Network Middleware Systems

Prof.dr.sc Siniša Srbljić

Dr.sc. Ivan Benc

Dr.sc. Daniel Skrobo

School of Electrical Engineering and Computing
Consumer Computing Laboratory

Lecture 3

Scalability

Prof.dr.sc. Siniša Srbljić

School of Electrical Engineering and Computing
Consumer Computing Laboratory

Lecture Outline

3 Scalability

3.1 Introduction

3.2 Large Scalability

3.3 Worldwide Scalability

3.1 Introduction

- **Scalability**
  - Toward to larger and larger system

- **Dimensions of scalability**
  - System size
    - Number of machines
  - Application size
    - Machine load and communication traffic

3.1 Introduction

- **Dimensions of scalability**
  - Geographic distribution
  - Information, computing, and communication technology
    - Hardware, software, protocols, languages, methods, …
  - Security and privacy
  - Social and legal dimension
  - Manageability

3.1 Introduction

- **Potential bottlenecks and problems**
  - Computational and communication complexity of application algorithm
    - Centralized algorithms
  - Network infrastructure
  - Architecture
    - Client-server architecture

3.1 Introduction

- **Potential bottlenecks and problems**
  - Communication, collaboration, and synchronization
    - Traffic and node load
    - Synchronous communication
    - Pushing information, server initiated communication
  - Data storage and management
    - Centralized data storage, linear lists, single file tree
  - Strict semantics, consistency, and coherence
  - Stateless and stateful solutions

3.1 Introduction

- **How to deal with scalability problem**
  - Decentralized algorithms
    - System partition into the smaller independent units
    - No machine has complete information about the system
    - Machines make decisions based only on local information
    - Failures of one machine does not ruin the algorithm
    - There is no implicit assumptions that a global clock exists
  - Data storage and management
    - Data placement, migration, replication and caching

3.1 Introduction

- **How to deal with scalability problem**
  - Communication, collaboration, and synchronization
    - Asynchronous communication
    - Pulling information, client initiated communication
  - Weaker guarantees for semantics, consistency, and coherence
  - Limited stateful solutions

3.1 Introduction

- **How to deal with scalability problem**
  - Run-time monitoring and adaptability
  - Distribution and automation of management and configuration
  - Network infrastructure improvement
    - Multiple servers, POPs, communication links, bandwidth
  - MIT professor Mildred Dresselhaus
    - A changing industry focus from software to hardware--specifically nanolevel electronics--due to hardware's approaching scalability threshold

3.1 Introduction

- **Classes of scalability**
  - Large scalability
  - Worldwide scalability

Lecture Outline

3.2 Large Scalability

- 3.2.1 Introduction to large-scale architecture
- 3.2.2 Intranet systems
- 3.2.3 Design principles
- 3.2.4 Design example
  - 3.2.4.1 GeoPlex distributed cache manager
- 3.2.5 Performance comparison

3.2.1 Introduction to large-scale architecture

- **Limited growth**
  - Physical limiters
    - Signal propagation and power dissipation
    - Bus-based multicomputer systems
      - 25-100 nodes

3.2.1 Introduction to large-scale architecture

- **Limited growth**
  - Hardware architectures limiters
    - Cross-sectional bandwidth
    - Bus-based multiprocessors
      - Up to 32 processors
    - Multiprocessors based on hierarchy of rings
      - Up to 100 processors

3.2.1 Introduction to large-scale architecture

- **Limited growth**
  - Software architectures limiters
    - Communication and information management protocols
    - Multicomputer systems
      - Up to 250 nodes

3.2.2 Intranet systems

- **Clusters**
  - Homogenous systems
  - Ultra-high-performance, special-purpose interconnection networks
  - High degree of centralized control

3.2.2 Intranet systems

- **Cluster**
  - Computational model
    - Synchronous communication
    - Distributed shared memory
    - Message passing
  - Programming
    - Resource allocation and processes management

3.2.2 Intranet systems

- **Local-area networks**
  - Heterogeneous systems
  - High reliable communication based on broadcast
  - Geographical distribution
  - Separate administration
  - Lack of global knowledge
  - Limited centralized control

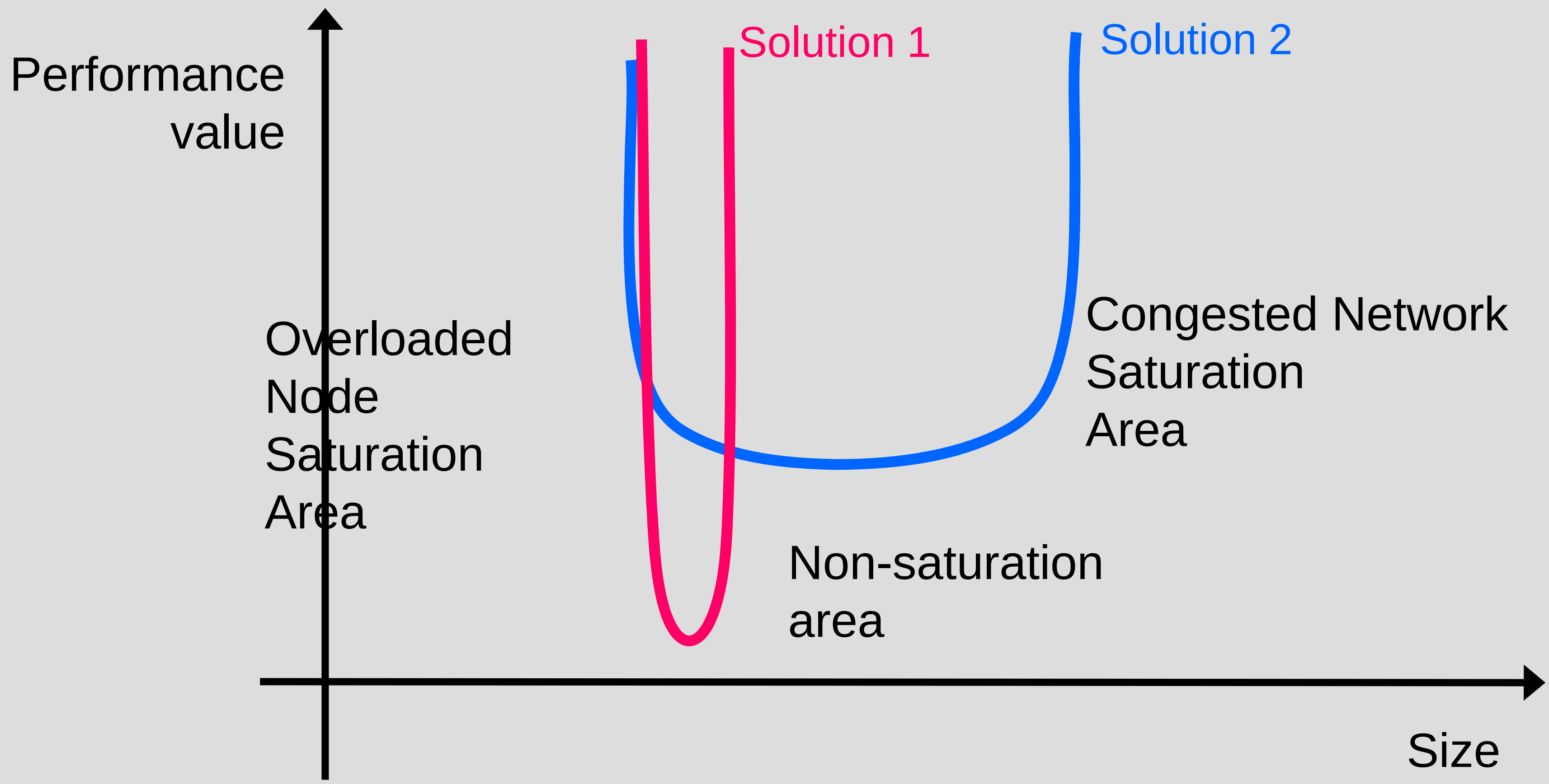
3.2.2 Intranet systems

- **Local-area networks**
  - Computational model
    - Loosely synchronous communication, RPC
    - CORBA, Java RMI, DCOM
    - Client/server
  - Programming
    - Connection to established services that encapsulate hardware resources or provide defined computational services

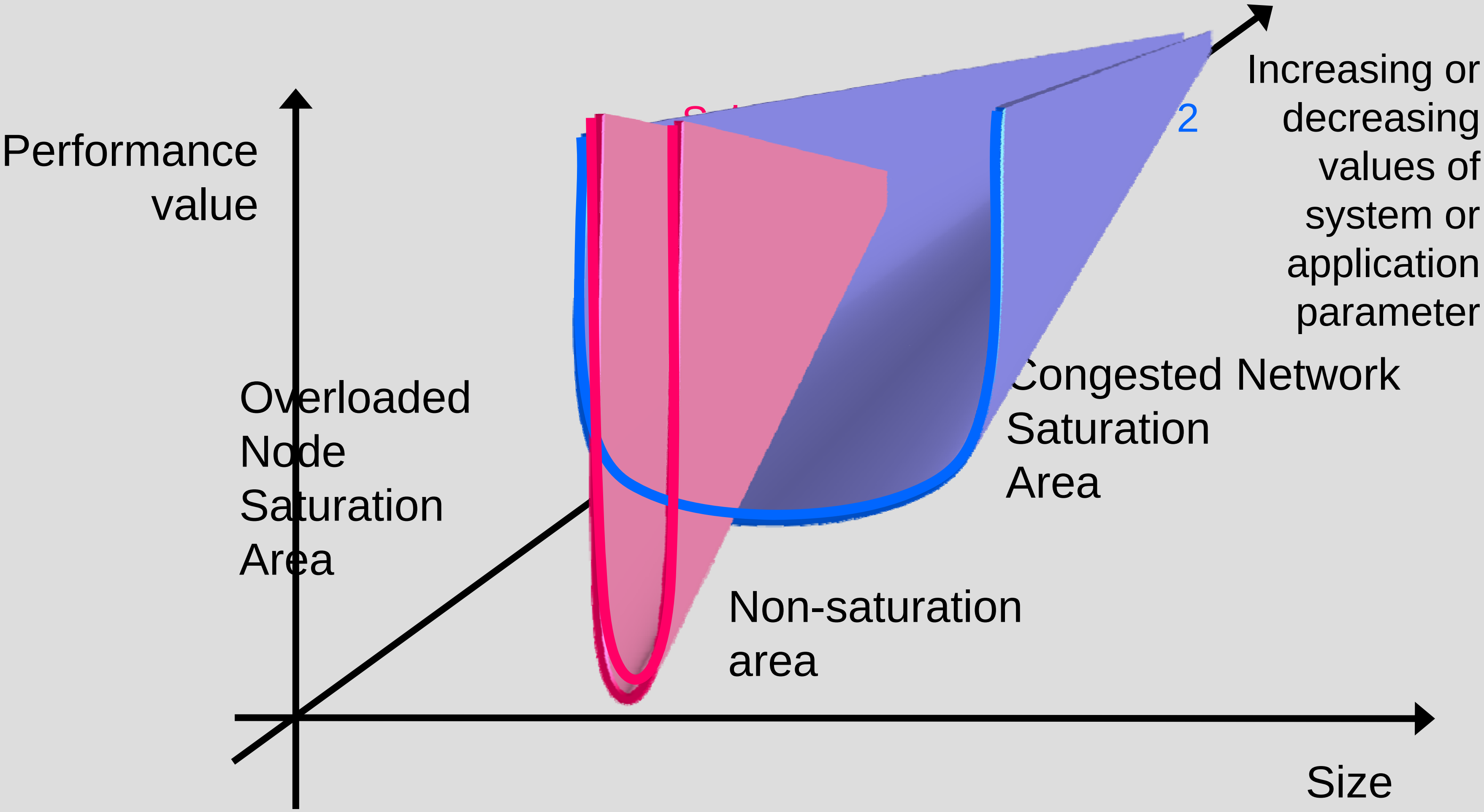
3.2.3 Design principles

- **Trade-offs of**
  - Performance
  - Security and privacy
  - Usability
  - Functionality
- **Performance**
  - Latency, traffic, and workload
  - Coherence and consistency
  - Energy dissipation

