

Applications cartographiques sur le web

Architecture et serveur



Jean-Christophe Desconnets
Jean-Christophe.Desconnets@ird.fr



Plan

- Quelques notions d'architecture
- Application cartographique web
 - Objectifs et place dans un SI
 - Fonctionnalités coté client, coté serveur
 - Architecture
- Serveur cartographique
 - Définition et vocabulaire
 - Architecture et composants
 - Les données délivrées
 - Optimisation des requêtes

Introduction

- Application cartographique web sont des applications distribuées, c'est à dire
- Un ensemble de programmes,
- Distribués sur un réseau de communication,
- Qui collaborent pour assurer un service.

Comme par exemple

- une grappe de calculateurs, une application de commerce en ligne, un calendrier partagé

Introduction

Pourquoi des applications distribuées ?

Besoin intrinsèque de l'application

- Les utilisateurs sont répartis (ex : site web)
- les données sont réparties (ex : station météo)
- Objectif de partage de données (ex P2P)

Mais aussi

- Besoin de performance (ex : grappe de calcul)
- besoin de disponibilité (ex : redondance)
- besoin de modularité (ex: découplage gestion client / gestion personnel)

Quelques notions d'architecture

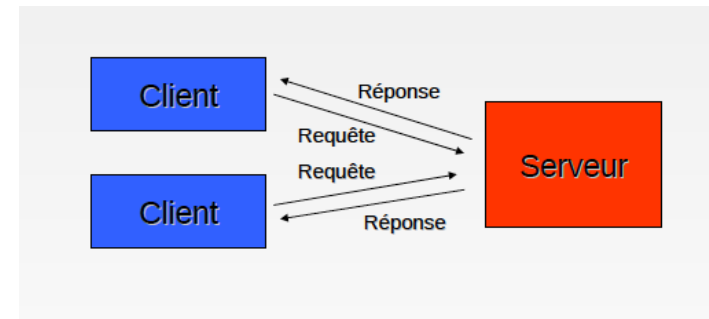


Trois grands types d'architecture

- Couplage fort (inter dépendance des composants)
 - ex : architecture « Peer 2 peer » (tous les processus ont le même rôles), grappes de calcul
- Couplage faible (pas d'interdépendance des composants)
 - architecture client/serveur
 - architecture ntiers
 - architecture 3 tiers
- Couplage très faible (les composants sont remplaçables)
 - architecture orientée service

Architecture Client/Serveur

- 2 types de noeuds :
 - un serveur
 - un client
- Un client fait une requête
- et recoit une réponse du serveur
- notion de session (ensemble de requêtes pour un même client)
- Exemples :
 - Client FTP/ serveur FTP
 - Navigateur Web / serveur web
 - ...



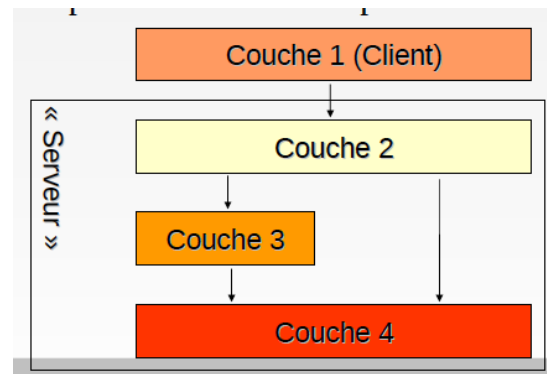
La majorité des applications sont construites sur le
modèle client / serveur

Architecture multi-couches

- Objectifs : Maitriser la complexité des applications par décomposition de l'architecture
- Principes :
 - Découper l'application en un ensemble de composants (ou couche distincts et faiblement couplés)
 - chaque composant est plus simple et l'application distribuée reste faiblement couplée
 - Les composants communiquent entre eux sur le modèle client/serveur
 - Les composants peuvent facilement réparties sur plusieurs machines

Architecture multi-couches

- Chaque couche d'une application multi couches est cliente de ses couches inférieures et serveur pour les couches supérieures



- La plupart des applications se ressemblent et composent :
 - de données
 - de traitements (sur ces données)
 - de présentation (sur les données et des résultats de traitements)

