CSci 5105

Introduction to Distributed Systems

Naming

Today

- Naming continued
- Chapter 5 TVS, Active Names paper

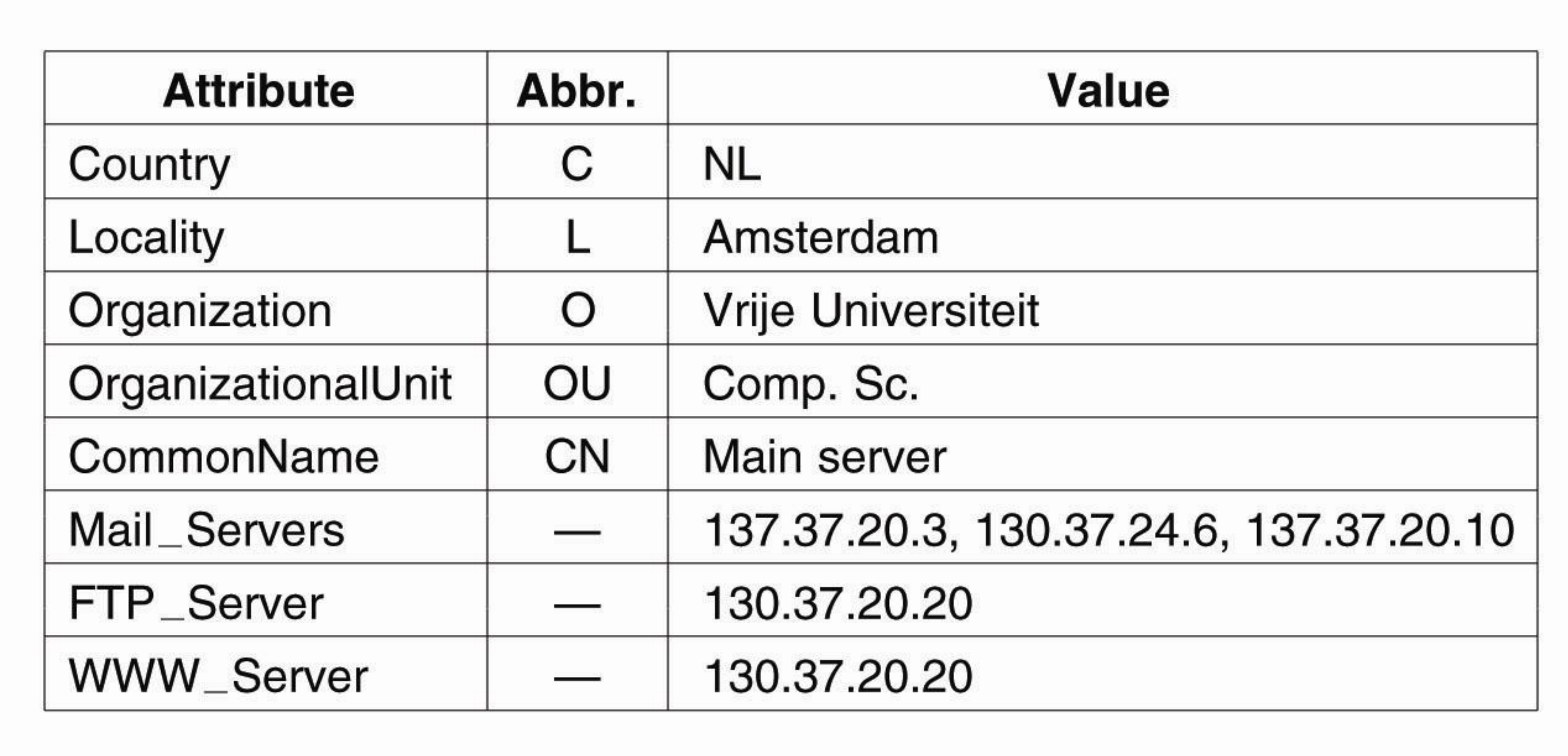
Attribute-Based Naming

- Name is attribute-value pairs
- Sometimes called directory services vs. naming services

