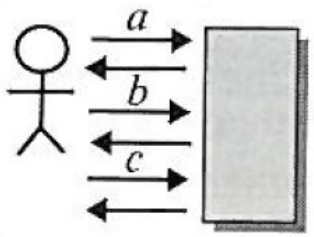
Styles d’architecture

Différents types de systèmes

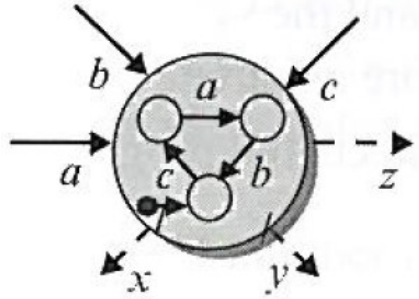
1. Interaction système et acteur, système répond à chaque requête de l'acteur, acteur commence et met fin à chaque interaction/processus
2. Système a un état, comportement réactif, reçoit des événements d'entités externes (ex: matériels, pas humain). Tous les événements ne sont pas forcément traités.
3. Réseau d'Activités de traitement d'info, typiquement en mode batch (peu ou pas d'interaction), pas d'état.
4. Systèmes persistants: capable de sauvegarder et rechercher des objets tout en isolant le support de stockage

1



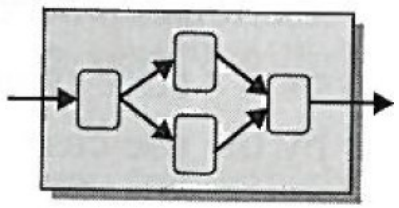
Système **interactif**

2



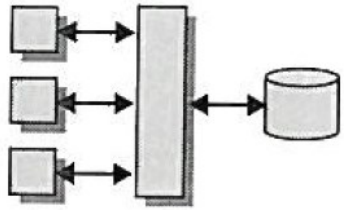
Système **dirigé par évènement**

3



Système **transformationnel**

4



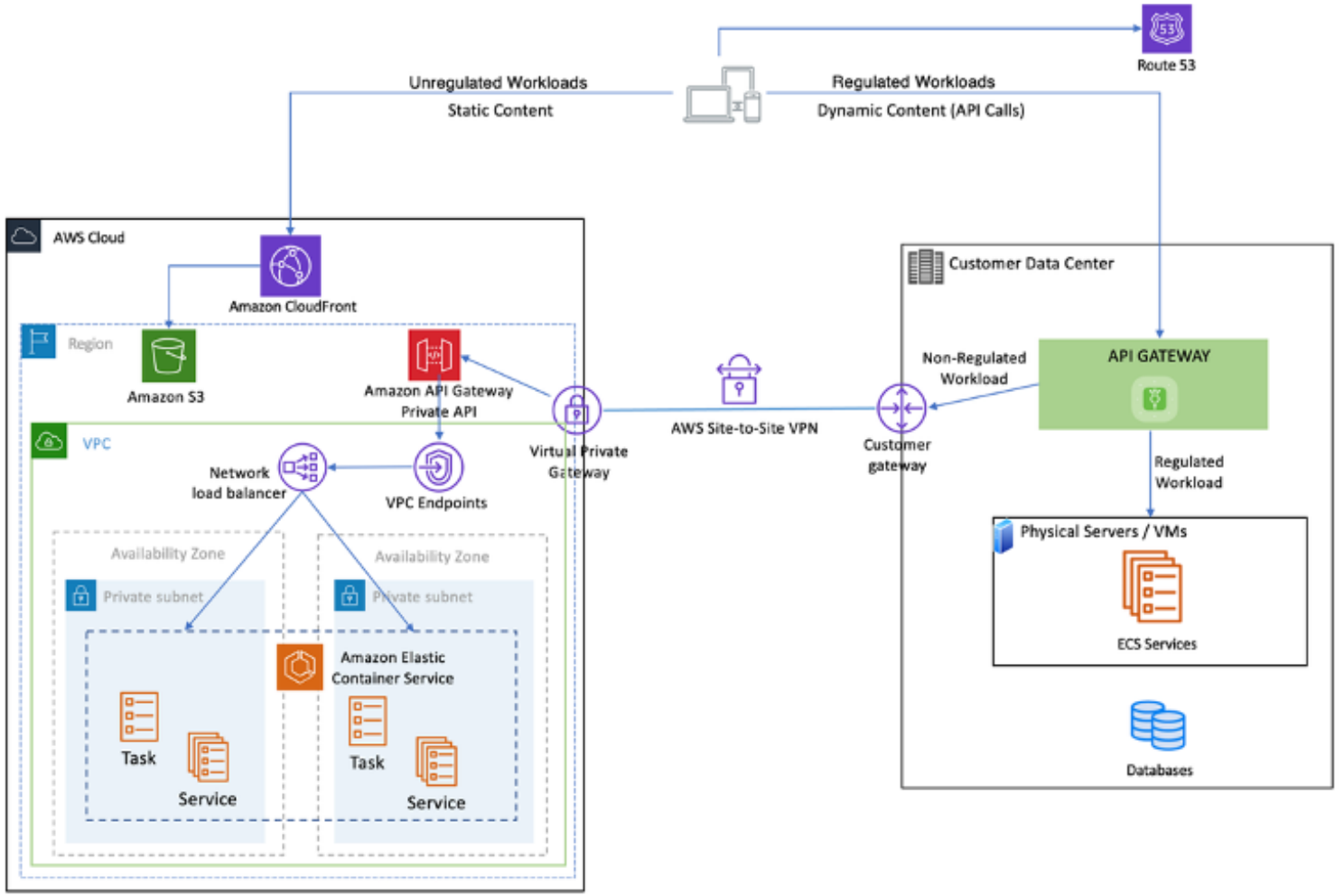
Système de **base de données**

Systèmes hybrides

La plupart des **systèmes** complexes sont **hybrides**.

