

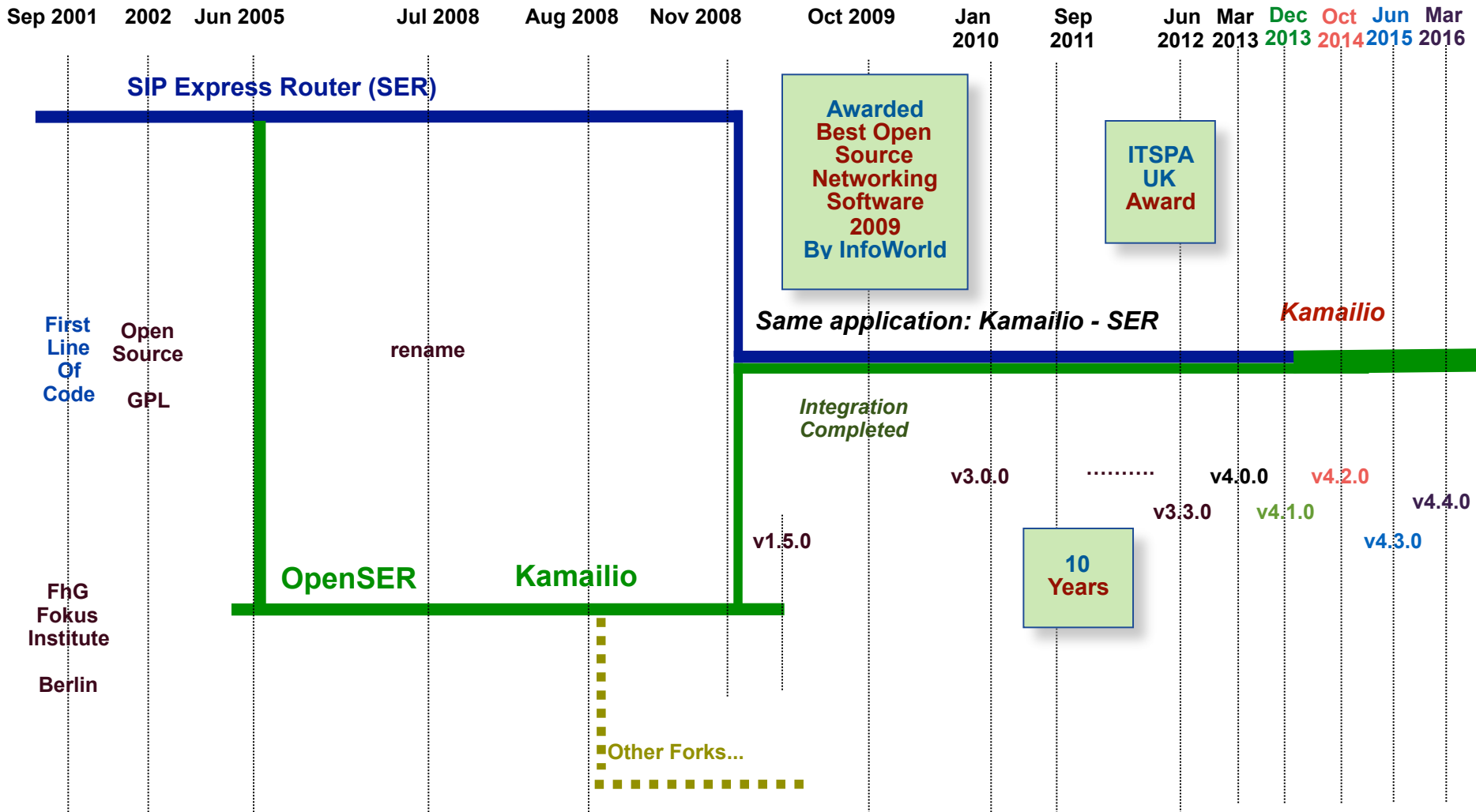


# KAMAILIO

project updates

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

# History

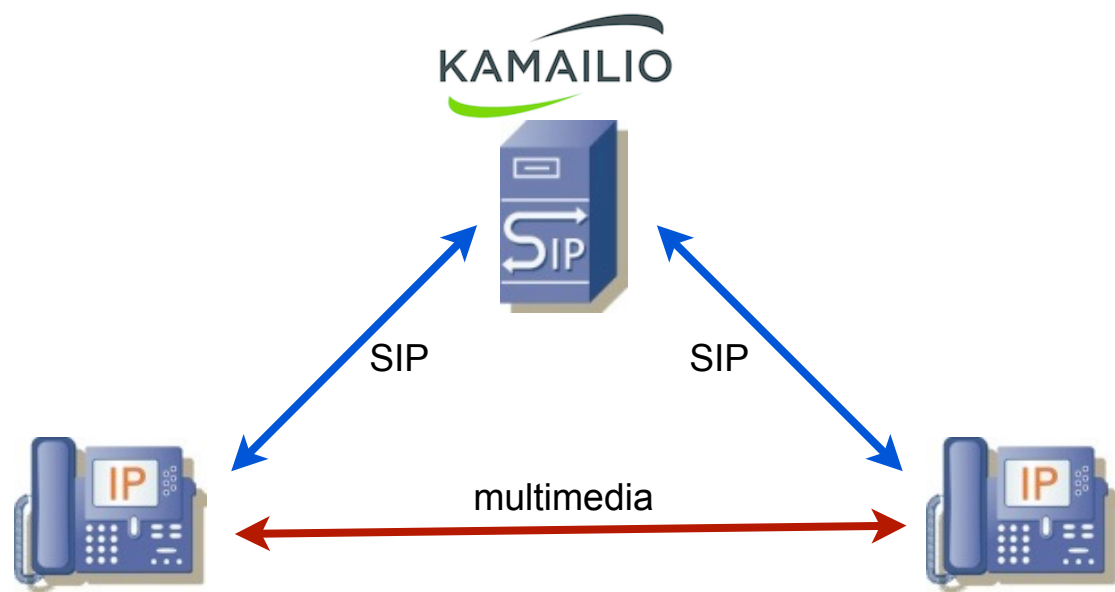


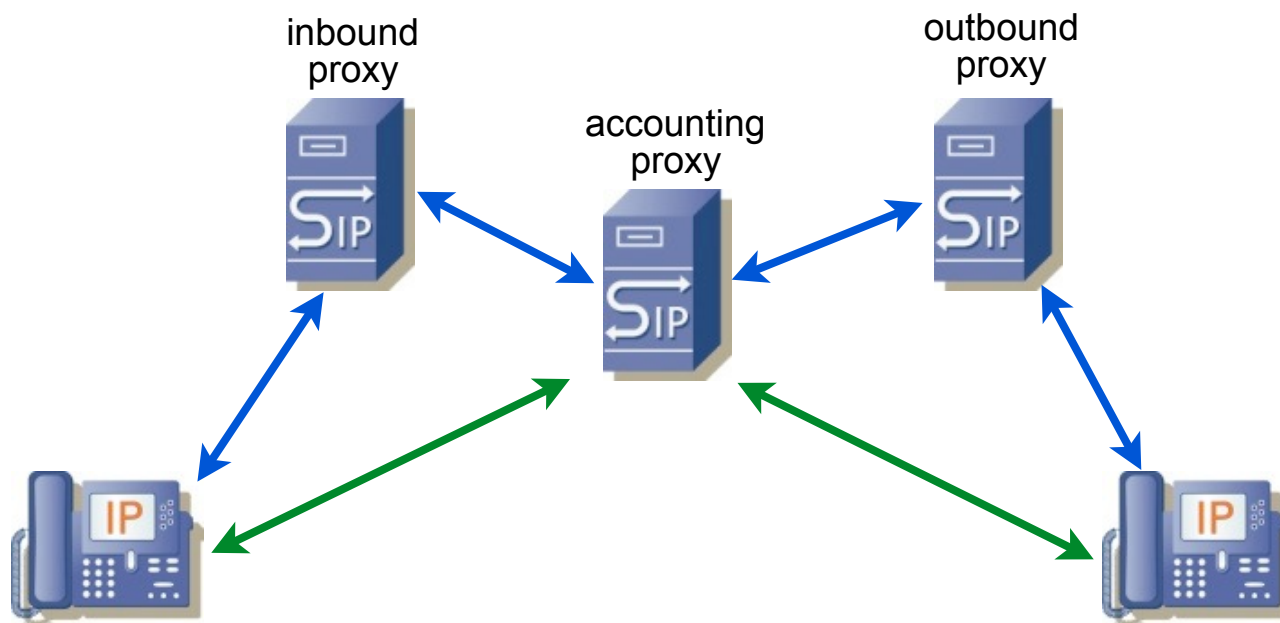


# Kamailio

## overview









---

## □ SIP Signaling Server

- SIP registrar
  - handle registrations from devices
- SIP location server
  - locate targets - lcr, load balancer, user location service
- SIP proxy server (router)
  - routing and re-routing huge volume of SIP traffic
- **SIP application server**
  - **rich telephony services**
  - **instant messaging and presence**
  - **integration with third party applications**
  - **integration with web 2.0 and social networking**



