

OpenSIPS Event Interface

- workshop -

Liviu Chircu

 @liviuchircu 

OpenSIPS Solutions

About me

- OpenSIPS developer for 4 years
- specialize in
 - code design / maintenance (best practices)
 - memory management
 - OpenSIPS troubleshooting
- contact
 - github.com/liviuchircu
 - twitter.com/liviuchircu
 - #opensips on Freenode
 - liviu@opensips.org

Motivation

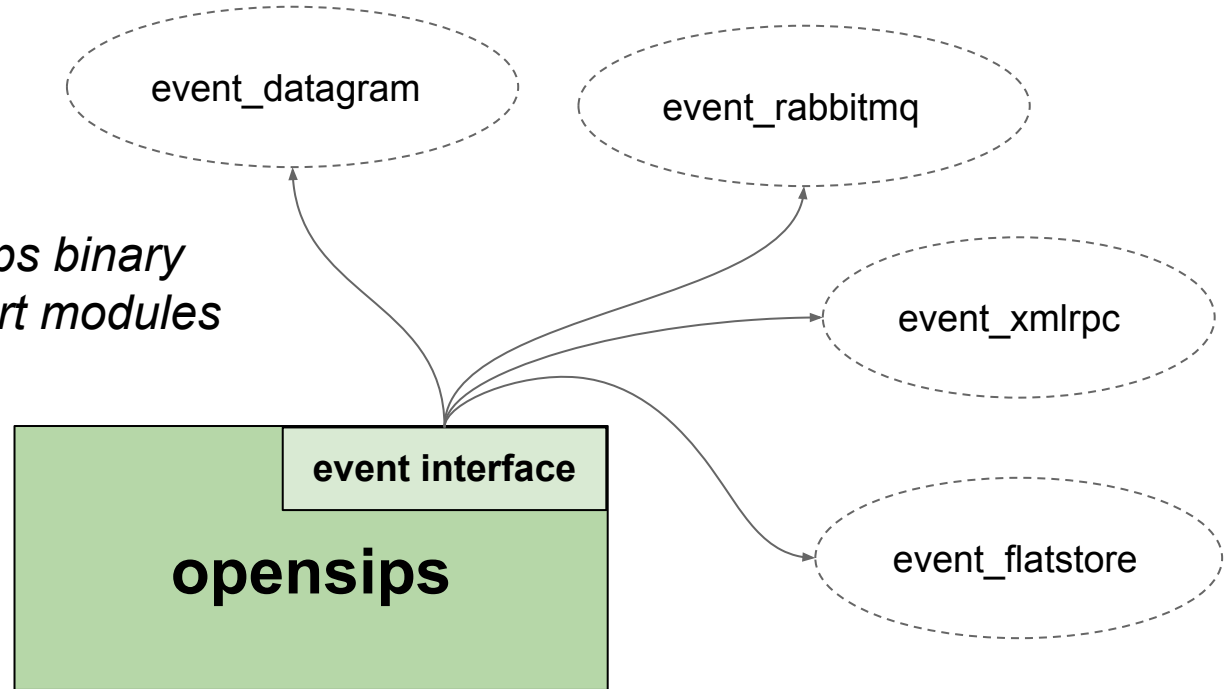
- *Some events cannot be logged at script level!*
 - *RTPProxy node state*
 - *“dispatcher” destination state*
 - *architectural bottleneck warnings (SQL, NoSQL, TCP, DNS)*
 - *out of memory...*
- *Multitude of destinations*
 - *log files*
 - *message brokers*
 - *HTTP servers*

