Amp: A digital collateral token to enable immediate settlement of payment transactions

November 24, 2020
amptoken.org

Abstract

Digital assets are quickly becoming the predominant medium of exchange for global commerce, but their universal acceptance is limited by the high cost of transaction validation. The key to unlocking ubiquitous digital payments is to efficiently mitigate the uncertainty of achieving transaction finality. These problems are economically resolved through an open, extensible collateral system utilizing public verification of state via distributed convergence mechanisms. Amp is a collateral token designed to decentralize the risk in a payment transaction, dramatically reducing the assurance cost from existing counterparty networks.

Amp incorporates a novel partition interface within an original framework of partition strategies to facilitate the interoperability of staking contracts for any surety mechanism. Using specific partition strategies, Amp can enable tokens to be conditionally allocated as collateral without requiring transfers to another smart contract. In this way, the system preserves asset custody, substantially improving the simplicity and safety of staking collateral.

Within distributed tokenized financial networks, Amp serves as a medium for accruing value while aligning the incentives of all participants. This is achieved via virtuous feedback loops of increasing spending capacity coupled with a non-inflationary reward distribution. Fundamental economic models are derived to demonstrate that Amp functions as low-volatility collateral, with its value compounding exclusively as a result of the utility it provides. By enabling decentralized ownership and participation in financial networks, applications built on Amp can become orders of magnitude more cost-efficient than existing systems, and help eliminate the overwhelming deadweight loss of traditional social and economic structures for financial transactions.

Contents

1 Introduction

Global payment networks were created by an alliance of the largest financial institutions in the world. After decades of operating with resilient oligopoly power, these networks have designed impenetrable barriers to entry by implementing closed and inaccessible infrastructure, capital clearing requirements, commercial complexity, and restrictive counterparty fragmentation. This has resulted in a platform optimized for rent-seeking and scale, but susceptible to fraud and irreversible social cost.

The typical network architecture for state-of-the-art merchant payment systems includes a linear sequence of service providers (*e.g.,* gateway, acquirer, processor, issuer), each maintaining their own data repositories and bespoke data security (*e.g.,* PCI DSS[1], GDPR[2]) environments. In order to conduct a financial transaction through these systems, sensitive payment information must be interpreted and analyzed by each service provider, often in-the-clear, resulting in inevitable data breaches that lead to identity theft and fraud loss on a massive scale. Despite international regulatory structures that incentivize financial institutions to further develop the speed, efficiency, and cost-effectiveness of their own systems, the majority of technological advancement in recent decades has been driven by private-sector technology companies. Still, financial transactions today are plagued by convoluted pricing models and conflicting specifications[3] for integrated circuit, contactless, and QR code implementations; due to these complex interdependencies and the prevalence of fraud, merchant settlement requires an average of two days for deposits (minus restricted rolling reserves) and three to six months for dispute finality. Most merchants have no alternative but to accept the myriad integration[3] and compliance costs, fees, and fraud liabilities continuously imposed by payment networks; a distinct minority have opted to develop their own proprietary payment interfaces that sideline the existing networks entirely.[4] As a result, merchants shoulder the burden in funding layers of global payment services, but the much greater deadweight loss is ultimately borne collectively.[5,6]

The primary function of existing payment networks is to *only* facilitate transaction-related messaging, while relying on financial institutions (associations, issuing banks) to mitigate risk for both the merchant and payer. At a fundamental level, the legitimacy and fungibility of money requires universal verification. Unequivocally, this is the primary value of distributed ledger technologies; this singular feature has the potential to open financial infrastructure and eliminate

[1]Payment Card Industry Data Security Standard (PCI DSS) [49]

[2]General Data Protection Regulation (GDPR) requirements [26]

[3]EMVCo, the privately-owned consortium created by payment networks, currently manages seven overlapping specifications for various forms of payment [20]

[4]Merchant Customer Exchange and various sole-entity payment services currently live or in development [64]

[5]Estimated 2% of United States GDP, Measuring the Costs of Retail Payment Methods [34]

[6]Estimated 1% of European Union GDP, Social Cost of Retail Payment Instruments [52]

the fraud and cost persistent within the existing oligopoly. To substantially increase commerce efficiency, the centralization of various service providers within the process is immaterial – the *risk* needs to be decentralized.

Amp is a digital token designed to universally decentralize risk in a financial transaction. It includes a novel interface to allocate condition-specific collateral for payments with potential Byzantine participants. Economically, Amp also serves as a vehicle for accruing value within a collateralized network, aligning the interests of *all* participants. By enabling decentralized ownership and participation, a new payment network has the potential to become orders of magnitude more efficient [15].

2 Digital payments and the Flexa network

Flexa is a merchant payment network designed to enable universal acceptance of digital assets. Payments for goods and services are authorized instantly (in-store or online) without fraud and at net cost less than interchange. The network includes various exchanges and financial institutions to provide compliant settlement[7] across multiple jurisdictions.[8] Flexa integrates natively with existing point of sale (POS) systems[9] and online platforms to enable payment in a typical checkout experience. The network Spend SDK is permissionless; mobile wallets or applications can create unique, interoperable authorization codes for conveyance.[10] In order to unconditionally and immediately guarantee all merchant payments *without* trusting external protocols and network participants, decentralized collateral is the critical foundation of Flexa. By requiring each transaction to be fully collateralized, the predominant costs associated with the challenges of funds verification and payment fraud are eliminated.

The Amp token serves as the singular type of collateral within Flexa to decentralize risk within the network. To enable payment functionality, applications and communities can collectively stake Amp tokens on behalf of users. As incentive for supplying collateral, the entirety of network transaction revenue funds the continuous open-market purchase of Amp tokens for redistribution as network rewards. Flexa effectively decentralizes transaction insurance, decoupling merchant settlement from the initial consumer payment to provide immediate *finality-as-a-service*.

2.1 Prevalence of digital assets

Physical cash is effectively unusable online, but meaningful digital proxies are quickly evolving, facilitated by the growth of electronic and contactless payments. Billions of people currently use

[7]Flexa offers merchant settlement via digital assets or fiat bank transfers

[8]As of September 2020 Flexa is permitted to operate in the United States and Canada

[9]Compatible with ISO/IEC 8583 messaging standard [40]

[10]Digital scans via backwards compatible continuous/discrete symbologies (*e.g.,* Code 128)

mobile apps for closed-network P2P payments and bank transfers, assuring digital payments will soon become ubiquitous. This is more evident when considering the recent proliferation of digital stored-value payments, loyalty/point programs, and direct-to-consumer incentives. Additionally, record low interest rates and unprecedented levels of fiscal stimulus have attracted considerable attention to new digital asset classes. The popularity of speculative token networks has also fostered a spectrum of innovative projects with features specific to spending utility, *inter alia*, low-volatility protocols, pegged assets, and synthetic commodities. Decentralized finance communities continue to accelerate mainstream adoption through robust capital services, liquidity protocols, and novel intra-platform incentive mechanisms. The inevitable release of digital currencies by sovereign nations, financial institutions, social platforms, and corporate consortiums further defines the evolution of international commerce.

3 Decentralized collateral

With macroeconomic demand for an array of numeraire goods, Flexa is designed to support many types of digital assets; Amp as decentralized collateral completely abstracts the finality risk from the merchant transaction, providing a universal medium-of-exchange framework. With traditional payment networks, verifying the state of digital assets is a complex and expensive process. This is compounded as merchants scale and provide international services, and prohibits acceptance of a variety of available assets (§2.1). Accordingly, transactions require intermediaries to provide third-party verification of sufficient funds, exchange rates, and authenticity of assets. Decentralized networks can uniquely allow for independent verification of state via open validator sets and distributed convergence mechanisms. This dramatically lowers the cost of verification, while also eliminating fraud, information asymmetry, and moral hazard risk. With a universal foundation of trust, digital assets can be safely authenticated and used more broadly in commerce. An open collateral system can be used to secure all payment transactions in a financial network, with all participants able to transparently verify a spectrum of digital assets. In this manner, decentralized collateral serves to remove expensive intermediaries, and efficiently distributes risk.

3.1 Finality assurance and scale

Within a distributed ledger technology (DLT) platform, a finality *guarantee* is that well-formed blocks will not be revoked from the chain at a future point, ensuring that transactions are permanent and can be trusted. However, in the absence of an organization endorsing transactions, *absolute* finality generally cannot be achieved regardless of the consensus mechanism. Transactions are typically considered irrevocable through various degrees of *probabilistic* finality, an empirical requirement of network block confirmations. A more pragmatic approach is *economic* finality, wherein requisite confirmations are based on transaction value and the explicit cost in updating

the ledger versus the potential yield from its reversal [10]. DLT assets vary tremendously in the time-inclusive economic quality of ledger security, resulting in discontinuities in measuring finality assurance. Digital retail payments require real-time settlement and universal economic finality; DLT native transactions are generally not feasible at scale.

