# Development of convergent J2EE applications for OpenSER

Elias Baixas Morató
Engineer
VozTelecom Sistemas

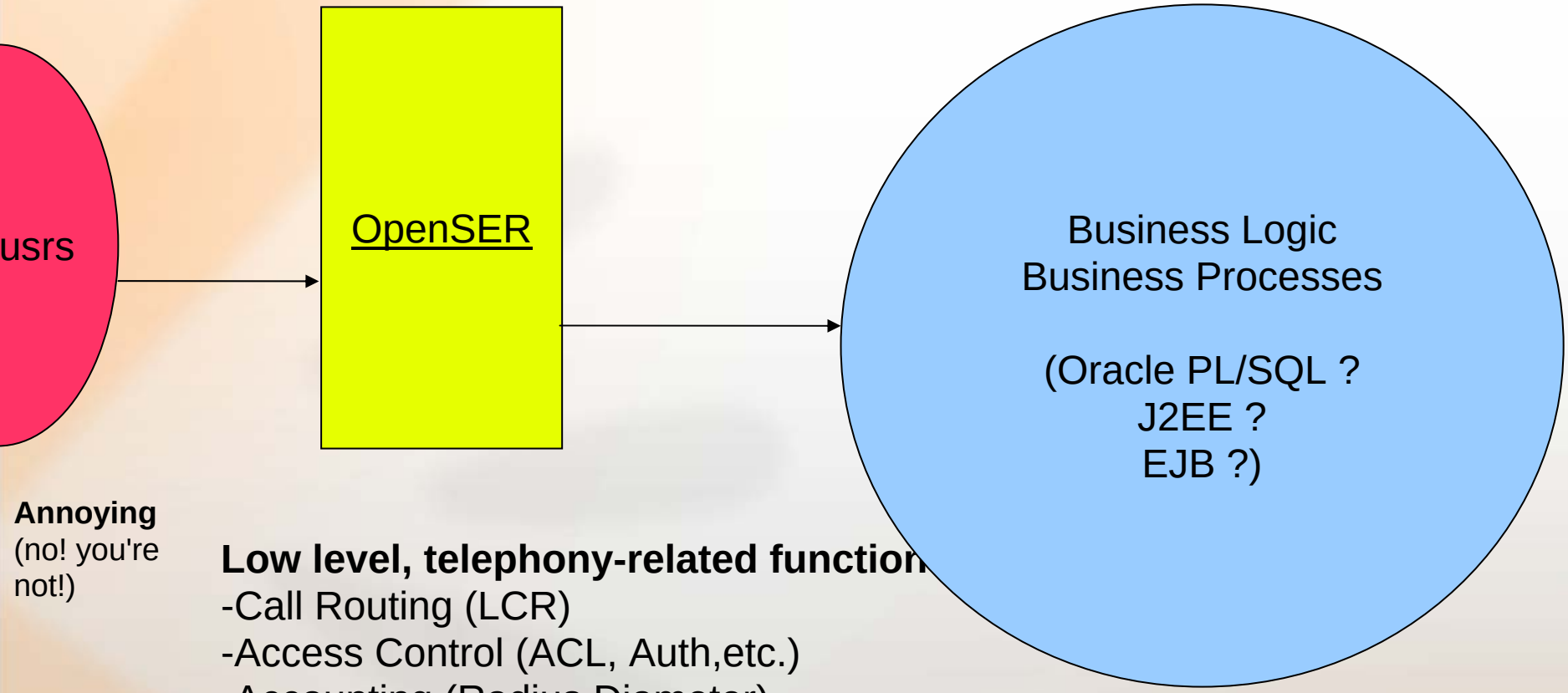
# Development of convergent J2EE applications for OpenSER

VozTelecom

- SIP is a difficult protocol (you know what I mean ?)
  - Have you ever received a CANCEL while parallel forking with a final success response already sent upstream ? (let me think...)
- IP Telephony is a difficult issue
  - LCR, auth, accounting, locating users...
- SIP Applications are cool, but if you try to face everything at once, you end up with a big and beautiful mess.
- “Divide and Conquer” always works.

