Technical University of Košice
Faculty of Electrical Engineering and informatics

Integration of open-source communication platforms into Telekom interoperability client

Master thesis

2024 **Bc. Artem Petrenko**

Technical University of Košice
Faculty of Electrical Engineering and informatics

Integration of open-source communication platforms into Telekom interoperability client

Master thesis

Study Programme:	Informatics
Field of Study:	9.2.1 Informatics
Department:	Department of Computers and Informatics (DCI)
Supervisor:	prof. Ing. Jaroslav Porubän, PhD.
Consultant:	Ing. Anton Vlasatý, Dr. Dmitry Sivchenko

Košice 2024 **Bc. Artem Petrenko**

Abstract in English

The main goal of this thesis is to integrate the open-source video conferencing technologies Jitsi and BigBlueButton into the corporate web application of the German company Deutsche Telekom. One of the most critical parts of this thesis is a detailed description of integrating these technologies into web applications based on the React framework. It also describes the relevance of such web applications and video conferencing technology and why it is an essential and integral part of a person today.

Keywords in English

Jitsi, BigBlueButton, video conference, integrating open-source video conference technologies

Abstract in Slovak

Hlavným cieľom tejto práce je integrovať open-source videokonferenčné technológie Jitsi a BigBlueButton do podnikovej webovej aplikácie nemeckej spoločnosti Deutsche Telekom. Jednou z najdôležitejších častí tejto práce je podrobný opis integrácie týchto technológií do webových aplikácií založených na frameworku React. Taktiež popisuje význam týchto webových aplikácií a videokonferenčných technológií a dôvody, prečo sú dnes nevyhnutnou a neoddeliteľnou súčasťou človeka.

Keywords in Slovak

Jitsi, BigBlueButton, videokonferencia, integrácia open-source videokonferenčných technológií

Bibliographic Citation

PETRENKO, Artem. *Integration of open-source communication platforms into Telekom interoperability client*. Košice: Technical University of Košice, Faculty of Electrical Engineering and informatics, 2024. 149s. Supervisor: prof. Ing. Jaroslav Porubän, PhD.

71827

TECHNICAL UNIVERSITY OF KOŠICE

FACULTY OF ELECTRICAL ENGINEERING AND INFORMATICS

Department of Computers and Informatics

DIPLOMA THESIS ASSIGNMENT

Field of study: **Computer Science**

Study programme: **Informatics**

Thesis title:

Integration of Open-source Communication Platforms into Telekom Interoperability Client

Integrácia komunikačných platforiem z otvoreným zdrojovým kódom do interoperabilného klienta Telekomu

Student: **Bc. Artem Petrenko**

Supervisor: **prof. Ing. Jaroslav Porubän, PhD.**

Supervising department: **Department of Computers and Informatics**

Consultant: **Ing. Anton Vlasatý**

Consultant`s affiliation:

Thesis preparation instructions:

1. Analyze the possibilities and technologies of open-source communication platforms and identify methods for integrating these platforms into the system.
2. Based on the analysis, propose extensions to the Telekom communication platform by integrating new open-source video conferencing technologies.
3. Implement the proposed extensions into the created client and integrate it with Telekom communication platform.
4. Verify and evaluate designed solution with appropriate method.
5. Prepare documentation according to the supervisor's instructions.

Language of the thesis: English

Thesis submission deadline: 19.04.2024

Assigned on: 31.10.2023

TECHNICKÁ UNIVERZITA V KOŠICIACH FAKULTA ELEKTROTECHNIKY A INFORMATIKY -11-

prof. Ing. Liberios Vokorokos, PhD.

Dean of the Faculty

Declaration

I hereby declare that this thesis is my own work and effort. Where other sources of information have been used, they have been acknowledged.

Košice, 13.5.2024 . . . . . .

Signature

Acknowledgement

I would like to thank my supervisor for:

- the time he gave me, because it is the most precious,
- the experiences he passed on to me, because they are priceless,
- all the comments, because it is only thanks to them that I have moved forward.

I am especially grateful to my parents who have supported me emotionally and financially. I always know that they believe in me, in all my ideas and want the best for me. Thank you to them for their support, motivation, for all they done for me.

Contents

List of Figures

List of Tables

Introduction

It is difficult to deny the role of exchanging information between inventors, scientists, and ordinary people who contributed to the progress of humanity. As intelligent beings, we are used to making technological discoveries that radically change our lifestyles. The wheel, the radio signal, the weasel, the Internet, the railroad, airplanes, cars, etc. - all of these have made our lives much more accessible.

Helpful information as knowledge can result from good, fruitful work that can change many people's lives. Initially, people exchanged knowledge with each other, then with the help of various radio signals, video communication is already quite familiar today. In the future, there may even be holograms.

Videoconferencing is a unique, convenient, and user-friendly communication tool. It becomes possible to exchange information through sound and eye contact between the interlocutors, which is essential for better perception and understanding of each other. Video conferencing can be used not only to show each other's images but also to visualize various information for better perception.

Today, video conferencing technology has become quite familiar to us as users of various gadgets and the Internet. We create and join conferences while studying, working, or just chatting with friends and family. Therefore, many different programs and services have already been developed that allow you to create video conferences with various functions and features.Great examples of such products are Skype or Teams from Microsoft, Webex from Cisco, Zoom from Saasbee, Inc. etc.

Over time, when creating new applications, including corporate ones, it became necessary to integrate such technologies for more comfortable use. This idea prompted the German company Deutsche Telekom for its corporate application.

The main idea of this application is to create a convenient video conference manager that can process links from various video conference technologies. As most of the popular technologies were already implemented, my task was to cre-

ate web-application with support for Jitsi and BigBlueButon which will be a part of Deutsche Telekom communication system. The main difference is that this system are open-source code, which indicates significant opportunities for integration, configuration, and customization of such platforms.

Formulation of the task

The main parts of this thesis can be divided into the following points :

- Write an introduction to long-distance communication.
- Creating an introduction to technologies for conducting video conferences, describing the importance of video conferences.
- Development of a web application as a platform for conducting video conferences.
- Integration of Jitsi and BigBlueButton technologies into the created web application.
- Evaluate ways of integrating Jitsi and BigBlueButton and describe the pros and cons of both systems when integrating and further use.
- Prepare documentation in accordance with the supervisor's instructions.

The individual parts of the work are written as follows:

- Introduction to long-distance communication is described in the chapter on pages 4-10.
- Characterization of video conferencing technology is described in the chapter on pages 10-50.
- Creating a web application as a basis for integration is described on the pages 50-80.
- Integration of Jitsi Meet is described in the chapter on pages 80-91.
- Integration of BigBlueButton is described in the chapter on pages 91-113.
- Deployment of the full web application is described in the chapter on pages 113-119.
- Analysis and comparison of the integration of Jitsi Meet and BigBlueButton systems is described in the chapter on pages 119-121.
- Web application user manual is described in the chapter on pages 121-134.

1 Analytical part

A wide range of communication media, such as mobile, satellite, and video communications, is available. Communication helps people learn new, necessary, and essential information, playing an integral role in modern society.

However, only some people think that all these various methods of exchanging information come down to multiple methods of transmitting signals, not only for connecting to a cellular operator or chatting but also at a shallow level. Various boards and microcircuits inside mobile phones, computers, or other electronic devices also send signals between their components. Communication and sending signals are much more than just communication - they are the exchange of information on which the entire system's stability may depend.

1.1 Evolution of long-distance communication

People have learned to send messages and information over long distances thanks to various signal methods, from ancient smoke signals and carrier pigeons to modern technologies. Today, we often use modern communication tools such as mobile phones and video conferences, which have significantly enhanced our ability to exchange information and connect with others across the globe.

Mobile phones, with their accessibility, have become an essential part of daily life. They enable voice communication, text messaging, and access to much information online. The convenience of mobile phones has not only transformed personal communication but has also revolutionized other industries such as business, healthcare, logistics, and education.

Video conferences represent another vital aspect of communication. They provide a visual and auditory connection beyond traditional phone calls, allowing for deeper exploration of new information and better, more nuanced, meaningful interactions. Video conferences also provide a platform for virtual meetings, collaborations, and presentations, reducing the distance between interlocutors, albeit virtually.

Thanks to their devices, mobile phones and video conferences have significantly increased human awareness of information. With these tools, individuals can share not only words but also facial expressions, hand-play, and the nuances of non-verbal communication. This modern communication experience improves understanding and more effective collaboration between conversation partners.

These technologies have also been reappreciated in global challenges, allowing people to stay connected with loved ones, control business remotely, and participate in virtual events. The ability to communicate over long distances has enhanced our personal lives and played a crucial role in forming today's globalized world. As technology advances, it promises to update further and expand our ability to perceive and share information across distances.

1.1.1 Evolution of writing

Communication technologies have evolved over a vast period, starting from the emergence of the first cave paintings, pictograms, and hieroglyphs around 40,000 years BCE [1].

Even in those ancient times, it enabled people to understand each other without being physically present. Various symbols in caves and on rocks could carry information about seasonal changes, weather, animal behavior, and even about other people or settlements living in a specific area. Similar symbols can be seen today in various history museums and ancient architecture. Even today, symbols are used to denote things, such as road signs indicating the passage for vehicles or pedestrians [2].

After the invention of writing, which we are familiar with today, the transmission of information became much more efficient, with established norms and rules for recording information. This allowed information written by hand to be transmitted in a form understandable to the majority, using symbols or alphabets.

Thanks to such records, drawings, and inscriptions, we continue to receive various information and knowledge about the existence of different cultures, animals, histories, and stories that were initially passed down orally and, when possible, immortalized.

Printing technology only magnifies the scale of this phenomenon. Woodblock printing, or printing on fabric and paper since the 7th century, was one of the earliest forms of printing that emerged in China around 220 AD.

The history of modern book printing began with the invention of Bi Sheng between 1041 and 1049. He devised the use of unique tiles with raised letters in mirror images, which were arranged to form text and then printed using a press

on paper [3].

The first printing presses, invented by the German Johannes Gutenberg in the mid-1440s, spread throughout Europe and worldwide. This significantly increased the number of printing houses and the dissemination of printed knowledge, books, and stories that were bought and passed from hand to hand [4].

Further development of printing presses:

- in 1867, the first automatic typesetting machine was created [5].
- in 1884, the Linotype typesetting machine was invented by American engineer Ottmar Mergenthaler, which already had a keyboard [6][7].
- in 1887, the Monotype was an automatic letter-casting typesetting machine created by American Tolbert Lanston that produced a set of characters using individual letters and spaces. Thanks to such inventions, it became much faster and easier to print information on an even larger scale [7].

Digital printing is used today - with special equipment capable of printing directly from electronic files. Thanks to digital printing technology, it is possible to quickly and efficiently print [8]:

- Sheet-fed digital printing - standard sheets, leaflets, and business cards. Various types of printers are used for this type of printing, with laser printers being the most accessible.
- Wide-format printing - used, for example, to create large-sized advertising banners or posters. Printing machines use inkjet printing principles, and materials can include regular paper and unique textile materials.

1.1.2 Evolution of telecommunication

Telegraphy

Telegraphy is a particular system or device that transmits information by special signals over distances.

An optical telegraph consists of a series of stations, usually towers, created to send written messages using special signals as a type of optical communication [9].

There are primarily two types this system:

- The semaphore telegraph, by movable indicators, to communicate based on their direction.
- The shutter telegraph is made of special panels that rotate to change light direction from the sky to send information.

Electrical telegraphs were developed as point-to-point text messaging systems and were utilized from the 1840s until the latter half of the 20th century. As the first electrical telecommunications system, they played a significant role in shaping the concept of telegraphy, revolutionizing the transmission of information without relying on physical transportation [10].

The first commercial solution was produced by Cooke and Wheatstone in 1837 [10]. The archetype of this system was Morse code, which was created by Samuel Morse in 1838, but in 1865, Morse became a standard for international communication [11].

Radio

Radio is a technology of transmitting signals and communicating by radio waves [12].

Radio waves are special electromagnetic waves that are generated by a transmitter. The antenna sends a signal with a frequency that can be from 3Hz to 300Ghz [12].

The history of radio began in 1873, from the moment when James Clerk, in his writings, namely the theory of electromagnetism (later called Maxwell's equation), described the phenomenon of Electromagnetic waves. The movement of magnetic and electric fields in space in the form of waves of different lengths was described [13].

The first radio communication system was created in 1895 by Guglielmo Marconi. Using a spark gap transmitter, they were able to send Morse code over a long distance [14].

Modulation is converting a signal from the transmitting side into an electrical signal that varies over time.

There are four main modulation methods used in radio systems [15]:

- AM(amplitude modulation) - AM transmitter increases or decreases the amplitude of the oscillation of the electromagnetic wave.
- FM(frequency modulation) - wave frequency of an electromagnetic signal.

- FSK(frequency modulation) - is often used to transmit a digital signal between devices.
- OFDM(orthogonal frequency-division multiplexing) - a special digital modulation method that, using high bandwidth, allows you to send a huge amount of information to mobile phones, digital audio broadcasting, television, and Wi-Fi.

Today, radio signals are widely used in navigation systems, remote controls, wireless networking, satellite communication, radio communication, and many other systems commonly used in modern life.

Television

Television today is one of the most widespread media that provides consumers not only with sound, but also with video and images of what is happening. Entertaining, educational content can be transmitted also in video format, which helps a person better assimilate the information received [16].

It was invented to teach and communicate at a distance, but it soon became a dynamic medium that made it possible to transfer knowledge worldwide.

The technical capabilities of the first devices were quite limited. They had only two colors, black and white (monochrome). Already in the middle of the 20th century, color televisions, as well as color television broadcasting, began to be widely used in the United States [16].

Thanks to technological progress, today's emphasis is placed on digital format transmission, picture quality, and widescreen image enlargement.

There are some modern ways of delivering television signal [16]:

- radio waves - signal transmission using radio waves using frequency modulation or orthogonal frequency-division multiplexing.
- broadcast satellite - sending a television signal using a satellite, which often requires a special antenna capable of receiving this signal.
- streamed through the Internet - thanks to various streaming and other services, you can watch the news, educational videos, and educational information directly from the Internet.
- optically recorded - Video material can be recorded on optical discs, which can be produced on various devices, such as DVDs, without connecting to the network or receiving any signal.

Mobile telephone

A mobile phone is a special portable device, and without it, it is pretty challenging to imagine our lives today. Thanks to telecommunications networks, it has become possible to transmit and receive audio, video, and even other types of data. Phones are connected to the public switched telephone network (PSTN) and are divided into [17]:

- Cellular telephone systems. It works on the cell principle, where large areas are divided into smaller cells, and base stations are used to send and receive signals. When the device is territorially overloaded from one cell to another, the station automatically changes without interrupting the signal exchange session [17].
- Global satellite-based telephony. The main advantage is the fast connection to the telephone system from any earth surface using a satellite signal. Phones of this type must have special devices that are capable of receiving and sending a signal to the satellite [17].

Today, mobile devices are necessary for life in modern society. A fast, accessible communication method allows communication with family or business management at a distance and exchanging necessary data or important knowledge.

Internet

The Internet is a system of interconnected networks that serve as a transfer of information for devices using protocols TCP/IP [18].

The Internet emerged in the 1970s due to research by the Advanced Projects Department of the US Department of Defense. The original idea was for US military networks to be connected to share a resource base.

Soon, many different companies and research centers began to finance their networks, which led to a merger in the 1990s [18].

Today, the Internet is an incredibly significant technology. Thanks to technological development, many devices began to access the Internet to obtain information, which gave enormous mobility. Thanks to the Internet, people can quickly and efficiently exchange various experiences, learn, and keep abreast of the latest events.

1.2 Video Conference

Nowadays, thanks to the incredibly rapid development of information technology, which includes almost all communication technologies, video conferences have become a relatively important part of our lives.

This made it possible to exchange information faster, more easily, and with better quality at different distances.
It has a stronger connection between interlocutors due to the transmission of video images.

There are two types of conducting, organizing, and processing video conferences [19]:

- **Point-to-point conferencing.** Two-way communication between people at a distance using only two devices. There are a huge number of scenarios for conducting this type of video conference:
  - private one-to-one conversations,
  - personal lesson,
  - interview,
  - various personal calls,
  - etc.
- **Multipoint conferencing.** It is a group conference where three or more devices can participate, depending on the platform's limitations.

  The primary role of this type of video conferencing is played by a particular multipoint control unit (MCU). This device is the central connection point for all devices in the conference. All digital data sent from the device of each conference participant comes to the multipoint control unit to exchange data with all other participants. Then, the same data is sent to all participants at the video conference.

  It is in this way that conference participants can communicate in different locations. This type of video conferencing is used in:
  - online team calls,
  - online exhibitions or conferences,
  - for any business conversations between people,

- group training,
- and many many other scenarios.

Video conferencing is an online technology that will allow people in different locations to keep in touch with each other without unnecessary movement. They can be used for various webinars, presentations, team calls, work conferences, interviews, communication with family, and much more.

Main video conference advantages [20]:

- Communication. During a video call, facial expressions and gestures significantly improve the efficiency of information exchange during the session, lowering the possibility of misinterpretation between participants.
- Comprehension. Video conferences help you understand and discuss information better using various built-in platform tools, such as screen sharing and whiteboard. This allows you to develop multiple projects as a team better and make decisions about any amendments much faster without physically being in the same area.
- Performance. Video calls allow participants to move quickly between calls with different participants, which is very important for business cause you don't need to waste time moving. Participants can make joint decisions rapidly and stay in touch with your team.
- Flexibility. Video conferences allow you to manage your time more flexibly and efficiently, allowing you to solve many tasks and responsibilities without leaving home or performing several assignments simultaneously.

Various video conferencing platforms often allow users to create special interactive rooms where two or more users can be present. Many platforms make it possible to specifically configure rooms depending on the tasks that need to be performed during the conference [21].

1.2.1 Evolution of video conference technology

The first video calls were implemented using special devices - videophones. To complement conventional two-way voice transmission, the researchers devised a device that could simultaneously transmit audio and video signals, allowing users to understand each other better. In addition to a conventional display, the

device has special circuits for converting video and audio signals that a regular telephone line can send [22].

The concept of videotelephony was first invented in the 1870s in the United States by the Bell Lab. It was then that the principle of human communication with the transmission of sound and video was described [22].

The videophone was first demonstrated on April 7, 1927. It was a one-way session between the United States Secretary of Commerce, Herbert Hoover, and the top management of the telecommunications company AT&T. The call spanned a distance of approximately 360 kilometers, connecting Washington, District of Columbia, to New York City [23].

Figure 1.1: One-way session video call demonstration on April 7, 1927

This event became the basis for the further development of video communication technologies, which made it possible to determine the vector for the further development of videophones.

The first two-way broadcast of video and audio using a videophone was demonstrated on April 9, 1930, linking AT&T Bell Laboratories, which developed the technology, and AT&T headquarters. For this call, ordinary television equipment of those times was used, which was closed between the participants in the video call [23].

By 1956, AT&T Bell Laboratories had developed the first videophone concept that used the regular telephone network for communication. It was a rather innovative and progressive technology that made it possible to communicate at a distance without creating any special infrastructure [23].

Despite this potential, this concept encountered several problems, mainly a relatively poor video image and a poor method of compressing the video signal. It was quite difficult for a video phone of that time to convey a clear picture based on the technological capabilities of that time [22].

Further research and attempts led to the first commercial products that were freely available to ordinary users. In 1964, Bell Labs, which developed video telephony, introduced the Picturephone Mod I at the New York World's Fair [24].

Figure 1.2: Picturephone Mod I presented in 1964 by Bell Labs

As a demonstration, eight special rooms were created into which people could enter and make video calls with each other, which was a real sensation at that moment. After this, the company decided to create such public rooms in various cities where participants in video calls would pay for the amount of time in a conversation [24].

Unfortunately, this relatively advanced technology did not achieve the expected number of users, and this idea failed. Firstly, due to the very expensive tariff, booking the device for a certain time was necessary to prepare the video

phone. Also, the user experience when using the device played a significant role [25].

The image was transmitted at 30 frames per second in black and white. The participants had to sit on the seat almost motionless so that the camera could transmit a clear picture. In the first six months, only 71 video communication sessions were carried out [24].

Based on the experience gained after developing Picturephone Mod I, AT&T decided not to stop and promote its technology further. The second model that made a splash in its time, Picturephone Mod II, was introduced in 1970. Global work was carried out to correct errors [23]:

Figure 1.3: Picturephone Mod II presented in 1970 by Bell Labs

- Screen was enlarged to dimensions of 5.25 x 5 inches, which made it possible to display a person well.
- 30 interlaced fps
- Thanks to the zoom lens and the new photo matrix, users were able to focus the image themselves.
- New graphic features.
- It became possible to see not only the interlocutor but also yourself during a call.

Having carried out a large-scale advertising campaign in 1970 and allowed the set to be sold for general use at home without creating special audiences, the device also did not succeed. The problem is that AT&T, in addition to the device itself, sold operator services, which were unaffordable for many users [24].

In 1973, AT&T CEO John De Butts decided to discontinue support for the Picturephone mod II [24].

In 1984, PicTel was founded on August 13; the name will soon be changed to PictureTel. They made a considerable contribution to the development of video communication transmission technology. The fact is that in 1986, a unique MCT algorithm was developed which made it possible to reduce the bandwidth required for retransmitting video images by almost three times [26].

In 1988, the company developed a new image encoding system called "hierarchical vector quantization," which further reduced video bandwidth [27]. Also in 1988, a special video codec C3000 and a special video communication system V- 2100 were introduced, which was a cabinet on wheels that allowed users to make calls from any room where the device was located [26].

A year later, an incredibly historical event occurred. In January 1989, AT&T and PictureTel conducted an international call with video and audio between PictureTel headquarters and AT&T's Paris office [26].

Given the fact that by the end of 1989, 70 percent of all PictureTel video conferencing systems were still unable to achieve the expected profit [26].

In 1991, PictureTel and IBM began working on the first video conferencing systems for Personal Computers. It was also this year that Cambridge students created the first web camera [27][26][25].

Connectix QuickCam is the first webcam and was introduced in 1994 as a commercial solution with a price tag of just a bit less than $100 [25].

Figure 1.4: Connectix QuickCam, first webcam introduced in 1994

In 1995, having made significant progress in developing software for personal computers, PictureTel held an online video conference with 50 participants to present its new systems [26].

In the following years, many companies such as Panasonic, British Telecom Presence, and many others began to release their solutions to enter the video telephone market, which soon began to be replaced by mobile telephones that could transmit a video signal.

Microsoft's first video conferencing software was introduced in 1996. It was the first software of its type to support all International Telecommunications Union standards [28].

In 2001, the world's first body surgery took place. The operation was performed by a French team of surgeons located in New York on a 68-year-old patient who was in Strasbourg, Germany [29].

Thanks to the use of high-tech equipment and telecommunication solutions from the Zeus company at that time, a special surgical robot completed the operation with the help of a team of surgeons.

Creating a minimum delay between the robot and the doctors was necessary for tasks of such a serious level. For this, France Telecom, and today Orange, provided a high-speed fiber optic connection [29].

In 2003, they presented the incredibly innovative Skype of that time, which allowed users to participate in conferences, exchange letters, and send various files completely free of charge [25]. The world's first HD video conference was conducted by Lifesize in 2004. The goal was to convey the most precise picture possible. It was this event that set the direction of technology for the following years [25].

In the following years, video telecommunications systems rapidly gained users and the attention of investors. Various companies began to release their solutions to various groups of their audience.

In 2006, Cisco TelePresence introduced its first telepresence system, which was created specifically for video conferencing with the participation of company employees. What was remarkable was the ability to deploy the system on already existing and configured IP networks. There was also a version of intercompany communication, which made it possible to expand the system in other companies actively [30].

Figure 1.5: Cisco TelePresence introduced in 2006

Several important events took place in 2010. To begin with, the Viber messenger launched the ability to create video calls directly from the chat; it was also possible to add up to 5 people to the conference [25].

Apple introduced its video communication platform called FaceTime. All iPhone or MacBook users could easily communicate with each other via video calling with minimal delay. After some time, the platform will add the ability to

use a regular cellular network for video communication [25]. Today, FaceTime is a prevalent application that improves and supports but also adds various new and popular functionality.

Google decided to buy the company Global IP Solutions to use their developments to make its own video calling system integrated into IP networks [25].

2011 is the year the Zoom video conferencing platform was created by former Cisco employees. Today, Zoom is one of the most famous platforms [31].

In 2013, Lifesize introduced its new Icon Series HD video conferencing system. This completely new system transmitted the picture with the best quality and was available to ordinary consumers. Also, 2013 is the year of the release of Slack, one of the most popular corporate systems today, where you can not only exchange messages but also create video conferences [25].

In subsequent years, companies that already had specific platforms added video calling functionality because everyone realized how convenient and productive it was:

- 2015:
  - The launch of online broadcasts on Periscope became a big and huge event. People from all over the world could select a person on an interactive map and watch the live broadcast directly from their phone.
  - The launch of Google Hangout, which at one time made it possible to hold large-scale events online.
  - WeChat is adding group calls directly from chats.
- 2016:
  - The launch of Google Duo, which was soon replaced by Google Meet as a more advanced version of the platform[32]. Today, Google Meet is actively used not only in Google services, but is also successfully integrated into the projects of other companies[33].
  - FaceBook has added the ability to make group calls through its Messenger chat.
- 2018:

– Launch of Cisco WebEx video conferencing platform, one of the most convenient and innovative at the moment. It is often used for corporate purposes, which greatly simplifies and optimizes the company's processes. WebEx can be used for both corporate use with chat, calendaring and call scheduling, as well as for daily calls, which can be useful for example for interviews.

– Group chats appeared on WhatsApp, Snapchat, and Instagram. This step allowed us to attract even more users, as well as further popularize video calls.

2020 year was a pretty good catalyst for the development of video conferencing technologies due to the pandemic. A large number of companies were forced to close their offices and transfer people to remote work [20].

Figure 1.6: Zoom video conference interface

The same thing happened with many educational institutions, exhibitions, events and much more. It was this period that showed how important video conferencing technologies are in today's world.

Many small and local services have been created to communicate with each other. Large companies began to pour even more money into supporting, improving, and integrating applications and products already existing on the market [21].

In just under 100 years, video calling technology has progressed greatly. From incomprehensible, complex, large, and expensive systems, humanity has come to quite convenient solutions on any user's computer or phone.

Video conferencing is very important in business, education, and communication today. Even if people are in different locations, they can continue communicating, saving time and financial opportunities.

1.2.2 Video conferencing during the COVID-19 Pandemic

The severe acute respiratory syndrome (SARS-CoV-2), commonly known as the coronavirus (COVID-19), began rapidly spreading in 2019 [34]. In 2020, the disease brought unprecedented events in many sectors, leading to measures to halt its spread by many countries [35]. Consequently, measures such as social distancing, transitioning to remote forms of interaction wherever possible, restrictions on border crossings, and much more were implemented. All of this led to crises in the economies, logistics, healthcare, and social spheres of many countries [36].

The pandemic disrupted not only the education process in schools and other educational institutions but also the established operations of many companies, organizations, and governments, leading to a significant popularization of various communication systems facilitating remote communication and decision-making processes [37].

Video conferencing platforms such as Zoom, Microsoft Teams, Cisco Webex, and many others became excellent solutions for maintaining communication between people during the pandemic, thereby significantly influencing the progress of technology development and the market for such communication systems. There was a need for more solutions to meet the high demand [37].

Despite all the difficulties and problems arising during the disease outbreak, video conferencing platforms played a key role in supporting established processes in education, business, organizations, companies, etc. The effect also reflected on the concept of technology itself, highlighting the importance of its development, support, and integration of innovative solutions [38].

The pandemic, relative to video conferencing platforms, led to [39]:

- **Accelerated technological progress.** Many companies have invested heavily in improving and optimizing their platforms, significantly enhancing the applications and the quality of video communication technology.
- **Accessibility.** Many companies have begun offering their solutions for free to attract an even more significant number of users.
- **Globalization.** During the pandemic, video conferences facilitated organizations and companies in simplifying collaboration.

- **Meetings.** The experience gained during the pandemic showed that meetings, webinars, and conferences can be conducted much faster and more efficiently online without the physical presence of each participant in the conference room.

- **Enhanced collaboration.** The broad functionality of video conferencing platforms has demonstrated to users that it is faster and more convenient to solve any problem together using co-browsing, screen sharing, or any other feature.

  The coronavirus pandemic has acted as a catalyst in developing communication systems designed for remote communication, including video conferencing.

1.2.3 How video conference works

One of the key features of video conferencing is the support of the most intimate contact between participants thanks to not only eye contact but also various interactive video conferencing functions [20].

Transmission of video images between participants in video conferences can be translated using special video transmission devices [20]:

- Built-in devices. These could be built-in video cameras on telephones or laptops, so you do not need to download special software. Often, built-in devices operate much more stable because the manufacturer configures and optimizes them.

- External devices. This type of device has pros and cons. It can be either ordinary webcams connected to the transmitted device via cable or the Internet or large sets with built-in cameras and speakers, which companies often purchase for meeting rooms.

  The main features of devices of this type are the user's ability to use cameras of various filming qualities and configurations and independently configure the location for filming the video camera.

Process of video conferencing

Video conferencing allows people to create a simulated face-to-face conversation in different locations. Users can conduct online meetings and conferences using all the platform's capabilities, including audio and video features.

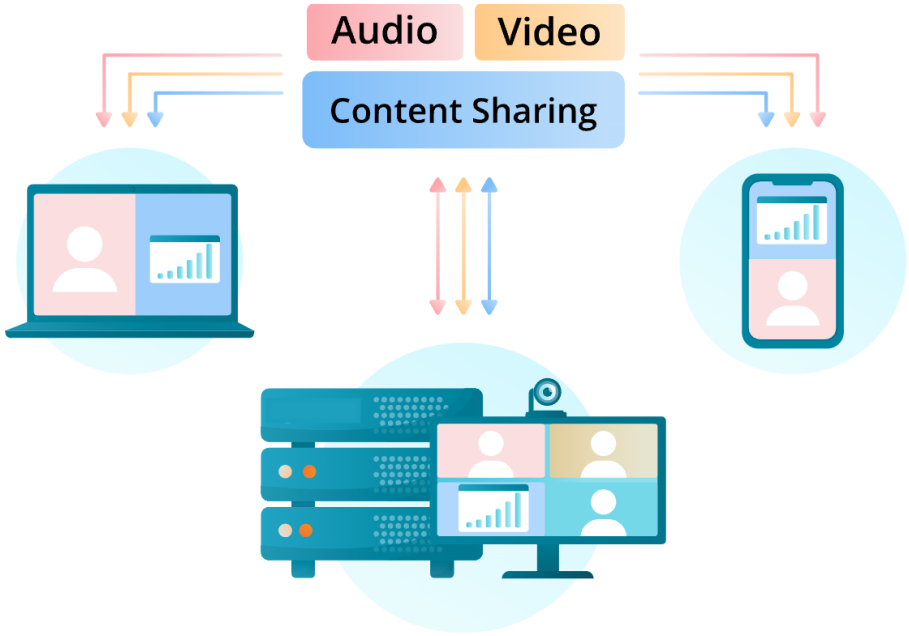

Figure 1.7: Video conferencing content sharing diagram

There are three main phases of video conferencing to ensure high-quality audio and video signals between participants on any platform, giving a good experience of communicating at a distance using video:

- Data compression.
- Data transfer.
- Data decompression.

Data compression

Data compression is the first step in holding and running the video conferencing and encoding stage. This phase is responsible for collecting analog data from cameras or microphones, which are waves with different amplitudes and frequencies [19].

