WebRTC With Java

Binod PG, Architect, Oracle
Amitha Pulijala, Oracle Product Management

Communications Business Unit
October 27, 2015

JavaOne
ORACLE

Safe Harbor Statement

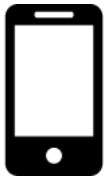
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Program Agenda

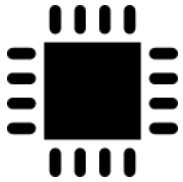
1. WebRTC Introduction
2. WebRTC Architecture Components
3. Java in WebRTC Eco-System
4. Common Development/Architecture Issues
5. Demo

Real Time Communications Meets The WEB

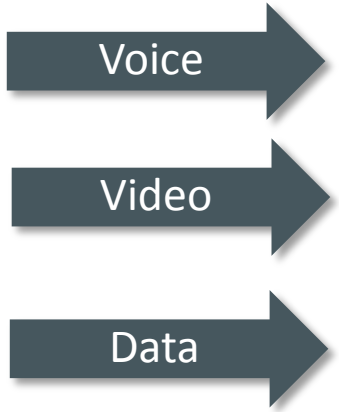
Browsers



Mobile



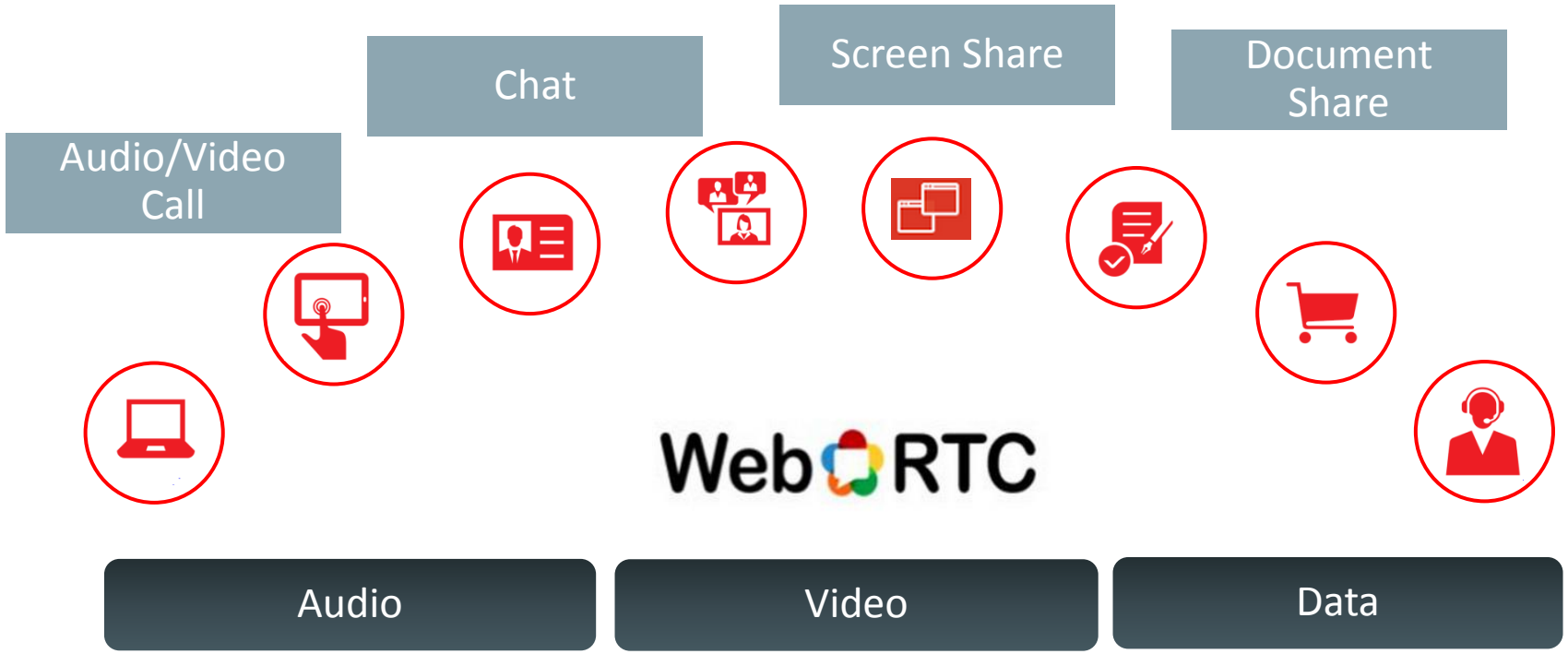
Machine to Machine/Embedded



- A state-of-the-art audio/video/data communication stack in your WEB*client
- Communications at web speed
- Billions of end-points
- Allows communications to be integrated into the “web” experience

WEB here is referred in a broader sense than just browsers

Communications as a feature

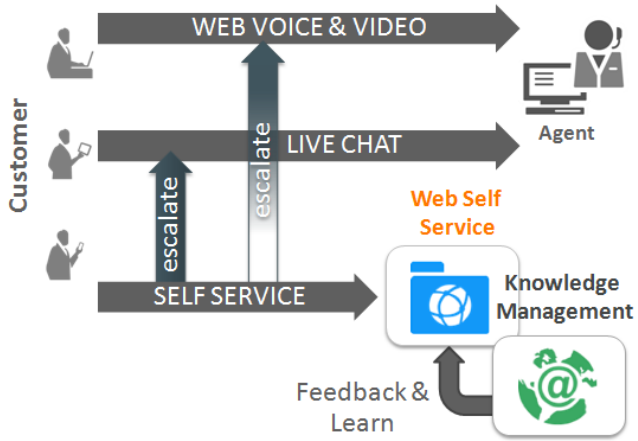


- Context Awareness
- Personalized Service
- Connected Interactions
- Rewarding Relationship

Connecting Customer's journey through your application with Communications

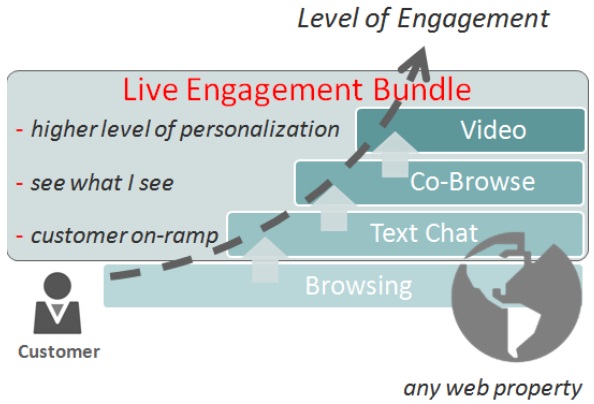
WebRTC Enabling Enterprises to Create Value for Customers

- Web Customer Service



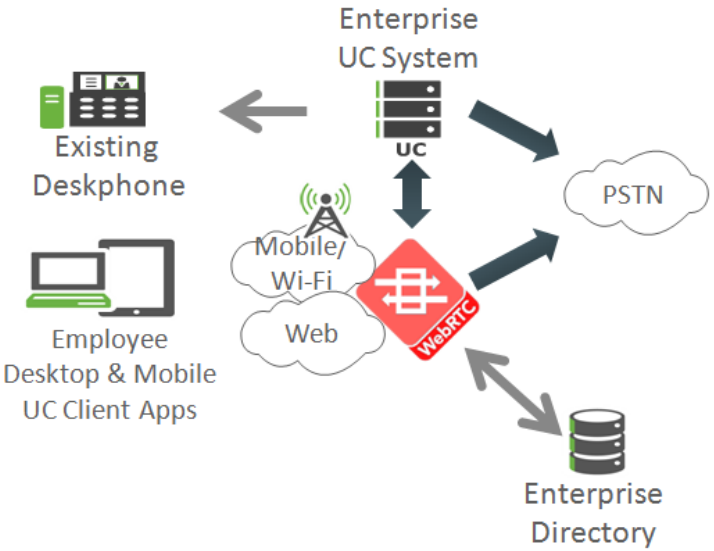
- Offer faster web-based service tools where most interactions with a business begin, on the Web

- Advanced Customer Experience



- Offer a 'Mayday Bundle' for advanced live engagement, but make it available for any website across any devices

- Unified Communications



- Provide BYOD mobile UC clients to employees as an extension of existing desk phone experience for mobility & remote worker

WebRTC Services

Vonage
Crazy generous.

Video Calling Video Messaging

AMERICAN EXPRESS

Now with voice calls
WhatsApp

amazon.com

Esurance Video Appraisal

What Does WebRTC Really Provide?

