

Getting Things Done

Projet multi-modules

- Génie logiciel à objets
- Objets distribués
- IHM

Louis-Alexandre Celton
Gaëtan Hervouet
Damien Lévin
Paul Vaillant

Master 2 ALMA



Sommaire

- I. Introduction
- II. Cahier des charges
 - 1) Exigences fonctionnelles
 - 2) Autres exigences
- III. Environnement
 - 1) Topologie du système
 - 2) Web
- IV. Conception
 - 1) Architecture
 - 2) Décisions
 - 3) Interface Homme-Machine
- V. Conclusion
 - 1) Rétrospectives
 - 2) Problèmes

I. Introduction

Getting Things Done :

- Méthode de gestion de tâches
- Notions importantes :
 - ❖ Tâche
 - ❖ Projet
 - ❖ Contexte
 - ❖ Idée
 - ❖ Note
- 4 étapes principales :
 - Collecte d'informations
 - Traitement des informations
 - Organisation des données
 - Mise au point

II. Cahier des charges

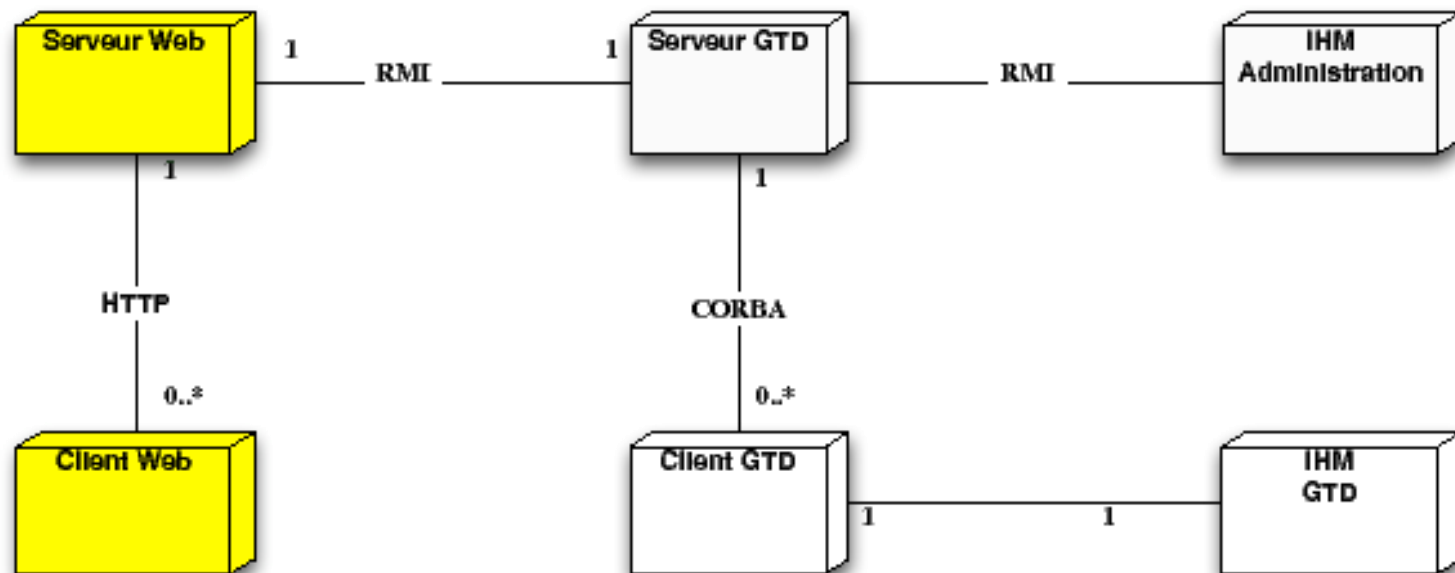
- 1) Exigences Fonctionnelles
 - a) Identification
 - b) Etapes de la méthode GTD :
 - Collecte d'informations
 - Traitement des informations
 - Organisation des données
 - Mise au point
 - c) Utilisation des serveurs GTD & Toodledo
 - Synchronisation

2) Autres exigences

- ✓ Utilisabilité
- ✓ Fiabilité
- ✓ Performance
- ✓ Adaptabilité
- ✓ Maintenabilité

III. Environnement

1) Topologie du système



Getting Things Done

2) Architecture Web

- a) Client léger accessible de n'importe quel endroit
- b) Compatibilité accrue
- c) Facilité de déploiement, aucune installation
 - ❖ Un poste équipé d'un serveur web
 - ❖ Connecté à Internet

IV. Conception

Problème d'expertise :

