

rtpengine media processing for fun and profit



Andreas Granig • 09.05.2017

<agranig@sipwise.com>

Media Processing?

Relaying

Transcoding

Encrypting

Injecting

Recording

Capturing

Transscribing

Monitoring

Agenda

Introduction

Relaying

Capturing

Encrypting

Recording

Monitoring

Introduction

rtpengine is...

- **a media relay**
- **used by (but not limited to) Kamailio**
- **kernel based**
- **highly available**
- **doing crypto stuff for WebRTC**
- **hosted at github.com/sipwise/rtpengine**
- **much more!**

Relaying

rtpengine Kamailio Module

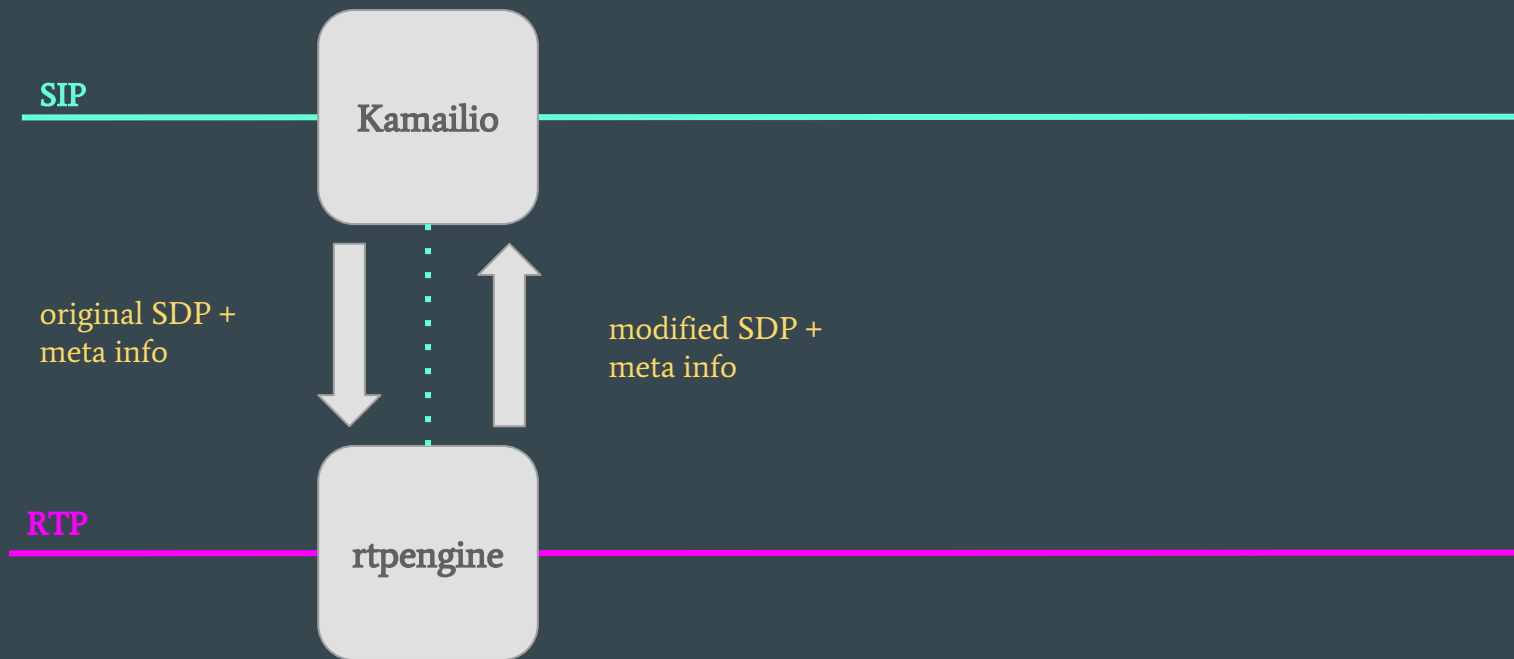
controlling

rtpengine Daemon

driving

rtpengine Kernel Module

Relaying



Relaying

Kamailio

```
loadmodule "rtengine.so"

modparam("rtengine", "rtengine_sock",
    "udp:127.0.0.1:12221")
```

```
route {
    ...

    rtengine_manage();

    ...
}
```

