Les workflow

Jean-Louis Boudart

<jeanlouis.boudart@gmail.com>

Bruno Bonfils

<asyd@asyd.net>

Ensemble,
réussissons
les grands projets du Libre

Groupe LINAGORA

27 rue de Berri
75008 PARIS
Tél. : 01 58 18 68 28
Fax : 01 58 18 68 29

Sommaire

- ✔ Définition
- ✔ Un mot sur le webflow
- ✔ Exemple d'utilisation
- ✔ Les aspects pratiques
- ✔ Présentation de JBPM

Définition

- Par workflow on entends beaucoup de choses...
  - Cinématique des interactions utilisateurs
  - La gestion des données
- **Toutes** les applications ont un workflow
  - Mais il est en général fixé, ou peu modifiable
- Quelques exemples :
  - Un wizard
  - Un formulaire suivi d'une validation

Définition

- Certaines applications métiers ont une nécessité particulière de devoir s'adapter aux besoins d'une entreprise
- Par exemple, tout gestionnaire de contenu (GED, CMS)
  - Rarement le même processus de validation
  - Un nombre de validation différents, par des acteurs différents (un relecteur, suivi d'une validation par un autre journaliste, puis une autre validation par un directeur de programme)
- Mais aussi toutes les applications liées à la sécurité
  - PKI (Gestion du cycle de vie des certificats)
  - Provisionning d'identité

Définition

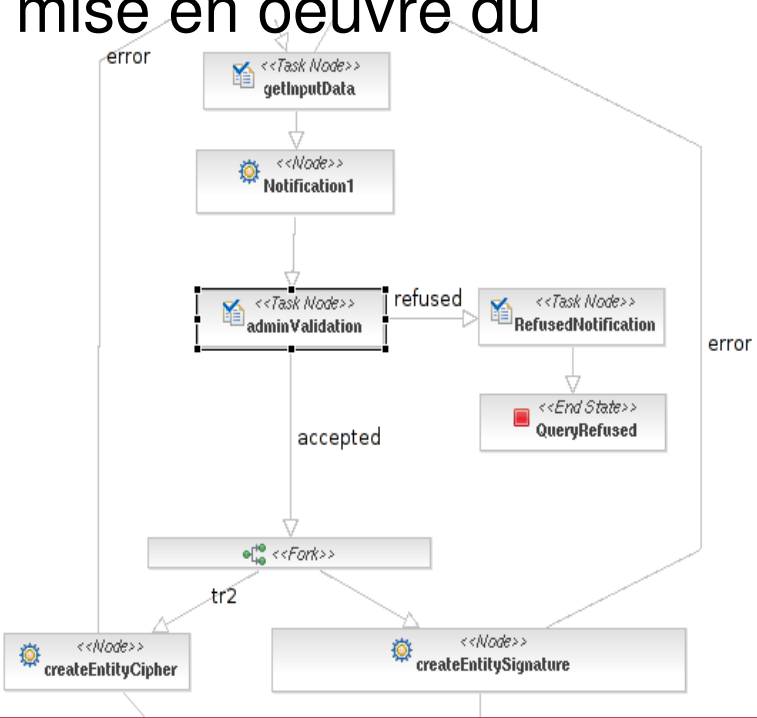
- Quelques constats :
  - Les développeurs sont rarement spécialisés métier
  - Les consultants métiers ont rarement des connaissances techniques suffisantes pour manipuler du code
- Les idées :
  - Création de moteur de workflow permettant un découpage entre les parties techniques et les parties métiers

Définition

- Certaines personnes ont réfléchis à des moteurs de workflow
  - WfMC (Workflow Management Coalition)
- Les idées :
  - Gérer, archiver les instances de workflow
  - Gérer les données manipulées au sein d'une instance de workflow
  - Orchestrer les différents modules que composent une instance de workflow (briques métiers)

Définition

- En pratique cela donne :
  - Une vision métier (représentation sous forme de graphique) d'un workflow, généralement réalisé par un consultant métier
  - Une « glue » technique permettant la mise en oeuvre du workflow métier