Standards Specifications

NAT Traversal

Secure Media

Data Transport

Software Stack

Free, Open Source

Media APIs & SDK

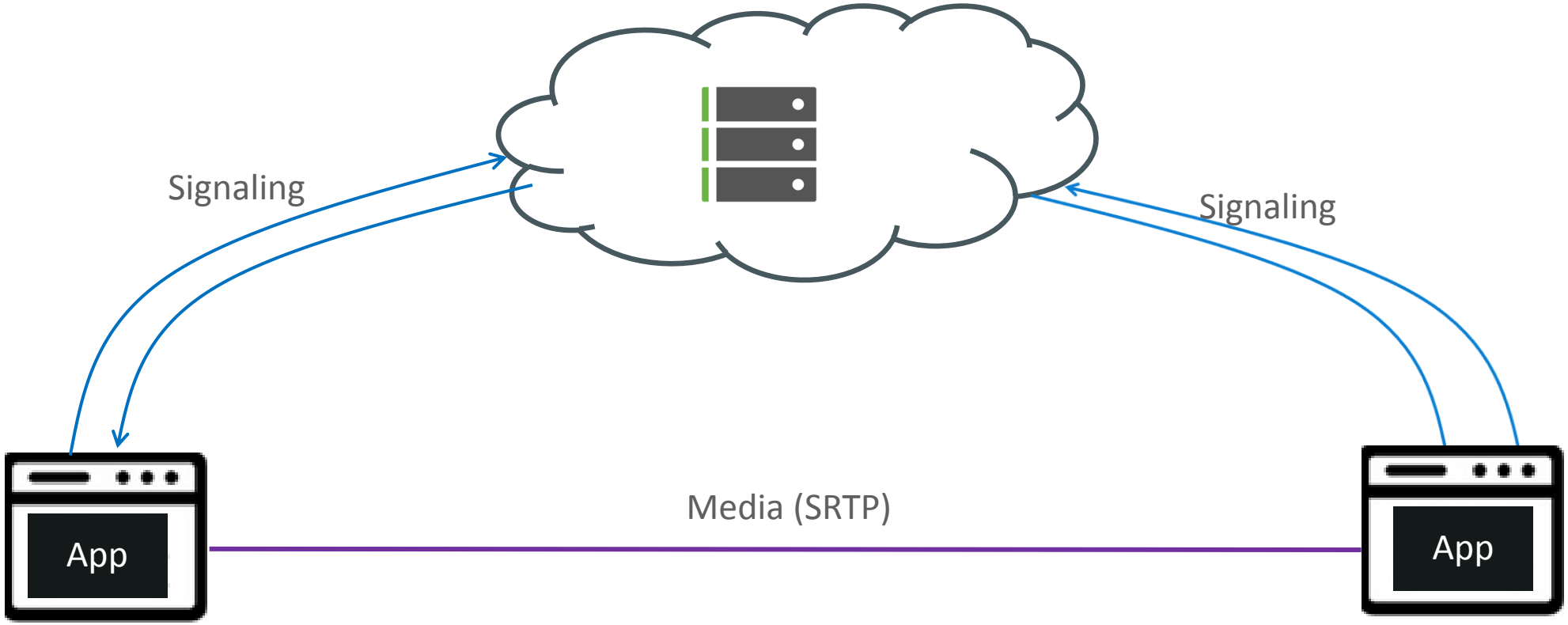
Media Codecs

No Plug-ins

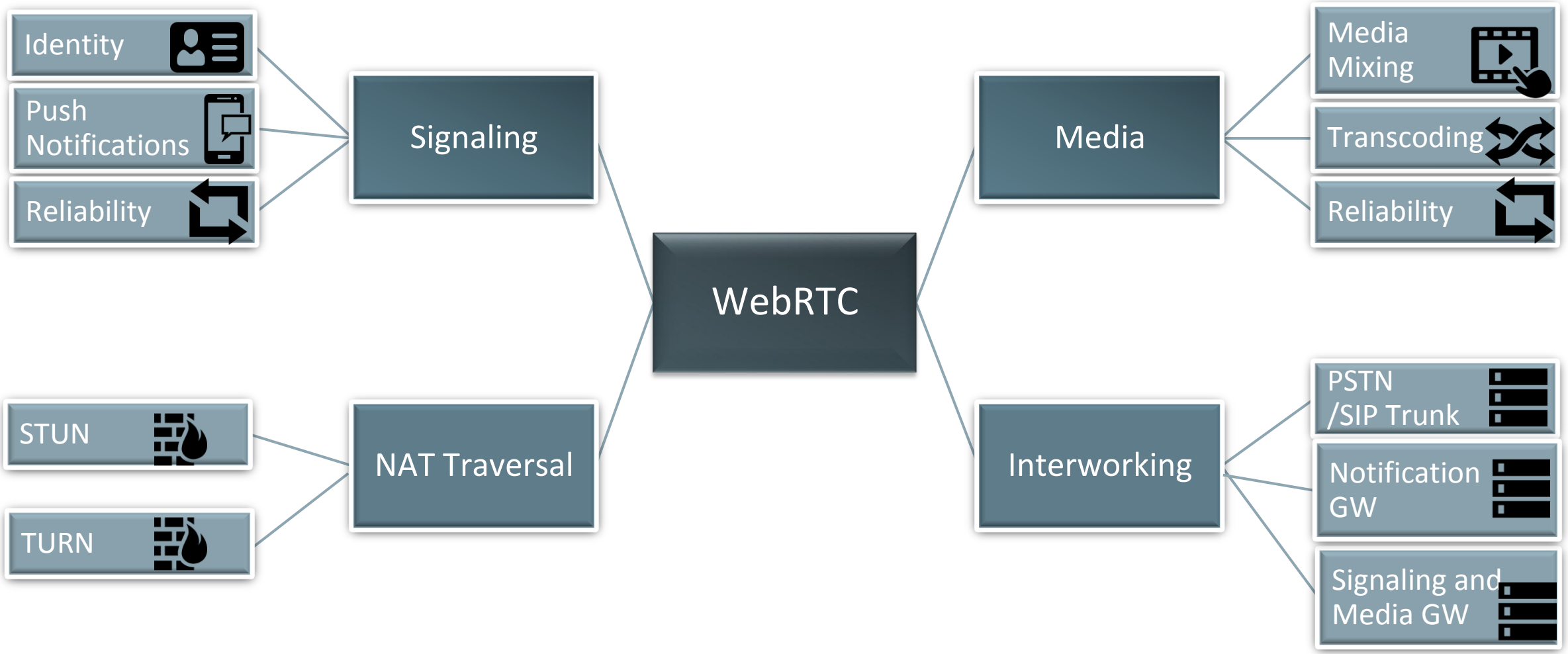
Agenda

1. WebRTC Introduction
2. WebRTC Architecture Components
3. Java in WebRTC Eco-System
4. Common Development/Architecture Issues
5. Demo

