

CS3.301 Operating Systems and Networks

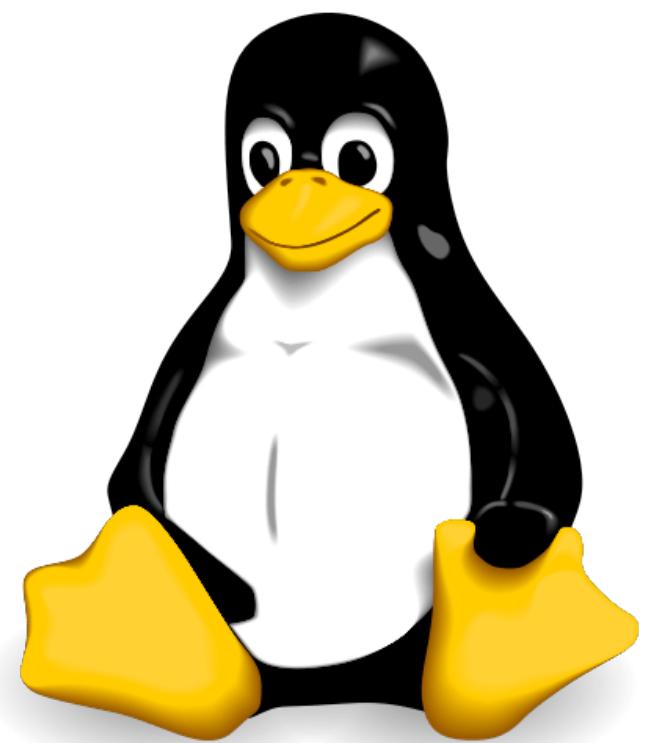
Introduction and Course Overview



Karthik Vaidhyanathan

<https://karthikvaidhyanathan.com>

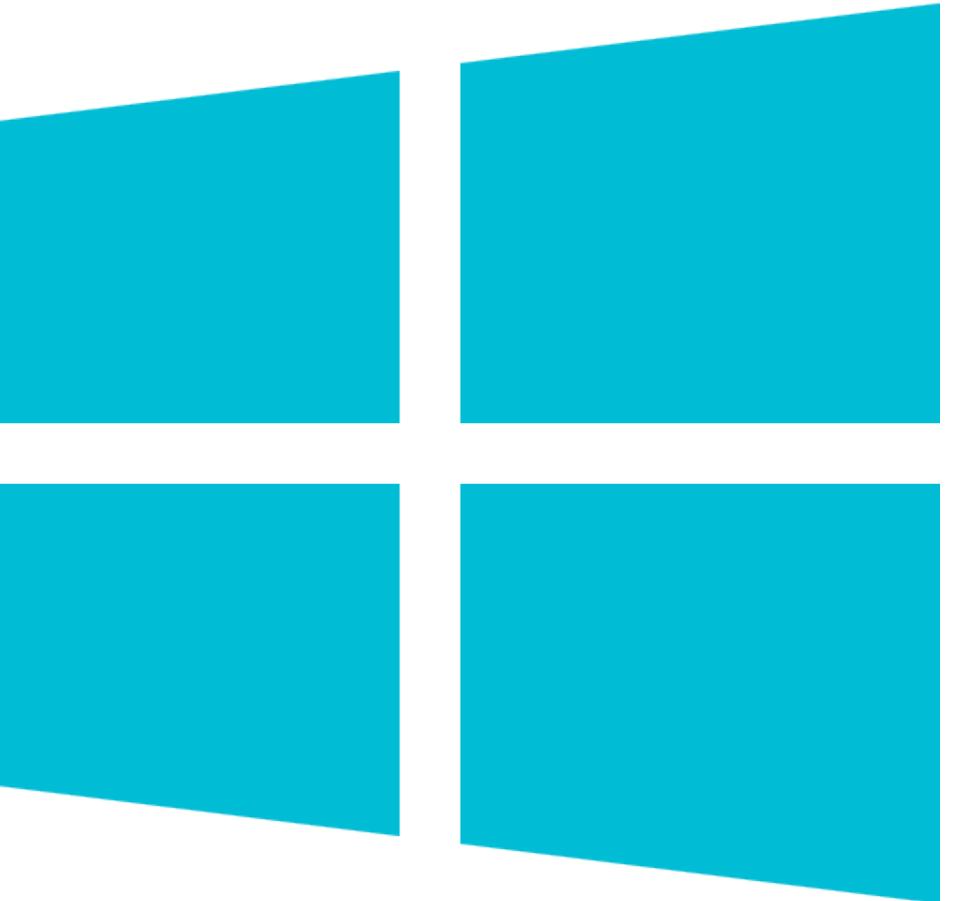
Which OS do you use?



Linux



ChromeOS



Windows



OSX



Wait! what is an OS anyway?

operating system

computing

Print

Cite

Share

Feedback

:

Also known as: OS

Written by [David Hemmendinger](#)

Fact-checked by [The Editors of Encyclopaedia Britannica](#)

Last Updated: [Article History](#)

operating system (OS), program that manages a [computer](#)'s resources, especially the allocation of those resources among other programs. Typical resources include



An operating system (OS) is a software that acts as an intermediary between computer hardware and the user applications. It manages the computer's hardware resources, such as the CPU, memory, disk storage, and peripheral devices like printers and scanners. The OS enables user programs to execute without needing to worry about the specifics of the hardware.

ChatGPT4 Version!

Caution: Use tools but be aware of the uncertainties they bring in!

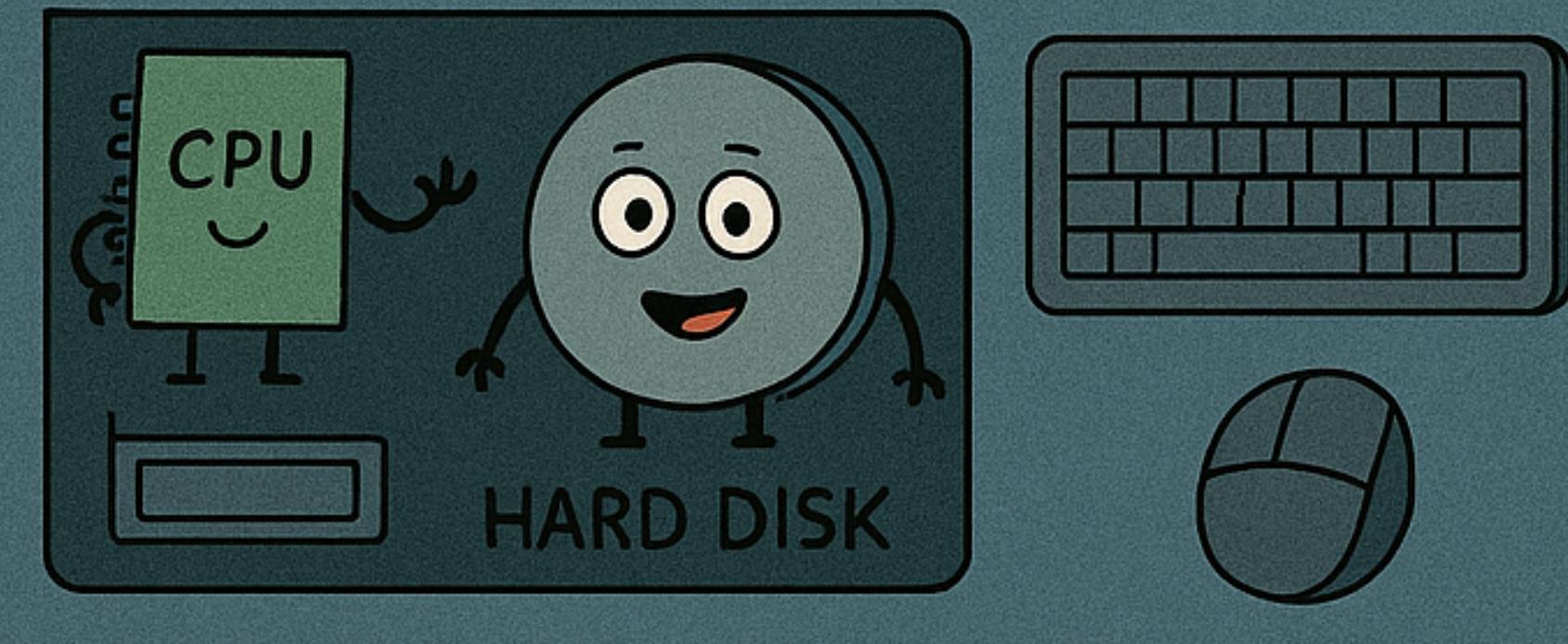


WHAT IS AN OPERATING SYSTEM?

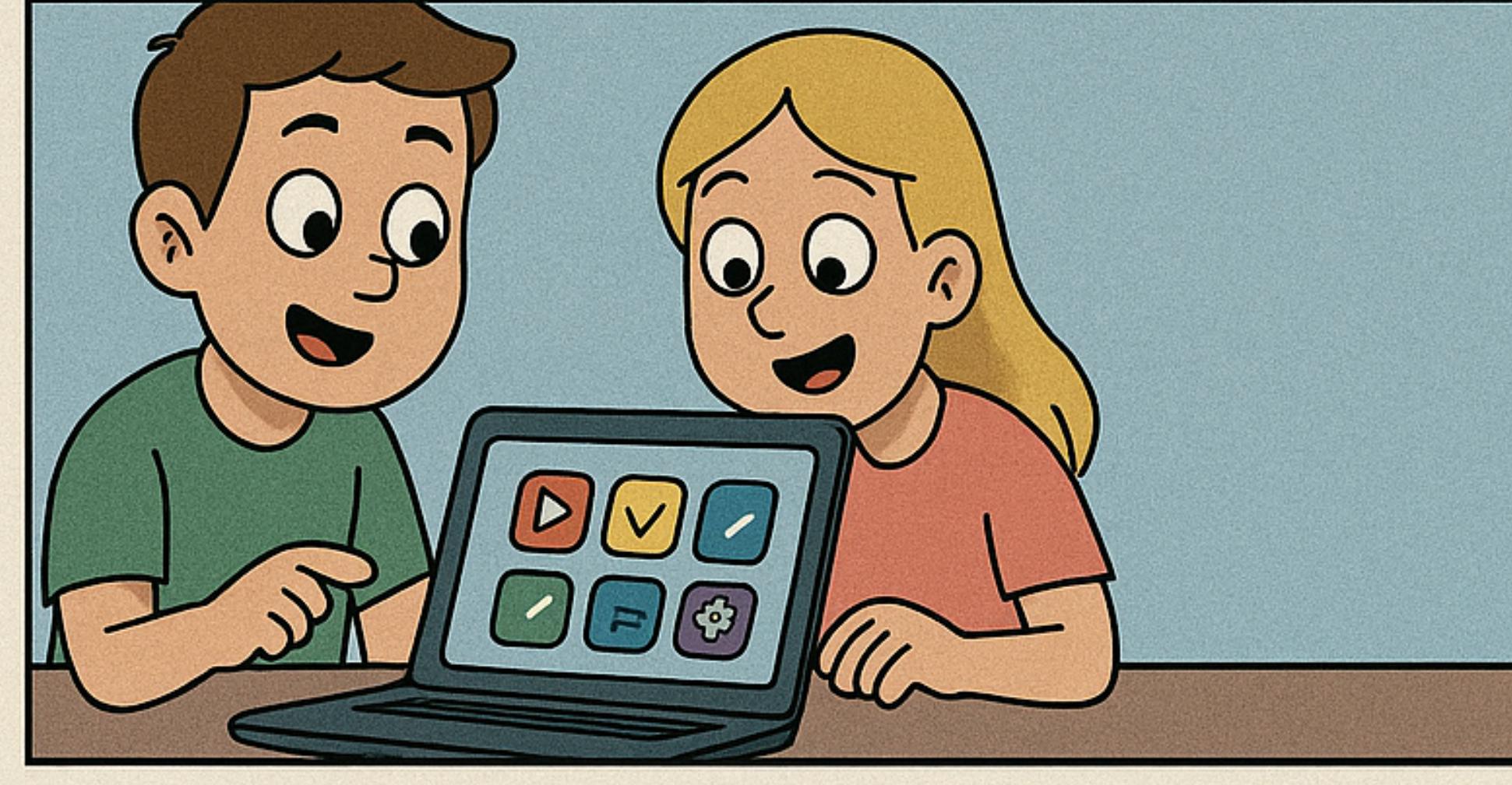
An OS is the software that manages your computer.



