

Évolution des Architectures Web

Les trois couches d'une application:

×Présentation ×Traitement ×Données

Le découpage et la répartition de ces couches permettent de distinguer les architectures applicatives suivantes :

×L'architecture 1-tiers, ×l'architecture 2-tiers, ×l'architecture 3-tiers, ×les architectures n-tiers

Abstraction d'une application

- Chaque niveau peut être placée sur une machine différente
- Chaque niveau peut s'exécuter sur un système différent
- Chaque niveau peut être programmée dans un langage différent

→Toute application possède ces trois parties

1 ère génération- Architecture centralisée (1-tiers) Informatique centralisée (Mainframe)

Avantage:

- Assure la haute disponibilité et l'intégrité des données, et offre à l'entreprise un système cohérent et fiable.
- Coût d'administration faible

Inconvénient:

- Interface utilisateur en mode caractère

2 ème génération- Architecture client-serveur (2-tiers)

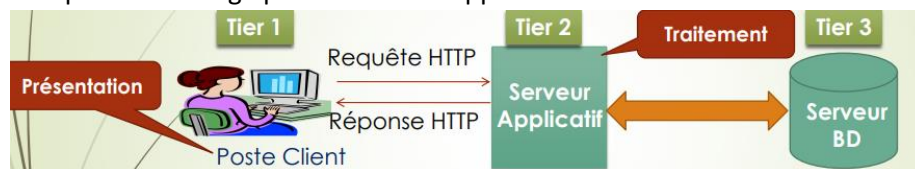
Avantage: interface utilisateur riche

Inconvénient: client lourd

3 ème génération- Architecture client-serveur (3-tiers)

Avantages: client léger, sécurité (pas d'exposition du schéma de la BD).

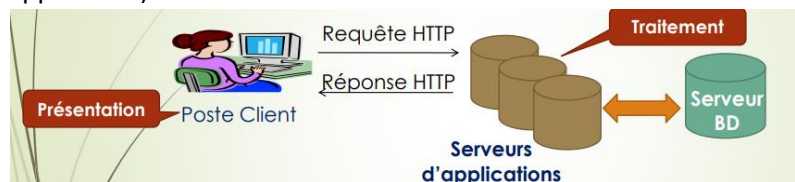
Inconvénient: importante charge pour le serveur applicative.



4 ème génération- Architecture n-tiers (1/4)

Avantage: la distribution d'application entre multiples serveurs

Inconvénient: problème d'interopérabilité (exprime la capacité du serveur d'application à communiquer et à utiliser les ressources d'autres serveurs comme, par exemple, les documents créés par une certaine application)



La communication entre les services

- On recherche en général un couplage faible entre les services
- Réutilisation de services existants
- Les services sont interopérables :
 - ➔ L'interopérabilité est la capacité des systèmes à communiquer, à échanger des données, à "travailler" ensemble, sans que l'utilisateur ait besoin de connaître les caractéristiques spécifiques à chaque système. L'interopérabilité est basée sur l'utilisation de standards définissant les interfaces des composants des systèmes

Le middleware

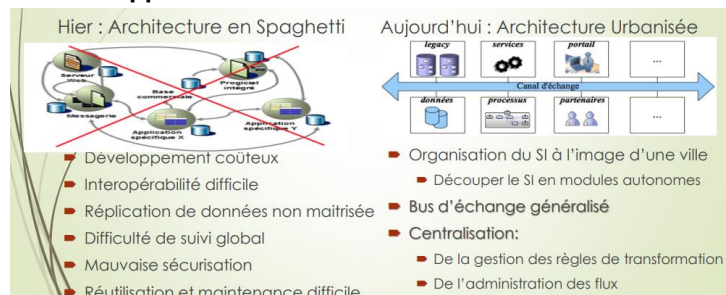
- Le middleware désigne les logiciels servant d'intermédiaire entre d'autres logiciels
- Les principales fonctions d'un middleware :
 - Fournir une interface (API) de haut niveau
 - masquer l'hétérogénéité des systèmes matériels et logiciels sous-jacents
 - Rendre la répartition aussi invisible ("transparente") que possible
 - Fournir des services répartis d'usage courant
 - Interopérabilité d'applications hétérogènes
 - Solutions qui ont fait leur preuve – CORBA, EJB, RMI, .Net Remoting,

Hétérogénéité du SI

Un système d'information d'entreprise est constitué :

- De nombreuses technologies différentes
 - De spécificités liées aux contraintes du moment
 - Des interfaces avec d'autres applications
- La création d'applications dans l'entreprise est très souvent pilotée par des besoins à très court terme
- Modélisation et développement dirigé par les choix/contraintes techniques
→ Pas de discussion entre maîtrise d'ouvrage (MOA) et maîtrise d'oeuvre (MOE)
- Décalage entre besoins métier et leur réalisation
- Pas de place pour la prise en compte de l'évolution des besoins fonctionnels au niveau de l'application