3.1.1 Proof-of-work and alternative scaling layers

Retail payments are impractical using `0/unconfirmed` transactions due to double spend exploits or explicit/inherited replace-by-fee. Finality is achieved probabilistically and is insufficient for in-person payments due to network latency. At the scale of global commerce, reorganization via majority attacks is also possible under certain economic conditions. Scaling layers are intended to provide more immediate, localized finality, but are generally not designed to reach assured finality (*e.g.,* settlement inevitably requires on-chain transactions). Off-chain bilateral ledgers with hashed time lock contracts (HTLCs) are inefficient for one-time payments, requiring prospective-cost security deposits, and introducing non-trivial complexity of opening/closing states. Sequential payments are not possible within the same channel, and locked funds in multi-hop transactions (especially with sequential HTLCs in parallel channels) elicit untenable griefing attacks. Specific to retail payments, free-option problems and dispute windows upon settlement create inaccurate finality assumptions. Commitments to multiparty off-chain state require participants to fully validate all computations and remain online; otherwise, intermediary nodes with autonomous access to private keys (to rebalance channels) are required. The practical usability of retail digital payments is often trivialized; DLT transactions are not merely replicable data. Existing methods of scaling data transfer (*e.g.,* packet switching) are not entirely applicable to one-way discrete transfers of ownership. In this sense, payments represent *immutable value*, a fundamentally more complex problem.

3.1.2 Proof-of-stake and high throughput networks

Proof-of-stake consensus algorithms attempt to reach absolute finality at the base layer with high levels of transaction throughput. The source of finality in a PoS blockchain ledger is derived from validator assurances of its integrity. While these networks[11] generally provide faster economic finality than PoW at scale, for payment network utility there exist myriad attack vectors and misaligned incentives. Validators themselves create new vulnerabilities such as precomputation attacks, stake bleeding, selfish endorsing, and P+epsilon attacks. Connectivity issues and node

[11]Applicable to the spectrum of existing PoS consensus protocols for linear chains and Directed Acyclic Graph (DAG) structures (*e.g.,* Delegated Proof-of-Stake (DPoS), Nominated Proof-of-Stake (NPoS), Proof-of-Authority (PoA), Proof-of-History (PoH), and Asynchronous Byzantine Fault Tolerance (aBFT)). The specific implementations are beyond the scope of this paper.

desynchronizations can momentarily disrupt the validation process; weak subjectivity is excellent for longer time spans but generally ineffective for instantaneous consensus. Low latency protocols tend to be effectively centralized *(e.g.,* utilizing collectively trusted sub-networks or membership nodes) or susceptible to collusion due to short-term metastability. At retail scale, the economic incentive for long-range attacks and posterior corruption also becomes non-trivial. PoS networks are inherently vulnerable since native tokens are liquid; validators have low opportunity cost to sell assets due to minimal infrastructure expenditure. Additionally, networks that assert immediate and absolute finality also have administrative exceptions to create discretionary ledger modifications.

3.1.3 Merchant acceptance

Native DLT-based settlement at scale is beyond the economic reality for multinational merchants. Finality assurance is paramount to mitigate coordinated fraud (*e.g.* simultaneous transactions online) and the financial incentives for attack. Beyond the lack of universal assurance, digital asset payments are also limited by deposit requirements, security concerns, regulatory uncertainty, and volatility. Operational complexity due to tax and accounting complications is intractable, especially for synthetic assets that consistently rebase units of account. Sustainable PoW/PoS protocols are not designed for instant, absolute finality. However, with sufficient duration, persistent economic finality is achieved based on a variety of empirical network factors (*e.g.,* consensus protocol, validator decentralization, hardware requirements, transaction value, ledger settlement cost). Collateral allows for the entirety of transactions (regardless of consensus scheme) to reach appropriate levels of economic finality while providing immediate finality from a merchant perspective.

3.2 Meta-staking and risk distribution

To access the Flexa network, applications can supply Amp to a designated smart contract. In this implementation, collateral is supplied via *meta-staking*; participants stake Amp into pools that secure the network. Collateral pools are permissionless and participants can supply/withdraw without time, financial, or competitive restriction. Network rewards are distributed *pro rata* within the pool, self-enforcing the decentralization of risk. The Amp token contract is immutable (*i.e.,* no administrative privileges exist), ensuring arbitrary collateral managers can perform various delegation functions (§4.2.2). For instance, custodial platforms can create interfaces for non-technical users to easily stake Amp. Meta-staking only involves contract execution, not requiring configuration processes, node/server hosting, or validator service. This is accomplished by calling the standard `transfer` method [23] or by using the novel, partition-based implementation `transferByPartition` to grant conditional access rights within the token contract

itself (§4.2.1). When using `transferByPartition`, participants can collateralize the network *while maintaining* custody of Amp tokens (*i.e.*, the tokens are not transferred), analogous to vote delegation in decentralized governance systems.

3.2.1 Microeconomic utility

Software wallets can provide spending utility to users by integrating the network Spend SDK and staking Amp. Network rewards are based on transaction volume, so wallets have continuous financial incentive (coalesced with Amp capital cost) to be rational actors. Individuals within the ecosystem may also participate in collateral pools for financial incentive and to provide spending capacity for desired assets or communities. Amp holders individually choose which wallets to collateralize, and earn network fees, augmenting token demand.

The Amp token partition interface enables efficient asset utilization by implementing novel *partition strategies* (§4.2), granting stake permissions rather than transfers, mint contracts, or proxy assets. Holders can maintain custody of assets while staking, eliminating the risk of destination errors (*e.g.*, sending directly to a token contract address) and potential loss of funds due to insecure contracts or compromised gateways. Further, Amp can also temporarily collateralize unrelated transactions in discrete instances such as deposits/withdrawals on exchanges, or the acceleration of margin relief. This transitivity explicitly allows for greater decentralization and network resilience, in addition to ideal technical and economic utility, similar to yield-generating tokens minted from liquidity provider contracts. The economic value of Amp is underpinned through its utility within the Flexa network, but holders minimize potential time-value opportunity cost.

3.2.2 Collateral integrity

A purpose-built staking token enables maximum technical extensibility while reducing integration complexity and multivariate attack surfaces. Amp is also critical for capturing the entirety of the value that Flexa creates via virtuous feedback cycles, continuously increasing spending capacity and counteracting volatility (§5.4). However, network utility is exclusively dependent on ensuring the integrity of the collateral itself. The economic framework for meta-staking ensures that the mechanics for verifying and valuing Amp are transparent, creating opportunities for sufficient liquidity. The market supply of Amp is distributed within the ecosystem and staked as collateral for wallets to provide payment services. By comparison, the inductive incentives of this scheme diverge from traditional PoS networks. Most PoS participants lack non-financial utility from validating transactions, optimizing instead for financial yield. Accordingly, valuation models for staking tokens approximate only the net present value of future transaction fees. Staking tokens actively held in an inflationary protocol are numerically dilutive, and the network is not a long-

term store of value. Amp was designed for the specific utility of collateral: token supply is fixed, and network rewards are *non*–zero-sum distributions (§5.2).

The value of Amp is derived entirely from its use, and successful collateralized transactions result in open-market token purchases. Other collateralized transactions can be subject to pro-cyclicality and uncorrelated supply and demand; multi-collateral derivatives exponentially compound these effects. By insulating the network with one collateral type, risk parameters are constrained to provide more efficient network stability. If a negative price shock occurs, more network-specific collateral needs to be staked (*i.e.*, increasing capacity), stabilizing price movement. Also, since collateral value is directly correlated to network spending volume, there is low covariance between a discrete transaction not settling (*i.e.*, resulting in consumed stake) and Amp utility value. Persistent collateral loss would be suboptimal *a priori;* defecting at scale devalues the entirety of self-supplied collateral, notwithstanding the significant opportunity cost of initial token acquisition. The re-deployability of an asset is also a major factor of its collateral quality; Amp is valuable to be sold intra-network because it provides staking utility. In the event of hypothecation, Amp is autonomously sold to the open-market, and cyclically repurchased as rewards. Ultimately, consumed Amp is transferred to the participants most financially motivated to use it, completely avoiding deadweight transfer cost. Thus, the Flexa network ensures a reliable collateral asset with low volatility is created via micro-prudential efficiencies in obtaining/staking Amp for Spend SDK pools, and macro-prudential distribution of rewards to ensure aggregate collateral liquidity in public markets.

4 Amp token contract

Amp is an ERC20-compatible [23] token that implements conditional access rights via smart contracts within a partition scheme. The token interface allows for universal interoperability with external transaction protocols. A conventional ERC20 token assigns balances to hexadecimal Ethereum addresses, and the aggregate amount of those balances is the total supply of the token.

Table 1: Token balance distribution for arbitrary addresses (truncated).

Address	**Balance**
0x67b1...4331	100
0xd2b9...6fdd	200
0x4ffb...3f00	300

Amp resembles a rudimentary token in that balances are assigned to Ethereum addresses, but the tokens also belong to a particular 32-byte partition which effectively serves as a second-dimension in the distribution array of the balances. Tokens are not reciprocal across partitions (consistent

with address parameters), so the sum of the balances across all addresses *and* partitions is the total supply of the token.

Table 2: Token balance distribution for addresses (column 1) and partitions.

