Viewpoints and Perspectives

CSCE 742 - Lecture 3 - 08/30/2018

Three Pillars of Architectural Design

- **Stakeholders** are the people impacted by the architecture, who have differing expectations and needs.
- **Viewpoints** are used to structure architecture definition by focusing on aspects of the system being designed.
- **Perspectives** focus on how a particular quality attribute impacts each viewpoint of the architecture.

Today's Goals

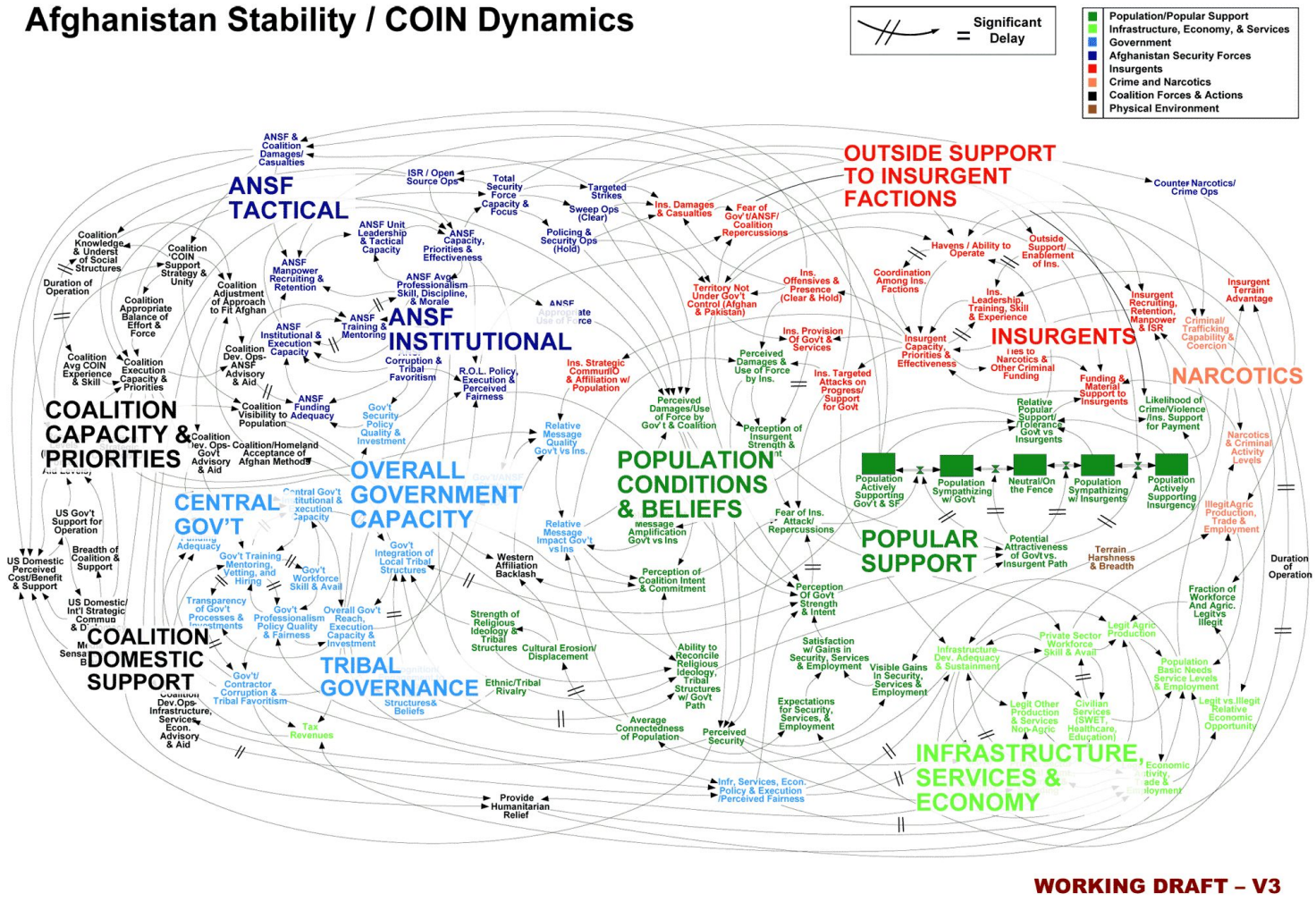
- Closely examine the concepts of views and viewpoints of architecture.
  - How we look at different aspects of architecture.
  - How we explain elements to stakeholders.
- Examine architectural perspectives.
  - Cross-cutting concerns based on non-functional quality properties.

Views and Viewpoints

Starting Questions

- How should the main system functions translate into architectural elements?
- How will those elements interact?
  - With each other? With the outside world?
- What information will be managed, stored, and presented?
- What physical hardware and software elements will be required?
- What operational features will be provided?
- What development, test, support, and training environments will be provided?

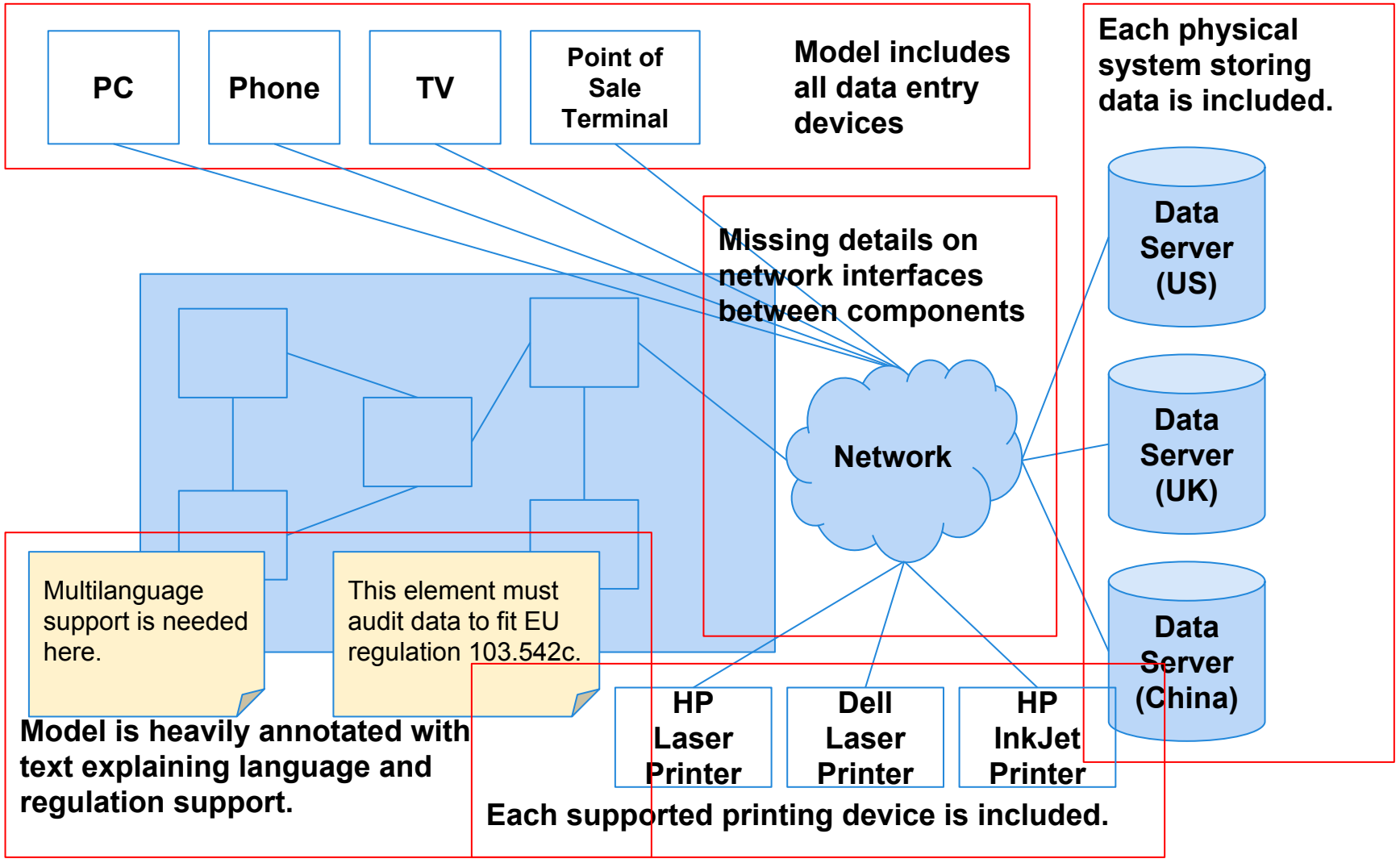
One Model to Rule Them All



The Airline Reservation System

- Complicating Factors:
  - Data is distributed across a number of systems in different physical locations.
  - Different data entry devices must be supported.
  - The system must present some information in different languages.
  - The system must print documents on a wide range of printers.
  - International regulations must be obeyed.
- The architect prepares a first draft of the architecture...

