BITS Pilani

Software Architecture

Designing an architecture

Viswanathan Hariharan
Edited by Vijayarajan

Contents

- Recap
- Designing Architecture #2 (part 1 was covered in the last session)
- Documenting the architecture
- Architecture Evaluation - ATAM Method

Assignment & Quiz 1

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

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

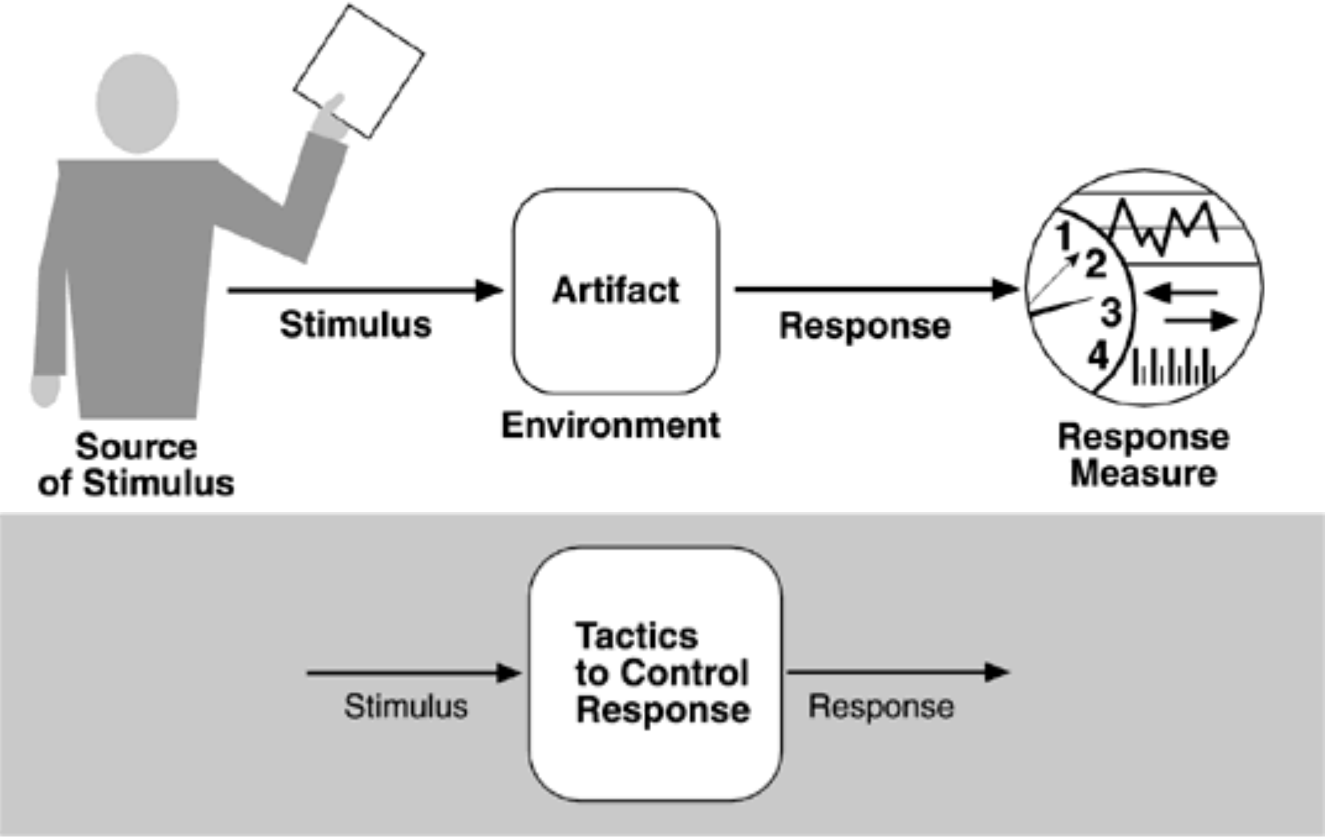
Usability

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 - Microsoft, AWS, IBM

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

- 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
- Making decision is a Process, Make Candidate Decisions then refine

Attribute-Driven Design

Purpose , Quality Attributes , Functionality, Architectural concerns , Constraints

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 …

Example: Judiciary system

Step 1: Choose an element

- Since we are designing a new system, the entire system is our element

Business / Mission presentation by sponsor

Example: Judiciary Management System

Business Mission

- To speed up the settlement of court cases

Motivation

- Huge backlog
- Complex process

User categories

- Judges,
- lawyers,
- court staff and
- citizens

Functional requirements	Non functional requirements
• Filing case • Scheduling • Record court proceedings • Record judgements	• Security of info. • No deletion • Logging