---

□ **It is not**

- SIP phone
- SIP media server
- SIP back-to-back user agent

□ **It handles only signaling**



# **Kamailio**

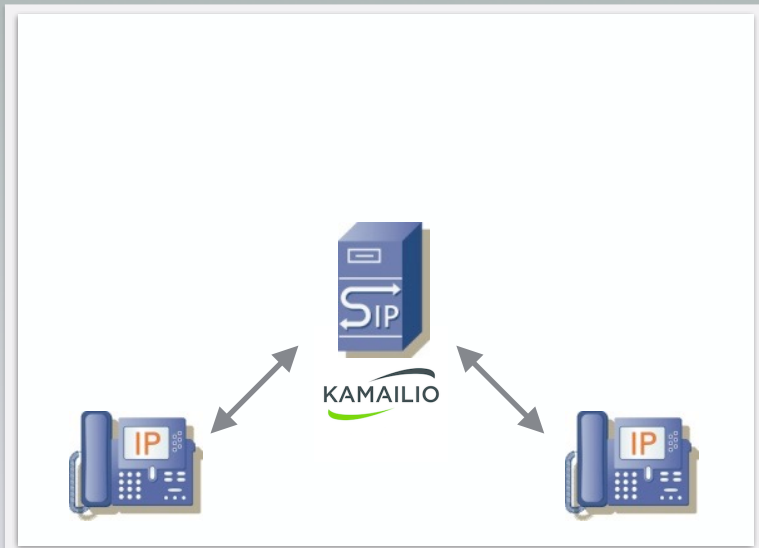
## **typical use cases**



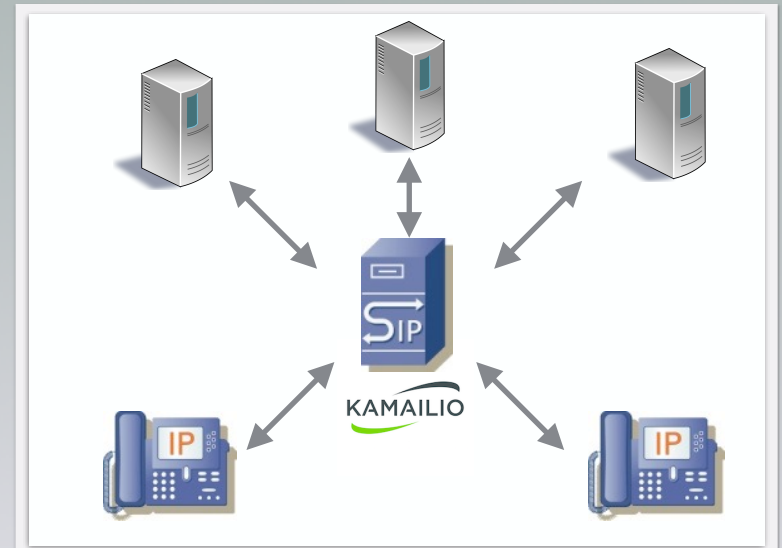


# TWO BASIC ARCHITECTURES

main signalling server



edge signalling server



# LOAD BALANCER

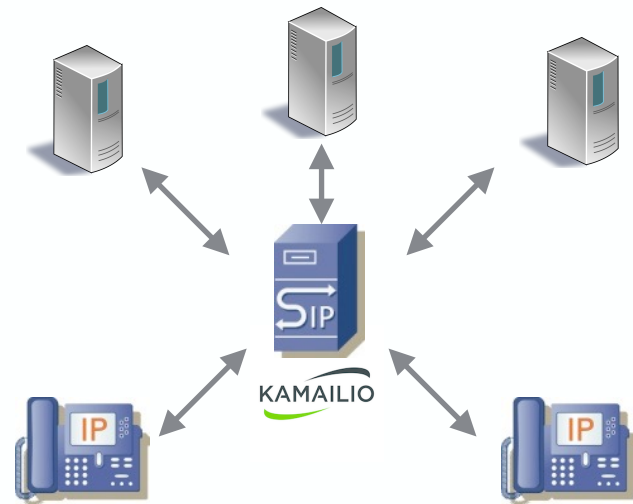
## *dispatcher module*

- 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

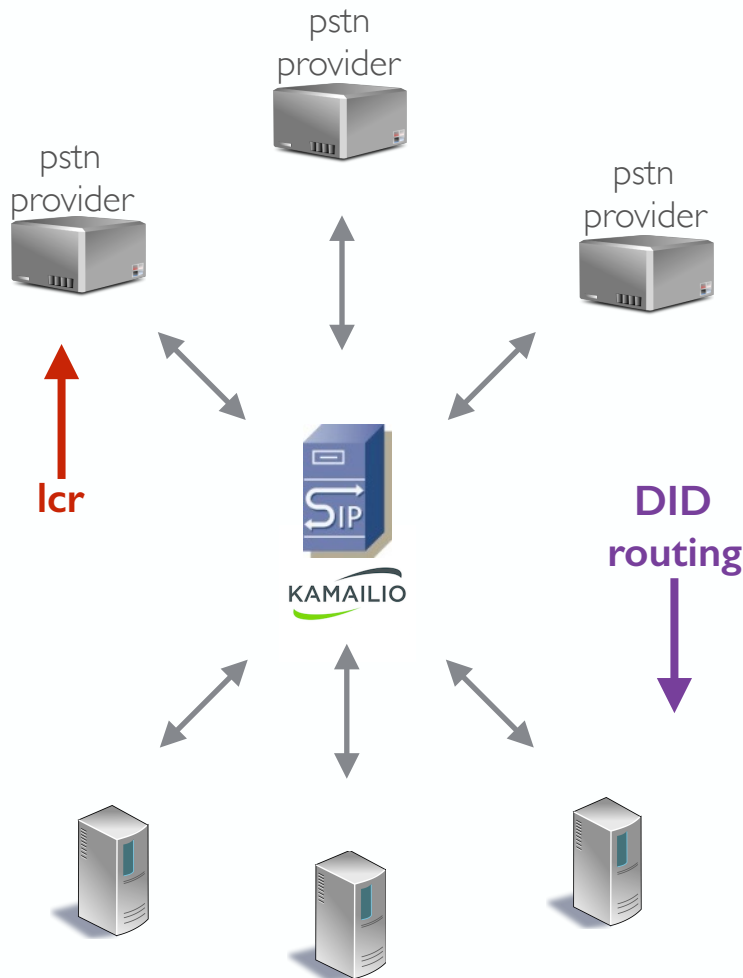
```
# Dispatch requests
route[DISPATCH] {
    # round robin dispatching on gateways group 'I'
    if(!ds_select_dst("I","4")) {
        send_reply("404", "No destination");
        exit;
    }
    xdbg("--- SCRIPT: going to <$ru> via <$du>\n");
    t_on_failure("RTF_DISPATCH");
    route(RELAY);
   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

## media servers - gateways - pbxes



# LEAST COST OR DID ROUTING



## *plenty of options*

- lcr module
- carrieroute module
- prefix\_route module
- drouting module
- combinations: mtree + dispatcher

