Video APIs

The post-Twilio Programmable Video world of interactive video solutions

May 2024

Licensed for [YOUR COMPANY]

Tsahi Levent-Levi
tsahil@bloggeek.me

BlogGeek.Me

Table of Contents

List of Figures and Tables

Figures

Tables

Introduction

What is this report all about?

Twilio Programmable Video is shutting down. This, along with many other shifts in our interactive video industry, is placing developers and entrepreneurs at flux. What should be the approach they take? Should they build the needed infrastructure on their own or rely on a third party? Would that third party be with them tomorrow as well?

In this report, I will try to provide you with as much information as possible to make an informed decision that will reduce the risks in front of you in this decision.

The report is suitable for:

- Companies who are using Twilio Programmable Video and are now forced to switch to another solution
- Other companies who are starting off a brand new project and need interactive video in their solution, wanting to decide which Video API vendor to use or should they go alone instead
- Video API vendors who want to better understand how the changes in the market are affecting their own business, to decide where to put their focus and resources

Key Findings

- The Video API market is a market in turmoil. Many trends in the industry are causing instability and may cause customers to shy away from relying on Video API vendors
- There are clear Video API market leaders and many niche players. The niche players put their focus on areas where the market leaders may have hard time addressing specialized requirements
- There is no 1-size-fits-all Video API vendor. You will need to figure out which vendor fits you based on your requirements and limitations
- Understanding the vendors, their origin story, features, capabilities and where they are header is critical for the selection process of a Video API vendor

About the author

My name is Tsahi Levent-Levi. I am a developer at heart. I have been working in the telecom and VoIP/UC industries for the past 25 years (and counting) in various roles: from a developer to project manager, product manager, CTO, CPO, co-founder and CEO. Most of that time, I was dealing with signaling and media products that were licensed to other developers who built their own products with them. This gives me a broad view of the market and an understanding of the challenges and opportunities that exist in the domain of VoIP and interactive video.

I came across WebRTC when it was first announced by Google and saw the potential in it. Since then, I have been watching the WebRTC space closely and writing about it on my blog: BlogGeek.me. From a hobby it became a "profession".

Today, I provide consulting services around WebRTC, CPaaS and ML/AI in communications, as well as offering an online course on WebRTC (webrtccourse.com).

I act as Senior Director of Product Management Cyara, after its successful acquisition of Spearline. Spearline acquired testRTC, a company developing a testing, monitoring and support platform for WebRTC applications. I was the Co-founder and CEO of testRTC prior to the acquisition.

Structure of the Report

This report is split into several chapters:

Report Tools

This chapter lists the additional tools provided as part of this report, geared towards assisting you in better understanding the Video API market dynamics and with your vendor selection process.

Defining Interactive Video

This chapter focuses on defining what interactive video is in the context of this report. We will review the main video use cases and detail which ones of them can be considered “interactive”. This being interactive video, we will also point out to the relevance of WebRTC as a technology in this market.

Development Approaches

This chapter goes through the various development approaches vendors take when implementing an interactive video use case. Here we will introduce terms such as COTS, OSS, CPaaS and COSS. Some are a bit less used in the interactive video space, but I believe they should be taken note of as general terms used for many IT projects these days.

Market Dynamics

In this chapter we will deal with the recent exit of Twilio and Mux from the interactive video market. We will see how the market has evolved in the last decade and try to deduce from that the various challenges and implications for both buyers and sellers in this market.

Origin Stories

Here, we will start looking more closely at the vendors. The first step is to understand where each vendor is coming from – their origin story. This will help us to better analyze the mindset and offering of each vendor later.

Trends in Video APIs

This chapter describes the trends that eventually got me to write this report to begin with. They are reshaping the market in a different way than what we've seen in the past decade. The trends we will review are: Twilio's exit from the interactive video market, the entrance of Zoom to this space, lowcode/nocode solutions, generative AI and finally, the Metaverse.

Porter's five forces analysis

This chapter is dedicated to looking at the Video API domain from Porter's five forces analysis. Here, we will look at the threat to new entrants and substitutes, bargaining power of customers and suppliers along with the competitive rivalry within the market itself. This different angle is an attempt to explain the level of threat each of these forces pose in the market.

