

Cours d'architecture logicielle

Évaluation d'architectures

Lydie du Bousquet

Lydie.du-Bousquet@imag.fr

Philippe Lalanda

Rappel

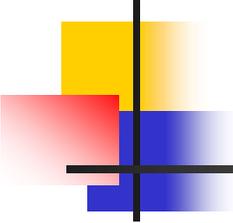
Approche pour la conception

- Dérivation des composants (analyse fonctionnelle)
- Utilisation de patterns
 - patterns d'architecture (ou styles) pour structurer
 - patterns (ou tactiques) de conception pour les propriétés NF
- Évaluation
 - Consistance (syntaxique) de la solution
 - Evaluation de la cohésion et du couplage
 - Revue d'architecture

Rappel

Approche pour la conception

- Dérivation des composants (analyse fonctionnelle)
- Utilisation de patterns
 - patterns d'architecture (ou styles) pour structurer
 - patterns (ou tactiques) de conception pour les propriétés NF
- Évaluation
 - Consistance (syntaxique) de la solution
 - **Evaluation de la cohésion et du couplage**
 - Revue d'architecture



Cohésion et couplage

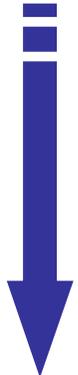
- Deux critères importants pour l'évaluation d'une architecture :
 - La cohésion de chaque module
 - Le couplage entre les modules
- Évaluer la cohésion d'un module c'est répondre à la question
Pourquoi les éléments d'un module ont-ils été mis ensemble ?
- Évaluer le couplage d'un module c'est répondre à la question
Quels liens existent entre ces deux modules ?

La cohésion

- « **largeur** » **fonctionnelle** d'un module
- Cohérence de ce module
 - c'est une mesure **qualitative**
- Une grande cohésion favorise l'évolution
 - facilite la recherche du module à **modifier**
 - **concentre** l'ensemble des modifications dans un **minimum** de modules

4. Cohésion et couplage

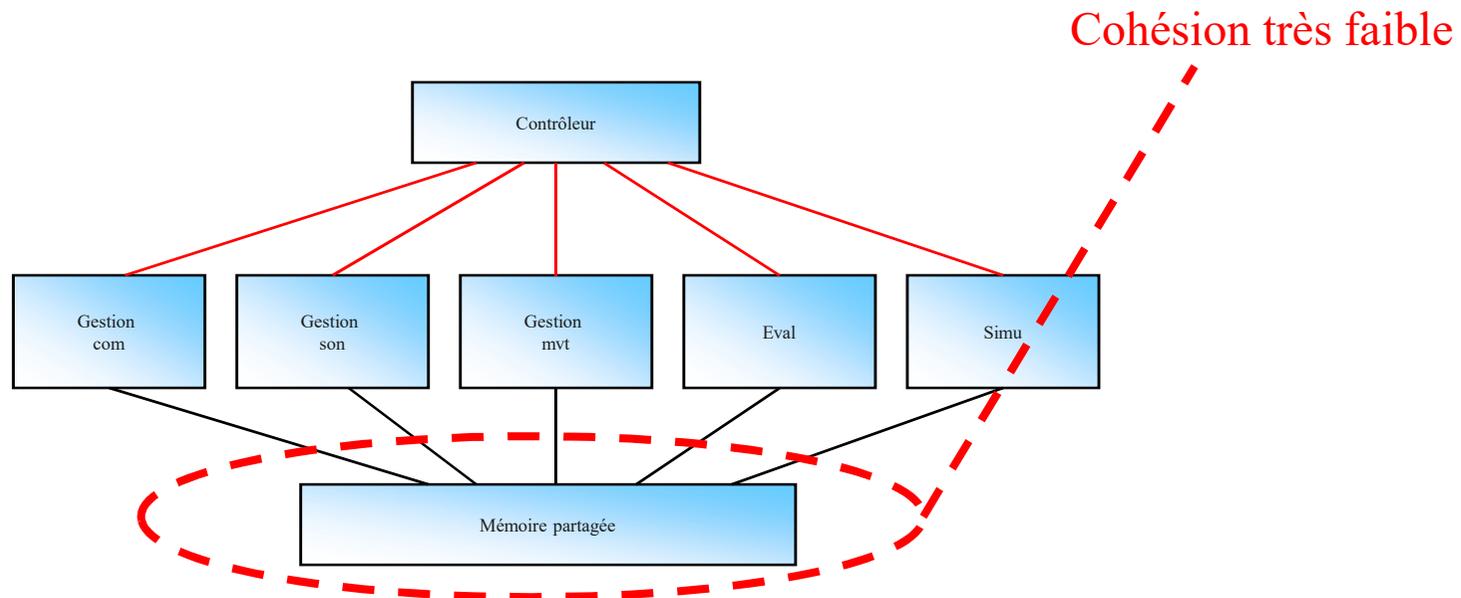
Niveaux de cohésion

- Au niveau architectural, on peut distinguer quatre types de cohésion
 - accidentelle
 - syntaxique
 - temporelle
 - fonctionnelle
- 
- Mauvais**
- Excellent**