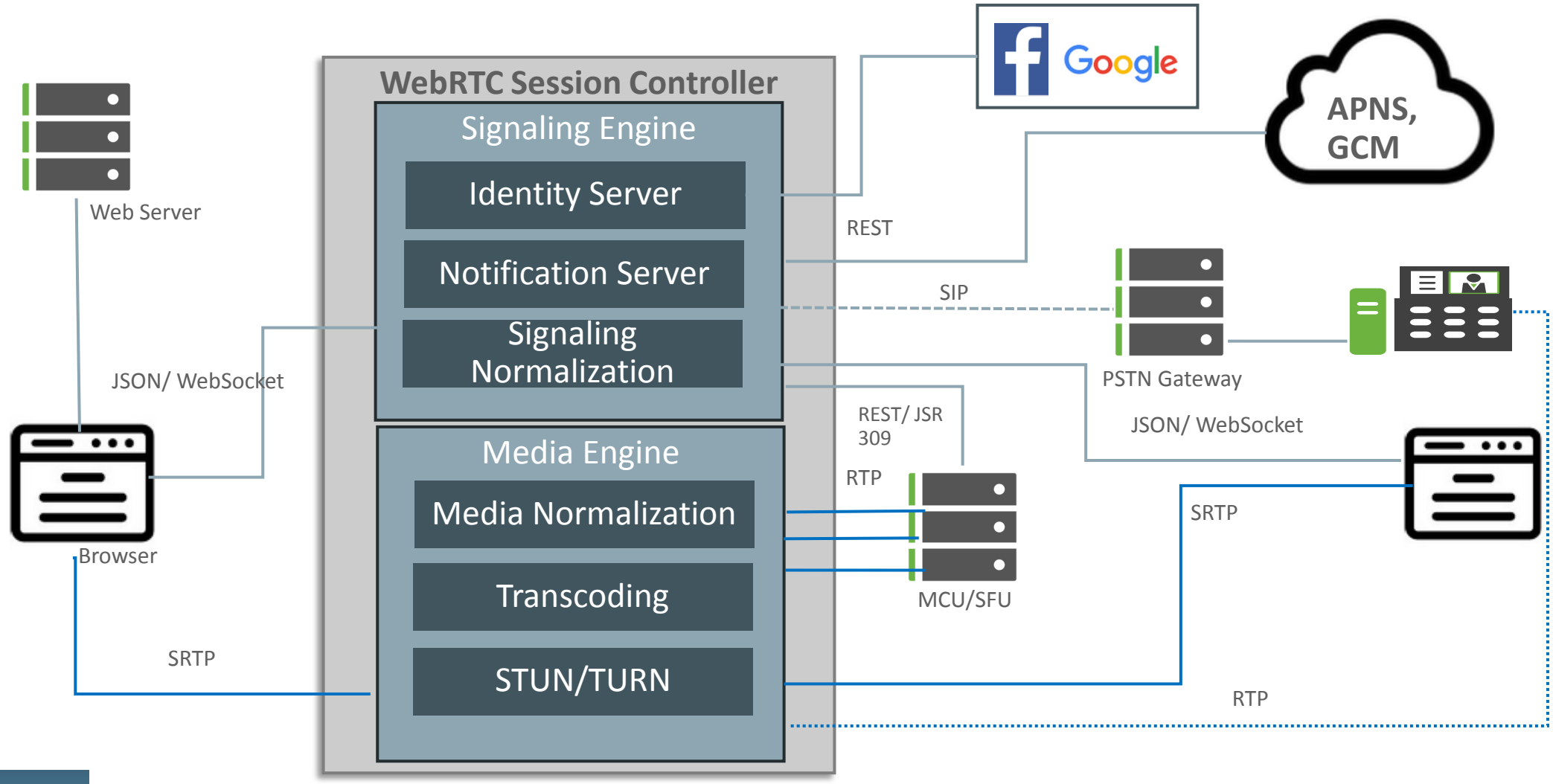
WebRTC Simple View



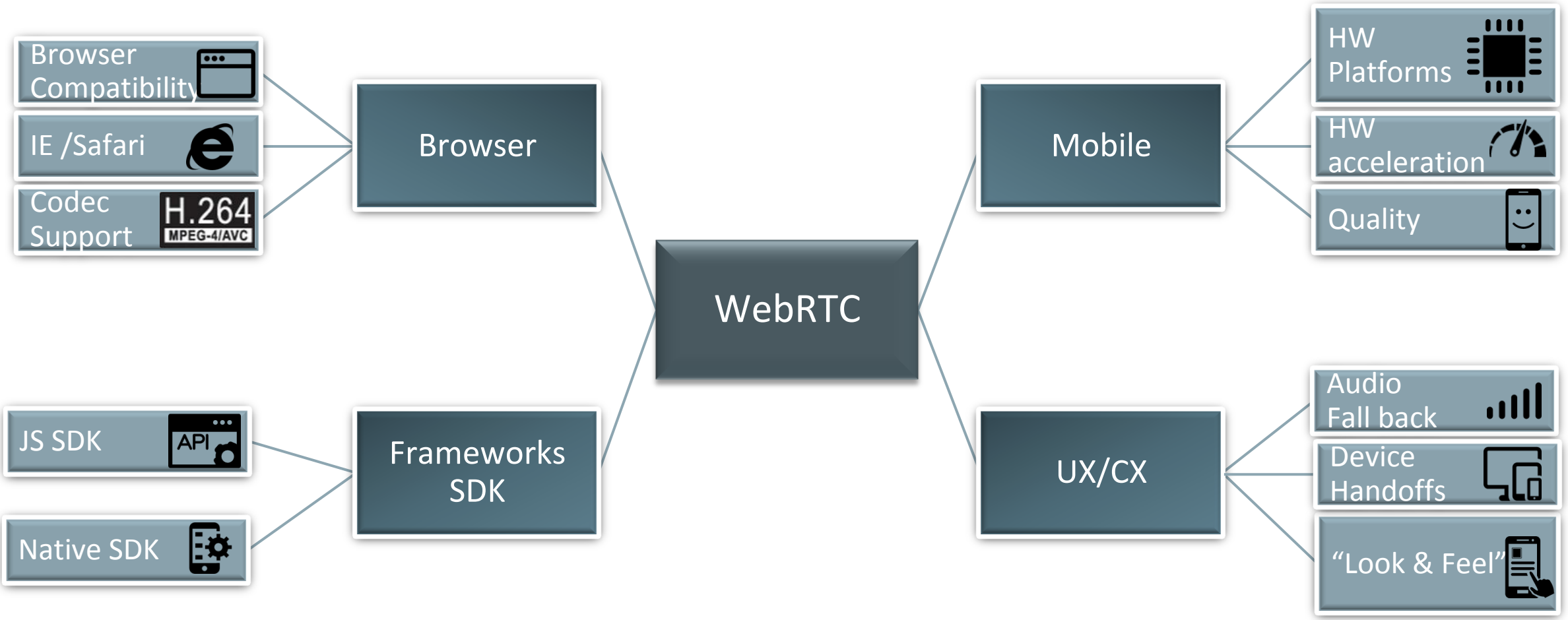
WebRTC Server-Side Considerations



WebRTC “Real-World” Architecture



WSC Client-Side Considerations



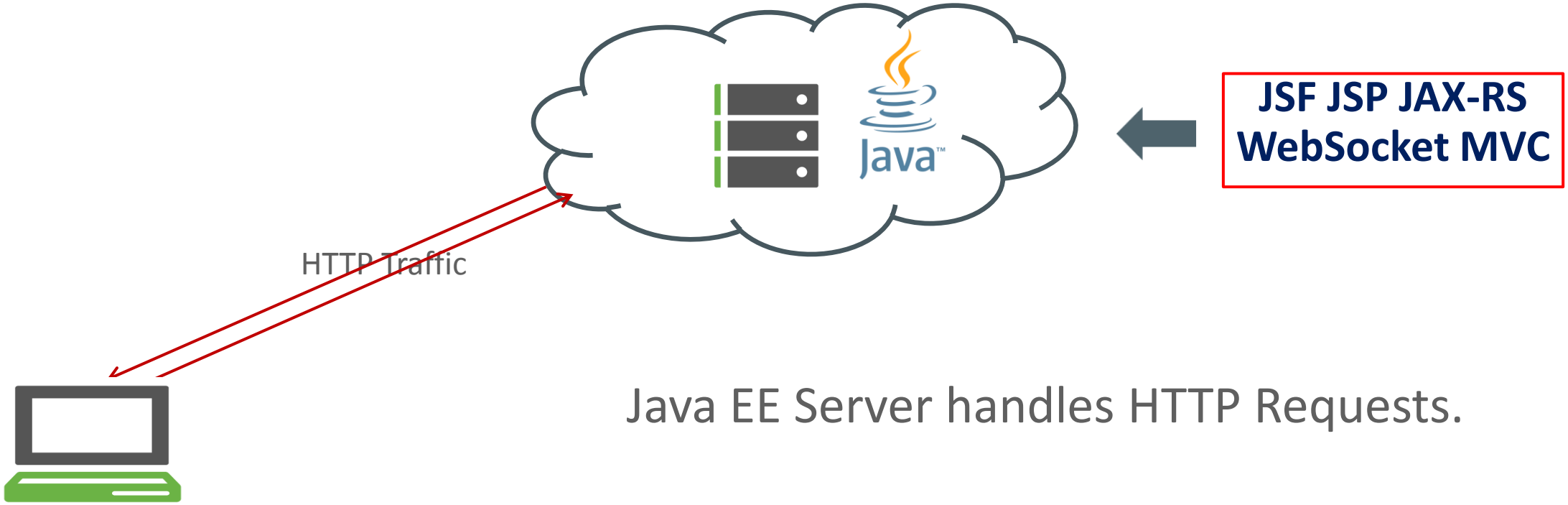
WebRTC Client Side SDK

- Broader client side reach – Native, Browsers, versions, hardware architectures
- Non-WebRTC Browser support
- Consistent feature evolution across native and browser SDKs
- Application specific signaling extensions and optimizations

Program Agenda

1. WebRTC Introduction
2. WebRTC Architecture Components
3. **Java in WebRTC Eco-System**
4. Common Development/Architecture Issues
5. Demo

Typical Web Application



Java EE Server handles HTTP Requests.

WebRTC is a W3C JavaScript API in the Browser

JavaScript API

Protocols

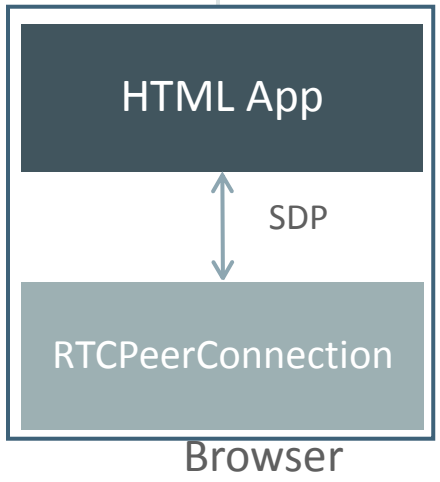
- Acquire Audio and Video
- Communicate Audio and Video Streams
- Communicate Arbitrary Data (Text, File

RTCMediaStream

- Represents stream of audio or video
- Stream from camera or microphone
- Navigator.getUserMedia

