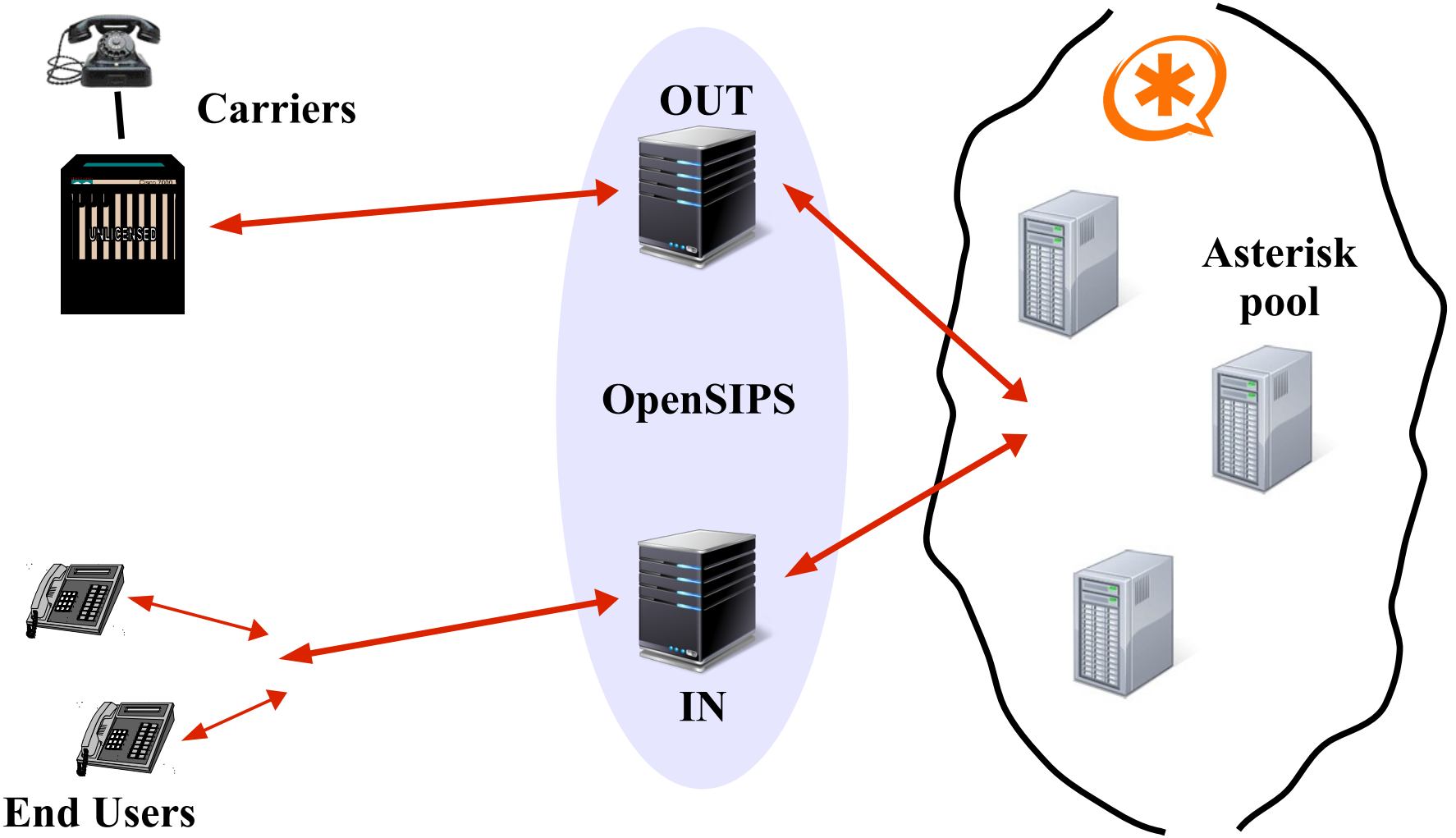
OpenSIPS – a service enabler for Asterisk

Vlad Paiu
OpenSIPS Project Developer
OpenSIPS Solutions


Carriers
UNLICENSED
OUT
OpenSIPS
IN
Asterisk
pool
End Users

Agenda

- Inbound Side
  - Transport Level
  - Security
  - Traffic Shaping
  - IM & Presence
- Outbound Side
  - Carrier Related Services
  - PSTN Routing