The Airline Reservation System



Problems

- Fails to please stakeholders.
  - Users find it too difficult, systems engineers ignore because it doesn't address network details, legal needs more detail on regulatory compliance.
- Difficult to update.
  - Each new data entry device or printer must be added.
- Diagram will become obsolete.
- Model will fail to inform development.
  - The complicating factors still exist, but this design failed to address them.

Separation of Concerns

- It is not possible to capture all features and quality properties of a complex system in a single model that is understandable and of value to all stakeholders.
- Instead, consider separation of concerns.
  - Split into different aspects of the architecture.
  - Network view, printing view, device view, regulatory view, structural view, etc.

Architectural Views

- A **view** is a representation of one or more structural aspects of an architecture that illustrates how the architecture addresses concerns held by its stakeholders.
  - Can be arbitrarily narrow or general, depending on the stakeholder.
  - Are based on specific concerns, which map to various stakeholders.
  - Can be detailed or high-level.

Viewpoints

- A **viewpoint** is a collection of patterns, templates, and conventions for constructing one type of view.
  - Framework for a “standard” view.
- Defines the stakeholders and concerns are reflected in the viewpoint.
- Defines guidelines, principles, and models for constructing a view.
- Standardizes language and approach.

Viewpoints

Supports system construction

Functional Viewpoint

Development Viewpoint

Information Viewpoint

Deployment Viewpoint

Concurrency Viewpoint

Operational Viewpoint

Describe software artifacts and the primary organization of the system.

Characterize the system once in a live environment.

Artifact Viewpoints

- Functional Viewpoint
  - Describes functional elements, responsibilities, and their interactions.
- Information Viewpoint
  - How the architecture stores, manipulates, and distributes information.
  - Offers a view of static data structure and information flow between elements.
- Concurrency Viewpoint
  - How concurrent processes are coordinated and controlled.

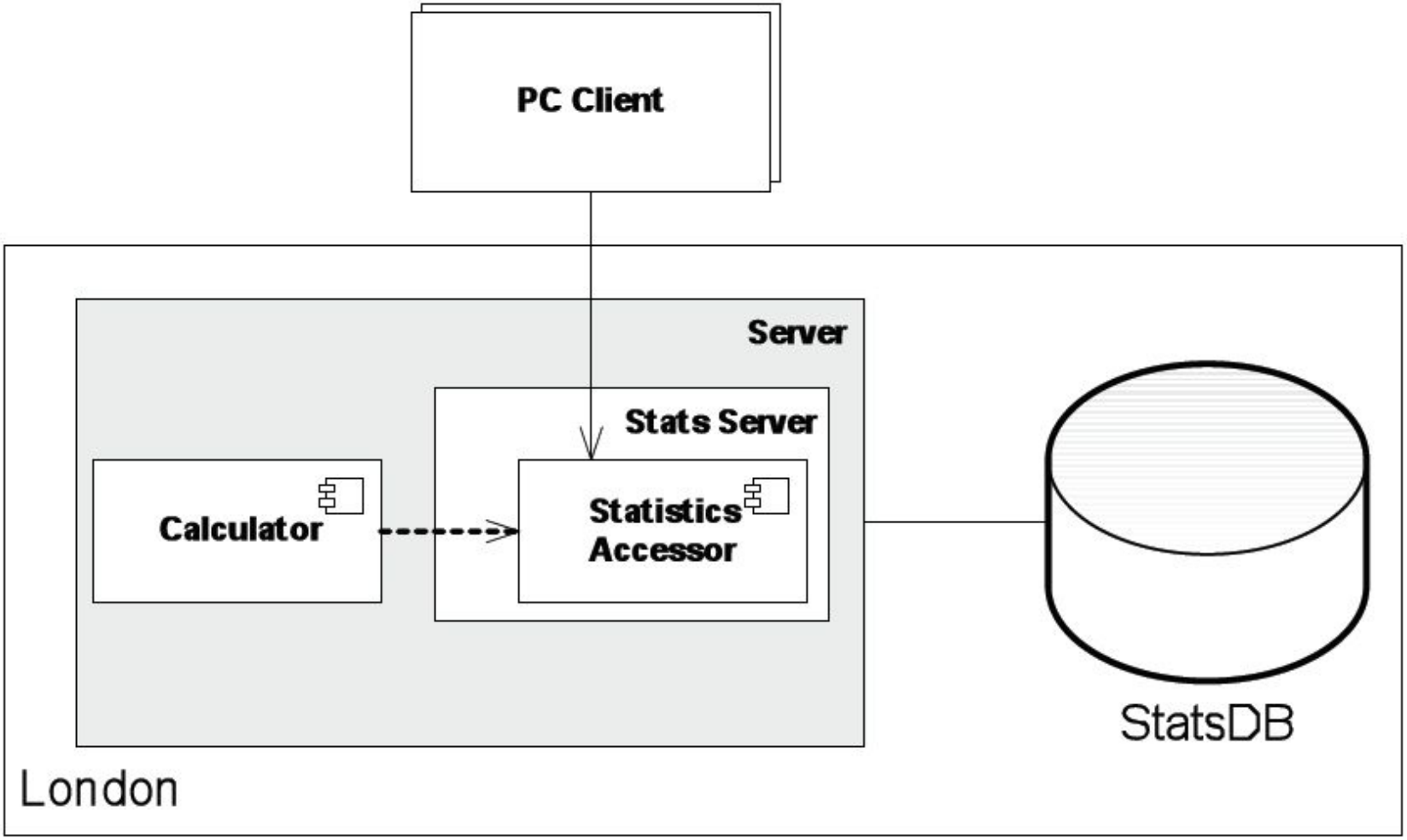
Environmental Viewpoints

- Development Viewpoint
  - Details the architecture in place to support building, testing, and maintaining the system.
- Deployment Viewpoint
  - Captures runtime software, hardware, and network dependencies of the system.
- Operational Viewpoint
  - Describes how the system will be used, administered, and supported when running in production.

Example: Statistics Processing

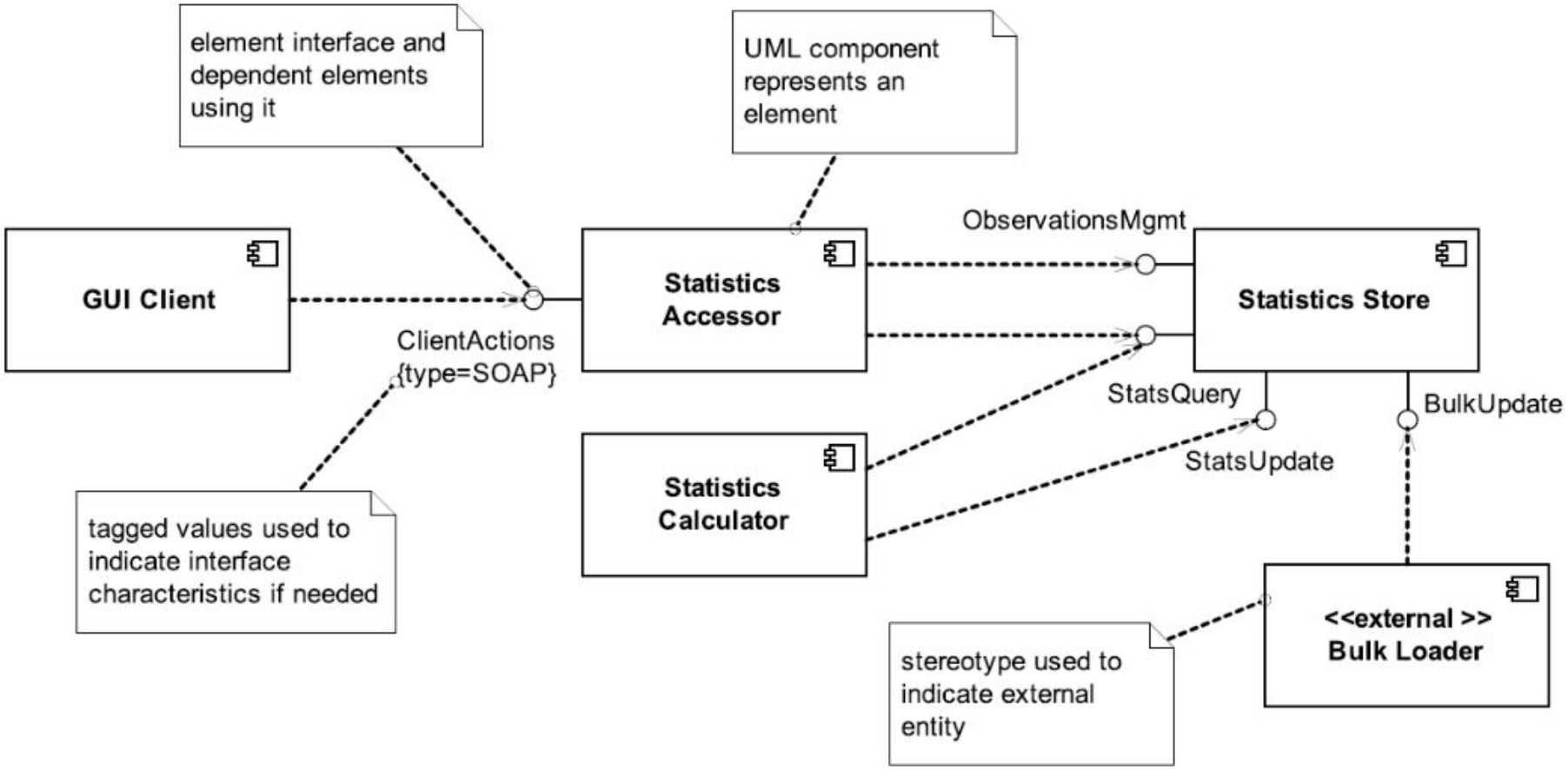
- Raw data is loaded into a database.
- Derived statistics are calculated automatically based on the data.
- Statisticians view the data and make reports.
- Clients access statistics and make deductions that are checked manually.

