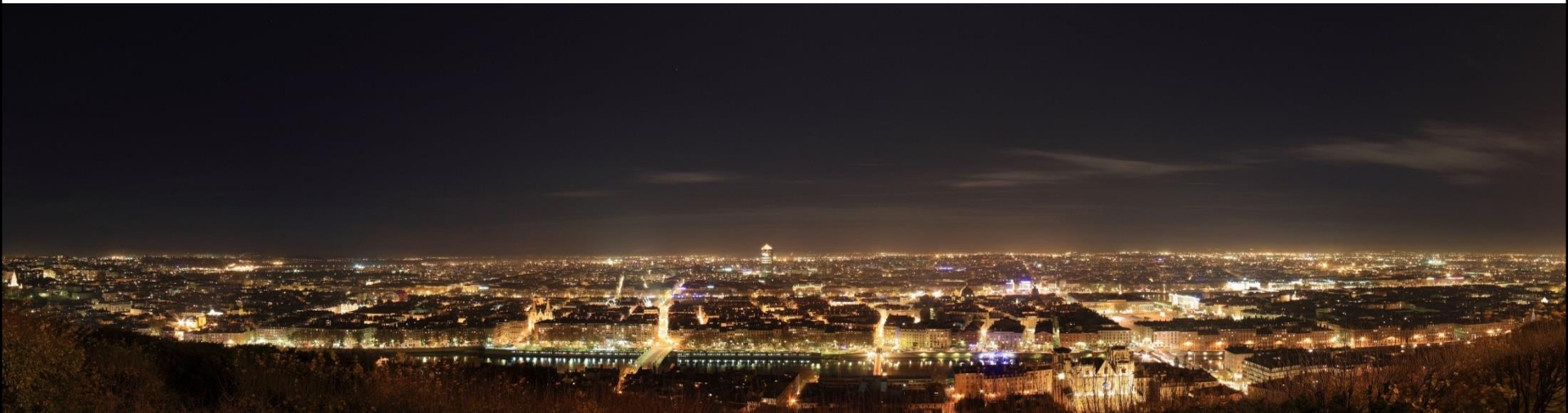


URBANISATION DES SYSTÈMES D'INFORMATION

UNIVERSITÉ LYON 1

26 MARS 2024



Présentation

WHO I AM ?

Julien VILLANTI (julien.villanti@worldline.com) [@JulienVillanti](https://www.linkedin.com/in/JulienVillanti)

[#SoftwareCraftsmanship](#) [#SpringFanboy](#) [#IntellijFandboy](#)

► **Unité : Mobility & e-Transactional Services** (pôle : *Tech. Direction & Innovation*)

► Fonctions

- Membre de la Direction Technique de l'unité MTS France depuis 2023 (équipe *Transversal & Tooling*)
- CoDirecteur Technique du pôle depuis 2020 / Architecte logiciel depuis 2015 / Senior Expert (Expert technique depuis 2012)
- Responsable d'application de 2009 à 2011 / Lead Dev de 2011 à 2012

► Missions

- Accompagnement technique de l'entité (orientation des choix technique, outillage, alignement de la stratégie, etc.)
- Architecture, Industrialisation, Bonnes pratiques, Support, Formations interne
- Appel d'offre, Recrutement, Accompagnement Lead Developer, etc.

► **Formation : Master 2 TI - Technologie de l'Information - Lyon 1**

Présentation

WHO I AM ?

Lionel VENET (lionel.venet@worldline.com)

► **Unité** : **Mobility & e-Transactional Services** (*pôle* : **Government Services Payment**)

► Fonctions

- Consultant fonctionnel depuis 2020
- Responsable d'application confirmé depuis 2013

► Missions

- Ateliers clients, Cadrage fonctionnel, Formations, Rédaction de spécifications métier
- Conception, Bonnes pratiques, Support
- Appel d'offre, etc.

► **Formation** : **Master 2 TI - Technologie de l'Information / Licence Pro - Lyon 1**

Who we are

GLOBAL PRESENCE WITH A STRONG LOCAL EXPERTISE

Worldline is a global leader in digital payments and transactional services.

We are at the forefront of the digital revolution that is shaping new ways of paying, living, doing business and building relationships.



€4.4 BN

Financial revenue (2022)

18,000+ WORLDLINERS

IN **40+** COUNTRIES/TERRITORIES

COMMERCIAL PRESENCE IN **170+** COUNTRIES

Dedicated global business lines

FOR SPECIALISED SOLUTIONS



MERCHANT SERVICES

Energising commerce with advanced payment services

c. €3.0 BN
(c. 70%)

1.25 M+
merchants served

c. 22 BN
acceptance and acquiring transactions per year



FINANCIAL SERVICES

Engineering the most trusted and advanced payment processing solutions

c. €1.0 BN
(c. 22%)

320+
financial institutions

c. 17 BN
payment transactions per year



MOBILITY & E-TRANSACTIONAL SERVICES

Bringing payment & regulation expertise to new markets

c. €0.4 bn
(c. 8%)

350+
clients in various industries

c. 1.9 BN+
SMS sent per year

Data based on 2020 full year proforma

Merchant Services

ENERGISING COMMERCE WITH ADVANCED PAYMENT SERVICES

Worldline covers the full retail value chain, online and in-store, with a 360 degrees approach. We deliver a real digital journey for retailers and their customers and facilitate consumer engagement via seamless services on any device – with payment at the heart of the shopping experience.

Services

- Points of Sale
- Online & Omnichannel
- Payment Acceptance
- Digital Retail Services

We're trusted by industry leaders

Sainsbury's



NEXTON



PARKEON
Solutions for Smart Cities



c. 1.1M+
merchants served

#1
merchant acquirer in
continental Europe

#3
European provider of
e- and m-payment solutions

c. 250K
e-commerce customers
and websites

c. 22 BN
acceptance and acquiring
transactions per year

WORLDLINE

Financial Services

ENGINEERING THE MOST TRUSTED AND ADVANCED PAYMENT PROCESSING SOLUTIONS

Worldline provides modern payment solutions that help financial institutions meet their customers needs. We offer a unique combination of payment processing on an industrial scale as well as innovative solutions for payment and card-related transactions.

Services

- Issuing Processing
- Acquiring Processing
- Account Payments
- Digital Services

We're trusted by industry leaders



COMMERZBANK



ING



320+

financial institutions

#1

European payment
processor

c. 17 BN

payment transactions per
year

c. 126 BN

cards under
management

c. 9.6 BN

issuing processing
transactions

c. 11 BN

acquiring processing
transactions

WORLDLINE

Mobility & e-Transactional Services

BRINGING PAYMENT AND REGULATION EXPERTISE TO NEW MARKETS

Worldline provides end-to-end digital transactional services leveraging data and payment for a stronger customer engagement. We support our clients' digital transformation with the creation of innovative solutions, leveraging our extensive experience across sectors, and forming partnerships to develop joint business cases.

Services

- Omnichannel Contact Centre solution
- e-Ticketing & Open Payment
- Digital transformation programs
- Mobile competence centers

We're trusted by industry leaders



350+

clients in various industries

c. 3.5 M+

connected objects

c. 1.9 BN+

SMS sent per year

300+

experts dedicated to enterprise customer relationship optimisation

WORLDLINE 

People use our products

SEVERAL TIMES A DAY

Worldline is at your side making your world safer and easier!



**Pay for the
subway**



**Make a
secured
Online
Payment**



**Pay with your
credit card**

Notre modèle du Build To Run

DES SERVICES SUR-MESURE À FORTE VALEUR AJOUTÉE



HÉBERGER, EXPLOITER, MAINTENIR ET AMÉLIORER CES SERVICES

Worldline

NOS MÉTIERS

Nous travaillons principalement avec des technologies dites Open Source

80 % de nos recrutements nécessitent une qualification bac + 5 en informatique

Rejoindre Worldline, c'est choisir l'innovation, l'expertise et les technologies de pointe dans un environnement en constante évolution.



An work-study contract with Worldline ?

REINVENTING YOUR IT CAREER

- ▶ Des sujets innovants orientés vers les **nouvelles technologies**
- ▶ De vraies **responsabilités** et une montée en **compétences**
- ▶ Une **intégration** dans une **équipe projet**
- ▶ Opportunité d'accéder aux **différentes phases d'un projet**
- ▶ **Formation** et **accompagnement**

Candidatures / Contacts :

- orphee.nsiami@worldline.com ou ophelie.sergent@worldline.com
- sites : Villeurbanne/Marseille/Valence
- <https://www.welcometothejungle.com/fr/companies/worldline>
- [Portail Lyon1 - Jobteaser](#)



CERTIFIED



Great
Place
To
Work®

HappyIndex® Trainees
FRANCE 2024

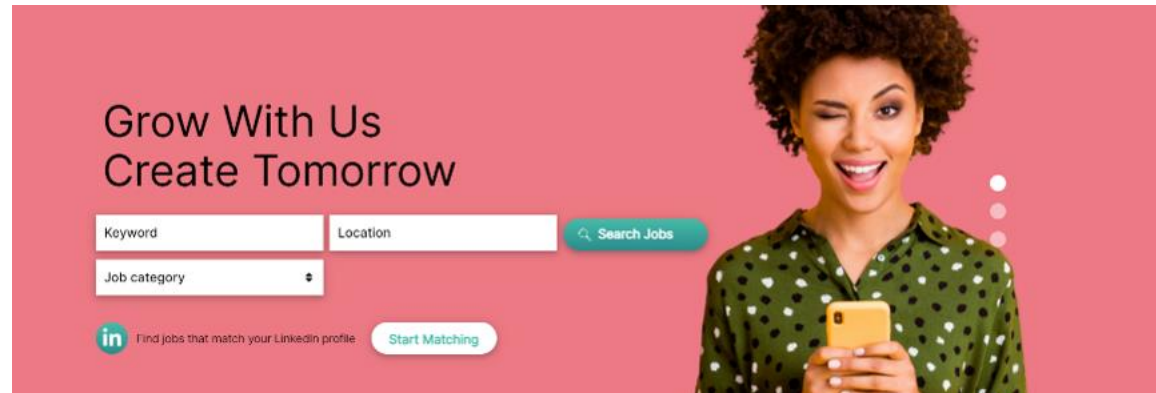
WORLDLINE 

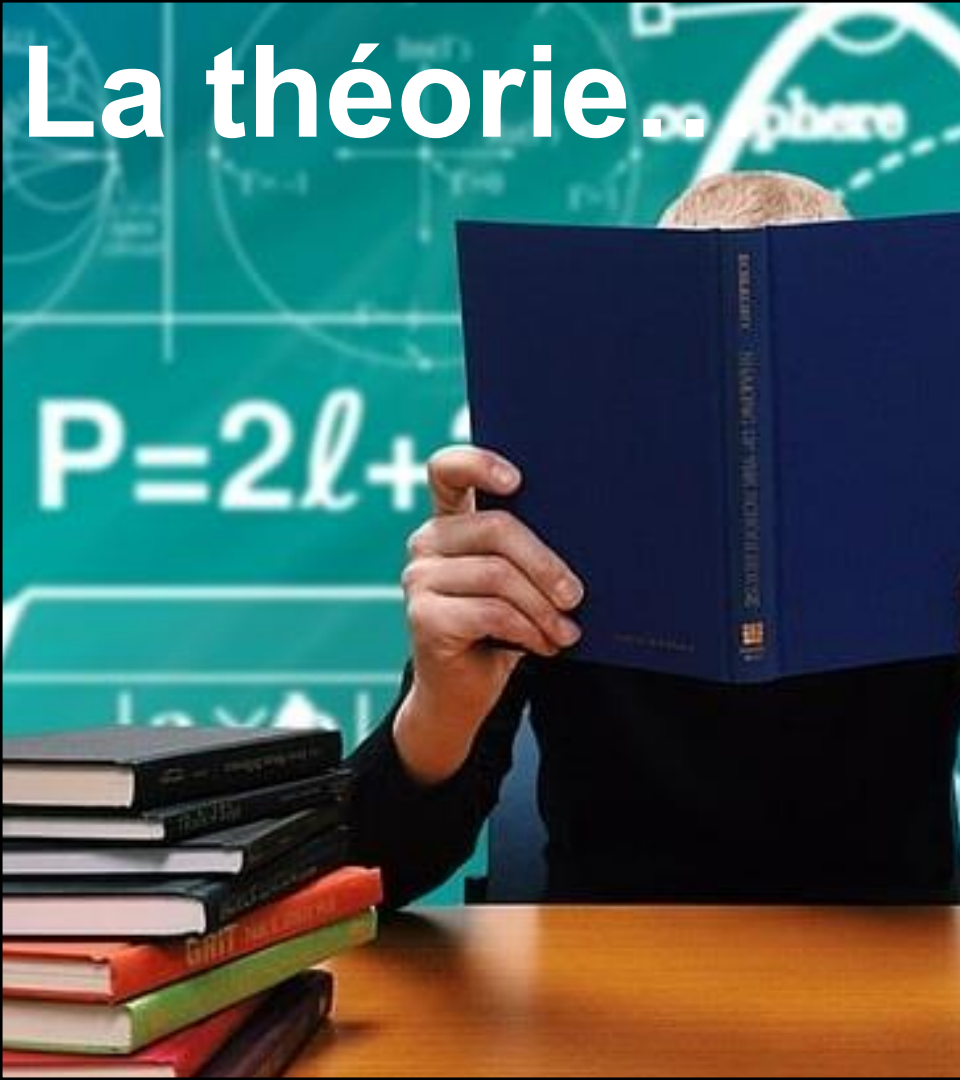
What about our offer ?

Participate to a great adventure !

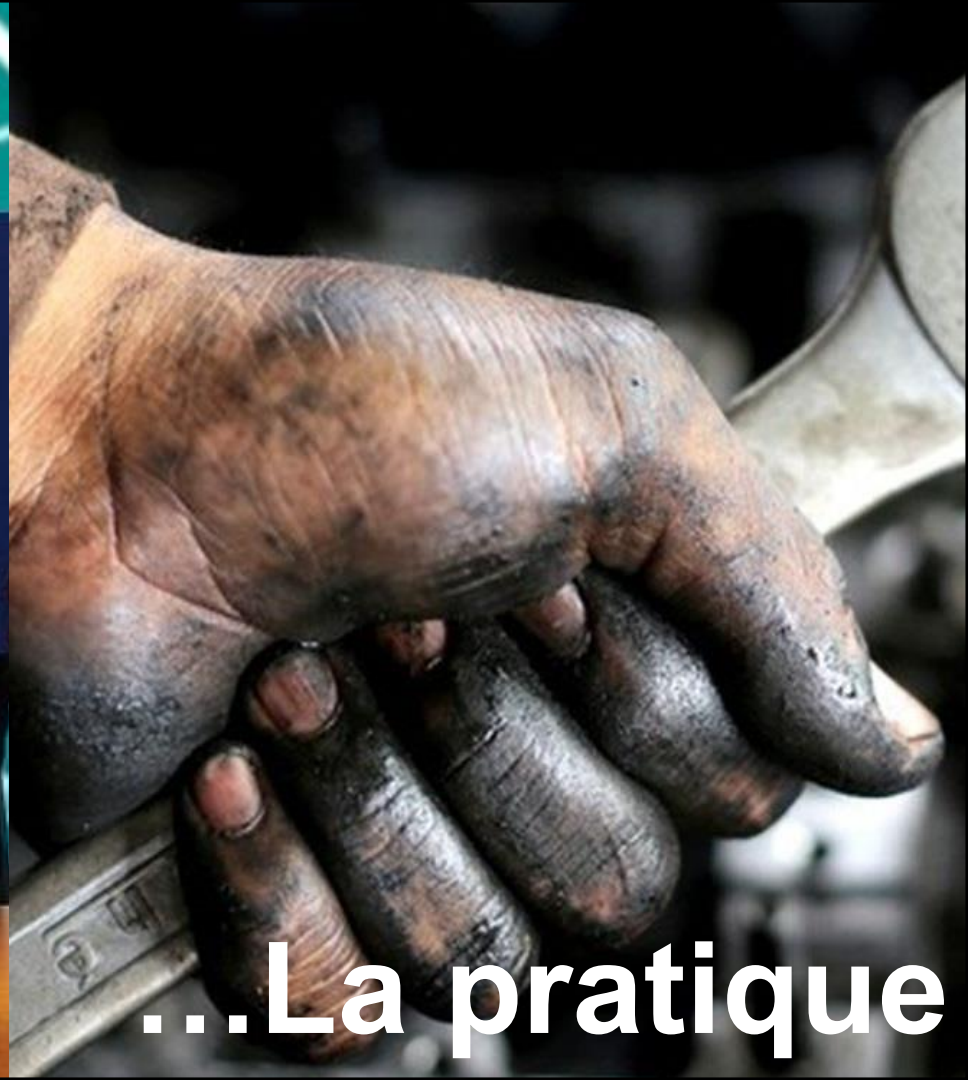
More than 50 work-study offers in France

In all domains : Development (Front, Back), Production, System, Network...





La théorie...



...La pratique



Définitions

Urbanisation

DÉFINITION SELON WIKIPÉDIA

L'**urbanisation du système d'information** de l'entreprise **est une discipline** informatique consistant à faire **évoluer le système d'information** d'une entreprise **dans son ensemble afin de garantir sa cohérence** vis-à-vis des **objectifs et du métier** de cette entreprise, en prenant en compte ses **contraintes externes et internes**, tout en tirant parti des opportunités de l'état de l'art informatique.

Cette discipline s'appuie sur une série de **concepts calqués sur ceux de l'urbanisation de l'habitat humain** (organisation des villes, du territoire), concepts qui ont été réutilisés en informatique **pour formaliser ou modéliser** le système d'information.

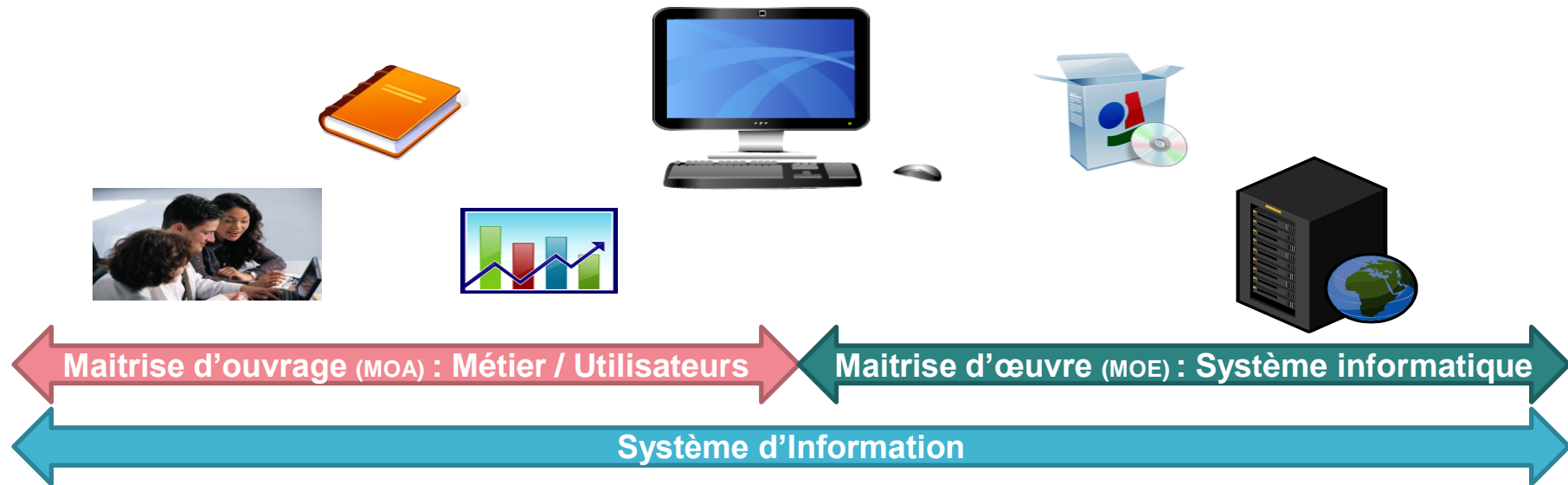
L'**urbanisme définit des règles** ainsi qu'un cadre cohérent, stable et modulaire, auquel **les différentes parties prenantes se réfèrent pour toute décision d'investissement** relative au management du système d'information.