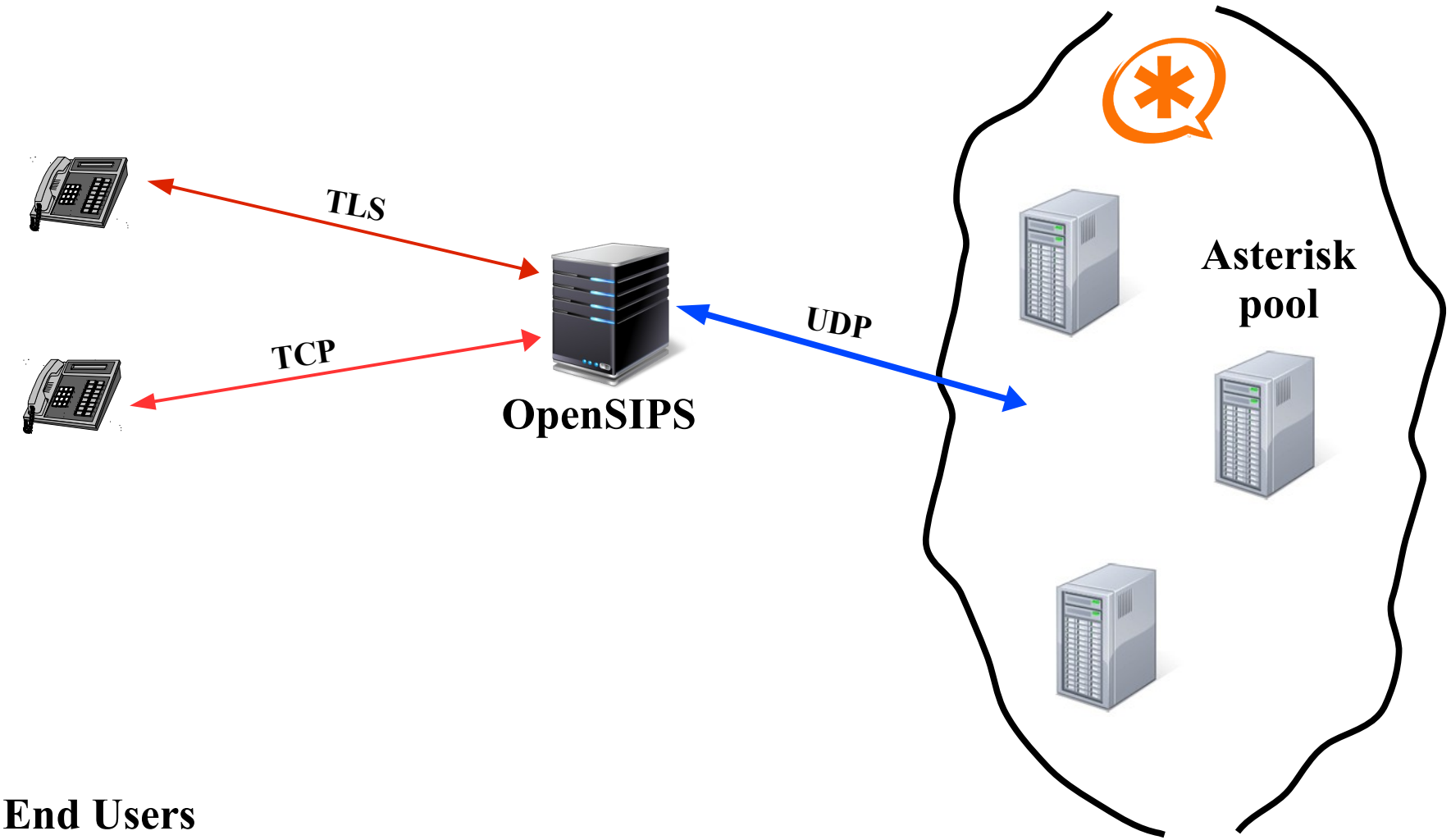
Inbound side

Transport level

- TLS / TCP to UDP conversion
- webRTC
- NAT traversal

Handle them at the border to be able to scale and outsource them from the core side of your service.