(color:blue), (size:large), …

Example: LDAP

- Lightweight directory access protocol

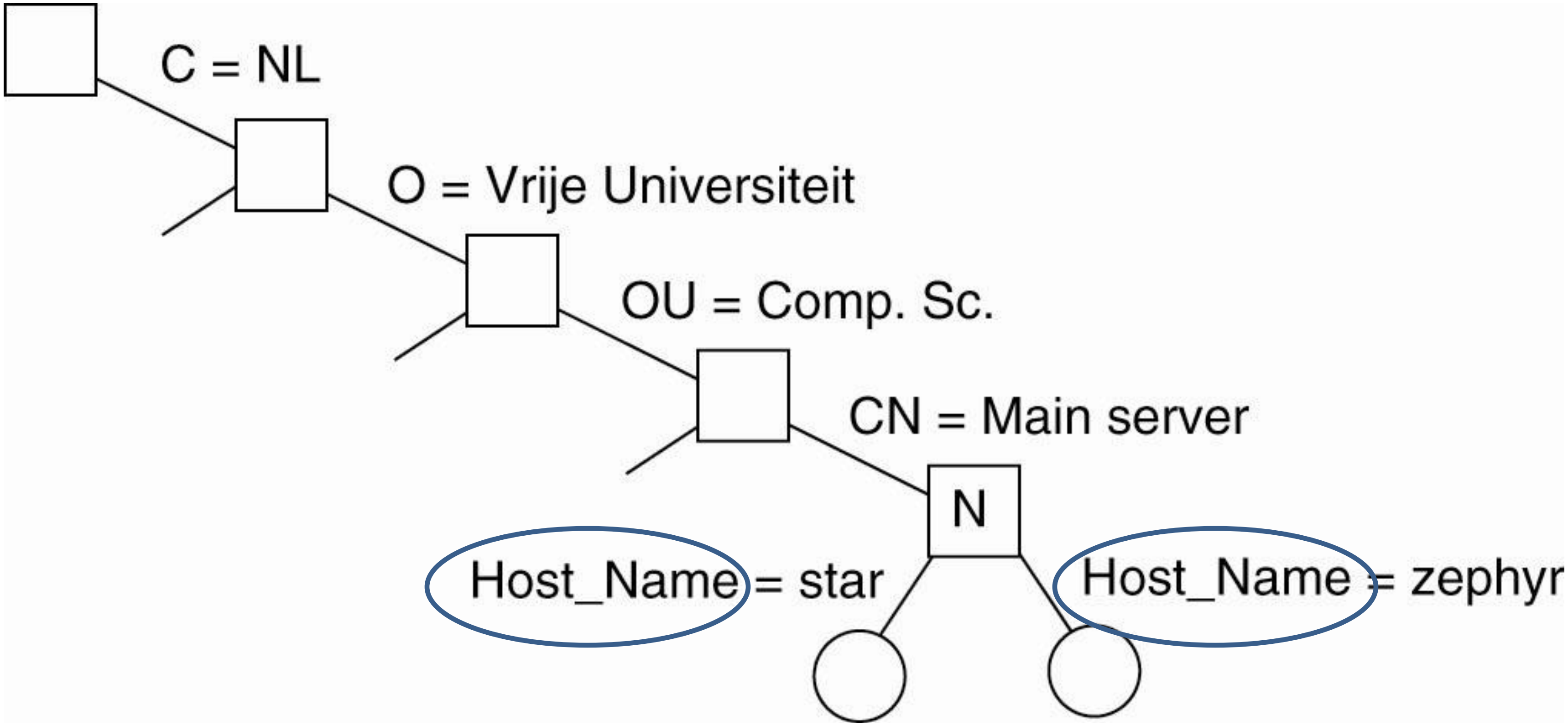


Attribute	Abbr.	Value
Country	C	NL
Locality	L	Amsterdam
Organization	O	Vrije Universiteit
OrganizationalUnit	OU	Comp. Sc.
CommonName	CN	Main server
Mail_Servers	—	137.37.20.3, 130.37.24.6, 137.37.20.10
FTP_Server	—	130.37.20.20
WWW_Server	—	130.37.20.20

- Name is: /C=NL/O=Vrjie Universiteit/O=Comp. Sc.

