



Exécution de processus

Mineure SOA

Cécile Hardebolle
cecile.hardebolle@supelec.fr

Programme

8 nov.	Introduction. Enjeux, rôle de l'architecte SI <i>Partie n°1 du cas d'étude</i>
15 nov.	Architecture et cartographie DI.I3E
22 nov.	Modèle SOA
29 nov.	Modélisation de processus <i>Partie n°2 du cas d'étude</i> DI.I3E
6 déc.	Web Services <i>Partie n°3 du cas d'étude</i> DI.I3E
13 déc.	Cloud <i>Partie n°4 du cas d'étude</i>
20 déc.	Exécution de processus DI.I3E
10 jan.	Compléments et ouverture. Conclusion <i>Partie n°5 du cas d'étude</i>

Deux intervenants :

 Olivier Besnard (Solucom)
 Cécile Hardebolle (Supélec)

27 jan.

Examen : présentation de vos travaux sur **l'étude de cas « Chaus'Star »**

Au programme ce matin...

- ▶ Les principes de l'exécution de processus
- ▶ Deux types de plateformes d'exécution de processus
- ▶ De la pratique !



à retenir



avancé



démo/exercice

Plan

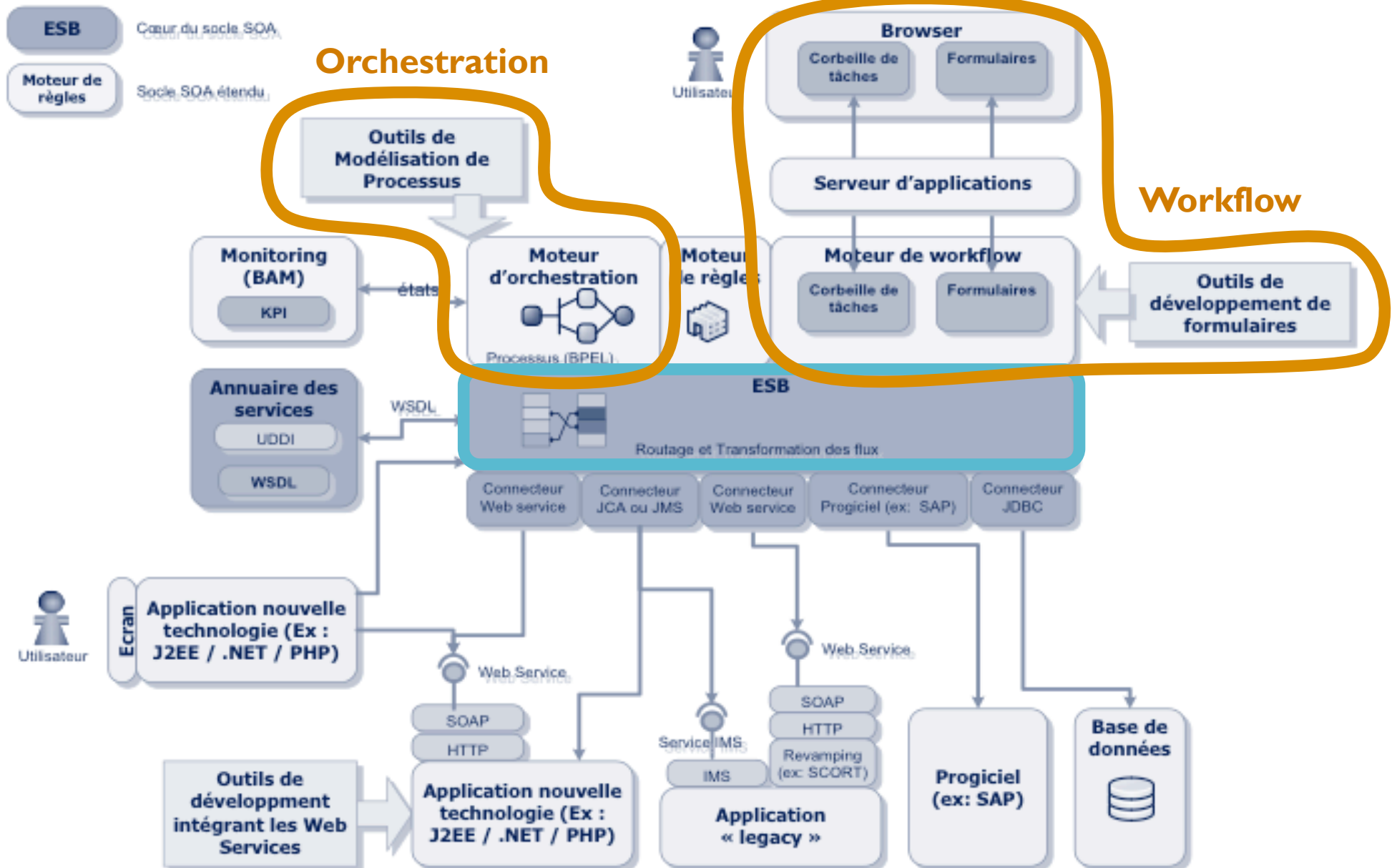
- ① Qu'est-ce qu'exécuter un processus ?
- ② Moteur de workflow
- ③ Moteur d'orchestration, BPEL

