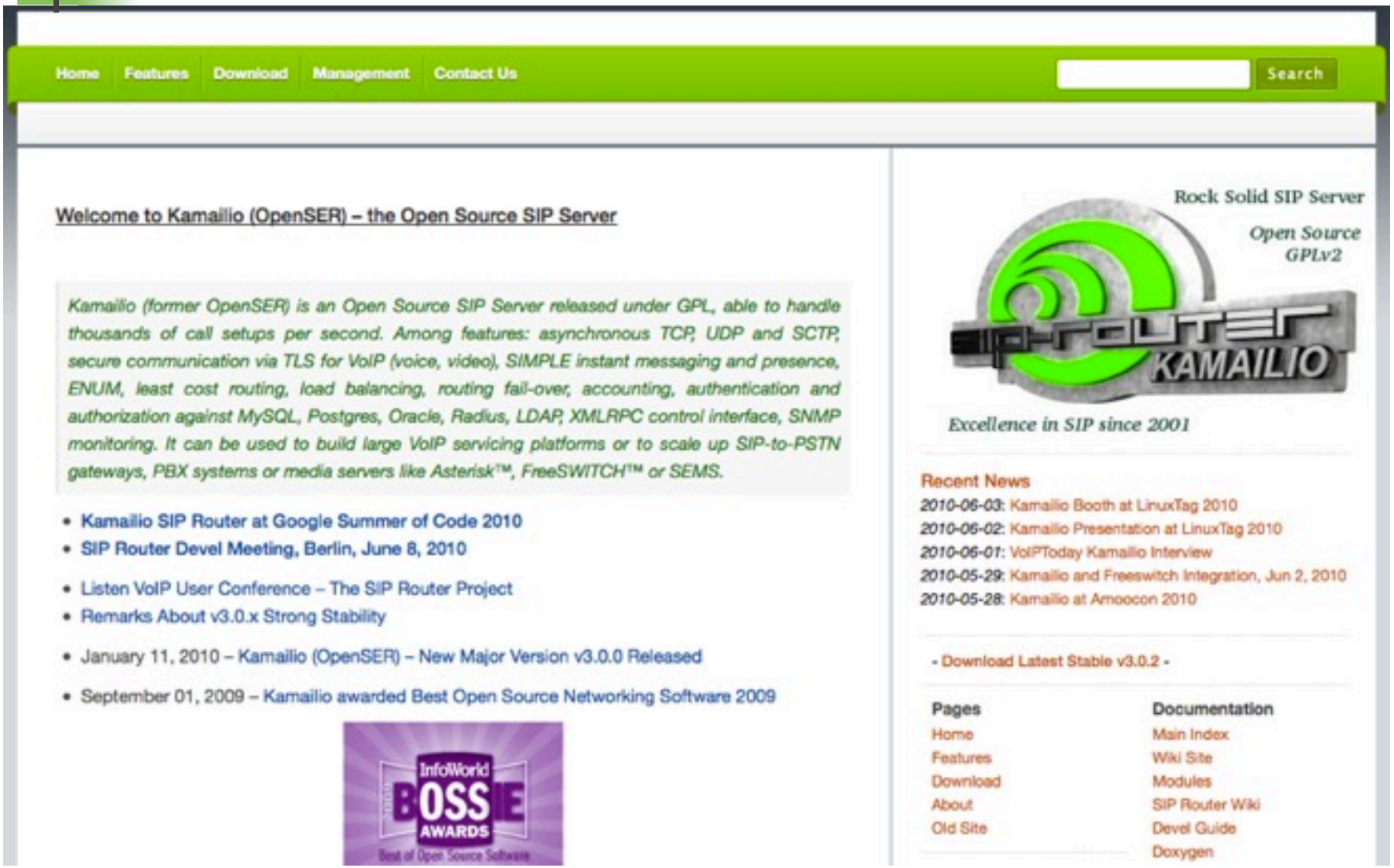
Building Your Own Secure Unified Communication Service

