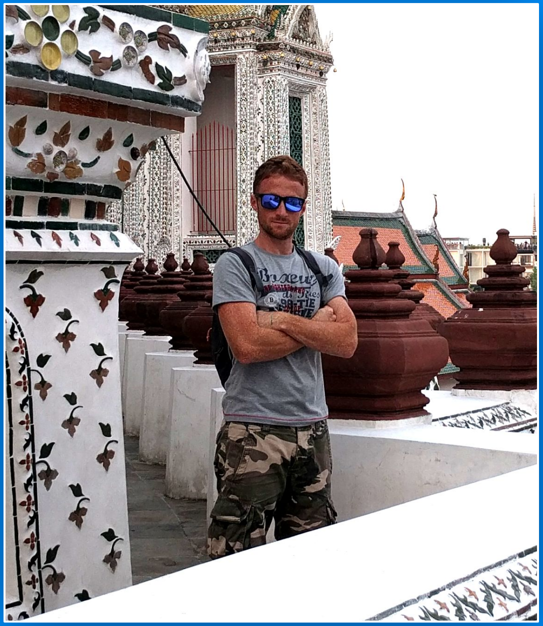
Asterisk WebRTC frontier: make client SIP Phone with

sipML5 - Janus Gateway

nethesis

Alessandro Polidori
@ale_polidori

Fosdem 2019 - Brussels
Realtime DevRoom



Alessandro Polidori

Software Engineer @Nethesis

#Node.js #WebRTC #OpenSource

alepolidori

@ale_polidori

https://medium.com/@ale_polidori

WebRTC: Web Real Time Communication

- 2011 Google project
- Javascript API for audio/video communications
- 3 Components
  - `getUserMedia:` camera, microphone, screen access
  - `RTCPeerConnection:` negotiation, encoding, decoding, nat traversal
  - `RTCDataChannel:` exchange data between browsers
- Other APIs: `getStats, MediaRecorder`

