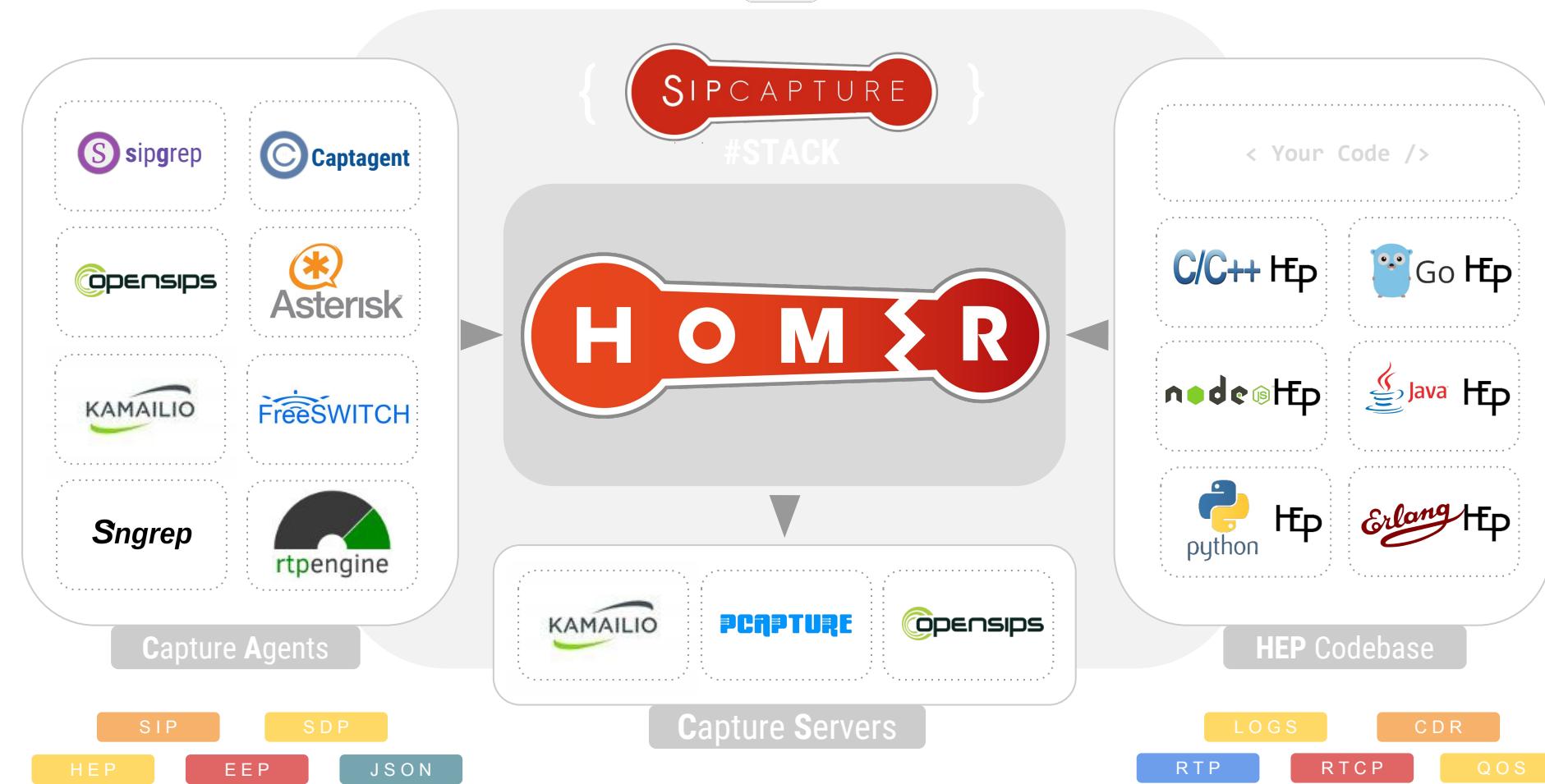


**H**

SIPCAPTURE





Senior Voice Engineer for QSC AG, one of the major German voice and data providers. Alexandr holds a diploma in physics of Odessa State University. He has 20 years of experience in telecommunication techniques and has contributed to many OpenSource projects like FeeSwitch, SER, Kamailio, SEMS, Asterisk, SIPP, Wireshark. Alexandr is the main developer of Homer SIP Capture project. He is also founder of IRC RusNet Network, one of the biggest national IRC networks in the world.

**QSC AG** is digitising the German SME sector. With decades of experience and expertise in the areas of Cloud, Consulting, Outsourcing, and Telecommunications, QSC accompanies its customers securely into the digital age. Today already, cloud-based procurement models offer increased speed, flexibility, and full service availability. The Company's TÜV and ISO-certified data centres in Germany and its nationwide All-IP network form the basis for maximum end-to-end quality and security. QSC's customers benefit from one-stop innovative products and services that are marketed both directly and via partners. For details visit: <http://qsc.de>

**QXIP BV {QuickSIP}** is an Amsterdam based R&D Company specializing in Open-Source and Commercial Voice Technology Development - Our flagship projects are **SIPCAPTURE HOMER** and **PCAPTURE** based on our mature and open encapsulation protocol **HEP/EEP** (*Extensible Encapsulation Protocol*) Our Open-Source solutions are deployed and trusted by thousands of businesses worldwide. Our Customers include large telephony and network operators, voice service carriers, voip service providers, cloud service providers, call center operators and voice equipment vendors. Our Capture Technologies are natively implemented in all major OSS voip platforms such as *Kamailio*, *OpenSIPS*, *FreeSWITCH*, *Asterisk*, *RTPEngine* and many tools such as *sipgrep*, *sngrep* and more. For full details please visit our website at <http://qxip.net>

# Things you already know about #HOMER

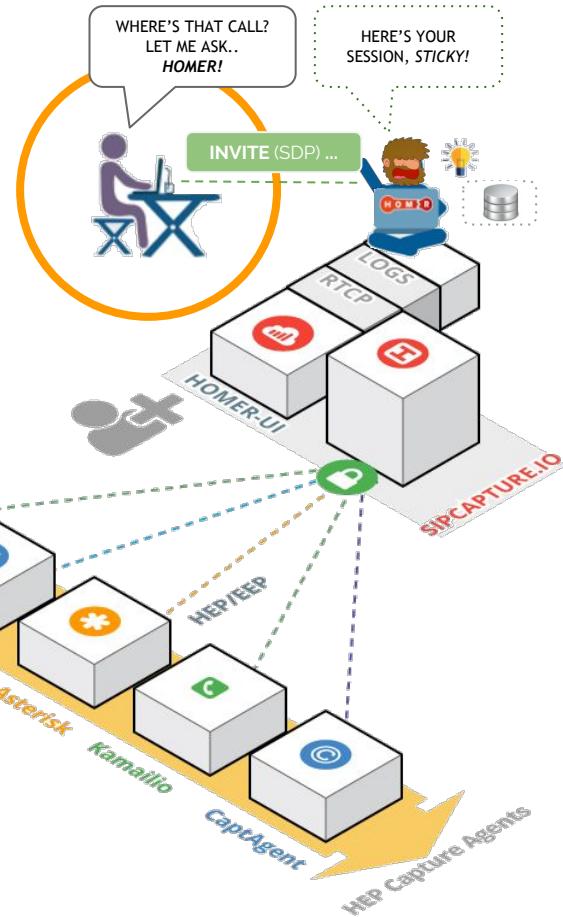
If you missed our Workshop, make sure you grab the PDF!

**HOMER** is part of the **SIPCAPTURE** stack, a robust, carrier-grade, scalable **RTC** and **VoIP** Capture and Monitoring application with built in support out of the box in many leading platforms, ready to process, index & store insane amounts of signaling, logs and statistics and providing instant search, end-to-end analysis and drill-down capabilities for ITSPs, VoIP Providers Trunk Suppliers as well as Enterprises and Developers using SIP signaling.

**HOMER** provides many features and advantages, including:

- Instant centralized access to present and past signaling & stats
- Full SIP/SDP payload retention with precise timestamping
- Automatic correlation of sessions, logs and reports
- Support for RTP and RTCP Media statistics and analytics
- Visual representation of multi protocol session call-flows
- Fast detection of usage and system anomalies
- System agnostic view of VoIP and RTC traffic flows
- Unlimited plug & play capture agents and HEP custom data feeds
- Multi-User and Customizable UI based on JS/Angular/D3
- PCAP Exporting and Sharing functionality with 3rd parties

... and much more!



