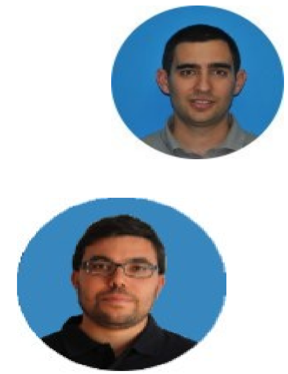
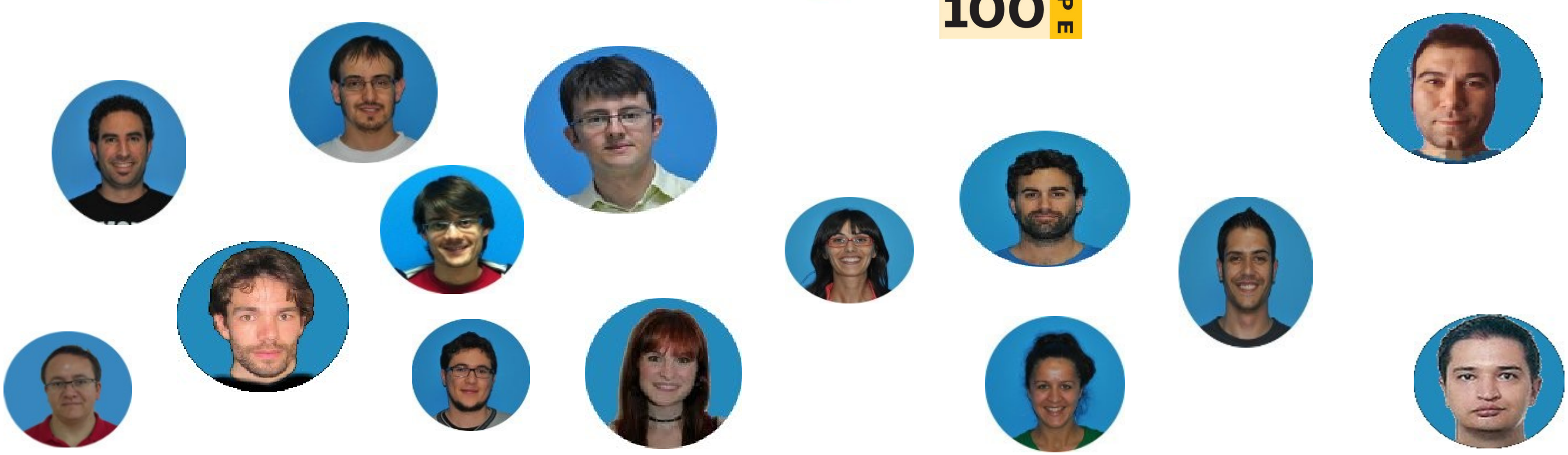
KAMAILIO WORLD
CONFERENCE & EXHIBITION

WebRTC Signaling

In a few words...

- We’re all about interconnection and security in UC
- Strong expertise on WebRTC technology
- Founded in 2006, privately held, no VCs
- Markets: telco and enterprise UC solutions
- HQ in Spain, worldwide sales
- Recent awards:





About WebRTC & Quobis

Quobis plays a key-role in WebRTC industry, as is running 35+ PoCs in Tier1-2 telcos in EMEA, LATAM, US and APAC.

Co-authoring (Víctor Pascual) the RFC7118 standard for SIP over Websockets, SIPoWS

Authors of QoffeeSIP, an opensource Javascript SIPoWS implementing RFC7118

Quobis' is co-chairing the SIP Forum WebRTC Task Group, whose objective is to enable deployment of WebRTC for SIP-based domains

Quobis is member of the ATIS DSI initative, which is leading the ORCA.js opensource project

- We are running around **30 PoCs**.
- We are learning **valuable lessons** from them.
- We used KAMAILIO + FreeSWITCH to implement our demo platform.
- ...but we also have to work with vendor solutions.

Signaling in WebRTC

W3C does not define the Signaling to use for WebRTC applications

Many signaling protocols have been adopted by developers and vendors

Signaling in WebRTC

Standard-based and adopted by Open Source community

SIP over Websocket: VoIP friendly, trickle ICE no direct to implement but doable, adopted by the main VoIP Open Source solutions.

XMPP over Websocket: Jingle over WS libraries, it also is used in Open Source solutions.

Signaling in WebRTC

Proprietary/non-standard solutions adopted by vendors or specific developments.