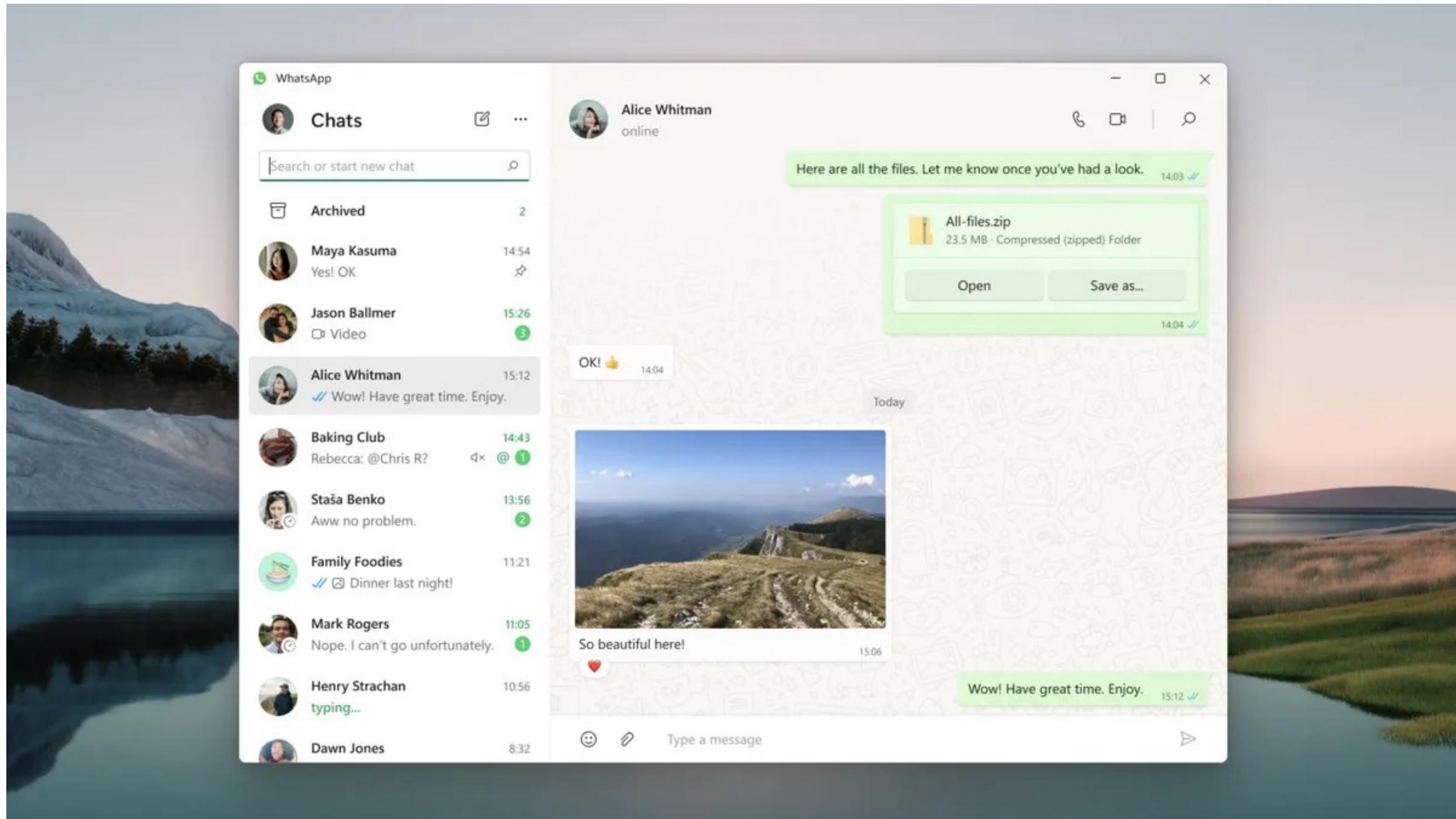
It is the middleman between your hardware...



The OS coordinates all the programs and tasks so that everything runs smoothly!



Develop Whatsapp like Messaging System



- Users authentication
- Send and receive chats
- Send and receive media contents
- Make audio, video calls
- Record audio, video
-



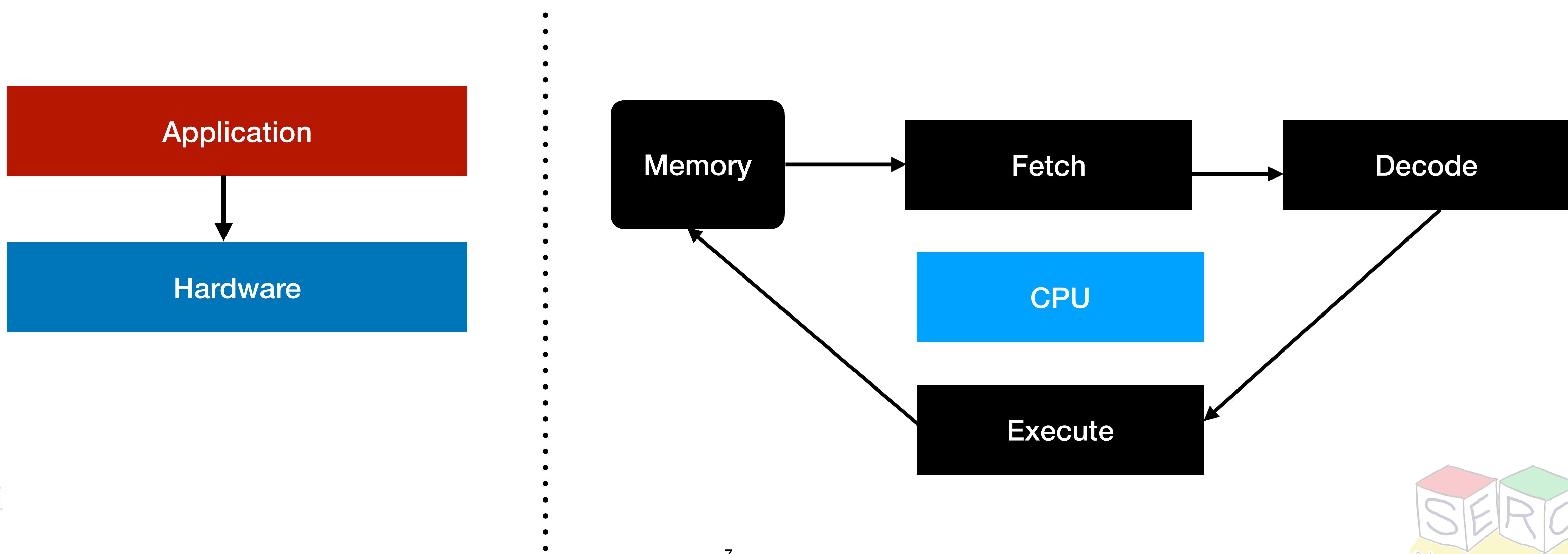
Any Program for that Matter!

- Once your app is build - Compiler comes into action (“c” -> “a.out”) to create the executable (“.dmg”, “.elf”, “.exe”)
- Executable contains: Instructions + data
- Instructions are run by the CPU
- CPU internally consists of registers
 - Program counter or PC
 - Operands of instructions, memory addresses

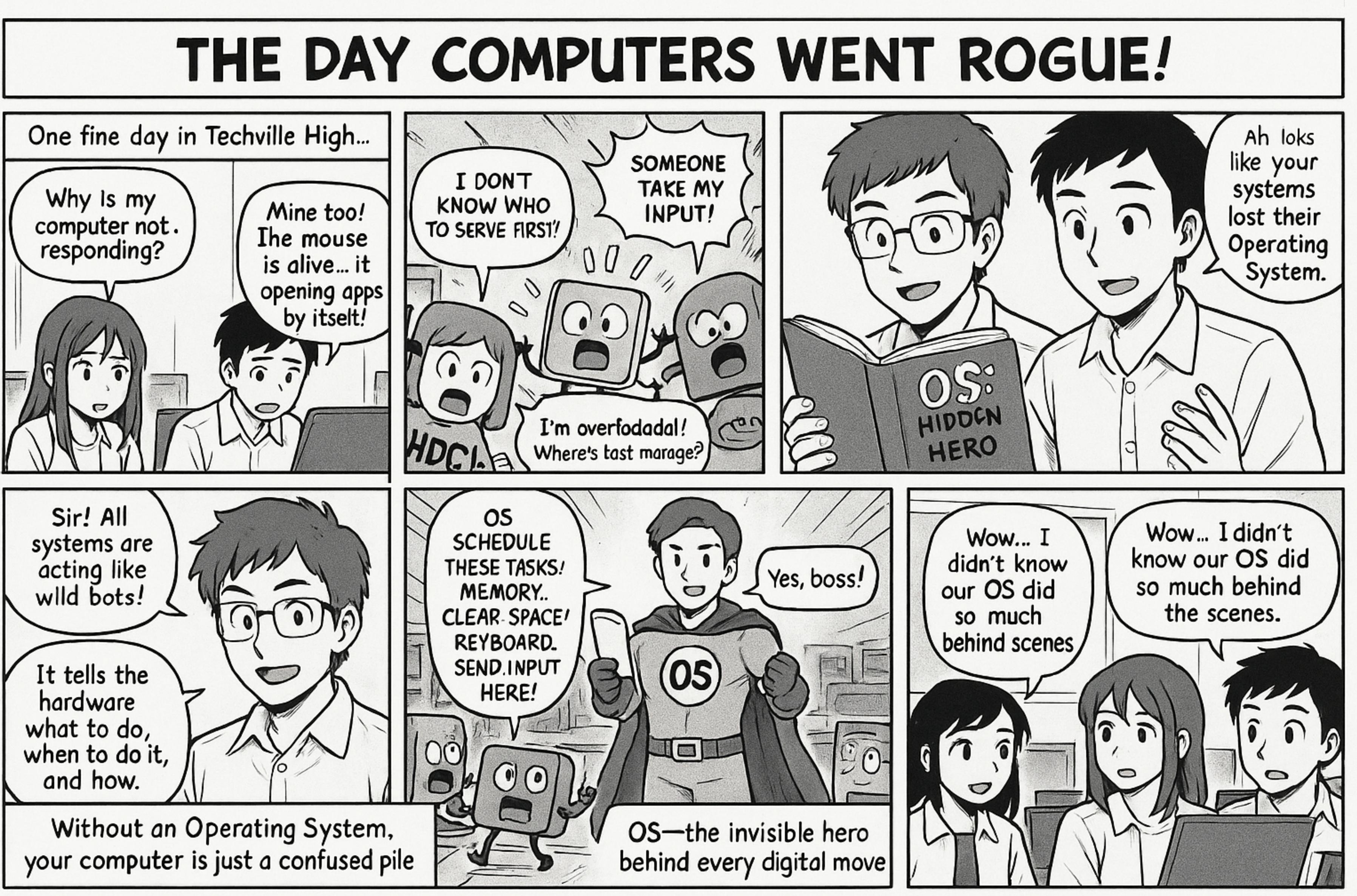


What if there were no OS?

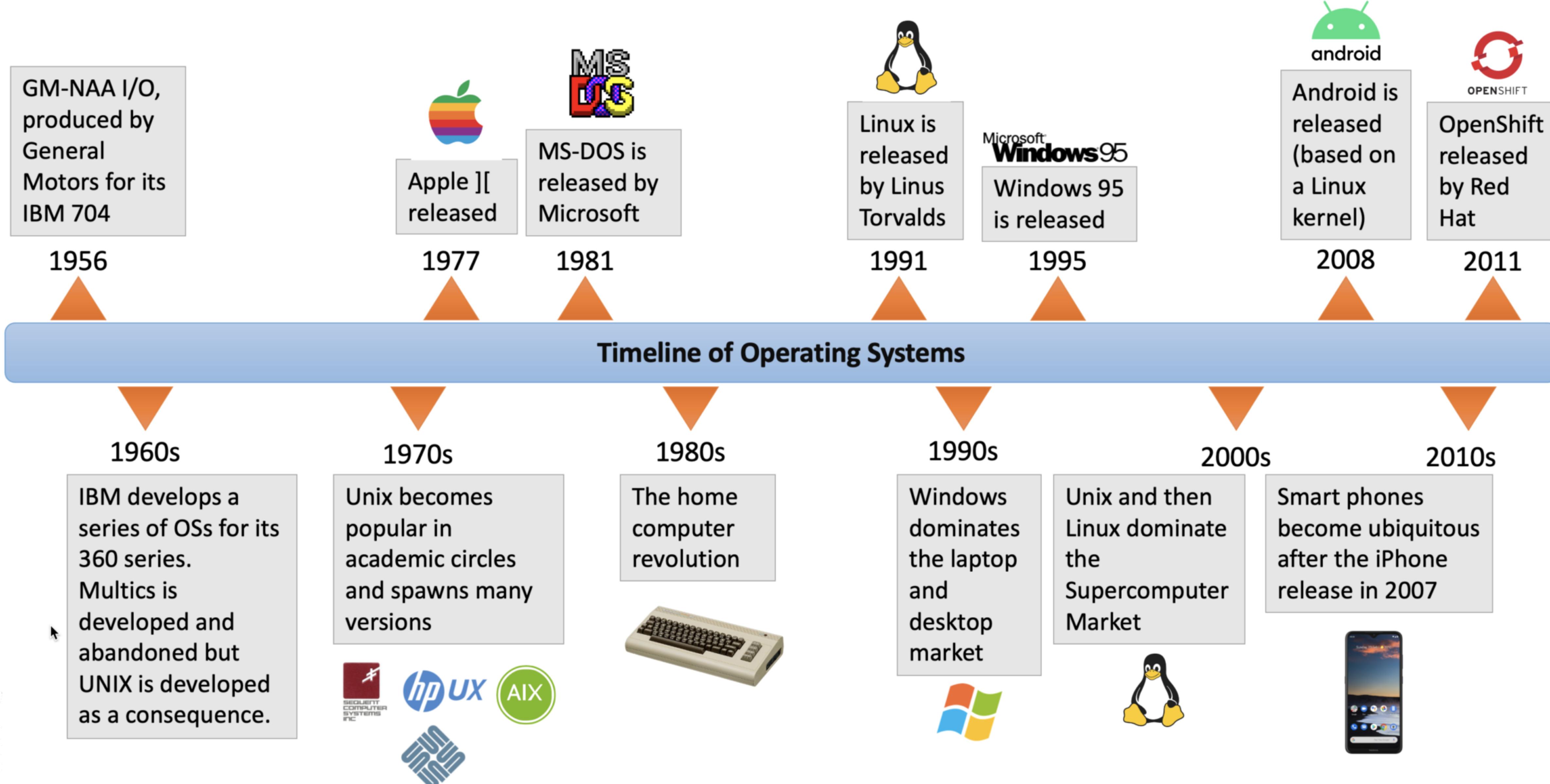
- Application is a program: Processor will Fetch -> Decode -> Execute, Continue
- CPU will have the PC which points to the instructions in the memory



