




KAMAILIO - PICK YOUR SIP ROUTING SCRIPTING LANGUAGE

DANIEL-CONSTANTIN MIERLA (@MIGONDA)
CO-FOUNDER KAMAILIO SIP SERVER PROJECT
WWW.ASIPTO.COM

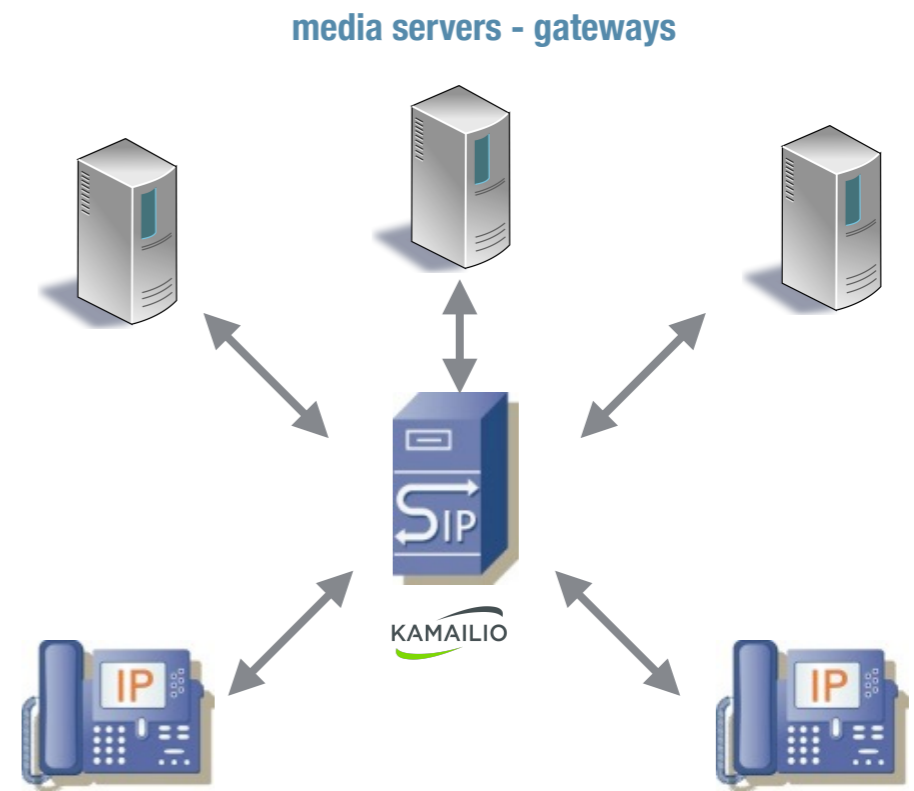
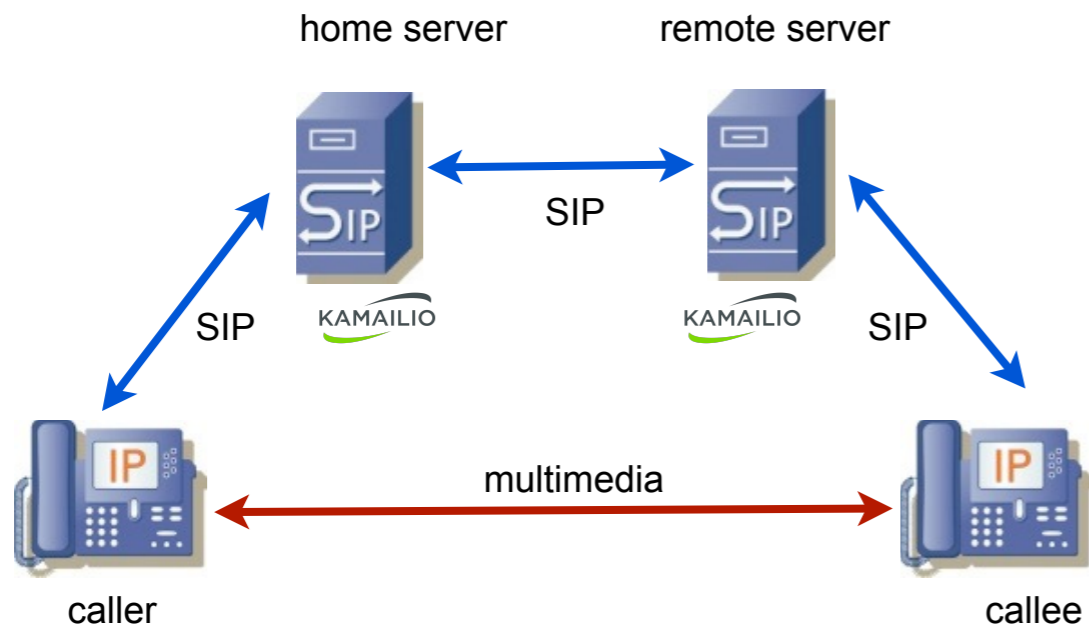
FOSDEM 2018 - BRUSSELS

