September 7-8, 2022

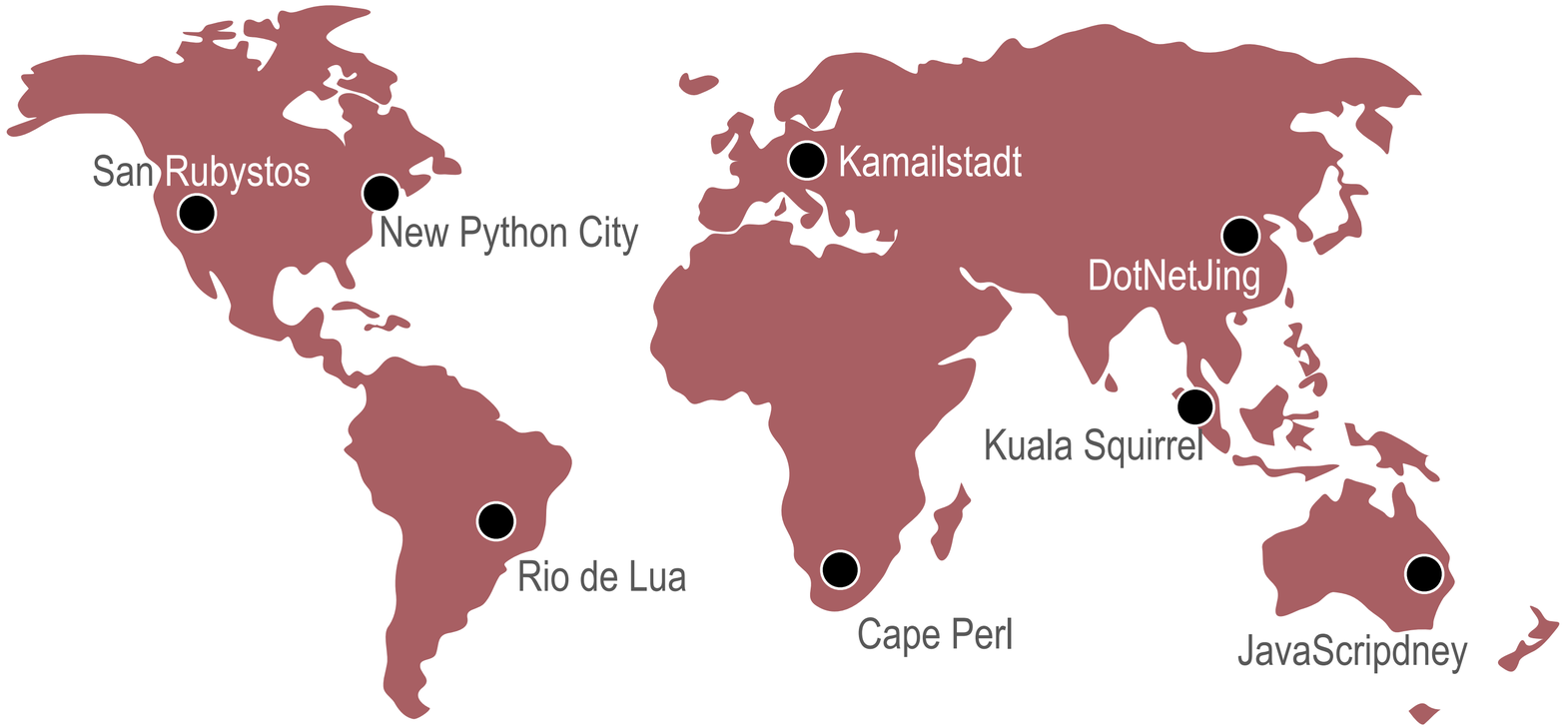
KEMI Scripting

Full Control Of SIP Message Content

KEMI Framework

Empowering Kamailio Since 2016

- introduced in Kamailio v5.0.0
  - use other scripting languages for writing SIP routing logic (route blocks)
  - exports a function one time and becomes available in embedded interpreters
  - ***allows reloading routing script without restart***
  - ***enables the use of an extensive number of extensions available in the scripting languages***
  - ***lua, python2, python3, javascript, ruby, squirrel***



