CSE 224:
OVERVIEW AND INTRODUCTION

George Porter
Jan 10, 2023

UC San Diego

ATTRIBUTION

UC San Diego

TODO

1. Go to Canvas and take the “Onboarding Survey”
2. Start project 0 (due Jan 17)

WELCOME!

CSE 224: NETWORKED SYSTEMS

- Add networking support to software
  - Between two computers
  - Between computer and datacenter (“The Cloud”)
- Develop software that is:
  - Scalable (handles 100s of M to 1+ billion users)
  - Fault-tolerant (survives failures)
  - Evolvable (how to update services without making them unavailable to end users)

MODERN SOFTWARE INCREASINGLY NETWORKED

Networked services / Backend
(cloud computing)

Application protocols

Endhost / Front-end
(phone, laptop, game console,
electric vehicle, …)

NETWORKED SERVICES DRIVEN BY DATA

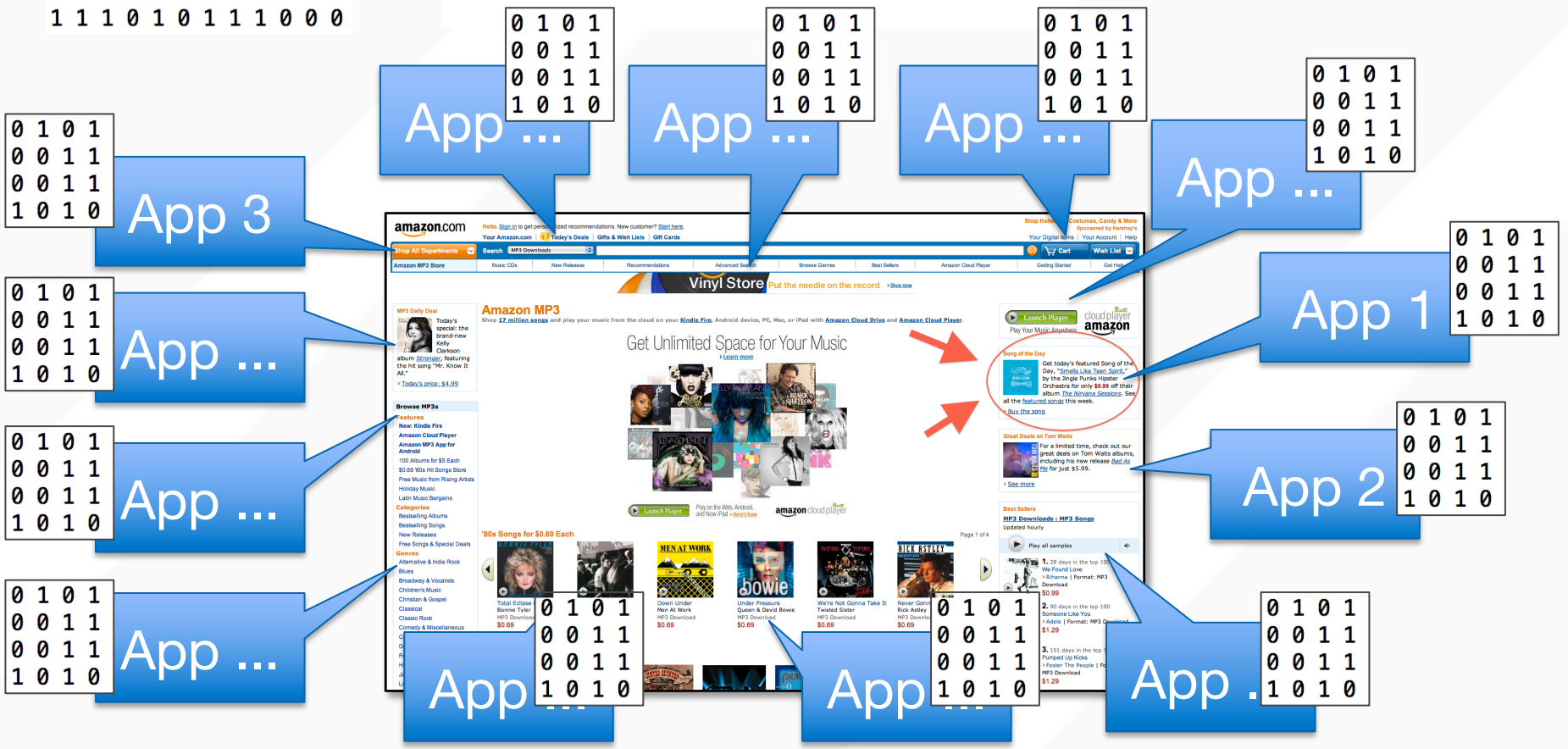
+ amazon.com = Product Recommendations

+ Spotify® = Custom Stations

+ Google = Personalized Search

DATA-DRIVEN, PER-USER CUSTOMIZATION + ML

Data + amazon.com = Product Recommendations



MAJOR THEMES OF THE COURSE

- Programming abstractions for communicating over the Internet through various network protocols
- Naming and indexing to find services and connect clients with servers (or clients with other clients)
- Managing scale; scale-out design
- Replicating and updating “mutable” data over the network
- Replicating and caching “immutable” data over the network (think Netflix, Disney+, Youtube, etc)
- Accessing and managing networked storage
- Managing fault tolerance

HOW CAN YOU WRITE SOFTWARE THAT WORKS DESPITE ADVANCEMENTS IN UNDERLYING TECHNOLOGY?

Think about the first computer you remember using…
- Can we find the oldest example here in class today? The most recent example?

Think about the first network you used (modem? Fiber optics? Mobile network?)
- Can we find the oldest example here in class today? The most recent example?

Discuss with the 3-4 people nearest you for 2 minutes and let's find out!

UC San Diego

THINK ABOUT HOW ONLINE NETWORK SERVICES HAVE CHANGED OVER THE PAST 20-ISH YEARS…

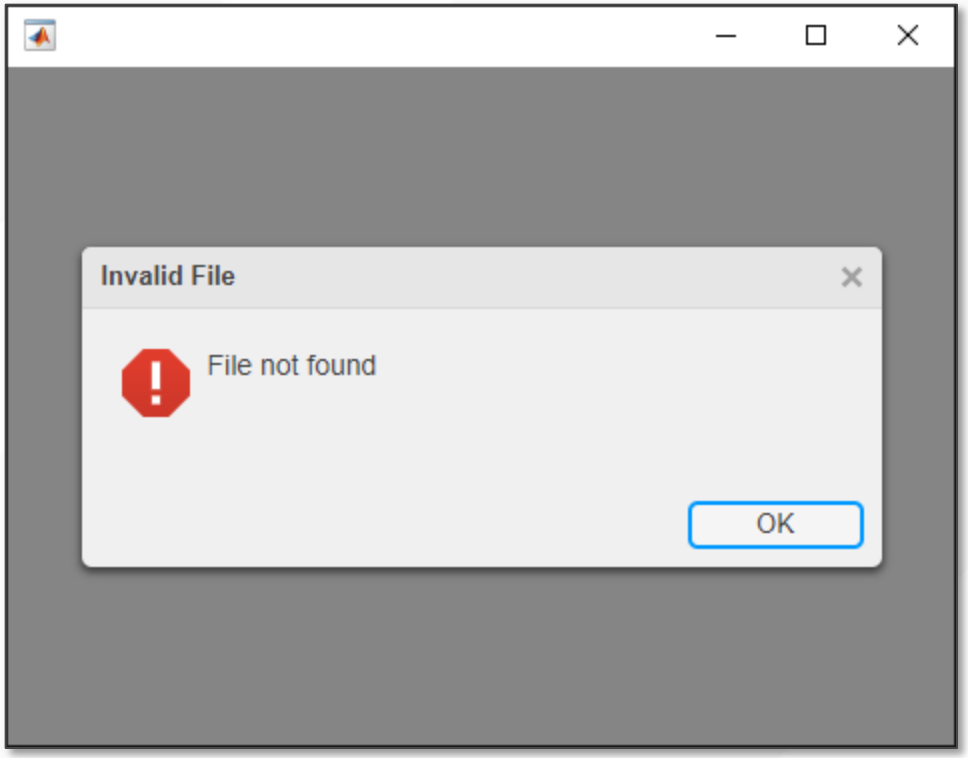
GOOGLE (1998)


Google!
BETA
Search the web using Google!
Google Search
I'm feeling lucky
Special Searches
Stanford Search
Linux Search
Help!
About Google!
Company Info
Google! Logos
Get Google!
updates monthly:
your e-mail
Subscribe
Archive
Copyright ©1998 Google Inc.