This data contains not only the outline of a person's face or color but also the shades and depth of the image. Then, this data is converted into ordinary digital data ready to be sent in packets. It is through the data compression process and compression algorithms that the use of bandwidth can be optimized as much as possible [19].

Data transfer

At this stage, the data that has been received and compressed is transported to the service provider, which in turn redirects the packets to other participants in the video conference. For this stage, the principles of the OSI model play an important role due to which packets are moved within the network [19].

Data decompression

The codec plays a very significant role in the last decompression stage. Essentially, it is responsible for compressing and decompressing the received signals.

The codec also has various filters or scalers that can significantly clean the sound from multiple noises and improve the quality of the video image [40].

Based on the data received from the user, packets are decompressed using the codec. Digital data is converted into audio and video signals, which in turn are displayed to other conference participants [19].

1.2.4 Video conference platforms

In today's society, video conferencing is not only used for ordinary communication between people. This technology has long been part of various business, work, and educational processes, making it possible to improve and develop work organization and communication between people repeatedly.

Various platforms and software have made operating video conferencing tools much easier for ordinary users.

Chat

This feature allows you to communicate with people via text messages, create groups, or have private chats within a video conference. One usage scenario may be the desire not to interrupt the person speaking at the moment or the need to quickly send something without a long explanation so as not to lose video or audio contact.

Chats can be used to exchange messages and send files that may be important during the conversation or after.

Recording

The video conference recording function helps people save everything during the conference in video format, allowing them to re-watch important moments of the

session or send recordings to users who were invited but could not attend at the appointed time.

Also, various webinars or presentations are often published online for even wider dissemination of information.

Sound filters

Sound filtering is a common feature during video conferencing and is often enabled by default. Various filters and special codecs are used to make the sound clearer without noise. Sometimes, if the user is in a noisy place during a conference, this function allows sending an audio signal without sounds in the background.

Sound filtering significantly improves the quality of a video conference by removing unnecessary sounds, making the conference process more comfortable for other participants.

Video effects

Overlaying video effects is a fairly flexible function that allows participants to adjust the background to their included video images. It is often used in presentations or job interviews for branding or simply hiding the background.

Also, some platforms allow users to wear masks to have fun or achieve more friendly communication. This option is often available in regular chats supporting video calls, providing a more entertaining and rich application experience.

Conference management tools

Conference management tools are specialized functions or features within a video conferencing platform that streamline the scheduling and display of available calls. Not all platforms offer this functionality, making it flexible.

Many existing platforms facilitate this process by creating specialized rooms or teams with their chat, allowing users to initiate conferences directly within these groups. Additionally, some platforms offer scheduling functionality, which is highly convenient as it allows participants to see when calls are scheduled to start in advance.

Overall, management tools significantly enhance video conference creation's ease, convenience, and organization.

Device switching

Device switching is a necessary functionality during video conferencing. Platform users can change input and output devices, often for video or audio signals.

This option allows participants to independently set up their workspace with the devices that are best suited for them.

Many different microphones or web cameras can be included in a video conference with the press of one button. Some applications allow conference participants to check the external device before connecting.

Emotions

Emotions are fairly innovative and not unnecessary or critically important. In many platforms, this function is played as a pop-up or toast object that shows the selected emotion to other users for a few seconds. The hand-raising tool is also very often used to indicate to all other participants the desire to speak out, be active, or ask a question during the conference.

Polls

Some platforms allow you to create polls directly during conferences. This fairly flexible functionality enables you to diversify a video call slightly, showing other users that they are listening, watching, and taking part by indicating their opinion on one or several of their possible options.

It is also possible to create tests so that the conference presenter looks at how many participants understand the material and repeats it if necessary.

After the Internet became public and accessible to many people, a huge number of different companies began to rapidly develop their solutions and old developments to create their video conferencing platforms [25].

Communication between people using the Internet is much cheaper than cellular communications, increasing the number of video users.

Today, many platforms allow us to conduct video conferences using a huge amount of functionality that helps improve communication between participants at a distance. All these platforms are divided into two types [41]:

- **Proprietary video conference platforms.** The creator of the platform finances the supported software. The technology's source code is closed, so integration of platforms of this type is possible only with the consent and

license from the owner.
The main advantage is that the platform is run by a team of specialists who developed it from the beginning, and these people will assist with any problems.

- **Open-source video conference platforms.** The community is interested in its development and fully supports open-source software. It is easy to integrate on various platforms and highly and flexibly customizable because the entire application's code is publicly available.
When problems are found, the interested community will be able to help solve them.

Proprietary video conference platforms

In today's world, many companies provide various services, including software.

Any software requires proper funding, developer support, technical support for users, and many other aspects due to which ordinary people, businesses, and organizations use various platforms from companies [42].

A huge number of different companies, educational institutions, and government institutions annually purchase a huge number of licenses of various software for comfortable work, communication, and the entire team. Ordinary users also- ten give preference to products from large, well-known companies due to their prevalence, reputation, ease of use and download, and high-quality technical support [41].

There are many such examples with technologies and software for video conferencing. The most popular platforms for video conferencing today are [43]:

- GoToMeeting.
- StarLeaf.
- Google Meet.
- Cisco Webex.
- Microsoft Teams.
- Zoom.

GoToMeeting

GoToMeeting is a web-based video conferencing platform developed in 2004. As a web application for facilitating video conferences, It emerged as a significant innovation in virtual collaboration.

In 2010, GoToMeeting underwent a global update, implementing important features that significantly enhanced its functionality and usability [44][43].

In 2017, GoToMeeting underwent a major rebranding due to collaboration between Citrix GoTo and LogMeIn's development departments. This action led to the rapid growth of the platform [44].

Since its initial release, the application has continually evolved in response to user preferences and current trends, signaling promising prospects for its future [43].

Over the years, GoToMeeting has continuously evolved, adapting to technological advancements and user demands. Especially in recent years, the platform has undergone significant changes, particularly in the application user interface. GoToMeeting maintains a balance between quality and accessible functionality effectively [44].

StarLeaf

StarLeaf was founded in 2008 and, from the very beginning, has been developing various technologies related to video conferencing. In 2013, the company revised its policy and changed its vector to create cloud systems, thereby introducing StarLeaf Cloud. This move allowed users to join from a variety of software systems. Since 2015, the company began to actively produce its systems for conference rooms, which helped to optimize the work of organizations significantly [43][45].

The use of the StarLeaf system has increased by more than 1000% since the beginning of the transition of people to remote work and study [46].

The platform has good security and all the necessary functionality for video conferencing. Thanks to collaborations with other companies specializing in telepresence systems, StarLeaf is actively developing in the field of creating video conferencing systems [45].

Google Meet

Google Meet is indeed an incredibly convenient platform for hosting video conferences, particularly due to its integration with all Google services [47][43].

Google Meet was initially introduced in 2017. Upon its launch, the application supported video conferences with up to 30 participants. Its key feature was its web-based nature, enabling users to access the platform from any operating system [48]. In 2022, Google Duo and Google Hangouts were merged into Google Meet, significantly enhancing resource allocation and making the application optimal not only for regular video calls but also for comprehensive video conferences [49].

Google Meet allows users to open an extended Google Workspace account tailored to business needs. This account enables users to [47][43]:

- Increase the number of participants in conferences depending on the chosen plan.
- Utilize the feature for recording video conferences and saving them to the server.
- Make calls using a dial-in number.

Security is paramount for the service, and confidentiality is upheld reasonably well. Users benefit from comprehensive security and data protection by connecting to Google Cloud corporate services. Additionally, the platform supports [47]:

- In-transit encryption. All video calls are encrypted in transit based on IETF security standards.
- Anti-abuse features. Individuals without a password, a specific invitation link, or initial access cannot join video conferences.
- Regulatory compliance. Google Meet is part of a suite of Google platforms that allows for regular privacy checks on users, regardless of the platform they use.

Cisco Webex

Webex has been a significant player in the video conferencing software arena for quite some time now. It's widely used for online meetings, rallies, and seminars, offering excellent functionality for businesses of all sizes, especially with its integration into the Cisco system [50][43].

The application is available on multiple platforms, making it accessible across various operating systems. Webex is valued for its reliability, confidentiality, and security [43].

Founded in 1995, Webex initially built its applications on the MediaTone platform, allowing for expansion and the creation of the Webex MediaTone network [43].

In 2006, Webex introduced its first corporate solutions, such as online employee chats, providing quick communication. The same year, the company launched its initial web communications platform, Web Connect [51].

Cisco acquired Webex in 2007, facilitating the platform's integration into the broader Cisco ecosystem and expanding its popularity [50].

In 2018, Cisco unveiled Cisco Spark, a product based on the Cisco Webex platform, which combined hardware devices for video conferencing with access to the Webex platform [52].

In 2020, Webex introduced the Webex Classrooms platform, offering simplified functionality for virtual conferences and aiming to attract a broader user base [50].

Today, Webex continues to evolve, focusing on improving security features and integrating various artificial intelligence technologies to enhance communication and information transfer [50].

Microsoft Teams

Microsoft Teams is an important video conferencing and communication platform for businesses and private use.

Microsoft Teams can be used in the browser and as a standard application on Windows, macOS, iOS, Android, and Linux. This platform is highly secure due to its full integration into the Microsoft Office system [43].

Microsoft Teams allows you to customize your workflow conveniently and flexibly. In addition to regular chats between platform participants, there is such a thing as teams, where users can create separate channels with discussions and save files globally for the entire team or each channel. When creating a team, you can use specially prepared customization templates [43][53].

For video calls, Microsoft Teams allows you to make group and individual calls by transmitting audio or video signals and using a shared screen or group board. If a person cannot accept a call, you can simply send a voice message to the interlocutor [53].

To plan video conferences, the integrated Microsoft Outlook is often used,

which allows users to add this connection not only to Teams but also to all other Microsoft platforms for this account [53].

Microsoft Teams offers the ability to integrate various applications into a team. This functionality is often used to improve the entire platform's experience. Users can add applications that will help them open a file directly in Microsoft Teams, conduct a critical team survey, or check the weather [43][53].

The first developments of Microsoft Teams were made after Microsoft acquired Parlano, including their MindAlign voice chat system, which soon grew into Skype [54]. While Microsoft was developing Skype for business purposes, Slack appeared on the market, which allowed much more functionality for corporate use [55].

Microsoft decided to create its own Slack opponent platform and introduced it in 2017 under the name Microsoft Teams [55]. This platform has proven itself to be excellent and was an excellent replacement for the already old Microsoft Classroom product from the Microsoft Office system [56].

In 2018, Microsoft created a free version of Teams that had almost all the functionality of the paid version, not counting various restrictions on the number of files in a team or the number of participants in a video conference [57].

At the moment, the platform is actively developing and is regularly updated. Microsoft Teams is an incredibly flexible and customizable platform that allows you to conduct meetings and video calls and communicate with your team efficiently, securely, conveniently, and quickly. This software is available not only for large corporations but also for educational institutions, small companies, and ordinary users [53][43].

Zoom

Zoom Meetings, or Zoom, is a strong contender in the video conferencing platform market, offering an excellent software solution for users who prefer simplicity over complexity in their conferencing needs [43][58].

The software is lauded for its user-friendly interface, stable performance, and robust security features, facilitating seamless long-distance communication. Users can initiate Zoom conferences via web browsers or across various operating systems such as Windows, macOS, Linux, Android, and iOS [58]. While the free version imposes a 40-minute time limit on conversations and slightly reduces signal quality, it remains a popular choice for many users [43].

Zoom's journey began in September 2012 with the launch of a beta version, allowing up to 15 participants in a video conference. Stanford University became its first client, significantly contributing to its early development [59]. In 2013, version 1.0 was released, expanding the participant limit to 25, and by year-end, Zoom boasted approximately 1 million users [60].

Subsequently, the company rapidly scaled up, forging partnerships with Slack, Skype for Business, Qualcomm Ventures, and others. In 2017, Zoom introduced its telehealth product, enabling remote surgeries. It integrated Microsoft Outlook, Google Calendar, and iCal, streamlining meeting scheduling and calendar management [61]. Zoomtopia 2017 announced integrations with Slack and Workplace by Facebook [61].

In 2020, Zoom unveiled its first hardware solution, pre-installed with the Zoom system and running on the ServiceNow platform [62]. Starting in 2021, the company ventured into developing translation solutions utilizing artificial intelligence [63].

Zoom continues to evolve, introducing new functionalities and enhancing existing features [58]. Its user-friendly interface and comprehensive capabilities make it an ideal choice for beginners or those seeking convenient video conferencing solutions [43].

Open-source video conference platforms

Open source software is software where the code, important files, and architecture are in the public domain and often do not require a license to use or integrate. The development of this type of software occurs according to the principles of open exchange of information, prototyping, and transparency with the participation of the community interested in the technology's development [64][65].

Applications, products, and open-source programs allow programmers and various companies to improve by adding new functionality, modifying, configuring for integration, or simply improving the performance of the technology [65][64].

People prefer to use open source software due to [64]:

- **Full control.** Users can access all functionality's program code, architecture, and logic. This flexibility allows for easy configuration of the technology and its components.

- **Security.** Many developers prioritize the integration and use of open-source technologies. Such programs often have robust security measures because all developers can access the original code, enabling them to enhance the software.

- **Stability.** Open-source software projects typically have long-term perspectives. Since the code and architecture are public, users can be assured of continued support and development as other developers can contribute.

- **Community.** Open-source software is often supported by a community interested in its development. This community creates various forums, discusses ways to improve the software, and often helps each other with questions and issues.

Today, many open-source technologies and solutions exist and are actively being developed. These applications are successfully integrated into the services of large and small organizations and are often used as standalone products that aid people in their daily lives.

One example is video conferencing technology, which is incredibly important in our modern and active lifestyles. Video conferencing technologies are actively integrated into various organizational services or used for personal communication [65][66].

Currently, the best open-source technologies for use and integration into video conferencing platforms are those with a strong background and active community support. Some of these include [66]:

- Jitsi Meet,

- BigBlueButton.

Jitsi Meet

Jitsi is an open-source video conferencing platform that allows users to create video conferences and webinars and call people. This technology is a web platform and there are also application versions for Windows, Linux, macOS, iOS, and Android [66].

Figure 1.8: Jitsi Meet video conference interface

Jitsi is quite a popular and frequently used communication platform, encompassing many functionalities. The main ones include [66]:

- **Collaboration.** Collaborative work on any document in a live broadcast or video conference using Etherpad functionality.
- **End-to-End encryption.** Ensures secure communication, hiding access to cryptographic keys from third parties.
- **Chat.** Communication between conference participants can be conducted in parallel using chat.
- **Screen sharing.** The platform allows sharing the screen with other users. Sharing an application window without showing the entire screen is also possible.
- **Low bandwidth mode.** For example, a special mode in which bandwidth is reduced by degrading video transmission to achieve more stable audio signal transmission.
- **Password encryption.** The goal is to enhance an account's or user's security during a video conference.
- **LDAP directories.** The platform supports LDAP directories, allowing for quick data retrieval on the network without location information.

- **Integration.** The platform can be easily integrated into any application or system.
- **Workspace management.** During a video conference, participants can figure out their workspace on the screen: move participant images and enlarge or reduce various conference windows.
- **Ease of conference creation.** Creating a conference requires a few clicks, allowing for a quick and hassle-free setup. There is also the option to create passwords for conferences.

Jitsi began its development around 2003 as part of a student project by Emil Ivov at the University of Strasbourg [67].
In 2009, Emil Ivov founded his own company called BlueJimp, which provided software support services and established a support center for ordinary users [67].

In 2013, Jitsi underwent significant global changes. It added the ability to create calls with multiple interlocutors and integrated support for WebRTC. The system was hosted at the meet.jit.si domain, enabling users to create conferences directly from their browsers [68][69].

In 2015, Jitsi introduced version 2.6, which included numerous security enhancements. It removed support for the MSN protocol and improved audio signal transmission. The most notable change was the introduction of an internal Java Hyper SQL Database, significantly enhancing the application's performance and optimizing functions such as file transfer and chat [70].

Also, in 2015, Atlassian acquired BlueJimp and its products and developments. While Atlassian focused on developing Jitsi Meet and other browser-based services, the community continued to support Jitsi Desktop, designed for video conferencing through a desktop application client [71].

In 2018, 8x8, Inc. acquired the Jitsi technology from Atlassian. The new owners added numerous functionalities that facilitated easier integration of Jitsi into various services [72].

The technology itself, the Jitsi platform, consists of a whole set of different services, thanks to which all the functionality of the communication platform operates in tandem [73]:

- Jitsi Meet. A standalone video-conferencing application that can also be embedded into web applications.
- Jitsi Video Bridge - JVB. Jitsi server engine that powers all the video conferences.
- Jitsi Gateway to SIP - Jigasi. It's a gateway service connecting SIP to meetings.
- Jitsi Broadcasting Infrastructure - Jibri. It helps broadcast, record, and stream live videos on YouTube and other streaming platforms.
- Jidesha. A browser extension for screen sharing.
- Jitsi Conference Focus - jicofo. This server-side focus component manages media sessions and acts as a load balancer between conference participants and the voice bridge.

Jitsi, although considered quite flexible and convenient communication tool, has several pros and cons [73].

- Pros:
    1. Allows for flexible and rapid creation of your own Jitsi Meet Server or connecting to an existing one, which may not be the most secure solution for organizations.
    2. It's possible to manually adjust the quality of video, audio, and image reception.
    3. Authentication options include CAS (Central Authentication Service), JWT (JSON Web Token), and LDAP (Lightweight Directory Access Protocol), which enhance user data security during conferences.
    4. Fully customizable UI, statistics, and functionality are available.
    5. Integration with popular services such as Google Calendar and Google Tables is supported.
    6. Users can exchange messages and various files in the chat.
    7. Excellent documentation is provided not only for ordinary users but also for developers.
- Cons:

1. Jitsi conferences are not inherently super secure despite utilizing various encryption methods. Emphasis is placed on simplicity and speed of use.
2. Jitsi is unsuitable for creating conferences with many users as it heavily burdens data storage and increases signal quality loss for all users.
3. Sending huge files is not feasible due to the significant amount of local server memory required.

Today, Jitsi boasts a fairly active community dedicated to further developing the technology and integrating it into various services offered by other companies or products [66].

BigBlueButton

BigBlueButton is a communication technology used to conduct video conferences. The main feature is that the platform is open source, which opens up quite broad possibilities for its use and integration [66][74][75].

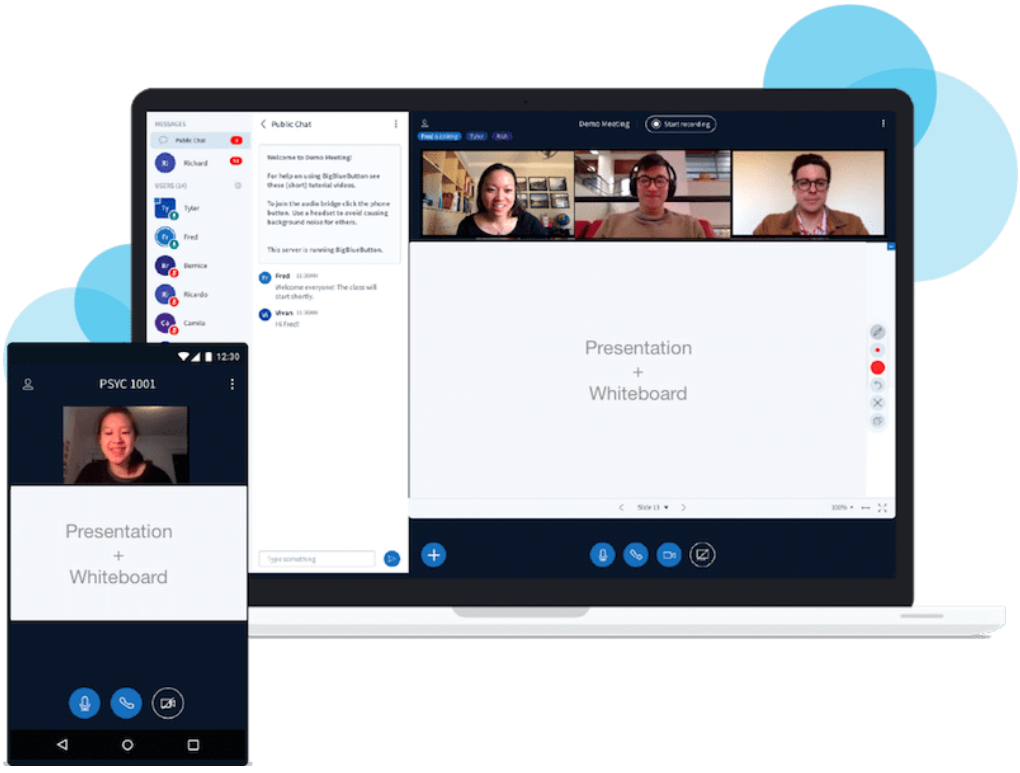

Figure 1.9: BigBlueButton video conference interface

The project's first developments began in 2007 at Carleton University with the help of student Richard Alam. At that time, the project was called Blindside, but after some time, it was decided to rename it to BigBlueButton, indicating the platform's ease of use [76][77].

In 2009, Blindside Networks was founded by Richard Alam, Denis Zgonjanin, and Fred Dixon to provide paid support services for organizations using the BigBlueButton platform. Also, in the same year, the source code of the video conferencing communication technology BigBlueButton was published on the Google Code platform [76].

From this moment, the platform became open source, a member of the Open Source Initiative, allowing the community to pay attention to the new technology, determine its development potential, and assist in further development [76].

In 2010, a special API (application programming interface) was created, allowing third-party applications or platforms to send requests directly to the BigBlueButton server. It was at this stage of the platform's development that plugins were made for integration with platforms such as Moodle, WordPress, Learning Activity Management System (LAMS), Tiki Wiki CMS Groupware, etc. [76].

In 2020, BigBlueButton version 2.2 was launched, which contained a completely redesigned client and server logic to support HTML5 [76].

In 2021, with version 2.3, the data storage technology, the database of BigBlueButton was changed. Support for NoSQL MongoDB databases was added, allowing for faster data processing [76].

BigBlueButton 2.6, released in 2023, added a lot of different functionalities or improvements [76][78].

- A new Whiteboard was added, which improved the user experience due to its simplicity and ease of use. It also introduced the ability to save each page in PDF format for further study after the video conference.
- Breakout Rooms Enhancement allowed for saving materials and information directly in the rooms. Thus, room participants would not waste time saving very important files; it would be enough to enter the corresponding room simply.

- Customization of the workspace was improved by allowing multiple participant windows to be pinned during the conference.
- Emotions were added, and in addition to raising a hand, additional emotions were introduced.
- The functionality for participating in and creating polls during video conferences was improved.
- Platform performance was enhanced.
- And much more.

The technology for creating video conferences, BigBlueButton, is quite popular and one of the highest-quality integration platforms. At the same time, some pros and cons can influence the choice of platform for organizations or ordinary users [78][77][74][79].

- Pros:
    1. **Integration with prepared plugins.** Simply integrate into specialized, prepared platforms, such as Learning Management Systems, with plugins prepared by developers. Such platforms can be Moodle, WordPress, Canvas, etc.
    2. **Security.** It incorporates many encryption methods and security features, such as end-to-end encryption, password-protected rooms, role-based access, etc.
    3. **User interface.** The platform is not just a conference window but also full-fledged management tools and breakout rooms that play a crucial role in user perception. This interface operates based on Greenlight, but it is also possible to fully customize all platform management tools. There is also the option to create user accounts.
    4. **Features.** It has a rich set of features that allow for convenient, comfortable, and quick video conferencing: chat, manual adjustment of connected additional devices such as audio and video, reactions, polls, recording video conferences, multifunctional whiteboard, screen sharing, etc.
    5. **Open-source platform.** Allows for very flexible customization of any communication platform settings.

6. **Easy to use.** The technology is very easy to use regarding functionality, room creation, and conferences.

- Cons:

1. **Complicated installation.** Installing and configuring the BigBlueButton server is quite complex and requires certain knowledge of information technology.
2. **Difficulty in integration and configuration into various platforms, systems.** There are no good and working libraries from developers for currently popular technologies such as Angular, React.js, and Vue.js. Integration is easy for already prepared platforms like Moodle, WordPress, Canvas, etc.
3. **Costs.** Expenses are required for server maintenance and hosting.
4. **Lack of free services.** There is no free service or server to connect and integrate into the platform. The BigBlueButton test server is only created to demonstrate functionality. It cannot be connected for integration and has limited functionality.
5. **Server limitations.** It is unsuitable for conducting large video conferences as it depends on the server's technological and practical capabilities.

As of the writing of the diploma thesis, the latest version of the platform is BigBlueButton 3.0.0-alpha.4, mainly aimed at improving functionality and optimization. Also, the Ubuntu version for running the server was changed from 20 to 22. Since this is only an alpha version, discussing any new major changes or functionality is difficult [78].

Today, BigBlueButton, a platform for creating video conferences, is successfully developing and being supported thanks to its community. They continue to communicate with their community through forums. The technology is actively used in organizations, educational institutions, and private video calls. BigBlueButton is often used as an integrated plugin in an already-prepared communication platform [77][76][74][66].

Comparison of video conference platforms

Today, there are many solutions for conducting video conferences. Large and small companies, organizations, and enthusiasts actively support and develop

various platforms. This is done so that their solution finds its audience and satisfies the largest number of users with its functionality and signal transmission quality.

Based on the goals of my diploma thesis, the Telekom platform already integrates technologies that I have already described: Zoom, Microsoft Teams, Google Meet, and Cisco WebEx. Since integration with open-source technologies BigBlueButton and Jitsi Meet is required, a clear analysis of the video conferencing technologies that are present and planned to be added is necessary.

Specifically, Zoom, Microsoft Teams, Google Meet, Cisco WebEx, BigBlueButton, and Jitsi Meet. This can help in further selecting technology or platforms for conducting video conferences for various organizations, firms, or private use.

Since all the selected video communication technologies are quite popular and commonly used, there is no particular point in comparing them based on such functions as [43][66]:

One of the main tasks of companies engaged in supporting and developing commercial products is not only to create the most optimal software to attract the largest audience. The financial aspect, which can offer various service packages and proposals, is important.

Platforms with open-source code are much more economical in this regard. Typically, payment is made for server support and the system on which the platform is running. Some companies charge for user support and maintenance in case of errors.

- Conference recording.
- Transmission of video and audio signals for communication between conference participants.
- Great quality of video.
- Chat for communicating with participants.
- Signal encryption.

Today, this functionality is already commonplace and is present in practically every platform related to video conferencing. Therefore, it is worth discussing more in-depth, specific features and characteristics to gain a more nuanced and comprehensive understanding of each communication system:

- Limitations.
- Whiteboard.
- Document upload.
- Co-browsing.
- Cloud-based.

Limitations

Platform limitations are a crucial aspect when comparing different platforms. This characteristic directly depends on the conditions of the communication system used to conduct video conferences. It is very important to take into account such features as:

- **Number of participants in conferences.** (**Capacity**) The maximum number of users in a conference for the free and most expensive package of platform services, if available.
- **Maximum duration of conferences.** (**Duration**) The maximum duration of a conference is for the free and most expensive package of services on the platform, if available.
- **Cloud storage per attendee.** (**Storage**) The maximum allocated storage for saving data for each user for the free and most expensive package of platform services, if available.

Limitations			
Platforms	Capacity	Duration	Storage
Zoom	100 - 1000	40m - 30h	Local - ∞
Google Meet	100 - 1000	24h	30GB - 5TB
Cisco Webex	100 - 1000	50m - 24h	1 GB - >50GB
Microsoft teams	100 - 300	60m - 30h	5 GB - 1TB
BigBlueButton	up to 250	system	system
Jitsi Meet	up to 250	system	system

Table 1.1: Limitations of platforms

Capacity The maximum number of conference participants in commercial platforms from different companies that work on these technologies directly depends on the choice of service package. However, we can observe that, in principle, all such technologies, namely Zoom, Google Meet, and Cisco Webex, have the same limitation for both the free version and the versions with the most extensive functionality, namely up to 100 users in the free version and up to 1000 active participants in the video conference in the largest subscription package.

At the same time, Microsoft Teams differs slightly in this regard, with the maximum number of conference participants in the largest service package being 300.

BigBlueButton and Jitsi Meet have the same maximum number of attendees, up to 250 active users in the conference. However, the capabilities of the server itself, on which all video conferences take place, play a significant role.

Duration The duration of video conferences varies significantly across different platforms. Zoom offers 40 minutes of the free version on all analyzed platforms and up to 30 hours of the largest package.

Meanwhile, Cisco Webex provides up to 50 minutes in the free version and 24 hours in the largest service package.

Microsoft Teams offers its services for up to 60 minutes in the free version available to all users and up to 30 hours of video conferencing experience in the largest service package from Microsoft.

In turn, Google Meet provides an incredibly unique opportunity to hold a conference for up to 24 hours for free versions and all other paid packages.

The open-source platforms BigBlueButton and Jitsi Meet have no limitations, but everything depends on the server's system capabilities.

Storage The allocated space for storing data for each user varies significantly between all technologies.

In the free version, Zoom provides storage to local device memory, but in contrast, in its largest set, it offers data storage without restrictions and limits.

Google Meet offers 30 gigabytes for each user in the free version and up to 5 terabytes in the largest set.

Cisco Webex allows 1 gigabyte to be used for users with the free version and up to 50 gigabytes in the largest package with the option to create a special request for even more space for the user.

Microsoft Teams allows users to use up to 5 gigabytes without choosing a particular subscription plan and up to 1 terabyte for those who have chosen the

largest one.

The BigBlueButton and Jitsi Meet platforms can allocate as much storage per user as the server hosting all video conferences will allow.

Whiteboard

Today, the whiteboard tool frequently presents information more understandably, engages conference participants, and adds interactivity. Various platforms offer advanced whiteboard functionality where multiple conference participants can contribute simultaneously.

Analyzing the functionality of evaluated platforms, only Google Meet lacks built-in whiteboard functionality, but there is the option to add this feature through extensions available on the Google Meet platform [80].

Document upload

Uploading documents is quite an important feature in today's world of various documents and manuals. The development, refactoring, or discussion of such files occurs within teams. Video conferences have significantly facilitated the analysis of documents within remote teams. Today, many different platforms already have such functionality.

Zoom, Cisco Webex, Microsoft Teams, and BigBlueButton have built-in document uploading capabilities during video conferences, which substantially facilitates teamwork at a distance. This helps all conference participants focus on the discussed material.

Google Meet does not have such a feature. To share any document, you must attach it before the conference begins, either by attaching it to the conference itself or using Google Calendar.

Jitsi also lacks built-in document uploading, but the Etherpad extension can solve this problem by being integrated into the platform.

Co-browsing

Co-browsing functionality is a very convenient and specific tool for conducting conferences. In a team setting, participants can simultaneously work on a project or write code through the video conference window without using third-party programs. Co-browsing options are uncommon in various video conferencing platforms, but they sometimes significantly simplify teamwork.

Among the compared technologies, only Cisco Webex has built-in Co-browsing functionality. All other platforms require the installation of extensions or additional applications for co-browsing during video conferences.

Cloud-based

When a system is Cloud-based, it comes with numerous advantages that positively influence its overall performance:

- Accessibility, enabling connection from any device.
- Ease of use and access, as all data is stored in the cloud and protected.
- Capability to process a large number of requests using the technical capabilities of the servers on which the platform is based.

Regarding the analyzed technologies, all commercial solutions, such as Microsoft Teams, Cisco Webex, Zoom, and Google Meet, are cloud-based platforms that play a crucial role in interacting with other company services.

