

Architecture and Mechanisms of Etherware for Cyber-Physical Systems

Kyoung-Dae Kim and P. R. Kumar

Electrical and Computer Engineering and
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

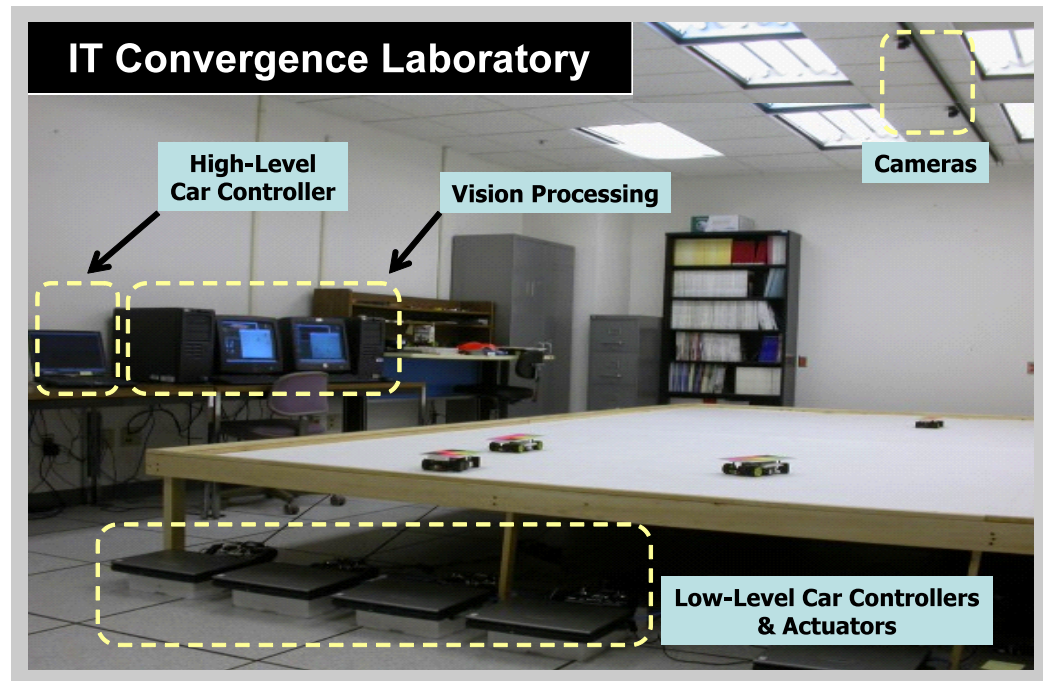
April 11, 2011

Outline

- Networked Control Systems
- Etherware : A Middleware for Networked Control Systems
- Real-Time Enhancement of the Etherware
- A Networked Inverted Pendulum Control System

Cyber-Physical System

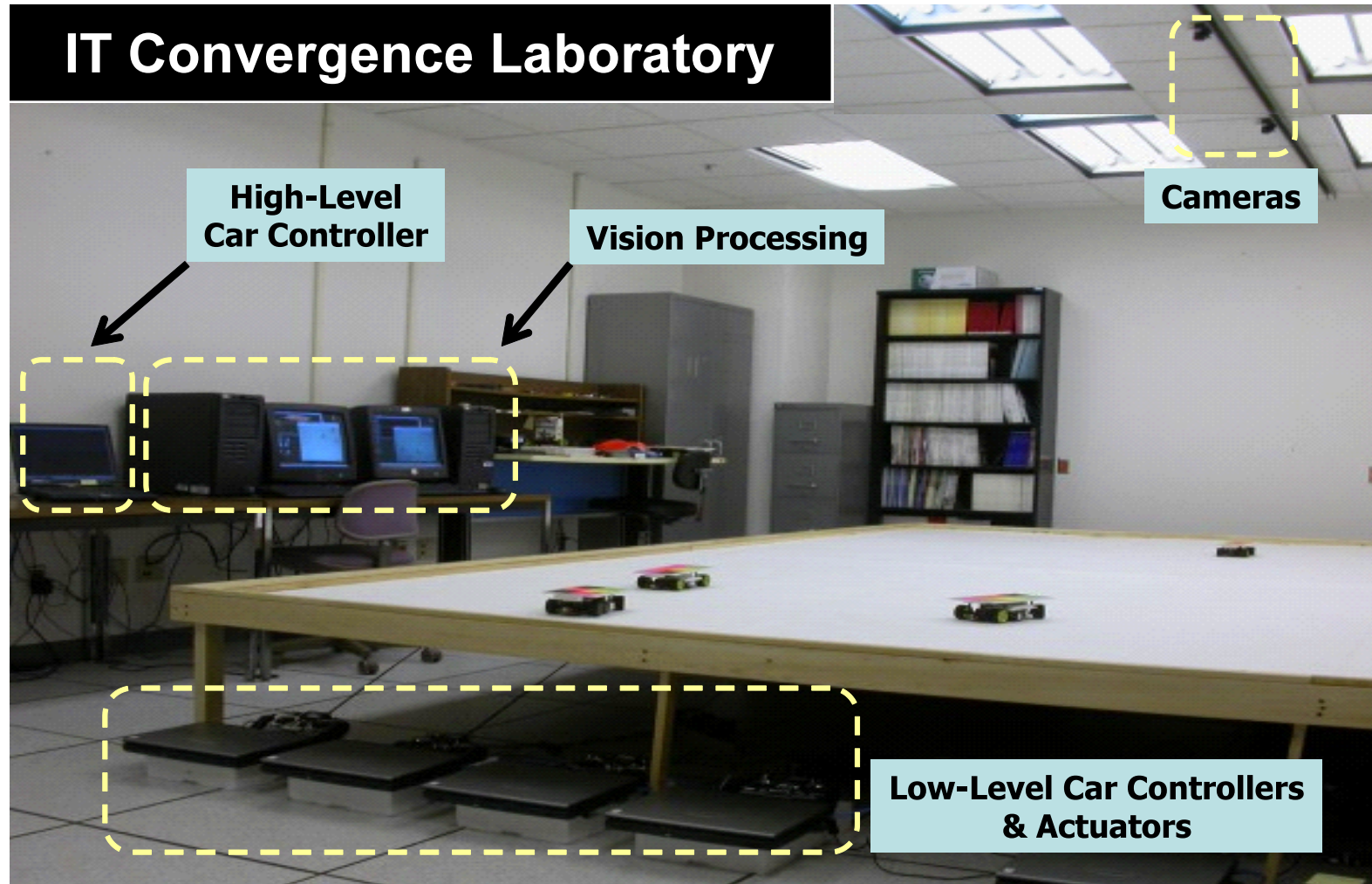
- Cyber-Physical Systems
 - Systems with computing, communication, and physical entities
 - Networked Control System (NCS)
- Characteristics of NCS
 - Large scale
 - Openness
 - Time-critical
 - Safety-critical



