How and why you should use Kamailio®?

Henning Westerholt

COMMCON 2019

July 2019 - London

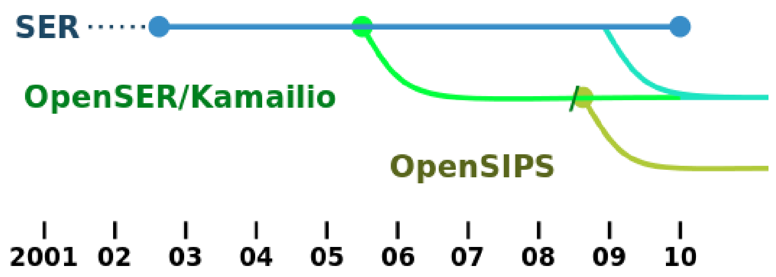
Agenda

- About me
- Kamailio® Overview
  - Usage and features
  - Open Source & Business Community
  - Release and management
- Features in upcoming version 5.3
- Integration of Kamailio into your stack
- Usage scenarios
- Performance and availability experience
- Contact

About me

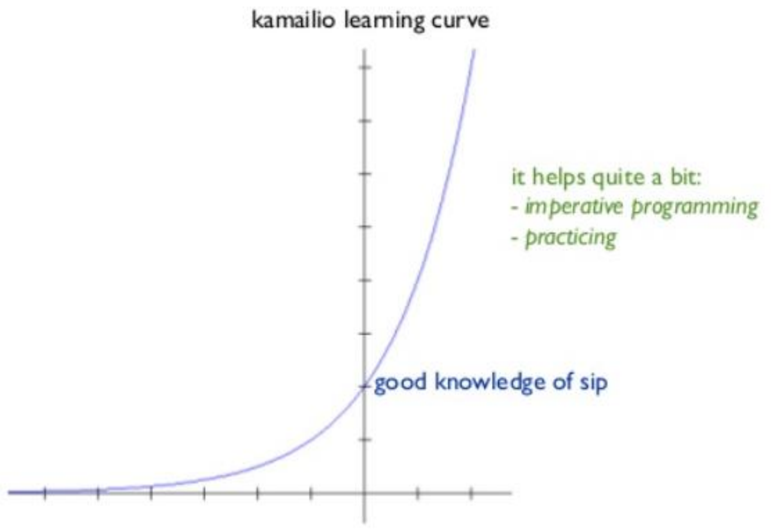
- Henning Westerholt
- With Kamailio project since 2007
- Core developer of the Kamailio® project, member of management board
  - Core, database work and different other modules
  - Administration, code quality, security testing, quality assurance
  - Kamailio fuzzing project
- Own company started in 2018
  - Consulting for Kamailio and Voice over IP services
  - Services and development, Workshops and trainings

Kamailio overview



- Kamailio®
  - Carrier grade SIP Server, Open Source (GPL), continuous development since 2001
  - Implementation of medium to large VoIP/Real-time Communication platforms
  - Building block for large commercial platforms and major institutions
  - Usual first use case for newcomer: scale existing PBX infrastructure
- Many core functions and over 220 extension modules
- Active and world-wide diverse development community
- Statics of the last 12 month
  - 71 authors with about 1770 commits to the git repository
  - One major release (5.2) and 9 maintenance releases created

