

# Build Web Based Video Call Application Using WebRTC

Fitra Aditya  
Sabtu, 24 Oktober 2015

- Fitra Aditya
- *Software Engineer* di **qiscus** 



**WebRTC**

# WebRTC

- WebRTC
  - Protokol
  - *Open source project*

# WebRTC

- WebRTC merupakan protokol yang memungkinkan *web browser* untuk melakukan panggilan audio, video, serta pertukaran data lainnya menggunakan jalur *peer-to-peer* tanpa perlu tambahan *plugin*.

# WebRTC



Web  RTC



Audio and Video Communication in the browser

# WebRTC

- WebRTC juga merupakan proyek *open source* yang menyediakan sekumpulan API bagi *web browser* dan aplikasi *mobile* untuk melakukan komunikasi secara *real-time*.
- Proyek WebRTC digawangi oleh Google, serta didukung oleh Mozilla, Opera, serta beberapa perusahaan lainnya.



Chrome



Firefox



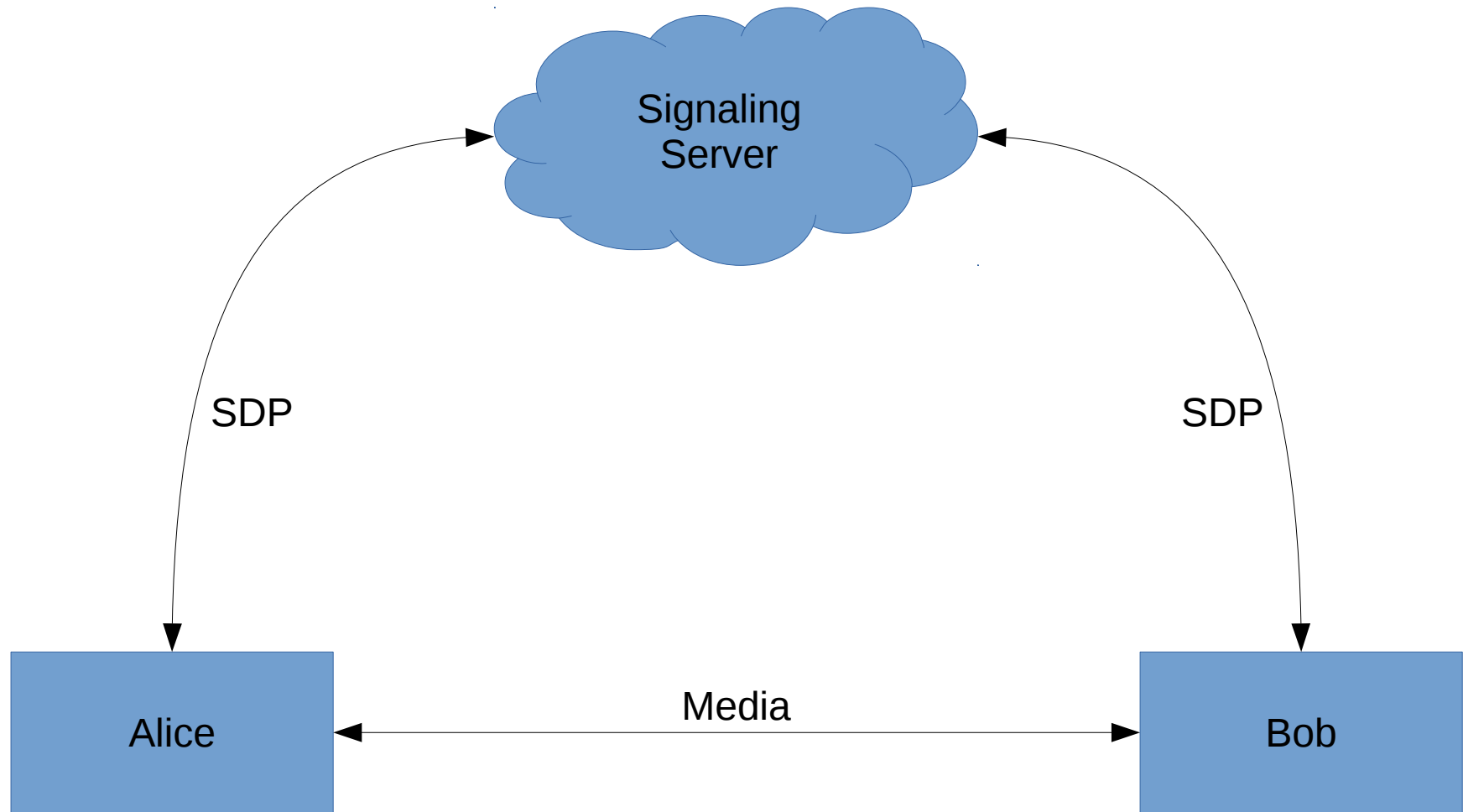
Opera



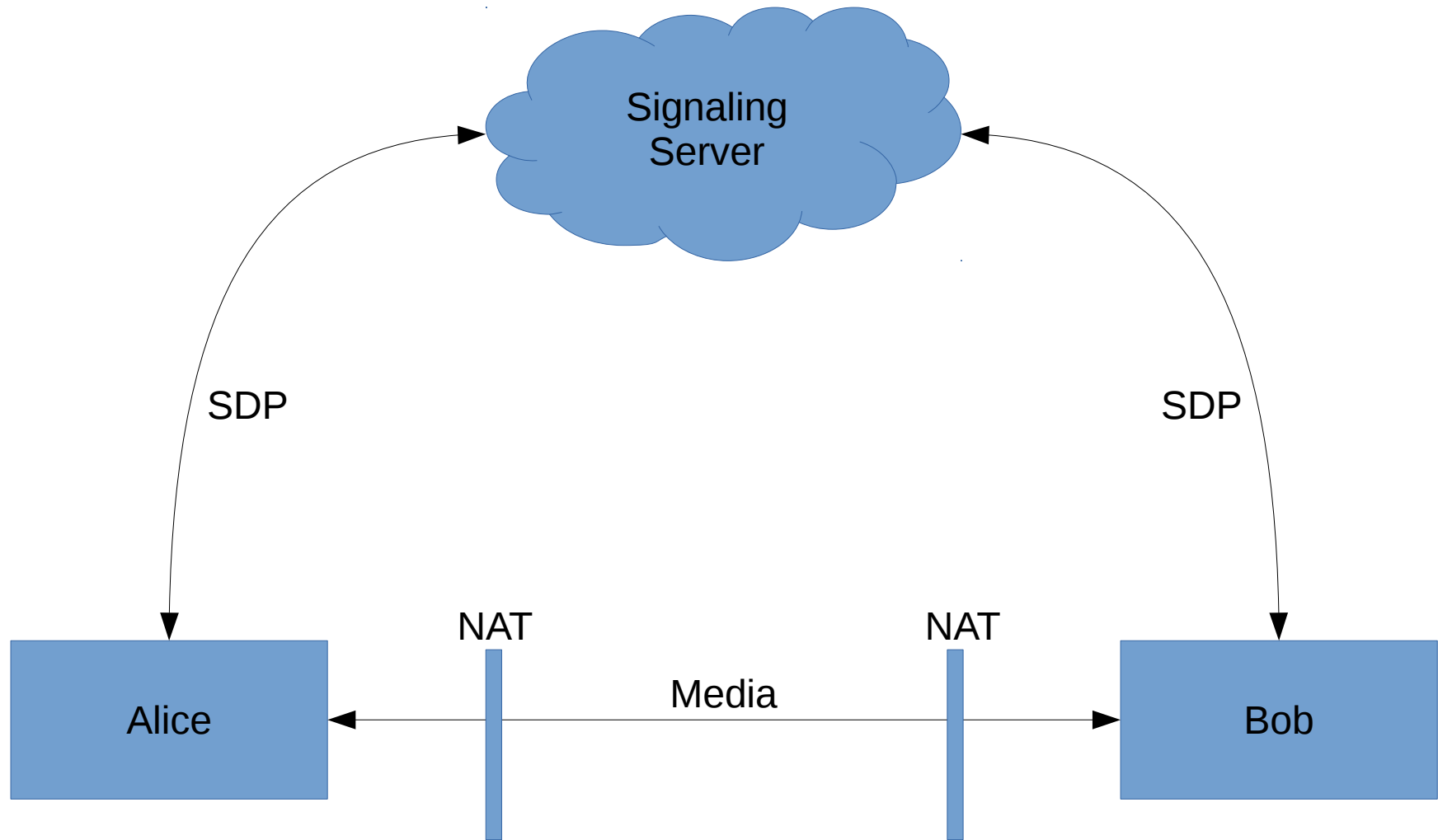
# Mengapa WebRTC?

