WebRTC: Protocol And Programming

Protocol: Establishment, Codecs, SDP, Security, NAT
Programming: API Tour, PeerConnection, Project

Real time communication (RTC) is intended for person-to-person communication in real time. WebRTC is built into modern standards-compliant web browsers, mobile devices and IoT devices. WebRTC supports RTC without needing plug-ins. WebRTC enjoys very wide-spread vendor support. Forms of communication supported include audio, video and data exchange (e.g. application data or docs).

WebRTC is the basis for RTC within many interactive apps that are widely used. What is less known is that WebRTC is a programming platform, and apps running in the browser & on mobile/IoT devices can interact with it. App developers striving to add unique features to their apps would be well advised to consider WebRTC, as once an understanding of how it works has been gained, it is not that time consuming to program. The results are quite powerful and appreciated by users.

This course starts with an introduction to WebRTC, then looks at the protocol in detail, and then looks at how to program it. Topics covered include codec selection, SDP, addressing, security, identity, specialist services, data exchange and lots more.

	Contents of One-Day Training Course	
Target Audience Networking professionals and senior software engineers who require a deeper understanding WebRTC – both the protocol and how to program with it. **Prerequisites** Programming experience in JavaScript and/or TypeScript. Good all-round foundational networking knowledge; attendance at our *Fundamentals Of TCP/IP Networking* course or similar experience.	**WebRTC Intro** What is is trying to achieve Widespread adoption A community A set of standard protocols A set of standard APIs - no plugins needed An API for programming An open source project **Architecture** Major components for audio, video, data How transport works Identity JavaScript and C++ APIs Layering **Foundational Technologies** RTP – Realtime Transport Protocol SDP – Session Description Protocol Interaction with SIP (VoIP) Future: QUIC and WebRTC? **WebRTC Protocol** Connection Esablishment Message flows Codec selection Buffering networking **WebRTC and SPIN/ICE** Issues with some networks (NAT) How to use NAT with WebRTC Forwarding **WebRTC And Security** How to identify participants Securing comms links Regulation: comms, GDPR, police, ..	**W3C Standards** W3C is working on many WebRTC stds API written in WebIDL, so are familiar to any JavaScript/TypeScript app dev Review of important ones, e.g.: WebRTC 1.0: RTC Between Browsers and Media Capture and Streams **Connection Programming** How to programmatically set up connections with remote parties Configuration Security Transport RTCPeerConnection **Media API** Passing track info to remote parties RTCRtpSender RTCRtpReceiver **MediaStream API** MediaStreamTrack interface MediaStream interface **Data API** Exchanging app data over WebRTC links RTCDataChannel **Application Issues** Developer environment setup for WebRTC Debugging Error handling with WebRTC WebRTC as part of a large application suite **Project** Detailed project exploring how modern UI apps can be shared via WebRTC's data channel