```
route[LCR] {  
    if (!load_gws("I")) {  
        send_reply("503", "Error loading gateways");  
        exit;  
    }  
    if (!next_gw()) {  
        send_reply("503", "No available gateways");  
        exit;  
    }  
    t_on_failure("RTF_LCR");  
    route(RELAY);  
    exit;  
}  
# Re-route in case of failure  
failure_route[RTF_LCR] {  
    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 (next_gw()) {  
            t_on_failure("RTF_LCR");  
            route(RELAY);  
            exit;  
        }  
    }  
}
```

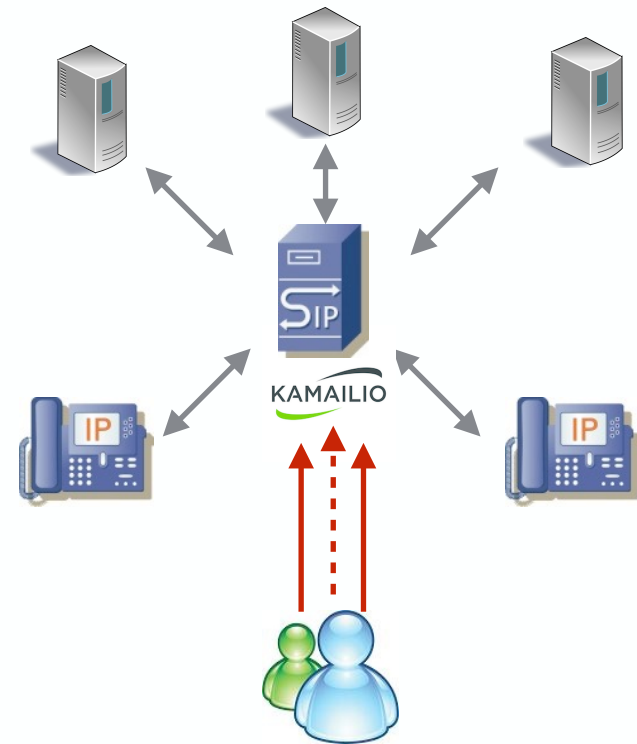
# SIP FIREWALL

## *block DOS or dictionary attacks*

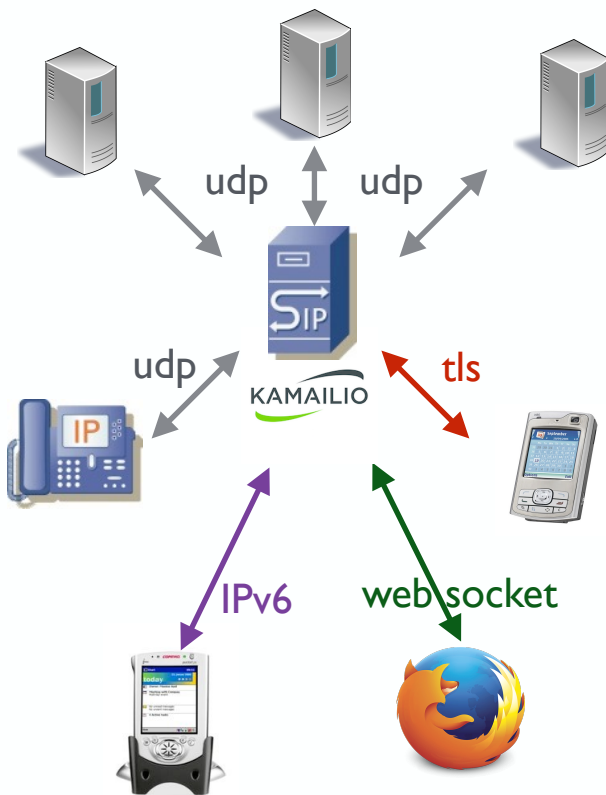
### *part of default kamailio.cfg*

- htable module
- pike module
- detect high volume of traffic from same source and block it for a specific interval of time

