PHIL LEGGETTER

HEAD OF DEVELOPER EVANGELISM AT PUSHER
& REALTIME WEB CONSULTANT

# History, Background, Benefits & Use Cases of Realtime

28 Oct 2013

This is part 1 of a 3 part series on **Choosing your Realtime Web App Tech Stack**. It's based on the talk I recently gave at the Future of Web Apps.

- Part 2: Fundamentals of the Realtime Web & Realtime Web Functionality
- Part 3: Choosing your Realtime Web App Tech Stack

For some time I've seen FOWA as one of the biggest web conferences around. So, when Future Insights approached me to talk at FOWA 2013 I was very excited (and nervous). Now that the event has been and gone I'm pleased to say that both the event and my talk went well.

Since leaving Pusher I've kept a keen eye on the realtime web tech scene. And since I'm now unaffiliated to one realtime company (although I have to admit I've still a soft spot for Pusher) I can now be even more open about other technologies and how they compare (I do like to think I was always very open and honest anyway).

Below you can find a summary of my FOWA 2013 talk. If you simply want to dive into the slides you can find them here.



My name is **Phil Leggetter**. I'm He
Developer Evangelism at Pusher.

I frequently write articles and give
Realtime Web Technologies and g
technology. Get in touch if you'd l
talk at your event or write an artic
I'm also very interested in develop
experience and productivity, APIs
customer service.

I'm also the co-author of the APre
beginners title "Realtime Web App

**Realtime Web Apps: With H
WebSocket, PHP, and jQue**



Buy the book I co-write with Jaso via Amazon.com or Amazon

A talk on **Choosing your Realtime Web App Tech Stack** seemed an appropriate way to both demonstrate that and share some of the knowledge I've aquired since working in and around realtime web tech since 2001.


In 1991 the first documented version of HTTP was released. HTTP is a request-response protocol where a client initiates a connection and request for data, the server responds and the connection is closed.

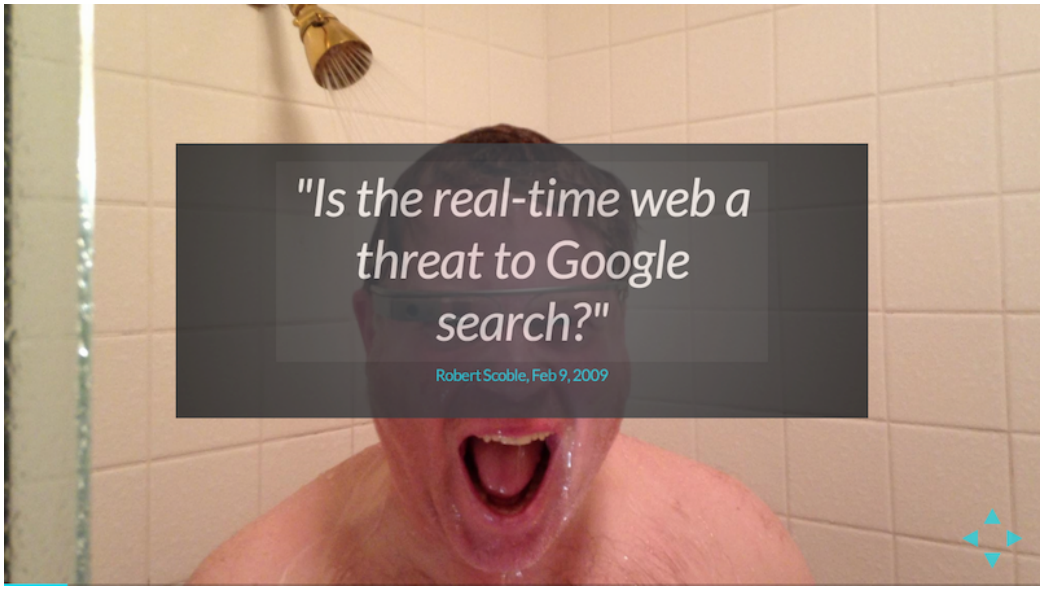
To achieve realtime web communication a connection needs to be maintained so that if either side of that connection has new data it can instantly be sent to the other party.

It wasn't until 1996 that it became possible to create a persistent client-server connection in a web browser with the help of a Java Applet.

However, Java Applets never really become a popular - or viable choice - due to a lawsuit, generally flakey Sun JVM LiveConnect implementation and a general resistence for a requirement on browser

plugins.

As ever, developer needs prevailed and around 2000 we managed to hack some purely native web-browser solutions using persistent HTTP connections.




So, realtime web technologies have been around since 2000 where it was primarily used by companies either in, or targeting, finance. But it wasn't until a number of years later that the use of the technology went mainstream.

In 2009 Robert Scoble wrote a blog post asking "Is the real-time web a thread to Google search?" At that time he was using a service called FriendFeed and was very excited that as soon as he posted information on the network that it was **instantly discoverable** through search.

