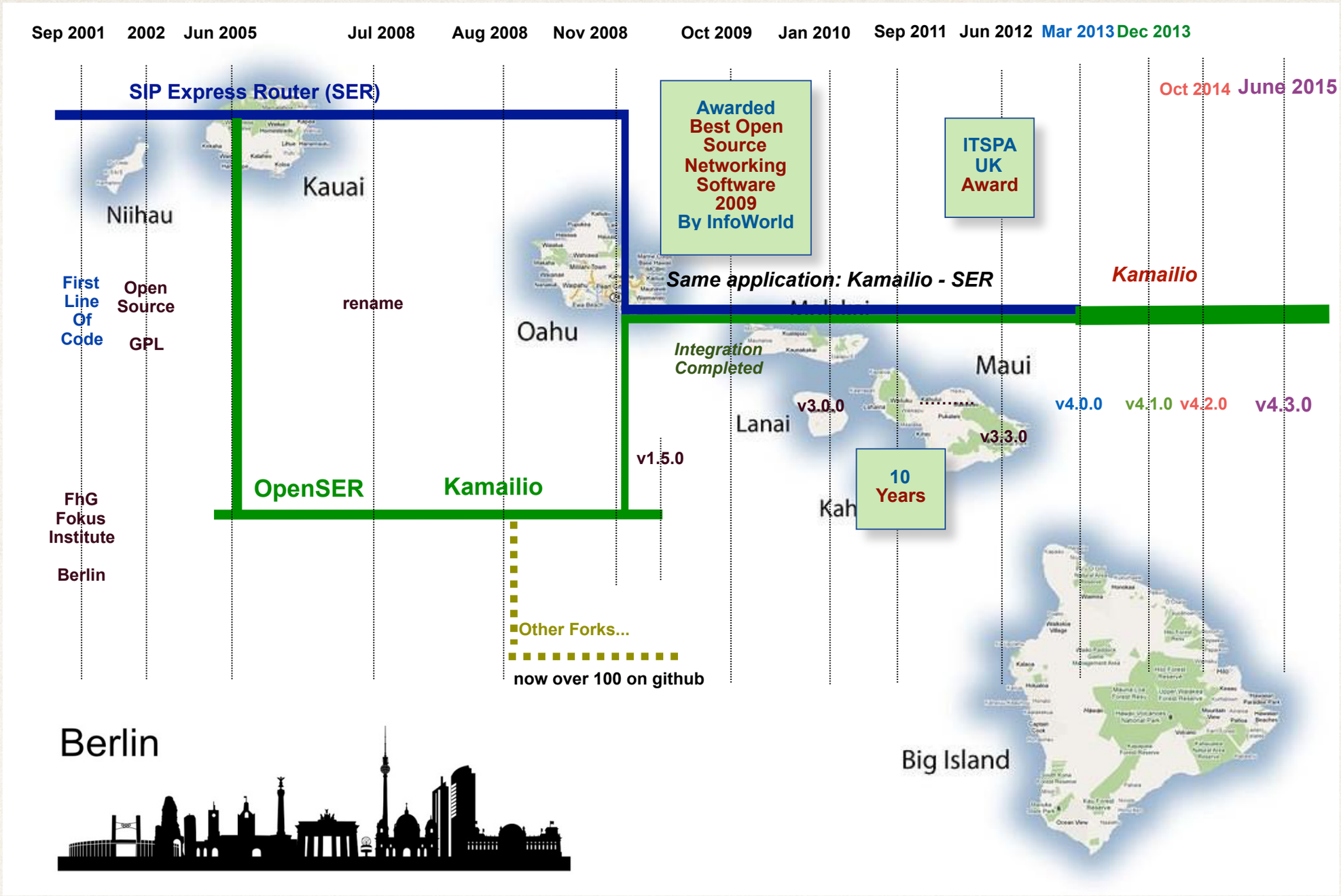
Designing High Performance RTC Signaling servers

