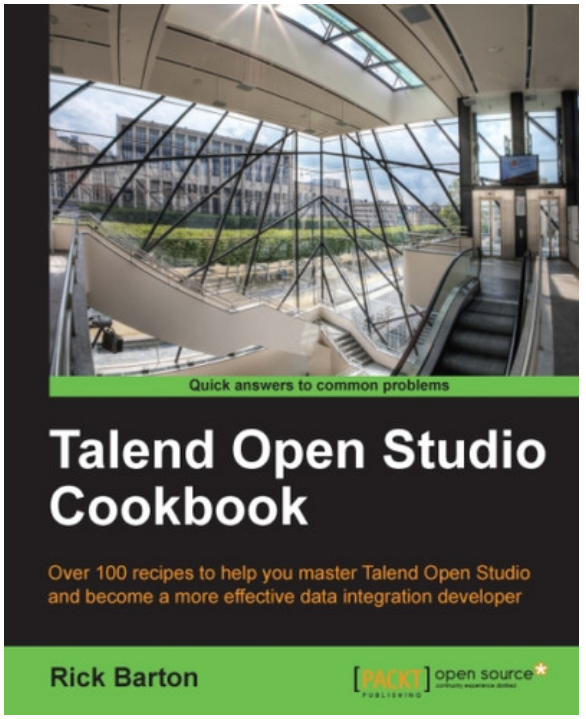
Un survol

Talend

- Outil ETL (Extract-Transform-Load)
  - Autre outil : Informatica
- Fondé à Paris en 2005
- Outil graphique, basé sur Eclipse (SWT)
- Générateur de code Java
- Plusieurs modules (perspectives) : Data Integration (DI), Enterprise Service Bus (ESB), Big Data (BD), Data Quality (DQ)

Bibliographie

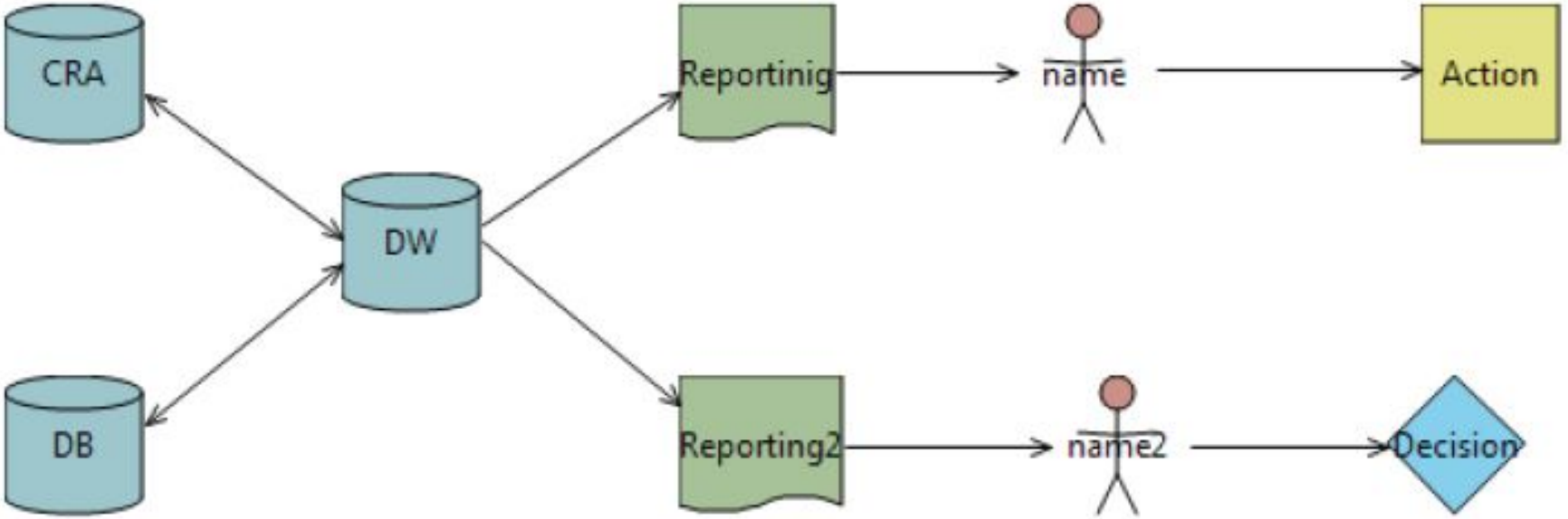


PARTIE I

Modèle d'affaires

Modèle d'affaires

Exemple d’un modèle d’affaires :