4. Cohésion et couplage

Cohésion accidentelle

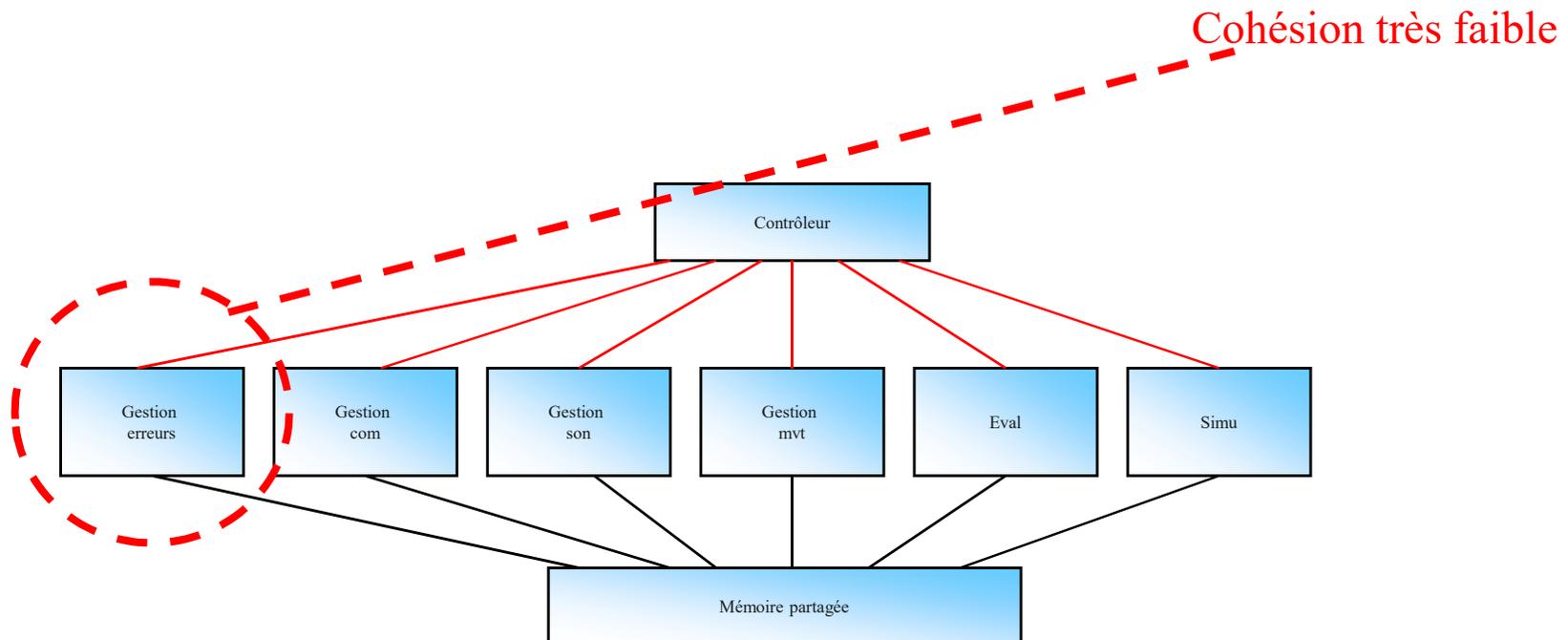
- Les éléments d'un composant sont regroupés sans logique apparente, voire « au hasard »
 - modules de type « fourre-tout » (misc)
 - entrepôts de données utilisés par de nombreux composants



4. Cohésion et couplage

Cohésion syntaxique

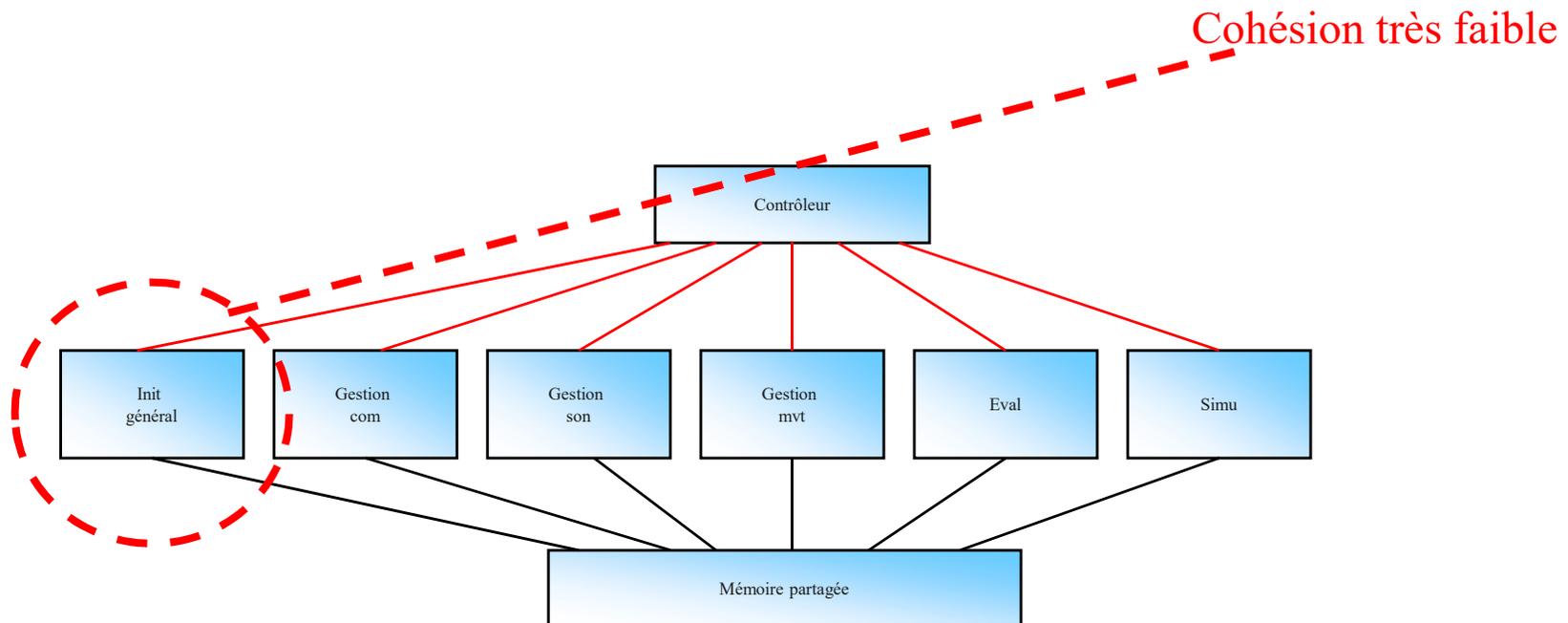
- Les éléments d'un composant sont groupés sur une base « syntaxique »
 - Par nom de classes ou de procédures
 - Par type de classes (gestion des erreurs, gestion des communication)



