Using the OpenSIPS b2bua

(back to back user agent)

Peter Kelly / pkelly@sourcevox.com

Who I am

- UK based Open Source VoIP software development and consultancy
- Work with Telco's, CLEC's and ITSP's in the UK and Europe and USA
- We use
  - OpenSIPS
  - Kamailio
  - Asterisk
  - FreeSWITCH
  - RabbitMQ
  - Redis
  - Hadoop
  - Homer
  - Sangoma
  - Dialogic
  - etc!

Reminder - OpenSIPS is a stateless proxy.

```
U 2017/05/01 05:35:25.396853 172.16.19.207:5060 -> 172.16.19.201:5060
INVITE sip:61403430508@us.voxbeam.com:5060;srcip=110.44.126.215 SIP/2.0.
Record-Route: <sip:172.16.19.207;lr;ftag=as0580ff93;did=4a4.a03f1ff4>.
Record-Route: <sip:108.59.2.135;lr;ftag=as0580ff93;did=4a4.9daa4383>.
Via: SIP/2.0/UDP 172.16.19.207:5060;branch=z9hG4bK4737.31af0746.0.
Via: SIP/2.0/UDP 108.59.2.135;branch=z9hG4bK4737.60ead922.0.
Via: SIP/2.0/UDP 108.59.2.134:5060;branch=z9hG4bK4737.74234082.0.
From: "M0430195618005443942" <sip:YOUIWEB@sbc.voxbeam.com>;tag=as0580ff93.
To: <sip:001110161403430508@sbc.voxbeam.com>.
Contact: <sip:caller@108.59.2.134;did=4a4.2d9492f6>.
Call-ID: 50727d94598c9b204307c43542b1d46c@sbc.voxbeam.com.
CSeq: 102 INVITE.
User-Agent: Asterisk PBX.
Max-Forwards: 67.
Date: Mon, 01 May 2017 05:56:18 GMT.
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY.
Content-Type: application/sdp.
Content-Length: 241.
```

Reads and checks incoming request (using cfg script)

Inserts a Record-Route header

Inserts a Via header

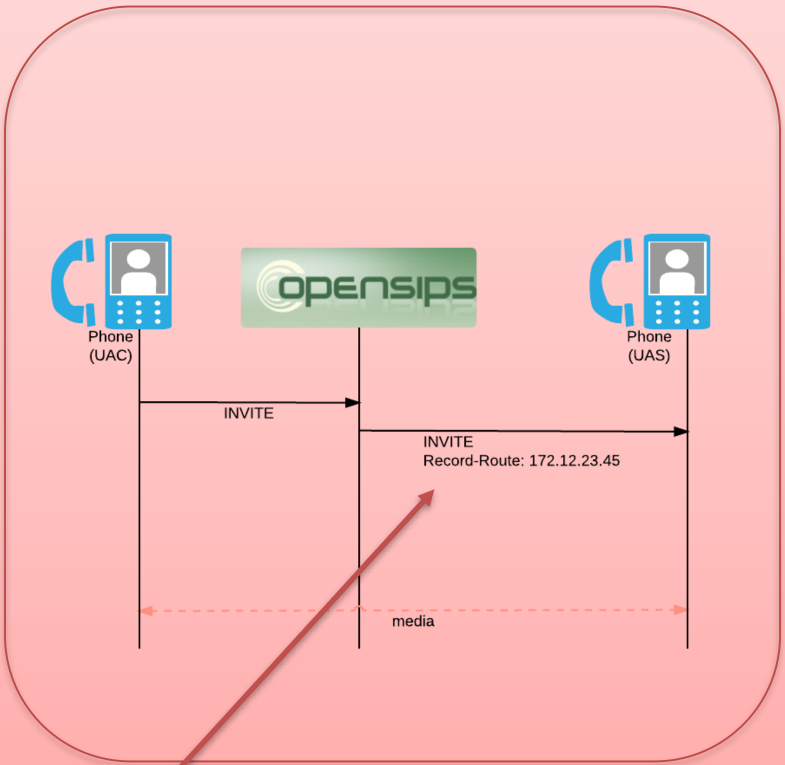
```
U 2017/05/01 05:35:25.398265 172.16.19.201:5060 -> 108.59.2.136:5060
INVITE sip:61403430508@213.230.176.1;srcip=110.44.126.215 SIP/2.0.
Record-Route: <sip:172.16.19.201;lr;ftag=as0580ff93>.
Record-Route: <sip:172.16.19.207;lr;ftag=as0580ff93;did=4a4.a03f1ff4>.
Record-Route: <sip:108.59.2.135;lr;ftag=as0580ff93;did=4a4.9daa4383>.
Via: SIP/2.0/UDP 172.16.19.201;branch=z9hG4bK4737.162c2af.0.
Via: SIP/2.0/UDP 172.16.19.207:5060;branch=z9hG4bK4737.31af0746.0.
Via: SIP/2.0/UDP 108.59.2.135;branch=z9hG4bK4737.60ead922.0.
Via: SIP/2.0/UDP 108.59.2.134:5060;branch=z9hG4bK4737.74234082.0.
From: "M0430195618005443942" <sip:YOUIWEB@sbc.voxbeam.com>;tag=as0580ff93.
To: <sip:001110161403430508@sbc.voxbeam.com>.
Contact: <sip:caller@108.59.2.134;did=4a4.2d9492f6>.
Call-ID: 50727d94598c9b204307c43542b1d46c@sbc.voxbeam.com.
CSeq: 102 INVITE.
User-Agent: Asterisk PBX.
Max-Forwards: 66.
Date: Mon, 01 May 2017 05:56:18 GMT.
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY.
Content-Type: application/sdp.
Content-Length: 241.
```

- Stateless by default!

