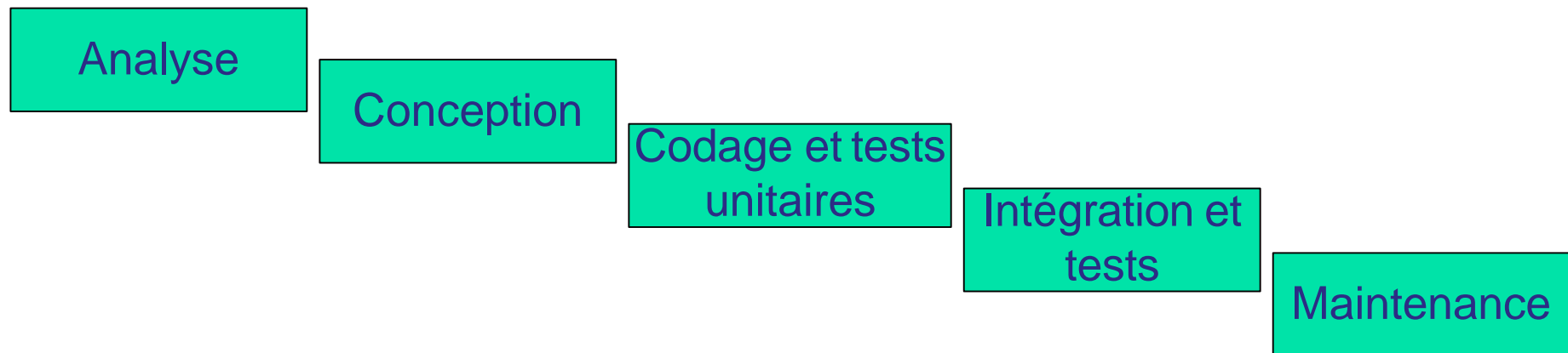


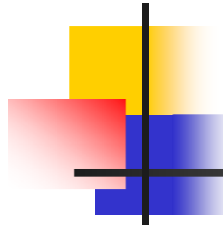
Architecture Logicielle



Architecture logicielle

- Analyse : récolte des exigences
- Cycle de vie





Plan

- Notion d'architecture
- Intérêt de l'architecture
- Représentation
- Conclusion

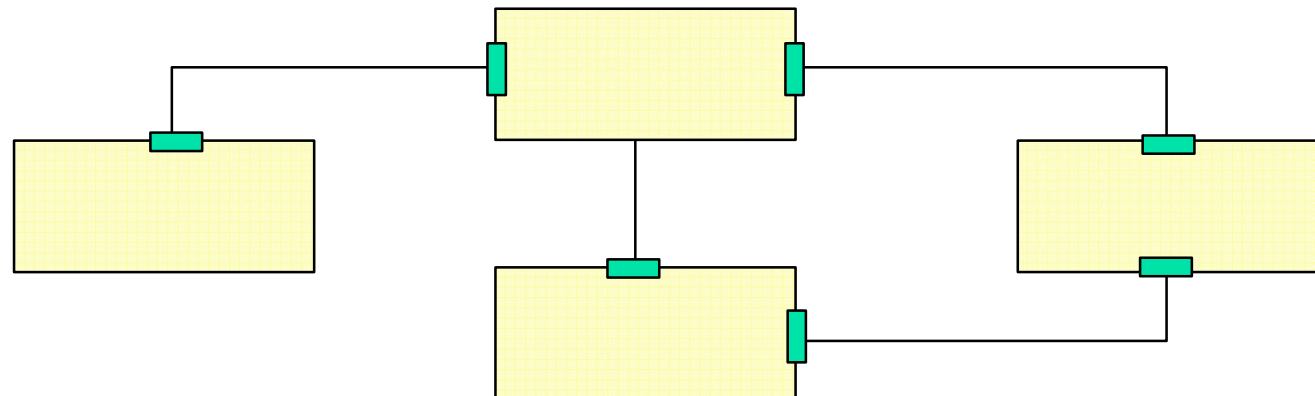


Importance de l'architecture

- Aujourd'hui reconnu comme une étape fondamentale du développement logiciel
 - elle apparaît dans tous les projets sérieux
 - les entreprises y attachent une grande importance
 - création de la fonction d'architecte
 - création d'équipes d'architectes (pour capitaliser et pérenniser les connaissances architecturales métier)
- Bill Gates est « Chief Software Architect » !

