

# WebRTC 서비스 구축 가이드와 PlayRTC

최진호 ([jinho.c@sk.com](mailto:jinho.c@sk.com)) SK Telecom

# WebRTC서비스 잘 만들기

---

- 높은 수준의 음성/영상 품질
- 모바일에서 에러 없거나 명확하게 알려주거나
- 다양한 예외 이벤트에 잘 대응
- 해외에서도 잘 지원함
- 다양한 플랫폼을 지원

# WebRTC서비스 잘 만들기

---

- 높은 수준의 음성/영상 품질
  - 미디어엔진 품질 확보
- 모바일에서 에러 없거나 명확하게 알려주거나
  - 테스트와 모니터링
- 다양한 예외 이벤트에 잘 대응
  - 테스트와 자동화
- 해외에서도 잘 지원함
  - Relay 서버, DPI
- 다양한 플랫폼을 지원
  - ORTC 혹은 PlayRTC

# 미디어엔진 품질 확보

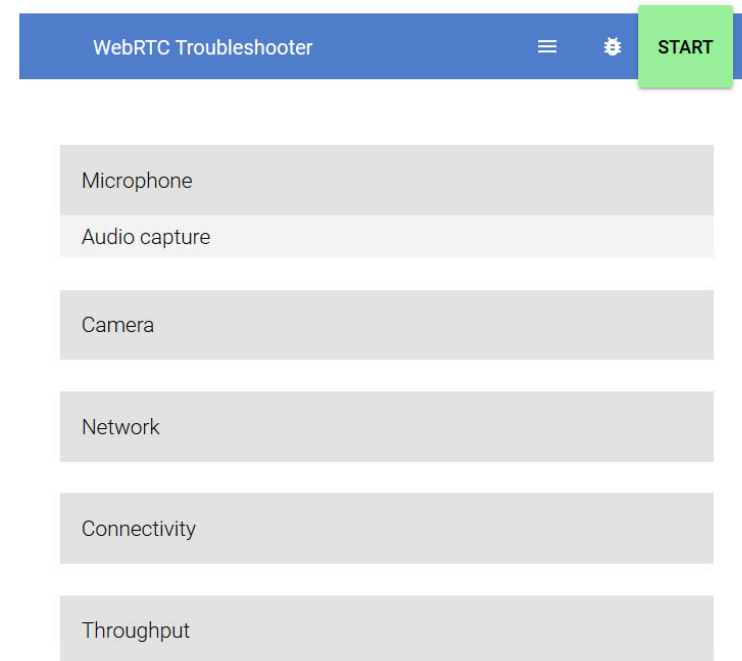
---

- WebRTC core의 모바일에서의 한계 (Voice)
  - Headset 혹은 통화모드의 경우: Okay
  - But, Speaker 모드의 경우
    - AECM의 half duplex
    - AGC와 AECM의 디바이스마다 다른 처리 route와 품질
    - Volume control의 튜닝이 저마다 필요함
  - 모바일 환경에 최적화 되어있지 않음
    - 불안정한 네트워크 대응
    - 다양한 클라이언트 환경에 따른 동적인 대응
- WebRTC Native Code에 대한 튜닝 필요
  - 스피커모드 혹은 검증되지 않은 장비의 경우

# Test와 자동화

---

- 클라이언트 환경 테스트
  - <http://github.com/webrtc/testrtc>
- 통신 오류의 대부분은 사용자 환경에서 발생



# Test와 자동화

---

- 통신 중 네트워크 Health Check
  - WebRTC Stats API 활용
  - Inboundrtp: 수신중 RTP 스트림 정보
    - fractionLost: 주기별 RTP 패킷 유실율
    - Jitter: 시점차이 변동값
    - packetLost:: 총 유실 RTP 패킷수
  - Outboundrtp: 송신 중 RTP 스트림 정보
    - roundTripTime: SRTCP round trip 시간

# Test와 자동화

---

- 자동화 테스트
  - Selenium+ nightwatch 혹은 xvfb
  - 크롬: --use-fake-device-for-media-stream --use-fake-ui-for-media-stream



# Relay 서버

---

- TURN사용 비율
  - 30~50% (서비스 특성마다 다름)
- Global 서비스를 준비한다면
  - TURN/STUN의 적절한 클라우드 활용 필수
- 국가별 특수성 고려 필요
  - 중국
  - 일부 ISP
- Rfc5766-turn-server보다는 coTURN을 추천
  - Scalable, 보안, multi-tenancy, 모니터링 측면
- Google public STUN!