Un mot sur le webflow

- Les webflow (ou pageflow) sont un cas particulier des workflow dans le cas d'une utilisation d'un site web
- Décris l'enchaînement des pages
- Pas de persistance systématique
- Une brique métier (au sens workflow) peut être composé d'un webflow lorsqu'il y a interaction avec l'utilisateur via une interface web

Exemple d'utilisation

- L'équipe sécurité travaille sur le workflow pour l'application de PKI EJBCA, pourquoi ?
- Aucun client n'a le même besoin métier
- Dans le cas d'une demande de certificat, il peut exister de très nombreux points d'entrée différents, et des cinématiques totalement différentes en fonction du point d'entrée
- Les workflow métiers peuvent évoluer au sein d'un client, d'où l'idée de faire une application très modulaire.

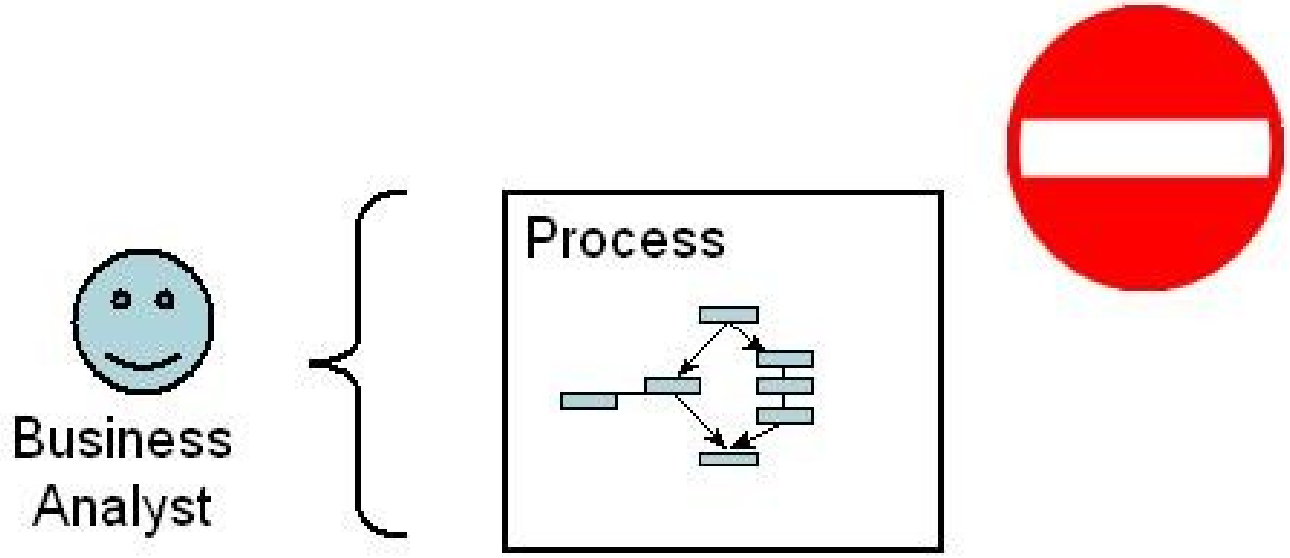
Exemple d'utilisation

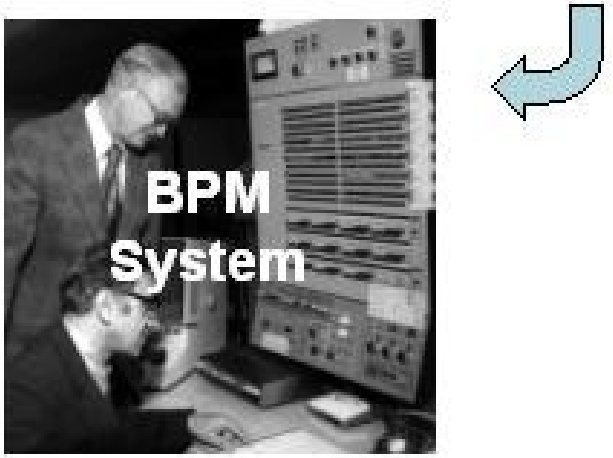
- Pour bien comprendre, prenons un exemple :
  - La DRH – à la signature d'un contrat – rempli un formulaire web (outil de gestion des identités interne à la société)
  - Cela provoque la demande de deux certificats pour cet utilisateur (signature et chiffrement)
  - A son arrivée, l'employé génère une carte à puce avec son certificat de signature
  - Plus tard dans la journée, obtention de son certificat de chiffrement