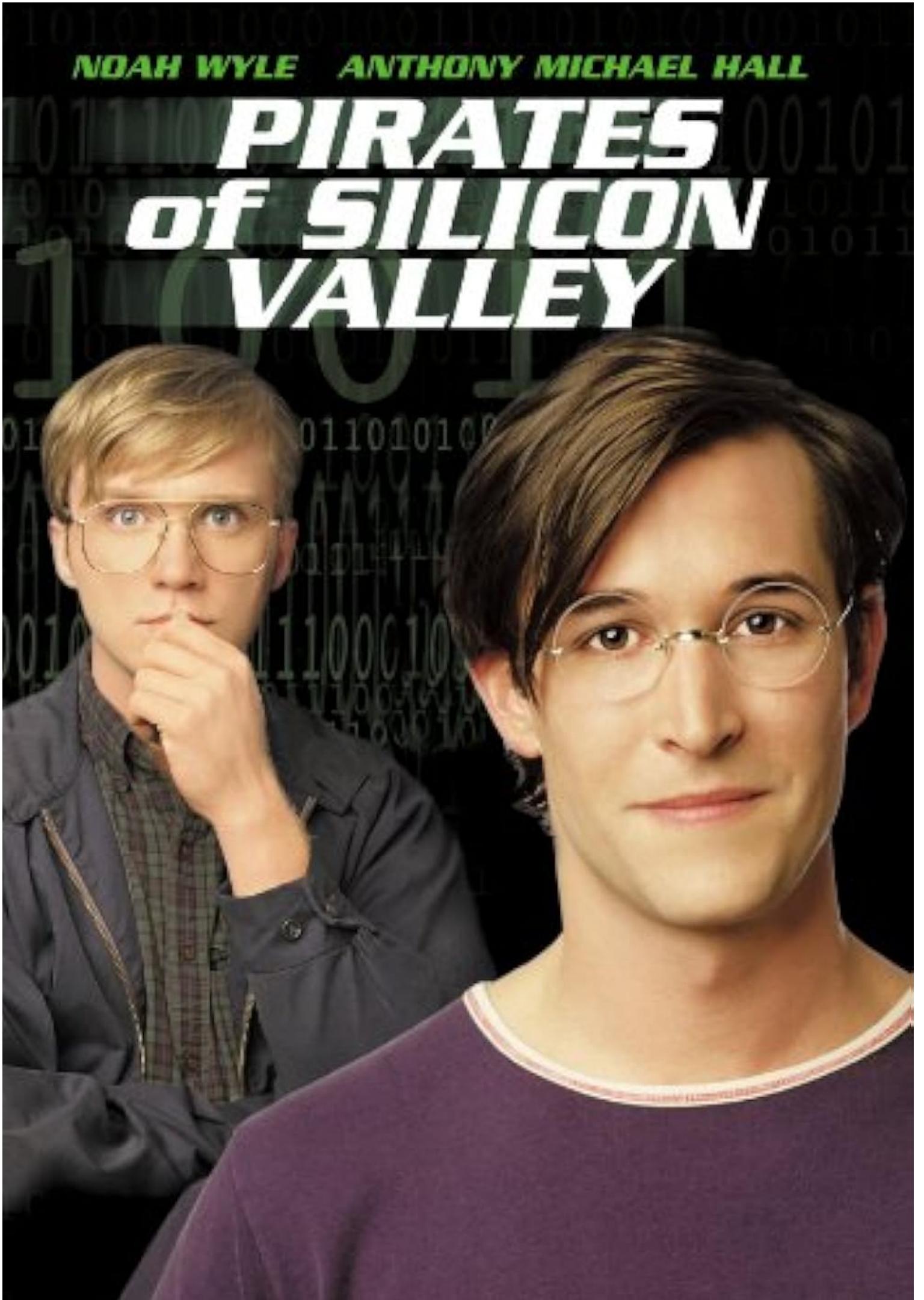
What if there were no OS?



OS: A Brief History

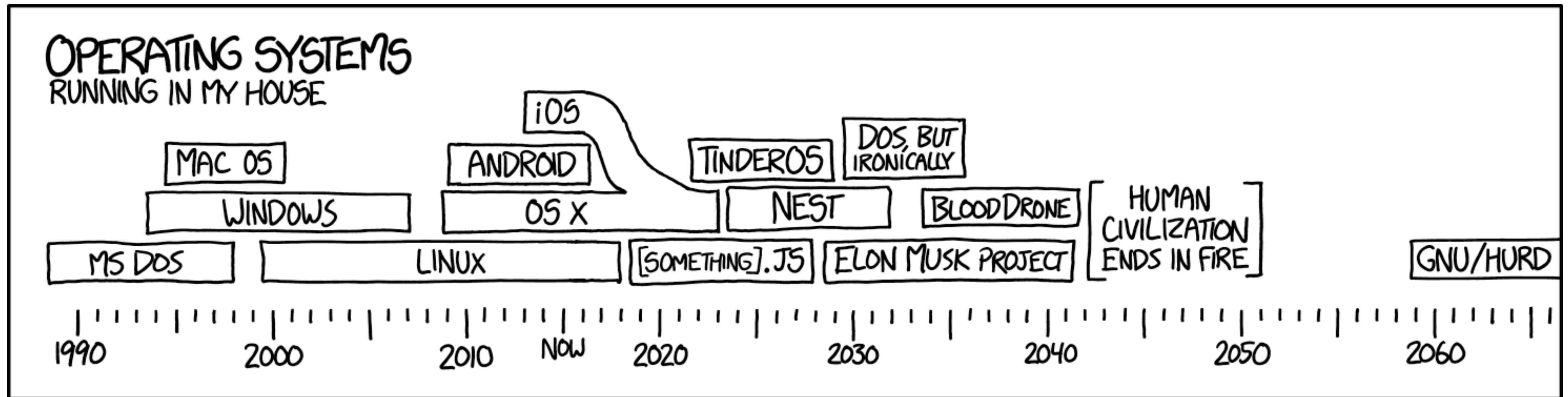


If you are into movies, do give a watch!



Source: imdb

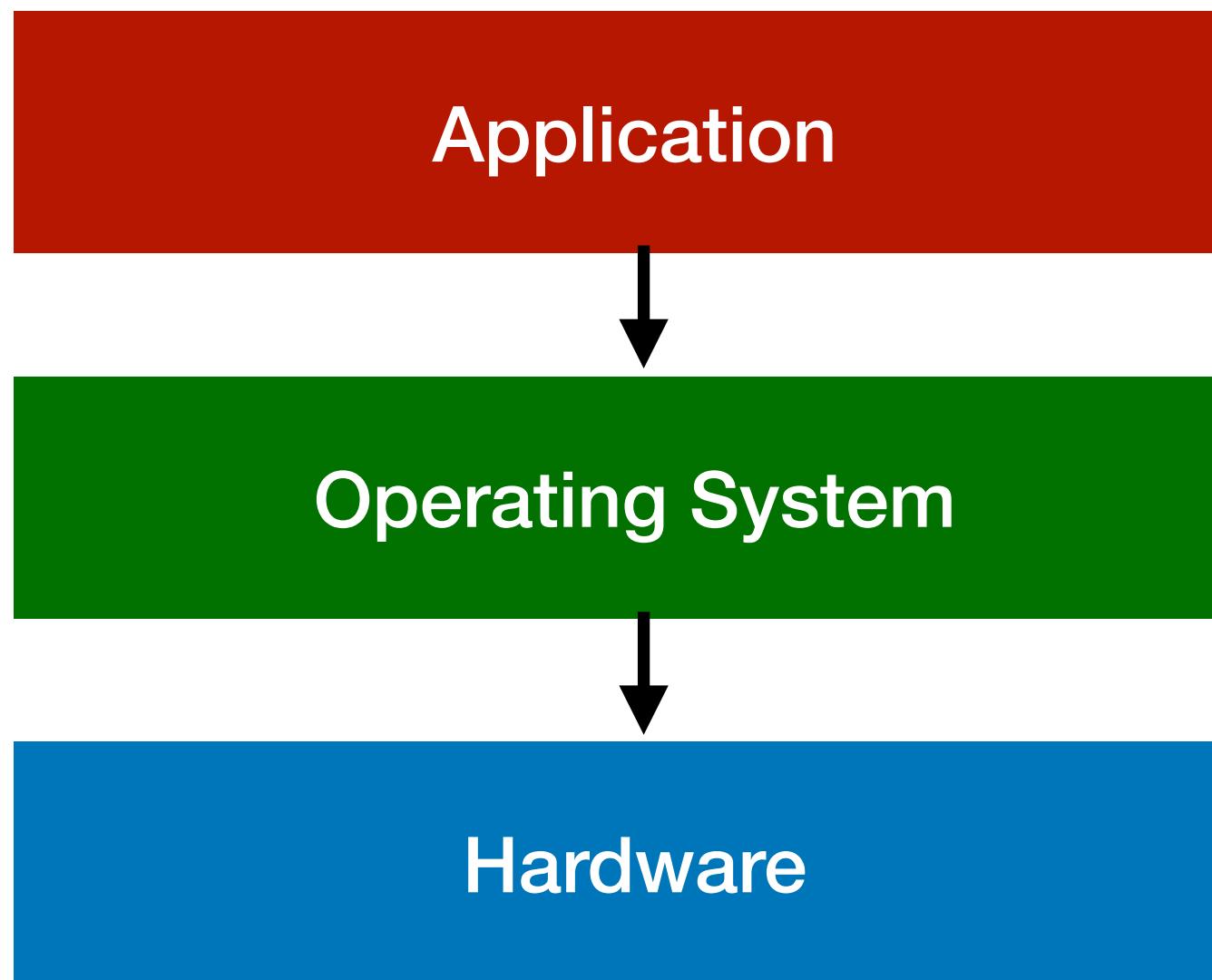
Forever Lasting!



Source: xkcd.com



Operating Systems: An Overview



- OS basically is a middleware between the hardware and the application/user program
- In charge of making sure that the system operates **correctly** and **efficiently**
- Does three key things:
 - Easy to execute programs
 - Manages memory
 - Handles the different I/O devices





Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

20% complete



For more information about this issue and possible fixes, visit <https://www.windows.com/stopcode>

If you call a support person, give them this info:

Stop code: CRITICAL_PROCESS_DIED

Whats the big deal?

- 8.5 million devices affected! - 1% of the entire windows users
- Airlines - Delta, KLM, Ryanair, Lufthansa,..
 - Many passengers were stranded in airports
- Financial organisations - London stock exchange, Visa, ...
- Healthcare - independant pharmacies and GP surgeries
- Media, retail, train operating companies.....

Economic and social impact!!!



Abstraction holds the key

Lets draw some parallels



What you see as Netflix!