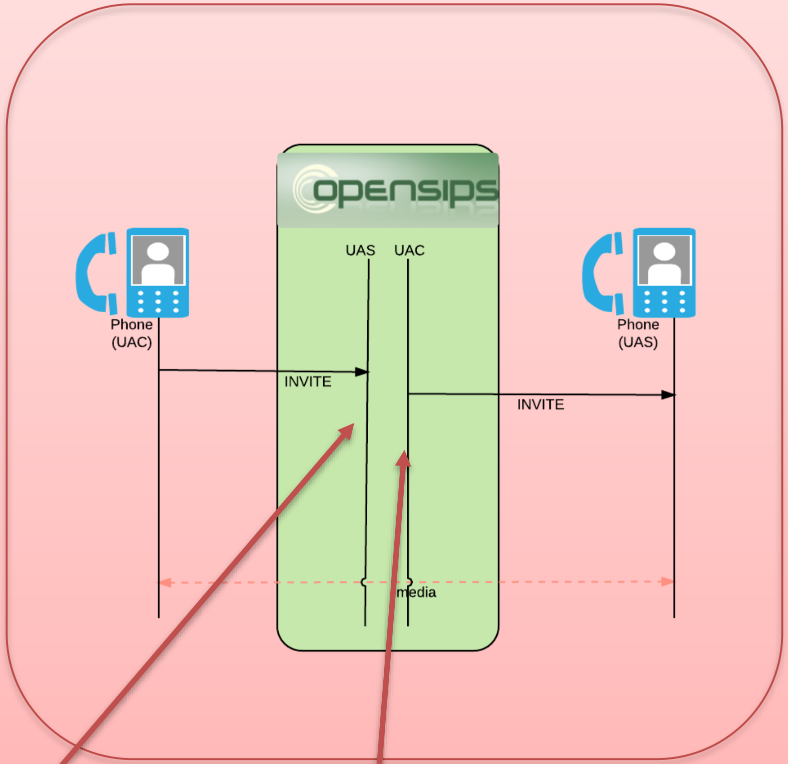
What is a b2bua?

- Not stateless

Standard call



b2bua call



OpenSIPS proxies the call statelessly

A UAS… …and UAC

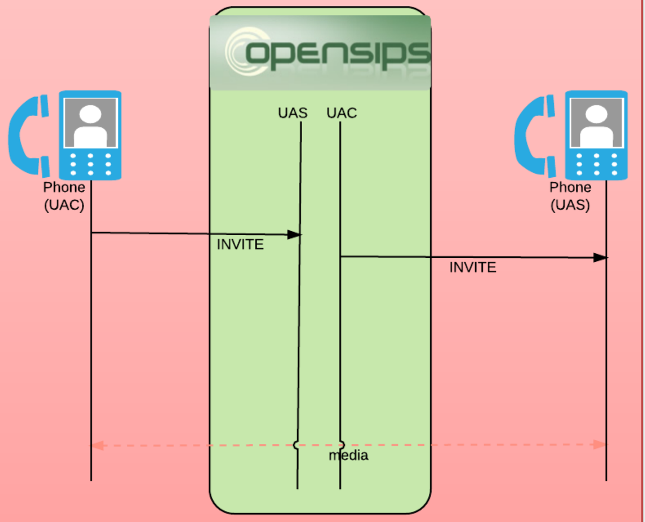
Are created within OpenSIPS.

- Each phone now sees OpenSIPS as an endpoint.
- No longer stateless

Why do I need a b2bua?

- It hides call topology

- It lets OpenSIPS behave like an endpoint (User Agent) to do things like:
    - Initiate reINVITE’s
    - Intercept in dialog requests (e.g. BYE)
    - Perform call transfer requests



Quick slide on topology hiding

- You can actually do this using *two* modules in OpenSIPS:
  - topology_hiding()
  - b2bua

- Benefits of topology hiding:

  - Hides both sides of a call from each other.
    - Useful when running services on a public network to discourage fraud or security breach attempts.

  - Changes the Call-ID of a call.
    - Useful when a customer or carrier may expose parts of their network within this header.

  - Reduces packet size.
    - Useful if you need to keep UDP packet sizes down.