# Development of convergent J2EE applications for OpenSER

VozTelecom




**Annoying**
(no! you're not!)

**Low level, telephony-related function**
-Call Routing (LCR)
-Access Control (ACL, Auth,etc.)
-Accounting (Radius,Diameter)
-Nat Traversal
-UsrLoc
-Etc...

# Development of convergent J2EE applications for OpenSER

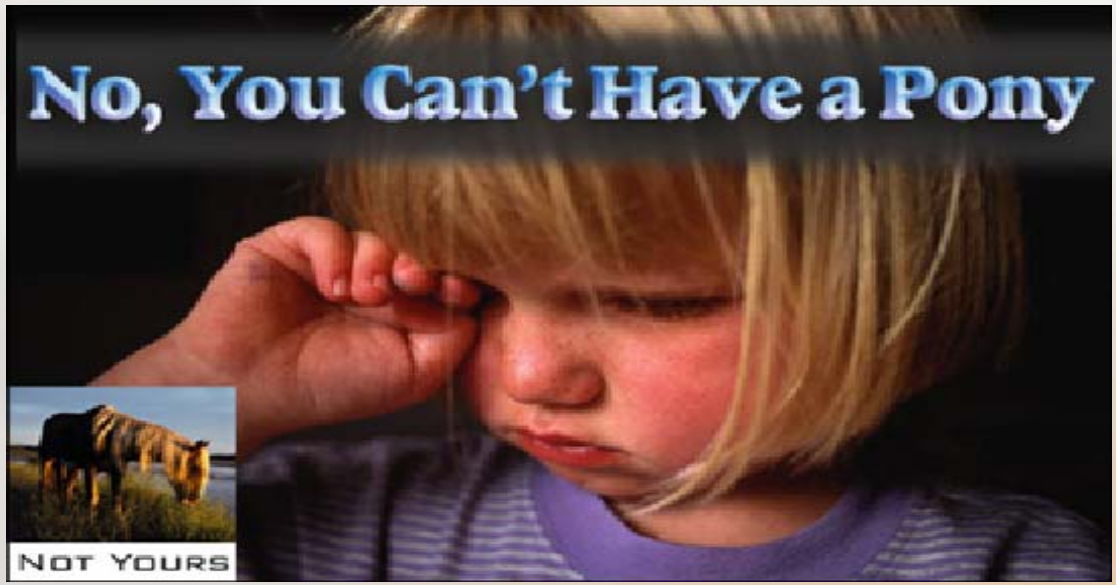
VozTelecom

- OpenSER script: Domain Specific Language (DSL)
  - Hyper efficient
  - Call handling
  - Feature-rich: auth, acc, loc, lcr, radius, etc.
  - Extensible through modules
  - Robust
- How do I integrate OpenSER into my business processes ?
  - Of course: Databases (MySQL), maybe “exec” module, XML-RPC ?

# Development of convergent J2EE applications for OpenSER

VozTelecom

I Want Business Logic, Model-Viewer-Controller, AJAX, Web 2.0, Servlet, J2EE, EJB integrated with OpenSER !

