



# HEARTHSTONE

## 2

CANARIA Kaira - FAREZ Lylia - MAHOUCHE Tania - ROLLET Gabrielle



# SOMMAIRE

1. Présentation du jeu
2. Chronologie des réalisations
3. Architecture et conception
4. Choix logiciels
5. Stratégies marketing
6. Conclusion

# Présentation du jeu



# Développer un hearthstone battleground like



## Mécaniques :

- A = Or
- B = Pouvoirs héroïques
- C = Composition
- D = Taverne de Bob
- E = Améliorer la taverne
- F = Rafraîchir
- G = Geler

**1**



**1**

**2**



**2**

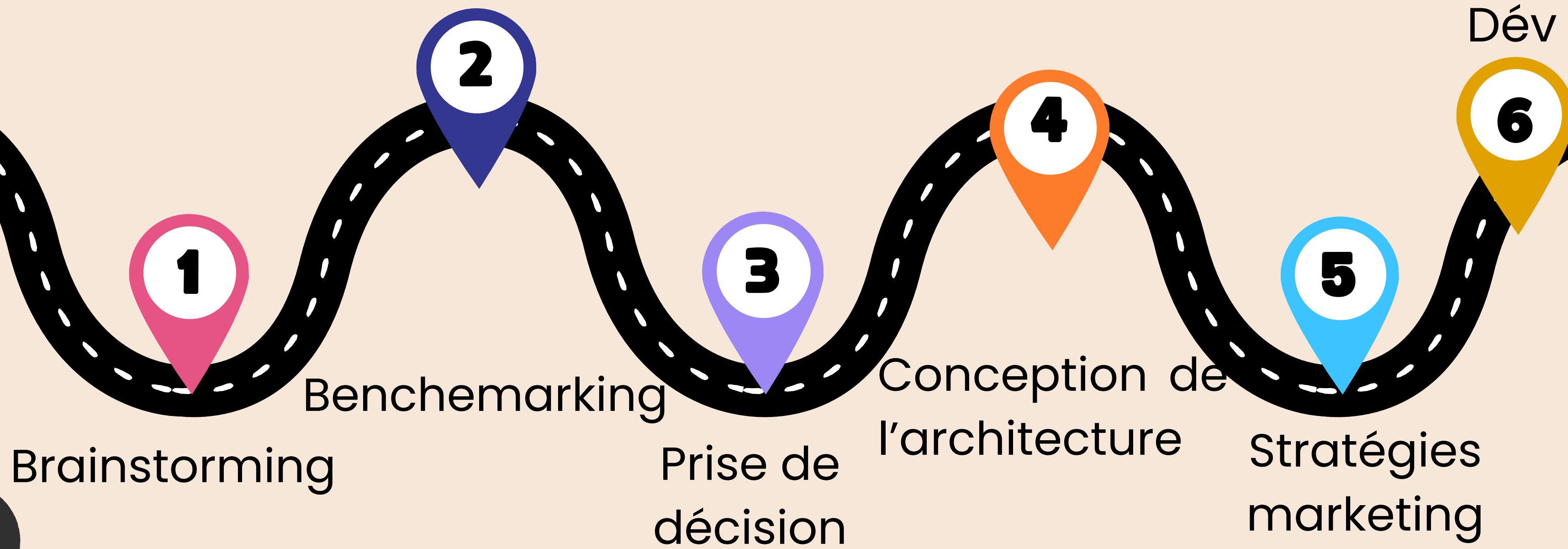
**3**



**3**

# **Chronologie des réalisations**





# **Architecture & Conception**

**Monolithique**

**Microservices**

## Monolithique

VS

## Microservices



### Avantages

- Simplicité
- Gestion des problèmes transversaux
- Performances



### Inconvénients

- Fiabilité/disponibilité
- Maintenance/Mise à jour
- Pile technologique



### Avantages

- Maintenance
- Déploiements indépendants
