Semi-Fungible Tokens: Theory, Standard and Practice v0.9.1

Solv Protocol

Table of Contents

Chapter 5: The Outlook of SFTs

- 5.1 Portfolio Allocations and Structured Finance
  - 5.1.1 A Background
  - 5.2.2 Package SFT
  - 5.2.3 Tranche SFT
- 5.2 The Derivation of Rights
- 5.3 Real-World Assets (RWAs)
- 5.4 More Possibilities of ERC-3525

References

Remark

Overview

The Ethereum Foundation approved ERC-3525 Semi-Fungible Token[1] as its 35th ERC token standard on September 5, 2022, putting semi-fungible tokens (SFT) on par with fungible tokens (ERC-20) and non-fungible tokens (ERC-721). But why do we need another token standard? Aren't the massive market sizes of ERC-20 assets and ERC-721 NFTs abundant evidence that the crypto world already has everything it needs to succeed? What can ERC-3525 really offer us that existing standards cannot? This book is dedicated to answering these questions.

[1] Will Wang, Mike Meng, Ethan Y. Tsai, Ryan Chow, Zhongxin Wu, AlvisDu, "EIP-3525: Semi-Fungible Token," Ethereum Improvement Proposals, no. 3525, December 2020. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-3525.

What is an SFT

ERC-3525 is a general-purpose and omni-asset token standard that combines the quantitative attributes of an ERC-20 token and the descriptive features of an ERC-721 token (NFT). While ERC-20 can represent currencies, company stocks, and point systems, and ERC-721 can represent arts, collectibles, and digital goods, ERC-3525 is a versatile digital representation of ownership with ties to certain *values*. Assets an ERC-3525 SFT could represent range from simple instruments like a gift card, loyalty card, check, or a store voucher, to heavy-duty things like a bond, future or option contract, ETF, ABS, or a land title.

SFTs are highly descriptive, capable of not just displaying jpegs and gifs but also real-time on-chain data feeds and attachment of files. A user can fractionalize an SFT or combine multiple SFTs in the same manner as they would ERC-20 tokens. This is because an ERC-3525 token is identified by a unique ID but can also perform quantitative operations on other SFTs so long as they share the same SLOT – a unique mechanism of ERC-3525 that gives a like-kindness to otherwise unique entities. For example, two ERC-3525 bond SFTs with the face value of $200 and $300 and the identical maturity date, interest rate, and issuer ID displayed on the image could be quantitatively combined into a $500 bond.

The ERC-3525 is open-source, meaning developers can use it to build any asset they desire.

FT, NFT, and SFT

ERC-3525 is semi-fungible, leveraging the middle ground between the fungibles and non-fungibles, and since these concepts are confusing to many, we will explain them in this section.

Tokens that are *fungible* can be exchanged for one another. A fungible token can represent a currency, company stock, or a point system. In 2015, Ethereum's co-creator Vitalik Buterin proposed that token systems with ties to conventional value, such as sub-currencies such as USD and company stocks, would be easy to implement on Ethereum. In the same year, the ERC-20 was released, allowing developers to create interoperable token applications using fungible tokens.

Unlike fungible tokens, *non-fungible* tokens (NFT) are unique and cannot be interchanged. An NFT has a unique ID consisting of the address of the governing smart contract plus a serial number, and metadata containing the attributes of the NFT. As each NFT is unique, an NFT represents a variety of digital goods and assets like music, sound clips, pictures, and gifs, and game items. ERC-721, proposed by William Entriken and others in early 2018, is the definitive standard for NFT, and there were more than 36 billion dollars worth of NFTs minted in early 2021.

Most of the crypto assets today are ERC-20 and ERC-721 tokens, and most of the EIP (which stands for *Ethereum Improvement Proposal*) are either the application-level extensions or the adaptations of these token types. In the same vein, ERC-3525 is an extension of ERC-721 is fully backward-compatible with ERC-721.

Before the SFT, several protocols in DeFi (decentralized finance) have recognized the immense potential of the NFT for its descriptiveness and went on to build related products related to the NFT: Uniswap V3 LP tokens utilize ERC-721 to represent the unique liquidity position of LPs; Centrifuge allows users to onboard real-world assets as NFTs and secure a loan with those NFTs. Despite their potential, NFTs used in DeFi (often called financial NFTs) are gradually losing their edge as certificates attesting financial ownership, partly due to their illiquidity, partly due to their inflexibility.

To fill this gap, ERC-3525 Semi-Fungible Token was created. An ERC-3525 token is dynamic, liquid and flexible representation of digital ownership because

- It is semi-fungible. With an unique ID and rich metadata, an SFT can describe complex information, but is also fractionalizable as its underlying asset is liquid like an ERC-20 token.

- **It can contain any digital assets**. The SFT is a digital asset container that allows users to not only store any type of digital asset, but also send and receive it directly from it.
- **It is intelligent**. An SFT can interact with a complex environment and executes a code when triggered by messages or transactions.
- **It is structured finance-ready**. An SFT can structure any assets within itself regardless of token type or fungibility.
- **It is expressive**. An SFT user can directly access real-time on-chain data about a financial position via the self-generated SVG.

Using the blockchain technology and smart contracts, ERC-3525 SFTs provide a transparent, trustless, efficient, and self-executing solution. By leveraging the full potential of distributed ledgers, they offer interesting applications and the ability to create smart assets, including but not limited to

- **Monetary gifts** (coupons, checks, and "red envelopes") – Deposit any digital asset into an SFT as the underlying asset which may be withdrawn or inspected by the owner.
- **Certificates of deposit (CDs) and annuity** – Lock or unlock an asset immediately or at equal or unequal intervals when certain conditions are met.
- **Pawn tickets** – Secure a loan with an SFT and redeem it after paying back the principal and interest.
- **Acceptances** – Receive the face value of an SFT from its issuer (who may or may not be a bank). At maturity, buyers of fractional shares of SFT will be entitled to redeem their shares.
- **Streaming payments** – Block-by-block payments in real-time between SFTs. Users can modify or halt the payment at any time.
- **Debt instruments** – Issue a flexibly-collateralized bond through an SFT. SFT bonds can be traded on the open market in whole or fractions, and buyers will receive principal and interest on a pro-rata basis at maturity.
- **Structured products** – Package indices, ETFs, etc., pre-packaged into an SFT to create a structured product or risk-adjusted portfolio. Allocate seniority-based structured SFTs to one or multiple investors.

Endnote

This book explores ERC-3525 semi-fungible tokens. To begin, we'll define a token, and then we'll explore what makes a token fungible (or non-fungible). Continuing from Chapter 1, we'll explore how some DeFi protocols, such as Uniswap V3 and Centrifuge, use NFT for financial purposes and why the so-called financial NFT doesn't deliver what is expected. In Chapter 3, we'll explore semi-fungible tokens thoroughly, including their use cases, core mechanisms, and other technical and non-technical implications. Chapter 4 examines the use of SFTs through some of the earliest products developed by Solv Protocol, the creator and first adopter of ERC-3525. In Chapter 5, we give a sneak peek into how Web3 assets will evolve and how the SFT can play an important role in that process.

The information in this book is accurate to the best of the Solv team’s knowledge. As crypto space is rapidly evolving, the content of this book is prone to error. We are open for feedback from those who wish to improve its validity and scope of the book. Reach out to us via ERC-3525@solv.finance.

Chapter 1

Token, NFT, and SFT

Smart contract tokens used as proof of ownership have inspired endless innovations since they were first proposed by Vitalik Buterin. Token standards like ERC-20 and ERC-721 are at the cutting edge of defining and expressing digital ownership, and have fueled the multibillion-dollar crypto market.

But what if this isn't the whole story? What if there's an untapped market for digital assets based on a brand new token standard that we haven't even heard of? Our goal in this book is to inform the reader that this market exists and that ERC-3525 Semi-Fungible Tokens (SFTs) will bring that market to life.

Understanding SFTs requires a deep understanding of crypto and its fundamentals. Therefore, we must go all the way back to the genesis chapter, namely, the definition of a token. As we move forward, we will discuss recent things such as fungible tokens (FT) and non-fungible tokens (NFT), and, finally, why semi-fungibility is important to the future of our economy, network, and freedom.

1.1 What is a Token?

A token represents identity, social status, ownership, rights, or value. Historically, physical tokens have been used to facilitate transactions or establish orders. In schools, children receive "stars" or badges for good behavior; military officers or soldiers wear shoulder marks that bear ranks or other insignia; and arcade gamers use coins for playtime.

With the advent of the Information Age, physical tokens became digital ones, and accessing control is a typical function of a digital token. Tapping a badge or card equipped with a chip to pass a turnstile and IBM-powered Token Ring (a data link for local area networks in which all devices are connected to a Ring and can pass one or multiple frames of data from one to another) are just a couple of examples of how our lives benefit from the use of digital tokens.

The definition of a token is never static, rather, it evolves with the technology of the day. In the blockchain era, tokens have a new meaning. In *Token Safe Harbor Proposal*[2], the U.S. SEC commissioner Hester M. Peirce defines a blockchain token as

a digital representation of value or rights

(i) that has a transaction history that:

(A) is recorded on a distributed ledger, blockchain, or other digital data structure;
(B) has transactions confirmed through an independently verifiable process; and
(C) cannot be modified;

(ii) that is capable of being transferred between persons without an intermediary party; and

(iii) that does not represent a financial interest in a company, partnership, or fund, including an ownership or debt interest, revenue share, entitlement to any interest or dividend payment.

In Peirce's definition of a token, we identify three mainstays:

[2] Token Safe Harbor Proposal 2.0, https://www.sec.gov/news/public-statement/peirce-statement-token-safe-harbor-proposal-2.0

1. It is a digital representation of value of rights;
2. It can be transferred;
3. It provides a full history of transactions recorded on blockchain.

Physical tokens, digital tokens, and blockchain-based tokens all share a fundamental characteristic: they serve as symbols, labels, or certificates that represent value or rights. Tokens are empowered by rules or systems of rules, and these rule-based systems can be software, network protocols, management systems, legal systems, or social conventions. The blockchain uses a different form of system: smart contracts.

1.2 Smart Contracts and Tokens

Human civilizations rely heavily on contracts. The English philosopher Thomas Hobbes (1588-1679) noted that a world without rights and contracts leads to individuals living "solitary, poor, nasty, brutish and short" lives. Contracts are legally enforceable promises. In more granular terms, they are **multistep, collaborative, and consensus-based protocols that govern the rights and obligations of their parties**. In a contract, the whats, the whos, and the hows must be spelled out in writing or in other forms, and the obligations that each party must fulfill as well as any consequences for failure to do so must be clearly stated. To make a contract work, we need three things: **the contract itself** (the totality of the terms), **an execution system** (the legal and/or reputation-based system that binds a contract), and a certificate attesting the contract, or **the proof of contract.**

(Note that popular contracts such as commercial papers, coupons, bonds, or cash notes, are so standardized these days to the point where they almost do not appear as contracts.)

