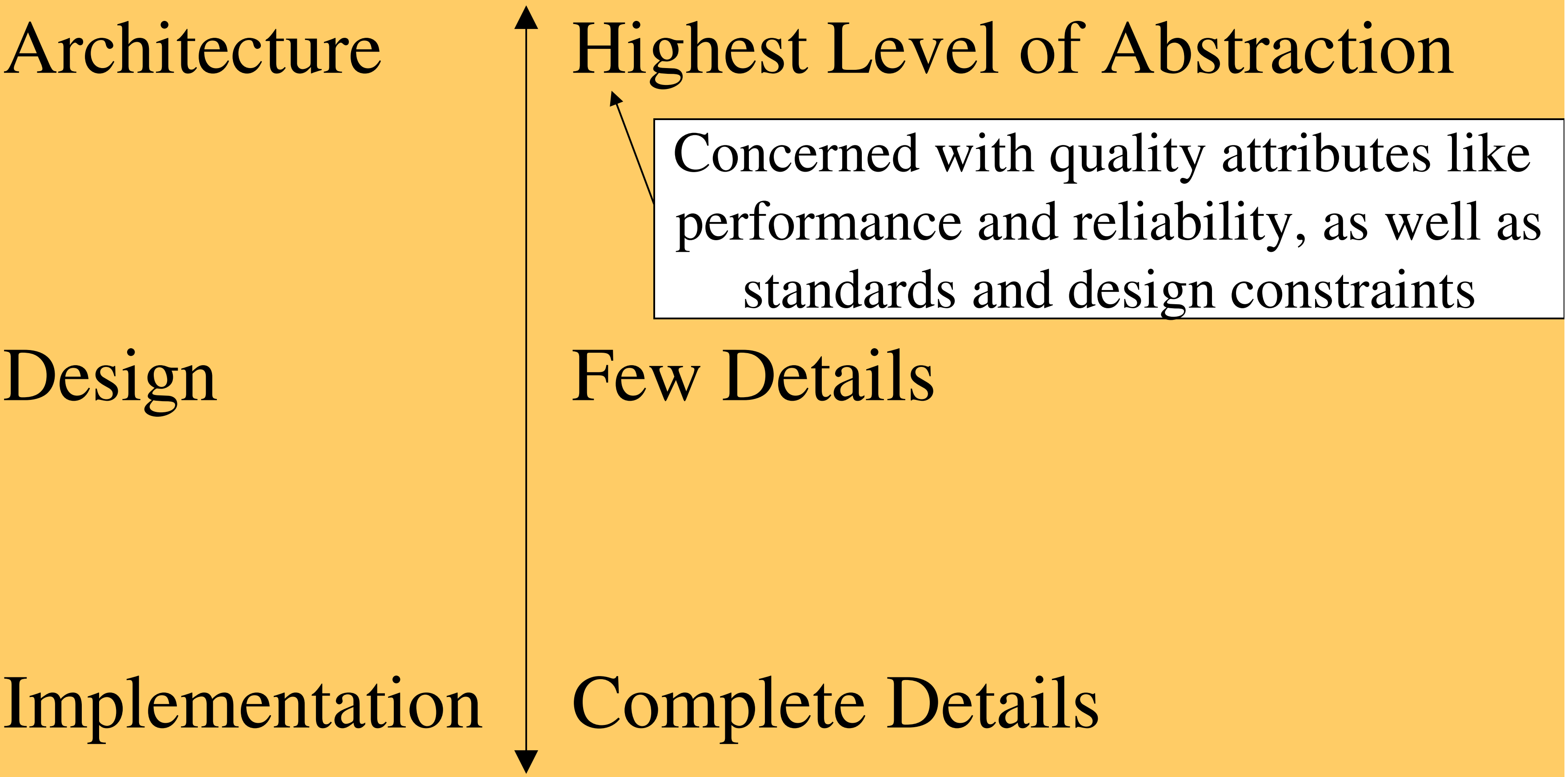
Software Architecture

- Definitions
  - http://www.sei.cmu.edu/architecture/published_definitions.html
  - ANSI/IEEE Std 1471-2000, Recommended Practice for Architectural Description of Software-Intensive Systems
    - Architecture is defined by the recommended practice as *the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.*

