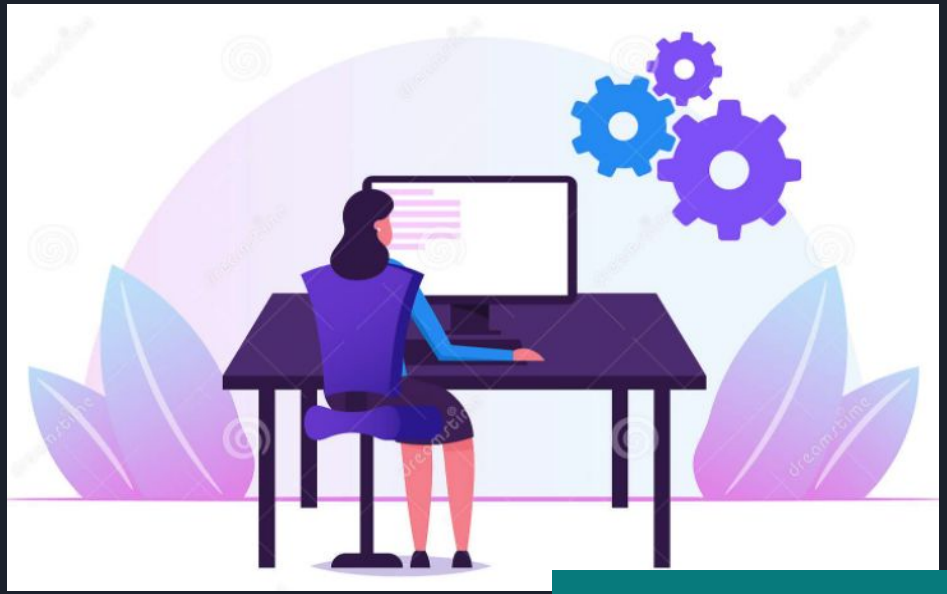
Welcome!

- We'll start in a moment :)
- We are NOT recording tonight's event. We may plan to take screenshots for social media.
    - ***If you want to remain anonymous***, change your name & keep video off.
- We'll introduce the hosts and break in-between for Q/A.
- We will make some time for Q&A at the end of the presentation as well.
- You can come prepared with questions. And, feel free to take notes.
- Online event best practices:
    - Don't multitask. Distractions reduce your ability to remember concepts.
    - Mute yourself when you aren't talking.
    - We want the session to be interactive.
    - Feel free to unmute and ask questions in the middle of the presentation.
    - Turn on your video if you feel comfortable.
    - Disclaimer: Speaker doesn't knows everything!

Check out:

- Technical Tracks and Digital Events
- Get updates – join the Digital mailing list
- Give us your feedback – take the Survey

