extension aux N agences de PROXIBANQUE

Gestion des listes : la demande et le contexte

- La demande:

- Pour un conseiller :

- Consulter la liste de ses clients, et la modifier

- Consulter la liste des comptes d'un client

- Pour un directeur d'agence :

- Consulter la liste des conseillers de l'agence

- Avoir les mêmes possibilités qu'un conseiller

- Contexte

- Demande urgente du client

- Délais de réalisation limitée

- Développer vite et bien

Gestion des listes : La liste des conseillers

PROXIBANQUE

Vous êtes connecté en tant que Untel

Liste des clients

Nom	Prénom	Adresse	Email	Code postal	Ville	Comptes	Modification	Suppression
CHORROT	MATTHIEU	21 rue de rueil	chorro@gtm.com	34000	Montpellier	Comptes	Modifier	Supprimer
CARIGNY	NATHALIE	10 rue des Antilles	carigny@gtm.com	34000	Montpellier	Comptes	Modifier	Supprimer
HILGER	JEROME	21 rue de la chute libre	hilger@gtm.com	34000	Montpellier	Comptes	Modifier	Supprimer
FRANTZ	RUDY	10 rue de la joebarteam	frantz@gtm.com	34000	Montpellier	Comptes	Modifier	Supprimer
LOIZILLON	FRANCK	35 rue du magicien	loizillon@gtm.fr	34000	Montpellier	Comptes	Modifier	Supprimer
PESCHER	AMELIE	10 rue du cinema	pescher@gtm.fr	34000	Montpellier	Comptes	Modifier	Supprimer

Gestion des listes: La liste des clients

PROXIBANQUE

Vous êtes connecté en tant que Dupond

Liste des clients

Nom	Prénom	Adresse	Email	Code postal	Ville	Comptes	Modification	Suppression
CHORROT	MATTHIEU	21 rue de rueil	chorro@gtm.com	34000	Montpellier	<div>Comptes</div>	<div>Modifier</div>	<div>Supprimer</div>
CARIGNY	NATHALIE	10 rue des Antilles	carigny@gtm.com	34000	Montpellier	<div>Comptes</div>	<div>Modifier</div>	<div>Supprimer</div>
HILGER	JEROME	21 rue de la chute libre	hilger@gtm.com	34000	Montpellier	<div>Comptes</div>	<div>Modifier</div>	<div>Supprimer</div>
FRANTZ	RUDY	10 rue de la joebarteam	frantz@gtm.com	34000	Montpellier	<div>Comptes</div>	<div>Modifier</div>	<div>Supprimer</div>
LOIZILLON	FRANCK	35 rue du magicien	loizillon@gtm.fr	34000	Montpellier	<div>Comptes</div>	<div>Modifier</div>	<div>Supprimer</div>
PESCHER	AMELIE	10 rue du cinema	pescher@gtm.fr	34000	Montpellier	<div>Comptes</div>	<div>Modifier</div>	<div>Supprimer</div>

