System Architecture Design Document for an Internet-based Collaborative Work Environment

Nick Forleo, Steven Lathrop, Manish Shrestha, Brian Stahl

Table of Contents:

1.0 Introduction

In today's rapidly evolving technological landscape, organizations are embracing the power of digitalization and connectivity to enhance efficiency, comfort, and sustainability within their buildings. A robust architectural design serves as the backbone of this transformation, offering seamless control, monitoring, and optimization of various communication services, all within a collaborative work environment accessible over the internet. The integration of an architectural design within a collaborative work environment brings numerous advantages.

By leveraging advanced collaboration technologies, this system enables intelligent control and real-time data analysis, empowering organizations to achieve energy efficiency, reduce operational costs, and create a comfortable, productive workspace for employees.

The concept of an internet-based collaborative work environment (IBCWE) has gained immense popularity in recent years. This approach allows organizations to transcend geographical boundaries, foster remote work capabilities, and encourage collaboration among employees regardless of their physical location. By extending this collaborative mindset to building automation, organizations can achieve a more flexible, dynamic, and user-centric approach to facility management.

The following system architectural design provides the requirements and guidelines for use by the Consultant in the design and construction of an internet-based collaborative work environment.

1.1 Description of the Architecture Documentation

This document serves as a comprehensive guide to implementing and utilizing a defined software architecture within an internet-based collaborative work environment. It covers essential topics such as components, installation, deployment, configuration, maintenance, and security considerations. By following the guidelines outlined in this document, organizations can leverage the full potential of a framework to optimize and enhance their collaboration among employees.

As you delve into the subsequent sections, we encourage all stakeholders to explore the possibilities that an internet-based collaborative work environment coupled with a robust architectural design can offer. Prepare to embark on a journey of improved efficiency, sustainability, and employee well-being within your organization's environment.

1.2 How Stakeholders Can Use the Documentation

Stakeholders play a vital role in leveraging the systems architectural design (SAD) documentation within an internet-based collaborative work environment. This comprehensive documentation serves as a valuable resource for various stakeholders, including facility managers, building owners, IT professionals, maintenance personnel, and even employees.

1.2.1 Facility Managers: The SAD documentation provides facility managers with essential insights into the system architecture, components, and functionalities. They can use this information to make informed decisions regarding system configuration and how to best manage in-house equipment, if necessary. Additionally, the documentation assists facility managers in understanding how to monitor and control the SAD remotely, enabling them to proactively address issues, perform maintenance tasks, and improve overall building performance.

1.2.2 Business Managers: Business managers have a vested interest in maximizing the value and performance of their employees. The SAD documentation offers business managers a comprehensive understanding of the benefits and capabilities of the system. They can use this knowledge to assess the return on investment (ROI) of implementing a SAD, evaluate cost-saving strategies, and make informed decisions about system upgrades or expansions. By utilizing the SAD documentation, business managers can align their goals with sustainable practices, enhance operational efficiency, and ensure long-term cost savings.

1.2.3 IT Professionals: In an internet-based collaborative work environment, IT professionals (developers, software engineers, UX engineers, etc...) play a crucial role in managing the technological infrastructure. The SAD documentation equips IT professionals with the necessary information to integrate the SAD with existing IT systems, such as network connectivity, data storage, and cybersecurity measures. By understanding the system's technical requirements, IT professionals can ensure seamless integration, troubleshoot connectivity issues, and implement robust cybersecurity protocols to protect sensitive data and maintain system integrity.

1.2.4 Maintenance Personnel: Maintenance personnel (tech support) are responsible for the day-to-day operation and upkeep of user issues. The SAD documentation provides them with detailed instructions on system installation, configuration, and maintenance procedures. They can refer to the documentation to troubleshoot common issues, perform routine maintenance tasks, and follow best practices to ensure optimal system performance. With this resource at their disposal, maintenance personnel can effectively diagnose problems, minimize downtime, and extend user happiness with the system.

1.2.5 Employees: Although not directly involved in the technical aspects of the SAD, employees (or general users) benefit from the system's functionalities within the collaborative work environment. The SAD documentation can provide employees with guidelines on how to utilize the system to customize their workspace environment, adjust lighting and temperature preferences, and optimize their comfort and productivity. By following the instructions outlined in the documentation, employees can create a personalized and conducive work environment, leading to increased job satisfaction and overall well-being.

2.0 System Overview

2.1 Business Goals

The business goals of an internet-based web collaboration company revolve around enhancing collaboration efficiency among geographically dispersed employees, minimizing employee productivity downtime or time spent on menial tasks, and reducing operating costs of business systems. These objectives are crucial for driving productivity, improving employee satisfaction, and maintaining a competitive edge in the digital landscape.

The goal is to break down communication barriers and enable seamless information sharing and real-time collaboration. By providing employees with user-friendly tools for document sharing, project management, and virtual meetings, the company empowers teams to collaborate effectively regardless of their physical locations. This boosts productivity, accelerates decision-making processes, and promotes a cohesive and connected workforce.

The next key goal is to reduce employee productivity downtime or time spent on menial tasks. Through the implementation of automation technologies and streamlined processes, the company aims to eliminate repetitive and time-consuming tasks that hinder productivity. By leveraging the internet-based web collaboration platform, employees can automate routine processes, access centralized information repositories, and utilize intelligent task management tools. This frees up their time and cognitive resources, enabling them to focus on higher-value activities and strategic initiatives, ultimately boosting their productivity and job satisfaction.

Finally, the company aims to reduce the operating costs of its business systems. By utilizing an internet-based web collaboration platform, the company can minimize the need for physical infrastructure, such as extensive office space or travel expenses for in-person meetings. The platform allows employees to work remotely, reducing overhead costs associated with maintaining large physical office spaces. Additionally, by integrating the web collaboration platform with other business systems and workflows, the company can optimize operational efficiency, reduce manual errors, and streamline processes, leading to cost savings in the long run. Our business goals are broken down into Table 1 seen below:

Table 1: Business Goals for the Building Automation System

Business Goals (Mission Objective)	**Goal Refinement (Engineering Objectives)**
Increase efficiency of collaboration between geographically dispersed employees	Ensure data privacy of user when connecting from unsecure networks during usage of any of the 5 services
	Ensure users have a nominal experience when using any of the 5 services from any location
	Optimize networking for when accessing the system
	Support the addition of a user to an existing session of any of the 5 services

	Support Desktop and mobile operating systems
	Implement an easy to use and intuitive interface for each service in the system
Reduce employee productivity downtime or time spent on menial tasks	Maintain high server uptime to keep productivity high
	Implement an improved process for integrating changes to any piece of the system
	Reduce the time spent troubleshooting disruptions in the system
	Increase efficiency when releasing changes for the teleconferencing services
	Increase efficiency when releasing changes for the system
Reduce the operating costs of our business systems	Decrease man hours spent making changes to the system
	Reduce the processing power and kilowatt usage of other services when a using service

2.2 System Context

Figure 1 below provides an operational view of the IBCWE system. The overall goal of the system is to enhance the productivity of the employees/users. Other stakeholders will use the system differently, but the main functionality revolves around the user experience. They will be able to access any of the 5 services (video, voice, chat, file storage, and/or whiteboarding) via a web browser.