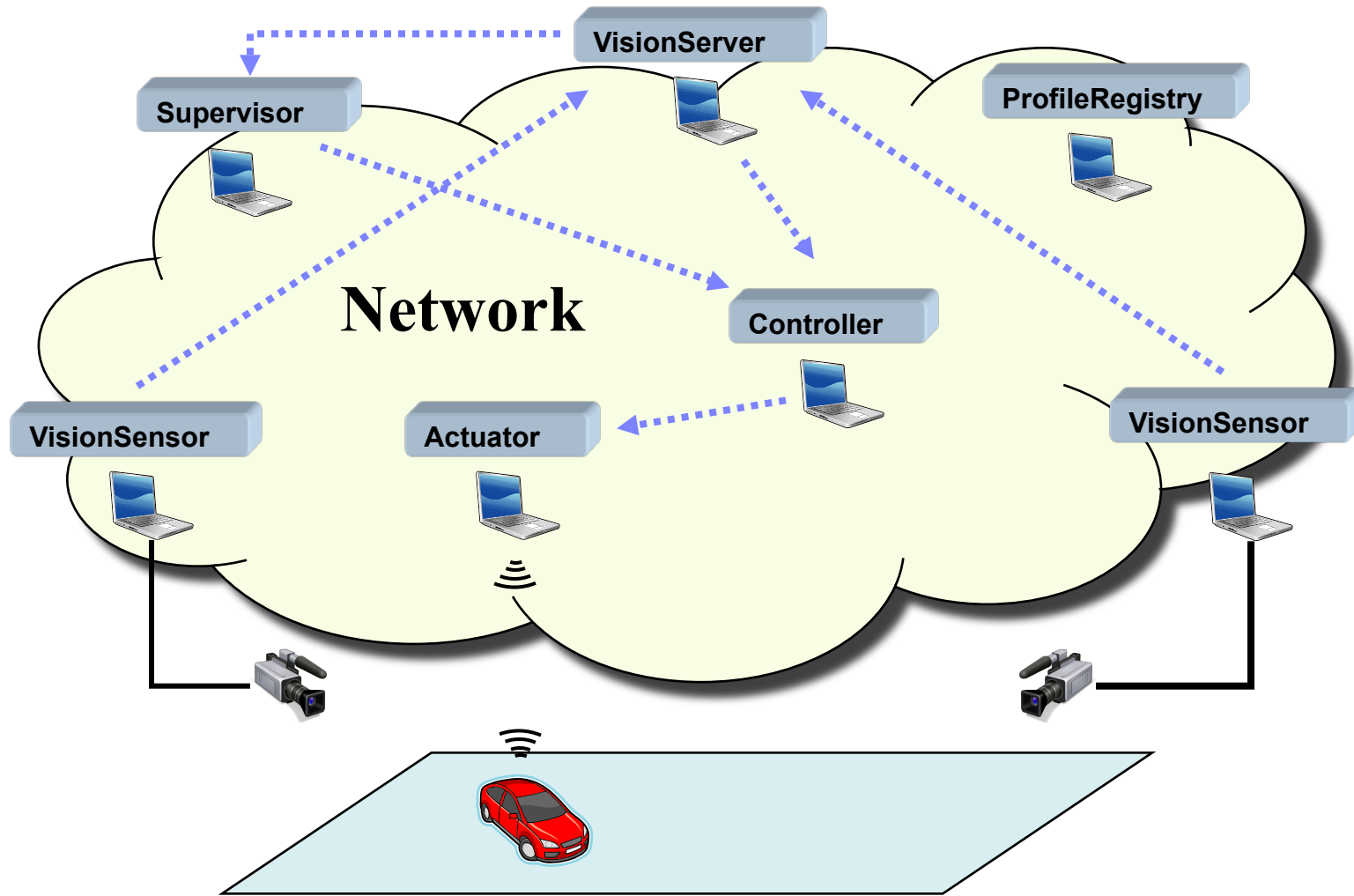
Challenges for NCS Development

- Challenges for NCS Development
 - Platform heterogeneity
 - Clock offset and skew difference
 - Communication delay and packet loss
 - Support for continuous system evolution
 - ...
- It is hard to develop a NCS application
- Need for a well-designed *Middleware* for rapid, reliable, and evolvable NCS application development

