Homer

... because sip capturing makes sense

Author: Alexandr Dubovikov
Co-authors: Torsten Schweizer, Heino Klier, Roland Haenel

QSC AG

About QSC

QSC – ICT solutions for small and mid-size enterprises

QSC AG, Cologne, is a service provider for voice and datacommunication, as well as the ICT services that build upon them. Established in 1997, the company has been focusing on small and mid-size business customers. QSC is the first provider to operate an Open Access platform, which unites a wide range of broadband technologies to offer national and international site networking, including Managed Services. QSC additionally supplies its customers and distribution partners with a comprehensive product portfolio that can be modularly adapted to every need. QSC was the first provider in Germany to build its own Next Generation Network (NGN), and therefore enjoys long years of experience in connection with IP-based telephony solutions, in particular. QSC employs a workforce of some 700 people and has been listed on the TecDAX index since 2004.

Capturing tools

- Tcpdump
- Ngrep
- Sipgrep
- Wireshark
- Sipspy

All these tools are just able to capture in realtime!

But we have to look into history!

Why do we need capturing?

Example Scenario:

- A customer complains experiencing problems with reaching a special phone number. So to discover the problem and locate the faulty device in the network you normally have to do a live trace together with the customer. But you do not want to bother him with test calls.
- This is the big benefit of HOMER! With HOMER we are able to search for the faulty call and get results retrospectively to the call flow from every involved network device.

A trace Tool with backtrace functionality is needed:

Homer

Homer Simpson © 20th Century Fox

A trace Tool with backtrace functionality is needed:

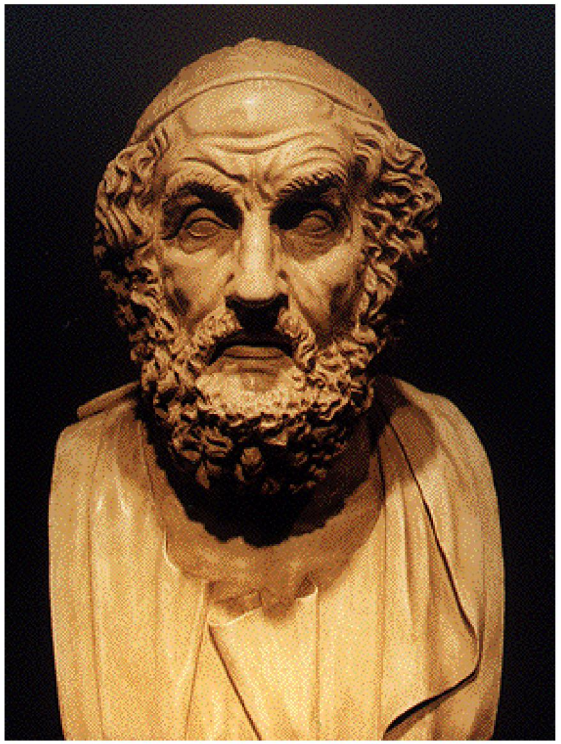
Homer

Homer Simpson © 20th Century Fox

A trace Tool with backtrace functionality is needed:

Homer

Homer Simpson © 20th Century Fox



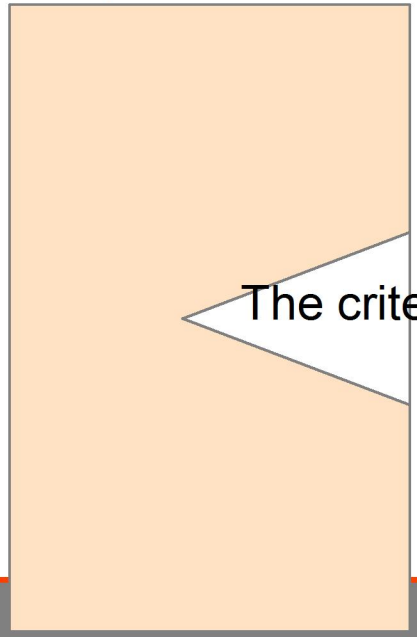
A trace Tool with backtrace functionality is needed:

Homer

Homer Simpson © 20th Century Fox

Why Homer?

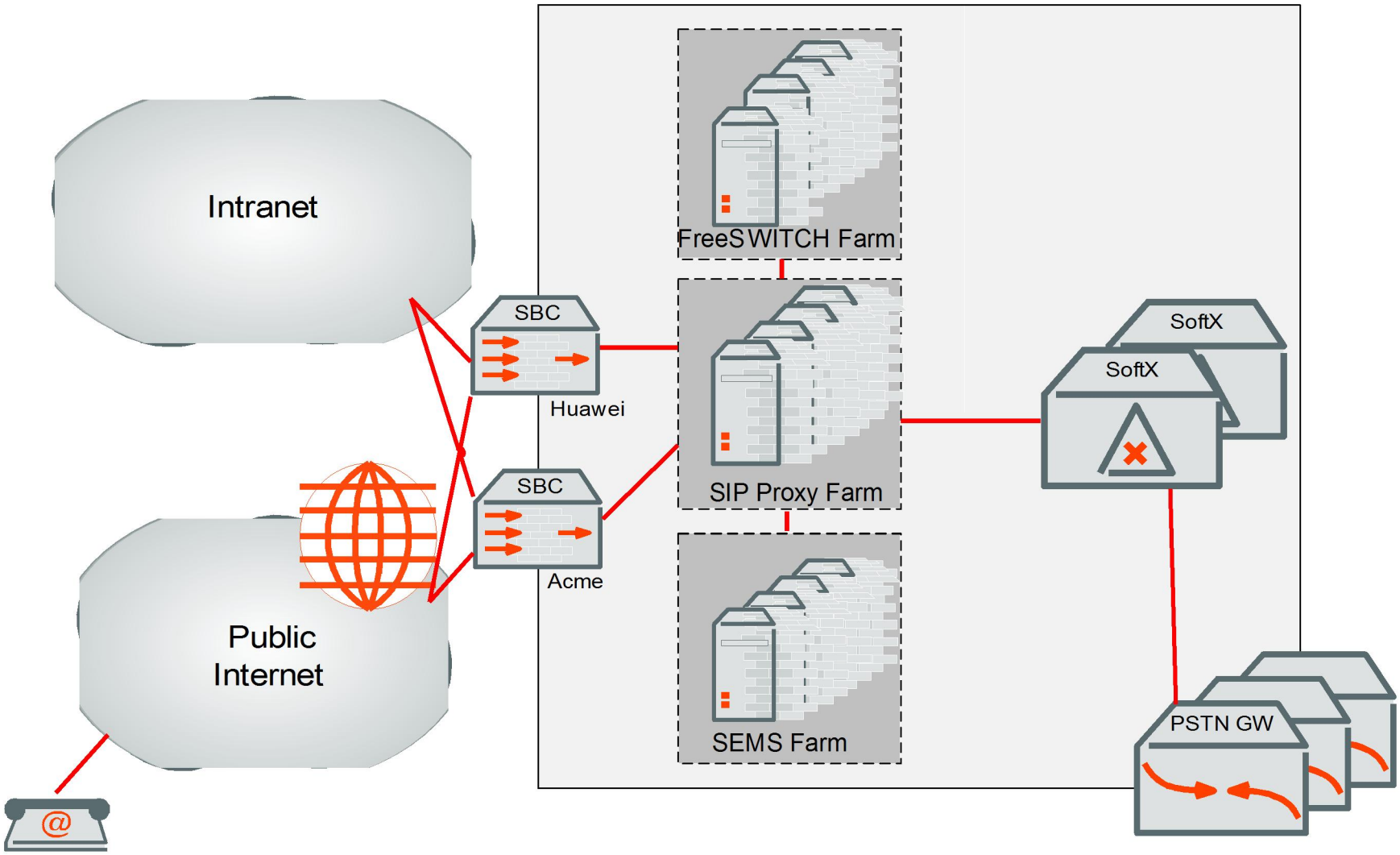
- it collects data and captured messages
- storing the collected data in DB
- querying, filtering and displaying of data via webinterface (GUI)



