BlogGeek.Me

WEBRTC TRAINING COURSES

May 2024

TSAHI LEVENT-LEVI
TSAHIL@BLOGGEEK.ME

TRAINING OVERVIEW

WebRTC is the most interesting VoIP technology in the last decade. It enables real time voice and video communications from browsers. Companies big and small are using WebRTC to fit into their business processes. The BlogGeek.me WebRTC Training courses are a set of courses meant to assist in the process of putting the best possible WebRTC architecture in place and successfully developing, launching and maintaining WebRTC applications.

The goal of these WebRTC courses is to bring students up to speed with WebRTC and enable them to make better product decisions, in effect, making them **professional WebRTC developers**.

WHY THESE WEBRTC COURSES?

There's a lot of information about WebRTC on the internet and a few courses and books as well. The BlogGeek.me WebRTC Training courses are **the only online resources on WebRTC** that are:

1. Up to date with the standard specification and the industry around it
2. Offer a simple, structured view of WebRTC, designed for easier learning experience
3. Include a thorough review of all adjacent technologies
4. Contain a mapping of open source and commercial WebRTC tools

WHO WOULD BENEFIT FROM THESE COURSES?

Anyone who needs to deal with WebRTC as part of his job:

- Developers and QA engineers
- System architects and Lead engineers
- Technical pre-sales, IT, DevOps and Support engineers
- Product and project managers

WHAT'S IN THIS TRAINING?

Depending on the exact course bundle you will pick, the courses include recorded video lessons along with a slew of additional documentation and resources.

The courses have been last updated in May 2024.

THE LEARNING EXPERIENCE

The courses are hosted in a modern LMS (Learning Management System), offering a focused learning experience. You can navigate between courses, modules and lessons; mark lessons as completed and favorite; take private notes.

To ease the learning and guide you through the process, a few additional tools are provided:

1. On-site chat widget
2. Slack workspace for collaboration
3. Monthly AMA meetings (Ask-Me-Anything, live sessions)

ABOUT THE INSTRUCTOR

Tsahi Levent-Levi is an Independent Analyst and Consultant for WebRTC. Tsahi Levent-Levi has over 20 years of experience in the telecommunications, VoIP and 3G industry as an engineer, manager, marketer and CTO.

Tsahi is an entrepreneur, independent analyst and consultant, assisting companies to form a bridge between technologies and business strategy in the domain of telecommunications.

Tsahi is the author and editor of BlogGeek.me, which focuses on the ecosystem and business opportunities around WebRTC. He is also a co-founder of Kranky Geek, a conference for WebRTC for developers, sponsored by Google.

Tsahi was the co-founder and CEO of testRTC, a company assisting companies to test, monitor and support their WebRTC based products. testRTC was acquired by Spearline in 2021 where Tsahi acted as the Chief Product Officer. Spearline was later acquired by Cyara in 2023, where Tsahi is now a Senior Director of Product Management.

AVAILABLE COURSES

ADVANCED WEBRTC ARCHITECTURE Tsahi Levent-Levi	19 hours 4 quizzes Monthly AMA	8 modules 56 lessons	$600
	Get you up to speed with WebRTC and enable you to make better decisions for your own product.		
WEBRTC: THE MISSING CODELAB Philipp Hancke & Tsahi Levent-Levi	4 hours	4 Modules 25 Lessons	$300
	A step-by-step guide to building your first WebRTC application.		
LOW-LEVEL WEBRTC PROTOCOLS Philipp Hancke & Tsahi Levent-Levi	6 hours	3 modules 14 lessons	$200
	Understand the protocols well enough to debug and how to approach implementing them.		
HIGHER-LEVEL WEBRTC PROTOCOLS Philipp Hancke & Tsahi Levent-Levi	7 hours	3 modules 9 lessons	$300
	Learn what protocols and algorithms WebRTC uses to improve media quality.		
SECURITY & PRIVACY ESSENTIALS Philipp Hancke & Tsahi Levent-Levi	? hours Ebook	? modules ? lessons	$400
	Get a grasp of the security and privacy tasks you need to take care of in your WebRTC application.		
WEBRTC TOOLING Tsahi Levent-Levi	4 hours	2 modules 27 lessons	$200
	Dive into the protocol and algorithms that assist in maintaining high media quality in WebRTC.		
SUPPORTING WEBRTC Tsahi Levent-Levi	8 hours	9 modules 48 lessons	$300
	Assist support teams who need to solve WebRTC related issues for their users.		

COURSE PLANS & BUNDLES

INDIVIDUAL

All courses offer 12-month subscription, online chat and Slack workspace access.

