

CSE 155: Final Report

Author¹

¹Adrian Darian, Orion Johnson, Shivanshu Gupta

adarian@ucmerced.edu, ojohnson2@ucmerced.edu, sgupta35@ucmerced.edu

1. Introduction

Our project will be a tool that allows a user to link multiple chatting or messaging services together. For example, if your company uses Slack or hangouts to communicate, but you have some private chats with co-workers on facebook messenger. The user can now link all the accounts together and read or send messages in between. Starting a meeting via video call, will be as simple as selecting which platform the meeting will be on from your linked accounts and sending an invite to that group. We are not forcing users to subscribe to our service or anything, it would be free to use for as much and as long as the user wishes. The idea behind it, mainly came from teams onboarding new members and watching managers explain how they use this service for X and this other service for Y, while for all other forms of talk they use service Z.

2. Problems

When working at a company you are typically introduced to a new platform to communicate with your team in or provided with one.

1. Older generation forced to adept to a new platform
2. Multiple platforms with similar channels to communicate
3. Different organizations that all would like their members to communicate together but do not want a new platform
4. Managers having to spend time teaching new employees or creating a "how to" doc for them

3. Why/How

Looking at this scenario, you are in a company that has an engineering team, logistics team and a marketing team. You are tasked with communicating with individuals from another team. Normally you would need to join what ever platform they use on their team and then message each individual there. Now you have multiple platforms to check each day at work. With a bridge you would only need to be invited to a channel or group from that team and you can start communicating right away.

4. Design

We built a single server that acts as an entry point to bridge different platforms together. This allows a user to message with their personal account on any connected platform and all other members that can view or have access to that channel will receive the message and may reply. Because we can do all this with one server the whole project is actually quite a simple design. The only difference we would need to perform would be what the names and avatar icons are when new members join the new channels.

5. Implementation

To connect each platform together was a little more difficult than imagined. For example to link specifically Slack and Discord to this bridge we need to build a bot for each platform and invite it to each discord server and slack workspace. This is by design so the bot is the one forwarding the messages to the other platforms. While for Microsoft Teams we were able to load the bridge as an application on Microsoft Azure and have that forward messages through it to the deployed bridge server. As per Gitter and Telegram platforms we simply needed to create a temporary user and all messages would filter through them. Leaving it to just invite the user to each channel and everything was setup. As for Whatsapp that was a lot more difficult, due to the amount of encryption in Whatsapp we could not simply add a bot. We had to follow Whatsapp's online guide and build a Webapp host and configure a publisher/subscriber connection stream between that and the channels in Whatsapp.

6. Results

We tested the deployed application out with a few friends and coworkers from different clubs and companies and got an overwhelming positive feedback. Not only was it more convenient to just open one app of choice and start talking to anyone on your team (test group), but the accessibility to even start a private conversation with someone when they were on Discord and the other was on Telegram, was seemless.

However there were many concerns raised about the security of opening the platforms up to one communication channel. This does lead to a single point of failure for a breach that leads to all of a user's messaging history. This is something that we found no solid answer for given how the project is set up at this time. We feel the ultimate answer to this would be to have 'OneStopShop' to be its own standalone application with high encryption.

7. Conclusion

In conclusion with more time to figure out how to connect more platforms together and researching more into encryption and security this could be a actually company requirement or necessity. Right now it just lacks the breath of how many platforms can be connected and especially how secure communicating to a channel on another platform is.

References