4. Cohésion et couplage

Cohésion temporelle

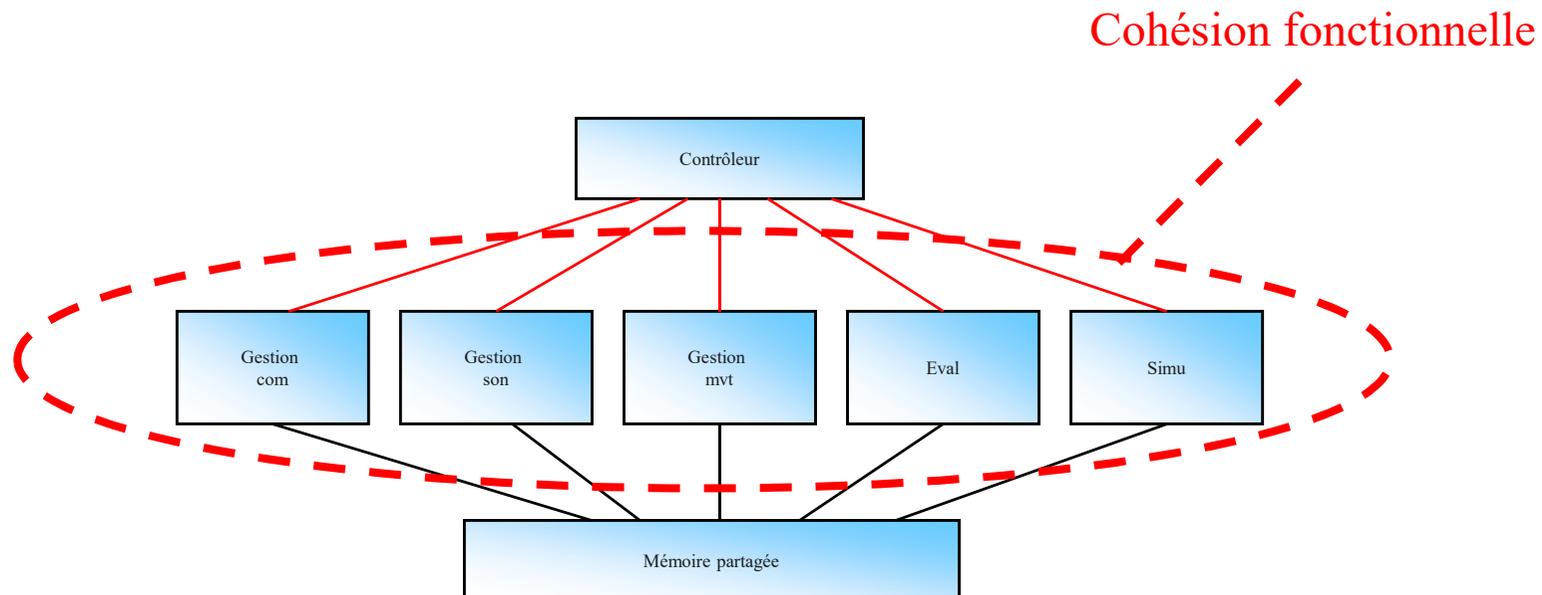
- Les éléments d'un composant sont groupés sur une base **temporelle**
 - éléments qui sont activés au même moment
 - éléments qui sont utilisés au même moment

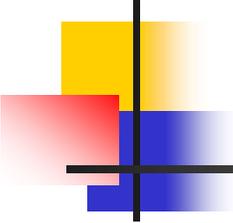


4. Cohésion et couplage

Cohésion fonctionnelle

- Un composant regroupe tous les éléments nécessaires à l'exécution d'une fonction

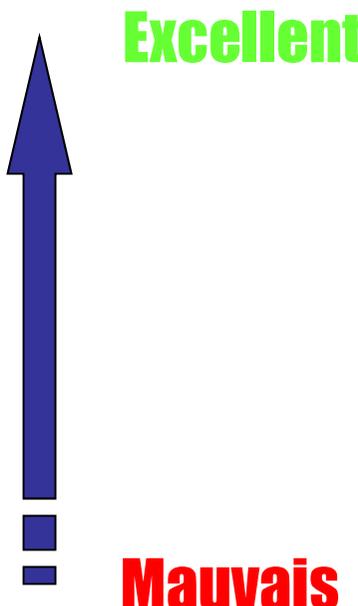




Le couplage

- Le couplage définit le niveau de dépendance entre deux ou plusieurs composants
 - c'est une mesure **qualitative**
- Un faible couplage favorise l'évolution
 - Concentre les modifications sur un composant
 - Pas ou peu d'effet de bord

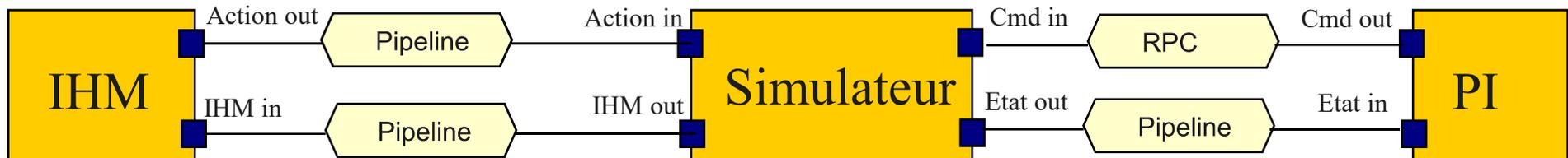
Niveaux de couplage

- On peut distinguer les niveaux de couplage suivants
 - absence de couplage direct
 - couplage de données
 - couplage par référence
 - couplage de contrôle
 - couplage externe
 - couplage commun
 - couplage de contenu
- 
- Excellent**
- Mauvais**

4. Cohésion et couplage

Absence de couplage direct

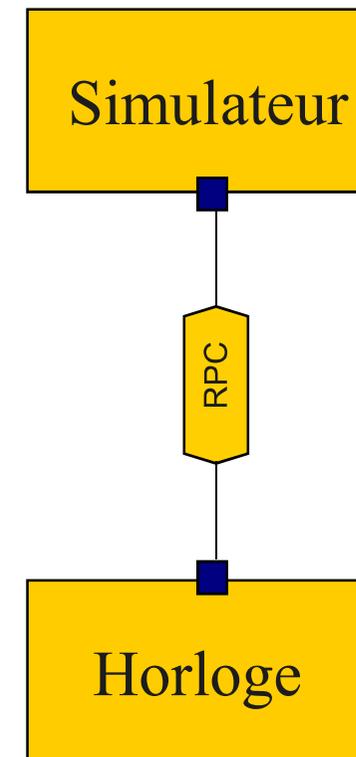
- Composants n'interagissant pas et ne partageant pas d'information
- Exemple
 - Deux modules très éloignés dans la structure architecturale qui ne partagent ni types ni données : IHM et PI
- Évolution
 - Aucun impact



4. Cohésion et couplage

Couplage de données

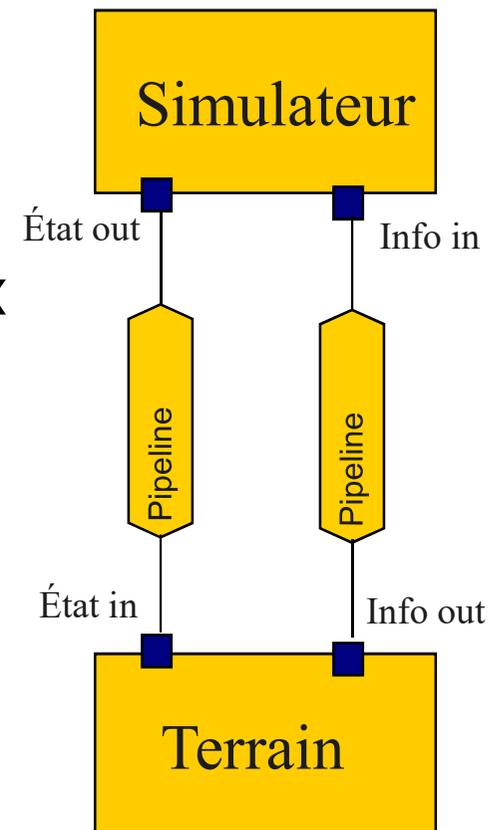
- Deux composants n'échangent que des données simples via leurs interfaces (**passage par valeur – pas de partage de données**)
- Exemple
 - Simulateur et Horloge
- Les composants sont sensibles aux évolutions
 - des **interfaces**



