

Styles Architecturaux

PLAN DU COURS

- STYLES ARCHITECTURAUX
 - DÉFINITION D'UN STYLE ARCHITECTURAL
 - ARCHITECTURE EN COUCHES
 - ARCHITECTURE EN TIERS
 - ARCHITECTURE MONOLITHIQUE
 - ARCHITECTURE MICRO SERVICE
 - MODÈLE DE 4+1 VUES



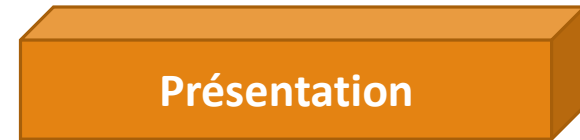
Qu'est ce qu'un style architectural?

- L'architecture d'un bâtiment, par exemple, suit un style particulier, tel que Oriental, American ou Tunisien.
- En architecture, un style architectural détermine le vocabulaire des composants et des relations qui peuvent être utilisés dans les instances de ce style, ainsi qu'un ensemble de contraintes sur la façon dont ils peuvent être combinés
- Une application utilise généralement une combinaison de styles architecturaux

Architecture en couches

Un système est découpé logiquement en trois couches d'abstraction:

- IHM: Interface Homme Machine
- Assure la communication entre l'utilisateur de l'application
- Gère la présentation des données
- Couche applicative qui assure le traitement métier des données,
- Partie fonctionnelle de l'application, qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche de présentation.
- Regroupe l'ensemble des mécanismes permettant la gestion des données stockées dans une base de données
- Gère l'accès aux données de l'application



Architectures en couches

- Réduction de la complexité,
- Amélioration de la modularité (diviser le programme en module),
- Réutilisabilité,
- Maintenabilité,

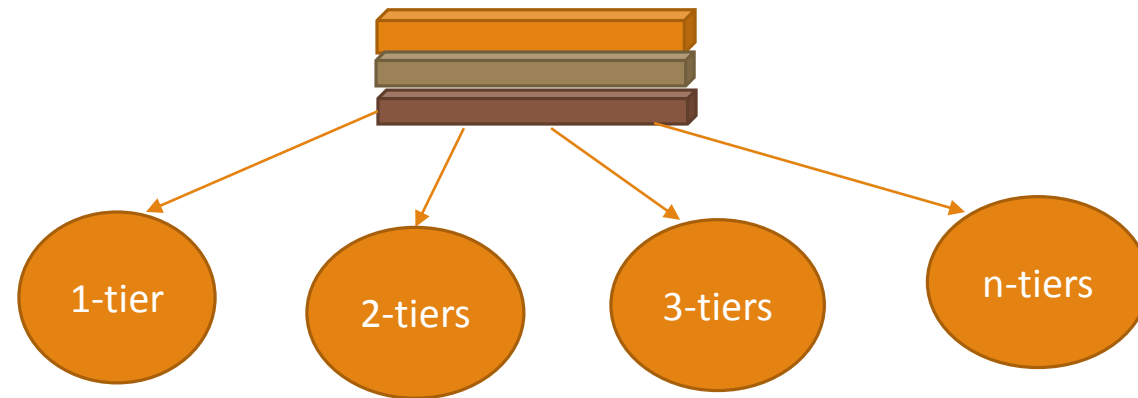


Critères d'évaluation

Développement	Déploiement	Testabilité	Performance	Evolutivité
Elevé: modèle non complexe à mettre en œuvre	Faible: difficulté de déploiement pour les applications volumineuses. La modification d'un composant nécessite le redéploiement de toute l'application	Elevé: les couches sont testées séparément	Faible: inefficacité de devoir passer par plusieurs couches de l'architecture pour répondre à une demande métier.	Faible: difficulté de mettre à l'échelle une application en couches, vu leurs couplage fort

