

# **Multi-level Resource Management: Making the Entire System Work Together**

Douglas M. Wells

The Open Group

<d.wells@opengroup.org>

# Multi-Level Resource Management

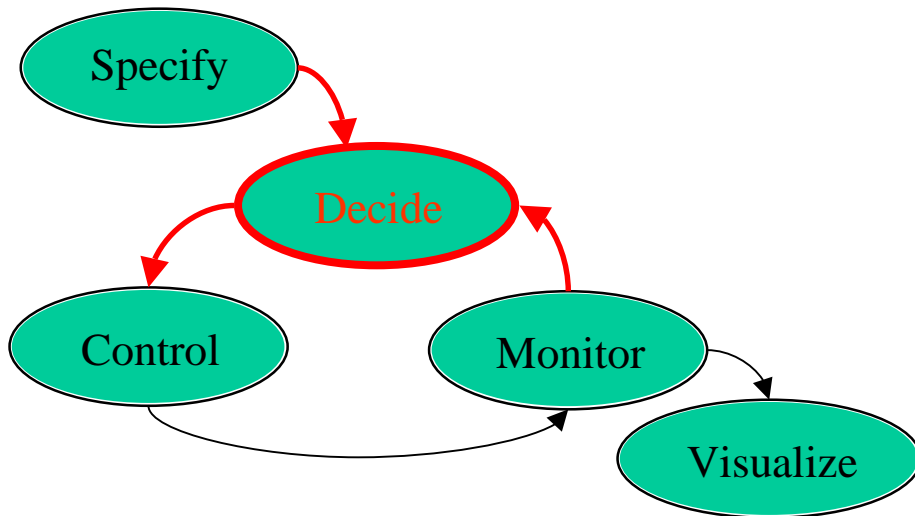
THE *Open* GROUP

## Outline

- ❑ **An Overview of the Model**
- ❑ A Reduction to Practice  
—w/ Example Applications
- ❑ Technical Challenges