NETFLIX (2002)

TWITTER (2007)

YOUTUBE (2004)



YOUTUBE (2005)

FACEBOOK (2004)

SURVEY

- When was the web protocol created? When was the first graphical web browser released?

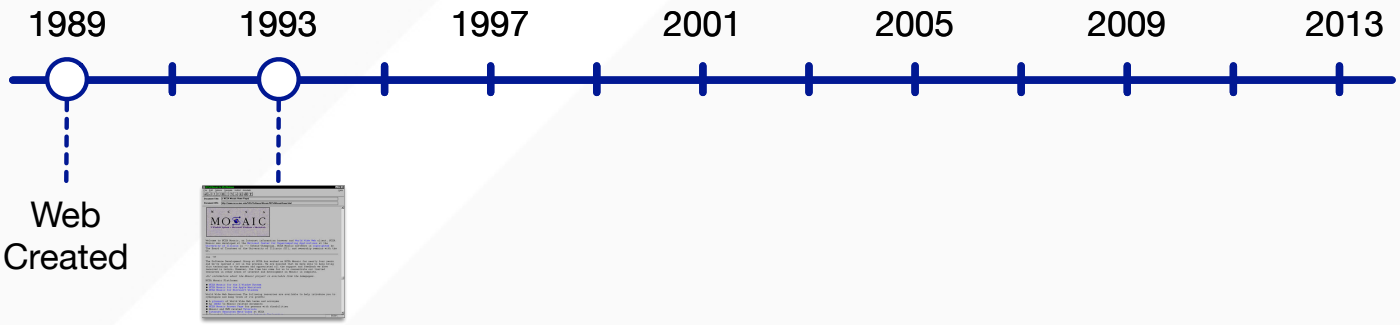
1. 1968 / 1972
2. 1974 / 1976
3. 1989 / 1993
4. 2001 / 2002

UC San Diego

THE DEPLOYMENT OF “THE WEB”

RISE OF THE WEB

RISE OF THE WEB

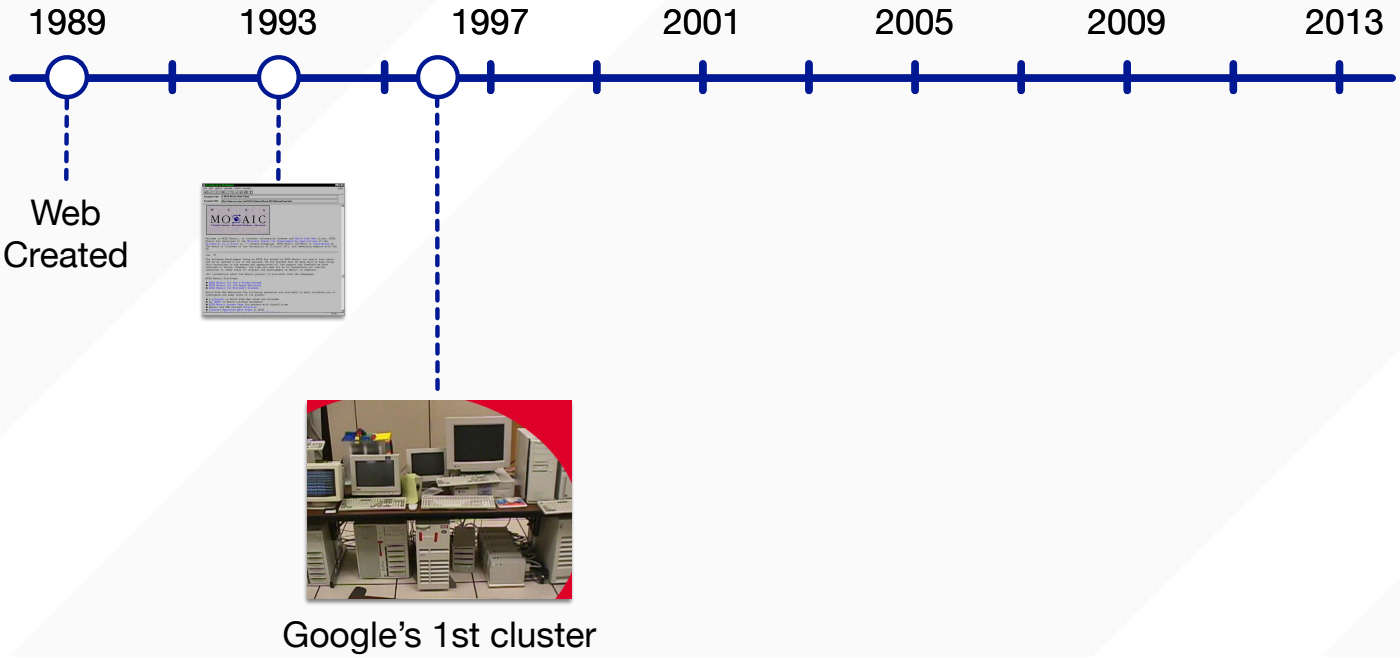


UC San Diego

WHERE DO NETWORK SERVICES EXECUTE?

THE FIRST WEB SERVER (NEXT WORKSTATION, 1991)

THE RISE OF THE "DATACENTER" (AKA CLOUD COMPUTING)



