



# TRANSACTION PROCESSING FLOW



Introduction.....	2
UniBroker Cache Usage .....	3
Real-time Diagrams: .....	4
Batch Diagrams.....	4
Real-time Processing Data Flow .....	5
Real-time Processing Data Flow (with HSM).....	6
Real-time Tokenization Data Flow .....	7
Real-time Tokenization Data Flow (through external tokenization service).....	8
Real-time Detokenization Data Flow.....	9
Batch Processing Data Flow (with tokens) .....	10
Batch Processing Data Flow (with card data) .....	11
Batch Processing Data Flow (with card data, BINs).....	12
Batch Tokenization Data Flow .....	13
Batch Tokenization Data Flow (with BINs) .....	14

## Introduction

This document describes how transaction processing is done depending on the configuration of UniPay. To learn more about UniPay configurations, review UniPay Deployment Configurations document.

The UniPay system supports real-time and batch transaction processing. Each of these processes has an individual life cycle that, along with configurations, affects how transactions are processed.

In addition to UniPay configurations, there are several factors that influence transaction processing:

- **Whether card data or token is used in a transaction submitted to the gateway.** When sensitive information is present in incoming data, an additional tokenization process is performed before the data is submitted to the gateway.
- **How tokenization is configured.** Tokenization can be arranged in the following ways, influencing transaction processing on a more precise level:
  - Tokenization is done by a *tokenization appliance (StrongAuth)*. This is the most common tokenization option in UniPay. Sensitive data is tokenized by StrongAuth right after it reaches UniBroker.
  - Tokenization is done by an *external service provider*. Sensitive data is stored in the UniBroker cache while data is exchanged between UniBroker and UniPay. To learn more about UniBroker cache usage, review the additional section in the document.
- **Whether detokenization is required or not.** Usually, detokenization is done through the API call. For real-time processing, detokenization can be done through the user interface when it is necessary to use card data in a system that is not integrated with UniPay. For example, it may be necessary for administrators of a hotel that processes transactions through UniPay to make a purchase for their guest (i.e. using his/her card data) in a store that has no integration with UniPay gateway.
- **Whether transaction processing involves BIN verification.** If BIN verification is done, transactions go through an additional verification phase before the actual processing.

## UniBroker Cache Usage

Tokenization can be done in two different ways – by local connection to StrongAuth or using another tokenization service.

If StrongAuth appliance is used, a credit card number is tokenized immediately as it enters UniBroker.


If a different tokenization appliance is used, UniBroker needs to form a request for tokenization to this external service in a corresponding format. UniPay is responsible for generating of this message. However, UniBroker is responsible for the submission of this message to the actual tokenization service because UniBroker has access to the card number.

Therefore, when the transaction with credit card information enters UniBroker, it generates temporary placeholder and forwards the proxied information to UniPay. When UniPay prepares the message and sends in back to UniBroker, UniBroker replaces the placeholder with an actual card number and addresses the request to the tokenization appliance. Usually, it takes a couple of seconds for the message to be generated in UniPay, and during that time the actual card number is stored in UniBroker cache. When the message is returned to UniBroker and credit card number is inserted in this message, a card number gets removed from cache. If any error occurs during message generation in UniPay and an exception is returned, the card number gets deleted from UniBroker cache after cache storage period expires.

Credit card data is stored in UniBroker cache for the time needed for UniPay to generate a message in the format of tokenization service. Usually, this period is less than 10 seconds. The period of sensitive card data storage in cache can be set individually in ***unipay.unibroker.account-data-caching-period*** property. The minimum value for the field is 240 seconds. If the field is empty, cache storage period is set as minimum value by default.

In case when any processing issue occurs, the card data is stored in cache for 10 minutes and after that, it is completely removed from cache automatically.





In the diagrams below, you can review how all the factors described above influence the transaction data flow. In most diagrams, tokenization process is done via a tokenization appliance, such as StrongAuth. However, it can be done in two ways depending on the merchant configuration:

- via a tokenization appliance directly through UniBroker. In this scenario, no integration with UniPay is required; the most common tokenization appliance is StrongAuth;
- via a processor, which supports tokenization, through UniPay. In this scenario, the integration with UniPay is required; multiple tokenization providers can be used. Usually, tokenization is done through the same processor that is used for transaction processing.

Diagrams are divided into two sections – real-time and batch diagrams.

### Real-time Diagrams:

1. **Real-time Processing Data Flow** – shows the flow of real-time transactions;
2. **Real-time Processing Data Flow (with HSM)** – shows the flow of real-time transactions using an HSM device;
3. **Real-time Tokenization Data Flow** – shows tokenization of data in real-time transactions;
4. **Real-time Tokenization Data Flow (through external tokenization service)** - shows tokenization of data in real-time transactions through the external tokenization service;
5. **Real-time Detokenization Data Flow** – shows detokenization of data in real-time transactions.

