Characterization of Distributed Systems

The components of a DS are located at *networked computers* and communicate and coordinate by passing *messages*

Chapter objectives

1. Introduction
2. Examples of distributed systems
3. Trends in distributed systems
4. Focus on resource sharing
5. Challenges
6. Case study: The World Wide Web

Study of this chapter

1. **Leverage the value of this lecture:**
   - Attempt solving the questions about Networking included at the beginning of Questionnaire SG-DS-1.0.pdf (Questionnaire no.1 of DS; paloalto.unileon.es/ds/SG/SG-DS-1.0.pdf).
   - Take class notes

2. **Have the textbook by Kindberg *et al.* at hand**
   - The physical book is in our Library
   - A pdf versión is available

3. **After the lecture**, solve all the questions included in the Questionnaire mentioned above
   - If you need assistance
     - After that, bring your questions to the next **B1** session
     - Attend one of the **Tutor Teaching** sessions

What is a Distributed System?

- A Distributed System is made up of a set of networked computers (physically separated) that coordinate their actions by only interchanging messages and which objective consists of sharing resources
  - Processes running at different hosts are **concurrent (Actually parallel)**
  - Hosts coordinate by **exchanging messages**, only
  - Accuracy of clock synchronization is limited, **no global clock**
  - **Independent failures**: Other nodes may still run after one fails
  - **Resource sharing** is the main motivation

Examples of Distributed Systems

Finance and commerce	eCommerce e.g. Amazon and eBay, PayPal, online banking and trading
The information society	Web information and search engines, ebooks, Wikipedia; social networking: Facebook and MySpace.
Creative industries and entertainment	online gaming, music and film in the home, user-generated content, e.g. YouTube, Flickr
Healthcare	health informatics, on online patient records, monitoring patients
Education	e-learning, virtual learning environments; distance learning
Transport and logistics	GPS in route finding systems, map services: Google Maps, Google Earth
Science	The Grid as an enabling technology for collaboration between scientists
Environmental management	sensor technology to monitor earthquakes, floods or tsunamis

Examples: Google Web search

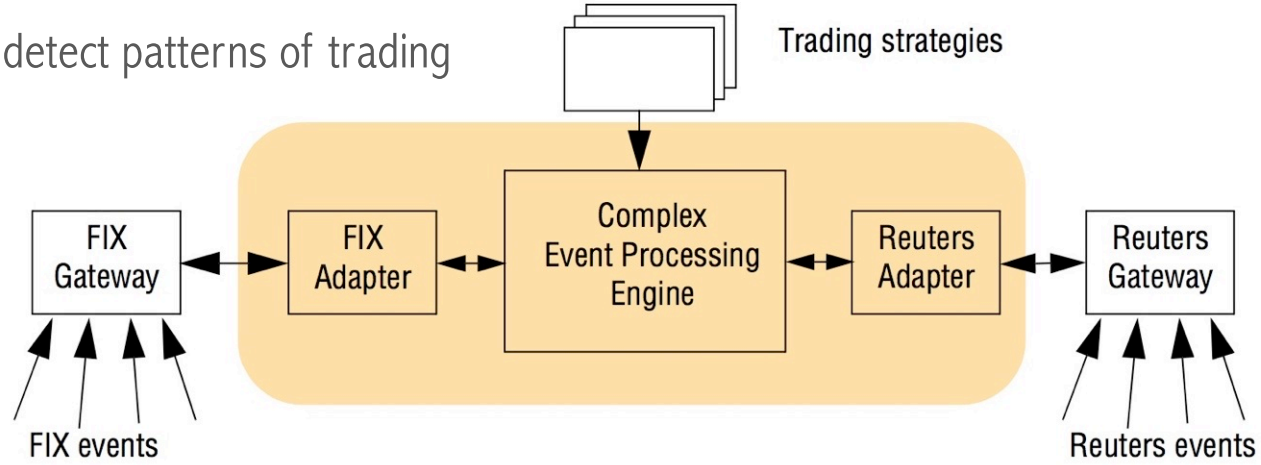
- Index the contents of the entire web (63 billion pages)
- A major challenge, one of the most complex DS in history
- Highlights
  - Large number of nodes
  - Distributed file system
  - Distributed storage system
  - Distributed file locking
  - Parallel programming allowed

