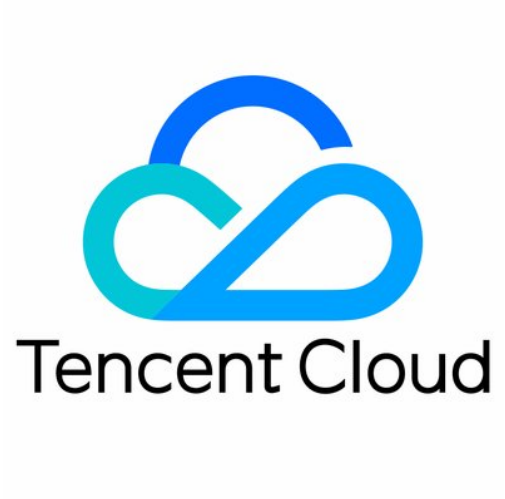
Chat

Migration

Product Documentation



Contents

Migration
Migration Solutions

Last updated：2024-02-07 17:33:32

Instant Messaging (IM) has a wealth of experience in high-concurrency and high-reliability operations. For app developers who are using self-developed or third-party instant messaging services and hope to integrate IM, the issue of migration needs to be considered. IM provides targeted migration solutions for different scenarios.

Terminology

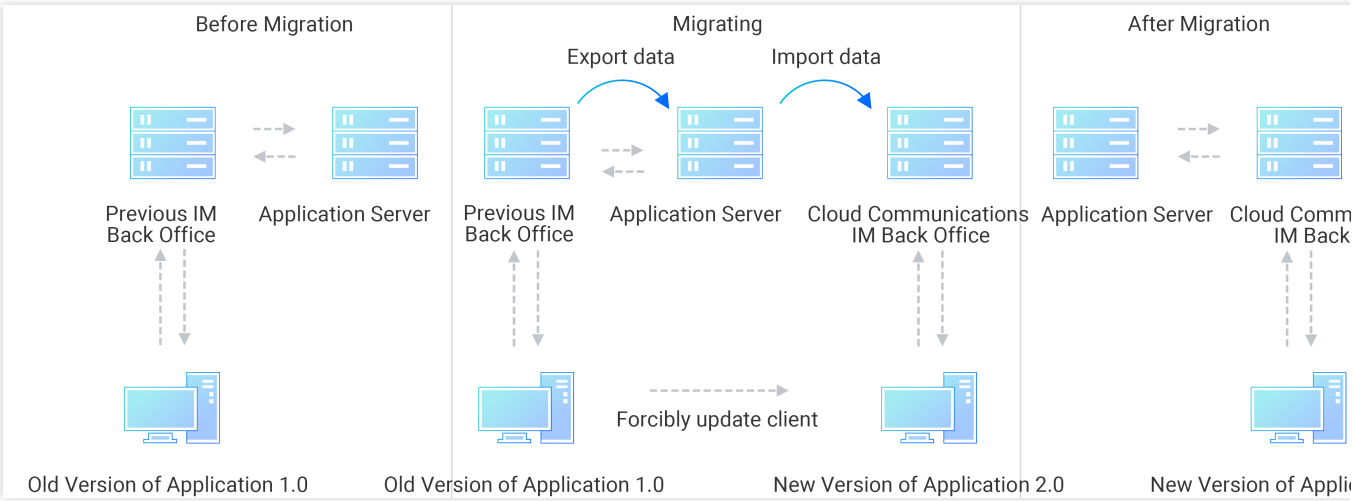
In subsequent documents, we will use the following terminology:
Old system: the original instant messaging service used by an app
New system: Tencent Cloud IM service
App 1.0: an app that implements the instant messaging feature based on the old system
App 2.0: an app that implements the instant messaging feature based on the new system
Message routing (message callback) service: after receiving a message, a third-party communication service provider forwards the message to the app backend. This is similar to the callback after sending one-to-one messages of IM.
The migration process is essential to switch the instant messaging service backend from the old system to the new system and upgrade App 1.0 to App 2.0.

Migration Solutions

IM provides the following two migration solutions for you to choose from. Different solutions have different migration effects and differ sharply in implementation difficulty. You need to take the overall existing instant messaging situation of your app into account when choosing a proper migration solution.