WebRTC

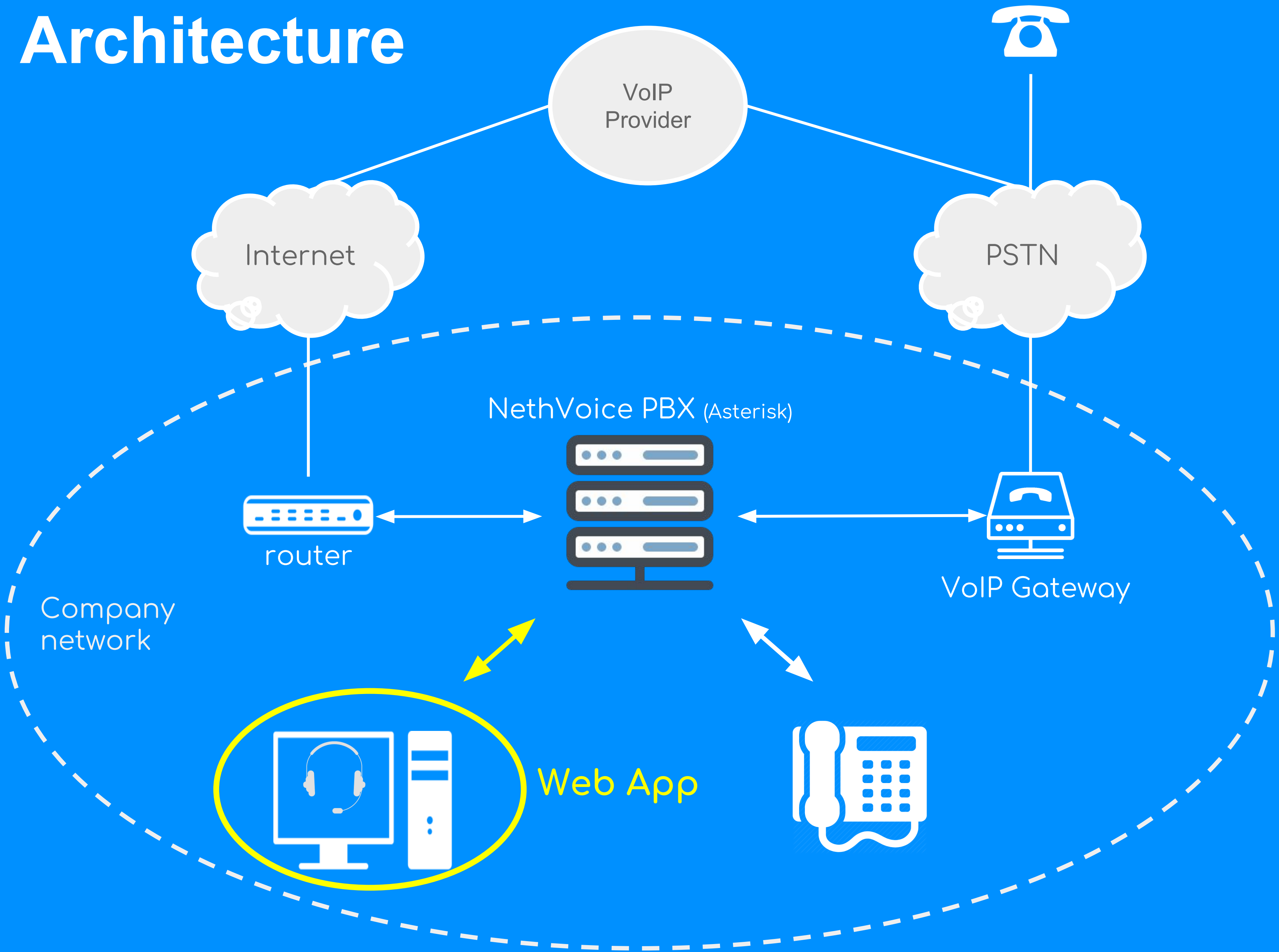
Protocols

- 2 groups
  - RTP → transport
  - SIP → signaling
- SRTP
  - secure real-time transport protocol
  - encryption
  - message authentication
- SDP - audio & video description
- STUN, TURN, ICE

```
INVITE sip:201@10.0.2.15:35050 SIP/2.0
Via: SIP/2.0/UDP 10.0.2.15:5060;rport;bra
From: "foo" <sip:200@10.0.2.15>;tag=afde3
To: <sip:201@10.0.2.15>
Contact: <sip:asterisk@10.0.2.15:5060>
Call-ID: 5e4769af-6478-466e-bebe-8a7e3f5c
CSeq: 27590 INVITE
Allow: OPTIONS, SUBSCRIBE, NOTIFY, PUBLIS
Supported: 100rel, timer, replaces, noref
Session-Expires: 1800
```

```
v=0
o=- 137183621 137183621 IN IP4 10.0.2.15
s=Asterisk
c=IN IP4 10.0.2.15
t=0 0
m=audio 18416 RTP/AVP 0 8 3 111 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:111 G726-32/8000
```