Daemon

```
# rtengine \
    --interface=ext/1.2.3.4 \
    --listen-ng=127.0.0.1:12221
```

Encrypting

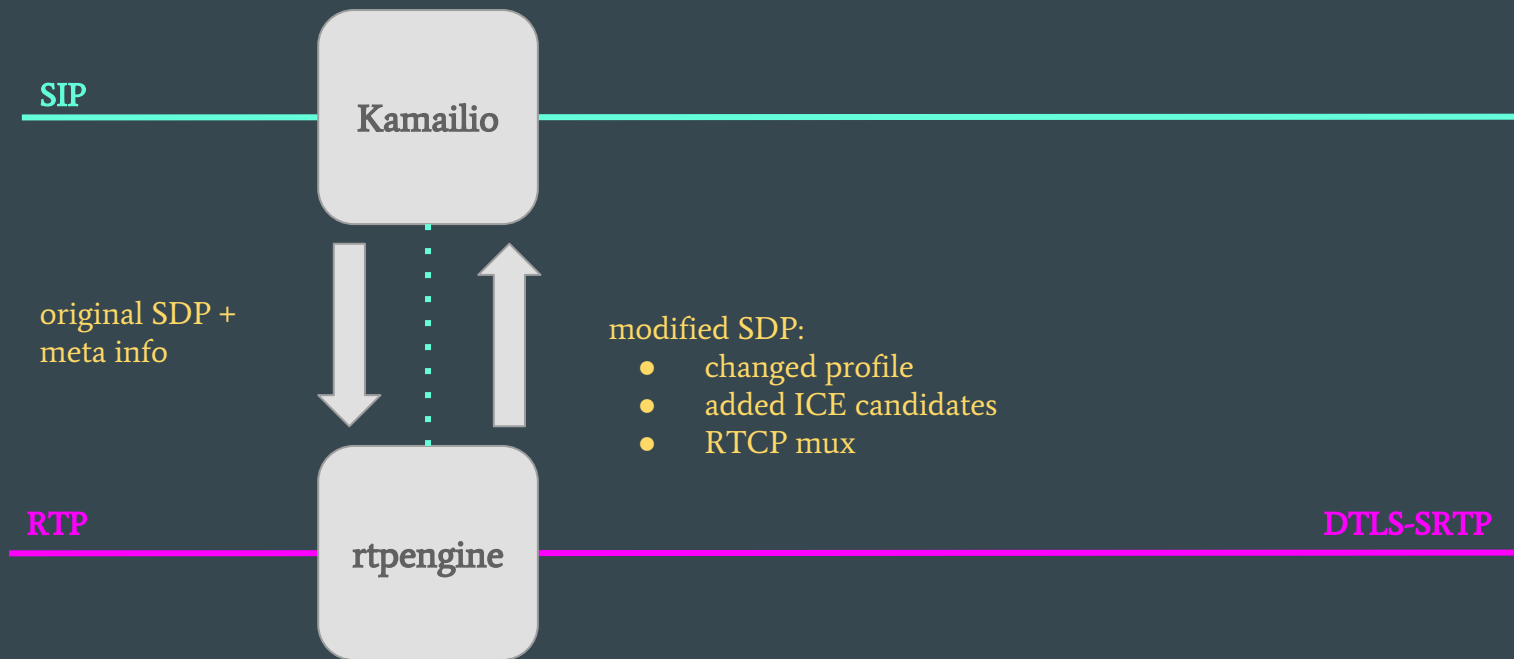
Transform SDP Profiles

for bridging SIP and WebRTC,

Encrypt/Decrypt DTLS-SRTP

in Kernel

Encrypting (eg. WebRTC Bridging)



Encrypting

Kamailio

```
loadmodule "rtengine.so"

modparam("rtengine", "rtengine_sock",
    "udp:127.0.0.1:12221")

route {
    rtengine_offer("RTP/SAVPF ICE=force ");
}

reply_route {
    rtengine_answer("RTP/AVP ICE=remove");
}
```

Daemon

```
# rtengine \
    --interface=ext/1.2.3.4 \
    --listen-ng=127.0.0.1:12221
```