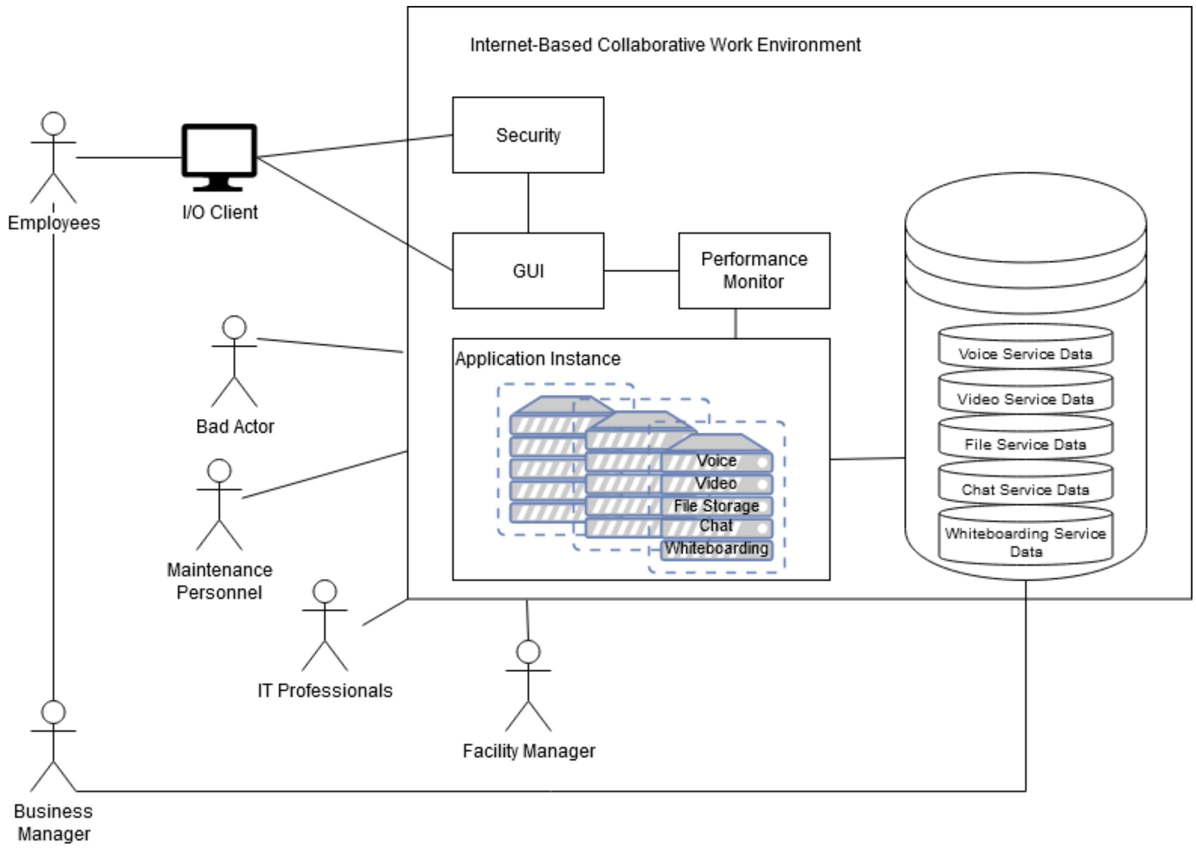

Figure 1: Operational view

Figure 2 below will demonstrate an operational scenario of the user accessing the system, the system receiving and processing the request, and then the user receiving a response with access to a service. This can be done concurrently by users with multiple input or output devices.

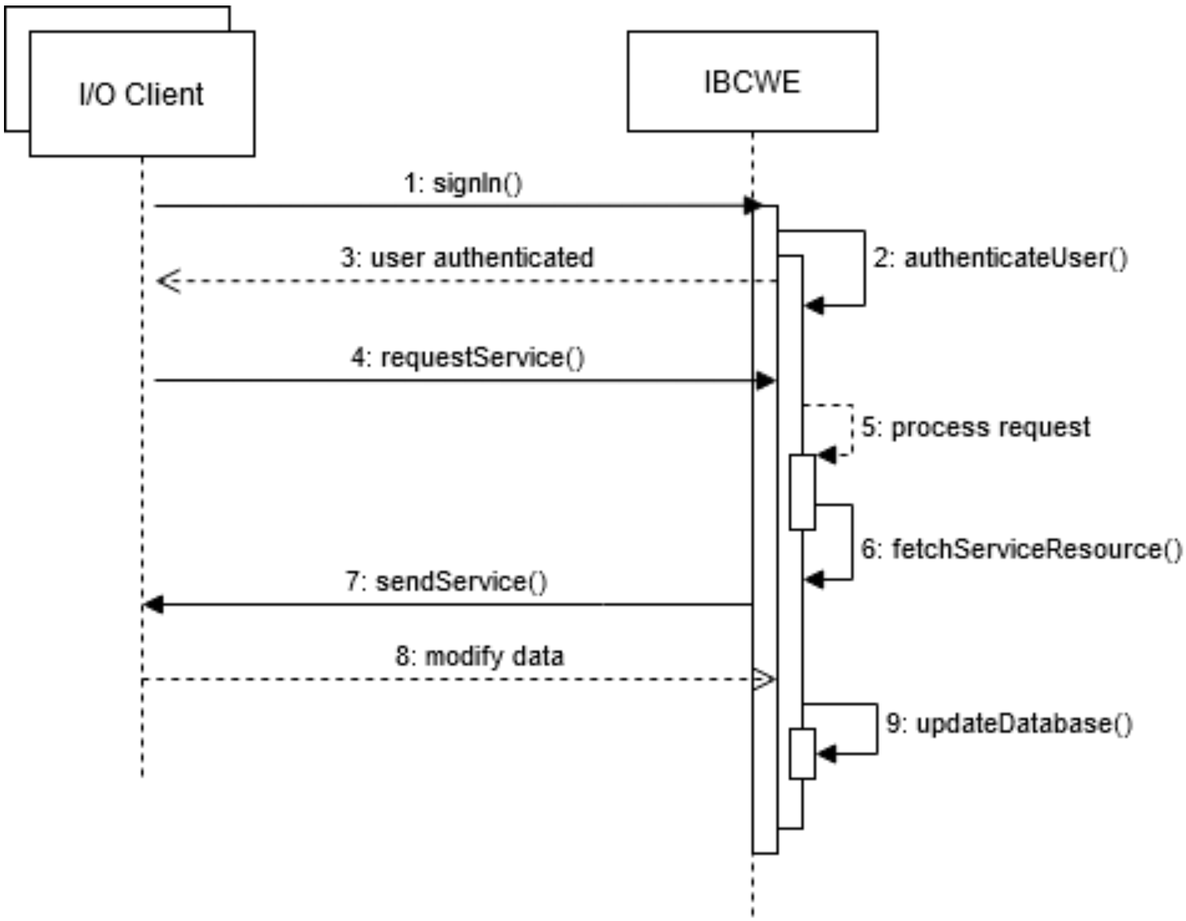

Figure 2: Operational scenario showing major flow of events (Key: UML)

The next diagram (Figure 3) demonstrates the overall context of the system.