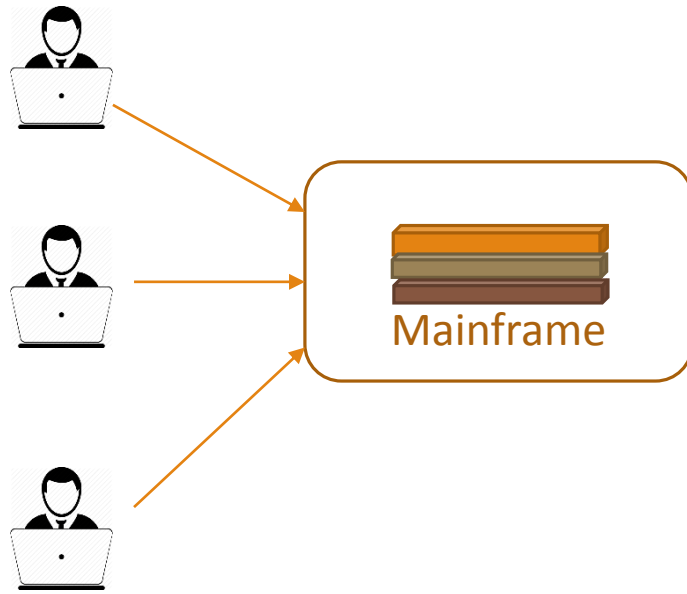
Architecture en tiers

- Ces trois couches logiques peuvent être imbriquées ou réparties entre différentes machines physiques et/ou virtuelles,
- Le découpage et la répartition de ces couches permettent de distinguer des architectures applicatives qui vont de l'architecture un-tiers à n -tiers.



Architecture 1-tier

- Les trois couches s'exécutent sur une même machine
- Dans des systèmes multi-utilisateurs, on peut avoir une application installée sur **un site central** avec des **clients passifs** possédant des terminaux



- + Facilité d'administration
- + Utilisation centralisée de ressources
- Interface graphique en mode caractère

Architecture 1-tier: Exemple

```
Account: 12345678      Billing Information      XYL1001
Company: 50            Account Summary

Plan: B2 QUARTERLY PAY PLAN      Status: ACCOUNT PAID TO DATE
Name: JONATHAN & SUE SMITH      Agent: BRING TO SELL AGENCY
      DB BURWELL BOAT ZINCS      UP3214
      P.O. BOX 987                P O BOX 12

      SOUTHEND-ON-SEA      SS      THORPE BAY      TB
      SS3 9QL              020 711 2121      SS3 7HN

=====
Inception:      25/10/01      Payment Method: CASH/CHECK
Due Date:       12/12/09      Bill Day:      12
Next Bill:      12/12/09      Suspend Billing: NO
Next Action:    12/12/09      Next Action Reason: BILL
Last Payment:   15/07/09      Unpaid NSF:    NO
Last Activity:  11/11/09      Cancel Outstanding: NO
Account Payer:  INSURED      Remarks Present: NO
Insured Min Due:      0.00      Est Next Bill:      3425.00
Insured Balance:      7857.00      Service Charge:      0.00
Total Min Due:        0.00      Pending Cash:        0.00
Total Balance:        7857.00      Cash Entered Today:  0.00

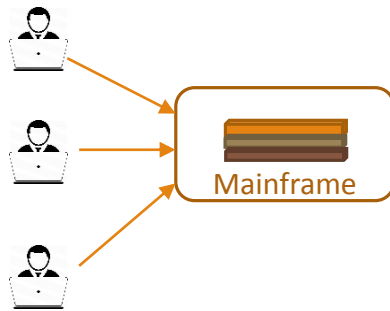
F1=Help  F3=Exit  F10=Policy List  F11=Account Transactions  F12=Invoices
```



Architecture 2-tiers

Afin de pallier aux défaillances de l'architecture 1-tier, il faut concevoir une architecture qui conserve les avantages du site central et ceux du partage de fichiers.

