# Unitas: A Decentralized, Exogenously Over-Reserved, USD-Denominated *Unitized Stablecoin* Protocol for Emerging Markets

V1 Whitepaper

https://unitas.foundation


by

Wayne Huang, Winston Hsiao, Aditya Gupta, Andrew Chen, Sun Huang

Telegram | Twitter | Email


Feb 14, 2023 (ver. 1.01)

(For reading convenience, we've translated this paper into 繁體中文 , 简体中文, 日本語, 한글, and Español. All latest versions can be found here.)

# 1 Preface

Unitas introduces the first *unitized stablecoin* protocol for multiple *emerging market currencies (EMCs, e.g., Indian Rupee INR)*. **A *unitized stablecoin* uses USD to express the "value" of 1 unit of an EMC**. Unitas aims to solve the problems associated with doing business in EMCs, and free emerging countries' markets, businesses, and entrepreneurs to pursue their financial potential.

Unitas' mission grew out of our own experiences in India.

Aditya grew up in a small Indian town and graduated from Indian Institute of Technology (IIT) Bombay. Winston attended high school and university in Pune (Savitribai Phule Pune University) and then started his career in India. Wayne grew up in Taiwan and the United States but conducted extensive business with India; his startup Armorize Technologies applied formal methods to identify vulnerabilities in source code, and India was one of the company's main markets.

We all understood the challenges posed by an EMC such as the Indian Rupee – USD liquidity was poor, banks were inefficient and unfriendly to foreigners, some banks were known to confiscate client money, and the regime lacked regulatory clarity.

When we studied Satoshi Nakamoto's Bitcoin whitepaper in 2012, we immediately recognized that one of blockchain's best applications was solving EMC banking and cross-border payment challenges.

We were inspired by USDT's adoption in 2015 and Dai's in 2019. Emerging markets have desperately needed a DeFi stablecoin protocol, and so Winston began to design Unitas in 2019.

Unitas is a decentralized EMC *unitized stablecoin* protocol. It mints unitized stablecoins whose value is pegged to an EMC, but are over-reserved by USD-pegged stablecoins (such as USDC, Dai, USDT, BUSD, Frax, etc.). The Unitas protocol guarantees unrestricted and unconditional conversion of its unitized stablecoins "back" to USD-pegged stablecoins. Unitas levels the playing field for emerging market businesses and entrepreneurs by elevating their financial sovereignty and allowing them the freedom to participate in global financial markets. It unleashes many emerging markets' potential by facilitating foreign investment, cross-border payment, global market access, DeFi participation, and more.

"Unitas" was the codename for Harry White's proposal [1] at the 1944 Bretton Woods conference, which gave birth to today's International Monetary Fund (IMF). White was the economist representing the US. Parts of this project resemble his "Unitas," since our goal is to create a DeFi "translator" that can unconditionally convert an emerging market stablecoin back to a USD-pegged stablecoin.

We sincerely invite you to join us if you care for inclusive financial access in emerging markets. Let's accelerate emerging markets by using decentralized blockchain protocols.

# 2 Overview

Unitas aims to create decentralized, exogenously over-reserved *unitized stablecoins* for multiple *emerging market currencies (EMCs, e.g., Indian Rupee INR)*.

**A *unitized stablecoin* uses USD to express the "value" of 1 unit of an EMC**. Let's understand using a simple example: today, 1 INR is worth approximately 1/78 USD. Therefore, if we divide 1 USD into 78 equal units, although each unit is still U.S. Dollar (1/78 USD), each one's "value" will equal to 1 INR.

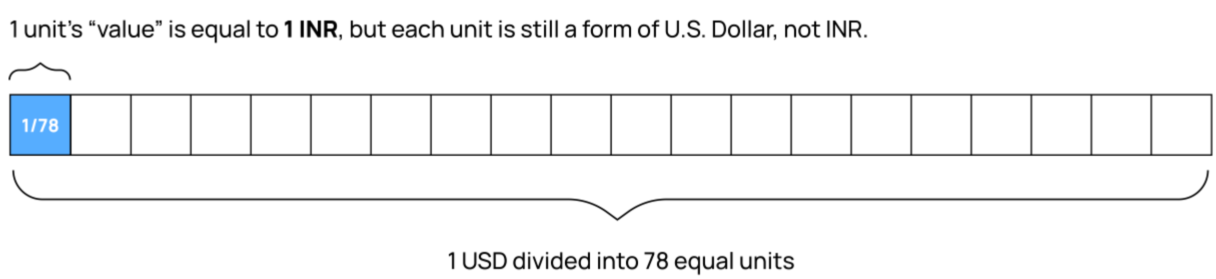


*Figure 1: A unitized stablecoin uses USD to express the "value" of 1 unit of an EMC.*

Similarly in the Unitas protocol, 1 unit of Unitas' Indian Rupee Unit (which we call *USD91*), is also 1/78 USD. Therefore, although 1 USD91 is simply 1/78 USD, its "value" is equivalent to 1 INR.

In this paper, we'll use the notion *USDPEG* to represent any USD-pegged stablecoin such as Dai [2], USDC [3], FRAX [4], USDT [5], BUSD [6], USDP [7], etc. A Unitas stablecoin is exogenously over-reserved by a basket of USDPEGs.

Unlike existing stablecoin protocols, **Unitas is just a simple "unit of account" protocol. Holding Unitas unitized stablecoins is equivalent to holding USDPEGs**, but the value held is constantly pegged to an EMC.

It's extremely important to stress that a Unitas unitized stablecoin:

a) is not an EMC-fiat-reserved stablecoin, and
b) does not guarantee redemption back to an EMC fiat currency.

In contrast, a Unitas unitized stablecoin:

a) is an exogenously over-reserved with USDPEGs, and
b) guarantees redemption for a USDPEG; **therefore, holding 1 Unitas unitized stablecoin (e.g., USD91) is equivalent to holding a constantly-changing amount of USDPEGs, whose value is pegged to 1 unit of a certain EMC (e.g., INR)**.

| Time | Unitized Stablecoin | Redeemable for |
|---|---|---|
| Apr 1, 2021 | 1 USD91 | 1/73.32 USDPEG |
| Jul 1, 2021 | 1 USD91 | 1/74.56 USDPEG |
| Oct 1, 2021 | 1 USD91 | 1/72.99 USDPEG |
| Jan 1, 2022 | 1 USD91 | 1/74.37 USDPEG |
| Apr 1, 2022 | 1 USD91 | 1/75.97 USDPEG |
| Jul 1, 2022 | 1 USD91 | 1/78.93 USDPEG |
| Oct 1, 2022 | 1 USD91 | 1/81.49 USDPEG |

*Figure 2: Holding Unitas unitized stablecoins is equivalent to holding USDPEGs.*

Exogenously over-reserved by the same basket of USDPEGs, the Unitas protocol provides decentralized conversions between its USD-pegged stablecoin USD1 and its EMC stablecoins (e.g., USD91 pegged to Indian INR, USD55 pegged to Brazilian BRL, USD52 pegged to Mexican MXN, USD90 pegged to Turkish TRY, USD234 pegged to Nigerian NGN, and USD54 pegged to Argentinian ARS). It also offers swaps between its own USD-pegged stablecoin USD1 and its supported USDPEGs.

"Money" serves three major roles – unit of account, medium of exchange, and store of value. Because a) Unitas is over-reserved with USDPEGs, and b) major USDPEGs are reserved with bank deposits and treasury bills, therefore in Unitas view:

1) Unitas unitized stablecoins serve as units of account, making it easy to price trades in EMC units,
2) USDPEGs (e.g., USDC, USDT, BUSD) serve as mediums of exchange, facilitating value transfers across blockchain wallets, and
3) commercial bank deposits, treasury bills, CBDCs, and tokenized real-world assets serve as (short-term) stores of value, providing stable reserves for USDPEGs.

| Role | Asset type |
|---|---|
| Units of account | USD_EMCs (e.g., USD91, USD55) |
| Mediums of exchange | USDPEGs (e.g., USDC, USDT, BUSD) |
| Stores of value | Bank deposits, treasury bills, CBDCs, tokenized real-world assets |

*Figure 3: Different asset types and their roles in Unitas' view.*

Unitas is specially designed for fiat currencies that: a) have shown a consistent depreciation against the USD [8] yet b) represent sizable GDPs. For example, over the past 10 years, the following currencies have steadily depreciated against the USD, yet represent high world-GDP rankings: Indian Rupee (-65.5%, #6), Brazilian Real (-304%, #12), Mexican Peso (-161%, #15), Turkish Lira (-765%, #20), Nigerian Naira (-263%, #28), and Argentinian Peso (-2444%, #31).

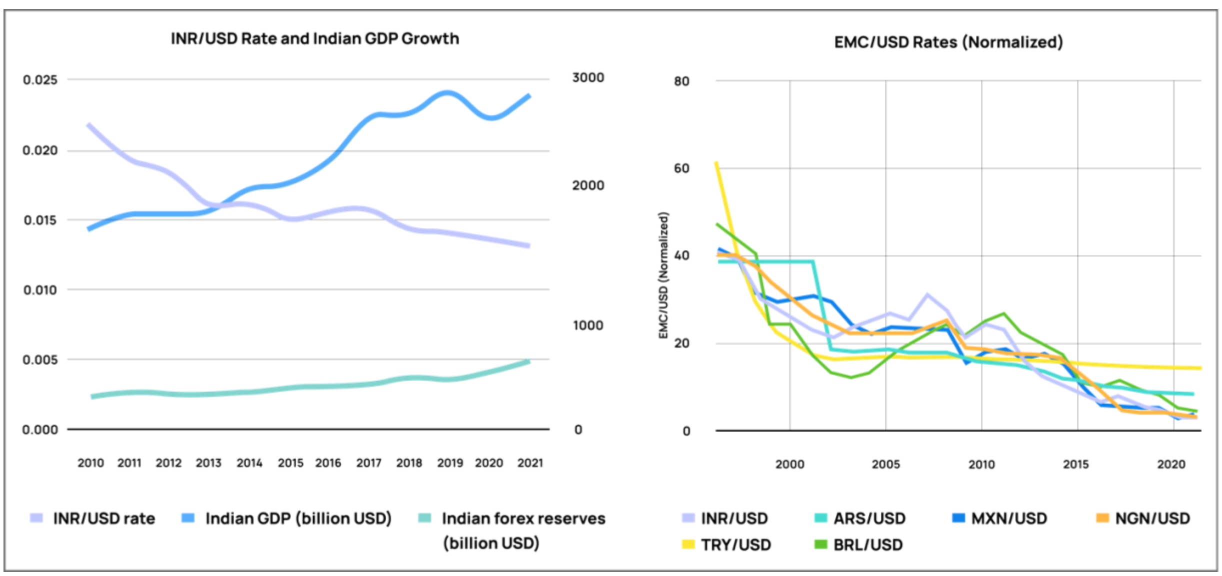


*Figure 4: Indian INR/USD, GDP, and forex reserves (sources: IMF [9], Indian Government [10]).*

By introducing the first stablecoins for multiple EMCs, Unitas' mission is to accelerate financial inclusion in emerging markets.

