Distributed Computing

Intro

Schedule

- 9 lectures
- 18 labs

- Attestation:
  - All Labs + Test

Who Am I?

Gleb Radchenko

- Ph.D. in Physical and Mathematical Sciences
- Associate Professor of System Programming Department
- Deputy Dean of the Faculty of Computational Mathematics and Computer Science
- Deputy Head of supercomputer simulations Laboratory

The main topics of the course

1. Fundamentals of Distributed Computing Systems
2. Multi-tier client-server architecture
3. Organization of interaction of remote systems
4. Service-oriented architecture (SOA)
5. Cloud computing

Fundamentals of Distributed Computing Systems

Distributed system

«A distributed system is one on which I cannot get any work done because some machine I have never heard of has crashed»

Leslie Lamport,

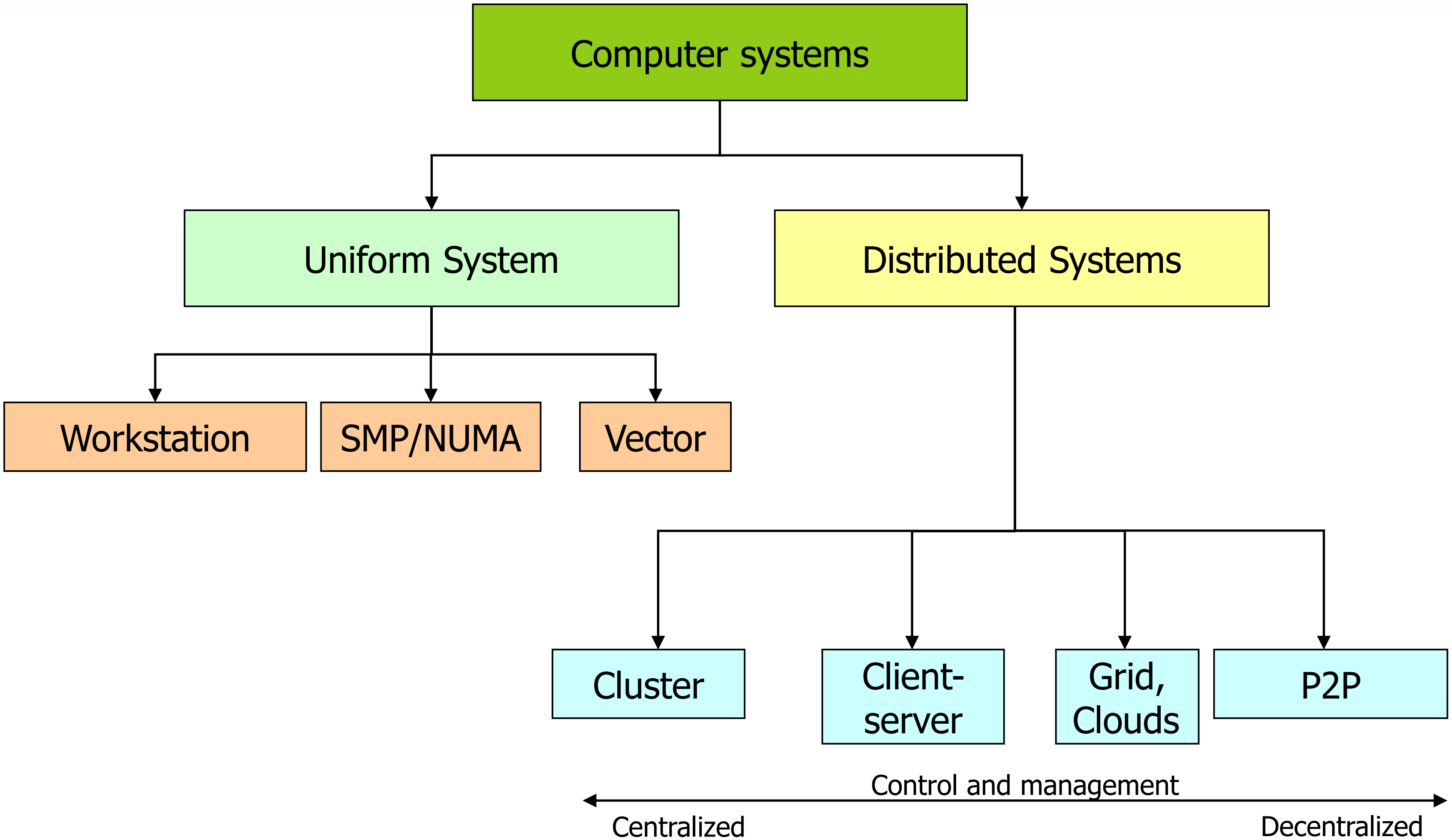
Microsoft Corporation

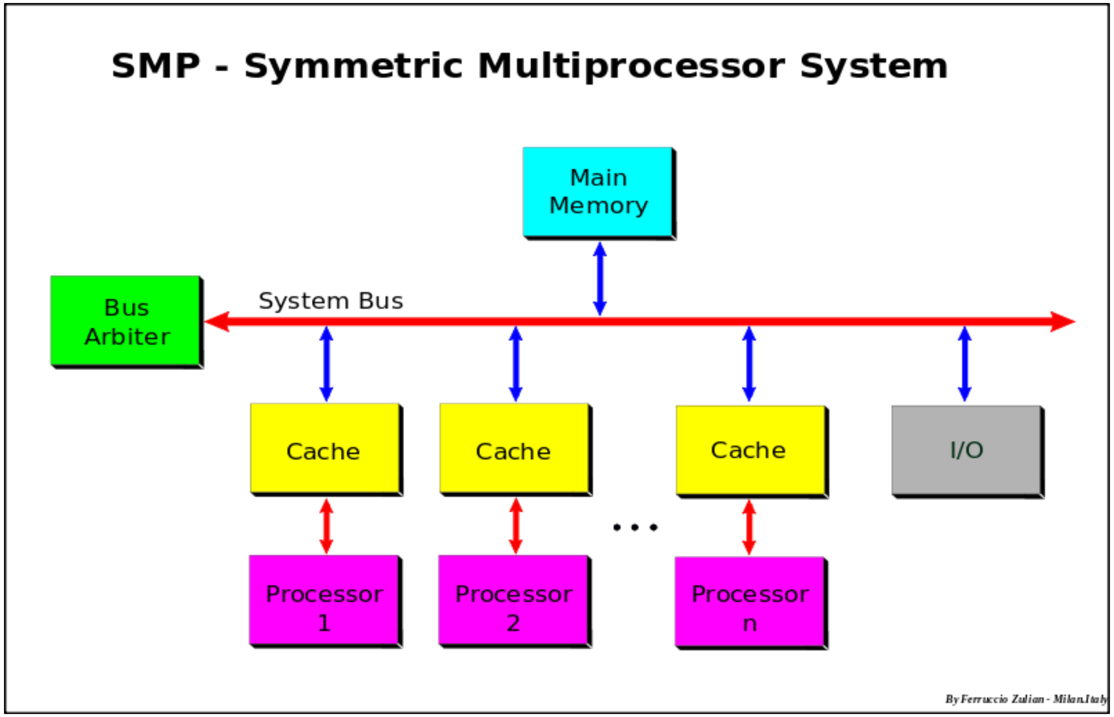
Distributed system

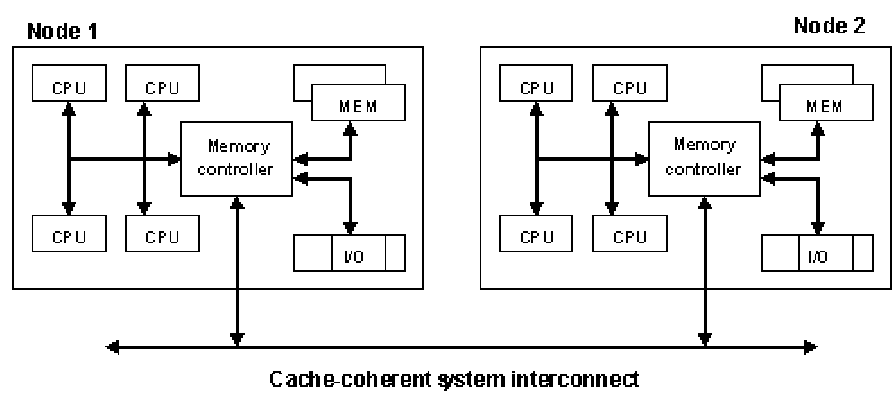
«**A distributed system is** a collection of independent computers that appears to its users as a single coherent system.»

