

Keynotes

OpenSIPS 2.3

Bogdan Andrei Iancu

- 2 May 2017 -



Welcome



AMSTERDAM **OPENSIPS** SUMMIT 2017



OpenSIPS 2.3 - when ?



- Beta release - 16th of March
- Stable Release - 26th April 2017
- General Availability - 2nd of May 2017

Keyword “Integration”



-
- VoIP platforms/systems are more than SIP Engines/OpenSIPS
 - OpenSIPS must have the ability to integrate and work together in all possible means with other projects, protocols, systems or concepts.

Benefits of the “Integration”



- Ability to build complex systems
 - More possibilities means more value
 - Brings in third-party value
- Reduces the user effort
 - As required knowledge
 - As required work

The Success of the “Integration”



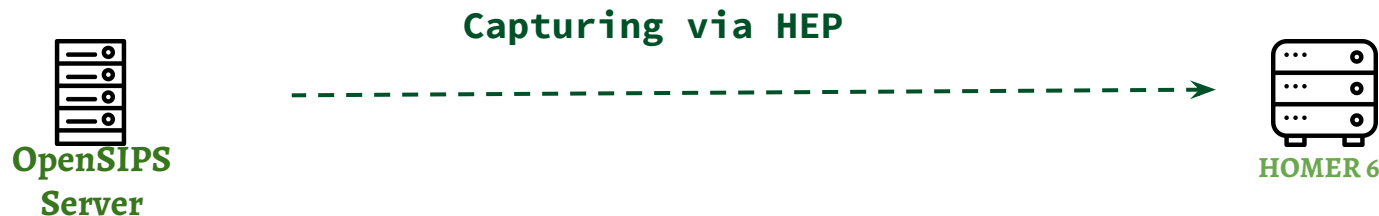
-
- A collaboration with the teams of the partner projects in terms of ideas, brainstorming, solutions and of course, work.
 - A collaboration that resulted in solutions and benefits for all the involved communities.

SIPCapturing/HOMER Integration

Integrated capturing



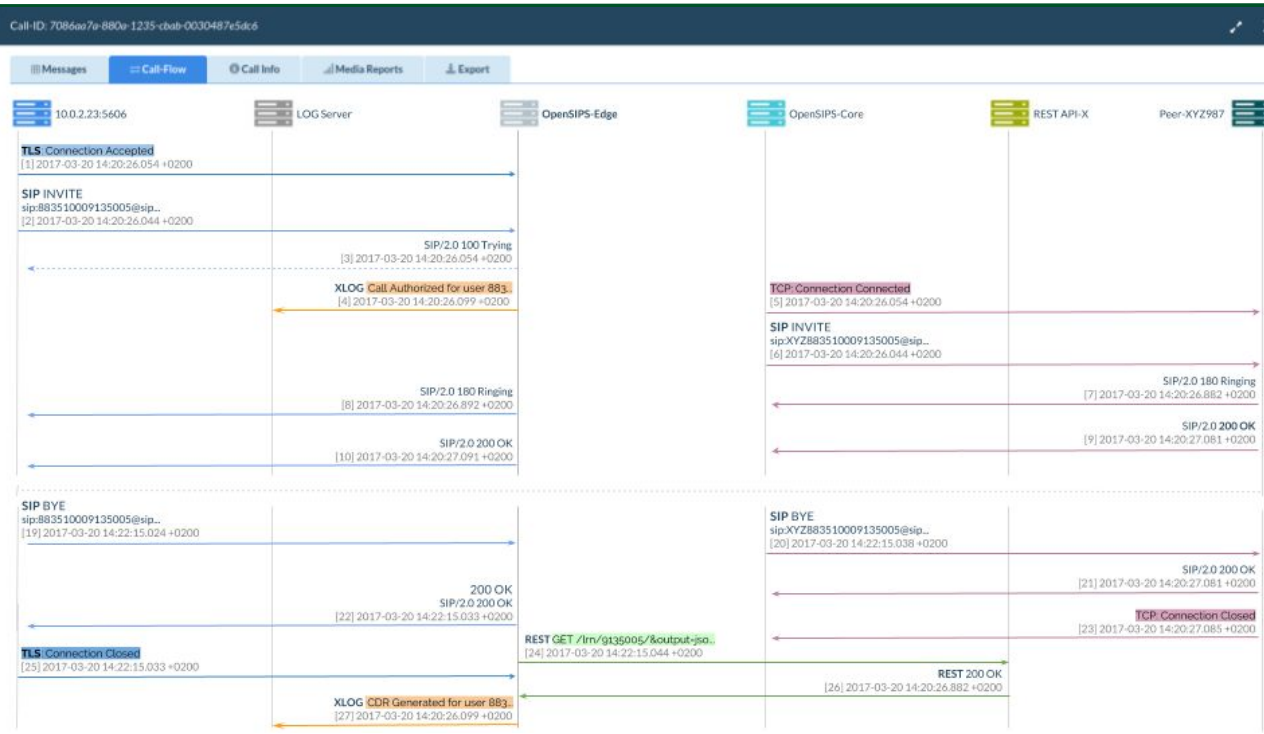
- non-SIP tracing - tracing more than SIP
- data correlation - see the being picture



