

Architecture des applications



Objectifs

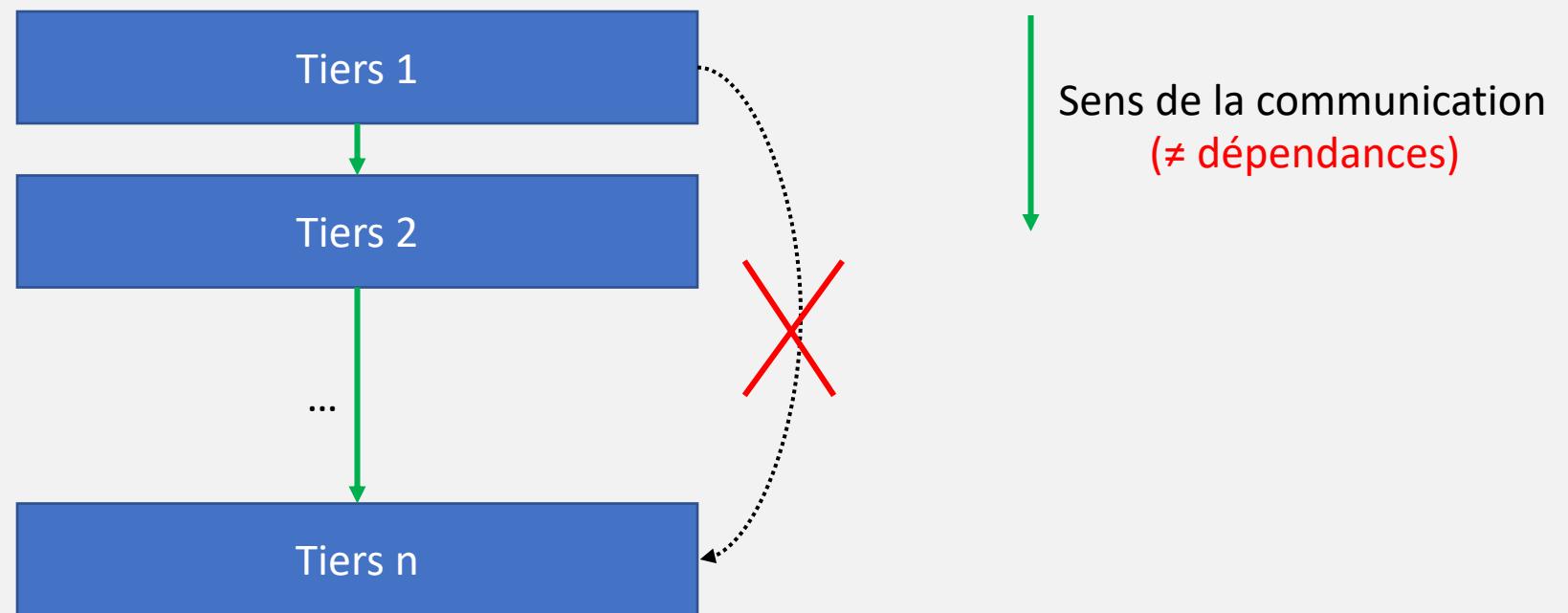
- Introduction aux architectures n-tiers
- Exemples de découpages haut niveau

Architecture logicielle

- L'architecture logicielle réfère à la description de l'organisation des différents éléments d'un ou plusieurs systèmes informatique
 - Relations
 - Communications
- Ici, on ne décrit pas l'aspect fonctionnel d'une solution (le quoi), on décrit les moyens et le comment
- Il existe plusieurs types d'architecture logicielle :
 - N-tiers
 - SOA
 - Microservices
 - Etc.

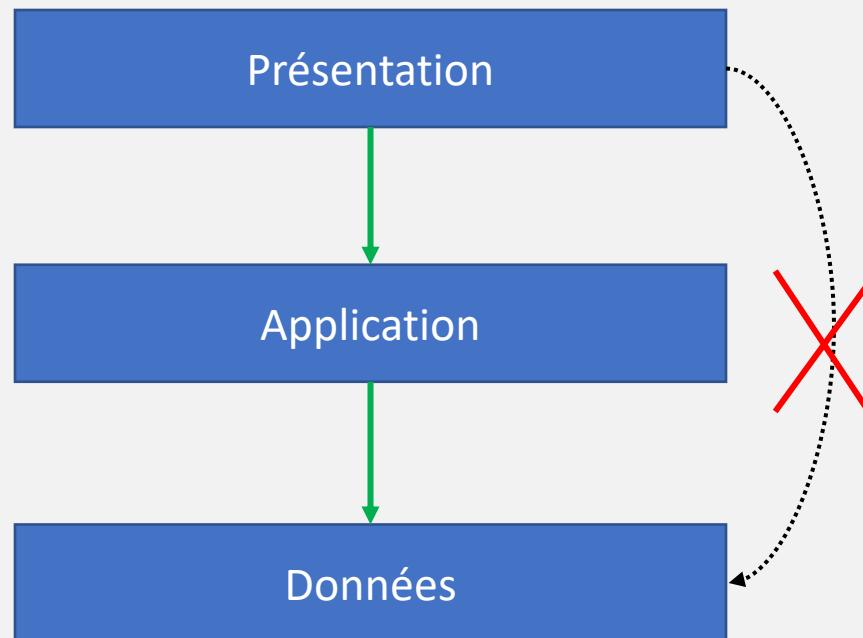
Architecture n-tiers

- Une architecture n-tiers est une architecture logicielle qui est organisée en n tiers empilés les uns sur les autres
- Un tiers ne peut communiquer qu'avec le tiers suivant