# Resource Management

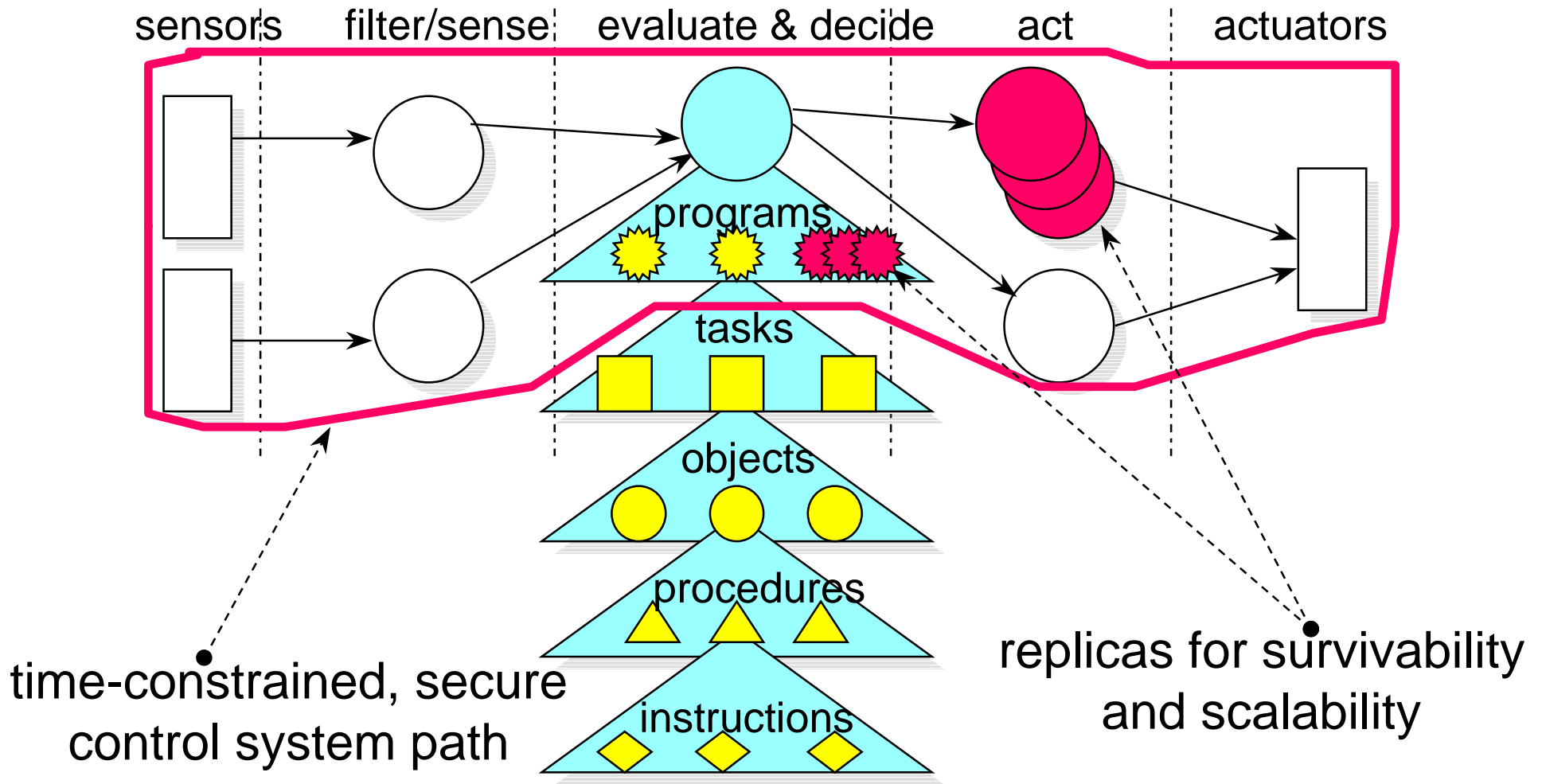
THE *Open* GROUP



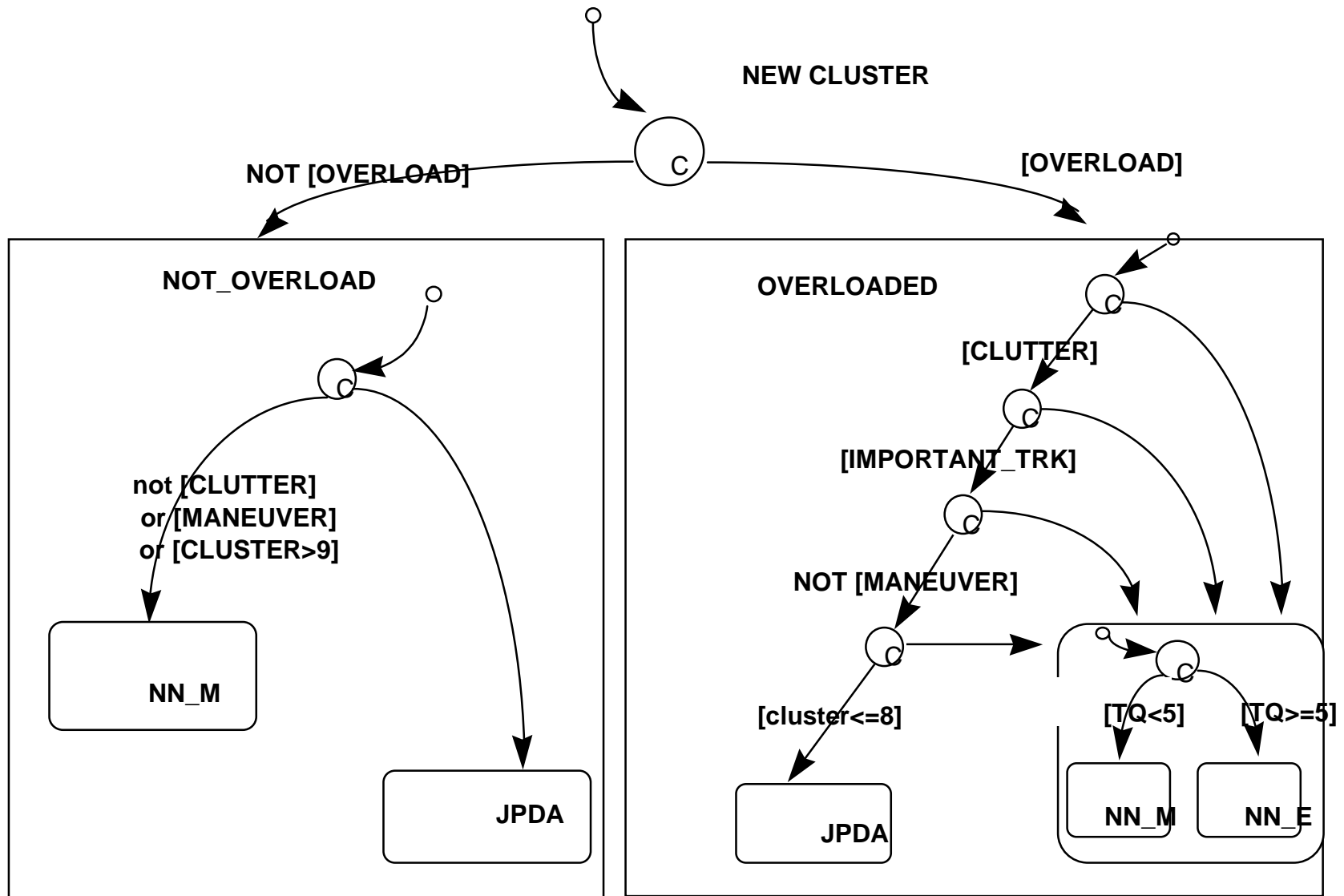
- ❑ QoS management part of existing organizational structures and processes
  - Set service levels
  - Monitor for compliance
  - Take action to correct variance
- ❑ QoS automation mirrors human structures/processes
  - Faster response
  - Increased scale



# Application & QoS Models (DeSiDeRaTa)



# Radar Algorithm Selection (AWACS Surveillance Tracking)



# Purpose of a Resource Manager

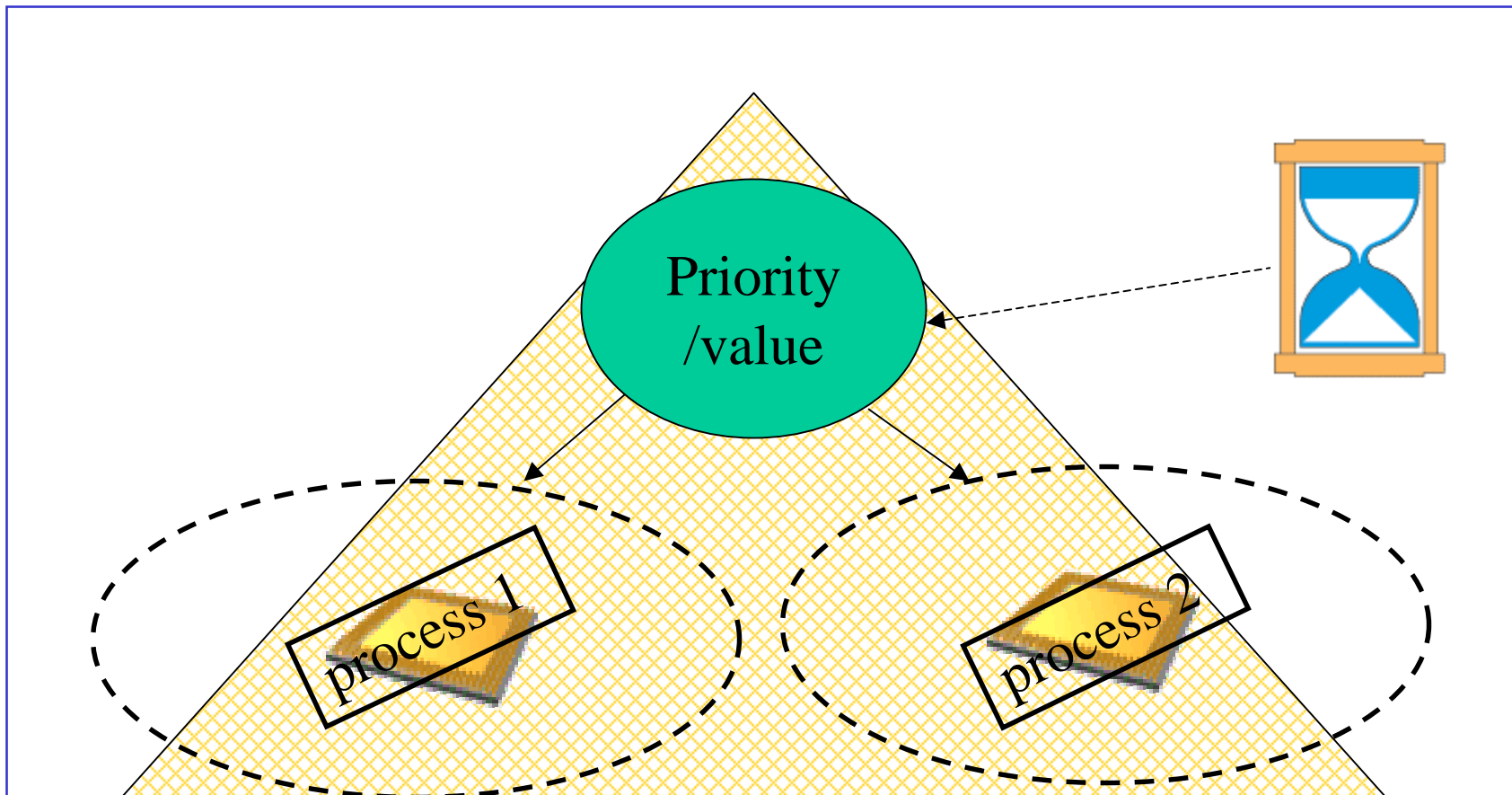
THE *Open* GROUP

- Allocate available resources to—
  - Achieve system objectives
    - Maximize benefit
    - Minimize costs
- Reconfigure system based on—
  - Changes in environment
  - Changes in system objectives