First Try...



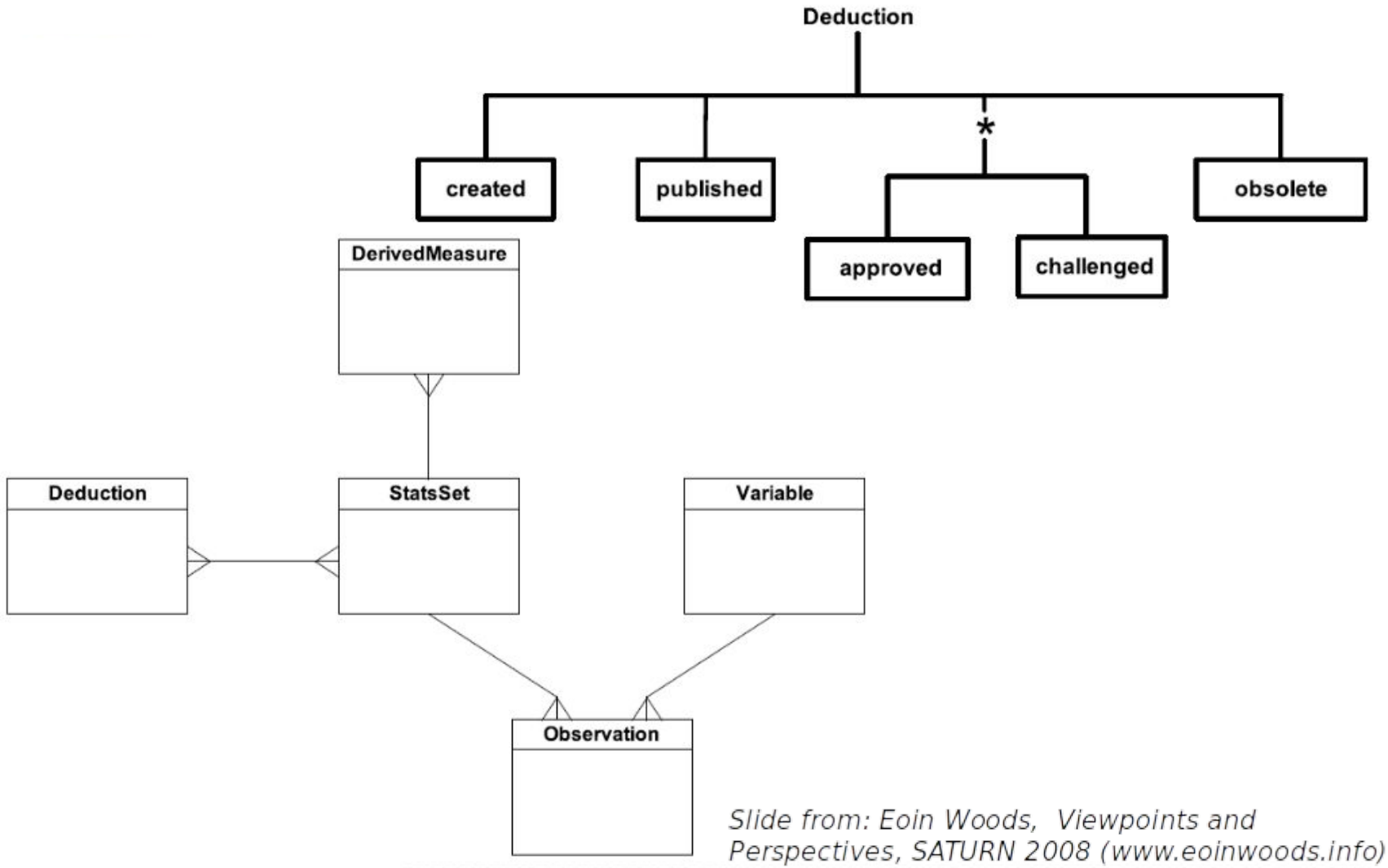
Slide from: Eoin Woods, Viewpoints and Perspectives, SATURN 2008 (www.eoinwoods.info)

Functional View



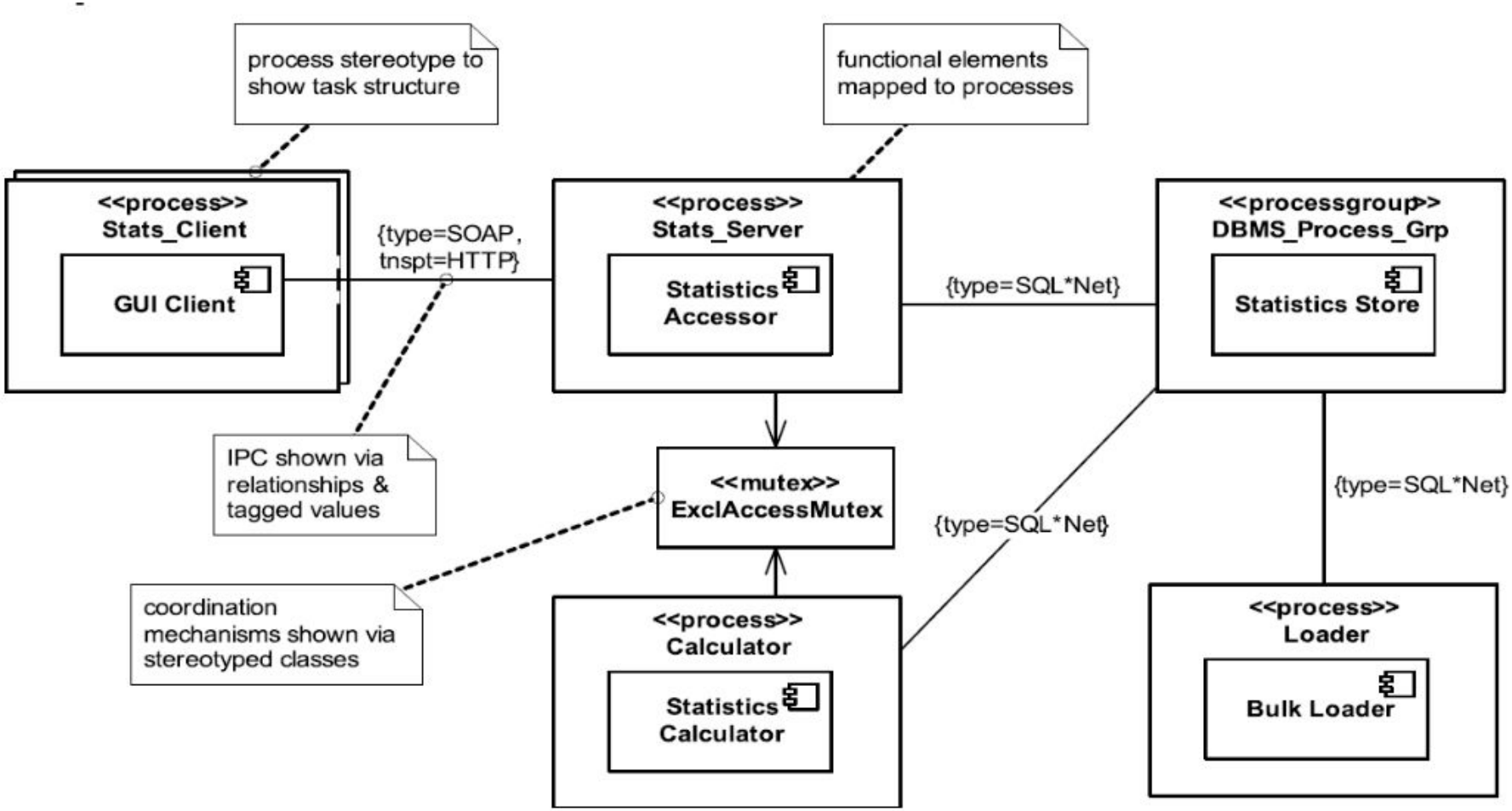
Slide from: Eoin Woods, Viewpoints and Perspectives, SATURN 2008 (www.eoinwoods.info)

