Software Architecture

Conference Management System

Professor Manish Singh

Vinta Reethu Havya Sree K Dheekshitha B Akash Tadwai

May 6, 2022

Contents

1 Overview **2**
1.1 System Overview . . . 2
1.2 System Context . . . 2
1.3 Stakeholders . . . 2
1.4 Scope of the Document: . . . 3
1.5 Definitions, Acronyms and Abbreviations . . . 3

2 Architecture Design & Analysis **4**
2.1 Architecture 1: Shared Data Style . . . 4
2.2 Architecture 2: 4-Tier Architecture . . . 6
2.3 Comparing Architectures . . . 10

3 Final Architecture of CMS **10**

Chapter 1

Overview

1.1 System Overview

Conference Management System is a web-based application that handles a variety of aspects of conference management, including user registration, conference registration, paper submissions by authors, reviewer registration, papers assignment for reviewers, review submission, conference notifications, paper acceptance, display of paper reviews.

1.2 System Context

The System Context is clearly defined in the SRS document. Conference Record database specified in the SRA is the main sink of the information as it contains all the information about the conference paper such as all the reviewers allocated, Author details, deadlines etc. Academicians and researchers are important data sources. The papers are submitted to various conferences by the authors. Status of acceptance is examined before the deadline based on the reviews received and emails regarding the deadlines and status of acceptance are sent to the respective authors and reviewers.

1.3 Stakeholders

The main stakeholders are users registering for the conference, reviewers registering for the conference, conference chair creating a conference, authors submitting the papers. Concerns of various stakeholders:

- **Admin:** High traffic, server and database maintenance, load balancing on servers, delivering a secure and improved user experience by continuously updating the platform, bug removal, and meeting the SRS for developers' future additions.
- **Conference Chair:** Response time should be reasonable. Usability of the system. Algorithm for reviewer allocation should be flexible, so that we can replace it in the future if needed.
- **Reviewer & Author:** Response time should be reasonable. Usability of the system.

1.4 Scope of the Document:

In this document, we describe two possible architectures for CMS, compare them for various quality attributes, and then choose the most appropriate one, which is our final proposed architecture. By discussing the two alternatives, we also provide the rationale for selecting the final architecture. For architecture, we consider only the component and connector view.

1.5 Definitions, Acronyms and Abbreviations

Definitions:

- **Paper Allocation Algorithm**: Once all the submitted papers are obtained, papers are allocated to three reviewers based on the research interests of them.

Acronyms and Abbreviations

- **CC**: Conference Chair
- **CMS**: Conference Management System

Chapter 2

Architecture Design & Analysis

2.1 Architecture 1: Shared Data Style

The architecture includes a data repository that stores all information regarding user accounts, conferences, submitted papers, rejected papers, and reviewers, as well as various components that perform a typically disjointed set of activities that interact via the shared repository.

The following table describes all the components of the architecture.

#	Component	Component Type	Description
1	Repository	Database	This module is the database containing information about the users, conferences, papers, etc.
2	User Profile	Processing (Database modification & access)	This module uses the data repository to set/get user-related information such as login, sign-up, profile updates, etc.
3	Conference Registration	Processing (Database modification)	User Registers for a conference as an author.
4	Reviewer Registration	Processing (Database modification & access)	This module accesses the data repository to get information about reviewers by means of both invitations to reviewers sent by conference chairs and users registering as reviewers for a particular conference and finalises the reviewers.
5	Paper Submission	Processing (Database modification)	Adds the paper submission of the author in the data repository and handles related tasks such as withdrawal etc.

6	Reviews Submission	Processing (Database modification)	Reviewers submits reviews to the paper assigned to them and they are stored in data repository. This also handles the tasks such as editing reviews before a deadline.
7	Paper Decision	Processing (Database access)	This module accesses the data repository to get a paper submission and set the paper decision and notify the author.
8	Paper Allocation	Processing (Database modification & access)	This module accesses the data repository to get the submitted papers and allocates them to the reviewers.
9	Admin	Processing (Database modification & access)	This module helps website admin to maintain server, database and improve user experience.
10	Conference creation & updating	Processing (Database modification & access)	This module allows the CC to create conference and add/update the conference details.
11	Chair Registration	Processing (Database modification & access)	This module uses the data repository to set/get chair-related information such as login, sign-up, profile updates, etc.
12	Event Notifications	Processing (Database access)	This module sends the email notifications such as deadlines and other information to all the authors/reviewers.

