SPRING FRAMEWORK

Session 2 – 11

Mars 2018

MINES Nantes

sopra steria

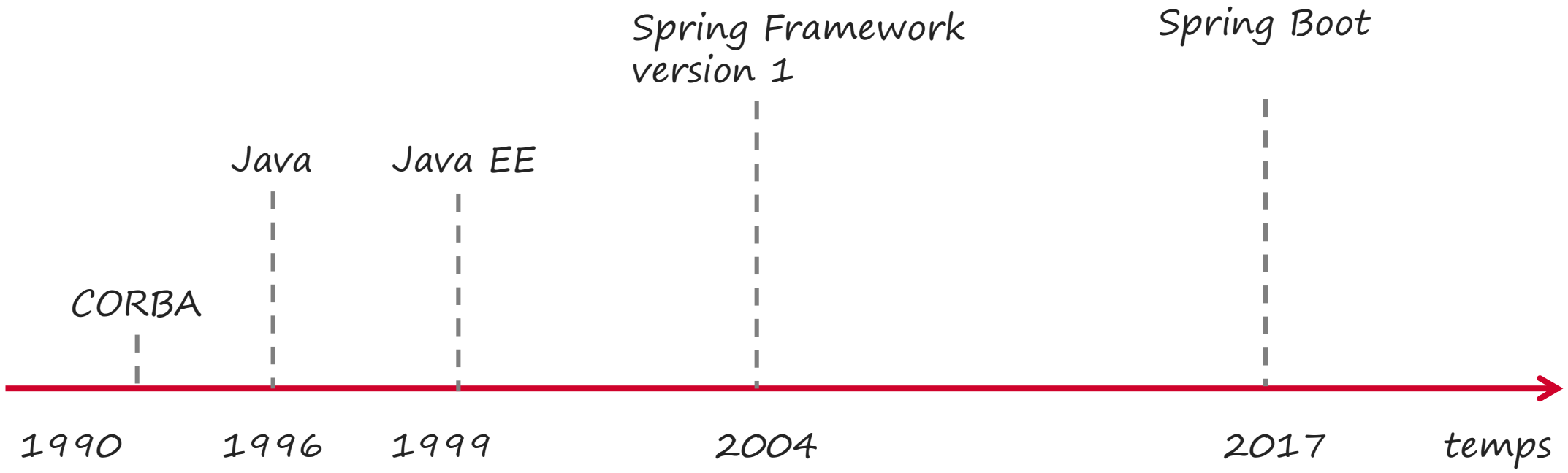
*Réussir la transformation. Ensemble.

Delivering Transformation. Together.

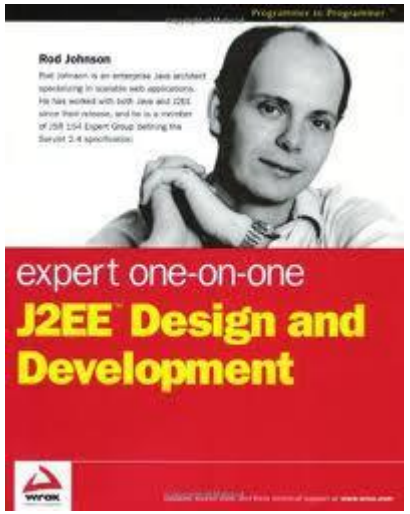
SPRING FRAMEWORK

Introduction

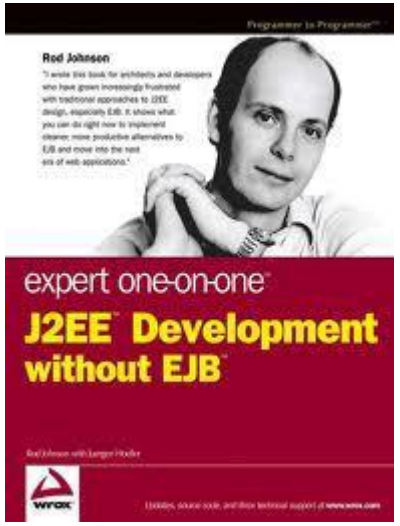
GENÈSE DE SPRING



ROD JOHNSON

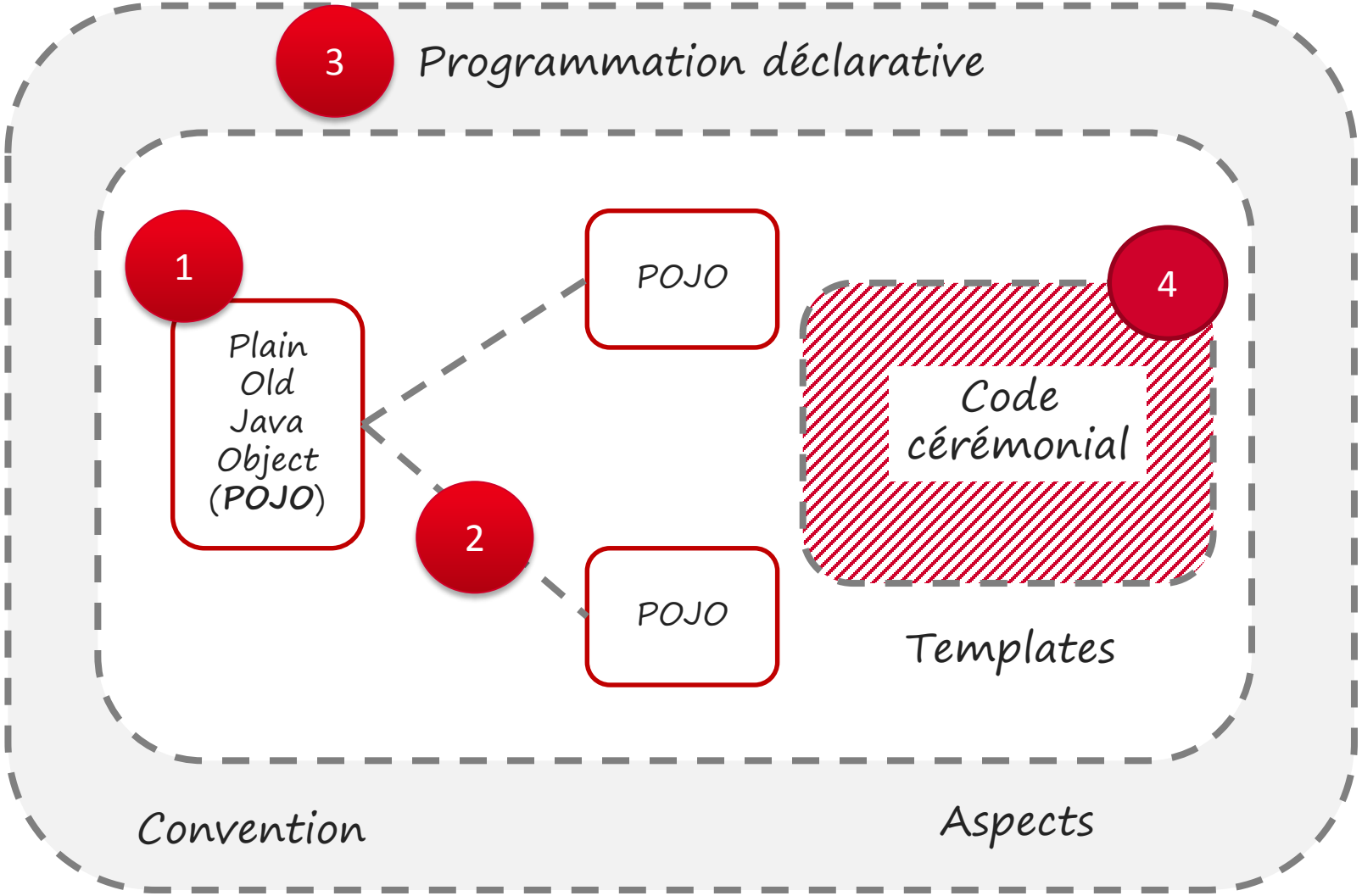


2002

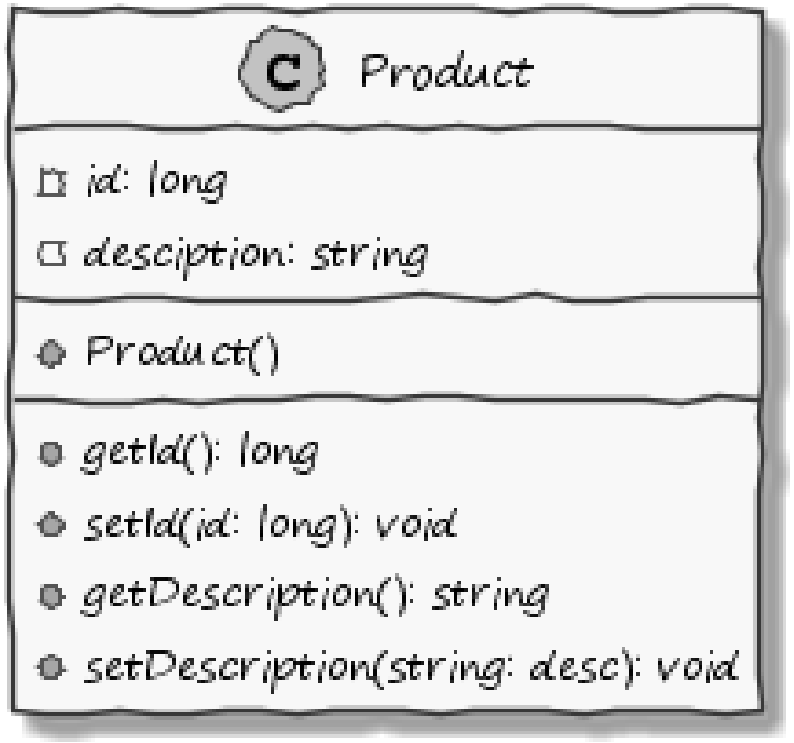


2004

APPROCHE DE SPRING FRAMEWORK



PLAIN OLD JAVA OBJECT (POJO)

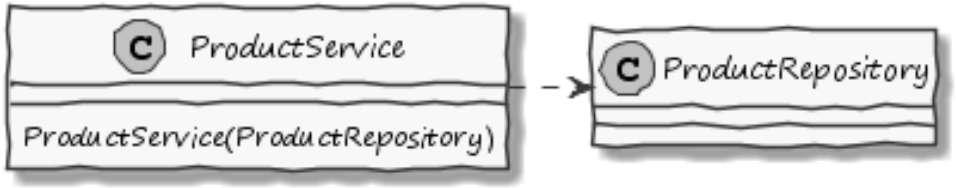



```java
package fr.emn.spring.productapp.domain;

public class Product {
        private Long id;
        private String description;

        public Long getId() {
                return id;
        }
        public void setId(Long id) {
                this.id = id;
        }
        public String getDescription() {
                return description;
        }
        public void setDescription(String description) {
                this.description = description;
        }
}
```