Jitsi is not entirely like others. It provides the option to use JaaS servers or set up a custom server connected to the cloud-based system.

BigBlueButton is not a cloud-based platform, but like Jitsi, it can be accessed via the Internet and easily integrated into an existing cloud-based system.

1.2.5 Main topics in creating video conference systems

Video conferencing systems are now widely used worldwide. Many companies create and support video conferencing solutions to attract as many users as possible. To do this, new functionality is added, security is ensured, and various entertainment goodies are added. All this is done to provide the most optimal product for multiple purposes.

When creating a system for video conferencing, you need to take into account aspects that are present in almost all popular solutions today [81].

- Features.
- Security.
- Interactivity.
- Video Productivity.
- Artificial intelligence.

Features

In today's market of distance communication platforms, the main factor in competition is the set of operations or functions that the technology can provide. Based on the platform's functionality, organizations looking to choose a specific solution can determine whether it suits their needs.

The main options that today's video conferencing platforms possess are:

- Chat.
- Audio signal.
- Video signal.
- Ability to use third-party devices for signal transmission.
- Recording video conferences.
- Ability to share screens.
- Ability to manipulate device activity (microphone on/off).
- etc.

It is also very important to consider the quality of execution of the existing functionality on the platform.

Security

Security is crucial when choosing a platform or creating one's own. This is a highly relevant issue today, and users are often wary of entrusting their confidential data to unverified and unprotected services.

Security is responsible for preventing the loss or theft of user information and ensuring privacy and role hierarchy during video conferences.

For effective and reliable management and preservation of data, the platform should utilize:

- End-to-end encryption.
- Multi-factor authentication and authorization.
- Use of temporary encrypted tokens.
- Support for security protocols.

Interactivity

The aspect of interactivity allows users to provide a better user experience with the platform, stay actively engaged in conferences, and potentially attract them to continue using this platform. It enables effective and rapid problem-solving efforts and facilitates collaborative planning for future organizational events.

Examples of such features in a communication system could include:

- **Emotions.** Expressing emotions during video conferences through emojis or various signs to convey agreement or disagreement about the ongoing conversation.
- **Virtual Masks.** Masks are used as entertainment, replacing a person's face with a virtual mask.
- **Polls.** Facilitating the organization of choices among existing participants to provide clear statistics on all polling options.
- **Stickers, GIFs.** Sending stickers or GIF images to diversify the chat.
- **Co-browsing.** Collaboratively solving favorite tasks or any tasks in the virtual space with multiple conference participants.
- **Virtual background.** This functionality allows for highlighting the silhouette of a conference participant, replacing their background with a chosen image, or applying an effect.
- **etc.**

Video Productivity

The video signal is the central aspect of the video conferencing concept. Based on today's trend of creating platforms for such communication, platforms strive to deliver increasingly clear, beautiful, high-resolution images. Unfortunately, there may often be situations where users have a less-than-optimal connection, causing packets of important information to simply not reach the participant, negatively impacting the user experience of using the platform.

An incredibly important characteristic of the platform is the automatic detection and correction of poor signals. Additionally, during video conferences, users often have the option to manually select the configuration of the video connection quality, which can sometimes play a crucial role during a meaningful conversation.

Artificial intelligence

Given today's trend of active development and popularization of artificial intelligence, more and more companies are integrating artificial intelligence into their communication platforms.

In this case, it can be used not only to increase the level of interactivity on the platform but also to process audio and video signals better and optimize the most important and labor-intensive processes.

Artificial intelligence can also be used to expand or create new functionality.

Examples of such could be:

- Translator during a video conference in text or audio format.
- Overlaying various visual effects. With the help of artificial intelligence, this can be done much more qualitatively.
- Gathering analytics based on data collected from conferences.
- etc.

1.2.6 Future of video conferences

Many individuals, organizations, and companies utilize these technologies to create and conduct video conferences daily. Today's popular platforms may seem to have reached their functional maximum, and there might appear to be no further room for development. However, this is not the case. Numerous aspects should and even need improvement.

Innovations shaping the future of video conferencing include [82]:

- **Enhanced Security.**
- **Optimization.**
- **Interactivity.**
- **Cutting-Edge Technologies.**

Enhanced Security

Security comes first. Private information is a very important commodity that can affect multiple life processes. Therefore, many engineers work daily to improve the security of applications, including video conferencing platforms. But despite the efforts of many companies working on the safety of their products, more and more new ways of obtaining information are appearing, so the continuous development and support of system security is significant [81][83].

Enhanced Optimization

As the application develops, more and more new plugins and components are integrated. This means an expansion of the entire system architecture, significantly affecting performance. Optimization has a positive impact on the operation of the whole system, allowing it to improve not only performance but also how technologies can be further integrated into applications [81].

Interactivity

Interactivity makes it possible to make the user experience on the platform comfortable and exciting. This allows conference participants to be interested in using the platform without obstacles. Considering that today, more and more tools and functions are appearing, it is imperative to add new ones and improve old ones based on the trends and needs of users of this type of platform [81].

Cutting-Edge Technologies

The addition of various new technologies allows us to improve the user experience many times over and create innovation in using platforms for video conferencing. Examples of such technologies could be [81][82]:

- **Artificial Intelligence.** Adding artificial intelligence, for example, for instant translation or transcription of a speaker's words into a foreign language, is another way AI can optimize existing functionality, such as improving sound filtering or video quality in poor internet connections. At present, it is already being used in various elements of platforms, and in some time, this technology will be tightly integrated into many systems [84].
- **Virtual Reality, Augmented Reality.** Technologies of augmented or virtual reality are actively developing today, and the integration of these elements

into the process of conducting video conferences is quite logical and predictable [85].

1. **Virtual Reality.** For conducting video conferences, special virtual rooms or spaces can be created where participants cannot only talk but also show something with their hands or, for example, draw on a board. This will improve the platform participants' sense of presence in the same room.
2. **Augmented Reality.** This method of implementation is currently quite complex. Still, the idea itself that an avatar or the participant themselves could be added to real space using a hologram or displayed in augmented reality glasses allows enthusiasts to work more diligently on such solutions.

1.2.7 Conclusion about video conference technology

Video conferencing technologies have profoundly changed communication and access to information. This potent tool allows for learning, obtaining up-to-date information, and working in teams from different locations.

Video conferences enable us to perceive information better and work as a team.

The importance of this tool was once again demonstrated during the pandemic when millions of people were forced to switch to remote forms of learning, work, and communication. The pandemic became a real test for people and played a significant role in ensuring communication.

Communication platforms for conducting video conferences are progressive technologies and tools constantly evolving, improving, and optimizing. Each day, these platforms become more accessible to attract the most significant number of users and organizations.

2 Synthesis Part

Today's market for long-distance communication systems is an environment for developing technology and innovation. Such systems are based on various platforms and technologies that allow people to communicate at a distance using the Internet [43].

Communication systems of this type are often used in ordinary communication and business, organization, and education. One of the most commonly used platforms is for online video conferencing, which in turn has proven itself very well during the coronavirus (COVID-19) [34]. An essential aspect of such systems is the launch methods or availability on different types of operating systems [37]. Often the more operating systems are supported, the greater the reach of users it may be of interest to. An excellent solution is a web application that is accessible using a regular browser.

This part of my thesis describes in detail the creation of my communication platform, a web application for video conferencing, and the integration of Jitsi Meet and BigBlueButton technologies.

2.1 Web application

A web application is a program or application that runs directly in a web browser by connecting to a server where all the application logic is hosted.

Today, many businesses, organizations, and companies use web applications to create their solutions because it is a simple and understandable way of communication between the service and the user. Applications of this kind not only allow sending forms to the server or dynamically changing display components but also creating, editing files, exchanging information with other users of the platform, and much more [86][87].

There is a concept of a dynamic web page. This is a regular web page that will change its content to be displayed depending on the user's actions [88].

Depending on the tasks and the complexity of functionality, web applications require [87]:

- **Client-side:** It displays application information, design, animations, and navigation between components.
- **Server-side:** It runs any logic or handles requests from the client side and then returns a ready response.
- **Database:** A remarkable collection of data with its schema of dependencies that stores records about various users, metadata, activities, etc.

In general, web applications operate according to the following principle [86]:

1. A user on the application's client side sends a request to the server side after a significant event is triggered. This could be, for example, sending a registration form or clicking a button to delete any data.
2. The web application server receives the request, validates it, verifies it, and checks for access to carry out any operation for this user.
3. The server performs a task, for example, updating a database or running an algorithm for mathematical calculations.
4. The server generates a response and sends the required data back to the client.
5. A visual confirmation appears on the client side, indicating the successful completion of the task.

Web applications have many advantages that make them stand out from other applications. The main aspects are [89][87]:

- **Accessibility.** Web applications are accessible from anywhere with Internet access. You do not need to download applications to your device to do this. Everything is available online.
- **Easy to use.** To work with a web application, you often do not need any additional knowledge because we try to make the functionality as simplified as possible to obtain a good user experience.

- **Development.** Web applications can be expanded simply by creating various services tailored to the robot with a specific task for the entire system.
- **Updates.** Web applications automatically receive updates that include not only newly added functionality but also improvements in design, styles, and optimization. All this is updated on the server on which the application is running, and users eventually have the current version.

DTMeet Web Application

Based on the points of my thesis, one of the main tasks is to create a communication platform called **DTMeet** by integrating Jitsi and BigBlueButton video conferencing technologies. The creation of the web application was divided into four parts:

1. Client-side of application (Front-end) - creating an application client using technologies:
   - HTML - HyperText Markup Language.
   - CSS - Cascading Style Sheets.
   - React
2. Server-side of application (Back-end) - creating the server side of the application together with the database using:
   - Java 17
   - SpringBoot 3.1.4
   - MySQL
3. Integration of technologies for video conferencing into the client part:
   - Jitsi Meet
   - BigBlueButton
4. Deploy - deploying the application for access via the Internet:
   - Docker
   - Ngrok

The principle of operation of a web application

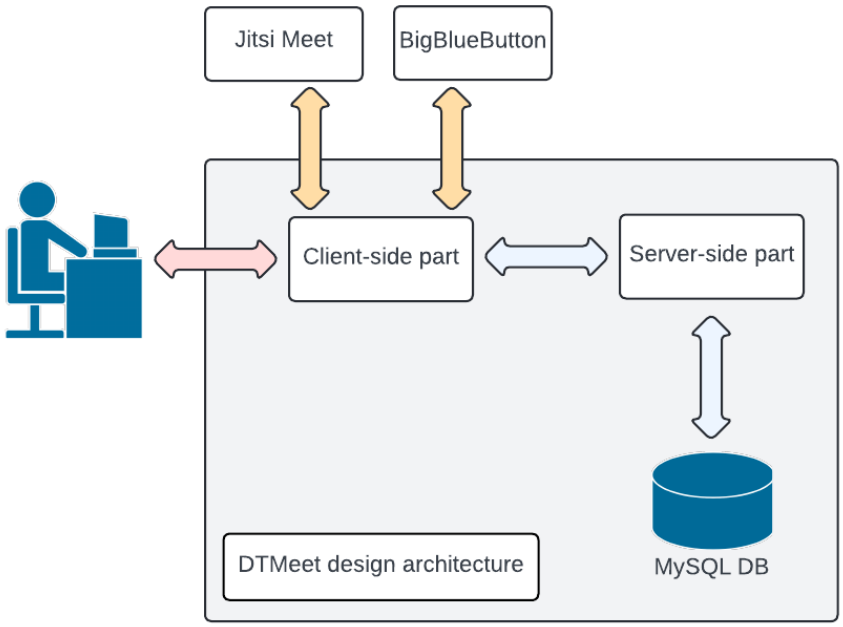

Figure 2.1: DTMeet design architecture

The Client-side part of the web application is connected to the server part as well as the BigBlueButton and Jitsi Meet servers. The server part, in turn, fills the database and sends the required data from the database to the client part.

Using the example of a scenario for creating a video conference, let's look at how the application works:

1. The user enters the details for creating a video conference, name, password, etc.

2. The data entered by the user into the client part is sent to the server where the user, data, permission to create or connect to the conference are checked. At this stage, communication with the database and updating of data occurs.

3. After all checks, the requested action is performed by the user and the response is sent further to the client part.

4. If the previous request is successfully processed, the client part sends a request to the video conference technology server.

5. If the response has success status, a window with a video conference from the selected technology for video conferencing, Jitsi or BigBlueButton, is displayed on the client side. This is the last part where the user can freely communicate with other participants in video conferences.

Web-application Security

Web application security plays an essential role in the operation of any application. Data breaches are relatively common on the Internet, and they can have drastic consequences not only for ordinary platform users but also for severe organizations, governmental bodies, and educational institutions. Platform users should be confident that their personal information, passwords, addresses, and other data will be protected.

Various functions and features are included in the DTMeet platform itself, which aims to ensure the safe and comfortable use of the entire system.

- **Input fields on the client-side application.** All fields where users input data undergo validation based on the expected data type. For instance, if a field is meant for the user's email address, regular expressions will control its state. This principle operates in all application areas where users can input their data.
- **Validation of received requests.** When a request is received on the server-side, data validation occurs using additional libraries at the Controller stage.
- **Encryption of database records.** Different data, passwords, and keys are protected by one-way encryption.
- **JSON Web Token (JWT).** JWT is used to access user accounts and services. Users can access data by sending a temporary encrypted key along with the request body.

  Supporting JWT tokens allows users to stay logged into the application continuously, significantly improving the user experience. If the JWT token expires, the DTMeet application uses a refresh token to renew the JWT web token for further application operation.
- **Video conferencing technologies.** JitsiMeet and BigBlueButton are inherently secure systems due to their internal encryption keys and end-to-end encryption during conferences.
- **Invitation link.** The link is encrypted with a unique one-way key when generating a link to invite users to a conference via the DTMeet platform. A checksum attribute is added to the main invitation link, to which an encrypted link is assigned.

This allows the application to verify whether the link was indeed generated by the system rather than by an external user, thereby controlling participants' access to the video conference.

2.1.1 Client-side part

The web application's client side is the entire application's main user interface. The user's device displays everything that happens in the whole application using text, animations, and various user interface elements [88].

In modern development of the client side of web applications, the following stack of tools is mainly used [88]:

- **HTML - HyperText Markup Language.** Today, almost all web applications are created using the hypertext markup language HTML. The main function is to indicate the structure of the entire web page using tags and blocks.
- **CSS - Cascading Style Sheets.** The CSS tool - Cascading Style Sheets - also plays an important role. Cascading Style Sheets Allows you to designate a document and add animation.
- **JavaScript / TypeScript.** The programming languages `JavaScript` and `TypeScript` are often used during web application development. They are responsible for the logic that occurs during any user action. The range of functions is incredibly large; it can be used for more complex animation of details while using the application, sending requests to the server part or other services, processing and parsing data [90].

  `TypeScript` complements the JavaScript programming language, providing typing, more dynamic behavior, and security during development.

DTMeet Client-side design architecture

Since video conferencing technologies `Jitsi` and `BigBlueButton` allow users to use their functionality without authorization, adding this feature to DTMeet was very important.

The main difference between an authorized and unauthorized user is that an authorized user has additional tools, such as creating rooms for collaborating with other participants, thereby saving the state of conferences and their participants and collecting statistics.

This approach may interest small teams and ordinary Internet users who want to hold a conference quickly without worrying about the complications of setting up a video conference.

The client side of the DTMeet application was built using the stack:

- **HTML - HyperText Markup Language.**
- **CSS - Cascading Style Sheets.**
- **React framework.** React is a front-end web application development library that is widely used for developing dynamic web applications.

  The idea is that developers can use the same components when creating applications using their flexible configuration. This made it possible to use the **Virtual DOM - Document Object Model**. When you change any application component, the entire application is not rerendered; only the necessary part depends on the user's actions, thanks to the `virtual DOM` feature.

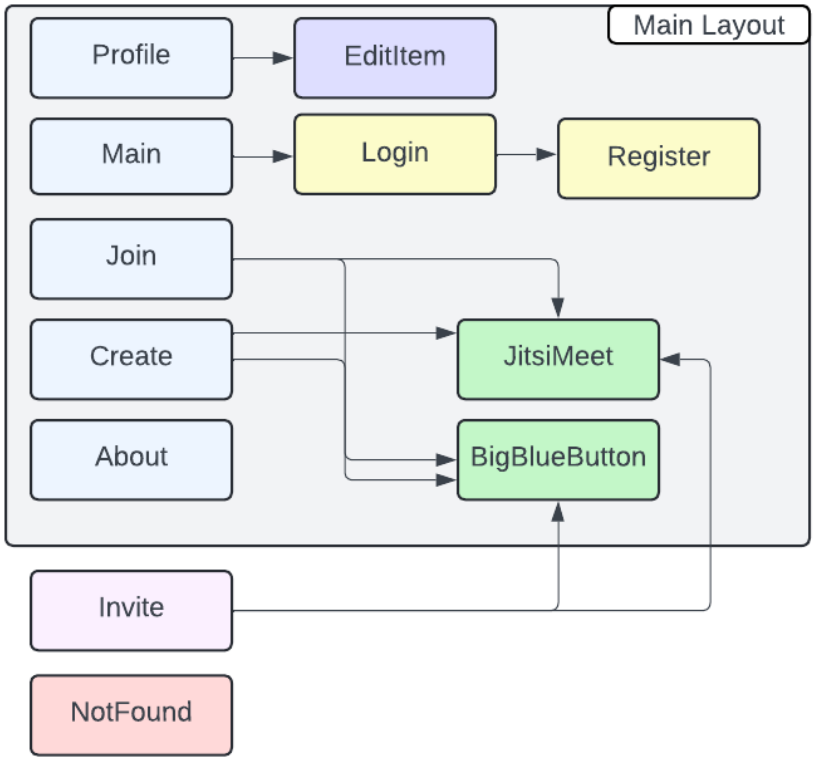

Figure 2.2: DTMeet client-side part of server architecture

The DTMeet web application uses routing, breaking the application into components and stages. The client part of the web application consists of seven main components that the user can refer to:

- Profile.
- Main.
- Join.
- Create.
- Invite.
- About.
- NotFound.

The practical implementation of the navigation routing functionality in the application is in `App.jsx`:

```
<BrowserRouter>
  <div className="App">
    <Routes>

      <Route path='/' element={<MainLayout props={...}/>}>

        <Route path='/' element={<Main props={...}/>} />
        <Route path='*' element={<NotFound/>} />
        <Route path='/about' element={<About/>} />
        <Route path='/signin' element={<Login props={...}/>} />

        <Route path='/profile' element={<Profile/>} />
        <Route path='/profile/edit'
          element={<EditItem props={...}/>} />

        <Route path='/create' element={<CreateForm props={...}/>} />
        <Route path='/create/BigBlueButton' element={<BBB/>} />
        <Route path='/create/Jitsi' element={< Jitsi/>}/>

        <Route path='/join' element={<JoinForm props={...}/>} />
        <Route path='/join/BigBlueButton' element={<BBB/>} />
        <Route path='/join/Jitsi' element={<Jitsi/>} />

      </Route>

```

```
      <Route path='/invite' element={<Invite props={...}/>} />

    </Routes>
  </div>
</BrowserRouter>
```

Listing 2.1: DTMeet web-application routing. App.jsx

This piece of code describes the routing between the various components of the web application and has a few key values to understand:

- `<Route />`. Initializes a navigation element that contains data about the domain path to the component and the element itself.
- `path = " "`. Initializes the domain path for displaying the element in routing.
- `element = <YourComponent />`. This is a component that will be called based on the given path.

Main Layout

The `Main Layout` is a component that is a template for the entire web application. This indicates the interface elements available in all child components in the hierarchy. Also, this component has a stylization of the main features of child components.

The practical implementation of MainLayoutComponent:

```
<>
  <div className={styles.header}>
    <Menu panelStatus={panelStatus} onChangePanel={onChangePanel}/>
      {auth.id ?
      <Avatar showToast={showToast}/>
      :
      <></>
    }
  </div>

  <Outlet/>
</>
```

Listing 2.2: DTMeet web-application Main Layout component . MainLayout.jsx

The main purpose of the component is to display template components that will be present as part of the `virtual DOM` in child components. However, since the application is intended to be functional for an unauthenticated user, hiding some interface parts is very important.

- `<Menu />`. This component is a sliding window with navigation buttons. If the user wants to switch to another part of the application, for example, view a profile or create a video conference, they need to open the menu to select the desired path.
- `<Outlet />`. This component is the rendering marker of child components in the `virtual DOM`. Component imported from `'react-router-dom'`.
- `<Avatar />`. This component looks like a button in the top right, which opens a dialog box with a button to log out of the account or go to the profile part of the active user's component.

Profile

The `Profile` component stylizes an authorized user's data and allows users to update the data of a user-selected field.

```
<div className={styles.InfoContainer}>

  <div className={styles.blockContainer}>
    <div className={styles.titleText}>
      <h2>Basic Info</h2>
      <h3>Some info may be visible to other people using DT Meet.</h3>
    </div>

    <InfoItem title={'Name'} blockValue={auth.name}/>
    <InfoItem title={'Surname'} blockValue={auth.surname}/>
    <InfoItem title={'Username'} blockValue={auth.username}/>
    <InfoItem title={'Birthday'} blockValue={'1.01.2002'} lastItem={
   true}/>
  </div>

  <div className={styles.blockContainer}>
    <div className={styles.titleText}>
      <h2>Contact Info</h2>
    </div>
      <InfoItem title={'Email'} blockValue={auth.email}/>
      <InfoItem title={'Phone'} blockValue={auth.phone_number}
   lastItem={true}/>
```

```
21   </div>
22 </div>
```

Listing 2.3: DTMeet web-application Profile component . Profile.jsx

This class describes the display logic of the `Profile` component. Data is collected from the `auth` object and displayed directly using the component.

The `<InfoItem />` component is a block upon action with which the user will be redirected to the stage of editing the selected object.

Main

This is the central component of the web application and performs several functions.

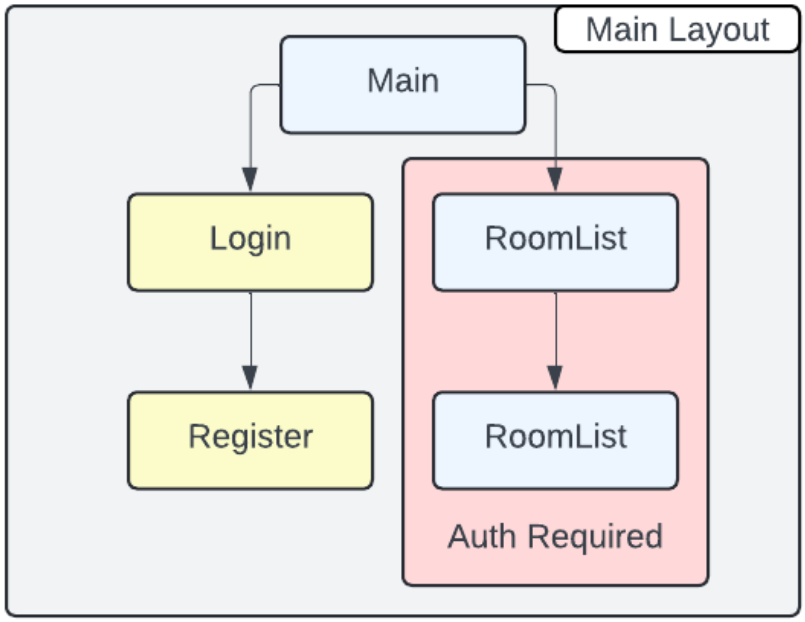

Figure 2.3: Front-end Main component diagram

When switching to a web application, a check is initially made to see if the user is authorized based on the presence of the `JSON Web Token` in the browser storage.

If such a token exists, a request with this token is sent to the server part to obtain user data about the authorized user. After receiving a response from the server, the client part displays the `<RoomListComponent />` component, which contains all the rooms in which the user participates.

Suppose the user is logged out or some error occurs during data processing to hide important information. In that case, the user will be forced to switch to the logging component in the application. If this user has no account, he can register using <RegisterForm />.

```
<>
  {auth.id || localToken != null ?
    <RoomListComponent showToastEvent={showToastEvent}/>
    :
    <>
      <h1>Main</h1>
      <h2>This is main page of This App</h2>
      <h3>Rooms just for authorized users</h3>
      <SimpleButton hadleButtonFunction={handleButtonClick} btnText={"
    Sign in"} />
    </>
  }
</>
```

Listing 2.4: DTMeet web-application Main component . Main.jsx

This code describes the functionality of the <Main /> component. The status of the localToken or auth.id attribute is checked. If this condition is true, the user will see all the rooms he has been added to, allowing him to continue communicating with participants in video conferences. Otherwise, the user will see a prompt to log in or register.

Join

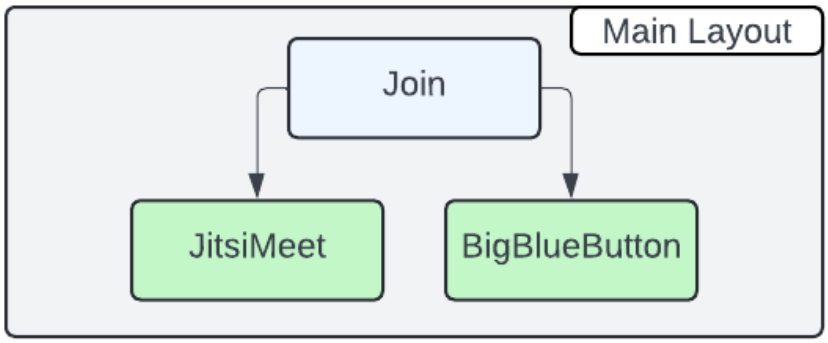

Figure 2.4: Front-end Join component diagram

The Join form has two fields:

- **Username**. Name of user.
- **URL**. Link of video conference.

If authorized, specifies only the input field for the video conference link for the connection. The username must also be entered if the user is not authorized.

Thanks to the regular expression tool, the entered link is validated to determine whether the application supports this technology for video conferencing.

If such a technology is supported, a redirect to the `Jitsi Meet` or `BigBlueButton` technology component is made, and the decision is also made using the same regular expression.

Otherwise, a message stating that this technology is not supported will be displayed.

```
<>
  <h2>Join To Meeting</h2>

  <form onSubmit={onSubmitHandler} className={styles.form_container}>
      {auth.id ?
        <></>
        :
        <Input labelText={"Username"} props={...} />
      }

      <Input labelText={"Url"} props={...} />

      <SubmitButton btnDisabled={btnStatus} btnText= {'Join to meeting
    '}/>
  </form>
</>
```

Listing 2.5: DTMeet web-application Join component . JoinForm.jsx

The form is a fairly convenient tool for filling out data, validating data, and processing it further

Hiding the username field is controlled by the `auth.id` value, if the user is logged in, he does not need to enter a username. It will be taken from the active user's data.

After clicking on the `<SubmitButton ... />` component button, the `onSubmitHandler` function will be executed, which in turn updates the database and redirects to the component of the selected technology for video conferencing.

Create

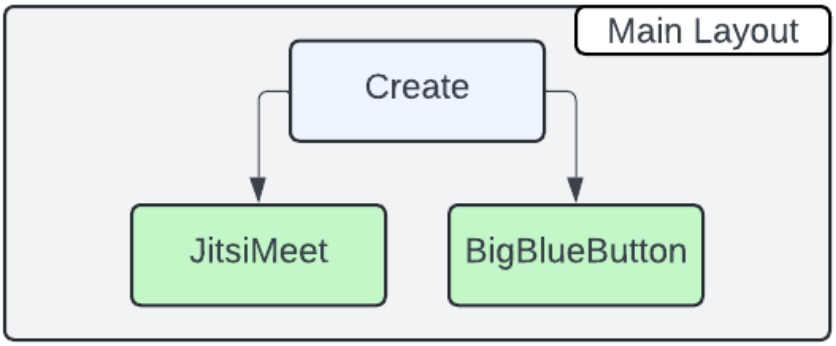

Figure 2.5: Front-end Create component diagram

The `Create` component is responsible for creating a video connection in the technology platform selected by the user. The component is a dynamic form with fields for entering the necessary credentials to create a video conference. Based on the desired technology, the video form changes.

Fields for creating video conferences:

- **Jitsi Meet**
  1. User Name.
  2. Meeting Name.
- **BigBlueButton**
  1. User Name.
  2. Meeting Name.
  3. Attendee Password.
  4. Moderator Password.

```
<form onSubmit={onSubmitHandler} className={styles.form_container}>
  <h2>Create Meeting</h2>
  <div className={styles.toggleContainer}>
    <h3>Jitsi</h3>
    <div className={styles.btnsContainer}>
      <ToggleBtn toggleBtnChange={onToggleBtnHandle}/>
    </div>
    <h3>BigBlueButton</h3>
  </div>

```

```
  {auth.id ?
    <></>
    :
    <Input labelText={"User Name"} props={...}/>
  }

  <Input labelText={"Meeting Name"} props={...}/>

  {urlData.technologyName === Technologies.BBB &&
    <div>
      <Input labelText={"Attendee Password"} props={...}/>
      <Input labelText={"Moderator Password"} props={...}/>
    </div>
  }

  <div className={styles.btnsContainer}>
    <SubmitButton btnDisabled={btnStatus} btnText={'Create meeting'}/>
  </div>
</form>
```

Listing 2.6: DTMeet web-application Create component . CreateForm.jsx

The <ToggleBtn ... /> component is part of the user interface, ahead of the selected `Jitsi Meet` or `BigBlueButton` technology. The choice of technology specifies the fields that will be processed in the form, as well as the data that will be sent to the server.

In this case, a decision is made whether the selected technology needs to accept information about the attendee password and the moderator password, which is important only for `BigBlueButton`.

This component also checks whether the user is authorized by the `auth.id` attribute. This is required to determine whether the user should specify his name or whether the data is already present.

Invite

This component is important in connecting the user via an invitation link. Such a user immediately has two authorization options:

- **Using account credentials.** The username will be taken from the information about the authorized user.
- **Without Authentication.** The user needs to enter a username to start a video conference.

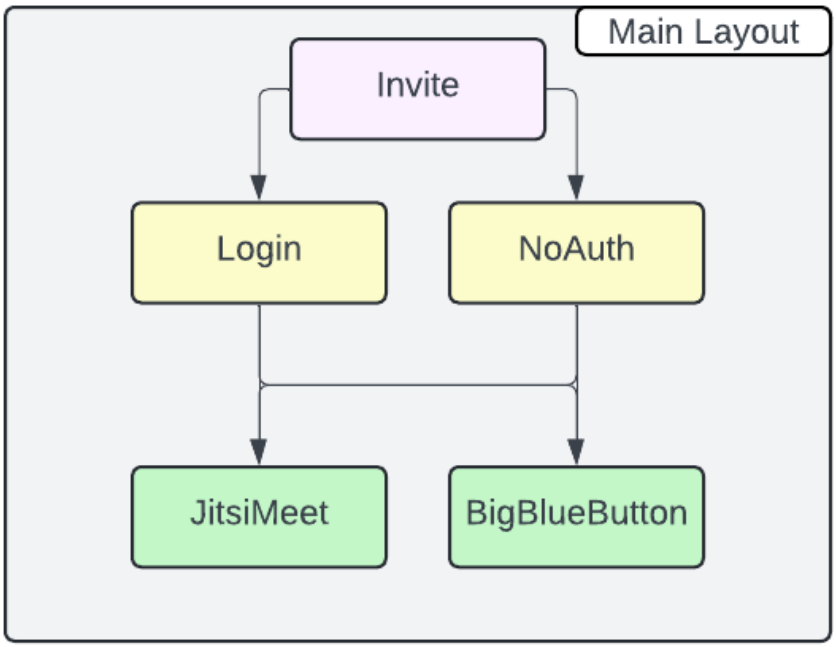

Figure 2.6: Front-end Invite component diagram

This choice allows users to become participants in a video conference without additional settings or registrations.

