



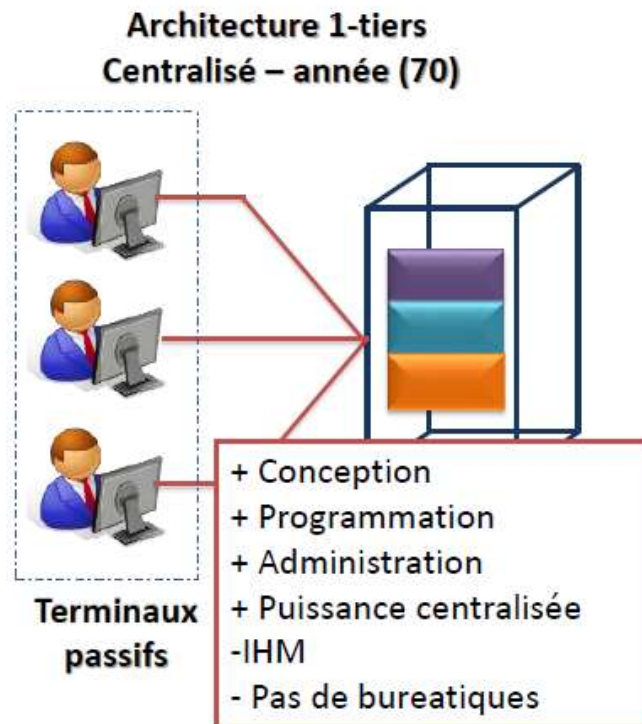
# ARCHITECTURES N-TIERS

---

Cours présenté par Tarek Ayari

# 1-TIERS

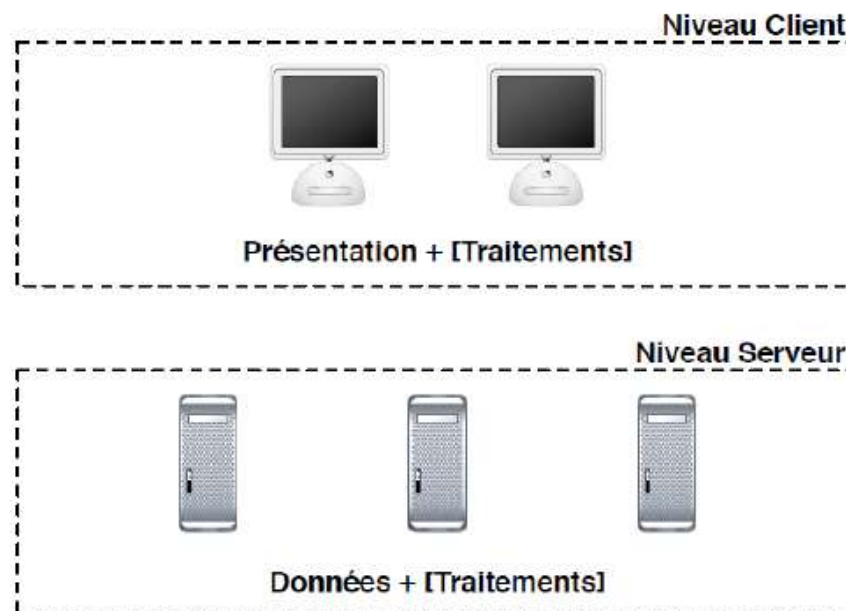
Dans une approche d'application de type 1-tiers, les **trois couches**(présentation, traitement, données) sont fortement et intimement liées, et **s'exécutent sur la même machine**



## 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 traitement** peut être située sur le **client** (au sein même de l'IHM), 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**

