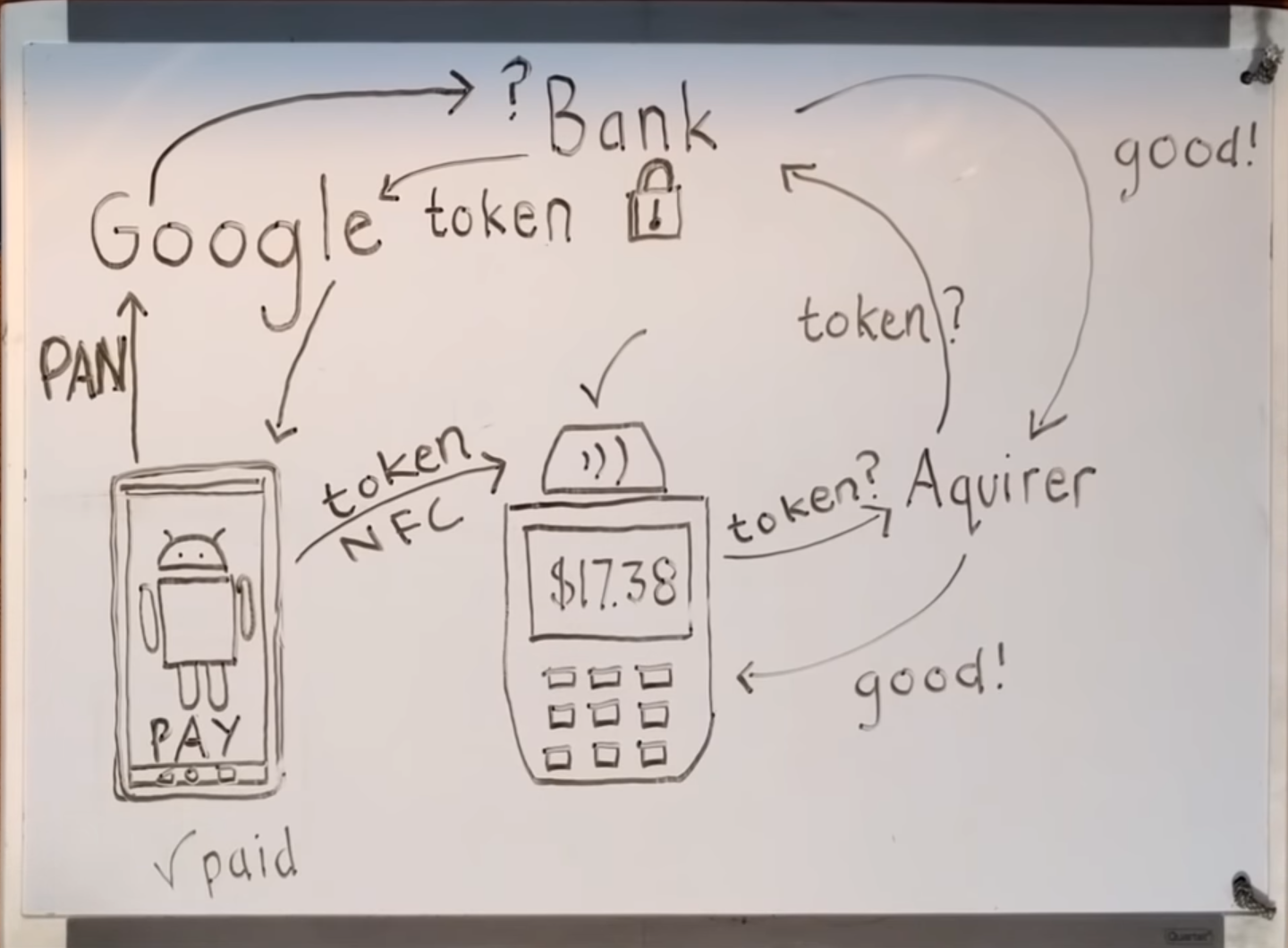
# **Credit Card Tokenization**

## Introduction

This project proposes the development of a credit card tokenization system and simulates the cycle of paying with a phone. The system will consist of three programs: End-User Program (e.g. Samsung Pay), Tokenization System (e.g. Bank or Credit Card Provider) and Merchant

## Cycle of Tokenization-Based Transaction



1. The user adds their credit card credentials, such as their credit card number and expiration date, to the payment app.
2. When a user wants to pay a merchant. The payment app sends a request to the bank with the data encrypted plus data about the merchant.
3. The bank receives the encrypted data and decrypts it. It then applies a tokenization scheme (Hash function in our case) to the data to generate a token, which is a unique identifier that replaces the user's credit card number. This token can be used only with a specific merchant.
4. The bank sends the token back to the payment app, also encrypted.
5. The payment app sends the token to the merchant. The merchant does not receive the user's credit card credentials; it only receives the token which is one way so he can’t get the actual data from the token.
6. The merchant sends the token to the bank to verify that it is valid (thus the merchant who is allowed to use the token). If the token is valid the transaction is completed.
7. When a transaction is completed the bank deactivates this token, so the merchant can’t use it again and when making a new transaction the payment app sends a request to the bank to reactivate the token.

In this way, the merchant never receives the user's actual credit card data, which helps to protect the user's privacy and security.

### Additional details:

* Tokenization: Tokenization is a process of replacing sensitive data with a non-sensitive substitute, called a token. This helps to protect the sensitive data from unauthorised access and misuse.
* Token vault: A token vault is a secure database that stores tokens. The tokens are encrypted and can only be accessed by authorised users.
* The token we get from the bank when using a tokenization system for paying with a credit card will typically be available for a limited amount of time. This is to help protect your security by preventing hackers from using the token to make unauthorised purchases. The specific length of time that the token will be available will vary depending on the merchant and the payment processor.??
* In some cases, you may get a new token every time you pay. This is more common for online payments, where the merchant may not want to store your token for a long period of time. However, it is also possible to get a token that is valid for a longer period of time, such as 30 days or 60 days.

The system is also expected to improve the efficiency of credit card transactions. By eliminating the need to store and process credit card numbers, the system will reduce the risk of fraud and improve the speed of transactions.

# **Expected Input**

* Credit card credentials of the user is an input to the payment app.
* When paying the payment app gets data about the merchant to be sent with credential to the bank

# **Expected Output**

## Successful Transaction:

Product was bought by the user app token.

Successful Transaction implies that the amount of money in the user's bank decreases by the product price.

## Failed Transaction:

Error status code will be given for every error:

* The user money isn’t enough to buy the product (Error code : 1)
* Another user (application) is using the token to buy something (Error code :2)
* The merchant tried to use the same token twice without activating it again by the user (Error code: 3)

## Successful Tokenization:

The process between the bank and payment app was successfully done, and a token was registered to this application.

## Failed Tokenization:

App tried to ask the bank for tokenizing a card that isn't present in the bank.

# **Assumptions**

1. The bank has a secure database for Credit card credentials, so the tokenization is done for already present (existent) credit cards, that’s the reason unsuccessful tokenization may occur.
2. During tokenization, the end-user (e.g., Samsung Pay) communicates with the bank through an encrypted mechanism with authentication to be secure from man in the middle attacks. This encryption can be done through certificates or public key authorities. In the worst-case scenario, we assume that this is a secure layer and communicate with encryption and simple authentication as we assume man in the middle attack is unlikely to happen.
3. All communication can be simulated in a local machine.
4. We will use the SHA-512 hash function, which is a cryptographically secure hash function. Collisions in SHA-512 are very rare, so we will assume that they are impossible to happen.