Migration between Video API vendors

Twilio is exiting the market, and many customers are trying to decide who to migrate to. Other customers migrate from time to time from one vendor to another. Others are still trying to figure out whether they should migrate as well and what the risks involved are. This chapter looks at the migration process itself.

Comparative analysis

In this chapter we will be looking at the vendors in this market comparatively, trying to plot them in a 2D matrix based on their capabilities and strategy.

Vendor selection KPIs

This chapter outlines and details the various KPIs – key performance indicators – that I use when looking at the Video API vendors and trying to match the best vendor to the task at hand.

Video API vendors

Here, each vendor analyzed in the report gets their own space. This isn't meant to replace researching the vendor's website and offerings, but rather to offer a quick and concise summary of the most important aspects before you start your research.

Appendixes

The report also contains a few appendixes that provide additional case studies around the use of API platforms and the importance of making the right choice to fit your needs.

Membership Site and Tools

The Video APIs Report comes with a membership site that gives you access to additional articles, resources and tools.

Access to the membership site is done through the WebRTC Course site. If you don't have the account information to access these additional articles, please contact us at https://bloggeek.me/contact.

Within the membership site, you will find the following content.

Presentation Visuals

A Microsoft PowerPoint deck containing all the visuals and tables from this report.

You can use the visuals in this PowerPoint deck for your own internal presentations at your company.

CPaaS 2020 eBook

In 2020 I wrote an extensive eBook about the trends that are causing some long-term tectonic shifts within the bigger CPaaS industry.

The 3 trends covered are:

1. The pandemic, raising all boats
2. IaaS vendors getting into the CPaaS space
3. Twilio and how it is shaping the market

This eBook is available for download as part of the membership site and is as relevant as ever.

Lowcode & Nocode in Communication APIs

In 2022 I wrote an eBook about the introduction of Lowcode and nocode services into Communication APIs. An important part of this trend is the relatively new introduction of Prebuilt approaches for programmable video APIs.

This eBook is available for download as part of the membership site.

Defining Interactive Video

You sit on your sofa and switch on your television. You don't know what you want to watch, so you browse through Netflix until you settle on one show or another. You click play, wait a couple of seconds and the show starts. For that to happen, somewhere there are video files of that show that now get served to you. These have been pre-compressed, encoded in various configurations to suit different network and device types. There is also no real rush to play them back to you – people are fine with waiting a few seconds before a show starts.

The level of interactivity here? Low.

Try another scenario. You now want to tune in to the Super Bowl match. In this classic example, you're living in an apartment building. Would you be willing to hear the cheering crowds outside your window from your neighbors who got to see the kick of the winning goal two seconds before you see it? Would that be reason enough to look for a different service provider?

Your expectation here is to watch the video stream live. Within a second or two of the actual "action".

Here is our last scenario of the day. You now go online for a business meeting. As many meetings today, it is remote and virtual. During the discussion, it feels to you like the others are in outer space or on a boat far away. How? It seems like each person needs to say the word "over" and then wait a second or two until the other person replies.

This is neither interactive nor live. Our expectation though is for it to be live and interactive, with no perceptible delay. This is usually referred to as sub-second latency.

Latencies, quality, effort… and protocols

There are two important things we need to put front and center before moving forward to discuss Video APIs. These are **latency versus effort** and **latency versus quality**. Let's plot both associated concepts in a simple chart:

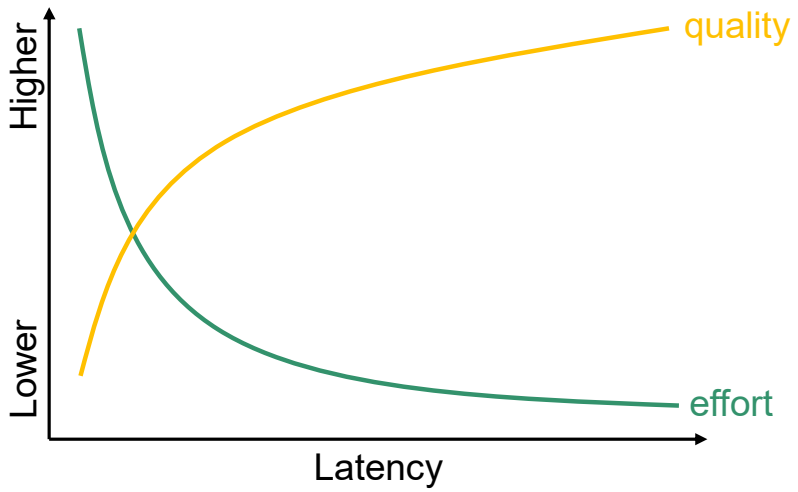

Figure 1: Latency versus effort and quality

The lower the latency, the higher the effort needed to deliver the video is and the lower the quality you can achieve is going to be.

When within that graph would we consider interactive video? The more we move to the left of the axis, reducing latency, the closer we will be to interactive video. For me, once you've reached sub-second latencies, that's when interactive video takes place. At least within the scope that this report looks at.

It is important to understand the types of protocols used to deliver video over the internet. These can be broken down based on the latency they offer. The diagram below sums up the major protocols involved:

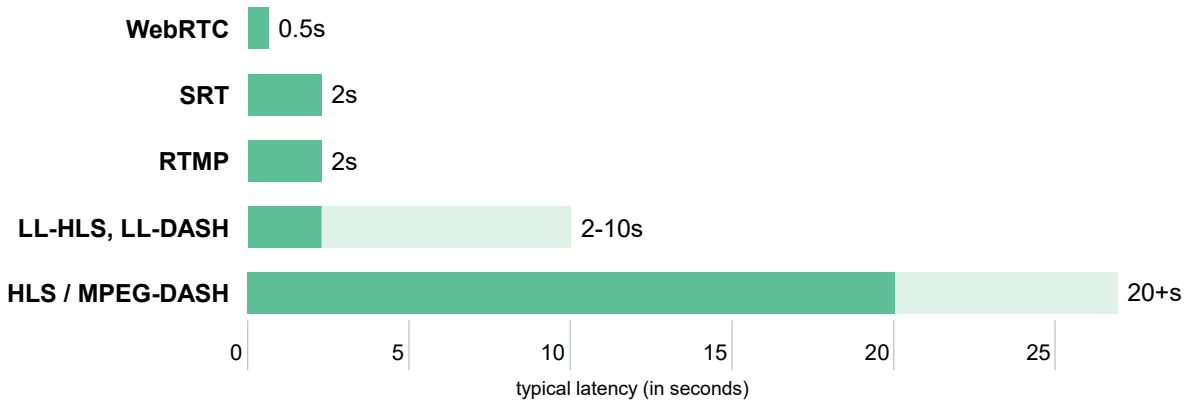

Figure 2: Media delivery protocols and their typical latency

As the illustration above shows, the only sub-second protocol that is commonly used is WebRTC. The reasons behind these are numerous. The main two are mostly likely the following:

1. WebRTC is supported natively in all modern web browsers
2. WebRTC sacrifices quality for latency. It does that since it is sole purpose is interactivity

Both these capabilities are derived from its name:

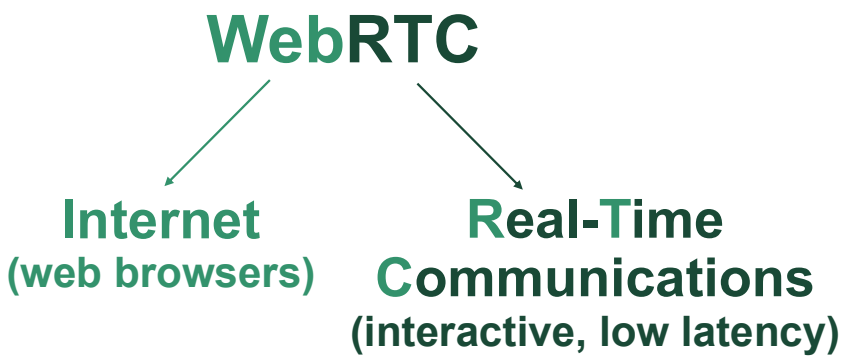

Figure 3: WebRTC in interactive video

This report won't delve into the technicalities and innerworkings of WebRTC itself. What I do want to emphasize here is that WebRTC isn't alone on the scene. There are other proprietary methods of video delivery that compete with it. When operating outside of the web browsers, vendors are free to use other protocols and technologies, and when implementing interactive video for web browsers, proprietary solutions can still be implemented using other web technologies such as WebSocket, WebTransport, WebCodecs and WebAssembly. Some of these will be discussed in the chapter covering **Zoom vs the world**.