En résumé, urbaniser, c'est diriger la transformation continue du Système d'Information en vue de le simplifier et garantir sa cohérence.

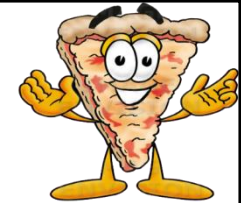
Système d'information

DÉFINITION SELON WIKIPÉDIA

Un **Système d'Information (SI)** est un **ensemble organisé de ressources** (matériels, logiciels, personnel, données et procédures) **qui permet de collecter, stocker, traiter et de distribuer de l'information.**



fil range



► Petite entreprise de livraison de pizza

► Actuellement

- Les commandes se font par téléphone
- 3 postes : téléphone / cuisine / livraison

► Souhaite

- S'ouvrir sur internet
- Trouver des nouveaux moyens de communication avec ses clients

► Objectif

- Doubler de chiffre d'affaires en 2 ans

La démarche d'urbanisation



La démarche d'urbanisation

LES 4 GRANDES ÉTAPES

- ▶ **Définition des objectifs** : Stratégie commerciale
- ▶ **Analyse de l'existant**
 - Lister le patrimoine
 - Cartographier les différentes couches (métier, fonctionnelle, applicative, ...)
- ▶ **Identification du SI cible**
 - Impact sur les différentes couches
 - Prise en compte des contraintes (humaine, matériel, ...)
 - Définition des livrables
- ▶ **Mise en évidence de la trajectoire**
 - Planning
 - Définition des paliers



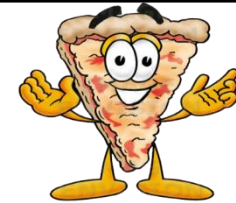
G

O

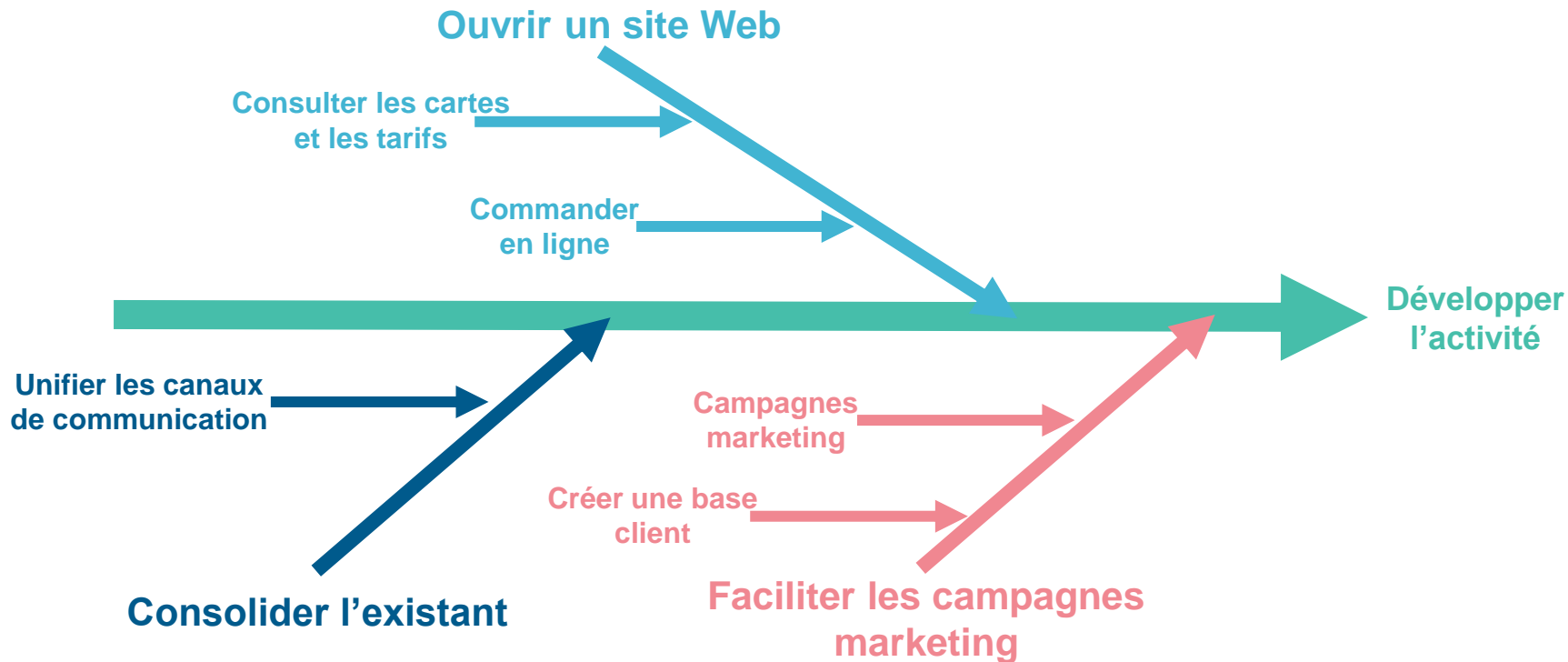
a

I

S



OBJECTIFS STRATÉGIQUES





CHECKLIST



L'état des lieux

Analyse de l'existant

LISTER LE PATRIMOINE

► Capital matériel

- Logiciels (commerciaux, open source, progiciels, ...)
- Matériels (serveurs, réseaux, postes de travail, ...)

► Capital immatériel

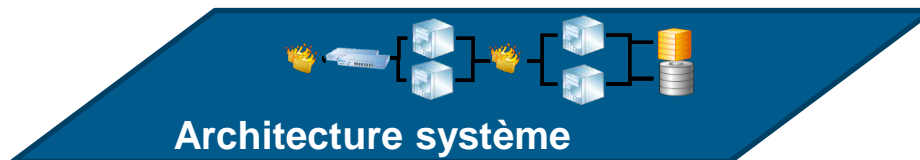
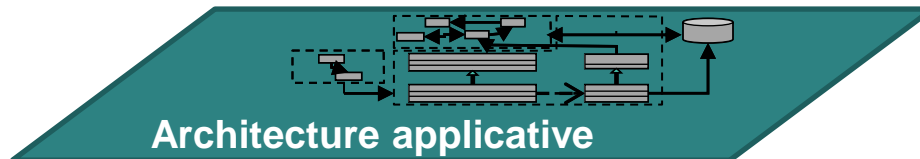
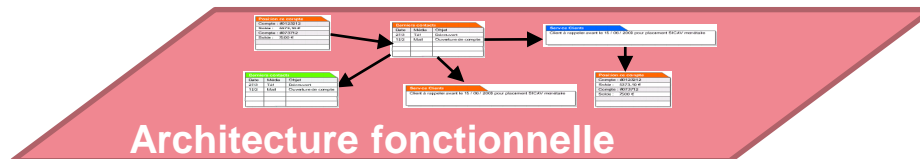
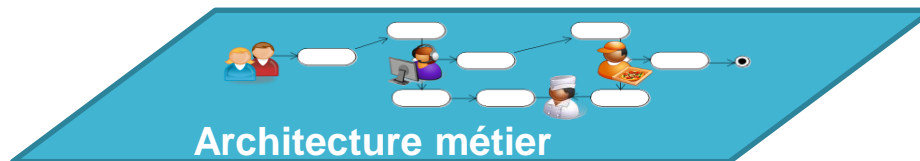
- Savoir-faire (procédure, formation, expertise, ...)
- Contribution à l'activité



Remarques : suivant la taille de l'entreprise, le SI représente d'une dizaine à plusieurs milliers d'applications

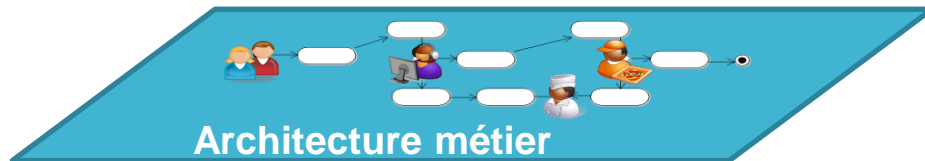
Analyse de l'existant

CARTOGRAPHIER LES DIFFÉRENTES COUCHES



Analyse de l'existant

ARCHITECTURE MÉTIER



► Identifier les « processus métiers »

- Qui fait quoi et pourquoi ?
- Concept simple mais en général, personne n'a la vision complète

► **Forme** : Différents formats BPMN (*Business Process Model and Notation*), EPC (*Event-driven Process Chain*), etc.

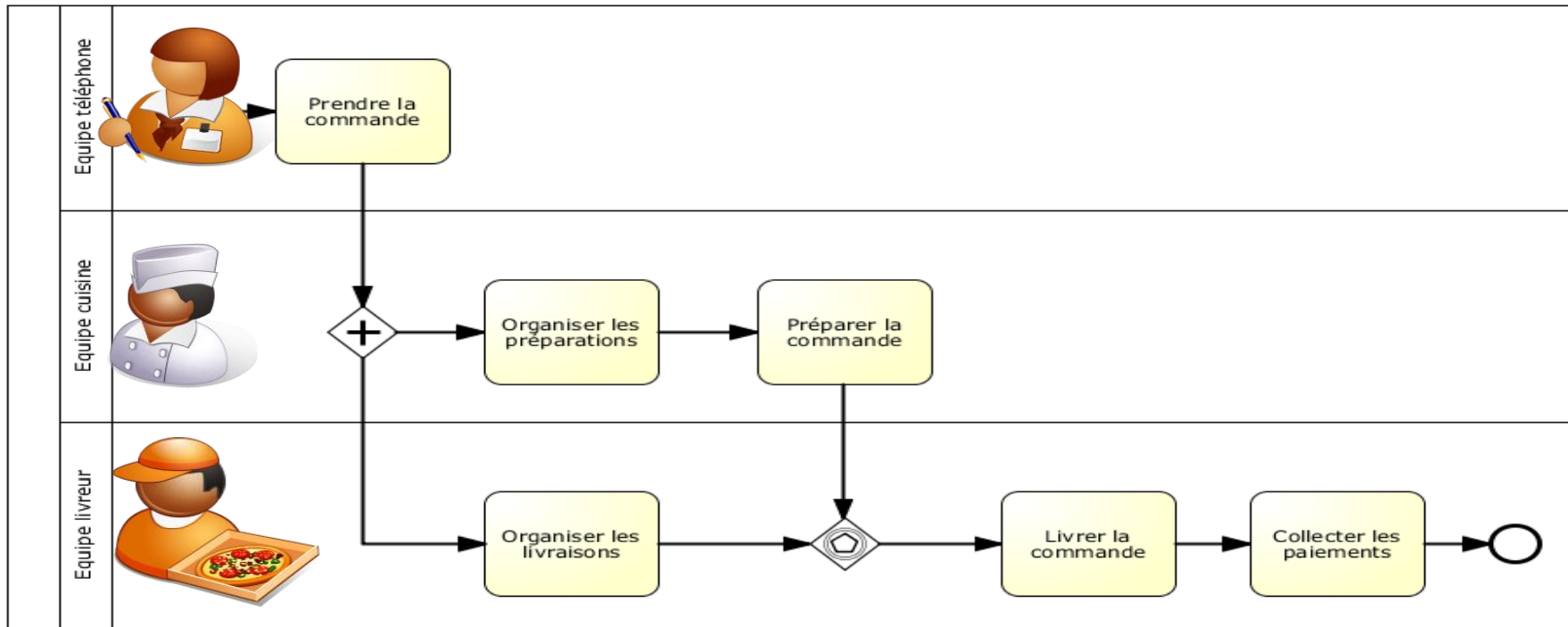
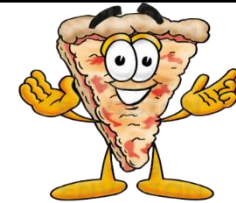
► **Difficultés**

- Sensible, touche à l'organisation et aux personnes
- Son importance est souvent sous-estimée

► **Valeur ajoutée**

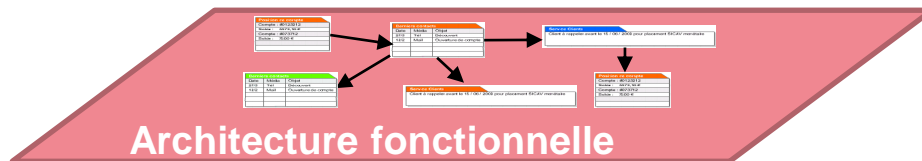
- Améliore la compréhension globale, bon terrain de dialogue MOA / MOE
- Démultiplie les possibilités d'optimisation

ARCHITECTURE MÉTIER



Analyse de l'existant

ARCHITECTURE FONCTIONNELLE



► Identifier les « blocs fonctionnels »

- De quoi a-t-on besoin pour réaliser les processus métiers ?

► Forme

- Basée sur un découpage classique en zones (échanges, cœur de métier, données de référence, données de production, activités support, pilotage)

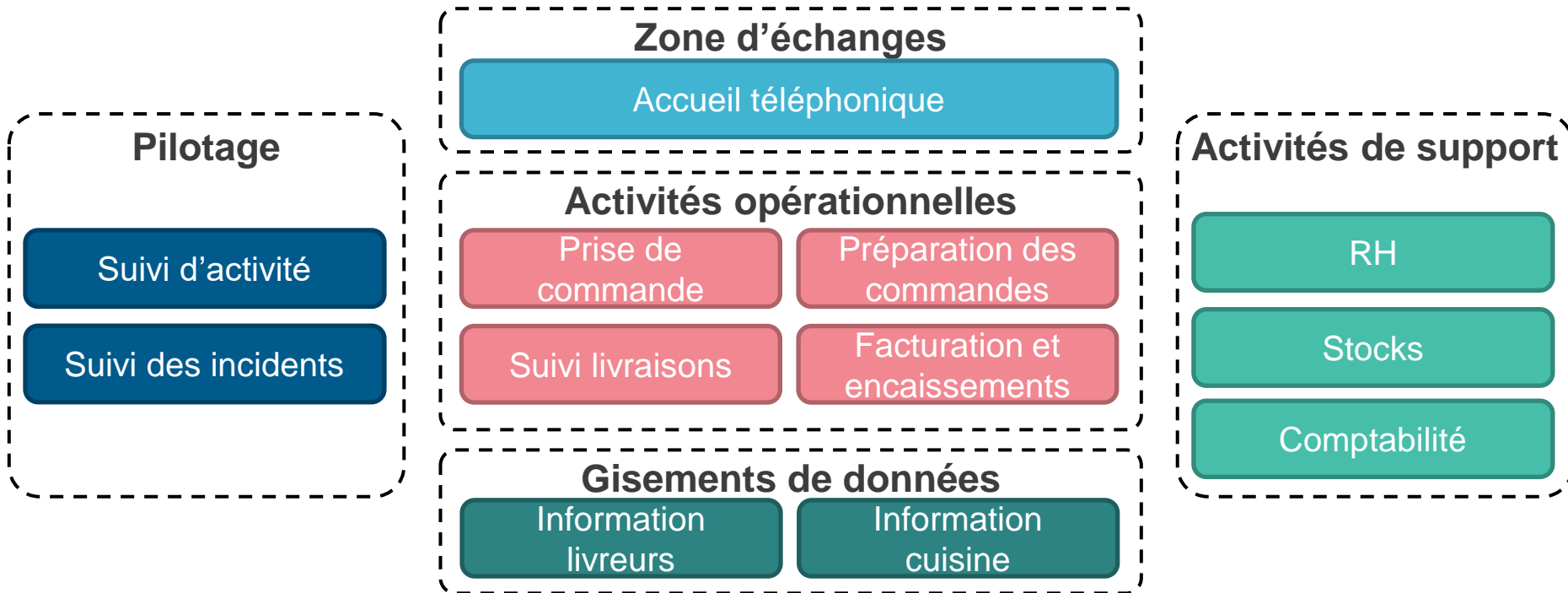
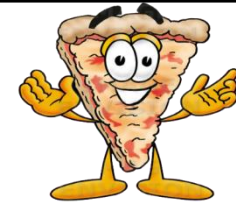
► Difficultés

- Être cohérent avec les processus métiers
- Choisir le bon niveau de détail

► Valeur ajoutée

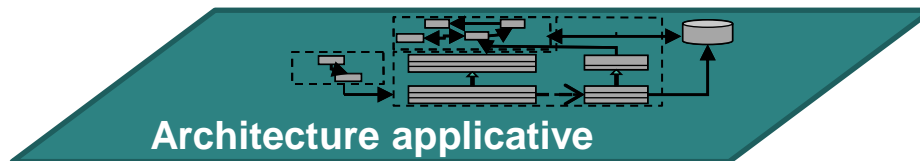
- Présentation hiérarchique, permet le découpage des tâches

ARCHITECTURE FONCTIONNELLE



Analyse de l'existant

ARCHITECTURE APPLICATIVE



► Identifier les applications

- Comment va-t-on réaliser les fonctionnalités ?