WOMEN WHO CODE



WWCode Digital + **Backend Study Group**

May 5, 2022

WOMEN WHO CODE

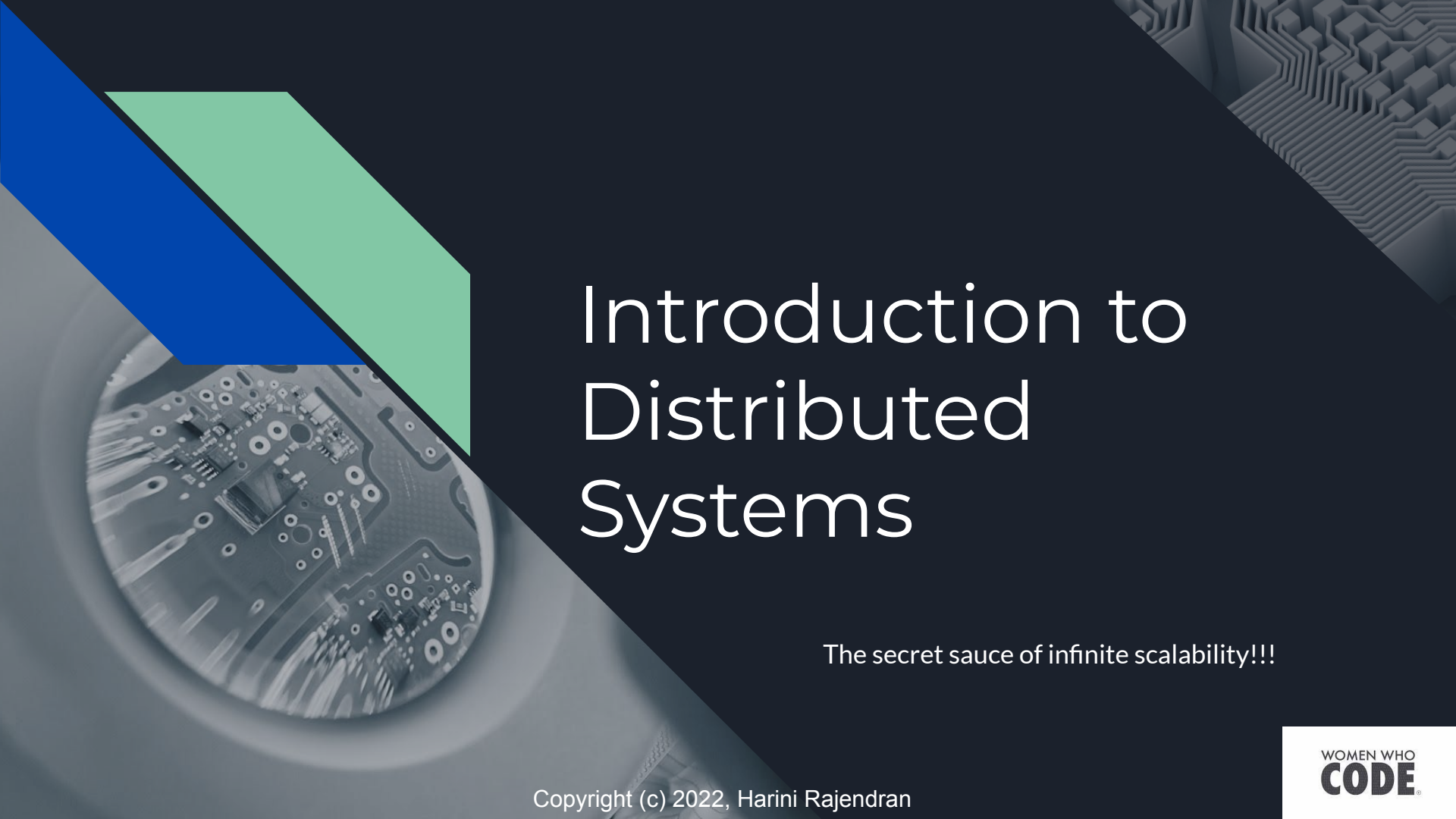
Backend Study Group

- Welcome from WWCode!
- Our mission: Inspiring women to excel in technology careers.
- Our vision: A world where women are representative as technical executives, founders, VCs, board members and software engineers.

Harini Rajendran
Lead, Women Who Code San Francisco
https://www.linkedin.com/in/hrajendran/

Prachi Shah
Director, Women Who Code San Francisco
https://www.linkedin.com/in/prachisshah/

WOMEN WHO CODE



Introduction to Distributed Systems

The secret sauce of infinite scalability!!!

WOMEN WHO CODE

Agenda

What is Distributed Systems

Why Distributed Systems

Characteristics of Distributed Systems

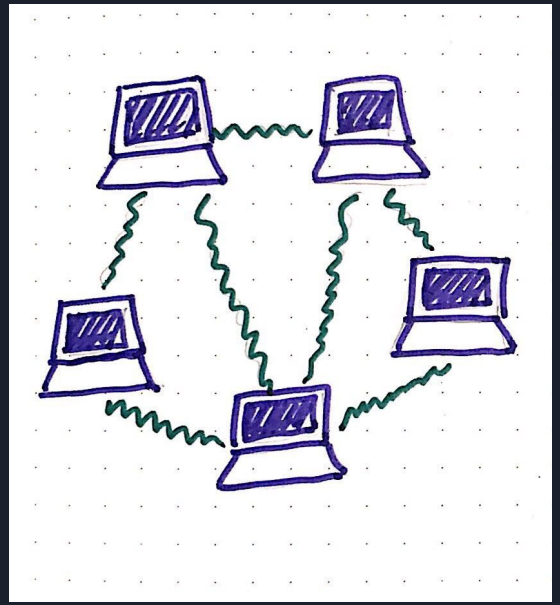
Challenges of Distributed Systems

Distributed Systems Categories

Q & A

WOMEN WHO CODE

What is Distributed Systems



A group of computers working together but appears as a single computer to the end user.