- Cukup menggunakan *web browser*
- Tidak perlu menggunakan plugin tambahan
- *Open source*

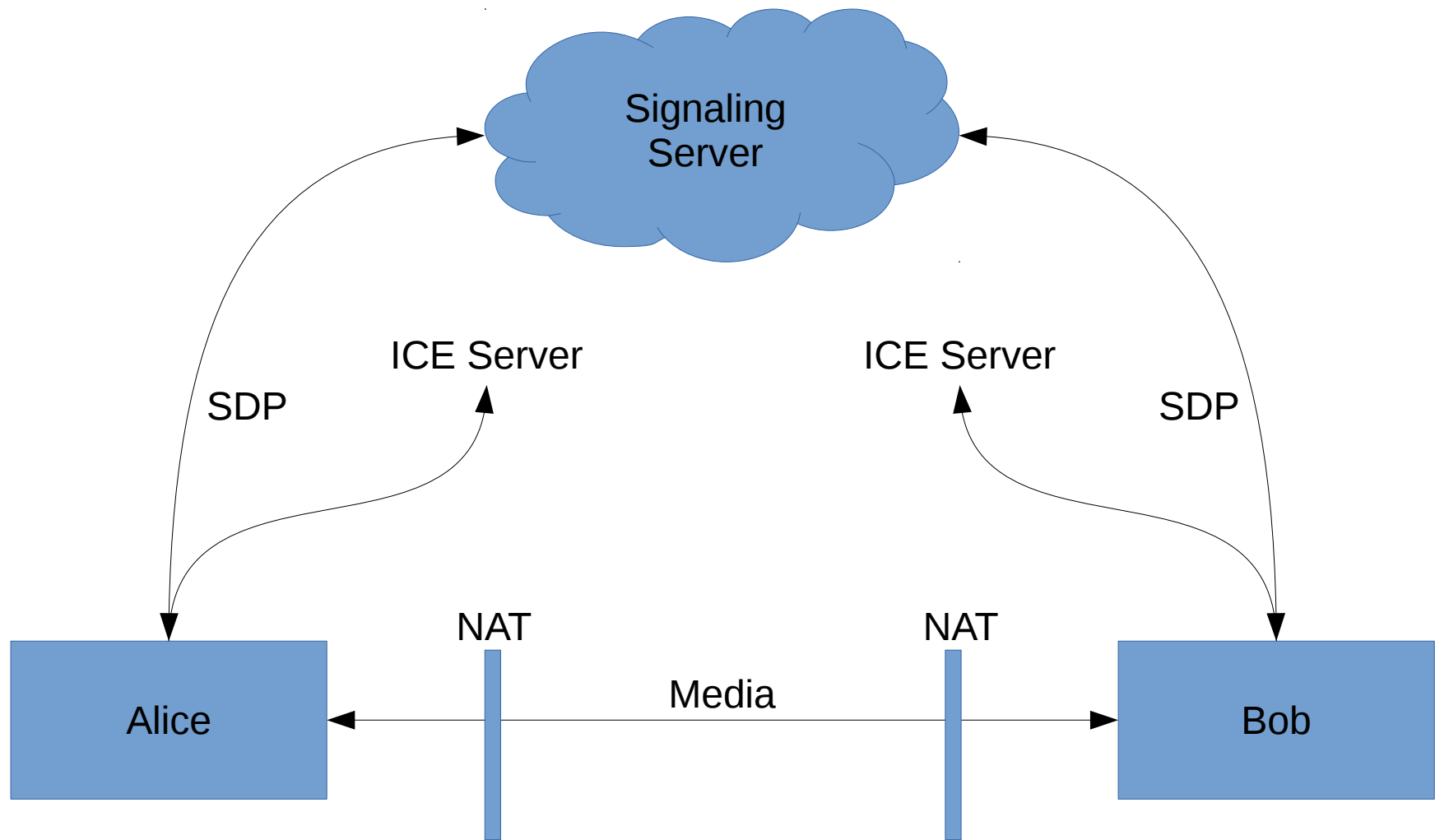
# Bagaimana WebRTC Bekerja?