Address | Partition	**0x0000...0000**	**0x0a65...2da9**	**0x6c1a...4750**
0x67b1...4331	0	50	50
0xd2b9...6fdd	100	100	0
0x4ffb...3f00	300	0	0

Amp supports transfers between addresses and partitions through the `transferByPartition` function, and includes the `approveByPartition` function that authorizes an address to transfer tokens on behalf of the caller, but only from a particular partition. To maintain backwards compatibility with ERC20, the *zero partition* (*i.e.,* the *default partition*) is used for all ERC20 operations, including `transfer` and `approve`. In Table 2, address `0xd2b9...6fdd` cumulatively holds 200 tokens, but can only transfer 100 tokens via the ERC20 `transfer` method. The `transferByPartition` function could be used to transfer the 100 tokens allocated to the `0x0a65...2da9` partition.

For tracking balances off-chain, two events are emitted with every transfer:

- Transfer: contains the to and from addresses, as well as the amount. If the transfer only changes the partition and not the address, the event will be emitted, but the principal addresses will be the same.
- TransferByPartition: contains the same data as Transfer, as well as the `to` and `from` partitions and any metadata or `operatorData` parameters (§4.1).

4.1 Operators and partition scopes

In addition to the `approve` and `approveByPartition` functions that authorize transfers up to a maximum amount, holders can designate an *operator* for their tokens within a particular partition. This grants the operator the ability (until revoked) to transfer an unlimited number of tokens from the delegating address and partition. The partitions themselves have semantic meaning. The first 4 bytes of the partition (the *partition prefix*) correspond to a *partition scope*, which can be used to apply a custom set of rules to transfers to and from partitions in the space (*e.g.*, a partition prefix that matches the partition scope). Below is a representation with `aff8…ed6b` partition and prefix of `aff82582`.

```
      Prefix
    |-------|
0x aff82582 98ef1148f5e95598d0dde87c55853a9207f3c0d94ff43c33c517ed6b
```

4.2 Partition strategies

The core innovation of Amp is the *partition strategy*, an external contract that implements the `IAmpPartitionStrategyValidator` interface, which can be encoded with special rules (*e.g.,* to automatically grant operator statuses, call other hooks, and authorize discrete transfers based on external oracles). Partition strategies can be used to systematically grant *controller*-like permissions to various actors in the ecosystem. This enables the Amp contract to execute common implementation situations for collateral managers, creating a more efficient developer experience and permitting additional trust to be incorporated within the contract itself. To support flexibility in strategy implementations, the `transferByPartition` method contains an open `operatorData` field; this can be used by callers to embed data for the partition strategy used to validate transfers. As external communities identify new approaches and advancements within base layer technologies, entirely new strategies can be assigned to partition prefixes via the `setPartitionStrategy` function. The set of partition strategies is append-only (*i.e.,* the base rules for a partition are immutable), and a strategy can never be set for the zero prefix `0x00000000`, as this scope includes the default partition used for ERC20 compatibility. Transfers to partitions within a scope but without an assigned strategy is disallowed. This prevents a strategy from being added that would retroactively control tokens within its partition scope. Beyond defining strategies for new prefixes, the Amp contract owner has no ability to limit Amp transfers between addresses or partitions.

4.2.1 Distinct partition validator

The distinct partitional validator is a partition strategy that defines a collateral manager with control of tokens within the `0xaaaaaaaa` partition scope. A collateral manager is an external contract that has operator permissions on a subset of the partitions within the scope. The partition itself defines the delegated collateral manager `HolderCollateralPartitionValidator` as an operator.

Below is a partition with the 4-byte prefix `0xaaaaaaaa`. The 20-byte suffix is the address of the collateral manager, also called the *partition owner*. This implies that the contract `0xec9f…c4ff` is an operator on all `0xaaaaaaaa` partitions that include its address. The 8 bytes between the prefix and suffix (the sub-partition) can be used in any way the collateral manager wishes to manage the tokens.

```
      Prefix   Sub-partition         Collateral manager address
    |-------|----------------|----------------------------------------|
  0x aaaaaaaa 088d937174315e03 ec9f0d42921543787bfefd83d0f119284b3ec4ff
```

This scheme allows a single collateral manager to control a large number (2^{64} or 1.8×10^{19}) of partitions. Upon a transfer from a partition in the scope of this strategy, the `AmpTokensSender` transfer hook of the partition owner will be called. This enables the collateral manager to restrict a transfer, even if it is not *from* its address, if any tokens in its collateral partitions should not be moved by the holder (*e.g.,* due to staking conditions). The partition owner is given permission to call `transferByPartition` for any address for any partition within its owned space. This strategy allows for a *stake-in-place* collateralization mechanism where a holder retains the tokens at their address while simultaneously providing the tokens as collateral to the delegated manager. Any changes that affect the balance (*e.g.*, if staking rewards are granted or collateral is consumed) are reflected directly in the partition by on-chain transfers executed by the collateral manager.

In the example shown in Table 3, collateral manager `0xec9f...c4ff` controls 300 tokens (denoted with an asterisk*) across two partitions; the amount staked by each address in each partition is recorded independently. The three addresses have total balances of 100, 200 and 300 tokens, respectively, and each have delegated control of only a portion of their tokens to the collateral manager.

Table 3: Token balance distributions with partitions and collateral manager.

Address | Partition	**0x0000...0000**	**0x0a65...2da9**	**0xaaaa...c4ff**	**0xaaaa...c4ff**
0x67b1...4331	0	50	50*	0*
0xd2b9...6fdd	0	50	50*	100*
0x4ffb...3f00	200	0	0*	100*
0xec9f...c4ff	1000	0	0*	0*

The collateral manager itself holds 1000 tokens in the default partition, which it could use to grant additional tokens as network rewards to stakers. For example, if `0xec9f...c4ff` granted 25 tokens to each non-zero staked address-partition combination, participants could observe their distributions on chain within the scope of their own addresses; Amp ensures the total supply remains unchanged. Table 4 shows collateral manager `0xec9f...c4ff` with control of 400 tokens across two partitions after staking rewards of 100 tokens have been distributed.

Table 4: Token balance distributions with rewards transfers.

Address | Partition	**0x0000...0000**	**0x0a65...2da9**	**0xaaaa...c4ff**	**0xaaaa...c4ff**
0x67b1...4331	0	50	75*	0*
0xd2b9...6fdd	0	50	75*	125*
0x4ffb...3f00	200	0	0*	125*
0xec9f...c4ff	900	0	0*	0*

To unstake, a holder can invoke the `transferByPartition` function to transfer the tokens from the `0xaaaaaaaa`-prefixed partition to the default partition (or any other partition). The collateral manager will receive the `tokensToTransfer` hook, which is capable of rejecting the operation if the transfer is not authorized. This could be used to enforce rules custom to the collateral manager, such as staking duration requirements or withdrawal limits.

4.2.2 Pool partition validator

The pool partition validator partition strategy defines a collateral manager with control of tokens within the `0xcccccccc` partition scope. Similar to the distinct partition validator (§4.2.1), the partition itself defines the delegated collateral manager `CollateralPoolPartitionValidator` as an operator.

Below is a partition with the 4-byte prefix `0xcccccccc`. The 20-byte suffix is the address of the collateral manager, also called the *partition owner*. This implies that the contract `0xb8fa…9a5a` is the collateral manager for all `0xcccccccc` partitions that include its address. The 8 bytes in between the prefix and suffix (the sub-partition) can be used in any way the collateral manager wishes to manage the tokens. In this strategy, tokens delegated to the collateral manager must be transferred to the collateral manager address, constituting a single pool of collateral per sub-partition.

```
     Prefix   Sub-partition         Collateral manager address
    |-------|----------------|----------------------------------------|
0x cccccccc 57d3df89104df9b6 b8fae86ffe3cf75123760d4c67936699a64d9a5a
```

In the example shown in Table 5, a total of 300 tokens have been delegated across two partitions belonging to the `0xb8fa...9a5a` collateral manager. The balances within each pool can be tracked within the collateral manager contract, or by off-chain methods. To maintain solvency (*i.e.*, ensuring cumulative staking rewards and principal is withdrawable by all participants), the collateral manager can transfer additional tokens to the partitions as required.

Table 5: Token balance distributions with partitions and collateral pool manager

Address | Partition	**0x0000...0000**	**0x0a65...2da9**	**0xcccc...9a5a**	**0xcccc...9a5a**
0x67b1...4331	0	50	0*	0*
0xd2b9...6fdd	0	50	0*	0*
0x4ffb...3f00	200	0	0*	0*
0xb8fa...9a5a	1000	0	100*	200*

Table 6 shows that the `0xb8fa...9a5a` manager has granted 50 tokens in rewards for each pool. The immutability of Amp total supply is guaranteed by the token contract, while participants can observe the overall pool balances increase. Every address is considered to be an operator on all partitions within the `0xcccccccc` prefix scope; limits on transfers out of these partitions are completely delegated to the collateral manager. Therefore, although Amp enforces standard limits on balances within a given address and partition (*i.e.*, transfers cannot exceed the balance), the collateral manager is responsible for tracking the amount transferrable out of `0xb8fa...9a5a` partitions for individual stakers.

