

Task – Remote communication software

Functional requirements

A company creates a new remote communication software to serve business needs for the communication between employees inside a multi-national company. The application should allow employees to make video and voice calls, share their desktop, and exchange files and messages.

The software provides the following features to the employees:

- *Virtual meeting room:* Employees could initiate, and join virtual meeting rooms through the application website. The creation and joining of rooms is done through the company emails and passwords, and using the cooperate users' directory. By starting the virtual room from the website, a user-interface (UI) starts on the user desktop. The virtual room UI provides several sharing possibilities:
 - *Desktop sharing:* One participant in the room could have the role of the presenter and share his desktop. In addition, the user could specify only certain applications on the desktop to be shared. In case, he does not want to share his whole desktop.
 - *Whiteboard:* A common document to write and draw on. Multiple users could write on the board.
 - *Voice call:* The initiator of the meeting room has the right to associate a voice communication to the meeting room. In this case, the voices from mic from different participants in the meeting room are combined/mixed to provide an alternative for using expensive telephone calls.
 - *Video sharing:* Participants to the virtual room have the option to share their video. In this case, camera inputs from their computers are captured and shared to other members of the room in an organised manner on the user interface. Multiple-users could share their video with a maximum number of 6 users at the same time. Users could enable and disable their video sharing.
 - *Files and messages sharing:* Participants could send messages and files publically to all members of the room, as well as privately to certain members of the room.
- *Manage communication:* Creators of virtual meeting rooms have the flexibility to control the type and quality of their communication. The user can decide on the communication protocols, and media formats (codecs).
 - The users can communicate over a number of different Voice over IP (VoIP) protocols (e.g. XMPP, SIP). These settings need to be specified before starting a virtual room session.
 - The system should consider that each type of sharing might require different protocols. While protocols could also be combined with each other in a pipeline. For example, protocols like SIP and XMPP are used to determine the user addresses and media format, and then provide information to the RTP protocol to move voice and video over the internet.
 - The user can control the type of media formats. In this, the user need to make a compromise between high quality and performance. A higher quality media format could affect the performance.

Non-Functional requirements

- *Portability*: The application needs to run on Windows, Apple and Linux operating systems.
- *Maintainability*:
 - The system should be designed to facilitate developing with different protocols and adding new protocols within 2 weeks of development. Taking into consideration that protocols have differences in their capabilities (e.g. SIP do not support server-stored contact list).
- *Performance*: Audio and video streaming should be real-time.
- *Scalability*: The system should be scalable to the increase in the number of employees in the company. Currently there are more than 100,000 employees. An increase in the number of employees is expected within the next 5 years with over the double.

Constrains

- After analysis, there is no common technology for capturing video from web camera from the 3 operating systems. The architect decided to use 3 different technologies: DirectShow on Windows, QTKit on Mac and Video on Linux. Each is developed using different programming language.

Task goal

The architect would like to design and decompose the logical and physical architecture of the system. The logical architecture specifies the components of the system, and their relationships. While the physical architecture specifies the system components and their deployment on servers. Propose and model the logical and physical architecture for the system, which fulfils the requirements and constraints. In your design, specify and justify the components, their relationships, and the different architectural solutions (e.g. patterns), which you used in your design.

To design the logical architecture of the system, follow the following three steps:

- 1) Identify possible components of the system and their dependencies. A component could be a user interface component, a database or a processing component.
- 2) **Use the following patterns to structure the system:**
 - a. Pipes and filters architecture pattern.
 - b. Peer to peer architecture pattern.
 - c. Publish/subscribe architecture pattern.

To design the physical architecture of the system, **use the following patterns:**

- a. Workload distribution cloud pattern.
- b. Cloud elastic resource capacity architecture pattern.

The solution of the task is expected to be one or two architecture models and an explanation for the architecture, which justify the design decisions.