DATACENTERS:
THE HOME OF ALL THIS COMPUTING AND STORAGE

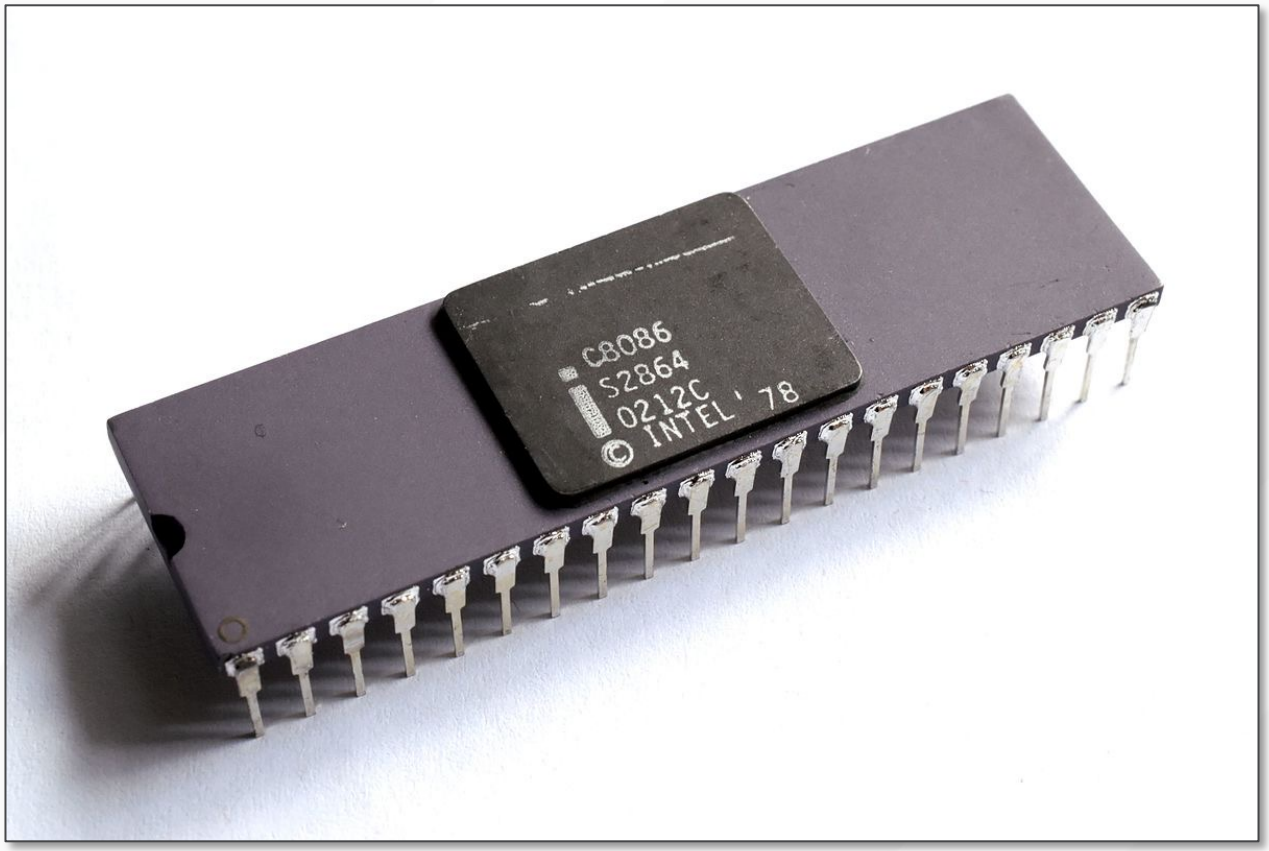
Google 2012

Microsoft


Google


Facebook

HARDWARE HAS EVOLVED AS WELL. STARTING WITH CPUS…

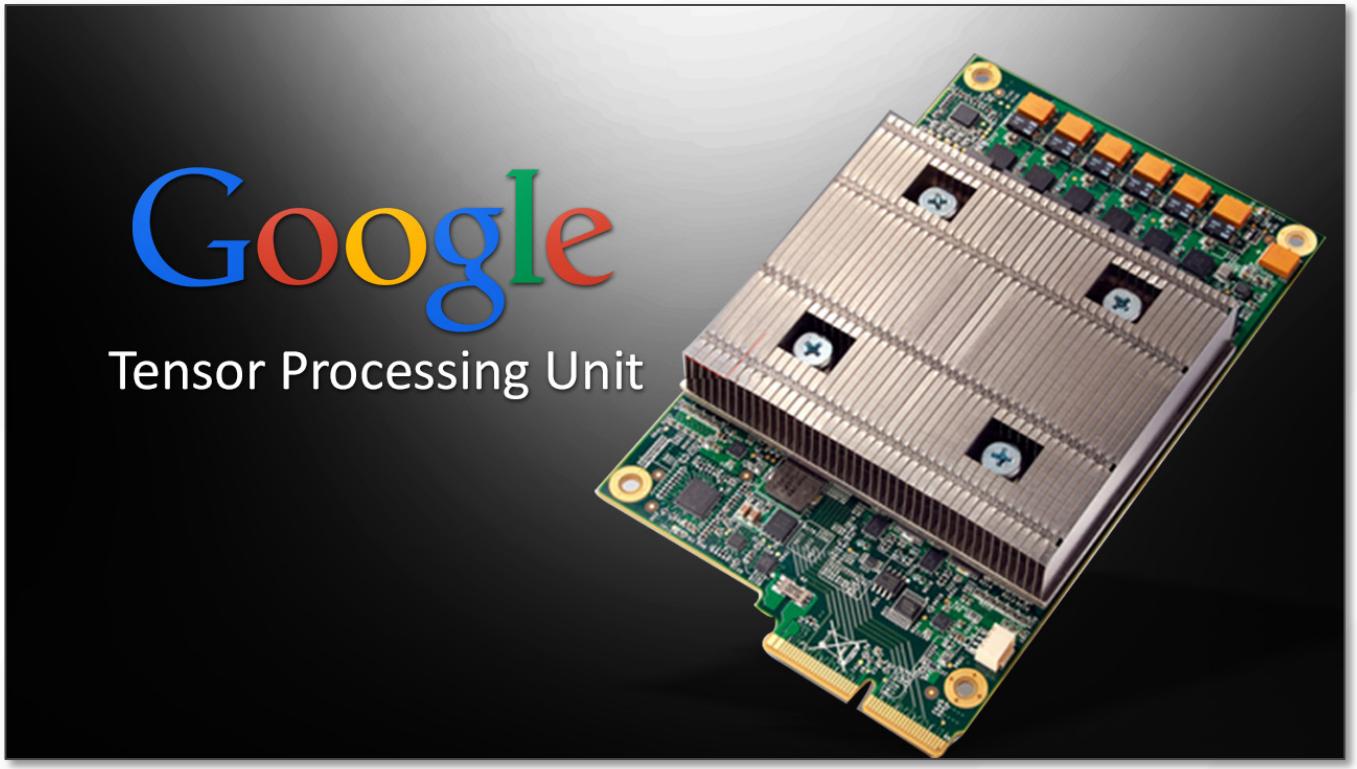


TO GPUS…

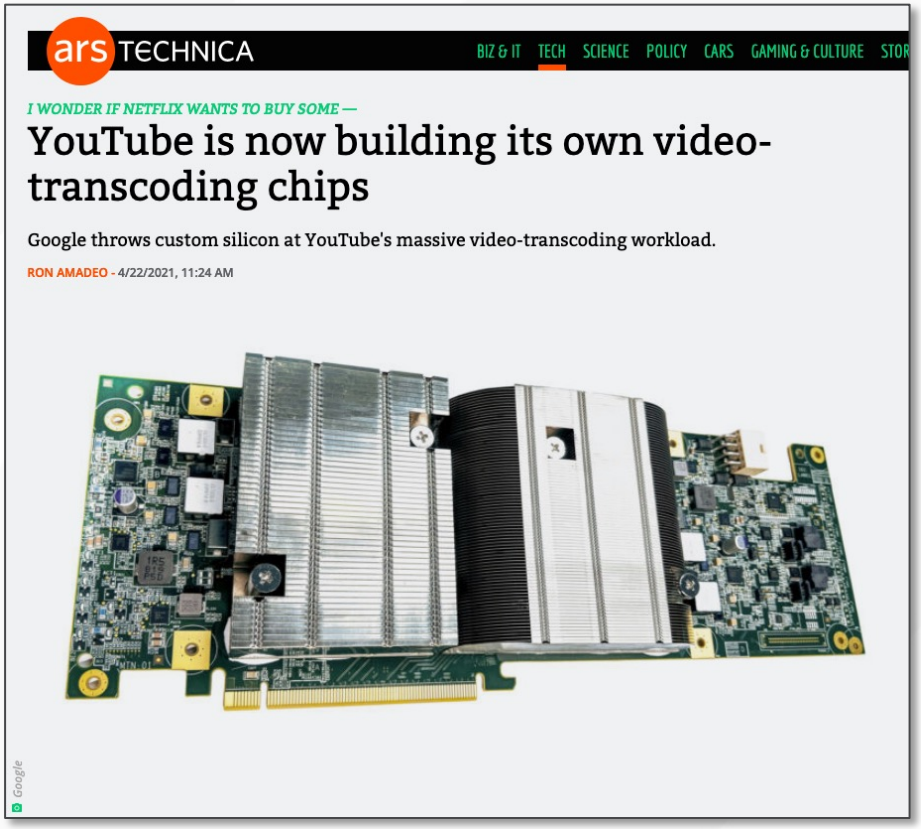

GEFORCE GTX
GEFORCE GTX
GEFORCE GTX
GEFORCE GTX
GEFORCE GTX
GEFORCE GTX
GEFORCE GTX
GEFORCE GTX
GEFORCE GTX

TO PROGRAMMABLE FPGAS…

TO CUSTOM DESIGNED CHIPS



CHUSTOM VIDEO TRANSCODING CHIP



ars TECHNICA

BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE STOR

I WONDER IF NETFLIX WANTS TO BUY SOME —

YouTube is now building its own video-transcoding chips

Google throws custom silicon at YouTube's massive video-transcoding workload.