Kamailio overview

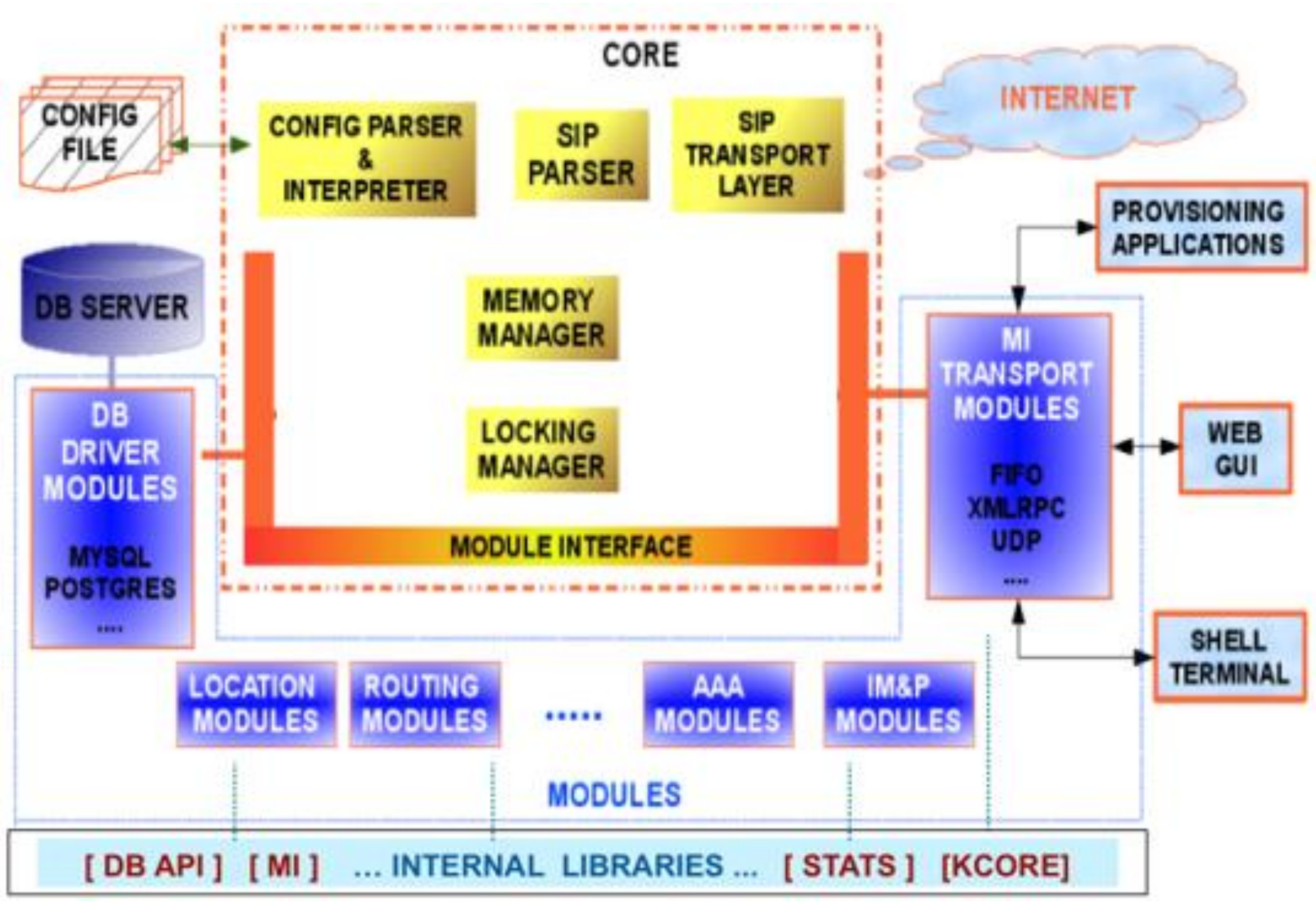


- Kamailio is in its core a SIP proxy
  - No PBX like Asterisk, no media server
  - It does not provide a „B2BUA“, therefore no strict in/out dialog separation
- Learning curve for Kamailio is different than other VoIP server software
  - No dial plan or other similar cfg, it is more like a programming language
  - Good SIP knowledge recommended and necessary for effective usage
- Kamailio operation and maintenance can be done with little effort
  - Stable, good performance and hardware utilization
  - Low dependencies in core and major modules, packaged for major distributions

Kamailio features overview

asynchronous TCP, UDP and SCTP	secure communication via TLS for VoIP (voice, video, text)	WebSocket support for WebRTC	IPv4 and IPv6	SIMPLE instant messaging and presence with embedded XCAP server and MSRP relay
asynchronous operations and timer trigger	VoIP applications server	IMS extensions for VoLTE	ENUM or other API triggered routing	DID and least cost routing
load balancing	SIP proxy and registration	Security and VoIP application level firewall	routing fail-over	Accounting to different targets
authentication and authorization	MySQL, Postgres, Oracle, Radius, LDAP, Redis, Cassandra, MongoDB, Memcached..	JSON and XMLRPC control interface	SNMP and Prometheus monitoring	Embedded scripting languages

