



OT2004 - TU1

*Software Architecture Using  
Viewpoints and Perspectives*

Eoin Woods  
Artechra Limited  
eoin.woods@  
artechra.com

Nick Rozanski  
French Thornton  
nick.rozanski@  
french-thornton.co.uk

# Timetable

00:00 – 00:10	Introductions
00:10 – 00:40	Overview of viewpoints
00:40 – 00:50	Identify example systems
00:50 – 01:15	Ex1: identify viewpoints
01:15 – 01:30	Break
01:30 – 01:45	Overview of perspectives
01:45 – 02:15	Ex2: identify perspectives
02:15 – 02:30	Review and discussion

# Introductions

## Nick Rozanski

- Software architect
- IT/information systems/systems integration
- French Thornton

## Eoin Woods

- Software architect
- information systems/Internet/system software
- Artechra (independent)

# Introductions

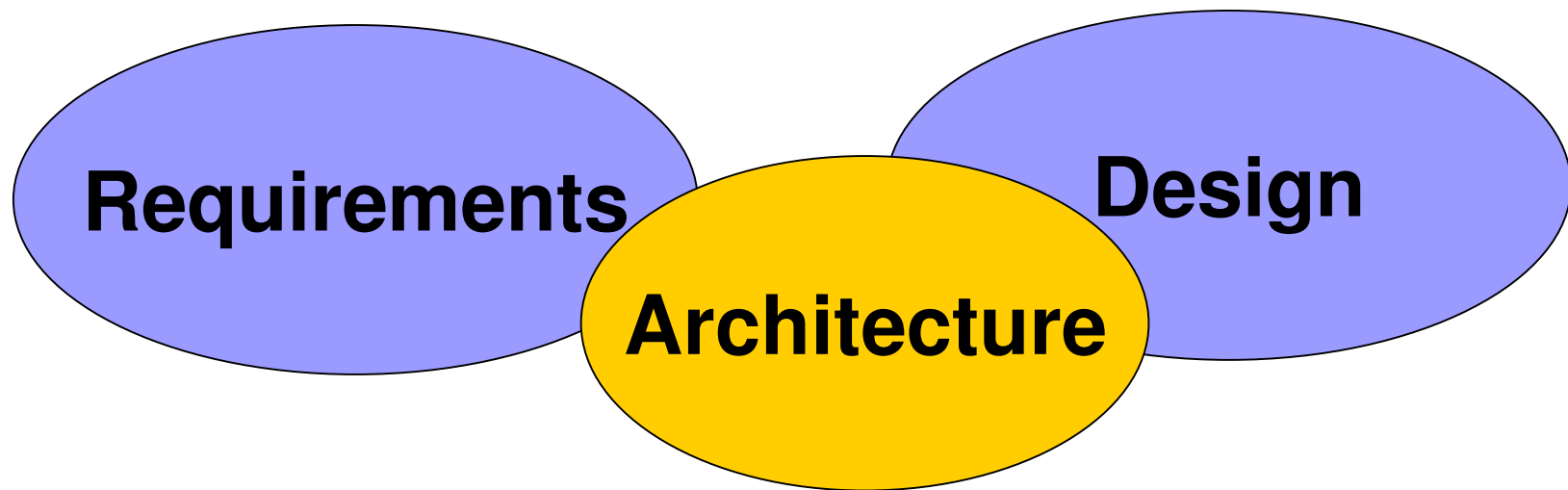
## Background

- Met at Sybase (1994 – 1999)
- Once asked “*what’s the best architecture book?*” - we didn’t have a good answer
- Subsequent research led us to
  - IEEE 1471
  - 4+1, Siemens, RM-ODP, Garland and Anthony
  - Our own viewpoint set
  - Definition of “perspectives” for qualities
  - Forthcoming book

# Timetable

00:00 – 00:10	Introductions
<b>00:10 – 00:40</b>	<b>Overview of viewpoints</b>
00:40 – 00:50	Identify example systems
00:50 – 01:15	Ex1: identify viewpoints
01:15 – 01:30	Break
01:30 – 01:45	Overview of perspectives
01:45 – 02:15	Ex2: identify perspectives
02:15 – 02:30	Review and discussion