# RM Example #1

THE *Open* GROUP

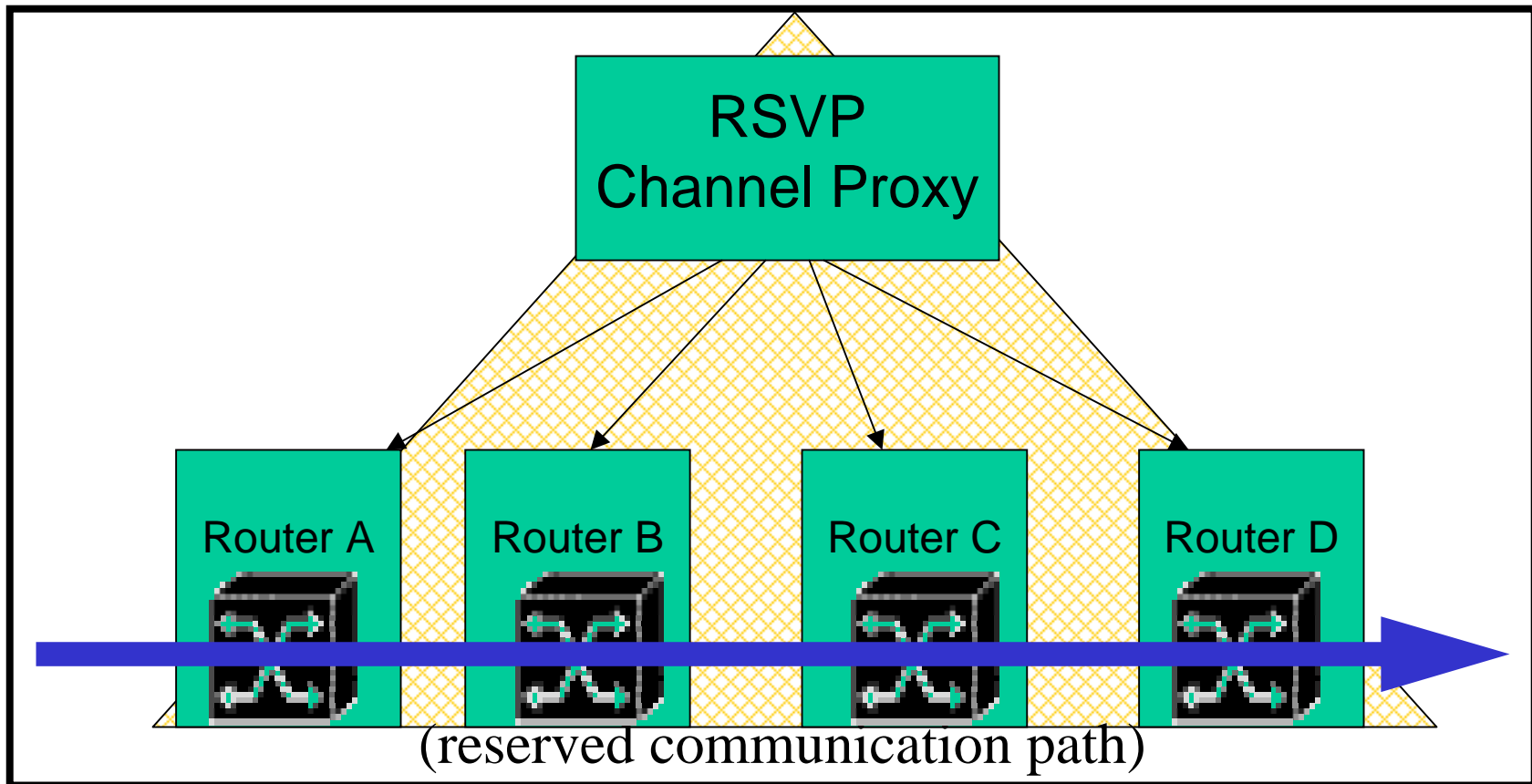
## Operating System CPU Scheduler



# RM Example #2

THE *Open* GROUP

## RSVP Internet Link

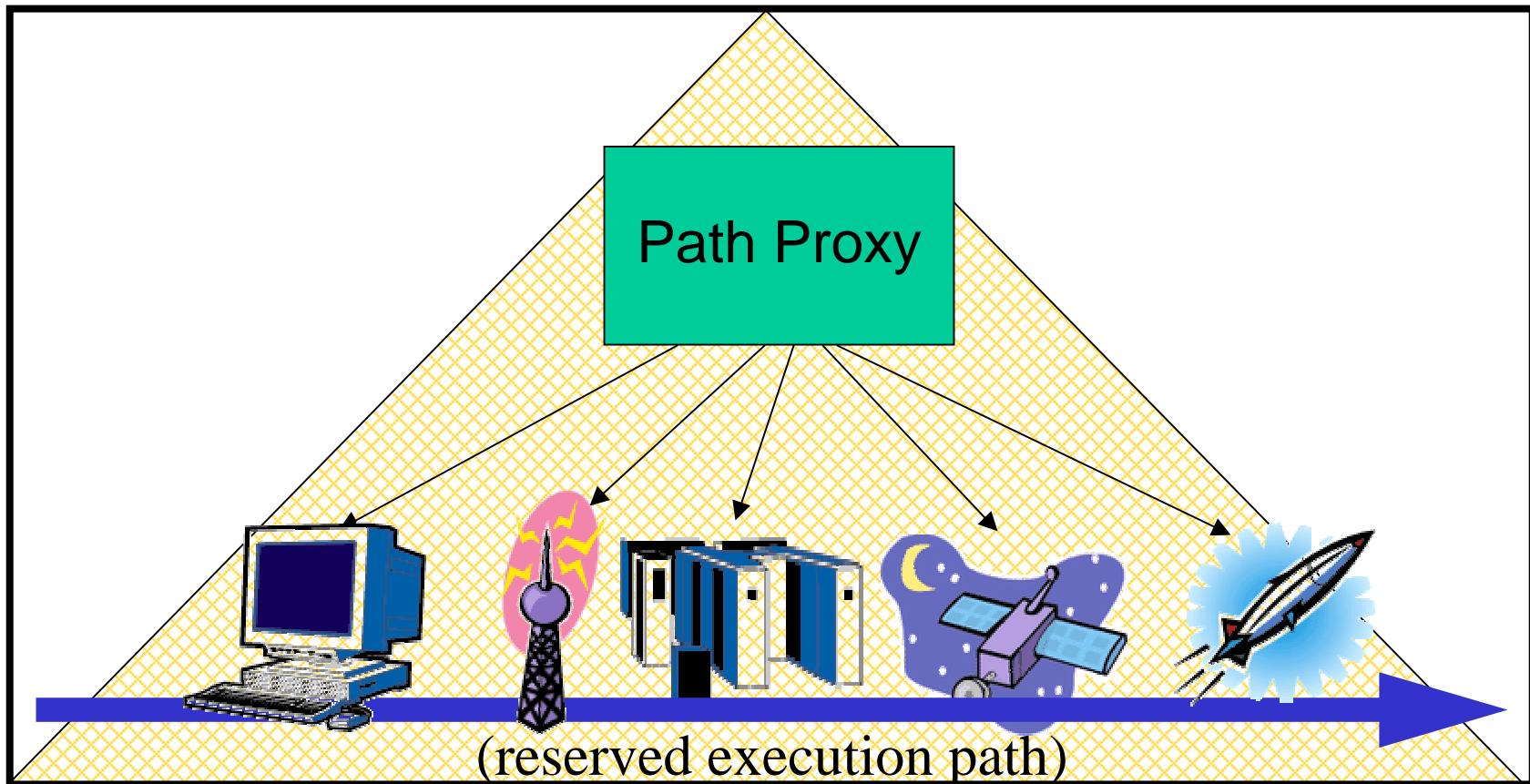




# RM Example #3

THE *Open* GROUP

## End-to-End Execution Path



# Resource Management Model

THE *Open* GROUP

## Goals

- ❑ Simplify reconfiguration of resources
- ❑ Simplify reassignment of tasks
- ❑ Allow composition of separate subsystems
- ❑ Manage wide range of resources
- ❑ Manage wide range of applications
  - Multiple applications
    - Competing
    - Cooperating
- ❑ Standardization and certification