Gestion des listes: La liste des comptes

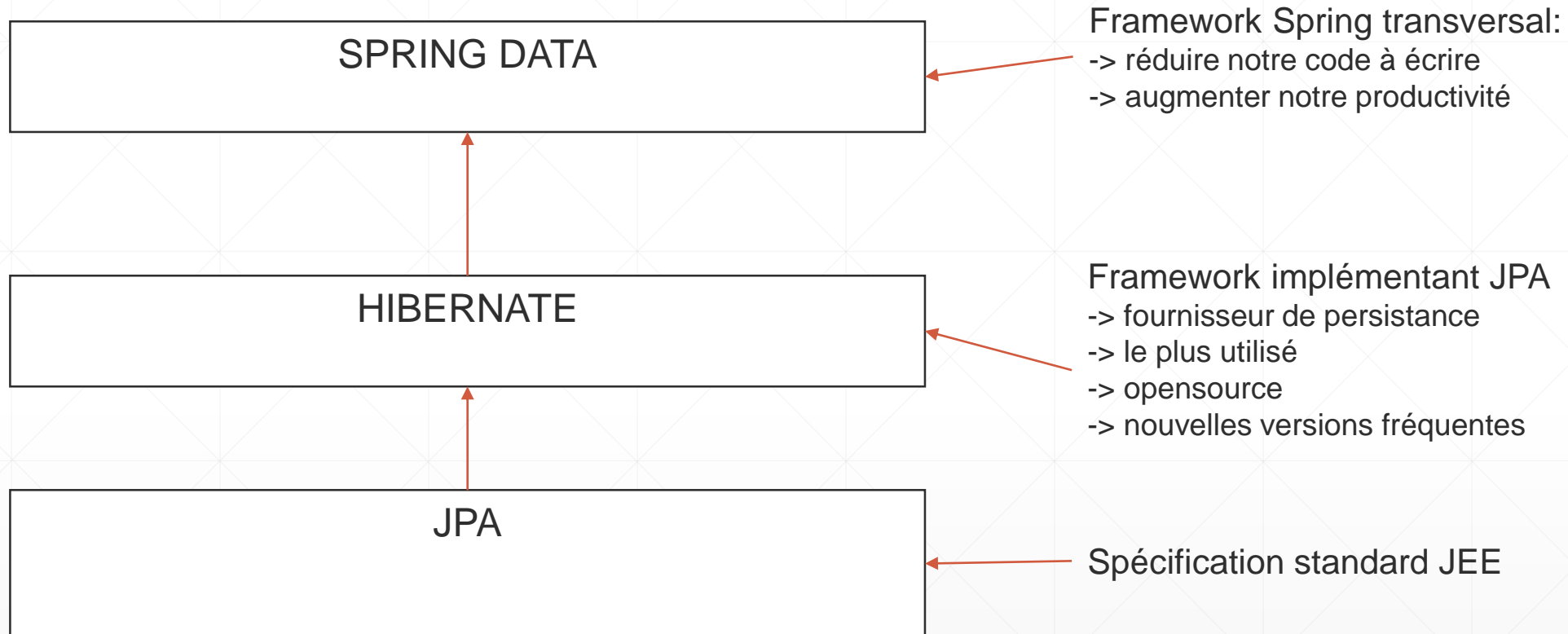
PROXIBANQUE

Liste des comptes du client

Titulaire du compte	Numero Compte	Solde
PESCHER AMELIE	12001205	10000.0
PESCHER AMELIE	22002205	10000.0

retour

Gestion des listes : Solutions techniques



Gestion des listes : La couche domaine

```
@Entity
@Getter
@Setter
@ToString
@Inheritance(strategy=InheritanceType.JOINED)
public class Personne {
```

Classe personne

```
SELECT * FROM `personne`
```

Table personne

```
@Id
@GeneratedValue(strategy=GenerationType.AUTO)
private int id;
```

```
private String nom;
private String prenom;
private String email;
private String adresse;
private String ville;
private int codePostal;
```

```
// constructeur
```

```
public Personne() {
    super();
}
```

Propriétés de la classe personne

```
public Personne(String nom, String prenom, String email,
    String adresse, String ville, int codePostal) {
    super();
    this.nom = nom;
    this.prenom = prenom;
    this.email = email;
    this.adresse = adresse;
    this.ville = ville;
    this.codePostal = codePostal;
}
```

Propriétés de la table personne

Nombre de lignes : 25

Trier sur l'index: Aucune

+ Options

← T →						id	adresse	codePostal	email	nom	prenom	ville	
<input type="checkbox"/>		Modifier		Copier		Effacer	1	12 par la	75000	untel@proxibanque.fr	Untel	Didier	Paris
<input type="checkbox"/>		Modifier		Copier		Effacer	2	45 rue de l impasse	75000	martin@proxibanque	Martin	jean	Paris
<input type="checkbox"/>		Modifier		Copier		Effacer	3	21 rue de rueil	92200	chorro@gtm.com	CHORROT	MATTHIEU	rueil
<input type="checkbox"/>		Modifier		Copier		Effacer	4	10 rue des Antilles	75000	carigny@gtm.com	CARIGNY	NATHALIE	Paris
<input type="checkbox"/>		Modifier		Copier		Effacer	5	45 rue de l impasse	75000	dupond@proxibanque	Dupond	pierre	Paris
<input type="checkbox"/>		Modifier		Copier		Effacer	6	21 rue de la chute libre	78200	hilger@gtm.com	HILGER	JEROME	Vaucresson
<input type="checkbox"/>		Modifier		Copier		Effacer	7	10 rue de la jobarteam	78000	frantz@gtm.com	FRANTZ	RUDY	Versailles
<input type="checkbox"/>		Modifier		Copier		Effacer	8	22 avenue de palavas	34000	girac@proxibanque	Girac	Kendji	Montpellier
<input type="checkbox"/>		Modifier		Copier		Effacer	9	35 rue du magicien	75000	loizillon@gtm.fr	LOIZILLON	FRANCK	Paris
<input type="checkbox"/>		Modifier		Copier		Effacer	10	10 rue du cinema	75000	pescher@gtm.fr	PESCHER	AMELIE	Paris



☐ Tout cocher

Pour la sélection :



