

---

# Distributed Computing Environment

---

M. Habibullah Pagarkar

Aniket Kate

Prashant Ponde

# DCE Basics

---

- Vender Independent Distributed Computing Environment
  - Not An Os Or An Application
  - An Integrated Set Of Services And Tools That Can Be Installed As A Coherent Environment On The Top Of Existing Os
  - Serve As A Platform For Building & Running Distributed Applications
-

# Layers

Application
Operating System
Comp Hardware

DCE Applications
DCE Software
OS & Network

- Middleware Software Layer DCE
  - Conceals Differences Between Machines & Provides Transparency
-

# DCE: Blend Of Technologies

## Components:

- Thread Package
  - RPC Facility
  - Distributed Time Service
  - Name Services
  - Distributed File Services
  - Distributed Security Services
-

# DCE Cells

- Group Of Users, Machines Or Other Resources Having A Common Purpose & Share Common DCE Services
- Helps To Break Down A Large System Into Manageable Units
- Minimum Configuration :
  - Cell Directory Server
  - Security Server
  - Distributed Time Server
  - One Or More Client Machines

# Factors For Cell Boundaries

- Purpose
  - Administration
  - Security
  - Overhead
-

---

# Threads in DCE

- Fundamental building blocks
  - On top of OS+DCE components
  - Run mainly in User Space
  - Thread package – collection of user level library procedures that allow processes to manipulate threads
  - 4 States – Running, Ready, Waiting, Terminated
-

# Problems with threads

- Problem 1 – Signals
    - Synchronous – memory violation handled by current thread
    - Asynchronous – external given to the threads (if any) waiting for it
-

# Problems in Threads

- Problem 2 – standard library
    - If allocating memory and thread switched : inconsistent data
  - Solution
    - Jackets around some library procedures (read, fork are jacketed)
    - mutex
-

# Problems in Threads

---

- Problem 3 – Error No.
    - ❑ System calls returned with error no.
    - ❑ If one makes call, but just after completion another scheduled, error no. gets over-written.
  - Solution
    - ❑ Thread specific version of err no. saved and restored
-

# Scheduling

- Similar to scheduling processes
  - Threads in DCE have priorities
  - DCE has three scheduling algorithms
    - FIFO
    - Round Robin
    - Default
-

# Synchronisation

## ■ mutex

- ❑ Used when it is essential to prevent multiple thread from accessing the same resource at the same time. Eg. Moving through a linked list
- ❑ Fast mutex – analogous to a lock
- ❑ Recursive mutex – if subroutine also wants a lock, it is granted
- ❑ Attempt to lock a mutex that is already locked returns an error

## ■ Condition Variables

- ❑ Conjunction with mutex
-

# RPC

- DCE based on client/server model
  - RPC system makes it possible for a system to access a remote system by simply calling a remote procedure
  - RPC hides details from clients
  - Handles message transports in both directions
  - As a result, client and servers are highly independent
-

# RPC

- Interface definition defines the syntax of calls
  - Written in IDL – Outputs header, client stub, server stub
  - Server location done in 2 steps
    - Locate server machine
    - Locate correct process on that machine
  - To communicate reliably, an end point is required which are dynamically assigned
  - Performing RPC – At-most-once
-

