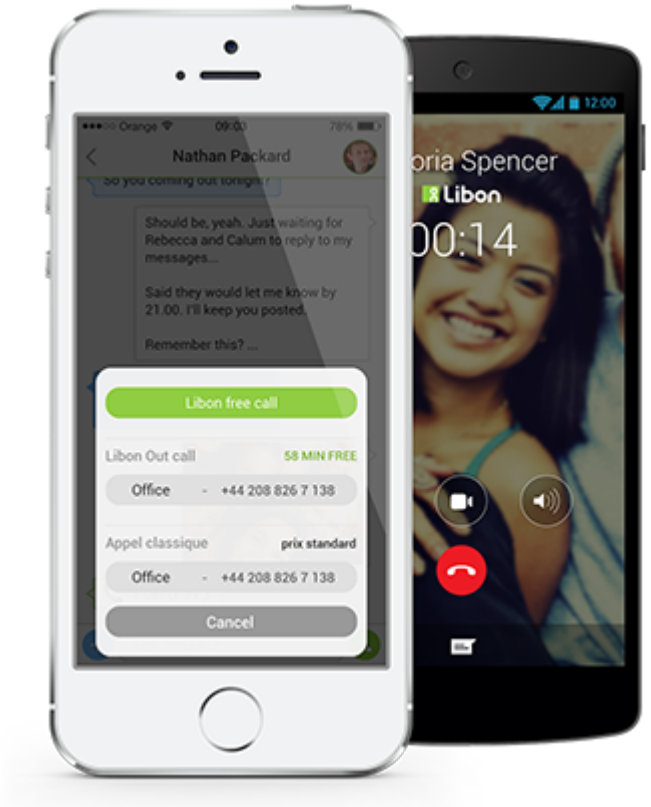
Kamailio
in a Mobile World

Federico Cabiddu
Senior VoIP Developer

About me

- Working in **VoIP** since 2001
- Working with **OpenSER/Kamailio** since 2005
- **Libon Voice Team** Tech Coordinator

What is Libon?



app available on

VoIP and VoIP out calls

App2app calls and calls to landlines and mobiles in more than 100 destinations. Free app to app calls.

Instant messages

Instant messaging with Libon and non-Libon users (OpenChat).

Customized voicemail

Create customized greetings for each one of your contacts or groups of contacts.

Contents

- Constraints of a mobile VoIP application
- A commonly used approach
- Multidevice scenarios
- The TSILO module
- Working example

