
OPENSER

The Open Source SIP Server



Daniel-Constantin Mierla

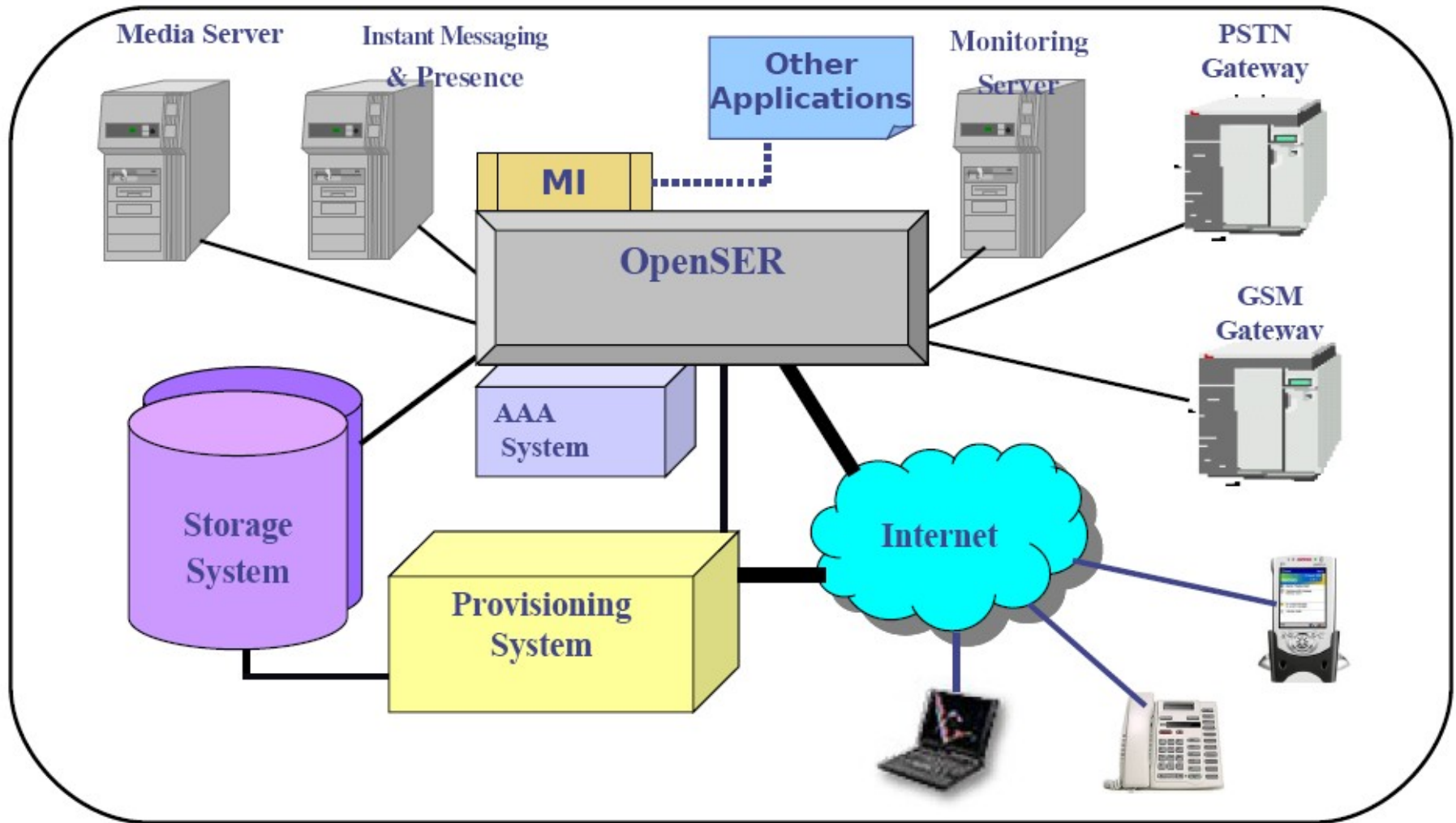
Voice System SRL

<http://www.voice-system.ro>

<http://www.openser.org>

Overview

openser deployment prototype



- *it is a signaling VoIP server*
- base protocol is SIP: Session Initiation Protocol (RFC3261)
- aka:
 - SIP registrar
 - SIP proxy
 - SIP router
 - SIP redirect server
 - SIP application server
- it is not
 - media server
 - back-to-back user agent
 - SIP phone

- handle VoIP phone registrations
- route calls between VoIP phones
- route calls to third-party devices (gateways, media servers)
- redirect calls
- load balance and dispatching
- trunking

- opener alone can't ...
 - provide voice mail
 - provide announcement services
 - provide audio conferences
 - provide two stage dialing
 - end calls in the middle
 - gateway to PSTN, h323, ...

