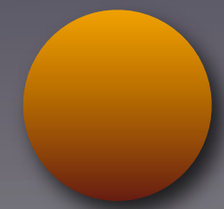
Unifying SIP and WEB Worlds with LUA



FOSDEM 2011

Brussels, Belgium

Daniel-Constantin Mierla

Co-Founder Kamailio SIP Server Project

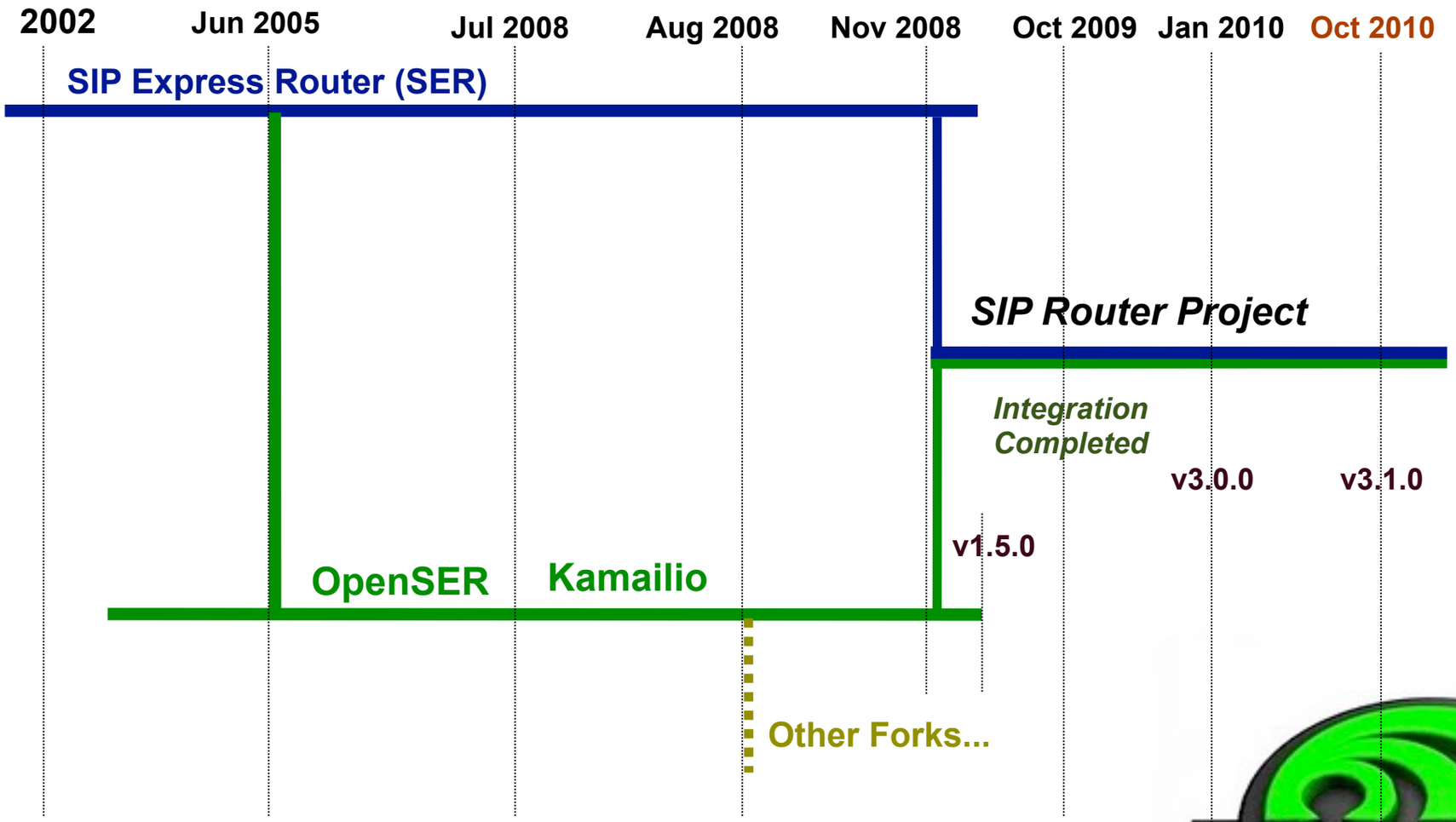
www.asipto.com

It is all about love

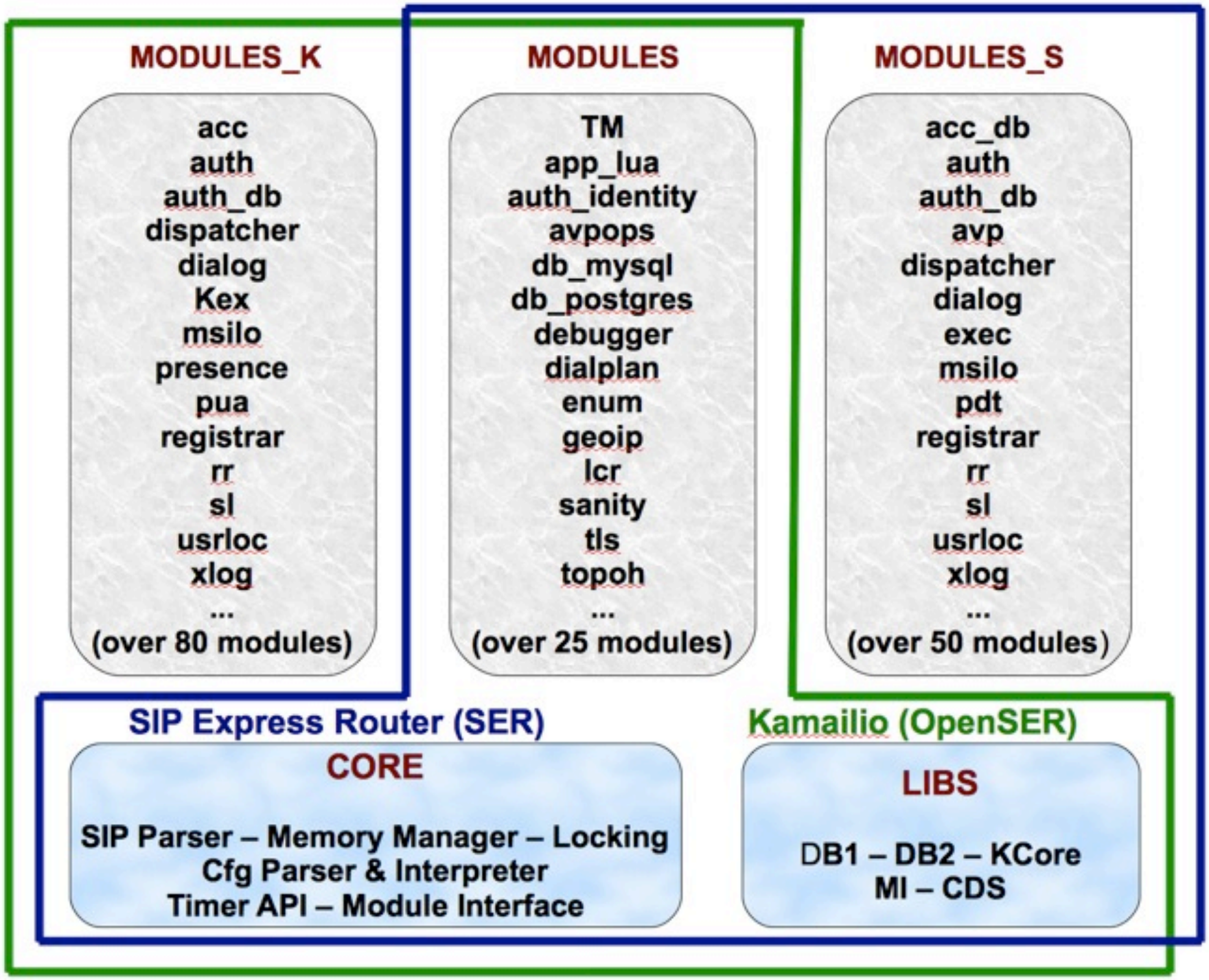
Wine, moon ... and you

... hacking SIP and Web

A bit of Kamailio project history



Kamailio and SER



New in Kamailio v3.1.x

Flexibility → Maintenance → Performance → Features

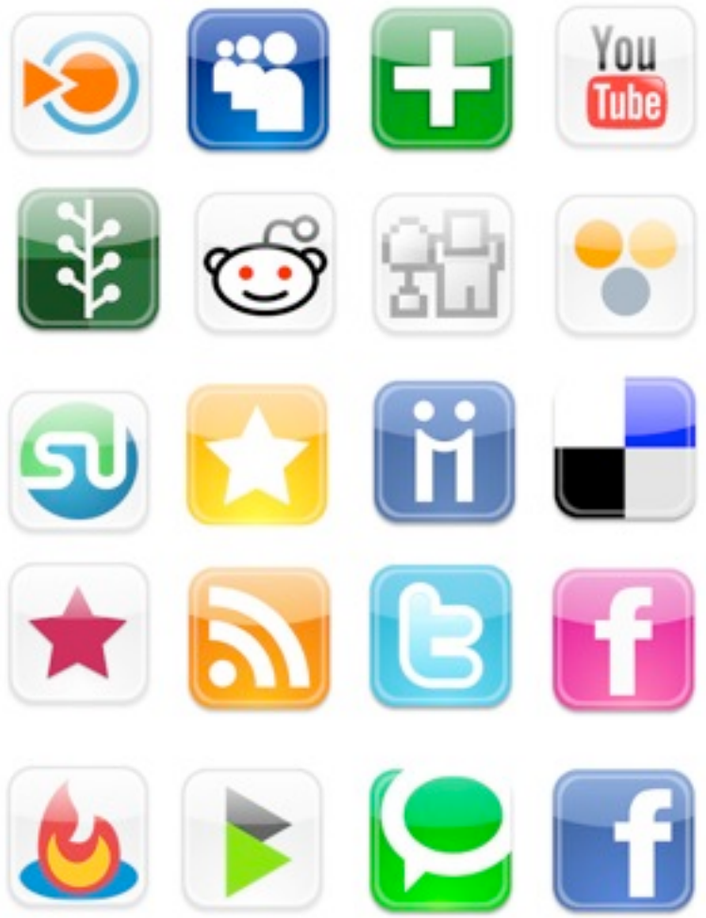
- **Embedded Lua**
- Embedded Python
- Extended preprocessor directive
  - #!define
  - #!subst
- New variables

- Interactive config debugger
  - step-by-step execution