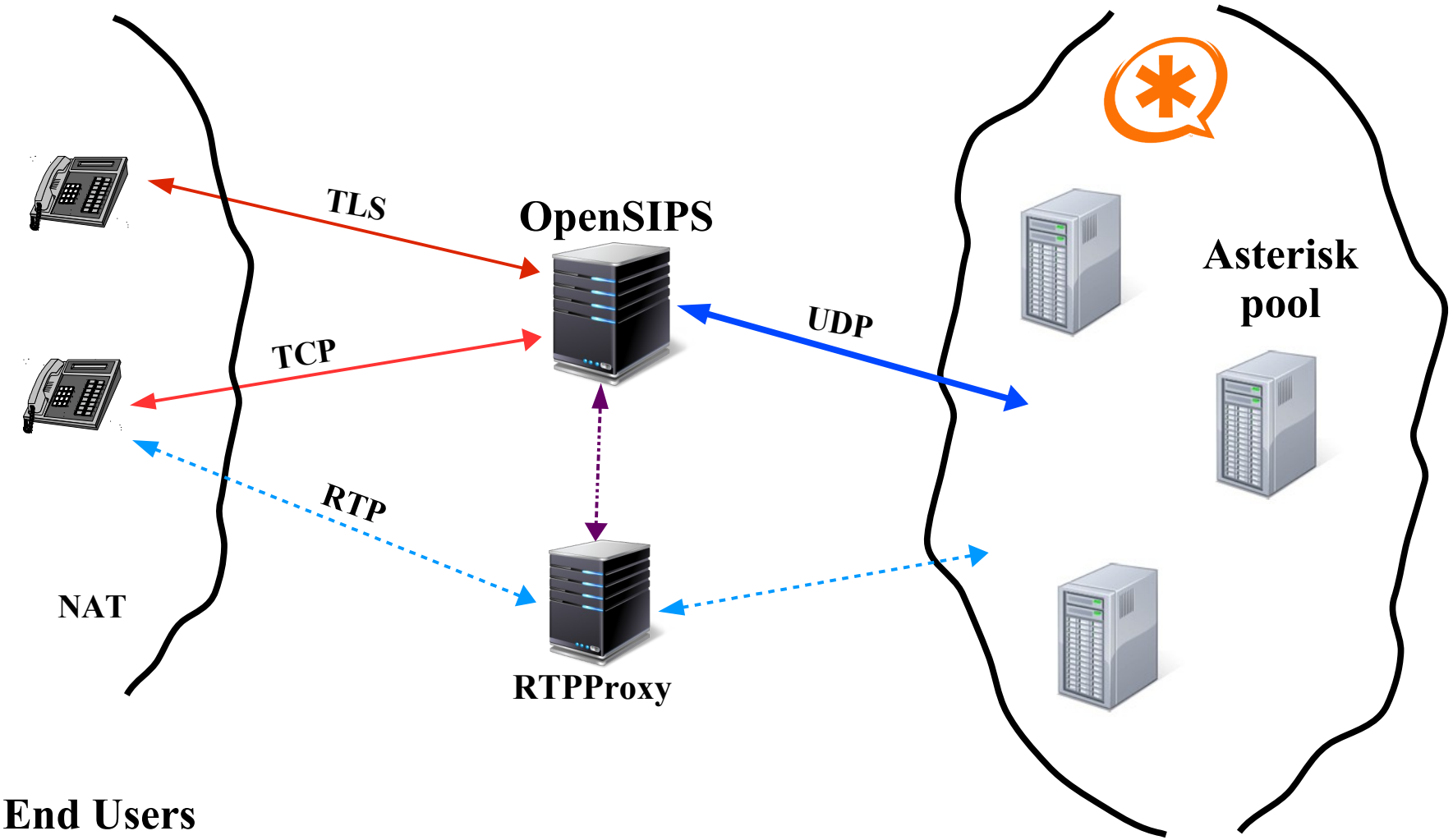
Protocol Conversion



End Users

Protocol Conversion

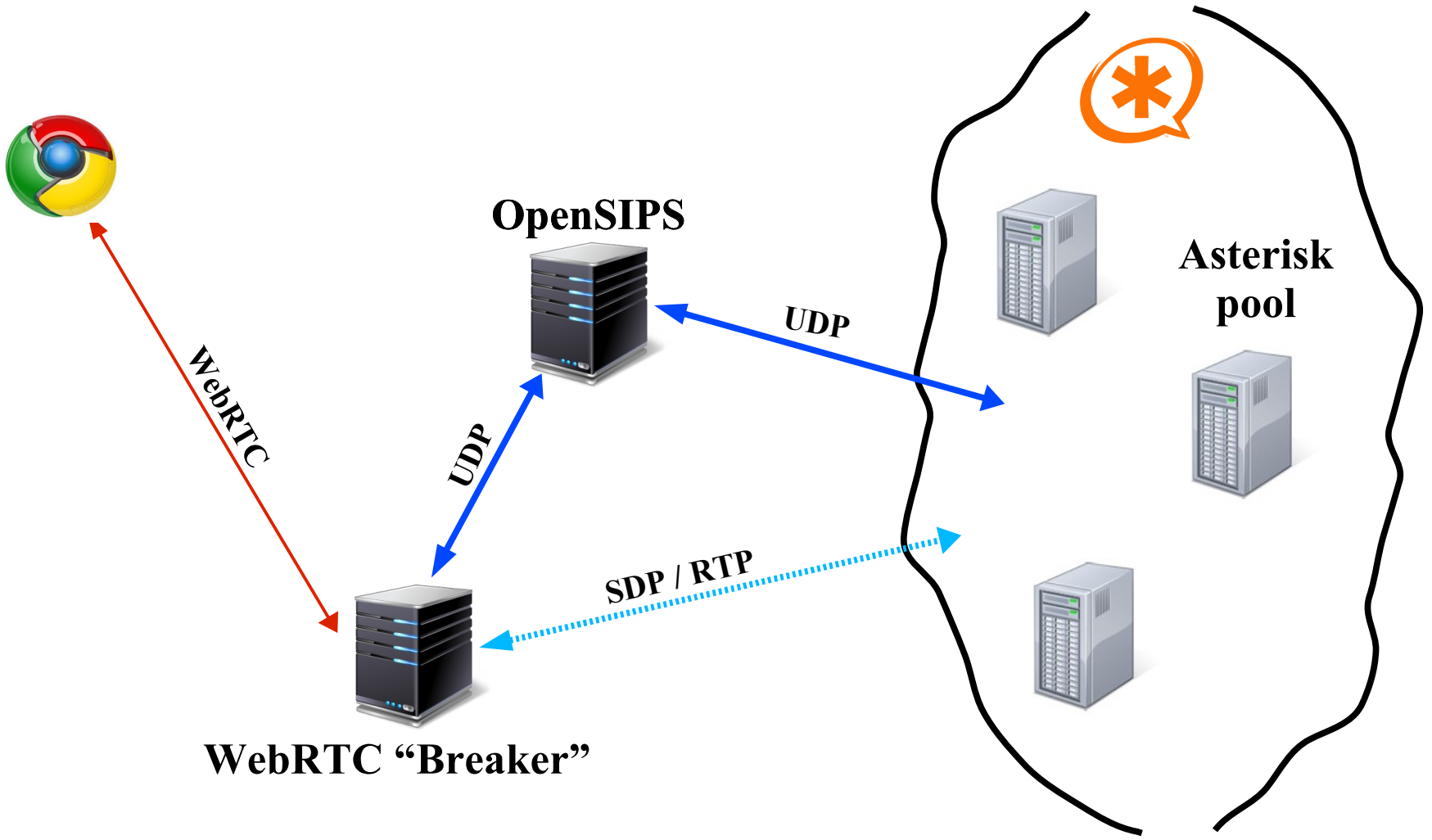
```
disable_tcp=no
listen=udp:CORE_IP:5060 use_children 10
listen=tcp:PUBLIC_IP:5060
tcp_children = 10
…..
modparam("rr", "enable_double_rr", 1)
…..
if (!has_totag() && dst_ip == PUBLIC_IP) {
        force_send_socket(udp:CORE_IP:5060);
        $du = “sip:ASTERISK_IP:5060”;
        record_route();
        t_relay();
        exit;
}
if (has_totag()) {
        loose_route();
        t_relay();
}
```