To summarize: Interactive video is video that is generated and delivered in the span of a second or less.

The two main use cases for interactive video are: video meetings and live streaming

Interactive video, Video APIs and CPaaS

When implementing an interactive video application, there are usually two ways to get the work done:

1. **Build**. Use open source or commercial components to build out the video delivery infrastructure needed
2. **Buy**. Use a third-party hosted video delivery infrastructure platform, implementing the specific use case on top

There are many reasons for choosing Build over Buy and vice versa, and there is no single good answer here. Those who end up selecting the Buy approach, using a third-party vendor, end up adopting a CPaaS vendor.

CPaaS stands for Communication Platform as a Service and in a way, this is a catch-all phrase for using APIs on top of a communication infrastructure of one vendor used by another. Under the roof of CPaaS we usually find building blocks and low-level APIs for voice and SMS. To that, sometimes vendors would add social messaging (WhatsApp, Messenger, Viber, …), email and video. Higher level abstractions in the form of number verification, notification, chat and lowcode/nocode environments are also the norm today.

In recent years, Twilio has been the guiding light of what a CPaaS vendor offering looks like. That has changed in the last year or two, starting with the acquisition of Segment, a CDP (Customer Data Platform) vendor. This has placed Twilio on the path of data management, shifting focus from its communications background.

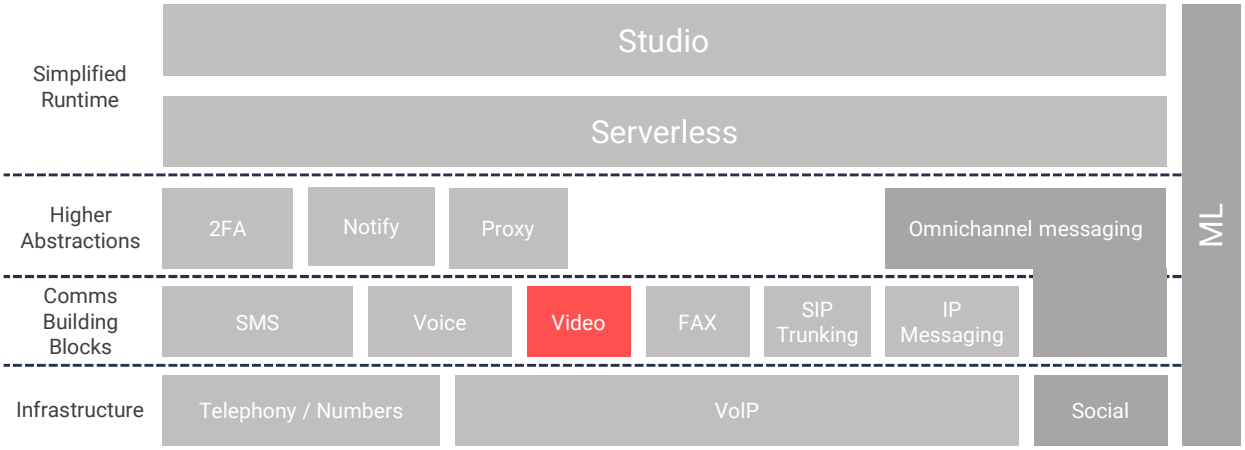

Figure 4: Typical services in a CPaaS offering

The height of this shift was selling its IOT business, sunsetting its Live video streaming service and most recently the closure of its Programmable Video service. We will dive deeper into the story of Twilio in our **Trends in Video APIs** section.

With this departure of Twilio from the realm of video, I decided to start using another term for vendors who offer interactive video infrastructure for others to build applications on top: Video APIs

To me, a Video API vendor is a CPaaS vendor that offers interactive video capabilities as part of its offering. It may or may not have support for other CPaaS related services such as voice and SMS, preferring to focus solely on video. Video might also be a small part of its offering, a complementary capability to something else (usually voice and SMS).

