



 *Cheyenne*

---

**Un serveur web léger et ouvert**

## A propos de moi...



- Nenad Rakocevic
- Programmeur depuis 25 ans: C/C++, \*Basic, ASM, REBOL, Javascript,...
- Co-fondateur de Softinnov, entreprise IT à Paris
- 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)



# Pourquoi créer un nouveau serveur web ?

- Fournir un conteneur natif pour les applications web écrites en REBOL
- Léger, efficace, multi-plateforme, facile à déployer, facile à étendre
- Test de charge pour REBOL

## Cheyenne Web Server: Introduction (2/3)



Ce Cheyenne n'habite pas dans la Freebox !



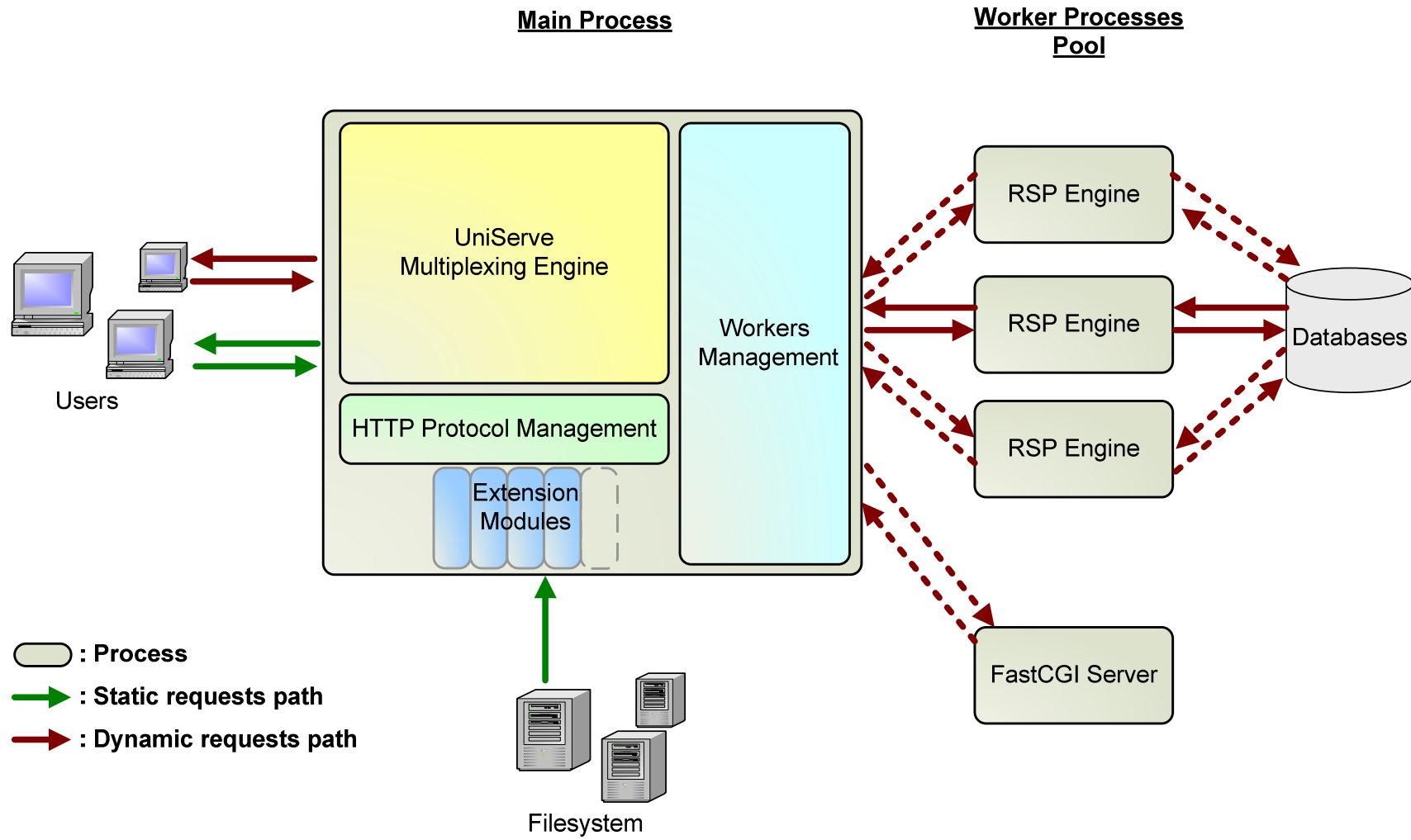
## Cheyenne Web Server: Introduction (3/3)