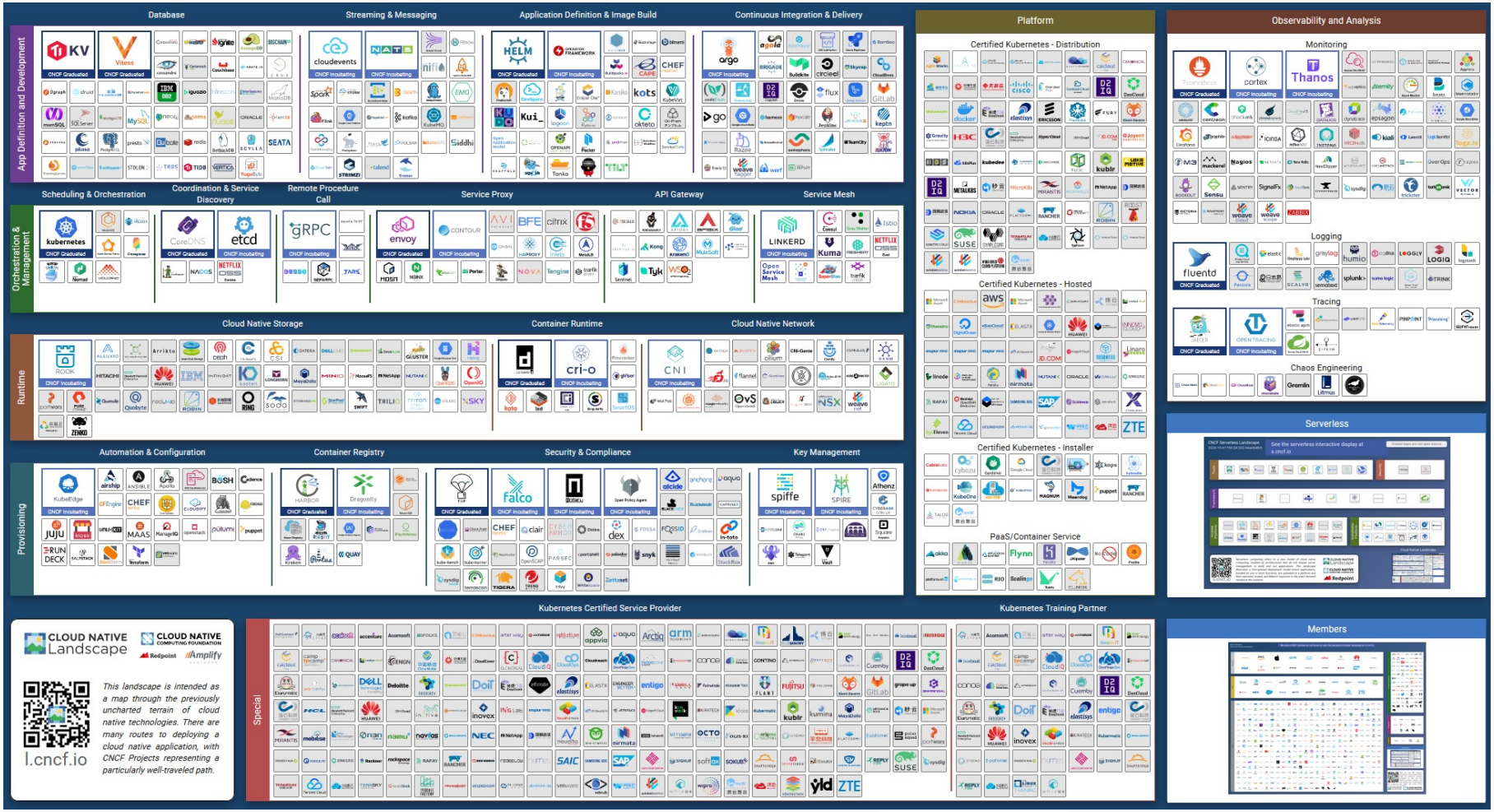
RON AMADEO - 4/22/2021, 11:24 AM

Google

CLUSTERS OF CUSTOM ASICS FOR AI/MACHINE LEARNING

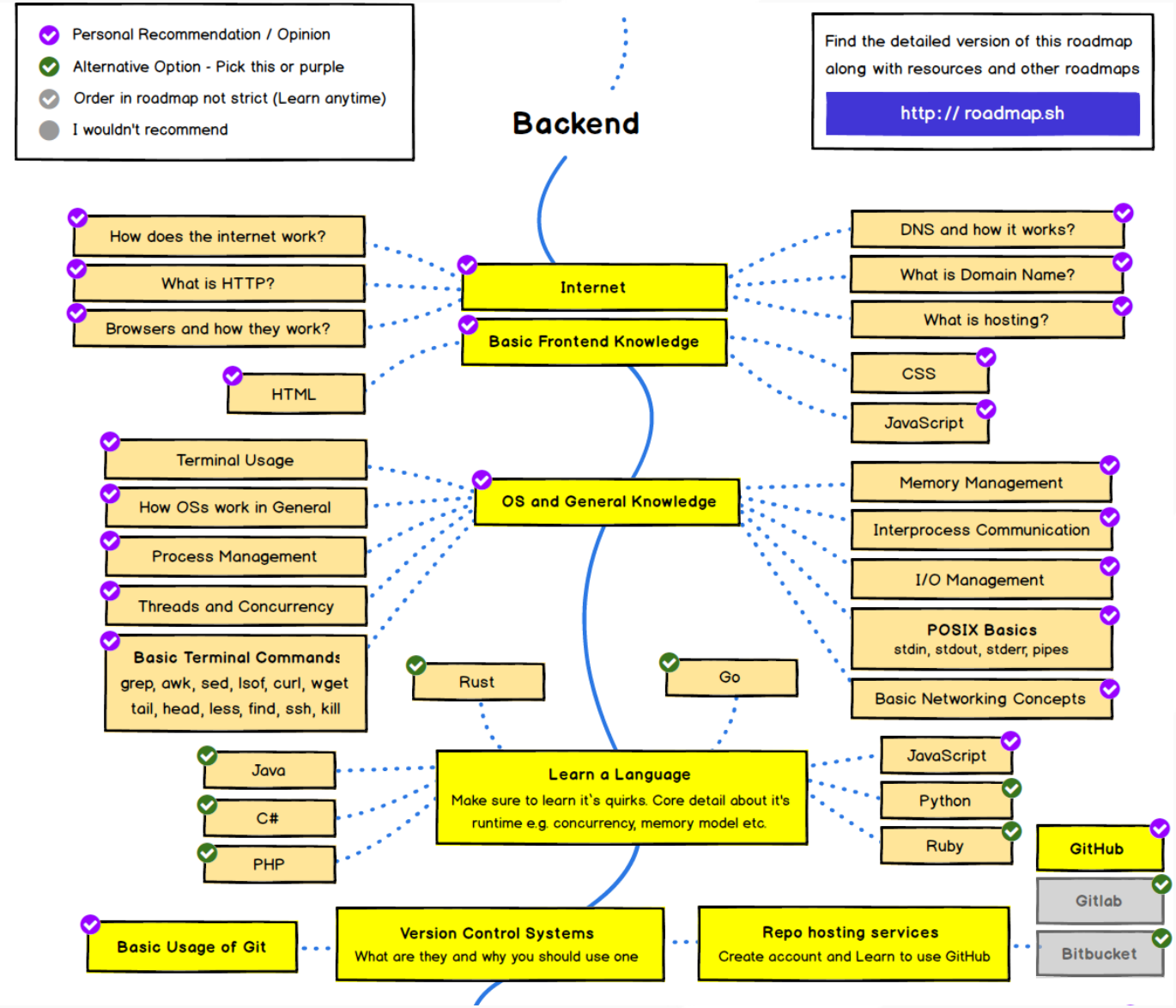
Source: google.com

FULL CLOUD NATIVE LANDSCAPE

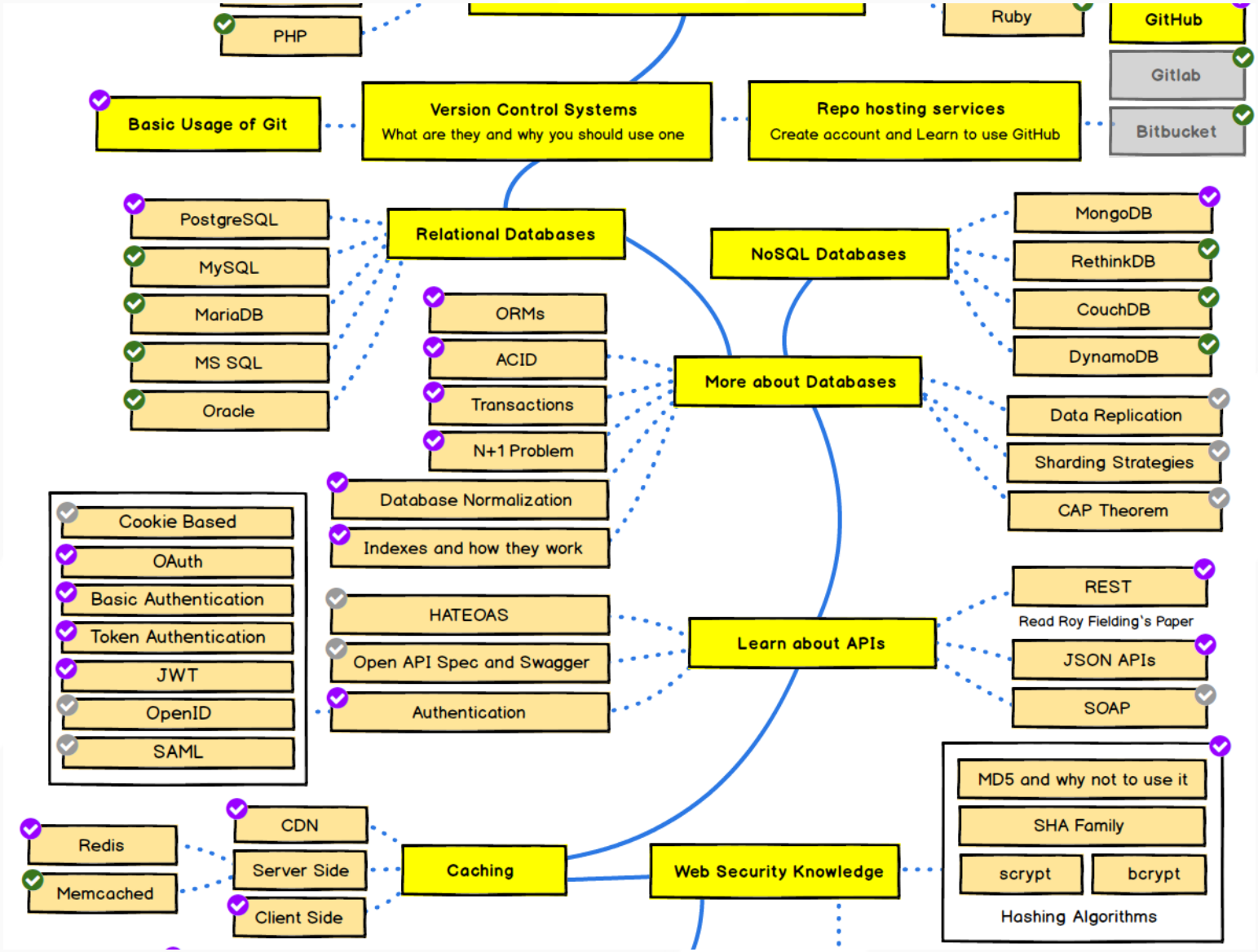