  - Makes OpenSIPS "Look like" an SBC
    - Fixes compatibility problems with operators who have their own interpretation of RFC3261!
    - "Of course, we always send responses to the Contact header"

OpenSIPS b2bua modules

```
#### B2BUA modules
loadmodule "b2b_logic.so"
loadmodule "b2b_entities.so"
```

- 2 modules needed
- b2b_logic - runs UA scenario's
- b2b_entities – handles all the UAS and UAC functions

- Single function call to enable the b2bua!

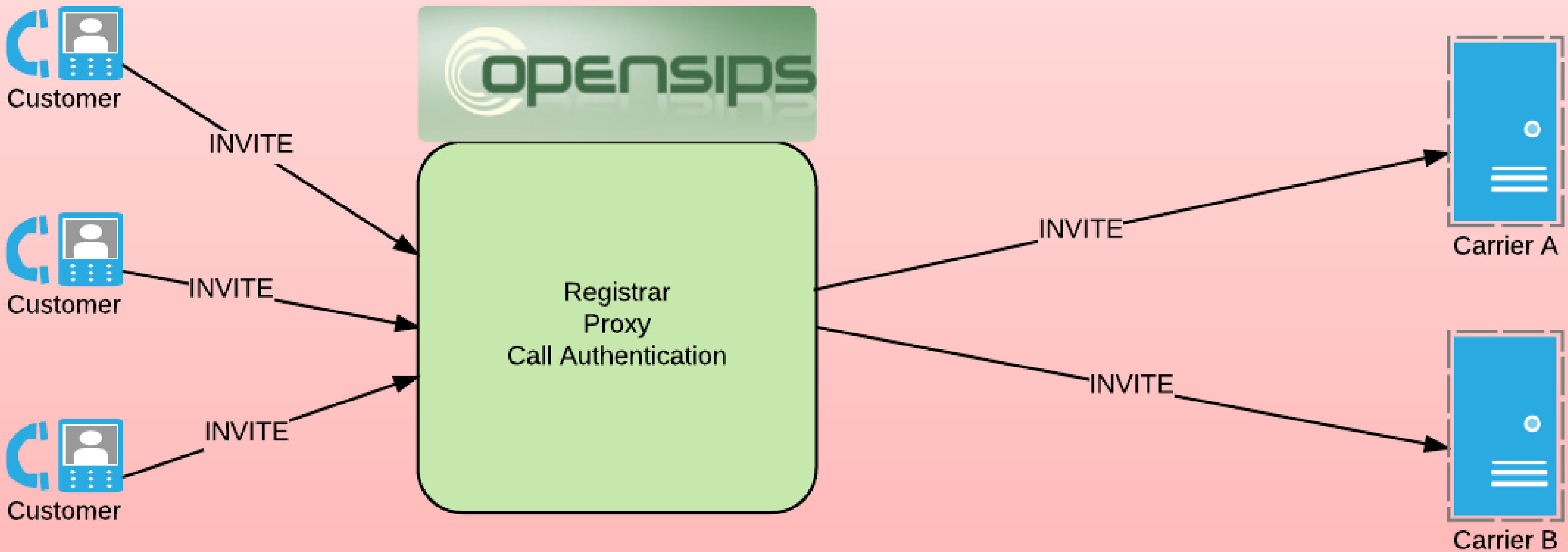
```
b2b_init_request("top hiding");
```

- This immediately tells OpenSIPS to execute the "toplogy hiding" scenario spoken about in previous slides.

- !!! You will now lose all control of further requests/responses. !!!

- The real power of the module happens when you give OpenSIPS a scenario to execute.