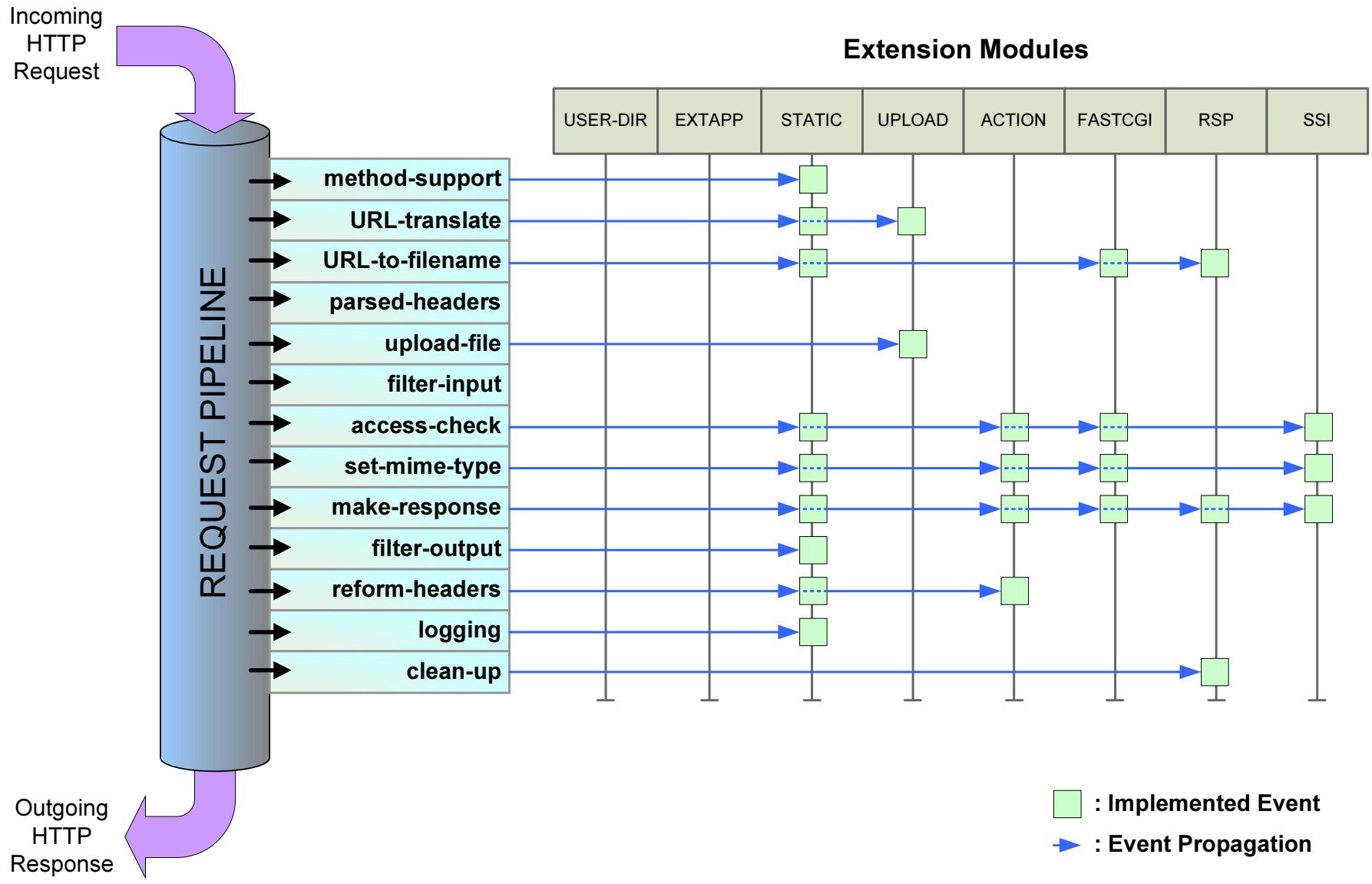
### Quelques éléments clés...