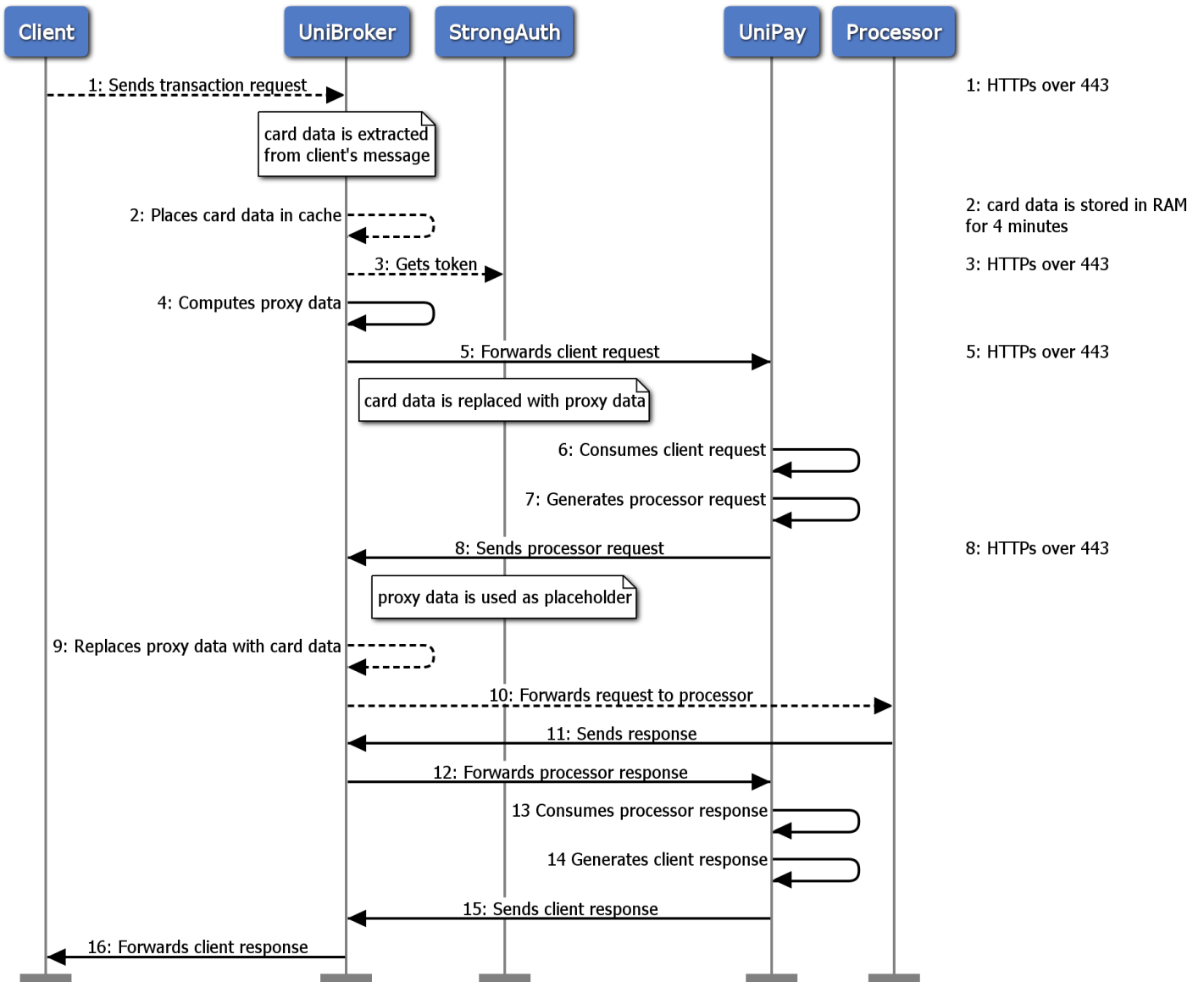
### Batch Diagrams:

1. **Batch Processing Data Flow (with tokens)** – shows the flow of batch transactions with sensitive data represented as tokens;
2. **Batch Processing Data Flow (with card data)** – shows the flow of batch transaction with non-tokenized sensitive data;
3. **Batch Processing Data Flow (with card data, BINs)** – shows the flow of batch transactions with non-tokenized sensitive data inside, including BIN identification;
4. **Batch Tokenization Data Flow** – shows tokenization of data in batch transactions;
5. **Batch Tokenization Data Flow (with BINs)** – shows tokenization of data in batch transactions, including BIN identification.