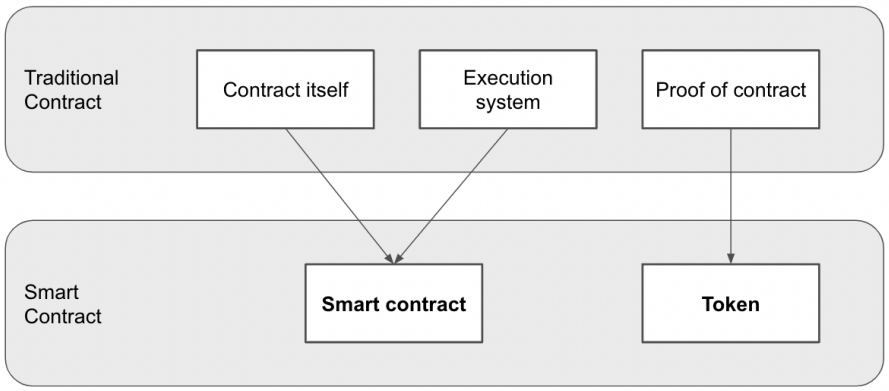

Figure 1.1 The smart contract unifies the contract itself and its execution system

By blurring the fine line between the contract itself and execution system, blockchain technology has helped to create a much more unified and efficient structure for contract creation and execution, and this structure is the smart contract. A smart contract is a set of instructions that follow a simple *if/when p then q* rule that is verified through a digital signature and executed on a distributed, immutable and transparent network called a blockchain. Smart contracts store and execute rules, while tokens represent ownership and rights. In this sense, a blockchain token is **a proof of a smart contract**. Not all blockchain tokens, however, are created equal. Among the key factors we consider when classifying blockchain tokens is their fungibility.

1.3 What Makes a Token Fungible (and Non-fungible)?

Most crypto users have used fungible tokens, but few could identify them. In order for a token to be considered fungible, it must be able to be replaced by another identical token. A fungible token is computable, which means that multiple tokens can be added together (or subtracted).

On the blockchain, fungible tokens are usually ERC-20 tokens (which stands for "Ethereum Request for Comments 20"), although a minority of fungible tokens exist in ERC-20 variants, such as BEP-20, SLP, and TRC-20. ERC-20 is the most widely used token standard on the Ethereum network, and it was proposed just four months after the mainnet went live. ERC-20 tokens can be used for a variety of things, including stock shares, loyalty points, and

virtually all cryptocurrencies. Currently, ERC-20 tokens are worth over US$100 billion.

Non-fungible tokens, or NFTs, fall on the other end of the spectrum from ERC-20 tokens. While ERC-20s are interchangeable, NFTs have their own ID and metadata, which makes them unique. The ID consists of a serial number and the address of a maternal smart contract, while metadata describes the NFT's diverse attributes. The uniqueness of NFTs makes them suitable for storing digital goods like art, domain names, sound clips, in-game items, and more.

A NFT has many advantages over fungible tokens, making it a great choice for the proof of a smart contract. Let's take a closer look at these advantages.

Descriptive

NFTs are ideal for representing complex or simple contracts, because metadata allows a broad array of display and storage options.

Expressive

Since NFTs can display a wide range of information, they are ideal for describing contract terms (such as strike price, interest rate, maturity, and installment terms), which an ERC-20 token could not accommodate easily.

Cost-effective

The use of NFTs can significantly reduce the development, deployment, and maintenance costs of smart contracts and reduce the number of smart contracts needed to implement complex transactions.

The deployment of one new token for each component of portfolio allocation, redemption rules, and fee structures would increase the cost of running the transaction. NFTs require only a single smart contract to be deployed by the developer.

Capital-efficient

As ERC-20s support only fungible liquidity pools and are not suitable for integrating other smart contracts or expanding to other pools, this results in some DeFi protocols shackled by the inefficiency of ERC-20s. An NFT, on the other hand, allows management of multiple heterogeneous asset pools with a single smart contract, significantly improving capital efficiency.

1.4 The Semi-Fungible Token and ERC-3525

A semi-fungible token (SFT) is a type of digital asset based on blockchain technology that blends fungible and non-fungible characteristics. More specifically, SFTs combine the quantitative attributes of a fungible token and the descriptive features of a non-fungible token.

To understand SFTs, we must first consider what kind of ownership or rights they are meant to represent. An SFT can be used to represent any digital asset that has quantitative attributes and is required to be fractionalized or combined in certain circumstances. Bonds, coupons, vouchers, invoices, promissory notes, and land titles are some examples of such assets, as are futures, options, and ABS. In a financial application, using an SFT is probably most straightforward when combining a $20 bond and a $30 bond into a $50 bond.

The ERC-3525 Semi-Fungible Token Standard (approved September, 2022) is the first EIP standard for semi-fungible tokens, and it is fully scalable into a universe of use cases and functionalities. Solv Protocol, the creator of the standard, uses the ERC-3525 as a container for digital assets, in which users can package unlimited amounts and types of tokens for various purposes. A user of the ERC-3525 token could also transfer digital goods directly into and out of it freely, making it easy to repay bond issuances to multiple lenders quickly and efficiently.

An ERC-3525 token can empower a Web3 user in several ways.

1. Create advanced digital financial assets, such as CDs, checks, bonds, options, futures, swaps, insurance policies, equities, or ABS. An ERC-3525 token provides liquidity to the user, since it could be split into infinitely many pieces. Further, it provides strong visualization to make sure that the contract and any attachments are easily accessible.

   Perhaps the greatest advantage of the ERC-3525 token used as a financial asset is its look-through, fund-in-fund transparency enabled by the blockchain, by which users can find out how the underlying investment assets perform through the complex structure of an ERC-3525 token.

2. Create fractionalized virtual lands, upgradeable/mergeable in-game items, virtual membership cards, gift cards, raffle tickets, etc.

3. Onboard real-world assets (RWAs). ERC-3525 tokens could embed legally binding contract documents (in PDF files, for example) and enable access through blockchain oracles to real-world information for the related assets.

   ERC-3525s can be used as tokenized RWAs in a number of different ways, including tokenizing solar panels and capturing their unearned profits as well as tokenizing real estate. Users can split or merge tokens by quantitative parameters such as surface area (of a solar panel) or square footage (of a real property). As tokenized RWAs, ERC-3525s are regulator-friendly since they're good at reducing transaction frictions, maintaining secure decentralized records, and authenticating provenance.

4. Fractionalized wallet. The ERC-3525 protocol can nest any crypto asset inside, so an SFT can act as a blockchain wallet for any digital assets, plus fractionalization and transferability.

5. Tokenize and visualize smart contracts. Every ERC-3525 token contains a full-blown smart contract that can store, send, or receive a given crypto asset if invoked. ERC-3525 can be used to also tokenize traditional contracts like trade agreements, lease agreements, or loan agreements, where relevant parties can leverage the standard for fast, automated, and foolproof contract execution.

SFT's developers worked hard to ensure that it will be just as gas-efficient as ERC-721 - if not more. And every wallet, protocol, or marketplace that supports NFT will be able to integrate ERC-3525 assets since ERC-3525 is backward-compatible with ERC-721.

Furthermore, ERC-3525 is completely open-source to encourage collaboration and innovation. To date, protocols such as FujiDAO and Buffer Finance have begun building on-chain options markets upon ERC-3525, and Solv Protocol is the first and largest market that uses ERC-3525 tokens as fractionalized on-chain bonds to help institutions get financing and access credit markets.

1.5 Summary

In this chapter, we have first covered the fundamentals of tokens. A token is a physical or symbolic object that represents identity, social status, ownership, rights, or value. Digital tokens act similarly to physical tokens except that digital tokens can enable digitally-based access control. A blockchain token is an advanced form of digital token because it is decentralized and offers a transparent and immutable transaction history that can't be achieved with traditional tokens.

Fungible tokens are interchangeable with each other, and non-fungible tokens are not. While fungible tokens are ideal for assets focused on liquidity, non-fungible tokens are ideal for storing and displaying unique visually-rich jpegs or gifs, or text-rich financial contracts. An SFT powered by ERC-3525 combines the quantitative attributes of an ERC-20 token with the descriptive attributes of an ERC-721 token, and is suitable for representing sophisticated contracts such as bonds, futures, lease agreements, or even smart contracts.

Understanding the intricate relationships between FTs, NFTs, and SFTs may still be tricky for some. Therefore, bridging concepts like *financial NFTs* may be a good place to start.

Chapter 2

Financial NFTs: Applications and Technologies

Decentralized Finance, or DeFi, is a financial system that uses cryptocurrency and blockchain technology, instead of centralized intermediaries, to manage financial transactions. Similar to traditional finance (TradFi), DeFi's primary objective is to maximize capital efficiency when allocating crypto assets.

In the broadest sense, financial NFT refers to tokenized ownership of advanced digital assets. However, leveraging financial NFTs without a proper infrastructure would be like building a house without a foundation - sooner or later, it would collapse. Two Ethereum token standards that use NFTs are ERC-721 and ERC-1155. ERC-721 NFTs are blockchain-based cryptographic assets with unique ID codes and metadata that distinguish them from each other, and ERC-1155 Multi-token Standard enables the implementation of multiple token types on a single deployed contract. Each of these standards serves as a financial NFT infrastructure in a slightly different way.

In this chapter, we'll first delve into how each of these standards are contributing their skill sets to DeFi and then discuss their weaknesses as financial NFTs. In the

second half of the chapter, we will provide a technical overview of these standards for the technologically-inclined readers of this book.

2.1 ERC-721 Financial NFTs

ERC-721 NFTs have been utilized by numerous DeFi protocols for a variety of purposes. To demonstrate the skills of NFTs in DeFi applications, we cover Uniswap, Centrifuge, and Solidly. Please note that this section is for educational purposes only and does not provide financial advice.

Uniswap

Uniswap is a decentralized exchange (DEX) that offers automated market-making (AMM) instead of the "highest bid-lowest ask" model. Uniswap v1 is an on-chain system of smart contracts on the Ethereum blockchain, implementing an automated liquidity protocol based on a what's called "constant product formula." Each Uniswap V1 pair stores pooled reserves of two assets, and provides liquidity for those two assets, maintaining the invariant that the product of the reserves cannot decrease. Uniswap V2 is a new implementation based on the same formula, which also enables the creation of arbitrary ERC20/ERC20 pairs, rather than supporting only pairs between ERC20 and ETH. The issue with V2, however, is that liquidity providers (LPs) have no active position management. Without it, liquidity is distributed in all price ranges between zero and infinity instead of within the range of LP's own choice.

Uniswap V3[34] solves the position management problem by allowing LPs to create their own price range and receive an ERC-721 NFT representing the range. LP NFTs strike the right balance between capital efficiency and risk parity by providing the option of customizing laser-focused liquidity positions instead of committing capital unproductively to a wide range.

[3] Uniswap V3 Whitepaper, https://uniswap.org/whitepaper-V3.pdf
[4] Uniswap V3–Is NFT the only way to upgrade DeFi?, https://medium.com/solv-blog/solv-talk-uniswap-V3-nft-is-the-only-way-to-upgrade-defi-2ed2686bf1a3

Figure 2.1 Concentrated liquidity position in Uniwap V3

A transaction fee is used to reward LPs for taking on concentrated (customized) liquidity positions. As figure 2.2 shows, following the implementation of Uniswap V3, the protocol's trading volume has increased.

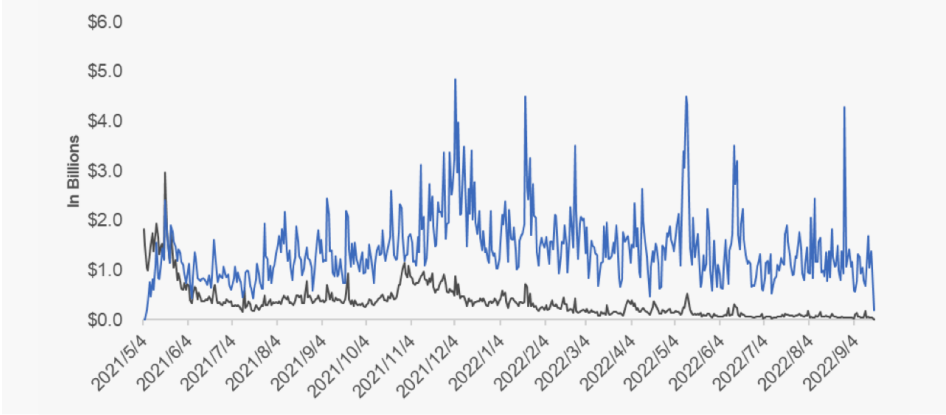

Figure 2.2 Comparing the volume of trade between V2 (gray) and V3 (blue) (Source: Nomics, Dune Analytics)

With more capital flowing into DeFi, LPs need better risk-reward allocation and granular capital control. Therefore, the NFT will earn a reputation as a safe and efficient capital management tool, not just a nice-looking jpeg.