# Bagaimana WebRTC Bekerja?



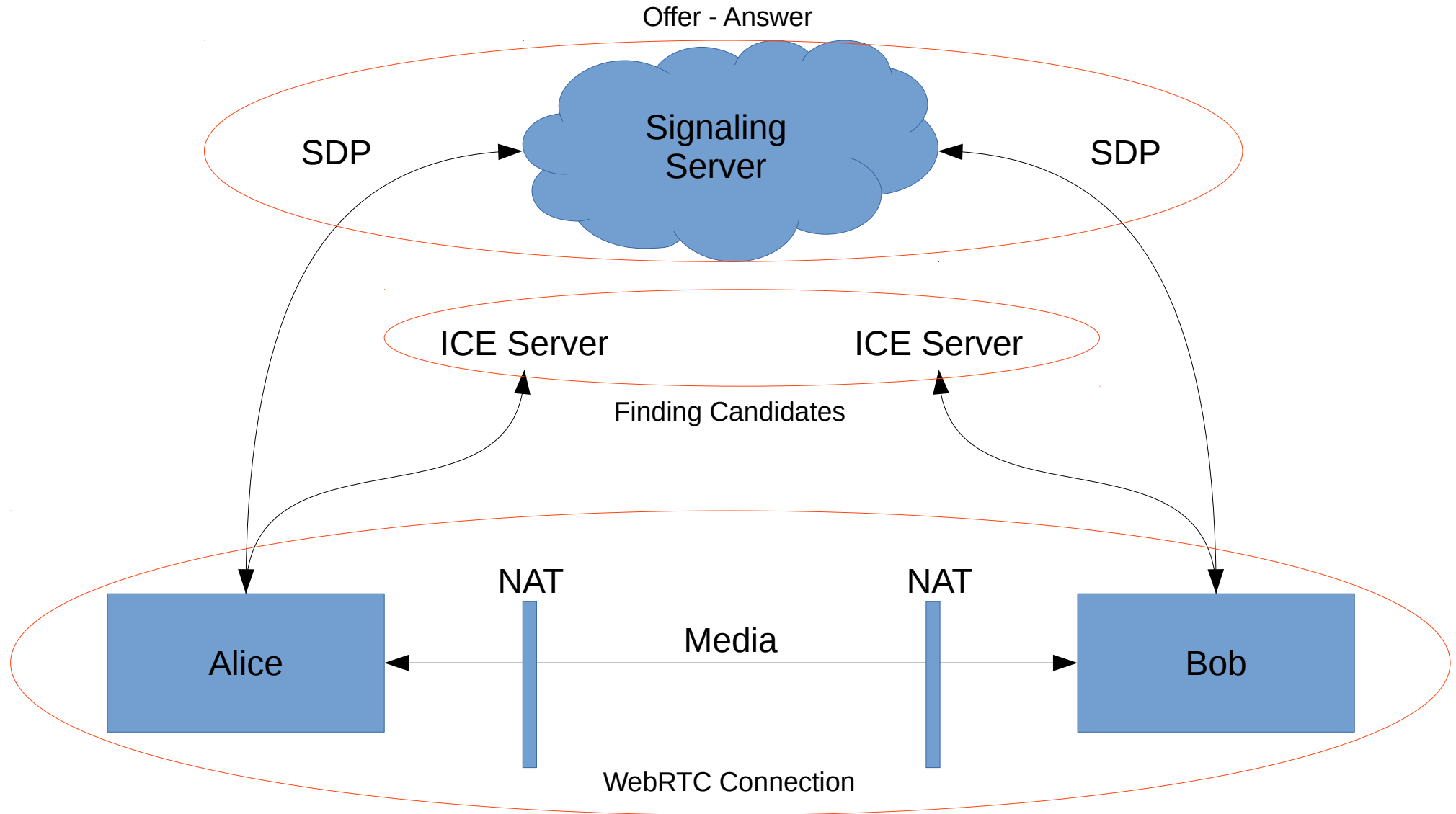
# Bagaimana WebRTC Bekerja?



# Bagaimana WebRTC Bekerja?

- Misalkan Alice dan Bob ingin melakukan *video chat* menggunakan WebRTC
- Alice dan Bob perlu bertukar informasi mengenai informasi jaringan internet
- Proses ini dinamakan dengan '*finding candidates*'
- Alice dan Bob juga perlu bertukar informasi mengenai media yang dimiliki
- Proses ini dinamakan dengan '*offer – answer*'

# Bagaimana WebRTC Bekerja?



# Bagaimana WebRTC Bekerja?

- *Finding candidates*
  - Alice
    - Buat *peer object* untuk mendapatkan *candidates* menggunakan ICE (*interactive connectivity establishment*) → ICE *candidates*
    - Kirim ICE *candidates* menggunakan *signaling channel* (contoh: websocket)
  - Bob
    - Simpan ICE *candidates* yang diterima untuk digunakan kemudian

# Bagaimana WebRTC Bekerja?

## Alice

```
pc = new RTCPeerConnection(...);  
pc.onicecandidate = function (event) {  
    ...  
};
```

## Bob

```
pc = new RTCPeerConnection(...);  
pc.addIceCandidate(new RTCIceCandidate(...));
```



# Bagaimana WebRTC Bekerja?

- *Offer – answer*
  - Alice
    - Panggil *createOffer()* → *localDescription*
    - Kirim *localDescription* ke Bob
  - Bob
    - Simpan *localDescription* yang dikirim oleh Alice → *remoteDescription*
    - Panggil *createAnswer()* → *localDescription*
    - Kirim *localDescription* ke Alice
  - Alice
    - Simpan *localDescription* yang dikirim oleh Bob → *remoteDescription*
  - Ping!