Normal SIP/VoIP network components

- SBC (Session Border Controller)
- Softswitch/Gateway
- SIP proxy/registrar/router
- SIP applications/voice2email/fax2email/IVR

NGN Network Overview



Centralized – Vendor independent

- There are many different system components in a SIP network.
- By default many vendors support IP Proto 4 (IP in IP encaps.) for capturing solutions. e.g. ACME Packet, Huawei ...
- Our goal is to bring all SIP components together in a centralized controlling and monitoring system.
- As a result you have the complete call flow through all components of your VoIP network.

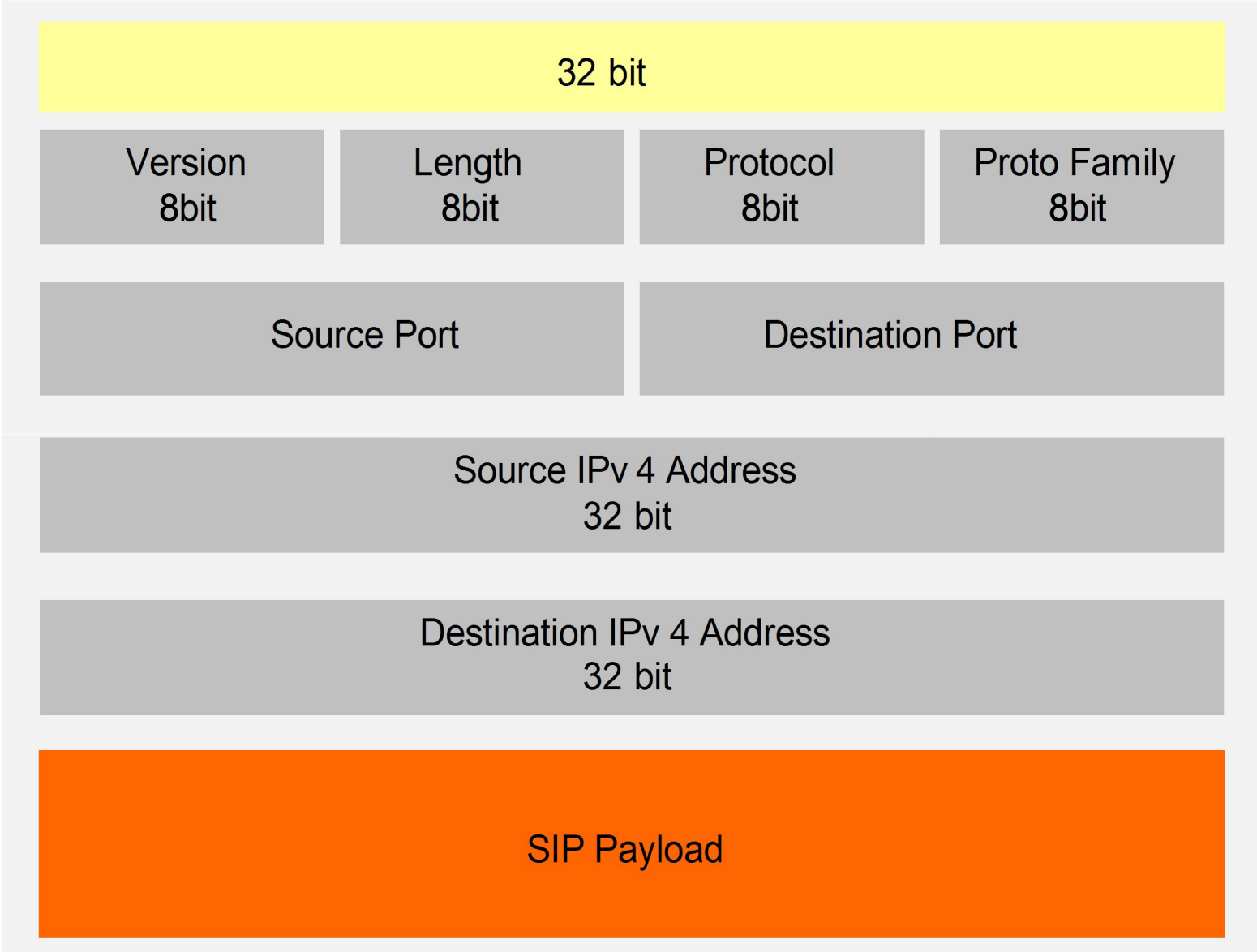
Homer is based on:

- External capturing agent (if needed)
- Capturing nodes
- Capturing database
- Web frontend (GUI)

