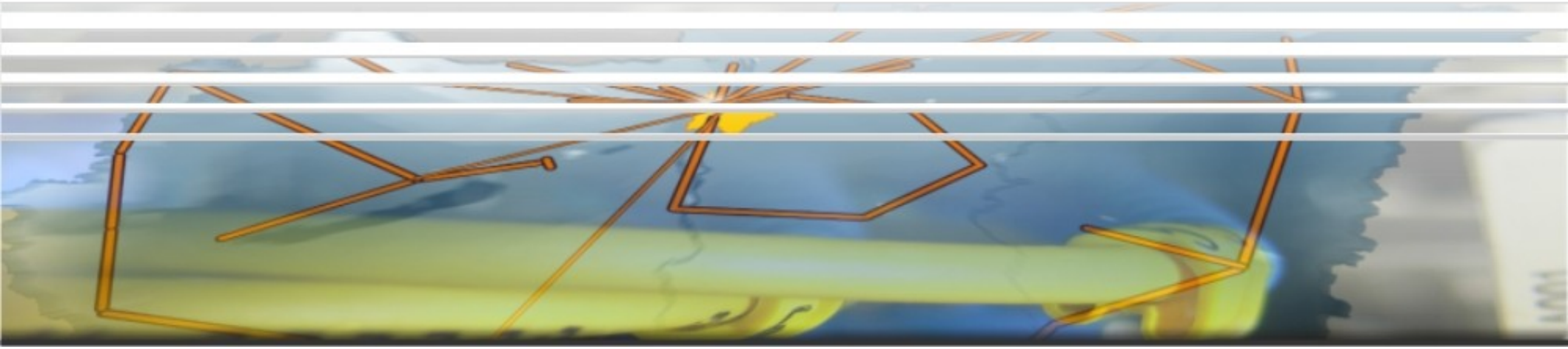
openSIPS



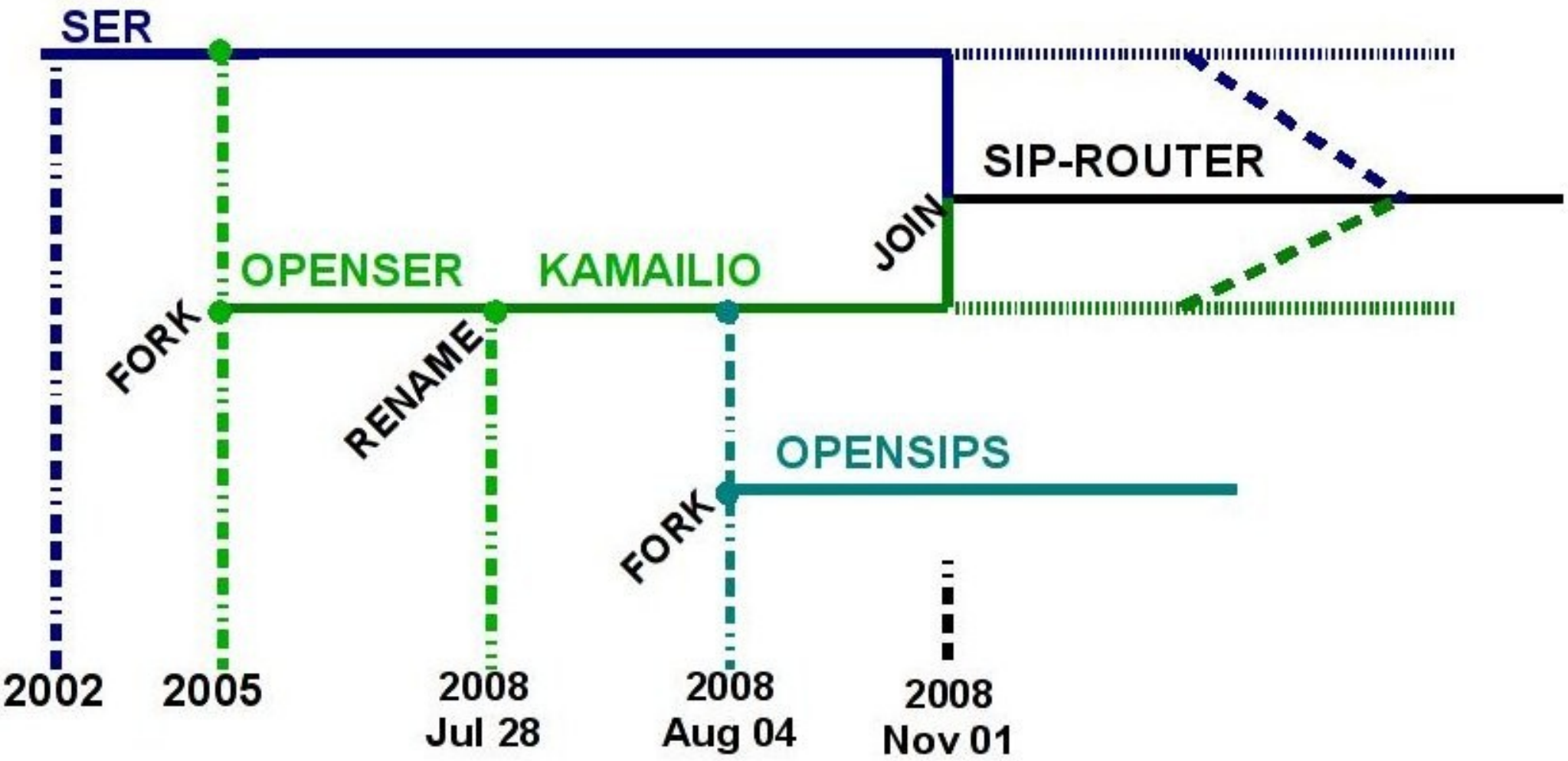
03/12/10
Budapest / Hungary

Mészáros Mihály

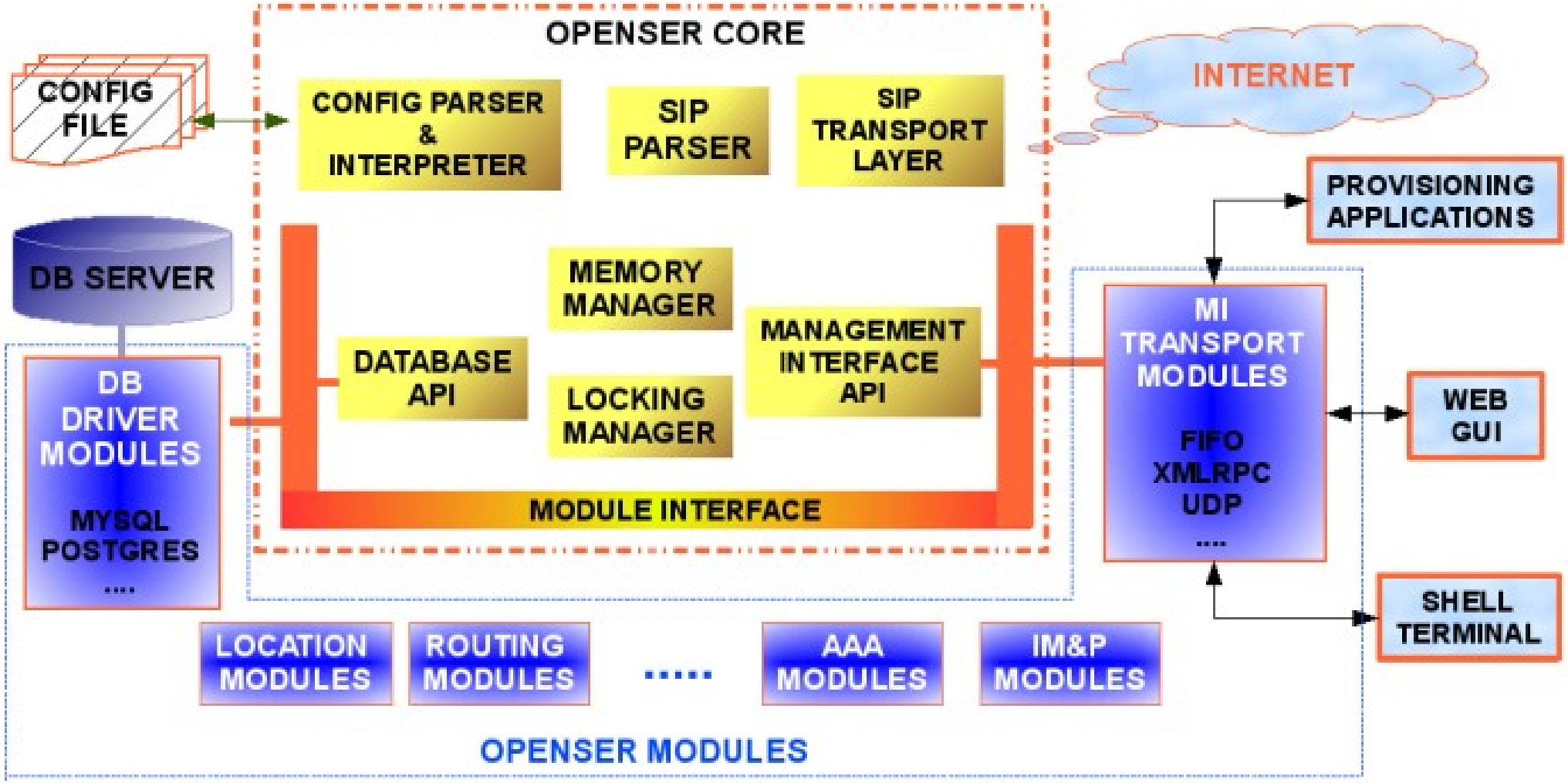
OpenSIPS most important features

- SIP PROXY/Registrar/Redirect server
- Stateless/statefull
- IPv4/IPv6
- UDP/TCP/TLS/SCTP
- DNS SRV/NAPTR
- ENUM
- PRESENCE
- MySQL
- RADIUS
- LDAP/H.350
- Multi Homed/Multi domain
- Simple/Robust
- Forking processes
- straightforward failover and redundancy
- Simple core + module architecture
- Variables
  - script variables
  - pseudo-variables (access to the SIP messages)
  - AVPs (values persistent per SIP transactions)
- Routing blocks

History


SER
SIP-ROUTER
JOIN
OPENSER
KAMAILIO
FORK
RENAME
OPENSIPS
FORK
2002
2005
2008
Jul 28
2008
Aug 04