# Resource Management Model

THE *Open* GROUP

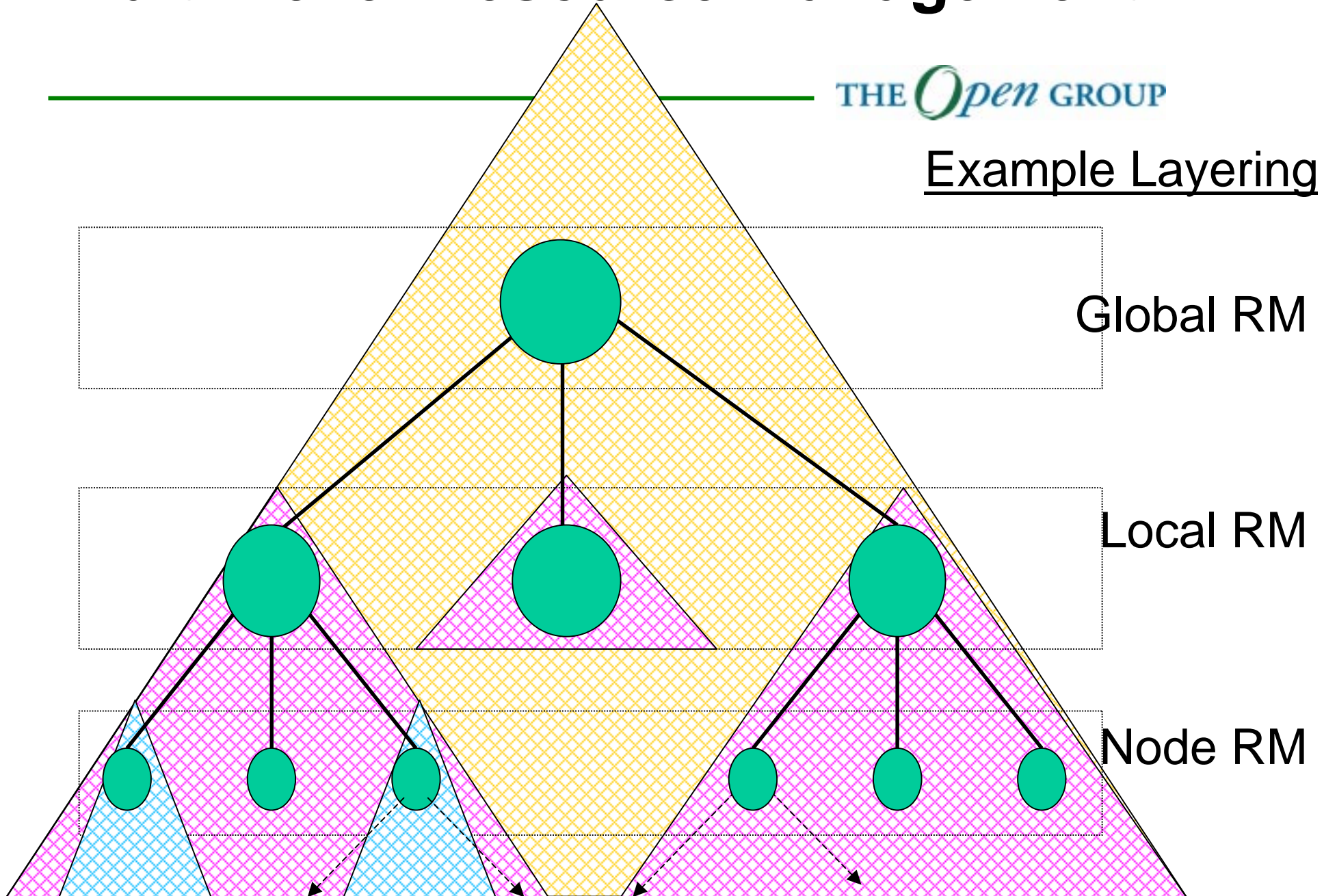
## Abstract Architecture

- ❑ Composable
  - Common interfaces
- ❑ Separation of policy and mechanism
  - Policy—application/mission specific
  - Mechanism—implementation specific
- ❑ Modeled on practical computer configurations augmented by human organization principles

# Multi-Level Resource Management

THE *Open* GROUP

## Example Layering



# Resource Domain

THE *Open* GROUP

A Resource Domain is—

- —a natural set of resources
- —organized together in a
- —hierarchical fashion to
- —perform a common function
- —under the control of
- —a single resource manager.