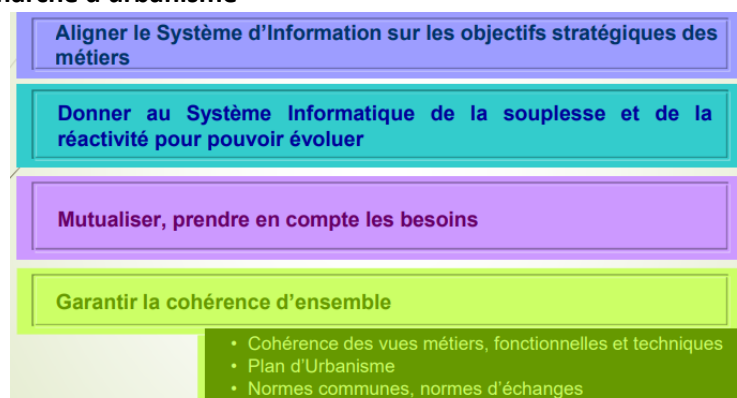
Évolution des Architectures Applicatives



Pourquoi l'urbanisation

- Réduire les coûts et les durées de développement et d'intégration
- Pallier rapidement aux changements : fonctionnels, technologiques ...
- Être compétitif
- Maîtriser son système d'information et ses évolutions
- Pallier à l'hétérogénéité des architectures
 - Maîtriser les risques

Objectifs de la démarche d'urbanisme



Urbanisation – Hétérogénéité du SI

- L'urbanisation des SI travaille sur :
 - Processus métiers: Un processus métier est ensemble d'activités qui délivrent collectivement une valeur ajoutée tangible par une collaboration de plusieurs unités et acteurs.
 - Communications inter applicative
 - Sur la mise en place d'un référentiel transverses
- Le système d'information d'une entreprise:
 - L'architecture métier : processus métier
 - L'architecture fonctionnelle : blocs fonctionnels et flux d'information supports à la réalisation des processus métier, indépendamment des technologies mises en œuvre,
 - L'architecture applicative : blocs applicatifs et échanges, supports à la réalisation des blocs fonctionnels et des flux.
 - L'architecture technique : infrastructure sur laquelle sont implémentées et exécutés les blocs fonctionnels

Architectures urbanisées

Solutions: EAI, B2Bi, les ERP, ...

Objectifs:

- Maîtriser les flux d'information entre les différentes applications,
- Rendre flexible et évolutif les SI,
- Diminuer les coûts d'intégration des applications.

Les progiciels de gestion intégrés ou ERP (Enterprise Resource Planning)

Le principe fondateur d'un ERP est de construire des applications informatiques de gestion (paie, comptabilité, stock, etc.) de manière modulaire (modules indépendants l'un des autres) tout en partageant une base de données unique et commune.

Avantages:

- Optimisation des processus de gestion
- Cohérence et homogénéité des informations (une seule table article, une seule table client...

Inconvénients:

- Mise en œuvre complexe: Le temps moyen d'intégration des ERP dans les entreprises est de l'ordre 3-36 mois.
- Coût d'installation par utilisateur est élevé (matériel + licence + intégration + formation + maintenance)

Le Workflow

- Le workflow a pour objectif la coordination automatisée de tâches réalisées par des intervenants humains
- Le moteur de workflow transfère des documents entre les participants d'un processus en leur assignant des tâches (valider le document, effectuer une modification, etc.)

Avantages:

- les concepts sont clairs,
- les outils relativement aisés à mettre en place.

Inconvénients:

- les participants humains, ne tiennent pas compte des applications du système d'information. L'intégration du workflow aux systèmes est une tâche difficile qui nécessite beaucoup de code propriétaire
- Les documents et les tâches ne sont pas suffisants pour l'automatisation des processus métiers. Il est nécessaire d'avoir un niveau d'abstraction supplémentaire, où l'on parle plus généralement de services, et d'informations

EAI (Enterprise Application Integration)

L'EAI est une architecture intergicielle permettant à des applications hétérogènes de gérer leurs échanges.

Sa particularité est d'échanger les données en temps réel.

Avantages :

- Flux centralisés
- Flux traités « au fil de l'eau »
- Flux réutilisable

Inconvénients :

- Flux massif: pour les flux massifs (par exemple : mise à jour de 10 000 articles en même temps), la logique du traitement unitaire de l'information est très lente.
- Coût initial: le coût de mise en place de l'infrastructure est assez élevé.

B2Bi (Business to Business integration)

Les outils B2Bi visent à assurer la collaboration entre entreprises partenaires

Les outils de B2Bi visent à définir les processus de collaboration entre entreprises partenaires, partant du principe que les processus intra- entreprise, donc d'EAI, n'ont pas les mêmes contraintes que les processus externalisés.

Hétérogénéité du SI (conclusion)

- Entreprises découpées en départements fonctionnels y compris le SI
- Processus métiers de + en + inter-départementaux
- Les processus franchissent les frontières de l'entreprise qui doit pouvoir prendre en compte les activités et processus des partenaires pour être réactive

➤ Coûts considérables dans la gestion des flux entre départements et dans l'intégration de leurs SI

→Vers une Architecture Orientée Services

Notion de Service

- Périmètre fonctionnel qu'on souhaite exposer à un certain type de consommateurs.
- Ensemble de fonctionnalités qui ont un sens.
- Expose un petit nombre d'opérations offrant un traitement de bout en bout.
- Est implémenté par un fournisseur et utilisé par un consommateur.