2008
Nov 01

OpenSIPS architecture



Configuration file

- Static text file
- the file is loaded and processed only at startup
- flex, bison

  flex and bison are used to parse the configuration file and build the actions tree that are executed at run time for each SIP message.
- Currently no reloading
  - Some "config" can be dynamically changed using DB backend.
- Most important global options
  - Worker numbers (fork)
  - Protocols(UDP/TCP/TLS/SCTP)

- Config file Structure
  - Global options
  - Modul loading

    `loadmodule "modulname"`
  - Modul parameters

    `modparam("modul","praram eter",value)`
  - Route block

    `route[1]{....}`

Most important route blocks

- Route / request_route

  Request routing block. It contains a set of actions to be taken for SIP requests.

- branch_route

  Request's branch routing block. It contains a set of actions to be taken for each branch of a SIP request.

- failure_route

  Failed transaction routing block. It contains a set of actions to be taken each transaction that received only negative replies (>=300) for all branches.

- on_reply_route

  Reply routing block. It contains a set of actions to be taken for SIP replies.

Other route blocks

- error_route

  The error route is executed automatically when a parsing error occurred during SIP request processing. This allow the administrator to decide what to do in case of error.

- local_route

  The local route is executed automatically when a new SIP request is generated by TM, internally (no UAC side).