Mandatory upgrade solution

The mandatory upgrade policy forcibly upgrades App 1.0 to App 2.0 after IM data synchronization is completed. This solution is easy to implement, and compatibility between the new and old apps after the upgrade does not need to be addressed. The following figure shows the detailed solution:



The main process is as follows:

1. Import historical data to IM, including:

Accounts

User profiles

User relationship chains

Message history of one-to-one chats

Group data

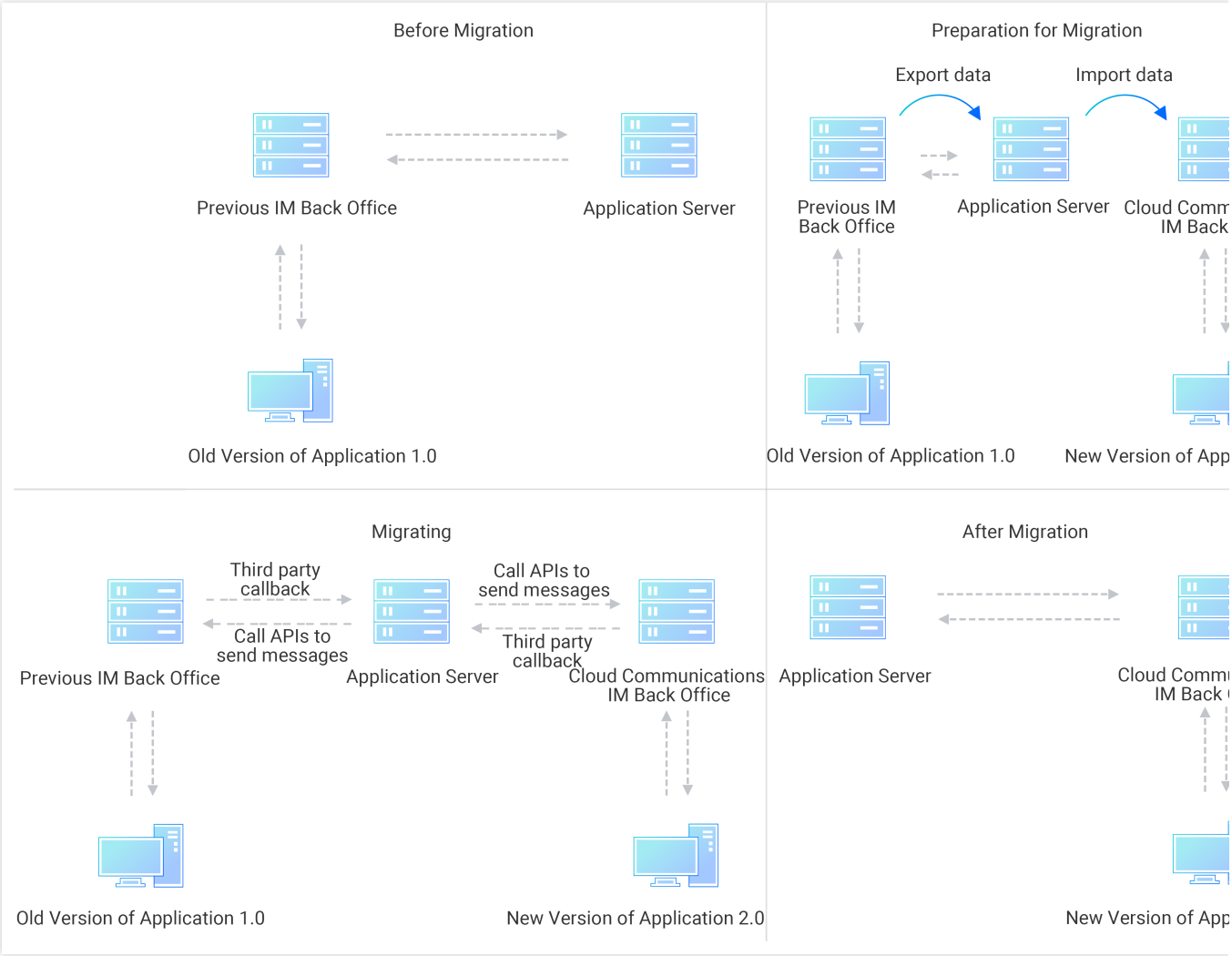
Message history of group chats

2. Force users to upgrade App 1.0 to App 2.0.

3. The old system is discontinued, and all user communication activities are processed in the new system.

New and old app compatibility solution

In this solution, the new and old apps can coexist, and messages are synchronized between both apps. Before App 1.0 is disabled, the app backend needs to maintain real-time two-way synchronization between the new and old systems. This solution is relatively complicated, but provides a better experience for end users. The following figure shows the detailed solution:



The main process is as follows:

1. Import historical data to IM, including:

Accounts

User profiles

User relationship chains

Message history of one-to-one chats

Group data

Message history of group chats

2. Synchronize app data between the new and old systems in a two-way manner, including:

Synchronize one-to-one messages in real time

Synchronize group data and group messages in real time

3. The new and old systems coexist, messages are synchronized between both systems, and the old app will be naturally phased out.