TLS
OpenSIPS
TCP
UDP
Asterisk
pool
RTP
NAT
RTPProxy
End Users

```
loadmodule “nathelper.so”
loadmodule “usrloc.so”
loadmodule “registrar.so”
loadmodule “rtpproxy.so”
modparam("usrloc", "nat_bflag", “NAT_BFLAG”)
modparam("nathelper", "natping_interval", 30)
modparam("nathelper", "ping_nated_only", 1)
…..
route[NAT_DETECTION] {
        force_rport();
        if ( nat_uac_test(“19”) ) {
                # Rewrite contact with source IP of signalling
                if (is_method(“REGISTER”)) {
                        setbflag(NAT_BFLAG);
                        fix_nated_register();
                } else {
                        fix_nated_contact();
                }
        }
}
```

```
route[NAT_PROCESSING] {
        if (isbflagset(NAT_BFLAG) {
                if (is_method(“INVITE”) && has_body("application/sdp")) {
                        rtpproxy_offer();
                        t_on_reply(“REPLY_PROCESSING”);
                }
        }
}

reply_route[REPLY_PROCESSING] {
        if (has_body("application/sdp")) {
                rtpproxy_answer();
        }
}
```


OpenSIPS
Asterisk
pool
UDP
WebRTC
UDP
SDP / RTP
WebRTC "Breaker"

Security

- DoS detection and protection
- Fraud detection

Security issues must be handled as soon as possible, before reaching the core service.

DOS Protection

- Pike Module
- Detects all types of floods
  - Not just SIP requests
- Provides the detection mechanism
  - Banning offenders is to be done externally
    - Fail2Ban
    - Event Route