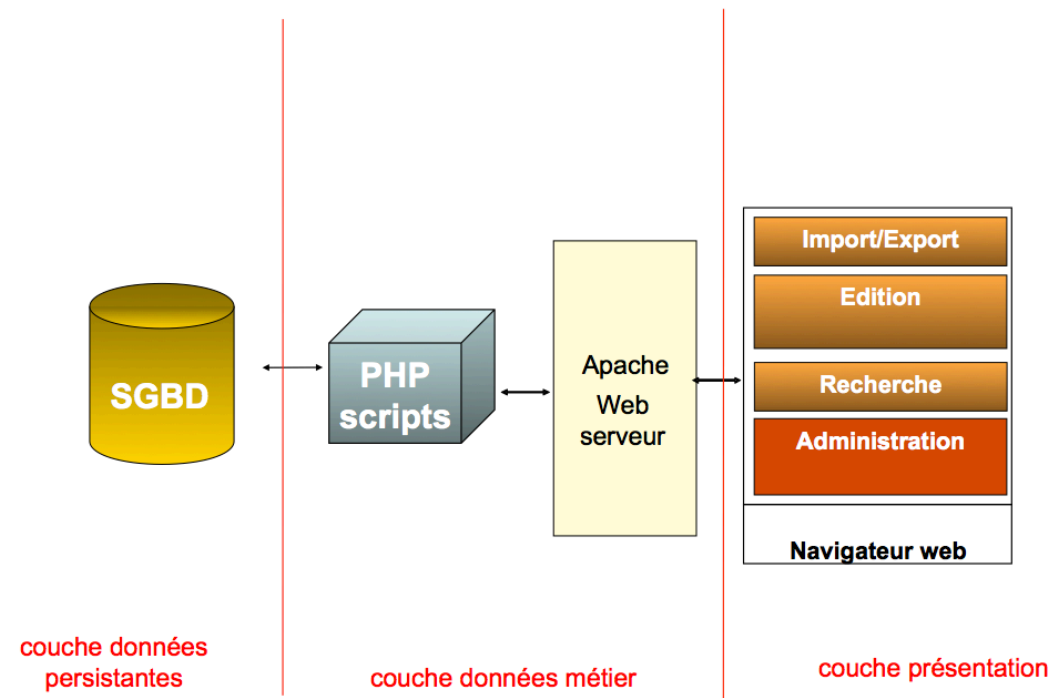
3 grandes couches = architecture 3 tiers

Architecture à 3 couches (3 tiers)

- Le coeur de l'application (couche métier) :
 - modèle Objet et traitements spécifiques au domaine de l'application
- Les données persistantes (couche d'accès aux données)
 - couche basse faisant le lien entre le modèle métier et le stockage physique des données (système fichiers, SGBD, ...)
- L'interface utilisateur (couche présentation)
 - interface utilisateur permettant à l'utilisateur d'agir sur le modèle métier (interface graphique, interface web, ...)

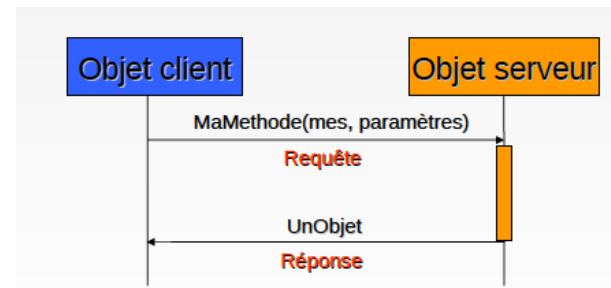
Architecture 3 tiers

- Exemple d'une application web construite sur une architecture 3 tiers: MDweb

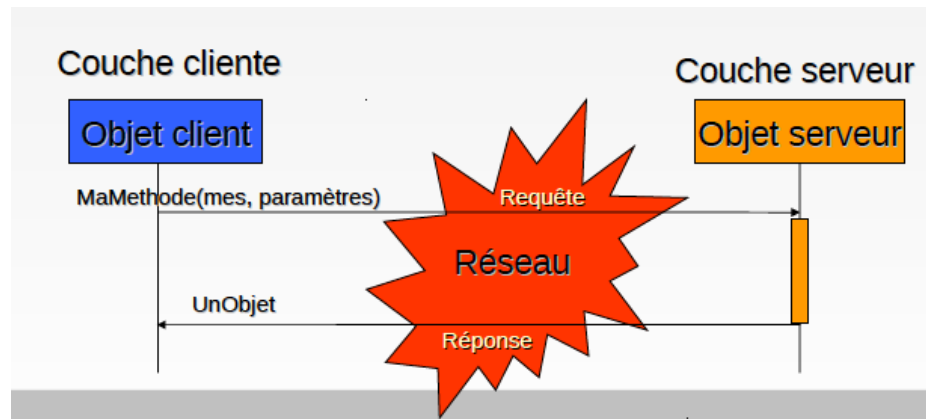


Communication entre les couches

- Dans le paradigme Objet, les communications sont déjà sur un modèle client/serveur :
- Requête : appel de méthodes
- Réponse : valeur de retour