Exécution d'un processus ?

- ▶ **Exécuter** un processus = transformer le processus en **programme**
 - ▶ Variables pour stocker les données
 - ▶ Instructions pour réaliser les actions
 - ▶ Entrées/sorties/connecteurs pour l'interaction avec l'utilisateur et le reste du SI
 - ▶ Boucles, conditionnelles...
- ▶ Nécessité d'une **plateforme d'exécution**
= un programme qui exécute le processus-programme
 - ▶ Gère la mémoire, exécute les instructions, gère les entrées/sorties/connecteurs...
- ▶ Plateformes utilisées dans la démarche SOA, appuyées sur l'ESB :
 - ▶ **Moteur de workflow** : processus semi-automatisés
 - ▶ **Moteur d'orchestration** : processus tout automatisés



Enterprise Service Bus (ESB)



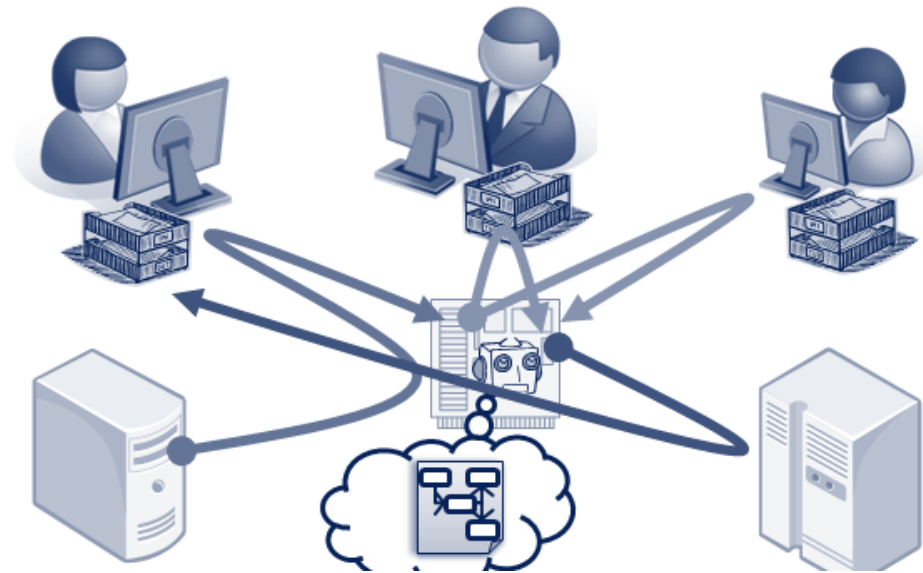
Plan

- ① Qu'est-ce qu'exécuter un processus ?
- ② Moteur de workflow
- ③ Moteur d'orchestration, BPEL