CLOUD NATIVE LANDSCAPE IN A 10-WEEK QUARTER

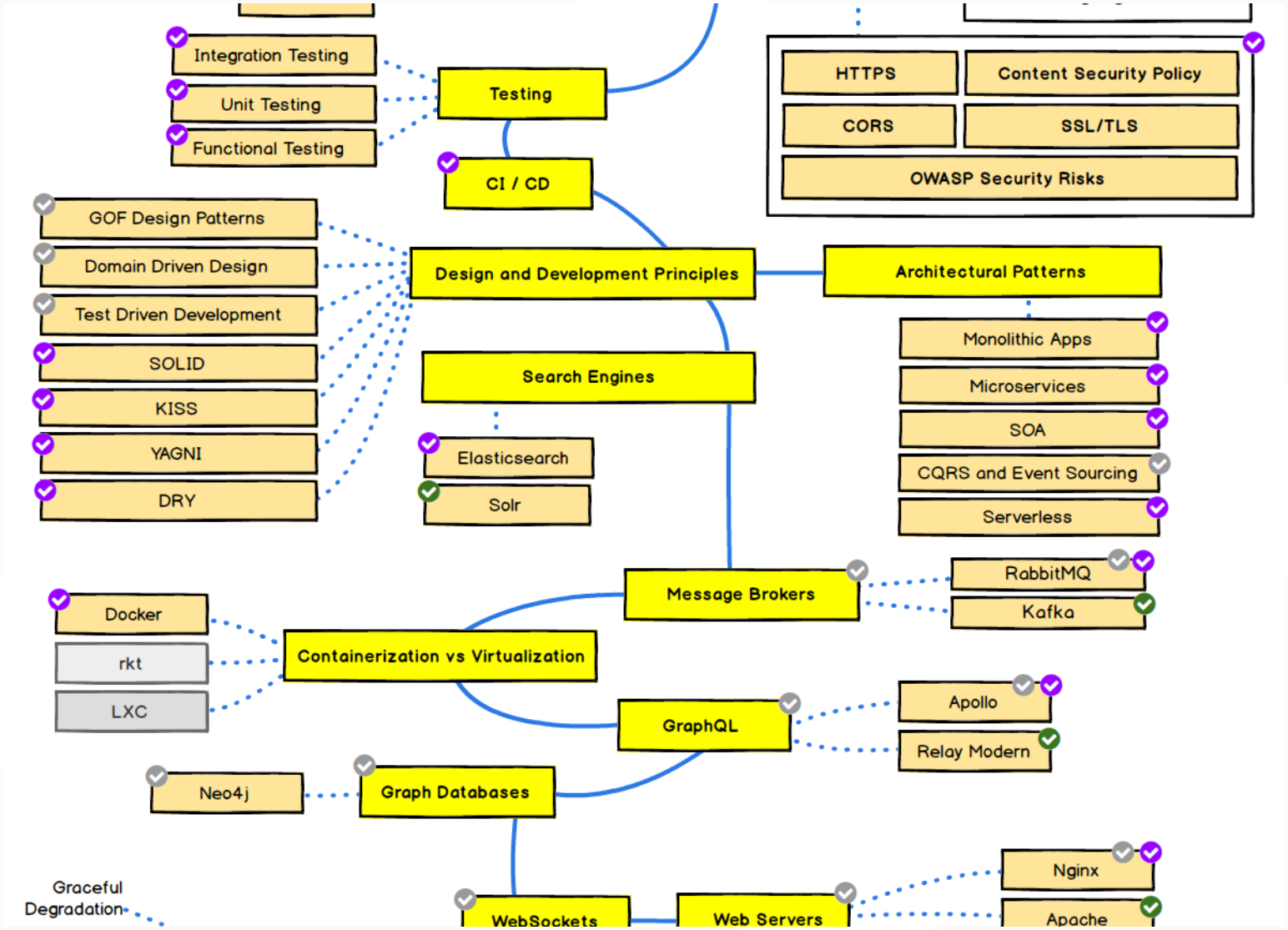
BACKEND DEVELOPMENT ROADMAP



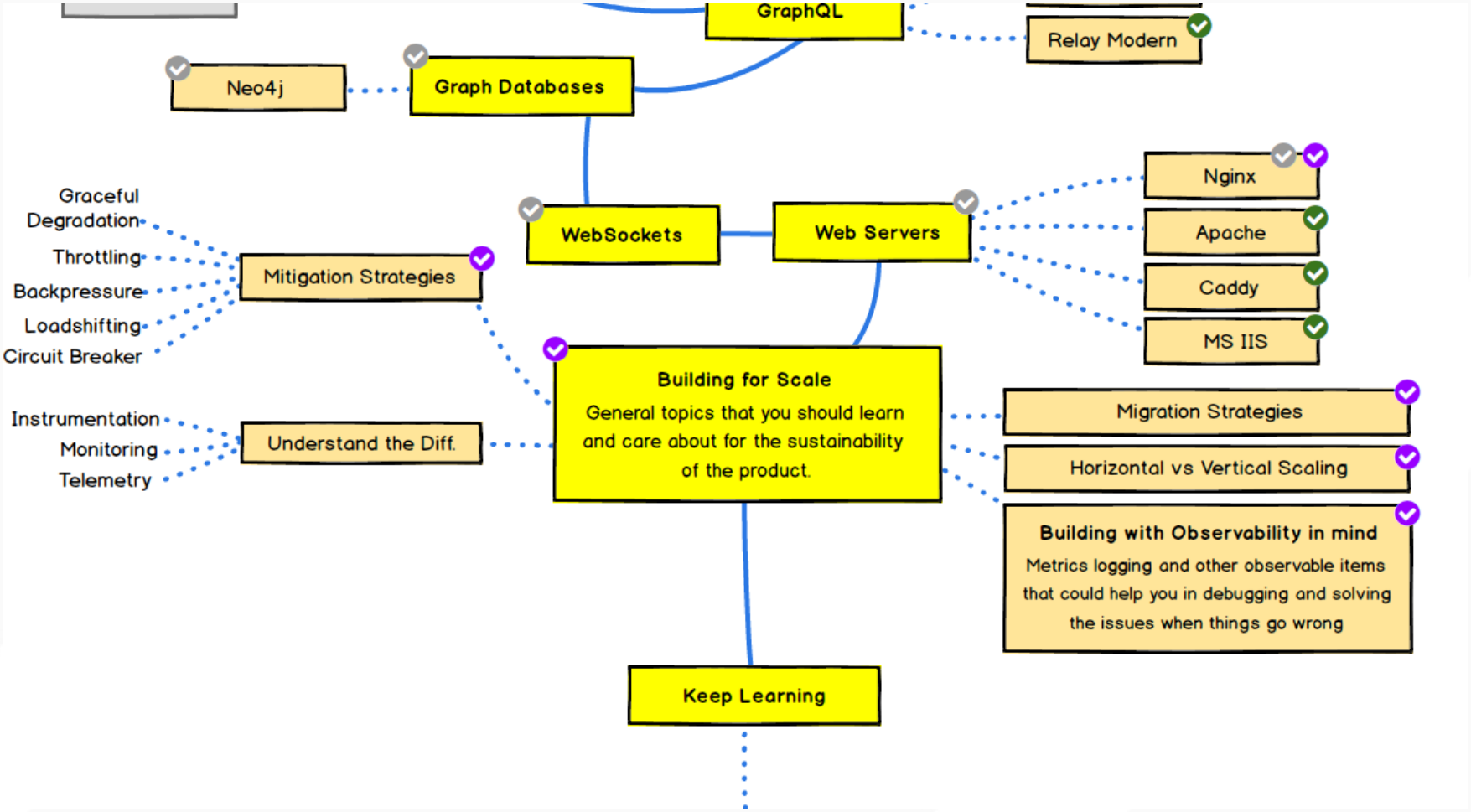
BACKEND DEVELOPMENT ROADMAP



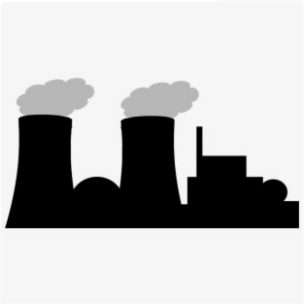
BACKEND DEVELOPMENT ROADMAP



BACKEND DEVELOPMENT ROADMAP



