

Génie Logiciel : Conception Architecture Logicielle (intro)

Sébastien Mosser
INF-5153, Hiver 2019, Cours #4.2



1

Architecture?

2

“ 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]

3

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

[Deryck Cooke, the Language of Music]

4

Architecte ?

5

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

[Stephen J. Mellor]

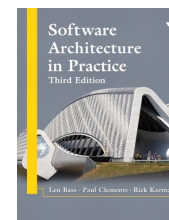
7



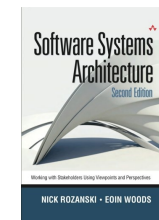
Edna Mode [Les indestructibles]

6

Bibliographie



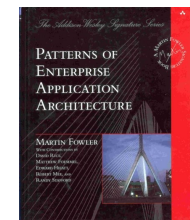
[SAIP, 2012]



[SSA, 2011]



[BA, 2009]



[PoEAA, 2002]

8

Architecture Logicielle : Définition

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

[SAiP]

9

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]

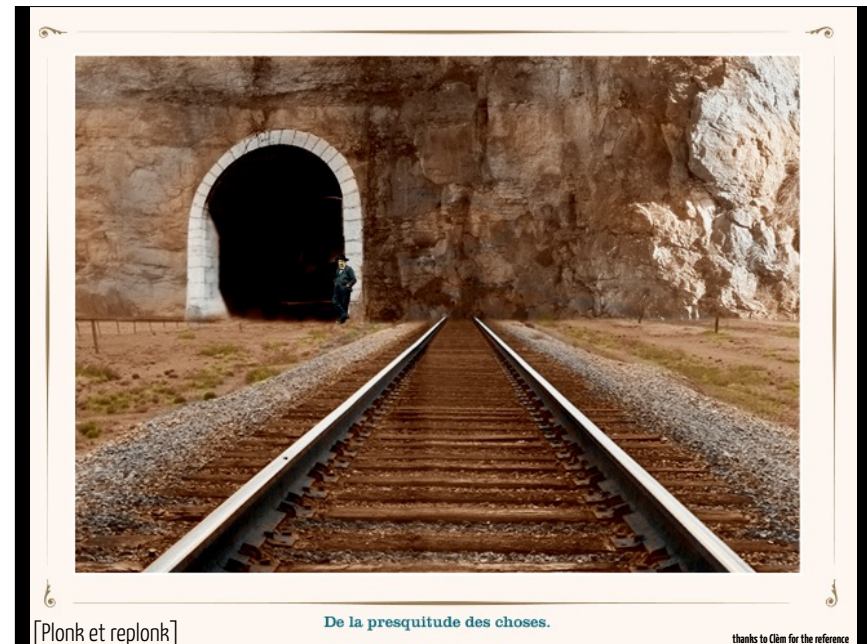
11

Architecture versus Conception ?

Architecture \subset Conception

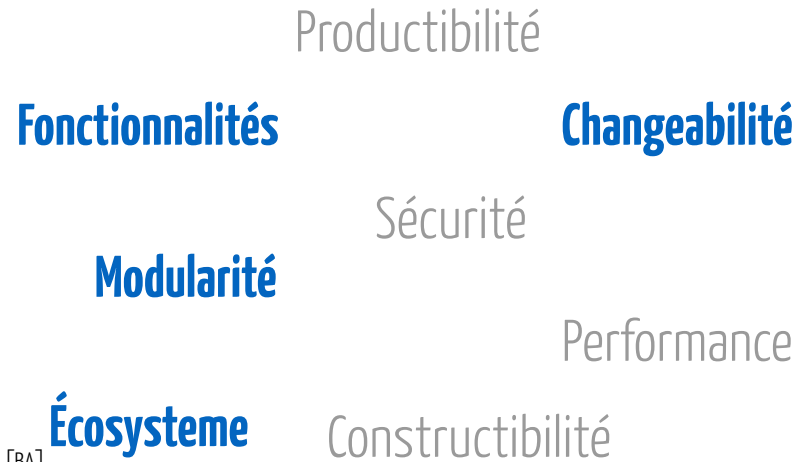
Conception “Externe”