WOMEN WHO CODE

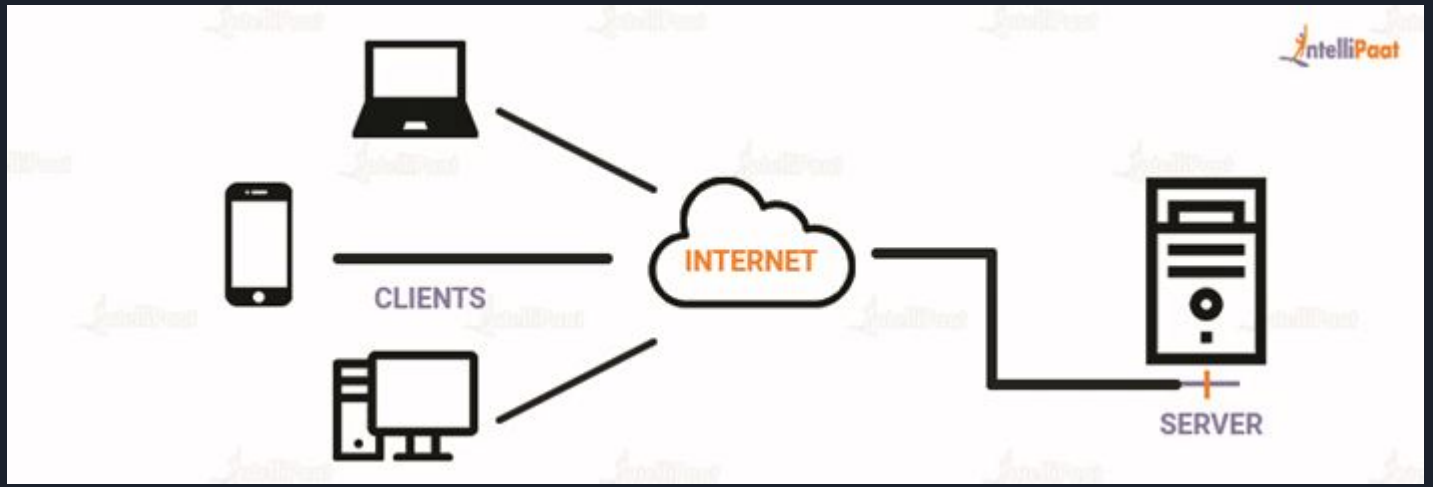
What is Distributed Systems

- Shared state, operate concurrently
- Can fail independently without affecting the whole system's uptime. No Single Point of Failure
- The devices split up the work, coordinating their efforts to complete the job more efficiently than if a single device had been responsible for the task.
- All components are connected by a network for communication with each other

Types of Distributed Systems Architecture

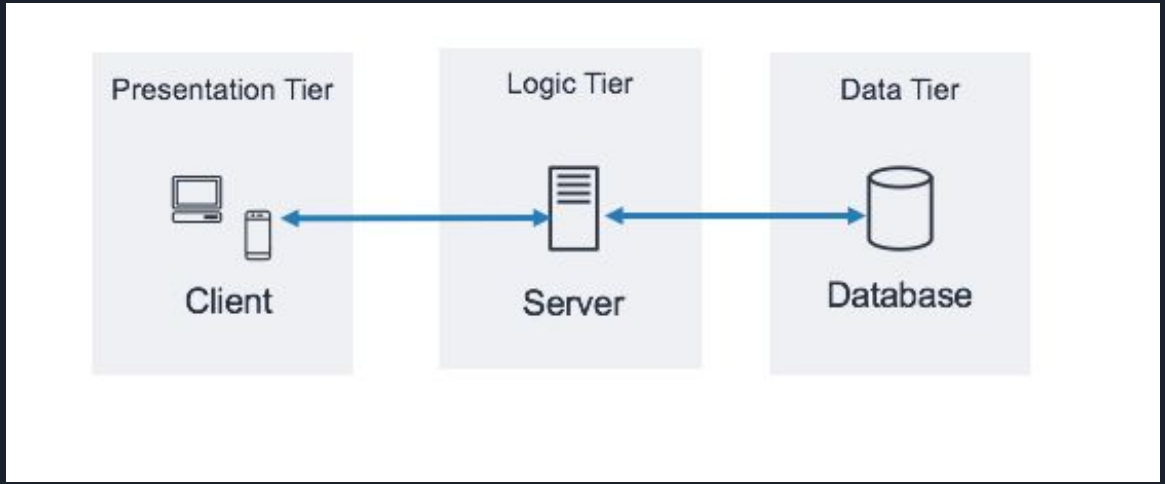
- Client Server
- Multi Tier
- Peer to Peer

Client Server Architecture



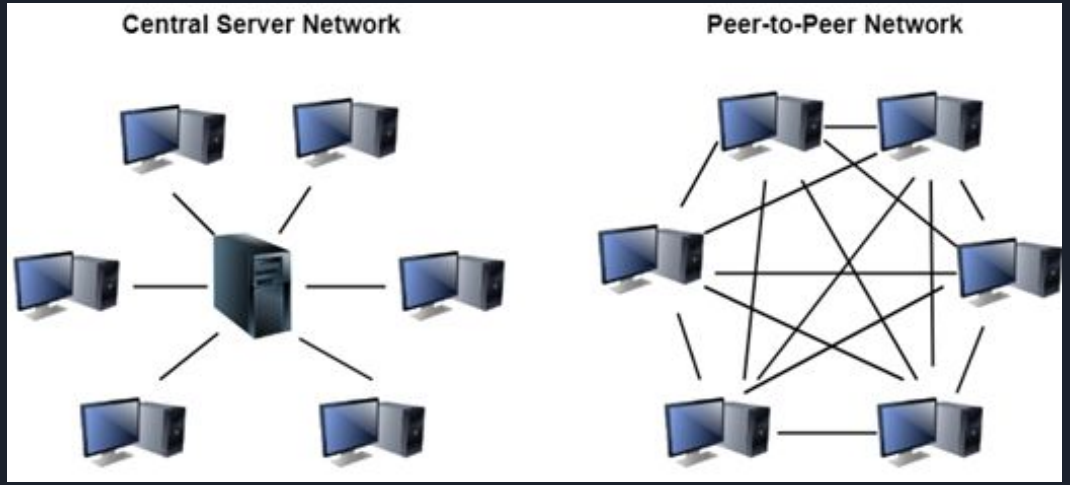
- Many clients (remote processors) request and receive service from a centralized server (host computer).
- Server does all heavy-lifting
- Ex: Printers in library