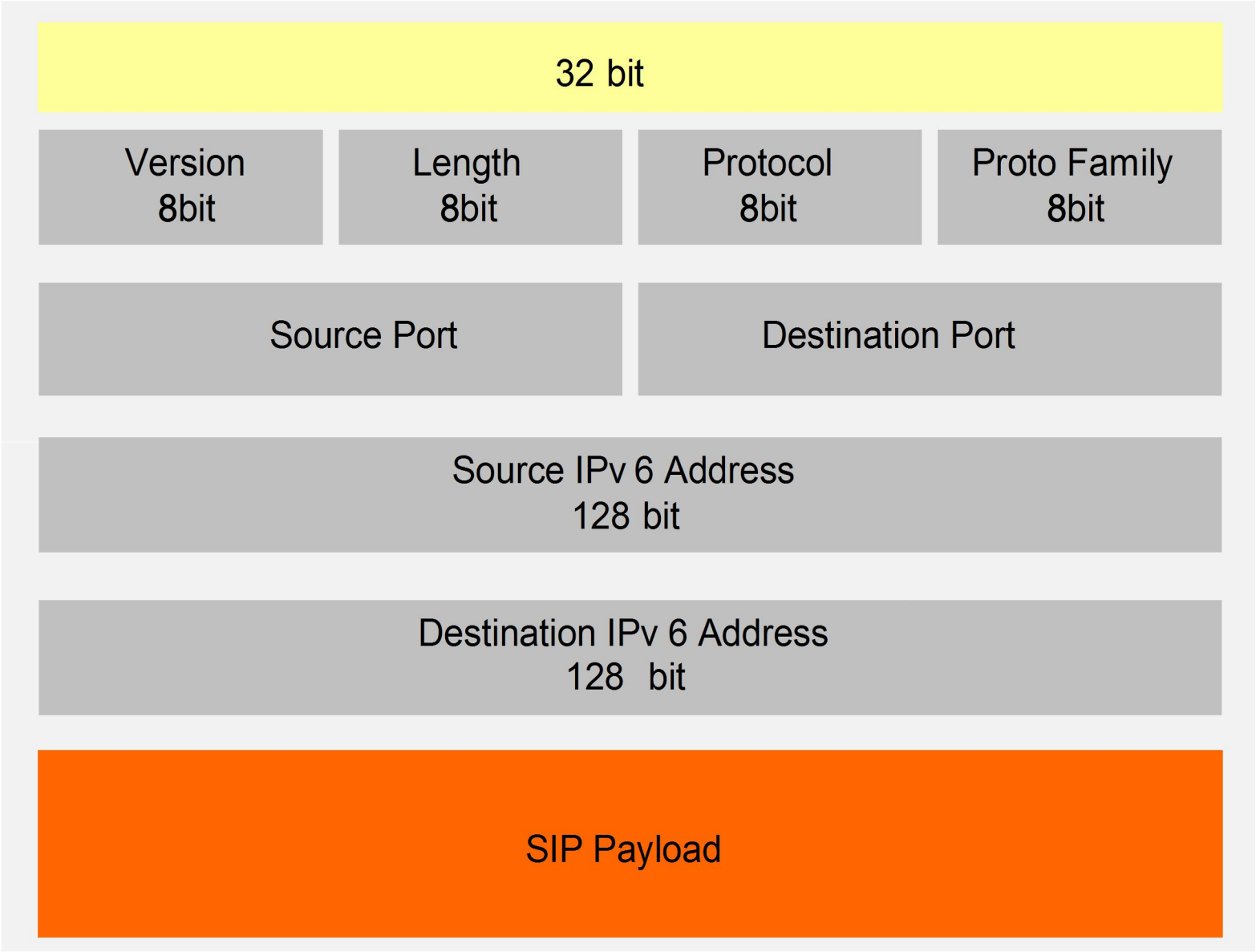
HEP - Homer Encapsulation Protocol

- self developed encapsulation protocol
- no need of root privileges or kernel changes like IPIP
- IPv6 and IPv4
- support many IP protocols (TCP,UDP,SCTP)
- can be used not only for SIP