# ORTC란?

---

- WebRTC 차기 버전(1.1?)
- 특징점
  - Mobile First: Continuous nomination, Mobile SDK
  - 다양한 조건의 스트림 지원: 비대칭적 Audio/Video 채널 사용. Media forking 및 한번의 offer/answer에 의한 채널 재사용. SimulCast/SVC
  - SDP 불필요 및 다양한 제어 가능한 API 제공 (codec/media flow/forking)
- MS Edge Browser 적용 예정
- Google Chrome 역시 2016년 상반기 지원 예정

# WebRTC API

**getUserMedia**

**PeerConnection**

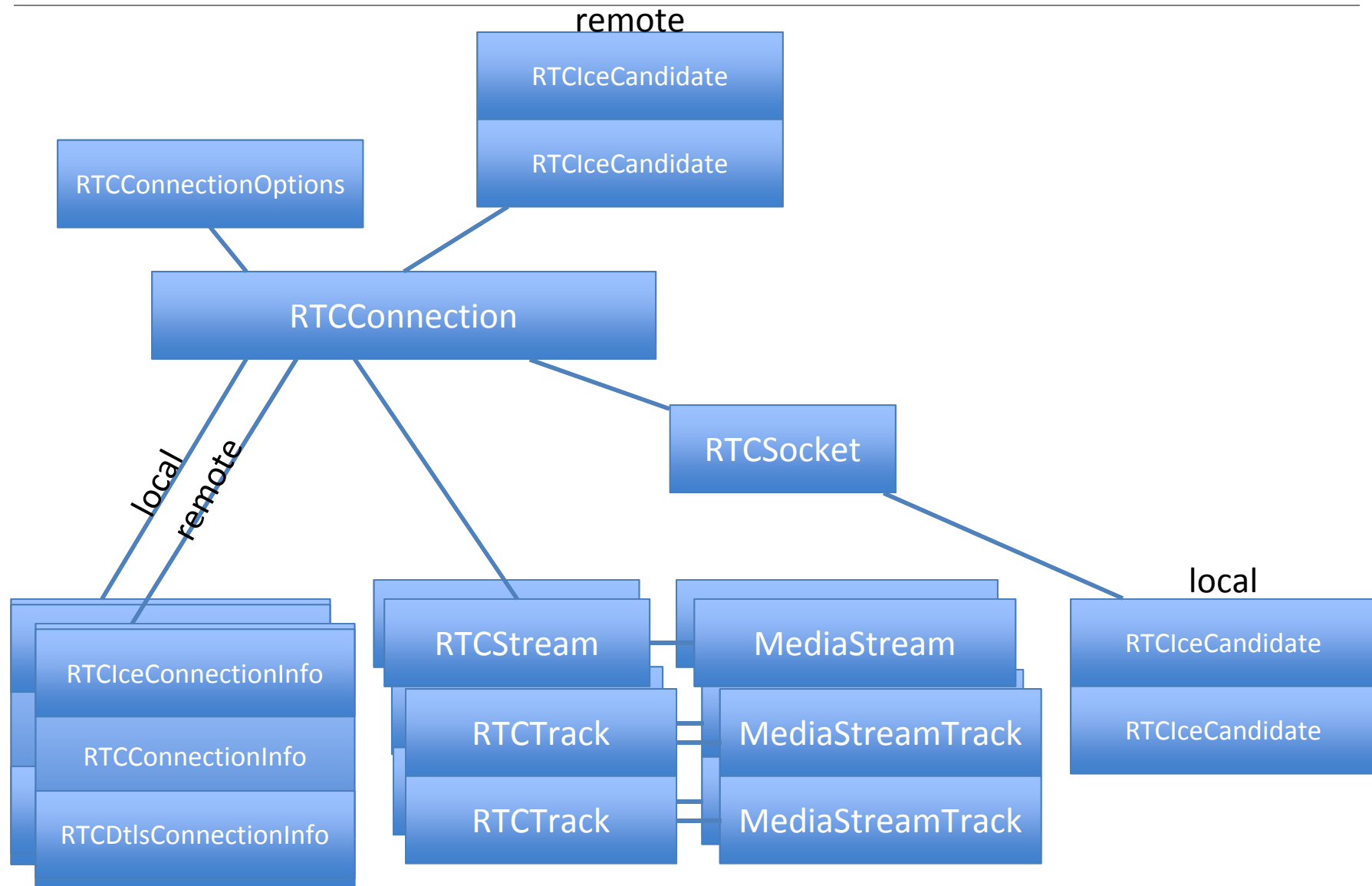
**DataChannel**

+

```
v=0
o=- 0 0 IN IP4 10.19.220.21
s=session
c=IN IP4 10.19.220.21
b=CT:99980
t=0 0
m=audio 16592 RTP/AVP 114 9 112 111 0 8 116 115 4 97 13 118 101
a=candidate:Mwxm5UD925g8PxUzNCIwTgnyNJ129B9VjMocFJzux4o 1 epVykxik2Fk1MisJQavxQ UDP 0.830 10.19.220.21 16592
a=candidate:Mwxm5UD925g8PxUzNCIwTgnyNJ129B9VjMocFJzux4o 2 epVykxik2Fk1MisJQavxQ UDP 0.830 10.19.220.21 16593
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80 inline:awhUcDORdO5r0rHRhEYe3KqP3LFFh0uBcdHvty0[2^31]:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:FK8zbO8E6ayonB/DWw9tgwgy8kMptTLFYNOsqwQ0[2^31]:1
a=crypto:3 AES_CM_128_HMAC_SHA1_80 inline:uEL6Brk+7HAB9kuLC9ID+8d8zcmLoiSuqNcLCTQ[1]2^31
a=maxptime:200
a=rtpmap:114 x-msrta/16000
a=fmtp:114 bitrate=29000
a=rtpmap:9 G722/8000
a=rtpmap:112 G7221/16000
a=fmtp:112 bitrate=24000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:116 AAL2-G726-32/8000
a=rtpmap:115 x-msrta/8000
a=fmtp:115 bitrate=11800
a=rtpmap:4 G723/8000
a=rtpmap:97 RED/8000
a=rtpmap:13 CN/8000
a=rtpmap:118 CN/16000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=encryption:optional
```

