

## Génie Logiciel : Conception Propriétés Architecturales

Sébastien Mosser  
INF-5153, Hiver 2019, Cours #2.3



# Architecture?

“ The **art** or **science** of building; esp. the art or practice of **designing** and **building** edifices for **human use**, taking both **aesthetic** and **practical factors** into account.

[Oxford English Dictionary, 2002]

“ We speak of the  
“**architecture**” of a symphony,  
and call architecture, in turn,  
“**frozen music**”.

[Deryck Cooke, the Language of Music]

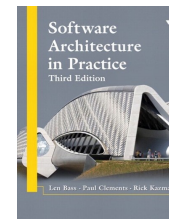
# Architecte ?



“ Beautiful  
architectures  
do **more** with **less**. ”

[Stephen J. Mellor]

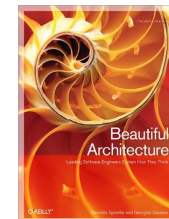
## Bibliographie



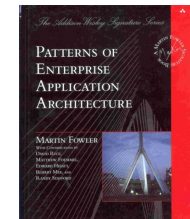
[SAIP, 2012]



[SSA, 2011]



[BA, 2009]



[PoEAA, 2002]

## Architecture Logicielle : Définition

“The **structure** of the system, which comprise **software elements**, externally **visible properties** of those elements, and the **relationships** among them.

[SAiP]

## Architecture versus Conception ?

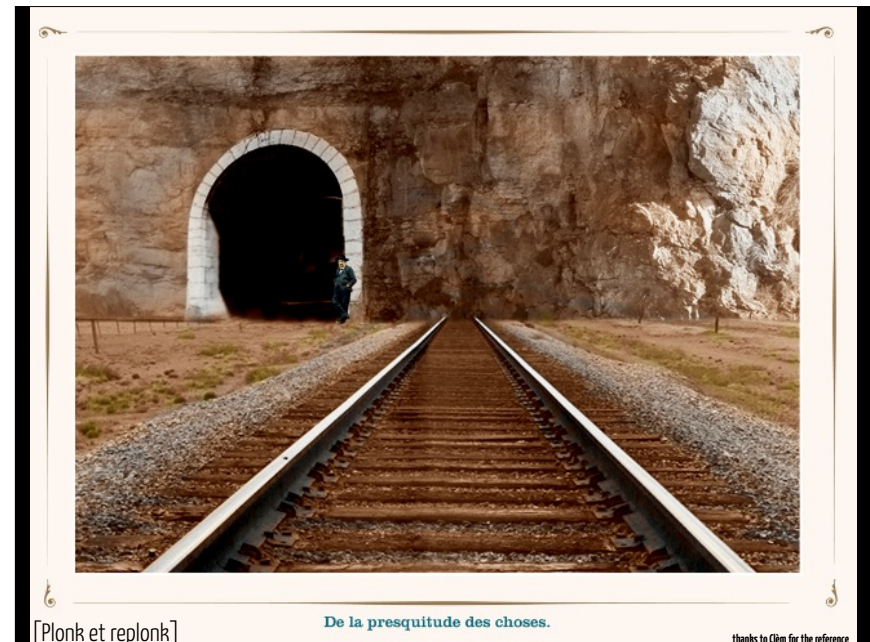
Architecture  $\subset$  Conception

Conception “Externe”

## Objectifs de l'Architecture

- Le système possède les **fonctionnalités requises** par le client;
- Il est **constructible de manière sûre** et dans les délais;
- Ses **fonctionnalités** font ce qu'**elles sont censées faire**;
- Il est **fiable**;
- Il est **utilisable** par l'audience ciblée;
- Il est **abordable**, pour sa construction et sa maintenance.

