



Surfing Big Waves of SIP with Style

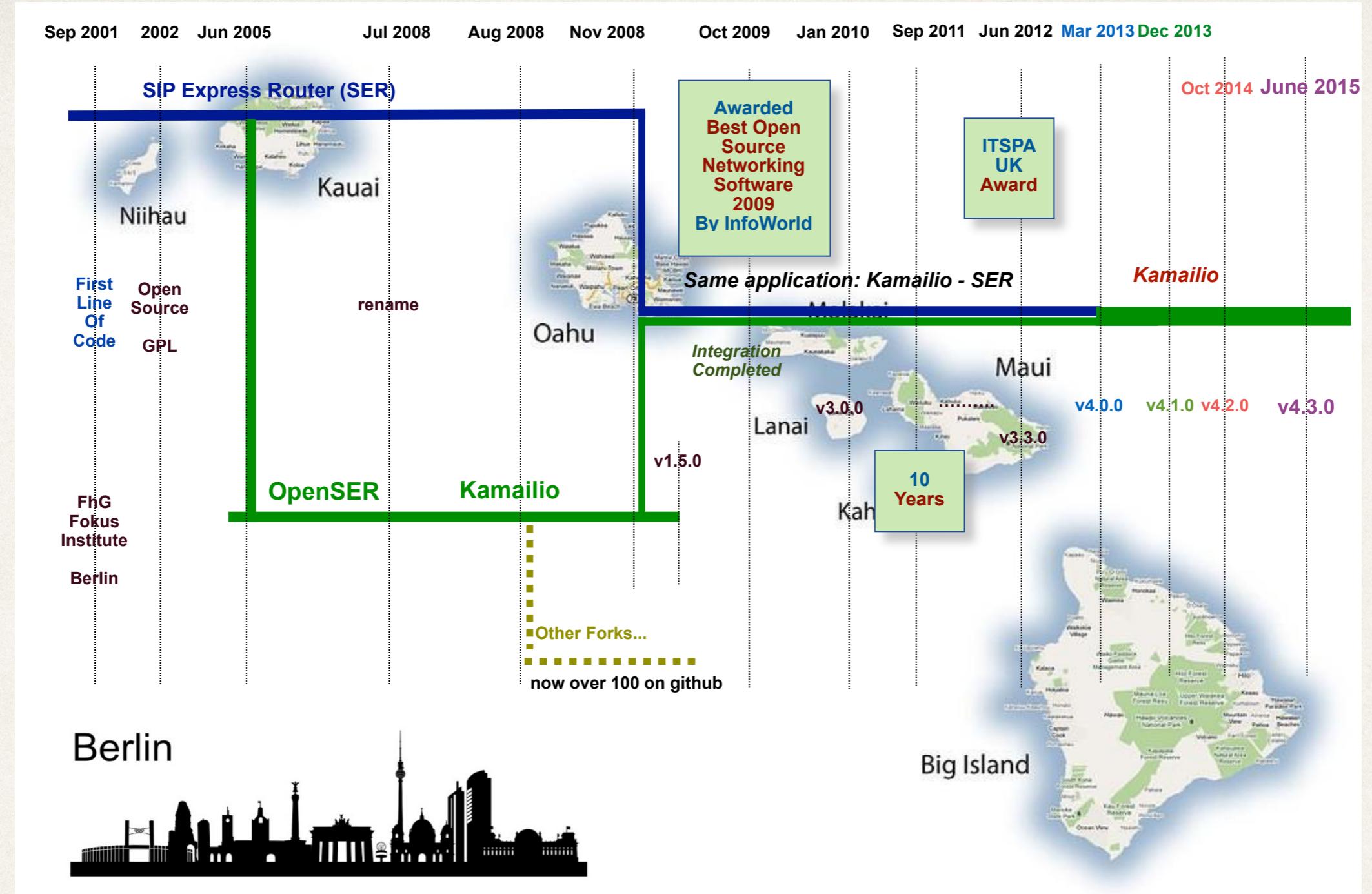
Daniel-Constantin Mierla

@miconda - www.asipto.com - Co-Founder Kamailio



October 2015



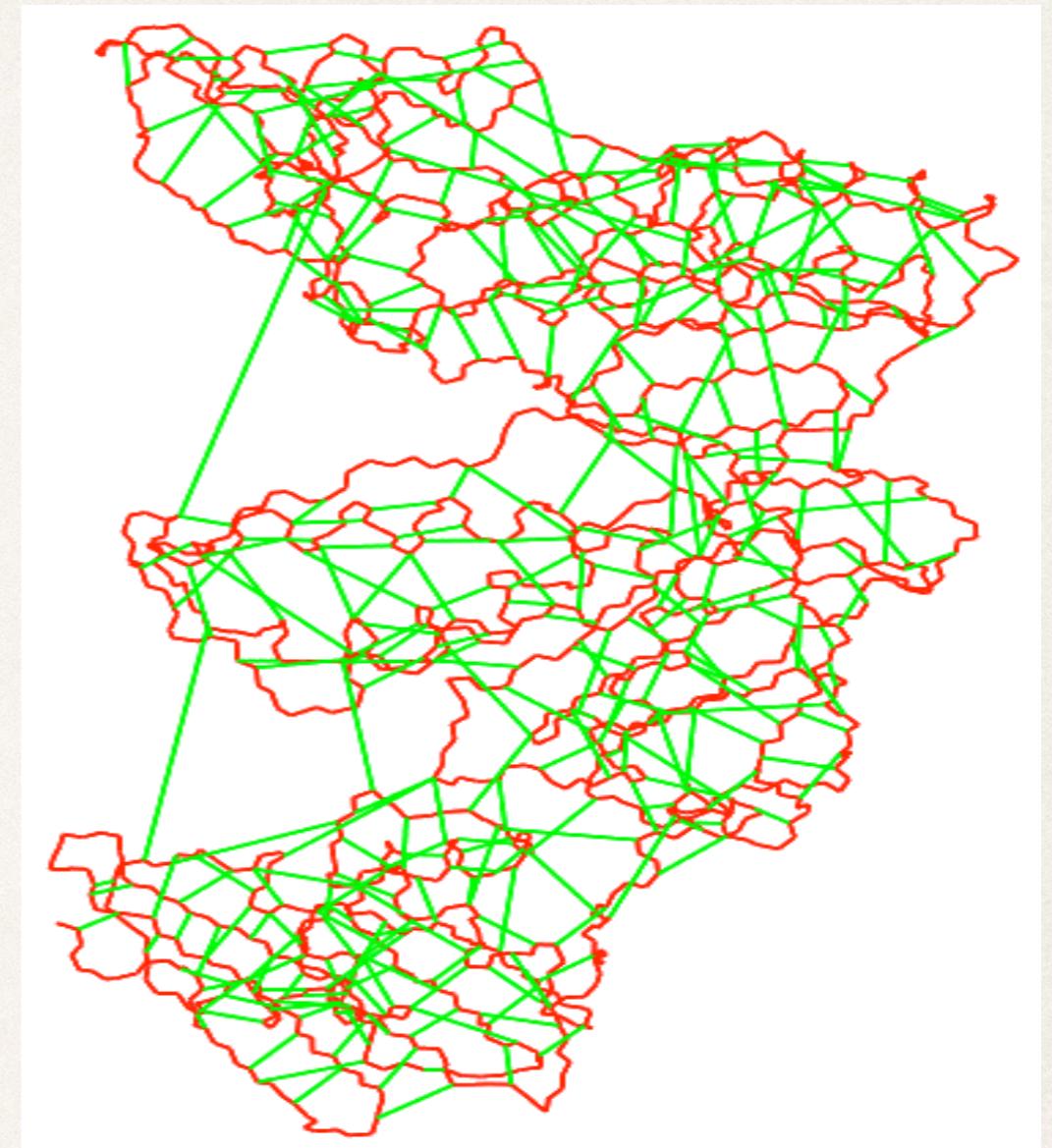


kamailio

continuous development since 2001

a very large set of features

- 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



Dealing with high SIP traffic volume

Detect and require a retry after a while

Increase performances of a Kamailio instance

Scale the RTC platform

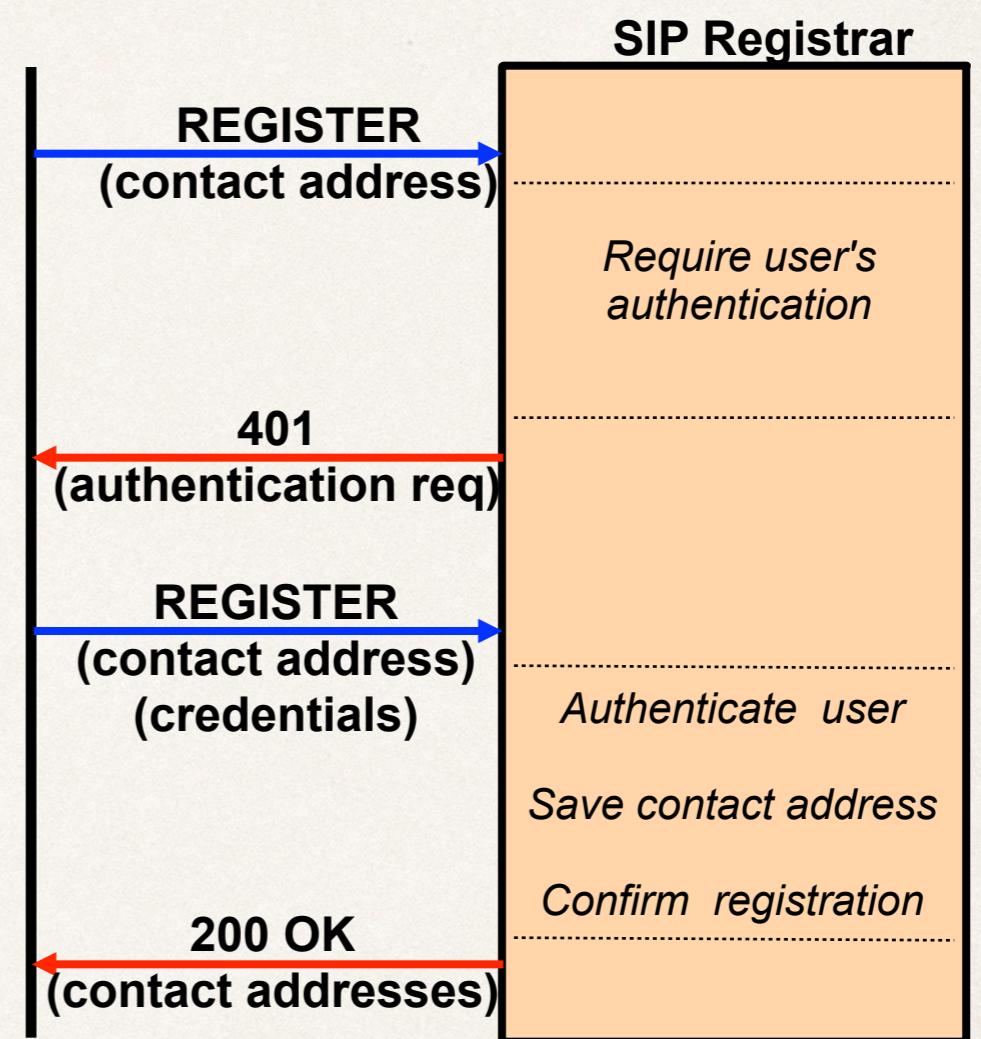




detect and require a retry after a while

Registration

sip registrar tunings



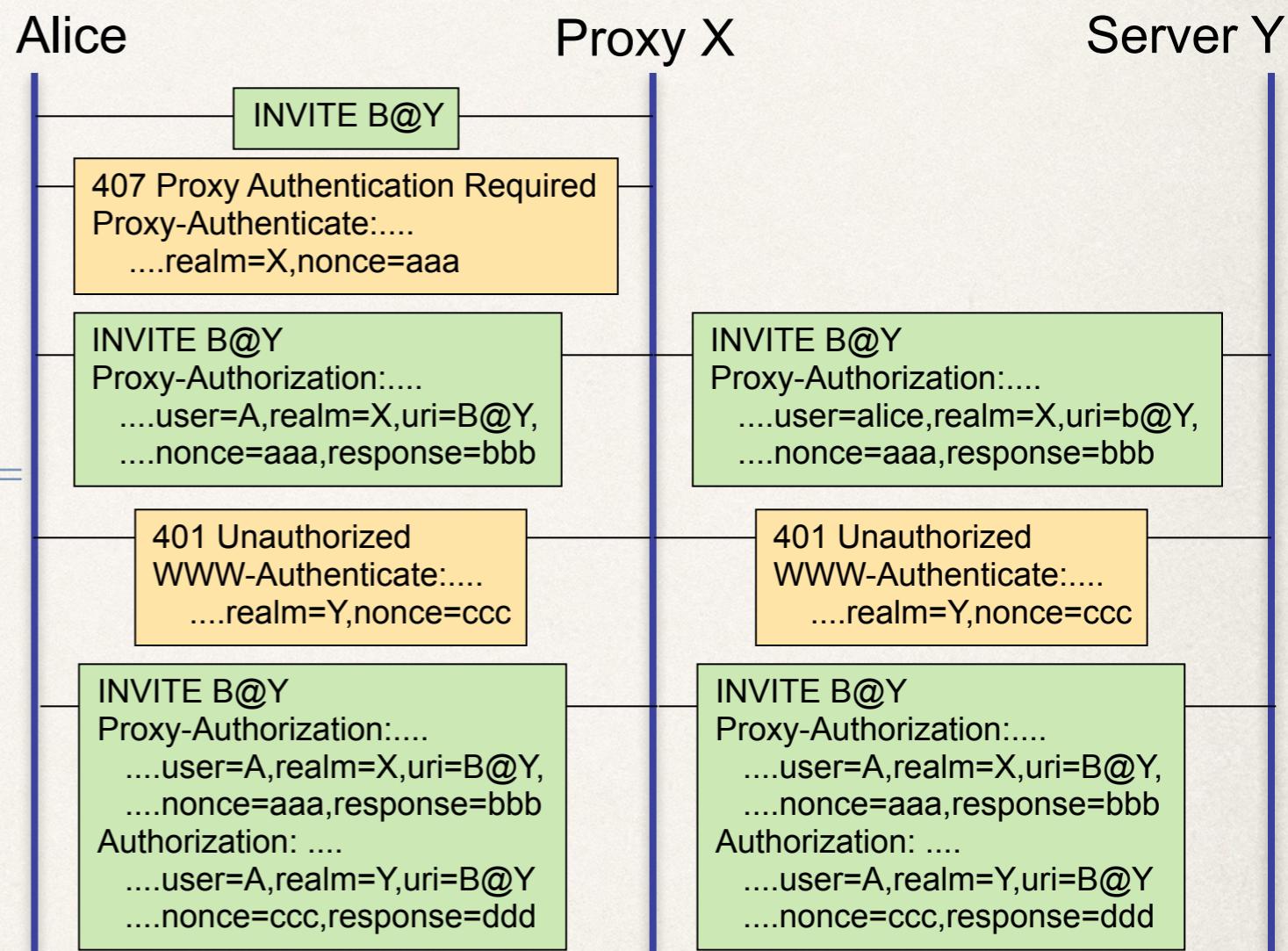
- ❖ registration storms

- ❖ reply to retry after
- ❖ randomize expires value

```
...
