

★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



Webmagic

# 10 Most Amazing Use Cases of WebSockets — Go Real-time!!



Vikas Sood

Follow

6 min read · Sep 29, 2017



181





In today's world consumers expect their apps to be faster and want things done in real-time by the touch of their finger on a smartphone or a smartwatch. Whether playing a game, interacting with colleagues at work,

socializing with friends and family or even shopping online. Users expect to see things on their screen instantly and hence an application's real-time nature plays a great role in building better user engagement and it's a success.

Also, never ever before there has been a greater need for IoT Applications where in devices talk to each other in Real-Time. And the world is seeing a change with the emergence of IoT based consumer products. Scalable real-time delivery of artificial intelligence is another area world is looking at.

Be it a web application, a mobile app or an IoT app, Real-time nature seems to be an obvious thing to be a part of your technology stack.

But unfortunately going real-time is often not as easy as you may imagine it to be. There are multiple complexities to deal with starting from what frameworks to choose, what networking protocols to choose, building the reliability of delivery and worst of all is the scalability of the solution to consider since most applications designed today are for web-scale.

## **Evolution of Websockets**

The traditional approach towards building web applications that require real-time communication between client and a server has required a rather intimidating use of HTTP to continuously poll the server to fetch updates and send upstream data via distinct HTTP calls.

The approach might have worked, nonetheless, it was still an abuse of the HTTP protocol. This resulted in a variety of problems.

In order to understand what are the benefits that come with the HTML Websocket protocol, first let's try and understand what were the traditional approaches and what were some of the associated problems and how HTML 5 Websocket is not merely a solution but in fact a giant leap especially for real-time, event driven web applications.

### **Traditional Ways of Building Near Realtime Experience**

Consider a web application developer developing a real-time stock prices web application, or an instant messaging app, or a news app, has to send an HTTP request to fetch data from the server. Now, this response is sure to get stale by the time the HTTP response arrives and the page gets rendered. In

order to provide the most updated data to the user, the web developer will now need to constantly poll the web server in order to provide near real-time experience to his user.

Hence most approaches that attempt to provide real-time behavior over web application tend to implement one or the other hack using polling.

## **1. Polling**

This was one of the early approaches to solve real-time problem. In this approach, an HTTP request is sent at regular intervals to the web server to get an immediate response. If you are familiar with JavaScript, you may imagine it like a `setInterval` function calling a callback function every `T` secs. The callback function sends an HTTP request gets the response and updates the page.

The problem is the nature of real-time data, that it is not predictable when an event will occur so you will end up making unnecessary HTTP requests flooding the network. And the number of network requests will only increase as the number of users increase. Not to forget each request will

carry the HTTP header as well adding an overhead every time you poll for each user.

## **2. Long Polling**

Imagine this technique as an HTTP request that does not get an immediate response, rather the server holds the request for a certain time period, during which it expects an event to happen. If an event of interest occurs during that interval the response is sent immediately, otherwise, an empty response is sent indicating that the request stands canceled.

This technique does not provide much improvement to the Polling technique. A web server could also potentially run out of request threads and end up discarding further requests.

## **3. Comet Approach**

A more favored form of comet approach is streaming, in which an application opens a single HTTP request and keeps it open allowing a web server to push data to a web browser without any explicit requests being

made for different data. Neither the server side nor the client side close this persistent connection.

The problem is that streaming is still encapsulated in HTTP messages so proxy servers may buffer the responses and thus increasing the message delivery latency.

### **HTML5 Websockets — The Life Saver**

HTML 5 Websockets provides for a full duplex bidirectional channel for communications over the web through a Transport layer socket. As described in the websocket protocol [RFC 6455](#).

Combined with the WebSocket API , web socket protocol provides an alternative to HTTP polling for two-way communication from a web page to a remote server.

With a full duplex bidirectional capability, there is no longer the need to exploit the HTTP by polling, long polling or comet approach to build real-time experience for your users.

This opens up space for endless possibilities for the Application Developers to leverage and build amazing applications.

## **Some Most Amazing Use Cases of Websockets**

### **1. Real-time Feeds**

Give your apps the ability to give real-time feeds, notifications, social likes, shares and instantly share all that with your users in real-time. Wouldn't that be extremely engaging for your audience?

### **2. Real-time Multiplayer Gaming**

Websockets API allow you to create the same gaming experience that users get on a mobile device. With the ability to send receive data at any given point of time without having to poll the web servers, creates space for building real-time multiplayer gaming.

### **3. Real-time Collaborative Editing**



Now you all have used google docs and have wondered how this works. With emergence of HTML5 Websockets, creating collaborative applications that allow users to work on same document becomes greatly simplified.

#### **4. Real-time Data Visualization**

Be it marketing trends, sales graphs, analytics, or data science plots, with HTML5 Websockets you can create visually appealing data representations that will automatically update as and when new data arrives in your backend, and that too without having the need to poll the data.