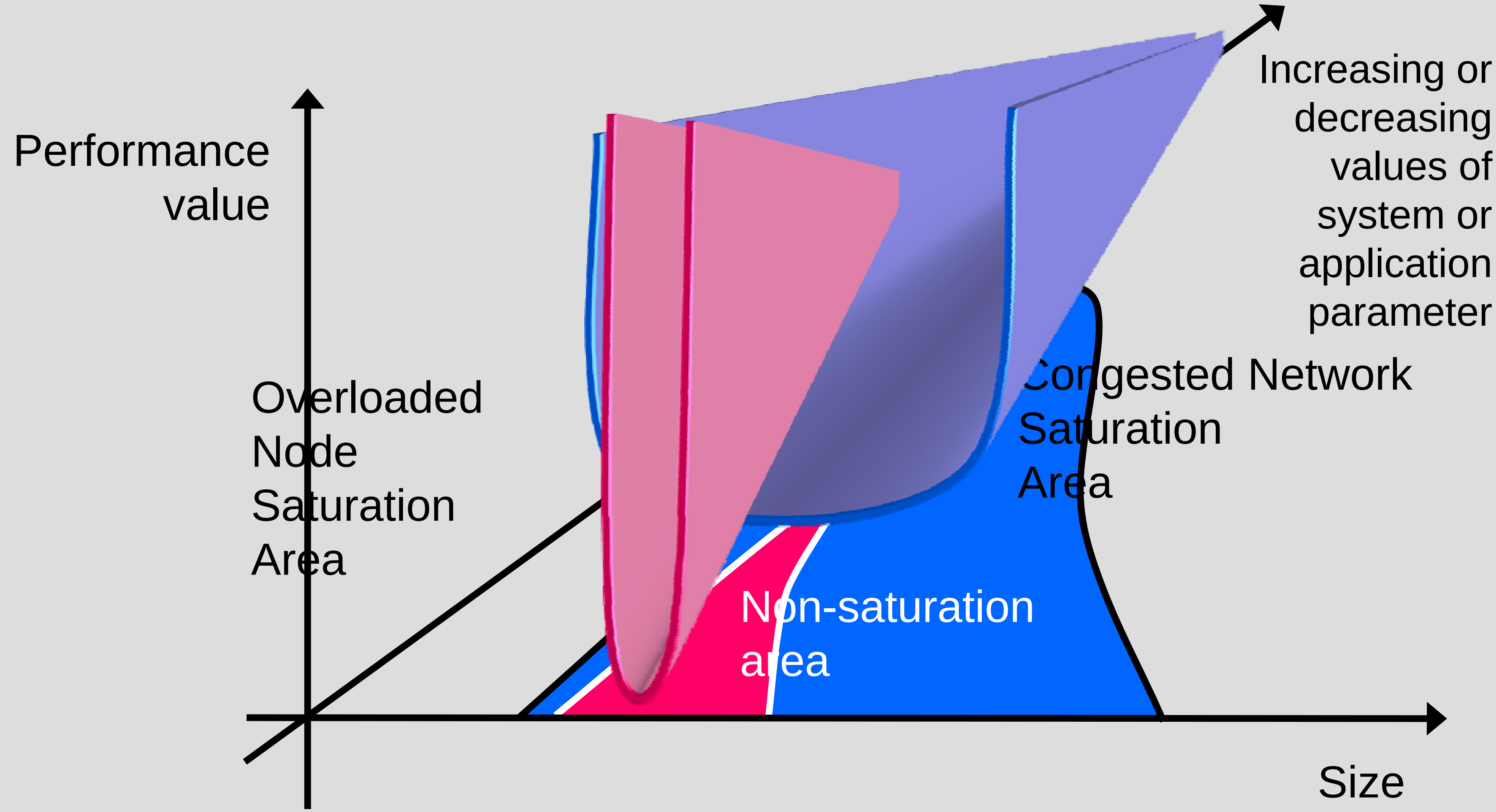
3.2.3 Design principles



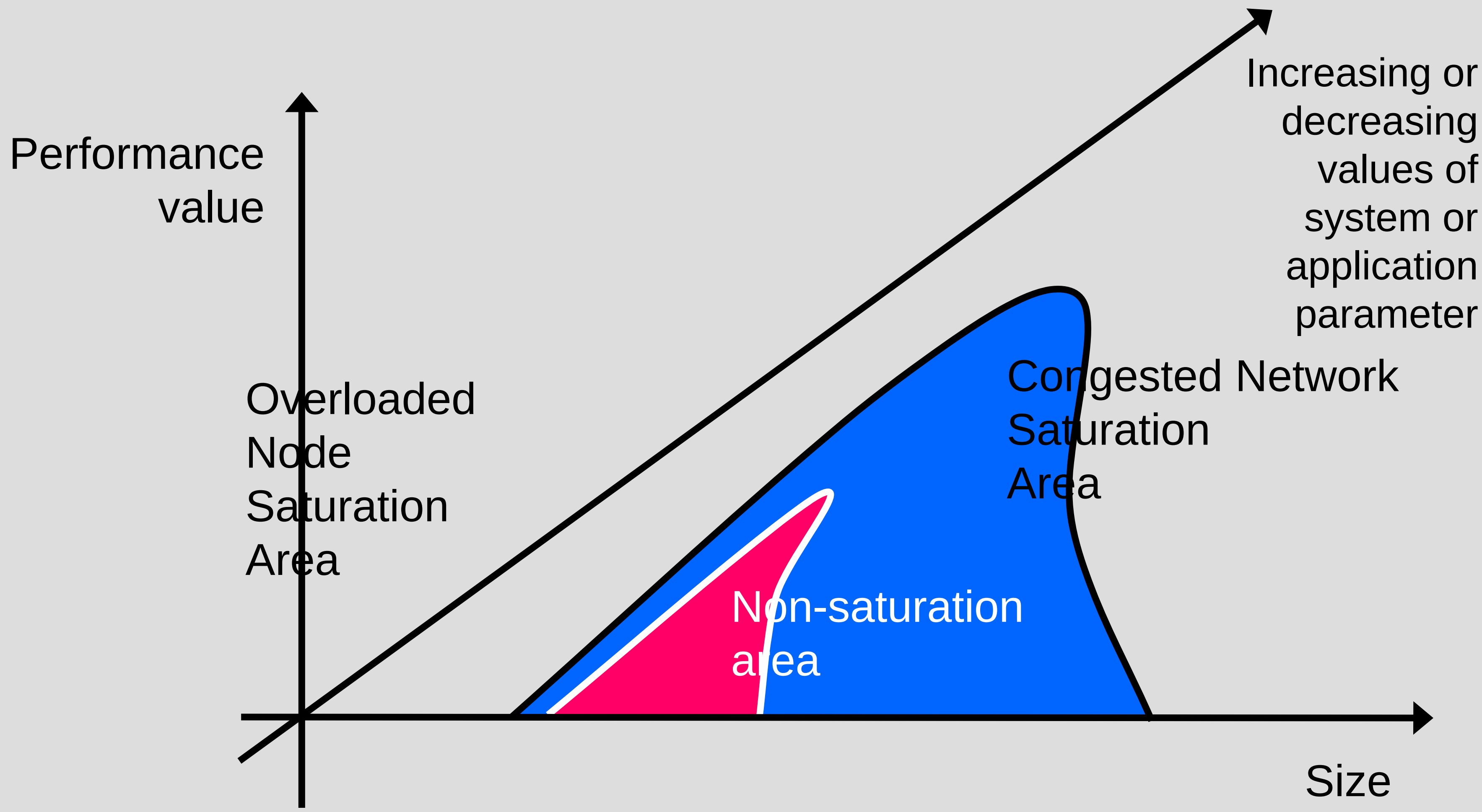
3.2.3 Design principles



3.2.3 Design principles

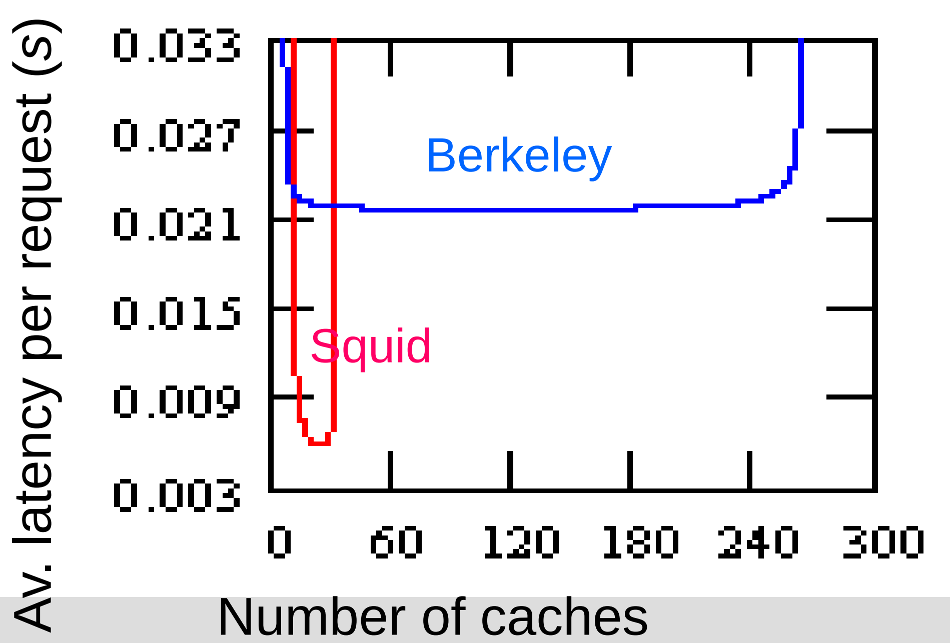


3.2.3 Design principles



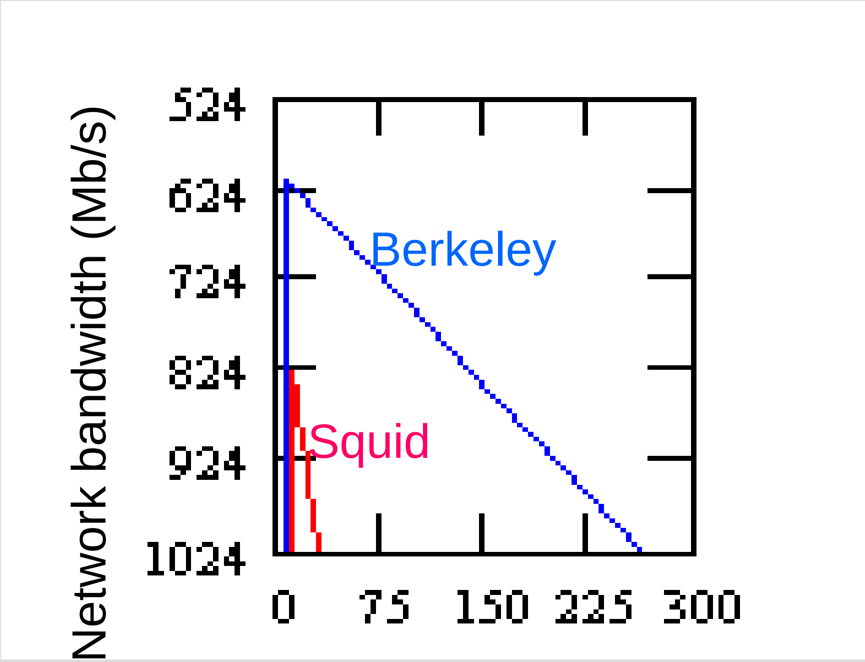
3.2.3 Design principles

Distributed Cache Managers

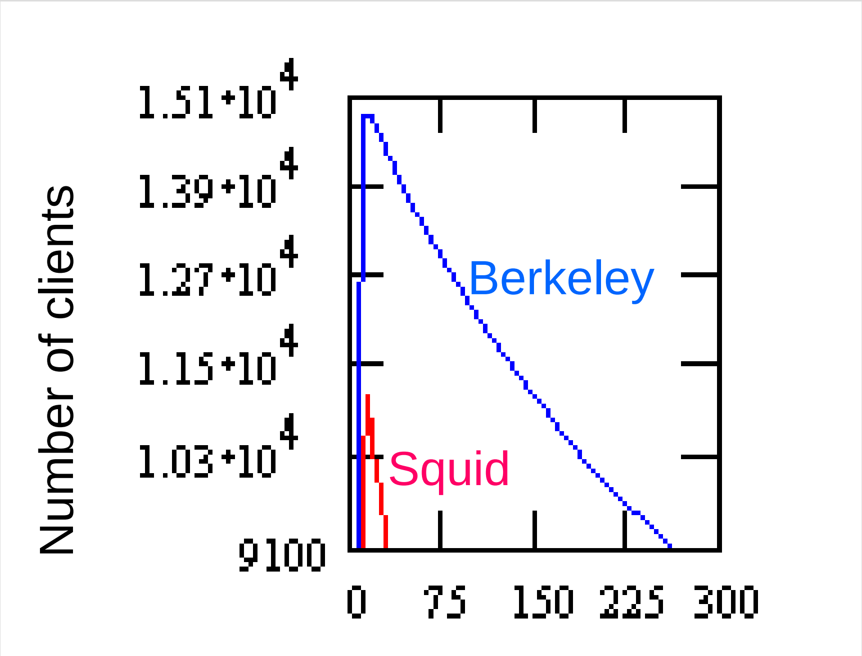


3.2.3 Design principles

Distributed Cache Managers



Number of caches



Number of caches

3.2.4 Design example

- **AT&T Labs, IP Technology Organization**
  - 1995 – 2000
  - Middletown, NJ, San Mateo, San Jose, CA

