INDIAN INSTITUTE OF TECHNOLOGY MADRAS

सिद्धिर्भवति कर्मजा

# IIT Madras

ONLINE DEGREE

**Modern Application Development – 1**
**Professor Nithin Chandrachoodan**
**Department of Electrical Engineering**
**Indian Institute of Technology, Madras**
**Introduction to the Web**

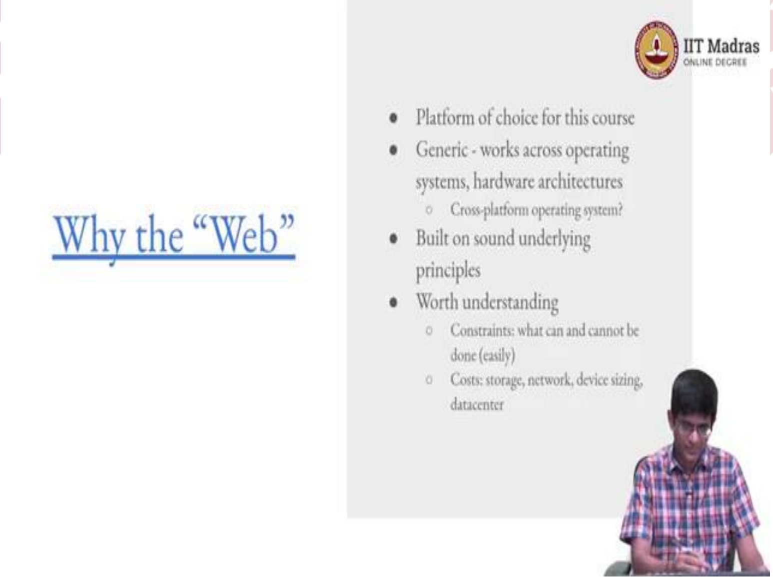
Hello, everyone, and welcome to this course on Modern Application Development.

(Refer Slide Time: 0:16)

The "Web"

Now, let us look at what exactly we mean by the web, because that is something fundamental that we are going to be using as our underlying platform for all the implementations that we will do in this course.

(Refer Slide Time: 0:29)



Why the "Web"

- Platform of choice for this course
- Generic - works across operating systems, hardware architectures
  - Cross-platform operating system?
- Built on sound underlying principles
- Worth understanding
  - Constraints: what can and cannot be done (easily)
  - Costs: storage, network, device sizing, datacenter

So, why the web? What is the web and why the web? So, it is the platform of choice that we are going to use for this course. Why? The main reason is that this is something generic, it works across operating systems, across hardware architectures, we can almost at this point, think of it as a cross platform operating system. So, what does cross platform mean?

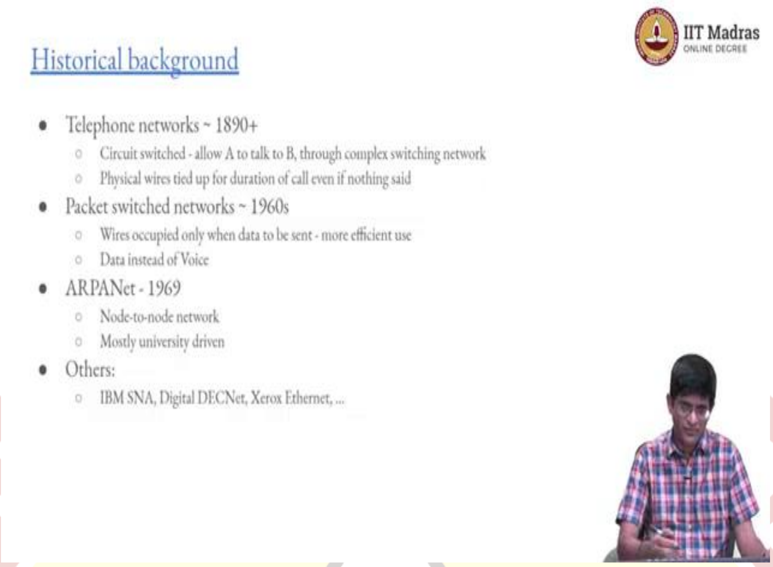
Some of you are probably watching this on a Windows PC. Some of you may be having a Mac, some may be having a Linux based laptop or desktop. Others may be watching it on a tablet, an iPad, or a mobile phone. Now, each of those has its own operating system, it has its own interfaces, various things like does it have a keyboard? Does it have a touchscreen? Does it have a camera? Does it have a tilt sensor?