COURSE / PLAN	PLAN PRICE
Advanced WebRTC Architecture	**$600**
WebRTC: The missing codelab	**$300**
Low-level WebRTC protocols	**$200**
Higher-level WebRTC protocols	**$300**
Security & Privacy Essentials	**$400**
WebRTC Tooling	**$200**
ALL INCLUDED Developer Bundle • Advanced WebRTC Architecture • WebRTC: The missing codelab • Low-level WebRTC protocols • Higher-level WebRTC protocols • WebRTC Tooling	**$1,490** ~~$2,000~~
Supporting WebRTC	**$300**

CORPORATE

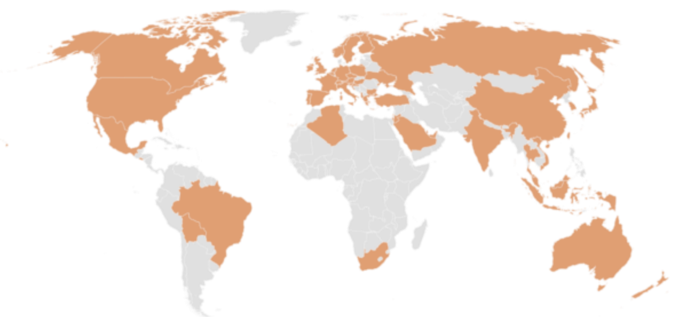
- Designed for enterprises who need to send multiple employees to the course
- Show corporate logo on course partners page (webrtccourse.com/partners/)
- Each participant has access to ALL available courses, with additional features:
  - Unlimited yearly access to all materials
  - Group management and progress tracking
  - Monthly AMA meetings
  - 2 private consultation calls for personal live training

PLAN	SINGLE	CORP 5	CORP 20	CORP 50	UNLIMITED
Plan price	$1,490	$5,200	$10,400	$22,000	$31,000
# of students included	1	5	20	50	-
Per student price	$1,490	$1,050	$520	$400	-

COMPANIES SENDING STUDENTS TO OUR WEBRTC COURSES

amazon | coviu | Foxconn | hopin | modality.ai | etisalat
motorola | LG U+ | facebook | cisco | twilio | NVIDIA
healthdirect | Ascension | salesforce | Restream | daily | puzzel.
NICE inContact | RingCentral | Honeywell | freshworks | talkdesk | HireVue
INTERMEDIA The Business Cloud | Enghouse Networks | sohonet | Dolby
crowdcast | SONY | COMCAST | kaltura
PROXIMIE | ZOHO | VONAGE | tytocare
vowel | TREND MICRO | UBISOFT | TeamViewer | bics | AIRTAME
estos | Blinkin | VSee | Gather | POSTEO | enablex
Phantom Auto | COREDIAL | Intrado | BrightTALK | ESSERT | jtel
Elbit Systems | SONETEL | globalRELAY | navis | doxy.me | webRTC.ventures

Join the **1,600+** students from more than **250** companies



Students enrolling to our WebRTC courses come from all over the globe, from companies big and small. If you are

WHAT ARE STUDENTS SAYING?

The bloggeek.me WebRTC courses are excellent! They cover the WebRTC topic comprehensively and in depth and provide perspectives for devs and architects. The course is clearly presented, well organized and easy to understand.
Tsahi Levent-Levi also provides extra resources to help with further study. One of the best features of the course is the Slack group. It brings the course alive and keeps it current. Tsahi is always available there to answer questions and he presents the 'monthly AMA' sessions which are brilliant.

Raphaella Caplan | Chief of Staff | Coviu

As a software engineer inexperienced in networking, this course gave a crystal-clear overview of how WebRTC and networking in general work. It focuses on how the market and the ecosystem are evolving, what are the challenges services are facing, and what are the pros and cons of the possible solutions.

Attila Szenczi | Software Engineer 2 | Microsoft

Tsahi's unique perspective from 30,000 ft overviews of the industry's future to detailed codelabs on WebRTC issues is difficult to match in its breadth and depth. Highly recommend his writings/reports/courses to anyone - whether it's a developer learning WebRTC, or a CPO looking for the right vendor. bloggeek.me will save you hundreds of hours of groundwork.

Aniket Behera | COO & Co-founder| 100ms

Your courses are really amazing, and I know they represent an unbelievable amount of work. The content shows your effort, obvious dedication and care for your customers. It's really, really great. Thank you.

Jamieson Becker | Founder & Chief Inventor | Userify

WebRTC is the most significant and transformative technology in enterprise communications -- and has been for some time now. Throughout its journey, I have regularly turned to Tsahi for guidance and interpretation. He is the most knowledgeable commentator on the subject, bar none. He has packaged his extensive knowledge in a course that makes knowledge transfer, from him to any student, seamless, effective, and affordable. Bravo and recommended.

Dave Michels | TalkingPointz

ADVANCED WEBRTC ARCHITECTURE COURSE

Instructors: Tsahi Levent-Levi

Prerequisites:
- Understanding software development
- Grasp of programming languages (desirable)