► Forme

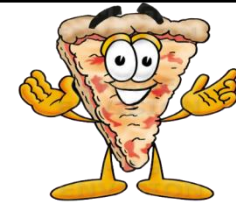
- Basée sur un découpage classique N-Tiers

► Difficultés

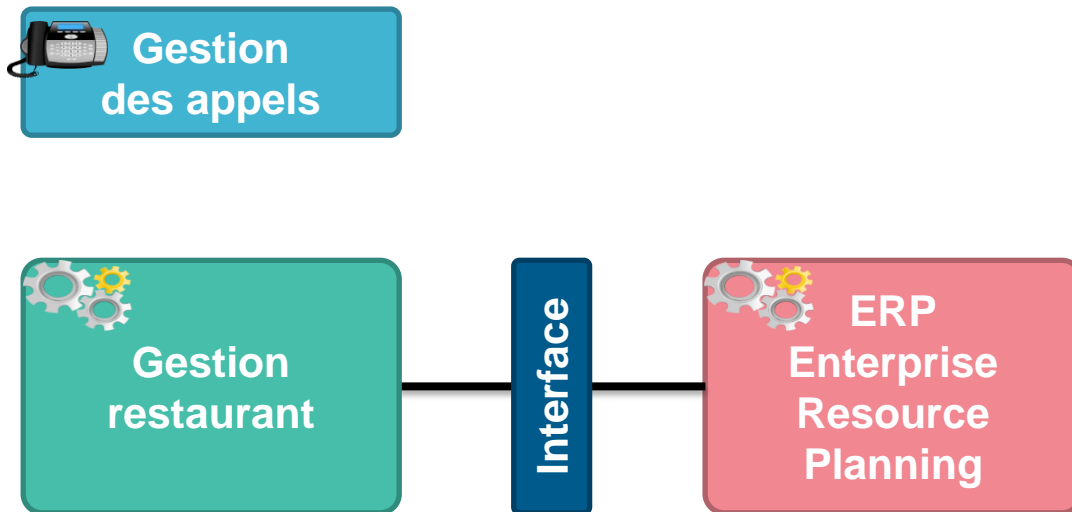
- Apporter de la valeur et des solutions par rapport à l'architecture fonctionnelle
- Rester compréhensible par le commun des mortels

► Valeur ajoutée

- Jette les bases de la réalisation (grands choix technologiques, etc.)

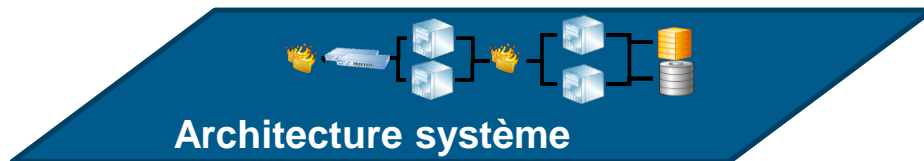


ARCHITECTURE APPLICATIVE



Analyse de l'existant

ARCHITECTURE SYSTÈME



► Identifier les composants techniques

- Avec quoi et où fonctionnent les applications

► **Forme** : Basée sur un découpage classique en zones techniques (sécurité, stockage, etc.)

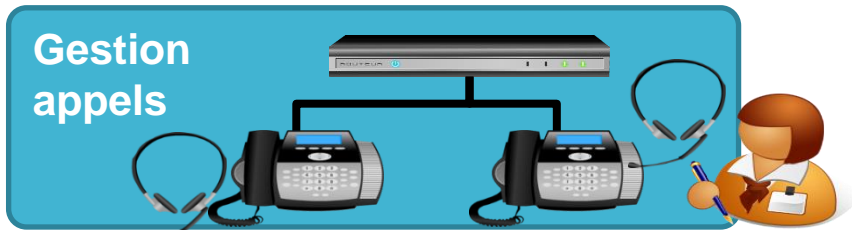
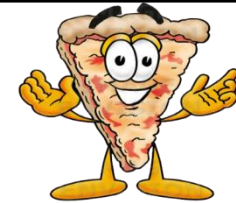
► Difficultés

- Faire le lien entre les applications et les serveurs
- Rester compréhensible par le commun des mortels

► Valeur ajoutée

- Apporte du concret et du structurant
- Indispensable pour évaluer le coût du système (On-premise vs Cloud)

ARCHITECTURE SYSTEMÈ

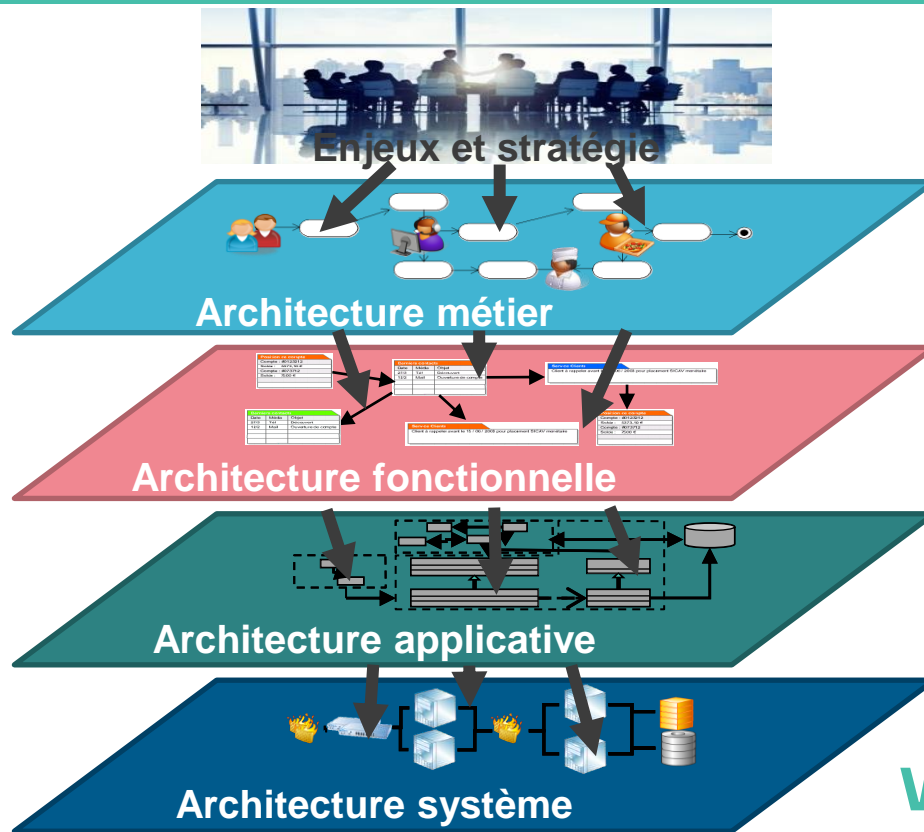


La cible



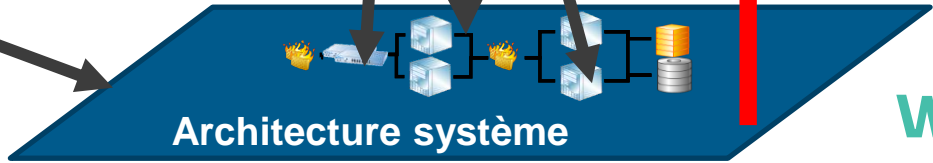
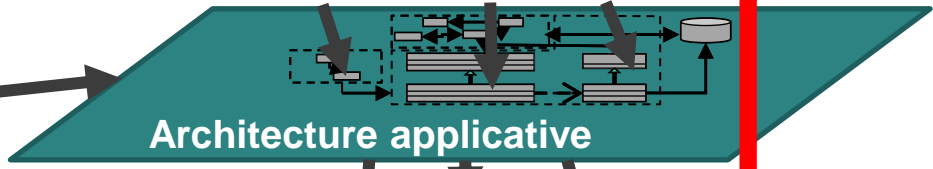
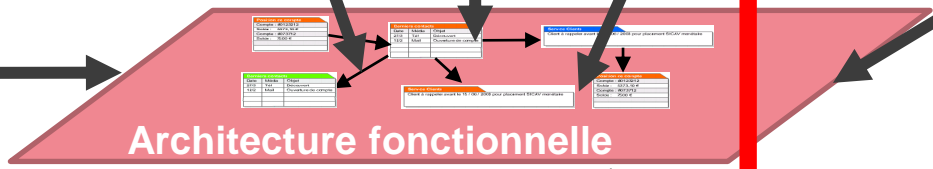
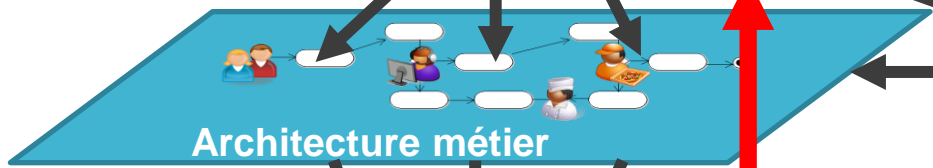
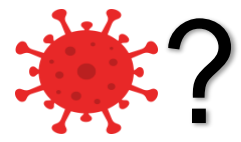
SI Cible

IMPACT SUR LES DIFFÉRENTES COUCHES



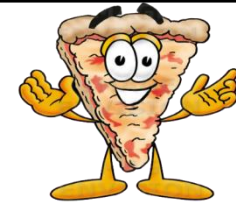
SI Cible

"LES IMPONDÉRABLES"

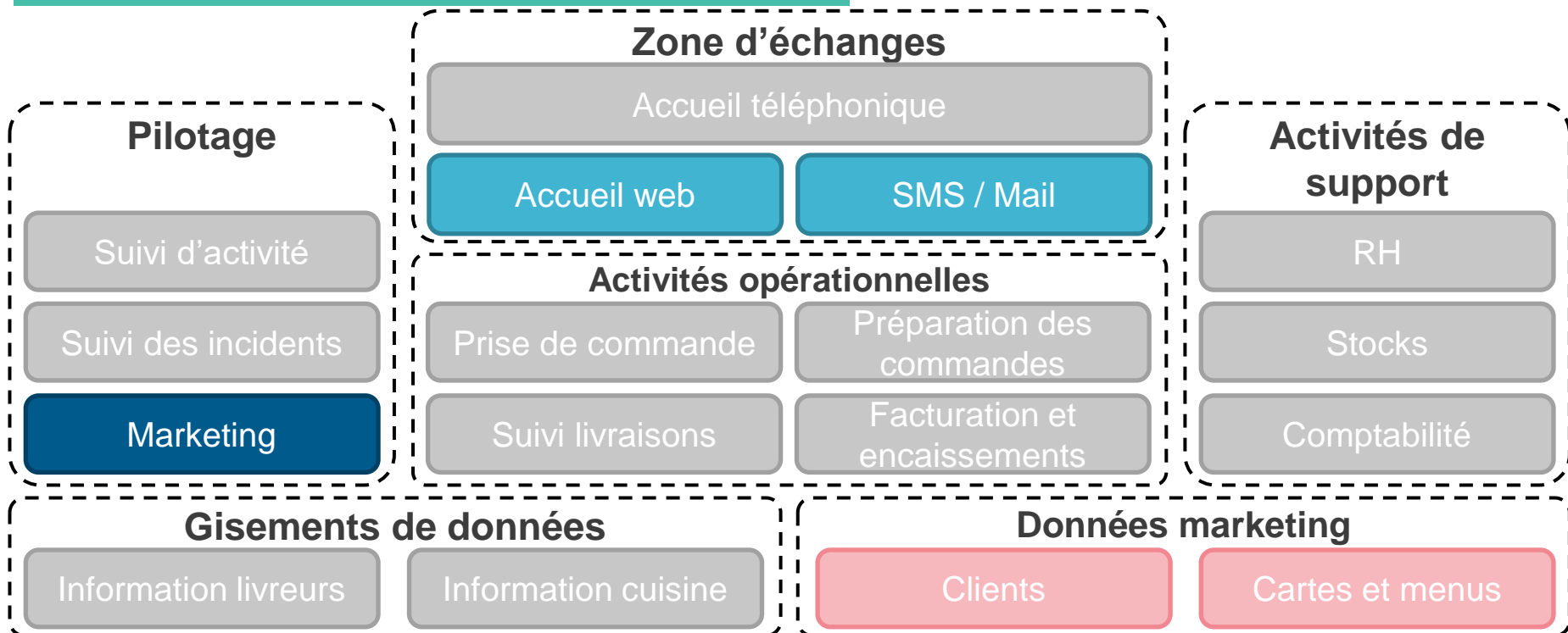


Patch Management
Évolution TLS/SSL
Common Vulnerabilities & Exposures
Etc.

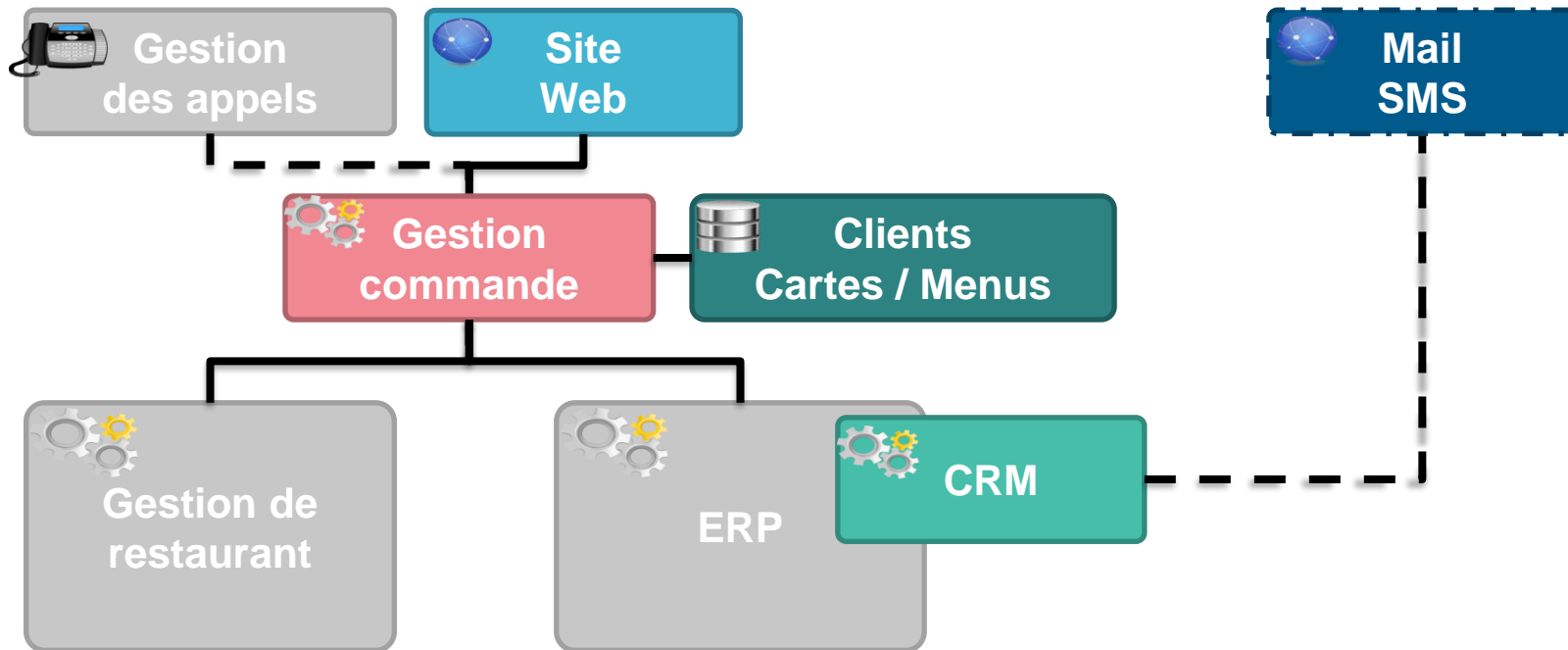
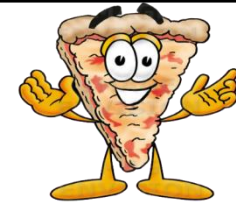




CIBLE FONCTIONNELLE



CIBLE APPLICATIVE



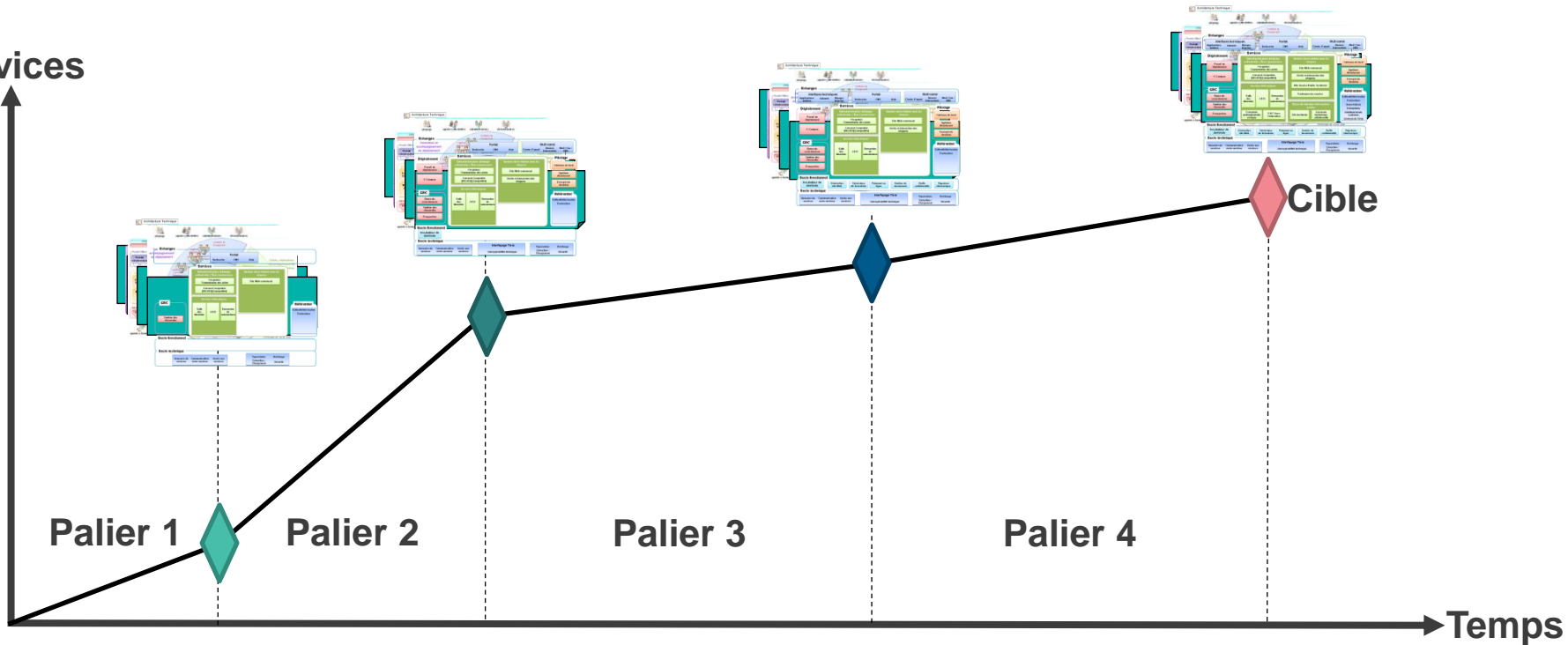


La trajectoire

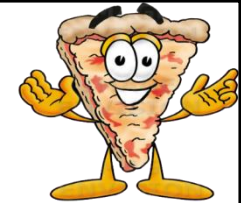
Définition de la trajectoire

PALIER PAR PALIER

Services



LES PALIERS / LE PLANNING



► Etape 1 : T0 + 3 mois

- Mise en place de la gestion de commandes et du référentiel client

► Etape 2 : T0 + 4 mois

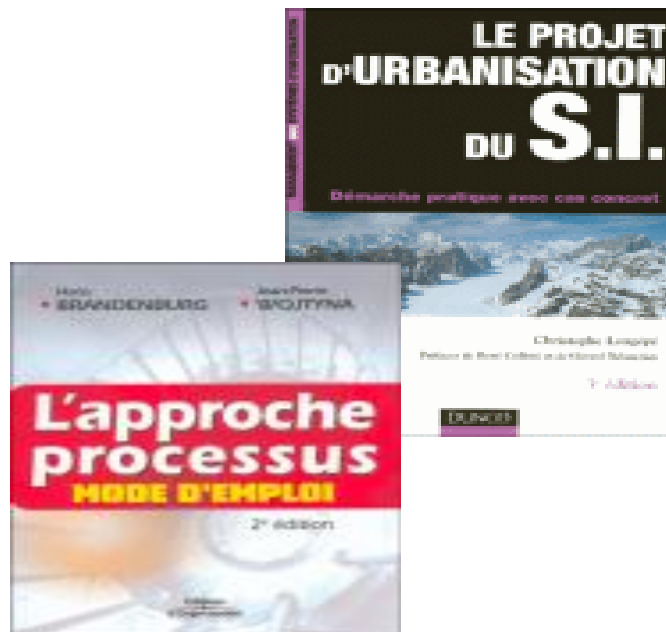
- Ouverture du site Web

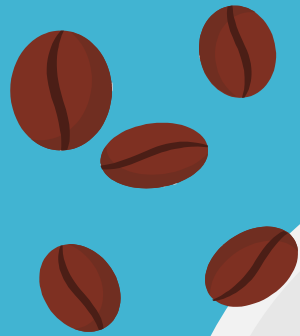
► Etape 3 : T0 + 6 mois

- Nouvelles fonctionnalités CRM : campagne de promotion par mail, etc.