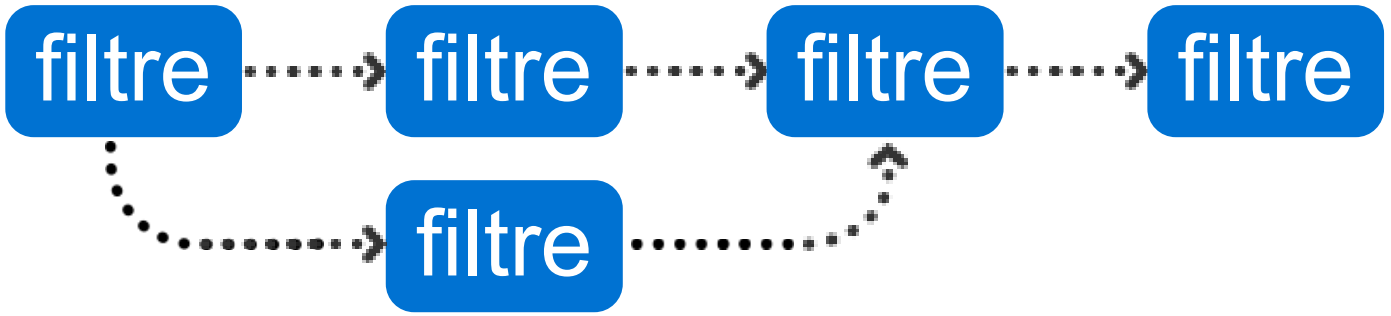
Le type du système complet est généralement une combinaison de différents types de sous-systèmes.

- Application interactive
- Mise à jour automatique dirigé par évènements
- Long traitement par une série de transformations
- Gestion des données



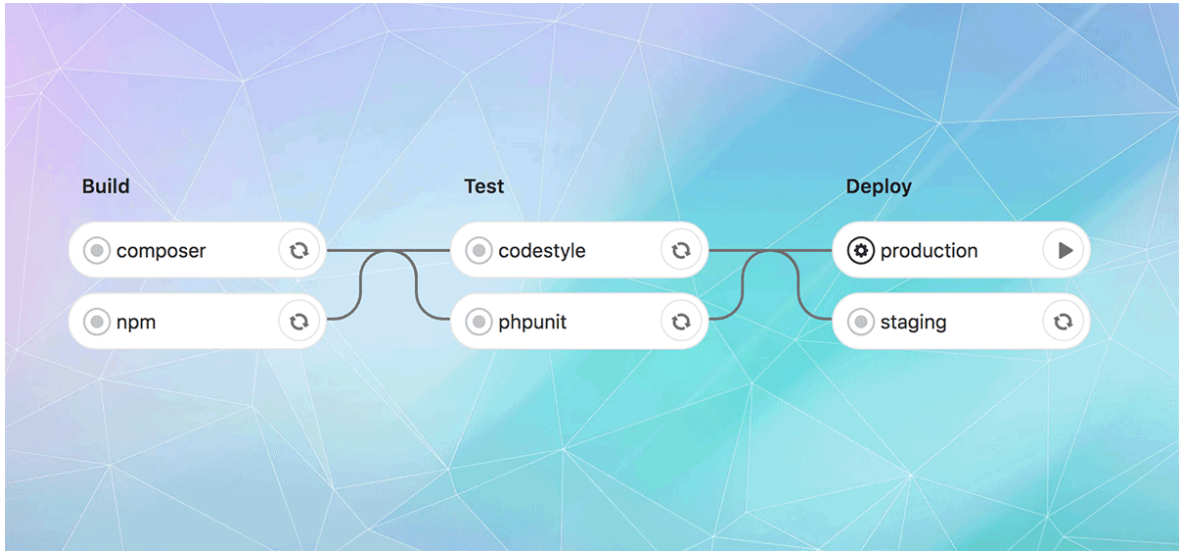
Pipes et filtres

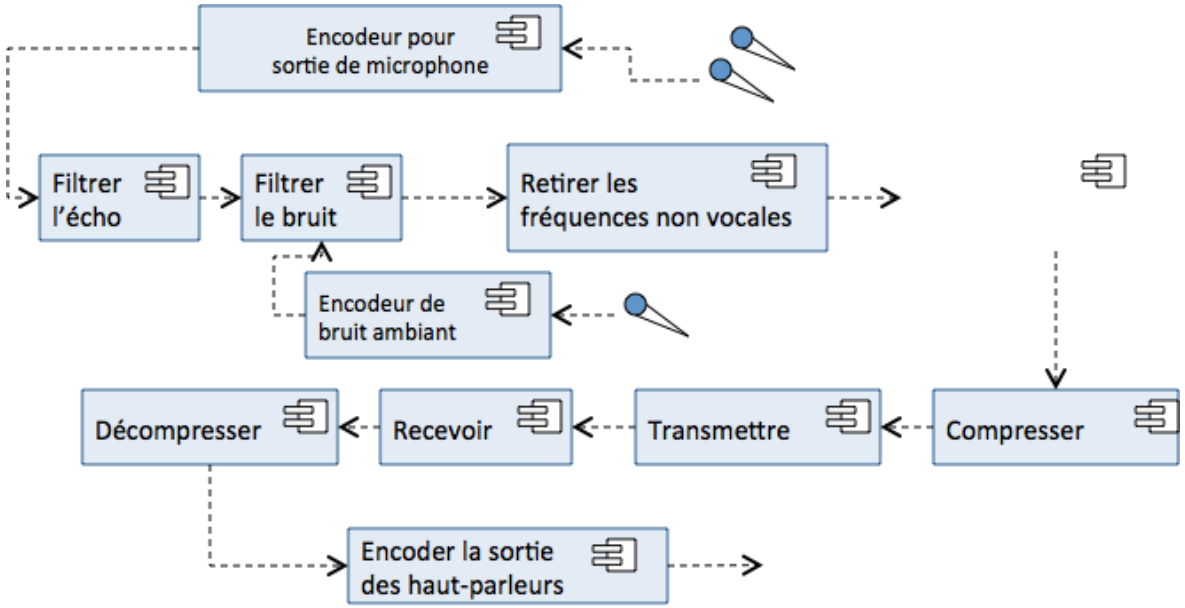
- Modules organisés en filtres communiquant via des pipes
  - Communication locale
- Traitement batch en plusieurs étapes
- Exécution concurrente de filtres, synchronisation des flux parfois nécessaire