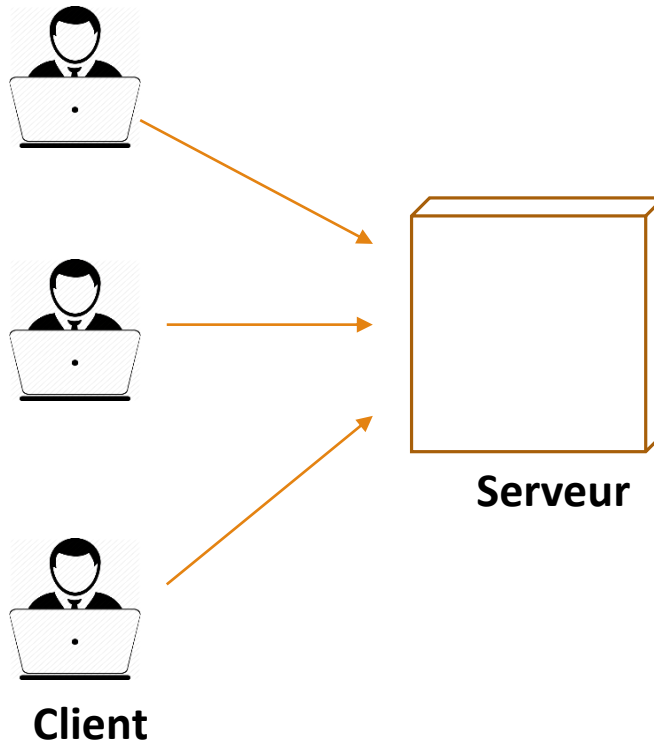
- Fiabilité des solutions sur site central
- Ergonomie de l'IHM



Architecture 2-tiers

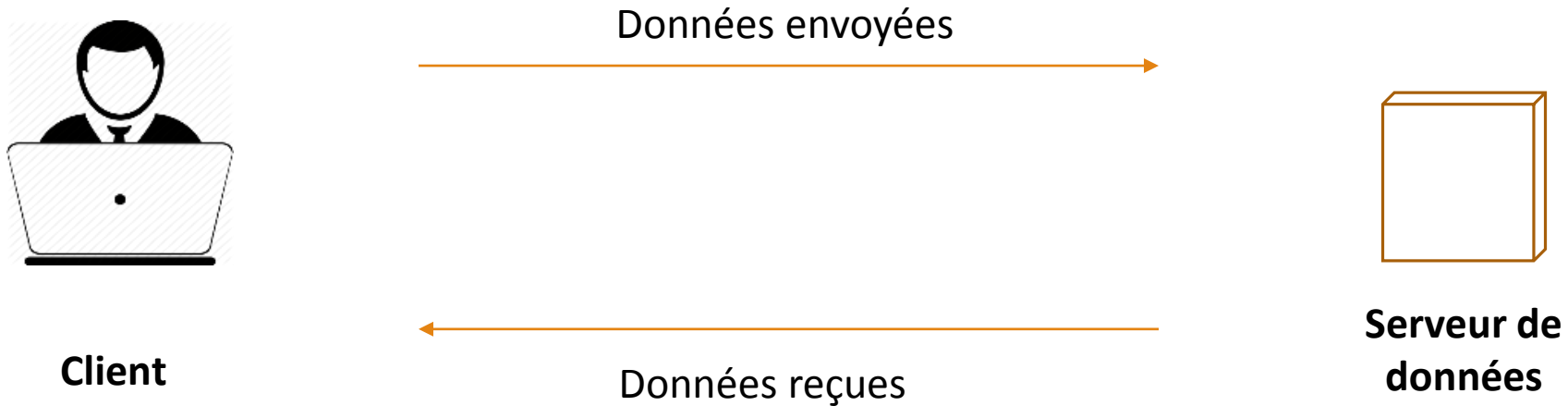
Architecture 2-tiers

- Gestion centralisée de données
- Gestion locale de l'interface utilisateur
- Architecture **Client/Serveur**