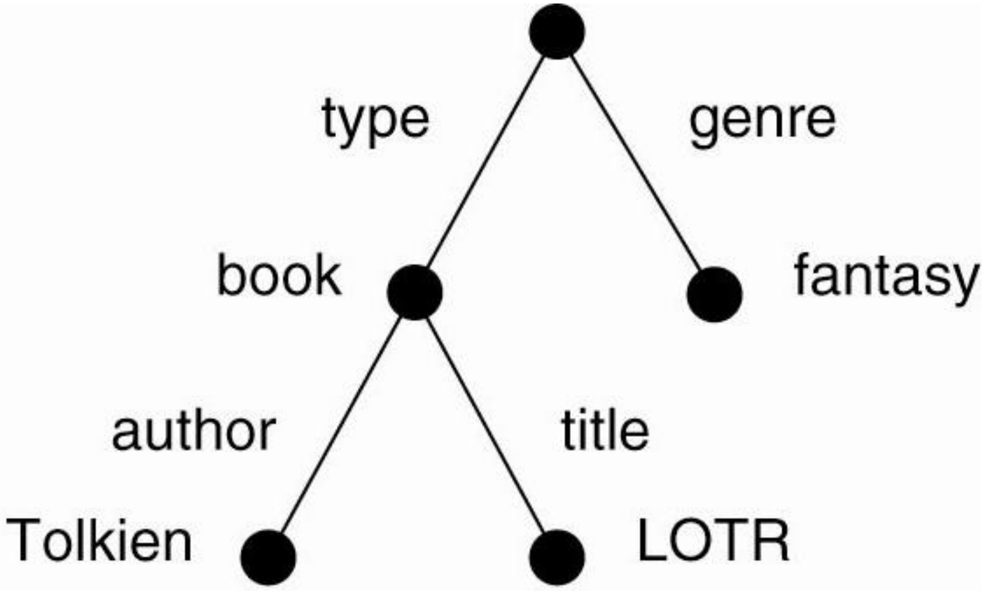
Hierarchical Implementations: LDAP

- Directory Tree



Decentralized: DHT

```
description {
    type = book
    description {
        author = Tolkien
        title = LOTR
    }
    genre = fantasy
}
```



H1: Hash (type-book)
H2: Hash (type-book-author)
…

Challenge: ranges, or + and

Naming: Active Names

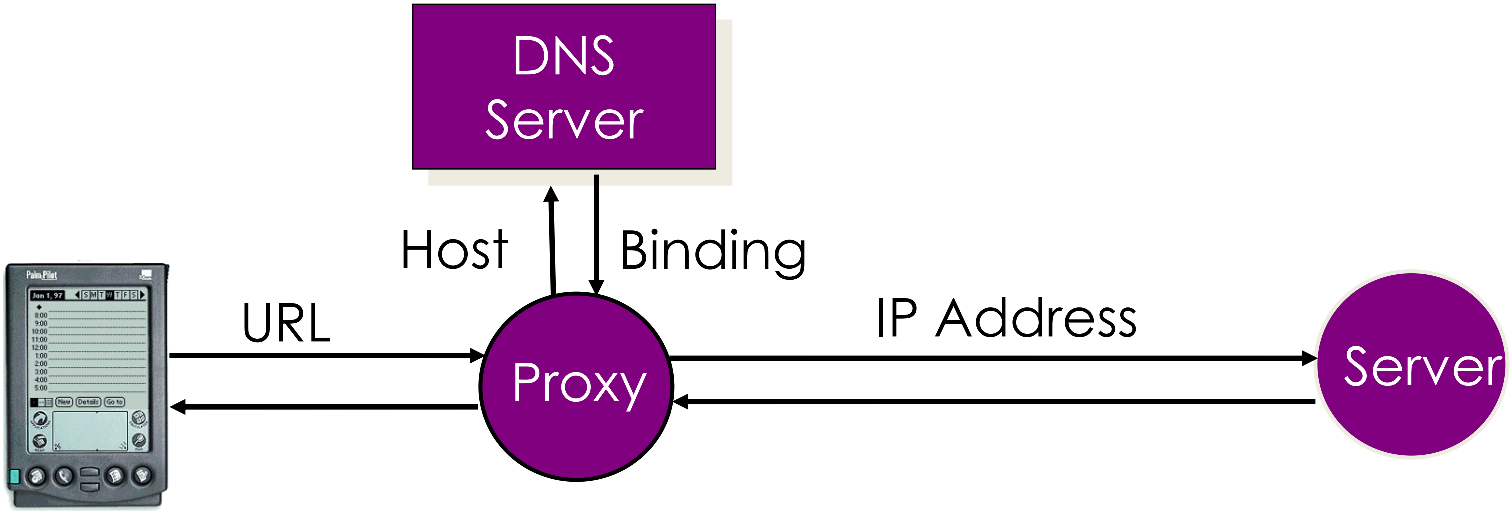
Active Names: Flexible Location and Transport of Wide-Area Resources

Vahdat, Dahlin, Anderson, Aggarwal

The Problem

- Accessing remote resources and services is limited by rigid naming schemes
  - need a way to insert flexible (i.e. customizable) services between clients and servers
  - current solutions: done inside the network elements or deferred to application
  - either client-side or strictly server-side

Traditional Internet Naming



Traditional Model

- *Static* name -> IP address binding
- Naming and transport separate
  - http AND www.cnn.com
- Not flexible or extensible

