Software Deployment Architecture and Quality-of-Service in Pervasive Environments

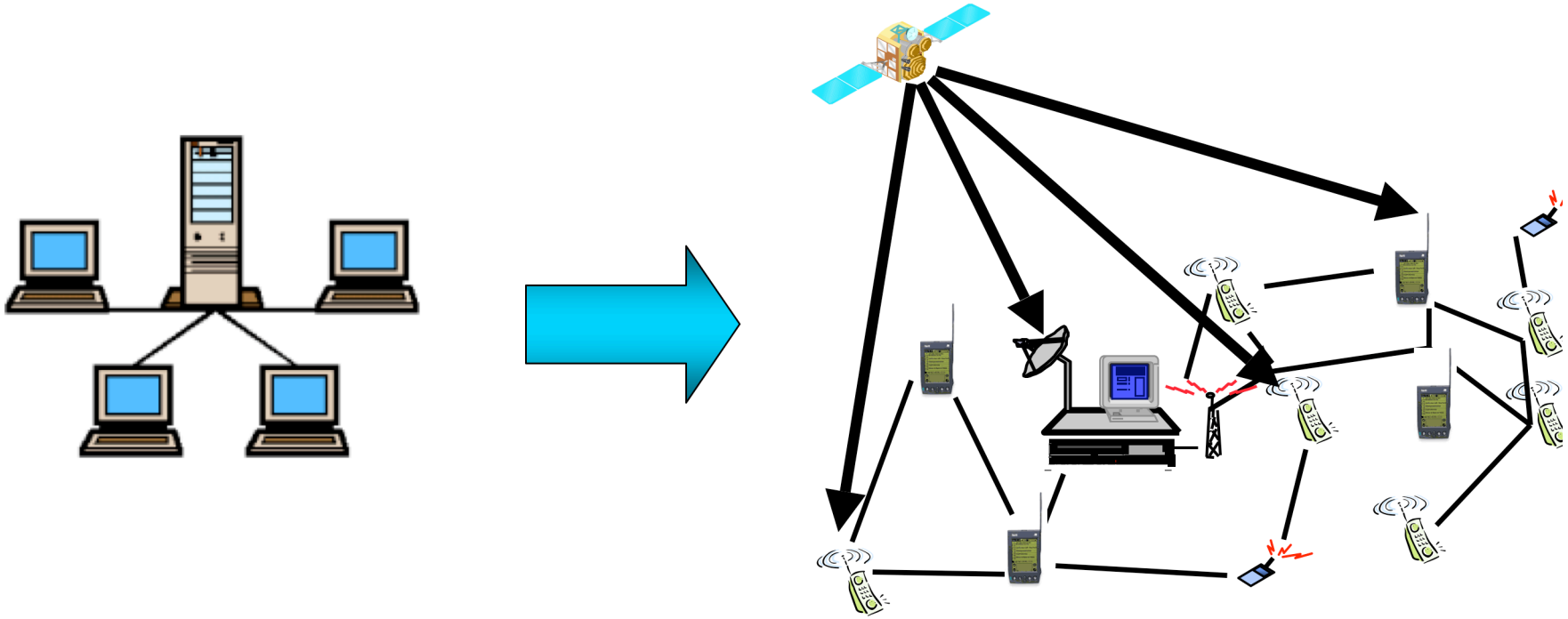
Presenter: Yuriy Brun
Computer Science Department
University of Southern California

Nenad Medvidovic
Computer Science Department
University of Southern California

Sam Malek
Department of Computer Science
George Mason University

Software Engineering Trends

- "Embedding" of Software
  - A wider spectrum of distribution and heterogeneity
  - Hardware and software mobility is becoming the norm
  - Software systems are becoming more complex



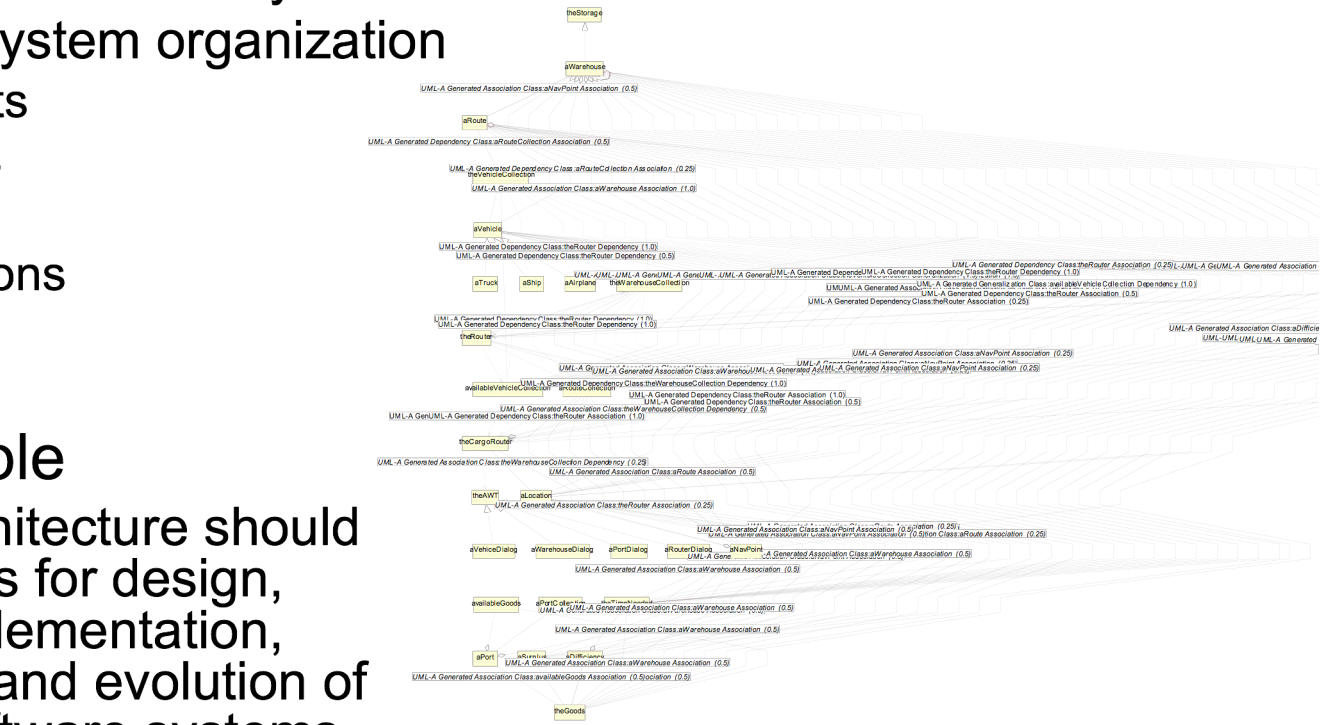
Software Architecture

- Software Architecture
  - A high-level model of a system
  - Represents system organization
    - Components
    - Connectors
    - Events
    - Configurations

- Guiding principle
  - Software architecture should form the basis for design, analysis, implementation, deployment, and evolution of pervasive software systems

Software Architecture

- Software Architecture
  - A high-level model of a system
  - Represents system organization
    - Components
    - Connectors
    - Events
    - Configurations