4. Cohésion et couplage

Couplage par référence

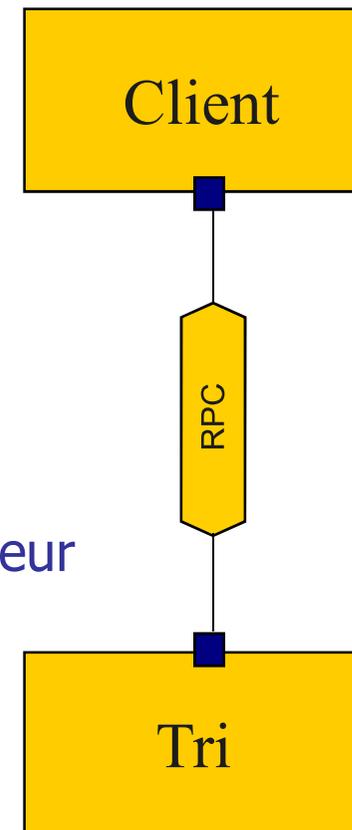
- Deux composants échangent des données structurées via leurs interfaces (éventuellement passage par adresse)
- Exemple
 - Simulateur et Terrain
- Les composants sont sensibles aux évolutions
 - des interfaces et
 - des structures de données



4. Cohésion et couplage

Couplage de contrôle

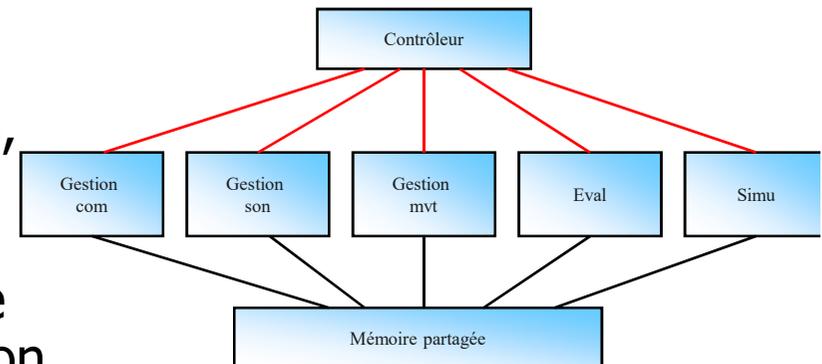
- L'interface d'un composant permet d'influencer son comportement
- Exemple :
 - Un composant de tri à qui on passe en paramètre la méthode de tri
- Le composant client est sensible aux évolutions
 - des interfaces,
 - des structures de données et
 - des fonctions internes du composant serveur

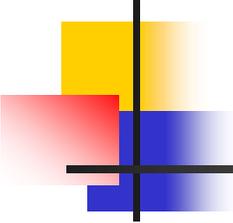


4. Cohésion et couplage

Couplage externe

- Deux composants communiquent par l'intermédiaire d'un tiers
- Exemple :
 - Les composants de simulation
- Évolution
 - Dépendance sur les données, voire sur le comportement. Le canal de communication n'étant pas identifié, il risque d'être oublié lors de l'évolution





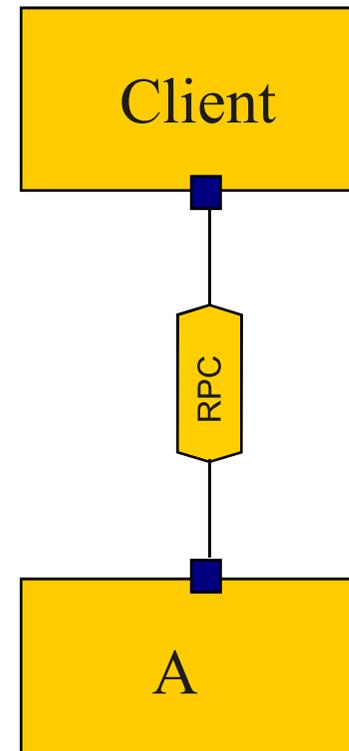
Couplage commun

- Deux composants partagent des variables globales
- Exemples :
 - Communication à base de variables partagées
 - Instruction COMMON de Fortran
- Évolution
 - L'utilisation de variables globales réclame une grande discipline d'utilisation
 - Plus une variable est utilisée, plus il est difficile de faire évoluer sa structure

4. Cohésion et couplage

Couplage de contenu

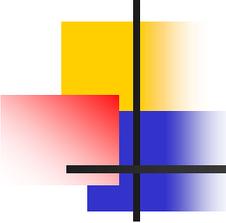
- Un composant connaît et exploite le contenu d'un autre (accès à des variables privées, à des constantes, à la structure logique,...). Ces informations n'étant pas publiées sur les interfaces.
- Exemples
 - Un composant « sait » que le résultat d'un composant de tri est une liste triée
- Évolution
 - Une fois que le contenu d'un module est connu et exploité, il ne peut plus évoluer.



Rappel

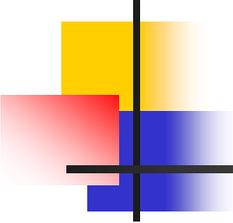
Approche pour la conception

- Dérivation des composants (analyse fonctionnelle)
- Utilisation de patterns
 - patterns d'architecture (ou styles) pour structurer
 - patterns (ou tactiques) de conception pour les propriétés NF
- Évaluation
 - Consistance (syntaxique) de la solution
 - Evaluation de la cohésion et du couplage
 - **Revue d'architecture**



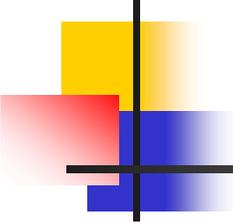
Références bibliographiques

- « Software Architecture in Practice »,
Len Bass, Paul Clemens, Rick Kazman
Addison Wesley Second edition
chapitre 11
- « Evaluating Software Architecture »
Paul Clemens, Rick Kazman, Mark Klein
Addison Wesley



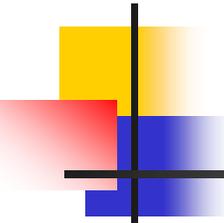
Objectif de ce cours

- Évaluer une architecture
- Méthode d'évaluation ATAM
- Adaptation de ATAM



Évaluer une architecture : pourquoi ?