NATIVE & KEMI SCRIPTING

- Writing configuration file SIP routing logic in different scripting languages
- Kamailio exports its functions via **KSR** module (library/object): KSR.func() or KSR.mod.func()
- Equivalent of routing blocks
  - request_route { … } — ksr_request_route()
  - reply_route { … } — ksr_reply_route()
  - onsend_route { … } — ksr_onsend_route()
  - branch_route[ID], onreply_route[ID], failure_route[ID] — function names that are set as callbacks via KSR.tm.t_on_branch(…), KSR.tm.t_on_reply(…) and KSR.tm.t_on_failure(…)
  - event_route[tm:branch-failure:id] — function name that is set as callback via KSR.tm.t_on_branch_failure(…)
  - event_route[module:event-name] — function name is set via module parameter **event_callback** or **evcb_xyz** (e.g., evcb_reply_out)

```
route[NATDETECT] {
    if (nat_uac_test("19")) {
        if (is_method("REGISTER")) {
            fix_nated_register();
        } else {
            if(is_first_hop()) {
                set_contact_alias();
            }
        }
        setflag(FLT_NATS);
    }
    return;
}
```

```
function ksr_route_natdetect()
    if KSR.nathelper.nat_uac_test(19)>0 then
        if KSR.is_REGISTER() then
            KSR.nathelper.fix_nated_register();
        elseif KSR.siputils.is_first_hop()>0 then
            KSR.nathelper.set_contact_alias();
        end
        KSR.setflag(FLT_NATS);
    end
    return 1;
end
```

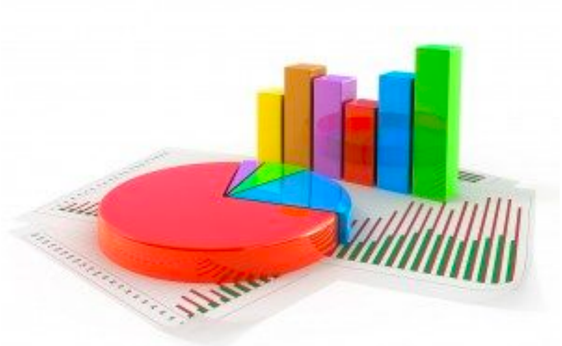
https://github.com/kamailio/kamailio/blob/master/misc/examples/kemi/kamailio-basic-kemi-lua.lua

https://github.com/kamailio/kamailio/blob/master/misc/examples/kemi/

http://kamailio.org/docs/tutorials/devel/kamailio-kemi-framework/

KEMI Performances For v5.2

- **Done in November 2018, just before the release of Kamailo v5.2**
  - The focus was to compare the execution time of request_route {} from native kamailio.cfg with the equivalent KEMI callback functions for processing SIP REGISTER requests with user authetication
    - Done for Lua and Python
- **Testing Environment**
  - Server: Intel NUC 7i7DNHE; CPU: I7-8650U @ 1.90GHz; CPU Cores: 4; CPU Threads: 8; Memory: 16GB
  - Operating system: Debian Testing
  - SIP traffic tool: sipp
- **Testing metrics**
  - Each iteration was done with sending 20 000 requests
  - For each iteration was collected
    - cnt - number of SIP messages processed (counter)
    - sum - the sum of execution times for request_route or ksr_request_route() (microseconds)
    - min - the minimum execution time for request_route or ksr_request_route() (microseconds)
    - max - the maximum execution time for request_route or ksr_request_route() (microseconds)
    - avg - the average execution time for request_route or ksr_request_route() (microseconds)