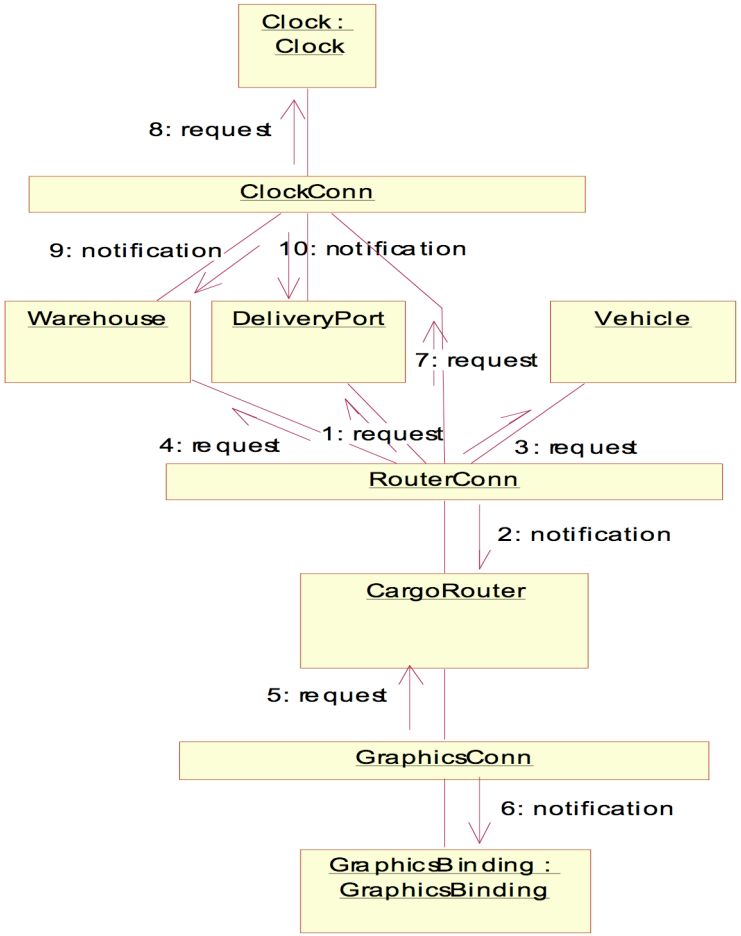


- Guiding principle
  - Software architecture should form the basis for design, analysis, implementation, deployment, and evolution of pervasive software systems

Software Architecture

- Software Architecture
  - A high-level model of a system
  - Represents system organization
    - Components
    - Connectors
    - Events
    - Configurations

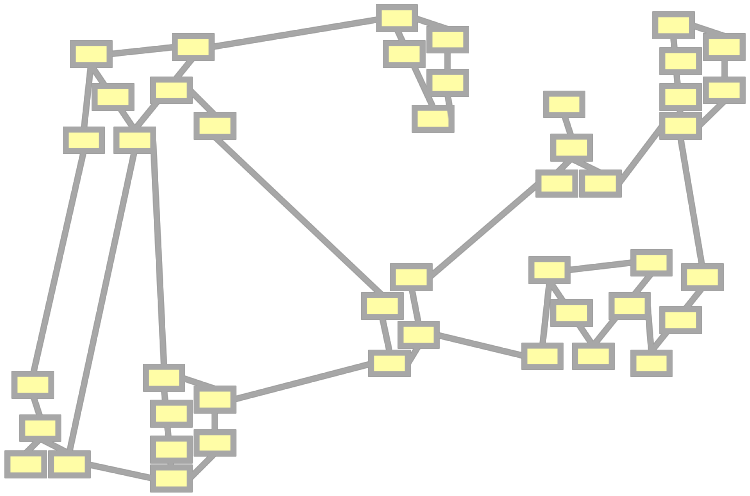
- Guiding principle
  - Software architecture should form the basis for design, analysis, implementation, deployment, and evolution of pervasive software systems



Architectural Decisions

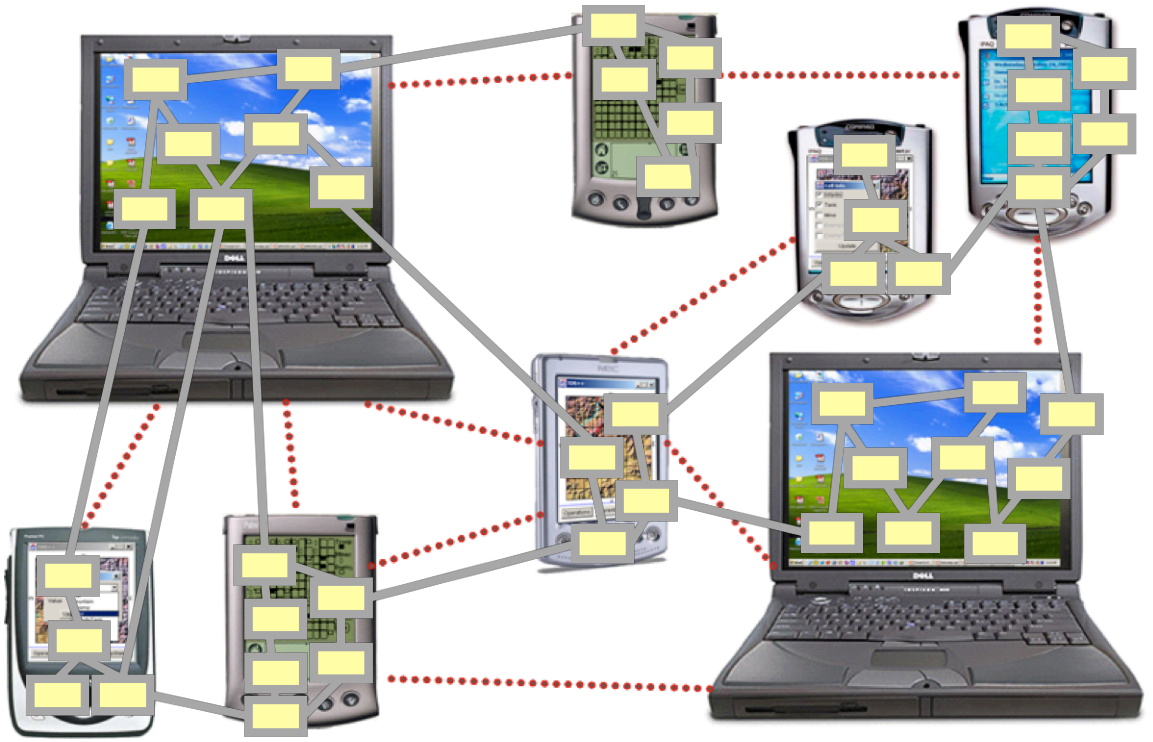
- Architectural decisions impact non-functional properties of the system
- Non-functional property
  - Quality level at which an expected functionality is delivered
  - a.k.a. Quality of Service (QoS)
- Making the architectural decisions has remained an art form
  - Lack of quantification and measurement techniques
  - Reliance on domain expert knowledge

