WebRTC

Christoph Betschart

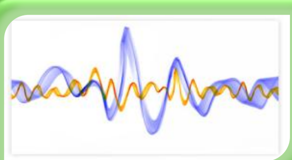
Dario Maggi

CS561 Seminar: Verteilte Systeme
19.11.2013

Index

- Aquiring
  - Audio and Video
  - getUserMedia()
- Communicating
  - Audio and Video
  - RTCPeerConnection()
- Communicating
  - Arbitrary Data
  - RTCDataChannel()
- Utilities
- Beyond Browsers

RTCPeerConnection



Signal Processing

Codec handling

Peer to Peer communication

Bandwith management

Security

Offer/Answer model and Signalling

- Metadata exchange:
  - Media information (Codecs…)
  - Network information (IP, Port, …)
- With session description protocol (SDP)
  - Place to go, if you wan‘t to choose the settings
- Network information can be sent splitted in candidates. (ICE Candidate Trickling)

```
m=audio 52705 RTP/SAVPF 109 0 8 101
a=rtpmap:109 opus/48000/2
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 1692467199 87.102.133.31 52705 typ srflx raddr 192.168.1.46 rport 52705
```

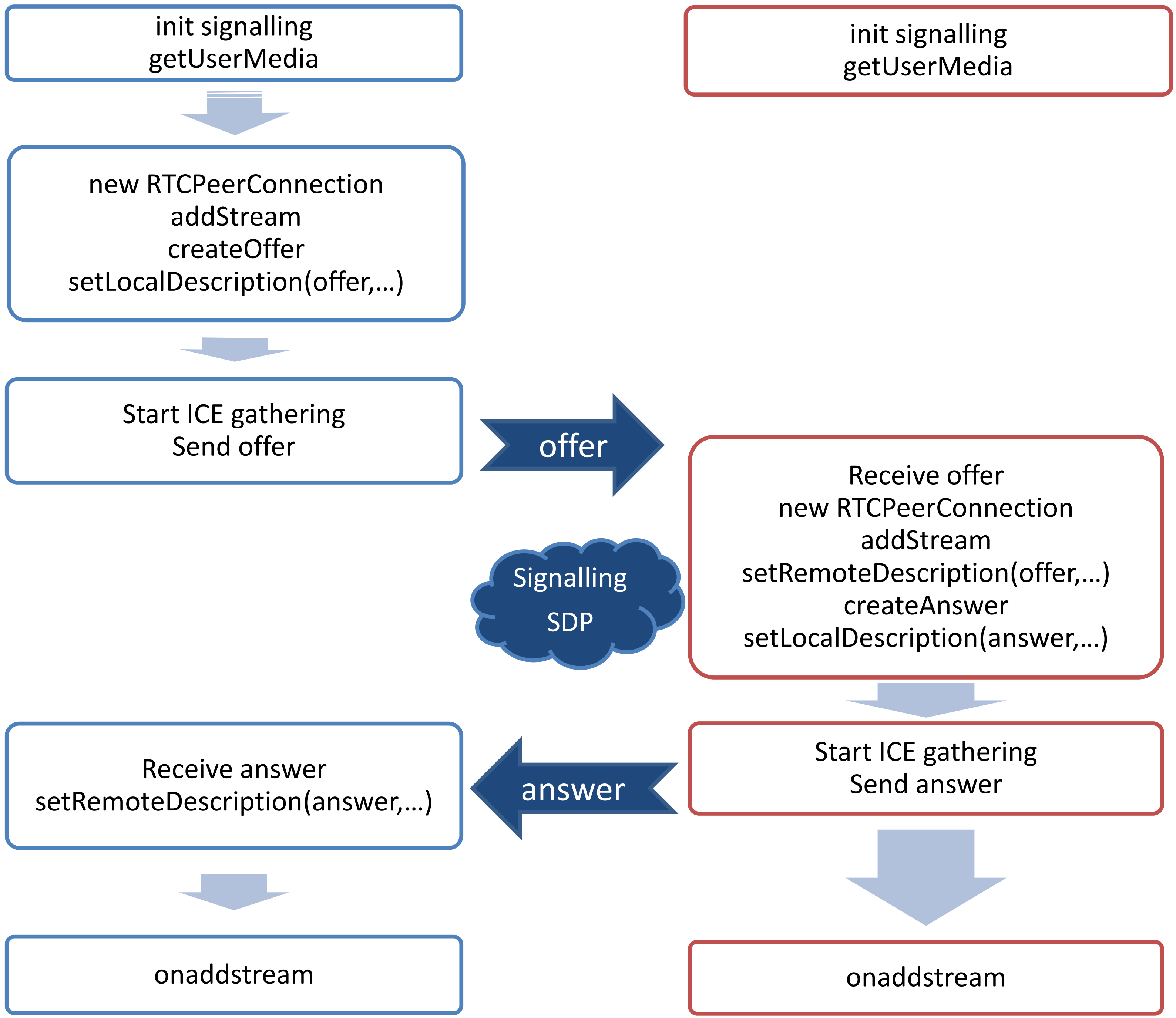
Signalling, server-side

- Nodejs example with socket.io

```
socket.on('send',function(evt){
  socket.broadcast.emit('onmessage',evt);
});
```

RTCPeerConnection

- Audio/Video connection runs when:
  - Both clients have the local and remote session description