Table 6: Token balance distributions with pool-based rewards transfers.

Address | Partition	**0x0000...0000**	**0x0a65...2da9**	**0xcccc...9a5a**	**0xcccc...9a5a**
0x67b1...4331	0	50	0*	0*
0xd2b9...6fdd	0	50	0*	0*
0x4ffb...3f00	200	0	0*	0*
0xb8fa...9a5a	900	0	150*	250*

Although collateral manager implementations using the pool strategy have more complexity than distinct strategies, there are opportunities for augmenting throughput as illustrated in the reference Flexa collateral manager implementation (§4.4).

4.3 Token hooks

Amp supports token transfer hooks[13] on chain, with time-of-transfer calls to external (non-Amp) smart contracts that are configured to receive and react to individual transfer operations. All metadata included with the Amp transfer (*e.g.,* `from`, `to`, operator, amount, and partitions) are included as parameters in the transfer hook calls. This enables the hook implementations to act/react on the full breadth and scope of individual transfers. For instance, the `data` and `operatorData` parameters included within the `transferByPartition` method are principal to the hook implementations.

All `transfer` and `transferByPartition` calls will invoke the following hooks if an implementation is present and has registered the supported interface. [25]

- `tokensToTransfer` is called on behalf of the token sender (`from` address). If an implementation of `AmpTokensSender` has been registered by or on behalf of the sender, this transfer hook will be called; this hook is generally used to gate/block a transfer.

[13]Inspired by concepts introduced in EIP777: ERC777 Token Standard [24]

- `tokensReceived` is called on behalf of the token receiver (`to` address). If an implementation of `AmpTokensRecipient` has been registered by or on behalf of the receiver, this transfer hook will be called. This hook is used to perform additional processing of transfer data such as storing on-chain versions or propagating the data to an off-chain system through the emission of a bespoke `event`, or reject an invalid transfer (*e.g.,* lack of appropriate `data` or unsupported partition).

In either case, both the sender and receiver hook implementation can `revert` the transaction. This is possible since ownership of the tokens is not retroactively removed (*i.e.,* a sender can block their own transfer, or the receiver can block the reception of Amp from a third party, but not *vice versa*). Token transfer hooks are not required for standard account transfers, but are critical for smart contract collateral managers to react to new collateral and perform scope specific features (*e.g.,* withdrawal, authorization and processing rewards).

4.4 Flexa collateral manager

In the reference implementation of the Flexa collateral manager staking Amp provides real-time finality assurance of network payments subject to the following constraints:

- The network must be able to liquidate supplied collateral if a payment is not settled.
- Amp rewards are distributed to suppliers based on successful payments facilitated by the staked collateral.

For Flexa enabled payments, the maximum payment rate, (and therefore collateral-related operations) may exceed the maximum transaction throughput of Ethereum, necessitating the use of the pool partition validator strategy. Network rewards are calculated on a per-transaction basis in an off-chain oracle system, and represented on chain using periodic batch transactions. As platform scalability increases, Flexa can release an updated collateral manager to leverage distinct collateral partition validator strategies, enabling simplified collateral transfers and on-chain balances.

4.4.1 Staking

To access the open network, each mobile wallet application generates a unique partition within the scope of the strategy and collateral manager to which application-specific rewards are deposited (successful settlement) and from which consumed collateral is transferred (failed settlement). These pools are represented as sub-partitions within the partition scope for the collateral manager.

```
    Prefix    Application          Collateral manager address
   |-------|----------------|----------------------------------------|
0x cccccccc 57d3df89104df9b6 b8fae86ffe3cf75123760d4c67936699a64d9a5a
0x cccccccc ed03f1d5186ea41a b8fae86ffe3cf75123760d4c67936699a64d9a5a
0x cccccccc 400f04ddc7aebbeb b8fae86ffe3cf75123760d4c67936699a64d9a5a
```

To stake collateral to a particular application on the Flexa network, Amp token holders transfer tokens to the partition corresponding to the desired application using the `transferByPartition` function. Valid partitions must be registered within the collateral manager, and transfers to partitions outside of the `allowlist` are blocked by the transfer validation hook from the Amp contract. There are no other restrictions on Amp token holders supplying tokens to the Flexa collateral manager.

4.4.2 Unstaking

Since all addresses are operators on the partitions within the collateral manager partition scope (per the collateral pool partition validator registration), any user can call `transferByPartition` on Amp, with the `from` address of the collateral manager, the `from` partition (a Flexa collateral pool), and any `to` address. The validation of the transfer includes calling the `tokensToTransfer` hook on the collateral manager to approve/disapprove the transfer.

To approve transfers, Flexa maintains a set of authorized outgoing transfers on the collateral manager contract. In order to handle frequent requests, withdrawal authorizations are hashed [4] within Merkle trees, and corresponding roots are published regularly within the contract itself. Proofs supplied to holders are not valid for subsequent trees, so a continuous set of valid roots is available to provide adequate time to execute withdrawals (inclusive of on-chain confirmation). To ensure a withdrawal is authorized and multiple valid roots can only be executed once, a withdrawal authorization ledger is maintained across the trees.

Withdrawal authorization ledger The transactions recorded within the Merkle tree represent an updated, withdrawable balance for a given account (*e.g.*, 300 wei of tokens at address `0x6c41...5b9e` and partition `0xcccc...f418`). If the account balance increases by 100 wei, another transaction will update the total balance of 400 wei. To ensure that multiple instances of balance updates cannot be claimed, each transaction includes a nonce and a maximum last-nonce. The latest used nonce for every address/partition combination is stored within the collateral manager contract. For the preceding example, the authorizations would be:

Withdrawal authorization 1:

```
Supplier: 0x6c41...5b9e
Partition: 0xcccc...f418
Balance: 300 wei
Nonce: 1
MaxLastNonce: 0
```

Withdrawal authorization 2:

```
Supplier: 0x6c41...5b9e
Partition: 0xcccc...f418
Balance: 400 wei
Nonce: 2
MaxLastNonce: 0
```

If the user executes a withdrawal based on either authorization above, the other is effectively invalidated, since `MaxLastNonce` will be exceeded. A new authorization with an increased maximum last-nonce will allow the user to execute a new withdrawal. If the balance of 300 wei was withdrawn, a new authorization can be made for the remaining 100 wei:

Withdrawal authorization 3:

```
Supplier: 0x6c41...5b9e
Partition: 0xcccc...f418
Balance: 100 wei
Nonce: 3
MaxLastNonce: 2
```

In addition to tracking the last withdrawal nonce used by each account and partition, the contract also tracks the cumulative sum of executed withdrawals. This is not applicable to the standard process, but is used for fallback withdrawals (§4.4.3).

Recording withdrawal authorizations When a withdrawal authorization balance is updated, its hash is stored in a Merkle tree along with previous balance updates for batch inclusion. The complete withdrawal authorization data included in each leaf node hash is:

- Supplier: the token staker address
- Partition: the partition from which the tokens will be withdrawn
- Token amount: the number of tokens to be authorized for withdrawal

- Maximum last nonce: the maximum nonce value of the last executed withdrawal authorization for the user and partition

The nonce is stored with the Merkle tree root, and is homogenous for all authorizations contained within the tree.

Executing a withdrawal After a collateral withdrawal has been authorized, the user can execute it using the `transferByPartition` method. This is similar to a standard transfer, except that the withdrawal authorization data is included in the `operatorData` field (an open `bytes` parameter) and is passed to the collateral manager for the explicit purpose of opening a channel to the supplier.

The data included in the field for withdrawals include:

- Withdrawal type (32-byte `0xaaaa...aaaa` to signal the transfer is a user withdrawal)
- Supplier
- Maximum last nonce
- Merkle tree proof

These values are encoded using the standard Ethereum ABI specification [55]. Note, this scheme is not enforceable by the Amp contract, but it is preferred for all collateral manager implementations to ensure compatibility with standard tools such as web3.js.