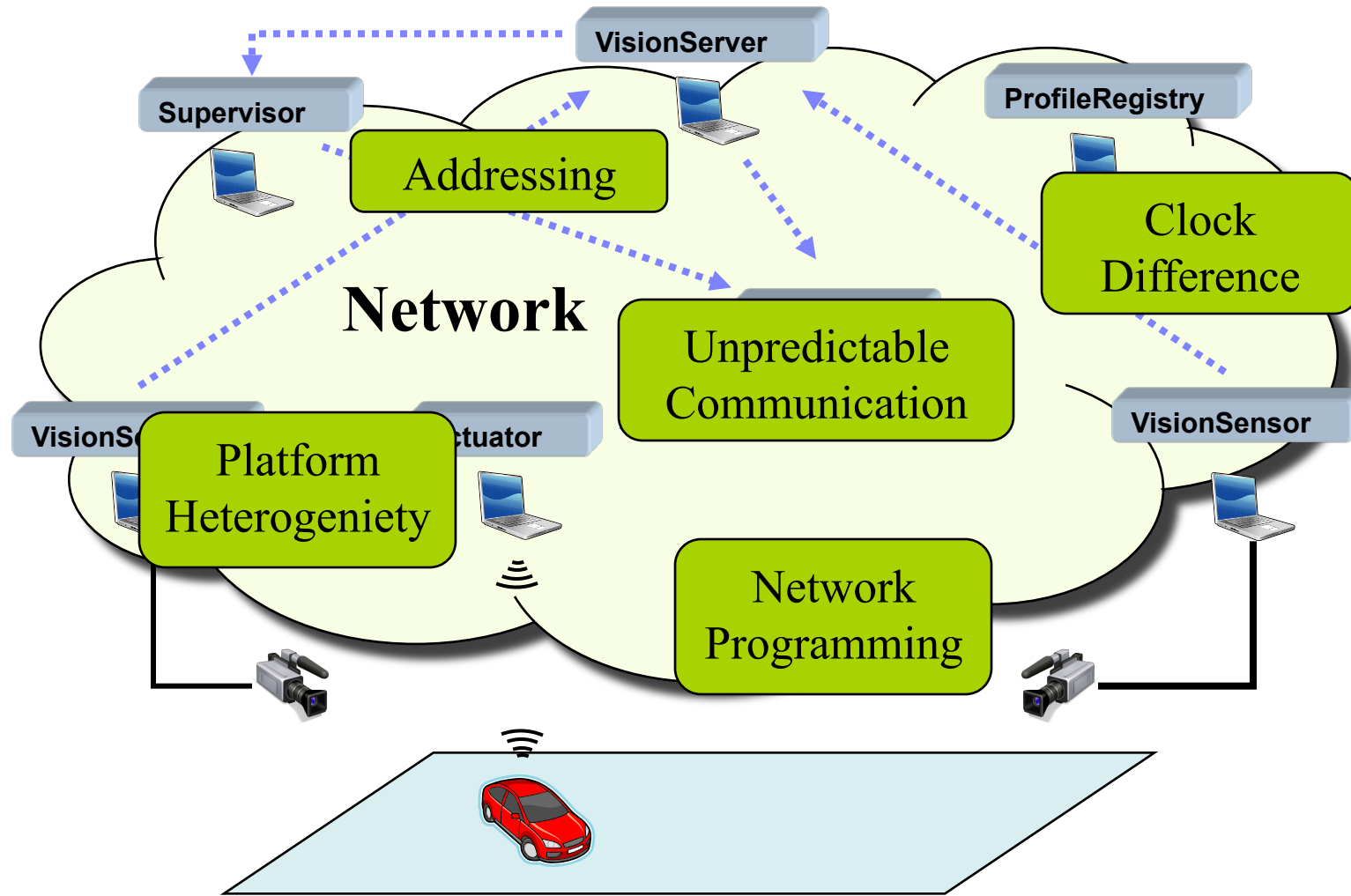
Abstraction: Virtual Collocation



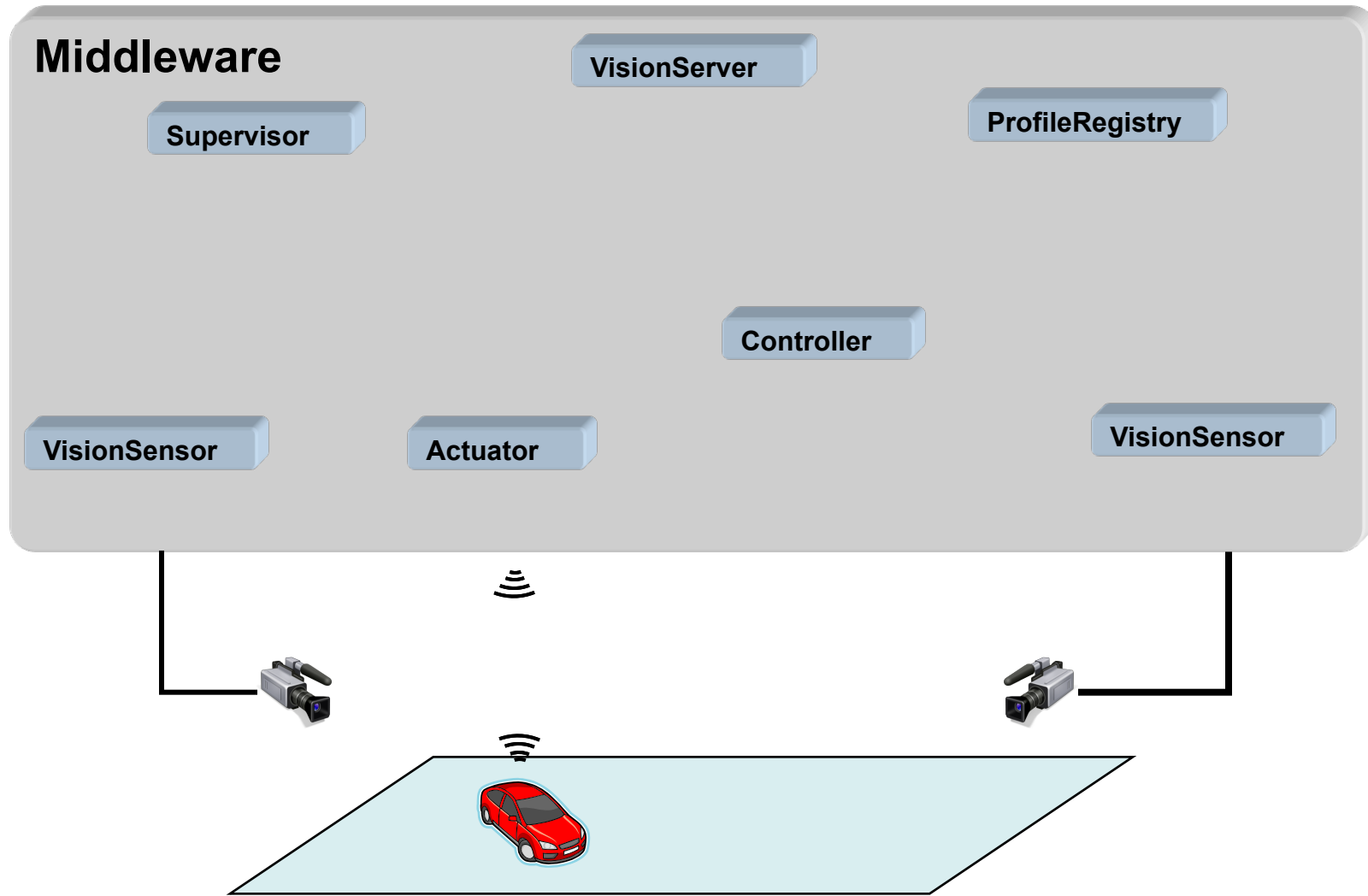
Abstraction: Virtual Collocation



Abstraction: Virtual Collocation



Abstraction: Virtual Collocation

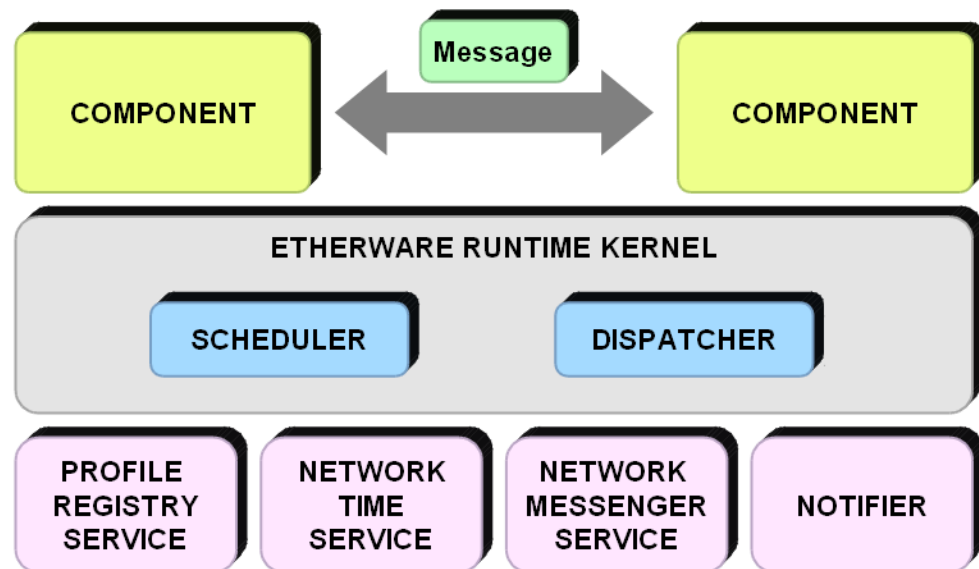


Outline

- Networked Control Systems
- **Etherware : A Middleware for Networked Control Systems**
- Real-Time Enhancement of the Etherware
- A Networked Inverted Pendulum Control System