Scenario brainstorming - snapshot

Judge:

- The proceedings should not be lost or tampered with, by unauthorized people, including those working in the court (Security)

Court staff

- Proceedings will consist of text, photos, images & videos (Store pictures)

Citizen

- The system should clearly indicate the steps to file a case (Ease of use)

Lawyers

- I should be notified when the proceedings are posted (Notification)

Scenario refinement & elaboration

Scenario: The proceedings should not be lost or tampered with by unauthorized people, including those working in the court

Scenario Refinement & Elaboration

- Proceedings once recorded should not be lost even in case of disk crash
- Proceedings can be viewed by concerned lawyers, judge and court staff
- Only judge can modify the proceedings and these changes should be logged

Utility tree: Understanding business value & Impact on architecture - snap shot

Quality attribute	Attrubute refinement	Scenrario	Business value	Architecture impact
Security	Integrity	The proceedings should not be lost or tampered with by unauthorized people, including those working in the court (1)	High	High
Usability	Workflow	Once the staff enter the proceedings, it should come to the concerned judge for review, changes and approval (3)	High	High
Modifiability	Criteria specification	The cases should be intelligently scheduled considering the age of the case, criticality, etc. (4)	Medium	Medium
Interoperability	Status notification	The system should send notification to the concerned lawyer and parties when hearings are scheduled (5)	High	High
Performance	Response time	Proceedings will consist of text, photos, images & videos (b)	Medium	Medium
Usability	Understanding user model	Information can be categorized as evidence, arguments, facts, etc. System should aid in the entry of such information and make the data entry efficient (2)	High	Low
Usability	Intuitiveness	Registering a case should be very easy (2)	High	Medium
Interoperability	Status notification	I should be notified about the hearing date (5)	High	High
Usability	Understanding user model	I should be able to upload documents (2)	High	Low
Interoperability	Status notification	I should be notified when the proceedings are posted and I should be able to view them in chronological order (2)	High	High
Usability	Intuitiveness	I should be able to file affidavits and petitions online (2)	High	Low

Architectural drivers

Presented by architects & refined by stakeholders

Architect's understanding of architectural drivers (based on discussion with sponsor):

1. Security of information
2. Easy to use
3. Work flow management
4. Scheduling of cases
5. Alerts & notifications

Comments by stakeholders (Judges, Lawyers, Court Staff and representative citizens)

a. Easy to search a case (elaboration of 2)

b. Store pictures & scanned documents and link them to a case

c. Ability to highlight important aspects in proceedings (elaboration of 2)