- Un seul fichier binaire de ~500Ko, "zero-install", configuration simple
- Open source (BSD), dépôt hébergé sur Google Code
- Architecture interne modulaire (d'inspiration Apache)
- Technologies clés supportées: FastCGI, Websocket, ...
- Propulsé par un moteur d'E/S réseau asynchrone
- Performance au niveau des 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





# Configurer

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



## Cheyenne: Configurer (2/2)



```
softinnov.org [  
    root-dir %/c/dev/si-org/           ;-- root folder using an absolute path  
    default %index.shtml              ;-- default file when no target  
    disable-log                       ;-- HTTP access logs disabled for this virtual host  
  
    alias "/test" %show.rsp           ;-- map /test path to %show.rsp script  
    socket-app "/chat.rsp" chat       ;-- declare a web socket application  
  
    on-status-code [  
        404 "/404.rsp"               ;-- redirect 404 return codes to a custom script  
    ]  
  
    webapp [  
        virtual-root "/testapp"      ;-- RSP webapp definition  
        root-dir %www/testapp/       ;-- URL path associated to this webapp  
        ;debug                        ;-- root folder of the webapp  
        ;debug                        ;-- debug mode commented (disabled)  
    ]  
]
```



## Servir du contenu

- 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)

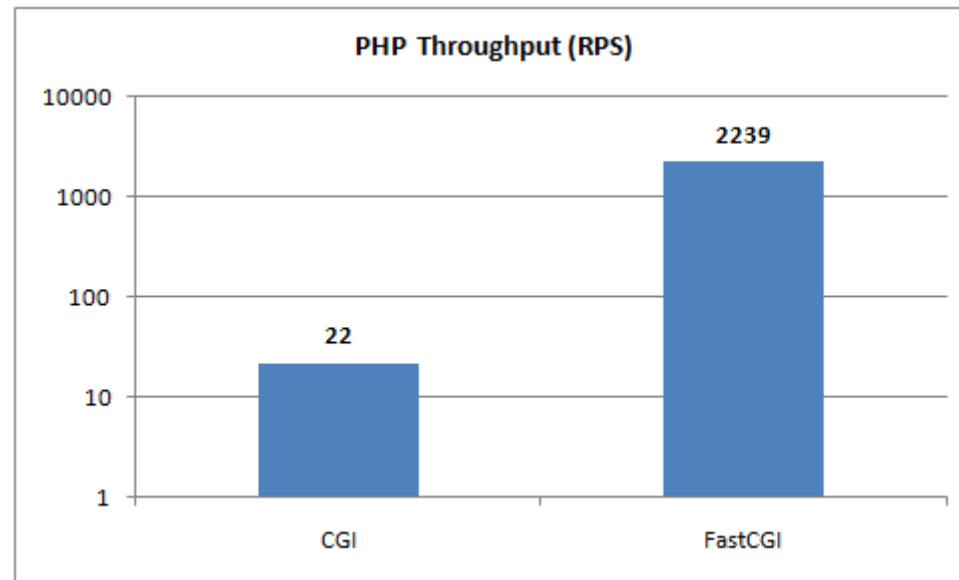


## Concepts de base

- Templates incluant des balises dynamiques: `<%...%>`, `<%=...%>`
- [API](#) riche (Request, Response, Session, ...) et documentée
- Execution rapide et concurrente (pre-compilé + caché en mémoire + processus en tâche de fond)
- Gestion des sessions clientes: manuelle ou automatique



## Applications PHP via FastCGI



- Simple à déployer et à configurer
- Multi-plateforme (Windows, Linux, Mac OS)
- Accès à l'immense bibliothèque d'applications PHP !

## Cheyenne: Support PHP (2/2)



```
extern-app [  
    name      php-fcgi  
    url       fastcgi://localhost:9999  
    command   "..\php\php-cgi.exe -b 127.0.0.1:9999"  
    ;command  "/usr/local/bin/php-cgi -b 127.0.0.1:9999"  
  
    environment [  
        PHP_FCGI_MAX_REQUESTS 0          ; Windows=>0, other=>500  
        PHP_FCGI_CHILDREN     4  
    ]  
  
    channels 1                          ; Windows=>1, other=>n  
  
    ; -- uncomment the following line for UNIX/OSX  
    ;delay 2  
]
```

## Cheyenne: Autres aspects... (1/2)



- **Fonctionnement en Service Windows**
  - Basculer du mode "utilisateur" en mode "service" en un clic depuis la barre d'icônes Windows
  
- **Maintenance via une console externe**
  - Accéder au code en "live" du serveur depuis une console REBOL
  - Possibilité de faire des corrections à chaud !