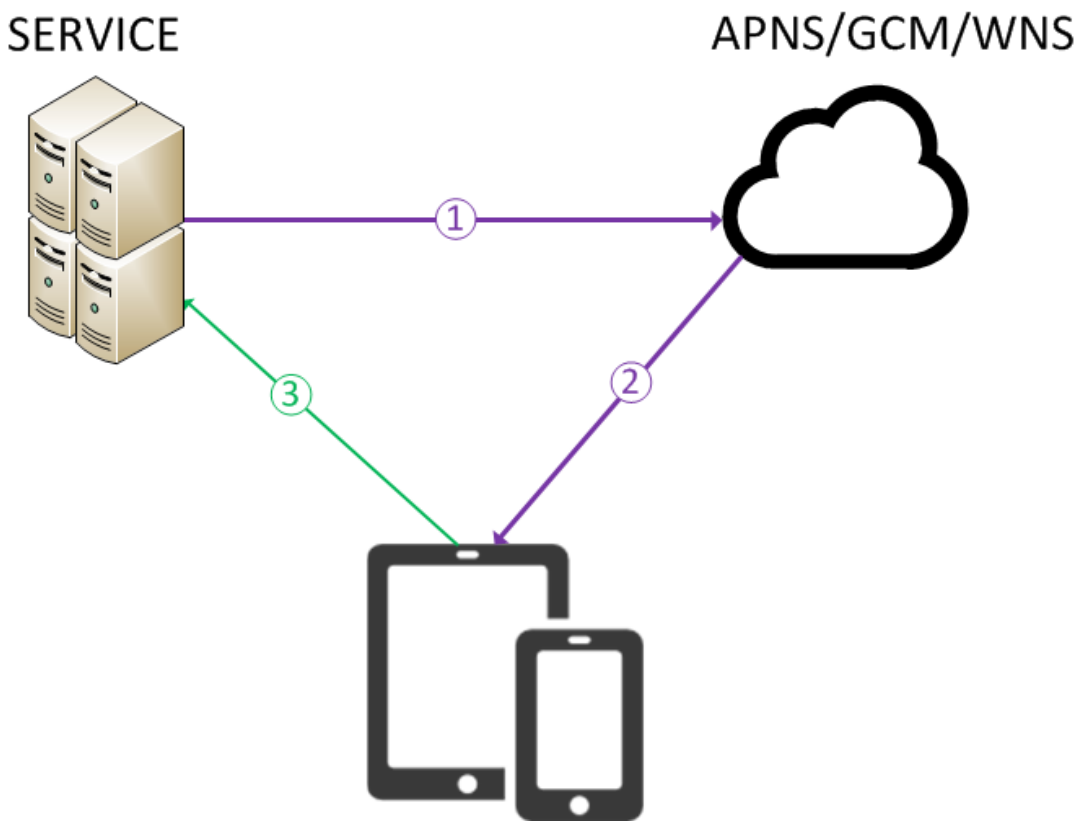
Mobile VoIP apps constraints

- Power consumption
- Data consumption
- Socket unreliability
- Constructor constraints: iOS background mode(s)