modparam("registrar", "expires_range", 30) # expires within [0.7*expires .. expires]
modparam("registrar", "retry_after", 30)
...
$var(retry) = 10 + ($RANDOM mod 300);
append_to_reply("Retry-After: \r\n");
send_reply("500", "Try later");
...
```

Authentication

sip authentication tunings



-
- ❖ caching user profile
 - ❖ password and other attributes
 - ❖ check first the cache, if not found, then fetch from database and store in memory

```
...
loadmodule "auth.so"
loadmodule "auth_db.so"
loadmodule "htable.so"
...
modparam("auth_db", "db_url", DBURL)
modparam("auth_db", "calculate_ha1", yes)
modparam("auth_db", "password_column", "password")
modparam("auth_db", "load_credentials", "$avp(password)=password")
modparam("auth_db", "use_domain", MULTIDOMAIN)
...
modparam("htable", "htable", "users=>size=10;autoexpire=300;")
...
```

```
...
route[AUTH] {
.....
    if (is_method("REGISTER") || from_uri==myself)
    {
        # authenticate requests
        if($sht(users=>$fU)!=$null) {
            if (!pv_auth_check("$fd", "$sht(users=>$fU)", "0", "I")) {
                auth_challenge("$fd", "0");
                exit;
            }
        } else {
            if (!auth_check("$fd", "subscriber", "I")) {
                auth_challenge("$fd", "0");
                exit;
            }
            $sht(users=>$fU) = $avp(password);
        }
        # user authenticated - remove auth header
        if(!is_method("REGISTER|PUBLISH"))
            consume_credentials();
    }
...
notrunc
```