Levels of Abstraction



Architecture

Highest Level of Abstraction

Concerned with quality attributes like performance and reliability, as well as standards and design constraints

Design

Few Details

Implementation

Complete Details

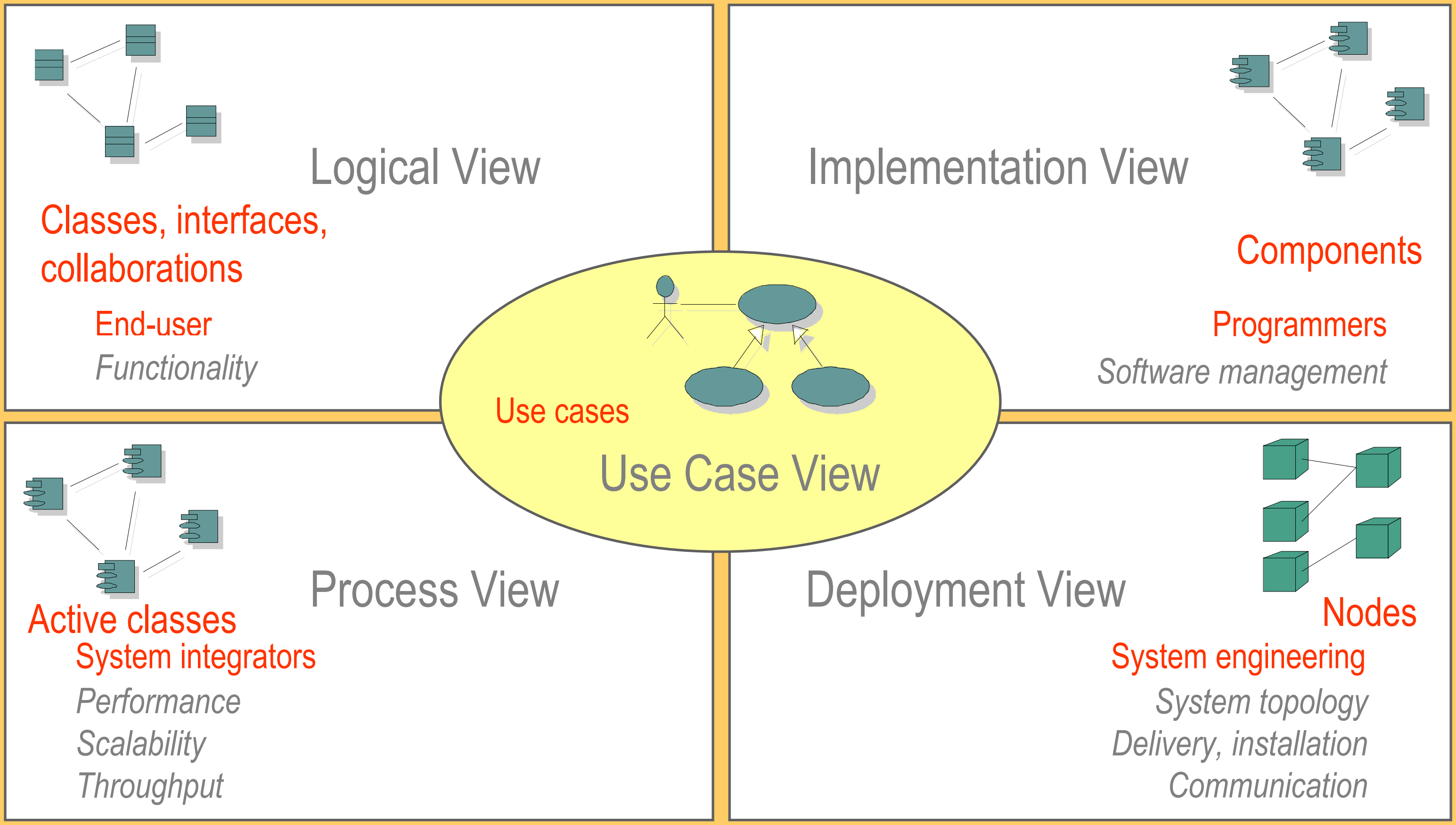
Typical Elements

Architecture	Deployment Diagrams, Architectural Styles
Design	Class Diagrams, Design Patterns
Implementation	Generics, Inheritance, Annotations, Statements