Capturing

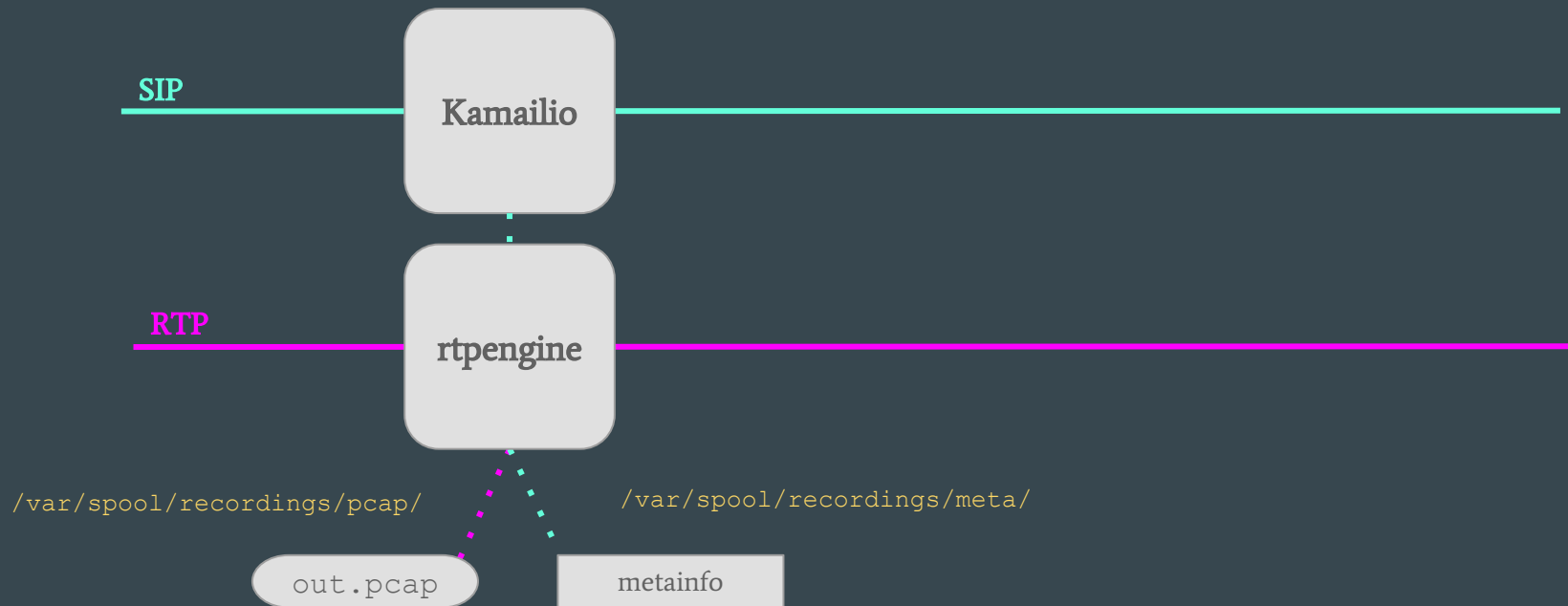
Write Streams as PCAP

into spool directory,

Inspect with Wireshark

for further analysis.

Capturing



Capturing

Kamailio

```
loadmodule "rtengine.so"

modparam("rtengine", "rtengine_sock",
    "udp:127.0.0.1:12221")

route {
    ...

    rtengine_manage();

    ...
}
```

Daemon

```
# rtengine \
    --interface=ext/1.2.3.4 \
    --listen-ng=127.0.0.1:12221 \
    --recording-method=pcap \
    --recording-dir=/tmp/rtp \
    --recording-format=eth
```