Deployment Architecture Impacts QoS



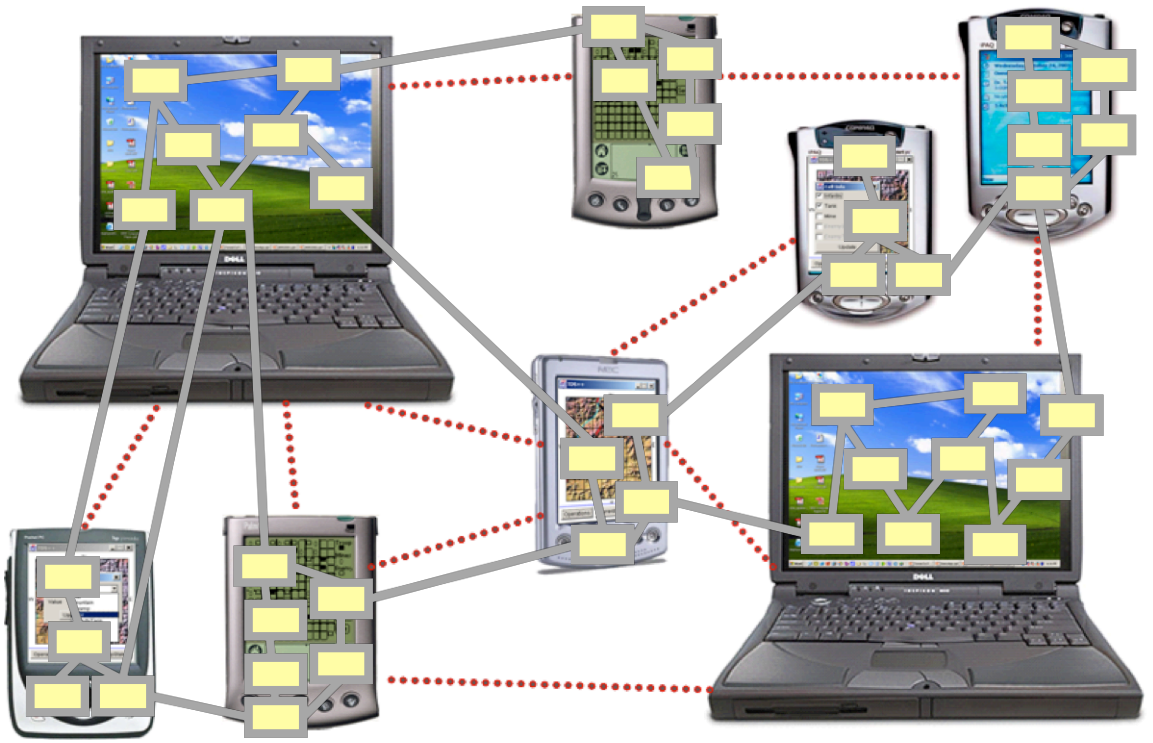
- *Deployment Architecture*: allocation of software components to hardware hosts

Deployment Architecture Impacts QoS



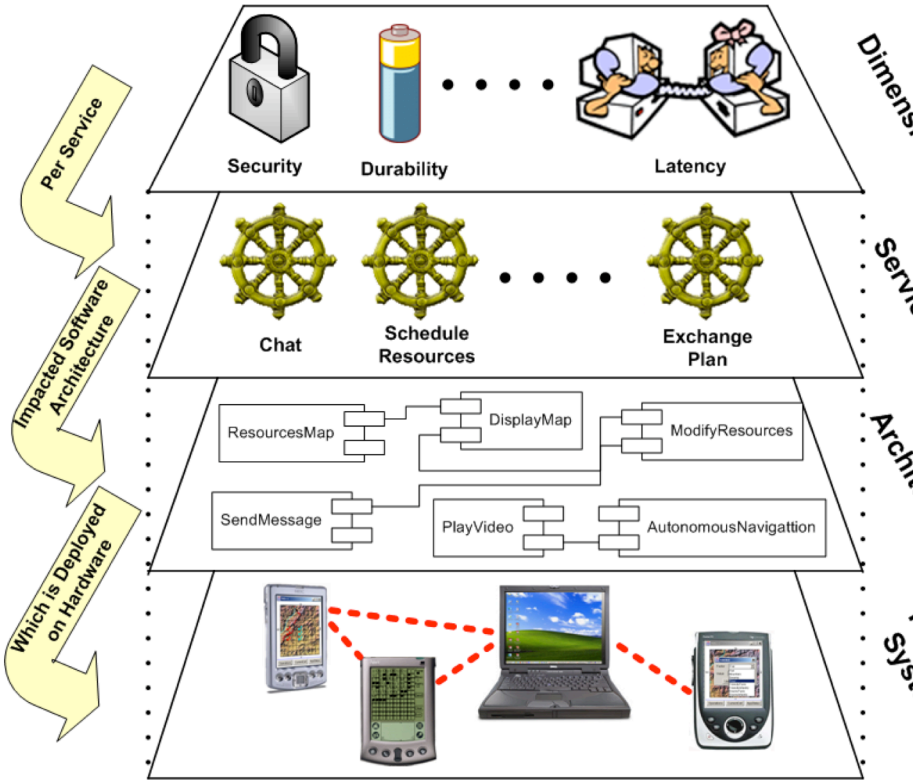
- *Deployment Architecture*: allocation of software components to hardware hosts

Deployment Architecture Impacts QoS



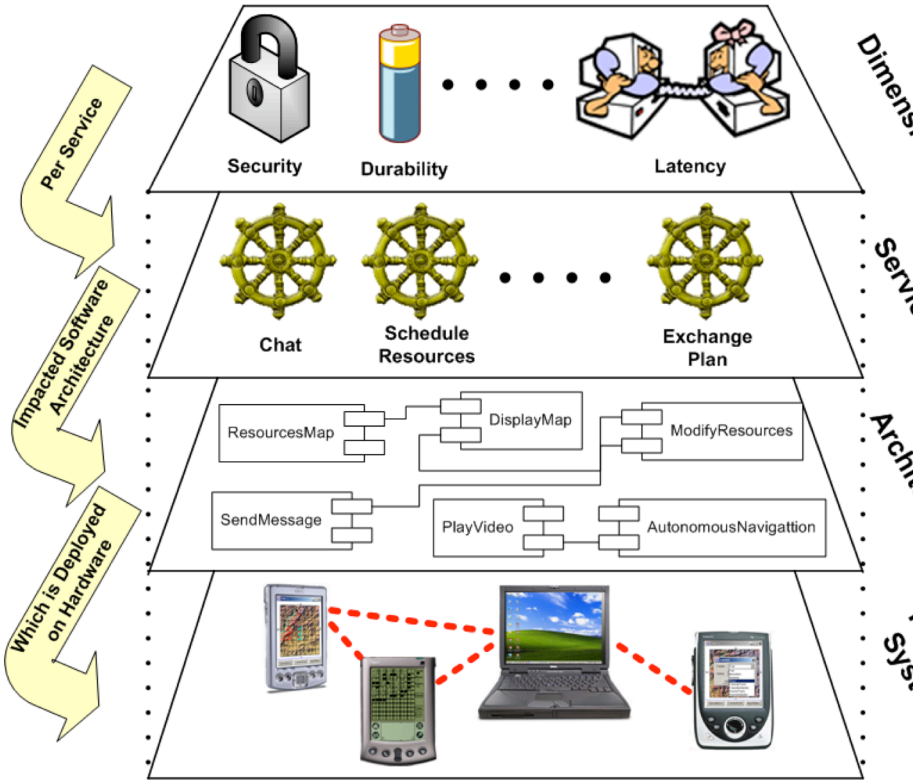
- *Deployment Architecture*: allocation of software components to hardware hosts
- h^c deployment architectures are possible for a given system
  - Many provide the same functionality
  - Provide different qualities of service (QoS)