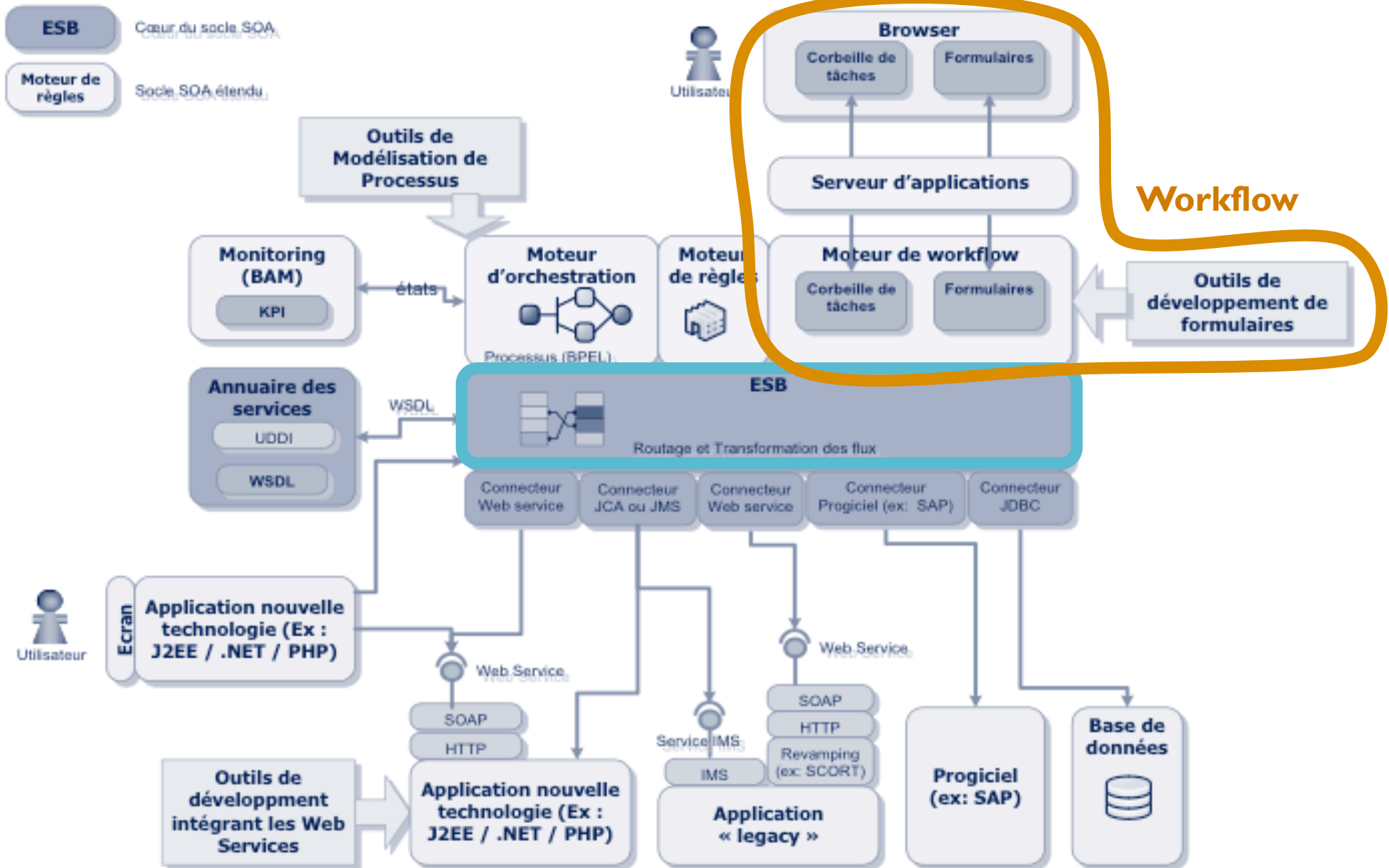


Moteur de workflow

- ▶ **Editeur** pour le développement processus-programme
 - ▶ Tâches **humaines** : génération automatique ou semi-automatique d'applications (formulaires, pages web...)
 - ▶ Tâches **automatiques** : utilisation de connecteurs (web services, emails...)
- ▶ **Plateforme** d'exécution pour gérer les liens logiques entre les tâches
 - ▶ Serveur d'application
 - ▶ Application de suivi des tâches
- ▶ **Administration** et suivi



Enterprise Service Bus (ESB)