Kamailio architecture



Open Source & Business Community

Awards 2019

SimCon Open Source excellence

Google Open Source Peer Bonus

Open Source community

Management board

Core developers

Module maintainer and packager

Different Contributors from companies or academia

Users and developer on mailing list

Business Community

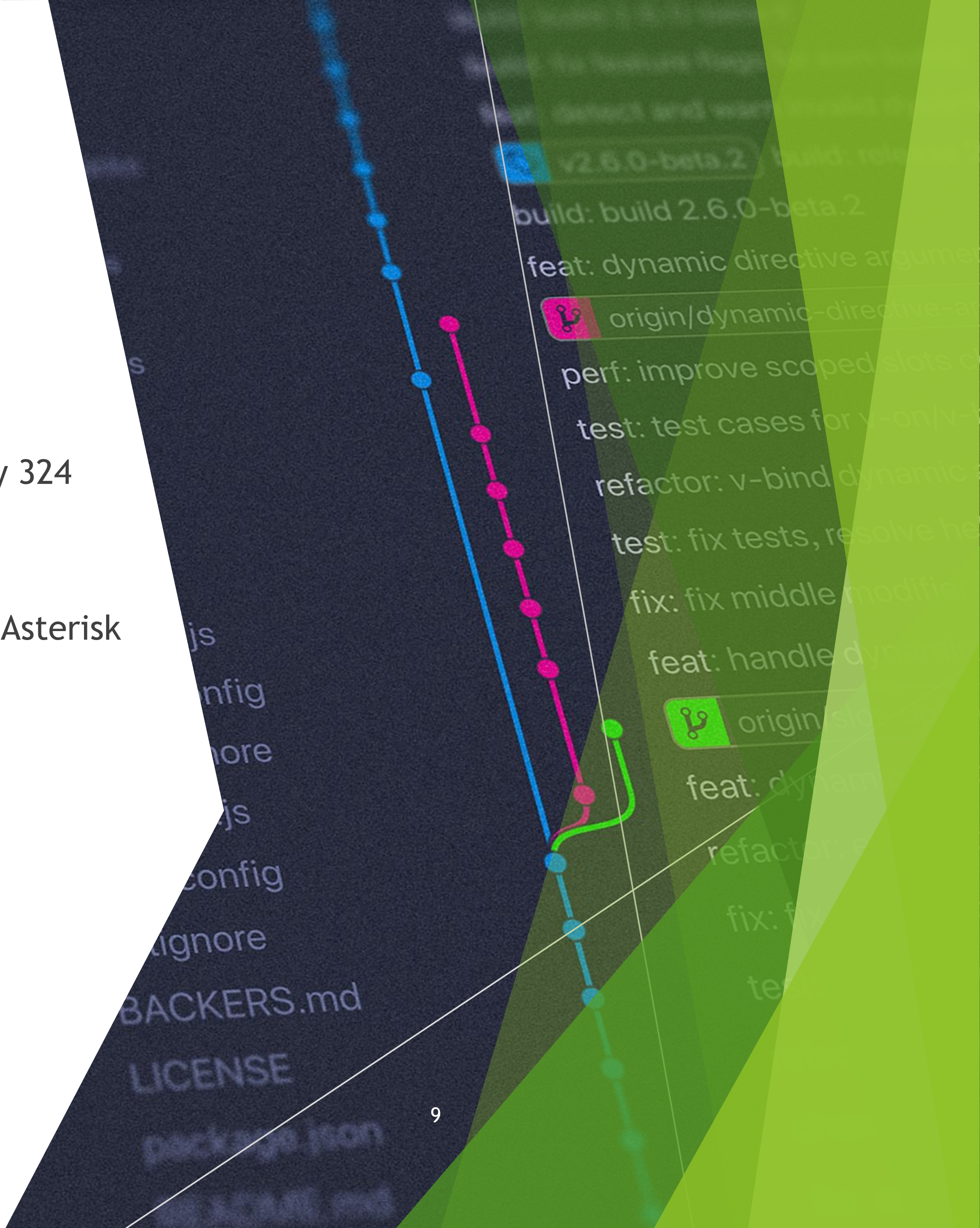
Many companies provides services for Kamailio world-wide

