Intro

This document explains the common user experience issues of Ethereum/EVM wallets, and how smart wallets and in particular APY Wallet solve most of them.

It requires some level of prior knowledge of the Ethereum ecosystem and cryptocurrencies.

Throughout this document, when we say Ethereum, we mean the broad EVM ecosystem including all EVM chains like Polygon, Arbitrum, Optimism, Avalanche, Fantom, BSC, etc

Ethereum wallets have a tough problem to solve

The Ethereum and EVM ecosystem has always been particularly challenging to build a wallet for, due to a few underlying characteristics that ultimately lead to user inconveniences, due to leaky abstractions (complexity "leaking" to the UX layer):

How transaction fees work: gas price and gas limit

Transaction fees have always been particularly unintuitive for users: for example, you may be inclined to think about increasing the gas limit in the hopes of getting a transaction mined faster, when in fact achieving the opposite.

Fee parameters are even more complicated since EIP1559, where the gas price is split in base fee and tip. To make matters more complicated, you need to broadcast the transaction with a 'realistic' base fee, but the tip is the main parameter that matters when competing with other transactions.

Historically wallets have opted for exposing the underlying complexities without an attempt to educate users or visually distinct values that should be modified (gas price) vs values that should be automatically calculated (e.g. gas limit).

Finally, gas must be paid in ETH, which is particularly inconvenient for someone who's just starting out their crypto journey and doesn't hold ETH yet, adding an extra step to their journey.

Nonce management and stuck transactions

Transactions in Ethereum are sequential, meaning that you can't mine a subsequent transaction if there's a one pending before it.

However, most wallets are not doing a good job of conveying this, leading users to believe that they can 'speed up' a transaction by increasing its fee, despite it being blocked by a prior transaction.

Security implications of arbitrary code execution

In Ethereum, every transaction can trigger arbitrary code execution, making it potentially dangerous. It's very difficult to visualize the full consequences of a transaction in a wallet, but existing wallets haven't even tried: normally what wallets visualize is just the "to" (interacting with) address, value (ETH sent) and hex data, which is misleading, as multiple value transfers may be happening in this transaction, and obviously not human friendly, as most users can't read the hex data.

This poor visualization has lead to other issues, such as users sending tokens to smart contract addresses because the "to" field of existing transactions that send tokens is not actually the user they sent the tokens to, but the smart contract address, leading them to copy the wrong destination address.

ERC20 approvals UX and security (as a result of arbitrary code execution)

There's a lot to say about ERC20 approvals. Approvals are a function of ERC20 (and similar) tokens where a permission is granted to an address to spend a specific (or infinite) amount of your tokens. This may sound strange, but it's necessary for smart contracts: tokens are smart contracts themselves (just arbitrary pieces of code) - as such, if you just send a specific amount of tokens to a DEX address, it would be lost, because the DEX contract will not know it received the tokens because it's code won't execute. The solution is to call the DEX contract itself and it will "pull" the tokens from you.

Approvals are alright from an engineering perspective as they allow contracts to "pull" tokens without reentrancy risks, but they're a classic example of "leaky abstractions": a set of decisions that makes sense on a low level leaks to the UX layer and creates multiple user difficulties:

Key management and opsec

Cryptocurrency private keys (and seed phrases) are notoriously hard to keep safe, especially in the hands of the previously inexperienced.

Why smart wallets solve most UX issues

Smart wallets have been discussed a lot in the past: you may have heard of a similar concept called "account abstractions". Basically the idea is that each Ethereum account will be a smart contract, which helps solve a lot of the forementioned UX issues:

- Batched transactions: one transaction can execute multiple calls or "sub transactions"
- Minimal approvals via batching: rather than having to bother the user with a separate approval procedure or work this around via infinite approvals, we can just batch together the approval with the specific action, making the approval seamless to the user, as well as solving the security problems that arise from infinite approvals, by approving only the specific amount for the context of the transaction
- Fee payments in stablecoins or any ERC20: transaction fees can be paid in any token, eliminating the need for the user to have ETH in their wallet. This is achieved by having a relayer under the hood which actually broadcasts the given transaction on-chain, but it's reimbursed for it's gas costs with a separate ERC20 payment that's actually part of the same transaction. Furthermore, even if the user has a token that the relayer does not accept as fee, they can still transact providing that they have one at the *end* of their transaction batch. For example, if you start with 100 MADEUPCOIN, but the relayer only takes stablecoins or ETH, your first transaction could be a swap of 10 MADEUPCOIN to 1000 USDC and the fee will be taken from that amount.
- Upgradable security. You can start with a hot wallet (software wallet) or even an email/password account (more on that later) and transfer control to a hardware wallet later on without having to move funds or positions.

Simple DeFi interactions aka zaps: transaction batching allows complex DeFi interactions such as depositing into vaults (often a multi-step process), providing liquidity, leveraging Aave exposure, etc. - to be done in one transaction Replacing & cancelling transactions: since transactions are batched, there's no reason for a user to have more than one pending transaction - when a user requests something else to be executed, it just gets added to the pending batch; as such, there's effectively no nonce management we eliminate stuck transactions

Simulating transactions: while this is not particularly a smart wallet trait, smart wallets make it easier to simulate the outcome of transactions, fetching the user balances after it, the logs, and other potential information which can identify unintended consequences of transactions before they get executed

Wallet stability: because of the additional layer of abstraction, actual key management wallets only need to produce signed messages, and won't break or degrade with hard forks (e.g. EIP1559)

Front-running protection: while not specifically a feature of smart wallets, having an account abstraction makes it easier to use alternative transaction broadcasting mechanisms such as Flashbots and Eden. This enables fron-trunning and sandwiching protection.