As an example, if the values were:

- Withdrawal type (32-byte `0xaaaa...aaaa`)
- Supplier (20-byte address `0xd0e3d9e8d595279615bb29884f2242ace5d8db33`)
- Maximum last nonce (uint256 `0`)
- Merkle tree proof (bytes32[] with a single element of `0xd29e4d9dd8f4cd0bcfd77e51c6143d1201dca026f71bb6e218054299302fdeb3`)

Then, the `operatorData` would be a 192-byte array containing the following:

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
000000000000000000000000d0e3d9e8d595279615bb29884f2242ace5d8db33
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000080
0000000000000000000000000000000000000000000000000000000000000001
d29e4d9dd8f4cd0bcfd77e51c6143d1201dca026f71bb6e218054299302fdeb3
```

The Amp contract passes the operator data (along with the source, destination, and number of tokens as separate parameters) to the collateral manager through the standard token transfer hooks, and the validation of the supplied data against the set of current Merkle roots is performed on chain. This approach demonstrates the extensibility of the open interface between Amp and various collateral manager contracts, and gives Flexa the ability to rapidly authorize transfers regardless of Ethereum network conditions.

Requesting withdrawal authorizations A possible attack against the collateral manager could be as follows:

1. Create a new application on the Flexa network
2. Supply Amp as collateral backing the new app
3. Complete payments without the intention to settle
4. Withdraw the collateral before Flexa can consume it

As a result, staked collateral is not automatically authorized for withdrawal, and must be requested. Prior to releasing collateral and appending a withdrawal authorization Merkle tree, all payments backed by the collateral must be settled (or if not settled, then the Amp must be consumed). After this process is completed, the requested or maximum allowed withdrawal (whichever is less) is authorized on chain, and is considered *released*. Stakers request a withdrawal by calling the `requestRelease` function on the Flexa collateral manager. Once the collateral is released, Flexa provides the amount authorized, nonces, and Merkle tree proof; on-chain release requests ensure that account authentication cannot be compromised (*e.g.,* via replay or man-in-the-middle attacks). The response from Flexa is safe to broadcast publicly, as the collateral manager only allows withdrawal transfers to be initiated by the original staker or approved operators of the collateral manager contract (*i.e.,* operators are immutable for withdrawal authorizations).

Withdrawal automation The release request function and event contain an open `data` field that can be used to incorporate destination information. This metadata can be used by Flexa to autonomously complete withdrawal transfers. The specification for this field is developed as open source, allowing for custom implementations for particular suppliers (*e.g.,* a custody provider that balances supplied collateral determinant on user preferences). Irrespective of the contained data, it will be based on the ABI-encoding scheme used for the `transferByPartition` method operator data.

4.4.3 Fallback withdrawals

Given the permissionless platform that decentralized finance provides, withdrawal functions were designed to ensure autonomous operation in the unlikely scenario that Flexa ceases to publish withdrawal roots for any reason (*i.e.,* tokens will always remain recoverable from collateral managers). The contracts include a time-locked *fallback withdrawal* mechanism that allows users to recover funds after a period of inactivity. The fallback mechanism achieves this by proactively tracking the number of unreleased tokens available for withdrawal on chain (inclusive of balance updates due to collateral consumption and earned rewards), as well as which supply receipts have been unrecognized and are therefore reversible.

Supply receipt tracking The `tokensReceived` hook from Amp is designed to not only seek authorization for the receipt of tokens from collateral managers, but also to store metadata for received transfers. This is leveraged in the Flexa collateral manager by storing a record of every received transfer along with a nonce. For example, the first three deposits might be:

Table 7: Collateral manager supply receipts by partition and amount.

Nonce	Address	Partition	Amount
1	0x3096...4abf	0xcccc...5f5a	500
2	0x1cd0...1f00	0xcccc...5f5a	10000
3	0x7076...401e	0xcccc...5f5a	500000

The same data is also emitted in an event that is observable off-chain, which is useful for tagging aggregate data published asynchronously to the contract.

Authorizing fallback withdrawals To authorize the retrieval of the balance of all accounts and partitions in the event that Flexa discontinues operating the collateral manager contract, the full set of balances is preemptively and routinely published. A fallback Merkle tree is generated (separately from the standard withdrawal authorization Merkle trees) and published to the contract. Each leaf in the tree contains the following data:

```
Address: 0x6c41...5b9e
Partition: 0xcccc...f418
Maximum cumulative withdrawal amount: 400 wei
```

The maximum cumulative withdrawal amount is the sum of all previous withdrawal amounts, currently authorized withdrawal amounts, and the current number of unreleased tokens for the address and partition. Note this value increases geometrically, but given Amp total supply of

100,000,000,000 tokens, approximately 10^{48} withdrawals of total supply are required prior to uint256 value overflow. This data is generated for all addresses and partitions, and subsequently hashed to generate the fallback withdrawal authorization Merkle tree. The root of the fallback tree is published, along with the latest observed supply receipt nonce to reset the configured fallback time-lock period. Unlike the standard withdrawal process, this does not require a ledger of authorized trees; only the most recent values are stored on-chain.

Executing fallback withdrawals After the configured time-lock period has expired, stakers can withdraw their full balances using the Amp `transferByPartition` function. Similar to standard withdrawals, the authorization data validated against the current fallback authorization Merkle tree is ABI-encoded and sent in the `operatorData` parameter:

- Withdrawal type (always 32-byte `0xbbbb...bbbb` to signal that the transfer is a fallback withdrawal)
- Supplier
- Maximum cumulative withdrawal amount
- Merkle tree proof

As an example, if the values were:

- Withdrawal type (32-byte `0xbbbb...bbbb`)
- Supplier (20-byte address `0xa34d6a2d9c7db5e3cdb2a533be9c1786e455ac07`)
- Maximum cumulative withdrawal amount (uint256 `200`)
- Merkle tree proof (bytes32[] with a single element of `0x68610245b27875a95c607cac04283c2d1f121786e8936d81061c5cb085372bcc`)

Then the `operatorData` would be a 192-byte array containing the following:

```
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
000000000000000000000000a34d6a2d9c7db5e3cdb2a533be9c1786e455ac07
00000000000000000000000000000000000000000000000000000000000000c8
0000000000000000000000000000000000000000000000000000000000000080
0000000000000000000000000000000000000000000000000000000000000001
68610245b27875a95c607cac04283c2d1f121786e8936d81061c5cb085372bcc
```

The Amp contract passes the operator data (including the source, destination, and number of tokens) to the collateral manager through the standard token transfer hooks, and its validation against the current Merkle root is performed on chain.

Reversing supply receipts All stake receipts with nonce value less than or equal to the record highest deposit nonce are accounted for within the fallback withdrawal data, while supply receipts with higher nonces are not. It is possible that transfers are made to the collateral manager contract after the last fallback withdrawal authorization root is published; those stakes should also be unlocked. To make these tokens available in the fallback scenario, there is an additional withdrawal type that can be executed by stakers to reverse transfers via the standard Amp `transferByPartition` function with the following `operatorData`:

- Withdrawal type (always a 32-byte `0xcccc...cccc` to signal that the transfer is a supply refund)
- Supply receipt nonce

As an example, if the values were:

- Withdrawal type (32-byte `0xcccc...cccc`)
- Supply receipt nonce (uint256 `3`)

Then, the `operatorData` would be a 64-byte array containing the following:

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
0000000000000000000000000000000000000000000000000000000000000003
```

The entire fallback withdrawal mechanism is natively integrated with the hooks provided by the Amp token. Although there is no protocol enforcement for the existence of fail-safe retrievals from collateral managers, it is a recommended approach to minimize collateral management contract risk.

4.5 Further extensibility

The Amp token smart contract has been designed with consideration of existing network limitations, and is compatible with scaling enhancements such as zero-knowledge proof (ZKP) systems, optimistic rollups, and Ethereum 2.0. It is expected that at scale, collateral managers will have access to secure micro-collateralization transactions on-chain. Additionally, future Amp rewards distributions can be made using verifiable and trustless ZKPs to decrease transaction costs as well as preserve network data privacy. Token partitions can also enable the issuance of new types of collateral assets for simultaneous use across platforms. Collateral managers can mint bearer tokens based on any staked collateral; for instance, allowing non-custodial transfers of proxy yield tokens derived from Amp. This method is also practical for DLT-based networks to facilitate cross-chain token minting for rapid collateral deployment.

5 Token economics

Amp has been designed as a low-volatility collateral token that continuously appreciates in value as a direct result of its utility. It mirrors a shift toward open token networks wherein users create and derive all intrinsic platform value through endogenous economic incentives. Since Amp is backed only by its literal use and not extrinsic assets, it is critical to model its economic foundation. Ultimately, this model is more cost efficient and productive than existing business models. Amp employs simple and transparent financial primitives (*e.g.*, fixed supply, rudimentary staking mechanics), and avoids complicated synthetic or derivative instruments, rebasing mechanisms, multi-asset algorithmic models, and artificial constraints that are overly complex to users. Instead, Amp focuses on providing high-quality collateral, stability, and self-sustaining characteristics to create exponentially more utility.

Flexa network collateral Amp is the fundamental collateral token used to secure retail payments within the Flexa network, wherein wallets/applications are staked with Amp to enable spending capacity. For each successful transaction, merchants are charged a small percentage-based fee (for comparison, less than the prevailing interchange rate). This remuneration is the only acceptance expense for the finality-as-a-service and elimination of fraud that Flexa provides. These proceeds are then used to open-market purchase Amp tokens for autonomous distribution to collateral contracts. Network participants directly receive these non-inflationary network rewards *pro rata* based on the quantity of Amp staked. This self-reinforcing cadence of increasing payment utility (spending throughput), collateral requirements, and compounding rewards is the framework to ensure all Flexa network value is captured within Amp tokens.