# Software Architecture in Context

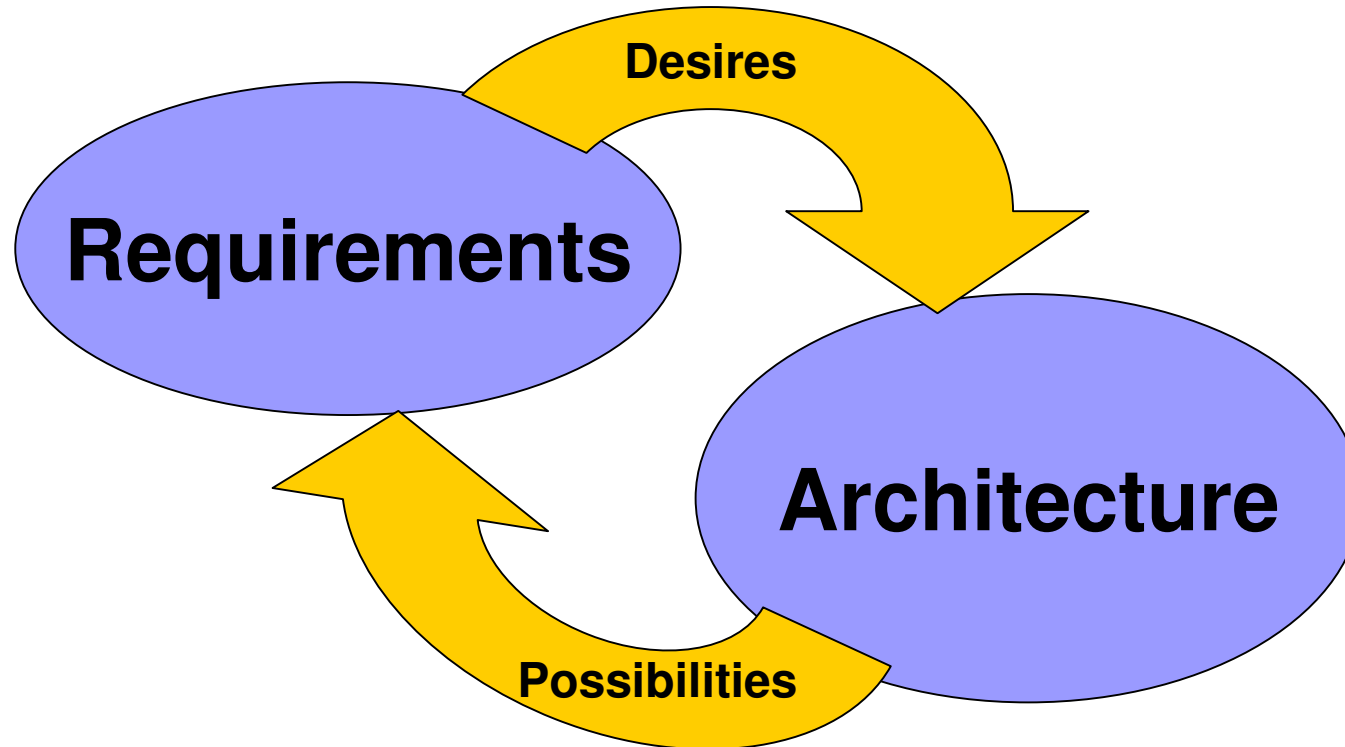


The crucial bridge between requirements and design

# Software Architecture in Context

- Requirements frame the architectural problem
  - Stakeholder needs and desires
- Yet, architecture influences requirements
  - “The art of the possible”
  - Helps stakeholder understanding of risk/cost
  - Helps stakeholder understanding of possibilities

# Architecture and Requirements



This interplay is core to the architectural process

# Overview of Viewpoints

*The **software architecture** of a program or computing system is the **structure or structures of the system**, which comprise software **elements** the externally visible **properties** of those elements, and the **relationships** among them*

Bass, Clements and Kazman (SEI)  
*Software Architecture in Practice*

# Overview of Viewpoints

- Architecture is about defining structures – not one but many
  - Functional structure
  - Information structure
  - Concurrency structure
  - Design time structure
  - ...

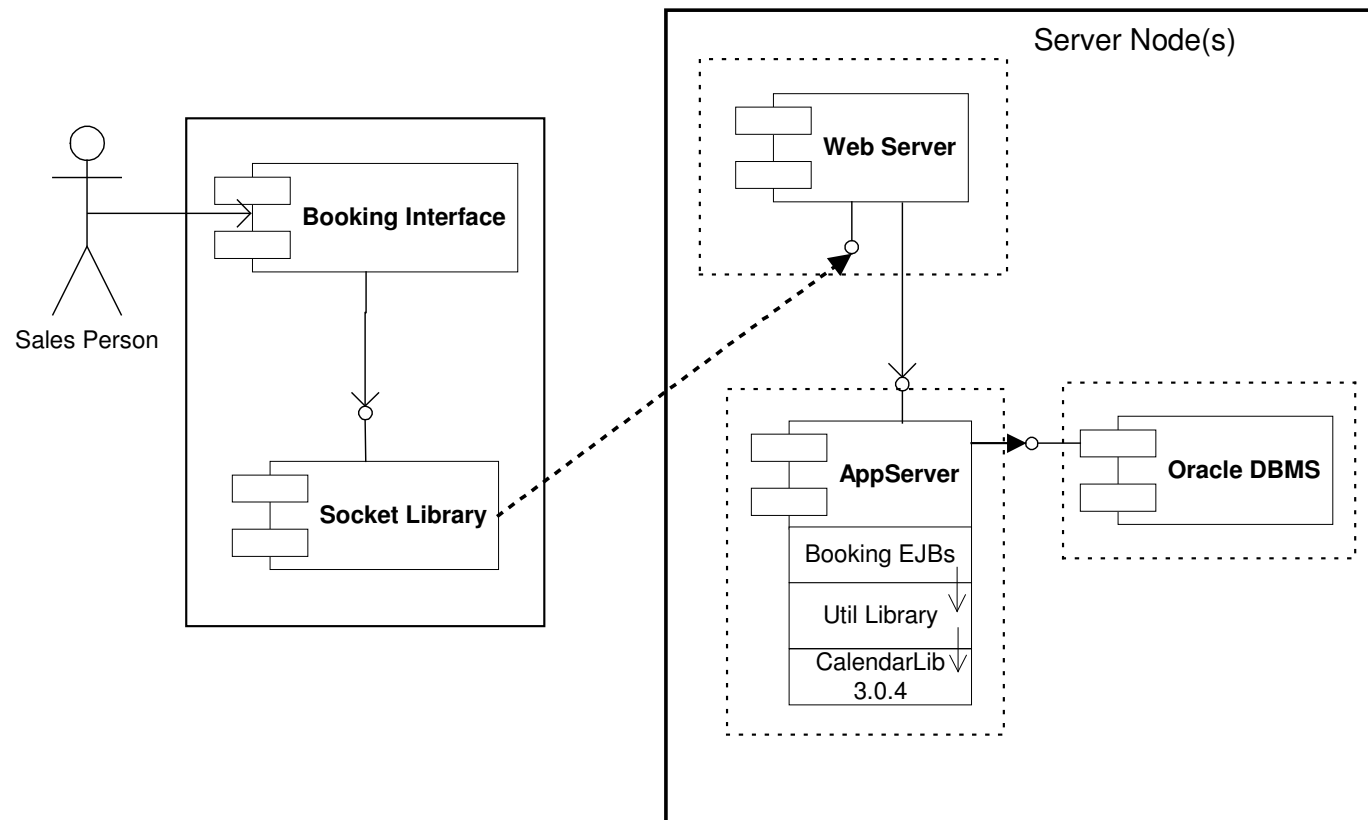
*It's also very much about properties – we'll get to these later*

# Overview of Viewpoints

- Dealing with many structures is hard
  - Organization of ideas
  - Understanding different aspects simultaneously
  - Separating concerns
  - Dealing with different aspects equally
  - Consistency

# Overview of Viewpoints

- Typical representation ...



# Overview of Viewpoints

- Architecture today is largely ad-hoc
  - Little standardisation in description
  - Difficult to compare and discuss alternatives
  - No process for developing architectures
- Need a conceptual framework to
  - Organise the architectural design process
  - Allow classification of ideas
  - Capture knowledge for discussion and reuse

# Overview of Viewpoints

- Use views as the organising abstraction for the design and description

*A **view** is a **representation** of all or **part of an architecture**, from the perspective of one or more **concerns** which are held by one or more of its stakeholders.*

[IEEE Standard 1471 – Recommended Practice for Architectural Description]

# Overview of Viewpoints

- Views are fine but ...
  - Little more than a documentation convention
  - No help for structuring architectural knowledge or guidance
  - No guidance on process
  - Many common pitfalls for the unwary

