Software Architecture

Designing Architecture

BITS Pilani

Vijayarajan

Contents

- Recap
- Designing Architecture #1 (will continue in the next class also)

Assignment & Quiz 1

innovate achieve lead

A1 - Build an Architecture for an App

- The App should at minimum include the technologies Web, Mobile, IOT, cloud and analytics.
- Each team will select an application and get the approval of the TA.
- Duplication May not be allowed …
- The submitted architecture will be evaluated by the TA
- The team will be evaluated for their developed architecture

A2 – Research paper on real life architecture / latest trends etc

- Each team has to select a topic and get the approval of the faculty.
- Duplication may not be allowed.
- Final Paper should be submitted as per the agreed upon template and schedule

Assignment & Quiz 1

innovate achieve lead

Quiz 1

The quiz will not be available until Sunday, 14 February 2021, 12:00 AM

This quiz will close at Sunday, 21 February 2021, 11:59 PM

20 Questions - 30 minutes

Once you start you should complete in 30 minutes….

Example:

Resynchronizing the state of a repaired component with the current state of operation and then re-introducing this component is the tactic for improving which of the following attribute.

A. Security
B. Performance
C. Testability
D. Availability

Evolution of SW Architecture

We design and implement information systems to solve problems and process data.

As problems become larger and more complex and data becomes more voluminous, so do the associated information systems

- Structured programming, Data Structure, Higher Level languages, software engineering, Object Oriented etc

Computing become Distributed, on the cloud, Mobile as a front end

As the problem size and complexity increase, algorithms and data structures become less important than getting the right structure for the information system.

Specifying the right structure of the information system becomes a critical design problem itself

< Example from Construction Industry>

Importance of Quality attributes & Tactics

- Functional requirements help us to define the modules
- Quality attributes help us to **structure** the system