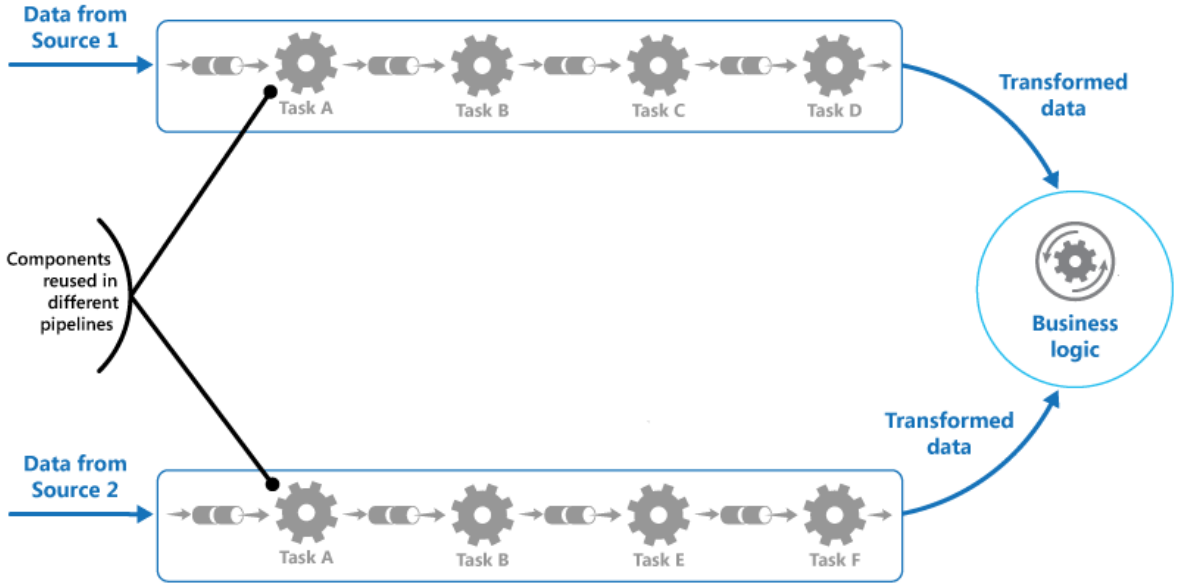


Exemple

Pipe-et-filtre

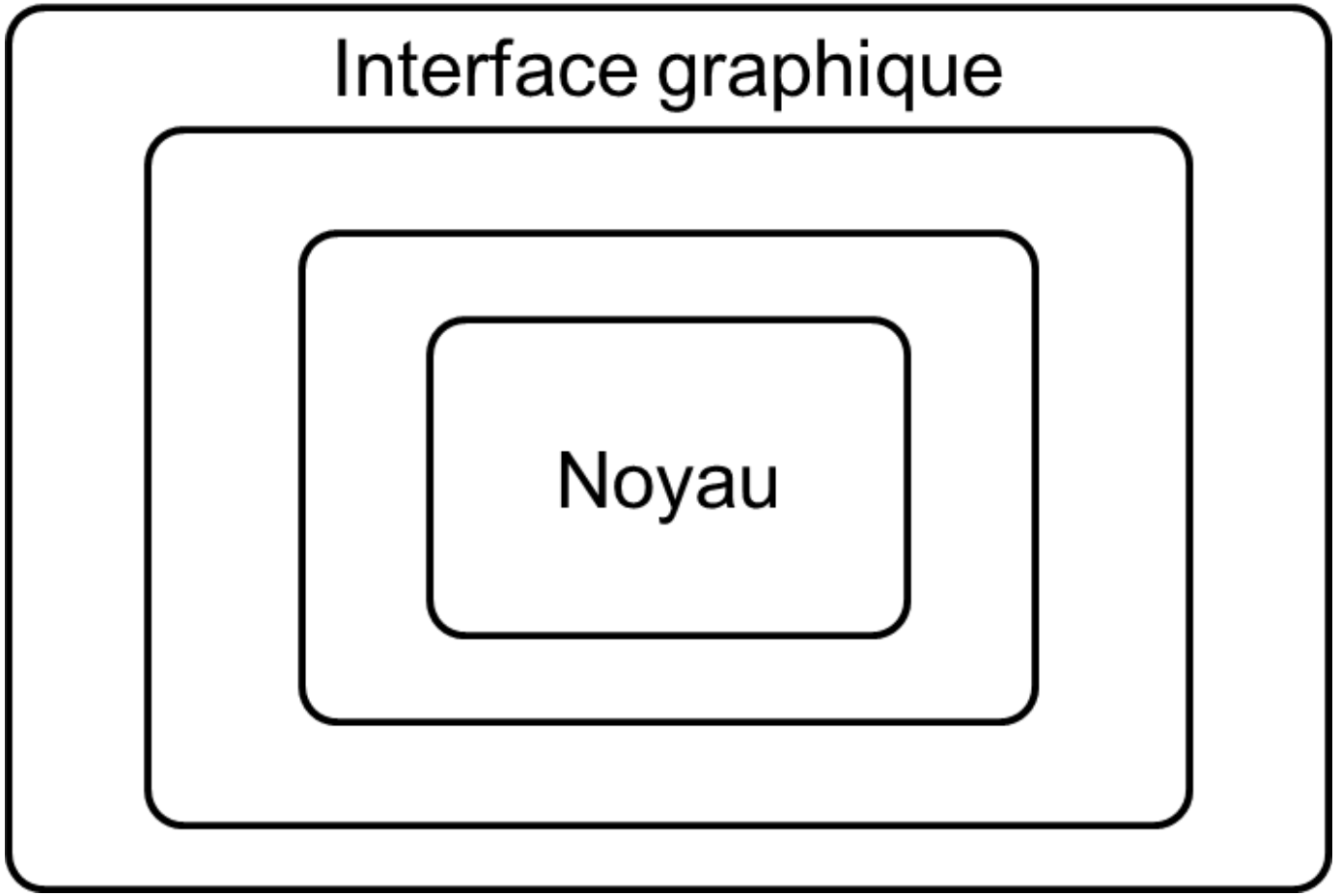






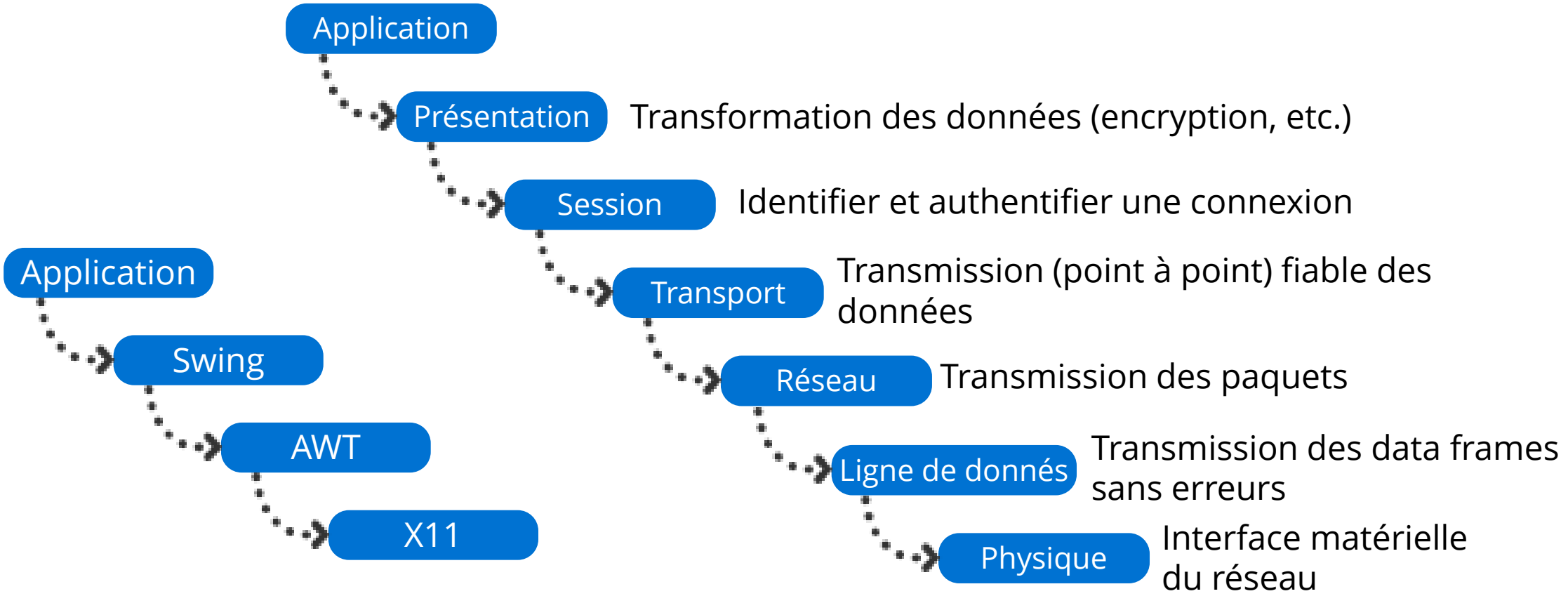
Architecture en couche

- Système organisé en couches hiérarchiques