Amp is the first project to enable participants to stake collateral while maintaining custody of the underlying assets. This novel implementation of partition schemes and modular collateral managers (§4.3) allows for dramatically simplified user interactions to facilitate mass adoption and decentralization of the Flexa network. Participants can stake Amp to provide universal, permissionless spending utility while also earning the entirety of network proceeds. This virtuous cycle is fundamental to innovative token networks that are required to challenge and circumvent the existing system (§1.0). Various models are explored to determine the relationship between payment utility and expected aggregate network value. The Amp cryptoeconomic model is explicit, but corresponding token value (*i.e.*, the utility of the collateral itself) projections involve interpreting system dynamics such as platform growth, staking composition, and velocity/stability rates. Amp collateral value is analyzed using an amalgamation of existing models for economic growth, capital asset pricing, continuous market buy-pressure, and discounted cash flow.

5.1 Production model

Flexa is assumed to be a genericized economy with conventional elements of input (capital, labor) and output. Traditionally, this can be modeled using a production function to determine how various consumption factors contribute to resultant utility. A straightforward approach is the use of the Cobb-Douglas production function (CDPF) [60] to analyze a production process. The CDPF represents the technological relationship between production output as a function of economic inputs as:

$$Y = AL^{\beta}K^{\alpha} \tag{1}$$

In this relationship, Y is the total production (*i.e.*, value of goods produced), and L, K are the labor inputs and capital inputs, respectively. A is the total-factor productivity (*i.e.*, the ratio of aggregate output to aggregate input) while constants α and β are the output elasticities of capital and labor, respectively. While not directly applicable to the Flexa network and its parameters, this is a reasonable introductory foundation. Accordingly, the AK model (Equation 2) of economic growth [60] (a specialized case of CDPF) is more amenable as it incorporates high endogenous development [62] due to technological change and the absence of diminishing returns to capital. This allows for the microeconomics of network growth to be a viable proxy for the macroeconomics of economy gross production. For instance, allocative efficiency can be represented as Amp staking propensity, along with Y as the cumulative output of more efficient spending utility. The AK model also integrates the time series of output, encompassing the long-term growth of technology innovation, network participants, and capital investment.

$$Y = AK^{\alpha}L^{(1-\alpha)} \tag{2}$$

For the Amp-specific scenario, define A as technology efficiency: the ratio of human capital invested to total network value. Increases in A result in growth from both sides of the network (*i.e.*, wallets and merchants), compounding transaction volume. As payments become more efficient, technological innovation ultimately preserves value by protecting network effects produced by Amp economic incentives. An economy is fundamentally based on transactions in a market; Amp enables spending which drives the microeconomy.

5.2 Tokenization model

Production functions provide an overview of the network economy for estimates of total platform value. However, spending/rewards utility is achieved at the individual level; analysis of asset pricing specific to token networks is required. Amp generates the entirety of value within the

network, and a novel valuation approach is considered, as tokenization creates alignment between technology, labor, *and* capital within the ecosystem.

Symmetric incentives A distributed token network allows for powerful, mutual incentive structures among its users. Amp token economics align the incentives of each network participant (*i.e.,* the literal owners of the protocol), eliminating the principal-agent problem. In this sense, Flexa maximizes user value *and* protocol value; these two concepts are equivalent. Amp is designed to incentivize and reward *value creation*, generating efficient payments that are the essence of productivity growth. [43] A token network avoids *value extraction*, the systematic process of appropriating portions of platform value through economic rent-seeking. Amp represents a more circumscribed value creation and value capture structure than traditional equities.

Non–zero-sum participation Proof-of-stake block rewards are generated by minting new tokens as staking incentives. Due to capital infrastructure costs and minimum ownership requirements, validators exclusively receive these rewards, maintaining or increasing their percentage ownership of the network. Correspondingly, individuals are subjected to universal dilution rates; non-stakers inherently lose financially and have no economic incentive to hold tokens. The only counter to this zero-sum dilemma is to delegate/bond tokens and be subject to fees, extensive lock-up periods, and surrendering custody. Validator networks also have a propensity to supplement the wealth of only the largest holders, where only participants with significant investment will reap staking rewards.

As a fixed-supply contract token, Flexa-generated Amp rewards are non-inflationary and distinctly *non*–zero-sum. No taxation, subsidies, or seigniorage is required; the virtuous Amp re-purchase and distribution model does not reward participants as a corresponding loss for others. Additionally, open market re-purchases are economically superior to burning tokens (*i.e.,* permanently removing tokens from circulation) since this action has no direct contribution to the productive capability of the network. Token burning ultimately does not incentivize behaviors that create network utility (or demand for network utility) and is primarily a signaling technique, since the tokens are non-transferable, but still technically exist. By contrast, Flexa implements an efficient smart contract staking/reward process, focused on democratically increasing direct utility for the network.

5.2.1 Asset pricing

A fundamentals-based asset pricing model for tokenized networks is considered, where the value of Amp is explicitly anchored to its spending utility. Tokens can be modeled as a numeraire commodity, providing platform-specific convenience yield when staked. Necessarily, platform

adoption and growth in the network economy proportionally stimulate these utility incentives. The recent model proposed by Cong, Li, and Wang (CLW) [16] incorporates these endogenous network effects into an otherwise canonical dividend growth formula. Incorporating elements of a multi-projection consumption capital asset pricing model (CCAPM) and consumer utility flow, a model is developed using platform productivity as a state variable. For a generic token network, the asset price P_t can be expressed as follows:

$$P_t = \frac{N(A_t)S(A_t)A_t}{M}\left(\frac{1-\alpha}{r-\mu_p(A_t)}\right)^{\frac{1}{\alpha}} \tag{3}$$

where the aggregate transaction need $S(A_t)$ is defined by:

$$S(A_t) = \int_{u(A_t)}^{U} e^u \, dG(u) \tag{4}$$

In these identities A_t is platform productivity, N is the platform user base, and M is the total fixed token supply. Further, μ_p is the expected price appreciation (a univariate function of A_t), and $\alpha \in (0,1)$ is an autoregressive constant. The user type u has distribution $G(u)$ (*e.g.,* normal distribution of an adoption curve) with transaction need e^u, a log-normal distribution which drives token demand.

The general CLW model can be used to estimate a similar relationship for Amp token dynamics. In this scenario, tokens are *not* transactional, but instead create spending utility (proportional to A_t). For simplicity, user types are standardized so $\int dG(u) = 1$, implying $S(A_t)$ is constant, an intuitive observation since aggregate transaction need can be defined as required utility. Further, A_t can be refined to represent the unconditional payment volume of the Flexa network at time t. This yields a derivative model inclusive of dynamic adoption where utility flow is derived by staking Amp as follows:

$$P_t = \frac{N_t U_t^P A_t}{M_t}\left(\frac{1-\alpha}{r-\mu_t^P}\right)^{\frac{1}{\alpha}} \tag{5}$$

Amp token pricing is based on user demand for staking yield, spending utility, and, expectation of future productivity growth. In this model, A_t is the total network payment volume, $U_t^P$is the aggregate collateral utility (staked value), with M_t equal to the circulating token supply with fixed maximum. The effective carry cost $r - \mu_t^P$ is due to expected returns at time t (μ_t^P) with a risk neutral return rate r (*i.e.,* the hurdle rate).

This modified Flexa network pricing model accounts for total users, representing their willingness to pay for the utility flow of the platform. Accordingly, as capacity is created there is net capital increase to the network: the systematic design of collateralization. Staking Amp increases overall utility, rising proportionally as more transactions and users (*e.g.,* consumers, wallets, merchants) join the network. [28] Self-enforcing network rewards in the form of Amp compound adoption, since *participants* perpetually share the economic benefits created. Economic growth of the network comes from adoption and payment volume (*i.e.,* productivity output) as a result of new technology innovations that provide simplified access across new wallets and asset types (§2.1). [59] Ostensibly, as token price increases, adoption (*i.e.,* staking) increases, and the Amp staking cycle becomes systematic and more correlated to consumption (§5.3).

5.2.2 Continuous-time liquidity

An important model to consider is the effect of continuous buy pressure on the token due to platform use. As the network completes payments, a percentage of each intra-network transaction takes liquidity for open-market Amp token purchases. These rewards are autonomously distributed and capitalized on a regular basis, a process that can be replicated with a time-series model involving time-varying liquidity demand from Amp rewards. Since this mechanism has two core variables, *stock* (aggregate market order book) and *flow* (purchase volume) a multivariate intensity model is used to allow representation of intertemporal point processes. [6][21][22] The buy pressure can be measured using an autoregressive model which specifies the conditional intensity as follows:

$$\lambda(t_i, \mathcal{F}_t) = X_t \lambda_0(t) \tag{6}$$

$\mathcal{F}_t$ is conditional intensity of the counting process $\{N(t), t \geq 0\}$, with autoregressive component X_t and baseline intensity function $\lambda_0(t)$. X_t can be defined as a generic autoregressive model:

$$X_t = \phi_1 X_{t-1} + \cdots \phi_p X_{t-p} + \varepsilon_t \tag{7}$$

where μ_t is the process mean with stochastic term ϕ and noise function $\varepsilon_t = X_t - \mu_t$. By definition, the terms $X_{t-1}, X_{t-2}, \cdots X_{t-p}$ are lagging values. Since past values of μ_t (purchase volume) impact future activity *a priori*, the standard auto regressive moving average (ARMA) model of order (p, q) can be used for purchase data:

$$X_t = \phi_1 X_{t-1} + \cdots \phi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \tag{8}$$

where (p, q) is the order of the autoregressive polynomial X_{t-p}, and moving average polynomial $\theta_q \varepsilon_{t-q}$, respectively, with standard moving average model parameter θ. The ARMA model realizes the resultant purchase intensity based on the conditional mean of the process (*i.e.,* the time dependent market purchases based on Flexa payment volume). [56] Evaluating continuous purchases at points in time would follow a positively autocorrelated process, a direct indication of increased asset value. [42] Intuitively, buy pressure $X_t \lambda_0(t)$ will be a significant determinant of volatility (§5.4) and returns. This implies that each transaction (as part of a time-intensity series) has significant predictive power for both the conditional mean and the conditional volatility of asset value.