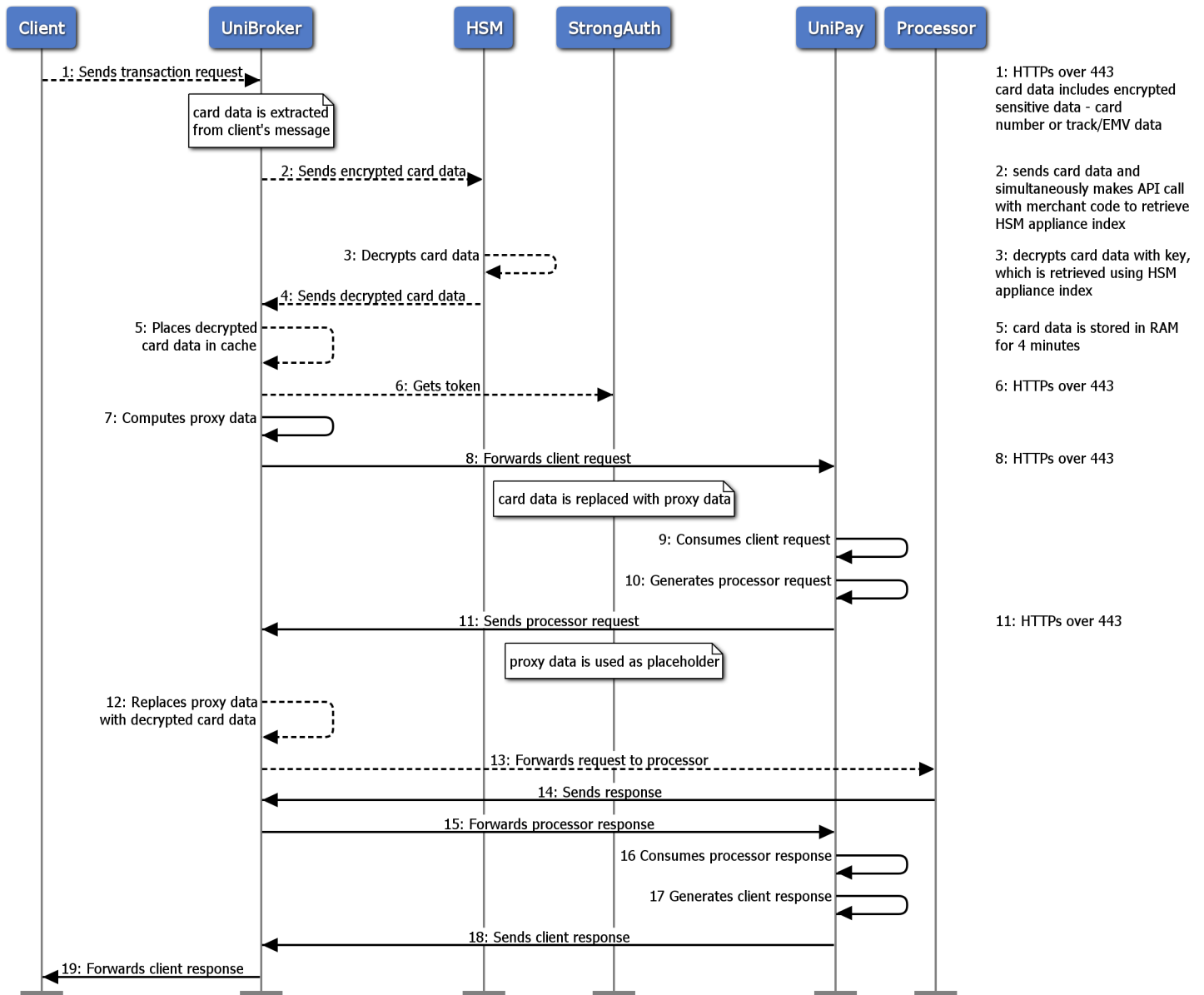
*Also, there are several notation keys in the diagrams that will be useful for you:*

- *The sequence of the transaction flow phases is marked by digits.*
- *Useful notes for the steps of the transaction flow are located in the right section of each diagram.*
- *Dashed lines of flow indicate the presence of sensitive data. Solid lines indicate the presence of encrypted/tokenized data.*

