**- Discuss the most influential architectural drivers of the system.**

Functional requirements:The ability to have a video conversation.

Without a working video and audio conversation the program won’t be fulfilling its main purpose.

Quality attributes

Scalability

The systems needs to be reliable when the number of users increase. Since conversations are handled between two peers rather than over a centralized server the system, the system should scale rather well. This is because each peer handles its own communication.

Reliability

Given that external factors are not the problem, the system should not crash. Should it crash, attempting recovery should be started automatically in less than 2 seconds.

Since the system is communicating directly between two users we eliminate the problem of external systems failing.

Performance

The system should be able to capture, process and display video data with less than 3 seconds delay between peers.

**- Considering the SEI View Model, describe in detail the ‘Component-and-connector structures’ and ‘Allocation structures’ views of the system.**

See fig. 1 and fig. 2.

The system has temporal coupling and referential coupling, due to being a peer-to-peer video-conversation system.

**- Describe the architectural style that, in your opinion, meets best the system requirements.**

We believe the peer-to-peer architectural style is the most descriptive style for this system.

In peer-to-peer all peers are individual and equally important, if one fails the failure doesn’t threaten the health of the system as a whole. This provides high scalability and flexibility, which is what this system desires.

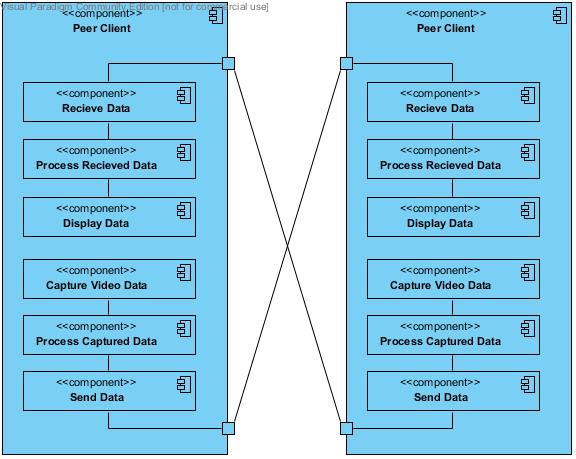
**- Your colleague argues that other architectural styles may fit well to the specification of the system. What could those styles be? Defend your choice of style against your colleague’s proposed solutions. Discuss the relevance of the other alternative styles to the described system.**

Pipes and filters -

While the system internally uses a ‘pipes and filters’-style we find that peer-to-peer better describes how the system as a whole works, with clients communicating with each other, which is the entire point of defining an architectural style: To easily describe the system as a whole.

This style can consists any numbers of filters, and filters can have any numbers of input/output pips. Therefore one user can participate in server conversion. Filters can transform or filter data(segment video,subsitute background,encode and compress data) before passing it via network to another filter. Also all the filters can work at same time, which means two parties can communicate by sending video simultaneously.

*Fig. 1 - Component and Connector Diagram*



*Fig. 2 - Allocation Diagram*