Architecture 3-tiers

- L'architecture la plus classique est celle en 3-tiers :
 - Tiers présentation : affiche l'information
 - Tiers traitement métier / application : effectue les traitements
 - Tiers données : accès et persistance des données



Tiers vs couche

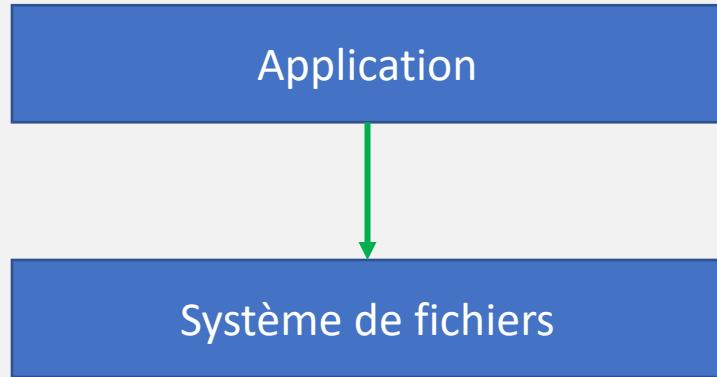
- Dans le vocabulaire commun et dans beaucoup de publications, les notions de tiers et de couche sont confondues
- Les notions généralement admises sont :
 - Un tiers est une structuration physique au niveau infrastructure
 - Une couche (layer) est une structuration logique d'une solution logicielle
 - Plusieurs couches peuvent être déployées sur un même tiers
 - Plusieurs tiers peuvent être déployés sur un même équipement physique ou virtuel
- Dans le cadre du cours :
 - Tiers : assimilable à un processus de système d'exploitation. Il est composé d'une ou plusieurs couches
 - Couche : assimilable à un composant. En .Net, on aura un projet Visual Studio par couche

Exemple – Tiers / couche



Exemple 1 – Logiciel classique

Rôles



Application :

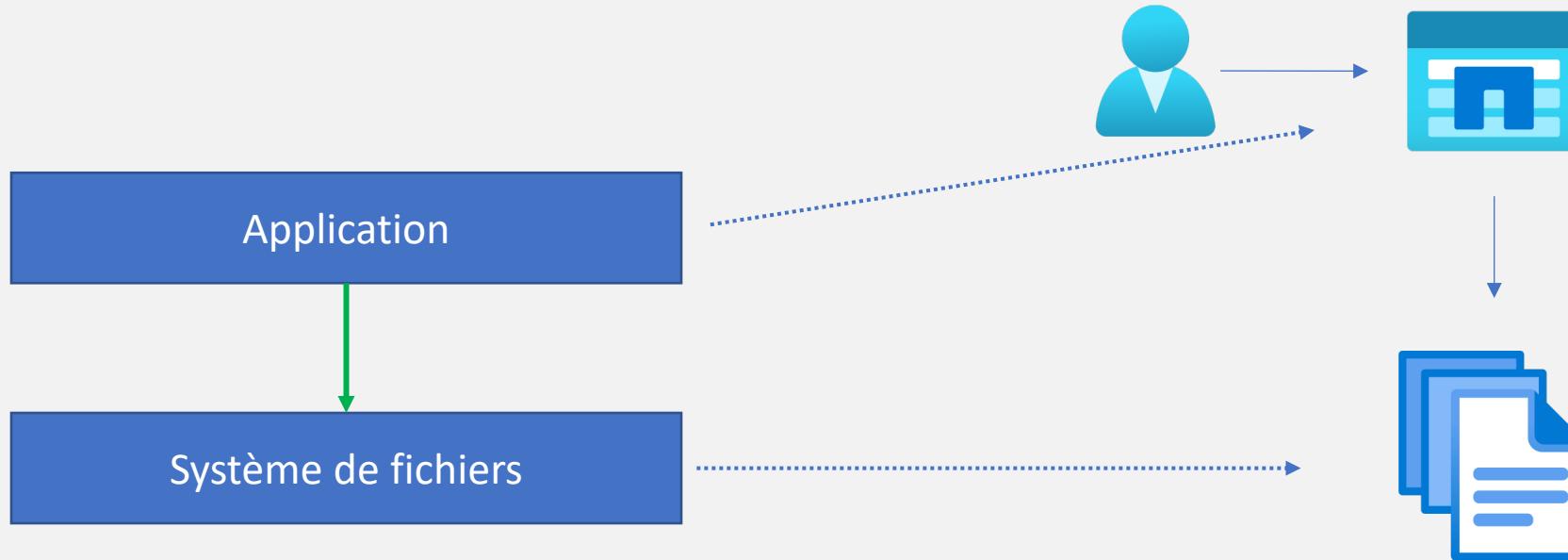
- affiche l'interface utilisateur
- transige avec le système de fichiers pour stocker et lire des documents

Système de fichiers :