History

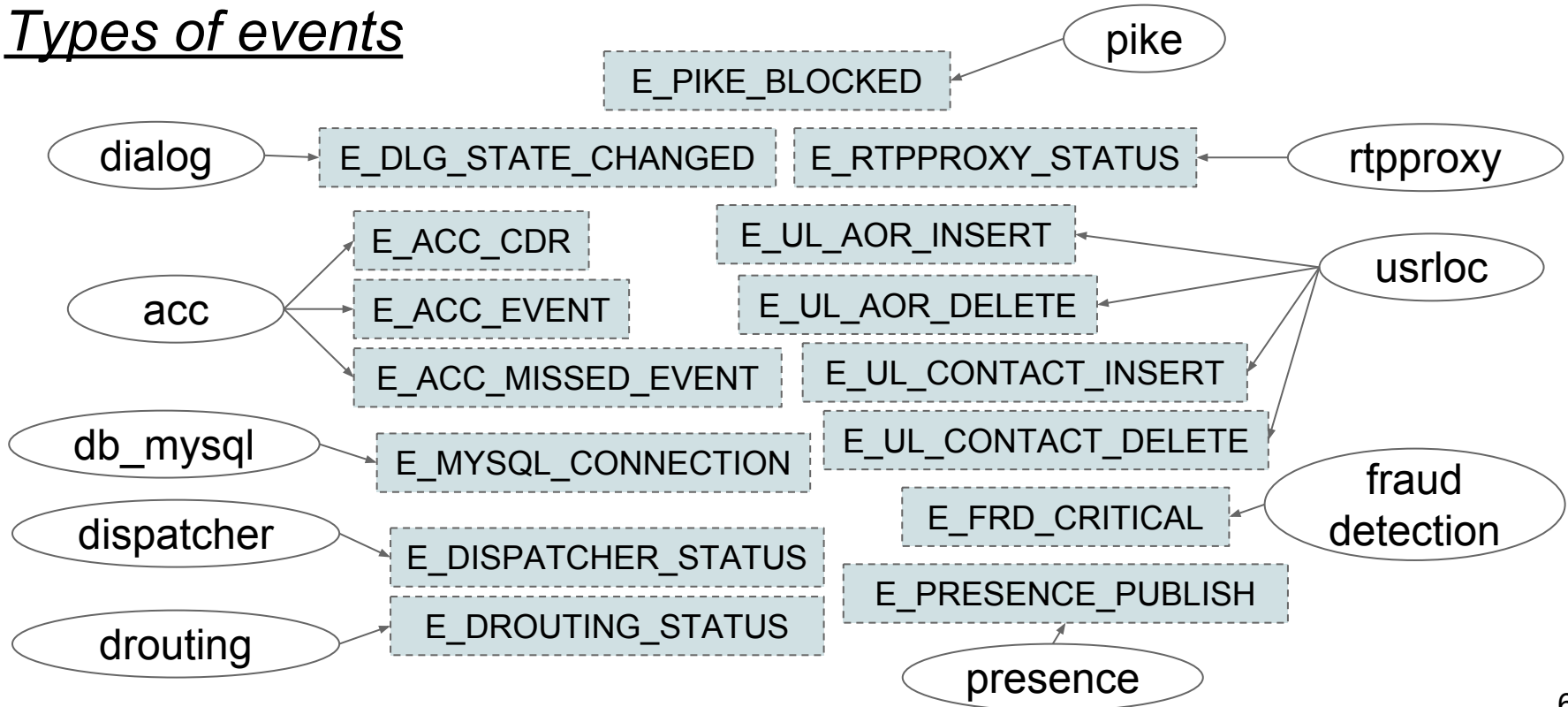
- *OpenSIPS 1.7*
 - `+++ event_datagram`
- *OpenSIPS 1.8*
 - `+++ event_rabbitmq`
- *OpenSIPS 1.9*
 - `+++ event_xmlrpc`
 - `+++ event_route`
- *OpenSIPS 2.2*
 - `+++ event_flatstore`
 - `+++ event_virtual`