Tanenbaum & Van Steen, Distributed Systems

Types of computer systems



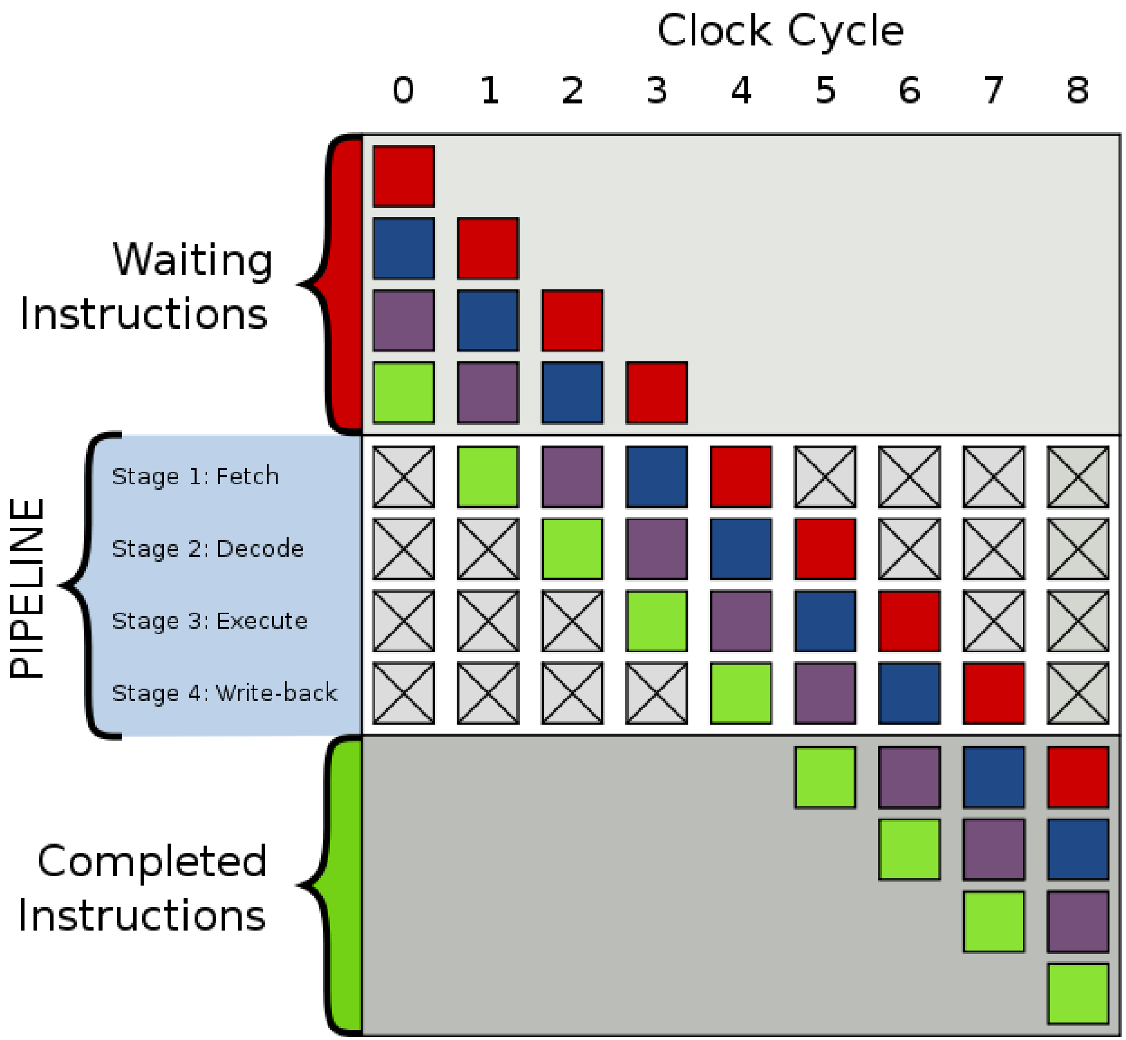




NUMA

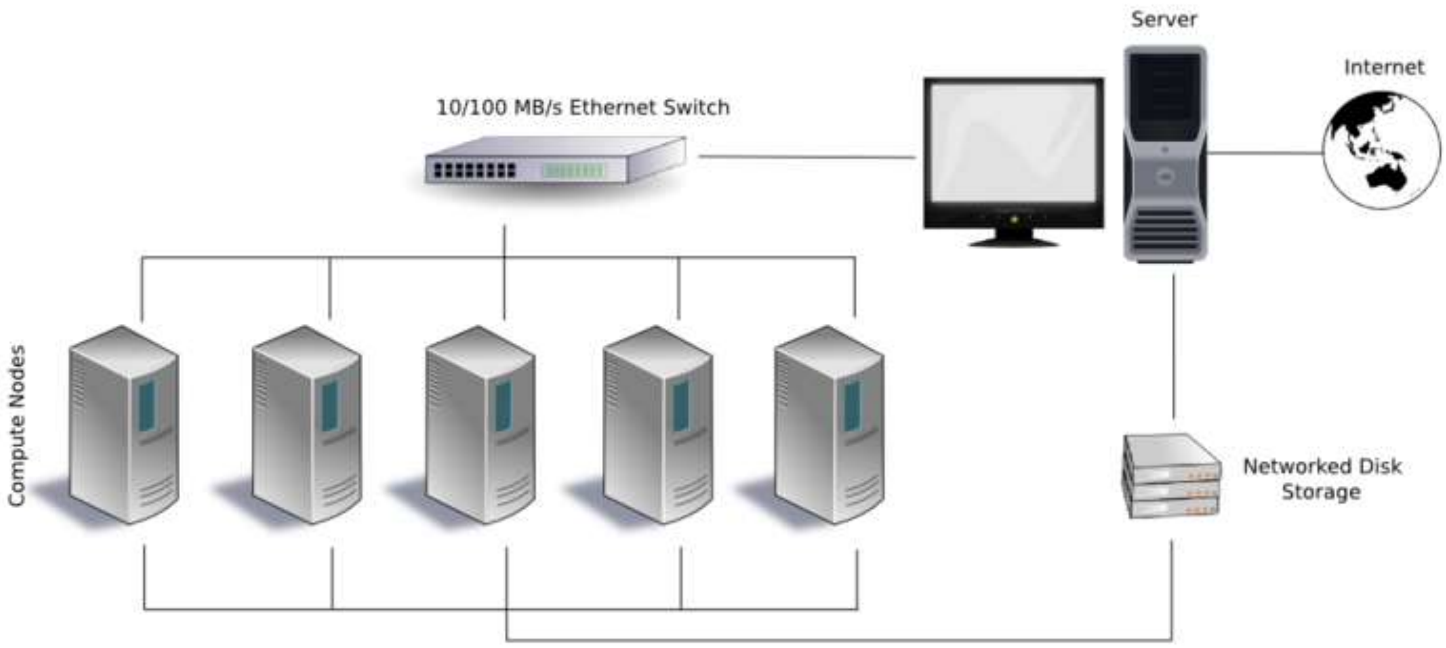
wikipedia

Vector computing



wikipedia

CLUSTER



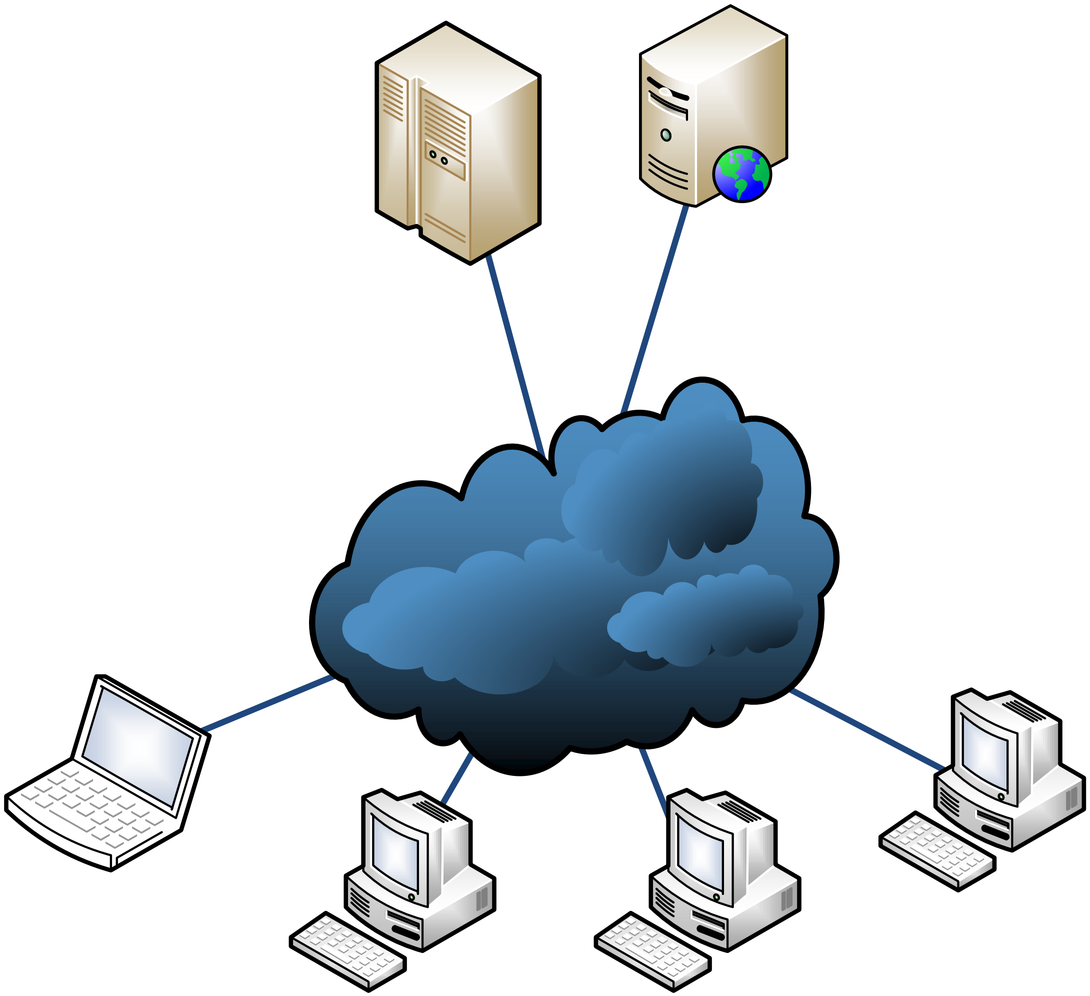
wikipedia

Cluster