- renseigne sur les fichiers existants
- stocke / organise des fichiers

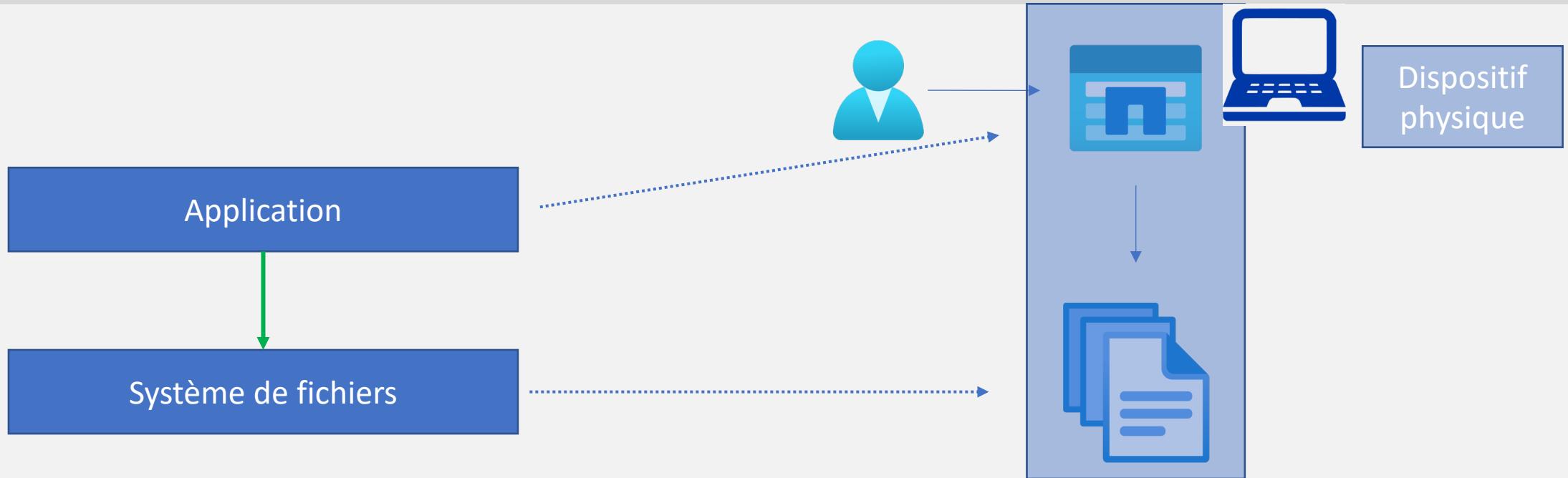
Exemple 1 – Logiciel classique

Services



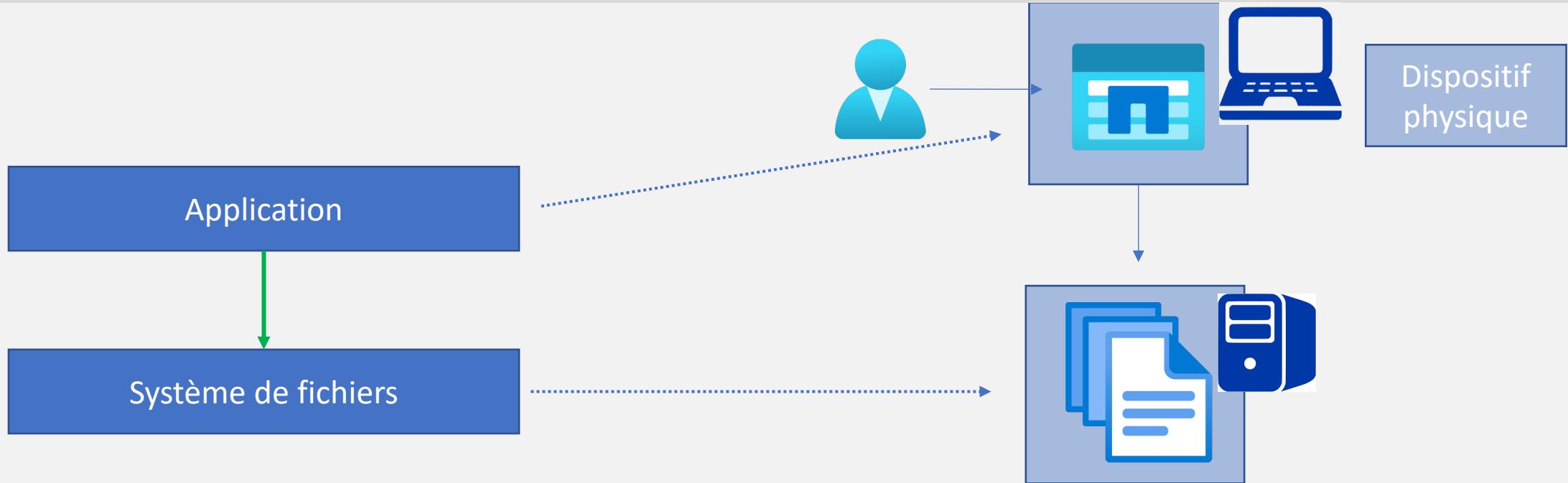
Exemple 1 – Logiciel classique

Déploiement 1



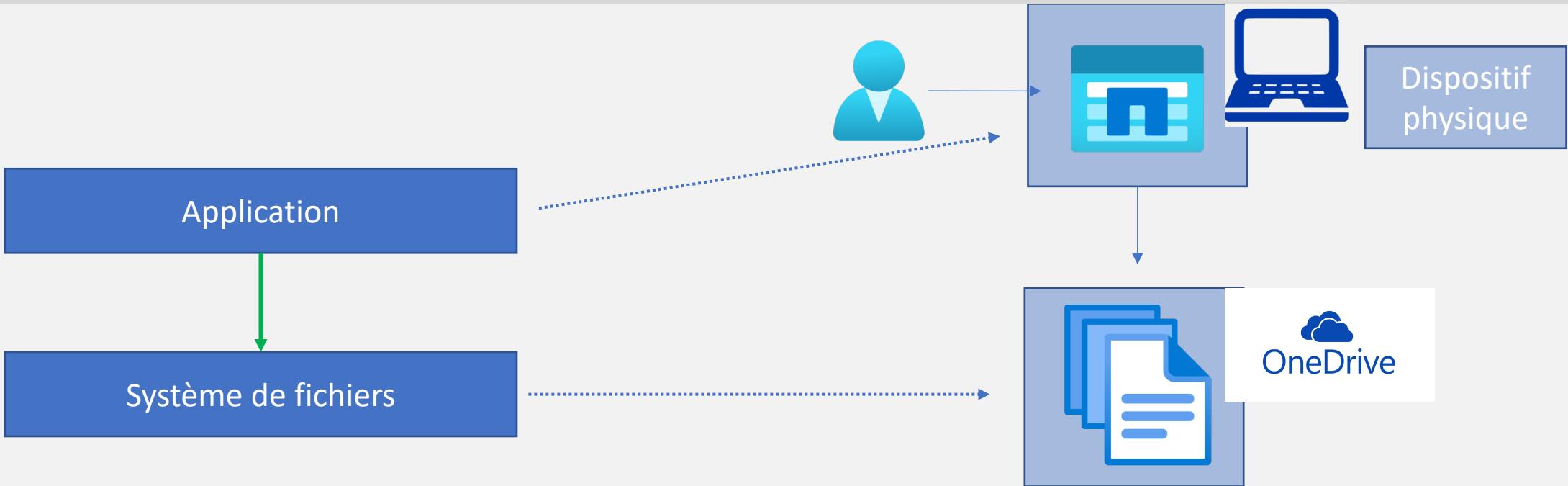
Exemple 1 – Logiciel classique

Déploiement 2



