

Spring R00



Agenda

- **Introduction**
- **Fondamentaux**
- **Etude de cas**
- **Roadmap**

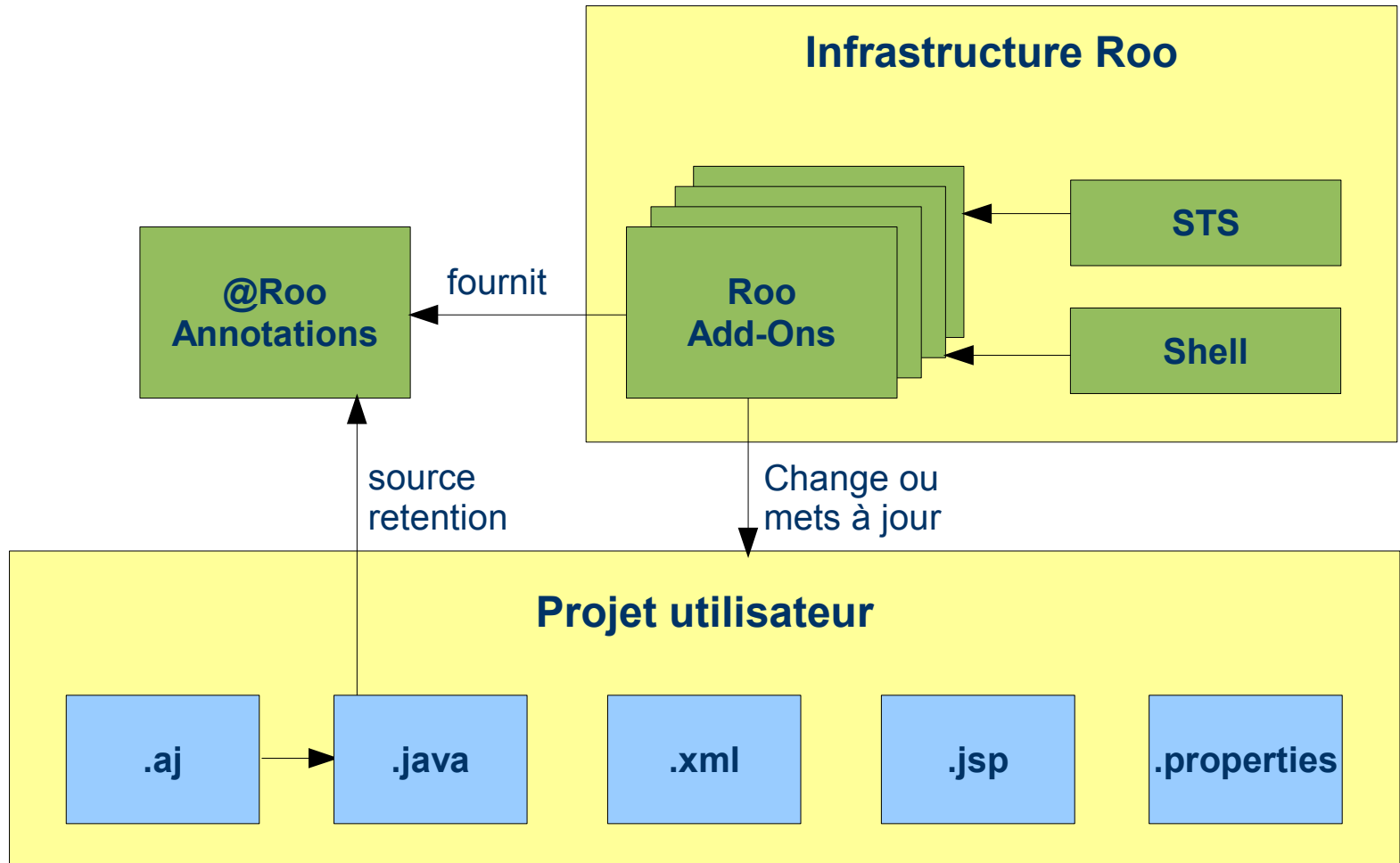
Philosophie

"Roo's mission is to fundamentally and sustainably improve Java developer productivity without compromising engineering integrity or flexibility"