Now, as far as the web is concerned, it abstracts all that away, and gives us a cross platform, common background that we can use. It is built on sound underlying principles, what are the principles, we will look at them a little bit more moving forward? And in any case, it is worth understanding what it can and cannot do? In particular, what are its constraints? What are we limited from doing that we will not be going into too much detail over here.

But at some point as in when you start developing apps, you will realize that that information is important to you, there are certain things you can do in a web based app, certain things that simply cannot be done, and certain things that can be done with difficulty. The other thing, as far as understanding the web is concerned is that because it is primarily a network oriented system that we are looking at over here, there are certain costs, there is storage, there is networking, there are data centres, where the actual web servers are operating.

And as a result, how do we sort of balance those costs? How do we estimate what the costs are? How do we estimate the different trade-offs involved, all of those are things that we need to consider here.

(Refer Slide Time: 2:30)



Historical background

- Telephone networks ~ 1890+
  - Circuit switched - allow A to talk to B, through complex switching network
  - Physical wires tied up for duration of call even if nothing said
- Packet switched networks ~ 1960s
  - Wires occupied only when data to be sent - more efficient use
  - Data instead of Voice
- ARPANet - 1969
  - Node-to-node network
  - Mostly university driven
- Others:
  - IBM SNA, Digital DECNet, Xerox Ethernet, ...

So, let us start with some historical background. Now, the telephone networks as we know them, or at least in their original form, started in the 1890s. After Alexander Graham Bell invented the telephone, there were already telephone networks in place by the 1890s, even before the twentieth century started. Now, telephone networks, by their very nature are what are called circuit switched networks.

Now, in a circuit switch network, what we mean is, let us say that A wants to talk to B, I ensure that there is a direct set of wires that are forming a closed circuit between A and B. That is, whatever A says will be taken on a wire as some kind of an electrical signal, and transmitted all the way through up to B and B's response will come all the way through back to A. There are physical, you can actually identify the physical wires that will exist in taking the data from A to B.

Now, what that means is those physical wires are also tied up for the duration of the call, meaning that nobody else can use them. So, even if A and B are not really saying very much, there is mostly silence on the call, or, in any case, the bandwidth used by voice is pretty low. The rest of the wire cannot be used by anyone else for any other purpose. This led to an idea called the packet switch network.

And in a packet switched network, people said, look, maybe certain wires. So, for example, the last link, which actually takes the thing from my telephone to the nearest telephone exchange, that has to be tied up to me. I mean, it cannot be going to anyone else, because physically it is a wire that exists between the telephone exchange and myself. I cannot really use it in any other way.

But what about the lines connecting to exchanges together? There is no point in, let us say, there are 100 people on one side and another 100 people on another side, I do not really need to have 100 wires going or maybe 200 wires going between those two places. Maybe I can aggregate the data in one place, and connect all of that, collect all of the data into a single wire and transmit it at a higher speed. The physical costs of the wire reduce, the physical, just the requirements in terms of material also reduces.

The trade-off is that now I need to be able to pack more information per second. And I need to obviously convert the data, I cannot just use voice and directly put it onto the wire and multiplex many of them together onto the same wire. So, this, in turn led to the idea behind packet switched networks.

The idea of a packet switch network is that you could now take the voice data, break it up into small packets of information after digitizing of course, you convert it into a set of numbers, you take those numbers and arrange them into packets. And those packets are now sent across the wire instead of the direct analog signal corresponding to a voice.