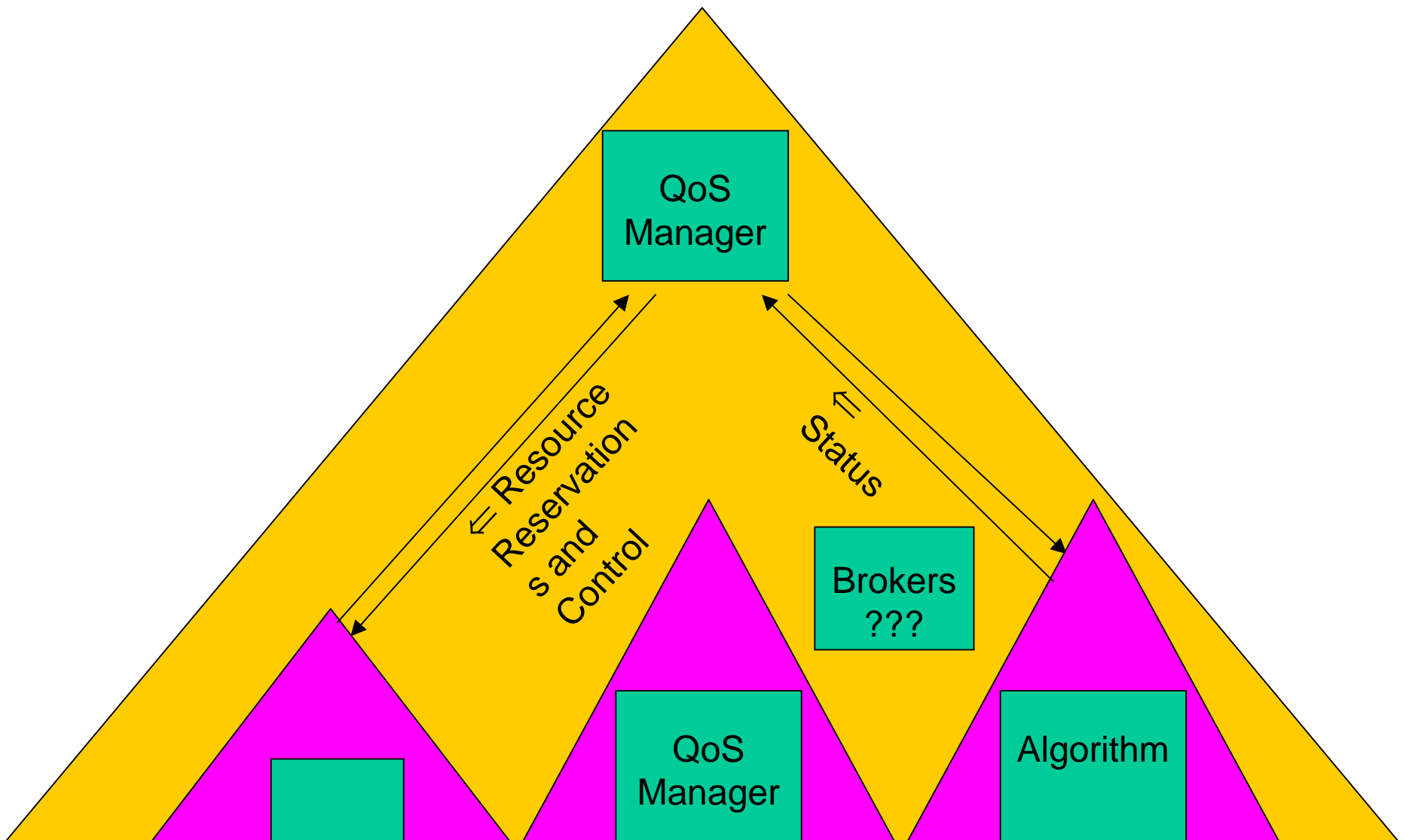
# Resource Management Roles

THE *Open* GROUP

- ❑ Application
- ❑ Resource controller
- ❑ Resource arbitrator
- ❑ QoS manager
- ❑ Resource broker/trader
- ❑ Metrics system
- ❑ Communication infrastructure

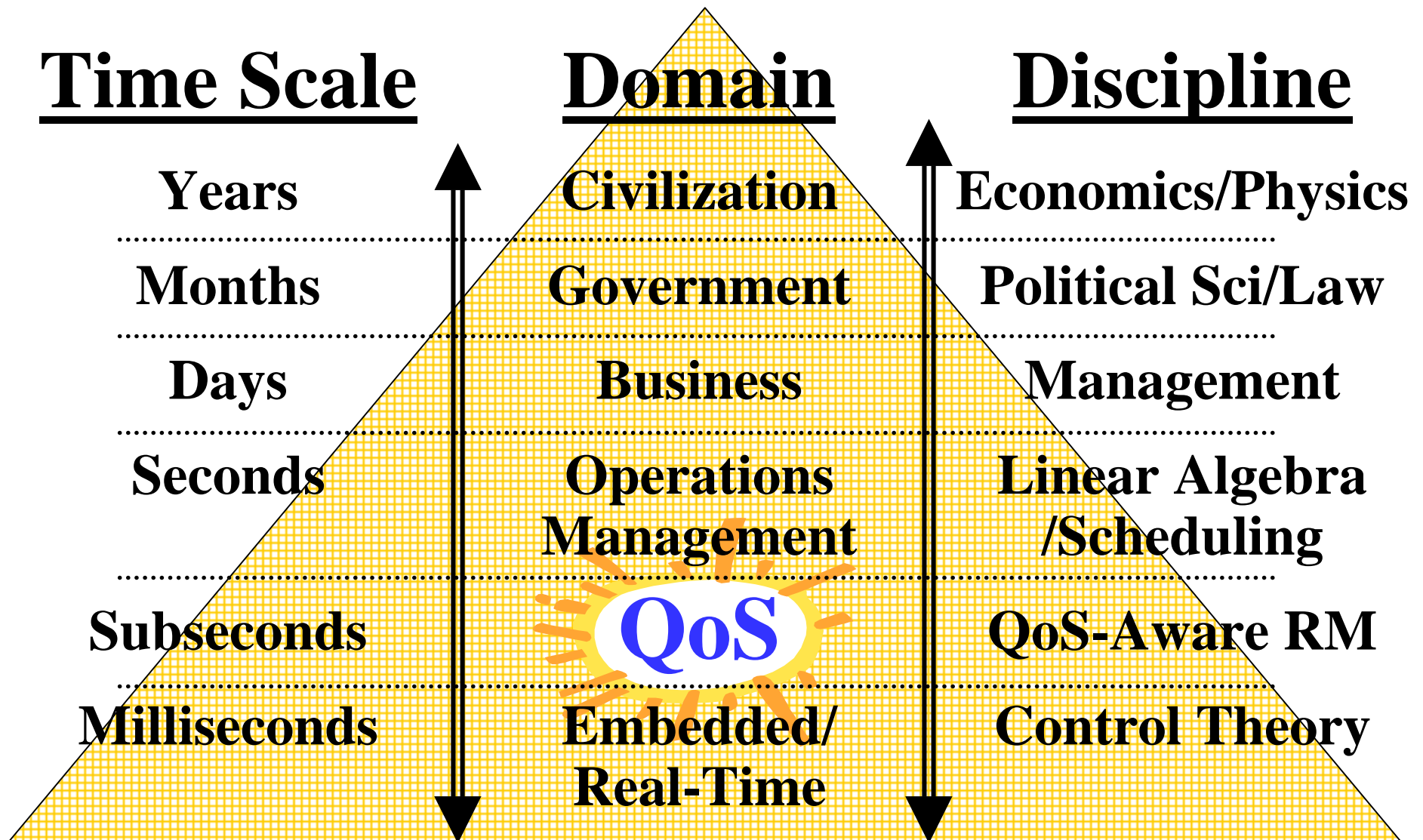
# Composable Resource Management Architecture