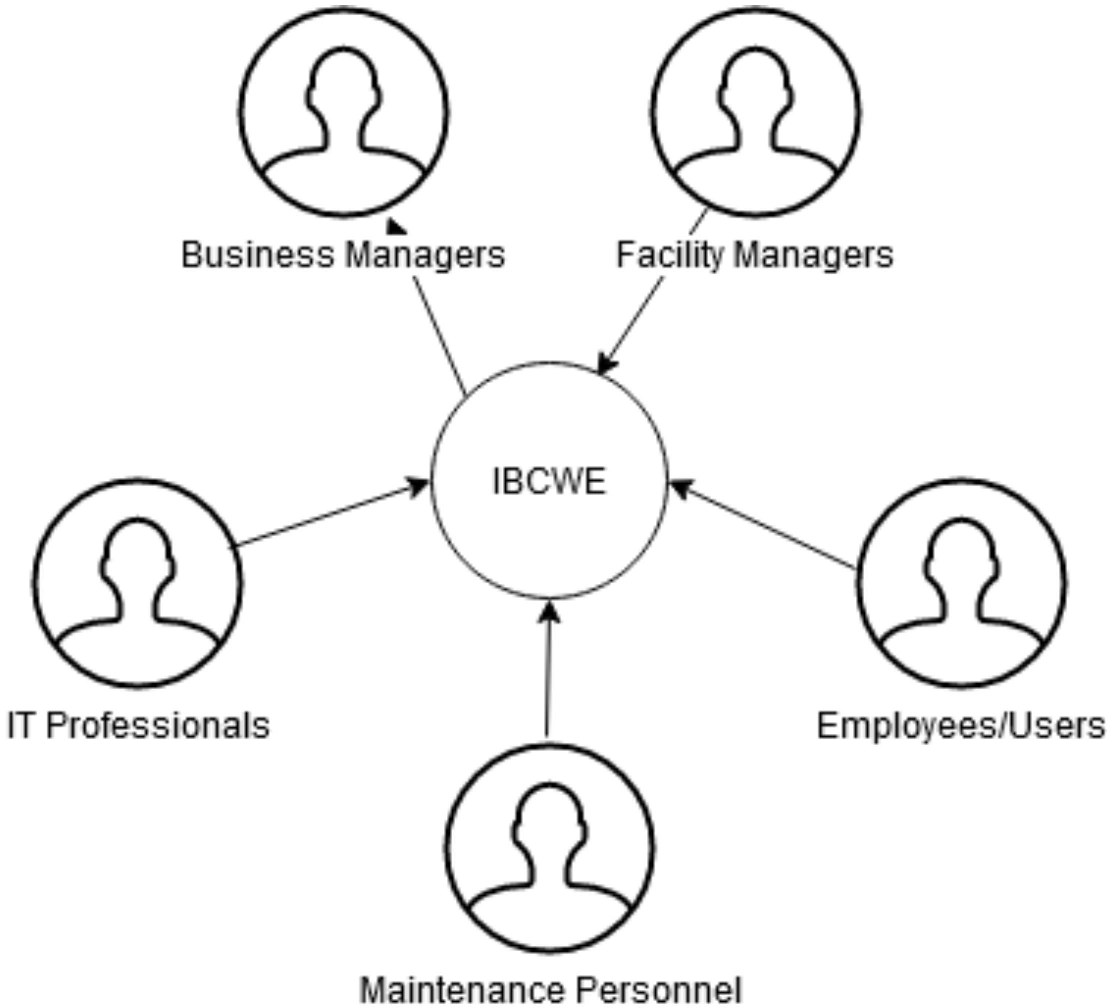

Figure 3: System context (Key: UML)

2.3 Functions

The IBCWE system shall provide a minimum set of functionalities to support the core objective. This includes, but is not limited to, managing permissions, supporting multiple devices, and accessing core services. The figure below (Figure 4) breaks the above diagram (Figure 3) down by a few use cases and how they are intended to use the system.

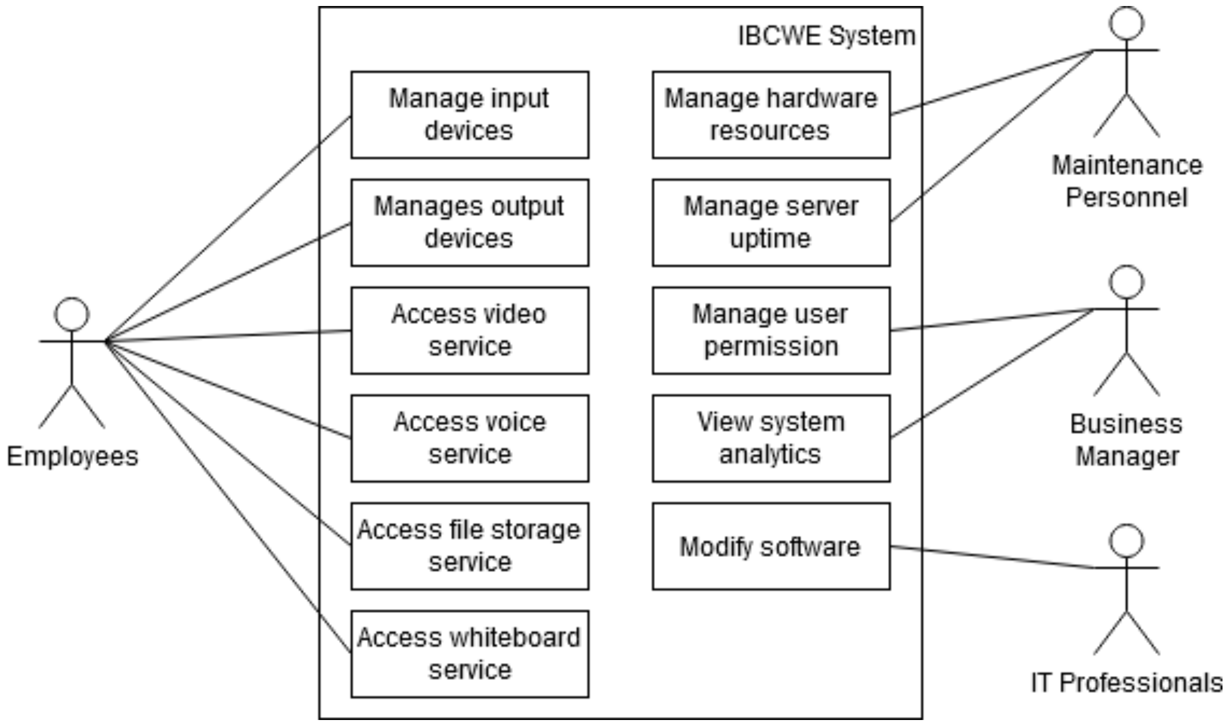

Figure 4: Use cases (Key: UML)

2.4 Quality Attribute Requirements

These requirements ensure seamless access and usage of services, robust authentication and role-based permissions, adaptive connection management, and reliable data transfer. Usability is a crucial attribute for the company, aiming to provide users with effortless access to any of the five services at any time. The architecture should prioritize user-friendly interfaces, intuitive navigation, and reliable connectivity. Users should be able to connect to and utilize the services without encountering connectivity issues or interruptions. This emphasis on usability enhances the user experience, encourages adoption, and promotes efficient collaboration.

Security is a paramount attribute for an internet-based web collaboration company. The architecture must verify user credentials during the sign-in process and ensure that only features aligned with the user's role permissions are accessible. Robust authentication mechanisms and role-based access control (RBAC) should be implemented to protect sensitive data and prevent unauthorized access. This ensures a secure collaboration environment where users can confidently share information and work collaboratively while maintaining data confidentiality and integrity.

Performance is a critical attribute that directly impacts the user experience. The architecture should be capable of detecting changes in connection strength or availability while a user is actively using a service. If a change occurs, the system should promptly reconfigure settings within 10 microseconds to optimize the connection. In the event that network disruption is unavoidable, the user should be promptly alerted. Furthermore, to optimize performance, the architecture should include data compression and decompression mechanisms. When a user's data in any of the five services is sent to the server, it should be compressed and subsequently decompressed upon receipt at the client side. This ensures efficient data transfer, without any packet drops, enhancing overall system performance.