- **JSONoWS:** Web developer friendly, easy to implement trickle ICE, flexible (you have to invent everything)
- **REST API** + (Websocket || Long-polling) for events: Web developer friendly, massively used in web environments.
- **Over Datachannel:** used for in-dialog signaling, less latency

Signaling in WebRTC

Drawbacks of signaling atomization

1. One application developed for a specific signaling does not work for a different one.

2. Web Developers should not care about the signaling used by the server/Gateway.

3. Signaling stacks offer a different API but they do similar things using different ways.

Signaling in WebRTC

Any solution?

A standard API for **WebRTC** Apps

covering standard use cases

orca.js

The idea is to provide a common API for Web Developers.

Each vendor can provide an orca.js compatible JS library.

SIPoWS stacks can offer orca.js compatible APIs

Signaling Stack

W3C API management

WebRTC GW/Signaling Server

orca.js

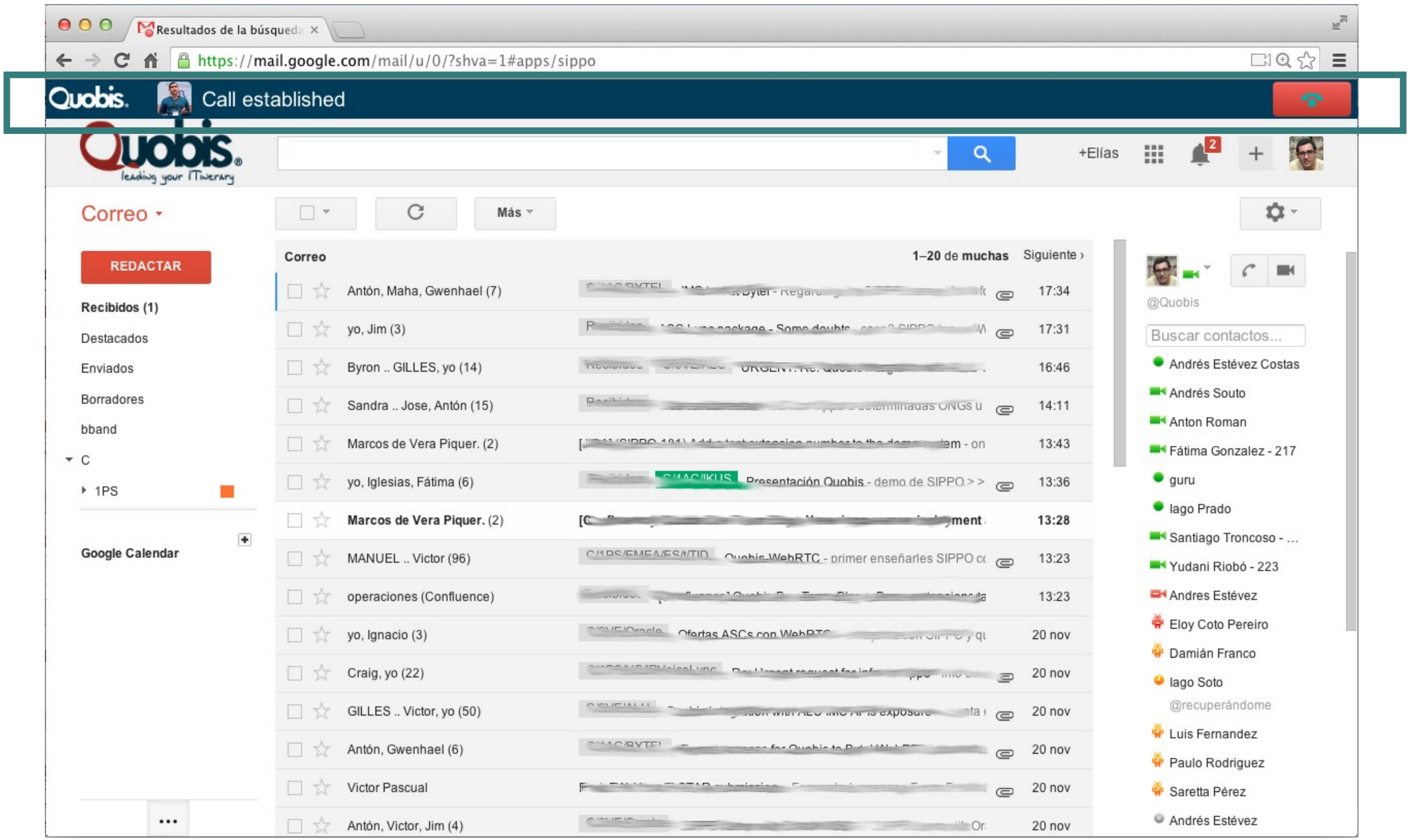
How it looks?:

```
session = createSession(userid, token, sessionConfig);
session.connect();
session.on(incomingcall,handle_incomingcall);
call = session.createCall(callee, video);
call.connect();
call.hold();
call.resume();
call.disconnect();
```