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

KAMAILIO

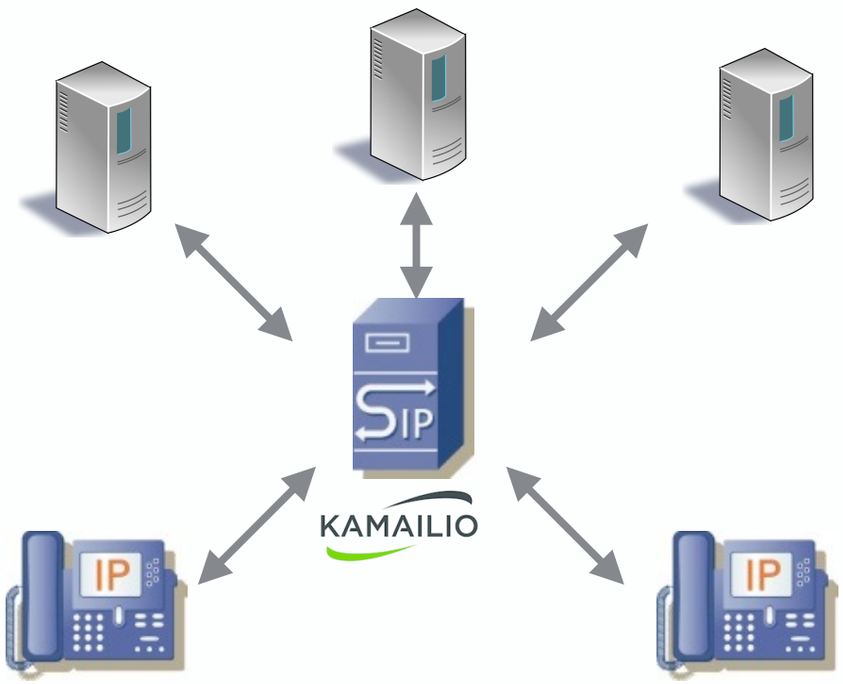
Fosdem - January 2016 - Brussels



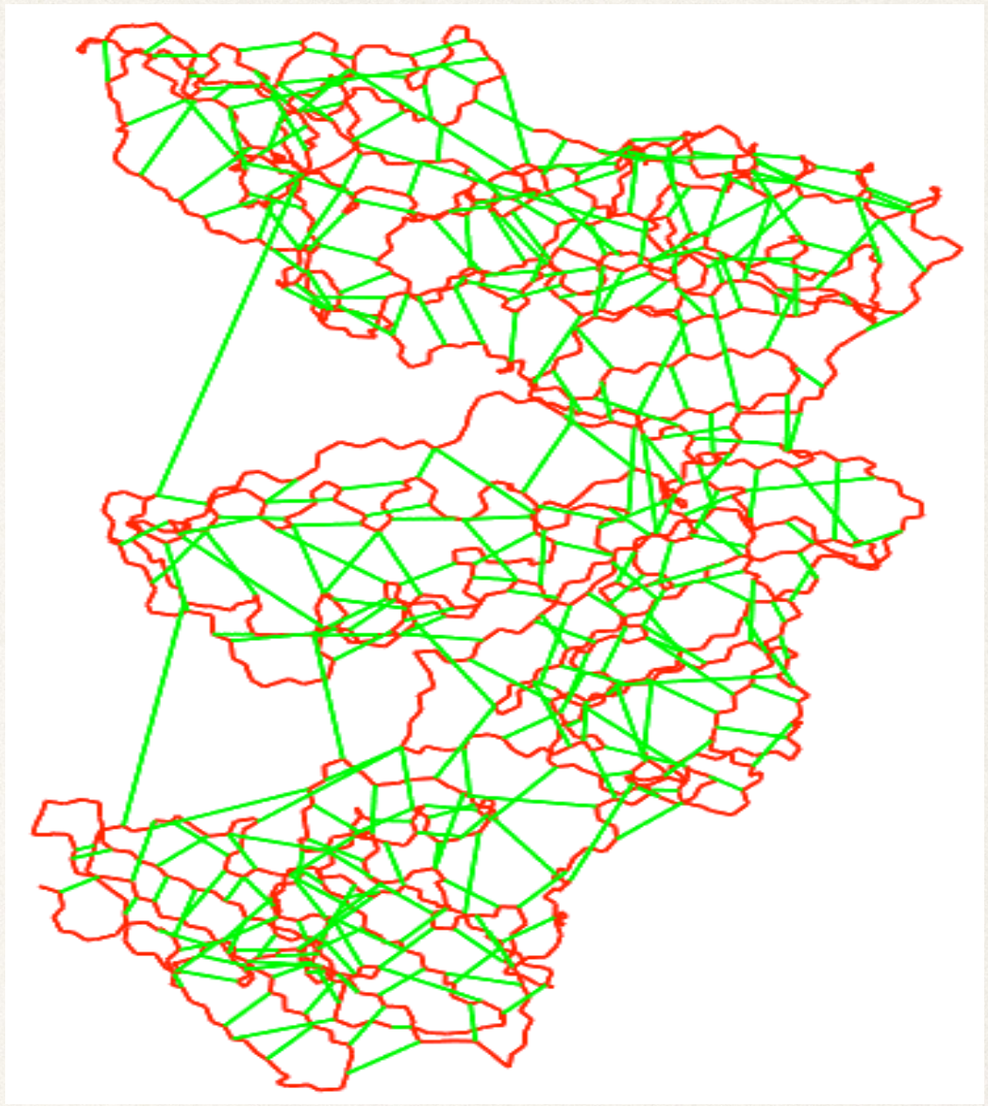
kamailio

continuos development since 2001

a very large set of features


SIP
KAMAILIO
IP
IP

- SIP signalling routing
  - fast
  - reliable
  - flexible
- In other words
  - not initiating calls
  - not answering calls
  - no audio-video processing