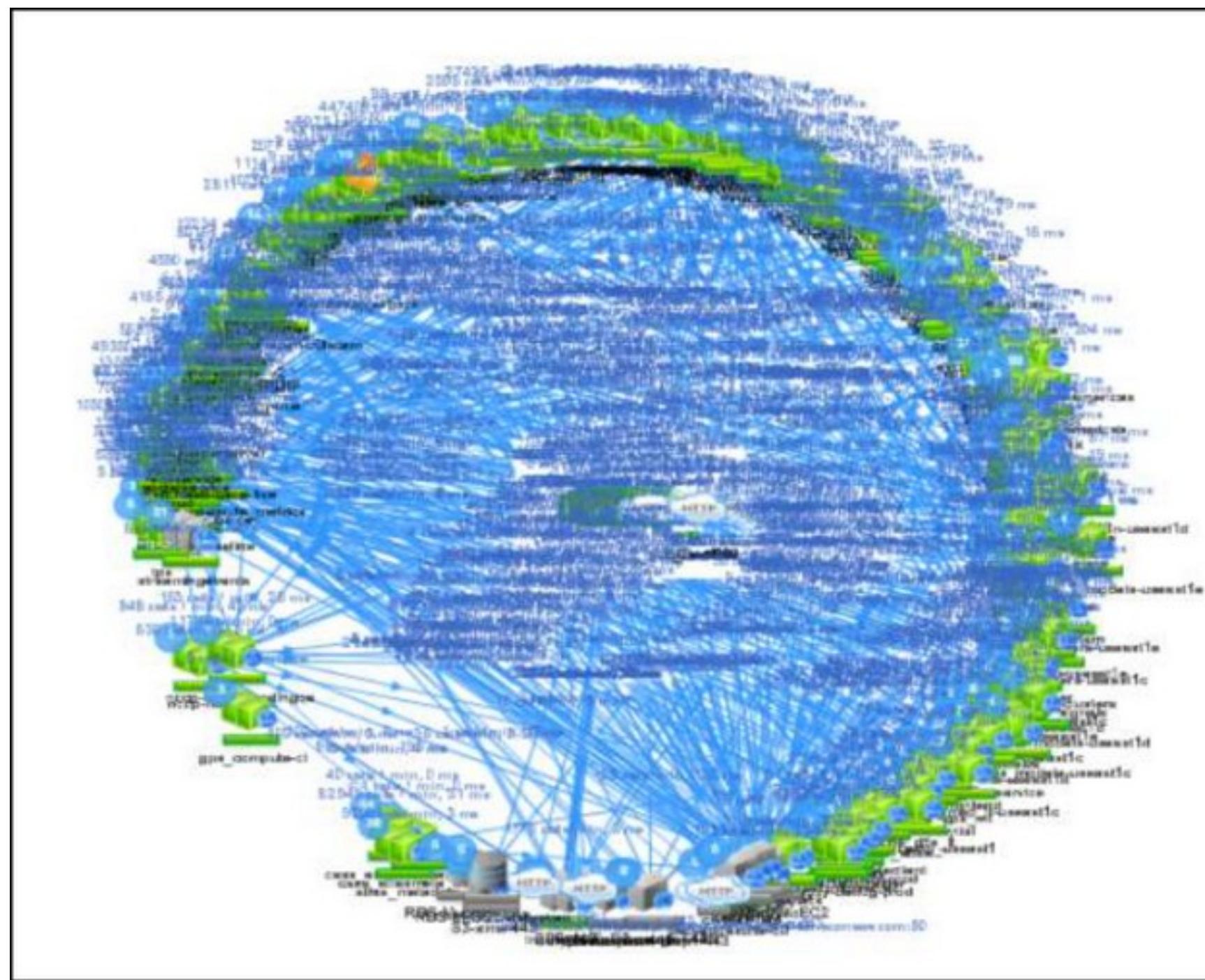
Source: [netflix.com](https://www.netflix.com)

- One of the key principles in system design - Try to keep things simple
- Abstraction and decomposition - Two main principles in Software Engineering
- Think of libraries in programming languages
 - Provides abstraction in terms of functions
 - All you need to know is functions to be called - Not how they work



Abstraction holds the key

Lets draw some parallels

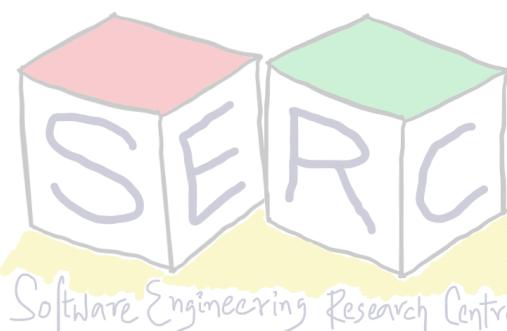


Behind the scenes!

Source: netflix.com

- One of the key principles in system design - Try to keep things simple
 - Abstraction and decomposition - Two main principles in Software Engineering
 - Think of libraries in programming languages
 - Provides abstraction in terms of functions
 - All you need to know is functions to be called - Not how they work

OS can be also thought of as a black box!



Keeping things Simple is the Key!

1134

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 47, NO. 6, JUNE 2021

Evolution of the Unix System Architecture: An Exploratory Case Study

Diomidis Spinellis^{ID}, Senior Member, IEEE and Paris Avgeriou^{ID}, Senior Member, IEEE

Abstract—Unix has evolved for almost five decades, shaping modern operating systems, key software technologies, and development practices. Studying the evolution of this remarkable system from an architectural perspective can provide insights on how to manage the growth of large, complex, and long-lived software systems. Along main Unix releases leading to the FreeBSD lineage we examine core architectural design decisions, the number of features, and code complexity, based on the analysis of source code, reference documentation, and related publications. We report that the growth in size has been uniform, with some notable outliers, while cyclomatic complexity has been religiously safeguarded. A large number of Unix-defining design decisions were implemented right from the very early beginning, with most of them still playing a major role. Unix continues to evolve from an architectural perspective, but the rate of architectural innovation has slowed down over the system's lifetime. Architectural technical debt has accrued in the forms of functionality duplication and unused facilities, but in terms of cyclomatic complexity it is systematically being paid back through what appears to be a self-correcting process. Some unsung architectural forces that shaped Unix are the emphasis on conventions over rigid enforcement, the drive for portability, a sophisticated ecosystem of other operating systems and development organizations, and the emergence of a federated architecture, often through the adoption of third-party subsystems. These findings have led us to form an initial theory on the architecture evolution of large, complex operating system software.

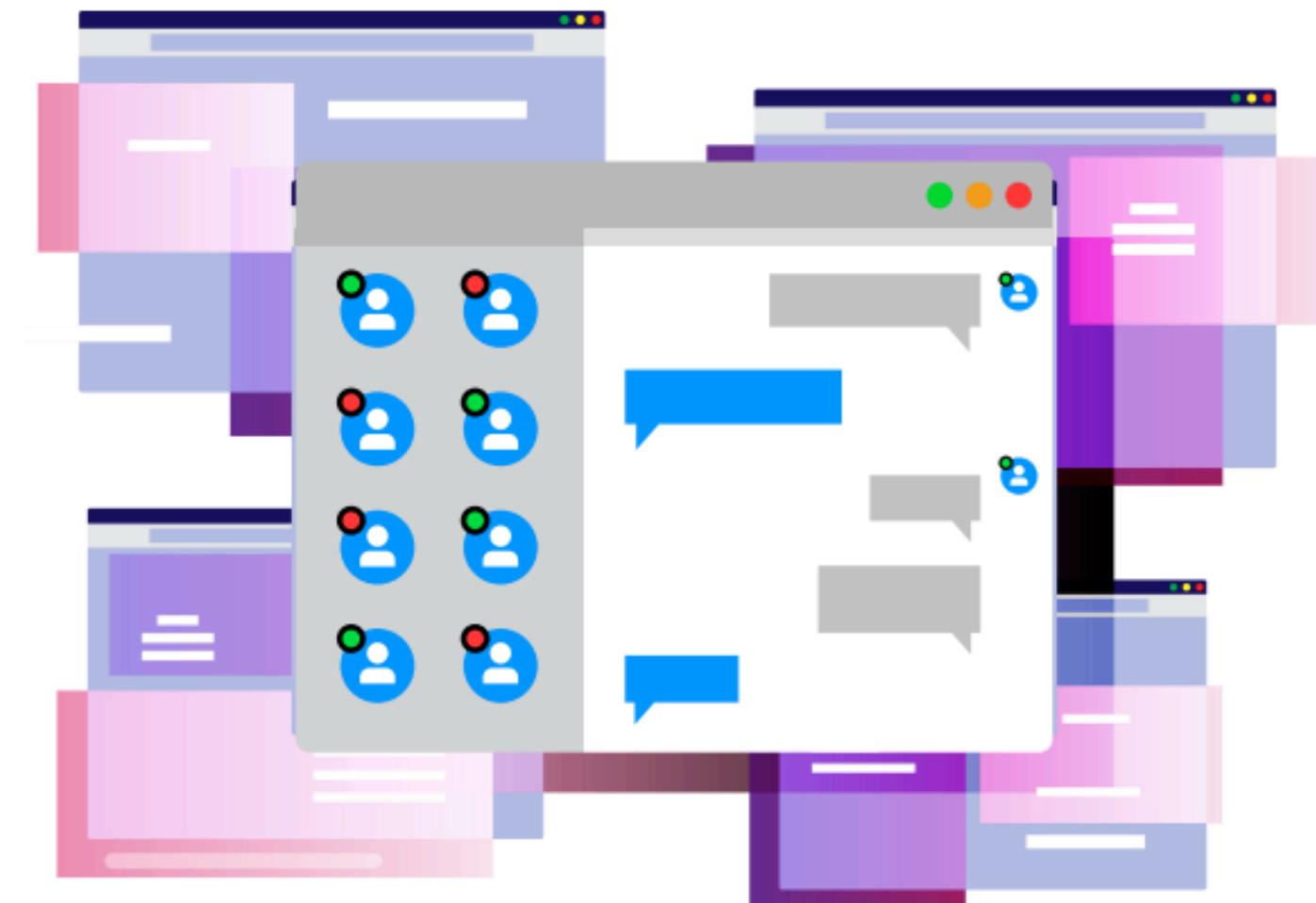
Index Terms—Unix, software architecture, software evolution, architecture design decisions, operating systems



What does an OS abstract?

Think about the messaging system that needs to be developed

- The messaging application runs on the CPU
- It requires RAM to run but it feels that it has infinite RAM
- There are also other application that needs to run simultaneously
- The application also needs to store data -> images, videos, documents...
- Application also have to interact with devices (Camera, USB,...)



The Three Pillars of Operating Systems

Virtualization
(Process and Memory)

1.OS gives every process a feeling that it has own CPU

2.Every process feels that its enough memory

Concurrency

- 1.Multiple process can run at the same time without resulting in problems
2. OS Provides mechanisms to make them work together

Persistance

- 1.Disk is an I/O device. It needs to be managed and handled.
- 2.OS handles interactions with the disk and performs storage management



Process Virtualization

- How many CPU does your computer have?
- Is the number of process equal to the number of CPU your computer have?
- Lets look into a simple example...



Memory Virtualization

- Every process requires memory to run
- How many processes are active in your system?
- What's the total memory they require?
- How's CPU managing this?



Concurrency

- OS has to juggle between multiple processes
- Heard of multi-threaded programs? But wait heard of threads?

It was a dark and stromy night

Spell check in Microsoft Word

Process Name	Mem...	Threads	Ports	Pl...
Toolbox for Keynote	621.5 MB	6	754	48:
WhatsApp Helper (Renderer)	583.9 MB	21	235	34:
Notion Helper (Renderer)	579.2 MB	17	197	71:
Microsoft PowerPoint	564.4 MB	77	35,422	44:
java	522.0 MB	83	320	29:
Microsoft Teams Helper (GPU)	453.0 MB	14	231	91:
Google Chrome	432.7 MB	45	2,246	18:
Notion	398.2 MB	33	530	6:
Microsoft Word	395.2 MB	44	3,784	48:
mysqld	384.1 MB	40	73	!

How many threads are active in Microsoft Word?



Spell check image source: <https://creativepro.com/check-spelling-dynamically/>

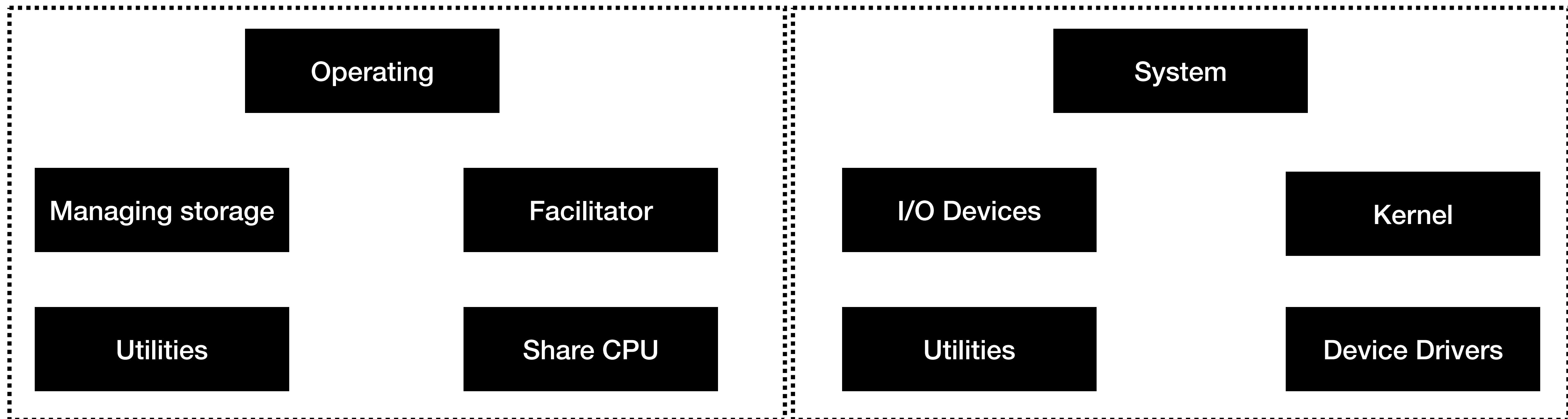
Persistence

- RAM is Volatile
- Hardware and software are needed to store data persistently
 - Hardware: I/O devices such as hard drive, SSDs, etc.
 - Software:
 - File system manages the disk
 - File system is responsible for any files that the user creates
 - Read, writes are handled by file system which interacts with low level device drivers