Ex1: Reading balance before a call

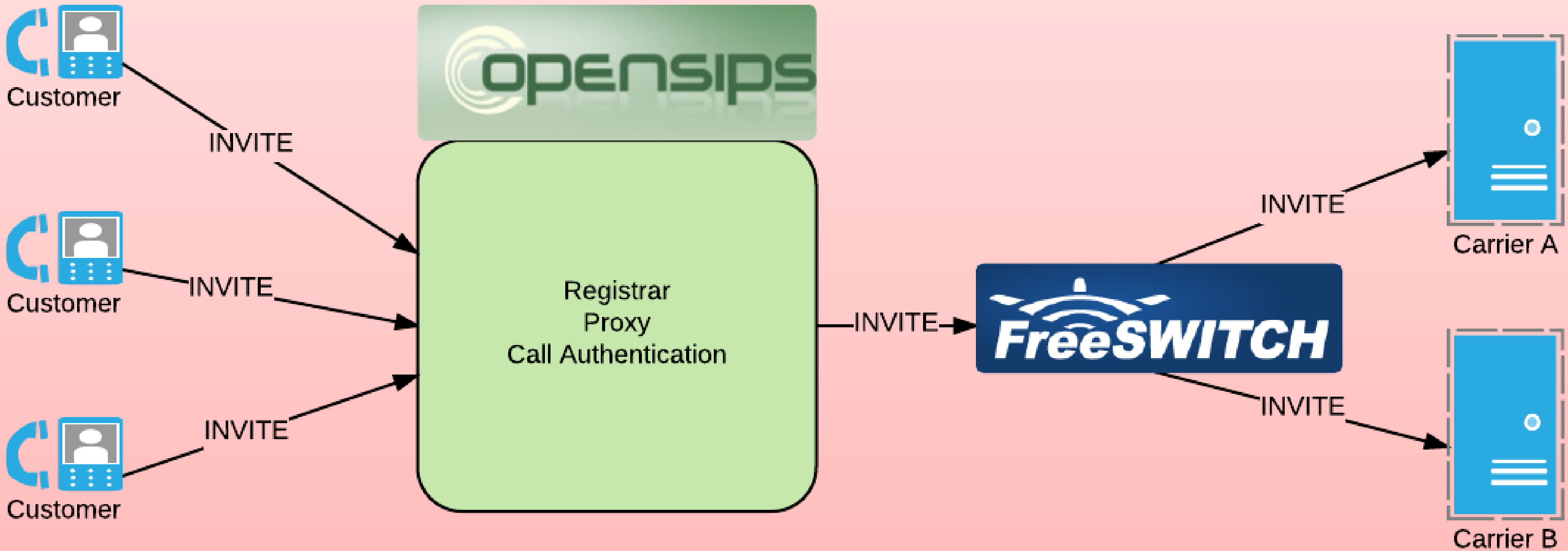


SCENARIO:

- Simple Callingcard service
- Customer sends calls to OpenSIPS,
  - Auth, checks balance, routes call.

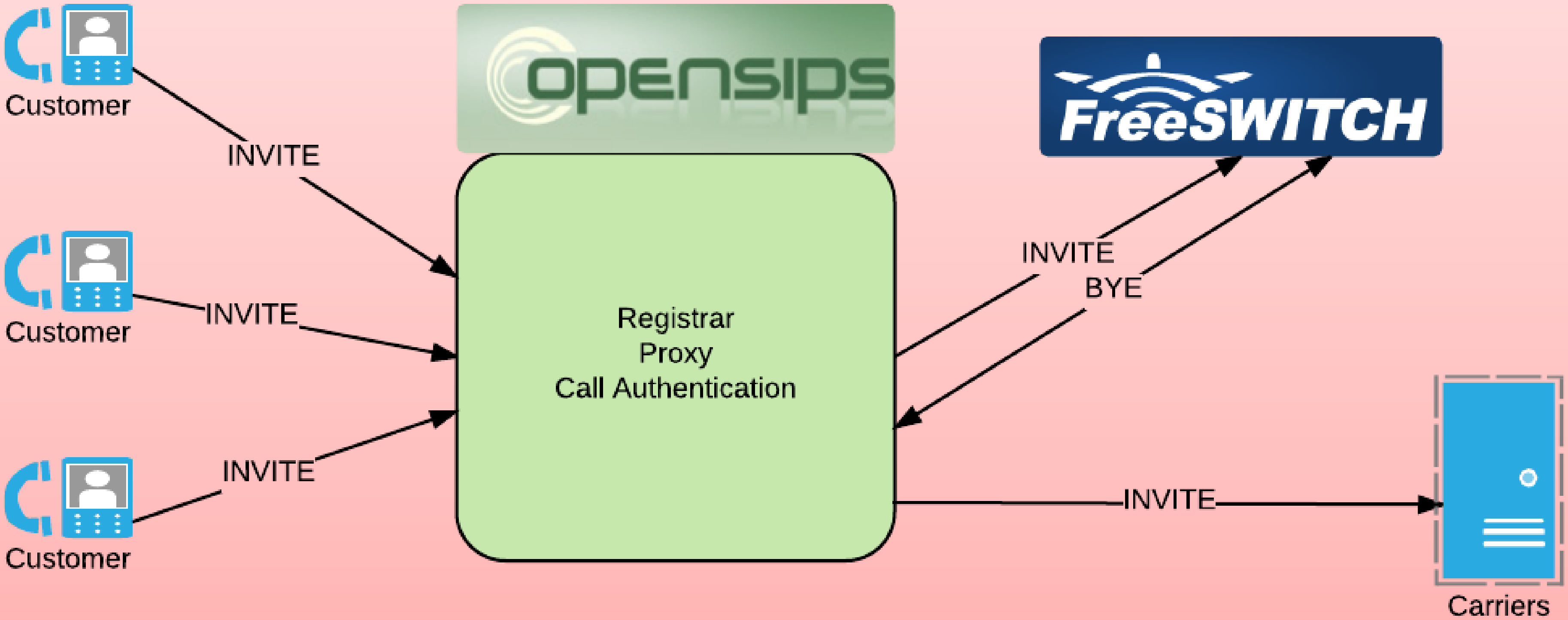
- How can we play a balance to the customer before a call?