Over 20 officially listed in our business directory

Used from 2nd largest telephony provider in Germany and many other providers world-wide

Source code and release planning

- Source code statistics over project
  - About 964.000 lines of code, almost 30.000 commits made by 324 contributors
  - 256 years of effort
- Comparison: roughly double the size of Opensips, half the size of Asterisk
- Kamailio uses a time-based release schedule
- A major version is released every 10-12 months
  - Development phase of about 8 month
  - Followed by a code freeze/testing phase of 2 month
- Minor version are released roughly every 2 month
- Release date for 5.3 release after summer vacation in autum



New in version 5.3 (1/2)

- High-level overview
  - 5 new modules
  - Changes in 32 existing modules
  - Many changes in core, API refactoring and tools
  - About 1000 commits from 62 authors (development still ongoing!)
- New modules: rtp_media_server, secfilter, xhttp_prom, kemix, app_lua_sr
- Important changes in existing modules
  - IPv6 and TCP support for IMS IPSEC modules
  - Direct modification of dispatcher routing in-memory over RPC
  - Major additions to rtpengine (play media, forward media, improved load balancing)
  - Support for RFC3327 (302 redirects with Path vector)

New in version 5.3 (2/2)

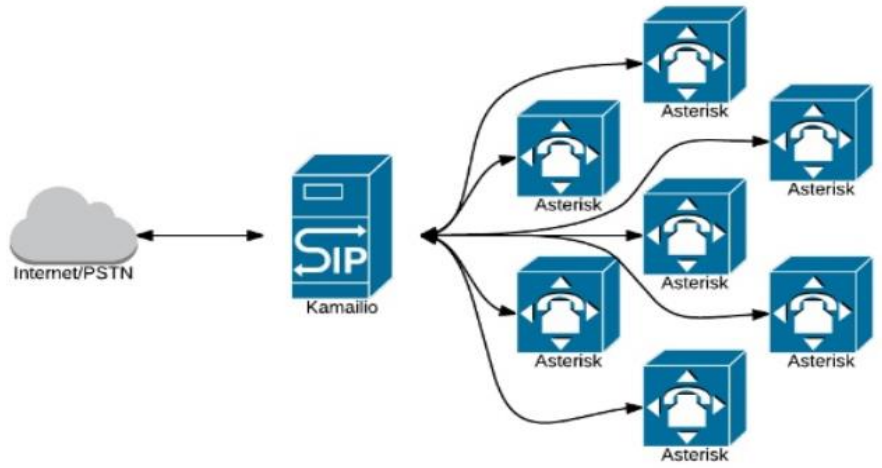
- Several new cmd-line parameters, help with dynamic deployments e.g. container
- Many extensions to the Kamailio scripting language interface KEMI
- The uac module works much better in the dialog variable storage mode
- The tls module works now properly with OpenSSL >1.1.x
- Major extensions and refactoring for the imc module
- Many new pseudo-variables to interact with Kamailio or access data from cfg
- Add support for link-local IPv6 addresses
- Unification of memory allocation error handling
- Support for HAProxy protocol, e.g. Kamailio behind AWS ELB balancers
- Unification of database version table error handling