*We need another concept to work along with views*

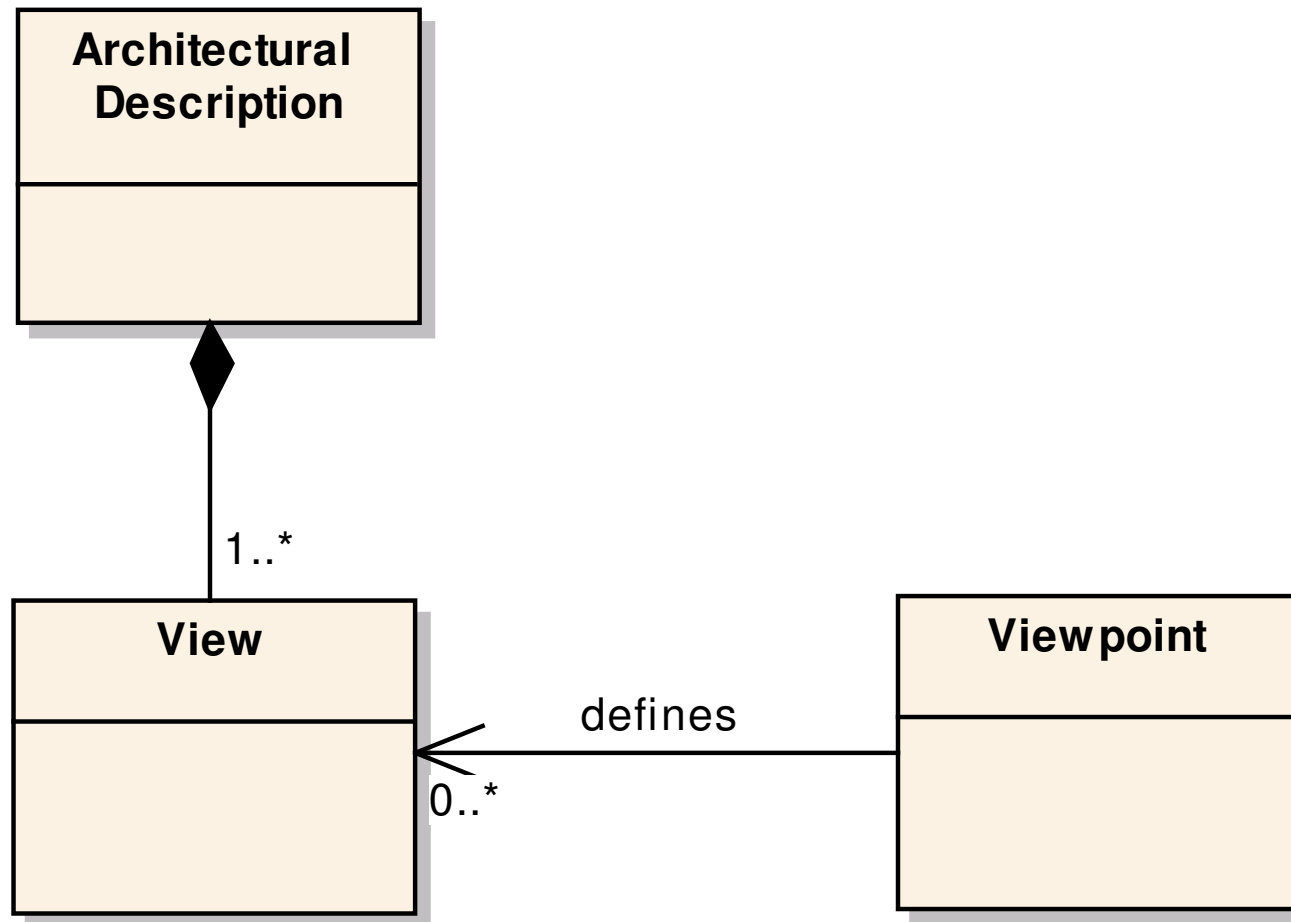
# Overview of Viewpoints

- Use viewpoints as the organising abstraction for the process

*A **viewpoint** is a collection of **patterns**, **templates** and **conventions** for **constructing one type of view**. It defines the **stakeholders** whose concerns are reflected in the viewpoint, and **guidelines** and **principles** and **template models** for constructing its views.*

[IEEE Standard 1471]

# Primary Relationships



# Rationale

- Viewpoints provide
  - A store of knowledge and experience
  - A guide to the architect
  - Templates to guide the process
- Views provide
  - A structure for description
  - A separation of concerns
  - Improved stakeholder communication

# Overview of Viewpoints

- A number of people have suggested ways of using views for architecture
  - Phillippe Kruchten (“4+1”, 1995)
  - Hofmeister, Nord, Soni (“Siemens Set”, 1999)
  - RM-ODP (1995/96)
  - Jeff Garland and Richard Anthony (2003)

All slightly different ideas, leading to some confusion

# Overview of Viewpoints

- IEEE Standard 1471-2000

*“IEEE Recommended Practice for Architectural Description”*

- Standardises the fundamental definitions for the approach
- Rationale, context, content, etc.
- Provides a meta-model so we can all talk about the same things

# Overview of Viewpoints

- Viewpoints generally defined in sets
  - Philippe Kruchten (“4+1”)
    - Logical, Process, Development, Physical
  - Hofmeister, Nord and Soni
    - Conceptual, Module, Execution, Code
  - Rozanski and Woods
    - Evolution/specialisation of Krutchen’s “4+1” set
    - Functional, Information, Concurrency, Development, Deployment, Operational

# Overview of Viewpoints

- Rozanski and Woods set
  - Aimed at modern, large scale, distributed information systems
  - Extension and refinement of Kruchten's set
    - Renamed & evolved “Logical”, “Process” and “Physical”
    - “Information” and “Operational” added
  - Defined the contents of the viewpoints
    - Not just noted their existence

# Overview of Viewpoints

- Content of our viewpoints
  - Concerns
  - Applicability
  - Stakeholders
  - Models (content, notation, activities)
  - Pitfalls and how to avoid them

