OpenSIPS Summit - Keynotes

Bogdan-Andrei Iancu
Founder OpenSIPS Project
OpenSIPS Solutions

The Amsterdam story

Where, When, Why ?

- Amsterdam, the Schipol airport
- 7 years ago
- For user's needs

“by users, for users”

OpenSIPS

Users

OpenSIPS evolution

Evolution

- **1.4 version**

Evolution

- **1.11 version**

Evolution

- **2.1 version**

Evolution

- **3.1 version**

Releases

2.1 major release

- 2.1 stable release on 7th of May
- 1.11 is still maintained (as LTS)
- 1.8 and 1.10 are no longer maintained

OpenSIPS 2.1

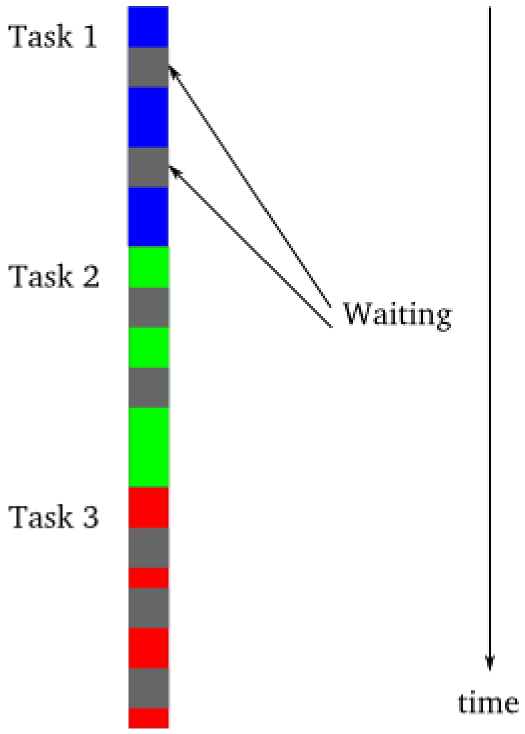
A big leap

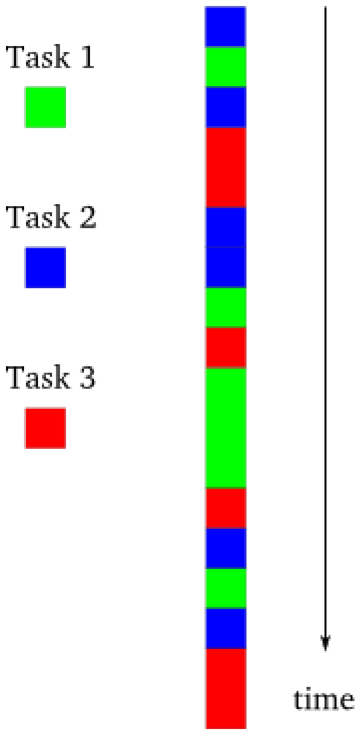
- The first OpenSIPS benefiting from the OpenSIPS Experimental results
- A new internal architecture (async reactor based)
- New concepts (processing context, execution resume)

Async I/Os

- Support for async operations from script.
- Initial support for external interactions:
  - REST client
  - SQL queries
  - exec() calls
- Requires TM support and scripting enhancements.
- No more I/O blocking !!!

Sync versus Async

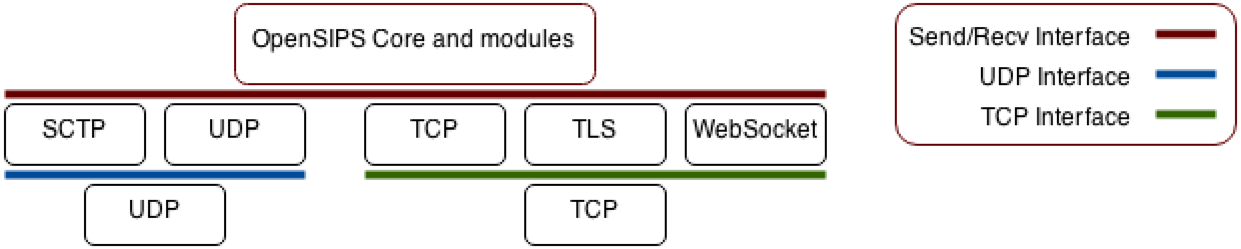




Transport protocols

- Transport protocols are now encapsulated as separate modules
- The core itself implements the low level network protocols (UDP and TCP management), while the transport modules implement the SIP transport layer (reusing the network proto implementation from core)