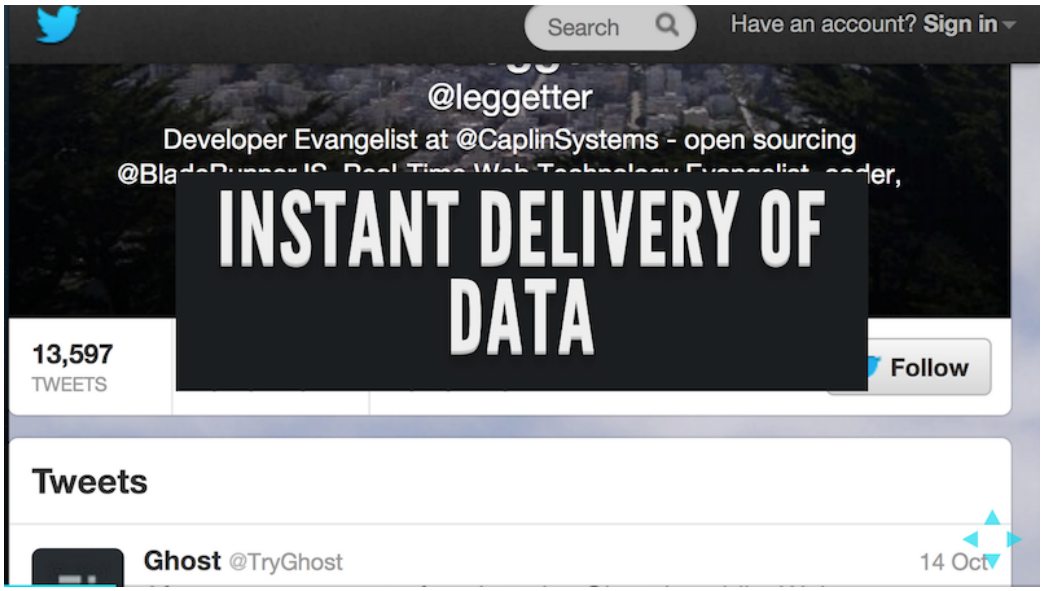
Back in 2009 we were all quite used to waiting days for our new online content to be discovereable. We would post a blog post, or create a new website and wait days for search engines like Google to find out content.

> *Let's do a search for anyone who has written about the Canon 5D MK II but lets constrain that to posts that have at least one like and at least four comments. […] Note that the post I wrote just one minute ago is already in the results page. This is the real-time web.*

FriendFeed's functionality suggested that this no longer had to be the case; that the realtime web was going to change all that. Ok, this wasn't to the same scale as Google, but it was to a reasonable scale.

Back then, even Wikipedia had no definition for "real-time web". The first definition was added in mid-2009 and was search-focused.


Now, however the definition is very much focused on the **Instant Delivery of Data**.




This has been partly enabled by social networks' ability to know who is interested in data. We now:

- Follow people on Twitter
- Friend people on Facebook
- Add people to Circles on Google+

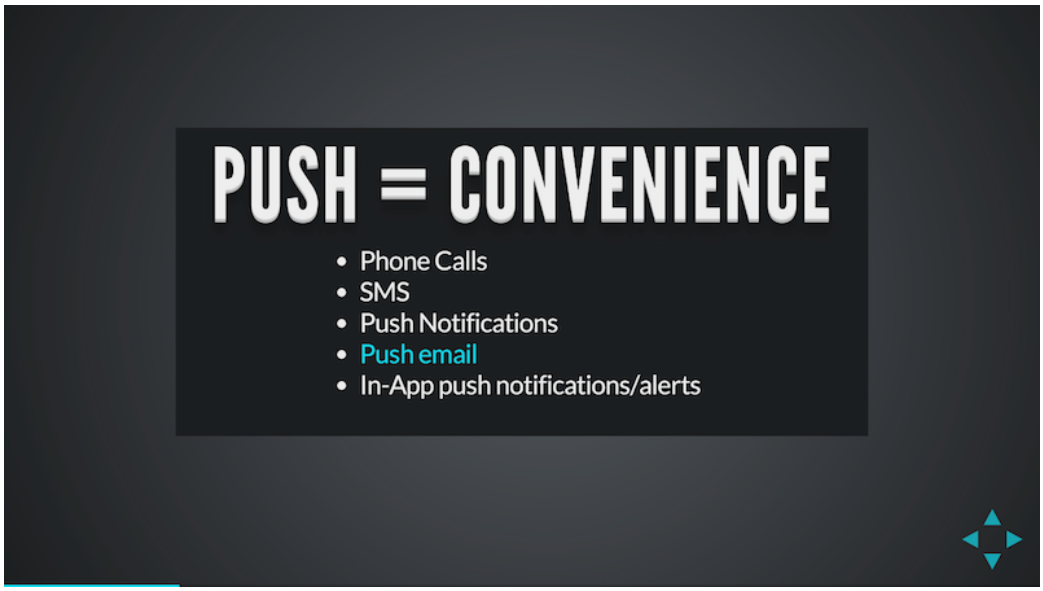
All of these represent a data subscription to updates and posts. This means these social networks know what data we are interested in so as soon as it's available it can be delivered to us. This type of functionality is ideal for realtime web technology.