Modèle d'affaires

- Le plus haut niveau des modèles Talend
- Représente les besoins d'affaires en faisant abstraction de l'implantation technique (quoi plutôt que comment).
  - Identifier les angles morts
  - Limiter les dépassements de coûts
  - Communiquer à travers l'équipe
- A faire dès le début du projet

Modèle d'affaires

- Distinguer l'environnement (les parties externes, sur lesquels nous n'avons pas de contrôle), et le système dont nous sommes le propriétaire
- Les noeuds : source de données/documents, acteurs, décisions, actions, noeud terminal.
- Le arcs : échange d'information uni- ou bidirectionnelle.

Modèle d'affaires

Liens avec les autres modules:

- on peut associer une source de données du modèle d'affaires à une méta-donnée.
- on peut associer une action à une job, ou encore à une route.
- on peut documenter chaque élément du modèle d'affaires

PARTIE II

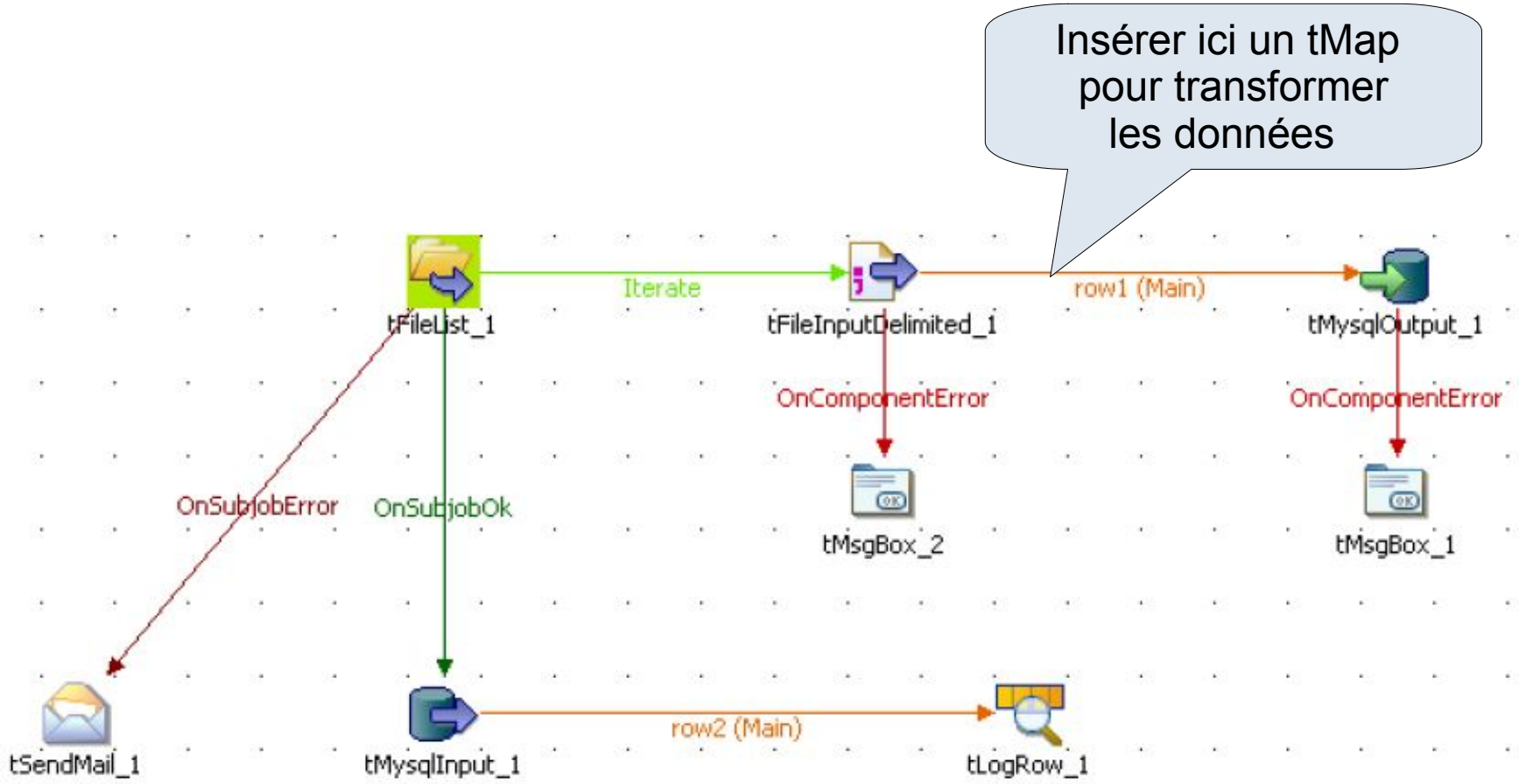
Intégration de données

Intégration de données

- Cas d'utilisation de l'intégration de données
  - Analytique : rapports, intelligence d'affaires
    - Transformer des données en information
  - Intégration opérationnelle
    - Import de données de source diverses externes (sans contrôle sur le format) vers une application interne
  - Migration de données
    - Données d'un système patrimonial (ou plusieurs) vers un nouveau système, correction de données.

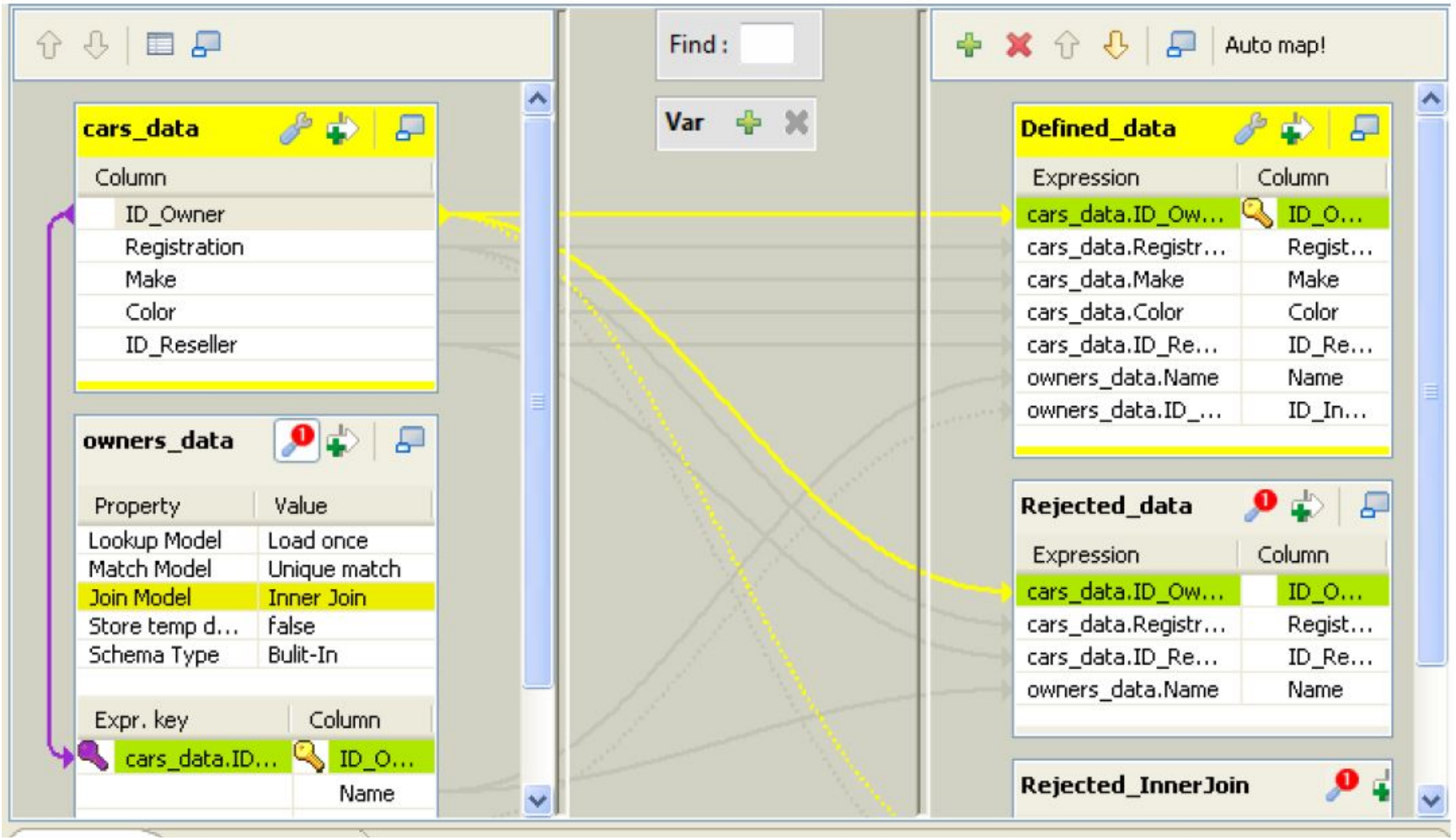
Intégration de données

Un exemple de tâche d'intégration de données:



Intégration de données

Un exemple de correspondance (tMap) :



Intégration de données

- Règle de Pareto
  - Les composants Talend remplissent les besoins 80% du temps
    - Extract : tFileInputDelimited, tFileInputJson..
    - Transform : tFilterRow, tMap, tSortRow..
    - Load: tDatabaseOutput..
  - Dans les autres cas (20%), utiliser le code natif
    - méthode Java dans un tMap (ex: trim(input_string))
    - tSetGlobalVar, tJava, tJavaRow, etc.

Intégration de données

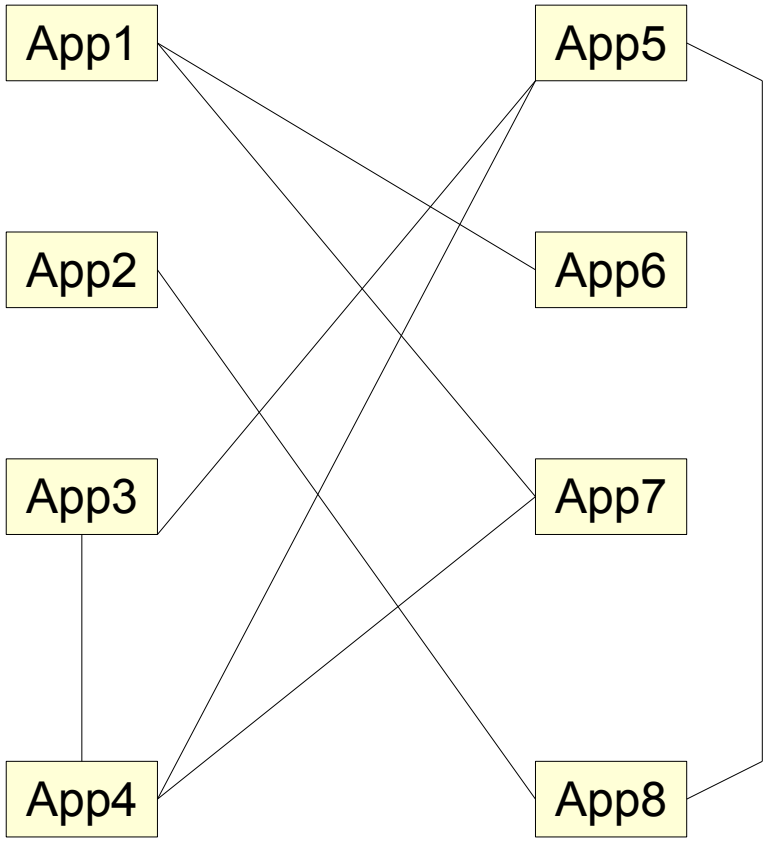
- Cas-type
  - Entrée : un fichier CSV provenant de données ouvertes (www.donneesquebec.ca)
    - Par exemple, ensemble des postes de pompiers du Québec
    - On n'a pas le contrôle sur le format
    - Lire avec un tFileInputDelimited
  - Transormation :
    - tFilter : ignorer les postes fermés
    - tMap : convertir les dates, code postaux, villes, dans un format standard
  - Chargement :
    - tDatabaseOutput : alimenter notre BD applicative