- Résilience

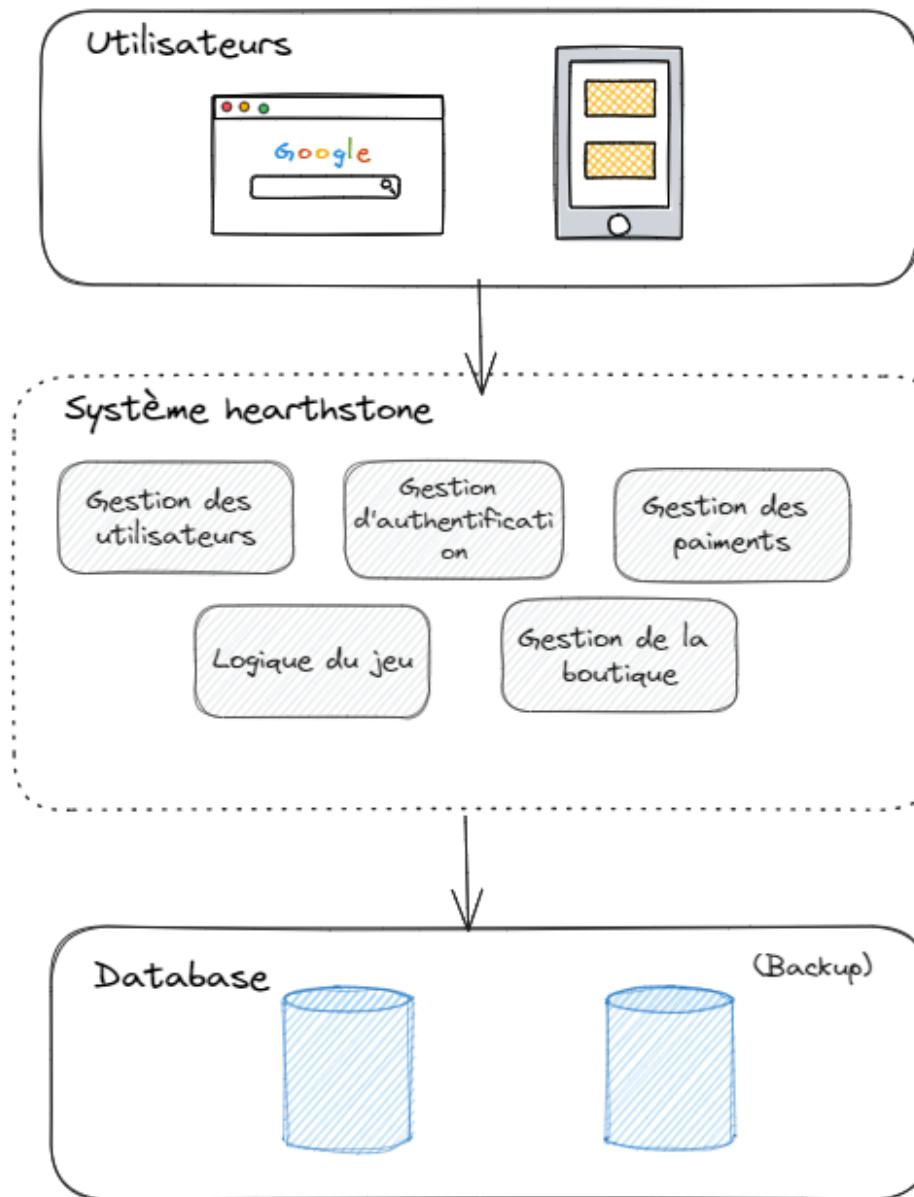


### Inconvénients

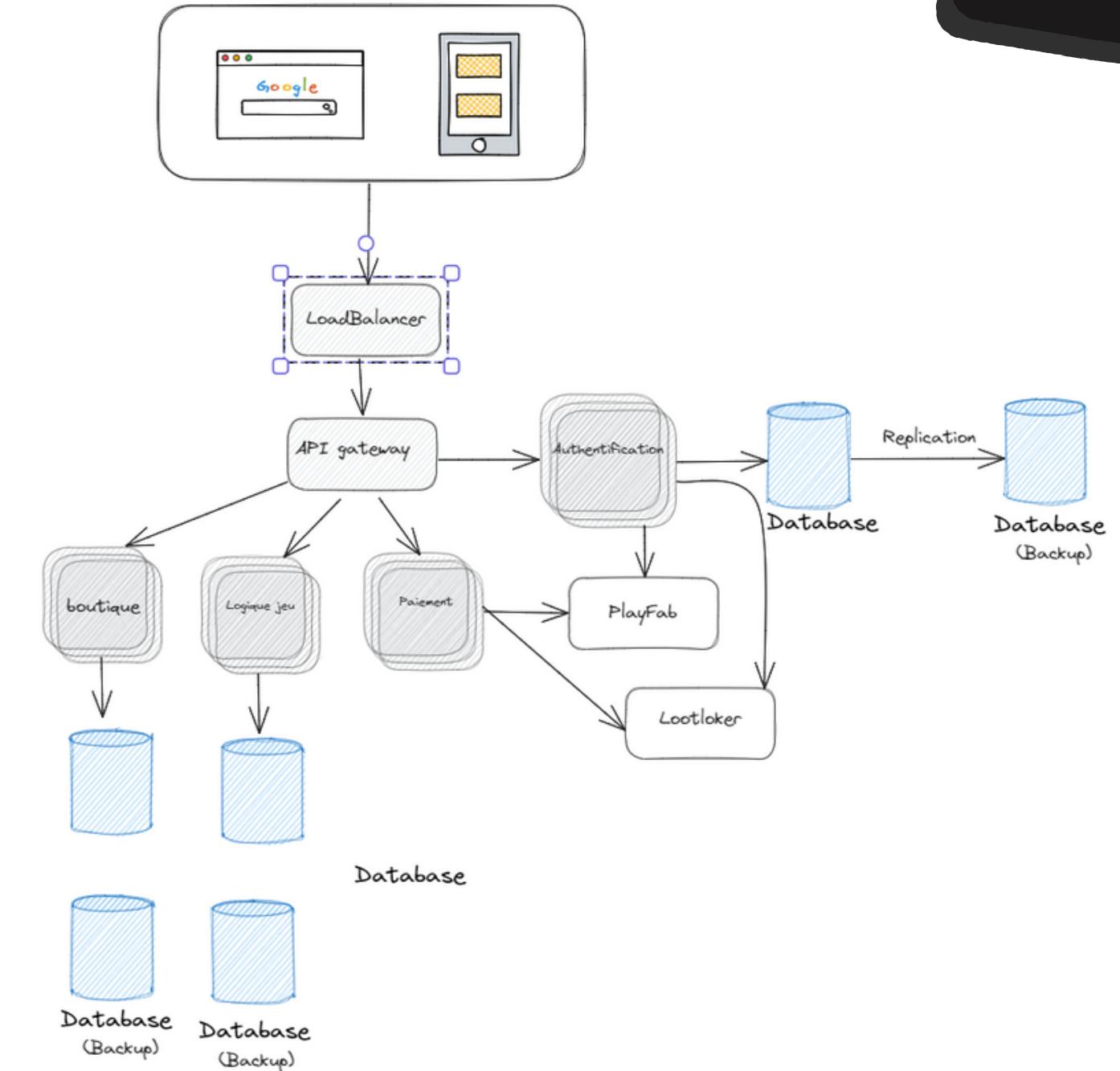
- Expertise
- Surcharge de performances potentielle
- Défis de la gestion des données

# ARCHITECTURE

## Architecture monolithique



## Architecture microservices

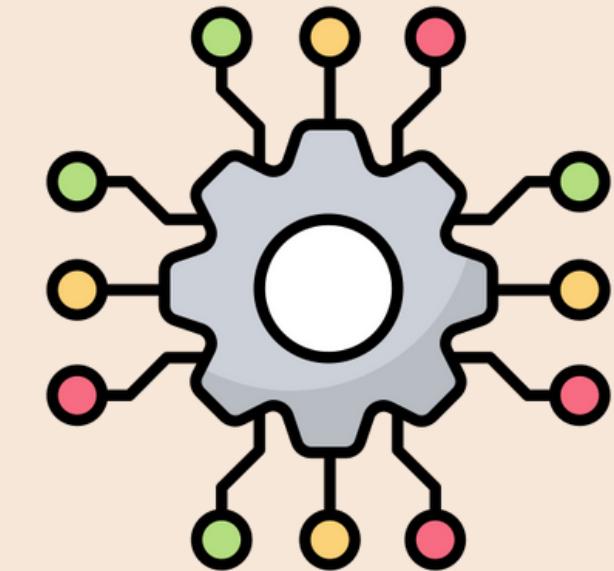




# **Microservices**

# Architecture utilisée: Microservices

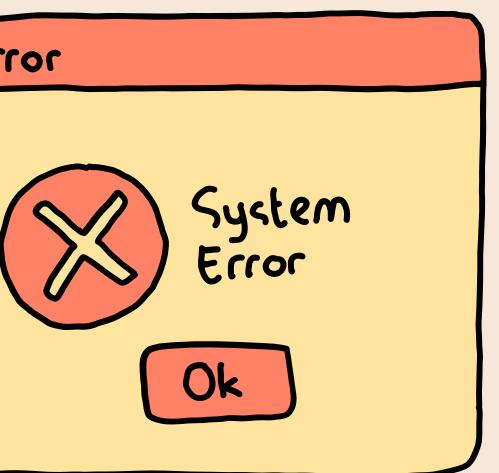
Pourquoi ce choix?