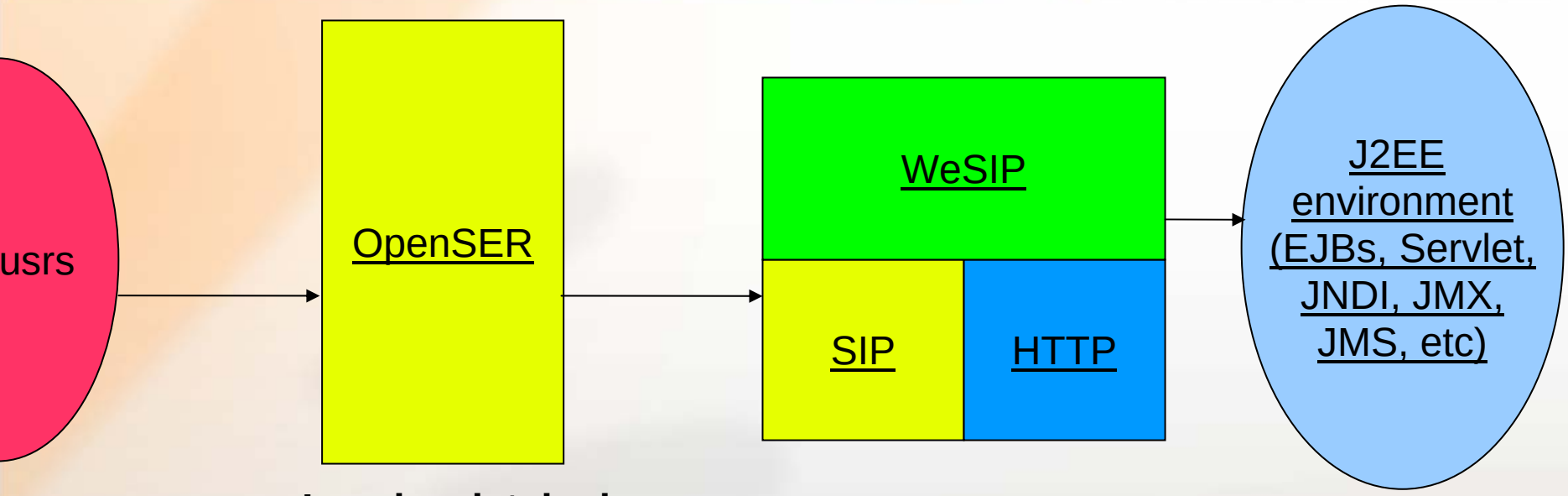


VozTelecom

# You Have WeSIP !!!

# Development of convergent J2EE applications for OpenSER

VozTelecom




**Annoying**
(no! you're not!)

**Low level, telephony-related functions:**
-Call Routing (LCR)
-Access Control (ACL, Auth,etc.)
-Accounting (Radius)
-Nat Traversal
-UsrLoc
-Etc...

**Application Logic**
B2BUA
Web or SIP triggered PBX functionality (JAVA APIs)
Trigger IVR
Trigger Media Services
Intelligent call routing

**Business Logic**
-Provisioning
-Invoking thrid party services
-Billing
-(Accounting)
-Statistics
-...

# Development of convergent J2EE applications for OpenSER

VozTelecom

- OpenSER script: Domain Specific Language (DSL)
  - Hiper efficient
  - Call handling
  - Feature-rich: auth, acc, loc, lcr, radius, etc.
- JAVA: General Purpose Language
- SipServlet: Application Programming Paradigm
- J2EE: Business Logic programming Environment.

# Development of convergent J2EE applications for OpenSER

VozTelecom

- OpenSER DSL
- 1-Authenticate
- 2-Account
- 3-UsrLoc
- 4-Nat Traversal
- .
- .
- .
- N-Invoke App-Server

```
if (!www_authorize("foo.bar", "subscriber"))
        www_challenge("foo.bar", "0");
        return;
```

```
# labeled with this flag, its co
modparam("acc", "log_flag", 1 )
```