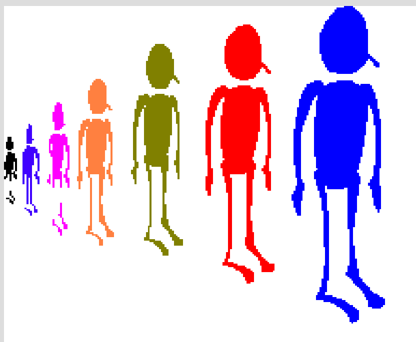
- **GeoPlex Platform**
  - The common open IP platform is a collection of reusable software components creating a framework for deploying secure, authenticated IP services over the open Internet or in internal intranets

3.2.4 Design example

- **GeoPlex Distributed Cache Manager**
  - Patents
    - Inventors: S. Srbljic, P.P. Dutta, T.B. London, D.F. Vrsalovic, and J.J. Chiang
    - Assignee: AT&T Corp (New York, NY, USA)
      - US5933849: Scalable distributed caching system and method
        - » Issued/Filed Dates: Aug. 3, 1999 / April 10, 1997
      - US6154811: Scalable network object caching
        - » Issued/Filed Dates: Nov. 28, 2000 / Dec. 4, 1998

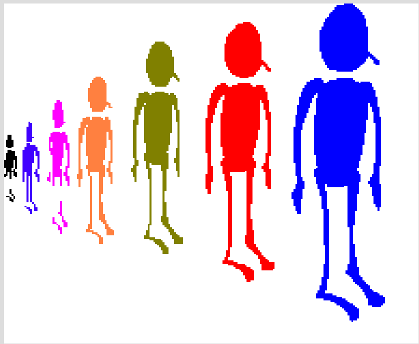
3.2.4.1 GeoPlex distributed cache manager

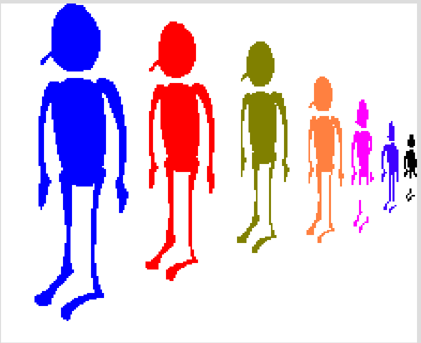
- **Motivation**
  - Ambiguity of definition of the coherence
  - Lack of coherence protocol
  - Performance



3.2.4.1 GeoPlex distributed cache manager

- **Motivation**
  - $T_{Avr} = r_{Hit} \times t_{Hit} + (1 - r_{Hit}) \times t_{Miss}$
  - To reduce the time parameters
  - To increase the hit rate
  - Harvest cache





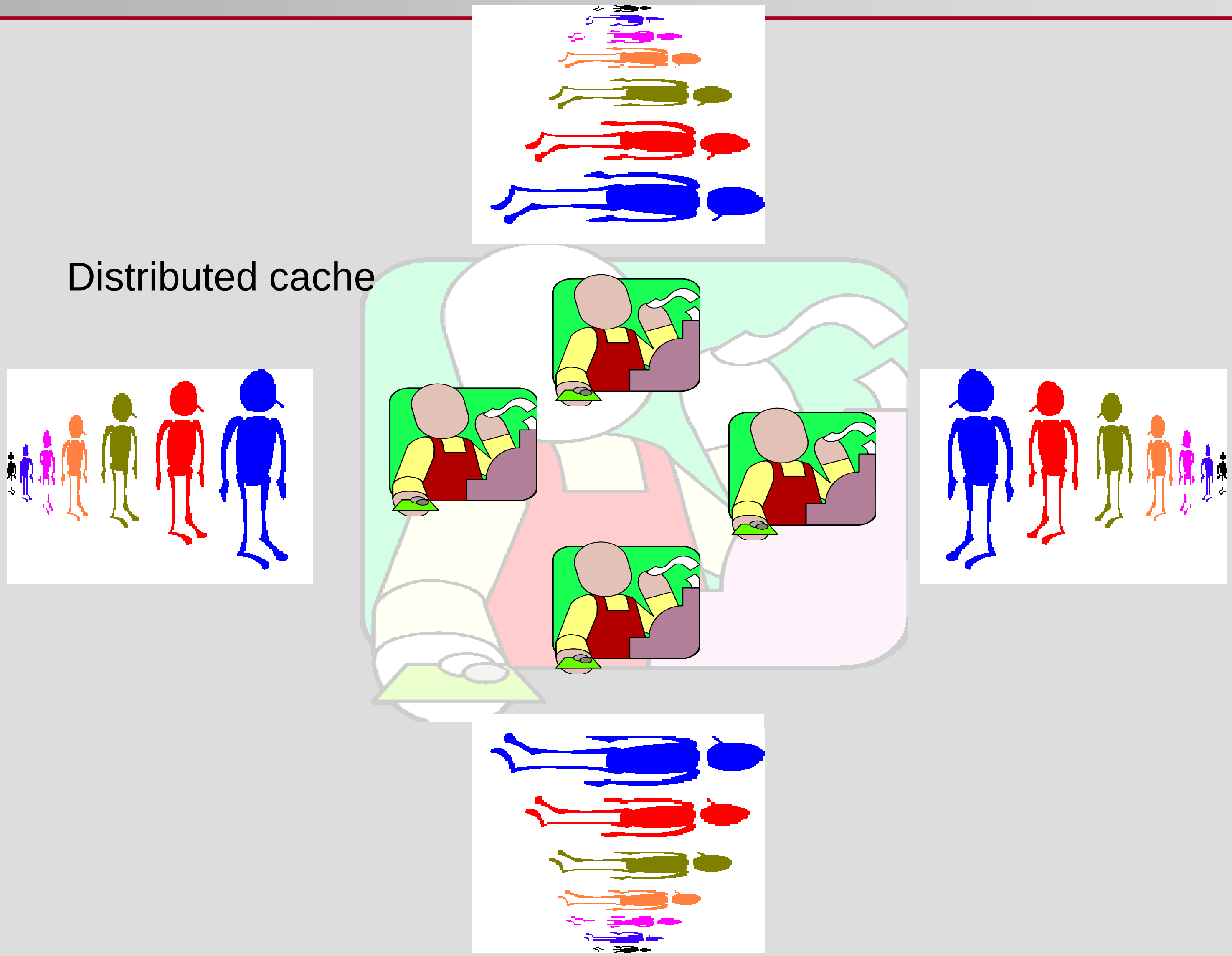
3.2.4.1 GeoPlex distributed cache manager

- **Experience**
  - To increase the hit rate
    - Connecting the larger number of clients to the same proxy machine increases the hit rate
    - Increasing the number of clients increases the probability that they are interested in the same data object
  - Proxy load and communication limited capabilities
    - Proxy machine that runs the cache limits the number of the clients connected to the same machine

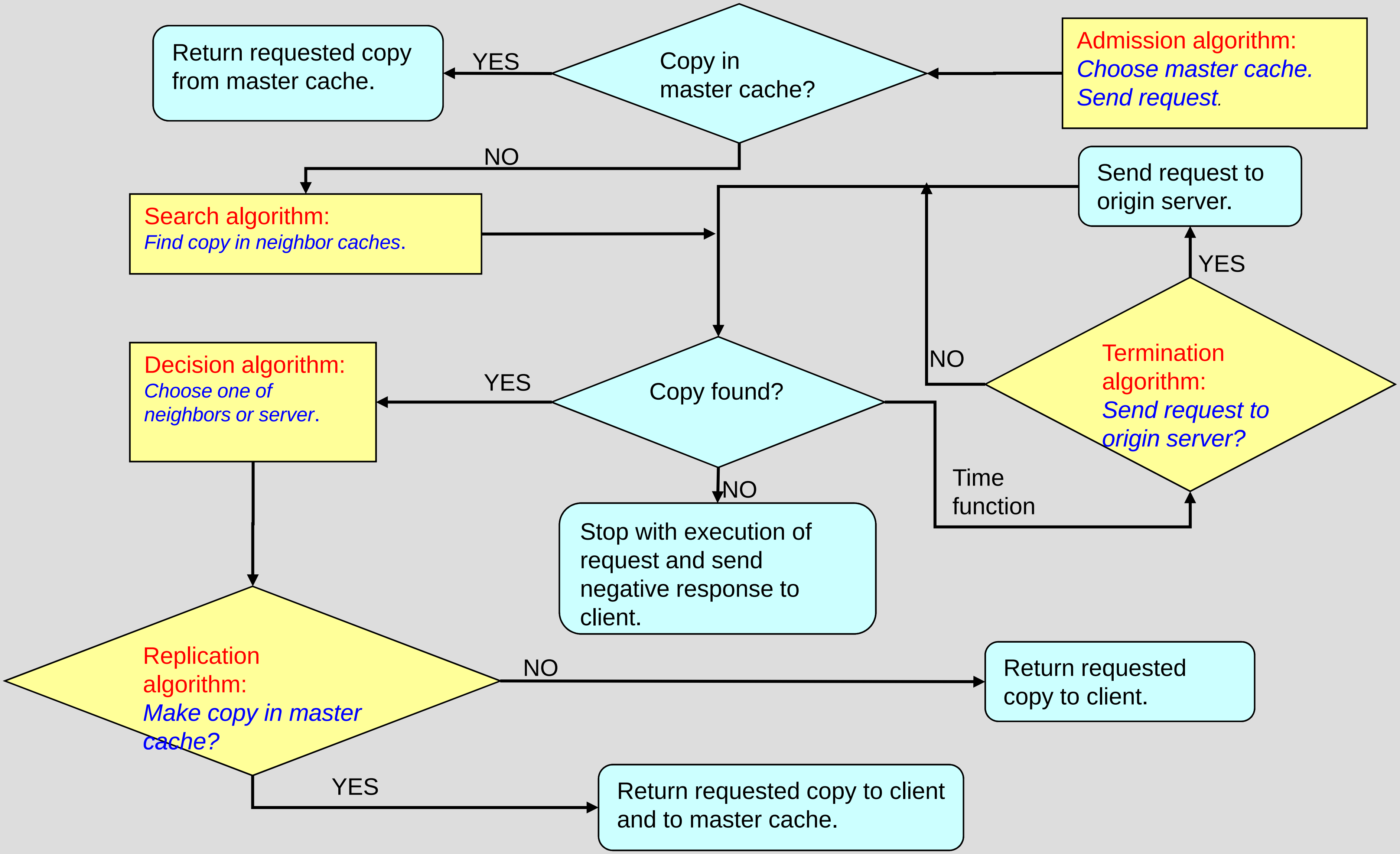
3.2.4.1 GeoPlex distributed cache manager

- **Solution**
  - Distributed cache
    - To enable the proxy machines to communicate
    - The communicating proxy machines act as a single distributed cache
  - Scalable distributed cache manager
    - In order to increase the hit-rate, the number of clients per one distributed cache should be increased
    - The main design issue is the scalability of distributed cache manager

