

A propos de moi...



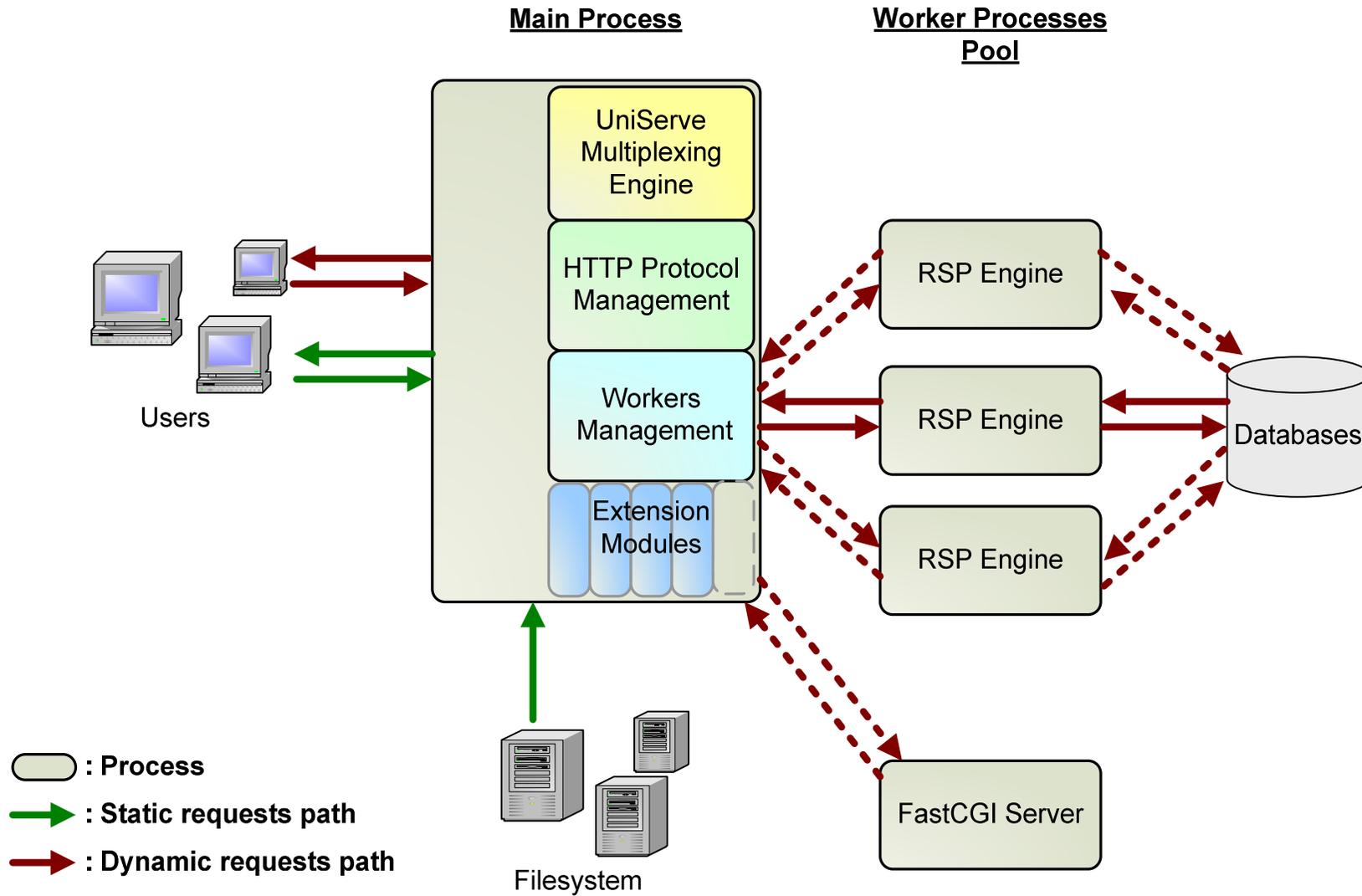
- Nenad "DocKimbel" Rakocevic
- Programmeur depuis 25 ans: C/C++, *Basic, ASM, REBOL, Javascript,...
- Co-fondateur d'une entreprise IT à Paris: Softinnov
- Auteur de plusieurs librairies pour REBOL:
 - Drivers natifs: MySQL, PostgreSQL, LDAP
 - Driver Windows NTLM
 - UniServe: moteur de communication asynchrone
 - CureCode: outil web de suivi d'anomalies très léger et rapide (look à la Mantis)
 - Divers autres outils, jeux, demos...
- Ancien utilisateur Amiga et développeur BeOS enregistré