Caractéristiques d'un Service

Large Granularité (coarse-grained)

→ Les opérations proposées par un service encapsulent plusieurs fonctions et opèrent sur un périmètre de données large.

Interface

→Un service peut implémenter plusieurs interfaces, et aussi plusieurs services peuvent implémenter une interface commune.

Localisable

→Avant d'appeler (bind, invoke) un service, il faudra le trouver (find).

Instance unique

→À la différence des composants qui sont instanciés à la demande et peuvent avoir plusieurs instances en même temps, un service est unique.

Couplage faible (loosely-coupled)

→Les services sont connectés aux clients et autres services via des standards

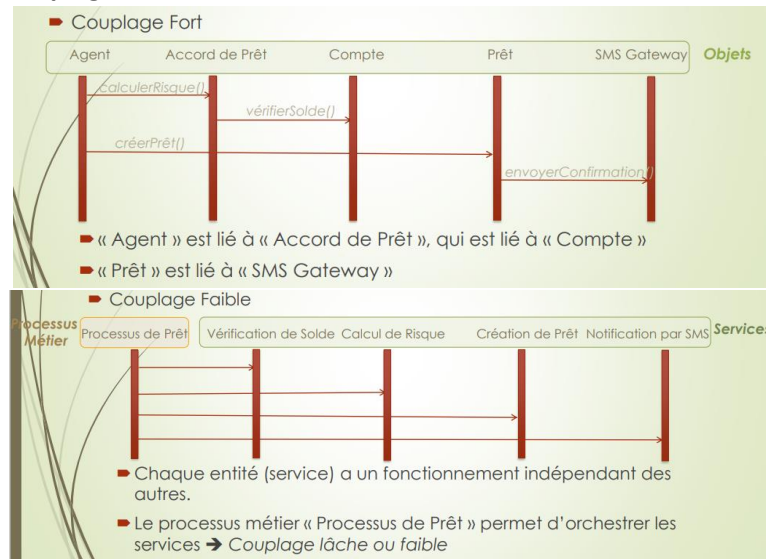
→Ces standards assurent le découplage, c.-à-d. la réduction des dépendances.

→Ces standards sont en général des documents XML.

Synchrone ou Asynchrone

→attente de réponse après l'utilisation d'un service ou non.

Couplage Fort vs. Couplage Faible



Types de Services

- Les services de présentations ou de référencement
- Les processus métiers
- Les services de gestion et d'accès aux bases de données
- Les services d'intégration

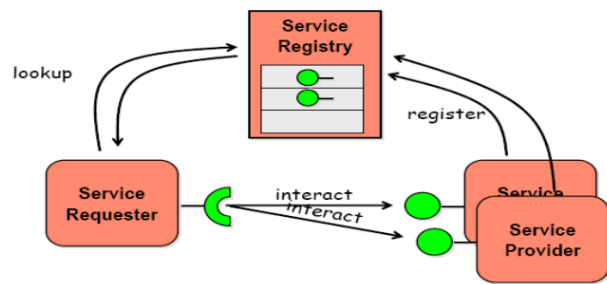
Architecture Orientée Services

Style d'architecture urbanisée et organisée à partir de services métiers mutualisés pour un ensemble de processus métiers ou d'applications.

Objectifs :

- Décomposer une fonctionnalités en un ensemble de fonctions basiques
- Décrire finement le schéma d'interaction entre ces services

les 3 acteurs principaux dans SOA



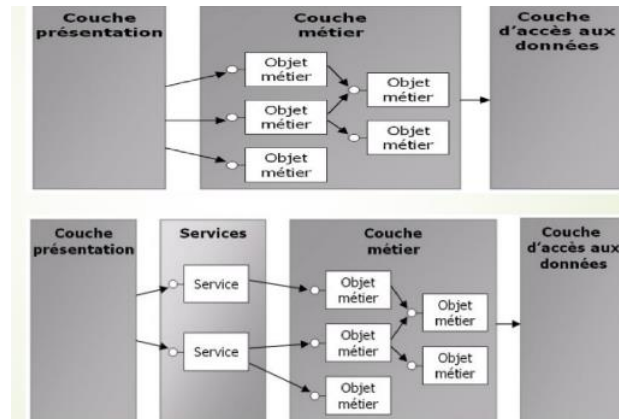
3 acteurs dans SOA :

- Producteur, fournit un service
- Consommateur, utilise un service
- Annuaire stocke des informations sur les services publiés

Communication unifiée au moyen d'un contrat d'échanges

- Contractualisation fonctionnelle
- Qualité de service

Comparaison de POO et POS



Bénéfices des services

- Améliorer l'agilité et la flexibilité du métier
- Faciliter la gestion des processus métier
- Réduire la complexité de la solution
- Construire les services une seule fois et les utiliser fréquemment