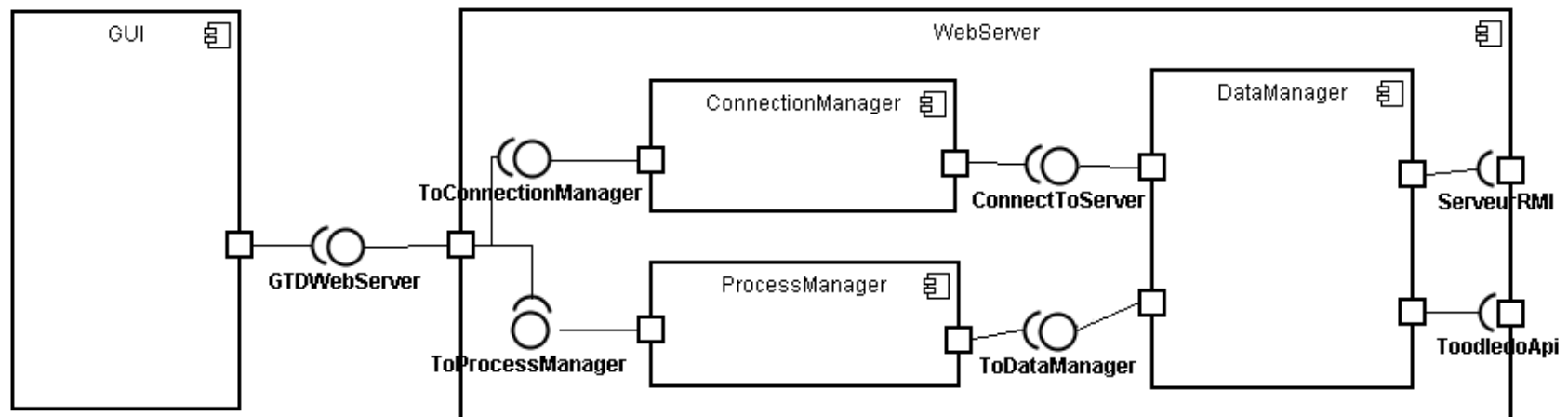
- Aucune persistance nécessaire
 - Aucune base de données
- L'interface pouvait être directement branchée au serveur GTD central.

Rajout d'un serveur supplémentaire, Toodledo :

- Problèmes de synchronisation à résoudre...