Problem in a Nutshell



Problem in a Nutshell

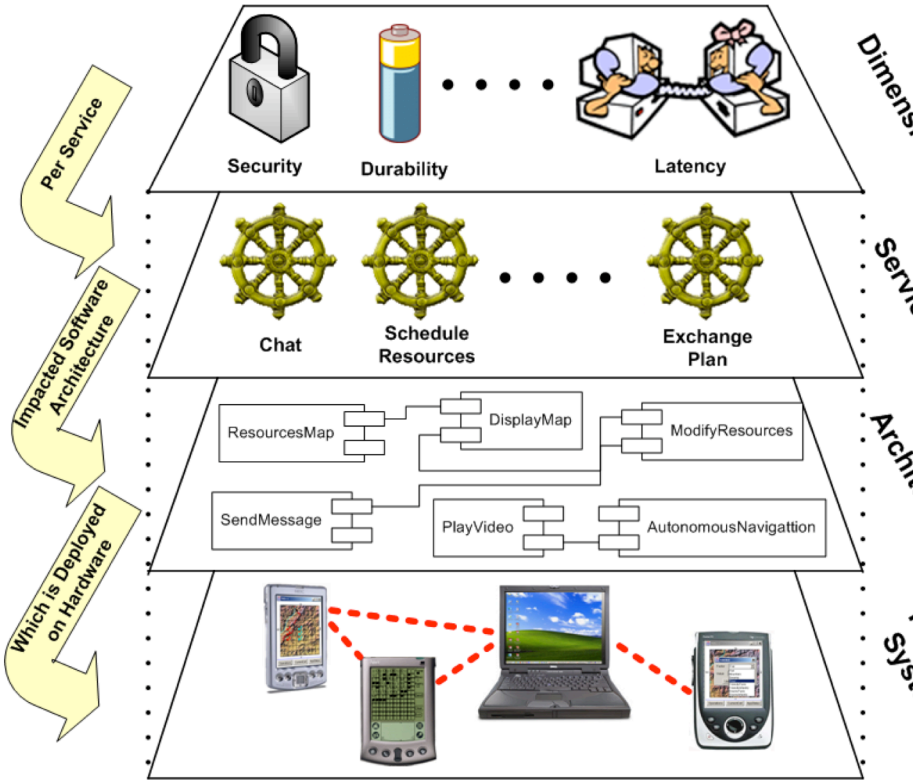
- **Research Question**
  - **How could we find and effect a deployment architecture that improves multiple QoS dimensions?**
    - Where other possible solutions such as caching, hoarding, replication, etc. are not appropriate or ideal



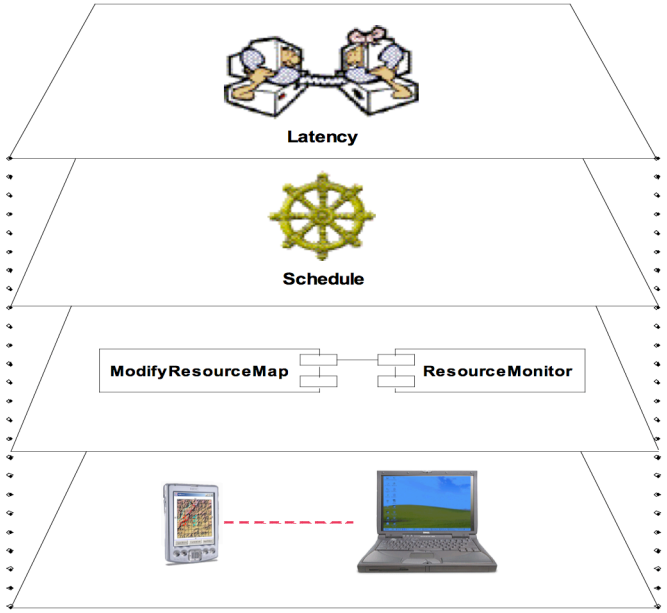
Problem in a Nutshell

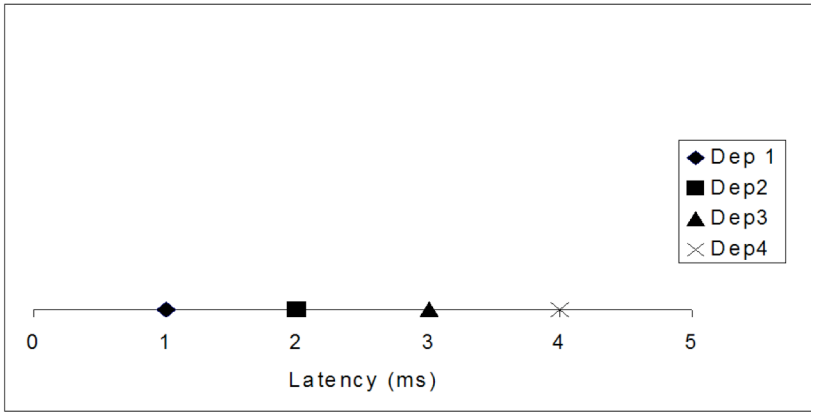
- **Research Question**
  - **How could we find and effect a deployment architecture that improves multiple QoS dimensions?**
    - Where other possible solutions such as caching, hoarding, replication, etc. are not appropriate or ideal

- **Research Objective**
  - **Devise a solution that is applicable to many classes of pervasive systems**
    - No particular definition of QoS dimensions, prescribed models of systems, etc.



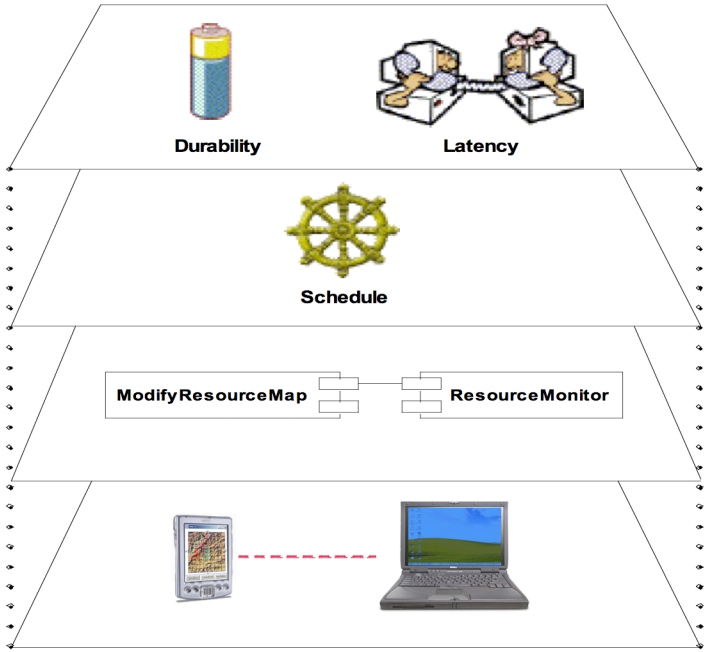
Scenario with a Single QoS Dimension

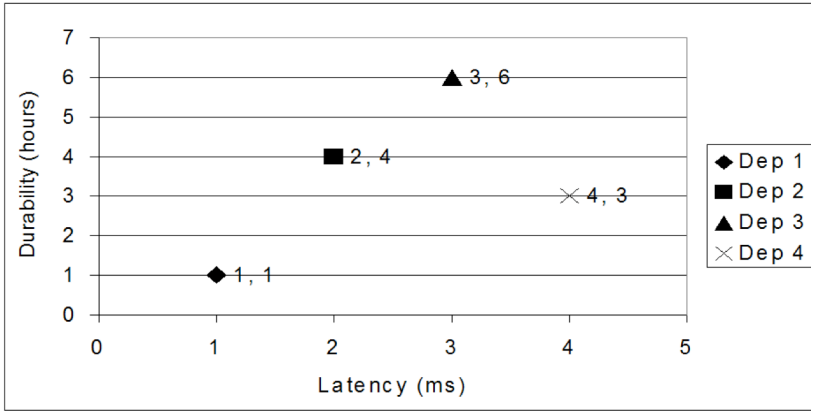




- Objective is to minimize latency
- The *optimal* deployment architecture is deployment 1
- Most all related approaches stop here