What is the advantage? It means that the wire is occupied only when data is being sent, it is more efficient. It also means that a single wire between two exchanges can probably carry several different conversations. Now, one further observation as a result of this is that you are now sending data, you do not even need to think of it as voice that is being transported between these two points.

What that in turn means is that you are no longer limited to voice, you could have think of other kinds of data, maybe you have just a lot of data, a set of numbers that have been collected in some form, a set of alphabets or letters corresponding to some document can also be sent across in the same way. An image, a picture could also be sent across like this.

In other words, now, the network has become neutral to the type of data, it does not care about whether it is voice, video, text, or any other information that you might want to transmit. At the end of the day, everything is a set of bits. So, packet switched networks, essentially lead to much more efficient use of the wires that are available.

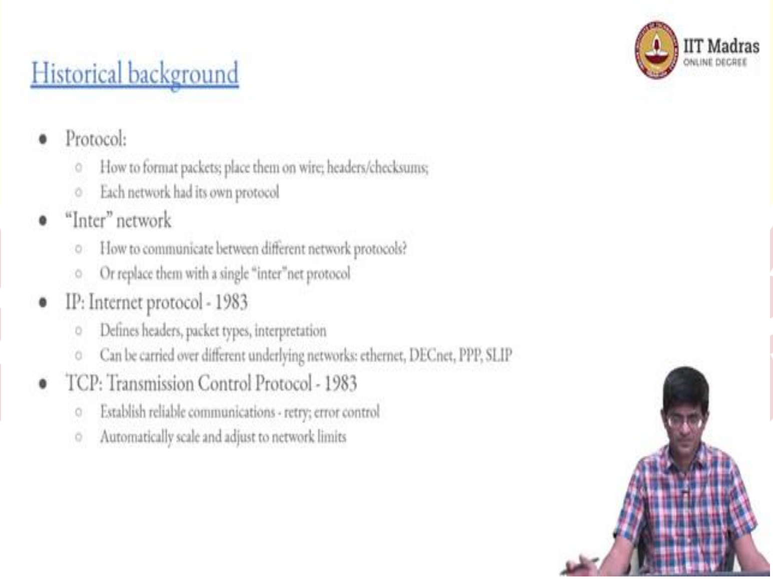
Now, building further on top of packet switched networks, there were a number of different primarily University oriented networks, but some also by several companies, which came up in the late 60s, which had different forms of networking. One of the biggest of those is what is called the ARPANET. ARPA stands for the Advanced Research Projects Agency. It is one of the US government, Department of Defense funding agencies.

And ARPANET was a sort of node to node network, it essentially allowed two nodes to directly communicate with each other. Now, if one node was connected to two other nodes, it mean, it meant that those two other nodes could also relay information through this node to each other.

It was largely University driven, but there were several others. So, for example, IBM, DEC Digital Equipment Corporation, and Xerox, those are all big companies that in fact, still survive to this day in various forms, that were involved in some kind of networking or network related activities around that time.

The problem was, each one of them had their own idea of how the network should work. How should I take the data? How should I put it into packets? How should I put it on the wire? What kind of wires should I use? All of those who have questions that each of these companies and each of these universities solved but in slightly different ways. What did that mean? It meant that they could not really talk to each other, or at least not easily.

(Refer Slide Time: 8:12)



Historical background

- Protocol:
  - How to format packets; place them on wire; headers/checksums;
  - Each network had its own protocol
- "Inter" network
  - How to communicate between different network protocols?
  - Or replace them with a single "inter"net protocol
- IP: Internet protocol - 1983
  - Defines headers, packet types, interpretation
  - Can be carried over different underlying networks: ethernet, DECnet, PPP, SLIP
- TCP: Transmission Control Protocol - 1983
  - Establish reliable communications - retry; error control
  - Automatically scale and adjust to network limits

So, that is where the idea of a protocol comes into the picture. Let us say that you do have IBM and Xerox and you want their computers to be able to talk to each other, you need to decide on some kind of a protocol that both of them are going to follow. And a protocol, although it sounds fancy, is something very simple. All that it says is how should I format packets? How do I place them on the wire? What kind of electrical signals do I use? But more importantly, than electrical signals? What kind of data formatting?