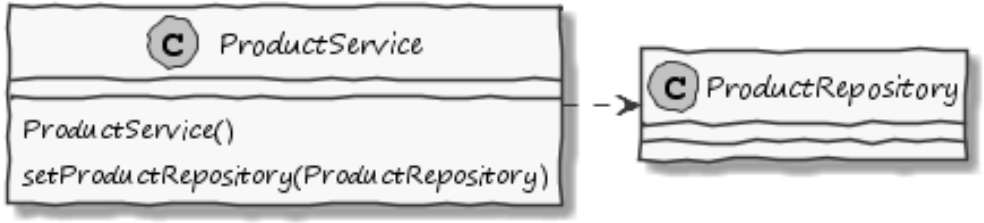
INJECTION DE DÉPENDANCE *(DEPENDENCY INJECTION)*

INJECTION DE DÉPENDANCE

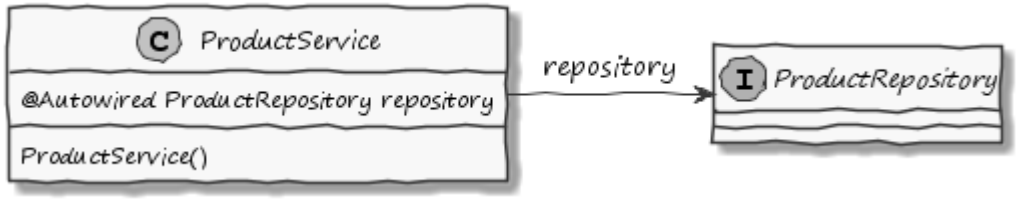
1 Par constructeur



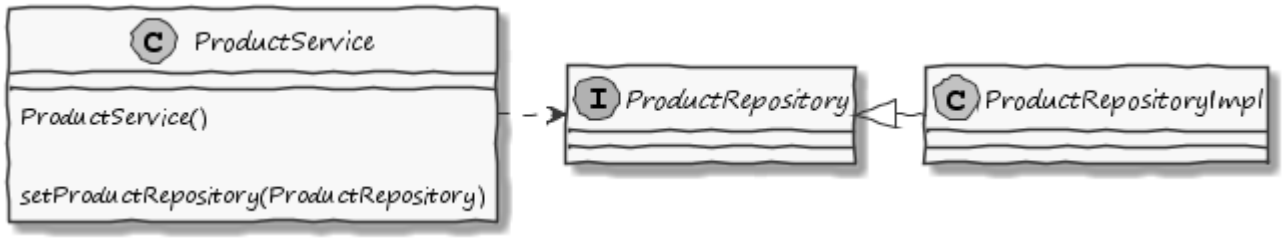
2 Par modificateur



2’ Par champ



3 Par interface



L'INJECTION DE DÉPENDANCE AVEC SPRING

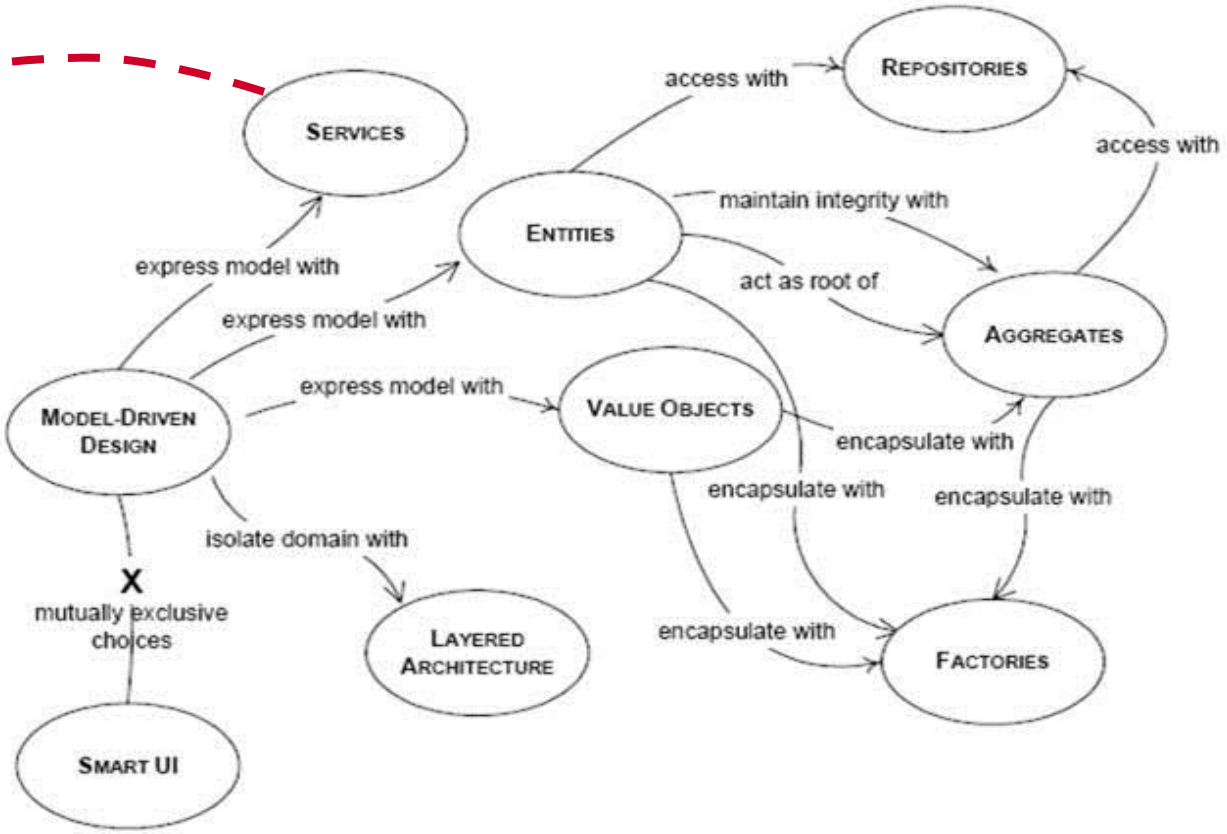
```
@Repository
```

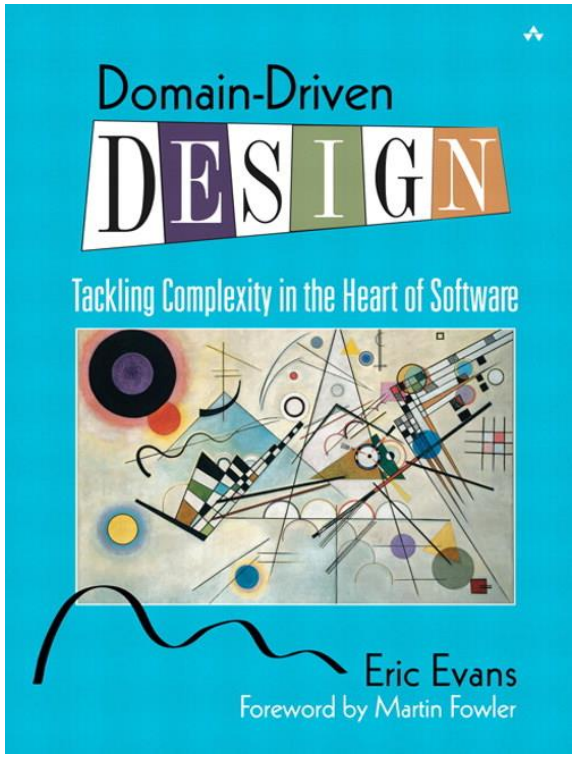
```
@Service
```

```
@Controller
```

```
@Component
```

```
@Autowired
```





INVERSION DE CONTRÔLE

INVERSION DE CONTRÔLE

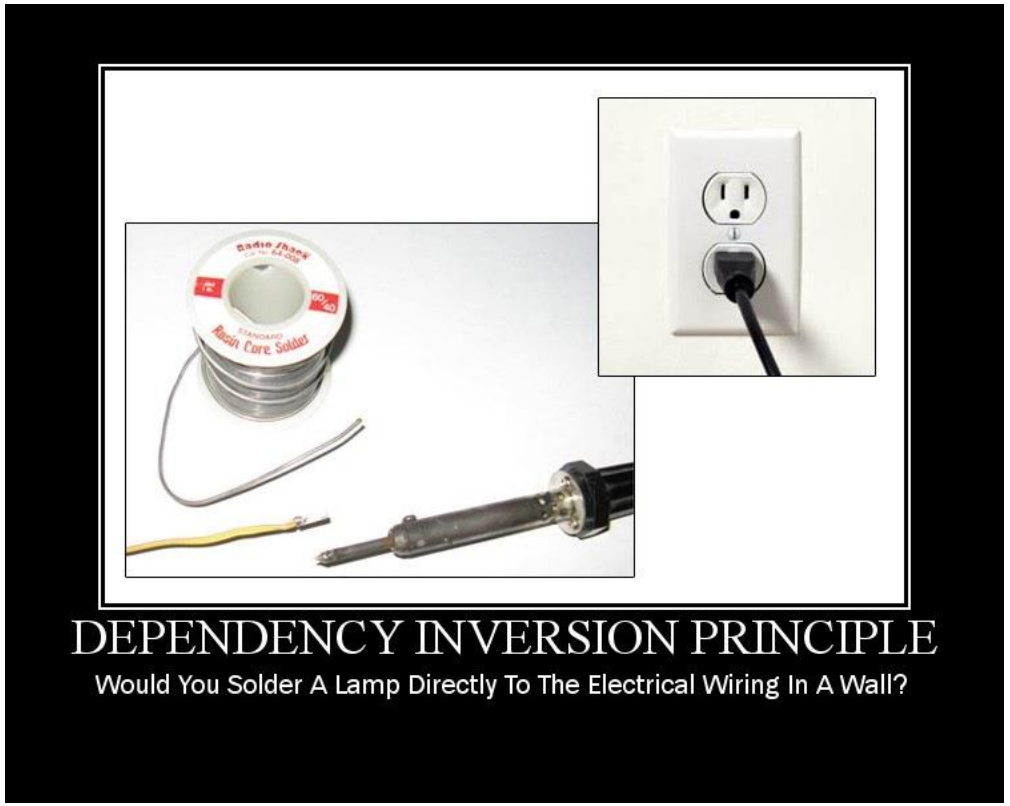
> “ One important characteristic of a **framework** is that the methods defined by the user to tailor the framework will often be called from within the framework itself, rather than from the user's application code. The framework often plays the role of the **main program** in coordinating and sequencing application activity. This **inversion of control** gives frameworks the power to serve as extensible skeletons. The methods supplied by the user tailor the generic algorithms defined in the framework for a particular application.

- Ralph Johnson and Brian Foote

INVERSION DE CONTRÔLE

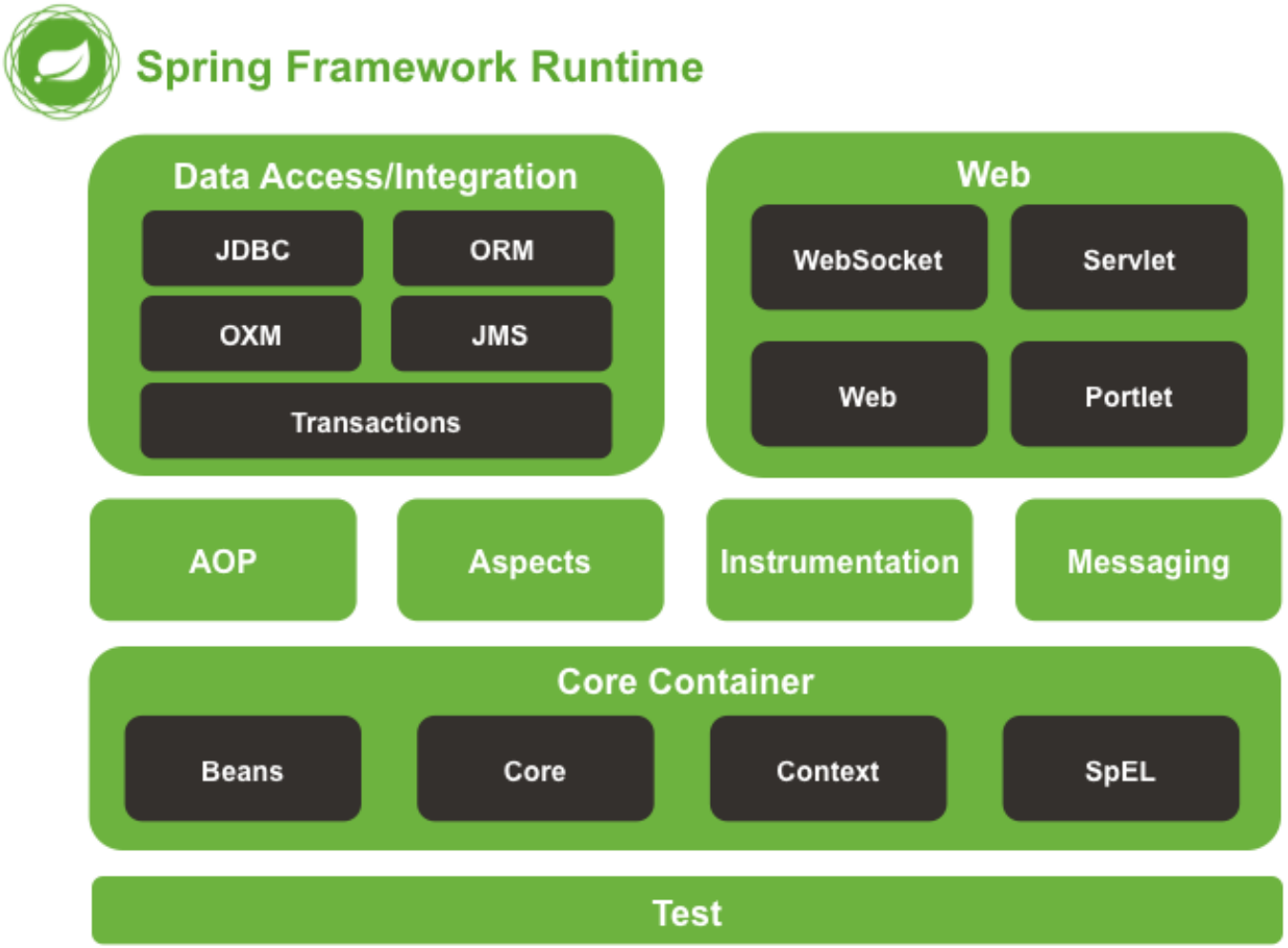
> **Inversion of Control** is a key part of what makes a **framework** different to a **library**. A library is essentially a set of functions that you can call, these days usually organized into classes. Each call does some work and returns control to the client.