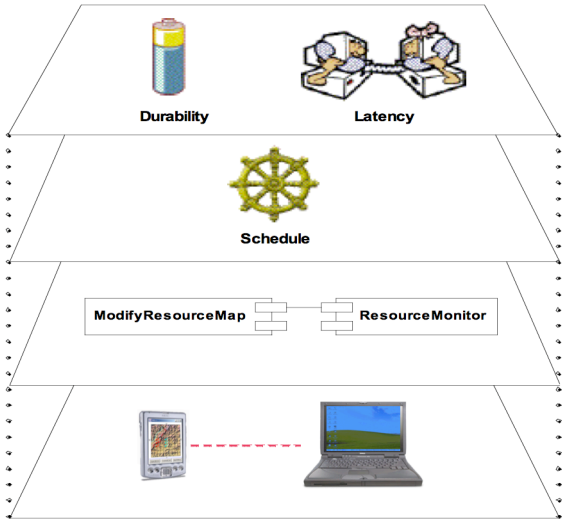
Conflicting QoS Dimensions



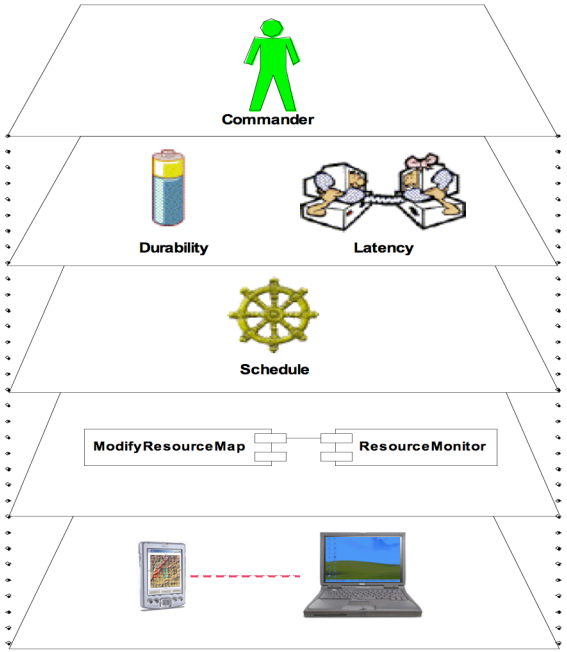


- Objective is to minimize latency and maximize durability
- There is no optimal deployment architecture!
- Frequently encountered phenomenon in multidimensional optimization problems

Resolving Trade-Offs between QoS Dimensions

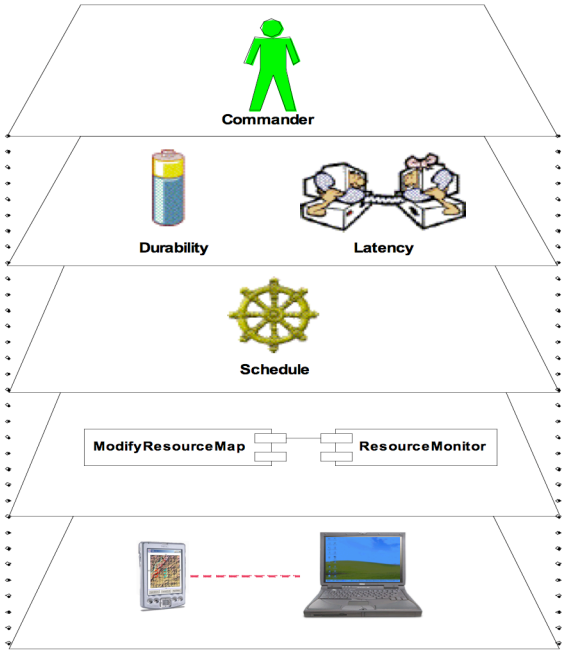


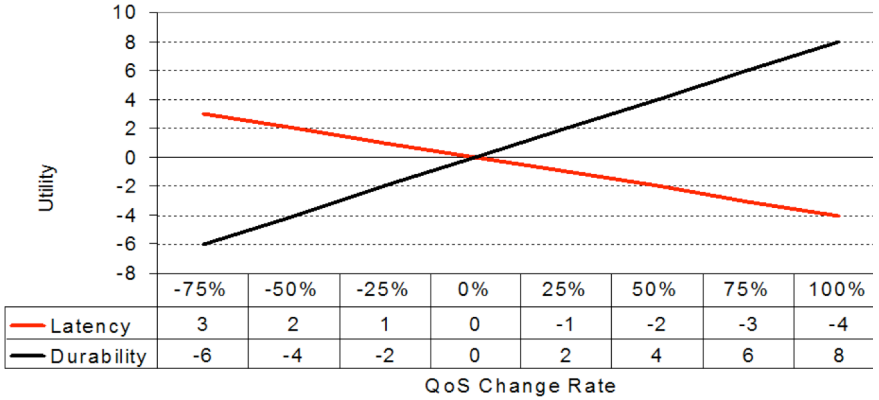
Resolving Trade-Offs between QoS Dimensior



- Guiding Insight
  - System users have varying QoS preferences for the system services they access

Resolving Trade-Offs between QoS Dimension

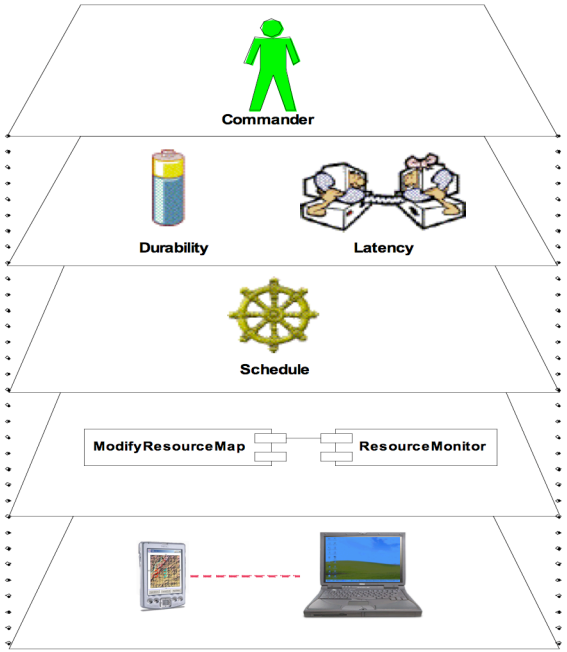


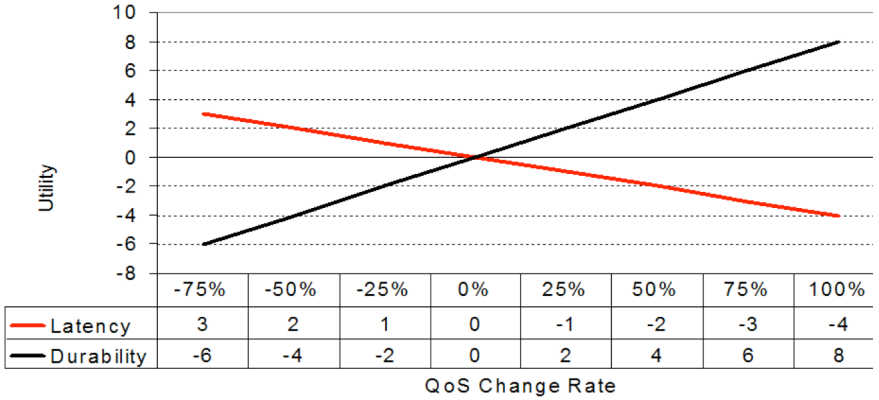


- A *utility function* denotes a user's preferences for a given rate of change in a QoS dimension