By incorporating these architectural attribute requirements of usability, security, and performance, the internet-based web collaboration company can deliver a reliable, secure, and high-performing platform that enables users to seamlessly connect, collaborate, and share information. These quality attribute requirements are built into Tables 2 and 4. These tables show their connection to their respective engineering objectives and give additional scenarios for further description.

Table 2: Quality attributes and scenarios derived from engineering objectives

Engineering Objective	Quality Attribute	Quality Attribute Scenario	Priority
Ensure data privacy of user when connecting from unsecure networks during usage of any of the 5 services	Security	The user’s data is encrypted using industry standard encryption algorithms when transmitting over unsecure networks, ensuring confidentiality and preventing unauthorized access.	H
Ensure users have a nominal experience when using any of the 5 services from any location	Usability	The users can easily join sessions from any locations by simply clicking a link or entering a meeting ID. The user interface is intuitive, providing clearing audio and video controls, and supports a variety of devices and operating systems for seamless access.	H
Optimize networking for when accessing the system	Performance	The system automatically adjusts network settings to optimize audio and video quality, minimizing latency, upload speeds, and ensuring a smooth experience.	H
Support the addition of a user to an existing session of any of the 5 services	Scalability	The system can seamlessly accommodate new users joining ongoing meeting, maintaining Performance and user experience for any service.	M
Support Desktop and mobile operating systems	Cross-platform compatibility	The 5 services are compatible with a wide range of desktop and mobile operating systems, including Windows, macOS, iOS, and Android, allowing users to access and utilize the services seamlessly ensuring accessibility across different devices.	M

Implement an easy to use and intuitive interface for each service in the system	Cohesive Integrity	The 5 services offer a user-friendly design. The users can easily navigate and access features, such as audio/video settings, white board sharing, resulting in a smooth and efficient collaboration experience.	M
Maintain high server uptime to keep productivity high	Availability	The system aims to maintain a 99% uptime, ensuring that users can access the services and collaborate without significant interruptions or downtime.	H
Implement an improved process for integrating changes to any piece of the system	Integrability	A developer can implement a new feature into the system without affecting other components	L
Reduce the time spent troubleshooting disruptions in the system	Testability	A service member can troubleshoot and find a solution to an issue in the system within one days	L
Increase efficiency when releasing changes for the system	Deployability	A developer can push an update to the system using a CI/CD pipeline without manual interactions.	L
Reduce the processing power and kilowatt usage of other services when using a service	Energy Efficiency	The system dynamically allocates computational power based on the user's focused service, effectively minimizing power consumption by reducing resources allocated to other services that are not actively used. When a user focuses on one system service, other open services utilize half of the allocated computational power.	L
Decrease man hours spent making changes to the system	Modifiability	A developer can efficiently identify and implement changes within 30 minutes, enabling a streamlined and agile development process with reduced time spent on modifications.	M

2.5 Constraints

Constraints are critical for providing a seamless user experience, maintaining compatibility across different devices and platforms, and adhering to legal and regulatory obligations. The system must be designed to support a variety of hardware and operating system inputs and outputs. This constraint ensures that users can access and utilize the web collaboration platform regardless of the devices they are using, whether it is desktop computers, laptops, tablets, or mobile devices. The system architecture needs to be flexible and scalable, allowing for compatibility with different operating systems such as Windows, macOS, Linux, iOS, and Android. By accommodating various hardware and operating system configurations, the company can reach a wider user base, enhance accessibility, and enable seamless collaboration among users with diverse technology preferences.

Additionally, the architectural constraint of requiring registered users to be over the age of 13 is necessary to comply with legal and regulatory requirements, particularly those related to online privacy and data protection. This constraint aligns with regulations such as the Children's Online Privacy Protection Act (COPPA) in the United States, which imposes restrictions on the collection and processing of personal information from children under 13 years of age without parental consent. Implementing age verification mechanisms during the registration process ensures that the company maintains compliance and protects the privacy and safety of underage users. It also helps to establish a level of trust and accountability within the web collaboration platform, fostering a secure and appropriate online environment for users. Tables 3 visualizes the constraints below:

Table 3: Constraints		
Category	**Factor**	**Description**
Product	End User Devices	The system shall be able to support a variety of hardware and operating system inputs and outputs
Legal	User base	Any registered user must be over the age of 13

2.6 Architectural Concerns

Architectural concerns are essential for ensuring modularity, scalability, and efficient management of the system. By designing the system with independent components, each responsible for specific functionalities, the company can achieve flexibility and maintainability. This architectural approach enables individual components to be developed, tested, and updated separately, without impacting the overall system. It also facilitates scalability, as new components can be added, or existing ones modified without disrupting the entire system. Moreover, independent components foster collaboration and parallel development among different teams, enhancing overall productivity and enabling efficient troubleshooting or bug fixing processes.

The architectural concern of continuous and quick deployment emphasizes the need for an agile and efficient development lifecycle. In an internet-based web collaboration company, where user demands and market trends evolve rapidly, the ability to deploy new features, updates, and bug fixes quickly becomes crucial. Employing continuous integration and continuous deployment (CI/CD) practices, the company can automate the build, testing, and deployment processes, reducing manual intervention and ensuring a smooth and rapid deployment cycle. This approach enables faster time-to-market for new features and enhancements, allows for quick response to user feedback, and improves overall system reliability and performance.

Finally, the architectural concern of storing metadata in a database addresses the efficient management and retrieval of information within the web collaboration system. Metadata provides contextual information about various aspects of the system, such as user profiles, documents, project details, and activity logs. By storing metadata in a database, the company can ensure centralized and structured storage, making it easier to search, retrieve, and analyze data. This enables efficient collaboration and information sharing among users, facilitates personalized experiences, and supports data-driven decision-making processes. Additionally, a database-backed metadata system provides scalability, data integrity, and security measures, ensuring the confidentiality and availability of critical information. These architectural concerns can be seen visually in Table 4.

2.7 Architectural Drivers

Many factors influence the overall architecture and guide its design. In this case, we look at the design purpose, primary functional requirements, quality attribute requirements, constraints, and architectural concerns. Once a foundation is laid out (Sections 2.1 to 2.7), quality attribute requirements are ranked according to their importance. Table 4 combines all the previous sections into a fine-tuned list:

Table 4: Architectural Drivers for Internet-based Collaborative Work Environment		
ID	**Architectural Drivers**	
Design Purpose		
DSN-1	The system design shall support increased efficiency of collaboration between geographically dispersed employees	
DSN-2	The system design shall allow for the product to differentiate itself in the product in a competitive market	
DSN-3	The system shall reduce employee downtime and increase productivity	
Primary Functional Requirements		
UC-1	The system shall support 5 different services including voice communication, video conferencing, instant chat, file sharing, and collaborative whiteboarding.	
UC-2	Real-time features such as voice, video, and whiteboarding shall have a response time of no more than 100ms.	
UC-3	The system must be profitable	
Quality Attribute Requirements		
QA-1	Usability - A user attempts to access any of the 5 services. They	(H, H)

	are able to connect and use any of the services at any time.	
QA-2	Security - A user will sign into the product and the system will verify the credentials. Once authenticated, only features that are within the users' role permission will be available.	(H, H)
QA-3	Performance - The system detects a change in connection strength or availability while a user is using a service. It will try to reconfigure the settings for an optimal connection within 10 microseconds or it will alert the user of possible network disruption.	(M, M)
QA-4	Performance - When a users' data in any of the 5 services is sent to the server, it will be compressed and then decompressed once the client receives it. No packets will be dropped.	(L, M)
Constraints		
CON-1	The system shall be able to support a variety of hardware and operating system inputs and outputs	
CON-2	Any registered user must be over the age of 13	
Architectural Concerns		
CRN-1	The 5 system components shall be developed and run independently within the system	
CRN-2	The system shall be continuously and quickly deployable	

CRN-3	Metadata for the system shall be stored in a database

3.0 View Template

A view's primary function is to show the structure that it represents. Its documentation, therefore, consists of:

- *Primary Presentation*: shows the elements in a structure and the relationships among them.
- *Element Catalog:* details at least those elements and relations depicted in the primary presentation; these details include the interfaces of the elements and how these elements behave at runtime.
- *Architecture Background*: explains the design rationale, analysis results and assumptions.