Examples: MMOGs

- Massively Multiplayer Online Games
- Over 50000 simultaneous players
- Fast response times
- Real time propagation of events to many players
- Solutions proposed:
  - 1. Client-server architecture, a single copy of the state of the world
    - Server: A powerful cluster
    - Load is partitioned
  - 2. Geographycally distributed architecture (Physical. Proximity)
  - 3. Radically new architectures based on peer-to-peer computing (decentralized)

Examples: Financial trading

- Cutting edge distributed systems used in this industry
- Essential: Reliable and and timely delivery of events
  - Architecture: Distributed event-based systems
  - FIX (Financial Information eXchange protocol)
    - Support heterogeneity
    - Adapter systems
    - Real time processing to detect patterns of trading
- Manage risk
- Regulations compliance
- Fraudulent transactions



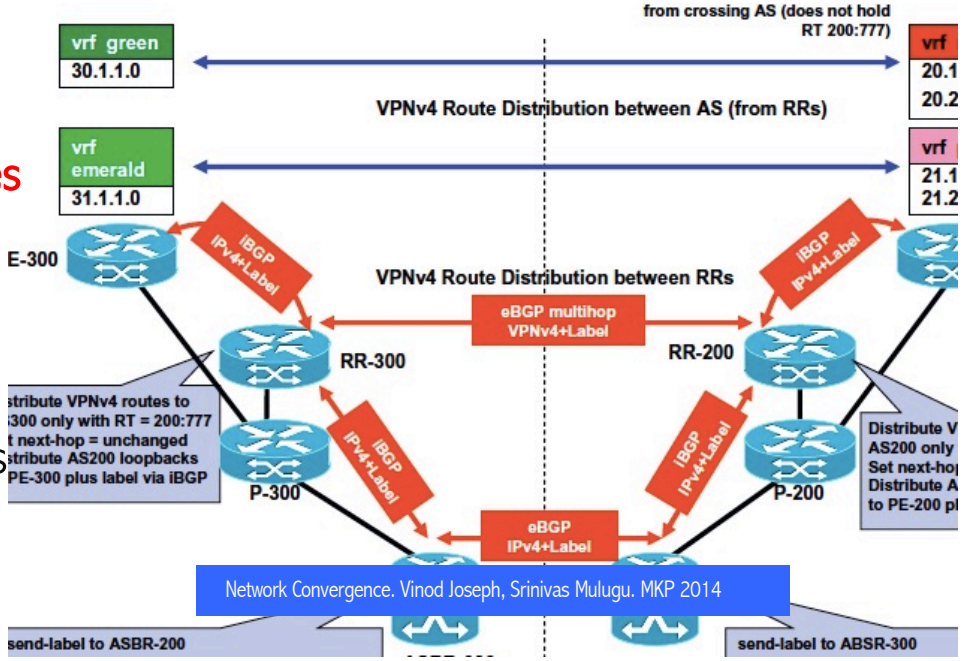
Trends in DS

DS are undergoing significant changes at present

- Pervasive Networking and Internet
- Mobile and Ubiquitous Computing
- Distributed Multimedia Systems
- Distributed Computing as a Utility