Should I use any kind of checksum something to guard against errors? Should I have some header information, which says this is where the packet is coming from, this is where it is meant to go? All of that information is what determines the protocol being used by the network. Now the problem, of course, was at that time, each network had its own protocol.

And the obvious question that came up next is can we create an inter network, an inter network protocol. And the idea was an inter network protocol would allow, would be some kind of a higher level protocol, which would allow different networks to communicate with each other. So, let us take something like the Ethernet.

Ethernet was developed by Xerox, The Xerox Corporation. And it specifies things including what kind of wires to use, what kind of signalling to use on the wires, the so called physical layer for those of you who have studied a little bit about the communication aspects of it. We do not need to know too much about that, which is why I am not going into details here.

But Ethernet would specify all of that. On top of that, it would also specify how do you say what is the source that is which is the machine that is sending this packet of information? What is the destination which would be which is the machine that is supposed to receive and act on this information. And because it had all of that it could then be used in some kind of a broadcast manner. The information would just be put on the wire, and whichever other machine picked it up would look at the to address; this is meant for me, respond.

Now, Ethernet allows you to do that. But obviously, IBM had different ideas on how to do it, digital Dec net, they had other ideas, some of the universities had their own ideas. The Internet, essentially came about as one protocol, which said, let us put a layer on top of this. I do not care if Company A wants to use Ethernet and Company B wants to use Dec net, or university C wants to use their own protocol, whatever it is.

When they want to talk to each other, they need to take the actual data that they want to transfer and put some new headers on top of it, which we will now standardize. And that became the Internet Protocol. It was an inter network communication protocol. Now, Internet, in that sense, essentially came into existence on Jan first 1983. Why on that particular date, it is not as though, on December 31, of 1982, people suddenly had a brainwave and decided this is how to do things.

The protocol had been in development for many years from the 70s in fact. Many variants, people were discussing it, seeing how it would work. And finally on January first, 1983, they said, this is it, let us standardize on this, because otherwise, we are getting nowhere. You can keep on trying to improve it. But we have to decide and move on.

So, in 1983, two things, were essentially standardized and formed the basis of what we now know as the Internet. One is the Internet Protocol itself, which essentially gives the definitions of headers, packet types, how they should be interpreted. And essentially said that this means that this is how a packet can be created. And how it is actually transported across from one network to another is left to the underlying network architecture.

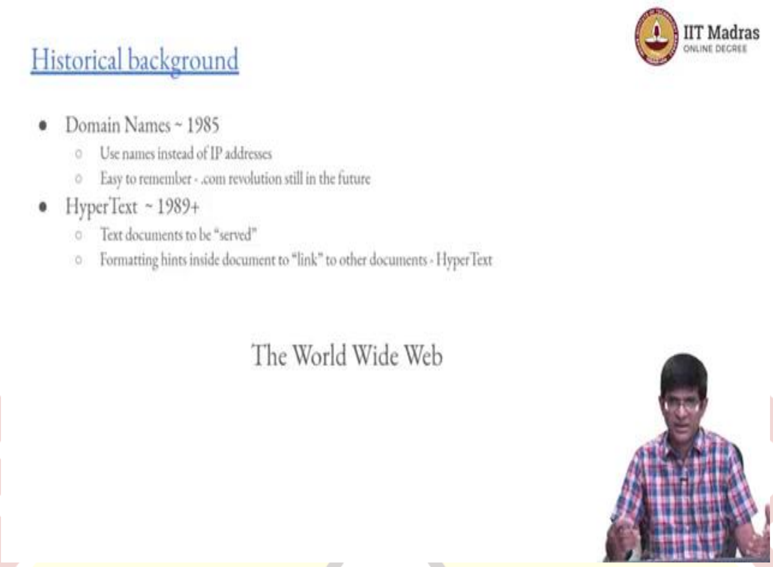
So, let us say you have an Ethernet network, all that you need to do is figure out how to transport these packets of information across it. On top of that, one more layer was what is called the TCP; Transmission Control Protocol. Transmission Control Protocol is a more specialized form of the Internet Protocol, it essentially says it is not enough just to randomly send data, you just send out a packet from A and hope that it eventually reaches B.