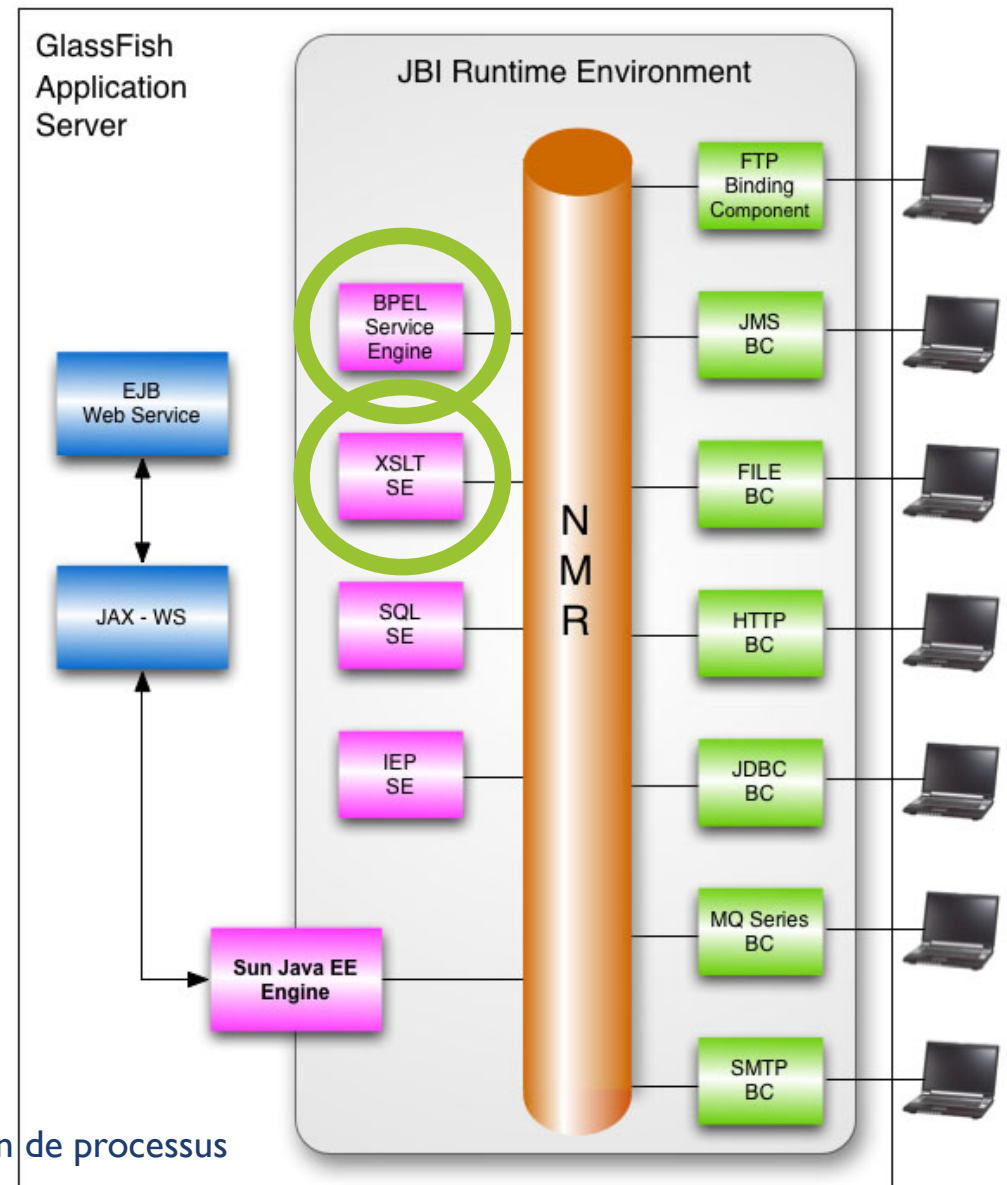
Plan

- ① Qu'est-ce qu'exécuter un processus ?
- ② Moteur de workflow
- ③ Moteur d'orchestration, BPEL