```
# ip ban htable with autoexpire after 5 minutes
modparam("htable", "htable", "ipban=>size=8;autoexpire=300;")
...
route[SAFEGUARD] {
    # flood detection from same IP and traffic ban for a specific interval
    # be sure you exclude checking trusted peers, such as pstn gateways
    # - local host excluded (e.g., loop to self)
    if(src_ip!=myself) {
        if($sht(ipban=>$si)!=null) {
            # ip is already blocked
            xdbg("request from blocked IP - $rm from $fu (IP:$si:$sp)\n");
            exit;
        }
        if (!pike_check_req()) {
            xlog("ALERT: pike blocking $rm from $fu (IP:$si:$sp)\n");
            $sht(ipban=>$si) = 1;
            exit;
        }
    }
    if($ua =~ "friendly-scanner") {
        sl_send_reply("200", "OK");
        exit;
    }
}
```



# TRANSPORT LAYER GATEWAY



## *core and tm module*

- set transport in R-URI or outbound proxy address
- force transport via dedicated function

## *bridging networks*

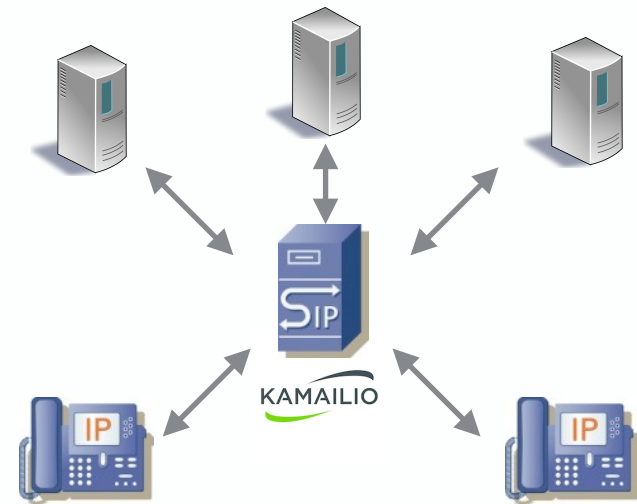
- public to private networks and back
- IPv4 to IPv6 and back
- any transport layer to another one and back