Architectural Styles

- Sequential
- Monolithic
- Pipe-and-filter
- Parallel/Distributed
- Layered/n-tier
- State-machine
- Client-Server
- Peer-to-Peer
- Event-Driven
- Component-Based
- Plugin
- Blackboard
- Service-oriented
- Space-based
- Representational State Transfer
- Database-centric

4+1 Architecture Views from the Unified Process



Logical View

Classes, interfaces, collaborations

End-user

Functionality

Implementation View

Components

Programmers

Software management

Use cases

Use Case View

Process View

Active classes

System integrators

Performance

Scalability

Throughput

Deployment View

Nodes

System engineering

System topology

Delivery, installation

Communication

Conceptual

Physical

UML Diagrams

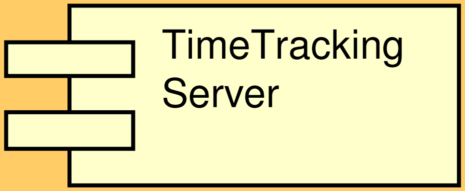
- Structural
  - Class, Object, Component, Deployment Diagrams
- Behavioral
  - Use-Case, Activity, Sequence, Communication/Collaboration, Statechart Diagrams

Logical View

- Describes an architecturally significant subset of the design model
- Contains a subset of classes, packages, and use-case realizations
- Concerns the functionality, behavior, use of frameworks and patterns
- Uses Class, Interaction, and State Diagrams

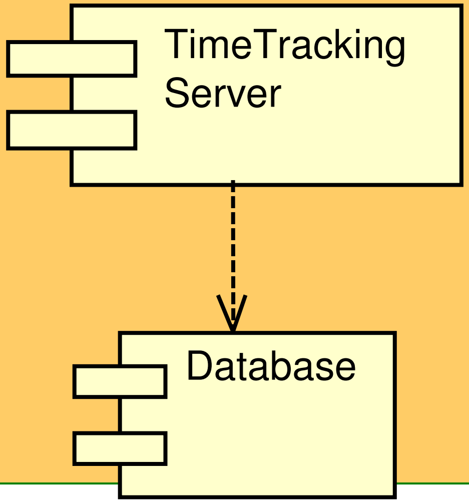
Process View

- Describes threads of control and communication between them
- Concerns the availability, reliability, scalability, performance, synchronization
- Uses Component, Class, and Collaboration Diagrams



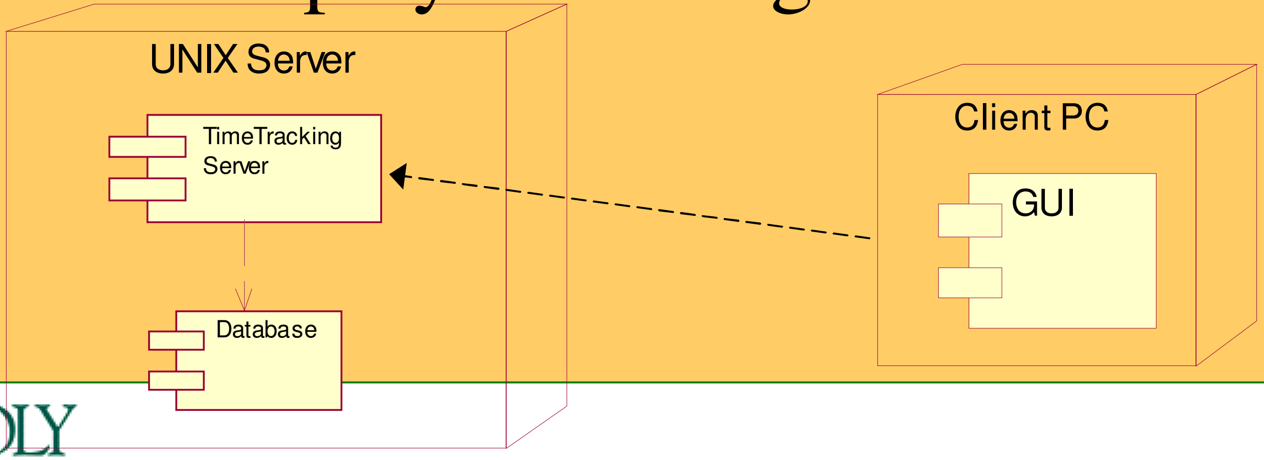
Implementation View

- Describes software component organization
- Concerns team organization and configuration management
- Uses Component Diagrams