Kamailio In One Slide




- * Open Source SIP (IETF RFC3261) Signaling Server implementation
 - * Developed since 2001 (SER - OpenSER - Kamailio) - started at FhG Fokus Research Institute in Berlin, Germany - now a community driven project
 - * Can be used for VoIP (Voice, Video, VoLTE/IMS, SIP-I/SIP-T), Instant Messaging, Presence, WebRTC, IoT, Diameter, SQL and NoSQL backends
 - * Built for flexibility and scalability - used by large telecoms, mobile operators and OTT services world wide
 - * IPv6/IPv4 - UDP/TCP/TLS/SCTP/WebSocket - asynchronous routing
 - * Classic SIP - WebRTC gateway using Kamailio + RTPEngine
 - * Embedded interpreters: Lua, Python, JavaScript, Squirrel, Perl, .Net, Java
 - * Over 200 modules (extensions) - <https://www.kamailio.org/docs/modules/stable/>
 - * Over 50 active developers each year (over 300 contributors over the time)
 - * Runs its own conference - Kamailio World
 - * the 6th edition: May 14-17, 2018, in Berlin, Germany: <https://www.kamailioworld.com>
- 

Kamailio In The Network




Kamailio Configuration File



- * Uses a scripting language developed from scratch since 2001
 - * Referred as native scripting language
 - * Two parts from functionality point of view
 - * Initialization of Kamailio
 - * Global parameters
 - * Loading modules
 - * Modules parameters
 - * SIP Routing Logic
 - * Known as routing blocks
 - * Sort of functions executed for each SIP message
- 

Native Scripting Language




- * Among benefits
 - * Behaviour of the interpreter controlled to the smallest details
 - * Can be optimized for specific needs of SIP routing
 - * Can control the evolution of the language
 - * Among drawbacks
 - * Reinvent the wheel
 - * Example: implement in C common string and arithmetic operations
 - * Focus not on scripting language, resulting in no extensive documentation for the language itself
 - * No external libraries that can be just imported
 - * No usage outside of Kamailio scope
 - * Some limitations due to initial design (e.g., no hot reload)
- 

reload of routing logic without restart
more features of the scripting language

KEMI - Kamailio Embedded Interface



- * Allow writing the routing logic in well known scripting languages by embedding interpreters
 - * reload without restart when possible
 - * Initial version launched with Kamailio v5.0
 - * Lots of improvements with Kamailio v5.1
 - * Implemented for:
 - * Lua
 - * Python
 - * JavaScript
 - * Squirrel
 - * The old inline interpreters still available
 - * Perl, .Net (C#), Java
- 

Native Scripting Language - Inline Lua Execution

myscript.lua

```
...  
function sr_append_fu_to_reply()  
    sr.hdr.append_to_reply("P-From: " .. sr.pv.get("$fu") .. "\r\n");  
end  
...
```

```
...  
modparam("app_lua", "load", "/usr/local/etc/kamailio/lua/myscript.lua")  
...  
request_route {  
    ...  
    if(!lua_run("sr_append_fu_to_reply")) {  
        xdbg("SCRIPT: failed to execute lua function!\n");  
    }  
    ...  
}  
...
```


kamailio.cfg

Routing Block - Native Scripting Language



<https://github.com/kamailio/kamailio/blob/master/misc/examples/kemi/kamailio-basic-kemi-native.cfg>

```
298 # Routing to foreign domains
299 route[SIP0UT] {
300     if (uri==myself) return;
301
302     append_hf("P-hint: outbound\r\n");
303     route(RELAY);
304     exit;
305 }
```