HEP – Homer Encapsulated Protocol IPv4



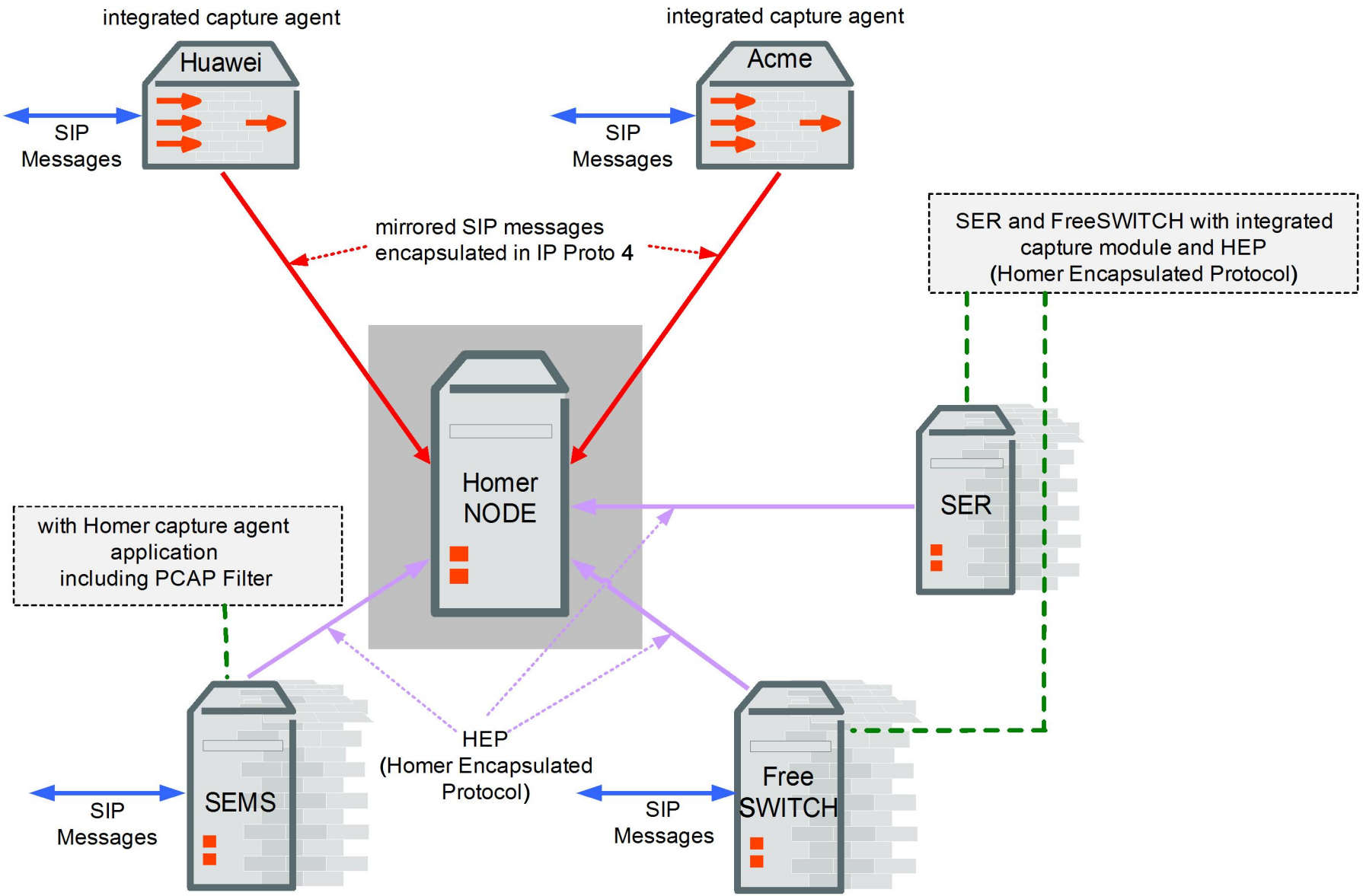
HEP – Homer Encapsulated Protocol IPv6



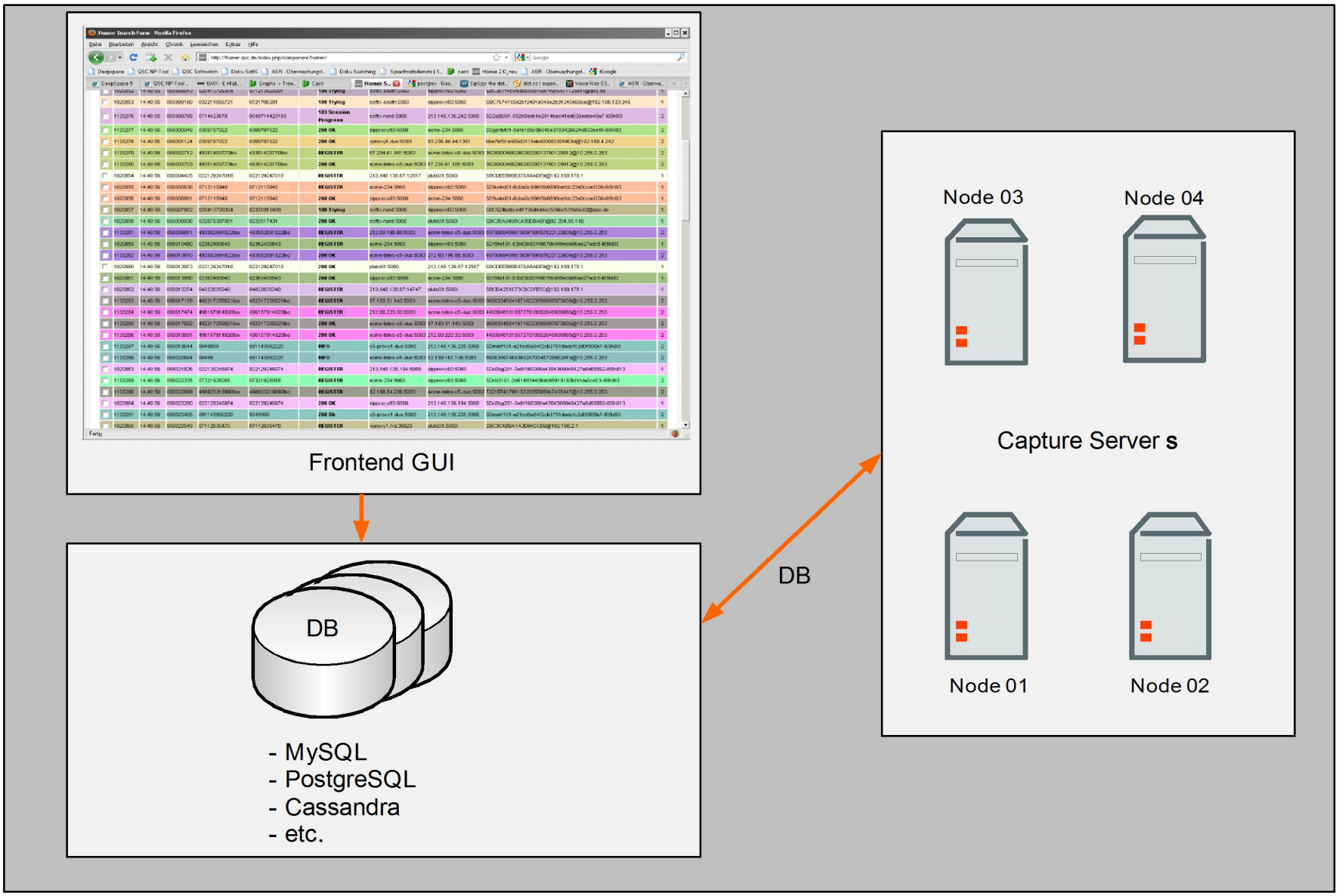
Capturing agent

- The capturing agent acts as a daemon process on operation systems like UNIX (also possible as a Windows component)
- The agent duplicates all SIP traffic in HEP to the Homer node.
- The agent uses the pcap lib. Therefore you can set up your own pcap filter to duplicate only needed traffic e.g. only outgoing messages .
- The agent is extremely small, with only 300 lines of C-code and therefore goes easy on resources.
- It will be nice if the capturing agent would be integrated in many other open source projects (OpenSIPS, SEMS, Asterisk, Yate), because it is already implemented in FreeSWITCH and Kamailio.