Availability

Modifiability

Performance

Security

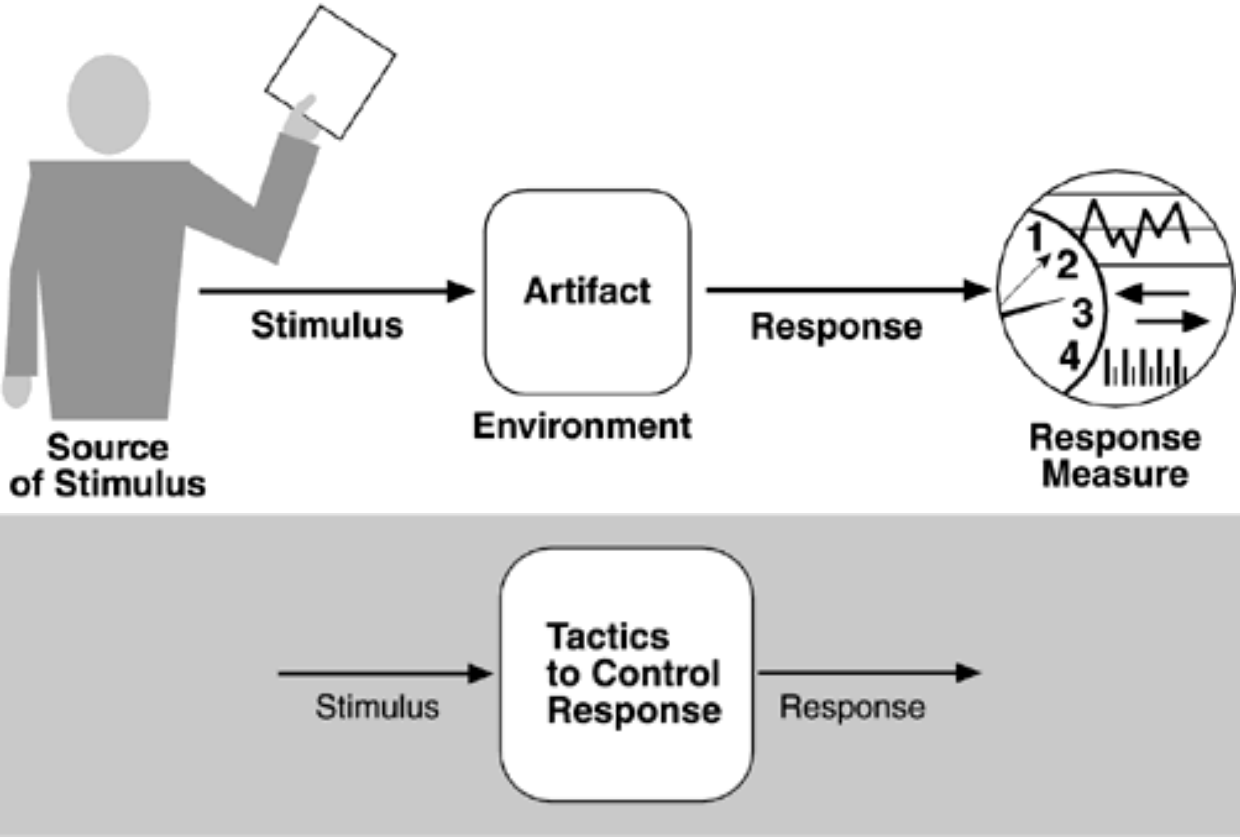
Usability

Interoperability

Scalability

Testability

Architecturally significant requirements



What is a Pattern

1. Addresses a recurring design problem
2. Documents existing, well proven design experience
3. Pattern identify and specify abstractions that are above the level of single classes and instances or of components

   Typically, a pattern describes several components, classes or objects, and details their **responsibilities** and relationships, as well as their **cooperation**.

   All components together solve the problem more effectively that the pattern addresses

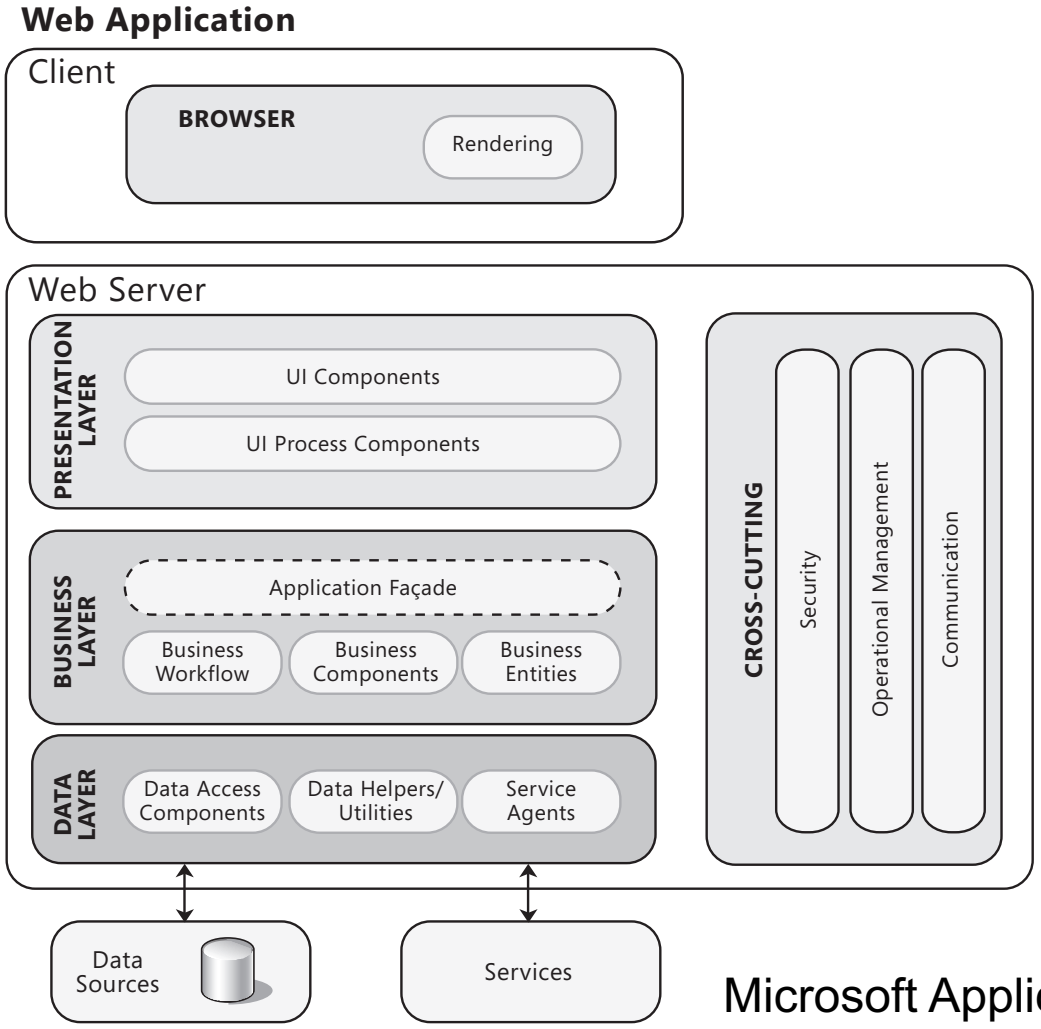
Because patterns are (by definition) found repeatedly in practice, one does not invent them; one discovers them.