Centrifuge

Centrifuge[5] is a network where real-world assets could be represented by ERC-721 NFTs for various purposes, such as lending. To finance real-world assets (RWAs), businesses ("asset originators") can tokenize their financial assets into privacy-enabled NFTs and then use those NFTs as collateral in an open marketplace of real-world asset pools.

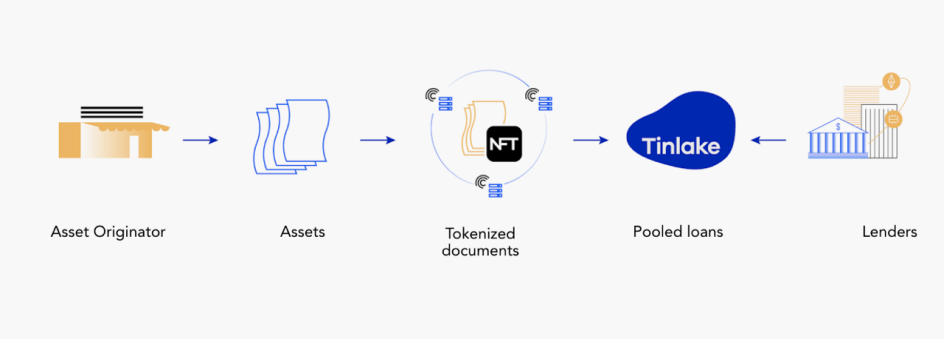

Figure 2.3 Borrowing and lending through Centrifuge (Source: Centrifuge whitepaper)

Users of Centrifuge bridge RWAs to DeFi through the network's P2P protocol. The process is simple: asset originators mint NFTs from off-chain Centrifuge documents containing information about their assets and create a pool. After that, DeFi investors and protocols lock their investment in DAI into the liquidity pool and receive an ERC-20 token (called TIN or DROP) that enables them to earn a stable return from the asset originators.

Bridging RWAs to DeFi is all about ensuring the document with specific field types representing the asset cannot be easily manipulated. Using Centrifuge's peer-to-peer (P2P) method, asset data could be tokenized into NFTs and verified between collaborators. Moreover, any service providers with document-level read access or field-level write access can be added as collaborators to demonstrate the verifiability of NFTs. (To learn more about how Centrifuge works, read its docs here: https://docs.centrifuge.io/)

[5] Centrifuge Whitepaper, https://docs.centrifuge.io/getting-started/p2p-protocol/

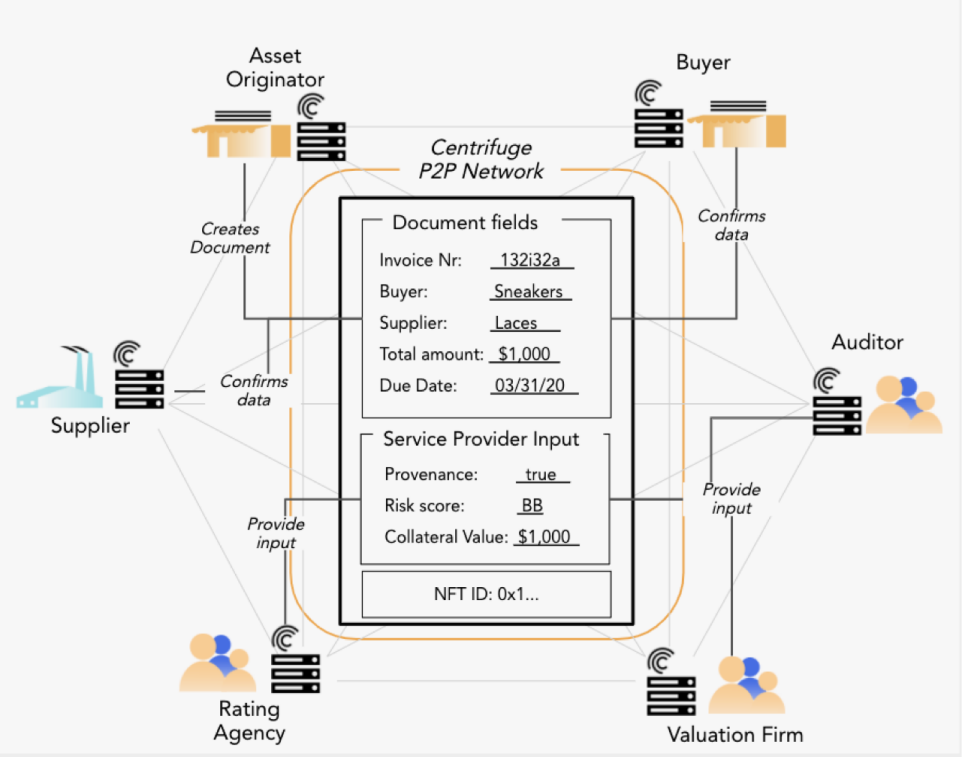

Figure 2.4 Centrifuge's P2P network (Source: Centrifuge Whitepaper)

Not surprisingly, regulations and compliance are mandatory for RWAs. Centrifuge's prospects are attractive precisely because NFTs are highly descriptive, fraud-proof, and immutable, all of which place NFTs in a favorable light in the eyes of regulators and law enforcement.

Solidly

As a Fantom-based DEX, Solidly was launched by Andre Cronje, a well-known DeFi developer dubbed the "Godfather of DeFi." Cronje is best known for his contributions to numerous Ethereum projects, including Yearn.

Among Andre's most exciting innovations is ve(3,3)[6], which utilizes "Flexible Emission Levels" to incentivize staking tokens into pools while creating a "win-win" value proposition for all participants. For example, if the total supply of a token is 10 million and 5 million tokens are staked, ve(3,3)'s flexible emission rate will yield 5 million tokens per week. These tokens are distributed to the market and LPs.

[6] ve(3,3): An Introduction into the New and Ambitious Solidly Exchange, https://avgjoescrypto.substack.com/p/ve33-an-introduction-into-the-new

There are two weaknesses to veTokens (tokens representing lock positions). First, as veTokens cannot be sold or bought (no liquidity). Second, veToken holders cannot stake more tokens through the same address in a separate timeframe (no flexibility).

The solution is NFT. By wrapping veTokens into an NFT ("veNFT"), an LP can sell his or her veNFT in an open market or, if needed, secure a loan with it. Now that veNFTs are here, LPs can own multiple locks and enjoy the flexibility that comes with it.

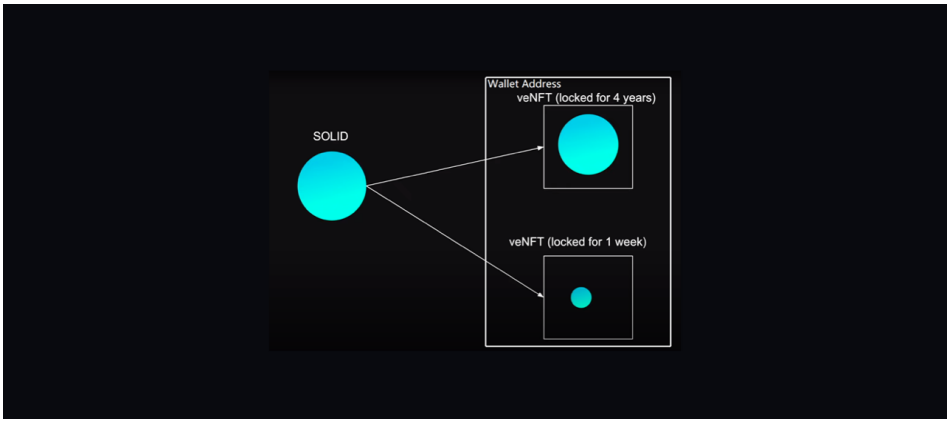

Figure 2.5 Lock NFT-ization of Solidly

2.2 ERC-1155 Financial NFTs

Fuji DAO

Fuji DAO[7] is an infrastructure protocol that aggregates lending-borrowing crypto markets. It sources its liquidity from three lending protocols: Compound, Aave and dYdX. Fuji DAO proposes isolated debt positions for each collateral/borrow pair, allowing for better risk management and more effective interest rate optimizations.

As a borrowing aggregator, Fuji DAO is generally required to map its internal positions relative to a protocol, using, for example, Aave debtToken standard, an ERC-20 position-mapping approach.

[7] Fuji DAO, https://medium.com/fuji-finance/nfts-and-defi-innovation-b05f49b81831

However, the ERC-20 standard has some complications that limit Fuji DAO's capacity as an aggregator, like deploying extraordinary amounts of smart contracts, needing more approvals for each transaction, hefty gas fees, as well as the fact that Aave debtToken was an interest-bearing and non-transferable asset.

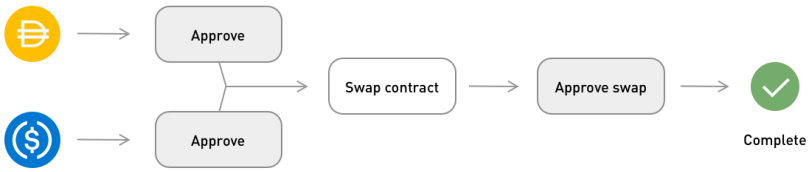

Figure 2.6 Fuji DAO's swap operations of ERC-20

The scheme in Figure 2.6 shows that to swap ERC-20 cryptocurrencies, the user must approve each transaction and then approve the swap operations. This is not only inefficient, but also will result in gas price spikes during periods of volatility and high volume of swapping activities.

The ERC-1155 standard is a token framework created by the blockchain-based gaming platform and asset marketplace, Enjin, which allows for more efficient trades by bundling transactions, making transfers less expensive.

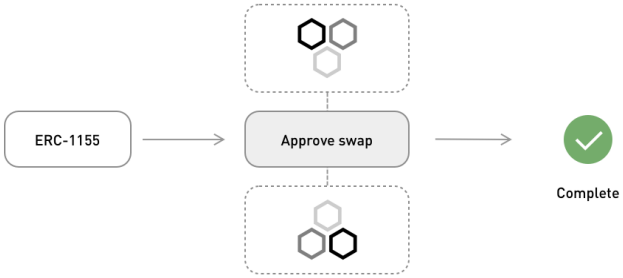

Figure 2.7 Swapping multiple currencies with ERC-1155

With ERC-1155 tokens, multiple currencies or assets can be swapped at once, and users can enjoy a host of benefits including fast transaction approval and reduced gas fees.

Alpha Homora V2

Homora is DeFi's first leveraged yield farming protocol, built by Alpha Venture DAO. In Alpha Homora V2[8], yield farmers (who are liquidity providers) can perform leveraged yield farming on numerous asset pairs and DEXes and receive higher farming APY and trading fee rewards than other no-leverage yield farming protocols.