  - execution trace
- xlog enhan's
  - print cfg line
- k&s modules integration

- Asynchronous TLS
- UDP raw sockets
- Multi-homed improvements
- Load balancing
  - weight
  - call load
- Traffic shaping

- GeoIP API
- Registration to remote servers
- Reason header for Cancel
- **Embedded HTTP** & XCAP servers
- Cfg tree cashing & message queue systems

HTTP and Web 2.0



Interaction:

- access Kamailio SIP world via HTTP
- access Web world from Kamailio via HTTP

HTTP vs SIP

```
GET /stats HTTP/1.1.
Host: 127.0.0.1:5060.
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.12) Gecko/20101206 Ubuntu/10.04 (lucid) Firefox/3.6.12.
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8.
Accept-Language: en-us,en;q=0.5.
Accept-Encoding: gzip,deflate.
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7.
Keep-Alive: 115.
Connection: keep-alive.
.
```

```
REGISTER sip:192.168.56.103 SIP/2.0.
Via: SIP/2.0/UDP 192.168.56.103:5062;branch=z9hG4bK2F912227.
CSeq: 7417 REGISTER.
To: "102" <sip:102@192.168.56.103>.
Expires: 900.
From: "102" <sip:102@192.168.56.103>.
Call-ID: 81570709@192.168.56.103.
Content-Length: 0.
User-Agent: kphone/4.2.
Event: registration.
Allow-Events: presence.
Contact: "102" <sip:102@192.168.56.103:5062;transport=udp>;methods="INVITE, MESSAGE, OPTIONS, BYE, CANCEL, NOTIFY,ACK, REFER".
.
```

Reuse SIP parser to handle HTTP requests!!!