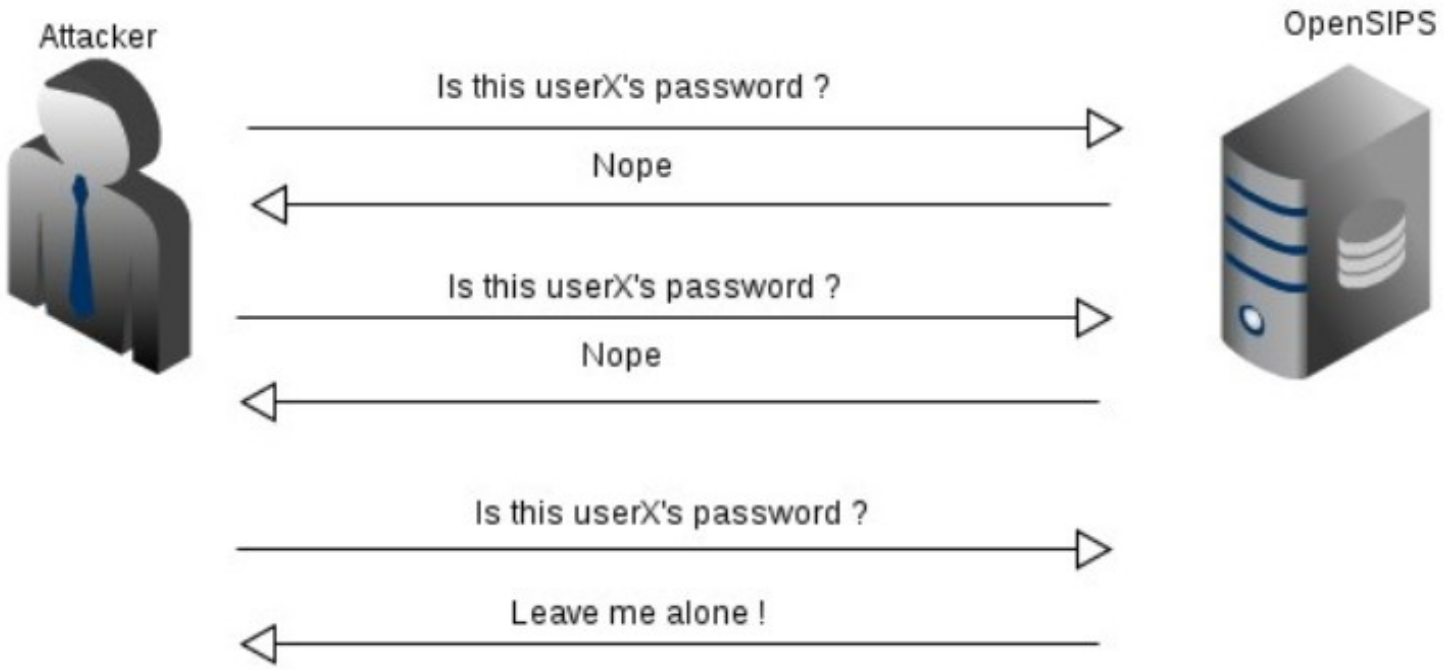
Pike Module Usage

```
loadmodule “pike.so”
loadmodule “event_route.so”
loamodule “exec.so”
modparam("pike", "sampling_time_unit", 2)
modparam("pike", "reqs_density_per_unit",60)
modparam("pike","check_route", "filter_pike")
modparam(“exec”, “async”, “1”)

route[filter_pike]{
        if ( route(IS_SRC_TRUSTABLE))
                drop;
        /* all other IPs are checked */
}
event_route[E_PIKE_BLOCKED] {
        fetch_event_params("$var(ip)”);
        exec_avp(“/usr/local/sbin/iptables_block_ip.sh $var(ip)”);
}
```

- **Passive Attacks**
  - Make use of information from the SIP systems
    - Addressed by topology hiding & transport encryption
- **Active Attacks**
  - Affect SIP systems operation / Alter system resources