- Détecter les problèmes au plus tôt
 - exigences fonctionnelles / non fonctionnelles
 - architecture = première étape de conception
 - évaluer pour s'assurer que les choix sont bons

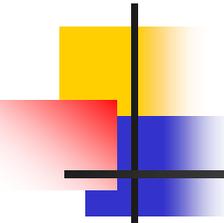


Évaluer une architecture : Quand ?

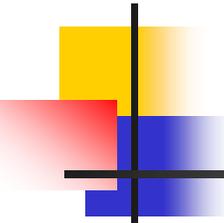
- Conception architecturale
 - s'assurer que les choix sont bons
- Après le codage
 - évaluer l'architecture d'un système existant
 - évaluer l'impact d'une modification

Évaluer une architecture :

Qui ?

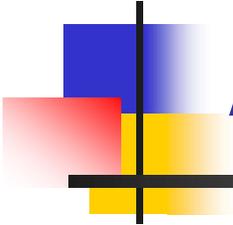


- Équipe d'évaluation
 - conduit l'analyse et l'évaluation
- Acteurs principaux (stakeholders)
 - parmi
 - L'équipe de dev.: codeurs, intégrateurs, testeurs,
 - L'équipe décisionnelle : chef projet, architecte, manager, client, sponsor ...
 - Pour articuler les principales exigences
 - Pour ordonner les priorités
 - c'est bien que ça soit réutilisable mais je ne paie pas pour ça
 - je veux une architecture maintenable (c'est critique pour moi)



Évaluer une architecture : Résultat ? un rapport

- Forme et contenu dépend de la méthode
- Répondre aux questions :
 - l'architecture est-elle adaptée aux besoins ?
 - laquelle des n architectures est la plus adaptée ?
- Deux critères pour l'évaluation par rapport aux besoins
 - le système pourra atteindre les critères de qualité fixés
 - il fonctionnera de façon suffisamment vite par rapport ...
 - il satisfera les exigences de sécurités attendues
 - ...
 - les ressources sont suffisantes
 - personnel, budget, temps, logiciels et matériels à utiliser



Alice aux pays des Merveilles

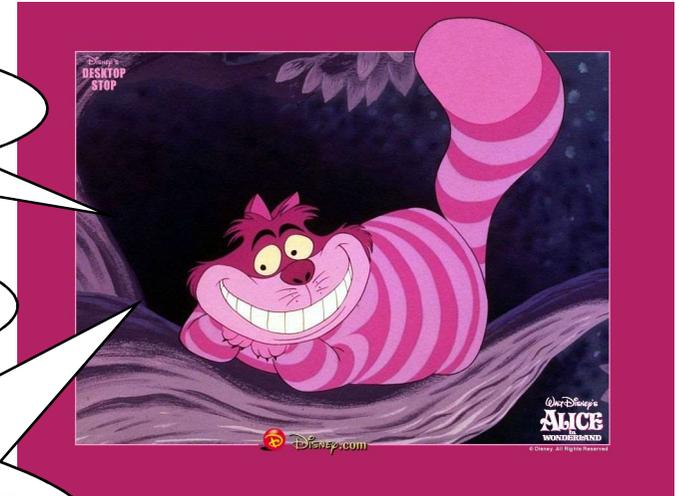
Évaluer une architecture : L'architecture est-elle adaptée ?

Peux-tu m'indiquer
mon chemin ?

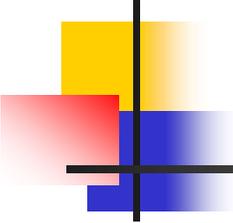
Où veux-tu aller ?

Où, ça m'est égal.

Alors peu importe
le chemin que tu
suivras !

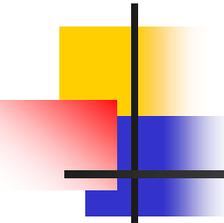


Si on ne peut pas décrire les objectifs qualité,
toute architecture conviendra



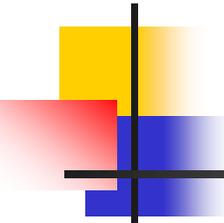
Évaluer une architecture : Quelles qualités ?

- Performance
- Fiabilité
- Disponibilité
- Sécurité
- Modifiabilité
- Portabilité
- Fonctionnalité
- Variabilité
- Décomposabilité



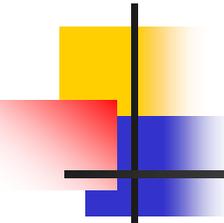
Évaluer une architecture : Affiner les attributs qualités

- Exemple d'exigences : le système doit être
 - Robuste // très modifiable // sécurisé // performant
- Exigences sujettes à interprétation
 - Système sécurisé par des mots de passe, mais quid des virus ?
 - Système robuste ou inadéquat ?
- Les attributs de qualités
 - non quantifiables
 - dépendant du contexte



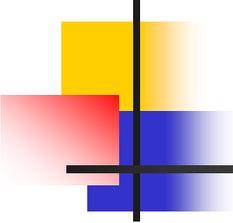
Évaluer une architecture : Affiner les attributs qualités

- Formuler les exigences en fonction du contexte :
- le système doit être
 - modifiable (ou non) par rapport à un certain type de changement
 - sécurisé (ou non) par rapport à un certain type de failles
 - fiable (ou non) par rapport à l'occurrence de certains types de fautes
 - performant par rapport à un ensemble de critères de performance
 - Etc
- Utiliser des scénarios pour exprimer des attributs de qualité



Évaluer une architecture : des méthodes

- ATAM
 - Architectural Tradeoff Analysis Method
- SAAM
 - Software Architecture Analysis Method
- ARID
 - Active Reviews for Intermediate Designs

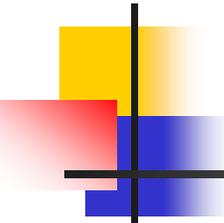


Objectif de ce cours

- Évaluer une architecture
- Méthode d'évaluation ATAM
- Adaptation de ATAM

Méthode ATAM :

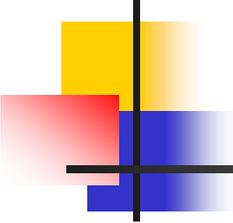
introduction



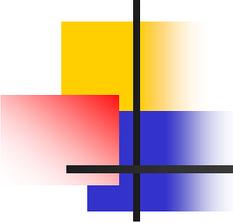
- Architectural Tradeoff Analysis Method
- Evaluation
 - de l'architecture par rapport à la satisfaction de certains objectifs qualité
 - ET des éléments concernant comment les objectifs qualités interagissent entre eux
- tradeoff = compromis

Méthode ATAM :