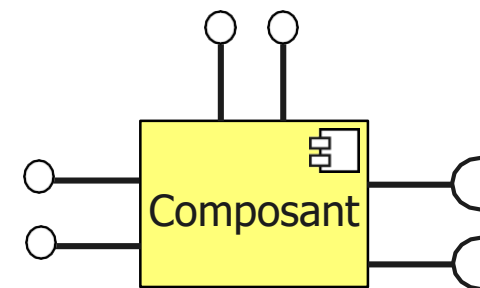
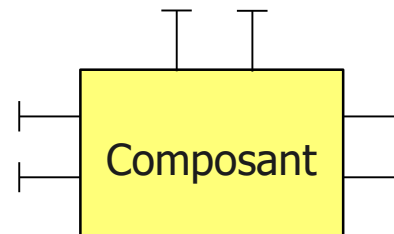
Définition de l'architecture

- Une architecture logicielle est une **représentation abstraite** d'un système exprimée essentiellement à l'aide de **composants logiciels** en interaction via des **connecteurs**



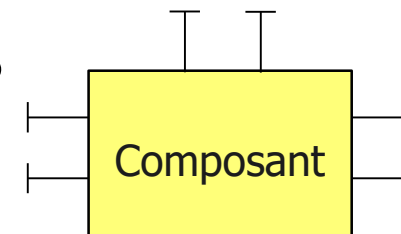
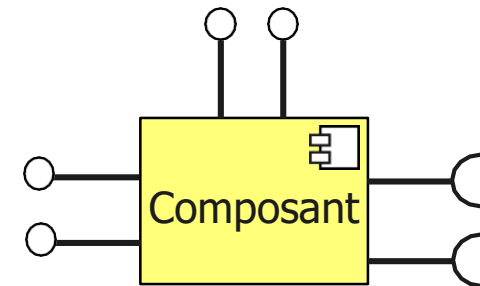
Composants logiciels

- Les composants sont des spécifications d'unités fonctionnelles
 - clairement définies
 - sémantiquement cohérentes et compréhensibles
- Développés ou acquis
- Ne pas confondre
 - spécification
 - réalisation



Description des composants

- Propriétés fonctionnelles
 - Services requis
 - Services fournis
 - Cycle de vie
- Contraintes
 - Type de communication
 - Ordonnancement
- Propriétés non fonctionnelles
 - Performance, robustesse, ...



Connecteurs

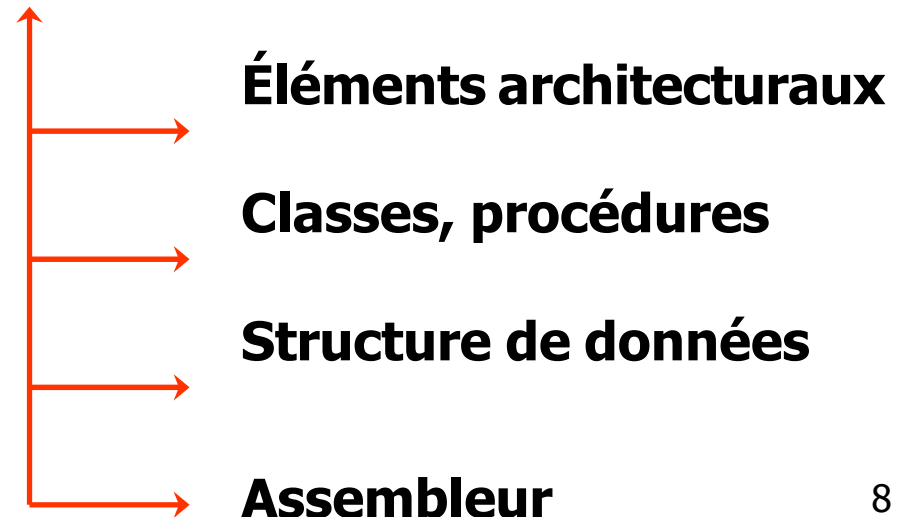
- ✓ Ce sont des objets du premier ordre
- ✓ Assurent les interactions entre composants
- ✓ Peuvent être de complexité variable
 - ✓ du simple appel de méthode à l'ordonnanceur
- ✓ Permettent la flexibilité et l'évolution
- ✓ Pas de langage de spécification de connecteurs