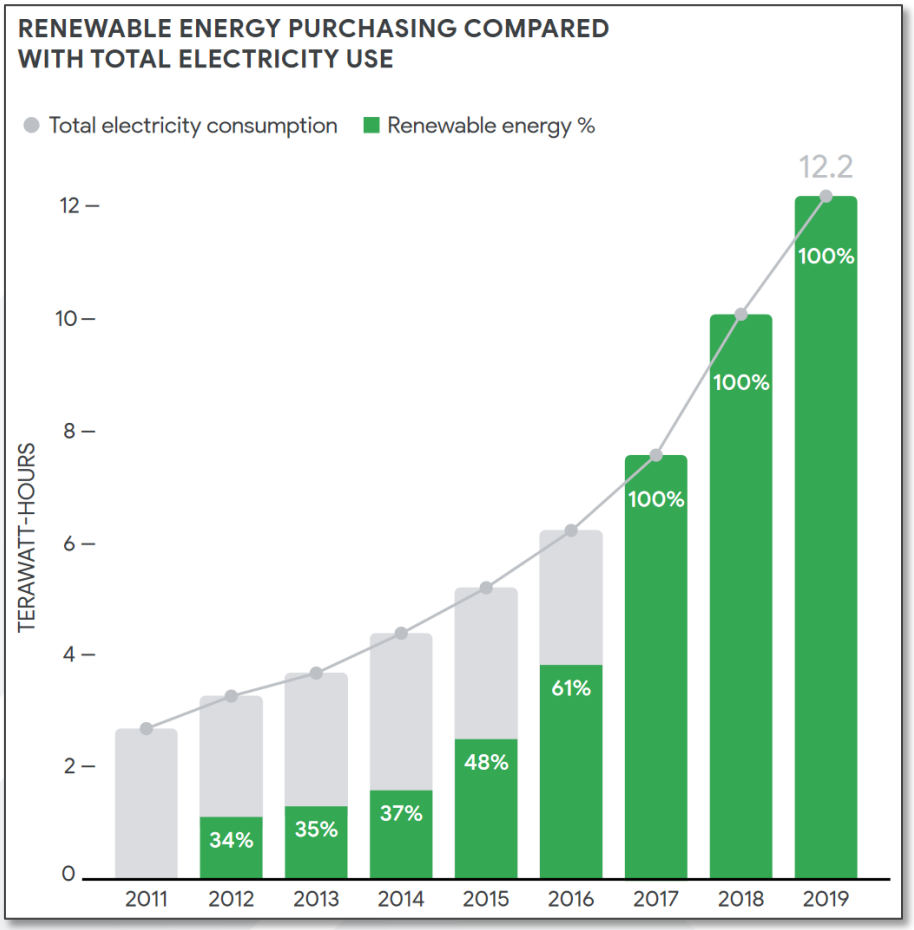
THE ENVIRONMENTAL IMPACT OF CLOUD COMPUTING



1. LBNL, 2013
2. NRDC report

- Carbon/energy footprint:
  - 1-2% of global energy consumption[1]
  - 140 billion kWh (50 power plants)[2]
  - 100 metric tons of carbon pollution per year[2]