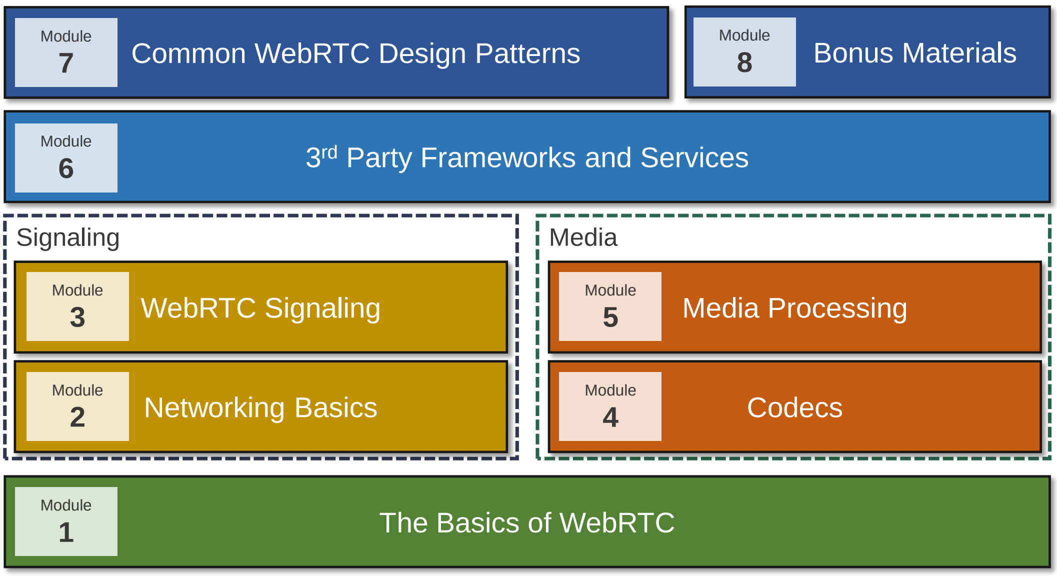
19 hours
4 quizzes
Monthly AMA
8 modules
56 lessons

This course was designed to get you up to speed with WebRTC and enable you to make better decisions for your own product. It will take you step by step through the building blocks that makeup WebRTC up to the ecosystem around it, giving you the ability to architect and design your own WebRTC applications.

To understand WebRTC, you need to know and learn many disciplines:

- How digital networks work
- What signaling is
- Ways to deal with NAT traversal
- How voice and video codecs compress media
- How to process media in large sessions
- Which tools are available at your disposal
- How to translate a use case to a solid architecture

This course answers all these questions and more. It does that through a structured building blocks system geared towards improved learning:



COURSE MODULES

MODULE 1: THE BASICS OF WEBRTC

A bird's eye view to what WebRTC is and how it works.

This module covers the history of WebRTC as well as its current state. Various touch points of WebRTC and what APIs are available to developers and where will be discussed.

Lessons:

- **Introduction** – introduction to the course. An explanation to all course modules and the logic behind them
- **What is WebRTC?** – a look at what is WebRTC and what is it used for
- **Browsers and device coverage** – understanding where WebRTC is available, on which browsers and mobile devices. What do developers do when they want to use it where it isn't available
- **WebRTC APIs** – an overview of the JS APIs provided by WebRTC and the basic structure of a peer connection and its media elements

MODULE 2: NETWORKING BASICS

WebRTC relies on computer networking. This module explains the various transport protocols with a focus on browser networking and WebRTC related aspects of it, such as NAT traversal.

You'll understand why and how ICE is used in WebRTC sessions.

Lessons:

- **TCP and UDP** – the underlying building blocks of the modern internet. A look at what these transport protocols are and what they are used for
- **HTTP** – HTTP is how we surf the internet today. This lesson will review how HTTP is used, along with the progress from HTTP 1.0 to today's HTTP/3
- **WebSockets** – the modern generation of interactive communications in browsers. This transport protocol plays an important role in most WebRTC applications
- **NAT Traversal** – in this lesson, we cover firewall and NAT limitations on communications and the solutions available to WebRTC: ICE, STUN and TURN
- **The 3 layers of VoIP** – a philosophical look at today's VoIP applications and where innovation takes place
- **Media quality metrics** – the main metrics affecting media quality that you need to understand and know about

This module includes a quiz

MODULE 3: WEBRTC SIGNALING

WebRTC has not defined any standard signaling, but it is still needed for WebRTC to work. Here we will touch the various transport and signaling protocols available to you and discuss topics such as SDP and security in WebRTC. By the end of this module, you will be able to select the signaling module for your application.

Lessons:

- **Transport protocols in WebRTC** – the underlying communication mechanism used by WebRTC applications is the transport protocol. The various alternatives are reviewed in this lesson
- **Signaling protocols in WebRTC** – WebRTC offers a wide variety of signaling protocol alternatives. From SIP and XMPP to MQTT as well as proprietary solutions. This lesson will review the options available and which signaling protocol to choose for what use case
- **SDP** – Session Description Protocol along with JSEP play an important role in connecting media sessions on WebRTC. This lesson covers this topic
- **Security** – security in WebRTC happens in all layers of the stack. This lesson deals with how security is handled by WebRTC and the application using it
- **Screen sharing** – how screen sharing is implemented in WebRTC applications

This module includes a quiz

MODULE 4: CODECS

To send audio and video means dealing with Codecs. This module covers all you need to know (and more) about the codecs that WebRTC uses. You will be able to pick the codecs you'll want to use for your WebRTC service once we're done with this module.

Lessons:

- **Basics of voice codecs** – an overview of voice codecs with a special focus on VoIP characteristics of voice codecs
- **Opus** – explanation of the most important voice codec in WebRTC
- **Other voice codecs (and WebRTC)** –what other voice codecs are widely used and how to connect them to WebRTC when needed
- **Basics of video codecs** – an overview of video codecs with a special focus on VoIP and real time communications characteristics of video codecs
- **VP8 and H.264** – the mandatory to implement video codecs in WebRTC. Where are they available, and when to use each of these alternatives
- **VP9 (and HEVC)** – a look at VP9 along with its royalty bearing alternative HEVC. We will see when and how are these video codecs useful
- **AV1** – a look at the future of WebRTC video coding. Here we cover AV1 and the Alliance of Open Media, trying to see when AV1 will become relevant

This module includes a quiz

MODULE 5: MEDIA PROCESSING

Group calling? Recording? These are tough nuggets in media processing. This is why we have this module, which goes through the various multiparty architectures, explains them in detail along with the best approaches of using them.