Correlation between:

- SIP
- LOGS
- REST queries
- MI commands
- Transport layer (TLS, WS, WSS)

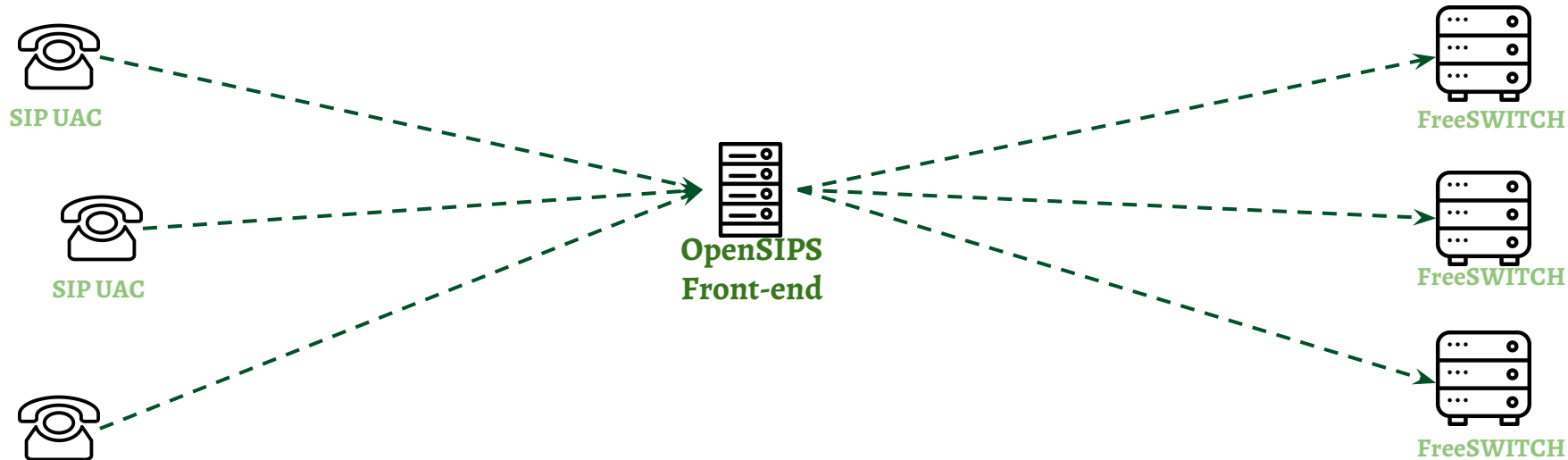
Global view



FreeSWITCH integration

-
- An OpenSIPS frontend can offload and help FreeSWITCH with REGISTER traffic processing.
 - OpenSIPS can prolong the registration sessions or can aggregate all the registration for the same AOR/subscriber.