FIND ALL ABOUT HOMER: <http://github.com/sipcapture/homer>

# What is HEP/EEP ?

HEP = Homer Encapsulation Protocol

EEP = Extensible Encapsulation Protocol

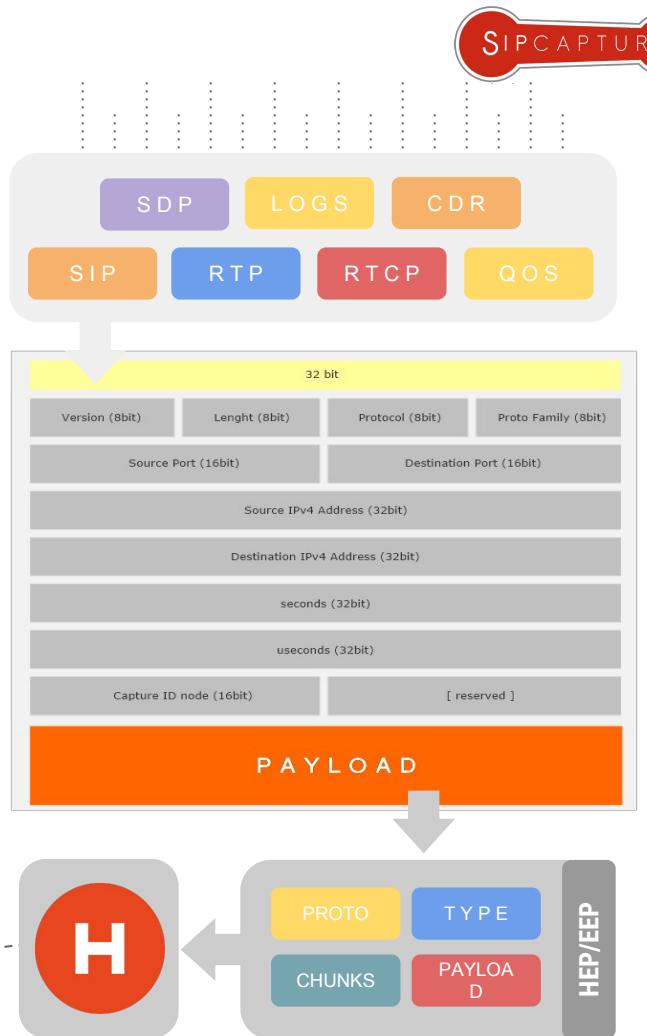
**HOMER's own** Encapsulation protocol (HEP/EEP) is used to wrap, argument and transfer captured packets between a capture Agent and capture Server.

The **HEP/EEP** Extensible Encapsulation protocol was designed to provide an efficient, modular and low--level framework to accurately duplicate passively obtained IP datagrams for remote collection over *UDP/TCP/SCTP* connections, where full retention of original datagram headers, timestamp, correlation pointers and original payload MUST be provided back to the collectors without any alterations or data loss.

The **HEP3/EEP** definition includes both generic (*internal*) and vendor- specific custom defined **chunk types** providing ground for implementors to extend the spectrum of the deliverable data within the HEP protocol alongside the encapsulated IP datagrams with custom data without protocol modifications.

**HOMER** currently supports HEP encoding/decoding for *SIP*, *XMPP*, *RTCP*, *RTCP-XR*, *HSP-MOS* and *Custom Logs* or *CDRs* in plain text or JSON format.

Find the full HEP/EEP specs at: <http://github.com/sipcapture/hep>



# HEP/EEP - Native Capture Agents

## Integrated Agents in OSS Platforms

**HEP** agents have been consistently integrated across leading OSS solutions - chances are you have one in your fleet *already!*

The following projects provide integrated HEP support:

- Kamailio
- OpenSIPS
- FreeSWITCH
- Asterisk / PJSIP
- RTP:Engine + RTC:Engine
- *sipgrep*
- *sngrep*

Examples are also provided for the following languages:

- C/C++
- Java
- Javascript / Node.JS
- Erlang
- Python
- Go
- Perl (*thanks @sipgate team*)



Find more on our wiki: [http://github.com/sipcapture/homer/wiki](https://github.com/sipcapture/homer/wiki)



### Kamailio Example:

<https://github.com/sipcapture/homer/wiki/Examples%3A-Kamailio>

### OpenSIPS Example:

<https://github.com/sipcapture/homer/wiki/Examples%3A-OpenSIPS>

### FreeSWITCH Example:

<https://github.com/sipcapture/homer/wiki/Examples%3A-FreeSwitch>

### CaptAgent Example:

<https://github.com/sipcapture/homer/wiki/Examples%3A-Captagent4>

### nProbe VoIP Example:

<https://github.com/sipcapture/homer/wiki/Examples%3A-nProbe>

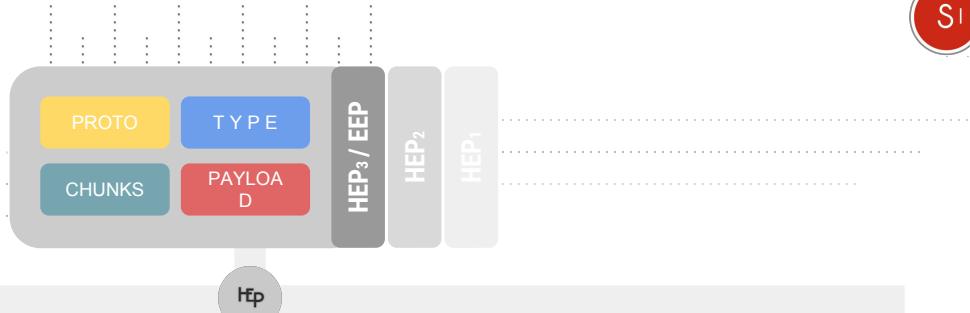
### ACME SBC Example:

<https://github.com/sipcapture/homer/wiki/Examples%3A-ACME-Packet>

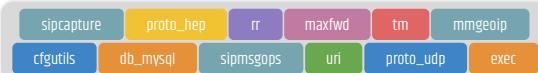


# Inside the CAPTURE SERVER

Nuts and Bolts behind the **HEP** Sockets



2.2



**sipcapture.opensips.cfg**



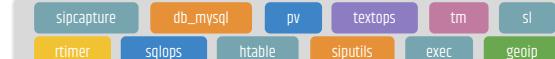
**HOMER 5** capture servers can be based on either **Kamailio 4.4+** or **OpenSIPS 2.2+** using the **SIPCAPTURE** module supporting **HEP / EEP** functionality in combination with any other available module to provide a programmable and modular **RTC** packet capture framework with no limitations and no presets, ready to extend and customize



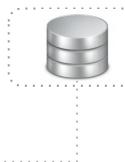
*Who's best? Only **YOU** decide!*



4.4



**sipcapture.kamailio.cfg**



# What's **NEXT** with **HOMER 5**

A brief look into the future development and roadmap

Recent Developments:

- ★ HEP3 support in FreeSWITCH 1.6.8
- ★ HEP3 support in Kamailio (siptrace)
- ★ HEP3 support in RTPEngine mr4.4.1
- ★ CaptAgent 6.1 with new modules
- ★ New protocols support in HOMER 5

# What's **NEXT** with **HOMER 5**

A brief look into the future development and roadmap

During our previous **workshop** a few questions and ideas from the audience were shared about **HOMER**:

- ★ HEP/EEP Event Socket
- ★ Elasticsearch support
- ★ Postgres SQL support

Now, *clearly* neither one of those features and/or integrations would be possible to achieve in a single day.

This would require a modular and extensible design upfront, an incredibly flexible set of tools and skills to perform everything on the road - not to mention much more time available and a comfortable desk.

# What's **NEXT** with **HOMER 5**

A brief look into the future development and roadmap

During our previous **workshop** a few questions and ideas from the audience were shared about **HOMER**:

- ★ HEP/EEP Event Socket
- ★ Elasticsearch support
- ★ Postgres SQL support

Now, *clearly* neither one of those features and/or integrations would be possible to achieve in a single day.

This would require a modular and extensible design upfront, an incredibly flexible set of tools and skills to perform everything on the road - not to mention much more time available and a comfortable desk.

Unless you are *like me.....*

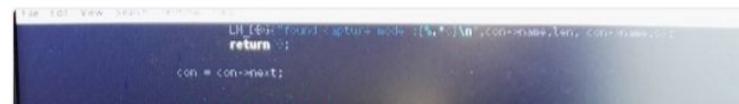


Alexandr Dubovikov  
@adubovikov



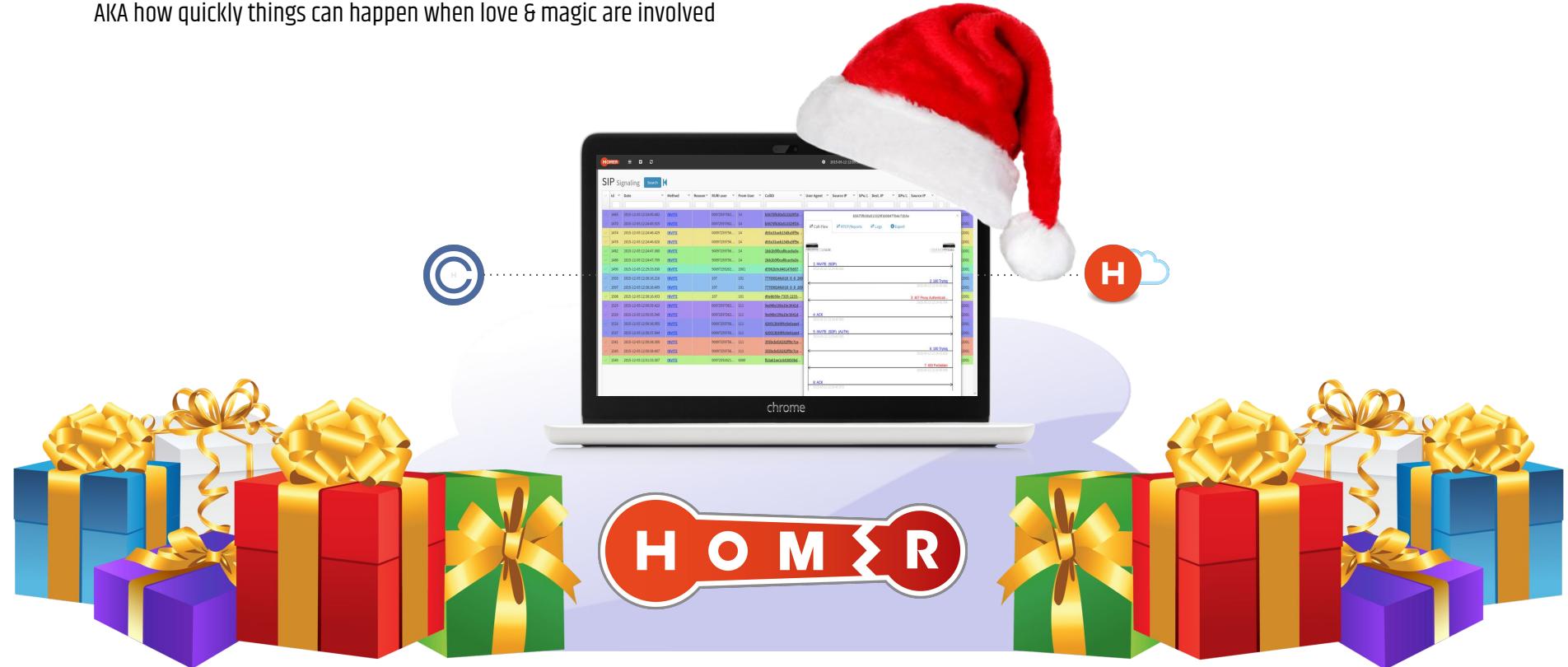
Following

Compiling kamailio and the new **@sipcapture** module at 300 km/h. On the way to **@KamailioWorld** with DB ICE