Live streaming vs video meetings

We've started with interactive video stating it is good for two different use cases: live streaming and video meetings. Time to make an additional distinction between these two groups, which is again going to be clustered using latency.

By and large, video meetings require lower latency than live streaming does.

For most live streaming cases, the latency of a second or a bit more is good enough. For video meetings, latencies should be well below 200 milliseconds of round trip, and that on the extreme upper limit of what we'd be comfortable using daily.

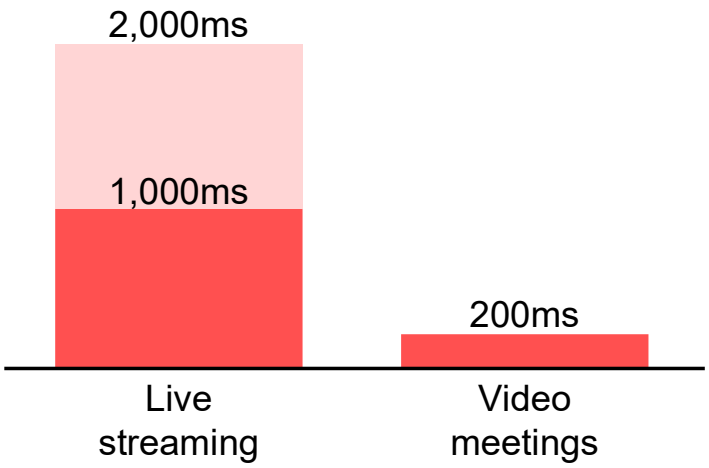

Figure 5: Latency tolerance of interactive video

This means that the technical tools available to us in each of these alternatives are quite different as well. For video meetings, we will almost always sacrifice quality to achieve a lower latency. For live streaming we can be a bit more lenient to gain better quality.

Another insight from this is that live streaming can use more media delivery protocols than web meetings. That is because it has less restrictions on its latency requirements. As such, we will focus less on live streaming in this report than we would on video

meetings. For the most part, the content and vendors covered in this report are those who offer video meetings solutions first and live streaming as a secondary feature.

There are many managed live streaming services out there. They didn't make it to this report either because they had no solution for video meetings or because their video meetings solution is weak compared to the alternatives. If what you are after is purely focused on live streaming, then this report doesn't delve into that domain enough.

WebRTC and interactive video

As we have seen earlier, WebRTC is the best media delivery protocol when it comes to the aspect of low latency. It is geared and finely tuned towards video meetings, sacrificing quality for lower latency in many scenarios. Over time, WebRTC gained adoption also by live streaming vendors: For use cases where lower latencies were desired or where these use cases need hybrid video meetings and live streaming experience.

For the scope of this report, there are a few things you need to know about WebRTC:

1. WebRTC is a standard specification that has been with us for more than 10 years now
2. The specification itself details how media is sent over the network as well as how the APIs look like in web browsers
3. It is available on all modern browsers. That includes Chrome, Edge, Firefox and Safari
4. There is a large and vibrant developer ecosystem and vendors around WebRTC. These rely on both open source and commercial solutions

Figure 6: A slide from my RTC @Scale 2024 session. Source: YouTube

Why am I drawn into WebRTC so much in this report? Simply because most of the Video API vendors out there today have WebRTC as a critical component of their technology stack. Some vendors rely purely on WebRTC, while others use it only on the edge of their infrastructure or in web browsers.

Choosing a vendor also means understanding a bit of the underlying technology used and how that affects the vendor's ability to innovate or provide certain capabilities. This report touches on these aspects as well.

For vendors supporting WebRTC only at the edge, on web browsers, and having their own proprietary technology everywhere else, my suggestion would be to conduct thorough media quality analysis on the browsers for them if you wish to use their services. This can be subpar to those who use WebRTC elsewhere, depending on your specific use case.

If you are looking for a more comprehensive introduction to WebRTC, then I can suggest my **WebRTC for Business People** report, which is available freely for download on my website.

Now that we've defined "interactive video" and explored a bit of its characteristics, it is time to shift gears and discuss the implementation part. We will do that by reviewing the various development approaches available to us.