Mid-registrar 1

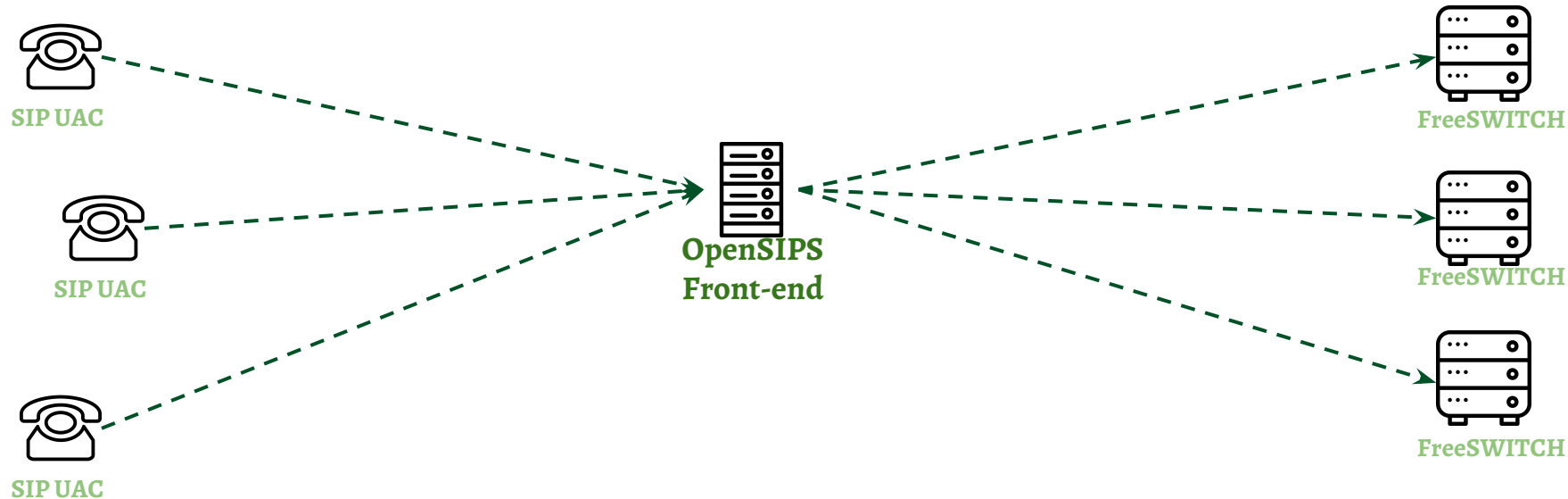


SIP UAC
SIP UAC
SIP UAC

60 seconds registration rate ->
60 registers / hour

3600 seconds registration rate ->
1 register / hour

Mid-registrar 2



3 devices per AOR ->

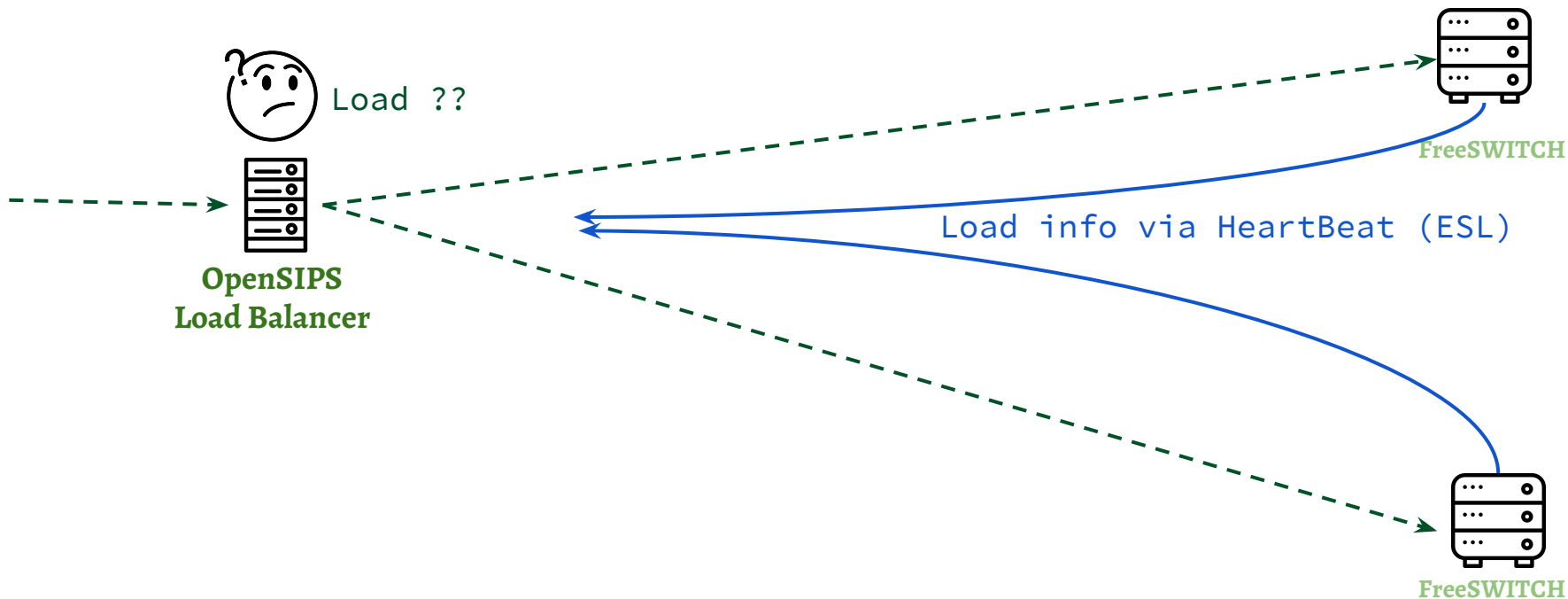
3 registers / AOR

1 contact per AOR

1 register / AOR

-
- get load feedback from FreeSWITCH via ESL