XHTTP module

Characteristics

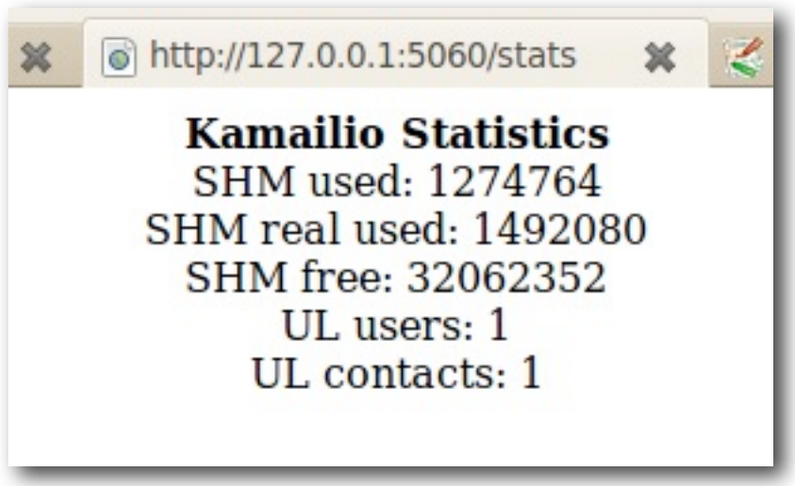
- handle HTTP requests sent to Kamailio
  - execute **event_route[xhttp:request]**
  - reuses the SIP parser from Kamailio core
  - all functions available for SIP requests can be used for HTTP requests, e.g.:
    - user authentication (www digest)
    - IP authentication
  - no forwarding support for HTTP request, just replying
  - no dependency on external libraries
- compatibility
  - http/1.0
  - http/1.1
- transport layers - all supported by Kamailio core
  - HTTP over TCP
  - HTTP over TLS (aka HTTPS)
  - HTTP over UDP (yeah!!!)
  - HTTP over SCTP (gosh!!!)

Browse the Kamailio Instance

```
event_route[xhttp:request] {
        set_reply_close();
        set_reply_no_connect();
        if(method!="GET")
        {
                xhttp_reply("404", "Not found", "text/html", "<html><body>Method not"
                                        " supported ($rm from $si:$sp)</body></html>");
                exit;
        }
#!ifdef WITH_XHTTPAUTH
        if (!www_authorize("xhttp", "subscriber"))
        {
                www_challenge("xhttp", "0");
                exit;
        }
#!endif
        if($hu=~"^/stats")
        {
                xhttp_reply("200", "OK", "text/html", "<html><body>"
                                        "<div align='center'>"
                                        "<strong>Kamailio Statistics</strong><br />"
                                        "SHM used: $stat(used_size)<br />"
                                        "SHM real used: $stat(real_used_size)<br />"
                                        "SHM free: $stat(free_size)<br />"
                                        "UL users: $stat(location-users)<br />"
                                        "UL contacts: $stat(location-contacts)<br />"
                                        "</div></body></html>");
                exit;
        }

        # no handler for URL
        xhttp_reply("404", "Not found", "text/html", "<html><body>URL not"
                                        " found ($hu from $si:$sp)</body></html>");
        exit;
}
```

Browse the Kamailio Instance

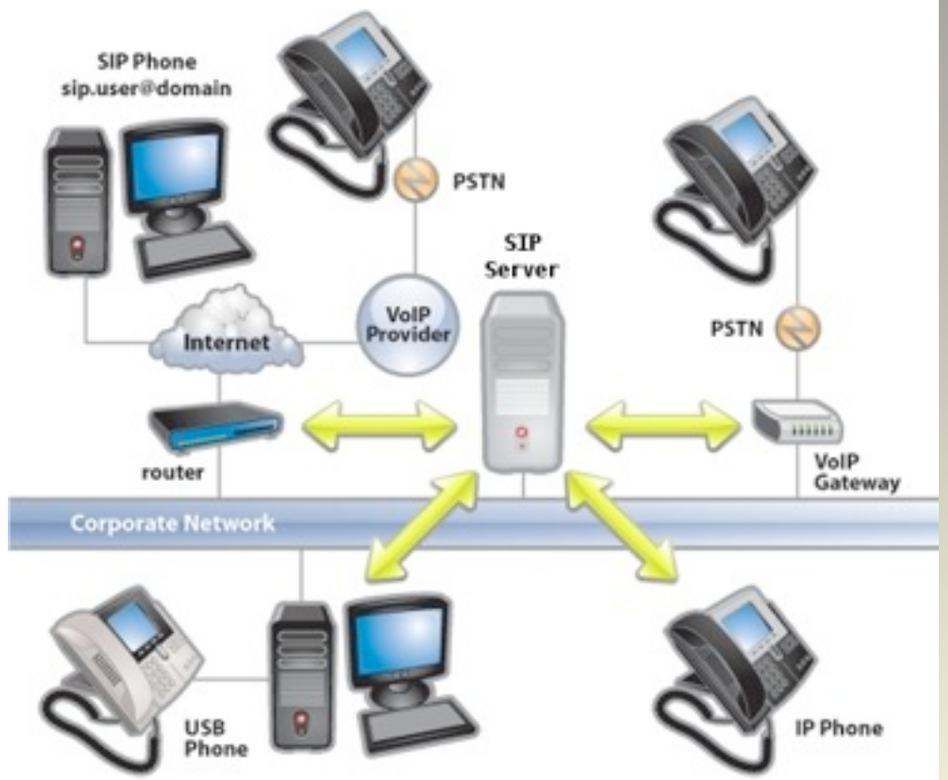
```
HTTP/1.1 200 OK.
Via: SIP/2.0/TCP 127.0.0.1:58750.
Content-Type: text/html.
Server: kamailio (3.2.0-dev2 (i386/linux)).
Content-Length: 206.
.
<html><body><div align='center'><strong>Kamailio Statistics</strong><br />SHM used: 1274764<br />SHM real
used: 1492080<br />SHM free: 32062352<br />UL users: 1<br />UL contacts: 1<br /></div></body></html>
```



