Mod-07 Lec-41 Introduction to Cloud Computing

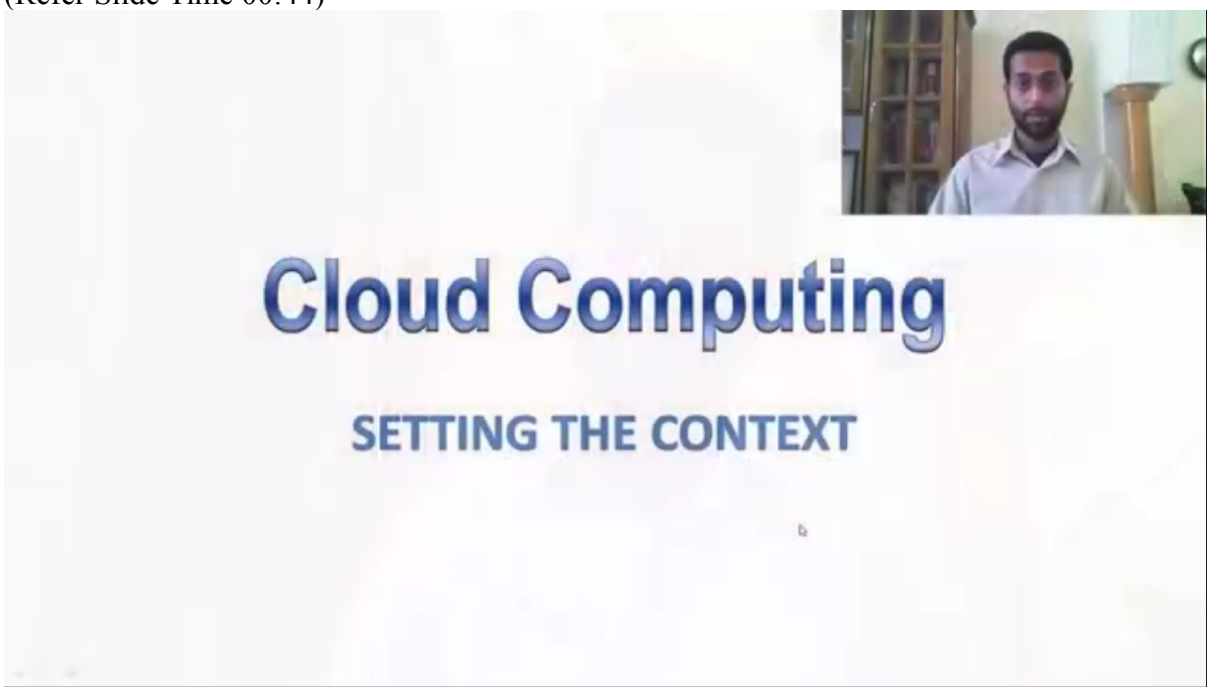
Balwinder Sodhi
Department of Computer Science and Engineering
IIT Ropar

Welcome back. My name is Balwinder and I am associated with IIT Ropar.

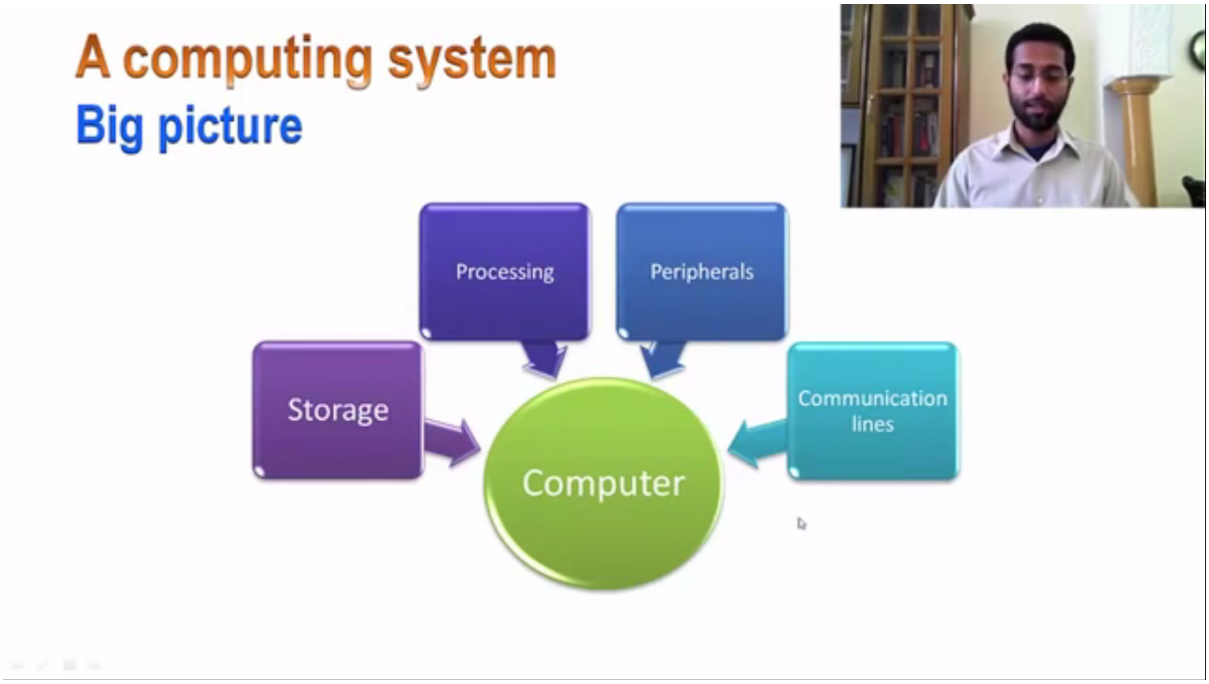
In this lecture, I will introduce you to the concepts of cloud computing. I will start by introducing you to what is the structure of a regular computer, what are the main components of a computer and then I will introduce you to a general structure of an application, software application, which runs on a computer.

Now building upon these two concepts, I will slowly introduce how cloud computing has changed this landscape of computing infrastructure as well as application development.

(Refer Slide Time 00:44)



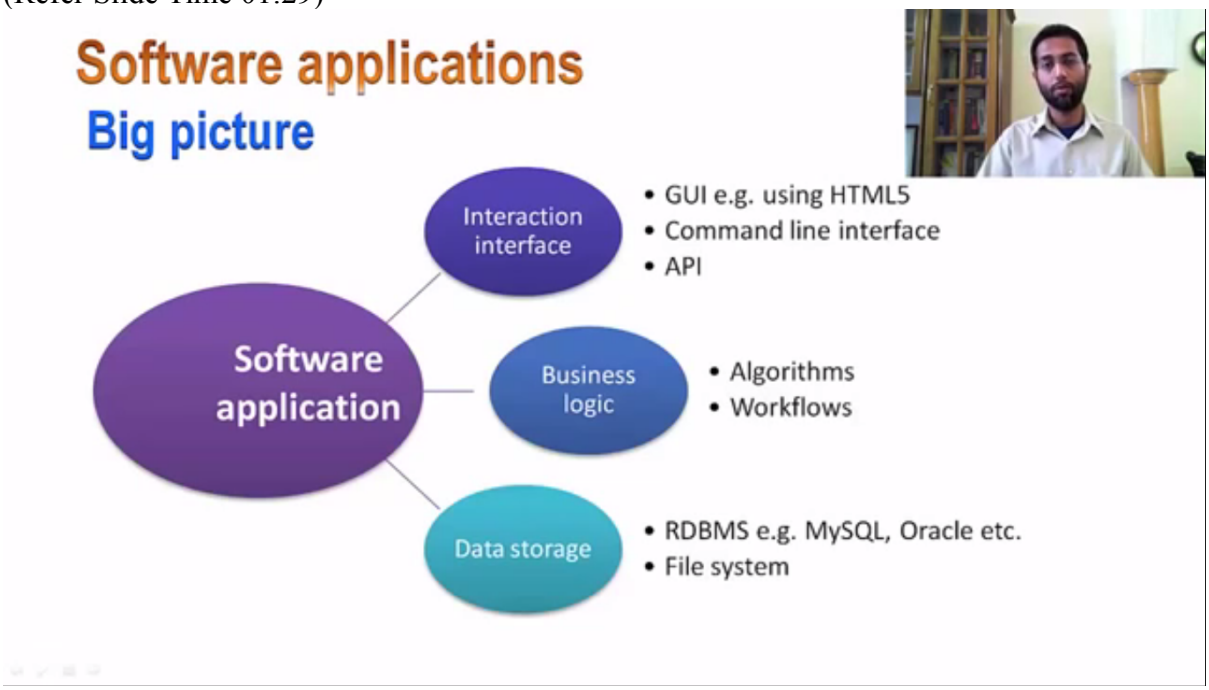
(Refer Slide Time 00:45)



Let's start by looking at general structure of a computer. A computer consists of four major elements, major components. One is storage. Another is a processing element or the CPU. You may have peripherals such as a camera, a keyboard or a monitor, and then it may have a communication interface, which allows this computer to be connected to a network, for example, the Internet.

Now these four components are what makes most of the computers. Different computers may have slightly less or more components, but generally this is how the hardware of a typical computer looks like.

(Refer Slide Time 01:29)



Now if you want to make use of that hardware, the raw computer, you have to put some useful applications, install some useful applications on top of that raw machine. Most data driven applications consist of these three elements, which I have highlighted here. One is interaction interface. Another is called business logic and the third one is known as data storage element.

Now the interaction interface is important because through interaction interface only a user can access the application or make use of the application. For example, if you are using an online banking application, the web interface through which you log on to your online banking application and perform various tasks such as doing some money transfer or checking your balance etc., so that interface is essentially what I'm referring to as interaction interface.

The interaction interface could be realized in different ways depending upon what functionality the application is exposing to its users. The example that I gave of online banking or online shopping etc., you may build the interaction interface via something known as graphical user interface, which could be constructed using a variety of technologies such as HTML5 or Java Swing or what you may have.

Another way to interact with the application is through something known as command line interfaces. Alternatively, if the application is offering some sort of a service for programmatic consumption, then that interface is typically called a application programming interface.

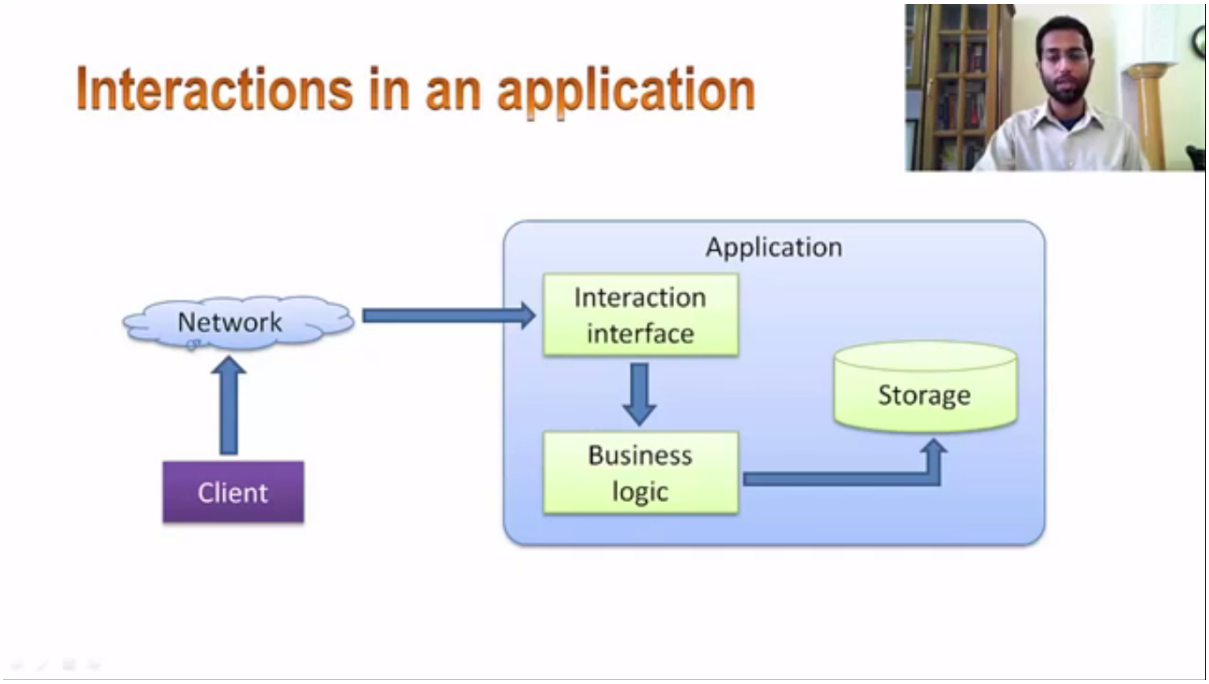
Business logic is the set of algorithmic steps, which application implements in some programming language in order to perform its tasks whatever the application is designed to perform. So those tasks are expressed in a well-defined ordered sequence of steps and they are expressed in some sort of a programming language. So that is what consists of the business logic component.

And data storage, any information that the application is processing while performing its tasks that have to be persisted or stored for later retrieval in some sort of a durable storage. So that could be realized by using some relational database management system such as MySQL or Oracle etc., or raw file system itself.

For example, the banking, online banking case that we talked about, the information about the customer and all the transactions that a banking customer has performed, they may be required to be persisted for a long duration of time for later retrieval or for generating various reports by the bank or for the consumer itself, for the user itself. So for those kind of needs, the applications, typically the data-driven applications make use of something called a data storage element.

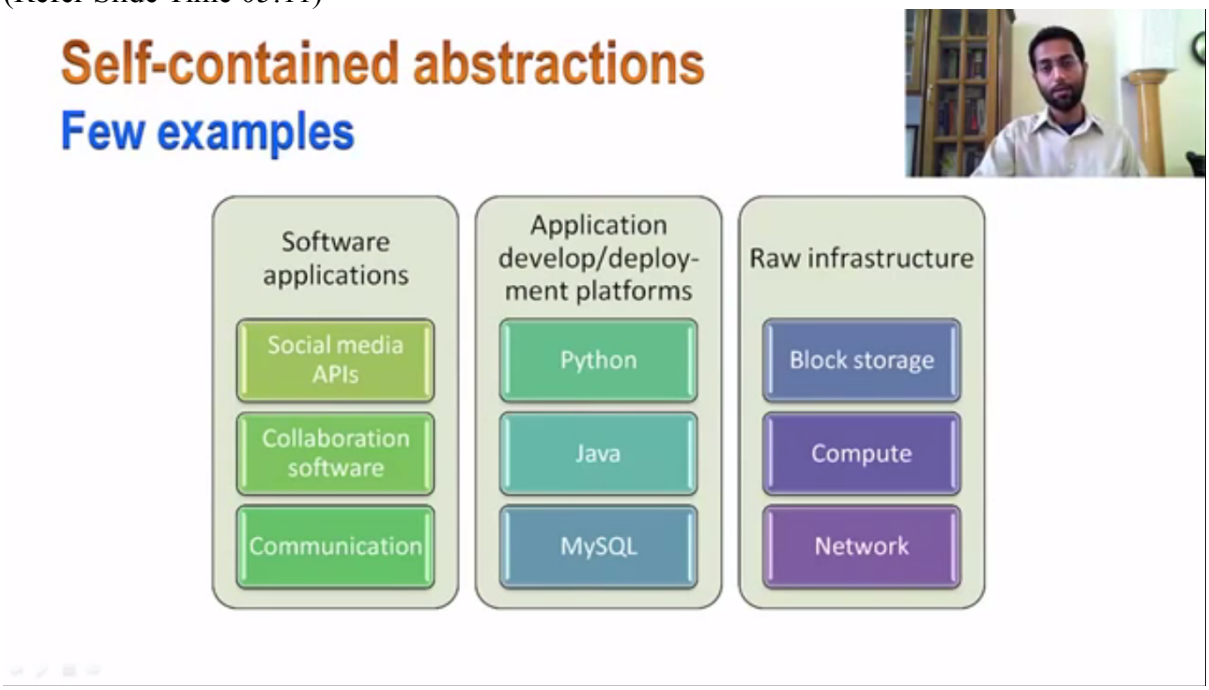
So roughly this is how a typical application looks like.

(Refer Slide Time 04:46)



Now if you have another view of, if you take a look at another view of the application, we have seen that an application consists of these three major elements: the interaction interface, the business logic and the storage. A client typically, for example, a browser, through a browser a user may connect to the interaction interface and perform various steps.

(Refer Slide Time 05:11)



So if you observe this landscape of raw computer hardware and the application, typical data driven application, you will observe that there are these three major abstractions which one can identify.

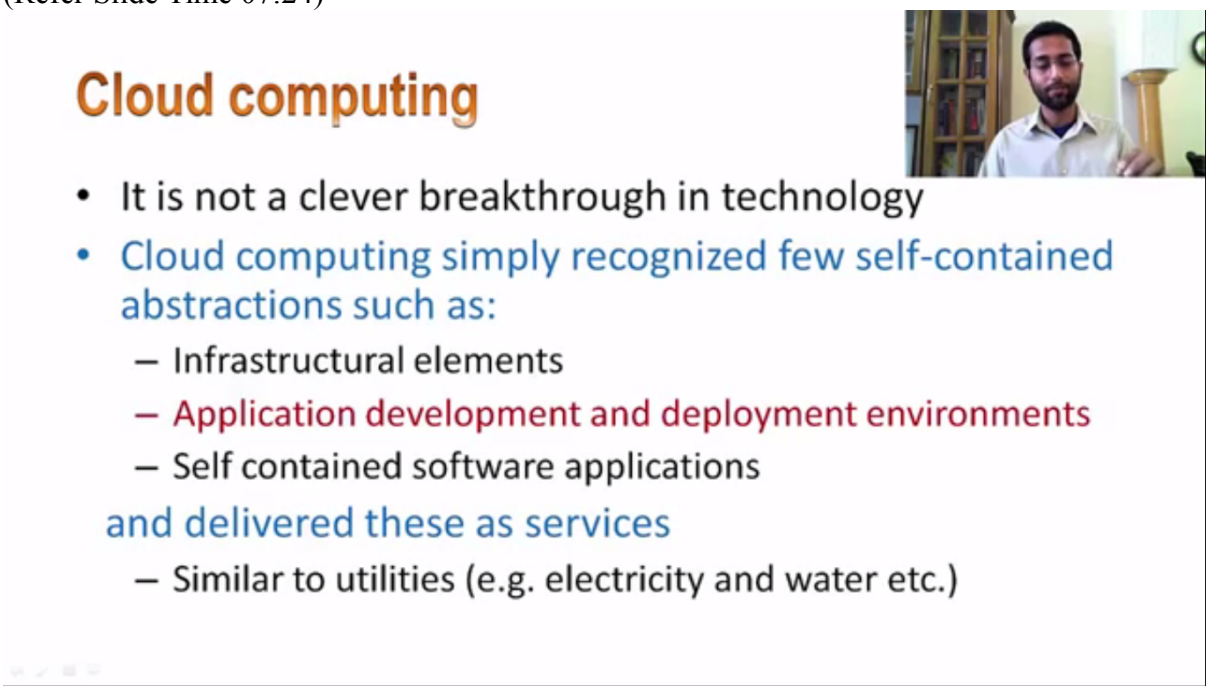
For example, you may have, you know, something like block storage, compute and network, these raw hardware elements clubbed under something that we call raw infrastructure so that can be treated as one abstraction.

And another one is something that application developer makes use of to create applications and deploy them for use. So this is generally looked at as a separate abstraction, which I am terming as application development or deployment platforms. So it could consist of various programming language runtime such as Python or Java or it may have other specialized services such as data storage using something like MySQL or others. So this is giving me a platform to develop and deploy the applications.

And then software application itself can be treated as an abstraction. It is an abstraction for the end user because end users want to perform certain useful tasks using that software. They don't worry about the internals of how the application has been implemented and realized, what programming language has been used, what computer configuration it requires to run etc., etc. The user is typically interested in just making use of the software application.

An example of this scenario could be let us say Gmail or Facebook. So as an end user, I just use the software application. I don't bother about looking at internals of how those things have been implemented. So in that sense, I can look at software itself, the application or the service itself as an abstraction. So these are the three major abstractions that I have talked about: raw infrastructure, application development and deployment platforms and the applications themselves.

(Refer Slide Time 07:24)



So what cloud computing does is it realizes the existence of these three abstractions that I talked about. That is raw infrastructure, application development and deployment stacks or platforms and the applications themselves. And they are giving these individual abstractions

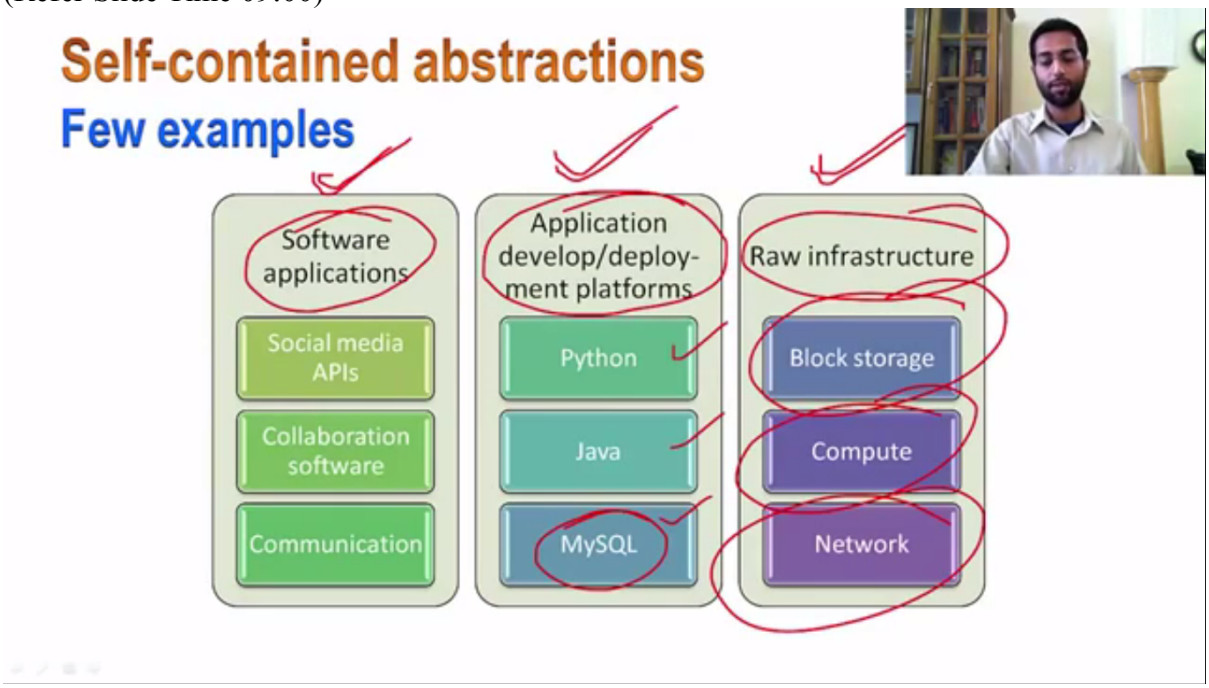
as services, selling them or providing them as services to the consumers who are interested in making use of those abstractions.

So you may have infrastructural elements given to you as a service or you may have application development and deployment platforms given to you as a service or self-contained software itself could be given as a service.

So pretty much like a utility such as electricity and water that we subscribe to for a regular recurring fee rather than setting up the entire electricity generation plant and distribution system on our own. We simply subscribe to the service.

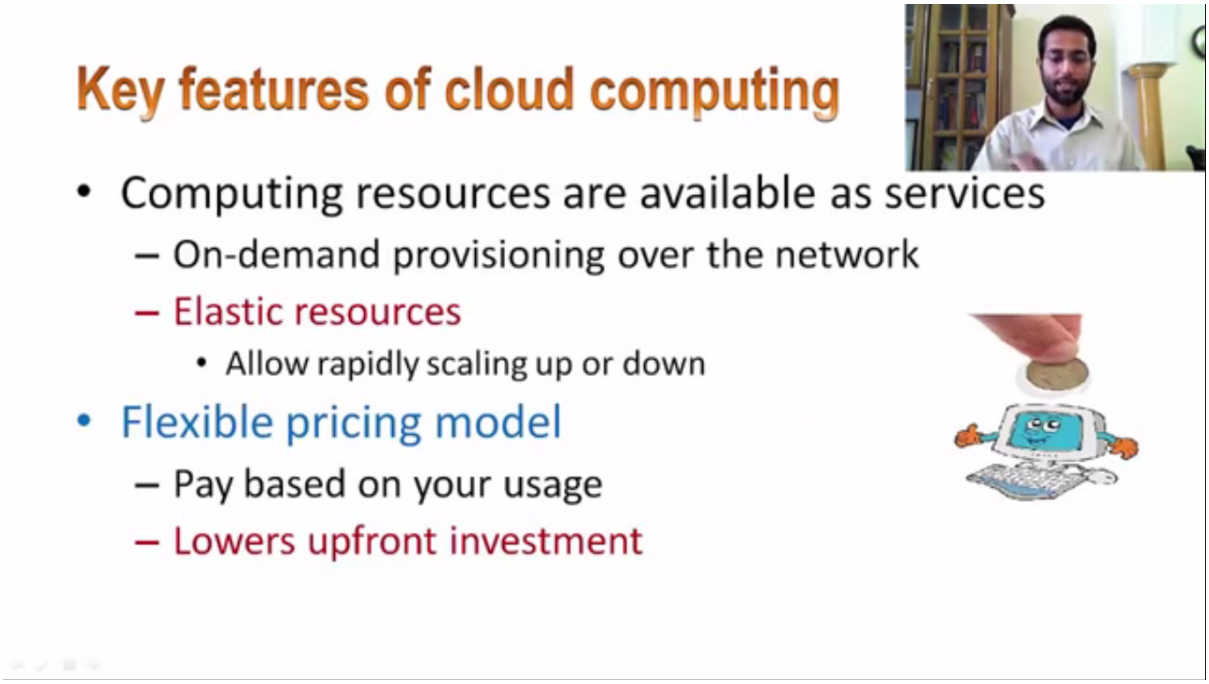
So in a similar manner we subscribe to a specific service from a cloud vendor. So they provide us something that we call infrastructure or something that we call platform or the software itself. So it's not cloud is not necessarily a clever technology breakthrough. It's just a model to deliver computing abstractions as services and these abstractions I have highlighted here.