Multi Tier Architecture



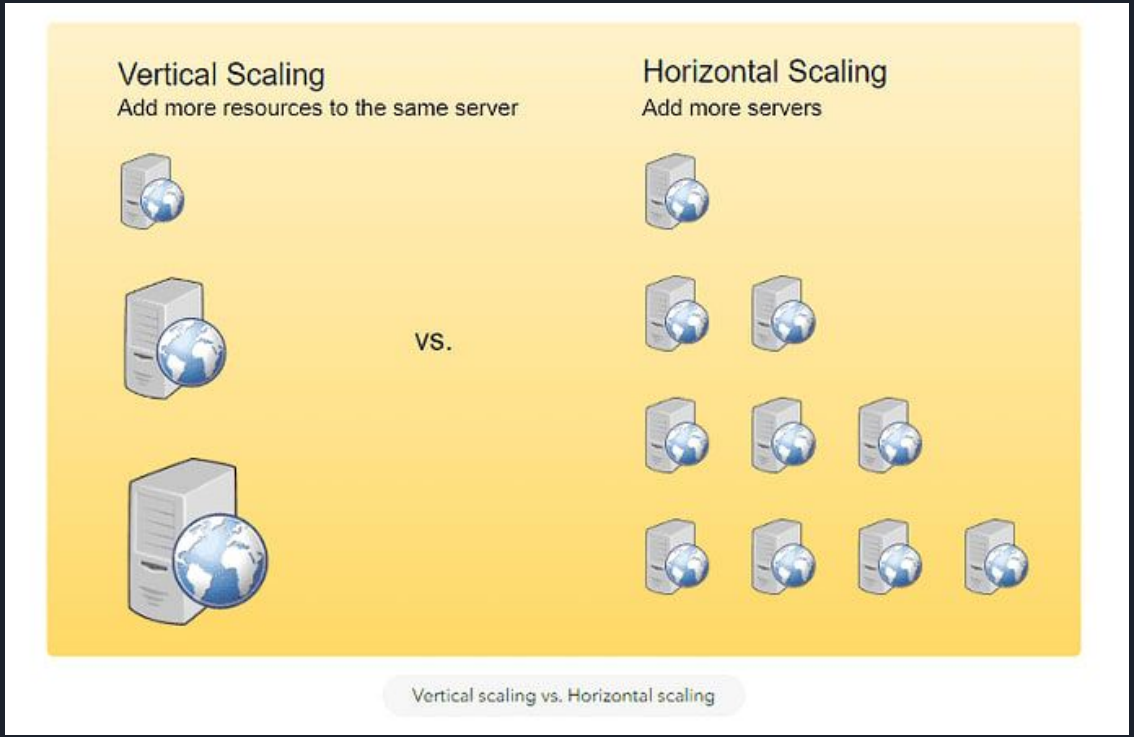
- Presentation Tier -> Application Tier -> Data Tier
- These tiers can be scaled up/down independently
- Each tier can be developed concurrently by different team of programmers in different programming languages