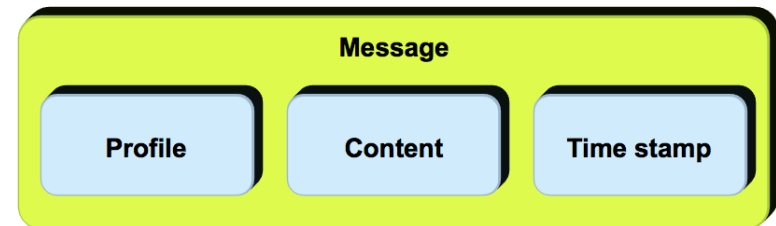
Etherware Architecture

- Etherware
 - A Middleware for NCS developed at the University of Illinois [Baliga'05, Kim'10]
 - For rapid implementation of a reliable and evolvable NCS application
- Microkernel Architecture
 - Etherware Kernel
 - Components
- Component-based Application Development
- Message oriented Communication

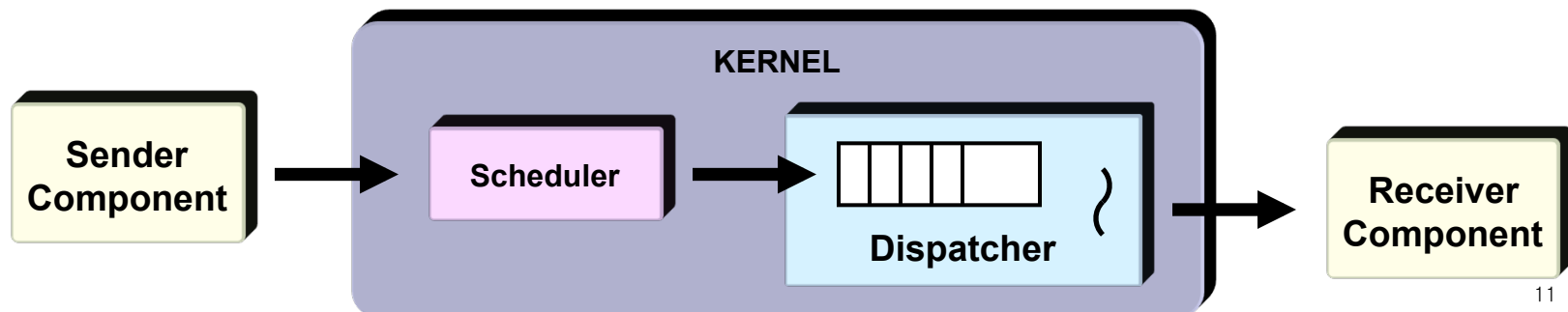


Message Delivery

- Message Class
 - An XML document
 - Profile: Name of the Message Receiver
 - E.g., controller for car 1
 - Content: Interaction semantics
 - Time Stamp: Time when Message is created

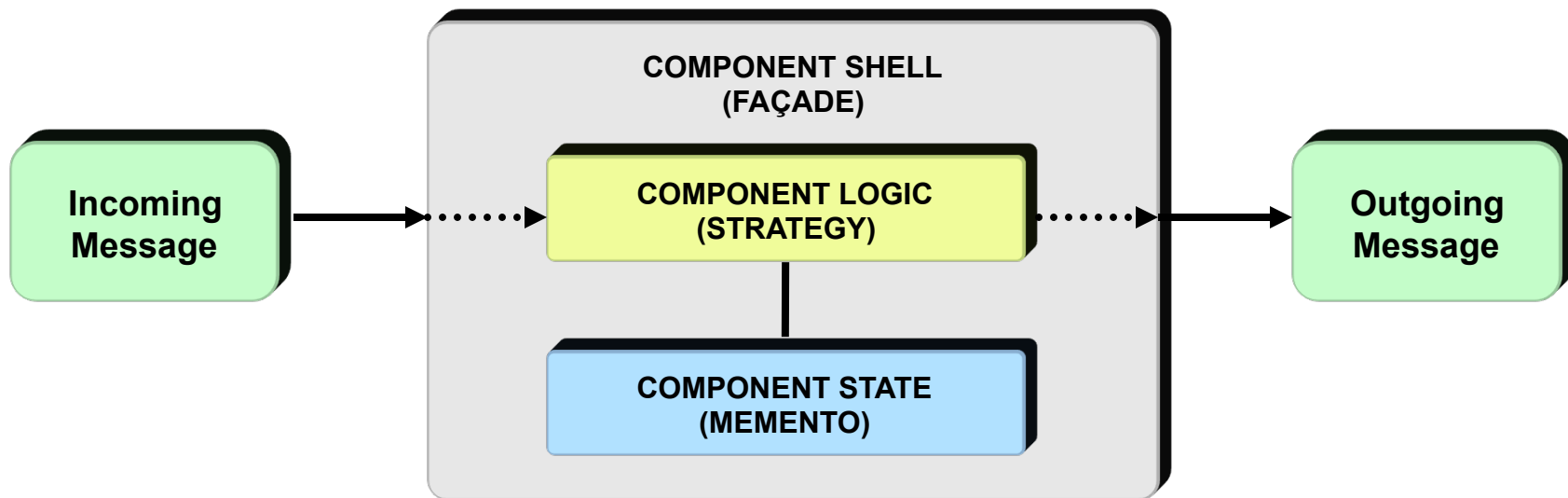


- Etherware Kernel
 - Deliverable address lookup from Profile
 - Receiver Component gets executed when it receives a Message
 - ➔ Event-driven system



Component Model

- Software Design Patterns
 - Façade, Strategy, Memento
- Provide Flexibility
 - Runtime replacement of Component Logic
 - Runtime externalization of Component State