RTCPeerConnection

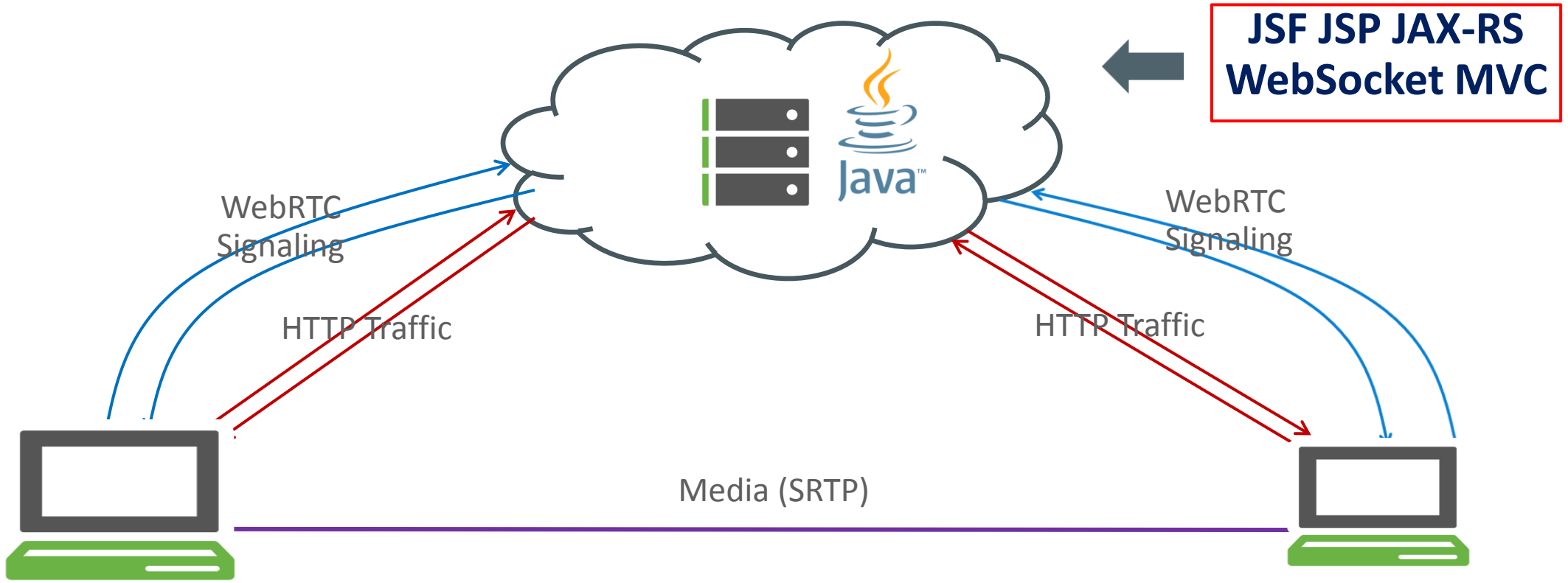
- Fundamental block of WebRTC API
- Represents Communication with a Peer
- Codec, Security, Bandwidth etc etc.



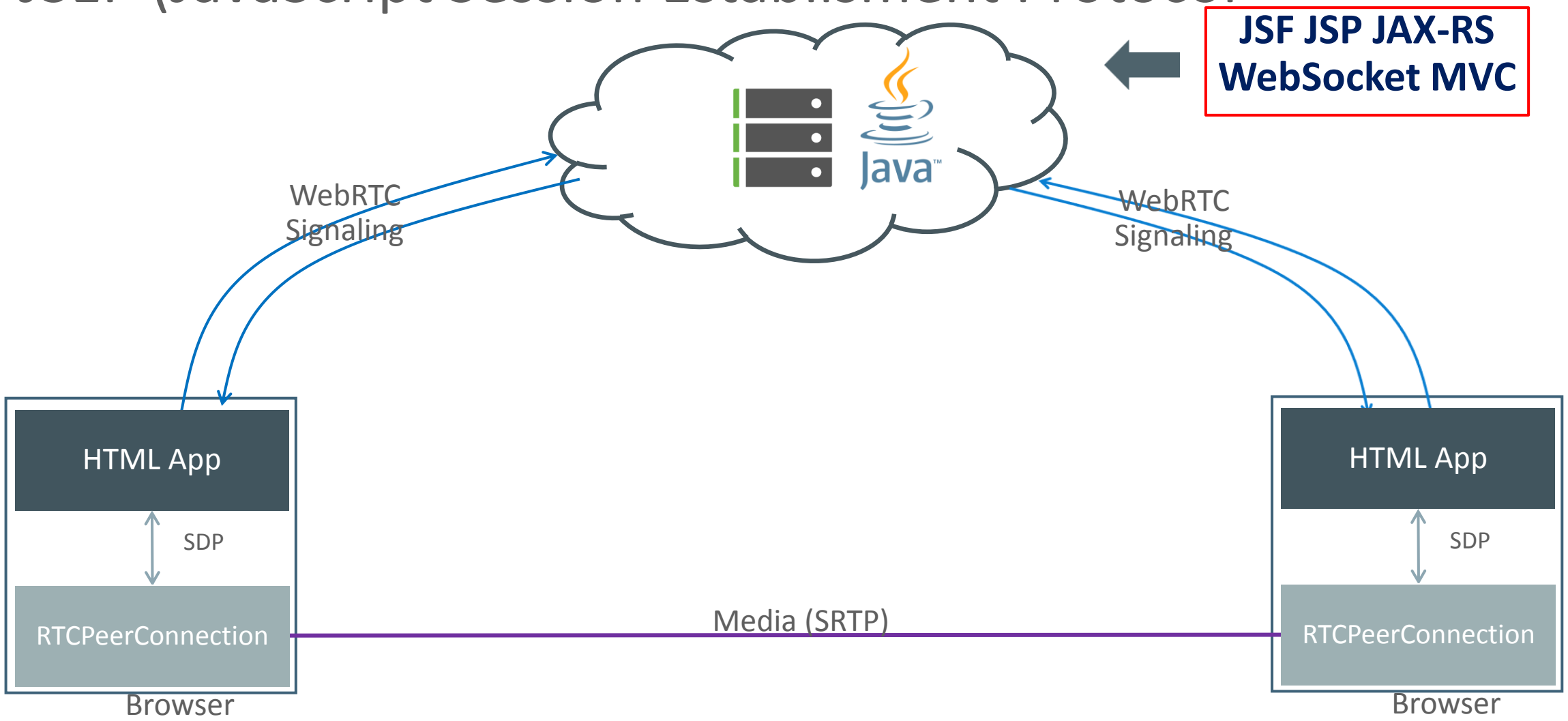
WebRTC API is in browser

How to get two PeerConnections to communicate?