3.2.4.1 GeoPlex distributed cache manager



3.2.4.1 GeoPlex distributed cache manager



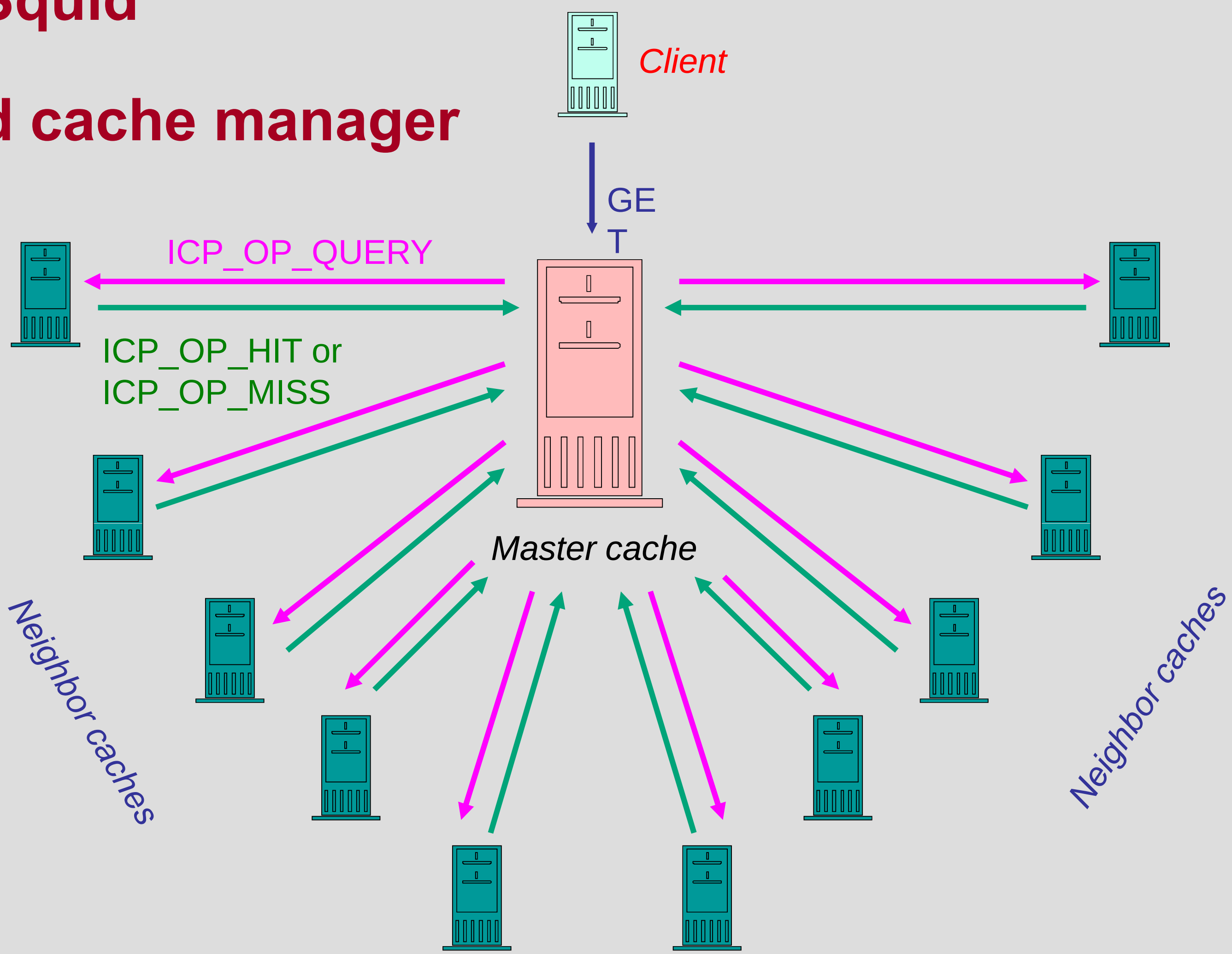
3.2.4.1 GeoPlex distributed cache manager

- **Harvest / Squid distributed cache manager**
  - Simple algorithm
  - Do not scale in the amount of the
    - Network traffic
    - Proxy load

3.2.4.1 GeoPlex distributed cache manager

Harvest / Squid

distributed cache manager

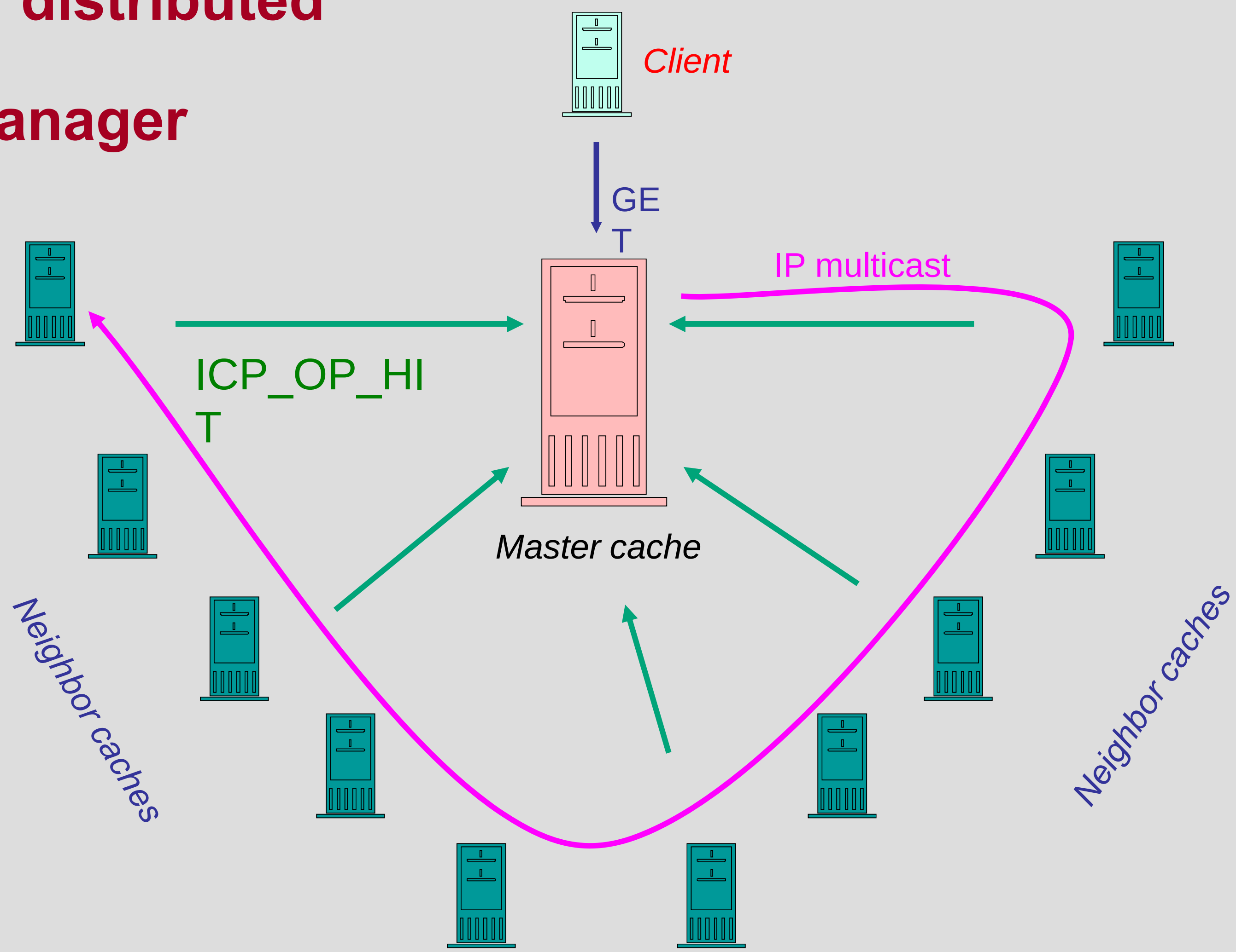


3.2.4.1 GeoPlex distributed cache manager

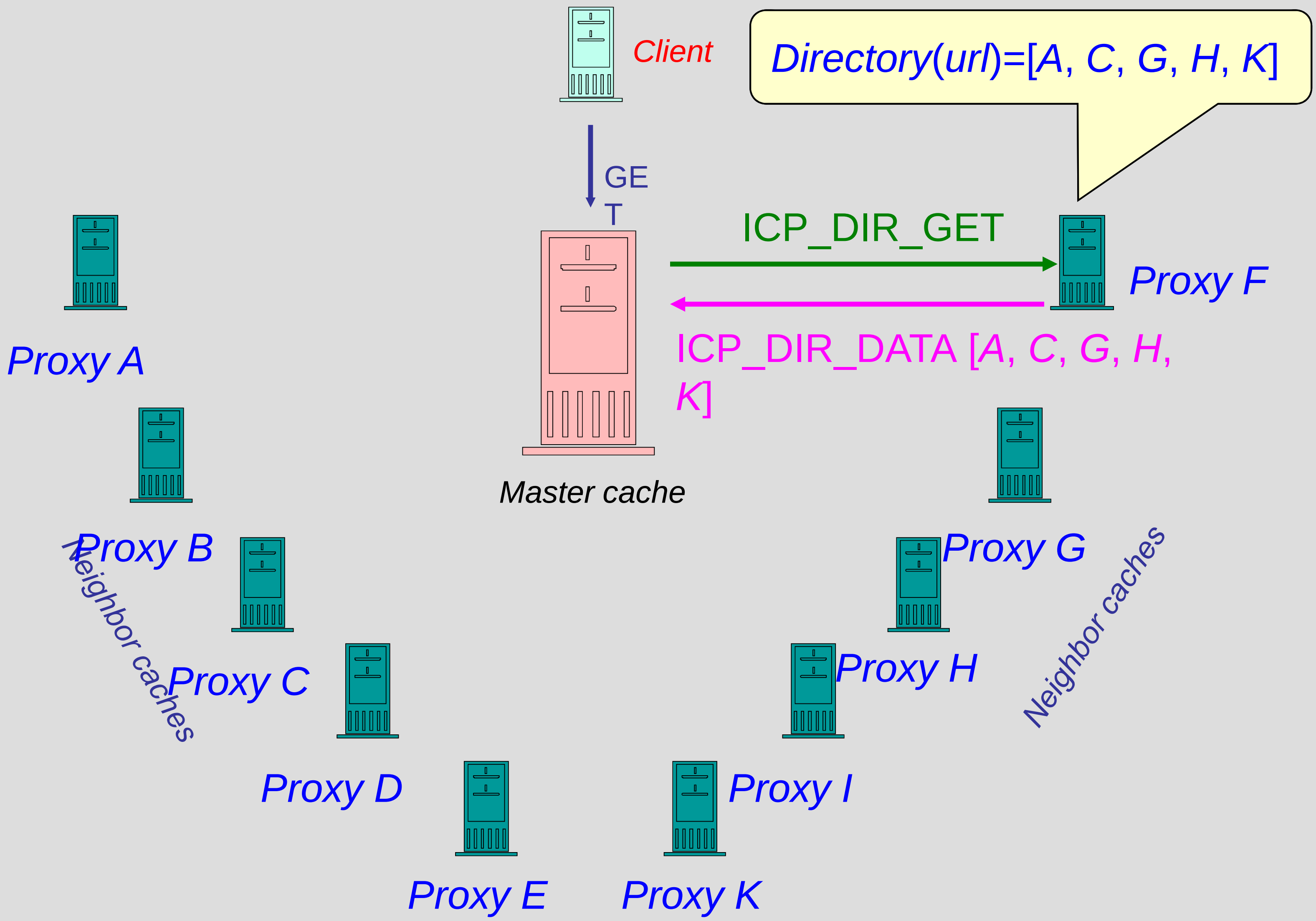
- **Berkeley distributed cache manager**
  - IP multicast instead of broadcast
    - Reduces the network traffic partially
  - Do not scale in the amount of the
    - Proxy load

