



Architectures logicielles



Qu'est-ce qu'une architecture logicielle ?

- L'architecture logicielle d'un système informatique est une **représentation du système** qui facilite la compréhension de son fonctionnement.
- Quand une application devient complexe, un moyen de gérer cette complexité est de concevoir l'application **selon ses responsabilités ou préoccupations**.



Qu'est-ce qu'une architecture n-tiers ?

- Les applications n-tiers sont des applications qui sont séparés en plusieurs niveaux. Aussi appelé «applications distribuées» et «applications multiniveaux» séparé en niveaux qui peuvent être distribués entre le client et le serveur.
- Dans un contexte distribué
 - Les tiers peuvent être exécutés sur des machines différentes
 - Certains tiers peuvent être sous-découpés
 - De nombreuses variantes de placement des tiers et de leur distribution

DIFFÉRENCE ENTRE COUCHE LOGIQUE ET TIERS PHYSIQUE

Le tier est une unité physique, où le code s'exécute.

Exemple : client, serveur d'applications, serveur de base de données ;

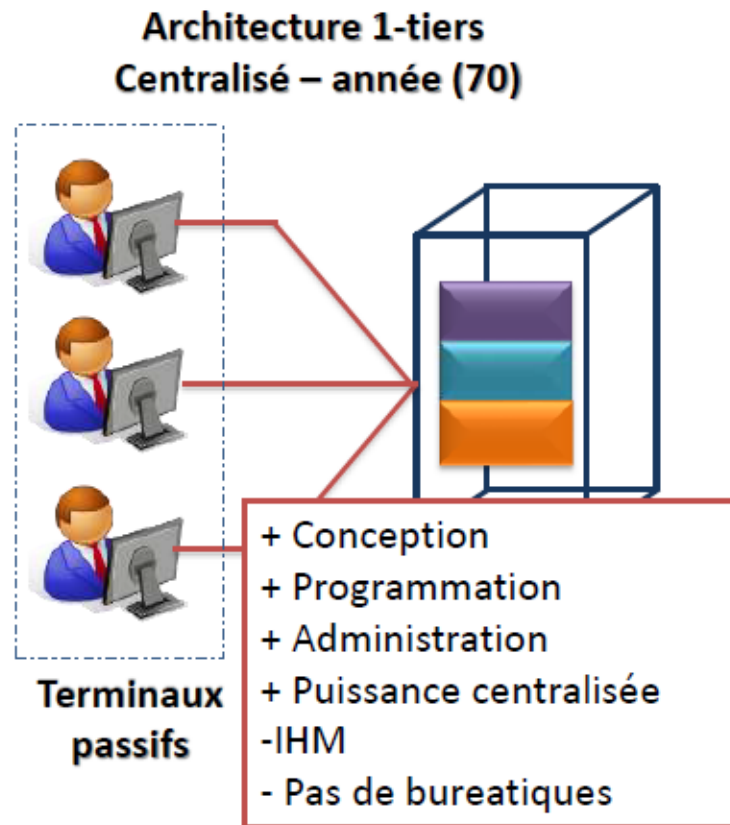
La couche est une unité logique, comment organiser le code qui peut ou non s'exécuter dans le même processus ou sur la même machine.

Les tiers sont des endroits où les couches sont déployées et exécutées.

Chaque couche peut s'exécuter dans un tier individuel. Cependant, plusieurs couches peuvent également être en mesure de s'exécuter dans un seul tier.