Identify the ASRs for the element - snap shot

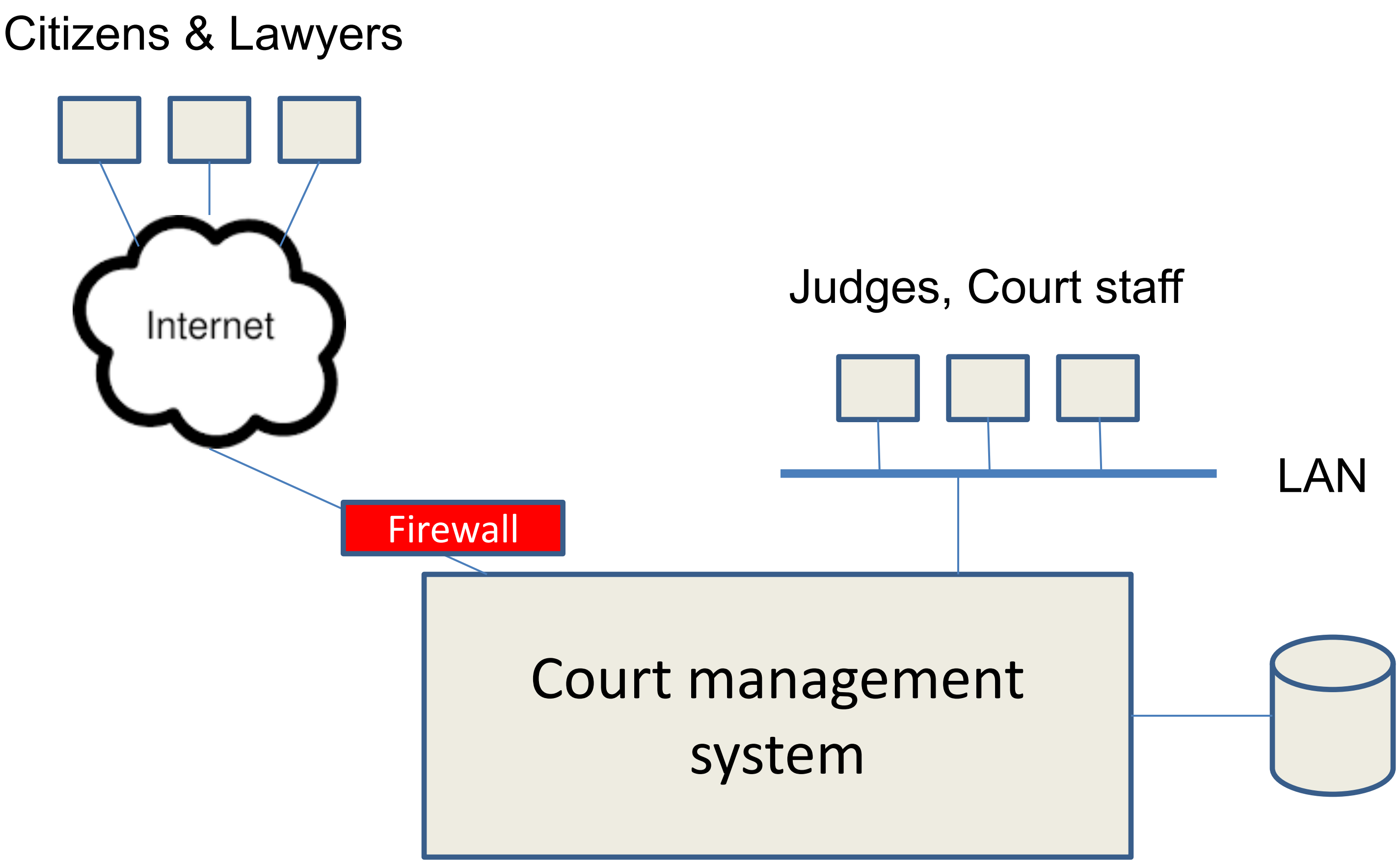
The critical ASRs of the **system** are:

- Security: No tampering
- Usability: Workflow, Notification, Intuitiveness
- Send Notification

Generate a design solution for the element

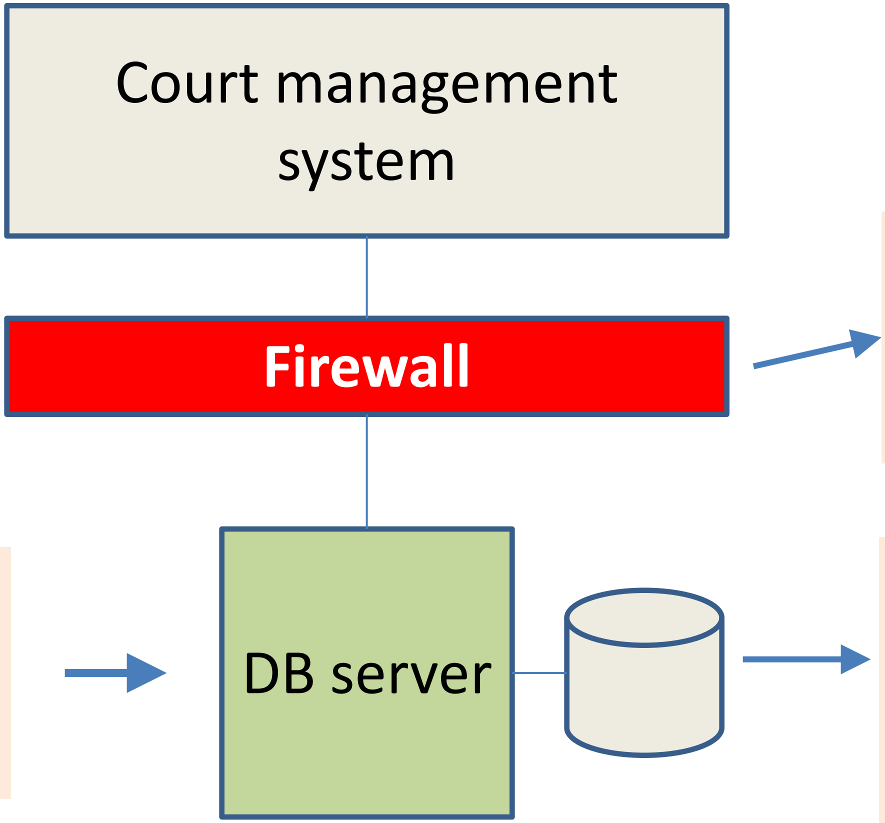
Quality Attribute	Scenario (ASR)	Tactics
Security	The proceedings should not be lost or tampered with by unauthorized people, including those working in the court	Assign access privileges to different types of users such as Judges, Court staff, Lawyers, Citizens
		Encrypt critical data, such as passwords, in the DB
		Store data on a separate database server and protect the server using a second firewall
Usability	Once the staff enter the proceedings, it should come to the concerned judge for review, changes and approval	Include a workflow engine that allows configuration of process steps (Defer binding)
	Registering a case should be very easy	1. Use words that are easy to understand. 2. Provide menu option for different user activities (Understand User’s mental model)
Interoperability	The system should send notification to the concerned lawyer and parties when hearings are scheduled	Interface with SMS gateway of telecom service provider