Architecture



CoDec

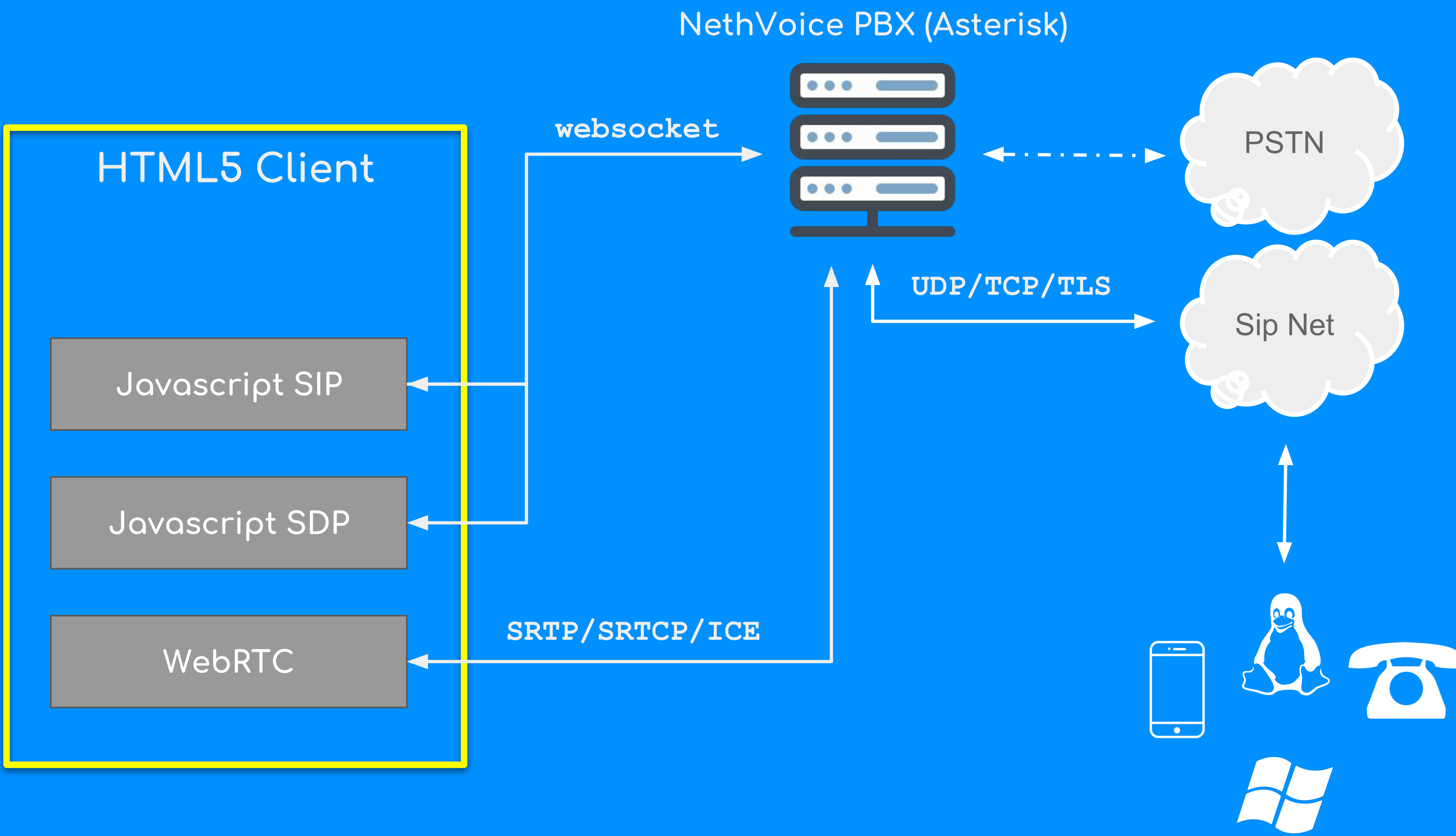
- Reduction / Compression / Decompression of data flow
- Bandwidth / Quality (MOS) / Latency
- Audio
  - **G.711** (64 kbps)
  - **Opus** (6-510 kbps - dynamic bitrate)
- Video
  - VP8, VP9, **AV1**
  - H.264

sipML5

sipML5

- First Open Source HTML5 SIP Client (Doubango Telecom)
- 100% Javascript: NO PLUGIN !!!
- Media stack on WebRTC
- SIP over WebSocket (UDP, TCP, TLS)
- Audio / Video Calls / Instant Messaging / Screen share
- Desktop & Mobile
- Google I/O 2012

sipML5 Architecture



sipML5: how to use

```
SIPml.init

new SIPml.Stack

newSession('register', ...)

newSession('call-audio', ...)
```

1. Engine initialization
2. Start SIP Stack
3. Extension registration
4. Start Audio/Video call

sipML5: how to use

```
SIPml.init

new SIPml.Stack

newSession('register', ...)

newSession('call-audio', ...)
```