The link used for the invitation contains data about the invitation and an encryption key, which is first validated to determine whether the application generated the link, and if so, the invitation page is rendered.

After determining the technology and all the necessary data to connect to the conference, the user is redirected to the component of the selected communication technology, `Jitsi Meet` or `BigBlueButton`.

```
<div className={styles.invite_container}>
  {validateInvitationUrl(new URL(window.location.href)) ?
    <>
      {noAuthStatus === true ?
        <div>

          <div className={styles.back_btn_container}>
            <Button label={arrowBackIcon} props={...}/>
          </div>
          <form onSubmit={onSubmitHandler} >
            <Input labelText={"Username"} props={...}/>
            <SubmitButton btnDisabled={btnStatus} btnText={"Connect"}/
   >
          </form>
        </div>
```

```
15          :
16          <>
17            <form onSubmit={onLoginHandler}>
18              <h2>Login</h2>
19
20              <Input labelText={"Login"} props={...}/>
21              <SecretInput labelText={"Password"} props={...}/>
22
23              <SubmitButton btnDisabled={btnStatus} btnText={"Connect"}/
    >
24            </form>
25
26            <div className={styles.separator}>
27              <div className={styles.separator_line}></div>
28              <div className={styles.separator_text}>or</div>
29              <div className={styles.separator_line}></div>
30            </div>
31
32            <SimpleButton btnText={'Connect without auth'}
    hadleButtonFunction={onNoAuthMode}/>
33
34          </>
35
36        }
37      </>
38      :
39      <h1>Wrong invite link, contact to administrator</h1>
40    }
41  </div>
```

Listing 2.7: DTMeet web-application Invite component . Invite.jsx

The `validateInvitationUrl()` function checks whether the link has been generated; if it is, then two forms will be shown for authorization or connection without authorization.

About

The `About` component displays information about the project. The user can read about the goals of the web application, how it works, and the main functionality.

The program code is the display of information for users.

NotFound

This component handles a non-existent component path. If such a component does not exist based on the browser URL, instead of the default not found page, information about the non-existent component will be displayed, but it will already be styled as a web application.

2.1.2 Server-side part

Web applications that process big data or have a database often use a server part that is responsible for such operations [91].

This allows you to display a large layer of information from the stored database ID to the user on the client side of the application [88]. The server allows you to display data on the client side that will be needed specifically for this user, for example, user profile data or selected products from the cart [88][91].

Previously, a large number of tasks were performed by the web application's server-side: rendering pages, processing various notifications, processing information, communicating with the database, and filtering access to services. Today, to speed up and optimize a web application, a large number of tasks are already performed on the client side of the application [91].

Communication between the client and servers often occurs using **HTTP - HyperText Transfer Protocol**. An HTTP request may include data from where the request was sent, the body of the request itself, the interpretation of the endpoint on the server side, cookies, metadata, and method. All these aspects help to communicate very simply and conveniently between the client and server parts [92][91].

The main and frequently used HTTP methods are [92]:

- **GET.** Used to receive information from the server to the client. For example, user data from a database.
- **POST.** Used to send data from the client side to the server side. For example, it is very often used to add new records to the database.
- **PUT.** Is a method of updating existing data. Used to update any records on a server or database.
- **DELETE.** The method is used to delete any data from the server side. It is also used to delete records from the database.

After processing the request on the server side, the response is returned to the client side with a response message that, in addition to the expected data, also contains data on the status of the request, whether it was successful or not [91].

The presence of a server in the architecture of the entire application allows you to store data, update it, and return it effectively, but also carry out various data processing, communicate with other servers to exchange information, and provide data to the client side in multiple formats [88][91].

DTMeet Server-side architecture

The server part of the application plays many roles in the whole platform, but the key ones are:

- Video conference access control.
- Collecting data to create analytics.
- Control access to individual application components.
- Saving data from platform platform.

The application is built on the basis of the Monolith architecture design pattern.

Technologies were used:

- **Java 17**. Programming language.
- **SpringBoot 3.1.4**. Back-end framework. External libraries:
  - `springdoc-openapi-starter-webmvc-ui.` swagger documentation
  - `jjwt-jackson, jjwt-api, jjwt-impl.` Implementation of JWT - Json Web Token functionality
  - `modelmapper.` Library for mapping data to different objects.
  - `spring-boot-starter-security.` Implementing security features and configuration.
  - `lombok.` Generates common parts of code by annotations.
  - `mysql-connector-j.` This library is important for connection beetween server and database.

  - `spring-boot-starter-web`. Web application starter environment kit.
  - `spring-boot-starter-validation`. Implements functionality for validating attributes of entities.

- **Maven**. framework for automating project assembly.
- **MySQL**. MySQL technology database.
- **Swagger Documentation**. REST API documentation.

The server part of the DTMeet web application is a `SpringBoot RestAPI` application that accepts requests from the client part manipulates data and saves or retrieves data from the database.

The entire back-end part logic of the application consists of 6 main aspects:

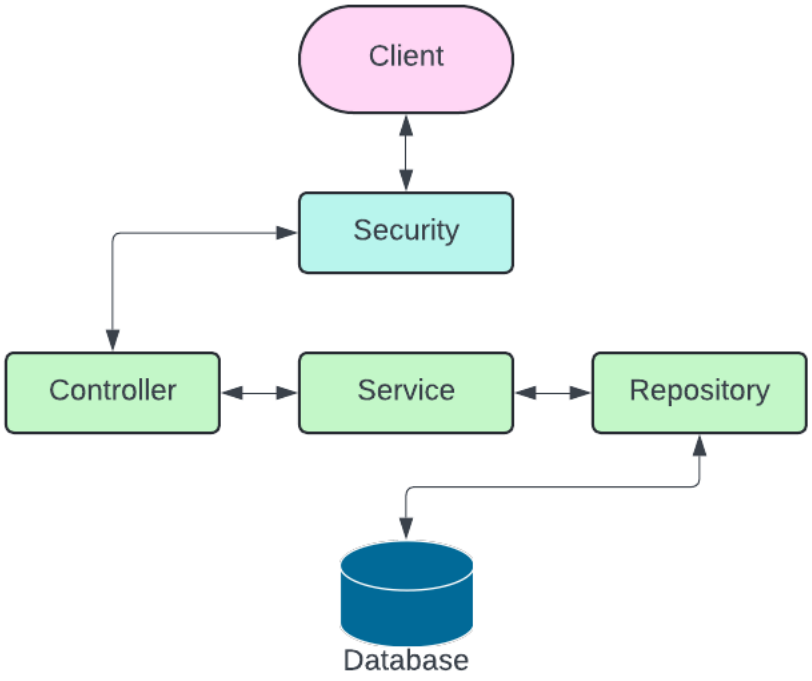

Figure 2.7: Back-end Design Pattern architecture

1. **Client.** In the diagram, it is designated as a whole client part whose task is to send requests to the server side and receive responses.
2. **Security.** Security is responsible for encrypting and managing access to endpoints and data.

3. **Controller.** In this server-side component, endpoint initialization occurs, request and response typing are defined, and the HTTP request method is determined.
4. **Service.** Defining the functional logic of the endpoint, calling the repository to retrieve data from the database, data processing, mapping.
5. **Repository.** Responsible for creating and sending transactions to receive, update, and add data to the database.
6. **Database.** Acts as reliable data storage. It has its scheme, rules, functions, and triggers.

SpringBoot application structure

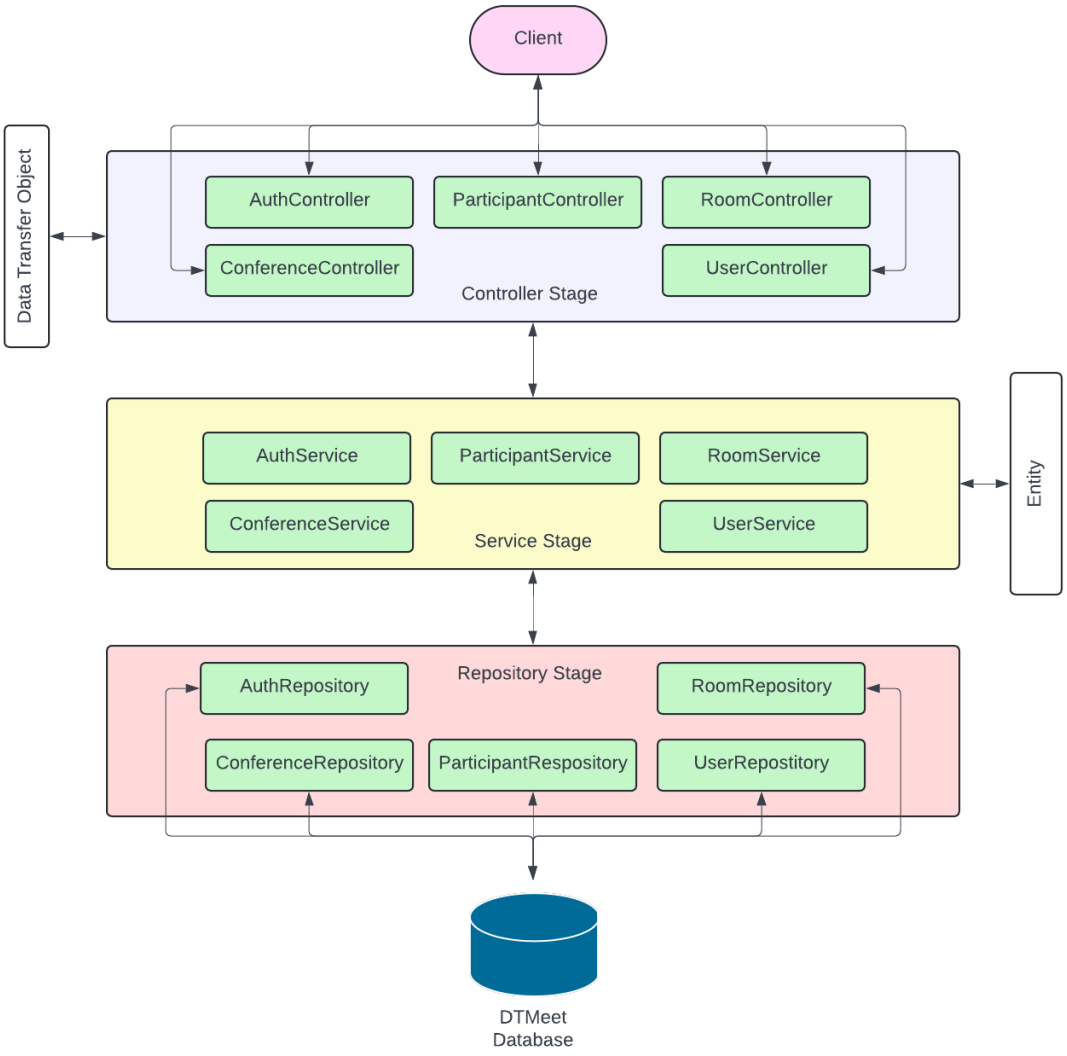

Figure 2.8: Back-end business logic architecture

The application is built based on the Controller-Service-Repository pattern where:

- **Controller** is responsible for accepting requests and validating the request body.
- **Service** is responsible for business logic and functionality implementation. The service utilizes repositories from other entities to access their respective database recordings. This is also crucial for mapping the results that will be returned to the application's client side.
- **Repository** is responsible for communicating with the database, data collection, and saving it to the database.

For working with various repositories and requests, saving, parsing, and mapping data are crucial. For this purpose, the application typically employs `DTOs (Data Transfer Objects), Mappers, and Entities`.

Figure 2.9: Data transfer object to entity diagram

- **DTO - Data Transfer Object.** This special object is handled upon reaching the Controller. Data from the request body is stored in the `DTO`. Additionally, it helps perform data validation on the server side of the application.
- **Mappers.** These tools allow rewriting data from objects of different types, such as mapping from `DTO` to Entity. They are used at the `Service layer`, where all the logic happens.
- **Entity.** An object that corresponds to a table in the application database. It serves as the primary instance of business logic.

To enable the full functionality of the application, five types of REST API endpoints were implemented:

1. **Conference.** Endpoints for processing requests related to Conference operations.

2. **Authorization** Endpoints process requests with authentication, authorization, and registration.

3. **Participant.** Endpoints for processing requests related to Participant operations.

4. **Room.** Endpoints for processing requests related to Room operations.

5. **User.** Endpoints for processing requests related to User operations. All these endpoints are configured and initialized in the Controller layer.

All these endpoints are configured and initialized in the `Controller layer`.

Conference This conference business logic supports seven endpoints:

1. **POST** "api/v1/conferences" - createConference.
   Create a conference, store data about the new conference in the database based on data from `ConferenceDTO`.

2. **PUT** "api/v1/conferences/conference-id/participants/participant-id" - addParticipantToConference. Add a participant to the conference based on identifiers. Store data in the database about the relationship between the conference and the participant.

3. **DELETE** "api/v1/conferences/conference-id" - deleteConference. Delete a record of the conference from the database based on the conference identifier.

4. **GET** "api/v1/conferences/conference-id" - getConferenceById. Retrieve data about the conference from the database based on the identifier.

5. **POST** "api/v1/conferences/add-conference" - addConference. Create a new conference with a new participant having a relationship with this conference. Store data in the database based on `NewConferenceEventDTO`.

6. **PUT** "api/v1/conferences/join-conference" - joinToConference.
   Add a user to an existing conference. Store data in the database based on `JoinConferenceDTO`.

7. **PUT** "api/v1/conferences/close-conference/conferenceName" - finishConference. Finish the conference, adding data to the conference by its name indicating that it has ended.

ConferenceService is using autowiring to inject instances of repositories:

- ConferenceRepository.
- ParticipantRepository.
- RoomRepository.

Authorization Authorization business logic supports four endpoints:

1. **POST** "api/auth/login" login. User authorization on the platform. Checking if such a user exists and comparing the login and password with those existing in the database. Based on the data in the `LoginDTO`, generating a `JSON web token` and refresh token for further access to the platform.
2. **POST** "api/auth/register" register. User registration. The data received from the client-side in the form of `RegisterDTO` undergo processing, after which the information about the new user is stored in the database.
3. **POST** "api/auth/refresh" refreshJwt. Updating the `JWT` based on the refresh token to extend user access. The update occurs based on the refresh token sent from the client-side.
4. **POST** "api/auth/login-credential" getUserCredentials. Retrieving user data based on the temporary `JWT` obtained from the request header. This token is decrypted in the service layer, after which a search for the user is conducted based on their login in the database. The response containing the user's data is then sent to the client-side.

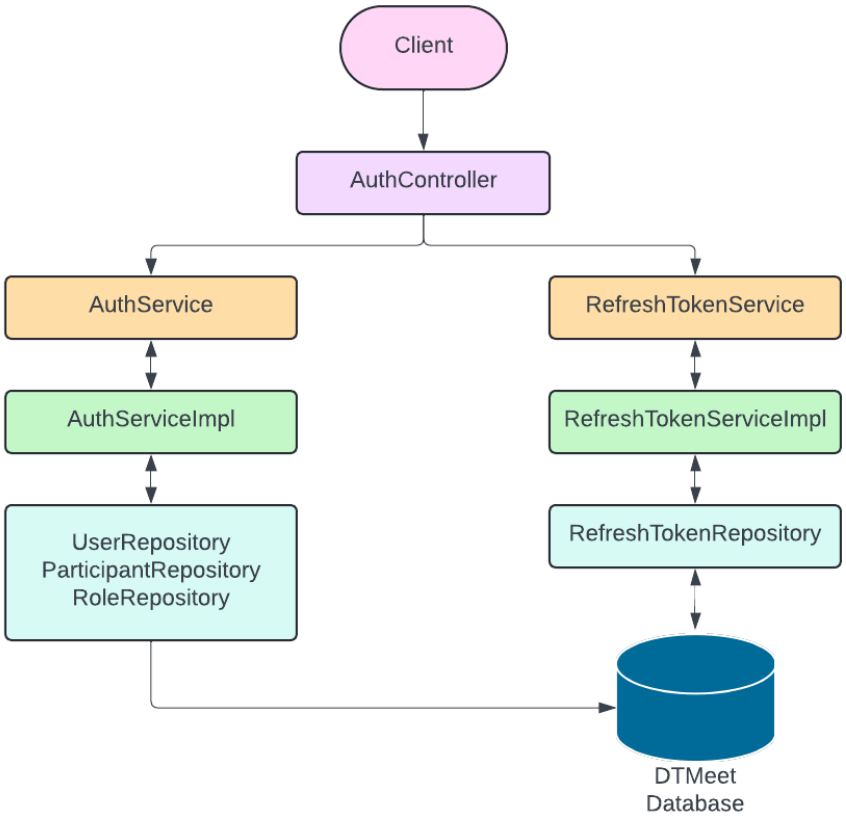

Figure 2.10: Authorization option diagram

This Controller utilizes the functionality of two services: `AuthService` and `RefreshTokenService`. This is done to separate the functionality between the Authorization Service, which primarily handles the generation of `JWT` tokens and processing platform user data, and the `RefreshTokenService`, responsible for updating and refreshing user access to the application.

`AuthService` and `RefreshTokenService` are using autowiring to inject instances of repositories:

- AuthService.
  - UserRepository.
  - ParticipantRepository.
  - RoleRepository.
- RefreshTokenService.
  - RefreshTokenRepository.

Participant Participant business logic supports four endpoints:

1. **POST** "api/v1/participants" createParticipant. Based on the data received from the client-sidepart of application in the form of `ParticipantDTO`, a new conference participant is created and this data is recorded in the database.
2. **GET** "api/v1/participants/id" getParticipandById. Retrieving data about a participant from the database using their identifier from the endpoint.
3. **GET** "api/v1/participants" getAllParticipants. Obtaining data about all conference participants existing in the database.
4. **DELETE** "api/participants/id" deleteParticipantById. Deletion of conference participant information from the database based on their identifier obtained from the endpoint.

`ParticipantService` is using autowiring to inject instances of `ParticipantRepository` to get an access to database information about participants.

Room. Room. Room business logic supports nine endpoints:

1. **POST** "api/v1/rooms/owners/owner-id" createRoomByOwner. Creating a room associated with a user. From the client side of the application, a `RoomDTO` containing data about the new room is sent to the server, along with the owner-id identifier obtained, which represents the user identifier from the endpoint. Based on this data, a connection is established between the user and a new record about the room in the database.
2. **GET** "api/v1/rooms/id" getRoomById.
   Retrieving room data from the database based on the room identifier.
3. **PUT** "api/v1/rooms/room-id/conferences/conference-id" createConferenceInRoom. Creating a relation between a room and a conference in the database based on the identifiers obtained from the endpoint.
4. **DELETE** "api/v1/rooms/id" deleteRoomById. Deleting room information from the database based on the room identifier obtained from the endpoint.
5. **PUT** "api/v1/rooms/id" updateRoomById.
   Updating room data in the database based on the room identifier.
6. **GET** "api/v1/rooms/users/user-id" getRoomsByUser. Retrieving data about all rooms from the database in which the user participates. User search is performed using the user identifier obtained from the endpoint.

7. **PUT** "api/v1/rooms/room-id/users/user-id" removeUserFromRoom. Removing a user from a room based on the user and room identifiers. The relation between these records is deleted from the database.

8. **PUT** "api/v1/rooms/add-user" addUserToRoomByEmail. Adding a user to a room based on their email, which is unique upon registration. `NewUserDTO` contains the data required to perform this operation. On the database side, a relation is created between the user and the room.

9. **PUT** "api/v1/rooms/roomId/users/userId/technologies/technologyName/add-conference" addConferenceToRoom. Creating a conference in a room with a connected user. User identifier, room identifier, and the name of the technology for conducting video conferences are taken from the endpoint. On the database side, a conference is created in the existing room, and a relation is established between the conference and the room. Additionally, a relation is established between the user and the conference.

`RoomService` is using autowiring to inject instances of repositories:

- RoomRepository.
- UserRepository.
- ParticipantRepository.
- ConferenceRepository.

User. User business logic supports five endpoints:

1. **POST** "api/v1/users" createUser. Creating a user using data received from the client-side of the application in the form of a `UserDTO`. Creating a record in the database for the new user.

2. **GET** "api/v1/users/id" findUserById. Retrieving user data from the database based on the user identifier from the endpoint.

3. **PUT** "api/v1/users/id" updateUserById. Updating data about an existing user in the database. The update information is sent in the form of a `UserDTO` from the client-side, and the user identifier from the endpoint is also used to determine the user being updated.

4. **PUT** "api/v1/users/user-id/rooms/room-id/remove" removeUserFromRoom. Removing a user from a room based on the room and user identifiers obtained from the endpoint. In the database, the relation between the room and the user is deleted.

5. **PUT** "api/v1/users/user-id/rooms/room-id/add" addUserToRoom. Adding a user to a video conference room based on the user and room identifiers obtained from the endpoint. On the database side, a relation is created between the user and the room.

`UserService` is using autowiring to inject instances of repositories:

- RoomRepository.
- UserRepository.
- ParticipantRepository.

DTMeet database

Databases are a very important component of any application. They store data for further use in various application services.

MySQL was used in the development. It is a popular open-source relational database management system (RDBMS). Reasons for choosing MySQL [93]:

- It is suitable for both small and large schemas.
- In case of system expansion or complexity of the entire schema, the database supports tools that may be necessary.
- It is supported by many operating systems used today: `OSX`, `Windows`, `Linux`.
- It has a fairly large community that can help with problem-solving.
- Provides robust security features to protect data.
- The system is very well optimized.

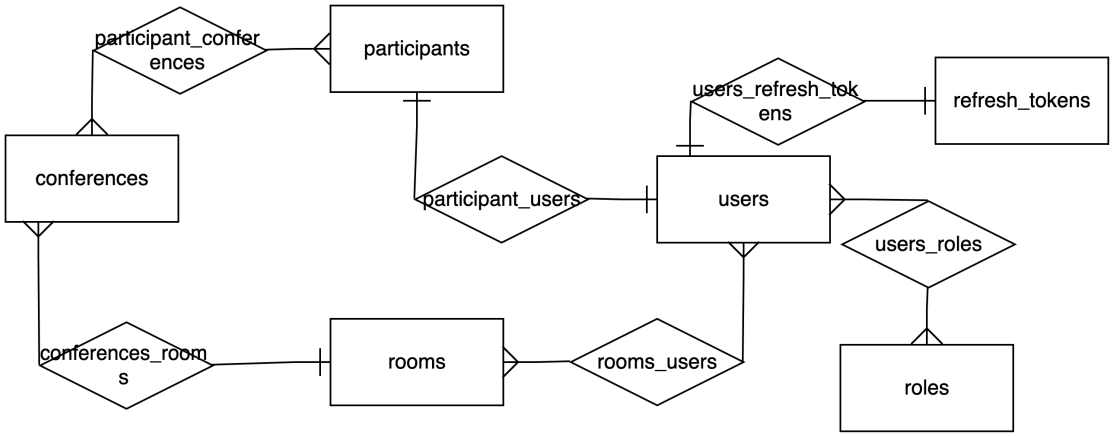

Figure 2.11: DTMeet relation diagram

Database scheme exists from six entities:

1. **users.** The user table contains user data. Records are added during the platform registration process. Data from this table is used for authentication or for mapping to response objects.

2. **roles.** The roles table is used to assign a specific role to a user and helps define access criteria for various users on the client and server sides of the application.

3. **refresh_tokens.** The refresh_tokens table stores special tokens for each user, which allow refreshing the `JWT - JSON Web Token` used to access the application's server-side functionality and for authorization on the client side. This token updates during application authorization.

4. **participants.** The participant's table contains data about conference participants during video conferences. The filling occurs at the beginning of the meeting for non-authorized users or when a new user of the platform registers. The participant's table is connected to the user's table as a `OneToOne` relationship. Each user has a record in the participants table, but not all participants have a record of a user.

5. **conferences.** The table contains data about all meetings taking place on the platform. There is a relationship between conference participants and the rooms in which conferences can take place.

6. **rooms.** The rooms table gathers records of rooms created by users. Platform users can add other users to a room for communication, thereby increasing the number of connections between the room and users.

The database schema was designed to be optimal for authorized users and those who do not want to spend time on registration and only need to make a video call using the platform.

For the conference to take place without authorization, the filling is done in the tables of participants and conferences, which are linked to each other by a `ManyToMany` relation.

An authorized user gains the ability to work with all tables due to additional functionality, such as creating rooms, adding users to rooms, and managing security keys.

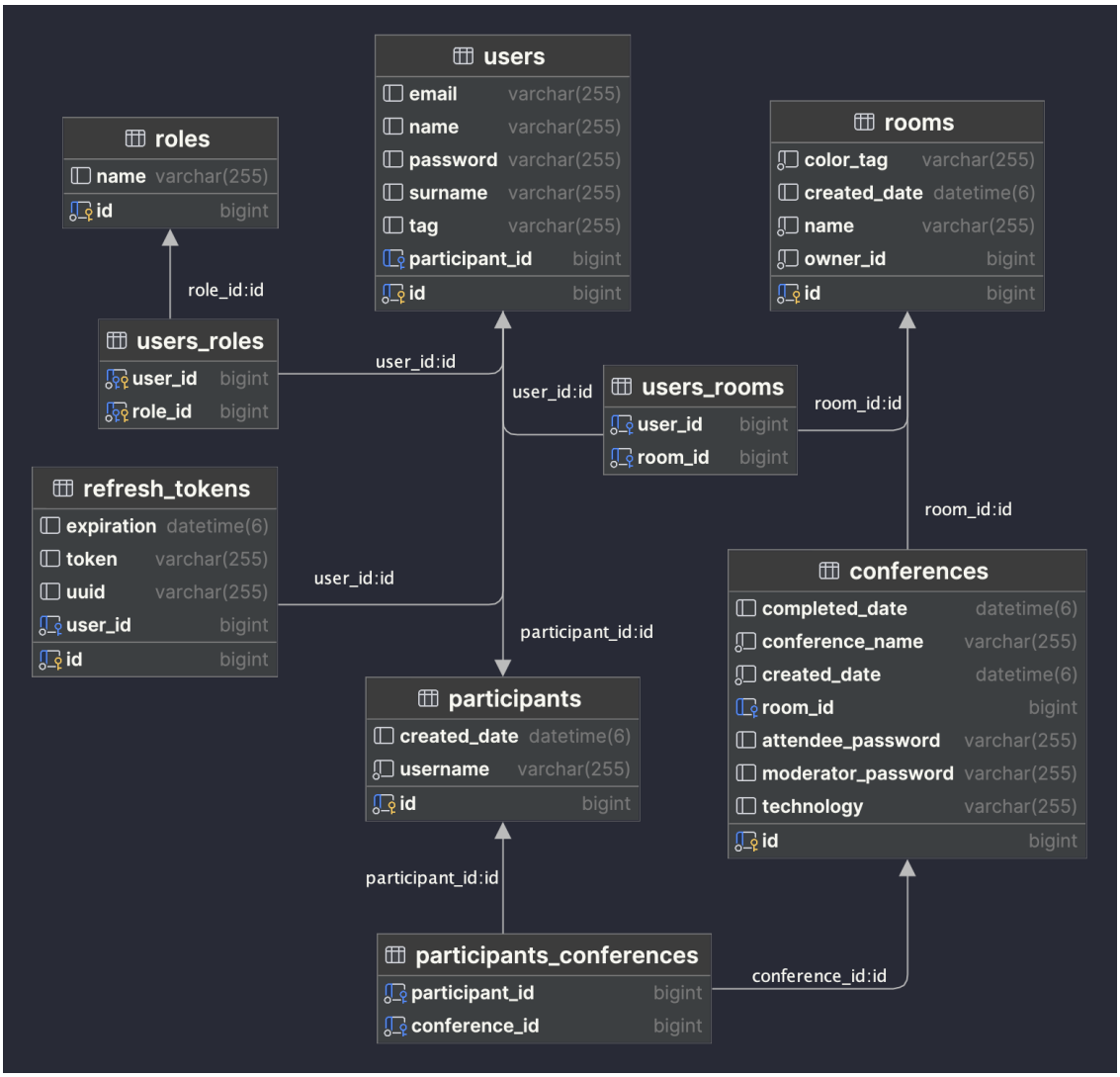

Figure 2.12: DTMeet database scheme

To provide relation in the database, tables were created that are responsible for the type of connection between entity tables:

- users_roles.
- users_rooms.
- participants_conferences.

2.2 Integration to web application

Today, there is a huge number of applications that help significantly optimize and simplify work, study, and everyday life. Integrating various technologies and systems allows the application to expand its capabilities with ready-made and stable solutions. This enables adding functionality to the application, attracting more users, and creating an alternative to other solutions.

For the same purpose, the application integrates technologies for conducting video conferences. One of the points of my thesis was the integration of `Jitsi Meet` and `BigBlueButton` technologies.

`Jitsi` and `BigBlueButton` are open-source technologies, which is a significant advantage. This fact allows integrating these technologies into platforms without much difficulty:

- No licenses are required for integration.
- It is possible to customize and tailor various components of applications very flexibly.
- The community creates special software development kits or libraries to simplify integration into third-party systems.
- There is a community that can promptly fix any errors or security vulnerabilities found.
- Both platforms support end-to-end encryption, which is crucial for companies and organizations, ensuring the security of the application.

2.2.1 Jitsi Meet

Jitsi Meet is a technology for conducting video conferences with a fairly wide range of functionality that can be suitable not only for private conversations but also for organizations, companies, educational institutions, etc [94].

As I mentioned in the analytical part, `Jitsi Meet` consists of several important components that, when combined, form the architecture of the technology [95]:

- Jitsi Meet
- Jitsi Videobridge (JVB)
- Jitsi Conference Focus (jicofo)
- Jitsi Gateway to SIP (jigasi)
- Jitsi Broadcasting Infrastructure (jibri)
- Additionally, there is Prosody, an XMPP server used for signaling.

