

OpenSER – an introduction

Welcome!

Henning Westerholt

OpenSER project

1&1 Internet AG

Spring Von.x 2008, 17.03.2008



Outline

- 1. overview of 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. how to use it by yourself*

OpenSER – open SIP express router

component of VoIP infrastructures

provides core services

proxy

registrar

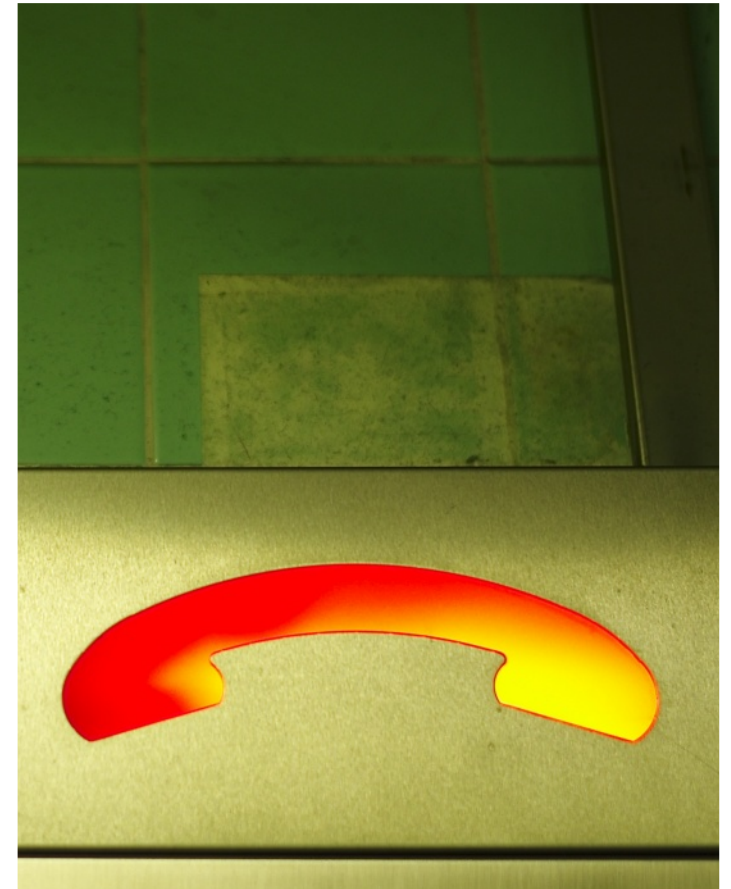
balancer or router

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 signalling, 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 slightly 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!



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 decisions 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 interchangeable

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 large 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

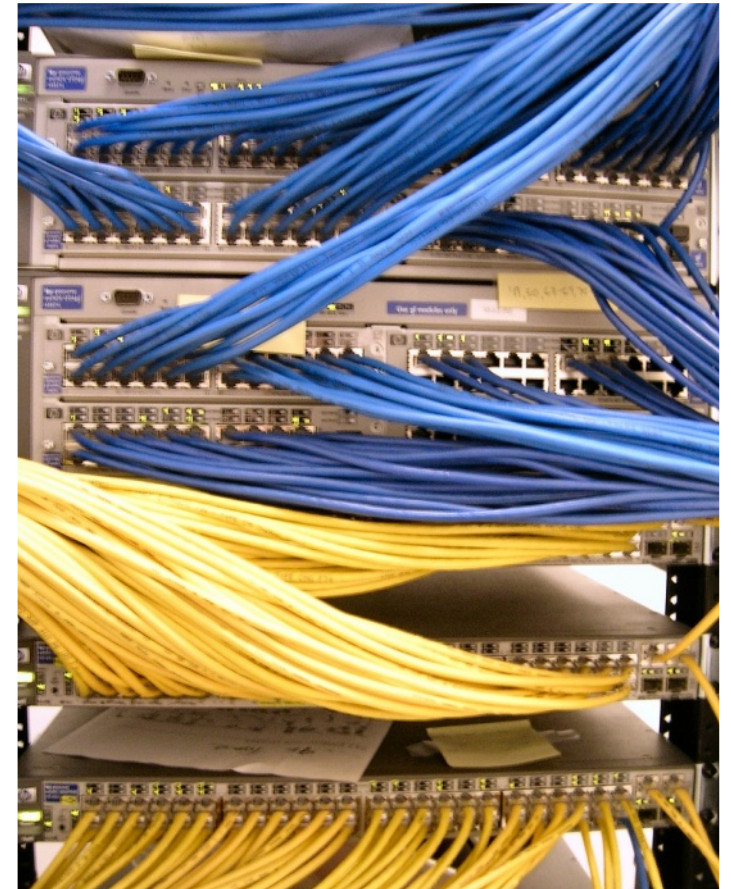
regular expressions are compiled

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 thousands 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

Project and community



healthy user community

26 developers, many more contributors

*easy and fast integration of bugfixes
and patches*

regular and short release cycle

fast decisions with little overhead

*regular meetings and courses in the
EU and USA*

helpful and friendly IRC channel

Actual release and roadmap

version 1.3.1 was released on the 11. March 2008

this release incorporates several new modules and many enhancements in existing 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



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 assurance is really important

participate in the community

Thank you very much!

Contact and further informations:

E-Mail: henning.westerholt@lund1.de, OpenSER user and developer mailing lists

Web: www.openser.org

IRC: #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 6: Gary Hunt, <http://www.flickr.com/people/e06158/>

slide 8: Gaetan Lee, <http://www.flickr.com/people/gaetanlee/>

slide 9: Emmanuel Schaffner, <http://www.flickr.com/people/dagring/>

slide 10: Aaron Kuhn, <http://www.flickr.com/people/aaronk/>

slide 14: Ard Hesselink, <http://www.flickr.com/people/docman/>

Licence of the slides: 

<http://creativecommons.org/licenses/by-nc-sa/2.0/de/>