

# Transactions et Composants d'entreprise

**Didier DONSEZ**

**Université Joseph Fourier (Grenoble 1)**

***IMAG/LSR/ADELE***

***Didier.Donsez@imag.fr***

# Les modèles de composants :

## caractérisation d'un composant

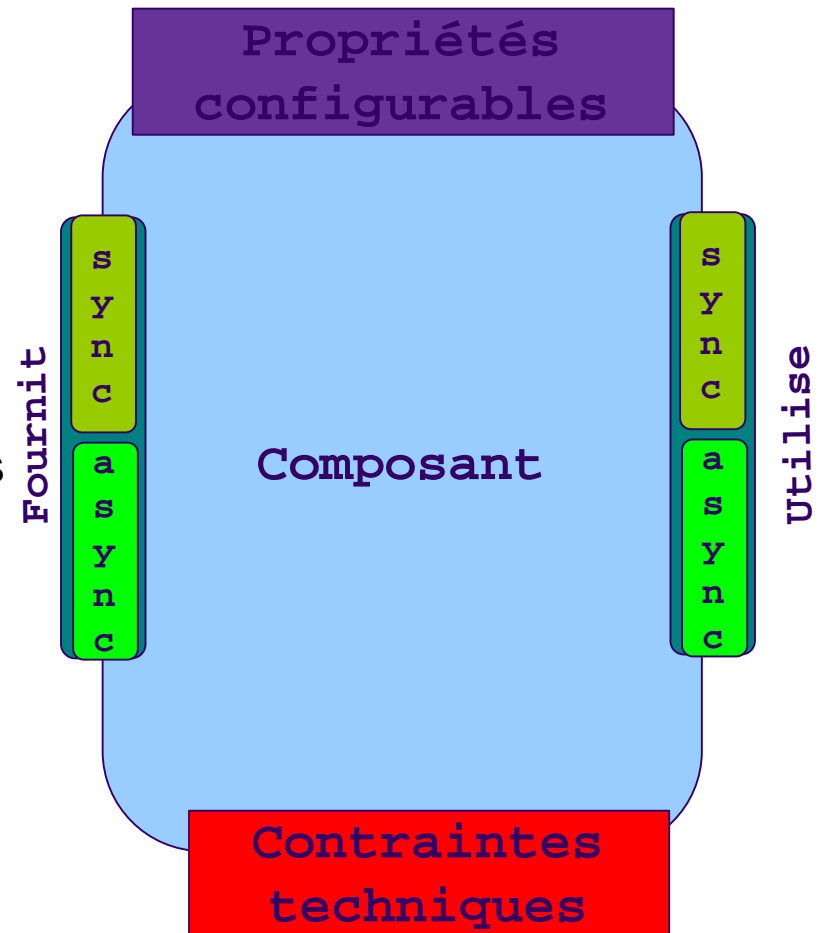
### → Comment coopère un composant

- Ce que fournit le composant (entrées)
  - composantes, interfaces, opérations, propriétés
- Ce qu'utilise le composant (dépendances)
  - composition et références aux autres composants
- modes de communication des connecteurs (synchrone, asynchrone, flots)

### → Propriétés configurables du composant

### → Contraintes techniques (QoS)

- middleware : placement, sécurité, transaction
- internes : cycle de vie, persistance
- implantation : OS, bibliothèques, version