9 étapes



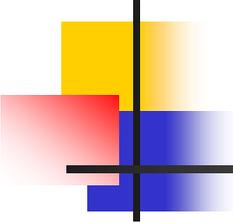
- **Présentation**
 1. Présentation de la méthode
 2. Présentation des objectifs business
 3. Présentation de l'architecture
- **Analyse**
 4. Identification des choix architecturaux
 5. Génération des attributs qualité
 6. Analyse des choix architecturaux
- **Test**
 7. Brainstorming et affectation de priorités pour les scénarios
 8. Analyse des choix architecturaux
- **Rapport**
 9. Présentation des résultats



Méthode ATAM :

9 étapes

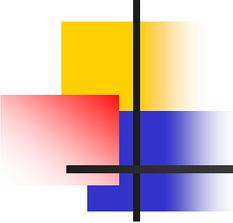
- **Présentation**
 1. Présentation de la méthode
 2. Présentation des objectifs business
 3. Présentation de l'architecture
- **Analyse**
 4. Identification des choix architecturaux
 5. Génération des attributs qualité
 6. Analyse des choix architecturaux
- **Test**
 7. Brainstorming et affectation de priorités pour les scénarios
 8. Analyse des choix architecturaux
- **Rapport**
 9. Présentation des résultats



Méthode ATAM :

3 étapes pour la présentation

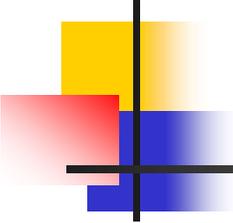
- Rappel : équipe d'évaluation
- But de la phase présentation : échanger les informations
- Présentation de la **méthode**
 - Contexte, objectifs, activités, résultats attendus
 - Fait par le responsable de l'évaluation
- Présentation des **objectifs business**
 - Présentation du contexte
 - Les fonctions principales, les contraintes, objectifs métiers et business, les acteurs
 - Fait par le chef de projet ou client



Méthode ATAM :

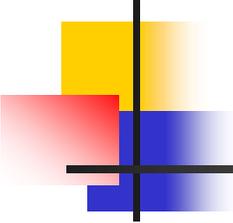
3 étapes pour la présentation

- Présentation (description) de **l'architecture**
 - Contient
 - Les exigences architecturales
 - Les contraintes techniques,
 - Les vues fonctionnelles, physiques, ...
 - Les COTS
 - Les choix architecturaux (styles, tactiques)
 - Les scénarios importants (usage, évolution)
 - Les risques, le glossaire
 - Basée sur les vues
 - Fait par l'architecte



Méthode ATAM : 9 étapes

- Présentation
 1. Présentation de la méthode
 2. Présentation des objectifs business
 3. Présentation de l'architecture
- **Analyse**
 4. Identification des choix architecturaux
 5. Génération des attributs qualité
 6. Analyse des choix architecturaux
- **Test**
 7. Brainstorming et affectation de priorités pour les scénarios
 8. Analyse des choix architecturaux
- **Rapport**
 9. Présentation des résultats



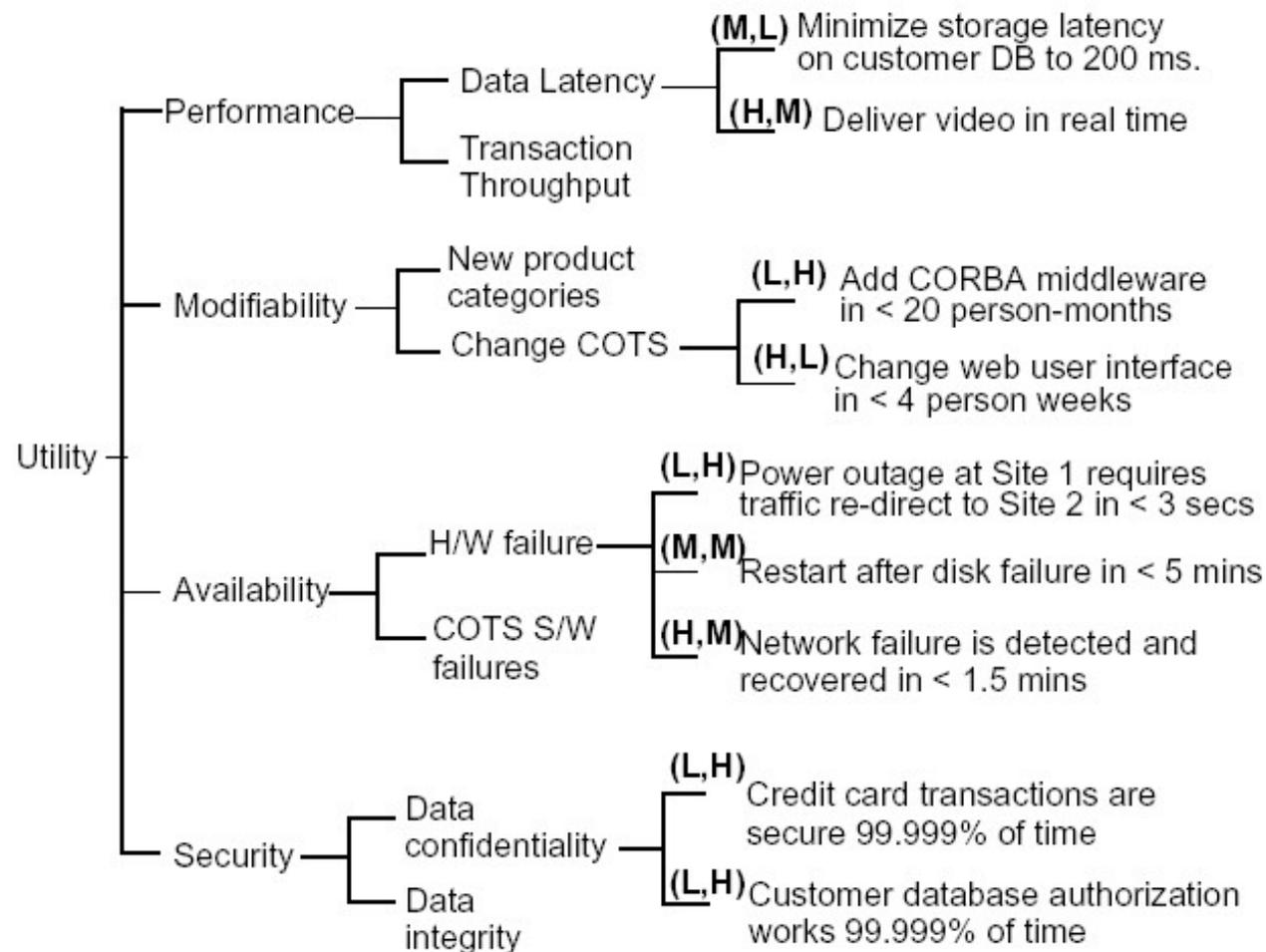
Méthode ATAM :