```
# Force UDP
route[TOUDP] {
    # round robin dispatching on gateways group 'l'
    record_route();
    $du = "sip:nexthop.com:5060;transport=udp";
    force_send_socket("udp:10.1.1.10:5080");
    route(RELAY);
    exit;
}
```

# OFFLOAD SIGNALLING PROCESSING

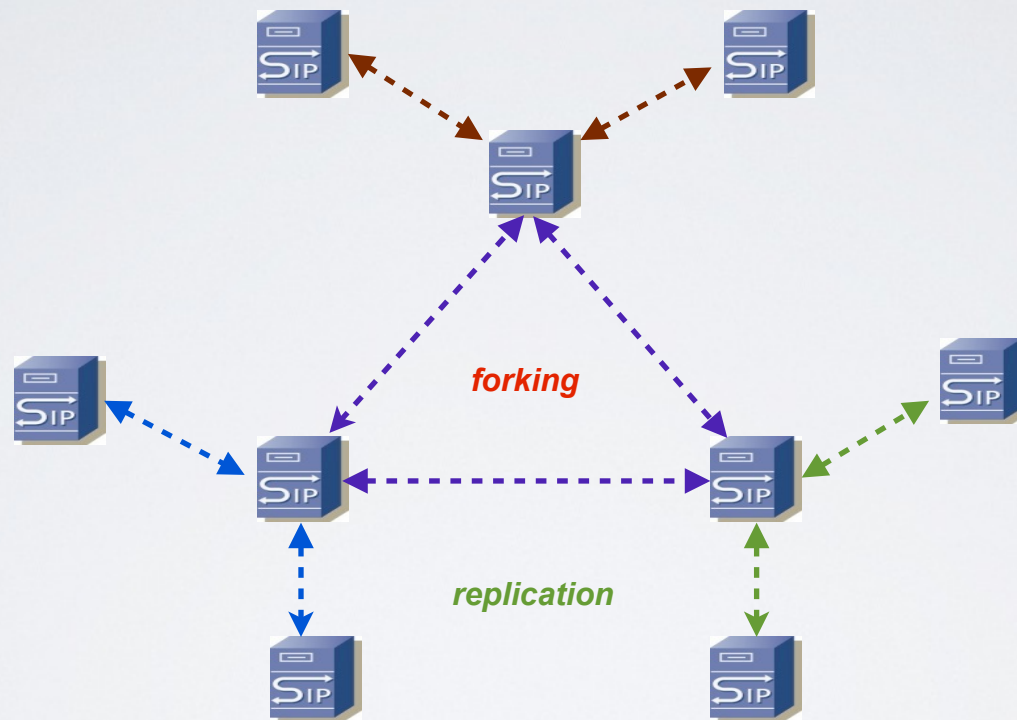
## *edge proxy*

- authentication
- nat traversal
- parallel forking
- serial forking
- call forwarding
- call blocking
- white/black listing
- DNS handling





# DESIGNED FOR SCALABILITY



# EXTRA FEATURES

## *plenty at signalling layer*

- instant messaging
- presence
- gaming
- notifications

## *additional functions*

- embedded XCAP server
- embedded MSRP relay
- IMS

## *extensibility*

- define your new request types
- flexibility in handling unknown requests, headers, etc.







# News

## Recent Years







---

□ security

- *strong encryption algorithms for tls*
- *flexibility to work with many certificates at the same time*
- *dnssec, config file crypto tools*

□ performance, scalability and redundancy

- *full asynchronous processing layer*
- *event driven api interface (json)*
- *nosql: redis, memcached, mongodb, cassandra*
- *distributed message queue between kamailio nodes*
- *new memory managers*

□ flexibility

- *embedded http client for API interaction*
- *dynamic memory manager selection*
- *update of routing parameters without restart*
- *more event routes*
- *json sip routing format*



---

□ enhancements for typical use cases

- *webrtc - secure websockets with support for fragmentation*
- *classic sip to webrtc gateway, including RTP to SRTP*
- *volte - ims: dozens of extensions*
- *sms routing handling*
- *more load balancing algorithms*
  - *call load distribution, relative weight, ...*
- *topology hiding and topology stripping*
- *user location partitioning and replication*
- *generic caching replication*
- *prepaid and call stateful control*
- *call rating limits*
- *detecting attacks and blocking them*
- *IPv6 cleanup*



# The Future

**Kamailio 5.0**



- 
- 
- ❑ native configuration with embedded interpreters
    - *lua, perl, python, .net, ...*
    - *ability to reload routing logic*
    - *highly optimized alternative of config with our own routing language*
  - ❑ continuous integration
    - *unit tests*
    - *automatic builds*
    - *source code restructuring*
    - *review options for a new build system*
  - ❑ api driven processing
    - *integration with external systems*
    - *keep kamailio as bare sip layer router*
  - ❑ your suggestions are welcome
    - *...*



# **The 4th Kmailio World Conference**

**May 18-20, 2016**

**Berlin, Germany**

**celebrating 15 years of Kmailio development**





**Celebrating 15 Years of Kamailio Development!**

May 18-20, 2016 - in Berlin, Germany

[www.kamailioworld.com](http://www.kamailioworld.com)



- ❑ Berlin, same nice location in the city center
  - *half day technical workshops*
  - *two days of conference*
- ❑ workshops
  - *IMS tutorial*
  - *continuous integration*
  - *troubleshooting*
  - *alternative: visit to Fraunhofer FOKUS testbeds (to be confirmed very soon)*
- ❑ conference
  - *SIP from classic telephony to webrtc and volte*
  - *Kamailio and other open source projects well represented*
    - ❑ Asterisk, FreeSwitch, Sems
  - *security and scalability*
  - *typical use cases*
  - *experiences from operating large telephony platforms*



**Celebrating 15 Years of Kamailio Development!**

May 18-20, 2016 - in Berlin, Germany

[www.kamailioworld.com](http://www.kamailioworld.com)



---

# Questions?



**Daniel-Constantin Mierla**  
Co-Founder Kamailio Project  
@miconda