AUDIO	VISUAL
	[TITLE CARD] CLIENT-SERVER ARCHITECTURE
You probably use the internet every day, whether on a computer or a smartphone or other device.	Woman at computer desk using smartphone [over blue background]
Have you ever wondered where that information comes from, and how it gets to you?	Question Mark appears above person
Most of the time it's through a behind-the-scenes technical infrastructure called CLIENT-SERVER ARCHITECTURE It's the way computers communicate with each other. This video will give you a basic overview of this Client-Server Architecture and how it works.	CLIENT-SERVER ARCHITECTURE appears below person
As you can see, there are two ends to the Architecture. At one side you have the CLIENT. The end-user. In our example, it's this guy, browsing the internet on a computer.	Left side - GUY is using laptop on couch - word CLIENT below him On the right side, a tall CPU SERVER.
When our guy types in a web address on a browser, he is sending a REQUEST for data. That REQUEST data is sent to that site's SERVER.	01100101 ICON with REQUEST appears between server and client, along with DOTTED ARROW pointing toward server.
The Server computer receives the request and returns RESPONSE data.	01100101 ICON with RESPONSE DATA appears between server and client, along with DOTTED ARROW pointing toward client.
In our case, a website featuring pictures of a cute cat.	[meow sound] Our guy smiles

When you hit the submit button, the request data from a CLIENT is sent through a web-form, which is kind of like the envelope that carries your request.	Close on CLIENT "Web Form" printed on an envelope
And the response data sent back comes in a number of ways, like a web page, file, or data from a data base. Or the server can pass the request on to another server.	Close on SERVER Arrow to a webpage, arrow to database, to file, to another server.
But how do these clients and servers know how to talk to each other and where to send the right requests and responses?	Lots of CLIENTS and SERVERS with arrows going every which way
Every device has a unique identifier, called a "Media Access Control" card. This is a piece of hardware in the device with a unique identifying number.	Phone, tablet, computer, printer appear in a row "Media Access Control" appears above them Each device gets a MAC tag (e.g., 03:FF:11:72:88:04)
Every time we use the Internet a connection is created between our device and the Internet Service Provider (ISP) The ISP assigns an Internet Protocol (IP) Number to us so the server can locate us. IP Numbers may look like this.	Tablet computer, printer disappear, leaving phone. ISP icon appears, wiggly connection goes from Tablet to ISP. "IP: 134.140.12.61."
Most of the time, we use the human language equivalent of this IP number to communicate with a server. That's what we type into the address bar on a web browser. In this example, it's the IP# to simmons.edu	IP # above address bar. Pointing to one another Close up of Browser Address Bar
Just like an envelope needs a stamp to be sent	In the address bar:

thru the mail, our request envelope needs a "digital stamp" to know HOW to send the data. This is called a PROTOCOL There are a number of types of PROTOCOLS. The most common one for web pages is http Hyper-Text Transport Protocol. Increasingly, websites are using a secure version of "http" called HTTPS.	http://web.simmons.edu Arrow points to new part introduced w/ name. May Protocols show: HTTP, HTTPS, FTP, SFTP, FILE, Hyper Text Transport Protocol
By breaking down this web address, we can tell who's site we're going to visit, what page we're going to see, and what folder or folders that page resides in.	
We see this in the HOSTNAME SECOND LEVEL DOMAIN, and the TOP-LEVEL DOMAIN.	web.simmons.edu = web (hostname or leaf domain); simmons (second-level domain) and edu as the top-level domain) [check slide to highlight correctly
This little tilda is an alias, or shortcut to a folder called jones. The alias helps the web server find the correct webpage to display on the screen.	http://web.simmons.edu/~jones/index.html
It can definitely get more complex than that, but this gives you a basic overview of the Client-Server architecture and how it works.	