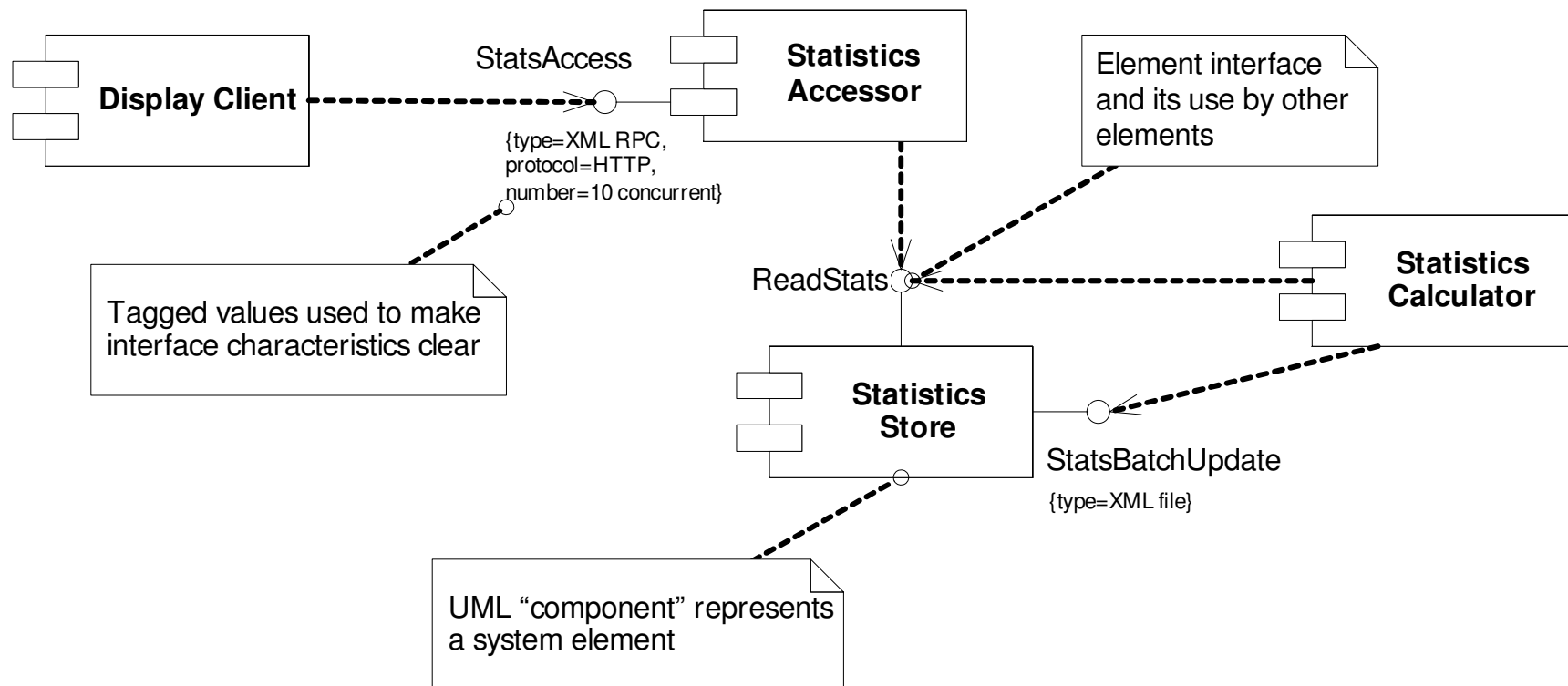
# Overview of Viewpoints

## ■ Functional

- The functional structure of the system
  - Fundamental pieces (components, classes, procedures, systems, ...)
  - Element, interface, interaction
- Functional structure model
- Concerns include
  - Functional coverage
  - Quality of functional structure

# Overview of Viewpoints

## ■ Functional – Example Content Fragment



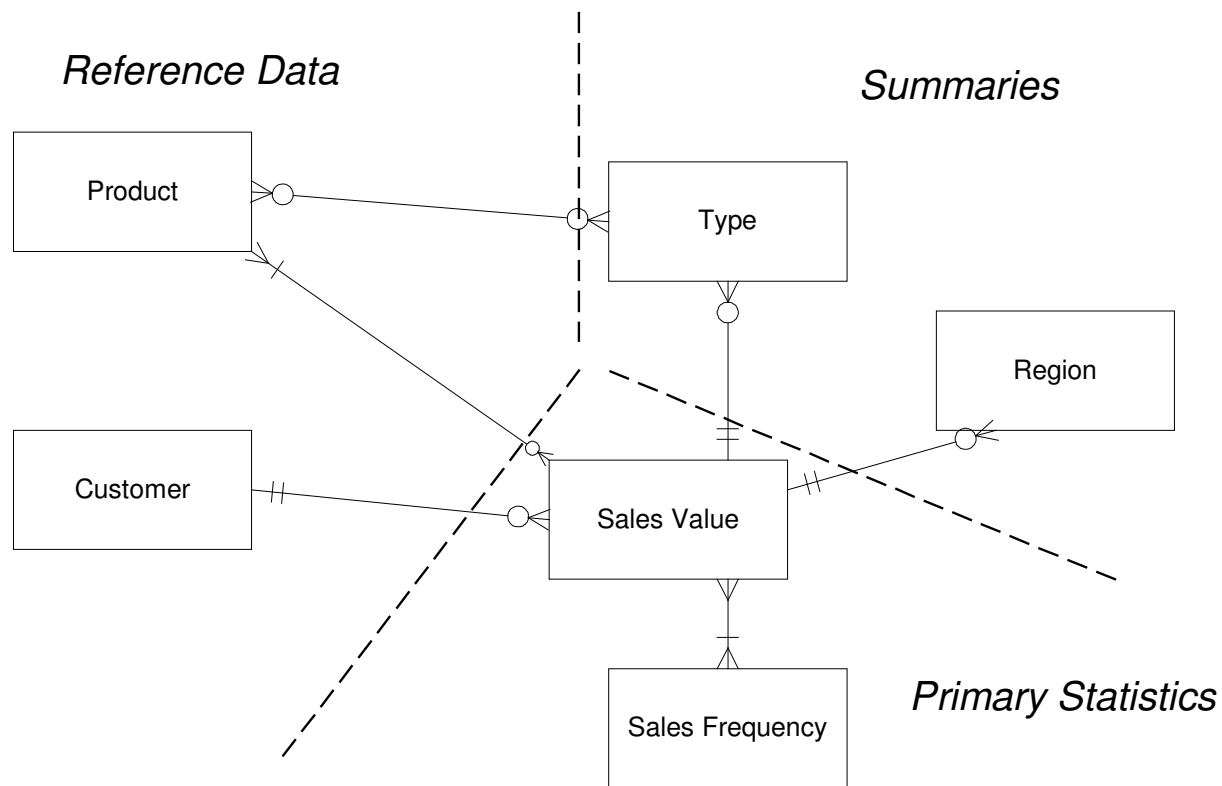
# Overview of Viewpoints

## ■ Information

- The information structure of the system
  - The information stored and how it is used
  - Entity/cluster, relationship, reference, owner, ...
- Augmented high-level data model
- Concerns include
  - Information content and structure
  - Ownership and manipulation of information

# Overview of Viewpoints

## ■ Information – Example Content Fragment



*Architectural static structure*

# Overview of Viewpoints

- Information – Example Content Fragment

	Data Warehouse	Statistics Calculator	Statistics Accessor
Primary Statistics	-	CRUD	R
Reference Data	CRUD	R	R
Summaries	RD	CRU	R