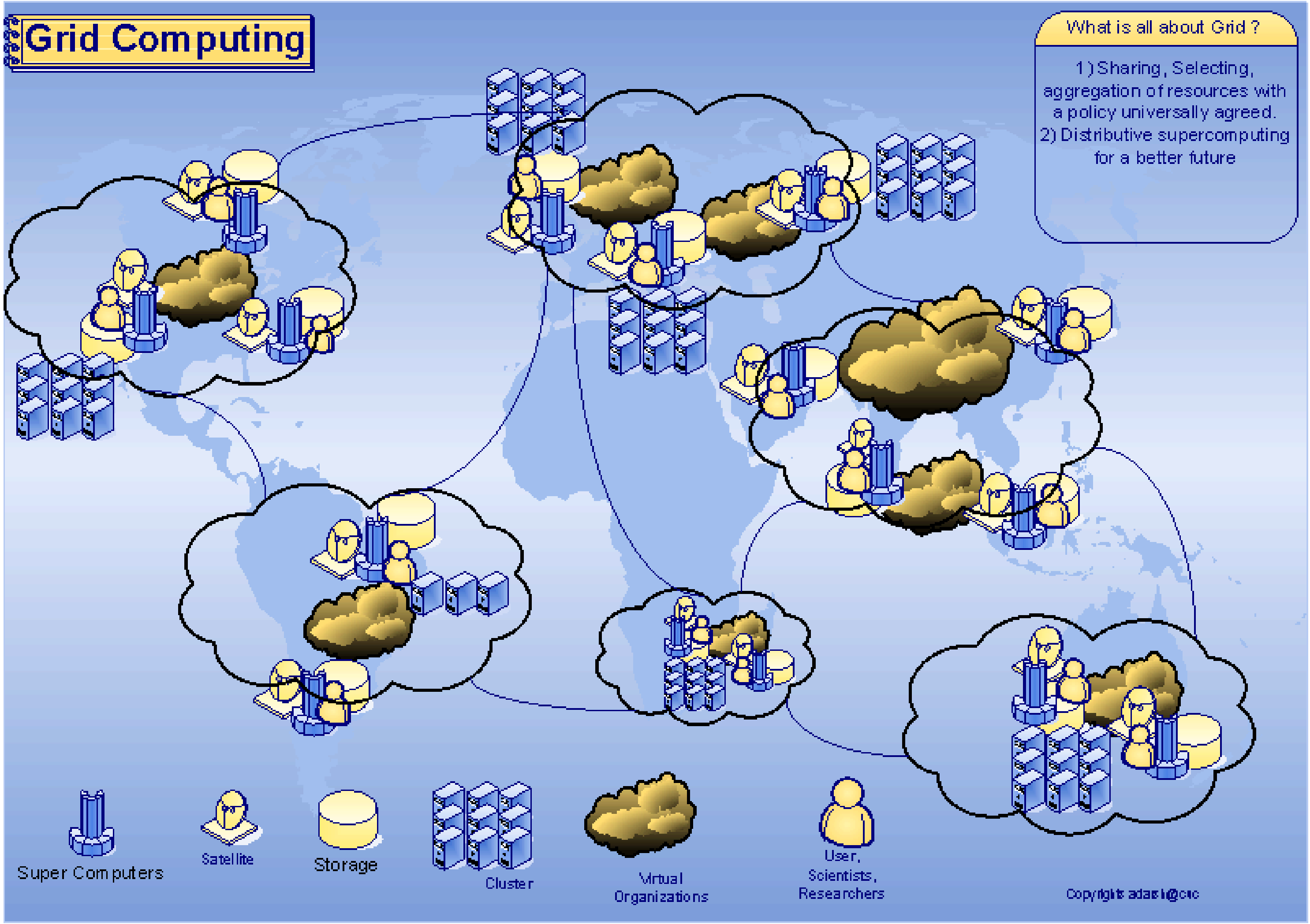
Tornado SUSU computing cluster

Tornado SUSU computing node

Client-Server

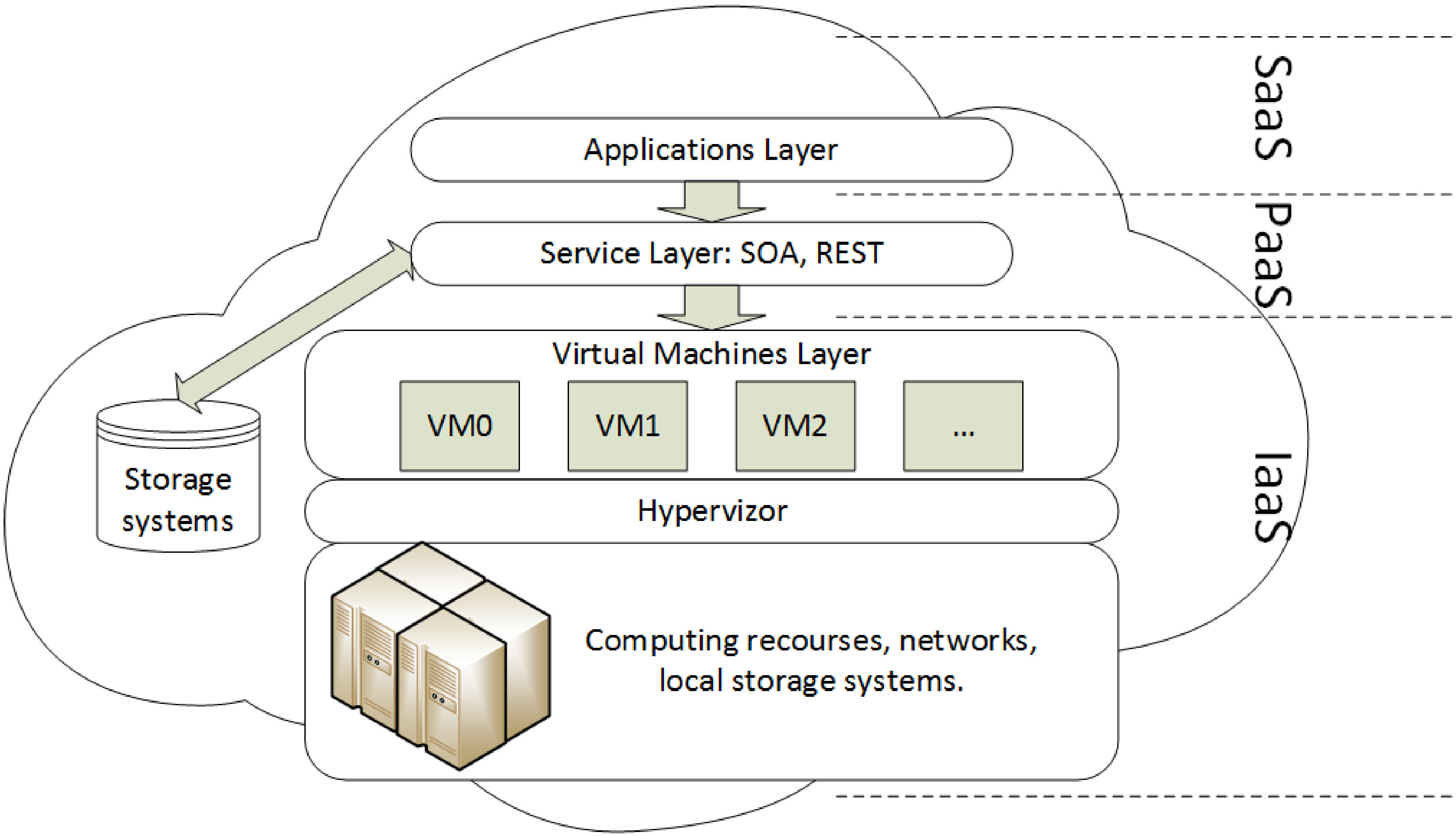


Grid computing

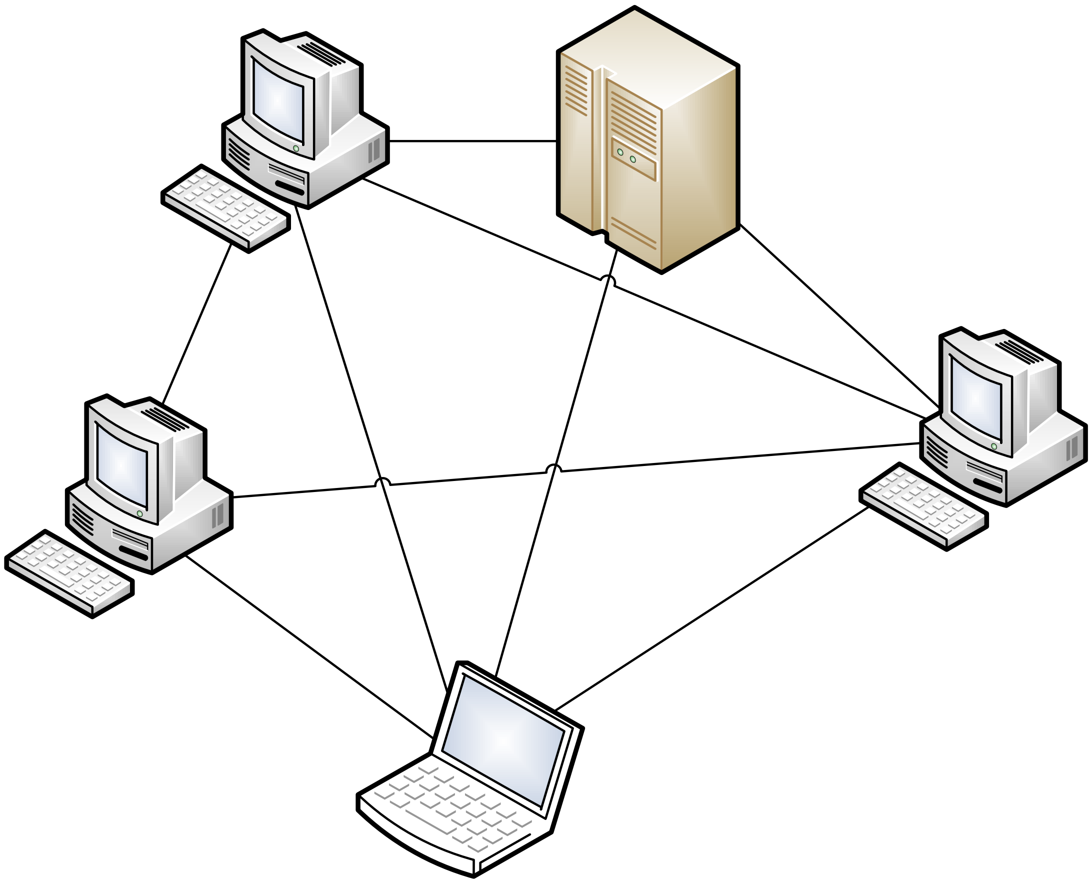


Adarsh Grid Computing Research http://www.adarshpatil.com/newsite/research.htm