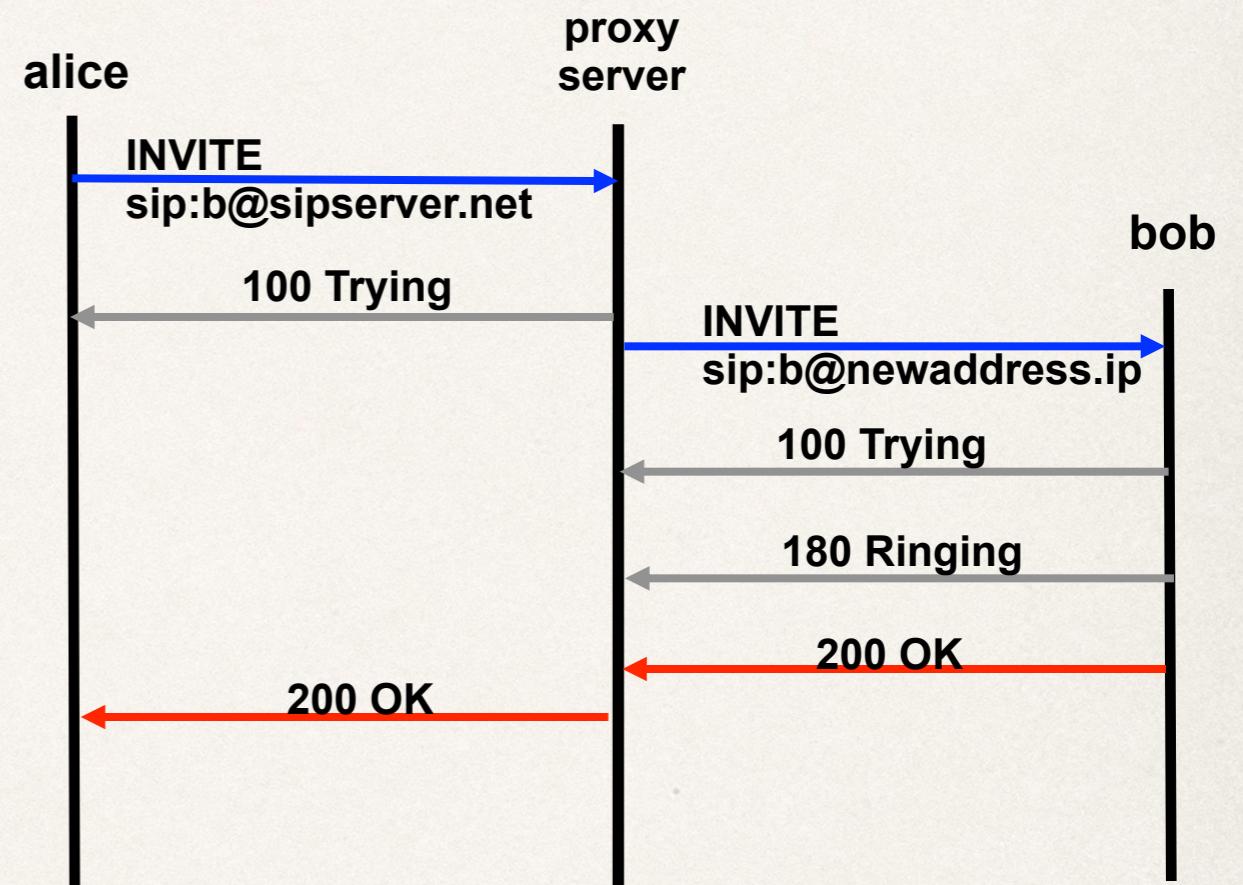
- ❖ client certificate or key based authentication

- ❖ avoid the auth challenge extra round trip
- ❖ tls with client certificate
- ❖ auth_xkey module

```
...
modparam("auth_xkeys", "xkey", "id=abc;name=xyz;value=secret;expires=72000")
...
...
auth_xkeys_add("X-My-Key", "abc", "sha256", "$Ri:$fu:$ru:$hdr(CSeq)");
...
if(!auth_xkeys_add("X-My-Key", "abc", "sha256", "$si:$fu:$ru:$hdr(CSeq)")) {
    send_reply("403", "Forbidden");
    exit;
}
remove_hf("X-My-Key");
...
```

Forwarding

sip proxy tunings



❖ pipelimit

- ❖ reply with retry after
- ❖ send redirect to another node

```
...
$var(retry) = 10 + ($RANDOM mod 300);
append_to_reply("Retry-After: \r\n");
send_reply("503", "Try later");
...
```

...

```
$var(limit) = 20;
if (!pl_check("$au", "TAILDROP", "$var(limit))) {
    pl_drop("10", "60");
    exit;
}
```

...

```
...
if (!pl_check("$au", "TAILDROP", "$var(limit))) {
    $ru = "sip:" + $rU + "@newserver.net";
    send_reply("302", "Moved temporarily");
    exit;
}
```

...



increase performances of a sip server instance

- ❖ 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)
...
```

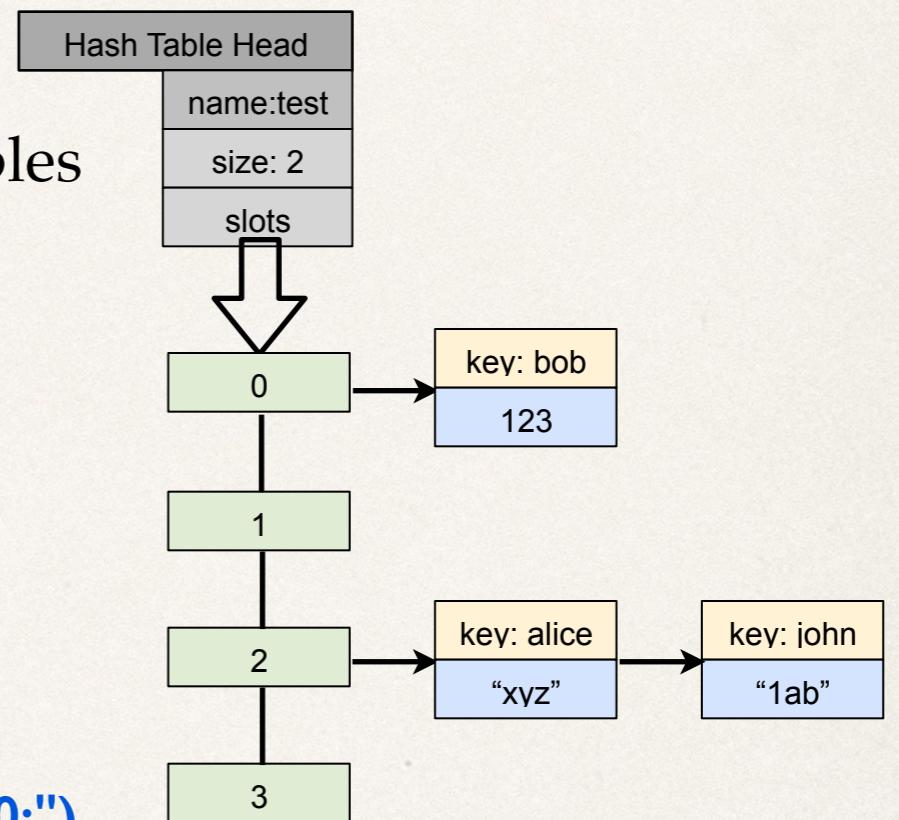
- ❖ internal hash sizes

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

- ❖ indexing of data in memory

- ❖ location records (usrloc), dialogs, generic hash tables