Daniel-Constantin Mierla
Co-Founder

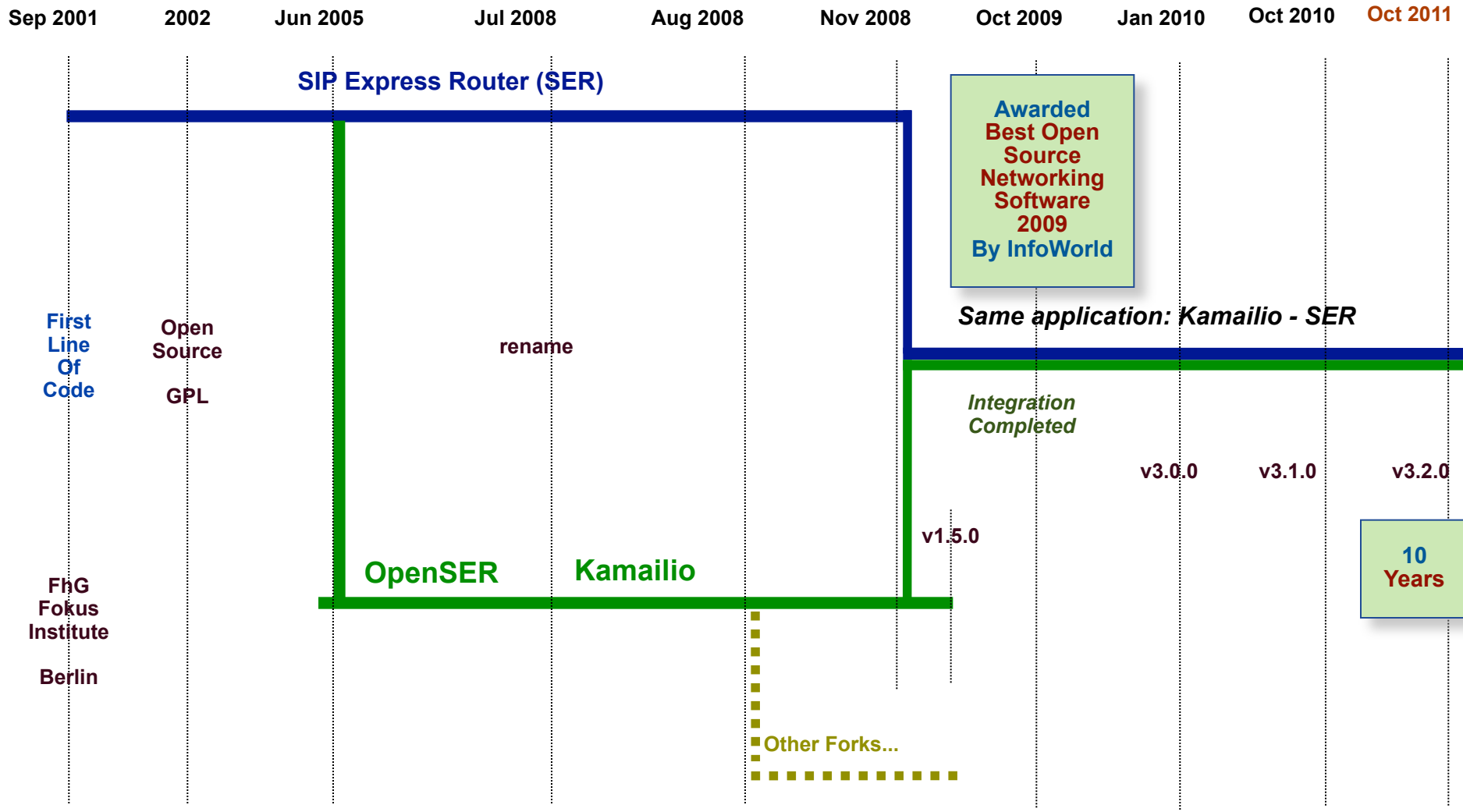
www.kamailio.org

www.asipto.com

www.kamailio.org



History


Sep 2001
2002
Jun 2005
Jul 2008
Aug 2008
Nov 2008
Oct 2009
Jan 2010
Oct 2010
Oct 2011
SIP Express Router (SER)
Awarded Best Open Source Networking Software 2009 By InfoWorld
First Line Of Code
Open Source
GPL
rename
Same application: Kamailio - SER
Integration Completed
v3.0.0
v3.1.0
v3.2.0
v1.5.0
10 Years
FhG Fokus Institute
Berlin
OpenSER
Kamailio
Other Forks...

Stats

Kamailio and SIP Express Router

Main Language: C

Total Lines of Code: 489,210

Active Contributors: 39

Commit Activity Timeline:

5 year commit count

Updated Feb 04, 2012

Cocomo

Kamailio and SIP Express Router - Project Cost

Include	Avg. Salary
Markup And Code	$ 55000 /year
Codebase	**Effort (est.)**
635,457 Lines	171 Person Years
Estimated Cost	**$9,379,931**

Updated Feb 04, 2012 more at

Factoids

- Mostly written in C
- Mature, well-established codebase
- Very large, active development team

Kamailio and SIP Express Router, updated Feb 04, 2012 more at

Features

SIP Application Server
proxy, redirect, registrar, location

Plug in module interface
(over 150 mods)
Small footprint
Customizable routing policy

IPv4-IPv6
Asynchronous
UDP/TCP/TLS/SCTP
DNS NAPTR & SRV
DNS Failover and Load Balancing
DNS Internal Cache

Presence & IM Services
End-to-End
SIMPLE Server
RCS - RCS-e
Presence User Agent
Resource Lists
XCAP Client & Server
MSRP Relay

Carrier Routing
Dynamic Routng
ENUM lookup support
Advanced routing
(Load Balancing and LCR)
DID, Aliases & speeddial

Multi-domain support
LDAP/H.350 support
Embedded HTTP Server

Features

Embedded Lua, Perl
Python, C#
Java SIP Servlet
programming interface

No-SQL
Memcached
Redis
Cassandra

NAT traversal
Security
permissions
anti-DOS attacks
User call preferences
Call Processing
Language

Link any application to Kamailio using
FIFO/UNIXSOCK/DATAGRAM/XMLRPC interfaces

Database API
MySQL
PostgreSQL
SQLite
UNIXODBC
BERKELEYDB
ORACLE
Text files
RADIUS

Gateway

SMS
XMPP

Accounting through log file,
database or Radius/DIAMETER
servers

New in 3.1.0

Flexibility

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

Maintenance

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

Performance

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

Features

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

New in 3.2.0 - *Oct 2011*

Reg-Info Implementation

RFC3860
pub-sub service for
location data

Embedded XCAP server

OMA - specs
If-Match cond

RLS

OMA specs
split NOTIFY bodies
XPath support within doc

Presence Server

data distribution across
many instances through
database

Presence User Agent

updates for latest
RL services

SQLite connector

use file based
database for
embedded
systems

Many native extensions to Lua

cfg routing logic all in Lua

Distributed Message Queue

Using SIP and Peer-to-Peer

New in 3.2.0

async module

run asynchronously parts of config file (route blocks)

Redis No-SQL

connector from config

Partitioned user location service

many nodes sharing location data

ipops module

a set of operations for handling IPv4/IPv6 addresses

New features in old parts

acc - write full CDR at once
dialog - attach extra attributes
core - more pre-processor directives
pv - new variables and transformations
tmx - export of async TM functions
sqlops - support for xavps
uac - enhancements to remote registration
siptrace - traffic replication enhancements
.....

sdpops module

SDP body management

JSON
JSONRPC

IMS Extensions
about 10 new modules (P-CSCF, I-CSCF, S-CSCF...)

