Introducing myself…

- Nenad aka "DocKimbel" Rakocevic,
- Programming since 25 years: C/C++, *Basic, ASM, REBOL, web client-side languages,…
- Founder of a software company in Paris: Softinnov
- Author of several librairies for REBOL:
  - MySQL, PostgresQL, LDAP native drivers
  - Windows NTLM driver
  - UniServe: asynchronous, event-driven network engine
  - CureCode: very fast web-based bug tracker (Mantis-like)
  - Various others tools, game, demos…
  - Was an happy Amiga user and registered BeOS developer

Cheyenne Web Server: Introduction

http://cheyenne-server.org

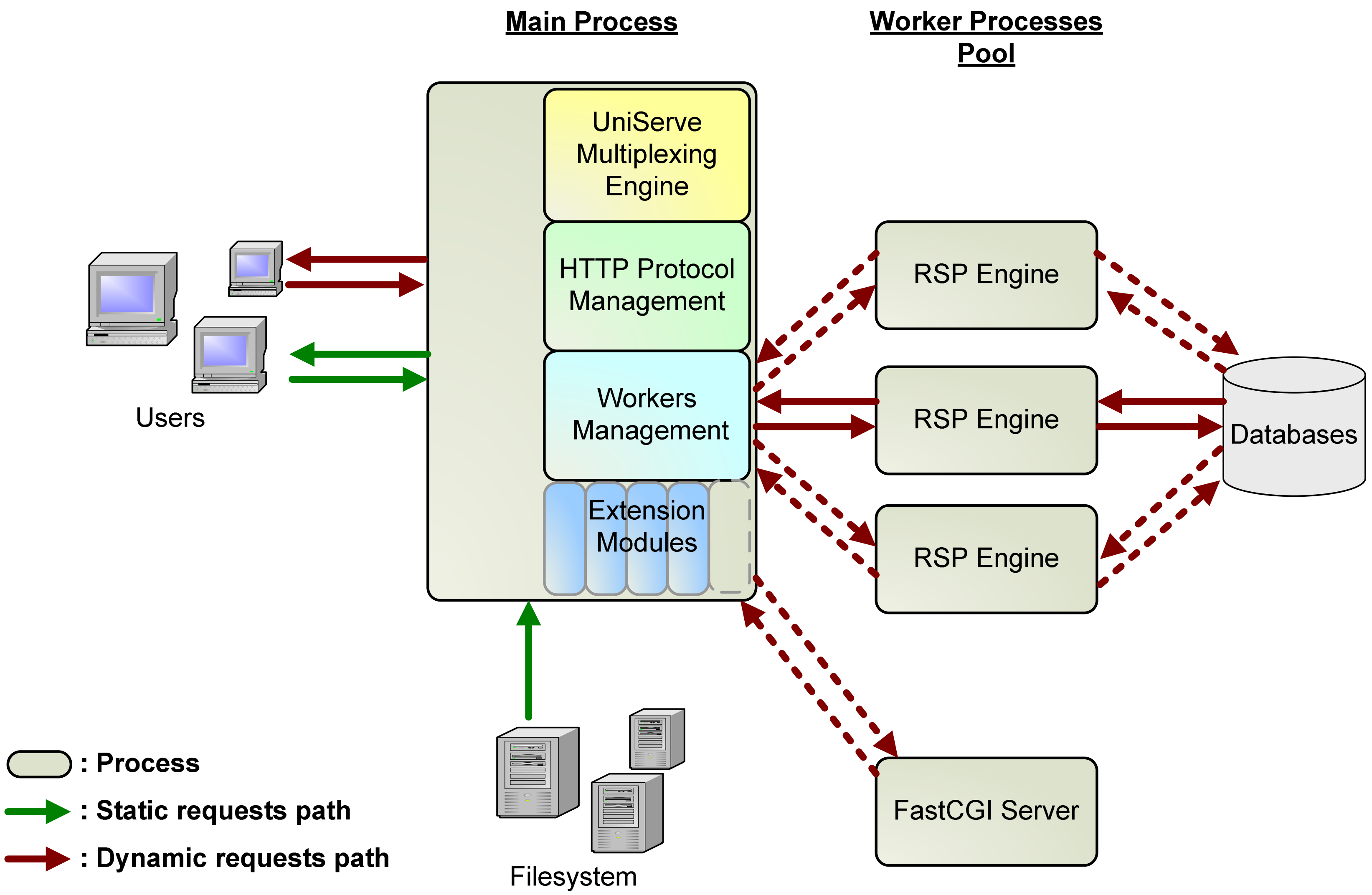
[illegible]