1-TIERS

Dans une approche d'application de type 1-tiers, les **trois couches**(présentation, traitement, données) sont fortement et intimement liées, et **s'exécutent sur la même machine**

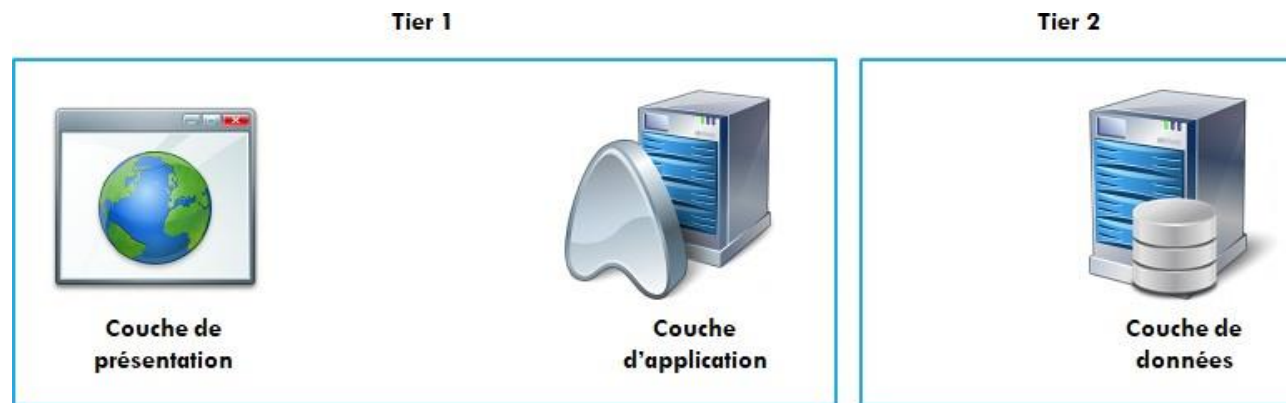


1-TIERS

- **Administration du système:** Facile
- **Sécurité :** Faible
- **Performance:** Faible
- **Extensibilité:** Faible
- **Réutilisation:** Faible
- **Facilité de développement:** Elevée
- **Flexibilité d'architecture matériel:** Faible
- **Relève en cas de pannes:** Faible
- **Coûts:** Facile

2-TIERS

Dans une approche d'application de type 2-tiers, le premier tier contient la couche de présentation et la couche d'application et le deuxième tier contient la couche de données.

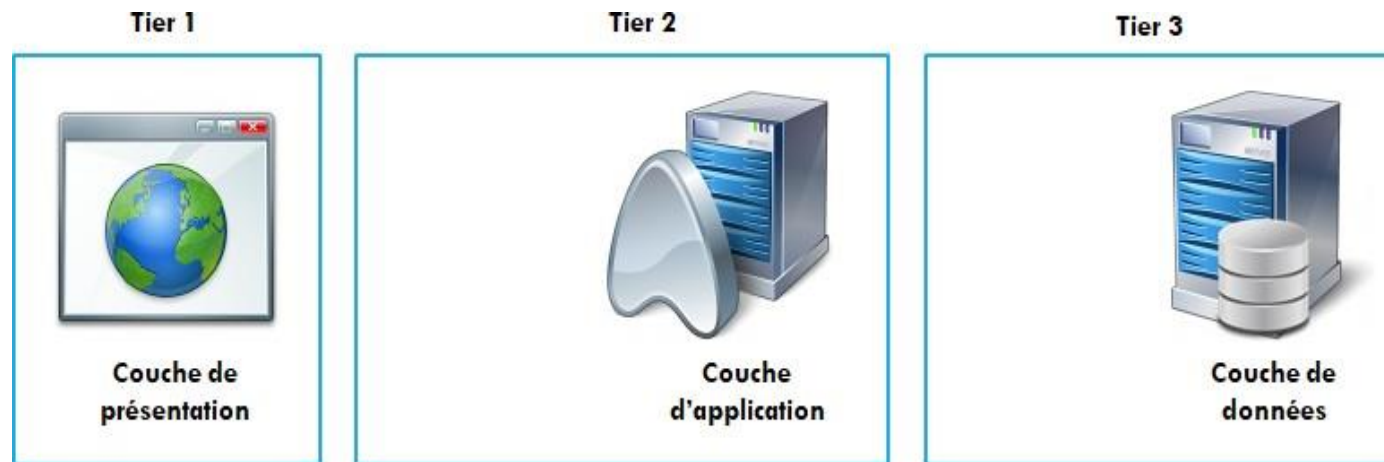


2-TIERS

- C'est le type d'architecture client-serveur le plus basique :
- La couche présentation est située sur le client
- La couche donnée est située sur le serveur (par ex. une base de données relationnelles SQL, Oracle,...)
- La couche Application peut être située sur le client (au sein même de l'IHM), ou sur le serveur (par ex. des procédures stockées sur la base de données), ou partagée entre les deux rôles.
- Il y a une relation directe entre les clients et les serveurs de données.
- (+) simple à mettre en œuvre
- (-) peu flexible et supporte difficilement la montée en charge