Like Fuji DAO, Alpha Homora V2 tokenizes farming positions into ERC-1155 financial NFT, which are treated as a class of ERC-20 tokens grouped by IDs. Since reward amounts for farming positions depend on two factors, LP shares (how much LP is staked by the user) and stake time, if two users stake at the same time, their rewards split will simply be proportional to their LP shares. Thus, positions with the same stake time will have the same ID and can receive rewards efficiently under the one ERC-1155 smart contract, instead of numerous ERC-20 ones.

Being a less "fragmented" solution to the pain point of deploying a separate contract for each new token, ERC-1155 enables projects like Homora V2 to reduce gas expenses as well as frictions to manage farming rewards by allowing them to interface with a multi token standard.

Furthermore, Alpha Homora V2 utilizes ERC-1155 tokens to tokenize farming positions to use as collateral, while the actual LP tokens must be staked at the corresponding staking contracts to earn yield farming rewards.

2.3 Technical Overviews of ERC-721 and ERC-1155

2.3.1 ERC-721 Non-fungible Token Standard

ERC-721[9] NFTs are blockchain-based cryptographic assets with unique ID codes and metadata that distinguish them from each other. What's at the core of an

[8] Alpha Homora V2: ERC-1155 Tokenized Positions, https://alphaventuredao.io/blog/alphahomorav2-tokenized-position
[9] William Entriken, Dieter Shirley, Jacob Evans, Nastassia Sachs, "EIP-721: Non-Fungible Token Standard," Ethereum Improvement Proposals, no. 721, January 2018. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-721

ERC-721 NFT is `_tokenID`, which is a 256-bit unsigned integer that enables the identification of an ERC-721 NFT. `_tokenID` represents the uniqueness of ownership over the token itself as well as whatever underlying (digital or physical) assets it has.

To describe the ownership, ERC-721 has a metadata extension featuring properties of `name`, `description`, and `image`, all of which delineates the key information associated with the token:

```
{
    "title": "Asset Metadata",
    "type": "object",
    "properties": {
        "name": {
            "type": "string",
            "description": "Identifies the asset to which this NFT represents"
        },
        "description": {
            "type": "string",
            "description": "Describes the asset to which this NFT represents"
        },
        "image": {
            "type": "string",
            "description": "A URI pointing to a resource with mime type image/*
representing the asset to which this NFT represents. Consider making any images at a
width between 320 and 1080 pixels and aspect ratio between 1.91:1 and 4:5 inclusive."
        }
    }
}
```

`_tokenID` is useful for the mapping of financial applications with users' various demands. Furthermore, the enhanced text description within ERC-721's metadata helps differentiate configurations with complex equities of a specific scenario, such as multi-position market-making. With visualization, the image file in the metadata will display information that can't be presented easily in plain text.

Another core feature of ERC-721 is address-to-address transfer, implemented through `transferFrom()` and `safeTransferFrom()` and executed on `_tokenID` (other than `_value` in an ERC-20 token):

```
    /// @notice Transfers the ownership of an NFT from one address to another address
    /// @dev This works identically to the other function with an extra data
parameter,
    ///  except this function just sets data to "".
    /// @param _from The current owner of the NFT
    /// @param _to The new owner
    /// @param _tokenId The NFT to transfer
    function safeTransferFrom(address _from, address _to, uint256 _tokenId) external
payable;
```

```
    /// @notice Transfer ownership of an NFT -- THE CALLER IS RESPONSIBLE
    ///  TO CONFIRM THAT `_to` IS CAPABLE OF RECEIVING NFTS OR ELSE
    ///  THEY MAY BE PERMANENTLY LOST
    /// @dev Throws unless `msg.sender` is the current owner, an authorized
    ///  operator, or the approved address for this NFT. Throws if `_from` is
    ///  not the current owner. Throws if `_to` is the zero address. Throws if
    ///  `_tokenId` is not a valid NFT.
    /// @param _from The current owner of the NFT
    /// @param _to The new owner
    /// @param _tokenId The NFT to transfer
    function transferFrom(address _from, address _to, uint256 _tokenId) external
payable;
```

Note that ERC-721s and ERC-20s differ in the token transfer process. Transferring an ERC-721 NFT from one address to another means transferring proof of ownership in its entirety while transferring ERC-20s means the transfer of numerical value of the same ERC-20 token.

2.3.2 ERC-1155 Multi Token Standard

ERC-1155 Multi-token Standard[10] enables the implementation of multiple token types on a single deployed contract. The standard was created so that multiple types of in-game items (fungible or not) could be deployed on a single deployed contract. Its widest usage is in blockchain games.

Technically, both ERC-20 and ERC-721 require a new contract to be deployed for each token type or collection. This mechanism adds superfluous bytecode to Ethereum and thus hurdles certain functionalities such as token-address separation.

[10] Witek Radomski, Andrew Cooke, Philippe Castonguay, James Therien, Eric Binet, Ronan Sandford, "EIP-1155: Multi Token Standard," Ethereum Improvement Proposals, no. 1155, June 2018. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-1155

ERC-1155's main data structure is `_id`, which represents configurable token types, and `_value`, a quantitative attribute representing the transfer amount of the corresponding token. Like the ERC-721, ERC-1155 also features a metadata extension defined by things like `name`, `decimals`, `description`, `image`, and other `arbitrary properties`. Here is ERC-1155's JSON Schema:

```
{
    "title": "Token Metadata",
    "type": "object",
    "properties": {
        "name": {
            "type": "string",
            "description": "Identifies the asset to which this token represents"
        },
        "decimals": {
            "type": "integer",
            "description": "The number of decimal places that the token amount should
display - e.g. 18, means to divide the token amount by 1000000000000000000 to get its
user representation."
        },
        "description": {
            "type": "string",
            "description": "Describes the asset to which this token represents"
        },
        "image": {
            "type": "string",
            "description": "A URI pointing to a resource with mime type image/*
representing the asset to which this token represents. Consider making any images at a
width between 320 and 1080 pixels and aspect ratio between 1.91:1 and 4:5 inclusive."
        },
        "properties": {
            "type": "object",
            "description": "Arbitrary properties. Values may be strings, numbers,
objects, or arrays."
        }
    }
}
```

Distinct from ERC-721's `_tokenID`, which represents a token's identification, ERC-1155's `_id` represents the configurable type of a token. An ERC-1155 powered sword may have several attributes in terms of its material, color, power, and so on. When a forge or transfer event occurs, ERC-1155 removes the need to

deploy multiple contracts for each of the sword's attributes and, instead, implements the event in a single transaction.

In financial applications, ERC-1155's `_id` and corresponding metadata can be utilized for differentiated, more nuanced configurations with complex equities, with certain quantitative operations on attribute `_value`. Because tokens under ERC-1155 with the same `_id` are fungible, ERC-1155's financial applications may be limited as financial contracts in a broad sense non-fungible.

Visualization is key to differentiating financial NFTs from NFT collectibles. ERC-1155 tokens' ability to embed image files in their metadata means they are good at delivering information visually which can't be delivered easily with plain text.

Another core feature of ERC-1155 is address-to-address transfer, implemented through `safeTransferFrom()` and executed through (`_id`, `_value`):

```
    /**
        @notice Transfers `_value` amount of an `_id` from the `_from` address to the
`_to` address specified (with safety call).
        @dev Caller must be approved to manage the tokens being transferred out of the
`_from` account (see "Approval" section of the standard).
        MUST revert if `_to` is the zero address.
        MUST revert if balance of holder for token `_id` is lower than the `_value`
sent.
        MUST revert on any other error.
        MUST emit the `TransferSingle` event to reflect the balance change (see "Safe
Transfer Rules" section of the standard).
        After the above conditions are met, this function MUST check if `_to` is a
smart contract (e.g. code size > 0). If so, it MUST call `onERC-1155Received` on `_to`
and act appropriately (see "Safe Transfer Rules" section of the standard).
        @param _from    Source address
        @param _to      Target address
        @param _id      ID of the token type
        @param _value   Transfer amount
        @param _data    Additional data with no specified format, MUST be sent
unaltered in call to `onERC-1155Received` on `_to`
    */
    function safeTransferFrom(address _from, address _to, uint256 _id, uint256 _value,
bytes calldata _data) external;
```

Batch Operations make it possible to operate over multiple tokens in a single transaction efficiently. ERC-1155 provides two functions, `balanceOfBatch()` and `safeBatchTransferFrom()`, that make querying multiple balances and transferring multiple tokens simpler and less gas-intensive.

```
    /**
        @notice Transfers `_values` amount(s) of `_ids` from the `_from` address to
the `_to` address specified (with safety call).
        @dev Caller must be approved to manage the tokens being transferred out of the
`_from` account (see "Approval" section of the standard).
        MUST revert if `_to` is the zero address.
        MUST revert if length of `_ids` is not the same as length of `_values`.
        MUST revert if any of the balance(s) of the holder(s) for token(s) in `_ids`
is lower than the respective amount(s) in `_values` sent to the recipient.
        MUST revert on any other error.
        MUST emit `TransferSingle` or `TransferBatch` event(s) such that all the
balance changes are reflected (see "Safe Transfer Rules" section of the standard).
        Balance changes and events MUST follow the ordering of the arrays
(_ids[0]/_values[0] before _ids[1]/_values[1], etc).
        After the above conditions for the transfer(s) in the batch are met, this
function MUST check if `_to` is a smart contract (e.g. code size > 0). If so, it MUST
call the relevant `ERC-1155TokenReceiver` hook(s) on `_to` and act appropriately (see
"Safe Transfer Rules" section of the standard).
        @param _from    Source address
        @param _to      Target address
        @param _ids     IDs of each token type (order and length must match _values
array)
        @param _values  Transfer amounts per token type (order and length must match
_ids array)
        @param _data    Additional data with no specified format, MUST be sent
unaltered in call to the `ERC-1155TokenReceiver` hook(s) on `_to`
    */
    function safeBatchTransferFrom(address _from, address _to, uint256[] calldata
_ids, uint256[] calldata _values, bytes calldata _data) external;
```

2.4 Summary

Flexibility in creating on-chain financial tools and assets will be in high demand in the coming years. The ERC-721 and ERC-1155 NFT standards may work, but they are not long-term solutions due to their lack of liquidity and flexibility. A major disadvantage of ERC-721s is that they cannot be fractionalized, do not support token-to-token transfer, and their metadata is only suitable for displaying

static art, not real-time data feeds. While ERC-1155 is a gamers-driven standard focused on representing and transferring homogeneous items, the level of numerical fine-tuning and granularity required by most sophisticated financial applications are not within the skill set of ERC-1155. Both ERC-721 and 1155 lack the expressivity and would rely on smart contract developers for each new use case that surfaces. Therefore, it may be appropriate to classify ERC-721s and 1155s as *simple* financial NFTs.

In order to build a more sustainable and decentralized Web3, we must learn from simple financial NFTs and build far better financial NFTs. Maybe the ERC-3525 Semi-Fungible Token Standard will fit the bill.

Chapter 3

ERC-3525: Semi-Fungible Token Standard

As discussed in the previous chapters, simple financial NFT standards like ERC-721 and 1155 tokens aren't suitable for designing and creating flexible, tailor-made Web3 financial assets. The ERC-3525 Semi-Fungible Token (SFT) standard, a general-purpose standard for advanced financial instruments, has the potential to solve problems too costly to solve with ERC-721s or 1155s. This chapter explores the ERC-3525 SFT standard and its inner workings.

3.1 Transcending Financial Instruments

In a broad sense, financial instruments[11] are monetary contracts between parties. All financial instruments are on some level financial *contracts*, and the easiest way for someone to enter into or get out of a financial contract is by purchasing or selling a financial instrument. The traditional finance sector (TradFi) is home to

[11] Financial Instrument, https://en.wikipedia.org/wiki/Financial_instrument

diverse financial instruments including bonds, commercial papers, options contracts, futures contracts, repurchase agreements (repos), swaps, ETFs, and ABS. These instruments (shown in figure 3.1) have a few things in common: they are fairly standardized, easy to use, and to a great extent, liquid.