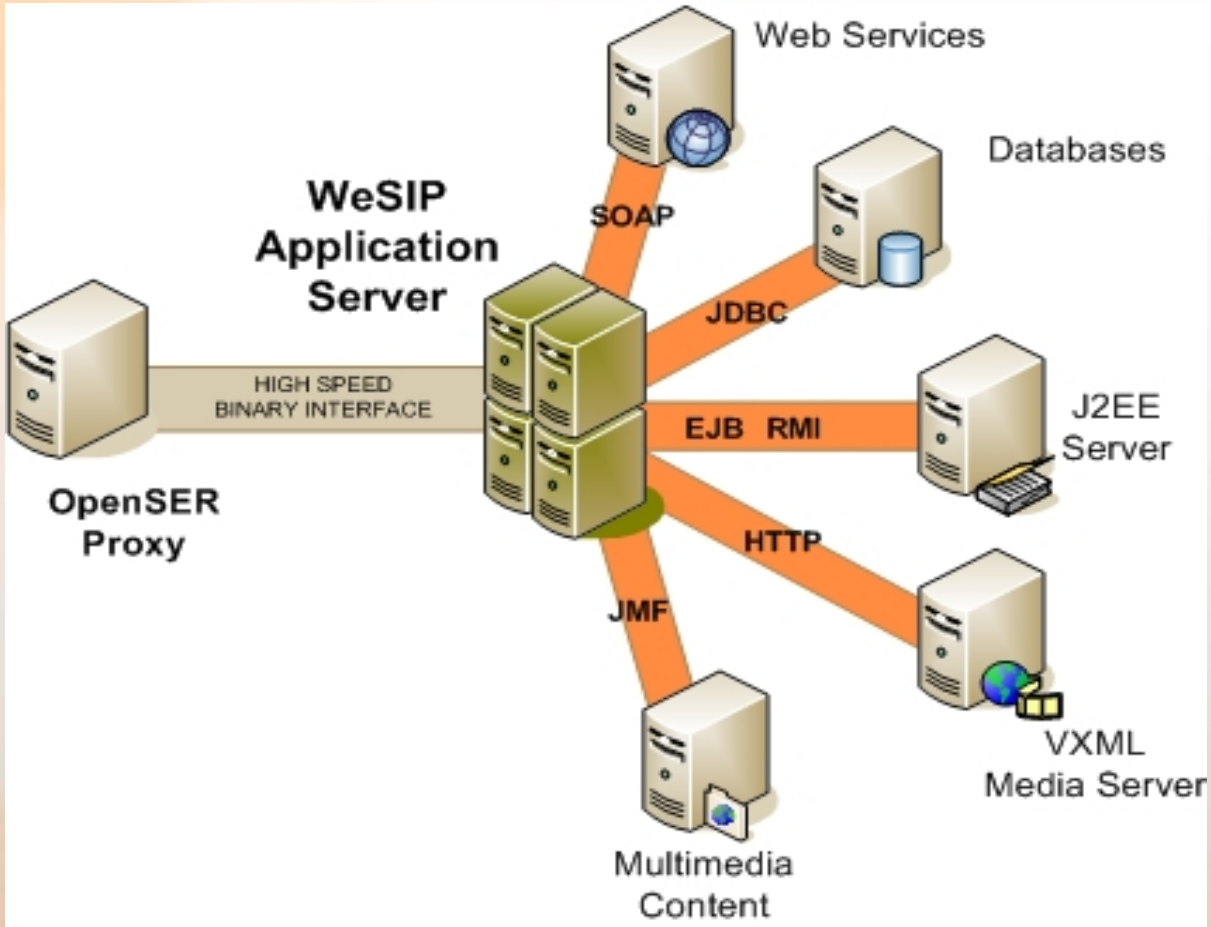
```
if (method=="REGISTER")
 save("location");
        log("REGISTER r
```

```
if (nat_uac_test("3")) {
   fix_nated_contact(); # I
   if (method == "INVITE")
      fix_nated_sdp("1");}
            force_rport();
```

```
if(!as_relay_t("app_server_1")){
        t_reply("500","Application Server error");
}
```