You may wonder - if smart wallets are so good, what prevented their adoption up until now. There are many factors but the primary ones are 1) technical limitations that have been solved recently (e.g. the introduction of EIP1271) and 2) that most innovators in the space like Argent and Dharma seem to be building mobile-only wallets, and the core crypto community is still mostly looking for web based solutions.

As such, we believe that smart wallets are the future of non-custodial EVM wallets.

Introducing APY Wallet

APY Wallet is the first power user wallet that is launched as a web app and is different from most non-mobile wallets, which are browser extensions. We believe that installing a browser extension is a huge hurdle to both crypto newcomers and crypto-curious users coming from a tech background who may have security concerns related to extensions. An extension will be added later on for the sole purpose of enabling connection to dApps that do not support WalletConnect.

APY Wallet has been built with security in mind, with multiple audits being conducted on both the smart contracts and the UI.

It supports the following features:

- Connect any dApp through WalletConnect
- Automatic transaction fee management
- Paying transaction fees in stablecoins
- Dashboard that automatically displays all your assets: tokens, NFTs and deposits to DeFi protocols

Transaction preview: before signing a transaction, we show a human-friendly description of what it does, step by step
Built-in swaps and cross-chain transfers
Multiple networks: Ethereum, Polygon, BSC, Avalanche,Fantom and more

Some features intended for users who are just starting out:
Sign up with an email/password without compromising the non-custodial nature of APY Wallet(read on to find out how this works)
Deposit FIAT
Gas Tank - prepay for gas fees and save funds

Some power user features:
Transaction batching: ability to do multiple actions in one transaction
Automatic front-running/sandwiching protection via Flashbots
Multiple signers (keys) can be used to control the same account: e.g. a hardware wallet AND a software wallet; those can be easily enabled or disabled

Target audience

At launch (December 1 2021), APY Wallet is focused on the existing crypto audience, namely MetaMask users who are unhappy with their current wallet experience: we believe that most people using Ethereum and other EVM chains are struggling with most wallet solutions on the market, based on our own observations in the space.

However, APY Wallet is also great for newbies: thanks to the FIAT on-ramps and Polygon/BSC/Avalanche support, as well as being web-based and allowing email/password logins, new users are able to sign up, fund their account through a credit card, and start buying tokens on Sushiswap or NFTs on Sudoswap - in mere minutes with no prior.

We believe that to pave the way for the next billion crypto users, we do need features such as seedless login and FIAT on-ramps, but at the same time this adoption isn’t going to happen if we don’t win over the crypto-natives first: our field is heavily recommendation based, and oftentimes new participants ask their more crypto-savy friends for advice.

The relayer and it's role

APY uses a backend service called "the relayer" which is responsible for actually broadcasting your transactions to the network. This is necessary due to the gas abstraction mechanism which 1) allows paying transaction fees in any token, 2) enables the relayer to manage gas fees automatically for you and 3) allows for the pre-payment of gas fees through the Gas Tank functionality, which allows you to credit the relayer in advance with an amount, which will be later used for gas fees.

The relayer is also responsible for keeping one of the two keys in the email/password authentication system (read on to find how that works).

APY can function without the relayer: all you need to do is run from source and set REACT_APP_RELAYER_URL to an empty string, but it loses some functionalities like paying gas fees in tokens, Flashbots, tracking transactions, email/password login, and others.

The APY Wallet Gas Tank

The APY Wallet Gas Tank is a smart wallet feature developed to enable user benefits such as savings and cashback, increasing the overall efficiency of APY Wallet as a crypto asset management tool and reducing significantly the transaction fee costs implicit to smart wallets, compared to standard wallets (EOAs).

When the Gas Tank is used in conjunction with the batching transaction feature (explained above), the resulted combo offers, to the best of our knowledge, optimal transaction fee outputs for the EVM ecosystem.

There are three streams through which the Gas Tank offers value to the users:

Savings from network fees — paid for each transaction relayed/sent to the relayer vs. one-time fee paid for a deposit to the Gas Tank. Acting as a form of prepaid credit for future operations, it helps users pay less for transactions and save on network fees over time.

Savings from bypassing cross-chain transfers — users will no longer need to make cross-chain transfers for gas, as with the Gas Tank enabled, APY Wallet allows them to pay transactions with whatever token is in their Gas Tank balance, regardless of the network;

Cashback — from the difference between the estimated gas fee (signed for at the moment of transaction) and the actual fee needed for the transaction to be completed on the block. This difference can only be calculated post-transaction and varies depending on fee parameters' values at the time of the transaction (e.g. network gas price, smart contract overhead etc.), covering any deviations that might appear and would make the transaction get stuck or fail.

The savings and the cashback are sent to the Gas Tank in the same tokens that were used to pay for their respective transactions. This of course is reflected in the updated balance and creates a sort of self-serving mechanism, a deposit loop, by helping refuel the Gas Tank.

Cashback & top-ups/deposits with the Gas Tank enabled require a longer processing/confirmation time (minutes)— this is because they are automatically updated after each 20 new blocks of transactions are written on-chain.

