



Distributed & Mobile Computing

IS384

Introduction to Distributed Systems



Ubiquitous Networks

Ubiquitous networks, often referred to as ubiquitous computing or ubiquitous networking, describe a concept where computing capabilities are seamlessly integrated into our everyday environment.

Introduction to Distributed Systems



Ubiquitous Networks

In a ubiquitous network, various devices and systems are interconnected, allowing information and communication to flow effortlessly across different platforms and locations.

Introduction to Distributed Systems



Distributed Systems:

Distributed systems are networks of independent computers that work together to achieve a common goal, emphasizing cooperation and resource sharing.

Introduction to Distributed Systems

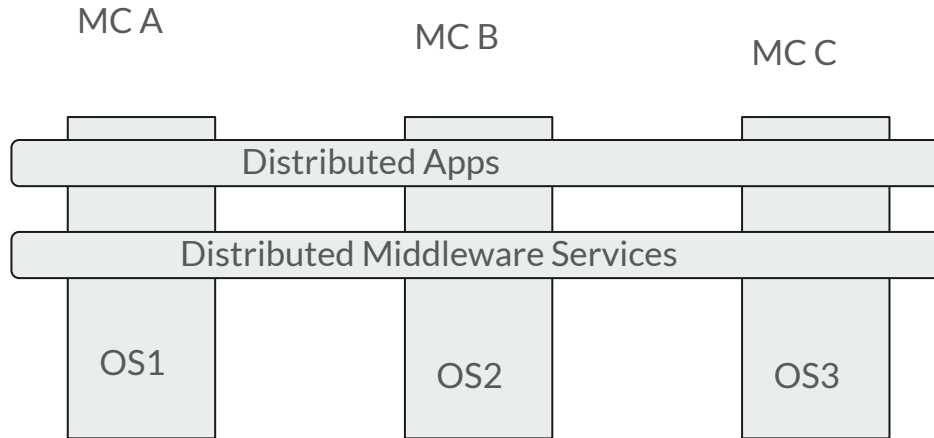


Distributed Systems:

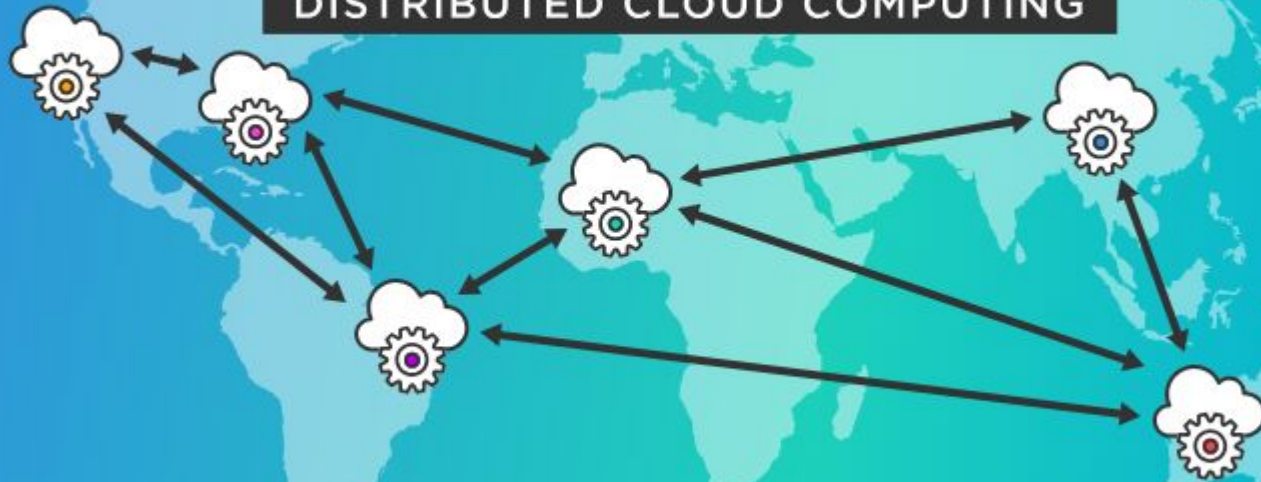
Distributed systems is a collection of autonomous computers linked by a computer networks and equipped with distributed systems softwares.

The software enables computer to coordinate their activities to share resources of system hardware, software and data.

Introduction to Distributed Systems



DISTRIBUTED CLOUD COMPUTING



Characteristics of Distributed Systems



- **Concurrency**
 - concurrent programs execution – share resource
- **No global clock**
 - programs coordinate actions by exchanging messages
- **Independent failures**
 - when some systems fail, others may not know

Characteristics of Distributed Systems



- **No shared memory**
 - This is a key feature that requires message-passing.
- **Geographical separation**
 - The geographically wider apart that the processors
- **Autonomy and heterogeneity**
 - The processors are “loosely coupled” in that they have different speeds and each can be running a different operating system.