  - A stream has been added to the RTCPeerConnection
  - Network allows communication


init signalling
getUserMedia
init signalling
getUserMedia
new RTCPeerConnection
addStream
createOffer
setLocalDescription(offer,...)
Start ICE gathering
Send offer
offer
Receive offer
new RTCPeerConnection
addStream
setRemoteDescription(offer,...)
createAnswer
setLocalDescription(answer,...)
Signalling
SDP
Start ICE gathering
Send answer
answer
Receive answer
setRemoteDescription(answer,...)
onaddstream
onaddstream

Example implementation

```
pc = new RTCPeerConnection(conf);
pc.onaddstream = gotRemoteStream;
pc.addStream(localStream);
pc.createOffer(createdLocalDesc,logError,mediaConstraints);
function createdLocalDesc(desc) {
   pc.setLocalDescription(desc);
   sigChan.send(desc);
}
function gotRemoteDesc(desc) {
   pc.setRemoteDescription(desc);
   if(desc.type=='offer'){
       pc.createAnswer(createdLocalDesc,logError,mediaConstraints);
   }
}
function gotRemoteStream(e) {
   remoteVidElem.src = URL.createObjectURL(e.stream);
}
```

ICE

setLocalDescription triggers ICE gathering

Client 1:

Every time a candidate is found onicecandidate is fired

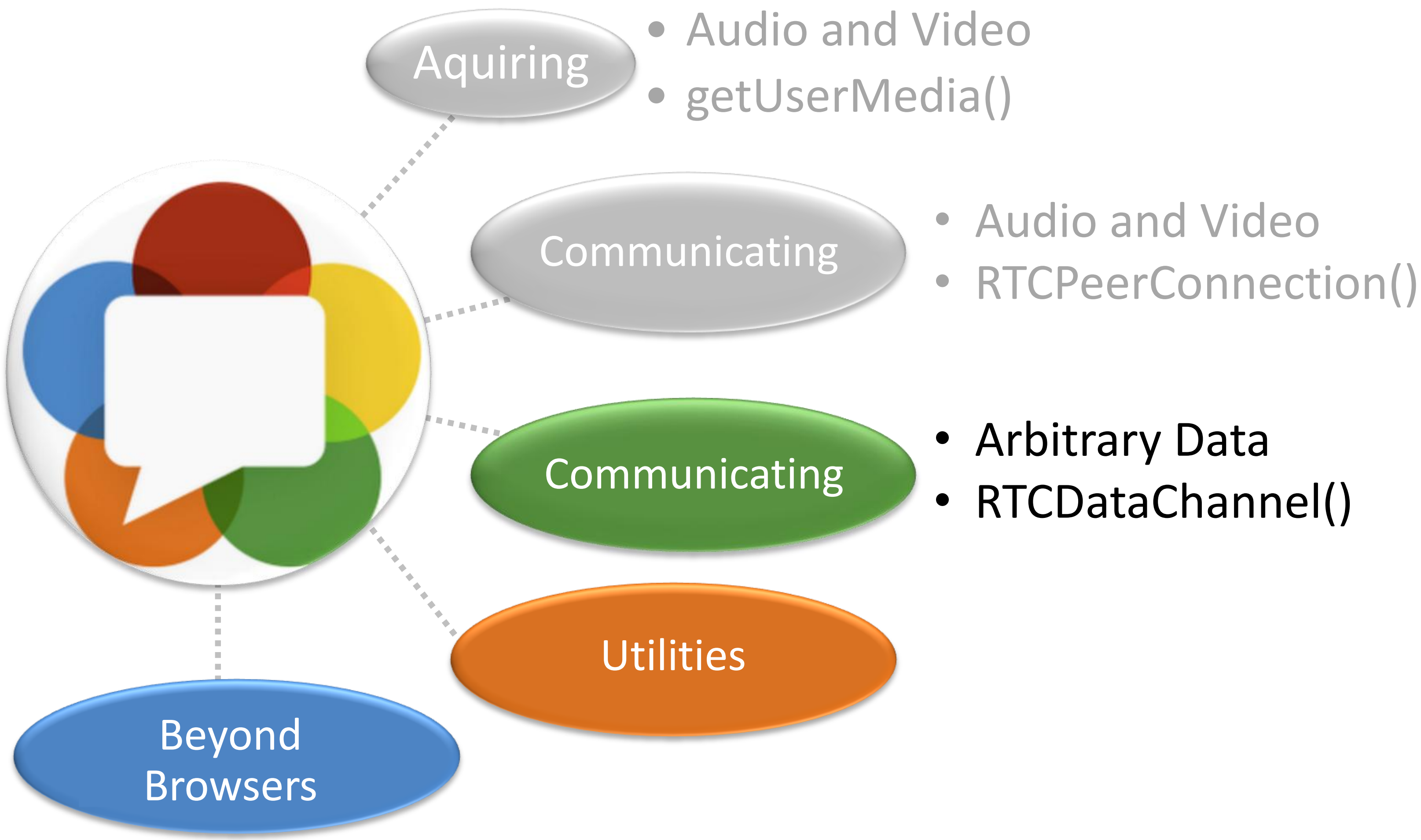
Client 2:

```
var candidate = new RTCIceCandidate(recv_candidate);
pc.addIceCandidate(candidate);
```