New in 3.2.0

http://www.kamailio.org/w/kamailio-openser-v3-2-0-release-notes/

http://www.kamailio.org/wiki/features/new-in-3.2.x

New in devel (3.3.0) - ***June 2012***

- New modules
  - **xhttp_rpc**
    - execute RPC commands via HTTP
  - **presence_profile**
    - get phone configuration via SIP Presence mechanisms
  - **app_mono**
    - embedded execution of managed code (C#)
  - **db_cassandra**
    - DB connector for Cassandra
  - **db_cluster**
    - generic DB clustering system
  - **msrp**
    - embedded MSRP relay
  - **tmrec**
    - time based recurrence matching (RFC2445)
- http://www.kamailio.org/wiki/features/new-in-devel

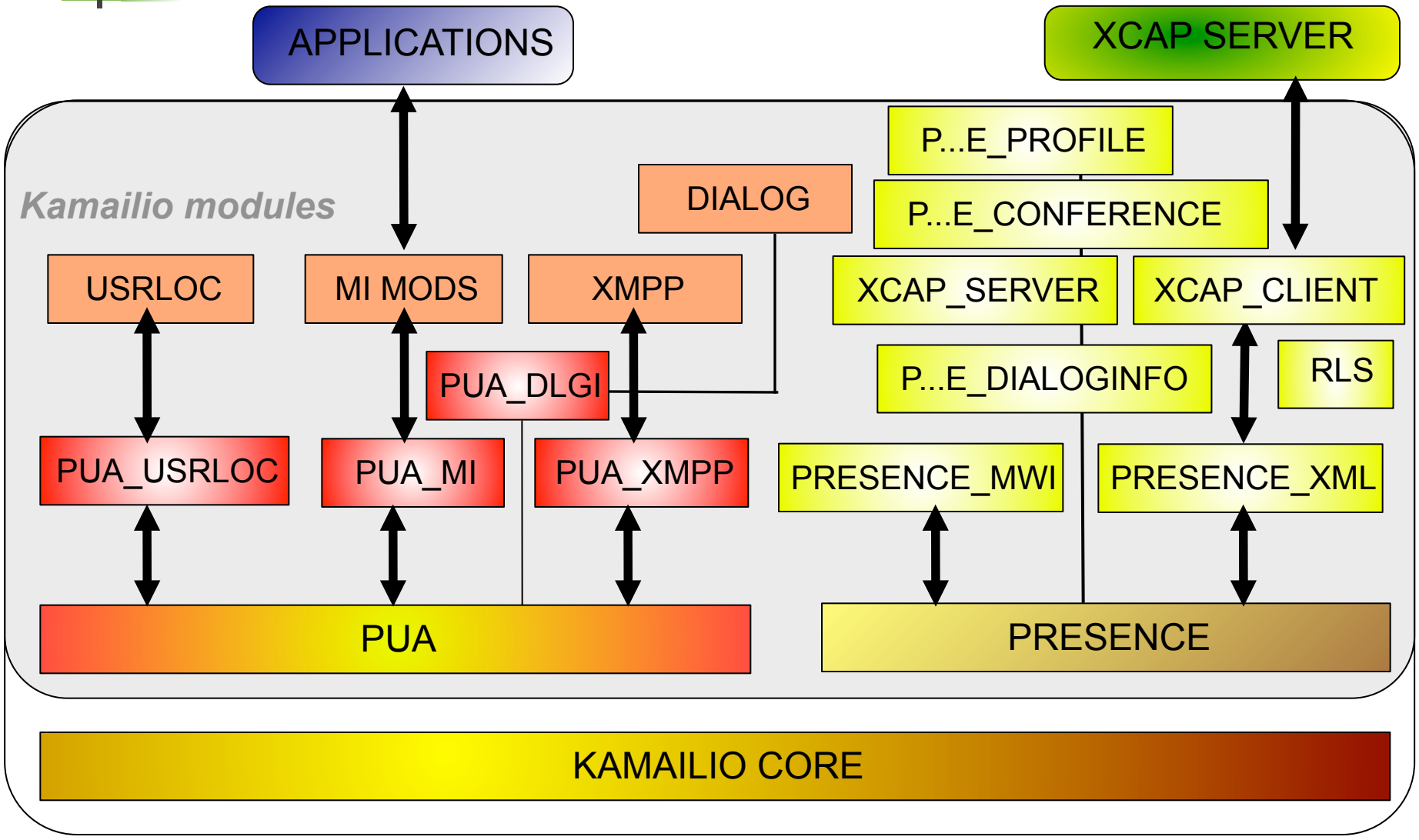
New in devel (3.3.0) - ***June 2012***

- Enhancements to existing modules
  - auth, auth_db
  - rr, app_lua
  - tls, textops
  - dialog, dialplan
  - usrloc and registrar (GRUU and SIP outbound)
  - presence, rls and xcap
  - sdpops, xlog
  - db_postgres, db_mysql
  - dispatcher, rtimer
  - pv, tm
- http://www.kamailio.org/wiki/features/new-in-devel

New in devel (3.3.0) - ***June 2012***

- Enhancements to core and internal libraries
  - embedded json parser and serialization engine
  - unique id generator
  - control number of workers per socket
  - control memory defragmentation
  - control memory safety operations
  - alert on time expensive config actions and database operations
  - DB bitwise operations and non-pooled connections
  - TLS max connections limit
  - TCP statistics
  - send() with outbound proxy
  - per socket advertised addresses
- http://www.kamailio.org/wiki/features/new-in-devel