All views contained in section 4 use this standard template.

4.0 Views

4.1 Module View

The module decomposition view involves breaking down the system into various modules that perform specific functions to support collaboration and communication among users. These modules containing many responsibilities are listed below in Figures 5 and 6 as well as Table 5:

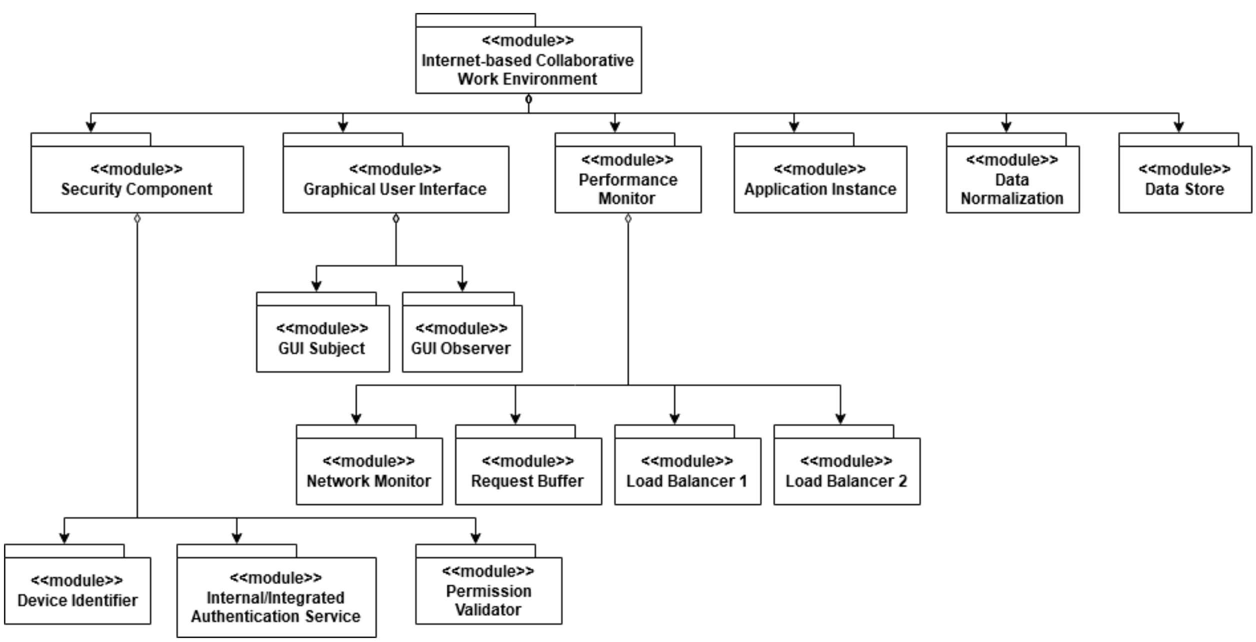

Figure 5: Module decomposition view (Key: UML)

Table 5: Major modules and their responsibilities		
#	**Module**	**Responsibilities**
1	Application Instance	1. Manages an instance of each of the five services 2. Reports resource usage back to load balancer 3. Serves service data to the GUI Subject via the Network Monitor 4. Reads data from the Data Normalization Layer 5. Writes data to the Data Normalization Layer
2	Authentication Service	1. Authenticates a user against our own database 2. Connects to third party authentication service to authenticate user 3. Requests permission for user from Permission Validator
3	Data Normalization	1. Receives write messages from an Application Instance 2. Receives read messages from an Application Instance 3. Manages concurrency for saving information to database
4	Data Store	1. Manages availability for all databases required for each of the services
5	Device Identifier	1. Records all metadata extracted from a client input device 2. Records all metadata extracted from a client output device
6	GUI Observer	1. Renders the data for a given service that is stored in the GUI Subject 2. Updates the data for a given service that is stored in the GUI Subject
7	GUI Subject	1. Maintains the current state of the requested service 2. Serves the state of a service to the GUI Observer
8	Load Balancer 1	1. Monitors resource consumption of the Application Instances 2. Assigns requests for resources to an Application Instance
q	Load Balancer 2	1. Same responsibilities as Load Balancer 1 2. Servers as a redundancy in case of failure
10	Network Monitor	1. Monitors all network traffic of the system 2. Reallocates networking resources to optimal usage
11	Permission Validator	1. Receives authenticated user information to administer permission and rights 2. Manages changes in user permissions
12	Request Buffer	1. Receives requests from the Network Monitor 2. Forwards

The diagram below highlights the major design decisions to improve the modifiability of the system. These decisions include:

1. The system will overall system will be accessed by a web browser. This increases the number of compatible devices that can use the system.
2. The GUI will utilize the Observer tactic to allow for a single state to be represented across multiple views
3. The Performance monitor includes a request buffer and load balancers to improve scalability. Application Instances can be scaled up to handle workload without having to make any changes to the code.
4. The addition of integrated third-party authentication services allows for the system to be more portable and fit within other business environments.

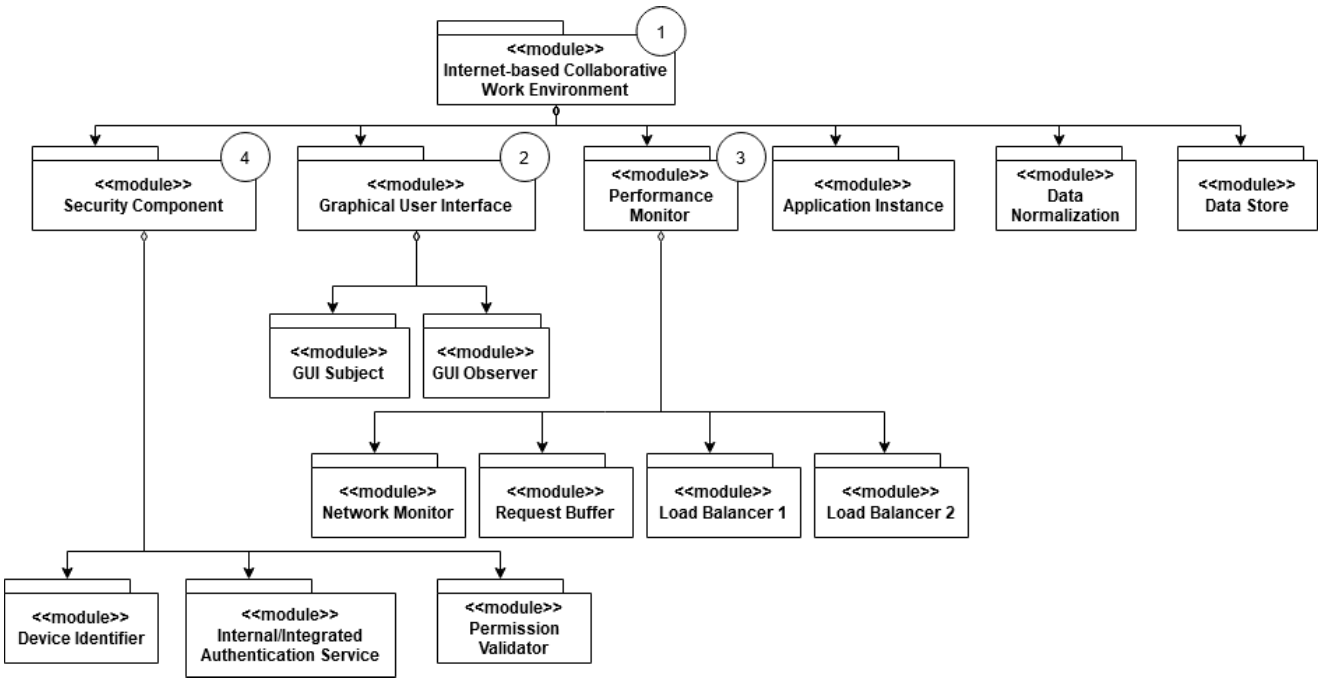

Figure 6: Module view with design decisions highlighted (Key: UML)