 - do an accurate load-balancing over a FreeSWITCH cluster
 - Adjust the load distributions to correct external influence

FreeSWITCH driven balancing



CGRates Integration

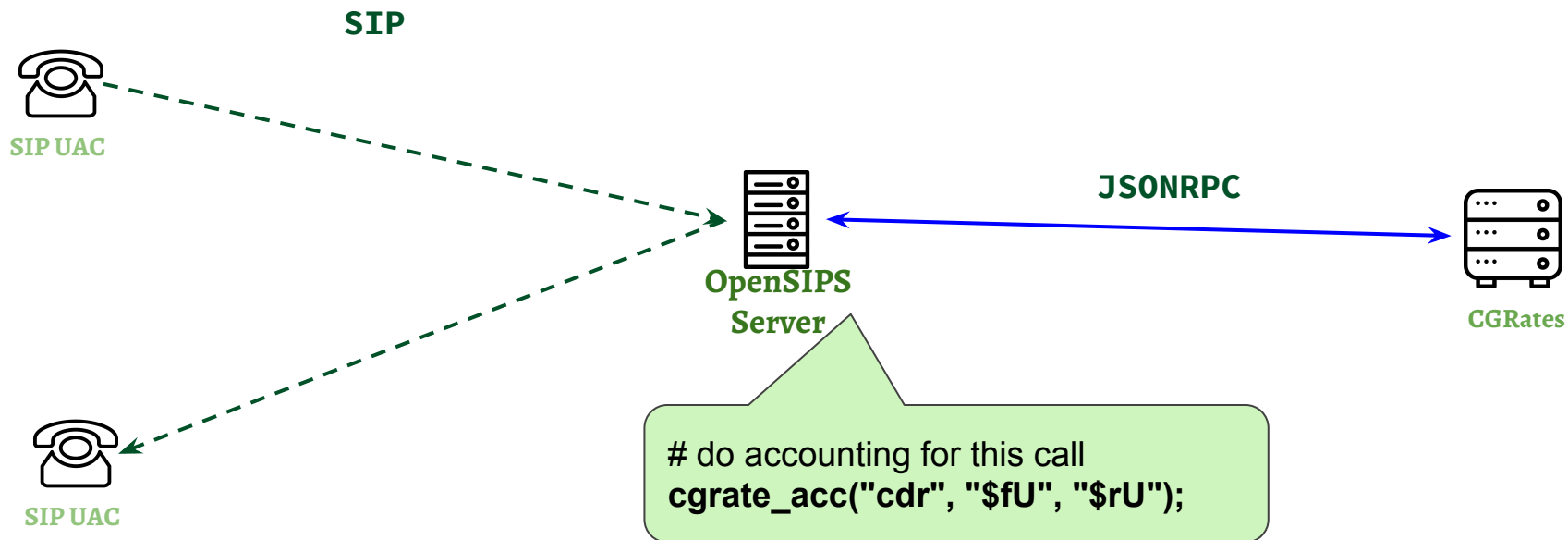
-
- SIP without billing is like a dinner without a good wine
 - CGRateS is an open-source rating engine used for
 - carrier-grade, multi-tenant, real-time billing
 - Post and pre paid