Dictionary Attacks

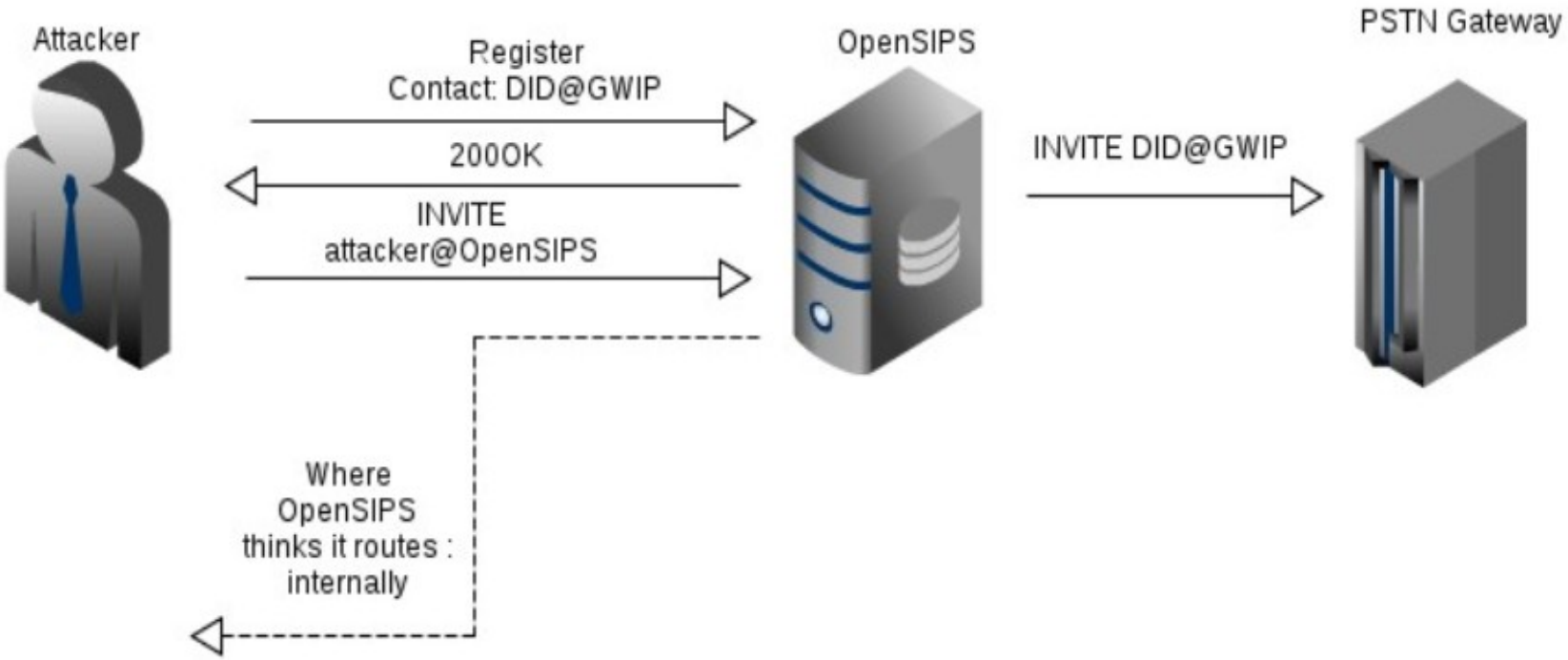


Dictionary Attacks

```
www_authorize("","subscriber");
switch ($retcode) {;
        case -3: # stale nonce
        case -2: # invalid passwd
        case -1: # no such user
                if ( cache_fetch("local","authF_$si",$avp(failed_no)) ) {
                        if ( $(avp(failed_no){s.int}) >= 20 ) {
                                xlog("SCRIPT: SECURITY ALERT: 20 failed auth from $si\n");
                                # TAKE ACTION HERE
                                exit;
                        }
                        cache_add("local","authF_$si",1,60);
                } else {
                        cache_store("local","authF_$si","1",60);
                }

                # continue challenging logic
```

Register Poisoning



Register Poisoning

```
$var(i) = 0;
while( $(ct[$var(i)])!=NULL ) {
    $var(host) = $(ct[$varv(i)]{nameaddr.uri}{uri.host});
    if (${var(host){ip.resolve)} == “GWIP” ) {
        xlog("SECURITY ALERT: $si registering $var(host)\n");
        send_reply("476", "Contact Unacceptable );
        exit;
    }
    $var(i) = $var(i) + 1;
}
```

Hijacked Accounts

- Actual stolen accounts
  - Weak passwords
- Badly configured phones
  - Unchanged default passwords
- Exploits in the phone software
  - Old Firmware

Detect fraud as anomalies in user's dialing pattern

- Pattern for the volume of the calls
- Pattern for the daily schedule of the calls
- Pattern for the usual destination zones of the calls

fraud_detection Module

- fraud_detection OpenSIPS module
  - Present in future OpenSIPS release, available in trunk for testing
  - DB provisioned fraud profiles and assign them to users
    - Total calls
    - Calls per minute
    - Concurrent calls
    - Sequential call attempts
    - Call duration
  - Per prefix and time-based rules
  - Warning and critical thresholds that will trigger event routes
- check_fraud("$avp(username","$rU","$avp(fraud_profile")
  - http://www.opensips.org/html/docs/modules/1.12.x/fraud_detection

Traffic shaping

- Concurrent call limitation
- Calls per second limitation

Concurrent Call Limits

- Dialog module
- Predefined profiles
  - Per account
  - Per asterisk box
  - Per gateway

Concurrent Call Limits

```
loadmodule “dialog.so”
modparam("dialog", "profiles_with_value", "caller")

if (is_method(“INVITE”) && !has_totag()) {
      create_dialog();
      set_dlg_profile(“caller”,”$fU”);
      get_profile_size(“caller”,"$fU”,"$var(total_calls)”));
      if($var(total_calls) > $avp(account_max_concurrent) ) {
          xlog(“SCRIPT:CCLIMIT: Max cc  limit reached for $fU \n”);
          t_reply(“500”,”Limit Reached”);
          exit;
      }
}
```

Calls per Second Limits

- Ratelimit Module
- Dynamic profiles
  - Per account
  - Per asterisk box
  - Per gateway

Calls per Second Limits

```
loadmodule “ratelimit.so”
modparam(“ratelimit”, “timer_interval”, 1)



if (!rl_check(“user_$fU”,"$avp(user_cps_limit)”) {
      xlog(“SCRIPT:CPS: User $fU has exceeded the CPS limit \n”);
      t_reply(“500“,”Limit exceeded”)
      exit;
}
```

Realtime Value Storage

- By default, in OpenSIPS internal memory
- Can be configured to external NoSQL database
  - Allowing for geo-distributed consistent limitations

```
loadmodule “cachedb_mongodb.so”
modparam(“dialog”, “cachedb_url”, “mongodb://IP:PORT/Dialog.Limits”)
modparam(“dialog”, “profiles_with_value”, “INcalls/s”)
```

SIP Simple

- IM end-2-end, IRC-like chatting
- Presence, Dialog info, MWI, BLF, SLA, XCAP
- Third-party integration via presence