- Martin Fowler

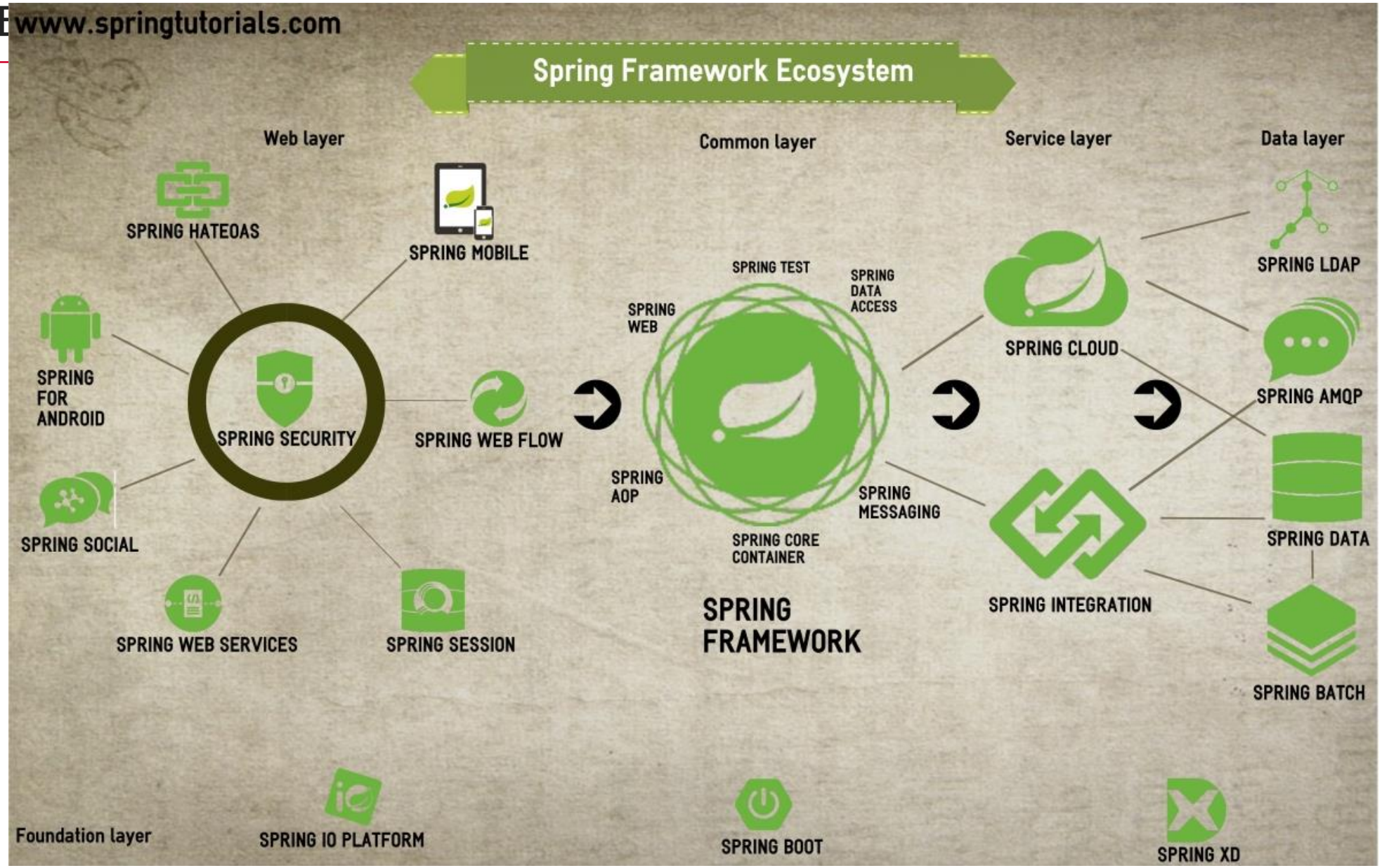


> High level modules should not depend upon low level modules. Both should depend upon abstractions.[…] Abstractions should not depend upon details. Details should depend upon abstractions.
>
> - Robert C Martin (Uncle Bob)

SPRING FRAMEWORK



SPRING PROJE



SPRING FRAMEWORK

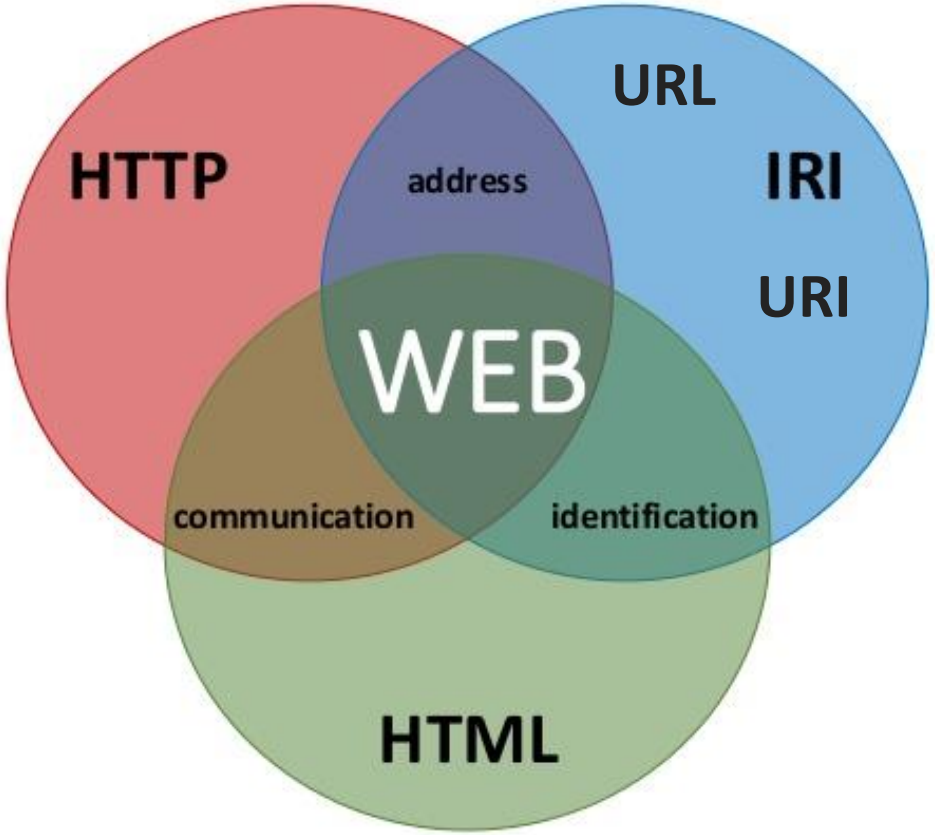
Spring Web

COURTE HISTOIRE DU WEB

Ted Nelson Hypertext **1965**	CGI **1993**	Javascript PHP **1995**	HTTP/1.1 **1996**	HTTP/2 **2015**

1989 Tim Berners-Lee World Wide Web HTML HTTP	**1995** Apache	**1999** J2EE 1.2	**2015** HTML 5

WEB



Source : Fabien Gandon

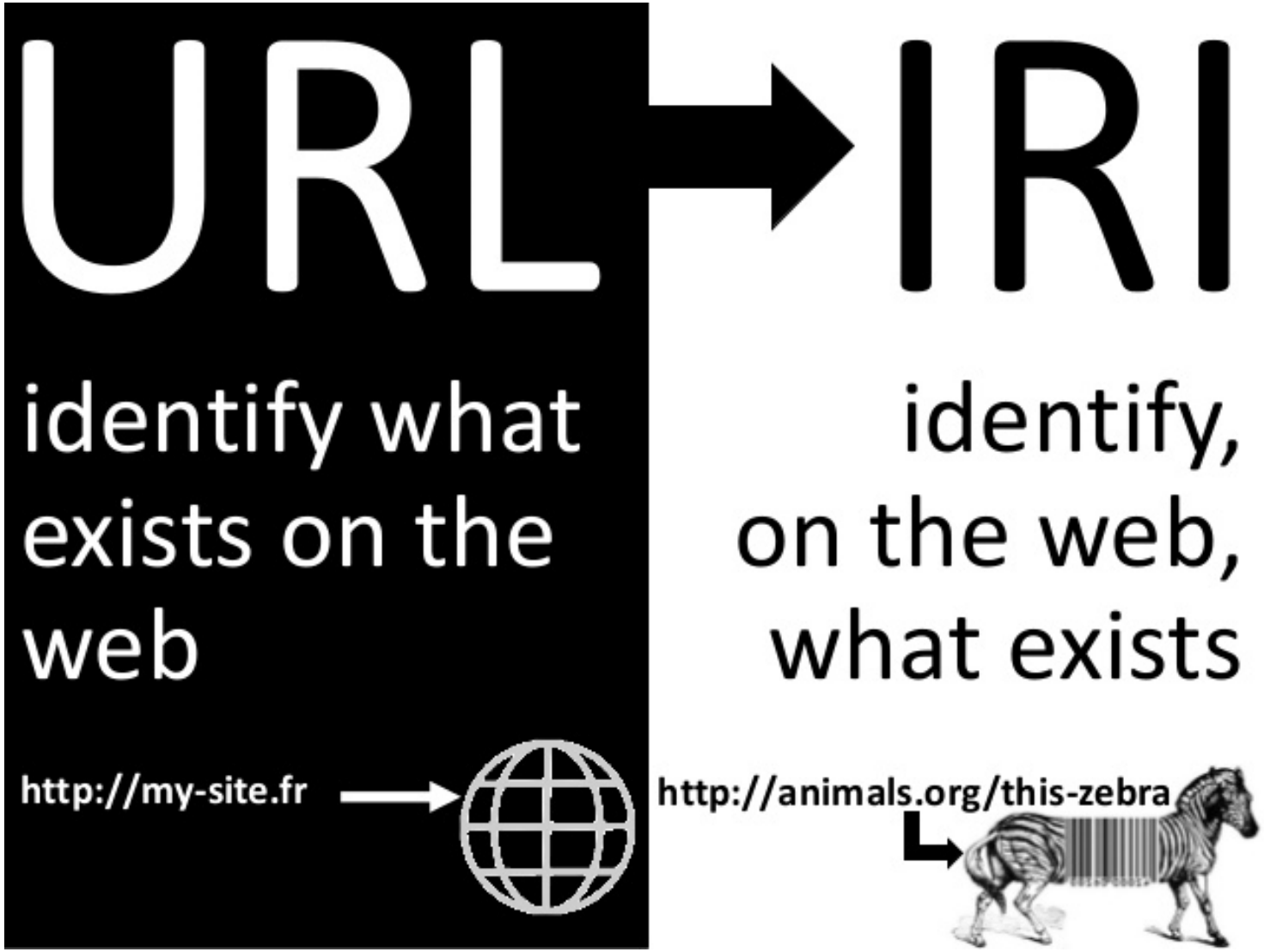
HTML

```
<html>
  <head>
    <title>An Example Page</title>
  </head>
  <body>
    <p>Hello World, this is a very
simple HTML document.</p>
  </body>
</html>
```

HTTP

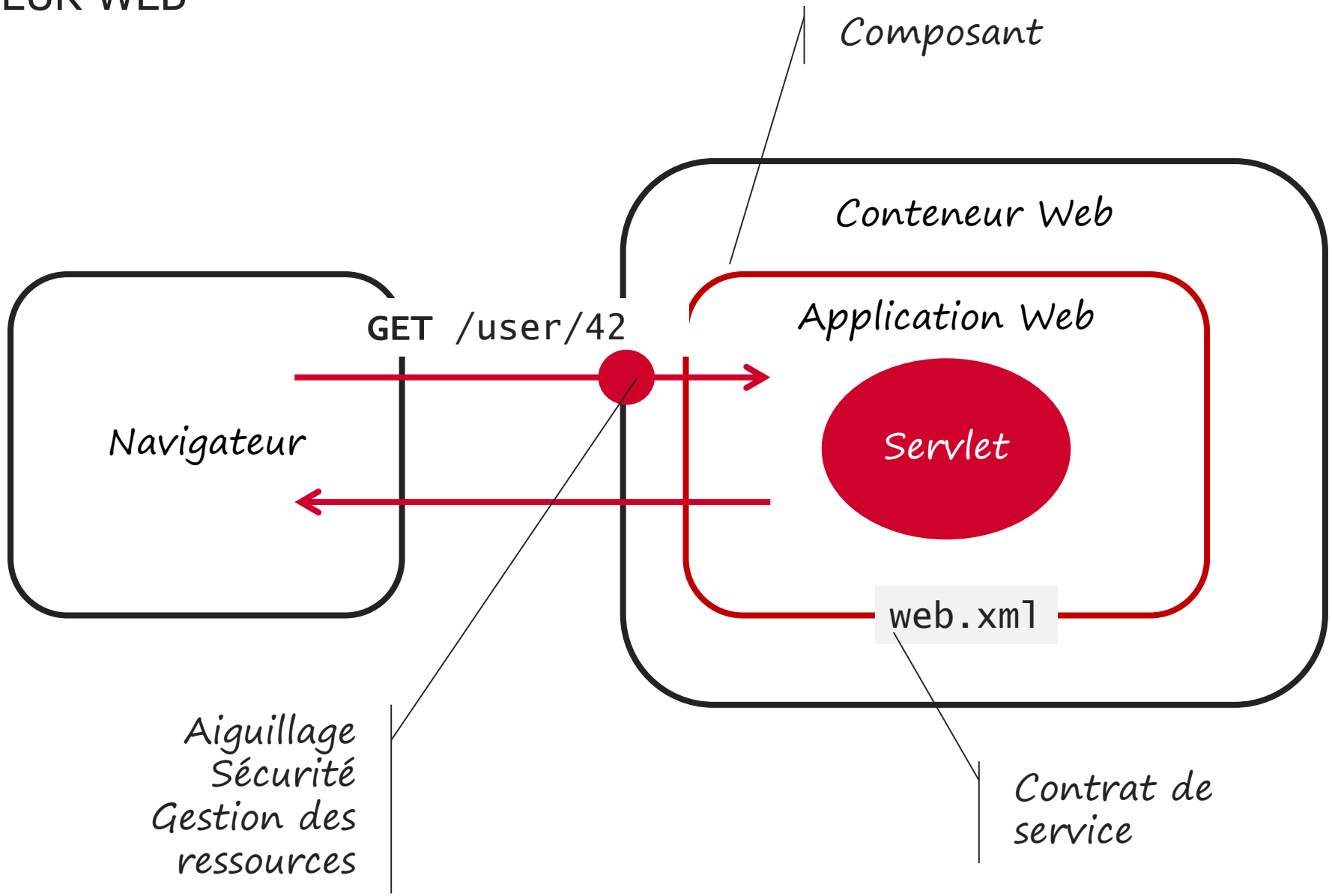
```
GET /index.html HTTP/1.1
Host: www.example.com
```

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/html; charset=UTF-8
Content-Encoding: UTF-8
Content-Length: 138
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
ETag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Connection: close