Peer to Peer Architecture

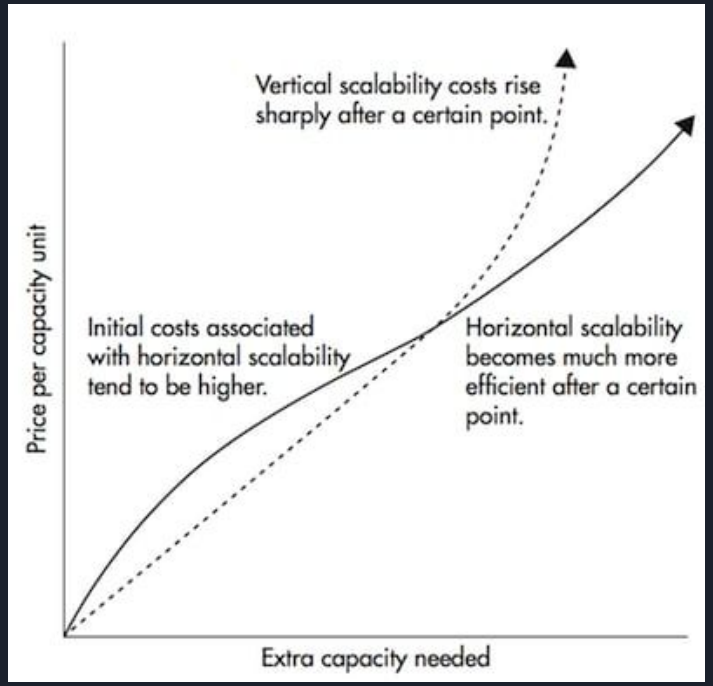


- No centralized or special machine that does the heavy lifting and intelligent work
- All the decision making and responsibilities are split up amongst the machines involved
- Each could take on client or server roles.
- Blockchain is a good example of this.

Why Distributed Systems



Why Distributed Systems



- Enables indefinite scaling(theoretically)
- Cheaper to scale because of commodity hardware
- Computation speed
    - Parallel computation on multiple computers
- No more single point of failure

WOMEN WHO CODE

Characteristics of Distributed Systems

- Scalability
  - Scalability is the capability of a system, process, or network to grow and manage increased demand.
  - Horizontal and vertical scaling
  - Reasons for scaling: Increased data volume, Increased workload

- Reliability
  - Reliability is the probability a system will not fail in a given period
  - The system will continue to work even when one or more components fail
  - Cheap commodity hardware - Failed nodes can be replaced easily
  - Achieved through redundancy of software components and data

- Availability
  - Availability is when a system remains operational to perform its required function in a specific period
  - Percentage of time a system is operational

Characteristics of Distributed Systems

- Efficiency
  - Latencies (Response time)
  - Throughput (Number of requests handled)
  - Decisions to scale the system up are usually based on these factors

- Serviceability or Manageability
  - Ease of maintenance and operations
  - Speed at which the system can be maintained or repaired
  - High maintenance can result in low availability
  - Ease of diagnosis, sufficient logging, sufficient operational metrics, intelligent monitoring and alerting, etc

Challenges of Distributed Systems

- Increased opportunities for failure
  - In a poorly designed system, a single node crash can bring the whole system down
  - Fault tolerance is not foolproof
- Synchronization between processes
  - No global clock
  - In a complex multi user system (Eg: Online video games), synchronization is a challenge because it is also dependant on public network which carries data
- Scalability and performance are not linear
  - 2x number of nodes doesn't mean 2x performance
  - Designing a distributed system with linear scalability is super complicated and has to take into account load balancing, bandwidth management, redundancy management, etc
- Security is complex
  - A single weak link in one node can expose the whole system to attack

Distributed Systems Categories

- Distributed Datastore
- Distributed Computing
- Distributed File system
- Distributed Messaging
- Distributed Ledger
- Distributed Applications

Distributed Systems Categories

Datastore	Apache Druid, Elasticsearch, Cassandra, DynamoDb
Computing	Hadoop MapReduce, Apache Spark, Apache Storm, Apache Flink, Apache Samza, Kafka streams
File System	HDFS - Hadoop Distributed File System
Messaging	Apache Kafka, Rabbit MQ, Amazon Kinesis
Ledger	Blockchain, Bitcoin
Applications	BitTorrent, Netflix, google.com, Most modern large scale systems

WOMEN WHO CODE

Distributed Datastore

- Most NOSql Databases
- Different types: key value store, graph based datastore, column based datastore, document based datastore

Cassandra

- Key value datastore
- Uses consistent hashing
- Massively scalable and high write throughput