You shouldn't use this technology to control the safety system in your neuclear power station. Nor should you use it to control the breaks in your car. But it's perfectly capable of being used to send data in less than 300ms. I have benchmarks that show that you can get the average lower than 100ms. So, we're talking "soft realtime".


What is very important is that the time between the data being generated and it being delivered needs to be small enough for data to maintain its relevance.

If the data isn't delivered within a timely manner it has the potential of losing relevance and context. Realtime web technologies actually enable you to instantly deliver data and therefore maintain context. **I think the fact that realtime technology can help you maintain context is going to be a big factor in the technology is going to be increasingly important**.


What are the benefits that realtime enable and how is this technology being used right now? Here are a few areas where I believe realtime is important along with a few examples where the technology is being used and making a difference.

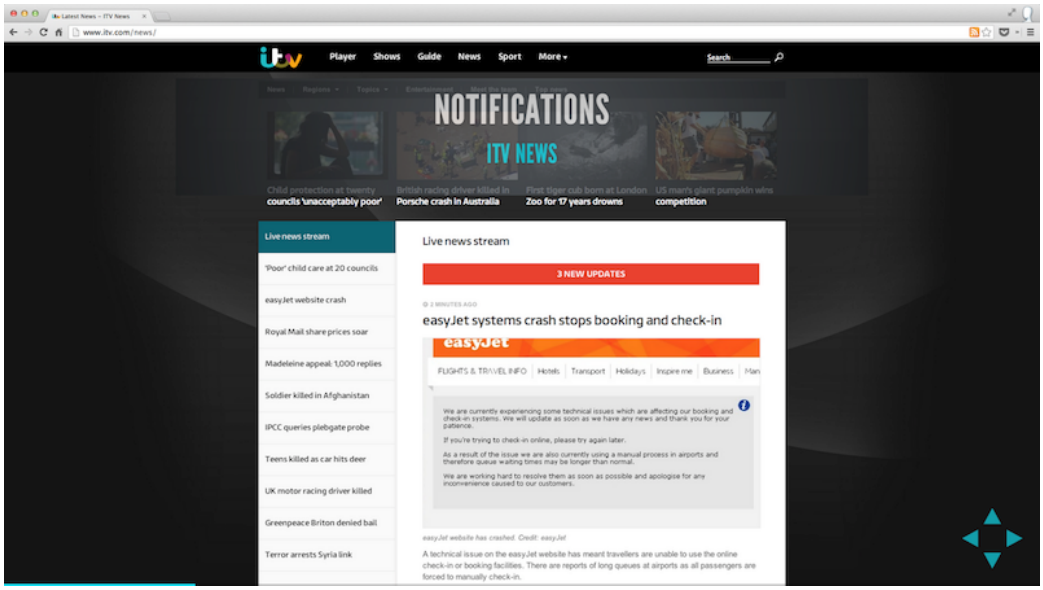



The fact that you can push information to those that have registered interest is *convenient*.

It's a bit of a silly example, but: if you had to open a phone or SMS application and keep hitting a button to see if you can any incoming phone calls or SMS messages that really wouldn't be convenient.

The same therefore goes for data within applications; as new information becomes available or some other call to action is available (e.g. phone rings -> pick up the phone) then the user should be made aware.

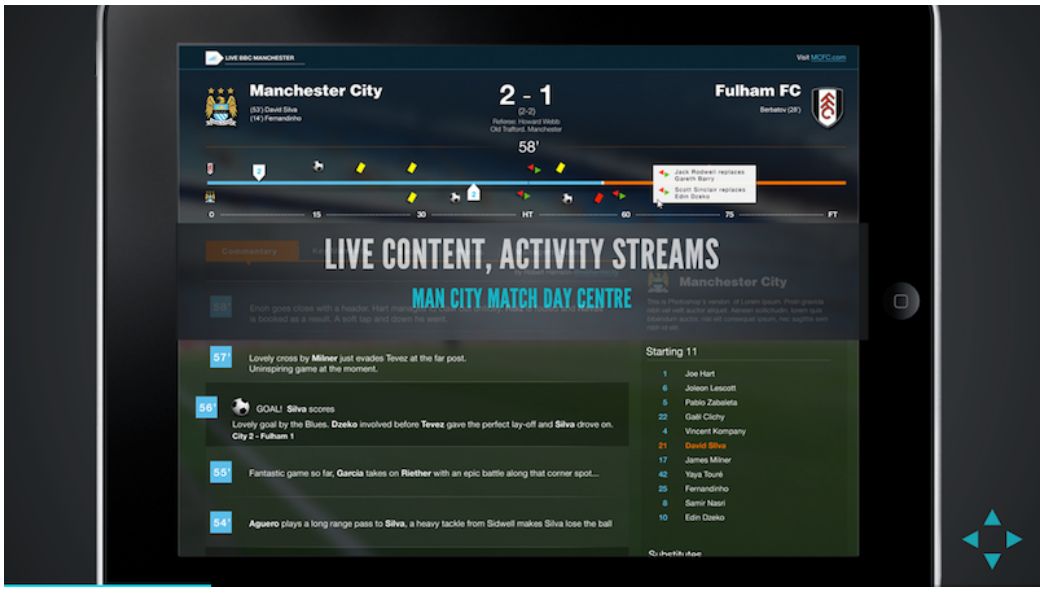
Realtime Web Tech makes this a possiblity in web and mobile applications.