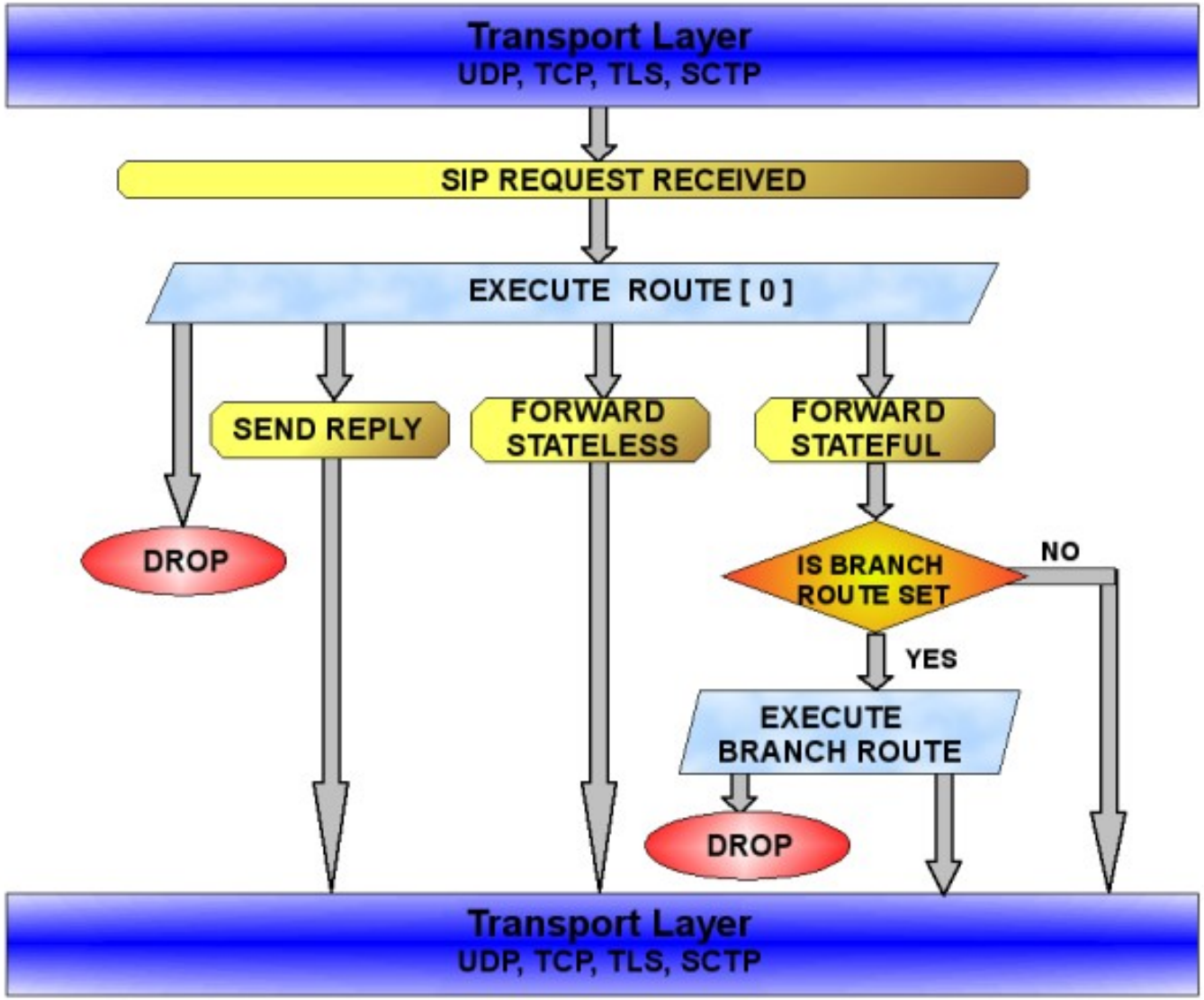
- startup_route

  The startup_route is executed only once when OpenSIPS is started and before the processing of SIP messages begins. This is useful if some initiation actions are needed, like loading some data in the cache, to ease up the future processing.

