Introduction

Definition: Distributed Systems

A collection of independent components, including

- Hardware
- Software Components
- Web services

Which work together and appear to its users as a single coherent system

1 Comparison

Networks:

- Focus: method and implementation of communication
- Data transmission: addressing, scheduling, quality control and error recovery

Distributed Systems:

- Components: heterogeneous (OS, H/W, S/W), physically loosely coupled, but are required to collaborate
- System architecture: transparency, performance, reliability
- Network technology: means to connect and organise DS components

2 Applications of distributed systems

- Rapid system development, reuse existing components or services
- Construct low cost and scalable system: link up many low cost machines to work together as a coherent system
- Improve system reliability or availability: based on redundant components

3 Design issues

Network issues:

- Highly variable bandwidth
- Possibly large and variable latency

Distributed event issues:

- No native support to clock synchronization among hosts
- Difficult to ensure data consistency

System failure issue:

- Unpredictable failures of components, may be due to failure of a network component or communication delay incurred on the communication path

Security issues

- Centralized systems:
  - Can rely on physical security
  - Users understand what trust to assign to the system
  - System administrators are responsible
- Distributed systems:
  - None of the above applies
  - Hard to know what is being trusted or what can be trusted

4 Transparency requirement

Hide the separation of components or their organisation and changes

Types of transparency	Description
Access	Hide differences in data representations and how a component is accessed
Location	Hide where a component is located
Migration	Hide that a component or a resource may move to another location(content distribution network)
Relocation	Hide that a component or a resource may be moved to another location while in use (caching)
Replication	Hide that a component or a resource may have multiple copies(provide fast or local access)
Concurrency	Hide that a resource may be shared by several competitive users(handle shared resources)
Failure	Hide the failure and recovery of a resource or a component