The Gas Tank also comes with the flexibility benefit of allowing deposits in different tokens, depending on what is supported on each network/chain . This is complemented by the freedom of paying for transactions with any token in the Gas Tank balance, on any network — even if it's not natively supported. For example, users could deposit MATIC on Polygon to the Gas Tank, and then use the MATIC to pay transactions on Ethereum, Moonriver, Arbitrum or Optimism etc.

There is no reverse option for the APY Wallet Gas Tank deposit, as its very function is to allow users to use it when they know they are going to make multiple transactions in the near future, so that they can save on network fees and receive gas fee cashback. This is intended and expected user behavior — similar to a prepaid sim card, once the credit is charged, the user is expected to consume it and if necessary recharge.

The feature was launched with an exclusive NFT promotional drop for the first 10.000 users that sent min. 100 USD equivalent in crypto to their Gas Tank, through one deposit. The promo was active from July 21st to July 31st, 2022 and airdropped NFTs with a special 1.25 multiplier for native WALLET token rewards, lasting for 3 months.

How is the transaction fee calculated when the Gas Tank is enabled?

Firstly, on-chain transaction fee formula is the standard fee formula calculated as the gas limit multiplied with gas price:

TxnFee = effectiveGasLimit * effectiveGasPrice

(there is also a difference between *actual* gas price and *effective* gas price, but this is not the object of our discussion here).

The London Upgrade to the Ethereum protocol introduced EIP1559 with the purpose of making transactions more predictable. Using it, the GasPrice breaks down into its base and tip components — resulting in the new transaction fee formula:

$$ TxnFee = effectiveGasLimit * effective(Base fee + Tip) $$

The same EIP1559 introduced the MaxFee concept — a maximum that a user is willing to pay for their transaction to be executed. This optional parameter (maxFeePerGas) has been adopted by the bulk of the smart wallet category as best practice, creating a fail-safe mechanism for transactions: the algorithm over-estimates the actual transaction fee intentionally, thus mitigating/covering fee deviations and delivering the transaction with success, regardless of how much the network gas price fluctuates. For a transaction to be executed, the max fee must exceed the sum of the base fee and the tip.

With the max fee implementation, the transaction formula is:

$$ txnFeeMax = maxGasLimit * max(Base fee + Tip) $$

So, even if it is impossible to predict what the actual on-chain cost of the transaction will be, with the EIP1559 implementation the algo ensures it will always be lower than what is initially calculated (and charged to/paid by the user). Secondly, the maxGasLimit parameter has an overhead of approx. 4%-40% Gwei, necessary to cover the transaction fee overestimation, as explained above regarding EIP1559 implementation. This can be taken into consideration and is actually where the cashback comes from, when enabling transactions using the APY Wallet Gas Tank.

For smart wallets, the maxGasLimit parameter has an extra overhead of 10%-20% of gasLimit, which comes from higher fees required because smart wallets deploy each transaction as a sort of small contract on the blockchain. This means that for each transaction fee, smart wallets calculate this parameter at an even higher cost than a standard wallet/EOA. However, the smart wallet advantage is that even though the transaction will cost more, it will also have more security, properties and transparency compared to transactions deployed through standard wallets/EOAs.

This last, 2-3K smart-wallet-specific overhead cannot be taken into consideration in the GasTank fee formula, nor considered for cashback, because of how the EIP1559 standard was created for smart wallets. Which is why when calculating the transaction fee for having the Gas Tank enabled, we have chosen to set aside 50% of the transaction fee difference between the max and effective fees, ensuring the transactions still go through seamlessly and do not get stuck or fail because of the specific smart wallet overhead.

The transaction fee in APY Wallet, when enabling the Gas Tank, is calculated as:

$$ txnFeeGasTank = (effectiveGasLimit * effectiveGasPrice) + 0.5*(txnFeeMax-effectiveTxnFee) $$

All in all, the difference between max (estimated) and effective (on-chain) transaction fee in APY Wallet comes from the EIP1559 integration and reflects both in maxGasLimit and maxGasPrice parameters — and this is where the cashback comes from, when Gas Tank is enabled. Since the maxGasLimit cannot be deducted entirely through the algo, the users get 50% of the difference between max and effective transaction fee as cashback, while the other 50% goes to APY Wallet as revenue stream. This is not intentional, but a consequence of EIP1559 limitations, as it is impossible to calculate this difference beforehand and account for the maxGasLimit in advance, without jeopardising transaction integrity/success.

APY Wallet's revenue model

Because APY Wallet is a smart wallet, it has an implicit revenue stream that is the byproduct of relayed transactions, as per transaction fee algorithm and calculation formulas, detailed above. This means that for each user transaction (or transaction batch) that is broadcasted to the relayer, APY Wallet collects the difference between the transaction cost that the user sets, approves and signs for — and the real, effective cost of the transaction, as is required for on-chain transaction deployment and after paying relayer fees.

As explained in the previous section, it is impossible to calculate this difference precisely, which is why the overhead/overestimation will always be a smart wallet transaction component, inherent to the EIP1559 standard's limitations.

However, APY Wallet has developed the APY Wallet Gas Tank as a feature that can generate savings and offer cashback to users that choose to enable it, thus empowering them to regain a big part of the difference between estimated and on-chain fees. The other method that users can employ to avoid fee overestimations is to use the wallet in 'relayerless mode', in this way broadcasting the transactions directly on-chain (but assuming the risk of having them get stuck or fail, while also remaining vulnerable to front-running and sandwiching attacks).

More details about the APY Wallet’s revenue model below, in WalletDAO revenue streams.