Modifier



Effacer



Exporter

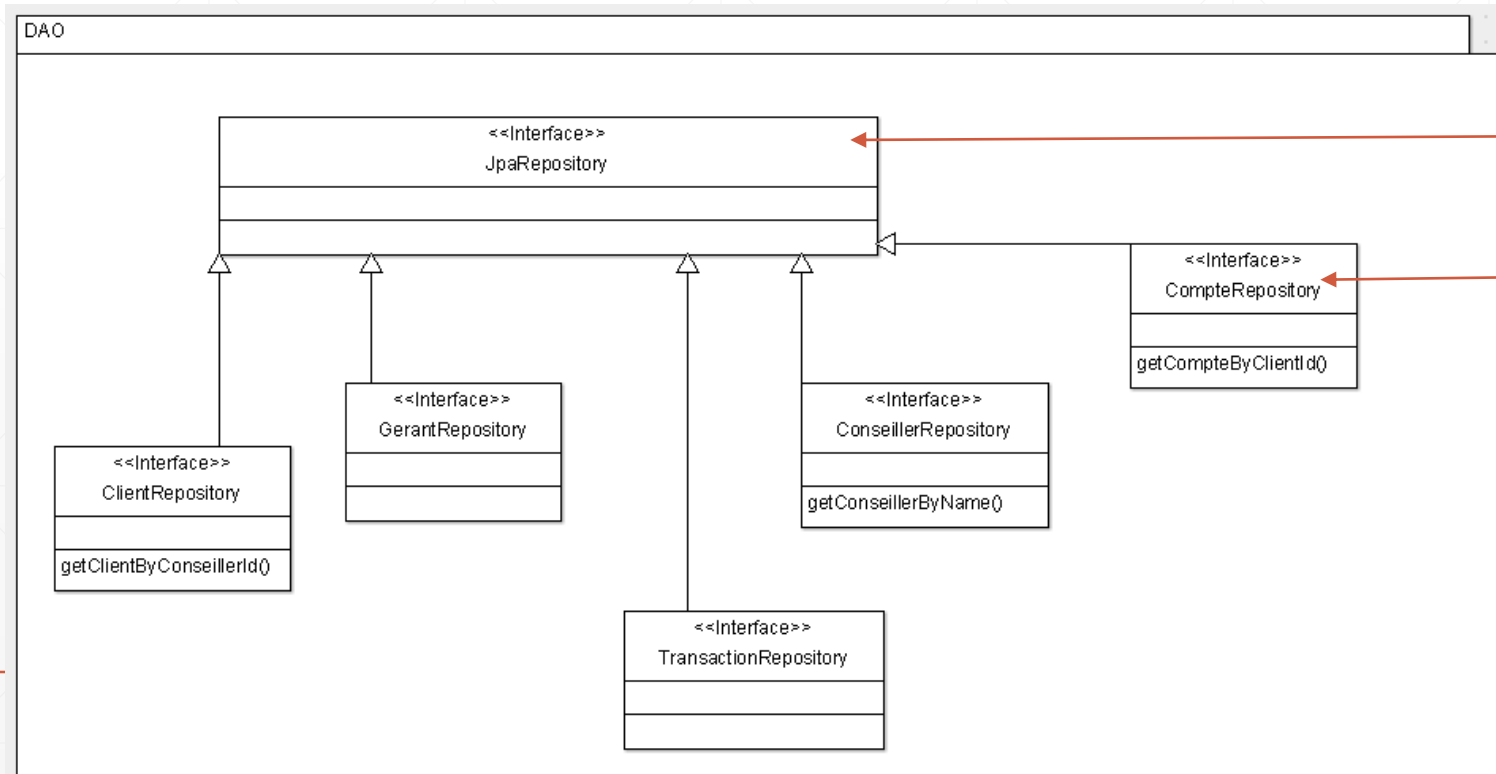
Gestion des listes: La couche de persistance

```
public interface ConseillerRepository extends JpaRepository<Conseiller, Integer>{  
  
    @Query("Select c from Conseiller c where c.nom= ?1")  
    Conseiller getConseillerByNom(String nom);  
  
}
```

Interface fournie par Spring:

- findAll()
- delete()
- Save()

Requête customisée



Interface fournie par Spring

Interface héritant de JpaRepository

Gestion des listes: la couche service

```
package fr.gtm.proxibanque.service;

import java.util.List;

@Service
public class ConseillerService {

    @Autowired
    private ClientRepository clientRepository;

    @Autowired
    private ConseillerRepository conseillerRepository;

    public Conseiller getConseillerByNom(String nom) {

        return conseillerRepository.getConseillerByNom(nom);
    }

    public List<Conseiller> findAllConseiller() {

        List<Conseiller> liste = conseillerRepository.findAll();

        return liste;
    }
}
```

Méthode de l'interface CompteRepository
Avec une requête customisée



Méthode de l'interface CompteRepository
héritée de l'interface JpaRepository



Gestion des listes: La couche présentation

Le bean managé

```
package fr.gtm.proxibanque.controller;

import java.io.Serializable;

@ManagedBean
@RequestScoped
@GET
@POST
public class ConseillerListeBean implements Serializable {

    @ManagedProperty("#{conseillerService}")
    private ConseillerService conseillerService;

    private static final long serialVersionUID = 1L;

    List<Conseiller> listeConseiller = new ArrayList<Conseiller>();

    public void getAll() {
        this.listeConseiller = conseillerService.findAllConseiller();
    }
}
```