Table 2.1: Architecture Components

Connectors in the architecture:

#	**Connector**	**Connector type**	**Description**
1	Read only	Database access	Represents reading of data from data repository
2	Write only	Database access,modification	Represents writing of data to the data repository
3	Read/Write	Database access,modification	Read as well as write by modules to the data repository.

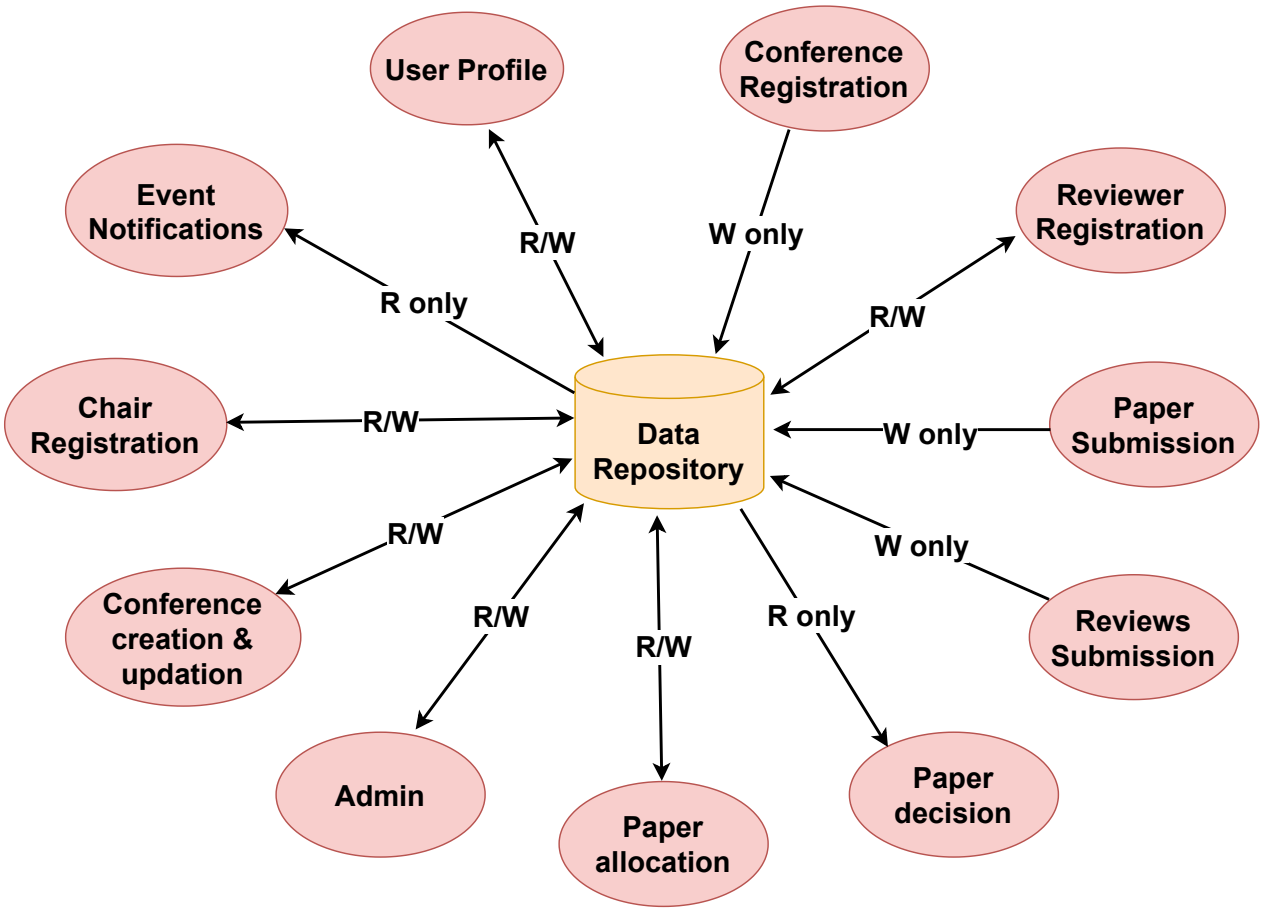

Figure 2.1: Architecture 1: Shared Data Style

2.2 Architecture 2: 4-Tier Architecture

The following table describes all the components of the architecture:

#	Component	Component Type	Description
1	User Repository	Database	This module is the database containing all the details of all the users of the conference management system (authors, reviewers and chairs)

2	Conference Repository	Database	This contains all the information about the conference such as name of the conference, papers submitted, authors, reviewers associated with it etc.
3	User Profile	Processing (Database modification & access)	This module uses the User repository and conference repository to set/get user-related information such as login, sign-up, profile updates, etc.
5	Conference Registration	Processing (Database modification)	User Registers for a conference as an author and all the info is stored in the user repository.
6	Reviewer Registration	Processing (Database modification & access)	This module obtains information on reviewers from the user repository and conference data from the conference repository, using both invitations to reviewers provided by conference chairs and users enrolling as reviewers for a specific conference, and then finalises the reviewers.
7	Paper Submission	Processing (Database modification)	Adds the paper submission of the author in the user repository and handles related tasks such as withdrawal etc.
8	Reviews Submission	Processing (Database modification)	Reviewers submits reviews to the paper assigned to them and they are stored in conference repository. This also handles the tasks such as editing reviews before a deadline.
9	Paper Decision	Processing (Database access)	This module accesses the conference repository to get reviews and decide whether to accept/reject the submission.

10	Paper Allocation	Processing (Database modification & access)	This module accesses the conference repository to get the submitted papers and allocates them to the reviewers.
11	Conference creation & updating	Processing (Database modification & access)	This module allows the CC to create conference and add/update the conference details. This information is stored in both conference and chair repositories.
12	Chair Registration	Processing (Database modification & access)	This module uses the user repository and conference repository to set/get chair-related information such as login, sign-up, profile updates, etc.
12	Event Notifications	Processing (Database access)	This module collects the the email notifications from the chair repository and sends them to authors/reviewers.

Table 2.2: Architecture Components

Connectors in the architecture:

#	**Connector**	**Connector type**	**Description**
1	Read only	Database access	Represents reading of data from data repository
2	Write only	Database access,modification	Represents writing of data to the data repository
3	Read/Write	Database access,modification	Read as well as write by modules to the data repository.

The differences between the architecture 1 and 2 are:

- One data repository is replaced by two repositories.
- 4-Tired architecture consisting of presentation layer, business logic layer, data access layer, data repository layer.

The two separate data repositories are:

- **User:** Stores the information of all users. Contains details like name, e-mail, conference he/she has submitted their papers to.

- **Conference:** Stores the information of all conferences. Contains details like conference name, deadlines, authors and reviewers registered for the conference, submitted papers for the conference, reviews etc.,

Data Access Layer separates Data Repository Layer and Business Logic Layer. The Data Repository Layer consists of 2 repositories as mentioned above. Presentation Layer deals with the display to end user by talking to the Business Logic Layer. One of the advantages of having Data Access Layer is that if there is any change in Data Repository Layer, changes are required only for Data Access Layer without affecting the rest of the system. Having two different databases helps in querying those databases faster as compared to the first architecture in which all information is stored in one database.