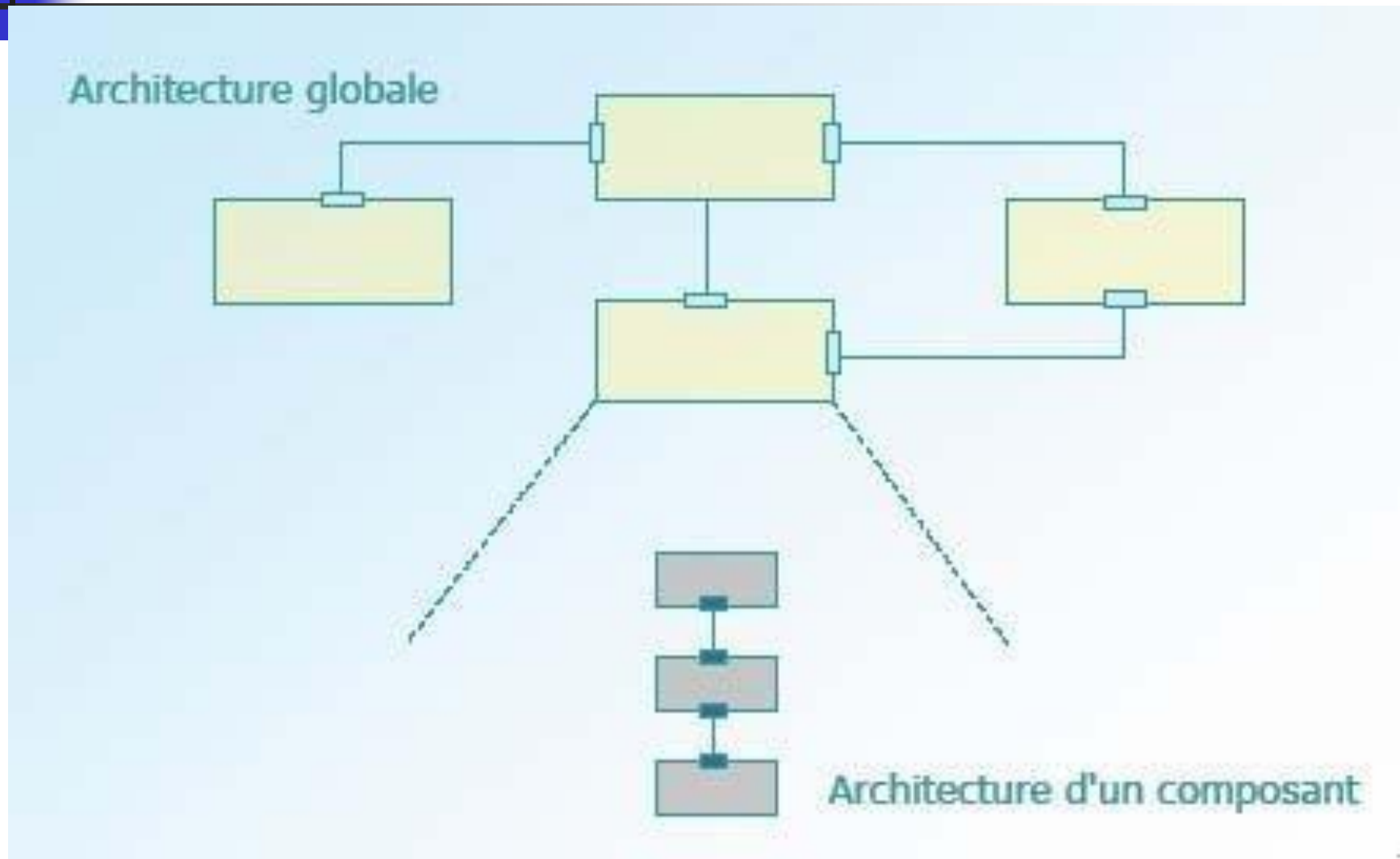
L'architecture est une abstraction supplémentaire

- Ne fournit que les propriétés externes des éléments structurants
- Ne se préoccupe pas des détails d'implantation

