**NFR Impact**

The implementation of a group chat feature to Jami has some effects on maintainability. As such, the chosen implementation allows message related data to be stored locally on user’ devices via the addition of a database within the LRC. In order to gain access to the group chatting functionality, users would be required to update their software. Regarding evolvability, developers can implement design changes and new features, such as reactions to messages and emojis, by making additions to the Plugins’ subsystem. The proposed enhancement and implementation method would allow for additional features to be added with ease. The group chatting mechanism can be continuously improved upon and further developed in a similar manner to how developers are currently maintaining Jami. In terms of testability, testing the newly added feature and its counterparts is essential to ensure that errors and bugs are eliminated. Additionally, pre-existing cases of Jami should be re-tested as to check if the new additions to the subsystems did not cause any unwarranted issues elsewhere. Finally, it is crucial that Jami maintains its structural integrity in order to adhere to its core principles. In this regard, latency, usability and reliability should not be worsened as a result of the added feature.

**SEI SAAM Architectural Analysis**

**Utilizing Existing Subsystems and OpenDHT to Implement Group Chat Functionality**

1. The major stakeholders are the users and the developers of Jami.
2. The users’ top NFR is privacy. Users are also likely interested in reliability, usability and response time. The developers’ top NFR is reusability. Developers of Jami are also likely interested in high cohesion and limited coupling of parts. In this regard, the developers would implement the enhancement feature by reusing existing source code and methods in relation to messaging. A key notion would be to reuse code and utilize existing systems to implement the feature without infringing upon existing features. Thus, to limit dependencies and adhere to existing principles in adding group chat functionality to the platform.
3. This implementation adds unique group chat subsystems within the already existing subsystems whilst maintaining the foundational peer-to-peer architectural style of Jami. From a user perspective, privacy is maintained since the existing subsystems are utilized (with some additions and modifications), messaging data is stored locally via the database within the LRC, and a peer-to-peer structure is maintained.

From a developers’ perspective, although reusability is limited, it can be maximized. Additionally, high cohesion and limited coupling of parts can be achieved.

1. In conclusion, by making necessary additions to existing subsystems and utilizing OpenDHT, user privacy is protected and the core principles of Jami are maintained.

**Dedicated Server to Implement Group Chat Functionality**

1. The major stakeholders are the users and the developers of Jami.
2. Given that they are using Jami, the users’ top NFR is privacy. Additionally, reliability, usability, and response time are likely important to users. The developers’ top NFR is reusability and maintainability. In this regard, they are interested in implementing this enhancement feature by reusing and building upon pre-existing libraries and source code. In terms of maintenance, they are interested in carefully updating and maintaining the newly added server.
3. From a user perspective, privacy is better maintained if the existing openDHT server is utilized and data is stored locally on users’ devices via the LRC. Usability is not affected in this implementation since the server would likely be online at all times (excluding times for server maintenance and large updates). However, reliability and response time could potentially be improved with the addition of a dedicated server (this is dependant on how well the server is maintained).

From a developers’ perspective, reusability is improved if a pre-existing server is used. As such, this implementation would not be ideal. However, maintaining a newly added, dedicated server will likely be easier, given that the server is well structured.

1. In conclusion, utilizing openDHT to add the group chat functionality is a safer approach as it protects user privacy and maintains a peer-to-peer architecture. As such, it is more in-line with Jami’s core principles. There exist some potential improvements that could result from this client-server approach. However, these improvements cannot be justified in terms of the purpose of Jami and user privacy.