Cheyenne Web Server: Introduction

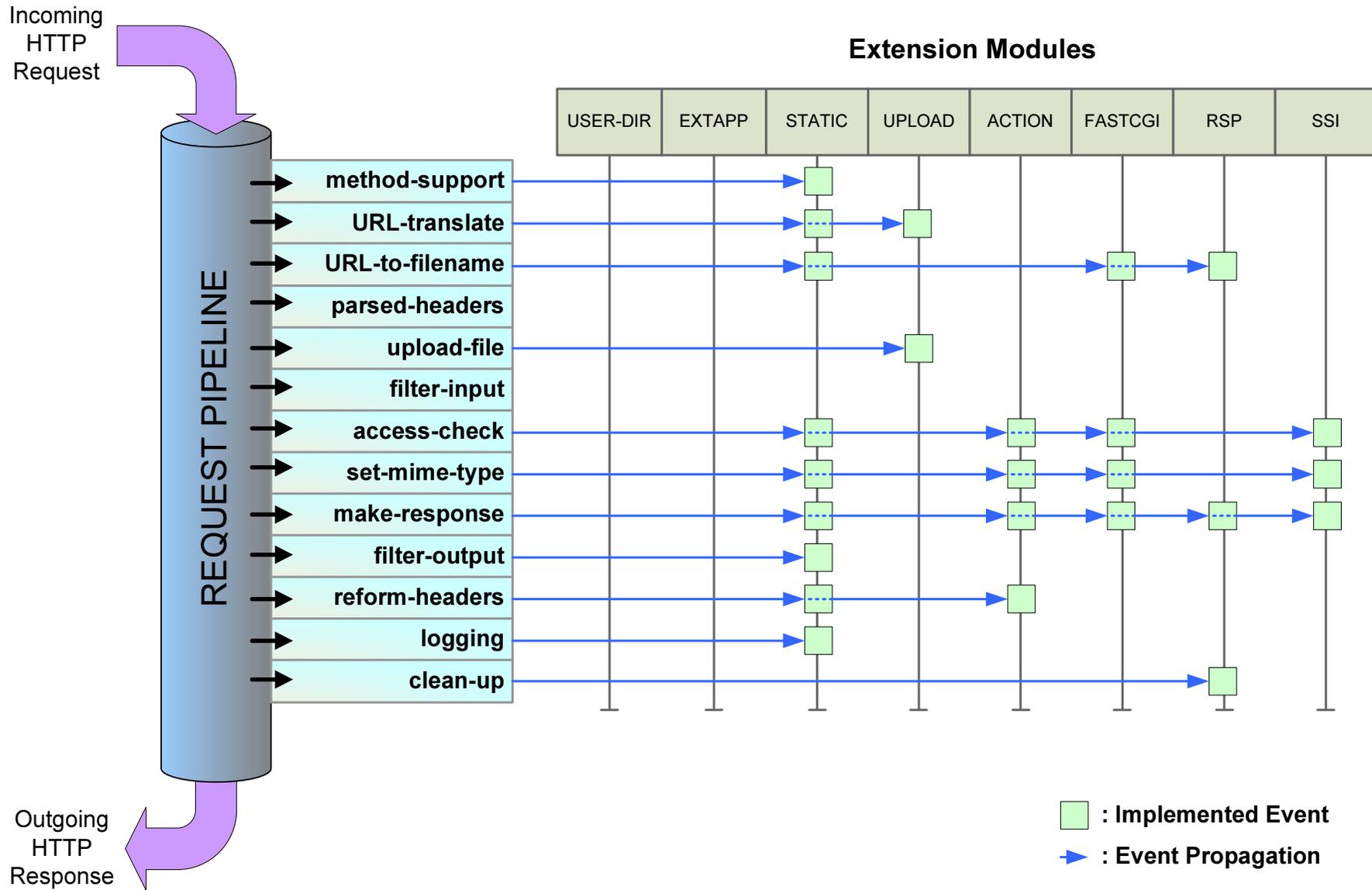


- **Pourquoi créer un nouveau serveur web?**
 - Fournir un conteneur natif pour les applications web écrites en REBOL
 - Léger, efficace, multi-OS, facile à déployer, facile à étendre
 - Test de charge pour REBOL
- **Quelques éléments clés...**
 - Executable de ~500Ko (~90Ko pour le code Cheyenne, le reste pour l'interpréteur REBOL)
 - Ne nécessite pas d'installateur
 - Open source (BSD), hébergé sur Google Code
 - Architecture interne modulaire (mods, handlers,...)
 - Technologies clés supportées: FastCGI, WebSockets, ...
 - Propulsé par un moteur d'E/S réseau asynchrone "rapide": UniServe
 - Performance aussi bonne que les autres serveurs webs interprétés (Mongrel,...)
 - Utilisé en production par plusieurs entreprises: Softinnov, RT, Synapse EMR,...

Cheyenne Architecture: Vue globale



Cheyenne Architecture: Le pipeline des requêtes



Cheyenne: Servir du contenu



- **Configurer**

- Fichier de configuration utilisant un dialect extensible
- Support des domaines virtuels
- *Panneau d'administration web pour la v1.0*

- **Servir**

- Contenu statique: tous fichiers jusqu'à 2Go, fichiers de < 64Ko sont mis en cache mémoire
- Contenu dynamique: SSI, CGI, RSP (REBOL Server Pages), ...
- Contenu fourni par des serveurs externes: serveurs FastCGI (ex: PHP)

Cheyenne: scripter en RSP



- **Concepts de base**

- Templates : `<%...%>`, `<%=...%>`
- [API](#) riche (Request, Response, Session, ...)
- Execution rapide et concurrente (pre-compilé + caché en mémoire + processus en tâche de fond)

- **Gestion des sessions**

- Clé de session passée par : Cookie / URL / en données POST