Detailed Migration Operations

Importing accounts

Importing accounts is the prerequisite for subsequent imports of various data.
The app backend needs to call the RESTful API for batch account imports to import all existing accounts into IM. If you need to import user nicknames and profile photos while importing accounts, call the RESTful API for importing a single account.

Importing user profiles

Call the RESTful API for setting profiles to import existing user profiles into IM.

Importing user relationship chains

Call the RESTful API for importing friends to import existing relationship chains into IM.

Importing message history of one-to-one chats

Call the RESTful API for importing one-to-one messages to import existing one-to-one messages into IM.

Marking one-to-one messages as read

Call the RESTful API for marking one-to-one messages as read to mark one-to-one messages as read.

Importing group data and message history of group chats

Follow the directions below to import the group data and message history of group chats:
1. Call the RESTful API for importing basic group profiles to create a group, and you can specify the initial group members while calling the API.
2. If you did not import group members while importing a group, you can call the RESTful API for importing group members to import group members.
3. Call the RESTful API for importing group messages to import the group chat message history.
4. To correct the unread message count for group members, you can call the RESTful API for setting the unread message count for members to perform the relevant operations.
One-to-one messages, group data, and group messages all need to be hosted to the new system. When new data of these types is generated in the new system, use the callbacks provided by IM to synchronize the new data to the old system. Meanwhile, new data generated in the old system also needs to be synchronized to the new system.

Synchronizing one-to-one messages

When new messages are generated in the old system, synchronize them to IM by calling the RESTful API for sending a single one-to-one message. When new messages are generated in IM, synchronize them to the old system by calling the callback after delivering one-to-one messages.

Synchronizing group data and group messages

Synchronizing group profiles

1. When basic group profiles in the old system change, call the RESTful API for modifying basic group profiles to synchronize the changes in real time.
2. When basic group profiles in IM change, call the callback after creating a group, callback after disbanding a group, and callback after modifying group profiles to synchronize the changes to the old system.

Synchronizing group member information

1. When group members are added or deleted in the old system, call the RESTful API for adding group members and RESTful API for deleting group members to synchronize the changes to IM.
2. When users join or leave a group in IM, call the callback after a user joins a group and callback after a user quits a group to synchronize the changes to the old system.

Synchronizing group messages

1. New group messages in the old system need to be synchronized to IM by calling the RESTful API for sending ordinary messages in groups.
2. New group messages in IM need to be synchronized to the old system by calling the callback after delivering group messages.

Caution

If the solutions provided here do not apply to the existing instant messaging service of your app, please contact a customer service agent or business manager to formulate an appropriate migration solution.

