



5.0

a look to the future

Daniel-Constantin Mierla
Co-Founder Kamailio Project
www.asipto.com
[@miconda](https://twitter.com/miconda)



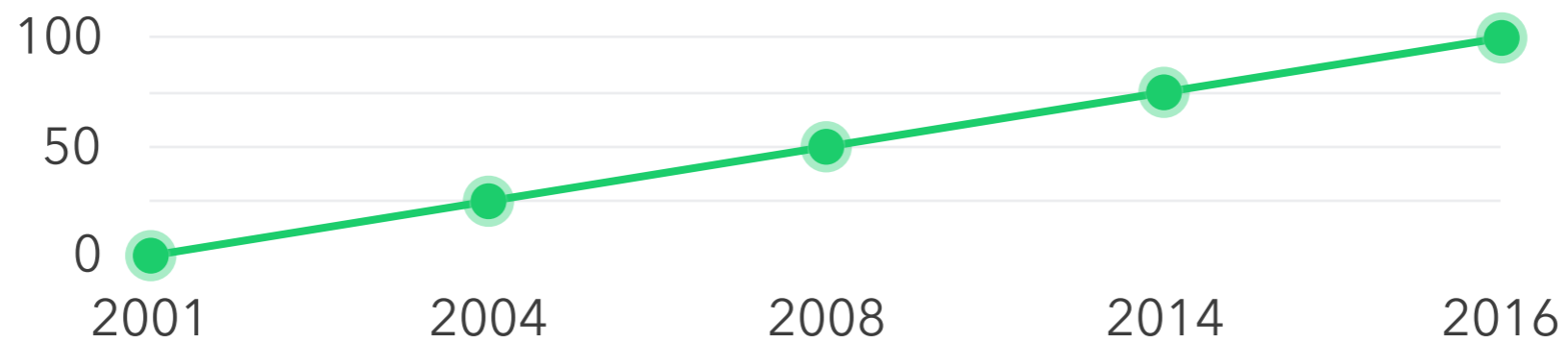
KAMAILIO WORLD

CONFERENCE & EXHIBITION

BERLIN, GERMANY, MAY 18-20, 2016



Evolution



15 years of
development



Market has transformed
People have changed
Technology has evolved
Demands have increased
Expectations are bigger

leveraging the past to build the future

Four Eras

1

Static

2001-2005

Static configuration
Precompiled routing

2

Dynamic

2005-2008

Variables
Transformations

3

Async

2008-2014

Suspend and resume
Non-blocking TCP/TLS

4

Apps

2014-

API-based routing
Beyond telephony

APPS



document

json
xml
serialization

json
jansson
rtjson
xmlops
transformations



transport

http
jsonrpc
sip
custom

utils
http_client
jsonrpc-c
evapi
uac

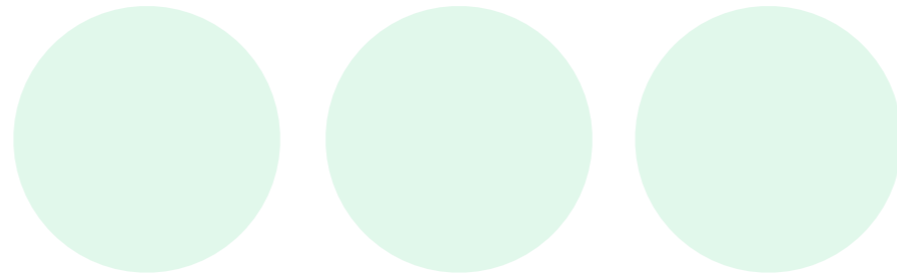


control

interpreter

native
lua
python
perl
.net - java

More Control



ROUTING LOGIC
USING DIFFERENT
INTERPRETERS

kemi

kamailio embedded interface

a framework that allows selecting
your favourite scripting language
for writing sip routing rules

kemi & kamailio.cfg

- not affected

- core parameters
- loading modules
- module parameters

- new

- specify the language for routing rules (routing blocks)
- `cfgengine="language"`
- at this moment, language can be: native, lua, python
- default is 'native'
- 'lua' requires 'app_lua' module
- 'python' requires 'app_python' module

native vs. lua

```
request_route {  
  
    # per request initial checks  
    route(REQINIT);  
  
    # NAT detection  
    route(NATDETECT);  
  
    # CANCEL processing  
    if (is_method("CANCEL")) {  
        if (t_check_trans()) {  
            route(RELAY);  
        }  
        exit;  
    }  
  
    # handle requests within SIP dialogs  
    route(WITHINDLG);  
}
```

```
function ksr_request_route()  
  
    -- per request initial checks  
    if ksr_route_reqinit() == -255 then return 1; end  
  
    -- NAT detection  
    if ksr_route_natdetect() == -255 then return 1; end  
  
    -- CANCEL processing  
    if KSR.pv.get("$rm") == "CANCEL" then  
        if KSR.tm.t_check_trans() > 0 then  
            ksr_route_relay();  
        end  
        return 1;  
    end  
  
    -- handle requests within SIP dialogs  
    if ksr_route_withindlg() == -255 then return 1; end
```