Ex1: Reading balance before a call



- Insert media device into the call path?
  - Extra media hop
  - More “mission critical” servers to manage

Ex1: Reading balance before a call



- Atarnative: Use b2bua
    - b2bua lets OpenSIPS make a call to FreeSWITCH first.
    - b2bua intercepts BYE from FreeSWITCH and sends call to Carrier

Ex1: Reading balance before a call: b2bua scenario file

```xml
<?xmlversion="1.0"?>
<scenario id="cc generic" name="Callingcard withFreeSWITCH balance announcement" param="2" type="script">
    <init>
        <bridge>
            <server>
                <id>server1</id>
            </server>
            <client>
                <id>client1</id>
                <type>message</type>
                <destination>
                    <value type="param">1</value>
                </destination>
            </client>
        </bridge>
        <state>1</state>
    </init>
    <rules>
        <request>
            <bye>
                <rule id="1">
                    <condition>
                        <state>1</state>
                        <sender>
                            <type>client</type>
                            <id>client1</id>
                        </sender>
                    </condition>
                    <action>
                        <send_reply>
                            <code>200</code>
                            <reason>OK</reason>
                        </send_reply>
                        <delete_entity/>
                        <bridge>
                            <!--<provisional_media>sip:INJECT_RINGING@108.59.2.143:5080</provisional_media>-->
                            <client>
                                <id>server1</id>
                            </client>
                            <client>
                                <id>client2</id>
                                <destination>
                                    <!--<valuetype="param">3</value>-->
                                    <!--<valuetype="uri">sip:CIDM_fail@10.15.20.58:5080</value>-->
                                    <value type="param">2</value>
                                </destination>
                            </client>
                        </bridge>
                        <state>2</state>
                    </action>
                </rule>
            </bye>
        </request>
    </rules>
</scenario>
```

Initial state of the call:

“bridge a call to client1 using the URI from parameter 1. Set the current state to ‘1’ ”

Rules for call progression:

“If A BYE is received from client1, whilst in state ‘1’, hangup the call to client1 and initiate a call to client2 using the URI from parameter 2. Set the current state to ‘2’ ”

Ex1: Reading balance before a call: call the b2bua module!

```
loadmodule "b2b_logic.so"
loadmodule "b2b_entities.so"
modparam("b2b_logic", "script_scenario", "/usr/local/opensips/etc/opensips/modules/b2b_logic/scenario_cc_generic.xml")
```

Tell OpenSIPS about the new scenario XML file

```
route{


	# Perform initial checks

	# Find account
	$avp(account) = "1234567";

	# Perhaps use dispatcher module to find an available FreeSWITCH machine?
	ds_select_dst("1");
	$avp(fs_uri) = $du;
	resetdsturi();

	# Work out whch carrier should take this call.
	do_routing("1");
	$avp(carrier_ru) = "sip:"+$rU+"@"+$du;

	b2b_init_request("cc_generic", "sip:READ_BALANCE_$avp(account)@$avp(fs_uri)", "$avp(carrier_ru)");

	#b2bua now handles  call setup and all in dialog requests!
```