Lessons:

- **RTP and RTCP** – how RTP, RTCP and SRTP are used in WebRTC as the media transport protocol. Includes an explanation of multiplexing and bundling of sessions
- **Bandwidth estimation** – review of the bandwidth estimation algorithms in WebRTC
- **Mesh** – mesh architectures in group calls
- **Mixing (MCU)** – Multipoint Conferencing Unit and mixing media inputs in large group calls
- **Routing (SFU)** – Selective Forwarding Unit and routing media inputs in large group calls. Includes explanation of simulcast and SVC
- **Recording** – recording alternatives in WebRTC. Includes both server-side and client-side recording solutions. When to choose which option

This module includes a quiz

MODULE 6: 3RD PARTY FRAMEWORKS AND SERVICES

Here's a secret for you: No one develops using WebRTC from scratch these days. Everyone uses a framework of sorts, be it open source or commercial. After going through this module, you'll understand the ecosystem around WebRTC and be able to pick tools into your WebRTC technology stack.

Lessons:

- **Development strategies for WebRTC** – how different vendors are tackling development of WebRTC applications
- **Signaling alternatives** – a review of available open source, commercial and hosted/managed alternatives for WebRTC signaling
- **Media server alternatives** – an explanation and high-level architecture review of the popular open-source media servers for WebRTC
- **WebRTC PaaS** – CPaaS, Communication Platform as a Service, offering WebRTC capabilities. Who uses them and why
- **VoIP frameworks** – a look at VoIP specific WebRTC solutions that need to connect to SIP and PSTN
- **Testing and monitoring** – a look at DIY, open source and commercial alternatives to testing and monitoring, along with challenges and best practices in this domain

MODULE 7: COMMON WEBRTC DESIGN PATTERNS

This module acts as a summary for this course. It holds a lot of common design patterns of applications, so you can go through them and see how all the lessons fit into a single architecture of an application.

Lessons:

- **Media flows in WebRTC** – examples of different media flows in WebRTC use cases, covering media servers and various NAT traversal scenarios
- **Meetings recorder** – how to add media recording capabilities to an application. What are the alternatives available? When to use which option
- **Multiparty conferences** – design of multiparty video service, with a deep dive to UI/UX and screen layout aspects of the service
- **Webinar / Low latency live broadcasting** – how to build and scale a broadcasting service using WebRTC
- **PSTN connectivity** – connecting WebRTC to legacy telephony networks
- **Optimizing large group calls** – an overview of the tools available to us in WebRTC when it is time to implement multiparty video or voice conferences
- **User allocation in large sessions** – walkthrough decisions you need to make when dealing with large scale, geographically dispersed group sessions

MODULE 8: BONUS LESSONS

The "etc" module of this course. Here, lessons that just didn't fit elsewhere but are highly relevant and important found their place. They range from requirements, through standardization processes all the way to media algorithms and TURN configuration and deployment.

Lessons:

- **WebRTC standardization** – explanation of how WebRTC standardization works and what to expect of it moving forward
- **Writing RFP requirements for WebRTC** – a suggested workflow and template for creating requirements for WebRTC applications
- **Media algorithms** – a look at adaptive jitter buffers, echo cancellation and other media algorithms used in VoIP and WebRTC
- **Mini: WebRTC server-side basics** – the free mini course on WebRTC servers
- **Mini: H.264 vs VP8?** – the free mini course on selecting a video codec for WebRTC applications
- **Mini: Effectively connecting WebRTC sessions** – how to connect more of your sessions properly
- **WebRTC 1.0** – a webinar recording about the changes made to WebRTC 1.0
- **Live: Video quality in WebRTC** – a webinar recording of aspects related to video quality of WebRTC applications
- **Live: Deploy (co)TURN on AWS** – a webinar recording of deployment configuration for COTURN servers on AWS

WEBRTC: THE MISSING CODELAB

Instructors: Philipp Hancke & Tsahi Levent-Levi

Prerequisites:
- Node.js and JavaScript

4 hours

4 modules
25 lessons

This WebRTC Codelab is what you were missing with WebRTC. Diving right into the WebRTC code by way of an example has never been easier.

During this WebRTC codelab, we will build together a Node.js application. The application will implement 1:1 WebRTC video calling service to web browsers.

This WebRTC codelab is like no other.

It comes as a clean, up to date implementation of a WebRTC P2P scenario. Developed by the one and only WebRTC expert Philipp Hancke.

We explain the code in several lessons. In each lesson we discuss the reasoning behind how the code was written. We talk through the pitfalls developers need to beware of – **this one is golden** – we'll get to prod into Philipp's head and the reasons he chose to use the WebRTC APIs the way he did, which can save you hours of debugging and users' frustrations in the future.

The codelab itself has a set of exercises. These exercises take the learners through an investigative tour of WebRTC elements. Something you need to do in real life while working on your own WebRTC project.