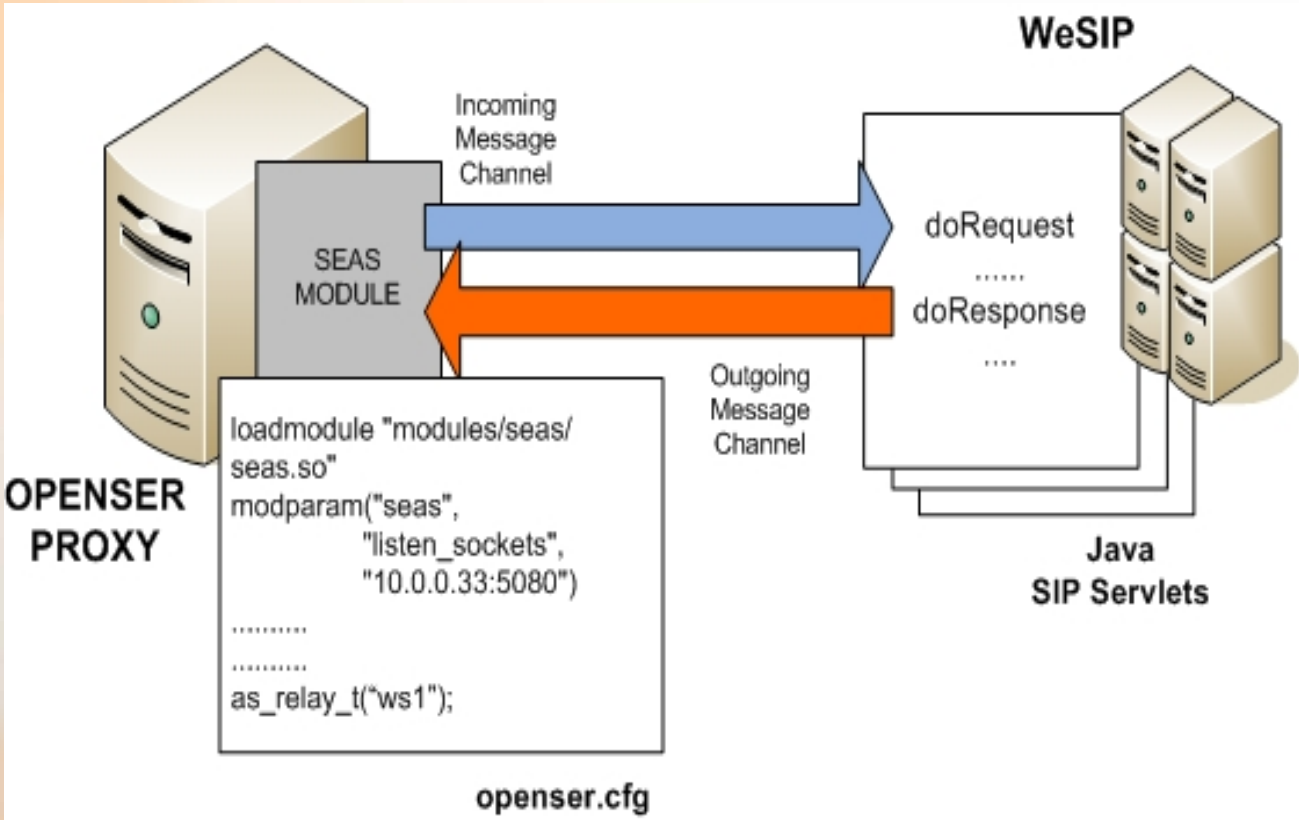
# Development of convergent J2EE applications for OpenSER

