# JUZU

**Protocol WhitePaper v1.0**

info@juzu.app

February 2022

**Abstract**

This document describes the definitions and theory behind the Juzu Protocol explaining the different aspects of the implementation.

**Contents**


**1. Introduction** **3**
1.1 Basic Concepts 5
1.2 Formal Definitions 6

**2. Protocol Architecture** **7**
2.1 The Factory Contract 8
2.1.1 Create Locker 8
2.1.2 Reward Minting Right 10
2.1.3 Configurations 10
2.2 The JuzuLocker Contract 11
2.2.1 Update Locker 11
2.2.2 Unlocking Locker 12
2.2.3 Staking & Earning Reward 14
2.2.4 Burn NFT 14
2.3 The JuzuERC721 Contract 15
2.1.5 Mint Right 15
2.4 The JuzuERC20 Contract 15

**3. Conclusion** **16**

# 1. Introduction

The Juzu protocol allows user to pack multiple crypto assets in a smart contract called Juzu Locker, set the conditions for unlocking and publishing NFT representation like ownership of that locker.


Figure 1: Japamala - Wikipedia

There are many use cases you can apply based on this Juzu protocol such as

- **Basket Trade**: By having 1 NFT representing all assets, you can easier trade assets that are difficult to price individually. You also can sell it or borrow money using NFT as collateral.
- **Lock Up:** You can lock up the tokens you give to initial investors for a specified period.

- **Stock Option:** DAO creates a Locker with their governance token, lock-up, and setting extra fee for the contributor, and setting conditions like stock options so that the contributor can benefit from the price increase after a lock-up period of time.
- **Stake & Earn**: Get $JUZ and add $JUZ to your own Locker, the owner will earn staking rewards.
- **Floor Price Guarantee:** By including stable coins in Juzu Locker, you can create NFTs with a guaranteed minimum price, like commemorative gold coins.

And many more cases that you can freely create.

## 1.1 Basic Concepts

Juzu Protocol allows users to create lockers, lockers can contain NFT ERC721, ERC1155, tokens, and coin native. Juzu provides a combination of different complex unlocking conditions that can be applied to many real-life cases. The types of unlocking conditions will be described in the following section.

A unique JuzuERC721 NFT will be issued to represent the owner for each Locker, the owner will have certain rights.

We will also distribute $JUZ token, if you keep this token in your Locker, you will also get rewards corresponding to the rate set by staking time.

## 1.2 Formal Definitions

| Variable | Description |
|---|---|
| T, **current times** | Current number of seconds defined by **block.timestamp** |
| $T_{year}$, **seconds** | Number of seconds in a year. $T_{year}$ = 3153600 |
| $T_{stakingPeriod}$, **seconds** | $JUZ staking time in seconds. |
| $T_{lastJuzDeposit}$, **seconds** | Latest time of $JUZ deposit in seconds. |
| $A_{stackReward}$ | Amount of reward that owner can claim. |
| $A_{juz}$ | $JUZ deposited amount. |
| $A_{lastReward}$ | Amount of reward before update staking. |
| $T_{latestStackClaimed}$ | Time of latest stack claiming in seconds. |
| $APR_{stack}$ | Current Annual Percentage Rate of staking of $JUZ token. |
| $A_{baseFee}$ | Fee Amount of $JUZ token to unlock Locker. |
| $A_{extraFee}$ | Extr Fee Amount to unlock Locker. |

## 2. Protocol Architecture

The current implementation of the protocol is as follows:

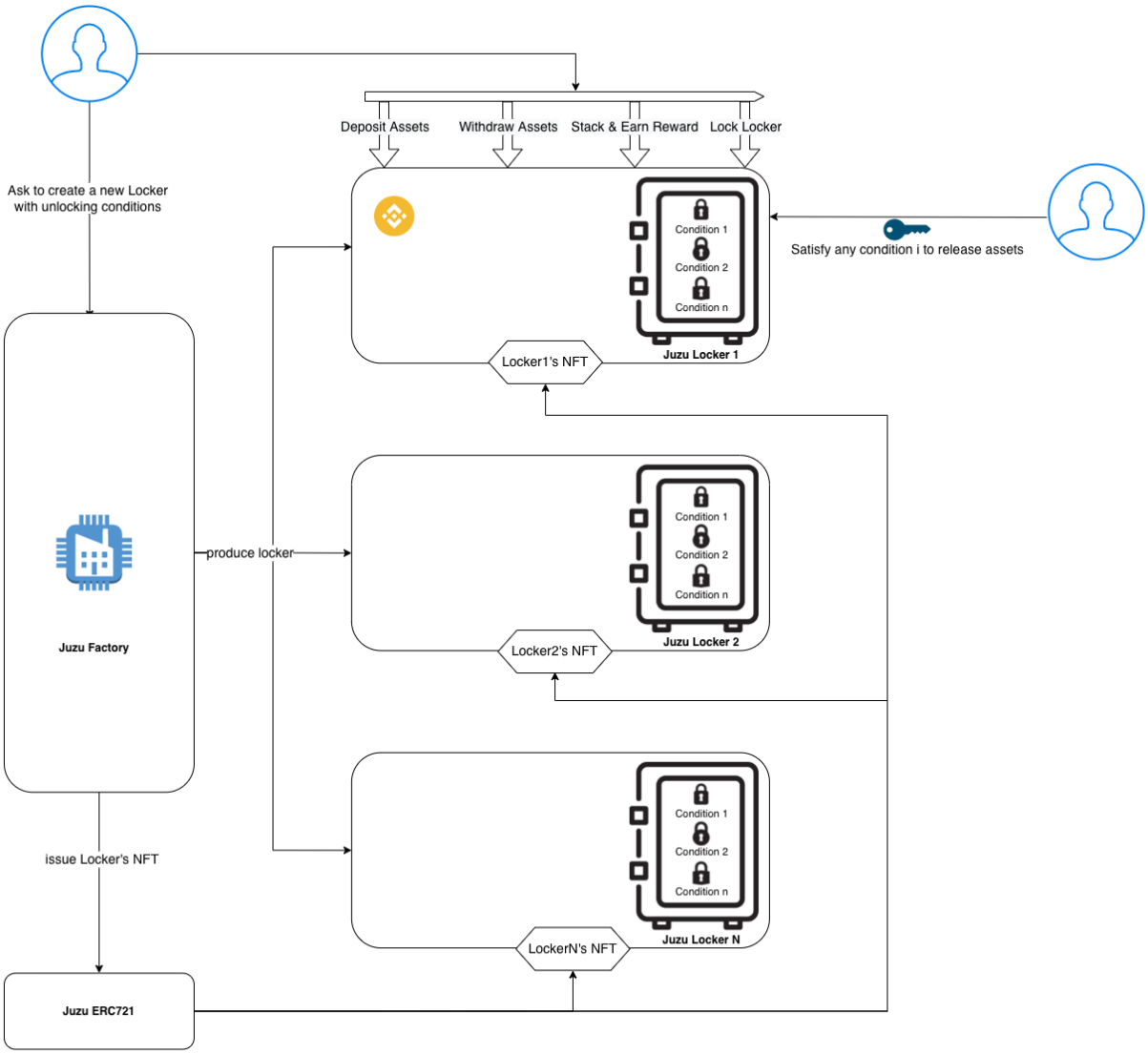


Figure 2: Protocol Architecture

The Juzu Factory contract is where:

- The lockers are created.
- Mint reward when a user claims reward.

The Juzu Locker contract is where:

- Preserve all assets of users who have deposited.
- Set up groups of unlock conditions, check to unlock requirements to release the assets.
- Issue JuzuERC721 representing each Locker

The Juzu ERC721 contract:

- Issue JuzuERC721 representing each Locker.

## 2.1 The Factory Contract

### 2.1.1 Create Locker

Before storing assets, users need to create a safe locker. You need to set up unlocking conditions. unlocking conditions is optional, it may not need to be set.

There are three main types of unlocking that the user can choose:

1. **Time Lock**: set the unlock time in the future, after that date, the user can unlock it.
2. **Extra Fee**: set any fee token amount that release-er needs to pay to a specified address to unlock Locker.
3. **Release by**: specify who has the right to unlock, we call release-er. If not set, the releaser will be set to owner by default.

In addition, an amount based fee equal to $JUZ will be default set as a based condition to unlock.

Locker can set priority groups of conditions, up to max 5 groups can be set. Each priority group has many unlock conditions. Up to 8 unlock conditions for all groups. And each unlocking condition can choose 1 to 3 types above.