- Chaque couche fournit un service à la couche supérieur et sert de client à la couche inférieure

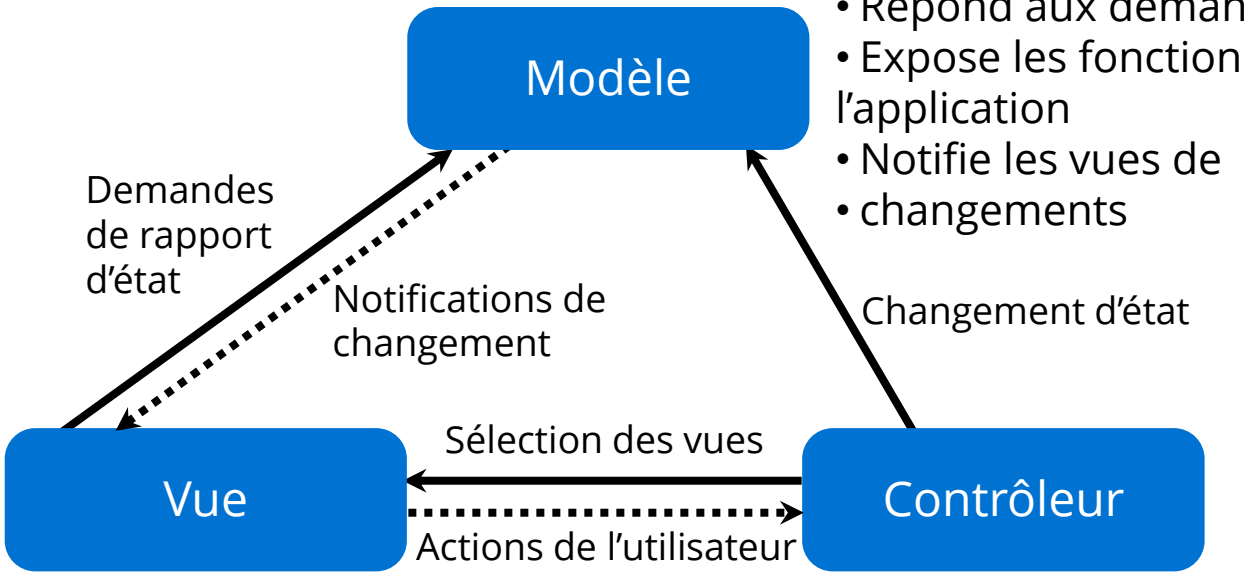


Exemple

En couche



Modèle-vue-controleur (MVC)

