**SealTalk**

**By ChuYu Xu, YanZhi Fang, YunQing Liu, YuXin Duan, ZuXin Li.**

*Wuhan University*

**Table of contents**

[1. Introduction 1](#_Toc21008125)

[2. Stakeholder Analysis 2](#_Toc21008126)

[2.1 Stakeholders 2](#_Toc21008127)

[2.2 Power-interest 3](#_Toc21008128)

[3. Context View 3](#_Toc21008129)

[3.1 System Scope 3](#_Toc21008130)

[3.2 External Entities 3](#_Toc21008131)

## 1. Introduction

Instant messaging is the most popular way for people to communicate on Internet, and various kinds of instant messaging software are emerging one after another. Service providers also provide more and more abundant communication services. SealTalk is an open source instant messaging App based on the Rongyun Instant Messaging SDK (IMKIt).

It aims to provide developers with IM development needs with reference development practices, thereby reducing the time from development to online application, and devoting more energy to the application's own business implementation. IMKit includes two UIs: session list and session page. SealTalk adds login UI, contact UI, discovery UI and "I" UI to provide a complete set of instant messaging App DEMO instances.

SealTalk contains session scenarios including single chat, group chat, chat room, discussion group and so on. Conversations between users can be accomplished through the delivery of multiple message types. It includes built-in message types such as text, voice, picture, location, and custom message types such as red packets, business cards, etc. Some information about the message, such as whether the message is sent successfully, whether the message is read or not, will be immediately fed back to the user.

In the following section, we will take the reader through the workings and development of SealTalk, showing our results of anaysis on it. The analysis will start by explaining about stakeholders and context view. Then in order to have a deeper insight on SealTalk's technical side, the analysis is continued by describing on functional view and development view. To understand SealTalk even further, evolution perspective, which shows the process of how SealTalk develops into what it is today, is also presented. At the end, we will have a conclusion.

## 2. Stakeholder Analysis

To start off our analysis, we will look at the stakeholders involved with SealTalk. A

stakeholder is an entity of a system architecture that consists of an individual or an

organization that has importance and interest to realize a system.

### 2.1 Stakeholders

**Acquirers**

RongCloud provides real-time communication cloud services. They release SealTalk-android as a demo application for their cloud services.

**User**

As a real-time communication software, SealTalk has a very broad user base. The user could use it as a tool for personal communication, or to send message in a company. Notice that SealTalk`s basic function are built for general usage, enterprise user may need to modified their part.

**Suppliers**

SealTalk-android is built upon Java, and deployed on Android. They use Android Studio as development environment.

**Development & Testing**

In GitHub record, most of the early commit are given by jenkinsrc and Jianli Zhou. They are the employees of RongCloud Inc. RongCloud has assigned a group of nearly 5 people to develop SealTalk-android.

**Competitors**

QQ and WeChat are strong competitors to SealTalk who provides similar services and have a much wider user.

**Maintainers**

Commits that submit later are mainly given by rc-huangxiujun, an employee of RongCloud. There is not so much fork record on GitHub, but individual developers may customize SealTalk-android for their purpose.

### 2.2 Power-interest

In the stakeholder analysis, many different actors with distinct roles, power and interest have been analyzed. The predominant stakeholder of SealTalk is RongCloud Inc., which has the most power and interest in this project.

RongCloud has already completed two rounds of financing:

Round A: RongCloud announced the completion of round A financing of 50 million yuan, which was led by ZTE.

Round B: RongCloud announced the completion of the B round financing of 100 million yuan, which was invested by JinPu investment, TianXing capital and Shanghai JiaCun investment.

Because these firms also invest in other various products, we can consider that the investment means these companies have high power but moderate interest in RongCloud’s product, as well as seal talk, to be specific.

Since Seal Talk is built upon Java and Android Studio (based on IntelliJ IDEA), the programming language and the developing environment can be regarded as more powerful actors than other suppliers, but having little interest. The developers, testers, maintainers can be places in the middle of the graph, a little bit over the moderate grade. In addition, RongCloud has also cooperated with some professional technology providers, which provide additional development support. They can be classified as having not so much power and maintaining some interest.

The two least powerful stakeholders are the individual users and the competitors, while competitors have the highest interest. Apparently, they want to figure out how does the Seal Talk perform, what technologies are included and its advantages and shortcomings. Companies who use SealTalk are considered to have more power than individual users, since the cooperation among companies will force RongCloud to optimize their products and implement more functions, thus providing higher services.

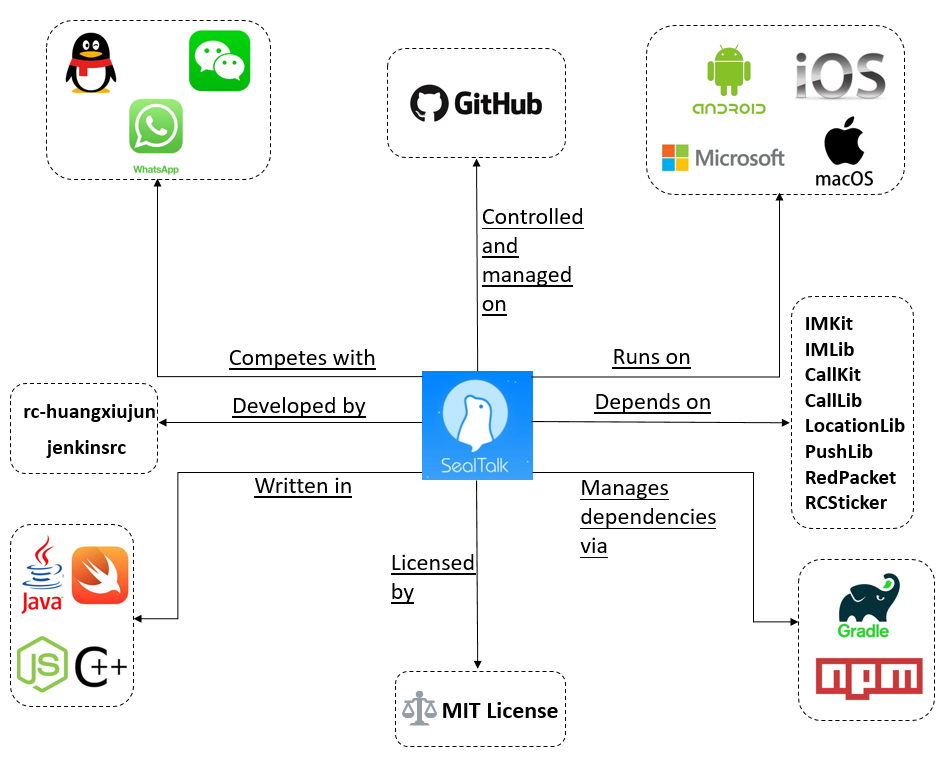
## 3. Context View

### 3.1 System Scope

SealTalk is an open source instant messaging Android app based on the RongCloud Instant Messaging SDK (IMKIt), designed to provide developers with development needs to develop development practices that reduce the time it takes for applications to go from development to launch. Invest in the application's own business implementation.

### 3.2 External Entities

The context model of SealTalk is illustrated in Figure 1. This modeling figure depicts SealTalk in the center, surrounded by the external entities it interacts with.



*Figure 1: External Entities*