The first and simplest example of using Realtime technology is ITV News. Around a year ago they had a site redesign and moved towards an **Activity Stream** look and feel. Along with this they added realtime content delivery and notifications. The content isn't instantly displayed. Instead a notification is shown to say the new content is available; when the notification it slides in.

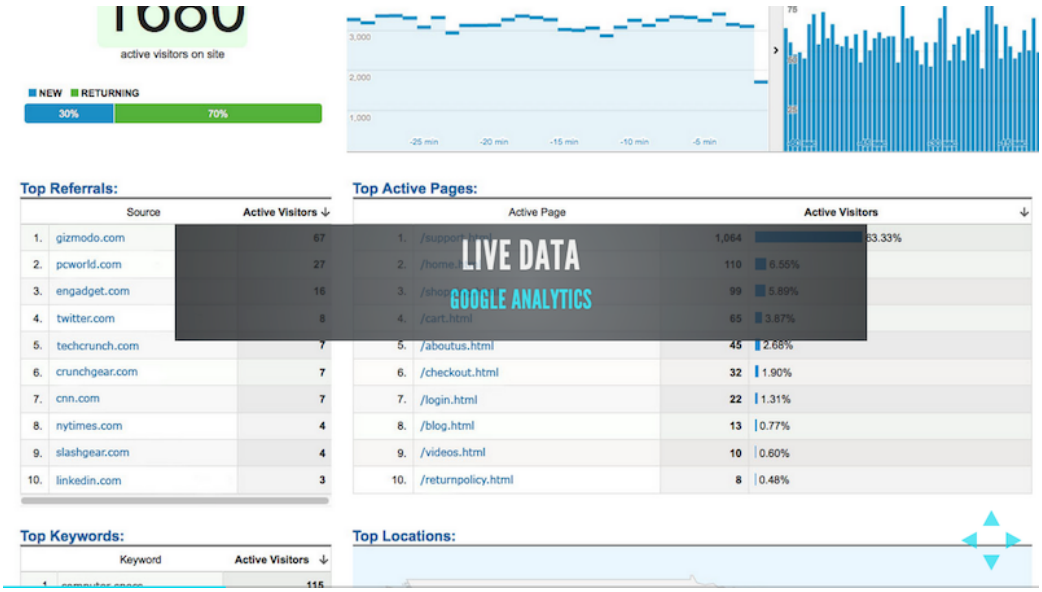
While this may not be the most innovative use of realtime technology it has worked. Around 6 months ago I was told that the site redesign, along with this realtime functionality, resulted in a 10-fold increase in site traffic.

ITV News has become a destination for **realtime news**.




Manchester City released a Match Day Centre application at the start of the season. It shows **realtime stats** about an in-play game. I used to love Championship Manager and Football Manager style games so I really get this. I even created a demo for Opta Sports when I was working on the Kwwika project that looked very similar to this.

I believe this is a great example of a **2nd Screen Experience** that offers lots of value. It really complements watching a live game - you just can't integrate all that data into the main screen. It's also really valuable standalone.