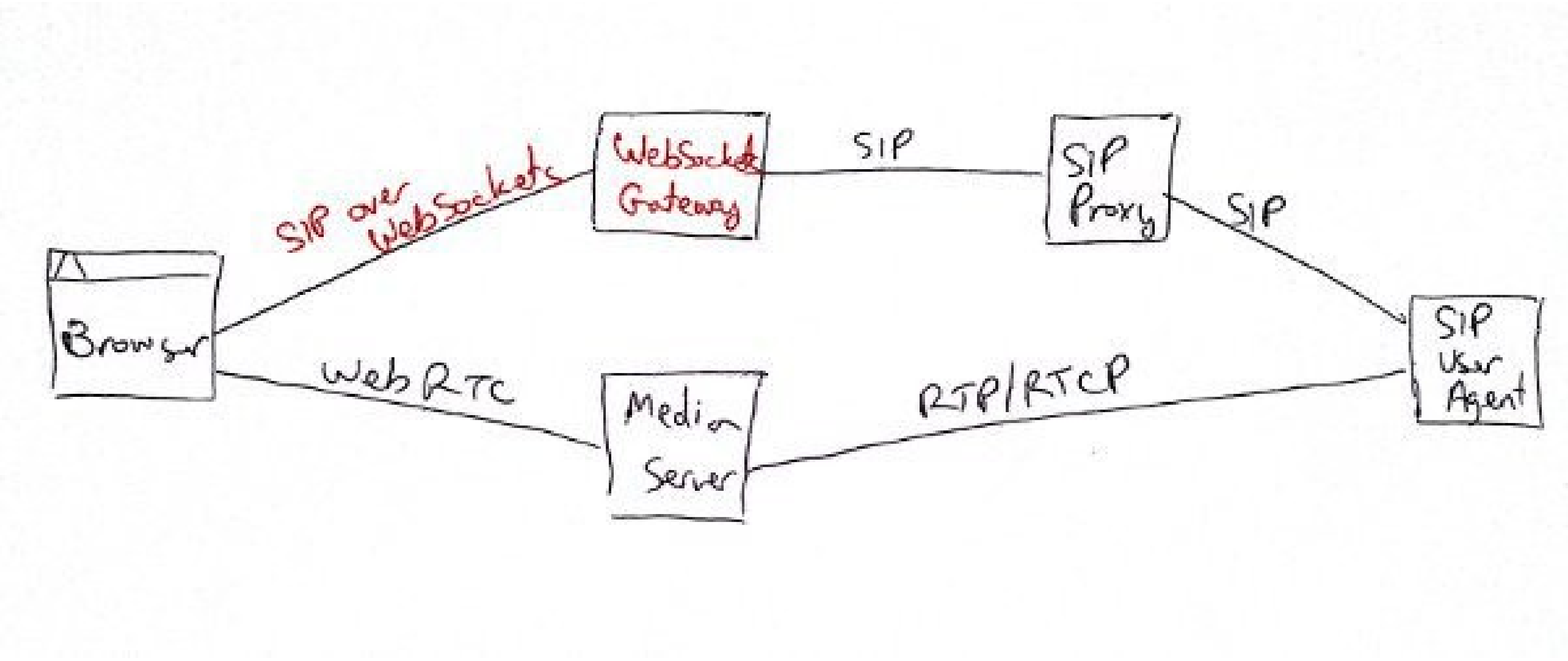
OpenSIPS 2.1


OpenSIPS Core and modules
SCTP
UDP
TCP
TLS
WebSocket
UDP
TCP
Send/Recv Interface
UDP Interface
TCP Interface

WebRTC

- WebSocket (WS) support – server side
- The WS transport module is build on top of the existing TCP management code
- OpenSIPS does WS<>SIP gatewaying, but the media part must be separately handled (RTPEngine).

WebSocket


Browser
SIP over WebSockets
WebSocket Gateway
SIP
SIP Proxy
SIP
SIP User Agent
WebRTC
Media Server
RTP/RTCP

Routing Data Partitioning

- Partition = a standalone set of routing data that can be separately managed
- Data partitioning allows using same module for multiple different scopes
- Partitions are completely separated (in DB and memory), can be individually reloaded
- Targeted modules : Dynamic Routing, Dispatcher, DialPlan

Fraud detection

- Fraud detection based on calling profiles
- A profile consists of time intervals, number of calls per interval, parallel calls, gray listed destinations
- A subscriber / trunk may get assigned a profile (to detect out of the ordinary calling patterns)
- The modules does detection and reporting (via events), but no action

