**Architecture Decision Record (ADR)**

**Contexte**

Nous développons un jeu de cartes similaire à Hearthstone, nécessitant la conception d'une architecture pour gérer les cartes, les joueurs, le plateau de jeu et les interactions entre ces éléments.

**Décision**

Nous avons opté pour une architecture en microservices pour ce projet, fragmentant les fonctionnalités du jeu en différents services indépendants.

**Justification**

1. **Scalabilité et portée mondiale :** En envisageant une commercialisation à l'échelle mondiale, une architecture monolithique pourrait devenir contraignante en termes de montée en charge et d'évolutivité. Les microservices offrent une solution pour répondre à cette évolutivité future.
2. **Isolation des responsabilités :** La séparation des fonctionnalités en services distincts permet une meilleure isolation des responsabilités, facilitant la maintenance et le développement parallèle.
3. **Flexibilité de déploiement :** Bien que plus complexe que le déploiement monolithique, l'architecture en microservices offre une flexibilité de déploiement, permettant la mise à jour indépendante de chaque service sans perturber l'ensemble du jeu.

**Conséquences**

1. **Complexité opérationnelle accrue :** La gestion de multiples services peut entraîner une complexité opérationnelle supplémentaire, nécessitant une surveillance et une gestion plus approfondies des services.
2. **Communication inter-service :** L'interaction entre les différents services nécessitera des protocoles de communication robustes pour garantir la cohérence et la fiabilité du jeu.
3. **Coût initial et temps de développement :** La mise en place d'une architecture en microservices impliquera un investissement initial plus important en termes de temps et de coûts de développement.