open source sip server

framework - toolkit

---

not designed as a typical telephony engine

Key Features

- Modular SIP Poxy, Registrar and Redirect server
- Designed for scalability and flexibility
- IPv4, IPv6, UDP, TCP, TLS, SCTP, WebSocket
- NAT Traversal, internal and external caching engines
- JSON, XMLRPC, HTTP APIs
- IMS Extensions, SIP-I/SIP-T, IM & Presence
- SQL and NoSQL backends
- Asynchronous processing (TCP/TLS, SIP routing), external event API
- Embedded interpreters (Lua, Perl, Python, .Net, Java)
- Load balancing, LCR, DID routing, Number portability

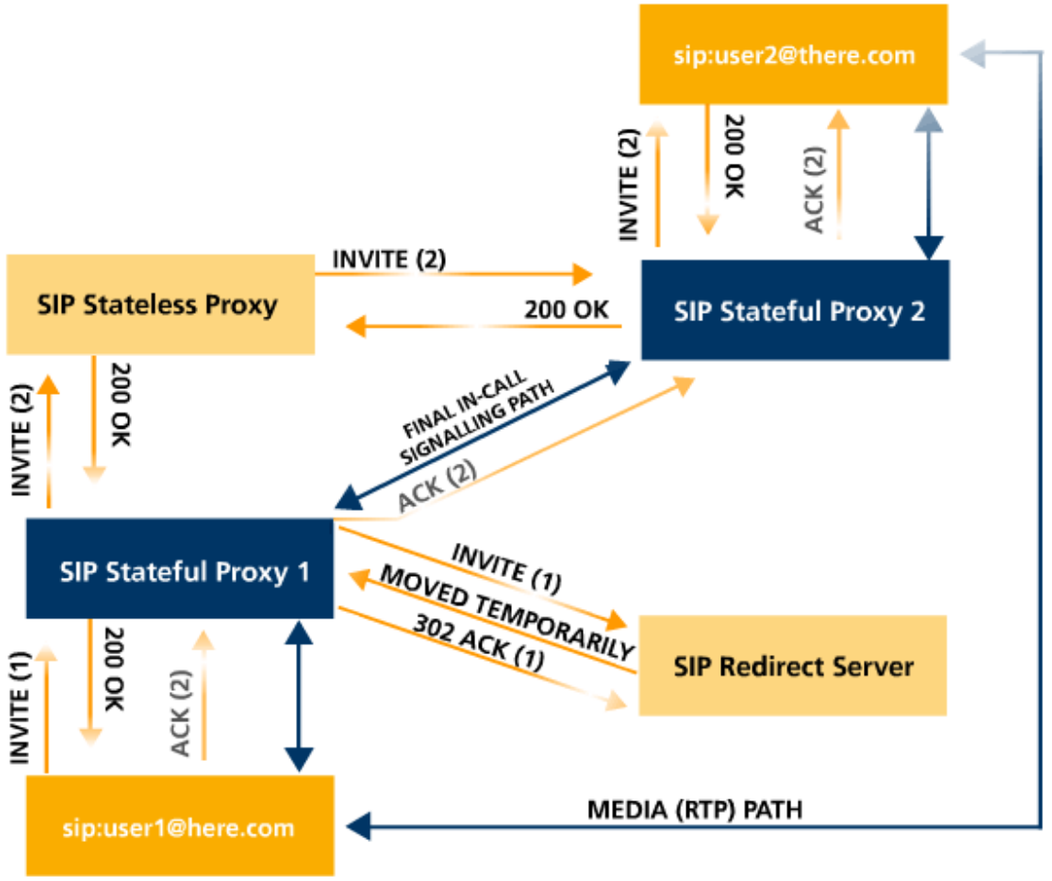
KAMAILIO WORLD
CONFERENCE & EXHIBITION
BERLIN, GERMANY, MAY 18-20, 2016

http://www.kamailioworld.com

YouTube KamailioWorld Channel
https://www.youtube.com/channel/UCElq4JNTPd7bs2vbfAAYVJA

Designing High Performance RTC Signaling Servers

- Protocol Message Parsing
- Memory Management
- Synchronisation
- Timers
- Caching