Pour aller plus loin...

LES LIVRES





PAUSE ???

RETOUR DANS 5 MINUTES !

Et en pratique ?



Différents niveaux

INTRA / INTER



**Approche intra
applicative**
(design pattern)

**Approche inter
applicative**
(architecture)

Approche intra applicative

ETAPE 0

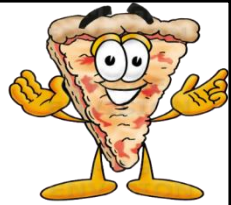


Etape 0

- JSP
- Frameworks

Pizza 3000

"THE" JSP



```
pizza-list.jsp
<%
html
java
sql
%>
```



```
<html>
<head>
  <title>Pizza 3000</title>
</head>
<body>
  <h1>Carte des Pizzas</h1>
  <%
  ResultSet result = null;
  try {
    Class c = Class.forName("com.mysql.jdbc.Driver");
    Connection conn = DriverManager.getConnection("jdbc:mysql:localhost/urba", "urba", "Ur8@");

    Statement stmt = conn.createStatement();
    result = stmt.executeQuery("SELECT * FROM Pizza");
  } catch (ClassNotFoundException | SQLException e) {
    System.out.println("Error occurred: " + e);
  }
  %>
  <ul>
    <%
    while(result.next()) {
      out.println("<li>" + result.getString(2) + "</li>");
    }
    %>
  </ul>
</body>
</html>
```

Mélange des préoccupations (métier/ technique)

Mélange langage (HTML / JAVA)

Impacts entre les différentes équipes (SQL / Dév back / Dév Front)

Gestion manuelle de l'accès à la BDD

Les Frameworks

LA JUNGLE



HIBERNATE



Struts²



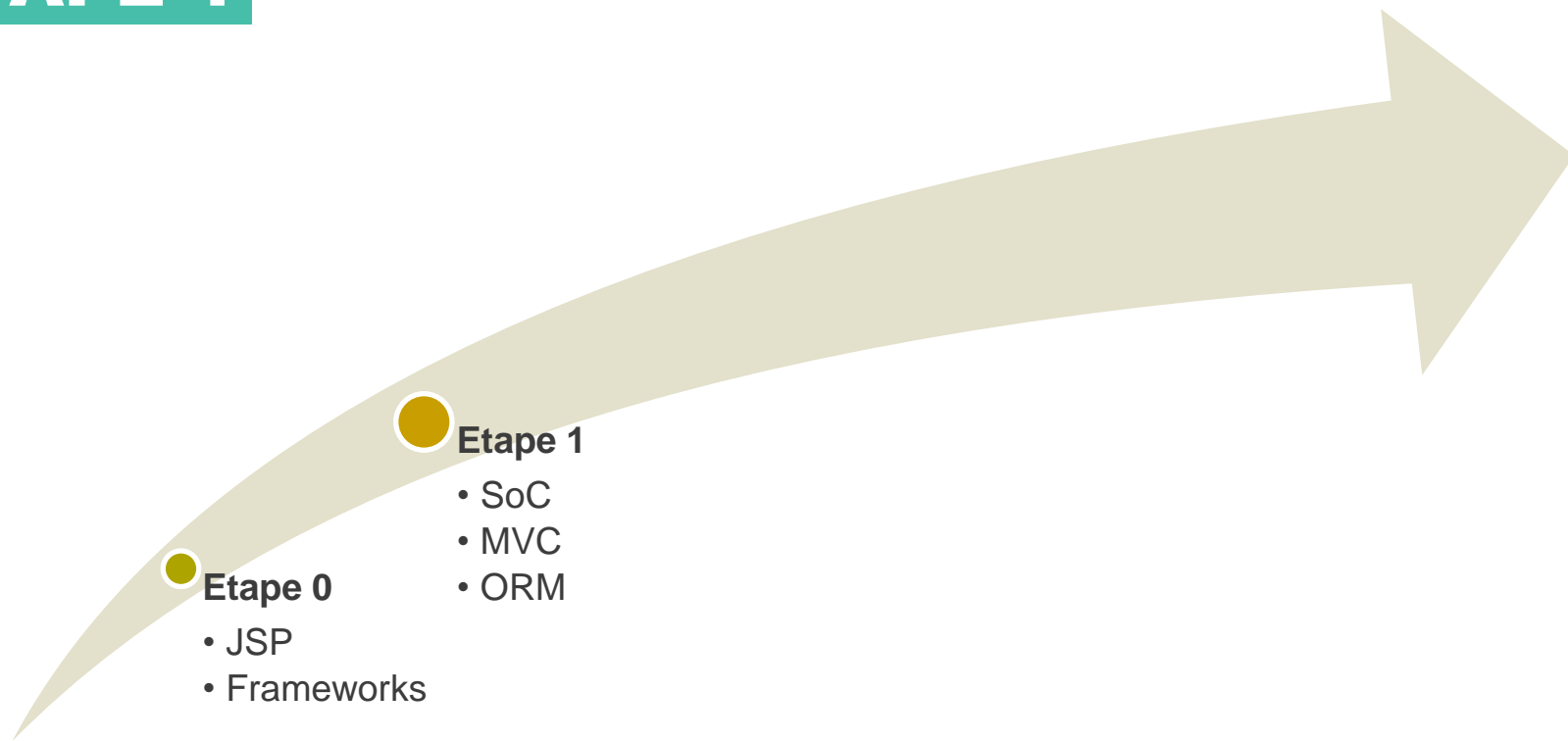
Les Frameworks

POURQUOI ?



Approche intra applicative

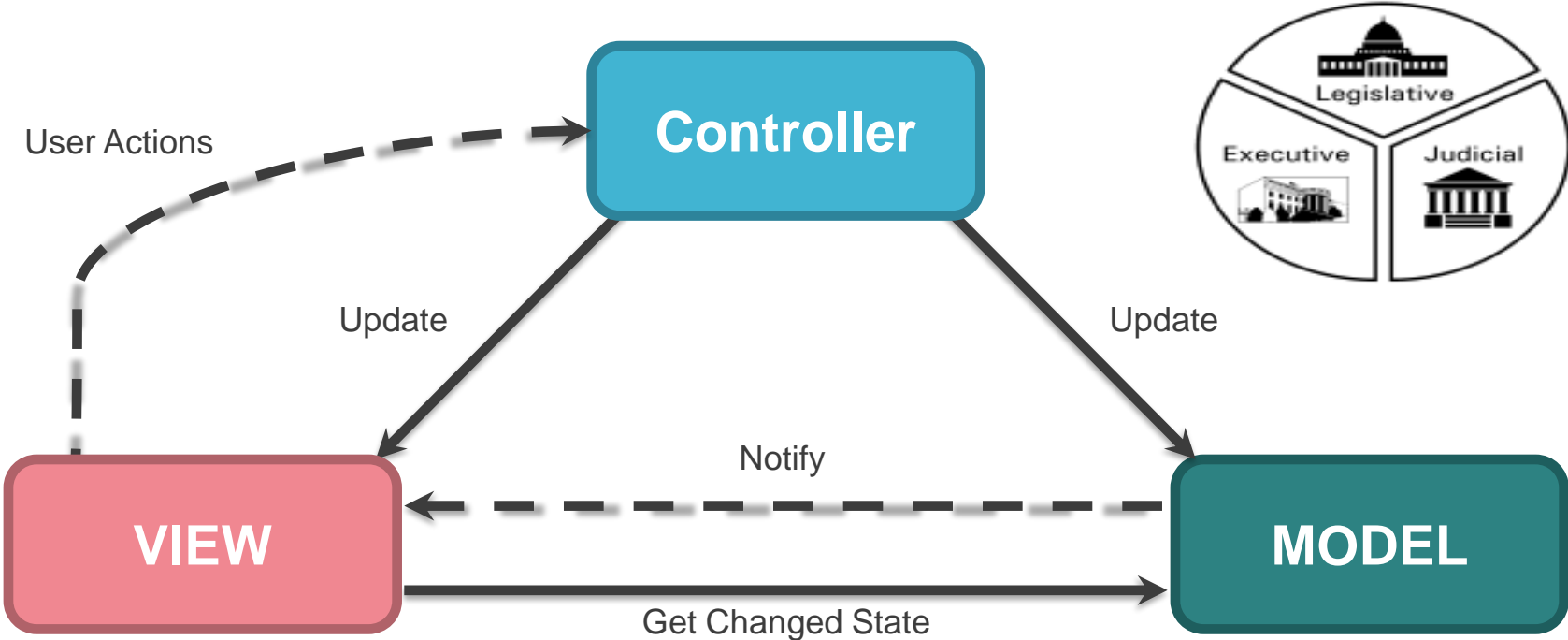
ETAPE 1



Separation of Concerns

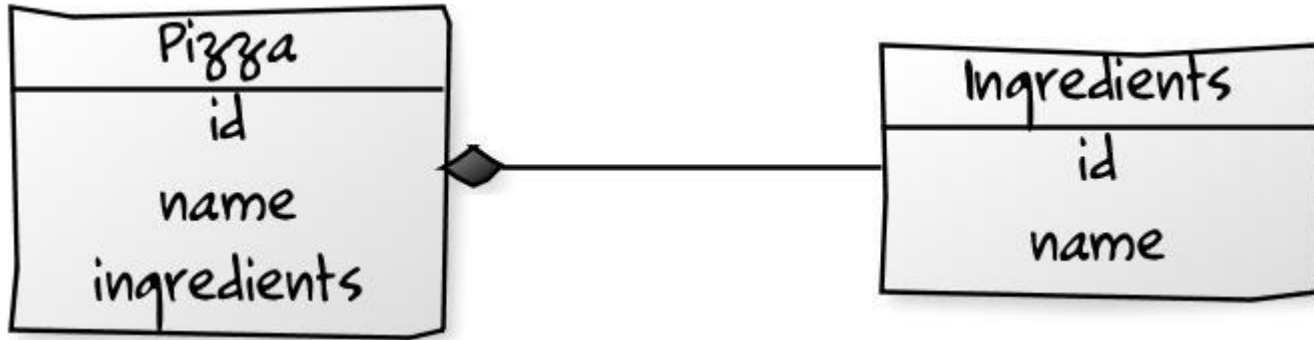
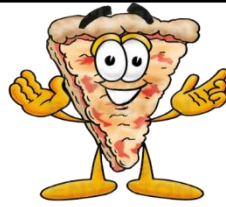
MVC PATTERN*

**(UNE VARIANTE POSSIBLE)*



Pizza 3000

MODEL



```
public class Pizza {
```

```
    private Long id;
```

```
    private String name;
```

```
    private Set<Ingredient> ingredients = new HashSet<>();
```

```
    public Long getId() { return id; }
```

```
    public void setId(Long id) { this.id = id; }
```

```
    public String getName() { return name; }
```

```
    public void setName(String name) { this.name = name; }
```

```
    public Set<Ingredient> getIngredients() { return ingredients; }
```

```
    public void setIngredients(Set<Ingredient> ingredients) {  
        Assert.notNull(ingredients, "ingredients must be not null");  
        this.ingredients = ingredients;  
    }
```

```
    // "equals(...)", "hashCode()" and "toString()" functions omitted for brevity
```

Remarques : Non modélisé ici (même logique)

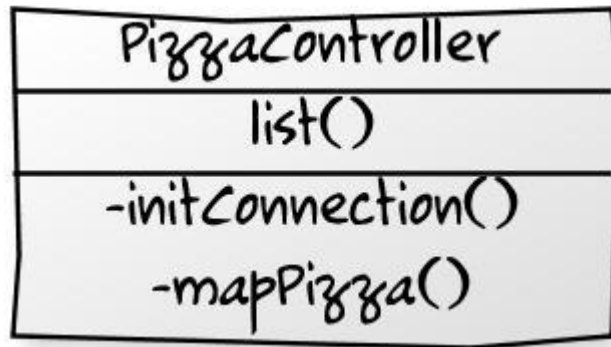
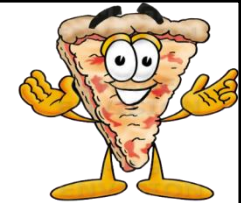


Ingredient

```
}
```

Pizza 3000

CONTROLLER




```
public class PizzaController {
```

```
    private Connection connection;
```

```
    public PizzaController() { initConnection(); }
```

```
    public List<Pizza> list() throws SQLException {
```

```
        Statement statement = connection.createStatement();  
        ResultSet result = statement.executeQuery("SELECT * FROM Pizza");
```

```
        List<Pizza> pizzas = mapPizzas(result);
```

```
        pizzas.forEach(System.out::println);  
        return pizzas;
```

```
    }
```

Le Controller joue son rôle de chef d'orchestre

Utilisation du model Pizza

```
private List<Pizza> mapPizzas(ResultSet result) throws SQLException {
```

```
    List<Pizza> pizzas = new ArrayList<>();
```

```
    while (result.next()) {
```

```
        Pizza pizza = new Pizza();
```

```
        pizza.setId(result.getLong(1));
```

```
        pizza.setName(result.getString(2));
```

```
        pizzas.add(pizza);
```

```
    }
```

```
    return pizzas;
```

```
}
```

Gestion manuelle de l'accès à la BDD

Mapping manuel

```
private void initConnection() {
```

```
    try {
```

```
        connection = DriverManager.getConnection("jdbc:mysql://localhost/urba", "urba", "Ur8@");
```

```
    } catch (Exception e) {
```

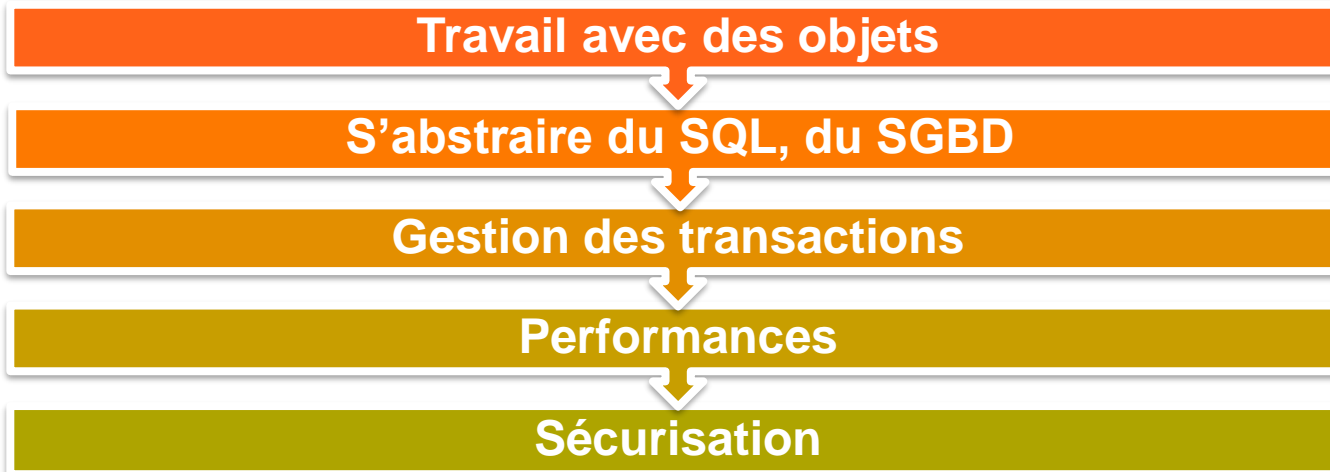
```
        System.out.println("Error occurred: " + e);
```

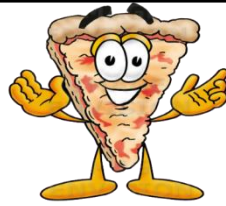
```
    }
```

```
}
```

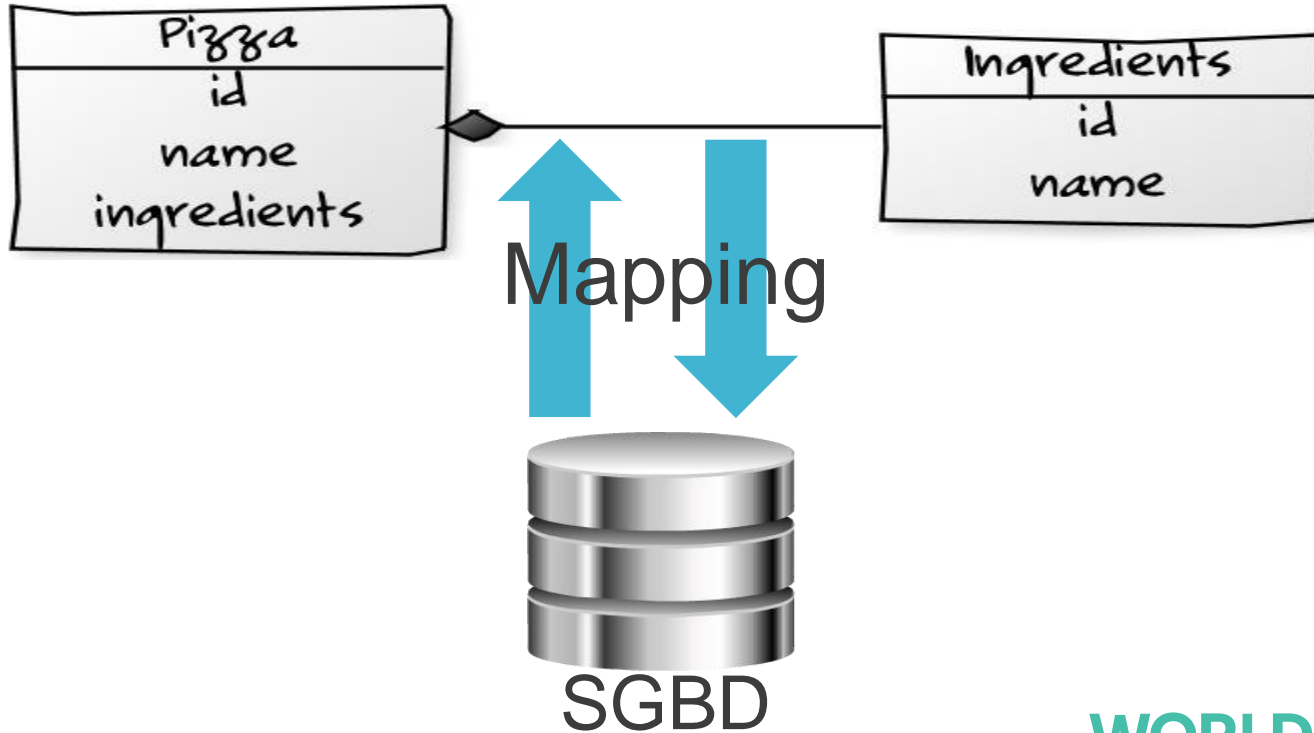
ORM : Object Relational Mapping

POURQUOI ?