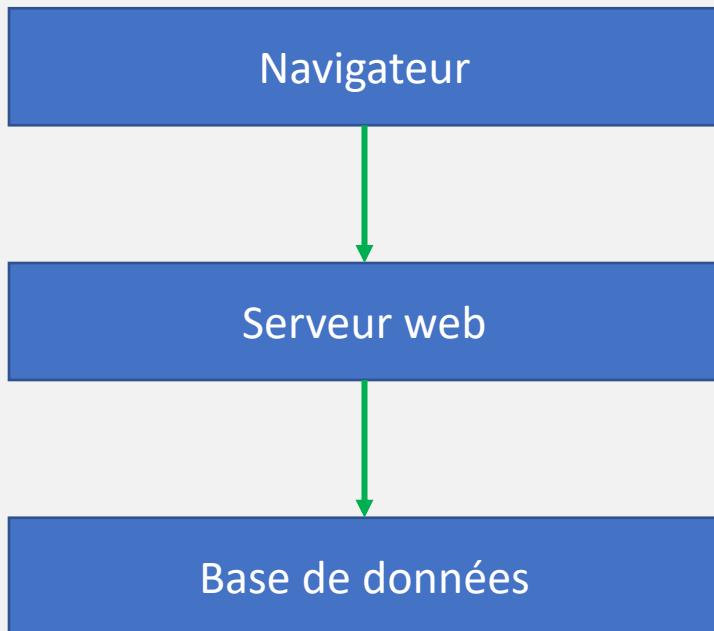
Exemple 1 – Logiciel classique

Déploiement 3



Exemple 2 – Site web (Client / serveur)

Roles



Navigateur :

- interprète les artefacts envoyés par le serveur applicatif
- transforme les actions de l'utilisateur en requêtes

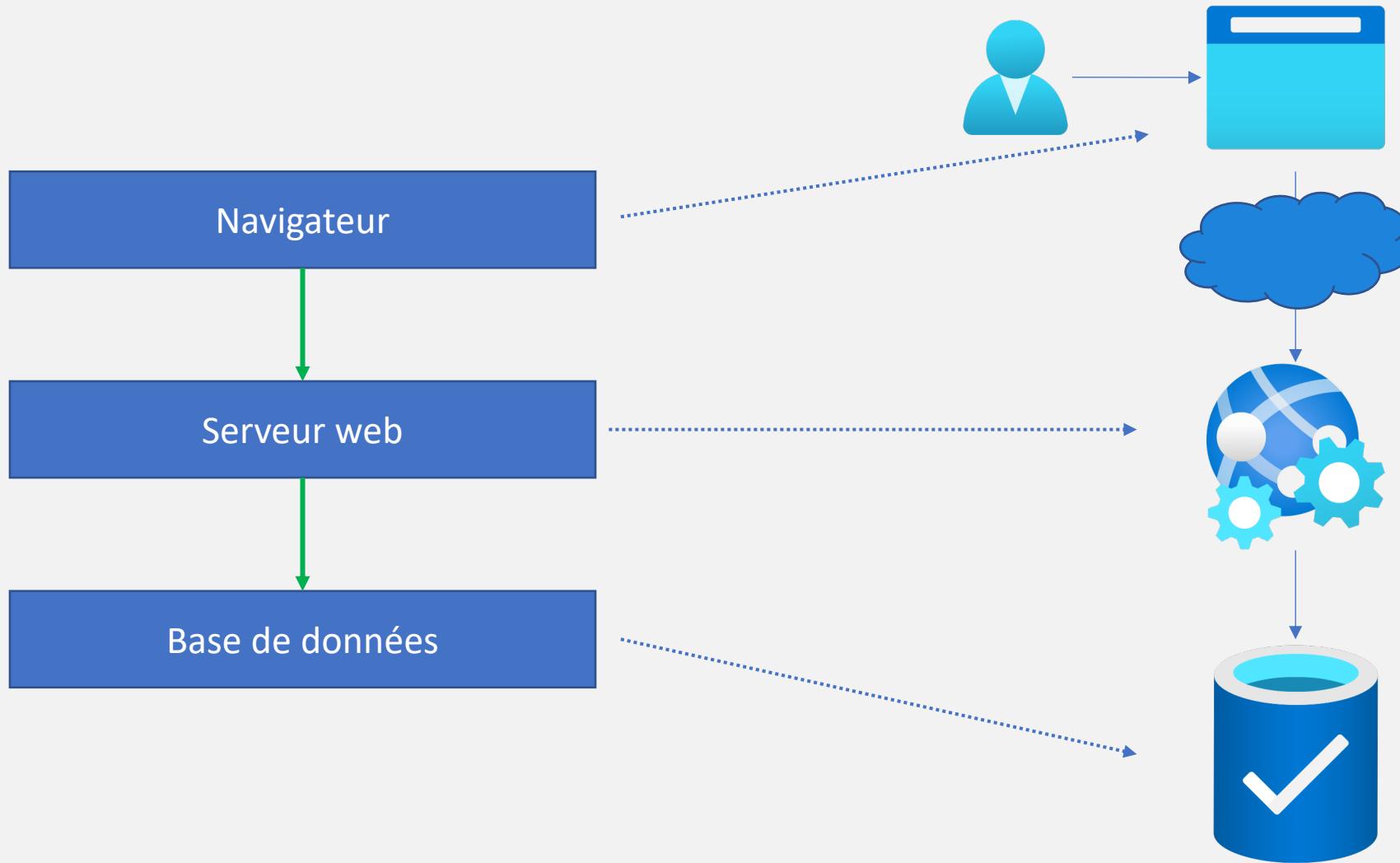
Serveur web :