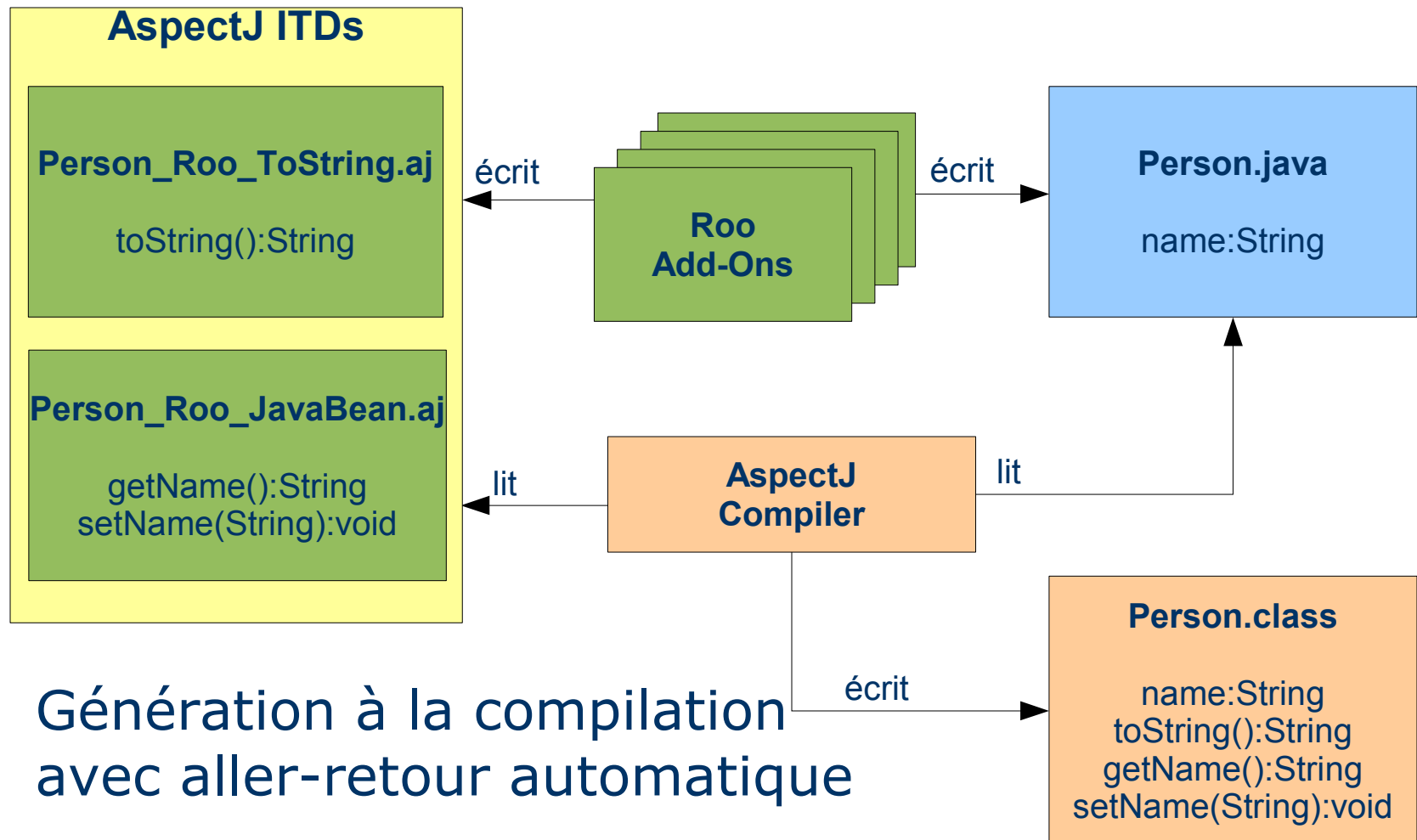
Introduction - Description

- **Générateur de code**
- **Extensible**
- **Apprentissage rapide**
- **Simple d'utilisation**
- **Utilisation de briques logicielles connues**
- **Java**