[BA]



[Plonk et replonk]

De la presquitude des choses.

thanks to Clém for the reference

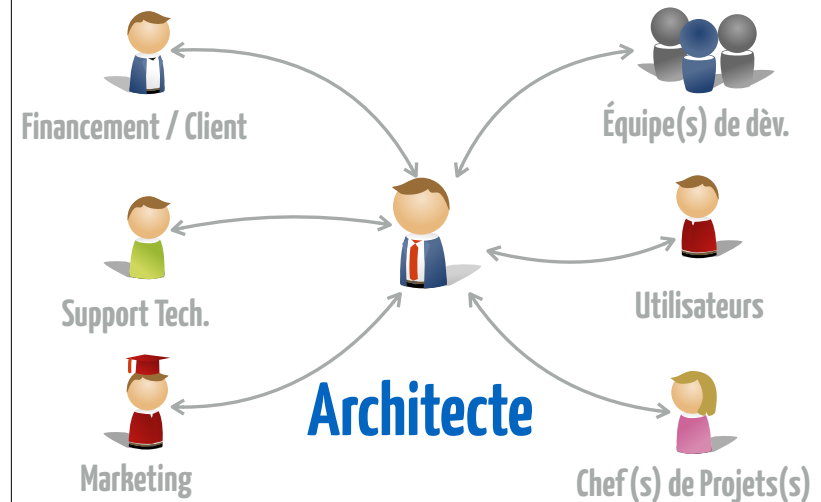
## Préoccupations d'Architecture



[BA]

13

## L'architecte et son environnement



**Prévoir** ce qui est **Imprévisible** !

“ Parameters that were  
**never** going to **change**  
now need to be **modified**. ”

[BA]

1 Règles d'or architectures

Principes des points de vues 2

3 Point de vue Fonctionnel

Point de vue Développement 4

# Architecture

## Règles d'Or



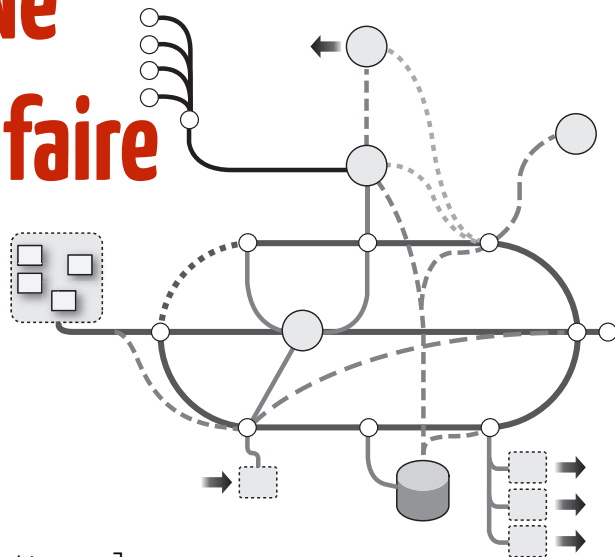
[Beautiful Architecture]

“

Software **architecture** is  
**not set in stone.**  
**Change** if you need it.

[Beautiful Architecture]

**Ne  
pas faire**



[Beautiful Architecture]

19

“

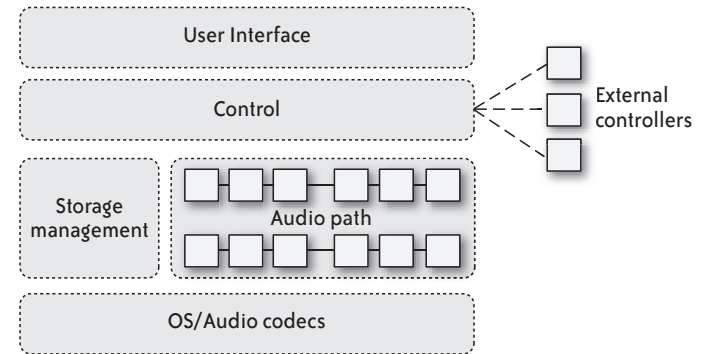
A **fuzzy architecture**  
leads to **individual** code,  
**duplication** of code and effort.

[Beautiful Architecture]

“**Bad** architectural **design**  
leads to further  
**bad** architectural **design**.”