		Financial Instruments			
		Securities	**Other Cash**	**Derivatives** (Exchange-Traded)	**Derivatives** (Over-the-Counter)
Asset Class	**Long-term debt** (>1 year)	Bonds	Loans	· Bond · Futures · Options	· Interest rate swaps · Interest rate caps and floors · Interest rate options · Exotic derivatives
	Short-term debt (≤ 1 year)	· Bills (e.g., T-bills) · Commercial paper	· Deposits · Certificates of deposit	Short-term interest rate futures	· Forward rate agreements (FRA)
	Equity	Stock	N/A	· Stock options · Equity futures	· Stock options · Exotic derivatives
	Foreign Exchange	N/A	Spot foreign exchange	Currency futures	· Foreign exchange options · Outright forwards · Foreign exchange swaps · Currency swaps

Figure 3.1 Types of financial instruments in TradFi (Source: Wikipedia)

The key premise of Web3 is to transform the financial instruments from TradFi into advanced, blockchain digital assets. Most currencies, stock shares, and point systems can be represented by ERC-20 tokens, but this is only the first leg of TradFi. As for financial instruments, the second leg of TradFi, there are no successful parallels before ERC-3525.

When it comes to defining the candidate asset for financial instruments, we need to consider several criteria. This includes the ability to send or receive payments directly from the token, which increases the efficiency of delivering a contract, and visualization for the ease of use and the look-through transparency. But

fundamentally, we think that such an asset must have two overarching characteristics in order to represent most, if not all, financial instruments:

- **Fungible** (can be fractionalized). A user should be able to split up a token denominated in $100 USD into two $50 tokens or four $25 tokens, etc. This same applies to combining tokens as well.
- **Transitive**. Fractionalizing and combining tokens should not alter the *token type* but only the value tied to the tokens.

The ERC-3525 Semi-Fungible Token is the only token standard to fit the bill and thus an ideal vehicle for Web3 digital assets.

The rest of this chapter provides a technical overview of the ERC-3525 Semi-Fungible Token standard. Feel free to skip to Chapter 4: *Semi-Fungible Tokens in Use* if you wish to learn about the real-world use cases of ERC-3525 tokens.

3.2 An Overview on Token Technology

The Ethereum whitepaper describes a currency, or a token system, as "a database with one operation."[12] A token or tokenized asset, in other words, is a certain type of data that can be operated upon.

This perspective provides an easy way to understand an ERC-20 token[13]: their core data structure is a 256-bit unsigned integer defined by `_value` or `balanceOf()`, and a decimal function, `decimals()`, will determine the token amount. Transferring tokens by this amount, which is implemented through `transfer()`, `transferFrom()`, and `approve()`, is the main operation for sending tokens from one address to another.

Unlike ERC-20s, ERC-721s abandoned the token amount in their data structure. As a result, the 256-bit unsigned integer that originally designates the token amount now defines a unique `_tokenID` which represents the ownership over digital or physical assets. Although NFTs share the same address-to-address operation as ERC-20s, the ERC-20 transfer is quantitatively operated on the token

[12] Ethereum Whitepaper: https://ethereum.org/en/whitepaper/
[13] Fabian Vogelsteller, Vitalik Buterin, "EIP-20: Token Standard," Ethereum Improvement Proposals, no. 20, November 2015. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-20.

amount, whereas the ERC-721 transfer is operated qualitatively on the `_tokenID` and the amount will always be "1," to simulate the transfer of ownership, implemented by `transferFrom()`, `safeTransferFrom()`, and `approve()`, etc.

Another noteworthy feature of ERC-721 is its metadata extension. With properties like `name`, `description`, and `image`, ERC-721 NFTs could for the first time in history display rich visual information on the blockchain. For example, in the metadata extension for Uniswap's V3 LP token, custom price curve and liquidity position are presented in an aesthetically pleasing way. On the other side of the coin, however, since ERC-721 can't be fractionalized, the liquidity for V3's positions is still largely limited.

A possible solution is the combination of the `_value`-based quantitative attributes of ERC-20s and the `_tokenID`-based qualitative data structures of ERC-721s. It may lead us to a happy semi-fungible middle between the fungible and the non-fungible.

ERC-1155 token standard was among the early efforts to explore the concept of SFT. In a nutshell, the ERC-1155 implements multiple token types on a single contract, and it does so by using an `_id` attribute in its data structure to represent a configurable token type and `_value`, a quantitative attribute, to represent the transfer amount of that token type. For the deployment of multiple token types, ERC-1155 also creates a new transfer model, the Batch Transfer `safeBatchTransferFrom()`, where transferring multiple types of tokens in different amounts from one group of users (addresses) to another is made possible.

It should be noted here that there's no decimal-like concept in ERC-1155, so it isn't possible to implement more precise data for financial applications. And the issue with using `_id` to represent token types is that tokens under the same `_id` are completely fungible. `_id` is a classifying function rather than signifying the non-fungible quality, like `_tokenID` does. As a result, ERC-1155s are efficient in scenarios involving multi-tokens like virtual in-game items but won't be compatible with ERC-721s, nor are they scalable in non-game applications.

A more practical schema for the SFT would be to bring in the `_value`-based, quantitative operations while keeping the `_tokenID`-based non-fungible aspects, with additional innovation on a more generalized classification mechanism like `_slot` for financial scenarios or business purposes. This new

schema is ERC-3525. With ERC-3525, we can enable more in-depth financial applications of SFTs by leveraging an organic balance between the fungible and the non-fungible.

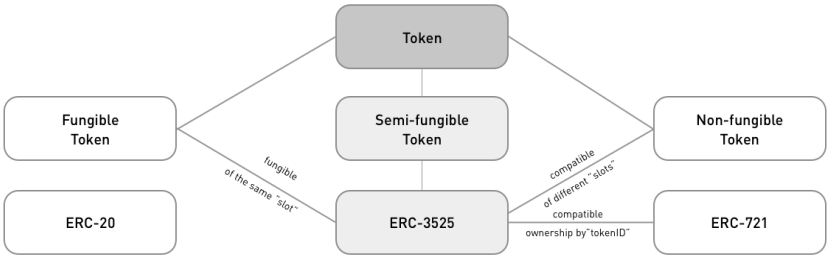

Figure 3.2 Semi-Fungible Token Standard

The ERC-3525 token standard inherits ERC-721's core structure: `_tokenID` and metadata extension. It also adopts the standard quantitative attribute `_value` and extends it with a categorizing attribute `_slot` (along with `Slot Metadata` to implement the categorizing function of `_slot`.) The new data structure within the ERC-3525 is defined as an `<ID, SLOT, VALUE>` Triple Scalar Model.

One of ERC-3525's major innovations on token operations is token(`_tokenId`)-to-token(`_tokenId`) transfer, or quantitative operations for NFTs, implemented by `function transferFrom(uint256 _fromTokenId, uint256 _toTokenId, uint256 _value) external payable` and `function safeTransferFrom(uint256 _fromTokenId, uint256 _toTokenId, uint256 _value, bytes calldata _data) external payable`. This means that ERC-3525 tokens sharing the same `_slot` are fungible to each other and that one ERC-3525 token(`_tokenId`) may be transferred to another. The feature marks the first time in history that an Ethereum token has gained equal *status* as users (externally owned accounts, or EOA) or smart contracts (contract accounts).

Furthermore, ERC-3525 has inherited the traditional address-to-address transfer within the ERC-721 standard, which is implemented by `transferFrom()`, `safeTransferFrom()`.

Figure 3.3 shows a comparison among ERC-20, ERC-721, ERC-1155, and ERC-3525 tokens in terms we've discussed so far.

	ERC-20	ERC-721	ERC-1155	ERC-3525
Fungibility	Fungible	Non-fungible	Non-fungible (Multiple Instances)	Semi-fungible
Data Structure	· `_value` (`balanceOf()`)	· `_tokenID` · metadata	· `_id` · `_value` · metadata	· `_tokenId` · `_value` · `_slot` · metadata
Operation	Transfer to address	Transfer to address	· Transfer to address · Batch Transfer	· Transfer to address · Transfer to token
Use Cases	· Cryptocurrencies · Staking tokens · Governance tokens	· Physical property · Virtual collectibles · *Negative-value* assets	· In-game items	· Token vesting · Convertible debt · Installment purchase agreements

Figure 3.3 A comparison of token standards: ERC-20, ERC-721, ERC-1155 & ERC-3525.

Furthermore, ERC-3525's token data is a two-layer structure that is an upgrade of one-layer data structures found within ERC-20, ERC-721 and ERC-1155 tokens. The `_value` in ERC-20, `_tokenID` in ERC-721, and (`_id`, `_value`) in ERC-1155 are owned by an address. While the `_tokenId` in ERC-3525 is owned also by an address, the `_value` is contained or "owned" by the `_tokenId`.

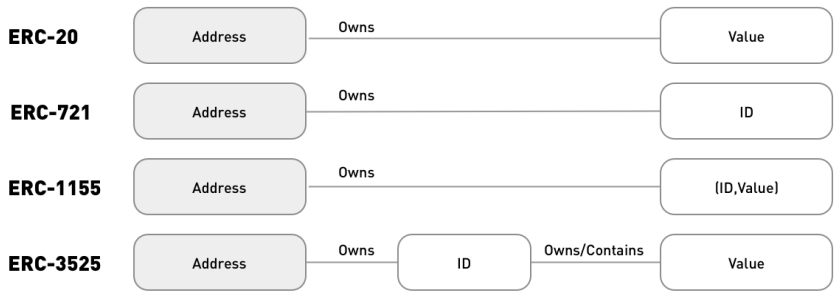

Figure 3.4 The two-layer structure of ERC-3525's token data

3.3 ERC-3525 Core Mechanisms

ERC-3525's greatest innovation is the `<ID, SLOT, VALUE>` Triple Scalar Model. `_slot`/`Slot Metadata` enables quantitative operations for NFTs through the utilization of standard `balanceOf()` / `_value` mechanism.

tokenId

The `_tokenId` in ERC-3525 is defined as a value type in terms of a uint 256, an unsigned integer in the size of 256 bits in Solidity, and is equivalent to the `_tokenID` in ERC-721. `_tokenId` reflects the non-fungible aspect of digital assets.

slot

`_slot` is a new attribute that categorizes variables in financial or business applications. One may implement `_slot` by hashing its underlying properties as a uint-256 abstraction.

A critical tip for implementation here is the uniqueness of the _slot value. As long as different slots have different values, there could be different solutions (such as uint 8).

value

Similar to the `balanceOf()` / `_value` mechanism in ERC-20, `balanceOf()` in ERC-3525 can be used to query the amount of the underlying asset of an ERC-3525 token, and its value type is defined as a Solidity-based uint 256. That means transferring ERC-3525's `_value` to other addresses works similarly to transferring `_value` for ERC-20 tokens.

3.3.1 Token Operations

The address-to-address transfer in ERC-3525 is compatible with that of ERC-721. Please refer to the code sample in 2.1.1 if needed.

What makes ERC-3525 special is its ability to transfer only a fraction of the underlying asset within an ERC-3525 token. Such a transfer is implemented through the token(`_tokenId`)-to-token(`_tokenId`) transfer `function transferFrom(uint256 _fromTokenId, uint256 _toTokenId, uint256 _value) external payable;` and the token(`_tokenId`)-to-address transfer `function transferFrom(uint256 _fromTokenId, address _to, uint256 _value) external payable;`.