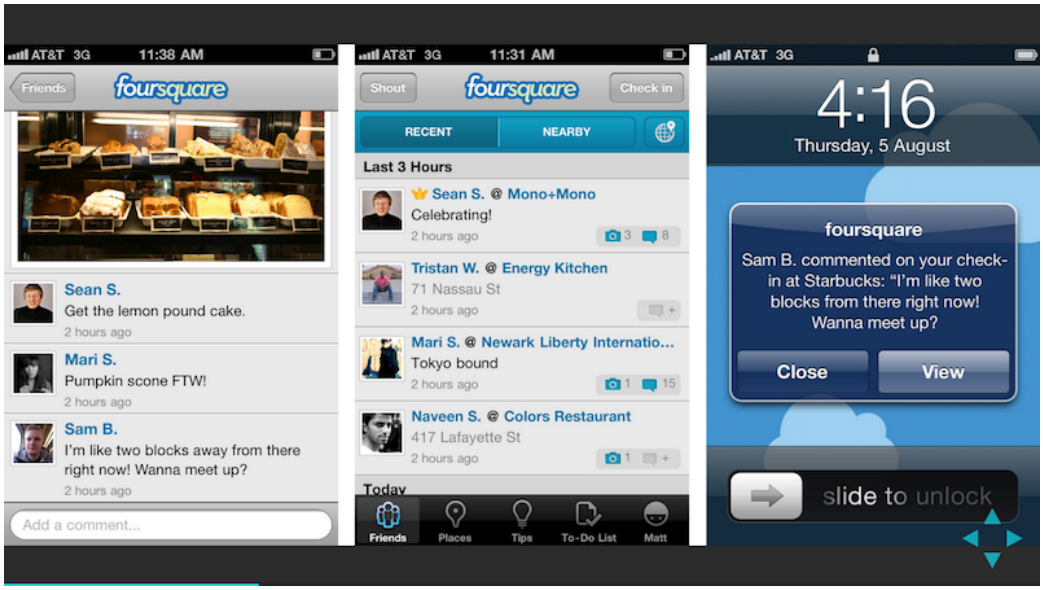

As most of us know, Google Analytics has a realtime option. But is this just a marketing gimmick? What real value does this offer?

When we publish new content it's great to see the reaction that the content is getting, what traffic it's resulting in and where that traffic is coming from. *And that's the real value and opportunity here*.

If you see a lot of traffic coming from a destination, or for a particular keyword, the **opportunity** that realtime gives you is to be able to **react** to that realtime information. You can tweak your content to get better conversions and to take advantage of the unexpected traffic.

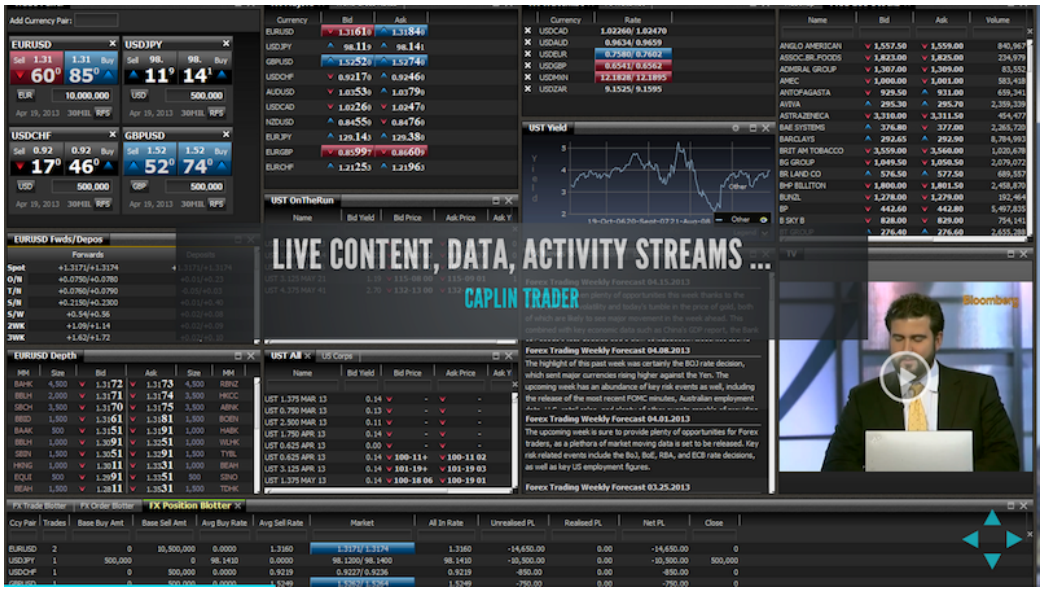
If you see the traffic drop you can **react** by looking at your server stats to check CPU, memory and disk usage.

The speed at which you can deliver information, and the speed at which you receive information, presents both the product or service provider (you) and the user with **opportunity**.




A simple example of the how realtime enables opportunity is one provided by FourSquare. If I check-in to a coffee shop and a friend happens to be near-by they will get a notification. We therefore have the opportunity of catching up over a coffee. If I've never actually met that friend in person before (that happens a lot nowadays) then we get the opportunity to finally meet up in person. Who knows what will come from that meetup?