# Bagaimana WebRTC Bekerja?

```
if (isCaller)
    pc.createOffer(gotDescription);
else
    pc.createAnswer(pc.remoteDescription, gotDescription);


function gotDescription(localdescription) {
    pc.setLocalDescription(localdescription);
    // send localdescription
}

if (localdescription) {
    pc.setRemoteDescription(
        new RTCSessionDescription(localdescription)
    );
}
```



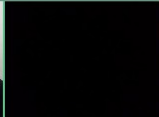

Qiscus | Messaging For G... Qiscus Videocall simplepeer - Settings | He... Transferring Apps | Herok... video\_controller.rb home.erb.html Qiscus | Messaging For G... ruby on rails - Error :The... Layouts and Rendering in... Afief

videocall.qiscus.us/room/p7yfgndy0lt6lytp26jrf64dws4

Apps SnsMed Qiscus Qiscus css js android vim emacs python others



End <http://videocall.qiscus.us/room/p7yfgndy0lt6lytp26jrf64dws4>



Q Elements Network Sources Timeline Profiles Resources Audits Console AngularJS

<top frame> Preserve log

1 message is hidden by filters. [Show all messages.](#)

⚠

getUserMedia() is deprecated on insecure origins, and support will be removed in the future. You should consider switching your application to a secure origin, such as HTTPS. See <https://goo.gl/rStGz> for more details.

adapter.js:402

> peers

< ▶ Object {RD0GxUdXP8GzhtAAAN: r, \_x2FyiEeLg8mwT4fAAAO: r, Vf1DYUH068k8LBGPAAAN: r}

>

Windows Taskbar

js torrent Qiscus Videocall - ... Cmdr C:\Users\Afief\Doc... Task Manager GitHub

16.00 08/09/2015

DEMO

# 15 Menit dengan WebRTC

- Membangun aplikasi one-to-one video call berbasis web dengan WebRTC
  - Kemampuan HTML dan Javascript dibutuhkan untuk membangun aplikasi ini

# 15 Menit dengan WebRTC

- Langkah-langkah
  - Ambil audio dan video menggunakam *microphone* dan webcam
  - Buat sambungan *peer-to-peer* antar *web browser*
  - Kirim video antar *peer*

# getUserMedia()

- Merupakan API untuk mengambil video maupun audio dari *hardware* (webcam dan *microphone*) untuk ditampilkan ke dalam halaman web
- Tiap *web browser* memiliki implementasi yang berbeda
  - navigator.getUserMedia
  - navigator.webkitGetUserMedia
  - navigator.mozGetUserMedia

# getUserMedia()

- Contoh: [geekcamp-webrtc/satu/index.html](http://geekcamp-webrtc/satu/index.html)

```
navigator.getUserMedia({ video: true, audio: true }, successCallback, errorCallback);

function successCallback (stream) {
  var video = document.querySelector('#localVideo');
  video.src = window.URL.createObjectURL(stream);
}
function errorCallback (error) {
  alert('getUserMedia error: ', error);
}
```



# RTCPeerConnection()

- Merupakan API untuk membangun sambungan *peer-to-peer* antar *web browser*
  - ICE *candidates*
  - NAT *traversal*
- Sambungan *peer-to-peer* ini digunakan untuk melakukan panggilan audio maupun video
  - Termasuk juga *encode / decode* audio dan video

# RTCPeerConnection()

- Membutuhkan *signaling server* untuk membuat sambungan *peer-to-peer* di awal
  - Websocket / Socket.io
- Membutuhkan ICE *servers* untuk mendapatkan ICE *candidates*
  - STUN / TURN

# RTCPeerConnection()

- Contoh tanpa signaling server dan ICE servers: [geekcamp-webrtc/dua/index.html](https://geekcamp-webrtc/dua/index.html)
- Contoh menggunakan signaling server dan ICE servers: [geekcamp-webrtc/tiga/index.html](https://geekcamp-webrtc/tiga/index.html)

# Open Source WebRTC Library

- SimpleWebRTC
- EasyRTC
- RTC.io

# Referensi

- [www.html5rocks.com/en/tutorials/getusermedia/intro/](http://www.html5rocks.com/en/tutorials/getusermedia/intro/)
- <http://www.html5rocks.com/en/tutorials/webRTC/basics/>
- <http://www.webrtc.org/>

Terima Kasih