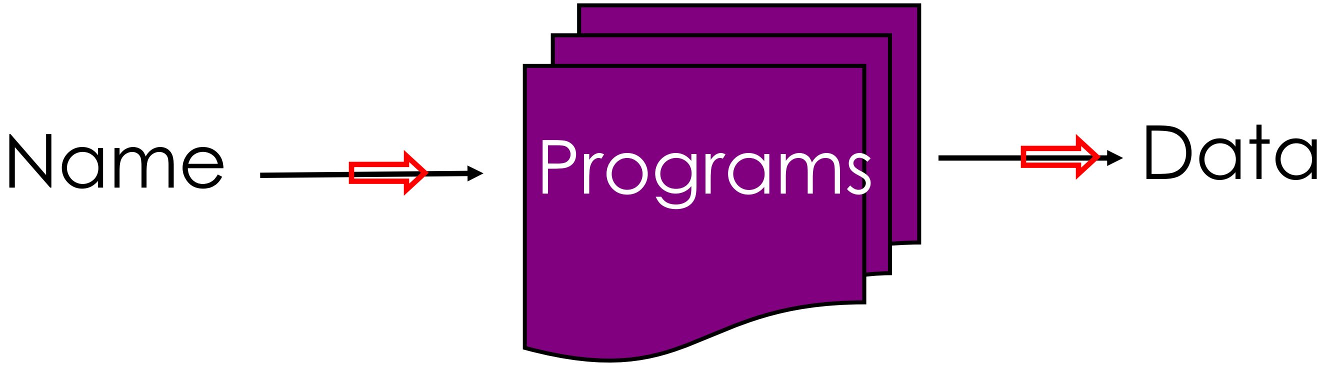
Motivation

- Consider Scenario: context-sensitive naming
  - User types cnn.com
  - If client is behind a modem, it gets back a b/w image
  - If client is a palm pilot, it gets a distilled image
  - If the client is in Europe, it goes to the European replica
- Combine naming and transport in one framework
- Provide flexibility and extensibility in the way wide area resources are accessed

Current Attempts to Add Flexibility to Name Binding

- HTTP redirect
- DNS round robin
- Cisco Local Director/Distributed Director
- Global object IDs (e.g., Globe, Legion)
- Web caches
- Mobile IP
- ... none of them are programmable

Active Names: Basic Idea



Name

Programs

Data

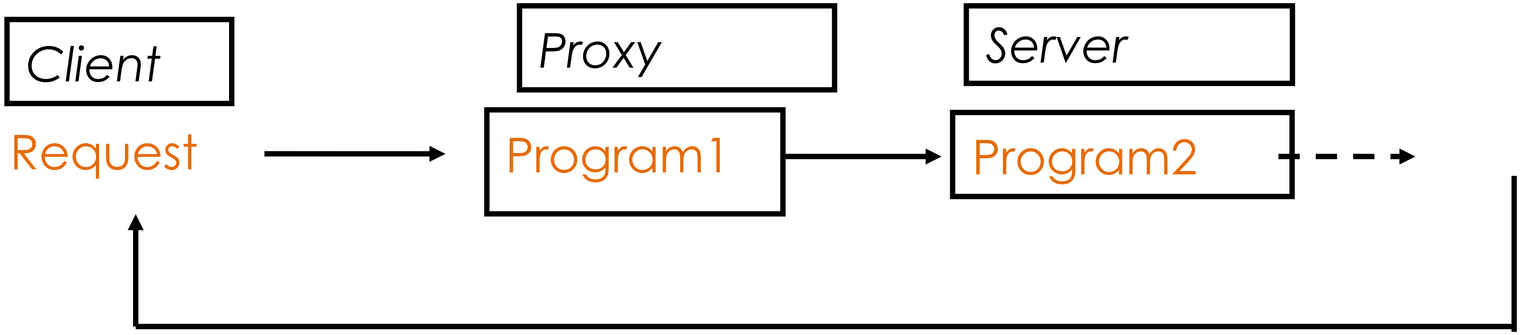
- Names resolved to mobile, secure programs
Flexibility

- Active Names organized into hierarchical *namespaces*. A program is associated with each namespace

- Namespace programs can be changed
Extensibility

- Active Names are connection oriented: better end-to-end semantics and performance