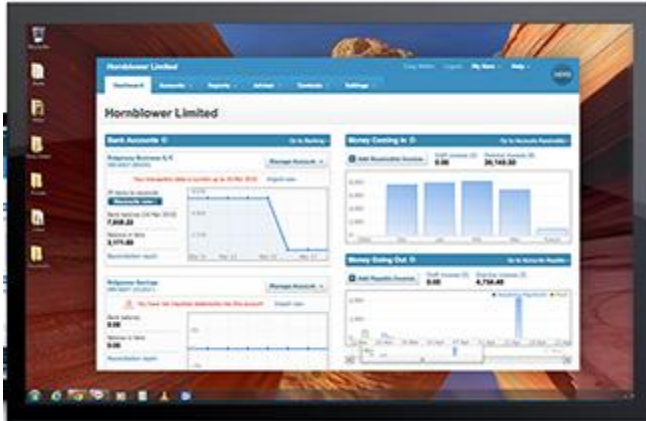


Architecture 2-tiers

- Architecture **Client/Serveur de données**
- La gestion de données est déléguée à un serveur spécialisé (SGBD)



Architecture 2-tiers



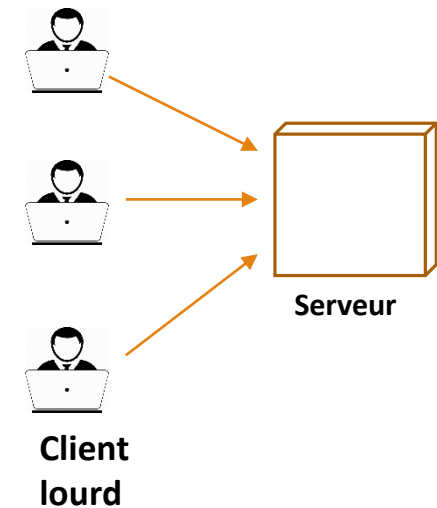
Client



Serveur de données

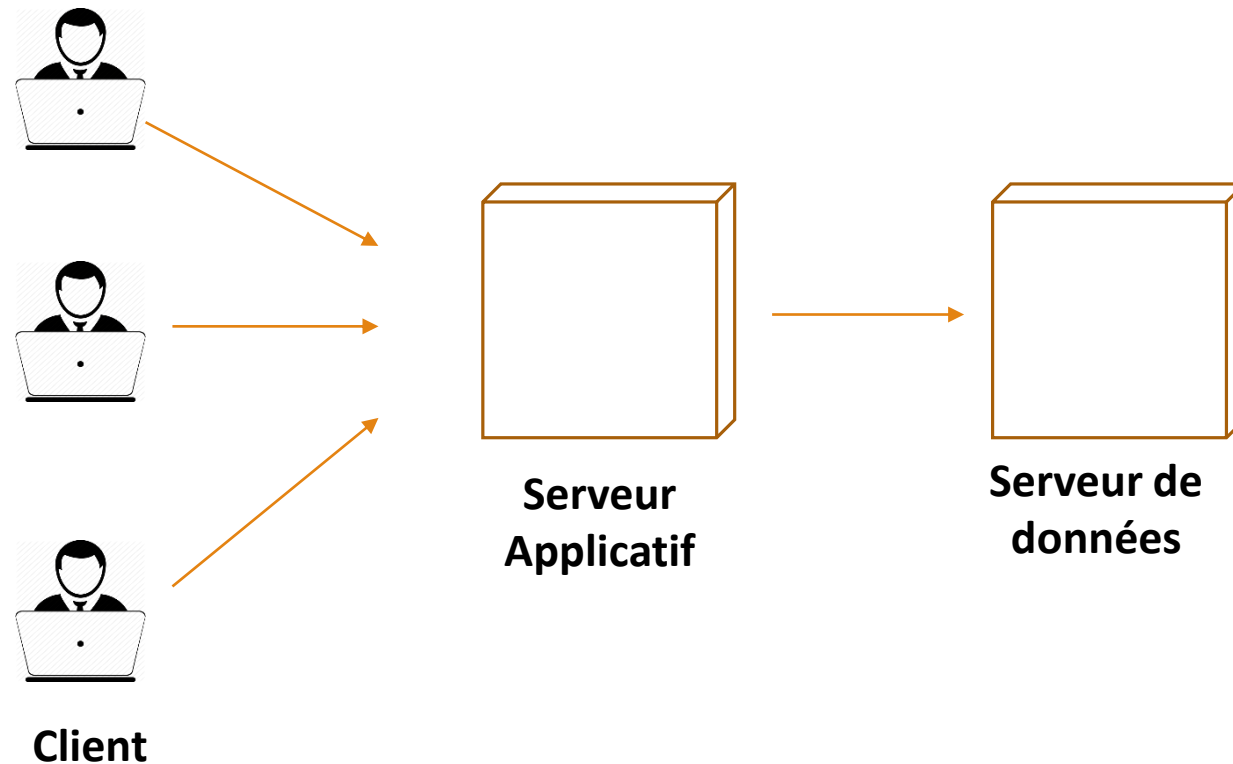
Architecture 2-tiers

- + Interface riche
- + Données centralisées
- + Intégrité de données
- Le client supporte la grande majorité des traitements applicatifs
- Complexité de la mise à jour régulière pour répondre aux besoins des utilisateurs
- Complexité d'évolution



Architecture 3-tiers

- Résoudre les limites et conserver les avantages de l'architecture 2-tiers, conduit à avoir l'architecture 3-tiers.