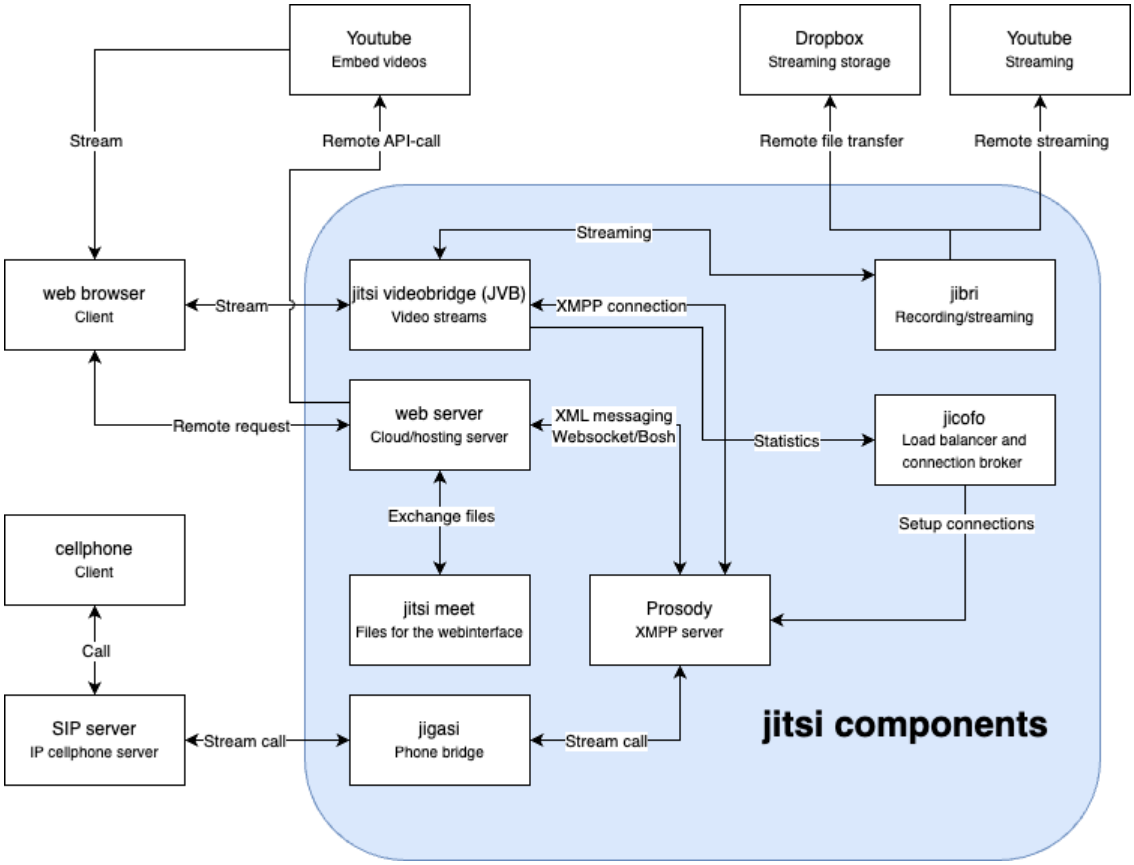

Figure 2.13: Jitsi Architecture Diagram

Apart from the Jitsi application, there are many other connections, the main one being the client or web browser. All control happens from here; it's through the client that users can manage actions within the entire application, such as accessing data, turning on or off the microphone, camera, etc [95].

The other components serve as additional services that support the technology, store conference and participant data, perform integrations with different technologies, applications, etc [95].

`Jitsi Meet` allows integration not only through commonly used tools but also based on frameworks. Possible Software Development Kits for integrating `Jitsi Meet` video conference technology include [96]:

- **IFrameAPI:** JavaScript JitsiMeetAPI library for integration.
- lib-jitsi-meet API (low level): JavaScript JitsiMeetAPI library for integration but at a lower level. It's used for full control of application processes with the ability to create a custom user interface.
- **Electron SDK:** Adding a toolkit for integrating Jitsi Meet into Electron-based applications.
- **React SDK:** Adding a configured component and functions for integrating Jitsi Meet into React applications.
- **Android SDK:** Jitsi Meet Android SDK allows adding all the functionality for integrating Jitsi Meet into an Android application.
- **iOS SDK:** Jitsi Meet iOS SDK adds all the functionality for integrating Jitsi Meet into an iOS application.
- **React Native SDK:** React Native SDK adds all the functionality for integrating Jitsi Meet into a React Native application, which can then be available on both Android and iOS platforms.
- **Flutter SDK:** Adding a toolkit for integrating Jitsi Meet into Flutter applications.

Another important aspect of Jitsi integration is the server where all conferences will occur. Jitsi Meet allows several options:

- Creating own Jitsi server where all video conferences will be running.
- Connecting to the official Jitsi server - `meet.jit.si` or 8x8 Jitsi as a Service - `8x8.vc`.
- Connecting to any third-party Jitsi server.

Integration of Jitsi Meet

Jitsi Meet is a fairly good and stable platform that is well suited for integration into various applications. Numerous software development kits are available that can significantly ease the integration process.

The integration of `Jitsi Meet` into an application consists of two stages:

1. Selecting a server for connection.
2. Integration into the application using software code.

Jitsi server connection

The choice of server for video conferencing platforms is crucial. It is the place where all key processes of the technology occur. Here, data is stored in databases and processed, events from the client side are handled, and the logic of the entire video conferencing platform functionality is implemented.

There are several ways to choose a server for connection:

- Hosting of own `Jitsi Meet server`.
- Connection to `Jitsi Meet application server` or `Jitsi as a Service server by 8x8`. This servers are official
- Connecting to existing `Jitsi Meet` services hosted by community.

Deploying own Jitsi Meet Server Creating your own server allows you to configure it with maximum flexibility due to the fact that the code is in public access.

An organization can customize as much as possible not only from the user interface but also from system connection or configuration components, such as Cross-origin resource sharing and firewall.

The process of setting up a server is detailed in the `Jitsi Meet` documentation, but the principle is quite simple [97]:

1. Launch a virtual machine or server on various platforms, for example, AWS - Amazon Web Services or Digital Ocean. The key is to ensure that the virtual machine's capabilities are sufficient to run the server.
2. Acquire a domain using `Route 53`, `GoDaddy`, or any other platform to maintain a hosted domain with configurable access. Add the public IP address of the virtual machine to the `DNS records` of the gotten domain.
3. Download `Jitsi Meet` onto the virtual machine, specifying the domain in the connection settings.
4. If all stages are successful, your `Jitsi Meet server` will be accessible through a browser using the acquired domain.

The big advantage of this solution is its flexibility in configuration. The server can be customized or branded in any way, depending on the organization. On the other hand, this is the price for server or domain support; often, it is not cheap.

Connection to official services It is noteworthy that the official Jitsi servers are not just a demonstration of the technology but also completely working solutions that can be used as servers to integrate the technology. At the moment, there are two official servers:

1. The official Jitsi Meet server, which has the domain `https://meet.jit.si/`, serves the primary purpose of demonstrating the technology and facilitating video calls through the Jitsi Meet web application interface.
2. The official Jitsi as a Service server with the domain `https://8x8.vc/` is operated by the company 8x8, which owns the Jitsi platform. The Jitsi as a Service service was created to monetize, allowing the integration of Jitsi Meet into other applications using 8x8 servers on a paid basis.

Initially, in the DTMeet application, Jitsi was integrated using the official Jitsi Meet server with the domain `https://meet.jit.si/`. The `DTMeet` platform was directly connected to the official server, the optimal and secure solution. This allowed seamless utilization of the technology without additional authorizations or configurations.

But starting from August 22, 2023, Jitsi decided to change the usage rules of the platform and required all users connecting to the official server to undergo authorization [98].

This change disrupted the concept of the DTMeet platform due to the mandatory authorization requirement. Therefore, a decision was made to switch to another server, using the Jitsi as a Service server by 8x8.

The free version of the server was used, allowing connection to conferences without authorization but with limited capabilities.

Several months later, the innovations from the official Jitsi server also reached Jitsi as a Service, which prompted me to seek another method of integrating Jitsi video conferencing technology into the communication platform DTMeet.

Connection Jitsi community instances Jitsi has a fairly large community that is actively developing and improving the entire technology. Enthusiasts quite often open their servers for free access and connection to other people.

After unsuccessful attempts to turn off forced authorization, it was decided to find a community-supported server that would not have this forced authorization. The official Jitsi documentation was actively expanded with a list of such servers from various regions [99][100].

After many attempts to connect to different servers, a server with the domain `meet.isf.es` was found, where no authorization was required. This server eventually became used for the DTMeet platform due to its compliance with all platform requirements.

Integration Jitsi Meet to client part of application

There are many software development kits for integrating Jitsi Meet into various applications. Still, in my case, the integration was carried out in a web application developed based on the React framework. Therefore, I had three methods of integration using:

- React Software development kit.
- IFrame API.
- lib-jitsi-meet API (low level).

During integration, adding functionality for calls and video conferences was important. Providing participants with the ability to communicate using the DTMeet platform, utilizing Jitsi technology for video conferencing. However, one of the main requirements during integration was interface customization or user interface customization.

Initially, I attempted integration using the `React Software Development Kit`. Still, unfortunately, it didn't allow the use of various Event Emitters or the ability to influence the Jitsi component from the parent component. However, the `React Software Development Kit` is excellent for integration without significant functionality or user interface intervention.

The **IFrame API** was an excellent solution because it supports JitsiMeet functions, commands, and events.

- **Functions** - API functions to control your Jitsi Meet conference.
- **Commands** - control of conference by calling executeCommand.
- **Events** - implements the EventEmitter API for emitting and listening for events.

These tools allow you to manage the conference practically from the client side, receive data about what is happening, and create events based on actions in the application.

Integration Jitsi Meet with IFrame API

To connect the `IFrame API` library, you first need to add Jitsi Meet API library scripts to the `index.html` file.

```
<script src='https://<your-domain>/external_api.js'></script>
```

Listing 2.8: Jitsi Meet API library script integration to index.html

<your-domain> - have to be changed to domain of Jitsi Meet server.

The necessary objects for this entire component to work are:

```
const [state, setState] = useState({
  isAudioMuted: false,
  isVideoMuted: false,
  isRaiseHand: false,
  isChatShown: false
  }
);

const formData = useLocation().state;
const navigate = useNavigate();
const jitsiMeetingName = formData.url.replace(`https://${
   JitsiConfigData.DOMAIN}/`, '');
const [meetUpStatus, setMeetUpStatus] = useState(false);
const [API, setAPI] = useState();
```

Listing 2.9: Jitsi component variables

- **state** - Object indicating the status of buttons during a video conference.
- **formData** - User data received from the form when starting or joining a conference.
- **navigate** - The variable is a representation of the navigation operation from react router.
- **jitsiMeetingName** - Name of the video conference.
- **meetUpStatus** - Status whether the video conference has started.

- **API** - Representation of the object responsible for setting up and initializing a video conference.

Integration of Jitsi Meet occurs in the form of a React component using the functionality of this framework.

Firstly, a function for initialization needs to be created. In my code, initialization looks like this.

```
const startMeeting = () => { // Jitsi Meet logic }
```

Listing 2.10: startMeeting function initialization

Inside `startMeeting` function, an API object is initialized, which will be the object responsible for everything that happens inside Jitsi Meet.

```
var api = new window.JitsiMeetExternalAPI(`${JitsiConfigData.DOMAIN}`,
    {
  roomName: jitsiMeetingName,
  parentNode: document.querySelector('#meet'),
  prejoinConfig: {enabled: false},
  configOverwrite: {
  toolbarButtons: ['closedcaptions', 'fullscreen',
    'fodeviceselection', 'profile', 'recording',
    'livestreaming', 'etherpad', 'settings',
    'videoquality', 'filmstrip', 'feedback', 'stats', 'shortcuts', '
    tileview']
  },
  interfaceConfigOverwrite: {SHOW_BRAND_WATERMARK: true,
    BRAND_WATERMARK_LINK: 'https://upload.wikimedia.org/wikipedia/
    commons/thumb/d/dd/Deutsche_Telekom_2022.svg/1200px-
    Deutsche_Telekom_2022.svg.png'},
  lang: 'en',
  width: '100%',
  height: 500,
  userInfo: {
  displayName: formData.username
  }
});
```

Listing 2.11: api Jitsi api object configuration

This way, the Jitsi Meet object is initialized into which various settings are written. The main configuration parameters i configured are `parentNode` and `configOverwrite`.

- **parentNode** defines the id of the element from the virtual DOM behind which the document with the Jitsi iframe will be attached.

- **configOverwrite** is responsible for overriding for default options of Jitsi.

More Jitsi iframe customization options can be found in the Jitsi documentation.

In the `startMeeting` function, it is crucial to set Event Listeners to the API object. The list of these functions and listeners can be found in the Jitsi documentation. After initializing the event listeners, the API state object is set by configuring the `api` Jitsi API object.

```

const handleMuteStatus = (audio) => {
  console.log("handleMuteStatus", audio);
  setState({...state, ['isAudioMuted']: audio.muted})
}

const handleVideoStatus = (video) => {
  console.log("handleVideoStatus", );
  setState({...state, ['isVideoMuted']: video.muted})
}

api.addEventListeners({
  readyToClose: handleClose,
  participantLeft: handleParticipantLeft,
  participantJoined: handleParticipantJoined,
  videoConferenceJoined: handleVideoConferenceJoined,
  videoConferenceLeft: handleVideoConferenceLeft,
  audioMuteStatusChanged: handleMuteStatus,
  videoMuteStatusChanged: handleVideoStatus
});

setAPI(api);
```

Listing 2.12: Event listeners initialization and setting API object

It is very important that when generating a component, the conference is initialized first. For this purpose, we can use `useEffect` to initialize the conference, inside of which the `startMeeting` function is launched.

```
useEffect(() => {
  if (window.JitsiMeetExternalAPI) {
    startMeeting();
    setMeetUpStatus(true);
  } else {
    alert('JitsiMeetExternalAPI not loaded');
  }
}, []);
```

Listing 2.13: useEffect in Jitsi Component

Another important function is the processing of events for pressing video conference buttons.

```
function executeCommand(command) {
  API.executeCommand(command);
  if (command === 'hangup') {
    if (API.getParticipantsInfo().length === 1) {
      axiosPrivate.put(`api/v1/conferences/close-conference/${
    jitsiMeetingName}`).then((response) => {
        console.log('Finish meeting response --> ', response.data);
      }).catch((err) => {
        console.warn('Somethig is wrong', err);})}
    navigate('/');
    window.location.reload();
  }
  if (command === 'toggleRaiseHand') {
    setState({...state, ['isRaiseHand']: !state.isRaiseHand})
  }
  if (command === 'toggleChat') {
    setState({...state, ['isChatShown']: !state.isChatShown})
  }
}
```

Listing 2.14: proccessing events functions

In this function, the important operation is the hangup operation, which initially checks if anyone remains in the conference. If user is last, the conference ends for all users in database. This is done to add the feature of conference management using rooms.

The last important function of this component is generating the invitation link to the conference through the DTMeet platform.

```
function generateJitsiInviteLink(jitsiMeetingName){
  let prepareLink = `${CLIENT_BASE_URL}/invite?conferenceName=${
    jitsiMeetingName}&technology=Jitsi`+SECRET_TOKEN;

  let summ = hex_sha1(prepareLink);

  return `${CLIENT_BASE_URL}/invite?conferenceName=${jitsiMeetingName
    }&technology=Jitsi&checksum=${summ}`;
}
```

Listing 2.15: generation of invitation link function

The main feature is the creation of an SHA-1 hash token based on the entire link. Then, this key is added as a checksum parameter. When you try to go to

an invitation link, the application will initially check whether this link is actually generated and not written manually.

The code written for the virtual DOM is the central part of executing and rendering the entire component. This is what the Jitsi Component itself returns.

```
<div className="App">
  <div style={{marginTop: '30px'}} id="meet"/>
    {meetUpStatus ?
      <div>
        <button onClick={() => executeCommand('toggleAudio')} title="
    Mute / Unmute">
          // microphone icon
        </button>

        <button className={styles.btn} onClick={() => executeCommand('
    toggleVideo')} title="Start / Stop camera">
          // video icon
        </button>
        <button className={styles.btn} onClick={() => executeCommand('
    toggleRaiseHand')} title="Rise your hand">
          // hand icon
        </button>
        <button onClick={() => executeCommand('toggleShareScreen')}
    title="Share your screen">
          // screen share icon
        </button>
        <button onClick={() => executeCommand('toggleChat')} title="
    Show chat">
          // show chat icon
        </button>
        <button onClick={() => executeCommand('hangup')} title="Leave"
    >
          // hangup icon
        </button>

        <div className={urlStyles.urlContainer}>
          <button ... > Copy join URL </button>
          <button ... > Copy invite link to meeting </button>
        </div>
      </div>
    :
    // meeting loading component
    }</div>
```

Listing 2.16: Jitsi component return value

The logic of the entire component is as follows. Upon component startup, a Jitsi video conference window is created based on the data received from the preceding component. After generating the Jitsi window, buttons appear in the element with `id=" meet"` which, upon clicking, triggers actions in the Jitsi window thanks to created functions and configured event listeners. Changing the status of a button, such as a microphone or camera, changes the icon on that button. In case of conference finishes, the conference ends in the database before exiting the conference.

2.2.2 BigBlueButton

BigBlueButton is an excellent platform for long-distance communication using the instrument of video conferences. It supports many different frames that can be suitable for video conferences of various workload levels and tasks. The platform is great not only for small private conversations between interlocutors but also for working in organizations, team calls, online lessons, and training [75].

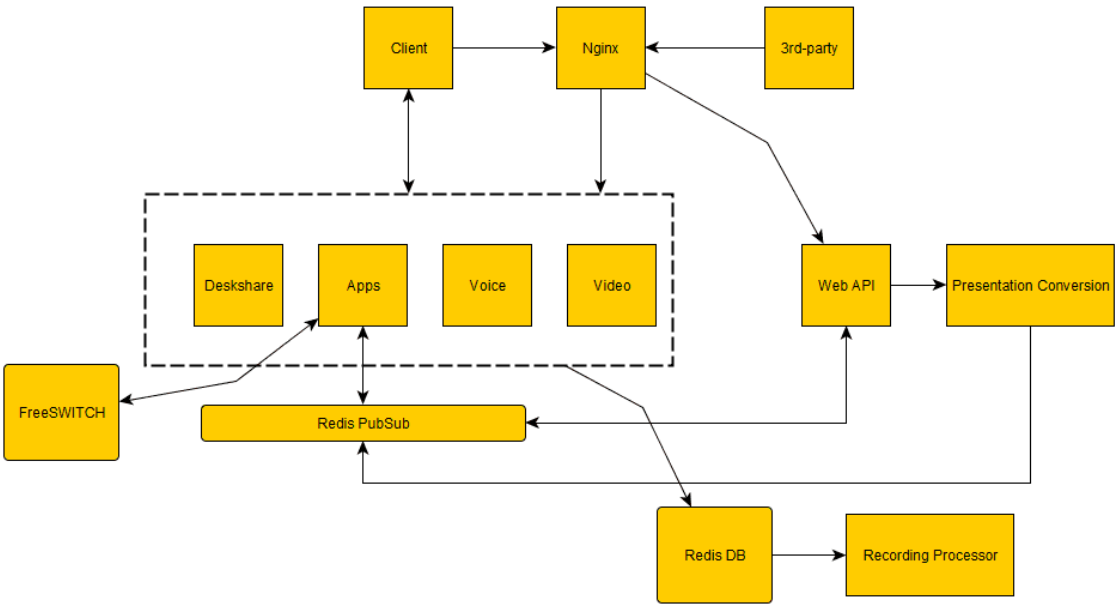

Figure 2.14: BigBlueButton architecture diagram

- `Deskshare, Apps, Video, Voice.` It provides communication between different applications running on the BigBlueButton server.
- `Redis PubSub.` It is the data repository where all video conferencing and screen recording events are stored. The `RecodingProcessor` provides processing of this data

- Redis DB. It is a fairly stable platform, which is why BigBlueButton is often integrated into the services of companies, educational institutions, startups, educational platforms, etc.
- `FreeSWITCH.` Software for handling audio.
- `Presentation conversion flow.` Presentation preprocessing.
- `Nginx.` This technology provides proxy configuration of the BigBlueButton server.

Integration of technology for video conferencing **BigBlueButton** consists of two stages:

- Launching the BigBlueButton server, setting up connections, and server configuration.
- Implementation of functions for sending requests to the BigBlueButton server. Displaying the video conference window to the user.

Integration of BigBlueButton

Integration of BigBlueButton is primarily implemented into platforms for which BigBlueButton developers have created specific plugins or extensions, such as `Moodle`, `WordPress`, `Nextcloud`, etc.

Furthermore, BigBlueButton does not have a free connection server. In the obligatory case, one needs to create a server for communication during integration.

Since the DTMeet platform is a React-based application, the only option was to use the API Mate server generated together with the server itself during installation.

BigBlueButton server connection

Creating your own BigBlueButton server is a mandatory requirement for integration. To accomplish this, I tested different versions of the BigBlueButton server, and as of the time of writing this thesis, the most stable version is `BigBlueButton 2.6`, in my opinion.

Today, the most popular platforms for server installation are `AWS - Amazon Web Services` and `Digital Ocean`. Both platforms have pros and cons; therefore,

I also decided to compare the installation and configuration of the BigBlueButton server on both.

The server's system requirements were important when setting up BigBlueButton.

- 4 CPU cores/8 GB of memory
- 50G of disk space
- IPV4 address only
- Ubuntu 20.04 (Focal Fossa)

You also need to get a domain address that the user can reach to install BigBlueButton on the server.

Setting up the BigBlueButton server on Amazon Web Services - AWS Amazon Web Services (AWS) is a powerful service that allows you to perform almost any cloud computing-related operation using the power of Amazon servers. The platform has many different applications that are part of one large infrastructure.

Using this `AWS`, you need to create an `EC2` instance on which a BigBlueButton server will be running. After choosing an `EC2` application, you first need to select the server region. This significantly affects the server response speed. The closer the server, the better it will be for the platform user.
The `region selection` can be found in the top right corner of the `AWS` interface.

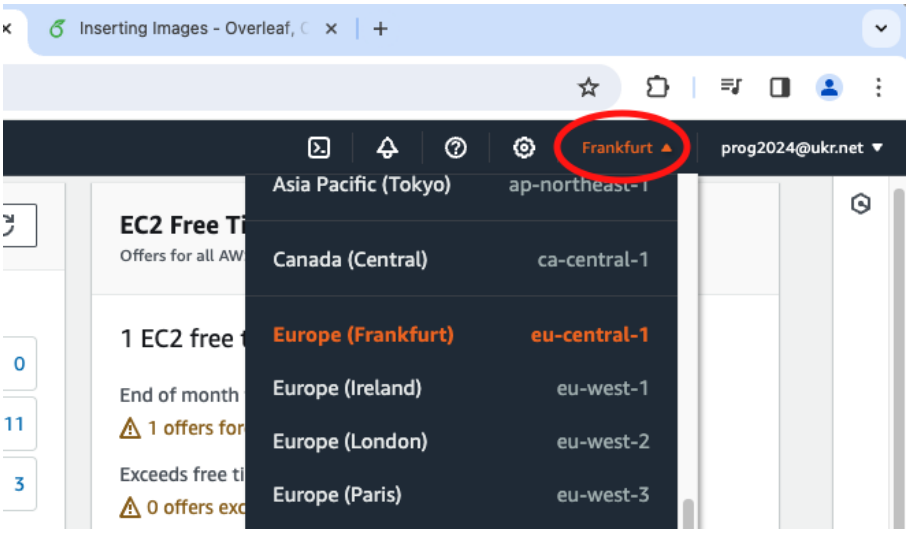

Figure 2.15: Region selection option in Amazon Web Services platform

In my case, the region selected was `Germany, Frankfurt` because this is the closest server to my location.

The next step is to create a Security group, which will later be used during the instance's configuration. The group is located in the `EC2` application `Network & Security` section. After selecting Security Groups, click on the `Create Security Group` button.

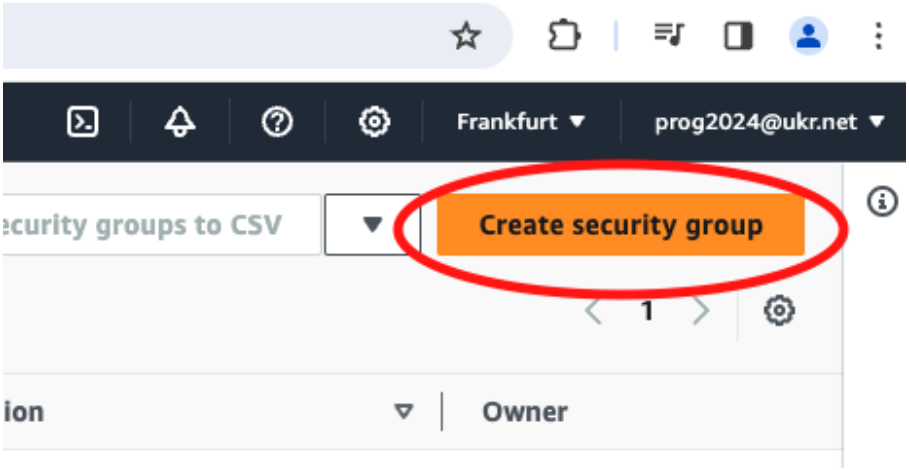

Figure 2.16: Create security group option

After assigning a name and description to the `Security group`, an important task is configuring the ports in the `Inbound Rules` section. This is one of the main requirements for running the BigBlueButton server.

This setting is important to allow various components of the BigBlueButton server to communicate with each other.

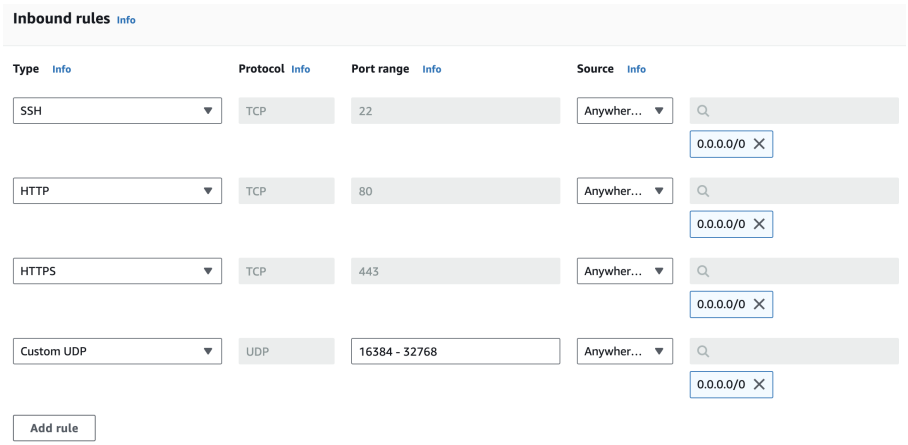

Figure 2.17: Security group inbound rules

- SSH = TCP = 22 = Anywhere-IPv4
- HTTP = TCP = 80 = Anywhere-IPv4
- HTTPS = TCP = 443 = Anywhere-IPv4
- Custom UDP = TCP = 16384 - 32768 = Anywhere-IPv4

After successfully creating a `Security Group`, you need to go to the `Instances` tab and select the `Instances` section in the `EC2` application. In the Instances section you need to click on the `Launch Instances` button.

After successfully creating a Security Group, you need to go to the `Instances` tab and select the `Instances` section in the `EC2` application. You need to click the `Launch Instances` button in the Instances section.

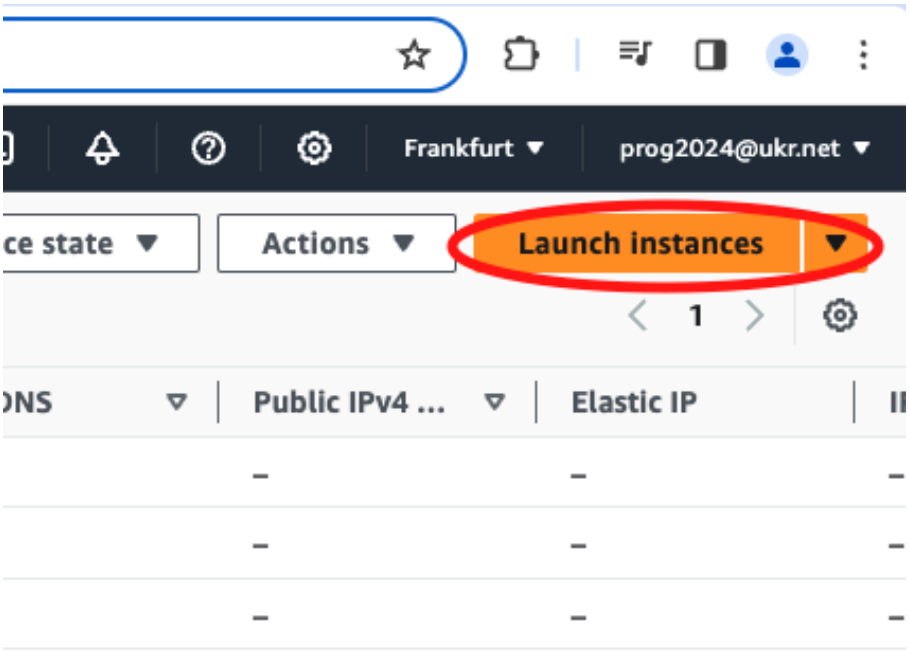

Figure 2.18: Launch Instances option

The `Launch an instance` is divided into several sections, so I will describe what you need to select from each of them:

- **Name and tags** - Enter the instance's name. You can also add specific tags to the server, facilitating searching for the instance, especially if many exist.
- **Application and OS Images (Amazon Machine Image)** - This is the section for selecting the server's operating system. To install BigBlueButton 2.6, Ubuntu version 20.04 is required, so you should choose the `Canonical, Ubuntu, 20.04 LTS, amd64 focal` image.
- **Instance type** - Section for selecting server specifications. BigBlueButton's minimum requirements are 4 CPU cores, 8 GB of memory, and 50 GB of disk space. Based on these requirements, you need to choose an in- stance type. I selected the `[c3.xlarge]` machine.
- **Key pair (login)** - Creating a key for connecting to the server allows you to secure the server from unauthorized access by individuals without permission. I used the RSA algorithm in `.pem` format to create the key.
- **Network Settings** - Network configuration of the instance. In the `Firewall` selection, you need to click on `Select existing security group` and then choose the previously created security group.

- **Configure storage** - Configuration of the storage of the instance. A mandatory requirement for installing the BigBlueButton server is a storage capacity of more than 50 GB. In the `Configure` storage section, you need to adjust the capacity to 50 GB.

These settings are sufficient for installing BigBlueButton on the created instance. After that, you need to click on the **Launch instance** button and wait until all checks are completed. To proceed to the next step, it is important that the `Instance state` status is `Running` and `Status check` has to be `passed`, so it's worth waiting.

It is very important to note that Amazon instances have 2 IP addresses, private and public.

1. **Private IPv4 address** - This is the permanent IP address of the instance specifically assigned to this server within the Amazon Web Services infrastructure. Specify the private IP address when adding any functionality to this server within Amazon Web Services.

2. **Public IPv4 address** - The IP address used to access the instance from outside the Amazon Web Services infrastructure. It changes after restarting the instance.

To receive instructions on connecting to the server, select your instance and click the `Connect` button. Several connection methods will then appear.

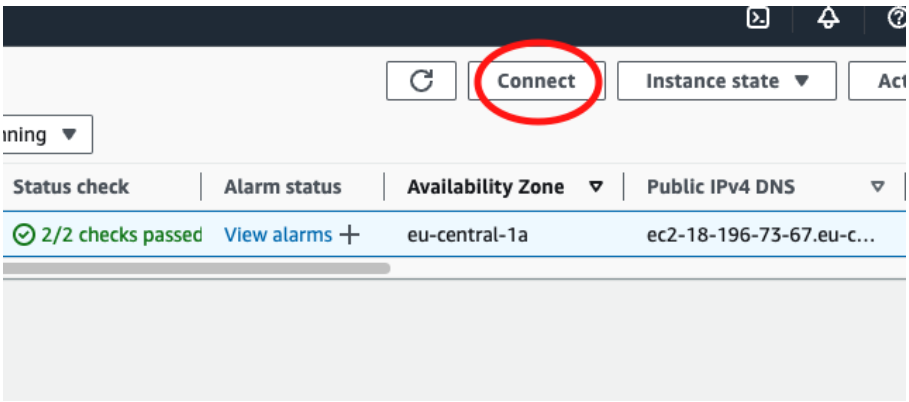

Figure 2.19: Connect BigBlueButton instance option

In my case, there were two available using EC2 Instance Connect and SSH client.

- **EC2 Instance Connect.** Server command line using a browser window.

- **SSH client.** Connecting to the instance via terminal using the secret key created during the instance configuration during creation.

Setting up the BigBlueButton server on DigitalOcean DigitalOcean is also a quite popular solution for setting up a BigBlueButton server. When creating your server, the creators recommend using the DigitalOcean platform. This platform also has a fairly wide range of features and is often used for creating servers, deploying projects, and building infrastructures from various systems that communicate with each other.