High level architecture



Now let us try to address the ASRs

Security of data



Allows only the Court management system to access the database

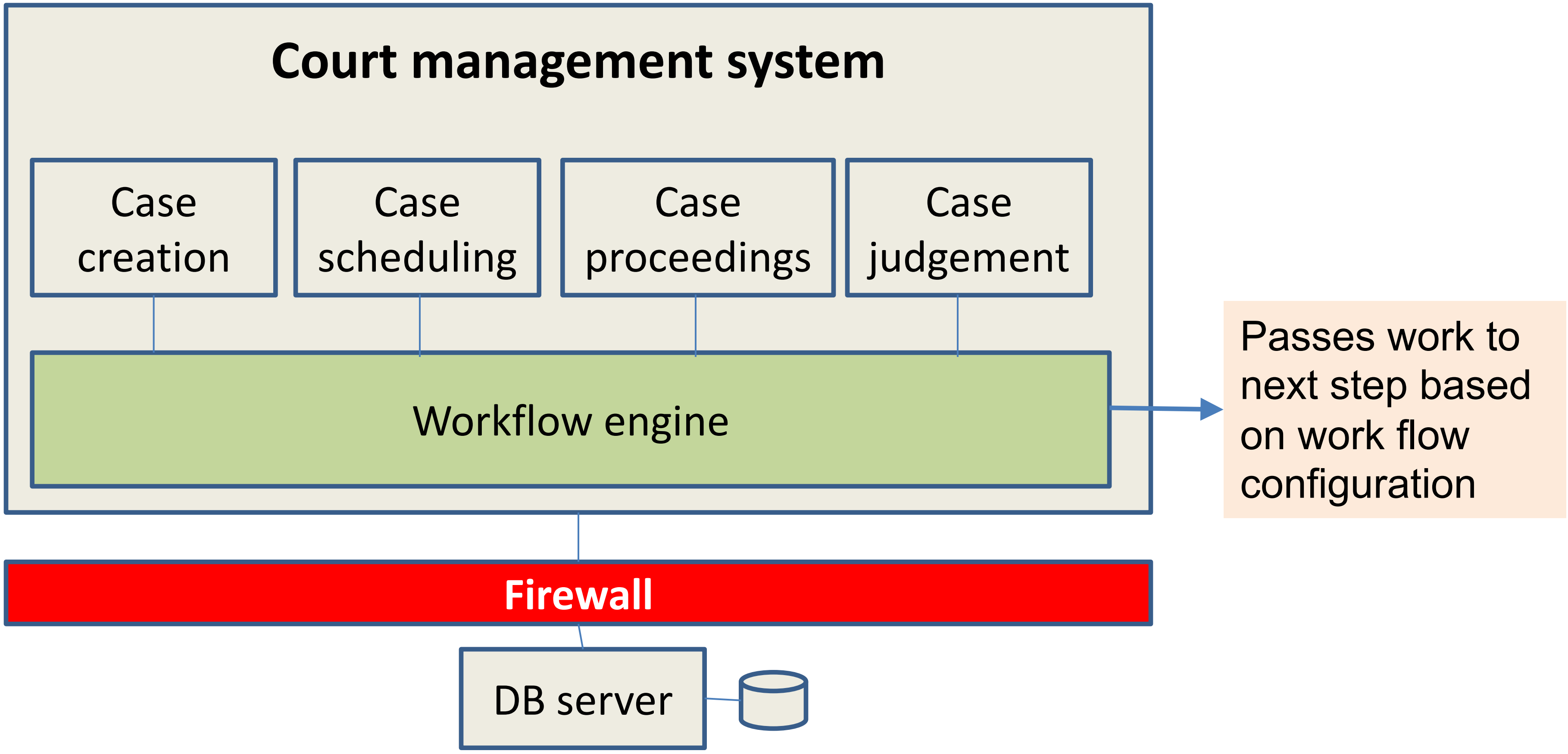
Separate DB server to protect data from being accessed by outside world