ORM : OBJECT RELATIONAL MAPPING



```
@Entity
```

```
public class Pizza {
```

```
@Id @GeneratedValue
```

```
private Long id;
```

```
@NaturalId
```

```
@NotBlank
```

```
private String name;
```

```
@ManyToMany(cascade = {CascadeType.PERSIST, CascadeType.MERGE})
```

```
private Set<Ingredient> ingredients = new HashSet<>();
```

```
public Long getId() { return id; }
```

```
public String getName() { return name; }
```

```
public void setName(String name) { this.name = name; }
```

```
public Set<Ingredient> getIngredients() { return ingredients; }
```

```
public void addIngredients(Ingredient... ingredients) {
```

```
    Assert.notNull(ingredients, "ingredients must be not null");
```

```
    Arrays.stream(ingredients).forEach(this.ingredients::add);
```

```
}
```

```
public void removeIngredients(Ingredient... ingredients) {
```

```
    Assert.notNull(ingredients, "ingredients must be not null");
```

```
    Arrays.stream(ingredients).forEach(this.ingredients::remove);
```

```
}
```

```
//"equals(...)", "hashCode()" and "toString()" functions omitted for brevity
```

```
}
```

Informations pour « décrire/expliquer » le mapping à Hibernate

rem: Attention à la sémantique des différentes annotations @OneToMany vs @ManyToMany, etc.

Tips : JSR 303 - Bean Validation

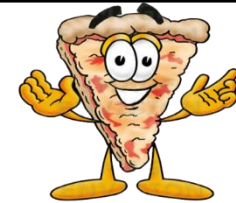
Non modélisé ici (même logique)

Tips : Restriction des éléments exposés

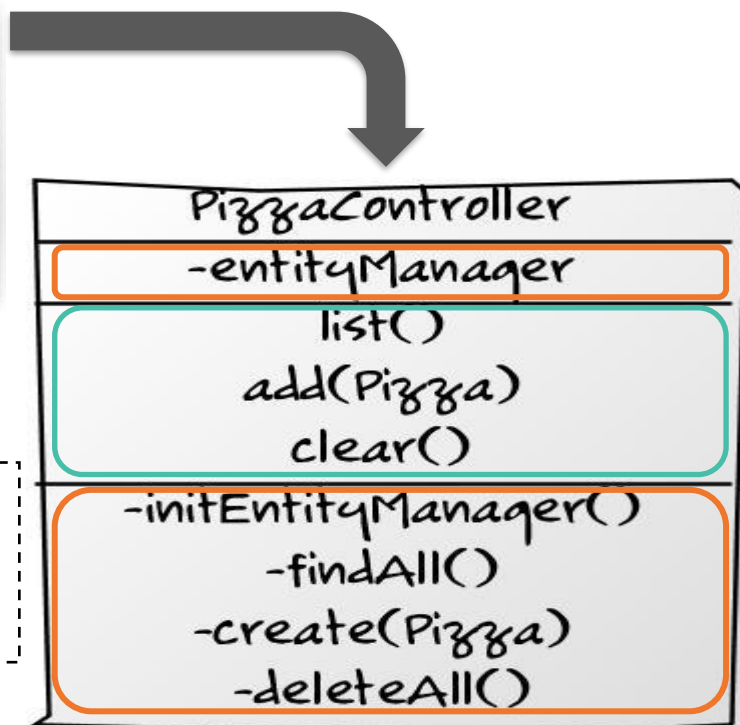
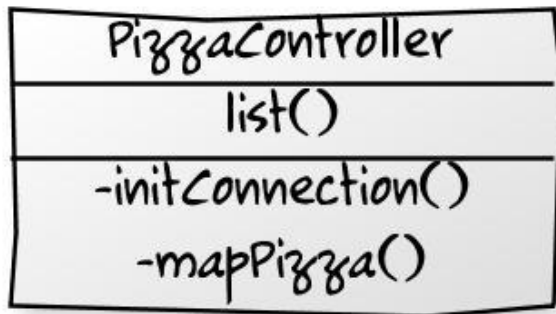
(on n'expose pas directement la collection)

Tips : Méthodes utilitaires

(synchronisation, sucre syntaxique, gestion bidirectionnelle, etc.)



ORM : OBJECT RELATIONAL MAPPING



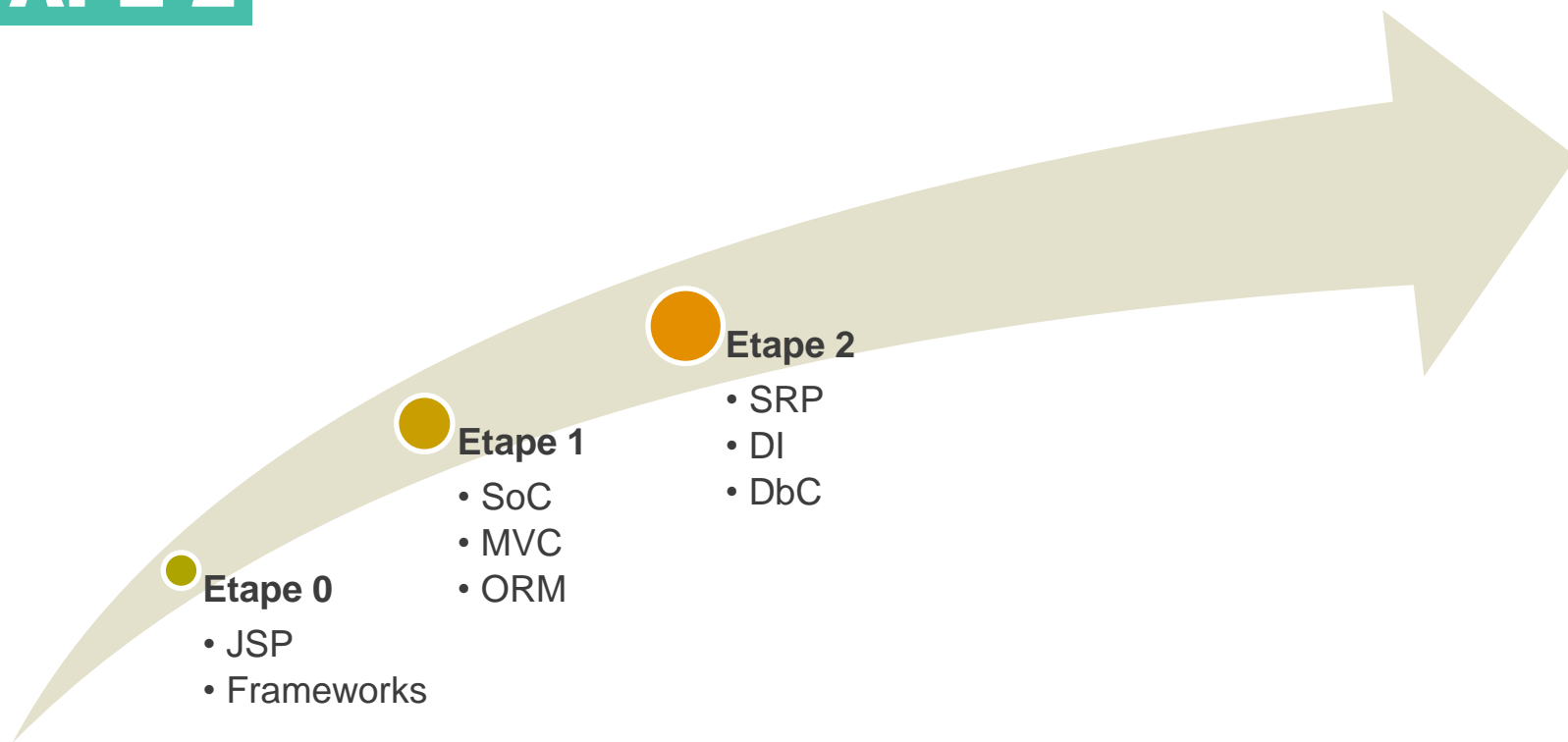
2 responsabilités

- Rôle officiel : Contrôleur
- Fait fonction (en l'état) de couche d'accès aux données

Apports de l'ORM
Facilitation et normalisation de l'accès aux données

Approche intra applicative

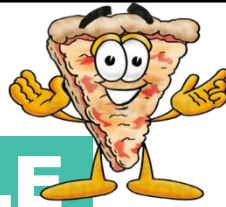
ETAPE 2



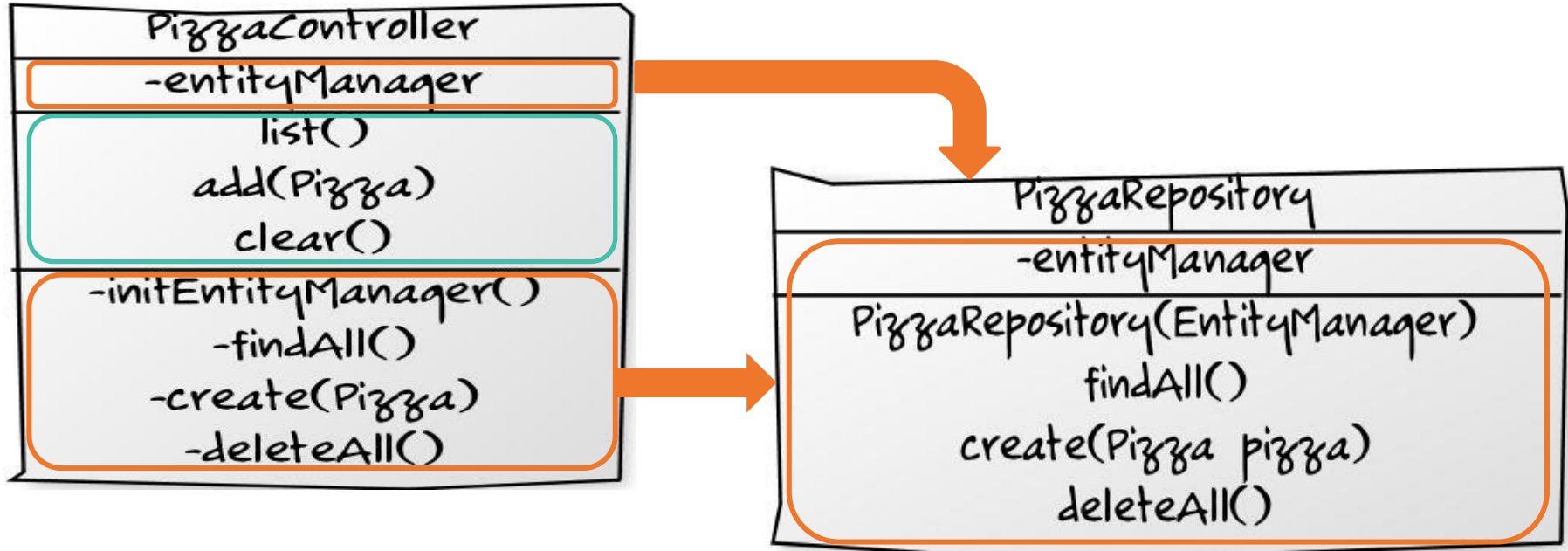
SRP : Single Responsibility Principle

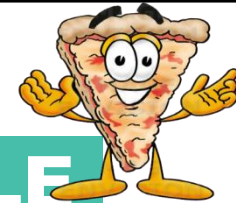


Pizza 3000

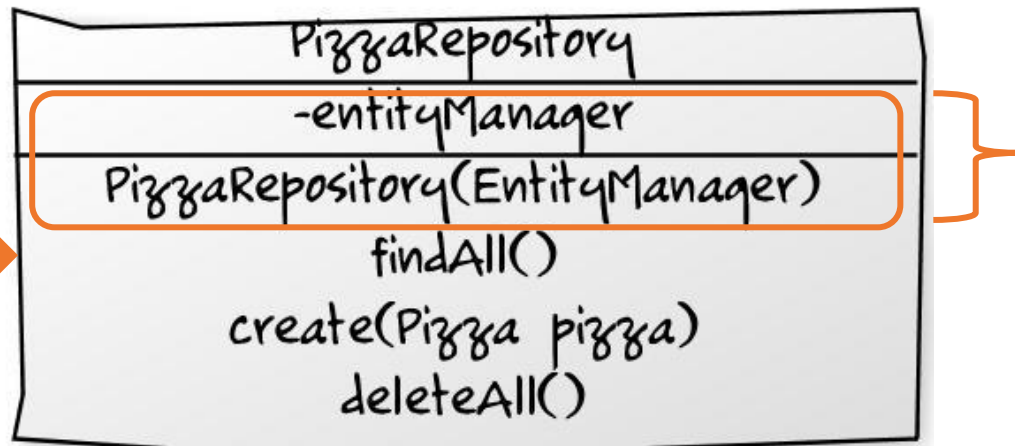


SRP : SINGLE RESPONSIBILITY PRINCIPLE



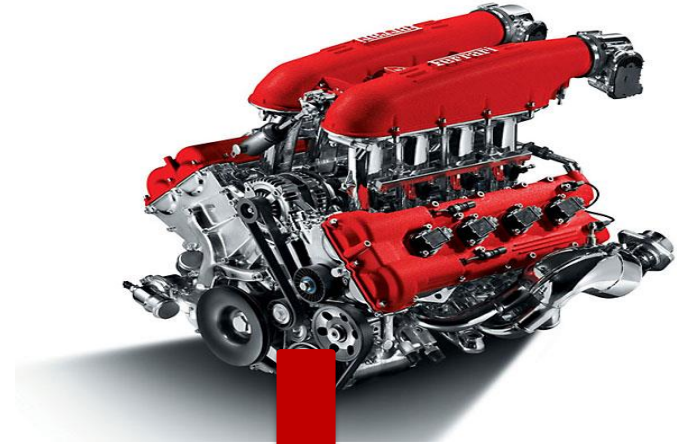


SRP : SINGLE RESPONSIBILITY PRINCIPLE

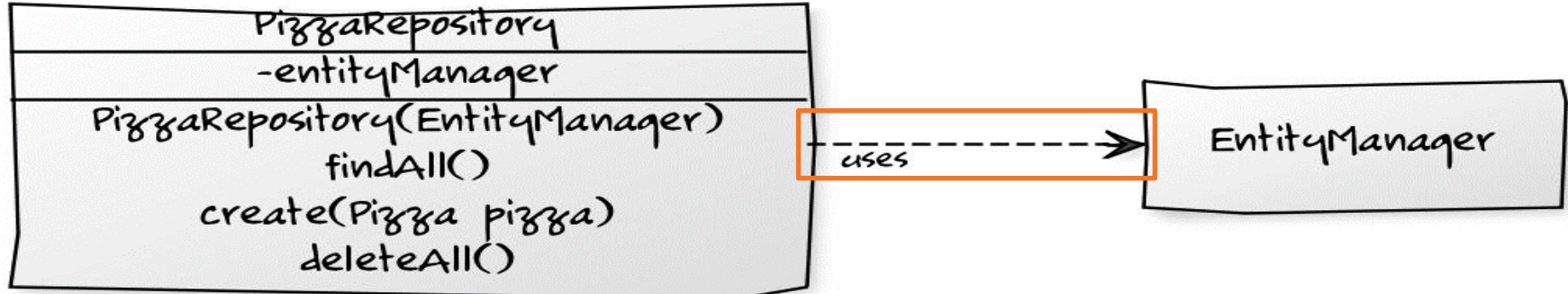
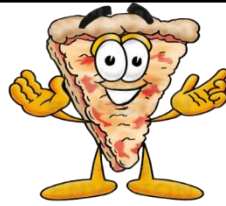


On voit apparaitre une dépendance au cœur du fonctionnement

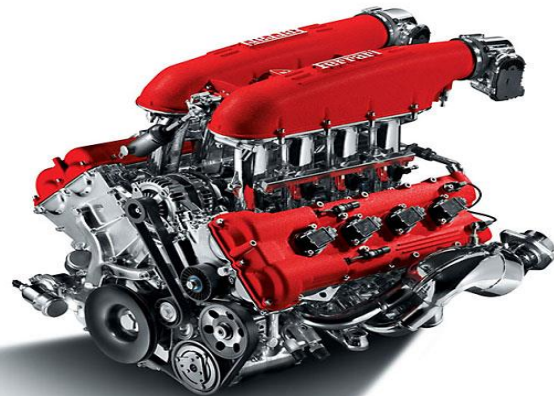
DI : Dependency Injection



DI : DEPENDENCY INJECTION



DI : Dependency Injection



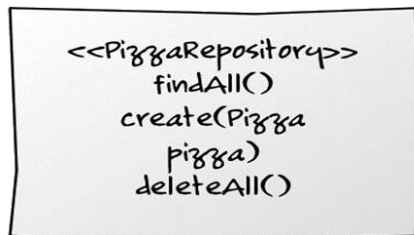
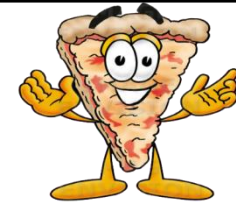
Les changements étant rendus plus aisés grâce l'injection de dépendances, il devient nécessaire de maîtriser leur périmètre !