La page.xhtml

```
<h:form>
    <p:panel header="Liste des conseillers" style="width: 700;" />

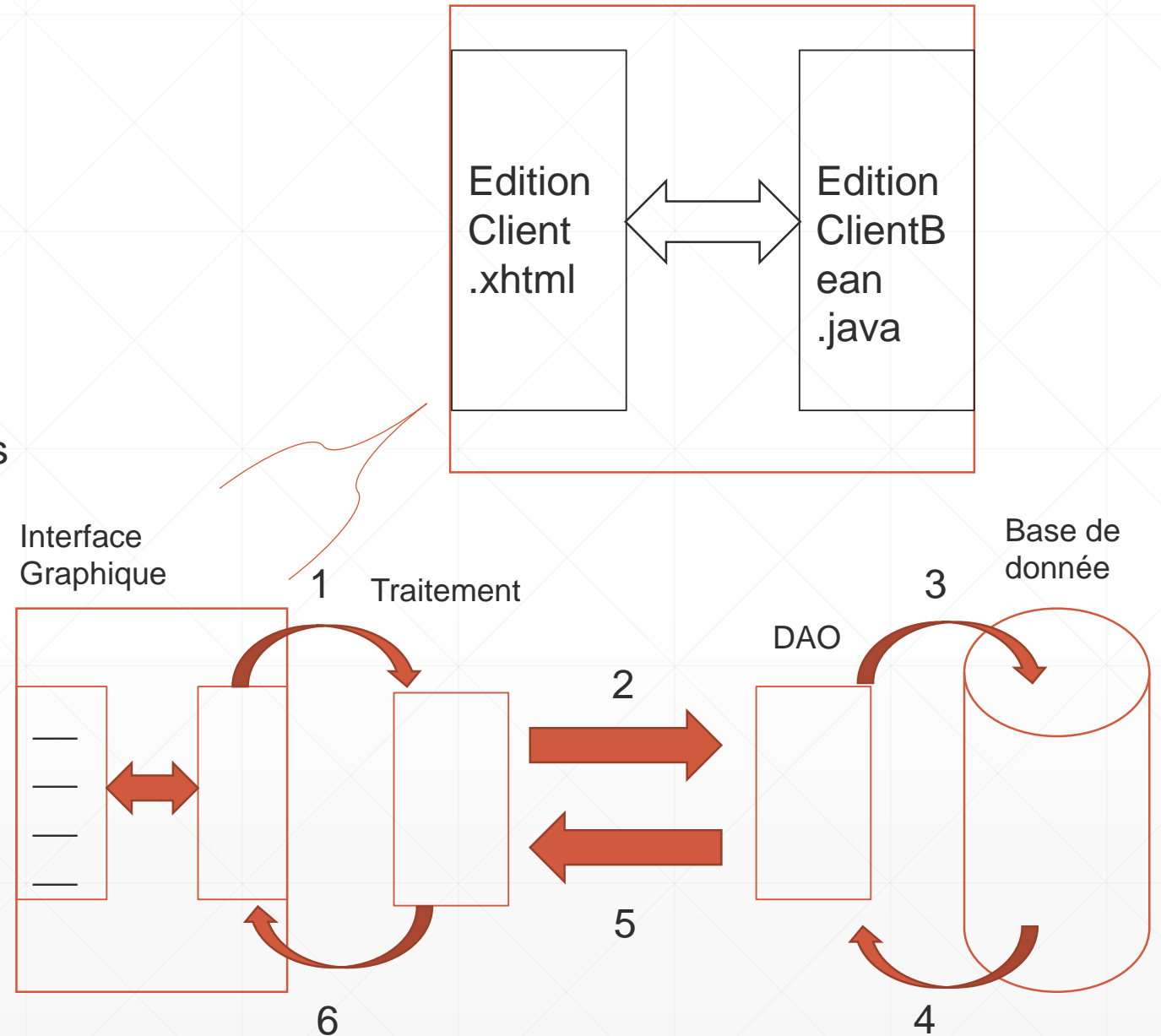
    <p:dataTable value="#{conseillerListeBean.listeConseiller}" var="o"
        id="liste" styleClass="order-table"
        headerClass="order-table-header"
        rowClasses="order-table-odd-row,order-table-even-row">
        <p:column headerText="Nom">#{o.nom}</p:column>
        <p:column headerText="Prénom">#{o.prenom}</p:column>
        <p:column headerText="Adresse">#{o.adresse}</p:column>
        <p:column headerText="Email">#{o.email}</p:column>
        <p:column headerText="Code postal">#{o.codePostal}</p:column>
        <p:column headerText="Ville">#{o.ville}</p:column>
    </p:dataTable>
</h:form>
<h:form>
    <p:commandButton id="listeclient" value="liste des clients"
        action="liste_client" />
</h:form>
```