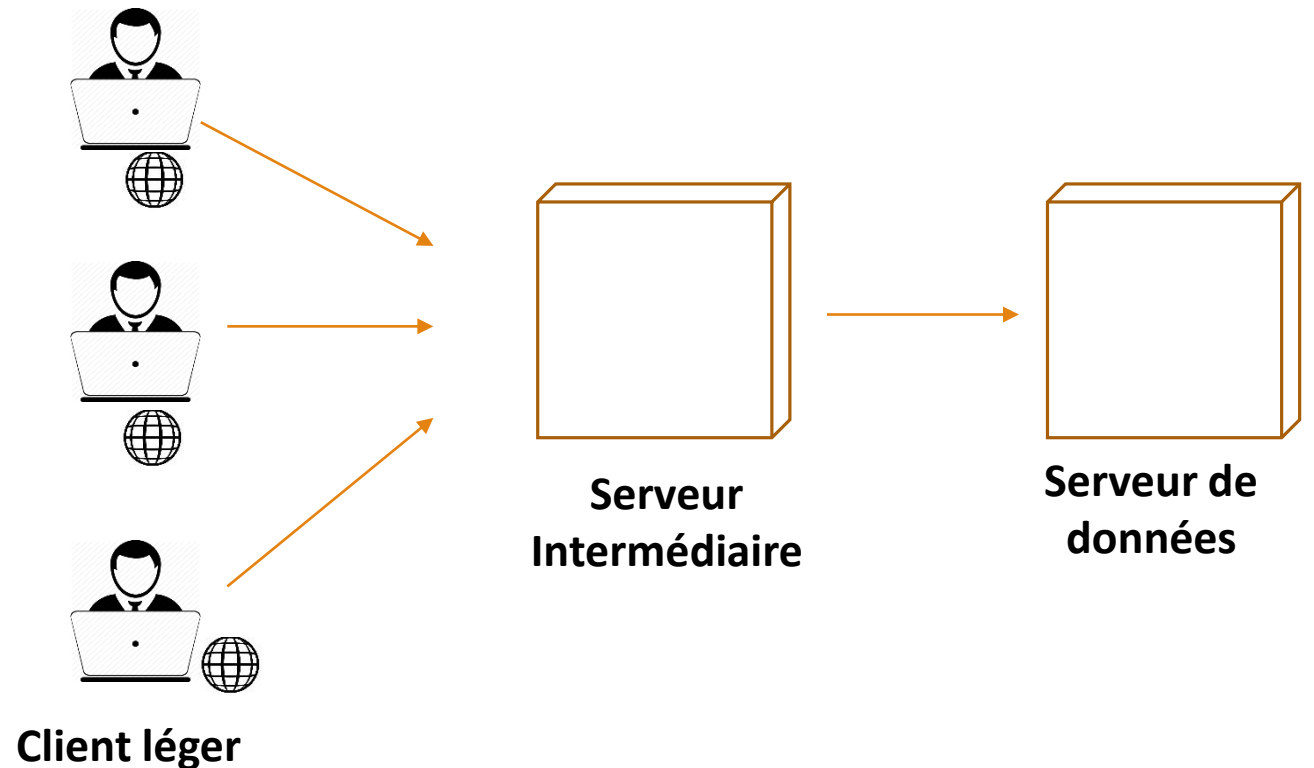


Architecture 3-tiers

- + Client allégé
- + Meilleure répartition de la charge entre différents serveurs d'application.
- + Amélioration de la sécurité des données, en supprimant le lien entre le client et les données.
- + Le serveur a pour tâche, en plus des traitements purement métiers, de vérifier l'intégrité et la validité des données avant de les envoyer à la couche de données.

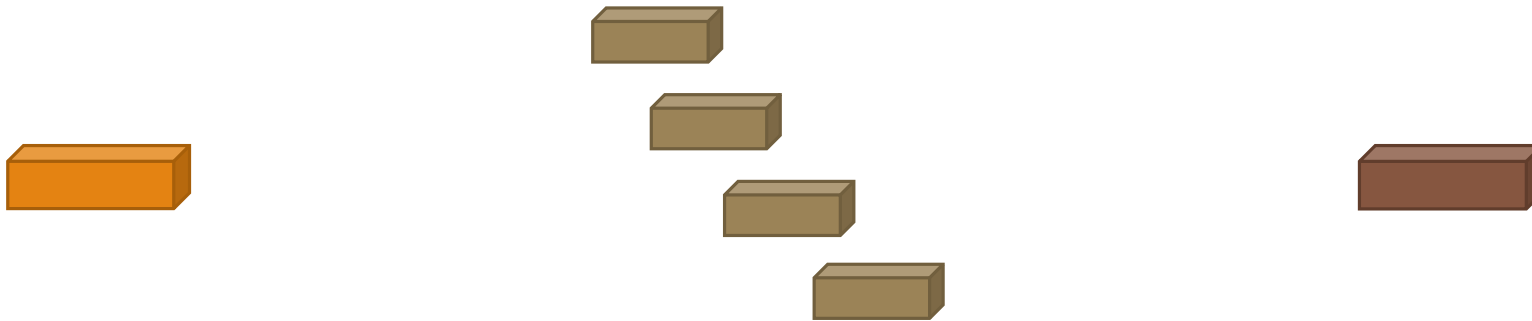
Architecture 3-tiers

- Avec l'arrivée du WWW (World Wide Web), le client devient léger et ne supporte aucun traitement,
- Limites:
 - Interface ergonomique moins riche
 - Serveur intermédiaire fortement sollicité
 - Montée en charge



Architecture n-tiers

- Pour pallier aux problèmes de l'architecture 3-tiers, il faut réfléchir à:
 - Distribuer plus librement la logique applicative,
 - Faciliter la répartition de la charge entre tous les niveaux



Logique applicative distribuée

Architecture n-tiers

- Les composants de la couche applicative sont spécialisés et indépendants
- Communiquent entre eux et peuvent exister sur des machines distinctes