<html>
  <head>
    <title>An Example Page</title>
  </head>
  <body>
    <p>Hello World, this is a very simple HTML document.</p>
  </body>
</html>
```

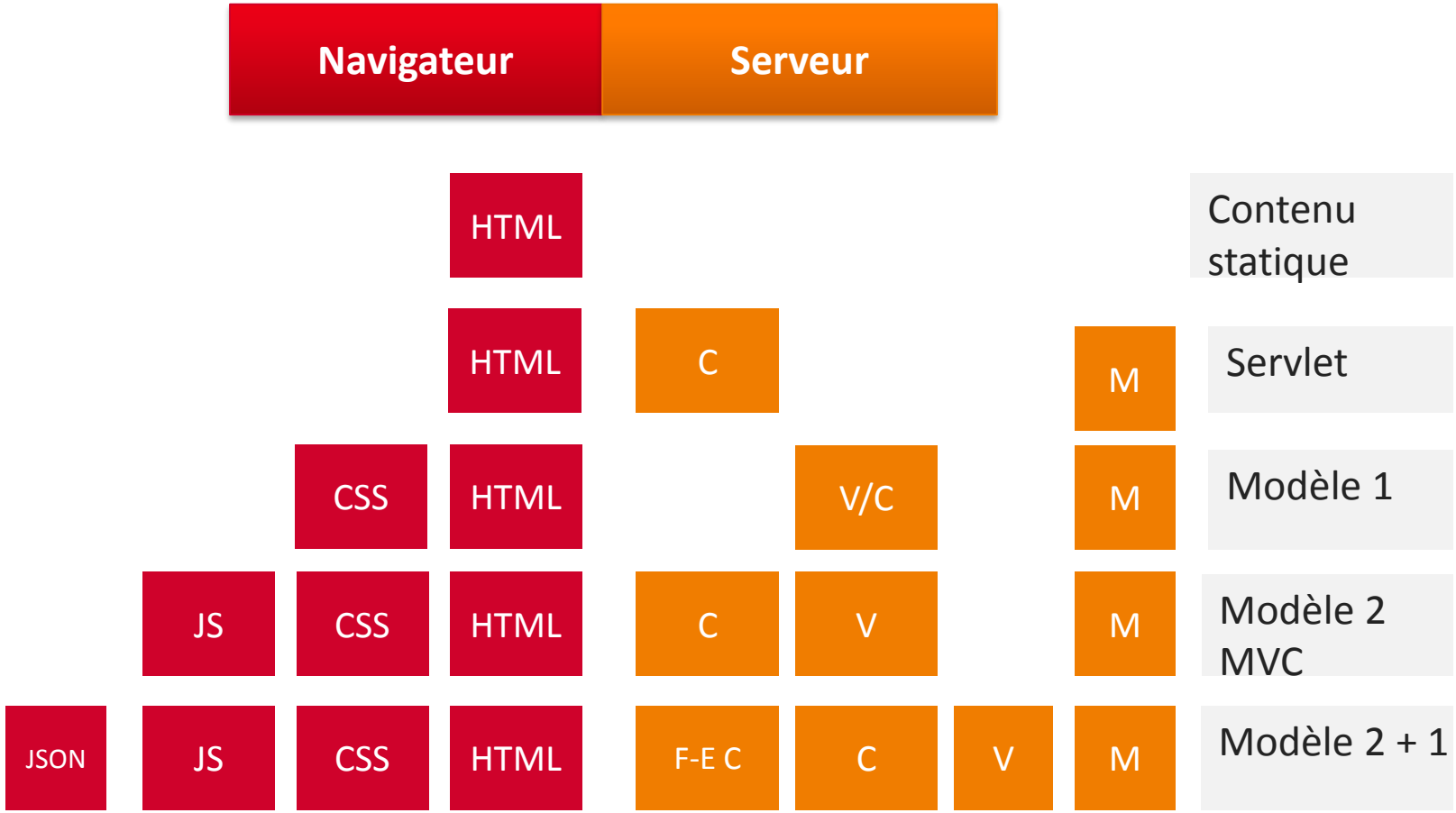


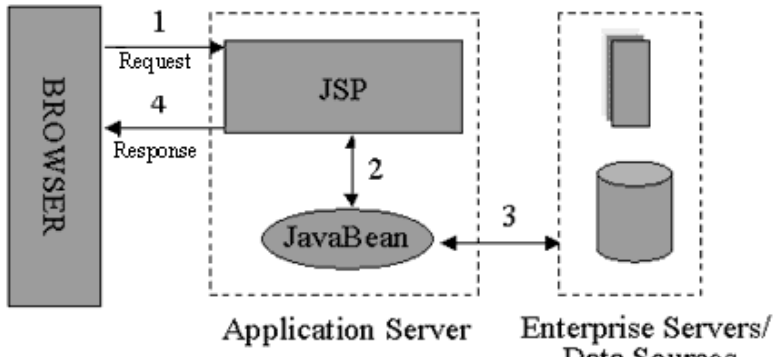
Fabien Gandon

CONTENEUR WEB



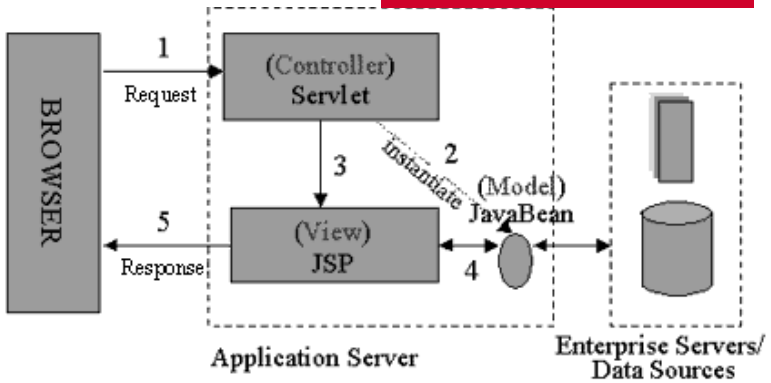
CONCEPTION APPLICATIVE

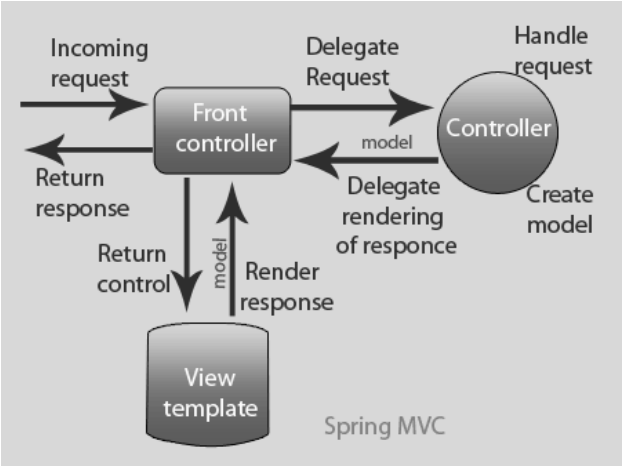




Modèle 1

Modèle 2





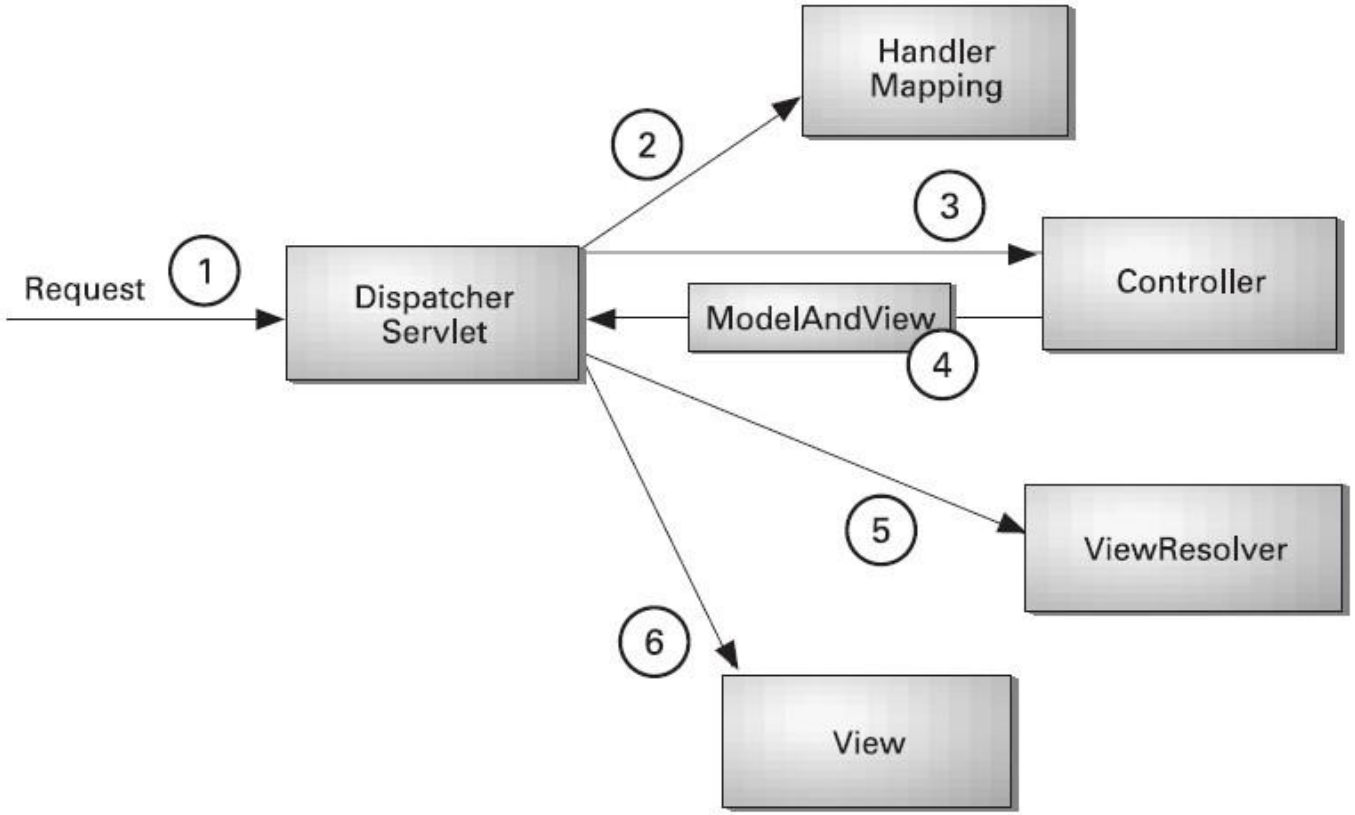
Modèle 2+1

JavaWorld, Understanding JavaServer Pages Model 2 architecture (1999)
Roseindia, Spring MVC framework Introduction (2012)

FRAMEWORK WEB

Matt Raible, Comparing JVM Web Frameworks (2014)

SPRING MVC



http://java4developers.com

CONTRÔLEUR

Controller

@Controller

```
package cours.spring.mvc.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
public class HelloWorldController {