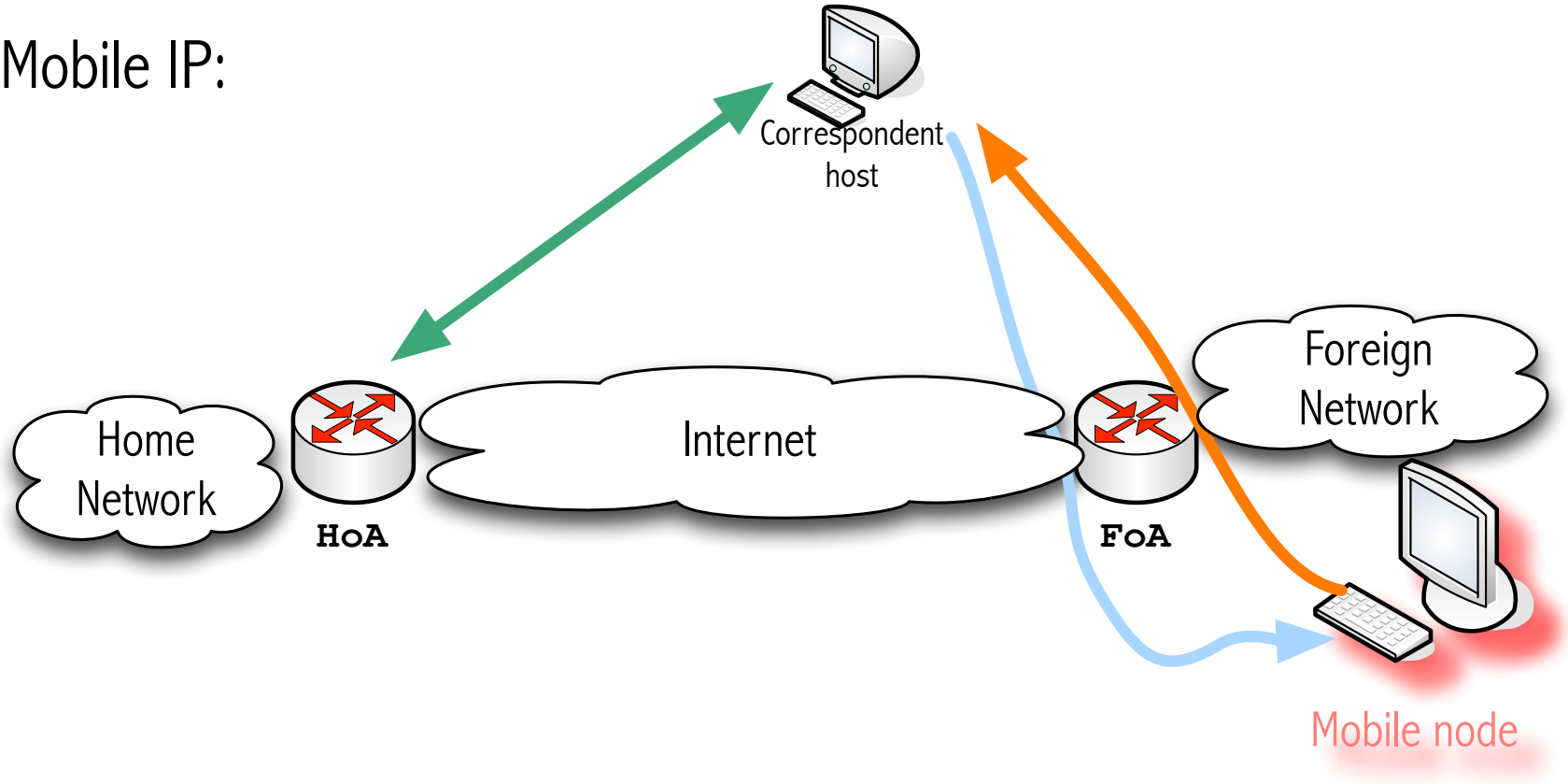
Pervasive Networking, Modern Internet

- Modern networking technologies
  - WiFI, WiMax, LTE, MPLS
- Networking is becoming pervasive
  - Affecting more and more aspects of persons lives
- Protocols
  - Abstracting a myriad of technologies
  - Programs can run anywhere and send messages to any node
- Web ≠ Internet
- Intranets, firewalls: filter incoming and outgoing messages



Mobile Computing

- Users who are away from their 'home' network
  - Can still access shared resources on their home networks
- Mobile IP:

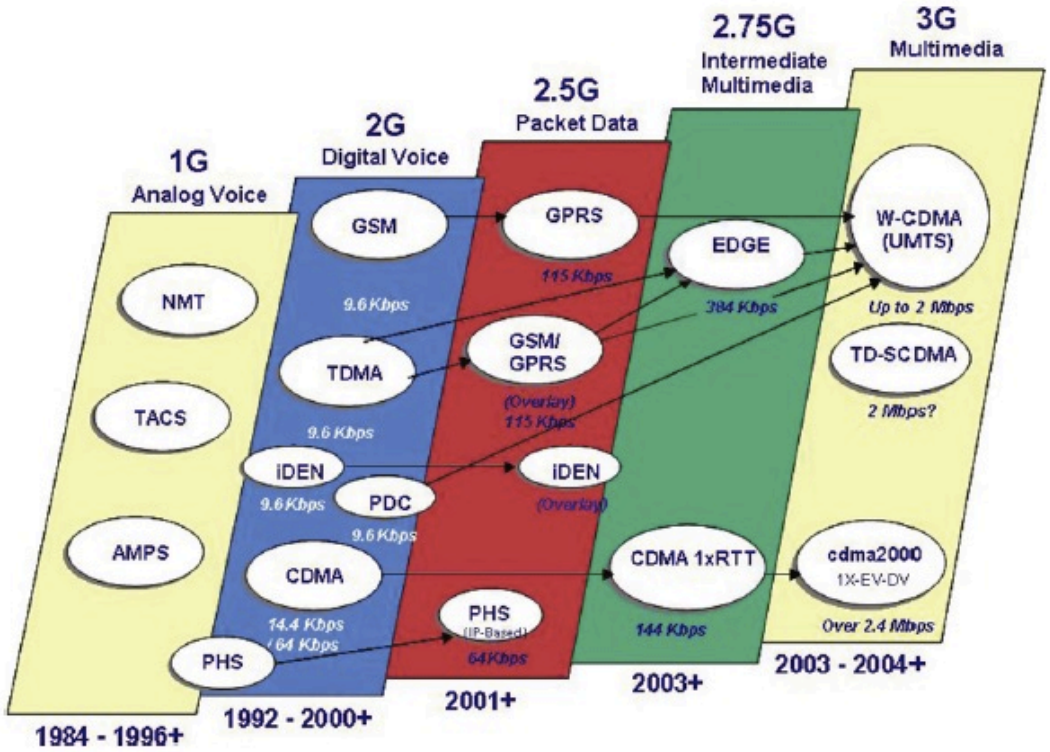


Ubiquitous Computing

- Connecting the cheap computing devices present in the physical environment
  - Home, office and natural things
  - IOT = Internet Of Things
  - IOE = Internet Of Everything
- Spontaneous interoperation
- Service discovery
- Device associations

Distributed Multimedia Systems

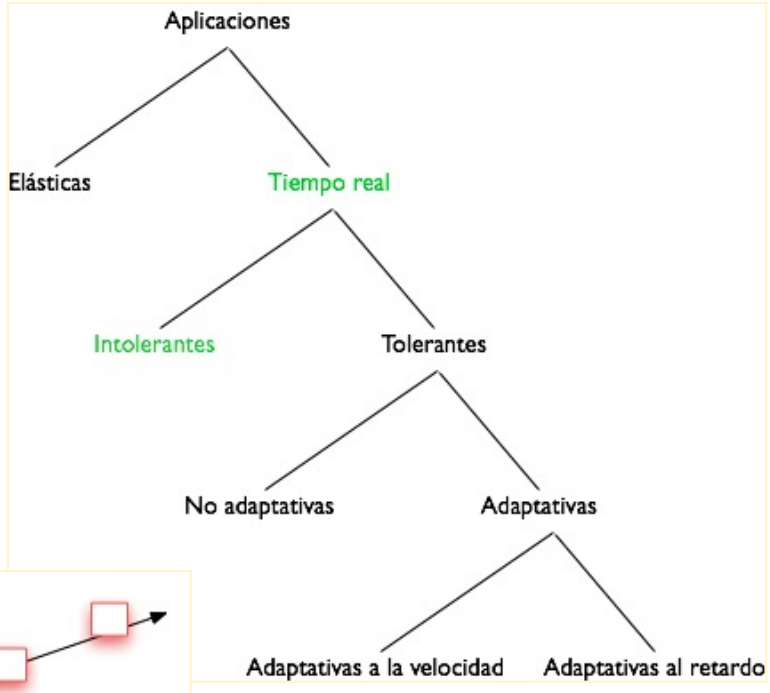
- Multimedia
  - Discrete media
    - Pictures
    - Text
  - Continuous media (Temporal dimension, considerable demands)
    - Audio
    - Video
    - IP telephony
    - WebCasting
    - Video Conferencing