Migration Solutions Lite

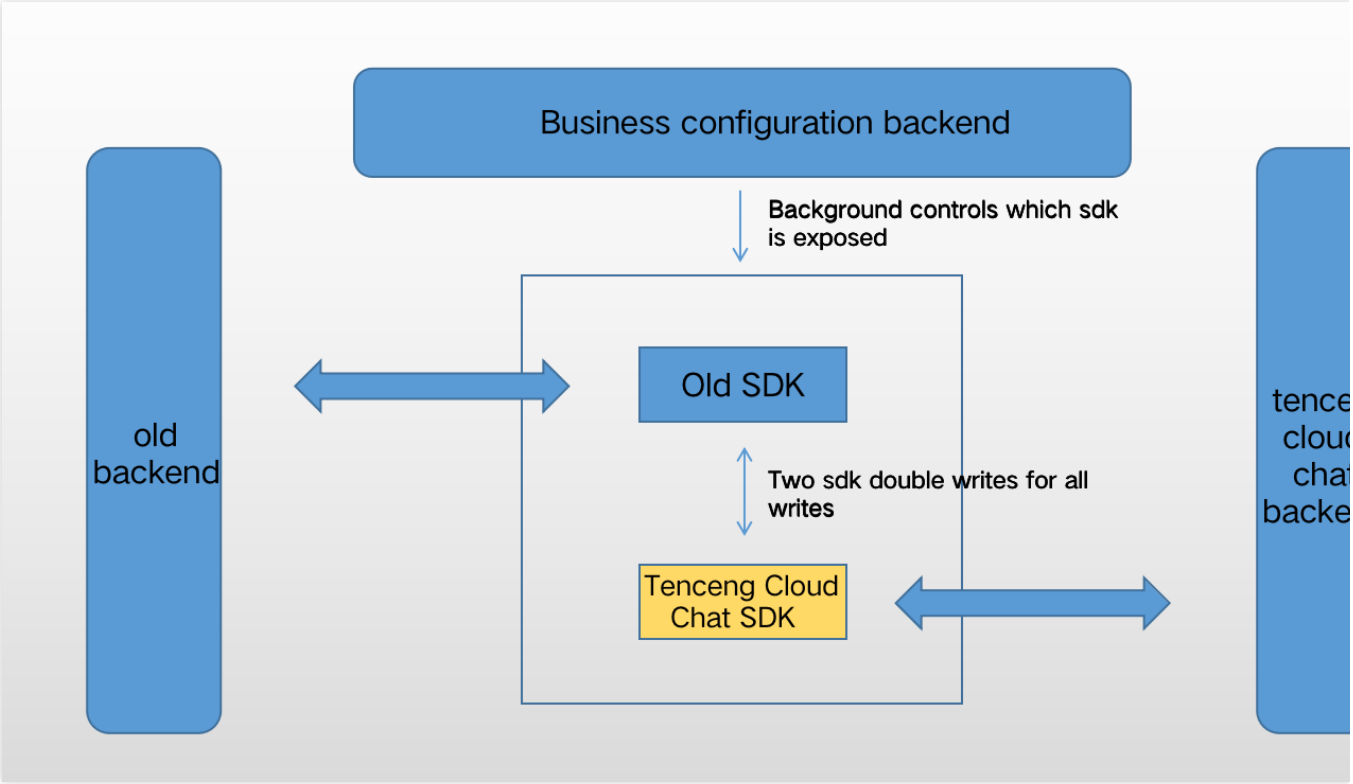
Last updated：2024-02-07 17:33:32

If App developers who are using self-developed or third-party instant messaging services want to access instant messaging IM, they need to consider migration. IM proposes a simplified version of the migration solution based on different scenarios. In this way, the migration workload is only slightly more than the newly accessed Tencent Cloud IMSDK. Compared with the full version of the smooth migration, the workload is greatly reduced.

Precondition

Historical messages from 1-3 months ago can be abandoned

Migration Plan



The main process is as follows:

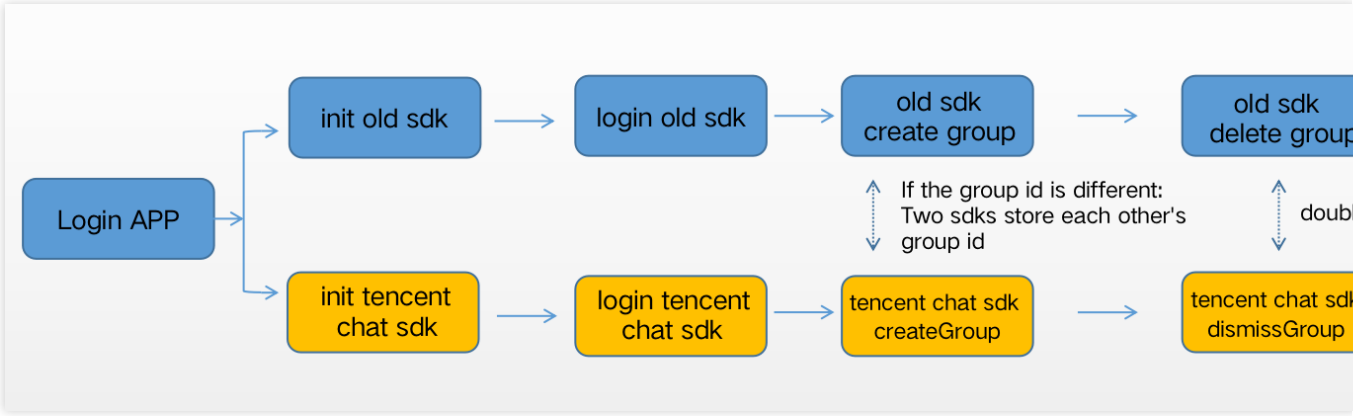
1. Simultaneously integrate Tencent Cloud chat SDK and old SDK, and both initialization and login SDK must be done

2. Double write at the SDK level for all IM related write operations
3. Default external display (actual display) of old SDK
4. Control which SDK is displayed on the terminal side through backend configuration and grayscale scaling
5. Double write in parallel for a period of time, after observing that there are no issues, switch all to the new Tencent Cloud SDK

Example

Scenario 1

After creating a group, dismiss it. Taking Android interface as an example:



Scenario 2

Send a group chat message and revoke it