    @RequestMapping ("/")
    @ResponseBody
    public String helloWorld () {
        return "helloWorld";
    }
}
```

@RequestMapping

View

- JSON/REST
- thymeleaf
- Groovy
- Velocity , FreeMarker
- JSP, JSTL
- Script templates (JSR-233)
- XML
- Tiles
- XSLT
- Document views (PDF/Excel)
- JasperReport
- Feed Vews (RSS)

REST

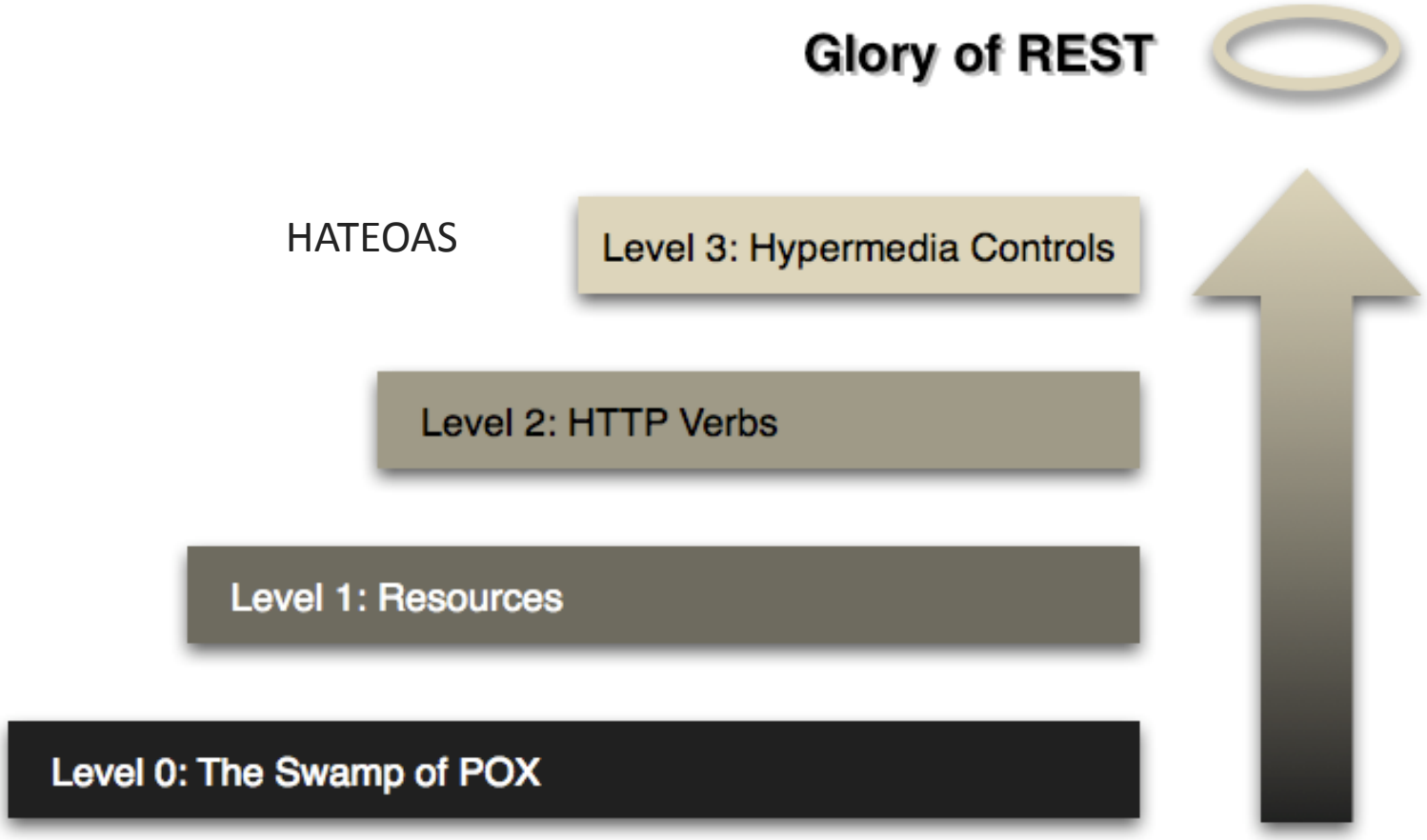
- *Representational State Transfer (*REST)
- Roy Fielding (2000)
- Style de conception conforme à la philosophie de HTTP 1.1
  - HTTP Verbes
  - HTTP Headers

VERBES HTTP

Verbe	Description	Idempotent
GET	Retourne une représentation totale ou partielle de la ressource	Oui
POST	Crée ou modifie la ressource	
PUT	Crée ou modifie une ressource	Oui
PATCH	Modifie une ressource	Oui
DELETE	Supprime une ressource (plus tard)	

Code	Description
1xx	Information
2xx	Succès
3xx	Redirection
4xx	Erreur client
5xx	Erreur serveur

MODÈLE DE MATURITÉ REST



http://martinfowler.com/articles/richardsonMaturityModel.html

SPRING FRAMEWORK

Spring Data

DATA

Source : corporatenews.lu

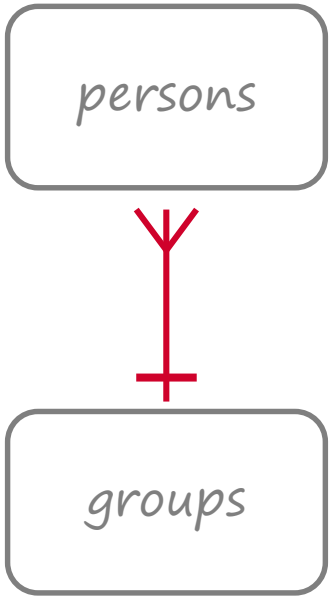
MODÈLE RELATIONNEL

persons

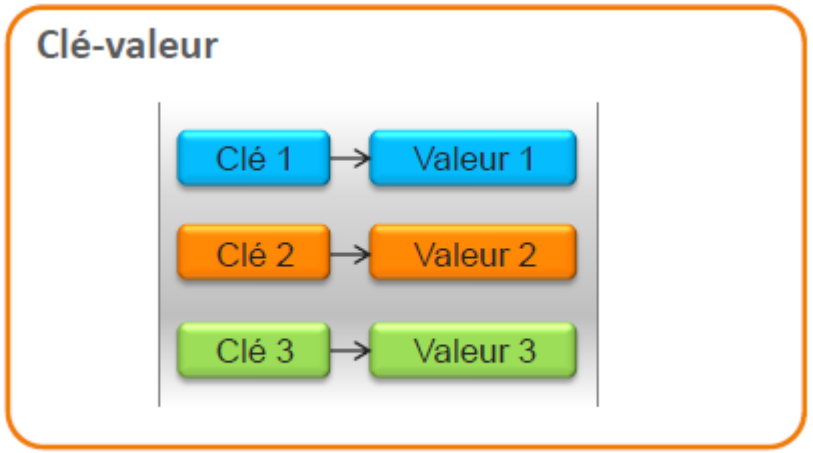
id	*first_name*	*last_name*	*group_id*
101	Rod	Johnson	3
134	Juergen	Hoeller	3

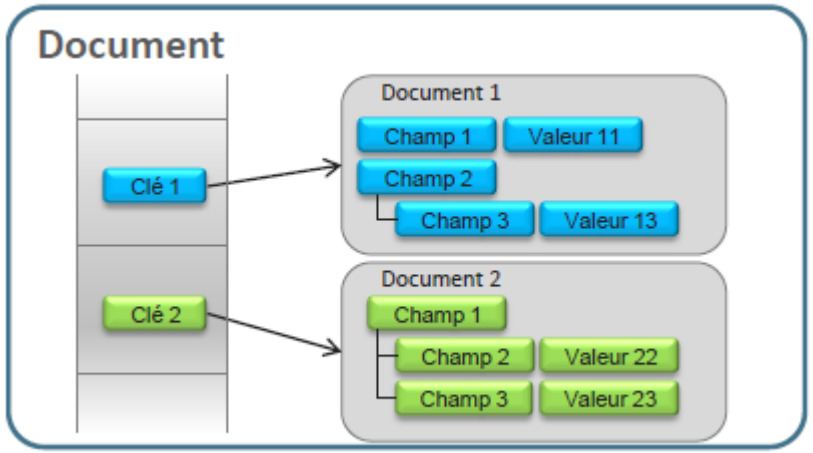