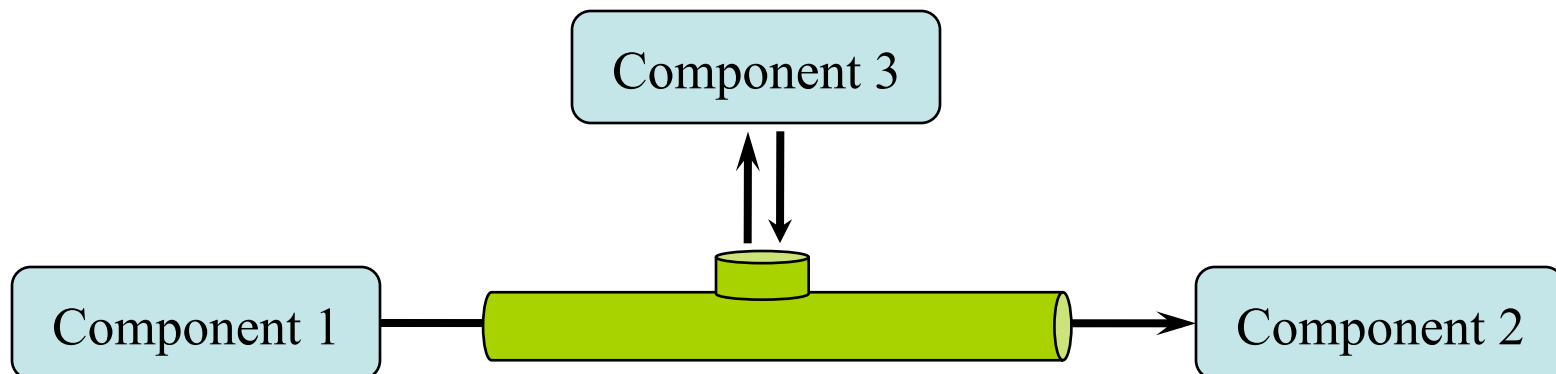


Etherware Services

- ProfileRegistry
 - Map a profile (semantic name) to a deliverable address
- NetworkMessenger
 - Maintain network connection between Etherware processes
 - Send/receive Message over network
- NetworkTime
 - Estimate time differences between computing nodes
 - Translate timestamp in each remote message from remote time to local time
- Notifier
 - Provide time-driven message, called Notification

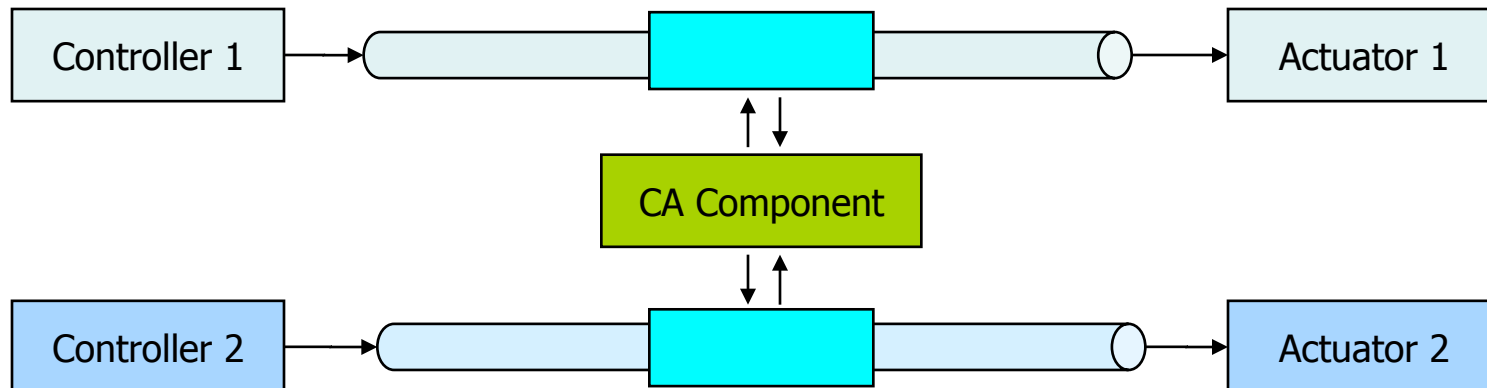
MessageStream and Filter Mechanisms

- MessageStream
 - 1-to-1 communication channel between components
 - Ordered and unreliable communication
 - Useful for control applications
- Filter
 - A mechanism for easy data collection
 - A Tap is created in a MessageStream to intercept messages



MessageStream and Filter Mechanisms

- MessageStream
 - 1-to-1 communication channel between components
 - Ordered and unreliable communication
 - Useful for control applications
- Filter
 - A mechanism for easy data collection
 - A Tap is created in a MessageStream to intercept messages



Outline

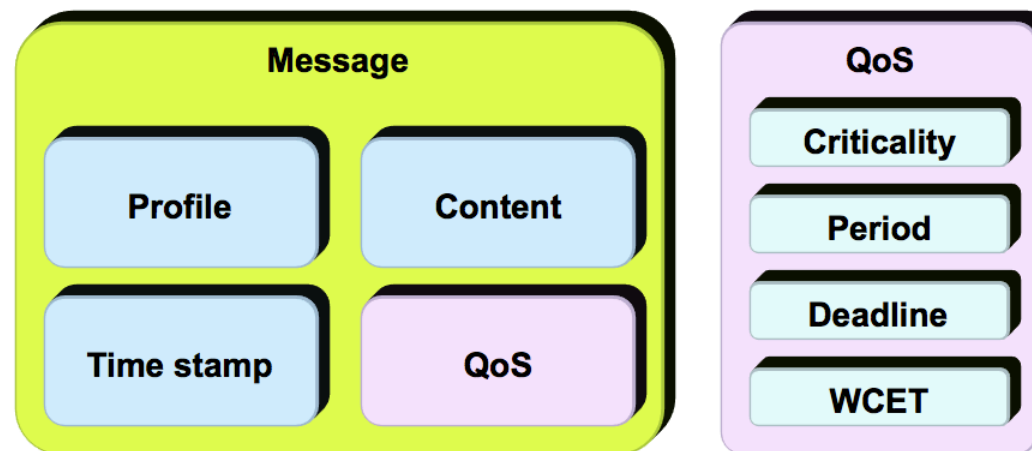
- Networked Control Systems
- Etherware : A Middleware for Networked Control Systems
- **Real-Time Enhancement of Etherware**
- A Networked Inverted Pendulum Control System

Design for Temporal Guarantees

- Message Delivery in Etherware
 - Non-concurrent
 - First In First Out (FIFO) order
- Design Goal
 - Temporal Predictability
 - Flexibility
- Hierarchical Scheduling Mechanism
 - Static classification at first stage
 - Dynamic ordering at second stage