```
...
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)
...
```



- ❖ 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)
...
```

- ❖ 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



scale the rtc platform

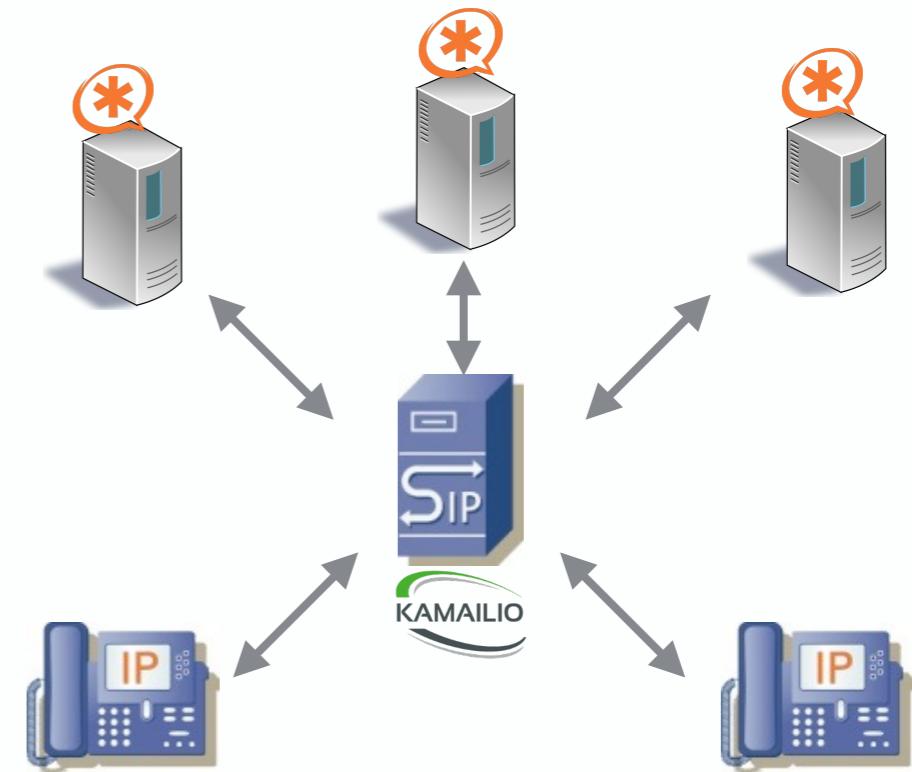
Load balancing

dispatcher module

- list of balancing nodes from file or database
- monitoring of nodes (activate/inactivate automatically)
- re-route in case of failure
- various algorithms: hashing, weight distribution, round robin, call load distribution, priority routing
- reload list of nodes without restart

```
# Dispatch requests
route[DISPATCH] {
    # round robin dispatching on gateways group 'I'
    if(!ds_select_dst("I","4")) {
        send_reply("404", "No destination");
        exit;
    }
    xdbg("--- SCRIPT: going to <$ru> via <$du>\n");
    t_on_failure("RTF_DISPATCH");
    route(RELAY);
    exit;
}

# Re-route in case of failure
failure_route[RTF_DISPATCH] {
    if (t_is_canceled()) {
        exit;
    }
    # next node - only for 500 or local timeout
    if (t_check_status("500") || (t_branch_timeout() && !t_branch_replied())) {
        if(ds_next_dst()) {
            t_on_failure("RTF_DISPATCH");
            route(RELAY);
            exit;
        }
    }
}
```



Externalize services

- ❖ lack of resources (e.g., development)
- ❖ centralized management
 - ❖ large volume of data
 - ❖ number portability database
 - ❖ billing, dids routing, etc

Parallel Forking

```
request_route {  
...  
    append_branch("sip:103@kamailio.org");  
    append_branch("sip:104@kamailio.org");  
  
    $var(newdst) = "sip:105@kamailio.org";  
    append_branch("$var(newdst)");  
  
    t_relay();  
}
```