Cheyenne Web Server: Introduction

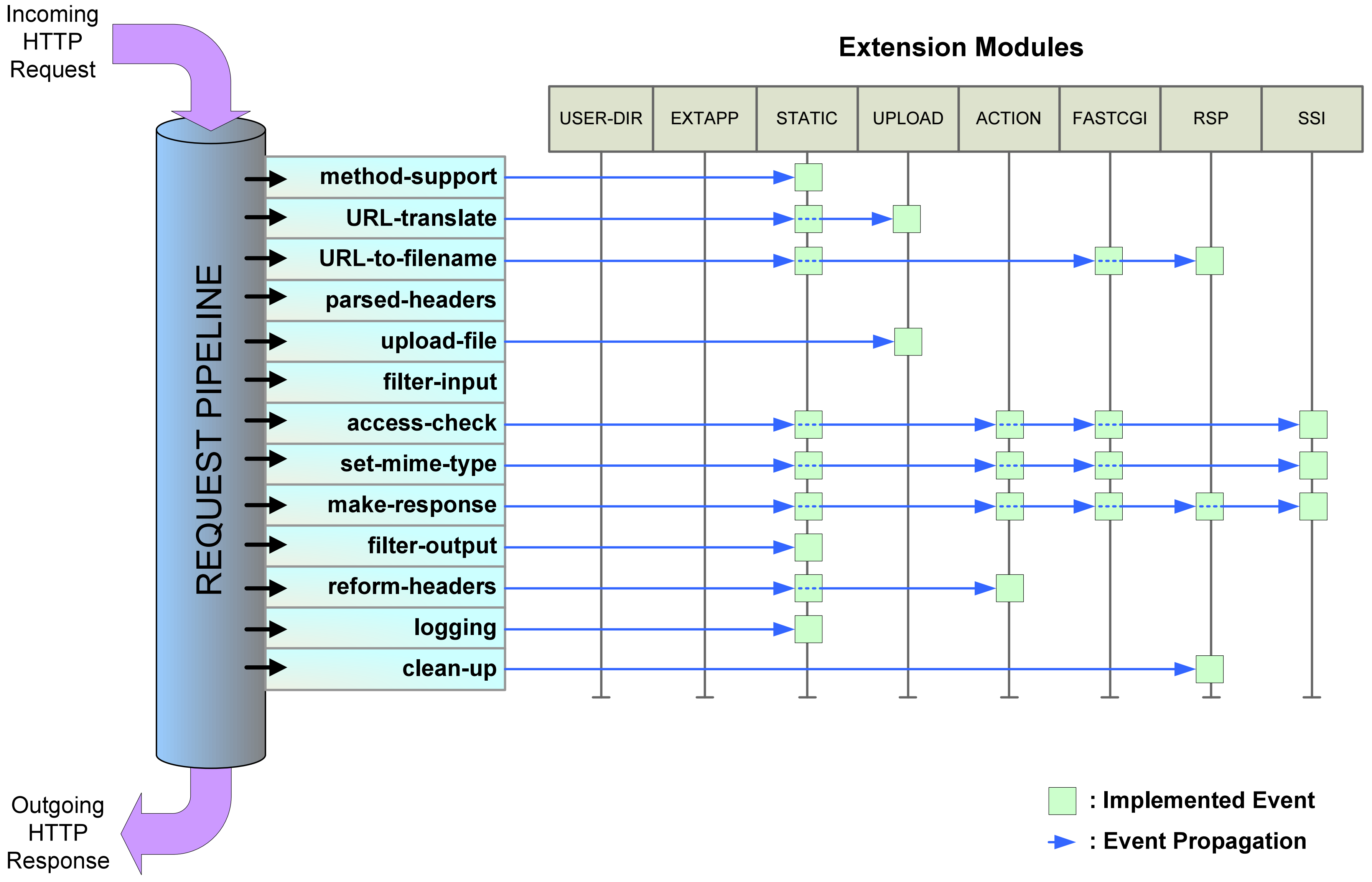
- **Why making yet-another-web-server?**
  - Provide a native container for REBOL-based web applications
  - Small, efficient, cross-platform, easy to deploy, easy to extend
  - Stress-test REBOL

- A few facts…
  - Binary is ~500KB (~90KB for Cheyenne code, the rest for REBOL interpreter)
  - No installer required
  - Fully open source (BSD), hosted on Google Code
  - Modular architecture (mods, concurrent handlers,…)
  - Key server technologies supported: FastCGI, WebSockets, …
  - Powered by a "fast" asynchronous I/O engine: UniServe
  - Performs as good or better than other interpreted Web Servers (Mongrel,…)
  - Used in production by several companies: Softinnov, RT, Synapse EMR,…

Cheyenne Architecture: Overview



Cheyenne Architecture: The Request Pipeline



Cheyenne: Content Serving

- **Configuring**
  - Configuration file with expandable dialect
  - Virtual Hosts supported
  - *GUI web panel to come for v1.0*

- Serving
  - Static content: any size up to 2GB files, < 16KB files are memory cached
  - Dynamic content: SSI, CGI, RSP (REBOL Server Pages), …
  - Content from external servers: FastCGI servers (e.g. PHP)

Cheyenne: RSP Scripting

- **Basic Concepts**
  - Templating system : <%...%>, <%=…%>
  - Rich API (Request, Response, Session, …)
  - Fast and Concurrent execution (pre-compiled + memory cached + worker processes)

- Session Handling
  - Session key passed by Cookie / in URL / in POST data
  - Session context: store, change, remove session-local data
  - Manual vs automatic session management
    - Manual => session/start, session/stop
    - Automatic => add a webapp entry in config file

Cheyenne: Web Apps

- **Application container**
  - Private / Protected / Public file hierarchy
  - Event hooks:
    - on-application-start, on-application-end
    - on-session-start, on-session-end
    - on-page-start, on-page-end
  - Database management abstraction layer (no need to open, close, reopen)
  - Localization basic support
    - In REBOL code, using SAY function
    - In templates, using #[text] literal form
- Application services
  - Session handling
  - User authentication state with redirection to login page

Cheyenne: Other Features…

- Maintenance using external console
  - Access all internal live code from a REBOL console
  - Make hot-patches!
- Embedded mode
  - Include Cheyenne inside any REBOL app, even graphical ones
  - Serve dynamic content from your application directly (API provided)
- Upload API for clients
  - Get file upload progress information from server (uploaded / remaining size)
- Experimental built-in services
  - CRON-like scheduler engine with its own DSL
  - SMTP server (MTA agent, currently limited to 8-bit support)
- Windows NT Services support
  - Switching from User to Service mode from systray icon in one click