DbC : Design by Contract

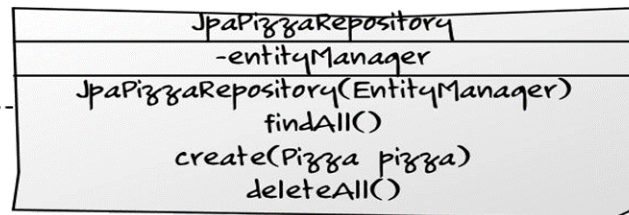


Pizza 3000

DBC : DESIGN BY CONTRACT



L'interface est
garante du contrat

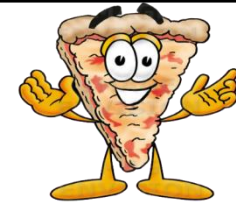


Implémentation technique
"sans impact visible en cas d'évolution"

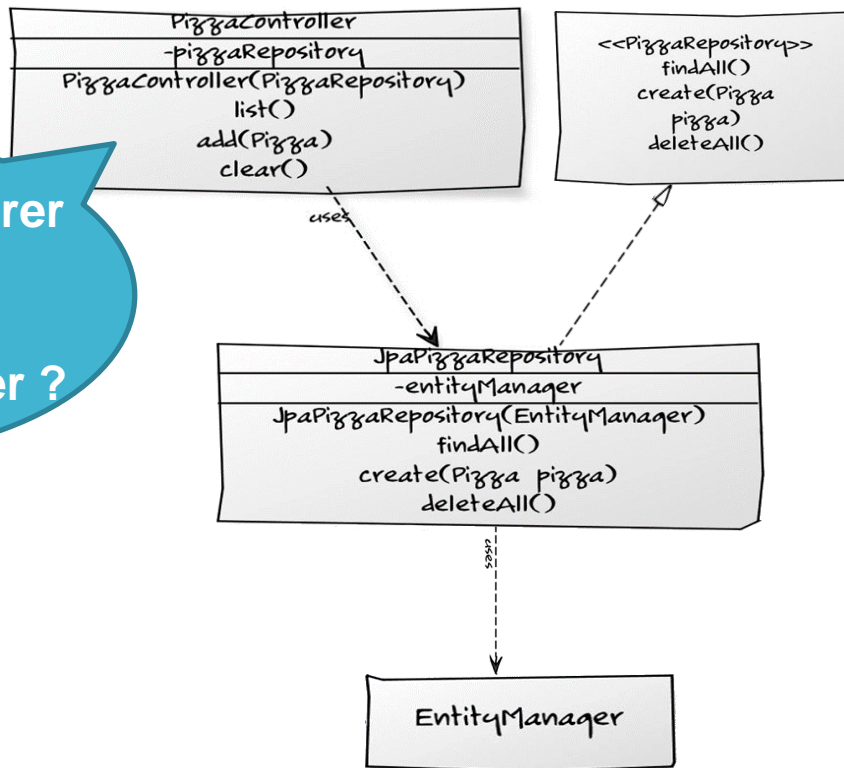


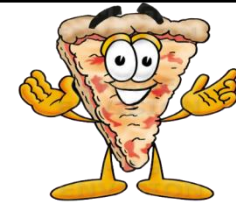
Pizza 3000

PROBLÉMATIQUE ?



Comment récupérer
une instance
de PizzaController ?





L'INSTANCIATION MANUELLE

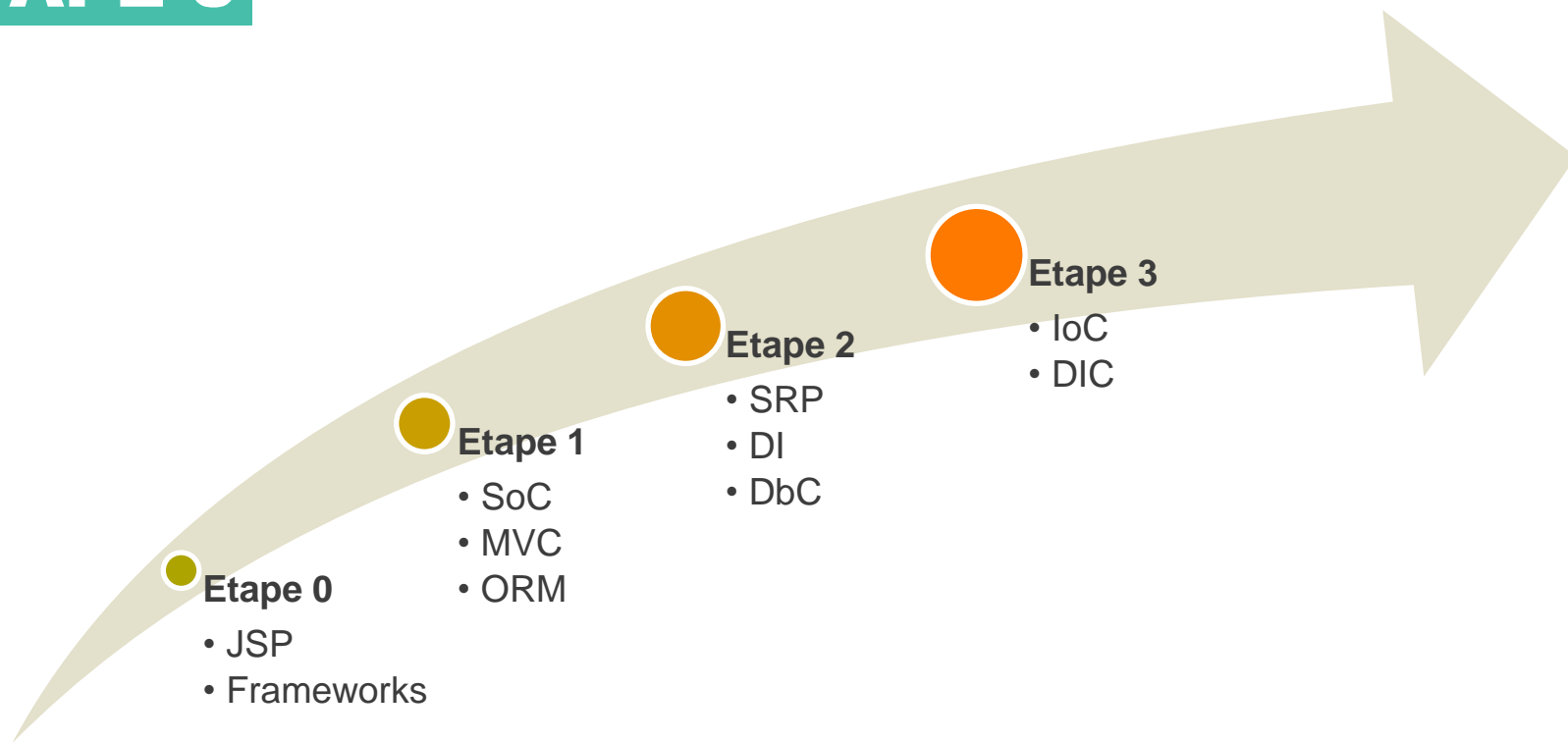
```
EntityManager em = entityManagerFactory.createEntityManager();
```

```
PizzaRepository pizzaRepository = new JpaPizzaRepository(em);
```

```
PizzaController pizzaController = new PizzaController(pizzaRepository);
```

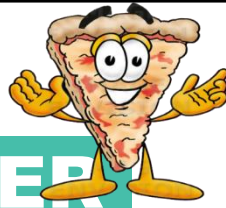

Approche intra applicative

ETAPE 3

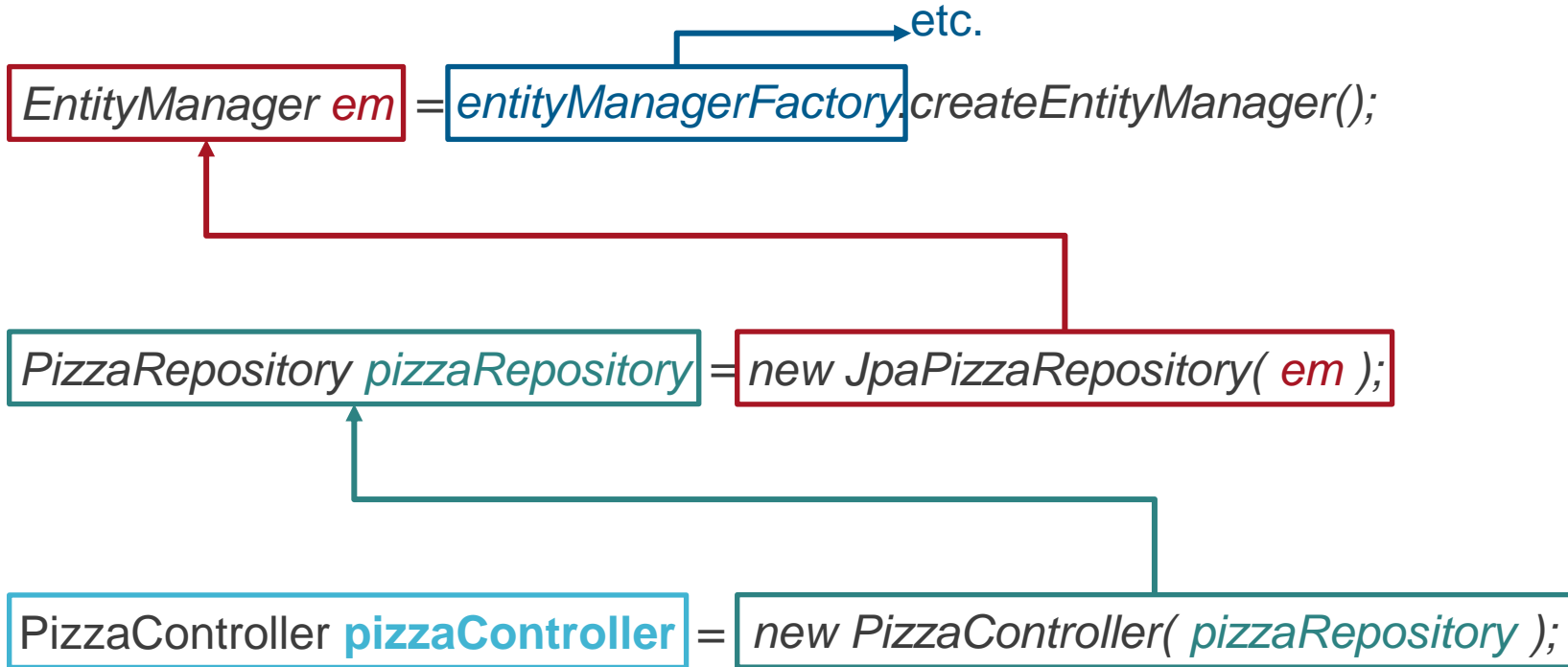


IoC : Inversion of Control





DIC : DEPENDENCY INJECTION CONTAINER



```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/tx
    http://www.springframework.org/schema/tx/spring-tx.xsd">

  <!-- Transaction, datasource, ... beans omitted for brevity -->
  <!-- .... -->

  <bean id="emf" class="org.springframework.orm.jpa.LocalEntityManagerFactoryBean">
    <property name="persistenceUnitName" value="urba"/>
    <!-- Other properties not defined in this example
    <property name="jpaProperties" ref="jpaProperties" />
    <property name="xxx">.../>
    -->
  </bean>

  <bean id="em" class="javax.persistence.EntityManager"
    factory-bean="emf" factory-method="createEntityManager"/>

  <bean id="pizzaRepository" class="com.worldline.demo.urba.repository.JpaPizzaRepository">
    <constructor-arg ref="em"/>
  </bean>

  <bean id="pizzaController" class="com.worldline.demo.urba.controller.PizzaController">
    <constructor-arg ref="pizzaRepository"/>
  </bean>
</beans>
```

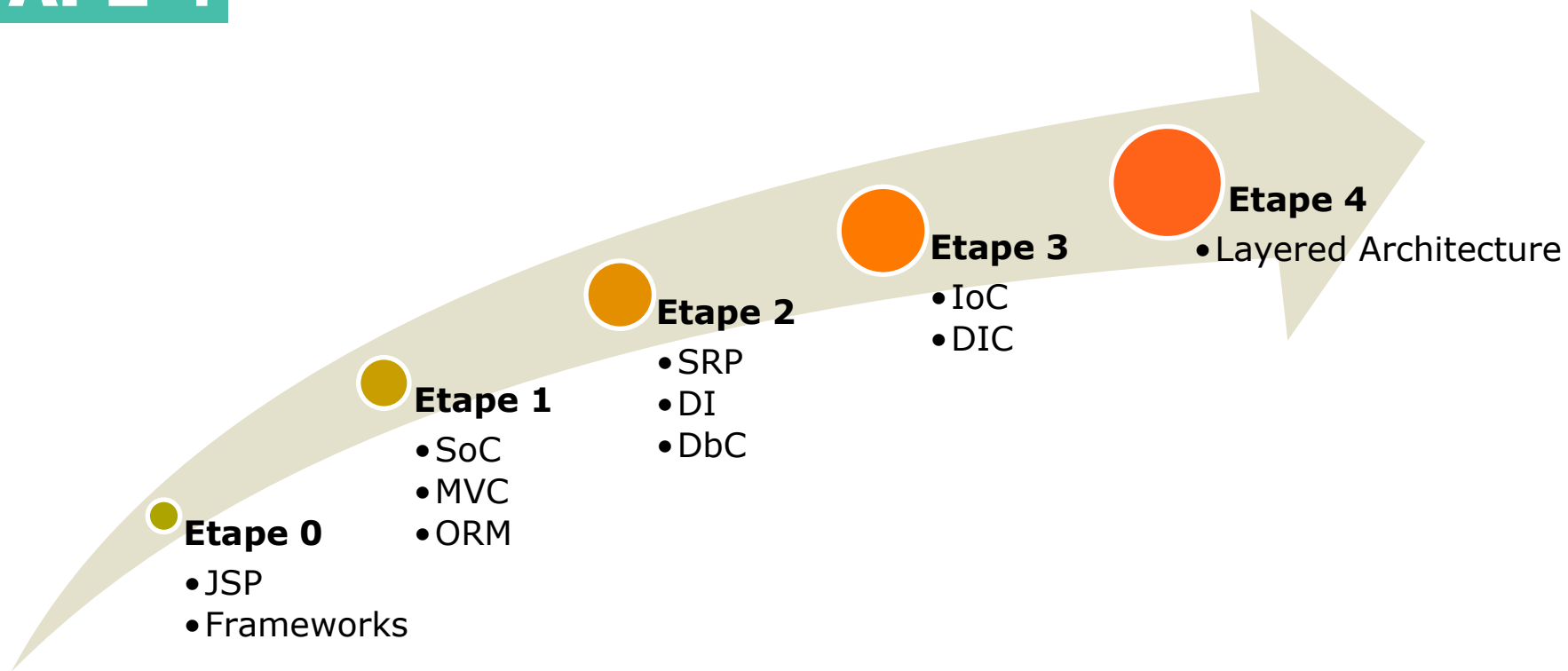
`@Component`

```
public class PizzaController {  
  
    private static final Logger LOGGER = LoggerFactory.getLogger(PizzaController.class);  
    private final PizzaRepository pizzaRepository;  
  
    //Optional since 4.3: implicit constructor injection for single-constructor scenarios  
    @Autowired public PizzaController(PizzaRepository pizzaRepository) {  
        Assert.notNull(pizzaRepository, "pizzaRepository must be not null");  
        this.pizzaRepository = pizzaRepository;  
    }  
  
    public List<Pizza> list() {  
        List<Pizza> pizzas = pizzaRepository.findAll();  
        LOGGER.debug("Pizza(s): {} (nb: {})", pizzas, pizzas.size());  
        return pizzas;  
    }  
  
    public Pizza create(Pizza pizza) {  
        Assert.notNull(pizza, "pizza must be not null");  
        pizza = pizzaRepository.save(pizza);  
        LOGGER.debug("Pizza '{}' successfully created (detail: {})", pizza.getName(), pizza);  
        return pizza;  
    }  
  
    public void delete(Long id) {  
        Assert.notNull(id, "id must be not null");  
        pizzaRepository.deleteById(id);  
        LOGGER.debug("Pizza with id '{}' successfully deleted", id);  
    }  
}
```

Informations pour « décrire/expliquer » à Spring les composants et éventuelles dépendances pour qu'il puisse les gérer pour nous

Approche intra applicative

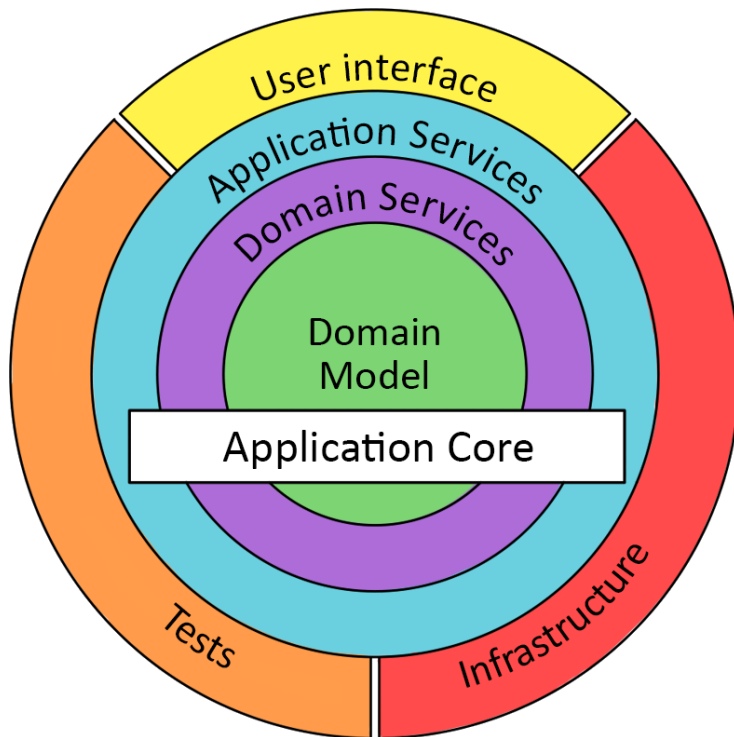
ETAPE 4



Layered Architecture

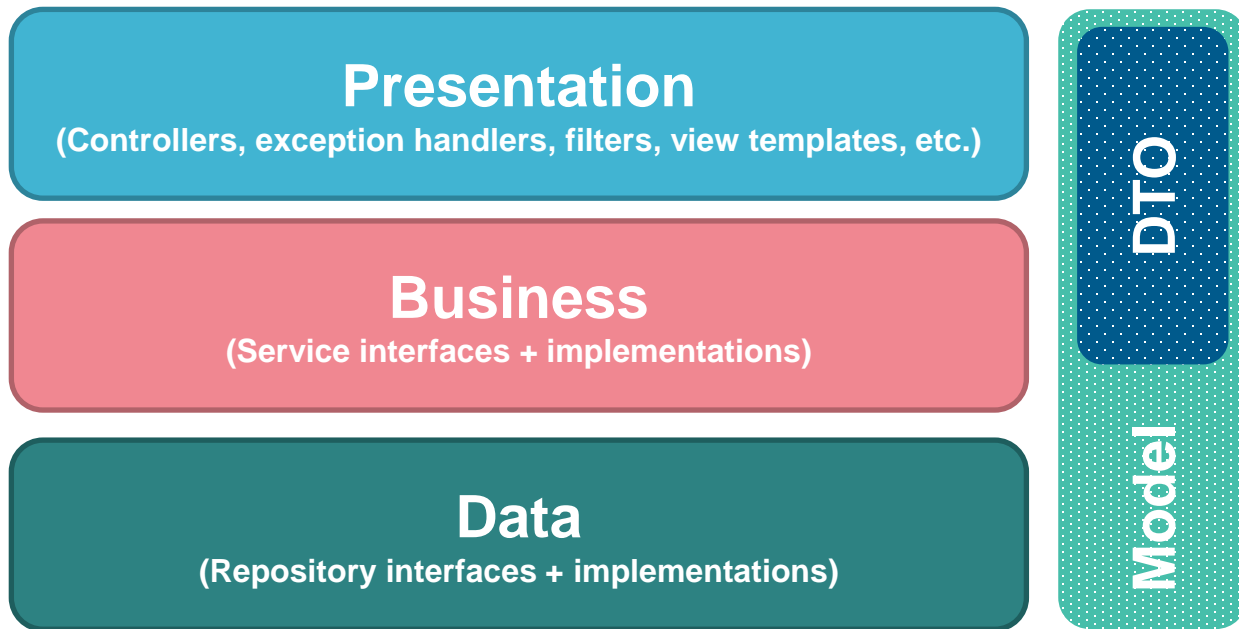
OIGNON STYLE*

**(UNE VISION/VARIANTE POSSIBLE)*



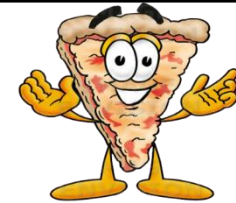
Layered Architecture

N-TIER



Pizza 3000

UNE ORIENTATION "WEB"



PizzaController

Web

(Controllers, exception handlers, filters, view templates, etc.)