# Mele Kalikimaka with HOMER 5

AKA how quickly things can happen when love & magic are involved



# Mele Kalikimaka with **HOMER 5**

AKA how quickly things can happen when love & magic are involved



**WARNING: Beer consumption might have affected the rest of the presentation @2AM**



# What's **NEXT** with **HOMER 5**

A brief look into the future development and roadmap

## History of HEP/EEP Support in Kamailio:

As the HEP encapsulation protocol evolved, so did the features of sipcapture module and Kamailio:

Generation 1: No handling, all HEP messages directly sent to default DB table  
*Kamailio 3.x*

Generation 2: No handling, HEP messages can be sent to arbitrary DB table from config  
*Kamailio 4.x*

Generation 3: Full handling, messages can be manipulated and sent according to any parameter  
*Kamailio 5.x*

# What's **NEXT** with **HOMER 5**

A brief look into the future development and roadmap



## Kamailio 5.x: HEP/EEP Event Socket

What could make HEP/EEP even more useful? An Event Socket of course! Some awesome things you can do:

- Parse, Process and Extract HEP/EEP header values and types
- Handle and Process non-SIP Messages (no parsing) directly and dynamically from the config script
- Generate custom statistics and Metrics based on the Capture Agent ID, Source, etc
- Forge and handle your very own HEP/EEP type in seconds for development and testing

Some of our common Types:

0x01	SIP
0x02	XMPP
0x03	SDP
0x05	RTCP (json serialized)
0x06	MGCP
0X29	WSS (webRTC)
0x64	LOGS (text or JSON)

# KAMAILIO EVENT SOCKET

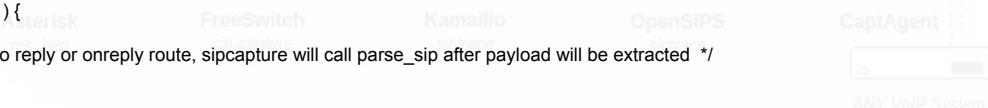
Example Usage of the Integrated Capture Agent (sipcapture module)

```
event_route[sipcapture:request] {
    xlog("received sipcapture request from $si:$sp\r\n");
    xlog("non-sip packet received - content [[$mb]] from [$si:$sp]\r\n");
    /* get proto type id from HEP header */
    hep_get("0x0B", "0x0000", "$var(data)");
    hep_get("0x0F", "0x0000", "$var(payload)");
    hep_get("0x11", "0x0000", "$var(correlation_id)");
    hep_get("0x0C", "0x0000", "$var(capture_id)");

    /* Statistics for capture id */
    if($ht(a=>captureagent::$var(capture_id)) == $null) $ht(a=>captureagent::$var(capture_id)) = 0;
    $ht(a=>captureagent::$var(capture_id)) = $ht(a=>captureagent::$var(capture_id)) + 1;
    /* you can make statistics here */

    $var(proto) = ${var(data){s.int}};

    if($var(proto) == 100) {
        /* send this data to logs_capture */
        report_capture("logs_capture", "$var(callid)");
        //dont send data to route/onreply_route
        drop;
    }
    else if($var(proto) == 50) {
        /* send this data to RTCP */
        report_capture("rtcp_capture", "$var(callid)");
        //dont send data to route/onreply_route
        drop;
    }
    else if($var(proto) == 1) {
        /* SIP */
        /* send this data to reply or onreply route, sipcapture will call parse_sip after payload will be extracted */
        return;
    }
}
```



# What's **NEXT** with **HOMER 5**

A brief look into the future development and roadmap



## Elasticsearch (deeper) Integration

**HOMER 5** natively supports querying *Elasticsearch* data from *Dashboard* widgets/charts, but not for session data... until now! Thanks to the power of our **HEP Event Socket** you can now ship selected packets to your powerful Elasticsearch Cluster or any other JSON store. Here's a fully working example using **sipcapture + jansson + http\_client\_async** modules

```
event_route[sipcapture:request] {
    xlog("received sipcapture request from $si:$sp\n");
    /* get proto type id from HEP header */
    hep_get("0x0B", "0x0000", "$var(data)");
    hep_get("0x0F", "0x0000", "$var(payload)");
    hep_get("0x11", "0x0000", "$var(correlation_id)");

    $var(proto) = $(var(data){s.int});

    if($var(proto) == 100) {
        # create a transaction to be paused, and resumed in route[HTTP_REPLY]
        t_newtran();
        #Create json object for ElasticSearch
        jansson_set("string", "body", "$var(payload)", "$var(json)");
        jansson_set("string", "postDate", "2016-05-20", "$var(json)");
        jansson_set("string", "title", "Data from Homer", "$var(json)");
        #Send to ES
        http_async_query("http://elasticsearch:9200/homer/logs", "$var(json)", "HTTP_REPLY");
        drop;
    }
}
```

```
else if($var(proto) == 1) {
    /* SIP */
    # create a transaction to be paused, and resumed in route[HTTP_REPLY]
    t_newtran();
    #Create json object for ElasticSearch
    jansson_set("string", "body", "$var(payload)", "$var(json)");
    jansson_set("string", "postDate", "2016-05-20", "$var(json)");
    jansson_set("string", "title", "SIP from Homer", "$var(json)");
    #Send to ES
    http_async_query("http://elasticsearch:9200/homer/sip", "$var(json)", "HTTP_REPLY");
    drop;
}

drop;
}

route[HTTP_REPLY] {
    if ($http_ok) {
        xlog("L_INFO", "route[HTTP_REPLY]: status $http_rs\n");
        xlog("L_INFO", "route[HTTP_REPLY]: body $http_rb\n");
    } else {
        xlog("L_INFO", "route[HTTP_REPLY]: error $http_err\n");
    }
}
```



# What's **NEXT** with **HOMER 5**

A brief look into the future development and roadmap

Here's a quick example we created around 2AM last night - make it your own and start pairing data in your **SIPCAPTURE** stack to **Elasticsearch**!

*Have a good Idea or good business case?*  
Send us a mail and help us shape the future of this feature to be as flexible as we can make it! ;)

#opensource

The screenshot shows the HOMER 5 application interface. On the left, there is a table of SIP signaling logs with columns for Id, Date, and Method. The logs include entries for REGISTER, 401, and 403 status codes. In the center, there is a detailed view of a specific log entry with tabs for Call-Flow, QoS Reports, Remote Logs, Export, and Session. A large arrow points from this detailed view down to an Elasticsearch result. The Elasticsearch result is displayed in a JSON format, showing a single document with fields like \_index, \_type, \_id, \_score, \_source, @timestamp, ts, value, string, and group. The \_source field contains the raw log data.

	Id	Date	Method
✓	23250	2016-05-20 00:21:44.023 +0000	REGISTER
✓	23249	2016-05-20 00:21:44.023 +0000	401
✓	23251	2016-05-20 00:21:44.079 +0000	REGISTER
✓	23248	2016-05-20 00:21:44.080 +0000	403
✓	23253	2016-05-20 00:21:56.318 +0000	REGISTER
✓	23252	2016-05-20 00:21:56.319 +0000	401
✓	23255	2016-05-20 00:21:56.434 +0000	REGISTER

Call-ID: 16b50d4f-5f64-4573-e3c7-d109-69-67-241-5080

Session duration: 00:00:00

Elasticsearch Result from :

```
{
  "_index": "homer",
  "_type": "logs",
  "_id": "AVTLetqotasK96kVfXPa",
  "_score": 1,
  "_source": {
    "@timestamp": "2016-05-20T01:00+0000",
    "ts": 1463700600000,
    "value": 1,
    "string": "Hello Elasticsearch, this is HOMER speaking!",
    "group": "callid"
  }
}
```

# What's **NEXT** with **HOMER 5**

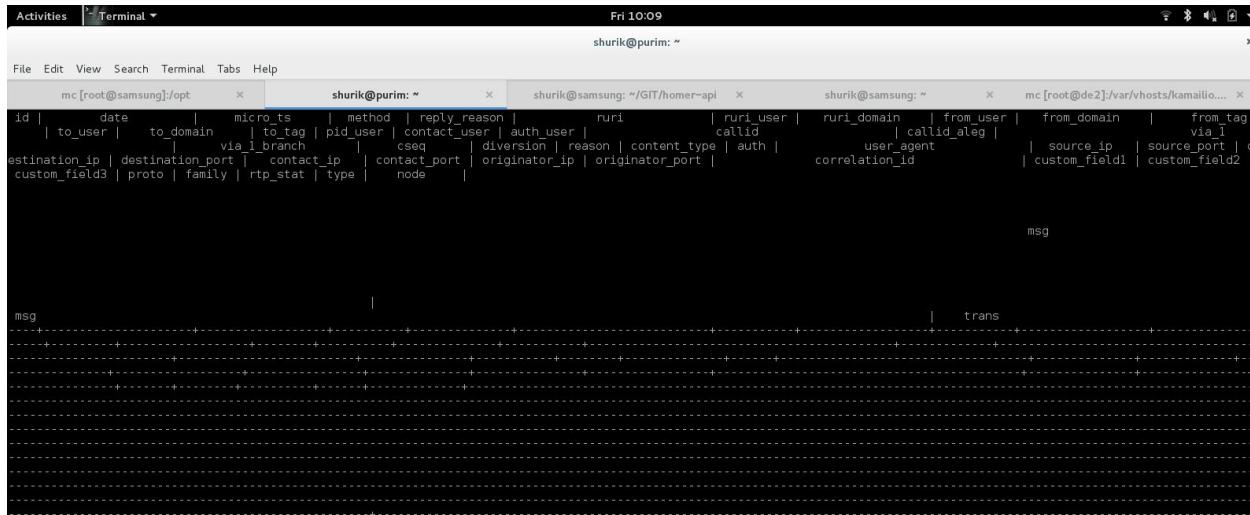
A brief look into the future development and roadmap



## Postgres SQL

**HOMER 5** natively ships with an “agnostic” database module virtually supporting multiple database backends, but in reality only the **MySQL** connector was fully developed.... Until last night!