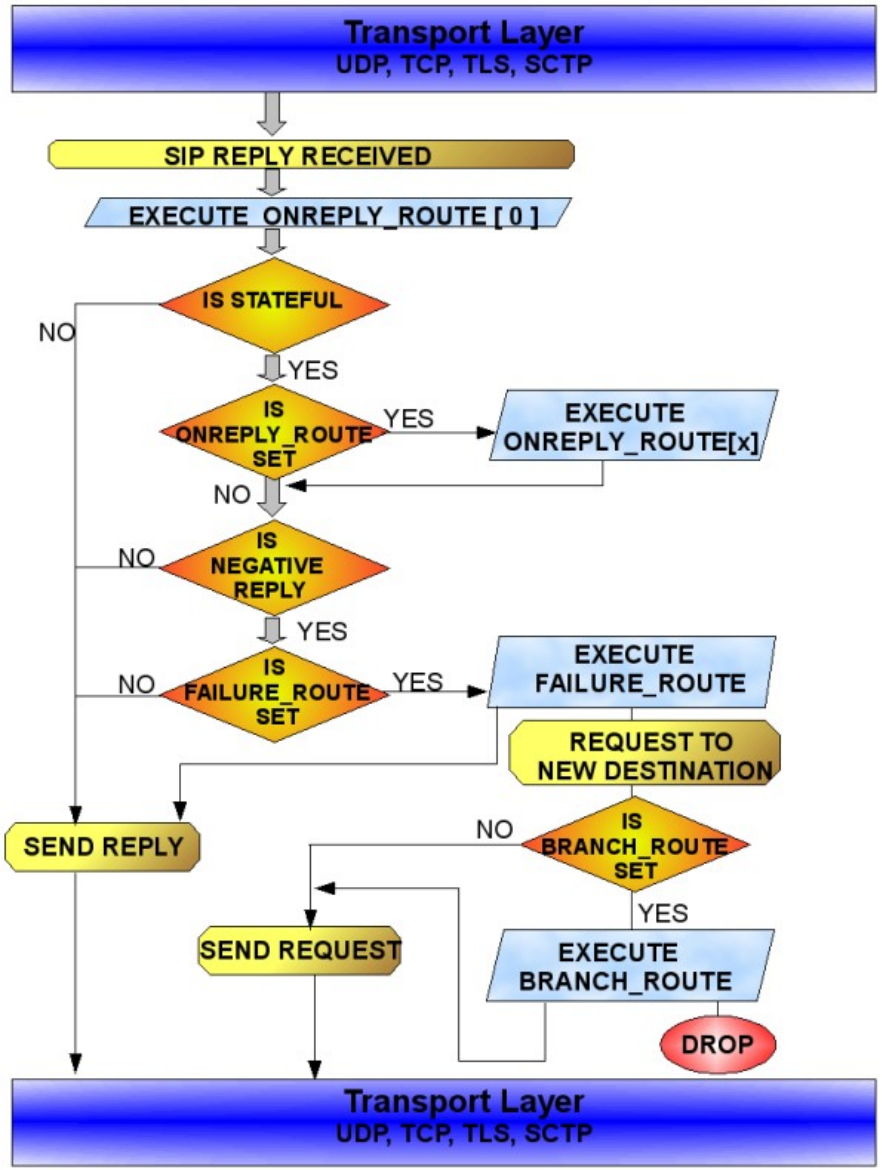
- timer_route

  The timer_route is as the name suggests, a route executed periodically at a configured interval of time specified next to the name(in seconds).

New request


Transport Layer
UDP, TCP, TLS, SCTP
SIP REQUEST RECEIVED
EXECUTE ROUTE [0]
SEND REPLY
FORWARD STATELESS
FORWARD STATEFUL
DROP
IS BRANCH ROUTE SET
NO
YES
EXECUTE BRANCH ROUTE
DROP
Transport Layer
UDP, TCP, TLS, SCTP

New response



Variables

- Variables
  - script variables
    - visible only in the routing blocks
    - process related
    - integer / string value
    - read/write
    - single value
  - AVP - Attribute Value Pair
    - Message or transaction related
    - single or multiple value
  - pseudo variables
    - Related to currently processed sip message
    - Mostly read only
    - single or multiple value

- All variables starts $
- Example
  - Script variable
    `$var(name)`
  - AVP
    `$avp(id) / $(avp(id)[N])`

    `“id” can be:`
    `i:number,s:string,alias`
  - Pseudo variables
    `$name`
- Multiple value

```
$avp(i:17) = "one";
# we have a sigle value
$avp(i:17) = "two";
# we have two values ("two","one")
$avp(i:17) = "three";
# we have three values
("three","two","one")
```

Most important core function and parameters

- core keywords
  - uri,method,msg:len
- core values
  - INET/INET6
  - TCP/UDP
  - myself
  - maxlen
- parameters
  - alias
  - children
  - debug
  - fork
  - listen

