6.1800 Spring 2024

Lecture #2: Naming

plus a case-study on DNS

Katrina LaCurts | lacurts@mit.edu | 6.1800 2024

6.1800 in the news

TECHNOLOGY

Internet Outage That Crashed Dozens Of Websites Caused By Software Update

Updated July 22, 2021 · 2:22 PM ET

DEEPA SHIVARAM

A widespread internet outage caused several major websites to shut down Thursday afternoon, including Amazon, Delta, Capital One and Costco.

Akamai, a content distribution network that helps with the spread of data around the internet, posted on Twitter that a software configuration update caused a bug in its DNS system.

A DNS, or domain name service, helps match a website's name to its IP address. If the DNS fails, it becomes impossible to search and connect to a website by name.

source: `https://www.npr.org/2021/07/22/1019333663/internet-outage-dns`

6.1800 in the news

This article was published on: 08/30/22

Home / Featured / Bad Ubuntu update crashes global Azure Kubernetes services

Featured

Bad Ubuntu update crashes global Azure Kubernetes services

Image courtesy Jeff Geerling.

A flawed Ubuntu systemd update appears to have taken Azure virtual machines running on Ubuntu offline by breaking DNS – causing a significant Azure Kubernetes outage for Ubuntu users.

source: `https://thestack.technology/azure-kubernetes-outage-ubuntu-dns/`

6.1800 in the news

1/25 **Post Incident Review (PIR) – Azure Networking – Global WAN issues (Tracking ID VSG1-B90)**

What happened?

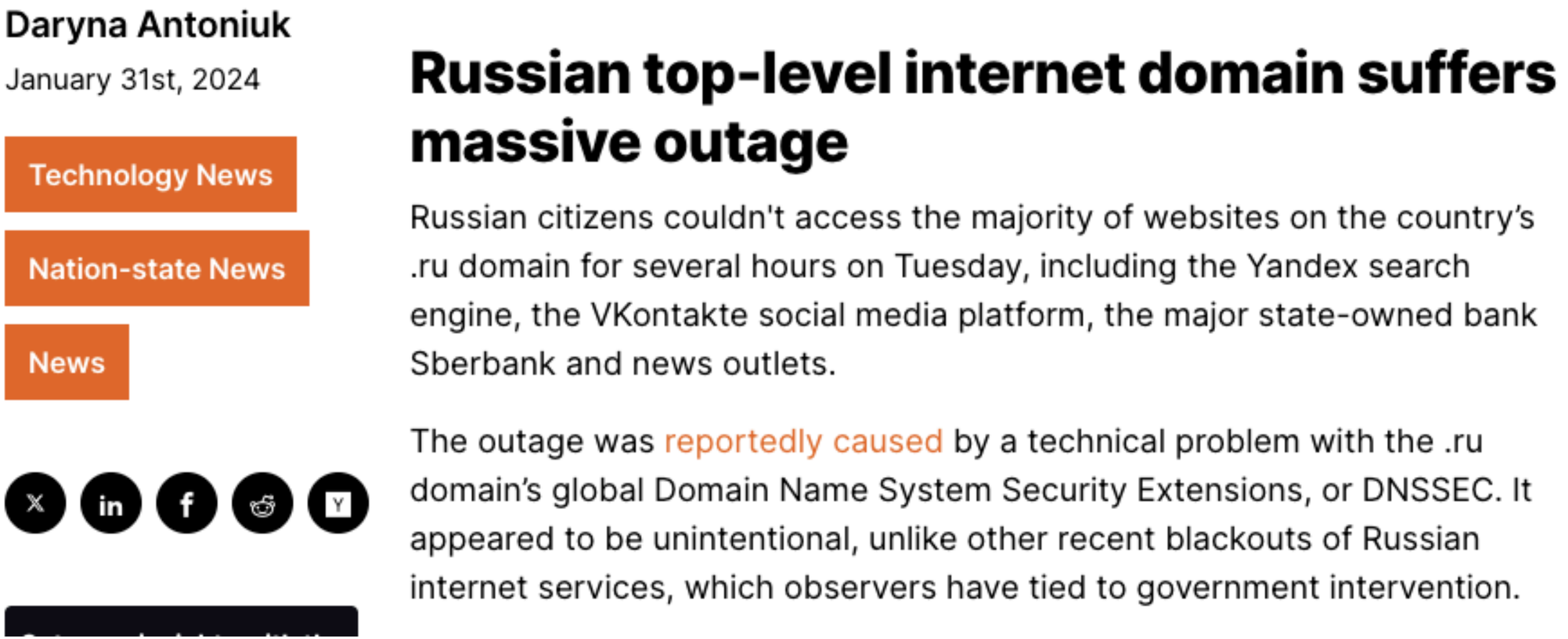
Between 07:08 UTC and 12:43 UTC on 25 January 2023, customers experienced issues with network connectivity, manifesting as long network latency and/or timeouts when attempting to connect to resources hosted in Azure regions, as well as other Microsoft services including Microsoft 365 and Power Platform. This incident also impacted Azure Government cloud services that were dependent on Azure public cloud. While most regions and services had recovered by 09:05 UTC, intermittent packet loss issues caused some customers to continue seeing connectivity issues due to two routers not being able to recover automatically. All issues were fully mitigated by 12:43 UTC.

How did we respond?

Our monitoring detected DNS and WAN issues starting at 07:11 UTC. We began investigating by reviewing all recent changes. By 08:20 UTC, as the automatic recovery was happening, we identified the problematic command that triggered the issue. Networking telemetry shows that nearly all network devices had recovered by 09:05 UTC, by which point most regions and services had recovered. Final networking equipment recovered by 09:25 UTC.

source: `https://status.azure.com/en-us/status/history/` (from 1/25/2023)

6.1800 in the news



Daryna Antoniuk

January 31st, 2024

Technology News

Nation-state News

News

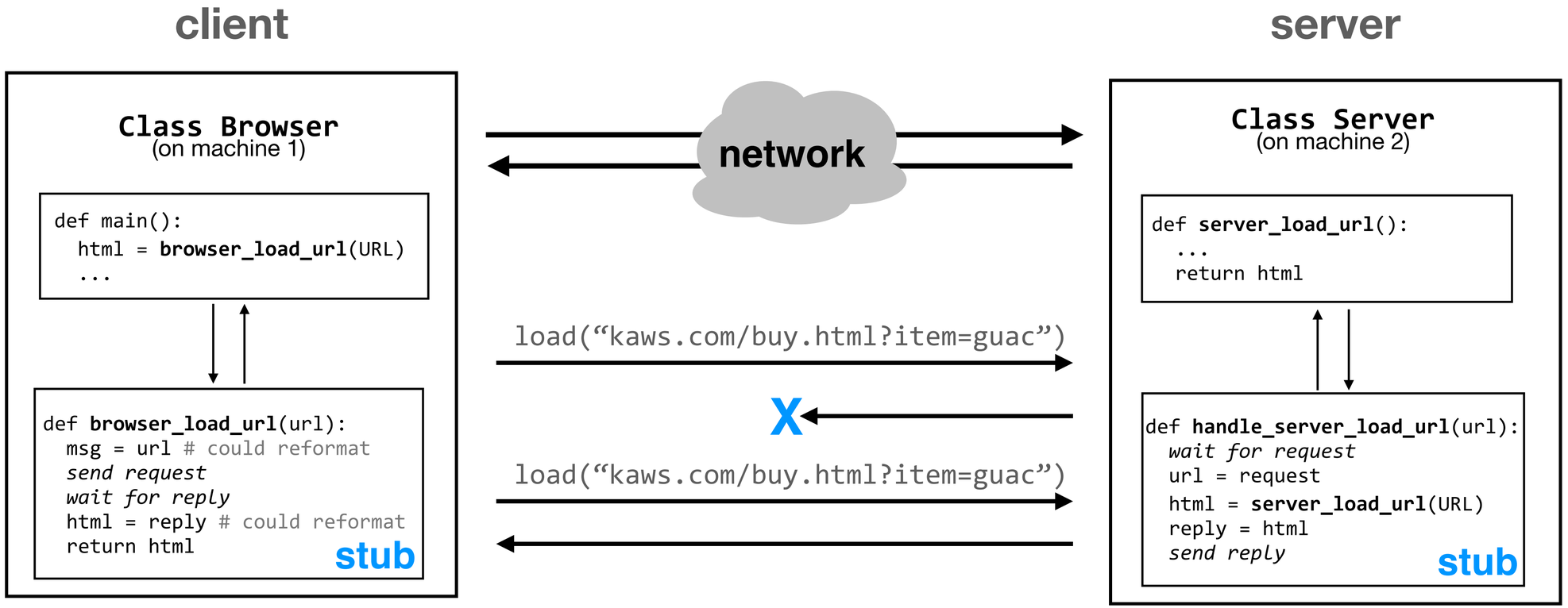
Russian top-level internet domain suffers massive outage

Russian citizens couldn't access the majority of websites on the country's .ru domain for several hours on Tuesday, including the Yandex search engine, the VKontakte social media platform, the major state-owned bank Sberbank and news outlets.

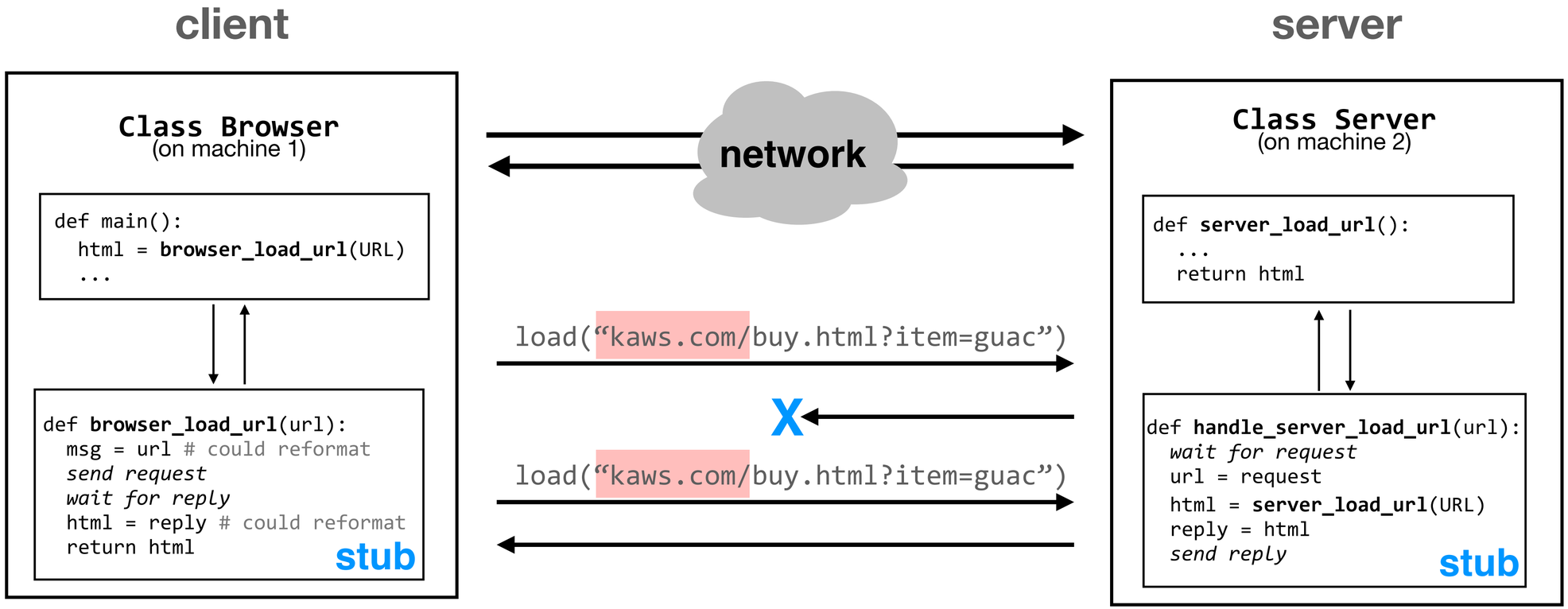
The outage was reportedly caused by a technical problem with the .ru domain's global Domain Name System Security Extensions, or DNSSEC. It appeared to be unintentional, unlike other recent blackouts of Russian internet services, which observers have tied to government intervention.

source: `https://therecord.media/russia-top-level-domain-internet-outage-dnssec`

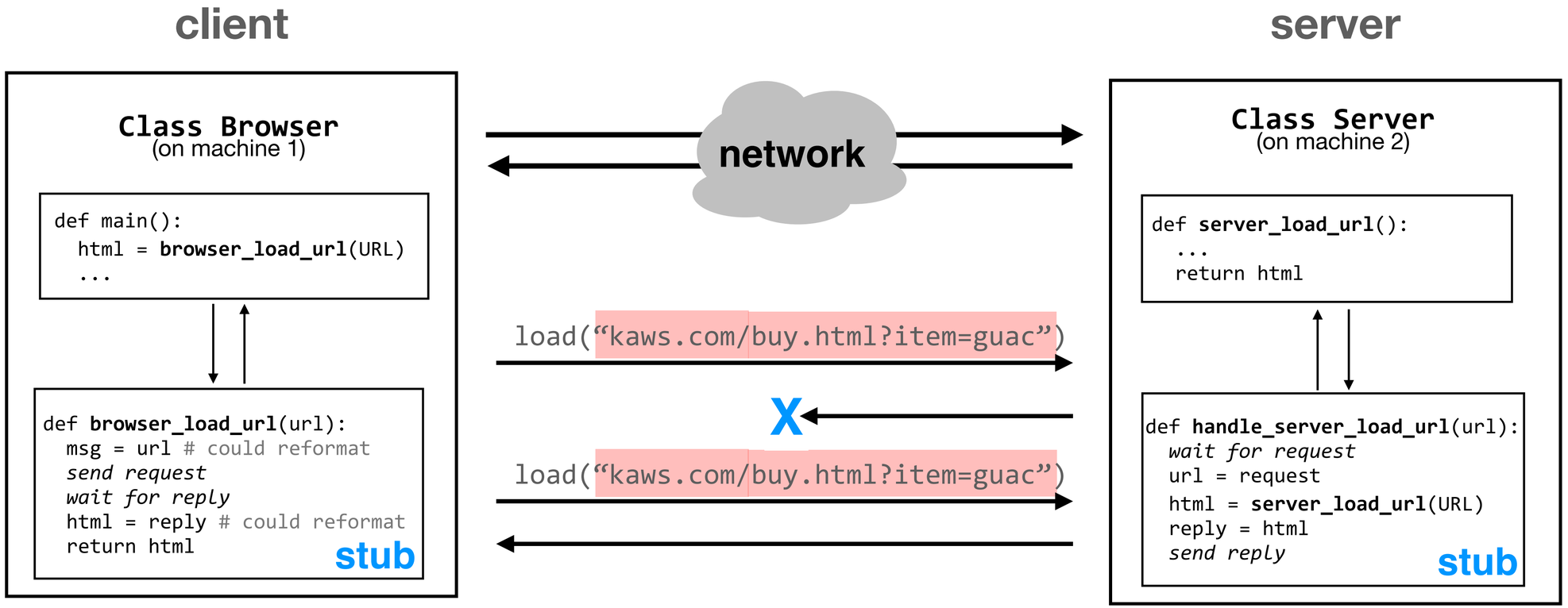
last time: enforced modularity via client/server



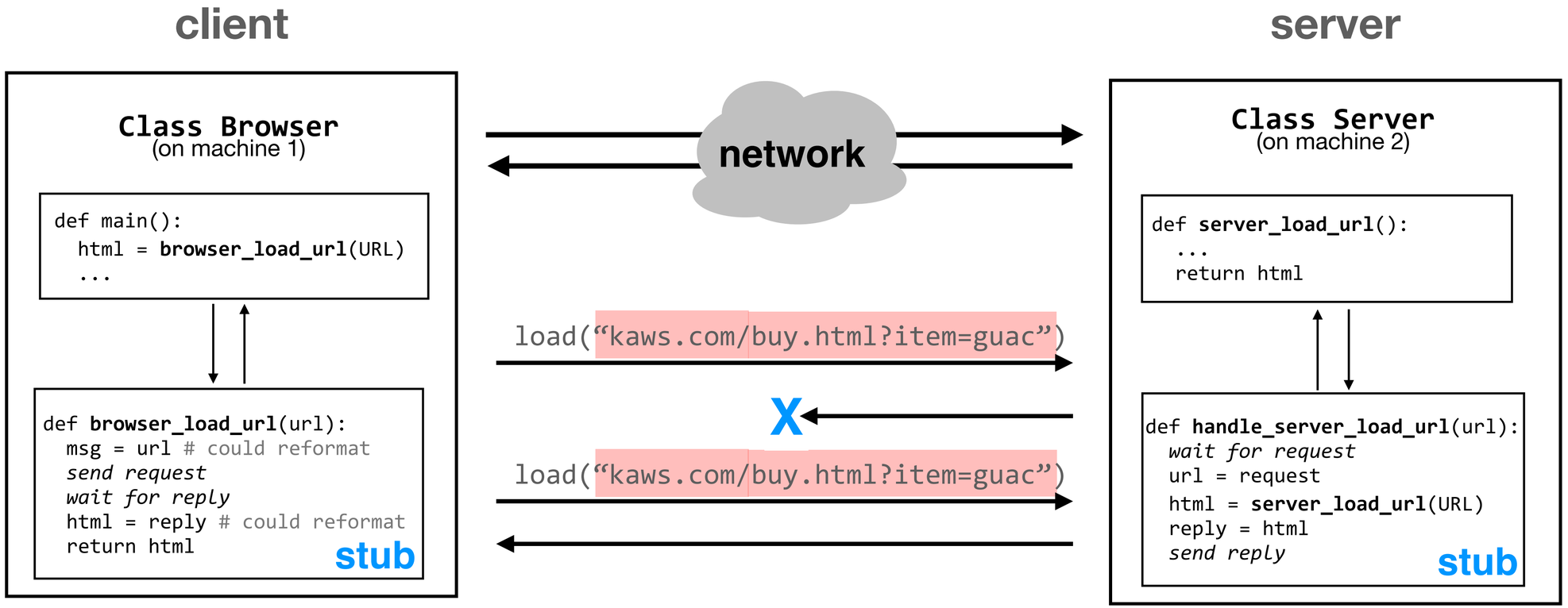
last time: enforced modularity via client/server



last time: enforced modularity via client/server



last time: enforced modularity via client/server



today: naming, which allows modules to interact

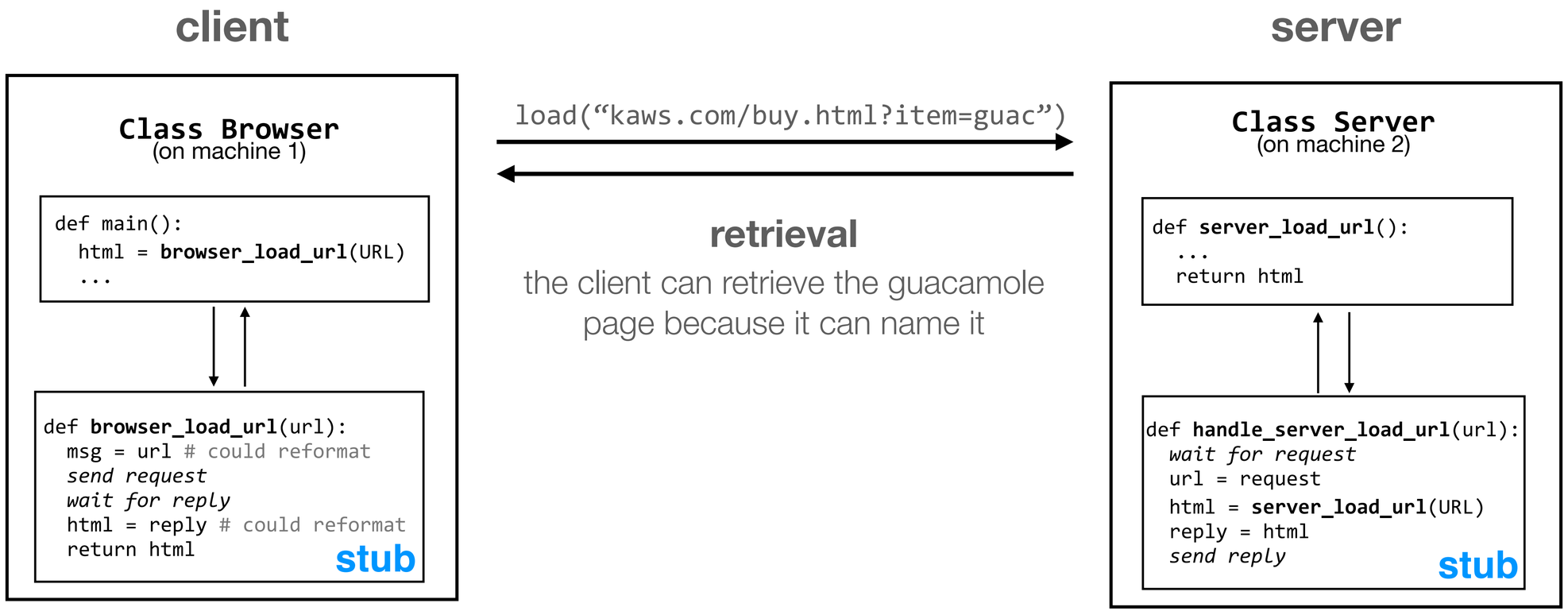
why use names?

why use names?

they let us achieve modularity by providing communication and organization, as well as a number of other properties

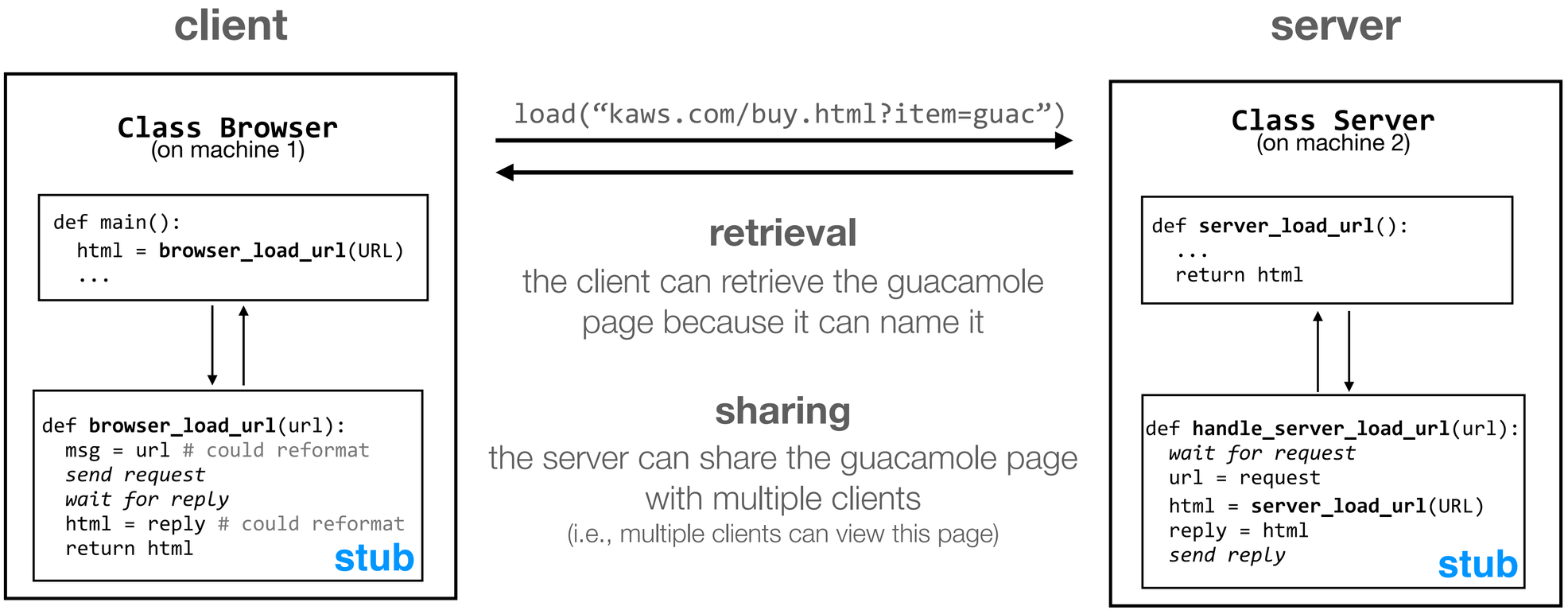
why use names?

they let us achieve modularity by providing communication and organization, as well as a number of other properties



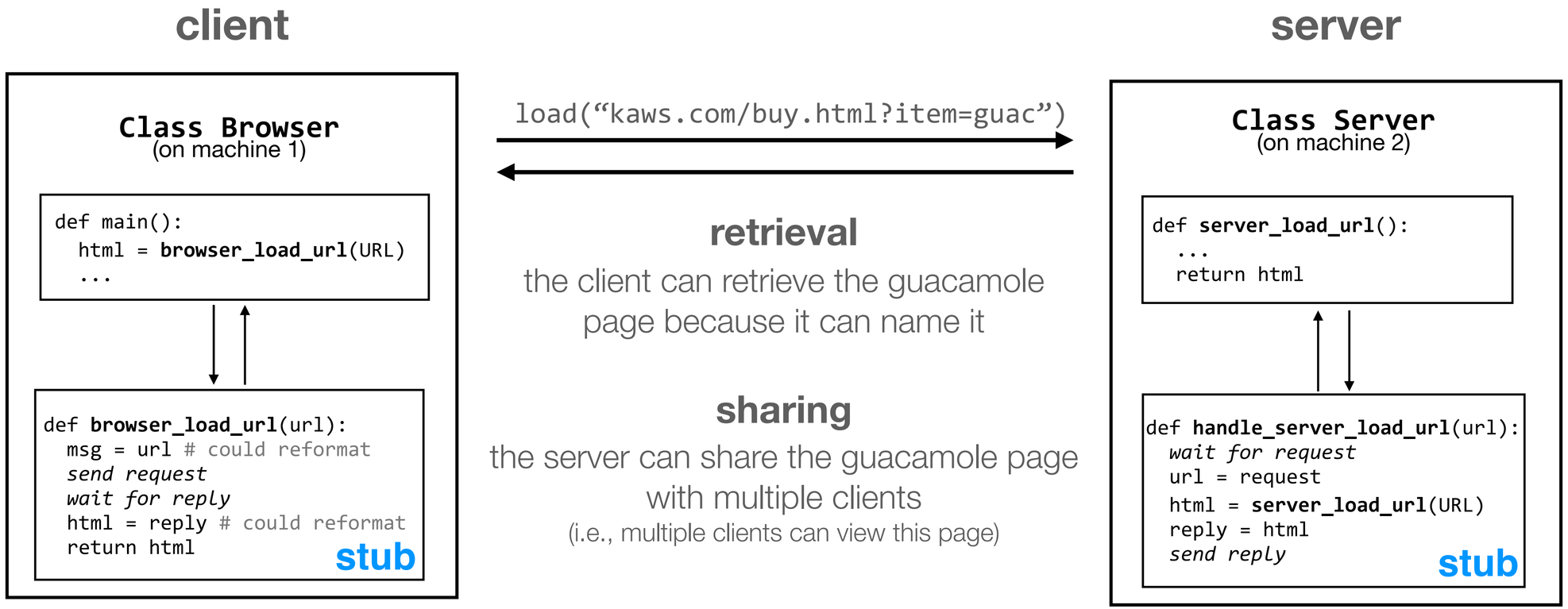
why use names?

they let us achieve modularity by providing communication and organization, as well as a number of other properties



why use names?

they let us achieve modularity by providing communication and organization, as well as a number of other properties



retrieval

the client can retrieve the guacamole page because it can name it

sharing

the server can share the guacamole page with multiple clients

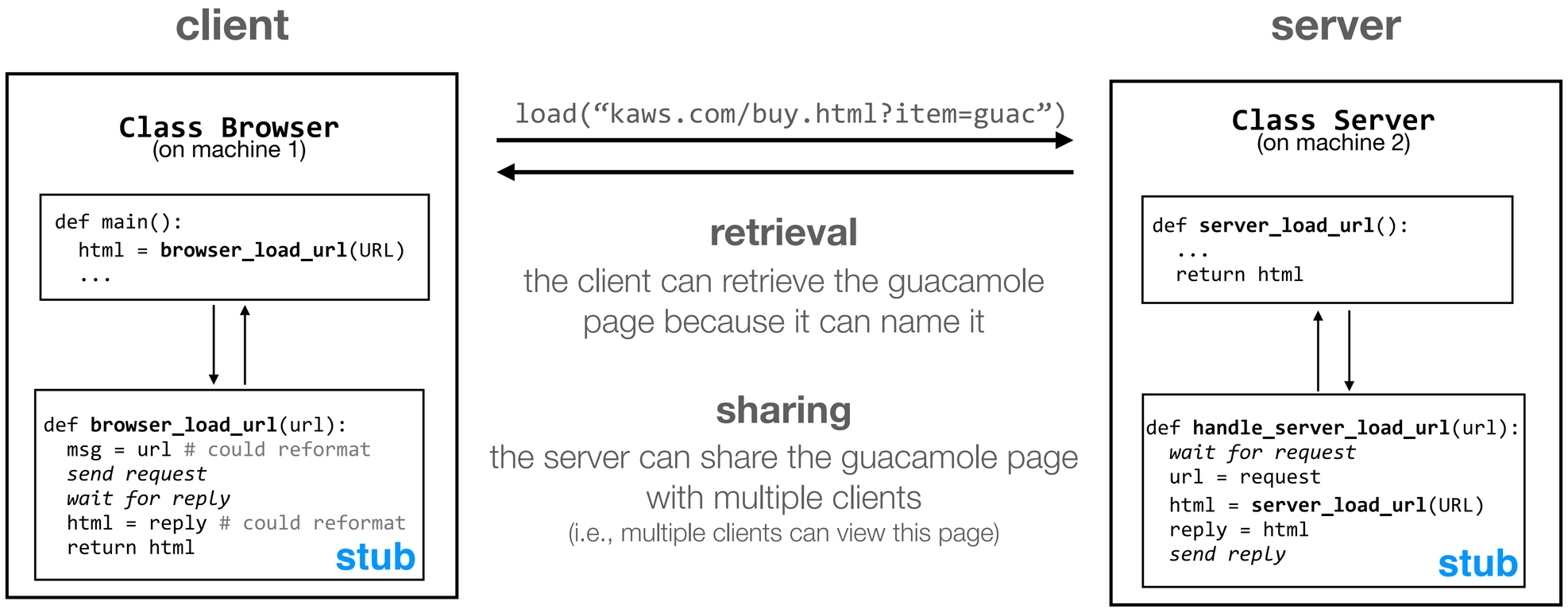
(i.e., multiple clients can view this page)

user-friendly IDs

`kaws.com` is easier to remember than (say) `18.25.4.171`; the variable name "html" is easier to remember than a particular location in memory

why use names?

they let us achieve modularity by providing communication and organization, as well as a number of other properties



retrieval

the client can retrieve the guacamole page because it can name it

sharing

the server can share the guacamole page with multiple clients

(i.e., multiple clients can view this page)

user-friendly IDs

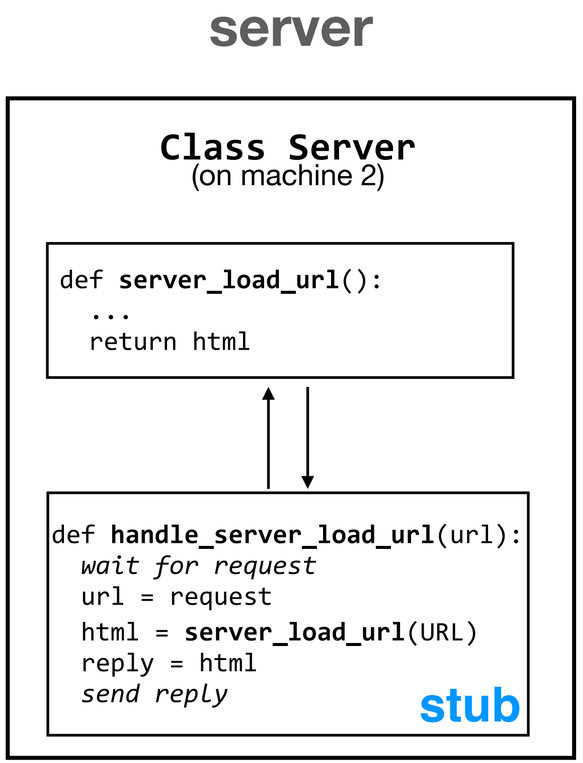
`kaws.com` is easier to remember than (say) `18.25.4.171`; the variable name "html" is easier to remember than a particular location in memory

addressing

some names also specify location information

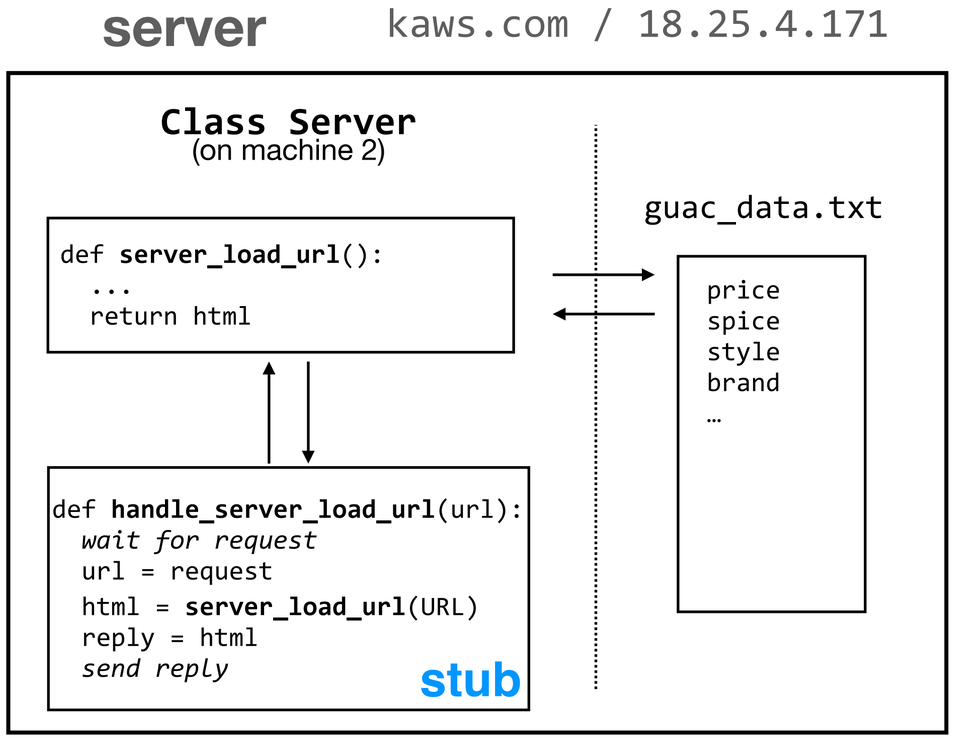
why use names?

they let us achieve modularity by providing communication and organization, as well as a number of other properties



why use names?

they let us achieve modularity by providing communication and organization, as well as a number of other properties

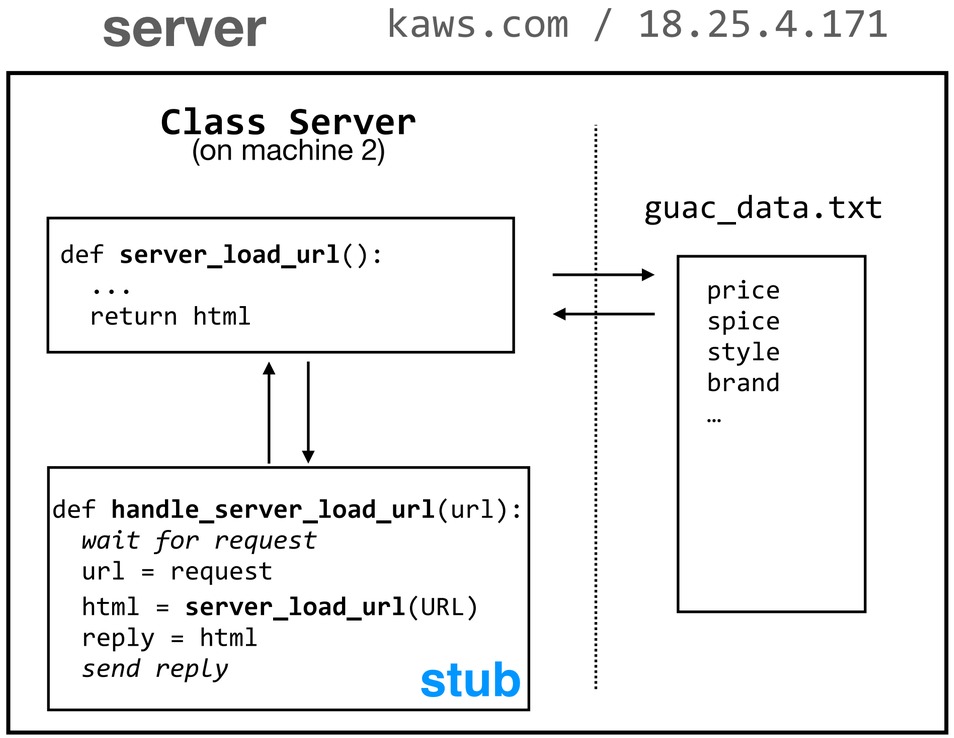


why use names?

they let us achieve modularity by providing communication and organization, as well as a number of other properties

hiding

code on the server can access `guac_data.txt` without having to worry about how the file is laid out in memory



why use names?

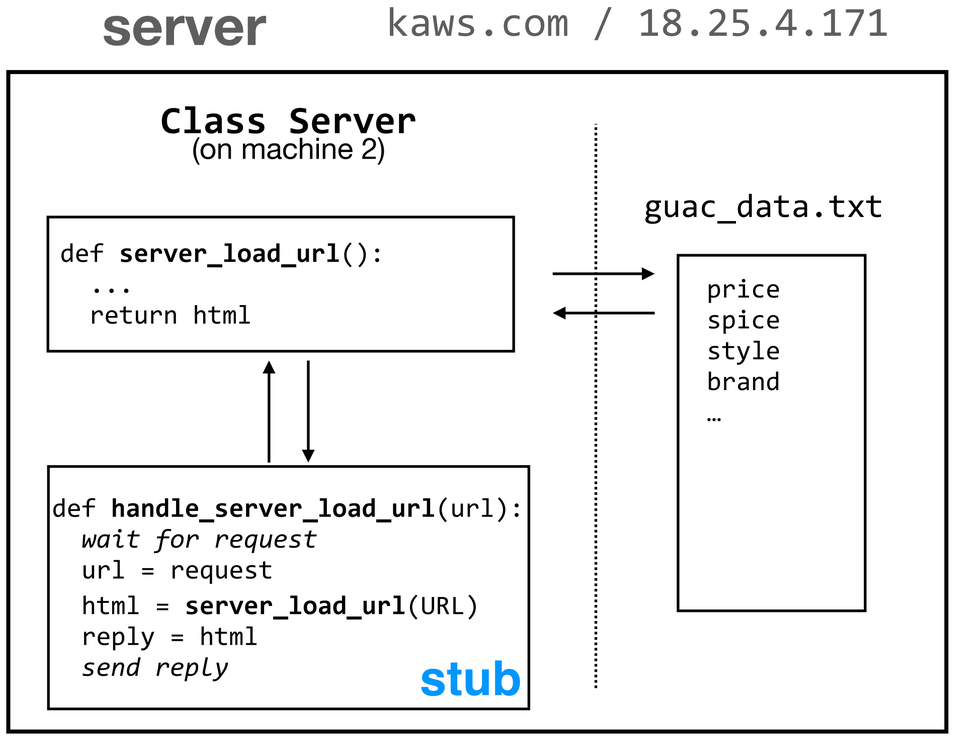
they let us achieve modularity by providing communication and organization, as well as a number of other properties

hiding

code on the server can access `guac_data.txt` without having to worry about how the file is laid out in memory

indirection

the server can change the memory layout of `guac_data.txt` without notifying the user



why use names?

they let us achieve modularity by providing communication and organization, as well as a number of other properties

why use names? they let us achieve modularity by providing communication and organization, as well as a number of other properties

the design of a system's naming scheme(s) helps it achieve these properties

why use names? they let us achieve modularity by providing communication and organization, as well as a number of other properties

the design of a system's naming scheme(s) helps it achieve these properties

a naming scheme includes

why use names? they let us achieve modularity by providing communication and organization, as well as a number of other properties

the design of a system's naming scheme(s) helps it achieve these properties

a naming scheme includes

1. the set of all possible **names**

why use names? they let us achieve modularity by providing communication and organization, as well as a number of other properties

the design of a system's naming scheme(s) helps it achieve these properties

a naming scheme includes

1. the set of all possible **names**
2. the set of all possible **values**

why use names? they let us achieve modularity by providing communication and organization, as well as a number of other properties

the design of a system's naming scheme(s) helps it achieve these properties

a naming scheme includes

1. the set of all possible **names**
2. the set of all possible **values**
3. a **look-up algorithm** to translate a name into a value (or a set of values, or "none")

6.1800 in the news

source: https://twitter.com/MosquitoCapital/status/1593541217136316416

naming case study: the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

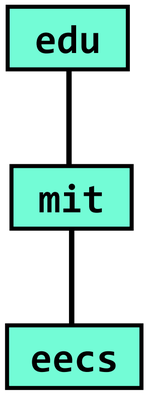
naming case study: the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation

naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

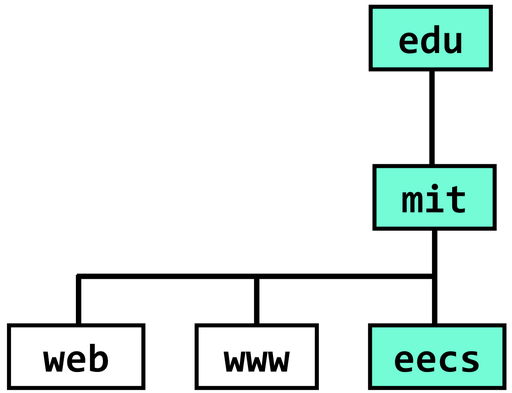
the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

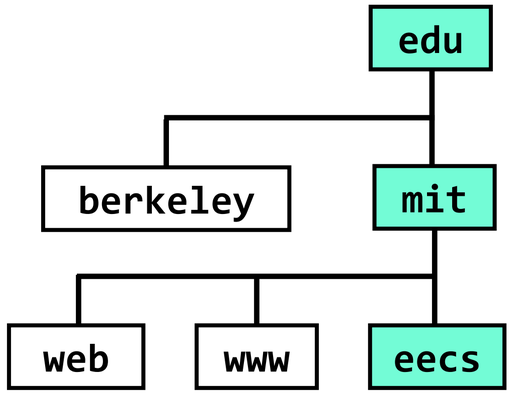
the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

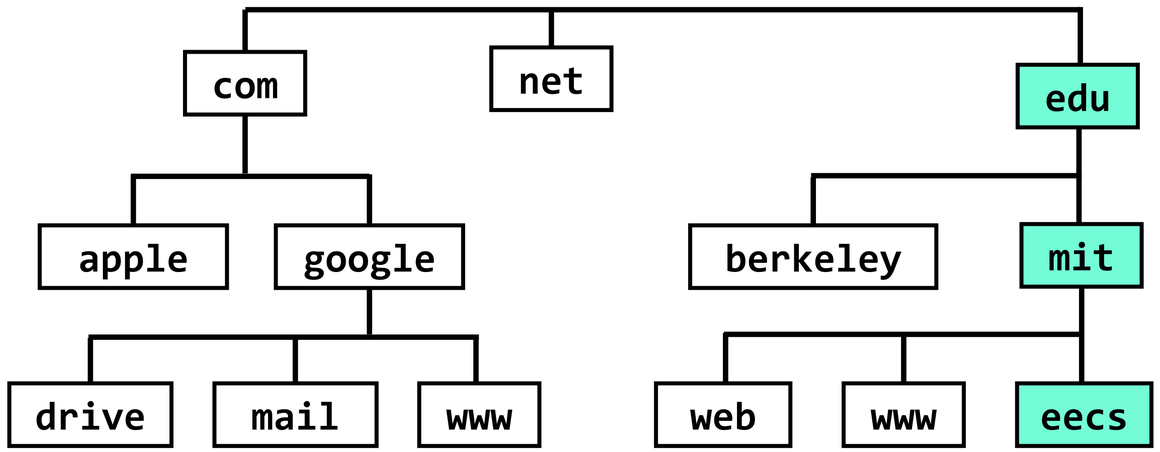
the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

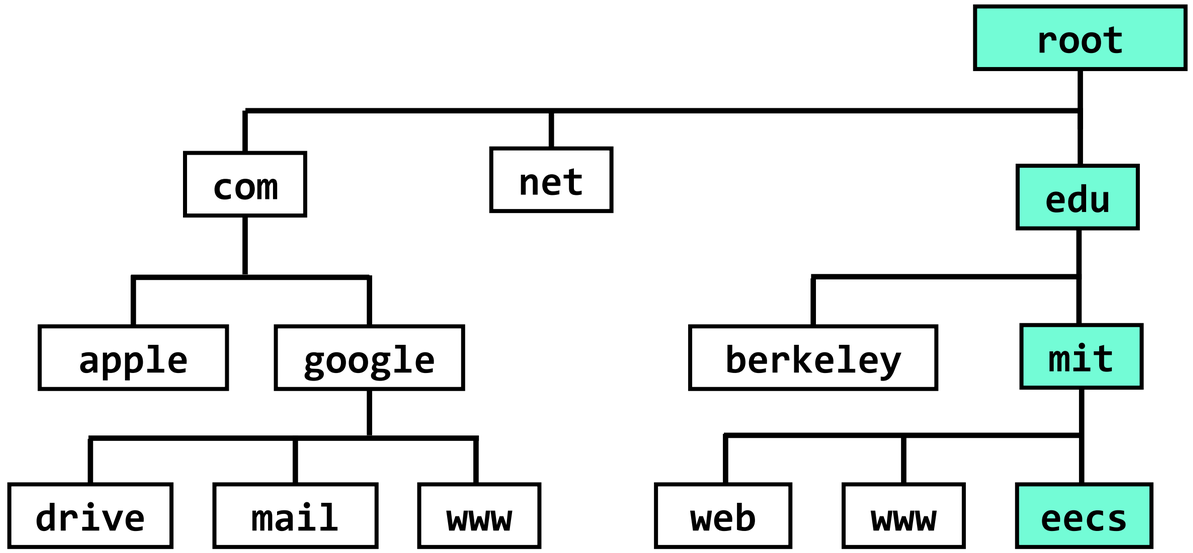
the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

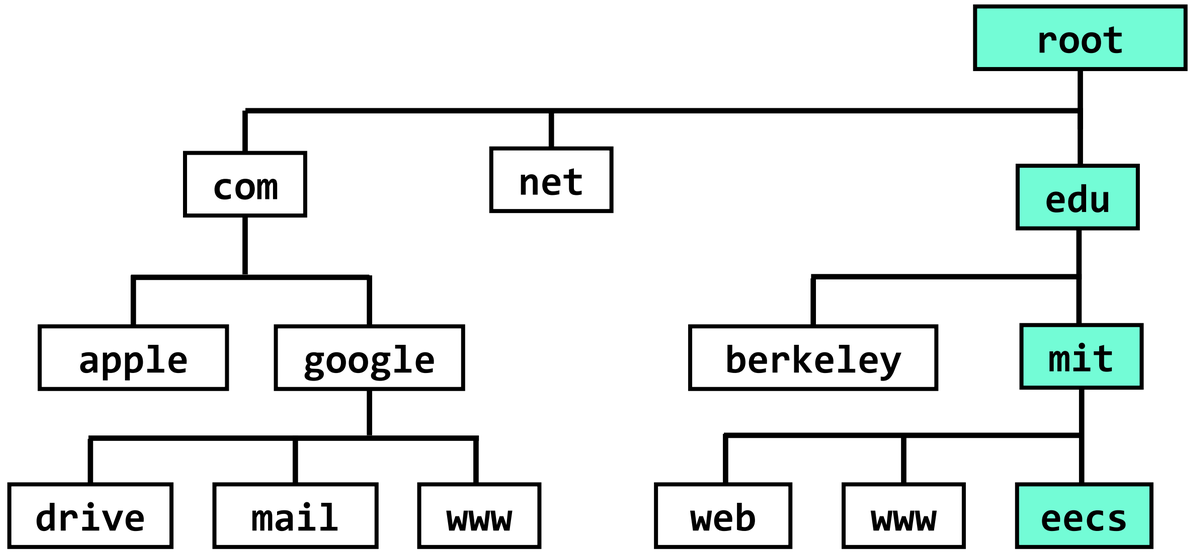
the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation

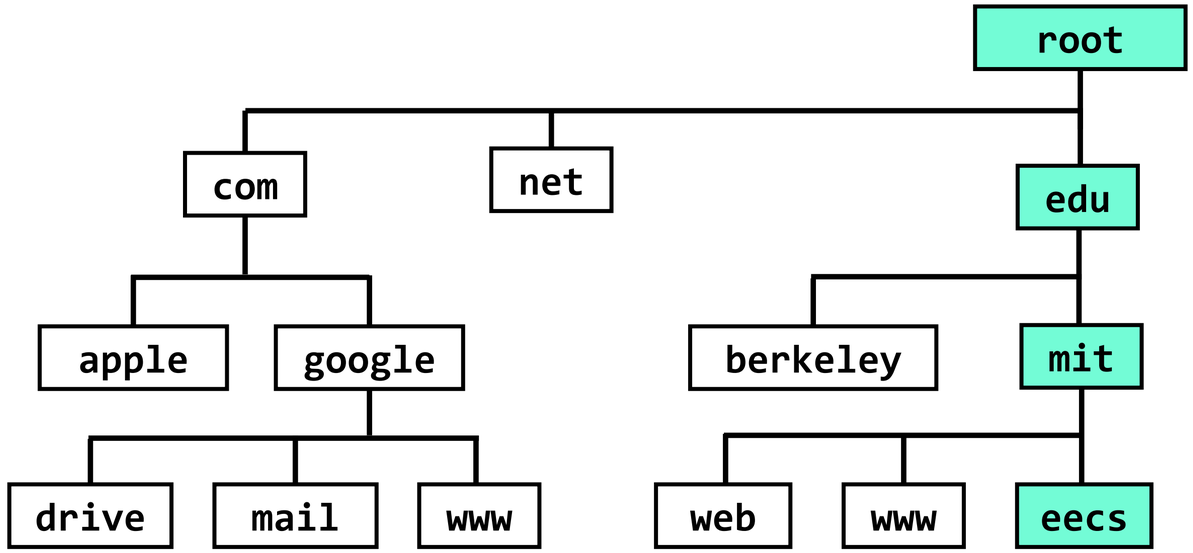


a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings

naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation

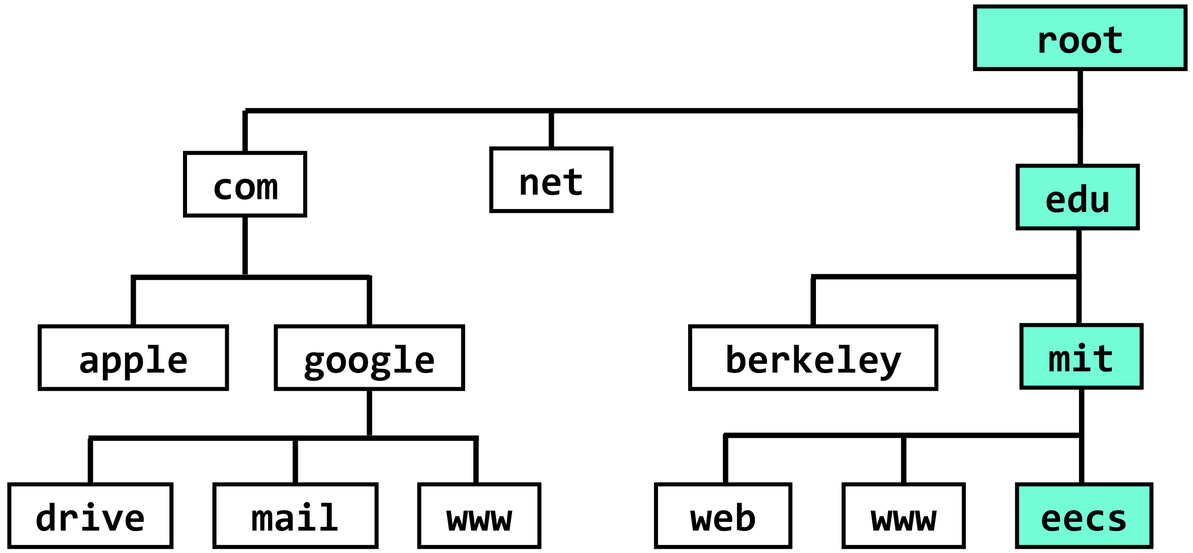


a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings

naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings

root nameserver

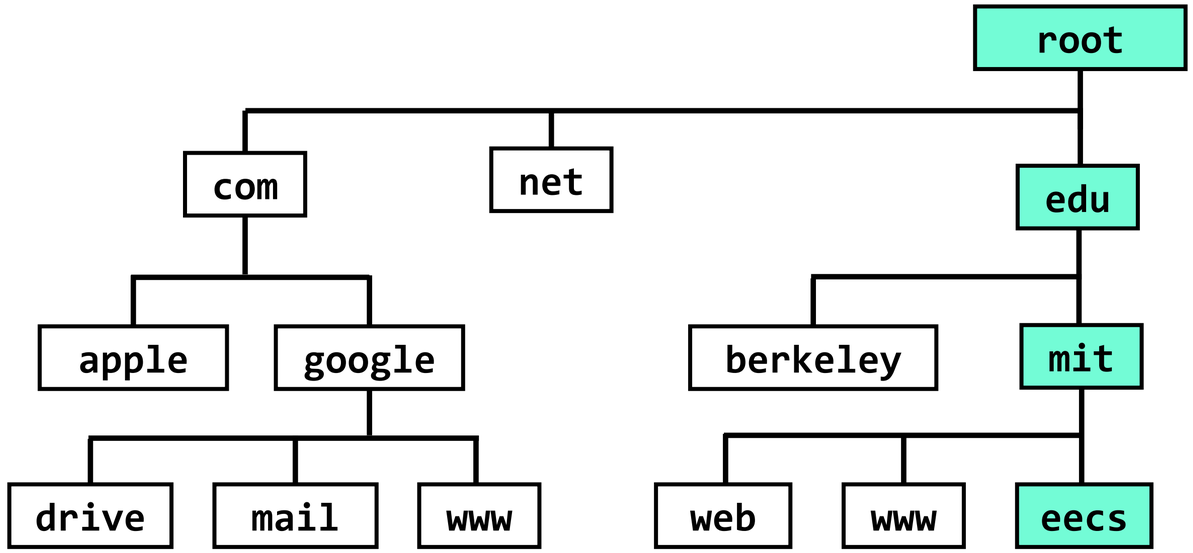
edu. nameserver

mit.edu. nameserver

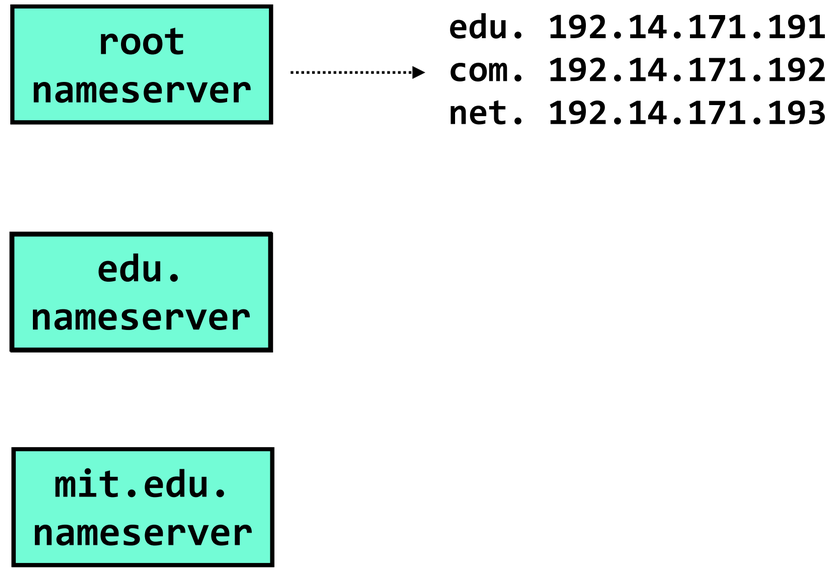
naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



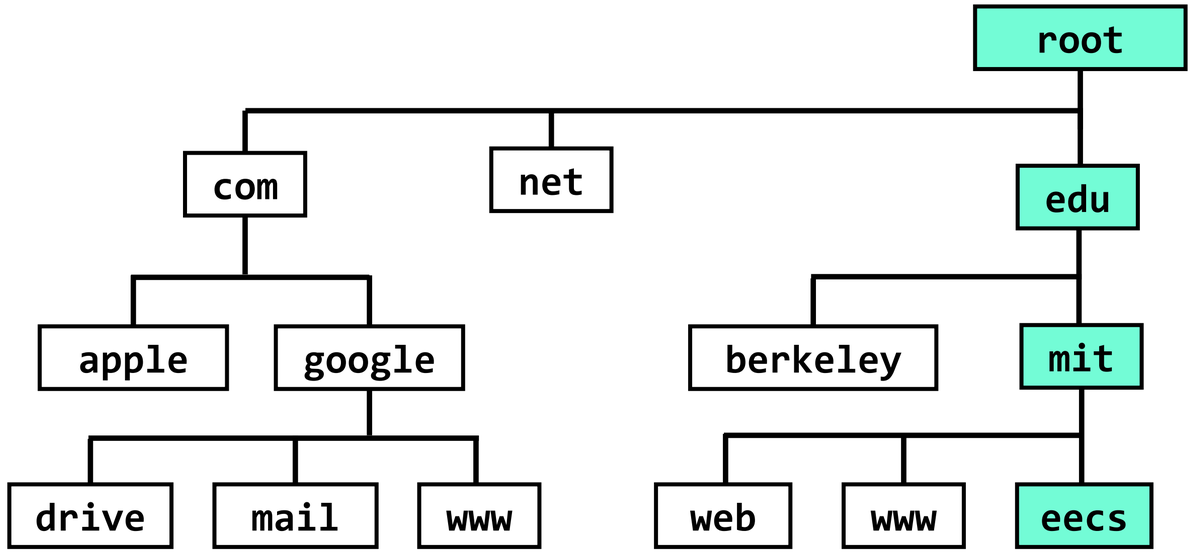
a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings



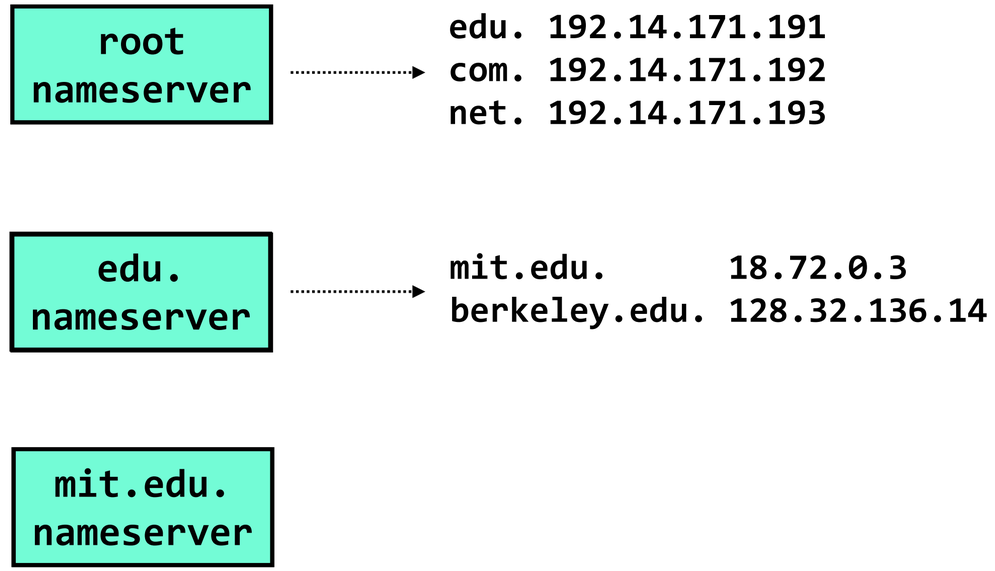
naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



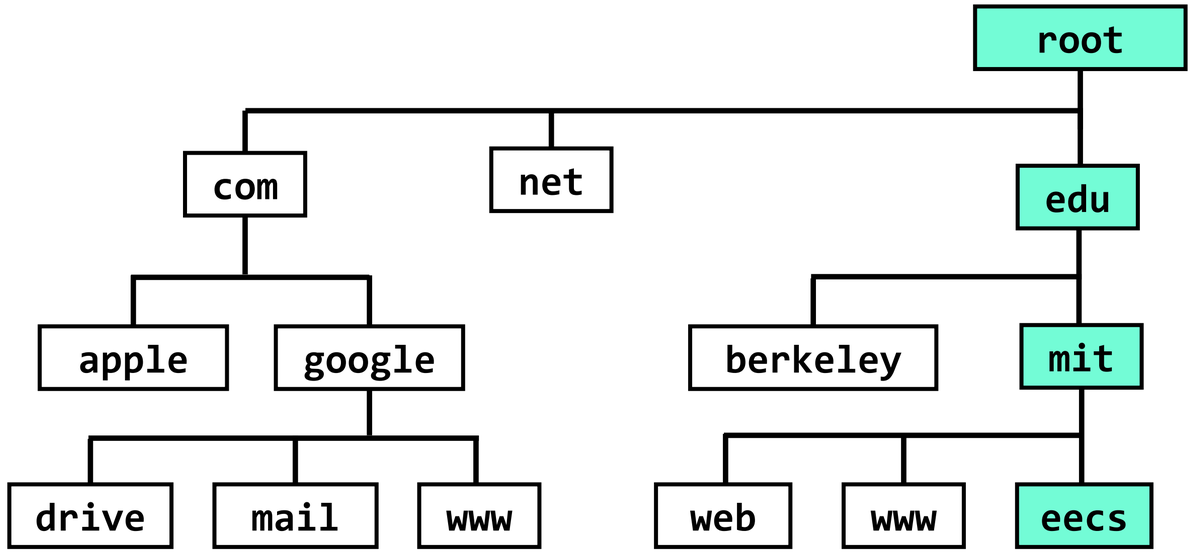
a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings



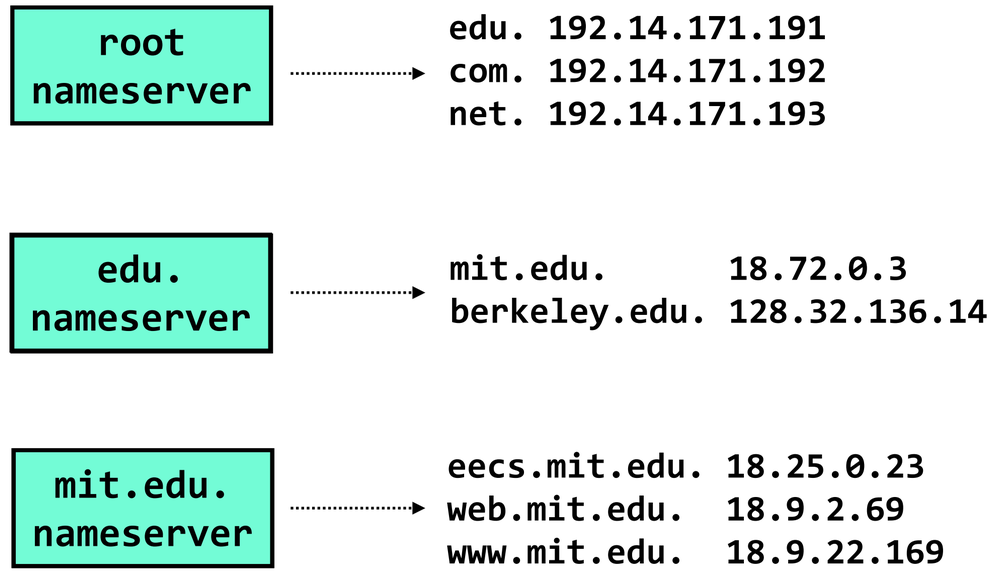
naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



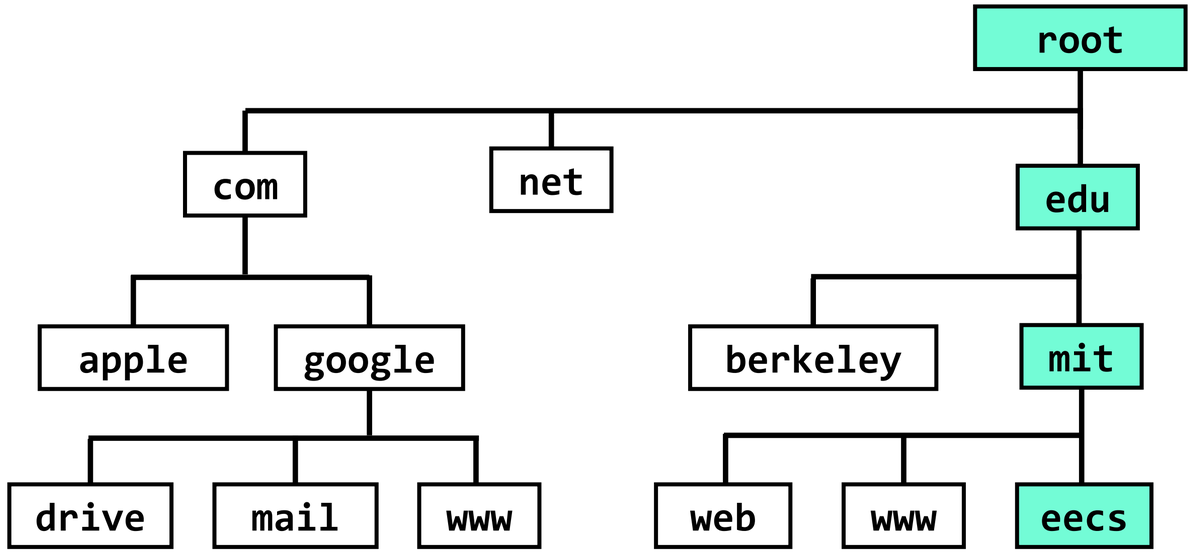
a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings



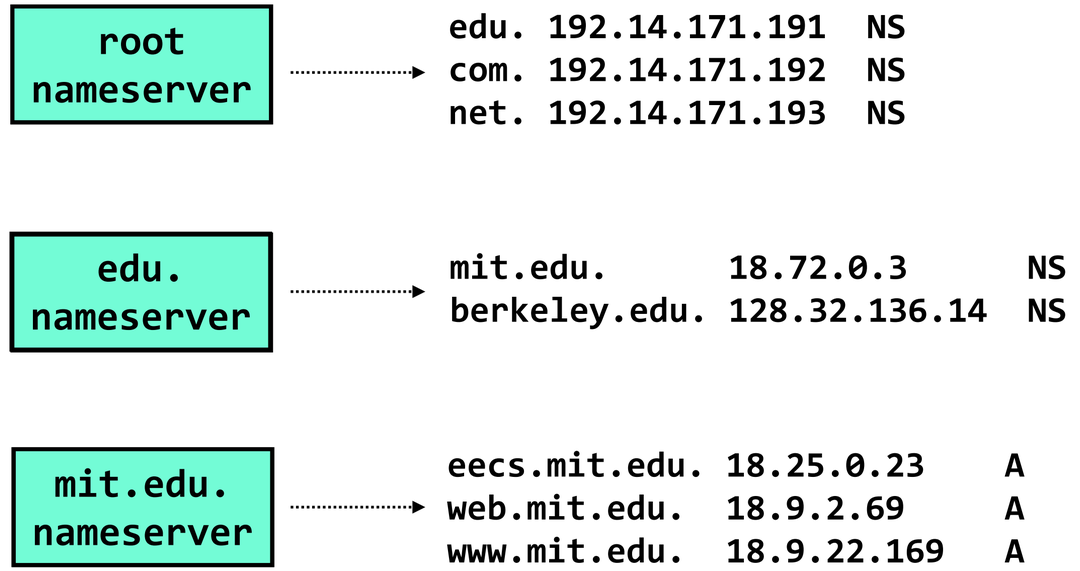
naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



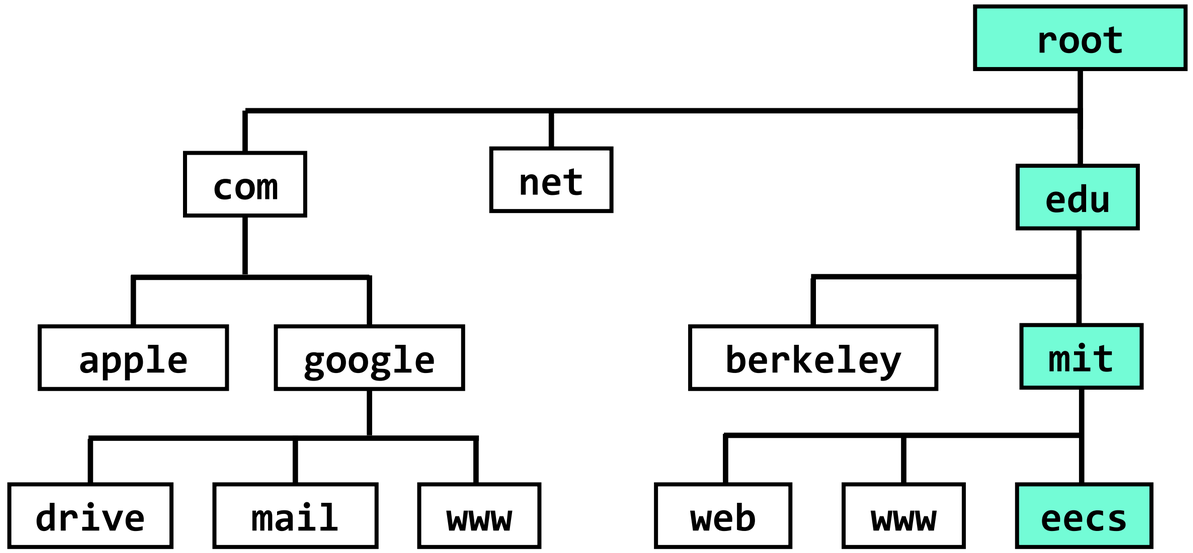
a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings



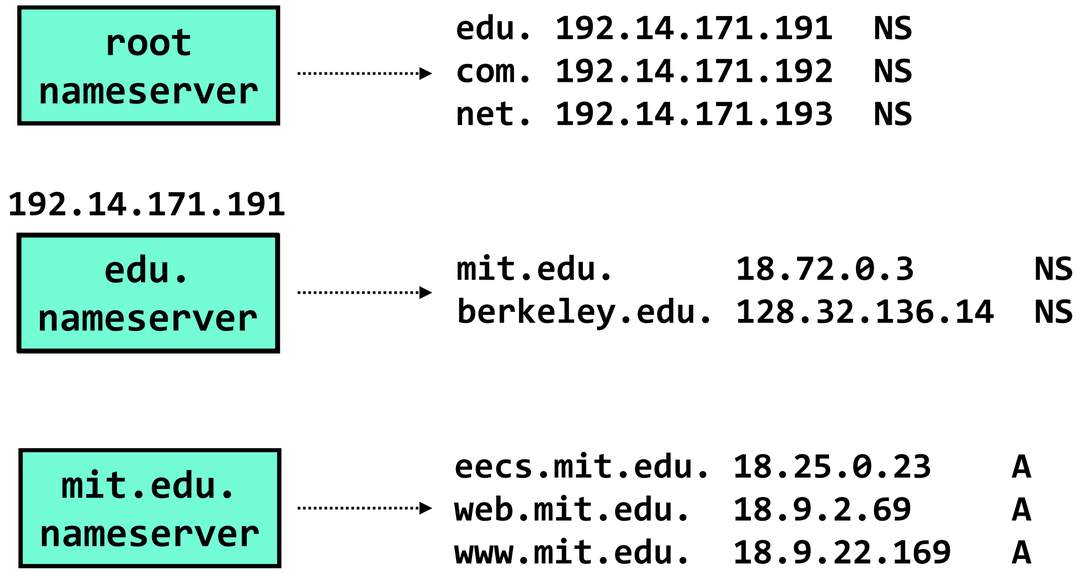
naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



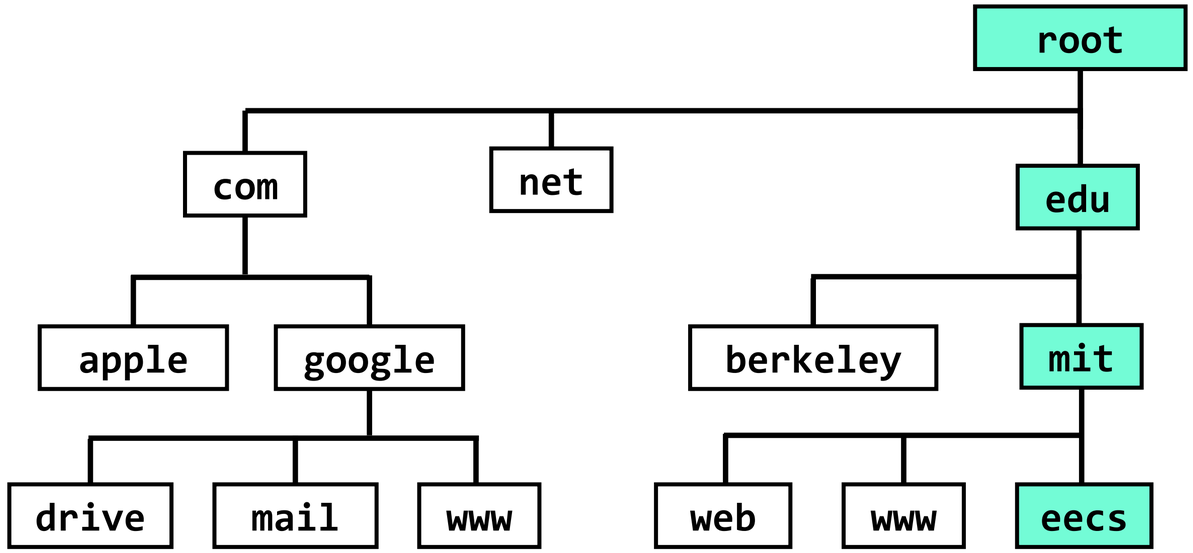
a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings



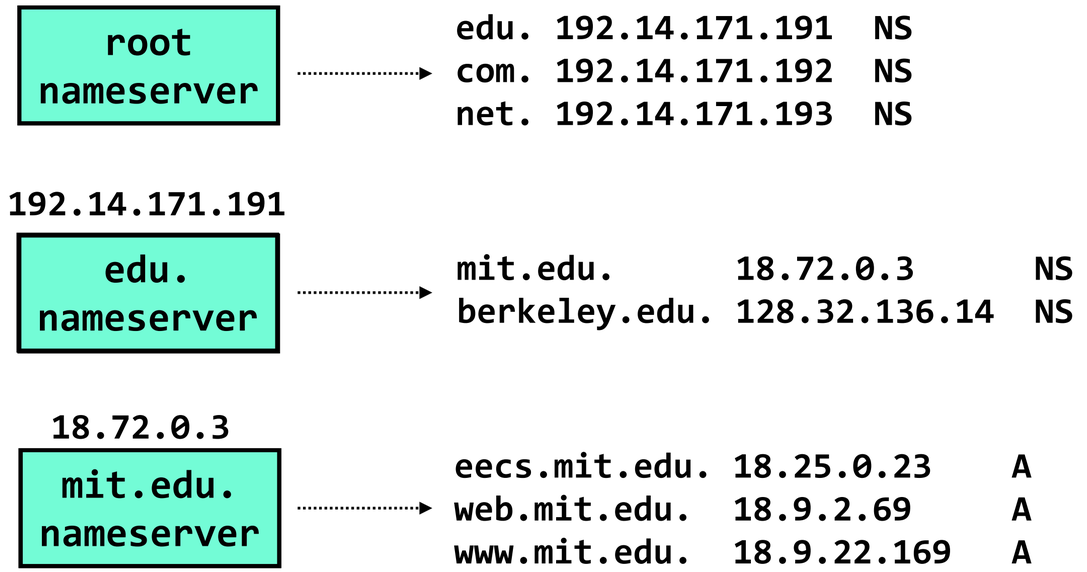
naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



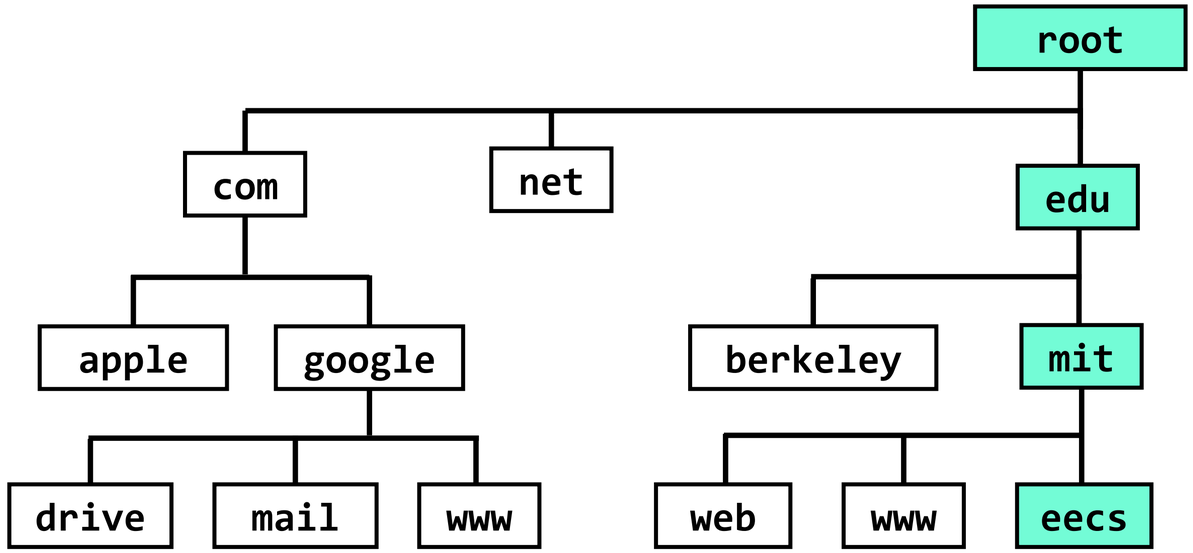
a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings



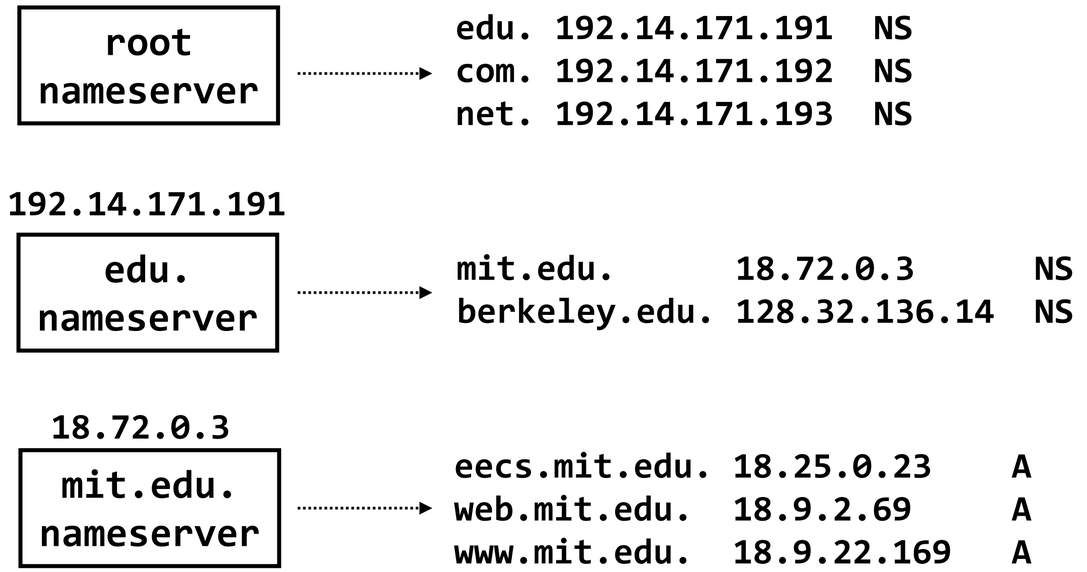
naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



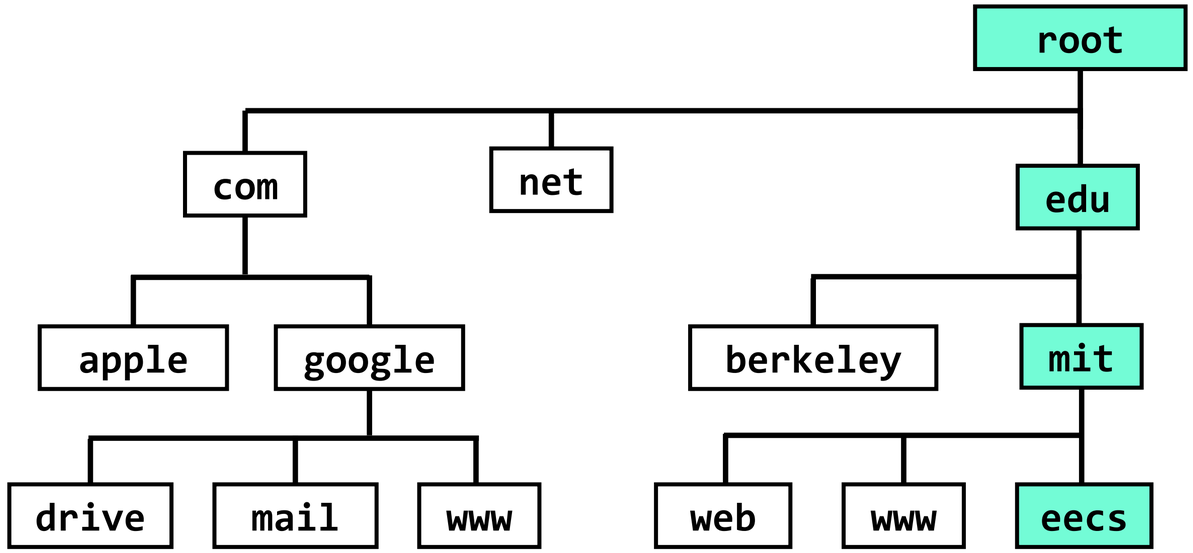
a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings



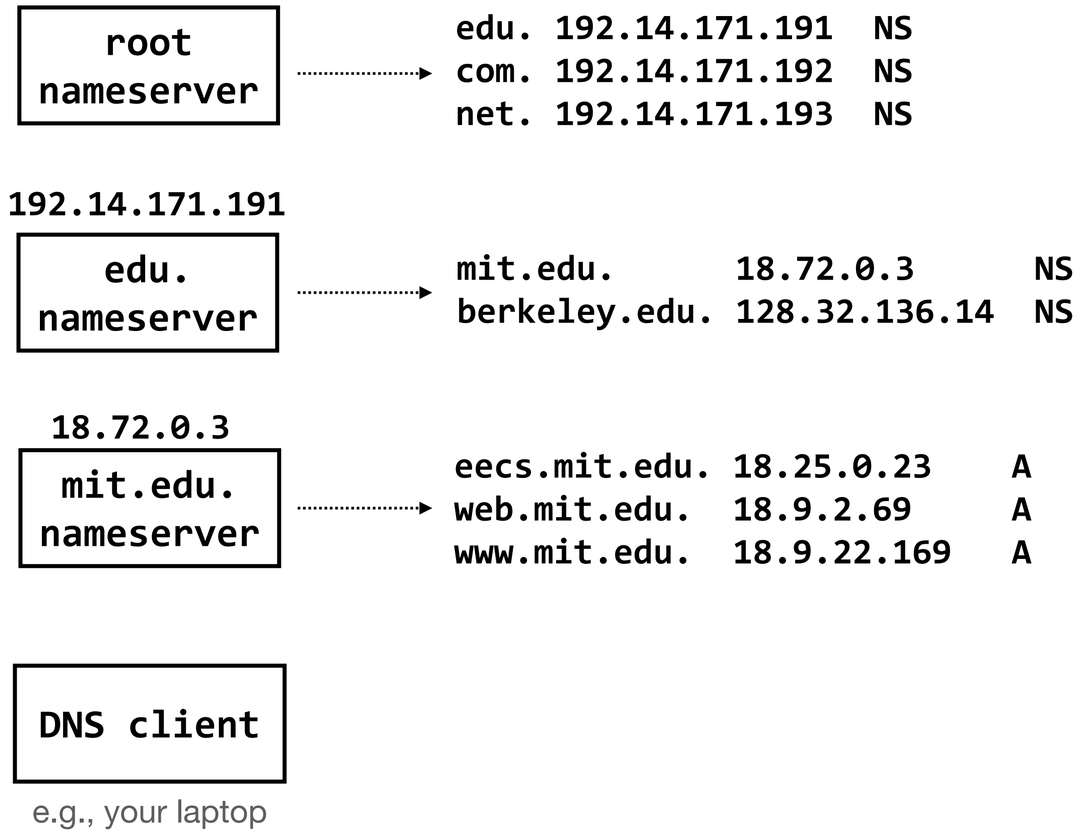
naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



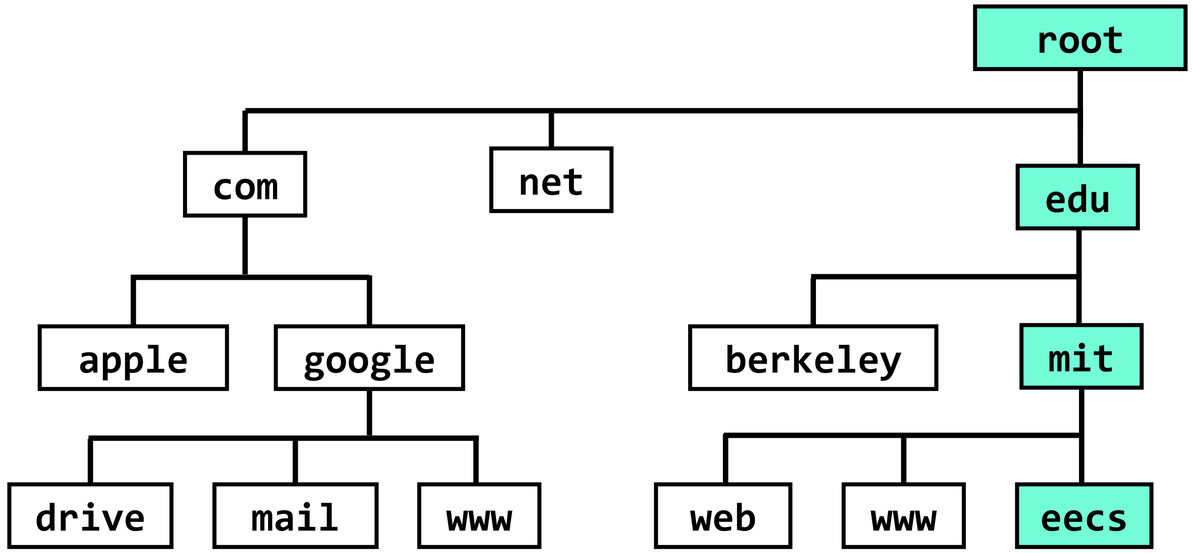
a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings



naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings

198.41.0.4

root nameserver

```
edu. 192.14.171.191  NS
com. 192.14.171.192  NS
net. 192.14.171.193  NS
```

192.14.171.191

edu. nameserver

```
mit.edu.      18.72.0.3      NS
berkeley.edu. 128.32.136.14  NS
```

18.72.0.3

mit.edu. nameserver

```
eecs.mit.edu. 18.25.0.23  A
web.mit.edu.  18.9.2.69   A
www.mit.edu.  18.9.22.169 A
```

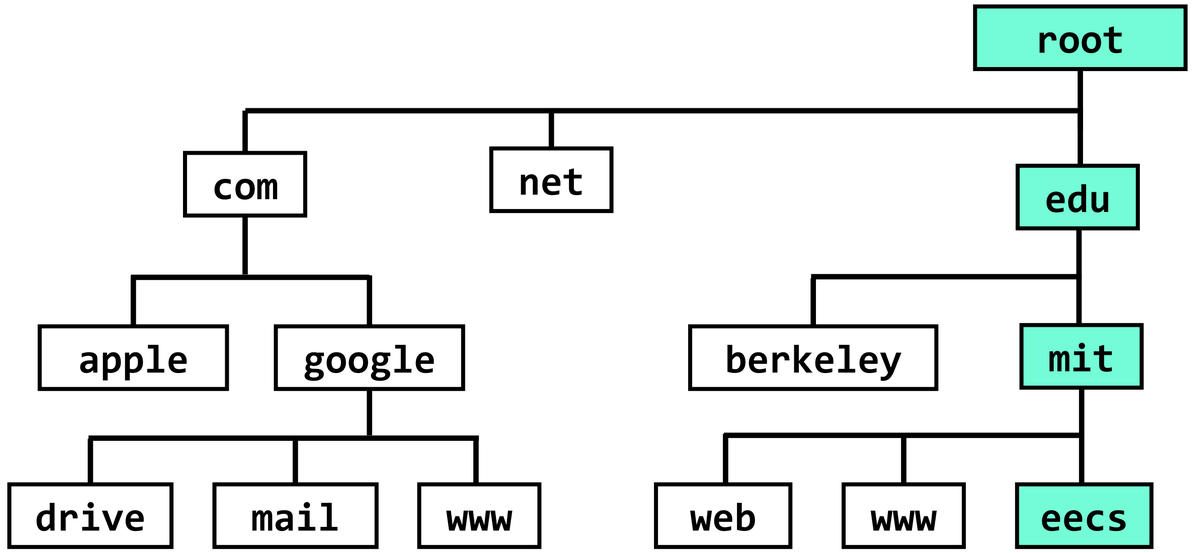
DNS client

e.g., your laptop

naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings

198.41.0.4

root nameserver

```
edu. 192.14.171.191  NS
com. 192.14.171.192  NS
net. 192.14.171.193  NS
```

192.14.171.191

edu. nameserver

```
mit.edu.      18.72.0.3      NS
berkeley.edu. 128.32.136.14  NS
```

18.72.0.3

mit.edu. nameserver

```
eecs.mit.edu. 18.25.0.23  A
web.mit.edu.  18.9.2.69   A
www.mit.edu.  18.9.22.169 A
```

DNS client

e.g., your laptop

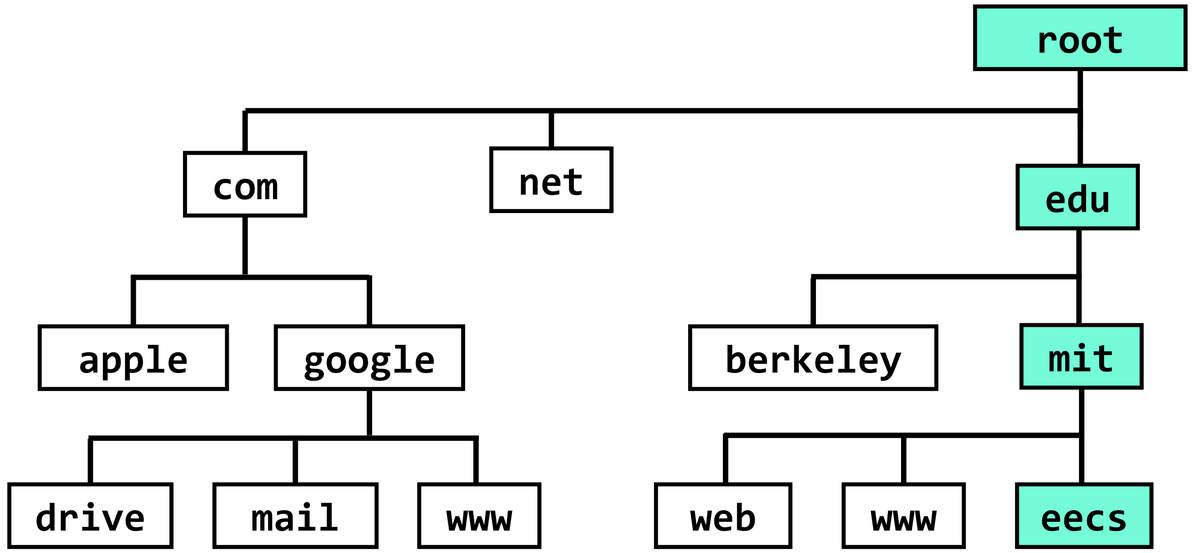
query sent to:

response:

naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings

198.41.0.4

root nameserver

```
edu. 192.14.171.191  NS
com. 192.14.171.192  NS
net. 192.14.171.193  NS
```

192.14.171.191

edu. nameserver

```
mit.edu.      18.72.0.3      NS
berkeley.edu. 128.32.136.14  NS
```

18.72.0.3

mit.edu. nameserver

```
eecs.mit.edu. 18.25.0.23  A
web.mit.edu.  18.9.2.69   A
www.mit.edu.  18.9.22.169 A
```

DNS client

e.g., your laptop

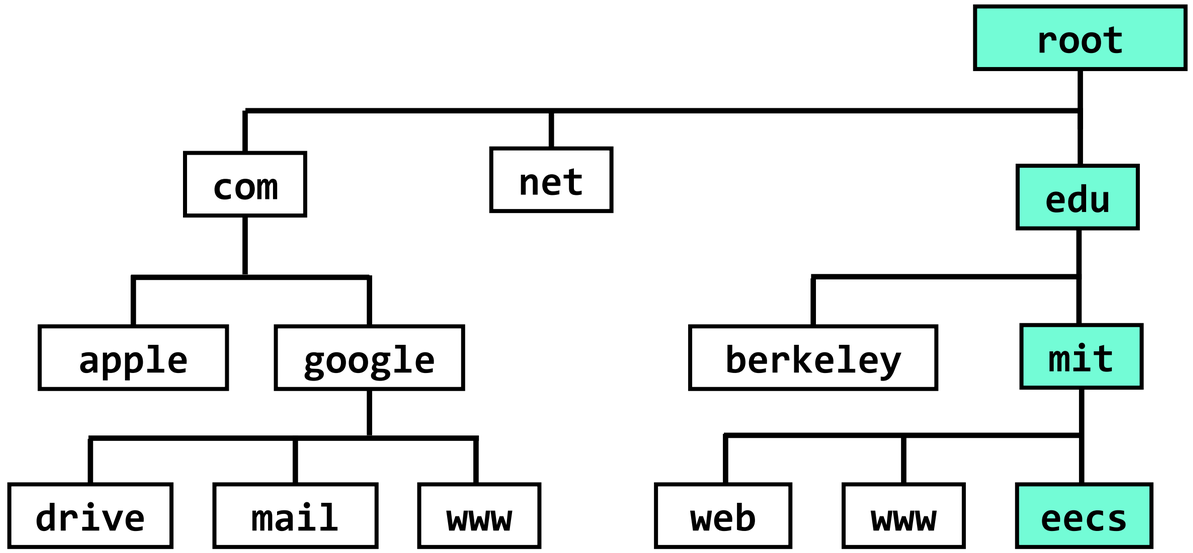
query sent to: 198.41.0.4

response:

naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings

198.41.0.4

root nameserver

```
edu.  192.14.171.191  NS
com.  192.14.171.192  NS
net.  192.14.171.193  NS
```

192.14.171.191

edu. nameserver

```
mit.edu.       18.72.0.3       NS
berkeley.edu.  128.32.136.14   NS
```

18.72.0.3

mit.edu. nameserver

```
eecs.mit.edu.  18.25.0.23    A
web.mit.edu.   18.9.2.69     A
www.mit.edu.   18.9.22.169   A
```

DNS client

e.g., your laptop

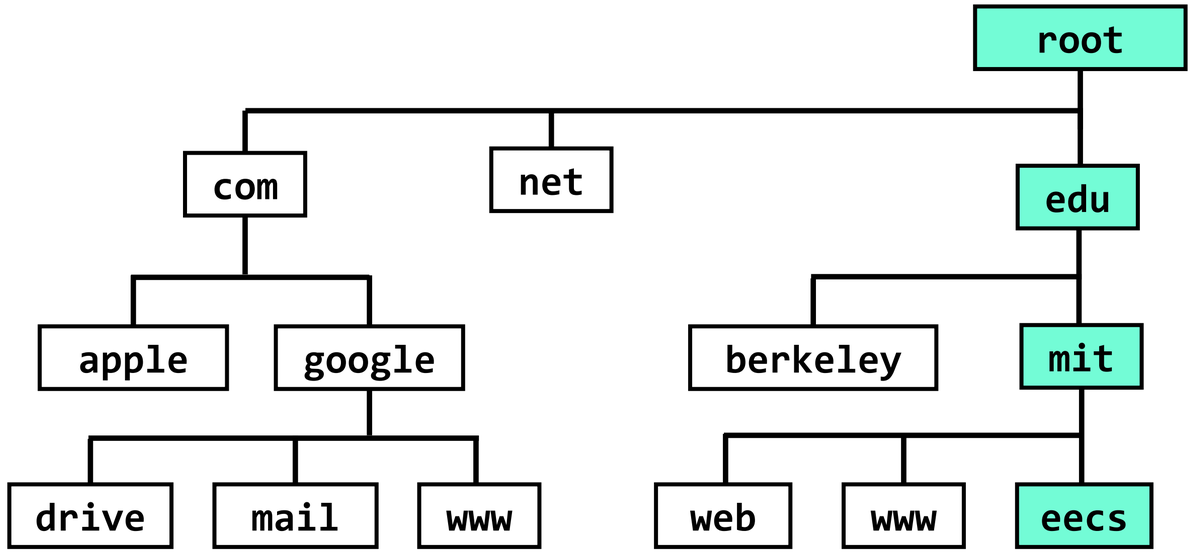
query sent to: **198.41.0.4**

response: try **192.14.171.191**

naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings

198.41.0.4

root nameserver

```
edu. 192.14.171.191  NS
com. 192.14.171.192  NS
net. 192.14.171.193  NS
```

192.14.171.191

edu. nameserver

```
mit.edu.      18.72.0.3      NS
berkeley.edu. 128.32.136.14  NS
```

18.72.0.3

mit.edu. nameserver

```
eecs.mit.edu. 18.25.0.23  A
web.mit.edu.  18.9.2.69   A
www.mit.edu.  18.9.22.169 A
```

DNS client

e.g., your laptop

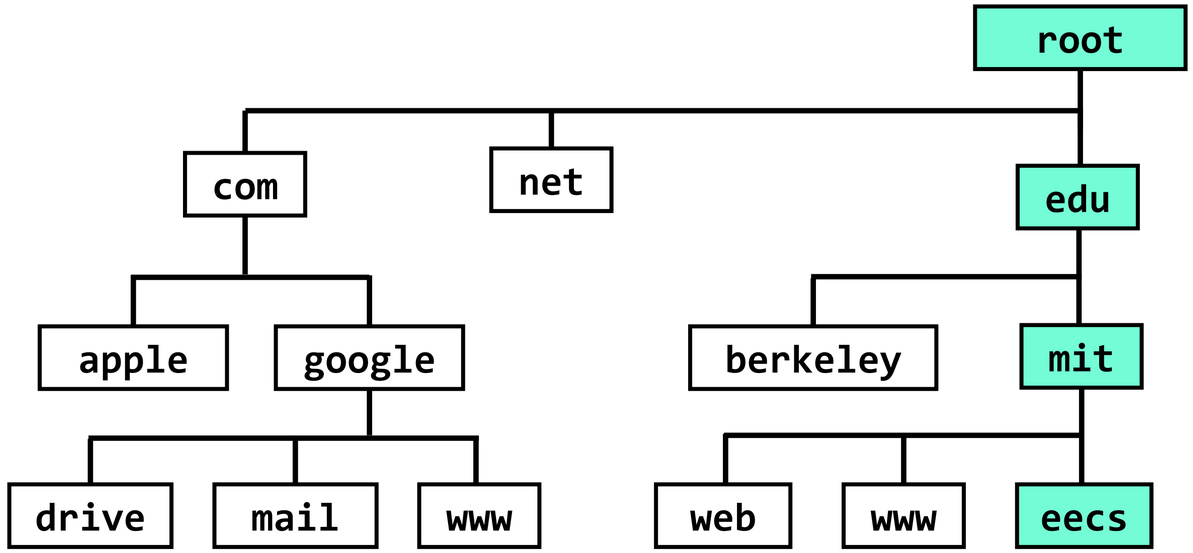
query sent to: 192.14.171.191

response:

naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings

198.41.0.4

root nameserver

```
edu. 192.14.171.191  NS
com. 192.14.171.192  NS
net. 192.14.171.193  NS
```

192.14.171.191

edu. nameserver

```
mit.edu.      18.72.0.3      NS
berkeley.edu. 128.32.136.14  NS
```

18.72.0.3

mit.edu. nameserver

```
eecs.mit.edu. 18.25.0.23  A
web.mit.edu.  18.9.2.69   A
www.mit.edu.  18.9.22.169 A
```

DNS client

e.g., your laptop

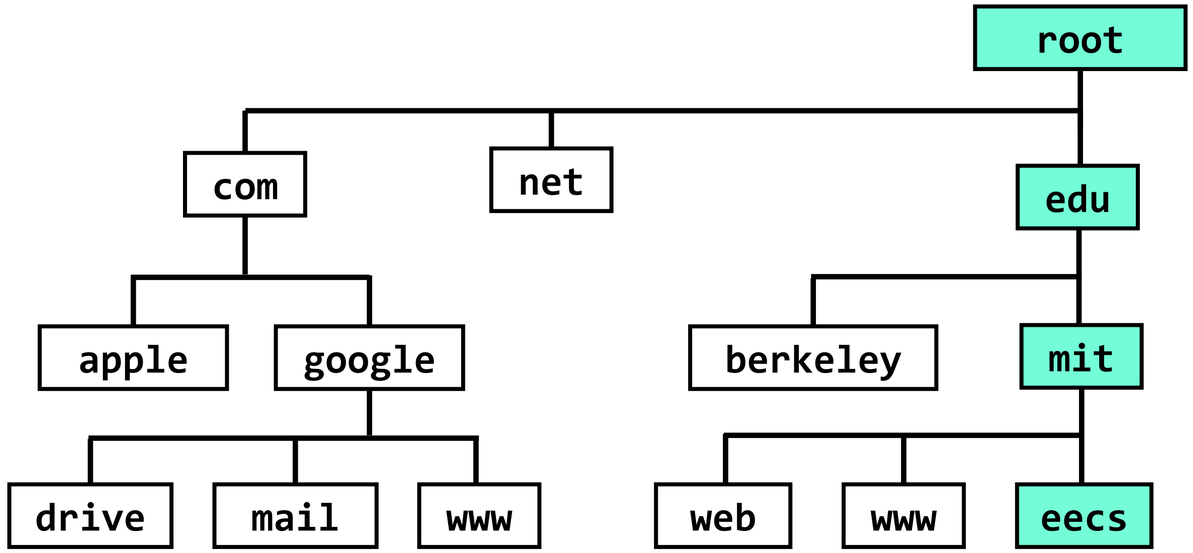
query sent to: **192.14.171.191**

response: try **18.72.0.3**

naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings

198.41.0.4

root nameserver

```
edu. 192.14.171.191  NS
com. 192.14.171.192  NS
net. 192.14.171.193  NS
```

192.14.171.191

edu. nameserver

```
mit.edu.      18.72.0.3      NS
berkeley.edu. 128.32.136.14  NS
```

18.72.0.3

mit.edu. nameserver

```
eecs.mit.edu. 18.25.0.23  A
web.mit.edu.  18.9.2.69   A
www.mit.edu.  18.9.22.169 A
```

DNS client

e.g., your laptop

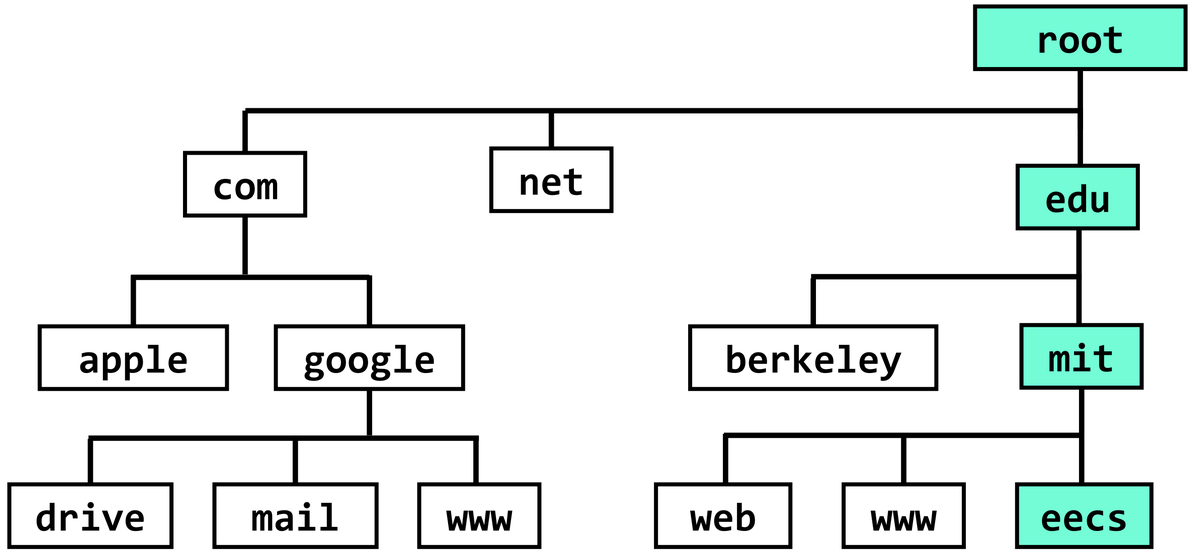
query sent to: **18.72.0.3**

response:

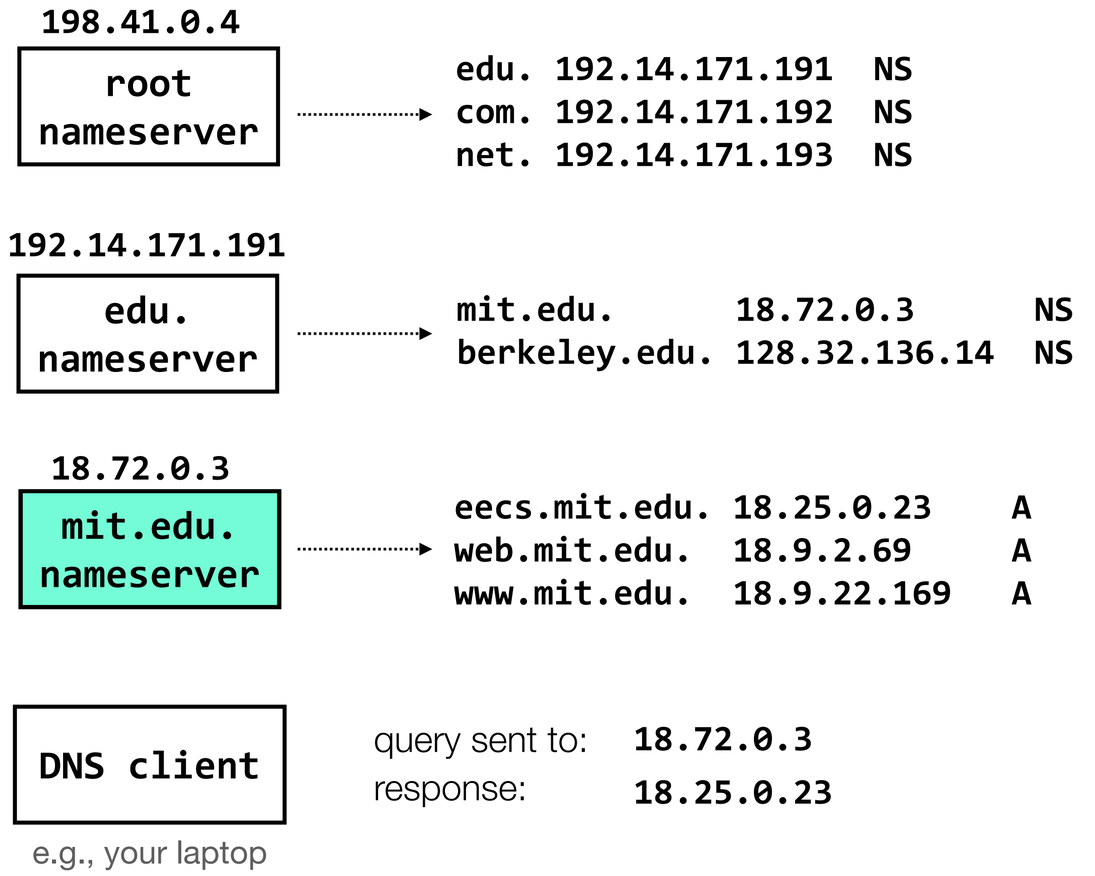
naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



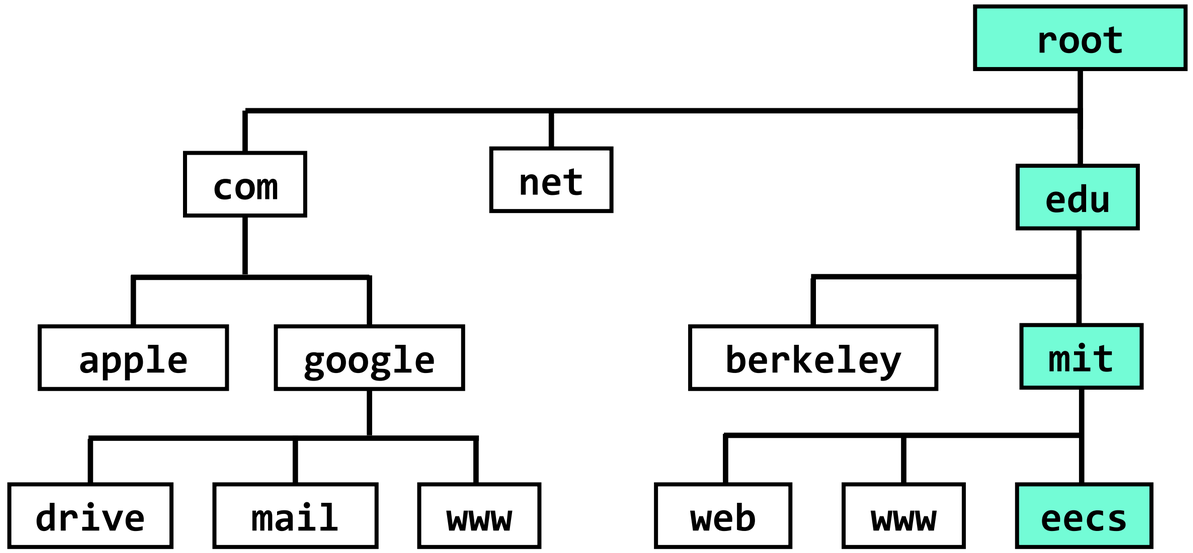
a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings



naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation

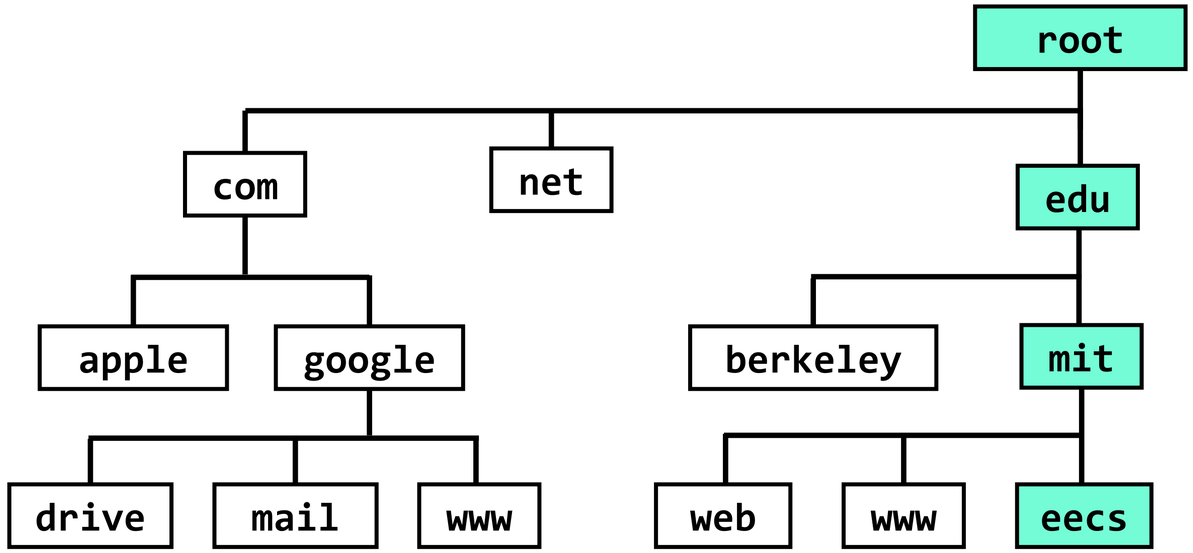


a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings

naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



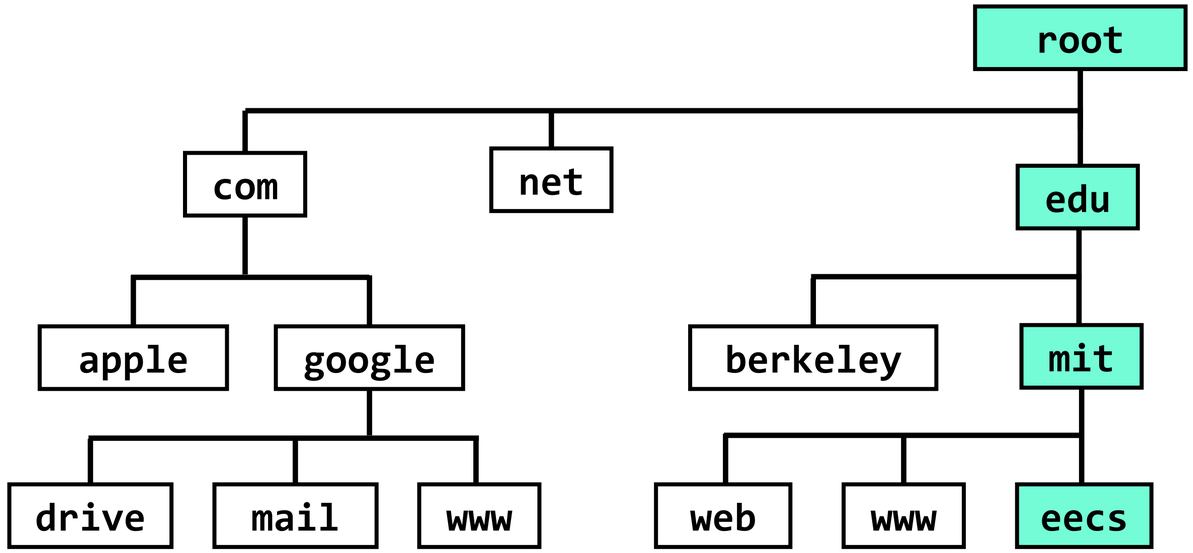
a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings

performance issue: this is a *lot* of queries, especially to the root server

naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings

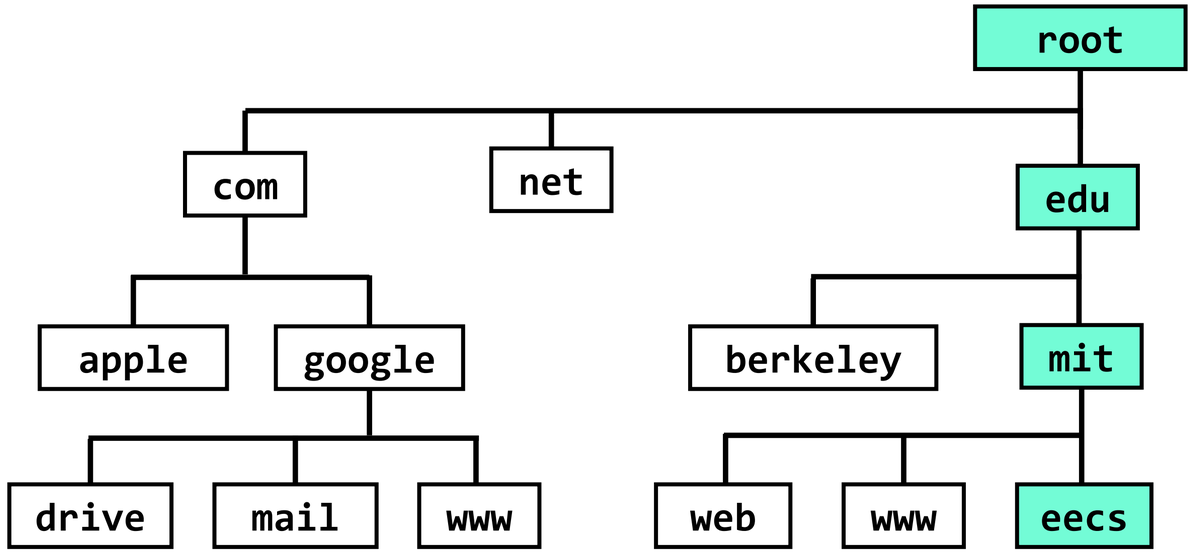
performance issue: this is a *lot* of queries, especially to the root server

reliability issue: what happens when a nameserver fails or (**security issue**) is attacked?

naming case study:

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)

the **look-up algorithm** has to scale to the size of the Internet, while dealing with constant updates and issues of delegation



a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings

performance issue: this is a *lot* of queries, especially to the root server

reliability issue: what happens when a nameserver fails or (**security issue**) is attacked?

control issue: who should manage the root server?

bonus case study:

Course 6 subject numbers, all of which changed recently (to many people's annoyance, but to my delight)

6.031: Elements of Software Construction
6.032: *didn't exist*
6.033: Computer Systems Engineering
6.034: Artificial Intelligence
6.035: Computer Language Engineering
6.036: Introduction to Machine Learning
6.037: Structure and Interpretation of Computer Programs
6.038: Representation & Inference in AI
6.039: Operating Systems Engineering

bonus case study:

Course 6 subject numbers, all of which changed recently (to many people's annoyance, but to my delight)

6.031: Elements of Software Construction
6.032: *didn't exist*
6.033: Computer Systems Engineering
6.034: Artificial Intelligence
6.035: Computer Language Engineering
6.036: Introduction to Machine Learning
6.037: Structure and Interpretation of Computer Programs
6.038: Representation & Inference in AI
6.039: Operating Systems Engineering

systems subjects interspersed with software engineering, AI, etc.

previous scheme: subject number alone gave you very little information about the topic of the class

bonus case study:

Course 6 subject numbers, all of which changed recently (to many people's annoyance, but to my delight)

6.1800: Computer Systems Engineering
6.1810: Operating Systems Engineering
6.1820: Mobile and Sensor Computing
6.1830: *doesn't exist*
6.1840: *doesn't exist*
6.1850: Computer Systems and Society
…
6.5810: Operating System Engineering (G)
6.5820: Computer Networks (G)

this is intentional, to allow for new subjects

current scheme: subject number tells you something about the topic (nearby numbers are in similar areas of EECS)

bonus case study:

Course 6 subject numbers, all of which changed recently (to many people's annoyance, but to my delight)

6.1800: Computer Systems Engineering
6.1810: Operating Systems Engineering
6.1820: Mobile and Sensor Computing
6.1830: *doesn't exist*
6.1840: *doesn't exist*
6.1850: Computer Systems and Society
…
6.5810: Operating System Engineering (G)
6.5820: Computer Networks (G)

this is intentional, to allow for new subjects

current scheme: subject number tells you something about the topic (nearby numbers are in similar areas of EECS)

why four digits after the 6? the registrar doesn't allow a department to reuse a permanent subject number for a new class within five years. solution: create a new three-digit based scheme, append 0's to make then four digits. after five years, revert back to three digits.

modularity and **abstraction** mitigate complexity.
a **client/server model** allow us to enforce modularity by putting modules on physically separate machines.

modularity and **abstraction** mitigate complexity.
a **client/server model** allow us to enforce modularity by putting modules on physically separate machines.

naming is what allows modules to interact, and can help us achieve other goals through properties such as indirection, user-friendliness, etc.

modularity and **abstraction** mitigate complexity. a **client/server model** allow us to enforce modularity by putting modules on physically separate machines.

naming is what allows modules to interact, and can help us achieve other goals through properties such as indirection, user-friendliness, etc.

the **domain name system** is a great case-study in naming, and also illustrates principles such as **hierarchy**, **scalability**, **delegation**, and **decentralization**

modularity and **abstraction** mitigate complexity. a **client/server model** allow us to enforce modularity by putting modules on physically separate machines.

naming is what allows modules to interact, and can help us achieve other goals through properties such as indirection, user-friendliness, etc.

the **domain name system** is a great case-study in naming, and also illustrates principles such as **hierarchy**, **scalability**, **delegation**, and **decentralization**

the example you saw in lecture was a fairly basic one; you will talk more about DNS's performance enhancements in recitation tomorrow, which change how some (many) DNS queries are resolved

modularity and **abstraction** mitigate complexity. a **client/server model** allow us to enforce modularity by putting modules on physically separate machines.

naming is what allows modules to interact, and can help us achieve other goals through properties such as indirection, user-friendliness, etc.

the **domain name system** is a great case-study in naming, and also illustrates principles such as **hierarchy**, **scalability**, **delegation**, and **decentralization**

and client/server models, and (tomorrow) caching, and (in May) security…

the example you saw in lecture was a fairly basic one; you will talk more about DNS's performance enhancements in recitation tomorrow, which change how some (many) DNS queries are resolved