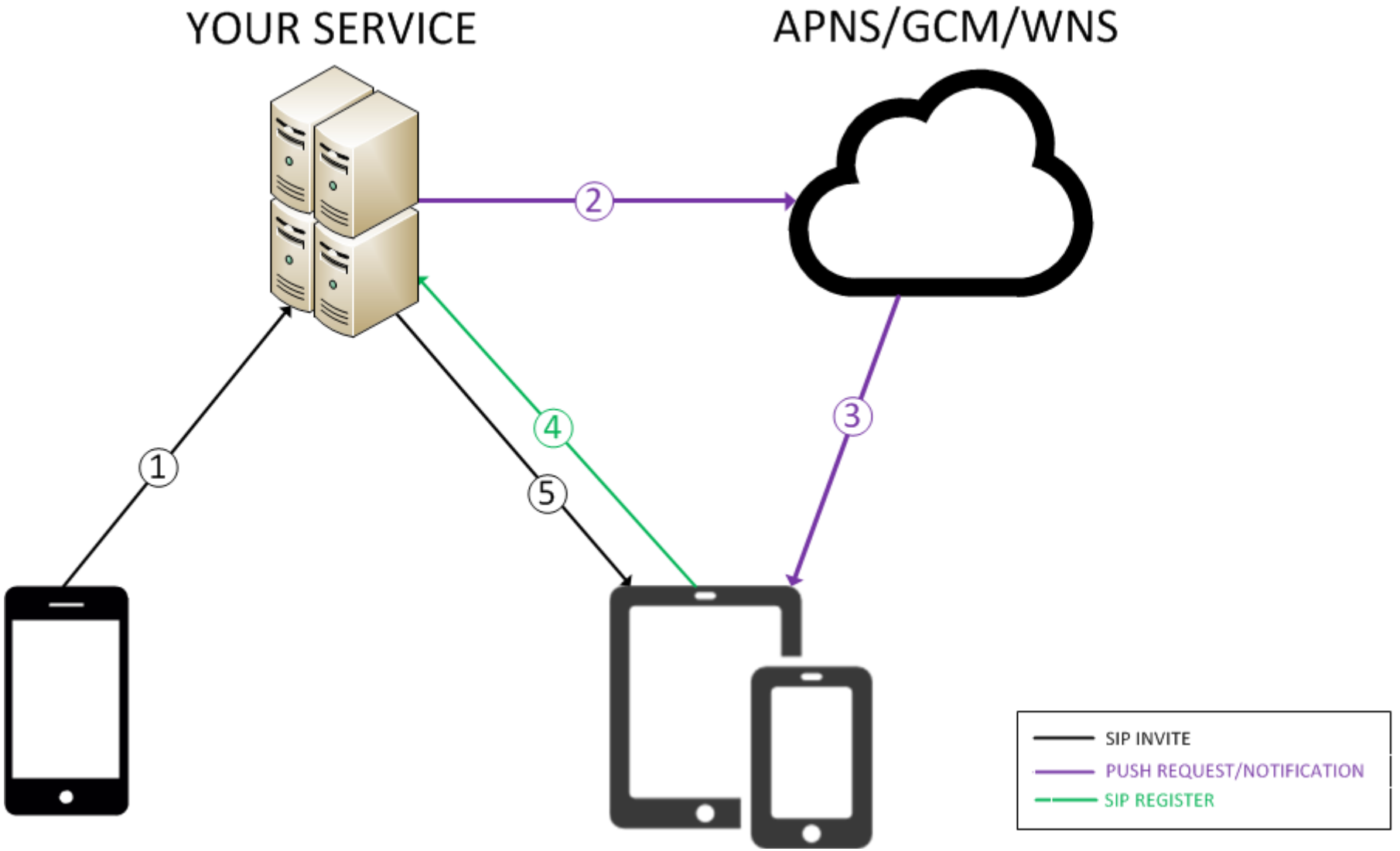
Push notifications

In mobile context:
a short message sent to a smartphone through a centralized service and targeted to a specific app.

It can be used to trigger an action from the app.



Push notifications & VoIP



Push notifications

Pros

- Simple way for a service to send notifications to user's device
- Available on several platforms/OS
- Reliable (APNS > 99% success rate)
- No need to deal with pinhole/NAT
- No need to permanently store users' location
- Use resources “on demand”

Push notifications

Cons

- No immediate delivery status report
- No delivery guarantee
- Not possible to predict delivery time
- Different API per provider
- Different UX per OS/Version:
  - Apple push before iOS 7 vs Apple “silent push”

Multi-device: new challenges

- Users can have many devices
- Only some of them may be reachable
- Each device may have a different Push Notification system

Multi-device: the question

How long do we have to wait
before relaying the INVITE?