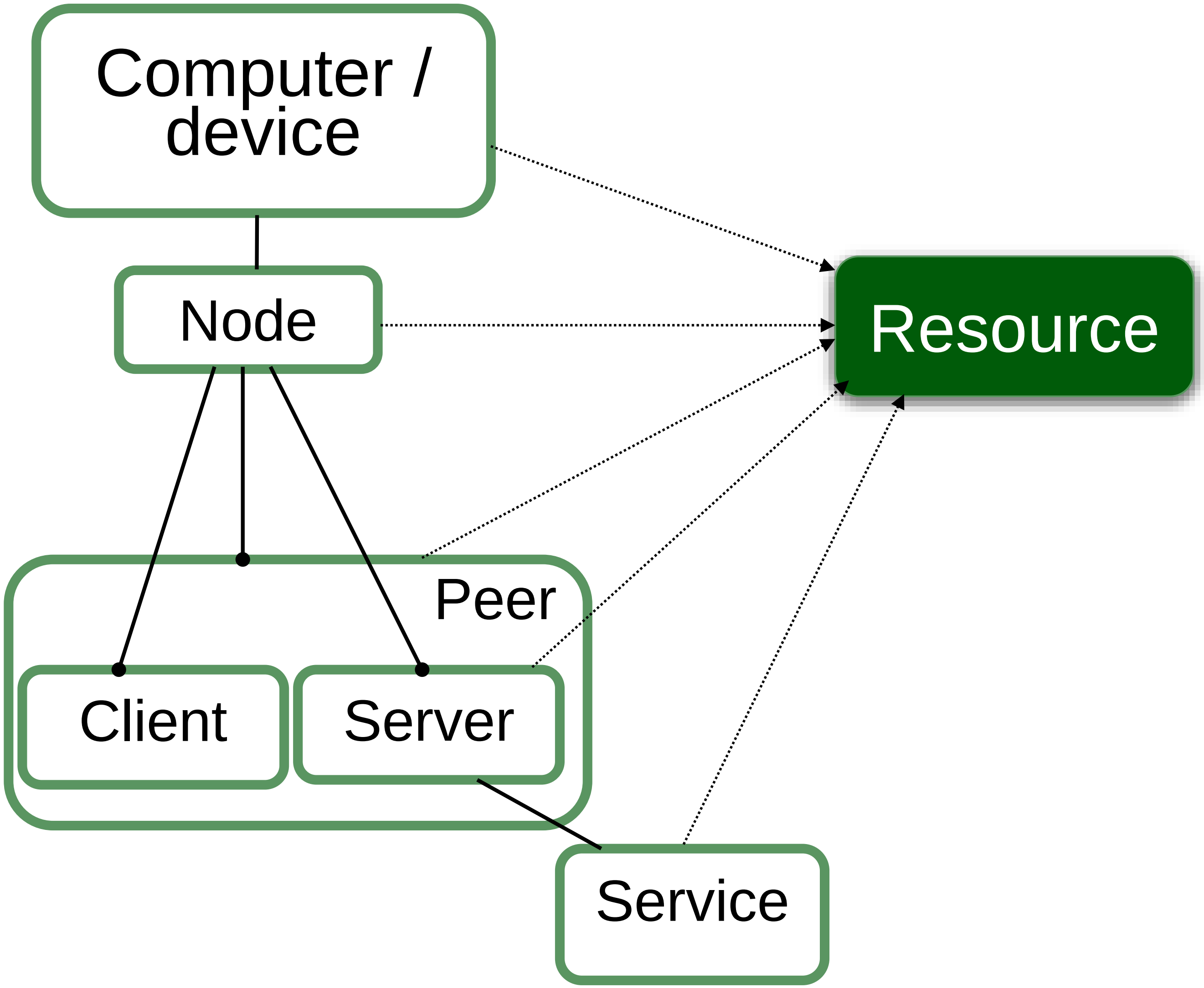
Cloud computing



P2P computing



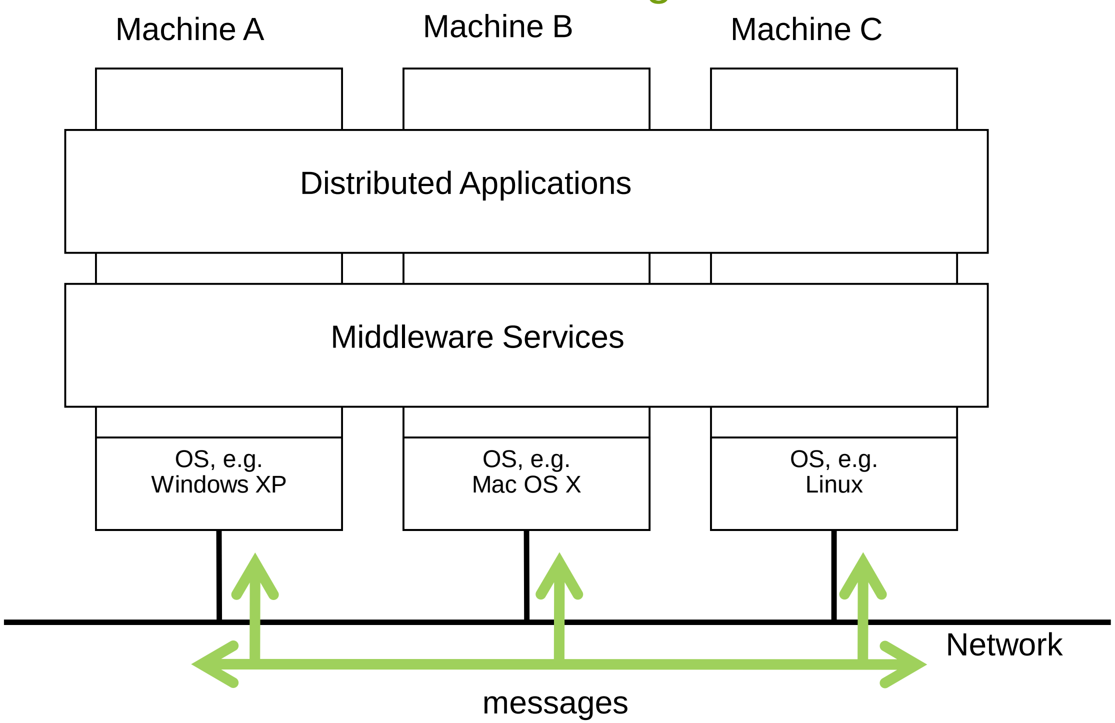
KEY TERMS



A Distributed System

A distributed system is a collection of independent computers that appears to its users as a single coherent system.