4.2 Component and Connector View

This section focuses on illustrating the system's architectural components and how they interact with each other through connectors to enable seamless collaboration and communication among users. One of the key components in this view is the User Interface (UI) component, which provides the graphical interface for users to interact with the web collaboration system. It includes elements such as menus, buttons, forms, and screens that allow users to perform various actions, such as creating and editing documents, sending messages, and scheduling meetings. The UI component connects with other components through connectors to receive and display data from the backend services and to send user inputs and commands to the appropriate modules for processing. This component plays a critical role in providing an intuitive and user-friendly interface that enables users to easily navigate the system and collaborate effectively.

Another important component is Communications, which handles the transfer of data and messages between users in real-time. This component ensures that communication is efficient, reliable, and secure, and it interacts with other components such as the User Management, Collaboration Services, and Data Store to facilitate the exchange of data and notifications. The Communication component also works closely with the UI component to update the user interface with real-time notifications and changes made by other collaborators.

The Component and Connector View showcases how different components interact and collaborate with each other to provide the desired functionality in an internet-based web collaboration company. By visually representing the system's components and their connections, this view helps in understanding the overall architecture and highlights the dependencies and interactions among the system elements. It enables efficient development, testing, and maintenance by providing a clear picture of how the components fit together and how data flows between them, ultimately resulting in a robust and scalable web collaboration system. Figure 7 breaks this down visually, while Table 6 lists their responsibilities:

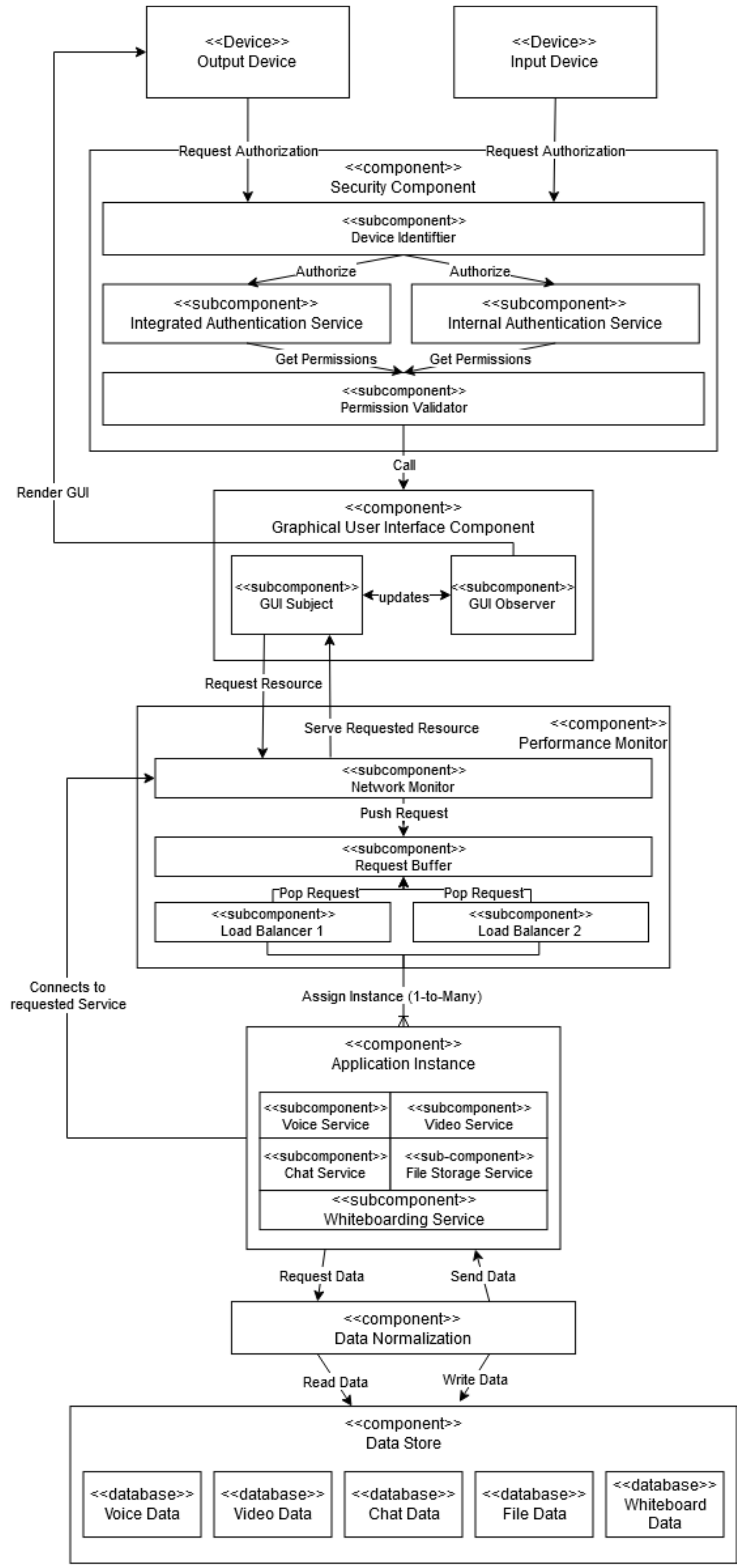

Figure 7: Component and Connector View

Table 6 contains each of the components and a high-level overview of the general functionality that they contribute to the IBCWE system.

Table 6: Major components and their responsibilities	
Component	**Responsibility**
Security Component	This component handles the authentication and general device monitoring of all the connected devices.
Graphical User Interface Component	Responsible for rendering the user interface and maintaining a current and accurate state of a given service.
Performance Monitor	Monitors network and usage data to automatically make optimizations as well as manage resources requests.
Application Instance	Managers multiple containers that hold each of the 5 services to use hardware as efficiently as possible.
Data Normalization	Normalizes all data of the system so it can be shared across services or integrated with new software if needed.
Data Store	Manages all the databases required to store all data pertaining to the services, including metadata and other analytical data.

The diagram below (Figure 8) demonstrates how the components interact with each other at runtime.

