OpenSER – an introduction

Welcome!

Henning Westerholt

OpenSER project

24C3 Berlin, 29.12.2007

Outline

1. what is OpenSER
2. SIP what?
3. why people use this server
4. why its fun to use it
5. why its fun to work with the project
6. and its get even better!
7. how to use it by yourself

OpenSER – open SIP express router

component of VoIP infrastructures

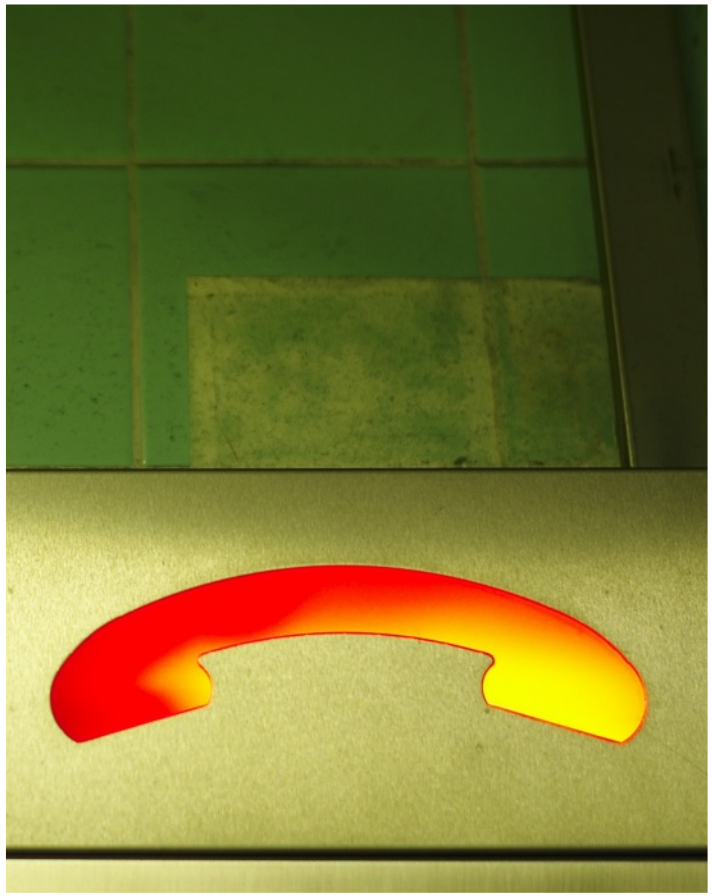
provides core services

- *proxy*
- *registrar*
- *balancer or router*
- *applikation server*

no PBX, more like a router

cares only about signaling, no media data

foundation for custom high-performance SIP services



some SIP and VoIP basics

SIP is a text based protocol, similar to HTTP

SIP does the signalising, RTP carries the media

Basic call setup is easy

- REGISTER, INVITE
- ACK, OK, CANCEL
- AUTH and ERROR messages

complex standard with many extensions

and every vendor implements its sightly different

some even complete wrong, and different per patch-level

this causes funny incompatibilities..

connecting people

telephony solutions for carriers and service providers

Usage at 1&1

- *800 million minutes per month*
- *1.6 million customers on the platform*
- *interfacing with asterisk, callweaver, legacy code*

telephony solutions for SMEs, nicely packaged in an appliance

telephony solutions for you!