However, APY Wallet has developed the APY Wallet Gas Tank as a feature that can generate savings and offer cashback to users that choose to enable it, thus empowering them to regain a big part of the difference between estimated and on-chain fees. The other method that users can employ to avoid fee overestimations is to use the wallet in 'relayerless mode', in this way broadcasting the transactions

directly on-chain (but assuming the risk of having them get stuck or fail, while also remaining vulnerable to front-running and sandwiching attacks).

Email/password accounts

APY Wallet implements traditional authentication with an email and a password like web2 apps. This authentication mode is non-custodial: it works via an on-chain 2/2 multisig: one of the keys is stored in the browser storage and is encrypted with the user's password, and the other key is stored on our backend via a HSM.

You can't access the funds using only one of the two keys, for example if you're an attacker who successfully compromised either a user (e.g. via malware) or the HSM. However, a recovery procedure can be started with one key only. The recovery procedure is a timelocked change of one of the two keys. If the recovery procedure was unintended (e.g. initiated by an attacker), any other key holder can cancel it. But if it was initiated legitimately (e.g. if you lost one of the two keys), you can just wait for the timelock, and you'll have access to your account back after this.

To summarize, email/password accounts are multi-signature wallets, that unlock:

when 2 signatures are supplied; used in normal mode of operation

or when 1 signature is supplied, but with a timelock; used for password recovery, or when the APY Wallet backend becomes unavailable

The second key is normally unlocked by a confirmation code specific to (derived from a hash of) the transaction, but other authentication methods such as OTP 2FA or FaceID can be used.

An additional benefit of this model is that the second key can enforce extra security rules like spending limits and checking for malicious contracts or calls (e.g. infinite approvals to EOAs). Since those rules are checked off-chain by the HSM, they can be easily modified or enhanced. Furthermore, sophisticated checks can be performed at no extra gas cost, enabling use of AI or ML in the future.

Signers

Signers are the actual keys that can control your account. Think of them as multiple keys that open the same lock. Each signer can be invalidated at any time if you're authorized with one of the other ones and a new signer can be added at any time.

The advantage of this is that you can move control of your account to a new key without having to move funds and positions individually. For example, if you suspect that your hardware wallet has been compromised, you can easily swap it out for a new one by changing the signer, rather than having to move funds to a new wallet.

The signer settings are specific to each blockchain network. When you create a new account, you start with the same signers on all supported blockchains (e.g. Ethereum, Polygon), but from that point on you change signers individually for each chain! There are a few types of signers:\
Email/password signer: those are added by default if you create an account with an email and password. Under the hood, those signers actually consist of two private keys, one of which unlocked with your email via a confirmation code, and the other with your password. Both are required to make transactions immediately, but only one of them can also send transactions with a 3 day timelock. This allows you to recover your account in case you forget the password.

Hardware wallets: Ledger/Trezor/Grid+Lattice1. This is the recommended way of using APY Wallet with larger sums. If you only have a hardware wallet signer on your account, you achieve the same security as using a hardware wallet itself directly, but with all the added features such as gas abstractions, batching, etc.

Software wallets: those are wallets built into the browser or installed as an extension. We generally do not recommend this option unless you're a power user and you're already comfortable with those. The email/password signer is more secure in most circumstances because it's actually a 2-out-of-2 multisig under the hood. With those wallets, you have one private key kept in memory that if compromised, could be drained immediately.

Plugins

Plugins allow APY Wallet to be extended easily to support new DeFi protocols and functionalities, thereby making APY Wallet truly community driven. The advantages are clear:
Security: by reducing the need to interact with external dApps, we minimize attack vectors such as phishing
Convenience: it's a huge improvement to crypto UX if everything you need is under the same roof, following the same design language

Developer-driven: allowing the APY community to extend it increases the sense of involvement and allows developers to essentially build for themselves
Network effect: new DeFi protocols and dApps will be incentivized to build plugins to easily access the APY Wallet’s audience
One-stop shop: paves the way for APY Wallet to become the go-to tool for everythingDeFi/web3 related

Our plugin system is heavily inspired by Gnosis Safe APPS, and it uses the same underlying architecture right now. Over time, we plan on extending this architecture by allowing various forms of 'deep' integration, for example allowing plugins to declaratively add functionality to tabs like Swap, Earn and others. Currently, the Swap functionality is essentially a plugin for SushiSwap, and it's so seamless that it's perceived as part of the core UI.

APY & APYDAO

APY is the governance token of the APY Wallet.

The APYDAO is a decentralized autonomous organization governed owned by APY token holders. Its purpose is to govern APY Wallet development and integrations. Since wallets are the main user-facing component of crypto infrastructure, integration decisions are usually the most contentious. Especially if the wallet is designed to be a full-featured DeFi dashboard like APY Wallet. For example, what if two DeFi protocols are competing over being the default provider of the Swap functionality? Their respective communities will likely lobby for their protocol to win. The APY token and the DAO give the opportunity for everyone to put their money where their mouth is by actually acquiring APY and competing through a governance system designed to elect a clear winner.

APY Token Economics

The APY token ("**APY**" or "**A**") is the native currency for the APY network and the keystone for a new, inclusive, and borderless digital economy. If blockchains are digital infrastructure, the APY token is the fuel that powers the network.

APY has several important characteristics that make it the ideal currency for a new generation of games, consumer applications, and the digital assets that will power them:

- **Diverse** use-cases
- **Broad** distribution
- **Low** monetary inflation

Each of these is explained in more detail below.