“Operating System” - Dual Roles

Resource Management and Hardware Abstraction

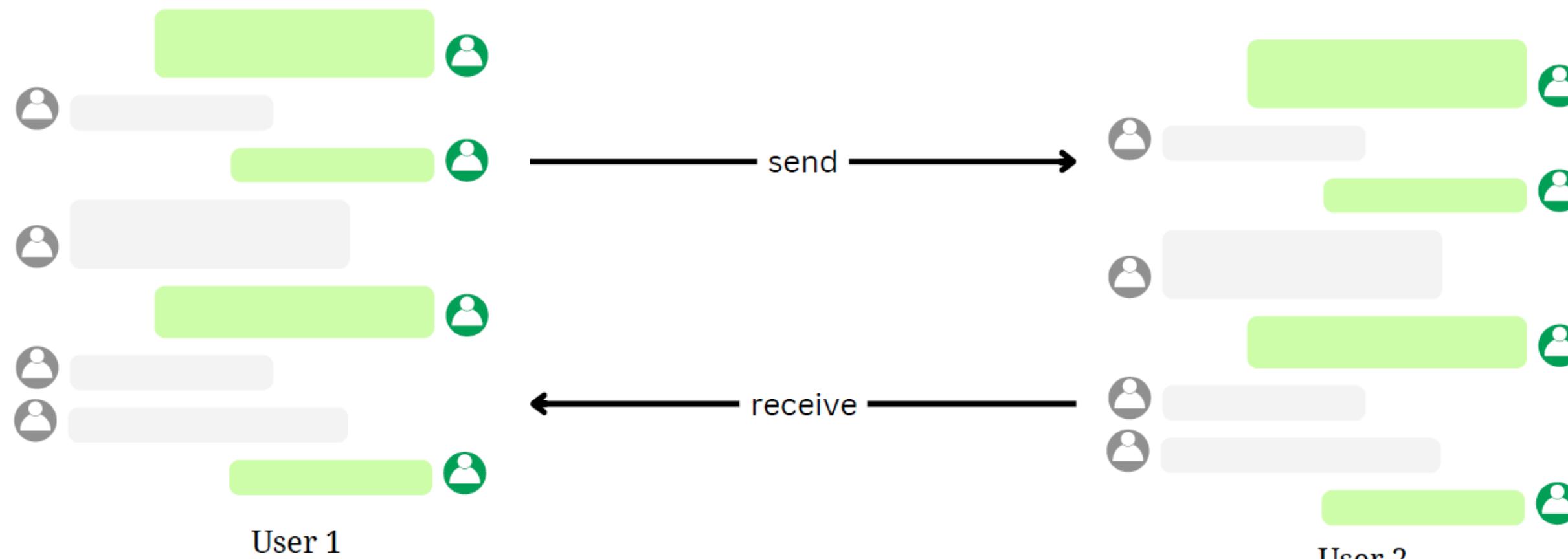


Key Design Goals Of OS

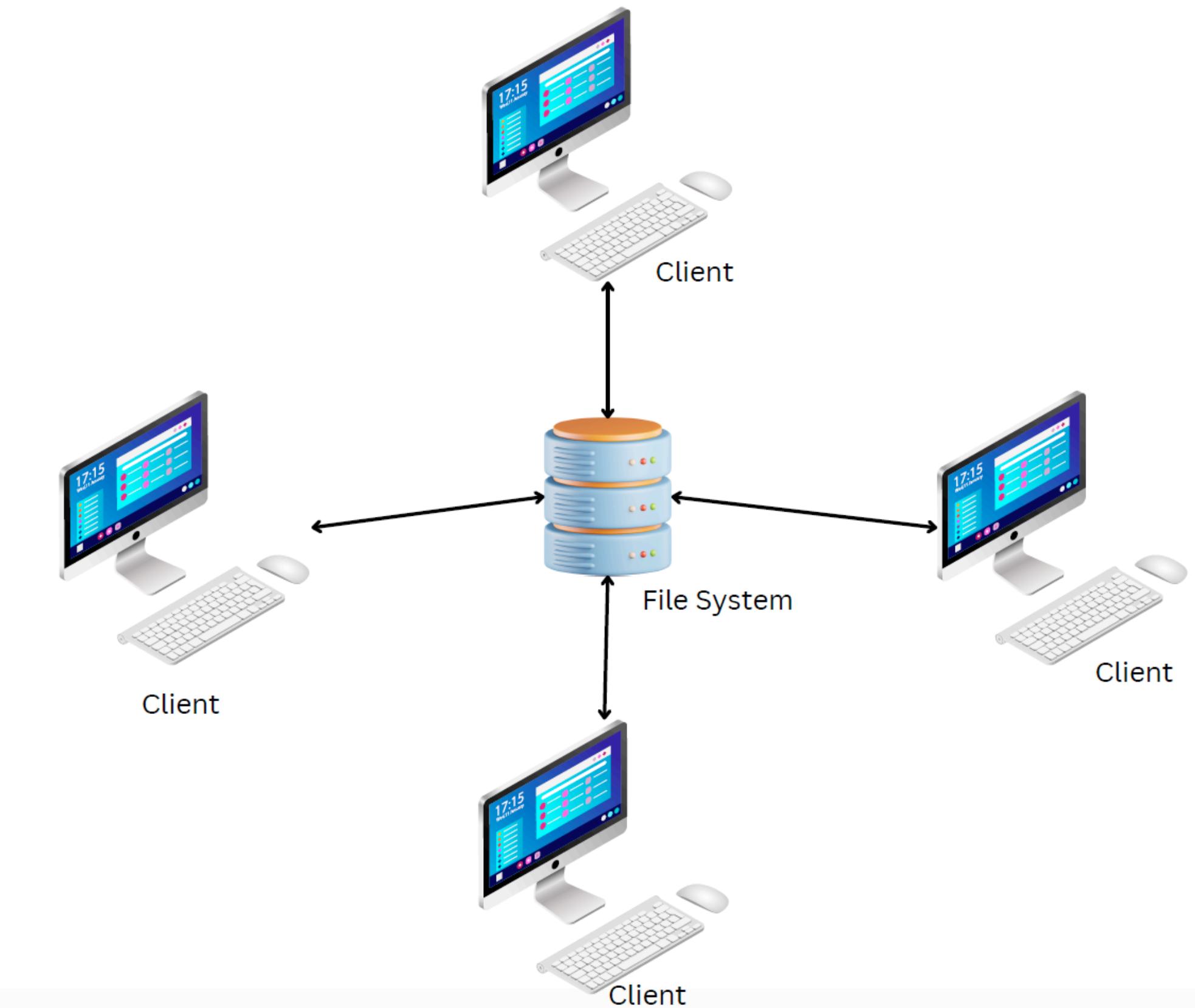
- **Abstraction:** Convenience and easy to use
- **High Performance:** Minimize overhead, Virtualisation should be done but minimise overhead
- **Reliability:** OS must continuously run without crashing
- **Other attributes:** Energy efficiency, Security, Mobility
- **Remember:** OS itself is a massively complex software (Softwares rely on OS to function correctly!!)



But how does multiple systems interact?



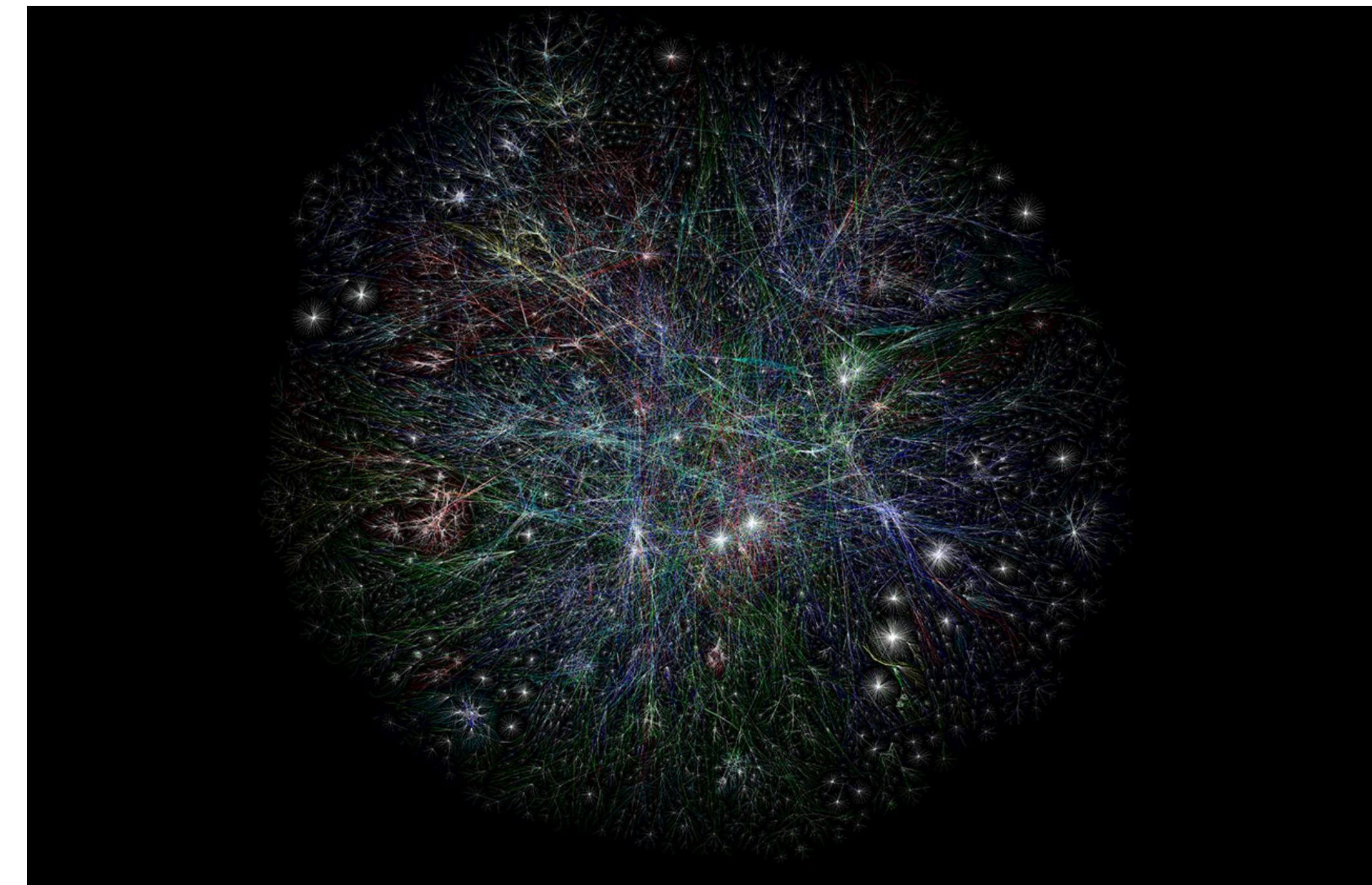
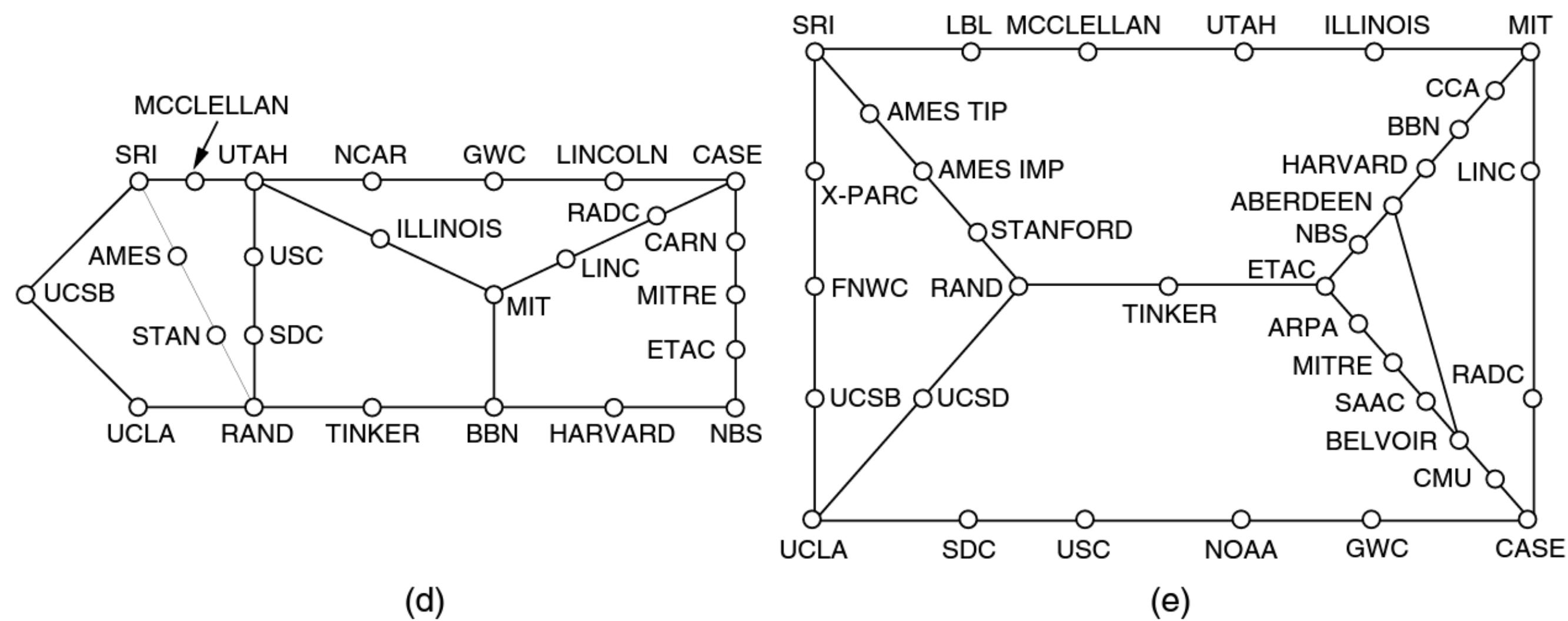
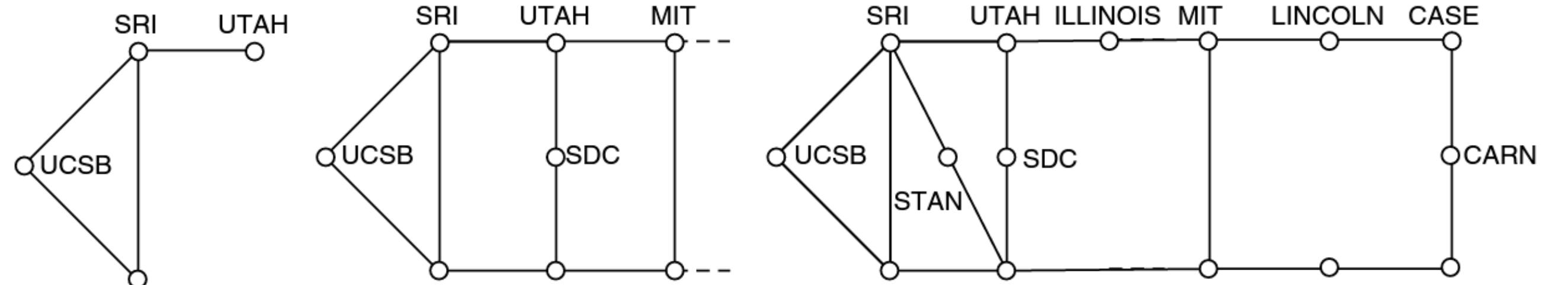
WhatsApp Chat example