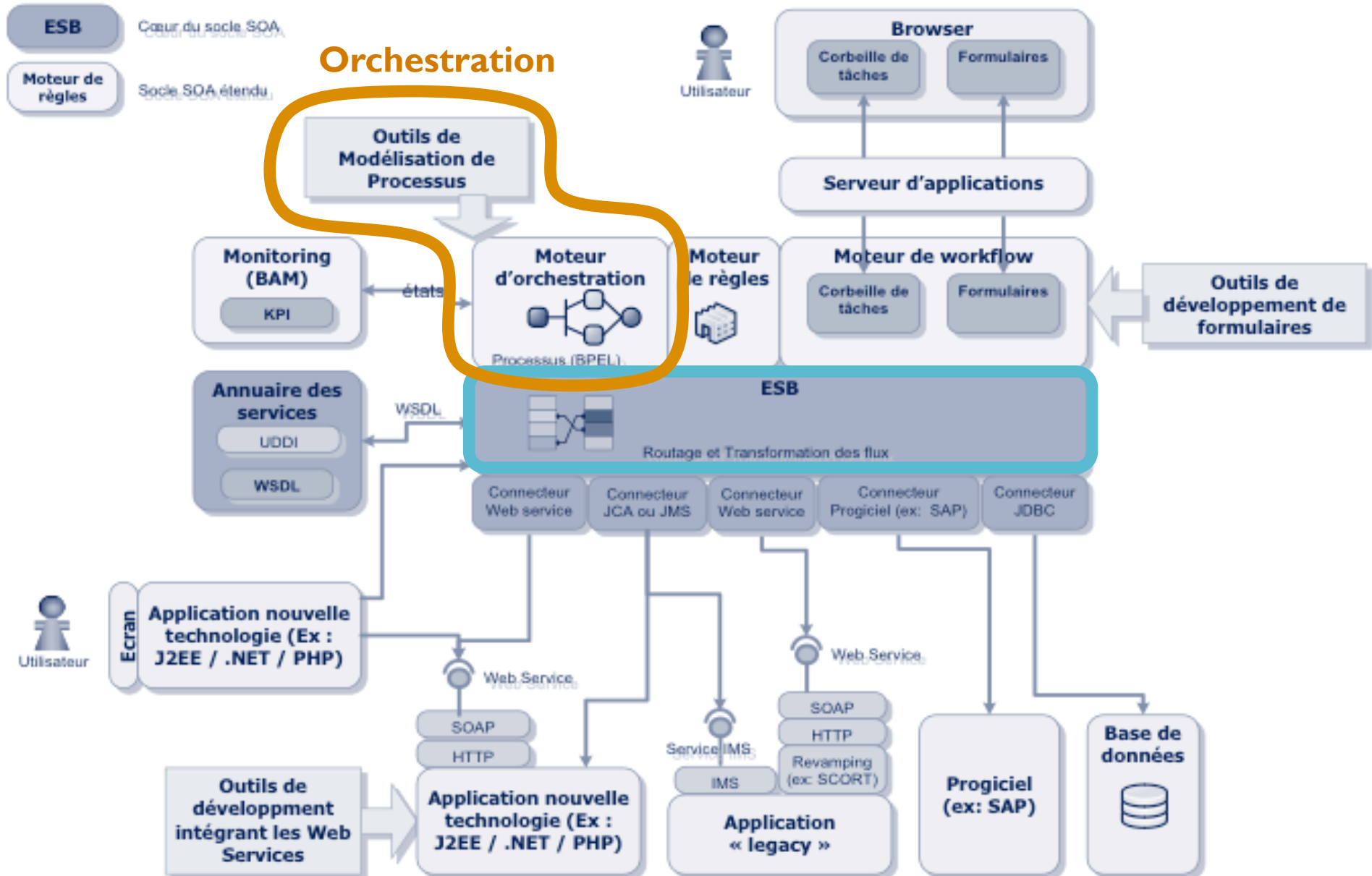
Exemple de moteur d'orchestration : GlassFish ESB



- ▶ **JB I (Java Business Integration)**
= norme qui définit l'architecture d'un **ESB** basé sur le serveur d'applications Java EE
 - ▶ Binding Components (BC)
 - ▶ Normalized Message Router (NMR)
 - ▶ Service Engine (SE)
- ▶ **GlassFish ESB** = ESB basé sur JBI
- ▶ Fonctionnalités :
 - ▶ **Orchestration** de web service
 - ▶ **Transformation** de données
 - ▶ Requêtes
 - ▶ Gestion des événements