groups

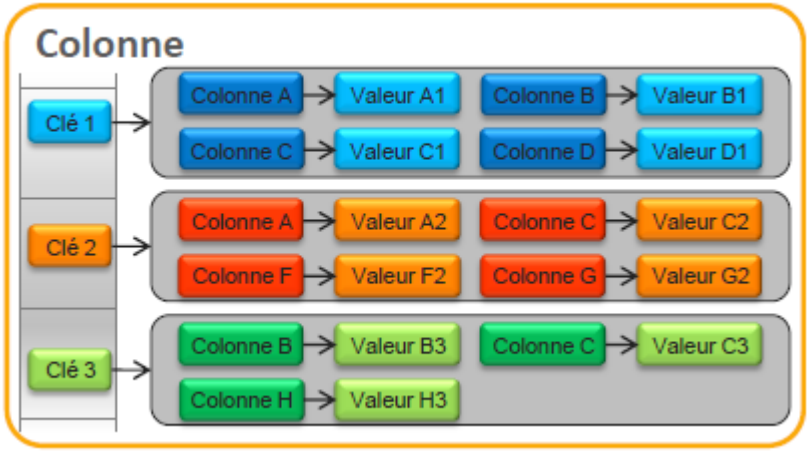
id	*name*
3	Spring Framework Team

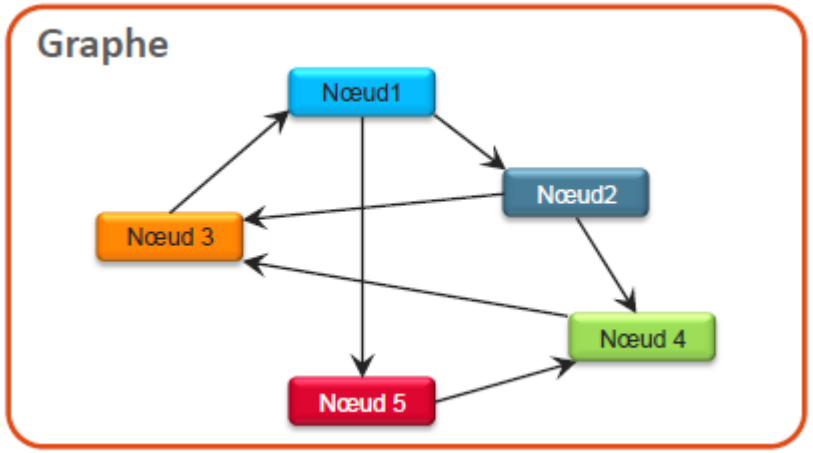


MODÈLES NON RELATIONNELS (NoSQL)

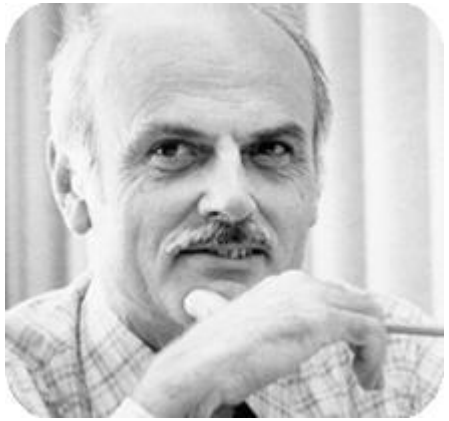








E.F CODD

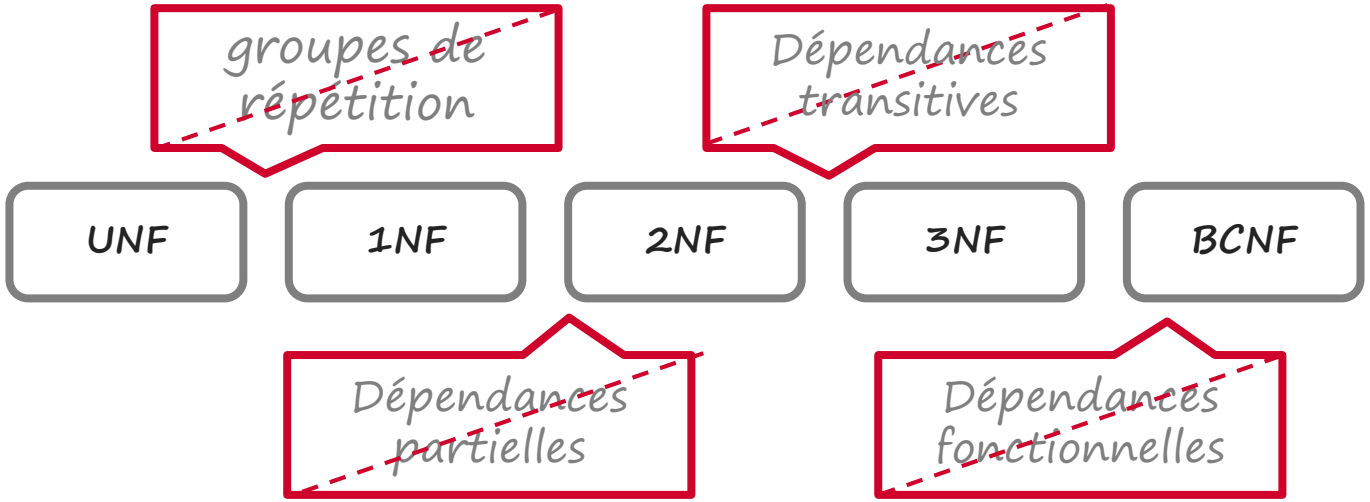


Edgar Frank « Ted » Codd

A Relational Model of Data for Large Shared Data Banks (1970)

Théorème de Codd
Une requête sur une base de données relationnelle est exprimable en calcul relationnel si et seulement si elle l'est en algèbre relationnelle.

FORMES NORMALES



> “The key, the whole key, nothing but the key
>
> - Chris Date

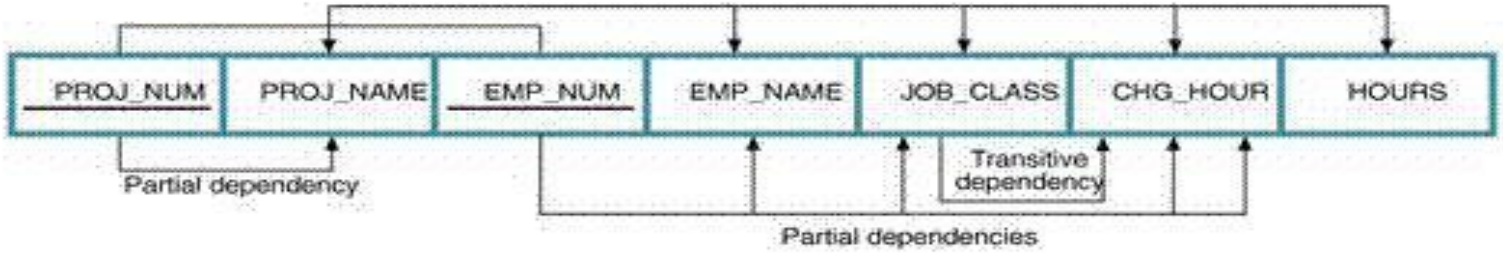


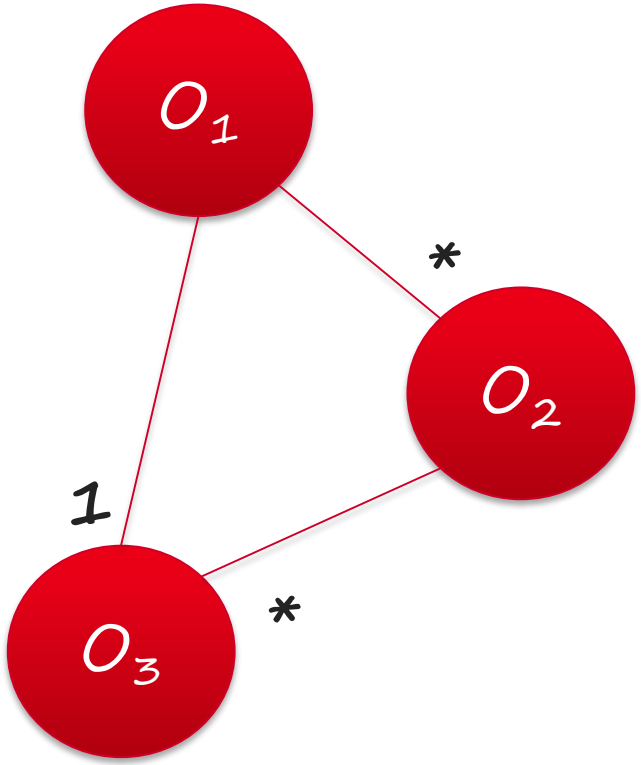
FIGURE 5.4 A DEPENDENCY DIAGRAM: FIRST NORMAL FORM (1NF)

DATA <-> OBJECT

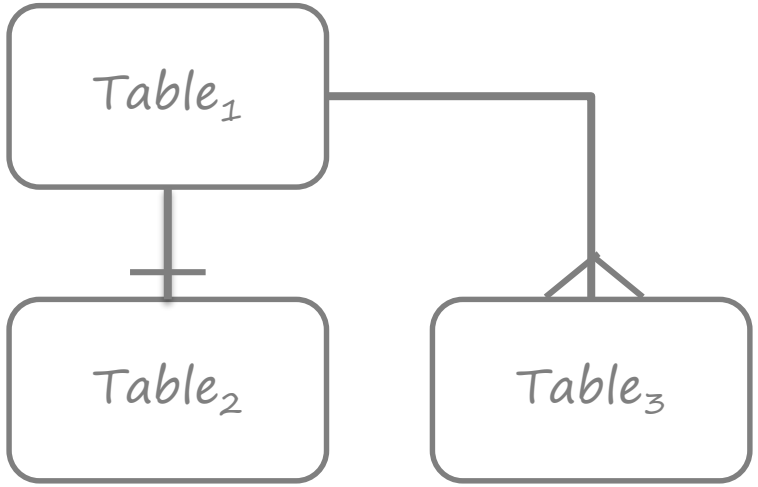
Source : corporatenews.lu

PROBLÈME D'IMPÉDANCE OBJECT-RELATIONNEL

Objets



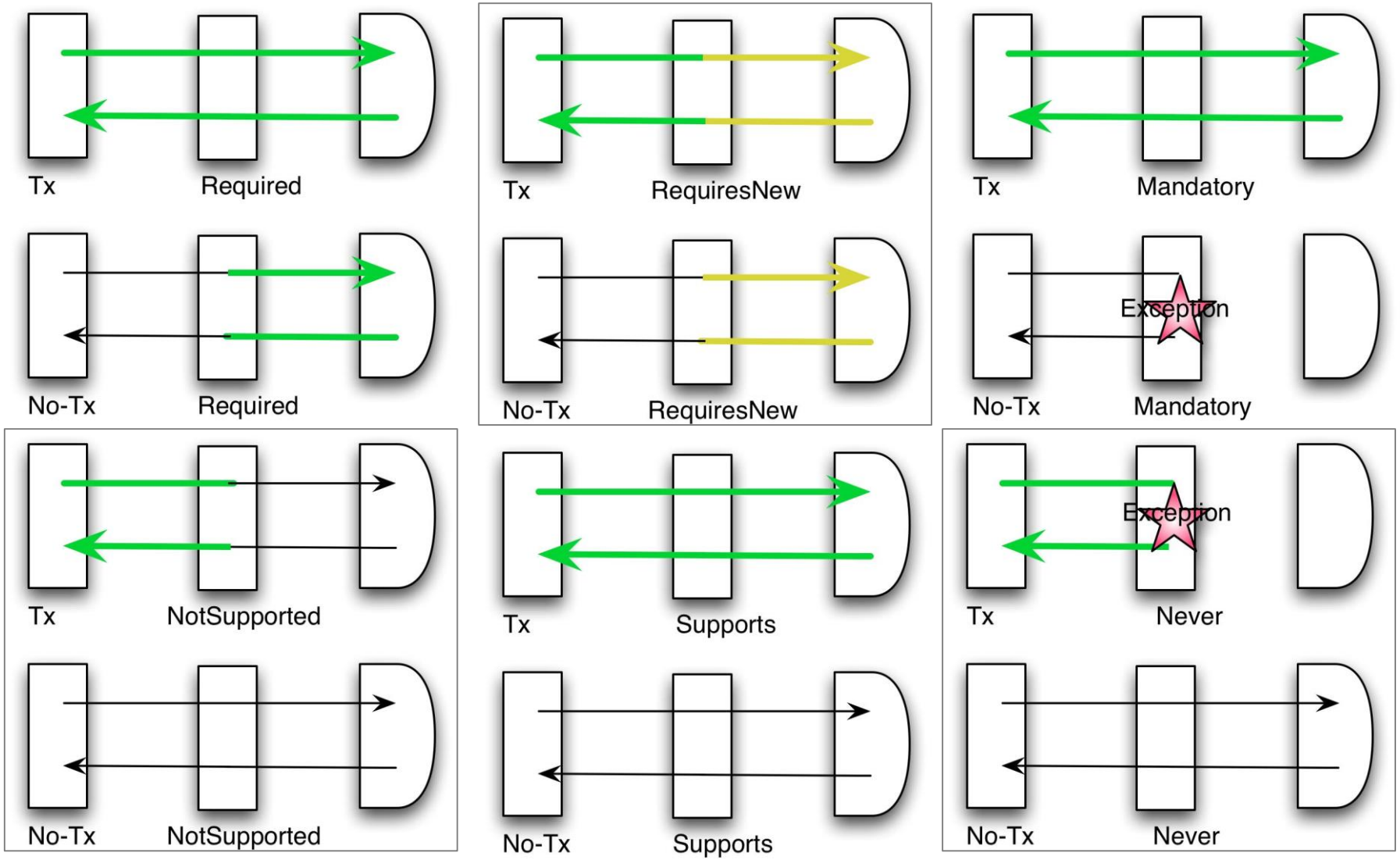
Table



ACIDITÉ DE TRANSACTION

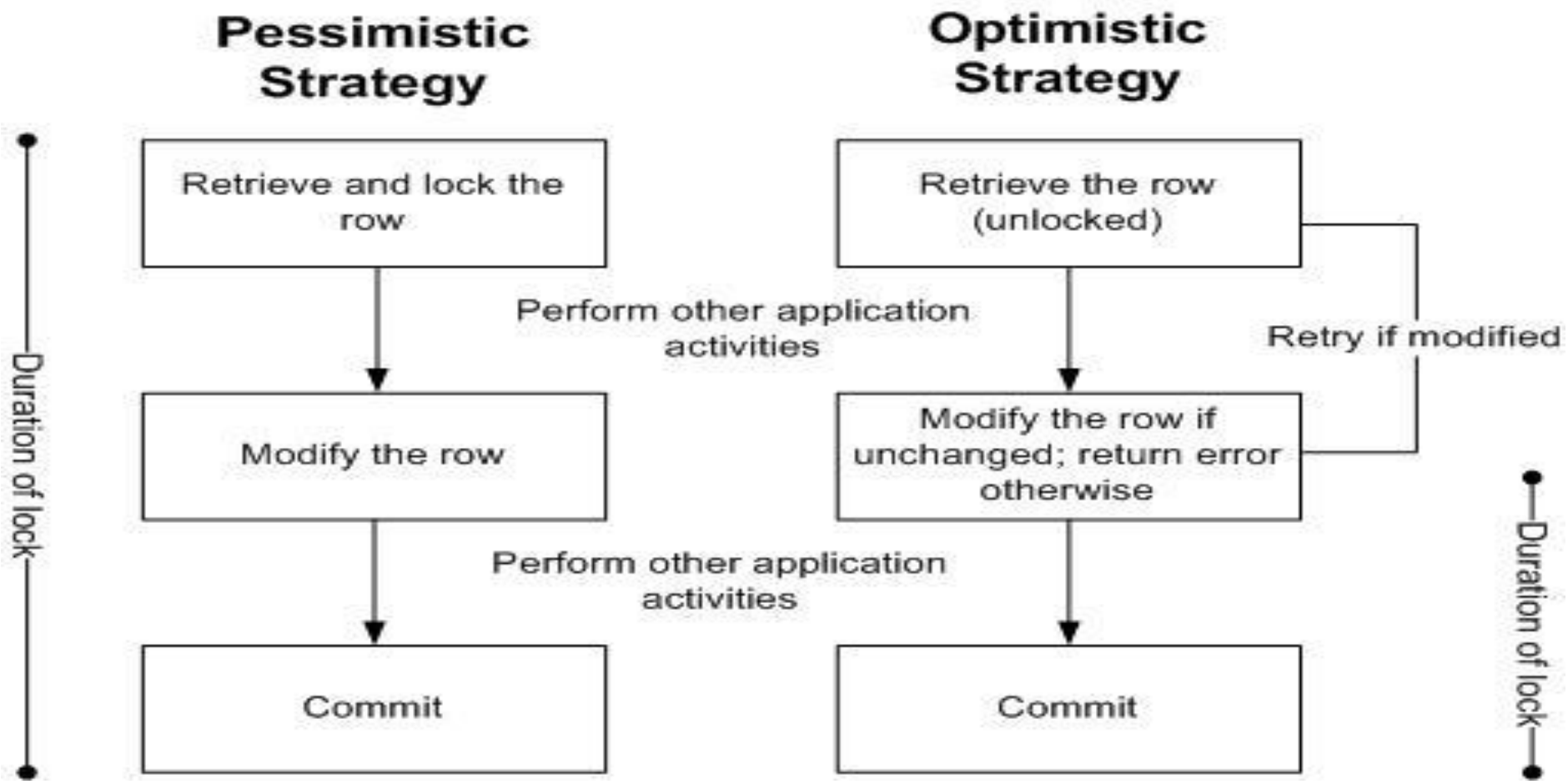
Atomicité	Tout ou rien.
Cohérence	Sauver seulement les données valides.
Isolation	Les transactions n’interfèrent pas.
Durabilité	Les données écrites ne seront pas perdues.

DÉMARCATION DE TRANSACTION



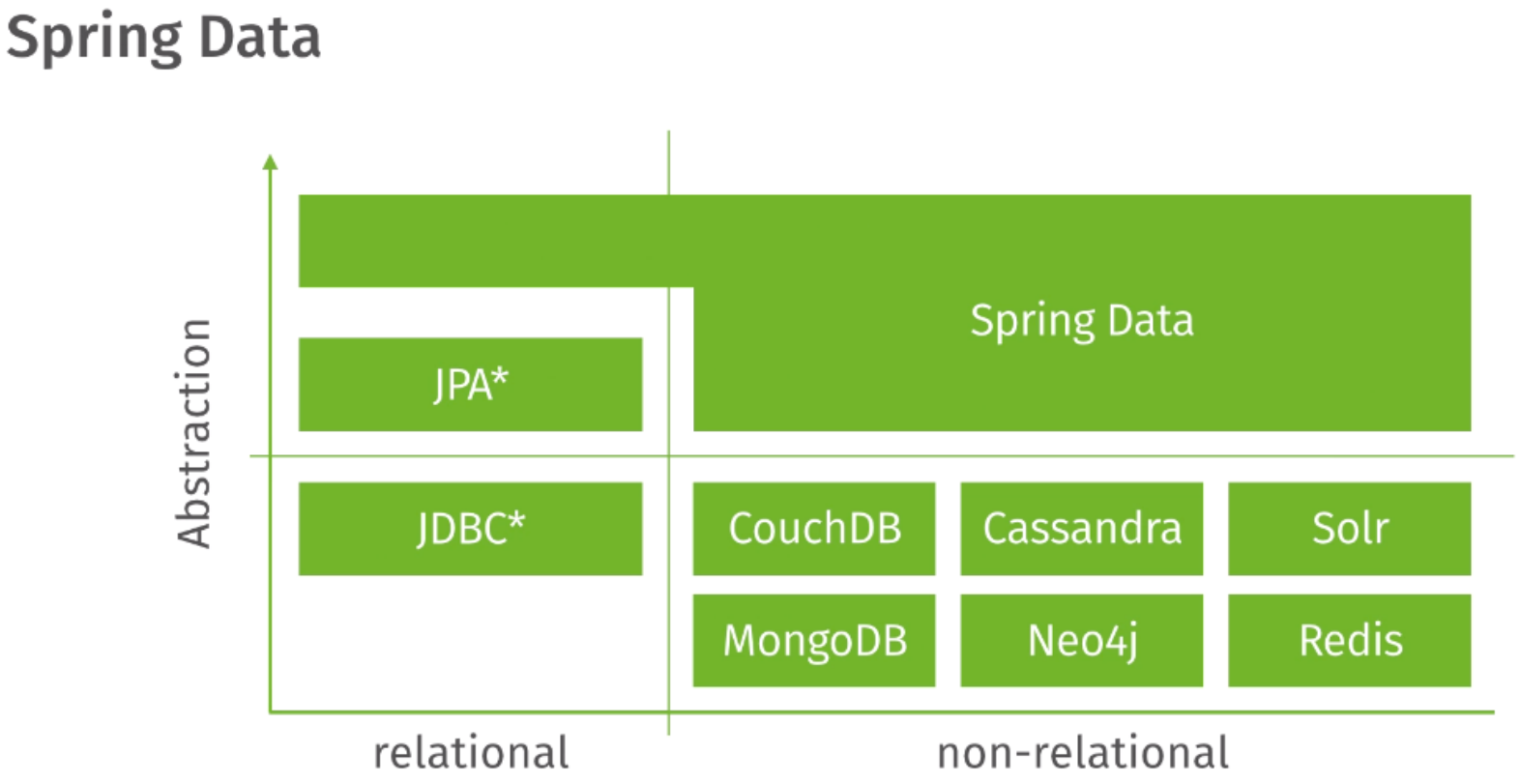
Source : http://techytap.com/

