

`marp: true`

`paginate: true`

`theme: default`



CM1 — Fondamentaux de l'architecture logicielle

Partie 1 — Pourquoi l'architecture ?

BUT Informatique — R4.01



Objectifs du CM

- Comprendre le rôle de l'architecture logicielle
- Identifier les **enjeux actuels** (complexité, durée de vie, IA)
- Poser les bases conceptuelles avant les principes techniques

? Pourquoi parler d'architecture logicielle ?

Un logiciel vit longtemps :

- il évolue
- il est maintenu par plusieurs personnes
- ses contraintes changent

👉 Sans architecture claire :

- le code devient difficile à comprendre
- chaque évolution coûte plus cher
- la qualité se dégrade avec le temps



Définition (simple)

Architecture logicielle =

L'ensemble des décisions structurelles qui organisent un système logiciel.

Ces décisions :

- impactent tout le système
- sont difficiles à changer
- imposent des contraintes aux développeurs



Exemples de décisions architecturales

- Découpage en modules / couches
- Sens des dépendances
- Séparation métier / technique
- Choix d'un style architectural (MVC, hexagonale, etc.)

! Ce ne sont *pas* des choix d'implémentation locale.



Développement et IA

Aujourd’hui :

- l’IA génère du code rapidement
- le code fonctionne souvent... au début

Mais :

- l’IA **n’architecte pas**
- elle applique ce qu’on lui demande

👉 Plus le système est gros, plus l’architecture est cruciale.

⚠ Le piège du “vibe coding”

Coder à l'intuition :

- fonctionne pour des petits projets
- devient dangereux à grande échelle

Problèmes :

- code peu cohérent
- responsabilités floues
- couplage excessif

👉 L'architecture sert à canaliser l'intuition.



Ce qui fait la valeur d'un développeur

- Écrire du code : accessible
- Comprendre un système : rare
- Le faire évoluer sans le casser : précieux



L'architecture est une compétence clé.



Transition

Avant de parler d'architectures concrètes,
nous devons comprendre les **principes fondamentaux**
qui s'appliquent à **tout logiciel bien conçu**.



Partie 2 : les principes fondamentaux.