**I love the serendipitous nature of this.**

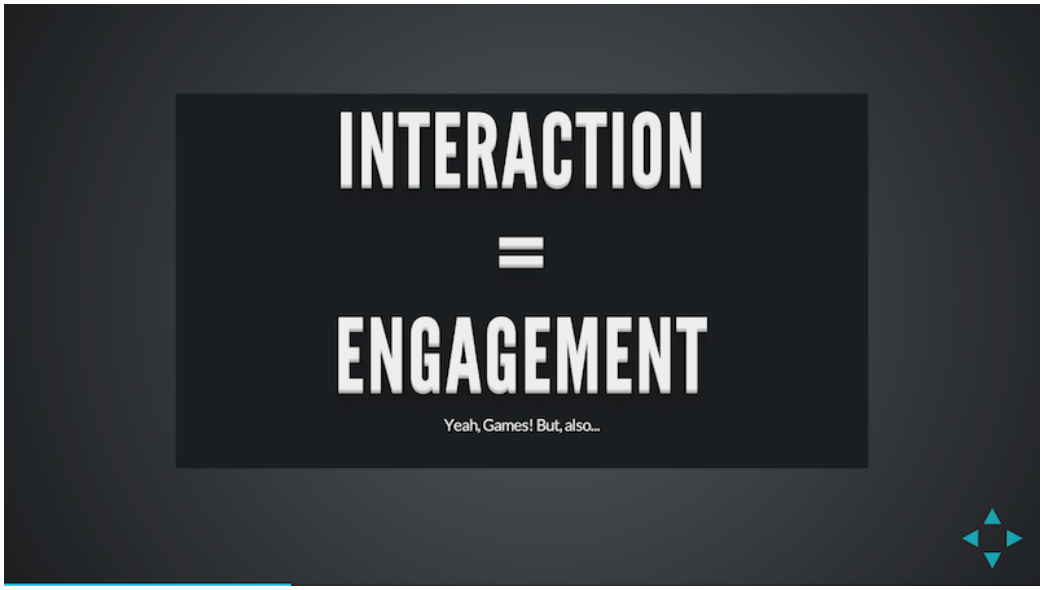



Back in 2001 I joined Caplin Systems as a university graduate. They were one of the pioneers of realtime web technology, focusing on providing realtime web technology to financial organisations.

The thing that makes their flagship front-end products - now Caplin Trader - interesting is how they have evolved. Initially the focus was on enabling realtime in-browser data displays to financial

organisations where realtime data really makes a difference; milliseconds can make a difference between lots of money made or lost. Now the focus is firmly on enabling realtime interactions with trading systems and traders - actually executing trades and realtime communication within a web trading front-end.

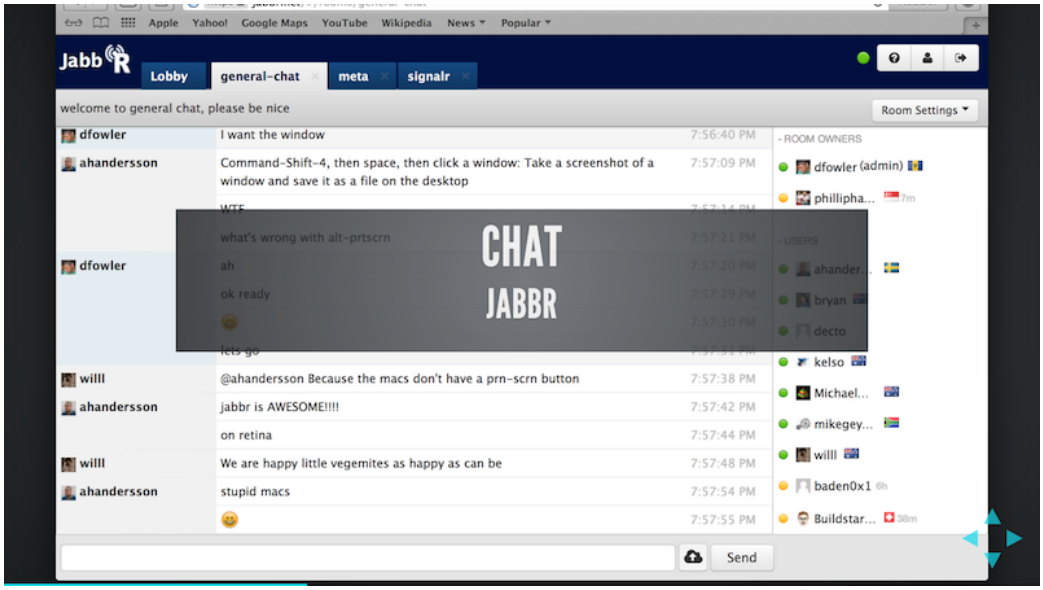
This demonstrates - for me - the most exciting use of the technology right now. **Enabling realtime communication and interaction**.




Twitter, Facebook, Google+ and many other platforms are successful because they are **communication platforms**. When we publish a tweet or a status update we don't do that in the hope that we're ignored. We do so in the hope that our update will be retweeted, favourited, replied-to or liked; that it will start some for of interactive communication experience.

