

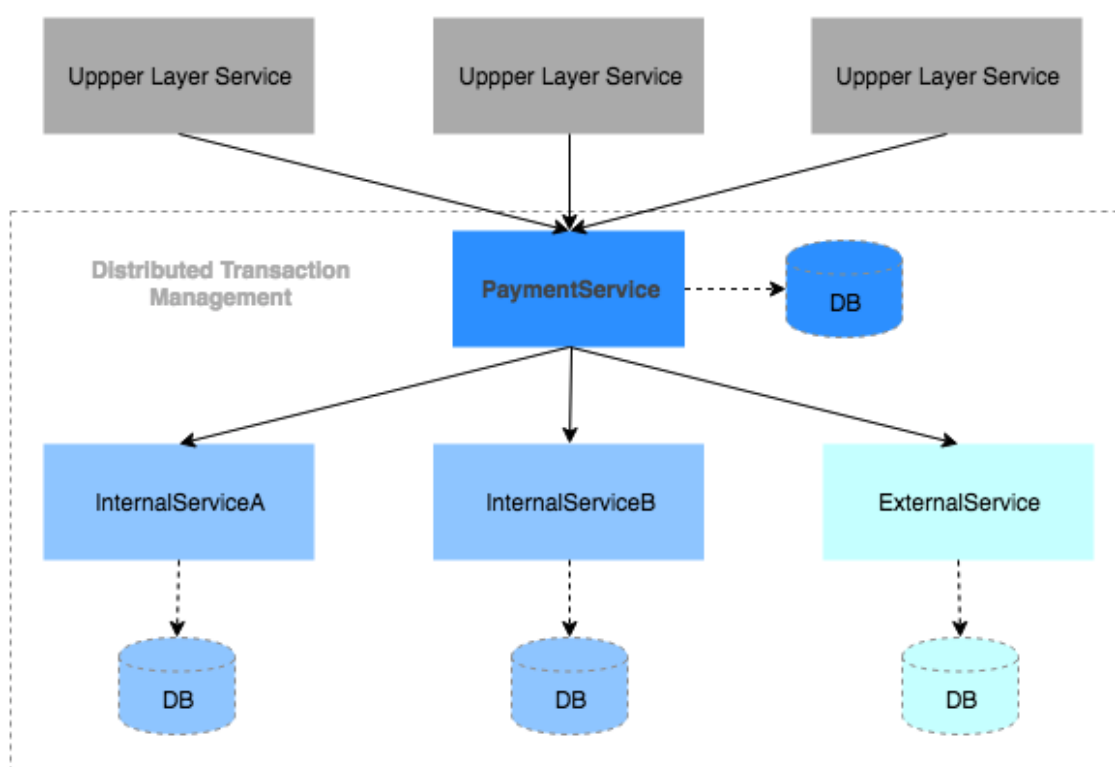
# MERIT Internship Report (Domestic company)

Department of Advanced Materials Science  
Lingyu Feng, D2

Company: Mercari Inc.

Team: Merpay payment platform team

Date: 2021.08.01 ~ 2021.09.31



Merpay develops payment systems with a microservice architecture. Among those microservices, PaymentService serves as the infrastructure service for payment transaction management, using various payment methods provided by lower-layer services (including external services) to provide the necessary payment flow to upper layer services (Mercari, NFC, code payment, etc.) as a common API. The above figure shows the structure of distributed transaction management. PaymentService requires accurate management of money movement among multiple services, payment transaction management is important and a system that ensures data consistency across multiple services was built.

Subdivision of transaction processing is one of the main methods used to achieve data consistency. When an error occurs during payment processing, the handling to be taken depends on the type of error that occurs and how far the processing has

gone. If the problem can be fixed by retrying, we only need to perform the necessary retry processing. If the problem cannot be fixed immediately, we can retry later in the batch. On the other hand, if the problem cannot be fixed by retrying, we perform the necessary rollback process including the dependent services and then terminate the process. In order to determine the necessary retry process and the necessary rollback process, we first need to know how far the process has gone. For example, if a process is executed as one large transaction, and it fails in the middle, it will be difficult to recover from the failure because we will not know where the failure occurred.

For this reason, PaymentService subdivides a single payment transaction process into multiple phases for execution. When a payment transaction is accepted, the internal transaction data and ID must be determined as one phase before processing. The progress of each phase is always recorded in each step. The granularity of the phase is determined by the ease of retry and rollback processing. The granularity of the process makes it easier to recover from a failure during the process by referring to the currently recorded phase and narrowing down the scope of retry or rollback process to be taken.

During this internship, I was working on creating automatic retrying and rollback mechanisms for handling errors from lower-layer services. In Merpay, each microservice in the backend is implemented using Go language. gRPC is used for communication between microservices, and a status code is returned to notify of an error ([https://grpc.github.io/grpc/core/md\\_doc\\_statuscodes.html](https://grpc.github.io/grpc/core/md_doc_statuscodes.html)). For each status code, we handle the error by different processes. For example, when receiving an ABORTED error, we perform the retrying process. As a result, these automatic retrying and rollback mechanisms are able to handle urgency problems and recover the transaction without manual handling.

I would like to deeply thank my team manager @rui for giving me this opportunity to have this internship. Thank my mentor @knsh14 for helping me with my onboarding and daily discussion. Also thank @abcdefuji, @ind, @po-an, and all the other team members in Merpay payment platform team for discussing with me and helping me a lot.