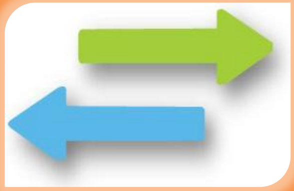
Example

http://87.102.133.31:2013/cs561/

http://www.simpl.info/rtcpeerconnection/


Aquiring
• Audio and Video
• getUserMedia()
Communicating
• Audio and Video
• RTCPeerConnection()
Communicating
• Arbitrary Data
• RTCDataChannel()
Utilities
Beyond Browsers

RTCDataChannel



Bidirectional Communication

API similar as Websockets

Unreliable or Reliable

Secure

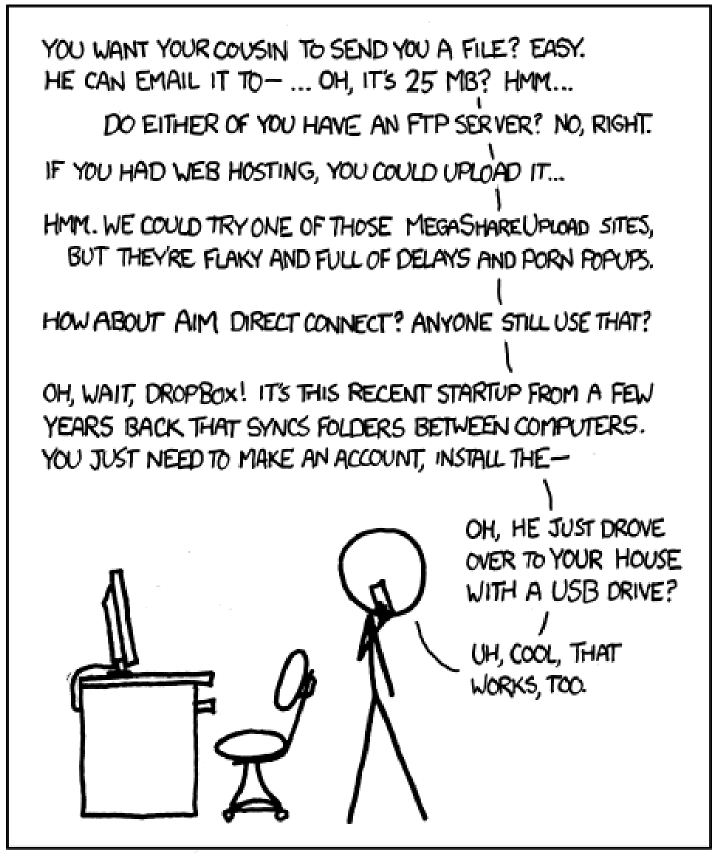
RTCDataChannel: Example

```
var pc = new RTCPeerConnection(servers);

pc.ondatachannel = function(event) {
   receiveChannel = event.channel;
   receiveChannel.onmessage = function(event){
        document.querySelector("div#receive").innerHTML = event.data;
   };
};

sendChannel = pc.createDataChannel("sendDataChannel", {reliable: false});

document.querySelector("button#send").onclick = function ({
   var data = document.querySelector("textarea#send").value;
   sendChannel.send(data);
};
```