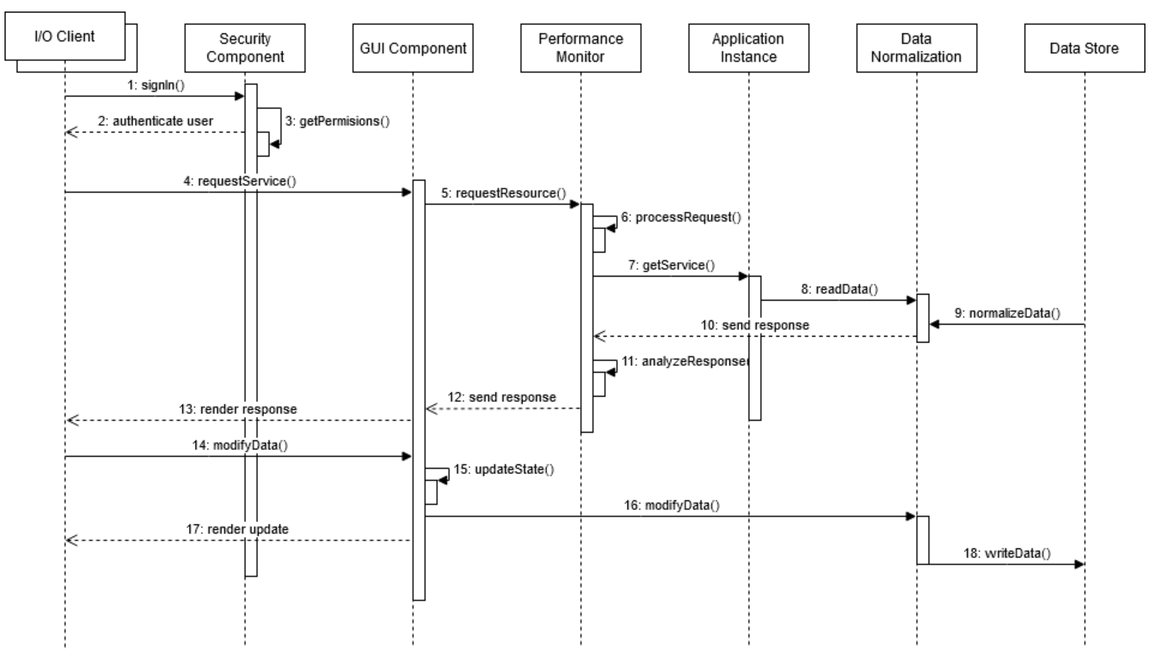

Figure 8: A process view showing component interactions

This diagram (Figure 9) highlights design decisions that were made to optimize the IBCWE system.

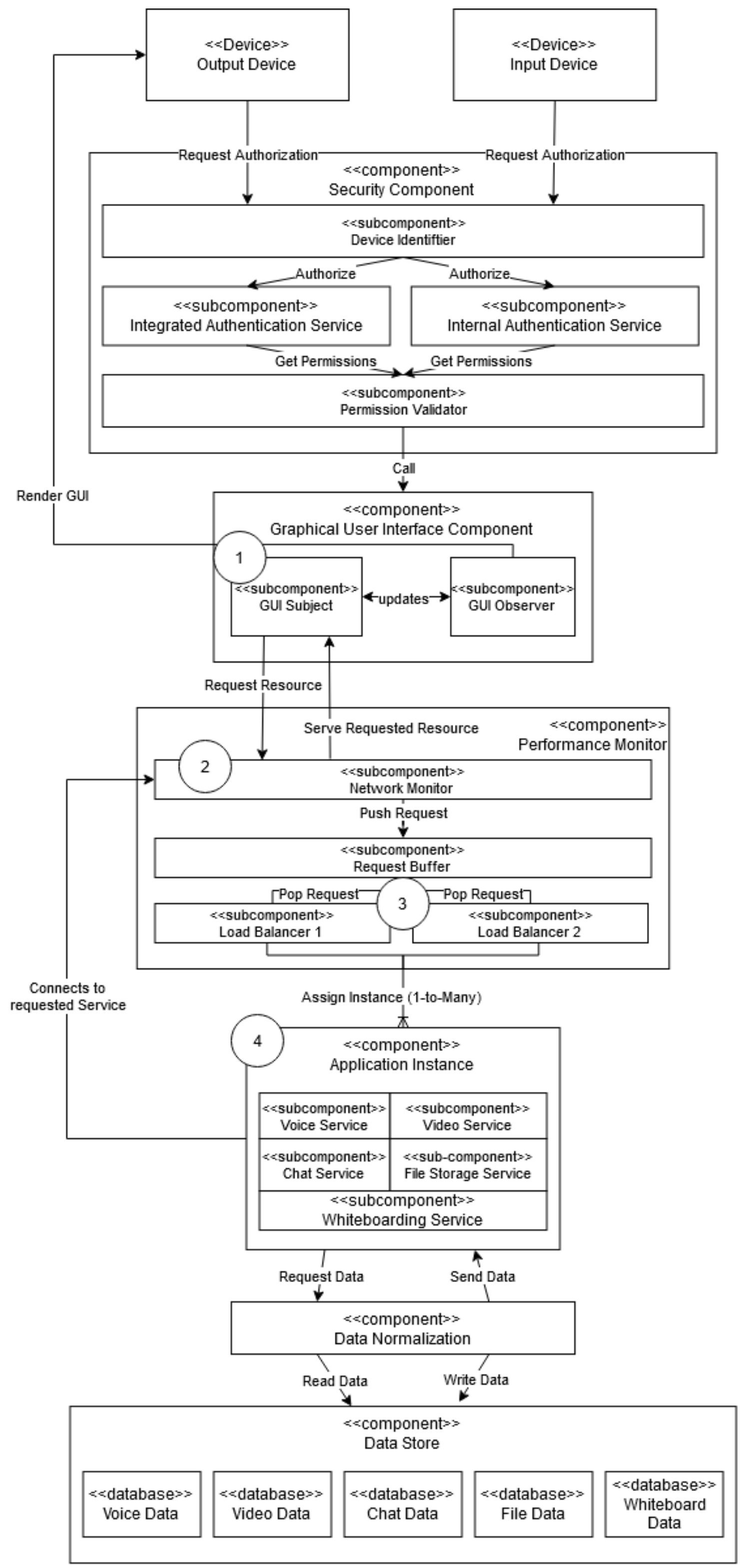

Figure 9: Component and connector view with design decisions highlighted (Key: UML)

1. The GUI is managed by a Subject/Observer model to manage the state and rendering of the front end more efficiently on the user device.
2. A Network Monitor was included to automatically optimize the client's network connection and reduce latency.
3. A load balancer was chosen to efficiently manage the resources allocated to a given user to access a service. A second load balancer was also included to add redundancy in case of failure.
4. Multiple instances of each of the five services are spun up to allocate resources to users more efficiently.

4.2.1 Performance Model

The table below attempts to represent the total computational cycles needed to utilize each service and the timing it would take for action to complete. We will assume that all the services have access to equal resources. CPU instructions take 0.00001 seconds to execute, writing to disk (physical I/O) takes 0.02s, and processing network requests takes 0.01s.

Table 7: Computational Resource Performance				
Service	**CPU Instructions**	**Physical I/O**	**Network Messages**	**Timing**
Voice	15	4	3	0.1115s
Video	31	6	8	0.2031s
Chat	9	3	1	0.1409s
File Storage	11	16	60	0.9211s
Whiteboarding	43	21	41	0.8343s

Based on this table, we can see that our system is performant for each of the 5 services. User experience may vary based on factors outside of our system.

4.3 Deployment View

The deployment view shown below describes the structure that the IBCWE system will be deployed in at runtime. This shows the relationship between components such that:

- Each component can be independently swapped into the system in case of failure.
- Each component can be allocated more resources to address bottlenecking within the system.