1. Engine initialization
2. Start SIP Stack
3. Extension registration
4. Start Audio/Video call

sipML5: how to use

```
SIPml.init

new SIPml.Stack

newSession('register', ...)

newSession('call-audio', ...)
```

1. Engine initialization
2. Start SIP Stack
3. Extension registration
4. Start Audio/Video call

sipML5: how to use

```
SIPml.init

new SIPml.Stack

newSession('register', ...)

newSession('call-audio', ...)
```

1. Engine initialization
2. Start SIP Stack
3. Extension registration
4. Start Audio/Video call

...more code...

sipML5: the library

1. vim

```html
<!doctype html>
<html>
  <head></head>
  <body>
    <script src="lib/SIPml-api.js"></script>
  </body>
</html>
```

1. sipML5: engine initialization

1. vim

```
SIPml.init(engineReadyCb, engineErrorCb);

function engineReadyCb(e) {
  createSipStack();
}

function engineErrorCb(e) {
  console.log(e);
}
```

2. sipML5: start SIP stack

1. vim

```
var sipStack = new SIPml.Stack({
  realm: 'nethvoice.nethesis.it',
  impi: '200',
  impu: 'sip:200@nethvoice.nethesis.it',
  password: 'password',
  display_name: '@ale_polidori',
  enable_rtcweb_breaker: false,
  websocket_proxy_url: 'wss://nethvoice.nethesis.it:8089/ws',
  events_listener: {
    events: '*',
    listener: sipEventsListener
  }
});
sipStack.start();
```

2. sipML5: start SIP stack

1. vim

```
var sipStack = new SIPml.Stack({
  realm: 'nethvoice.nethesis.it',
  impi: '200',
  impu: 'sip:200@nethvoice.nethesis.it',
  password: 'password',
  display_name: '@ale_polidori',
  enable_rtcweb_breaker: false,
  websocket_proxy_url: 'wss://nethvoice.nethesis.it:8089/ws',
  events_listener: {
    events: '*',
    listener: sipEventsListener
  }
});
sipStack.start();
```

2. sipML5: start SIP stack

1. vim

```
var sipStack = new SIPml.Stack({
  realm: 'nethvoice.nethesis.it',
  impi: '200',
  impu: 'sip:200@nethvoice.nethesis.it',
  password: 'password',
  display_name: '@ale_polidori',
  enable_rtcweb_breaker: false,
  websocket_proxy_url: 'wss://nethvoice.nethesis.it:8089/ws',
  events_listener: {
    events: '*',
    listener: sipEventsListener
  }
});
sipStack.start();
```

2. sipML5: start SIP stack

```
var sipStack = new SIPml.Stack({
  realm: 'nethvoice.nethesis.it',
  impi: '200',
  impu: 'sip:200@nethvoice.nethesis.it',
  password: 'password',
  display_name: '@ale_polidori',
  enable_rtcweb_breaker: false,
  websocket_proxy_url: 'wss://nethvoice.nethesis.it:8089/ws',
  events_listener: {
    events: '*',
    listener: sipEventsListener
  }
});
sipStack.start();
```

2. sipML5: start SIP stack

1. vim

```
var sipStack = new SIPml.Stack({
  realm: 'nethvoice.nethesis.it',
  impi: '200',
  impu: 'sip:200@nethvoice.nethesis.it',
  password: 'password',
  display_name: '@ale_polidori',
  enable_rtcweb_breaker: false,
  websocket_proxy_url: 'wss://nethvoice.nethesis.it:8089/ws',
  events_listener: {
    events: '*',
    listener: sipEventsListener
  }
});
sipStack.start();
```

2. sipML5: start SIP stack

1. vim

```
var sipStack = new SIPml.Stack({
  realm: 'nethvoice.nethesis.it',
  impi: '200',
  impu: 'sip:200@nethvoice.nethesis.it',
  password: 'password',
  display_name: '@ale_polidori',
  enable_rtcweb_breaker: false,
  websocket_proxy_url: 'wss://nethvoice.nethesis.it:8089/ws',
  events_listener: {
    events: '*',
    listener: sipEventsListener
  }
});
sipStack.start();
```