- expose un service qui écoute des requêtes
- interprète les requêtes et si nécessaire exécute un traitement
- fait des requêtes SQL au serveur de base de données
- renvoie le résultat de la requête

Base de données :

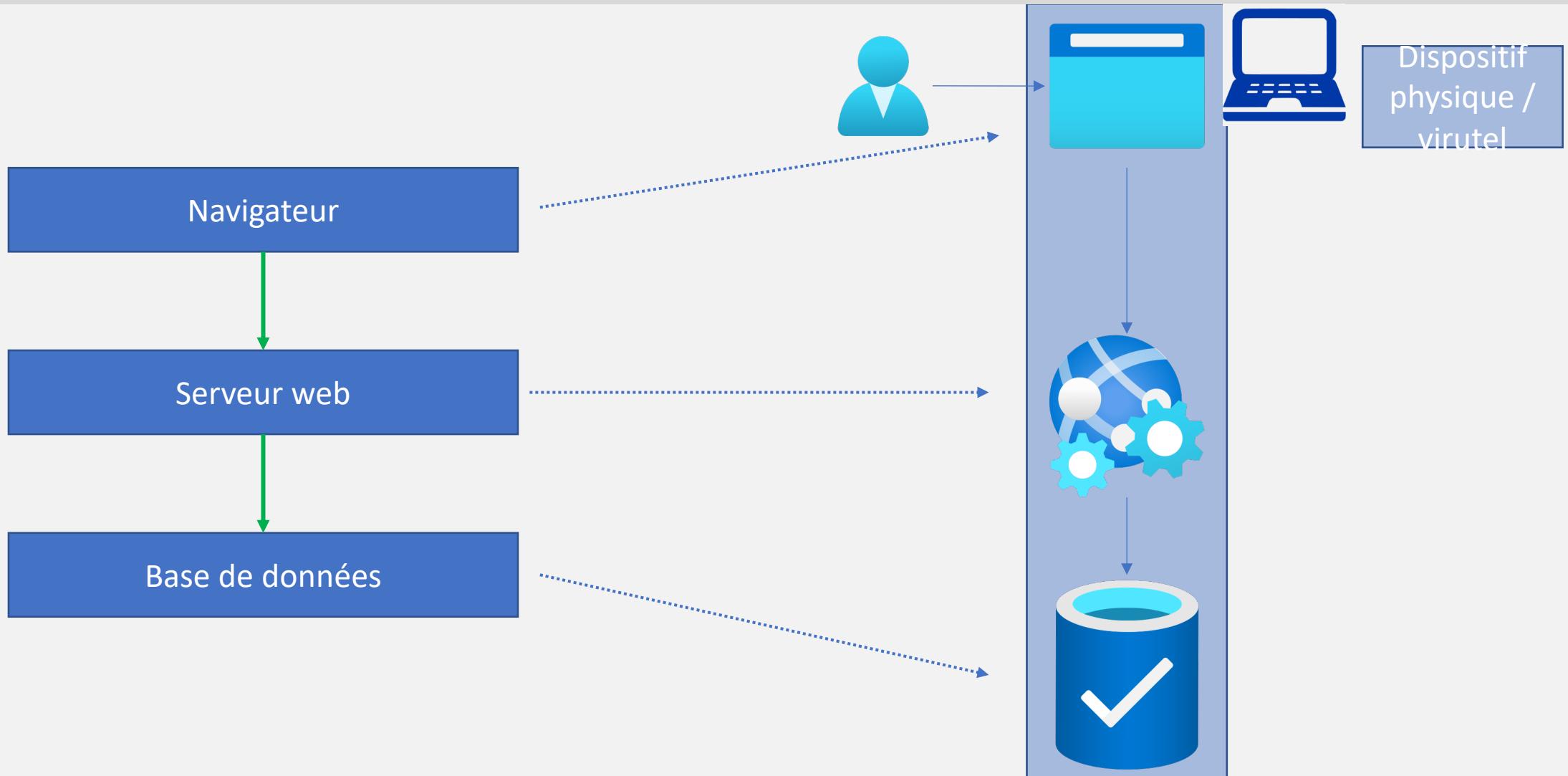
- reçoit des requêtes SQL et transmet les résultats
- stocke les données sur un support physique ou virtuel