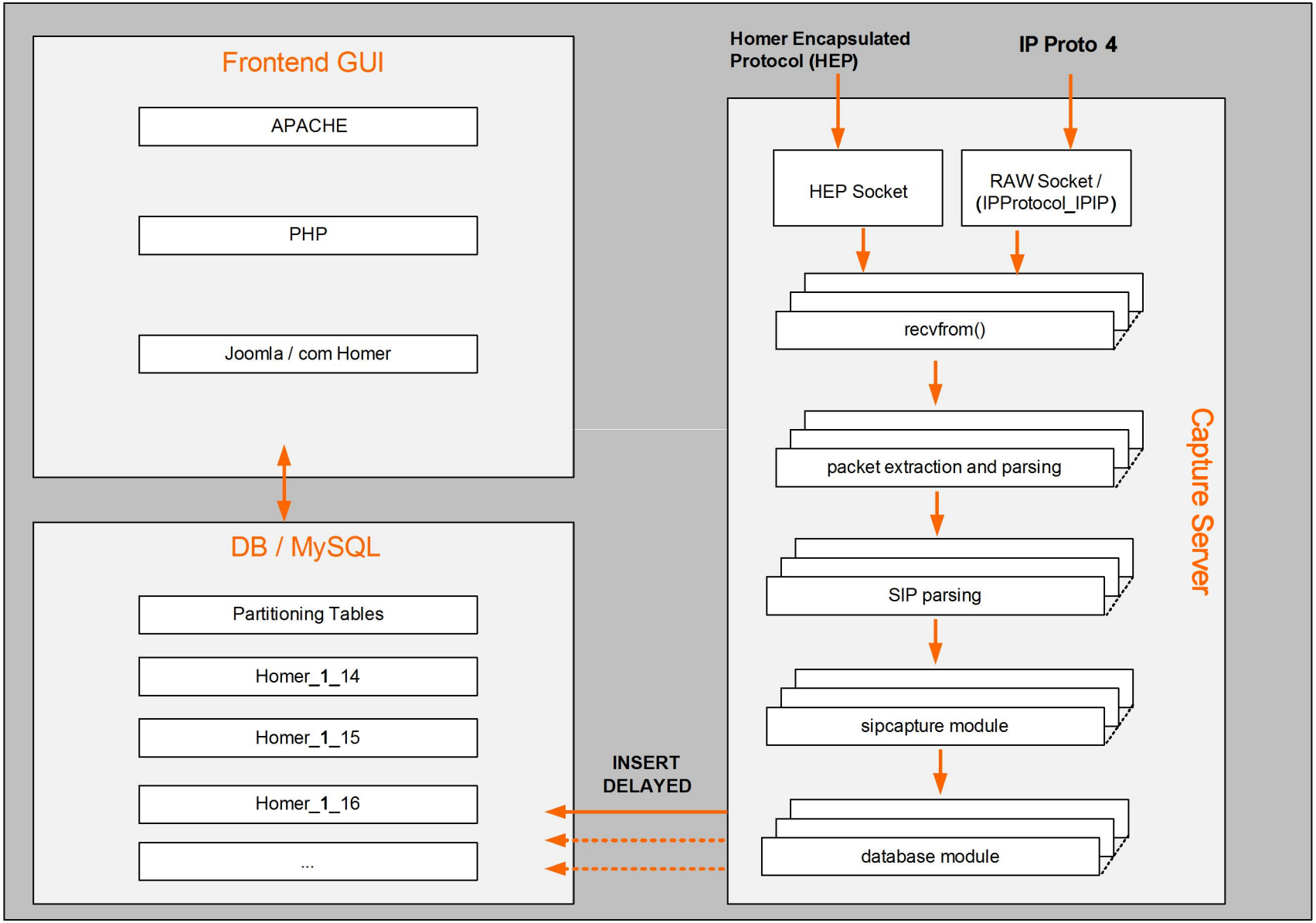
Homer Overview



Homer



Homer



Capturing node

The capturing node is a UNIX based server (in our case Ubuntu).

The core component of the node is the capturing application server which

- receives IP Proto 4 (IPIP) packets
- receives HEP packets
- validates if they are SIP
- parses the packets and
- inserts the values to DB

Our capture application is based on SIP-Router aka SER 3.x or kamailio 3.x, because of good core performance and effective SIP parser.

Capturing node

Why a SIP-Router (SER)?

- core of SER has a very good performance
- SIP parser is effective
- has support for MySQL, PostgreSQL, Oracle ….
- can be compiled on many different UNIX like systems
- big community
- Open Source

Capturing node

- raw socket mode for IPIP encapsulation
- UDP socket for HEP
- parsing the elements of the SIP packet
- inserted into a DB through sipcapture and database modules.
- In our case we use MySQL and INSERT DELAYED , which causes no socket IO-wait between SER and DB (insert and forget).

Capturing database

Normally you can use any relational DB (MySQL, PostgreSQL, Oracle ...) but if you want to build a really big capturing cluster we recommend to use key-value DB (Cassandra, MongoDB etc).

In case of key-value DB (Cassandra) all DB nodes will have the same capturing data which guarantees high availability.