Diverse use-cases

APY is the native currency for apps, games, and smart contracts built on top of the APY blockchain, and thus is the currency guaranteed to be available for developers and users to transact with on the network. Developers can easily build APY directly into their apps for peer-to-peer payments, charging for services, or enabling consumers to earn rewards for the value they create. APY can also be held, transferred, or transacted by users peer-to-peer.

Token holders can earn rewards by staking their APY as a security deposit and working to secure the network through running validator nodes – or delegating their stake to professional operators to run validator nodes on their behalf. Validator nodes receive staking rewards and transaction fees in exchange for providing the security, computation, and storage services the network needs.

Small amounts of APY token are also required for every activity on the network –

from new user accounts to storage for assets and smart contracts. As the network matures, APY token holders will be able to use their APY in an evolving number of ways:

- Payment for computation and validation services (i.e., transaction fees)
- Medium of exchange
- Deposit for data storage
- Collateral for secondary tokens
- Participation in governance

The perfect payment experience is seamless for all parties: buyers pay in any currency they have; sellers price and receive in any currency they want. Applications on APY can tap into this reality. APY has high throughput, low fees, and full ACID guarantees, allowing developers to implement decentralized exchanges (DEXs) that act as a clearing house between tokens. APY has the ability to use frequent batch auctions to defeat front-running attacks on these DEXs.

APY token's ubiquity on the network makes it the obvious "bridge asset" for currency exchanges between thinly traded token pairs. As the number of secondary tokens on APY becomes large, the number of possible trading pairs increases exponentially, meaning that some swaps will require an intermediary asset like APY

.

Importantly, APY is required for the creation and usage of all other tokens on the network – to pay for storage and/or serve as collateral. These details are outlined in the technical details section below, and will be fully specified in future whitepapers. The economic impact is that as more value is created on top of the APY blockchain, more demand is generated for APY token.

Broad distribution

The distribution of the native token is critical to the network's decentralization and long-term success. Centralized control over the token supply prevents easy access by developers, who require the native token to deploy new smart contracts and pay for transaction fees ("gas") as well as cover storage and account deposits on the network.

For APY to achieve its full potential, a safe and sustainable distribution strategy is critical: we must get APY token into the right people's hands. Alongside technical capability and crypto-economic security, we recognize that a healthy and sustainable distribution strategy is essential for making the APY blockchain successful in the long term.

APY will be the catalyst for diverse new communities to access blockchain and decentralized applications, building and benefiting from real use-cases rather than speculation. Good user experience design will make owning and using APY seamless. Ultimately, APY will bring all of the communities building on the network together to create and share value.

APY is pioneering several large scale engagement programs:

- **Cloudburst Partners:** organizations or individuals elected by APY holders to operate one or more APY validator nodes and distribute the rewards to developers, designers, artists, community organizers, and entrepreneurs building content for the APY network.
- **Floodplain Validators:** developers, infrastructure partners, and other ecosystem participants interested in supporting APY early and helping bootstrap the critical mass of content and decentralized resources necessary for a sustainable network.
- **Decentralized Reputation and Incentive Protocol (DRIP):** designed for accessibility and helping apps on the APY find an engaged userbase, DRIP distributes APY token to end users for purposes of, staking, delegation and active participation in the APY economy.

Thanks to robust technology, an amazing community, and strong incentive design, APY will enable today's early adopters to build tomorrow's open worlds.

Low Monetary Inflation

Blockchains like APY are powered by decentralized communities running the computer hardware ("validator nodes") that support the activity and secure the value of assets on the network. Other blockchains rely heavily on creating and distributing new tokens ("monetary inflation") in order to attract validator node operators to their networks.

Unfortunately, as in all economies, monetary inflation has a cost: the newly-created supply of tokens acts as a tax on holding or day-to-day usage by diluting all token holders. This is why APY has a cap on monetary inflation. In fact, inflation on APY will go down as network fees increase.

In its steady state, APY guarantees a set payout to node operators and only issues new tokens as necessary to make up the difference between transaction fees and that guaranteed payment. As transaction fees approach this payout amount, new issuance approaches 0%. If transaction fees exceed the payout amount, they are held in an escrow account and used to offset future inflation indefinitely.

In the first year of operation, monetary inflation will be higher to incentivize greater levels of staking while the collateral, payments, and other complementary parts of the APY economy mature.

Technical Details

Transaction Processing and Computation

As a decentralized network, APY charges users – or the apps they're using – for services on a per-action basis, similar to the way Amazon Lambda charges for processing power today.

There are two types of fees on the APY network:

- **Processing fees** cover the fees for a transaction to be submitted and included in a block.
- **Computation fees** are added for more complex operations that require computation beyond updating balances.