- written in C
- additional tools in Shell, Ruby, Java, Perl
- developed Linux
- support for most Unix-like platforms
- no support for Windows
- collaboration environment
- guided by standards, mainly from IETF and ITU
- integration of real-world deployments feedback
- decentralized management
- open source
- GNU Public License

- modular architecture
 - small footprint core (pretty much useless if used alone)
 - features plugged by modules (over 70)
- scripting language for configuration
 - combination of C and Shell styles
 - easy to understood when having programming background
- very strict in following standards
 - loose when receiving
 - strict when sending
- high level application interfaces
 - Java SIP servlets
 - PERL API
 - text-plain and XMLRPC

- difficult to adapt when coming from PSTN/PBX style
 - extension based routing
 - line by line prefix matching shall be translated in logical blocks
- hard to auto-provision the configuration file
 - text file
 - conditional blocks
 - logical expressions

- SIP proxy, redirect and registrar server
- User location service
- Digest authentication
- Customizable routing policy
- Plug-in module interface
- IPv4 and IPv6 support
- UDP, TCP and TLS
- ENUM
- Call Processing Language (CPL)
- AAA with database, radius and diameter
- Least Cost Routing
- Load balancing and traffic dispatching

- Presence support
 - end-to-end model
 - client-server module
 - XCAP support
 - BLA/SLA
 - user location presence
- Management interface to link external applications
 - fifo file
 - xmlrpc
 - unix sockets
- NAT traversal
- instant messaging
 - offline storage
 - im conferencing

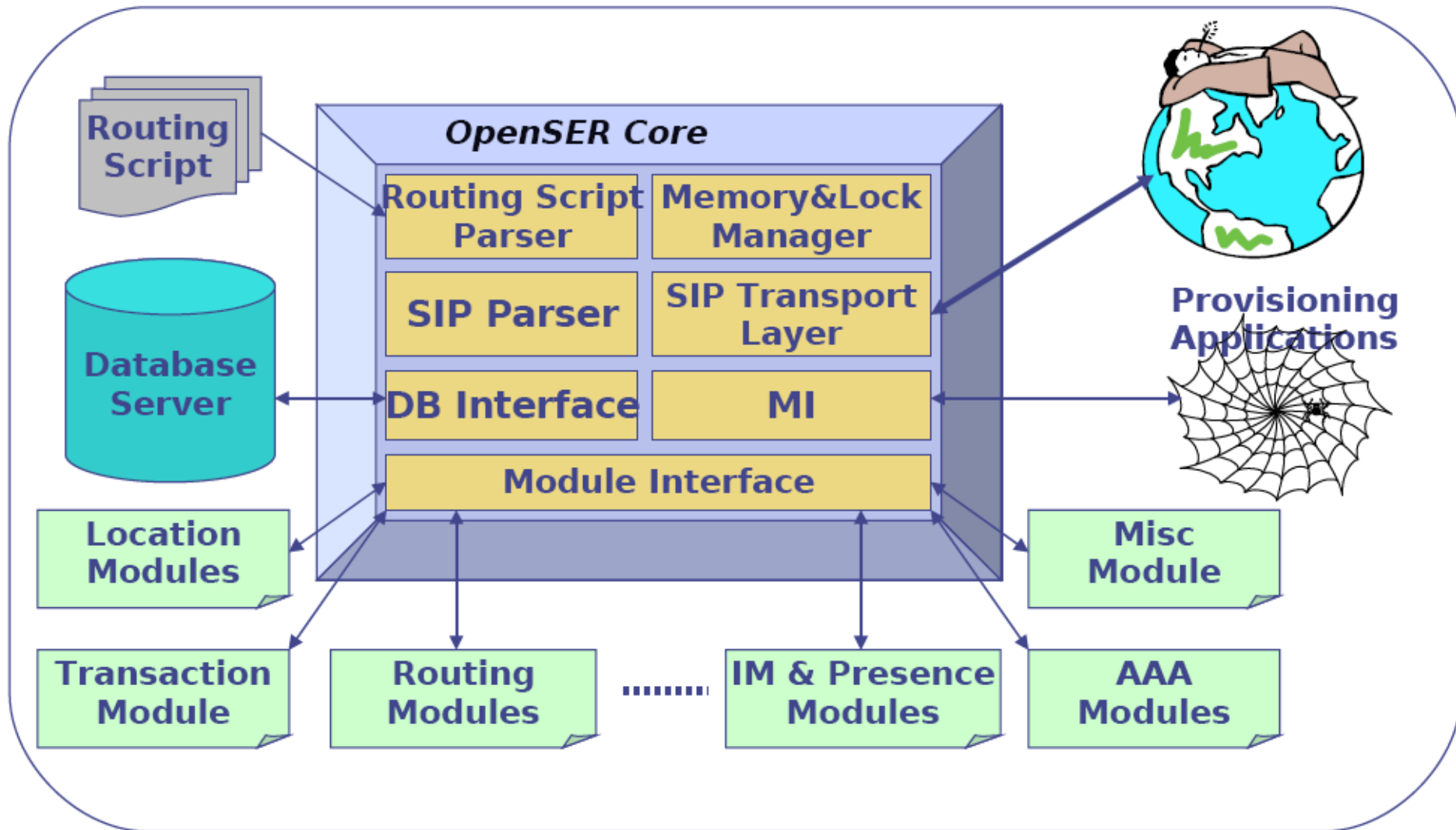
- Database interface
 - MySQL
 - PostgreSQL
 - unixodbc (Oracle, MSSQL, DB2, ...)
- gateway to SMS
- gateway to XMPP/Jabber
- OSP (open Settlement Protocol)
- Perl programming interface
- Java SIP Servlet programming interface
- PATH support
- DNS with/black lists
- SRV and NAPTR support
- SNMP interface