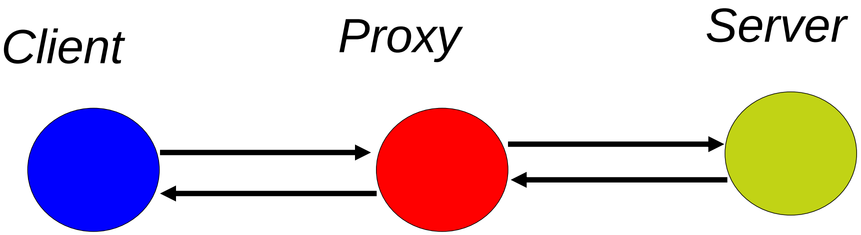
Programming Model

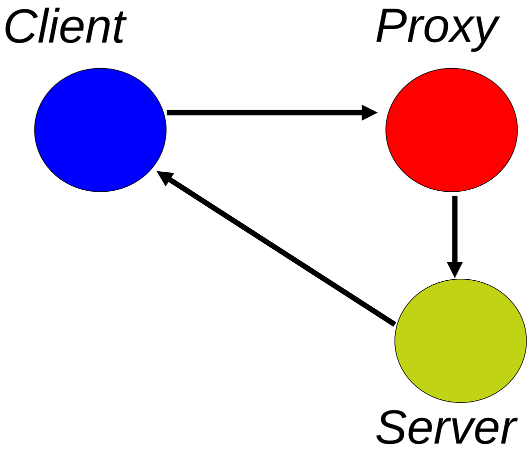


- Location independent programs
  - Programs may run on any AN node
- Stream data model
  - Each program operates on a data stream which is the result of the previous program
- Continuation passing style
  - Control does not have to return to the caller program

Performance Gains

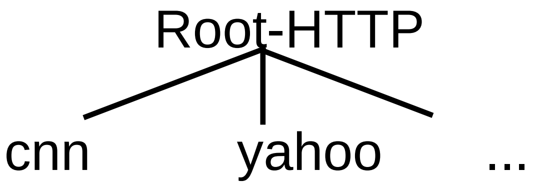
- Application customized transport protocols
- Programs are location independent. Location can be chosen to optimally utilize resources (e.g., distillation)
- Customization can be performed close to client instead of at the server (e.g., to cache dynamic content)
- 3 way RPC