Kamailio and SIP routing

Where we stand today?

- flexible configuration language, still limited pretty much to SIP only
- the power is in hands (and brain) of administrator
- SIP specific extensions added mainly by writing C modules
- for the rest: **Lua**, Perl, Python and Java



Lua

What is Lua?

- Lua is a powerful, fast, lightweight, embeddable scripting language.
- Lua combines simple procedural syntax with powerful data description constructs based on associative arrays and extensible semantics. Lua is dynamically typed, runs by interpreting bytecode for a register-based virtual machine, and has automatic memory management with incremental garbage collection, making it ideal for configuration, scripting, and rapid prototyping.
- http://www.lua.org

Lua

Why choose Lua?

- Lua has been used in many industrial applications (e.g., Adobe's Photoshop Lightroom), with an emphasis on embedded systems (e.g., the Ginga middleware for digital TV in Brazil) and games (e.g., World of Warcraft). Lua is currently the leading scripting language in games..

- Lua is supported by other telephony engines, e.g., Asterisk and FreeSWITCH

- http://www.lua.org

Lua

More about Lua?

- is fast
- is portable
- is embeddable
- is powerful (but simple)
- is small
- is free

- http://www.lua.org

Kamailio and Lua

Why together?

- need to integrate with other communication platforms
- it is about realtime communication, so being fast is important
- routing logic may require non-SIP related operations
- all characteristics of Lua indicates it a very good candidate

Kamailio and Lua

How does it work?

- load module: app_lua
- Lua scripts can be loaded at startup or on-the-fly at runtime
- entire Lua scripts can be executed or just functions from the scripts
- Lua scripts have access to SIP messages and config variables
- Lua scripts can return back config variables
- Lua scripts can execute functions from config file

http://kamailio.org/docs/modules/devel/modules/app_lua.html

Kamailio Lua API

Lua packages

- package **sr**
  - access to Kamailio core functions
  - method to execute functions exported by modules
  - container for other sub-packages
- package **sr.hdr**
  - header manipulation
  - get, add, remove
- package **sr.pv**
  - pseudo-variables manipulation
  - access to all pseudo-variables - hundreds of them
- package **sr.sl**
  - exports internal sl module API (stateless reply)
- many more packages in devel version
  - including direct access to **xhttp** module

http://sip-router.org/wiki/api/lua/devel

Browse the Kamailio Instance

```
function htmlStats()
        htmlBody = [[
<html><head><title>Kamailio SIP Server</title>
<style type="text/css">
#tstats {
  font-family:Arial, Helvetica, sans-serif;
  width:50%;
  border-collapse:collapse;
}
#tstats td, #tstats th {
  font-size:1em;
  border:1px solid #98bf21;
  padding:3px 7px 2px 7px;
}
#tstats th {
  font-size:1.1em;
  text-align:left;
  padding-top:5px;
  padding-bottom:4px;
  background-color:#A7C942;
  color:#ffffff;
}
#tstats tr.alt td {
  color:#000000;
  background-color:#EAF2D3;
}
</style>
</head><body>
        <div align='center'>
        <strong>Kamailio Statistics</strong><br />
        <table width='50%' id='tstats'>
                <tr><th>Description</th><th>Value</th></
tr>
                <tr><td>SHM used</td><td>
]]
```