- OpenSER as public project
 - started June 2005
- large number of developers :
 - 3 core developers
 - 21 main developers
 - ~30 developers
 - ~150 contributors
- worldwide community of users
- OpenSER Summit at VoN Berlin, November 2006
- developer's meeting in Paris, June 2007
- plans for Summit 2007 to VoN Boston

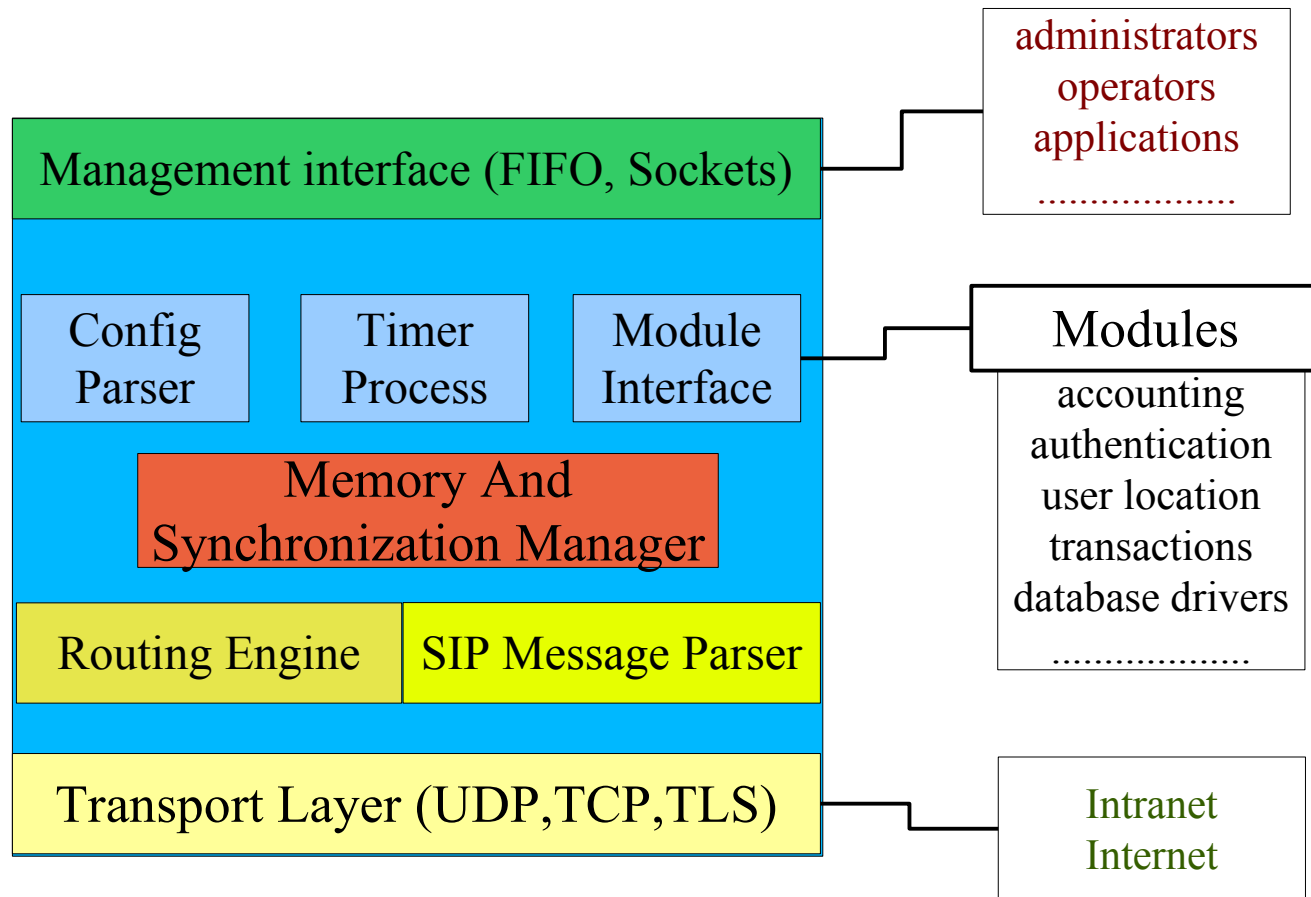
- from packages
 - Debian, Ubuntu, ...
 - Fedora
 - OpenSUSE
 - FreeBSD
 - Embedded systems
- from tarball
 - <http://www.openser.org/mos/view/Download/>
- from source repository
 - SVN
 - sourceforge.net: <http://sourceforge.net/projects/openser/>
 - <https://openser.svn.sourceforge.net/svnroot/openser>

Design

- multi-processing
- daemon type
- two main logical components
 - core
 - transport layer
 - memory system
 - synchronization mechanisms
 - SIP parser
 - management interface
 - database interface
 - configuration file and script variables
 - modules
 - features



openser interaction with the world



- the config values are pre-compiled to gain some speed
 - drawback in complex reloading or re-using the exported functions from other modules, or on the same fashion as you get it in the configuration file
- process context
 - there is no per-transaction or per call contexts
 - cannot go for asynchronous processing
- global initialization
 - basic initialization
 - sanity checks
 - command line parameters
 - configuration file
 - networking initialization

- lazy (incremental) parsing
 - stops when needed information is found
 - jumps over known parts, even if the content may be broken
 - few parts are parsed by default: first line, top most via