# 3 Problems Unitas Solves and Unique Challenges

Emerging market currencies (hereon referred to as "EMCs") often pose the following challenges to foreign businesses and market participants:

## 3.1 Dollar Liquidity

With today's global financial products and services mostly priced in USD, USD liquidity is crucial for emerging market participants. While it's easy to convert USD into an EMC, it's often challenging to convert an EMC back to USD [11] [12] [13] [14]. Causes are systemic and structural to emerging market economies and will take a long time to improve.

Unitas provides unconditional and unrestricted convertibility from a USD_EMC stablecoin such as USD91 back to USD1, and then from USD1 back to a USDPEG (e.g., Dai, USDC, USDT, etc. See *7.2.1 Unconditional Exit and Conditional Minting*).

## 3.2 Custodial Risk

It's much harder to find banks in emerging markets that rank world-top in terms of safety. It's not unusual for emerging market banks to confiscate client deposits [15] [16].

In contrast, Unitas stablecoins, whose value is pegged 1:1 to EMCs, can be self-kept in any blockchain wallet.

## 3.3 Capital and Market Efficiency

EMC banking and transfers can be extremely slow, cumbersome, and expensive in cross-border business, leading to poor capital efficiency.

In cryptocurrency markets, EMC-denominated trading pairs such as BTC/INR are often inefficient and have shallow liquidity. It's also common to see substantial price differences at different centralized exchanges.

Market makers of pairs such as BTC/INR cannot introduce good liquidity because:

a) EMCs such as INR suffer from 1) extremely slow bank transfers and 2) minimal liquidity when converting from INR to USD (or USDT); therefore
b) market makers cannot tap into major global liquidity pools such as BTC/USDC and BTC/USDT.

Unitas' stablecoins will enable CEXs and DEXs to list /USD_EMC trading pairs such as BTC/USD91, ETH/USD91, and USDPEG/USD91 (e.g., USDT/USD91, USDC/USD91).

Compared to /INR pairs (e.g., BTC/INR), /USD91 pairs (e.g., BTC/USD91) will have much deeper liquidity because:

a) USD91 transfers much faster than INR (because it's crypto), and
b) USD91 carries built-in conversion to a USDPEG.

/USD91 pairs allow EMC market makers to tap into global liquidity and DeFi markets. This increases capital and market efficiency for cross-border businesses and cryptocurrency traders.

## 3.4 Unique Challenges

We're able to design and implement Unitas thanks to what we've learned from designs and field experiences of existing stablecoin protocols such as Dai [2], FRAX [4], Angle Protocol [17], and many others. Focused on pegging to EMCs, Unitas faces unique challenges requiring innovation.

### 3.4.1 The First Stablecoin Pegging Challenge

Dai has successfully leveraged the existence of well-pegged (centralized) stablecoins such as USDC, USDT, and USDP to stabilize its own peg:

a) at the time of writing, USDC and USDP respectively make up 51.9% and 5.5% of Dai's overall collateral [18],

b) Dai’s *Peg Stability Module (PSM)* [19] allows for 1:1 swaps between Dai and USDC, and

c) the 3pool offering swaps between Dai, USDT, and USDC is one of the largest pools on Curve and provides over $1 billion worth of liquidity.


Whereas point (a) minimizes collateral value volatility, points (b) and (c) bolster Dai’s peg and introduce deeper liquidity.

In a similar effort to (a) and (b), Frax also heavily relied on USDC as its reserve when it bootstrapped.

In contrast, Unitas may not have such leverage because it will often be introducing the first stablecoin for an EMC (e.g., INR, BRL).

### 3.4.2 Inaccessibility of Forex Rates in Emerging Markets

We estimate that a significant percentage of USD_EMC/EMC and USDPEG/USD_EMC (e.g., USDT/USD91) swaps will take place offline in a peer-to-peer or OTC fashion (see *5.1 Unitas Stablecoin Users*, *Table 1* and *Figure 5*). Creating feedback loops that provide the protocol visibility into these swaps' exchange rates will be challenging.

Similarly, it’s also difficult for the protocol to know an EMC’s current exchange rate against the USD (e.g., USD/INR). While banks and central banks publish rates online, they are often not the people’s primary sources of USD liquidity. Instead, people’s main liquidity sources may be informal money transfer systems (IMTSs) [20] or informal value transfer systems (IVTSs) [21].

Unitas therefore incorporates multiple designs to access actual daily market USD/EMC rates reliably (see *10 USD_EMC Price Pegging*).

Unitas’ unique pegging challenge is in sharp contrast to those of:

a) MakerDAO, which uses ETH as one of its primary collaterals, and has leveraged the many ETH/USD online order books offered by major exchanges to build a robust price oracle, and

b) Frax, which used USDC as its primary reserve during bootstrapping, and has achieved similar results via Chainlink [22] [23] by stacking ETH/USD and ETH/FRAX order books to infer FRAX/USD rates.

Unitas incorporates novel designs to overcome these challenges (see *10 USD_EMC Price Pegging*).

# 4 The Conversion Experience

For simplicity, this paper will use the Indian INR to represent an EMC, and USDPEG to represent a USD-pegged stablecoin (e.g., Dai, USDC, Frax, USDT).

To enter the Unitas ecosystem, a user must first use 1 USDPEG to mint 1 USD1, where USD1 is Unitas' USD stablecoin exogenously over-reserved by a basket of USDPEGs.

Using 1 USD1, a user can mint out a certain amount (say 78) of USD91, where USD91 is Unitas' stablecoin pegged to INR, over-reserved by USD1. The amount minted depends on the *R(USD91)* rate at mint time; this rate is one of the many *R(USD_EMC)* rates constantly maintained by the protocol. The protocol will burn the 1 USD1.

Unlike MakerDAO [2], a Unitas stablecoin minter will NOT need to provide over-reservation; insurers provide it. The protocol must be sufficiently exogenously over-reserved (by insurers) in order for a user's mint request to be allowed.

The protocol's price stability mechanisms help maintain a peg of USD91 to INR (see *10 USD_EMC Price Pegging*), and users can use USD91 for business or crypto trading purposes. When wanting to convert USD91 back to a USDPEG, the user accesses the protocol and swaps USD91 for USD1 based on *R(USD91)*, then swaps USD1 for a USDPEG 1:1. For every swap, the protocol mints (e.g., USD91) and burns (e.g., USD1) simultaneously.

The protocol provides decentralized, unconditional, and unrestricted convertibility from an EMC stablecoin (e.g., USD91, USD55) back to a USDPEG (e.g., Dai, USDC, USDT; see *7.2.1*

*Unconditional Exit and Conditional Minting*). The user experience is akin to an asset swap rather than a loan. A swap process creates no debt positions.

# 5 Unitas Money Market Participants

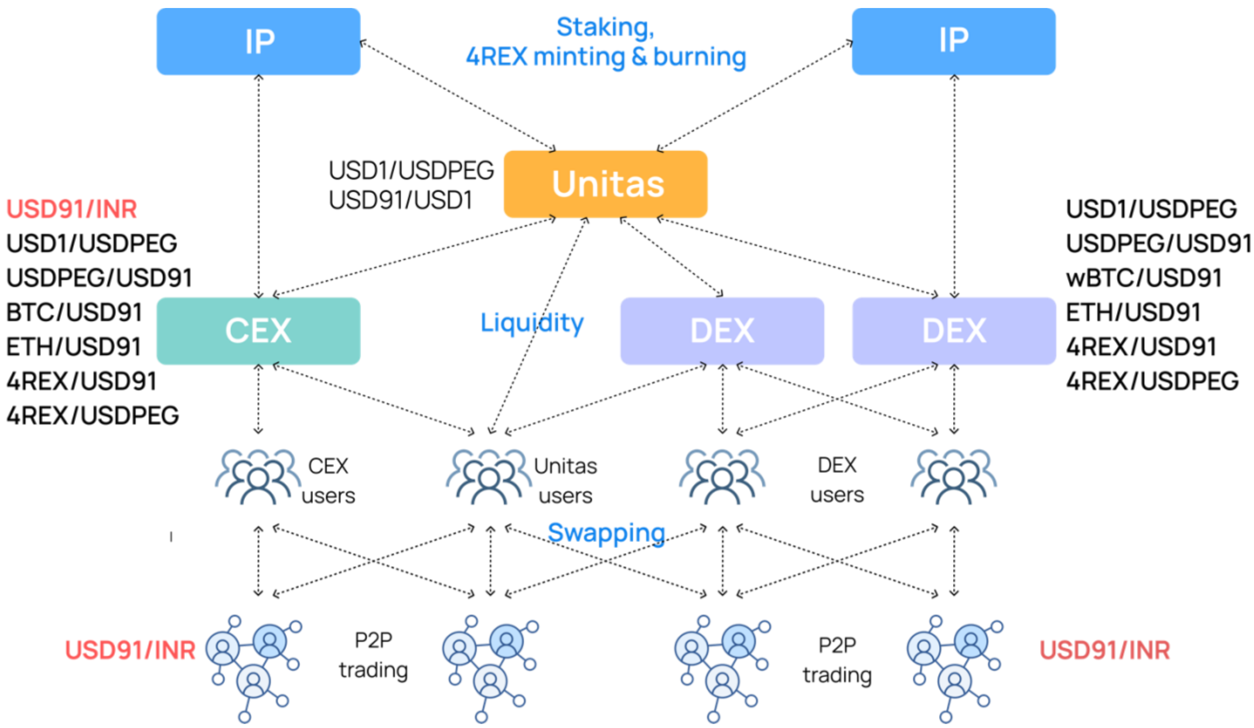


*Figure 5: Unitas money market participants.*

## 5.1 Unitas Stablecoin Users

Unitas will service at least two types of very different users. These include:

1) Cross-border businesses and users such as commodity traders, importers, manufacturers, money service businesses (MSBs), money transfer operators [24], multi-layered remittance networks [25], non-banking financial companies (NBFCs; 9000+ in India [26] [27]), correspondent banking networks (CBNs) [28], informal money transfer systems (IMTSs [20]), informal value transfer systems (IVTSs) [21], proprietary payment networks [29], neobanks, alternative payment methods (APMs) [30], and fintechs. Some users may not be crypto-savvy, but they understand stablecoins enough to use them for business purposes.

These businesses and users will convert between:

| Asset 1 | Asset 2 | Means of swap |
|---|---|---|
| USD_EMCs<br>(e.g., USD91) | EMCs<br>(e.g., INR) | 1. Local CEXs<br>2. Peer-to-peer offline trading |
| USD_EMCs<br>(e.g., USD91) | USD1 | 1. Local CEXs<br>2. Peer-to-peer offline trading<br>3. Unitas protocol (only the tech-savvy) |
| USD1 | USD-pegged stablecoins<br>(e.g. DAI, USDC) | 1. Local CEXs<br>2. Peer-to-peer offline trading<br>3. Unitas protocol (only the tech-savvy) |

*Table 1: Different swap types and means.*

Since many such users are often less familiar with DeFi, they'll find it easier to access trading pairs provided by their local CEXs or OTC partners. Unitas protocol serves as a liquidity provider or hedging protocol for these local CEXs and OTC desks.

Due to how the existing EMC money market operates, there will also be significant peer-to-peer or OTC-style trading [20] [26] [27] (see *Figure 5*).