Add WebRTC to your Java EE Web Application

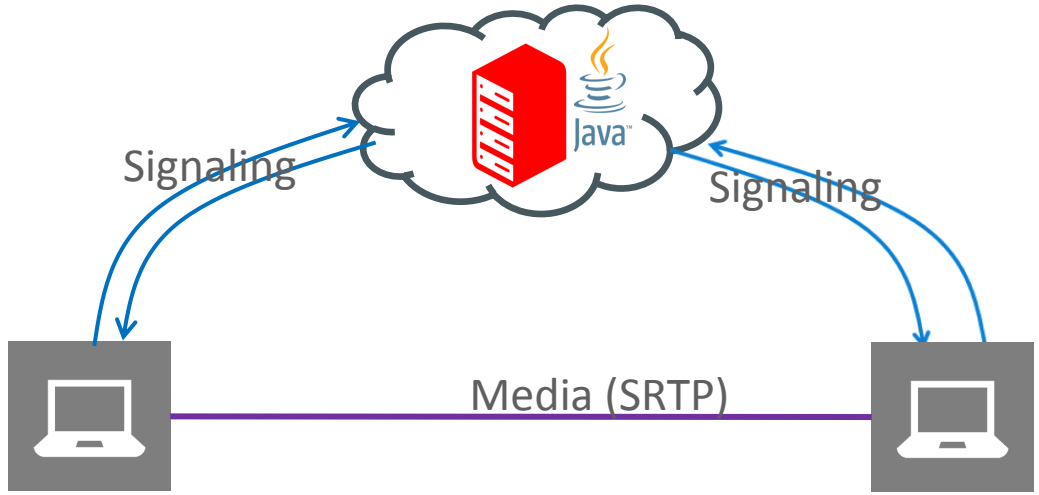


JSEP (JavaScript Session Establisment Protocol

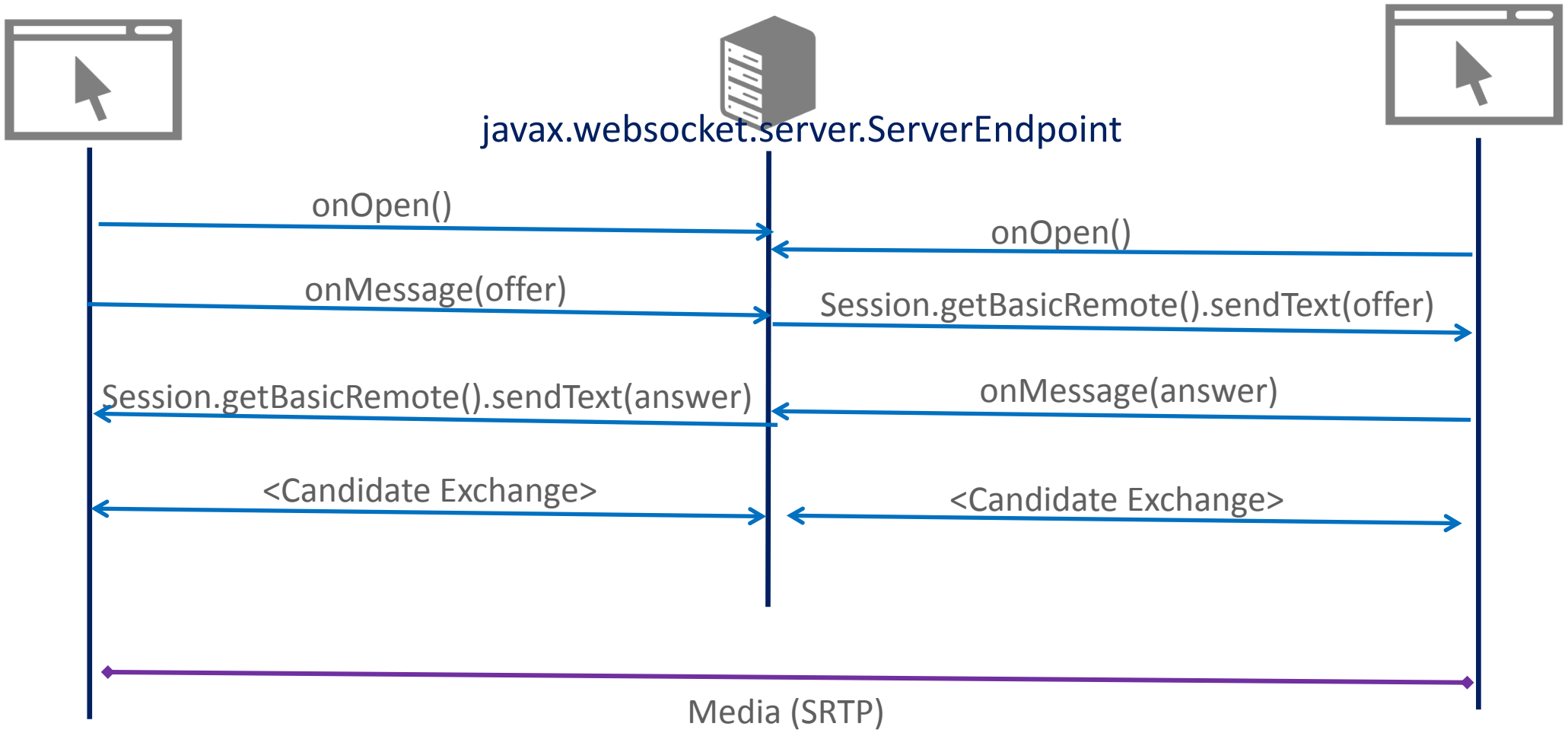


WebRTC Signaling Choices

- JAX-RS + SSE
  - Standard Paradigm
  - Wait for incoming (eg: disconnect) messages
  - How about incoming calls?
- Java API for WebSocket (JSR 356)
  - Bi directional
  - Suits communication signaling



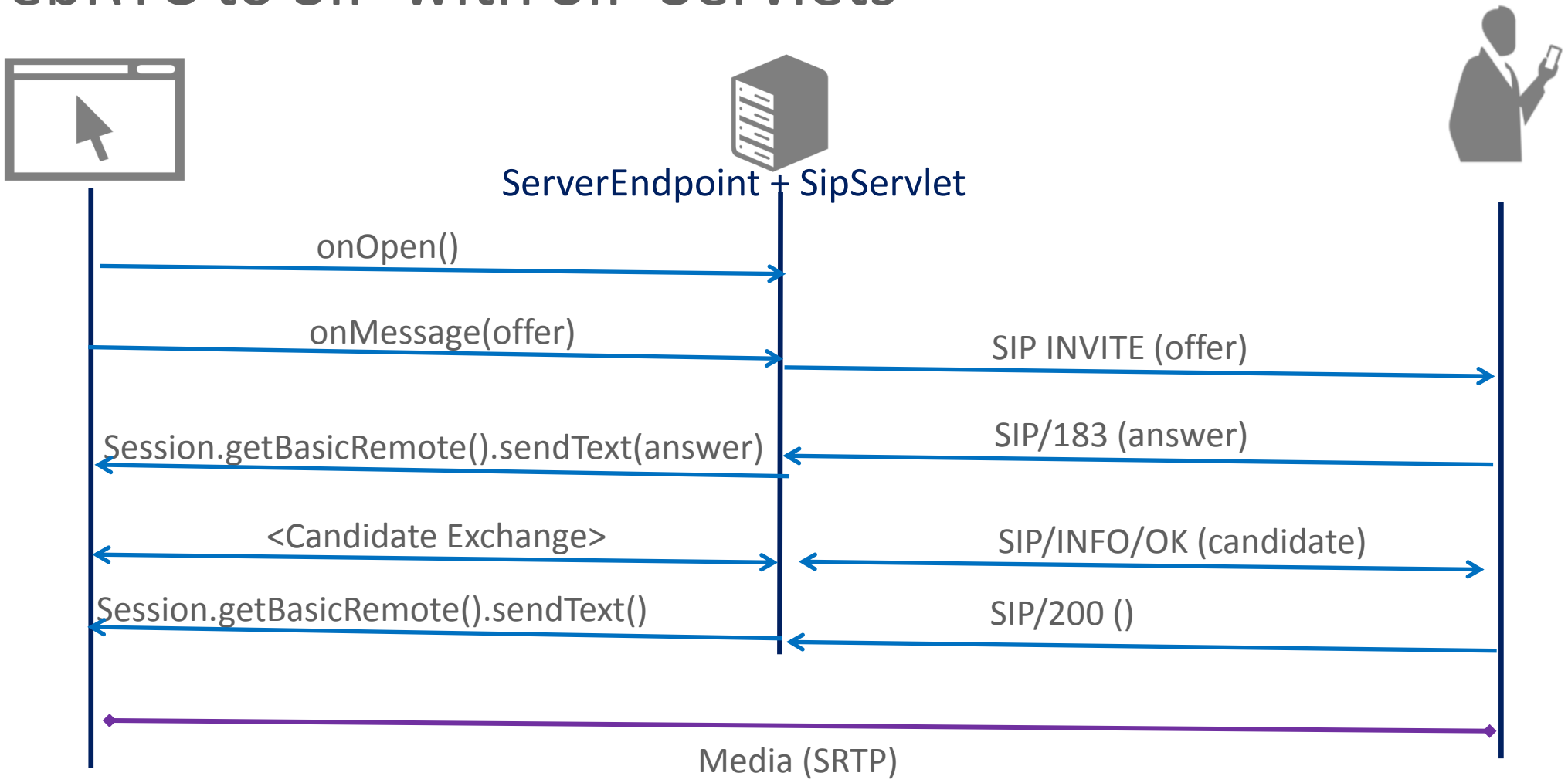
WebRTC in Web Application (Add a WebSocket Endpoint)



Extend Communication To Legacy Phone Networks

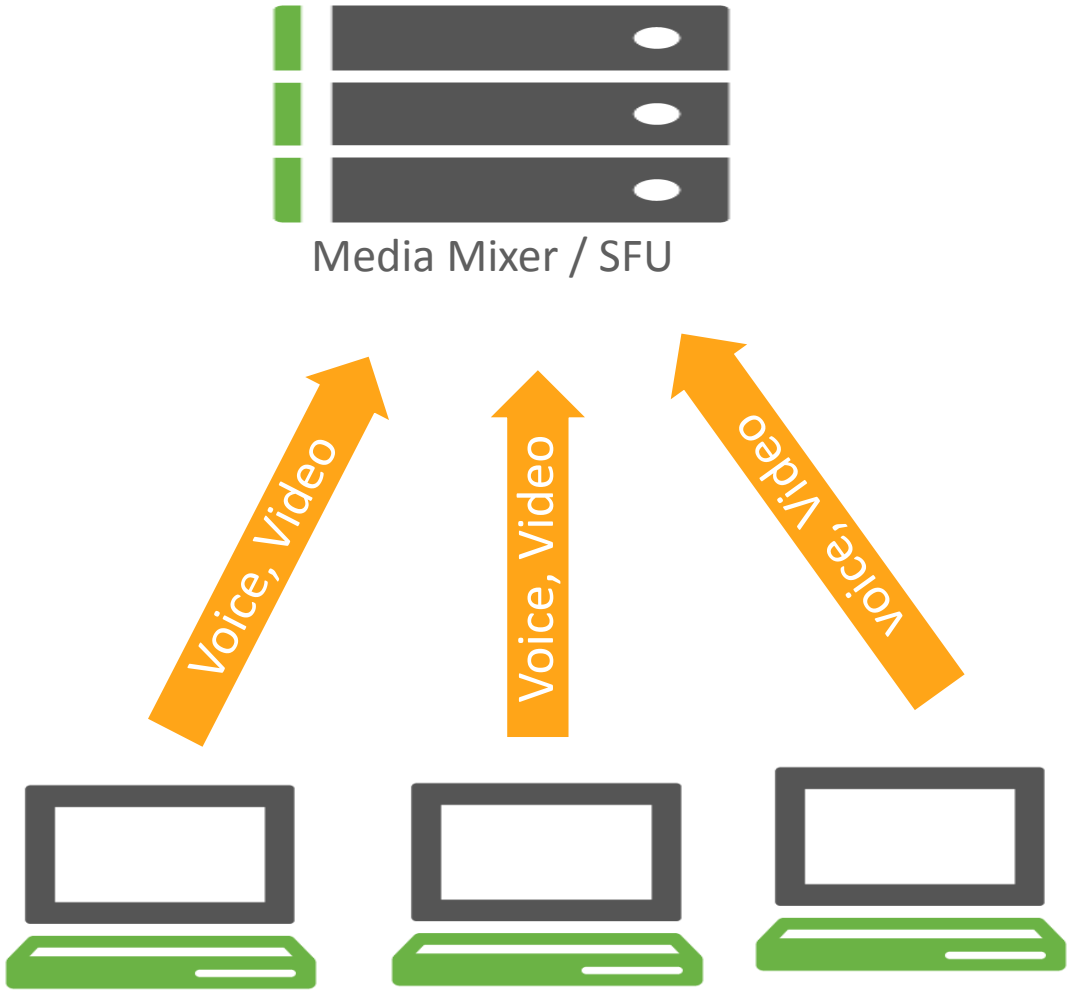
- SIP Servlet 2.0 API (JSR 359)
  - Integrates with Java EE 7
  - Allows you to connect with traditional SIP networks
  - Also supports SIP over WebSockets