3.2.4.1 GeoPlex distributed cache manager

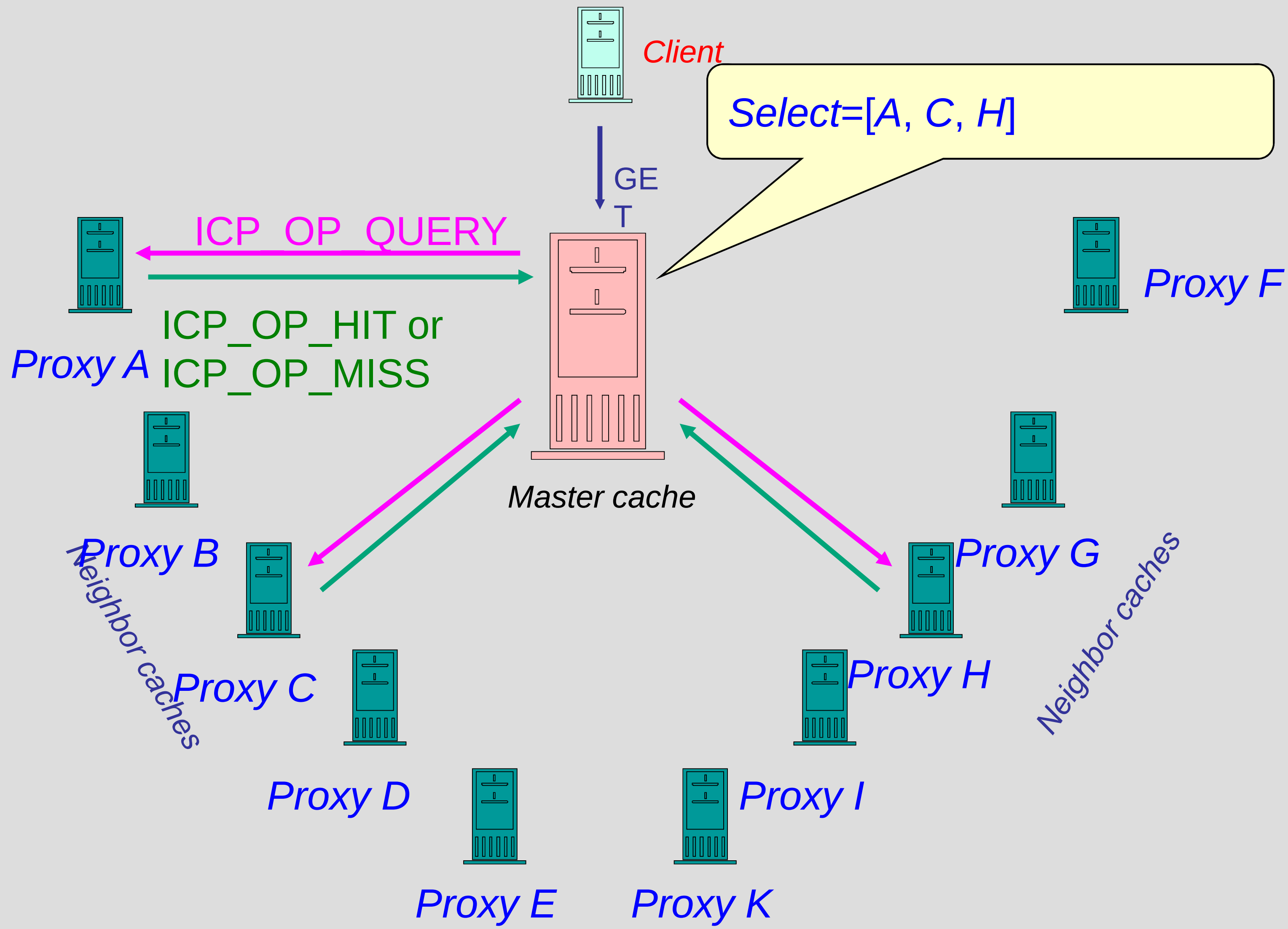
Berkeley distributed cache manager



3.2.4.1 GeoPlex distributed cache manager



3.2.4.1 GeoPlex distributed cache manager



3.2.4.1 GeoPlex distributed cache manager

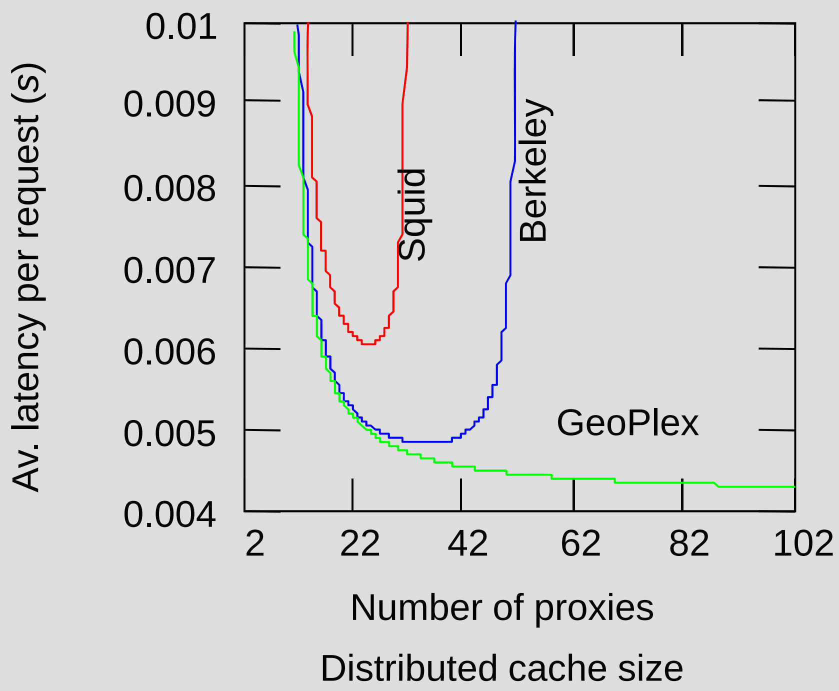
- **Directory reduces**
  - Network traffic
  - Proxy load

3.2.5 Performance comparison

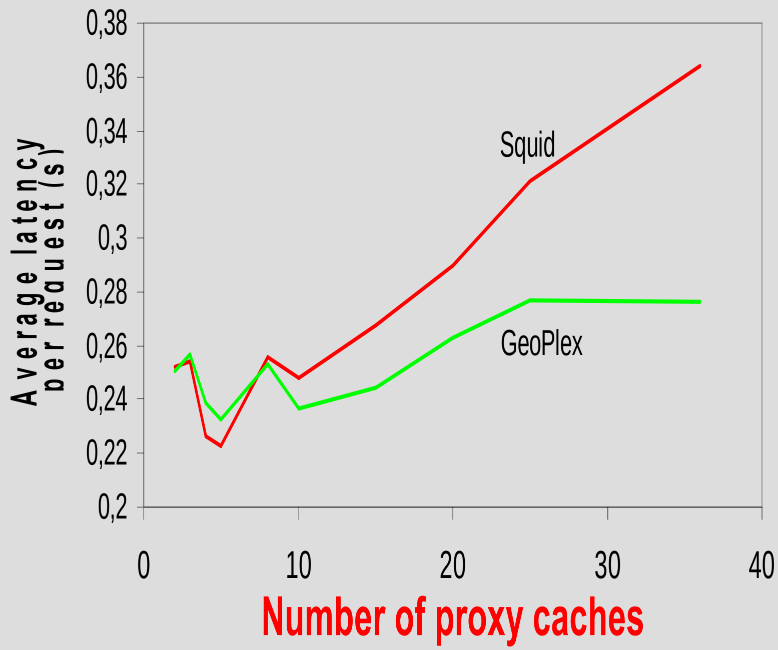
- **Siniša Srbljić**
  - Analytical performance comparison
- **Andro Milanović**
  - Performance measurement

3.2.5 Performance comparison

Analytical prediction

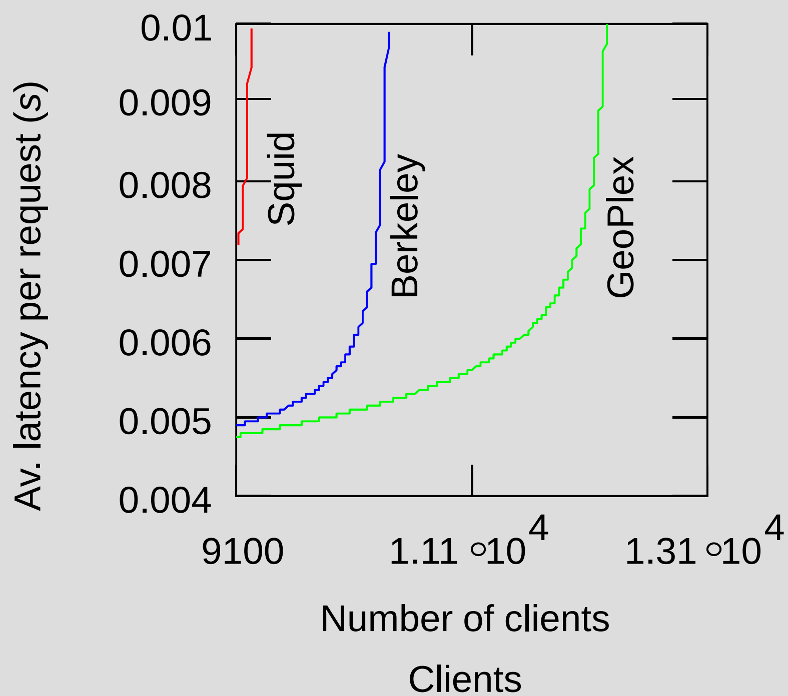


Measurment

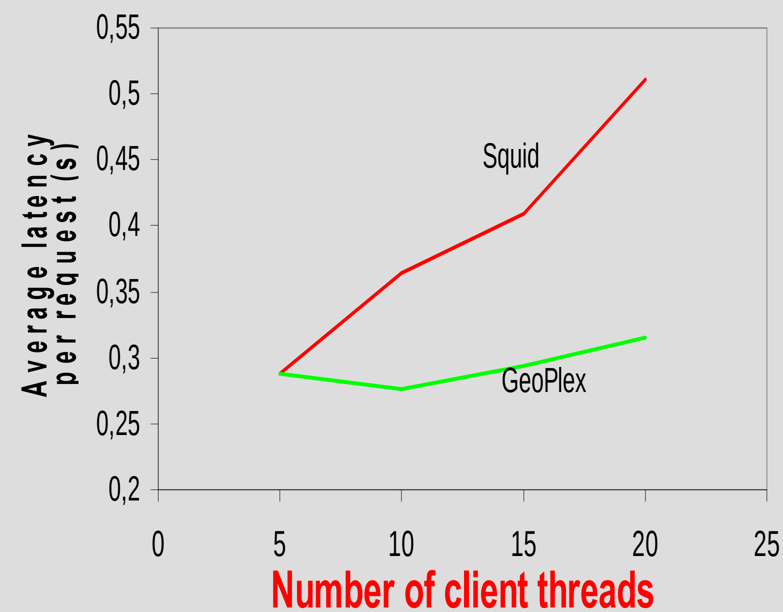


3.2.5 Performance comparison

Analytical prediction



Measurment

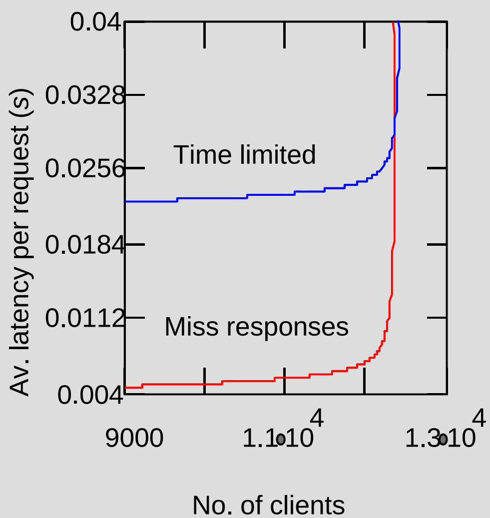


3.2.5 Performance comparison

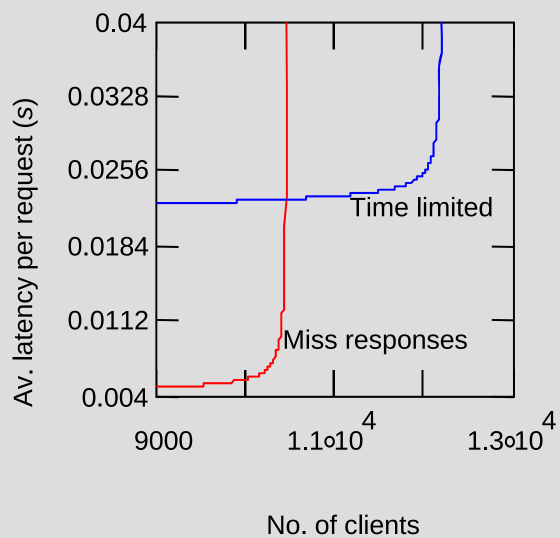
- **Termination Algorithm Selection**

Termination algorithm selection has impact only on performance

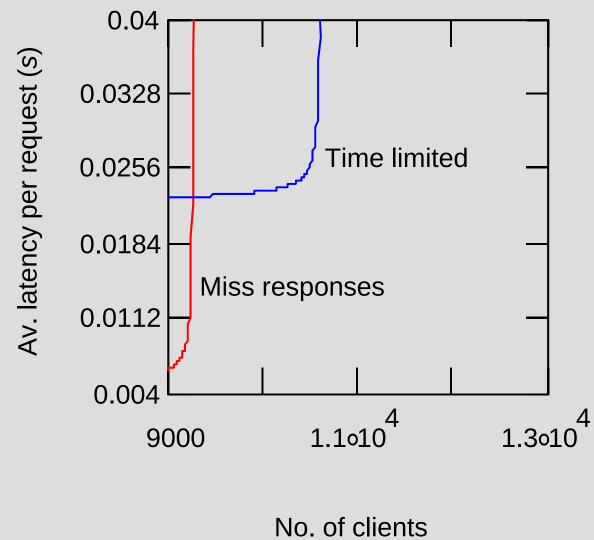
Termination algorithm selection has impact on both scalability and performance



GeoPlex protocol



Berkeley DCM protocol



Squid DCM protocol

3.2.5 Performance comparison

- **Based on deep knowledge of the system behavior**
  - Simple analytical model
    - Improves the performance of the system in the early phase of design
    - Guides the implementation of scalable system

3.2.5 Performance comparison

- **Scalable distributed cache manager**
  - Analytical performance prediction model
    - System design
    - System implementation
  - Performance measurement
    - Performance tuning

Lecture Outline

3.3 Worldwide Scalability

3.3.1 Introduction to worldwide-scale architecture

3.3.2 Internet systems

3.3.3 Design principles

3.3.4 Design examples

3.3.4.1 GeoPlex multiple clouds architecture

3.3.4.2 Domain Name System (DNS)

3.3.1 Introduction to worldwide-scale architecture

- **Unlimited growth**
  - Autonomous and independent domains
    - Separately managed and administrated
    - Distribution and hierarchy
  - Interaction and communication
    - Ad hoc and spontaneous
    - Message oriented, document based