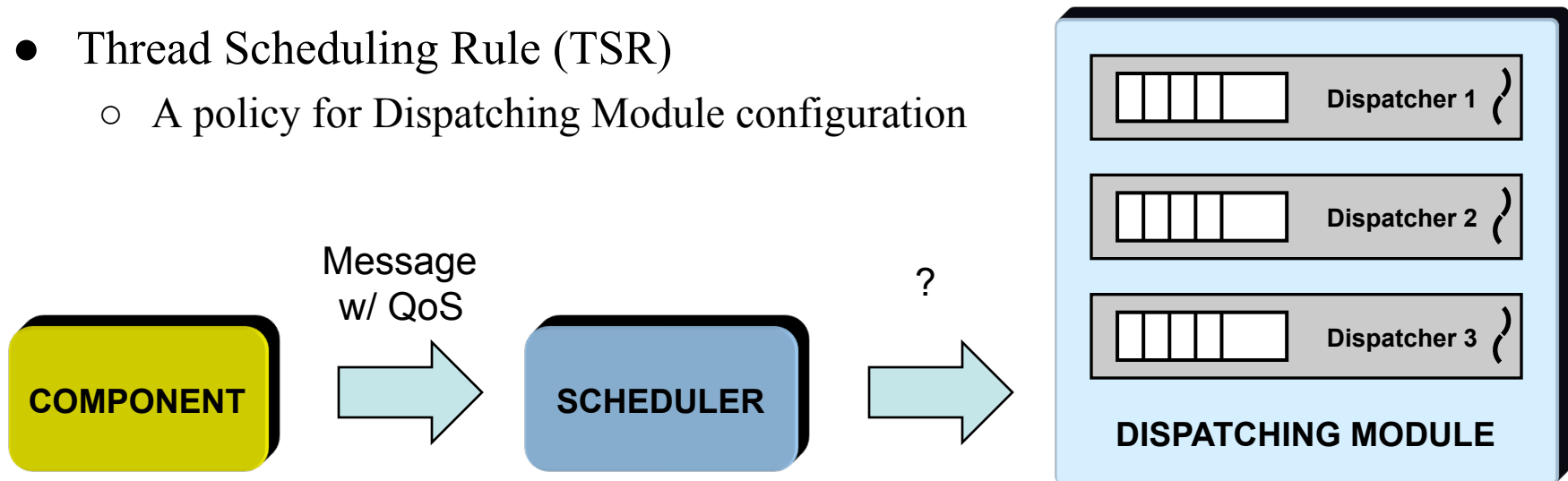
Quality of Service (QoS) of Message Delivery

- Quality of Service (QoS)
 - A collection of attribute that is used in scheduling for Message delivery
 - Period, relative deadline, absolute deadline, criticality of a Message
- QoS Specification
 - QoS XML element in Message class
 - Message is scheduled based on the QoS specification and a *scheduling policy*



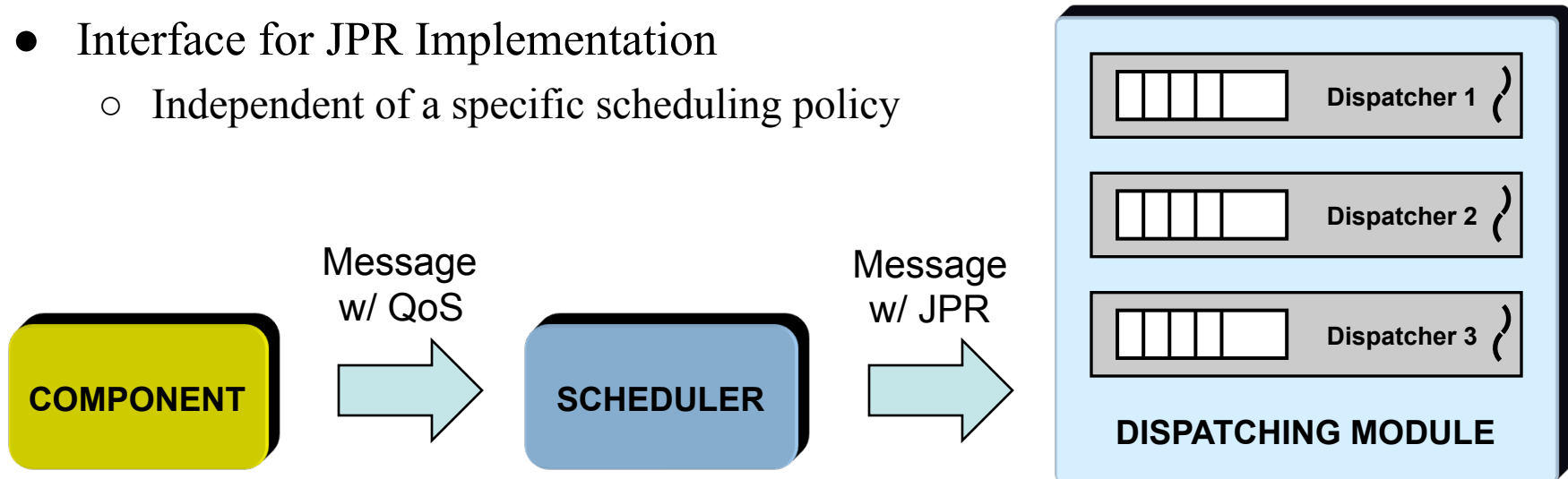
Preemptive Concurrent Message Delivery

- Concurrency
 - Dispatching Module: A set of Dispatchers
- Preemption
 - Assign a fixed *priority* to each Dispatcher
 - Each Dispatcher has a prioritized job queue
- Thread Scheduling Rule (TSR)
 - A policy for Dispatching Module configuration