- Heterogeneous computers – vendors/OS should be able to interoperate
- Should mask the heterogeneity from users (applications)
- Should be easy to expand and scale
- Should be permanently available (even though parts of it are not)
- Communication is based on messages



Distributed Systems Issues

- Distributed systems are inherently different from non-distributed systems.
  - Latency - network speed
  - Memory access - not shared
  - Partial Failure
    - remote failure does not mean local failure
    - no global coordination (like an OS)
  - Guaranteed Concurrency
    - combined with latency, events are not received in the same order as they are generated
  - Indeterminacy
    - Your system is not in control of the whole system
    - With partial failure, a system may just disappear with no indication of status.
    - was it the remote machine or a network link?

Check a latency

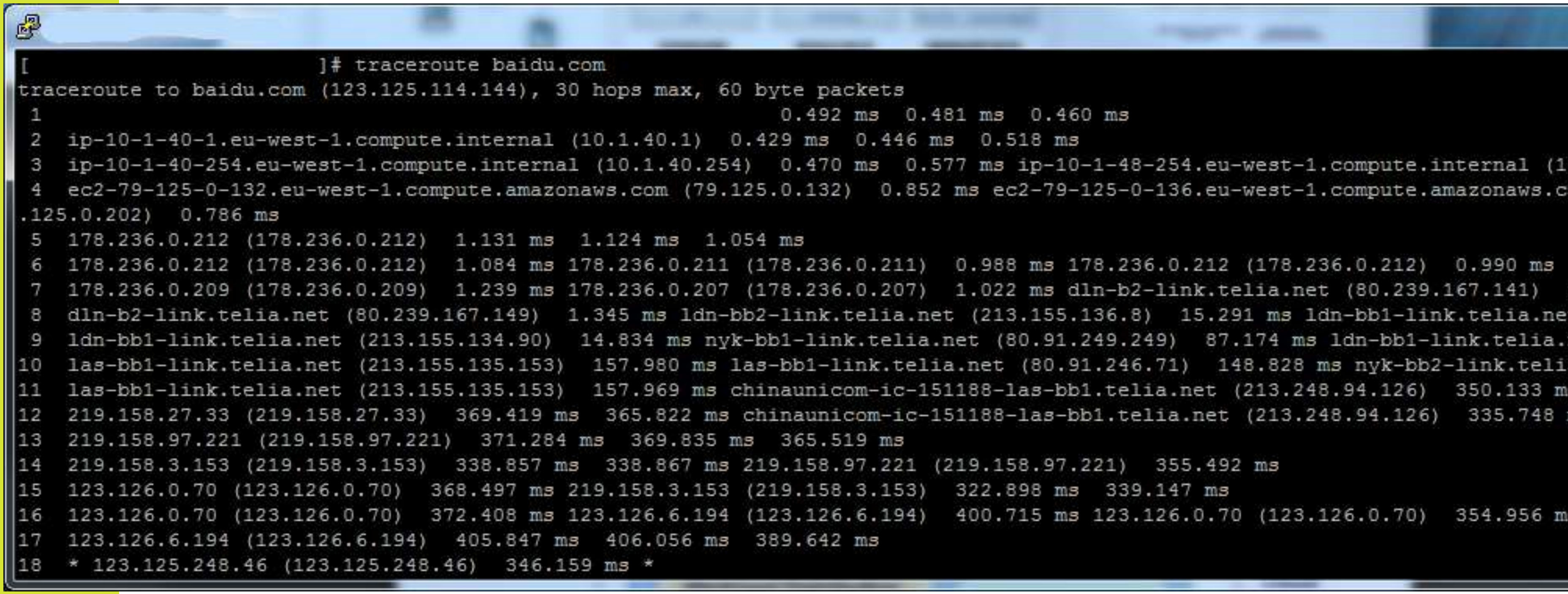
- Start -> run -> ping [URI or IP-address]

```
[                        ]# ping google.com
PING google.com (74.125.138.100) 56(84) bytes of data.
64 bytes from dn-in-f100.1e100.net (74.125.138.100): icmp_seq=1 ttl=44 time=14.4 ms
64 bytes from dn-in-f100.1e100.net (74.125.138.100): icmp_seq=2 ttl=44 time=14.6 ms
64 bytes from dn-in-f100.1e100.net (74.125.138.100): icmp_seq=3 ttl=44 time=14.5 ms
64 bytes from dn-in-f100.1e100.net (74.125.138.100): icmp_seq=4 ttl=44 time=14.4 ms
64 bytes from dn-in-f100.1e100.net (74.125.138.100): icmp_seq=5 ttl=44 time=14.4 ms
64 bytes from dn-in-f100.1e100.net (74.125.138.100): icmp_seq=6 ttl=44 time=14.4 ms
64 bytes from dn-in-f100.1e100.net (74.125.138.100): icmp_seq=7 ttl=44 time=14.4 ms
64 bytes from dn-in-f100.1e100.net (74.125.138.100): icmp_seq=8 ttl=44 time=14.5 ms
64 bytes from dn-in-f100.1e100.net (74.125.138.100): icmp_seq=9 ttl=44 time=14.3 ms
64 bytes from dn-in-f100.1e100.net (74.125.138.100): icmp_seq=10 ttl=44 time=14.7 ms
64 bytes from dn-in-f100.1e100.net (74.125.138.100): icmp_seq=11 ttl=44 time=14.7 ms
^C
--- google.com ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10612ms
rtt min/avg/max/mdev = 14.329/14.527/14.726/0.181 ms
[                        ]#
```

- time – what time it took to a network package to reach the destination

Check a path of a package

- Start -> run -> tracert [URI or IP-address]