Exemple 2 – Site web (Client / serveur) Services



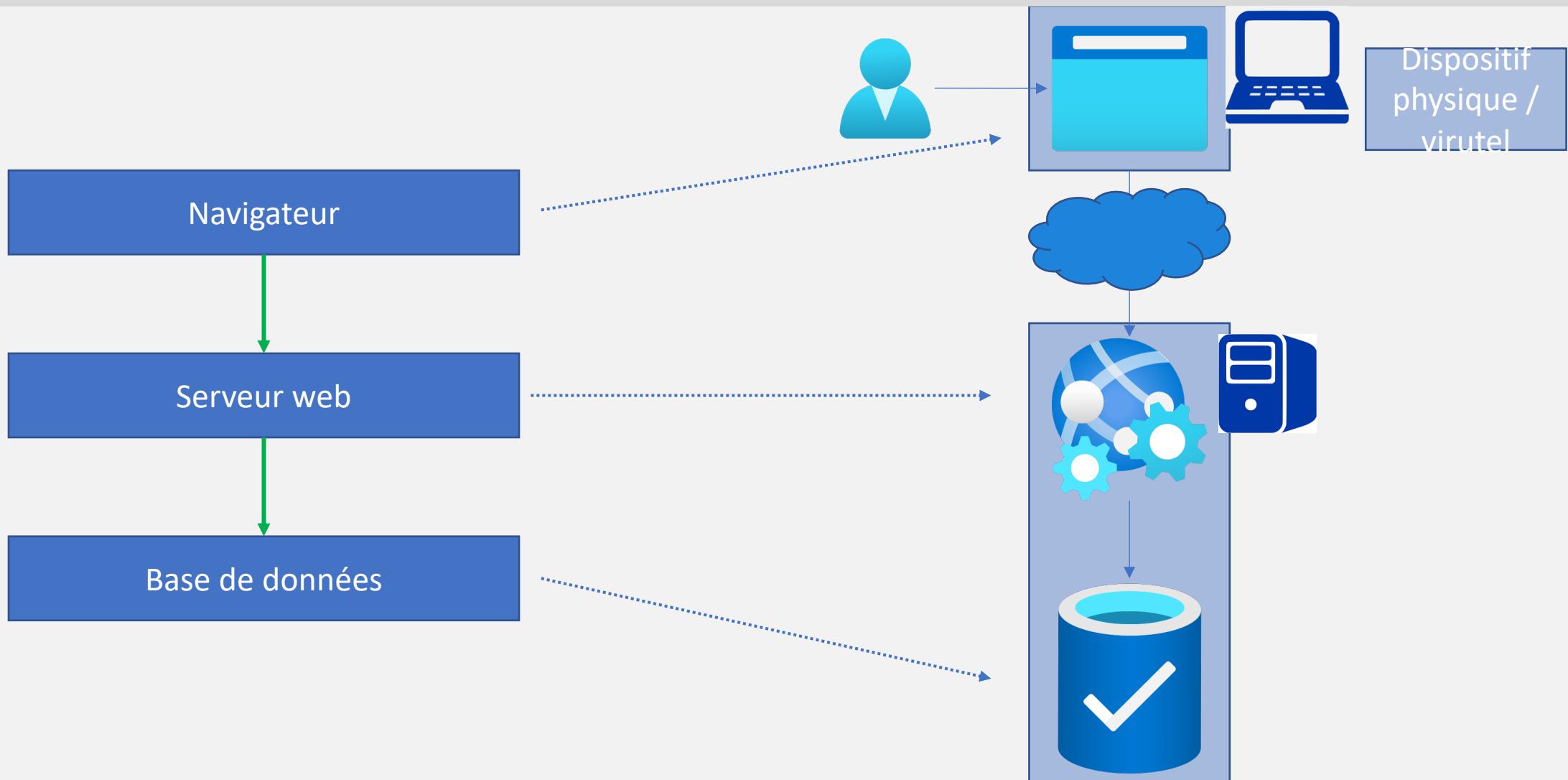
Exemple 2 – Site web (Client / serveur)