VozTelecom



# Development of convergent J2EE applications for OpenSER

VozTelecom



# Click2Call example

VozTelecom

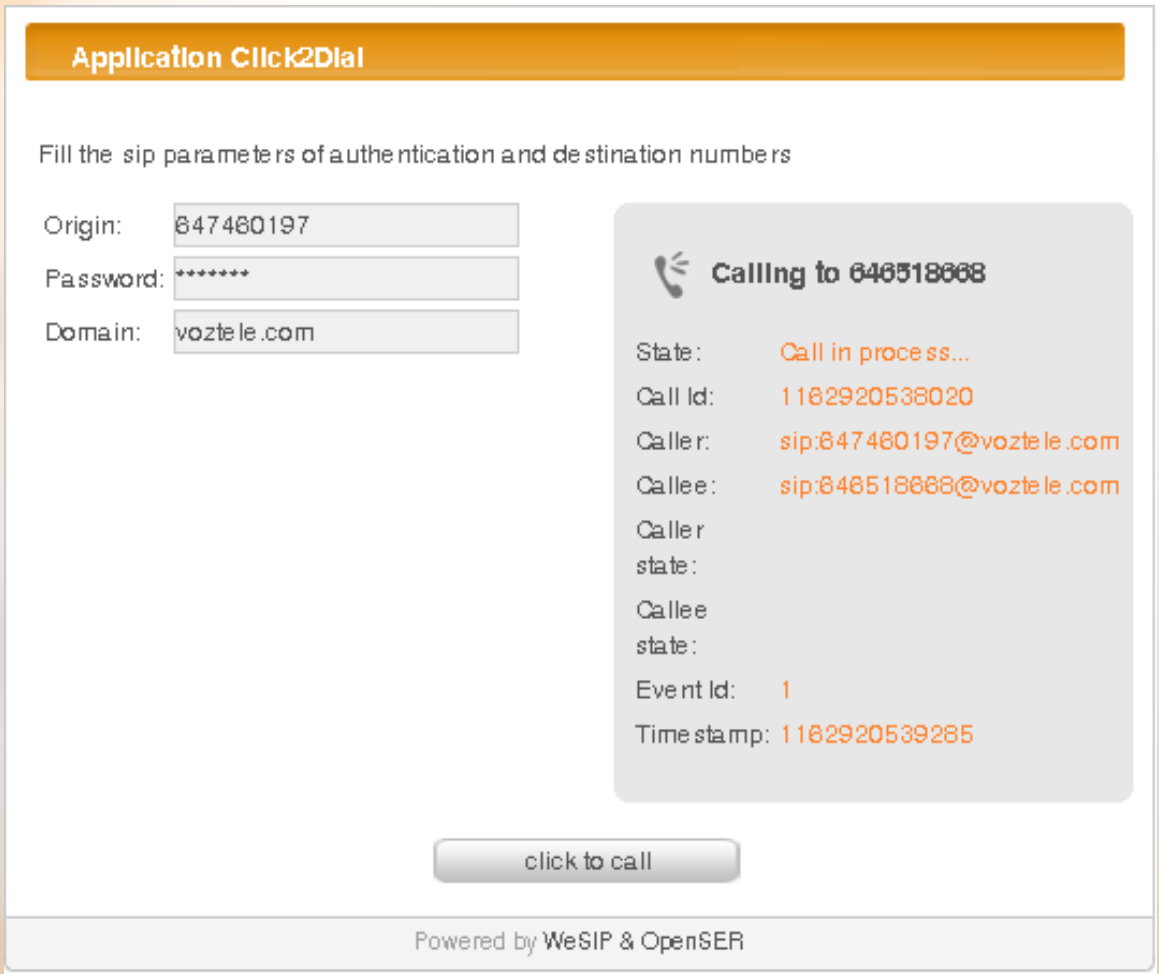
- SipServlet is invoked:
  - Involves:
    - 2 SipSessions (call-legs)
    - 1 HttpSession (Web user)
  - Web Technologies
    - Direct Web Remoting JAVA API
    - AJAX (At last SIP programmers can have their AJAX dose ! :P )
    - Javascript, CSS, DHTML...