the sip routing basics

Serial Forking

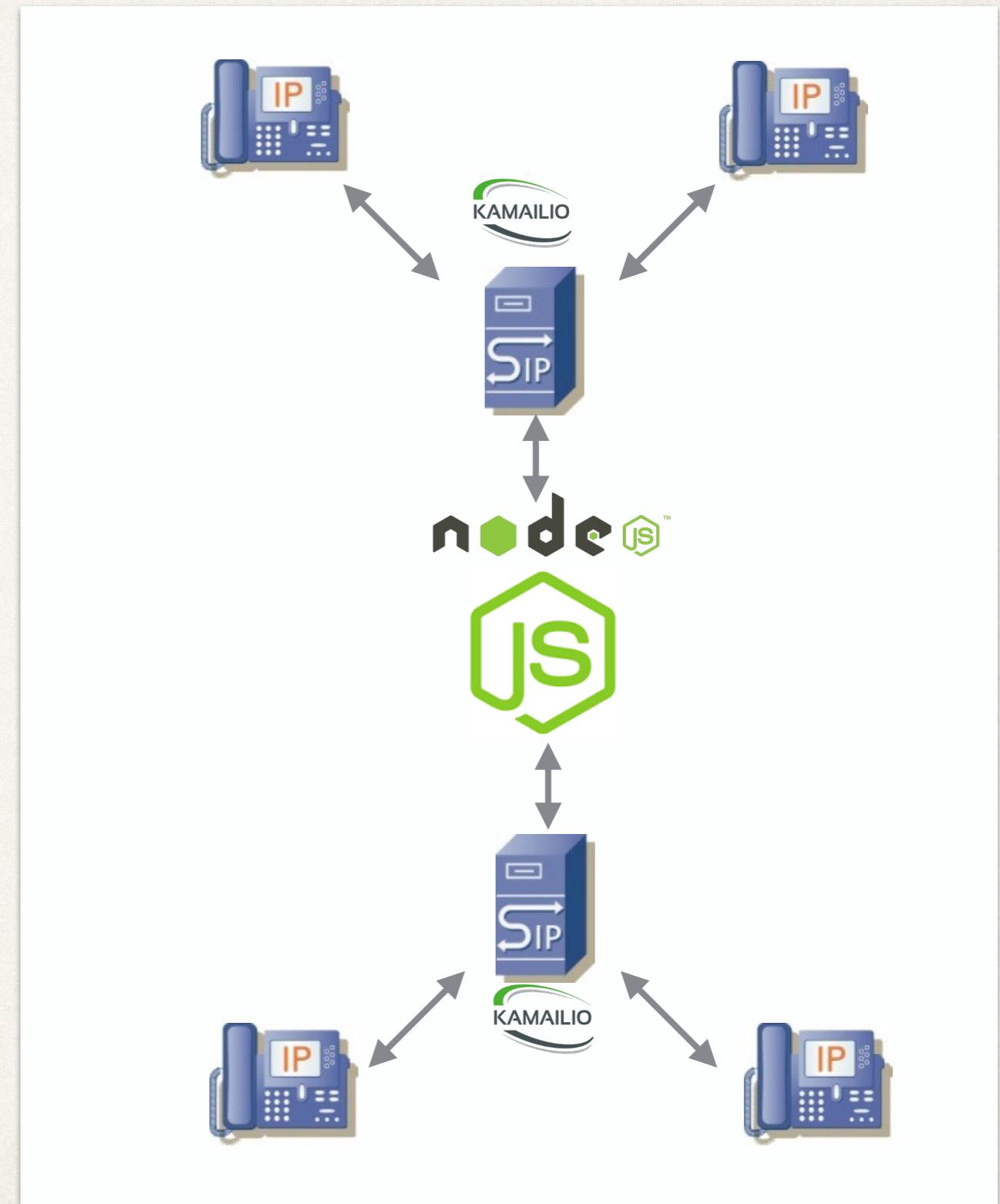
```
request_route {  
...  
  
    $ru = "sip:102@kamailio.org";  
    $xavp(newdst) = "sip:103@kamailio.org";  
    $xavp(newdst) = "sip:104@kamailio.org";  
  
    t_on_failure("RETRY");  
    t_relay();  
}  
  
failure_route[RETRY] {  
    if (t_is_canceled()) {  
        exit;  
    }  
    if($xavp(newdst) != $null) {  
        $ru = $xavp(newdst);  
        $xavp(newdst) = $null;  
        t_on_failure("RETRY");  
        t_relay();  
        exit;  
    }  
}
```

Building Blocks

- ❖ fetching the next routing hops
 - ❖ raw data
 - ❖ http/rpc results + parsing
 - ❖ structured data
 - ❖ rtjson
- ❖ processing
 - ❖ synchronous
 - ❖ config wait & act
 - ❖ asynchronous
 - ❖ evapi, mqueue & rtimer