Introduction - Implémentation



Introduction - Générateur de code



Fonctionnalités

- **Création de projet**
- **Support des entités**
- **Gestion de services**
- **Gestion des attributs**
- **Persistence**
- **Création des tests Unitaires**
- **Création de finders spécifiques**
- **Gestion Spring MVC, Spring Web Flow et Selenium**

Fonctionnalités

- **Gestion SMTP, JMS**
- **Gestion de logging**
- **Gestion de la sécurité par Spring Security**
- **Gestion OSGI via les Bundlor**
- **Gestion du build**

Création de projet

➤ Commandes

- “project” → crée un nouveau projet dans le répertoire courant
- “dependency add” → ajout de dépendances du POM
- “dependency remove” → inverse de l'ajout
- “backup” → crée un ZIP du projet

Objets génériques

- Roo annotations pour les classes Java
 - `@RooJavaBean` - accesseurs
 - `@RooConfigurable` - ajoute Spring `@Configurable`
 - `@RooToString`

- Roo maintient le code de manière transparente

Entités

- “entity” → génère une entité **@RooEntity**
 - Fournit une entité JPA
 - Cette entité JPA **@Entity** contient un id, une version, l'EntityManager et un constructeur sans argument
 - Ajout automatique des méthodes, findAll, findById, findEntries, persist, remove, merge et flush

Attributs

- **“field” fournit la plupart des types d'attributs**
- **Ajout d'options spécifiques**
 - Annotations Spécifiques JPA (--fetch, --column etc)
 - Utilisation des annotations de validation JavaBean (--max etc)
 - Java modifiers (--primitive, --transient etc)
- **Le Roo Shell mémorise la dernière entité créée**
 - Tout attribut créé lui sera automatiquement ajouté
 - On peut spécifier la classe par --class

Persistence

➤ “persistence setup”

- Spécification du --provider pour Hibernate, OpenJPA et EclipseLink
- Utilisation --database pour les base de données les plus communes

➤ Related commands

- “persistence exception translation” → via AspectJ
- “database properties list” → liste des propriétés JDBC
- “database properties set” → change les propriétés JDBC
- “database properties remove” → suppression des propriétés JDBC

Persistence

- **Il est possible de réutiliser “persistence setup”**
 - On peut donc ainsi passer de Hibernate à OpenJPA, à n'importe quel moment