# Security In DCE

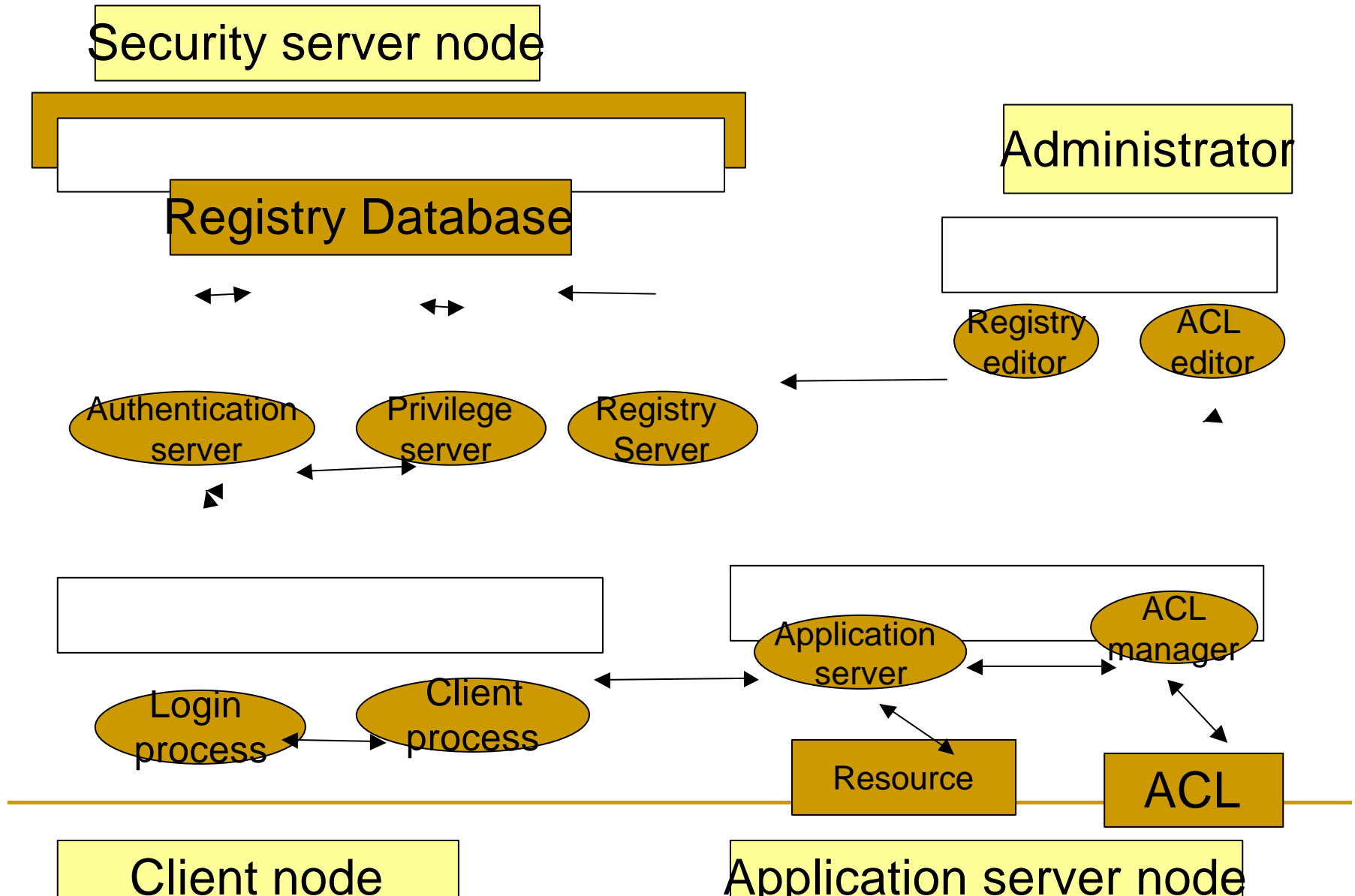
A Security Infrastructure That Provide:

- Authorization
- Authentication
- Encryption At Application

Selectable Levels

- Central Security Database

# Main Components



# DCE Distributed File System

- Derived from Andrew File System
  - Uses threads to handle multiple file access requests simultaneously and RPC for client/server and directory service
  - File Model – unstructured. One file can contain  $2^{42}$  bytes
  - File System Model –  
Files → Directories → Filesets → Aggregate.
  - One disk partition has one aggregate
-

# DFS

- Distributed File Services uses data caching model
  - DFS server machine has following components
    - Episode
    - Token manager
    - File exporter
    - Fileset server
    - Fileset location server
    - Replication server
-

# File Sharing / File Caching

- Every read operation on the file sees the effect of all previous write operations performed on that file
  - Local memory used for caching in diskless terminals
  - As long as client possesses the token for specified operation, data is valid.
-

# Replication

- Filename mapped to all file-servers having a replica of the file.
  - Unit of replication is a fileset
  - Replication server is responsible for maintaining consistency
  - Updates only on primary copy, replication server periodically sends updated copy outwards
  - Every update to a file results in a log
-

---

THANK YOU.....

---