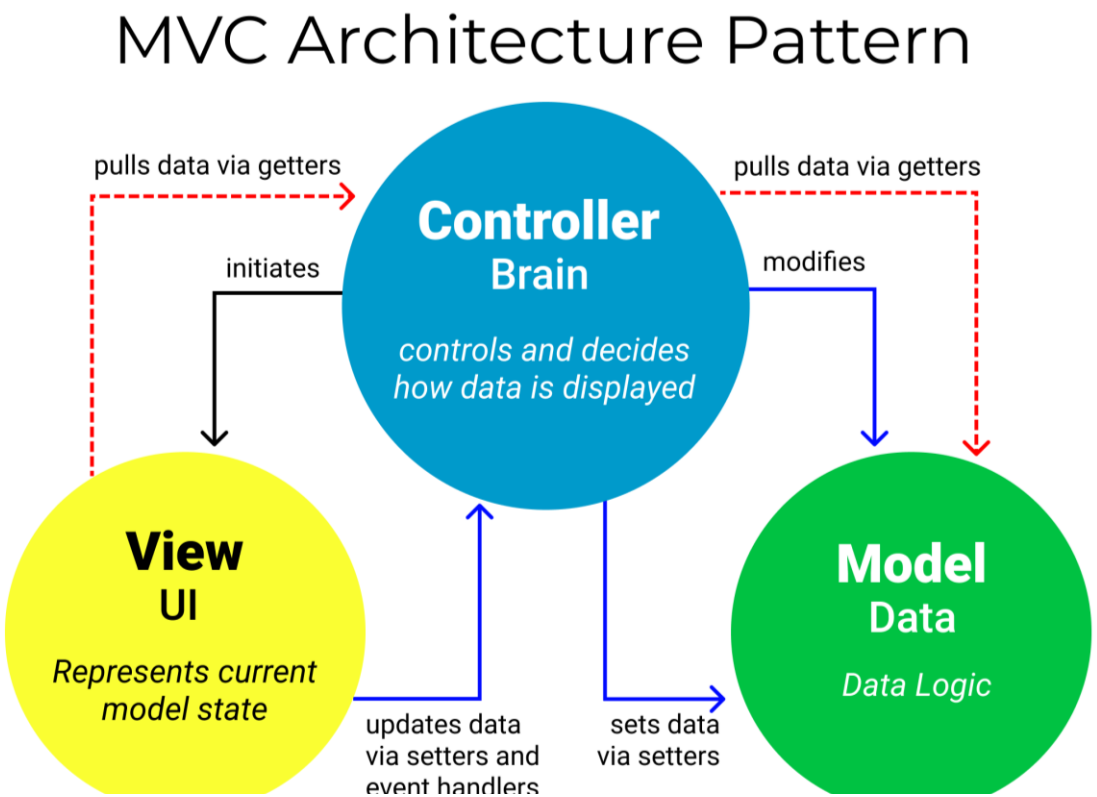


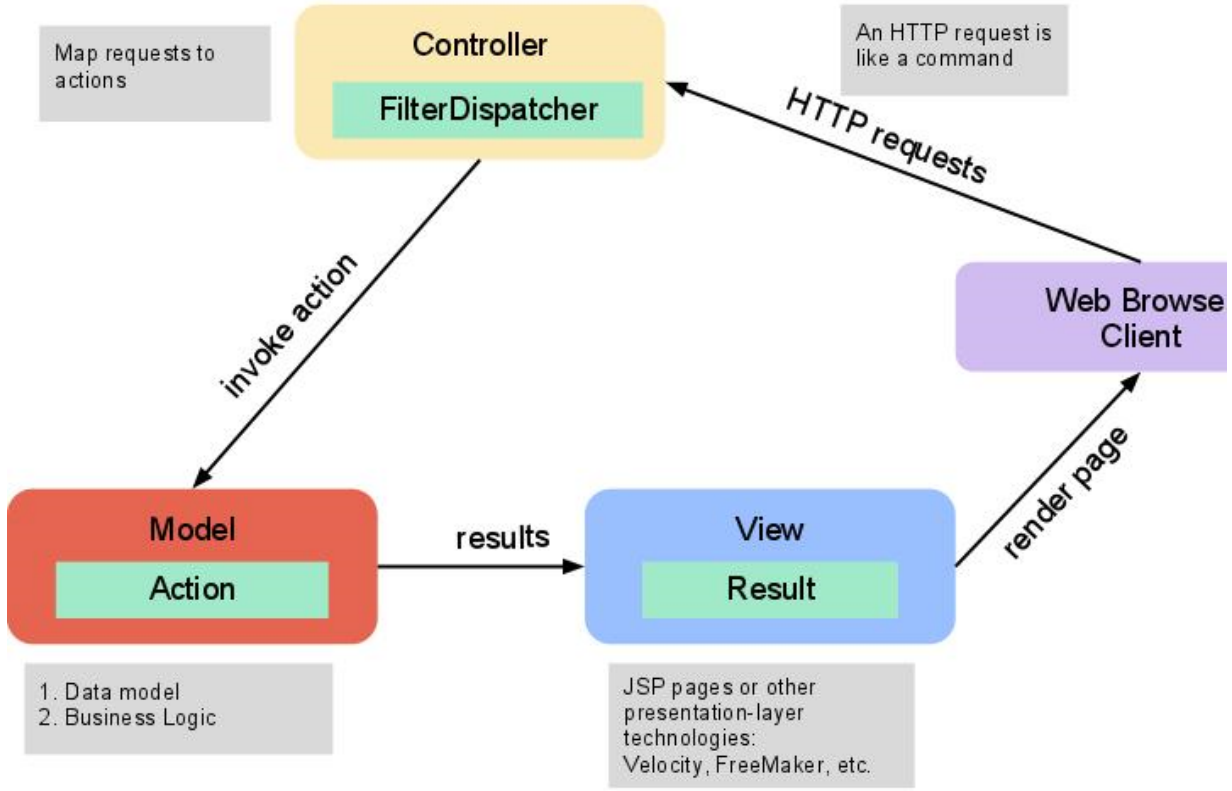
- Encapsule l'état de l'application
- Répond aux demandes d'état
- Expose les fonctions de l'application
- Notifie les vues de
- changements

- Fait le rendu du modèle
- Demande des mises à jour du modèle
- Envoie des actions de l'utilisateur au contrôleur
- Permet au contrôleur de sélectionner une vue

- Définit le comportement de l'application
- Mappe les actions de l'utilisateur aux mises à jour du modèle
- Sélectionne une vue pour la réponse