Architecture

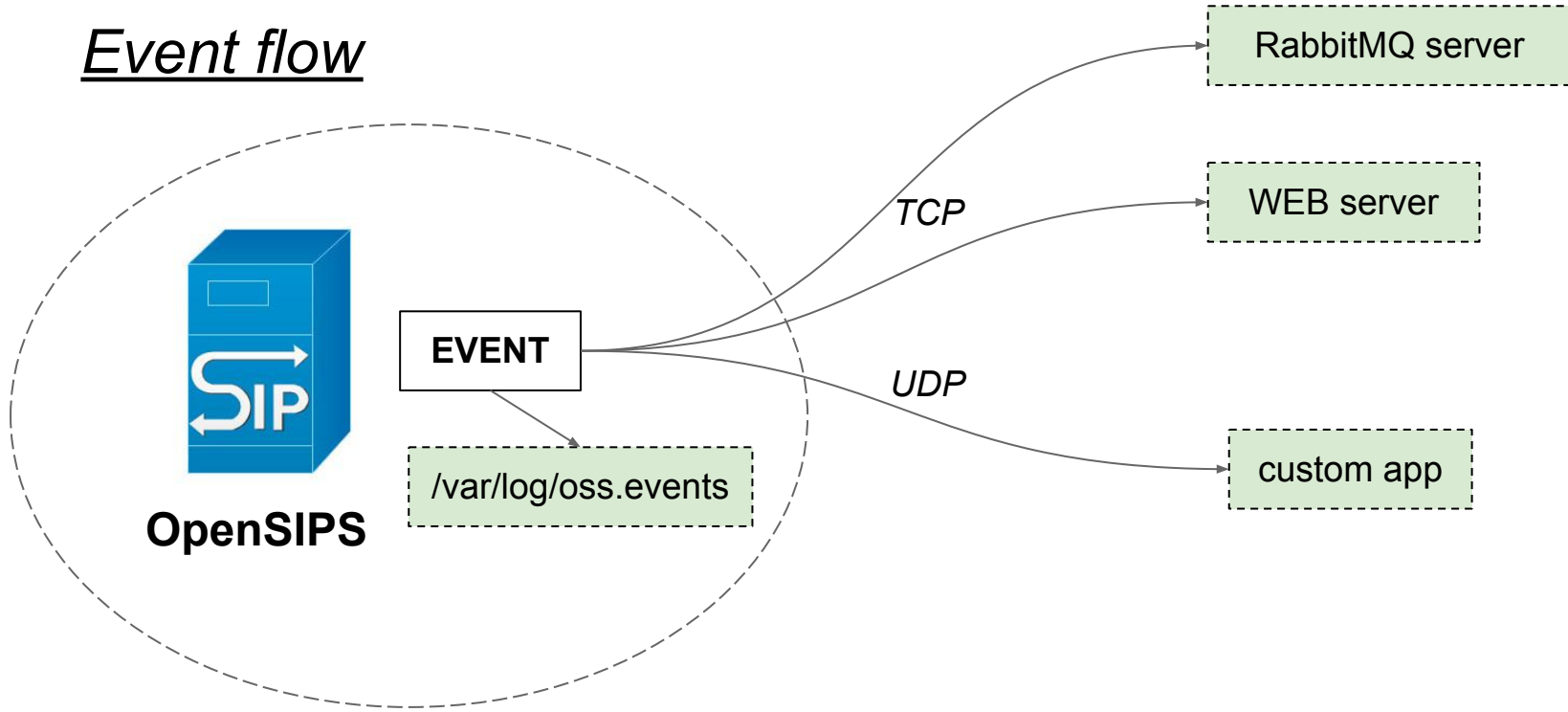
- *built into opensips binary*
- *optional transport modules*



Types of events



Event flow



Start-up

For each event transport module:

- `loadmodule "event_flatstore.so"`
- `modparam("event_flatstore", "max_open_sockets", 200)`