Maintenance



Scalabilité



Résilience

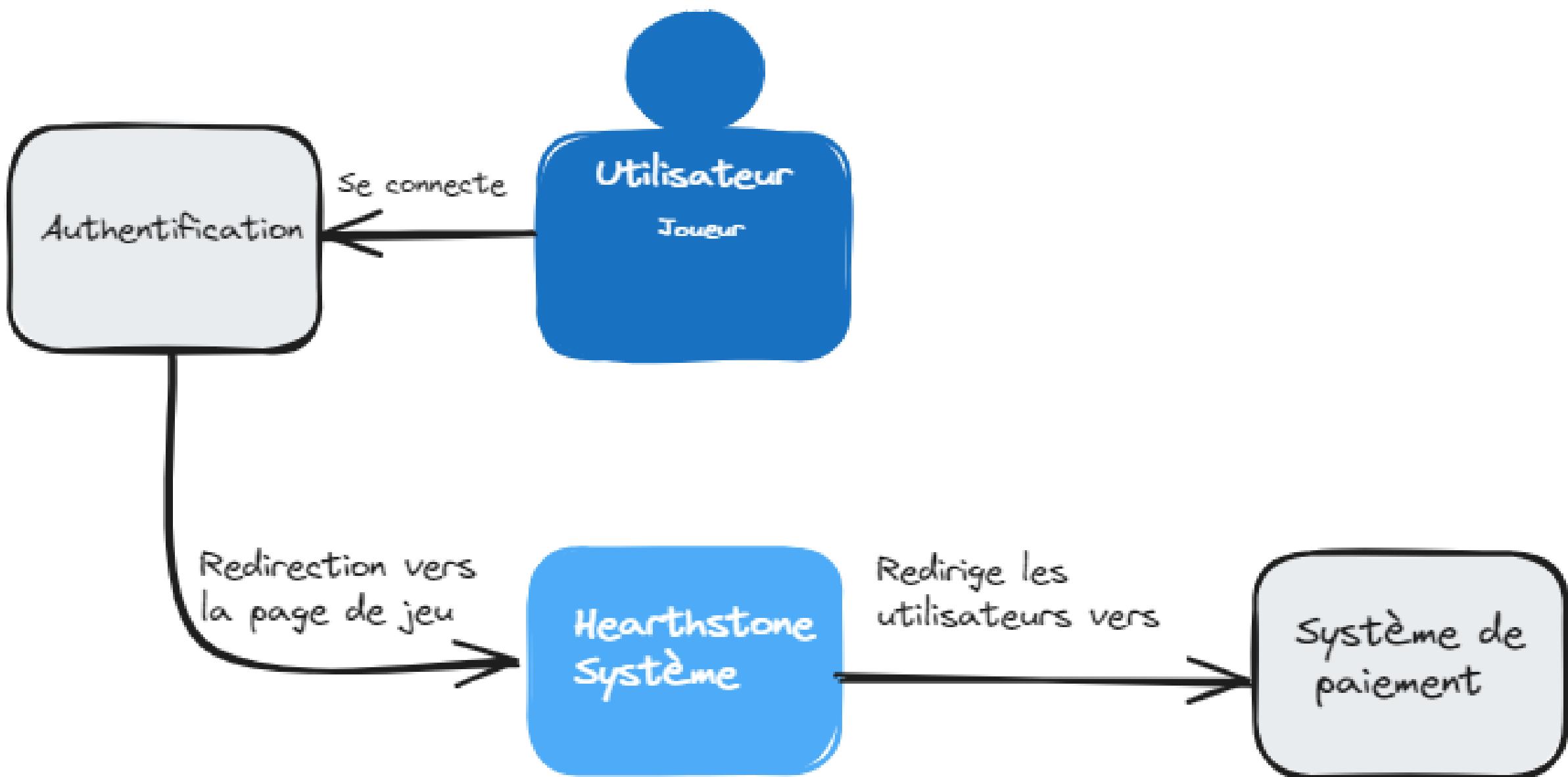


Réutilisation

# CONCEPTION

## Model C4

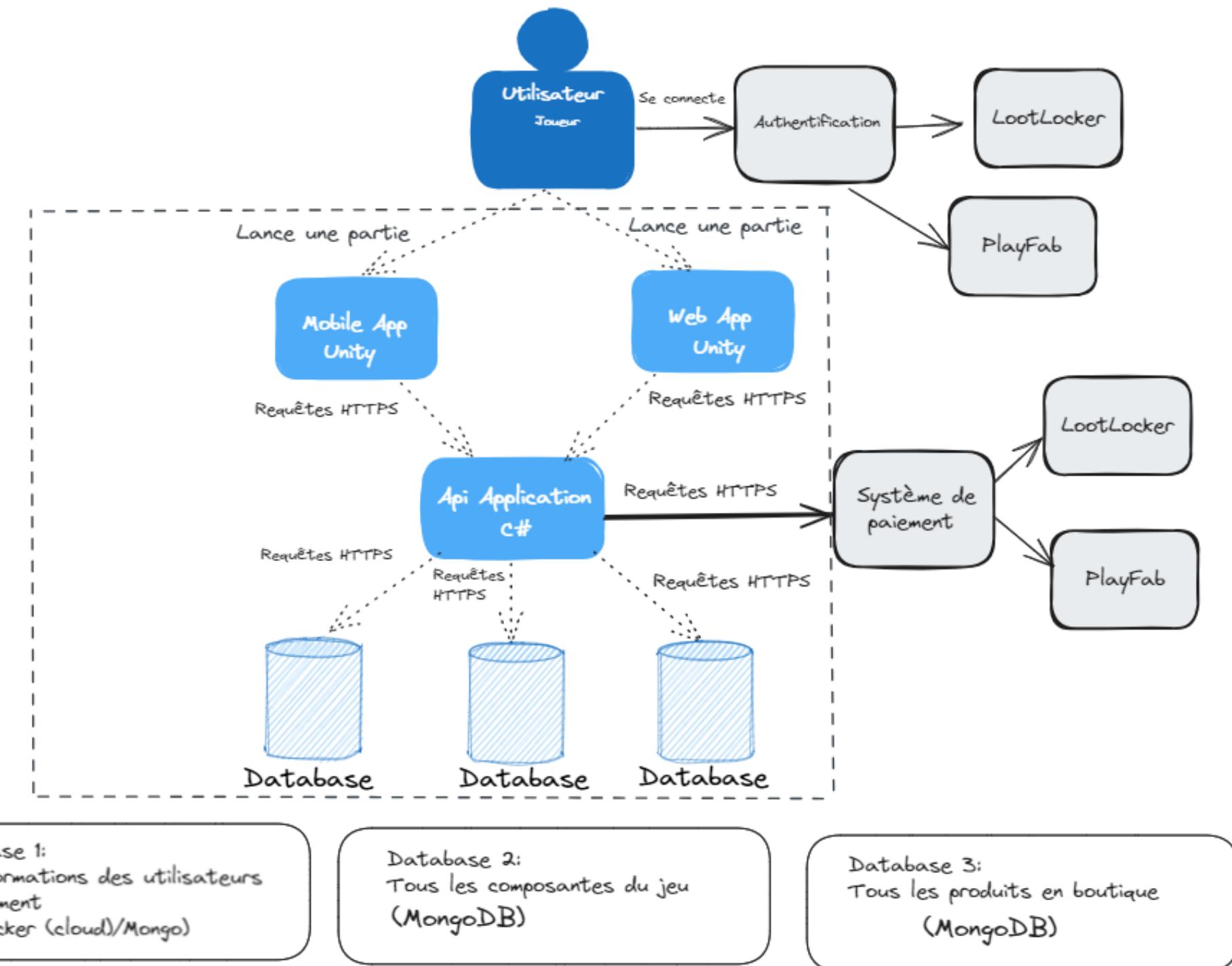
Niveau 1 : Contexte



# CONCEPTION

## Model C4

Niveau 2 : Container

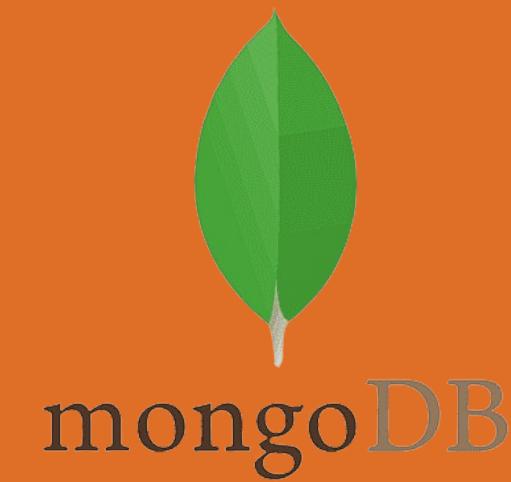


# Choix Logiciels



# **Systeme de Gestion de Base de données et services**





- Évolutivité horizontale
- Performance
- Réplication et résilience

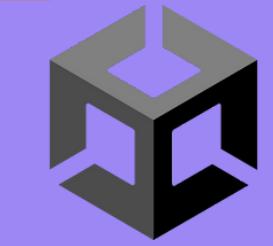


- Facilité d'intégration avec Unity
- Gestion de contenu centralisée
- Conformité au RGPD