HTTP Reply example

Lua script:

```
                .. sr.pv.get("$stat(used_size)")
                .. "</td></tr>"
                .. "<tr class='alt'><td>SHM real used</td><td>"
                .. sr.pv.get("$stat(real_used_size)")
                .. "</td></tr>"
                .. "<tr><td>SHM free</td><td>"
                .. sr.pv.get("$stat(free_size)")
                .. "</td></tr>"
                .. "<tr class='alt'><td>UL users</td><td>"
                .. sr.pv.get("$stat(location-users)")
                .. "</td></tr>"
                .. "<tr><td>UL contacts</td><td>"
                .. sr.pv.get("$stat(location-contacts)")
                .. "</td></tr>"
                .. "<tr class='alt'><td>Received reqs</td><td>"
                .. sr.pv.get("$stat(rcv_requests)")
                ..
[[              </td></tr>
        </table></div></body></html>]];

        sr.xhttp.reply(200, "OK", "text/html", htmlBody);
end
```

Browse the Kamailio Instance

HTTP Reply example

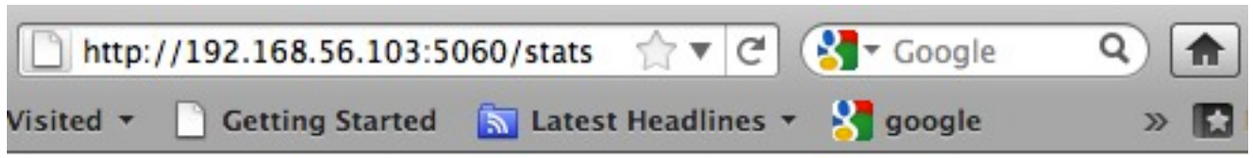
Kamailio config:

```
event_route[xhttp:request] {
        set_reply_close();
        set_reply_no_connect();
        if(method!="GET")
        {
                xhttp_reply("404", "Not found", "text/html", "<html><body>Method not"
                                        " supported ($rm from $si:$sp)</body></html>");
                exit;
        }
#!ifdef WITH_XHTTPAUTH
        if (!www_authorize("xhttp", "subscriber"))
        {
                www_challenge("xhttp", "0");
                exit;
        }
#!endif
        if($hu=~"^/stats")
        {
                if(!lua_runstring("htmlStats()"))
                {
                        xdbg("failed to send reply from Lua!\n");
                }
                exit;
        }

        # no handler for URL
        xhttp_reply("404", "Not found", "text/html", "<html><body>URL not"
                                        " found ($hu from $si:$sp)</body></html>");
        exit;
}
```

Browse the Kamailio Instance

HTTP Reply example

Result:



Kamailio Statistics

Description	Value
SHM used	1275492
SHM real used	1492520
SHM free	32061912
UL users	0
UL contacts	0
Received reqs	71

HTTP Query from Config File

UTILS Module

- ***http_query(url, result)***

```
http_query("http://10.10.10.10/allow.php?userid=$fU","$var(result)");
```

http://kamailio.org/docs/modules/stable/modules/utils.html

SIPWEET

twitter

SIPWEET

Send tweets on SIP events

- missed call notification
- instant messaging
- reminders

Monitor tweets for SIP services

- tweet-to-dial
- reminder setup
- scheduled calls
- profile update

SIPWEET

Design

- Lua twitter library
- Twitter operation is an HTTP request
  - can take some time to be processed
  - we cannot afford that when processing SIP signaling
  - solution: use asynchronous processing
    - config file message queue
    - dedicated process for twitter operations
- Kamailio modules
  - app_lua
  - mqueue
  - rtimer
  - sqlops
- Sample implementation
  - notification of a missed call
  - use of Twitter direct message

SIPWEET

Config

```
loadmodule "app_lua.so"
loadmodule "rtimer.so"
loadmodule "sqlops.so"
loadmodule “mqueue.so”

# ----- app_lua -----
modparam("app_lua", "load",
        "/usr/local/etc/kamailio/lua/sipweet.lua")

# ----- rtimer -----
modparam("rtimer", "timer",
        "name=sipweet;interval=10;mode=1;")
modparam("rtimer", "exec",
        "timer=sipweet;route=SIPWEET;")

# ----- sqlops -----
modparam("sqlops","sqlcon",
        "ca=>mysql://openser:openserrw@localhost/openser")

# ----- mqueue -----
modparam("mqueue", "mqueue", "name=sipweet")
```