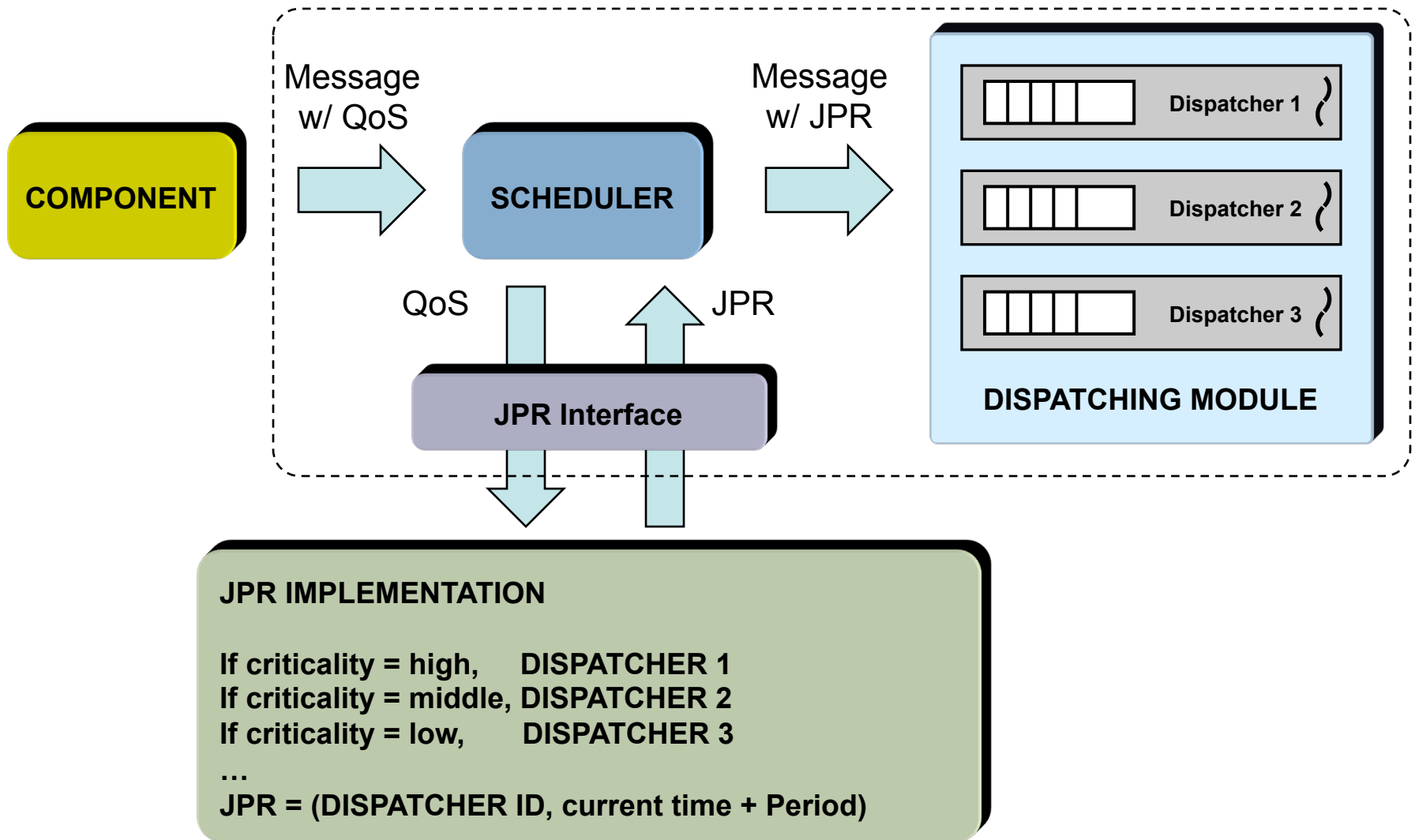


Preemptive Concurrent Message Delivery (continued)

- Job Placement Rule (JPR)
 - A pair of attribute for linear ordering within Dispatching Module
- Scheduling Policy
 - Map from a QoS specification to a JPR
 - JPR Implementation: An implementation of a specific scheduling policy
- Interface for JPR Implementation
 - Independent of a specific scheduling policy



Hierarchical Scheduling Mechanism

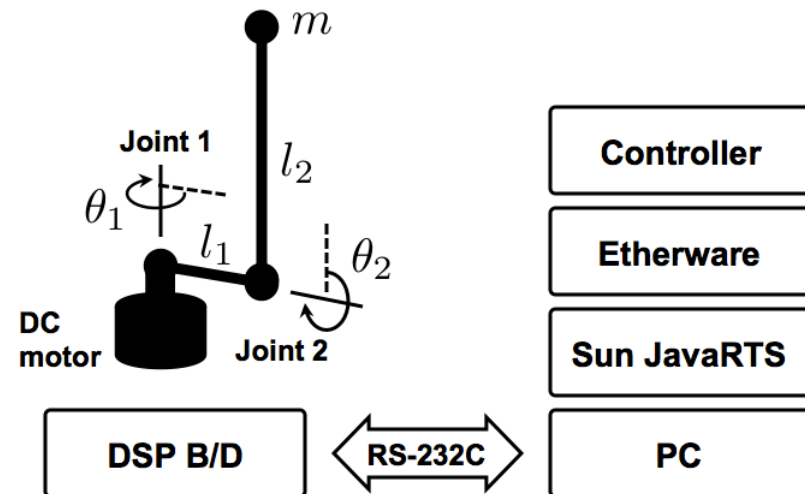
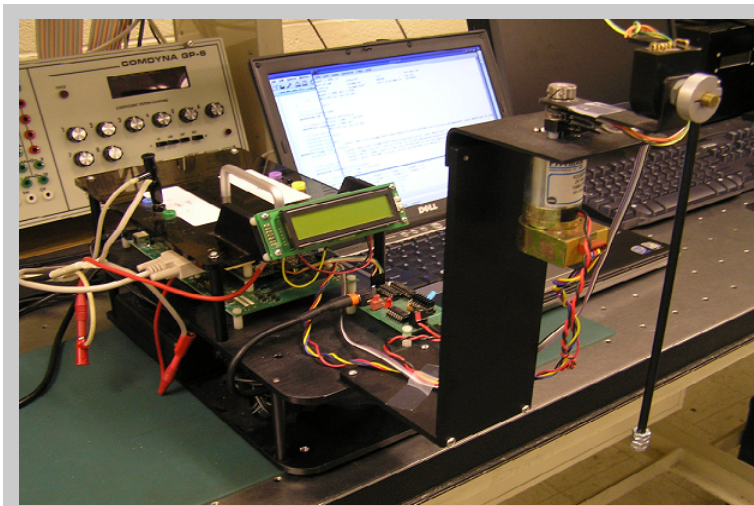


Outline

- Networked Control Systems
- Etherware : A Middleware for Networked Control Systems
- Real-Time Enhancement of the Etherware
- **A Networked Inverted Pendulum Control System**

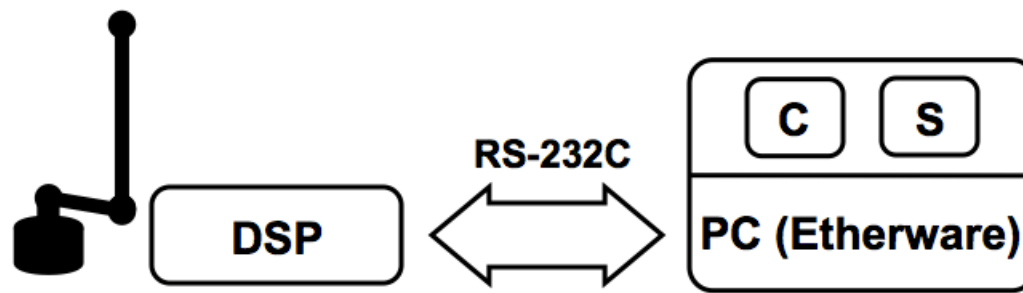
An Inverted Pendulum Control System

- Implementation Platform
 - Sun Java Real-Time System 2.0 with Solaris 10
- System Configuration
 - DSP Program
 - Return encoder values upon request from Controller
 - Deliver control command upon receive it from Controller