Fonctionnalité : Edition client

- Demande :
 - ProxiBanque souhaite que ses conseillers et/ou son gérant puissent modifier les informations personnelles des clients
 - Point de vigilance :
 - Vérifier la validité des données avant l'enregistrement en base de donnée
 - Contrainte :
 - Proposer un écran agréable à l'utilisation
-

Solution technique

- JavaServer Faces (JSF)
 - La validation et conversion des champs
 - Industrialisation des pages
 - Les Managed-Beans
- Composant additionnels
 - PrimeFaces



-> Procédure d'édition d'un client

Liste des clients

Nom	Prénom	Adresse	Email	Code postal	Ville	Comptes	Modification	Suppression
FRANTZ	RUDY	10 rue de la jobarteam	frantz@gtm.com	78004	Versailles	<button>Comptes</button>	<button>Modifier</button>	<button>Supprimer</button>
CHORROT	MATTHIEU	21 rue de rueil	chorro@gtm.com	92200	rueil	<button>Comptes</button>	<button>Modifier</button>	<button>Supprimer</button>
CARIGNY	NATHALIE	10 rue des Antilles	carigny@gtm.com	75000	Paris	<button>Comptes</button>	<button>Modifier</button>	<button>Supprimer</button>
HILGER	JEROME	21 rue de la chute libre	hilger@gtm.com	78200	Vaucresson	<button>Comptes</button>	<button>Modifier</button>	<button>Supprimer</button>
LOIZILLON	FRANCK	35 rue du magicien	loizillon@gtm.fr	75000	Paris	<button>Comptes</button>	<button>Modifier</button>	<button>Supprimer</button>
PESCHER	AMELIE	10 rue du cinema	pescher@gtm.fr	75000	Paris	<button>Comptes</button>	<button>Modifier</button>	<button>Supprimer</button>

Effectuer un virement

edition client

Prénom:

RUDY

Nom:

FRANTZ

Adresse:

10 rue de la jobartean

Code postal:

78004

Ville:

Versailles

Email:

frantz@gtm.com

Modifier

-> Les données du client que l'on souhaite modifier

edition client

Prénom:

RUDI

Nom:

FRANTZ

Adresse:

10 rue de la jobartean

Code postal:

75000

Ville:

PARIS

Email:

frantz@gtm.com

Modifier

->Les nouvelles donn  :

Pr  nom : Rudi
Code Postal : 75000
Ville : Paris

Liste des clients								
Nom	Prénom	Adresse	Email	Code postal	Ville	Comptes	Modification	Suppression
FRANTZ	RUDI	10 rue de la jobarteam	frantz@gtm.com	75000	PARIS	Comptes	Modifier	Supprimer
CHORROT	MATTHIEU	21 rue de rueil	chorro@gtm.com	92200	rueil	Comptes	Modifier	Supprimer
CARIGNY	NATHALIE	10 rue des Antilles	carigny@gtm.com	75000	Paris	Comptes	Modifier	Supprimer
HILGER	JEROME	21 rue de la chute libre	hilger@gtm.com	78200	Vaucresson	Comptes	Modifier	Supprimer
LOIZILLON	FRANCK	35 rue du magicien	loizillon@gtm.fr	75000	Paris	Comptes	Modifier	Supprimer
PESCHER	AMELIE	10 rue du cinema	pescher@gtm.fr	75000	Paris	Comptes	Modifier	Supprimer
Effectuer un virement								

Mais si les champs du formulaire d'édition ne sont pas correct
 -> JSF intervient

edition client

Prénom:	<input type="text"/>	formDataTable:prenom : erreur de validation. Vous devez indiquer une valeur.
Nom:	<input type="text" value="F"/>	formDataTable:nom : erreur de validation. La longueur est inférieure à la valeur minimale autorisée, "2".
Adresse:	<input type="text" value="10 rue de la jobartean"/>	formDataTable:Adresse : erreur de validation. La longueur est supérieure à la valeur maximale autorisée, "30".
Code postal:	<input type="text"/>	formDataTable:codepostale : erreur de validation. Vous devez indiquer une valeur.
Ville:	<input type="text" value="PARIS"/>	
Email:	<input type="text" value="frantzgtm.com"/>	Discordance du modèle d'expression régulière « [^] [_a-z0-9-]+(\\.[_a-z0-9-]+)*@[a-z0-9-]+(\\.[a-z0-9-]+)*(\\.[a-z]{2,6})\$».
<input type="button" value="Modifier"/>		

```

<h:outputLabel value="Prénom:" />
<p:inputText value="#{editionClientBean.prenom}" required="true"
  id="prenom">
  <f:validateLength minimum="2" maximum="30" />
</p:inputText>
<p:message for="prenom" display="text" />

<h:outputLabel value="Nom:" />
<p:inputText value="#{editionClientBean.nom}" required="true"
  id="nom">
  <f:validateLength minimum="2" maximum="30" />
</p:inputText>
<p:message for="nom" style="color:red" display="text" />

<h:outputLabel value="Adresse:" />
<p:inputText value="#{editionClientBean.adresse}" required="true"
  id="Adresse">
  <f:validateLength minimum="2" maximum="30" />
</p:inputText>
<p:message for="Adresse" style="color:red" display="text" />