- <u>Guiding Insight</u>
  - System users have varying QoS preferences for the system services they access

Resolving Trade-Offs between QoS Dimension

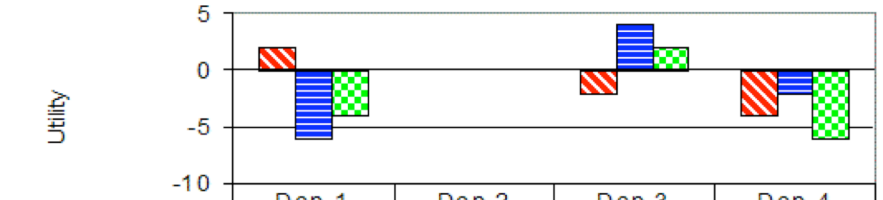




	-75%	-50%	-25%	0%	25%	50%	75%	100%
Latency	3	2	1	0	-1	-2	-3	-4
Durability	-6	-4	-2	0	2	4	6	8

QoS Change Rate

- A *utility function* denotes a user's preferences for a given rate of change in a QoS dimension



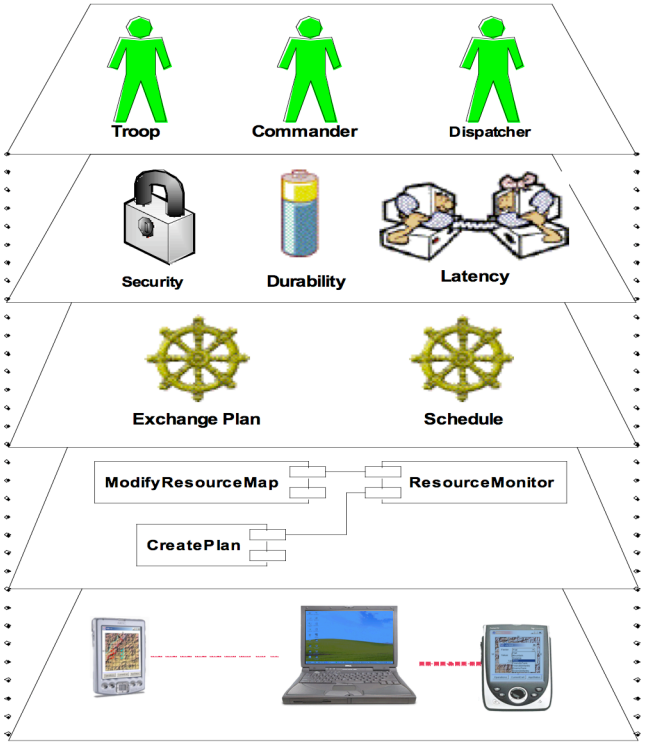
	Dep 1	Dep 2	Dep 3	Dep 4
Latency Utility	2	0	-2	-4
Durability Utility	-6	0	4	-2
Total Utility	-4	0	2	-6

Deployment Architecture

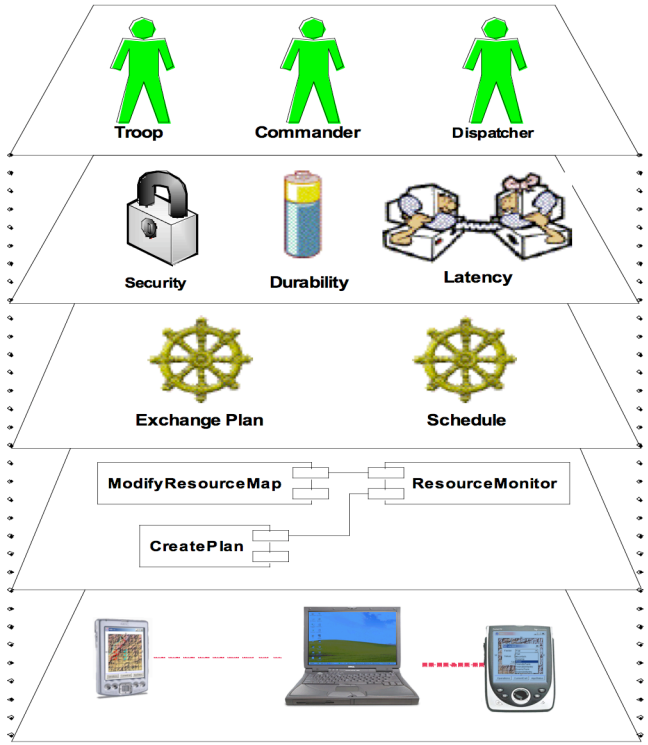
- Allows expression of multidimensiona optimization in terms of a single scala value

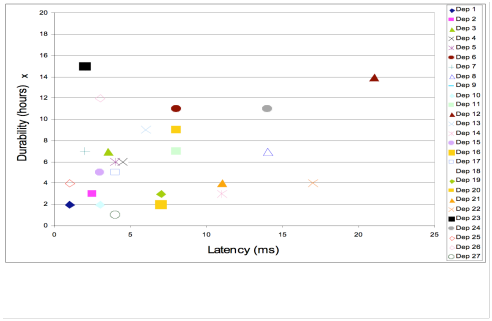
- Guiding Insight
  - System users have varying QoS preferences for the system services they access

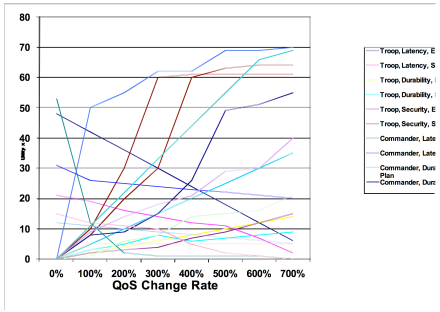
A Slightly Larger Scenario

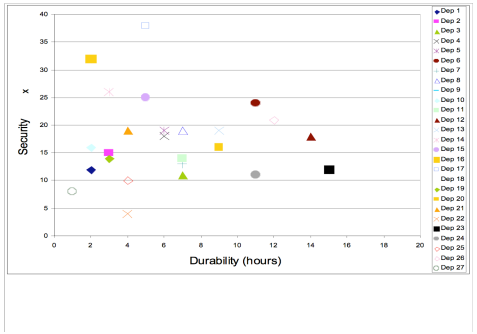


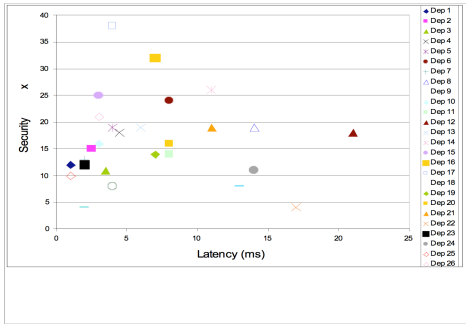
A Slightly Larger Scenario





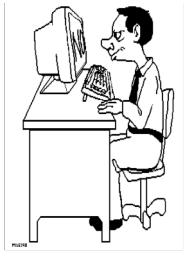






- 18 utility functions would have to be considered across 27 deployments
- **Challenge:** consider many users' preferences for the many QoS dimensions of many services
- "Eyeballing" the solution quickly becomes impossible!