example setup

Basic VoIP infrastructure for telephony services

- OpenSER with MySQL database
- one (Debian) Linux server

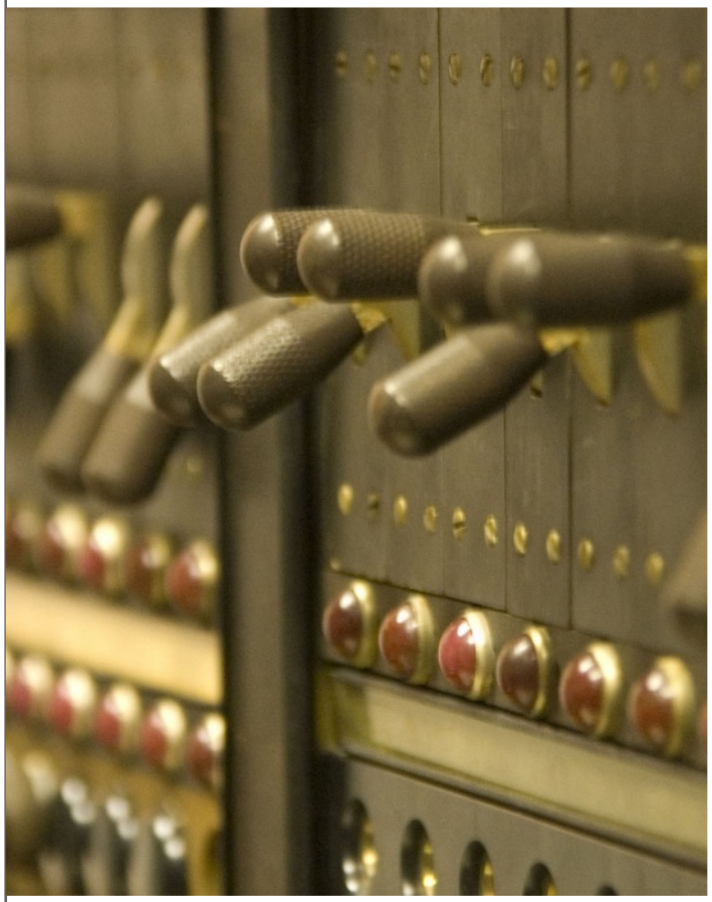
advanced services

- OpenSER, Asterisk and Callweaver
- dedicated (MySQL) database
- one server for each service (balancing, proxy, registrar)
- Mediaproxy or nathelper to get through the NAT
- PSTN gateways

high-availability infrastructure

- distribute services and databases to more server
- setup failover solution
- service and quality monitoring
- provide redundundant infrastrucures for all the dependencies..

Configuration and extensions



- configuration is done with a special C-like script language
- for every message this script is executed
- core and module functions are called, to modify the message flow and content
- routing descisions can be derived from
  - header fields or tag values
  - database content or previous messages
  - external (Perl) scripts
- easy extendability with modules

Modules

Modules are shared libraries

need to implement a certain interface

use services from the core

- parse SIP messages
- allocate memory from the pool
- access variables from the config script

databases access

- via a generic interface
- drivers are interchangable
- LDAP is also supported

can also access functions from other modules

- transactional or stateless sending
- user location, accounting

interfaces to other networks

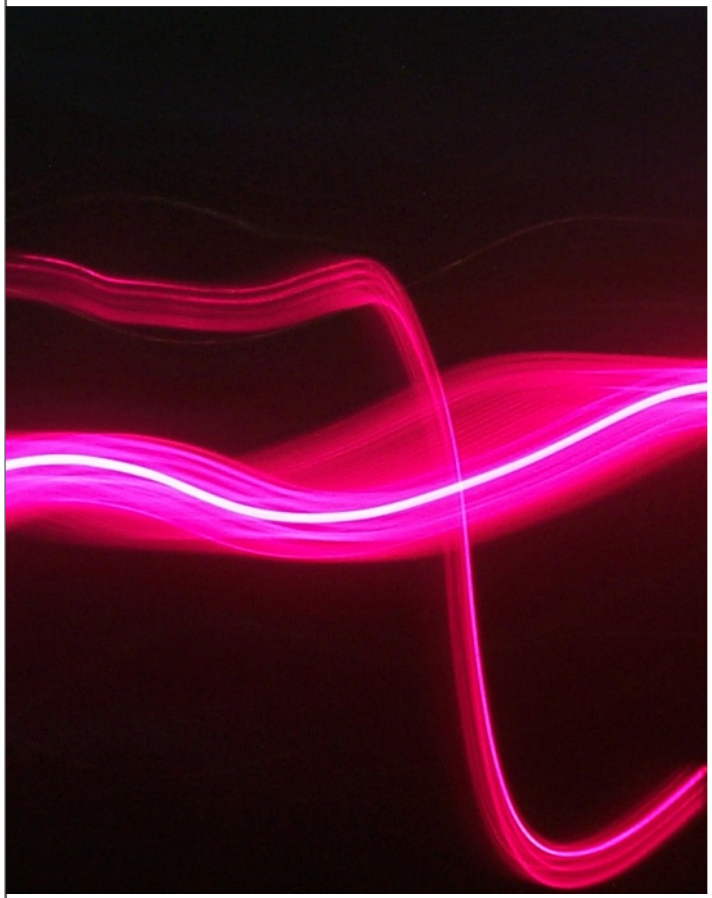
translate between different PSTN-gateways and telephony networks

instant messaging via XMPP (jabber)