TSILO: concept

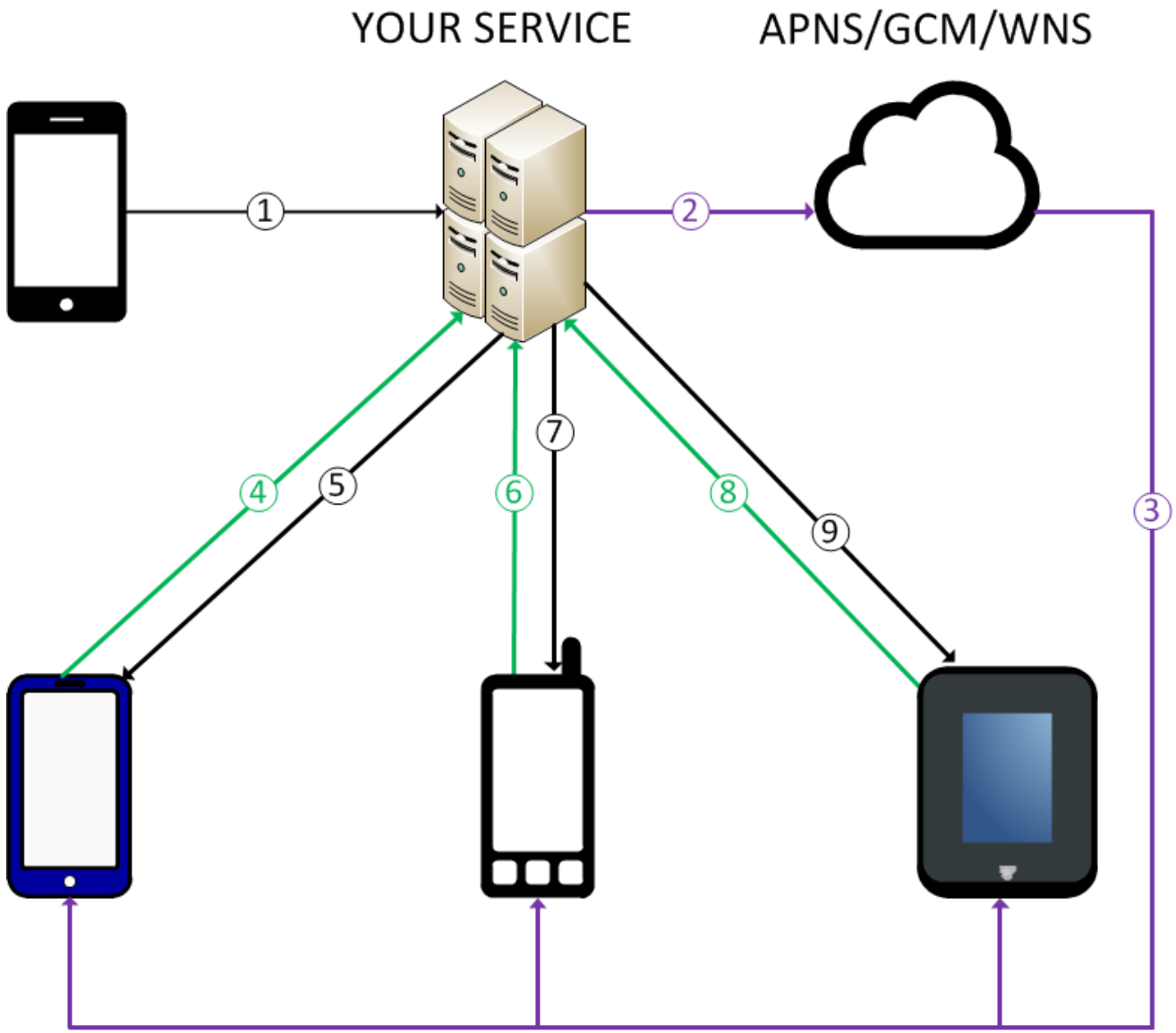
“Late” parallel forking

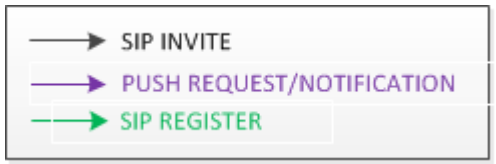
Instead of forking an INVITE just to the currently registered contacts, send the INVITE to the available ones and continue adding branches as soon as new ones register and as long as the transaction hasn't got a final reply.