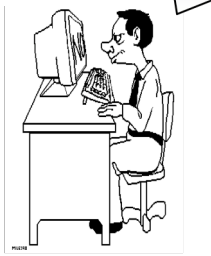
Overall Approach



Software Architect

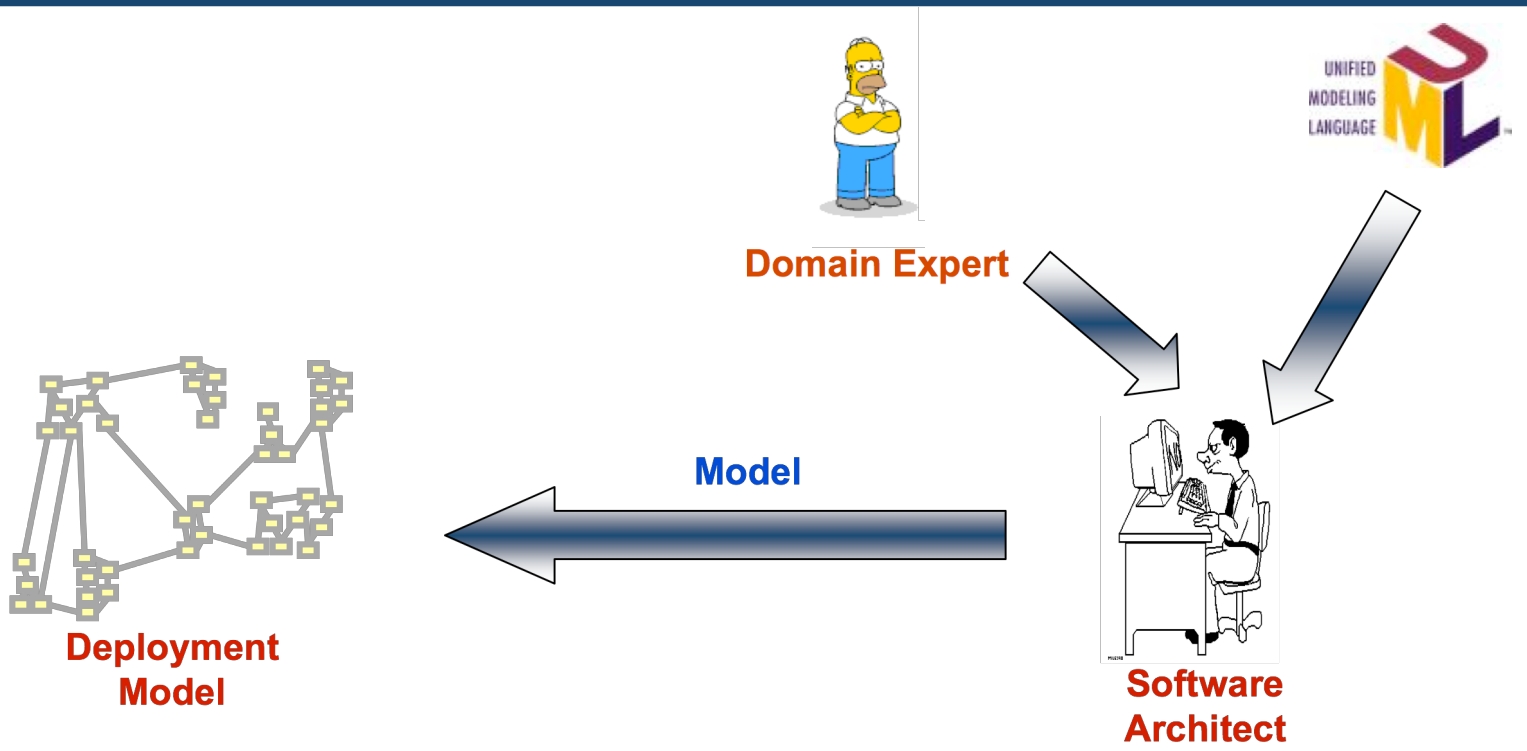
Overall Approach

Domain Expert

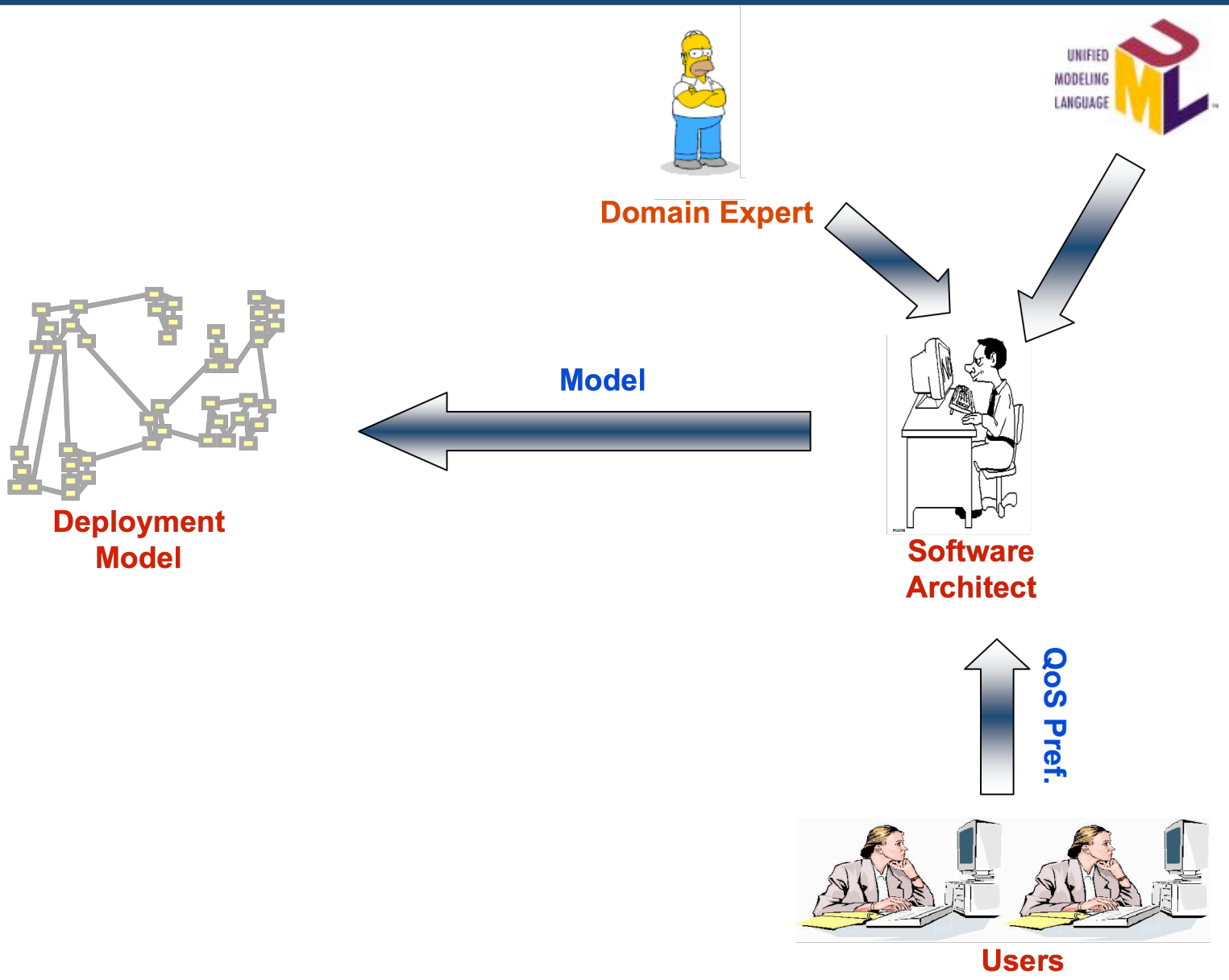


Software Architect

Overall Approach



Overall Approach



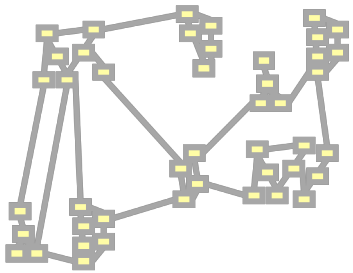
Overall Approach

Domain Expert

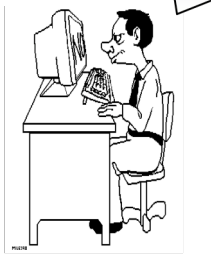
UNIFIED MODELING LANGUAGE

Model

Design-time Analysis



Deployment Model

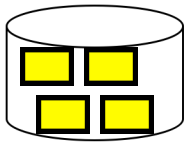


Software Architect

QoS Pref.