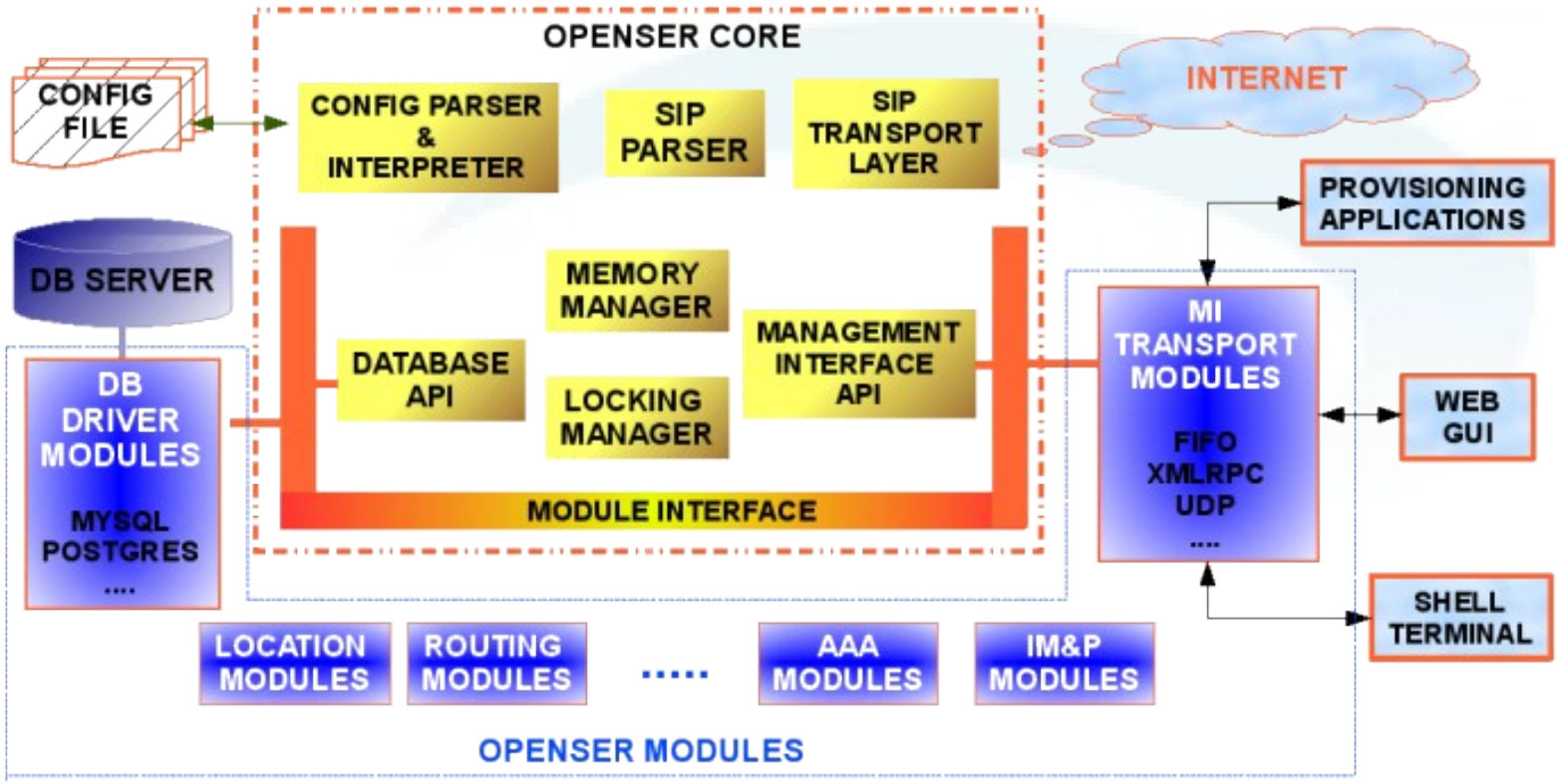
- functions
  - drop()
  - exit()
  - return()
  - rewriteuri()
  - append_branch()
  - serialize_branches()
  - prefix()
  - strip()
  - strip_tail()

SIP Parser, Modules, MI

- SIP Paraser
  - own implementation of SIP parser
  - incremental parser
- Modul Exporting
  - Parameters
  - functions
    - Route-block functions
    - Management Interface functions
- Management Interface
  - uptime
  - kill
  - ps

Database API, Backends

- Flat File
- TEXT
- Berkeley
- HTTP
- UNIXODBC
- Native
  - MySQL
  - PosgreSQL
- VIRTUAL
  - combination of above



Authentication

- Authentication
  - TLS client cert
    - Client cert authentication
  - digest (md5 hash)
    - RFC3261 stated basic is not allowed
    - HA1=MD5(A1)=MD5(username:relam:password)
    - HA2=MD5(A2)=MD5(method:digestURI)
    - response=(HA1:nonce:HA2)
- The most used Backends:
  (But any database backend can be used to authenticate requests)
  - MySQL
    - Plain text, HA1
  - Radius
  - LDAP
  - H.350

The most important modules

- sl
  - Stateless forwarding
- tm
  - transaction stateful forwarding
- rr
  - record-route header insert and loose_routing
- userloc
  - user location table management
- Registrar
  - save or lookup AoR in location table
- Maxfwd
  - Max-Forwards Header field check or add
- auth
  - static
  - variable based authentication
- auth_db
  - auth from DB
- db_mysql
  - MySQL as DB backend
- enum
  - ENUM lookup
- ldap
  - LDAP search
- h350
  - H.350 LDAP lookup based on ldap
- Pike
  - DoS protection based on IP address

Modul sl

- Stateless forwarding

  The SL module allows OpenSIPS to act as a stateless UA server and generate replies to SIP requests without keeping state. That is beneficial in many scenarios, in which you wish not to burden server's memory and scale well.