To create a server on the DigitalOcean platform, one needs to select `Create Droplet` in the menu.

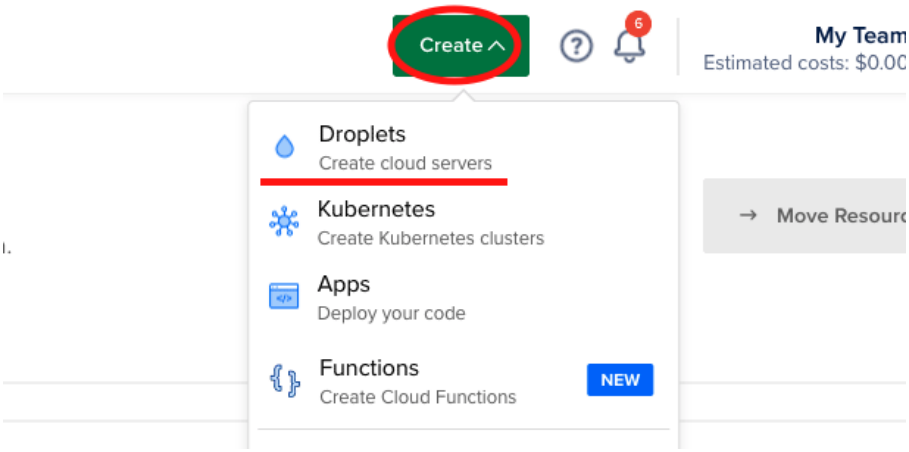

Figure 2.20: DigitalOcean create Droplet selection

The next step is to select the Data Center region. The server closest to the users will provide the best performance and minimum latency. In my case, this is `Frankfurt`.

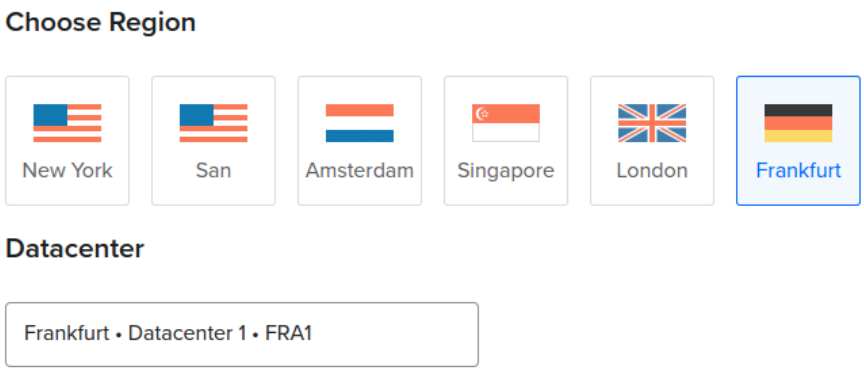

Figure 2.21: DigitalOcean droplet region selection

Next, in the `Choose an Image` section, you must select the system type. Based on the requirements of the BigBlueButton server, you must select the `Ubuntu operating system version 20.04 64-bit.`

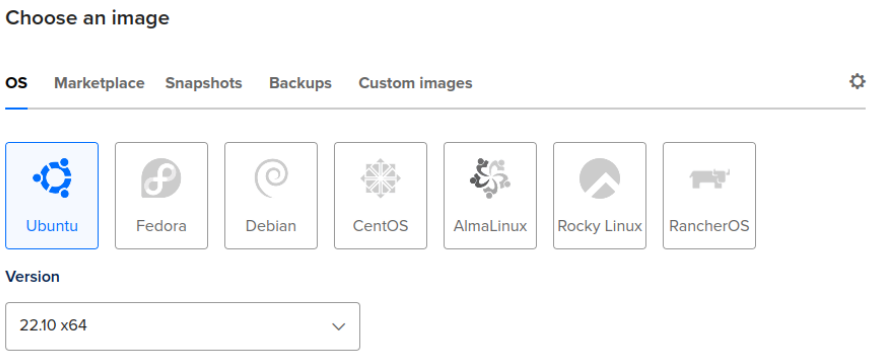

Figure 2.22: DigitalOcean droplet type selection

The `server configuration` selection section is also quite important. To run the BigBlueButton application on the server, you must have at least 4 CPU cores, 8 GB of memory, and 50 GB of disk space.

In this section, you must select a configuration meeting the minimum requirements for running BigBlueButton.

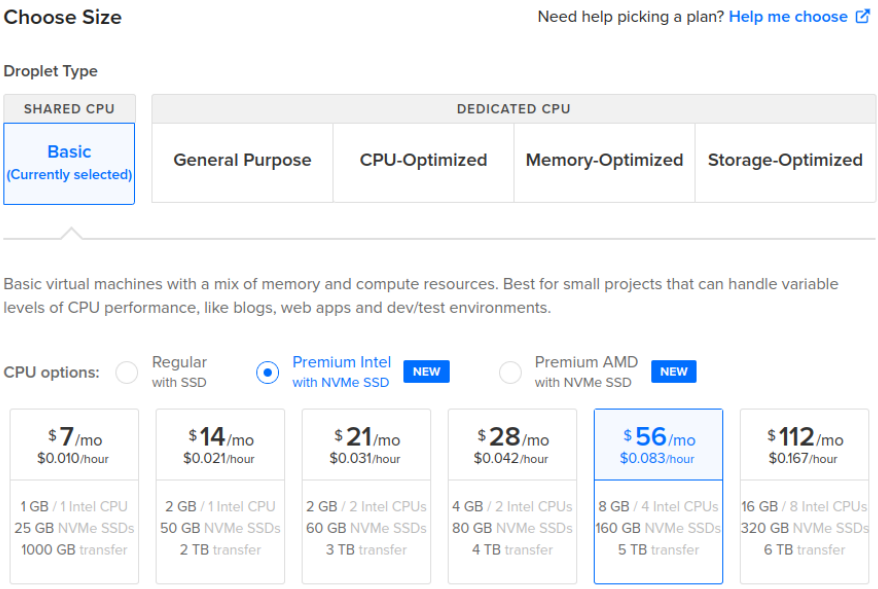

Figure 2.23: DigitalOcean Droplet server configuration selection

The last important part of creating a DigitalOcean droplet is selecting the `authentication method` during connection to the created Droplet.

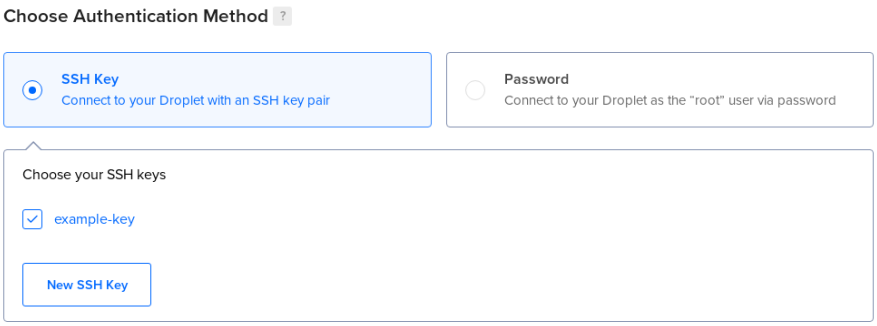

Figure 2.24: DigitalOcean Droplet server authentication method selection

All other settings for creating a DigitalOcean droplet are not mandatory when setting up a server.

After launching the created Droplet, you can connect to the server to perform further configurations within the system. There are two main methods for connecting: using the DigitalOcean web console and connecting via the chosen authentication method by terminal.

Comparison of Amazon Web Services (AWS) and DigitalOcean for BigBlueButton server `Amazon Web Services (AWS)` and `DigitalOcean` today are quite popular platforms that are often used not only by ordinary users or small organizations but also by large companies to launch and support their products. Both platforms have proven themselves to be quite reliable and flexible to use over time.

But during the launch of a BigBlueButton server, it's very important to consider the platform's peculiarities. Therefore, it's necessary to outline and analyze `Amazon Web Services (AWS)` and `DigitalOcean` for running BigBlueButton. This could be useful for companies or organizations which are planning to set up a BigBlueButton server.

Prices: `DigitalOcean` tends to be much cheaper to maintain than `AWS`. Services on `DigitalOcean` are more affordable, and in terms of quality, they are almost on par. On the other hand, `AWS` offers a much wider range of configurations and detailed settings. Additionally, `AWS` has many more services and tools.

Difficulty: Setting up an `AWS` server is more complex due to the creation of security groups. In `DigitalOcean`, everything is much simpler, and additional configurations or firewall settings are not required.

Additionally, `AWS` has two IP addresses, private and public. The public one is required to connect via the server's domain, and this address changes upon

instance restart. This requires additional `DNS rule settings` for the domain. The presence of two keys can incorrectly generate files during the installation of BigBlueButton on the server. However, having two keys allows for safer operation within the AWS infrastructure. In turn, a `DigitalOcean` droplet has a single static public key. Accordingly, there are no such problems.

Virtual machine configurations: Both platforms have a wide selection of server configurations. The only difference is that in `AWS`, you need to register the exact number of gigabytes of storage. `DigitalOcean` already has a certain amount of storage that can be added.

Server startup time: Launching an image on `DigitalOcean` is much faster than on the `Amazon web services (AWS)` platform.

Both platforms are great for running the BigBlueButton server. Also, both platforms have quite wide and sufficient functionality for running BigBlueButton.

Suppose BigBlueButton is planned to be part of a larger infrastructure based on AWS. In that case, using the `AWS EC2 instance` application is better, as it allows flexible configuration and management of the entire infrastructure in one place. However, on the other hand, installation on an `AWS instance` is more complex, and problems may arise when configuring configuration files for BigBlueButton itself.

The developers recommend `DigitalOcean` for BigBlueButton server. `DigitalOcean Droplet` is also much cheaper and easier to support and install.

It all depends on the system or infrastructure into which the BigBlueButton server will be integrated.

Process of pre-installation BigBlueButton The next important stage in integrating the BigBlueButton video conferencing technology is pre-installation. This is the process of preparing and configuring the server for the installation.

After successfully connecting to the server, it is necessary to initially login as the root user to ensure that all commands are executed correctly.

Firstly, the server's locale needs to be checked.

```
$ cat /etc/default/locale
// Expected Result
// LANG="en_US.UTF-8"
```

Listing 2.17: Check locale of server command

If the expected result is not the same as in the command-line interface, then you need to write commands.

```
$ sudo apt-get install -y language-pack-en
$ sudo update-locale LANG=en_US.UTF-8
// put again
$ cat /etc/default/locale
// Expected result
// LANG="en_US.UTF-8"
```

Listing 2.18: Uploading language packs command

This is done to upload language packs to the server.

Then, the system language environment needs to be checked.

```
$ sudo systemctl show-environment
// Expected result
// LANG=en_US.UTF-8
// PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Listing 2.19: Show system language environment command

If there is no `LANG=en_US.UTF-8` as inexpected result in the output, you should write this:

```
$ sudo systemctl set-environment LANG=en_US.UTF-8
// and put again to check does it works
$ sudo systemctl show-environment
// Expected result
// LANG=en_US.UTF-8
// PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Listing 2.20: Setting up system language environment command

The next step is to check the server's available memory. The minimum memory required for BigBlueButton is `8 GB`. To check the available memory on the server, you need to enter the following command in the command-line interface:

```
$ free -h
// Expected result
//               total     *
// Mem:            8G      *
// Swap:           *       *
```

Listing 2.21: Command to check avaliable memory of server

If the server has less than 8 GB of memory, you need to either add it if the launch takes place locally on your device or select another characteristic of the virtual machine on the platform where the server is running.

Checking the operating system is a very important step in the pre-installation process of BigBlueButton 2.6 on the server. To do this, you need to write the command:

```
$  cat /etc/lsb-release
// Expected result
// DISTRIB_ID=Ubuntu
// DISTRIB_RELEASE=20.04
// DISTRIB_CODENAME=focal
// DISTRIB_DESCRIPTION="Ubuntu 20.04.XX LTS"
```

Listing 2.22: Operating system check command

Checking the `Ubuntu 20.04` version:

```
$ uname -m
// Expected result
// x86_64
```

Listing 2.23: Operating system bit-version check command

Checking whether the server supports `IPv6`:

```
$ ip addr | grep inet6
// Expected result
// inet6 ::1/128 scope host
// ...
```

Listing 2.24: Server support of IPv6 command

If `inet6 ::1/128 scope host` is not in the list of systems, then after installing BigBlueButton, you need to change the `FreeSWITCH` configuration to disable support for `IPv6` [101].

Checking the Linux kernel version:

```
$ uname -r
// Expected result
// 5.4.x-xx-generic
```

Listing 2.25: Linux kernel version check command

Checking the number of server cores. It is very important to have at least `8 CPU` cores.

```
$ grep -c ^processor /proc/cpuinfo
// Expected result
// 8
```

Listing 2.26: Server cores count check command

The last important check is whether ports `80` and `443` are open. To do this, write the command in the command-line interface:

```
$ sudo ufw status
// Expected result
// ...
// 80        ALLOW   Anywhere
// 443       ALLOW   Anywhere
// ...
// 80 (v6)   ALLOW   Anywhere
// 443 (v6)  ALLOW   Anywhere
// ...
```

Listing 2.27: Server firewall ports command

If Ports `80` and `43` are not open, you need to write commands to open them:

```
$ sudo ufw allow 80
$ sudo ufw allow 443
```

Listing 2.28: Opening of ports for firewall

Acquisition of hosted domain Purchasing a hosted domain is a very important part of setting up a BigBlueButton server. Using the domain, you can access the BigBlueButton web application and thus conduct video conferences. You can also send requests to BigBlueButton from another application or browser.

To purchase a domain, I used the `GoDaddy` platform, where I first selected a domain and then configured it using `DNS records` to determine access to the server through the domain name.

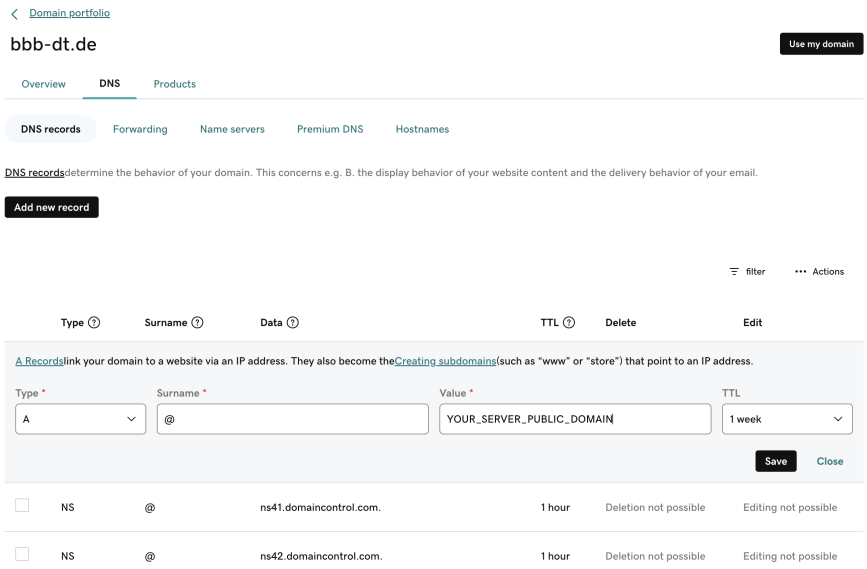

Figure 2.25: GoDaddy domain DNS settings

To associate a domain name with a server, you need to configure `DNS records` settings by setting a record with type `A` where the `Value` will be your BigBlueButton server's public `IPv4 address`.

Process of installation BigBlueButton Installation of BigBlueButton server version `2.6` is done by running the .sh script. Everything works automatically, so after running the command, you need to wait until everything starts. If you install `BigBlueButton 2.6`, installation methods for other versions can be viewed in the documentation. You need to enter the command:

```
$ wget -qO- https://ubuntu.bigbluebutton.org/bbb-install-2.6.sh | bash
    -s -- -v focal-260 -s bbb.example.com -e notice@example.com -g -w
```

Listing 2.29: BigBlueButton 2.6 install command

- **bbb.example.com** - Domain, hostname from which the server can be accessed.
- **notice@example.com** - Support mail. Here you need to specify the server administrator's email for technical support.
- **-w** - Automatic firewall configuration.
- **-g** - Downloading the `Greenlight` interface for the `BigBlueButton 2.6` web application. `Greenlight` adds beautiful styling to the entire user interface with branding options.

Important commands for the BigBlueButton server:

- **bbb-conf –check** - Check BigBlueButton server status.
- **bbb-conf –restart** - Check BigBlueButton server restart BigBlueButton server.
- **bbb-conf –status** - Check all BigBlueButton processes status.
- **bbb-conf –secret** - Displaying of important information for integration server. There is a secret key, a `URL link` to the BigBlueButton server, and a link to the `API-Mate` application for this server by default.

Setting up Cross-origin resource sharing (CORS) configuration BigBlueButton server, which hosts video conferences, a critical aspect is the communication between the client-side DTMeet application and the BigBlueButton server. Their communication is established with the DTMeet client, which sends API requests to the BigBlueButton server.

For this type of communication, BigBlueButton must be allowed to handle such requests from the DTMeet application. This issue can be resolved by configuring the `nginx` settings within the BigBlueButton server. Since `nginx` on the BigBlueButton server is a proxy server, it also includes access checks to the server. The solution to this problem involves configuring
`Cross-Origin Resource Sharing (CORS)` within the `nginx` component.

To configure `cross-origin resource sharing (CORS)` on the BigBlueButton server, you first need to connect to the server as a root user, for example, using a terminal.

The second step involves modifying the `nginx` configuration file. This can be done using the command:

```
$ nano /usr/share/bigbluebutton/nginx/web.nginx
```

Listing 2.30: BigBlueButton 2.6 modifying the nginx configuration file command

After that, the file `web.nginx` will open in the command-line interface, where you need to specify the CORS settings:

```
add_header 'Access-Control-Allow-Origin' '<DTMeet domain>';
add_header 'Access-Control-Allow-Credentials' 'true';
add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, HEAD';
```

Listing 2.31: nginx access-control settigns

Ultimately, the `nginx` configuration file should look something like this.

```
* Handle request to bbb-web running within a SpringBoot Tomcat
    embedded servlet container. This is for BBB-API and Presentation

location /bigbluebutton{
  proxy_http_version 1.1;

  location /bigbluebutton{
    proxy_pass  http://127.0.0.1:8090;
    proxy_redirect  default;
    proxy_set_header  X-Forwarded-For $proxy_add_x_forwarded_for;

    # Workaround IE refusal to set cookies in iframe
    add header P3P  'CP="No P3P policy available"' ;
    add header  'Access-Control-Allow-Origin' 'https://<dtmeet-client-
    domain>';
    add header  'Access-Control-Allow-Credentials'  'true';
    add_header 'Access-Control-Allow-Methods'  'GET, POST, OPTIONS,
    HEAD';

  }
...
}
```

Listing 2.32: nginx cross-origin resource sharing settigns

After changing the configuration, you need to restart the server using the command:

```
$ bbb-conf --restart
```

Listing 2.33: BigBlueButton restart server command

Integration BigBlueButton to client part of application

After installing and configuring the server, you can proceed to the final stage, namely integrating BigBlueButton into the DTMeet platform. Integration of BigBlueButton into the platform works as follows:

1. To begin with, the user uses the DTMeet interface to set up a conference or join using the invite link by putting a video conference into the platform.

2. First, the user configures the conference using the DTMeet interface or joins using an invite link, entering these details into input fields.

3. These data are sent to the server-side application, where a check is performed to determine whether the user has permission and can create a BigBlueButton conference with such settings. The response is then sent back to the client.

4. If the response is successful, a request is created on the client side and sent to the BigBlueButton server for connection.

5. Upon successful connection to BigBlueButton, a link is generated to connect to the BigBlueButton server conference, and an iframe object with the prepared link for connection is added to the application's `virtual DOM`. Thus, the conference window is integrated into the DTMeet platform.

Since DTMeet is a web application developed using the `React` library, a component responsible for the BigBlueButton conference was created.

Also BigBlueButton does not have a software development kit or any official library for integration, the only solution is to send requests to the server using regular `axios` requests with Java Script functionality.

I also found the library `bigbluebutton-js` [102] in the npm repository and used it solely for sending requests to the BigBlueButton server. This function's main feature is the high-quality parsing of data from the response. So, I created all other functions for generating links or requests myself, using the `SHA-1` encryption algorithm used in BigBlueButton.

The next part of this section will contain a detailed description of all the functionality and important functions of the BigBlueButton component.

Initialization of important variables for the proper functioning of the component:

```
1
2 // representation of location data object, saves data from previus
     component
3 const location = useLocation();
4 // state variable for generated join url of BigBlueButton
5 const [joinUrl, setJoinUrl] = useState('');
6 // representation of navigation object which is used for routing in
     component
7 const navigate = useNavigate();
8
9 // state variable for generated attendee invite url to BigBlueButton
10 const [attendeeURL, setAttendeeURL] = useState('');
11 // state variable for generated moderator invite url to BigBlueButton
```

```
const [moderatorURL, setModeratorURL] = useState('');

// object of variables from previous location
const {url, username, attendeePW, moderatorPW} = location.state;

// state variable which saves type of BigBlueButton window mode,
   create conference or join to the conference
const [operation, setOperation] = useState('');
```

Listing 2.34: BigBlueButton component initializing libraries. BBB.jsx

Functions for retrieving data from the link that was sent from the preceding component.

```

const getMeetingRoomName = () =>{
  var ur = new URL(url);
  var params = new URLSearchParams(ur.search);
  return params.get("meetingID");
}
const getModeratorPassword = () =>{
  var ur = new URL(url);
  var params = new URLSearchParams(ur.search);
  return params.get("moderatorPW");
}
const getAttendeePassword = () =>{
  var ur = new URL(url);
  var params = new URLSearchParams(ur.search);
  return params.get("attendeePW");
}
```

Listing 2.35: Url variables retrieve functions

Function for generating DTMeet invite link to the BigBlueButton conference. This function allows you to generate a link that can be sent to a person who wants to join the conference.

```

function generateBBBInviteLink(bbbMeetingName,password){

  let prepareLink = `${CLIENT_BASE_URL}/invite?conferenceName=${
    bbbMeetingName}&technology=${Technologies.BBB}&password=${password
    }`+SECRET_TOKEN;
  let summ = hex_sha1(prepareLink)

  return `${CLIENT_BASE_URL}/invite?conferenceName=${bbbMeetingName}&
    technology=${Technologies.BBB}&password=${password}&checksum=${summ
```

```
    }`;
}
```

Listing 2.36: Generate invite link BigBlueButton function

The `useEffect` hook in React allows running code inside its body before all other parts of the code, which is why the main logic of this component is written there.

```

useEffect(() => {

  const getData = async () => {

    if (url.includes('meetingID=')) {
      const op = getMeetingOperation(url);
      setOperation(getMeetingOperation(url))

        if (op === 'create') {
          const response = await http(url);

          if (response.returncode === 'SUCCESS') {
            setAttendeeURL(await createJoinUrl(response, '
    AttendeeUSERNAME', attendeePW))
            setModeratorURL(await createJoinUrl(response, '
    ModeratorUSERNAME', moderatorPW))
            setJoinUrl(await createJoinUrl(response, username,
    moderatorPW));

          } else alert('CANNOT CREATE MEETING');

        } else if (op === 'join') {
          const response = await http(generateIsMeetingExistsURL(
    getMeetingIdFromUrl(url)));
            if (response.running === true) {
              if (username.length === 0) setJoinUrl(url);
                else setJoinUrl(setNewUsernameToUrl(url, username));
              } else alert('Meeting not started');
            } else alert('THIS URL OPERATION NOT SUPPORTS');
          } else alert('WRONG URL');
        };
      getData();

    },
);
```

Listing 2.37: BigBlueButton useEffect function

1. To begin with, from the link sent from the preceding component, an operation is initiated either to create a conference or to join one.

2. If the operation is to create a conference, a request is sent to the BigBlueButton server for creation. After successful conference creation on the server, functionality is triggered to generate links for joining the conference, which will be used as part of the user interface. If the operation is to join a conference, this step is skipped.

3. After generating the links, a request is sent to the BigBlueButton server for connection. Subsequently, from the server response, a link is obtained that connects the `iframe` element from the component with the image of the BigBlueButton server conference.

Function to end the video conference by sending a request to the BigBlueButton server. The function is handled when pressing the `End Meeting` button.

```
1
2  const onHandleFinishBBB= async () => {
3
4    const response = await http(endMeeting(getMeetingRoomName(),
       getAttendeePassword()));
5
6    if (response.returncode === 'SUCCESS') {
7      axiosPrivate.put(`api/v1/conferences/close-conference/${
       getMeetingRoomName()}`).then((response) => {
8
9        navigate('/');
10       window.location.reload();
11     }).catch((err) => {
12         console.warn('Somethig is wrong', err);
13     });
14   }
15 }
```

Listing 2.38: finish BigBlueButton video conference function

1. When this function is triggered, the first step is to send a request to the BigBlueButton server.

2. In the event of a successful conference ending on the server, DTMeet sends a request to end the conference within the DTMeet platform. After that, the user is redirected to the main page of the web application.

The last important part is defining the elements that the BigBlueButton component returns.

```
1
2  <>
3    {
4      joinUrl.length === 0 ?
5        <div>BBB</div> :
6        <>
7          <iframe id='bbb' style={{marginTop: '30px'}} title='frame' src
     ={joinUrl} width="100%" height="500" allow="geolocation *;
     microphone *; camera *; display-capture *;" webkitallowfullscreen="
     false" sandbox="allow-same-origin allow-scripts allow-modals allow-
     forms "
8          ></iframe>
9
10         <SimpleButton hadleButtonFunction={onHandleFinishBBB} btnText
     ={"Finish BBB Meeting"} />
11
12       </>
13       }
14       {operation === 'create' &&
15
16         <>
17           <div className={styles.urlContainer}>
18             <button ... >Copy Attendee join URL </button>
19
20             <button ... > Copy Moderator join URL </button>
21           </div>
22
23           <div className={styles.urlContainer}>
24             <button ... >Copy Invite Attendee URL </button>
25             <button ... > Copy Invite Moderator URL </button>
26           </div>
27         </>
28       }
29 </>
```

Listing 2.39: BigBlueButton component return values

The functionality of the component is divided into two parts. The first part is an `iframe` that displays the image from the BigBlueButton server, while the second part is responsible for displaying buttons to invite users. The logic is designed so that only the conference creator can send invitation links. For privacy and security purposes, if a user creates a conference, they will see buttons.

2.3 Deployment

Software deployment is the process of setting up and preparing a system for general users. Deployment tools allow developers to release their applications on a publicly accessible platform for the purpose of testing, presentation, or use of the finished product.

In the process of developing the DTMeet system, it was necessary to deploy it for a full demonstration of the product's operation.

A full-fledged deployment that works in various companies or organizations often uses internal systems or those that have already proven themselves for such purposes, such as `Amazon web services (AWS)` and `DigitalOcean`, which run the application on cloud servers.

This section describes the deployment process and methods for launching the DTMeet web application platform in detail.

For the deployment of the DTMeet platform, two technologies were used:

- **Docker** - Platform designed to help developers build, share, and run container applications. Using Docker, the application will be launched locally on containers.
- **Ngrok** - Reverse proxy that creates a secure tunnel from a public endpoint to a locally running web service. It allows you to open local ports and assign them domains for access outside the local network.

2.3.1 DTMeet web application deploy

Since my deployment is more of a demonstration version, I haven't deployed all parts of the application architecture to any cloud server services such as `DigitalOcean` or `Amazon Web Services (AWS)`. However, the scripts written can be used on real platforms or integrated into the architecture of a cloud system.

The principle is to run the application locally using `Docker containers`. Then, using `Ngrok`, open the used local ports for general access via the Internet.

The principle is to first open the used local ports using `ngrok`.
`Docker containers` are launched locally with updated from git repositories.

My solution method is free and does not require the use of various machines or additional servers.

Setting up Ngrok.

To download `Ngrok`, you must first register on the official website, go through authorization, and receive your authorization token.

After that, download `ngrok` to your machine and connect your `ngrok account` to your computer using the `authorization token`.

```
ngrok config add-authtoken <TOKEN>
```

Listing 2.40: ngrok account connection

Since the client side of the application and the server are launched as two separate servers on different ports, the next step requires setting up multiple port forwarding using `ngrok`.

To configure multiple opening ports, you need to add settings to the configuration ngrok file.

- **Linux** /.config/ngrok/ngrok.yml
- **MacOS** /Library/Application Support/ngrok/ngrok.yml
- **Windows** C:\Users \'USER' \.ngrok2 \ngrok.yml

This file needs to be supplemented with this data:

```
authtoken: AUTHENTICATION_TOKEN
tunnels:
  first:
    addr: BACK_END_PORT
    proto: http
  second:
    addr: FRONT_END_PORT
    proto: http
```

Listing 2.41: ngrok multiple ports openning setting up

To run `ngrok` you need to write the command in the terminal:

```
ngrok start --all
```

Listing 2.42: ngrok open all prepared ports

After that, a `ngrok` window will appear with assigned domains to configured ports in the configuration file. These domains are very important for setting up subsequent components.

Configuring application components.

The next step is to configure the components before launching the application, for which you need to download the DTMeet project initially. You can download the DTMeet project from the `GitHub platform` using the link provided in the thesis attachment or download the project directly from the thesis attachment.

Since the script during container startup works to fetch the latest version of the application from `Git`, it is necessary to upload the client and server parts of the application to two repositories. It is important to have access to these repositories.

Docker-compose file configuring.

After opening the project, you will first see the `docker-compose.yaml` file, which needs to include data from the `Git` repositories of the server and client parts of the application, as well as configure ports.

1. **row 11:** Configuration of MySQL database server port. This port has to be free for setting up MySQL database container.
2. **row 21:** Configuration of port of server-side part of application. It has to be similar with port, which is configured in `ngrok.yaml` as back-end port.
3. **row 30:** Link to server-side part of application repository.It is very important to have an access to this project.
4. **row 39:** Configuration of port of client-side part of application. It has to be similar with port, which is configured in `ngrok.yaml` as front-end port.
5. **row 41:** Link to server-side part of application repository.It is very important to have an access to this project.

Client-side part configuring.

At this stage, it is necessary to upload updates on the client's domain settings to Git.

Inside the client-side application's repository, you need to locate the file containing technical data about the client, which is situated at `src/data/TechData.js`. Here, you need to add some changes.

1. **BASE_URL**: put the backend domain obtained from ngrok.
2. **CLIENT_BASE_URL** : put the front-end domain obtained from ngrok .

The next incredibly important part of the configuration is setting up the client side of the application to send requests to the servers on which the Jitsi and BigBlueButton conferences take place.

To configure the connection between DTMeet and Jitsi Meet, you need to change the value of the `DOMAIN` attribute of the `JitsiConfigData` object in the `JitsiConfig.js` file.

```
export const JitsiConfigData={
  DOMAIN: 'JITSI_DOMIAN'
}
```

Listing 2.43: Jitsi configuration file

To configure the connection between DTMeet and the BigBlueButton server, you need to change the values of the `serverConfig` object in the `BBBconfig.js` file:

```
const serverConfig = {
  url: 'DOMAIN_OF_BIGBLUEBUTTON_SERVER',
  secret: 'BIGBLUEBUTTON_SECRET_KEY'
}
export default serverConfig;
```

Listing 2.44: BigBlueButton configuration file

- **url** - domain for communication with the BigBlueButton server.
- `secret` - security identification key that is generated by BigBlueButton servers during installation.