Exemple d'utilisation

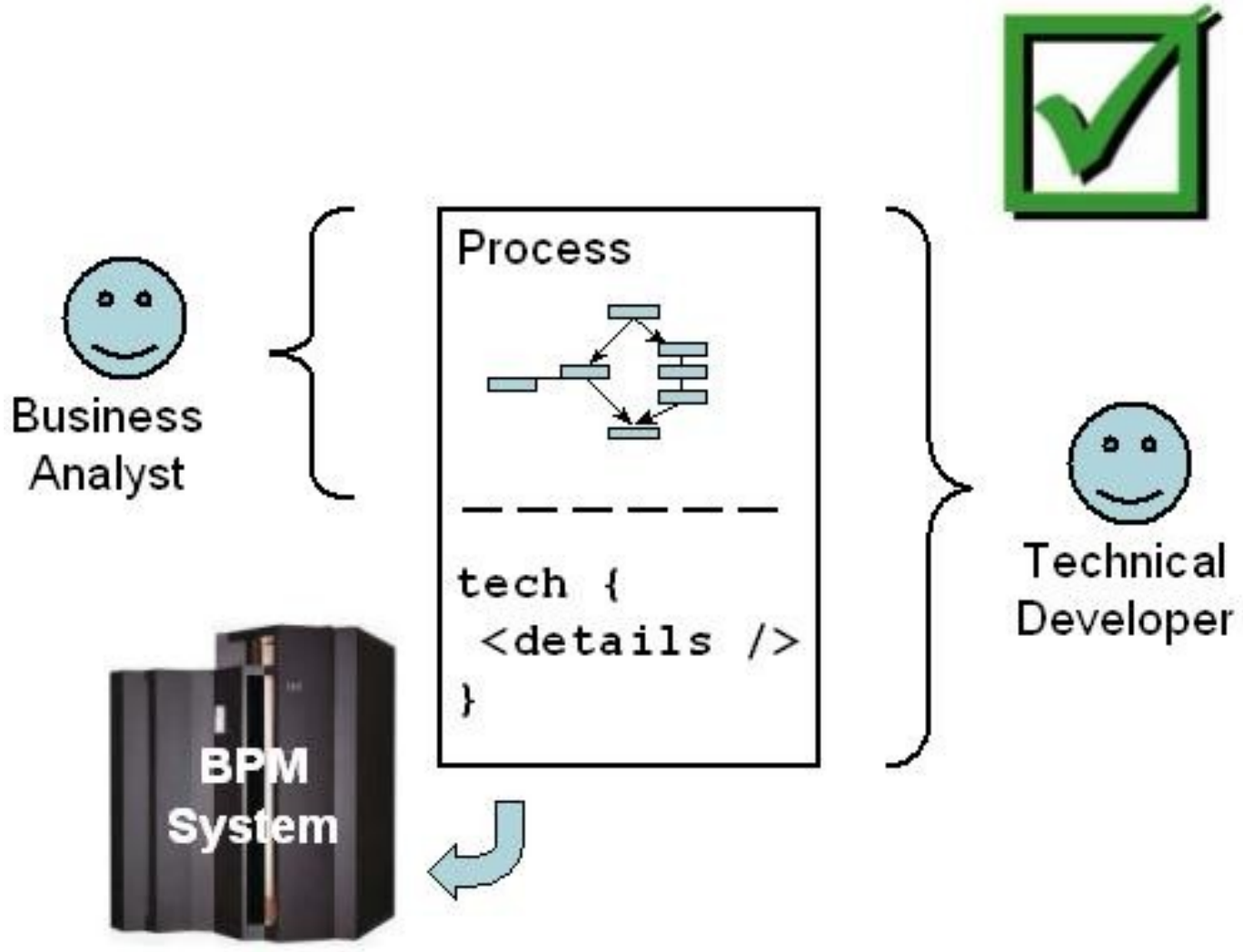
- Et un autre totalement différent :
  - Une entreprise souhaite équiper tous ses routeurs de certificats pour authentification ipsec
  - Les routeurs génèrent une demande de certificat (via le protocole SCEP) auprès de la PKI
  - Un administrateur doit valider les demandes