3. sipML5: extension registration

1. vim

```
var registerSession = sipStack.newSession(
  'register',
  {
    events_listener: {
      events: '*',
      listener: registerEventsListener
    }
  });
registerSession.register();
```

4. sipML5: call

1. vim

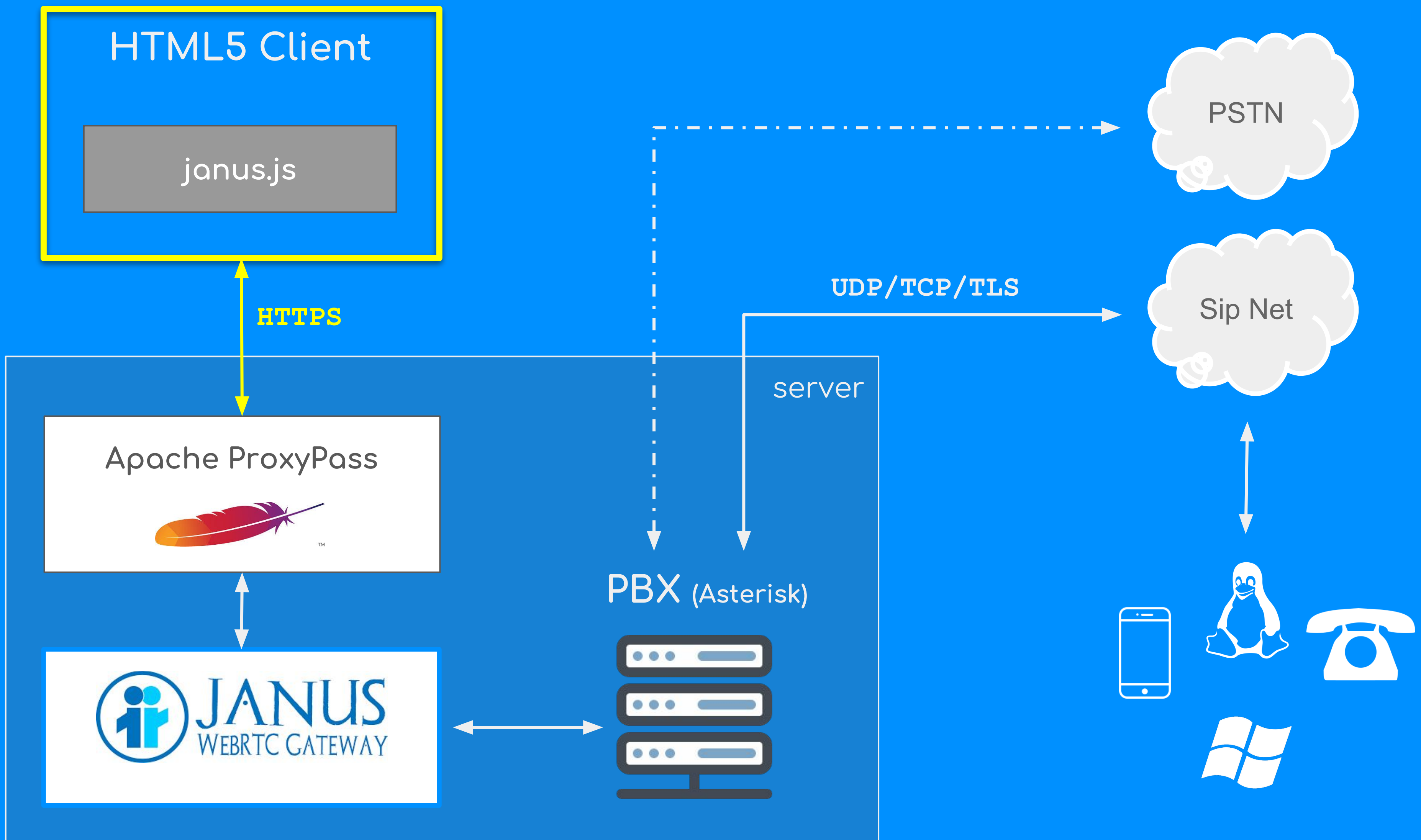
```
var callSession = sipStack.newSession(
  'call-audiovideo',
  {
    video_local: document.getElementById('video-local'),
    video_remote: document.getElementById('video-remote'),
    audio_remote: document.getElementById(audioId),
    events_listener: {
      events: '*',
      listener: callEventsListener
    }
  });
callSession.call('221');
```