MODULE 1: OVERVIEW

An introduction to the codelab:

- **Overview** – welcome to the codelab!
- **Preparation** – get the code and prepare your environment for the codelab
- **Run the sample** – get acquainted with the codelab sample application

MODULE 2: WALK-THROUGH

Review and understand the code in the sample application:

- **index.html walkthrough** – get acquainted with the index.html file that the browser will run in our reference app
- **Signaling 101** – an overview of the signaling mechanism we are using in this sample
- **Signaling 102** – walkthrough of our signaling protocol and its implementation on the browser side
- **Setting up STUN** – configuring STUN on our peer connection the correct way
- **Muting and unmuting** – how to mute and unmute audio and video in a WebRTC session
- **Screen sharing** – how do we switch the session to a screen sharing one?
- **Hanging up** – explaining how hangup and session closure takes place with WebRTC and the signaling around it
- **Connection states** – review some important connection states in WebRTC and what to do with them
- **Look at statistics** – time to see how we can make use pf getstats()
- **Summary**

MODULE 3: EXERCISES

Codelab related exercises (with solutions):

- **Adding TURN via Twilio** – how to add TURN server configuration using Twilio NTS
- **Change maximum bitrate** – add the ability to change the video bitrate dynamically in a session
- **Tracking disconnections in the server** – how to track call disconnections in the server
- **Adding calling and answering buttons** – letting users accept or reject calls
- **Adding statistics to the view** – using getstats() to track call quality metrics
- **Codec selection** – explicitly choosing the codecs to use in a session
- **Adding metadata to a stream** – introducing the concept of stream labels
- **Turn off camera light when muting video** – cross-browser implementation that works
- **Implementing non-trickle ICE** – getting our code to work without trickle-ICE enabling us to connect to legacy telephony

MODULE 4: WEBRTC FIDDLE OF THE MONTH

Every month we record a new publicly available lesson on a specific topic in WebRTC, where Philipp shares his code in a fiddle and explains the ins and outs of that topic.

LOW-LEVEL WEBRTC PROTOCOLS

Instructors: Philipp Hancke & Tsahi Levent-Levi

Prerequisites:
- Understanding of WebRTC at a high level
- Advanced WebRTC Architecture course (recommended)

6 hours

3 modules
14 lessons

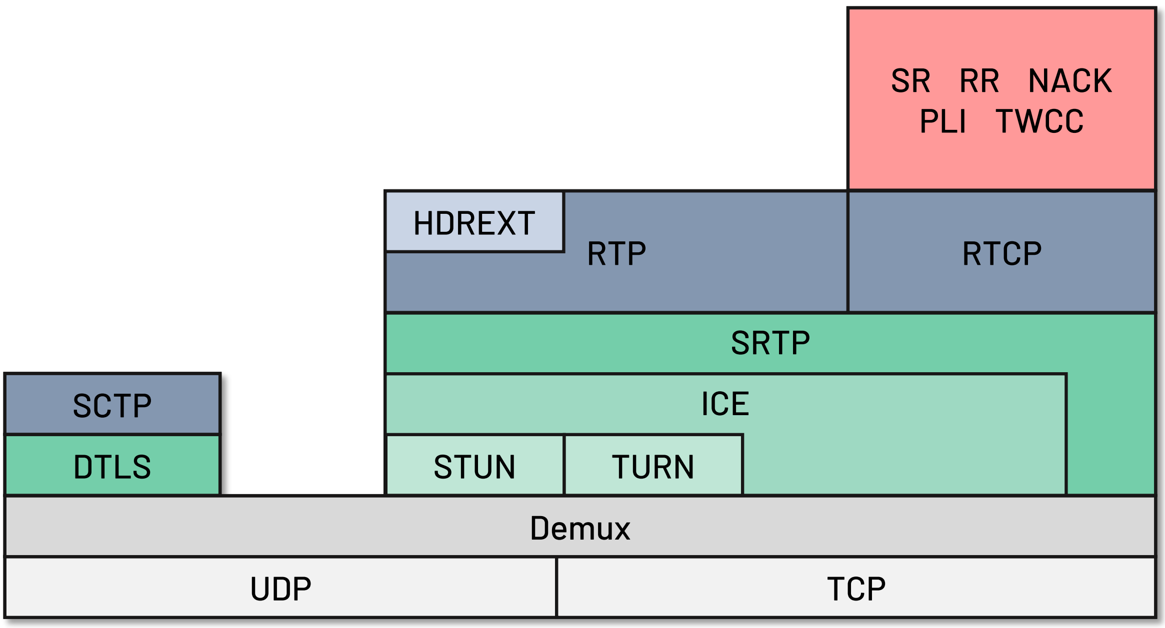
The Low-level WebRTC protocols course looks at how WebRTC behaves over the network, focusing on explaining and learning these protocols and how they interact with each other.

In each lesson of the course, we take a closer look at a network protocol that is used in WebRTC to understand its message structure and the interfaces you have with it as an application developer:

- WebRTC API calls
- Statistics available
- How it looks "on the wire"

Each lesson also includes a recorded walkthrough of using wireshark to see what you can find about this protocol while debugging your application.

This course is structured along the lines of the WebRTC protocols stack:



COURSE MODULES

MODULE 1: INTRODUCTION

An introduction to the course itself and the WebRTC protocol stack.