Only authorized users can access data

Sensitive data such as judge’s notes are encrypted

Now let us try to address the ASRs

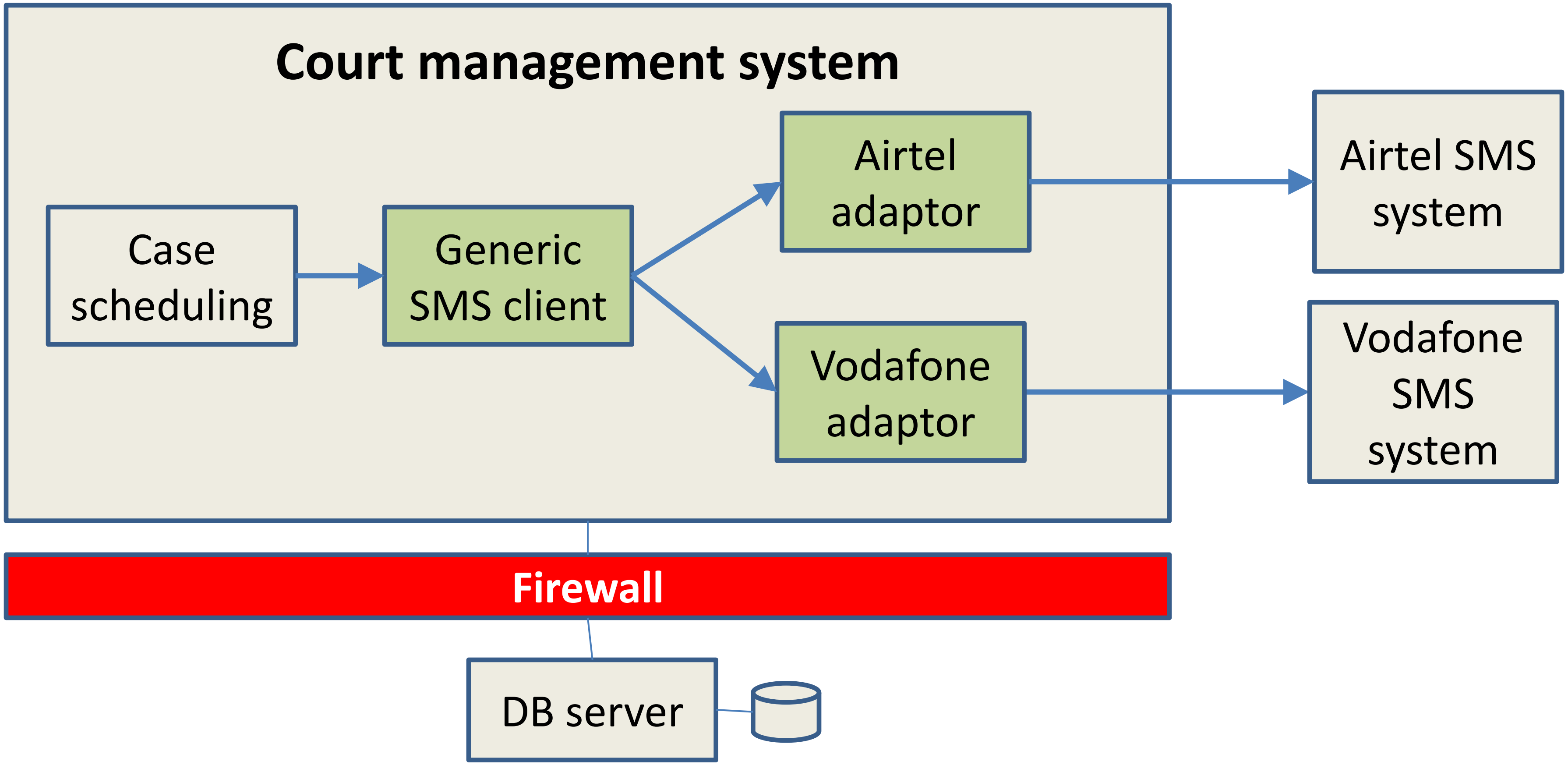
Workflow



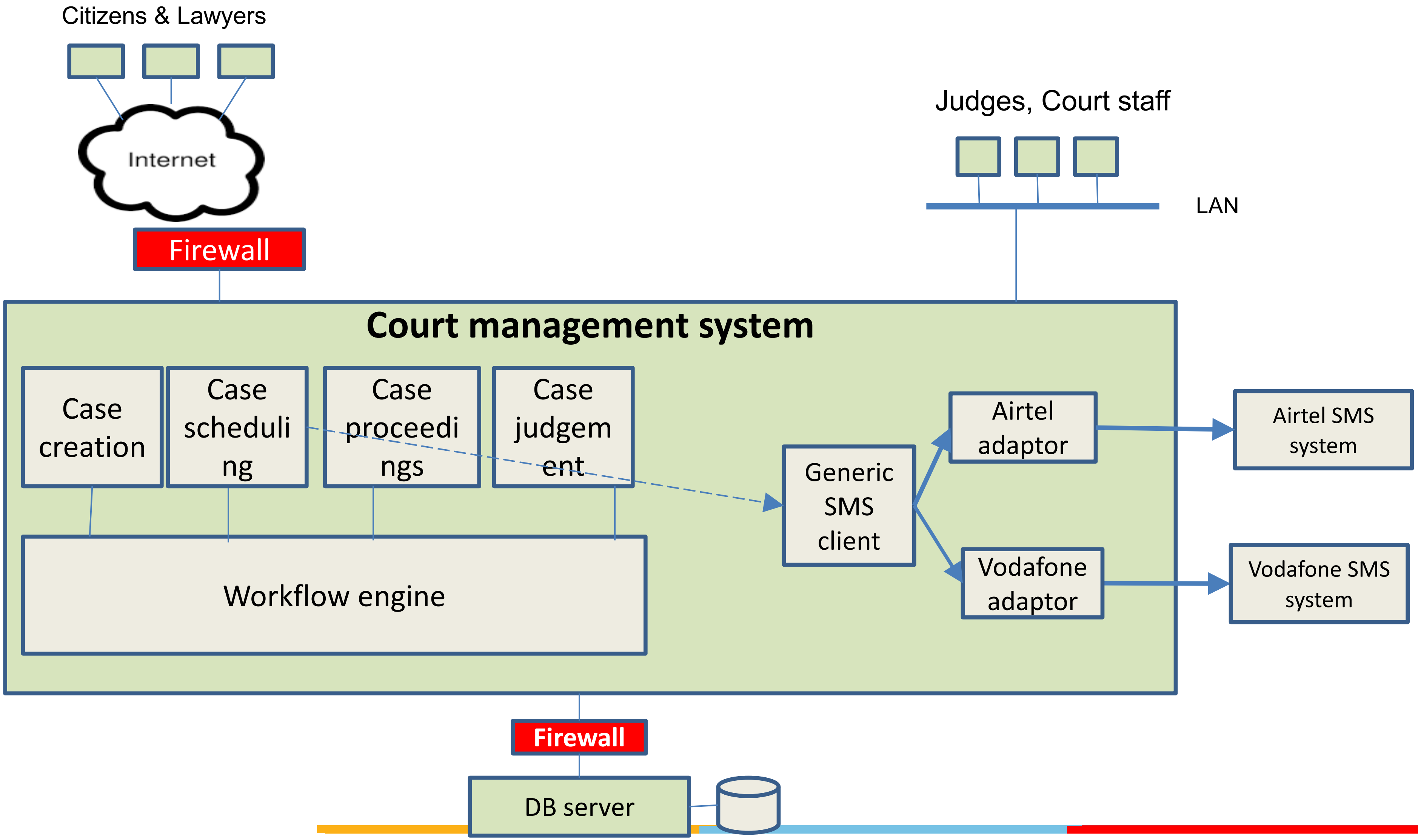
Passes work to next step based on work flow configuration

Now let us try to address the ASRs

SMS notification



Putting it all together



Verify & refine requirements and generate input for next iteration

A. Verify:

- Implement and test if it works as expected

B. Refine requirements:

- Sometimes expectations can not be met. In such cases the requirements may have to be refined in consultation with stakeholders
- Ex. In eCommerce system if Analytics group wants every user action to be logged and if this is severely impacting response time, we can discuss with stakeholders to understand what is critical to be logged and what need not be logged.

Have you come across any situation where you could not meet a requirement and had to discuss with stakeholders to modify the requirement?