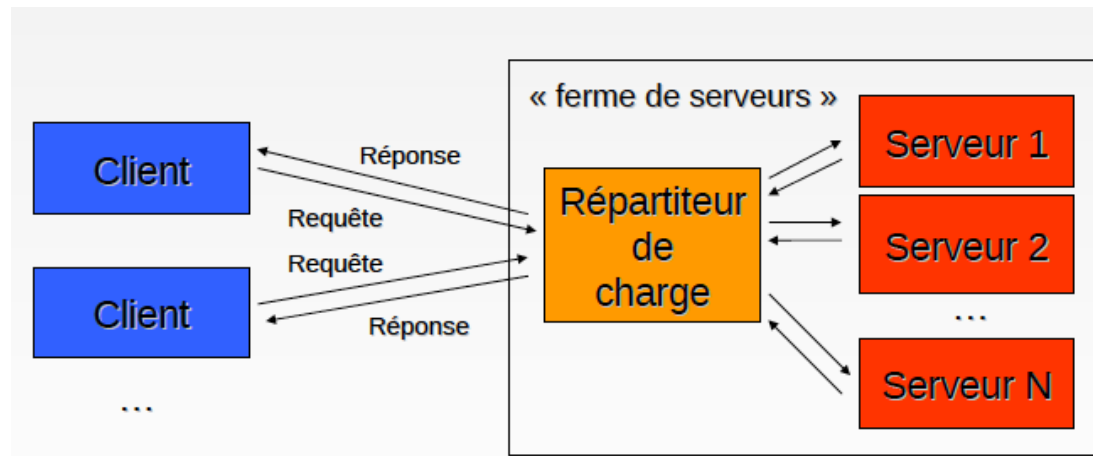


- Architecture multi couches : communication entre les différentes couches : Appel de méthodes sur des objets distants



Répartition de charge

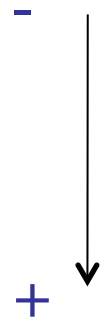
- Contraintes liées à ces architectures :
 - Un serveur traite simultanément les requêtes de plusieurs clients
 - Les requêtes de deux clients sont indépendantes
- Solution :
 - Paralléliser sur plusieurs serveurs identiques l'exécution des requêtes concurrentes (« load balancing »)
- Les requêtes des clients passent par un répartiteur de charge qui les répartit sur plusieurs serveurs



Application web cartographique



Objectifs et place dans un SI

- 
- Répond à quels objectifs :
 - De diffusion de ses cartes, **essentiellement**
 - D'accès à ses données spatiales pour leur partage, **plus rarement**
 - d'Analyse spatiale, **pourquoi pas** mais d'une manière limitée aujourd'hui
 - Place dans un SIG
 - **une application web cartographique n'est pas un SI** mais un module dédié à la diffusion des données sous forme cartographique (sous sa forme minimale)

Fonctionnalités attendues (1/3)

- Les besoins principaux auxquels elles doivent répondre :
 - Coté client (entre les mains des utilisateurs finaux)
 - Mettre en ligne des cartes ou des données spatiales
 - Permettre la navigation (zoom, pan, ..) dans la carte et l'exploration des objets représentés (requêtes)
 - Coté serveur (pour la construction des cartes)
 - Agréger des données spatiales issues de différentes sources et format
 - Réaliser des traitements pour construire la carte et sa symbologie

Fonctionnalités attendues (2/3)