THE *Open* GROUP



# Some (Over-)Generalizations

THE *Open* GROUP





# Multi-Level Resource Management

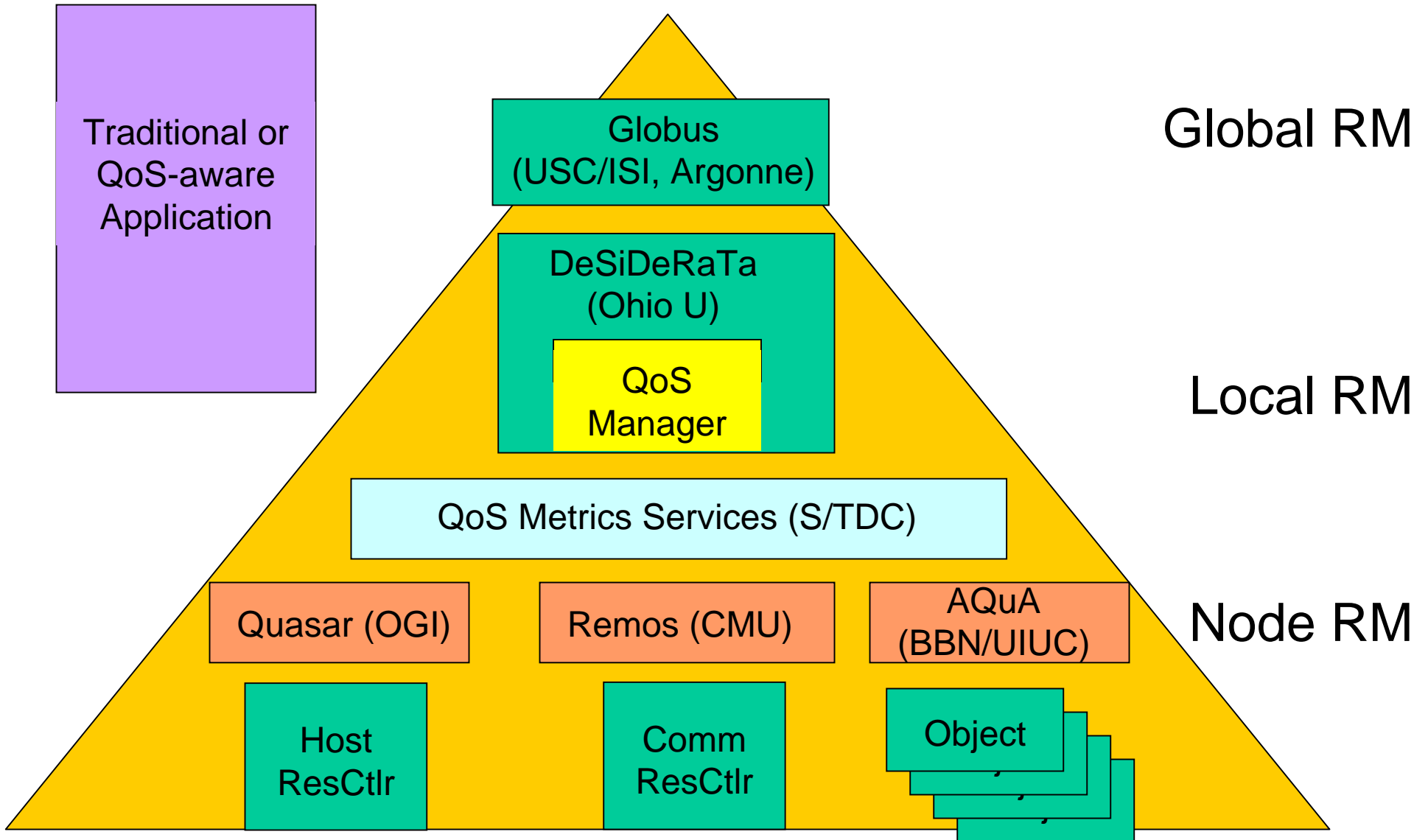
THE *Open* GROUP

## Outline

- ❑ An Overview of the Model
- ❑ **A Reduction to Practice  
w/ Example Applications**
- ❑ Technical Challenges