RTCDataChannel: Example

Simpl.info/rtcdatachannel

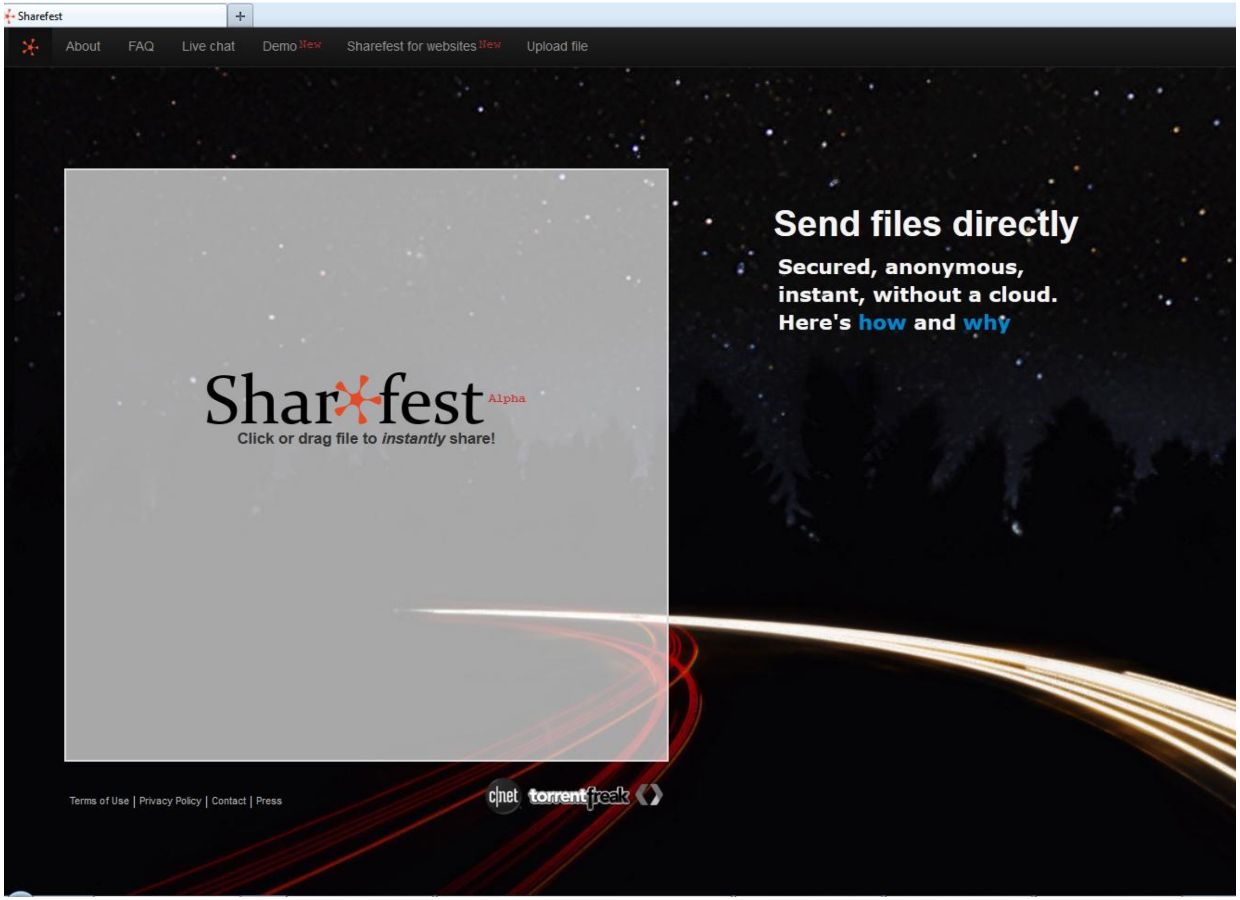
RTCDataChannel



I LIKE HOW WE'VE HAD THE INTERNET FOR DECADES, YET "SENDING FILES" IS SOMETHING EARLY ADOPTERS ARE STILL FIGURING OUT HOW TO DO.