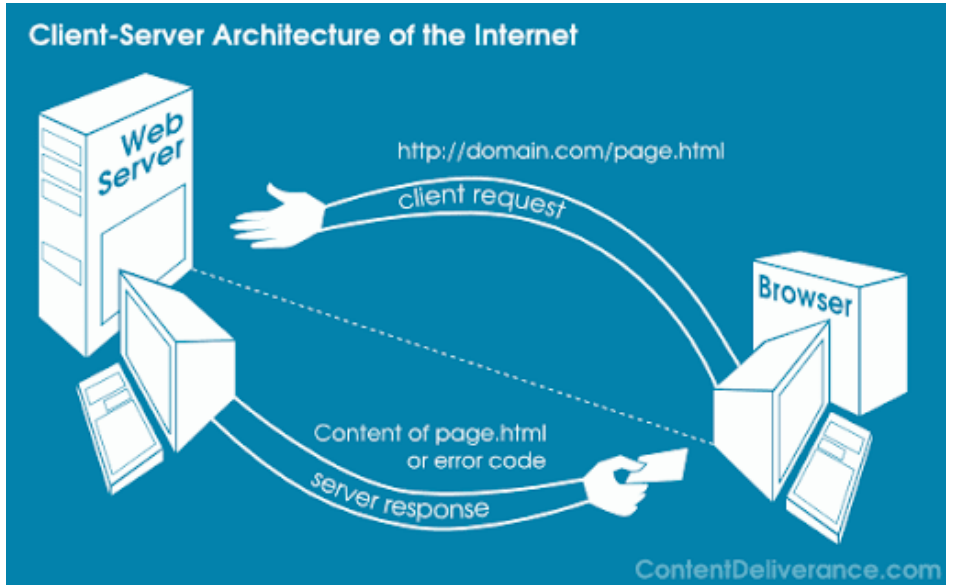
Exemple
MVC

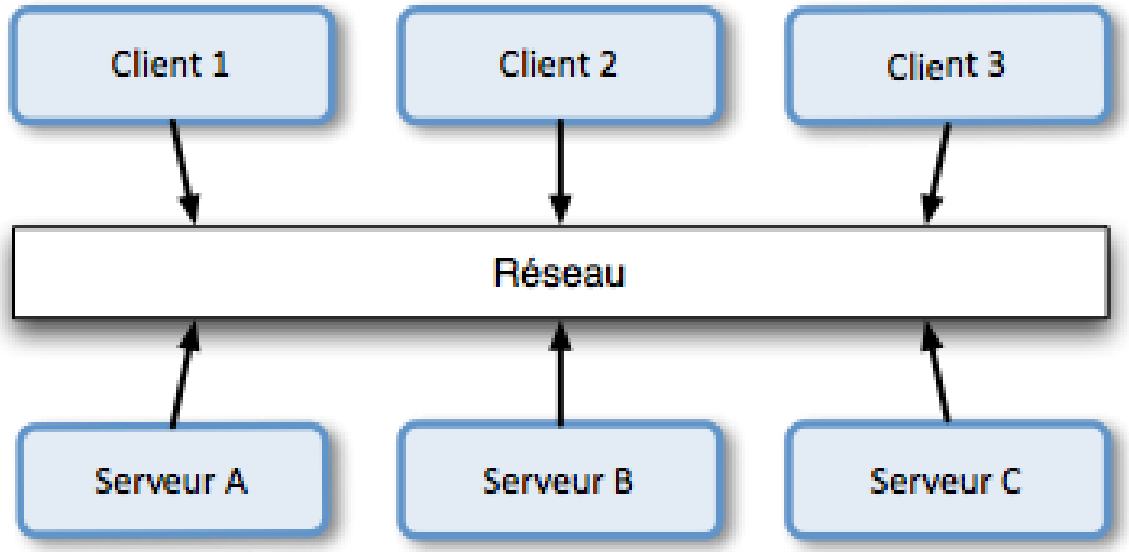

MVC Architecture Pattern
pulls data via getters
Controller
Brain
controls and decides how data is displayed
pulls data via getters
initiates
modifies
View
UI
Represents current model state
Model
Data
Data Logic
updates data via setters and event handlers
sets data via setters


Map requests to actions
Controller
FilterDispatcher
An HTTP request is like a command
HTTP requests
invoke action
Web Browse Client
render page
Model
Action
results
View
Result
1. Data model
2. Business Logic
JSP pages or other presentation-layer technologies:
Velocity, FreeMaker, etc.

Client-serveur

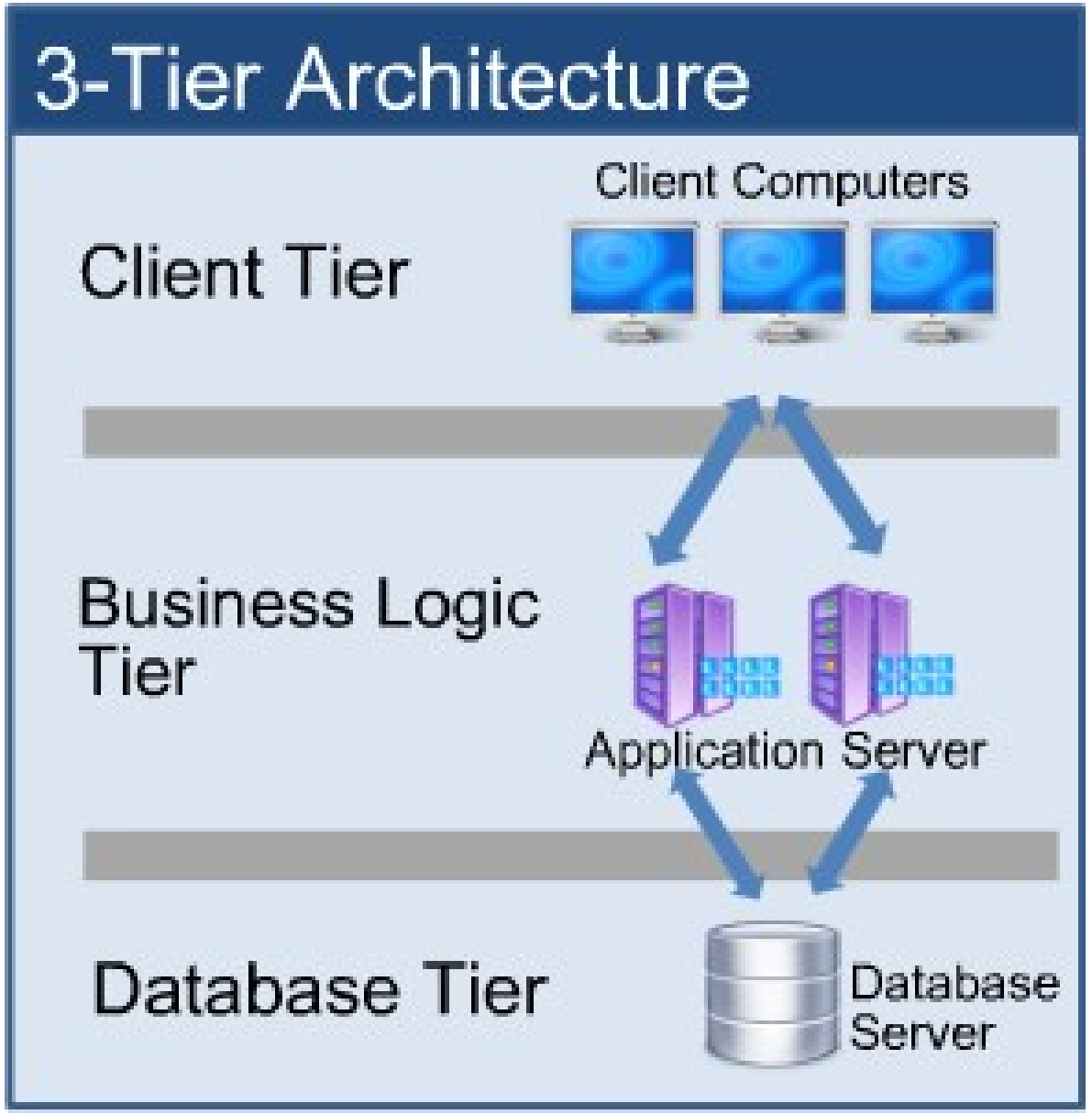
- Clients
  - Reçoivent des services des serveurs
  - Doivent connaître comment contacter les serveurs
    - Ex: adresse IP, port, etc.
- Serveurs
  - Fournissent des services aux clients et autres serveurs
- Extensibilité par ajout de serveurs