SIP compression

- Simple SIP-wise traffic compacting :
  - Headers with short names
  - Combine headers with same name
  - Filter out unwanted headers
- Compress the non-routing information (header and body) from a SIP message → a new body
- Reduces bandwidth, minimizes MTU related issues, avoids unnecessary SIP parsing

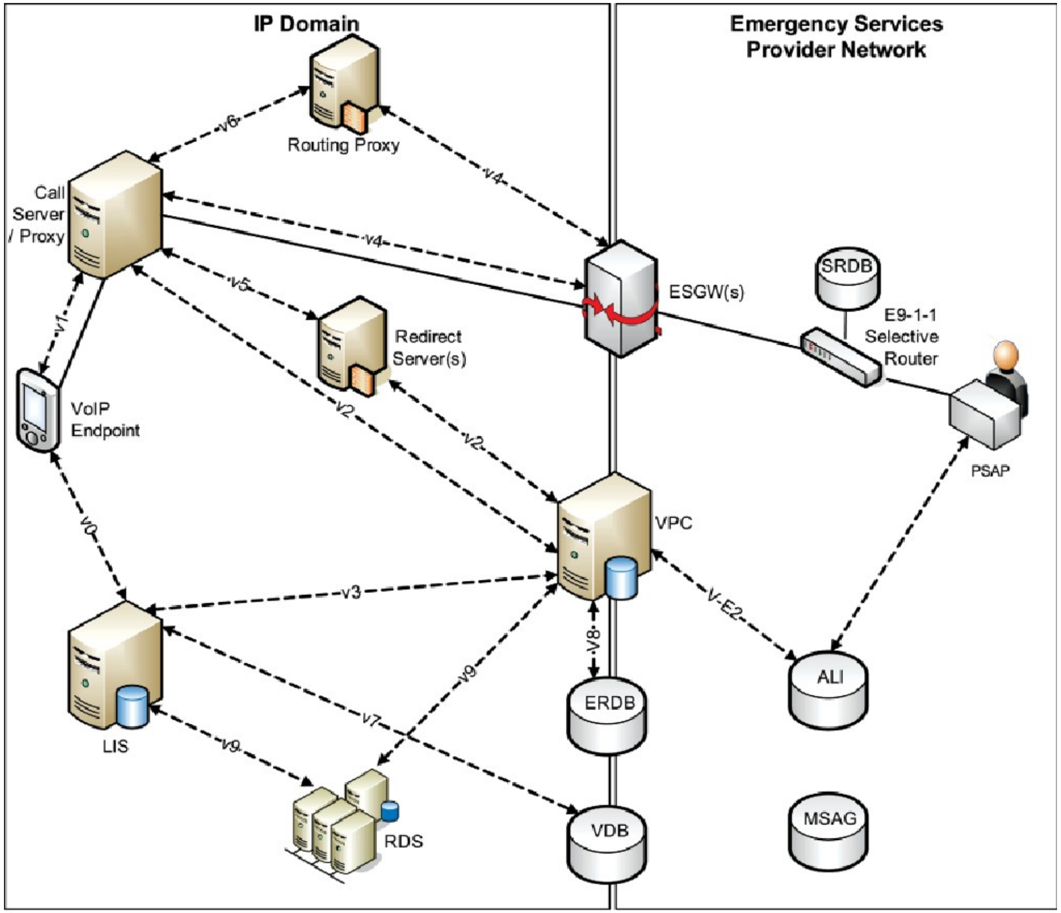
Quality based routing

- New module on top of routing engines that uses list of gateways/destinations (like Dynamic Routing, Dispatcher)
- Collect on the fly information about the call's quality (ASR, PDD, ACD, etc)
- Reorder in realtime the used gateways to remove poor quality gateways or to prioritize good quality gateways
- Complex but flexible system of thresholds (multiple levels), alerts and actions.

Emergency calling

- New module to perform routing for emergency calls
- Following the IETF specs (RFC 6881, RFC6442) and the i2 specification of the American entity NENA (National Emergency Number Association)

OpenSIPS 2.1



Legend:

- ▬▬▬ Voice Traffic
- - - - - Signaling messages (e.g., location and routing related information retrieval)

OpenSIPS 3.1

OpenSIPS 2.1

- Quality based routing
- Config file refactoring
- SQLite support
- Edge proxy support
- WebSocket client side + WSS
- Clustering user location
- More async – radius, TLS

Thank you for your attention

You can find out more at www.opensips.org

bogdan@opensips.org

www.opensips-solutions.com

Questions are welcome