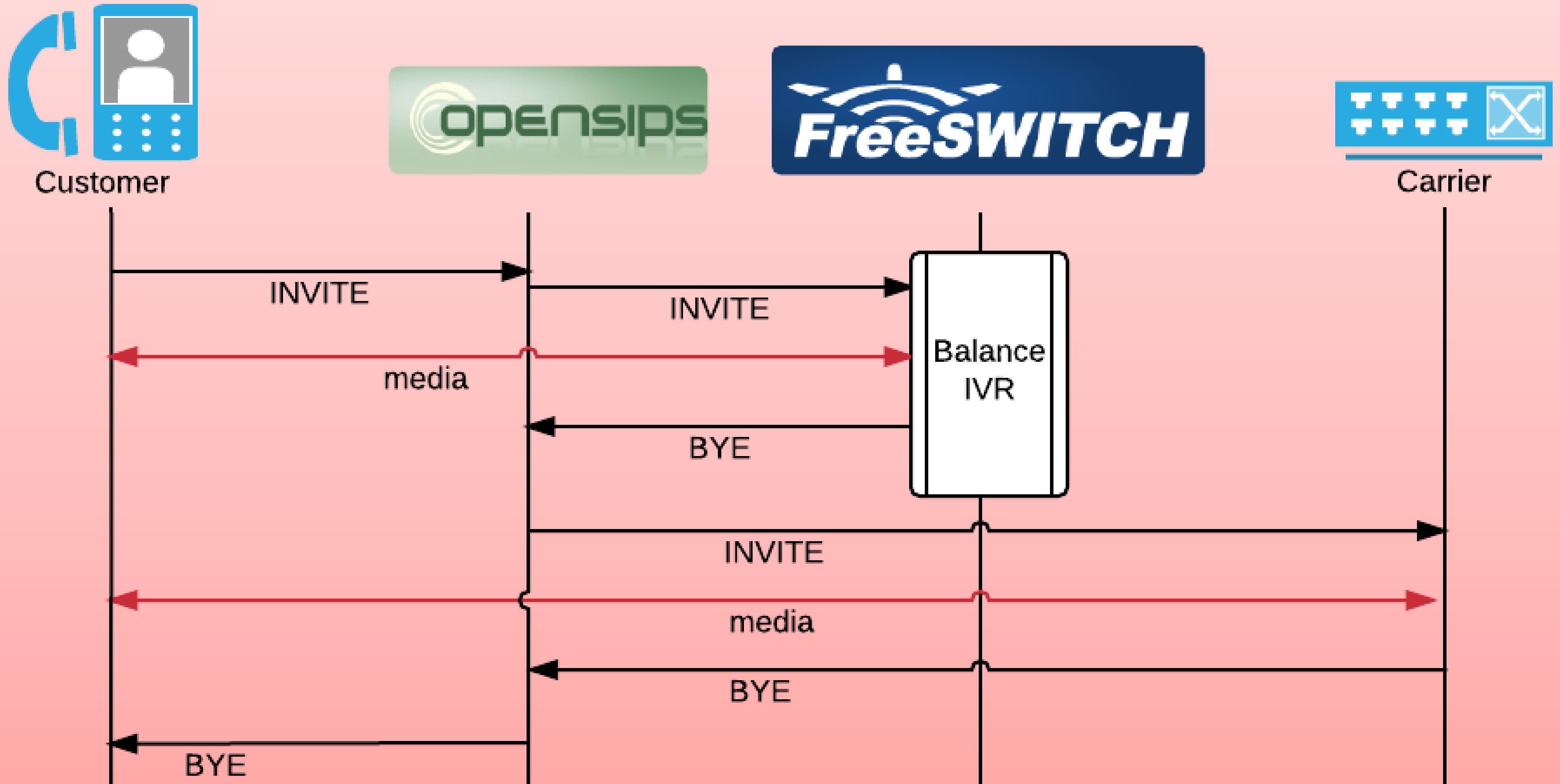
Name of scenario

Parameter 1

Parameter 2

Ex1: Reading balance before a call

- Resulting Call Flow.