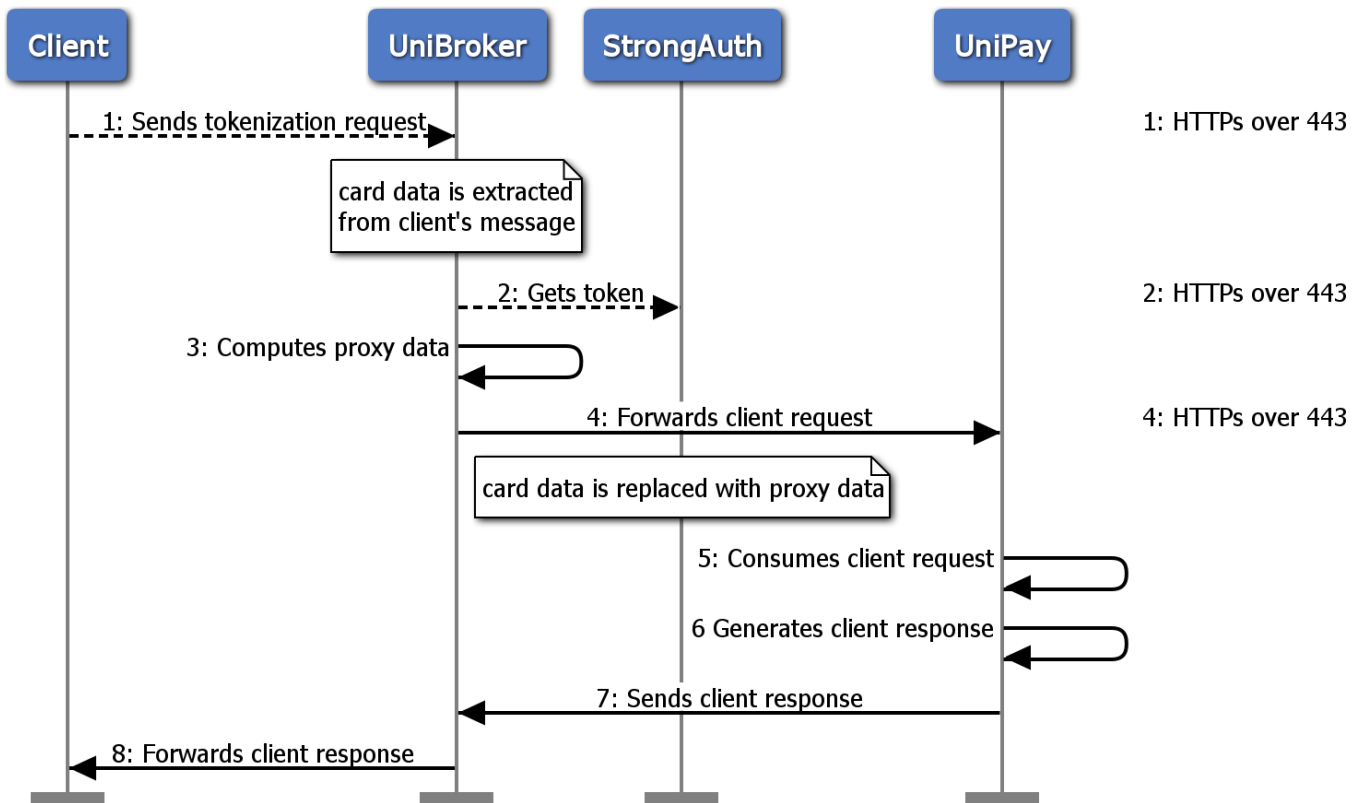
## Real-time Processing Data Flow



## Real-time Processing Data Flow (with HSM)