https://www.kamailio.org/wiki/kemi/performance-tests/5.2.x

KEMI Performances For v5.2

Native

cnt: 62752
sum: 4184315
min: 17
max: 1948
avg: 66.6802

Lua

cnt: 61197
sum: 3844555
min: 22
max: 1852
avg: 62.8226

KAMAILIO

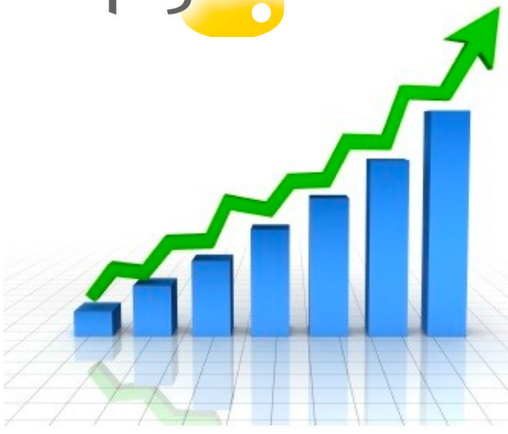
Python

cnt: 60785
sum: 4400362
min: 20
max: 2093
avg: 72.3922

https://www.kamailio.org/wiki/kemi/performance-tests/5.2.x

KEMI Performances For v5.2

- **Remarks**
  - the execution of **native scripting** routing blocks and **Lua scripting** functions **takes more or less same time**
  - sometimes is native scripting slightly faster, other times is Lua scripting slightly faster
  - the **average** (for both native and Lua scripts) is in the range of **60** to **80 microseconds** (1 / 1 000 000 of a second)
  - the minimum reflects more what it takes for the first REGISTER request without authorization header
    - which is **quickly challenged** with 401 response
  - the maximum reflect the processing of the REGISTER request **with valid authorization** header
    - which does **a query to database** to fetch the password
    - as well as **store/update the record** in the location hash table (writing to database on timer - usrloc db_mode 2).
  - there are variable number of **retransmissions**, being the reason for having cnt higher than 60000
    - a matter of sipp being able to match the response with the request at this high pace of stress testing
    - **4 000 registrations per second**, with a limit of **10 000 transactions at the same time**
    - stopping after **20 000 sent messages**
  - note that each registration is **challenged for authentication**, so it is resent with authorization header
  - the execution times for the **Python script** were in the **same range** as for native and Lua routing scripts



https://www.kamailio.org/wiki/kemi/performance-tests/5.2.x

KEMI

Full Control Of SIP Message Content

EXAMPLES IN LUA OR JAVASCRIPT

KSR FUNCTIONS

- Exported from core
  - Request, reply or methods checks
  - Set R-URI (and destination URI)

```
if(KSR.is_method("INVITE")) {
  ...
}
```

```
if KSR.is_INVITE() then
  ...
end
```

```
if KSR.is_method_in("IABC") then
  -- the method is INVITE, ACK, BYE or CANCEL
  ...
end
```

```
if KSR.is_GET() then
  ...
end
```

```
KSR.seturi("sip:alice@voip.com");
KSR.setuser("alice");
KSR.sethost("voip.com");
```

```
if KSR.is_KDMQ() then
  ...
end
```

```
KSR.setdsturi("sip:voip.com:5061;transport=tls");
```

```
INVITE sip:1002@asipto.lab;user=phone SIP/2.0
Via: SIP/2.0/UDP 192.168.178.62:1024;branch=z9hG4bK-ampxo9rfducs
From: "1001" <sip:1001@asipto.lab>;tag=6ehxxq7ebj
To: <sip:1002@asipto.lab;user=phone>
Call-ID: 31353437343538313039313233343 1-3xurg8bzzlop
CSeq: 1 INVITE
Max-Forwards: 70
User-Agent: snom370/8.7.5.35
Contact: <sip:1001@192.168.178.62:1024;line=vuw36p4a>;reg-id=1
Content-Type: application/sdp
Content-Length: 405

v=0
o=root 904410018 904410018 IN IP4 192.168.178.62
s=call
c=IN IP4 192.168.178.62
t=0 0
m=audio 62556 RTP/AVP 9 0 8 3 101
a=rtpmap:9 G722/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=ptime:20
a=sendrecv
```

https://www.kamailio.org/docs/tutorials/devel/kamailio-kemi-framework/core/

KSR.PV FUNCTIONS

- Propagated from pre-KEMI interpreter modules and extended
- Exported from core - pseudo-variable operations
  - Get and set values for pseudo-variables

```
v = KSR.pv.get("$ru")

v = KSR.pv.gete(“$avp(x)");

v = KSR.pv.getvn("$avp(x)", 0);

v = KSR.pv.getvs("$avp(x)", “foo");

v = KSR.pv.getw(“$avp(x)");
```

```
KSR.pv.seti("$var(x)", 10);

KSR.pv.sets("$rU", "alice");

if(KSR.pv.is_null("$rU")) {
  ...
}

KSR.pv.unset(“$du"); // set to $null
```

https://www.kamailio.org/docs/tutorials/devel/kamailio-kemi-framework/core/#ksrpv-submodule