Routing Block - Lua Scripting Language

- app_lua

- existing since 2010 offering inline execution of Lua scripts
- https://www.kamailio.org/docs/modules/stable/modules/app_lua.html



- highlights

- very small interpreter, fast
- decent number of extensions (native lua libraries)
- popular in gaming space, also in RTC (Asterisk, FreeSwitch)

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

```
324  -- Routing to foreign domains
325  function ksr_route_sipout()
326      if KSR.is_myself(KSR.pv.get("$ru")) then return 1; end
327
328      KSR.hdr.append_hf("P-Hint: outbound\r\n");
329      ksr_route_relay();
330      KSR.x.exit();
331  end
```

Routing Block - JavaScript Scripting Language

- introduced in Kamailio v5.1

- not in stable version
- https://www.kamailio.org/docs/modules/devel/modules/app_jsdt.html

- highlights

- duktape.org engine - small, easy to embed by importing source code files
- very popular scripting language
- no external dependencies
- reload of routing script
- not many extensions



<https://github.com/kamailio/kamailio/blob/master/misc/examples/kemi/kamailio-basic-kemi-jsdt.js>

```
331 // Routing to foreign domains
332 function ksr_route_sipout()
333 {
334     if (KSR.is_myself(KSR.pv.get("$ru"))) { return; }
335
336     KSR.hdr.append_hf("P-Hint: outbound\r\n");
337     ksr_route_relay();
338     KSR.x.exit();
339 }
```

Routing Block - Python Scripting Language

- app_python

- introduced in 2010 for inline execution of Python scripts
- https://www.kamailio.org/docs/modules/stable/modules/app_python.html

- highlights

- popular scripting language
- most extensive number of extensions
- object oriented, perceived as slower than other scripting languages
- reloading of the routing script not completed yet



<https://github.com/kamailio/kamailio/blob/master/misc/examples/kemi/kamailio-basic-kemi-python.py>

```
332     # Routing to foreign domains
333     def ksr_route_sipout(self, msg):
334         if KSR.is_myself(KSR.pv.get("$ru")) :
335             return 1;
336
337         KSR.hdr.append("P-Hint: outbound\r\n");
338         self.ksr_route_relay(msg);
339         return -255;
```

Routing Block - Squirrel Scripting Language

- app_sqlang

- introduced in Kamailio 5.1
- https://www.kamailio.org/docs/modules/devel/modules/app_sqlang.html

- highlights

- <http://www.squirrel-lang.org>
- very small interpreter
- embedded by importing its source code
- no external dependency
- reload of scripting script
- pre-compilation options



<https://github.com/kamailio/kamailio/blob/master/misc/examples/kemi/kamailio-basic-kemi-sqlang.sq>

```
332 // Routing to foreign domains
333 function ksr_route_sipout()
334 {
335     if (KSR.is_myself(KSR.pv.get("$ru"))) { return; }
336
337     KSR.hdr.append_hf("P-Hint: outbound\r\n");
338     ksr_route_relay();
339     KSR.x.exit();
340 }
```

KEMI Documentation



over 600 functions exported to kemi framework

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


examples

<https://github.com/kamailio/kamailio/tree/master/misc/examples/kemi>



The Other Inline Interpreters



- * Perl via `app_perl` module
 - * Embeds the standard Perl interpreter
 - * The first one added in 2006
 - * https://www.kamailio.org/docs/modules/stable/modules/app_perl.html
 - * .Net (C#) via `app_mono` module
 - * Embeds the Mono interpreter
 - * Besides C#, other languages should be supported by building the managed assemblies, such as F#, VisualBasic.NET, Scala — see:
 - * <http://www.mono-project.com/Languages>
 - * https://www.kamailio.org/docs/modules/stable/modules/app_mono.html
 - * Java via `app_java` module
 - * Embeds the JVM interpreter
 - * https://www.kamailio.org/docs/modules/stable/modules/app_java.html
- 

Kamailio SIP Server Project
kamailio.org
@kamailio

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

Thank You!

Hope to see some of you at Kamailio World 2018!
www.kamailioworld.com