List of patterns

1. Layer
2. Pipe & Filter
3. MVC
4. Publish & Subscribe
5. Client & Server
6. P2P
7. Shared Data
8. Broker
9. Map-Reduce
10. Multi-tier
11. SOA

Reference Architecture

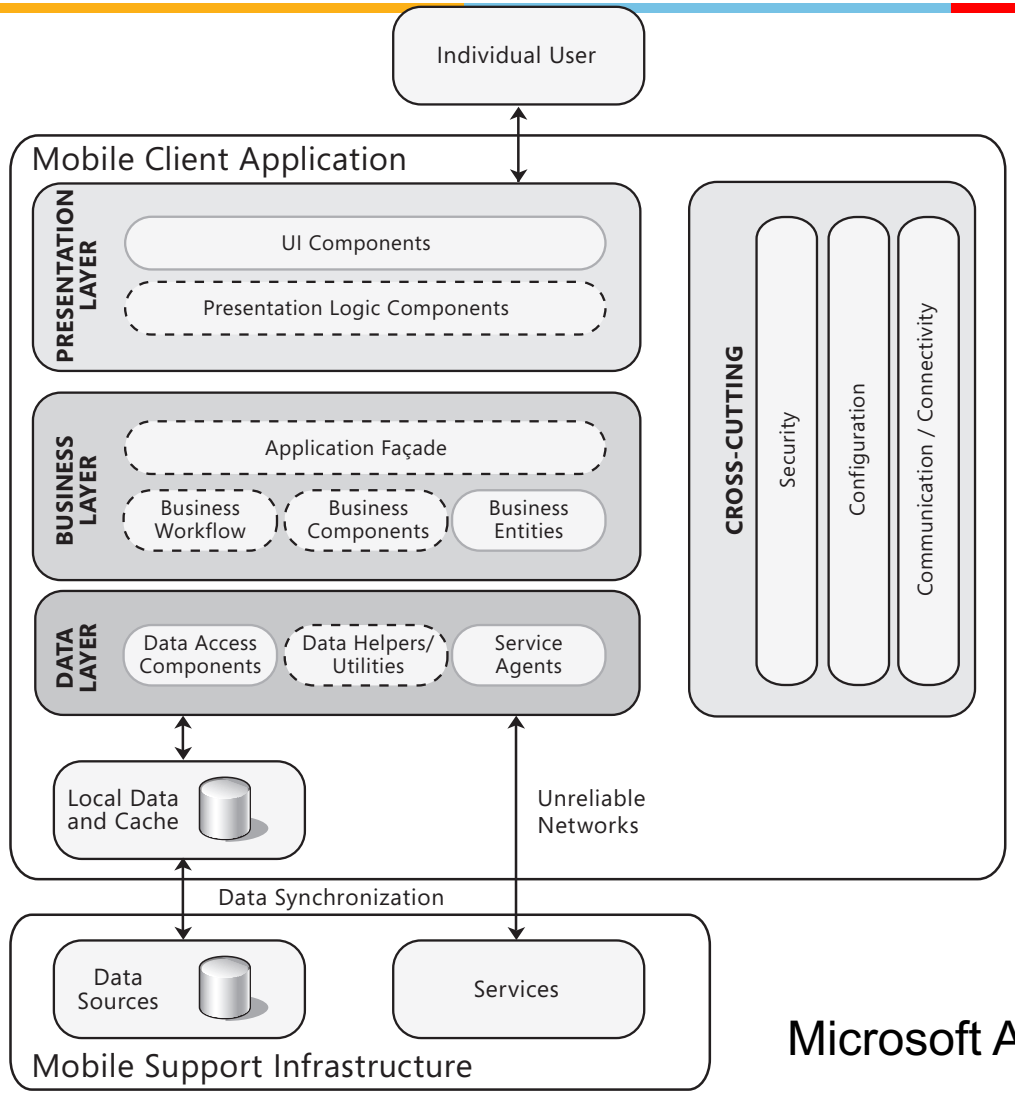
Web Application



Go to Page 278

Microsoft Application Architecture Guide

Mobile Application



Go to page 371

Microsoft Application Architecture Guide

IBM Web Page for IOT

Amazon Reference Architectures - IOT

Designing the Architecture

- My Experience

Drivers for Architecture Design

1. **Design purpose** – what we want to achieve

Proposal (mature / new domain)

Prototype (explore),

Development of new (Green Field Matured /Novel, Enhancements)

Development organizations future extension, scalability, re use, skill set, technology etc

2. **Quality Attributes** – obvious – priority, **utility tree** (Page 307 book), biz importance vs technical risk / complexity
3. **Functionality** – 10% of uses cases are primary

Allocation of responsibilities - promote modifiability, usability

use case -> QA (Movie streaming application)

Drivers for Architecture Design

4. **Architectural concerns (**not generally articulated, learnt over experience**)**

General – Structure, module to teams, start / Stop, organization of codebase, Delivery, deployment, Updates, server capacity

Specific – logging, authentication/authorization

Derived requirements – hierarchy of requirements; it has to work in outdoors – temp range … transport

Issues: Things came up during the last review

Generally, results in additional Quality Attributes

5. **Constraints**

On which you don't have control – schedule, team, backward compatibility, tools, Compliance (industry, company)

Types of Systems

1. Design of Greenfield Systems for Mature Domains

iteration	Focus	How do we go about
	Initial Overall System	Reference Architecture; external components; Deployment patterns; Best Practices
	Primary Functionality	Architectural patterns (domain objects); external components
	Refine & additional drivers	Tactics; architectural patterns, deployment patterns; external components

2. Design of Greenfield Systems for Novel Domains