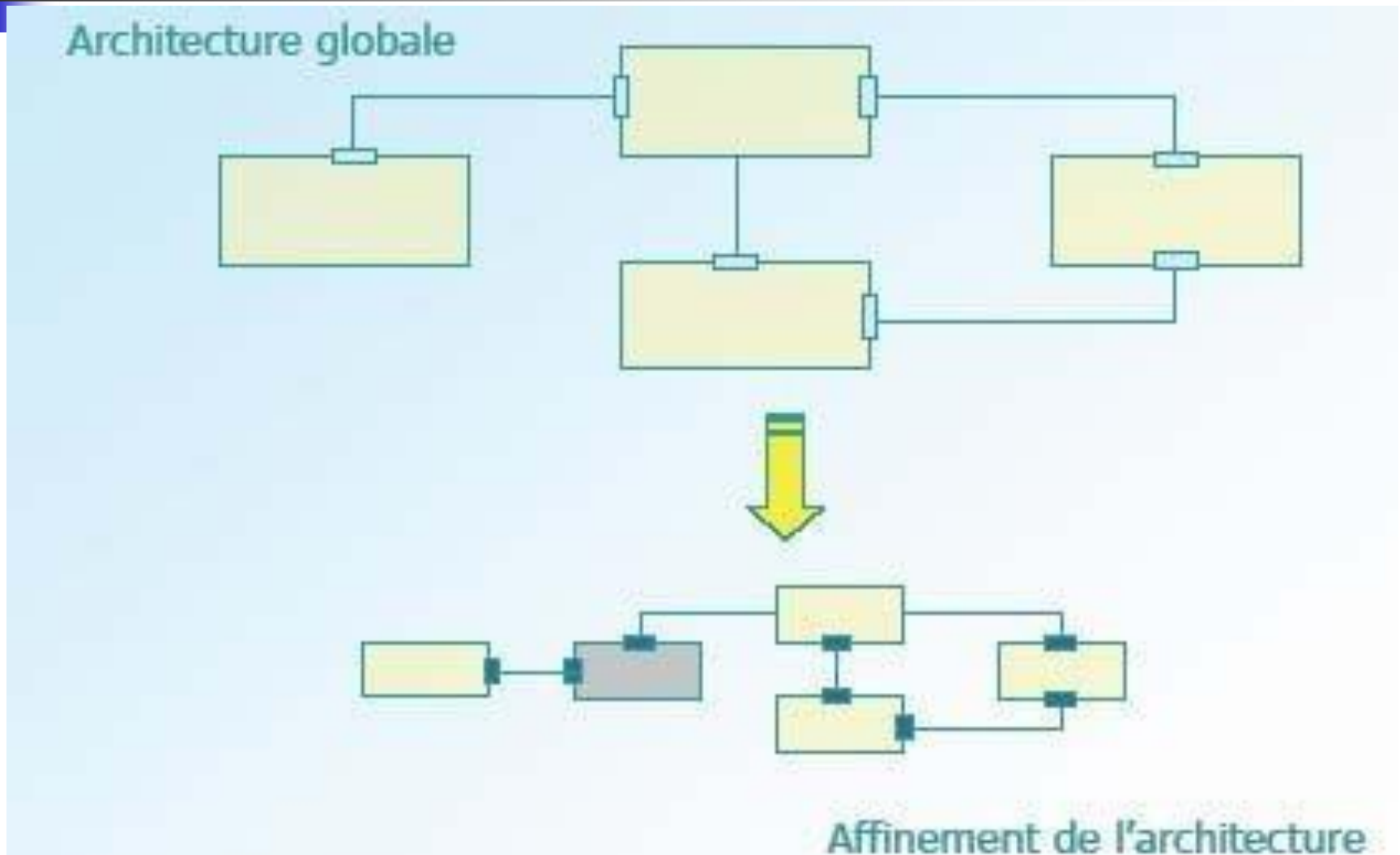
Abstraction



Une succession d'abstractions

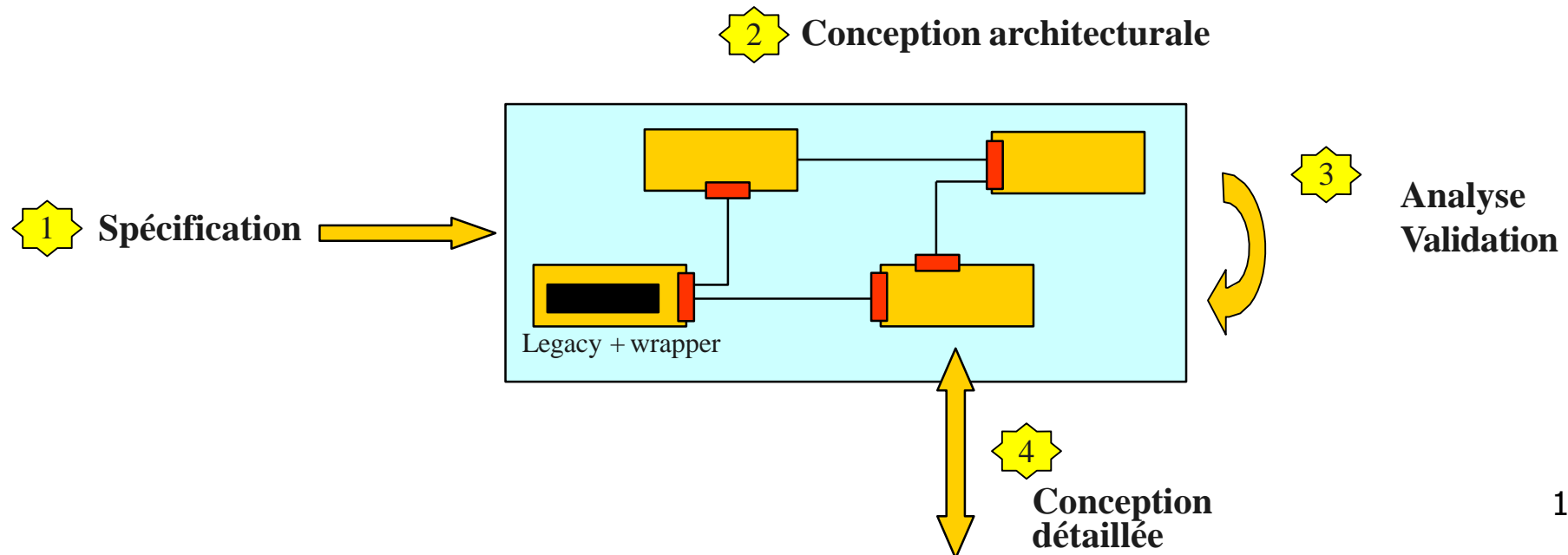


Une succession d'abstractions

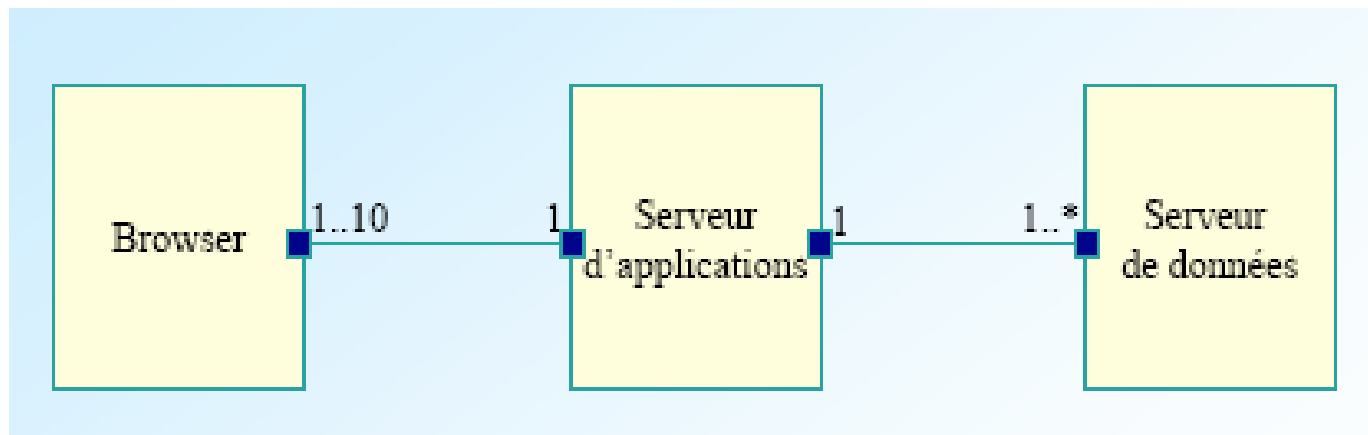


