
Day 12

This is CS50 for MBAs. Harvard Business School. Spring 2015.

Cheng Gong

Table of Contents

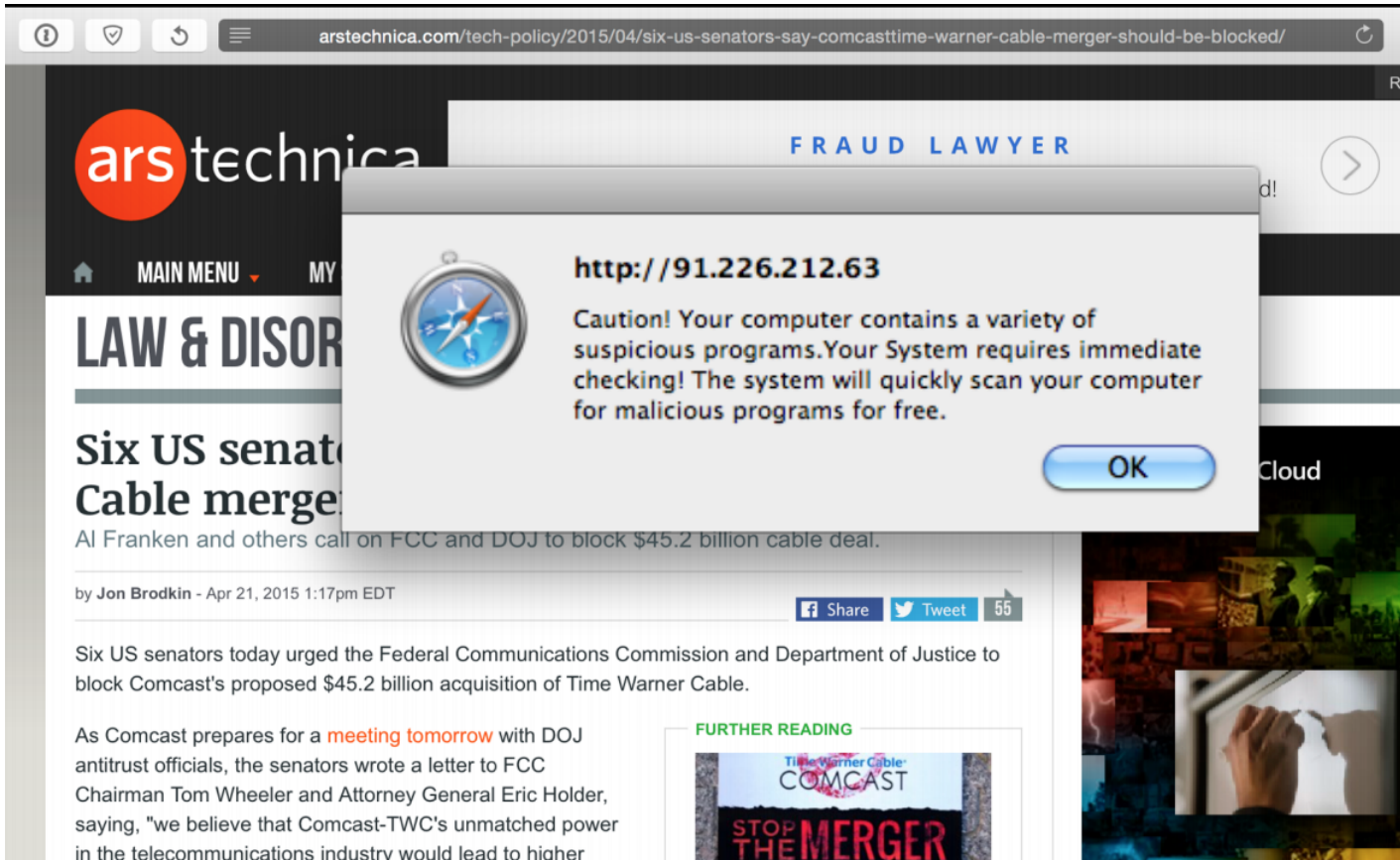
Agenda	1
Advertising	1
Abstraction	9

Agenda

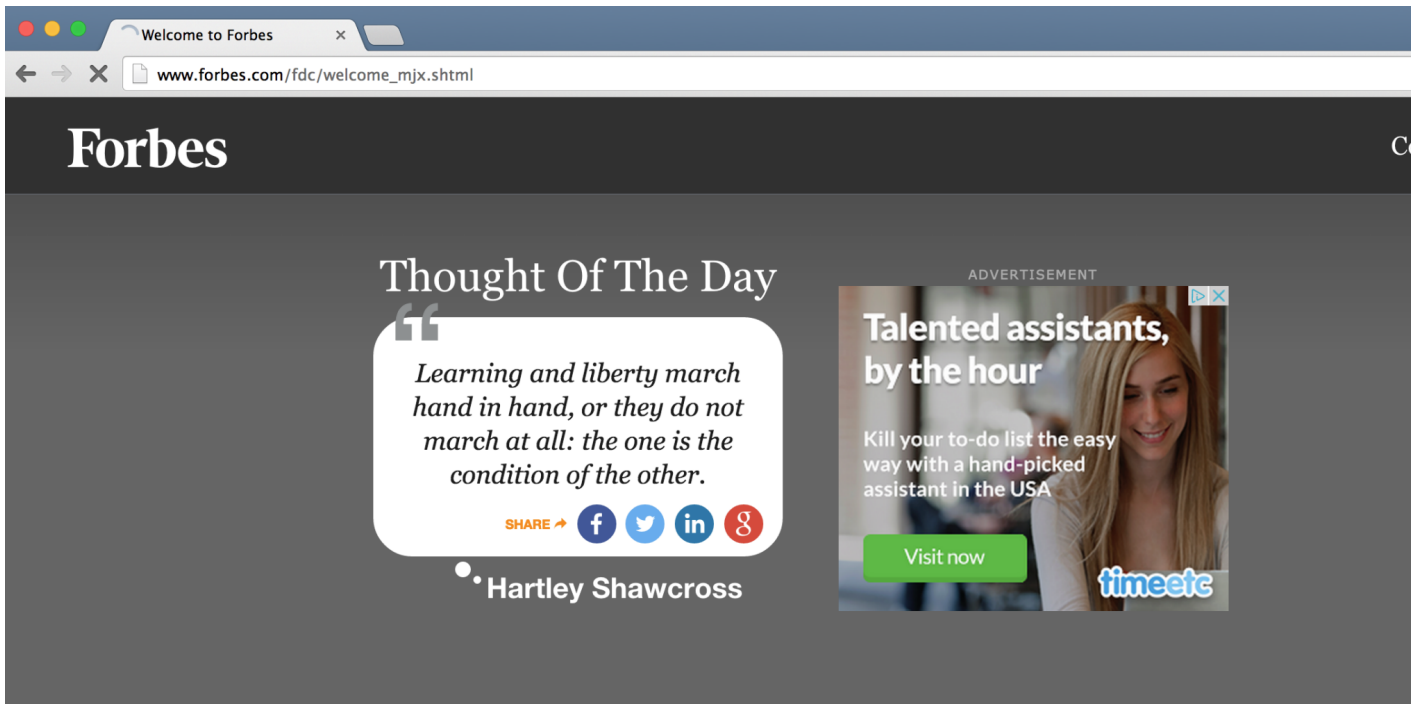
- Today Dan Armendariz, preceptor of computer science at the School of Engineering and Applied Sciences, will be talking about advertising and abstraction.

Advertising

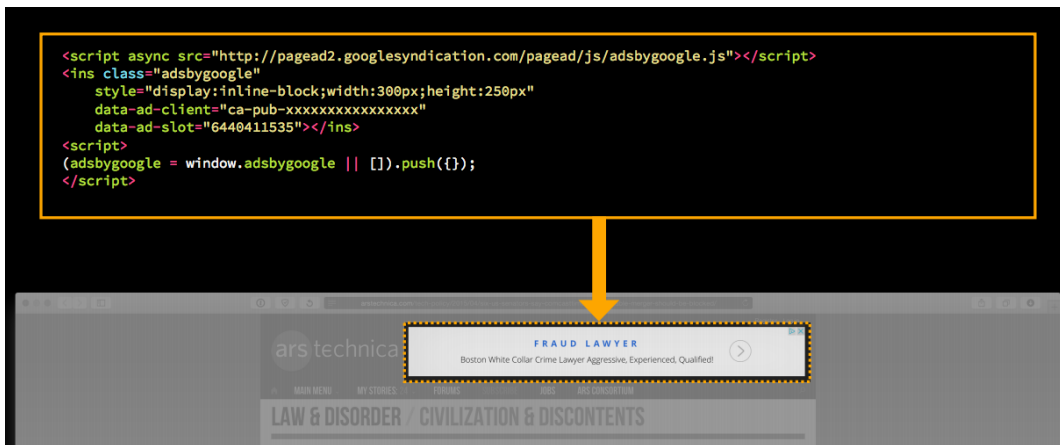
- Nowadays, we see advertisements on not just websites but also mobile apps.
- There are various types. On a website, we might see banner images, text ads, or videos on a page. The page might also trigger pop-ups, pop-unders, or alerts that might look like this:



- Another kind of ad is called interstitial:



- The technology behind some of these ads might look as simple as this:



This snippet of HTML code comes from Google’s ad platform, AdSense. The first line seems to include a script from Google’s servers, and then creates an `ins` element (similar to a `div`) on the page that will contain the banner.

This allows for the ad loaded to be dynamic and targeted, based on whatever Google thinks will be most effective. And a user who has cookies for one website can be identified only by that website in future visits, but users who visit multiple different websites, all of which embed this advertising script from Google, can be tracked across all of them.

Having the script be on the client-side also allows for the website providing content to respond more quickly, and have the ads load afterwards, rather than wait for Google to reply with the ads and send everything to the user all at once. But additional connections take time too, especially for mobile users on slower connections.

- Apple’s equivalent for iOS apps is called iAd:

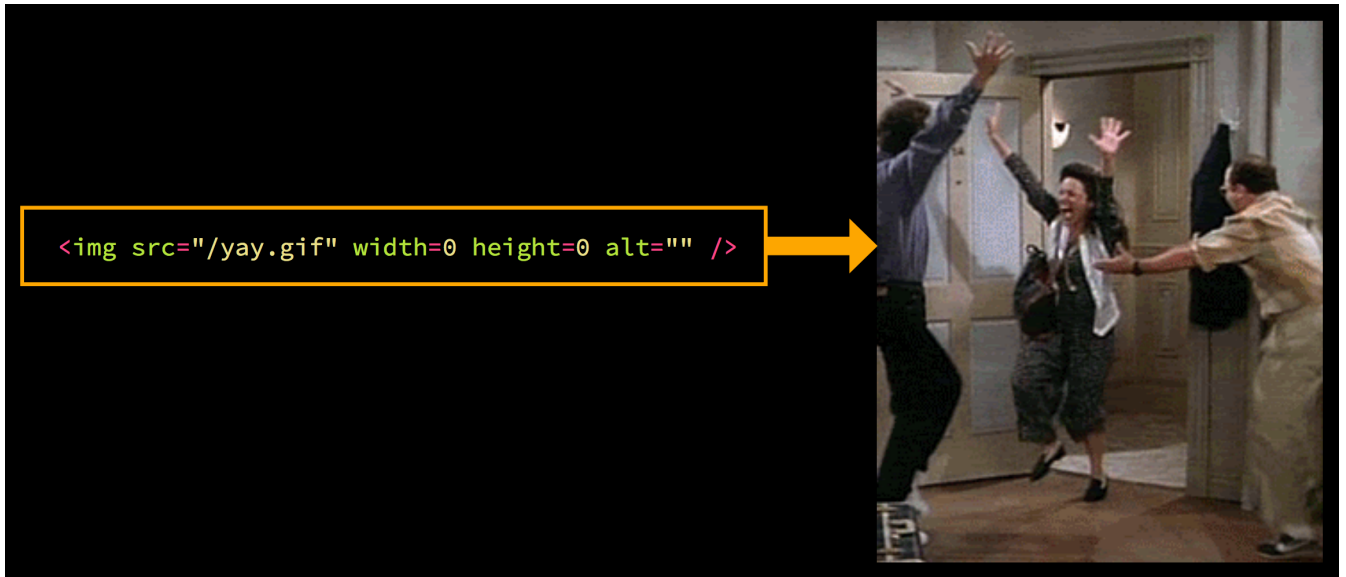


- # This too, only takes a line of code to allow apps to embed ads that Apple manages.
- # But the rest of the content will be resized, in order to fit the ad. And apps that work offline might now need to connect to the internet to receive ads.
- To combat fraud of websites receiving payment for illegitimate ad views or clicks, Google has some measures for validating whether users actually see or click on the ad, and having client-side JavaScript embedded from Google helps with implementing these anti-fraud techniques. With that said, there is also arguably a very important aspect of trust, both between websites that run Google's ads, and companies that pay Google to run ads, since it provides more long-term value for everyone involved.
- Going back to the content of the ads:

The screenshot shows a web browser displaying an article on Ars Technica. The article title is "Senators say Comcast/Time Warner merger should be blocked". Two ads are highlighted with red dashed boxes:

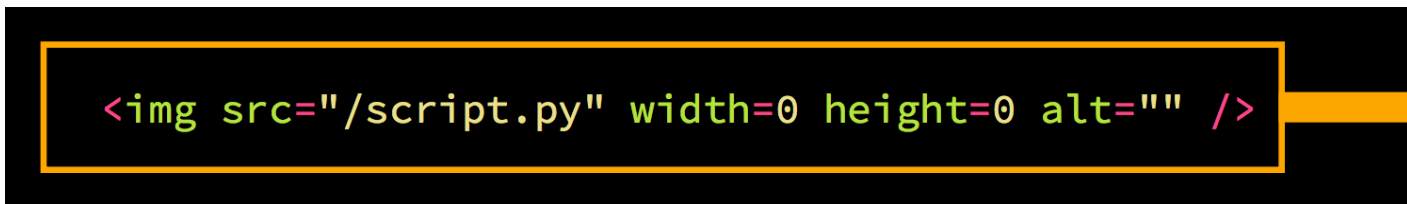
- Top Ad:** A white rectangular ad with a red dashed border. It features the text "FRAUD LAWYER" in blue, followed by "Boston White Collar Crime Lawyer Aggressive, Experienced, Qualified!". There is a right-pointing arrow icon and a close button in the top right corner.
- Right Ad:** A vertical rectangular ad with a red dashed border. It features the Microsoft Cloud logo at the top. The background is a collage of various images, including a hand pointing at a screen, a globe, and abstract digital patterns.

- # The one at the top seems to be advertising a fraud lawyer, particularly one in Boston. So it seems to have narrowed the visitor (in this case Dan) to likely be in Boston. And in the same browsing session, Dan was doing research on the latest information about click fraud for today's discussion, so the ad might have been linked to those searches too.
- # The one on the right has fewer details, but advertises Microsoft Cloud, a technology product that Dan might be interested in because the site underneath, Ars Technica, is a technology journalism site.
- We can go a little deeper into the technology for these targeted ads. The following code seems to add an image to a webpage:



When an HTML page with that code is loaded, the browser sends an additional request asking for that image.

- But what happens when the request isn't for a GIF file, but something else?

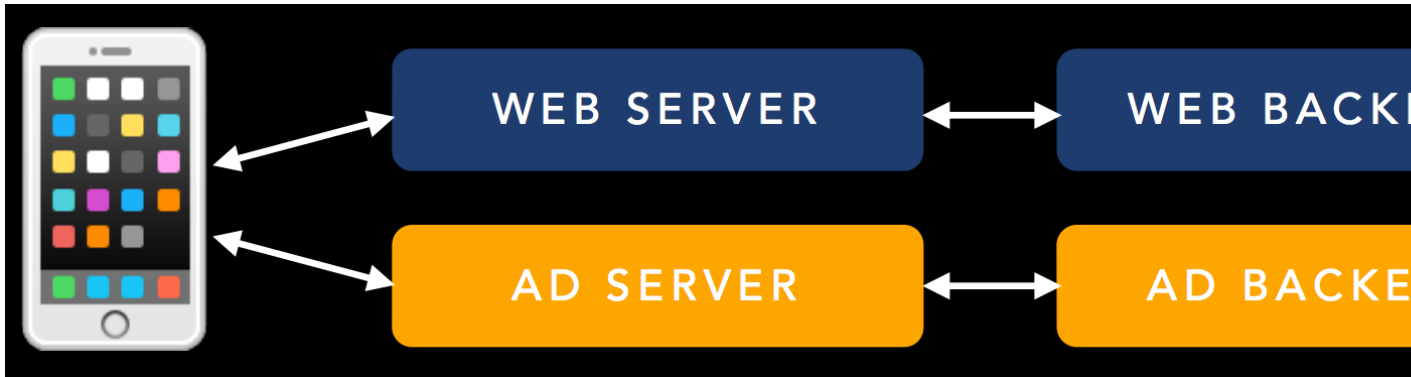


`.py` is the file extension for Python code, so when the browser makes that request, the server is probably running some Python script that returns an image that the browser can embed.

But the script can also not return anything, and instead simply log the browser request, with data such as the user's IP address, and from that track the user. For example, the user might be visiting `arstechnica.com`, but the HTML on that page has an image tag linking to something like `googlesyndication.com`, and now Google has information about the user, even with JavaScript disabled.

And the `width` and `height` of the image has been set to 0, so the browser won't even display anything on the page.

- So in general our devices communicate as follows:



- Certainly, there are privacy implications, and without going into deeper discussion of those, we realize that many users install ad blockers:

arstechnica.com/tech-policy/2015/04/six-us-senators-say-comcasttime-warner-cable-merger-should-be-blocked/

ars technica

MAIN MENU MY STORIES: 1 FORUMS SUBSCRIBE JOBS ARS CONSORTIUM

LAW & DISORDER / CIVILIZATION & DISCONTENTS

Six US senators say Comcast/Time Warner Cable merger should be blocked

Al Franken and others call on FCC and DOJ to block \$45.2 billion cable deal.

by Jon Brodtkin - Apr 21, 2015 1:17pm EDT 64

Six US senators today urged the Federal Communications Commission and Department of Justice to block Comcast's proposed \$45.2 billion acquisition of Time Warner Cable.

As Comcast prepares for a **meeting tomorrow** with DOJ antitrust officials, the senators wrote a letter to FCC Chairman Tom Wheeler and Attorney General Eric Holder, saying, "we believe that Comcast-TWC's unmatched power in the telecommunications industry would lead to higher prices, fewer choices, and poorer quality services for Americans—inhibiting US consumers' ability to fully benefit from modern technologies and American businesses' capacity to innovate and compete on a global scale."

The letter was signed by Sens. Al Franken (D-Minn.), Bernard Sanders (I-Vt.), Edward Markey (D-Mass.), Ron Wyden (D-Ore.), Elizabeth Warren (D-Mass.), and Richard Blumenthal (D-Conn.).

As we **wrote yesterday**, "DOJ antitrust lawyers are **reportedly** close to recommending that the government block the deal. Comcast might be able to save the merger by agreeing to conditions, but

FURTHER READING

ONE YEAR LATER, COMCAST'S MEGAMERGER FACES UNKNOWN FATE, DUBIOUS PUBLIC

\$45 billion Time Warner Cable buy has taken longer than Comcast expected.

LATEST FEATURE STORY

FEATURE STORY (2 PAGES)

Tech vs. terror: Drones data fight a new battle against poachers

Tech once developed to protect troops IEDs now guards endangered species

WATCH ARS VIDEO

- The page is much cleaner, and under the hood many of the tracking requests are now blocked, and the compromise is that websites lose some revenue, a debate we will also defer from going into.
- Technically, even if tracking requests are blocked, users can still be identified by so-called fingerprinting.
- <https://panopticlick.eff.org/> is a service that demonstrates this technique by using JavaScript to detect certain features of your browser or computer, and since a combination of many variables tends to be unique, this technique is surprisingly (and perhaps scarily) effective.