While you were enjoying the drinks kindly offered by the conference, we were busy making this happen!  
(Thanks William King for the onsite assistance fine tuning!)



The screenshot shows a terminal window with five tabs open, all displaying PostgreSQL tables. The tabs are:

- mc [root@samsung]:/opt
- shurik@purim: ~
- shurik@samsung: ~/GIT/homer-api
- shurik@samsung: ~
- mc [root@de2]:/var/vhosts/kamailio...

The first tab (mc) displays a table with columns: id, date, micro\_ts, method, reply\_reason, ruri, ruri\_user, ruri\_domain, from\_user, from\_domain, from\_tag, to\_user, to\_domain, to\_tag, pid\_user, contact\_user, auth\_user, callid, callid\_aleg, via, via\_branch, cseq, diversion, reason, content\_type, auth, user\_agent, source\_ip, source\_port, destination\_ip, destination\_port, contact\_ip, contact\_port, originator\_ip, originator\_port, correlation\_id, custom\_field1, custom\_field2, custom\_field3, proto, family, rtp\_stat, type, node.

The second tab (shurik@purim) shows a single row with msg.

The third tab (shurik@samsung) shows a single row with trans.

The fourth tab (shurik@samsung) shows a single row with msg.

The fifth tab (mc) shows a single row with trans.



# What's **NEXT** with **HOMER** 5

## A brief look into the future development and roadmap

# Postgres SQL

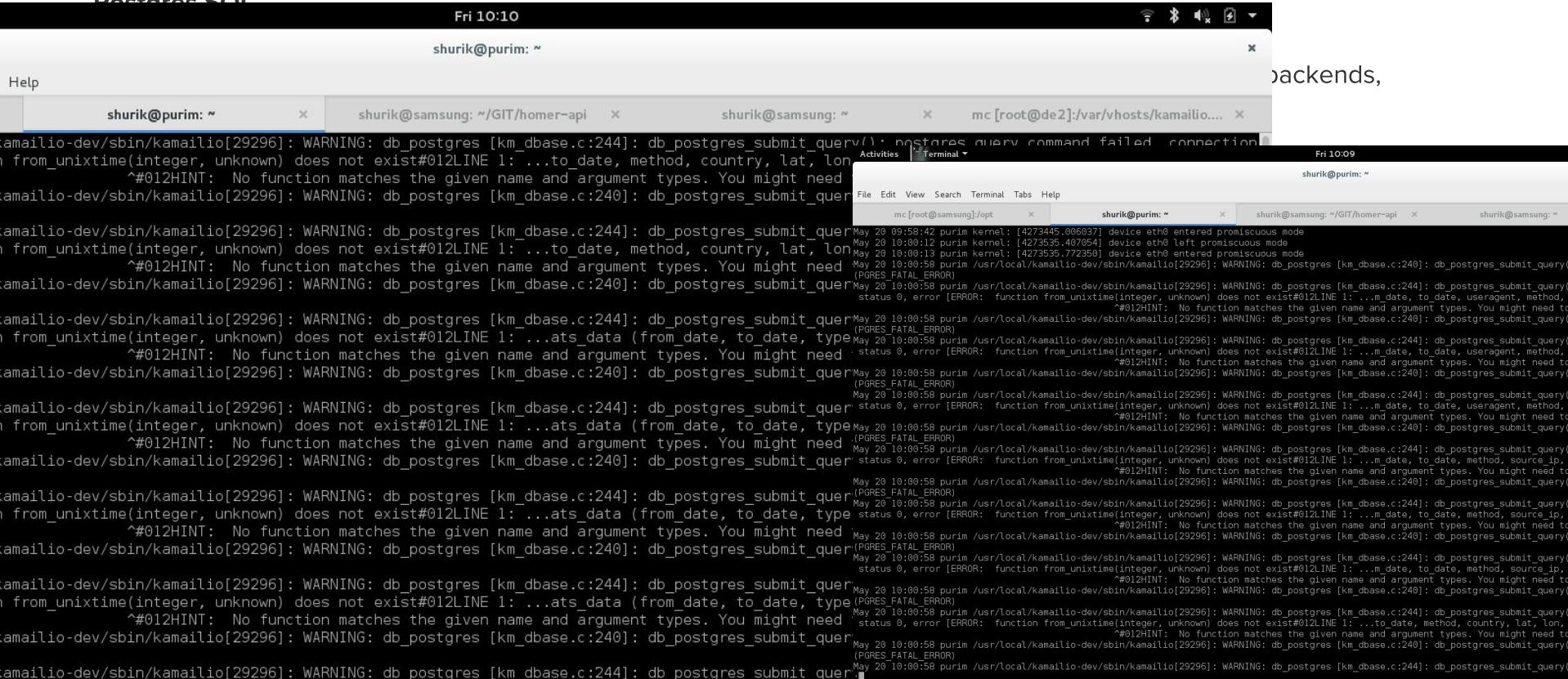
**HOMER 5** natively ships with an “agnostic” database module virtually supporting multiple database backends, but in reality only the **MySQL** connector was fully developed.... Until last night!

While you were enjoying the drinks kindly offered by the conference  
(Thanks William King for the onsite assistance fine tuning!)



# What's **NEXT** with **HOMER 5**

## A brief look into the future development and roadmap



## Activities

## Terminal

Fri 10:11

shurik@purim: ~

File Edit View Search Terminal Tabs Help

shurik@purim:

shurik@samsung: ~/GIT/homer-api >

urik@samsung: ~

mc [root@de2]:/var/vhosts/kamailio.... ×



# What's **NEXT** with **HOMER 5**

A brief look into the future development and roadmap

## Postgres SQL

Activities Terminal ▾ Thu 20:50

HC Ind Go Up Git Int Do Ho hoi Tal hoi G po Sc sql PG G po Po kai hti SQ hoi po G po H x Ho N Po Alexander

dev.sipcapture.org/#/result

**HOMER** + Q Panels Last 6 Hours C

SIP Signaling Search K Regex Filter ...

Id	Date	Method	Reason	RURI user	From User	To User	CallID	CallID_AL	User Agent	Source Host	SPort	Destination Ho...	DPort	Prox.	Node
2	2016-05-19 20:39:19.141 +...	<u>REGISTER</u>			104	104	ff4df6e1-cf634572-2328		PolycomVvx-V	92.204.1.151	5060	109.69.67.241	5060	udp	homer01:301
1	2016-05-19 20:39:19.143 +...	<u>200</u>	OK												
4	2016-05-19 20:39:30.649 +...	<u>REGISTER</u>													
3	2016-05-19 20:39:30.650 +...	<u>401</u>	Una												
6	2016-05-19 20:39:30.764 +...	<u>REGISTER</u>													
5	2016-05-19 20:39:30.765 +...	<u>403</u>	Forb												
9	2016-05-19 20:39:44.139 +...	<u>REGISTER</u>													
8	2016-05-19 20:39:44.140 +...	<u>401</u>	Una												
10	2016-05-19 20:39:44.188 +...	<u>REGISTER</u>													
7	2016-05-19 20:39:44.189 +...	<u>403</u>	Forb												
2	2016-05-19 20:39:56.885 +...	<u>OPTIONS</u>													
1	2016-05-19 20:39:56.885 +...	<u>OPTIONS</u>													
3	2016-05-19 20:39:56.929 +...	<u>200</u>	OK												

Call-ID: ff4df6e1-cf634572-23287b87@192.168.178.32

shurik@purim: ~

File Edit View Search Terminal Tabs Help

```
mc [root... × shurik@sa... × shurik@sa... × shurik@p... × shurik@p...
```

MSG ID: 2

Message Details

92.204.1.151:5060

2016-05-19 20:39:19 +0200 : 92.204.1.151:5060

REGISTER sip:sip2.botauro.com:5060 SIP/2.0

Via: SIP/2.0/UDP 192.168.178.32;branch=z9hG4bK

From: "104" <sip:104@sip2.botauro.com>;tag=80

To: <sip:104@sip2.botauro.com>

CSeq: 3053 REGISTER

Call-ID: ff4df6e1-cf634572-23287b87@192.168.178.32;method=REGISTER

Contact: <sip:104@192.168.178.32>;methods="INFO,MESSAGE,SUBSCRIBE,NOTIFY,PRACK,UPDATE"

Accept-Language: en

Authorization: Digest username="104",realm="sip2.botauro.com",uri="sip:sip2.botauro.com:5060",response="937143be-8718-f2ada1c40273",qop=auth,cnonce="W",o=