Message Parsing

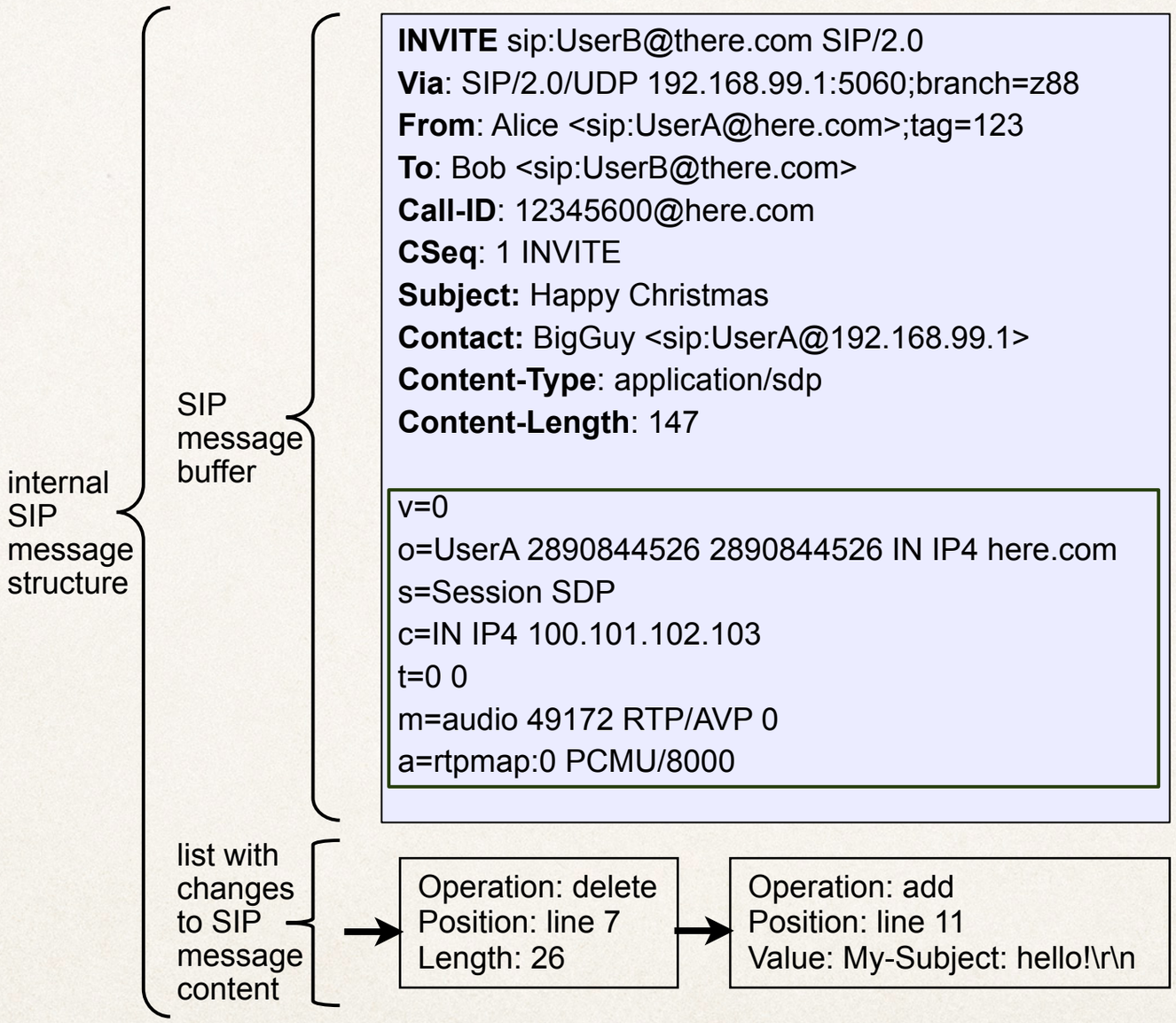
Lazy Parsing - Benefits

- parse only what is needed
- cache what was parsed
- no cloning - keep references to the receive buffer
- use private memory
- move to shared memory only if needed
- keep changes as diff list
- apply changes when sending out (or on demand)