TSILO = Transactions Silo

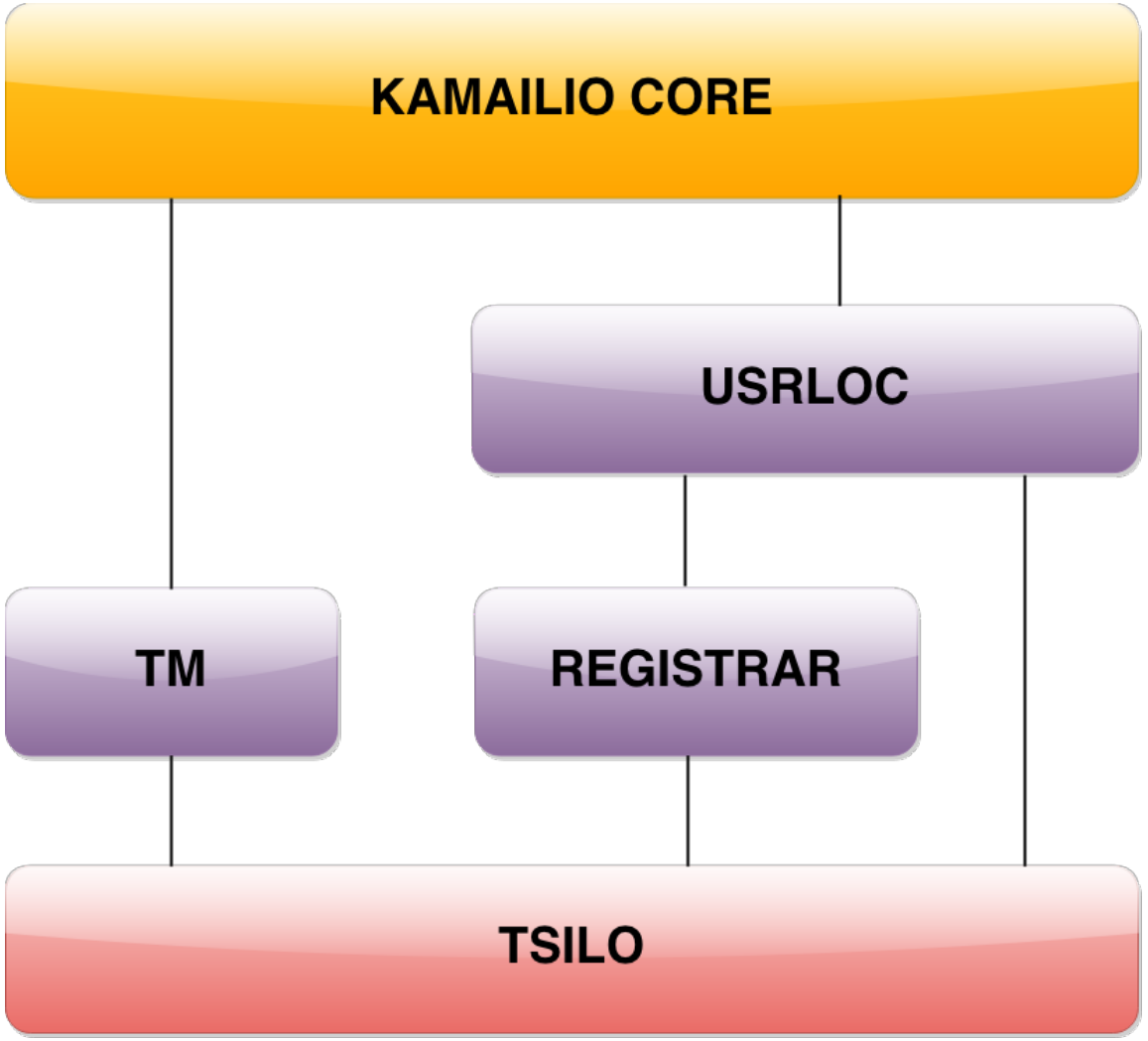
Push & multi-device

Callee's devices receive the INVITE (5,7,9) as soon as they register (4,6,8)





TSILO: details



TSILO: exported functions

- **ts_store()**
  store the current transaction
- **ts_append(domain[1], ruri)**
  add branches to all the transactions currently stored for ruri
- **ts_append_to(tindex, tlabel, domain[1])**
  add branches to a specific transaction identified by tindex and tlabel