3 étapes pour l'analyse

- Identification des choix d'architecture (étape 4)
 - Styles et tactiques (approches)
 - Représentent les priorités dans les choix d'architecture
 - Composants, topologie, interactions, ...
 - Bénéfices et points faibles
- Génération des attributs de qualité (étape 5)
 - Identifier, prioriser, raffiner les attributs
 - Arbre des attributs de qualité
 - Scénarios

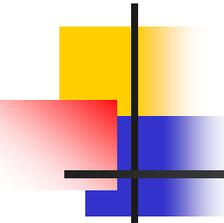
Méthode ATAM : (étape 5)

Arbre des attributs (exemple)

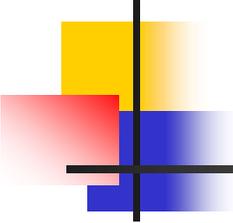


Méthode ATAM : (étape 5)

Scénarios



- Sont utilisés pour
 - Pour représenter les intérêts des acteurs
 - Comprendre les exigences relatives aux attributs de qualité
- Doivent
 - Anticiper les usages (use cases),
 - Anticiper les changements (scénarios d'élargissement), ou
 - Anticiper la charge du système (scénarios exploratoires)
- Doivent faire apparaître
 - quel est le stimulus qui déclenche le scénario
 - quelles sont les réponses



Méthode ATAM : (étape 5)

exemples de scenario

- Scénario d'usage
 - Remote user requests a database report via the Web during peak period and receives it within 5 seconds.
- Scénario "d'élargissement"
 - Add a new data server to reduce latency in scenario 2.5 to 1 seconds.
- Scénario exploratoire
 - Half of the servers go down during normal operation without affecting overall system availability.

=> Scénarios aussi spécifiques que possible.

Méthode ATAM : 6. Analyse des choix architecturaux

■ Scénarios Vs. Architecture

Scenario: S12 (Detect and recover from HW failure of main switch.)

Attribute: Availability

Environment: normal operations

Stimulus: CPU failure

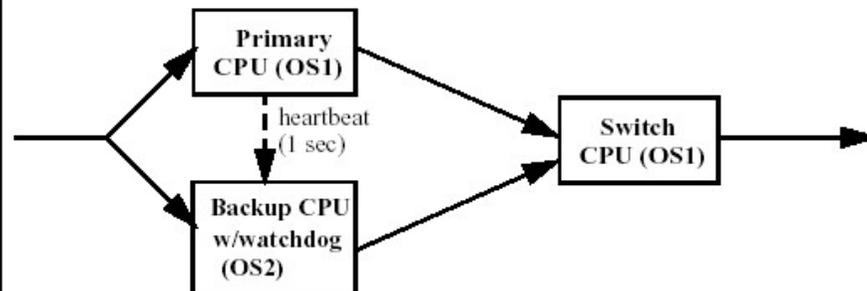
Response: 0.999999 availability of switch

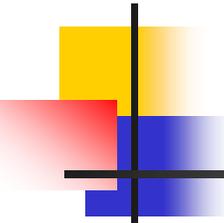
Architectural decisions	Risk	Sensitivity	Tradeoff
Backup CPU(s)	R8	S2	
No backup Data Channel	R9	S3	T3
Watchdog		S4	
Heartbeat		S5	
Failover routing		S6	

Reasoning:

- ensures no common mode failure by using different hardware and operating system (see Risk 8)
- worst-case rollover is accomplished in 4 seconds as computing state takes ...
- guaranteed to detect failure with 2 seconds based on rates of heartbeat and watchdog ...
- watchdog is simple and proven reliable
- availability requirement might be at risk due to lack of backup data channel ... (see Risk 9)

Architecture diagram:



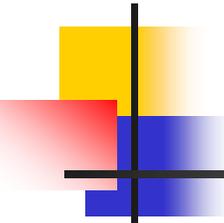


Méthode ATAM :

6. Analyse des choix architecturaux

L'équipe d'évaluation

- analyse les choix du point de vue des attributs qualité et évalue les risques
- Identifie les approches qui maximisent les attributs qualités
- Identifie les questions relatives aux attributs qualité de haute priorité
- Pose les questions relatifs à chaque attribut qualité
- Identifie des risques et non risques, des points sensibles et des compromis (tradeoffs)



Méthode ATAM :

6. Analyse des choix architecturaux

- Exemples
 - Performance
 - Comment les priorités sont assignées aux processus ?
 - Quelle est la fréquence d'arrivée des messages ?
 - Quelle est la durée de traitement d'une transaction ?
 - Modifiabilité
 - Quels sont les composants qui nécessitent une connaissance détaillée des formats des messages ?
 - Quels composants sont connectés de façon asynchrone ?

ATAM : étape 6

Points sensibles et compromis

- Point sensible = propriété d'un composant critique vis-à-vis du succès
 - Si le nombre de clients connectés à la la BD affecte le nombre de transactions par qui pourront être traitées par seconde alors c'est un point sensible pour la performance.
 - Maintenir un backup de la BD impacte la fiabilité
 - La nombre de bit d'une clé de chiffrement impacte la qualité du chiffrement.
- Compromis : propriété qui affecte plus d'un attributs ou point sensible
 - Pour augmenter la performance, on peut choisir de coder en assembleur. Ce faisant, on diminue la portabilité de l'application.
 - Maintenir un backup de la BD affecte la performance du système. Choisir la fréquence avec laquelle on fait ce Backup est un compromise entre fiabilité et performance

ATAM : étape 6

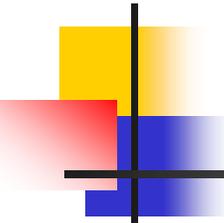
Risques & Non-Risques

Choisir de maintenir un backup a une trop grande fréquence

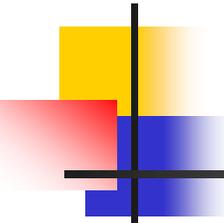
- est un **risque** pour la performance si le coût est excessif
- est un **non-risque** si le coût relatif à la performance est négligeable

Méthode ATAM :

9 étapes



- Présentation
 1. Présentation de la méthode
 2. Présentation des objectifs business
 3. Présentation de l'architecture
- Analyse
 4. Identification des choix architecturaux
 5. Génération des attributs qualité
 6. Analyse des choix architecturaux
- **Test**
 7. Brainstorming et affectation de priorités pour les scénarios
 8. Analyse des choix architecturaux
- **Rapport**
 9. Présentation des résultats