2) EMC-denominated crypto investors and traders who hold Unitas stablecoins and invest or trade crypto on DEXs and CEXs that offer pairs such as BTC/USD91 and ETH/USD91.

## 5.2 Centralized Fiat-Crypto Exchanges (CEXs) and Decentralized Exchanges (DEXs)

Centralized fiat-crypto exchanges (e.g., those that can offer the USD91/INR pairs) are valuable participants in the Unitas ecosystem. Unitas Foundation will work with CEXs to offer fiat-crypto order books such as USD91/INR. These books serve as important oracles to one of the protocol's price-pegging mechanisms (e.g., USD91 to INR peg, see *10.2.1 Direct Mode*).

Similarly, USDPEG/EMC pairs such as USDT/INR and USD1/INR make important oracles that help the protocol determine its exchange rates between USD1 and USD_EMCs (see *10.2.2 Inference Mode*).

DEXs such as Curve [31] [32], Uniswap [33], Sushiswap [34], and Balancer [35] [36] are also vital participants in the Unitas ecosystem. Unitas Foundation will work with CEXs (fiat-crypto, crypto-crypto) and DEXs to offer trading pairs like USDPEG/USD91, wBTC/USD91, ETH/USD91, and USDPEG/USD1. While these pairs help increase the protocol's liquidity, they also help stabilize its pegs by presenting important arbitrage corridors (see *Figure 14*).

Because a large percentage of Unitas stablecoin usage will be for cross-border businesses, we foresee many users accessing USD_EMC liquidity (e.g., USDT/USD91) via CEXs instead of directly interacting with the protocol.

While it's often more convenient for users to access liquidity and trade on CEXs and DEXs, it is the market makers (of CEXs and DEXs) who will frequently convert with the Unitas protocol to rebalance and hedge their positions.

## 5.3 Arbitragers and Price Stability

Arbitragers play a vital role in the protocol's price stability mechanisms (e.g., USD91 to INR peg). Trading pairs (on CEXs and DEXs) such as USD91/INR, BTC/USD91, ETH/USD91, and USDPEG/USD91 enable opportunities for arbitragers in the Unitas ecosystem (see *Figure 14*).

To foster the arbitrager community, Unitas Foundation will steadily develop and grow an excellent set of open-source toolkits that facilitate arbitraging between the Unitas protocol, CEXs, and DEXs.

## 5.4 Unitas Stablecoin Minters and Burners

We envision that market makers (for both CEXs and DEXs) will be the majority of minters and burners. Additionally, minters and burners can include CeFi and DeFi organizations or any individual capable of interacting with the Unitas protocol.

## 5.5 Insurance Staking

The protocol will incur a loss between constant minting and burning operations if the total value of all its supported EMCs (e.g., INR, BRL, etc.) continuously appreciates against the USD. To defend against this unlikely situation, the protocol must attract insurance providers (IPs) to supply its over-reservation. It rewards IPs by distributing its profits to them (see *8.5 4REX Locking and ve4REX Generation* and *9 Revenue Generation, Maturity, Distribution, and Accounting*).

If the over-reservation ratio falls below a threshold, the protocol suspends minting of new USD_EMCs and encourages staking (see *7.2.1 Unconditional Exit and Conditional Minting*).

## 5.6 USD-Denominated Reservation and Multi-Reserve Unitas

Unitas Foundation will work to attract users from the blockchain industry as well as from traditional sectors such as cross-border commodity trading (see *5.1 Unitas Stablecoin Users*). While many participants won't be familiar with cryptocurrencies, they will be experts in emerging market forex trading and hedging.

Unitas V1 will only support USDPEGs as its reserve asset because:

a) it's easier to peg USD_EMCs to very stable USDPEGs,
b) it's easier to acquire confidence if the protocol is exogenously over-reserved by very stable USDPEGs, and
c) it lowers the entry barrier for participants (e.g., cross-border businesses) that aren't sophisticated DeFi users.

As Unitas Foundation's ecosystem grows, we plan to expand the protocol to Unitas V2, which will support a multi-reserve basket that includes non-USDPEGs (e.g., ETH).

# 6 Basic Parameters

We will use USD91 as an example; USD91 is the Unitas stablecoin pegged to INR.

## 6.1 Rates

Between USD1 and USD_EMCs (e.g., USD91, USD55), the protocol honors constantly changing exchange rates of *R(USD91), R(USD55), etc.*

## 6.2 Governance Token and Seigniorage

Following Sams' *Seigniorage Shares* [37] model, USD1 and USD-EMCs fulfill the protocol's transactional demand while its governance token 4REX satisfies its speculative demand. The protocol mints out 4REX tokens to IPs when it accepts insurance staking and creates Collateralized Debt Positions (CDPs) (see *8.1 Insurance Staking and Redemption*). It burns 4REX tokens when IPs submit them back to the protocol in order to redeem their staked principals and close their CDPs.

4REX holders can lock their 4REX into ve4REX, allowing them to a) earn yield and b) participate in Unitas protocol voting. Ve4REX yield comes from the protocol's profits (see *8.5 4REX Locking and ve4REX Generation* and *9.1 Revenue Sources*).

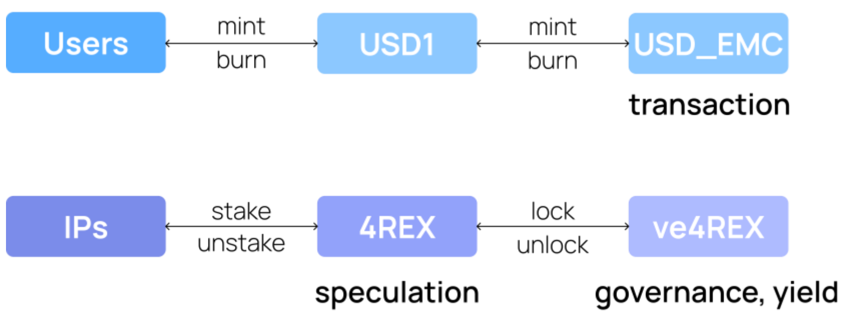


*Figure 6: Unitas' token model separates transactional, speculative, governance, and yield demands.*

## 6.3 Reserve and Collateral

The protocol has two independent collateral and reserve pockets holding USDPEG: $R_{minters}$ and $C_{insurers}$.

$R_{minters}$ – USDPEGs provided by minters upon minting USD1.
$C_{insurers}$ – USDPEGs staked by IPs.

We define total reserved USDPEG as $R_{total} = R_{minters} + C_{insurers}$.

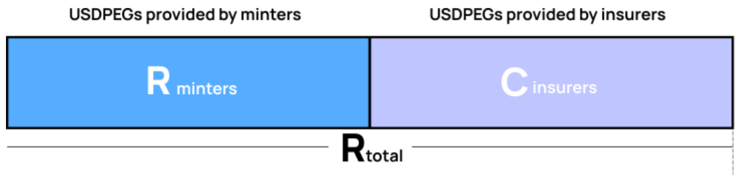


*Figure 7:* $R_{total} = R_{minters} + C_{insurers}$.

Using $R_{total}$, the protocol aims to achieve an overall (across all its stablecoins such as USD1, USD91, USD55, USD52, etc.) minimum over-reservation rate $OC_{min}$, which is configurable between 1.3 to 2.5, via the protocol's voting mechanism.

At the same time, the protocol aims to achieve (across all pairs) the maximum over-reservation rate $OC_{max}$, which is configurable via the protocol's voting mechanism (see also *7.1 Reservation Tracking and Reserve Operations* and *8.2 Minting 4REX Tokens)*. $OC_{max}$ must be greater than $OC_{min}$.

## 6.4 Surplus Reserve

The protocol generates revenue from the fees it collects in USD1 and distributes them to 4REX (more precisely, to ve4REX; see *8.5 4REX Locking and ve4REX Generation*) holders. Before distribution, it holds the revenue in the surplus reserve $S_{protocol}$.

## 6.5 Protocol-Controlled Liquidity and Protocol-Owned Liquidity

Because IPs cannot redeem their assets staked into $C_{insurers}$ before maturity, the protocol "rents" insurance funding from IP providers by paying out seigniorage dividends. Collaterals under staking become protocol-controlled liquidity (PCL) [38], which the protocol uses to a) insure $R_{total}$ and b) generate yield via external yield venues.

The protocol pays liquidity rent using its generated fees, not its governance tokens. As the protocol generates revenue from fees and yields, it uses most profits to reward (pay rent to) IPs. It burns the remaining to reduce its liability (see *9.3 Reserve Growth Burn (RGB)* ).

This liability-decreasing process increases the protocol's reserve ratio and accumulates protocol-owned liquidity (POL) [39], eventually causing 4REX to become deflationary (see *9.3.1 Deflationary 4REX*).

# 7 Minting, Burning, and Reserving Unitas Stablecoins

During its initial phase, Unitas will pick a single supported USDPEG, which we'll hereafter refer to as $USDPEG_A$.

To enter the Unitas ecosystem, a user must first use $USDPEG_A$ to mint USD1, where USD1 is Unitas' USD stablecoin exogenously over-reserved by $USDPEG_A$.

Once a user has USD1, they can either a) swap USD1 for USD_EMC (e.g., USD91) or b) swap USD1 back to $USDPEG_A$. Once a user has USD_EMC (e.g., USD91), they can always swap back to USD1 and subsequently swap back to $USDPEG_A$.

Every swap triggers minting and burning of two involved assets (except for USD1/$USDPEG_A$, where the protocol only mints or burns USD1 while locking $USDPEG_A$ into a reserve pool). The protocol charges a fee in USD1 in every swap.

## 7.1 Reservation Tracking and Reserve Operations

The protocol keeps track of the total minted stablecoins via:

a) $M_{total}(USD91)$ – total outstanding USD91,
b) $M_{total}(USD55)$ – total outstanding USD55,
c) $M_{total}(USD52)$ – total outstanding USD52,
d) $M_{total}(USD1)$ – total outstanding USD1 (including revenue in USD1 held by the protocol),
e) etc.

The protocol's total liability (in $USDPEG_A$) would be:

$$L_{total} = (M_{total}(USD91) / R(USD91)) + (M_{total}(USD55) / R(USD55)) + \ldots + (M_{total}(USD52) / R(USD52)) + M_{total}(USD1).$$

Minting a Unitas stablecoin is only allowed if the protocol is sufficiently exogenously over-reserved. Therefore, to understand the protocol's reserve rate, we define the following:

a) reserve ratio $D_{minters} = R_{minters} / L_{total}$, and
b) reserve ratio $D_{total} = R_{total} / L_{total}$.

The protocol's best-case scenario is if $D_{minters}$ > 1, meaning that the protocol is fully reserved with just reserves from minters: $R_{minters}$.

The protocol's minimum-viable scenario is $D_{total}$ > 1, which means $R_{total}$ fully reserves the protocol. $D_{total}$ <= 1 would be the protocol's doomsday scenario and will immediately trigger a global settlement process. During a global settlement event, insurers will lose most of their staked funds, while Unitas stablecoin holders will be made whole by receiving equivalent values of USD1, according to *R(USD_EMC)*. They can then swap USD1 for $USDPEG_A$.