3.3.1 Introduction to worldwide-scale architecture

- **Unlimited growth**
  - Computational model
    - Collaboration and competition
    - Data mining
  - Programming
    - Brokering, negotiation, and trading

3.3.2 Internet systems

- **Wide-area networks, internetworked systems**
  - Worldwide distribution
  - Unreliable, point-to-point communication
  - Lack of centralized control
  - International issues
  - Communication
    - Asynchronous communication
  - Program execution
    - Different mobile code models
    - Remote control

3.3.3 Design principles

- **Aggregation**
- **Lazy evaluation**
- **Replication based on caching**

3.3.3 Design principles

- **Aggregation**
  - Individual entities of a given type owned by one domain are aggregated and exported as a single unique entity
  - It reduces the amount of information about a given domain that is exported to other domains
  - Both time and space efficient and scalable

3.3.3 Design principles

- **Lazy evaluation**
  - Actions are only partially (“lazy”) evaluate by one domain
  - Partial evaluation uses as input parameters only the entities of the given domain
  - The results of the partial evaluation are sent to the another domain, where the rest of the evaluation is done by using entities from that domain
  - Only space efficient and scalable, but it could be time consuming
  - The execution of the action could be spread out through multiple domains, and time for execution could be long

3.3.3 Design principles

- **Replication based on caching**
  - In order to improve the performance of the lazy evaluated functions, some of the values of the entities are replicated from one to the another domain
  - Replication is done only by request (Lazy replication, or caching)
  - This means that value of the entity is replicated only if action that is executed need this value

3.3.3 Design principles

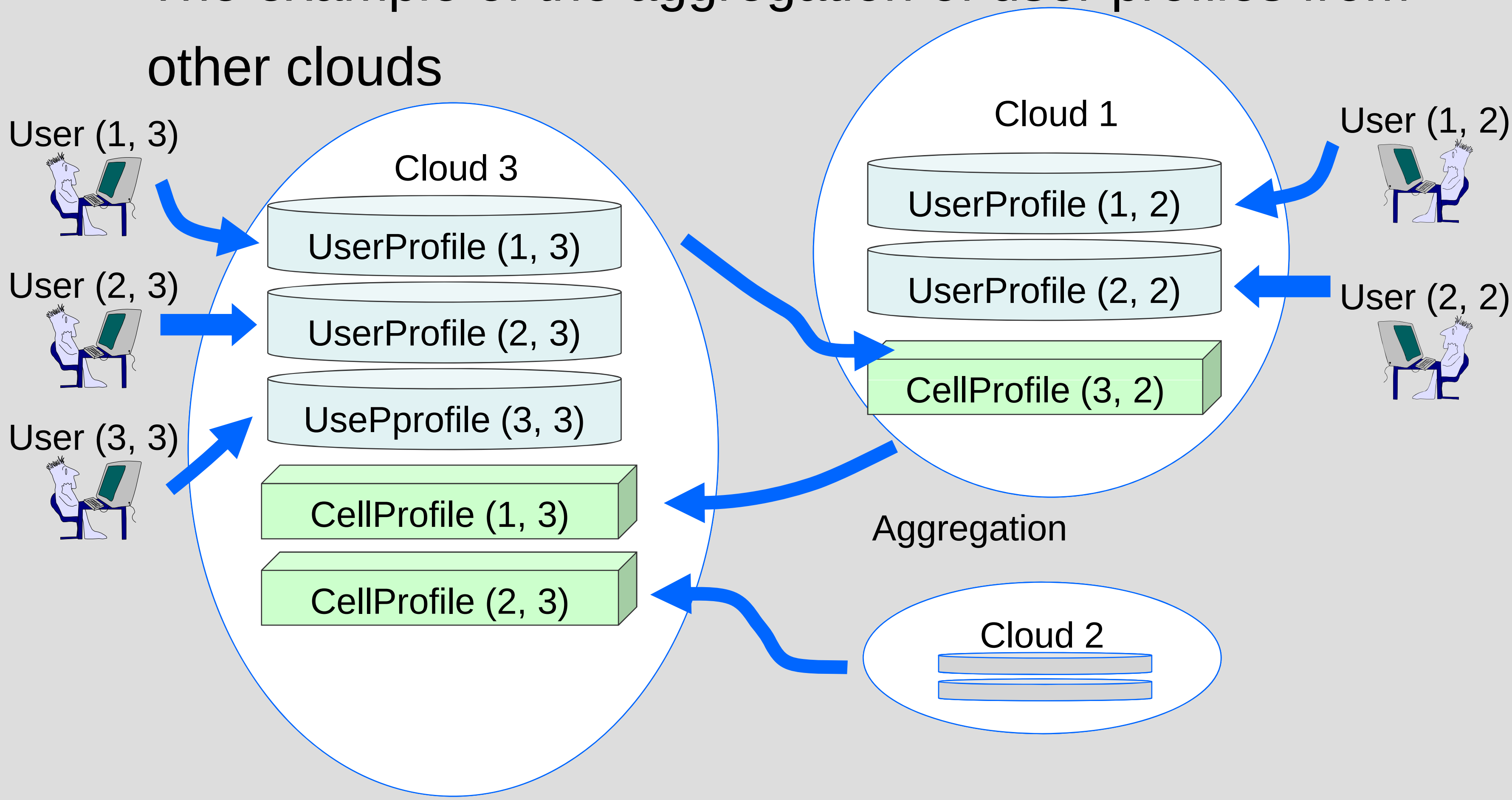
- **Replication based on caching**
  - Since the copy of the value must be coherent with the value of the entity in the originating domain, coherence protocol should be introduced
  - Protocol maintains copies of the same value coherent
  - The basic features of the coherence protocol are weak coherence and coherence window

3.3.4.1 GeoPlex multiple clouds architecture

- **Clouds**
  - Autonomous and independent domains
  - Constitutes an authentication trust and a single registration domain
  - Centralizes authentication, access control and security
  - Cloud registers, authenticates, and authorized users, services and other clouds

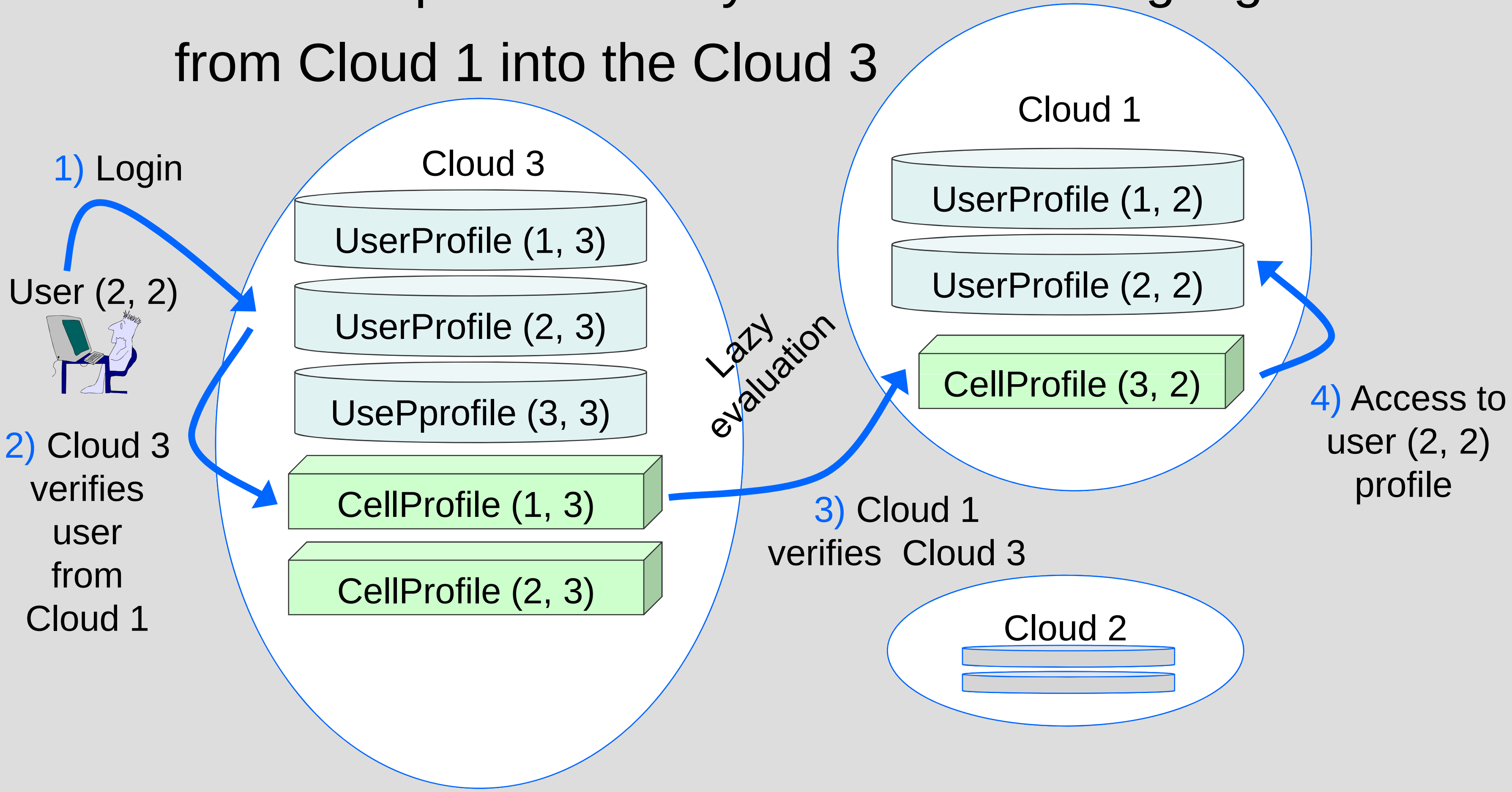
3.3.4.1 GeoPlex multiple clouds architecture

- **Aggregation**
  - The example of the aggregation of user profiles from other clouds



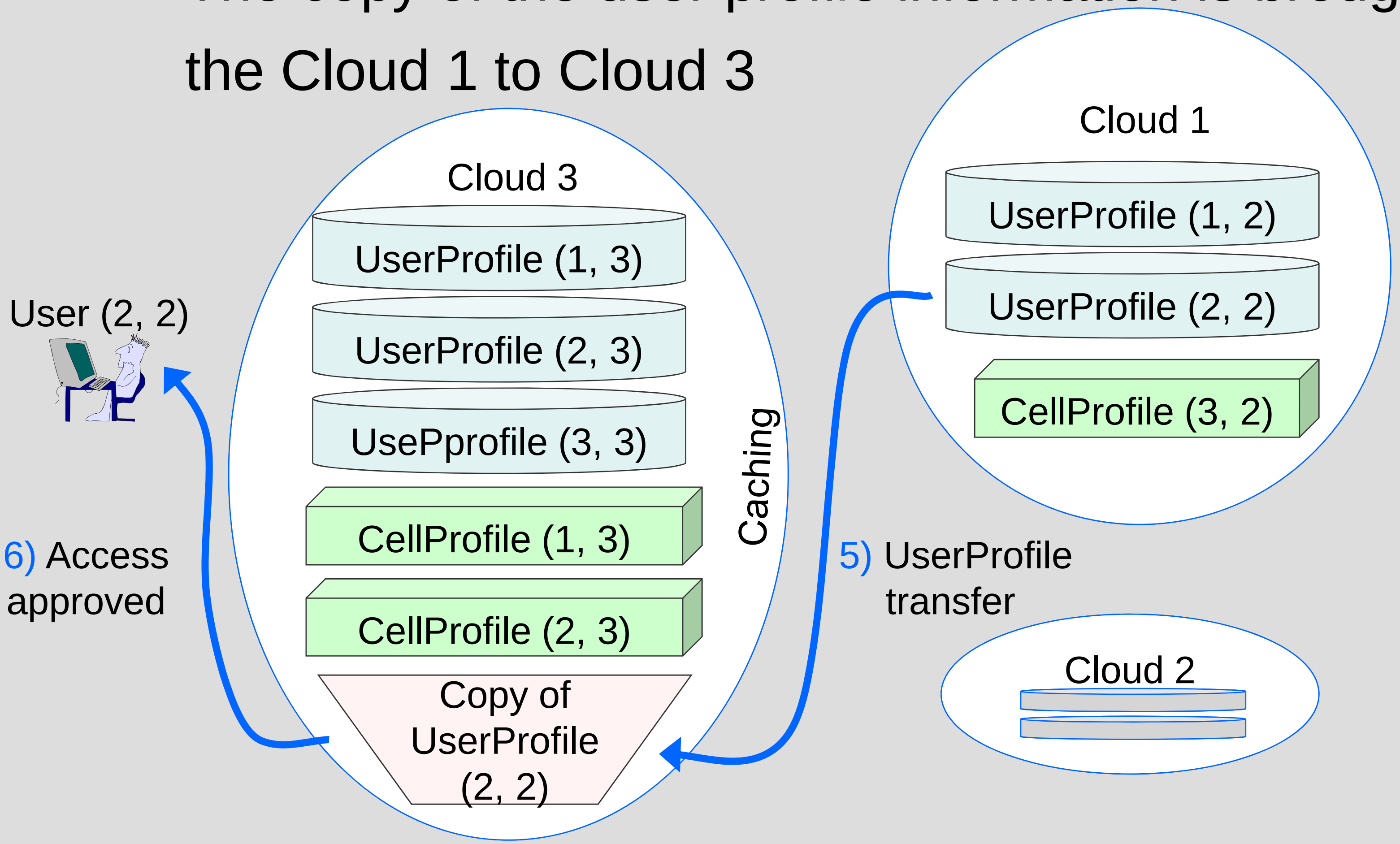
3.3.4.1 GeoPlex multiple clouds architecture