1) Architecture

a) Composants

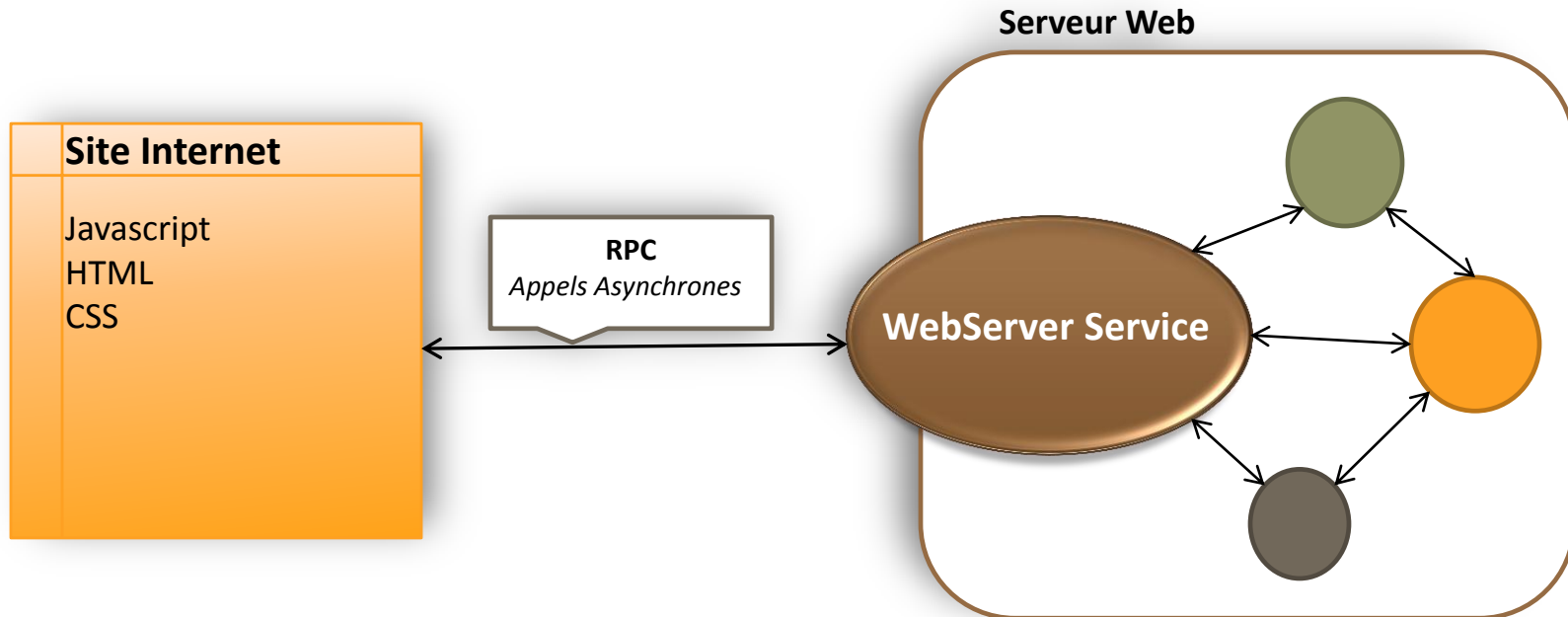


a) Technologies

Google Web Toolkit :

- ❖ Pourquoi **GWT** ?
- ❖ Google Web Toolkit (GWT) ensemble d'outils logiciels développés par Google
- ❖ Principe
 - Ecriture du code en java
 - Mode développement
 - Compilation
 - Code client JavaScript optimisé
 - Service web exécutable sur un serveur Tomcat
 - Déploiement facilité vers l'Appengine de Google (Cloud Computing)

Google Web Toolkit



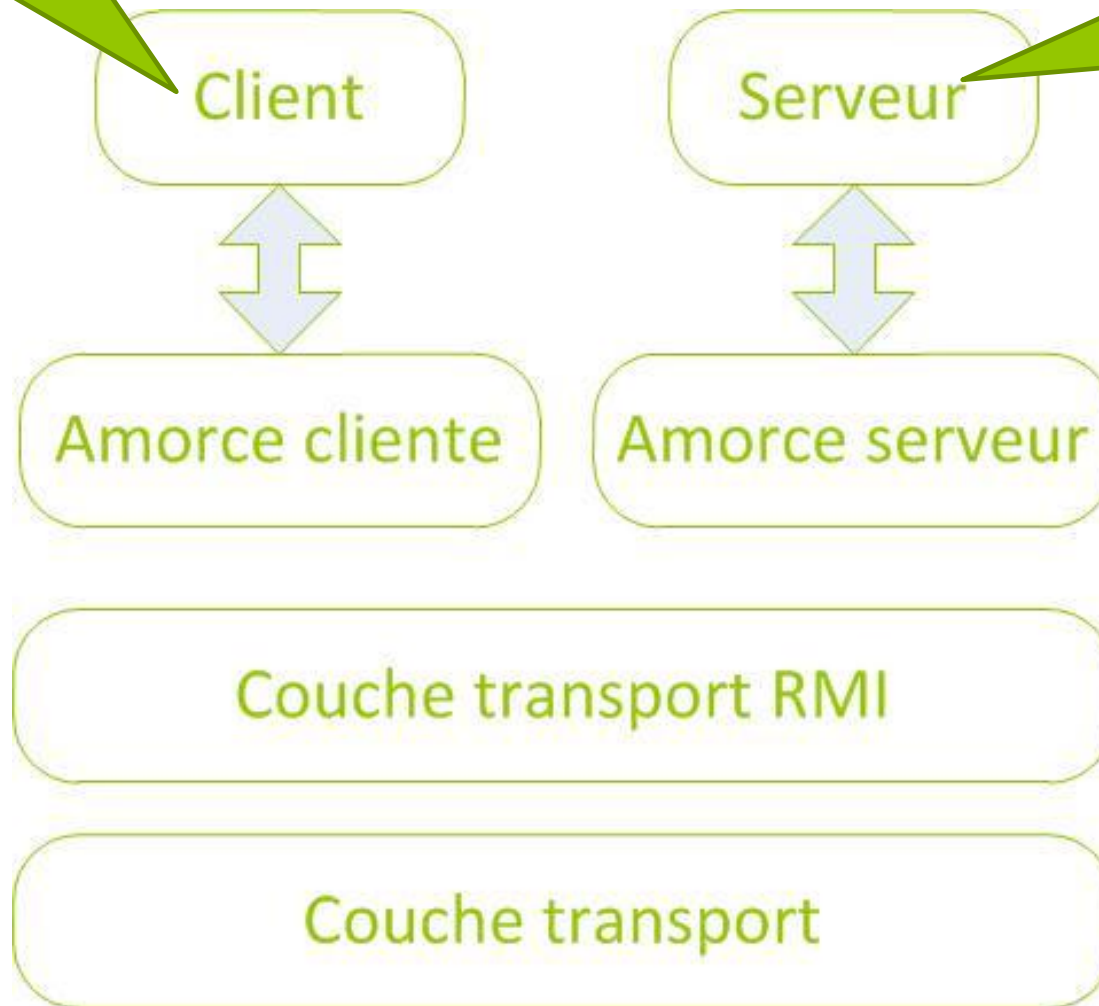
Fonctionnement de GWT dans AlmaWebGTD

Getting Things Done

Remote Method Invocation (**RMI**):