- function
  - sl_send_reply(code, reason)

    For the current request, a reply is sent back having the given code and text reason. The reply is sent stateless, totally independent of the Transaction module and with no retransmission for the INVITE's replies.

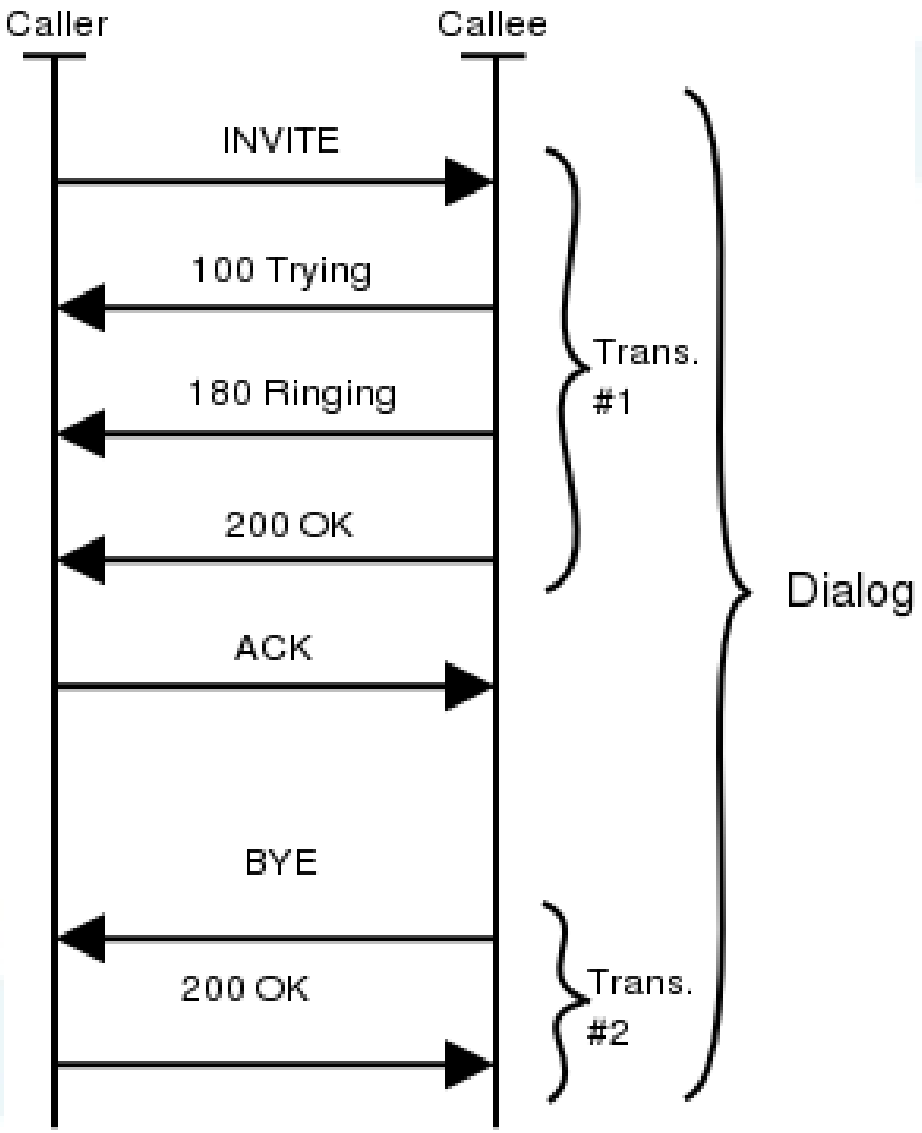
  - sl_reply_error()

    Sends back an error reply describing the nature of the last internal error.