RTCDataChannel: Example II

[Sharefest]

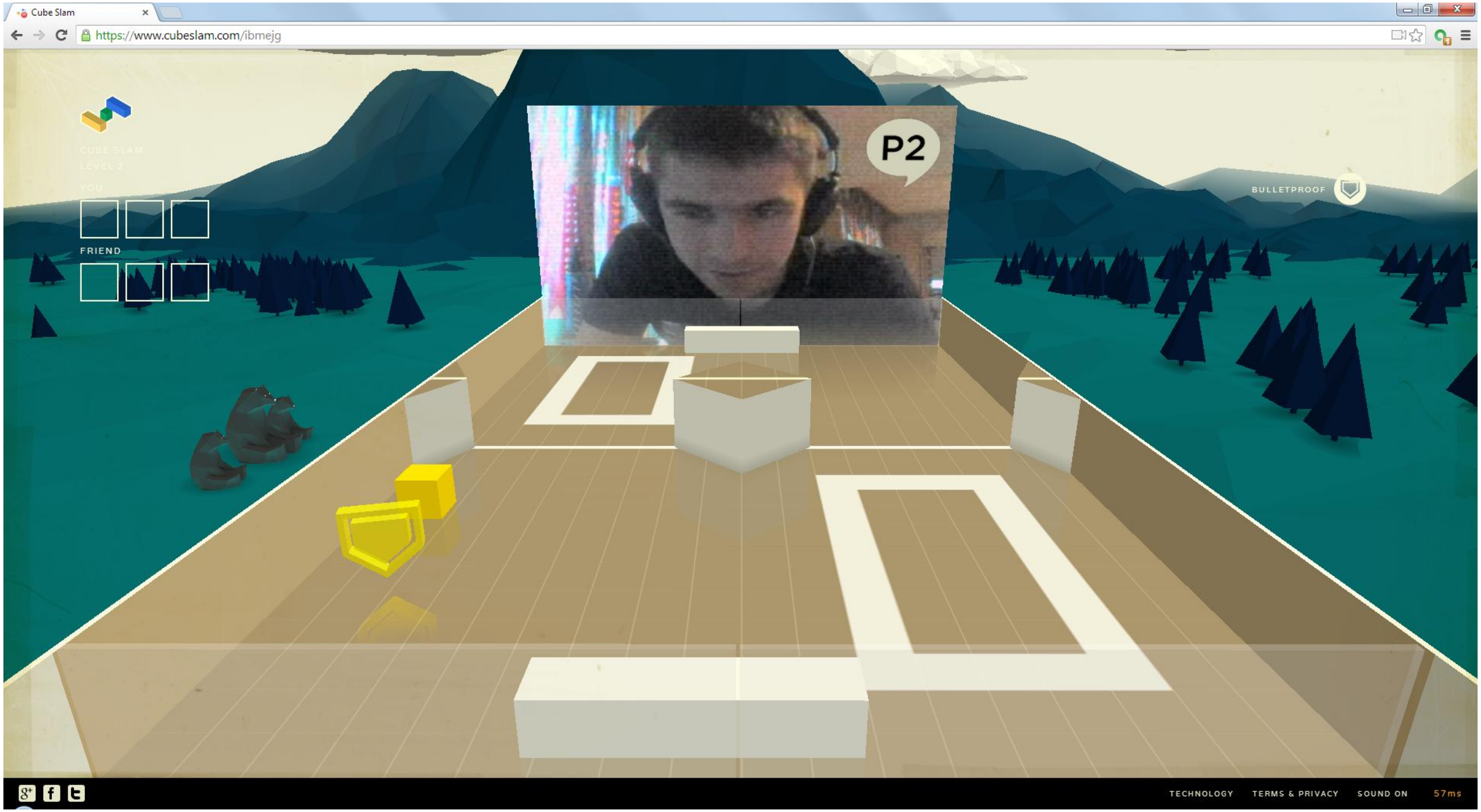


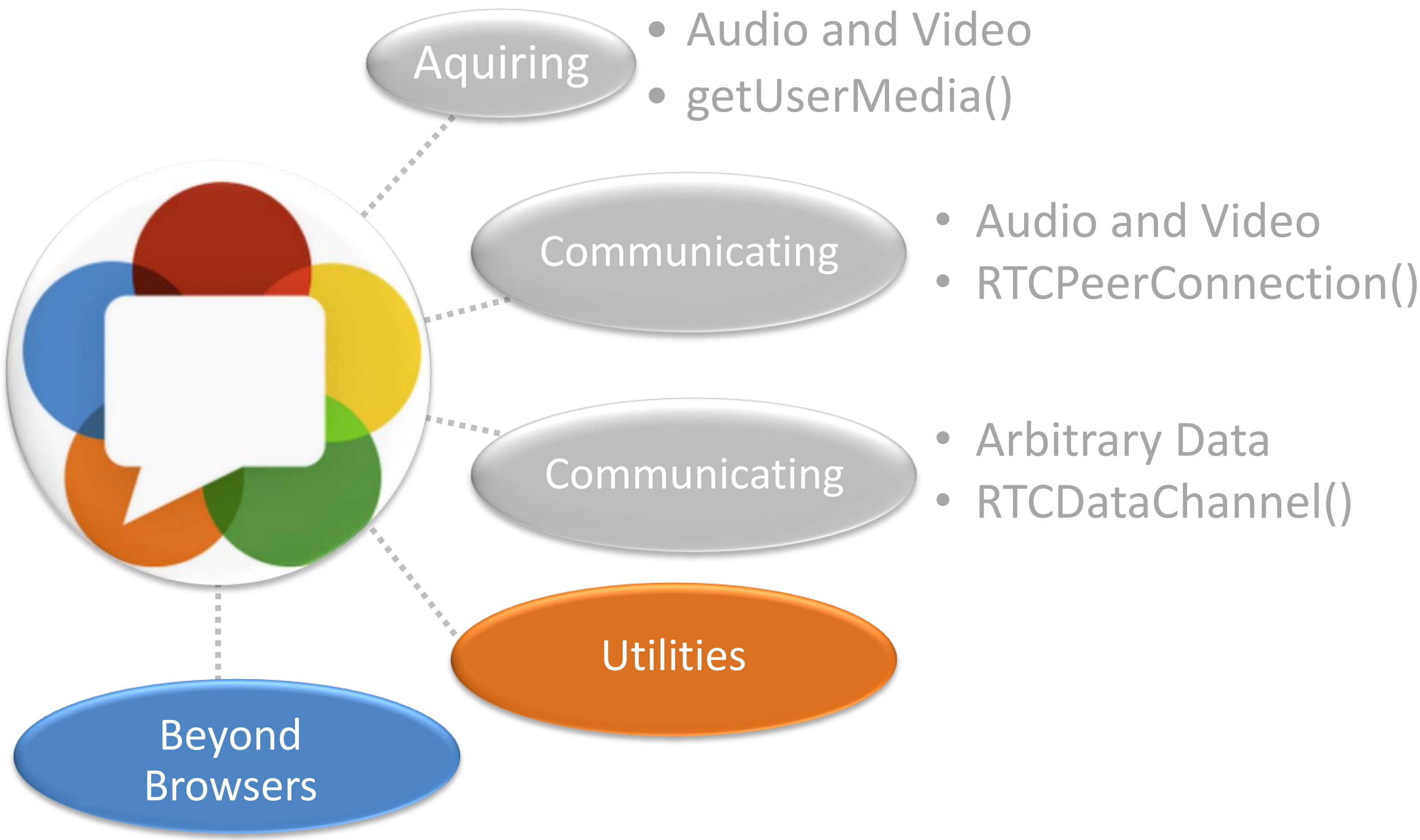
RTCDataChannel: Example III

BananaBread

RTCDataChannel: Example IV

[CubeSlam](https://www.cubeslam.com)