5.3 Network efficiency model

The marginal value of network productivity is positive (Equation 5), inferring that equilibrium dynamics of user adoption and token valuation will be achieved. At maturity, the complete network model is best represented by direct consumption. Since the Flexa collateral manager is public and permissionless (with no staking restrictions), large collateral positions (*i.e.,* staked wallets) will assuredly develop market efficiency with yields approaching parity to r. [9] The utility demand will determine Amp collateral requirements, with corresponding yield valued relative to network payment volume.

When a technology matures, its value more closely resembles its terminal discounted cash flow (DCF) value. [48] Applying previous consumption-based asset pricing models, the simple net return of an asset η at date t, is defined as:

$$\eta_t = \frac{P_t + \rho_t}{P_{t-1}} - 1 \tag{9}$$

where ρ_t is the reward distribution at date t. The natural logarithm of the *gross return* R_t is a continuously compounded return $\ln(R_t) = \log(1 + \eta_t)$, since Amp rewards distributed from the Flexa collateral manager are capitalized in real time with R_{t+1} as the gross return on an asset from t to $t + 1$. Consumption-based asset pricing models normalize expected discounted returns to 1, providing the expectation: $E_t(M_{t+1} R_{t+1}) = 1$. The pricing kernel M_{t+1} is defined as:

$$M_{t+1} = \xi_t \frac{\tau(C_{t+1})}{\tau(C_t)} \tag{10}$$

Where the utility flow τ is a function with respect to the level of consumption C. By definition, M_{t+1} is equal to the intertemporal marginal rate of substitution (MRS) in consumption. ξ_t is

defined as the *network efficiency ratio* given by $\prod_{i=1}^{n} F_{i,t}$, the product of n efficiency rates (*e.g.* wallet and contract collateralization ratios). The return on one-period, risk-free rate $R_{f,t+1}$ is substituted in Equation 10 with expectation E_t as: $1 + R_{f,t+1} = 1/E_t\left(M_{t+1}\right)$. Rearranging and defining $\nu_t = (1/\xi_t)$ provides:

$$E_t(M_{t+1}) = \frac{\nu_t}{(1 + R_{f,t+1})} \tag{11}$$

Since the quantity of Amp staked is predominantly a function of network volume, it is relevant to further explore the i elements of efficiency rates F_i and their equilibria. At scale, there are four fundamental parameters in determining overall efficiency (collateralization) rate:

- Manager-specific, based on liquidity
- Wallet-specific, based on spending utility variance and expected growth
- Network-wide, based on fragmentation and disproportionate allocation
- Network-wide, based on staking and dormancy rates (inclusive of market liquidity)

The product of these rates, the *network efficiency ratio*, provides an estimate of the hypothetical aggregate collateral available (*i.e.,* the total value of Amp supply). This demonstrates that given time t, the ratio of collateral value (predominant factor ν_t) to maximum transaction volume is of order 10^4. Ultimately, this relationship requires that the network captures considerably more *utility* value than the sum of payment transactions alone. Notionally, the network is significantly capital inefficient, but reduces net costs exponentially.

5.4 Stability analysis

Due to the dynamics of staking collateral, the time-dependent utility efficiency decreases as network adoption increases. This inverse relationship yields an *over-capacity* condition with respect to total protocol value (*i.e.,* aggregate value of circulating Amp tokens). However, this results in multiple reinforcing mechanisms and is an optimal state of the platform. The primary function of Amp is to decentralize payment risk; axiomatically the more inefficient the network is the more distributed the collateral will be. Given a more heterogeneous staking gradient, collateral quality (§3.2.2) increases due to greater stability against price shocks.

Accordingly, Amp provides sustained stability through its economic model and supply inelasticity (fixed issuance). Since collateral needs to be staked indefinitely to provide perpetual spending utility, Amp tokens are likely to have significant holding duration. Using the monetary equation of exchange ($MV = PQ$), [63] this implies that tokens with low *velocity* (*i.e.,* staked tokens) will

be more valuable than other comparable assets. It is also noted that utility is maximized further due to Amp partition functionality; tokens can seamlessly and safely be used as auxiliary collateral.

All of these factors actively contribute to minimize collateral value deviations during dynamic periods. In this context, to model the volatility of Amp due to a market variance, a general autoregressive conditional heteroscedasticity (GARCH) model of order (p, q) can be used. [19] [47] Assuming a time series $t \in \{0,1,\cdots\}$, the period t return on a financial asset (Equation 9) can be expressed as:

$$\eta_t = \mu_t + \varepsilon_t \tag{12}$$

where $\mu_t = E_t(\eta_t)$, $\varepsilon_t = \sqrt{h_t} s_t$, subjected to random shocks with $E_t(s_t) = 0$ and $Var_t(s_t) = 1$. The linear function of the conditional variance h_t can be expressed as:

$$h_t = \omega + \alpha_1 \varepsilon_{t-1}^2 + \cdots + \alpha_q \varepsilon_{t-q}^2 + \beta_1 h_{t-1} + \cdots + \beta_p h_{t-p} \tag{13}$$

using $(p, q) = (1,1)$,

$$h_t = \omega + \beta h_{t-1} + \alpha \varepsilon_{t-1}^2 \tag{14}$$

with positive constants ω, β, and α. Initializing the volatility shock with a stochastic variable such that h_t is the conditional variance of η_t, the relationship becomes:

$$h_t = \omega + (\alpha + \beta) h_{t-1} + \alpha(\varepsilon_{t-1}^2 - h_{t-1}) \tag{15}$$

The conditional variance of the asset return h_t is the sum of a constant autoregressive term $(\alpha + \beta)h_{t-1}$ and recursive shock term $\alpha(\varepsilon_{t-1}^2 - h_{t-1})$. This demonstrates that h_t is of the same order of Equation 8. Therefore, network payment volume (regardless of market conditions) has the potential to yield buy pressure of similar magnitude to stabilize shocks. Intuitively this is reasonable, since active utility requirements and expectation of future token yield (aligned with longer duration hold time) stabilizes token value in the presence of platform productivity shocks. Correspondingly, this implies the majority of a liquid market consists of tokens re-circulated from yield, minimizing equilibrium disturbances. Ultimately, economic models demonstrate that a virtuous cycle of price support can be guaranteed by the use of the asset itself, a hallmark of decentralized token networks.

6 Summary

Amp

- ERC20-compatible, fixed-supply, immutable token (no admin privileges)
- Designed as low-volatility collateral with compounding value, backed only by its utility
- Implements token partition framework to allow restricted access (staking) without transfer
- Can resolve practicality issues by *retaining* staked assets at an owner address
- Open-source collateral manager contracts are extensible to any project
- Robust withdrawal and fallback system to ensure security and custody of staked assets
- Amp is the exclusive collateral token of the Flexa network

Flexa network

- Existing payment networks are vulnerable to fraud, data breaches, and structural cost
- Digital assets/currencies are quickly becoming the predominant form of global spending
- Native DLT payments are not tenable at scale due to finality assurance complexity
- Flexa is a merchant network that accepts low-cost digital payments without fraud
- Payments are guaranteed in real-time using Amp as collateral, decentralizing risk
- Amp is permissionlessly staked for wallet applications to enable consumer spending
- Proceeds from merchant fees are used to autonomously open-market purchase Amp tokens
- Stakers receive all Amp re-purchased, retaining aggregate network value
- Virtuous cycle of staking and rewards distribution supports collateral integrity
- Production models for economic growth can serve as a proxy for estimating network utility
- Token pricing models predict positive correlation of payment volume and utility
- Intertemporal market buy-pressure models demonstrate resilience to market shocks
- Maturity models predict broad, inefficient collateral distribution is utility maximizing

7 References

[1] S. Athey, I. Parashkevov, V. Sarukkai, J. Xia. Bitcoin Pricing, Adoption, and Usage: Theory and Evidence. *Working Paper, Stanford Graduate Business School*, 2016.

[2] L. Baird. Hashgraph Consensus: Fair, Fast, Byzantine Fault Tolerance. 2016.

[3] M. Bergman, G. Guibourg, B. Segendorf. The Costs of Paying: Private and Social Costs of Cash and Card Payments. *Sveriges Riksbank Working Paper* 212, 2007.