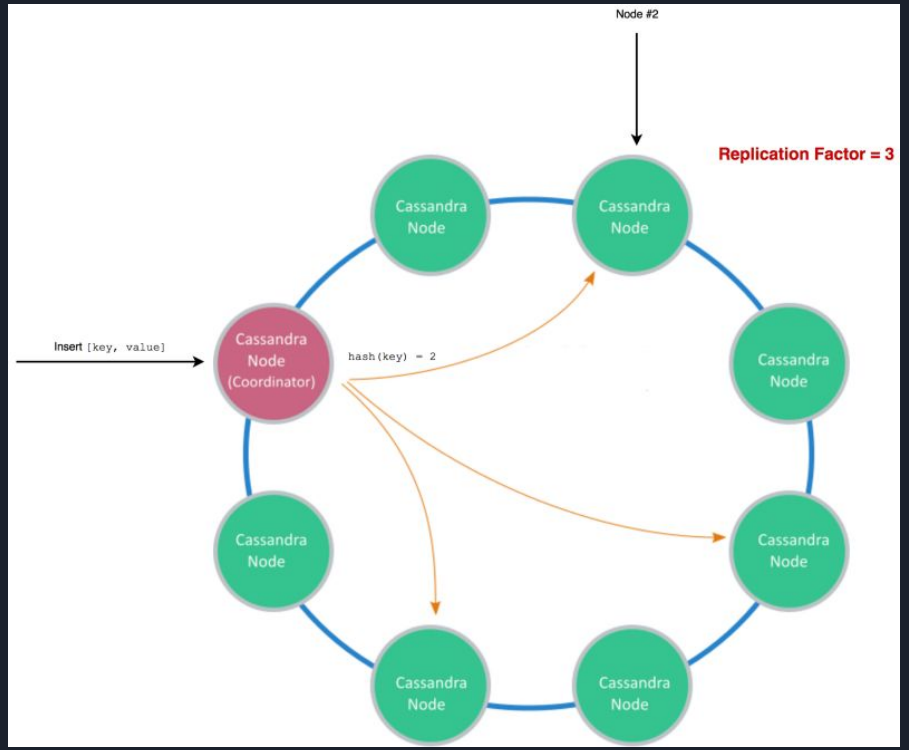


Image Credit

WOMEN WHO CODE

Distributed Computing

- Key to modern scale high volume data processing
- Splitting enormous task across multiple systems which no one computer can handle

MapReduce

- Founded by Google
- Two stages: Map and Reduce
- Each map node works on subset of data
- Intermediate step: Shuffle (Arrange and partition data)
- Final stage: Reduce - Each node works subset of partitions

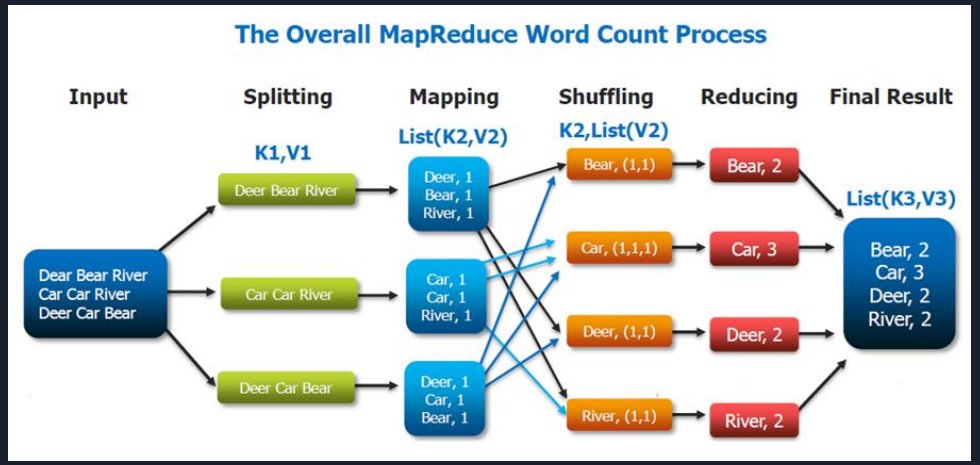


Image Credit

WOMEN WHO CODE

Distributed File System

- Similar to Distributed Datastores
- Large amount of files are stored across multiple nodes

HDFS

- Namenodes and Datanodes
- Namenode - Has all metadata about cluster
- Namenode acts as coordinator
- Namenode is responsible for data placement
- Datanodes - This is where data resides
- Datanodes are responsible for storing, deletion and replication of data blocks