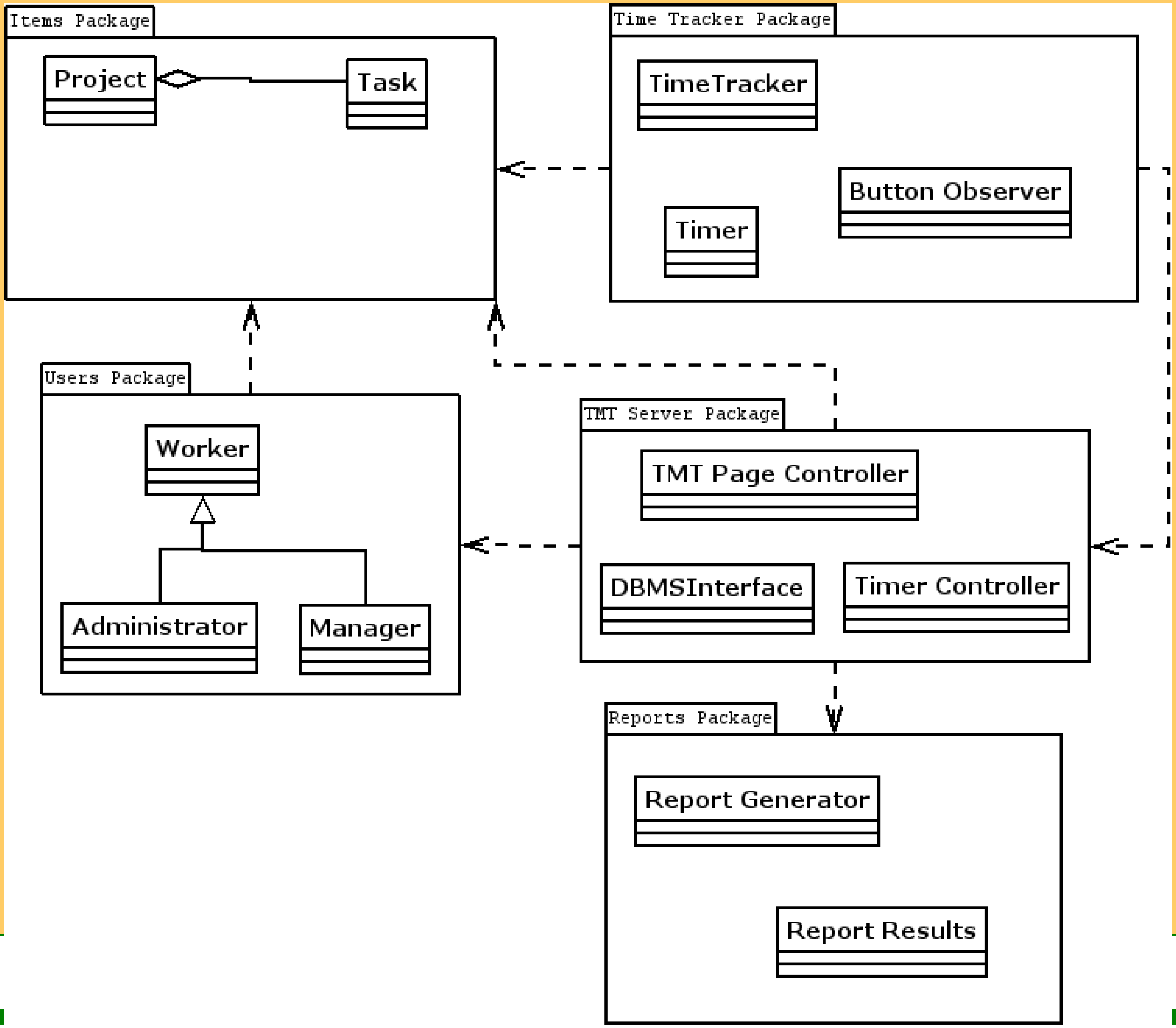


Deployment View

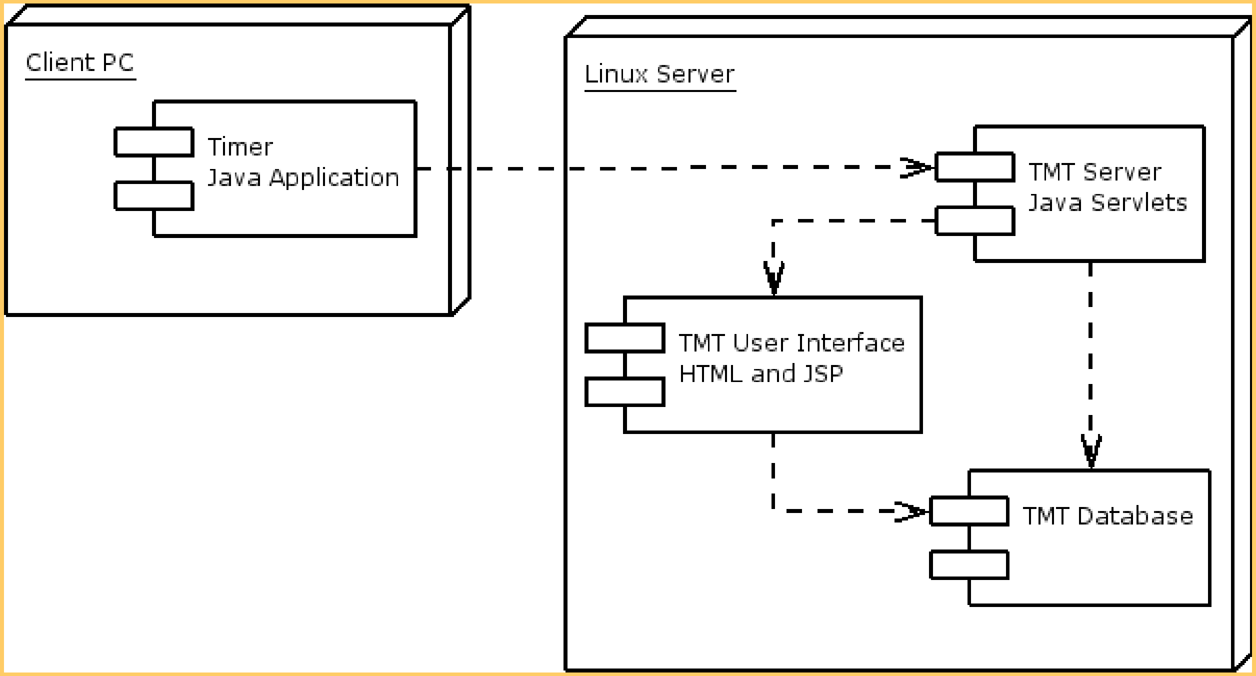
- Describes physical network configurations
- Concerns the performance, throughput, fault-tolerance, availability, installation, and maintenance
- Uses Deployment Diagrams