Here we review the course's structure and discuss what you can expect to learn in each lesson.

MODULE 2: FOUNDATIONS

The Foundations module is the main bulk of the course. Each lesson tackles a specific network protocol or concept, diving into its details. Note that we won't be teaching you why these protocols are used or for what exactly – it is assumed that you already have that knowledge.

- **STUN** – review of the STUN protocol and how it manifests itself in WebRTC
- **DTLS** – how DTLS is used to create peer-to-peer security and privacy in WebRTC
- **RTP/RTCP** – a look at real time media transport in WebRTC, with a focus on techniques such as rtcp-mux and BUNDLE
- **RTC header extensions** – the extended use of header extensions for RTP in WebRTC and how to unravel them
- **SRTP** – understand how media encryption is implemented in WebRTC
- **Demuxing RTP, STUN and DTLS** – WebRTC multiplexes most of its protocols on a single socket. Here we'll see how to demultiplex it all back while debugging
- **TURN** – handling of relay of media via TURN servers
- **ICE** – how the ICE protocol is used to orchestrate STUN and TURN traffic
- **Datachannels** – management of arbitrary data transfer over SCTP connections in WebRTC
- **Integers in wire level protocols** – an explanation to handling integers over networks (which is different than inside modern CPUs)

MODULE 3: RECAP

In this module, we review what we've learned in the Foundations module to put it all together:

- **SDP recap** – a look at SDP and how it interacts with the various network protocols
- **Interactions between the layers** – looking at how various layers and protocols interact with each other
- **Full stack recap** – summing it all up

HIGHER-LEVEL WEBRTC PROTOCOLS

Instructors: Philipp Hancke & Tsahi Levent-Levi

Prerequisites:
- Understanding of WebRTC at a high level
- Low-level WebRTC Protocols (mandatory)

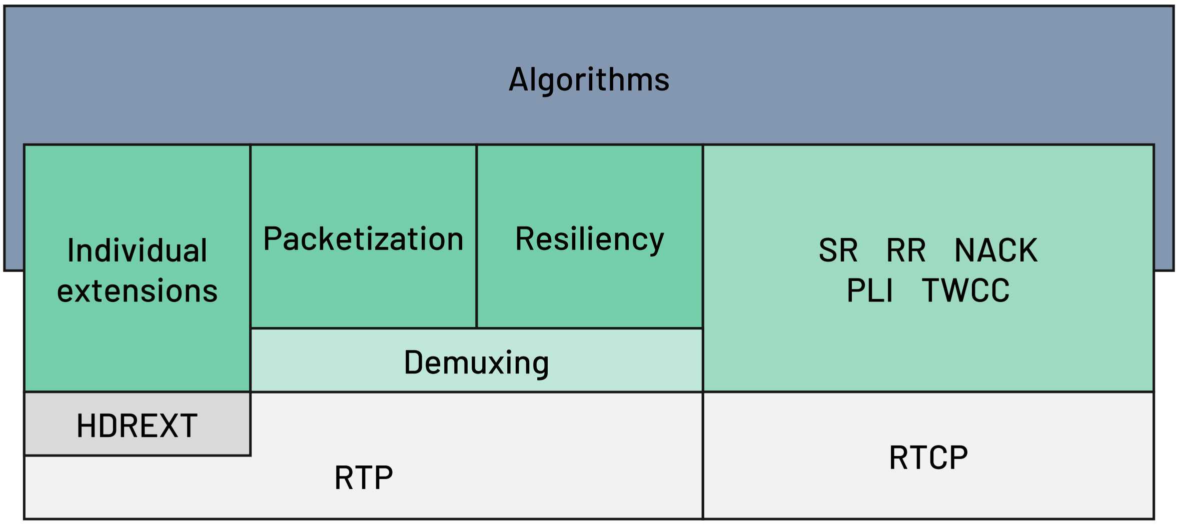
7 hours

3 modules
9 lessons

The Higher-level WebRTC protocols course focuses on the protocols and algorithms that operate on top of RTP and RTCP with a focus on maintaining high media quality.

Each lesson looks at a different aspect or algorithm in WebRTC, explaining its concepts, the protocols (standard as well as proprietary) that are used, and how it contributes to improving media quality.

This course is structured along the lines of the WebRTC protocols stack:



COURSE MODULES

MODULE 1: INTRODUCTION

An introduction to the course itself and the structure and protocols we will be looking at.

Here we review the course's structure and discuss what you can expect to learn in each lesson.

MODULE 2: PROTOCOLS

The Protocols module dives into the protocols that make up the layer on top of RTP and RTCP. These protocols are sometimes standardized but often proprietary or in a draft status only. Their main purpose is to enable WebRTC to maintain high media quality over the network.