Information View



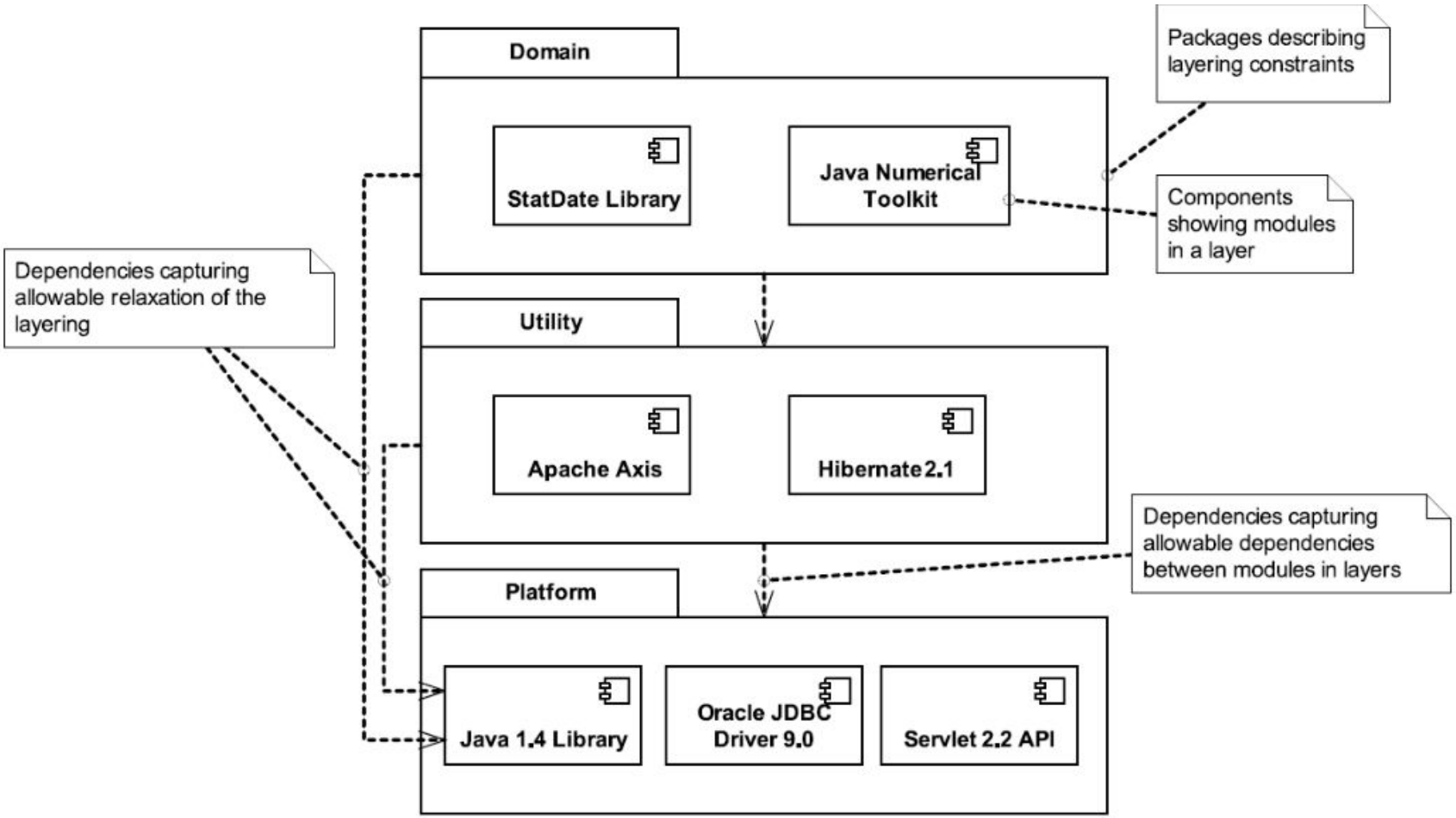
Slide from: Eoin Woods, Viewpoints and Perspectives, SATURN 2008 (www.eoinwoods.info)

Concurrency View



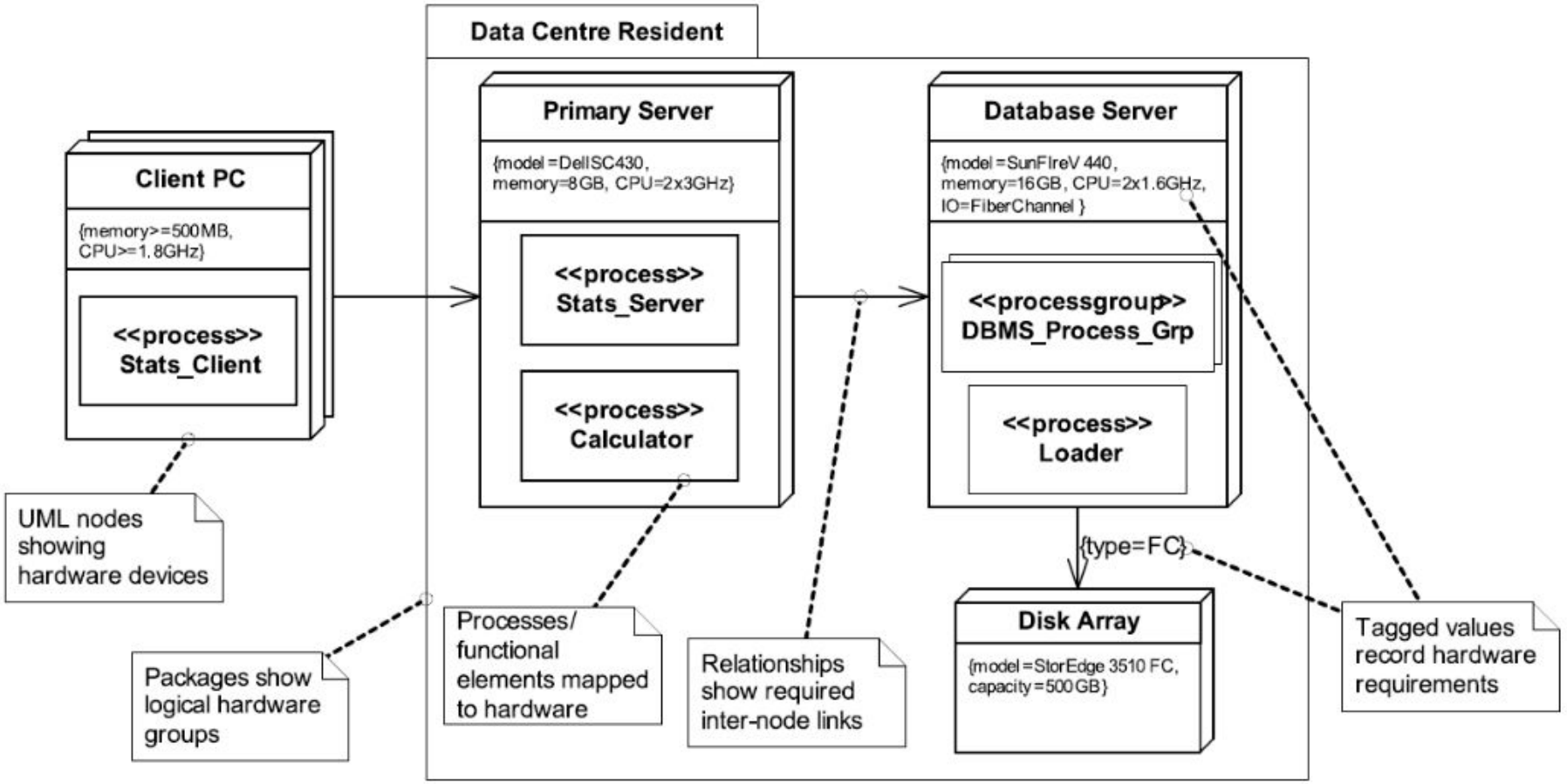
Slide from: Eoin Woods, Viewpoints and Perspectives, SATURN 2008 (www.eoinwoods.info)

