Managing Calls with OpenSIPS Call API

Liviu Chircu
- May 25th, 2023 -
Outline

- Introduction
- Architecture
- Writing API Clients
- API Scenarios
- Demo
Introduction
Call API - Scope

- **Manage** SIP call flows from external applications (not a SIP UA!)
- Abstracts the handling of complex SIP scenarios
- Uses advanced in-dialog SIP handling in order to emulate the behaviour of a SIP B2BUA or UAC
Call API - Implementation

- Stand-alone tool
- Written in Golang (1.14+)
- JSON-RPC over WebSocket
  - CLI tools are available
- Modular design
- Highly scalable
  - asynchronous design
  - efficient goroutines
Call API on GitHub

- razvancrainea add support for setting a SIP uri for proxy
- cmd rework events interface
- config add support for setting a SIP uri for proxy
Architecture
Call API - Interconnection Diagram

Application Layer

WebSocket API & Events

Call API

OpenSIPS

UDP MI & EVI

SIP

SIP UA

Liviu Chircu - OpenSIPS Project / SIPhub

- OpenSIPS Summit 2023 -
Call API - Application Interconnection

**JSON-RPC Requests**
- Triggered from Application Layer
- Parameters are validated
- Replies are immediately sent

**JSON-RPC Notifications**
- Triggered from the Call API
- Carries useful payload data
- Updates regarding the status of the call (success, fail, etc.)
Call API - Application Interconnection

Call API

WebSocket

API Calls

WebSocket

Events

Application Layer

Application Layer

Liviu Chircu - OpenSIPS Project / SIPhub

[Code]

```json
{
    "method": "CallStart",
    "params": {
        "caller": "sip:alice@10.0.0.10",
        "callee": "sip:bob@10.0.0.11"
    },
    "id": "831717ed97e5",
    "jsonrpc": "2.0"
}

{
    "method": "CallStart",
    "params": {
        "cmd_id": "b8179f1e-b4e4-4ac7-9990-4bf64f084178",
        "status": "CallerAnswered",
        "data": {
            "caller": "sip:alice@10.0.0.10",
            "callee": "sip:bob@10.0.0.11"
        }
    },
    "jsonrpc": "2.0"
}
```
Call API - OpenSIPS Interconnection

Management Interface

- MI datagrams
- Triggers MI commands
- Subscribes to different events

Event Interface

- Event datagram
- Sends asynchronous events

---

Liviu Chircu - OpenSIPS Project / SIPhub

- OpenSIPS Summit 2023 -
Call API - OpenSIPS Interconnection

```
{
    "jsonrpc": "2.0",
    "id": 48,
    "method": "t_uac_dlg",
    "params": {
        "body": "v=0\n        o=click-to-dial 0 0 IN IP4
        0.0.0.0\n        s=session\n        c=IN IP4 0.0.0.0\n        t=0 0\n        m=audio 9 RTP/AVP 0\n        a=rtpmap:0 PCMU/8000\n        
        "headers": "From:\n        \n        "To:\n        \n        "Contact:\n        \n        "Content-Type: application/sdp\n        CSeq: 1
        INVITE\n        Call-ID: 9def6687-0b58-4c34-9199-aa7726584041\n        
        "method": "INVITE",
        "ruri": "sip:liviu@fosdem.api-demo.org"
    }
}
```

```
{
    "jsonrpc": "2.0",
    "method": "E_CALL_TRANSFER",
    "params": {
        "callid": "f909267f-c5a8-4652-a0c0-d414cdbe62a",
        "leg": "callee",
        "transfer_callid": "ZDFjNmY2ZjAxYTAyZjg4NQ5MTk4YzQ2YWZmOWMwZGI.",
        "destination": "sip:chili@192.168.0.199:5077",
        "state": "ok",
        "status": "200 OK"
    }
}
```
Tips on Writing API Clients

- your language needs WebSocket support
  - C, C++, Python, Java, Golang, JavaScript, Rust, Haskell, Erlang...
- JSON-RPC is trivial, may be easier to do by hand
- 🌟 concurrently re-use a WebSocket connection
- Example Go client:
  
github.com/OpenSIPS/call-api/tree/master/cmd/call-api-client
- Extension idea: REST + Webhook support
API Scenarios
Call API - Commands

- **CallStart** - start a call
- **CallEnd** - terminate a call
- **CallHold** - put a call on hold
- **CallUnhold** - resume an on-hold call
- **CallBlindTransfer** - unattended
- **CallAttendedTransfer**
Call API - Call Hold

App | Call API | OpenSIPS | SIP Caller | SIP Callee
--- | --- | --- | --- | ---
| WS:CallHold | MI:call_hold | SIP:INVITE | | |
| | EVI:CHStart(A) | | | |
| | WS:CHStart(B) | | | |
| | | | | |
| | WS:CHSuccess(A) | | | |
| | | | | |
| | WS:CHSuccess(B) | | | |

Liviu Chircu - OpenSIPS Project / SIPhub
Call API - Call Start

- **App**
  - WS:CallStart
  - WS:Started
  - WS:Answered(A)
  - WS:Answered(B)

- **OpenSIPS**
  - MI:t_uac_dlg
  - MI:call_transfer

- **SIP Caller**
  - SIP:INVITE
  - SIP:200OK
  - SIP:REFER
  - SIP:BYE
  - SIP:INVITE

- **SIP Callee**
  - SIP:INVITE
  - SIP:200OK
  - SIP:200OK
The CallOps Module

- Abstracts SIP signaling MI cmds into high-level primitives
  - call_transfer, call_hold, call_unhold
- NOT a B2BUA!
- Injects in-dialog requests
  - Re-INVITE for on-hold
  - REFER for transfer
- Intercepts replies and treats them accordingly
  - Triggers failure or success events based on them
- Commands can be triggered both through script and MI
DEMO
DEMO: Click-to-dial service

Browser → Call API → OpenSIPS 

WebSocket API & Events → MI & EVI → SIP

SIP → SIP UA (alice) → SIP UA (bob)

Liviu Chircu - OpenSIPS Project / SIPhub - OpenSIPS Summit 2023 -
Install/Upgrade to \textit{latest} Golang

```
~/.bashrc
============
export GOROOT=$HOME/bin/go
export PATH=$GOROOT/bin:$PATH
```

```
# download go1.20.4.linux-amd64.tar.gz
https://go.dev/doc/install

rm -fr /home/liviu/go
rm -fr /home/liviu/bin/go

tar -C /home/liviu/bin -xzf
~/Downloads/go1.20.4.linux-amd64.tar.gz

go version

go version go1.20.4 linux/amd64
```
Call API - References

- **Source Code**
  [https://github.com/OpenSIPS/call-api](https://github.com/OpenSIPS/call-api)

- **Command Docs**

- **Call API Blog Post**

- **OpenSIPS CallOps Module Docs**
  [https://opensips.org/docs/modules/3.4.x/callops.html](https://opensips.org/docs/modules/3.4.x/callops.html)
Take-Away Message

Build class 5 services on top of OpenSIPS

- Liviu Chircu
  - OpenSIPS Project: www.opensips.org
  - Email: liviu@opensips.org