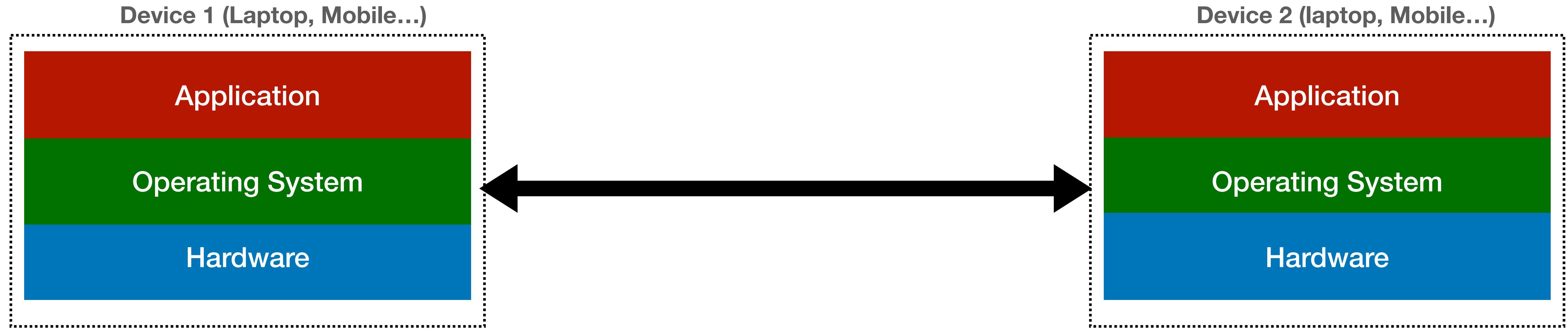
Network File System



Any Guess on this?



Computer Networks: The horizontal



- How can application from device 1 and 2 communicate?
- How to ensure the data/information reaches reliably?
- What if multiple applications are running and are communicating?



Networks - Different Components

- Different types of network - PAN, LAN, MAN, WAN,..
- Some terminologies - subnets, hosts, routers, switches, transmission lines, interconnected networks (internet) not Internet
- **internet** - formed when distinct networks are interconnected (network of networks)
- Subnets - collection of communication lines and routers owned by network operators
- Protocol - Agreement between communicating parties on how the communication needs to proceed

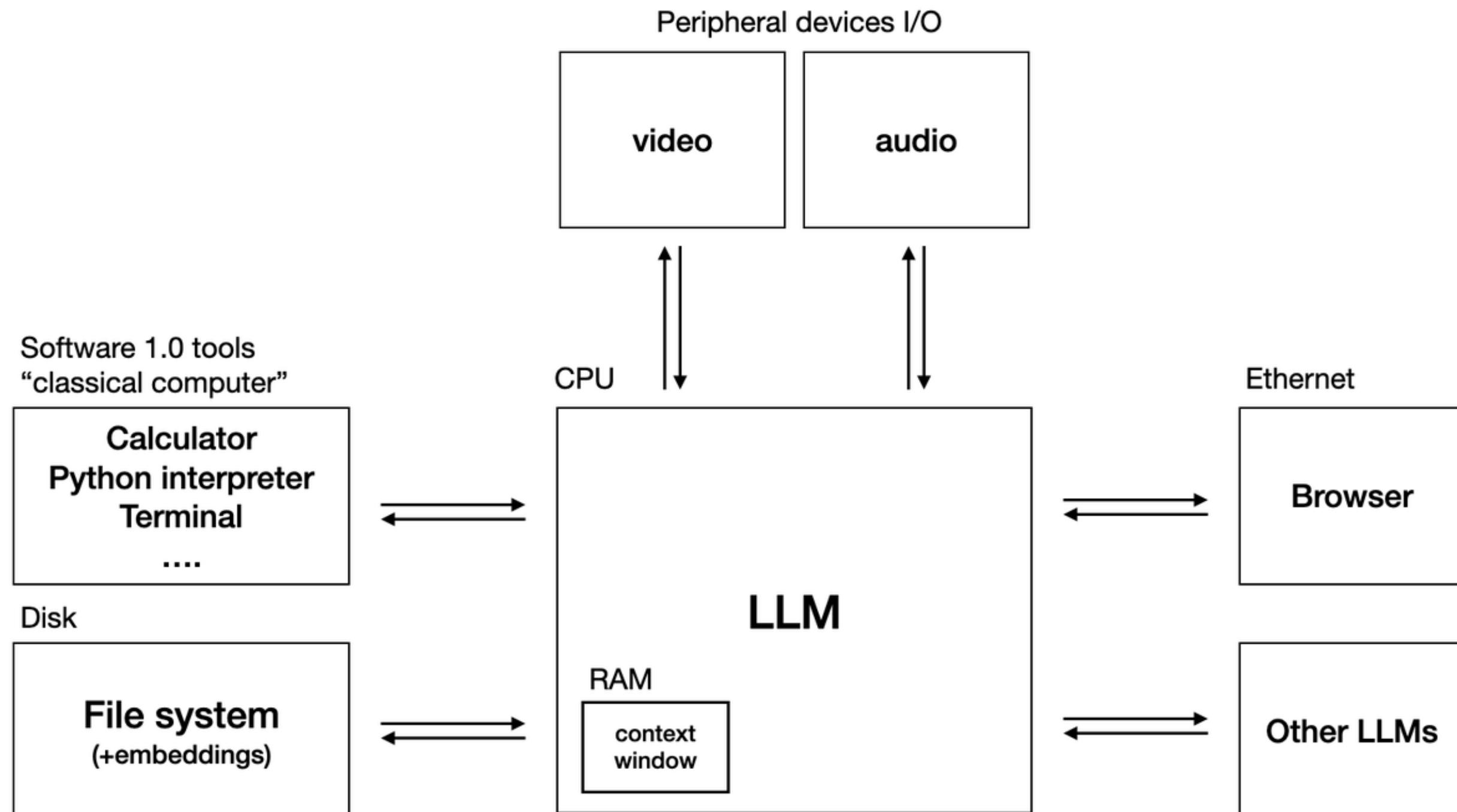
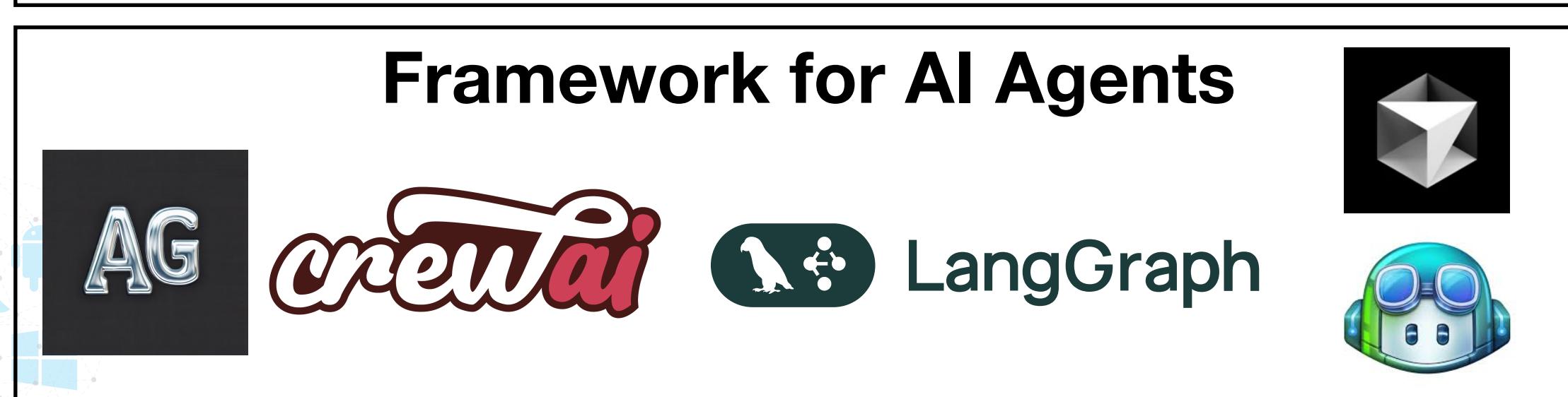
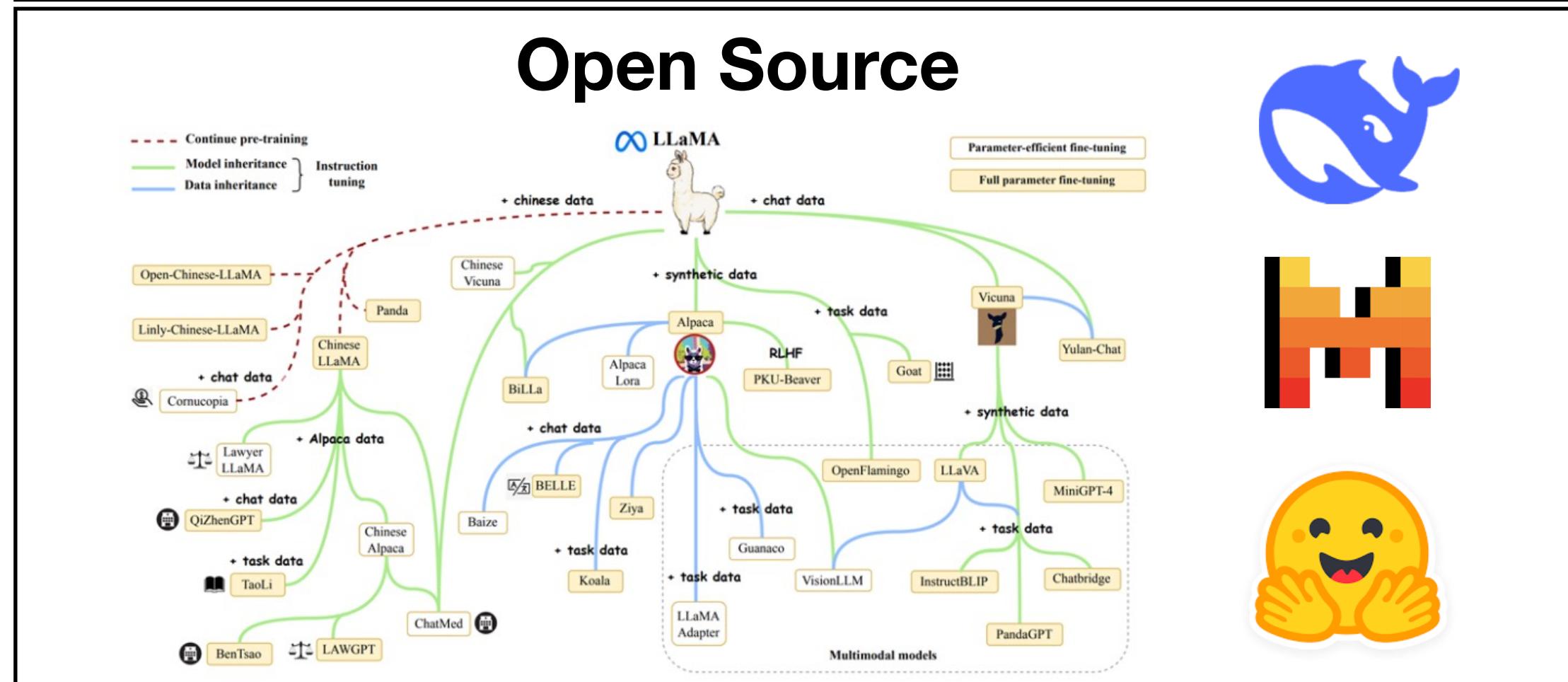


The Different Pieces of Networks

- As more and more networks started joining ARPANET, need for common protocol arise
 - Invention of TCP/IP model
- Implementing TCP/IP on different platforms was encouraged
 - **Sockets** - 4.2BSD Release of Berkeley Unix
 - Using networks with sockets proved effective and efficient
- As scale increased, difficult to remember the address of the system
 - **Domain Name System (DNS)**
 - How to find route from one point to another?



Lot is happening....