Unreal Engine



Unity



## Avantages

- Graphisme de Haute Qualité
- Blueprints visuels
- Large éventail de fonctionnalités



## Inconvénients

- Taille de fichier d'execution
- Plus orienté graphisme
- Configuration matérielle requise



## Avantages

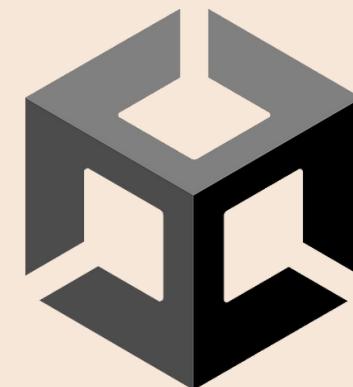
- Facilité d'utilisation
- Prix et Modèle Économique
- Rapidité de Prototypage



## Inconvénients

- Personnalisation Limitée du Moteur
- Optimisation et Performances
- Langage de Scripting Unique

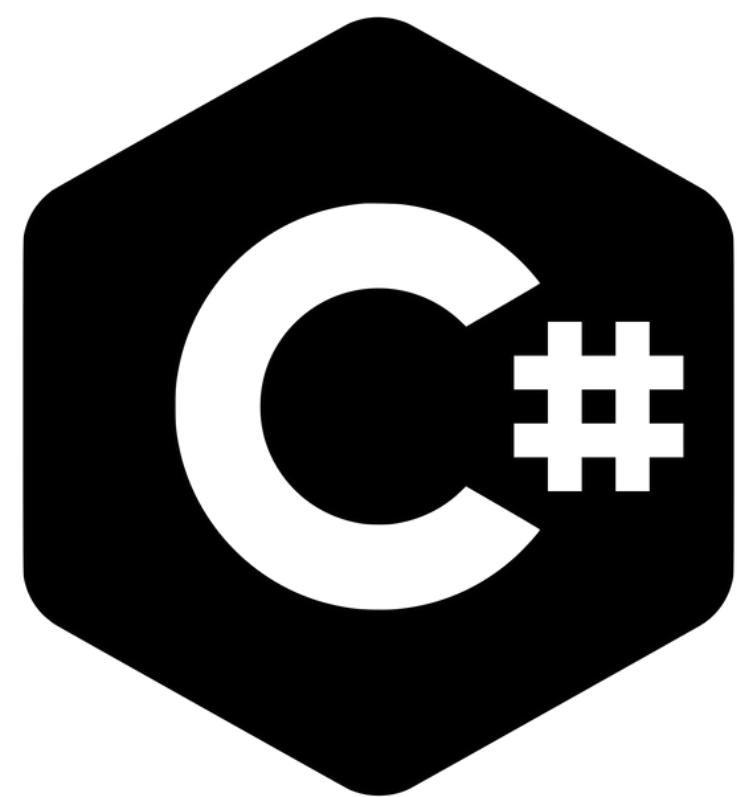
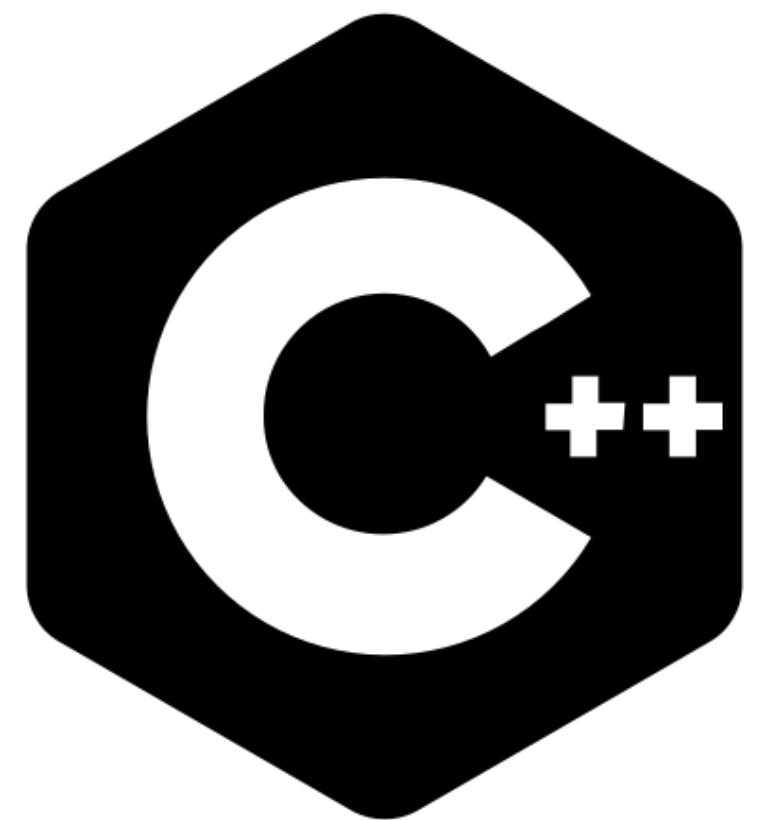