KSR.HDR FUNCTIONS

- Propagated from pre-KEMI interpreter modules and extended
- Exported from core - header operations
  - Check, append (+after), insert (+before), remove (+append), get (with index)
  - Append to reply

```
if(KSR.hdr.is_present("X-My-Hdr") > 0) { ... }
```

```
KSR.hdr.append_to_reply("X-My-Hdr: " + KSR.kx.get_srcip() + "\r\n");
```

```
KSR.hdr.append("X-My-Hdr: " + KSR.kx.get_srcip() + "\r\n");
```

```
KSR.hdr.append_after("X-My-Hdr: " + KSR.kx.get_srcip() + "\r\n", "Call-Id");
```

```
KSR.hdr.insert("X-My-Hdr: " + KSR.pv.getw("$si") + "\r\n");
```

```
KSR.hdr.insert_before("X-My-Hdr: " + KSR.pv.getw("$si") + "\r\n", "Call-Id");
```

```
KSR.hdr.remove("X-My-Hdr");
```

```
KSR.hdr.rmappend("X-My-Hdr", "X-My-Hdr: abc\r\n");
```

```
v = KSR.hdr.get("X-My-Hdr");
```

```
v = KSR.hdr.get_idx("X-My-Hdr", 1);
```

```
if(KSR.hdr.match_content("X-My-Hdr", "in", "test", "o")) { ... }
```

https://www.kamailio.org/docs/tutorials/devel/kamailio-kemi-framework/core/#ksrhdr-submodule

KSR.KX FUNCTIONS

- Exported by kemix module - KEMI extenssions
  - Dedicated functions that can be used instead of pseduo-variable variants

```
ua = KSR.kx.gete_ua();
```

```
[ ua = KSR.pv.get("$ua") ]
```

```
ruri = KSR.kx.get_ruri();
furi = KSR.kx.get_furi();
turi = KSR.kx.get_turi();
```

```
code = KSR.kx.get_status();
```

```
ruser = KSR.kx.get_ruser();
fuser = KSR.kx.get_fuser();
tuser = KSR.kx.get_tuser();
```

```
body = KSR.kx.get_body();
blen = KSR.kx.get_bodylen();
```

```
rhost = KSR.kx.get_rhost();
```

```
msg = KSR.kx.get_msgbuf();
len = KSR.kx.get_msglen();
```

```
callid = KSR.kx.get_callid();
srcip = KSR.kx.get_srcip();
```

https://www.kamailio.org/docs/tutorials/devel/kamailio-kemi-framework/modules/#kx

KSR.TEXTOPS FUNCTIONS

- Exported by textops module - text based operations
  - A few overlap with KSR.hdr functions
  - Check, search, replace, substitute, remove and set parts of the SIP message

```
KSR.textops.replace("alice", "carol");
KSR.textops.replace_all(“alice”, "carol");
KSR.textops.replace_hdrs(“alice”, "carol");
KSR.textops.replace_body(“Hello!”, “Hi!");
KSR.textops.replace_body_str(“Hello!”, “Hi!", “a”);
KSR.textops.search_append(“John ", “Smith");
KSR.textops.subst_uri(“/alice/carol/“);
KSR.textops.remove_hf_match(“X-Hdr", "in", “alice");
```

```
KSR.textops.subst("/^o=(.*) /o=$fU /");
KSR.textops.subst_body(“/^o=(.*) /o=$fU /");
KSR.textops.subst_hf(“From”, “/alice/carol/“, “f");
KSR.textops.set_body(“Hello!”, “text/plain“);
KSR.textops.set_body_multipart_mode();
KSR.textops.append_body_part(“Hello!”, “text/plain“);
if(KSR.textops.has_body()>0) { ... }
KSR.textops.filter_body(“application/sdp”);
```

https://www.kamailio.org/docs/tutorials/devel/kamailio-kemi-framework/modules/#textops