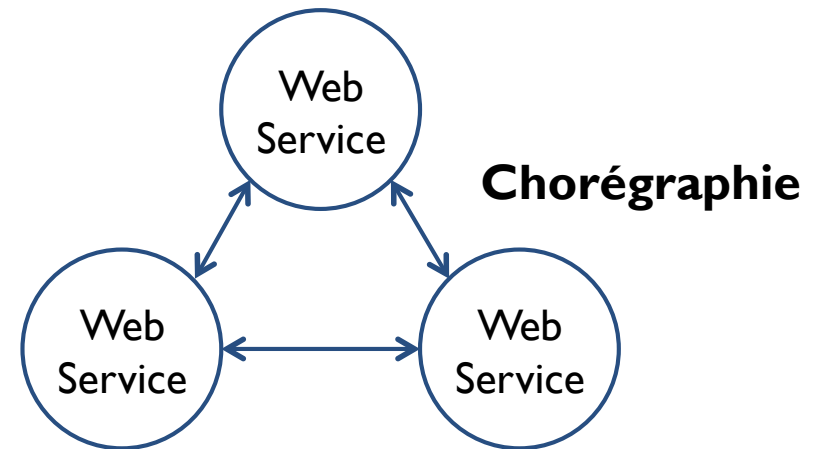
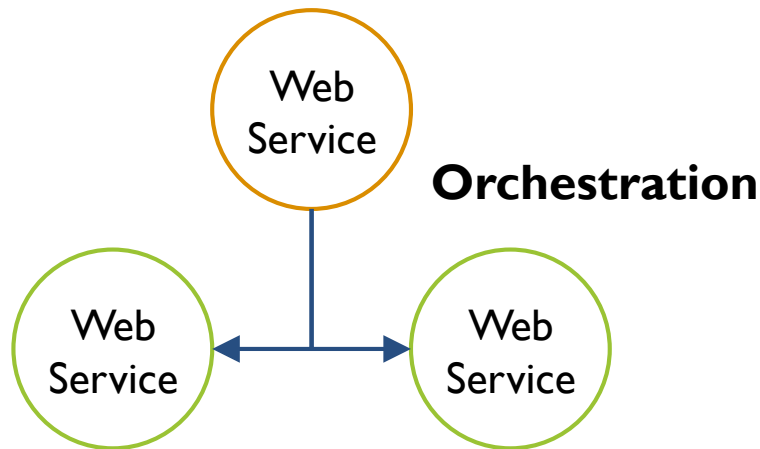
Enterprise Service Bus (ESB)



BPEL 2.0

(Business Process Execution Language)

- ▶ **BPEL** = langage pour décrire des **processus exécutables**
≈ langage de programmation très simple
 - ▶ Textuel (mais il existe des représentations graphiques)
 - ▶ Basé sur XML
- ▶ BPEL est dédié à l'**orchestration** de web services



- ▶ **Complémentaire de BPMN** : un sous ensemble de BPMN peut être transformé en BPEL pour être exécuté sur un moteur d'orchestration