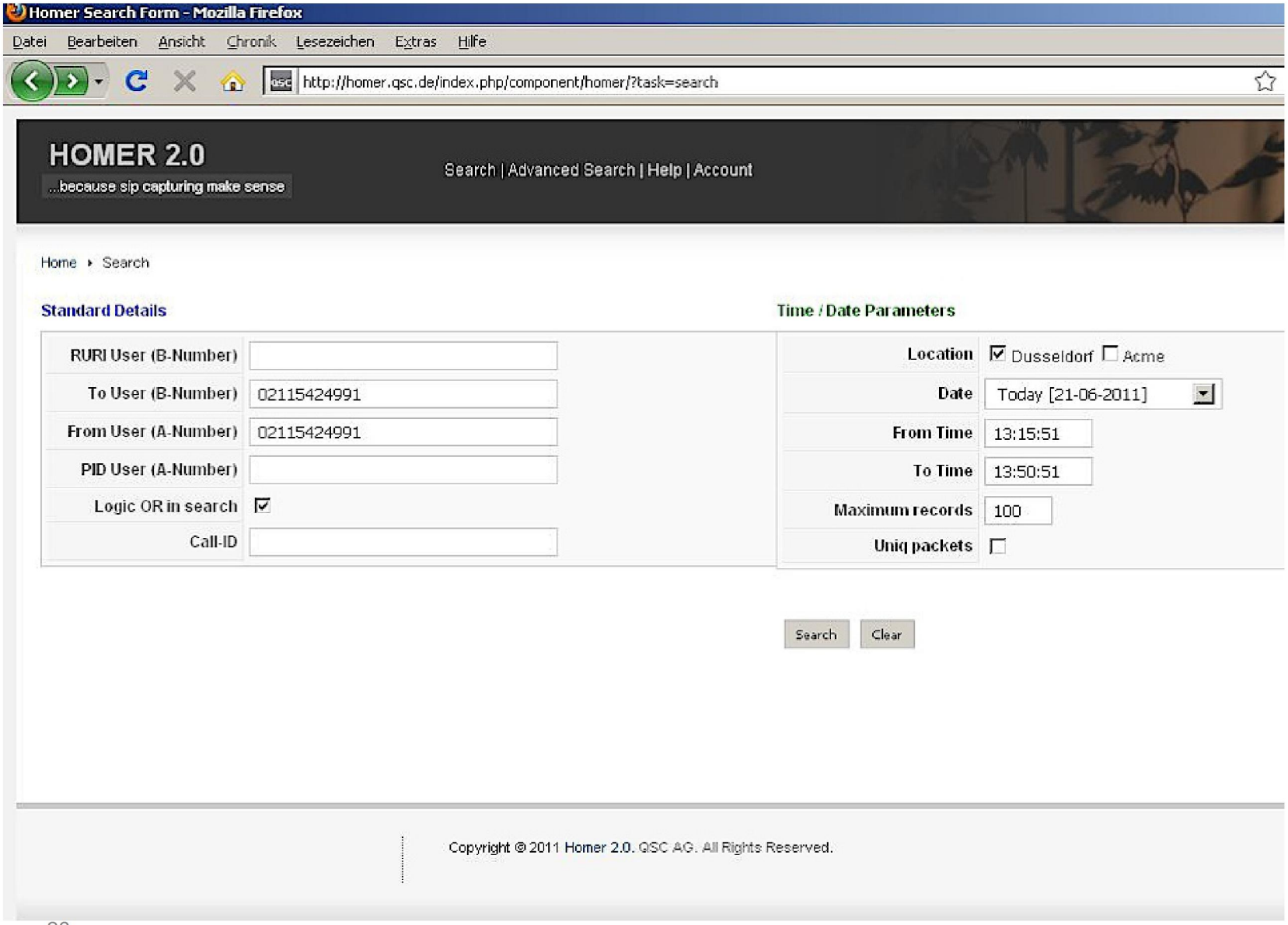
Frontend

The Homer GUI is based on Joomla CMS which is also Open Source. Joomla has an internal user management and a good php API.

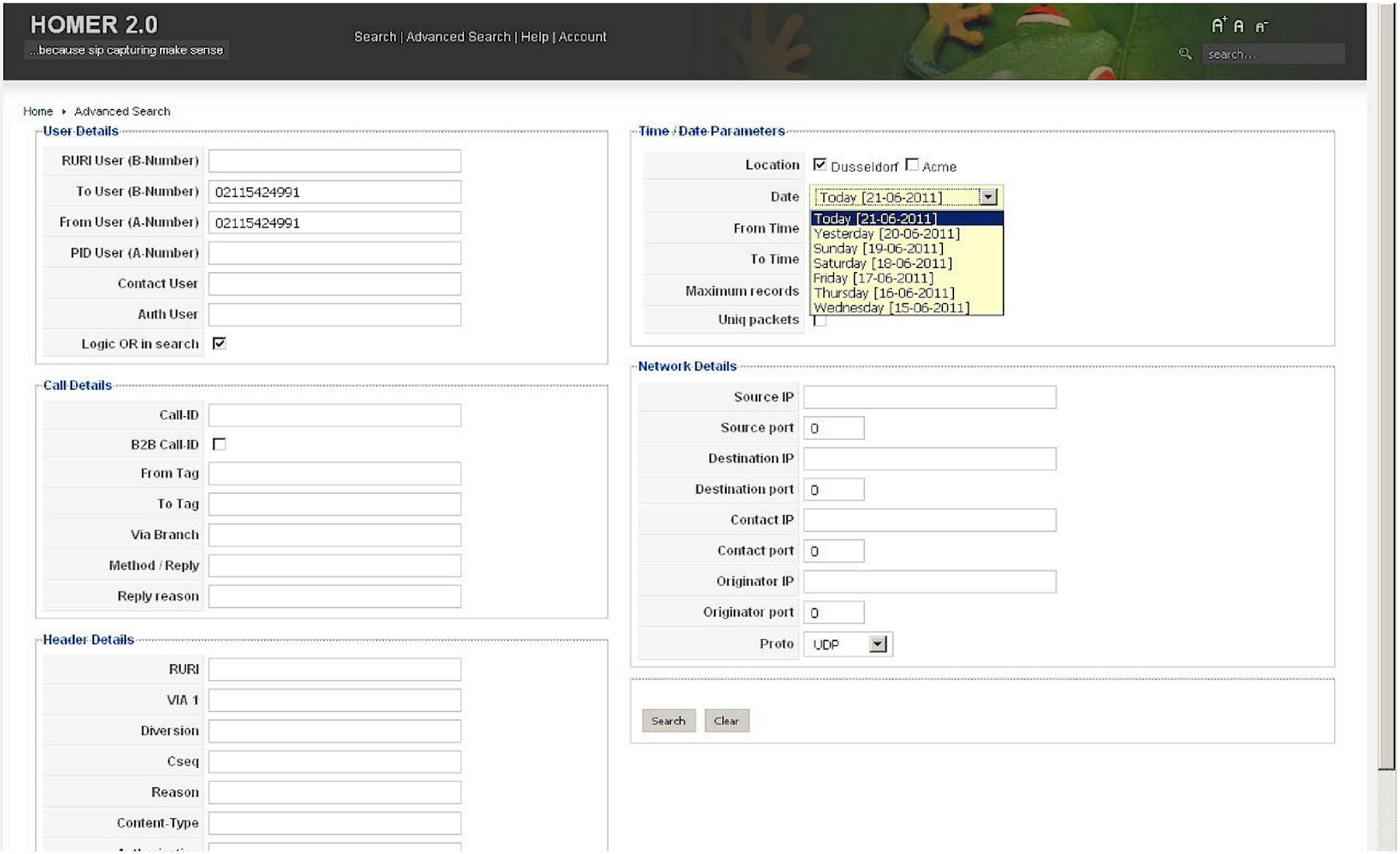
Our frontend provides the following operational capabilities:

- Search on many different parameters (A-number, B-number, Date,Time, Call-ID, From Tag, To Tag, Method Type, User Agent, Source IP, Destination IP, Port, Protocol Type etc.)
- combining search options
- get detailed information by selecting a single message
- display information with CallFlow sequence diagram
- for a quick overview calls are grouped in different colors
- convert and save trace output as pcap file.