Janus Gateway

Janus

- Gateway general purpose by **Meetecho**
- WebRTC audio/video
- JSON messages
- Plugin Architecture → SIP Plugin
- Monitoring
- Interfaces HTTP, WebSocket, RabbitMQ

Janus Architecture



ale_polidori

Janus: how to use

```
Janus.init

new Janus({...});

Janus.attach({...});

pluginHandle.createOffer
```

1. Engine initialization
2. Create a session
3. Link SIP plugin
4. Start Audio/Video call

Janus: how to use

```
Janus.init

new Janus({...});

Janus.attach({...});

pluginHandle.createOffer
```

1. Engine initialization
2. Create a session
3. Link SIP plugin
4. Start Audio/Video call

Janus: how to use

1. vim

```
Janus.init

new Janus({...});

Janus.attach({...});

pluginHandle.createOffer
```

1. Engine initialization
2. Create a session
3. Link SIP plugin
4. Start Audio/Video call

Janus: how to use

```
Janus.init

new Janus({...});

Janus.attach({...});

pluginHandle.createOffer
```

1. Engine initialization
2. Create a session
3. Link SIP plugin
4. Start Audio/Video call

...more code...

Library

- Janus Client lib

  meetecho/janus-gateway

- WebRTC adapter

  webrtc/adapter

```html
<!doctype html>
<html>
  <head>
  </head>
  <body>
    <script src="lib/janus.js"></script>
    <script src="lib/adapter.js"></script>
  </body>
</html>
```

1. Janus: engine initialization

1. vim

```
Janus.init({
  debug: 'all',
  callback: function () {
    console.log('Initialized');
  }
});
```

2. Janus: create a session

1. vim

```
janus = new Janus({
  server: 'https://server/janus',
  success: function () {
    // attach to sip plugin
  },
  error: function (error) {
    // ...
  },
  destroyed: function () {
    // ...
  }
});
```

3. Janus: link SIP plugin

1. vim

```
janus.attach({
  plugin: 'janus.plugin.sip',
  success: function (pluginHandle) {
    // ...
  },
  error: function (error) {
    // ...
  },
  onmessage: function (msg, jsep) {
    // ...
  },
  onlocalstream: function (stream) {
    // ...
  },
  onremotestream: function (stream) {
    // ...
  },
  oncleanup: function () {
    // ...
  }
});
```

Handle to interact with plugin

4. Janus: audio/video call

```
sipcall.createOffer({
  media: {
    audioSend: true,
    audioRecv: true,
    videoSend: true,
    videoRecv: true
  },
  success: function (jsep) {
    var body = {
      request: 'call',
      uri: 'sip:200@server'
    };
    sipcall.send({
      'message': body,
      'jsep': jsep
    });
  },
  error: function (error) {
    Janus.error('WebRTC error...', error);
  }
});
```

Call destination

VoIP PBX

NethServer VoIP PBX

- NethServer Linux distro
- **Asterisk & FreePBX** based
- Open Source - community.nethserver.org
- **NethVoice** Enterprise version
- **NethCTI** WebApp