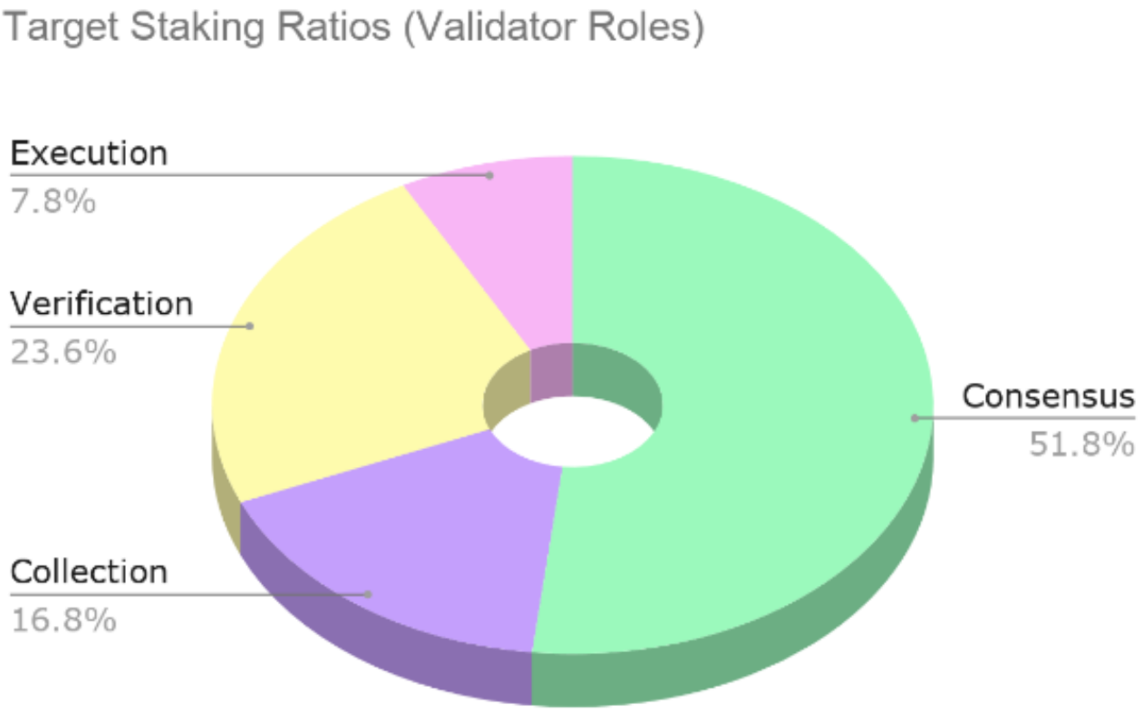
Especially in the early days of the network, transaction fees will be low, starting at 0.001 APY, or 1 mF (milli APY).

Inflation

As a proof of stake network, the APY blockchain requires validator nodes to lock a security deposit denominated in APY tokens in order to participate as part of the infrastructure that runs the blockchain. This is known as staking. Staking prevents low cost "sybil" attacks (where one actor masquerades as many individuals to gain undue influence over the network) and acts as a bonded deposit that can be seized if the validator attempts an attack on the network.

APY distributes a fixed portion of the total APY token supply each year as rewards to validator node operators through a combination of new issuance (inflation) and transaction fees (when combined with inflation, the "total reward"). The total reward will be chosen carefully to be as small as possible while preserving the security of the network (currently contemplated to be set at 3.75% of the total token supply per annum). Excessive inflation can create a range of unintended consequences and makes the token less attractive for non-staking uses.

To ensure that stakers are incentivized to move into the node roles that are most needed at any given time, the total revenue assigned to each role is adjusted through a set of multipliers known as the reward coefficients. These values are adjusted by the protocol automatically: if a particular role is consistently under-staked relative to the others, the protocol will increase the payouts to that node type until the actual staking balance converges on the target ratio. The initial split between node pools (calculated to optimize security) on the graphic below:



APY nodes follow the procedures defined in the protocol (based on their role) in order to receive rewards. Any deviation from the protocol can result in decreased reward payments or punishments. This reward and punishment structure is designed to guarantee the security of the protocol and optimize performance over time.

Severe infractions, which undermine the safety of the network, can lead to some or all of the staked tokens being confiscated from the offending node(s) and destroyed. This is known as “slashing”. This document outlines the most severe infractions against the protocol which result in slashing (“slashing conditions”). Enforcing these conditions is critical to the cryptoeconomic security of the protocol. APY considers only severe threats to safety and liveness to be slashable conditions and as such, there are no performance-related slashing penalties.

Stablecoins on **APY**

Stablecoins are cryptographic tokens whose value is stabilized relative to a given fiat currency – or basket of currencies.

The value of stablecoins in consumer applications and games is that, especially initially, mainstream consumers (and the businesses that serve them) may prefer to transact in their local currency. Similarly, businesses that need to make forward commitments will value predictability and the ability to book revenue in the same currency as their costs.

Stablecoins are simple to implement on APY – and several are already on the way. There are generally two kinds of stablecoins; both require APY tokens to pay for their network resources:

- **Fiat-backed stablecoins** are fungible tokens whose supply is based on an equivalent amount of fiat currency available for redemption, typically in an audited bank account. Like all APY users, holders of fiat-backed stablecoins still require a minimum balance of APY – this can be provided on their behalf by the application.
- **Algorithmic stablecoins** use APY token itself as collateral to create a secondary token whose supply is adjusted automatically to stabilize its value relative to given fiat currencies. APY has reserved a significant allocation of APY tokens to bootstrap the collateral for at least two implementations of algorithmic stablecoins on the network whose security is rooted in the native APY token itself.

Over time, as users learn to value APY for its functionality on the network, the native token may start being preferable as a medium of exchange based on its inherent liquidity and direct usage.

APY Token Distribution

The APY network was designed from the ground up as the foundation for a new digital economy. An economy that is owned and governed by its participants.