Recording

Expose call data in meta files

to pick up new calls,

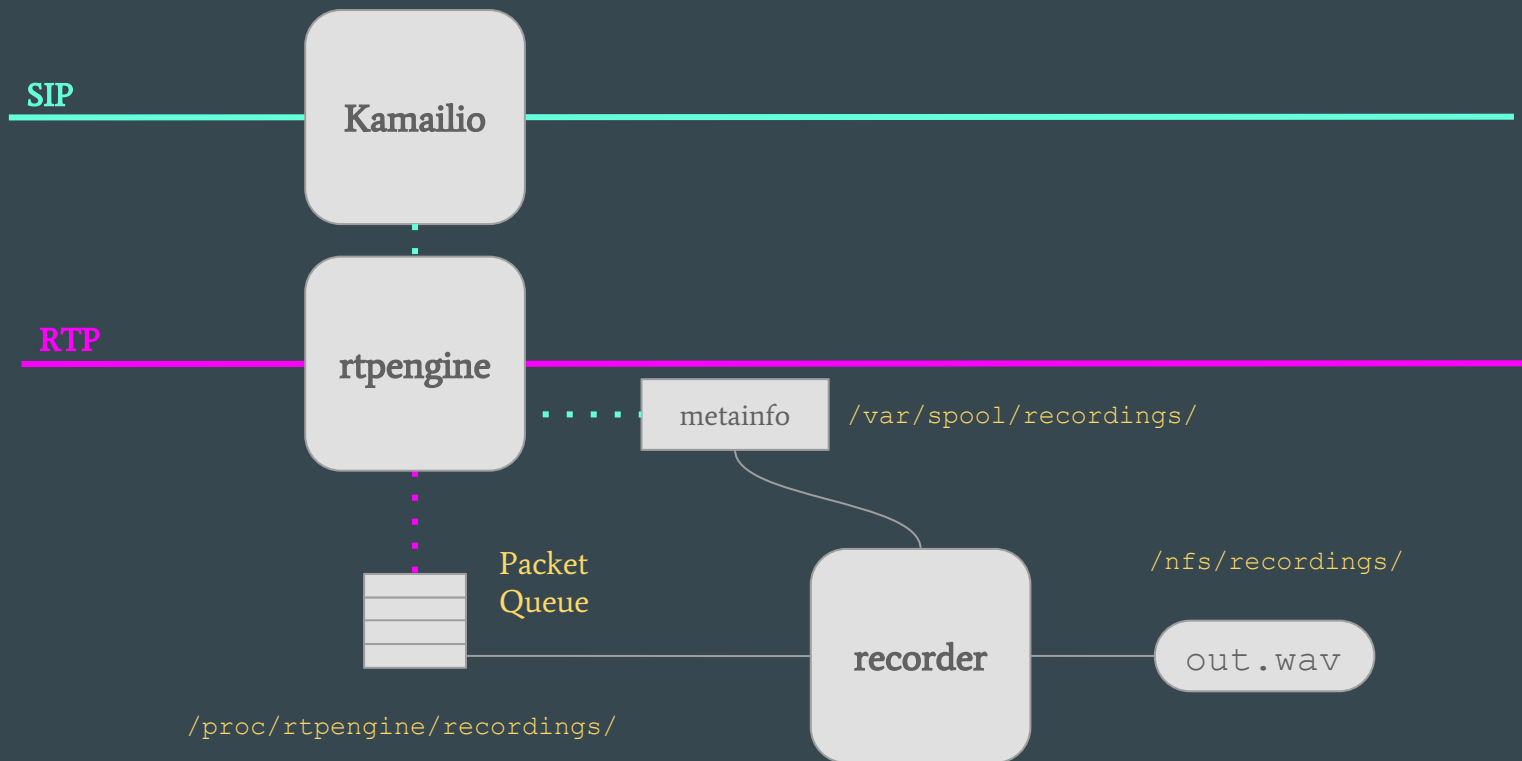
Emit RTP via /proc

for further processing,

Read, transcode & store Data

as WAV or MP3

Recording



Recording

Kamailio

```
loadmodule "rtengine.so"

modparam("rtengine", "rtengine_sock",
    "udp:127.0.0.1:12221")

route {
    ...

    rtengine_manage("record-call
metadata=uuid:$var(aid)|uuid:$var(bid)");

    ...
}
```

Daemon

```
# rtengine \
    --interface=ext/1.2.3.4 \
    --listen-ng=127.0.0.1:12221 \
    --recording-method=proc \
    --recording-dir=/var/spool/rtengine
```