*Data ownership model*

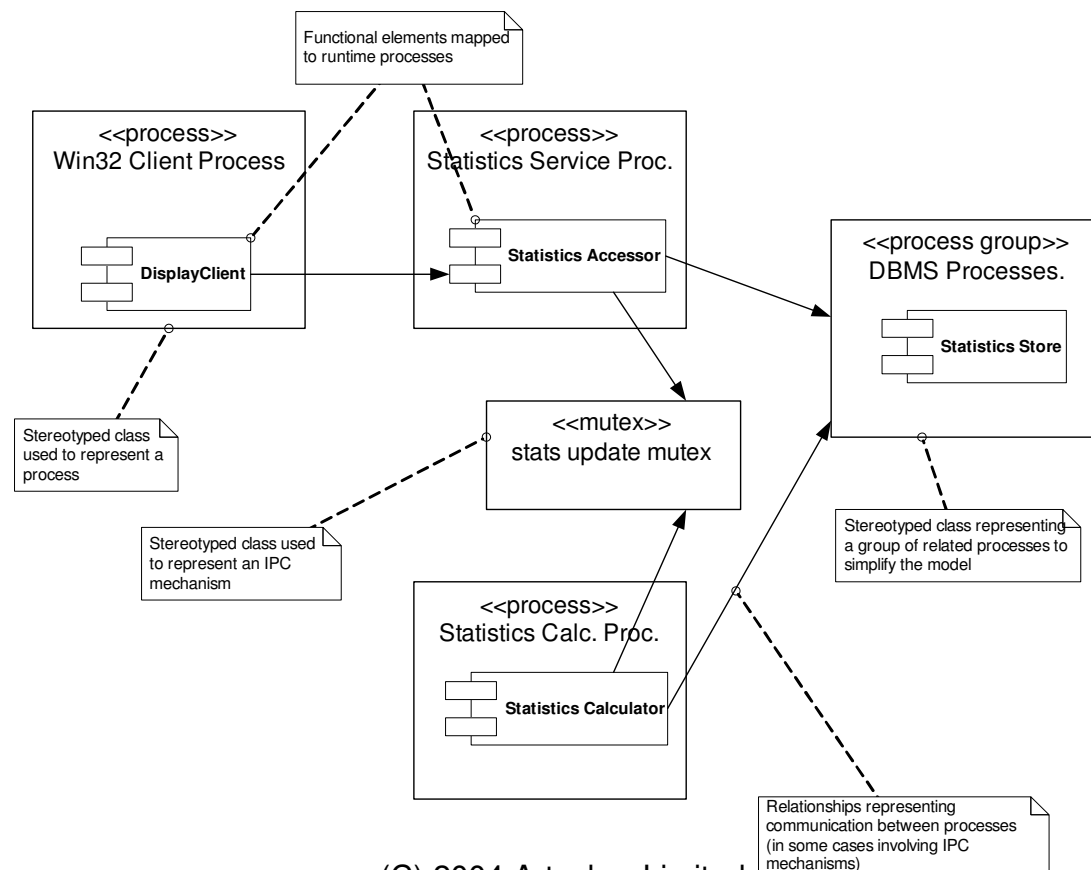
# Overview of Viewpoints

## ■ Concurrency

- The runtime structure of the system
  - Mapping elements to runtime elements
  - Process/thread, IPC, state
- Task and coordination model
- Concerns include
  - Placing elements in appropriate runtime elements
  - Integrity of runtime structure

# Overview of Viewpoints

## ■ Concurrency – Example Content Fragment



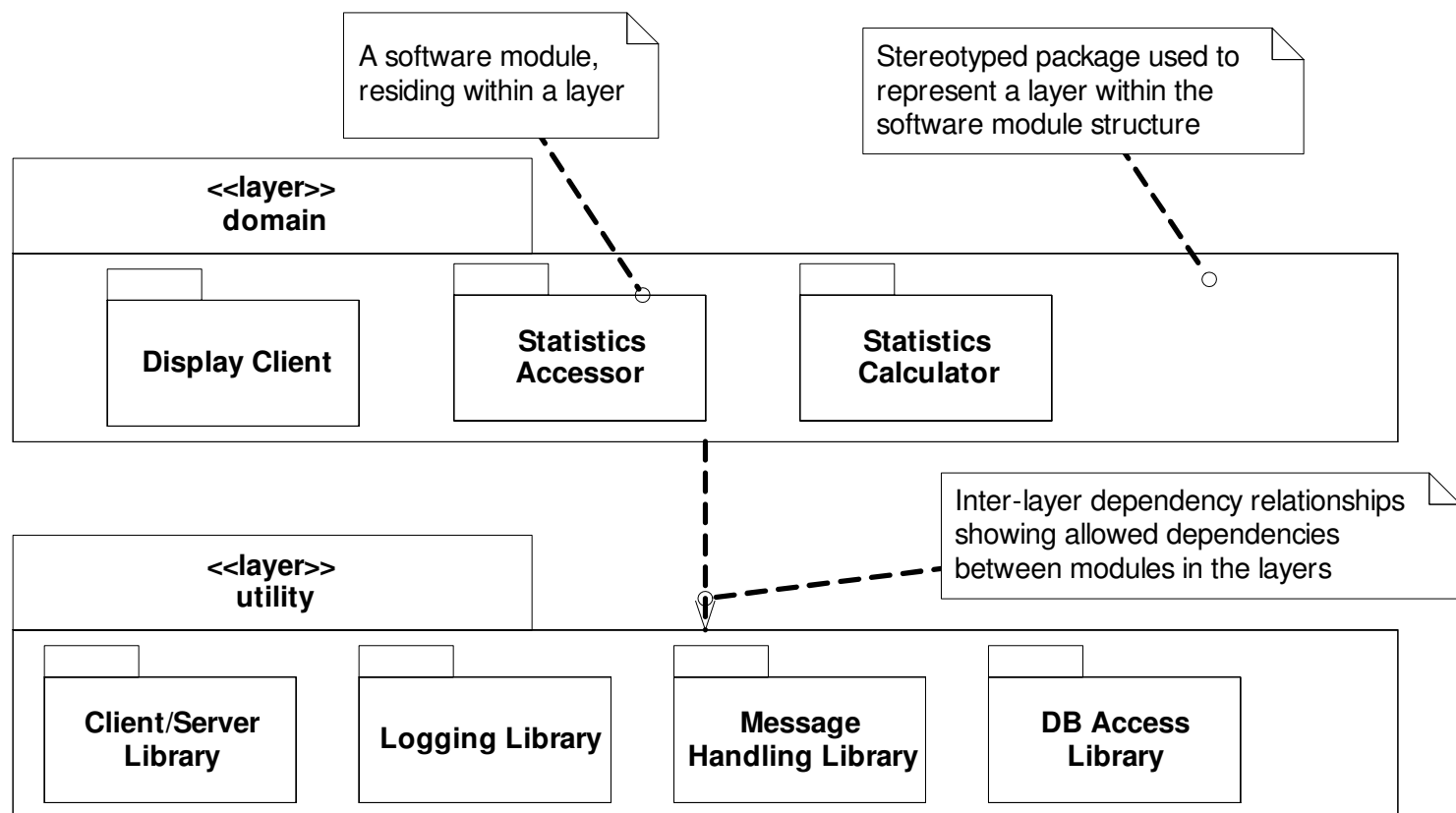
# Overview of Viewpoints

## ■ Development

- The architectural constraints on design/code
  - What must and must not be done
  - The shared development environment
- Module structure, common design & codeline
- Concerns include
  - Common processing and design style
  - Control of development environment