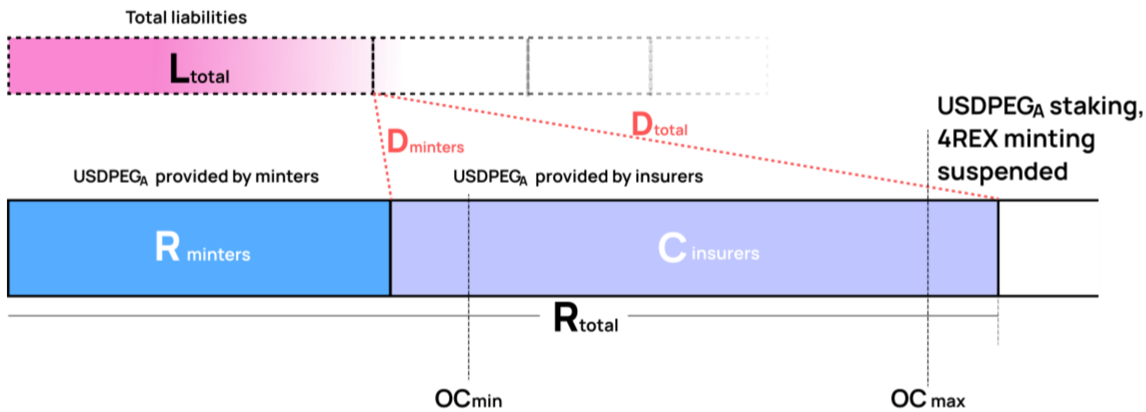


*Figure 8: Reserve ratio, $OC_{min}$,* and *$OC_{max}$.*

### 7.1.1 Stability Burn

If $D_{total}$ < $OC_{min}$, the protocol hasn't reached its minimum over-reservation ratio. The protocol will perform *Stability Burn* – it will burn USD1s from $S_{protocol}$ (which will increase $D_{total}$) until $D_{total}$ >= $OC_{min}$ (we elaborate on this in *7.4 Reserve Loss and Growth* and *9.2 Revenue Maturity*). The protocol will also accept staking from insurers, which we will cover in *8.1 Insurance Staking and Redemption*).

Suppose the protocol's insurance staking generates good returns for stakers and constantly attracts stakers who want to participate in the protocol's insurance staking. In that case, $D_{total}$ should be close to $OC_{max}$, which is the protocol's reservation ceiling (see *Figure 8* and also *8.2 Minting 4REX Tokens*).

## 7.2 Minting Unitas USD_EMC Stablecoins

A user first uses $USDPEG_A$ to mint USD1; the protocol transfers the $USDPEG_A$ into $R_{minters}$ and mints USD1 on the spot. The protocol collects a fee (in USD1) of $Fee_{mint\text{-}usd1\text{-}with\text{-}usdpega}$, which is configurable by voting.

Next, the minter submits $X$ amounts of USD1 to the protocol in exchange for $X*R(USD91)$ amounts of USD91; the protocol burns the USD1 and mints USD91 on the spot.

The protocol collects a fee (in USD1) of $Fee_{mint\text{-}usd91\text{-}with\text{-}usd1} * X$. $Fee_{mint\text{-}usd91\text{-}with\text{-}usd1}$ is configurable by voting.

### 7.2.1 Unconditional Exit and Conditional Minting

*"Getting to the top is optional; getting down is mandatory."* – Ed Viesturs, American mountaineer, husband and father.

---

The protocol allows unrestricted and unconditional exit from EMCs back to $USDPEG_A$ – that is, conversions from USD_EMC to USD1 and then from USD1 to $USDPEG_A$.

In contrast, the protocol performs the following operations only if $D_{total} > OC_{min}$ :

a) minting USD1 using $USDPEG_A$ (i.e., converting $USDPEG_A$ to USD1), or
b) minting USD_EMC using USD1 (i.e., converting USD1 to USD_EMC).

| | | | |
|---|---|---|---|
| $USDPEG_A$ | ← | USD1 | Unconditional |
| USD1 | ← | USD_EMC | Unconditional |
| $USDPEG_A$ | → | USD1 | *Only if* $D_{total} > OC_{min}$ |
| USD1 | → | USD_EMC | *Only if* $D_{total} > OC_{min}$ |

*Table 2: Each swap's preconditions.*

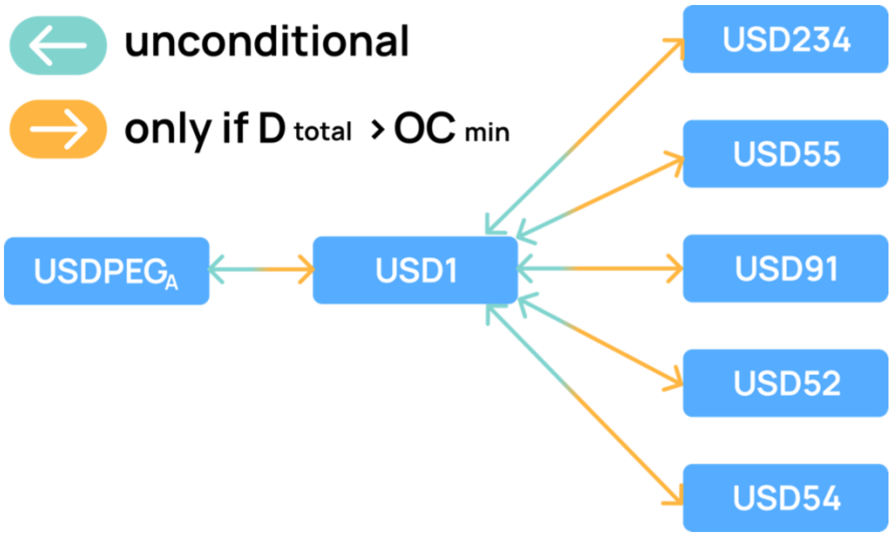


*Figure 9: Unconditional versus conditional conversions.*

If $D_{total}$ <= $OC_{min}$, the protocol suspends minting of new USD_EMCs (converting from an USD1 to an USD_EMC) until otherwise (see *Figure 10*); it increases $D_{total}$ using two mechanisms:

1) burning all its revenue in USD1 (see *7.1.1 Stability Burn*), and
2) encouraging staking (see *8.1 Insurance Staking and Redemption*).

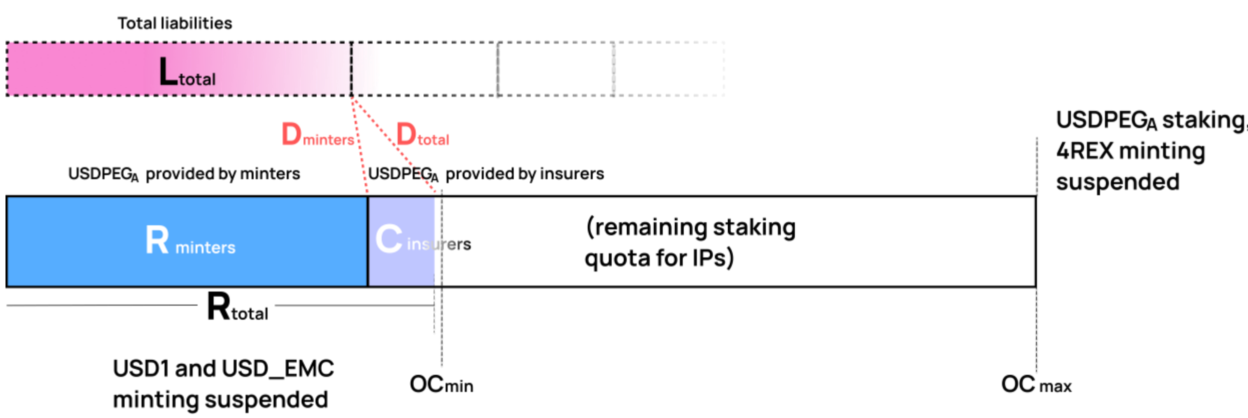


*Figure 10: USD1 and USD_EMC minting restrictions.*

Alternatively, a minter can also act as an insurer and submit an insurance staking request attached with a USD_EMC minting request. This allows the insurer to self-provide the required insurance funds, and guarantees the minting request's acceptance.

Even when $D_{total}$ > $OC_{min}$ and minting of new USD_EMCs are allowed, the protocol burns a percentage of its revenue in USD1 to further increase $D_{total}$ (see *7.4 Reserve Loss and Growth* and *9.3 Reserve Growth Burn (RGB)* ).

## 7.3 Burning Unitas USD_EMC Stablecoins

### 7.3.1 USD_EMC to USD1

During USD_EMC stablecoin burning, the minter submits *Y* amounts of Unitas USD_EMC stablecoin (e.g., USD91) to the protocol in exchange for *Y / R(USD_EMC)* amounts of USD1; the protocol burns the USD_EMC and mints USD1 on the spot.

The protocol collects a fee (in USD1) of $Fee_{burn\text{-}usd91\text{-}for\text{-}usd1}$ * (Y / R(USD91)). $Fee_{burn\text{-}usd91\text{-}for\text{-}usd1}$ is configurable by voting.

### 7.3.2 USD1 to $USDPEG_A$

The user can decide to keep the minted USD1 because USD1 can be converted easily to any USD_EMC supported by the protocol. If the user wants to exit the Unitas ecosystem, the protocol allows unrestricted and unconditional conversions from USD1 to $USDPEG_A$.

During USD1 to $USDPEG_A$ conversion, the protocol transfers $USDPEG_A$ from $R_{minters}$ to the user and burns USD1 on the spot. If the $USDPEG_A$s stored in $R_{minters}$ isn't sufficient for the swap, the protocol uses those from $C_{insurers}$.

The protocol collects a fee (in USD1) of $Fee_{burn\text{-}usd1\text{-}for\text{-}usdpega}$, which is configurable by voting.

## 7.4 Reserve Loss and Growth

Reserve loss may occur between continued USD1/USD_EMC swaps due to EMCs' steady appreciation against the USD. The net result will demonstrate a continued decrease in $D_{total}$. Conversely, because Unitas Foundation will deliberately and selectively support only steadily depreciating EMCs (against the USD) [8], $D_{total}$ should naturally increase over time.

The protocol incorporates the following mechanisms to accrue its reservation and increase $D_{total}$:

a) supporting only depreciating EMCs (against the USD),
b) burning revenue (in USD1) during temporal $D_{total}$ decrease (see *7.1.1 Stability Burn* and *9.2 Revenue Maturity*), and
c) regular burning a percentage of revenue (in USD1, see *9.3 Reserve Growth Burn (RGB)* ).

The results will demonstrate a steady increase in $D_{total}$, and the protocol suspending (and not needing) insurance staking for an extended time. Since staking is the only means of minting 4REX tokens, staking suspension freezes minting new 4REX tokens (see also *8.2 Minting 4REX Tokens* and *Figure 12* ).

# 8 Insurance Staking and 4REX Minting

## 8.1 Insurance Staking and Redemption

The protocol attracts IPs to stake $USDPEG_A$. IPs stake their $USDPEG_A$ with the protocol to a) receive yield and b) mint 4REX tokens.

Since each staking is redeemable after maturity, a healthy Unitas ecosystem should see yield earning as a primary reason for most IPs.