- **RTP demultiplexing** – in WebRTC, we try to reduce the number of connections we have. Doing that requires the ability to demultiplex the various RTP streams. Here we review the process used, detailing the rules used to determine which RTP stream an incoming packet belongs to
- **RTP header extensions** – WebRTC makes extensive use of RTP header extensions. Here, we will explain what this is exactly, and review the (many) header extensions that WebRTC uses
- **Inside RTCP** – RTCP has many different report blocks which WebRTC ends up using. We will review them all in this lesson
- **Resiliency in RTP** – there are multiple resiliency mechanisms built-into RTP. We will review these mechanisms as well as understand their advantages and drawbacks
- **RTP packetization** – how do you take a media frame and convert it into one or more packets? This is where we cover this topic

MODULE 3: ALGORITHMS

In this module, we look at the protocols in the previous module and how these are used by WebRTC in its various algorithms:

- **Jitter buffers and frame buffers** – how buffering is handled by WebRTC
- **Bandwidth estimation** – what bandwidth estimation mechanisms are available in WebRTC and how they work
- **Resiliency to packet loss** – when to use which resiliency mechanism available in WebRTC

WEBRTC TOOLING

Instructors: Tsahi Levent-Levi

Prerequisites:
- Good understanding of WebRTC
- Advanced WebRTC Architecture course (recommended)

4 hours

2 modules
27 lessons

The WebRTC Tooling course offers extra materials. These offer quick and focused insights on a given framework or technique in WebRTC.

MODULE 1: IN 10 MINUTES OR LESS

A collection of focused interviews with the people behind the most popular WebRTC tools used by developers:

- **Janus** – Lorenzo Miniero
- **Jitsi** – Saúl Ibarra Corretgé
- **Kurento** – Juan Navarro
- **mediasoup** – Iñaki Baz Castillo and José Luis Millán
- **openVidu** – Pablo Fuente
- **adapter.js** – Philipp Hancke
- **Pion WebRTC** – Sean Dubois
- **Ant Media** – Ahmet Oguz Mermerkaya
- **Frozen Mountain** – Anton Venema
- **Red5 Pro** – Dima Nazarenko

MODULE 2: SNIPPETS

A growing collection of short lessons (3-8 minutes each) covering specific topics in an actionable format.

Snippets:

- WebRTC session disconnections
- ICE configuration
- A quick review of QUIC
- Synchronizing data channel and video
- What's the difference between Plan B and Unified Plan
- How does WebRTC estimate bandwidth?
- HTTP/2 or WebSocket
- Fault tolerance and high availability
- mDNS and <host>.local ICE candidates
- Best practices in media server location
- ICE, Trickle ICE, ICE lite and ICE-TCP
- What and how to log in WebRTC
- How are ICE connectivity checks conducted?
- Resolutions and bitrates for codecs
- Live transcription in WebRTC

SUPPORTING WEBRTC

Instructors: Tsahi Levent-Levi

Prerequisites:
- Technical background and inclination around software support

8 hours

9 modules

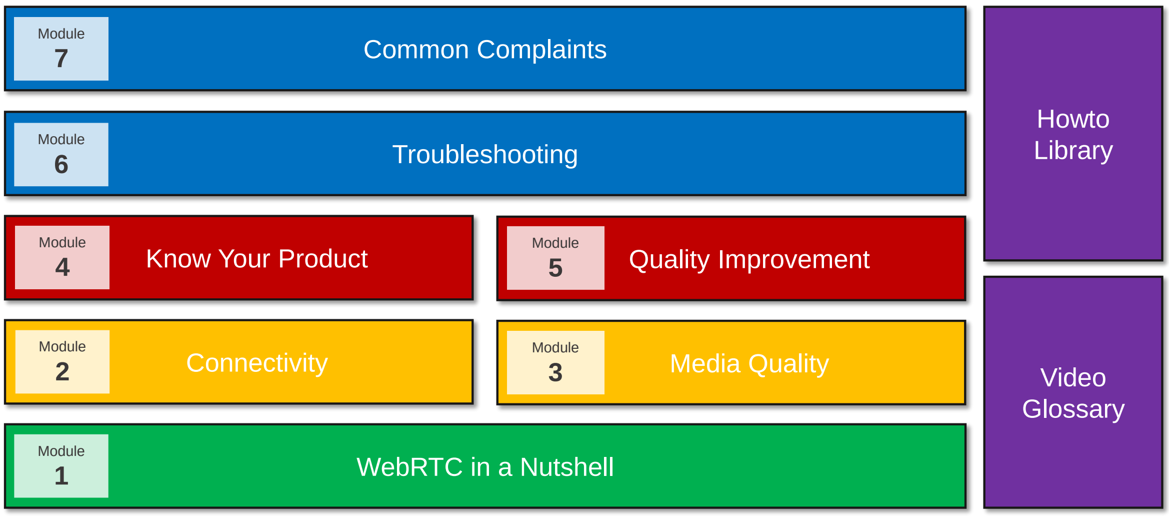
48 lessons

The Supporting WebRTC Training course is meant to assist people in support positions to analyze user complaints around network connectivity and media quality. The goal of this course is to bring students up to speed with WebRTC and enable them to solve user complaints faster and more efficiently.

This course has been created for support engineers, customer success teams, technical pre-sales and professional services experts who need to understand WebRTC at a level that allows them to communicate between the end users and the developers.

The course has been designed to be suitable for both people who come from the VoIP industry as well as those who have a web background.

It is structured as 7 modules and two additional content libraries:



COURSE MODULES

MODULE 1: WEBRTC IN A NUTESHELL

A bird's eye view to what WebRTC is and how it works.

This module covers the history of WebRTC as well as its current state.