SIP Beyond VoIP - Presence Services



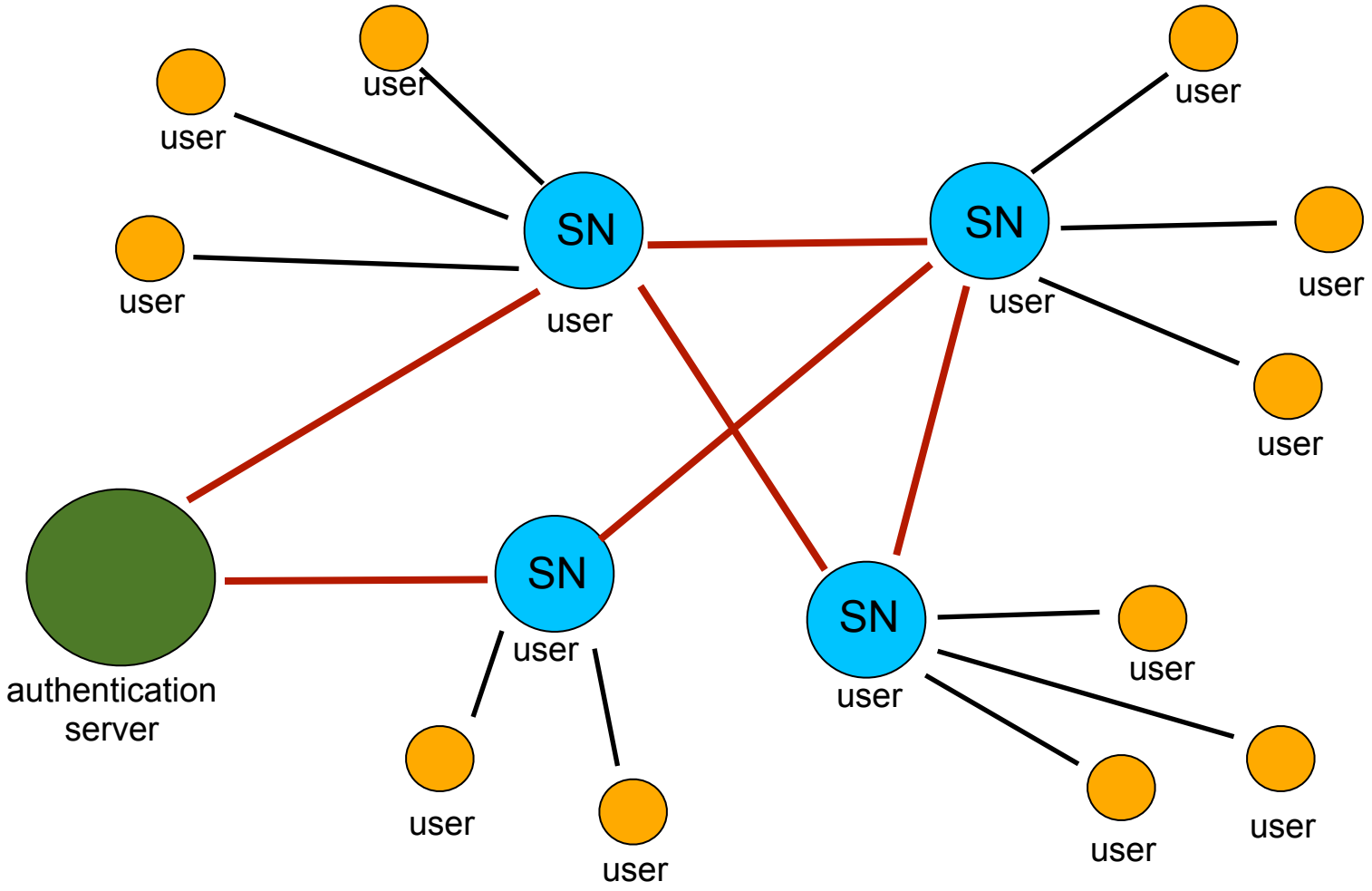
Target Installation

Skype-like UC service

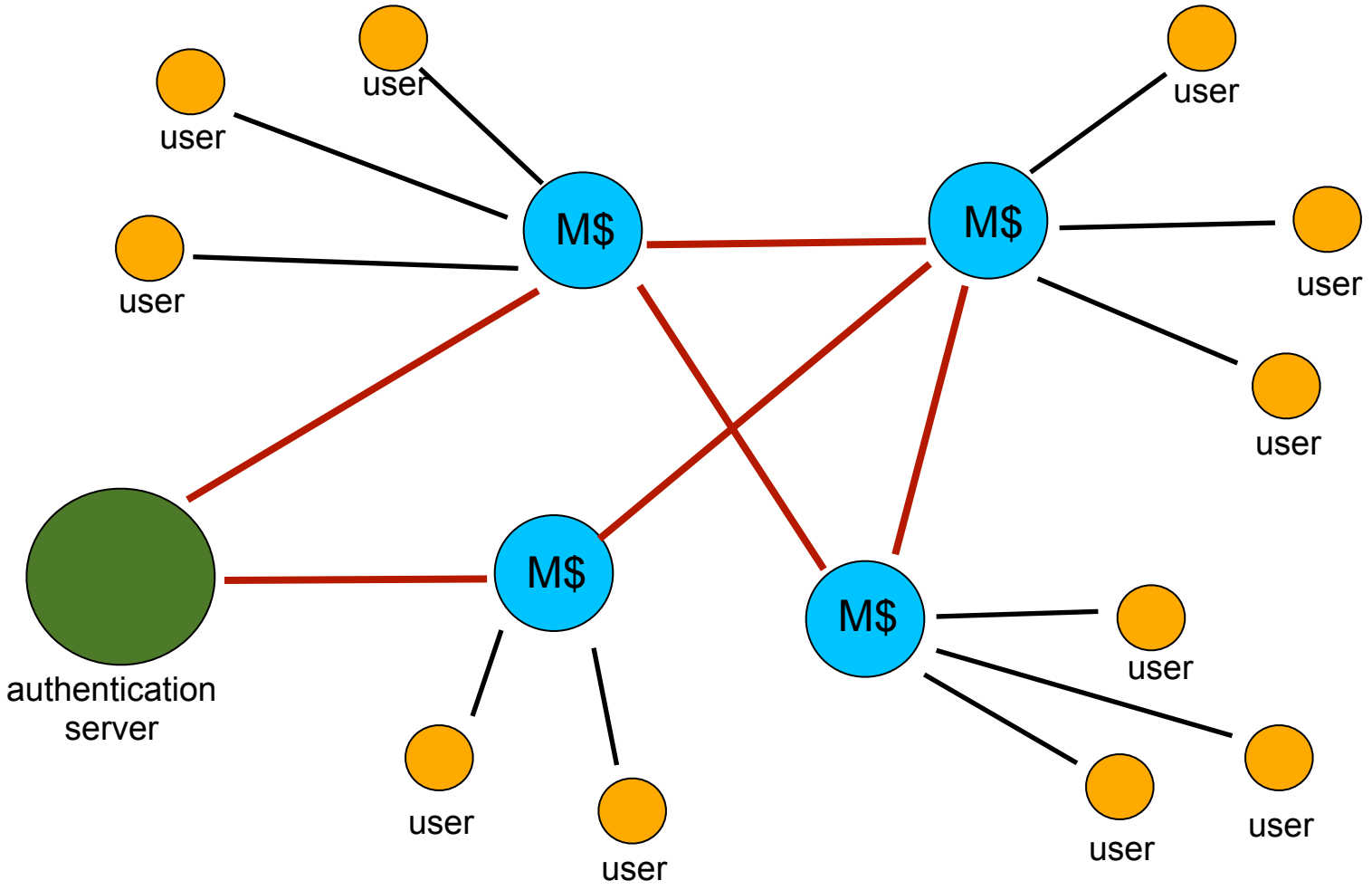
What's there?

- the main services
  - instant messaging
  - voice calls
  - video calls
  - presence and buddy list
  - encrypted communication
- additional services
  - voice conferencing
  - video conferencing
  - desktop sharing
  - file transfer
  - calls to classic telephony network (pstn) - paid service