Development View



Slide from: Eoin Woods, Viewpoints and Perspectives, SATURN 2008 (www.eoinwoods.info)

Deployment View



Slide from: Eoin Woods, Viewpoints and Perspectives, SATURN 2008 (www.eoinwoods.info)

Deployment View (2)

Client PC	▪ Windows XP SP1 ▪ Java JRE 1.4.2_06 or later ▪ Internet Explorer 6.0 SP1
Primary Server	▪ Windows 2003 server, w/sec patches ▪ Java SDK 1.4.2_06 or later ▪ Apache Tomcat 5.5.9 or later

Database Server	▪ Solaris 9.0 w/Aug05 patch cluster
	▪ Oracle 9.2.0.2 Std Edition
	  ▪ 10GB buffer cache, auto sized SGA
	  ▪ auto storage management, 2 table spaces
	▪ OEM 9.2.0.2 installed and working

Slide from: Eoin Woods, Viewpoints and Perspectives, SATURN 2008 (www.eoinwoods.info)

Operational View

- Installation Model
  - Installation groups
  - Dependencies and constraints
  - Backout strategy
- Operational CM Model
  - Configuration groups and dependencies
  - Configuration parameter sets
  - Operational control (switching between sets)
- Administration Model
  - Monitoring and control facilities required and provided
  - Required routine operational procedures
  - Required operational action in case of error conditions

Slide from: Eoin Woods, Viewpoints and Perspectives, SATURN 2008 (www.eoinwoods.info)

Viewpoint Benefits

- Helps structure architecture definition.
  - Usually a very unstructured activity.
- Enables separation of concerns.
  - Helps design, analysis, and communication by abstracting unimportant details.
- Allows communication with stakeholders.
  - Guide stakeholders to parts of the architecture description based on their concerns.
- Improves developer focus
  - Ensures right system gets built, and that all details are captured.

Viewpoint Pitfalls

- Inconsistency between views
  - Cross-checking is a tedious manual process.
- Selection of the wrong views
  - Requires careful consideration of stakeholders and their concerns.
- Fragmentation
  - Many views can make the “big picture” hard to follow.
  - Each view requires significant effort.
  - Eliminate views that do not address significant concerns for the system.

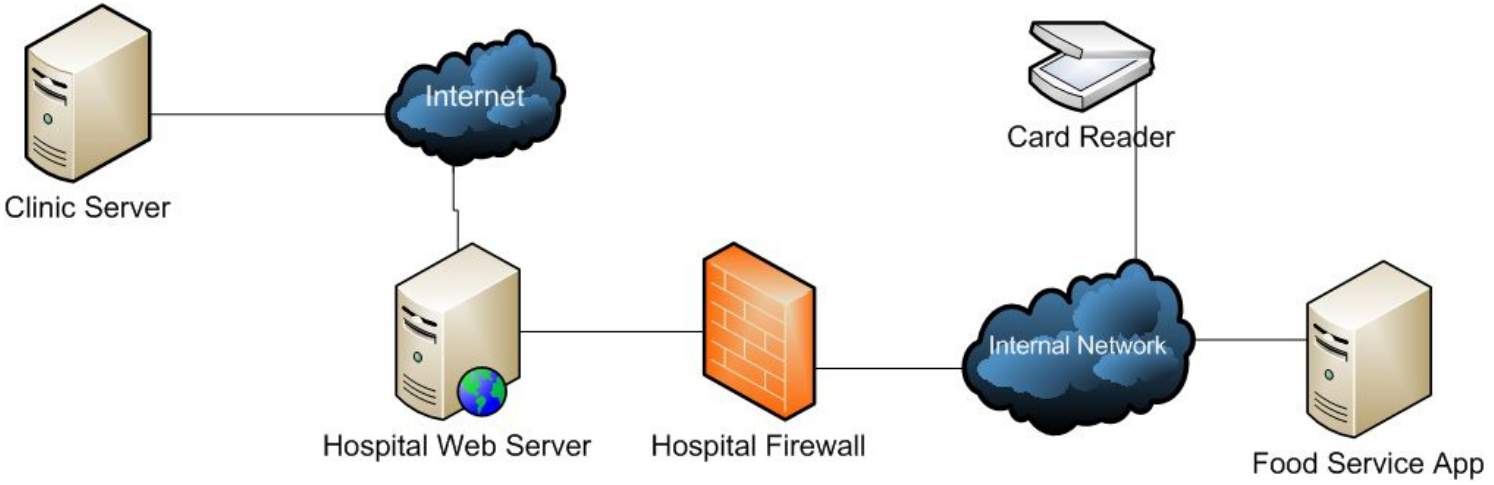
View Exercise

Hospital Food

- A hospital has purchased a small clinic across the city.
- The hospital will have to provide meals to the clinic.
- The patients should be able to choose their meals using standard hospital meal tickets.

System Constraints

- Hospital/clinic are connected to the internet.
  - Current meal system is antiquated; managed with ticket readers and is on the internal network (behind the firewall).
  - In the current system, no authentication or encryption.
- The volume of traffic will likely be low
  - (anticipated top demand is 30-40 users).
- System functionality: 1) Transmit menu to clinic 2) Allow patients to order food 3) Deliver food to clinic



How Would You Architect This?

1. Who are the stakeholders in this system?
   a. What questions would you ask them?
2. What information is missing from the system description?
3. What are the relevant views? (This can include aspects not covered in this lecture)
   a. What considerations belong to each view?

Many Possible Solutions

- Webapp
  - Web-based food application. Need to add authentication and security.
- VPN
  - Enter in meal tickets. Could duplicate card reader output over the VPN.

Think Outside of the Box

- Use a fax machine.
  - Fax meal card in the morning.
  - Fax back completed card.
  - Read completed faxes into existing card readers.
  - Very inexpensive, and can be automated.
    - But doesn't have to be...

Architectural Perspectives

Quality Attributes

- The architecture not only dictates *what* the system does, but *how* it does it.
  - How *quickly* it runs.
  - How *secure* it is.
  - How *available* its services are.
  - How easy it is to *modify*.
- **Quality attributes** describe desired non-functional properties of systems.
- An architect must prioritize quality attributes and design a system that meets thresholds.

What do you prioritize during development?

Examples of Qualities

- Performance
- Scalability
- Security
- Availability
- Resilience
- Modifiability
- Supportability
- Reliability
- Safety
- Portability
- Development Efficiency
- Time to Deliver
- Tool Support
- Geographic Distribution
- Others?