   Reference Architecture ? Patterns & Tactics can be used - Prototyping option

3. Design for an Existing System (Brownfield)

   Extension, Refactoring, Many Constraints

Architecting… #1

Identification of Design Concepts

Design Decisions - Selecting the Design Concept

Design Concepts

The Building Blocks for Creating Structures

Where do they come from:

Proven methods, reference architecture, Patterns, tactics, externally developed components - products (SQL DB), library, framework, Platform (java, .net)

Past experience, Best Practices, Expert Knowledge

For a specific problem, one will combine different types of design concepts. For example, for security driver, one can use a security pattern, a tactic, a framework, or some combination of these.

Look at alternatives…

Design Decisions

Design Process is making Decisions

Making decision is a Process

Make Candidate Decisions (Critical step)

choose one among them after (preliminary, High level) analysis

- Layers - how many layers, communication, distribution of responsibilities
- Cluster - how many servers, load balancers, physical location, network connectivity

Architecting … #2

Produce the Structure (Module, C&C, Allocation)

- Instantiating Elements (may require customization?)
- Associating Responsibilities
- Establishing Relationships Between the Elements

Defining Interfaces

- External Interfaces (could be a constraint, standard)
- Internal Interfaces (sequence Diagram, Control & Data flow)

Preliminary Documentation

Recording Sketches of the Views

Recording Design Decisions

Tracking Design Progress

Use of an Architectural Backlog

Attribute-Driven Design

Purpose , Quality Attributes , Functionality, Architectural concerns , Constraints

1. Review Inputs
2. Establish iteration Goal - Selecting drivers
3. Choose one or more elements of system to refine
4. Choose design concept(s) that satisfy the drivers
5. Instantiate the architecture, allocate responsibilities and define interfaces
6. Sketch views to record design decisions
7. Analyze the current design against the iteration goal - refine

Iterate …