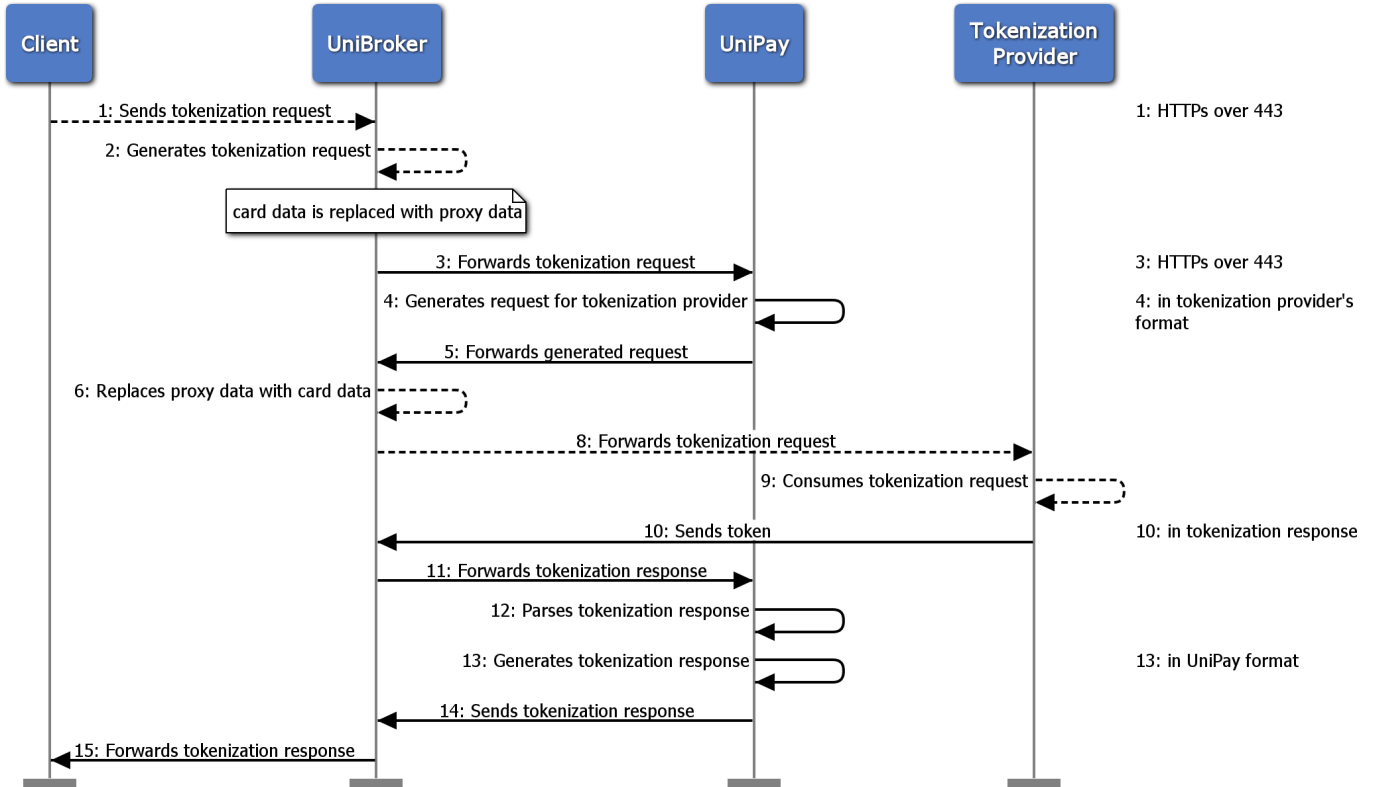


## Real-time Tokenization Data Flow

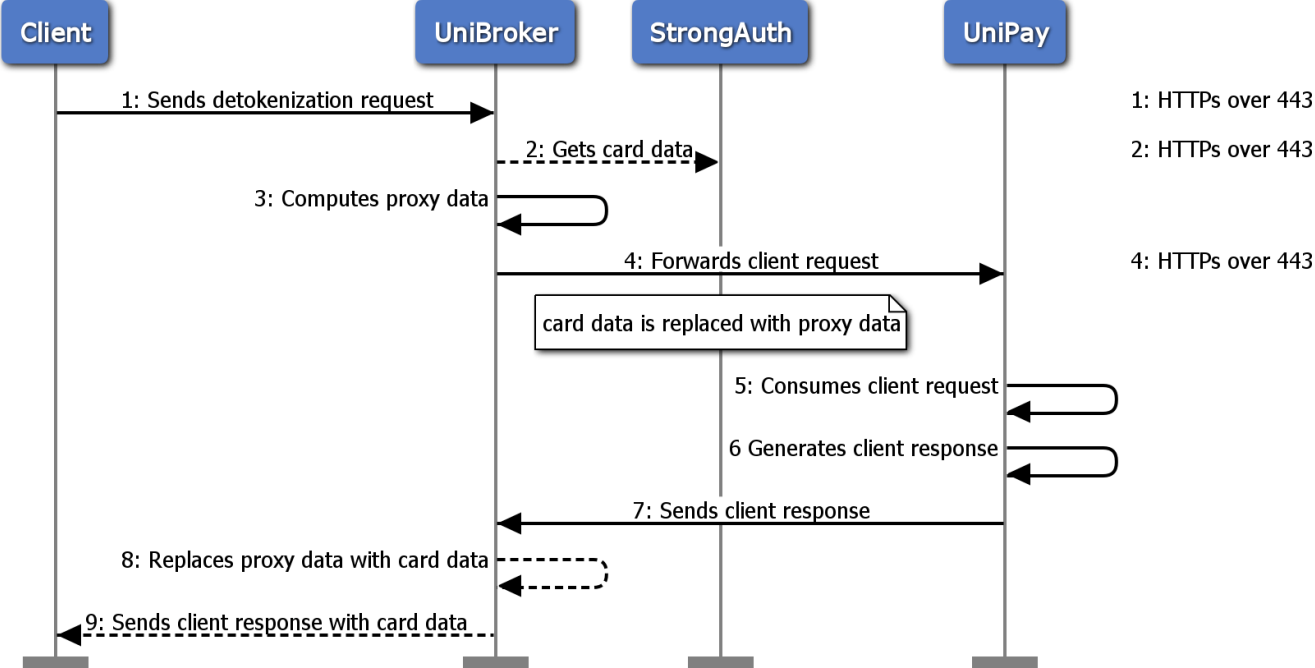