Add changes to the Git repository of the client-side part of the application.

Server-side part configuring.

On the server side of the application, you need to add the client-side ngrok domain to the list of allowed `cross-origin resource sharing (CORS)` domains.

To do this, you must go to the application server repository and find the configuration file along the path `src/main/java/net/tuke/dt/videoconferenceapi/config/CrossOriginConfig.java`.

- **row:19** -You need to add the domain generated in the `ngrok` window of the client part of the application to the List of Domains.

Add changes to the Git repository of the server-side part of the application.

BigBlueButton server configuring.

On the BigBlueButton server, you also need to change the list of allowed `cross-origin resource sharing (CORS)` domains in the `Nginx` configuration.

First, you need to connect to the BigBlueButton server as the root user of the command-line interface.

Next, you need to write a command in the command-line interface to change the `nginx` configuration file.

```
$ nano /usr/share/bigbluebutton/nginx/web.nginx
```

Listing 2.45: modify nginx configuration file

You need to add changes inside the `nginx` file and paste them into the corresponding header, which is the domain of the client part generated in `ngrok`. I described this in more detail in the section on BigBlueButton server configuration.

The last step is to save the changes of the `ngrok` configuration file and restart the BigBlueButton server using the command:

```
$ bbb-conf --restart
```

Listing 2.46: BigBlueButton restart server command

Running docker containers.

The last stage of deployment is the launch of docker containers. First, you need to run `Docker` on the computer on which the project will run.

The next step is to run `docker-compose.yaml` to build and launch all containers, pulling updates from the `Git` repositories. To do this, you need to use the command:

```
$ docker compose up
```

Listing 2.47: docker-compose.yaml run

After this, you need to wait until all containers are launched. Once you start, you can comfortably communicate using the DTMeet platform.

To connect through a browser, you need to specify the domain of the client-side part of the DTMeet application generated by the `ngrok` technology in the `URL` field.

To disable all running processes, you need to write the command:

```
$ docker compose down -v
```

Listing 2.48: docker break all processes

3 Evaluation

3.1 Comparison of integration Jitsi and BigBlueButton

BigBlueButton and Jitsi Meet technologies are open-sorce platforms with quite a lot of potential.

In my opinion, these systems are the best candidates for integrating a video conferencing tool actively supported by the community. However, comparing methods and processes for integrating both platforms in this issue is important.

Such a comparison will allow a better understanding of the pros and cons of both platforms as a part of another system.

3.1.1 Server

The server of such technology is an incredibly important part. This logical part processes and manages data between the components of the entire application architecture. However, the availability of the server installation also plays an important role.

Jitsi Meet offers free usage options for its server, which can be fully utilized to integrate projects of various sizes and levels of load. In addition, the Jitsi Meet server can be installed locally or using cloud services. Installing and configuring a full-fledged server allows for flexible customization and configuration of all components of the integrated Jitsi application.

On the other hand, BigBlueButton does not have the option to connect to public, free servers. You need to set up your own BigBlueButton server. This indicates serious platform security and architecture, which can be configured flexibly. To connect, you need to use a secret key and configure the nginx configuration. BigBlueButton servers are quite resource-intensive, which significantly affects the cost of using the platform.

3.1.2 Documentation

The documentation of both technologies is quite high level.

But in Jitsi, there are fewer technical aspects and settings about how the whole technology works, which cannot be said about BigBlueButton, where every function, every problem, and every way to solve it are explained in detail.

Jitsi has written quite a lot about the functional features and ways to integrate and use the platform as a third-party application; in BigBlueButton, this information is much less.

3.1.3 Development kits

During integration, it is important to know and understand the ways of integrating technologies. This helps better tailor the technological stack and architecture of the entire application.

Jitsi's tools and software development kits (SDKs) are much more extensive. In addition to various optimal solutions for JavaScript, there are many kits for mobile applications, cross-platform applications, and desktops. This significantly increases the number of platforms where Jitsi can be integrated.

It's a bit more complicated with BigBlueButton. The technology is primarily integrated into platforms with creators' ready-made plugins. Among the official libraries, there are solutions only for PHP and Ruby. For all other cases, API requests are used, which are sent directly to the server with commands to create or join a conference.

3.1.4 Community

Both technologies have a fairly active and large community that helps develop and maintain the platform technically and often assists in configuring any functionality.

3.1.5 Functionality

The functionality of the platform significantly influences the choice of technology for integration.

Jitsi has simpler and less burdened functionality, allowing video conference sessions to be conducted easily.

BigBlueButton has more functionality for small teams and large conferences and presentations. Additionally, BigBlueButton allows installing the Greenlight interface on its server, which allows users to use BigBlueButton not only as an integrated window of technology on another platform but also as a full-fledged web application with registration, authorization, and various conference management increments.

3.2 User Manual

The DTMeet application is a communication platform for organizing and conducting video conferences using the integrated Jitsi Meet and BigBlueButton systems. DTMeet was designed in such a way that platform users do not waste their time on any additional configurations that may complicate the process of using the platform.

DTMeet can be used even without authorization. In this case, the user cannot manage room conferences.

Video conferencing technologies Jitsi Meet and BigBlueButton are fairly reliable solutions for the open-source platform market that allow such functionality.

The DTMeet platform has quite wide functionality and a user-friendly interface that allows you to use the application without any barriers. This user manual was created to explain all aspects of using the platform.

3.2.1 Authorization

When entering the platform, the user is directed to the page for authentication. The user can authenticate using his account credentials using the login page. Additionally, on this view, there is a `Sign_up` button that redirects the user to the registration page.

Figure 3.1: Login page

The registration page consists of a form where the user enters information to create an account. After entering this information, the user will be redirected back to the login page.

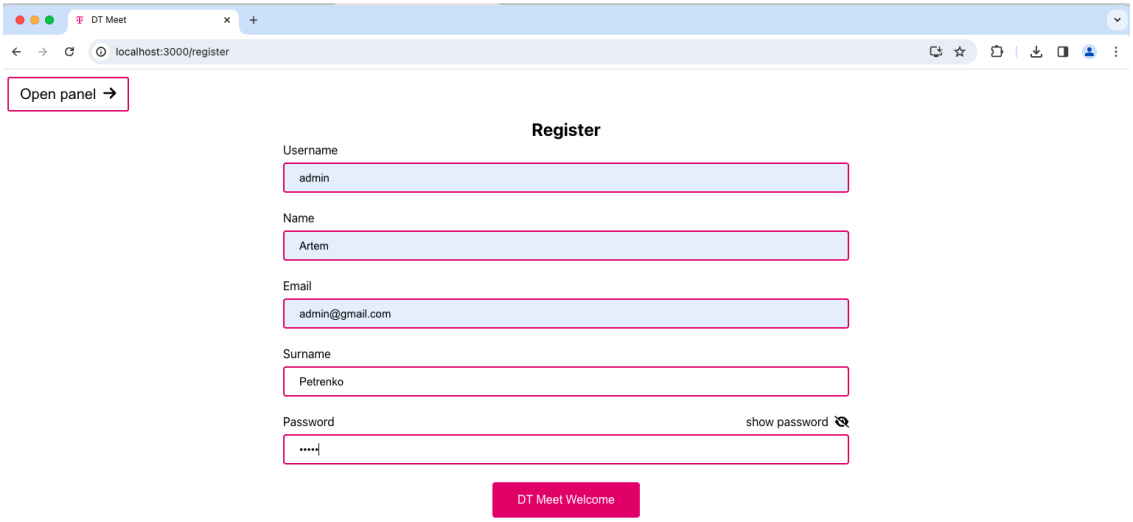

Figure 3.2: Registration page

After putting the credentials to the login, the user will be directed to the main application page.

One of the platform's features is the ability to use the application without mandatory authorization. Authorization adds additional functionality for the

Figure 3.3: Filled login form at login page

user.

3.2.2 Profile

After accessing the application's main page, which is available only to authorized users, a platform participant can open his profile. To do this, click on the avatar icon at the top right corner.

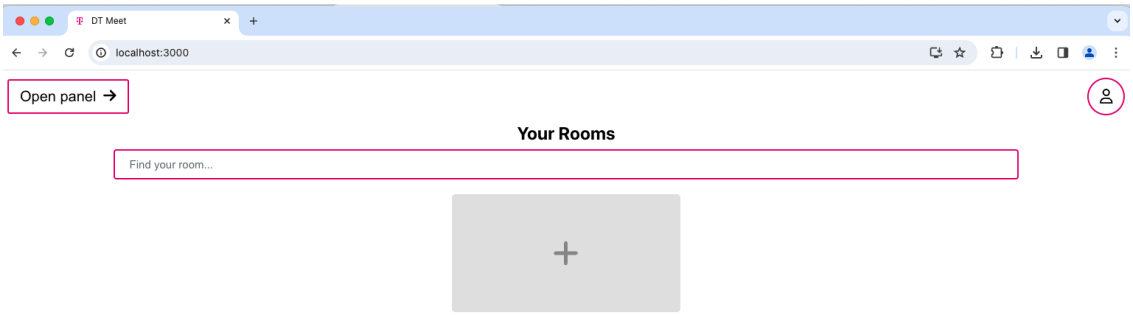

Figure 3.4: Main page of application

By clicking on this icon, the user will see a dialog box with :

- **Manage Profile button** - display the user's full profile.
- **Sign-out** - log out of the account.

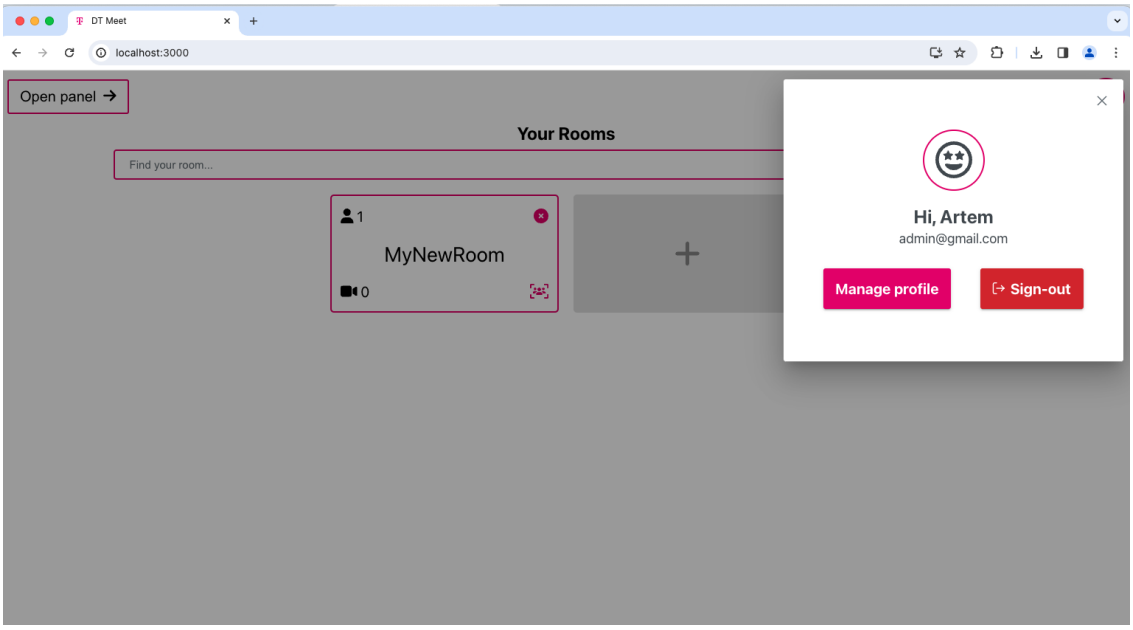

Figure 3.5: Avatar dialog

After switching to viewing the entire profile, the user will have detailed information about himself.

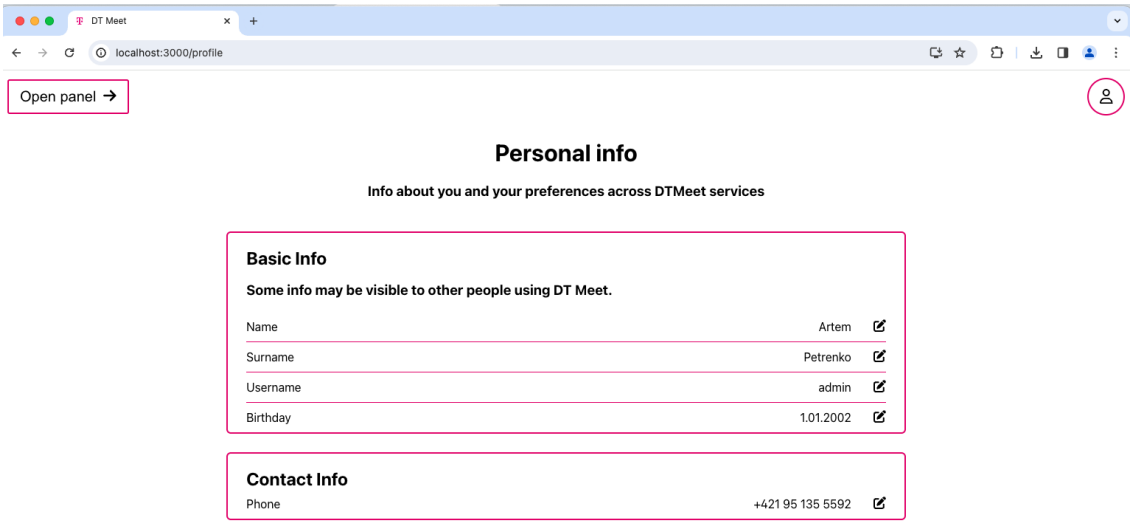

Figure 3.6: Profile page of application

The user can change information about himself by clicking on the edit button for the selected field, which will be updated throughout the system.

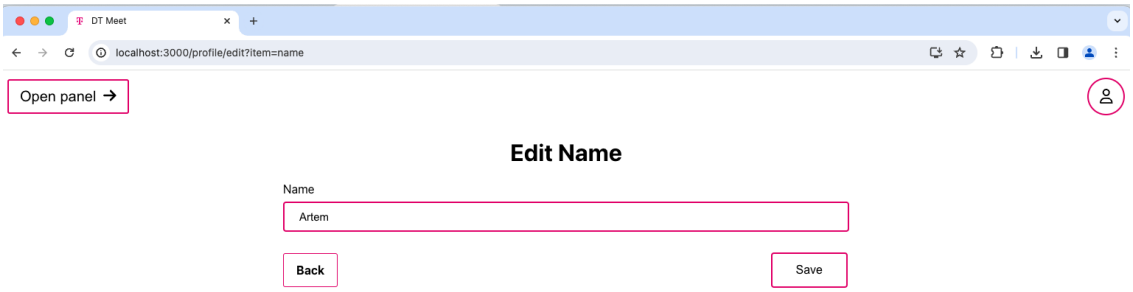

Figure 3.7: Edit profile information view

3.2.3 Rooms managment tool

This functionality is available only to authorized platform users. The application's main page is the Rooms management tool, which has the functions of grouping and saving conference states. Statistics about conferences are also indicated inside the room. You can create a room or be invited by another member. The room can also use the integrated Jitsi or BigBlueButton before the start of each call, all at the choice of the conference participants.

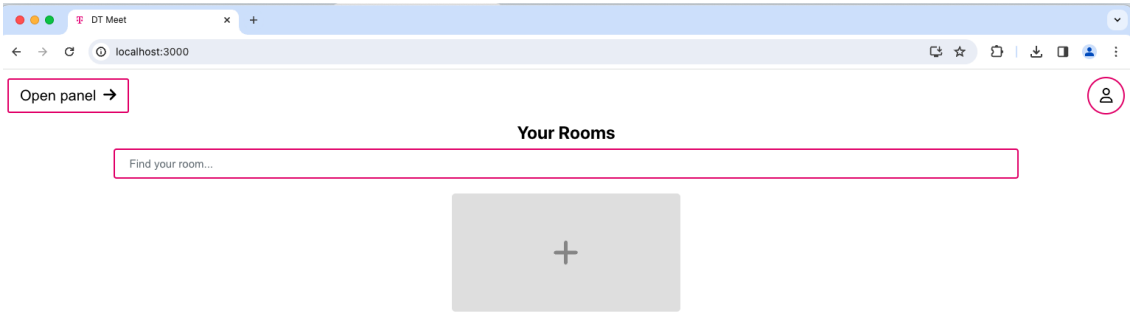

Figure 3.8: Main page of application

To create a conference, you need to start by clicking on the gray square with a plus sign. This will open a dialog window with a form to fill out information about creating a new room.

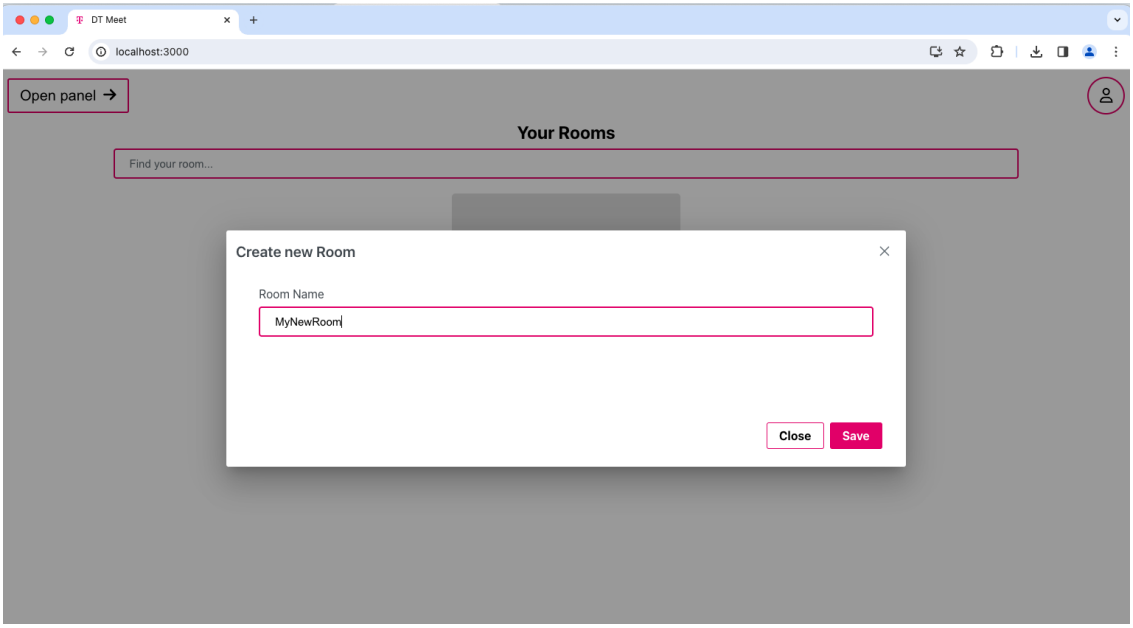

Figure 3.9: Create new room form

After saving, an object indicating a room will appear on the main page with additional information:

- how many conferences were made within the room.
- how many participants are connected to this room.

Also there are 2 action buttons:

- deleting the room.
- obtaining detailed information about the room.

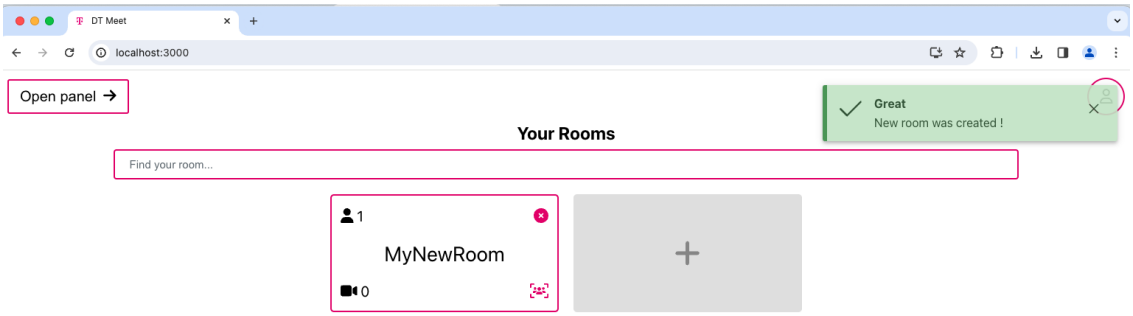

Figure 3.10: Main page with room view

After clicking on the user icon in the lower right corner of the room, the object will display a more detailed room window with additional functionality.

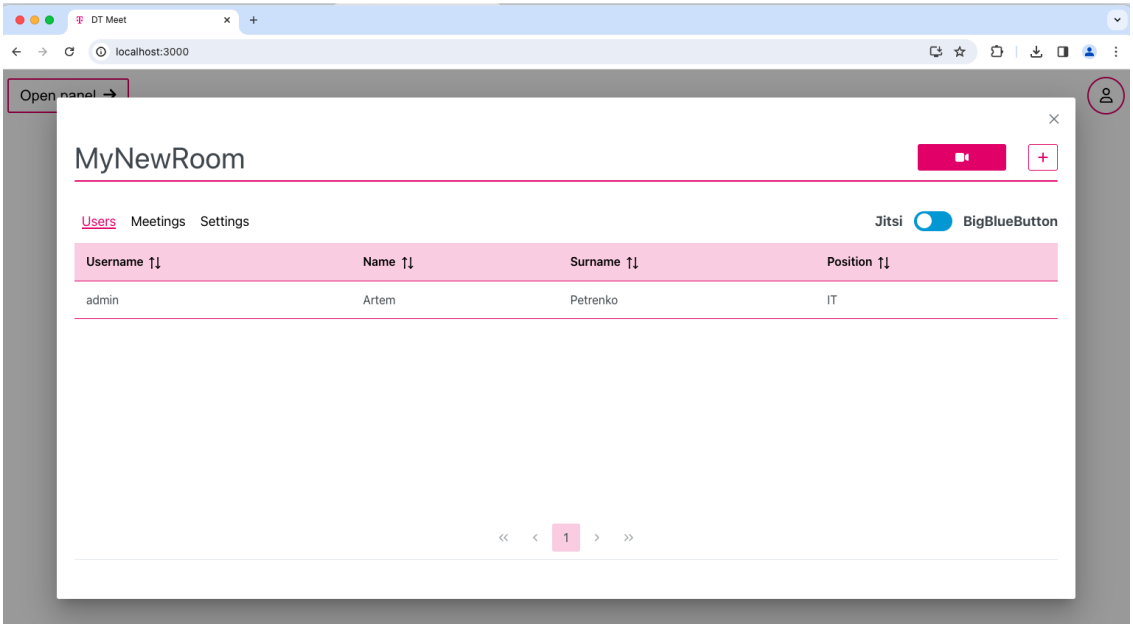

Figure 3.11: Detailed room view

This window contains many different functions:

- **Start conference button.** You can see a button with a white camera icon in the upper right corner of the opened window. This button is multifunctional. A conference will be created inside the room when you click on it. Other users who connect to a room in which a conference is already taking

place can click on the connect button. Also, the camera icon will be changed to the login icon.

- **Add new member of room.** The button for adding a new user to the room is located to the right of the conference launch icon and has a plus icon. When clicked, a field will open inside the dialog window where the user must enter the email address under which the added user is registered. During an ongoing conference in the room, adding a user is impossible; therefore, this button generates an invitation link to the conference. This can be useful for people who do not have an account on the platform.

- **User/Meeting/Settings pages.** This part of the functionality displays additional information for statistics, information about the meetings, and about connected users. There is also a room setting where you can change access passwords to room conferences and much more.

- **Toggle button of video conference technology.** This button is the user's choice of technology for holding a video conference before the start of the meeting in the room.

3.2.4 Navigation side-bar

When the user clicks on the Open panel button in the upper left corner, he will see a sidebar responsible for navigation between the various components of the platform.

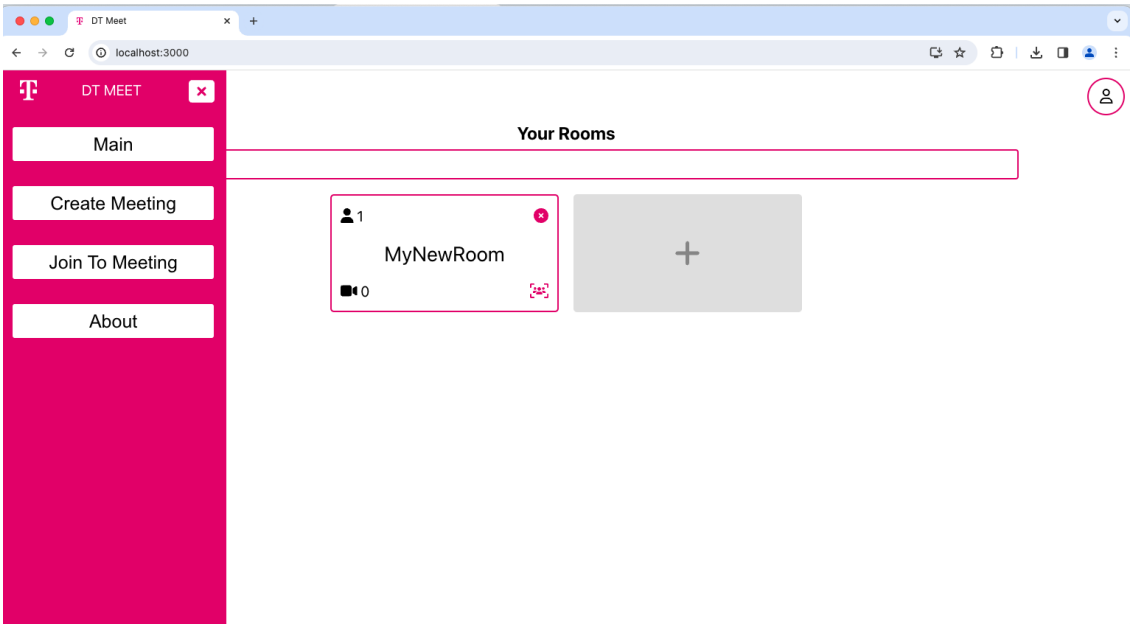

Figure 3.12: Navigation sidebar

3.2.5 Create, Join to the conference

Creating and joining a conference is that part of the functionality that looks almost the same to the end user, but on the side of working out the logic, they are very different. This is also part of the functionality available to authorized and non-authorized users. The only difference is that the authorized user will not have an input field for the username because this data was already received during registration.

Creating a conference involves a form input with a toggle button to choose between Jitsi or BigBlueButton technology. When selecting BigBlueButton, two mandatory fields will be added for launching the BigBlueButton conference: the attendee and moderator password fields. These are important aspects that affect the invitation and joining process on the BigBlueButton server. After submitting the form, the data will be processed. First, a room will be created using the selected integrated technology for video conferences. Then, the conference creator will be connected to it.

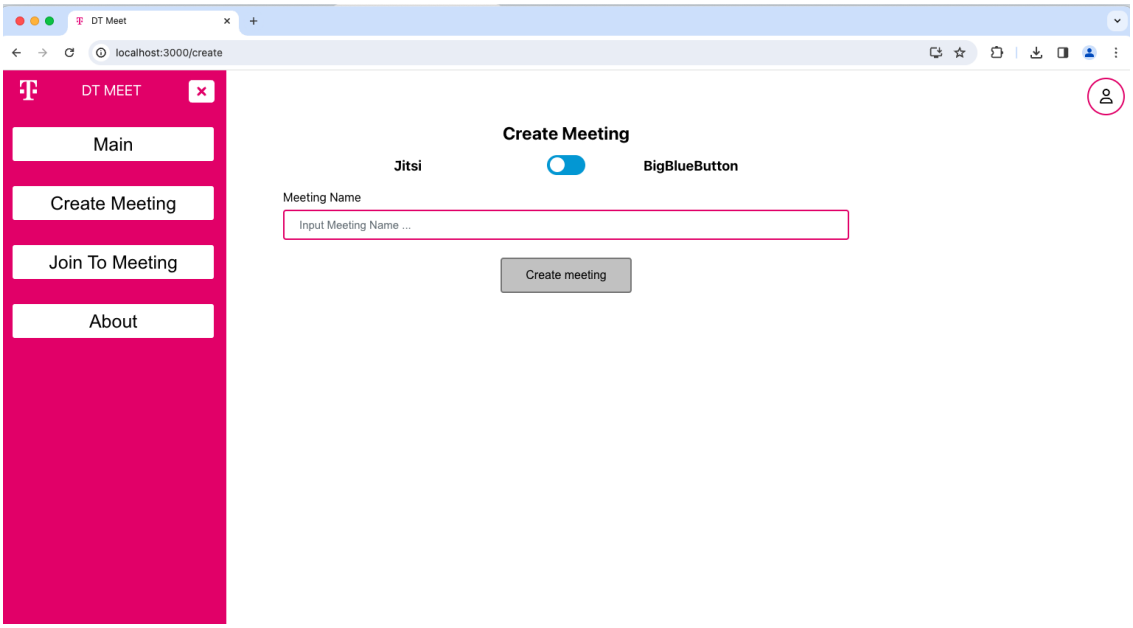

Figure 3.13: Create meeting view

Connecting to a conference is a form in which the main input field is the conference link. During processing, the link is checked to ensure the technology is supported and the data inside the link is correct. Upon completing the verification, the user will be redirected to the conference to which the links relate.