Demo

1 Start your NethServer VoIP PBX

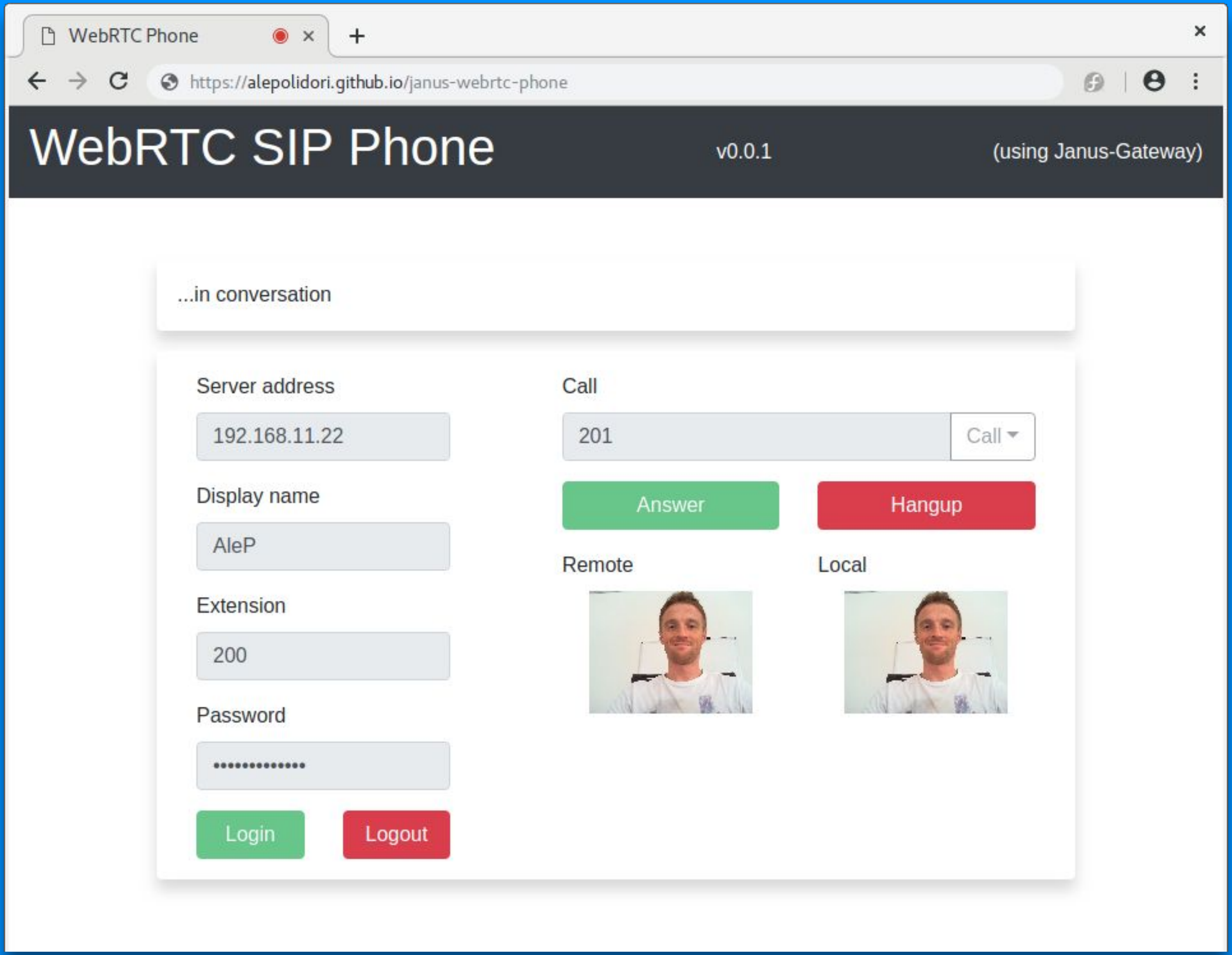
1. bash

```
$ git clone https://github.com/alepolidori/vagrant-files.git
$
$ cd vagrant-files/nethserver-freepbx-14.0.3.6
$
$ vagrant up
```