Positionnement

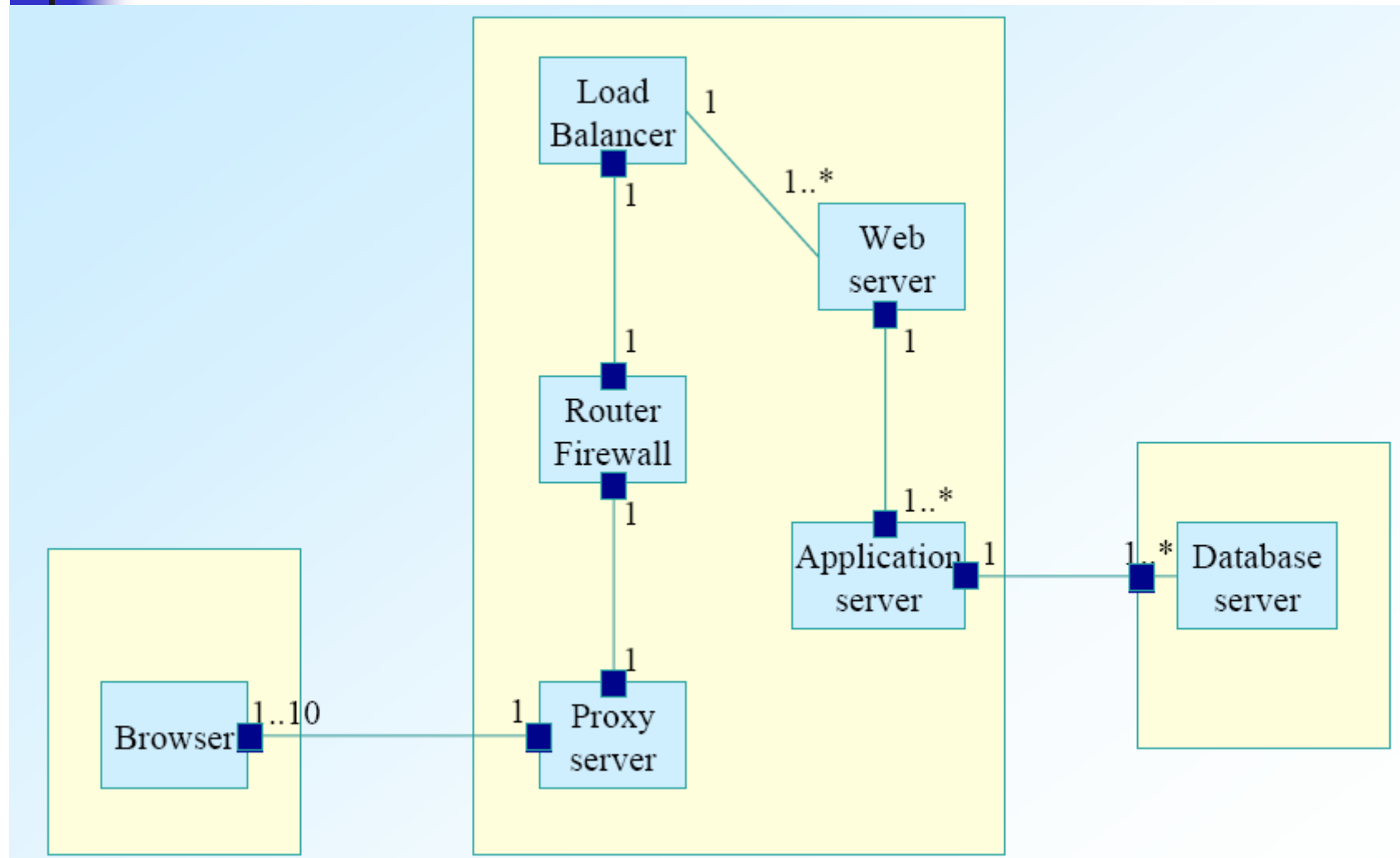
- L'architecture = première étape de conception
 - réduire la complexité du système abordé en le structurant en composants logiciels