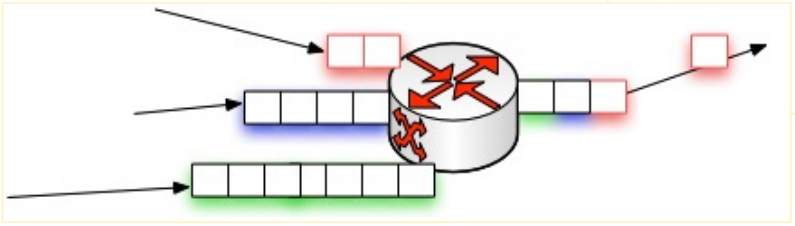


Network Convergence. Vinod Joseph, Srinivas Mulugu. MKP 2014

Quality of Service (QoS)

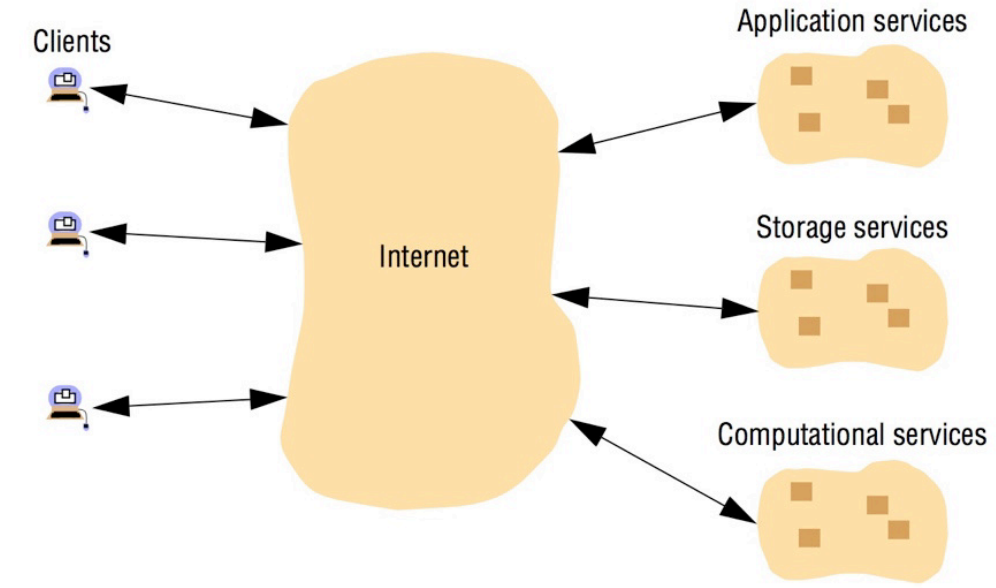
- Applies to Network and Operating Systems alike
- Non-functional requirements
  - Reliability
  - Security
  - Performance
    - Response time
    - Throughput
    - Delay and jitter
    - Loss