# What's **NEXT** with **HOMER 5**

A brief look into the future development and roadmap



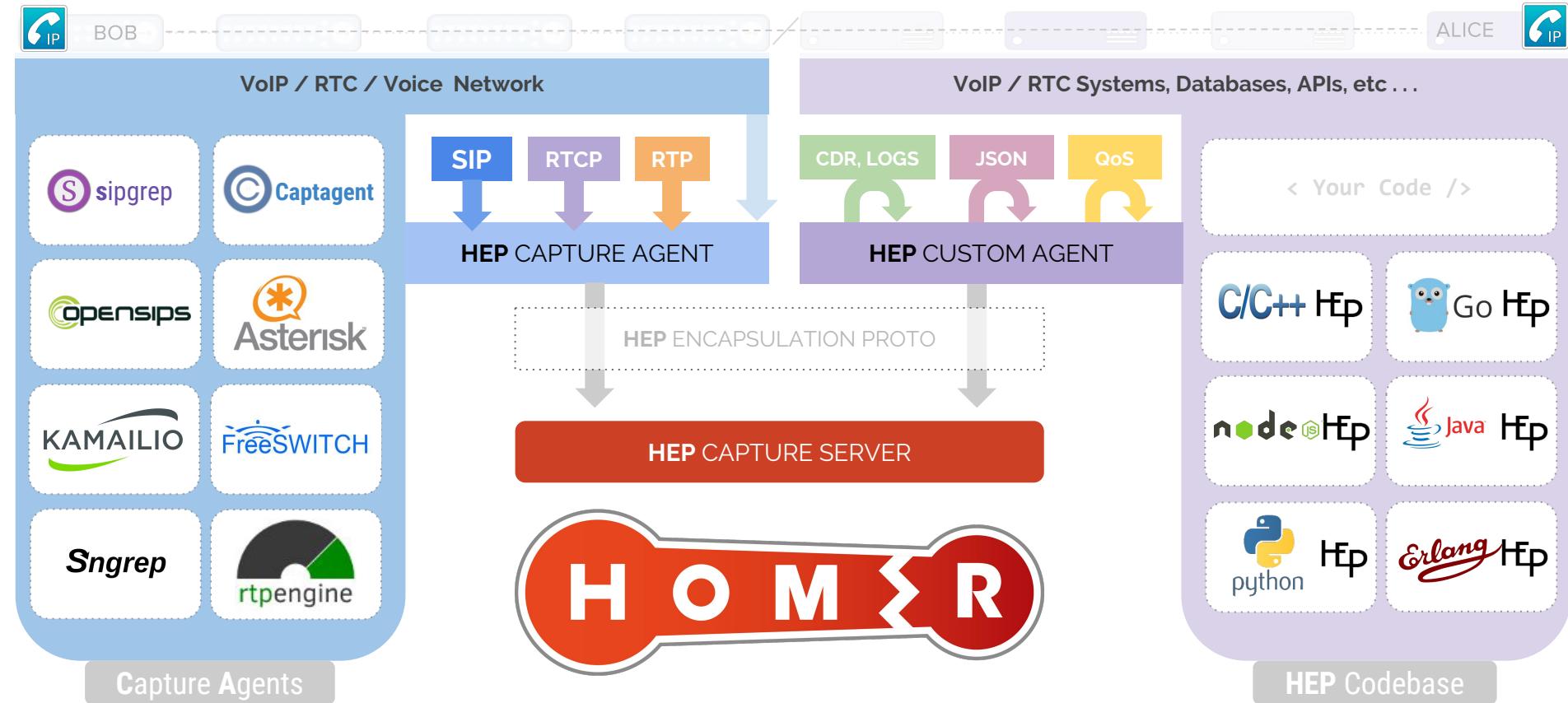
## HOMER ALARMS HOW-TO

Members of the audience asked us to add a few examples for Alarms creation and management - in a nutshell:

- 1) Create alarms in kamailio.cfg
- 2) Create notification block in kamailio.cfg
- 3) Create UI trigger in H5 user-interface admin

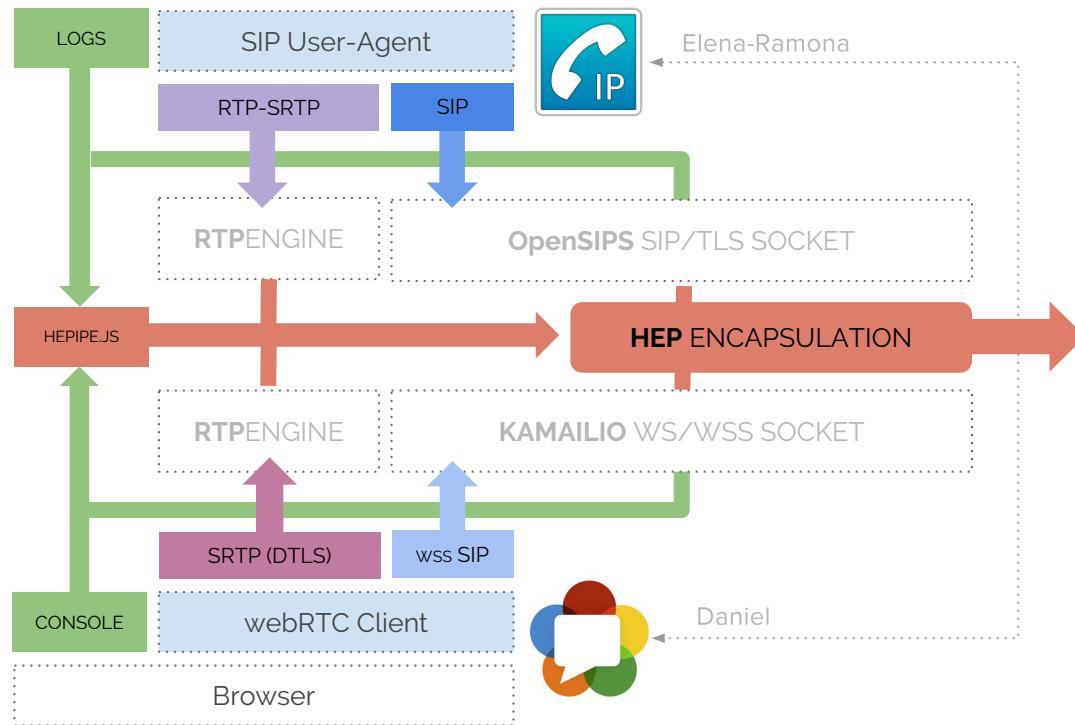
```
if($ua =~ "(friendly-scanner|sipvicious|sipcli)") {  
    $var(atype) = 'scanner';  
    sql_query("cb", "INSERT INTO alarm_data_mem (create_date, type, total, source_ip, description) VALUES(NOW(), '$var(atype)', 1, '$si', 'Friendly  
scanner alarm!') ON DUPLICATE KEY UPDATE total=total+1");  
    route(KILL_VICIOUS);  
}  
#Alarm for Scanner;  
    if($var(atype) == "scanner") {  
        sql_query("cb", "DELETE FROM alarm_data_mem WHERE type='scanner' AND total < $var(alue)");  
        if($var(anotify) == 1) {  
            sql_query("cb", "SELECT * FROM alarm_data_mem WHERE type='scanner' AND total >= $var(alue) LIMIT 2", "rd");  
            if($dbr(rd>rows) > 0) {  
                route(SEND_ALARM);  
            } sql_result_free("rd");  
        }  
    }  
  
route[SEND_ALARM] {  
    exec_msg('echo "Value: $var(thvalue), Type: $var(atype), Desc: $var(ename)" | mail -s "HOMER ALERT $var(atype) - $var(thvalue)" $var(aemail)');  
}
```