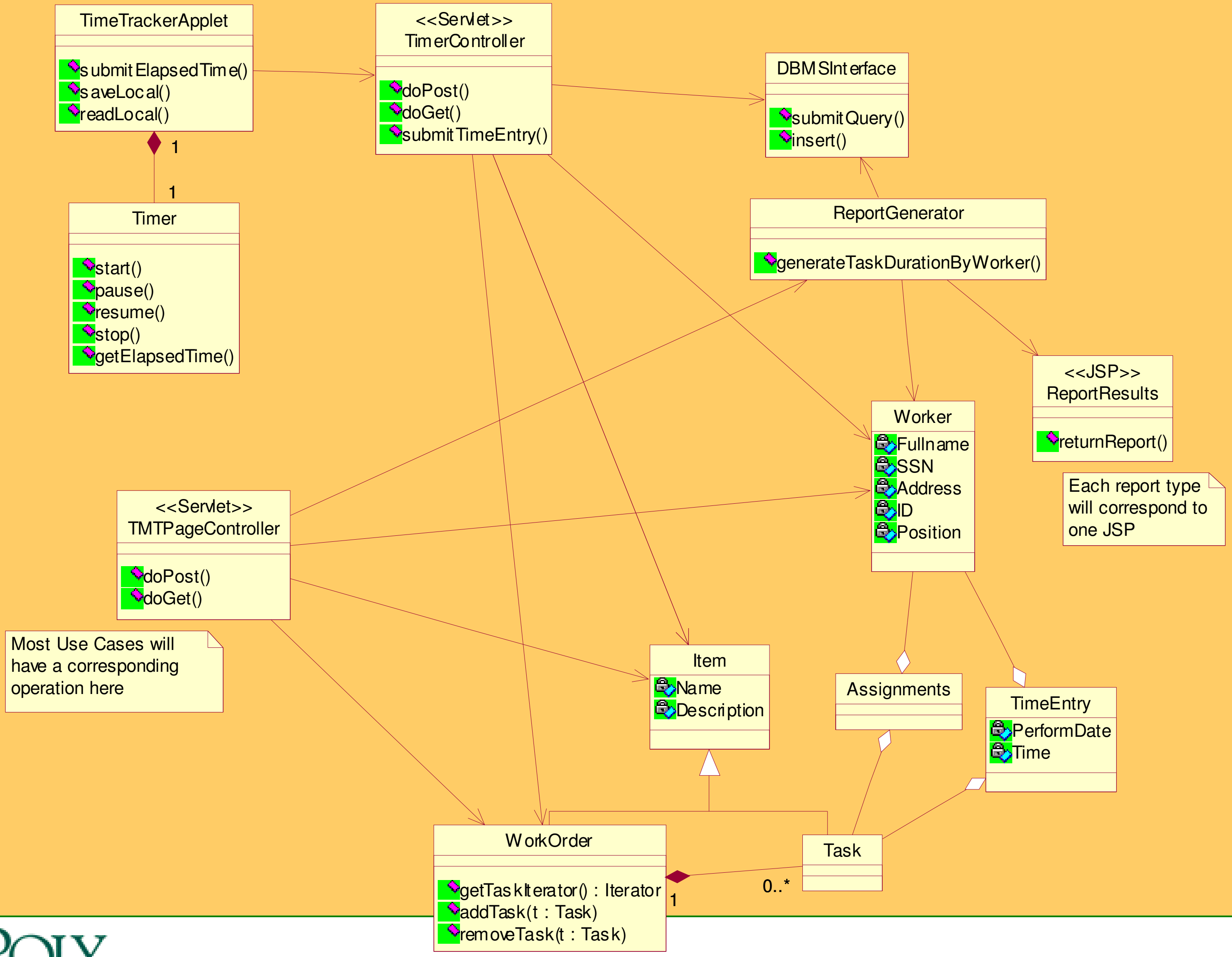
Architecturally Significant Classes Organized by Package



Deployment View



Architecturally Significant Classes



TimeTrackerApplet
submitElapsedTime()
saveLocal()
readLocal()

<<Servlet>>
TimerController
doPost()
doGet()
submitTimeEntry()

DBMSInterface
submitQuery()
insert()

1

1

Timer
start()
pause()
resume()
stop()
getElapsedTime()

ReportGenerator
generateTaskDurationByWorker()

<<JSP>>
ReportResults
returnReport()

Worker
Fullname
SSN
Address
ID
Position

Each report type will correspond to one JSP

<<Servlet>>
TMTPageController
doPost()
doGet()