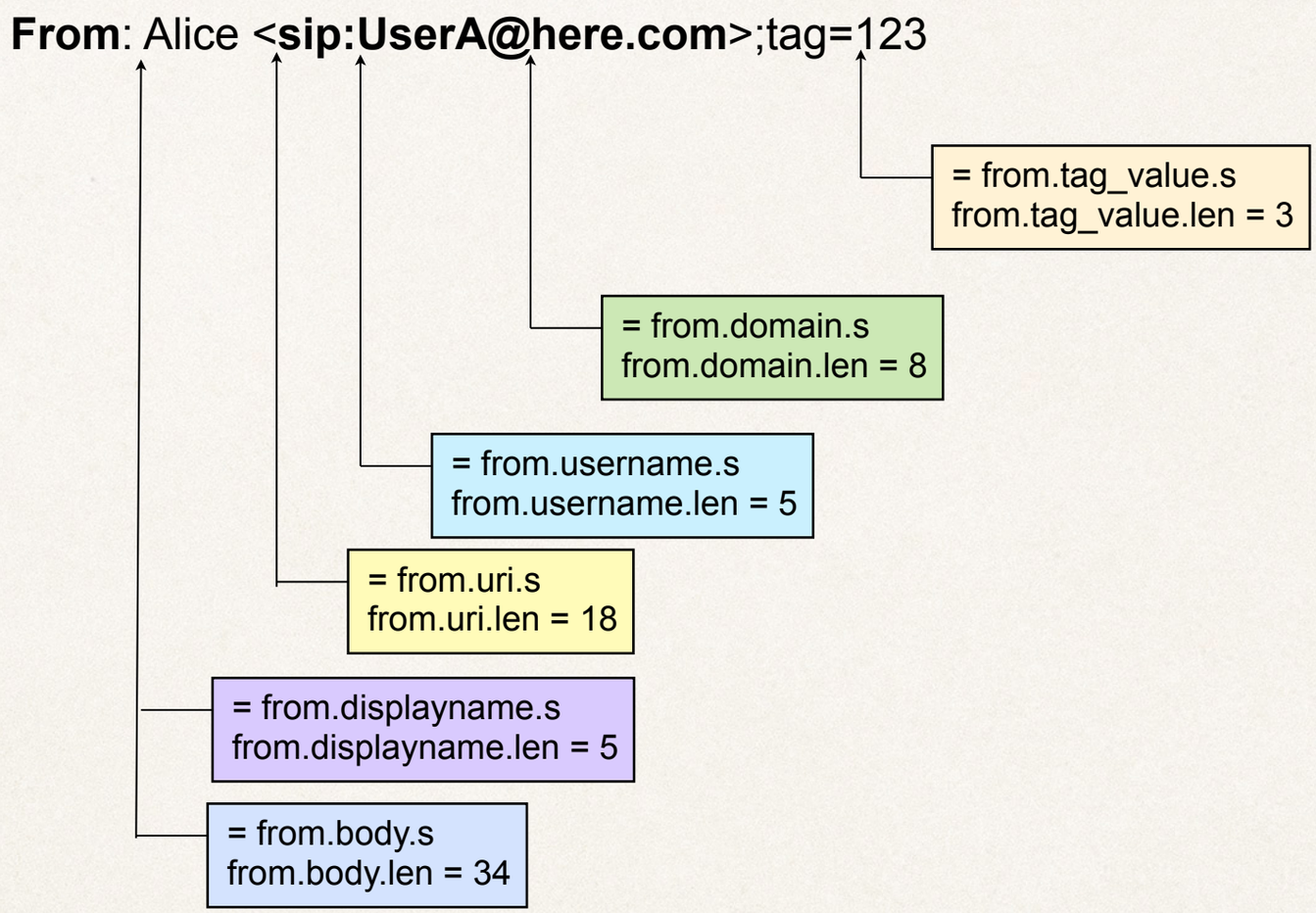
Lazy Parsing - Drawbacks

- not full validation of the message
- confusion for new comers regarding the changes