WebRTC to SIP with SIP Servlets

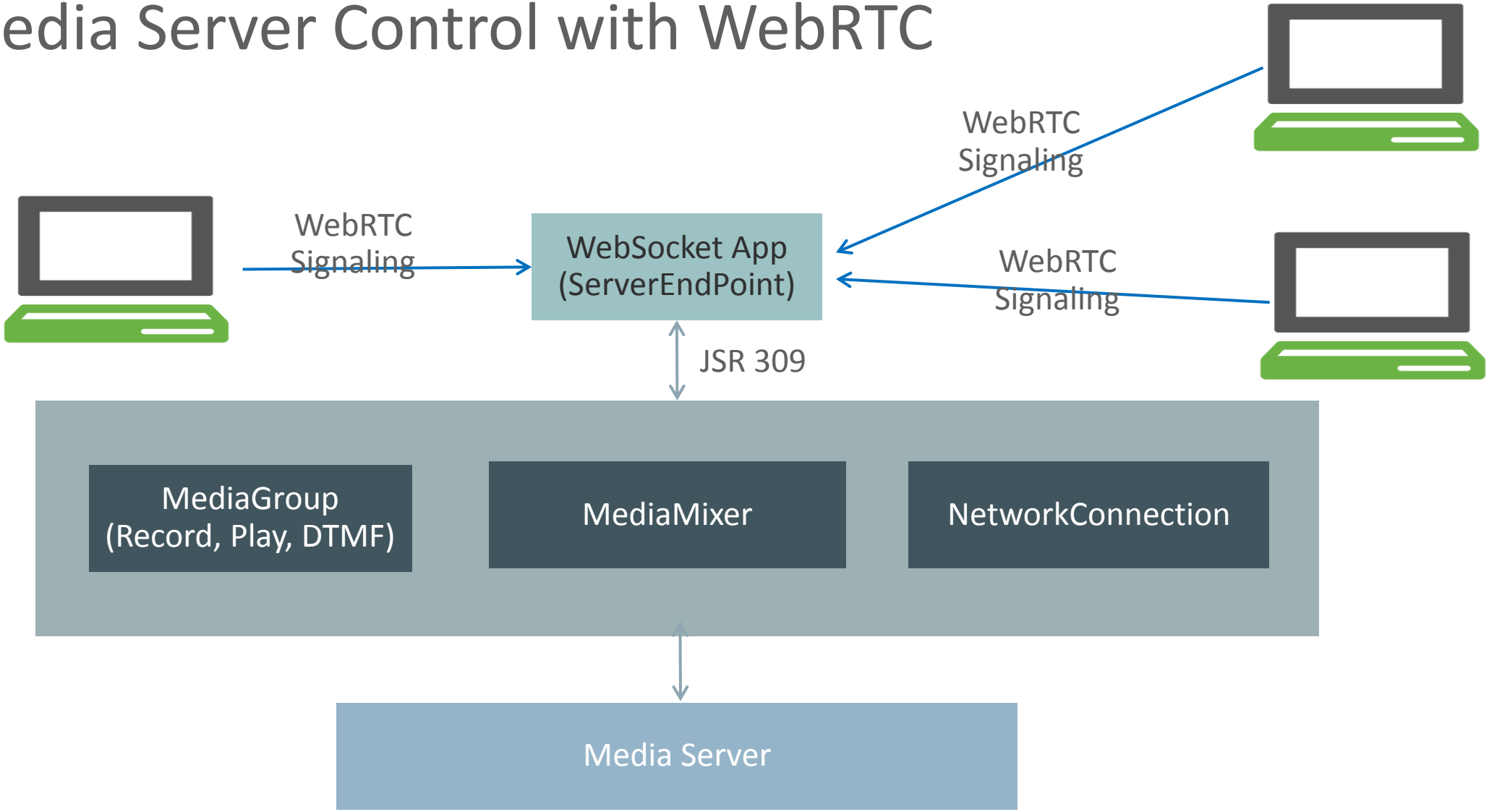


How about Conferencing?

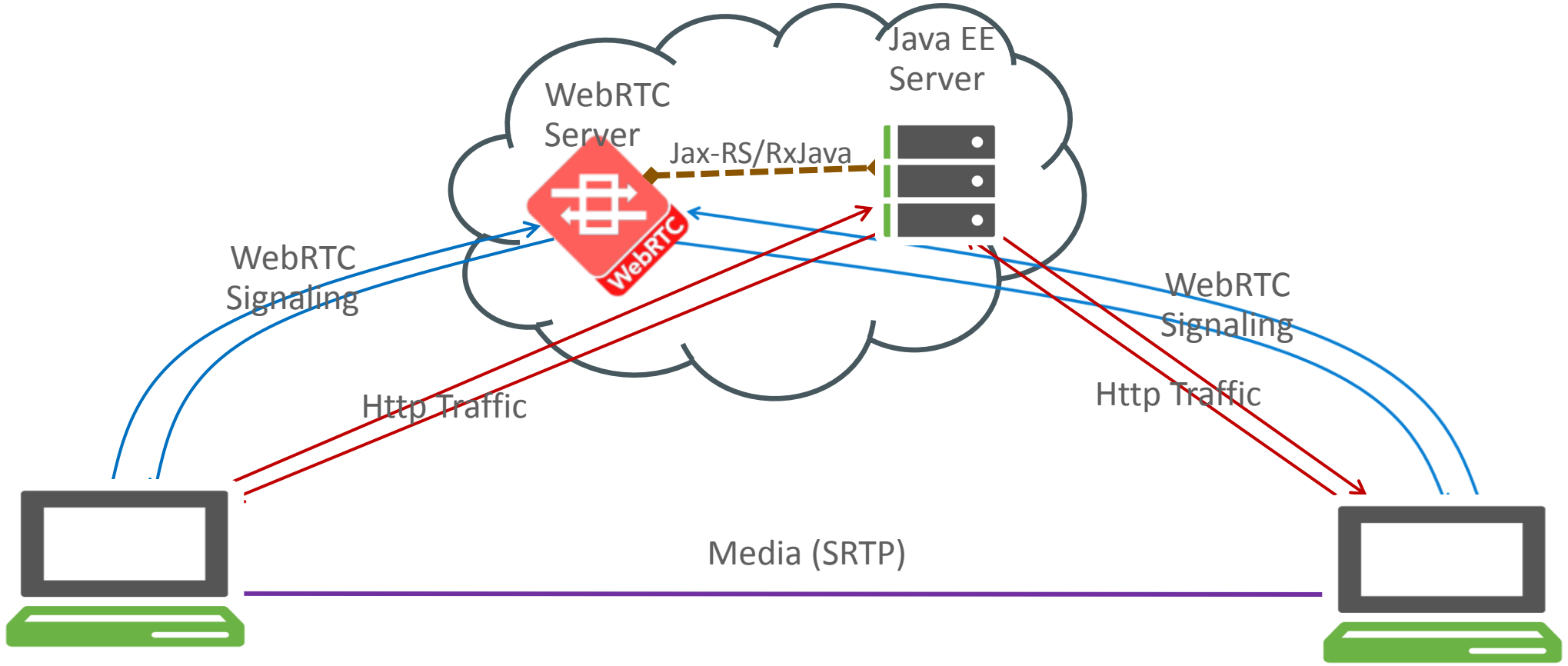
- Media Server Control API
  - Server Side Java API (JSR 309)
  - Abstract Interaction with Media Server
    - Dialogic, Radisys
  - Connect your `ServerEndpoint` with a Media Server



Media Server Control with WebRTC



How about a MicroService for WebRTC?

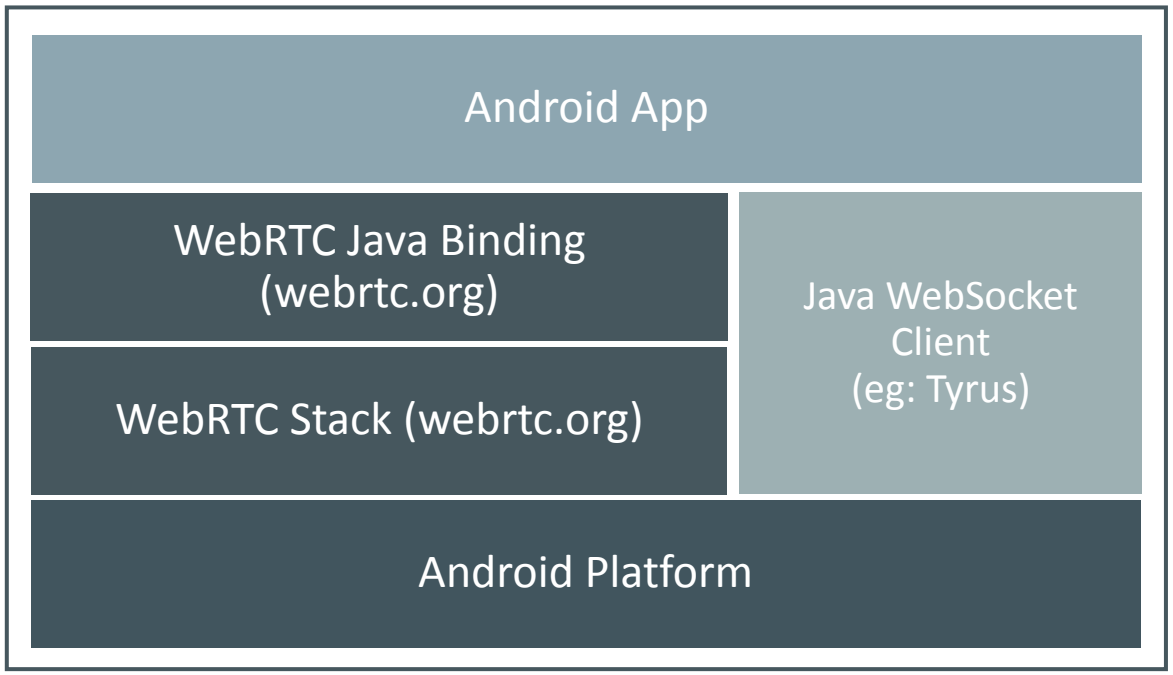


WebRTC in Android

- WebRTC Java Binding
  - RTCPeerConnectionFactory
  - https://chromium.googlesource.com/external/webrtc/
  - Google's open source project
  - Matches with W3C JavaScript API