Most Use Cases will have a corresponding operation here

Item
Name
Description

Assignments

TimeEntry
PerformDate
Time

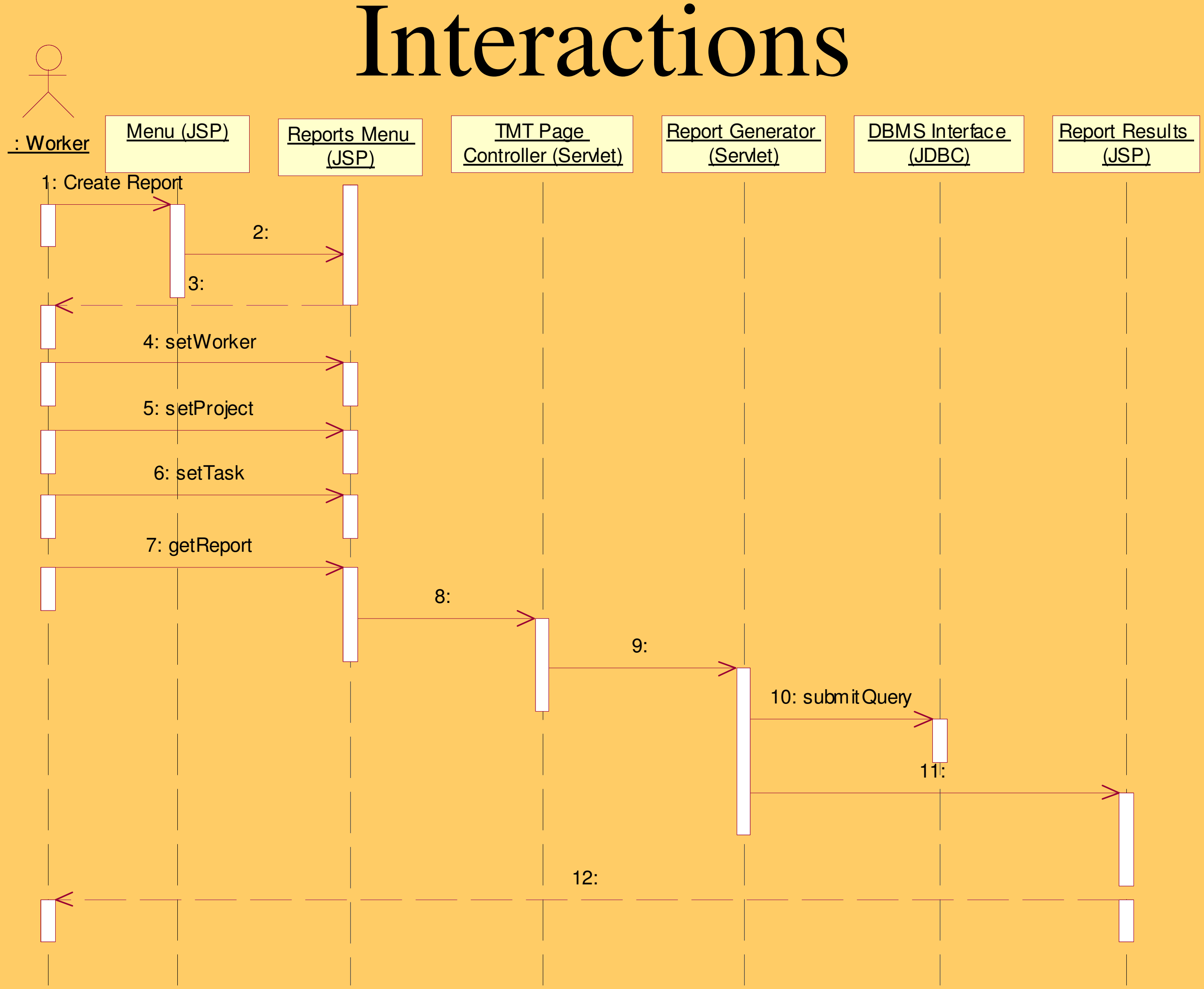
Task

WorkOrder
getTaskIterator() : Iterator
addTask(t : Task)
removeTask(t : Task)

1

0..*

Architecturally Significant Interactions



SW Architecture Process

- Chief Architect
  - with Architecture Review Board
- Democratic
  - Design by Committee
- SEI: http://www.sei.cmu.edu/architecture/
  - Architecture Tradeoff Analysis Method (ATAM)
- Visual Architecting Process
  - http://www.bredemeyer.com/howto.htm