- references to original SIP message
 - the values are not duplicated
 - changes to SIP message are not reflected until sending downstream
- optimizations to integer values for faster parsing and further processing
- tight integrated in openser
- no SDP (body content) parser

- abstract method to access database
- formalization of common database operations
 - insert
 - upgrade
 - delete
 - select
- enable seamless switching between different database servers
 - mysql
 - postgres
 - oracle, mssql, db2, ... (via unixodbc)
 - text file database
- ability to use different database servers at the same time
- allow creating virtual data sources

- communication with external application via simpler protocol and mechanism
- abstract layer between openser and MI transports
 - **fifo file**
 - text-plain protocol
 - line oriented
 - **xmlrpc**
 - xml based content
 - remote communication
- the content is specific for each MI command
- the mechanism used by **openserctl** to communicate with openser
- enables reinitialization of some internal parameters or reload of database records
- visualization of openser internals

- fifo command syntax

*request = first_line argument**

first_line = ':' command_name ':' reply_fifo '\n'

argument = (arg_name '::' (arg_value)?) | (arg_value)

arg_name = not-quoted_string

arg_value = not-quoted_string | ''' string '''

not-quoted_string = string - {' ', '\n', '\r'}

- example of fifo command

get_statistics:/tmp/reply_fifo\n

dialog:\n

tm:\n

\n

- references to parts of SIP message
- reference to headers
- values from the environment
- AVPs
- per-script variables
- special pseudo-variables