(Refer Slide Time 09:00)



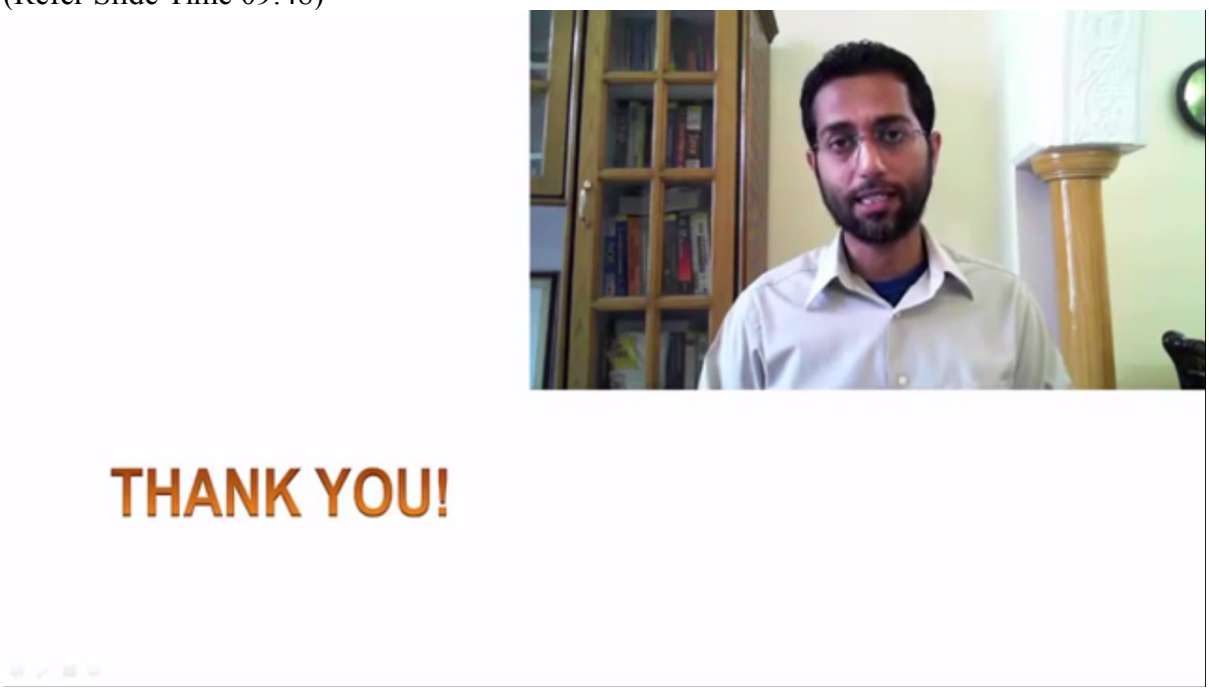
They could be called the infrastructure, application platforms and the software applications themselves.

(Refer Slide Time 09:11)



So cloud computing has certain specific features. So, for example, computing resources are available on-demand as services and the resources are elastic. That is you may add or relinquish the resources depending upon the need that you may have at hand. Pricing model is flexible. You don't have to do upfront investment in procuring the hardware and necessary software to make use of it. So you pay based on your usage. So that lowers the upfront investment.

(Refer Slide Time 09:48)



So what we have seen in this lecture is we started with looking at a high level structure of a typical computer, what it consists of. Then we looked at how an application, a data driven application looks like in its structure and by looking at these two things, we were able to recognize certain self-contained abstractions such as raw infrastructure, and then application

development and deployment platforms and then applications themselves. Then we said that cloud computing simply recognizes these key abstractions and offers these abstractions as services to the end-users.

So that's pretty much it for the introduction into cloud computing. Thank you.