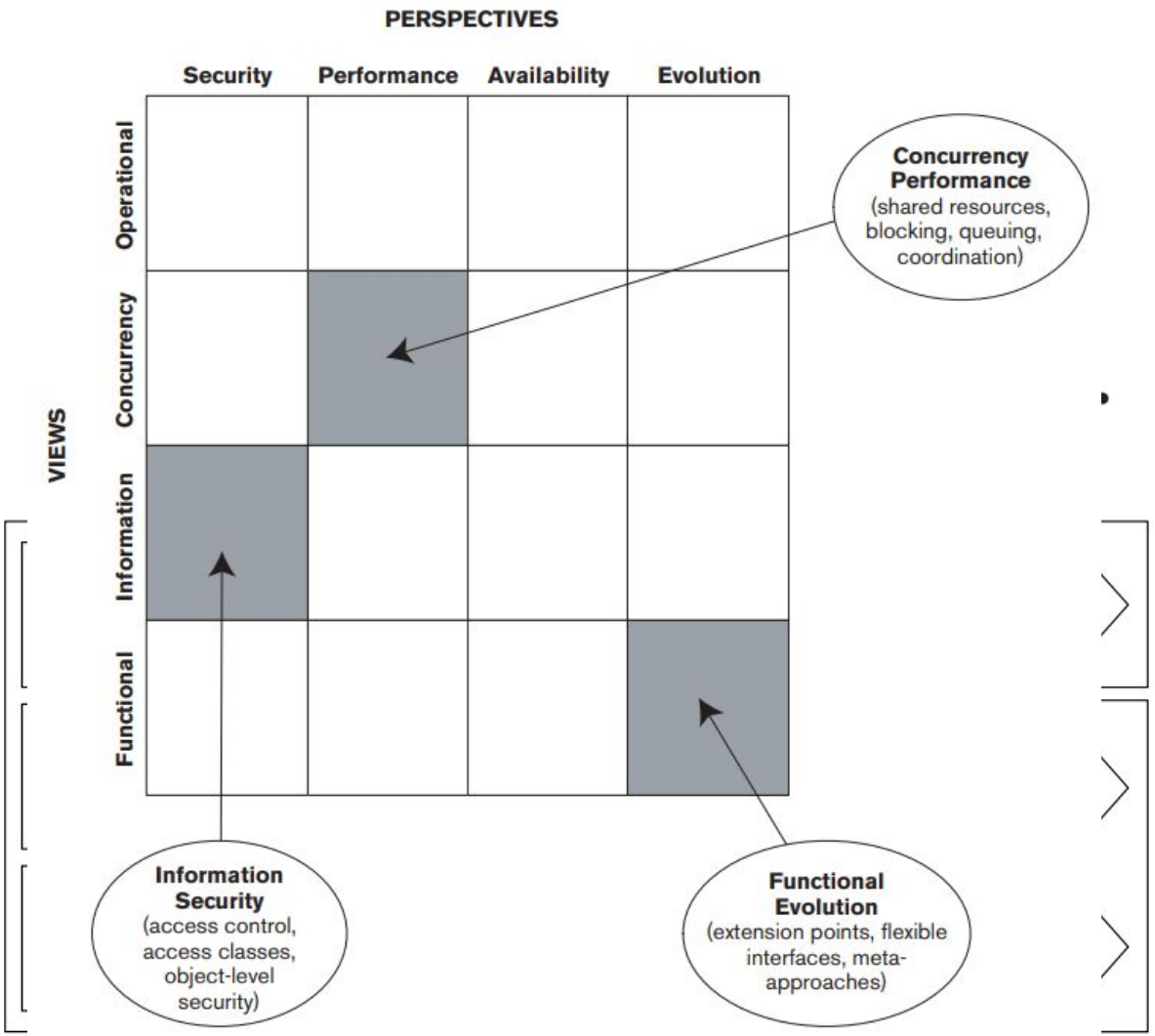
Qualities Impact Many Views

- Security is a very important quality. It must be detailed in the architecture.
- No security “viewpoint”, as it impacts all viewpoints.
  - Functional: System needs to identify/authenticate users.
  - Information: System must control classes of access to information at various levels of granularity.
  - Operational: System must maintain and distribute passwords, software dependencies must be patched.

Architectural Perspective

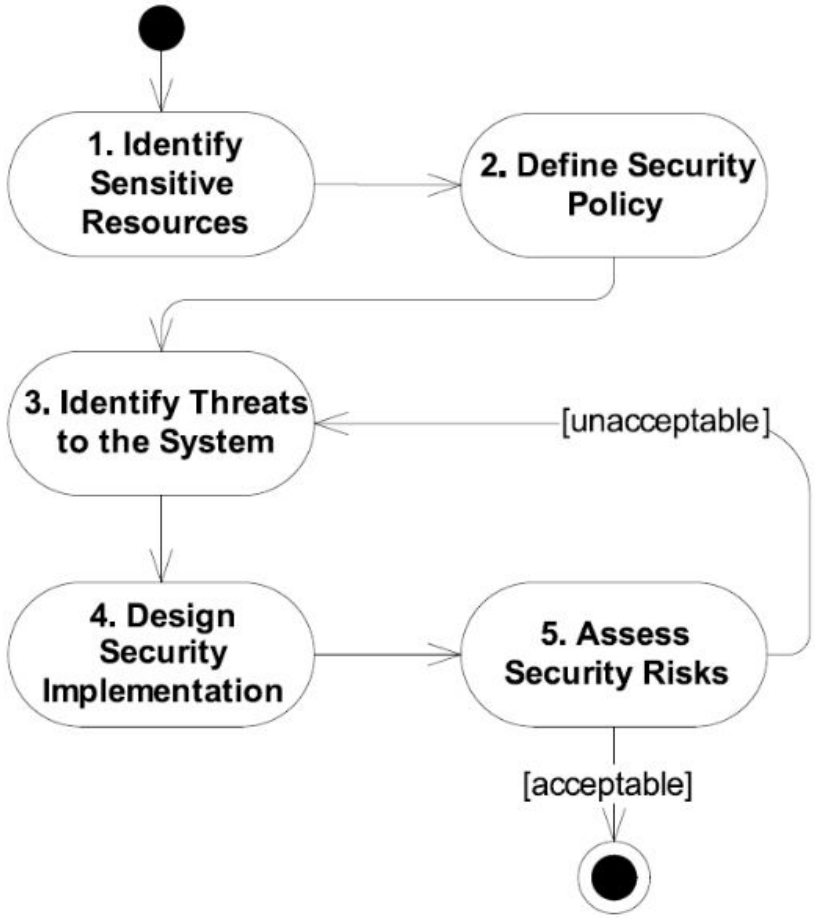
- **Architectural Perspectives** are used to discuss how particular quality attributes affect each view of the overall architecture.
- A perspective is a collection of activities, tactics, and guidelines used to ensure that a system exhibits the chosen set of quality attributes.
  - Systemizes what a good architect does: studies properties, assesses architectural choices, selects and applies tactics to ensure quality.

Applying Perspectives to Views



Example: Security Activity

- Perform activities, based on viewpoint, to identify how the perspective impacts that viewpoint.
- Suggests changes to the architecture.
  - Different partition to **functional** elements.
  - Introduce new hardware/software elements to **deployment** environment.
  - Identify new **operational** procedures to support secure operation.



Slide from: Eoin Woods, Viewpoints and Perspectives, SATURN 2008 (www.eoinwoods.info)

Example: Security Activity

- Sensitive Resources
  - Data in the database
- Security Threats
  - Operators stealing backups.
  - Admins querying data, seeing names.
  - Bribing investigating officers
  - Internal attack on database via network.

Slide from: Eoin Woods, Viewpoints and Perspectives, SATURN 2008 (www.eoinwoods.info)

Example: Security Activity

- Security Countermeasures
  - Backups: encrypt data in the database
    - Does this impact performance?
    - Does this impact the difficulty of maintaining availability?
  - Hiding names: use hash instead of names, protect names at a higher security level.
    - More complexity in development?
    - Possible performance impact?