La vision idéaliste vu par les managers





La vision réelle des choses



Un petit mot

- Un petit mot avant de passer à la technique...
  - L'utilisation d'un moteur de workflow n'est pas forcément facile à appréhender
  - Cela nécessite de (très) bien architecturé son application (cf. Domain Driven Design par exemple)
  - Penser aux performances
- Mais cela apporte...
  - Un moindre coût *a posteriori* de développement – une fois l'application finalisée – pour modifier le workflow, et y rajouter des composants
  - Des buzzword pour vendre !

La technique

Les aspects pratiques

- Dans un premier temps, il est question de définir un langage permettant de représenter le workflow, malheureusement il en existe un certains nombres :
  - BPM (Business Process Management)
  - XPDL (XML Process Definition Language)
  - BPEL (Business Process Execution Language)
- La définition de ces langages permettent d'utiliser indépendamment différents moteurs de workflow
- Permet également la création d'interface graphique permettant la manipulation des workflow

- Présentation de JBPM :
  - Projet racheté par JBoss
  - Licence apache
  - L'un (sinon le) des projets de workflow le mieux documenté
  - Très léger : 2 jar indépendants, **ne nécessite pas l'utilisation d'un serveur d'application** (ex: JBoss)
  - Le seul à supporter la plupart des langues (XPDL, BPEL, etc.)
  - Enrichi XPDL pour donner JDPL (JBPM Process Definition Language)

JDPL :
JBPM Process Definition Language

Quelques notions : les types de nodes

- Un workflow est composé d'un ensemble de node :
  - Un (unique) **startNode** : étape à l'origine de la création de l'instance du workflow
  - **TaskNode** : étape en attente d'une interaction avec l'utilisateur
  - **State** : étape en attente d'une action extérieure (webservices, attente de réponse d'un composant eterne à l'application)
  - Un ou plusieurs **endNode** : archivage de l'instance du workflow et libération des ressources
- Les nodes task et state ont un état **wait**