- **Lazy evaluation**
  - The example of the lazy evaluation during login of user from Cloud 1 into the Cloud 3



3.3.4.1 GeoPlex multiple clouds architecture

- **Caching**
  - The copy of the user profile information is brought from the Cloud 1 to Cloud 3



3.3.4.2 Domain Name System (DNS)

- **Directory service**
  - Keep track of locations of resources
  - Provide people-friendly names for resources
- **Cell organization**
  - Cell directory server (CDS)
    - Stores the names and properties of the cell's resources
  - Replicated and distributed database system
    - Worldwide scalable

3.3.4.2 Domain Name System (DNS)

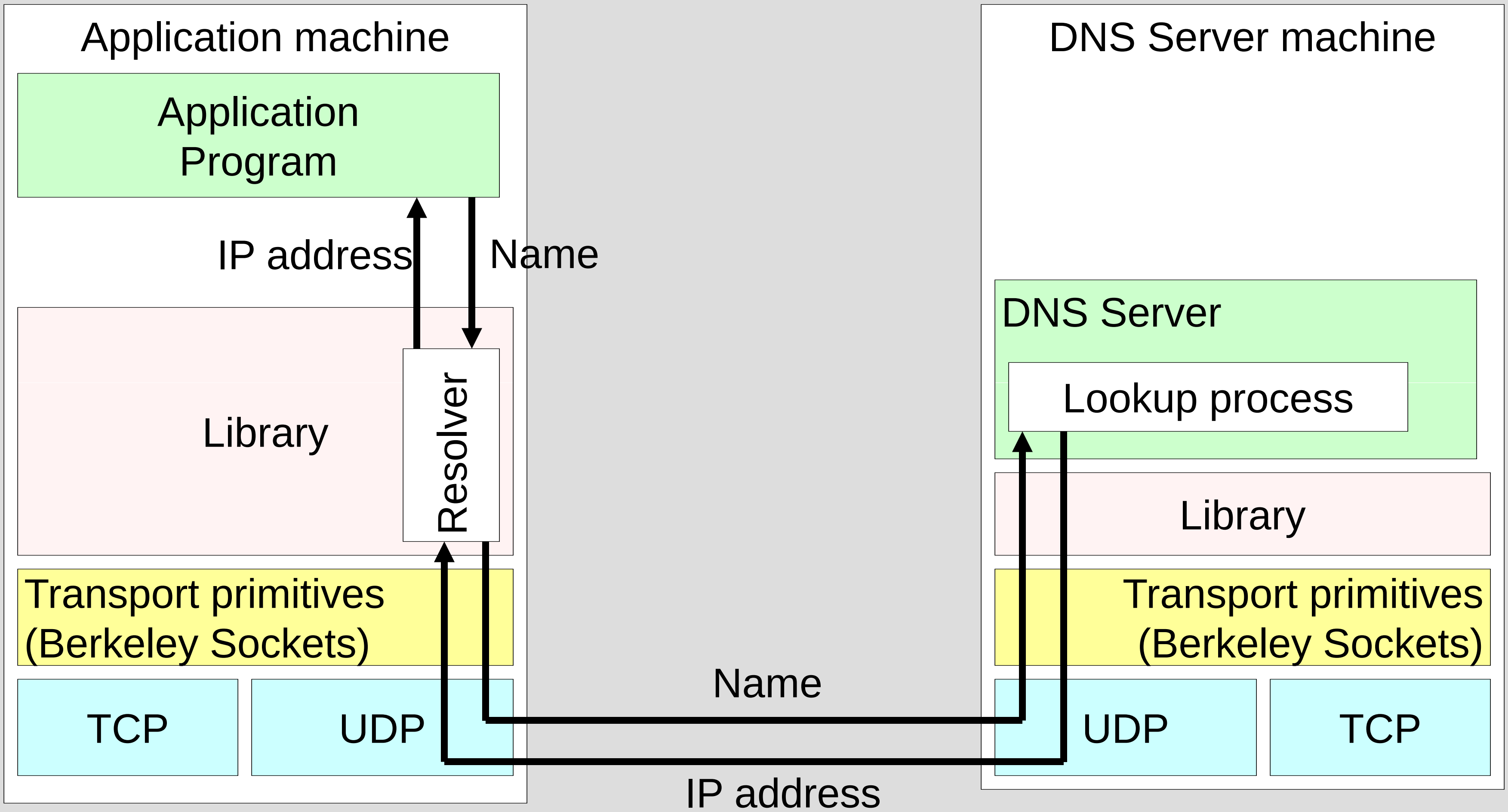
– Unique resource name
    - Name of cell followed by name used within cell
– Resource location mechanisms
    - GDS - Global Directory Service (It uses X.500 standard)
    - DNS- Domain Name Server (It uses Internet naming system)
    - ONS - Object Name System (EPC – Electronic Product Code, RFID)

3.3.4.2 Domain Name System (DNS)

- **Domain Name System**
  - Organized machines in cells – domains
  - Mapping
    - Mapping of host names and e-mail destinations to IP addresses
  - Generalized database system
    - Distributed, hierarchical, and worldwide scalable
    - Stores variety of information relating to naming

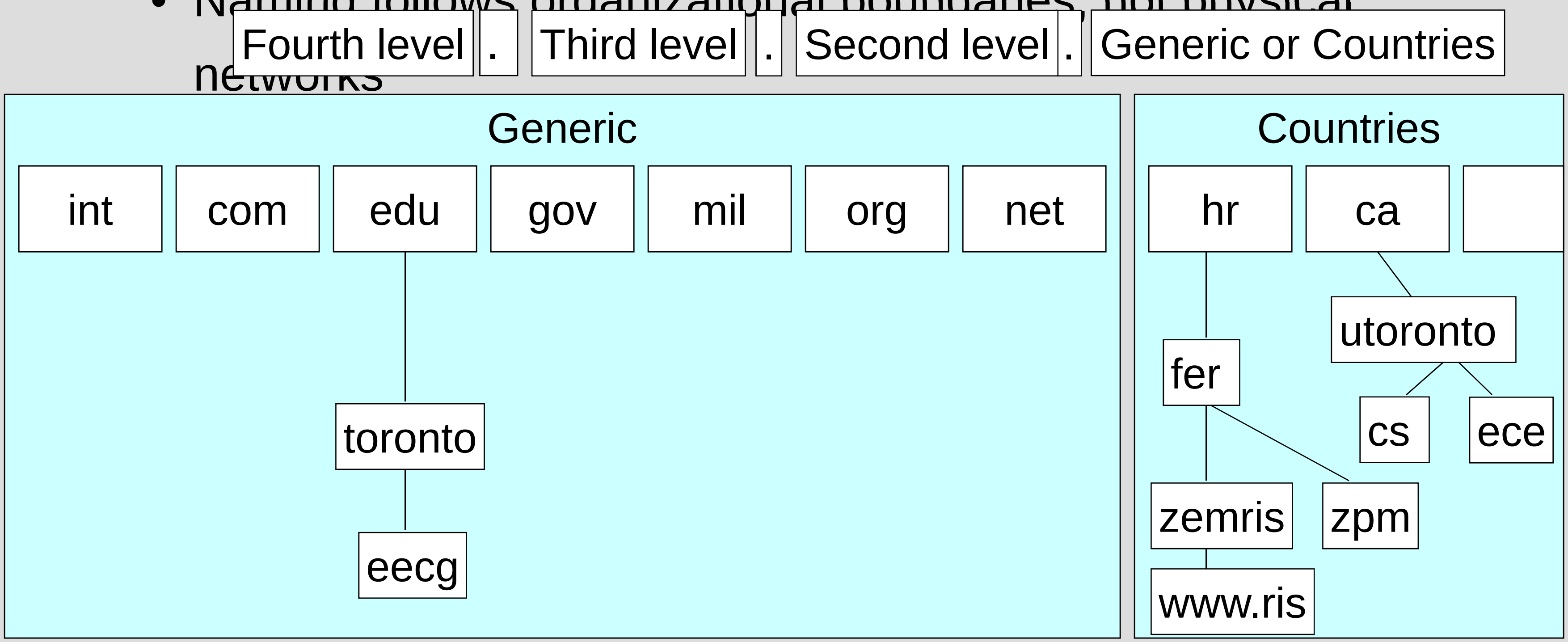
3.3.4.2 Domain Name System (DNS)

- **Mapping a name onto IP address**
  - Process initiation



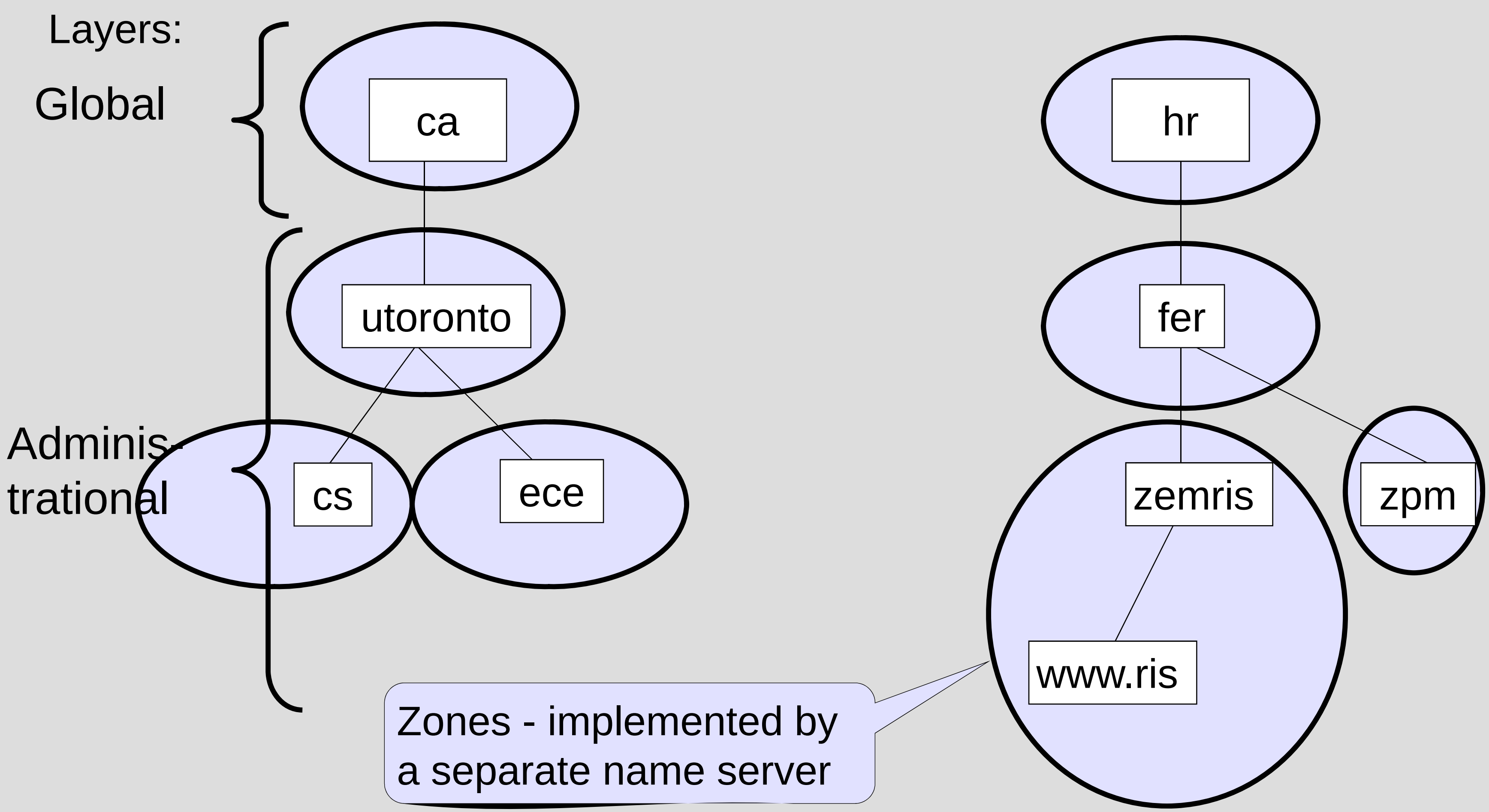
3.3.4.2 Domain Name System (DNS)