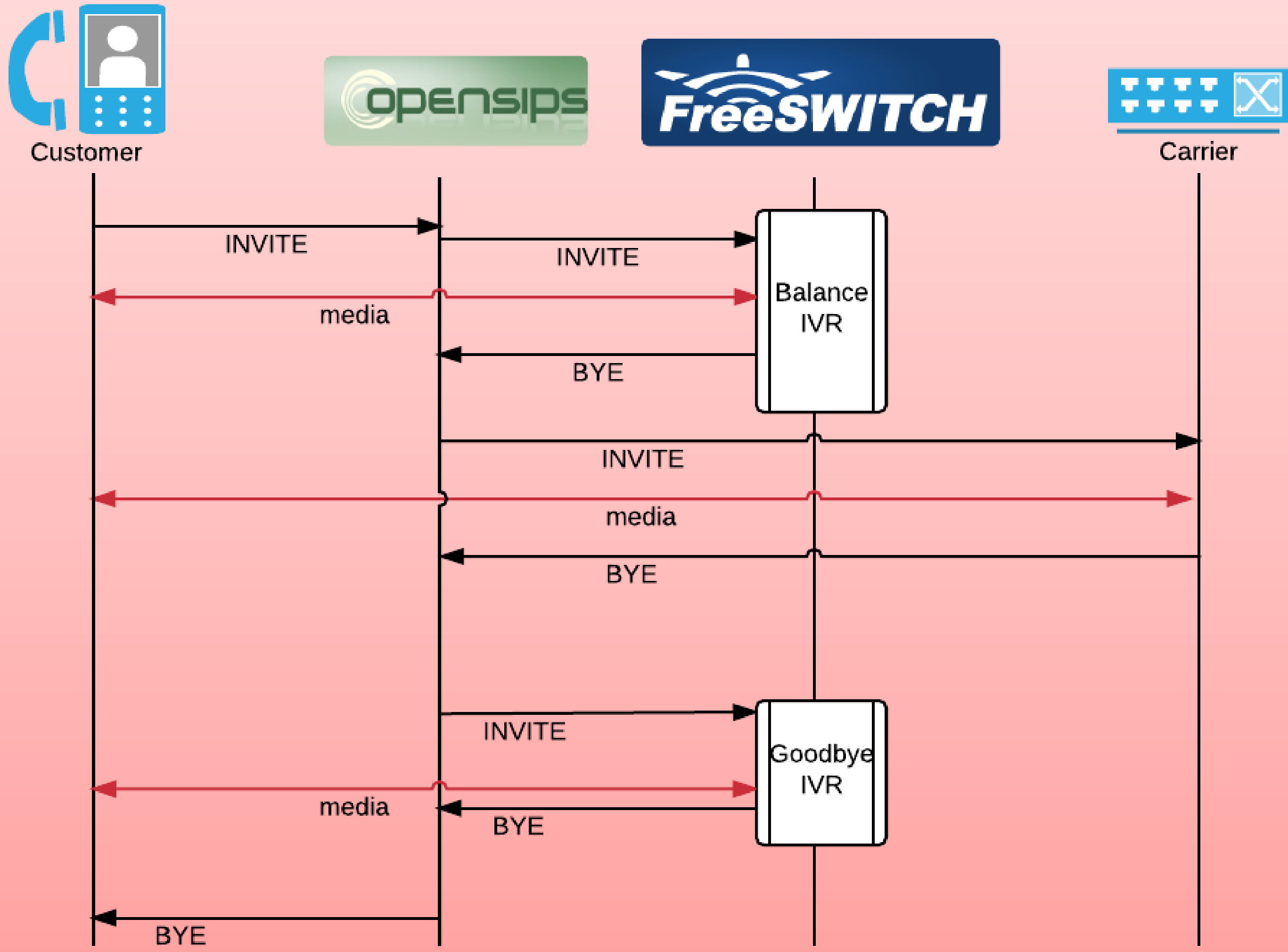
Ex1.5: Reading Balance after a call

- With an extra rule we can also read the balance after a call.

```
<ruleid="2">
    <condition>
        <state>2</state>
        <sender>
            <type>client</type>
            <id>client2</id>
        </sender>
    </condition>
    <action>
        <send_reply>
            <code>200</code>
            <reason>OK</reason>
        </send_reply>
        <delete_entity/>
        <bridge>
            <client>
                <id>server1</id>
            </client>
            <client>
                <id>client1</id>
                <destination>
                    <!--<valuetype="param">3</value>-->
                    <!--<valuetype="uri">sip:LP_opt_mob
                    <value type="param">3</value>
                </destination>
            </client>
        </bridge>
        <state>3</state>
    </action>
</rule>
```

Ex1.5: Reading Balance after a call

- With an extra rule we can also read the balance after a call.



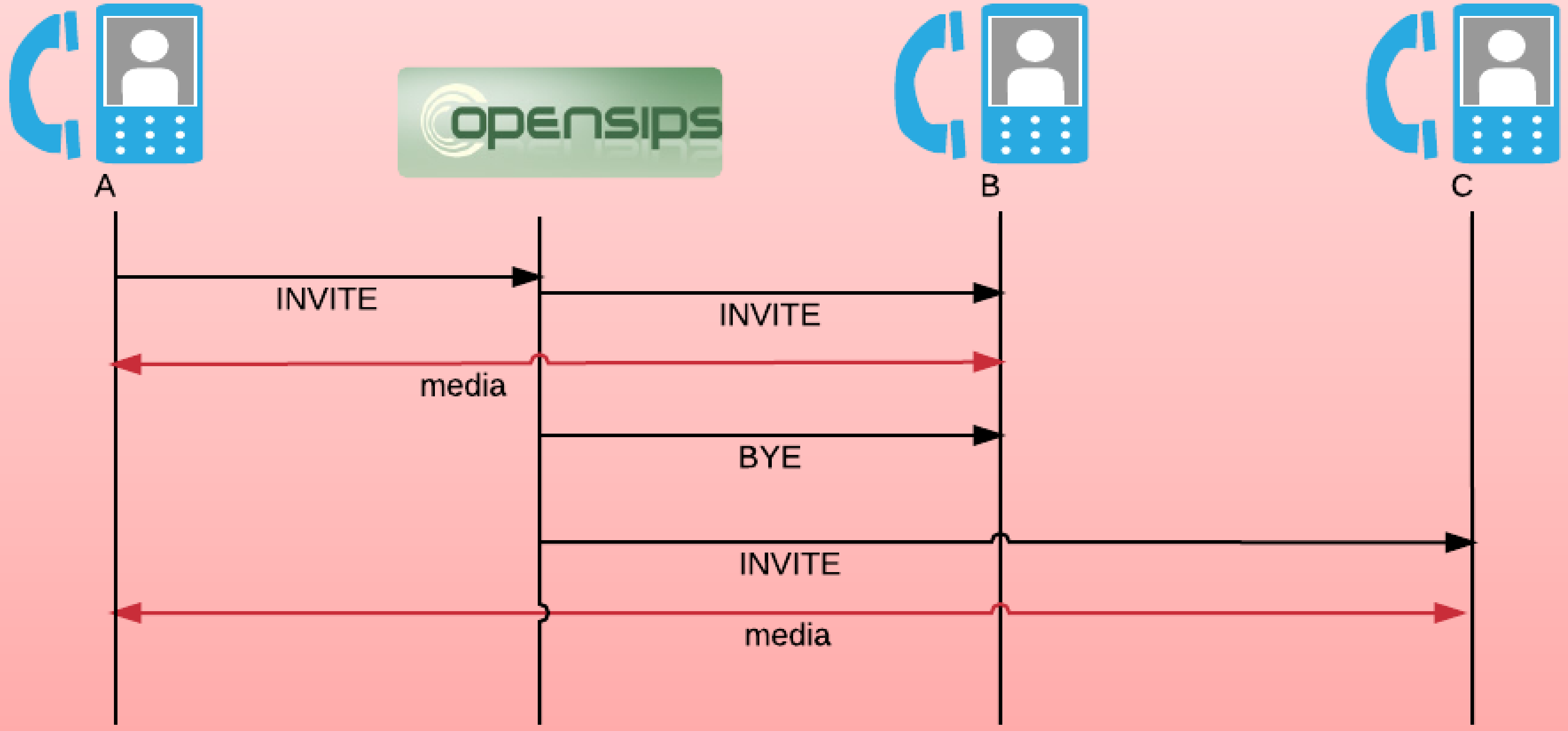
Ex2: using MI to call transfer

- Using the b2bua MI commands a call connected from A>B can be transferred from A>C or B>C using b2bua.

```
#!/bin/bash

# Use b2b_list to find your dialog
opensipsctl b2b_list

#Bridge caller to new URI
opensipsctl b2b_bridge 1020.30 sip:alice@opensips.org

#ALternatively bridge callee to new URI
opensipsctl b2b_bridge 1020.30 sip:alice@opensips.org 1
~
```