As mentioned previously, the token(`_tokenId`)-to-token(`_tokenId`) transfer applies to the transferring of ERC-3525 tokens that have the same `_slot`. This

type of transfer allows users to transfer an amount of the underlying asset within an ERC-3525 token(`_tokenId`) to another:

```
/**
     * @notice Transfer value from a specified token to another specified token with
the same slot.
     * @dev Caller MUST be the current owner, an authorized operator or an operator
who has been
     *  approved the whole `_fromTokenId` or part of it.
     *  MUST revert if `_fromTokenId` or `_toTokenId` is zero token id or does not
exist.
     *  MUST revert if slots of `_fromTokenId` and `_toTokenId` do not match.
     *  MUST revert if `_value` exceeds the balance of `_fromTokenId` or its allowance
to the
     *  operator.
     *  MUST emit `TransferValue` event.
     * @param _fromTokenId The token to transfer value from
     * @param _toTokenId The token to transfer value to
     * @param _value The transferred value
     */
    function transferFrom(uint256 _fromTokenId, uint256 _toTokenId, uint256 _value)
external payable;
```

The token(`_tokenId`)-to-address transfer applies to the transferring of a certain amount of underlying asset within an ERC-3525 token (`_tokenId`) to a different address. The token(`_tokenId`)-to-address transfer allows a new ERC-3525 token with the same `_slot` value as the original token to receive the asset transferred:

```
/**
     * @notice Transfer value from a specified token to an address. The caller should
confirm that
     *  `_to` is capable of receiving ERC-3525 tokens.
     * @dev This function MUST create a new ERC-3525 token with the same slot for
`_to` to receive
     *  the transferred value.
     *  MUST revert if `_fromTokenId` is zero token id or does not exist.
     *  MUST revert if `_to` is zero address.
     *  MUST revert if `_value` exceeds the balance of `_fromTokenId` or its allowance
to the
     *  operator.
     *  MUST emit `Transfer` and `TransferValue` events.
```

```
     * @param _fromTokenId The token to transfer value from
     * @param _to The address to transfer value to
     * @param _value The transferred value
     * @return ID of the new token created for `_to`, which receives the transferred
value
     */
    function transferFrom(uint256 _fromTokenId, address _to, uint256 _value) external
payable returns (uint256);
```

3.3.2 “Slot” Metadata

“Slot” metadata describes the details of the abstracted properties that are hashed as the value of the `_slot` attribute. The value of these properties may be `strings`, `numbers`, `objects`, or `arrays`. Slot metadata can implement various classifications to describe complex DeFi assets. Here’s the JSON schema of slot metadata:

```
{
  "title": "Slot Metadata",
  "type": "object",
  "properties": {
    "name": {
      "type": "string",
      "description": "Identifies the asset category to which this slot represents"
    },
    "description": {
      "type": "string",
      "description": "Describes the asset category to which this slot represents"
    },
    "image": {
      "type": "string",
      "description": "Optional. Either a base64 encoded imgae data or a URI pointing to
a resource with mime type image/* representing the asset category to which this slot
represents."
    },
    "properties": {
      "type": "array",
      "description": "Each item of `properties` SHOULD be organized in object format,
including name, description, value, order (optional), display_type (optional), etc."
      "items": {
        "type": "object",
```

```
        "properties": {
          "name": {
            "type": "string",
            "description": "The name of this property."
          },
          "description": {
            "type": "string",
            "description": "Describes this property."
          }
          "value": {
            "description": "The value of this property, which may be a string or a
number."
          },
          "is_intrinsic": {
            "type": "boolean",
            "description": "According to the definition of `slot`, one of the best
practice to generate the value of a slot is utilizing the `keccak256` algorithm to
calculate the hash value of multi properties. In this scenario, the `properties` field
should contain all the properties that are used to calculate the value of `slot`, and
if a property is used in the calculation, is_intrinsic must be TRUE."
          },
          "order": {
            "type": "integer",
            "description": "Optional, related to the value of is_intrinsic. If
is_intrinsic is TRUE, it must be the order of this property appeared in the calculation
method of the slot."
          },
          "display_type": {
            "type": "string",
            "description": "Optional. Specifies in what form this property should be
displayed."
          }
        }
      }
    }
  }
}
```

To further illustrate the concept of semi-fungible-*ness,* let's think for a second about the PhyloCode, a rule system governing the naming of taxa in biology. Take apples, for example. Under the phylogenetic classifications such as M. domestica species, malus genus, Rosaceae Family, or Eukarya Domain, all apples could be treated as indistinguishable or "fungible." Inspecting individual apples, however, we start to find discrepancies in *secondary properties* like size, shape, color, acidity, etc. It's these qualities that make apples non-fungible. PhyloCode may

deem these qualities negligible for scientific taxonomy, but it's exactly these qualities that dictate most, if not all, of our day-to-day shopping decisions. RWAs (real-world assets) are *fungible* in a given category (e.g., real estate property) but *non-fungible* when taking into account certain secondary qualities (e.g. location).

Returning to ERC-3525, slot metadata enables secondary-quality nuances so that tokens with different `_slot`s are non-fungible, and those with the same `_slot` value are fungible. The slot metadata is what gives the token fluidity in fungibility.

`_slot` also affects how `_value` works: All the NFTs fractionalized from the original NFT can be merged through simple arithmetic addition (or subtraction, if splitting) of their `_value`s.

What `_slot` actually represents in a financial application is arbitrarily defined. In a Vesting Voucher (an SFT that locks and vests tokens based on a predetermined schedule), different vesting schedules are assigned their corresponding `_slot` through the `getslot()` function[14]:

```
function getSlot(
        uint8 claimType_,
        uint64[] memory maturities_,
        uint32[] memory percentages_,
        uint64 term_
    ) internal pure virtual returns (uint256) {
        uint256 first = uint256(
            keccak256(
                abi.encodePacked(
                    claimType_,
                    term_,
                    maturities_[0],
                    percentages_[0]
                )
            )
        );
        if (maturities_.length == 1) {
            return first;
        }

        uint256 second;
        for (uint256 i = 1; i < maturities_.length; i++) {
```

[14] For the source code, please visit https://github.com/solv-finance/solv-v2-voucher/blob/main/packages/solv-voucher/contracts/ICToken.sol

```
            second = uint256(
                keccak256(
                    abi.encodePacked(second, maturities_[i], percentages_[i])
                )
            );
        }
        return uint256(keccak256(abi.encodePacked(first, second)));
    }
```

In this code, `claimType_` represents a certain release rule configuration in the Vesting Voucher: 0 (linear release), 1 (one-time release), and 2 (staged/custom release). Take the linear release, for example. `Term_` represents the vesting period (from the day the vesting starts to the fully vested date) for the Voucher's underlying asset in days. `Maturities_` marks the fully vested date, which is calculated based on the vesting period (e.g., 02:15 on December 5, 2028, UTC). `Percentages_` represents the percentage of remaining, unvested underlying assets. ("100%" in `percentages_` means the tokens are fully vested for a given vesting period.)

Slot metadata is a flexibility-centric parameter that will shape an SFT-based FI and the purposes it will serve.

3.4 A Paradigm Shift in Asset Transfer

ERC-3525's token(`_tokenId`)-to-token(`_tokenId`) transfer transforms the way blockchain-based assets are transferred. To understand why it is important, let's look at an example based on the use of bitcoin.

Imagine Alice and Bob each have $50 in bitcoin in their accounts. After Alice sends $20 to Bob, her remaining balance is $30, and Bob now has $70. Through Bitcoin's *state transition* function, the transfer process looks something like this:

```
APPLY({ Alice: $50, Bob: $50 }, "send $20 from Alice to Bob") = { Alice: $30, Bob: $70
}
```

With the advent of smart contracts, the original account system utilized in Bitcoin has been expanded into EOA (that are controlled by private keys) and contract accounts (that are controlled by their contract codes). An EOA has no code and

can allow anyone to send messages by creating and signing a transaction through itself. A contract account, on the other hand, can read and write to internal storage, send a new message, or even create a new contract when it receives a message from outside itself.

Ethereum's transfer function looks something like this:

```
function transferFrom(address _from, address _to, uint256 _value) public returns (bool success)
```

Since Ethereum has no restriction as to what type of account can transact with the other and what can't, it has built an entire ecosystem supported by asset transfers among EOAs and contract accounts.

ERC-3525 provides a paradigm shift in the asset transfer model, namely, token(`_tokenId`)-to-token(`_tokenId`) transfer. It enables receiving, storing, and transferring of an asset from *tokens*, rather than *addresses*:

```
function transferFrom(uint256 _fromTokenId, uint256 _toTokenId, uint256 _value) external payable
```

In ERC-3525, a token(`_tokenId`)-to-address transfer involves creating and transferring to the recipient's address a new SFT with the same `_slot` as the original. An address-to-token(`_tokenId`) transfer would involve custom programming based on the Ethereum Transaction Simulation.

Transfer model	Human (Address)	Smart Contract	Token (ID/SLOT)
Human (Address)	• BTC • Ethereum • Any blockchain	• Ethereum • Compatible blockchains	• Customized programming
Smart Contract	• Customized programming	• Customized programming	• Customized programming
Token (ID/SLOT)	• ERC-3525 (Simulation)	• ERC-3525 (Simulation)	• ERC-3525 (Standardized)

Figure 3.5 ERC-3525's token(id)-to-token(id) transfer

To sum it up, ERC-3525 SFTs have gained equal importance as EOAs and contract accounts. Like Ethereum smart contracts, ERC-3525 SFTs are capable of external interactions, conditional judgments, and smart executions – the whole nine yards.

3.5 Smart Contract Visualization

ERC-3525 also utilizes metadata extensions which enhance the ways strings, numbers, objects or arrays, or images of ERC-3525 tokens are visually delivered. ERC-3525's metadata's JSON schema is as follows:

```
{
    "title": "Token Metadata",
    "type": "object",
    "properties": {
        "name": {
            "type": "string",
            "description": "Identifies the asset to which this token represents"
        },
        "decimals": {
            "type": "integer",
            "description": "The number of decimal places that the token amount should
display - e.g. 18, means to divide the token amount by 1000000000000000000 to get its
user representation."
        },
        "description": {
            "type": "string",
            "description": "Describes the asset to which this token represents"
        },
        "image": {
            "type": "string",
            "description": "A URI pointing to a resource with mime type image/*
representing the asset to which this token represents. Consider making any images at a
width between 320 and 1080 pixels and aspect ratio between 1.91:1 and 4:5 inclusive."
        },
        "properties": {
            "type": "object",
            "description": "Arbitrary properties. Values may be strings, numbers,
objects, or arrays."
        }
    }
}
```

The text-rich description helps provide sophisticated information about a token, and the information-rich image in turn enhances the token's text description.

Slot metadata not only provides the details of the abstracted properties that are hashed into `slot` but also serves as the foundation of quantitative operations for SFTs, like splitting and merging.

Contract Metadata describes the smart contract in terms of name, text description, unitDecimals, etc. Its JSON schema is as follows:

```
{
    "title": "Contract Metadata",
    "type": "object",
    "properties": {
        "name": {
            "type": "string",
            "description": "Contract Name"
        },
        "description": {
            "type": "string",
            "description": "Describes the contract "
        },
         "unitDecimals": {
            "type": "integer",
            "description": "The number of decimal places that the units should display
- e.g. 18, means to divide the token units by 1000000000000000000 to get its user
representation."
        },
        "properties": {
            "type": "object",
            "description": "Arbitrary properties. Values may be strings, numbers,
object or arrays."
        }
    }
}
```

3.6 The Roadmap

October 2020: Inception of an ERC-721 compatible token standard in response to the absence of sophisticated financial instruments in DeFi
December 1, 2020: Created ERC-3525 proposal
April 2021: The first field-tested SFT product: Vesting Voucher (introduced by Solv Protocol)
July 2021: Submission of the first version of ERC-3525
October 2021: "Draft" stage
April 2022: Added token-to-token transfer
September 5, 2022: Final ERC

To learn more about ERC-3525, please visit Ethereum's GitHub repository at https://github.com/ethereum/EIPs/blob/master/EIPS/eip-3525.md.