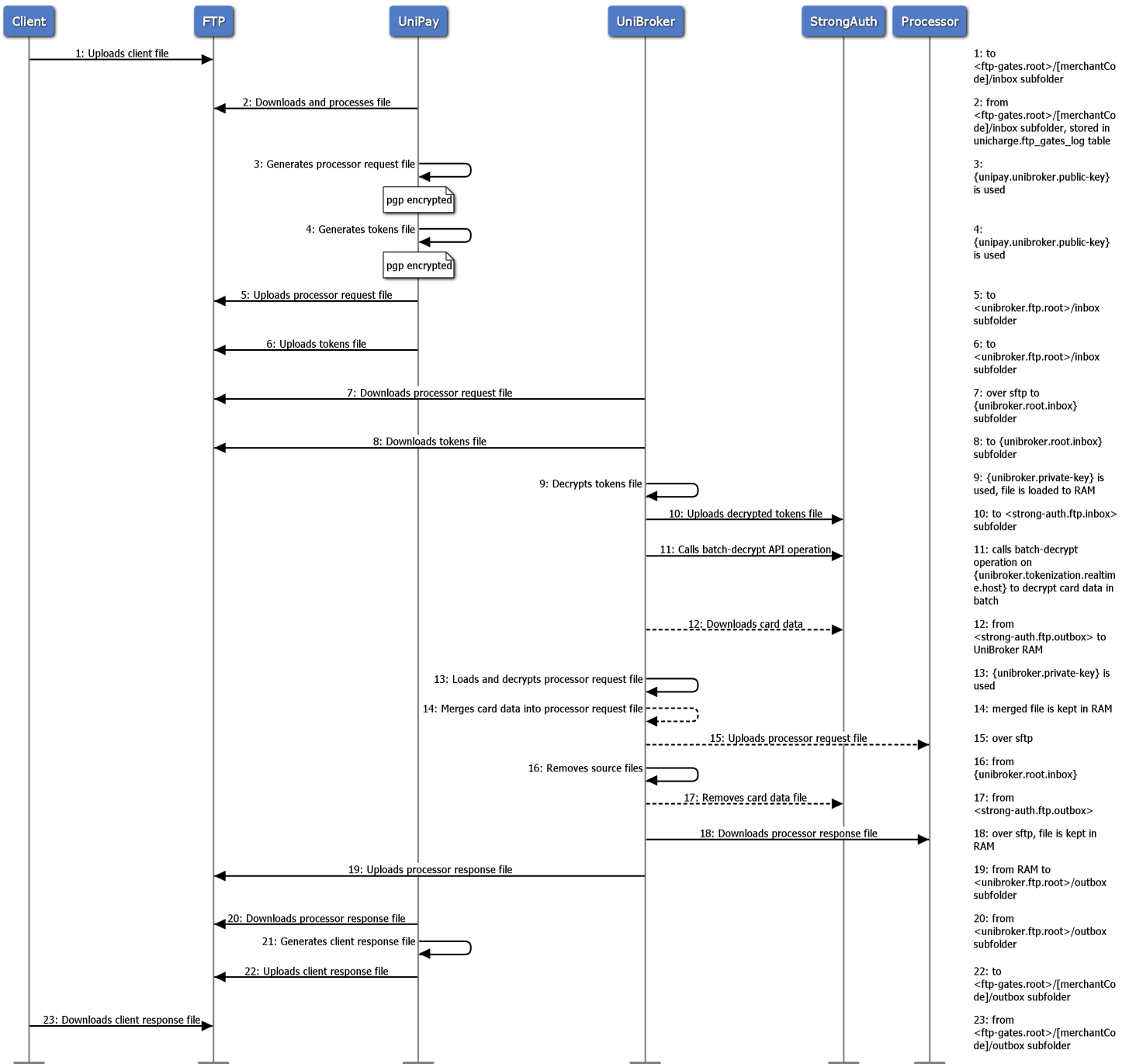
## Real-time Tokenization Data Flow (through external tokenization service)