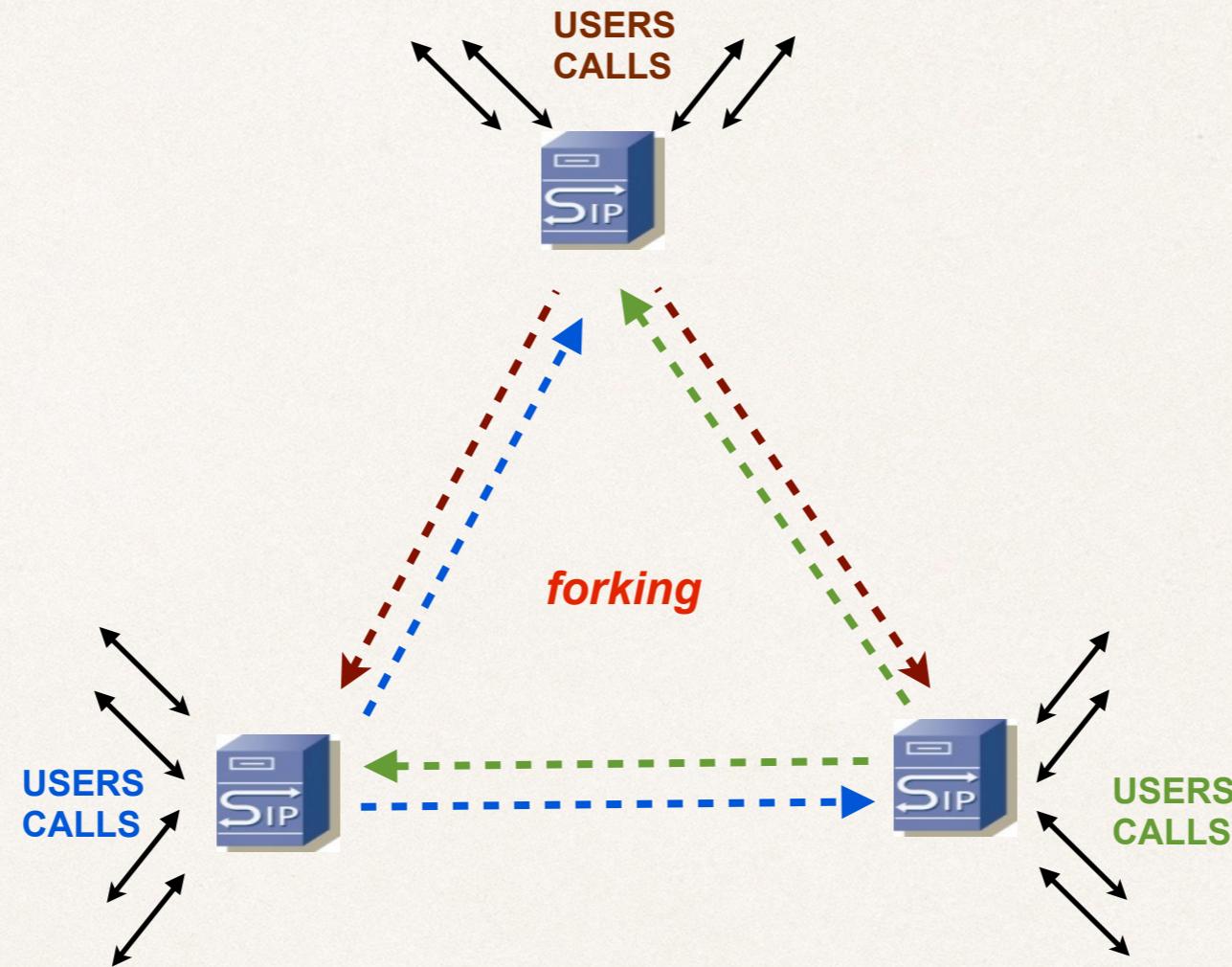


- ❖ kamailio config - combine:
 - ❖ evapi
 - ❖ jansson
 - ❖ rtjson
- ❖ external application
 - ❖ node.js
 - ❖ json data response
 - ❖ predefined structure

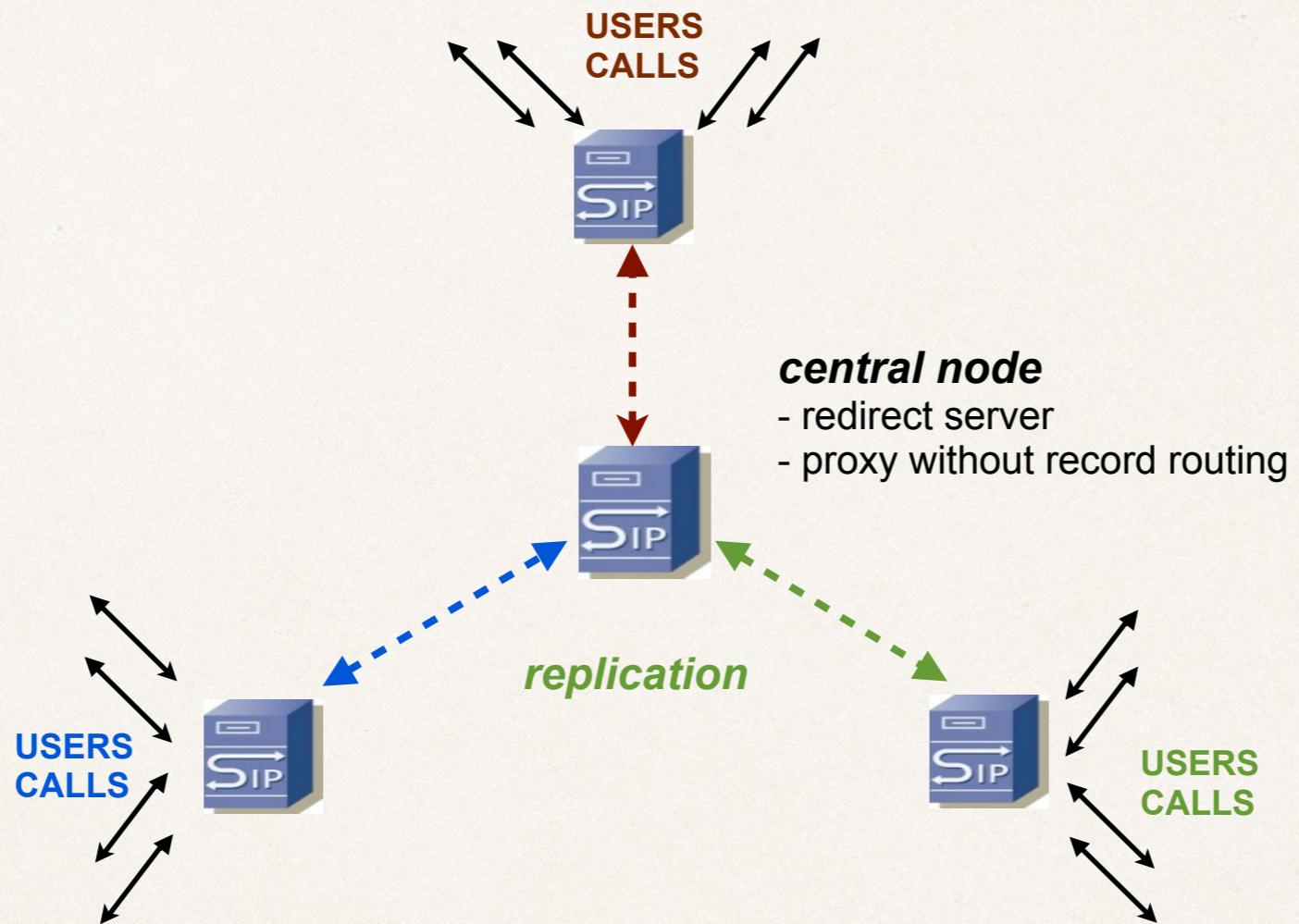


<http://kb.asipto.com/kamailio:k43-async-sip-routing-nodejs>

Platform Scalability



- ❖ not found on local location table and not coming from the other nodes
 - ❖ set destination to the other nodes (update \$ru and append_branch())
 - ❖ parallel forking (or serial forking with low transmission timeout)



```

# Handle SIP registrations
route[REGISTRAR] {
    if (is_method("REGISTER")) {
        if(isflagset(FLT_NATS)) {
            setbflag(FLB_NATB);
            # uncomment next line to do SIP NAT pinging
            ## setbflag(FLB_NATSIPPING);
        }
        if (!save("location"))
            sl_reply_error();

        $uac_req(method)="KUSRLOC"
        $uac_req(ruri)="sip:store@centralnode.kamailio.org";
        $uac_req(furi)="sip:server@server1.kamailio.org";
        $uac_req(hdrs)="Content-Type: text/kusrloc\r\n";
        pv_printf("$uac_req(body)", "$fu@fd");
        uac_send_req();

        exit;
    }
}