Exemple : application e-commerce

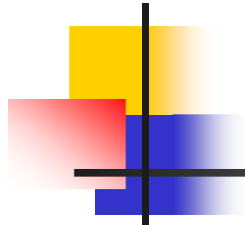


Exemple : application e-commerce



Exemple : application e-commerce





Plan

- Notion d'architecture
- **Intérêt de l'architecture**
- Représentation
- Conclusion

Enfin des réponses...



- Première étape de conception
- Permet de réfléchir et répondre aux questions:

Où développer ?

Comment développer ?

Quelles équipes, quelles techno ?

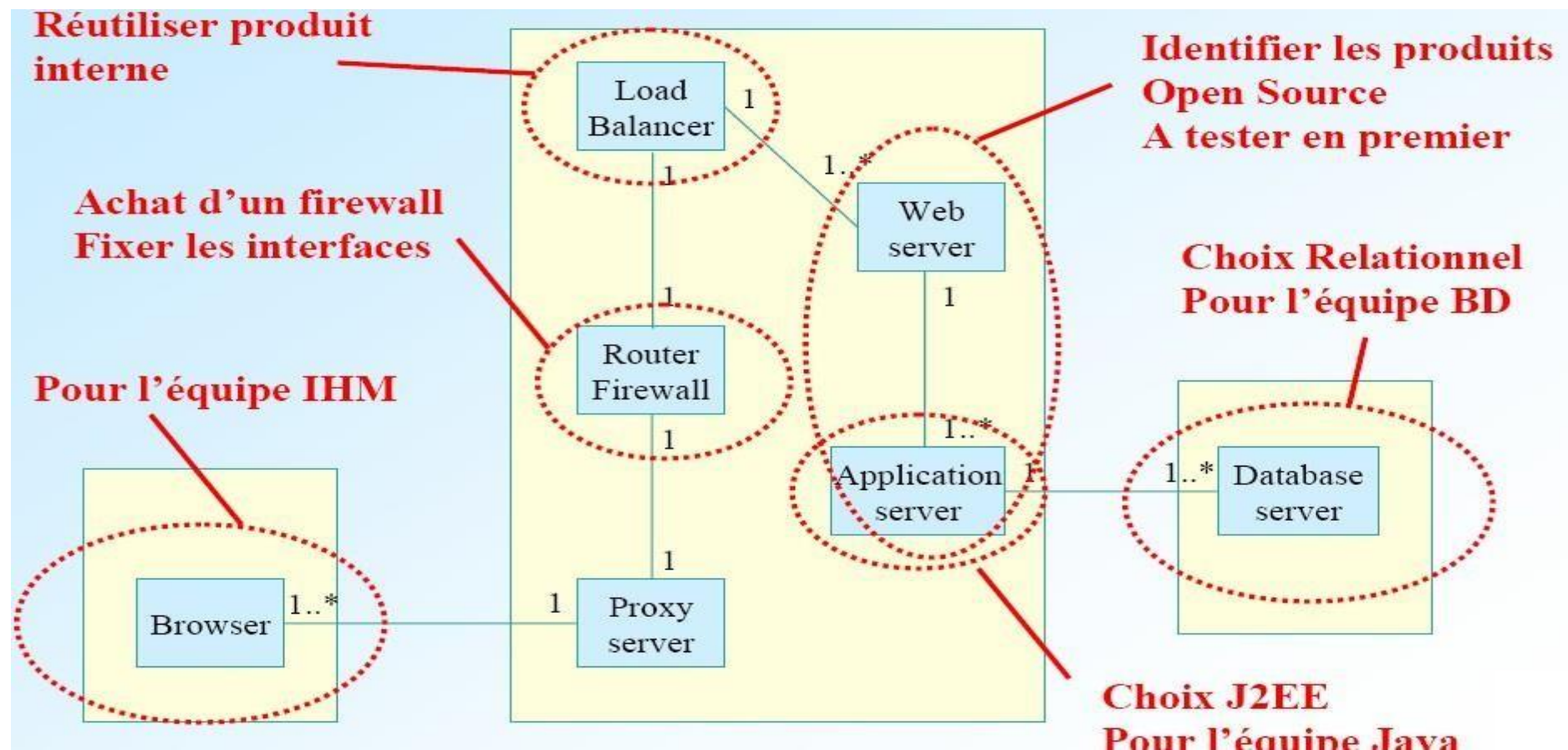
Quel coût ?