SIPWEET

Config

```
# Twitter routing
route[SIPWEET] {
        # consume tweeties
        while(mq_fetch("sipweet"))
        {
                xlog("Tweeting to $mqk(sipweet) [[$mqv(sipweet)]]\n");

                # get twitter user
                sql_query("ca",
                  "select twuser from sipweetusers where sipuser='$mqk(sipweet)'",
                  "ra");
                if($dbr(ra=>rows)>0)
                {
                        $var(twuser) = $dbr(ra=>[0,0]);
                        $var(twmsg) = $mqv(sipweet);
                        if(!lua_runstring("sipweetdm([[$var(twuser)]], [[$var(twmsg)]])"))
                        {
                                xdbg("failed to send dm to: $mqk(sipweet) - $var(twuser)!\n");
                        }
                }
        }
}
```

SIPWEET

Config

```
# Twitees queuing
route[TWQUEUE] {
        if(!is_method("INVITE"))
                return;
        mq_add("sipweet", "$rU", "Missed call from $fU ($Tf)");
}

route {
        ...
        if(!lookup(“location”)) {
                route(TWQUEUE);
                t_newtran();
                t_reply("404", "Not Found");
                exit;
        }
        ...
}
```

SIPWEET

Database table

```
CREATE TABLE `sipweetusers` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `twuser` varchar(64) NOT NULL,
  `sipuser` varchar(64) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY (`twuser`),
  UNIQUE KEY (`sipuser`)
);
```

SIPWEET

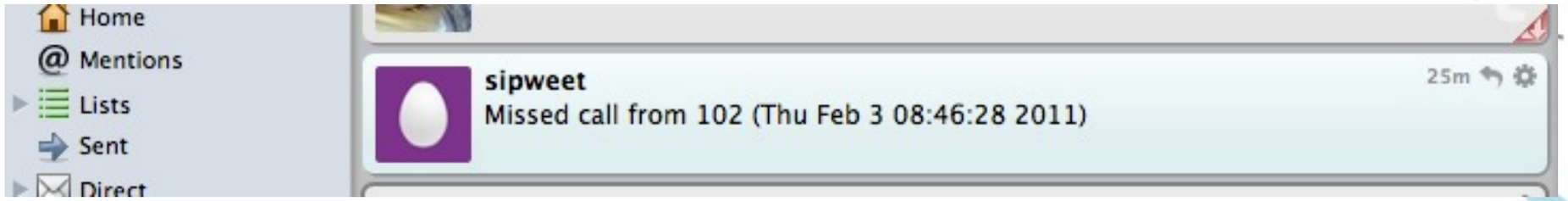
Lua script

```
-- SIPweet

-- loading module
require("twitter")

local initialized = 0
local mytw

function sipweetinit()
    if initialized == 0 then
            mytw = twitter.client("Consumer Key",
                    "Consumer Secret",
                    "OAuth Token",
                    "OAuth Token Secret");
        initialized = 1
    end
end

function sipweetdm(userid, message)
    sipweetinit()
        mytw:sendMessage{ user = userid, text = message }
end
```

SIPWEET

And the messages goes ...

- nice and quick to your twitter client



New since Kamailio v3.1.x

SIP:Provider - http://www.sipwise.com/products/spce/

* complete VoIP servicing platform using Kamailio for SIP routing
* administration interface and user portal
* ready to roll-out open source Community Edition
* easy to install with DEB packages - images for VMWare and VirtualBox

New since Kamailio v3.1.x

New since Kamailio v3.1.x

- Many native extensions to Lua
- Private memory troubleshooting
- RLS polishing and Presence server scalability enhancements
- cfg framework group - multivalues
- partitioned user location services
- IMS extensions based on OpenIMSCore
  - *approx. 10 new modules*
- new functionalities to many modules: textops, kex, acc, dialog, dispatcher, utils ...
- Planned
  - merge duplicated modules
  - more presence server optimizations and scalability
  - upgrades to RPC and atomic counter stats
  - new features ...

Concluding

- You can do HTTP GET requests from Kamailio config file
  - utils module
- You can access Kamailio internals via HTTP
  - xhttp module
- Using Embedded Lua is easier to build rich SIP-Web services
  - Lua scripts are fast executed

Questions?

Contact

- **Daniel-Constantin Mierla**
  - **twitter: miconda**
  - **daniel@asipto.com**
  - **http://www.asipto.com**
  - **http://www.kamailio.org**

Meet us in Barcelona

- Developer Training, Feb 10-11, 2011
- Social Networking event, Feb 10, 2011