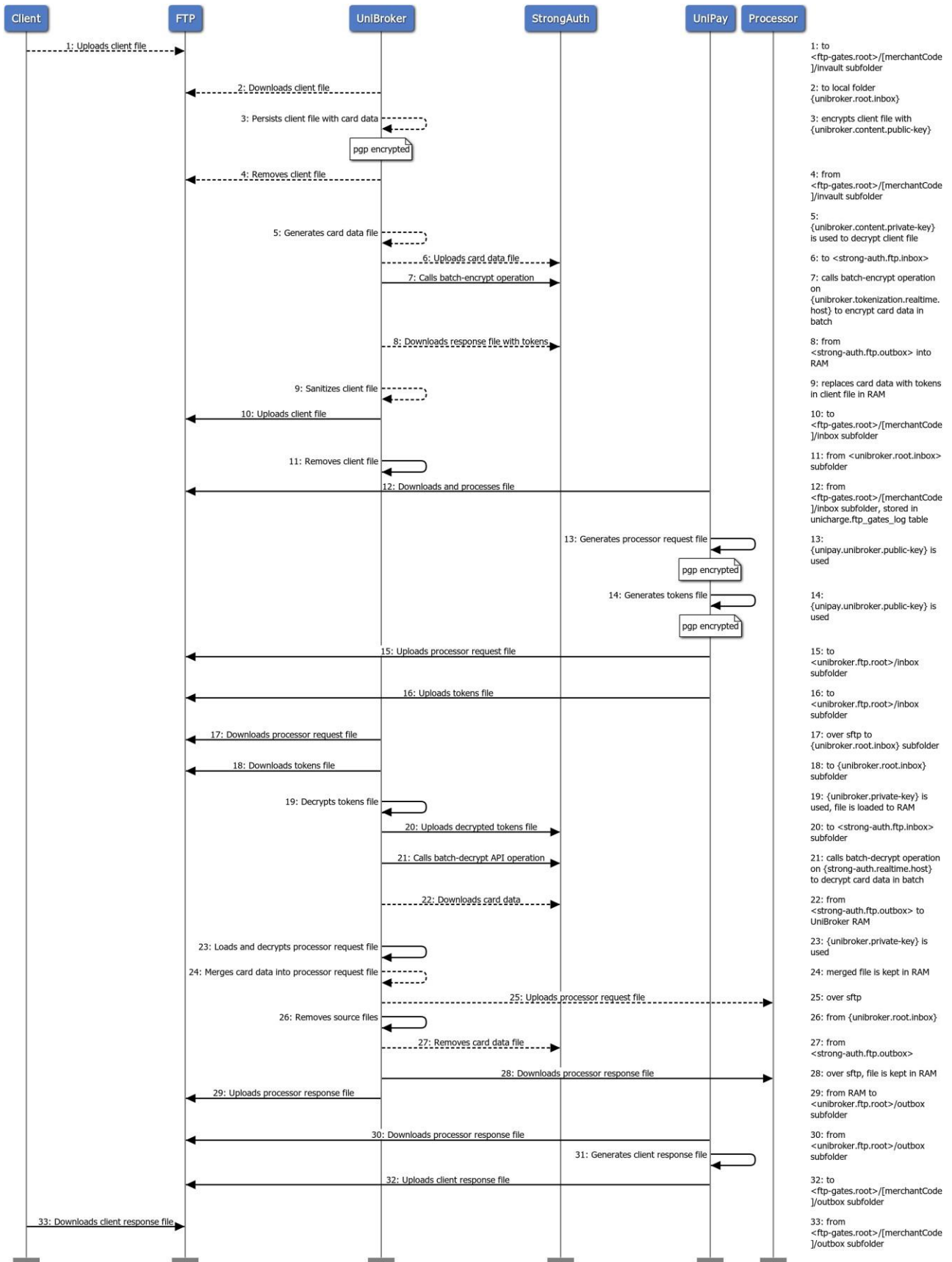
# Real-time Detokenization Data Flow



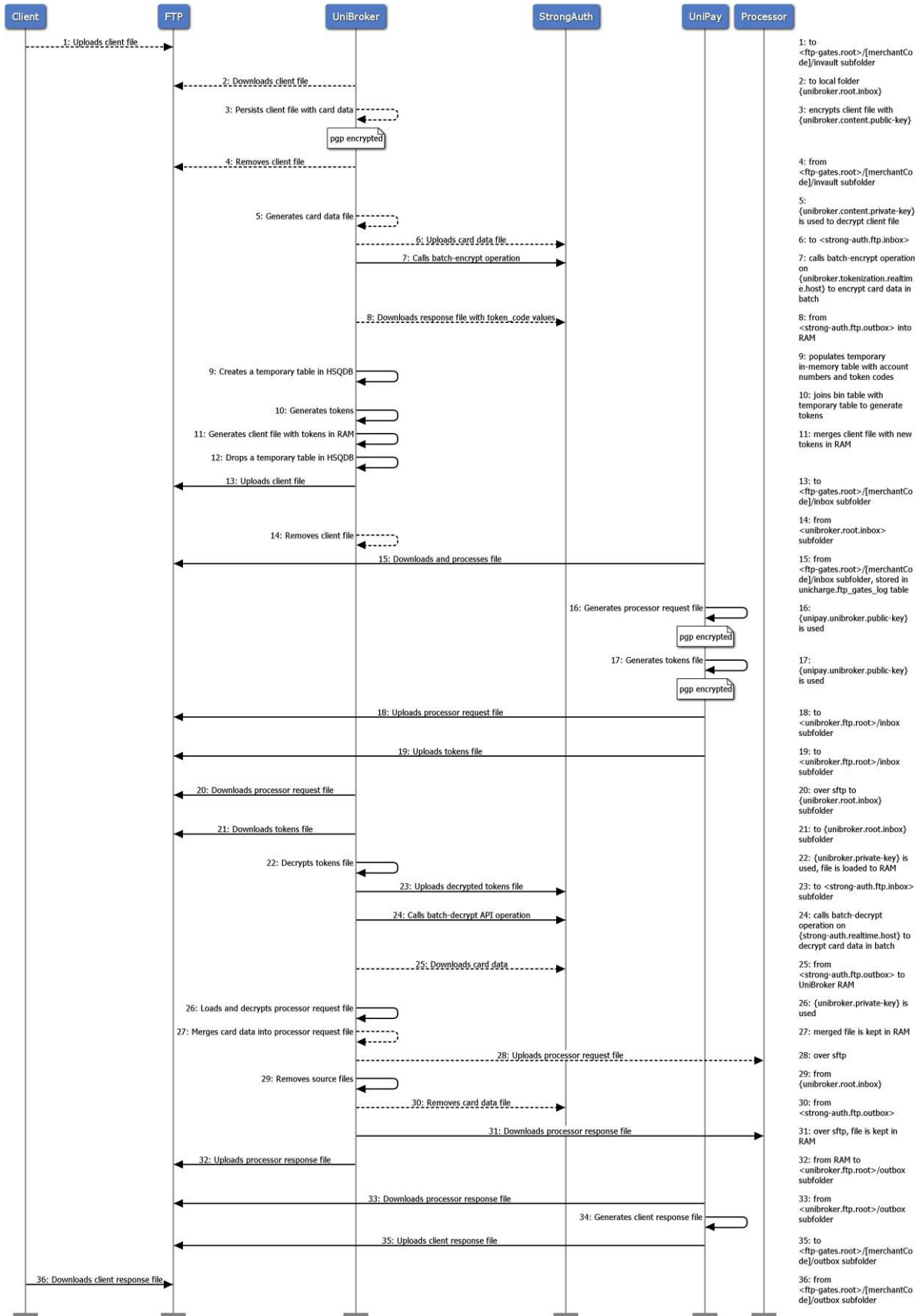
## Batch Processing Data Flow (with tokens)