- **Introduction** – introduction to the course. An explanation of all course modules and the logic behind them
- **What is (and isn't) WebRTC?** – a brief overview of what WebRTC exactly is and where it is used today
- **A short history of WebRTC** – an overview of the development of WebRTC through the years and where we are headed with it
- **Availability** – see where WebRTC is available today, across browsers, operating systems and devices. Understand how services are deployed in each such environment
- **Connecting a WebRTC session** – learn how sessions are connected in WebRTC and what server types are needed to get sessions to connect
- **WebRTC server characteristics** – understand what are the strains put on the various servers used in WebRTC sessions

MODULE 2: CONNECTIVITY

This module looks at how sessions are connected in WebRTC, touching the various challenges and potential places where things can break.

- **WebRTC signaling & SDP** – an overview of how signaling takes place in WebRTC and a glance at SDP messages and their content
- **Disconnections in WebRTC** – the most common reasons for sessions getting disconnected in WebRTC
- **NAT traversal and ICE negotiation** – dive into the ICE negotiation process, how STUN and TURN work and why we need all this to connect WebRTC sessions
- **ICE alternatives & additions** – additional mechanisms to ICE as well as several alternatives when connecting WebRTC sessions. Here we cover everything from ICE-lite to proxies and VPNs
- **Configuring firewalls & common issues** – firewall configuration issues and how to handle them, with some suggested common approaches
- **Location sensitivity** – how does location affect media quality of WebRTC sessions and what can be done about it

MODULE 3: MEDIA QUALITY

WebRTC has its own codecs and media processing. In this module, we cover everything related to codecs and the metrics associated with media quality.

- **Codecs 101** – an overview of how codecs work and what characteristics do they carry
- **Audio codecs in WebRTC** – a dive into Opus and G.711 and the role they play in WebRTC sessions
- **Video codecs in WebRTC** – an explanation of video coding and the various video codecs that are used in WebRTC
- **Media & networks** – the effects the network has over real time media
- **SRTP & jitter buffers** – how SRTP and jitter buffers work with real time media
- **Important quality metrics** – a review of the important quality metrics to look at in WebRTC sessions

MODULE 4: KNOW YOUR PRODUCT

Different products and use cases handle WebRTC differently. This module explains the various common options and assists you in mapping your own product, how it uses WebRTC, and what specific issues your users might complain about based on its architecture.

- **Your product map** – map the various infrastructure pieces of your product so you know what potential complaints users may have
- **Peer-to-peer** – when are peer-to-peer networks used in WebRTC and what support challenges do they have
- **Contact centers** – different uses of WebRTC in contact centers and how each affects the type of support needed
- **Media servers** – understand how media servers are deployed and what kind of issues may arise due to them
- **Mesh, mix, route** – overview of different group conference architectures and their uses in WebRTC
- **TURN deployment** – what you need to know about the way TURN has been deployed for your service

MODULE 5: QUALITY IMPROVEMENT

Group calling? Recording? These are tough nuggets in media processing. This is why we have this module, which goes through the various multiparty architectures, explains them in detail along with the best approaches of using them.

- **Improving the network** – how networks affect media quality and how can we assist users in improving it
- **Improving audio** – what can we do to improve audio quality in WebRTC sessions
- **Audio headsets** – what do you need to know about the headsets you are using to improve the audio quality for WebRTC sessions
- **Improving video** – what can we do to improve video quality in WebRTC sessions

MODULE 6: TROUBLESHOOTING

Once a client complains, you will need to start collecting data. This module focuses on the data and tools available to you and how to use them.

- **Reading webrtc-internals** – how is webrtc-internals file structured and how do we read it to find issues?
- **Using qualityRTC** – how to use testRTC's qualityRTC to understand the user's network condition and behavior

MODULE 7: COMMON COMPLAINTS

Once a client complains, you will need to start collecting data. This module focuses on the data and tools available to you and how to use them.

- **Zoom works better** – what to do when someone complains that your service doesn't work as good as Zoom (or others)?
- **It takes too long to connect a session** – how can you analyze why connecting a session for a client takes a long time?
- **Device is heating up** – what to do when users are complaining about their device (smartphone, table or laptop) heating up
- **Video resolution/quality not high enough** – dealing with complaints about video resolution and quality being too low

HOW TO LIBRARY

These are lessons that go over a few of the tools, just explaining how to use them to get the job done a bit faster.

- **How to download webrtc-internals file?** – quick instruction on how to download a webrtc-internals file

VIDEO GLOSSARY

Common terms used in WebRTC during support session, explained in short video clips.

- Latency (& round trip)
- Jitter
- Packet loss
- MOS
- Codec
- Bitrate
- Firewall
- NAT
- STUN
- TURN

BONUSES

Interview sessions conducted about WebRTC support issues with vendors.

- **Coviu**: How we support our users
- **Whereby**: Supporting with humans
- **Cosmos Video**: Troubleshooting user issues

STILL NOT SURE?

If you are still unsure if the courses are for you or your company, then I invite you to schedule a meeting with me. Just email me at tsahil@bloggeek.me.

During this meeting, we will conduct a quick discovery session, to review what are the challenges and requirements you and your team are facing with WebRTC. If needed, we can schedule a private webinar to go over the various courses available to the team, so you can better assess which plans are suitable for whom inside your company.