Composing Services

- Delegation
  - Active names organized in a hierarchy of namespaces

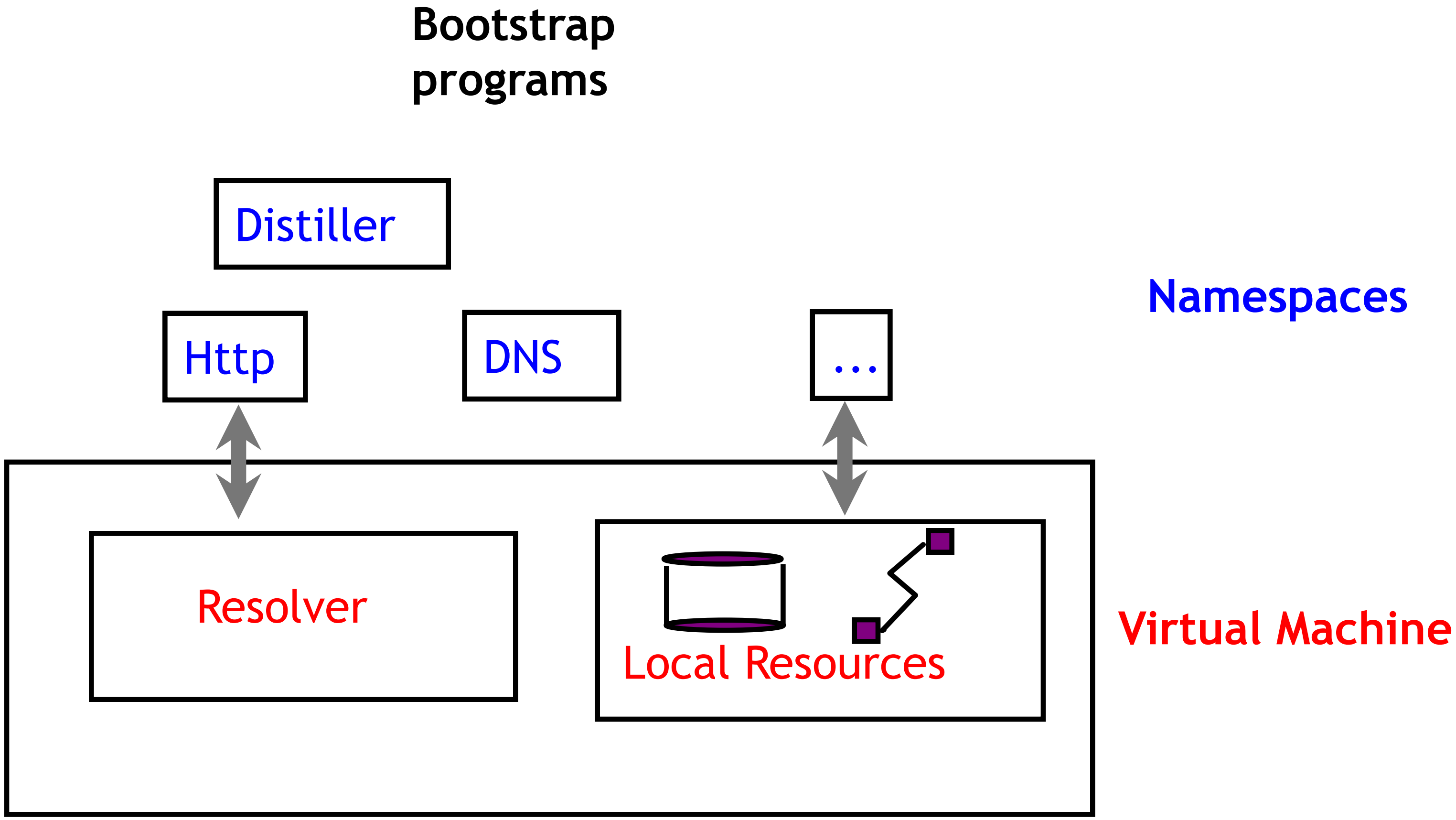


  - Namespace programs can delegate to subordinate namespaces
- After methods
  - Continuation passing style programming
  - Namespace programs bundle remaining work into “after methods” before passing control

Security

- Protection between active name programs provided by Java’s type safety mechanism
- Caller passes a certificate to the callee granting it a subset of its rights
- For instance, each caller might grant its callee the right to respond to the client
- Certificates are authenticated via encryption

Active Node Architecture



Bootstrap programs

Distiller

Http

DNS

...