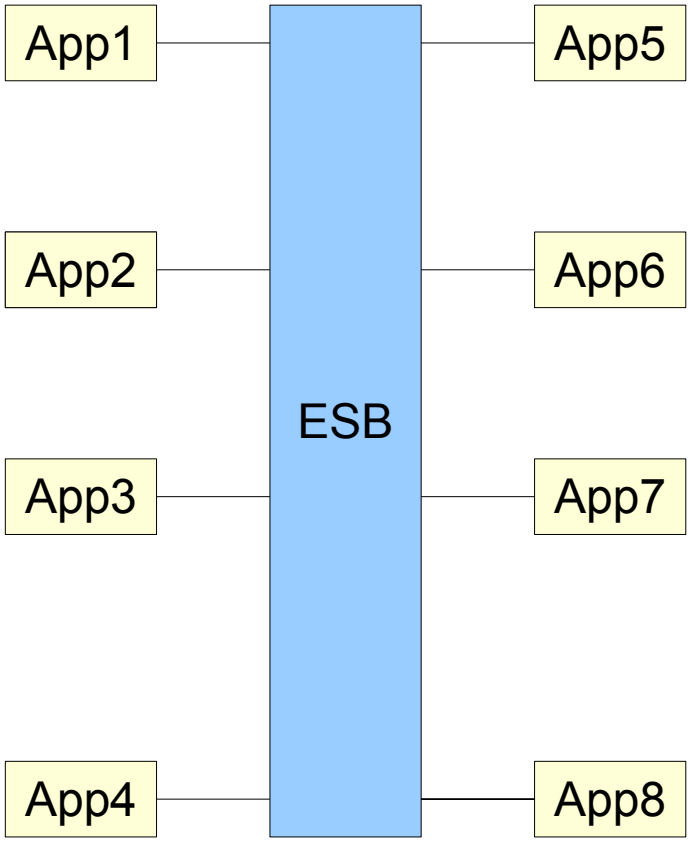
PARTIE III

Intégration applicative

Intégration applicative

- Interfacer (médiation) des applications disparates au sein d'une entreprise (ESB).





Intégration applicative

- Talend ESB : basé sur Apache Camel
- Apache Camel : implantation en Java des EIPs (Enterprise Integration Patterns)
- Envois de messages à travers des routes
  - EIPs: Message Router, Message Filter, Splitter, Aggregator, Normalizer..

Intégration applicative

- Apache Camel
  - Points d'extrémité : fichier, table, queue de messages, controller REST
  - Processeur (org.apache.camel.Processor)
    - Méthode process(Exchange exchange)
    - La classe Exchange définit Message getIn()
    - La classe Message contient un corps (body), des entêtes (headers) et des pièces jointes (attachments)
    - getBody(), getHeaders(), getAttachments()

Intégration applicative

- Apache Camel
  - Construction de route
    - classe RouteBuilder de org.apache.camel.builder
    - implanter la méthode configure() :
    - `from(srcEndPt).to(targetEndPt);`
    - `from(srcEndPt).process(processor).to(targetEndPt);`
    - `from(srcEndPt).pipeline(...)`
    - `from(srcEndPt).filter(...)`
  - Avec Talend ESB, on construit des routes graphiquement plutôt que d'utiliser la classe RouteBuilder de Camel

Intégration applicative

- Règle de Pareto (inversée)
  - Les composants Talend définissent les besoins de haut-niveau (20% du travail).
    - Ce sont des composants de haut-niveau, un peu comme les patrons de conception
    - Points d'extrémités : cMessagingEndpoint
    - Routage et processeurs : cMessageRouter
  - Pour la logique d'affaires (80% du travail), code Java personnalisé (implantation d'un processeur Camel)
  - Chaque classe implémentant `Processor` doit être testé unitairement, le paramètre `Exchange` étant simulé (*mocked).*
  - Viser 80% de couverture de code pour les processeurs.

PARTIE IV

Conclusion

Conclusion

Intégration de données (DI)	Intégration applicative (ESB)
• Axée sur le traitement des données. • Migrer les données, ou encore préparer les données pour être utilisées ultérieurement de façon efficace. • Traitement planifié (par exemple, déclenchement à minuit). • Système moyennement critique (on peut relancer manuellement la tâche, ou corriger les données en cas d'erreur) • 80% du travail fait par les composants Talend (tFileDelimited, tMap, etc.) • 20% du travail fait avec du code personnalisé (méthode dans un tMap, composant tJavaRow).	• Axée sur des messages transmis sur des routes. • Traiter les données efficacement (idéalement, elles devraient au préalable être nettoyées, corrigées). • Traitement à la demande, par une requête de service en temps réel. • Système très critique (communication inter-application). • 20% du travail fait par les composants Talend (construction de route, basé sur les EIPs implantés par Apache Camel). • 80% du travail fait par des processeurs Camel. • Doit avoir une bonne couverture de tests unitaires.