Structure d'un modèle BPEL

`<process>`

`<!-- Definition and roles of process participants -->`

`<partnerLinks> ... </partnerLinks>`

`<!-- Data/state used within the process -->`

`<variables> ... </variables>`

`...`

`<!-- Exception handling -->`

`<faultHandlers> ... </faultHandlers>`

`<!-- Error recovery – undoing actions -->`

`<compensationHandlers> ... </compensationHandlers>`

`<!-- Business process flow -->`
`(activities)*`

`</process>`

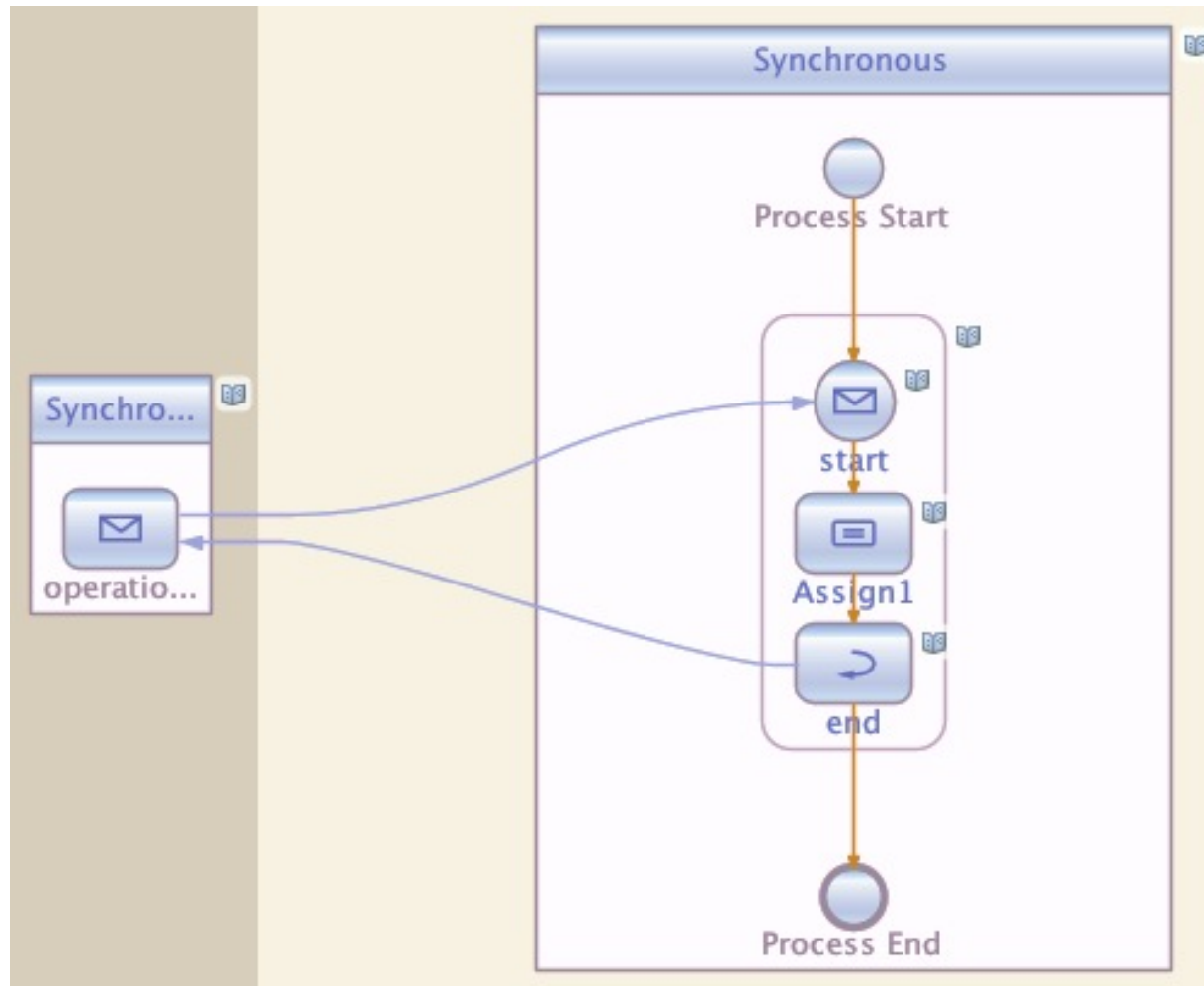


Exemple

```
...
<variables>
  <variable name="outputVar" messageType="ns1:responseMessage" />
  <variable name="inputVar" messageType="ns1:requestMessage" />
</variables>

<sequence>
  <receive name="start" partnerLink="Synchronous"
    operation="operation1" portType="ns1:portType1"
    variable="inputVar" createInstance="yes" />
  <assign name="Assign1">
    <copy>
      <from>$inputVar.inputType/ns2:paramA</from>
      <to>$outputVar.resultType/ns2:paramA</to>
    </copy>
  </assign>
  <reply name="end" partnerLink="Synchronous"
    operation="operation1" portType="ns1:portType1"
    variable="outputVar" />
</sequence>
```

Exemple : version graphique



MERCI!
THANK YOU!



FRAPAR.