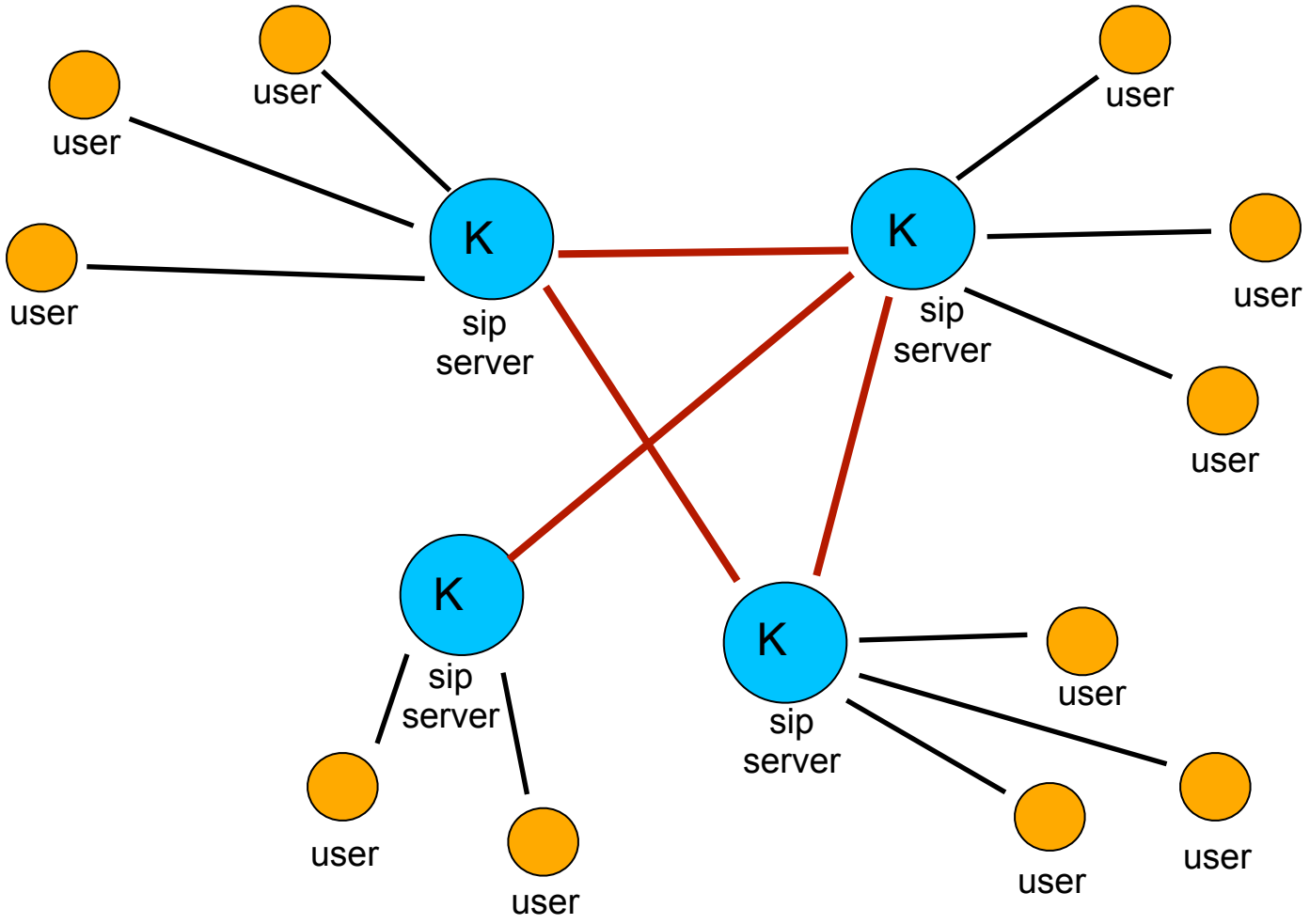
Skype Architecture



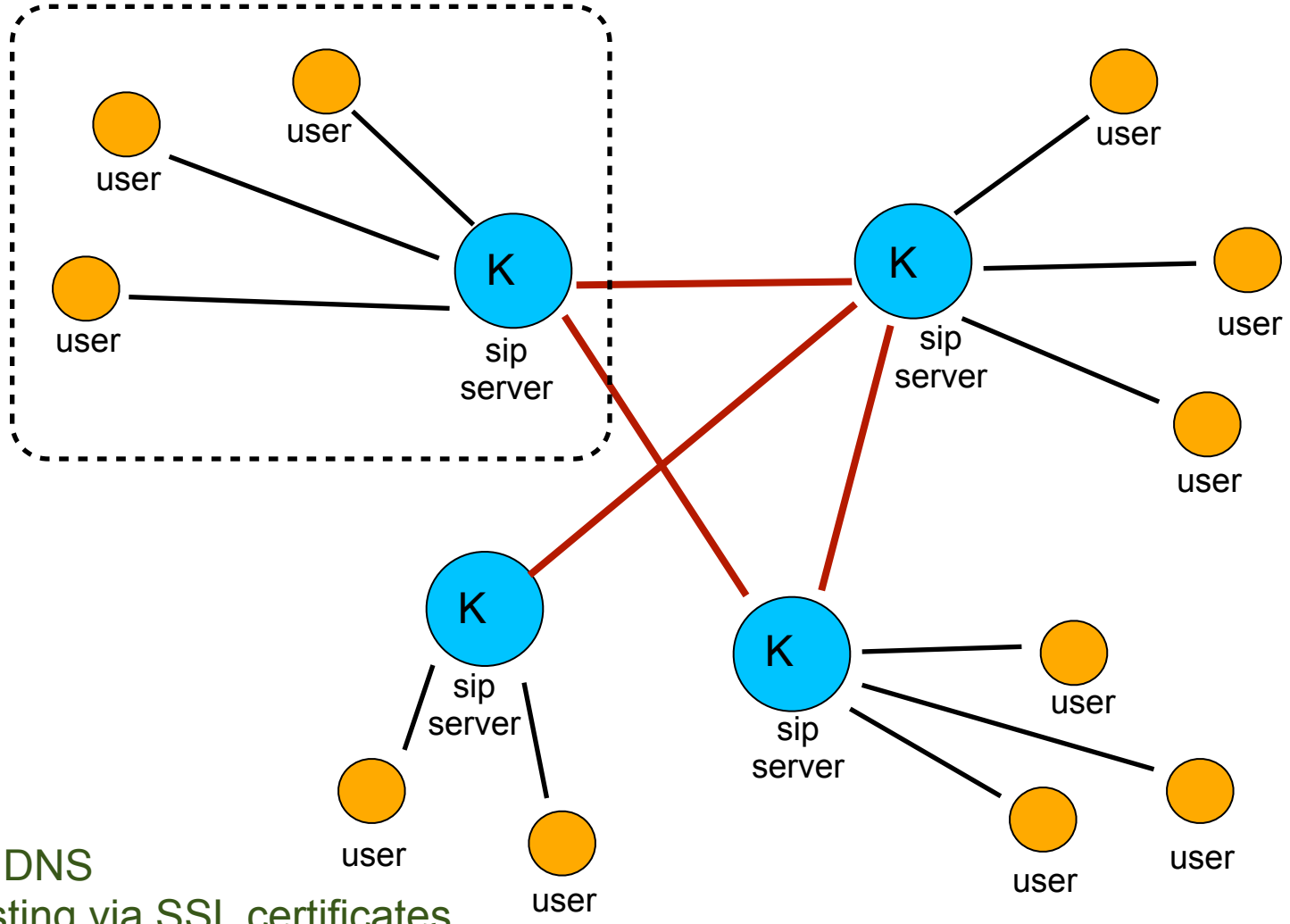
M$ Skype Architecture



SIP Peering Architecture



Building during workshop



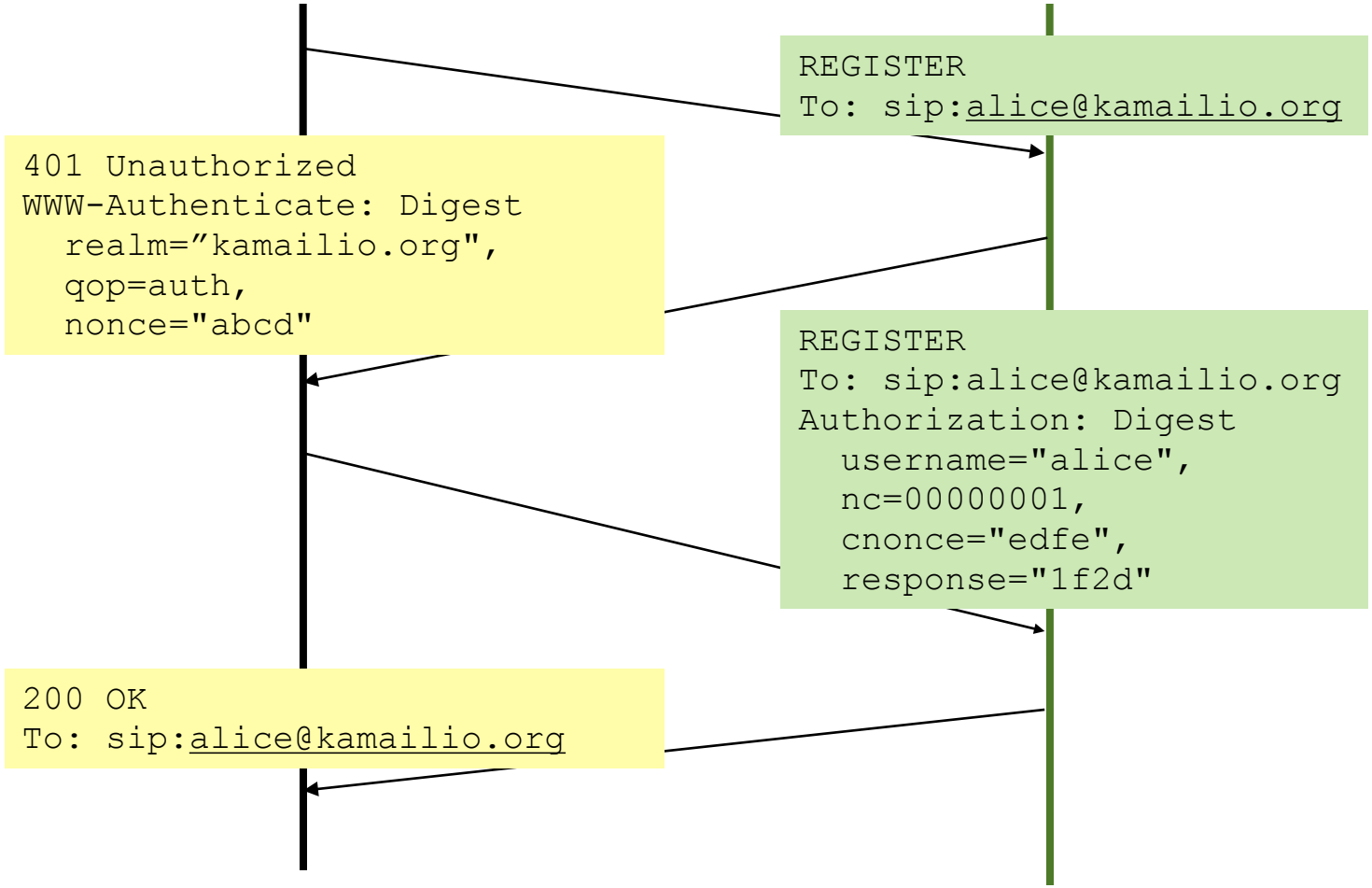
Peering

* via DNS
* trusting via SSL certificates

Secure Communication

Authorization and Confidentiality

Digest authentication



Security

Encrypted Transmission

TLS and Kamailio

- Dependencies
  - openssl, libssl
  - openssl-dev, libssl-dev
- Completely re-factored since v3.0.0
  - scalability
  - simplified installation
  - flexible configuration (modparams or own config file)
  - asynchronous communication
- Kamailio Config Requirementents
  - compile and install TLS module
  - load TLS module
    - loadmodule "tls.so"
  - enable tls in config
    - disable_tls=0
    - listen=tls:10.0.0.1:5061
  - *default config file -- add:* #!define WITH_TLS

TLS Config

- Config by module parameters
  - set tls attributes via modparam
