SSV2.0

Based Applications
Protocol

Whitepaper
January 2025

V1.0

This document is a vision for a possible future presented by SSV Labs to the ssv.network community and is subject to DAO approval

Table Of Contents

Based Applications Protocol

Extending Ethereum's Security To All Applications

Abstract

The Ethereum validator set is one of the largest and most advanced in the blockchain space. With close to 1.1M active validators (~35M ETH) run by thousands of different entities, it represents the core ethos of Ethereum - a credibly neutral settlement layer. Ethereum's roadmap favors a polylithic approach for resolving technical challenges (scale, transaction ordering, etc) rather than a monolithic one. The term rollup-centric roadmap was coined by Vitalik Buterin in 2020 and adopted by the wider community. In this paper, we suggest a new term, Based Applications(bApps), to describe a method for re-utilizing Ethereum validators for bootstrapping off-chain services. This approach recognizes the unique properties of Ethereum's validator set as superior to other forms of security, unlocking an infrastructural layer for a wide range of services.

Ethereum’s Validator Set

Ethereum validators form the backbone of the Ethereum network's Proof-of-Stake (PoS) consensus mechanism, introduced with the Beacon Chain in Ethereum 2.0. Validators are responsible for proposing and attesting to new blocks, ensuring the network's security and decentralization. To become a validator, a participant must stake at least 32 ETH in a smart contract. As of 2024, Ethereum boasts nearly 1.1 million active validators, collectively securing billions of dollars worth of assets. Validators earn rewards for correct behavior, such as timely attestations and block proposals, but face penalties for inactivity or malicious actions. The most severe penalty, slashing, occurs if a validator is proven to act against the protocol's rules (e.g., signing conflicting attestations), resulting in the loss of a significant portion of their staked ETH and eventual removal from the validator set. Validators operate with uptime and latency constraints, relying on software clients like Prysm, Lighthouse, Teku, and Nimbus, often supported by robust server infrastructure to ensure reliability.

Distributed Validator Technology

Distributed Validator Technology (DVT) enhances the Ethereum validator set by increasing its fault tolerance, decentralization, and security. DVT enables a single validator to be operated across multiple independent nodes, managed by different operators, instead of relying on a single machine, operator, or software client. This reduces the risk of downtime or slashing caused by individual node failures, as the system can tolerate partial outages while maintaining consensus. Additionally, DVT promotes decentralization by allowing smaller entities to participate in staking collaboratively without requiring complete trust among operators. By improving resilience and lowering the barriers to decentralized staking, DVT strengthens Ethereum’s overall network reliability and security.

Properties of Ethereum Validators

The Ethereum validator set embodies critical security properties like liveness, distribution, size, and reputation, together establishing its role as one of the most secure and decentralized consensus mechanisms in the blockchain space. Liveness is guaranteed through the constant activity of a large and globally distributed set of validators, ensuring the network continues to process transactions and finalize blocks even during partial outages or regional disruptions. This robustness is further supported by Ethereum's protocol incentives, which penalize downtime and reward consistent participation, incentivizing validators to maintain high availability.

Distribution across close to 1.1 million validators significantly enhances the network's resilience against attacks. No single entity or group can easily coordinate a majority due to the sheer size of the validator set, reducing the risk of collusion or censorship. The size of the validator set also ensures diversity, with participants operating from different geographies, infrastructures, and network conditions, which minimizes correlated failures and enhances the network's decentralization. Additionally, validators' performance is intrinsically tied to their economic stake; slashing penalties ensure that any validator attempting to act maliciously risks significant financial loss and permanent exclusion from the network. This high economic cost of misconduct incentivizes honest behavior across the board.

Economics

Ethereum staking involves validators locking up ETH to secure the network, earn rewards, and validate transactions. The economics of staking revolve around several types of rewards: consensus rewards, transaction fees, and Maximum Extractable Value (MEV). Consensus rewards are earned by validators who propose and attest to blocks, ensuring the security of the blockchain. Transaction fees, or gas fees, are distributed to validators for including transactions in their proposed blocks. These fees can vary significantly, depending on network congestion.

Additionally, MEV opportunities arise when validators profit from ordering transactions within a block in a way that maximizes revenue—often by relying on arbitrage or liquidations. MEV extraction has become a significant aspect of staking economics. The introduction of MEV-Boost has enabled validators to access MEV opportunities more efficiently by collaborating with third-party block builders.

The broader staking ecosystem has also been shaped by liquidity concerns and solutions like liquid staking derivatives (e.g., stETH or rETH), which provide stakers with tokens representing their staked assets, allowing them to retain liquidity and participate in DeFi. The rise of liquid staking has contributed to an increase in total ETH staked, with over 30% of the total supply now participating in staking by late 2024.

Beyond standard validator rewards, additional incentives enhance the decentralization and robustness of the staking ecosystem. Platforms like ssv.network have launched incentivized mainnet programs to promote DVT adoption. Participants earn extra rewards for adopting distributed infrastructure, ultimately benefiting Ethereum's security by supporting diverse validator operations. Such initiatives not only contribute to the increased participation but also enhance the stability of the Ethereum network.