What I would like to have is actually a closed loop that is to say send information from A to B, and B response. It actually gives me back an acknowledgment saying that yes, I received your information. What next? So, that allows you to set up some kind of a circuit switch on top of a packet switched network. So, Transmission Control Protocol takes care of things like reliable transmission.

Should you retry the transmission in case you did not get an acknowledgment from the other side? How fast can you transmit information? Can you adapt to the speed of the link, try sending data faster and faster, until you start seeing failures and then maybe back off a little bit until you actually say, this is a good speed for me to transmit at. All of that is handled by TCP, which is why nowadays, even now, you will see TCP slash IP as sort of the backbone of the internet in some sense.

As you can see, it is nearly 40 years old at this point, which means that there are certain deficiencies that have been noted in it over the years, it was designed at a time when network speeds were not as high as they are today. The computing capability of switches, routers, the end-user equipment were much, much lower than what they are today. So, there are deficiencies that have been identified in these protocols, people are working on modifying them, but to a large extent, they still form the core of the internet, as we know it, even now.

(Refer Slide Time: 14:10)


Historical background

- Domain Names ~ 1985
  - Use names instead of IP addresses
  - Easy to remember - .com revolution still in the future
- HyperText ~ 1989+
  - Text documents to be "served"
  - Formatting hints inside document to "link" to other documents - HyperText

The World Wide Web

So, now that TCP IP was in place, it meant that people could talk from one machine to another. But the problem was if I wanted to communicate from one machine to another, at that point, I still needed to know the address of the other machine, and the address was something which is defined as an IP address, the Internet Protocol address, which was a slightly non intuitive thing for easier, it was not easy to remember.

It was basically a set of 4 numbers, what they call the dotted quad notation. There are 4 numbers between the range 0 to 255, which are separated by dots. Now, why that particular structure? It is basically you take a 32 bit value, you break it up into 4 octets, that is 4 bytes 4 8 bit bytes. And each of those bytes, you then read it out as a decimal number and you separate them with dots, you could just as well give the 32 bit value and say this is the address, but that becomes even more difficult to remember.

So, for a long time, people use this kind of dotted quad notation. In fact, we still do. So, there are, there may be places where you come across an IP address. The problem is IP addresses are not easy to remember and they are not very easy to type in either. They are not very long, which is good. But bottom line is that you cannot really remember a large number of IP addresses.

If you think about it, even the phone numbers of your close friends, you probably remember a few tens of them. But definitely, you cannot remember hundreds or thousands, it is much easier to remember names. So, the Domain Name System came into existence somewhere around 1985.

And what domain names did is that they set up a hierarchical structure where there was some root servers, and then sub domain servers below that, which would allow you to essentially, let us say you had a name like apple.com, it would allow you to find out ".com". This is one machine that can tell me what are the names of all ".com" systems, apple.com, it would translate that name apple.com into an IP address.

You do not need to do this personally, there were computer programs that would do it for you, which meant that as far as the human beings were concerned, humans could now use names. Apple dot com is much easier to remember than an IP address, which meant and not only that the IP address could be changed quietly at the back end. So, you could switch to a new machine. And the humans would not know the difference. Apple dot com would still take you to something which hopefully was being hosted by the apple company.

So, the domain names made it possible to communicate using names instead of numbers, and also abstracted away so that you are no longer tied to individual servers, individual machines. And you could then sort of switch machines or have multiple machines at the back end, all responding to the same name.

And with all of this in place, so now you have networks, you have an inter network, you have domain names and the concept of hypertext is what was developed starting around 1989 or so. The idea itself had been around for longer than that. What exactly is hypertext and the core functionality behind it, but it was brought into one consolidated framework, which could actually be used based on work by Tim Berners Lee at CRN, the European Centre for Nuclear Research.