Source: Andrej Karpathy talk on LLMs

MemGPT: Towards LLMs as Operating Systems

Charles Packer¹ Sarah Wooders¹ Kevin Lin¹
Vivian Fang¹ Shishir G. Patil¹ Ion Stoica¹ Joseph E. Gonzalez¹

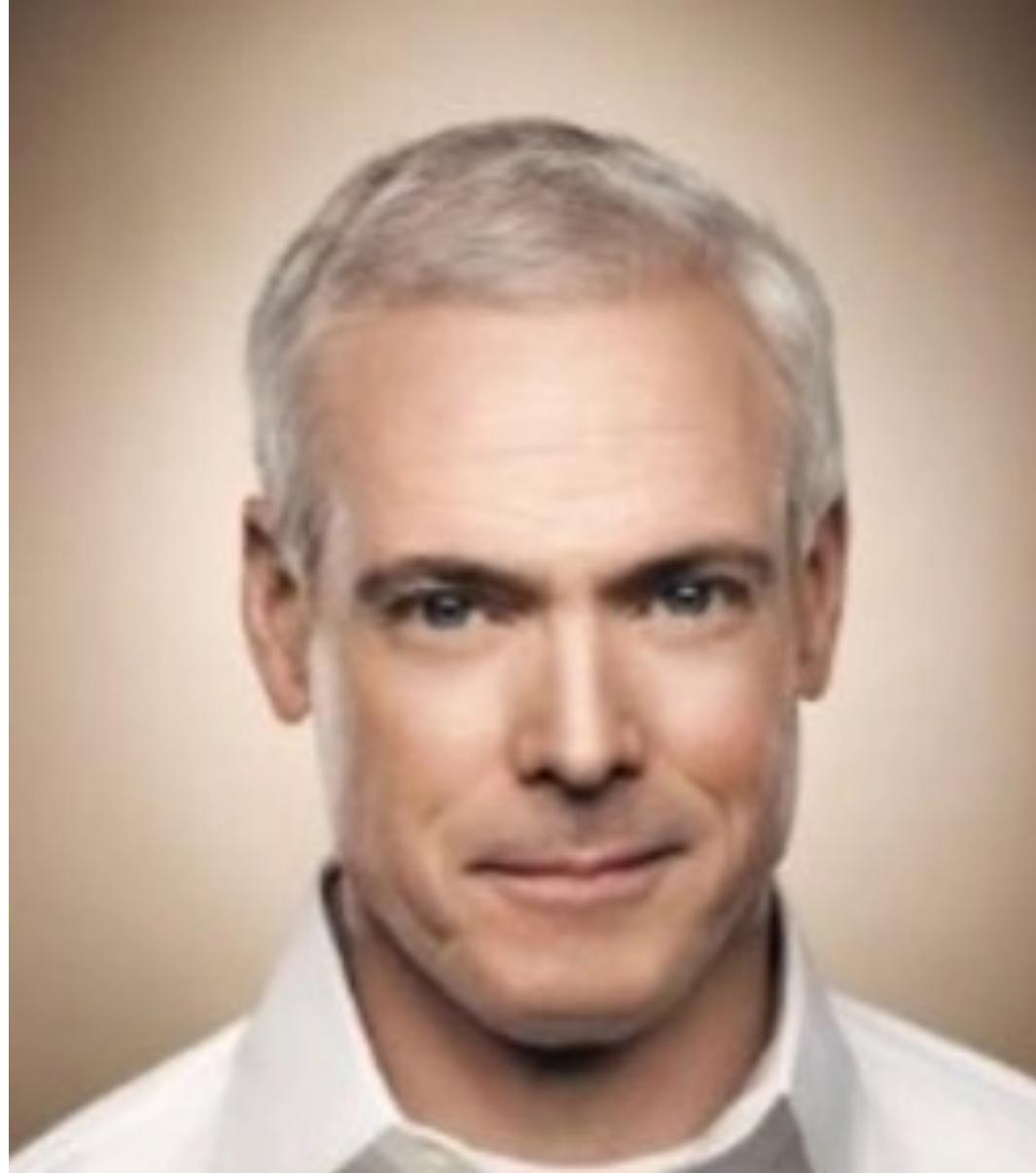
Abstract

Large language models (LLMs) have revolutionized AI, but are constrained by limited context windows, hindering their utility in tasks like extended conversations and document analysis. To enable using context beyond limited context windows, we propose *virtual context management*, a technique drawing inspiration from hierarchical memory systems in traditional operating systems which provide the illusion of an extended virtual memory via paging between physical mem-

LLMs can only support a few dozen back-and-forth messages or reason about a short document before exceeding their maximum input length (Touvron et al., 2023).

Directly extending the context length of transformers incurs a quadratic increase in computational time and memory cost due to the transformer architecture's self-attention mechanism, making the design of new long-context architectures a pressing research challenge (Dai et al., 2019; Kitaev et al., 2020; Beltagy et al., 2020). While developing longer models is an active area of research (Dong et al., 2023), even if we could overcome the computational chal-

Remember...

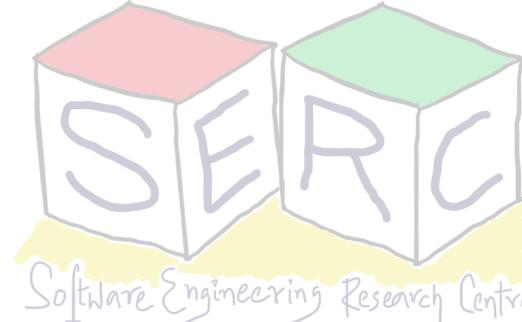
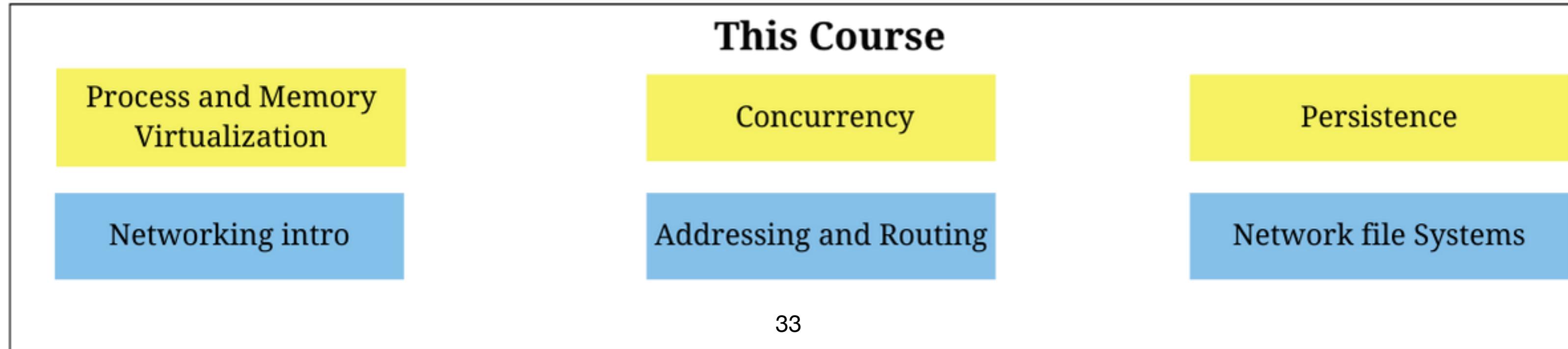
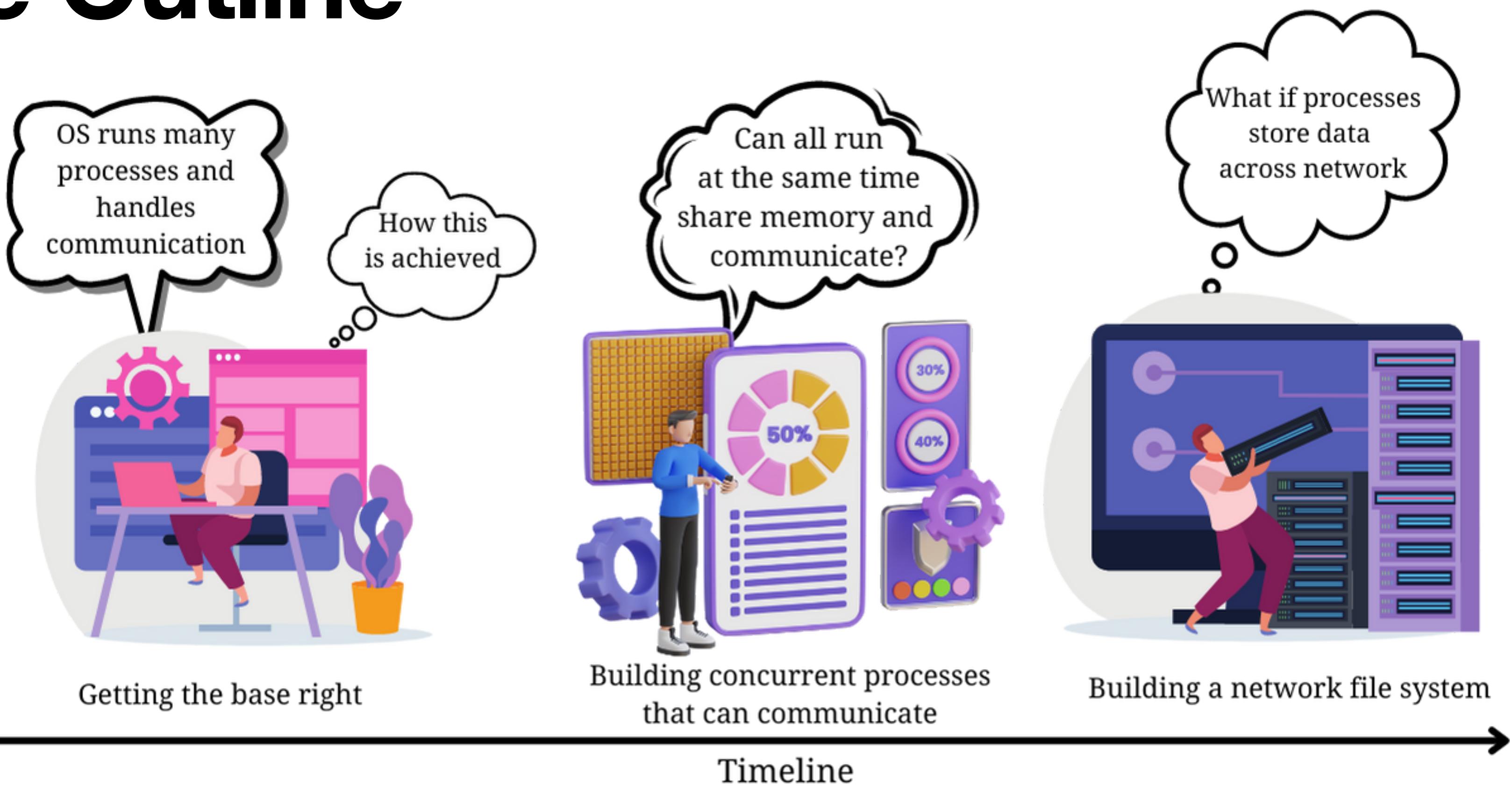


In a world of constant change, the fundamentals are more important than ever.

— *James C. Collins* —

- Understanding OS and Networks at fundamental level is very important
- Good system design and development becomes a natural consequence
- Provides solid foundation for understanding higher abstractions or to abstract!

Course Outline



Grade Distribution

Component	Weightage
Final Exam	30%
Mid-term Exam	15%
Quiz	15%
Mini projects	20%
In-class activities/Bonus	5%
Project	20%

Note: The instructor reserves the right to make any changes based on the course progress