# Batch Processing Data Flow (with card data)

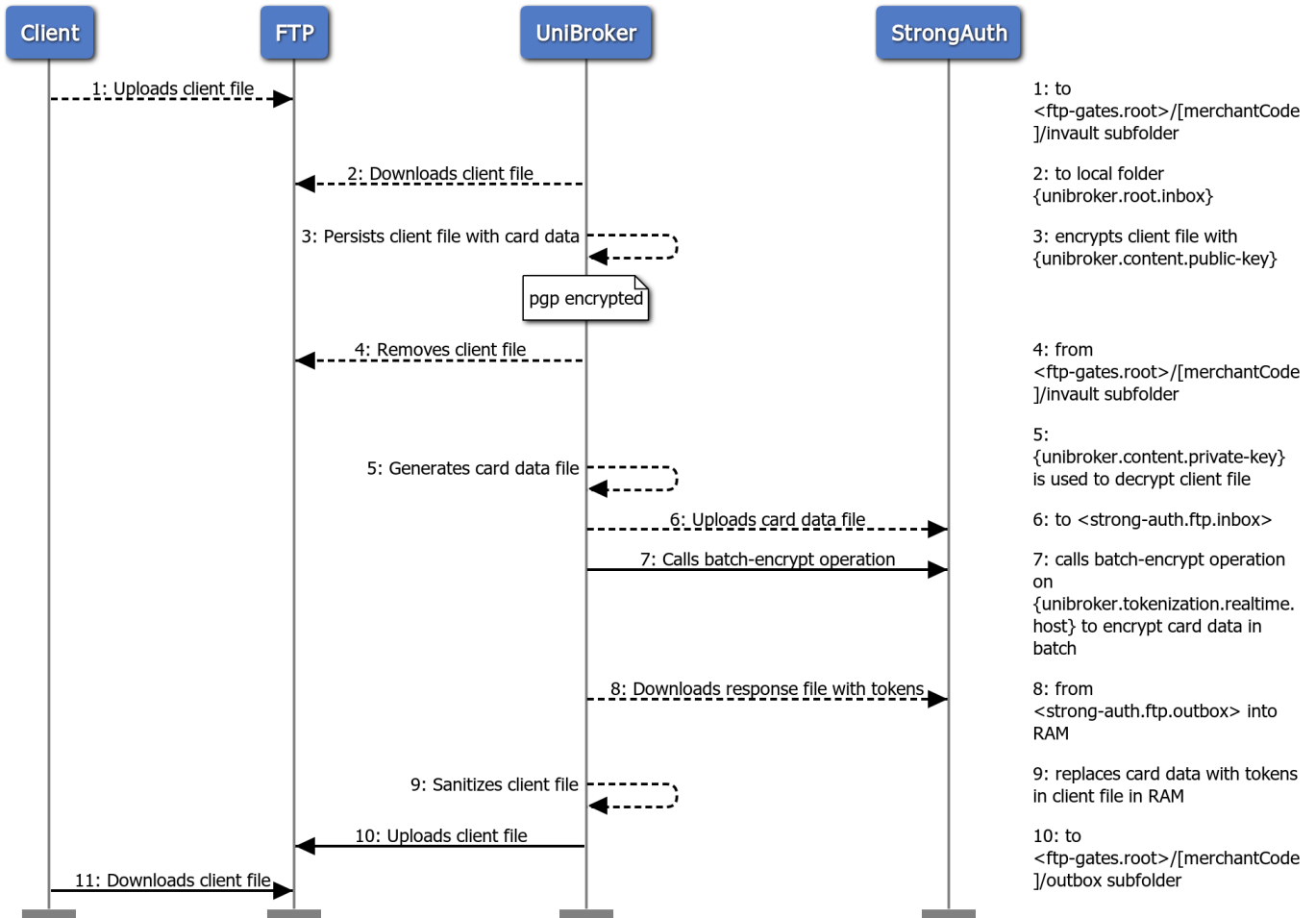


# Batch Processing Data Flow (with card data, BINs)





## Batch Tokenization Data Flow



# Batch Tokenization Data Flow (with BINs)