- standalone process
- run periodic tasks
 - retransmissions
 - location cleanup
 - nat traversal
 - offline storage cleanup
- millisecond precision
 - default accuracy - 100ms
- ability to register callbacks dynamically
 - developer's job to do it properly
- not visible configuration file

- three parts
 - global parameters
 - module parameters
 - routing blocks
 - request routing
 - failure routing
 - branch routing
 - reply routing
- C/Shell-like language
- modularity via function-style blocks
- conditional statements
- regular expressions support
- script variables

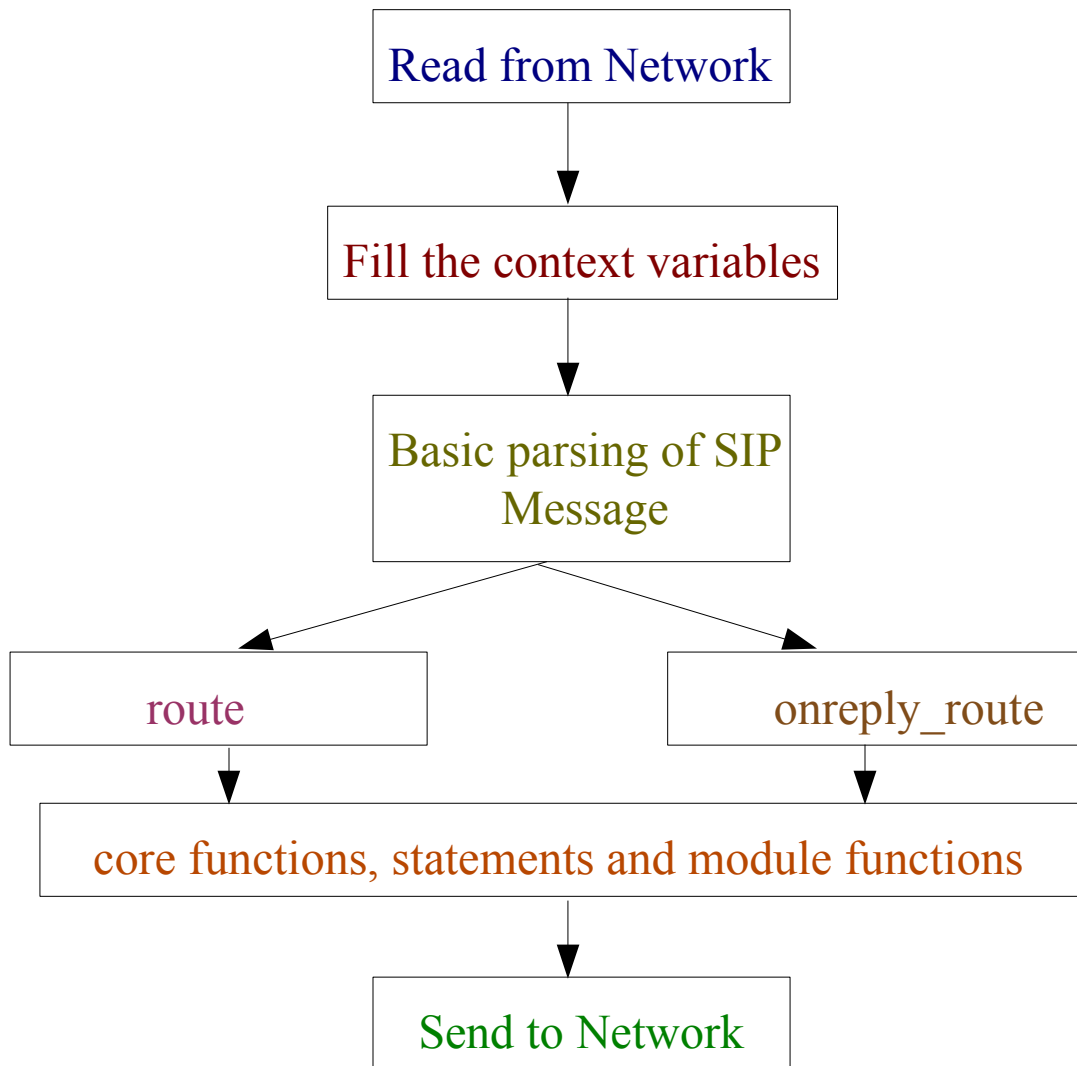
- the module is a dynamic library (shared object)
- identified by name
- exports a set of components to be used from other modules or configuration file
 - parameters
 - functions
 - statistics
 - mi commands
 - pseudo-variables
- special callback functions
 - global initialization
 - per worker initialization
 - destroy function
 - on-reply function