Lazy Parsing

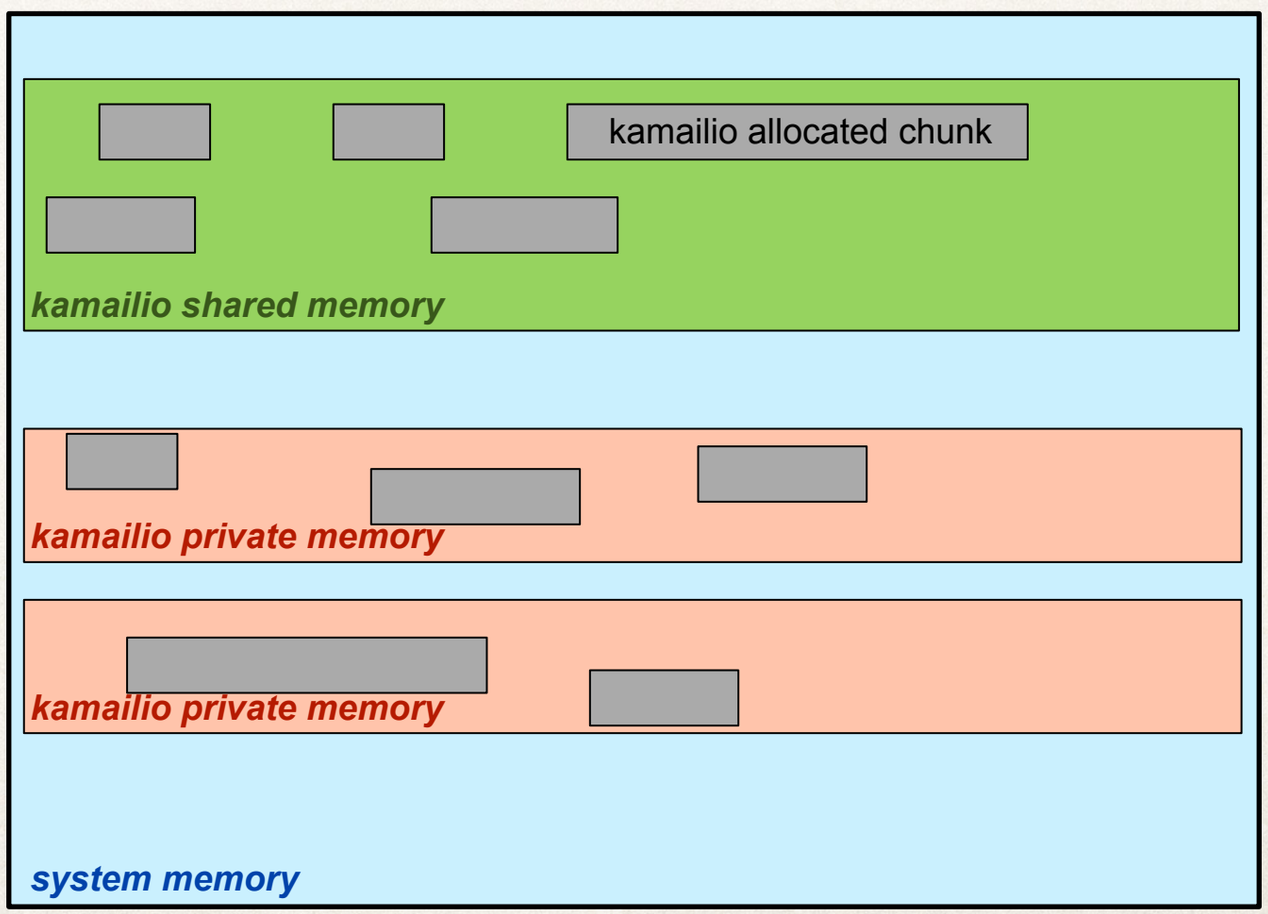


Lazy Parsing



Memory Management

- reserve zones of system memory for private and shared access
  - allocates smaller chunks in each zone
  - manage alloc, realloc, free and join operations



- benefits
  - optimizations for common chunk sizes
  - enable/disable join operations
  - select among different allocator algorithms (at startup)
    - fast malloc, quick malloc, tlsf malloc, doug lea malloc
  - avoid unnecessary locking
  - tunings for troubleshooting

- drawbacks
  - not easy to use various memory management tools
  - maintenance of code