## Cheyenne: Autres aspects... (2/2)



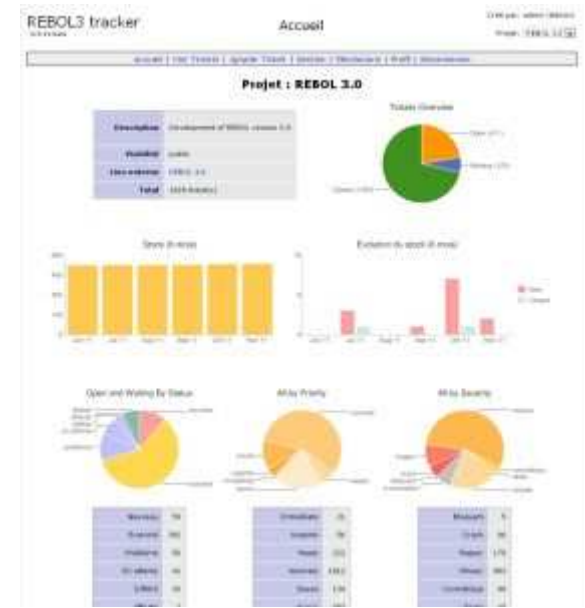
- API "Upload" pour les clients webs
  - Obtenir des stats durant l'upload d'un fichier (octets reçus / restants)
- Services expérimentaux inclus
  - Moteur d'évènement type CRON avec son propre dialecte
  - Serveur SMTP interne (agent MTA, actuellement limité au support 8-bit)

# Cheyenne Web Server: Exemples d'utilisation (1/2)



## Curecode: Outil de bugtracking en ligne

- "Mantis light", codé en RSP + MySQL
- Multi-projet, multi-instance
- Localisé (En, Fr)



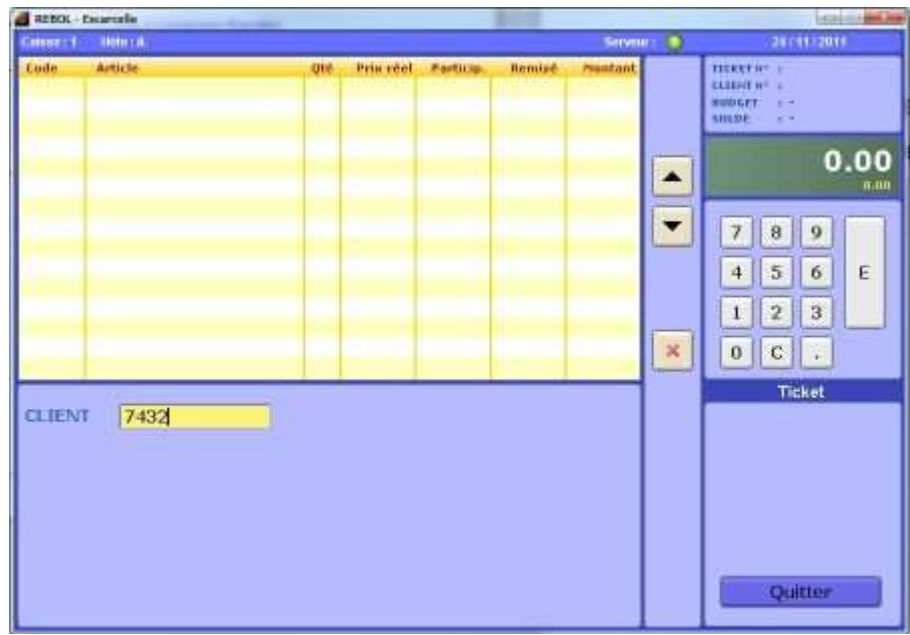


# Cheyenne Web Server: Exemples d'utilisation (2/2)



## Escarcelle: Gestion des Epiceries Solidaires

- ~150 clients, ~1000 utilisateurs quotidiens, ~100k requêtes RSP/jour
- Base de donnée: 1Go, 40 tables, ~10 millions d'enregistrements



# Cheyenne: Adéquation aux besoins



- **Avantages**

- Déploiement facile et rapide
- Très faible empreinte disque
- Performances suffisantes pour la plupart des petits et moyens sites
- Très bon niveau de sécurité
- Serveur d'application complet intégrée
- Support PHP en FastCGI
- Support commercial possible

- **Inconvénients**

- Performances sur fichiers statiques faible (nginx peut éliminer ce pb)
- Pas de SSL intégré (nginx, stunnel, ... peuvent fournir le SSL)
- Ne convient pas pour des sites avec charge importante (>1M req/jour)
- Serveur d'application intégré requiert la connaissance de REBOL

# Cheyenne Web Server: Fin... des questions ?



cheyenne-server.org