- Définition : *interface de programmation (API) pour le langage Java qui permet d'appeler des méthodes distantes*
- Protocole imposé

ServerGTDImpl.java



Interface
ServeurRMI.java

Toodledo

- ◉ Librairie toodledo-java
- ◉ Utilise l'architecture **REST**
(Representational State Transfer)
 - ◉ http
 - ◉ URI

2) Décisions

a) Synchronisation

Problèmes :

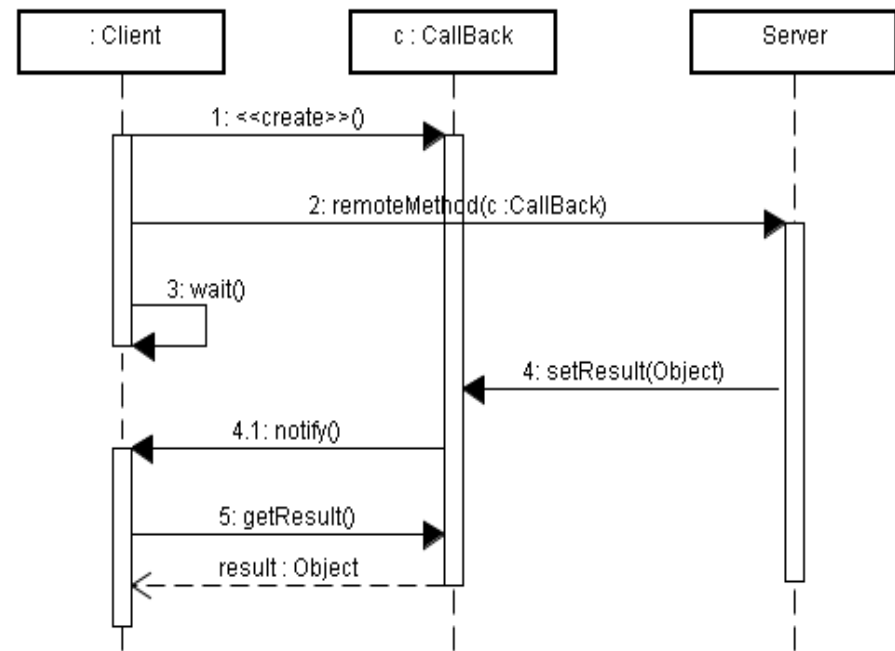
- 2 serveurs distincts (Serveur GTD & Toodledo)
 - Chacun accessible de différentes manières
- Concepts absents sur Toodledo
(idées, notes, hiérarchie de projets...)

Solution :

- Le serveur GTD sert de référence.

b) Callbacks

- Communication externe entre notre serveur et le serveur GTD en **asynchrone**
- Communication interne entre les composants de notre serveur en **synchrone**



Solution : Asynchrone vers Synchrone

c) Exceptions

- Générées par les serveurs :
 - RMI
 - REST
 - Mauvais identifiants
 - Serveur GTD (`CallBack::onFailure(Exception)`)
 - ...
- Remontées jusqu'à l'IHM

3) Interface Homme Machine

- Utilisation de la méthode LUCID
- Méthode GTD au centre de l'IHM
 - Axe principal au lancement de l'application
 - ❖ Permet de guider l'utilisateur dans le respect de la méthode GTD.
- Deux types d'interface
 - ✓ Mode « user friendly »
 - ✓ Mode avancé
- Fonctionnalités importantes
 - Méta-compte
 - Signalisation des différences entre les deux types de compte
 - Signalisation des erreurs
 - Messages précis
 - Signalisation des attentes dues aux connexions serveur

Getting Things Done

3) Interface Homme Machine



V. Conclusion

- Développement d'un projet multi-groupes de l'analyse à l'implémentation.
- Analyse et Conception mûres
- Manque d'un véritable chef de projet général
- Expérience enrichissante au niveau des connaissances et application du cours

1) Rétrospectives

- Plus de temps sur le développement de l'interface Homme-Machine pour coller à notre idée initiale
- Meilleure communication entre les groupes
- Mise en place de transactions afin de gérer les erreurs d'enregistrement propres à chaque serveur et de maintenir la cohérence des données

2) Problèmes

- Développement du serveur en même temps.
- Pas de choix d'une analyse de la méthode GTD comme axe central.
- Absence de persistance rend plus difficile les jeux d'essais.
- GWT ralentit les tests .