KSR.TEXTOPSX FUNCTIONS

- Exported by textopsx module - more text based operations
  - Several advanced SIP message and header content operations
  - Headers and body lines iterators

```
KSR.textopsx.remove_body();
```

```
KSR.textopsx.msg_apply_changes()
```

```
KSR.textopsx.msg_set_buffer("INVITE sip:...");
```

```
KSR.textopsx.assign_hf_value("X-Hdr", "udp");
KSR.textopsx.append_hf_value("X-Hdr", "tls");
KSR.textopsx.keep_hf_re("User[-]Agent");
KSR.textopsx.remove_hf_value("X-Hdr.proto");
KSR.textopsx.include_hf_value("Supported", "path");
KSR.textopsx.exclude_hf_value("Supported", "100rel");
```

```
body = KSR.kx.get_msgbuf();
ubody = string.upper(body);
KSR.textopsx.msg_set_buffer(ubody);
```

includes msg_apply_changes()

https://www.kamailio.org/docs/tutorials/devel/kamailio-kemi-framework/modules/#textopsx

- Header fields iterators

```
KSR.textopsx.hf_iterator_start(“i1”);
while(KSR.textopsx.hf_iterator_next("i1")>0) {
    if(KSR.textopsx.hf_iterator_hname(“i1”)==“My-Header”) {
        KSR.textopsx.hf_iterator_rm("i1");
    }
    if(KSR.textopsx.hf_iterator_hname(“i1”)==“My-Header”) {
        KSR.textopsx.hf_iterator_insert(“i1",
            "My-New-Header: abc\r\n");
   }
    if(KSR.textopsx.hf_iterator_hbody(“i1”)==“xyz”) {
        KSR.textopsx.hf_iterator_append(“i1",
            "My-Other-Header: def\r\n”);
   }
}
KSR.textopsx.hf_iterator_end("i1");
```

NO
msg_apply_changes()

```
INVITE sip:1002@asipto.lab;user=phone SIP/2.0
Via: SIP/2.0/UDP 192.168.178.62:1024;branch=z9hG4bK-ampxo9rfducs
From: "1001" <sip:1001@asipto.lab>;tag=6ehxxq7ebj
To: <sip:1002@asipto.lab;user=phone>
Call-ID: 31353437343538313039313233343 1-3xurg8bzzlop
CSeq: 1 INVITE
Max-Forwards: 70
User-Agent: snom370/8.7.5.35
Contact: <sip:1001@192.168.178.62:1024;line=vuw36p4a>;reg-id=1
Content-Type: application/sdp
Content-Length: 405

v=0
o=root 904410018 904410018 IN IP4 192.168.178.62
s=call
c=IN IP4 192.168.178.62
t=0 0
m=audio 62556 RTP/AVP 9 0 8 3 18 101
a=rtpmap:9 G722/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=ptime:20
a=sendrecv
```

- Body lines iterators


```
KSR.textopsx.bl_iterator_start("b1");
while(KSR.textopsx.bl_iterator_next("b1")) {
    line = KSR.textopsx.bl_iterator_value("b1");
    if(line.startsWith("a=info:abc")) {
        KSR.textopsx.bl_iterator_rm("b1");
        KSR.textopsx.bl_iterator_insert("b1",
                "a=info:xyz\r\n");
    }
}
KSR.textopsx.bl_iterator_end("b1");
```

NO
msg_apply_changes()

```
INVITE sip:1002@asipto.lab;user=phone SIP/2.0
Via: SIP/2.0/UDP 192.168.178.62:1024;branch=z9hG4bK-ampxo9rfducs
From: "1001" <sip:1001@asipto.lab>;tag=6ehxxq7ebj
To: <sip:1002@asipto.lab;user=phone>
Call-ID: 31353437343538313039313233343 1-3xurg8bzzlop
CSeq: 1 INVITE
Max-Forwards: 70
User-Agent: snom370/8.7.5.35
Contact: <sip:1001@192.168.178.62:1024;line=vuw36p4a>;reg-id=1
Content-Type: application/sdp
Content-Length: 405

v=0
o=root 904410018 904410018 IN IP4 192.168.178.62
s=call
c=IN IP4 192.168.178.62
t=0 0
m=audio 62556 RTP/AVP 9 0 8 3 18 101
a=rtpmap:9 G722/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=ptime:20
a=sendrecv
```