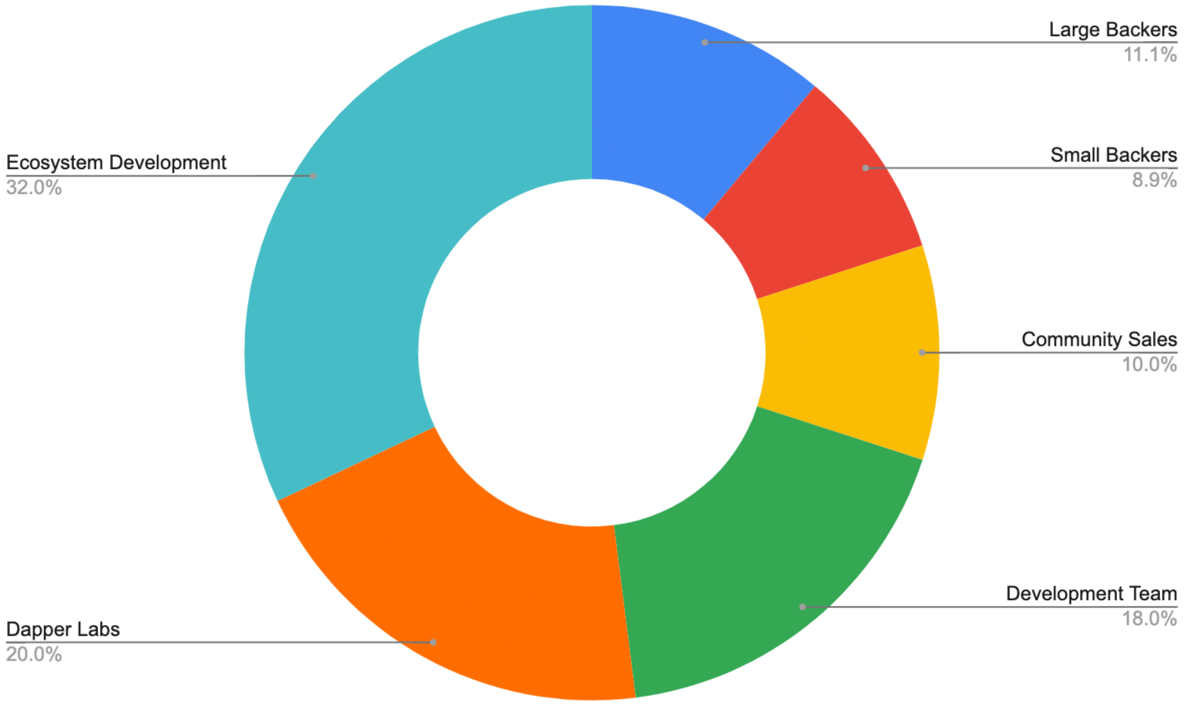
The ethos, architecture, and token economics of the APY network as a whole are covered in previously published documents:

The APY token is the native currency of the APY network, ultimately required for the network and all the applications on top of it to function. APY is designed as a payment method as well as long-term reserve asset for the entire APY economy. The token is a low-inflation asset that is used by validators, developers, and users to participate in the APY network and earn rewards. It is also used to transfer fees, serve as collateral for secondary tokens on APY, to pay for storage, and to participate in future protocol governance.

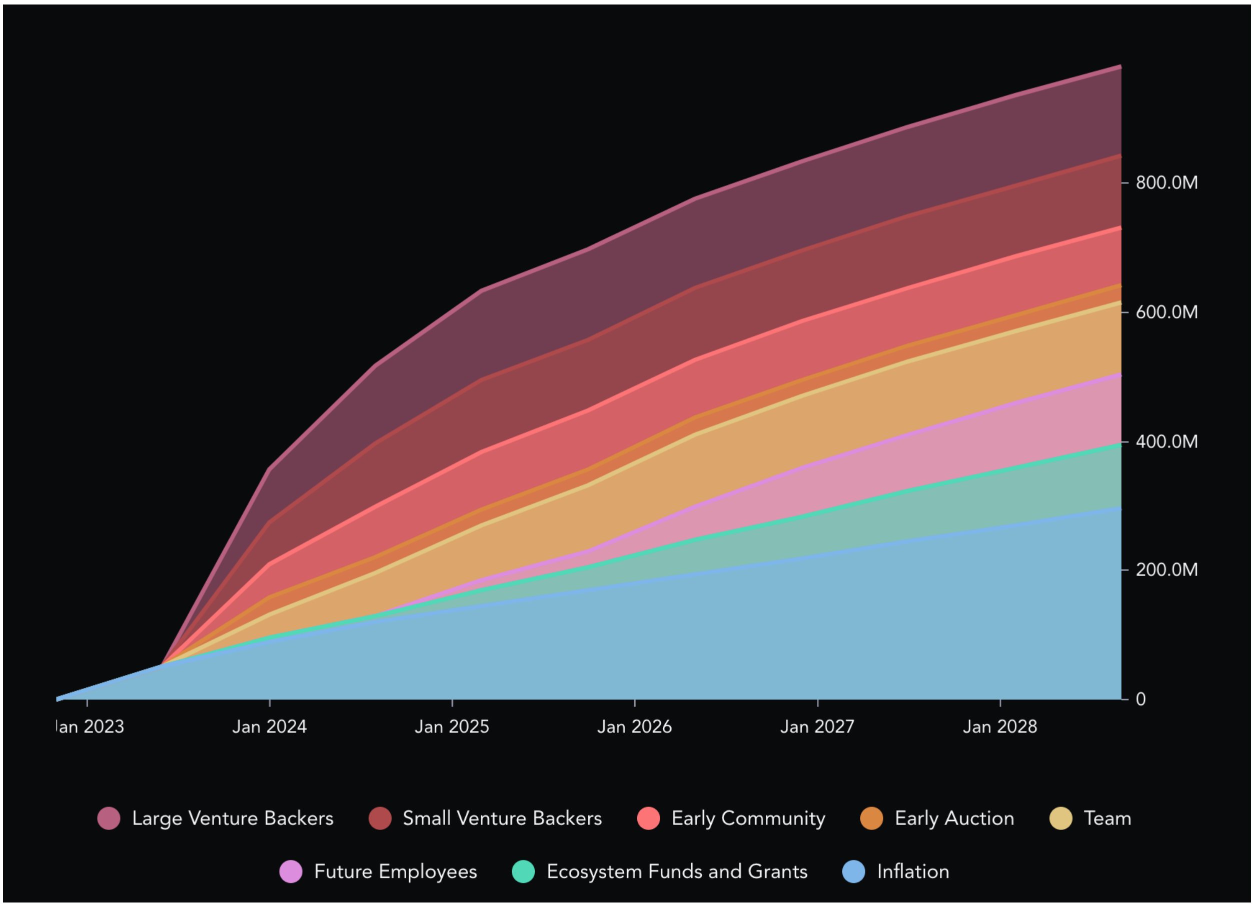
In the APY Token Economics paper, we outline the key principles of the APY token: diverse use-cases, broad distribution, and minimal monetary inflation. This paper will focus exclusively on the launch of the APY network and concurrent distribution of the APY token.