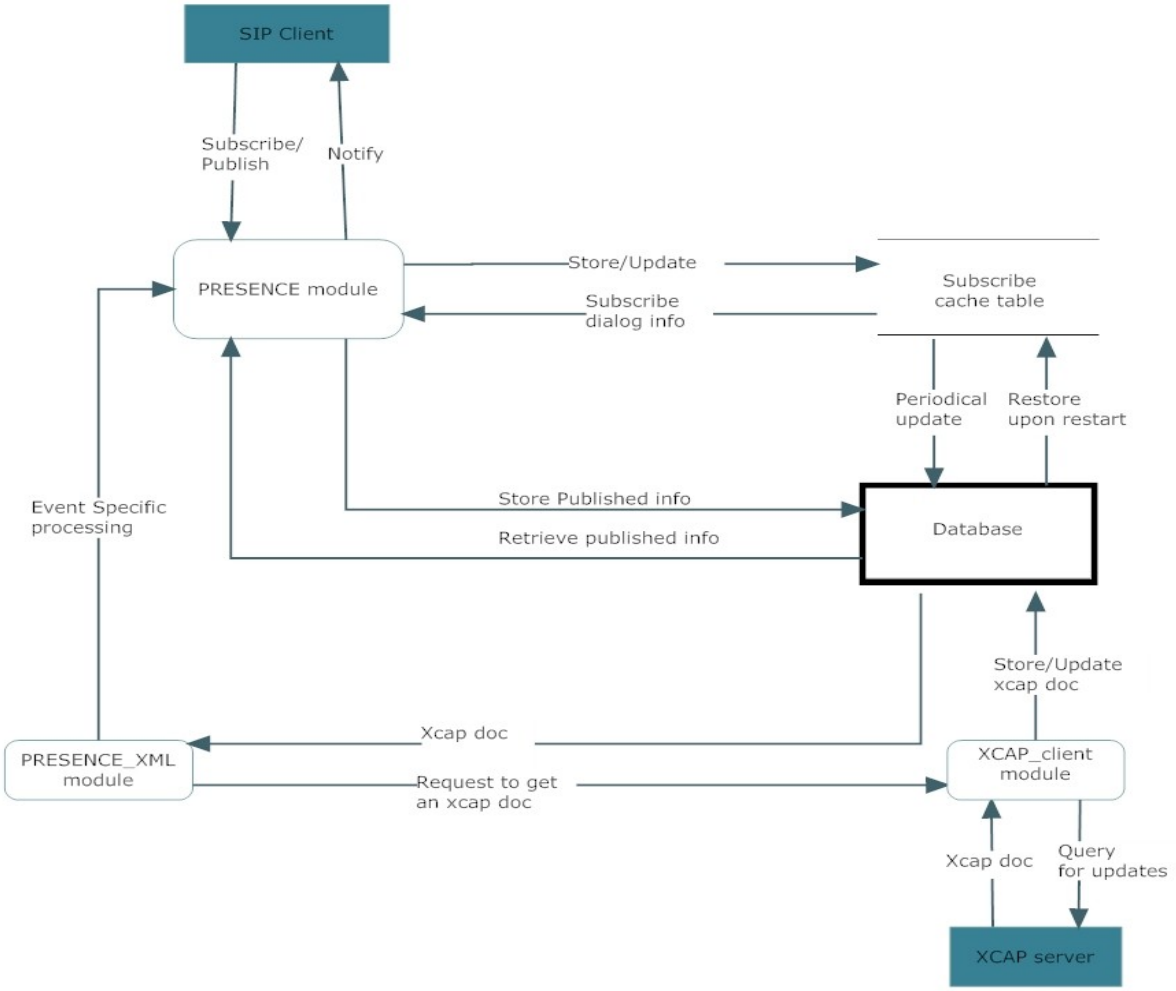
OpenSIPS acts as Presence Server and handles the presence/IM related traffic, transparently for the other services hosted by Asterisk.

- Handled via the SIP MESSAGE method
- OpenSIPS delivers the messages
  - On the spot, if the user is available
  - Stores the message & sends it when the destination becomes available

IM – msilo module

```
loadmodule “msilo.so”
modparam("msilo", "db_url",
"mysql://user:passwd@host.com/dbname")


…..
if (is_method(“MESSAGE”)) {
      if (!lookup(“location”)) {
            m_store(“$rU”) );
            t_reply(“202“,”Accepted for later delivery”);
            exit;
      }
      t_relay();
      exit;
}
…..
If (is_method(“REGISTER”)) {
      save(“location”);
      m_dump();
}
```


SIP Client
Subscribe/
Publish
Notify
PRESENCE module
Store/Update
Subscribe
dialog info
Subscribe
cache table
Periodical
update
Restore
upon restart
Event Specific
processing
Store Published info
Retrieve published info
Database
Store/Update
xcap doc
Xcap doc
PRESENCE_XML
module
Request to get
an xcap doc
XCAP_client
module
Xcap doc
Query
for updates
XCAP server

Basic Presence Server

```
modparam("presence|xcap","db_url","mysql://user:pw@host/db")
modparam("presence","server_address","sip:presence@MY_IP:5060")
....
....
if (is_method("PUBLISH|SUBSCRIBE")) {
        route(handle_presence);
        exit;
}
....
route[handle_presence]{
        is_method("PUBLISH")) {
                handle_publish();
        } else if (is_method("SUBSCRIBE")) {
                handle_subscribe();
        }
        exit;
}
```