For an IP, each staking generates the following:

a) a *Collateralized Debt Position (CDP)* [2], which the IP can use to redeem staked principal (but only after maturity), and
b) a certain amount of 4REX tokens, which the IP can lock with the protocol in exchange for yield-generating ve4REX tokens (see *8.5 4REX Locking and ve4REX Generation*).

Each CDP delineates:

a) the amount of staked principal,
b) the amount of minted 4REX tokens, and
c) the maturity date and time.

Closing a CDP and redeeming its staked principal requires:

a) submitting a closure request after the CDP matures, and
b) surrendering the amount of 4REX tokens minted with the CDP; the protocol burns these 4REX.

Once a CDP matures, its owner has the option to continue holding the CDP; closing a CDP upon maturity is not mandatory.

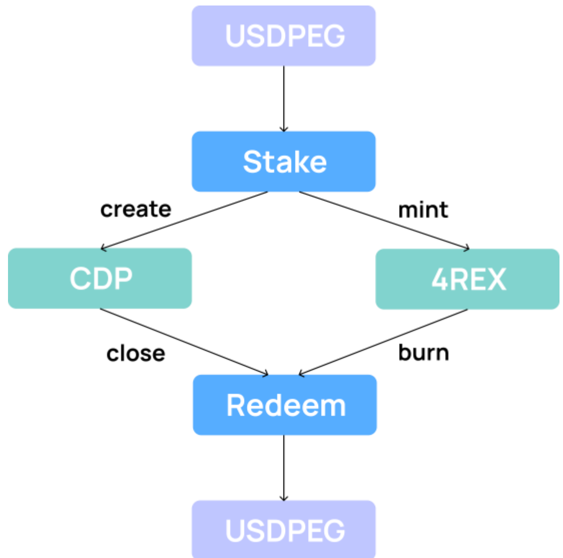


*Figure 11: Insurance staking and redemption.*

Receiving 4REX tokens in return for their staking allows IPs to:

1) lock 4REX into ve4REX to receive yield,
2) speculate against the protocol's transactional demand [37] (and therefore 4REX price) and trade 4REX,
3) have liquidity prior to CDP maturity by selling 4REX (see *8.3 Liquidity While Insurance Staking*), and
4) participate in protocol voting (see *6.2 Governance Token and Seigniorage*).

### 8.1.1 Redemption Gating

If the protocol is operating healthily and IP staking demand is high, $D_{total}$ will always remain around $OC_{max}$. If $D_{total}$ drops below $OC_{max}$ and further drops below $OC_{min}$, the protocol will temporarily suspend its CDP redemptions to ensure sufficient over-reservation. This situation could arise due to continued strong demand for USD1 minting, outpacing IP staking supply.

The protocol's swapping functionalities (e.g., USD1/USDT and USD91/USD1 swaps) will operate normally during redemption suspension. IPs having mature CDPs wanting to redeem can leverage their 4REX tokens for liquidity (see *8.3 Liquidity While Insurance Staking*).

CDP redemptions will reopen as soon as $D_{total}$ increases above $OC_{min}$.

## 8.2 Minting 4REX Tokens

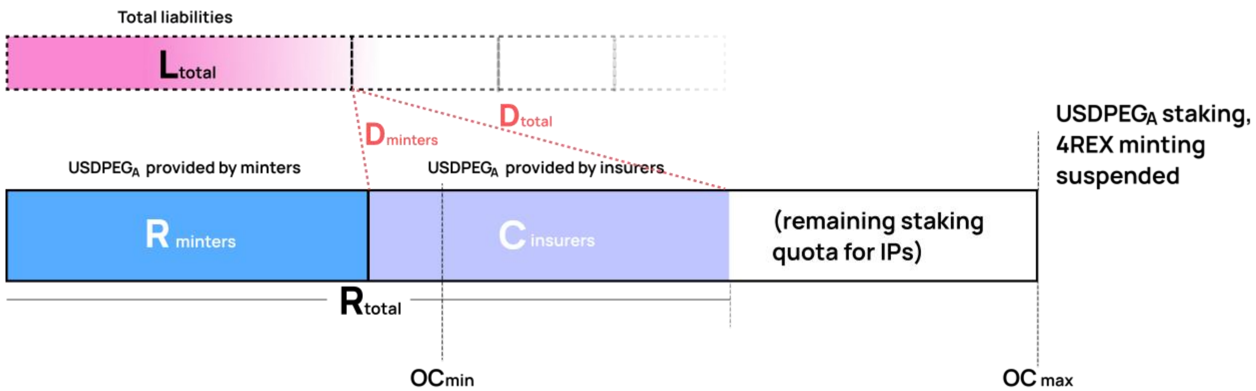


*Figure 12: Reservation ceiling and staking suspension.*

Insurance staking is the only way to mint 4REX tokens; each staking creates a CDP and mints a certain amount of 4REX tokens. To redeem a CDP's staked principal, the originally-issued amount of 4REX tokens must be surrendered to the protocol for burning. The protocol only allows staking if $D_{total} < OC_{max}$ (see *Figure 12*). Otherwise, the protocol has reached its reservation ceiling; it suspends a) USDPEG$_A$ staking and therefore b) 4REX minting.

## 8.3 Liquidity While Insurance Staking

Before a CDP matures, the owner of the CDP can:

a) lock the minted 4REX in exchange for ve4REX and earn yield (in USD1), or

b) obtain liquidity by selling 4REX for USDPEG$_A$ in secondary markets.

In the case of (b), the CDP holder won't receive any yield, and must buy back an equal amount of 4REX if they ever want to close this CDP and redeem staked USDPEG$_A$.

## 8.4 Insurance Staking Auctions

IPs go through a closed reverse auction process to stake USDPEG$_A$, create their CDPs, and mint 4REX. The protocol runs staking auctions when and only when $D_{total} < OC_{max}$. An ideal state is $D_{total} >= OC_{max}$, in which case staking auctions are closed. The following situations can cause $D_{total}$ to drop:

a) EMCs appreciating against the USD, or

b) users submitting new requests to mint USD1 using USDPEG$_A$.

Each bid involves the following attributes:

a) $B_{principal}$ : the amount of principal (in $USDPEG_A$) the bidder is staking,
b) $B_{maturity}$ : the maturity window – for how long the bidder is staking the principal, and
c) $B_{price}$ : 4REX price – how much (i.e., amount of 4REX) the bidder expects the protocol to mint in exchange for 1 $USDPEG_A$ (or what is 1 $USDPEG_A$'s price in 4REX, that the protocol must pay to the bidder).

### 8.4.1 Maturity-Based Auction Markets (MBAMs)

The protocol provides multiple *Maturity-Based Auction Markets (MBAMs)* to submit bids. Each MBAM stipulates a different maturity window – $MBAM_{1460}$ (1460 days or roughly 4 years), $MBAM_{1095}$ (roughly 3 years), $MBAM_{730}$ (roughly 2 years), $MBAM_{365}$, and $MBAM_{30}$.

Bidders must first select an MBAM to which they'll submit their bids. The protocol prioritizes MBAMs according to their maturity windows when launching an auction, giving higher priorities to those with longer windows.

An MBAM is *empty* if it contains no bids. The protocol identifies the highest-priority, non-empty MBAM to launch its next auction.

### 8.4.2 Reverse First-Price Sealed-Bid Auction (R-FPSBA)

Once the protocol determines the MBAM to process, it performs a *Reverse First-Price Sealed-Bid Auction (R-FPSBA)* against all outstanding bids within the selected MBAM.

An IP can conveniently think of a bid as the IP staking their $USDPEG_A$ with the protocol in exchange for 4REX. The IP can then lock these 4REX in exchange for yield-generating ve4REX. Under this rationale, $B_{price}$ represents how much 4REX a bidder expects the protocol to pay for every 1 $USDPEG_A$ staked. The protocol institutes a $B_{price}$ ceiling price of 1 (i.e., 1 4REX for 1 $USDPEG_A$), and the lowest-priced bid wins. For example if there are two bids: 1 4REX for 1 $USDPEG_A$, and 0.9 4REX for 1 $USDPEG_A$, the latter bid wins.

Amongst multiple lowest-priced bids, the protocol prioritizes seniority (those submitted earlier) and accepts as many bids as possible as long as $D_{total} < OC_{max}$.

## 8.5 4REX Locking and ve4REX Generation

4REX can only be minted by staking $USDPEG_A$, and can be locked to generate *vote-escrowed* [40] [41] [42] [43] 4REX (i.e., ve4REX). While 4REX can neither generate yield nor be used to vote, ve4REX is capable of both (see also *Figure 6* ).

Locking 4REX is the only way to obtain ve4REX, which isn't a standard ERC20 and cannot be transferred. Although ve4REX took inspiration from the earlier vote-escrow works of Curve [40], Yearn [41], and Cronje [42], it has a different design.

### 8.5.1 Time-Incentivized Locking

The protocol favors and incentivizes longer locks. For every 1 4REX locked for 30 days, the protocol generates 0.25 ve4REX; for every 1 4REX locked for longer, the protocol generates more than 0.25 ve4REX. *Table 3* defines the amounts generated.

| Lock Duration | Locked 4REX | Generated ve4REX |
|---|---|---|
| 30 days | 1 | 0.25 |
| 365 days (1 year) | 1 | 1 |
| 730 days (2 years) | 1 | 1.05 |
| 1095 days (3 years) | 1 | 1.16 |
| 1460 days (4 years) | 1 | 1.3 |

*Table 3: ve4REX generation based on lock duration.*

## 8.6 4REX Price Support

4REX price support comes from:

a) 4REX is required to redeem $USDPEG_A$ from a CDP, and 1 4REX can redeem at least 1 $USDPEG_A$,
b) 4REX can be locked into ve4REX to receive profit distribution (yield), and
c) ve4REX can be used to participate in protocol voting.

Point (a) results from R-FPSBA having a ceiling price of 1 4REX per $USDPEG_A$ staked. This implies that for every staked $USDPEG_A$, the protocol mints 1 or less 4REX. Therefore, given any CDP, 1 4REX can always redeem 1 or more $USDPEG_A$.

4REX will eventually become deflationary once the protocol generates enough revenue; see *6.5 Protocol-Controlled Liquidity and Protocol-Owned Liquidity* and *9.3.1 Deflationary 4REX*.

# 9 Revenue Generation, Maturity, Distribution, and Accounting

## 9.1 Revenue Sources

The protocol has three sources of revenue:

a) Transaction fees (in USD1) – generated from swapping two assets, such as USD1/ $USDPEG_A$ or USD_EMC/USD1. These fees include for example:
    1) minting fees such as $Fee_{mint\text{-}usd1\text{-}with\text{-}usdpega}$ and $Fee_{mint\text{-}usdxxx\text{-}with\text{-}usd1}$ (see *7.2 Minting Unitas USD_EMC Stablecoins*), and
    2) burning fees such as $Fee_{burn\text{-}usdxxx\text{-}for\text{-}usd1}$ (see *7.3.1 USD_EMC to USD1*) and $Fee_{burn\text{-}usd1\text{-}for\text{-}usdpega}$ (see *7.3.2 USD1 to USDPEGA*).
b) Yield generation by staking a part of the $USDPEG_A$ reserve $R_{total}$ into external yield venues (e.g., Compound [44], Aave [45] [46]) or into liquidity pools (e.g., Curve [32], Uniswap [33]). The protocol stores such yields in USD1; if an external yield protocol generates yields in $USDPEG_A$, Unitas uses its own mechanism to swap the $USDPEG_A$ for USD1 before storing them.
c) EMC depreciation (against the USD) [8], which increases $D_{total}$ and can eventually lead to a deflationary 4REX (see *9.3.1 Deflationary 4REX*).