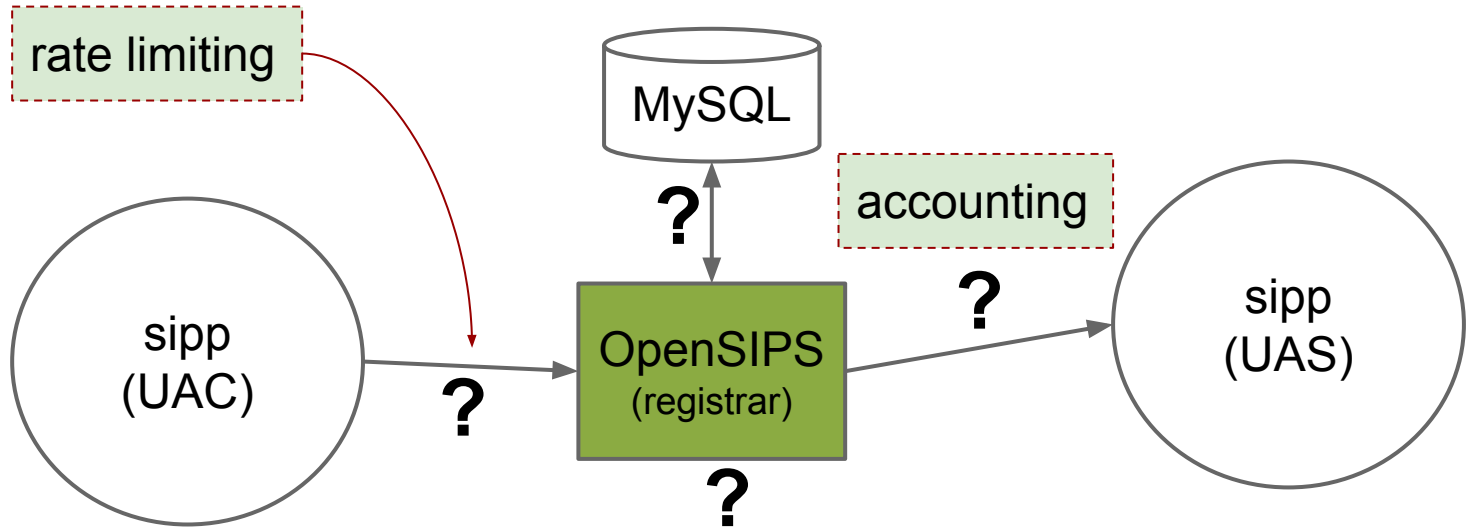
Runtime

Events can be processed:

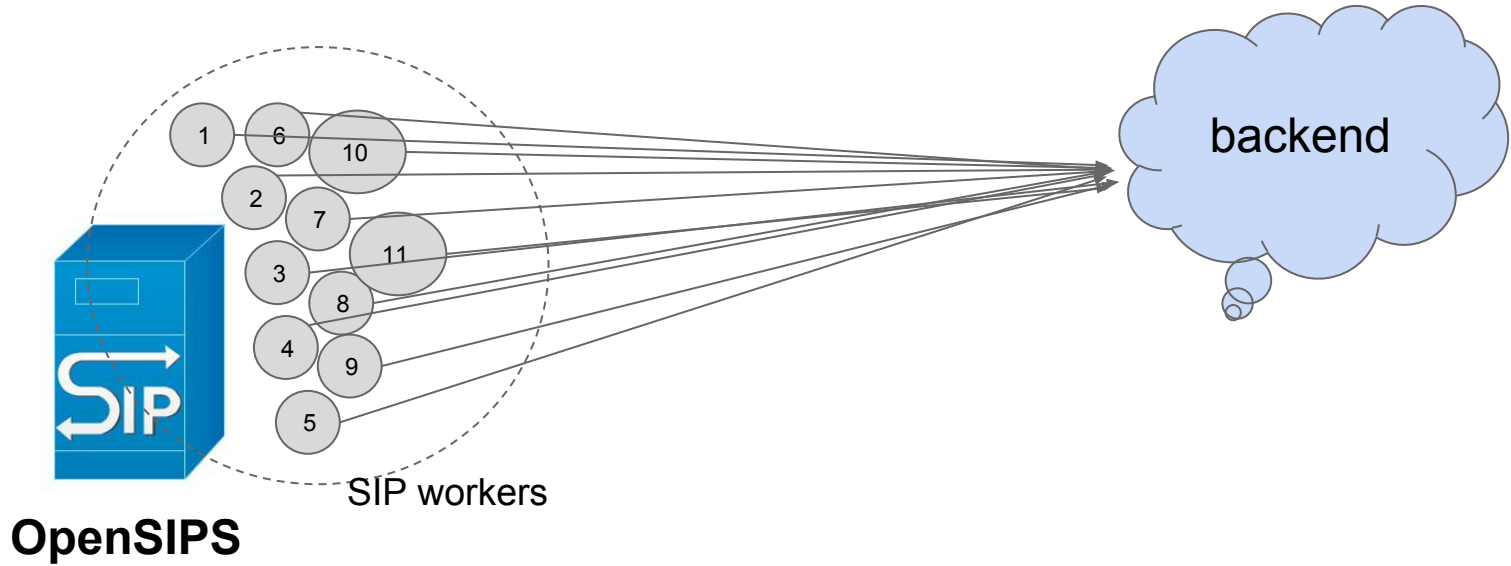
- *in the external app*
- *at script level, in the “**event_route**”:*
 - *Note: “event_route.so” module must be loaded*

```
event_route [E_PIKE_BLOCKED] {  
    fetch_event_params("ip=$avp(pike-ip)");  
    xlog("IP $avp(pike-ip) has been blocked\n");  
}
```