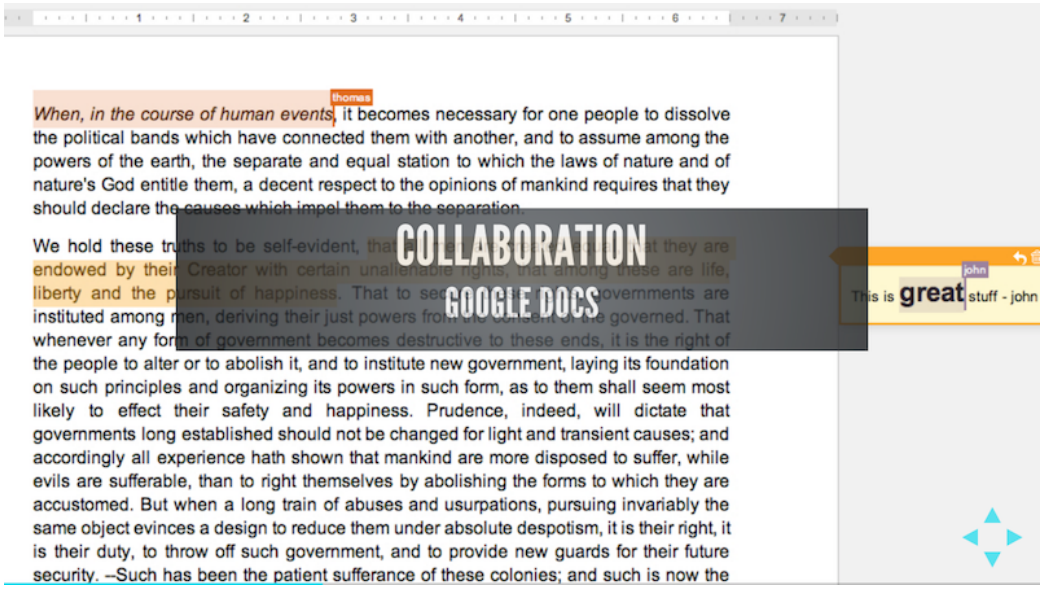
When we interact in an application we engage with it, we spend a lot more time using it. From an application creators point of view this is massively important.

**Realtime Web Technologies enable us to create interactive and engaging functionality within our applications**.




When it comes to trying out realtime web tech you'll probably start by building a chat application. Jabbr is one example.

We probably all use chat applications on a daily basis. So, we understand their value. Realtime web technologies make building chat functionalty really easy so we can integrate similar functionality into any application and start to get the benefit directly in our apps.




**Collaboration** is probably the area of communication where I'm most excited to see this technology used. We have some great examples of uses now, but I think we're going to see a lot of innovation in this space.

Most of use have used Google Docs before. It's a great product and saves us a tonne of time. How did we used to collaborate on docs previously?

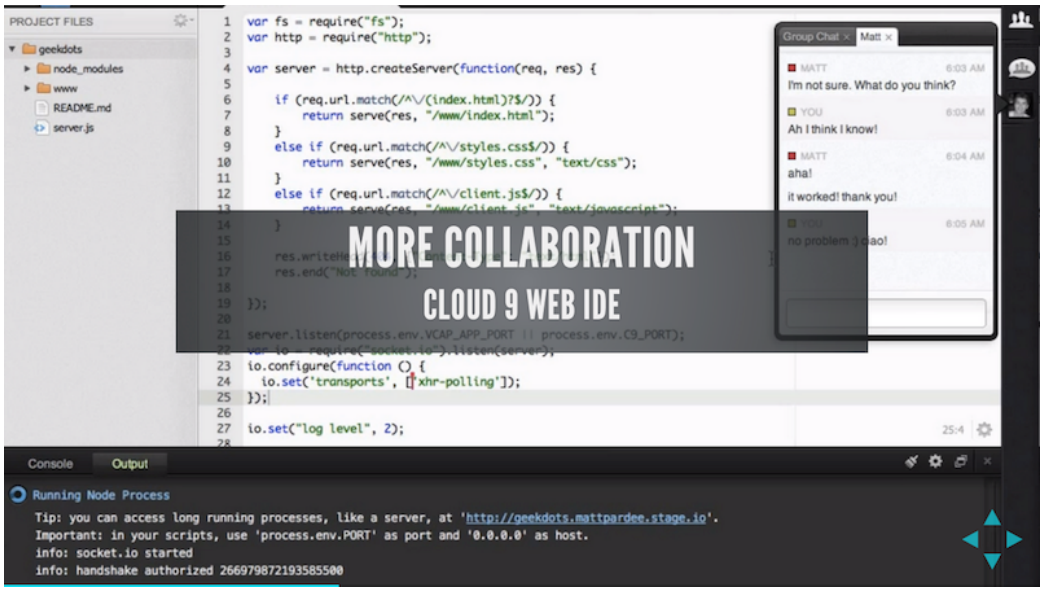
- Write initial document draft
- Add to Share Point (oh dear!) or email it as an attachment to single collaborator

- Collaborator turns on Track Changes and edits doc
- Wait for review to be checked back in to Share Point or emailed
- Make updates based on review and back to 1. again.