So, this is actually a research center, which is high energy physics, they have almost nothing to do with computers, except that if you look at it, even now, CERN does some really cutting edge development of high end computers. The reason being that, the kind of measurements that they are doing over require extremely precise computers and extremely precise measuring equipment.

So, there is a lot of interesting electronics development that has come out of physics research. And this, the whole idea of hypertext and its use in general applications came out from there. So, what exactly is hypertext? In general text is a set of alphabets. ABCD, anything, any sort of file of information, that you have on a computer can be interpreted as some kind of text.

Now, the text by itself is just raw data. It does not tell you what to do with it, it just is a pile of information. But you can sort of embed hints about how the text should be formatted inside the text itself. And we will be looking a little bit at this later, in one of the later videos. So, what exactly does it mean to embed formatting information?

What you are saying is that some codes or some pieces of information inside the text, if you have a suitable program, that program would read through the text, identify these hints, these codes, and say, this part of the text needs to be bold. This part of the text corresponds to a section heading, this part of the text corresponds to a hyperlink. A hyperlink, essentially was simply something which said, now, if you look at this, it is actually a code which tells you that the immediately following text corresponds to another file.

This is the name of another file that you need to load, that is all it is. It is all just text. But the way that you interpret it means that you now know that I need to go add another file and display that at that point. So, this business of linking to other documents essentially is what is called hypertext, which led to the creation of what was called the World Wide Web. Now, we rarely refer to it as the World Wide Web anymore. But we still use the term the web.

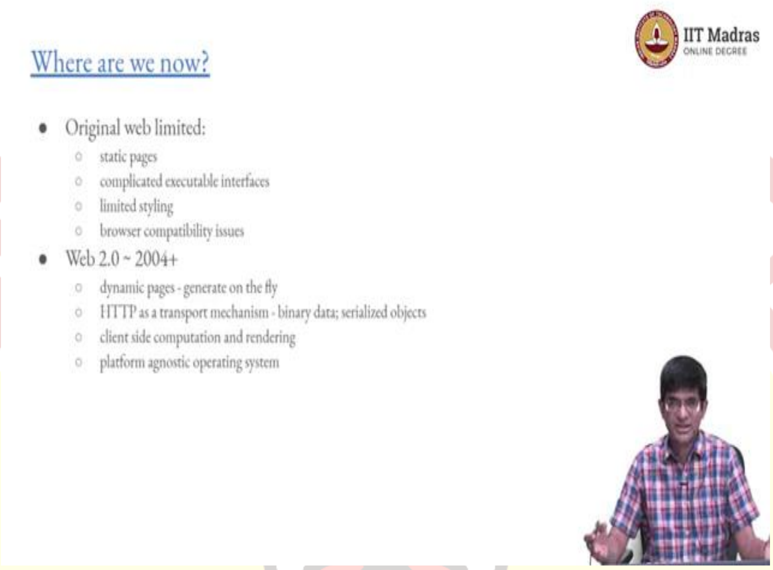
And this essentially is exactly it. Why is it called the web? Because it has this mental image of many different servers that are all sitting around talking to each other and there are these clients who are coming in from outside and pointing at those servers and retrieving information from them. And if you look at it from a distance, you see that there are all these wires criss-crossing each other, it starts to look like a spider web, so as good a time as any, use the term web to indicate that this is what you are talking about. And the name stuck.

So, now when we refer to the term, the web, this is pretty much what we mean, there is a huge network of servers all around the world, each of which has its own documents that it can serve. If somebody asks for a document, it will give you that document. What do the documents have? They basically have text. But with embedded formatting information, some of that formatting information is just telling you how to display the link, how to display the page.

Some parts should be highlighted, some parts should be in bold text, some should be on a bigger font size because they are section headings, something should be in italics, because you want to emphasize something. And there are other parts that are hyperlinks, which basically connect to other documents. This connection between the documents is ultimately what is the sort of the hallmark, the main feature of the web.