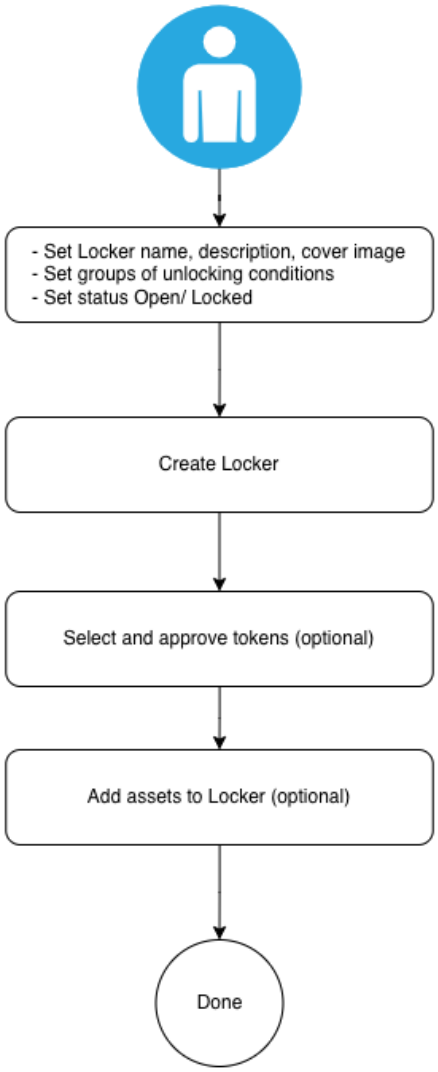


Figure 3: Create an Offer

- User can set unlocking condition or not, if left blank, default condition for unlocking is possession of NFT representative of that Locker.
- User can add assets after creating Locker.
- Locker has 2 selectable states when created, *Locked* and *Open:*
    - *Locked*: Owner cannot withdraw assets and edit unlocking conditions, can only withdraw through *release()* method if unlocking conditions are satisfied.
    - *Open:* Owner can withdraw assets and edit unlocking conditions.

### 2.1.2 Reward Minting Right

JuzuFactory is also a contract that has the right to mint $JUZ as rewards tokens. But this function can only be requested from the Locker contract.

### 2.1.3 Configurations

JuzuFactory has the right to:
- Set APR of $JUZ staking via *setApr()* function
- Set based $JUZU fee to unlock locker through *setBaseFeeAmount*()
- Proceed to stop/restore to create a new JuzuLocker on the current Factory instance through *pause()* and *unpause()*. Created Lockers will not be affected.

## 2.2 The JuzuLocker Contract

### 2.2.1 Update Locker

A Locker is in *open* state, owner can withdraw/deposit assets or update unlocking conditions but cannot withdraw assets or update conditions if in *locked* state.

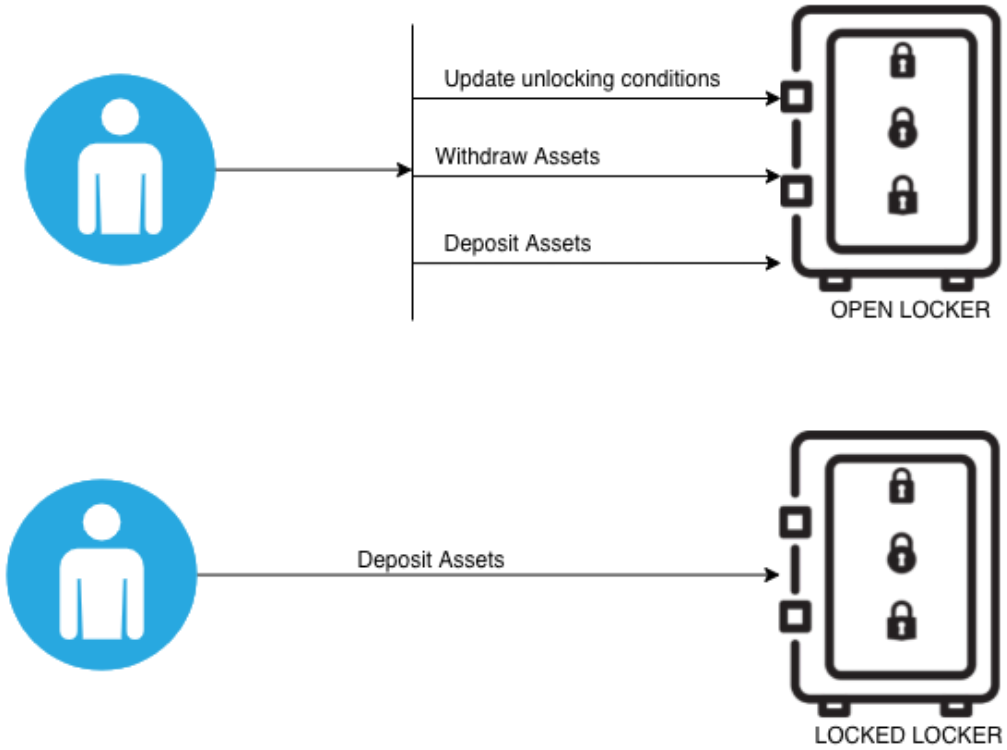


Figure 4: Update a Locker

### 2.2.2 Unlocking Locker

Unlocking conditions will be reviewed sequentially by priority groups from high to low. If any of the conditions are met, it will be unlocked and all assets in locker will be withdrawn to Releaser.