Abstraction

- We can define the notion of **abstraction** in a single sentence as separating the implementation of a system from its usage, through the use of an **interface**.
- This definition is a little abstract in itself, but we can provide some examples. A car, for instance, has an interface of a steering wheel, gas pedal, and brake pedal, which the driver uses. Under the hood (literally), the car might have a gas, diesel, electric, hybrid, or other type of engine (the implementation), but the driver doesn't need to know about these details in order to operate it. Instead, drivers can learn a single set of controls to operate many kinds of cars. An electrical outlet, too, has a standardized plug for all devices, regardless of what device is on the other side or where the power is generated from. A piano, too, has standard white and black keys in a particular layout that a performer might know, but these days it can be implemented traditionally with physical strings, electronically with physical keys and speakers, or even completely digitally on a screen. And finally, a program written in Python is a high level of abstraction when the developer doesn't need to know all the implementation details of the hardware, like how the electrons or bits are moved around in a CPU.
- Another benefit of abstraction is the ability to parallelize tasks. For example, if an interface is agreed upon, one engineer can work on the implementation of that interface, while another might work on another aspect of the project that uses the interface.
- And another benefit is being able to easily transfer programs from one implementation to another, if the higher level details share the same interface. A Python program, for example, runs the same on Windows and Mac computers, because the language was implemented for both systems using the same interface.
- The downsides to this include bugs or security issues that affect many programs at once. Recently, for instance, a bug in a popular networking library for iOS apps meant that all apps that used that library were susceptible to a security weakness.
- Another concept is the abstraction **barrier**, which is the level where the interface is. In the case of the car, breaking the abstraction barrier would be trying to steer the car through manipulating the steering rods normally hidden from the user. We can see how this would affect operation of the car, since the implementation of the car might expect the steering wheel to match the alignment of the wheels, or compatibility issues if a driver learning to drive in this strange way were to need to drive a car implemented with different mechanics.

- Breaking the abstraction barrier in programs might lead to unexpected behavior or compatibility or portability issues as well.
- Deciding on the level of abstraction is also important, since having too much abstraction might make something **over-engineered**.
- For example, one decision might be, "Should we run our product in the cloud?"
- Using the cloud would create another layer of abstraction, where the relevant interface would need to be learned or adopted, but the implementation detail of running the physical servers would be taken away. Running our own servers would mean that there is no level of abstraction, and we could control all the details of our back-end. And it's unclear which would be more costly, but generally maintaining a lower level of implementation details is more costly. So there are advantages and disadvantages to both choices, though one or the other might arguably be better for certain situations.
- Making the wrong choice, unfortunately, might mean that engineers have to spend more time on implementation details that could have been abstracted away, by another company or service or software library, or lose the ability to control some aspect of the implementation detail that might need to be changed.
- All functions in a program, too, contain a level of abstraction. Some functions, called higher-order functions, can actually take in other functions as arguments.
- For example, a function seen in [3.py](#)¹ called `apply_to_all` takes in some function, a list of items, and applies that function to all the items in the list by taking each item, applying the function to it, and building a new list of the modified items.
- A function like this is generally called `map`².
- A similar one is called `reduce`³, or fold, which takes in a list of items, applies some function that takes in two of the items at a time, until a single answer remains. [An example](#)⁴ would be the addition function.
- These functions can actually solve many problems in data processing. Optimizing map and reduce functions, too, can result in high efficiency since lists can be split among many servers.

¹ <http://cdn.cs50.net/2015/mba/classes/12/src12/3.py>

² https://en.wikipedia.org/wiki/Map_%28higher-order_function%29

³ https://en.wikipedia.org/wiki/Fold_%28higher-order_function%29

⁴ https://en.wikipedia.org/wiki/Fold_%28higher-order_function%29#Folds_on_lists

- [Hadoop](#)⁵ is one such framework that facilitates this parallelization.
- A real-world example can be seen in `mapreduce.py`⁶ which creates linguistic statistics on some text.

⁵ https://en.wikipedia.org/wiki/Apache_Hadoop

⁶ <http://cdn.cs50.net/2015/mba/classes/12/src12/mapreduce.py>