A partir de l'architecture, on peut

- Définir un plan de travail
- Répartir le travail entre les équipes
- Allouer les ressources
- Imposer des contraintes techniques
- Structurer les différentes étapes
 - le développement
 - Les tests
 - La documentation
 - La maintenance

Exemple de structuration

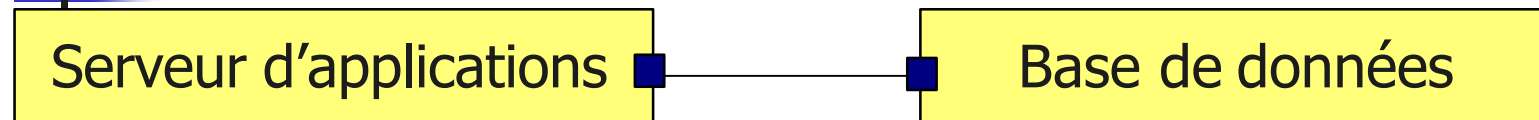




Impact sur la qualité logicielle

- ❑ L'architecture a une forte **influence** sur les propriétés finales d'un système
- ❑ La structuration architecturale **favorise ou pénalise** les propriétés non fonctionnelles telles que
 - ❑ Performance
 - ❑ Sécurité
 - ❑ Sûreté
 - ❑ Disponibilité
 - ❑ Maintenabilité
 - ❑ etc.
- ❑ Un premier niveau de **compromis** se fait au niveau architectural. Et ce, de façon quasi définitive ...

Exemple

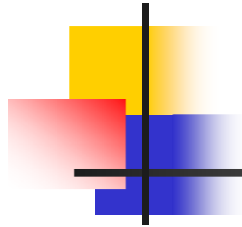


Sur la même machine

- + sécurité
- + performance (à voir)

Sur deux machines

- + disponibilité (caches, ...)
- + maintenabilité
- + sûreté (réplication possible)



Des éléments à prendre en compte

- ❑ Peu de composants favorisera la performance (moins de communication)
- ❑ Beaucoup de composants favorisera la maintenance (au détriment de la performance)
- ❑ La redondance peut favoriser la sûreté, mais pas la compacité ou la sécurité

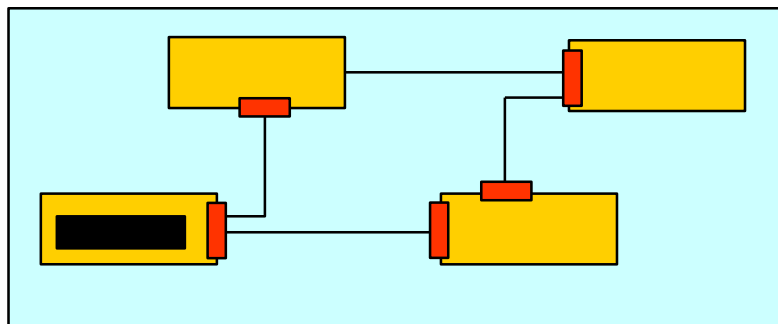
Architecture : première étape de validation