Déploiement 1



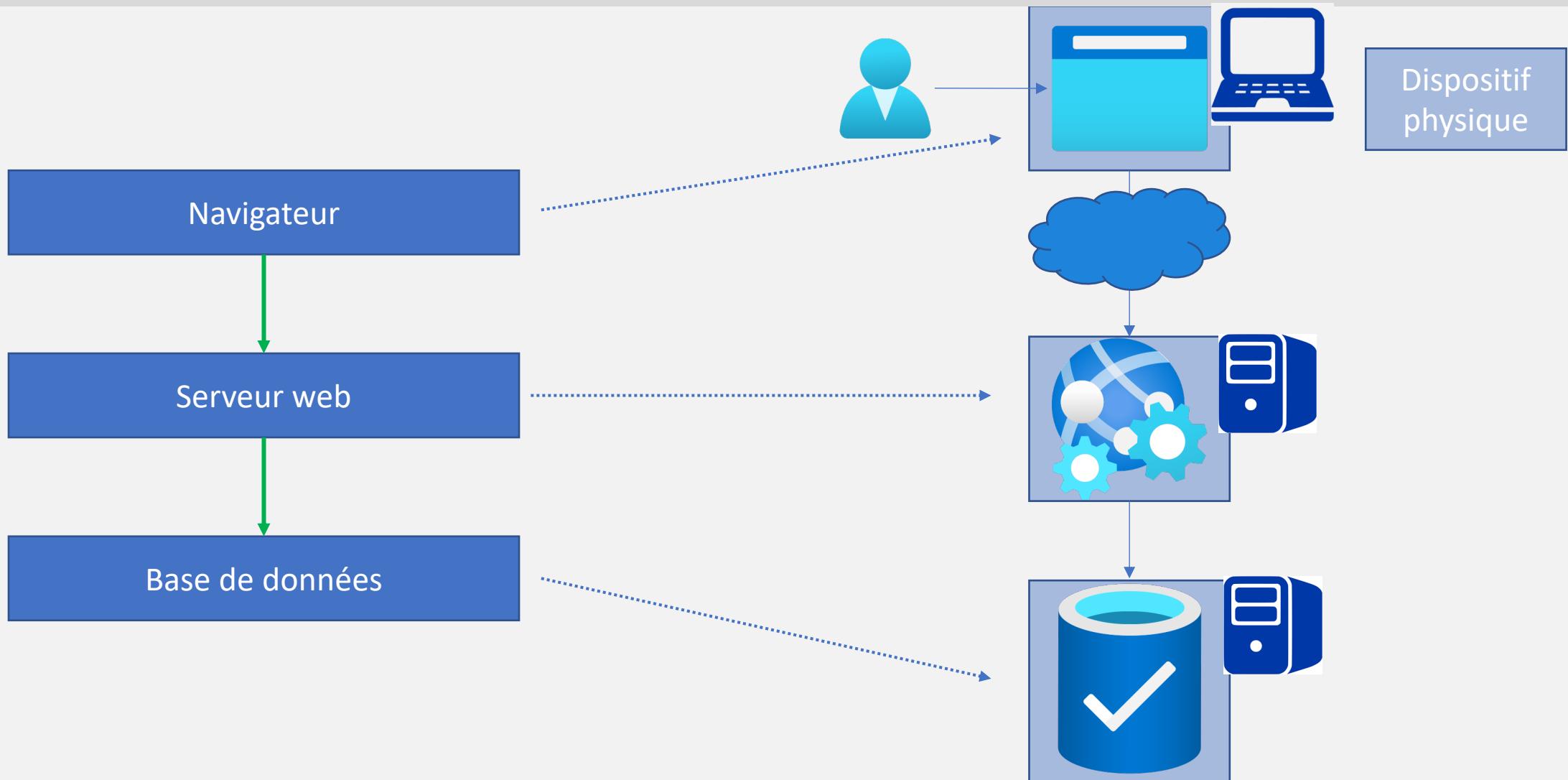
Exemple 2 – Site web (Client / serveur)

Déploiement 2



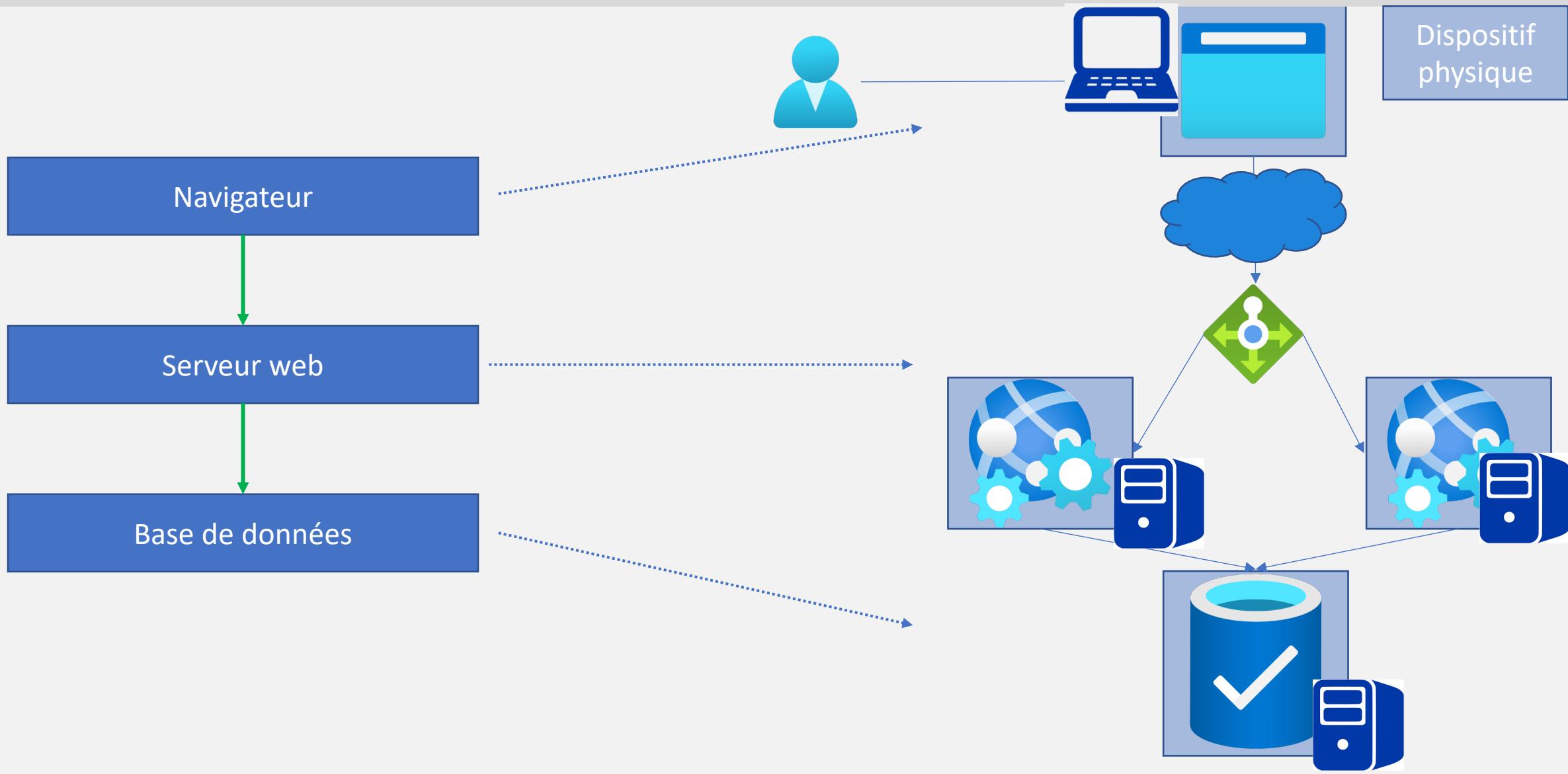
Exemple 2 – Site web (Client / serveur)