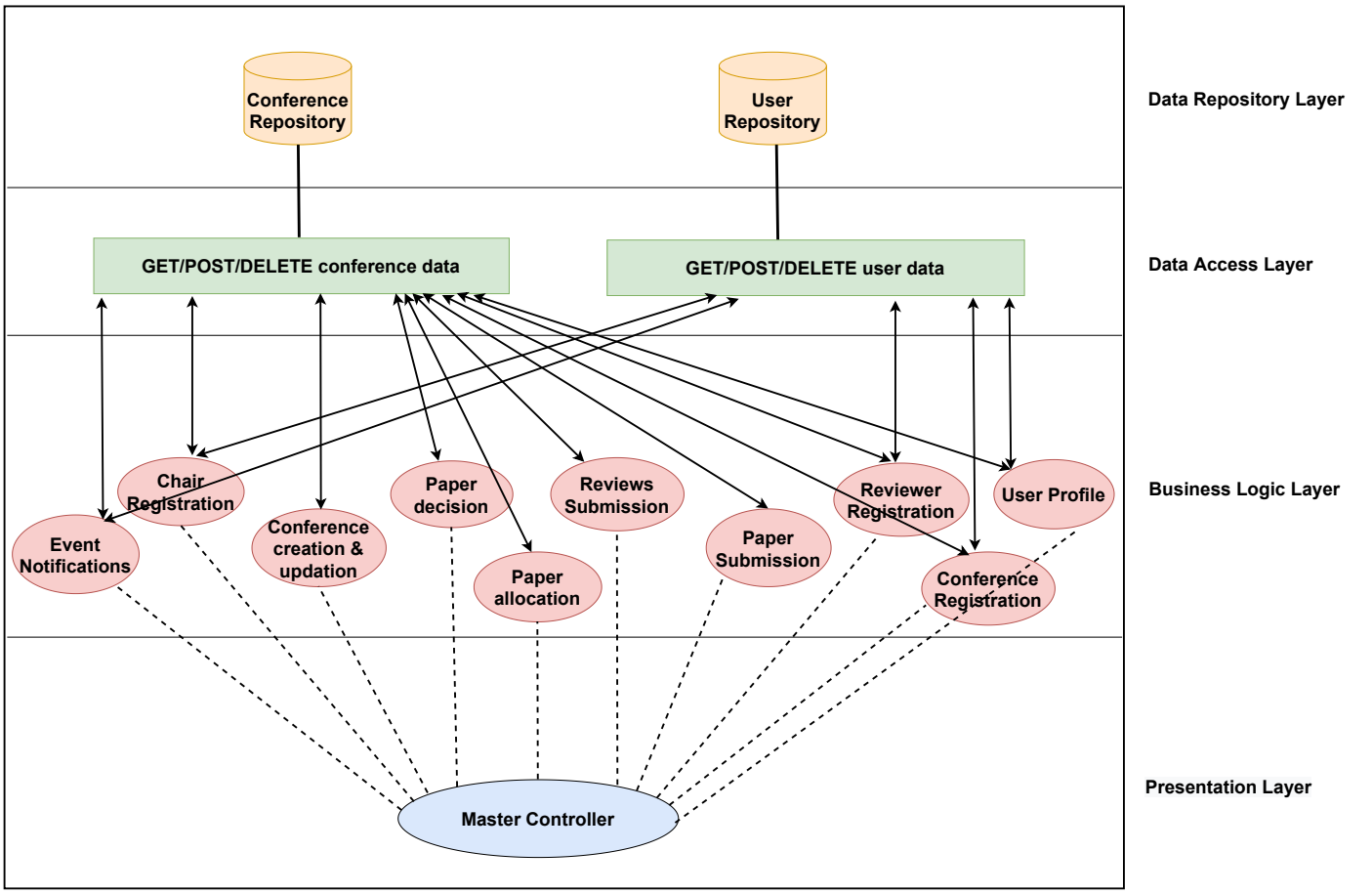

Figure 2.2: Architecture 2: 4-Tier Style

2.3 Comparing Architectures

To compare the above two architectures, we use ATAM analysis method.

Criteria	Architecture-1	Architecture-2
Security	Less secure since all the modules are accessing the same data repository	Higher security because of different layers making it less susceptible to attacks
Changes in Database layer	Not Easy, since code for all modules need to be changed	Easy, since only Data Access Layer requires the code changes.
Adding in functionalities	Easy	Easy
Querying Papers	Difficult and slower because of a unified data repository.	Easy and faster because of separate repositories.
Memory Efficient Scaling	No	Yes
Simplicity	Easy	Relatively easier than Arch 1
Debugging	Easy	Easier than Arch 1 (problems with different data repositories can be identified more easily).
R/W to Data Repository	Slow as all the tables are in the same repository	Faster as we can scale the database servers independently
Availability	High	Relatively higher
Response Time	Normal	Relatively less

Table 2.3: Comparison of Architectures

Chapter 3

Final Architecture of CMS

We can see from the above table of architectural comparisons that architecture 2 is **better** in many ways. It not only provides improved security, faster database switching, and easier changes in the data repository layer, but it also delivers significantly better performance when querying research papers and memory-efficient scaling. The second architecture is also predicted to have greater availability and response times. When compared to a single repository, segregation on the storage side (repositories) can also help reduce load. As a result, we propose that the ***second architecture*** be used for this project.