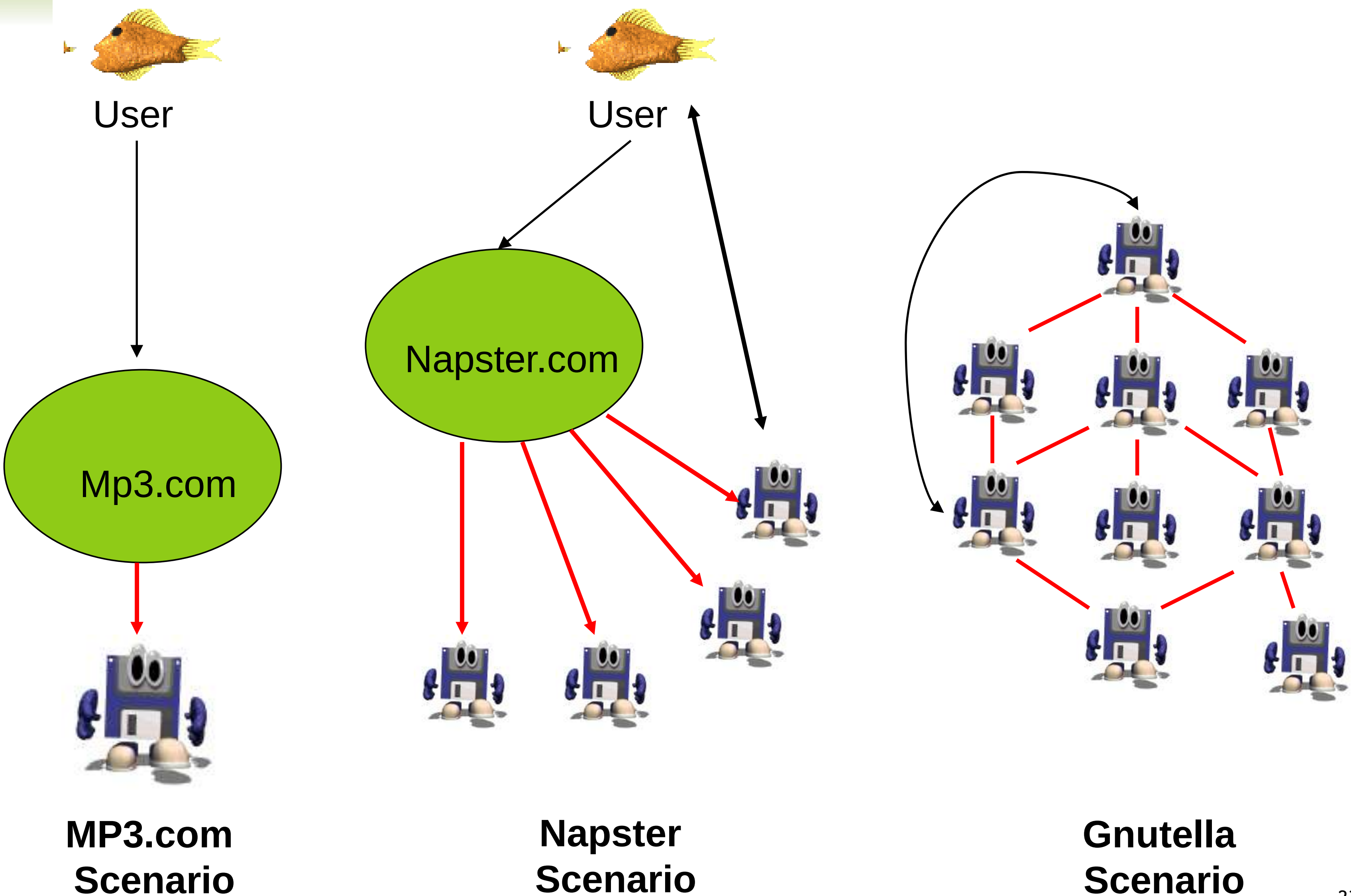
Taxonomy for Distributed Systems

Taxonomy is based on following factors and their relation to centralization:

1. Resource Discovery: Mechanism for discovering resources on a distributed system?
    - Examples: DNS, JXTA Rendezvous, Jini LUS, UDDI etc
2. Resource Availability: Scalability – do resources scale with network?
   - does access to them scale with network?

   See example...

MP3.COM, NAPSTER AND GNUTELLA



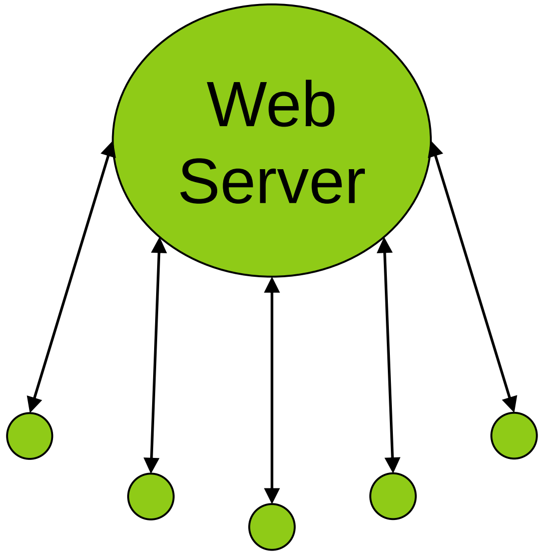
Ian J. Taylor and Andrew Harrison From P2P and Grids to Services on the Web: Evolving Distributed Communities

Taxonomy for Distributed Systems

Taxonomy is based on following factors and their relation to centralization:

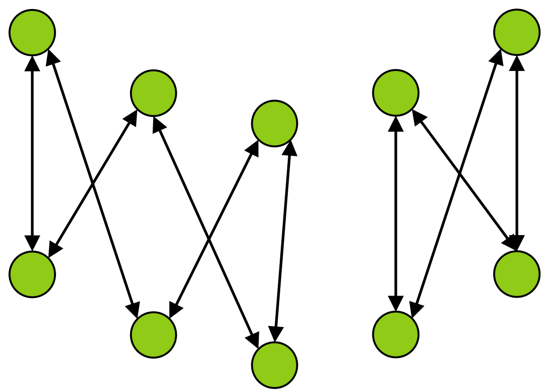
1. Resource Discovery: Mechanism for discovering resources on a distributed system?
   - Examples: DNS, JXTA Rendezvous, Jini LUS, UDDI etc
2. Resource Availability: Scalability – do resources scale with network?
   - does access to them scale with network?
3. Resource Communication: Two types:
   - Brokered Communication (centralized): communication is passed through a central server - resources do not have direct references to each other.
   - Point to point (decentralized -peer to peer): a direct connection between the sender and the receiver.

Centralization of Point-to-Point Connections



Many to one relationship between users and the web server and therefore this can be considered centralized communication

True Peer to Peer e.g. Gnutella



Equal Peers, communication is supposed to be even i.e. each provider is also a consumer of information and each node has an equal number of connections

This is not always the case … as we will learn in lecture 4

Taxonomy for Distributed Systems

Taxonomy is based on following factors:

1. Resource Discovery: Mechanism for discovering resources on a distributed system?
   - Examples: DNS, JXTA Rendezvous, Jini LUS, UDDI etc
2. Resource Availability: Scalability – do resources scale with network?
3. Resource Communication: Two types:
   - Brokered Communication (centralized): communication is passed through a central server - resources do not have direct references to each other.
   - Point to point (decentralized -peer to peer): a direct connection (although connection maybe multi-hop) between the sender and the receiver.

Centralized systems - typically, client/server based systems

Hybrid – combinations of the 2 extremes e.g. brokered architecture

Decentralized systems - Peer to Peer (P2P)

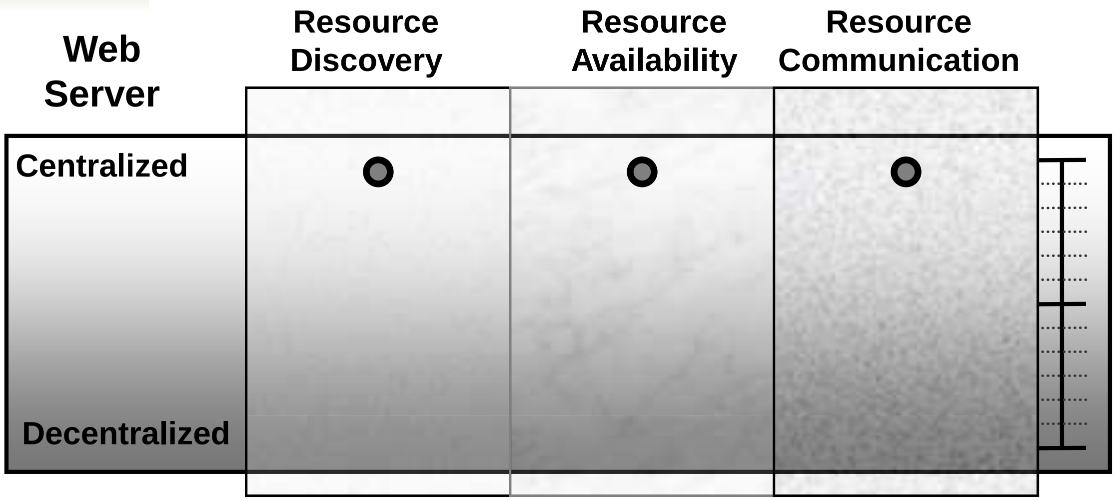
Centralized

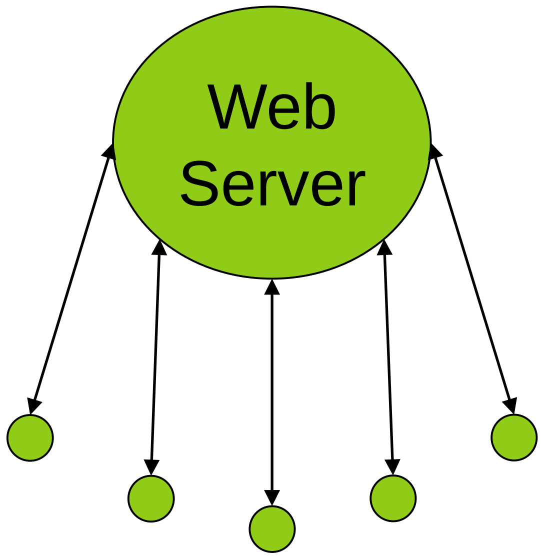
Hybrid

Decentralized

Ian J. Taylor and Andrew Harrison From P2P and Grids to Services on the Web: Evolving Distributed Communities