# Les modèles de composants : conteneurs et structures d'accueil

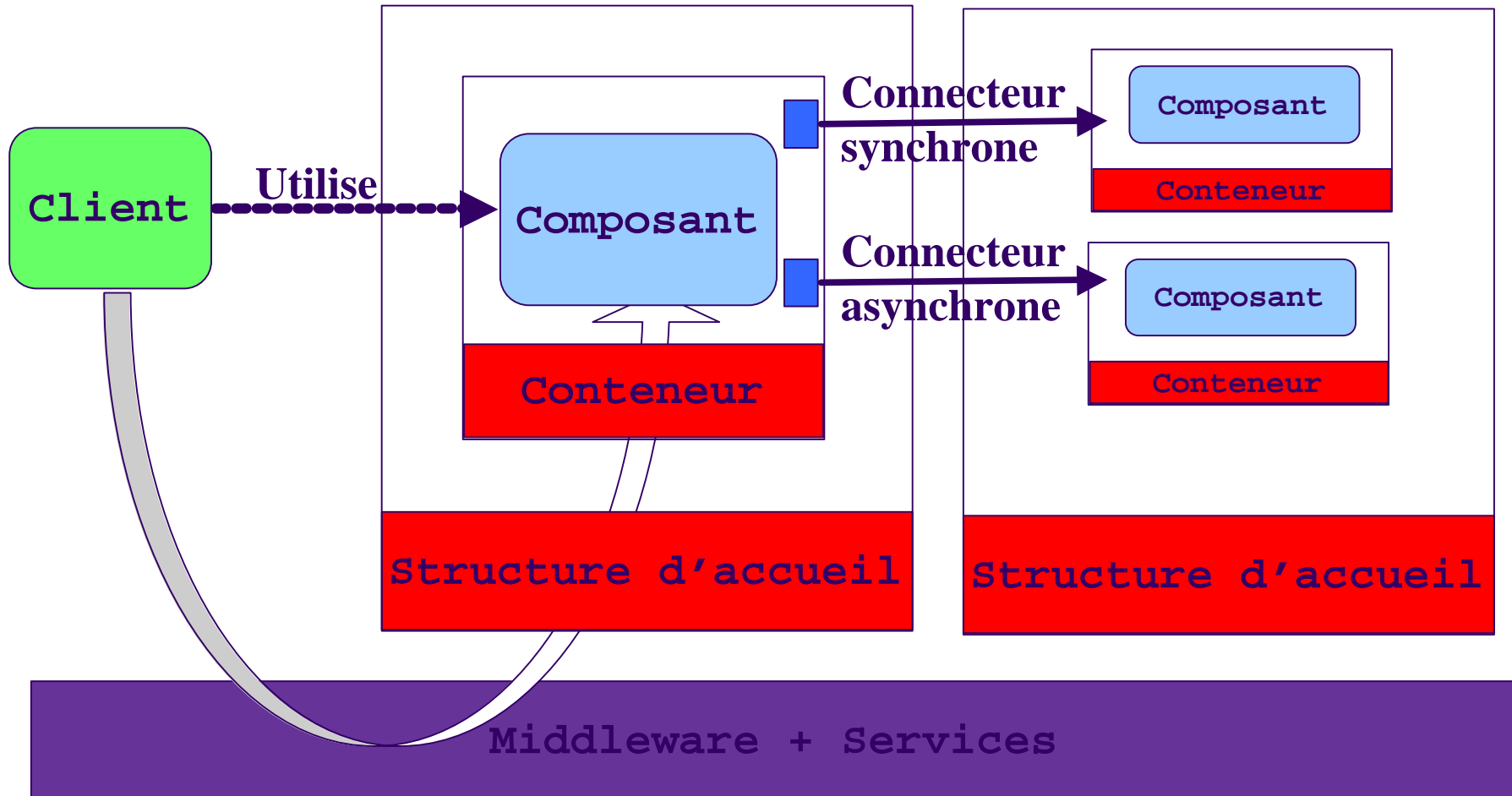
## → Conteneur

- encapsulation d'un composant (et ses composantes)
- prise en charge (masque) les services systèmes
  - nommage, sécurité, transaction, persistance ...
- prise en charge partielle des connecteurs
  - invocations et événements
- techniquement par interposition (ou délégation)

## → Structures d'accueil

- espace d'exécution des conteneurs et des composants
- médiateur entre les conteneurs et les services systèmes
- des + comme le téléchargement de code (navigateur)