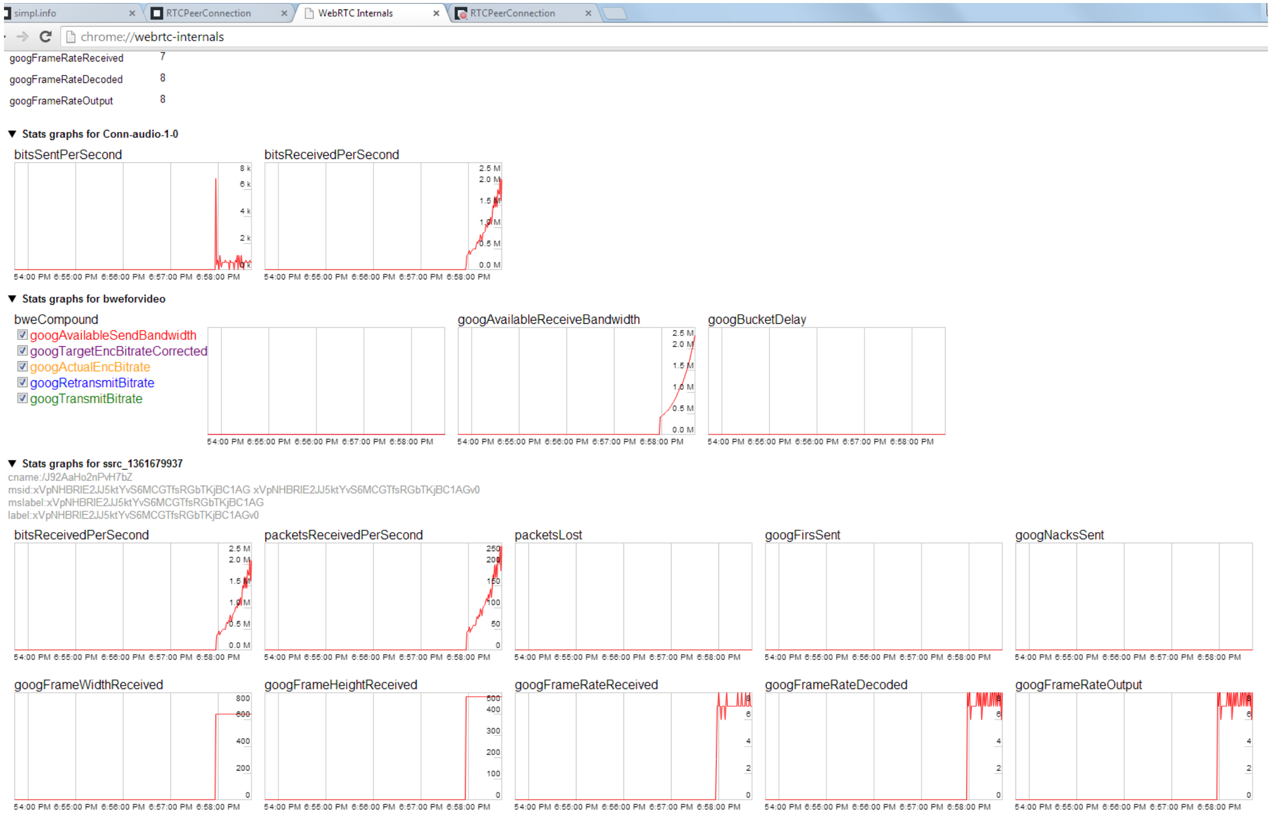




Aquiring
Audio and Video
getUserMedia()
Communicating
Audio and Video
RTCPeerConnection()
Communicating
Arbitrary Data
RTCDataChannel()
Utilities
Beyond Browsers