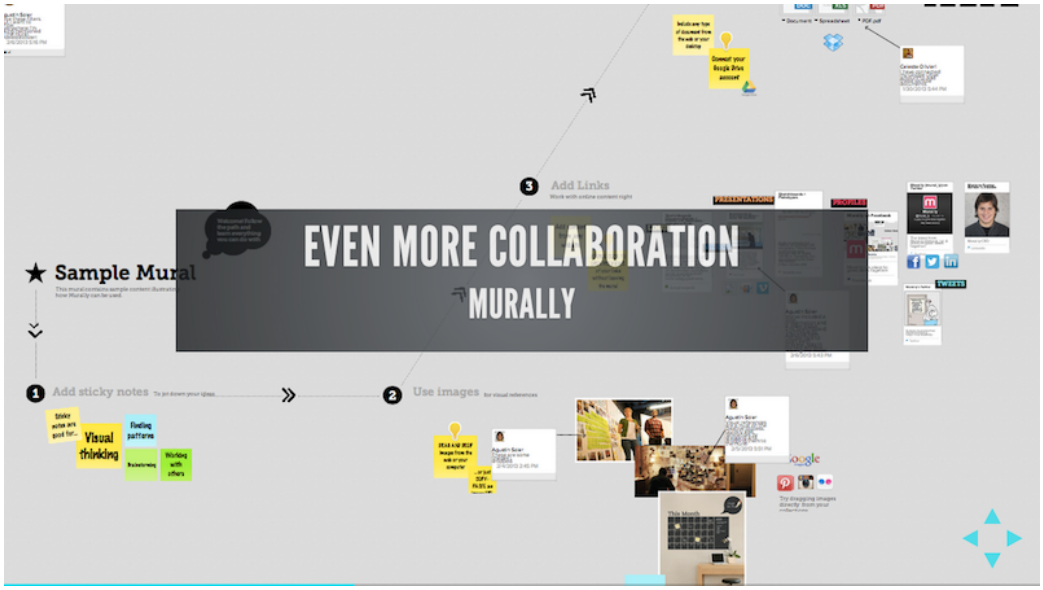
This could eat up a whole lot of time. The process is now:

- Write initial document draft
- Invite collaborators, edit and dicuss in realtime

How many hours or days does this save?

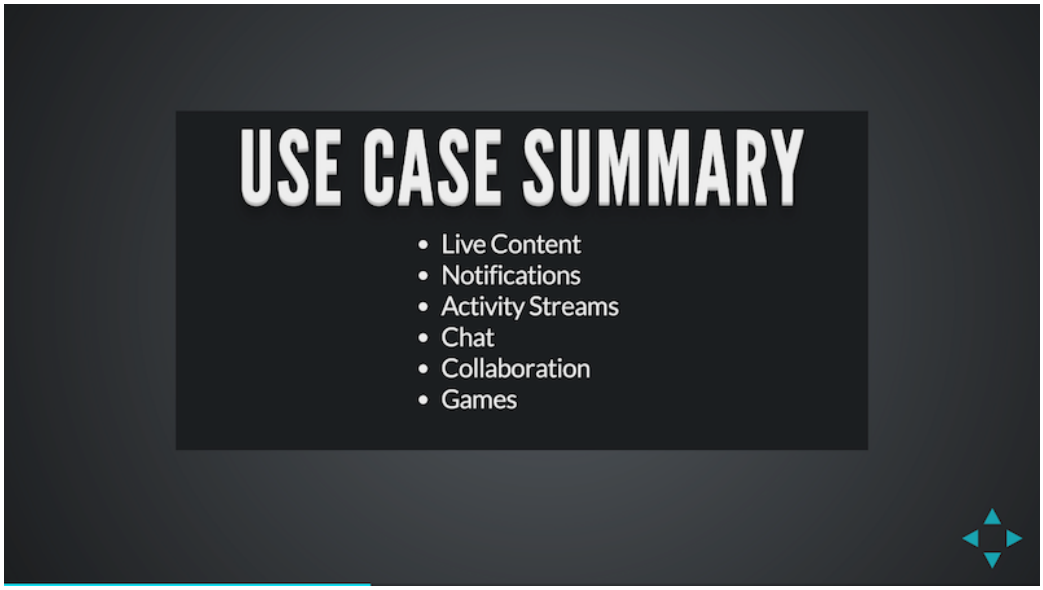



We can take this experience to coding. In this case we have Cloud 9 IDE, an in-browser collaborative code editor. Using C9 we can have multiple developers editing and discussing code, and we can also jump into a **collaborative debugging session**.




Finally, we can take an experience which usually requires physical space - think whiteboarding.

Murally offer an app that lets multiple users collaborate on large open murals where you can bring in text, pictures and annotations. A great way of exploring and fleshing out ideas.




There are many uses of realtime technology but the most common use cases are live content, notifications, activity streams, chat, collaborative experience and games.

As we start to innovate futher I'm sure this list will continue to expand - I didn't even touch on the **Internet of Things where realtime web tech is going to be massively important**.

## Next: Fundamentals of Realtime and Realtime Functionality

That's it for part 1. In part 2 I'll cover the fundamentals of realtime and realtime functionality. Finally, in part 3 I'll cover 5 points that you should consider when **Choosing your Realtime Web App Tech Stack**.

---

GET IN TOUCH

## Social

## Email

phil@leggetter.co.uk