3-TIERS

c'est le cas le plus simple de l'architecture N-Tier. Les **trois couches** représentent **trois tiers** différents.



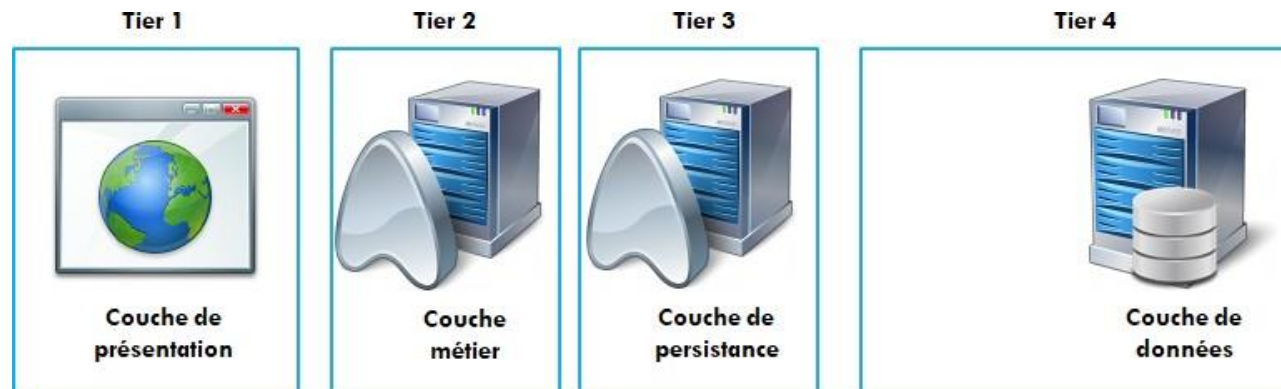
3-TIERS

- L'architecture 3-tiers est une **extension du modèle client-serveur** qui introduit un rôle spécifique pour la couche de traitements **métiers**.
- Il y a donc décomposition d'un même service sur 2 types de serveurs:
- Un type de **serveur** dédié à la gestion des **traitements métiers**.
- Un type de **serveur** dédié à la gestion des **données** persistantes.
- (+) plus évolutif que le 2-tiers
- (+) **meilleure répartition des charges** de travail coté serveur
- (+) économiquement moins cher, surtout lors de la **montée en charge**
- (-) administration et mise en œuvre **plus compliquée** que le 2-tiers

N-TIERS

- **Architecture à 3 tiers ou plus.** Certaines couches de l'architecture à 3 tiers peuvent être divisées en plusieurs couches, par exemple : **couche application = couche métier + couche de persistance.**

Ceci est idéalement implémenté pour une application Web.





Architecture propre (architecture en oignon)

- L'architecture propre met la **logique métier** et le **modèle d'application** au **centre** même de l'application.
- Cette architecture est souvent **représentée sous la forme d'une série de cercles concentriques**, à l'image des couches d'un oignon.
- Au lieu que la logique métier dépende des préoccupations de l'accès aux données ou d'une autre infrastructure, cette dépendance est inversée : les détails de l'infrastructure et de l'implémentation dépendent du noyau de l'application.