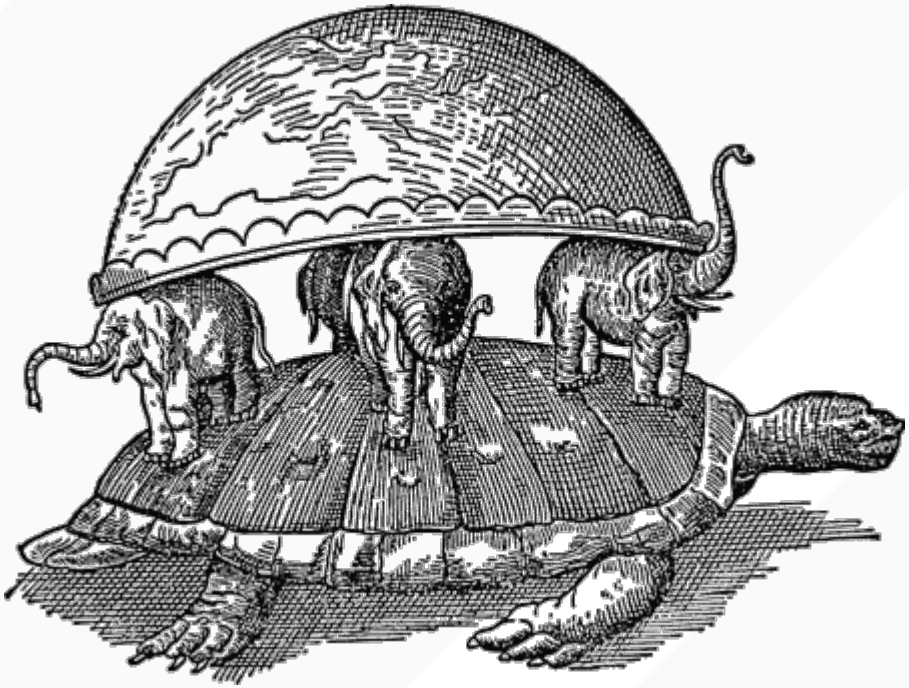
Google's energy footprint



SCALING ACROSS TECHNOLOGY IMPROVEMENTS

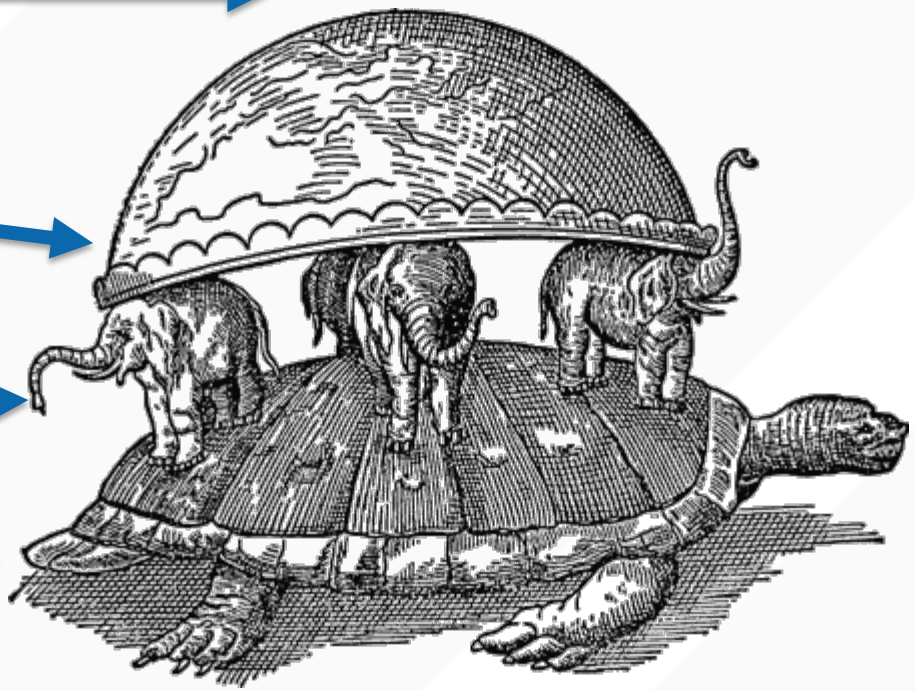
- Network primitives are designed to scale
- Techniques we learn are directly applicable to global-scale services like Google, Facebook, …
- Your projects will be tested in small scale
  - Yet could scale immensely with minimal to no modifications

HOW TO BUILD SUCH LARGE SYSTEMS?



HOW TO BUILD SUCH LARGE SYSTEMS?

- Systems…
- Built on top of abstractions…
- Built on software…
- Built on hardware…



We will cover the software abstractions to enable you to write networked software

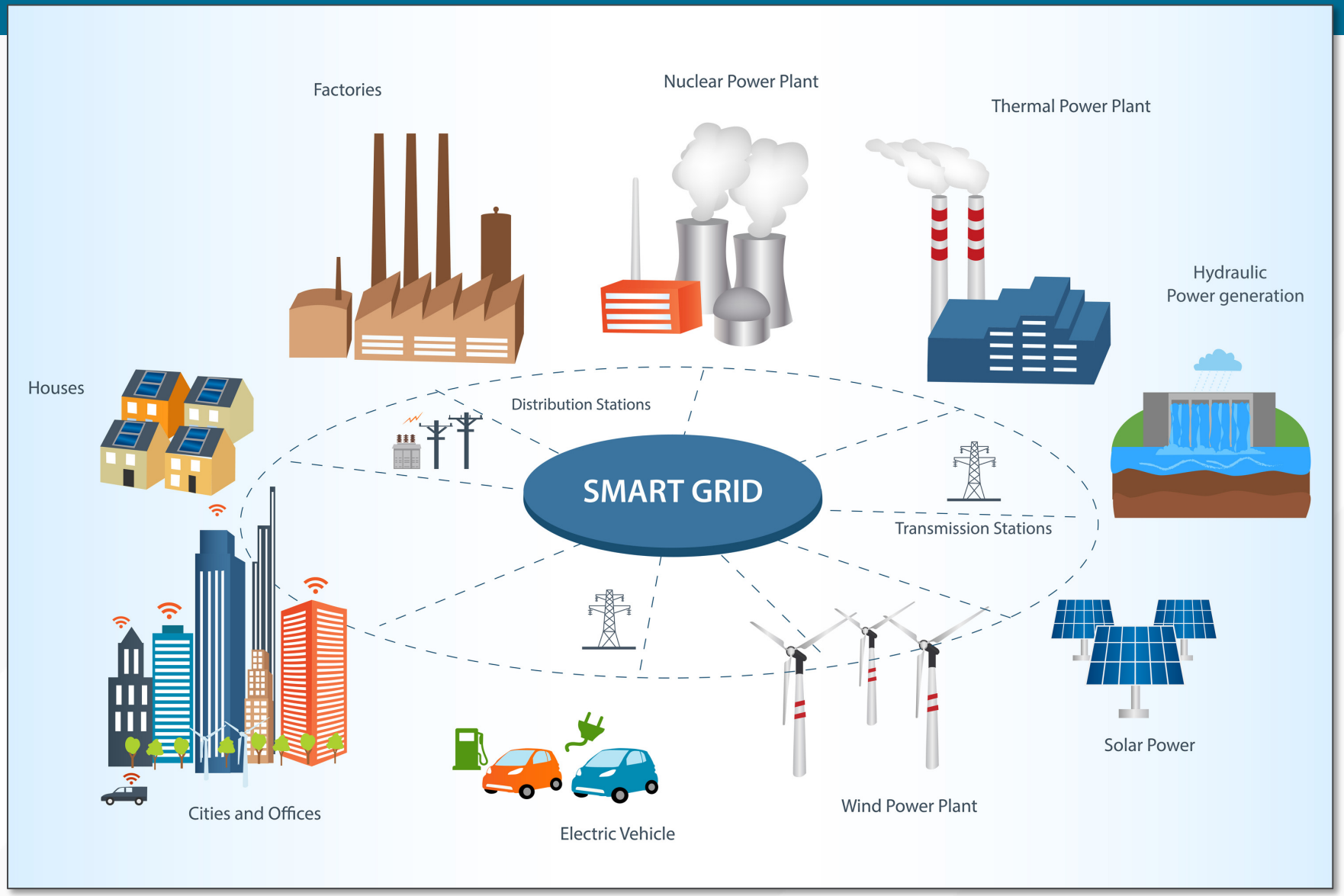
IT’S NOT JUST WEBSITES AND SOCIAL MEDIA THOUGH!