You can check the code at:
https://github.com/orcajs/orca.js

Example of app using orca.js compatible API

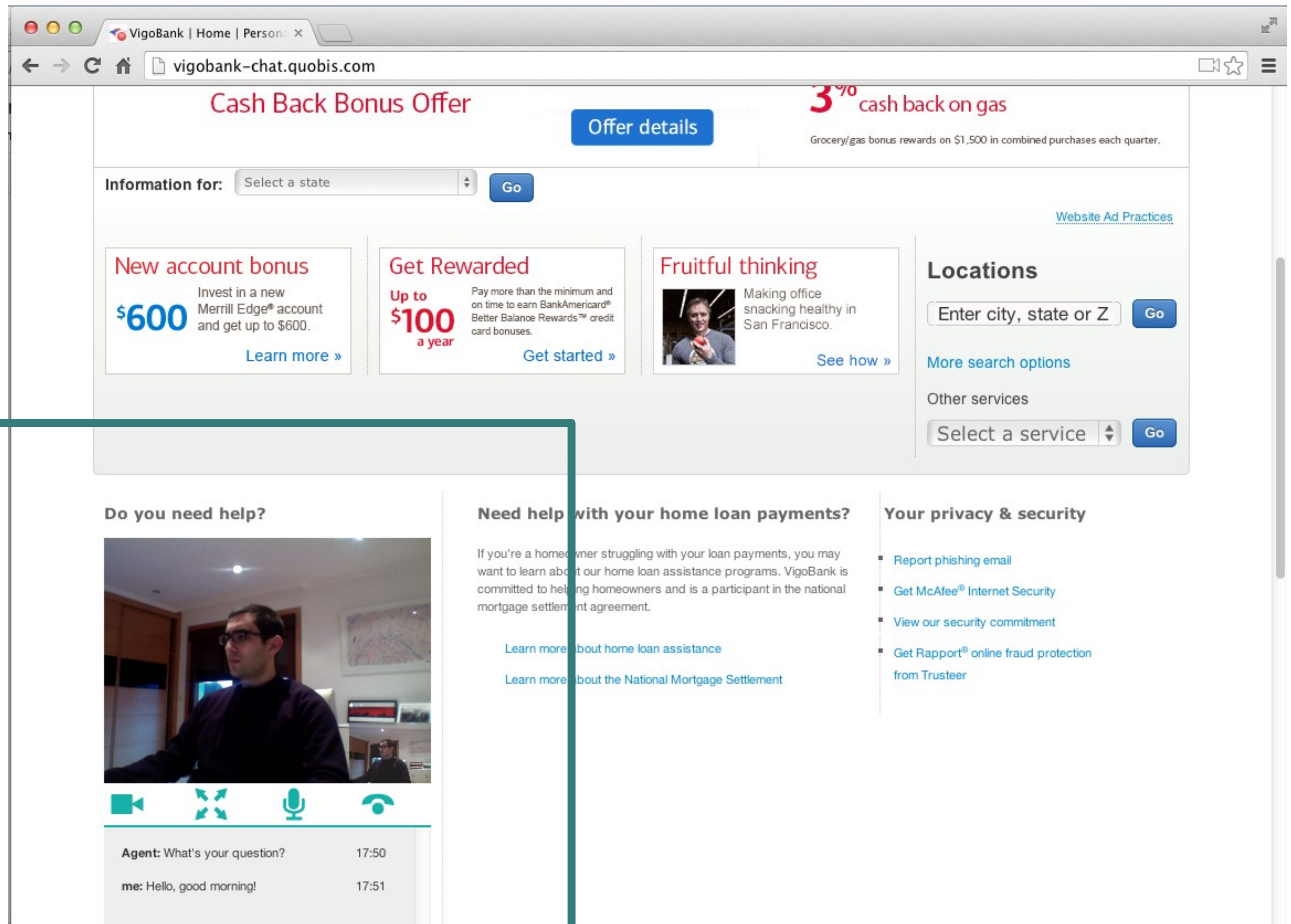
SIPPO GMAIL CONNECTOR



Click to call example

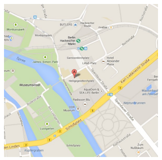
CLICK TO CALL

Click-to-call integration with IM and contextual information



Some apps we are developing

CLICK TO CALL CONSIDERATIONS



The user is anonymous but we need to gather **interesting contextual information** to offer to the agent answering the call and for further **BI analysis**.

Open to DoS attacks. Click to call applications are likely to suffer **DoS and even fraud attacks**. It is important to mitigate this by avoiding too many simultaneous or consecutive calls from same IP/port.

You must **limit the whole number of calls** your customer can handle. This allows to minimizes the impact of **DDoS attacks** and also to **collapse the call center with legitimate traffic**.