# Les modèles de composants : conteneurs et structures d'accueil

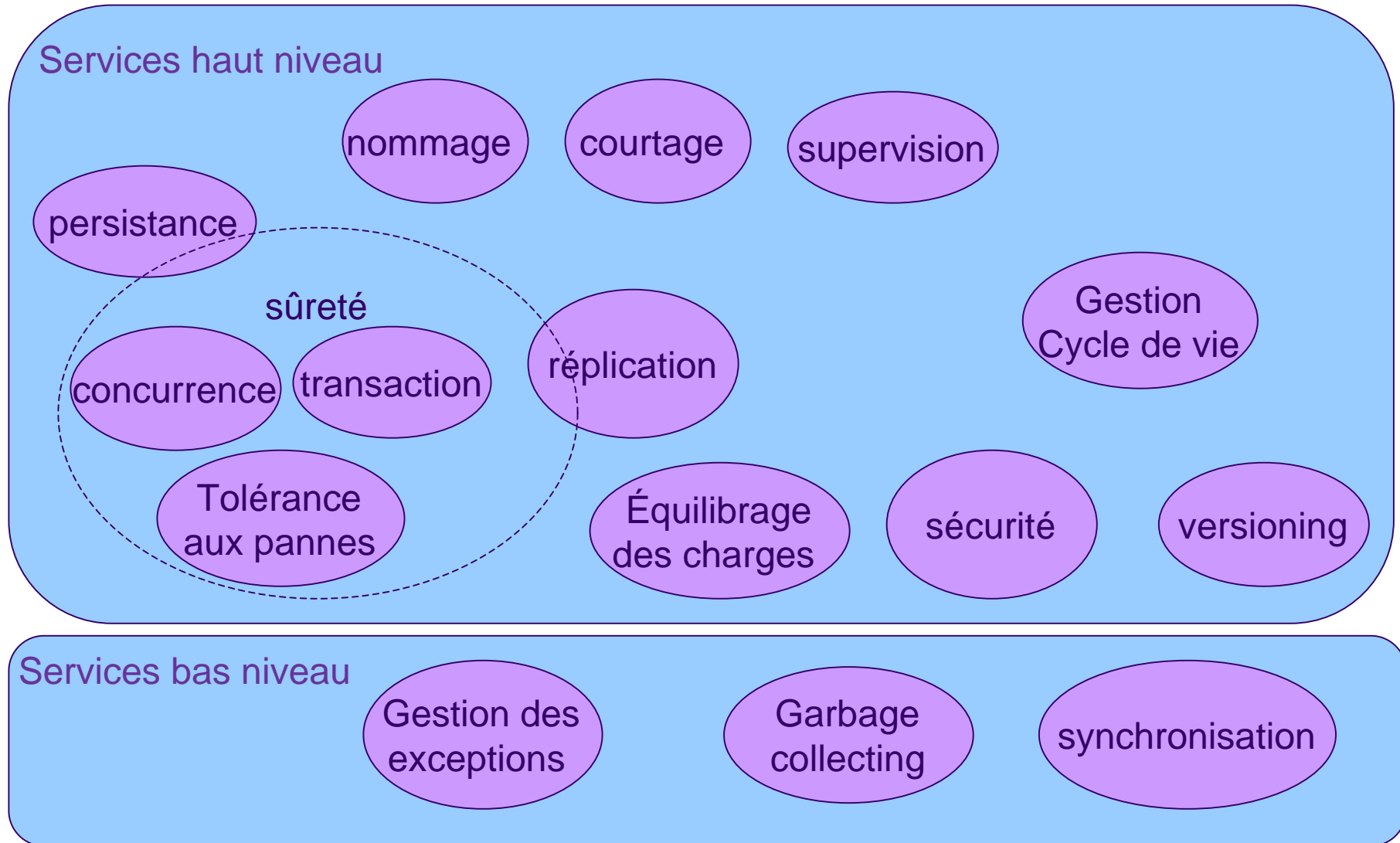


# Service non fonctionnel

## → Définition

- Service qui n'ajoute pas de nouvelle fonctionnalité à l'application
- Apporte une qualité de service supérieure
  - Sûreté de fonctionnement
    - Transactions, Contrôle de concurrence, ...
  - Sécurité
  - Nommage ...
- La mise en œuvre des SNF est généralement complexe et spécialisé
- Le développeur du composant doit en être déchargé

# Classification



# Comparaison des services non-fonctionnels



	<b>Enterprise JavaBeans</b>	<b>CCM</b>	<b>COM+ / .NET</b>
<b>transaction</b>	<b>JTA et JTS</b>	<b>OTS et concurrency service</b>	<b>MTS</b>
<b>persistance</b>	<b>JDBC/JMS</b>	<b>Persistent State Service</b>	<b>ADO .NET</b>
<b>nommage</b>	<b>JNDI</b>	<b>Naming/Trading service</b>	<b>LDAP/UDDI</b>
<b>sécurité</b>		<b>Security service</b>	<b>Passport</b>
<b>versioning</b>	<b>CVS</b>		<b>DLL</b>
<b>Événements</b>	<b>JMS</b>	<b>Event service (MOM)</b>	<b>MSMQ</b>

# EJB : Caractéristiques principales

- L'architecture EJB identifie les éléments suivants :
  - composants logiciels ou beans (EB),
  - conteneurs,
  - serveurs,
  - clients.
- Les conteneurs isolent les beans du client et d'une implémentation spécifique d'un serveur
  - Rappel : les beans sont dans la partie serveur
- Conteneurs et serveurs implémentent les services non fonctionnels utilisés par les applications
  - transactions, persistance, gestion mémoire, sécurité, ...