```

```

<h:outputLabel value="Code postal:" />
<p:inputMask value="#{editionClientBean.codePostal}"
  required="true" id="codepostale" mask="99999">
</p:inputMask>
<p:message for="codepostale" style="color:red" display="text" />


<h:outputLabel value="Ville:" />
<p:inputText value="#{editionClientBean.ville}" required="true"
  id="ville">
  <f:validateLength minimum="2" maximum="30" />
</p:inputText>
<p:message for="ville" style="color:red" display="text" />

<h:outputLabel value="Email:" />
<p:inputText value="#{editionClientBean.email}" required="true"
  id="email">
  <f:validateRegex
    pattern="^[_a-z0-9-]+(\\.[_a-z0-9-]+)*@[a-z0-9-]+(\\.[a-z0-9-]+)*(\\.[a-z]{2,6})$" />
</p:inputText>
<p:message for="email" style="color:red" display="text" />

```

La fonctionnalité Virement 1/6

- ProxiBanque souhaite que ses conseillers et ses directeurs d'agence puissent réaliser des virements de compte à compte à partir de l'application.

 - 2 enjeux importants pour Proxibanque :
 1. Conserver un historique des transactions
 2. L'ensemble des opérations doivent se faire au complet ou pas du tout (rollback)
 Atomicité d'une transaction.
-

La fonctionnalité Virement 2/6

Réponses aux contraintes

- L'historique des virements sera assuré par l'inscription de chaque virement dans une table spécifique de la base de données
 - L'atomicité des transactions sera assurée grâce à un service fourni par Spring : Spring Transaction Management
-

La fonctionnalité Virement 3/6

La solution technique : Spring

➡ Framework transversal qui apporte de nombreux services, dont :

- L'injection de dépendance (Intérêt technique : couplage faible – Bénéfice : réutilisation et évolutivité du code)
 - L'interception (permettra d'implémenter une fonctionnalité de log – Bénéfice : assurer une traçabilité plus fine des transactions)
 - Spring Data pour l'accès aux données (améliore la maintenabilité du code)
 - Spring Transaction Management, pour gérer l'atomicité des transactions
-

La fonctionnalité Virement 4/6

PROXIBANOUE

Vous êtes connecté en tant que Dupond

Liste des clients

Nom	Prénom	Adresse	Email	Code postal	Ville	Comptes	Modification	Suppression
Chorro	Matthieu	21 rue de rueil	chorro@gtm.c	92200	Montpellier	Comptes	Modifier	Supprimer
Carigny	Nathalie	10 rue des Antilles	carigny@gtm.c	75000	Sètes	Comptes	Modifier	Supprimer
Hilger	Jerome	21 rue de la chute libre	hilger@gtm.co	78200	Montpellier	Comptes	Modifier	Supprimer
Frantz	Rudy	10 rue de la jobarteam	frantz@gtm.cc	78000	Sètes	Comptes	Modifier	Supprimer
Loizillon	Franck	35 rue du magicien	loizillon@gtm.	75000	Sètes	Comptes	Modifier	Supprimer
Pescher	Amélie	10 rue du cinema	pescher@gtm.	75000	Sètes	Comptes	Modifier	Supprimer

[Effectuer un virement](#)

PROXIBANOUE

liste compte

Titulaire du Compte	Numero du Compte	Solde
Chorro Matthieu	12001200	2000.0
Carigny Nathalie	12001201	13750.0
Hilger Jerome	12001202	6000.0
Frantz Rudy	12001203	7000.0
Loizillon Franck	12001204	9000.0
Pescher Amélie	12001205	10000.0
Chorro Matthieu	22002200	3000.0
Carigny Nathalie	22002201	4000.0
Hilger Jerome	22002202	6000.0
Frantz Rudy	22002203	7000.0
Loizillon Franck	22002204	9000.0
Pescher Amélie	22002205	10000.0