CGRates Integration



- Call authorization
- Accounting and charging
- CDR generation

CGRates billing



SIP-I Integration

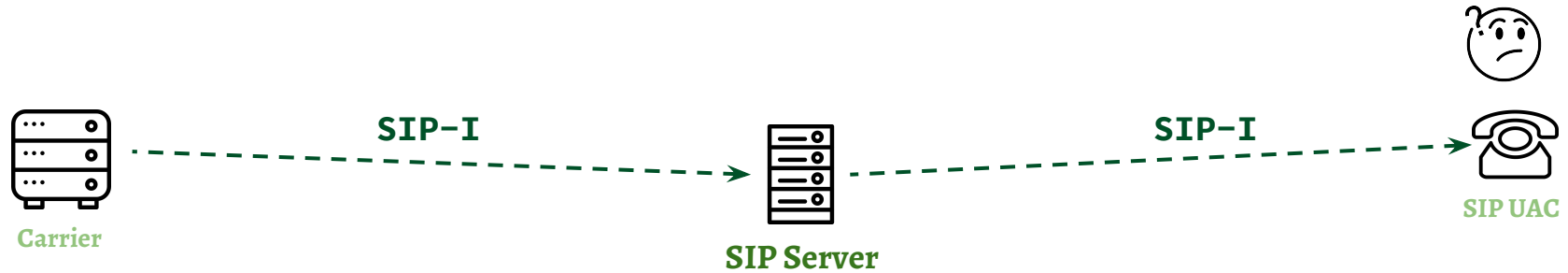
- SS7 interconnections are costly and difficult to set



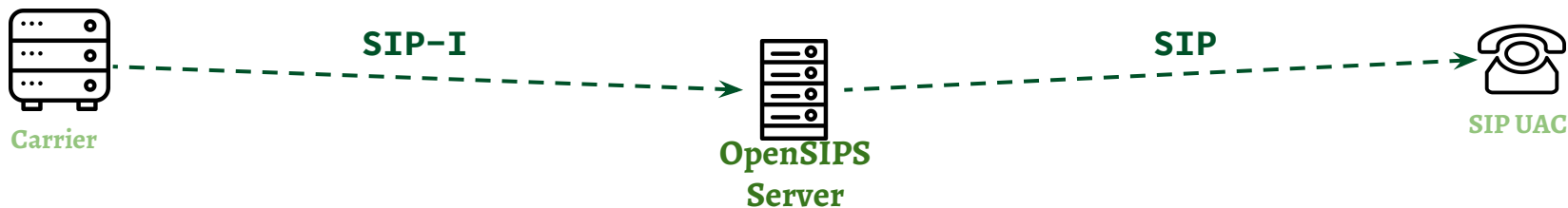
SIP-I is very powerful as:

- it is SIP based, so simple to deal with (comparing to SS7, of course ☐)
- it exposes SS7 specific data via the ISUP payload

SIP-I is not SIP



SIP-I gatewaying



Simple script usage:

- `add_isup_part("ANM")`
- `strip_body_part("application/isup")`
- `$isup_param(Called Party Number | Address signal)`

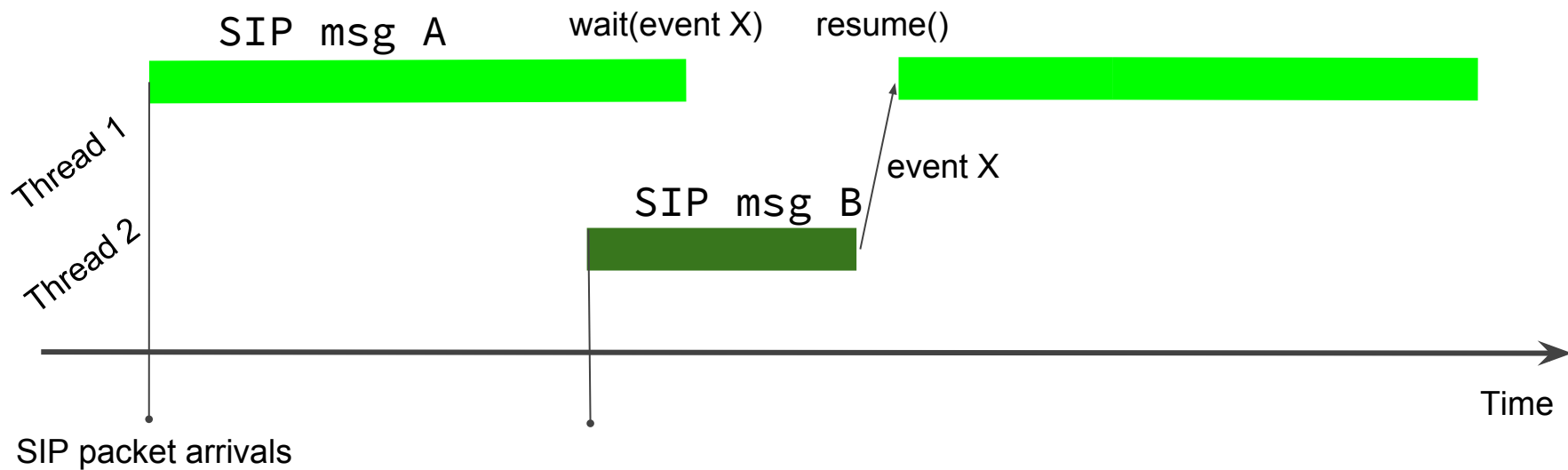
Event Based Routing

Addressed problems:

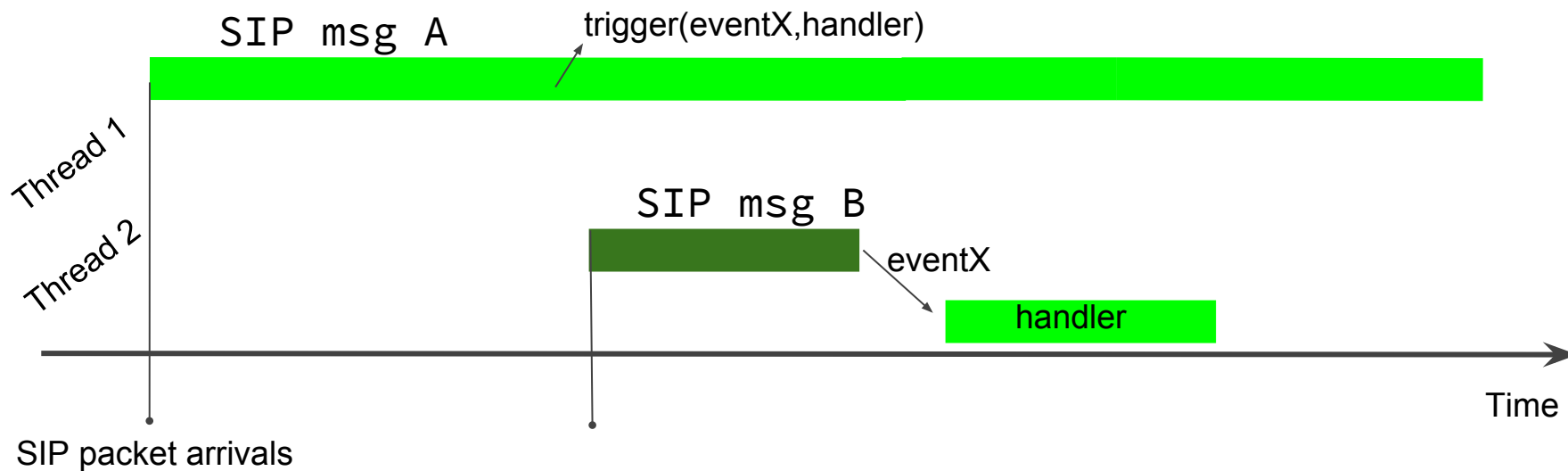
- communicate and exchange data with the processing of a different SIP message
- synchronize or wait for actions related to the processing of a different SIP message

-
- Implements a **Subscriber - Notify** model
 - **Events** - triggered by actions / data processing during runtime; events carry custom data.
 - **Subscriber** - any IPS msg processing may subscribe for events (with filtering)

Event async waiting



Event notification



EBR scenarios



- Push notification
- Call pickup
- DTMF call driving

Push Notification scenario



- we want to deliver the call right away to already registered UACs, without any waiting due the PN logic.
- we want to do parallel forking (simultaneous ringing) to newly registered UACs, as soon as it registers (as a result of PN)
- we want to keep the call on, even if the registered UACs declines it (maybe due to DND) and give a chance to the PN to trigger the mobile device registration.

Push Notification



```
Route[do_pn]{
    t_newtran();
    t_wait_for_new_branches();

    $avp(filter) = "aor="+$rU+"@"+$rd;
    notify_on_event("E_UL_CONTACT_INSERT", "$avp(filter)", "fork_call");

    launch ( exec("/my/pn_triggering_script.sh") );

    if (lookup("location"))
        t_relay();
}

route[use_new_contact] {
    t_inject_branches("event");
}
```

And More....

RabbitMQ enhancements



- Direct trigger of RMQ messages from script
- Custom data can be pushed into RMQ messages
- Custom formatting (attributes/headers) of RMQ messages
- Fine tuning of the AMQP protocol options



XML wise variables



- provides parsing and manipulation of XML documents
- Very useful for Presence related data manipulation

```
$xml(body) = $rb(application/dialog-info+xml);
```

```
$var(identity) = $xml(body/dialog-info/dialog/remote/identity.val);
```

Extended Async Support



“Launch()” script statement

- Spinoff some processing without waiting for it to finish
- Continue the script execution (no resuming)

...

```
launch( avp_db_query("insert into .....") );
```

```
xlog("insert started, continuing...\n");
```

...

More Reliable Accounting



A more reliable way to populate ACC data

- `$acc_extra[]`
- `$acc_leg[]` (instead of old complicated AVP)

```
modparam("acc", "leg_fields", "log: caller; callee")
```

```
...
```

```
acc_new_leg();
```

```
$acc_leg(callee) = $ru;
```

```
...
```

Enjoy OpenSIPS 2.3

- Bogdan Iancu
 - OpenSIPS Project: www.opensips.org
 - Email: bogdan@opensips.org