# SIPCAPTURE HEP/EEP Stack



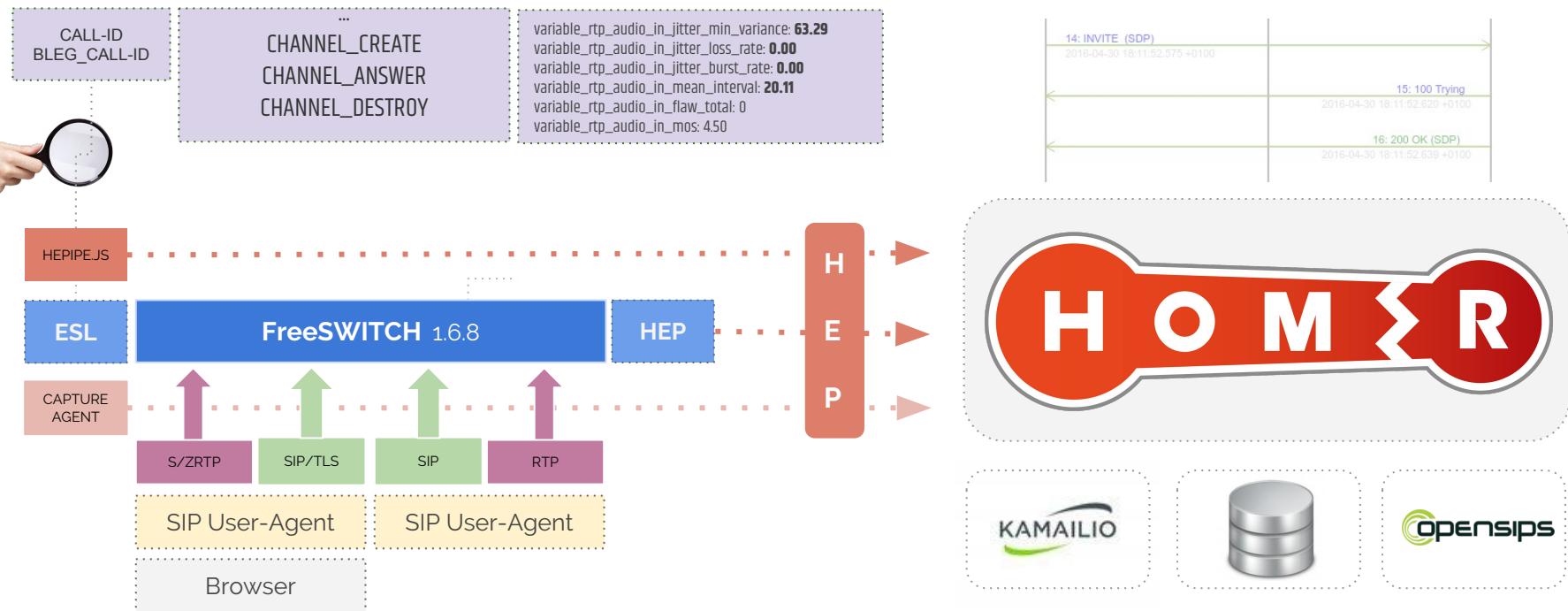
# Kamailio/SER Monitoring

Example Illustration SIP to SIP/WSS via SER Proxies and RTP Relays



# FreeSWITCH Monitoring

Example Illustration of Multi-Level Monitoring in FS HEP + ESL



# FreeSWITCH HEP/EEP Configuration

Example Usage of the Integrated Capture Agent for Monitoring



**FreeSWITCH** ships with a built-in HEP agent used to mirror/transfer packets unmodified and carries timestamp and several session key values in its headers, designed for capturing simple and complex scenarios with minimal configuration efforts.

To enable **HEP** capturing, open *sofia.conf.xml* and set capture-server param:

```
<param name="capture-server" value="udp:10.0.0.1:9060" />
```

**NEW!** Freeswitch v1.6.8 (*master git*) now supports **HEPv2 + HEPv3/EEP** encapsulation & parameters:

```
<param name="capture-server" value="udp:10.0.0.1:9060;hep=3;capture_id=100" />
```

To enable the **HEP** capture agent globally, open *internal.xml* and change sip-capture param to "yes"

```
<param name="sip-capture" value="yes" />
```

To enable/disable the **HEP** capture agent on demand, you can use CLI commands:

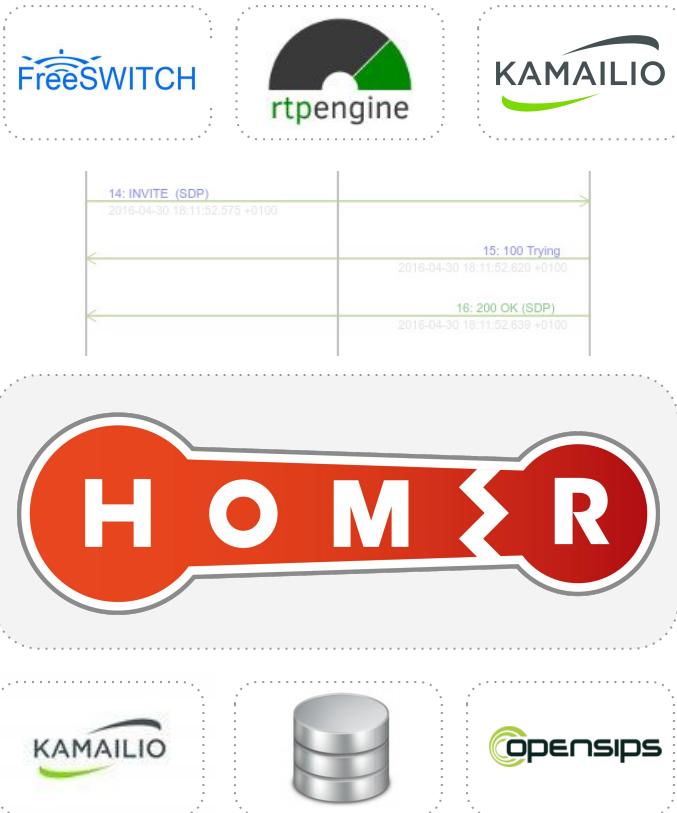
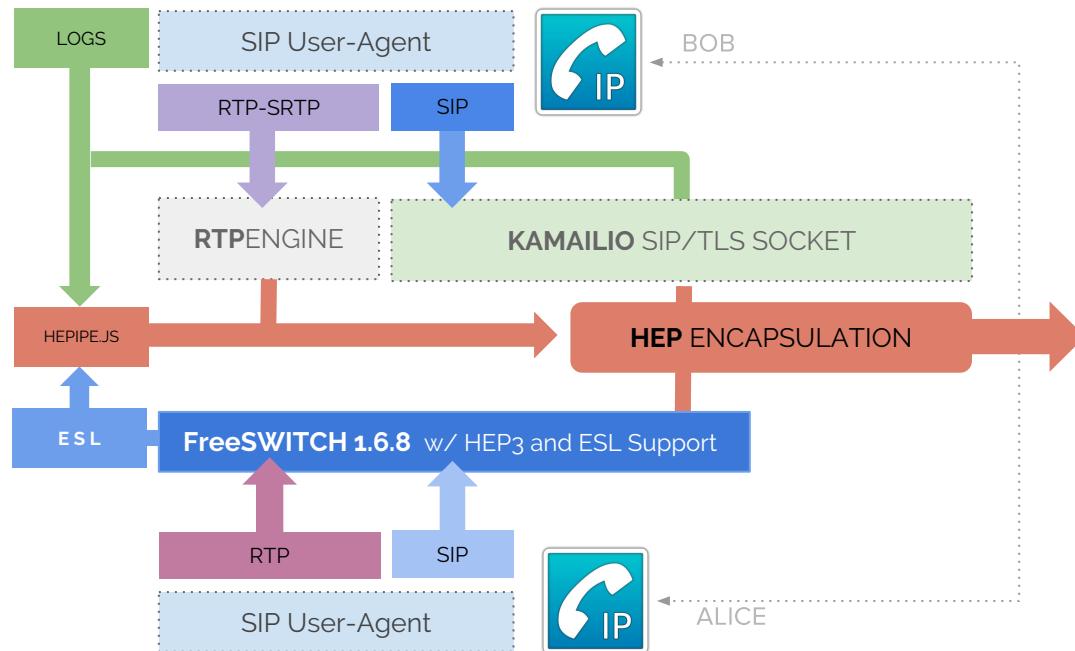
```
freeswitch@fsnode04> sofia global capture on
+OK Global capture on
freeswitch@fsnode04> sofia global capture off
+OK Global capture off
```

To enable/disable the **HEP** capture agent on a specific profile:

```
freeswitch@fsnode04> sofia profile internal capture on
```

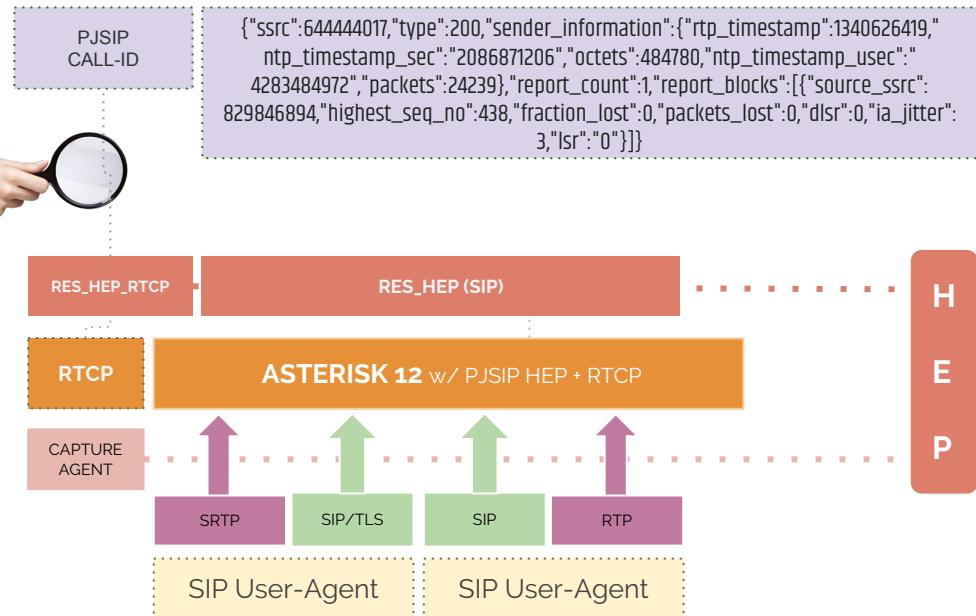
# FreeSWITCH + Kamailio Monitoring

Example Illustration SIP + RTCP via Load Balancer w/ Correlation



# Asterisk Monitoring

Example Illustration of Multi-Level Monitoring in Asterisk 12+



C/C++ HEP

14: INVITE (SDP)

2016-04-30 18:11:52.575 +0100

15: 100 Trying

2016-04-30 18:11:52.620 +0100

16: 200 OK (SDP)