- Contexte de la session : définition possible de données locales à la session stockables, modifiables, supprimables
- Gestion manuelle vs gestion automatique de la session :
 - Manuel => `session/start`, `session/stop`
 - Automatique => nécessite de définir une "webapp" dans le fichier config

Cheyenne: Application web (Web App)



- **Conteneur d'application**

- Hiérarchie de fichiers Privés / Protégés / Publics
- Evènements implémentables:
 - on-application-start, on-application-end
 - on-session-start, on-session-end
 - on-page-start, on-page-end
- Fine couche d'abstraction pour l'accès aux BDD (gestion transparente des connexions)
- Support minimaliste pour l'internationalisation (i18n) :
 - Dans le code source REBOL, en utilisant la fonction SAY
 - Dans les templates, en utilisant la forme littérale #[texte]

- **Services fournis aux applications**

- Gestion automatisée des sessions
- Gestion d'un status "authentifié" des utilisateurs (avec redirection automatique vers la page de login)

Cheyenne: Autres aspects...



- **Maintenance via une console externe**
 - Accéder au code en "live" du serveur depuis une console REBOL
 - Possibilité de faire des corrections à chaud!
- **Mode embarqué**
 - Inclure Cheyenne dans une application REBOL, même graphique
 - Servir du contenu dynamique depuis l'application directement (API dédiée)
- **API "Upload" pour les clients webs**
 - Obtenir des stats durant l'upload d'un fichier (octets reçus / restants)
- **Services expérimentaux inclus**
 - Moteur d'évènement type CRON avec son propre dialect
 - Serveur SMTP (agent MTA, actuellement limité au support 8-bit seulement)
- **Fonctionnement en Service Windows**
 - Basculer du mode "utilisateur" en mode "service" en un clic depuis la barre d'icônes Windows