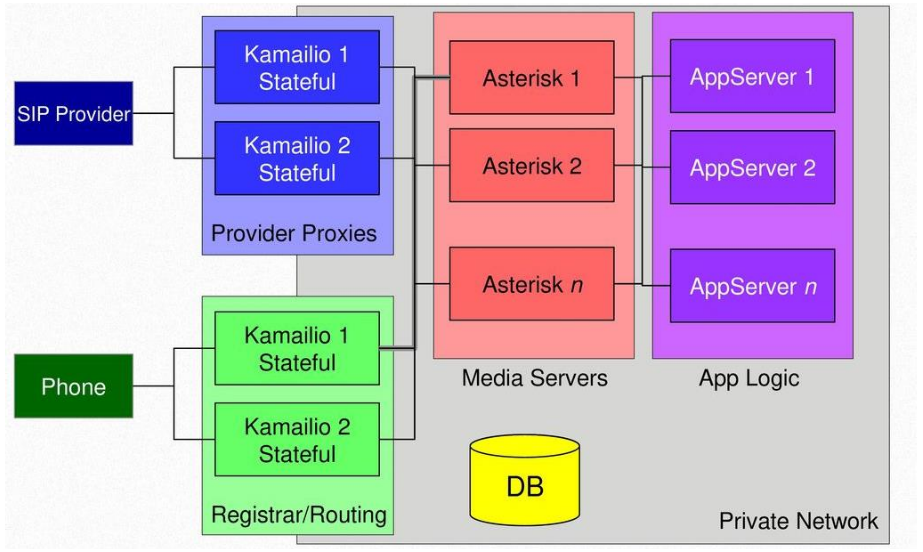
More information: https://www.kamailio.org/wiki/features/new-in-devel

Integration of Kamailio into your stack

*not all supported yet

- Development strategy
  - With existing programming knowledge or code in scripting languages use „KEMI“
  - Python, Lua or Javascript are the common ones, more are available
  - Kamailio supports also test-driven development models, if you develops that way
- Accessing data from Kamailio
  - Existing open source databases works great, use MySQL or PostgreSQL
  - Proprietary DBs can be connected with unixODBC or with REST/JSON API interface
- Getting data out of Kamailio
  - From database or log files
  - with REST/JSON/XML API





Common scenarios

- Secure and scale your existing PBX infrastructure
- Adapt or translate between different SIP dialects
- Implement stable and high-performance SIP application services

High-performance setup

- How to route thousands of calls per second on a standard server/VM?
- Easy, just install Kamailio ☺
- You will have a hard time actually overload your server with Kamailio
- I/O performance and external requests are usually the bottlenecks
- Analyse performance issues in a structured way
- Usual sizes for setup (depending on complexity and architecture)
  - 10.000 - 20.000 users: 2-4 server
  - 20.000 – 100.000 users: about 10 server
  - 1&1 with more than 10 million subscriber: hundreds of server

Kamailio in a distributed setup

- Kamailio works fine with usual distributed databases
  - Master-slave or Master-Master setup
  - Different Cluster technologies
- For new deployments you can get rid of the database completely with DMQ
  - Distributed message queue on Kamailio to enable the passing & replication of data between multiple Kamailio servers
  - DMQ nodes are connected together in a logical "bus" topology and are able to communicate with each other
  - DMQ handles the node discovery, consistency, retransmissions etc..
  - Support for user location, dialogs, htable and other modules available

Kamailio operation

- Everybody started once small
- No “secret sauce” for operating a large high-availability setup
- Security and process quality are the most important areas
- Iterative approach works best in my experience

Stay secure with Kamailio

Keep your system up to date

Use a support distribution release

Actually apply the updates in time (at least monthly)

Protect your system

Only login for necessary users, SSH keys

Provisioning only necessary services

Bind databases and other APIs only to localhost or private network

Use fail2ban to protect your ssh login

Keep Kamailio up to date

The last two stables releases are supported (e.g. 5.1 and 5.2)

End of maintenance will be announced

Minor releases don’t need any cfg adaptions

Operational notes

To operate Kamailio a certain SIP knowledge helps (again), some tools (sngrep, homer etc..) are available

Do not change too many things at once, keep track of your changes, preferable in git/svn, use a ticket system

Add logging information in your configuration when a check triggers and something is blocked

You will spend a significant time on certain user agents and also carriers that not behave correctly

Thank you for your attention

- Contact:
  - Henning Westerholt
  - mail@skalatan.de
  - https://skalatan.de/services