UC San Diego

SELF-DRIVING CARS AND SMART CITIES

https://www.electronicproducts.com/wp-content/uploads/automotive-fig1-automotive-5g-c-v2x-qualcomm.png

SMART CITIES AND SMART GRIDS



https://innovationatwork.ieee.org/wp-content/uploads/2018/04/bigstock-127573223.jpg

CSE 224 VS {221,222A,223B}

- 224: Graduate Networked Systems
  - How to program networked software
  - Socket programming, RPC, protocol design and implementation, consensus and consistency, security, TLS, …
  - Designed as a *broad survey* of systems thinking
  - Learn through hands-on, programming-based projects
- 224 Target audience:
  - MS “comps” students and BS/MS students
  - Non-systems MS “thesis” and non-systems Ph.D. students
- Note:
  - Cannot receive credit for both 124 and 224

- Research-focused depth sequence
  - 221: Operating Systems
  - 222A: Networking
  - 223B: Distributed systems theory
  - Deep dives into peer-reviewed literature
  - Learn through close readings and in-class discussion of 4 research papers per week
- 221/222A/223B Target audience:
  - Systems MS ”thesis” and Systems Ph.D students

THE CHALLENGE OF NETWORKING

- CS undergraduate curricula includes:
  - Algorithms
  - Programming languages
  - Architecture
  - Data structures
  - Etc...
- How does the network change each of these areas?

RESOURCES

- Website
  - https://canvas.ucsd.edu/courses/43955
  - Gradebook, links to assignments + deadlines, PDFs of lecture slides, in-class demos and exercises
- Piazza discussion board (linked off Canvas)
- Github (for managing your projects)
- Gradescope (for submitting your projects)
- Two books
- TA discussion section (1x week)

TEACHING ASSISTANTS (PAGE 1)

	Aditya Barsainya
	Qihuang Chen

	Xiuqi Chen
	Tanmay Gujar

TEACHING ASSISTANTS (PAGE 2)

Linfang He

Saketh Khandavalli

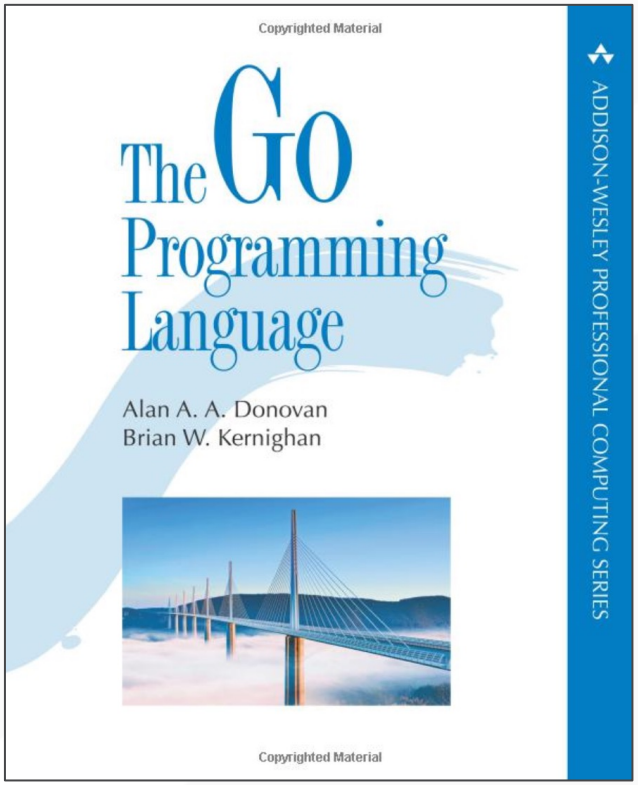
Amanda Tomlinson

Abhishek Vijeev

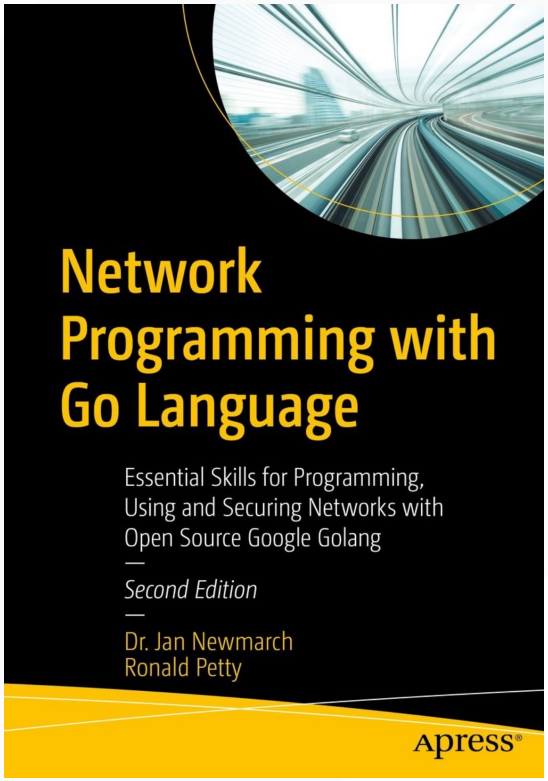
CLASS MEETINGS

- Mostly putting the material that you read into context
- Live coding demos, activities, some “mini lectures” on algorithms, protocols, etc.
- You are responsible for everything that happens during class
  - Will podcast, but can’t guarantee that system works flawlessly
- Will be asking for feedback on what works and what doesn’t work a lot during the class

BOOKS



Free if accessed through the UCSD library



Free if accessed through the UCSD library

PROGRAMMING SKILLS FOR THIS CLASS

- We'll be using the "Go" language
  - golang.org
  - Designed at Google in 2007
  - Goals: improve programming productivity in an era of multicore, networked machines, and large codebases
  - Kernighan (of 'C' fame) co-created
- Why?
  - Simple, readable, no mem allocation (similar to Python)
  - High-performance networking
  - Concurrency/parallelism
  - Static typing and efficient runtime
  - Industry-quality and deployed at massive scale

CLASS ROADMAP / PROJECTS / GRADING

1. Pre-lecture review question sets [5%]
2. Projects [60%]
    1. [5%] Single-node sort
    2. [10%] Distributed network sort
    3. [15%] Build your own web server
    4. [10%] SurfStore “Dropbox clone”
    5. [5%] Scaled-out SurfStore backend
    6. [15%] Fault-toleranct SurfStore backend
3. Exams [35%]
    1. [15%] Midterm (Thu Feb 9 during class time)
    2. [20%] Final exam (Tue Mar 21, 3-6pm)

DEFAULT UCSD GRADING SCHEME

View/Edit Grading Scheme

eGrades

Select Another Scheme

Name:	Range:	
A+	100 %	to 97.0%
A	< 97.0 %	to 94.0%
A-	< 94.0 %	to 90.0%
B+	< 90.0 %	to 87.0%
B	< 87.0 %	to 84.0%
B-	< 84.0 %	to 80.0%
C+	< 80.0 %	to 77.0%
C	< 77.0 %	to 74.0%
C-	< 74.0 %	to 70.0%
D	< 70.0 %	to 60.0%
F	< 60.0 %	to 0.0%

COURSE AT A GLANCE

- Basics of networking, sockets API, DNS
- Remote procedure calls w/ Google RPC (gRPC)
- Distributed storage as an application
- Scale-out techniques and methods
- Replicating immutable state via CDNs
- Replicating mutable state with two-phase commit and replicated state machines (+ deep dive on the RAFT protocol)

IMPORTANT CONTEXT

- In this course you'll learn some things in class (mostly tested via the exams), and you'll learn some things by *doing—working on the projects.*
- We'll cover the big themes and high-level ideas in class, but you'll be learning a lot of the details in the projects
  - That's why the projects are NOT designed to be done in one long session or all-nighter—you should work on them a bit each day so you can research what you need to complete them, or to talk to the TAs/myself, etc.
  - Start early—start often!

TAKE THE ONBOARDING SURVEY

- Required by UCSD for some reason or another regarding Federal financial aid
- BUT also, there is a very important question for those of you who need this class to graduate and plan to graduate this term…
  - I'm going to export **more** of you this term, not less
  - While I'm not going to enforce it, you really need to be physically attending class every time, not just relying on podcasts

COMPUTING RESOURCES

- You can use the lab computers in the building, or the “ieng6” servers that can be accessed via ssh
- ssh <username>@ieng6.ucsd.edu

TODO

1. Go to Canvas and take the “Onboarding Survey”
2. Start project 0 (due Jan 17)

UC San Diego

UC San Diego