2016-04-30 18:11:52.839 +0100



# Kamailio WSS Monitoring

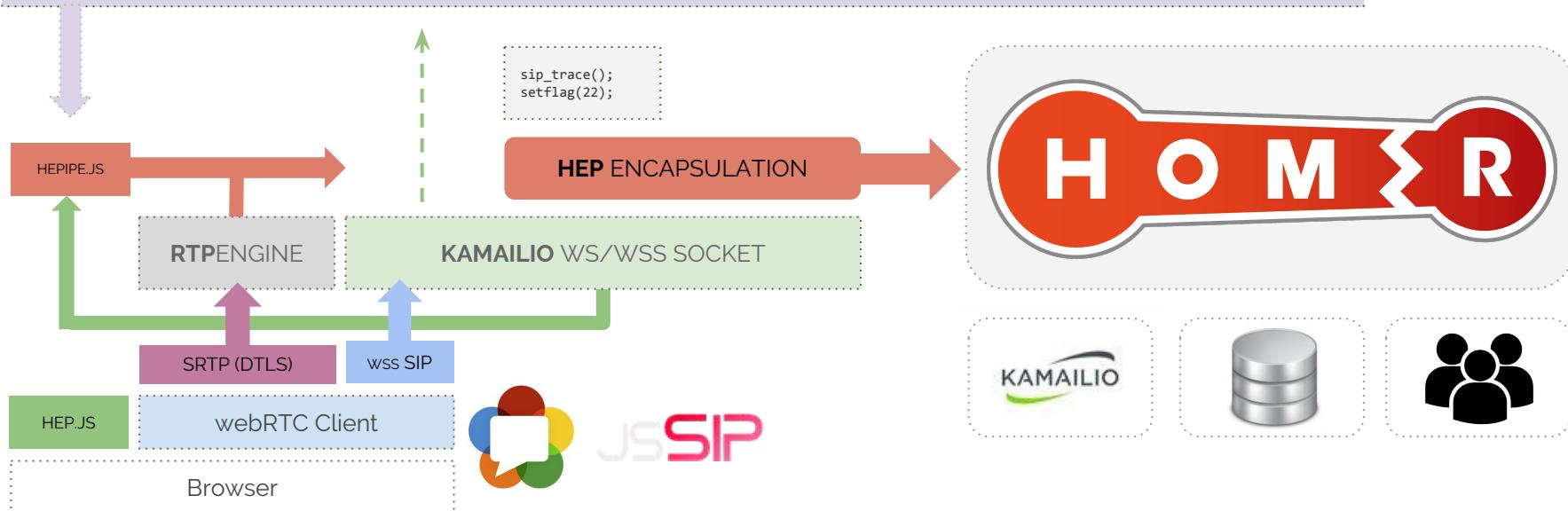
<http://github.com/sipcapture/wiki>



```
if (proto == WS || proto == WSS) { setflag(SRC_WS);

    xlog("L_INFO", "homerwss CID: [$ci], SIP: Method: $rm, CSEQ: $cs, RU: $rU, WSS Request: RM: $var(wss_rm), RU: $var(wss_ru),
          UAC: $var(wss_uac), Connection: $var(wss_connection), Upgrade: $var(wss_upgrade), Origin: $var(wss_origin),
          Host: $var(wss_host), Sec_Proto: $var(wss_sec_proto), Sec-Key: $var(wss_sec_key), WS_VERSION: $var(wss_sec_version)");

}
```



# HOMER 5: WSS Call Flow

WSS to SIP Call Troubleshooting



	SIP Signaling		Search																	
✓	ID	Date	Method	Reason	RURI user	From User	To User	CallID	CallID_AL	User Age...	Source H...	SPort	Destinati...	DPort	Pr...	Node	☰			
✓	304	2016-01-26 00:16:15.671	<a href="#">200</a>	OK		201	101	<a href="#">4dar0hqpfkfm8o5q0b4q</a>	<a href="#">4dar0hqpfkfm8o5q0b4q</a>		<a href="#">WSS1000</a>	52645	<a href="#">WSS1000</a>	4443	3	homer01...	▲			
✓	306	2016-01-26 00:16:15.672	<a href="#">200</a>	OK		201	101	<a href="#">4dar0hqpfkfm8o5q0b4q</a>			<a href="#">WSS1000</a>	4443	<a href="#">WSS1000</a>	47682	3	homer01...	▲			
✓	307	2016-01-26 00:16:15.718	<a href="#">ACK</a>		lq1pna1u	201	101	<a href="#">4dar0hqpfkfm8o5q0b4q</a>		JsSIP 0.7....	<a href="#">WSS1000</a>	47682	<a href="#">WSS1000</a>	4443	3	homer01...	▲			
✓	308	2016-01-26 00:16:22.192	<a href="#">BYE</a>		lq1pna1u	201	101	<a href="#">4dar0hqpfkfm8o5q0b4q</a>		JsSIP 0.7....	<a href="#">WSS1000</a>	47682	<a href="#">WSS1000</a>	4443	3	homer01...	▲			
✓	309	2016-01-26 00:16:22.192	<a href="#">BYE</a>		lq1pna1u	201	101	<a href="#">4dar0hqpfkfm8o5q0b4q</a>		JsSIP 0.7....	<a href="#">WSS1000</a>	5060	<a href="#">WSS1000</a>	52645	3	homer01...	▲			
✓	310	2016-01-26 00:16:22.258	<a href="#">200</a>	OK		201	101	<a href="#">4dar0hqpfkfm8o5q0b4q</a>			<a href="#">WSS1000</a>	52645	<a href="#">WSS1000</a>	4443	3	homer01...	▲			
✓	311	2016-01-26 00:16:22.259	<a href="#">200</a>	OK		201	101	<a href="#">4dar0hqpfkfm8o5q0b4q</a>			<a href="#">WSS1000</a>	4443	<a href="#">WSS1000</a>	47682	3	homer01...	▲			

Call-ID: [4dar0hqpfkfm8o5q0b4q](#)

Call-Flow
QoS Reports
Logs
Export
Session Duration: 00:00:15.495

1: INVITE (SDP)  
2016-01-26 00:16:07.764

2: 100 trying -- your cal...  
2016-01-26 00:16:07.764

3: INVITE (SDP)  
2016-01-26 00:16:07.765

4: 100 Trying  
2016-01-26 00:16:07.817

Call-ID: [4dar0hqpfkfm8o5q0b4q](#)

Call-Flow
QoS Reports
Logs
Export
Session Duration: 00:00:15.495

Filter Logs

```
Jan 26 00:16:07 de2 /usr/local/kamailio-dev/sbin/kamailio[30724]: INFO: <script>: homerwss CID: [4dar0hqpfkfm8o5q0b4q], SIP: Method: INVITE, CSEQ: 2592, RU: 101, WSS Request: RM: GET, RU: GET, UAC Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.82 Safari/...
```

```
Jan 26 00:16:15 de2 /usr/local/kamailio-dev/sbin/kamailio[30723]: INFO: <script>: homerwss CID: [4dar0hqpfkfm8o5q0b4q], SIP: Method: ACK, CSEQ: 2592, RU: lq1pna1u, WSS Request: RM: GET, RU: GET, UAC Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.106 Safari/537.36, Connection: Upgrade, Upgrade: websocket, Origin: https://www.qxip.net, Host: 127.0.0.1:4443, Sec_Proto: sip, Sec_KgmBBo/ekG74S2v3wEBQ==, WS_VERSION: 13
```

# UA Remote Log Monitoring

<http://github.com/sipcapture/hepipe-js>



```
JsSIP:Transport WebSocket disconnected (code: 1006) +2m
jssip.js:22725 JsSIP:ERROR:Transport WebSocket abrupt disconnection +0ms
jssip.js:22550 JsSIP:Transport trying to reconnect to WebSocket wss://1.2.3.4:4443 jssip.js:
22550 JsSIP:Transport connecting to WebSocket wss://1.2.3.4:4443 +4s
jssip.js:22550 JsSIP:Transport WebSocket wss://1.2.3.4:4443 connected +132 ms
```



HEPIPEJS

HEP ENCAPSULATION

RTPENGINE

WS/WSS SOCKET

SRTP (DTLS)

wss SIP

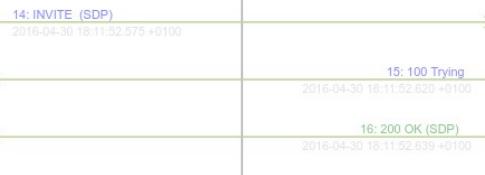
HEP.JS

D

webRTC Client

JsSIP

Browser



# CAPTAGENT 6.1

## Universal Modular Capture Agent w/ HEP3 Support

**Captagent** is a powerful, flexible, completely modular capture agent framework ready for virtually any kind of protocol and encapsulation method - past, present *and future* ;)

Currently available modules:

- ★ **SOCKET** Modules
  - Socket PCAP, Socket RAW, Socket RTCP-XR, Socket PF\_RING
- ★ **PROTOCOL** Modules
  - SIP, RTCP and other signaling and controlling Protocols
- ★ **API** Module
  - HTTP JSON REST API for Control and Statistics
- ★ **TRANSPORT** Modules
  - HEP Encapsulation output (v1/2/3)
  - JSON Serialization output
- ★ **ENCRYPTION** Modules
  - Encryption and Compression Module for HEP3
  - TLS Decryption pipeline for supported key exchange methods
- ★ **DATABASE** Modules
  - HASH Table
  - Redis interface
  - MySQL interface