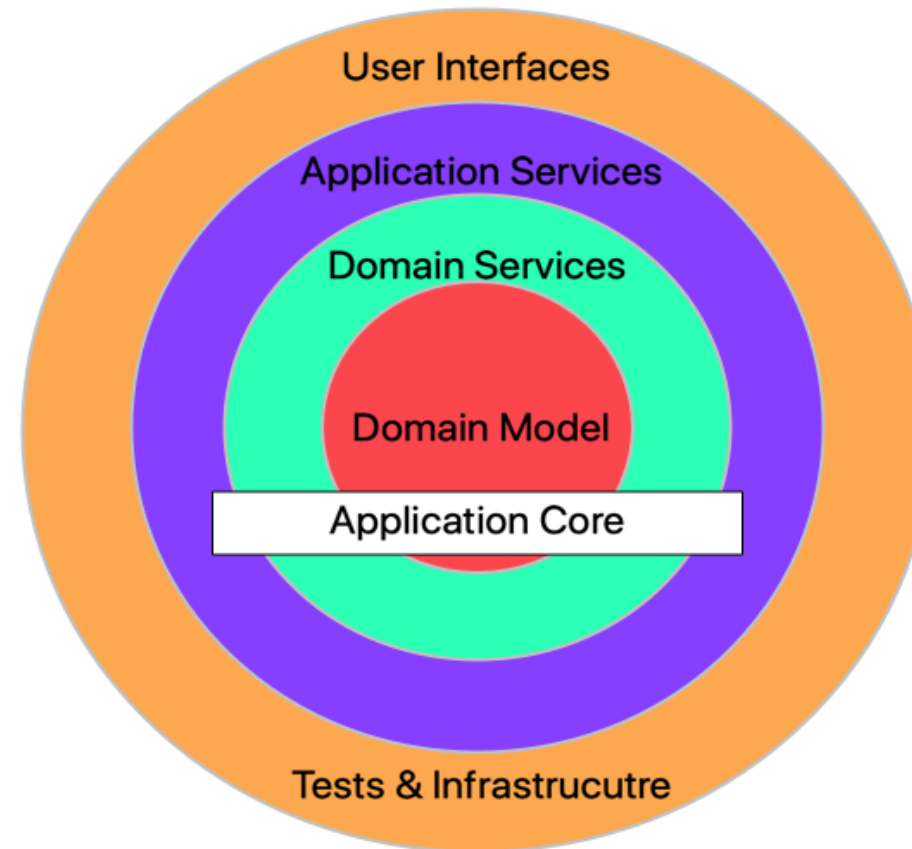




Introduit par Jeffrey Palermo en 2008.

Fournit une solution aux problèmes des architectures à 3 et n couches:

- **Couplage fort.**
- **Séparation des préoccupations.**

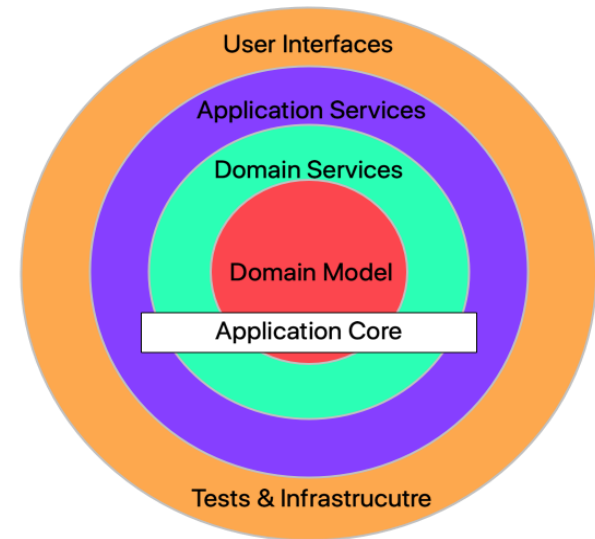




Architecture propre (architecture en oignon)

Le flux des dépendances va du cercle extérieur vers le cercle le plus intérieur

- La couche **Noyau de l'application** tire son nom de sa position au cœur de ce diagramme. la couche Noyau de l'application n'a aucune dépendance vis-à-vis des autres couches de l'application.
- Juste après vers l'extérieur, mais toujours dans la couche Noyau de l'application, viennent **les services de domaine**, qui implémentent généralement les interfaces définies dans le cercle central.