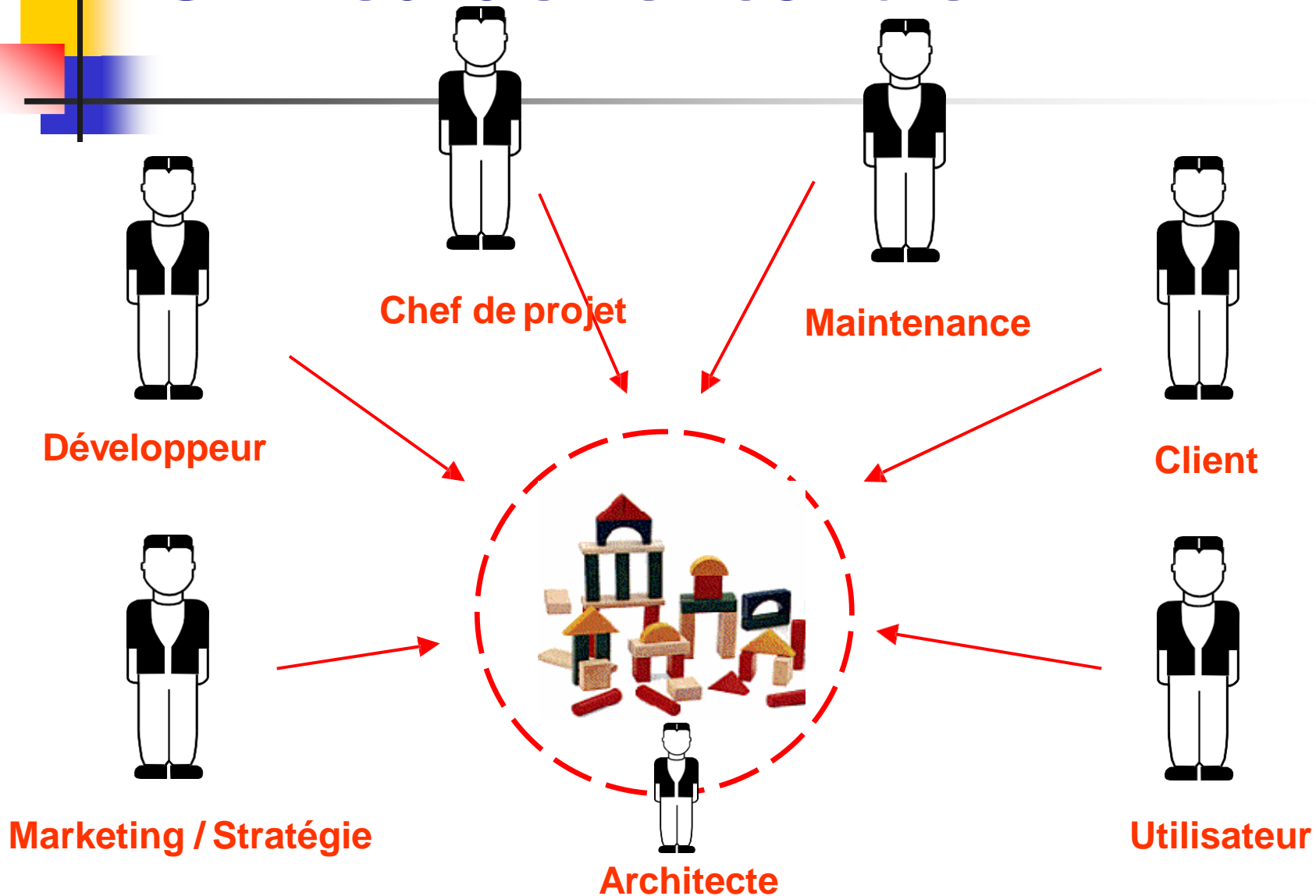
- Les décisions architecturales ont un impact important et durable
- À valider soigneusement et au plus tôt
 - revues d'évaluation
 - développement de prototypes
 - évaluation de technologies clef
 - utilisation de techniques formelles
- L'architecture influence les qualités mais ne les garantie pas

Vecteur de communication

- L'architecture fournit un canevas permettant à tous d'exprimer ses intérêts et de négocier
 - réunion de tous les intervenants autour de l'architecture
 - négociation des exigences avec les utilisateurs
 - négociation des évolutions à apporter
 - présentation régulière aux clients et au management des avancées (fonctions / coûts / échéances)
 - structuration des équipes et allocations des ressources



Un lieu de rencontre

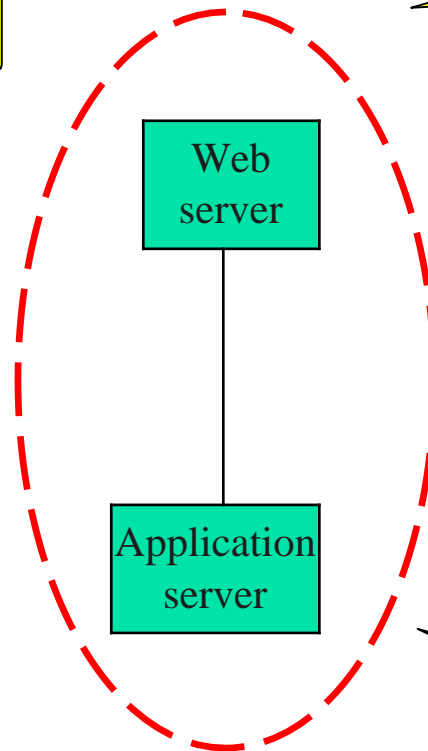


Exemple

Architecte : nécessaire vu la complexité des applications

Utilisateur : est-ce que les interfaces ne vont pas changer tout le temps ?

Client : non, trop cher !



Programmeur : super cool !

Maintenance : non, trop instable !

Manager : cool, cela va être cher !



Synthèse

- Une architecture est représentée par plusieurs vues complémentaires
 - Séparation des préoccupations
 - Difficile d'assurer la cohérence et la complétude
- Vues importantes
 - Vue logique
 - Vue dynamique
 - Vues d'allocation (ou de déploiement)
- Pas de formalisme standard
 - Utiliser un formalisme connu quand c'est possible (UML par exemple)
 - ajouter les informations que l'on juge importantes de la meilleure manière possible (notes ou notation personnelle)