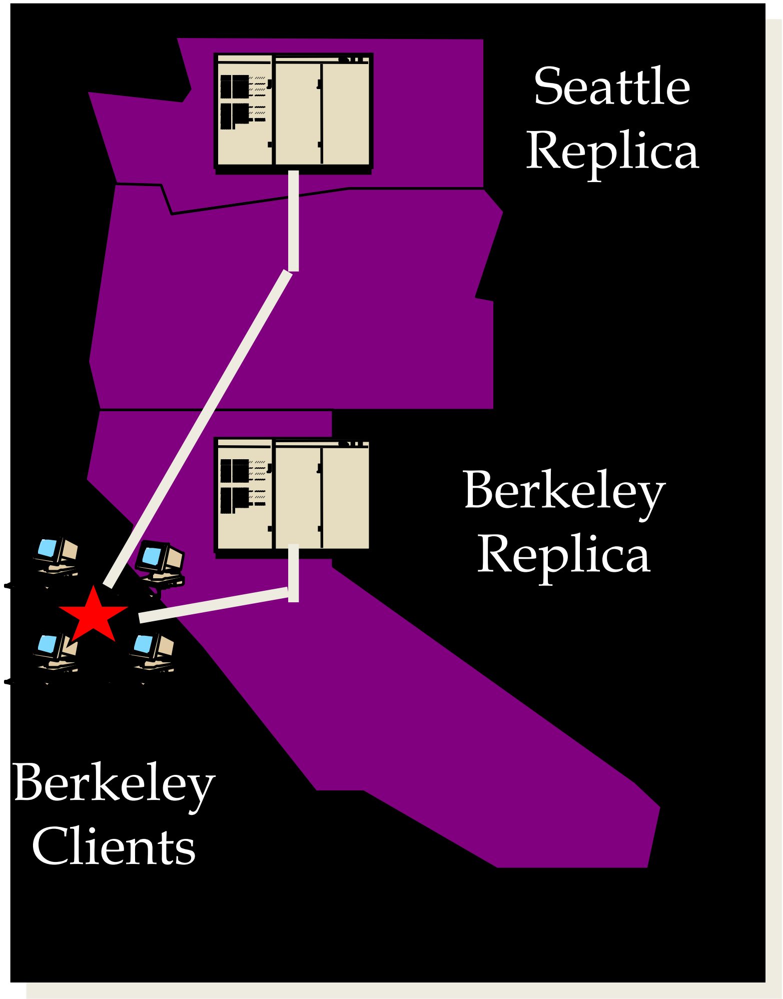
Namespaces

Resolver

Local Resources

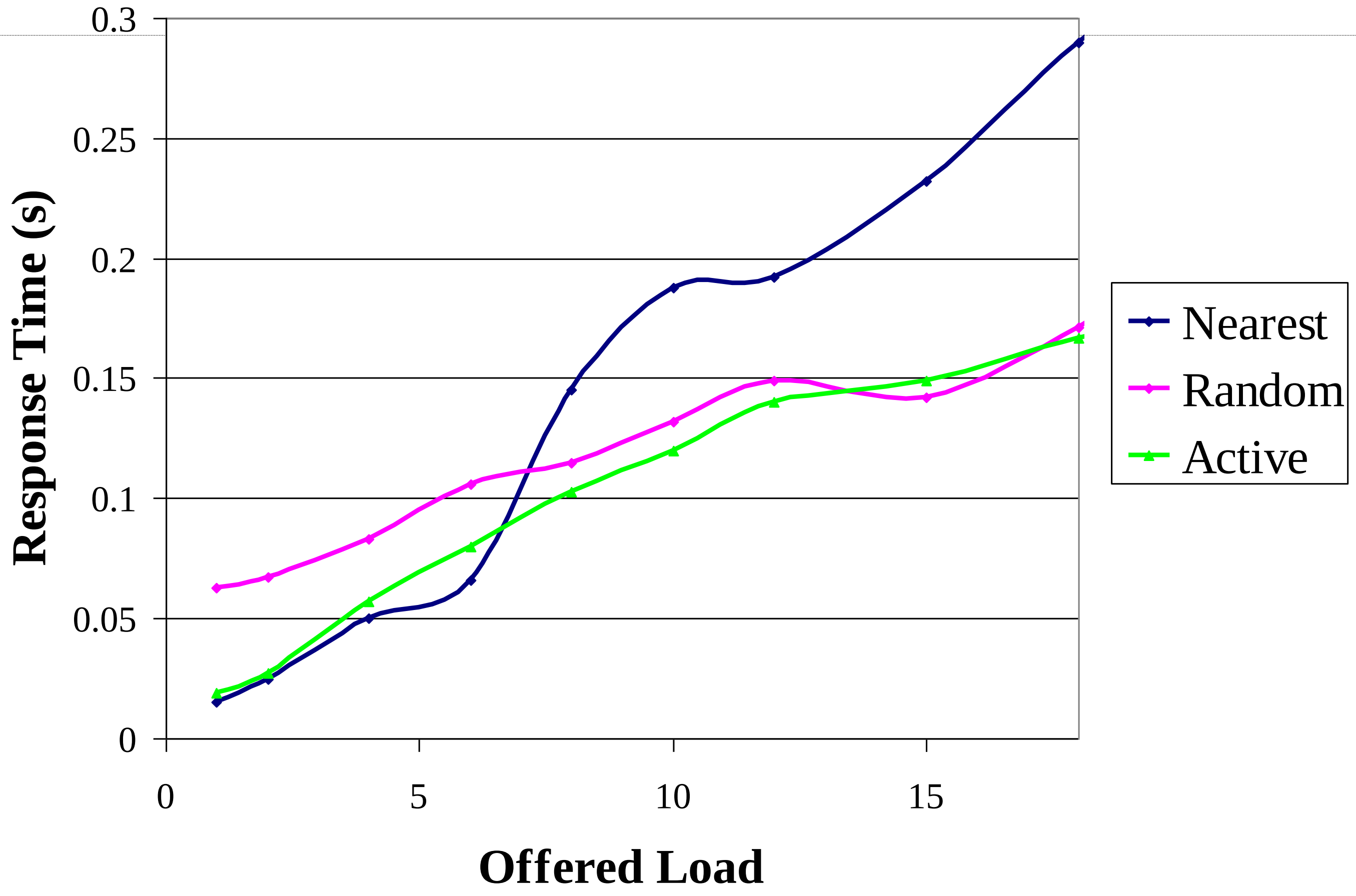
Virtual Machine

Application 1: Replica Selection



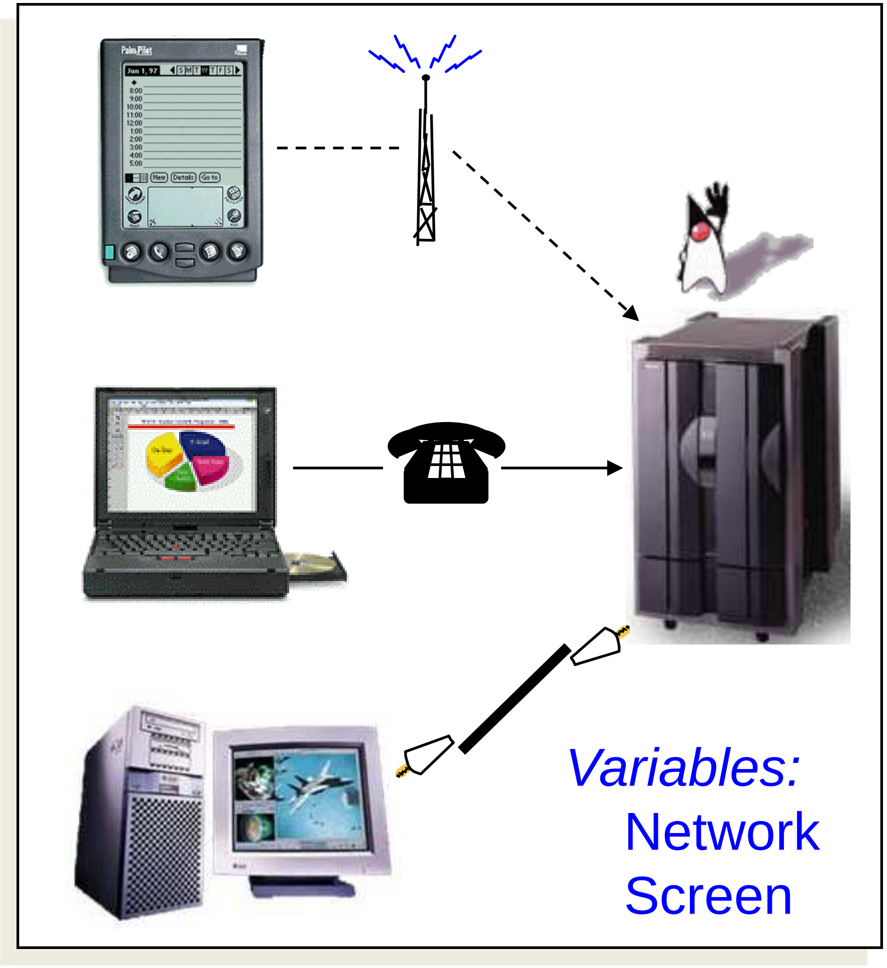
- DNS Round-Robin
  - » Randomly choose replica
  - » Avoid hot-spots
- Distributed Director
  - » Route to nearest replica
  - » Geographic locality
- Active Naming
  - » Previous performance, distance
  - » Adaptive