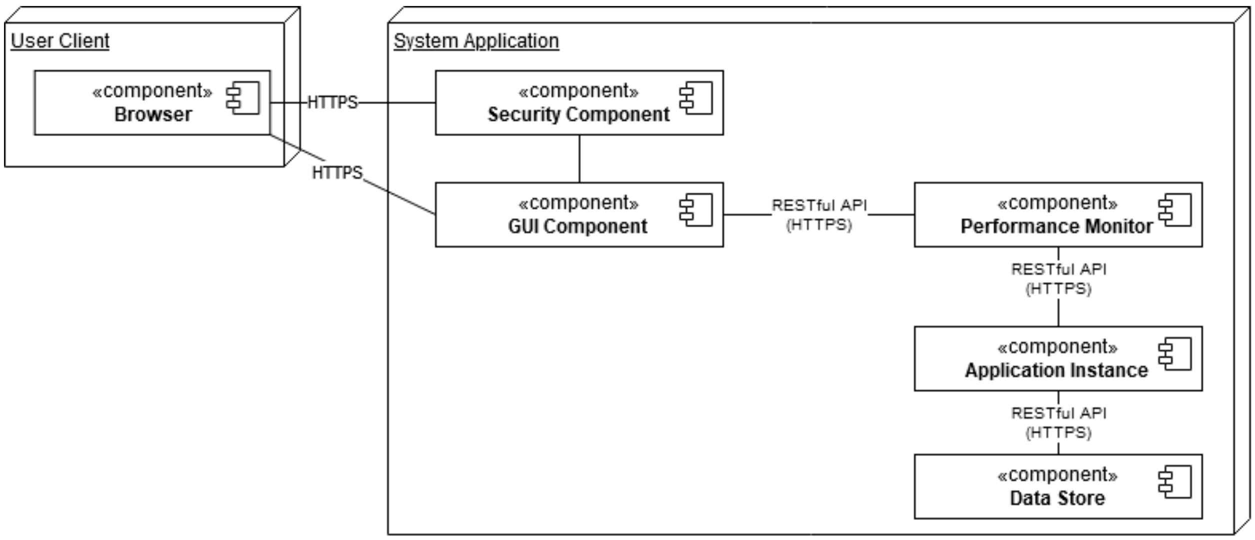

Figure 10: Deployment view (Key: UML)

The table below lists the major components that are to be deployed for the IBCWE system.

Table 8: Major components and their responsibilities	
Component	**Description**
Application Instance	This component contains and manages the executables for each of the 5 services that make up the core functionality of the system
Browser	A web browser application running JavaScript that will render the various user interfaces for each of the 5 services to the user.
Data Store	This component contains and manages the resources for all the databases that are required to store and manage data for each of the 5 services
GUI Component	This component is responsible for loading the necessary data to render the front end on the client device
Performance Monitor	This component handles network optimization as well as resource allocation for

	all the requests that are sent to and from the client device
Security Component	The component is responsible for user authentication, client device monitoring, permission validation and management.

Figure 11 below shows the decisions made to deploy the system. Each component was specifically chosen to be deployed independently to increase system availability. In this diagram, the data normalization layer is not explicitly shown, as it itself is not a major component of the system. However, it can be considered part of the Data Store deployed component since reading/writing data from the various databases would be ineffective without the data normalization layer.

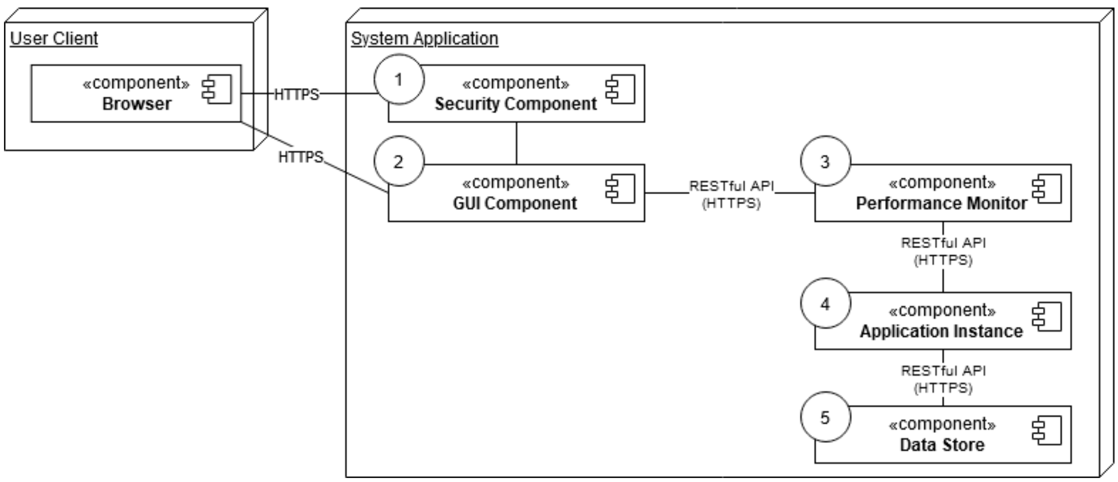

Figure 11: Deployment view with design decisions highlighted (Key: UML)

5.0 Mapping Between Views

Table 9 below shows how each of the modules map to the different components of the IBCWE system and their relationships.

Table 9: Mapping between module and component-and-connector (C&C) views

Module View	C&C View	Relation
Security Component	Security Component	Each of these modules are packages together as a part of the mapped component
Device Identifier		
Internal/Integrated Authentication Service		
Graphical User Interface	GUI Component	
GUI Subject		
GUI Observer		

Performance Monitor	Performance Monitor	
Network Monitor		
Request Buffer		
Load Balancer 1		
Load Balancer 2		
Application Instance	Application Instance	
Data Normalization	Data Normalization	
Data Store	Data Store	

Table 10 below shows the mappings between the C&C view and the deployment view.

Table 10: Mapping between component-and-connector (C&C) and deployment views

C&C View	**Deployment View**	**Relation**
Security Component	Security Component	The named C&C components are part of the corresponding deployed component
Device Identifier		
Integrated Authentication Service		
Internal Authentication Service		
Permission Validator		
Graphical User Interface	Graphical User Interface	
GUI Subject		
GUI Observer		
Performance Monitor	Performance Monitor	
Network Monitor		
Request Buffer		
Load Balancer 1		
Load Balancer 2		
Application Instance	Application Instance	
Voice Service		
Video Service		
Chat Service		
File Storge Service		
Whiteboarding Service		
Data Store	Data Store	
Data Normalization		

6.0 Rationale

6.1 Business Context

Our company looked to design a system that increased collaboration between geographically diverse users, reduce employee downtime, faster decision making, and reduce unnecessary business travel. We looked to design a system that would achieve these goals as efficiently and cost effectively as possible. This service could also be sold to other companies as a product to diversify and increase our revenue. A key concept behind the design was ensuring that the system would have as little downtime as possible to keep users working and customers happy. The architectural decisions were made to help to achieve this.

6.2 Key Features

- *High Availability:* Our design will make sure resources are used efficiently and when something does go wrong, it can be fixed quickly.
- *5-in-1:* Our system contains 5 major features in one system, which helps us stand out in the market.
- *Speed:* The architecture of the system means that connections and resource allocation are constantly being optimized behind the scenes for the user.
- *Interoperability:* Our system works easily with a company's single sign on (SSO) service for a seamless integration of our product in the customer's business environment,

7.0 Appendices

7.1 Appendix A: Acronyms

7.1.1	SAD	Systems Architectural Design
7.1.2	ROI	Return of Investment
7.1.3	COPPA	Children's Online Privacy Protection Act
7.1.4	CI/CD	Continuous integration & Continuous deployment
7.1.5	RBAC	Role-based access control
7.1.6	UI	User Interface
7.1.7	IBCWE	Internet-Based Collaborative Work Environment
7.1.7	SSO	Single Sign On