So, more than the network of servers that are connected to each other, you now have this notion of documents that are connected to each other. And that is where we call this the web rather than the net, the internet, which is just the network of servers that are connected to each other.

(Refer Slide Time: 22:00)



IIT Madras
ONLINE DEGREE

Where are we now?

- Original web limited:
  - static pages
  - complicated executable interfaces
  - limited styling
  - browser compatibility issues
- Web 2.0 ~ 2004+
  - dynamic pages - generate on the fly
  - HTTP as a transport mechanism - binary data; serialized objects
  - client side computation and rendering
  - platform agnostic operating system

So, where are we now? We are now in 2021, the original web started about 30 years ago, 1989-1990, it was fairly limited, it basically had this notion of static web pages, just the text files, which could be served by an internet server. Now, this meant that it, you could have some setup by which you would be able to execute an arbitrary program on the server and retrieve information corresponding to that, but it was not straightforward.

And therefore, the executable interfaces, how you can execute information on the server end and get dynamic information something which depends on the user or something which the user is asking for, was not very straightforward. There was something called CGI. I mean, that still is but the CGI was pretty much only way the Common Gateway Interface was the only way of interfacing with the network servers.

It was somewhat complicated in the sense that it did not make it easy to do some kinds of dynamic page creation. There were fairly severe limitations on the kind of styling that could be used. And because of the fact that, pretty much everyone was just creating their own browser, whoever felt like it would create a browser, there were a lot of browser compatibility issues around that time.

So, in fact, pretty much the hallmark of the 90s was the so called browser wars. Microsoft had their own browser. There was Mozilla and well, before Mozilla there was, before Firefox, there was Mozilla, which in turn came from something called Netscape Navigator, which in turn came from NCSA mosaic. So, there were a whole sort of series of browsers that were building up around then and there was always this controversy about which one is the right way to do it.

Well, where are we now? Since around 2004 or so people started using the term web 2.0. And this was actually based on a non-trivial observation, it was a fairly good observation, the fact that by generating dynamic pages, that is web pages that would be generated on the fly, you could now take this HTTP, the Hypertext Transport Protocol as just a transport mechanism. You do not need to worry about whether you are actually transferring text. It need not be hypertext, it could be any kind of data. It could be an image, it could be video, it could be a form with information. It could be files in a PDF format, or it could be just plain text as well.

But the point is now HTTP has become a transport mechanism rather than a hypertext transport protocol, a general transport mechanism. This means that a number of changes were made to the protocol itself, that made it somewhat easier for servers to generate pages dynamically which meant that now I could have something where depending the server could actually look at information about me where I am connecting from, what kind of browser I have. And based on that, it could change what it sends back to me.

And it also meant that the server could then keep track of who I am, and maybe allow me to shop by basically maintaining some kind of a shopping cart, or some kind of tracking my information and tracking what kind of interaction I have had with the server in the past. A lot of this also allowed, the browser's also started becoming more powerful around that time, simply because, the processor is the machines that people had were also more powerful. So, when a browser is more powerful, it means that you can now offload some of the work there.

Things like JavaScript came into the picture, which essentially ran on the browser side, and said I can take care of some of the work I can beauty up the page a little bit more, I can do a lot more interaction on the page than was possible previously. And the user interaction becomes much more fluid and much nicer. But fundamentally, what has happened is this dynamic generation of web pages means that the nature of the web has changed.

Servers are no longer just simply serving files, they are actually computing and generating information that is specific to you. This is what in some sense, eventually leads us to be able to talk of this as some kind of a platform agnostic operating system. Platform agnostic, because I do not really care whether it is a Windows PC, or a Linux desktop, or a Mac, or a tablet, which did not even exist in 2004, at least not the way that we know it now. So, effectively, this almost became an operating system.

Once you know that there is a browser which will run on all of these systems and behave pretty much the same way, I can now target that browser and write my code and write my applications for that browser and the server that at the other end, without worrying about what kind of machine you are actually working on.