10



12

Préoccupations d'Architecture

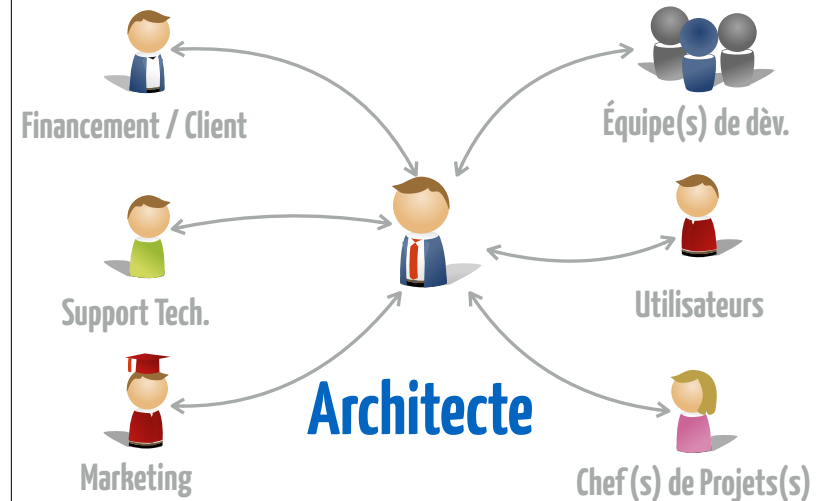


[BA]

13

13

L'architecte et son environnement



14

Prévoir ce qui est **Imprévisible** !

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

[BA]

15

1 Règles d'or architectures

Principes des points de vues **2**

3 Point de vue Fonctionnel

Point de vue Développement **4**

16

Architecture

Règles d'Or

[Beautiful Architecture]

17

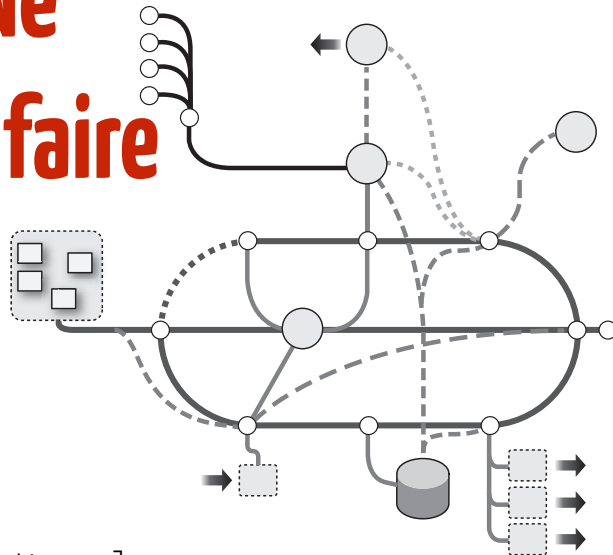
“

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

[Beautiful Architecture]

18

**Ne
pas faire**



[Beautiful Architecture]

19

19

“

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

[Beautiful Architecture]

20

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

[Beautiful Architecture]

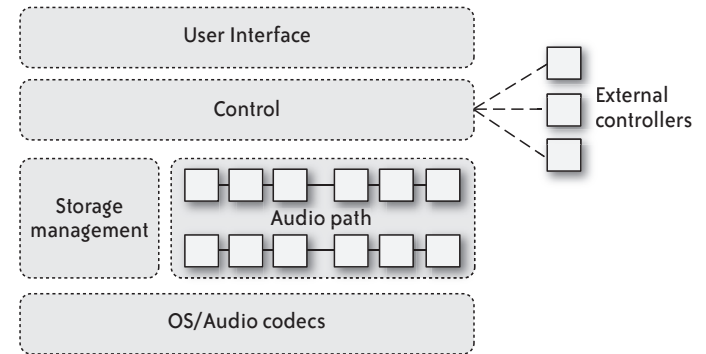
21

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

[Beautiful Architecture]

23

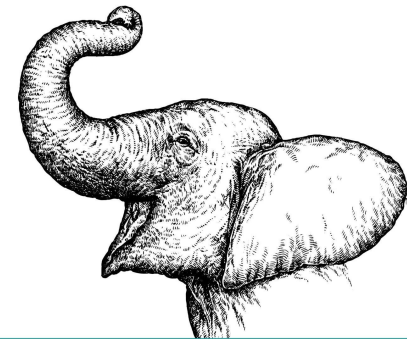
Correct



[Beautiful Architecture]

22

The answer to every programming question ever conceived



It Depends

The Definitive Guide

O RLY?

@ThePracticalDev

24

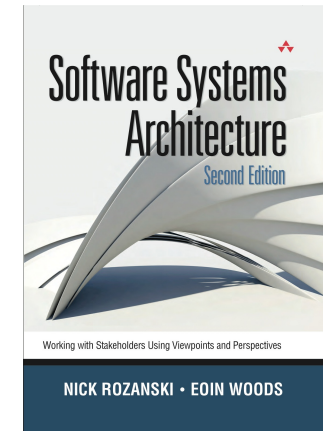
Principes des Points de Vue

2

25

Bibliographie

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



[SSA, 2011]

26

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

[SSA]

27

27

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

[SSA]

28

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

[SSA]

29

Avantages & Inconvénients

Séparation des
préoccupations

Focus sur des
sous-parties du système

Communication avec
les parties prenantes

Gestion de la
complexité

[SSA]