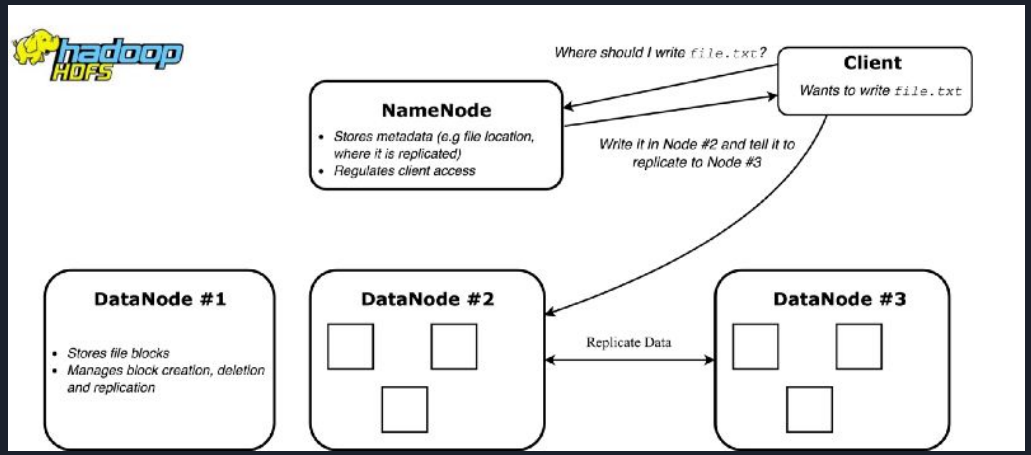


Image Credit

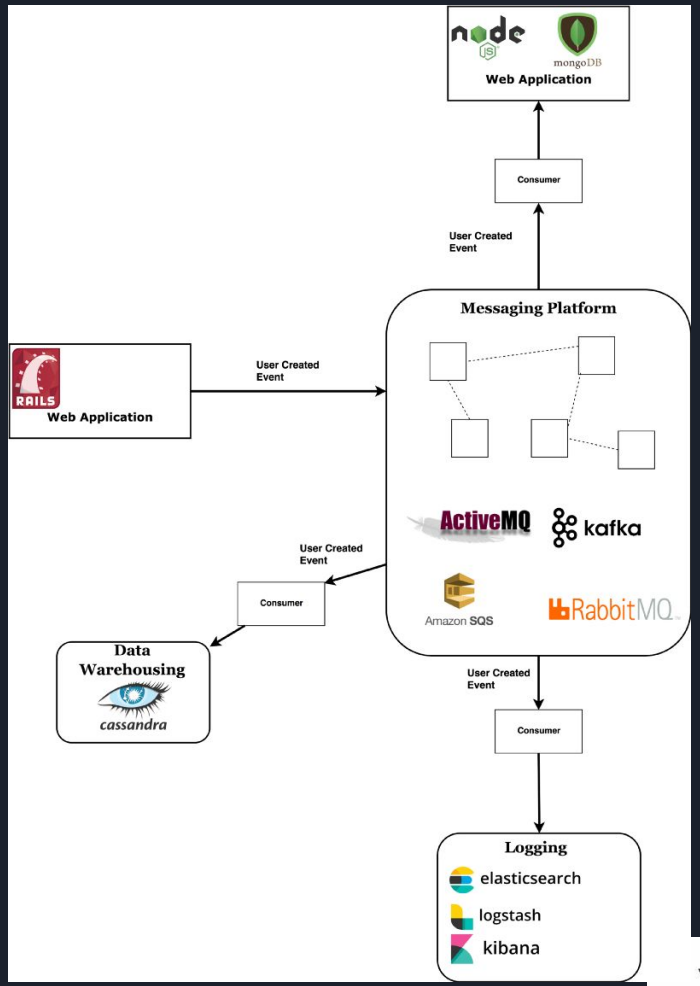
WOMEN WHO CODE

Distributed Messaging

- Distributed message queues
- Used for storage and propagation of enormous number of events in a centralized place
- Decouples application logic from directly talking to other systems

Apache Kafka

- Collection of nodes called brokers
- Producers and Consumers
- Data is organized as topics
- Topics are split into partitions
- [Introduction to Kafka slides]

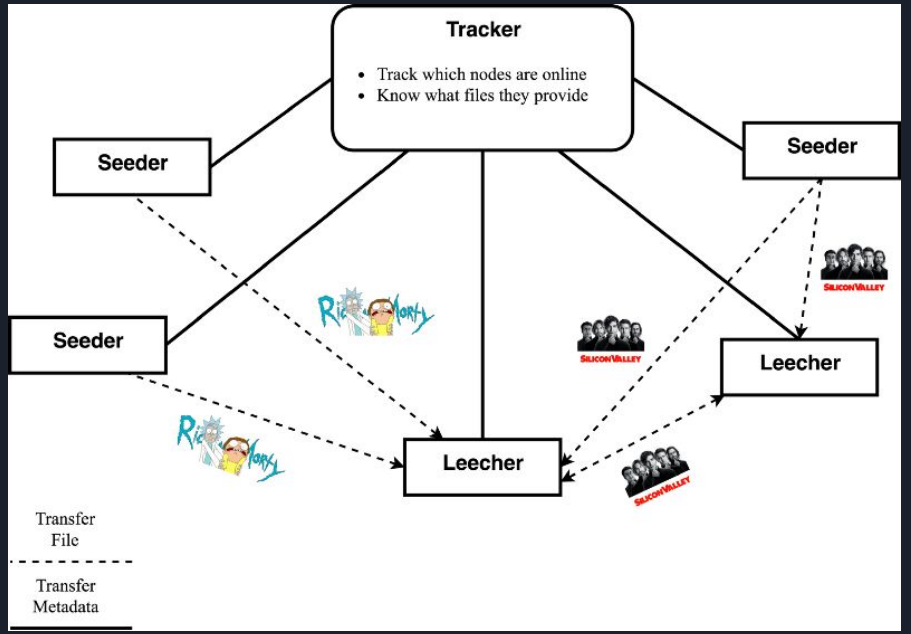


[Image Credit]