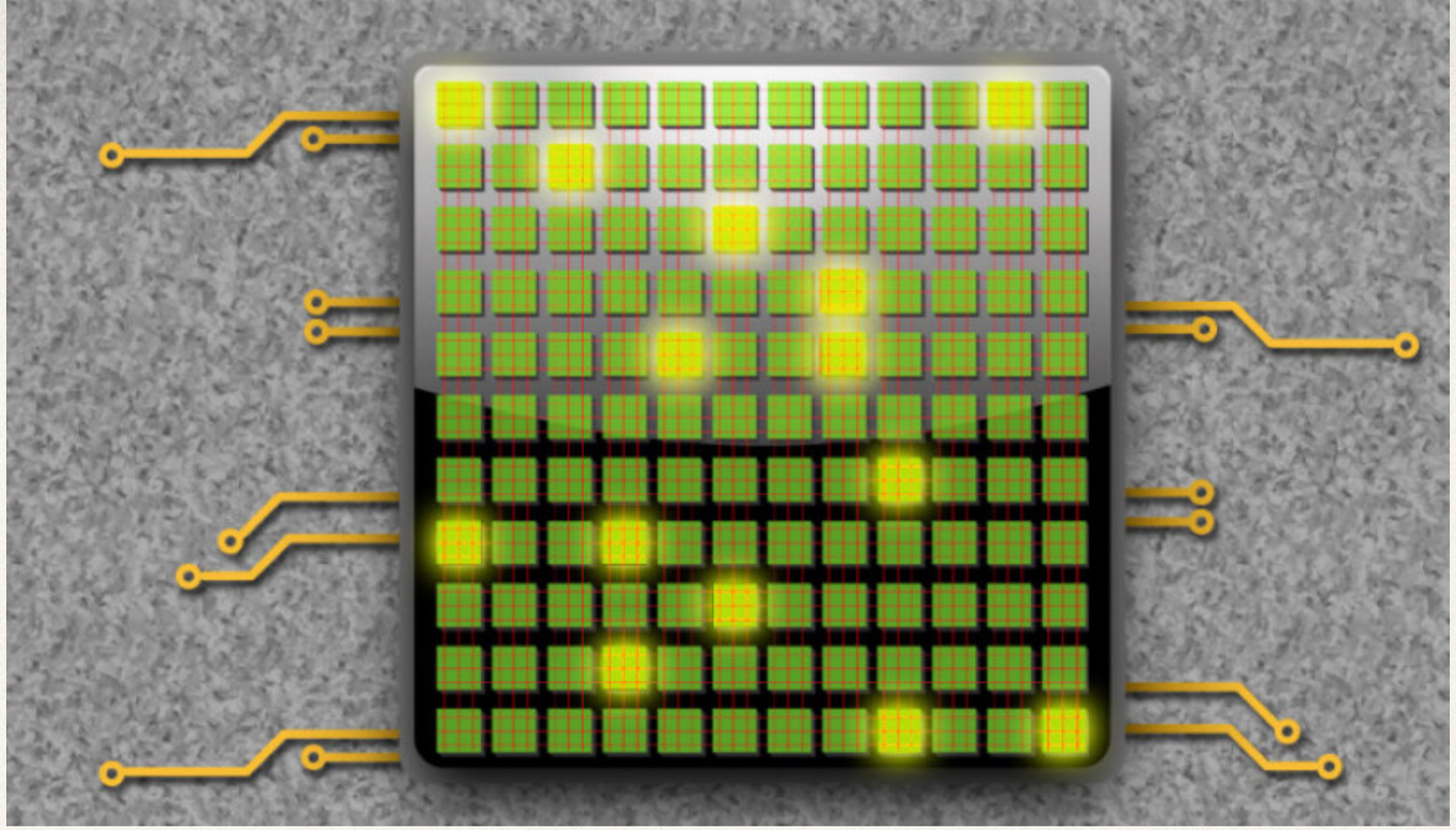
Synchronization

- relevant facts
  - private memory vs shared memory
  - mutexes - standard (posix) vs custom (busy loop)
  - message queues
  - memory bariers

Timers

- timer processes
  - lazy operations for many modules
    - keep alives, cleaning expired data
  - increase (or reduce) the timer interval

```
…
modparam("usrloc", "timer_procs", 4)
…
modparam("nathelper", "natping_processes", 6)
…
modparam("dialog", "timer_procs", 4)
modparam("dialog", "ka_timer", 10)
modparam("dialog", "ka_interval", 300)
…
```