## 9.2 Revenue Maturity

The protocol stores its revenue in the surplus buffer array $S_{protocol}$ [i] of 7 before distributing them to ve4REX holders. Each array (i.e., $S_{protocol}$ [0] to $S_{protocol}$ [6]) is used to store one day of revenue and has a maturity period of 7 days.

An $S_{protocol}$ [x] *matures* if 7 days pass and $D_{total}$ consistently exceeds $OC_{min}$. Before an $S_{protocol}$ [x] reaches maturity, if at any time $D_{total}$ falls below $OC_{min}$ ($D_{total}$ <=$OC_{min}$), then in order to increase $D_{total}$, all its USD1 will be burned. We introduced this burn mechanism in *7.1.1 Stability Burn* and *7.4 Reserve Loss and Growth*.

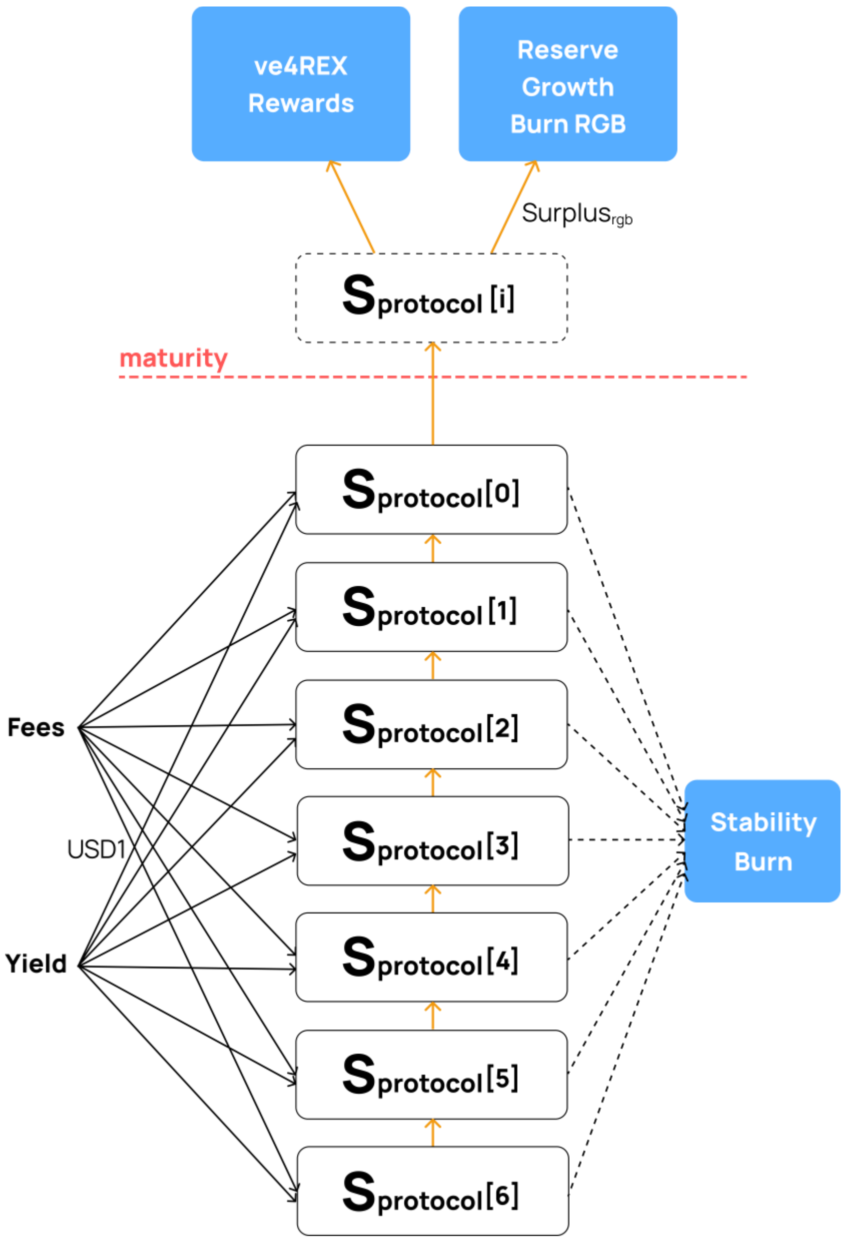


*Figure 13: Revenue generation, maturity, and distribution.*

## 9.3 Reserve Growth Burn (RGB)

Once an $S_{protocol}$ [x] matures, the protocol:

a) burns $Surplus_{rgb}$ percent of its USD1 (explained below), and

b) distributes the remaining USD1 to all ve4REX holders according to the amount of ve4REX owned by each.

The protocol burns a percentage ($Surplus_{rgb}$) of its matured revenue to increase reservation rate $D_{total}$ and protocol-owned liquidity [39], strengthening protocol stability (see *6.5 Protocol-Controlled Liquidity and Protocol-Owned Liquidity*). We call this mechanism *Reserve Growth Burn (RGB)*. $Surplus_{rgb}$ is configurable by voting.

### 9.3.1 Deflationary 4REX

As described in *6.5 Protocol-Controlled Liquidity and Protocol-Owned Liquidity*, the protocol's over-reserve pocket $C_{insurers}$ consists of:

1) $USDPEG_A$ staked by IPs (Protocol-Controlled Liquidity, PCL), and
2) $USDPEG_A$ contributed from the protocol's profits (Protocol-Owned Liquidity, POL).

By executing RGB, a decisive goal of Unitas is to eventually fund the entire $C_{insurers}$ pocket (the over-reserve pocket) with its POL. This means that POL grows sizable enough so that insurance staking is no longer necessary. At this point, 4REX becomes deflationary once the following occurs in parallel:

1) RGB keeping $D_{total}$ above $OC_{max}$ (see *Figure 12*), prohibiting further 4REX minting even if USD_EMC minting continues to happen, and
2) IPs burning 4REX to close their CDPs and redeem their principals.

## 9.4 Profit Loss Accounting Against Forex Volatility

As we summarize in *9.1 Revenue Sources*, the protocol has three sources of revenue. Source (c) EMC exchange rate movement (against USD) incurs both profit as well as loss. Therefore, with this type of revenue, the protocol implements profit-loss accounting to offer transparency to all stakeholders and market participants. Profit-loss analytics provides valuable insights into the protocol's health and sustainability.

The protocol has two types of liabilities: outstanding USD1 and USD_EMCs. Because USD1 swaps 1:1 with $USDPEG_A$, the protocol focuses its profit-loss accounting on USD_EMC liabilities, which are directly impacted by EMCs' volatility against the USD. While the protocol tracks its overall reserve ratio using $D_{total}$, it's also important to track its profit and loss across continued USD_EMC/USD1 swaps.

### 9.4.1 Booking Deferred Revenue and Cost

The protocol tracks its profit and loss using an accrual-based accounting model. When a user converts *d* USD1 into *c* USD_EMC (i.e., mints USD_EMC), the protocol simultaneously a) books *d* USD1 as new *deferred revenue* and b) books *d* USD1, which is *c* USD_EMC/*R(USD_EMC)* and time-varying, as new *deferred cost*.

The protocol also tracks the total *outstanding* USD_EMC (i.e., total minted minus total burned). The total deferred cost will vary as *R(USD_EMC)* fluctuates.

### 9.4.2 Recognizing Revenue and Cost

When a user converts *c'* USD_EMC into *d'* USD1, the protocol simultaneously:

a) calculates and recognizes *earned revenue* in USD1,
b) recognizes *d'* USD1 as *realized cost*,

### 9.4.3 Profit and Loss of Each USD_EMC/USD1 pair

Because the protocol will support multiple EMCs, we denote $USD\_EMC_i$ as the i-th EMC that the protocol supports (e.g., USD91, USD55, USD52, etc.). We denote $R_t(USD\_EMC_i)$ as the protocol's rate $R(USD\_EMC_i)$ at time t.

We define:

1. Total *deferred cost* of an $USD\_EMC_i$, evaluated at time t:

$$DC_i(t) = value\ of\ total\ outstanding\ USD\_EMC_i\ at\ time\ t\ denominated\ in\ USD1$$
$$= M_{total,t}(USD\_EMC_i) \div R_t(USD\_EMC_i)$$

2. Total *realized cost* of an $USD\_EMC_i$, recognized during time interval [s,t]:

$$RC_i(s,t) = \sum_{s \le u \le t} amount\ of\ USD\_EMC_i\ burned\ at\ time\ u \div R_u(USD\_EMC_i)$$
$$= cumulative\ amount\ of\ USD1\ redeemed\ via\ USD\_EMC_i\ burning\ during\ [s,t]$$

3. Total *deferred revenue* (post-mint) for the $USD\_EMC_i$ at time t:

$$DR_i(t) = cumulative\ amount\ of\ USD1\ submitted\ for\ minting\ USD\_EMC_i - \sum_{u \le t} ER_i(u),$$

where $ER_i(u)$ is defined below.

4. *Earned revenue* of an USD_EMC$_i$, generated at time t:

$$ER_i(t) = \frac{amount\ of\ USD\_EMC_i\ burned\ at\ time\ t}{total\ amount\ of\ outstanding\ USD\_EMC_i\ at\ time\ t\ (pre - burn)} * DR_i(t_-),$$

where $DR_i(t_-)$ is the total deferred revenue right before the USD_EMC$_i$ burning at time t.

Even though it may seem that we've defined the two revenue types in a circular reference, we emphasize that cumulative earned revenue would be zero until an USD_EMC$_i$ burning happens for the first time. This suggests that, with the following initial condition, we can obtain both types of revenue (iteratively) by first having $DR_i(t_{0-})$ and then obtaining $ER_i(t_0)$, where $t_0$ is the first time point that a burning of the USD_EMC$_i$ happens:

$$\sum_{u<t_0} ER_i(u) = 0$$

With these definitions,

5. The amount of USD1 that the protocol holds as reserve for USD_EMC$_i$ is

$$DR_i(t) + \sum_{u \le t} ER_i(u) - RC_i(0, t),$$

which simply says that it's equal to the sum of the two revenue types minus the realized cost.

Using these definitions, the protocol can readily calculate each USD_EMC$_i$'s profit and loss.

# 10 USD_EMC Price Pegging

The protocol's most important goal is to peg a USD_EMC to an EMC as closely (to 1:1) as possible. This means when two users swap their USD_EMC and EMC, their rate is very close to 1.

## 10.1 Pegging Methodology

The critical factor in pegging a USD_EMC to an EMC is the protocol's rate *R(USD_EMC)*, which the protocol honors when executing swaps between USD_EMC and USD1. The protocol implements the following pegging methodology:

Peg-Stage1) Collect data on current (real-time) USD_EMC:EMC rate $R_P(EMC, Time)$.