# Langage utilisé: Unity

## Pourquoi ce choix?

- Facilité d'utilisation
- Multiplateforme
- Vaste Ecosystème
- Performance
- Coût





## Avantages

- Contrôle de la memoire
- Performance
- Grande bibliothèque de Ressources

## Inconvénients

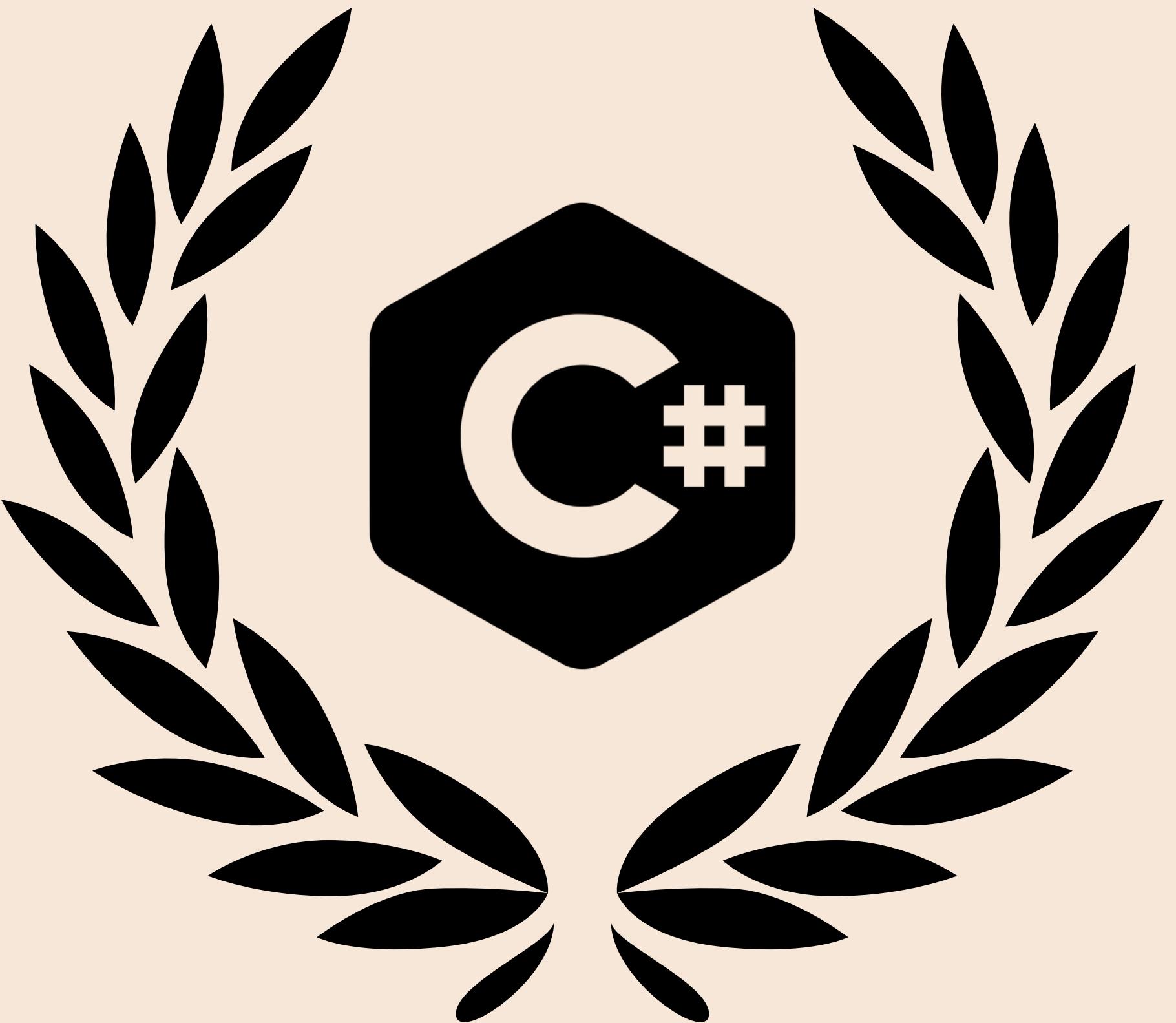
- Garbage collection
- Compatibilité et portabilité