3-tier

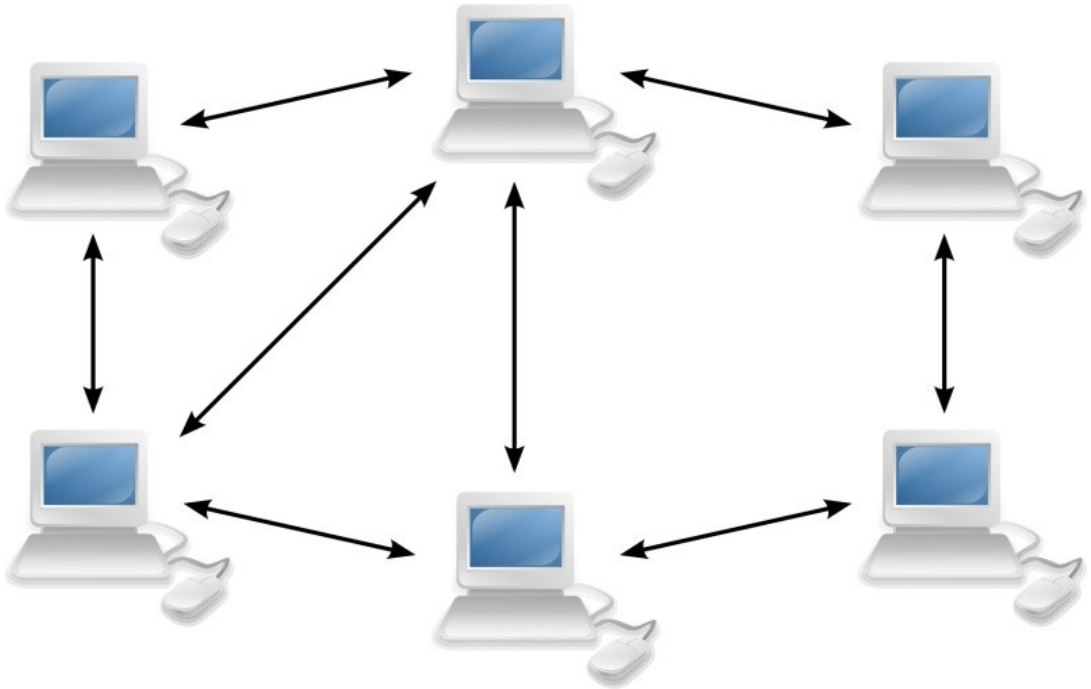
- Client léger: présentation (ex: HTML + JavaScript)
- Logique d'application (ex: PHP, Java)
- Couche des données (ex: SQL)
- Architecture multi-tier

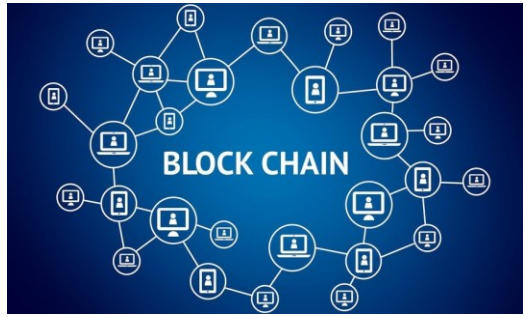


Peer-to-peer

- Chaque nœud joue le rôle à la fois du client et du serveur (aucun serveur central)
- Modèle hybride : un serveur central conserve l'information au sujet des pairs

✓ **Fiabilité (tolérance aux pannes)**
✓ **mise à l'échelle facile (grandit avec le nombre de pairs)**
× **Complexité, rôle plus lourd des pairs**





Micro-services (cloud computing)

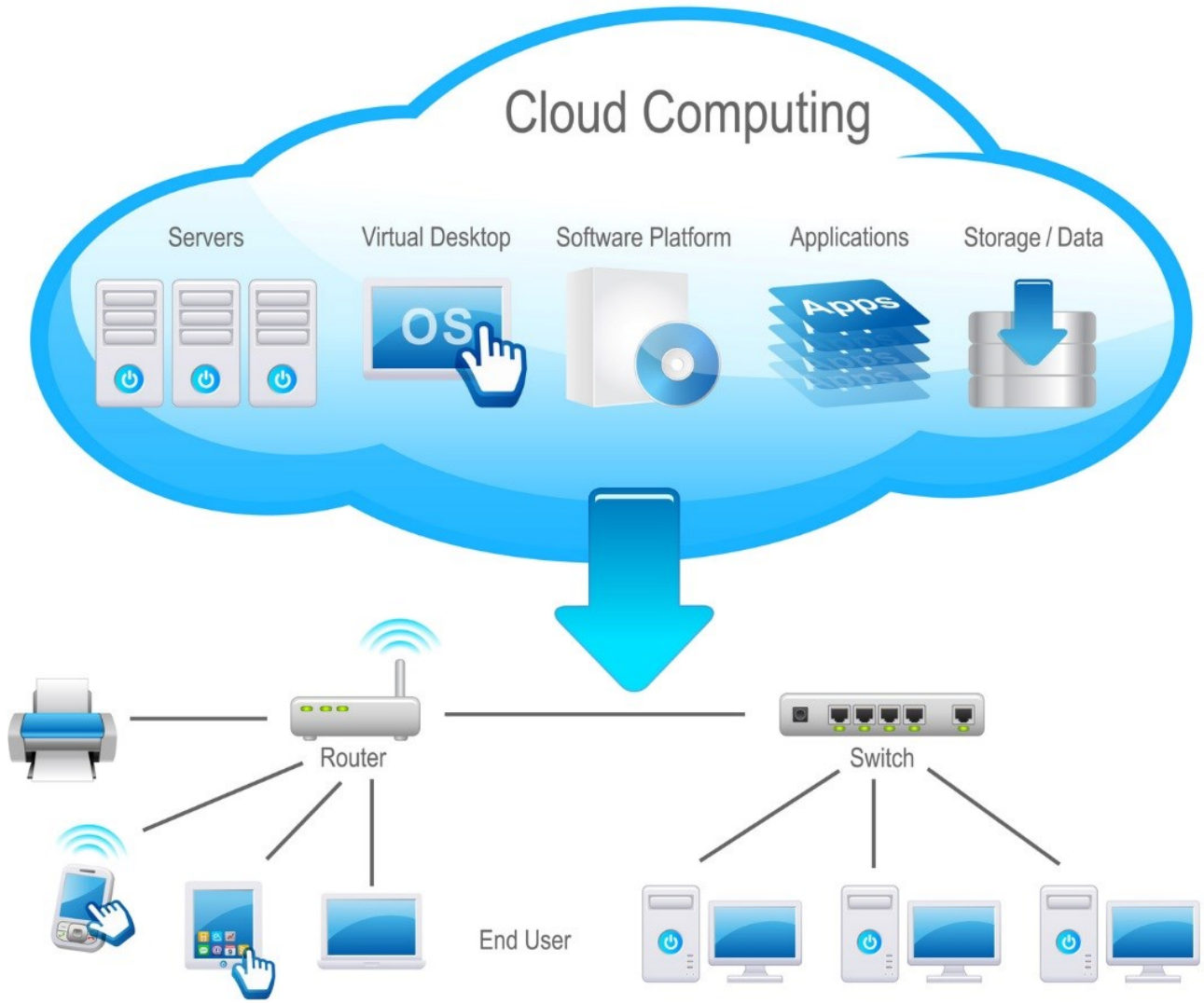
- Serveur est géré par fournisseur de service, accessible par internet