- module initialization
 - variables initialization
 - locking mechanisms and shared objects
 - load of persistent data
 - importing APIs
 - registering timer callbacks
- per-worker initialization
 - connections to database
 - initialization of per-processes variables
- destroy function
 - cleanup of internal structures
 - save of persistent data
 - destroy of locks and shared memory

- ability to use functionality implemented in other module from C code
- shall be careful with fixups
- modules exporting API
 - tm – sending stateless requests and replies
 - sl – sending stateless replies
 - usrloc – location management
- best to have a wrapper to functions exported to configuration file

- basically is **IP** routing
- **DNS** is the main routing mechanism
- **NAPTR** - Naming Authority Pointer Lookup
- **SRV** - Service Location Lookup
- **ANAME/CNAME** - Address Lookup
- **parallel** and **serial** forking
- black (**white**) lists

- R-URI
 - the main routing element
 - domain part and transport parameter
- Route headers
 - for inside dialog requests
- Replies are routed based on Via and received parameter
- support for outbound proxy
 - destination uri
- support for PATH headers for REGISTER
- customizable routing extensions
 - via script variables, headers, parameters
- least cost routing
- load balancing



- route
 - request handling
- failure_route
 - request handling upon negative reply receiving
- branch route
 - request handling on branch environment
- onreply_route
 - reply handling
- error_route

- IP authentication for known hosts
- check From user and authentication user to prevent spam
- don't allow not-authenticated anonymous calls, only authenticated requests from local users allowed
- don't allow direct access to critical resources (gateways, media servers)
- anti-flood detection via pike
- calls from foreign domain should be always considered untrusted unless via secure channel (TLS)

- ability to store records to
 - syslog
 - database
 - mysql
 - postgres
 - flat file
 - Oracle, MSSQL, ..., via unixodbc
 - radius
 - diameter
- event based accounting, reporting
 - start event
 - stop event
- full cdr by aggregation of accounting events

1.2.x and next

- major release on the 12th of March 2007 - 1.2.0
 - minor update on the 23rd of May 2007 – 1.2.1
- performances
 - user location – loading and runtime operations
 - timers – accuracy of retransmissions, smarter handling
 - DNS – black lists, SRV failover
- flexibility
 - configuration file variables
- presence
 - client-server SIMPLE extensions
 - user location and external interface
- im
 - gateway to XMPP and IM conferencing

- application lever
 - Java SIP Servlet interface and Application server – WeSIP
 - PERL API
- management interface
 - new abstract layer
 - FIFO and XMLRPC transports
 - SNMP
- sip session timers
- automatic error handling while parsing
- secure federations and peering
- infrastructure ENUM
- XCAP server
- Config Wizard

- presence
 - MWI
 - BLA/SLA
 - RLS
- SIMPLE-XMPP presence gateway
- security
 - TLS - GNU
 - Identity
- scalability
 - database interaction
 - virtual data sources
 - nat traversal
- datagram transport for MI

sip presence enabled whatsoever ...



sip presence enabled whatsoever ...



- FOSDEM 07
- VoN Spring, San Jose
- ROSDEV 07
- eLiberatica
- LinuxTag
- OpenSER Development Course
- VoN Europe

- ClueCon

- VoN Italy
- OpenSER Summit 2007
- VoN Autumn

development course, Paris 2007



- website: <http://www.openser.org>
- mailing lists
 - users: users@openser.org
 - developers: devel@openser.org
 - business: business@openser.org
 - team: team@openser.org
 - management board: board@openser.org
- web forum
 - <http://www.voipuser.org>
- wiki pages
 - <http://www.openser.org/dokuwiki/>
 - <http://www.voip-info.org>

Thank you!

daniel.mierla@voice-system.ro

Questions?