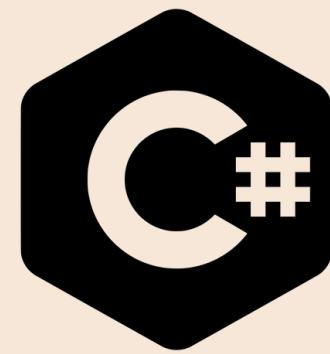
## Avantages

- Facilité d'utilisation
- Performance
- Productivité et rapidité développement

## Inconvénients

- Contrôle sur la memoire
- Dépendance à un écosystème particulier

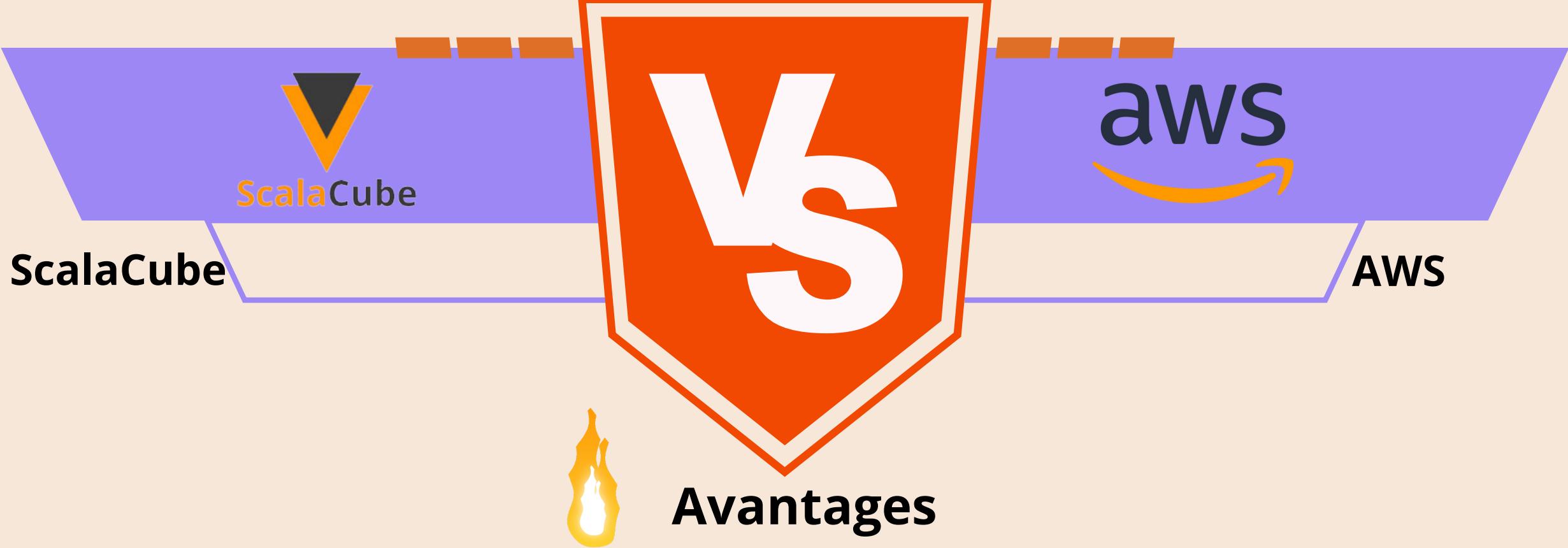




# Langage utilisé: C#

## Pourquoi ce choix?

- Facilité d'utilisation
- Langage principalement utilisé Unity
- Ressources et documentations riche
- Scripting et programmation
- Multiplateforme



- Spécialisation dans le domaine des jeux.
- Forfaits plus économiques.

- Flexibilité et personnalisation.

## Inconvénients

- Limitations de personnalisation avancée.

- Coût.
- Complexité et configuration.



# Stratégies marketing



Boutique en  
ligne



Publicité



Abonnement



# Conclusion



# Merci pour votre attention



# Resources



- <https://www.lemagit.fr/conseil/Architecture-monolithique-vs-microservices-avantages-et-inconvénients>
- aspect sonore: <https://www.linkedin.com/advice/0/how-do-you-use-sound-enhance-your-games-skills-gaming-industry>
- lootlocker: <https://docs.lootlocker.com/players/authentication>

