

Horde Leader, a Framework to Build Cluster Aware Erlang Web Administration Console

November 9, 2006

Jérôme Sautret



CONTENTS

- **History and status of the project**
- **Purpose**
- **Architecture**
 - Core Library
 - Monitoring Service
 - Plugins
- **Demonstration**
- **Non-erlang applications**
- **Roadmap**

- **Team Leader, a web console for ejabberd**
 - Currently usable
 - Some ejabberd specific code in the core application



- **Horde Leader, a framework to build web consoles**
 - Generic
 - To be release in Open Source
 - Work in progress

■ A Framework to build Web Consoles

- **Monitoring**
- Control
- Configuration
- Statistics
- ...

The screenshot displays the TeamLeader web console interface. At the top, there is a header with the TeamLeader logo and a user name field labeled "User Name (TODO)" with a "logout" button. Below the header are navigation tabs for "Cluster", "Statistics", "Users & groups", and "Configuration". The main content area is divided into two sections. On the left, a sidebar shows a tree view under "Cluster" with sub-items "Cluster" and "Nodes", where "Nodes" is expanded to show "ejabberd2@zero" and "ejabberd@zero". The main area is titled "nodes" and contains a sub-section "Cluster nodes" with a table of node information.

node id	state	online users	started	uptime	CPU time
ejabberd2@zero		0	2006/11/02 15:49:47	0.00	0.00
ejabberd@zero		0	2006/11/02 15:41:21	505.83	0.78

Team Leader, a Web console for ejabberd, the Erlang Jabber server, Copyright © 2006, [Process-one](#).

■ A Framework to build Web Consoles

- Monitoring
- **Control**
- Configuration
- Statistics
- ...

The screenshot shows the TeamLeader web console interface. At the top left is the TeamLeader logo. To the right, there is a user name field containing "User Name (TODO)" and a "logout" button. Below the header is a navigation bar with tabs for "Cluster", "Statistics", "Users & groups", and "Configuration". The "Cluster" tab is selected. On the left side, under the "Cluster" heading, there is a sub-section for "Cluster" containing a "Nodes" list. The list includes "ejabberd2@zero" and "ejabberd@zero". Under "ejabberd@zero", there are several actions: "Stop", "Restart", "Reopen the log", "Database table", and "Statistics". On the right side, the status for "ejabberd@zero" is displayed, showing it is currently "running". A list of statistics is provided: "Started on 2006/11/02 15:41:21 (859.092 seconds ago)", "Client connections0", "PID of the Erlang VM8329", "CPU Usage0.80", and "Cannot get memory occupation for this node". At the bottom of the page, there is a footer that reads: "Team Leader, a Web console for ejabberd, the Erlang Jabber server, Copyright © 2006, Process-one."

■ A Framework to build Web Consoles

- Monitoring
- Control
- **Configuration**
- Statistics
- ...

Configuration

- **Configuration**
 - Global
 - Virtual Hosts
 - Statistics

global

Virtual Hosts

- ◆ zero
- ◆ localhost

Delete selected Virtual Hosts

Add a new virtual host Add

User self-registration

Allowed Denied

Envoyer

Administrators

- ◆ jerome

Remove the administrator role of selected users

Give administrator role to user id

Add

Team Leader, a Web console for ejabberd, the Erlang [Jabber server](#), Copyright © 2006, [Process-one](#).

■ A Framework to build Web Consoles

- Monitoring
- Control
- Configuration
- **Statistics**
- ...



■ **A Framework to build Web Consoles**

- Monitoring
- Control
- Configuration
- Statistics
- ...

■ **Plugin based**

- All the console logic is provided by plugins
- Plugin can be added or removed to customize or enhance the console
- No deployment on remote application node