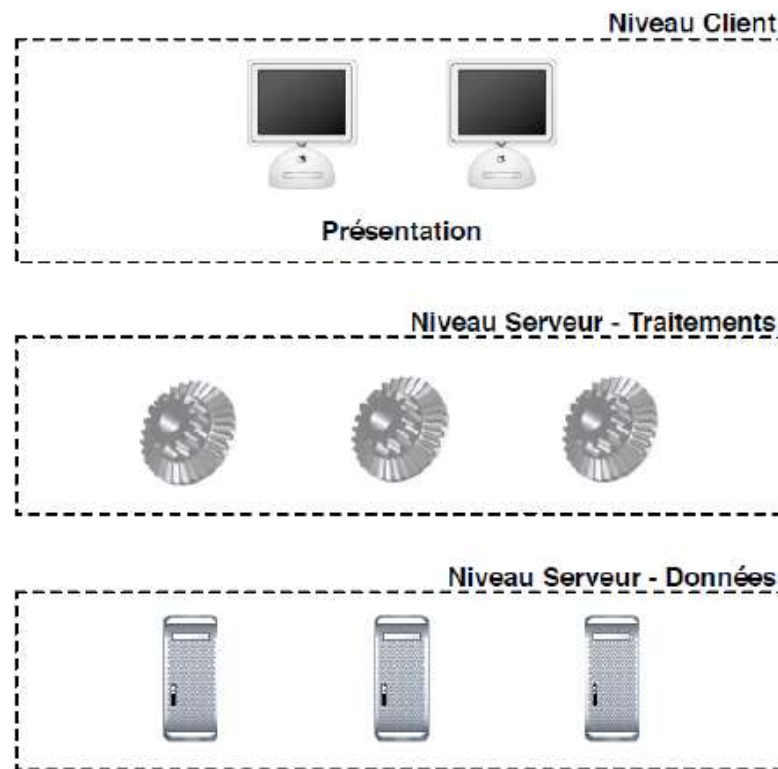
# 2-TIERS



# 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

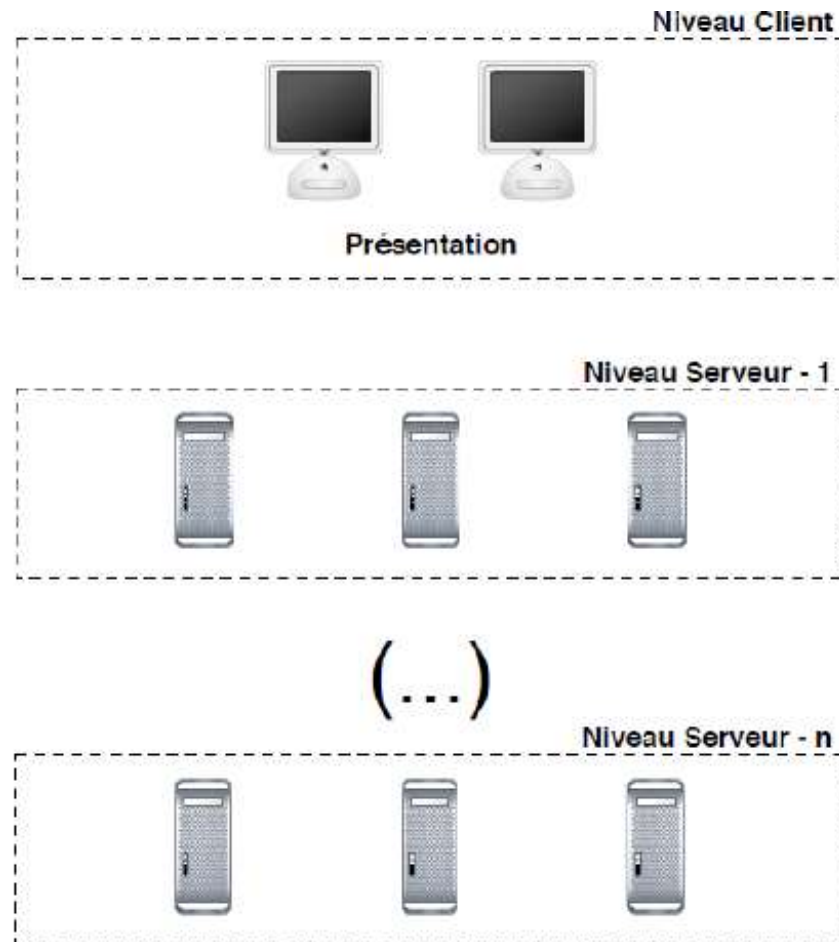
# 3-TIERS



# N-TIERS

- Généralisation de l'architecture 3-tiers
- La couche serveur peut être divisée en autant de sous-rôles que voulus
- Sensiblement les mêmes avantages et inconvénients que la couche 3-tiers.

# N-TIERS





# EXEMPLE: ARCHITECTURE 4-TIER