Goals of Distributed Systems

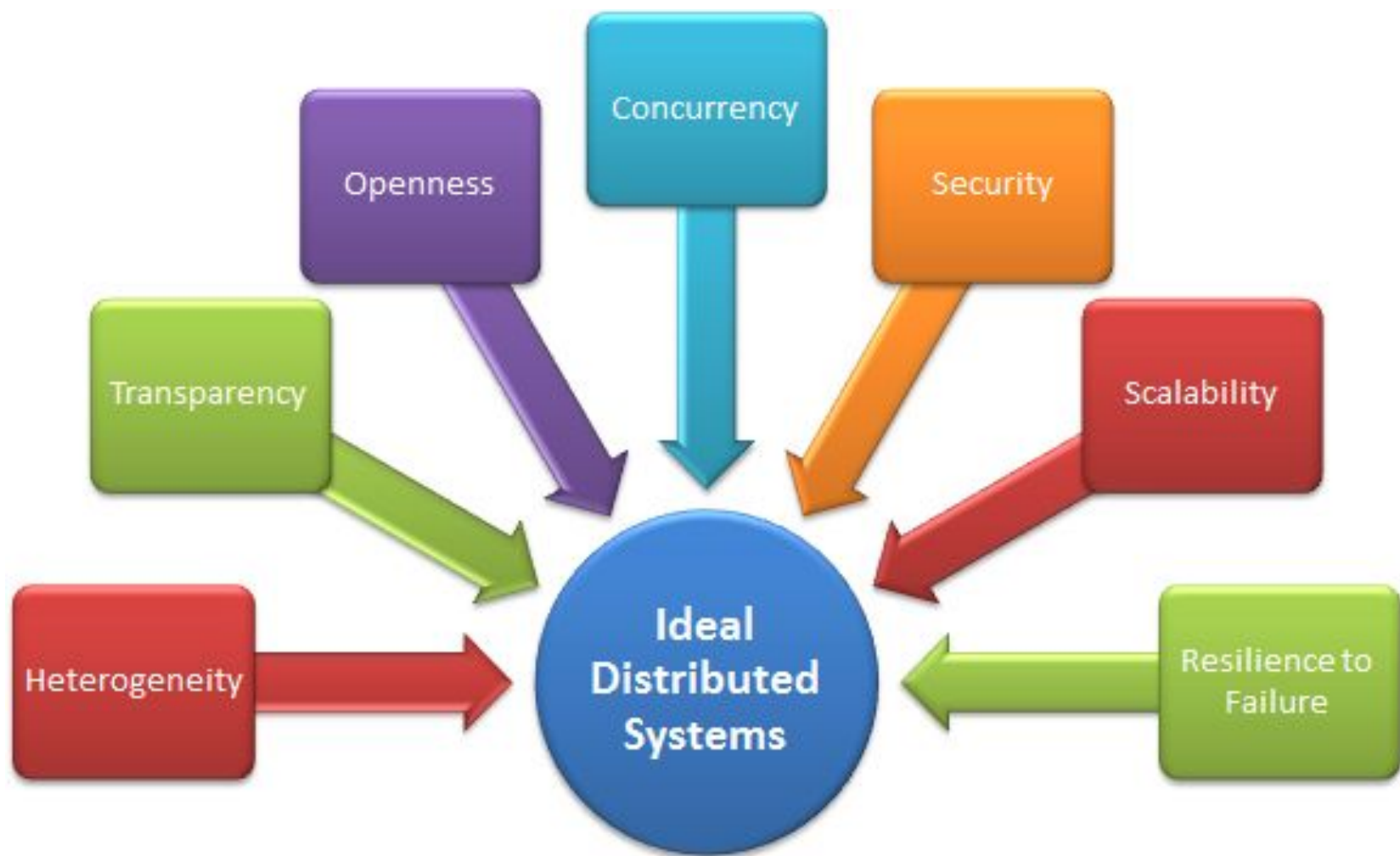


- **Reliability**: Ensure uninterrupted operation despite hardware or network failures.
- **Scalability**: Efficiently handle increased workloads by distributing tasks across multiple nodes.
- **Transparency**: Make the distributed system appear as a cohesive entity to users and applications.
- **Interoperability**: Enable seamless communication between diverse system components.
- **Security**: Implement robust measures to protect data and ensure secure communication.

Goals of Distributed Systems



- **Efficiency:** Minimize resource usage, reduce latency, and maximize throughput.
- **Fault Tolerance:** Design the system to tolerate and recover from unexpected failures.
- **Maintainability:** Facilitate easy system maintenance and updates to minimize downtime.
- **Transparency:** Hide the complexity of distributed operations from end-users.



Examples of Distributed systems



- **World Wide Web (WWW):** The internet itself is a distributed system where servers, databases, and clients work together to deliver web content and services.
- **Cloud Computing Platforms:** Services like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform leverage distributed systems to provide scalable and flexible computing resources.
- **Distributed Databases:** Systems like Apache Cassandra, MongoDB, and Amazon DynamoDB distribute data across multiple nodes for improved scalability and fault tolerance.
-

Examples of Distributed systems



- **Social Media Platforms:** Platforms like Facebook, Twitter, and Instagram use distributed systems to handle user interactions, data storage, and content delivery.
- **Content Delivery Networks (CDNs):** CDNs, such as Akamai and Cloudflare, distribute content across multiple servers globally to enhance the speed and reliability of content delivery.
- **Blockchain Networks:** Cryptocurrencies like Bitcoin and Ethereum rely on distributed ledger technology to maintain decentralized and secure transaction records.

Examples of Distributed systems



- **Distributed File Systems:** Examples include Google File System (GFS), Hadoop Distributed File System (HDFS), and Apache HBase, which distribute data storage across multiple nodes.
- **Peer-to-Peer Networks (P2P):** File-sharing networks like BitTorrent and communication systems like Skype use distributed architectures where nodes communicate directly with each other.
- **Online Banking Systems:** Banking systems often use distributed databases and networks to handle transactions, customer accounts, and ensure availability.