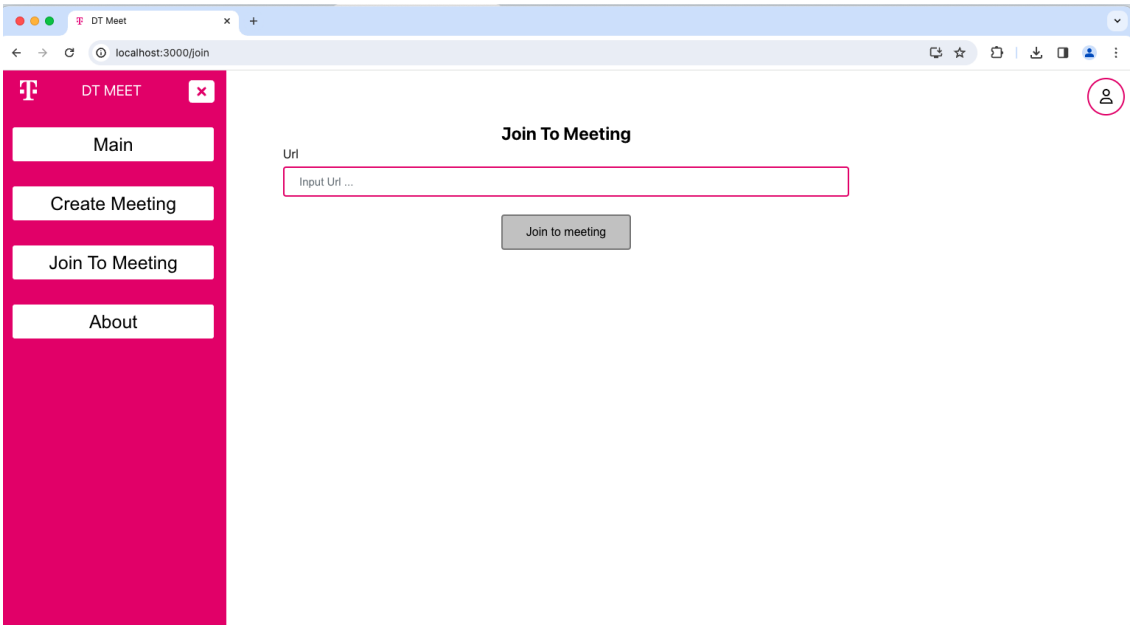

Figure 3.14: Join to meeting view

3.2.6 Jitsi Meet

After redirecting the user to the component responsible for the Jitsi video conferencing technology, the user will see the Jitsi window, which is transmitted directly from the server, and buttons for controlling the conference functionality styled for the DTMeet platform. You can also see invitation buttons at the bottom. You can

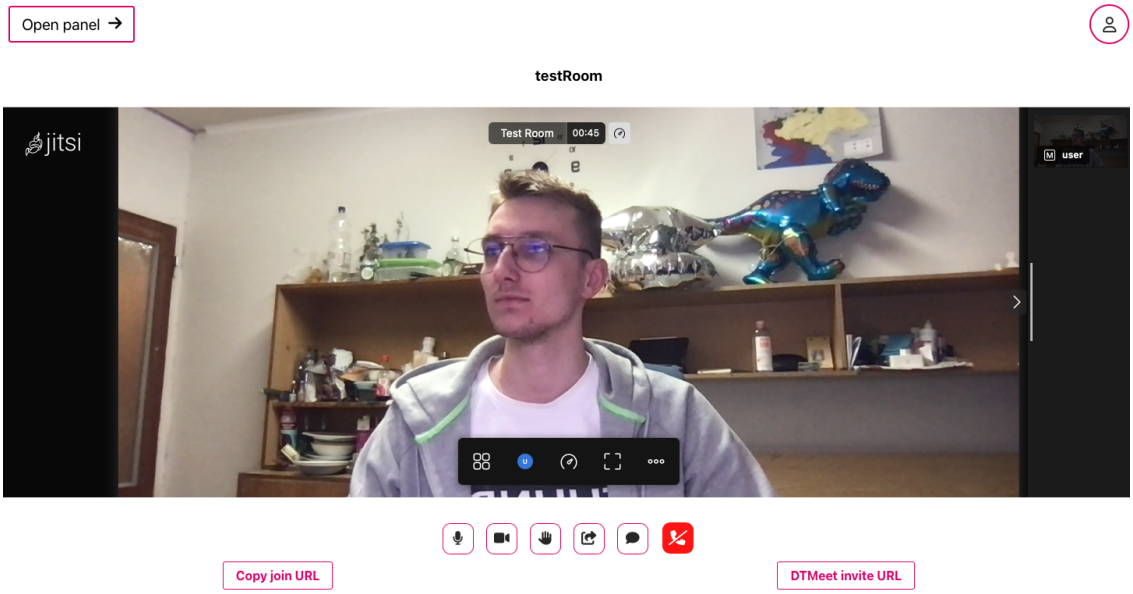

Figure 3.15: Jitsi Meet view

also see invitation buttons at the bottom of the window.

- **Copy join URL button**. Invitation via the Jitsi link to join the conference without using the DTMeet platform. This link goes directly to the Jitsi server.
- **Copy invite link to meeting**. Invitation to join a conference using the invite service of DTMeet.

3.2.7 BigBlueButton

After choosing BigBlueButton video conferencing technology, the user will be redirected to the BigBlueButton component. In this case, the user can use the integrated BigBlueButton technology for communication.

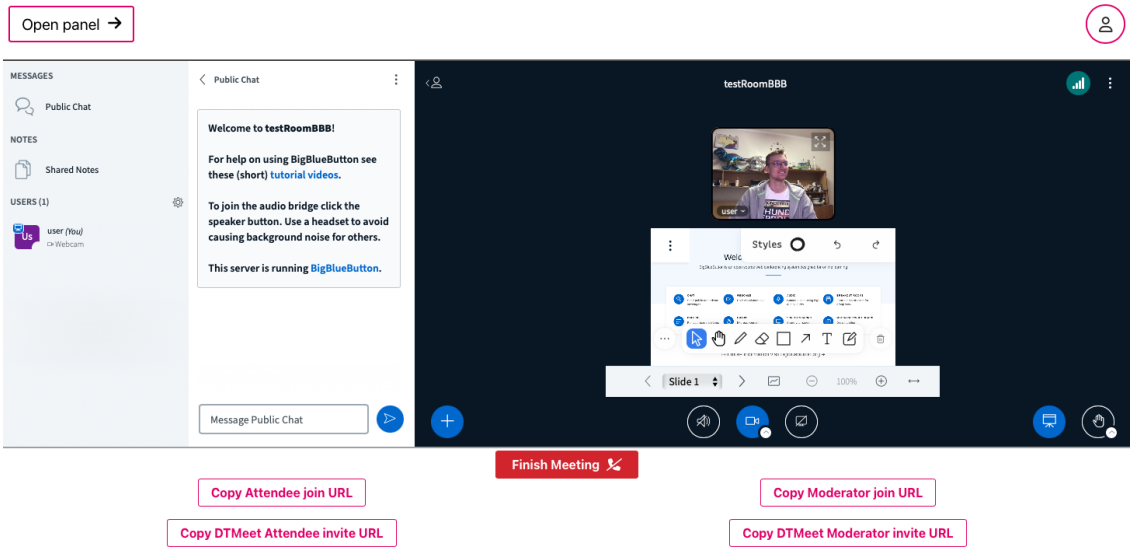

Figure 3.16: BigBlueButton view

Below the video conference window you can see additional buttons for conference management.

- **Copy Attendee join URL** for copying the link for the attendee of BigBlueButton conference to connect directly to the BigBlueButton server.
- **Copy Moderator join Copy URL** for the moderator of BigBlueButton conference to connect directly to the BigBlueButton server.
- **Copy Invite Moderator join URL** for the moderator of BigBlueButton conference to connect via DTMeet technology with BigBlueButton server integration.
- **Copy Invite Attendee join URL** copying link for attendee of BigBlueButton conference to connect via DTMeet technology with BigBlueButton server integration.
- **Finish BBB Meeting.** button to end the meeting BigBlueButton.

3.2.8 About

The About page plays the role of an information page that describes general understanding and information about the project, how it was created, and various support links.

3.2.9 Invite

The invitation service is a special page for connecting to a conference using an invitation link via the DTMeet platform. The application generates the link to which the invited user is redirected. First, the invite component checks the link to ensure it follows the platform's rules and standards. At this stage, a check is made to see if the link has been generated by the DTMeet application.

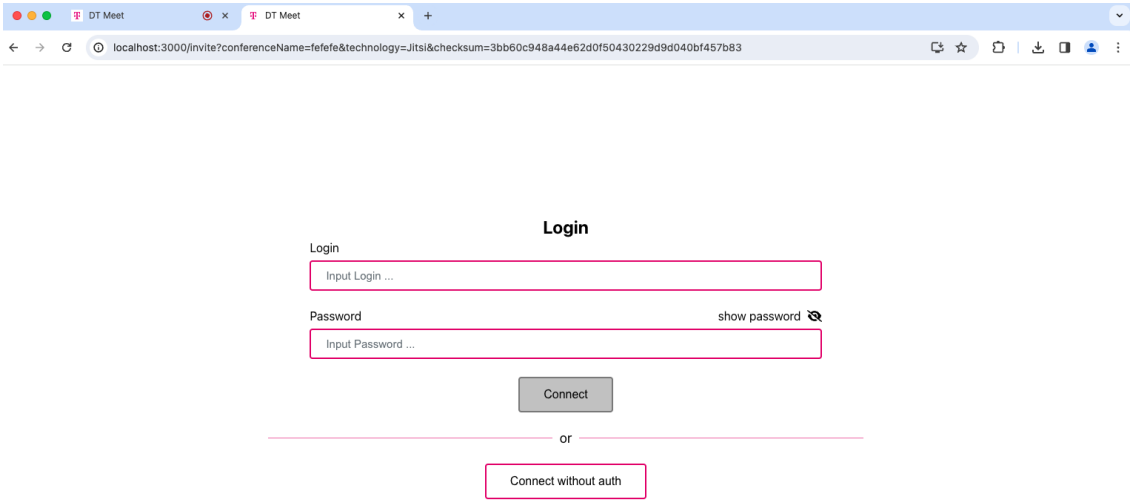

Figure 3.17: Invite service view

The invitation page is a multi-functional component that prompts the user to select a joining method:

- Authorization as an existing user using the account credentials.
- Connecting to a conference without authorization. Users need to enter a username, which will be displayed to all participants in the video conference.

3.3 Usability testing

3.3.1 Background Summary

DTMeet is a web application-based platform used for conducting video conferences. The key feature of the application is the use of open-source technologies

such as Jitsi and BigBlueButton for video conferencing. Additionally, the application supports functionality for both authenticated and non-authenticated users.

As this application serves as the client that will be used as the basis for Deutsche Telekom's communication system, testing all functionalities is a crucial aspect.

3.3.2 Goal

The goal of the testing is to verify all functionalities of the web application, identify platform malfunctions, and address conceptual shortcomings of the application. The testing also aims to find and fix any aspects of the application that could confuse users or degrade the user experience.

3.3.3 Methodology

To test the entire application, four testing scenarios were created that encompass almost all the functionalities of the DTMeet platform. For conducting usability testing, five platform users aged between 20 to 25 years were selected.

Four testing scenarios have been developed:

- Rooms management system test
    1. Registration process.
    2. User authentication.
    3. Creation of a room for video conferences.
    4. Adding another user to the room.
    5. Launch and use of the BigBlueButton video conference.
    6. Exiting the conference.
    7. Launch and use of the Jitsi Meet video conference.
    8. Exiting the conference.
    9. Logging out of the account.
- Create video conference test
    1. Opening the navigation sidebar.
    2. Navigating to video conference creation.
    3. Creating the chosen video conference.
    4. Exiting the video conference.

- Join to video conference test
    1. Opening the navigation sidebar.
    2. Navigating to join a video conference.
    3. Inserting a link prepared by the administrator into the appropriate field.
    4. Connecting to the video conference.
- DTMeet invitation link test
    1. Using an invitation link through the platform, connect to the application by clicking on it.
    2. Choose the method of connection, either with authentication or without.
    3. Connect to the video conference.

Rooms management system test

Based on the data received from the tested users, the following results were obtained.

Test difficulties				
User	Account	Rooms management	Conference connection	Sign out
User 1	No	No	No	Yes
User 2	No	Yes	No	No
User 3	No	No	No	Yes
User 4	No	Yes	No	No
User 5	No	No	No	No

Table 3.1: Rooms management system test difficulties

Account

Based on the data received, none of the users had problems creating an account on the platform or going through the authorization process.

Rooms management

There were issues only for users 2 and 4 at the next stage of room management usage. They needed help understanding how to add a new user to the room for a joint conference.

Conference connection

None of the users had any problems launching conferences in both integrated systems.

Sign out

Only user 1 and user 2 encountered issues during the logout stage. It wasn't immediately clear where to click to access the logout window.

Create video conference test

Based on the data received from the tested users, the following results were obtained.

Test difficulties				
User	Sidebar	Navigation	Setting up conference	Finish conference
User 1	Yes	No	No	No
User 2	No	No	No	No
User 3	Yes	No	No	No
User 4	No	No	No	No
User 5	No	No	No	No

Table 3.2: Create video conference test difficulties

Sidebar

There were problems opening the Sidebar with navigation for users 1 and 3. They couldn't understand how it should open.

Navigation

Navigation for everyone was fine.

Setting up conference

No one encountered any issues during conference setup or video conference technology selection.

Finish conference

At the end of the video conference, no one had any difficulties using the video conference technology.

Join to video conference test

Based on the data received from the tested users, the following results were obtained.

Test difficulties				
User	Sidebar	Navigation	Link insert	Finish conference
User 1	Yes	No	No	No
User 2	No	No	No	No
User 3	Yes	No	No	No
User 4	No	No	No	No
User 5	No	No	No	No

Table 3.3: Join to video conference test difficulties

Sidebar

There were problems opening the Sidebar with navigation for users 1 and 3. They couldn't understand how it should open.

Navigation

Navigation for everyone was fine.

Link insert

When connecting users to the conference via a link, none of the users had any difficulties.

Finish conference

At the end of the conference, users had no difficulties.

DTMeet invitation link test

Based on the data received from the tested users, the following results were obtained.

Test difficulties			
User	Connection method	Connecting to video conference	Finish conference
User 1	No	No	No
User 2	Yes	No	No
User 3	Yes	No	No
User 4	No	No	No
User 5	Yes	No	No

Table 3.4: DTMeet invitation link test difficulties

Connection method

When choosing the connection method, users 2 and 3 encountered difficulties. User 3 couldn't understand how to start a conference without authorization, and user 2 couldn't find the instructions for creating an account in the invitation link window.

Connecting to video conference

Users had no problems connecting to the video conference.

Finish conference

Users had no difficulties ending the video conference.

3.3.4 Test conclusion

All users rated the DTMeet platform quite highly for communication purposes.

According to the users, the main features of DTMeet are its:

- simplicity of use,
- ease of creating and joining conferences,
- great interface,

- access to video conferences without authorization,
- security.

The main difficulty mentioned was the slightly cluttered view of the invitation to video conferences through the platform.

All users agreed that it would be beneficial to add:

- chat functionality within the DTMeet platform for communication between users and within rooms and to add user role capabilities within rooms,
- integrating various planning systems, such as Google Calendar, Microsoft Outlook, etc.
- integrating Google, Microsoft, Facebook authorization,
- The ability for users to customize the theme and main colors of the DTMeet interface in the settings menu, including implementing a black theme.

4 Summary

Since ancient times, people have been actively developing various systems and methods of communication over long distances. In those days, this greatly simplified the lives of many people. With the advent of new technological breakthroughs, people came up with inventions that made it possible to expand the possibilities of communication in increasing distance and quality. One such example could be video conferencing technologies.

Today, it is quite an important tool for anyone. Technologies for video conferencing are quite effective communication tools. Many companies and organizations need good, high-quality solutions that can significantly optimize people's processes and make cooperation more accessible and simple. Such goals have become the basis for the widespread use and holding of video conferences by various companies, organizations, educational institutions, and ordinary users.

This tool has shown its significance during the COVID-19 pandemic, which, in turn, became a real challenge for modern society. This period also became part of the active development of video conferencing technologies, creating new solutions and improving and refining existing systems. Open-source video conferencing platforms were no exception.

Video conferencing greatly simplifies our lives, especially when integrated into various systems, as it allows users to use just one application for multiple tasks. This area is actively developing today, enabling conference participants to increase the sense of presence among interlocutors. In addition, systems of this type have many advantages, making them somewhat competitive with commercial solutions from large companies, especially during integration.

Video conferencing technologies like Jitsi Meet and BigBlueButton are popular tools for integrating existing systems. Because these platforms are open-source, many organizations prioritize such technologies because of code stability, low support costs, and a community that can always help with issues.

Jitsi Meet and BigBlueButton also have a wide range of features for use during calls with a small number of participants and large online presentations.

Integrating such technologies into various systems allows users not only to improve their experience with the platform but also contributes to the further development of the video conferencing field. Overall, video conferencing remains an important tool for ensuring effective communication and collaboration in the modern world.

Bibliography

1. BRUMM, Adam. "Oldest cave art found in Sulawesi. *Science Advances*. 2021.
2. DAVID, Diringer. The Book Before Printing. *Ancient, Medieval and Oriental, Courier Dover Publications*. 1982.
3. NEEDHAM, Joseph. Science and civilisation in China: Paper and printing. *Cambridge University Press.* 2001.
4. HANS-JÜRGEN. Geschichte der Druckpressen. *Frankfurt/Main: Interprint*. 1974.
5. NEMIROVSKY E., Teplov L. New Information about Petr Knyagininsky. *Polygraphic Production*. 1949.
6. End of story for Linotype. *Deseret News*. 1970.
7. Printing Industries of America. *A Composition Manual*. 1953.
8. WHITBREAD, David. The Design Manual. *University of New South Wales Press.* 2009.
9. Claude Chappe (French engineer). *Encyclopædia Britannica*. 2009.
10. WENZLHUEMER, Roland. The Development of Telegraphy, 1870–1900: A European Perspective on a World History Challenge. *History Compass*. 1742.
11. COE, Lewis. The Telegraph: A History of Morse's Invention and Its Predecessors in the United States. *McFarland*. 2003.
12. ELLINGSON, Steven W. Radio Systems Engineering. *Cambridge University Press*. 2016.
13. MAXWELL, J. Clerk. A Dynamical Theory of the Electromagnetic Field. *Philosophical Transactions of the Royal Society of London.* 1865.
14. BONDYOPADHYAY, Prebir K. Guglielmo Marconi – The father of long distance radio communication – An engineer's tribute. *25th European Microwave Conference.* 1995.

15. FARUQUE, Saleh. Radio Frequency Modulation Made Easy. *Springer Publishing*. 2016.

16. DONALD G. FINK A. Michael Noll, David E. Fisher. *television* [The Encyclopaedia Britannica]. [N.d.]. https://www.britannica.com/technology/television-technology/Electronic-systems..

17. BORTH, D. E. mobile telephone. *Encyclopedia Britannica*. 2024.

18. TIMBERG, Craig. A Flaw in the Design. *The Washington Post*. 2015.

19. WICKRAMASINGHE, Shanika. *Codec* [
Cloud Infrastructure Service]. [N.d.]. https://cloudinfrastructureservices.co.uk/how-does-video-conferencing-work/.

20. KAGAN, Julia. *Video Conferencing: How It Works, How to Use It, Top Platforms* [Investopedia]. [N.d.]. https://www.investopedia.com/terms/v/video-conferencing.asp.

21. CHAI, Wesley. *video conferencing* [techTarget]. [N.d.]. https://www.techtarget.com/searchunifiedcommunications/definition/video-conference.

22. CONCISE, McGraw-Hill. Videotelephony. *Encyclopedia of Engineering*. 2002.

23. BORTH, David E. *videophone* [The Encyclopaedia Britannica]. [N.d.]. https://www.britannica.com/technology/videophone.

24. Picturephone. *Today's Engineer*. 2014.

25. ASHENDEN, Pauline. *The History of Video Conferencing from 1870 to Today* [Lifesize]. [N.d.]. https://cisco-telepresence.blogspot.com/2010/07/evolution-of-video-communications.html.

26. PRESS, James. PictureTel Corp. History. *International Directory of Company Histories*. 1999.

27. JOHN SIMLEY, Frank Uhle. *PictureTel Corp.* [ENCYCLOpedia.com]. [N.d.]. https://www.encyclopedia.com/social-sciences-and-law/economics-business-and-labor/businesses-and-occupations/picturetel-corp.

28. CORP., Microsoft. *Microsoft NetMeeting Conferencing Software Provides Easy Voice, Data Internet Communications; Available on the Web Now* [News Microsoft]. [N.d.]. https://news.microsoft.com/1996/05/29/microsoft-netmeeting-conferencing-software-provides-easy-voice-data-internet-communications-available-on-the-web-now/.

29. J, Marescaux. Opération Lindbergh. *Ann Chir*. 2002.

30. *Evolution of Video Communications* [blogspot]. [N.d.]. https://cisco-telepresence.blogspot.com/2010/07/evolution-of-video-communications.html.

31. MALDOW, David S. Zoom's Full Featured UME Videoconferencing Platform Exceeds Expectations. *Telepresence Options*. 2019.

32. ROTH, Emma. Google Meet meets Duo Meet, with Meet in Duo but Duo isn't going into Meet. *The Verge*. 2022.

33. FULAY Amit; Adan, Yariv. Saying hello to Allo and Duo: new apps for smart messaging and video calling. *Google: The Keyword Blog*. 2016.

34. LADIS, Marko. *Virtual Reality (VR) and Augmented Reality (AR) Integration in Video Conferencing: The Future or Just a Fun Gimmick?* [LinkedIn]. [N.d.]. https://www.linkedin.com/pulse/virtual-reality-vr-augmented-ar-integration-video-future-marko-ladis/.

35. GKATZELIS G.I.; Gilman, J.B.; The global impacts of COVID-19 lockdowns on urban air pollutiona critical review and recommendations. *Elem. Sci. Anthr*. 2021.

36. GSHARMA M.; Mindermann, S.; Understanding the effectiveness of government interventions against the resurgence of COVID-19 in Europe. *Commun*. 2021.

37. *Impact of COVID 19 on the Video Conferencing Market.* [Research and Markets]. [N.d.]. https://www.researchandmarkets.com/reports/5013565/impact-of-covid-19-on-the-video-conferencing/.

38. *Web and Video Conferencing SaaS—Global Market Trajectory and Analytics.* [Research and Markets]. [N.d.]. https://www.researchandmarkets.com/reports/5141267/web-and-video-conferencing-saas-global.

39. *How Video Conferencing Improved During the Pandemic* [SalesRoom]. [N.d.]. https://www.salesroom.com/solution/how-the-pandemic-transformed-and-improved-video-conferencing.

40. *What Is a Codec?* [ezTalks]. [N.d.]. https://eztalks.com/video-conference/what-is-a-codec.html.

41. *Difference between Open source Software and Commercial Software* [Geeks For Geeks]. [N.d.]. https://www.geeksforgeeks.org/difference-between-open-source-software-and-commercial-software/.

42. *Difference between Open source Software and Commercial Software* [Turing]. [N.d.].

43. WALSH, Mike. *The 7 Best Video Conferencing Software Platforms for 2024* [DGI Communications]. [N.d.]. https://www.dgicommunications.com/video-conferencing-software/.

44. BRAME, Daniel. *GoToMeeting Review* [PCmag]. [N.d.]. https://www.pcmag.com/reviews/gotomeeting.

45. *StarLeaf* [Perfect video conferencing]. [N.d.]. https://www.perfectvc.com/products-services/our-partners-2/star-leaf2/.

46. *Meet the boss behind StarLeaf, the UK tech firm vying with Zoom* [The Standart]. [N.d.]. https://www.standard.co.uk/business/meet-the-boss-behind-starleaf-the-british-tech-vying-with-zoom-a4550546.html.

47. *Everything you need to know about Google Meet* [Sherweb]. [N.d.]. https://www.sherweb.com/blog/g-suite/google-meet/.

48. JOHNSTON, Scott. Meet the new Hangouts. *Google*. 2017.

49. PIERCE, David. Google is combining Meet and Duo into a single app for voice and video calls. *The Verge*. 2022.

50. BRAME, Daniel. *GWebex by Cisco Review* [PCmag]. [N.d.]. https://www.pcmag.com/reviews/cisco-webex-meetings.

51. FERGUSON, Scott. *AOL Launches AIM Pro Service* [eWeek]. [N.d.]. https://www.eweek.com/enterprise-apps/aol-launches-aim-pro-service/.

52. *Cisco Turns Their Spark into WebEx Teams* [videocentric]. [N.d.]. https://videocentric.co.uk/news/cisco-turns-spark-webex-teams/.

53. DUFFY, Jill. *Microsoft Teams Review* [PCmag]. [N.d.]. https://www.pcmag.com/reviews/microsoft-teams.

54. *Microsoft to Acquire Parlano* [News Microsoft]. [N.d.]. https://news.microsoft.com/2007/08/29/microsoft-to-acquire-parlano/.

55. WARREN, Tom. *Microsoft Teams launches to take on Slack in the workplace* [The Verge]. [N.d.]. https://www.theverge.com/2016/11/2/13497992/microsoft-teams-slack-competitor-features.

56. LAYCOCK, C. *Microsoft classroom preview has officially been 'dropped'.* [Tech community Microsoft]. [N.d.]. https://techcommunity.microsoft.com/t5/Classroom/Microsoft-Classroom-Preview-has-officially-been-dropped/td-p/66657..

57. RODRIGUEZ, Salvador. *Microsoft debuts free tier in competitive workplace chat app market* [Reuters]. [N.d.]. https://www.reuters.com/article/us-usa-microsoft-teams-idUSKBN1K22U0/.

58. DUFFY, Jill. *Zoom Meetings Review* [PCmag]. [N.d.]. https://www.pcmag.com/reviews/zoom-meeting.

59. MOSSBERG, Walter S. A Chance To Call 15 Friends To Video Chat In High Def. *The Wall Street Journal*. 2012.

60. ETHERINGTON, Darrell. *Zoom Raises 6M Series A, Launches Version 1.0 Of Its Radically Different Virtual Conferencing Tool* [TechCrunch]. [N.d.]. https://techcrunch.com/2013/01/28/zoom-raises-6m-series-a-launches-version-1-0-of-its-radically-different-virtual-conferencing-tool/?guccounter=1.

61. DIGNAN, Larry. *Polycom, Zoom forge video conferencing, collaboration pact* [ZDNet]. [N.d.]. https://www.zdnet.com/article/polycom-zoom-forge-video-conferencing-collaboration-pact/.

62. SEVILLA, Gadjo. *Zoom Launches Hardware-as-a-Service Products* [PCmag]. [N.d.]. https://www.pcmag.com/news/zoom-launches-hardware-as-a-service-products.

63. BONIFACIC, Igor. *Zoom is buying a startup to bring real-time translation to video calls* [Engadget]. [N.d.]. https://www.engadget.com/zoom-kites-real-time-machine-translation-184824115.html?guccounter=1.

64. *What is open source?* [opensource.com]. [N.d.]. https://opensource.com/resources/what-open-source.

65. *What is open source?* [RedHat]. [N.d.]. https://www.redhat.com/en/topics/open-source/what-is-open-source.

66. JETHVA, Hitesh. *10 Best Open Source Video Conferencing Software Self Hosted* [Cloud Infrastructure Services]. [N.d.]. https://cloudinfrastructureservices.co.uk/10-best-open-source-video-conferencing-software-self-hosted/.

67. *SIP Communicator: Interview with Emil Ivov* [gulli.com]. [N.d.]. https://archive.ph/20130126010545/http://www.gulli.com/news/sip-communicator-interview-2009-07-15#selection-521.0-521.42.

68. IVOV, Emil. *Authentication on meet.jit.si* [jitsi.org]. [N.d.]. https://jitsi.org/blog/authentication-on-meet-jit-si/.

69. *Here's the roadmap that we currently have in mind* (*all items are also represented by issues that could be found in the issue tracker*) [jitsi.org]. [N.d.]. https://desktop.jitsi.org/Development/Roadmap.html.

70. *Jitsi 2.6.5390* [GitHub]. [N.d.]. https://github.com/jitsi/jitsi/releases.

71. *Atlassian acquires video conferencing company Blue Jimp* [Zdnet]. [N.d.]. https://www.zdnet.com/article/atlassian-acquires-video-conferencing-company-blue-jimp/.

72. *8x8 Acquires Jitsi Video Communications Technology From Atlassian* [businesswire]. [N.d.]. https://www.businesswire.com/news/home/20181029005192/en.

73. DAS, Bhaskar Narayan. *Differences Between Jitsi Meet vs Zoom Video Meetings* (*Comparison*) [Cloud Infrastructure Services]. [N.d.]. https://cloudinfrastructureservices.co.uk/jitsi-vs-zoom/.

74. DAS, Bhaskar Narayan. *Best Top 5 Webrtc Video Conference Open Source Solutions* [Cloud Infrastructure Services]. [N.d.]. https://cloudinfrastructureservices.co.uk/best-top-5-webrtc-video-conference-open-source-solutions/.

75. SUTHERLAND, Richard. *BigBlueButton review* [TechRadar.pro]. [N.d.]. https://www.techradar.com/reviews/bigbluebutton.

76. *The History and Latest Features of BigBlueButton: An In-Depth Look* [HigherEdLab]. [N.d.]. https://higheredlab.com/bigbluebutton-history/.

77. *FAQs* [Docs BigBlueButton]. [N.d.]. https://docs.bigbluebutton.org/support/faq/#why-is-this-project-called-bigbluebutton.

78. *Releases* [Git Hub]. [N.d.]. https://github.com/bigbluebutton/bigbluebutton/releases.

79. *Advantages and Disadvantages of BigBlueButton Configuration* [Yatharthriti]. [N.d.]. https://www.yatharthriti.com/blog/advantages-and-disadvantages-of-bigbluebutton-configuration/.

80. *Collaborate in Google Meet using Jamboard* [Support Google]. [N.d.]. https://support.google.com/meet/answer/10071448.

81. LIDWIN, Will. *Video Conferencing Trends and Innovations Shaping the Future* [elevITy]. [N.d.]. https://www.elevityit.com/blog/video-conferencing-trends.

82. *Navigating the Future of Video Conferencing in 2024* [onpassive]. [N.d.]. https://onpassive.com/blog/navigating-the-future-of-video-conferencing-in-2024.

83. KING, Nicole C. *Why Video Conferencing Is The Future Of Communication* [LinkedIn]. [N.d.]. https://www.linkedin.com/pulse/why-video-conferencing-future-communication-nicole/.

84. CHATTERJEE, Amriten. *Revolutionizing Meetings: How AI and Automation are Transforming Video Conferencing Tools* [LinkedIn]. [N.d.]. https://www.linkedin.com/pulse/revolutionizing-meetings-how-ai-automation-video-tools-chatterjee/.

85. LADIS, Marko. *Virtual Reality (VR) and Augmented Reality (AR) Integration in Video Conferencing: The Future or Just a Fun Gimmick?* [LinkedIn]. [N.d.]. https://www.linkedin.com/pulse/virtual-reality-vr-augmented-ar-integration-video-future-marko-ladis/.

86. ACADEMY, EDGE. *WHAT IS A WEB APPLICATION?* [StackPath]. [N.d.]. https://www.stackpath.com/edge-academy/what-is-a-web-application/.

87. CONTRIBUTOR, TechTarget. *web application (web app)* [TechTarget]. [N.d.]. https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app.

88. *What do client side and server side mean? | Client side vs. server side* [CloudFlare]. [N.d.]. https://www.cloudflare.com/learning/serverless/glossary/client-side-vs-server-side/.

89. *What is a Web Application?* [AWS]. [N.d.]. https://aws.amazon.com/what-is/web-application/.

90. MCKENZIE, Cameron. *JavaScript vs. TypeScript: What's the difference?* [Tech Target]. [N.d.]. https://www.theserverside.com/tip/JavaScript-vs-TypeScript-Whats-the-difference.

91. *Introduction to the server side* [developer mozilla]. [N.d.]. https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Introduction.

92. *An overview of HTTP* [developer mozilla]. [N.d.]. https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview.

93. SHAH, Aakash. *Top Advantages of MySQL for Developers.* [Linkedin]. [N.d.]. https://www.linkedin.com/pulse/top-advantages-mysql-developers-aakash-shah-3ihxc/.

94. ALLEN, Kieron. *Jitsi Meet review* [techradar.pro]. [N.d.]. https://www.techradar.com/reviews/jitsi-meet-review.

95. *Architecture* [Jitsi Meet Handbook]. [N.d.]. https://jitsi.github.io/handbook/docs/architecture.

96. *SDKs* [Jitsi Meet Handbook]. [N.d.]. https://jitsi.github.io/handbook/docs/category/sdks.

97. *Self-Hosting Guide - Overview* [Jitsi Meet Handbook]. [N.d.]. https://jitsi.github.io/handbook/docs/devops-guide/.

98. *Authentication on meet.jit.si* [Jitsi]. [N.d.]. https://jitsi.org/blog/authentication-on-meet-jit-si/.

99. *Community-run instances* [Jitsi Meet Handbook]. [N.d.]. https://jitsi.github.io/handbook/docs/community/community-instances/.

100. *Community-run instances* [Jitsi Meet Handbook]. [N.d.]. https://web.archive.org/web/20231208185450/https://jitsi.github.io/handbook/docs/community/community-instances/.

101. *Troubleshooting Guide* [bigbluebutton doc]. [N.d.]. https://docs.bigbluebutton.org/support/troubleshooting.

102. *npmjs bigbluebutton-js library* [https://www.npmjs.com/package/bigbluebutton-js]. [N.d.].

List of Appendixes

Appendix A CD medium – final thesis in electronic form,

Appendix B User manual,

Appendix C Deployment manual