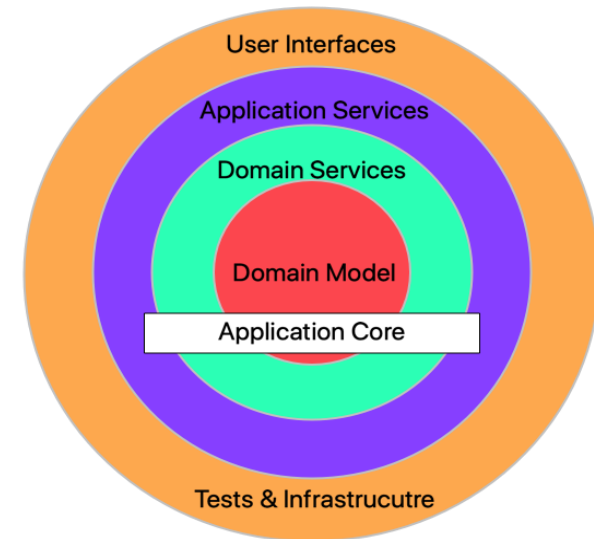




Architecture propre

Noyau de l'application Infrastructure* Interface utilisateur

- **La couche Noyau de l'application (Application Core)** contient le modèle métier, qui définit les entités, les services et les interfaces.
- **Le projet Infrastructure** inclut généralement des implémentations de l'accès aux données. Dans une application web ASP.NET Core conventionnelle, ces implémentations comprennent la classe DbContext d'Entity Framework (EF), les objets Migration EF Core qui ont été définis, ainsi que les classes d'implémentation de l'accès aux données.
- **La couche Interface utilisateur** est le point d'entrée de l'application. Ce projet doit référencer le projet Noyau de l'application, et ses types doivent interagir avec l'infrastructure uniquement par le biais des interfaces définies dans la couche Noyau de l'application



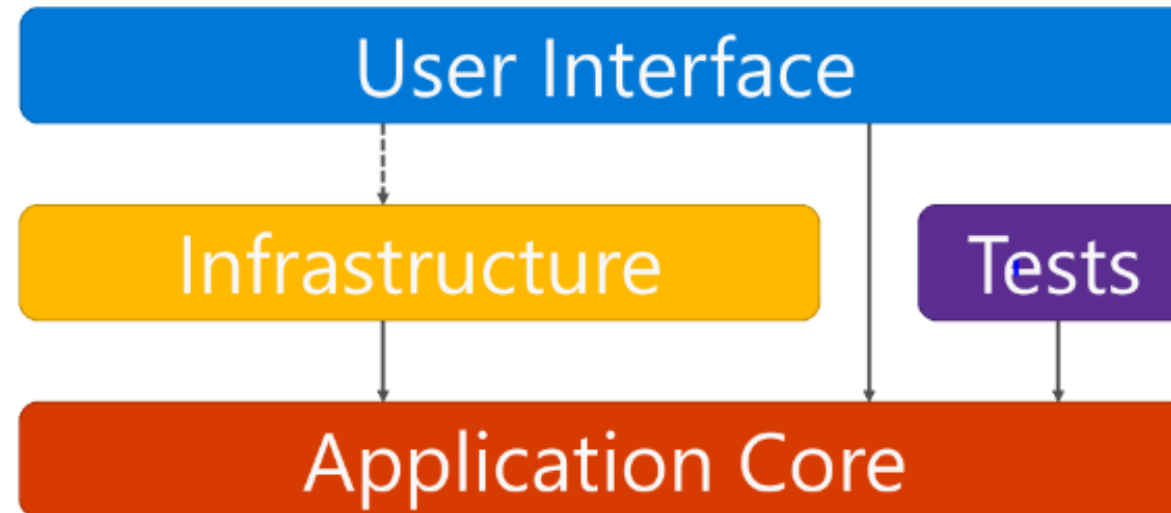


Architecture propre

Représentation horizontale des couches

Clean Architecture Layers

---> Optional Compile-Time Dependency
--> Compile-Time Dependency



Architecture propre ; représentation horizontale des couches