Virement

Compte débiteur

12001201

Compte créditeur

12001200

Montant

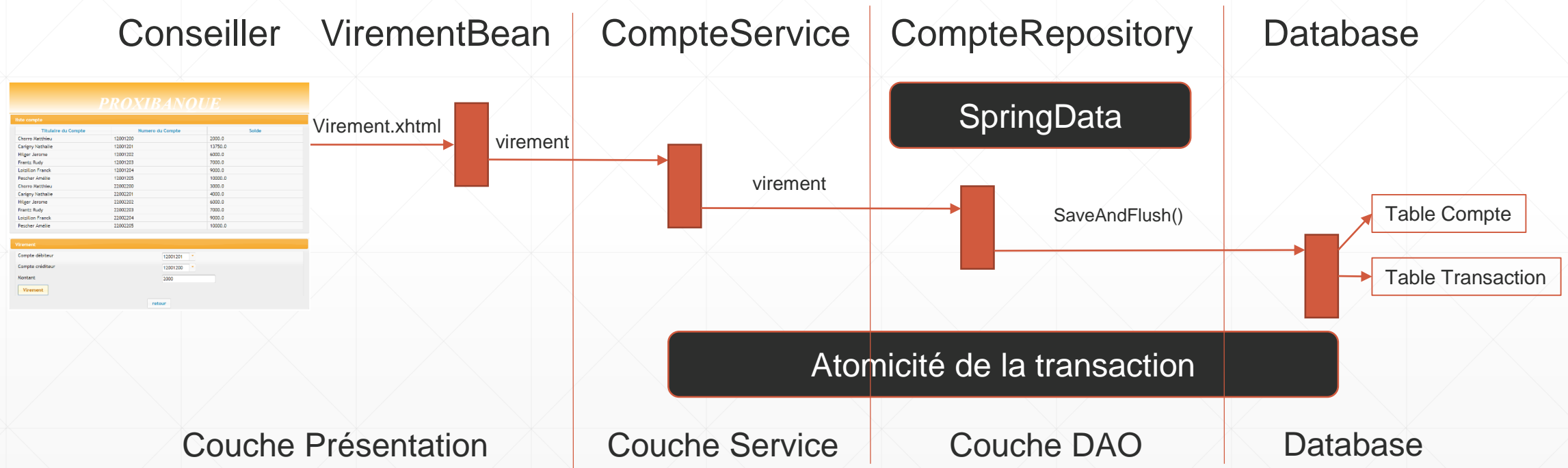
2000

Virement

retour

La Fonctionnalité Virement 5/6


Représentation technique d'un virement



La Fonctionnalité Virement 6/6

- Mise en œuvre de Spring Transaction Management

```
@Configuration
@ComponentScan
@EnableTransactionManagement
@EnableJpaRepositories
public class SpringConfiguration {
```



```
@Transactional
public void virement(Compte compteDebiteur, Compte compteCrediteur, double montant) {

    DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
    // get current date time with Date()
    Date date = new Date();
    //System.out.println(dateFormat.format(date));

    compteDebiteur.setSolde(compteDebiteur.getSolde() - montant);
    compteCrediteur.setSolde(compteCrediteur.getSolde() + montant);
    compteRepository.saveAndFlush(compteCrediteur);
    compteRepository.saveAndFlush(compteDebiteur);

    Transaction transaction = new Transaction();
    transaction.setDate(date);
    transaction.setNumeroCompteCrediteur(compteCrediteur.getNumeroCompte());
    transaction.setNumeroCompteDebiteur(compteDebiteur.getNumeroCompte());
    transaction.setMontant(montant);
    transactionRepository.save(transaction);
}
```


Sécurisation de l'application

Problématique : Toutes ces fonctionnalités utilisent des **données sensibles**
Craintes du Directeur de banque sur la **protection des données** des clients
Seuls les **employés autorisés** peuvent utiliser l'application

Contraintes : **Il faut séparer les rôles !**

Un conseiller ne possède pas les droits du Directeur de banque!
Un conseiller ne peut accéder aux clients d'un autre conseiller



Implémentation d'un service de sécurité



Implémentation du service JAAS

JAAS : Java Authentication and Authorization Service

Pourquoi JAAS ? JAAS est une Spécification Standard de Java Entreprise Edition fournie par le serveur d'application