Monitoring

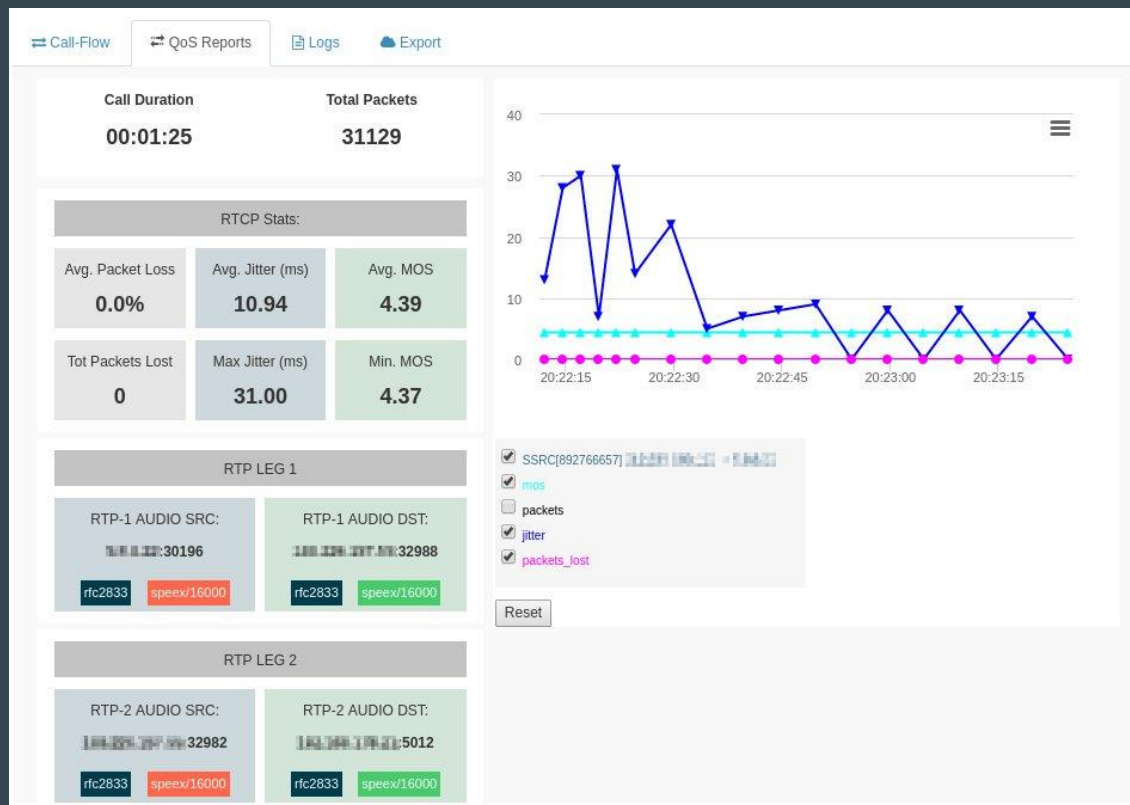
Collect and Analyze RTP & RTCP

to calculate MOS
or duplicate to Homer,

Expose to Kamailio on delete

for storing and reporting

Monitoring



Monitoring (Homer)

Kamailio

```
loadmodule "rtengine.so"

modparam("rtengine", "rtengine_sock",
    "udp:127.0.0.1:12221")

route {
    ...

    rtengine_manage();

    ...
}
```

Daemon

```
# rtengine \
    --interface=ext/1.2.3.4 \
    --listen-ng=127.0.0.1:12221 \
    --homer=10.10.10.5:9060 \
    --homer-protocol=udp \
    --homer-id=99
```

Monitoring (native)

Kamailio

```
loadmodule "rtengine.so"

modparam("rtengine", "rtengine_sock",
    "udp:127.0.0.1:12221")

modparam("rtengine", "mos_min_pv",
    "$avp(mos_min)")

modparam("rtengine", "mos_max_pv",
    "$avp(mos_max)")

modparam("rtengine", "mos_average_pv",
    "$avp(mos_avg)")
```

```
route {

    # invite:

    rtengine_offer();

    # bye:

    rtengine_delete("to-tag");

    xlog("L_INFO", "Call-ID $ci MOS
    $avp(mos_min)/$avp(mos_max)/$avp(mos_avg)");

    # curl, acc, db/ndb for further exposure

}
```

Questions?

<agranig@sipwise.com>