# Overview of Viewpoints

## ■ Development – Example Content Fragment



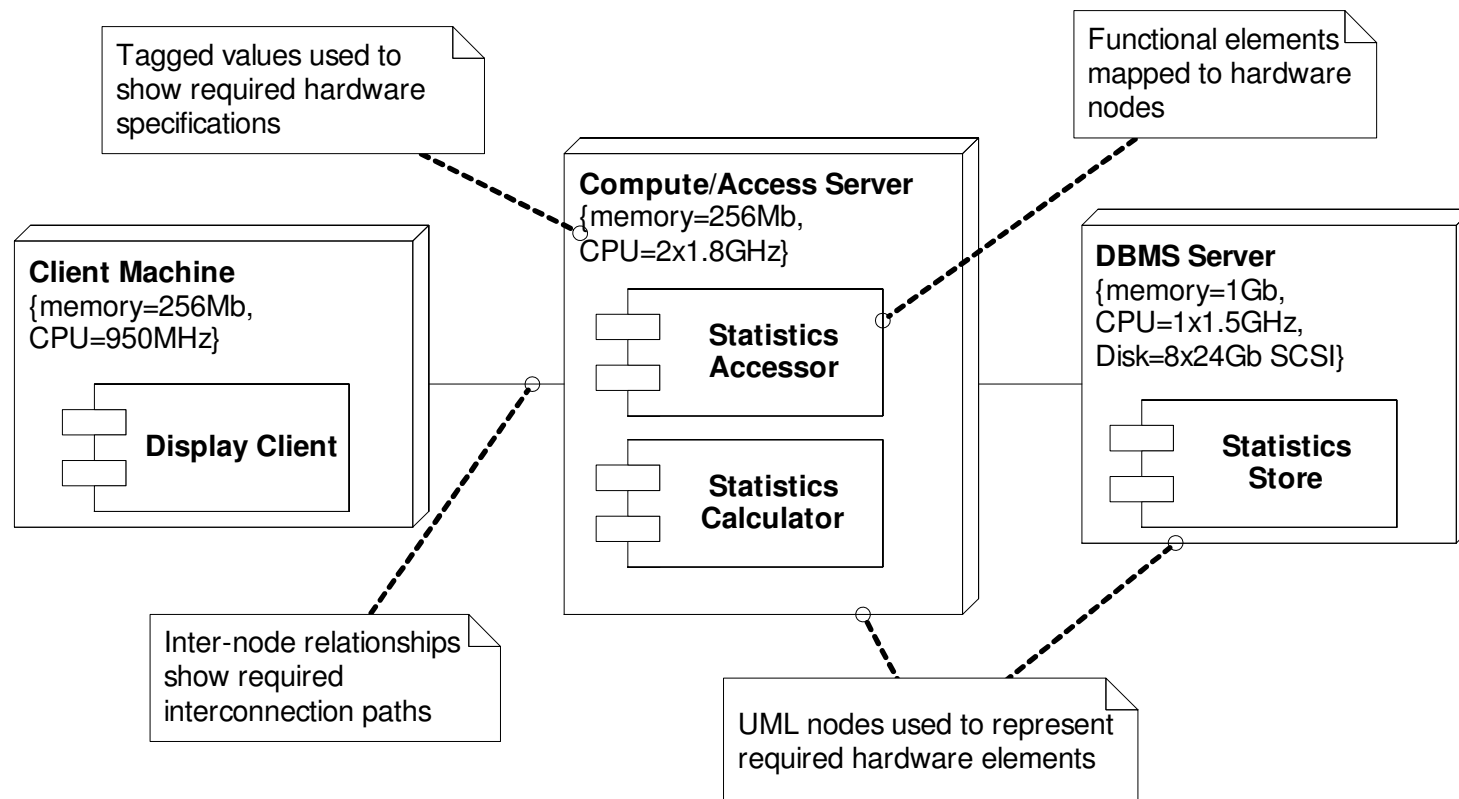
# Overview of Viewpoints

## ■ Deployment

- The production runtime environment
  - Identification of runtime platforms
  - Node, communication link, dependencies
- Platform, network & dependency models
- Concerns include
  - Platform capacity and specification
  - Network link capacity
  - Software stacks required on nodes

# Overview of Viewpoints

## ■ Deployment – Example Content Fragment



# Overview of Viewpoints

## ■ Operational

- The structures required around the system
  - Getting into production, staying reliable there
  - Install elements, tools, dependencies, ...
- Install, migration, CM, admin, support models
- Concerns include
  - Installation, migration and backout
  - Effective monitoring and control
  - Recognising and responding to problems

# Overview of Viewpoints

- Operational – Example Content Fragment
  - Installation, migration and backout strategies
  - Monitoring and Control strategies
  - Operational configuration management
  - Support model (who is responsible for what, escalation)
- All at the architecture, not detail level (constraints & strategies, not procedures)
- Not possible to show a lot here (space)

# Overview of Viewpoints

- Administration of the Statistics Reporting System
  - **Monitoring and Control**
    - *Server Message Logging*: All server components will write information, warning and error messages to the Windows Event Log.
    - *Client Message Logging*: The client software will log messages if an unexpected error is encountered. The log will be written to the hard disk of the client.
  - **Operational Procedures**
    - *Backup*: Data in the database will need backed up. This will involve [...]
    - [...]
  - **Error Conditions**
    - *Database Out of Log Space*: If transaction volume rises above a certain point then it is possible that the transaction log will fill. This will [...]
    - [...]
  - **Performance Monitoring**
    - *Database Counters*: The following performance counters can be [...]
    - [...]