To learn more about the EIP process, please visit https://github.com/ethereum/EIPs/blob/master/EIPS/eip-1.md.

Chapter 4

Semi-Fungible Tokens in Use

In this chapter, we will explore the real-world use cases for the SFT through several field-tested products developed by Solv Protocol, a protocol that provides a starter kit and a marketplace for Web3 advanced digital assets. These products are the Vesting Voucher, Convertible Voucher, and the Bond Voucher.

We'll begin with four design guidelines for these products (used by Solv) and then explore the ways in which we can leverage the SFT in scenarios like fundraising, community-building, or treasury diversification.

4.1 Fundamental Design Guidelines

Generally speaking, designing an SFT product revolves around four fundamental guidelines, or design dimensions: **conditional lockup and vesting**, **embedded promise to repay**, **payment-splitting**, and **asset container**. They are the

underlying mechanistic principles for all existing SFT products, and are key to understanding the shape into which future SFTs will come.

Conditional locks

Sophisticated financial operations never stop at sending assets. For stock or token vesting, we need to think holistically about a product that conditionally delivers based on the vesting schedule, timestamp, interest rate (if applicable), and real-time trading price of a given asset.

Because an SFT can embed time locks and can be fractionalized, it is possible to divide a token into multiple “daughter” tokens while having them subject to the same set of rules for their underlying asset. A crypto project can leverage this feature to incentivize hundreds of thousands of users at once.

Embedded promise to repay

The crux of a credit market is the borrower’s promise to repay the debt. In TradFi, this promise exists as a copy of a signed contract. Smart contracts have replaced such traditional contracts, by which all borrowing activities are decentralized.

An SFT-powered debt can embed a promise to repay inside a token, so the lenders can easily access it as an BAYC owner inspects the art. This feature is widely used by borrowers such as Strips Finance and iZUMi Finance, who want to enhance the credit profile of their bond issuance.

Multiparty clearance

An SFT allows the issuer to make one-time payment to multiple recipients on a pro-rata basis.

Asset container

An SFT can receive, store, and send digital assets to another SFT, which is made possible by the IPO (“input-process-output”) method. As opposed to traditional bank accounts where financial transactions are recorded between the bank and a customer (depositor), with SFTs, users can transact directly from one token to another in a decentralized way.

4.2 Solv-Powered SFTs: Vouchers

4.2.1 Vesting Voucher

The Vesting Voucher is a popular token vesting SFT to manage and distribute allocations (an allocation is an allotment of tokens). A Vesting Voucher can lock and vest any crypto assets according to a predefined schedule. Users can customize the vesting schedule (either immediate, linear, or customized) in a no-code front end.

Vesting Vouchers are suitable for projects that want to conduct token sales that will have no direct impact on the token value or those that want to use token locks to bootstrap a community.

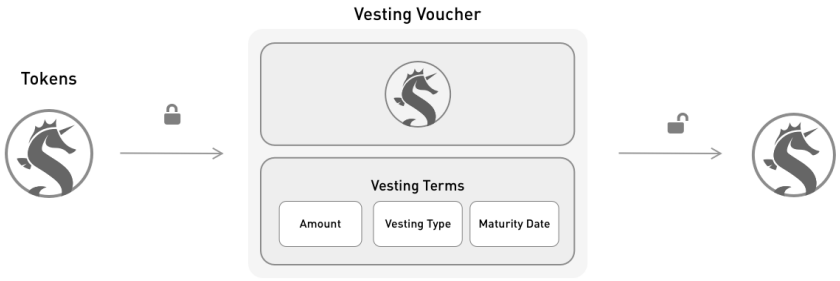

Figure 4.1 The process of locking up and vesting digital assets via a Vesting Voucher

The Vesting Voucher #240 (figure 4.2) is scheduled to linearly vest 50,000 $SOLV tokens (the underlying asset) over 180 days, from 2022-09-30 to 2023-03-29.

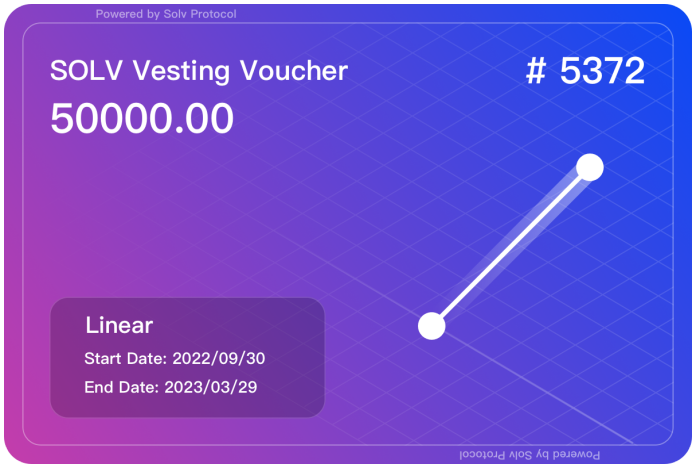

Figure 4.2 Vesting Voucher #240

4.2.2 Bond Voucher

The Bond Voucher is an SFT debt that allows DAOs or institutions to raise capital by issuing a convertible bond, after removing the embedded convertibility, a pure debt. It offers the lender consistent incremental fundings to accelerate growth and cover operational expenses.

The Bond Voucher is issued at a discount to face value. At maturity, the lender burns the Voucher and receives the full face value profit. If the market price of the underlying token exceeds a predetermined price (conversion price), then the lender receives both the face value and the upside of the token.

Figure 4.3 Bond Voucher #15 (sample)

The Bond Voucher in figure 4.3 has a few defining components as follows:

- **Face value**: the amount for which the bond can be redeemed at maturity, denominated in any currency.
- **Maturity date**: the date on which the bond will mature and on which the face value can be redeemed.
- **Conversion price**: the predetermined price at which the bond will convert into an amount of underlying asset at maturity. The lender can exercise the convertibility only if the price of the underlying asset is greater than or equal to the conversion price.

4.2.3 Convertible Voucher

Another member of Solv Protocol's SFT family is Convertible Voucher, a structured product with a similar convertibility feature to that of the Bond Voucher, except that a Convertible Voucher has a flexible payout mechanism based on future market price of the underlying asset and a predefined price range. It allows DAOs or institutions to borrow capital by backing the Voucher with sufficient collateral, it allows lenders to earn a return on a promising asset in an unstable market condition.

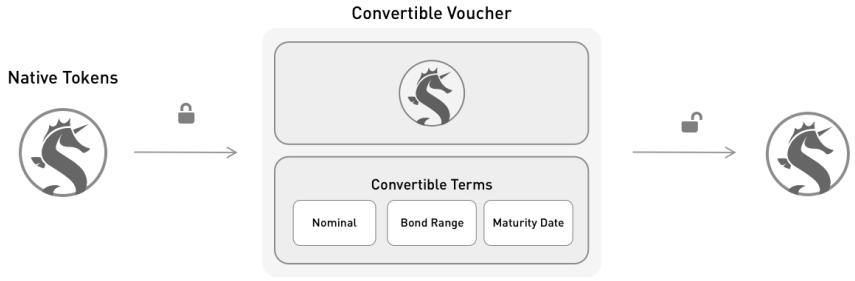

Figure 4.4 The process of investing in a Convertible Voucher

The Convertible Voucher #1 (figure 4.5) is scheduled to mature on December 31, 2022, has a face value of 10,000 USDT, and is secured by $SOLV tokens. At the issuance, the issuer defines a price range between $5.00 and $10.00. This means if the market price of $SOLV lands between the range the issuer (borrower) pays the lender a full face value profit of 10,000 USDT. Outside the range, the borrower pays the full amount of 1,000 (if above the range) or 2,000 $SOLV (if below the range).

Figure 4.5 Convertible Voucher #1

4.3 Use Cases for Vouchers

4.3.1 Initial Voucher Offering (IVO)

Initial Voucher Offering (IVO) provides a one-stop solution to issue a Vesting Voucher, Convertible Vouchers or a Bond Voucher in a primary market open to all investors. The IVO enables incentive alignment between a crypto project and sophisticated DeFi users and rapid and permissionless fundraising processes.

In a typical Vesting Voucher IVO, an early project can conduct token sales to retail investors by minting and issuing a Vesting Voucher with native tokens nested within on the Solv platform. The advantage of launching a Vesting Voucher IVO is the launch will have no immediate impact on the token's value since locks do not affect a token's circulating supply. The IVO also allows retail investors to invest in an early project on a level playing field with institutional investors as they wouldn't be able to in a traditional primary market.

An early crypto project usually has financing-related anxieties after the initial financing rounds and before the token launch:

- No way to incentivize early users besides airdrops
- Post-IDO sell pressure
- Difficulty to single out real users from mercenary traders
- Hefty marketing expenses

- Workload involved in planning an IDO
- No way to effectively pitch top-tier institutional investors

The Initial Voucher Offering (IVO) offers a one-stop solution to these pain points and is more effective than the IDO at attracting sophisticated DeFi users.

In the next section, we'll present previous Voucher offerings to further your understanding. If you are already comfortable with the concept of IVO, feel free to skip ahead!

$CLH Vesting Voucher Offering by Clear

In January 2022, Clear, a decentralized platform for creating custom derivatives products, announced the adoption of the Vesting Voucher for the $CLH distribution management. Existing holders of $CLH tokens were on track to receive $CLH Vesting Vouchers on a pro-rata basis. Owners of $CLH Vesting Vouchers can freely trade their Vouchers at a secondary Voucher market at Solv Protocol without diluting the spot $CLH already circulating.

$SOLV Vesting Voucher Offering by Solv Protocol (pre-IDO)

On December 13, 2022, Solv Protocol, a decentralized marketplace for tailor-made Web3 financial assets, sold 1 million $SOLV tokens at 0.7 USDT via $SOLV Vesting Vouchers listed on the BNB chain and Ethereum mainnet. $SOLV Vesting Vouchers sold in this offering are customized to linearly vest $SOLV tokens over the period of 6 months beginning on its IDO day (pending) and have allowed the protocol to raise 700,000 USDT successfully within an hour. 800 unique addresses participated in the $SOLV Vesting Voucher offering grew to 1,800 in the secondary market before the sellout, with the token's price surging from 0.7 USDT to 3.5 USDT.

As we are writing this, the total transaction volume of the $SOLV Vesting Voucher has increased from US$700,000 at the end of the offering to approximately US$4 million, with the unique holders having increased from 800 to 7,014.

$DOP Vesting Voucher Offering by Drops

On March 25, 2022, Drops, an NFT lending platform, offered a $DOP Vesting Voucher of US$300,000 in value with a 10% discount on the spot $DOP. $DOP Vesting Voucher was scheduled to be gradually vested over 14 days and was 53%

sold after 48 hours of the launch. To incentivize the community, Drops airdropped whitelisted users a 5% of the Voucher's value in $DOP as a bonus.

$PERP Convertible Voucher Offering by Perpetual Protocol

On February 18, 2022, Perpetual Protocol, an on-chain perpetual futures DEX, issued a $PERP Convertible Voucher worth US$3 million, secured by $PERP tokens.