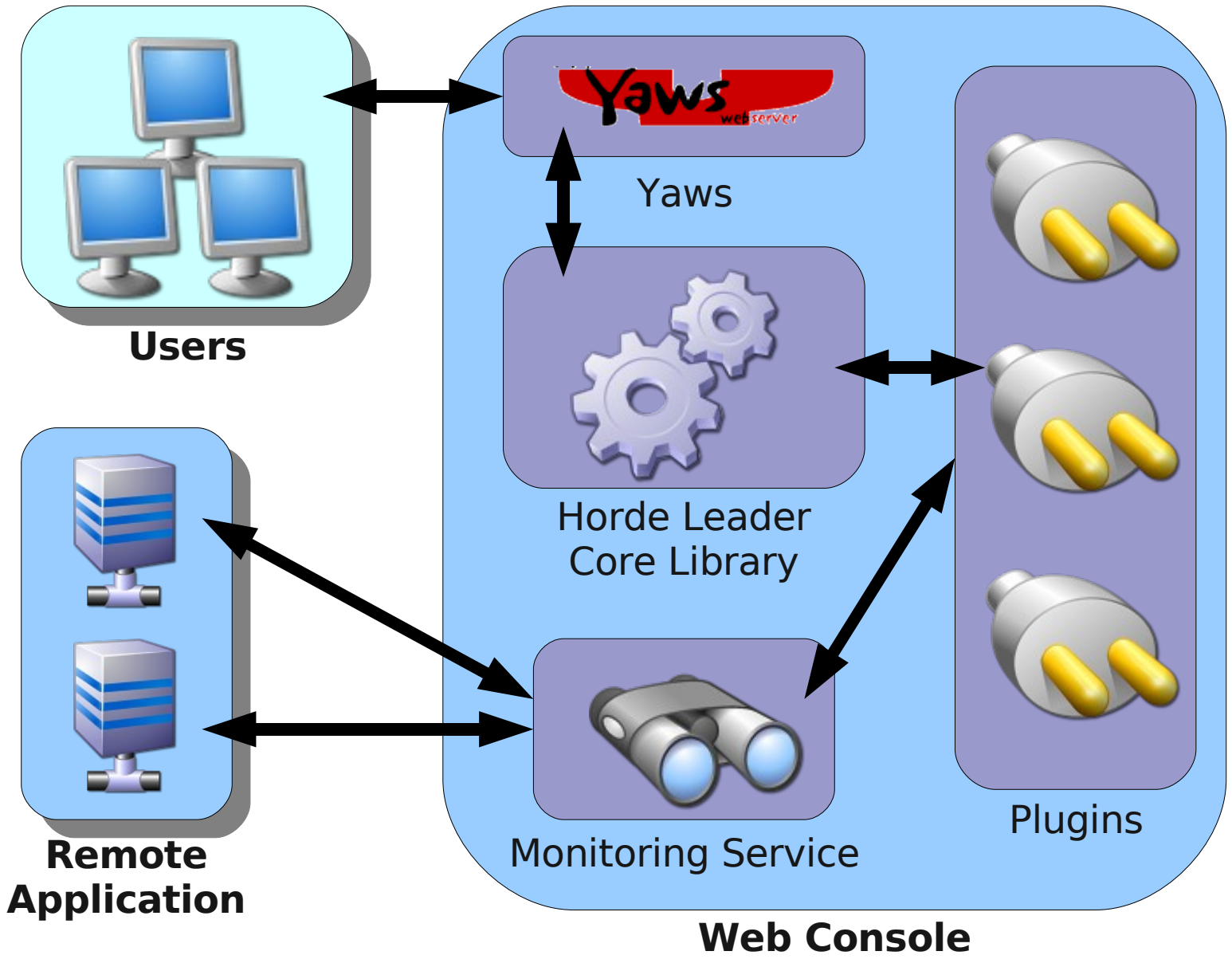
■ **Cluster aware**

- The console can handle an application running on several nodes

■ **XHTML/CSS**

- no need for templates

Architecture

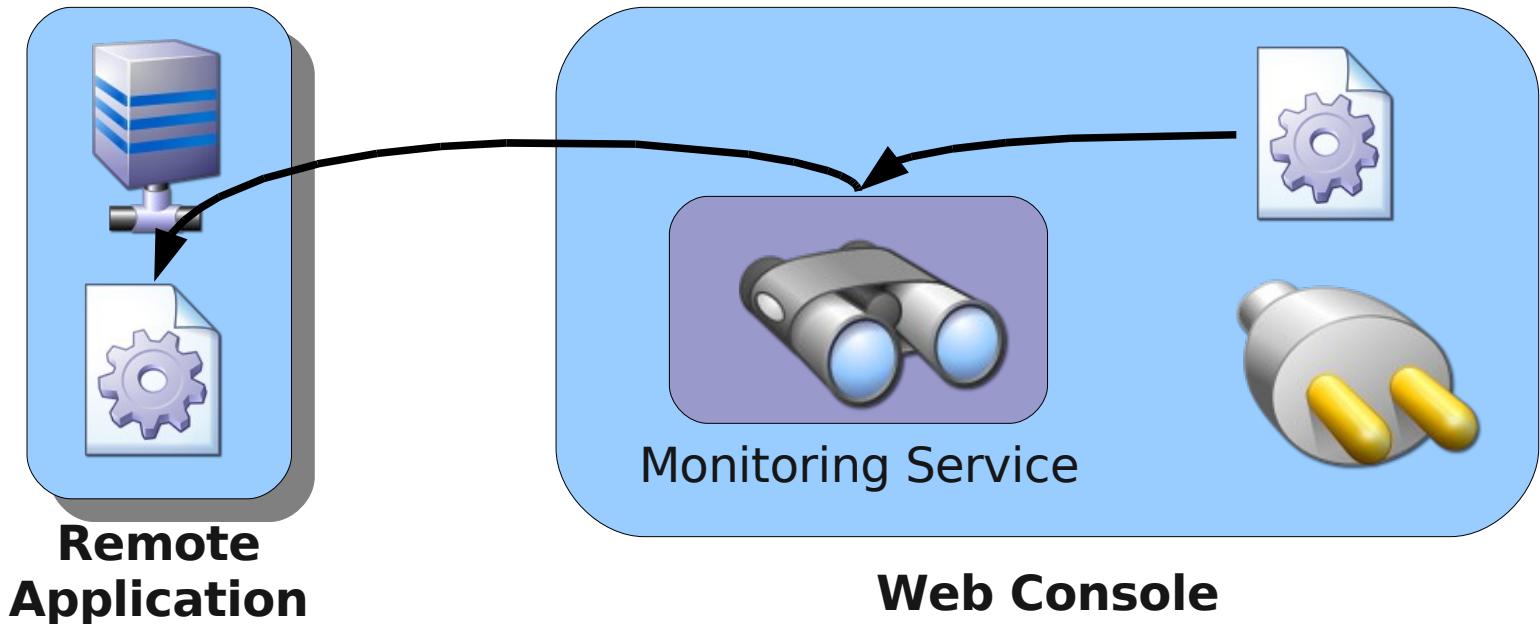


■ **Core Library**

- Authentication
- URL -> Plugin function
- Page display
 - **Header**
 - **Footer**
 - **Navigation Menus**
- **Various Helper Functions**
 - **Widgets**
 - **Lists of items**
 - **Information/Error messages**
 - ...

■ Monitoring Service

- Monitor remote node states
- Store node states in database
- Communicate with remote nodes using RPC
- Send module to remote nodes



■ Plugins

- Handle page content
- Manage all the console logic

■ What is a Plugin

- Erlang module
- Adds entries in the navigation tree

```
?INIT(_A) ->
    #plugin{
        id=?MODULE_STRING,
        menu=[
            ?MENU_ITEM([], "users", "Users & groups", 20, index),
            ?MENU_ITEM(["users"], "all", "All users", 10, all),
            ?MENU_ITEM(["users"], "online",
                "Online users", 20, online),
            ?MENU_ITEM(["users"], "create",
                "Create", 30, create),
            ?MENU_ITEM(["users"], "delete",
                "Delete", 40, delete),
            ?MENU_ITEM(["users"], "search",
                "Search", 50, search),
            ?MENU_HIDDEN_ITEM(["users"], "user", user)
        ]}.
}
```

■ How a Plugin display a page

- The Core Library handle the generic parts
- The plugin returns only the content as Yaws HTML

```
adduser(A) ->
  team_leader:get_common_page(
    A,
    {ehtml,
     {form, [{method, "post"}, {action, ?ROOT_URL(A)++"users/create"},