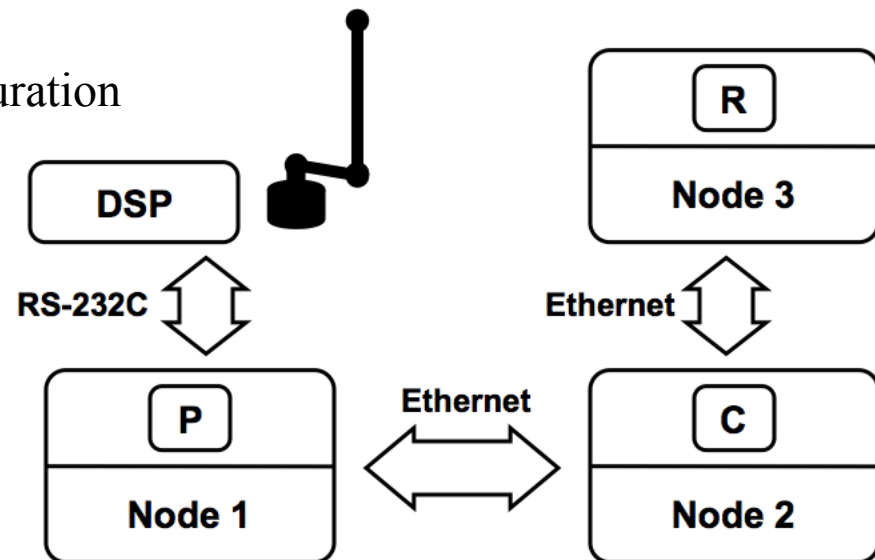
Periodic Control under Stress

- Periodic Controller Component
 - Period: 15ms
 - Execution priority: High
- Stressing Component
 - Period: 1s
 - Avg. Execution time: 100ms
 - Execution priority: Low

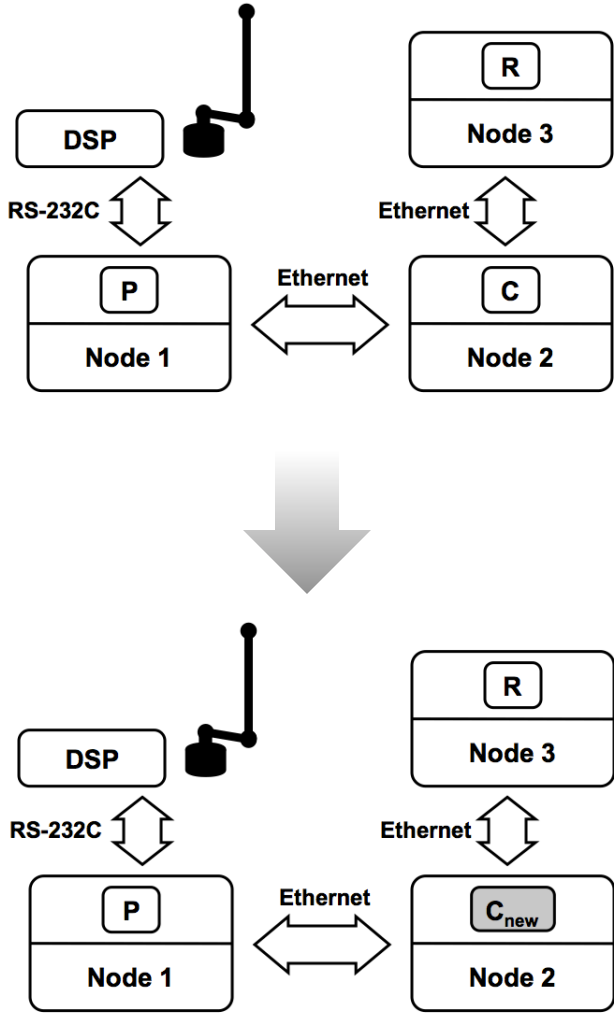


A *Networked* Inverted Pendulum Control System

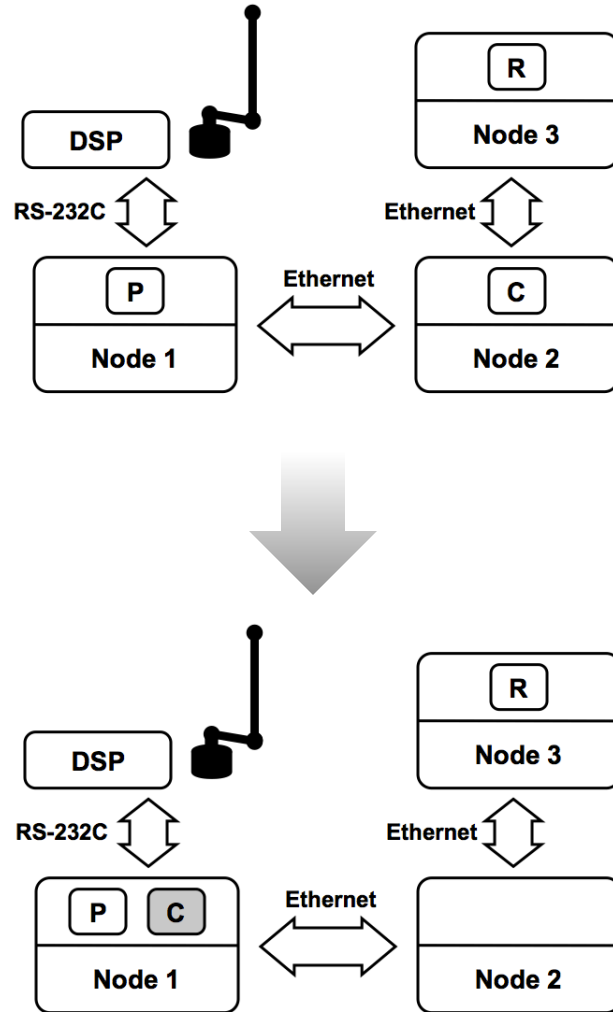
- Periodic Controller Component
 - Period: 15ms
- DSPProxy Component
 - Intermediate interaction between Controller and DSP program
- Request Component
 - Request runtime system reconfiguration
 - Controller upgrade
 - Controller migration



Controller Upgrade



Controller Migration



Thank You !