

I402A Architecture logicielle

Séance 12

Architecture micro-kernel

Sébastien Combéfis

2017–2018



Ce(tte) œuvre est mise à disposition selon les termes de la Licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Pas de Modification 4.0 International.

Système adaptable

- Nécessité d'avoir des **systèmes adaptables**
 - Un système doit pouvoir évoluer au cours du temps
 - Fonctionnalité doit être ajustée, ajoutée, supprimée
- Il faut penser en mode ***“Design for Change”***

Designer son système pour supporter son évolution



Microkernel

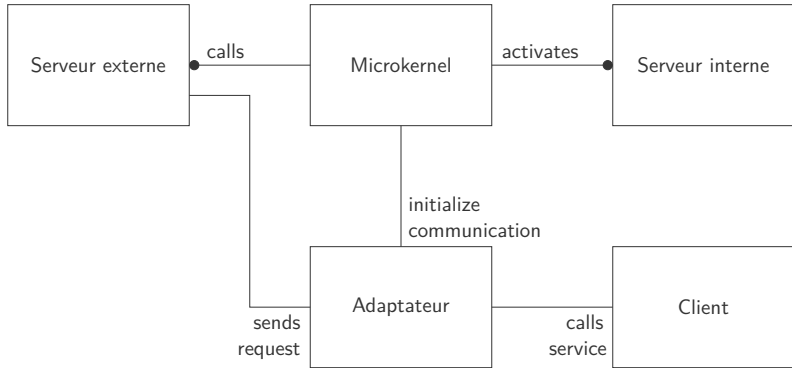
Microkernel

- Définition de **fonctionnalités cœurs** du système
 - Séparation des fonctionnalités avancées du système
 - Distinction des parties spécifiques aux clients
- Coordination d'extensions de type **plugins**

Possibilité de charger/décharger et contrôler des extensions
- Architecture formées avec **cinq composants**

Microkernel, serveurs (int/ext), connecteur client, adaptateur

Architecture microkernel (1)



Composant (1)

- Le **microkernel** encapsule tous les services de base
 - Gestion des ressources (processus, fichier...)
 - Fournit la communication interprocessus
- **Serveur interne** réservé à l'utilisation par microkernel
 - Services additionnels outsourcés comme driver de périphérique
 - Encapsule les dépendances systèmes
- Vue spécifique sur le microkernel offerte par **serveur externe**
Processus spécifiques qui sont exécutés sur le microkernel

Composant (2)

- Présence de plusieurs **adaptateurs** relais avec serveur externe
 - Émule une API de plateforme pour un serveur externe
 - Permet de découpler le client des serveurs externes
- Un microkernel est utilisé par des **clients**

Utilisent les serveurs externes via les adaptateurs

Avantage

- Permet une **grande portabilité** du microkernel

Effort de portage réduit de part sa petite taille

- **Bonne flexibilité** et possibilité d'extension

- Grâce au système de serveur externe et adaptateur
- Séparation du mécanisme (processus dans microkernel)
de la politique (scheduling dans un serveur externe)

- Amélioration de plusieurs critères pour des **systèmes distribués**

Scalability, robustesse et transparence par rapport à la localisation

Inconvénient

- **Moins efficace** qu'un système monolithique

Plus grande distance avec microkernel, et temps communication

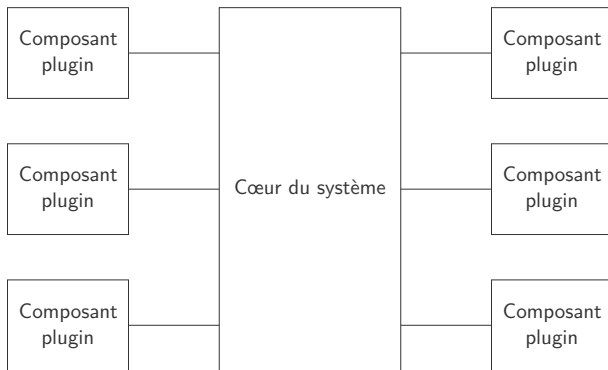
- Plus **grande complexité** de design et implémentation

Découpe et séparation des fonctionnalités est importante

Plugins



Architecture microkernel (2)



Crédits

- <https://www.flickr.com/photos/erix/582358634>
- <https://www.flickr.com/photos/theabbott/2781713451>