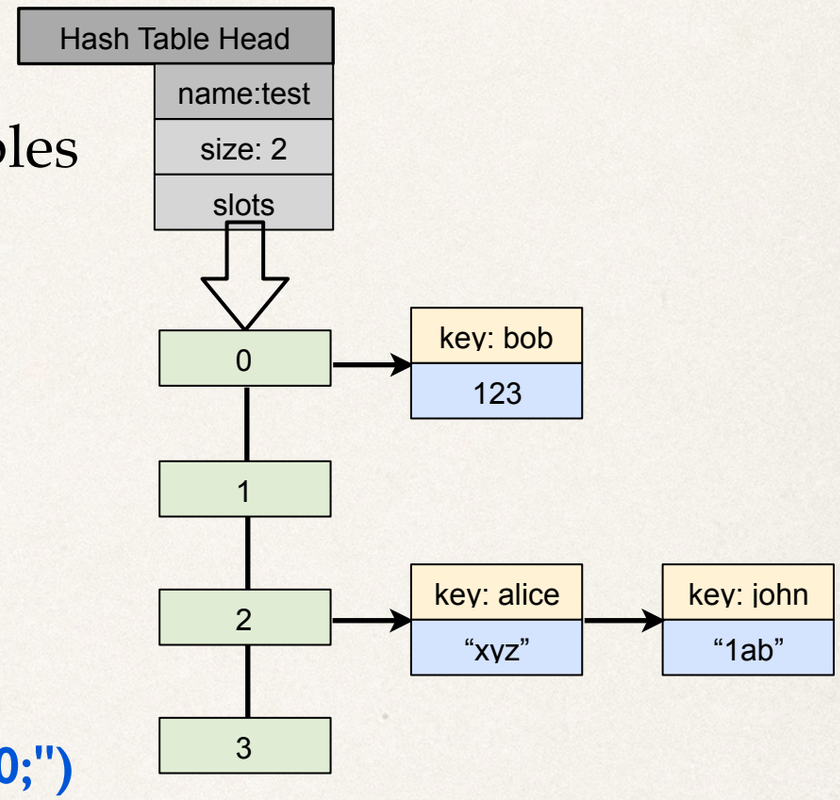


KAMAILIO

caching

- config or internal hash sizes
  - indexing of data in memory
  - location records (usrloc), dialogs, generic hash tables

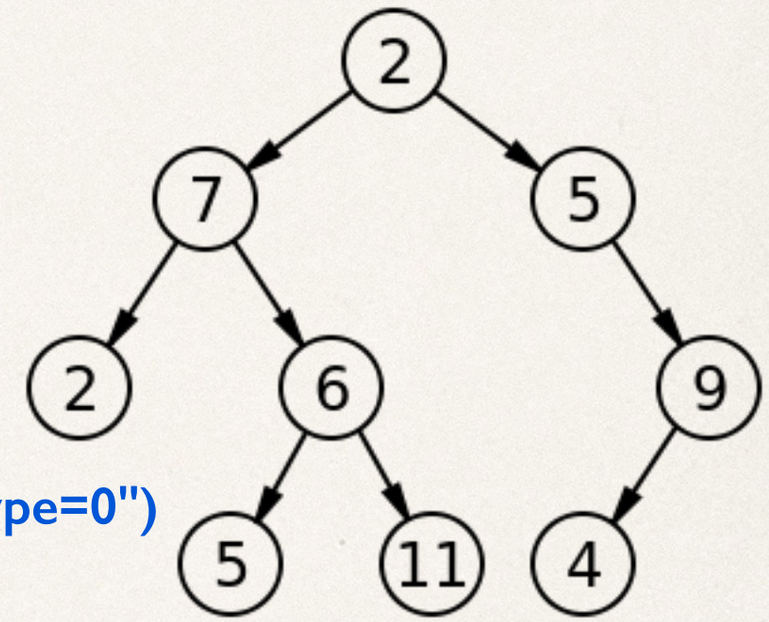
https://en.wikipedia.org/wiki/Hash_table




```
…
modparam("usrloc", "hash_size", 12)
…
modparam("htable", "htable", “a=>size=4;autoexpire=7200;")
modparam("htable", "htable", "b=>size=8;")
…
modparam("dispatcher", "ds_hash_size", 9)
…
```

https://en.wikipedia.org/wiki/Tree_(data_structure)

- config or internal tree structures
  - indexing of numbers in memory
  - pdt, mtree, userblacklist




```
…
loadmodule "mtree.so"
…
# ----- mtree params -----
modparam("mtree", "db_url", DBURL)
modparam("mtree", "mtree", "name=didmap;dbtable=didmap;type=0")
modparam("mtree", "char_list", "0123456789*+")
modparam("mtree", "pv_value", "$var(mtval)")
…
        if(mt_match("didmap", "$rU", "0")) {
                $avp(dsid) = $(var(mtval){s.int});
                route(DISPATCH);
                exit;
        }
…
```

- asynchronous processing
  - delegate the execution to other workers than sip routing processes
    - async module
    - tmx (suspend) - mqueue (transmit) - rtimer (process)
  - async database queries (mysql)
  - async http/jsonrpc interactions

```
…
async_workers=4
…
modparam("sqlops","sqlcon","ca=>dbdriver://username:password@dbhost/dbname")
sql_query_async("ca", "delete from domain");
…
modparam("acc", "db_insert_mode", 2)
…
```

http://www.kamailio.org/events/2014-KamailioWorld/day2/26-Daniel-Constantin.Mierla-Kamailio.cfg-Async.pdf

- bonus
  - children
    - number of worker processes
    - tcp - tls
      - max connections
      - file description limits
  - internal dns caching
    - blacklisting

- don’t forget
  - database indexes
  - syslog asynchronous mode
  - dns infrastructure availability
  - api services responsiveness

Thank you!

Questions?

@miconda

Kamailio World 2016 - A Special Edition

Kamailio Project

15 YEARS OF DEVELOPMENT

2001-2016

from SER to Kamailio

www.kamailioworld.com

the photos used inside this presentation were found on various web pages and there was no copyright or restriction of reuse mentioned

if some conditions were overseen, that was unintentional, please contact the author of the presentation for appropriate corrections