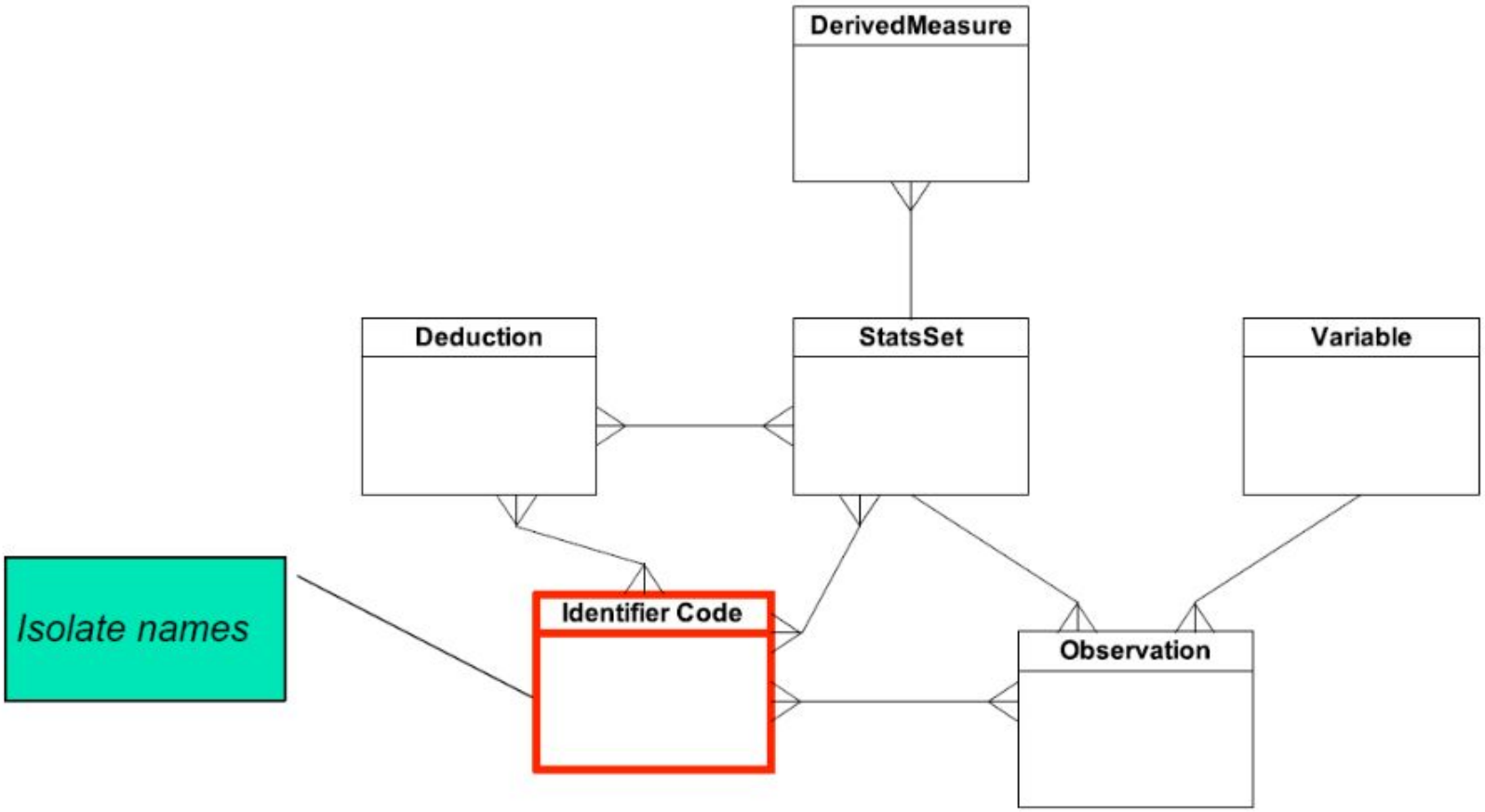
Slide from: Eoin Woods, Viewpoints and Perspectives, SATURN 2008 (www.eoinwoods.info)

Example: Security Activity

- Security Countermeasures
  - Bribery: add audit trail for data access
    - Does this impact performance?
    - Does this impact implementation complexity?
    - How do you protect the audit trail?
  - Network attacks: harden database, firewalls
    - Deployment and admin cost?
    - Hardware and operational cost?

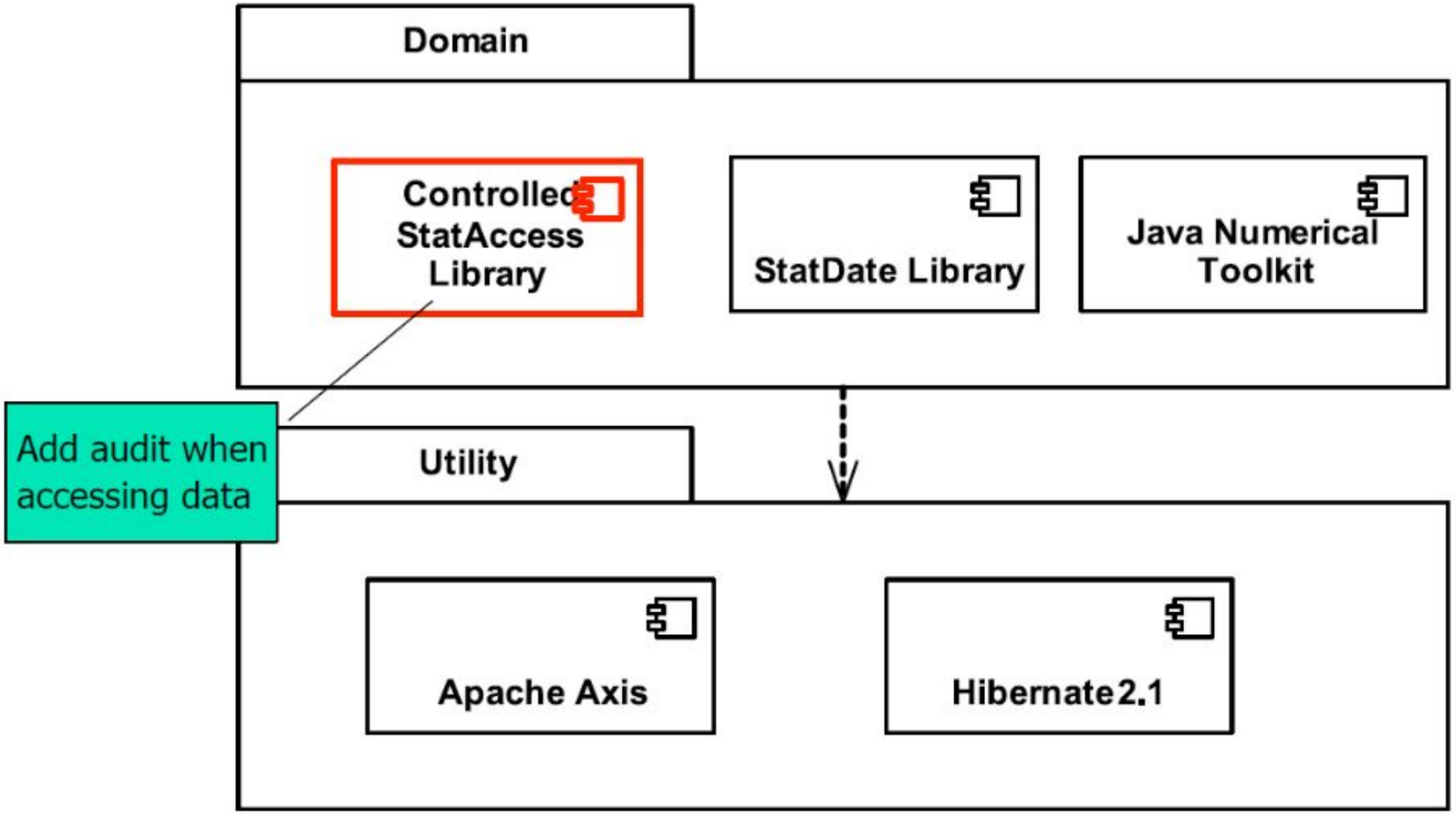
Slide from: Eoin Woods, Viewpoints and Perspectives, SATURN 2008 (www.eoinwoods.info)