Some apps we are developing

CLICK TO CALL CONSIDERATIONS

This scenario can be implemented with Kamailio + Solution to handle media.

Contextual info can be easily transported by modifying a SIP over Websocket stack to send custom headers (use WSS for this, please). e.g. `Geolocation` (we can include the info provided by W3C geolocation API), `origin-url`, etc.

We can get **source IP and port in Kamailio for CDRs** and to implement DoS protections.

We can store all this information in the **CDRs directly using Kamailio.**

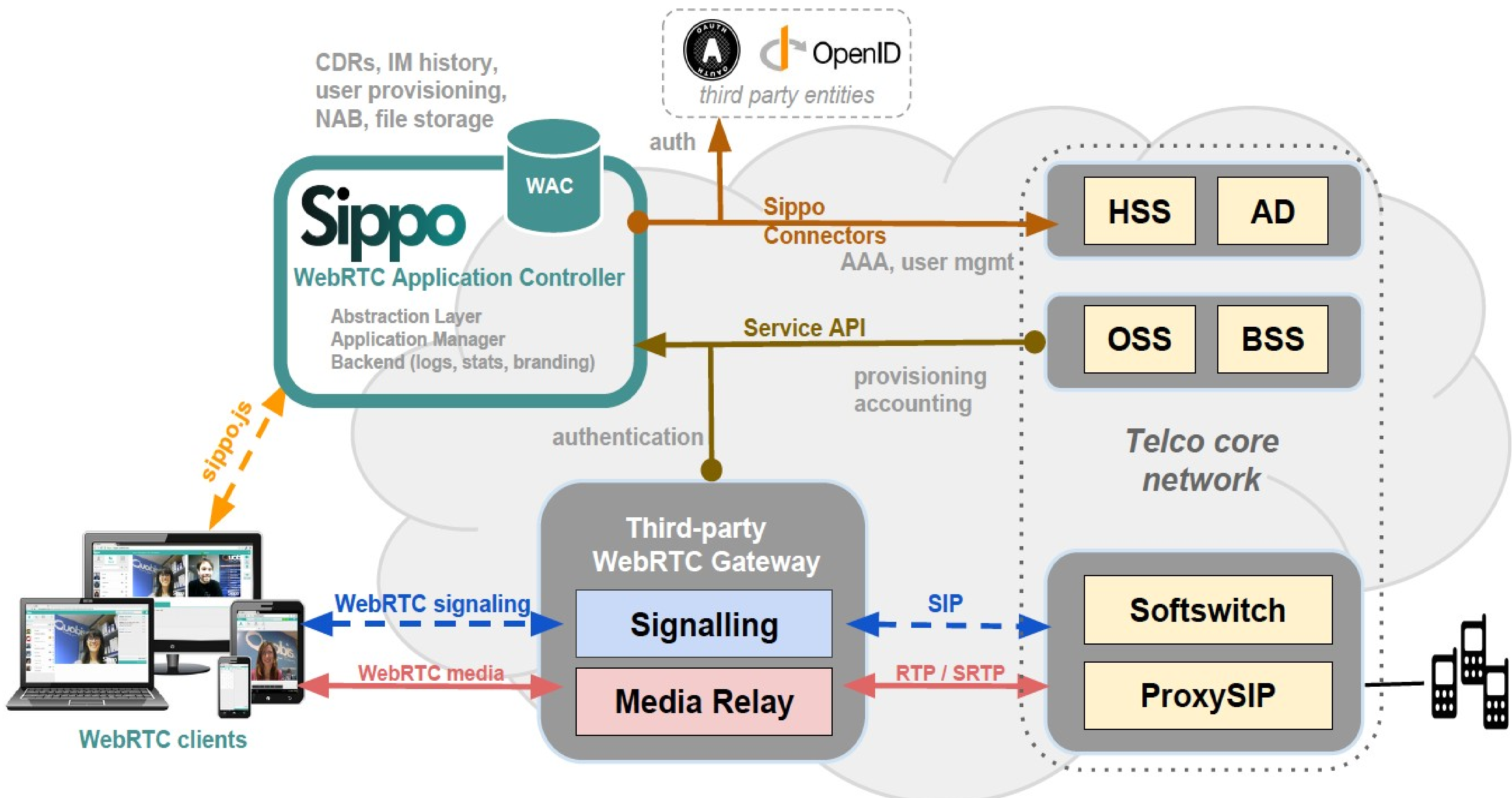
KAMAILIO

```
INVITE agent@customer1.quobis.com
…

Geolocation: 52.52076229+13.40186559
Url: demo.quobis.com/c2c

...
```

Sippo WebRTC Application Controller



Quobis®
leading your ITinerary