CAPTAGENT: <https://github.com/sipcapture/captagent>

```
<!-- CORE MODULES -->
<module name="transport_hep" description="HEP Protocol" serial="2014010402">
  <profile name="hepsocket" description="Transport HEP" enable="true"
  serial="2014010402">
    <settings>
      <param name="version" value="3"/>
      <param name="capture-host" value="127.0.0.1"/>
      <param name="capture-port" value="9061"/>
      <param name="capture-proto" value="udp"/>
      <param name="capture-id" value="2001"/>
      <param name="capture-password" value="myhep"/>
      <param name="payload-compression" value="false"/>
    </settings>
  </profile>
</module>

<!-- PROTOCOLS -->
<module name="socket_pcaps" description="HEP Socket" serial="2014010402">
  <profile name="socketpcap_sip" description="HEP Socket" enable="true"
  serial="2014010402">
    <settings>
      <param name="dev" value="eth0"/>
      <param name="promisc" value="true"/>
      <param name="reasm" value="false"/>
      <param name="capture-plan" value="sip_capture_plan.cfg"/>
      <param name="filter">
        <value>portrange 5060-5091</value>
      </param>
    </settings>
  </profile>
</module>
```

captagent.xml

# CAPAGENT 6.1

(continued)

```

<module name="socket_pcap" description="HEP Socket" serial="2014010402">
  <profile name="socketspcap_sip" description="HEP Socket" enable="true"
  serial="2014010402">
    <settings>
      <param name="dev" value="any"/>
      <param name="promisc" value="true"/>
      <param name="reasm" value="false"/>
      <param name="tcpdefrag" value="false"/>
      <param name="capture-plan" value="sip_capture_plan.cfg"/>
      <param name="filter">
        <value>portrange 5060-5091</value>
      </param>
    </settings>
  </profile>

  <profile name="socketspcap_rtcp" description="RTCP Socket" enable="true"
  serial="2014010402">
    <settings>
      <param name="dev" value="any"/>
      <param name="promisc" value="true"/>
      <param name="reasm" value="false"/>
      <param name="capture-plan" value="rtcp_capture_plan.cfg"/>
      <param name="filter">
        <value>portrange 30000-50000</value>
      </param>
    </settings>
  </profile>
</module>

```

socket\_pcap.xml

Full examples: <https://github.com/sipcapture/captagent>

## Cascading Capture Plan example for SIP

```

#sip_capture_plan.cfg
capture[pcap] {

  # here we can check source/destination IP/port, message size
  if(msg_check("size", "100")) {

    #Do parsing
    while(parse_sip()) {

      /* packet processing pipeline */
      clog("NOTICE", "parsing SIP message ");

      if(source_ip("10.0.0.1")) {
        # Multiple profiles can be defined in transport_hep.xml
        if(!send_hep("hepsocket_homer01")) {
          clog("ERROR", "Error sending HEP!!!!");
        }
      } else {
        # Multiple profiles can be defined in transport_hep.xml
        if(!send_hep("hepsocket_homer02")) {
          clog("ERROR", "Error sending HEP!!!!");
        }
      }
    }
  }
  drop;
}

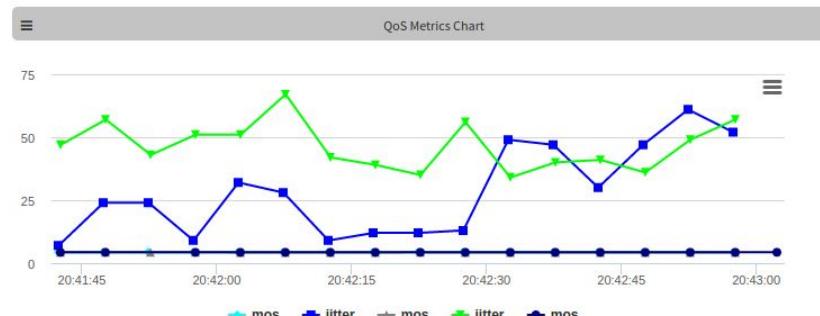
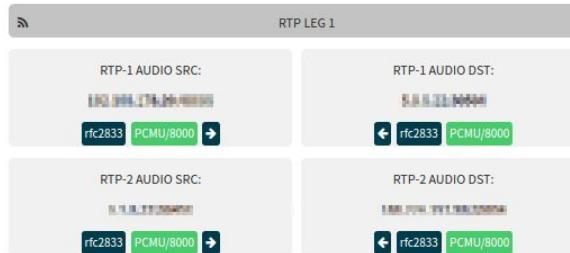
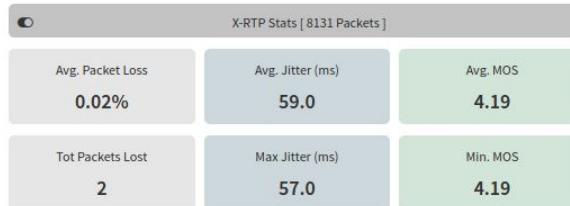
```

sip\_capture\_plan.xml

# CAPTAGENT 6.1 HEP/EEP RTCP + SIP Mirroring

Example Usage of the Universal Capture Agent for Monitoring

If you configured everything correctly, your HOMER 5 **QoS statistics** will start being populated:



RTP[4037467638]

- mos
- packets
- jitter
- packets\_lost

RTP[143887173]

- mos
- packets
- jitter
- packets\_lost

RTPXR

- mos
- jitter
- packets\_lost



# RTPAGENT PRO Modules

Commercial Capture Extensions with Advanced Functionality

**RTPAgent** is a “privacy-friendly” Analytics and Reporting probe for **HOMER 5** performing wire-speed RTP session and network packet analysis in-transit and in real-time without storing any data to disk (unless desired) and delivers granular periodic and final reports with a full stack of dedicated metrics at each interval:

- Source/Destination IP/PORT/MAC
- Bytes/Packets Total, Expected
- Packet Loss
- Jitter (min/man/mean)
- RTT Delta/Skew (min/man/mean)
- Codec ID, Clock Rate
- MOS Estimation
- R-Factor Estimation

RTP Reporting frequency can be defined by the integrator or self-adjusted by the probe to send higher number of periodic QoS reports for sessions where suspect quality issues are identified and to automatically reduce the number of reports for those delivering high scores in order to minimize the bandwidth overhead.

**RTPAgent** is designed to deal with multi-party and multi-codec calls including video sessions and can automatically detect/report a vast number of conditions.

## Additional Modules:

- ★ On-Demand, Filtered Stream Recording to Disk (SIP/RTP/RTCP)
- ★ Lawful Interception (X1/2/3 ETSI 232)

```
{
  "CORRELATION_ID": "56a211936328-fgbtmubkimot",
  "RTP_SIP_CALL_ID": "56a211936328-fgbtmubkimot",
  "DELTA": 19.980,
  "JITTER": 0.023,
  "REPORT_TS": 1453461919,
  "TL_BYTE": 0,
  "SKEW": -0.180,
  "TOTAL_PK": 510,
  "EXPECTED_PK": 510,
  "PACKET_LOSS": 0,
  "SEQ": 0,
  "MAX_JITTER": 1.892, "MEAN_JITTER": 0.126,
  "MAX_DELTA": 35.547, "MAX_SKew": -15.615,
  "MIN_MOS": 4.385, "MEAN_MOS": 4.394, "MOS": 4.394,
  "RFACTOR": 92.449, "MIN_RFACTOR": 92.013, "MEAN_RFACTOR": 92.444,
  "SRC_IP": "192.168.178.34", "SRC_PORT": 58320, "DST_IP": "192.168.60.70", "DST_PORT": 32728,
  "SRC_MAC": "00-04-13-29-64-22", "DST_MAC": "34-31-C4-38-24-00",
  "CODEC_PT": 9, "CLOCK": 8000, "CODEC_NAME": "g722", "DIR": 1,
  "REPORT_NAME": "192.168.178.34:58320", "PARTY": 0, "TYPE": "PERIODIC"
}
```



# Q & A

# HOMER 5

## Dashboard and Widget management

**Homer 5** brings a lot of changes and much more flexibility in line with other popular data-mining platforms around. Our evergreen **Wiki** on **Github** provides a number of useful resources to get started (or refreshed) including:

- ★ How to Install and Update Homer
- ★ How to get started with the User-Interface
- ★ How to customize Panels and Widgets
- ★ How to manage Users and Aliases
- ★ How to configure HEP Capture Agents
- ★ How to configure HEP Custom Agents
- ★ How to correlate Sessions and Reports
- ★ How to make your own Statistics and Widgets

..... and much more !

*"Just HEP Yourself, to my SIPs... "*

<https://github.com/sipcapture/homer/wiki/>

The screenshot shows the "Welcome to the SIPCAPTURE WIKI!" page. At the top, there's a header with a "Pages" button and a search bar. Below the header is a sidebar titled "Homer 5 Wiki" containing icons for the logo and "HOMER". The main content area features a dashboard with several charts: a green bar chart for "Call Duration", a blue bar chart for "Call Volume", and a pie chart for "Session Status". To the right of the charts is a "Time Range" section with dropdown menus for "Last 10 min", "Last 1 hr", "Last 8 hrs", "Last 24 hrs", and "Last 7 days". Below the charts is a "QUICK Widgets" section with options like "List of sessions", "List of calls", "List of users", and "List of nodes". At the bottom of the dashboard, there's a "Get started with HOMER by SIPCAPTURE.ORG" section featuring three icons: a person with a brain and a lightbulb, a document with a checkmark, and a stack of papers with a red "H". Below these icons are the phrases "Learn all about our Project", "Design your capture", and "Easily Install". The sidebar on the right contains a tree-like navigation menu with sections for "How-to Install" (Docker, Packages, 5.x), "Using Homer 5" (Dashboards, Administration, Searching, Visualizers, Alarms, Aliases, Correlation, QoS Reports, Rison Parameters), and "FAQ & Troubleshooting".

*That's all Folks!*



Time's UP! Want to go further? "HEP" Yourself!

<b>SIPCAPTURE</b> @GITHUB	<a href="http://sipcapture.org">http://sipcapture.org</a> + <a href="http://sipcapture.io">http://sipcapture.io</a>
<b>HOMER</b> @GITHUB	<a href="http://github.com/sipcapture/homer">http://github.com/sipcapture/homer</a>
<b>CAPTAGENT</b> @GITHUB	<a href="http://github.com/sipcapture/captagent">http://github.com/sipcapture/captagent</a>
<b>HEPIPE.JS</b> @GITHUB	<a href="http://github.com/sipcapture/hepipe.js">http://github.com/sipcapture/hepipe.js</a>
<b>MAILING-LIST</b> @USERS	<a href="https://groups.google.com/forum/#!forum/homer-discuss">https://groups.google.com/forum/#!forum/homer-discuss</a>