A Web Server: Centralized

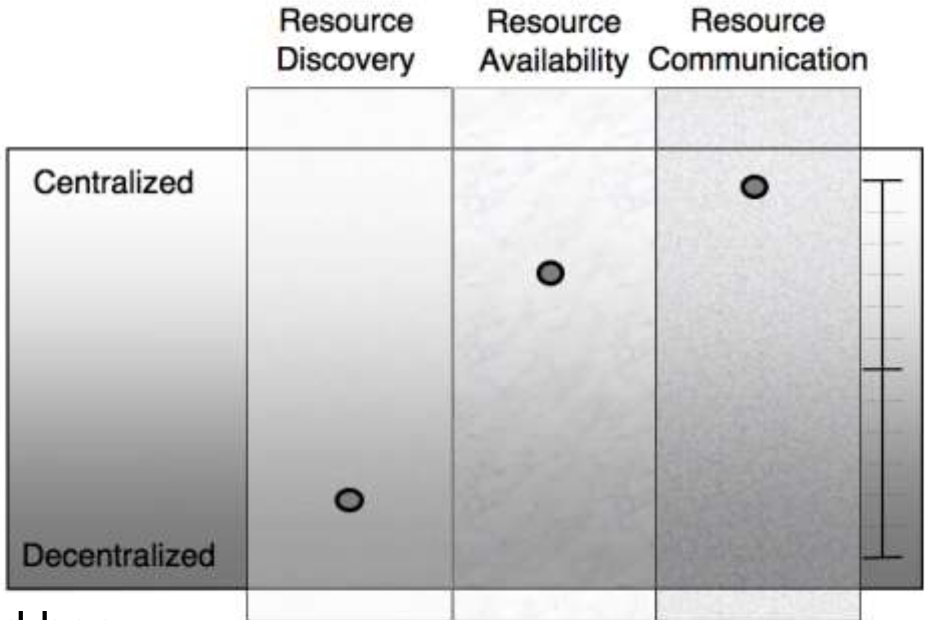




- Clients (i.e. users) use their web browser to navigate web pages on one or more web sites.
- Web site is static to particular domain

• Discovery: Centralized, DNS
• Availability: available or not
• Communication: centralized to the particular web server

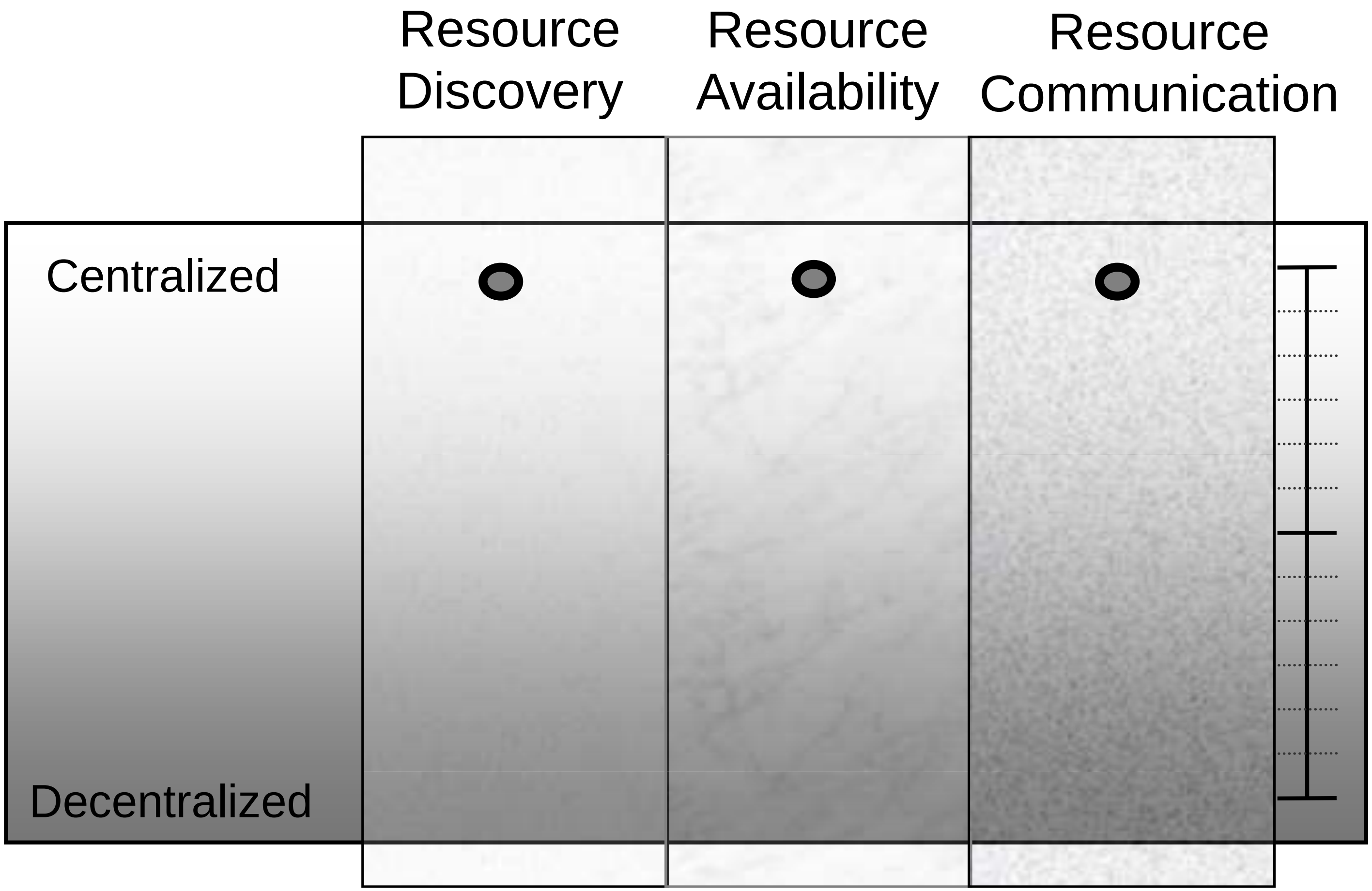
The Web as a whole



- **Discovery** - ad hoc
  - Often highly centralized, e.g. Google, but is also highly decentralized - the Web of links, e.g. the blogosphere and out of bounds
- **Availability** - depends on the granularity of the request
  - There is a level of replication on the Web and caching can be used to duplicate availability
- **Communication** - centralized
  - Communication happens via a centralized entity, e.g. Facebook, MySpace, Flickr, blogs, etc

Ian J. Taylor and Andrew Harrison From P2P and Grids to Services on the Web: Evolving Distributed Communities