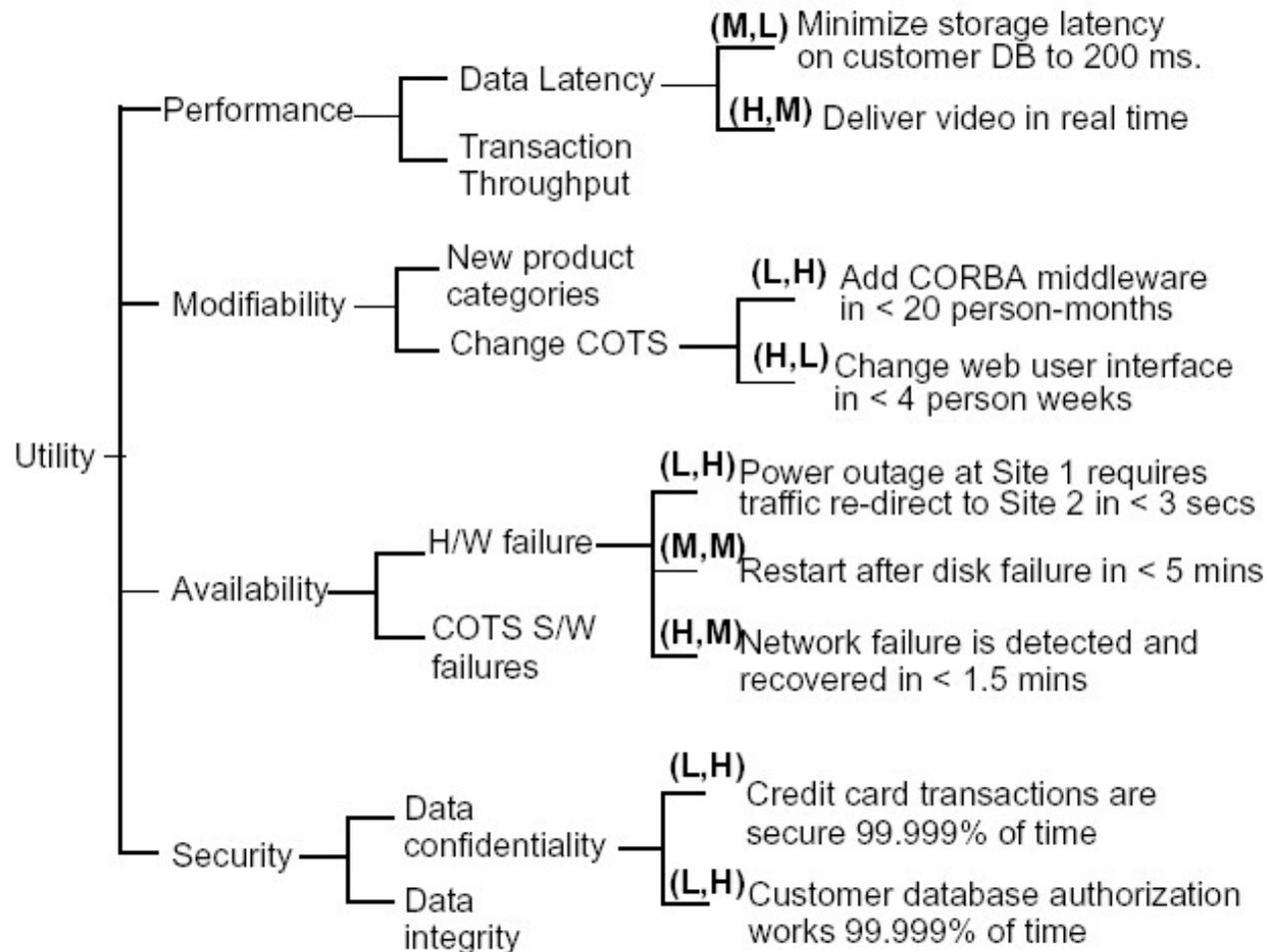


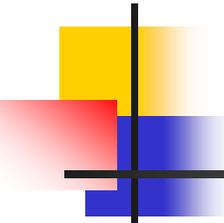
Méthode ATAM : étape 7

Brainstorm & Prioritisation des Scenarios

- Génération des scénarios par brainstorming
 - Les scénarios aux feuilles des arbres servent d'exemple
 - Les nouveaux scénarios sont ajoutés

Méthode ATAM : Arbre des attributs (exemple)





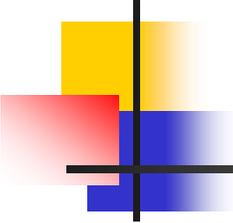
Méthode ATAM :

8. Analyse des choix architecturaux

- Identifier les choix architecturaux impactés par les nouveaux scénarios
- Reprendre la phase 6
- Évaluer les risques et non-risques.
- Poursuivre le travail d'annotation de l'architecture

Méthode ATAM :

9 étapes



- Présentation
 1. Présentation de la méthode
 2. Présentation des objectifs business
 3. Présentation de l'architecture
- Analyse
 4. Identification des choix architecturaux
 5. Génération des attributs qualité
 6. Analyse des choix architecturaux
- Test
 7. Brainstorming et affectation de priorités pour les scénarios
 8. Analyse des choix architecturaux
- **Rapport**
 9. Présentation des résultats

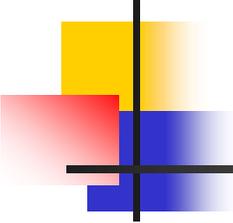
Méthode ATM

9. Présentation des résultats

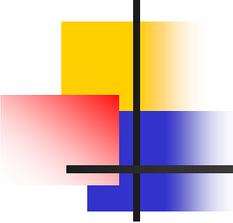
- Les choix architecturaux
- L'arbre des attributs
- Scénarios
- Risques et "non-risques"
- Points sensibles et compromis (tradeoffs)

Méthode ATAM :

9 étapes



- **Présentation**
 1. Présentation de la méthode
 2. Présentation des objectifs business
 3. Présentation de l'architecture
- **Analyse**
 4. Identification des choix architecturaux
 5. Génération des attributs qualité
 6. Analyse des choix architecturaux
- **Test**
 7. Brainstorming et affectation de priorités pour les scénarios
 8. Analyse des choix architecturaux
- **Rapport**
 9. Présentation des résultats



Evaluation d'une architecture

- Présentation
 1. Présentation de la méthode
 2. Présentation des objectifs business
 - 3. Présentation de l'architecture**
- Analyse
 - 4. Identification des choix architecturaux**
 5. Génération des attributs qualité
 - 6. Analyse des choix architecturaux**
- Test
 7. Brainstorming et affectation de priorités pour les scénarios
 8. Analyse des choix architecturaux
- Rapport
 - 9. Présentation des résultats**