native vs. lua

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

```
-- Caller NAT detection
function ksr_route_natdetect()
    KSR.force_rport();
    if KSR.nathelper.nat_uac_test(19)>0 then
        if KSR.pv.get("$rm")==="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
```

native vs. python

```
class kamailio:
```

```
def ksr_request_route(self, msg):
```

```
    # per request initial checks
```

```
    if self.ksr_route_reqinit(msg)==-255 :  
        return 1;
```

```
    # NAT detection
```

```
    if self.ksr_route_natdetect(msg)==-255 :  
        return 1;
```

```
    # CANCEL processing
```

```
    if KSR.pv.get("$rm") == "CANCEL" :  
        if KSR.tm.t_check_trans()>0 :  
            self.ksr_route_relay();  
        return 1;
```

```
    # handle requests within SIP dialogs
```

```
    self.ksr_route_withindlg(msg);
```

```
# Caller NAT detection
```

```
def ksr_route_natdetect(self, msg):
```

```
    KSR.force_rport();
```

```
    if KSR.nathelper.nat_uac_test(19)>0 :
```

```
        if KSR.pv.get("$rm")== "REGISTER" :
```

```
            KSR.nathelper.fix_nated_register();
```

```
        elif KSR.siputils.is_first_hop()>0 :
```

```
            KSR.nathelper.set_contact_alias();
```

```
        KSR.setflag(FLT_NATS);
```

```
    return 1;
```

What About Code?

```
static int lua_sr_maxfwd_process_maxfwd(lua_State *L)
{
    int ret;
    int limit;
    sr_lua_env_t *env_L;
    env_L = sr_lua_env_get();
    if(!(_sr_lua_exp_reg_mods&SR_LUA_EXP_MOD_MAXFWD))
    {
        LM_WARN("weird: maxfwd function executed but module not registered\n");
        return app_lua_return_error(L);
    }
    if(env_L->msg==NULL)
    {
        LM_WARN("invalid parameters from Lua env\n");
        return app_lua_return_error(L);
    }
    if(lua_gettop(L)!=1)
    {
        LM_WARN("invalid number of parameters from Lua\n");
        return app_lua_return_error(L);
    }
    limit = lua_tointeger(L, -1);
    if(limit<0)
    {
        LM_WARN("invalid parameters from Lua\n");
        return app_lua_return_error(L);
    }
    ret = _lua_maxfwddb.process_maxfwd(env_L->msg, limit);
    return app_lua_return_int(L, ret);
}
```

*app_lua - old style of exporting
export c function from maxfwd with custom struct
bind to maxfwd and write wrapper in app_lua*

What About Code?

kemi

*- no change in app_lua
- just export in maxfwd*

```
/**
 *
 */
static sr_kemi_t sr_kemi_maxfwd_exports[] = {
    { str_init("maxfwd"), str_init("process_maxfwd"),
      SR_KEMIP_INT, process_maxfwd_header,
      { SR_KEMIP_INT, SR_KEMIP_NONE, SR_KEMIP_NONE,
        SR_KEMIP_NONE, SR_KEMIP_NONE, SR_KEMIP_NONE }
    },

    { {0, 0}, {0, 0}, 0, NULL, { 0, 0, 0, 0, 0, 0 } }
};

/**
 *
 */
int mod_register(char *path, int *dlflags, void *p1, void *p2)
{
    sr_kemi_modules_add(sr_kemi_maxfwd_exports);
    return 0;
}
```

Kamailio 5.0

- embedded interpreters for routing blocks
- unit testing
- restructuring the source code tree
- review the building system
- relocate unmaintained or unnecessary modules
- add new features

get on board

suggestions - discussions - coding - documentation - packaging

kemi & kamailio.cfg

- equivalent of kamailio-basic.cfg
 - security, sanity check, nat traversal, authentication, location, ...
- inside the source tree - see **examples/kemi/**
 - <https://github.com/kamailio/kamailio/tree/master/examples/kemi>
- common configuration
 - <https://github.com/kamailio/kamailio/blob/master/examples/kemi/kamailio-basic-kemi.cfg>
- native routing blocks
 - <https://github.com/kamailio/kamailio/blob/master/examples/kemi/kamailio-basic-kemi-native.cfg>
- lua script
 - <https://github.com/kamailio/kamailio/blob/master/examples/kemi/kamailio-basic-kemi-lua.lua>
- python script
 - <https://github.com/kamailio/kamailio/blob/master/examples/kemi/kamailio-basic-kemi-python.py>

Open Discussion



discuss now



implement tomorrow



merge on monday

Thank You



2021

20 years of Kamailio

meanwhile

Kamailio World 2017

an may RTC events around the globe

Daniel-Constantin Mierla
Co-Founder Kamailio Project
www.asipto.com
[@miconda](https://twitter.com/miconda)



KAMAILIO WORLD

CONFERENCE & EXHIBITION

BERLIN, GERMANY, MAY 18-20, 2016