Distributed computing as a utility

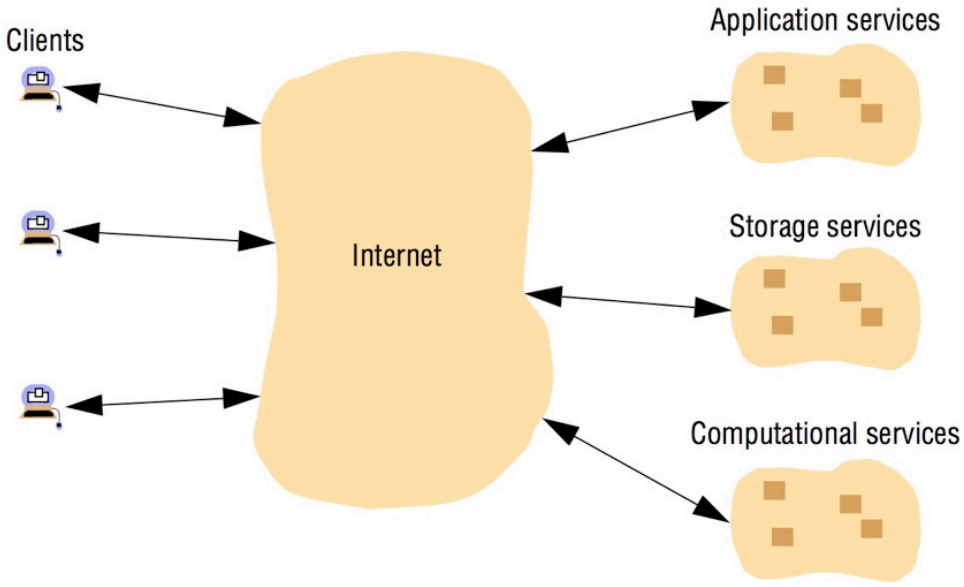
- Distributed resources as a commodity or a utility
  - Regardless of company or brand
    - Water supply
    - Electricity
    - Wheat
- Resources are rented, not owned
- Physical resources
  - Sophisticated Data Centers
  - OS virtualization:
    - Virtual nodes, not physical nodes
    - Flexibility
- Software services



From Coulouris, Dollimore, Kindberg and Blair. Distributed Systems: Concepts and Design, Edition 5, © Addison-Wesley

Utility computing = *Cloud computing*

- Everything is a service
- Reduces requirements of users devices
- Implemented on cluster computers
  - CLUSTER: High speed LAN interconnection of hosts collaborating in speeding up a large computational load
    - Blade servers in a rack



From Coulouris, Dollimore, Kindberg and Blair. Distributed Systems: Concepts and Design, Edition 5, © Addison-Wesley

Focus: *Resource Sharing*

- When Distributed Systems are programmed in an Object-Oriented language (Java; C++)
  - Resources are encapsulated as objects
    - Client objects
    - Server objects

Challenges: Heterogeneity

heterogeneity | ˌhɛt(ə)rə(ʊ)dʒɪˈniːɪti |

noun *[mass noun]*

the quality or state of being diverse in character or content: *the genetic heterogeneity of human populations.*

- Networks
- Hardware
- OS
- Languages
- Implementations
  - Masked by the abstractions of Communication Protocols

- What about data types?
  - Before data can be exchanged there must be an agreement about their physical representation: *Marshalling*
- Middleware
  - CORBA is a specification of which many implementations exist
  - JAVA RMI is a part of the Java platform

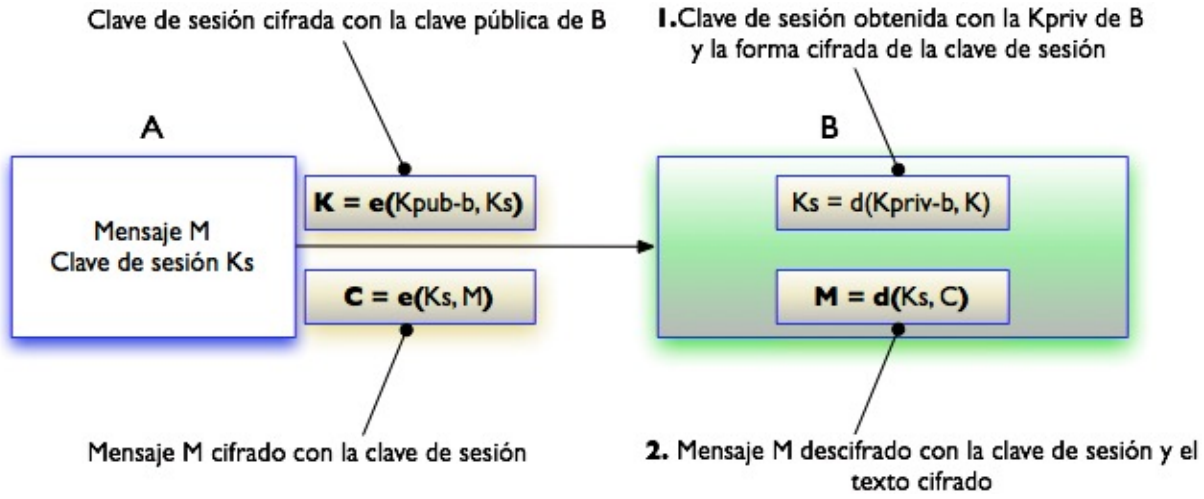
Challenges: Openness

Refers to DS components specification and documentation

- If key interfaces are public, then:
  - New implementation of DS is possible
  - RFCs
- Open distributed systems are extensible
- What is then an Open Source System? Is an OSS *open?*