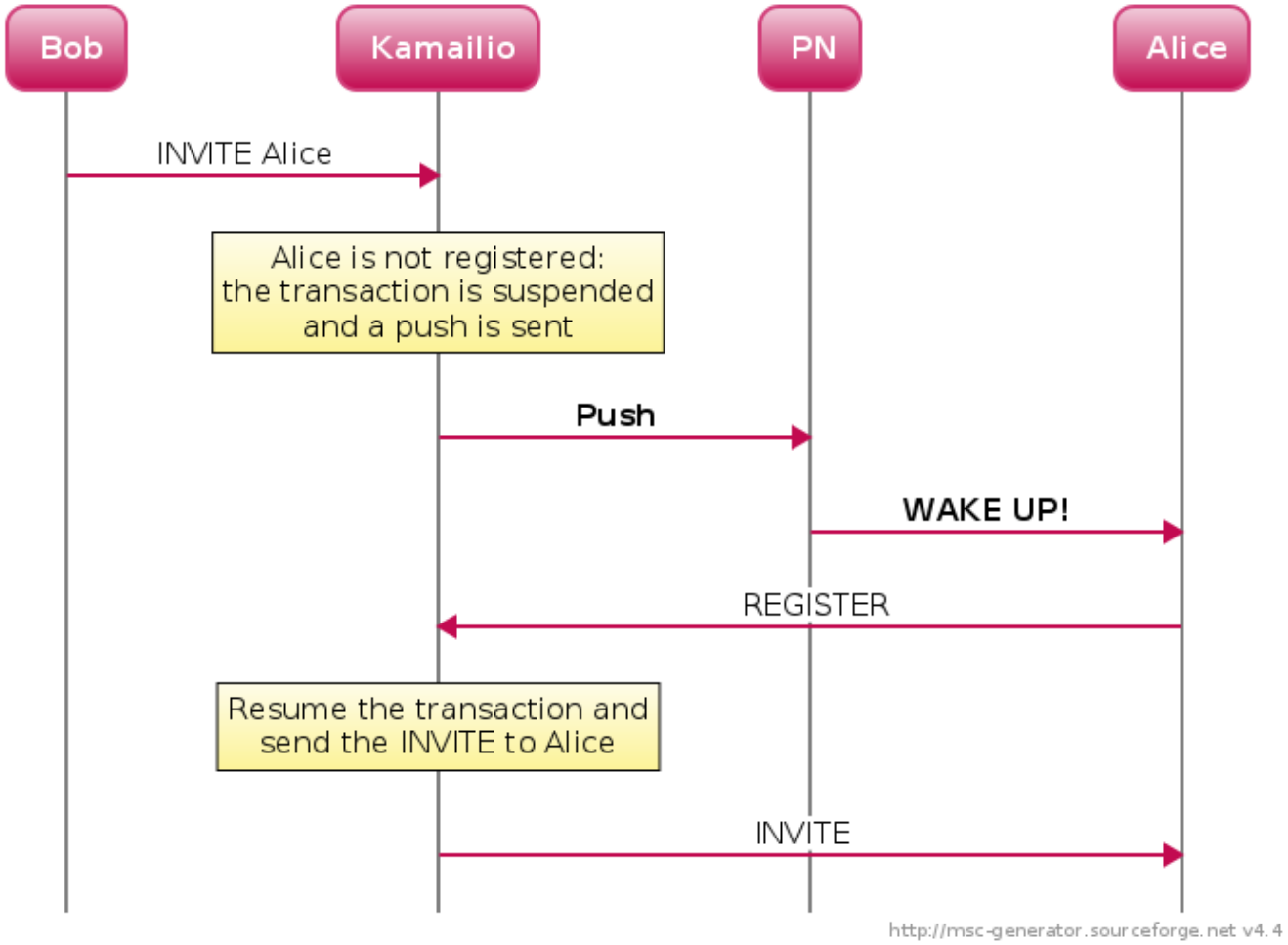
(1) the table over which the internal lookup is performed

TSILO: implementation details

- r-uris stored as entries of a hash table (separate chaining with linked lists for collisions)
- transactions per r-uri stored as linked list of the hash table entries
- t_append*() call lookup_to_dset (REGISTRAR module) and t_append_branches (TM module)
- For each transaction a TM cb for TMCB_DESTROY event is registered

TSILO: example (preamble)

Based on KamailioWorld 2014 Async Push example
by Daniel Constantin-Mierla



TSILO: example

```
loadmodule “tsilo.so”

# ----- htable params -----
modparam("htable", "htable",
"vtp=>size=10;autoexpire=120;dbtable=htable;dbmode=0")

request_route {
    ...
    if (is_method(“INVITE”))
    {
        route(INVITE);
    }

    if (is_method(“REGISTER”))
    {
        route(REGISTER);
    }
    ...
}
```

TSILO: example

```
# manage incoming INVITEs
route[INVITE] {
    if (!lookup("location"))
    {
        send_reply("100", "Trying");
        route(SUSPEND);
    }
    else
    {
        t_relay();

        ts_store();

        $sht(vtp=>stored::$rU) = 1;
        xdbg("stored transaction [$T(id_index):$T(id_label)] $fU
=> $rU\n");
    }

    route(SENDPUSH);
}
```

TSILO: example

```
# suspend the transaction
route[SUSPEND] {

    if(!t_suspend())
    {
        xlog("failed suspending trasaction [$T(id_index):
$T(id_label)]\n");
        send_reply("501", "Unknown destination");
        exit;
    }

    xdbg("suspended transaction [$T(id_index):$T(id_label)] $fU
=> $rU\n");

    $sht(vtp=>join::$rU) = "" + $T(id_index) + ":" +
$T(id_label);

    xdbg("htable key value [$sht(vtp=>join::$rU)]\n");
}
```