#### **5. Real-time Multimedia Chat**

Let's face it. If you are an application developer, you have always wanted to create a chat server, either as a college project or as one of your client's project feature. But you hit the wall by adopting the traditional approaches that I discussed above. Well, try HTML5 Websockets if you want to build a chat server this time. Or connect with my team, they have developed one and are already using in production now. [Contact me.](#)

## **6. Audio / Video Chat with WebRTC**

HTML5 Websockets are a best candidate to be used as a signaling mechanism for WebRTC. If you are not familiar with WebRTC, read my post [“The Amazing Things WebRTC Can Do”](#). The WebRTC draft leaves the signaling layer as an abstract and offers the implementer to create his own signaling mechanism. Websockets are a great candidate for that as they offer full duplex communication.

## **7. E-Learning Applications**

With HTML5 Websockets, E-learning is one industry that is being revolutionized rapidly than any other industry. With developers creating out of the box real-time features, transforming online learning as an extremely interactive way to engage students and offer remote classrooms just as real as a physical class room.

## **8. Real-time Location Apps**

Building applications like location based intelligence, Geo fencing, track and

trace becomes all the more simplified with HTML5 Websockets allowing developers to share location updates in real time and create amazing work flows with the real time data.

## **9. Real-time User Behavior**

A lot of E-commerce space is using click stream data to analyze user behavior in real time and that is helping them analyze how user interacts with there web applications and instantly offer them recommended content. Another aspect of it allows error detection faced by consumers to be reported in real-time.

## **10. Real-time Sports / Event Updates**

Users are so used to of real-time facebook and twitter updates, and that has set very high standards for other web and mobile applications. By using HTML5 Websockets you can use a common platform for all your channels to provide real time updates to your users and hence build better and more engaging apps.

## Wrapping Up

I am quiet convinced that Websockets are indeed revolutionizing the way applications are built now and how HTTP is slowly becoming obsolete. The question is how soon are you going to include websockets in your technology stack? Need help with websockets, we are just a click away. [Reach us for a free consultation and demo.](#)

. . .

*Originally published at [pushfyi.com](https://pushfyi.com) on September 29, 2017.*

Html5

Programming

Web Development

Software Development

Technology



## Published in Webmagic

30 followers · Last published Jun 14, 2018

Follow

Webmagic Technologies is driving the digital innovation front for its customers by helping them build great software products and solutions. Explore our success stories and connect with us.



## Written by Vikas Sood

24 followers · 7 following

Follow

15+ Years in the IT Industry, and currently Architect and CTO at PushFYI Inc.

No responses yet



Medium



Search



Write



All your favorite parts of Medium are now in one sidebar for easy access.

[Okay..got it](#)



Profile

What are your thoughts?



Stories



Stats



Following



Gaurav Goel



The Medium Blog



Reshma Bidikar



Find writers and  
publications to follow.

[See suggestions](#)

## More from Vikas Sood and Webmagic



In Webmagic by Vikas Sood

### Building High Performance Applications - Lessons Learnt, Pa...

A compilation of lessons I learnt while developing high throughput applications. I...

May 13, 2018



31



In Webmagic by Vikas Sood

### The amazing things about WebRTC and why should you adopt it?

I'm, sure you have heard that by 2021, video traffic is going to be more than 82% of all...

Sep 29, 2017

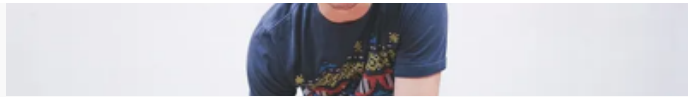



40



1





 In Webmagic by Vikas Sood

## Building High Performance Applications — Lessons Learnt,...


Welcome to the Part 3 in the series. In this Part, I will suggest you to adopt a staged...

Jun 14, 2018



3



 In Webmagic by Vikas Sood

## Building High Performance Applications-Lessons Learnt, Par...

Part 2 in this series covers how software architectures have evolved over the last...

May 22, 2018



44



See all from Vikas Sood

See all from Webmagic

## Recommended from Medium



Tosny

## 7 Websites I Visit Every Day in 2025

If there is one thing I am addicted to, besides coffee, it is the internet.



Sep 23




5.3K



189



Will Lockett 

## The AI Bubble Is About To Burst, But The Next Bubble Is Already...

Techbros are preparing their latest bandwagon.



Sep 15



14.9K



508



The CS Engineer

Forget JSX—These 4 Data



Abhinav


Docker Is Dead—And It's About



## Forget JSON—These 4 Data Formats Made My APIs 5× Faster

A 1.2 KB JSON payload transformed a smooth request into a 120 ms wait. Switching format...

★ Sep 30 🖱️ 1.2K 💬 35 📖<sup>+</sup> ⋮

 Himanshu Singour

## How I Learned System Design


– The honest journey from total confusion to clarity

Aug 7 🖱️ 5.7K 💬 181 📖<sup>+</sup> ⋮

## Docker is Dead—And It's About Time

Docker changed the game when it launched in 2013, making containers accessible and...

★ Jun 9 🖱️ 7K 💬 200 📖<sup>+</sup> ⋮

 In Level Up Coding by Fareed Khan

## Building an Agentic Deep-Thinking RAG Pipeline to Solve Complex...

Planning, Retrieval, Reflection, Critique, Synthesis and more

★ 5d ago 🖱️ 927 💬 7 📖<sup>+</sup> ⋮

[See more recommendations](#)

[Help](#) [Status](#) [About](#) [Careers](#) [Press](#) [Blog](#) [Privacy](#) [Rules](#) [Terms](#) [Text to speech](#)