PizzaService

Business

(Service interfaces + implementations)

PizzaRepository

Data

(Repository interfaces + implementations)

DTO

Model

Pizza / Ingredients

SRP/DbC & DI/IoC & Layered Architecture



Différents niveaux

INTRA / INTER



Approche
intra applicative
(design pattern)

**Approche inter
applicative**
(architecture)

Approche inter applicative

ETAPE 1

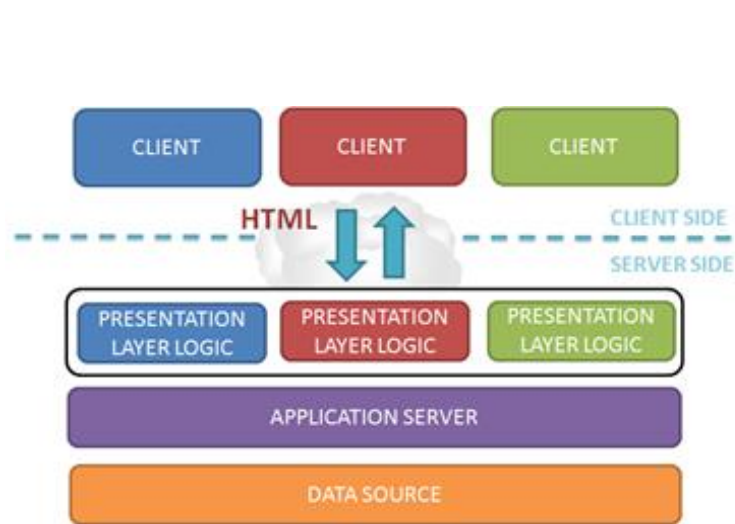


Etape 1

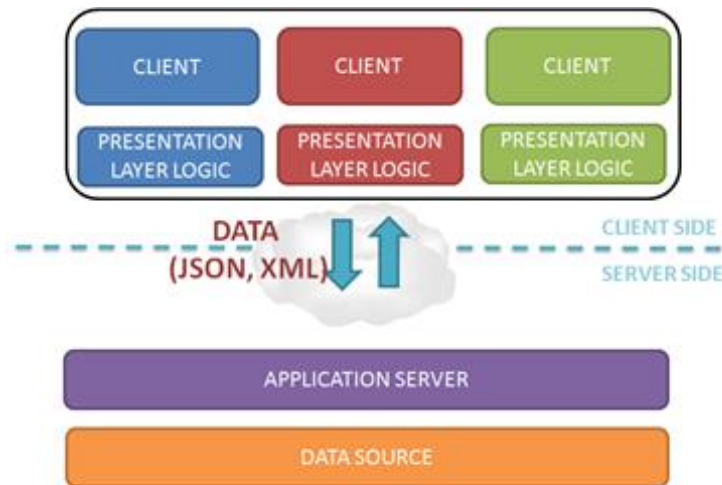
- WebServices

Les WebServices

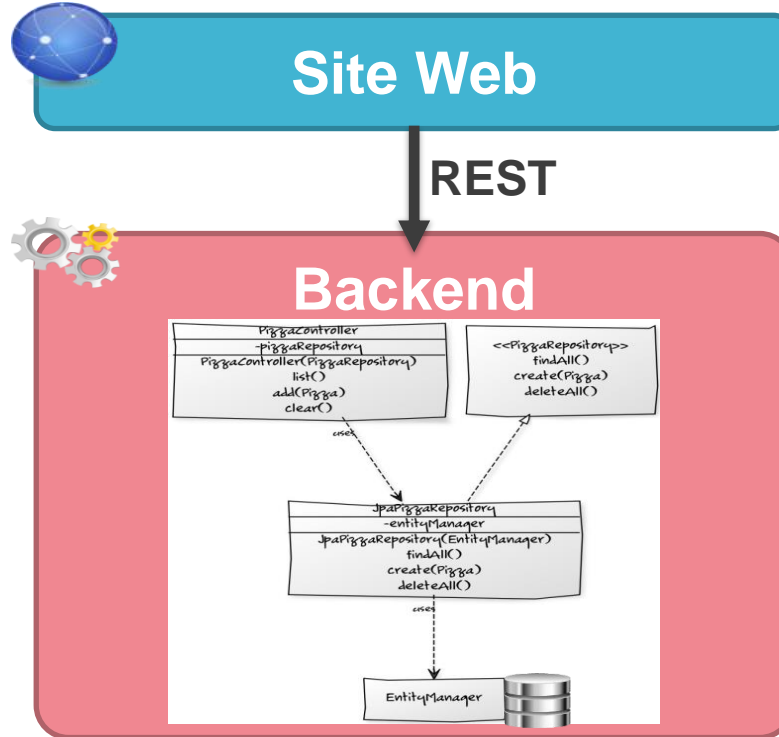
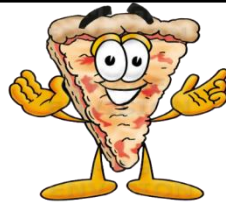
UNE ARCHITECTURE POUR S'OUVRIR / S'EXPOSER



OLD SCHOOL WEB APPLICATION ARCHITECTURE



NEW SCHOOL WEB APPLICATION ARCHITECTURE



@Validated Tips : JSR 303 - Bean Validation

@RestController @RequestMapping("/api/pizzas")

```
public class PizzaController {
```

```
    private static final Logger LOGGER = LoggerFactory.getLogger(PizzaController.class);  
    private final PizzaRepository pizzaRepository;
```

```
    @Autowired //Optional since 4.3: implicit constructor injection for single-constructor scenarios
```

```
    public PizzaController(PizzaRepository pizzaRepository) {  
        Assert.notNull(pizzaRepository, "pizzaRepository must be not null");  
        this.pizzaRepository = pizzaRepository;  
    }
```

Informations pour « décrire/expliquer »
nos API REST à SpringMVC

@GetMapping

```
    public List<Pizza> list() {  
        List<Pizza> pizzas = pizzaRepository.findAll();  
        LOGGER.debug("Pizza(s): {} (nb: {})", pizzas, pizzas.size());  
        return pizzas;  
    }
```

rem : Attention à la sémantique des annotations
(GET vs POST, code de retour, etc.)

@PostMapping @ResponseStatus(HttpStatus.CREATED)

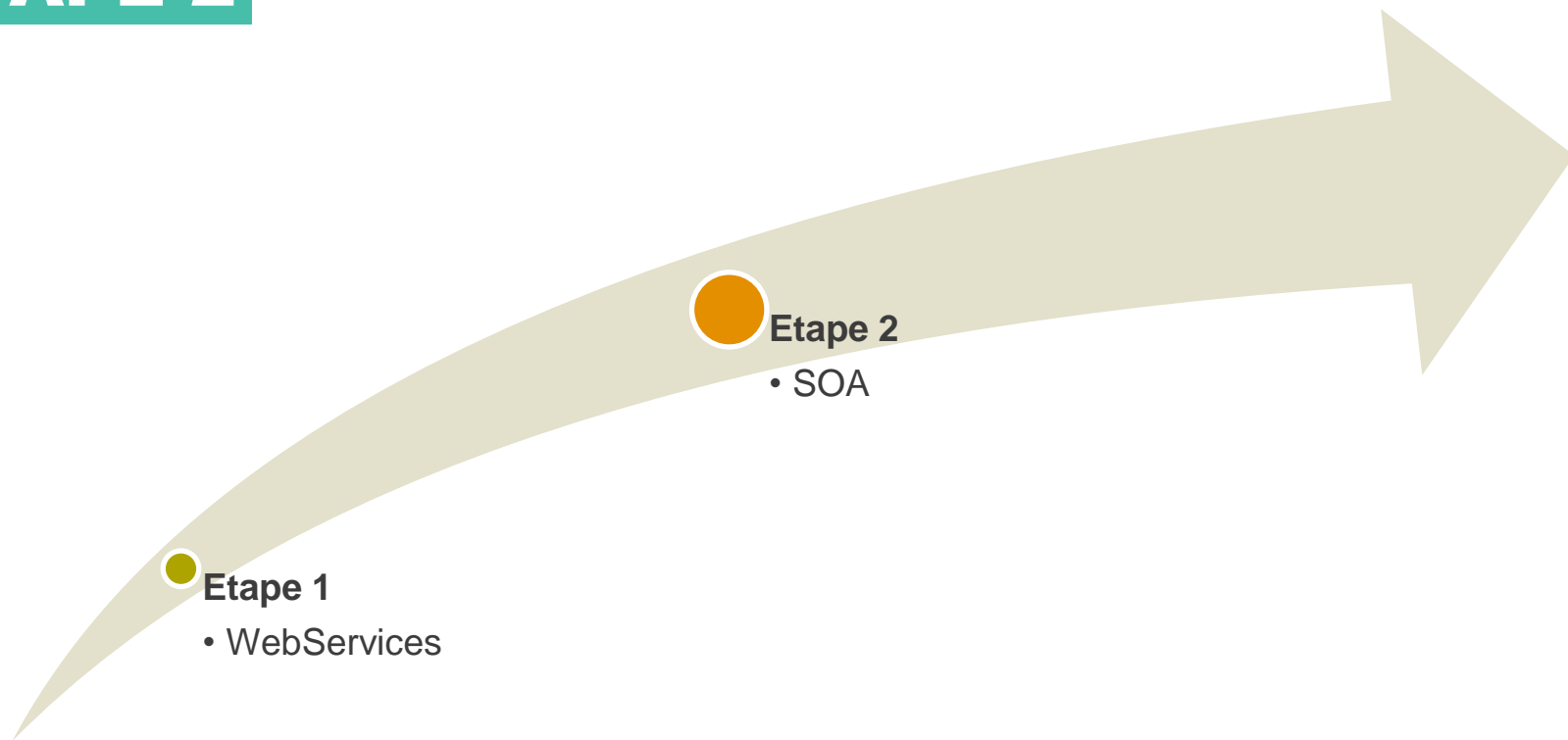
```
    public Pizza create(@Valid @RequestBody Pizza pizza) {  
        pizza = pizzaRepository.save(pizza);  
        LOGGER.debug("Pizza '{}' successfully created (detail: {})", pizza.getName(), pizza);  
        return pizza;  
    }
```

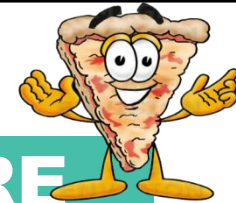
@DeleteMapping(path = "{id}") @ResponseStatus(HttpStatus.NO_CONTENT)

```
    public void delete(@Valid @PathVariable Long id) {  
        pizzaRepository.deleteById(id);  
        LOGGER.debug("Pizza with id '{}' successfully deleted", id);  
    }  
}
```