  Ex: Amazon Web Services, Azure

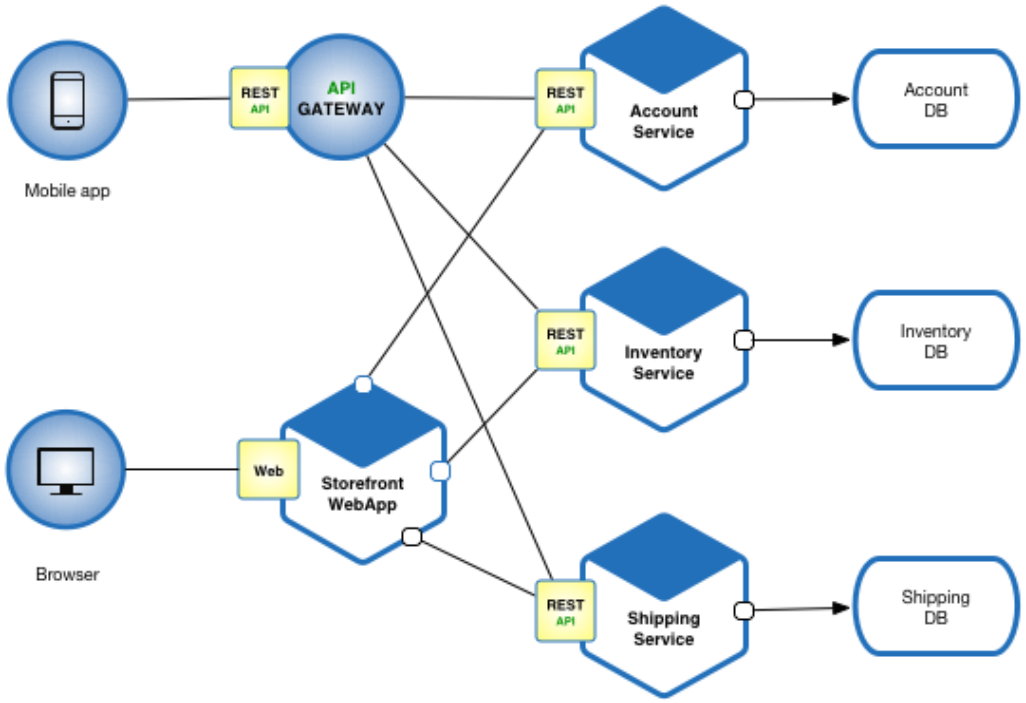
✓ **Facilité d'exploitation et garantie (sécurité, fiabilité)**

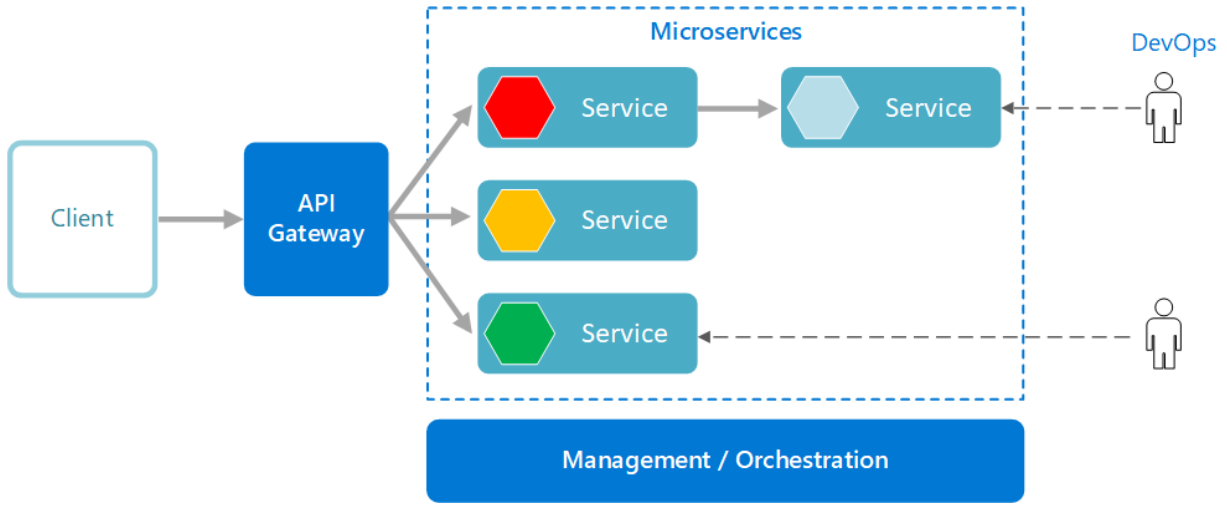
✓ **Mise à l'échelle (scalable)**

× **Contrôle limité**



Exemple
Micro-services





Importance de l’architecture

- L’architecture du logiciel est cruciale au produit
  - Le workflow des exigences peut être corrigé durant le workflow d’analyse
  - Le workflow d’analyse peut être corrigé durant le workflow de conception
  - Le workflow de conception peut être corrigé durant le workflow d’implémentation
- Mais il n’y a **pas moyen de surmonter une architecture non-optimale plus tard**
  - Il faut absolument reconcevoir l’architecture immédiatement!