Finders spécifiques

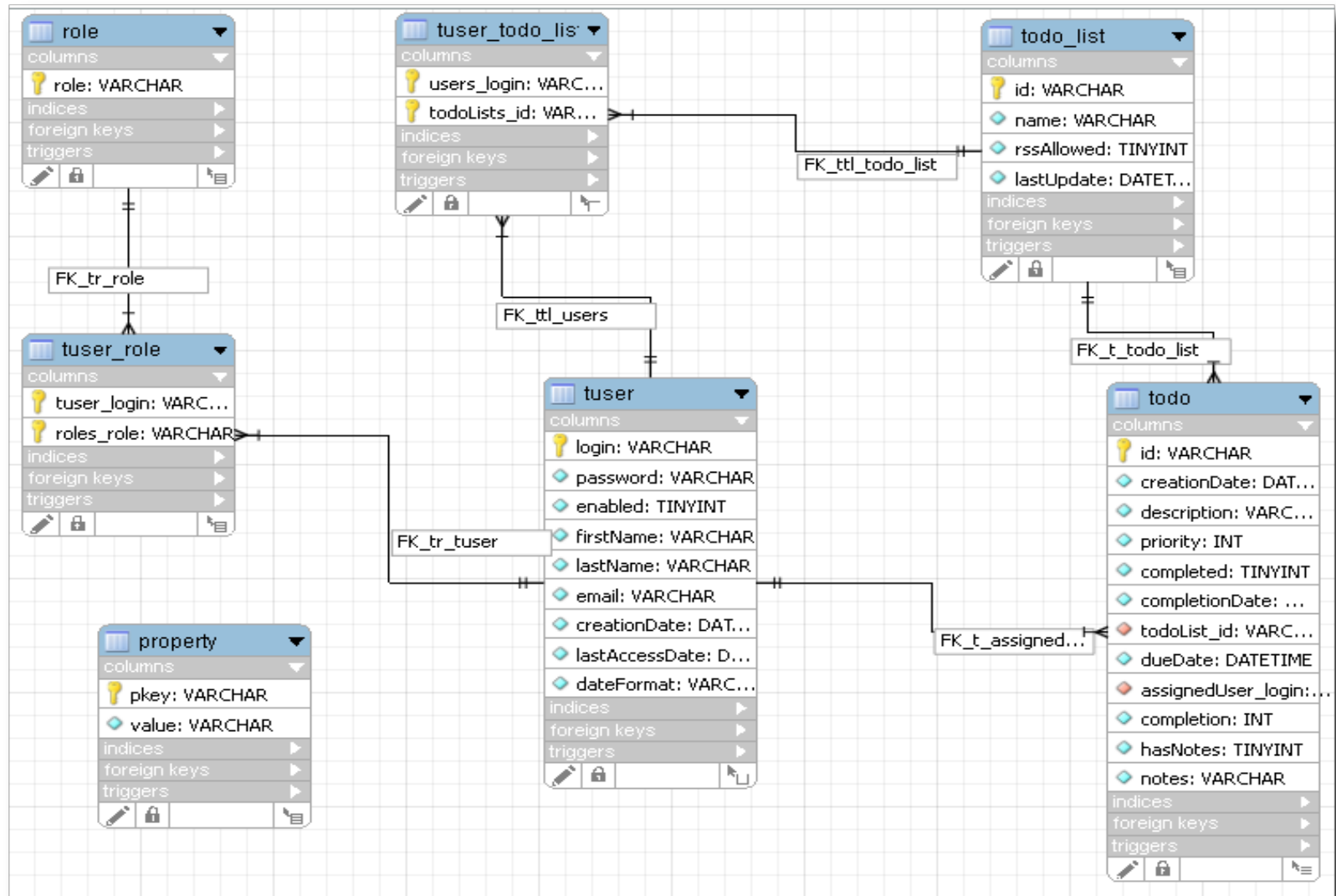
- **Il est possible de générer des finders JPA**
 - Roo les génère de manière automatique

- **Les finders sont ajoutés comme méthode statique aux entités**
 - Pas de compatibilité avec les entités abstraites
 - “finder list” → donne la liste des finders disponibles
 - “finder add” → ajout d'un finder spécifique

Spring Web MVC

- **“controller scaffold”** → création de controller Web
 - Utilisation de REST pour une entité donnée
 - Création des JSP
 - Format de date spécifique possible
- **“controller all”** → scaffold toutes les entités
- **“controller class”** → création d'un squelette de controller
 - Il faudra en coder la logique

Etude de cas - TuduList



Etude de cas - Tudulist

Google

Welcome to tudulist

http://localhost:8080/tudulist/

Hotmail | Personnaliser les l... | Windows Media | Windows | employer une nou... | jeff buckley | Autres favoris

ROO Spring

TODO

- Create new Todo
- List all Todoes

USER

- Create new User
- List all Users

TODOLIST

- Create new TodoList
- List all TodoLists

ROLE

- Create new Role
- List all Roles

Welcome to tudulist

Welcome to tudulist

Spring Roo provides interactive, lightweight and user customizable tooling that enables rapid delivery of high performance enterprise Java applications.

[Home](#) | Language: | | Menu position: [left](#) | [right](#) | Sponsored by SpringSource

Roadmap - Version actuelle

Spring Roo 1.0.0.RC2

Livré depuis Octobre 2009

Roadmap - Version 1.0.0.GA

Version 1.0.0.RC3 prévue pour Novembre

Version 1.0.0.GA sera livrée courant Décembre

Roadmap - Prochaines versions

Spring Roo 1.1.0

Prévu pour avril 2010

Plus de fonctionnalités pour Spring MVC

Gestion dynamique des add-on

Spring Roo 1.5.0

Prévu pour septembre 2010

Ajout des principaux projets Spring restants

Spring Roo 2.0.0

Génération de clients RIA (Flex/GWT)

Liens

Home → <http://www.springsource.org/roo>

Forum → <http://forum.springsource.org>

Issues → <http://jira.springframework.org/browse/ROO>

Twitter → [#roo](#) hash key