            {class, "form"}]},
     {fieldset, [],
      [?ASK_QUIT_CONFIRMATION,
       {legend, [], ?TXT("Add a new jabber user")},
       {label, [{for, "userid"}, {class, "required"}],
        {title, ?TXT("User id, without the @domain")}],
        ?TXT("User id")},
       {input, [{type, "text"}, {name, "userid"}, {class, "field"}], []},
       {label, [{for, "password"}, {class, "required"}],
        {title, ?TXT("Password of the new user")}],
        ?TXT("Password")},
       {input, [{type, "text"}, {name, "password"}, {class, "field"}],
        []},
       {input, [{type, "submit"}, {name, "submit"}, {class, "button"}],
        []}
      ]}
    )}
  ).
```

Demonstration of Team Leader (Horde Leader + ejabberd specific plugins)

- **A console for non-erlang applications ?**
 - Console/Application communication uses erlang RPC
 - Remote Module can be send dynamically
- **Prerequisites**
 - Remote module able to communicate with application
 - Read logs
 - Edit configuration files
 - System calls
 - Erlang installed on all remote nodes
 - Erlang VM running on all remote nodes

```
su -c "erl -sname hl_agent >/dev/null 2>&1" - horde_leader <<EOF &
erlang:set_cookie(node(), 'secret').
Ref = make_ref().
receive
  Ref ->
    ok
end.
EOF
```

- **Team Leader**
 - Work in progress
- **Remove all ejabberd specific code from the Core Library and the Monitoring Service**
 - Work in progress
- **Write some generic documentation**
 - When the API is stable
- **Publish the code on <http://forge.process-one.net>**
 - When all ejabberd specific code is gone
- **Write more plugins...**