Modul tm (transation stateful)

- Stateful transaction processing
  - Transaction accounting
  - Forking must implemented statefully
  - Retransmission
  - Timer based Failover
    - fr_timer (final reply)
    - fr_inv_timer (final reply invite)
  - DNS based Failover
    - RFC3263 (NAPTR, SRV)

Dialog, Loose route, Record-Route



Modul rr (record-route)

- Parameters
  - enable_full_lr
    ;lr=on instead of just ;lr
  - enable_double_rr
    - cross protocol forwarding
    - or forwarding between two disconnected network
- Functions
  - record_route

    Adds a new Record-Route header field
  - loose_route

    The function performs routing of SIP requests which contain a route set. The name is a little bit confusing, as this function also routes requests which are in the “strict router” format.

Modul registrar / usrloc

- Usrloc is the backend what is used by registrar no exported functions
- Usrloc Memory and database sync
  - db_mode
    - The usrloc module can utilize database for persistent contact storage. If you use database, your contacts will survive machine restarts or SW crashes. The disadvantage is that accessing database can be very time consuming.
- Regsitrar can be used as Single or Multi domain
- Registrar Functions
  - Save

    Add / Remove / Modify (depending on experies and contact header fields) binding between AoR and contact(s). Save the binding(s) to backend database.
  - Lookup

    This function is tring to find all contacts using Request-URI from userloc. Request-URI is rewritten to contact URI with the highest “q” value.
    - Flag 'b' (no Branches) - this flag controls how lookup function processes multiple contacts. If there are multiple contacts for the given username in usrloc and this flag is not set, Request-URI will be overwritten with the highest-q rated contact and the rest will be appended to sip_msg structure and can be later used by tm for forking. If the flag is set, only Request-URI will be overwritten with the highest-q rated contact and the rest will be left unprocessed.

Module acc (Accounting)

- Account transactions
  - The fixed minimal accounting information is:
    - Request Method name
    - From header TAG parameter
    - To header TAG parameter
    - Call-Id
    - 3-digit Status code from final reply
    - Reason phrase from final reply
    - Time stamp when transaction was completed
  - Multiple legs due forwarding
- Based on flag or function, for example
  - Parameter
    - modparam("acc", "db_flag", 2)
    - Can triggered with use setflag(2)
  - Function
    - acc_db_request(comment, table)

Modul siptrace

- Offer a possibility to store incoming/outgoing SIP messages in database.
- by calling explicitely the sip_trace() method in OpenSIPS configuration file. In this case the original message is processed.
- By setting the flag equal with the value 'trace_flag'parameter of the module. In this case, the message sent forward is processed. The logging mechanism is based on TM/SL callbacks, so only messages processed with TM/SL are logged.
- Tracing dialog using function trace_dialog()
- Database schema
  - Timestamp, callid, traced_user, msg, method, status, From ip, toip, fromtag, direction

Debug, Log

- Syslog
  - Facility
  - Log levels
    - L_ALERT (-3) - this level should be used if the error requires immediate action.
    - L_CRIT (-2) - this level should be used if the error is a critical situation.
    - L_ERR (-1) - this level should be used to report errors during data processing which do not cause system malfunctioning.
    - L_WARN (1) - this level should be used to write warning messages.
    - L_NOTICE (2) - this level should be used to report unusual situations.
    - L_INFO (3) - this level should be used to write informational messages.
    - L_DBG (4) - this level should be used to write messages for debugging.
  - Log()
    - Core function
    - logging
  - xddbg()
    - Modul xlog
    - Formated log

Database schema MySQL

- version
- Location
- aliases
- registrar
- subscriber
- dbaliases
- acc
- sip_trace
- speed_dial
- presentity
- watchers
- active_watchers

Config example step by step

References

- http://www.opensips.org
  - http://www.opensips.org/Training/Webinars
  - http://www.opensips.org/Resources/DocsCookbooks
  - http://www.opensips.org/Resources/Features
- http://www.sip-router.org
- http://www.asipto.com/pub/openser-devel-guide/
- http://ftp.iptel.org/pub/ser/0.8.14/doc/html/sip_introduction.html
- http://wiki.voip.niif.hu/index.php/Link
- http://www.iptel.org/files/sip_tutorial.pdf
- http://www.kamailio.org/
- http://sip-router.org/
- http://www.voice-system.ro/docs/ser-syslog/index.html

Thank You!