GUI simple search form



GUI advanced search form



GUI search result

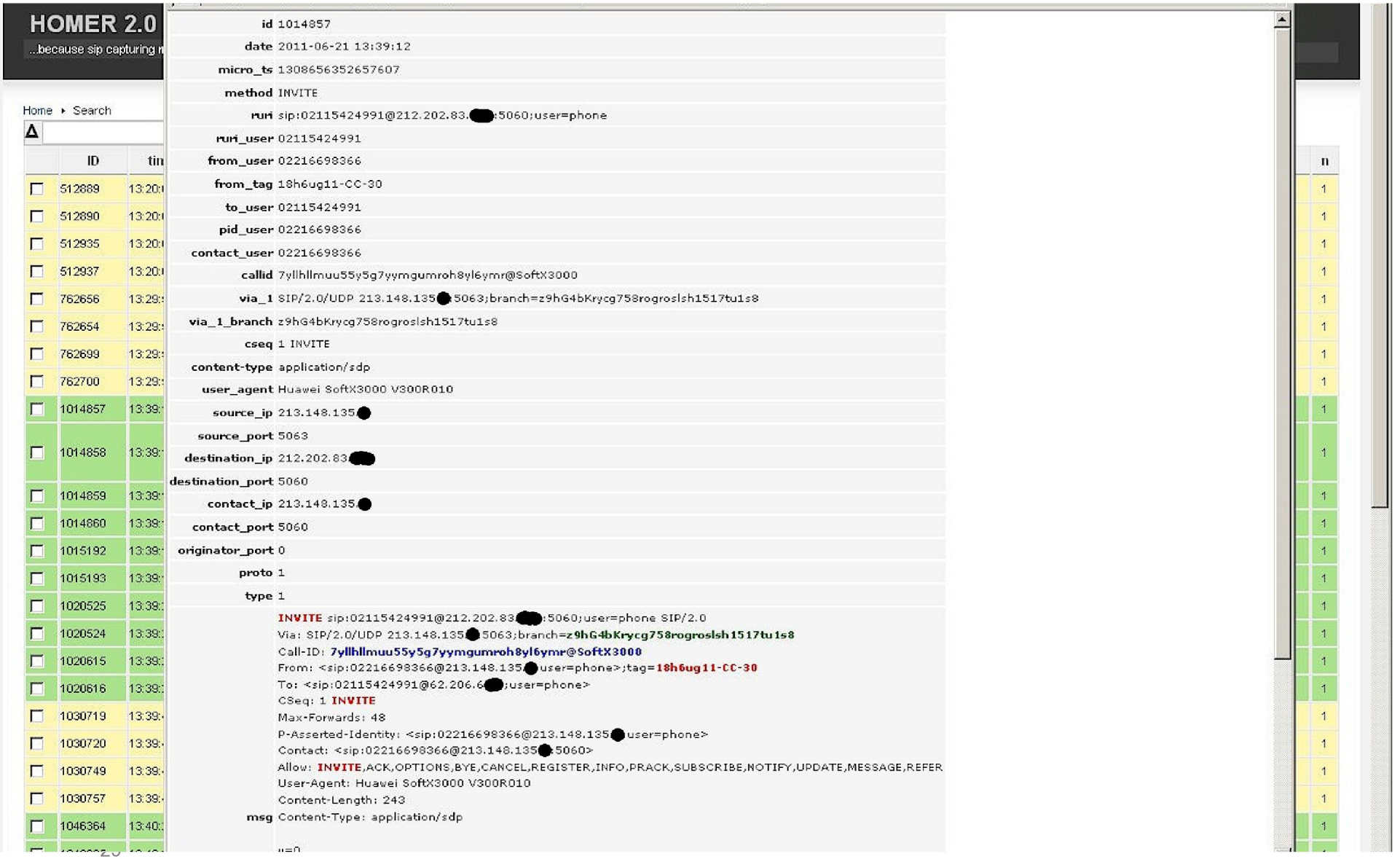
Home ▸ Search

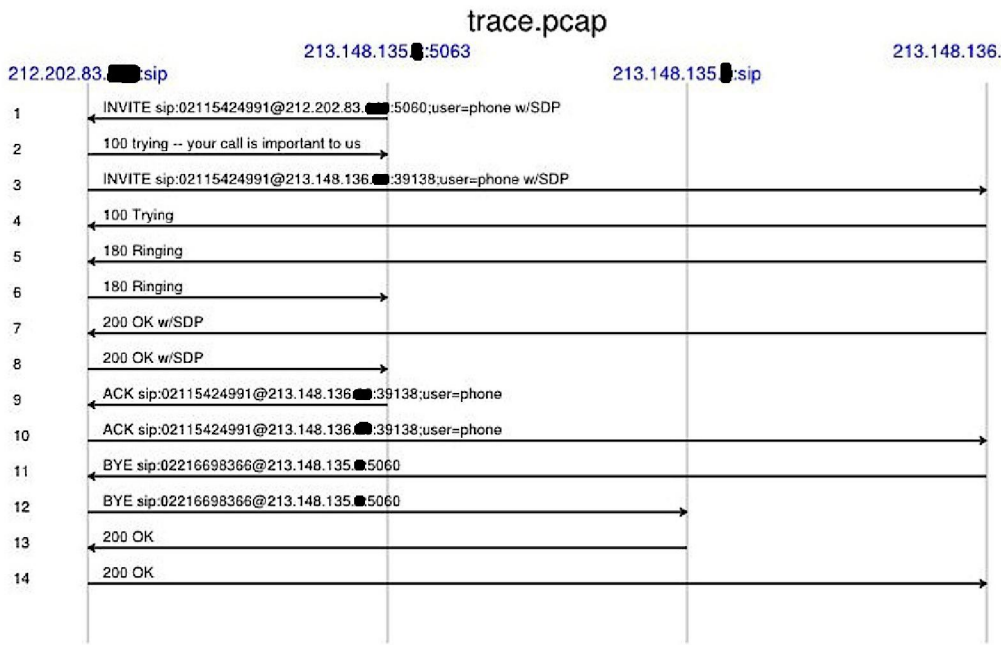
Δ 13086563526576 - 13086564232976 = 706400 µ