TSILO: example

```
route[SENDPUSH] {
    $var(luaret) = 0;
    if(lua_runstring("do_push([[$hdr(X-VxTo)]], [[$tU]], [[$hdr(X-
VxFrom)]], [[$fU]], [[$ci]])")<0)
    {
        send_reply("501", "No link to destination");
        exit;
    }
    if($var(luaret)!=1)
    {
        send_reply("501", "Unknown destination");
        exit;
    }
}

# manage incoming REGISTERs
route[REGISTER] {
    if (!save("location"))
        sl_reply_error();

    route(PUSHJOIN);
    exit;
}
```

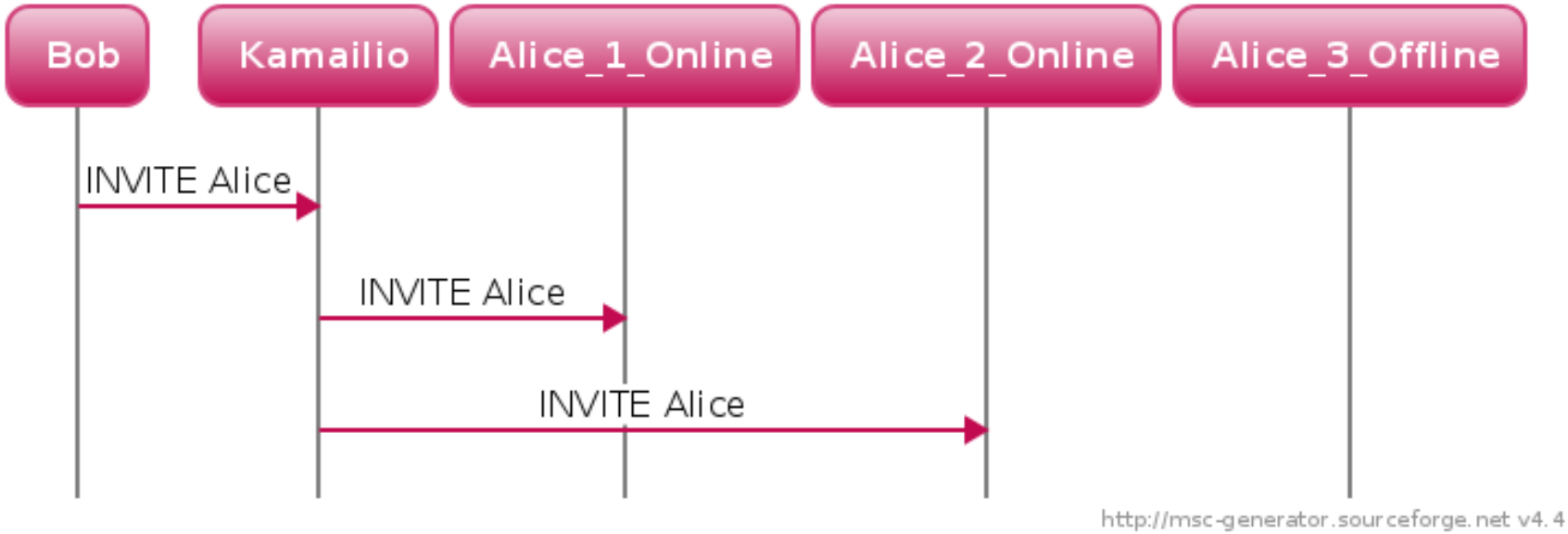
TSILO: example

```
# append branches or resume the transaction
route[PUSHJOIN] {
    $var(hjoin) = 0;
    lock("$tU");
    $var(hjoin) = $sht(vtp=>join::$tU);
    $var(hstored) = $sht(vtp=>stored::$tU);
    $sht(vtp=>join::$tU) = $null;
    unlock("$tU");

    if ($var(hjoin)==0)
    {
        if ($var(hstored))
            ts_append("location", "$tU");
        return;
    }

    $var(id_index) = $(var(hjoin){s.select,0,:}{s.int});
    $var(id_label) = $(var(hjoin){s.select,1,:}{s.int});
    xdbg("resuming trasaction [$var(id_index):$var(id_label)] $tU
($var(hjoin))\n");
    t_continue("$var(id_index)", "$var(id_label)", "INVRESUME");
}
```