Peg-Stage2) Based on $R_P(EMC, Time)$, adjust *R(USD_EMC)*. If a USD_EMC is traded lower than EMC, we decrease *R(USD_EMC)* to encourage USD_EMC burning (i.e., swapping USD_EMC for USD1) to reduce its supply. Conversely, we increase *R(USD_EMC)* to encourage minting to increase its supply.

The protocol announces the change via suitable feeds (e.g., Chainlink).

Peg-Stage3) Allow arbitrage to occur via multiple corridors. Wait for a certain duration and loop back to Peg-Stage1.

Below are some corridors that arbitragers can leverage during Peg-Stage2:

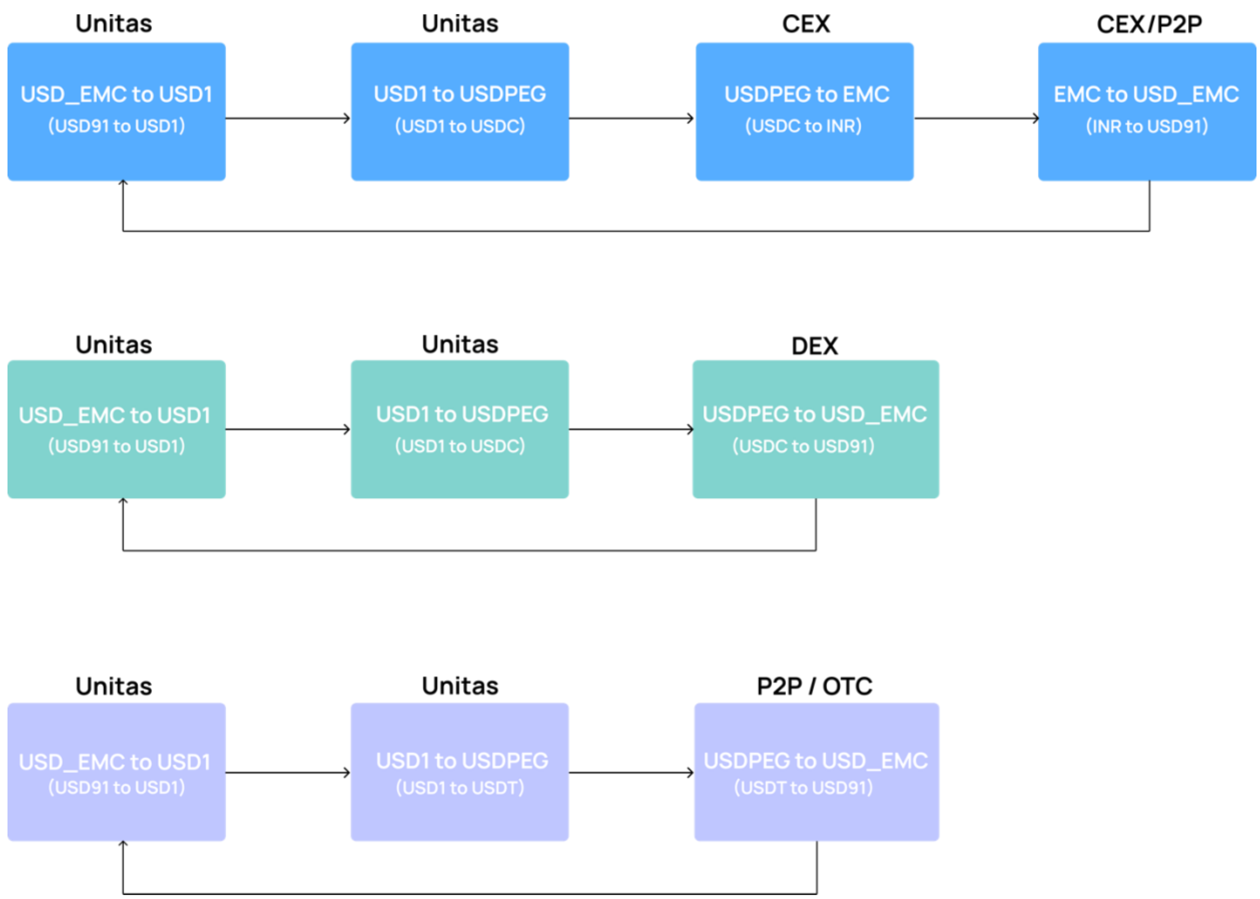


*Figure 14: Some arbitrage corridors.*

## 10.2 Different and Independent Pegging Modes

The three most common means of user swapping are:

a) peer-to-peer offline trading,
b) OTC-style offline trading, and
c) USD_EMC/EMC (e.g., USD91/INR) open book trading on CEXs.

We note that while the protocol will have direct visibility into the swapping rates of (c) via oracles, it won't have equal visibility into the rates of (a) or (b).

The protocol implements three independent pegging modes: a) Direct Mode, b) Inference Mode, and c) Estimation Mode.

### 10.2.1 Direct Mode

Direct Mode directly collects (via oracles) open book rates of USD_EMC/EMC. This rate collector only works with fiat-enabled CEXs that offer open USD_EMC/EMC books. Unitas Foundation will launch partnership and liquidity programs to encourage and facilitate fiat-enabled CEXs to offer USD_EMC/EMC order books.

### 10.2.2 Inference Mode

When the protocol cannot observe USD_EMC/EMC exchange rates directly, Unitas Foundation can decide to leverage Inference Mode, which infers rates by stacking up two open books. For example:

a) ETH/EMC (e.g., ETH/INR) and ETH/USD_EMC (e.g., ETH/USD91)
b) $USDPEG_A$/EMC (e.g., USDT/INR) and $USDPEG_A$/USD_EMC (USDT/USD91)

### 10.2.3 Estimation Mode

If neither Direct nor Inference Modes are possible, the protocol degrades to Estimation Mode, which:

a) unlike Direct and Inference Modes, doesn't directly look at how a USD_EMC is trading with an EMC, but instead
b) tries to estimate the right *R(EMC)*, (and sets the right *R(USD_EMC)*).

Estimation Mode follows the following formula:

*R(EMC)* = $O_{USD}$*(EMC)* * *B(EMC)*, where:

- $O_{USD}$*(EMC)* : An oracle against a formal or informal USD/EMC rate, for example:
  - a) Chainlink's INR/USD price feed [47], and
  - b) CoinDCX's open book on USDC/INR [48].
- *B(EMC)* : A bias-correction quotient applied to an $O_{USD}$*(EMC)*.

*B(EMC)* is necessary for an $O_{USD}$*(EMC)* that has a dislocation to the rate offered by major liquidity providers (see *3.4.2 Inaccessibility of Forex Rates in Emerging Markets* and *5.1 Unitas Stablecoin Users*). For example, at the time of writing:

a) Chainlink's INR/USD price feed [47], which is based on real-time data provided by the Reserve Bank of India (RBI), has USD/INR at 1:76.52, while

b) CoinDCX's open book on USDC/INR [48] is at 1:80.89, trading USDC to INR at a 5.71% premium against RBI's USD to INR.

This type of premium is the norm in emerging markets where central banks provide very limited USD liquidity and people rely on informal money transfer systems (IMTSs) [20] or informal value transfer systems (IVTSs) [21].

To conclude the example above, if the protocol sets $O_{USD}$*(INR)* to Chainlink's INR/USD feed, it should set *B(EMC)* to around 1.0437.

## 10.3 Adjusting *R(USD_EMC)* (Peg-Stage2)

In Direct and Inference Modes, the protocol adjusts *R(USD_EMC)* every *R_ADJ(EMC)* hour and in *R_INC(EMC)* increments.

Whereas Direct and Inference Modes can achieve a high level of autonomy, Estimation Mode will require frequent manual tuning, requiring that Unitas participants frequently vote to adjust *B(EMC)*. The protocol aims to implement Direct or at least Inference Mode; however, it may need to use Estimation Mode during early protocol stages.

## 10.4 Fostering Arbitrage Corridors (Peg-Stage3)

Because arbitrage efficiency is essential to the protocol's USD_EMC pegs, Unitas Foundation must help foster multiple arbitrage corridors (see *Figure 14*).

#### 10.4.1 Direct Pairs

Unitas Foundation aims to partner with as many fiat-crypto exchanges as possible and create many USD_EMC/EMC pairs (e.g., USD91/INR). These pairs traded at significant volume will allow Oracle-Direct rate collection and bolster USD_EMC/EMC pegs.

#### 10.4.2 USDPEG Pairs

Another essential pair type is USDPEG/USD_EMC – for example, USDT/USD91, USDC/USD91, and DAI/USD91. Because only fiat-crypto CEXs can offer them, USD91/INR pairs may be time-consuming to create and vulnerable to regulatory uncertainty. In contrast, the USDPEG/USD_EMC pair type is significantly easier to create, especially on DEXs.

#### 10.4.3 High Volume Pairs

High-volume pairs such as (w)ETH/USD91 and (w)BTC/USD91 on CEXs and DEXs will also help provide solid arbitrage corridors. Both USDPEG and high-volume pairs will allow Oracle-Inferred data collection.

# 11 Unitas V2

One main direction of Unitas V2 is a multi-reserve strategy. There's been much debate regarding today's *"algorithmic stablecoins,"* and the term itself has had changing definitions. Instead of using the "collateralized versus algorithmic" model to categorize stablecoins, we offer a new perspective to compare different designs.

## 11.1 Towards a Multi-Reserve Strategy: Endogenous Versus Exogenous Reserves

One significant risk factor for stablecoins is chain reactions between fiat currencies, crypto markets, and their reserve asset types; the less linkage between these, the stronger the risk segregation.

Centralized stablecoins such as USDC, USDT, and USDP are reserved with cash or cash-equivalent assets such as short-term treasury bills and commercial bank deposits. These assets are both a) exogenous to crypto markets and b) exogenous to the stablecoin issuers.

MakerDAO (Dai) bootstrapped using ETH as its primary collateral; although ETH is endogenous to crypto markets, it is at least exogenous to the MakerDAO ecosystem. Continued MakerDAO developments led to USDC becoming more than half of the protocol's collateral. Although this has made MakerDAO more centralized, it has also allowed MakerDAO to have half of its collateral in an asset (USDC) that is both a) exogenous to crypto markets and b) exogenous to the MakerDAO ecosystem.

Frax bootstrapped using USDC as its reserve almost entirely but incorporated designs that would deliberately and algorithmically introduce its native token FXS as a part of its reserve. FXS is both a) endogenous to crypto markets and b) endogenous to the Frax ecosystem. At the time of writing, Frax has roughly 8% of its reserve in FXS.

| Stablecoin | Collateral / Reserve | Exogenous to the Protocol | Exogenous to Crypto Markets | Risk Segregation |
|---|---|---|---|---|
| USDC, USDT | Short-term US treasury bills, commercial bank deposits | Yes | Yes | 2/2 |
| Dai | ETH, wBTC, other cryptos, liquidity pools (40%) | Yes | No | 1/2 |
| | USDC (58%) | Yes | Yes | 1.8/2 |
| | Real-World Assets (0.6%) | Yes | Yes | 2/2 |
| Frax | FXS (8%) | No | No | 0/2 |
| | USDC held directly by Frax (1%) | Yes | Yes | 2/2 |
| | USDCs held indirectly by Frax via Automated-Market Operations (91%) | Yes | Yes but with exposure to external yield venues | 1.5/2 |

*Figure 15: The "Exogenous vs. Endogenous" perspective on different stablecoin designs.*

Unitas Foundation will leverage this new perspective to analyze different USDPEGs' risks. It will also examine non-stablecoin reserve types such as ETH, CBDCs and tokenized treasury bills, and develop a multi-reserve strategy to further diversify its risks and stabilize its pegs.