android

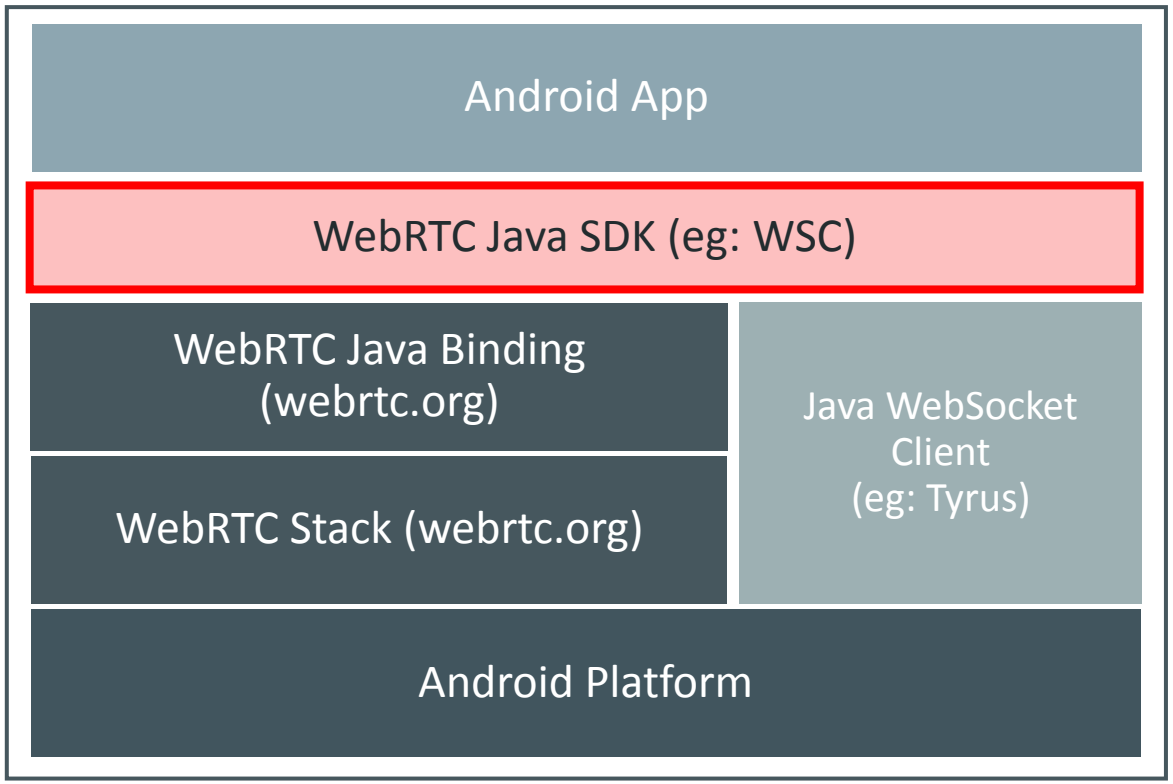
WebRTC in Android - Architecture



Initiate a WebRTC call in Android

```
PeerConnectionFactory pcf = …;
PeerConnection pc = pcf.createPeerConnection(iceSrvrs, constraints, obsrvr);
pc.addStream(localStream); //add the stream from local camera/mic
pc.createOffer(this, offerConstraints);
public void onCreateSuccess(final SessionDescription offer) {
    sendMessage(offer) ; // send on signaling channel
}
```

Java WebRTC SDKs for Android

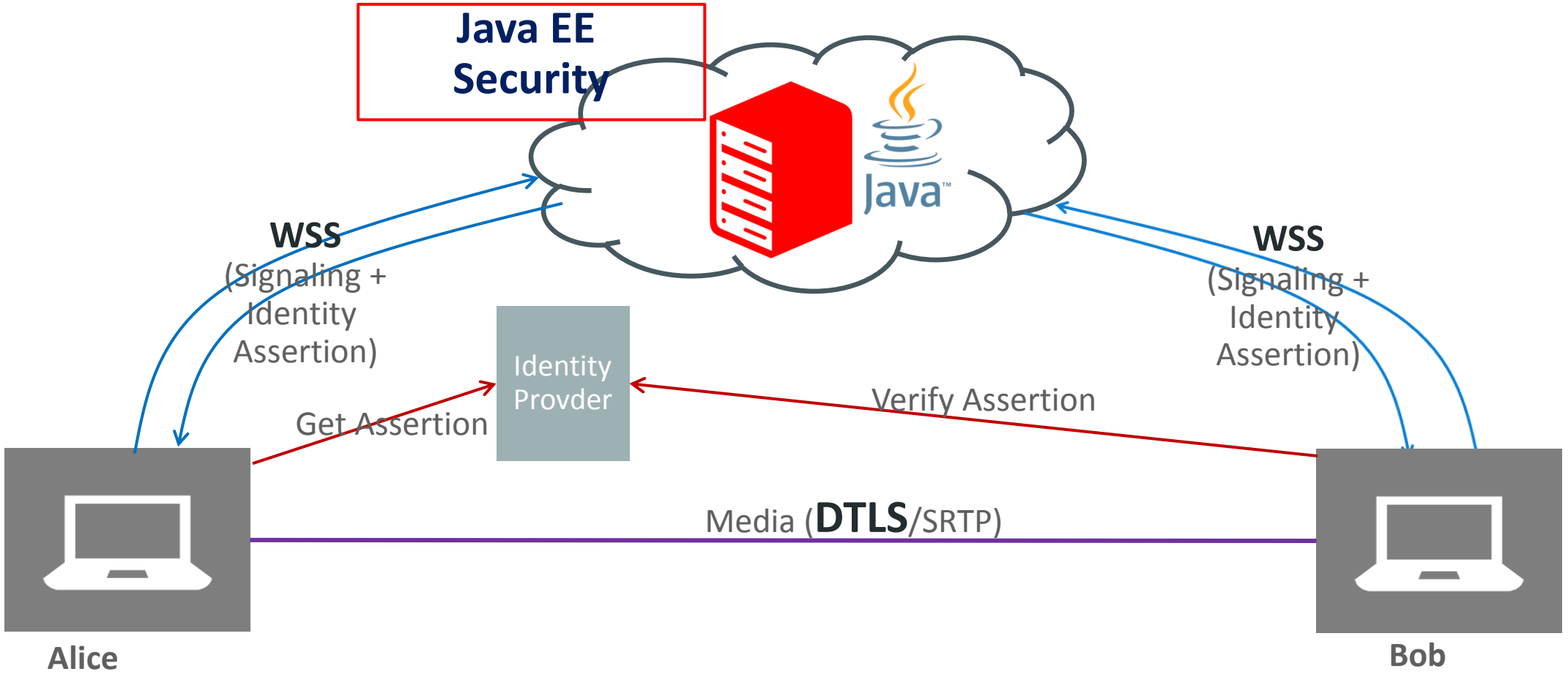


Similar stack exists for IOS as well

Program Agenda

1. WebRTC Introduction
2. WebRTC Architecture Components
3. Java in WebRTC Eco-System
4. Common Development/Architecture Issues
5. Demo

Security, Authentication and Authorization



Peer Authentication

- Alice calls Bob
- Alice's browser fetches assertion from IDP (JAX-RS)
- Alice's browser app sends the assertion to Bob (Java WebSocket)
- Bob's browser verifies the assertion (JAX-RS)
- Bob's browser displays authorization
- Bob accepts the call

Session Connectivity and Reliability

- Customers expect a seamless experience across
  - Web-style Browser reloads
  - Web-style “Back Button” navigation
  - Native app crashes
  - IP network connectivity changes (WiFi <-> 4G)
  - Device Handoffs
  - Server-side failures
- This can be solved using the concept of Session Rehydration
  - Ability to keep the session alive when connectivity is interrupted and recreate it as soon as the connectivity is re-established