# Click2Call

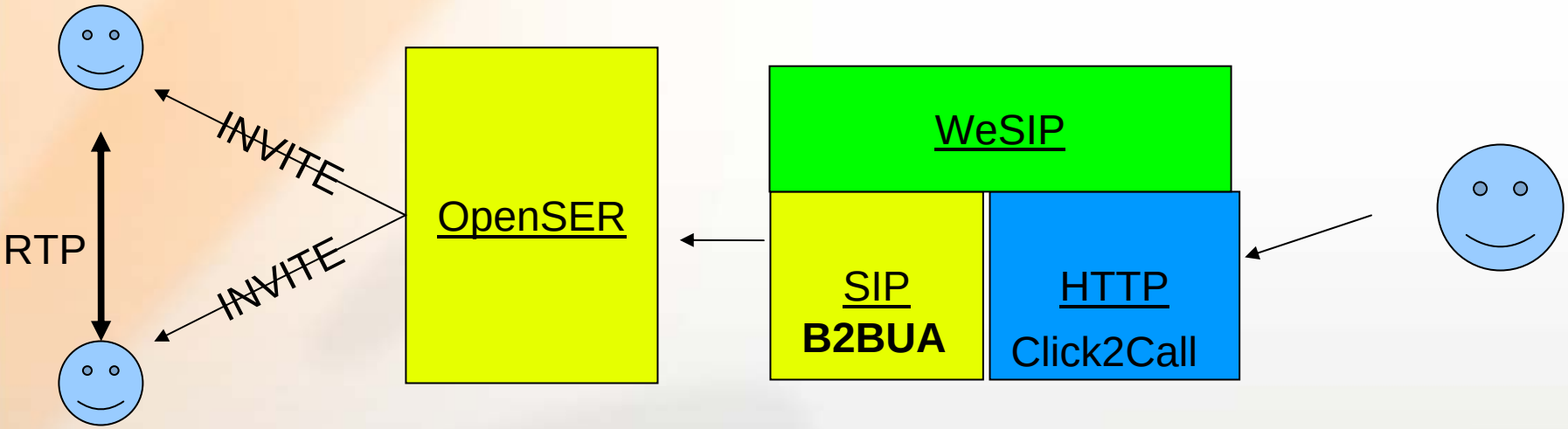
VozTelecom

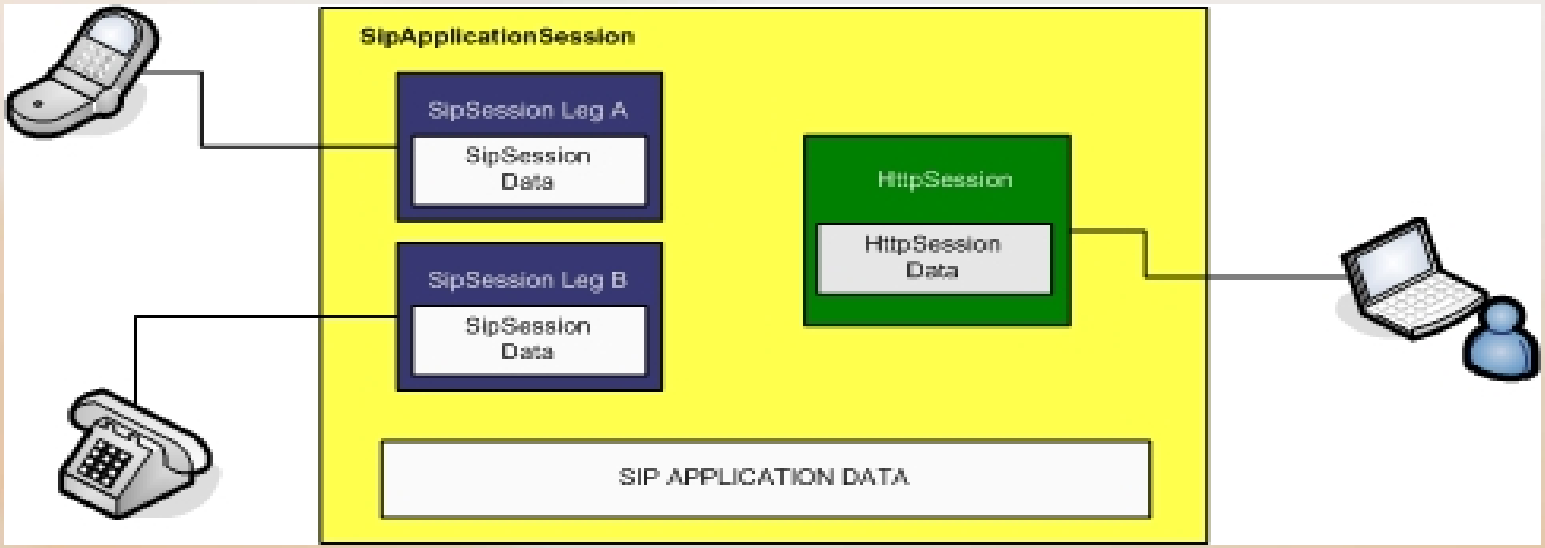
# Click2Call

VozTelecom



# Click2Call


INVITE
INVITE
RTP
OpenSER
WeSIP
SIP
B2BUA
HTTP
Click2Call


SipApplicationSession
SipSession Leg A
SipSession Data
SipSession Leg B
SipSession Data
HttpSession
HttpSession Data
SIP APPLICATION DATA

# Click2Call

VozTelecom

```
public int c2c(String src, String dst, ServletContext sc, HttpSession httpSession, ServletConfig servletConfig){
    try{
        //Getting the C2CSession of ServletContext
        C2CSession c2cSS = (C2CSession) sc.getAttribute(src);

        if(c2cSS != null){
            // If exists get the Click2Call servlet name for to dispatch the process
            // of the following message received.
            String click2call = servletConfig.getInitParameter(CLICK2CALL);

            //Setting Src and Dst address in C2CSession
            c2cSS.setSrcAddress("sip:"+src+"@"+ c2cSS.getDomain());
            c2cSS.setDstAddress("sip:"+dst+"@"+ c2cSS.getDomain());

            //Getting the SipFactory of ServletContext
            SipFactory sipFactory = (SipFactory) sc.getAttribute(SipServlet.SIP_FACTORY);

            //Creating the INVITE request for the SRC user.
            SipServletRequest inviteOrigin = sipFactory.createRequest(sipFactory.createApplicationSession(), "INVITE",

            // Assinging the SipServlet that to process the logical.
            inviteOrigin.getSession().setHandler(click2call);

            //Setting the C2CSession in the application session
            inviteOrigin.getApplicationSession().setAttribute(C2CSESSION, c2cSS);

            // Send the INVITE to the origin user
            try{
                inviteOrigin.send();
            }catch(Throwable e){
                inviteOrigin.getApplicationSession().removeAttribute(C2CSESSION);
                httpSession.removeAttribute(C2CSESSION);