**Due to the high risks involved with endogenously-reserved (aka "algorithmic") assets, Unitas is committed to keeping its reserve and collateral free of its own endogenous asset, such as 4REX.**

## 11.2 Progressive Decentralization

Decentralized money has many advantages over centralized money: transparency, censorship resistance, inclusivity, and fairness. The stablecoin industry is progressively achieving decentralization, which we define using three distinct aspects.

### 11.2.1 Ledger Decentralization

While USD exists on centralized ledgers, first-generation (centralized) stablecoins (e.g., USDC, USDT, USDP) exist on permissionless, decentralized ledgers. Anyone can analyze the minting, burning, and movement of these stablecoins. Anyone can set up a self-custodial wallet and hold and use them. In this respect, the name *"centralized stablecoins"* given to USDC, USDT, and USDP understates their efforts towards a decentralized future.

"Centralized stablecoins" was, in fact, one of the most solid product-market fits for decentralized ledgers.

### 11.2.2 Operations Decentralization

While their tokens reside on decentralized ledgers, the operations of centralized stablecoins often remain black boxes to outsiders. Market participants receive little insight into their reserve operations – which banks hold their reserves, what commercial papers they hold, etc. Everyone is forced to trust audit reports published by their chosen auditors.

Compared to centralized stablecoins, DeFi stablecoins such as Dai and Frax implement most of their operations in smart contracts, which provide excellent transparency to market participants, and have fueled the growth of collaboratively organized analytics dashboards such as Dune. Anyone is free to analyze a DeFi stablecoin's reserve operations.

*Report on Stablecoins* published by the President's Working Group on Financial Markets (PWG, joined by the FDIC and the OCC) [49] highlighted another essential concern – operational availability and continuity. During market turmoil and liquidity stress, simply having enough reserves isn't enough. To bolster market confidence, stablecoin issuers must provide excellent liquidity, which hinges on their operational excellence.

In contrast, DeFi stablecoins are inherently better with operational continuity thanks to most of their operations being implemented in smart contracts and executed mechanically without requiring human intervention.

### 11.2.3 Decision-Making Decentralization

The proliferation of DAO tools at every layer of the DAO stack [50] [51] [52] has helped accelerate DAO and SubDAO implementations across a broad spectrum of decentralized organizations. MakerDAO has demonstrated how the decision-making of a stablecoin organization can progressively become decentralized. At writing, MakerDAO is one of the most decentralized protocols. Although other DeFi stablecoin projects have accomplished ledger and operation decentralization, their decision-making remains relatively centralized at writing.

To grow a fair and inclusive money system and unleash the full potential of DeFi, Unitas Foundation aims to progressively decentralize all its aspects – ledger, operations, and decision-making. We welcome you to join this inclusive finance journey with us.

# 12 Join the Movement

Stablecoins are one of blockchain's most solid product-market fits, and Unitas Protocol presents a unique opportunity to accelerate financial inclusion in emerging markets.

We encourage you to:

1) ask us questions about stablecoins or Unitas over Telegram,
2) learn more about stablecoins or Unitas from our Wiki, or
3) follow us on Twitter.

Please join us! Unitas Foundation is actively seeking collaborators including (but not limited to):

- People passionate about decentralized finance and stablecoins
- People passionate about improving financial inclusion in emerging markets
- Cross-border payment professionals and organizations
- Stablecoin issuers (centralized and decentralized)
- Exchanges (CEXs and DEXs)
- Market makers (CeFi and DeFi)
- OTC desks, money transfer operators (MTOs), money service organizations (MSOs) and liquidity providers
- Digital wallets, payment providers and credit card issuers
- Banks and electronic money institutions (EMIs)
- Central banks
- Lenders and lending platforms (CeFi and DeFi)
- Cross-border trade associations
- Regulators and law enforcement
- Compliance and legal professionals
- Economics and structured finance experts
- DeFi and smart contract developers
- Games wanting to adopt stablecoins

Let’s build the decentralized finance infrastructure that will unleash emerging market potentials!

# References


[1] Sema Y. Genç, bSeda Aydin (2018.). *Modern Journal of Language Teaching Methods (ISSN: 2251-6204), vol. 8, no. 4, pp. 17-22.* "Hegemony War In Bretton Woods: John Magnard Keynes And Harry Dexter White."

[2] MakerDAO, (2020, Feb.). "The Maker Protocol: MakerDAO's Multi-Collateral Dai (MCD) System."

[3] Wikipedia, "USD Coin."

[4] Sam Kazemian (2019, Sep 20). *Ethereum Research.* "DeFi Algorithmic Stablecoin: FRAX (feedback wanted)."

[5] Wikipedia, "Tether (cryptocurrency)."

[6] (2021, Nov 17). *Binance Blog.* "BUSD: All You Need To Know About the Stablecoin."

[7] Charles Cascarilla (2021, Aug 24). "Pax Dollar (USDP) Whitepaper v2.01."

[8] Vivek Mishra (2021, Nov 3). *Reuters.* "More trouble ahead for erratic emerging market currencies: Reuters poll."

[9] IMF, (2021, Oct.). "World Economic Outlook Database, October 2021 Edition."

[10] Government of India, Department of Economic Affairs, (2022, Jan 20). "Economic Survey 2021-22 Statistical Appendix."

[11] Emele Onu (2022, Mar 8). *Bloomberg.* "Nigeria Companies to Pay Dollar Debts in Naira on Forex Shortage."

[12] Anusha Ondaatjie, Lilian Karunungan (2022, Mar 7). *Bloomberg.* "Sri Lanka's Debt Crisis Lingers as Foreign-Currency Reserves Slip."

[13] Ebru Tuncay (2022, Mar 9). *Reuters.* "Turkish banks hike forex deposit rates to meet shortage – sources."

[14] Samia Nakhoul, Tom Perry (2022, Feb 1). *Reuters.* "Lebanon plan sees 93% currency slide, turns bulk of FX deposits to pounds."

[15] Dalal Saoud (2021, Dec 21). *United Press International.* "Depositors turn to courts to free money from Lebanese banks."

[16] Rajendra Jadhav, Nupur Anand (2019, Oct 1). *Reuters.* "India's PMC Bank created over 21,000 fake accounts to hide loans: complaint."

[17] Angle Labs Core Team, (2021, Jul 7). "Angle Whitepaper -- A Decentralized, Over-Collateralized and Capital Efficient Stablecoin Protocol."

[18] Mariano Conti "daistats.com."

[19] Sam MacPherson (2020, Nov 9). "MIP29: Peg Stability Module."

[20] Leonides Buencamino, Sergei Gorbunov (2002, Nov.). *United Nations Department of Economic and Social Affairs.* "Informal Money Transfer Systems: Opportunities and Challenges for Development Finance."

[21] Dina Habjouqa, Joanna Clendinning (2020, Sep 8). *Dow Jones.* "Hawala: Ancient Money Transfer System Poses Very Modern Risks."

[22] Chainlink "Chainlink FRAX/USD Data Feed."

[23] Lorenz Breidenbach, Christian Cachin, Benedict Chan, Alex Coventry, Steve Ellis, Ari Juels, Farinaz Koushanfar, Andrew Miller, Brendan Magauran, Daniel Moroz, Sergey Nazarov, Alexandru Topliceanu, Florian Tram`er, Fan Zhang (2021, Apr 15). "Chainlink 2.0: Next Steps in the Evolution of Decentralized Oracle Networks."

[24] Erik Feyen, Jon Frost, Harish Natarajan, Rara Rice (2021, Oct.). *Bank for International Settlements.* "BIS Working Papers No 973 -- What does digital money mean for emerging market and developing economies?."

[25] Financial Stability Board, (2020, Apr 9). "Enhancing Cross-border Payments -- Stage 1 report to the G20: Technical background report."

[26] Reserve Bank Staff College, (2020, Aug 7). "Functions and Working of RBI."

[27] Reserve Bank of India, (2021, May 17). "Reserve Bank of India Annual Report 2020-2021."

[28] Florian Seeh (2021, Feb 23). *Ernst & Young.* "How new entrants are redefining cross-border payments."

[29] Dawei Wang (2021, Jul 9). *Thunes.* "Demystifying cross-border payments – Part Two."

[30] Stephen Ferry (2019, Aug.). *Pay360.* "Why alternative payment methods (APMs) are no longer an optional extra."

[31] David Zhang (2020, Jul 27). "Curve.fi 101 — How it works and its meteoric rise."

[32] Curve Finance, (2022, Feb 25). "Curve Documentation -- Release 1.0.0."

[33] Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, Dan Robinson (2021, Mar.). "Uniswap v3 Core."

[34] Sushi, (2020, Sep 9). "The SushiSwap Project -- An evolution of Uniswap with SUSHI tokenomics."

[35] Fernando Martinelli, Nikolai Mushegian (2019, Sep 19). *Balancer.* "A non-custodial portfolio manager, liquidity provider, and price sensor."

[36] Stanislav Kozlovski (2021, Mar 27). "Balancer V2 — A One-Stop-Shop; How Balancer V2 Has Everything You Would Want From An AMM."

[37] Robert Sams (2014, Oct 24). "A Note on Cryptocurrency Stabilisation: Seigniorage Shares."

[38] Owen Fernau (2021, Oct 14). *The Defiant.* " DeFi 2.0 Wave of New Projects Test Liquidity Mining Alternatives."

[39] Andrew Nardez (2021, Dec 27). "What is Protocol owned liquidity? A Primer on the model developed by Olympus DAO."

[40] Ben Hauser (2021, Mar 9). "Curve DAO: Vote-Escrowed CRV."

[41] @0xJiji, @banteg, @daryllautk, @HAtTip3675, @onlylarping, @vany365, @Wot_Is_Goin_OnYIP-65 (2021, Dec 19). "YIP-65: Evolving YFI Tokenomics. Yearn Finance."

[42] Andre Cronje (2022, Jan 6). "ve(3,3)."

[43] Juan Pellicer (2022, Jan 12). *IntoTheBlock.* "Understanding veNomics – Why the “veTKN” economic model for tokens is gaining traction amongst DeFi protocols."

[44] Robert Leshner, Geoffrey Hayes (2019, Feb.). "Compound: The Money Market Protocol."

[45] AAVE (2020, Jan.). "AAVE Protocol Whitepaper V1.0."

[46] AAVE (2020, Dec.). "AAVE Protocol Whitepaper V2.0."

[47] Chainlink, "Chainlink INR / USD Price Feed."

[48] CoinDCX, "USD Coin to Indian Rupee Price."

[49] President’s Working Group on Financial Markets (PWG), FDIC, OCC, (2021.). "Report on Stablecoins."

[50] Steven McKie (2019, Mar 24). "The Year of the DAO Comeback -- Understanding What’s Leading the Charge."

[51] Cooper Turley (2021, Jun 25). "DAO Landscape."

[52] Cooper Turley (2021, Nov 30). "How to SubDAO."