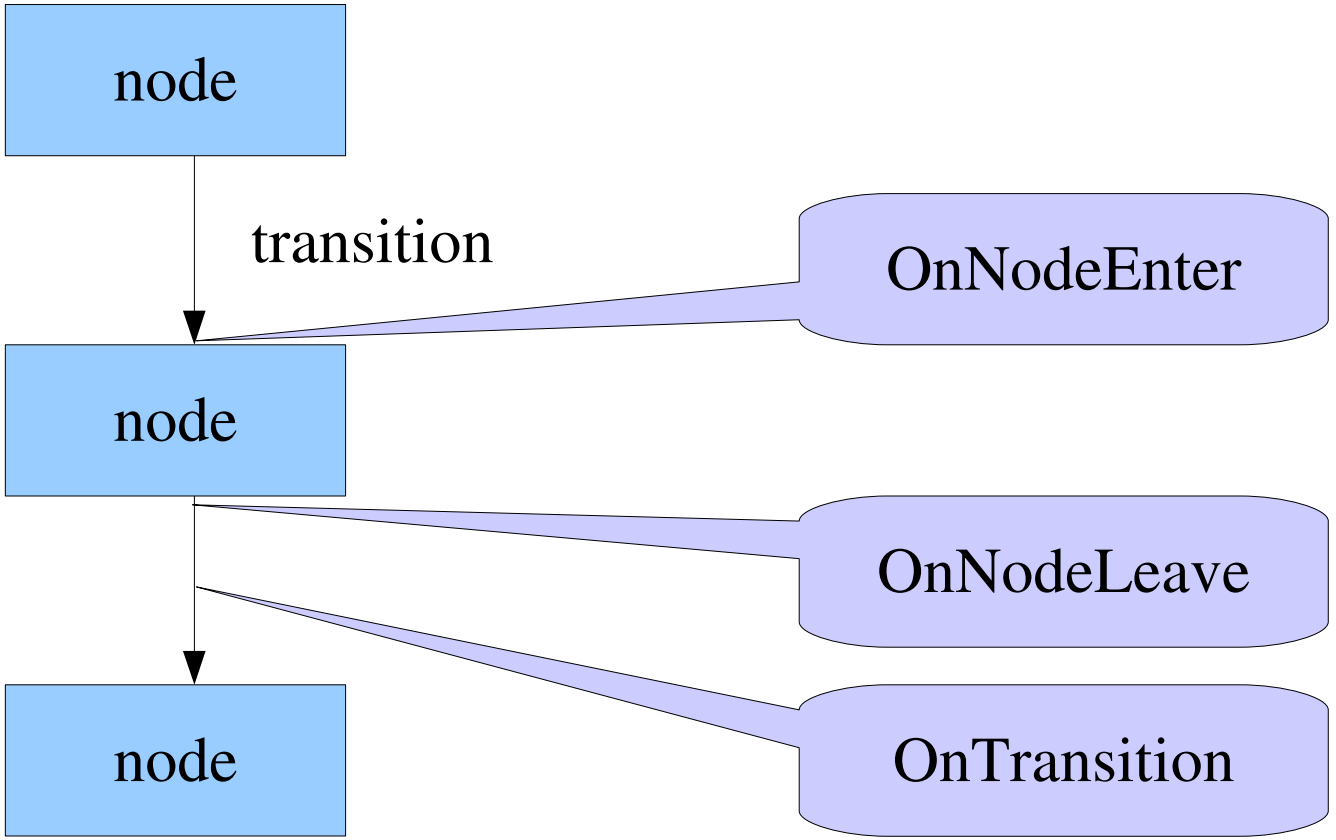
Quelques notions : les types de nodes

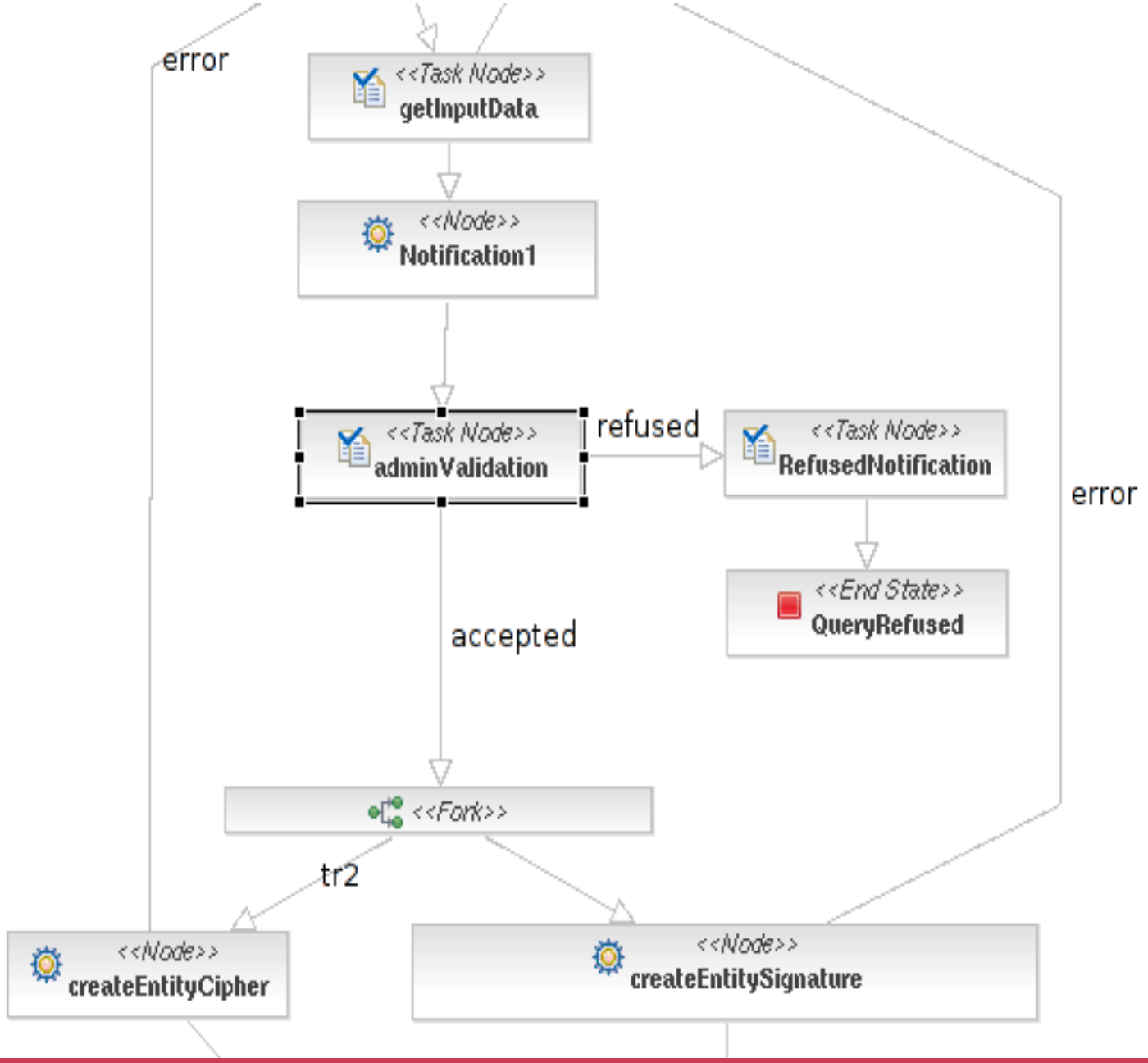
- D'autres types de nodes :
  - **Fork :** séparation du workflow en N branches devant se réunir via un node de type **join**
  - **DecisionNode :** condition sur une variable de l'instance de workflow (utilisation d'un langage simple ou délégation à une classe java)
  - **Node** : action interne à l'application mais n'ayant pas d'interaction ni avec l'utilisateur ni avec un élément externe (exemple : notification par courriel)
- On relie tout ces nodes par des **transitions**, représentant le changement d'un node à un autre au sein de l'instance

le comportement d'un node

- Chaque node possède trois événements déclencheurs :
  - **onNodeEnter** : déclenchée à l'entrée du node
  - **onNodeLeave** : déclenchée à la sortie du node (n'a pas conscience du node suivant)
  - **onTransition** : déclenchée lors de la transition vers le node suivant
- Pour information : un événement n'est pas lié qu'à une seule action, mais peut déclencher plusieurs actions

La gestion des déclencheurs




error
<<Task Node>>
getInputData
<<Node>>
Notification1
<<Task Node>>
adminValidation
refused
<<Task Node>>
RefusedNotification
error
<<End State>>
QueryRefused
accepted
<<Fork>>
tr2
<<Node>>
createEntityCipher
<<Node>>
createEntitySignature

La gestion des données dans un workflow

- **Très important** : un node n'a pas à connaître l'existence des autres nodes, il manipule un ensemble de données dans un contexte d'exécution
- **Traduction** :
  - Il n'y a pas de variables directement passées d'un node à l'autre
  - Un node utilise l'instance du workflow pour manipuler ses données
  - Limitation possible de la visibilités des variables en utilisant le mécanisme des swimlanes

Place à la démo !

LINAGORA GROUPE

PARIS - TOULOUSE - LYON

MERCI DE VOTRE ATTENTION

Ensemble,
réussissons
les grands projets du Libre

Groupe LINAGORA

27 rue de Berri
75008 PARIS
Tél. : 01 58 18 68 28
Fax : 01 58 18 68 29

www.linagora.com | www.08000linux.com | www.linagora.org