Utilities I

- Chrome://webrtc-internals



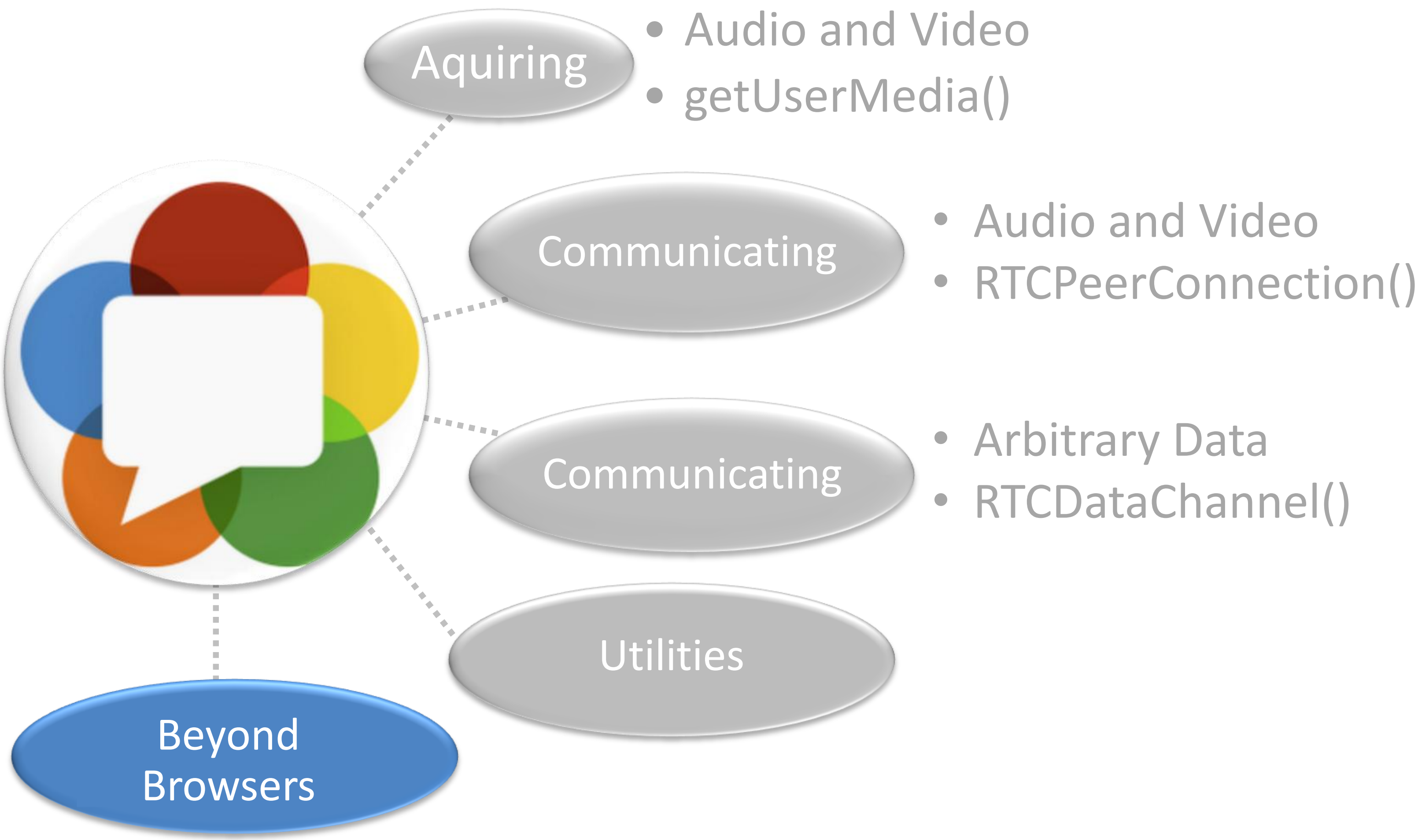
Utilities II

- Adapter.js

W3C Standard	Chrome	Firefox
getUserMedia	webkitGetUserMedia	mozGetUserMedia
RTCPeerConnection	webkitRTCPeerConnection	mozRTCPeerConnection
RTCSessionDescription	RTCSessionDescription	mozRTCSessionDescription
RTCIceCandidate	RTCIceCandidate	mozRTCIceCandidate

Utilities III

- JavaScript frameworks:
  - Video:
    - SimpleWebRTC, easyRTC, webRTC.io
  - Peer-to-peer data:
    - PeerJs



Aquiring
Audio and Video
getUserMedia()
Communicating
Audio and Video
RTCPeerConnection()
Communicating
Arbitrary Data
RTCDataChannel()
Utilities
Beyond Browsers

Beyond Browsers

- Communication to:
  - telephones or VoIP systems with gateway servers
    - [Zingaya], [Tethr] and [OpenBTS]
  - apps (iOS or Android) with native library

Web RTC

Sources

- www.webrtc.org/
- Tutorial: http://www.html5rocks.com/en/tutorials/webrtc/basics/
- WebRTC: http://www.w3.org/TR/webrtc/
- SDP: http://tools.ietf.org/id/draft-nandakumar-rtcweb-sdp-01.html