Replica Selection



Application 2: Mobile Distillation

Client-Specific Naming



Variables:

Network

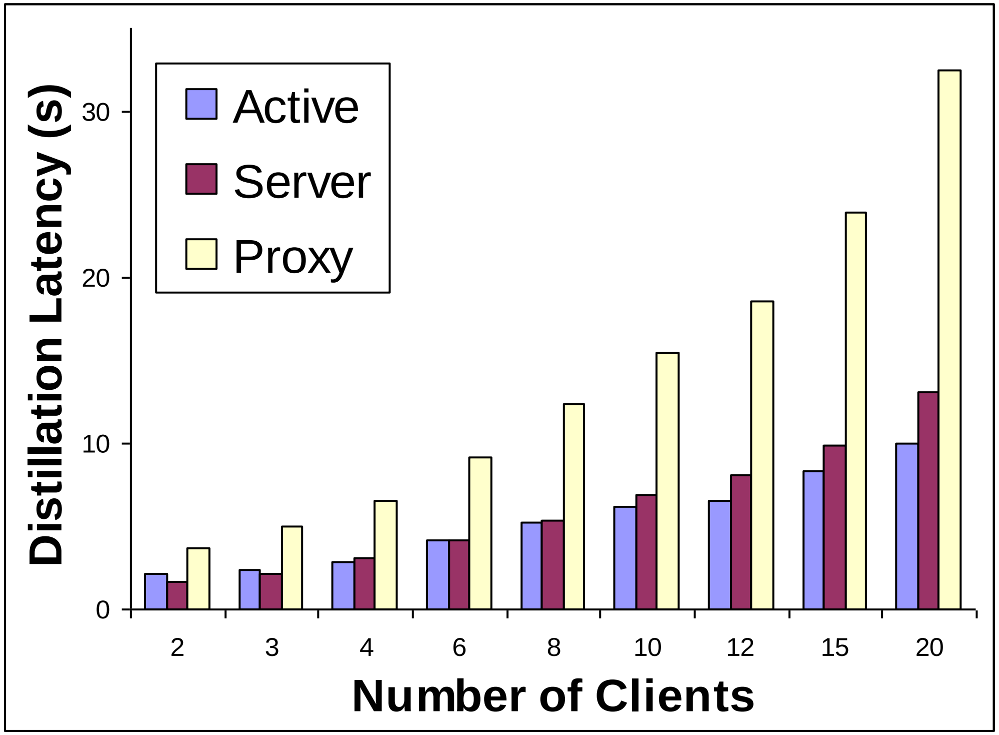
Screen

- Clients name a single object
- Returned object based on client
  - Network connection, screen
- Current approach: proxy maintains client profile
  - Requests object, distills
- Active naming
  - Transmit name + applet
  - Flexible distillation point
  - Tradeoff computation/bandwidth
  - Support mobile clients

Application 2: Mobile Distillation

Distillation at

- Server: Saves bandwidth
- Proxy: Saves server CPU cycles
- Active: Cost estimate of both approaches



Summary

- Active name paradigm
  - Decouples name from location
  - Allows specialized processing “in the network” based on client/server conditions

Next Time

Next topic: Synchronization, Mutual Exclusion

Read Chapter 6 TVS

Have great weekend!