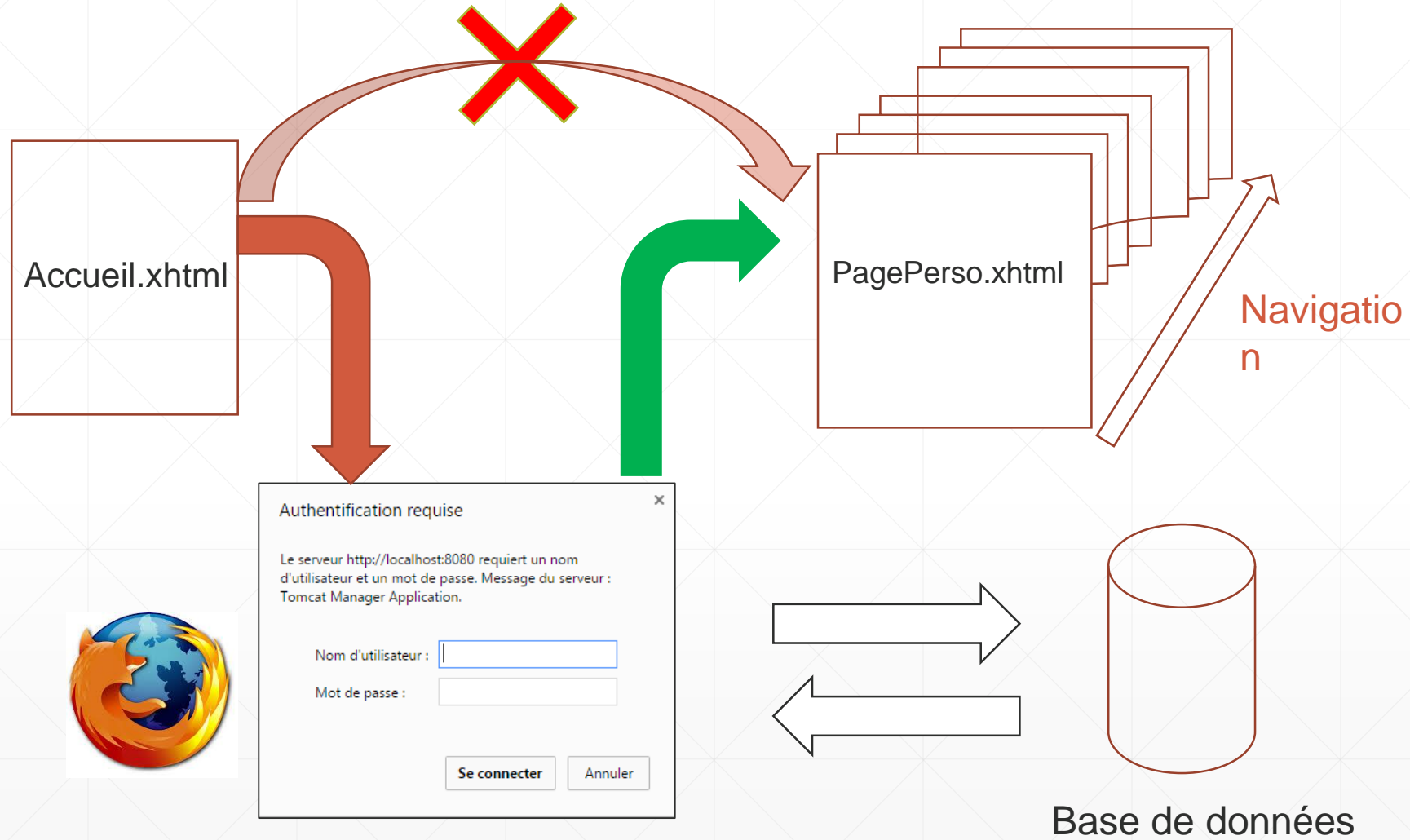
JAAS répond à 3 questions claires

Quoi ? Quelles pages web doivent être protégées?

Qui ? Qui a le droit d'accès à ces pages et à leurs fonctions?

Comment ? Quelle méthode implémenter pour protéger mon application?

Fonctionnement de JAAS



JAAS nécessite la configuration du serveur d'application et de la base de donnée

Configuration de JAAS

Serveur d'Application TOMCAT

Quoi?
Qui?
Comment?

Web.xml

Où trouver les
Informations de
Login Password

Context.xml

Base de données MySQL

nom	pwd
Martin	azerty
Untel	P@ssw0rd

users

nom	role
Martin	conseiller
Untel	gerant

user_roles



Travail de configuration assurée par le gestionnaire de Base de données

Configuration de JAAS ?

```
<security-constraint>  
  <web-resource-collection>  
    <web-resource-name>HTML Manager interface (for humans)</web-resource-name>  
    <url-pattern>/PageWeb/*</url-pattern>  
  </web-resource-collection>  
  <auth-constraint>  
    <role-name>gerant</role-name>  
  </auth-constraint>  
</security-constraint>
```

```
<security-role>  
  <description>  
    The role that is required to access the HTML Manager pages  
  </description>  
  <role-name>gerant</role-name>  
</security-role>
```

```
<login-config>  
  <auth-method>BASIC</auth-method>  
  <realm-name>Tomcat Manager Application</realm-name>  
</login-config>
```

Conclusion

```
<security-constraint>  
  <web-resource-collection>  
    <web-resource-name>HTML Manager interface (for humans)</web-resource-name>  
    <url-pattern>/PageWeb/*</url-pattern>  
  </web-resource-collection>  
  <auth-constraint>  
    <role-name>gerant</role-name>  
  </auth-constraint>  
</security-constraint>
```

```
<security-role>  
  <description>  
    The role that is required to access the HTML Manager pages  
  </description>  
  <role-name>gerant</role-name>  
</security-role>
```

```
<login-config>  
  <auth-method>BASIC</auth-method>  
  <realm-name>Tomcat Manager Application</realm-name>  
</login-config>
```

Conclusion

Partie Projet :

- Très intéressant de se retrouver « en situation » réelle
- Intérêt du travail en équipe
- Difficulté sur l'intégration des différentes parties de l'application

Partie Formation :

- Acquisition conséquente de connaissances et de savoir-faire dans le domaine du développement JAVA JEE
 - Petite préférence pour la partie back-office, même si j'ai aussi du plaisir à travailler sur la couche présentation
-