Result: 21-06-2011 13:15:51 - 21-06-2011 13:50:51

	ID	time	µ	From User	To User	PID	Method	From ip	To ip	Callid	n
☐	512889	13:20:03	203834861	02115424991	02115424991		REGISTER	vproxy1.dus:39138	sipproxy03:5060	SBC2999968522@10_0_0_200	1
☐	512890	13:20:03	203834968	02115424991	02115424991		401 Unauthorized	sipproxy03:5060	vproxy1.dus:39138	SBC2999968522@10_0_0_200	1
☐	512935	13:20:03	203919169	02115424991	02115424991		REGISTER	vproxy1.dus:39138	sipproxy03:5060	SBC2999968522@10_0_0_200	1
☐	512937	13:20:03	203921012	02115424991	02115424991		200 OK	sipproxy03:5060	vproxy1.dus:39138	SBC2999968522@10_0_0_200	1
☐	762656	13:29:55	795300879	02115424991	02115424991		REGISTER	vproxy1.dus:39138	sipproxy03:5060	SBC2999968522@10_0_0_200	1
☐	762654	13:29:55	795300975	02115424991	02115424991		401 Unauthorized	sipproxy03:5060	vproxy1.dus:39138	SBC2999968522@10_0_0_200	1
☐	762699	13:29:55	795391233	02115424991	02115424991		REGISTER	vproxy1.dus:39138	sipproxy03:5060	SBC2999968522@10_0_0_200	1
☐	762700	13:29:55	795394760	02115424991	02115424991		200 OK	sipproxy03:5060	vproxy1.dus:39138	SBC2999968522@10_0_0_200	1
☑	1014857	13:39:12	352657607	02216698366	02115424991	02216698366	INVITE	softx-nord:5063	sipproxy03:5060	7yllhllmuu55y5g7yymgumroh8yl6ymr@SoftX3000	1
☐	1014858	13:39:12	352661349	02216698366	02115424991		100 trying -- your call is important to us	sipproxy03:5060	softx-nord:5063	7yllhllmuu55y5g7yymgumroh8yl6ymr@SoftX3000	1
☐	1014859	13:39:12	352661386	02216698366	02115424991	02216698366	INVITE	sipproxy03:5060	vproxy1.dus:39138	7yllhllmuu55y5g7yymgumroh8yl6ymr@SoftX3000	1
☐	1014860	13:39:12	352664891	02216698366	02115424991		100 Trying	vproxy1.dus:39138	sipproxy03:5060	7yllhllmuu55y5g7yymgumroh8yl6ymr@SoftX3000	1
☐	1015192	13:39:13	353467950	02216698366	02115424991		180 Ringing	vproxy1.dus:39138	sipproxy03:5060	7yllhllmuu55y5g7yymgumroh8yl6ymr@SoftX3000	1
☐	1015193	13:39:13	353468070	02216698366	02115424991		180 Ringing	sipproxy03:5060	softx-nord:5063	7yllhllmuu55y5g7yymgumroh8yl6ymr@SoftX3000	1
☐	1020525	13:39:24	364793285	02216698366	02115424991		200 OK	vproxy1.dus:39138	sipproxy03:5060	7yllhllmuu55y5g7yymgumroh8yl6ymr@SoftX3000	1
☐	1020524	13:39:24	364793302	02216698366	02115424991		200 OK	sipproxy03:5060	softx-nord:5063	7yllhllmuu55y5g7yymgumroh8yl6ymr@SoftX3000	1
☐	1020615	13:39:25	365017649	02216698366	02115424991		ACK	softx-nord:5063	sipproxy03:5060	7yllhllmuu55y5g7yymgumroh8yl6ymr@SoftX3000	1
☐	1020616	13:39:25	365017770	02216698366	02115424991		ACK	sipproxy03:5060	vproxy1.dus:39138	7yllhllmuu55y5g7yymgumroh8yl6ymr@SoftX3000	1
☐	1030719	13:39:46	386780641	02115424991	02115424991		REGISTER	vproxy1.dus:39138	sipproxy03:5060	SBC2999968522@10_0_0_200	1
☐	1030720	13:39:46	386780745	02115424991	02115424991		401 Unauthorized	sipproxy03:5060	vproxy1.dus:39138	SBC2999968522@10_0_0_200	1
☐	1030749	13:39:46	386875131	02115424991	02115424991		REGISTER	vproxy1.dus:39138	sipproxy03:5060	SBC2999968522@10_0_0_200	1
☐	1030757	13:39:46	386878263	02115424991	02115424991		200 OK	sipproxy03:5060	vproxy1.dus:39138	SBC2999968522@10_0_0_200	1
☐	1046364	13:40:23	423262775	02115424991	02216698366		BYE	vproxy1.dus:39138	sipproxy03:5060	7yllhllmuu55y5g7yymgumroh8yl6ymr@SoftX3000	1
☐	1046365	13:40:23	423262959	02115424991	02216698366		BYE	sipproxy03:5060	softx-nord:5060	7yllhllmuu55y5g7yymgumroh8yl6ymr@SoftX3000	1
☐	1046396	13:40:23	423297671	02115424991	02216698366		200 OK	softx-nord:5060	sipproxy03:5060	7yllhllmuu55y5g7yymgumroh8yl6ymr@SoftX3000	1
☑	1046395	13:40:23	423297682	02115424991	02216698366		200 OK	sipproxy03:5060	vproxy1.dus:39138	7yllhllmuu55y5g7yymgumroh8yl6ymr@SoftX3000	1

GUI SIP message details





Static | Dynamic | Frames

Trace source

```
Frame 11 (489 bytes on wire, 489 bytes captured)
Arrival Time: Jun 21, 2011 13:40:23.111627000
Internet Protocol, Src: 213.148.136.[redacted] (213.148.1
User Datagram Protocol, Src Port: 39138 (39138),
Session Initiation Protocol
    Request-Line: BYE sip:02216698366@213.148.135
        Method: BYE
        Request-URI: sip:02216698366@213.148.135.
            Request-URI User Part: 02216698366
            Request-URI Host Part: 213.148.135.[redacted]
            Request-URI Host Port: 5060
        [Resent Packet: False]
    Message Header
        Via: SIP/2.0/UDP 213.148.136.[redacted]:39138;bra
            Transport: UDP
            Sent-by Address: 213.148.136.[redacted]
            Sent-by port: 39138
            Branch: z9hG4bK3aed86ac6d51e7173ca48d
        Route: <sip:212.202.83.[redacted];lr;ftag=18h6ug
        Call-ID: 7yllhllmuu55y5g7yymgumroh8yl6ymr
        From: <sip:021154249910@62.206.6.[redacted];user=p
            SIP from address: sip:021154249910@62.
                SIP from address User Part: 02115
                SIP from address Host Part: 62.20
            SIP tag: 211818439
        To: <sip:02216698366@213.148.13[redacted];user=p
            SIP to address: sip:02216698366@213.1
                SIP to address User Part: 0221669
                SIP to address Host Part: 213.148
            SIP tag: 18h6ug11-CC-30
        CSeq: 1 BYE
            Sequence Number: 1
            Method: BYE
        Max-Forwards: 70
        User-Agent: S450 IP/022270000000
        Content-Length: 0
```

Capturing capability

- Our experience has shown that DB can easily handle up to 10 mio. packets per hour (depending on hardware).
- Actually we receive 5-6 mio. packets per hour (on two nodes).
- In case of expansion the system can be clustered just by adding new nodes to the system.

CPU Dual Core Xeon 5520, 8 G RAM – 3M packets/hour:

- 8% CPU - MySQL
- 0.2% CPU kamailio in capture mode

load average: 0.25, 0.18, 0.12

What Homer is now...

- IPv4 and IPv6 support
- Scalability
- Good performance
- Capture agent integrated in FreeSWITCH, Kamailio
- Can easily be used in any SIP networks

..and Homer in the future...

- support for XMPP protocol
- Casandra database module
- integration in other SIP Projects
- more powerful web interface
- timestamp in HEP protocol (version 2)
- ….

Thank you

URL: http://homer.googlecode.com/
E-mail/IM: alexandr.dubovikov@gmail.com