Impact on Information View



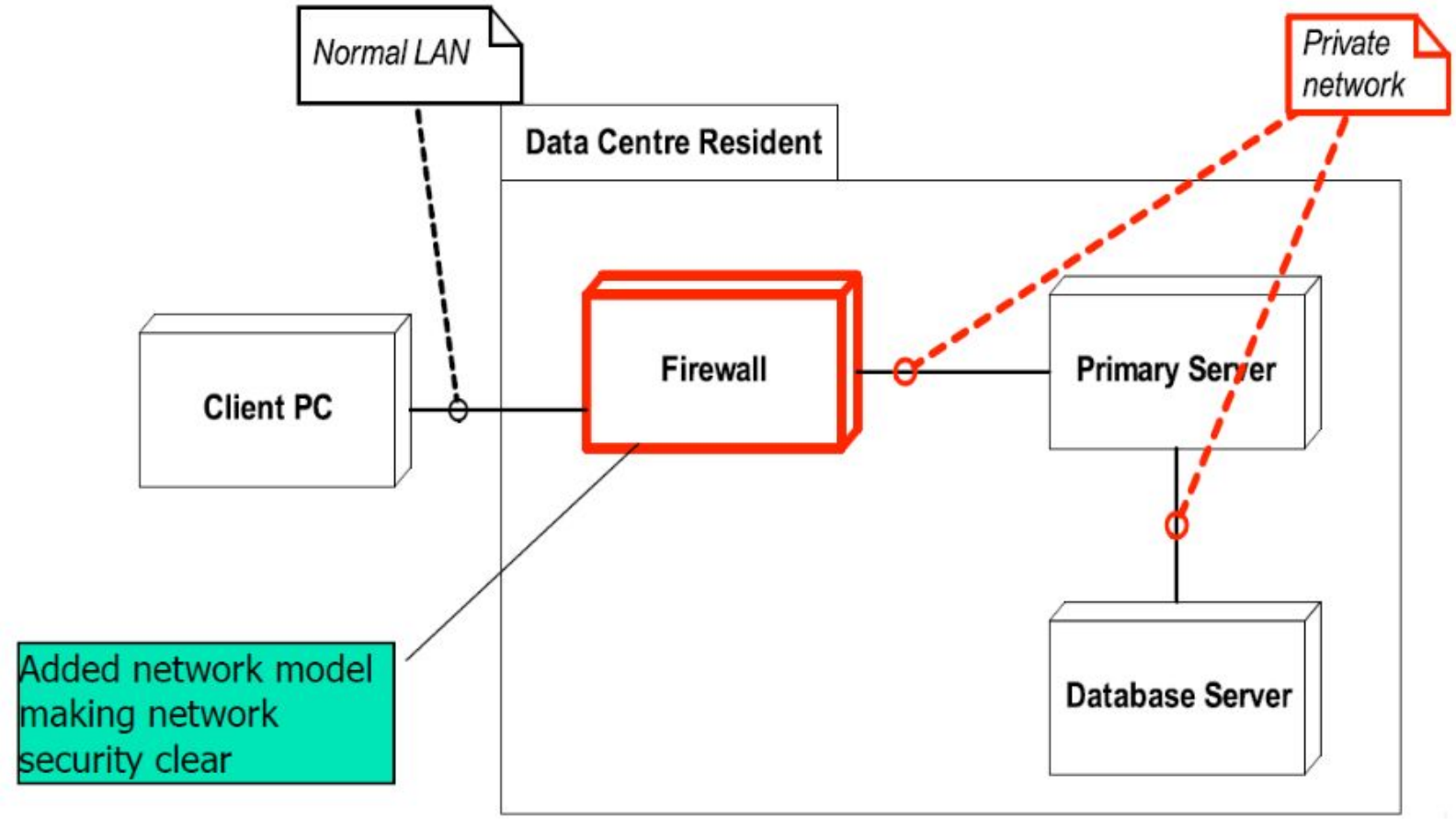
Slide from: Eoin Woods, Viewpoints and Perspectives, SATURN 2008 (www.eoinwoods.info)

Impact on Development View



Slide from: Eoin Woods, Viewpoints and Perspectives, SATURN 2008 (www.eoinwoods.info)

Impact on Deployment View



Slide from: Eoin Woods, Viewpoints and Perspectives, SATURN 2008 (www.eoinwoods.info)

Other Impacts

- Revisions to information or deployment view with changes to database security.
- Need to capture impact on operational view.
- Consider impact on other quality attributes (performance, availability).
  - Need to alter performance models to allow for encryption, audit, etc.
- **Key point:** perspectives cause changes to multiple views. Make sure you maintain consistency.

Consequences of Perspectives

- Insights into the system's ability to meet a quality property.
  - Applying a perspective tends to lead to a model demonstrating that an architecture meets a quality property or misses it.
- Improvements to the chosen architecture.
  - Leading to changes in models
- Artifacts that offer supporting architectural information and significant lasting value.

Perspective Benefits

- Offers conventions for **describing** qualities.
  - Performance perspective offers standardized measures and guidance for measuring them.
- Defines **concerns** that guide decisions.
  - Performance perspective defines concerns, i.e., response time and predictability.
- Describes how to **validate** quality.
  - Performance perspective offers advice for enacting simulations to predict performance.
- Offers **solutions** common across systems.
  - Performance perspective describes how hardware can be multiplexed.

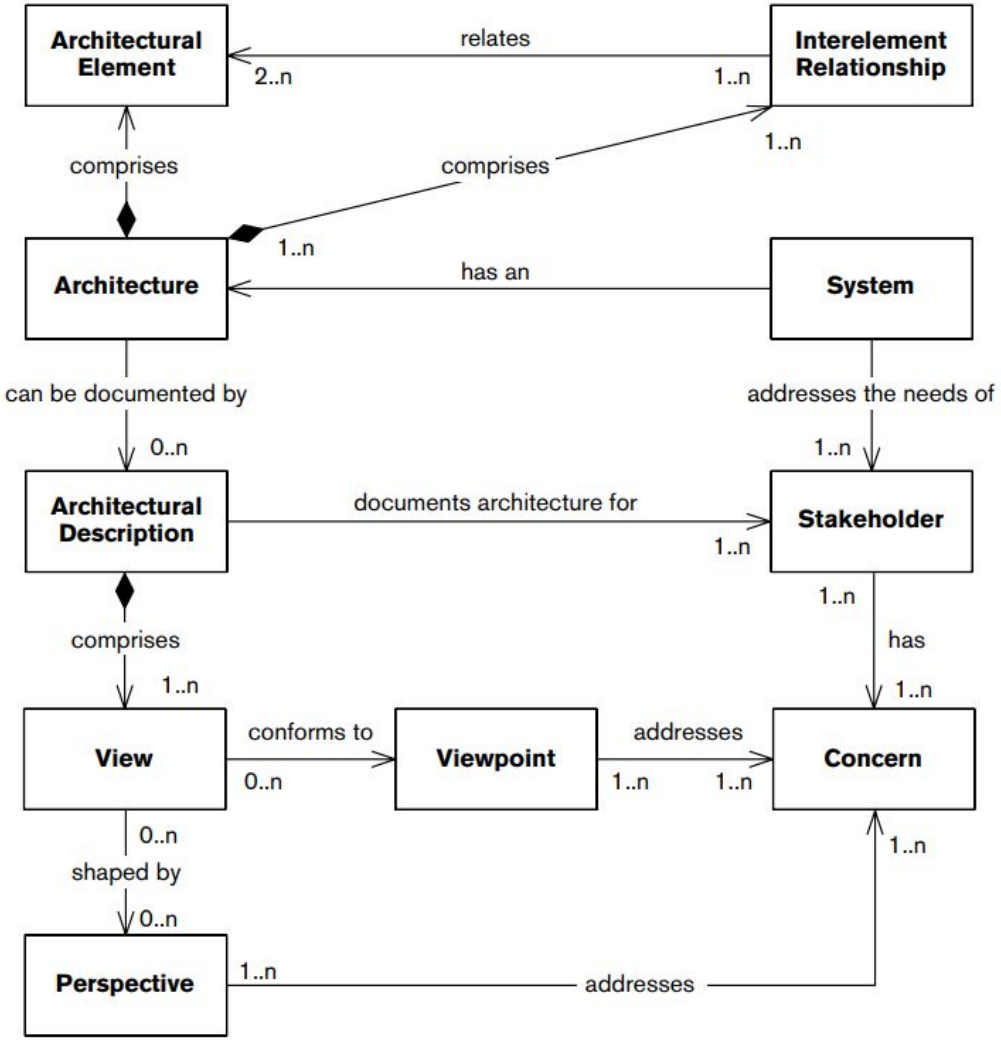
Perspective Pitfalls

- Each perspective addresses a single set of concerns and will conflict with other perspectives.
- Stakeholder concerns and priorities are different for every system. How much you need to consider each perspective varies.
- Perspectives offer general advice. Your situation may differ.

Key Points

- Viewpoints and perspectives offer a way to organize architectural information.
  - Allows detailed thinking about different aspects of design while abstracting other details (focus on the software structure, then on information format, etc.)
  - Describes expected activities and common pitfalls for each view of the system.
  - Must be tailored to your product.

Bringing it Together



Next Time

- Architecture Definition
  - Sources: Rozanski & Woods: ch. 5-7

- Homework: Team selections due next class. Instructions on course website.
- Reading Assignment 1:
  - Whalen, Gacek, Cofer, Murugesan, Heimdahl, Rayadurgam. *Your "What" is my "How": Iteration and Hierarchy in System design*
  - Due September 6th

Reading Assignment

- Whalen, Gacek, Cofer, Murugesan, Heimdahl, Rayadurgam. *Your "What" is my "How": Iteration and Hierarchy in System design*
- Individual assignment. Paper on website.
- Read the paper and turn in a one-page write-up:
  - Summary of the paper.
  - Your opinion on the work.
    - Do the authors' arguments make sense in the real world?
    - Where does their opinion fall short?
  - Your thoughts on how these ideas could be improved and extended.