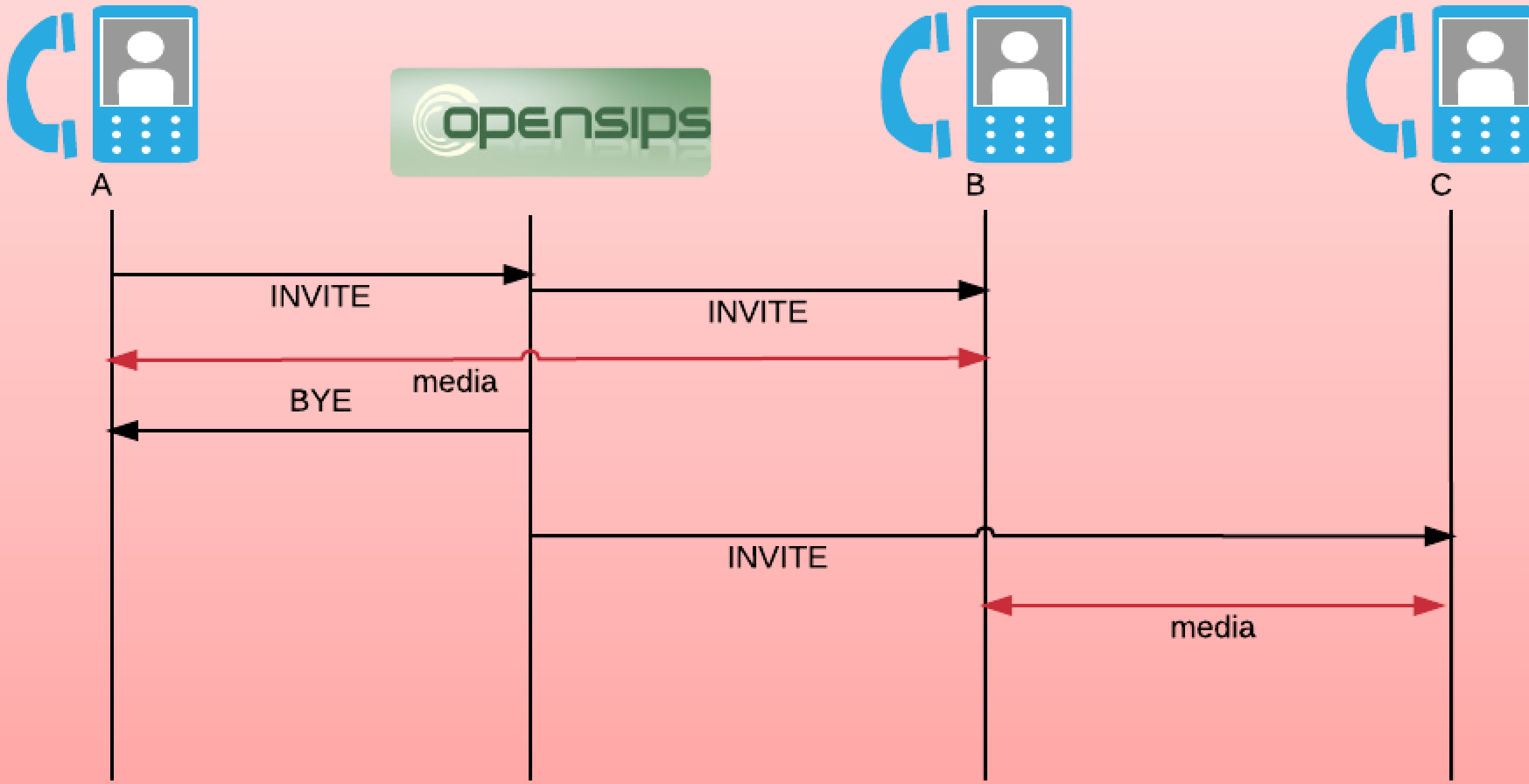
Ex2: using MI to call transfer

- A>C



Ex2: using MI to call transfer

- B>C



Conclusion

- B2bua module is a useful tool if you know how and when to wield it.
- Lets you perform certain UA type actions using OpenSIPS
- Saves complexity in product design
- Can be used to save bandwidth or restrict media farm usage (saving on hardware, licensing costs etc).
- Allows for powerful external application design when combined with MI commands to control call flows.
- Remember! You lose control of the script when you initiate the module

Thank You

Peter Kelly
pkelly@sourcevox.com

Thank You