30

30

Avantages & Inconvénients

Inconsistence

Définition des
mauvaises vues
(inutilité / coûts)

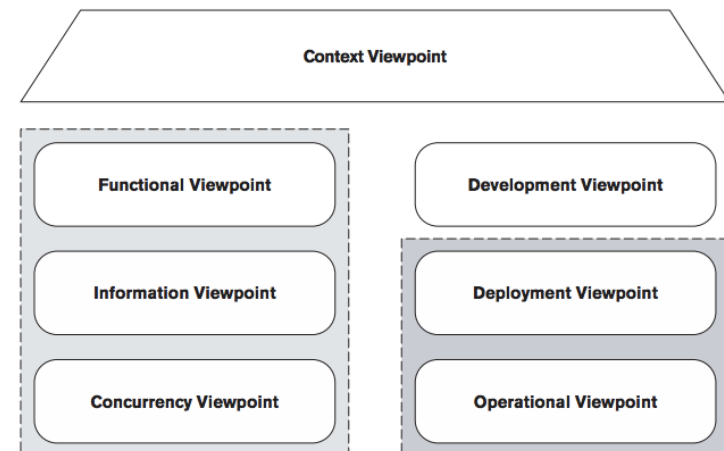
Fragmentation

[SSA]

31

31

Catalogue de Points de Vue



[SSA]

32

Importance des points de vues

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

[SSA]

33

33

Point de vue Fonctionnel



34

Définition

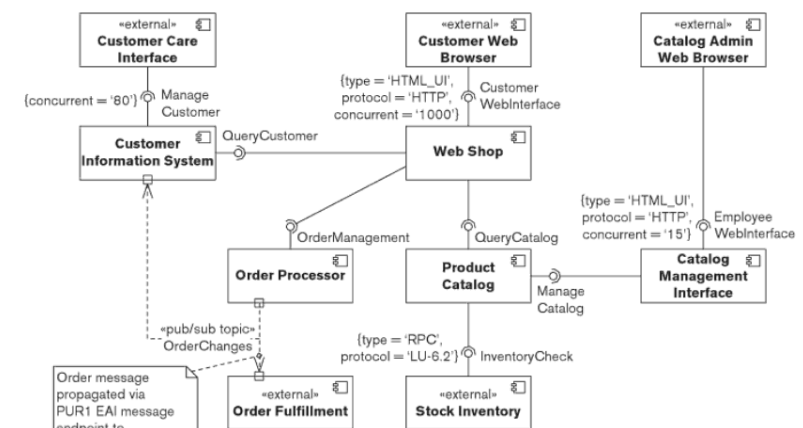
“ Describes the system’s runtime **functional elements** and their **responsibilities, interfaces** and **primary interactions** ”

[SSA]

35

35

Travail aux Interfaces



[SSA]

36

36

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]

37

Point de vue
Développement

4

39

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)

38

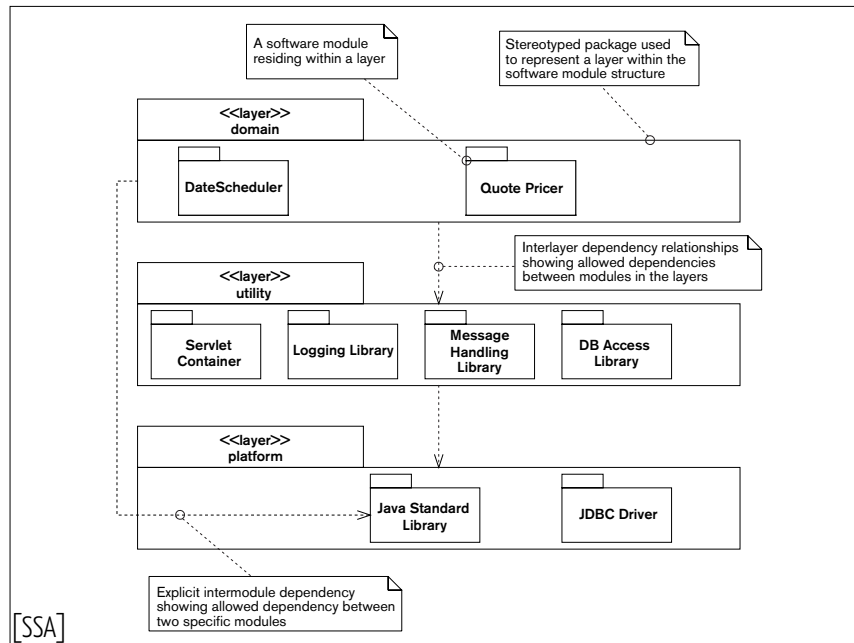
Définition

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

[SSA]

40

40



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]

42

Erreurs classiques

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

Nécessite de l'expertise

[SSA]

43