Challenges: Security and scalability

- Of paramount importance today
  - Confidentiality
  - Integrity
  - Availability
  - Authentication



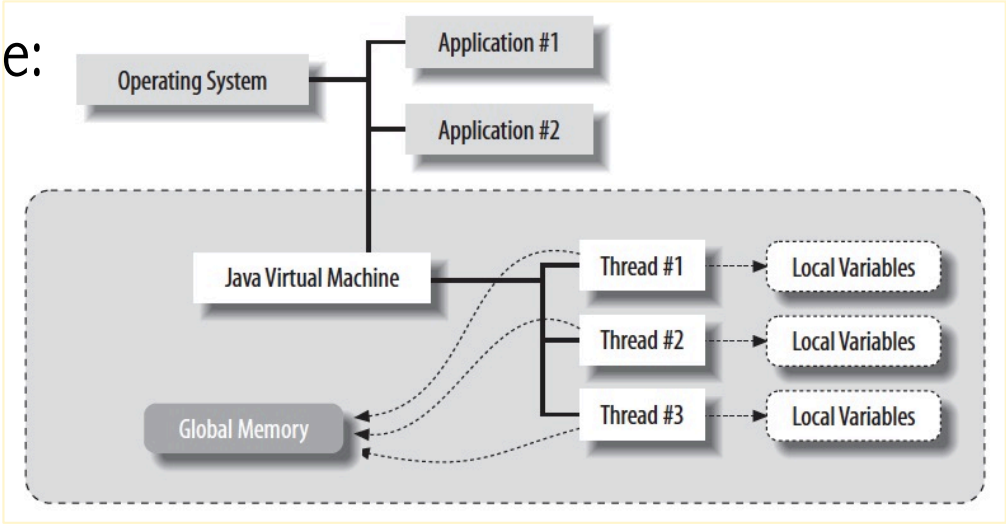
- Scalability
  - System performance will be O(n) if more resources are added
    - n users → O(n) servers then, system is scalable

Failures in DS

- Detecting failures
  - CRC, checksum
- Tolerating failures
- Recovery
- Redundancy
  - Redundant components
    - Two routes to any router
    - DNS
    - Databases
- High availability

Concurrency

- Several clients will attempt to access a resource simultaneously on a server
- If server serves one request at a time:
  - Limits throughput
  - But, will guarantee data consistency
- If server can serve several requests concurrently
  - Improves thorughput, but
  - Concurrent operations (By several threads) on one object *may* interact among themselves and produce inconsistent results
  - Non-deterministic, in general



Distributed System *transparencies*

The separation of components of a DS remains hidden to the user and the programmer

- **Access transparency**

  Enables local and remote resources to be accessed using identical operations

- **Location transparency**

  Enables resources to be accessed without knowledge of their physical or network **location** (for example, which building or IP address)

Distributed-System Transparencies

The separation of components of a DS remains hidden to the user and the programmer

- **Concurrency**

  Enables several processes to operate concurrently using shared resources without interference between them

- **Replication**

  Enables multiple instances of resources to be used to increase reliability and performance without knowledge of the replicas by users or application programmers

Distributed-System Transparencies

The separation of components of a DS remains hidden to the user and the programmer

- **Failure**

  Enables the concealment of faults, allowing users and application programs to complete their tasks despite the failure of hardware or software components.

- **Mobility**

  Allows the movement of resources and clients within a system without affecting the operation of users or programs

Distributed-System Transparencies

The separation of components of a DS remains hidden to the user and the programmer

- **Performance**

  Allows the system to be reconfigured to improve performance as loads vary.

- **Scaling**

  Allows the system and applications to expand in scale without change to the system structure or the application algorithms.