VERROUILLAGE (*LOCKING*)



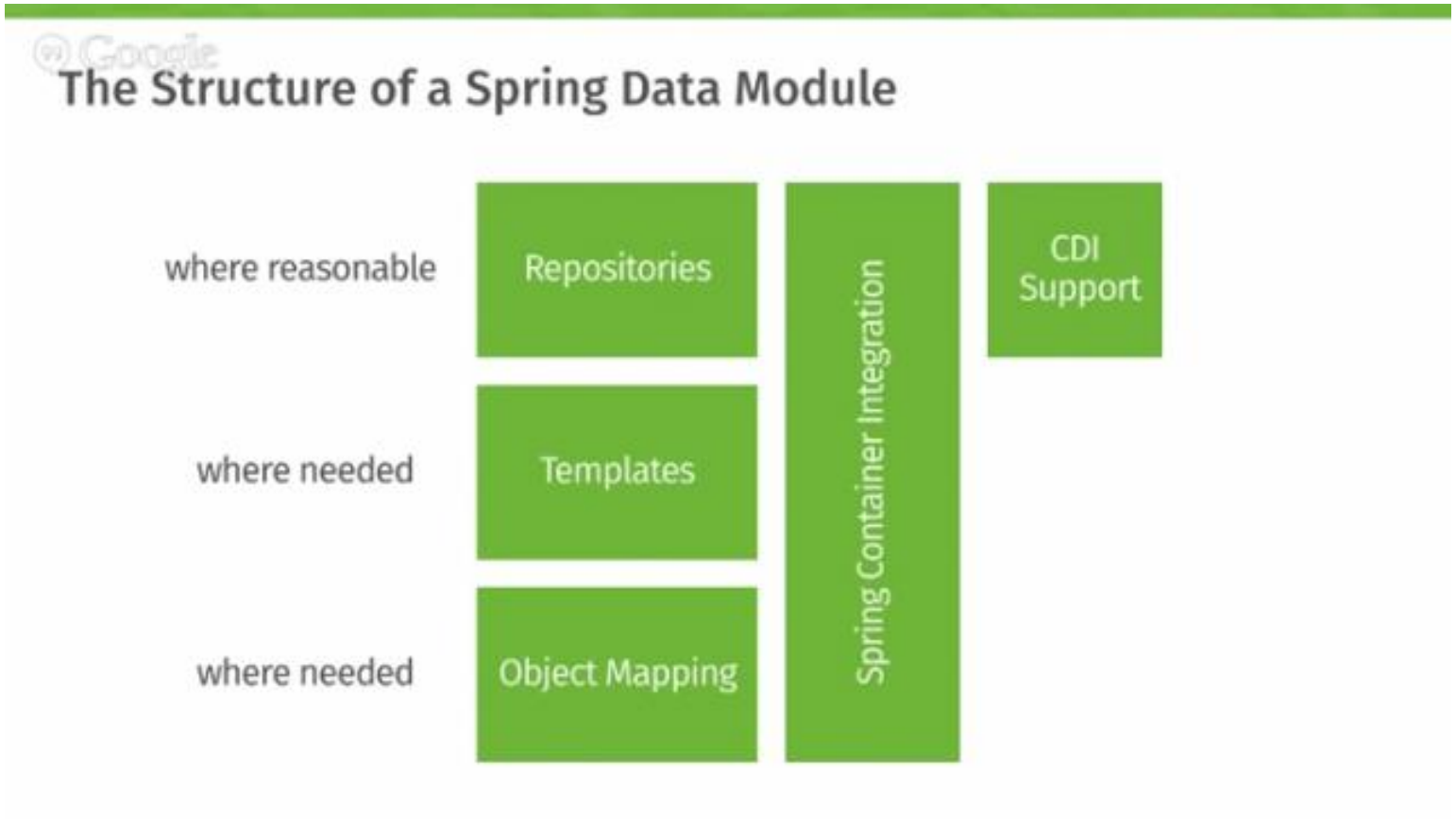
SPRING DATA

PÉRIMÈTRE



Source : Zeroturnaroud

Organisation d'un module Spring Data



Source : Zeroturnaroud

AVANTAGES

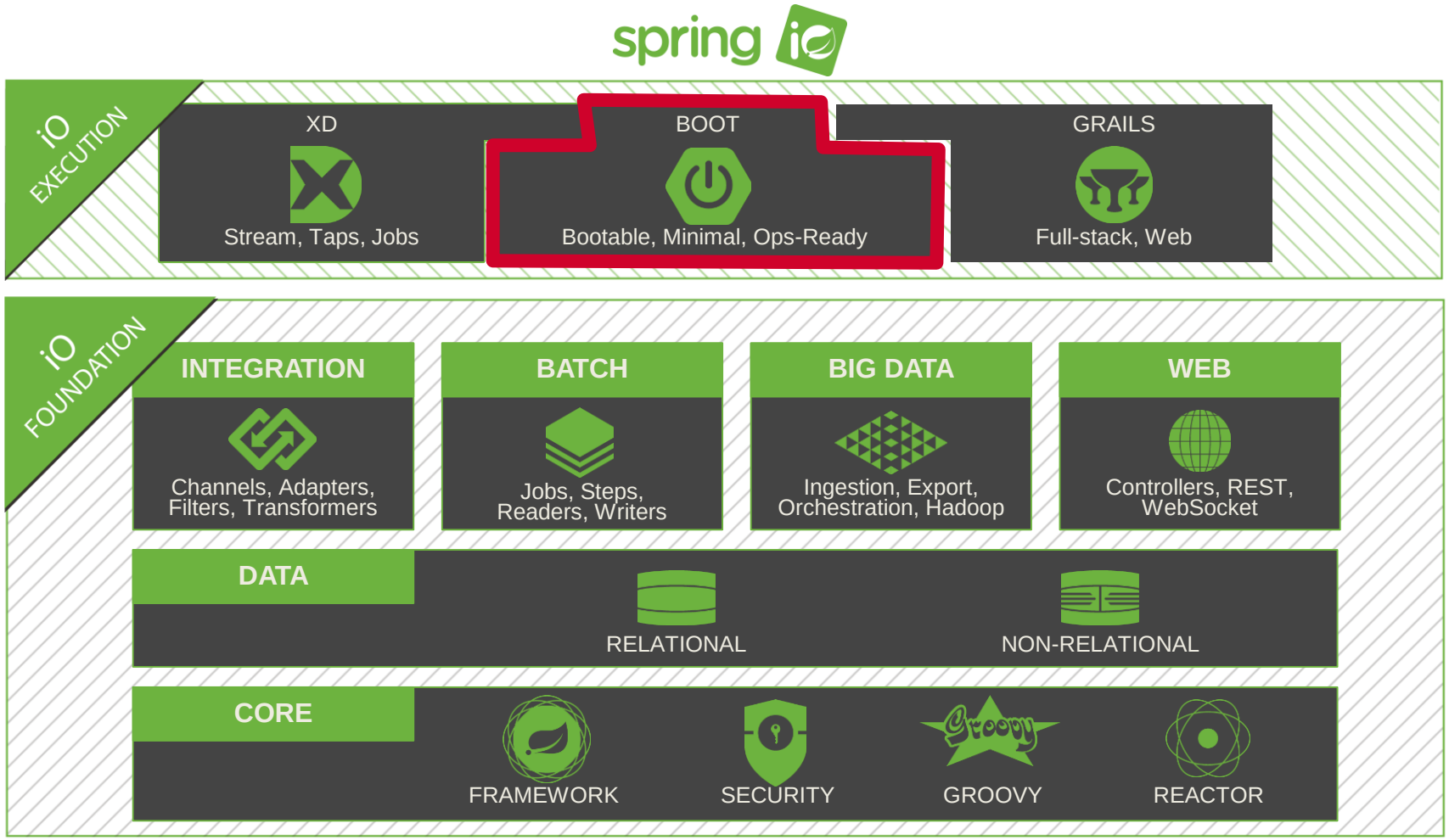
- Indépendance de JEE (JTA, JNDI, EJB)
- Pile de persistance transparente (ORM, JDBC)
- Traduction des exceptions venant du driver JDBC de la base de données
- Gestion des ressources transparente
- Simplicité de l'API
- Modèle transaction plus souple et plus riche que JTA : Rollback rules, advices

Mais

- Pas de transaction globale

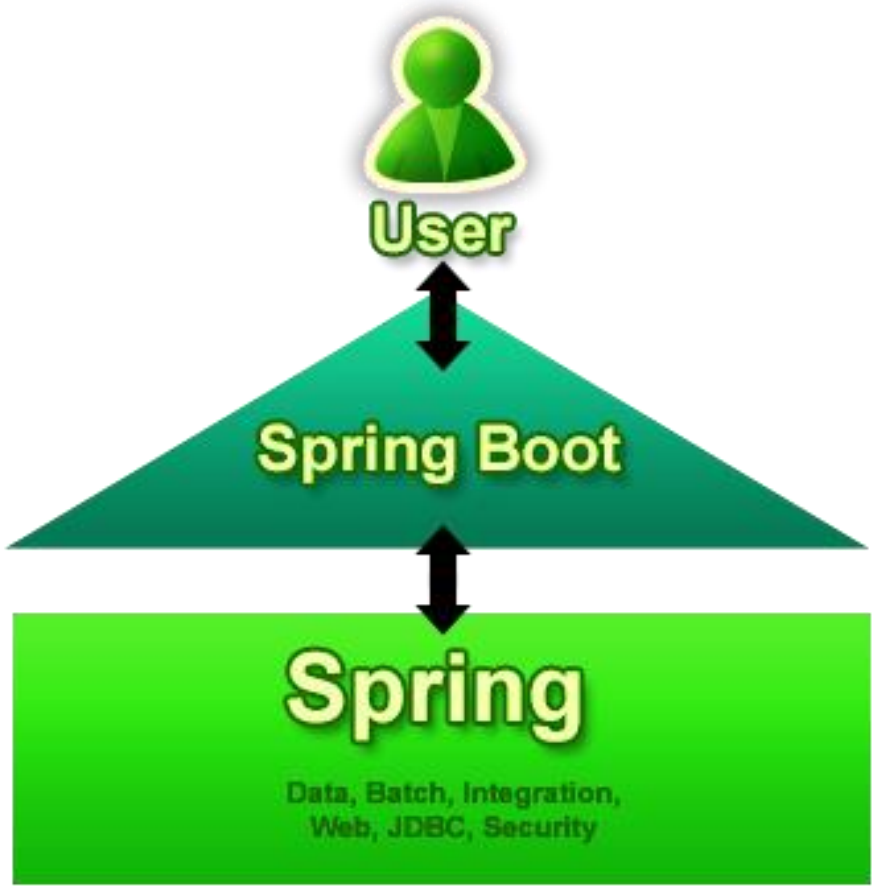
SPRING FRAMEWORK

Spring Boot


spring io
iO EXECUTION
XD
Stream, Taps, Jobs
BOOT
Bootable, Minimal, Ops-Ready
GRAILS
Full-stack, Web
iO FOUNDATION
INTEGRATION
Channels, Adapters, Filters, Transformers
BATCH
Jobs, Steps, Readers, Writers
BIG DATA
Ingestion, Export, Orchestration, Hadoop
WEB
Controllers, REST, WebSocket
DATA
RELATIONAL
NON-RELATIONAL
CORE
FRAMEWORK
SECURITY
GROOVY
REACTOR

SPRING BOOT

- Point d'entrée
- Démarre rapidement avec Spring
- Suggère l'opinion de Spring sur la façon d'utiliser Spring Framework
- Adaptable
- Exigences non-fonctionnelles (sécurité, métriques, …)
- Sans génération de code, sans configuration XML