                throw e;
            }

            // Return 0 if all is correct
            return 0;
        }else{
```

# Click2Call

VozTelecom

```
protected void doSuccessResponse(SipServletResponse response) throws ServletException, IOException {
    // Getting the C2CSession of the ApplicationSession attribute.
    C2CSession c2cSS = (C2CSession) response.getApplicationSession().getAttribute(C2CSESSION);

    // Getting the method of this response
    String method = response.getMethod();

    // Check the status of response.
    switch(response.getStatus()){
        case 200:
            if(method.equals("INVITE")){
                // If method is a INVITE getting src SipSession
                SipSession srcSipSession = response.getSession();

                // Creates one new INIVTE for the callee user.
                SipServletRequest inviteDestination = sipFactory.createRequest(response.getApplicationSession(), "I

                // Assign the new SipSession created to the C2CDstCallSipServlet
                inviteDestination.getSession().setHandler(destCallServletName);

                // Interchanging sessions like attributes of session of the same ones.
                inviteDestination.getSession().setAttribute(PEER_SESSION, srcSipSession);
                srcSipSession.setAttribute(PEER_SESSION, inviteDestination.getSession());

                // In the srcSipSession setting the success response received like attribute.
                srcSipSession.setAttribute(RESPONSE, response);

                // Setting the session description of successResponse like content of new Invite created.
                inviteDestination.setContent(response.getContent(), response.getContentType());

                // Sending the new Invite.
                inviteDestination.send();
            }
    }
}
```

# Development of convergent J2EE applications for OpenSER

VozTelecom