Course Logistics

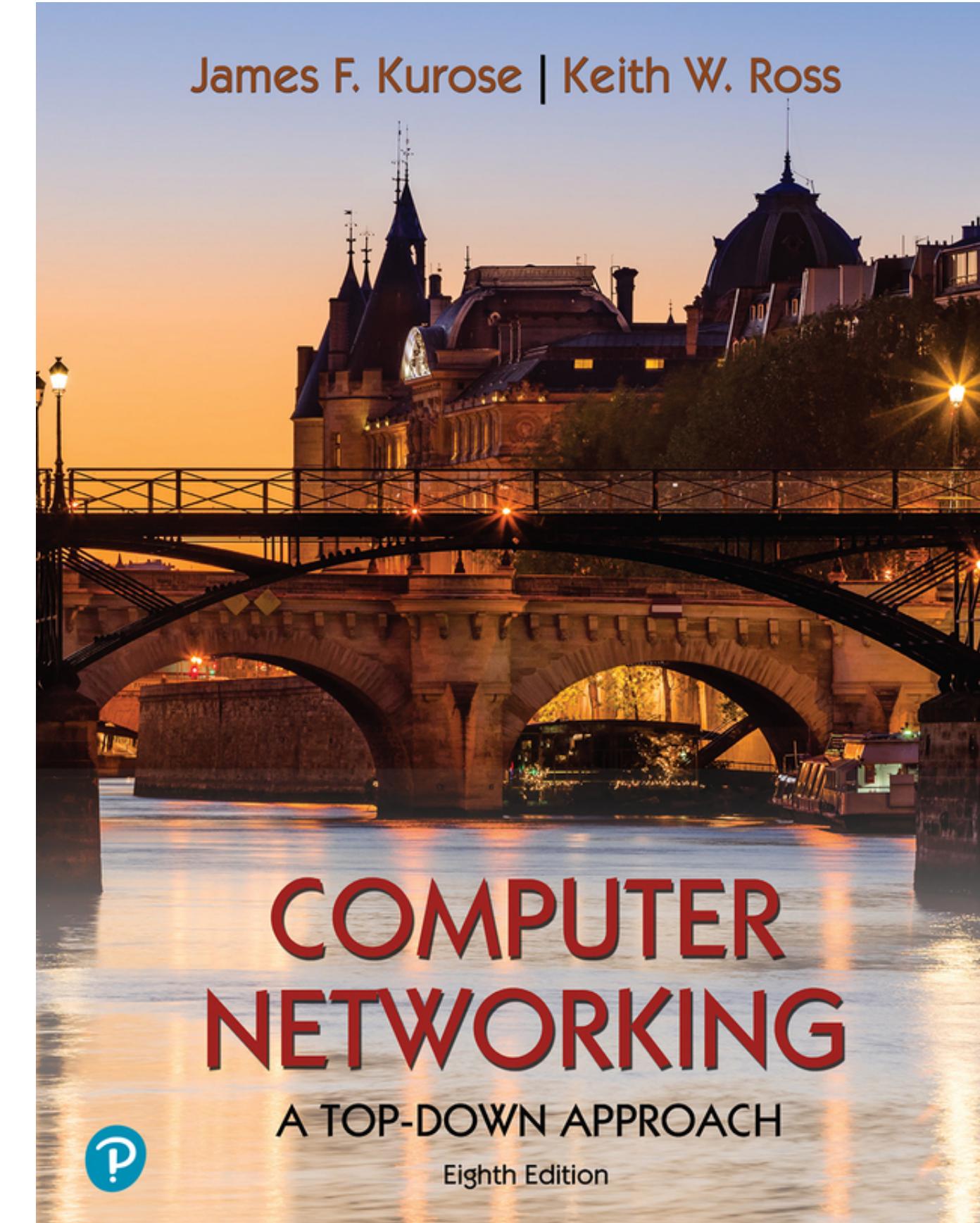
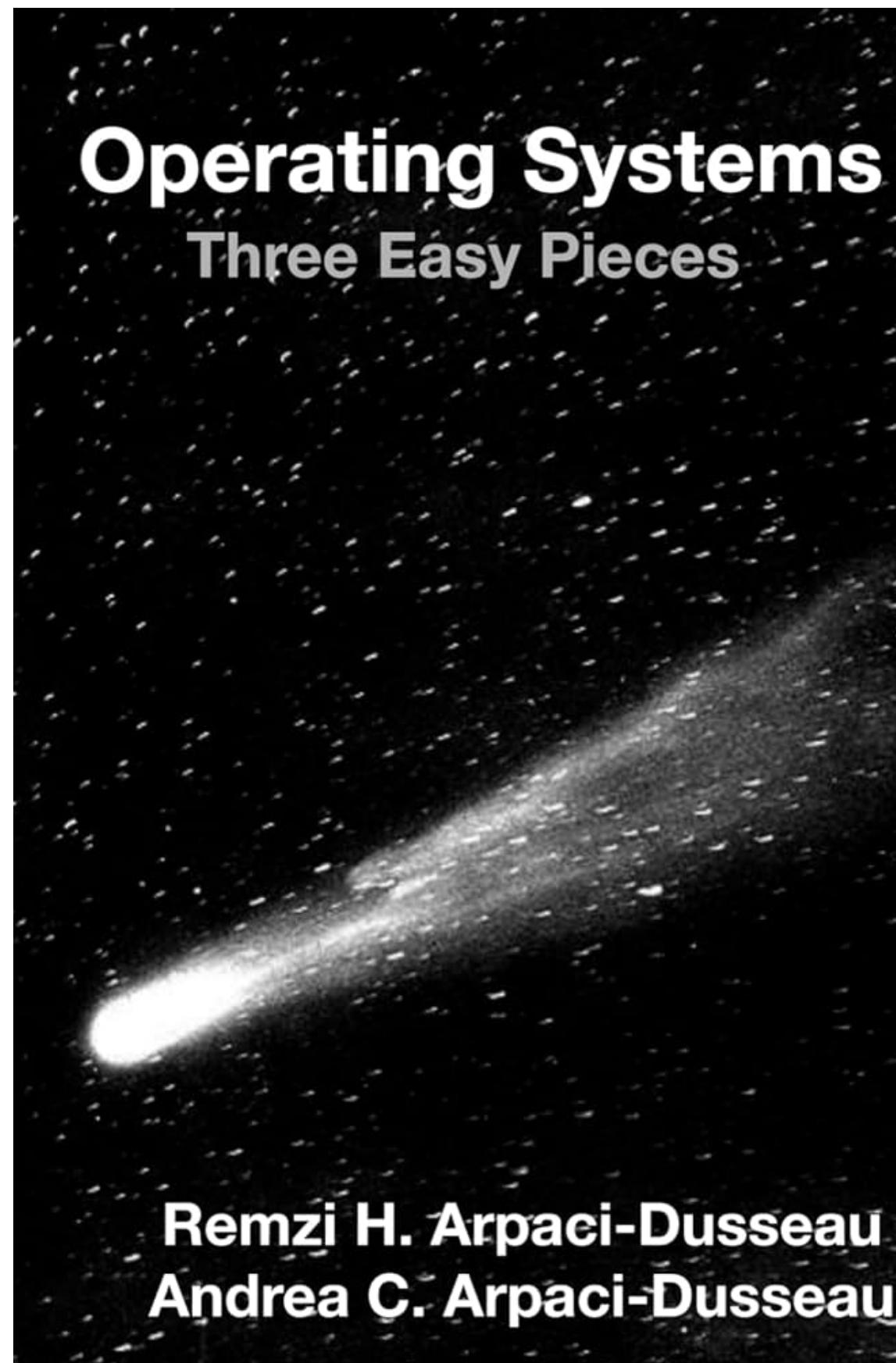
- Course announcements and management - Moodle 
- Assignments and projects - Github Classroom
- All resources, information and materials - https://karthikv1392.github.io/cs3301_osn/
- The complete course schedule is already posted (minor changes may happen!)
- Instructor office hours - Tuesday and Thursday (5:30 to 6:00 PM)
- Lets keep the classes lively and interactive!
- Feedbacks are always welcome!

OSN Continuous Q&A



Course Logistics

Materials



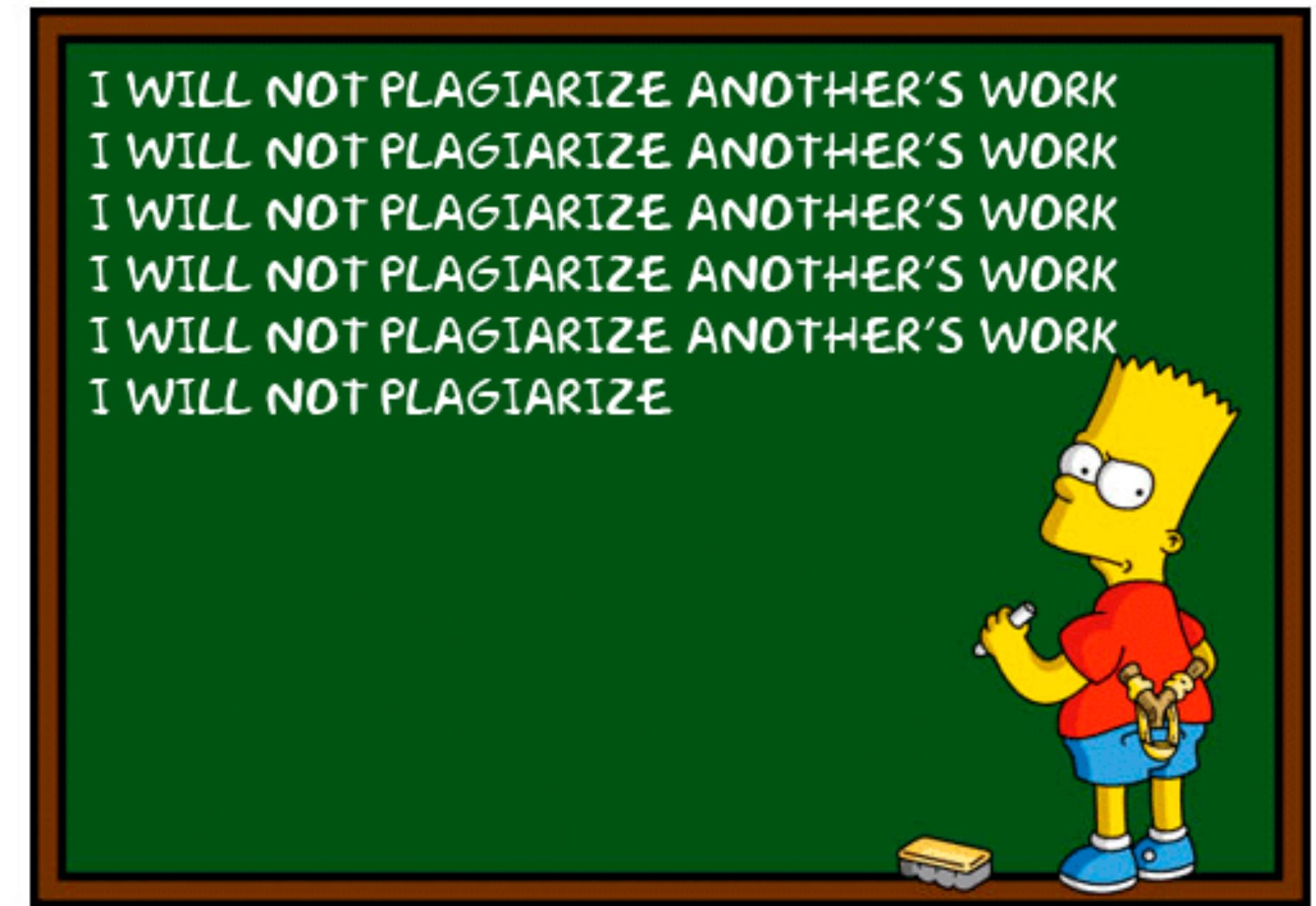
- The OS book is available as e-copy: <https://pages.cs.wisc.edu/~remzi/OSTEP/>

- Very important to read the **book + the slides + website posts + being attentive!!**

Course Logistics

Some Ground Rules

- **Zero Tolerance on Plagiarism!**
 - Every submission will go through different checks
 - If found, the points will be 0 and may result in an **F** grade
 - Check the institute policy on Academic misconduct
- You may use AI tools but use them wisely! - Check course policy page
 - Try to understand rather than blindly using generated code!



- Please respect the class timings!



Course Logistics

Communication and Engagement

- We have planned many ways to make this a very engaging course
- **Take part in tutorial sessions: Not just for projects but for more understanding - Bonus questions during the exam from tutorials**
- Instructor office hours are available - Walk into office for any queries!
- For every 20-25 students - A dedicated TA will be allocated
- TA Office hours - Please do use them but kindly respect the timings!
 - Remember TAs are also students and they have their own commitments
- Document will be shared for posting questions
 - Check the questions before you post one!
 - Add questions to the respective category



OSN Continuous
Q&A



Course Logistics

Projects

- There will be a mini project 0 (Warmup) and two other mini projects each about a month duration
 - All the mini projects will be graded, there will be viva as well (total 20%)
- **Final course project (20%) - Team of 2**
- No extensions will be granted unless there is any emergency (get approval):
 - Use extra days and soft deadlines
 - Any submission over the hard deadline will not be considered and will be evaluated with 0



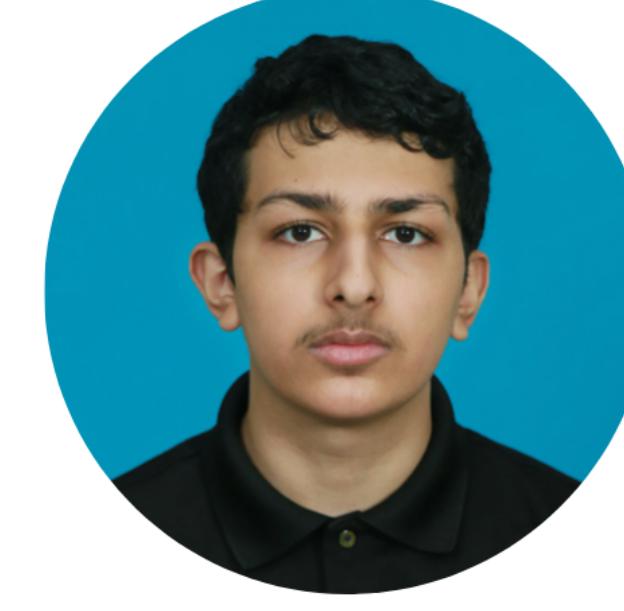
Meet the Team!



Akhila Matathammal



Ananya Halagatti



Anirudh Vempati



Aviral Gupta



Karthik Vaidhyanathan



Divijh Mangtani



Eshaan Sharma



Kriti Gupta



Prakhar Jain



Prasoon Dev

**Special Mention:
Previous year(s) TAs**



Miryala Sathvika



Shlok Sand



Shubham Goel



Varun Gupta





Thank you

Course site: karthikv1392.github.io/cs3301_osn

Email: karthik.vaidhyanathan@iiit.ac.in

Twitter: [@karthyishere](https://twitter.com/karthyishere)