```

```

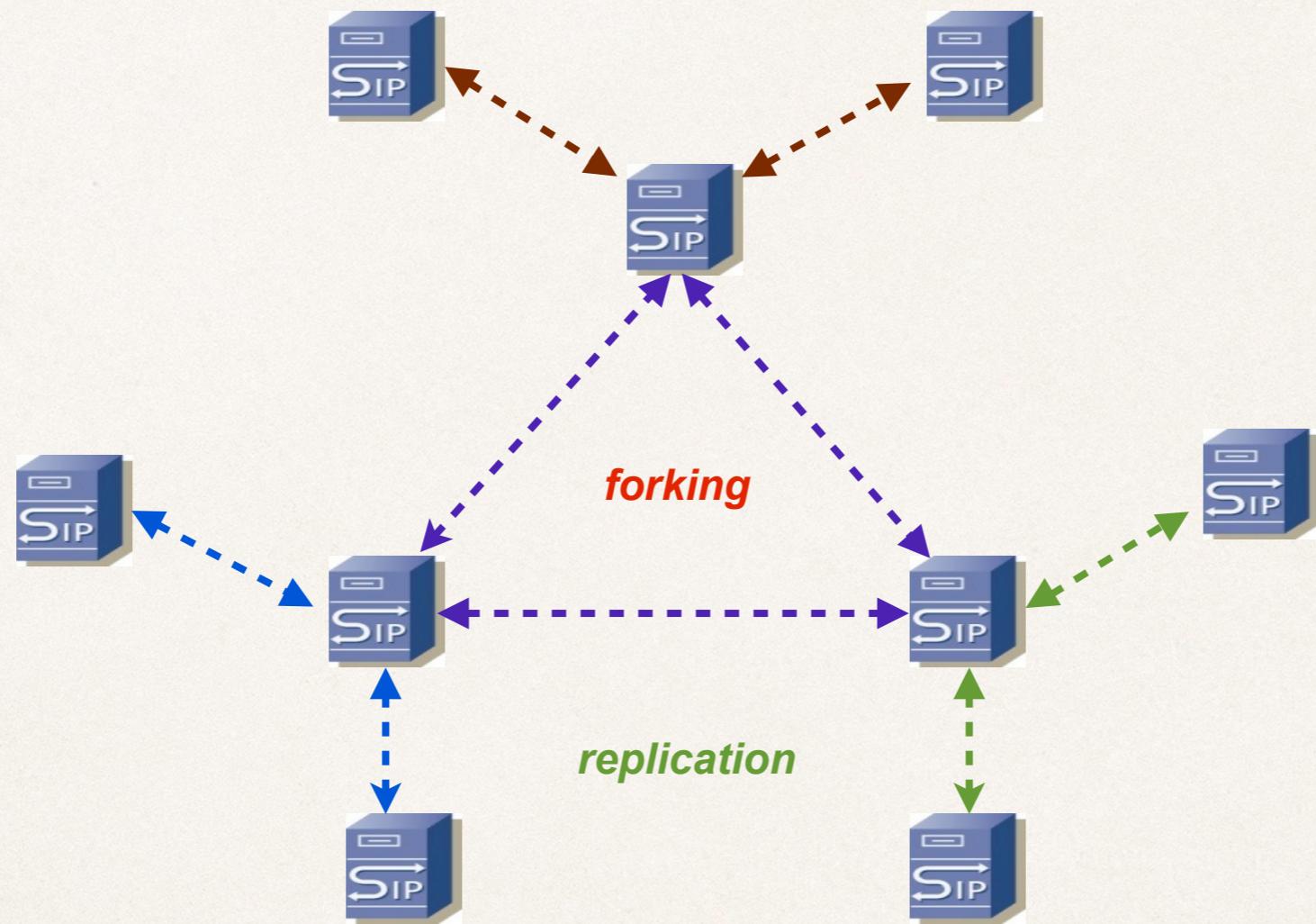
route[TOMAIN] {
    $du = "sip:CENTRALNODEIP";
    route(RELAY);
    exit;
}

# USER location service
route[LOCATION] {
#ifndef WITH_SPEEDDIAL
    # search for short dialing - 2-digit extension
    if($rU=~"[0-9][0-9]$")
        if(sd_lookup("speed_dial"))
            route(SIPOUT);
#endif
#ifndef WITH_ALIASDB
    # search in DB-based aliases
    if(alias_db_lookup("dbaliases"))
        route(SIPOUT);
#endif

$avp(oexten) = $rU;
if (!lookup("location")) {
    if(src_ip!=CENTRALNODEIP)
        route(TOMAIN);
}
}

```

```
modparam("htable", "htable", "kusrloc=>size=10;autoexpire=7200;")  
...  
request_route {  
    # add here ip authorization, etc...  
  
    # handle location update notification  
    if(method=="KUSRLOC" && $rU=="store") {  
        $sht(kusrloc=>$rb) = "sip:" + $si + ":" + $sp + ";transport=" + $pr;  
        send_reply("200", "Stored");  
        exit;  
    }  
    # handle standard SIP requests  
    if($sht(kusrloc=>$rU@$rd)!=$null) {  
        $du = $sht(kusrloc=>$rU@$rd);  
        t_relay();  
        exit;  
    }  
    send_reply("404", "Not found");  
    exit;  
}
```





tutorials
mailing lists
reference documentation

www.kamailio.org



<http://www.asipto.com/sw/kamailio-admin-book/>



KAMAILIO WORLD CONFERENCE & EXHIBITION

BERLIN, GERMANY, MAY 27-29, 2015

<http://www.kamailioworld.com>

YouTube KamailioWorld Channel

<https://www.youtube.com/channel/UCElq4JNTPd7bs2vbfAAVJA>

Thank you!
Questions?
@miconda



Kamailio World 2016 - Planning a Special Edition

Kamailio Project

15 YEARS OF DEVELOPMENT

2001-2016
from SER to Kamailio

www.kamailioworld.com