Source : Phil Webb, Spring Boot – Simplifying Spring for Everyone (2013)

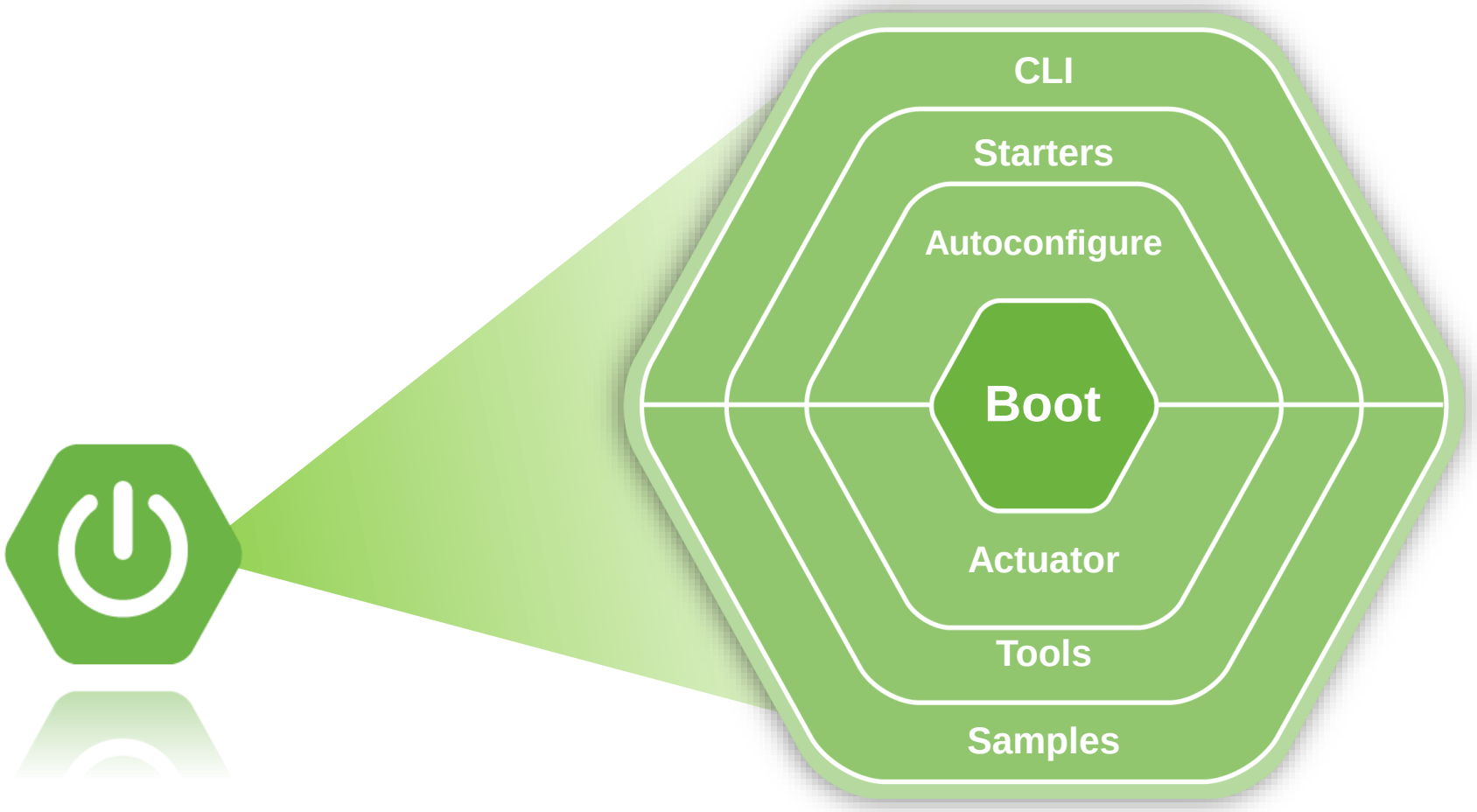
SPRING BOOT *N'EST PAS*

- Un outil de prototypage
- Seulement pour les applications avec conteneur embarqué
- Une sous-expérience de Spring
- Seulement pour les débutants en Spring

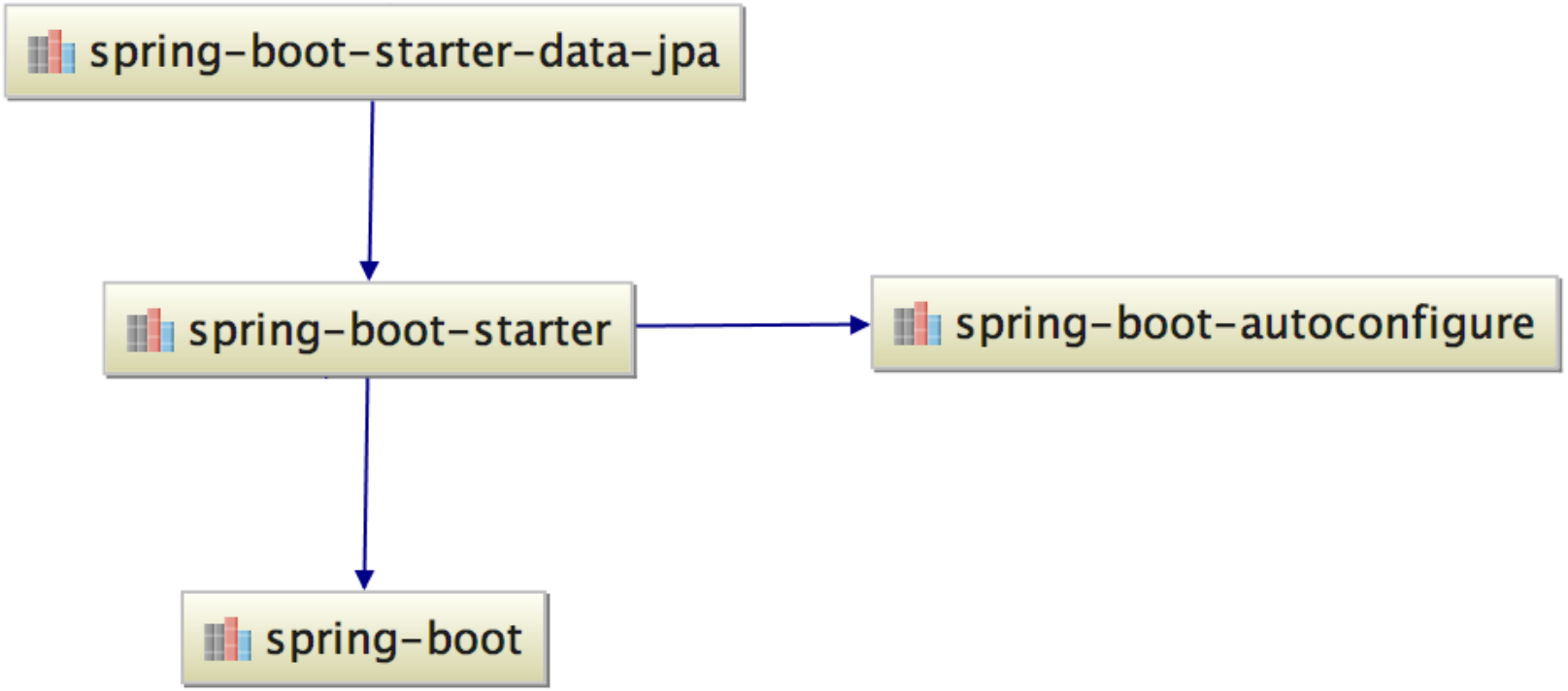
“Spring Boot lets you pair-program with the Spring team.

- Josh Long

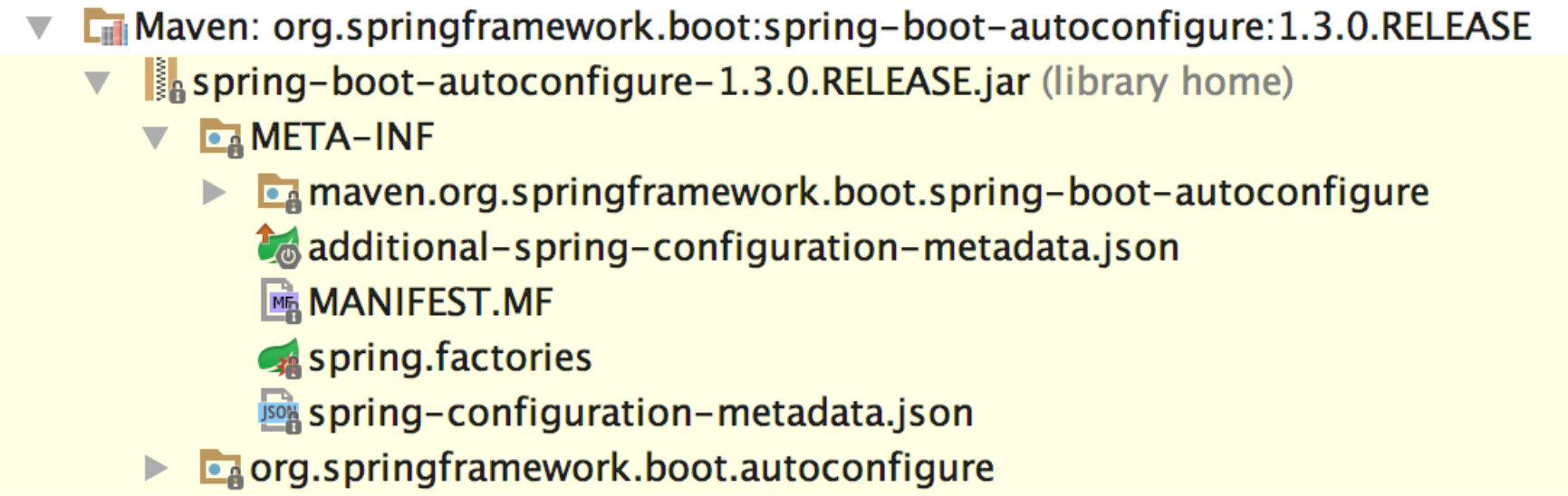
SPRING BOOT MODULES



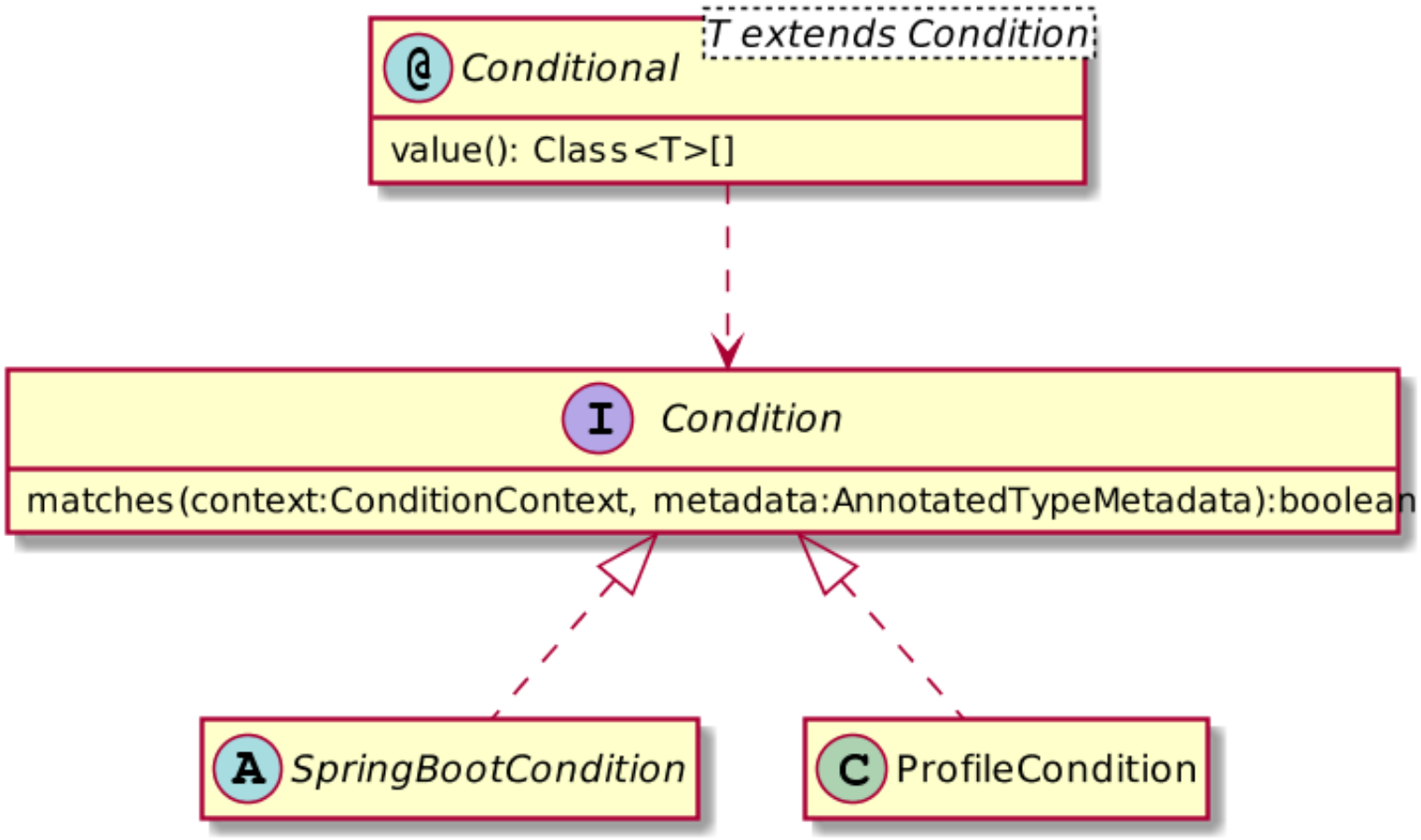
DÉPENDANCES



SPRING-BOOT-AUTOCONFIGURE.JAR



```java
@Configuration
@ConditionalOnBean(DataSource.class)
@ConditionalOnClass(JpaRepository.class)
@ConditionalOnMissingBean({JpaRepositoryFactoryBean.class,
  JpaRepositoryConfigExtension.class })
@ConditionalOnProperty(
  prefix = "spring.data.jpa.repositories",
  name = "enabled", havingValue = "true",
  matchIfMissing = true)
@Import(JpaRepositoriesAutoConfigureRegistrar.class)
@AutoConfigureAfter(HibernateJpaAutoConfiguration.class)
public class JpaRepositoriesAutoConfiguration {}
```


T extends Condition
Conditional
value(): Class<T>[]
Condition
matches(context:ConditionContext, metadata:AnnotatedTypeMetadata):boolean
SpringBootCondition
ProfileCondition

Delivering Transformation. Together.