Déploiement 3



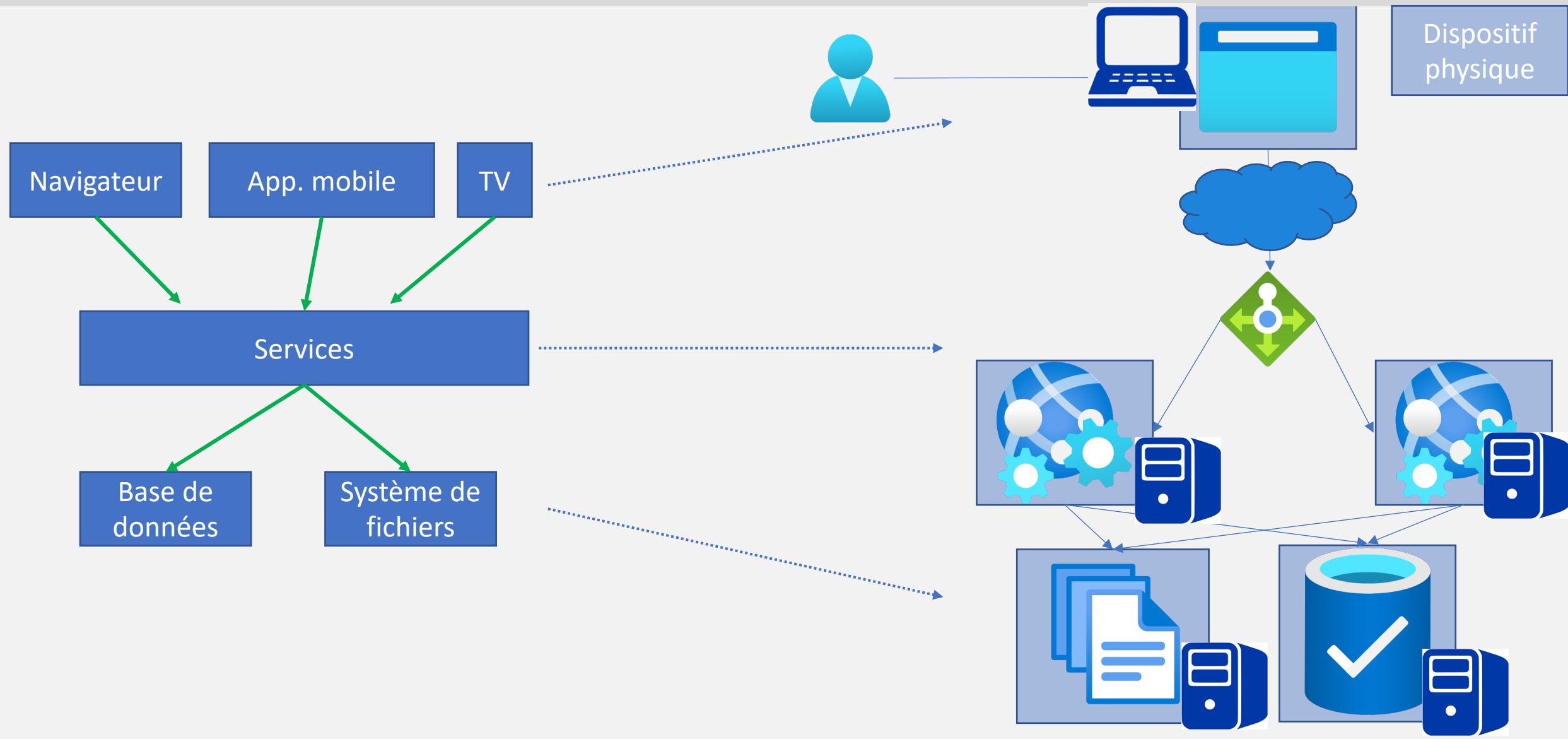
Exemple 2 – Site web (Client / serveur)

Déploiement 4



Exemple 3 – Applications avec services

Déploiement



Avantages

- Déploiement « indépendant » des tiers et des couches
- Mise à l'échelle possible suivant les communications en œuvre entre les tiers
- Avec plusieurs couches et des efforts modérés :
 - Possibilité de déplacer des couches d'un tiers à un autre
 - Être moins sensible aux changements technologiques