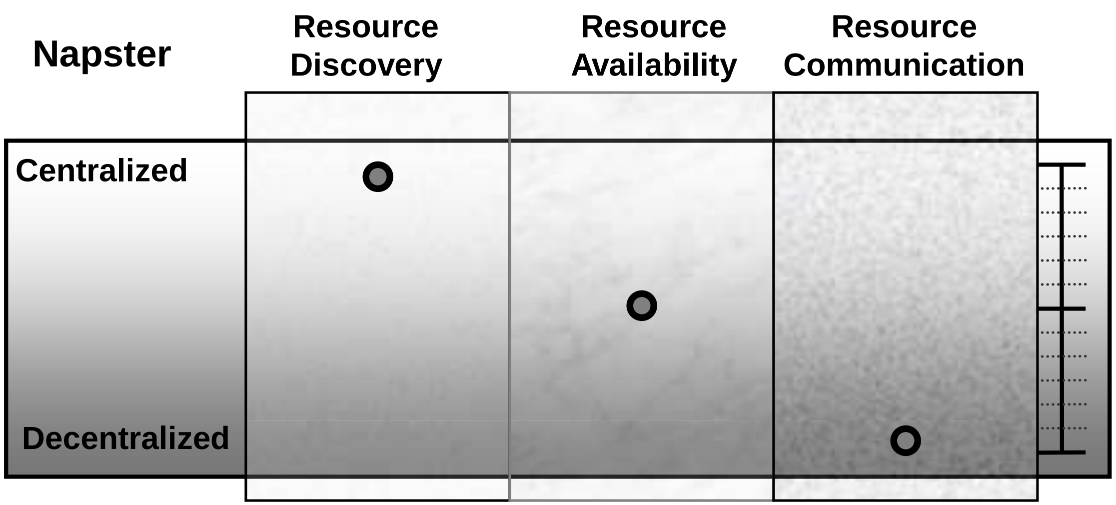
SETI@HOME

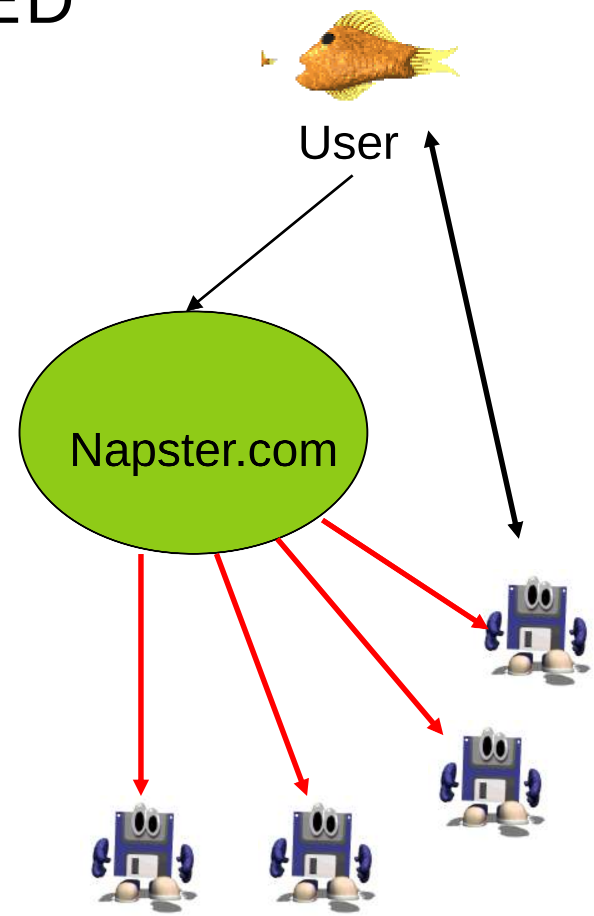


Search for Extraterrestrial Intelligence@home – volunteer computing system

generalized to BOINC API

Napster: Brokered

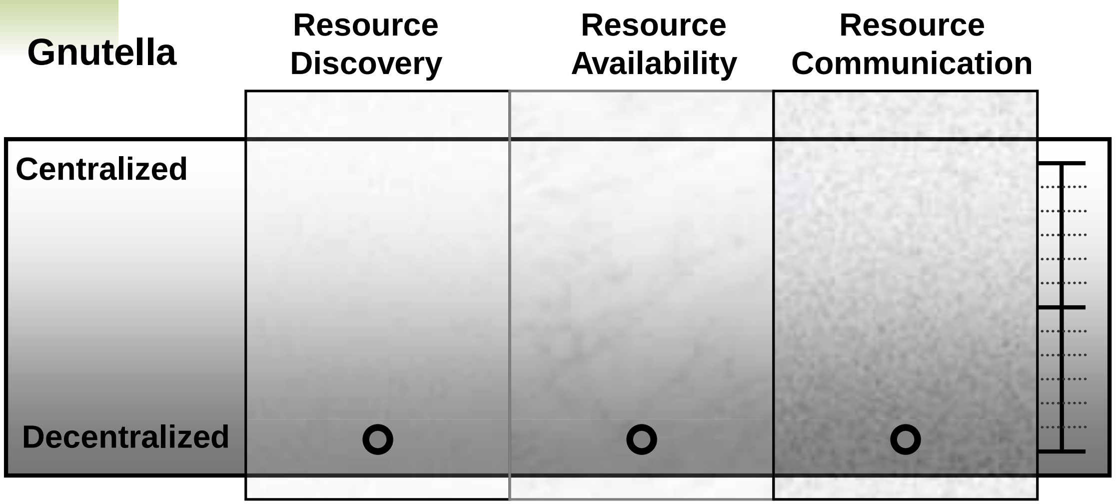


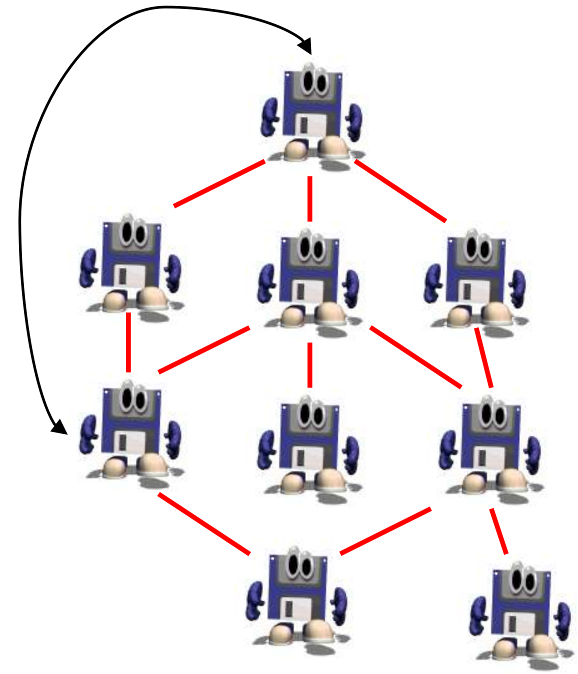


Clients search through Napster web site (well, they used to….)

- Discovery: Centralized through web site
- Availability: Once discovered via web site, availability is decentralized.
- Communication: decentralized between peers (MP3 sharers)

GNUTELLA: DECENTRALIZED

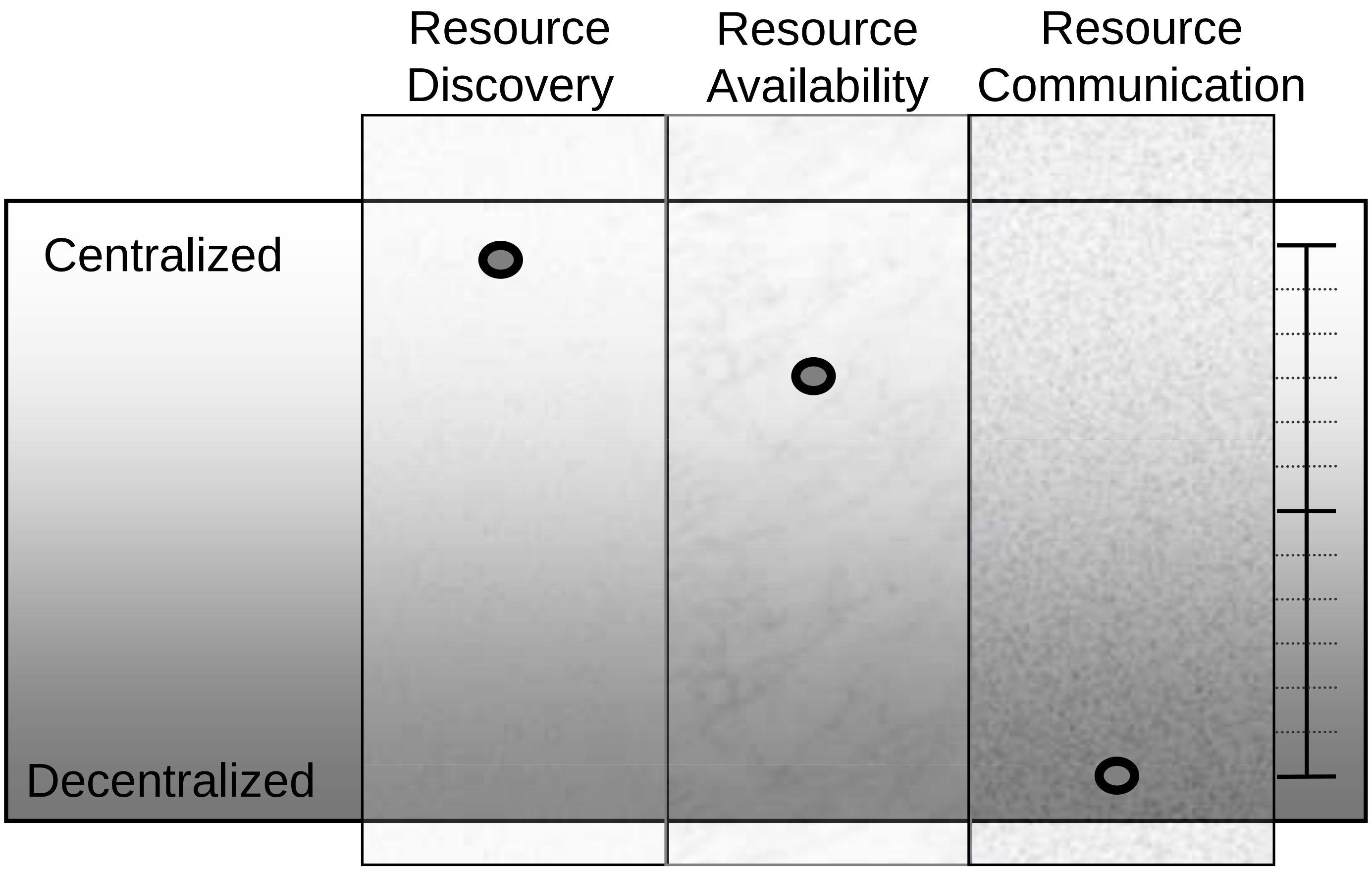




- Discovery: Decentralized through Gnutella messages (ping/pong mechanisms)
- Availability: Often an alternate path to resource
- Communication: point to point: decentralized between peers

Ian J. Taylor and Andrew Harrison From P2P and Grids to Services on the Web: Evolving Distributed Communities

Web Services



- Discovery: Centralized through registry
- Availability: Once discovered via registry, availability is decentralized.
- Communication: decentralized between provider and consumer

Note: This is for the current Web Services, technology stack
- in principal you can host web services in a number of ways

Concluding Remarks

1. Taxonomy
   a) Criteria
      a) Resource Discovery
      b) Resource Availability
      c) Resource Communication
   b) Taxonomy
      a) Centralized
      b) Hybrid
      c) Decentralized
2. Relevance
   a) Course relates distributed systems to this taxonomy