  - tls method (sslv1, sslv2, tlsv1), ciphers list, certificates, timeouts, ...

```
...
loadmodule "tls.so"

modparam("tls", "private_key", "/etc/kamailio/kamailio-selfsigned.key")
modparam("tls", "certificate", "/etc/kamailio/kamailio-selfsigned.pem")
modparam("tls", "ca_list", "/etc/kamailio/calist.pem")
```

TLS Config

- Config by .ini-like file
  - dedicated file which can contain tls attributes
  - can include config for more than one server
  - can include config specific for clients

```
...
modparam("tls", "config", "/etc/kamailio/tls.cfg")
...
```

```
[server:default]
method = TLSv1
verify_certificate = no
require_certificate = no
private_key = default_key.pem
certificate = default_cert.pem
ca_list = default_ca.pem
```

TLS module

```
xlog("L_INFO","$$tls_version                    = '$tls_version'\n");
xlog("L_INFO","$$tls_description                = '$tls_description'\n");
xlog("L_INFO","$$tls_cipher_info                = '$tls_cipher_info'\n");
xlog("L_INFO","$$tls_cipher_bits                = '$tls_cipher_bits'\n");
xlog("L_INFO","$$tls_peer_subject               = '$tls_peer_subject'\n");
xlog("L_INFO","$$tls_peer_issuer                = '$tls_peer_issuer'\n");
xlog("L_INFO","$$tls_my_subject                 = '$tls_my_subject'\n");
xlog("L_INFO","$$tls_my_issuer                  = '$tls_my_issuer'\n");
xlog("L_INFO","$$tls_peer_version               = '$tls_peer_version'\n");
xlog("L_INFO","$$tls_my_version                 = '$tls_my_version'\n");
xlog("L_INFO","$$tls_peer_serial                = '$tls_peer_serial'\n");
xlog("L_INFO","$$tls_my_serial                  = '$tls_my_serial'\n");
xlog("L_INFO","$$tls_peer_subject_cn            = '$tls_peer_subject_cn'\n");
xlog("L_INFO","$$tls_peer_issuer_cn             = '$tls_peer_issuer_cn'\n");
xlog("L_INFO","$$tls_my_subject_cn              = '$tls_my_subject_cn'\n");
xlog("L_INFO","$$tls_my_issuer_cn               = '$tls_my_issuer_cn'\n");
```