Verify & refine requirements and generate input for next iteration

C. Generate input for next iteration:

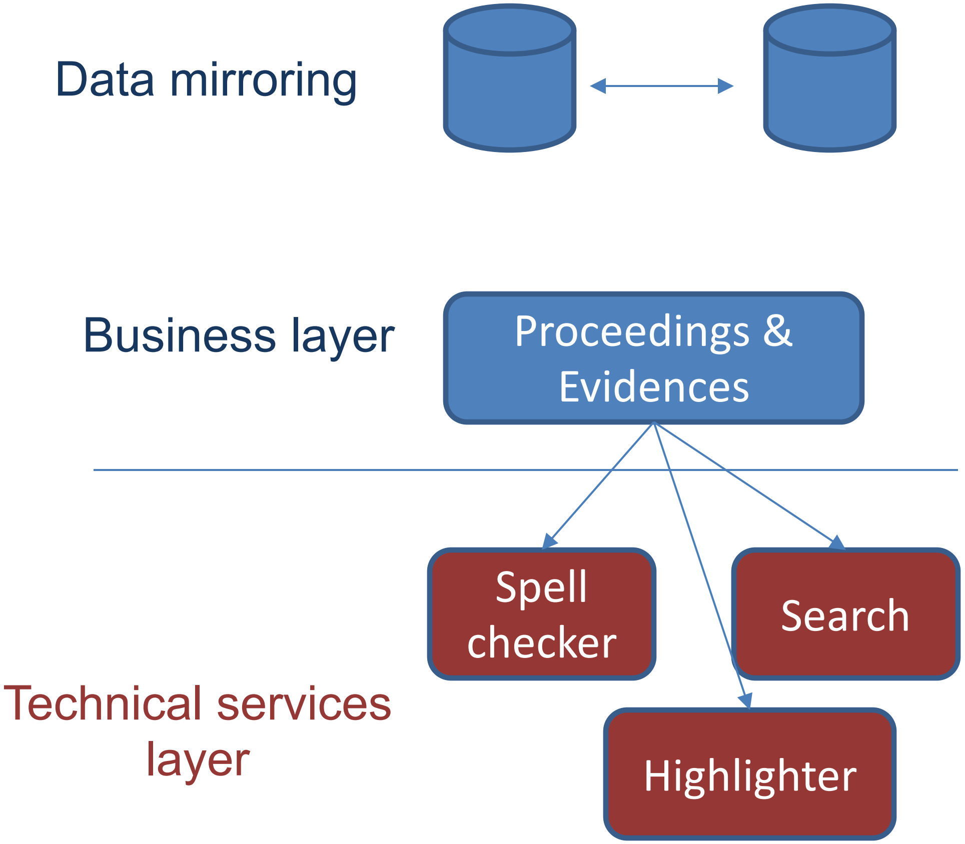
- When you implement the design, new elements might be introduced.

  For example, while designing access control for users, we may decide to use a central access control component, which already exists in the organization. It is possible that based on our requirements, the existing access control module may have to be enhanced.

Example: Architecting one module

Module: Proceedings & evidences

- Replicate data
- Create roles and access privileges
- Usability enablers
  - Automatic spell checks
  - Search keywords
  - Highlighter
  - Timestamp for viewing in chronological order