# Timetable

00:00 – 00:10	Introductions
00:10 – 00:40	Overview of viewpoints
<b>00:40 – 00:50</b>	<b>Identify example systems</b>
00:50 – 01:15	Ex1: identify viewpoints
01:15 – 01:30	Break
01:30 – 01:45	Overview of perspectives
01:45 – 02:15	Ex2: identify perspectives
02:15 – 02:30	Review and discussion

# Exercise Preparation

- Each group needs a system to discuss
  - Break into groups
  - 5 – 10 minutes, choose a system one of you has worked on
  - You will use this system as the example during the exercises
  - Ideally a mid-size distributed information system, but this isn't crucial
  - Explain the system to your group

# Exercise Preparation

- For your system capture
  - Context diagram
  - Outline functional description
  - Key quality requirements
    - Security, performance, availability, ...
  - Main functional elements and interfaces
  - Anything else “unusual”

# Timetable

00:00 – 00:10	Introductions
00:10 – 00:40	Overview of viewpoints
00:40 – 00:50	Identify example systems
<b>00:50 – 01:15</b>	<b>Ex1: identify viewpoints</b>
01:15 – 01:30	Break
01:30 – 01:45	Overview of perspectives
01:45 – 02:15	Ex2: identify perspectives
02:15 – 02:30	Review and discussion

# Exercise 1: Identify Viewpoints

- For your example system
  - Identify the views that are most important to describe it
  - Using the VP definitions as a guide, sketch views for your system on flipcharts
  - Identify any missing aspect to the description
    - Where does this go?
    - Identify any views that you think are missing

# Timetable

00:00 – 00:10	Introductions
00:10 – 00:40	Overview of viewpoints
00:40 – 00:50	Identify example systems
00:50 – 01:15	Ex1: identify viewpoints
<b>01:15 – 01:30</b>	<b>Break</b>
01:30 – 01:45	Overview of perspectives
01:45 – 02:15	Ex2: identify perspectives
02:15 – 02:30	Review and discussion

# Break

- 15 minutes

# Timetable

00:00 – 00:10	Introductions
00:10 – 00:40	Overview of viewpoints
00:40 – 00:50	Identify example systems
00:50 – 01:15	Ex1: identify viewpoints
01:15 – 01:30	Break
<b>01:30 – 01:45</b>	<b>Overview of perspectives</b>
01:45 – 02:15	Ex2: identify perspectives
02:15 – 02:30	Review and discussion

# Overview of Perspectives

- Think back to exercise 1
- What were you doing to identify structures?
  - What the system had to do
  - How system had to do it
- The “how” is referred to as the “quality properties” of the system

# Overview of Perspectives

- Quality properties are the non-functional characteristics of the system (“-ilities”)
  - Performance
  - Efficiency
  - Security
  - Maintainability
  - Availability
  - ...

# Overview of Perspectives

- Quality properties crucial to stakeholders
  - Slow functions don't get used
  - Unavailable systems cause business interruption
  - Security problems cause headlines
  - Unmaintainable systems become irrelevant

*Yet viewpoints don't help at all with QPs!*

# Overview of Perspectives

- QPs are often an afterthought
  - Often expensive to “retro-fit”
  - Disruption to existing operations
  - May conflict with existing QPs
- Addressing QPs is key architectural task
  - Understanding stakeholder “real” needs
  - Trading off between conflicting needs
- Need a framework for thinking about QPs

# Overview of Perspectives

- Use *perspectives* as the abstraction for dealing with quality properties

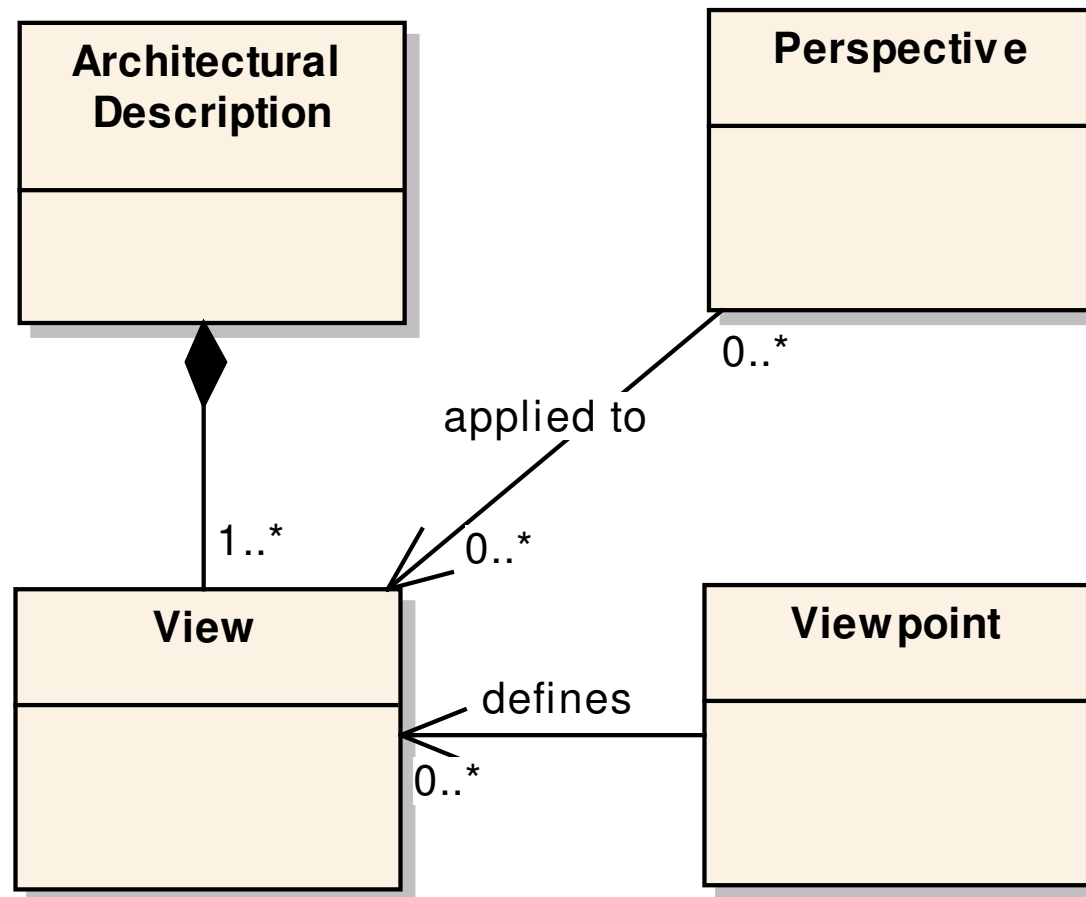
*An architectural **perspective** is a collection of **patterns, templates and guidelines** to guide the process of ensuring that a **system exhibits** a particular set of closely related **quality properties** that require consideration across a number of the system's views.*

[Rozanski & Woods]

# Overview of Perspectives

- A simple idea, but captures the key ideas
  - A store of knowledge and experience
  - A guide to the architect
  - Templates to guide the process
- Analogous to viewpoints but for QPs
- You *apply* perspectives to the architecture to ensure QPs are acceptable and guide its development

# Overview of Perspectives



# Overview of Perspectives

- Our initial core set
  - Performance and Scalability
  - Security
  - Availability and Resilience
  - Evolution
  - Also: Location, I18N, Usability, Regulation, ...