Fonction de représentation

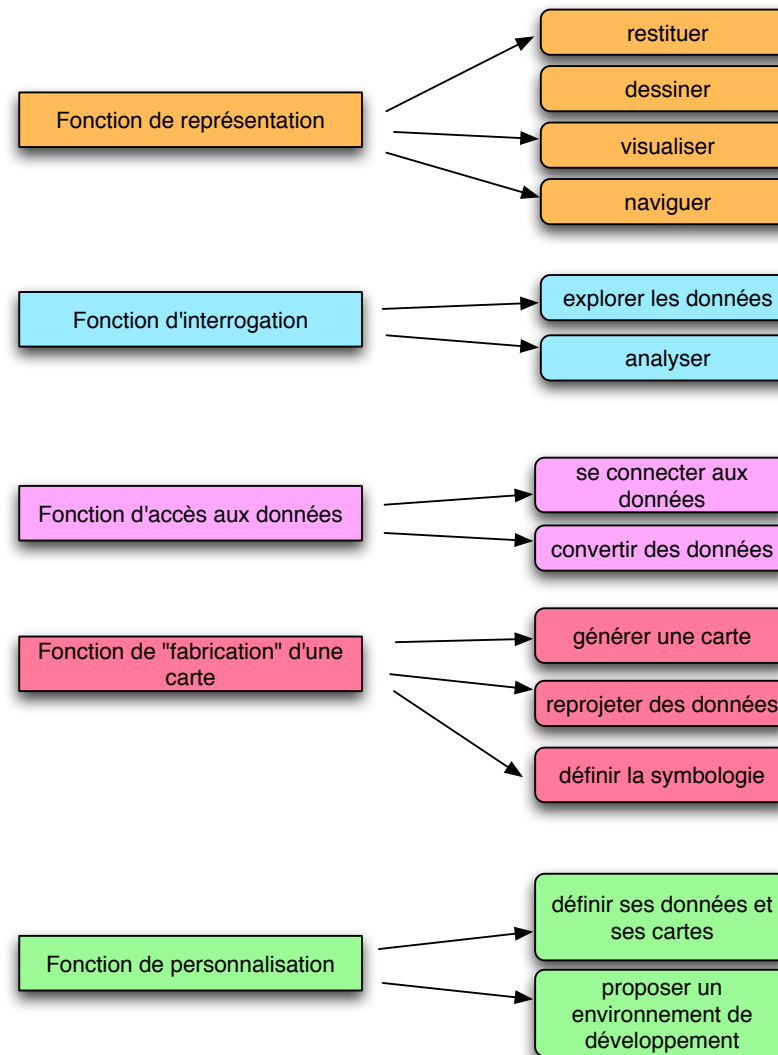
Fonction d'interrogation

Fonction d'accès aux données

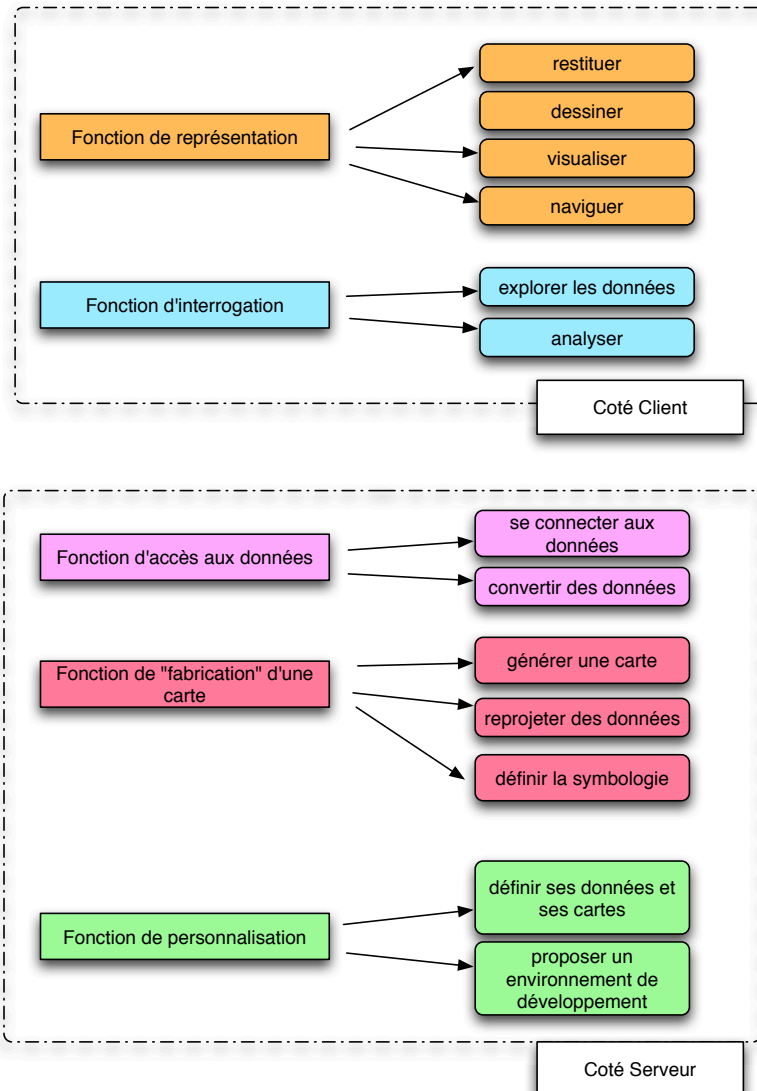
Fonction de "fabrication" d'une
carte

Fonction de personnalisation

Fonctionnalités attendues (3/3)



Fonctionnalités attendues (3/3)



Architecture d'une application web cartographique