Example: Architecting one module

Module: Database

1. Disk mirroring
2. 2nd firewall to protect from accessing the database directly from outside the network

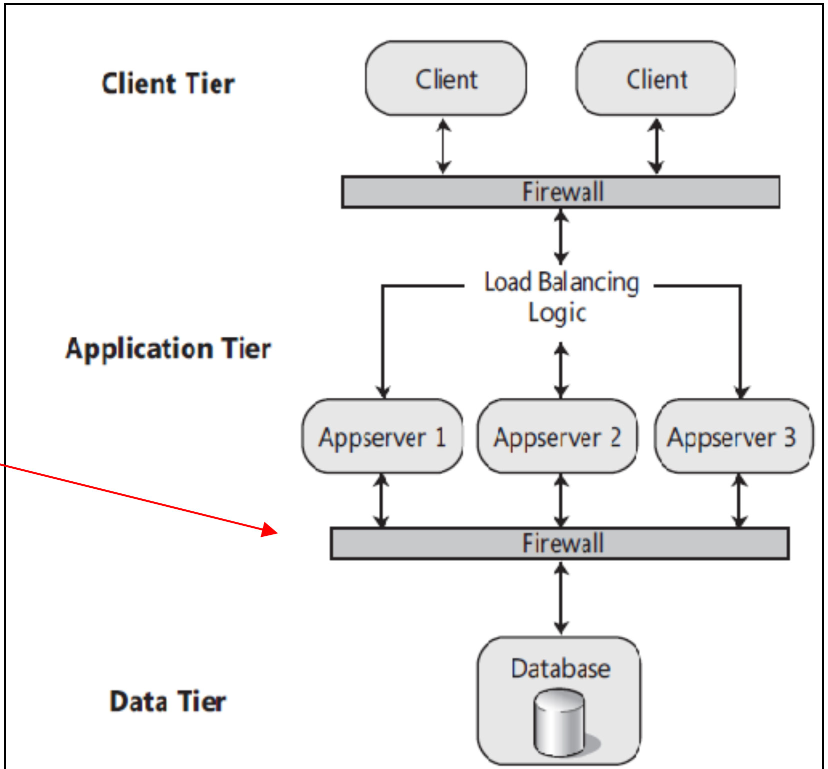
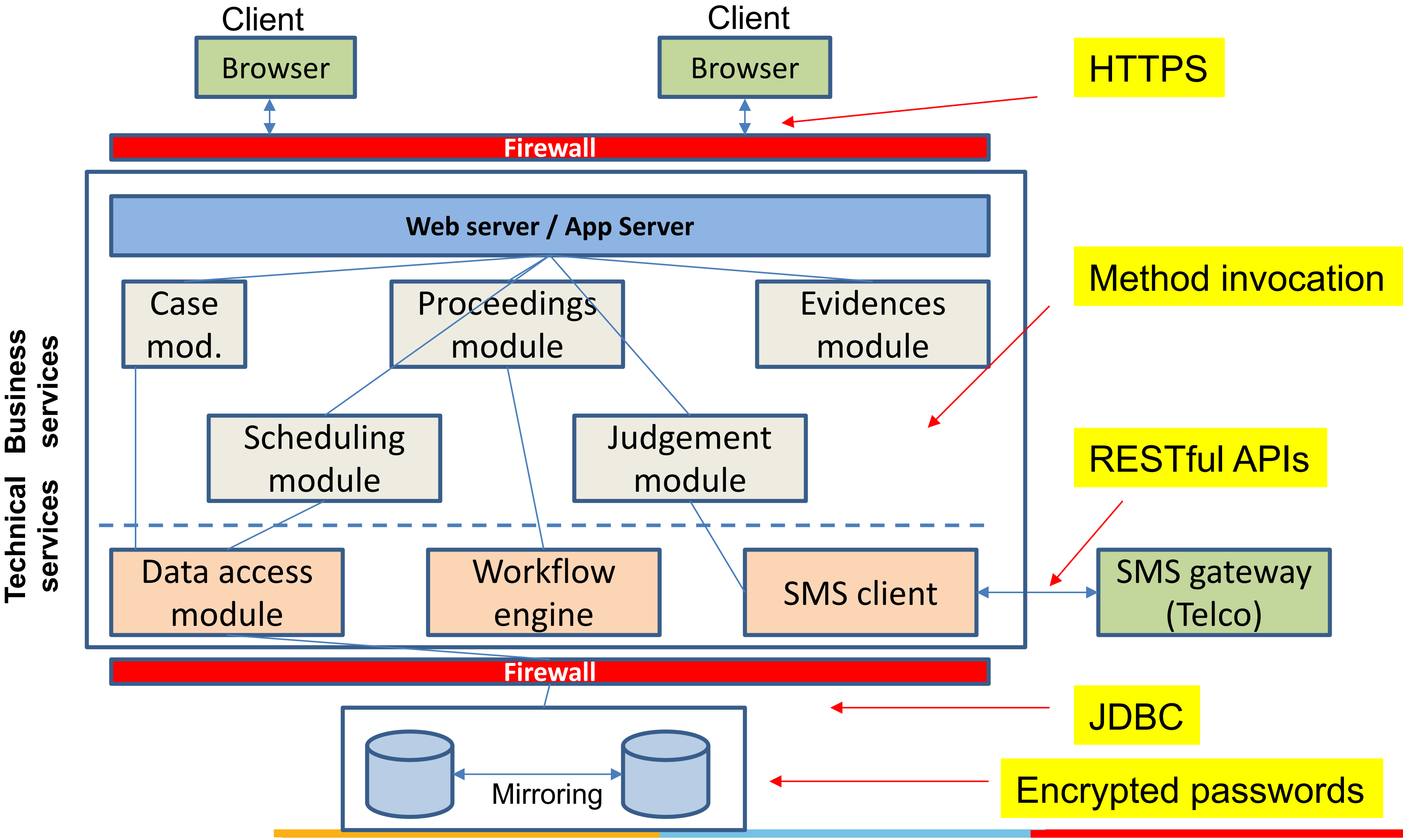


Figure 9
A load-balanced cluster

Example architecture



Maxio - Robotic System