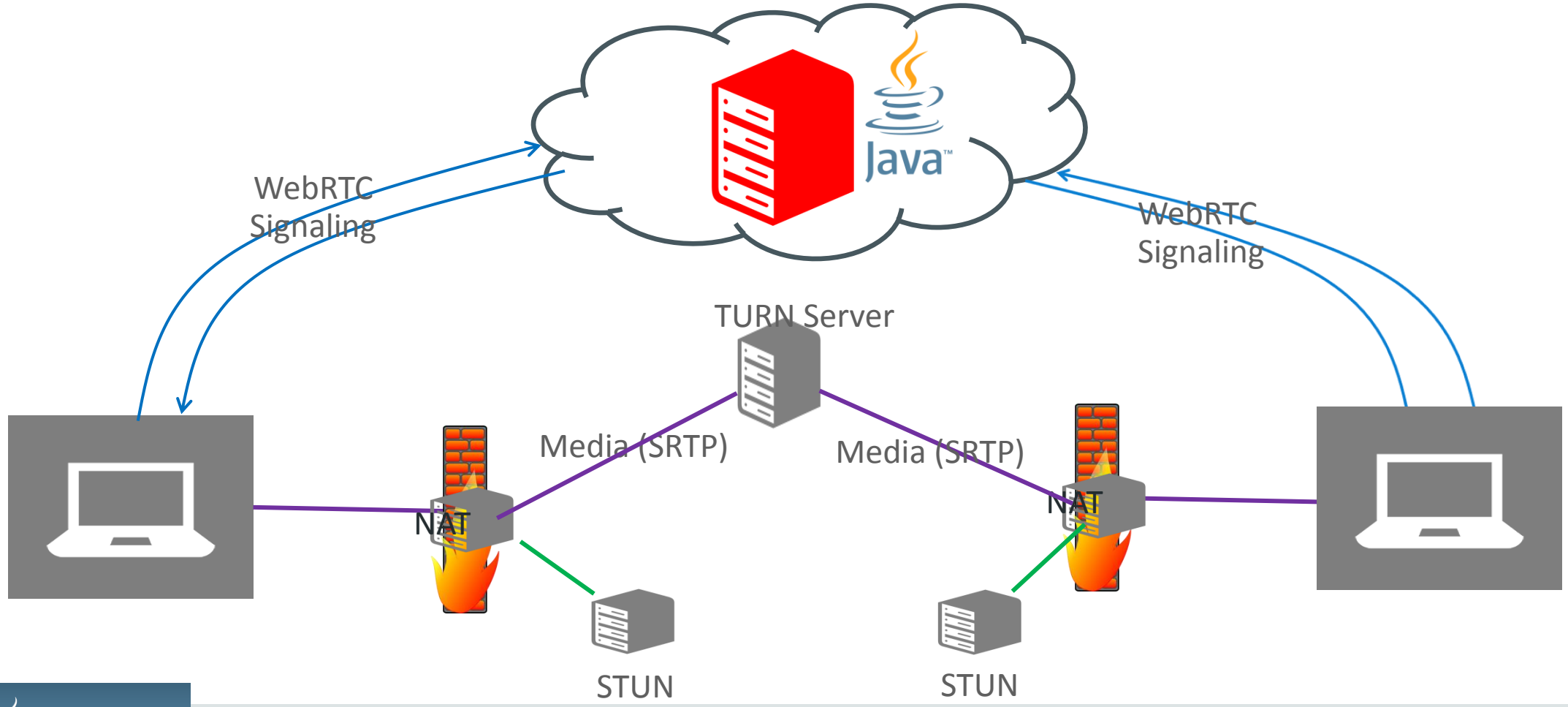
Session Rehydration

- In the event that the local app state is reinitialized, either due to a user reload of the page, or a decision within the app to reload itself it is possible to keep an existing session alive, via a process called "rehydration“
- Inspired by the approach described in IETF rtcweb-jsep-03 draft
- Upon reconnect, resurrect the session (voice, video, Data Channel)
  - Client Information (sessionId etc) is stored in LocalStorage
  - Completely reliable signaling protocol
  - WebSocket connection is kept for a short time and the message resynchronization happens when clients is reconnected
  - Restart ICE procedures, send new SDP

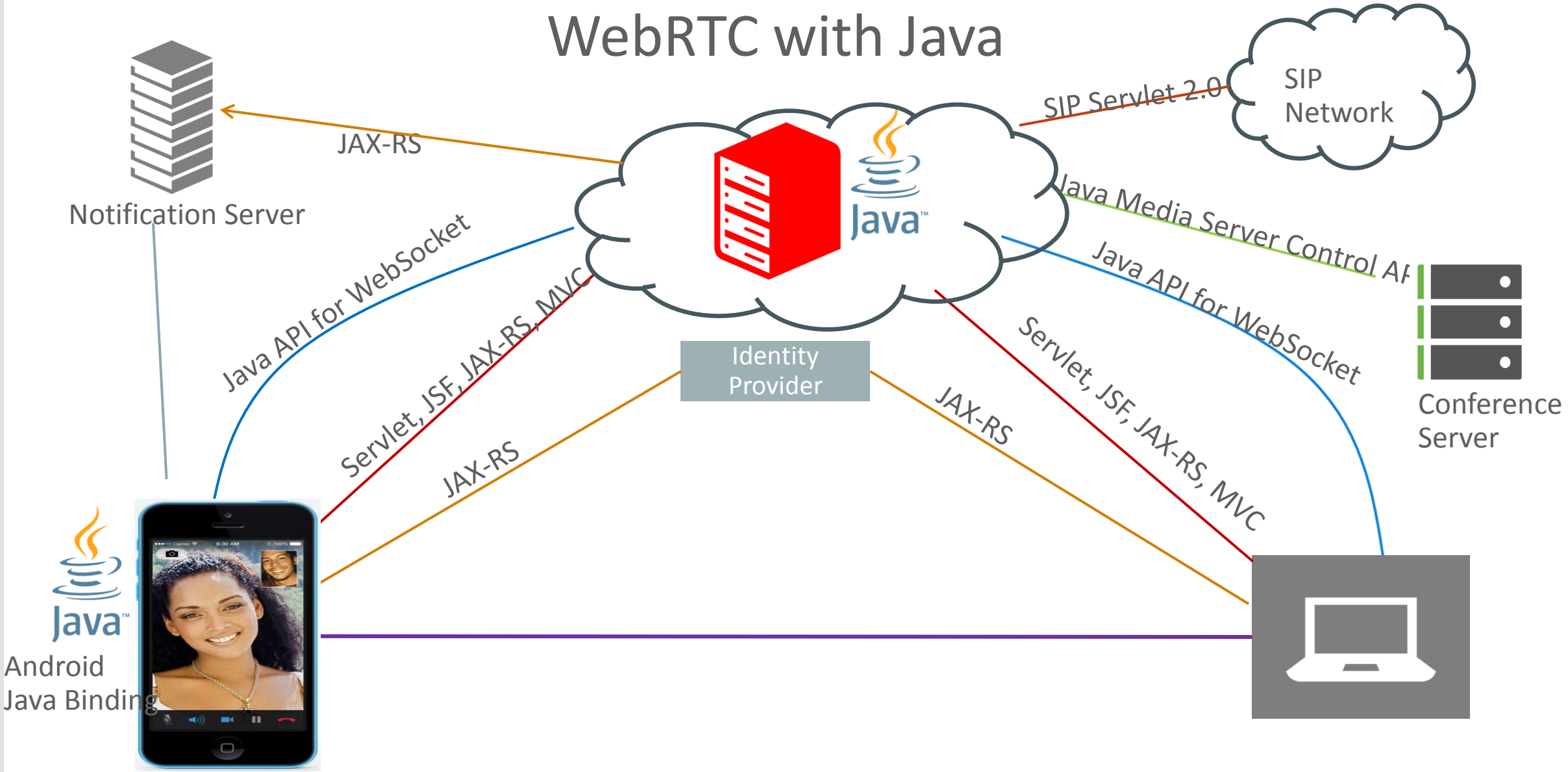
The World is a Chatty Place!

- Customers expect to stay “engaged” when they wander away from the app without draining device resources – battery power
- This can be solved by:
  - Optimizing the WebSocket connections with Push Notifications
    - Hibernation of the session during periods of inactivity
    - Session rehydration upon wake up/ call resume
  - Mobile Push Notification Gateway
    - Manages connectivity to APNS, GCM; registers and activates multiple apps
    - Supports templates
    - Delivers push notifications to iOS and Android
  - Chrome Push Notifications – Service Worker, W3C API
    - On desktop and mobile browsers
  - WebRTC server uses JAX-RS to send messages to notification server

Is network that simple? No!



WebRTC with Java



Additional Developer Resources

- Oracle WebRTC Developer Page : http://www.oracle.com/technetwork/developer-tools/webrtc/overview/index.html
- Oracle WebRTC Session Controller: http://www.oracle.com/us/products/applications/communications/web-rtc-session-controller/overview/index.html
- Oracle WebRTC Documentation: http://docs.oracle.com/en/industries/communications/webrtc-session-controller/index.html
- Sandbox (partner maintained): http://tadhack.optaresolutions.com/

Demo

Session Surveys

Help us help you!!

- Oracle would like to invite you to take a moment to give us your session feedback. Your feedback will help us to improve your conference.
- Please be sure to add your feedback for your attended sessions by using the Mobile Survey or in Schedule Builder.

JavaOne™

ORACLE®