[Beautiful Architecture]

**Correct**



[Beautiful Architecture]

“A **clear architectural**  
design leads to a  
**consistent system**.”

[Beautiful Architecture]

## **Cohésion** versus **Couplage**

### **Cohésion** :

"how related functionality is gathered together".

### **Couplage** :

"Measurement of interdependency between modules".

[Beautiful Architecture]

X
?

Faible cohésion,  
Fort couplage

Forte cohésion,  
Faible couplage

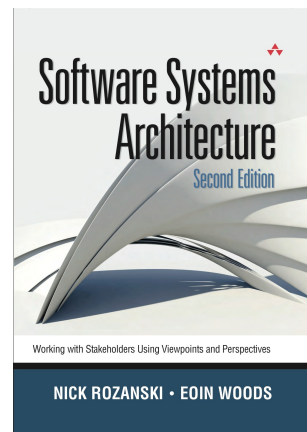
[Beautiful Architecture]

# Principes des Points de Vue

# 2

## Bibliographie

- Chapitres
- #3: Viewpoints and Views
- #17: Functional Viewpoint
- #20: Development Viewpoint



[SSA, 2011]

“A **complex system** is much more  
**effectively** described by **a set of**  
**interrelated views** [...] than by  
**a single overloaded model**

[SSA]

“ A **view** is a representation of one or more **structural aspect** of an **architecture** ”

[SSA]

“ A **viewpoint** is a **collection** of **patterns, templates, and conventions** for **constructing one type of view** ”

[SSA]

## Avantages & Inconvénients

---

Séparation des  
préoccupations

Focus sur des  
sous-parties du système

Communication avec  
les parties prenantes

Gestion de la  
complexité

[SSA]

31

## Avantages & Inconvénients

---

Inconsistence

Définition des  
mauvaises vues  
(inutilité / coûts)

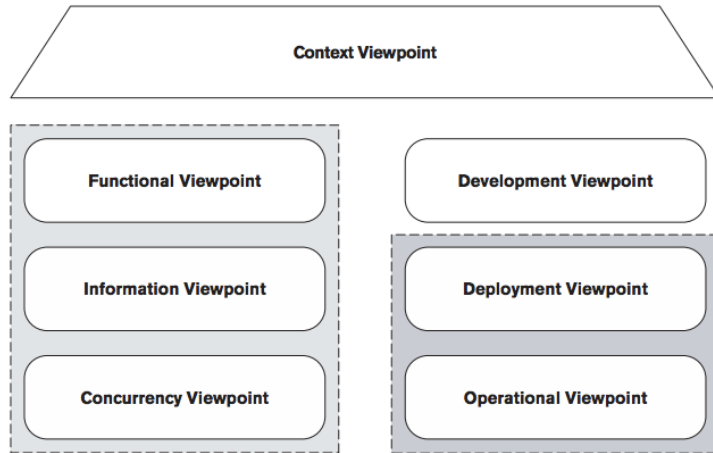
Fragmentation

[SSA]

32



# Catalogue de Points de Vue



[SSA]

# Importance des points de vues

	Information System	Middleware	High-volume website	Enterprise package
Context	+	-	~	~
Functional	+	+	+	+
Information	~	-	~	~
Concurrency	-	+	~	+/-
Development	+	+	+	+
Deployment	+	+	+	+
Operational	+/-	-	~	+

[SSA]