[4] G. Bertoni, J. Daemen, M. Peeters, G. van Assche. The Keccak SHA-3 Submission. https://keccak.team/files/Keccak-submission-3.pdf.

[5] F. Black, M. Scholes. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy* 81, 637-654, 1973.

[6] C. G. Bowsher. Modeling Security Markets in Continuous Time: Intensity Based, Multivariate Point Process Models. *Nuffield College Discussion Paper* W22, 2002.

[7] M. K. Brunnermeier, Y. Sannikov. A Macroeconomic Model with a Financial Sector. *American Economic Review* 104, 379–421, 2014.

[8] E. Buchman. Tendermint: Byzantine Fault Tolerance in the Age of Blockchains. 2016.

[9] J. Y. Campbell. Financial Decisions and Markets: A Course in Asset Pricing. *Princeton University Press*, 2017.

[10] N. Carter. It's the Settlement Assurances, Stupid. https://medium.com/@nic___carter/its-the-settlement-assurances-stupid-5dcd1c3f4e41.

[11] C. Catalini, J. S. Gans. Initial Coin Offerings and the Value of Crypto Tokens. *Discussion Paper, National Bureau of Economic Research,* 2018.

[12] K. Chan, W. M. Fong. Trade Size, Order Imbalance, and the Volatility-Volume Relation. *Journal of Financial Economics*, 57, 247–273, 2000.

[13] L. Chen, L. Cong, Y. Xiao. A Brief Introduction to Blockchain Economics. *Working Paper*, 2019.

[14] J. Chiu, T. Koeppl. The Economics of Cryptocurrencies: Bitcoin and Beyond. *Working Paper*, 2017.

[15] L. Clausen, C. Heymann. Bankless: The Token Maximalist Thesis. https://bankless.substack.com/p/the-token-maximalist-thesis.

[16] L. W. Cong, Y. Li, N. Wang. Tokenomics: Dynamic Adoption and Valuation. *Fisher College of Business Working Paper Series,* 2020.

[17] C. Decker, R. Wattenhofer. A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels. 2015.

[18] E. Deirmentzoglou, G. Papakyriakopoulos, C. Patsakis. A Survey on Long-Range Attacks for Proof of Stake Protocols. 2019.

[19] J. Duan. The GARCH Option Pricing Model. *Mathematical Finance* 5, 13-32. 1995.

[20] EMVCo. Annual Report, https://www.emvco.com/wp-content/uploads/documents/EMVCo-Annual-Report-2019-Updated.pdf.

[21] R. F. Engle, G. M. Gallo. A Multiple Indicators Model for Volatility Using Intra-Daily Data. *Journal of Econometrics* 131, 3-27, 2006.

[22] R. F. Engle, V. Ng. Measuring and Testing the Impact of News on Volatility. *Journal of Finance* 48, 1749-1778, 1993.

[23] Ethereum Improvement Proposal 20: ERC20 Token Standard. https://eips.ethereum.org/EIPS/eip-20.

[24] Ethereum Improvement Proposal 777: ERC777 Token Standard. https://eips.ethereum.org/EIPS/eip-777.

[25] Ethereum Improvement Proposal 1820: Pseudo-introspection Registry Contract. https://eips.ethereum.org/EIPS/eip-1820.

[26] European Union. Regulation (EU) 2016/679 of the European Parliament. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679.

[27] G. Fanti, L. Kogan, P. Viswanath. Economics of Proof-of-Stake Payment Systems. *Working Paper*, 2019.

[28] R. C. Feenstra. Functional Equivalence Between Liquidity Costs and the Utility of Money. *Journal of Monetary Economics* 17, 271–91, 1986.

[29] K. French, G. W. Schwert, R. F. Stambaugh. Expected Stock Returns and Volatility. *Journal of Financial Economics* 19, 3–30, 1987.

[30] N. Gandal, H. Halaburda. Can We Predict the Winner in a Market with Network Effects? Competition in Cryptocurrency Market. *Games* 7, 16, 2016.

[31] D. Garcia, C. J. Tessone, P. Mavrodiev, N. Perony. The Digital Traces of Bubbles: Feedback Cycles Between Socio-Economic Signals in the Bitcoin Economy. *Journal of the Royal Society Interface* 11(99), 2014.

[32] D. Garcia-Swartz, R. Hahn, A. Layne-Farrar. The Move Toward a Cashless Society: Calculating the Costs and Benefits. *Review of Network Economics* 5(2), 198-228, 2006.

[33] L. M. Goodman. Tezos: A Self-Amending Crypto-Ledger. *Position Paper*, 2014.

[34] F. Hayashi, W. Keeton. Measuring the Costs of Retail Payment Methods. *Economic Review*, 2012.

[35] J. D. Hamilton. Time Series Analysis. *Princeton University Press*, 1994.

[36] A. S. Hayes. Cryptocurrency Value Formation: An Empirical Study Leading to a Cost of Production Model for Valuing Bitcoin. *Telematics and Informatics* 34(7), 1308–1321, 2017.

[37] S. Heston, S. Nandi. A Closed-form GARCH Option Pricing Model. *Review of Financial Studies* 13, 585-625, 2000.

[38] F. J. Hinzen, J. Kose, F. Saleh. Proof-of-Work's Limited Adoption Problem. *NYU Stern School of Business*, 2019.

[39] B. Hollifield, Miller, R. Miller, P. Sandas, J. Slive. Liquidity Supply and Demand in Limit Order Markets. *Centre for Economic Policy Research*, 2002.

[40] International Organization for Standardization: ISO/IEC 7816. https://www.iso.org/obp/ui/#iso:std:iso-iec:7816:-8:ed-4:v1:en.

[41] R. Kan, G. Zhou. A Critique of the Stochastic Discount Factor Methodology, *Journal of Finance* 54, 1221-1248, 1999.

[42] A. Koch, S. Ruenzi, L. Starks. Commonality in Liquidity: A Demand-Side Explanation. *The Review of Financial Studies* 29(8), 1943–1974, 2016.

[43] W. Lazonick, J.S. Shin. Predatory Value Extraction: How the Looting of the Business Enterprise Became the US Norm and How Sustainable Prosperity Can Be Restored. *OUP Oxford*, 2019.

[44] Y. Liu, A. Tsyvinski. Risks and Returns of Cryptocurrency. *Working Paper, National Bureau of Economic Research*, 2018.

[45] D. Mazieres. The Stellar Consensus Protocol: A Federated Model for Internet-Level Consensus. 2015.

[46] S. Nagel. The Liquidity Premium of Near-Money Assets. *The Quarterly Journal of Economics* 131, 1927–1971, 2016.

[47] D. B. Nelson, Stationarity and Persistence in the GARCH (1,1) Model. *Econometric Theory* 6, 318-334, 1990.

[48] E. Pagnotta, A. Buraschi. An Equilibrium Valuation of Bitcoin and Decentralized Network Assets. *Working Paper, Imperial College*, 2018.

[49] Payment Card Industry Standards Council. Data Security Standard: Requirements and Security Assessment and Procedures v3.2.1. 2018.

[50] J. Poon, T. Dryja. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. 2016.

[51] M. Raskin, D. Yermack. Digital Currencies, Decentralized Ledgers, and the Future of Central Banking. *National Bureau of Economic Research*, 2016.

[52] H. Schmiedel, G. Kostova, W. Ruttenberg. The Social and Private Costs of Retail Payment Instruments: A European Perspective. *European Central Bank Occasional Paper Series* 137, 2012.

[53] D. Schwartz, N. Youngs, A. Britto. The Ripple Protocol Consensus Algorithm. 2014.

[54] M. Sockin, W. Xiong. A Model of Cryptocurrencies. *Working paper*, 2018.

[55] Solidity. Contract Application Binary Interface (ABI) Specification. https://solidity.readthedocs.io/en/v0.6.12/abi-spec.html.

[56] E. Stein, J. Stein, Stock Price Distribution with Stochastic Volatility: An Analytic Approach. *Review of Financial Studies* 4, 727-753, 1991.

[57] R. L. Tweedie. Drift Conditions and Invariant Measures for Markov Chains. *Stochastic Processes and their Applications* 92, 345-354, 2001.

[58] J. von Neumann, O. Morgenstern. Theory of Games and Economic Behavior. *Princeton University Press*, 1944.

[59] E. G. Weyl. A Price Theory of Multi-Sided Platforms. *American Economic Review* 100, 1642-72, 2010.

[60] Wikipedia. AK Model of Industrial Growth. https://en.wikipedia.org/wiki/AK_model.

[61] Wikipedia. Cobb-Douglas Production Function. https://en.wikipedia.org/wiki/Cobb%E2%80%93Douglas_production_function.

[62] Wikipedia. Endogenous Growth Theory https://en.wikipedia.org/wiki/Endogenous_growth_theory.

[63] Wikipedia. Equation of Exchange. https://en.wikipedia.org/wiki/Equation_of_exchange.

[64] Wikipedia. Merchant Customer Exchange. https://en.wikipedia.org/wiki/Merchant_Customer_Exchange.