- **The DNS name space**
  - Internet is divided over 200 top-level domains
  - Domains are partitioned into hierarchy of subdomains
  - Subdomains have autonomy in naming process
  - Naming follows organizational boundaries, not physical networks



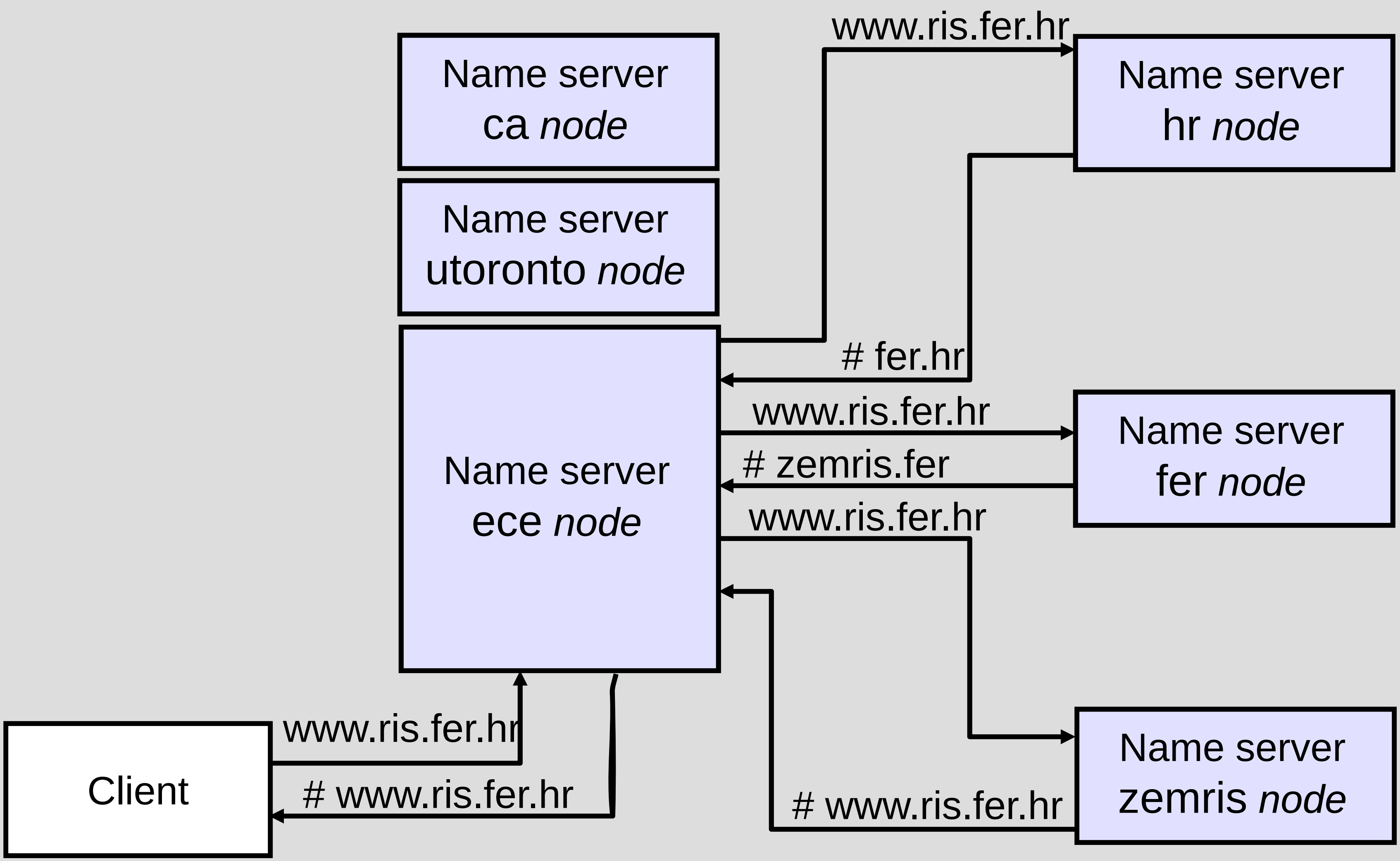
3.3.4.2 Domain Name System (DNS)

- **The implementation of DNS Name Space**



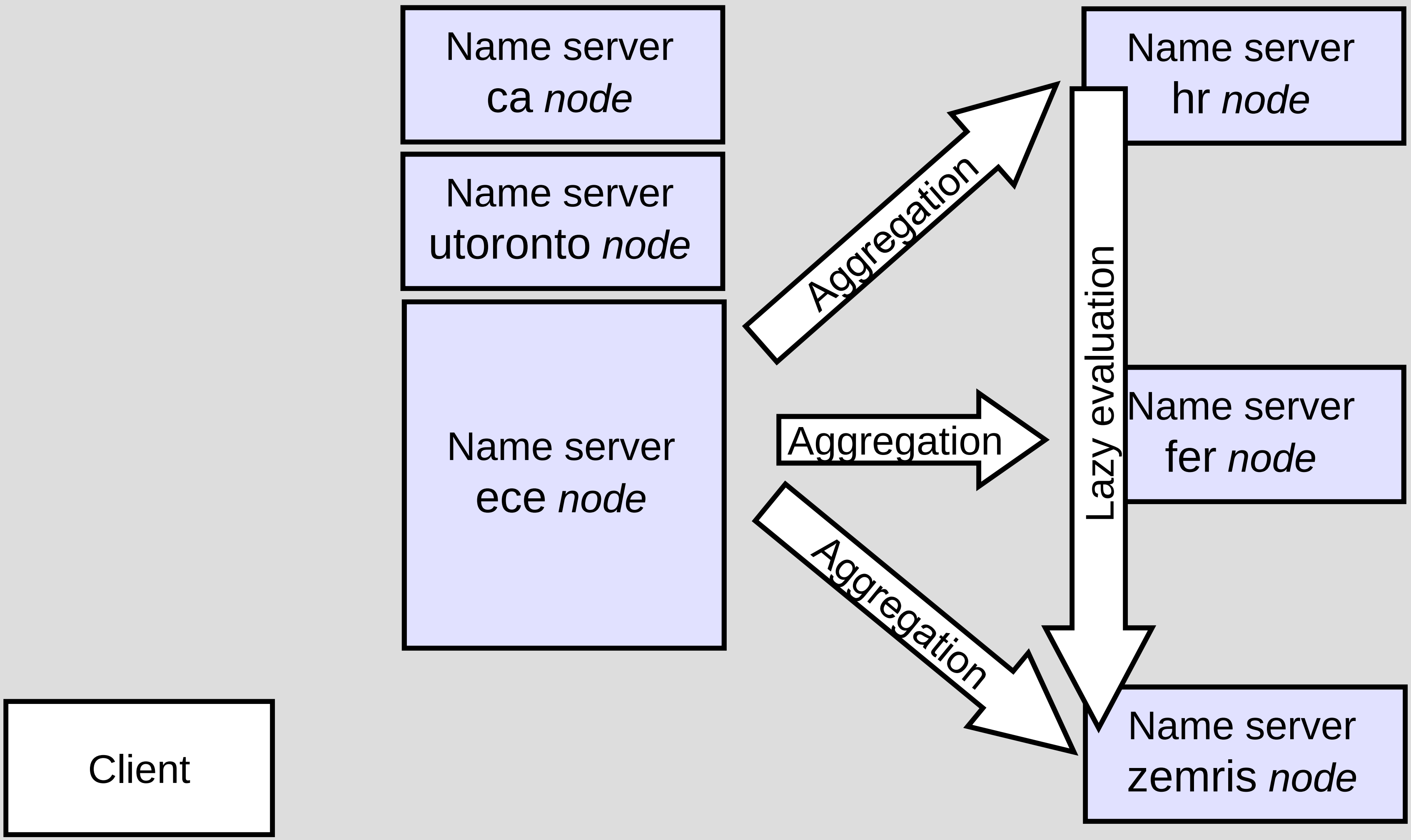
3.3.4.2 Domain Name System (DNS)

- **Iterative name resolution**



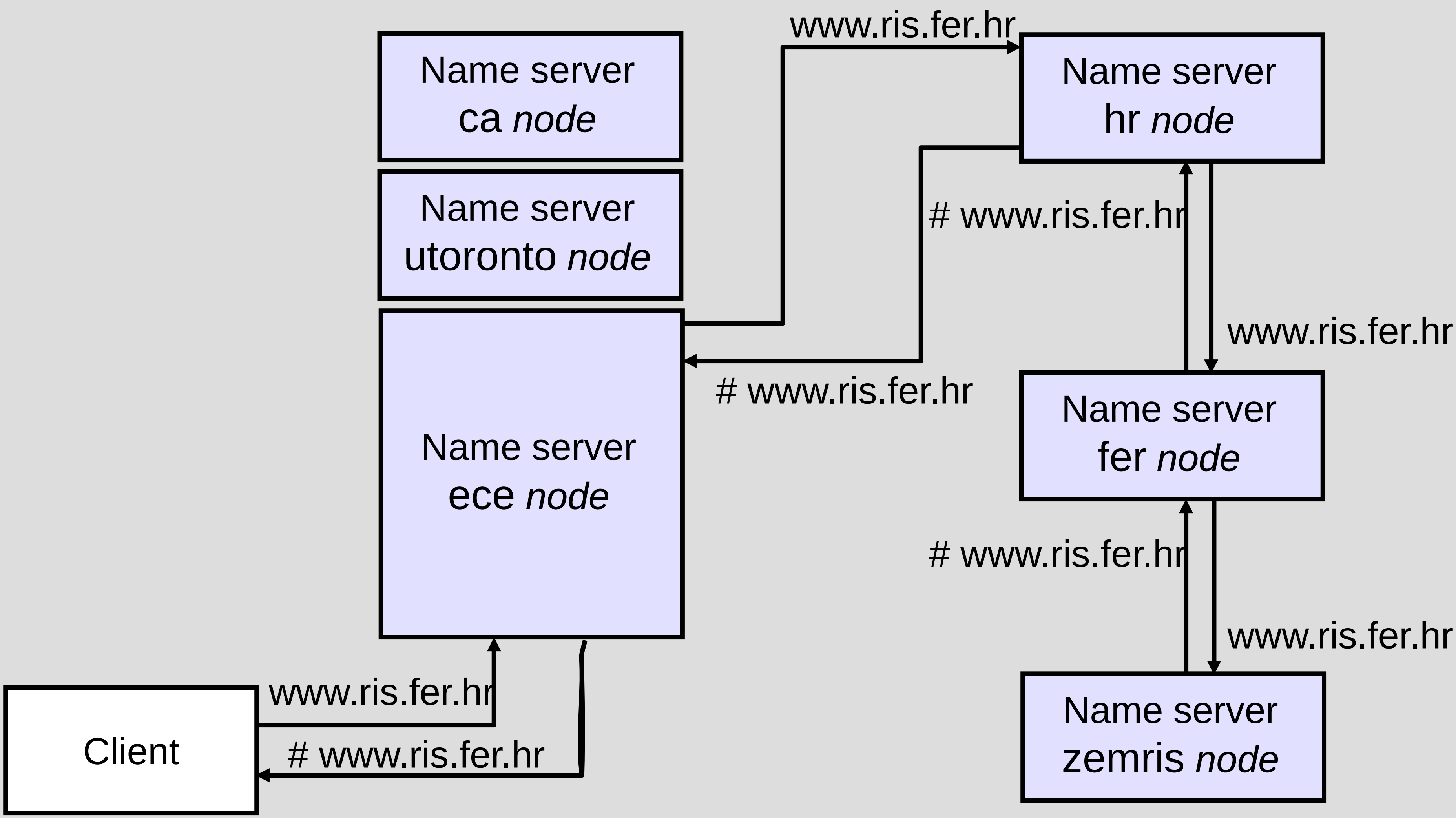
3.3.4.2 Domain Name System (DNS)

- **Iterative name resolution**



3.3.4.2 Domain Name System (DNS)

- **Recursive name resolution**



3.3.4.2 Domain Name System (DNS)

- **Iterative name resolution**