Outbound side

Carrier oriented services

- Topology Hiding
- LNP dipping
- CNAME lookup

Topology Hiding

- Hide your internal topology
  - Both from the carriers and the clients
- Hide the used carriers from your clients
- Relies on the dialog module
- topology_hiding()
  - “U” flag parameter to pass caller username
  - “C” flag parameter to also hide the callid
  - Has the ability to preserve end-to-end functionality by passing various Contact parameters

LNP / CNAME Lookups

- OpenSIPS exposes various ways of integrating with such external services
- Most commonly used options include
  - Direct database query
  - HTTP query
  - SIP Redirect

HTTP Query

```
loadmodule “rest_client.so”
modparam(“rest_client”, “connection_timeout”, 1)
modparam("rest_client", "curl_timeout", 1)

if (!rest_get("http://URL/?no=$rU", "$avp(reply)", "$var(ct_type)",
"$var(rcode)") || $var(rcode) != 200) {
    xlog(“SCRIPT:LRN: No reply from the LRN query for $rU \n”)
} else {
    if ($var(ct_type) == “text/plain”) {
        xlog(“LRN reply is $avp(reply) \n”);
    } else if ($var(ct_type) == “application/json”) {
        $json(rpl) := $avp(reply);
        xlog(“LRN repliy is $json(rpl/lrn) \n”);
    }
}
```

SIP Redirect

```
route[RELAY_TO_LRN] {
    $du = "sip:LRN_IP:LRN_PORT";
    $T_fr_timeout=2;
    t_on_failure("lrn_reply");
    t_relay();
}
failure_route[lrn_reply] {
    if (t_check_status("302")) {
        $var(lrn_rpl) = $(<reply>ct.fields(uri){param.value,rn});
        if ($var(lrn_rpl) == NULL || $var(lrn_rpl) == "") {
            $var(lrn_rpl) = $rU;
            xlog("SCRIPT:LRN:DBG: $rU is not ported \n");
        } else {
            xlog("SCRIPT:LRN:DBG: $rU translates to $var(lrn_rpl) \n")
            $rU = $var(lrn_rpl);
        }
    }
    # continue processing the call
}
```

PSTN Routing

- Prefix based routing / LCR
- In-memory routing able to handle millions of rules with thousands of gateways.

DROUTING Module

- Routing Info stored in DB and loaded at startup
- In order or weight-based routing to the destinations assigned to a prefix
- Gateway automatic failure detection and re-enabling
- Inbound & Outbound routing capabilities
- Easy Script integration

DROUTING Module - Outbound

```
loadmodule “drouting.so"
modparam(“drouting”, “gw_id_avp”, ‘$avp(gw_id)’)
....
....

if (do_routing(“0“,”W”)) {
        xlog(“SCRIPT:ROUTING: Routing $rU to GW $avp(gw_id) \n”);
        t_on_failure(“GW_FAILURE”);
        $T_fr_timeout=2;
        $T_fr_inv_timeout=20;
        t_relay();
        exit;
} else {
        send_reply(“500”,”PSTN Not Available”);
        exit;
}
```

DROUTING Module - Outbound

```
failure_route[GW_FAILURE] {
    if ( (t_check_status("408" ) && t_local_replied("all")) || t_check_status("[56][0-9]
    [0-9]") {
        xlog("SCRIPT:ROUTING: GW $avp(gw_id) has failed \n");
        dr_disable();
        if (use_next_gw()) {
            xlog("SCRIPT:ROUTING: Re-routing $rU to GW $avp(gw_id);
            t_on_failure("GW_FAILURE);
            t_relay();
            exit;
        } else {
            send_reply("500","PSTN Not Available");
            exit;
        }
    }
}
```

DROUTING Module - Inbound

```
loadmodule “drouting.so"
modparam(“drouting”, “gw_id_avp”, ‘$avp(gw_id)’)
…..
…..


if (is_from_gw("","i") {
     xlog(“SCRIPT:INBOUND: New call from GW $avp(gw_id) \n”);
     if (do_routing(“1“,”LC”)) {
          xlog(“SCRIPT:DIDS: Received call on DID $rU \n”);
          # do DID processing
     } else {
          send_reply(“Unknown DID \n”);
          exit;
     }
}
```

Quality Routing

- Runs on top of the DROUTING module
- SIP defined quality parameters :
  - PDD
  - AST
  - ACD
  - ASR
- GW thresholds that allow
  - Reordering within a prefix rule, based on quality
  - Complete removal from routing

Quality Routing

- **Integrated with Event Interface**
  - Get notifications when certain parameters exceed thresholds
- **Transparent from scripting point of view**
- **Work in progress, will be present in future OpenSIPS release**

Thank you for your attention
You can find out more at www.opensips.org
vladpaiu@opensips.org
www.opensips-solutions.com

Questions are welcome