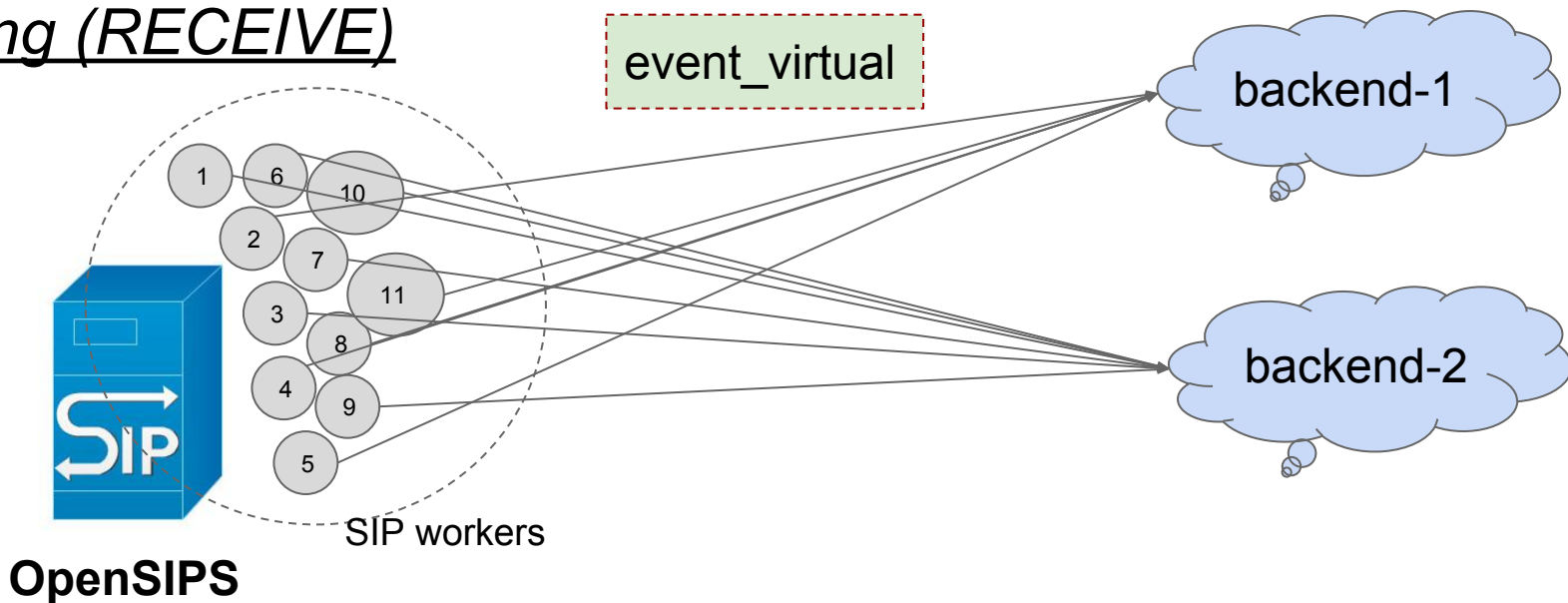
Workshop



Scaling (SEND)