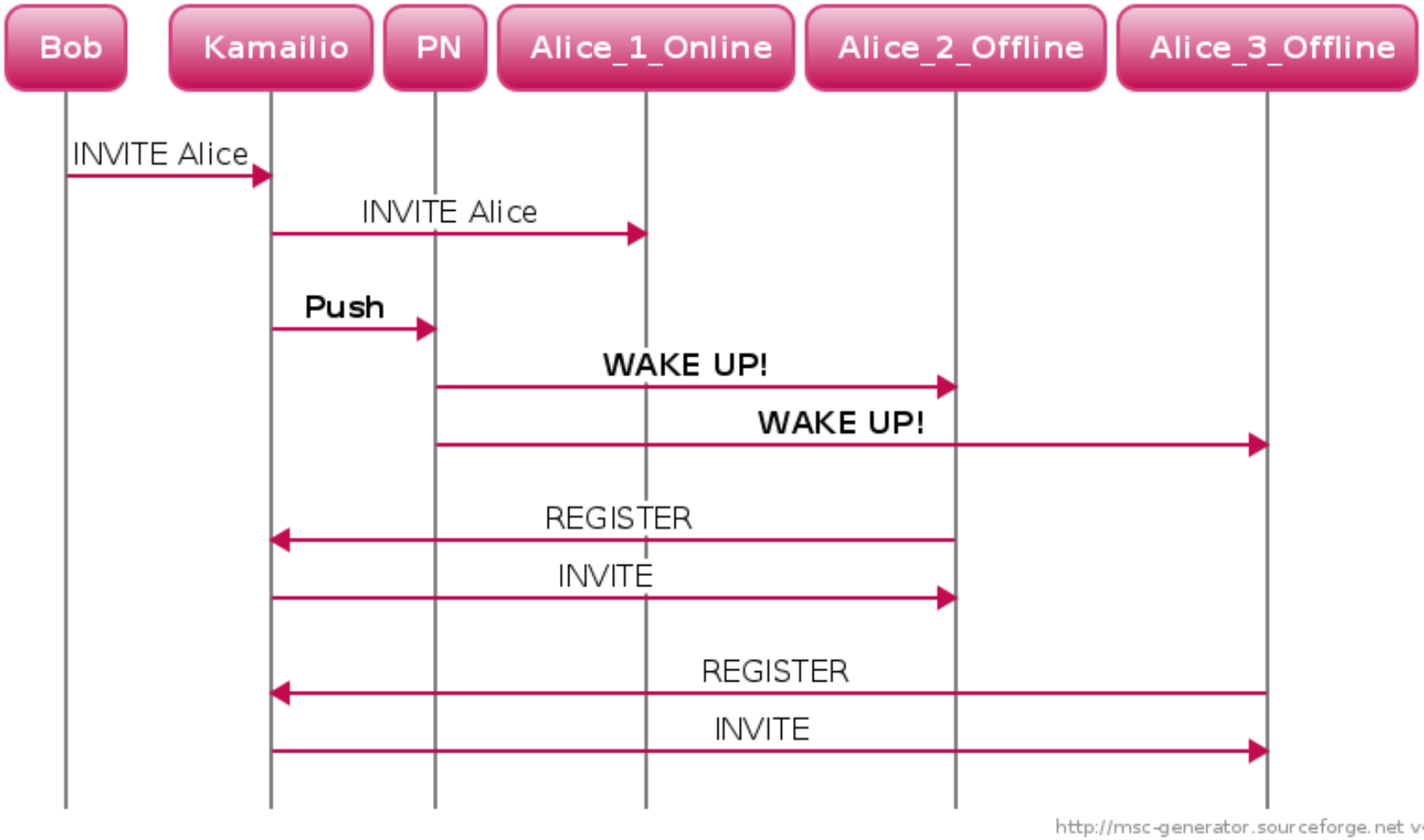
TSILO: example

```
# lookup and relay after resuming transaction
route[INVRESUME] {
    lookup("location");
    t_relay();

    ts_store();

    $sht(vtp=>stored::$rU) = 1;
    xdbg("stored transaction [$T(id_index):$T(id_label)] $fU =>
$rU\n");
}
```

Multi-device without Push & TSILO



Multi-device with Push & TSILO



TSILO: improvements

- Stats
- More RPC commands
- USRLOC callbacks?

Thank you!

Questions?

email: federico.cabiddu@gmail.com