WOMEN WHO CODE

Distributed Application

- Any application whose back end is run across multiple nodes with distributed storage and computation is a distributed application
- Appears as one system to the end user
- Example: Facebook, Netflix, Whatsapp, Instagram, etc



BitTorrent

- Famous for downloading pirated movies :)
- Peer to peer network
- Tracker: Coordinator which coordinates a download/upload
- Seeder: Source of distributed files
- Leecher: Downloaders of the files

Image Credit

WOMEN WHO CODE

Distributed Ledger

- Peer to peer network
- Immutable, append-only database that is replicated, synchronized and partitioned across all nodes in the distributed network

Blockchain for Bitcoin

- Blockchain is the underlying technology
- Distributed ledger carries ordered list of transactions
- Transactions are grouped and stored in blocks across nodes in the network
- Blocks are computationally expensive to create and tightly linked to each other using cryptography
- Miners create these blocks using extensive CPU

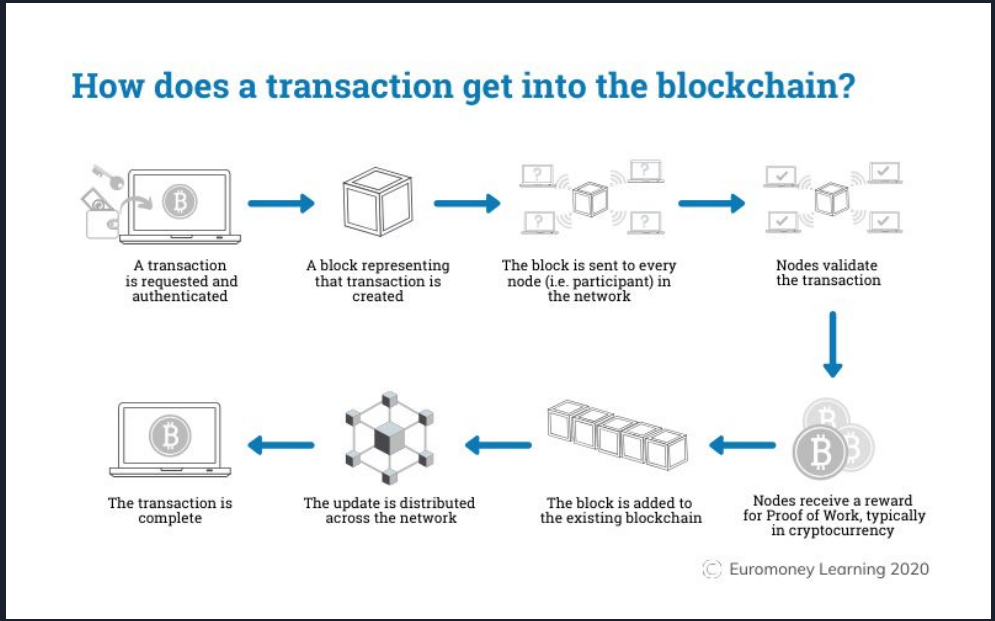
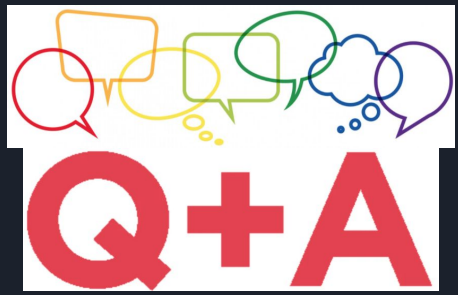


Image Credit

Backend Study Group

WWCode Slack Handle: Harini Rajendran

https://www.linkedin.com/in/hrajendran/



Resources and References:

- https://www.techtarget.com/searchsoftwarequality/definition/3-tier-application
- https://www.confluent.io/learn/distributed-systems/
- https://www.toolbox.com/tech/cloud/articles/what-is-distributed-computing/
- https://github.com/WomenWhoCode/wwcsf-backend-study-group/blob/main/meetupevents/2-3-2022/SQL%20vs%20NoSQL%20databases_WWCode_Backend%20Study%20Group_2-3-2022.pdf
- https://github.com/WomenWhoCode/wwcsf-backend-study-group/blob/main/meetupevents/4-14-2022/Introduction%20to%20Apache%20Kafka.pdf
- https://dev.to/mukulalpha/system-design-101-everything-to-know-about-distributed-systems-2hd6
- https://www.freecodecamp.org/news/a-thorough-introduction-to-distributed-systems-3b91562c9b3c/
- https://pipeandpiper.co.uk/2021/08/16/what-is-cryptocurrency/

Backend Study Group:

• Presentations and session recordings found here: WWCode YouTube channel

You can unmute and talk or use the chat.

WOMEN WHO CODE