# Transaction : Gestion Déclarative

## → **Modèle de transaction plat**

- pas de transactions imbriquées (nested)

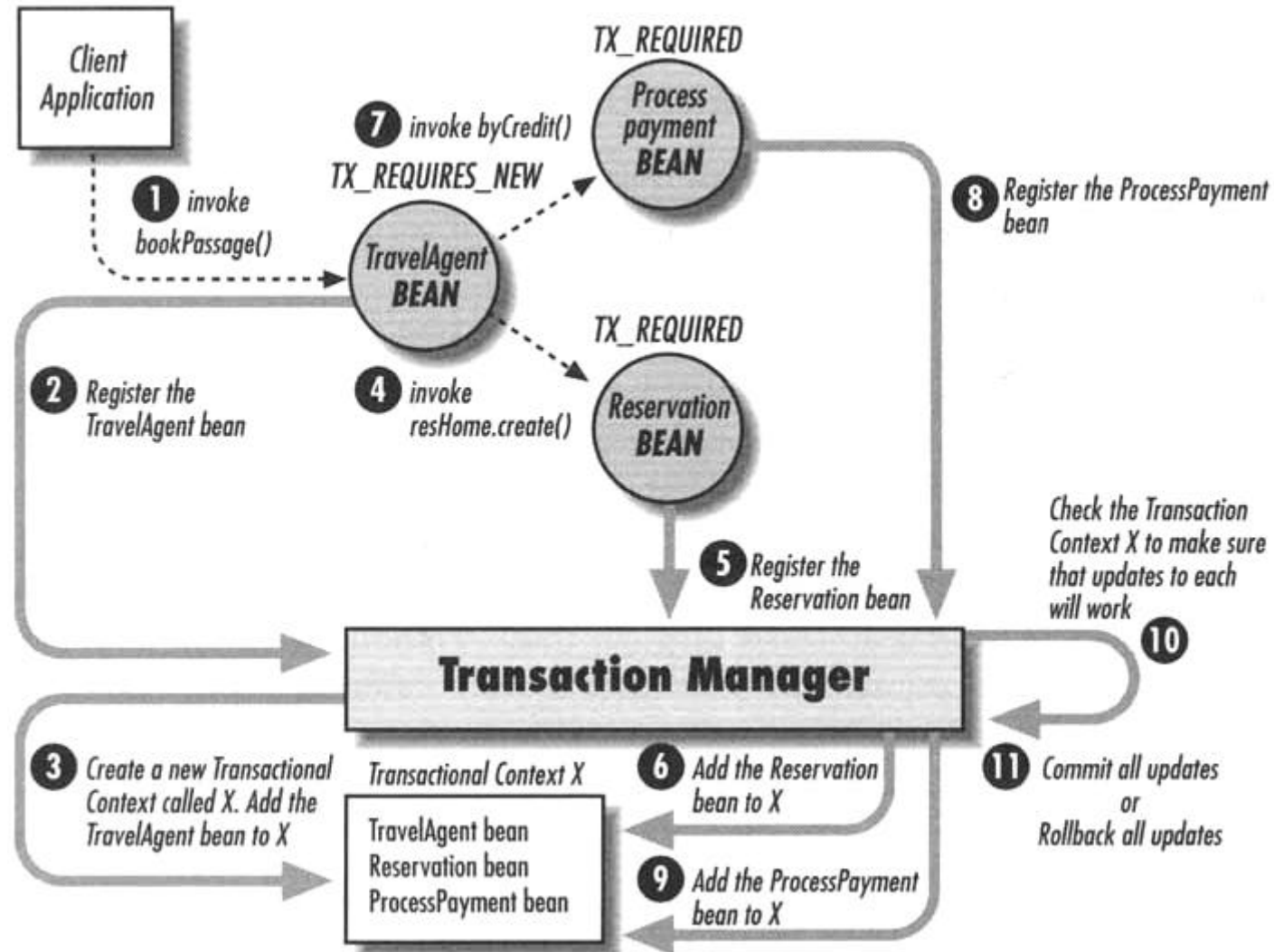
## → **Gestion “déclarative” des transactions**