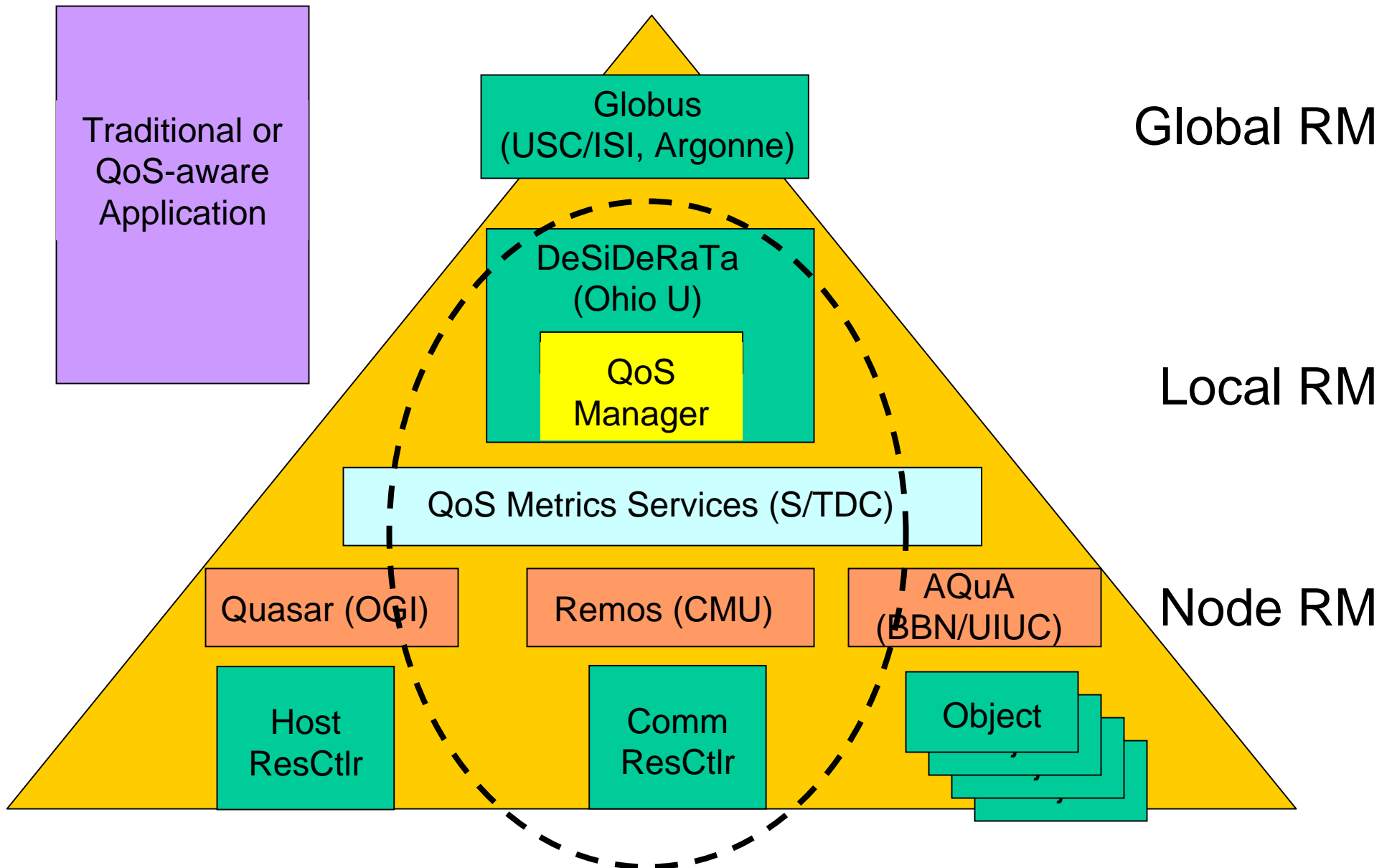
# QUITE Resource Management System

THE *Open* GROUP



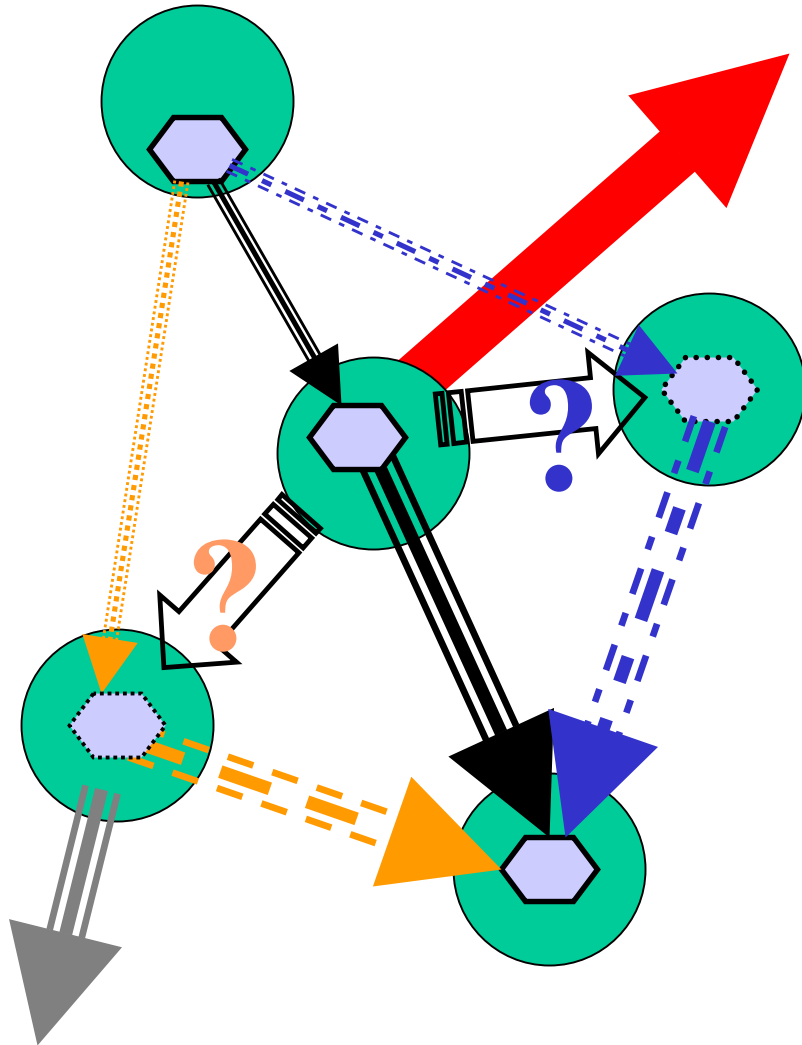
# QUITE Experiment on Path-aware Process Placement (Components)

THE *Open* GROUP



# QUITE Experiment on Path-aware Process Placement (Application)

THE *Open* GROUP



1) Stable operation

2) Disruptive event—due to unknown and unexpected communications traffic

3) Move process to another node

4) Need to select target node based on application model that incorporates comm usage