2

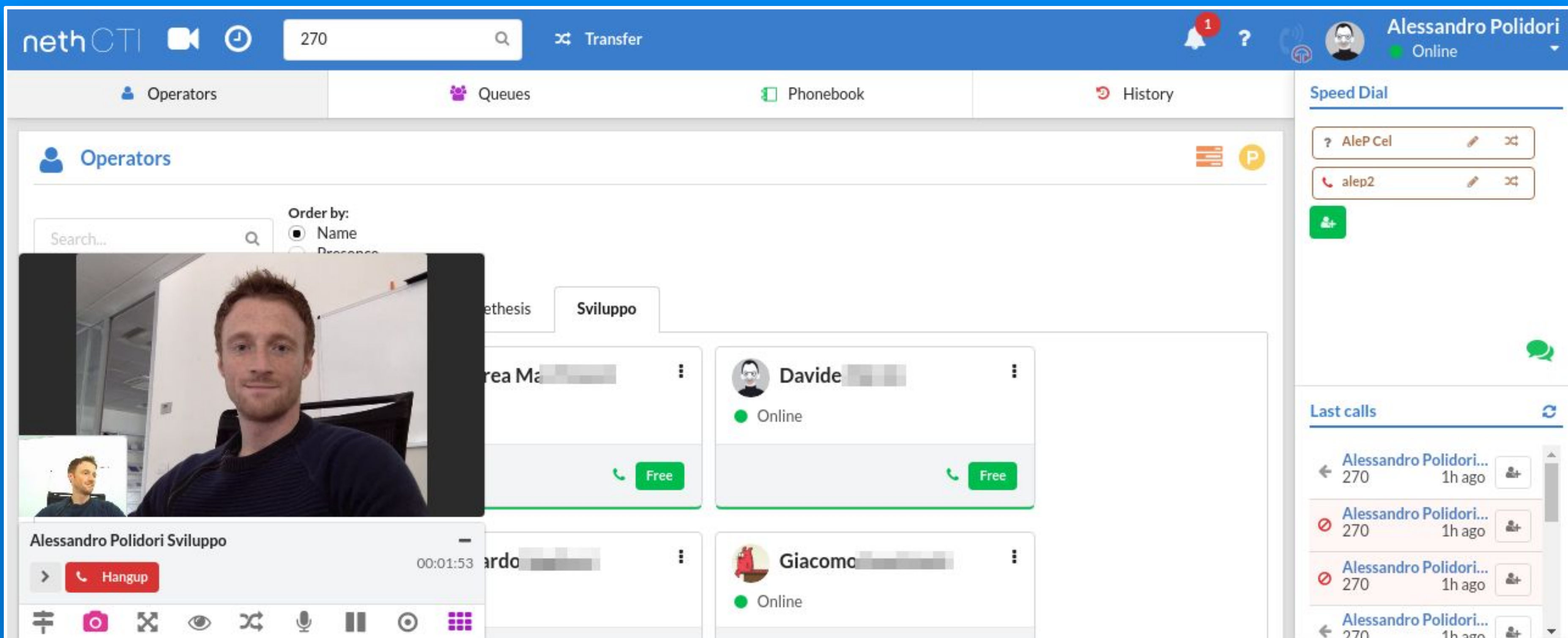
go to:

https://alepolidori.github.io/janus-webrtc-phone



ale_polidori

nethCTI


nethCTI
270
Transfer
1
Alessandro Polidori
Online
Operators
Queues
Phonebook
History
Speed Dial
AleP Cel
alep2
Operators
Search...
Order by:
Name
ethesis
Sviluppo
Davide
Online
Free
Free
Giacomo
Online
Alessandro Polidori Sviluppo
00:01:53
Hangup
Last calls
Alessandro Polidori...
270
1h ago
Alessandro Polidori...
270
1h ago
Alessandro Polidori...
270
1h ago
Alessandro Polidori...

References

https://github.com/alepolidori/fosdem-2019

https://bloggeek.me

Thank you !

alepolidori

@ale_polidori

https://medium.com/@ale_polidori

nethesis