- l'attribut transactionnel associé aux méthodes des EBs est affecté soit lors de la phase d'implémentation du bean, soit lors de son déploiement

# Attributs transactionnels EJB

Beans Transaction Attribute	Client's Transaction	Transaction associated bean's method
TX_NOT_SUPPORTED	None T1	None None
TX_NEVER	None T1	None Error
TX_REQUIRED	None T1	T2 T1
TX_SUPPORTS	None T1	None T1
TX_REQUIRES_NEW	None T1	T2 T2
TX_MANDATORY	None T1	Error T1
TX_BEAN_MANAGED	None T1	

# Transaction : Exemple



# Modèle de transactions avancées (i)

## → Close Nested Transactions

- OMG' OTS

## → Open Nested Transactions

- Nested Transactions can commit update before the main transaction completion
  - break the Isolation rule
- Compensation are fired if main transaction aborts

## → Bourgogne Transactions

- Flexible models
- Transaction may delegate the commit or the abort of updates to other transactions

## → SAGAS, Contracts, ...

## → Projects : PEPiTA & IMPACT

- SOAP and BTP integrations in JOnAS

# Modèle de transactions avancées (ii)

## → Nouveaux attributs transactionnels pour les transactions imbriquées (fermées et ouvertes)

Beans Transaction Attribute	Client's Transaction	Transaction associated bean's method
TX_REQUIRES_NEWSUB	None T1 T1.x	T2 T1.x (Close) T1.x.y (Close)
TX_MANDATORY_NEWSUB	None T1 T1.x	Error T1.x (Close) T1.x.y (Close)
TX_REQUIRES_NEWOPENSUB	None T1 T1.x	T2 T1.x (Open) T1.x.y (Open)
TX_MANDATORY_NEWOPENSUB	None T1 T1.x	Error T1.x (Open) T1.x.y (Open)

## → Compensation configurable

- Thèse de Sergiy Nemchenko (Valenciennes)
- Composants « ombres » implantant une politique particulière de compensation
  - Une des politiques est choisie au déploiement
- Nombreux problèmes ouverts :
  - Cascades de compensation, Contrôle de Concurrence

# Composants Avancées

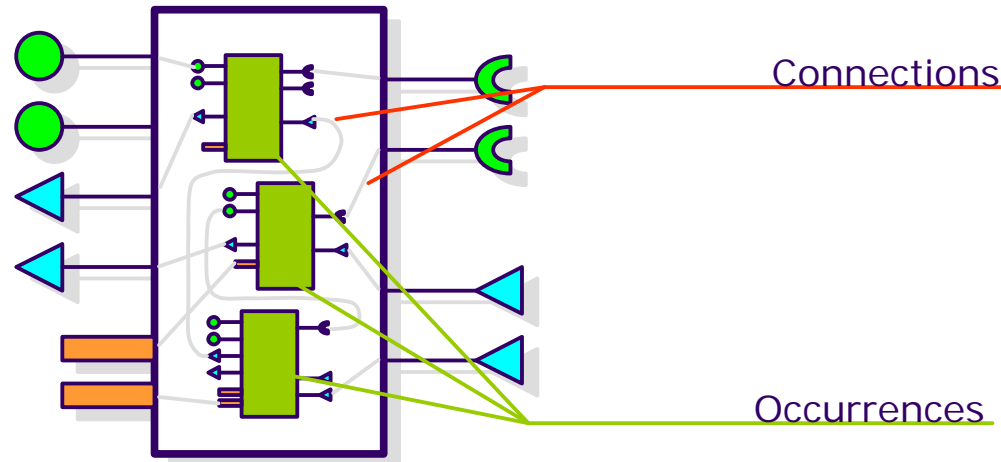
## → Conteneurs Ouverts ou Extensibles

- Pas de services non fonctionnels prédéfinis
- Ajout de services NF custom au déploiement
  - Monitoring, Modèle de transaction avancées, ...
- Techniques : AOP, Proxy dynamique, ...
- Avantages
  - Extensibilité sans passer par la normalisation
  - Pas d'overhead à payer systématiquement

# Composants Avancées

## → Composants récurrents

- Un composant peut être un assemblage (agrégat) de composants connectés



## → Problème ouvert

- propagation des attributs NF sur la composition !
  - Sécurité, Transactions, Compensation !