The genesis block was created in December 2021, with 1 billion APY.

For transparency, the breakdown of genesis block holders is outlined below:



Liquid Supply Curve



Short Term The Flow blockchain will have an initial bootstrapping phase to create the initial circulating supply for the network through staking rewards paid to anyone willing to lock their FLOW token and participate in the network. During the bootstrapping phase there will be around 7.2% monetary inflation. Staking rewards begin accruing after November 2, 2020 and are paid out every epoch (approximately weekly).

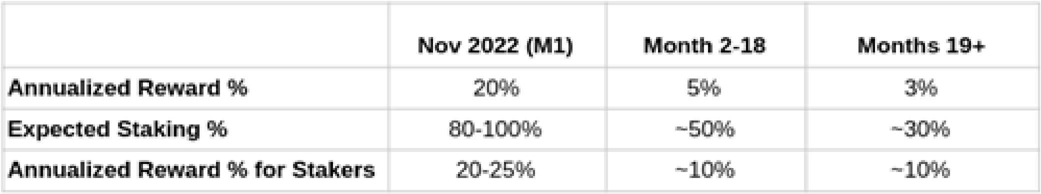
Long Term New issuance on Flow is used for staking rewards in cases when transaction fees are not sufficient to compensate delegators and validator node operators. On or before December 15, 2020, the total new issuance rate across the network is capped at 5% per year. This cap will go down to 3% after June 1, 2022. New issuance “tops up” the transaction fees to a minimum guaranteed income for node operators. If fees cover the income guarantees, new issuance goes to 0. Therefore, as the Flow blockchain gains adoption and fees increase, the issuance should eventually remain at zero while fees make up a majority of network rewards.”

Staking Rewards

Validator rewards were enabled on mainnet in December 2021.

These reward tokens are liquid for use on the network as soon as they are withdrawn by the node operator.

On APY, 100% of inflation is distributed to stakers – meaning holders of APY will not be diluted as long as you are actively participating. In other words, new issuance is only distributed to validators staking and performing work to support the network, or delegators directly pledging their tokens against a specific validator's dependability. While the community will ultimately be able to adjust reward parameters, an indicative inflation schedule is shown below.



	Nov 2022 (M1)	Month 2-18	Months 19+
Annualized Reward %	20%	5%	3%
Expected Staking %	80-100%	~50%	~30%
Annualized Reward % for Stakers	20-25%	~10%	~10%

The expected staking percentage is expected to decline over time as additional use cases become available, such as infused tokens. As this occurs, APY is designed to minimize inflation and will optimize this against providing sufficient rewards for validators.

Over the long term, Apy is designed to limit new issuance of APY tokens as much as possible, with a total target pool established to pay to validators consisting of i) transaction fees paid to the network; and ii) new APY tokens instantiated. New issuance is offset by the fees collected by the network. Because of this, high levels of transaction throughput results in lower annual issuance.

Ecosystem Development

400 million APY tokens have been set aside for ecosystem development to help bootstrap network effects and ensure a diverse and accessible community over the long term.

Recipients of Apy ecosystem support include entrepreneurial support organizations, non-profits, and academic institutions including Berkeley, Purdue, UC Davis, and Rochester Institute of Technology. These groups share APY credits with their communities and broaden accessibility.

APY ecosystem development programs are designed to reward the efforts of a decentralized community building sustainable value – not speculation. As a result, APY tokens distributed through these programs in the first year will be subject to lockups and transfer restrictions that expire no sooner than the first unlock date applied to early backers and the team. Ecosystem development programs also include token leases for purposes of staking, allowing reputable community organizations to participate in the network and earn rewards.

Development Team

APY has been developed and brought to market by one of the most innovative and interdisciplinary teams in the world.

The development team pool unlocks over 3 years, ensuring all stakeholders are aligned.

Dapper Labs

As the corporate entity that funded the development of the APY technology, Dapper Labs has been allocated 250 million tokens which it intends to hold as part of its long-term treasury.