**SDP**

# ORTC API



# ORTC-기타

---

- ORTC-Lib : <https://github.com/openpeer/ortc-lib>
  - ORTC mobile용 라이브러리 프로젝트
  - WebRTC Native code를 기반으로 ORTC 인터페이스 및 모바일 부분 API 제공
  - Android/iOS/WinRT용 지원
- 그러나 구글 WebRTC Native Code가 향후 ORTC API를 수용하는 계획 중

# WebRTC PaaS

---

- 다양한 플랫폼 지원 및 고품질, 인프라제공을 목적으로 하는 통신용 PaaS
- 개발 편의성과 아울러 서비스의 지역적 특징, 안정성 및 요구되는 품질등을 꼼꼼히 따져봐야 함

## 크게 4가지 PaaS로 구분

- 통신사 주관 PaaS: Tokbox, AT&T, PlayRTC
- 전통적인 Communication API 플랫폼에서 출발: Twilio, Plivo 등
- 순수 WebRTC PaaS: Temasys, Tokbox, EasyRTC, PlayRTC
- 영상회의나 UC 솔루션에서 출발한 플랫폼: bistri, appear.in, cafeX, viblast

# PlayRTC의 활용 사례

---

O2O

IoT

SNS

교육

## 향후 전망

---

- MS와 구글, Firefox의 놀랍도록 빠른 협력 진행 중
- 2016년 상반기 음성과 영상 모두 호환성 테스트 완료 예상
- ORTC기반의 보다 편하고 모바일에 최적화된 서비스 개발의 시대 예상
- SimulCast,SVC등 지원으로 보다 활발한 모바일 기반 Conference Call, Broadcasting 서비스 예상