Unlocking conditions in the same priority group will have equal priority

The following is an example of an unlock pattern:

**# $JUZ 100 Based Fee**

**# Priority Group 1**:

- Unlock Condition 1.1:
  1. ***Unlock at: 01/12/2025***
  2. ***Pay 50 USDT to 0x6b54840f6f9d2030ff2f90a085d15ddd6ee08fc4***
  3. *Release by: 0x759ada5d6fd076d9f8d9aad7750cbb89acf59087*
- Unlock Condition 1.2:
  1. ***Unlock at: 01/1/2026***
  2. *Release by: 0x759ada5d6fd076d9f8d9aad7750cbb89acf59087*

**# Priority Group 2:**

- Unlock Condition 2.1:
  1. ***Unlock at: 01/12/2024***
  2. *Release by: 0x5f4e81aa5546e2b9533f5cf4e65b337ebf211634*

Assuming the user has deposited enough 100 $JUZ based fee for all cases then order of priority for unlocking:

- From ***01/12/2024*** to before ***01/12/2025:*** *Only 0x5f4e..11634 can unlock the locker*
- From ***01/12/2025*** to before ***01/1/2026:***
  - *If 0x759…9087 not pay 50 USDT yet to 0x6b5…08fc4, 0x5f4…11634* can call release() to unlock locker. (*Condition 1.1 has not been fully activated yet, so Condition 2.1 is still allowed to activate*)
  - If 0x759…9087 already paid 50 USDT to 0x6b5…08fc4, only (Condition 1.1) 0x759ada5d6fd076d9f8d9aad7750cbb89acf59087 can unlock the locker (*Since Condition 1.1 has been fully activated, Condition 2.1 will never be activated because Condition 1.1 is in Priority Group 1 with a higher priority than Group 2*)

- After ***01/1/2026:***
    - Since Condition 1.1 and Condition 1.2 have equal priority, whoever meets the condition first will be able to unlock the Locker first

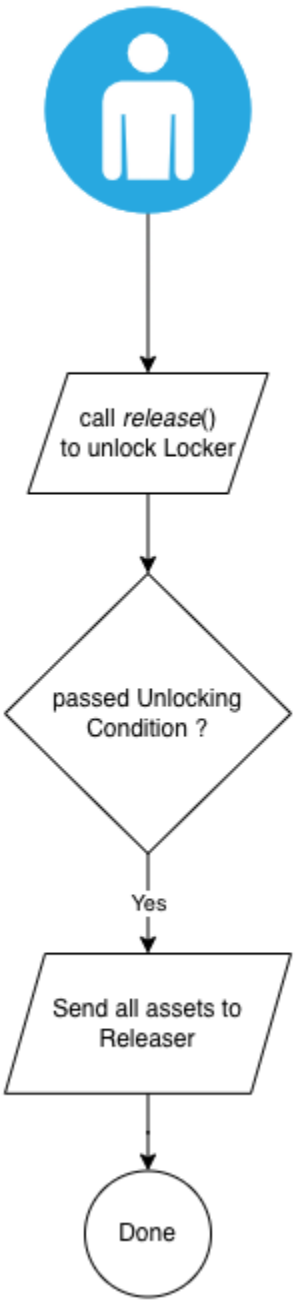


Figure 5: Unlock a locker

### 2.2.3 Staking & Earning Reward

If the user adds $JUZ to the locked locker. The system will start to stack and calculate the reward based on the apr setting and the amount you have stacked.

If $JUZ is already available in an open locker. The reward calculation period only starts when the locker is changed to a locked state until the owner claims time.

The reward will not be released at the time of unlocking the locker. Rewards can only be called and received by Locker's NFT owner.

If the system updates the new staking APR, the owner needs to claim to receive the effect of the new staking APR, otherwise, the reward is still calculated based on the old apr value and valid up to 1 year, After that time, the reward for that locker will not be born anymore. They need to claim to activate the new APR.

The reward will not auto be added to current $JUZ locked token amount to calculate new reward.

$$T_{stakingPeriod} = min(T_{current} - T_{latestStackClaimed}, T_{year})$$

$$A_{stackReward} = \frac{(A_{lastReward} + (A_{juz} * APR_{stack} * (T_{stakingPeriod} - T_{lastJuzDeposit})}{T_{year}}$$

Figure 6: Staking Reward Formula

### 2.2.4 Burn NFT

After Locker is unlocked, Owner can burn NFT and get reward token (if reward greater than zero).

## 2.3 The JuzuERC721 Contract

### 2.1.5 Mint Right

Factory has the right to mint Juzu ERC 721 NFT tokens, corresponding to ownership for each Locker

## 2.4 The JuzuERC20 Contract

$JUZ token will be generated when user staking $JUZ in locker and will be burned corresponding to the amount of based-fee salary when unlocking locker.

## 3. Conclusion

Juzu Protocol is a protocol that allows users to more easily manage various assets, with complex unlocking conditions that can serve many different practical needs. Juzu Protocol is not responsible for compensation for users who use this protocol and cause loss of crypto assets.