includes prediction of move of communications load from current to target node

# AWACS Radar Tracking

THE *Open* GROUP

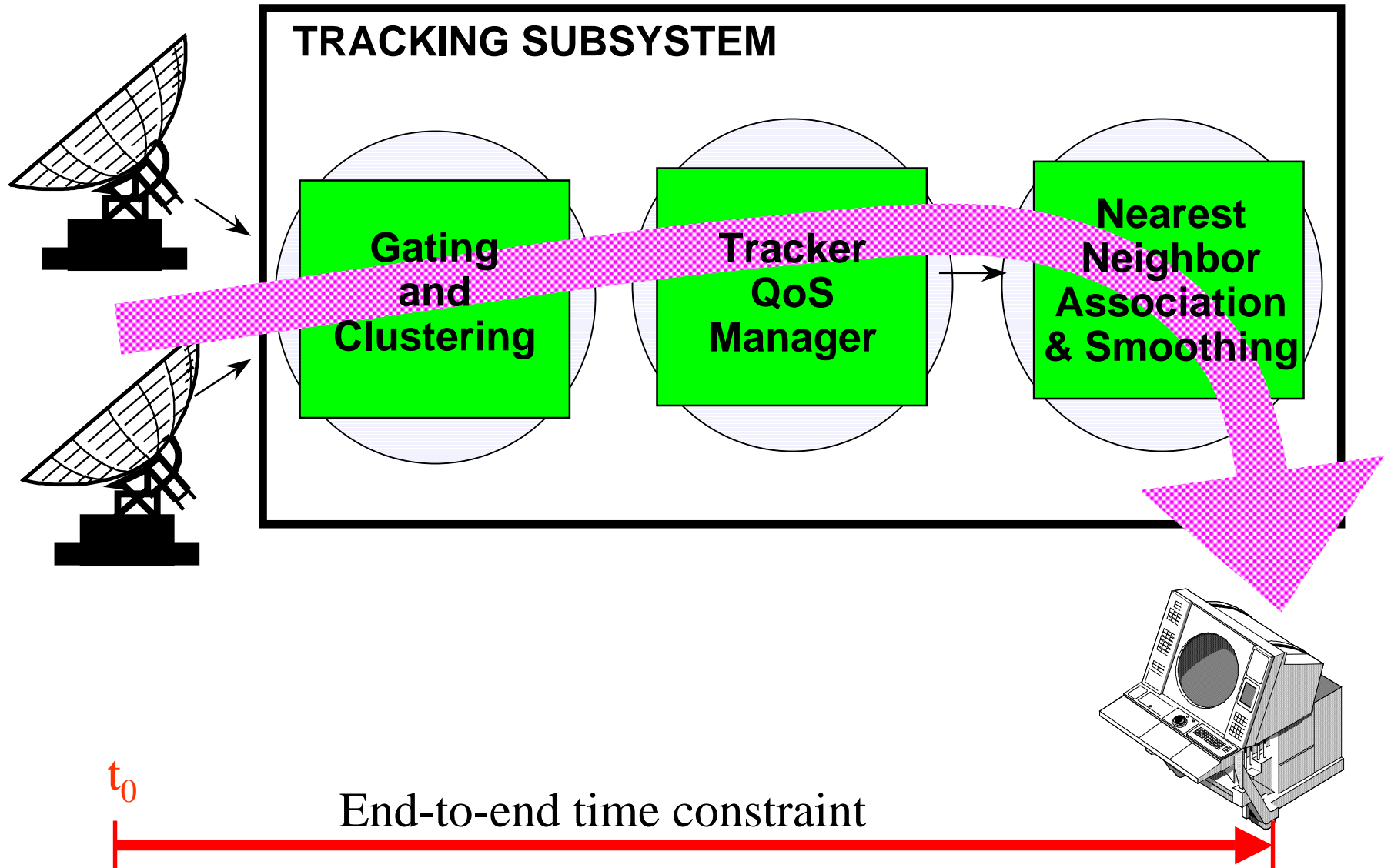
- ❑ Advanced Warning and Control System is
  - An airborne radar system
  - For employment of tactical airpower
- ❑ AWACS surveillance missions
  - Generate aircraft tracks for Command and Control
- ❑ Too many sensor reports can overload the system
  - Causing sectors of the sky to “go blank”

---

**ALLOCATE RESOURCES  
to the RIGHT TRACKS  
for this MISSION  
at this TIME**

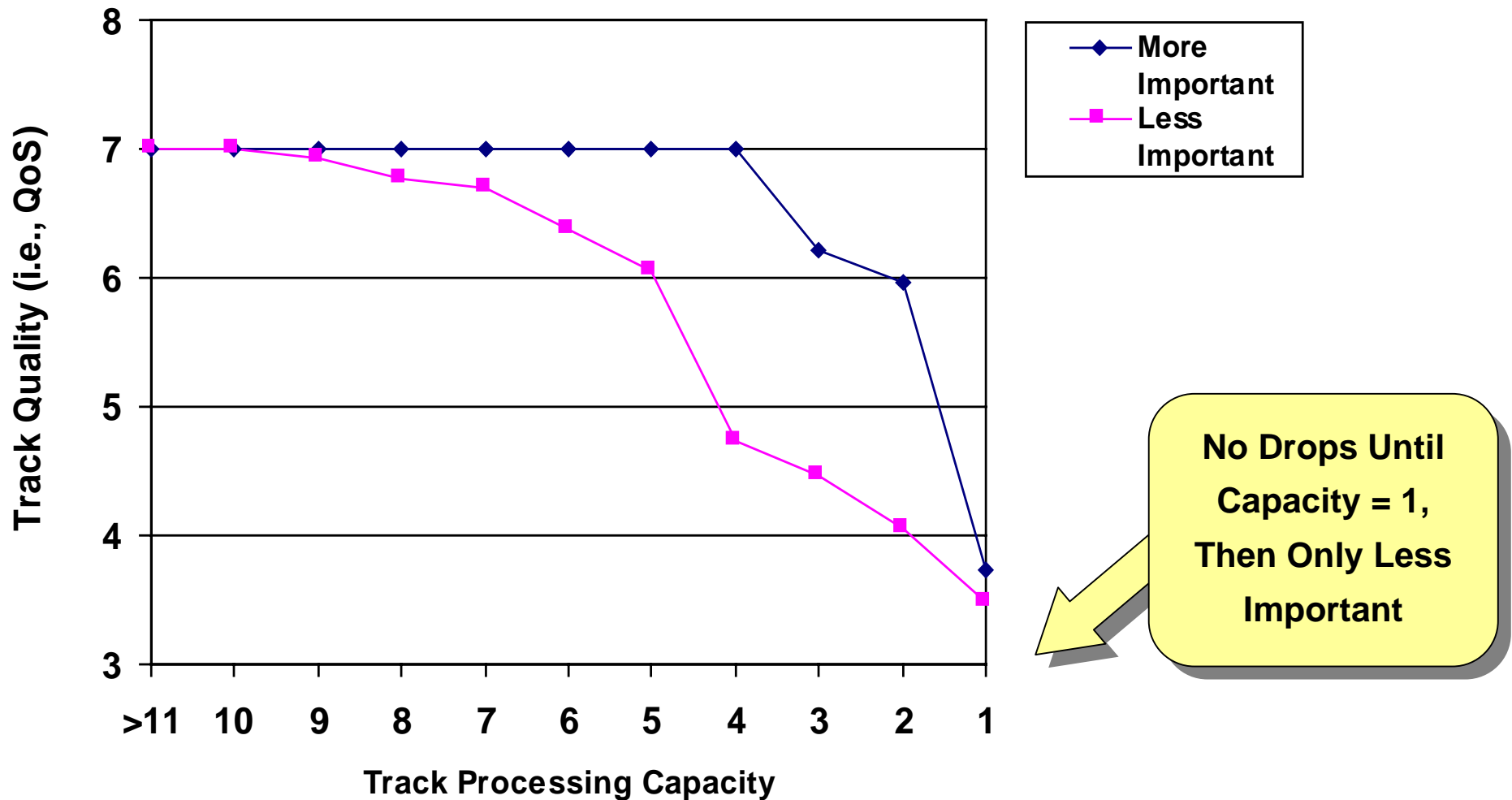
# QoS-Driven Adaptive Tracking

THE *Open* GROUP



# Managed QoS in AWACS Surveillance Mission Tracking

THE *Open* GROUP



# Technical Challenges

THE *Open* GROUP

- ❑ Identify design patterns in QoS controllers
- ❑ Identify design patterns in QoS-aware applications
- ❑ Identify common communication patterns between components and applications
- ❑ Incorporation of existing and evolving standards and practice
- ❑ Foster creation and enhancement of QoS standards and identify practical certification requirements