*These will be different for different domains*

# Overview of Perspectives

- Content of our perspectives
  - Desired qualities
  - Applicability
  - Concerns
  - Activities
  - Architectural strategies
  - Pitfalls and how to avoid them

# Overview of Perspectives

## ■ Performance and Scalability

- *Concerns*: processing volume, response time, responsiveness, throughput, predictability
- *Techniques*: performance requirements definition, performance modelling, workload characterisation

# Overview of Perspectives

## ■ Security

- *Concerns*: authentication, authorisation, confidentiality, integrity, accountability, availability, intrusion detection, recovery
- *Techniques*: threat identification, threat assessment, vulnerability analysis, application of security technology

# Overview of Perspectives

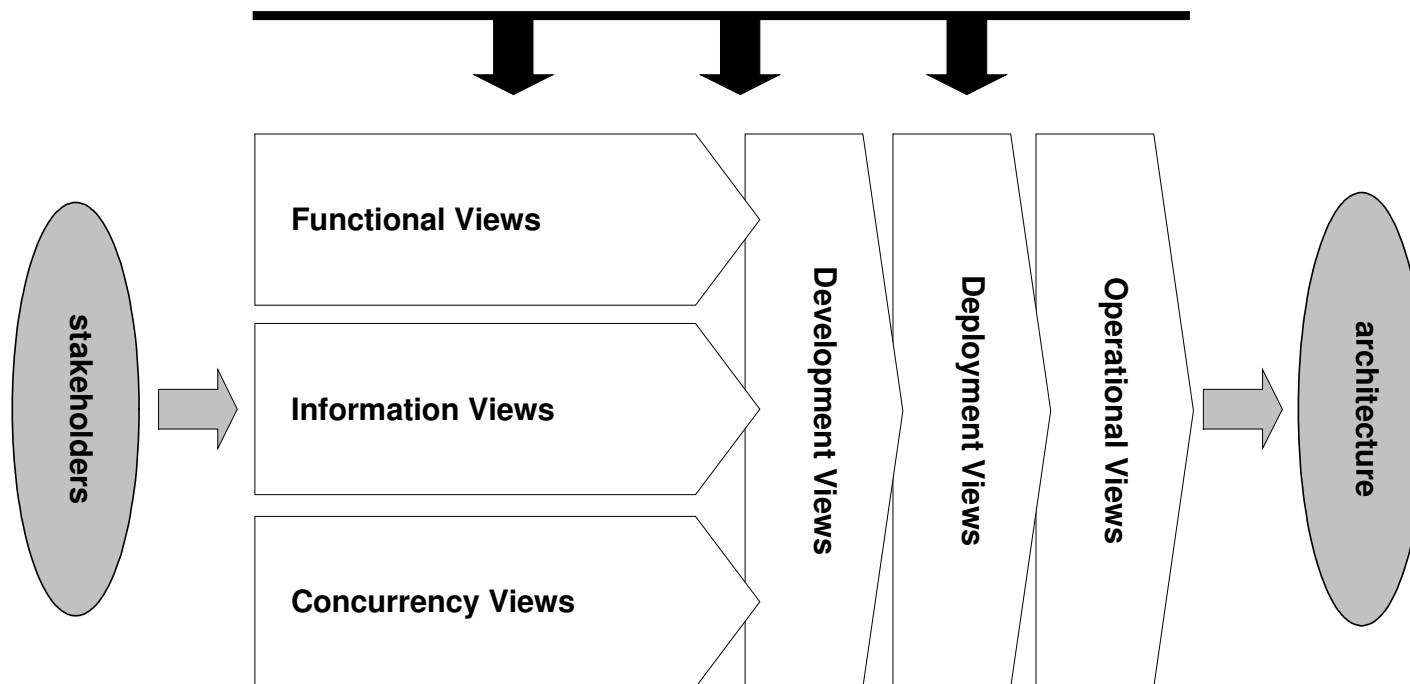
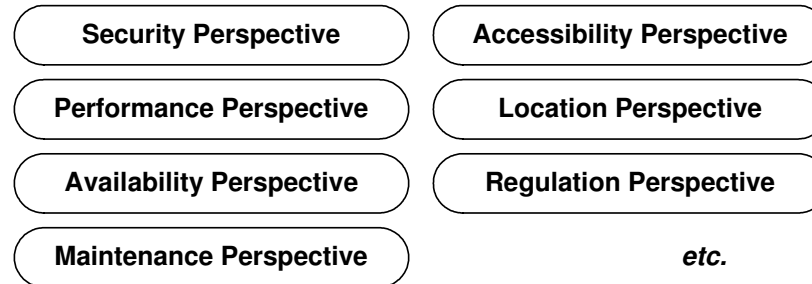
- Availability and Resilience
  - *Concerns*: classes of service, planned / unplanned downtime, mean time between failures, mean time to repair, disaster recovery, redundancy, clustering, failover
  - *Techniques*: MTBF and MTTR prediction, availability schedules, availability models, availability technology application

# Overview of Perspectives

## ■ Evolution

- *Concerns*: flexibility, extensibility, functional evolution, deployment evolution, integration evolution
- *Techniques*: design for change, architectural assessment, configuration management, automated testing, build and release management

# Overview of Perspectives



# Timetable

00:00 – 00:10	Introductions
00:10 – 00:40	Overview of viewpoints
00:40 – 00:50	Identify example systems
00:50 – 01:15	Ex1: identify viewpoints
01:15 – 01:30	Break
01:30 – 01:45	Overview of perspectives
<b>01:45 – 02:15</b>	<b>Ex2: identify perspectives</b>
02:15 – 02:30	Review and discussion

## Exercise 2: Identify Perspectives

- For your example system
  - Identify the perspectives that are most important to consider for it
  - Using the perspective definitions as a guide, sketch views the affect of applying them
  - Identify any missing QP considerations
    - Where are they addressed?
    - Identify any perspectives that you think are missing

# Timetable

00:00 – 00:10	Introductions
00:10 – 00:40	Overview of viewpoints
00:40 – 00:50	Identify example systems
00:50 – 01:15	Ex1: identify viewpoints
01:15 – 01:30	Break
01:30 – 01:45	Overview of perspectives
01:45 – 02:15	Ex2: identify perspectives
<b>02:15 – 02:30</b>	<b>Review and discussion</b>

# Review and Discussion

- Do we all have the same VP and P sets?
  - Why / why not?
- What VPs and Ps still need to be defined?
- How effective are VPs?
- How effective are Ps?
- What is missing/could be improved?

# Review and Discussion

*Thanks for coming to the session, we  
hope you found it useful and  
interesting.*

*Let us know how you get on with  
viewpoints and perspectives.*