34

Point de vue  
**Fonctionnel**

3

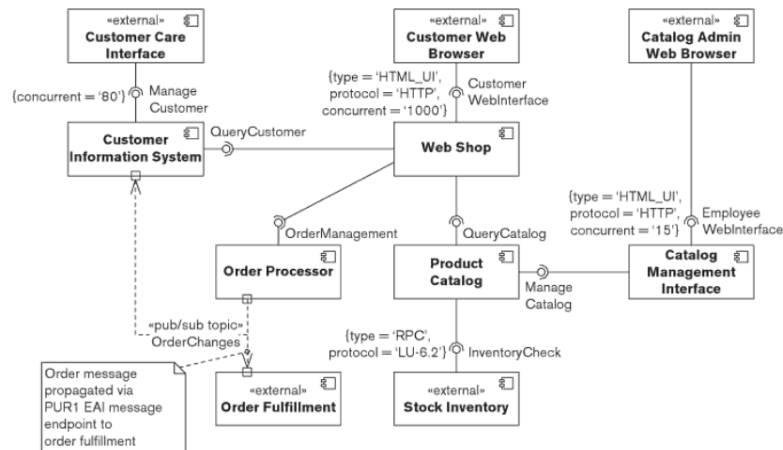
# Définition

“ Describes the system’s runtime **functional elements** and their **responsibilities, interfaces** and **primary interactions** ”

[SSA]

36

# Travail aux Interfaces



[SSA]

37

# Processus d'élicitation

Exigences → Interfaces

1. Identifier les concepts
2. Distribuer les responsabilités
3. Concevoir les interfaces
4. Concevoir les connecteurs
5. Tracer avec les fonctionnalités
6. Valider les scénarios d'acceptation
7. Analyser les traces d'exécution
8. Évaluer la flexibilité

[SSA]

# Liste de contrôle

- Maximum de 15-20 éléments de premier niveau
- Noms, responsabilités et interfaces clairement définies
- Toutes les interactions passent par des interfaces adéquates
- Compromis Cohésion / Couplage acceptable
- Scénarios d'acceptation validés
- Les scénarios d'acceptation couvrent les exigences métier
- Les points fixe et les points de variations sont identifiés

(tout ceci reste "subjectif", c'est toujours de la conception)

Point de vue  
**Développement**

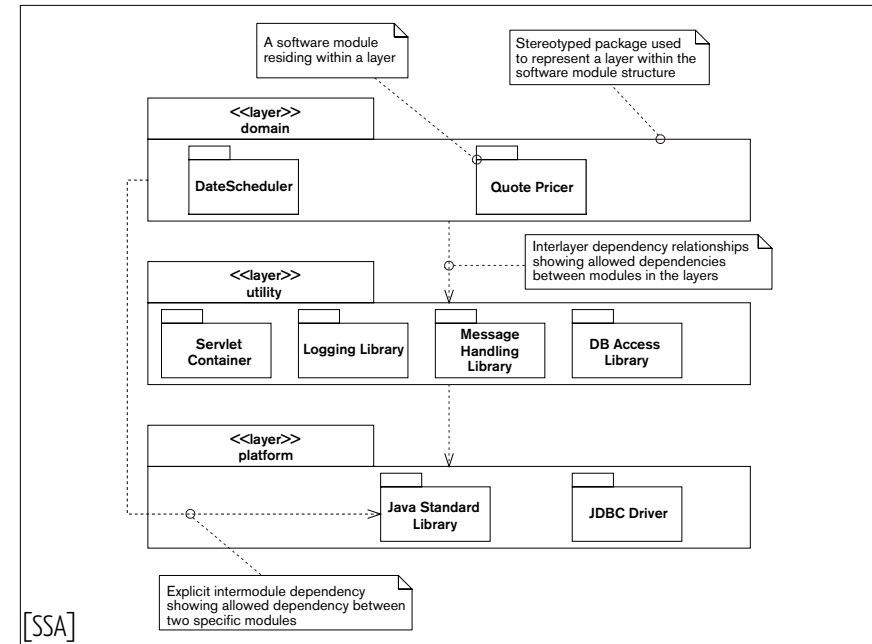
4

## Définition

“ Describes the **architecture** that supports the **software development process** ”

[SSA]

41



## Processus d'élicitation

1. Identifier les modules
2. Classifier les modules
3. Identifier les dépendances entre modules
4. Identifier les "couches" applications

Exigences



Modules

[SSA]

## Erreurs classiques

- Trop de détails
- Description architecturale surchargée
- Niveau de détail inégal (incohérent)
- Manque de focus
- Manque de précisions

**Nécessite de l'expertise**

[SSA]