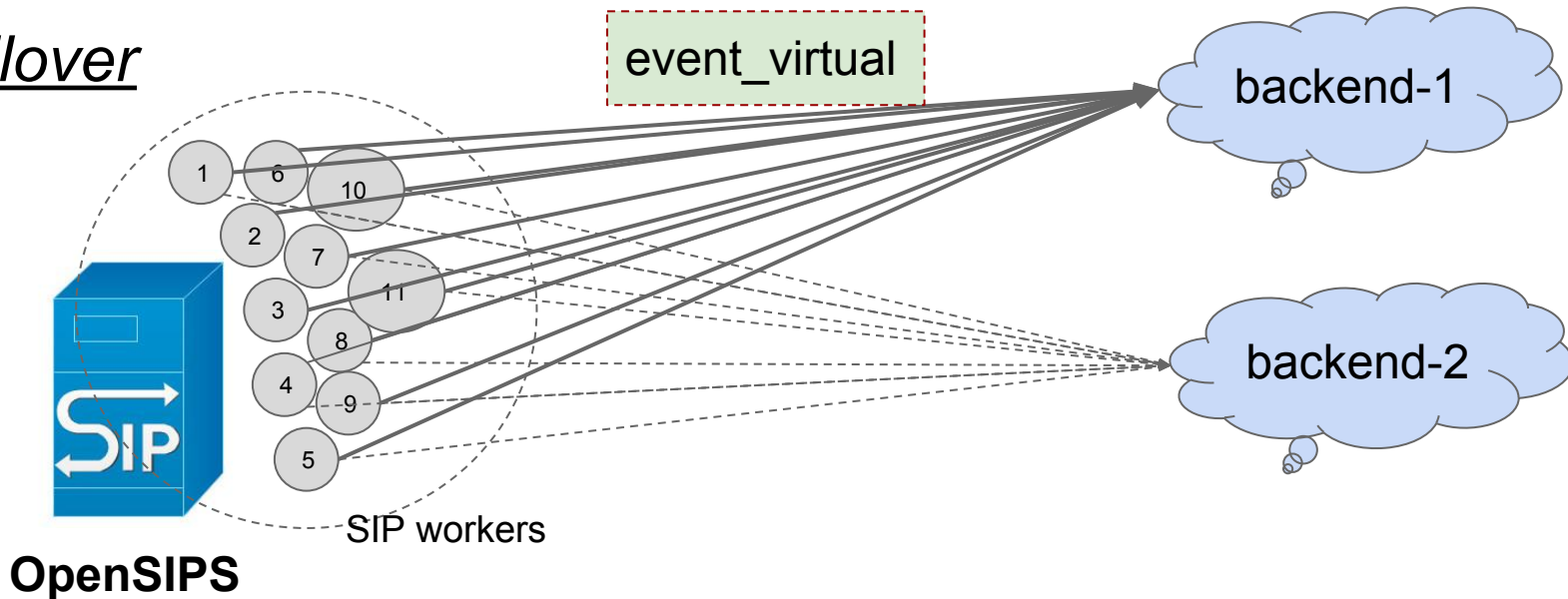


Scaling (RECEIVE)



```
subscribe_event("E_ACC_CDR",  
  "virtual:PARALLEL rabbitmq:127.0.0.1/opensips flatstore:/tmp/opensips.events")
```

Failover



```
subscribe_event("E_ACC_CDR",  
  "virtual:FAILOVER rabbitmq:127.0.0.1/opensips flatstore:/tmp/opensips.events")
```

Resources

- event interface tutorial
 - <http://www.opensips.org/Documentation/Tutorials-EventInterface>
- module documentation
 - <http://www.opensips.org/Documentation/Modules-2-2>