KSR.POSOPS FUNCTIONS

- Exported by posops module
  - Operations over the SIP message buffer using position (index)

```
ps = KSR.posops.pos_headers_start();
pe = KSR.posops.pos_headers_end();
ps = KSR.posops.pos_body_start();
pe = KSR.posops.pos_body_end();

KSR.posops.pos_insert(100, “1.2.3.4”);
KSR.posops.pos_append(100, “1.2.3.4”);
KSR.posops.pos_rm(100, 10);
KSR.posops.pos_set_char(100, “X”);
p = KSR.posops.pos_find_str(100, “kamailio”);
p = KSR.posops.pos_findi_str(100, “kamailio”);
p = KSR.posops.pos_rfind_str(100, “kamailio”);
p = KSR.posops.pos_rfindi_str(100, “kamailio”);
p = KSR.posops.pos_search(-100, “[0-9]+”);
p = KSR.posops.pos_rsearch(100, “[0-9]+”);
```

0, 1, 2 …

-1, -2, -3 …

```
INVITE sip:1002@asipto.lab;user=phone SIP/2.0
Via: SIP/2.0/UDP 192.168.178.62:1024;branch=z9hG4bK-ampxo9rfducs
From: "1001" <sip:1001@asipto.lab>;tag=6ehxxq7ebj
To: <sip:1002@asipto.lab;user=phone>
Call-ID: 31353437343538313039313233343 1-3xurg8bzzlop
CSeq: 1 INVITE
Max-Forwards: 70
User-Agent: snom370/8.7.5.35
Contact: <sip:1001@192.168.178.62:1024;line=vuw36p4a>;reg-id=1
Content-Type: application/sdp
Content-Length: 405

v=0
o=root 904410018 904410018 IN IP4 192.168.178.62
s=call
c=IN IP4 192.168.178.62
t=0 0
m=audio 62556 RTP/AVP 9 0 8 3 101
a=rtpmap:9 G722/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=ptime:20
a=sendrecv
```

https://www.kamailio.org/docs/tutorials/devel/kamailio-kemi-framework/modules/#posops

OTHER MODULES

- KSR.siputils - exported by siputils module
- KSR.corex - exported by corex module
- KSR.nathelper - exported by nathelper module
- KSR.uac - exported by uac module
- KSR.uac_redirect - exported by uac_redirect module
- KSR.pvx - exported by pv module

https://www.kamailio.org/docs/tutorials/devel/kamailio-kemi-framework/modules/

THANK YOU!

Daniel-Constantin Mierla
Co-Founder Kamailio Project
@miconda
asipto.com

Hoping For A Berlin Edition In 2023
(late May - early June)

www.kamailioworld.com

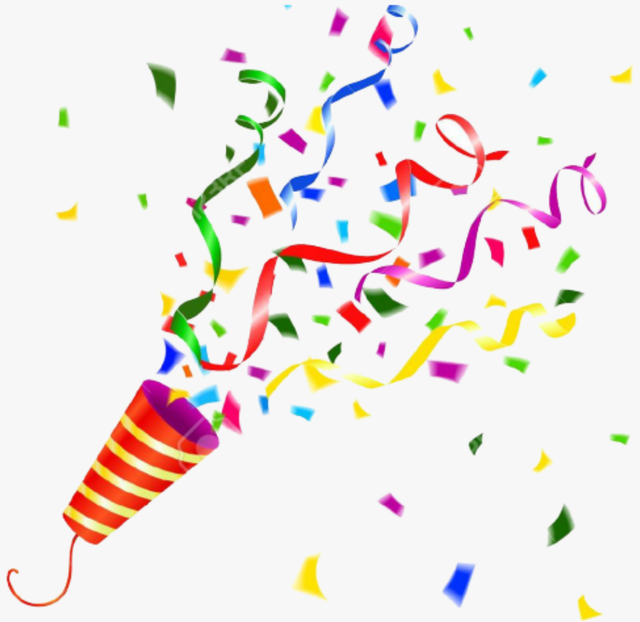
KAMAILIO

2001 - 2022

development

KAMAILIO WORLD CONFERENCE

2013 - 2022



KAMAILIO

2002 - 2022

Open Source - GPL