$STRP Bond Voucher Offering by Strips Finance

On April 12, 2022, Strips Finance, the leading interest rate derivatives exchange, issued a US$2-million STRP Bond Voucher with a 90-day maturity, making Strips the first protocol that issued an on-chain bond. The Voucher's 13% APR was structured to distribute in three batches: the first 8% in $STRP and $SOLV tokens right after the sale and the last 5% in USDC paid at maturity.

4.3.2 Treasury Management

A treasury is the pillar of a DAO, without which product development, business operations, marketing campaigns, community-building, and protocol-level investment would not be possible. A treasury's fund comes from protocol revenues and other non-operating income. Most projects, however, can't generate sufficient revenue to cover their working capital and usually have to resort to shrewdly selling native tokens or borrowing.

In a bear market, DAOs would conduct treasury sales at fire-sale prices, inducing panic in the market. Borrowing on lending platforms (such as Compound or Aave) is viable only for DAOs whose native tokens are liquid enough to be used as collateral. Further, these platforms need sufficient collateral (which may trigger negative market sentiment about the token) and threaten the borrower with liquidation risks.

An SFT-powered DeFi derivative provides a solution to DAOs who like liquidation risk, flexible collateral, and open market access. An option-embedded SFT also allows the lender (investor) with a unique risk-reward profile to express a specific market view.

There are two types of derivatives products in DeFi: debt derivatives and equity derivatives, either of which can be used under different circumstances to manage or diversify a treasury. Equity derivatives such as the Vesting Voucher function similarly to traditional company stocks but differ from stocks in that they offer the issuer far more optionality than raising capital through stocks. While projects raising capital with equity derivatives have no repayment obligation, using equity derivatives involves token sales at steep discounts, which may induce panic in the market.

A debt derivative, like the Bond Voucher, is a derivative that allows a protocol to take on a debt without selling their native tokens. With debt derivatives, there is no immediate price dilution (which is common in token sales), no lower liquidation risk, and there's an option to use non-liquid assets as collateral.

Chapter 5

The Outlook of SFTs

Through this book, we've demonstrated various ways in which an SFT can represent almost any financial instrument or asset. In this chapter, we'll conclude this book by unleashing our imaginations and considering a few alternative (non-standard) assets that could be built upon the ERC-3525 infrastructure, and how they can benefit our lives.

5.1 Portfolio Allocations and Structured Finance

5.1.1 A Background

During the 1950s to 1970s, three major innovations revolutionized financial economics. Firstly, there was Modern Portfolio Theory (MPT), which was presented in Portfolio Selection (1952) by American economist Harry Markowitz. According to Markowitz, investors can achieve the best investment results by balancing high-risk, high-return investments with low-risk, low-return investments in accordance with their risk tolerance. A second innovation was the Black-Scholes formula, which was discovered in 1973 by Fischer Black, Myron Scholes, and Robert C. Merton. The third innovation was the emergence of

mortgage-backed securities pioneered by Lewis Ranieri during his tenure at Salomon Brothers. Though later criticized for his role in the 2008 housing crisis, Ranieri was attributed to the innovative practice of *securitization*, a practice of pooling various financial assets and selling their cash flows as securities, and the boom of the mortgage-backed security market in the 1980s, which reached US$150 billion in 1986.

This section explores two upcoming SFT products – Package SFT and Tranche SFT – that draw inspiration from these financial innovations. Historically, their TradFi counterparts - portfolios and structured products - have proved extremely popular with investors.

5.1.2 Package SFT

A Package SFT is an upcoming ERC-3525 powered asset packaging solution that will enable various assets to be packaged into a single SFT, resembling a traditional investment portfolio. The Package SFT provides the look-through and fund-in-fund transparency for any underlying assets, and it is fractionalizable for liquidity management.

Furthermore, the immutability of data held in the digital ledger (the blockchain) and the excellent visualization of ERC-3525 will enhance what regulators desire to achieve - clarity and protection for investors.

5.1.3 Tranche SFT

A Tranche SFT is an ABS (asset-backed security) product that slices the underlying assets' income into tranches. Using tranches in TradFi, an individual can create one or more security classes whose rating is higher than the average rating of the underlying asset pool, allowing for portfolio diversification.

To redistribute the repayment risk and credit risk efficiently among various tranches, a Tranche-SFT will have two tranches: credit tranching and time tranching. In credit tranching, all credit losses are first absorbed by the equity tranche, then by the mezzanine tranche, and finally by the senior tranche. Through time tranching, repayment risk – the risk that borrowers may default by repaying the principal early to take advantage of interest rate movements – is redistributed. Portfolio SFTs can also be combined with Tranche SFTs to generate unusual risk-reward profiles.

Transparency and information symmetry are the blockchain's major contributions to finance. ERC-3525 ensures consistent reporting standards for loan-origination data and portfolio performance, as well as full transparency into on-chain securitization.

5.2 The Derivation of Rights

Thanks to the Bundle of Rights, it is not uncommon these days to use a piece of property as collateral without stripping the owner of other rights over the property. In the blockchain, there's still an absence of a system for separating various rights contained in the Bundle of Rights in order to maximize the utility of a property.

A Derive SFT represents a single or multiple rights *derived* from the Bundle of Rights with a property. A person may secure a loan with a Derive SFT that represents the right to solely to dispose of his or her StepN sneaker, not to use it to run.

A Installment SFT is a two-part tokenized installment contract to finance asset acquisitions. With it, a buyer has the option to make a series of partial payments for an item over time and receive the full right to use it (through a Utility Voucher) right after making a down payment. All the while, the seller of the item keeps a Ownership Voucher, enabling him or her to reserve the right to revoke the Utility Voucher from the buyer should a default happens.

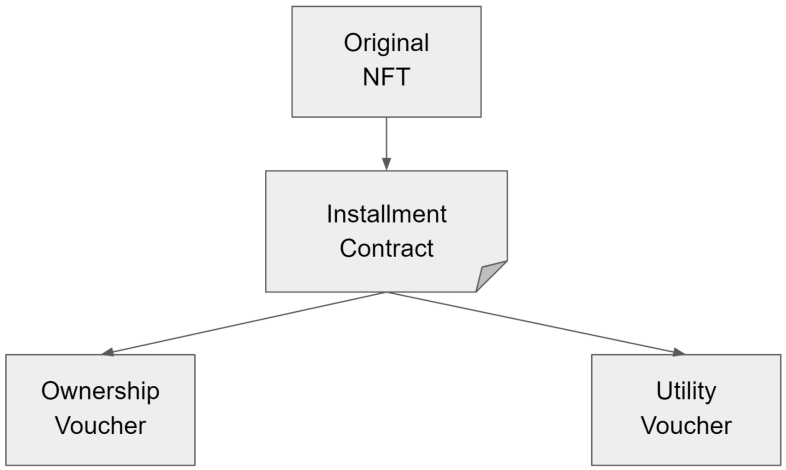

Figure 5.1 Installment SFT

The Derive SFT can allow a user to:

- Fractionalize and sell the right to derive an income from an asset
- Exhibit an art NFT without really owning it
- Use an game item as collateral without undermining its performance in the game
- Package multiple Derive SFTs into an asset-backed security (ABS) to generate an income or transfer its future income to others.

5.3 Real World Assets (RWAs)

Before the blockchain, the digital profile of an RWA would be stored in a centralized database. In the past decade, the tokenization of RWAs has been stagnant, because a powerful and versatile digital representation for RWAs was missing. With the superior descriptiveness for real-time data and static or animated images and the liquidity features of an ERC-20 token, ERC-3525 fills the gap.

The key promise of the blockchain technology is to bridge real-world assets from the traditional world to the blockchain world, and doing so has many advantages:

Global access

The Ethereum blockchain that we access is borderless, permissionless and open 24/7. Physical assets that were previously unreachable by certain regional buyers are now made accessible through RWA tokenization.

Programmability

Any digital tokens on Ethereum are programmable. This means they can be made to behave in a certain way and create certain outputs based on various inputs.

Ownership protection

Since an RWA is tokenized and onboarded to the blockchain, its ownership is immutable unless the owner verifies a change. This is different from historical reliance on a centralized organization for custody and, as a result, can reduce ethical malpractices and credit-related risks.

Transparency

Since transaction records on the blockchain are immutable, the owner of a tokenized RWA can't change the asset's history to make it appear attractive. This allows investors to access the entire history of an asset to make more informed decisions.

5.4 More Possibilities of ERC-3525

ERC-3525 is a general-purpose, versatile, and highly scalable token standard whose applications should not be limited to just ownership of financial instruments or assets. Here are some further scenarios any developer with the proper coding skill can easily materialize with the ERC-3525 Semi-Fungible Token:

- Create a SoulBound Token (SBT) that leverages the token-to-token payment and quantitative features of ERC-3525 for identity management
- Issue a loyalty card, rewards card, points card, or club card on-chain
- Tokenize a Web2 or Web3 digital wallet, fractionalize, and transfer the wallet to others
- Create a virtual item, game item, e-book, or an on-chain magazine

ERC-3525 Semi-Fungible Token empowers everyone to tap into a multi-trillion dollar market through the DeFi, Web3, and the blockchain technology. And if you don't care about any of that, know that now we have the power to upgrade our quality of living by mapping the right symbol of ownership to the things we find useful. It's that easy.

References

1. Will Wang, Mike Meng, Ethan Y. Tsai, Ryan Chow, Zhongxin Wu, AlvisDu, "EIP-3525: Semi-Fungible Token," Ethereum Improvement Proposals, no. 3525, December 2020. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-3525.
2. Token Safe Harbor Proposal 2.0, https://www.sec.gov/news/public-statement/peirce-statement-token-safe-harbor-proposal-2.0
3. Uniswap V3 Whitepaper, https://uniswap.org/whitepaper-V3.pdf
4. Uniswap V3–Is NFT the only way to upgrade DeFi?, https://medium.com/solv-blog/solv-talk-uniswap-V3-nft-is-the-only-way-to-upgrade-defi-2ed2686bf1a3
5. Centrifuge Whitepaper, https://docs.centrifuge.io/getting-started/p2p-protocol/
6. ve(3,3): An Introduction into the New and Ambitious Solidly Exchange, https://avgjoescrypto.substack.com/p/ve33-an-introduction-into-the-new
7. Fuji DAO, https://medium.com/fuji-finance/nfts-and-defi-innovation-b05f49b81831
8. Alpha Homora V2: ERC-1155 Tokenized Positions, https://alphaventuredao.io/blog/alphahomorav2-tokenized-position
9. William Entriken, Dieter Shirley, Jacob Evans, Nastassia Sachs, "EIP-721: Non-Fungible Token Standard," Ethereum Improvement Proposals, no. 721, January 2018. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-721.
10. Witek Radomski, Andrew Cooke, Philippe Castonguay, James Therien, Eric Binet, Ronan Sandford, "EIP-1155: Multi Token Standard," Ethereum Improvement Proposals, no. 1155, June 2018. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-1155.
11. Financial Instrument, https://en.wikipedia.org/wiki/Financial_instrument
12. Ethereum Whitepaper, https://ethereum.org/en/whitepaper/
13. Fabian Vogelsteller, Vitalik Buterin, "EIP-20: Token Standard," Ethereum Improvement Proposals, no. 20, November 2015. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-20.

Remark

All credits for this book go to SFT Labs. Kudos to the SFT Labs team for making SFTs accessible!

Authors: Yan Meng, Will Wang, Ryan Chow, Hugh Hu, John Chak, Zhi Li, Samuel Lin
Editor: Ethean Yu
Advisors: Jemma Xu, Ethan Tsai, Ming Zhang

September 2022