SIMPLE presence support to get user status

straightforward interfacing with gateways for PSTN and mobile connections

Scalability



usable from small DSL routers to big carrier installations

carrier grade solutions

multiple server setup with load-balancing and failover

appliances

dedicated boxes for office connectivity

Usage in embedded systems

DSL router (Linksys NSLU and other) with IP-phones for small SoHo setup

Performance

routing engine written in C

macros, inline functions for critical code

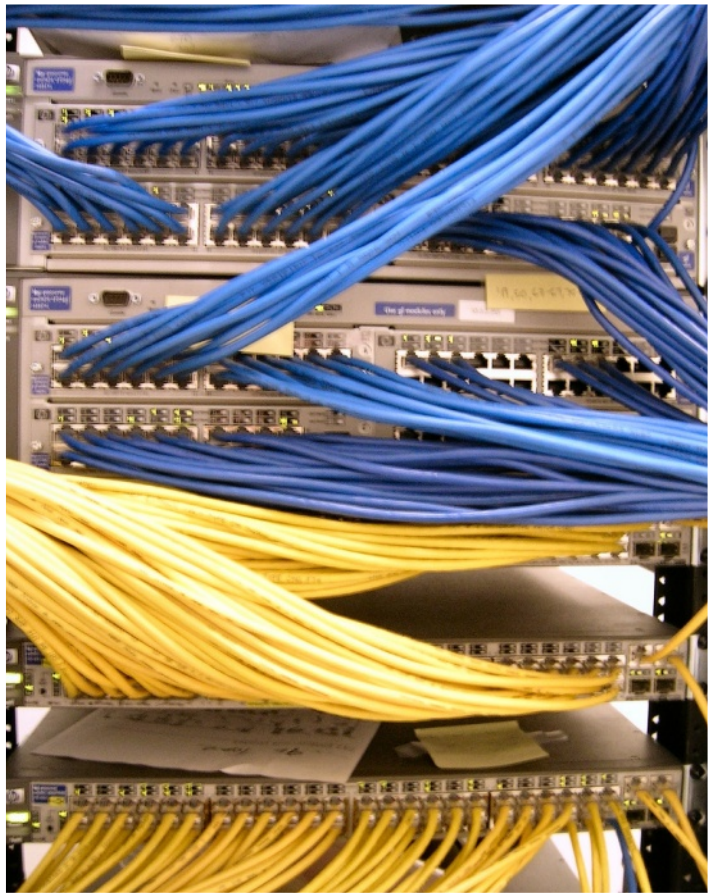
config script is "fixed" on startup

- *string variables are replaced with integers*
- *regular expressions are compiled*
- *variables are checked*

custom memory manager serves request from preallocated pool

custom datatypes for storage

low overhead module and database API



Performance

on a standard server several tousands calls per second transactional throughput is no problem

a server with enough memory could manage 300.000 users

proxy and registrar performance depends on database speed

using memory as cache helps, trade-of between safety and performance

adjust private and shared memory to use all available memory

local read-only mysql slave helps too

uses multi-process modell to use all available CPUs or cores

Users and competition

OpenSER is used from carriers and service providers like Arcor, Telefonica, 1&1, sipgate..

Cisco sells nice boxes with OpenSER

Microsoft has now a SIP server too..

SER

- *OpenSER was forked from this base some time ago the codebases and configuration now differ somewhat*

projects like Yate, FreeSWITCH or Asterisk have another focus

project and community

healthy user community

more than 25 developers

easy and fast integration of bugfixes and patches

regular and short release cycle

fast decisisions with little overhead

regular meetings and courses in the EU and USA

helpful and friendly IRC channel

Licence and contributions

plain GPL v2 (or later)

contributors don't need to give away their copyright

recent contributions

- berkeley db driver from Cisco
- carrierroute and core enchancements from 1&1
- LDAP and H.350 modules from University of California
- SCTP support from Conneticut College
- perlvdb and benchmark modules from Collax
- presence modules and core enchancements from Voice Systems
- and many others..

Contributors need to maintain their module for one year to get developer status

Actual release and roadmap

version 1.3.0 was released on the 13. december 2007

this release incorporates several new modules and many enchancements in exitings modules and in the core

most of the functionality that is needed is probably now available

now more focus on cleanups and enhancements

documentation is a problem, as usual

Roadmap

cleanups for better maintainability

- remove code duplication
- refactor existing code
- unify core APIs

features

- smaller core enhancements
- extend carrierroute, make it more flexibility
- database API improvements

documentation

- add doxygen to existing code
- a good tutorial is really missing..
- improve the wiki
- improve module documentation

Do it yourself

the basics are easy

the problem lies in the details

especially if you want to earn money with your service

NAT, accounting and compatibility issues

high-availability and failover

quality assurence is really important

How to start your own VoIP service

install the actual (1.3.0) stable release of the server

install a mysql database

enable proxy and registrar functionality in the configuration

setup some user accounts

configure your phones

if something don't work, look in the wiki, increase the debug level and ask google

then ask at the user list..

have fun!

Thank you very much!

Contact and further informations:

henning.westerholt@1und1.de
www.openser.org, OpenSER user and developer mailing lists, #openser on freenode

Pictures:

slide 3: Bill Liao, http://www.flickr.com/people/liao/
slide 5: Björn Söderqvist, http://www.flickr.com/people/kapten/
slide 7: Gary Hunt, http://www.flickr.com/people/e06158/
slide 9: Gaetan Lee, http://www.flickr.com/people/gaetanlee/
slide 10: Emmanuel Schaffner, http://www.flickr.com/people/dagring/
slide 11: Aaron Kuhn, http://www.flickr.coom/people/aaronk/
slide 13: Jim Frazier, http://www.flickr.com/people/jimfrazier/
slide 18: Ard Hesselink, http://www.flickr.com/people/docman/

Licence of the slides: