

ARCHITECTURE SOA

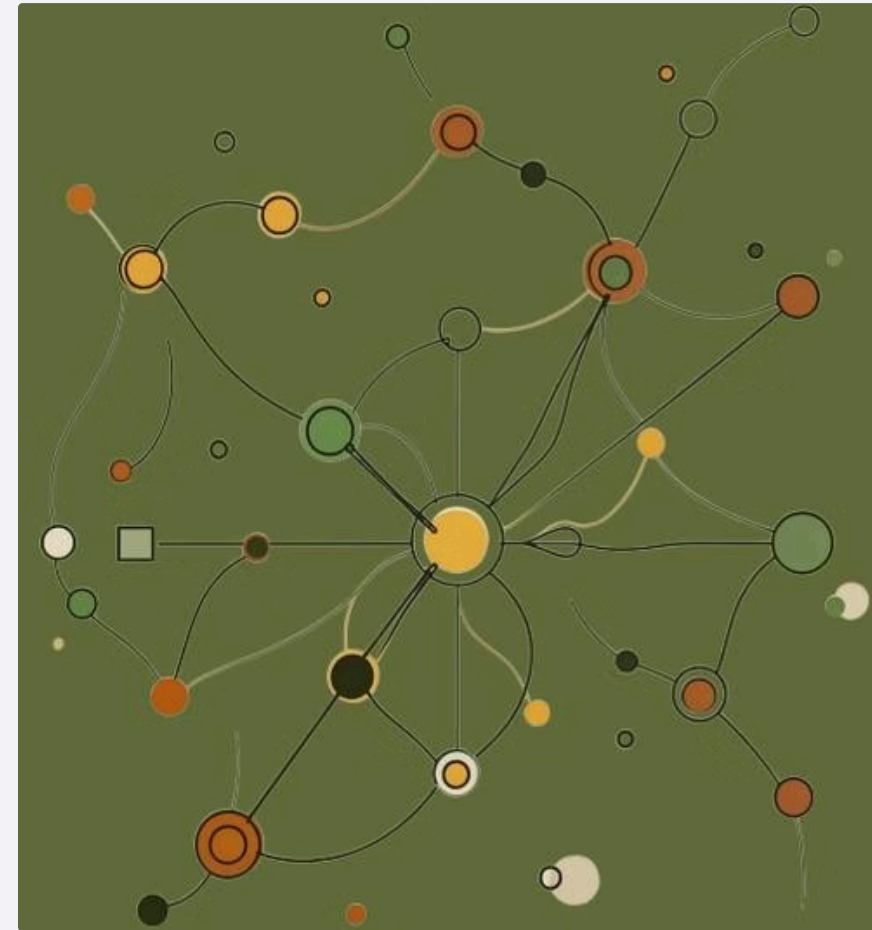
SYSTÈMES DISTRIBUÉS

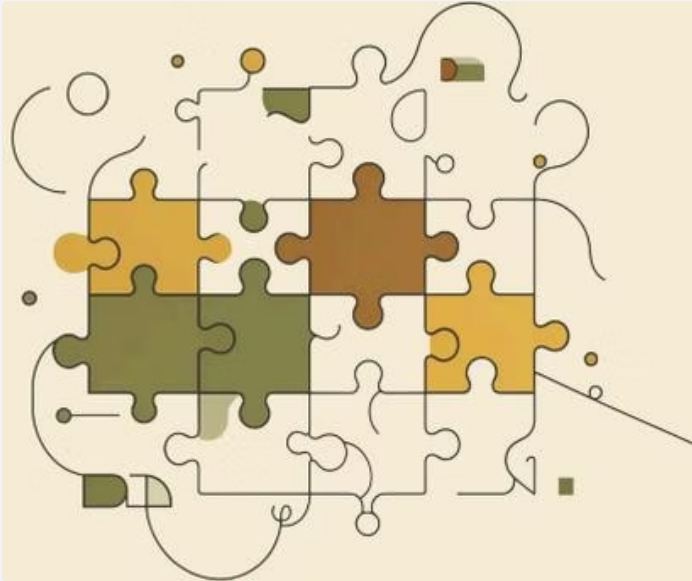
Composition des Web Services

Assembler des services indépendants pour construire des applications distribuées **puissantes et évolutives**.

Présenté par :

Chafai Nadhir • Mebdoua Hareth Anis





DÉFINITION

Qu'est-ce que la composition ?

La **composition des web services** est le processus d'assemblage et d'orchestration de plusieurs services web afin de réaliser une fonctionnalité globale cohérente.

Chaque service joue un rôle précis dans la chaîne. Leur collaboration produit un résultat qu'aucun service ne pourrait atteindre isolément.

- ① Portée par l'essor des architectures SOA, cette approche est au cœur des systèmes distribués modernes.

Types de Composition

Composition Statique

Les services et leurs interactions sont **définis à l'avance**, lors de la conception.

- Flux d'exécution fixe et prévisible
- Facile à tester et à valider
- Idéale pour des processus stables

Ex : processus de paiement dans une application e-commerce.

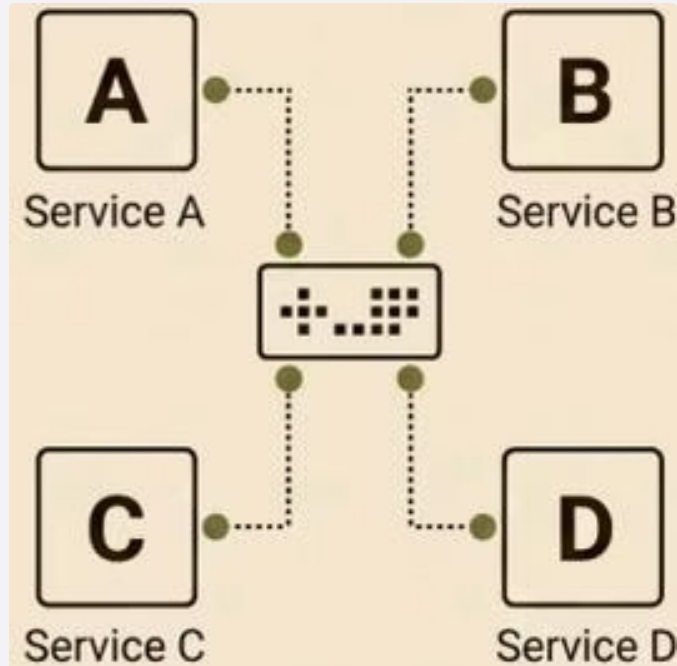
Composition Dynamique

Les services sont **sélectionnés à l'exécution** selon le contexte et les besoins.

- Hautement flexible et adaptable
- Convient aux environnements changeants
- Utilisée dans les systèmes intelligents

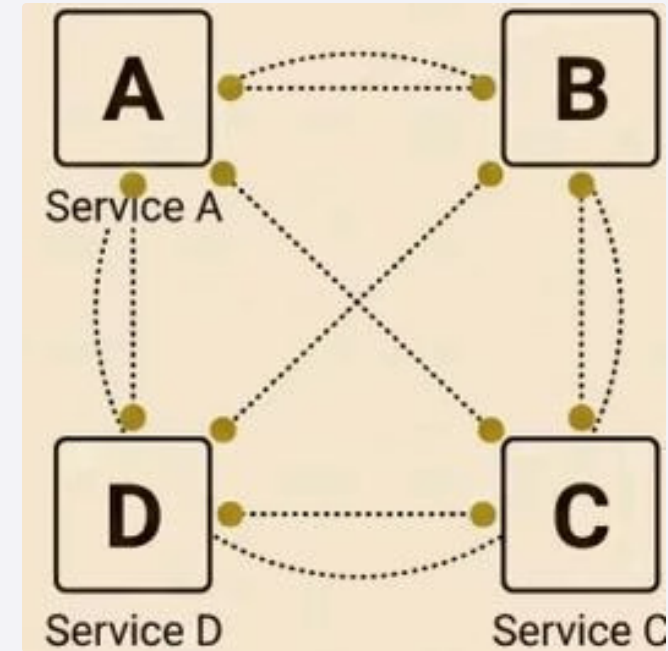
Ex : sélection du meilleur service de livraison en temps réel.

Orchestration vs Chorégraphie



Orchestration

Un **service central** (orchestrateur) contrôle l'ensemble des participants et définit l'ordre d'exécution. Analogie : un chef d'orchestre dirigeant ses musiciens.



Chorégraphie

Aucun contrôleur central. Chaque service **connaît son rôle** et interagit directement avec ses pairs. Plus distribuée et plus résiliente aux pannes centralisées.

Technologies Clés



SOAP

Protocole basé sur **XML** pour l'échange de messages structurés et fortement typés entre services.



WSDL

Langage de **description** des services web : opérations exposées, types de données et points d'accès.



REST

Basé sur **HTTP**, léger, sans état. L'approche dominante dans les APIs modernes et les microservices.



BPEL

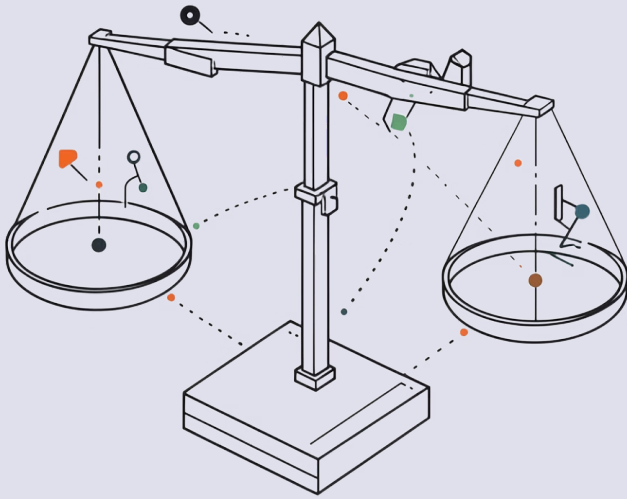
Langage XML dédié à la **définition de l'orchestration** : séquence, conditions, boucles et gestion d'erreurs.

Étapes de la Composition



Chaque étape est critique : une identification imprécise ou une mauvaise gestion des erreurs peut compromettre l'ensemble de la composition et dégrader l'expérience utilisateur.

Avantages & Inconvénients



✓ Avantages

- **Réutilisation** des services existants sans redéveloppement
- **Modularité** et flexibilité architecturale
- Réduction du **temps de développement**
- **Adaptabilité** rapide aux changements métiers

⚠ Inconvénients

- **Complexité** de gestion et de débogage
- Risques de **sécurité** accrus entre services
- **Dépendances** pouvant fragiliser l'ensemble
- **Latence réseau** cumulée entre appels



CAS PRATIQUE

Réservation en Ligne

Trois services indépendants composés pour offrir une expérience complète et fluide à l'utilisateur final.

Service de Recherche

Interroge les disponibilités en temps réel.

Service de Paiement

Sécurise et valide la transaction financière.

Service de Notification

Envoie la confirmation par e-mail ou SMS.

Bonnes Pratiques



Sécuriser les échanges

Utiliser HTTPS, OAuth 2.0 et des tokens JWT pour protéger chaque appel inter-services.



Monitorer en continu

Mettre en place des outils d'observabilité (logs, métriques, traces) pour détecter les anomalies rapidement.



Gérer les défaillances

Implémenter les patterns **Circuit Breaker** et **Retry** pour une résilience accrue face aux pannes.



Versionner les contrats

Versionner les APIs et documenter les interfaces (WSDL, OpenAPI) pour éviter les régressions.

Conclusion

La composition des web services est un **pilier de l'architecture logicielle moderne**. Elle permet de construire des systèmes distribués flexibles, évolutifs et maintenables en assemblant des briques de valeur réutilisables.

Orchestration

Contrôle centralisé avec BPEL

Chorégraphie

Collaboration décentralisée

REST & SOAP

Protocoles complémentaires