Out there

- TLS Tutorial - The README for TLS Module
  - http://kamailio.org/docs/modules/stable/modules/tls.html

GREEN VoIP Research Project at Columbia University

Some interesting results:

- one instance of SIP server with *500 000 online users* (mixed users – behind and not NAT routers) – consumed energy *210W*
- one instance of SIP server with *1 000 000 online users* (no NAT involved) – consumed energy *190W*
- on a 32-bit machine with 4GB of memory and with 2.5GB reserved for SIP server, the server could support *43 000 simultaneous TLS connections* – consumed energy *209W*

http://www.kamailio.org/w/2011/05/green-voip-energy-efficiency-and-performaces-of-v3-0/

Installation

OS Example

- ***Ubuntu 10.04***

- work as root to avoid access issues
- apt-get - command to install/remove packages
  - man apt-get
  - apt-cache search packagename
- you can install/reinstall whatever you need
- text editors
  - gvim, gedit
- useful tools
  - wireshark, ngrep
- sip softphones
  - x-lite, twinkle, ekiga

Installation from GIT

- *Note: apt-get commands are valid for Debian/Ubuntu*
- requires root access
- prerequisites
  - - git client
    - apt-get install git-core
  - - gcc compiler and build tools
    - apt-get install gcc make
  - - flex
    - apt-get install flex
  - - bison
    - apt-get install bison
  - - libmysqlclient-dev
    - apt-get install libmysqlclient-dev
  - - libssl-dev
    - apt-get install libssl-dev

Getting Sources

- create working directory

mkdir -p /usr/local/src/kamailio-3.2
cd /usr/local/src/kamailio-3.2

- download sources from GIT repository

```
# git clone --depth 1 git://git.sip-router.org/sip-router kamailio
# cd kamailio
# git checkout -b 3.2 origin/3.2
```

- download tarball sources

  http://www.kamailio.org/pub/kamailio/3.2.3/src/kamailio-3.2.3_src.tar.gz

Compile and Install

- you can compile kamailio via

make FLAVOUR=kamailio include_modules="db_mysql" cfg

make all

- you can get a full output of the compilation process using:

make Q=0 all

- when the compilation is ready, install kamailio with the following command: *make install*

```
# make Q=0 all
bison -d -b cfg cfg.y
cfg.y: conflicts: 1 shift/reduce
flex cfg.lex
Compiling action.c
gcc -g -O9 -funroll-loops -Wcast-align -Wall -minline-all-stringops -falign-loops -ftree-vectorize -mtune=athlon64    -
      DNAME='"kamailio"' -DVERSION='"3.2.3"' -DARCH='"i386"' -DOS='"linux"' -DCOMPILER='"gcc 4.1.2"' -
      D__CPU_i386 -D__OS_linux -D__SMP_yes -DCFG_DIR='"/usr/local/etc/kamailio/"' -DPKG_MALLOC -
      DSHM_MEM  -DSHM_MMAP -DUSE_IPV6 -DUSE_MCAST -DUSE_TCP -DDISABLE_NAGLE -
      DHAVE_RESOLV_RES -DSTATISTICS -DF_MALLOC  -DSVNREVISION='"2:5870"'  -DFAST_LOCK -
      DADAPTIVE_WAIT -DADAPTIVE_WAIT_LOOPS=1024  -DHAVE_GETHOSTBYNAME2 -
      DHAVE_UNION_SEMUN -DHAVE_SCHED_YIELD -DHAVE_MSG_NOSIGNAL -
      DHAVE_MSGHDR_MSG_CONTROL -DHAVE_ALLOCA_H -DHAVE_TIMEGM -DHAVE_EPOLL -
      DHAVE_SIGIO_RT -DHAVE_SELECT -c action.c -o action.o
```

Post Installation Facts

- The binaries and executable scripts were installed in:
  - /usr/local/sbin
- These are:
  - kamailio - openser server
  - kamctl - script to manage and control kamailio server
- To be able to use the binaries from command line, make sure that '/usr/local/sbin' is set in PATH environment variable. You can check that with 'echo $PATH'. If not and you are using 'bash', open '/root/.bash_profile' and at the end add:
  - PATH=$PATH:/usr/local/sbin
  - export $PATH

Post Installation Facts

- Kamailio modules are installed in:
  - /usr/local/lib/kamailio/modules/
  - /usr/local/lib/kamailio/modules_k/
  - /usr/local/lib/kamailio/modules_s/
- The documentation and readme files are installed in:
  - /usr/local/share/doc/kamailio/
- The man pages are installed in:
  - /usr/local/share/man/man5/
  - /usr/local/share/man/man8/
- The configuration file was installed in:
  - /usr/local/etc/kamailio/kamailio.cfg

Maintenance

- requires ‘root’ privileges to execute following commands:

cd /usr/local/src/kamailio-3.2/kamailio

git pull origin

make all

make install

/etc/init.d/kamailio restart

- now you have the latest kamailio v3.2.x running on your system.
- notification about GIT commits are sent to the mailing list: sr-dev@lists.sip-router.org. Each commit notification contains the reference to the branch where the commit has been done. If the commit message contains lines like:

```
Branch: 3.2
```

- then an update has been made to kamailio version 3.2.x and it will be available to the public GIT in no time.

Client Application

JITSI

(SIP Communicator)

Jitsi

- ***http://www.jitsi.org***
  - download and install for your preferred OS
  - portable - Java application
  - multi-protocol support

Live Demo

Questions?

Contact

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