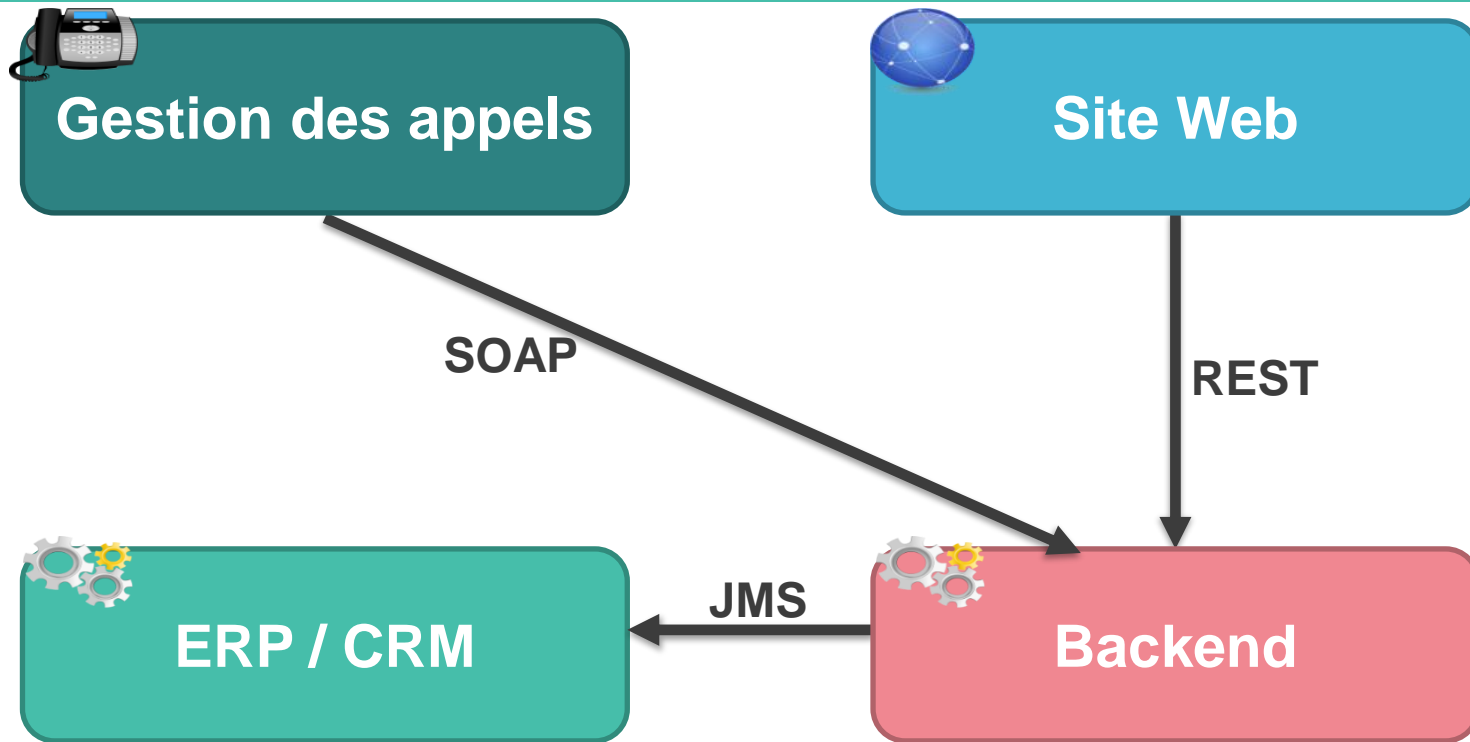
Approche inter applicative

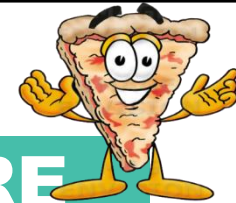
ETAPE 2



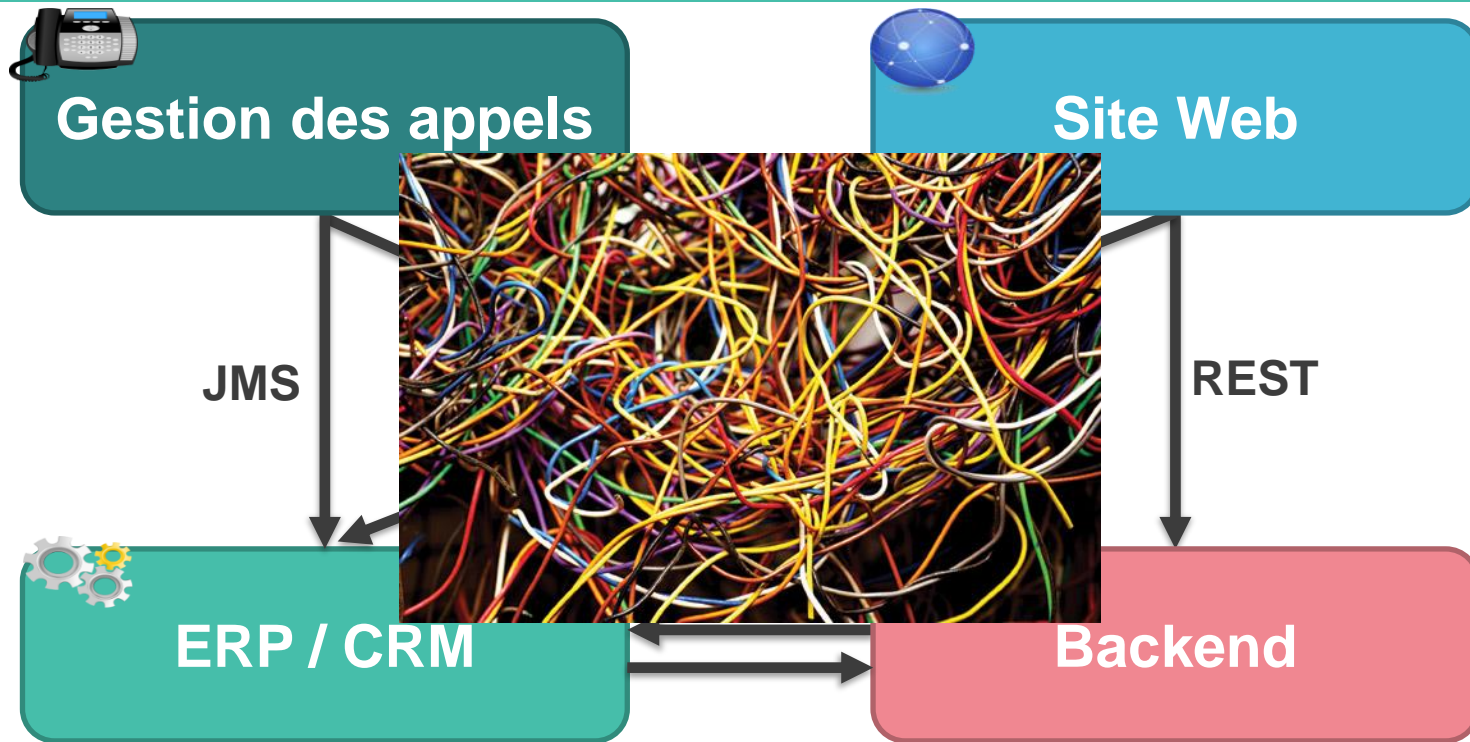


SOA : SERVICE ORIENTED ARCHITECTURE



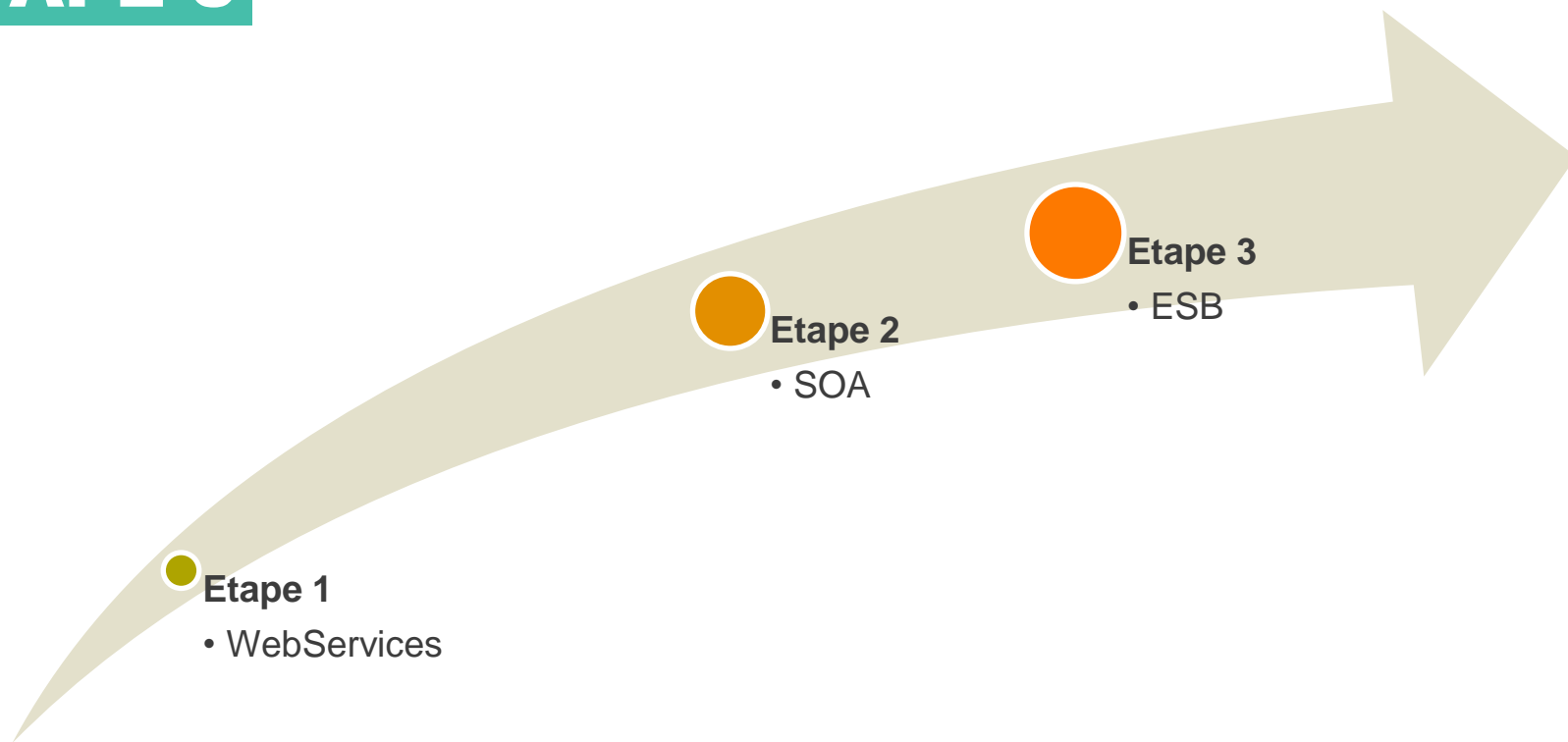


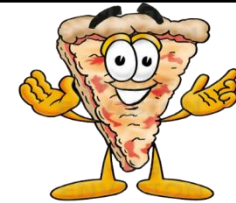
SOA : SERVICE ORIENTED ARCHITECTURE



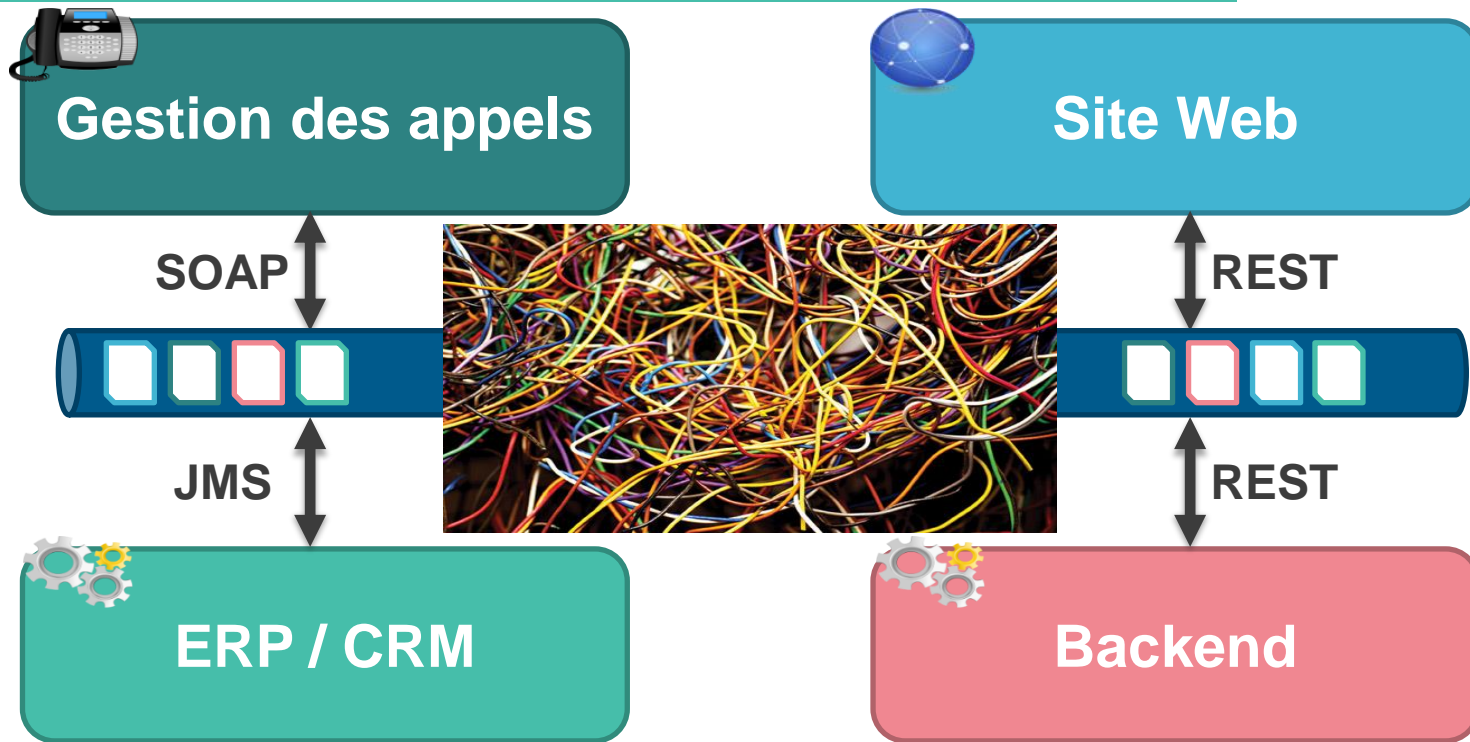
Approche inter applicative

ETAPE 3



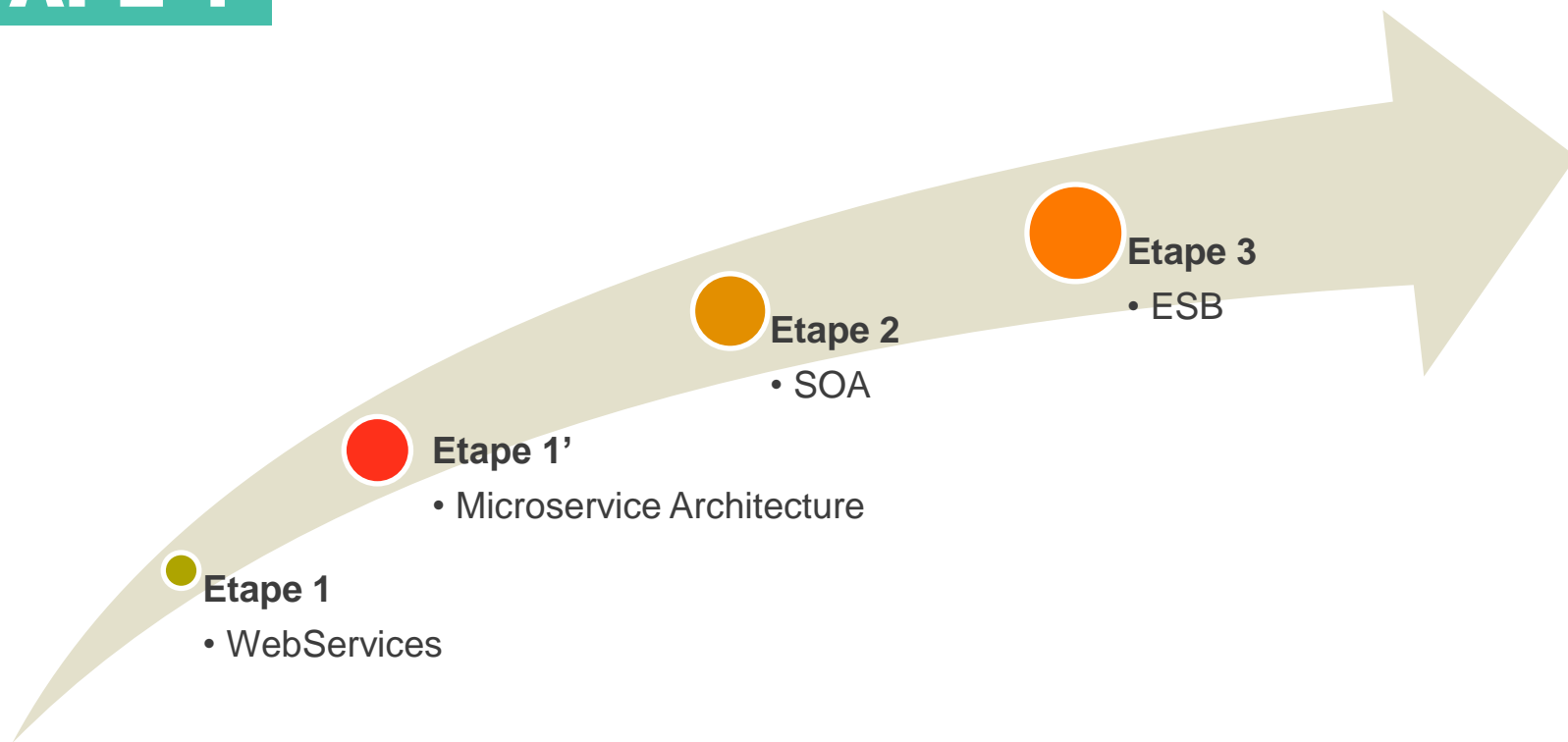


ESB : ENTERPRISE SERVICE BUS



Approche inter applicative

ETAPE 1'



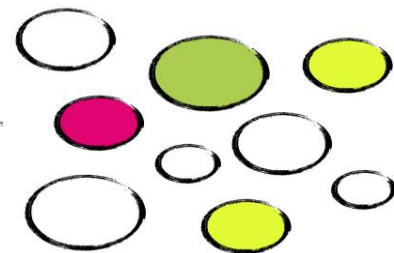
Microservice Architecture

DU MONOLITHE AUX MICROSERVICES (BACK OFFICE)



Vision monolithique

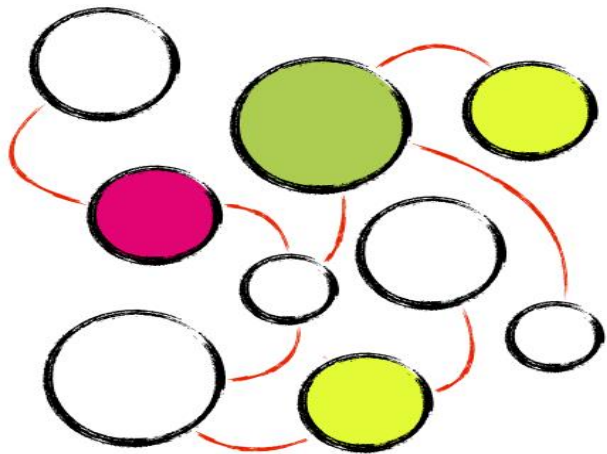
- Scalabilité, simplicité et résilience par "Design"
- Moins d'inertie / Impact limité / Technologie appropriée
- Propice à une vision en unité : Conteneur
- etc.
- Amplifie de nombreuses problématiques (automatisation des livraisons, importances des surveillances, organisation, etc.)
- Importance de l'asynchrone
- Ne résout pas les problématiques d'interconnexion
- etc.



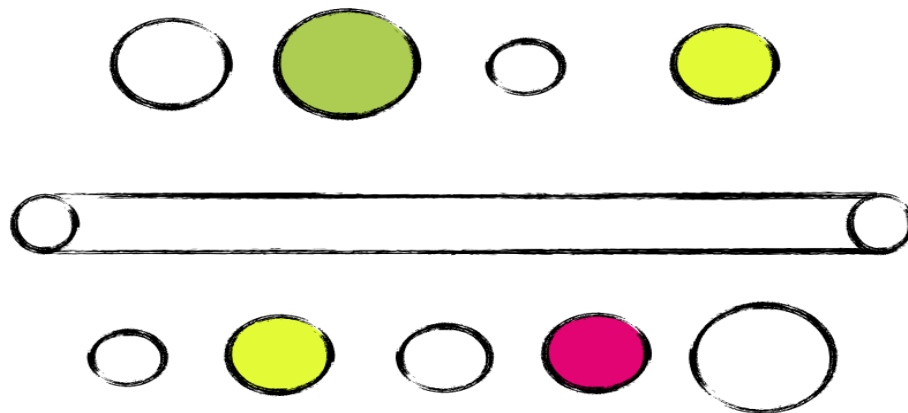
Vision microservice

Microservice Architecture

INTERCONNEXION ?



Fortement couplé

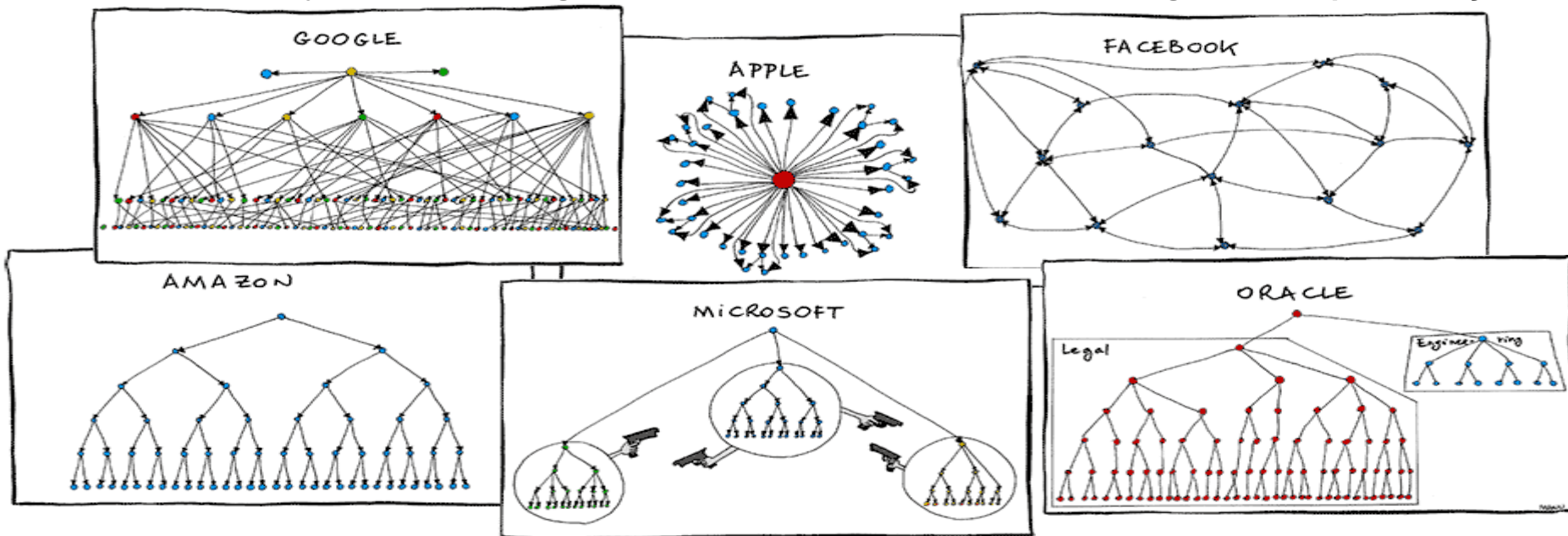


Bus de message

Microservice Architecture

CONWAY'S LAW (1968)

« La structure d'un système est à l'image de la structure de communication de l'organisation qui l'a conçu »



► Suivez le travail Worldline et des Experts

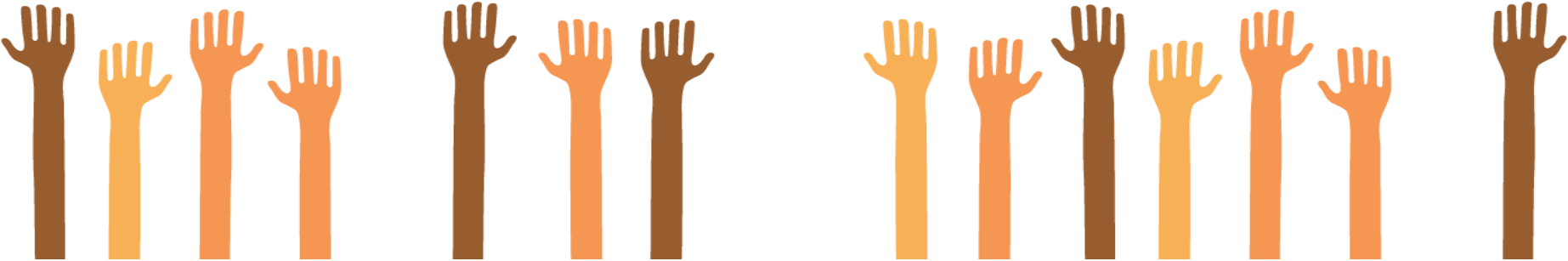
- [@worldlinexpert](https://twitter.com/worldlinexpert)
- <https://blog.worldline.tech>
- <https://github.com/worldline>

► Crédit images

- 1 : [Fourvière de nuit](#)
- 14 : [Theory](#)
- 15 : [Dictionnaire](#)
- 20 : [The human gait](#)
- 22 : [Goals](#)
- 58 : [Xkcd - Exploit of a Mom](#)
- 63 : [Bill Brookman](#)

- 79 : [Onion architecture is interesting](#)
- 85 : [The new distribution of the 3tiered architecture changes everything](#)
- 964 : [Organizational-charts](#)

Questions ?



MERCI
pour votre attention