Users

Overall Approach

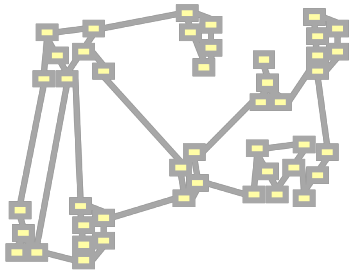


Component Repository

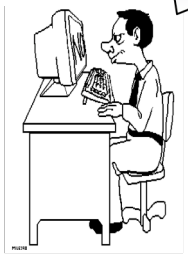
Domain Expert

Model

Design-time Analysis



Deployment Model



Software Architect

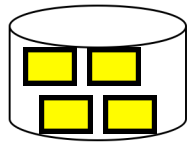
Deploy

Hardware System

QoS Pref.

Users

Overall Approach

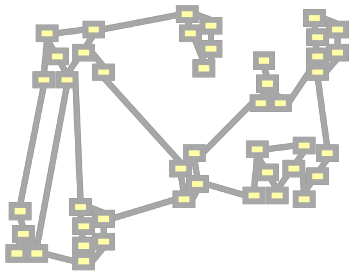


Component Repository

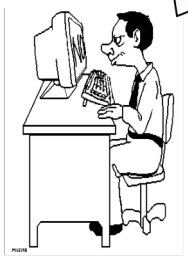
Domain Expert

Model

Design-time Analysis



Deployment Model



Software Architect

Deploy

Monitoring Data

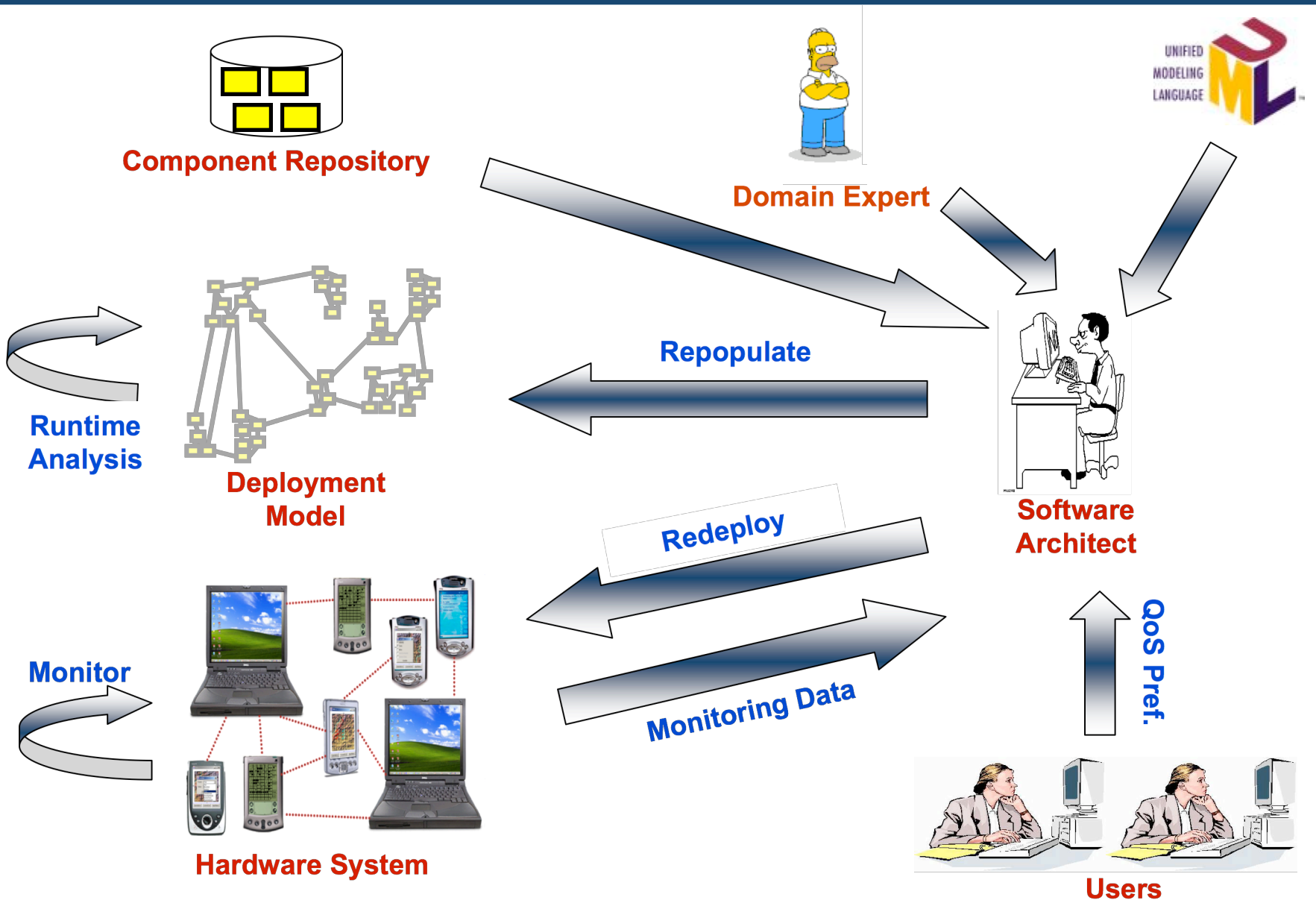
QoS Pref.

Monitor

Hardware System

Users

Overall Approach



Emerging Questions

- What is the best approach to measuring the quality of a pervasive system's architecture, including its deployment architecture?
  - Is the best architecture the one achieving the maximum QoS, or is it the one that degrades the least as a result of fluctuations in the system parameters?

- What is the most effective approach to representing the users' QoS preference in pervasive systems?
  - The users are frequently not easily accessible, or change at runtime. Even when the users are available, they may not be able to express their preferences in a rigorous manner.

Questions for the Broader Community

- There are many architectural decisions that impact the system's non-functional (QoS) properties that often depend on one another.
  - One may improve system's availability in several ways: caching or hoarding of data, replicating components, redeploying the software system, adapting the component's interfaces, etc.
- There is a lack of quantitative frameworks that allow the architect to reason about the architectural decisions and adaptation techniques.
  - What is the best way of modeling and relating architectural decisions?
  - What is the best approach to analyze and reason about them?
  - How can we affect the results of the analysis in pervasive environments?

Software Deployment Architecture and Quality-of-Service in Pervasive Environments

Presenter: Yuriy Brun
Computer Science Department
University of Southern California

Nenad Medvidovic
Computer Science Department
University of Southern California

Sam Malek
Department of Computer Science
George Mason University