Penalties are an essential mechanism in Ethereum staking to maintain security and integrity by penalizing improper validator behavior. Penalties are predefined and depend on various factors such as performance and correlation. Penalties become increasingly severe when multiple validators engage in similar offenses simultaneously, thereby enhancing the network's resilience against collusion or widespread downtime. By enforcing clear slashing rules, Ethereum ensures validator accountability and incentivizes honest participation.

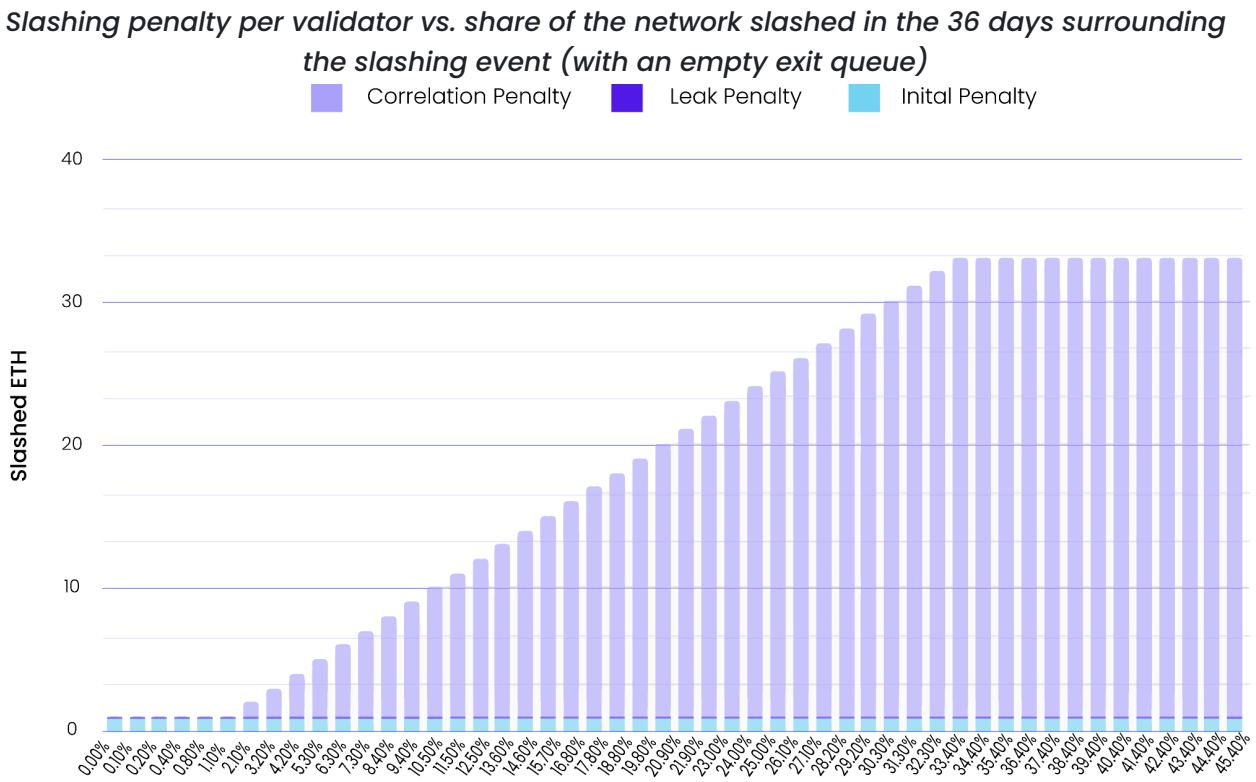

Figure 1: Ethereum slashable percentage as a function of aggregated slashed validators

The Bootstrap Problem

Cryptoeconomic Security of Proof-of-Stake Systems

To understand the Bootstrap problem, it's necessary first to examine the key concepts used to analyze the security of a PoS system. These concepts reflect the system's ability to deter attacks through a carefully designed set of economic incentives and penalties. A fundamental aspect of this security is the system's slashing mechanism, which penalizes validators for malicious behavior. Following definitions in the Stakesure by Deb et al., the key concepts that define the security of PoS systems include:

- **Cost-of-Corruption (CoC):** The capital loss incurred by an attacker due to slashing penalties during an attack.
- **Profit-from-Corruption (PfC):** The capital gain an attacker achieves from successfully compromising the system.
- **Cost-to-Attack (CtA):** The capital expenditure required to execute the attack.

For a PoS system to be cryptoeconomically secure, it must ensure that CoC exceeds PfC. Additionally, the system should aim for a high CtA, making it prohibitively expensive for an attacker to acquire the necessary capital to compromise the network. However, achieving these metrics is particularly challenging for new PoS systems.

Bootstrapping Challenges

The bootstrapping process is one of the most significant challenges faced by new proof-of-stake (PoS) decentralized systems. To establish a secure and functional system, these networks require a sufficiently large and distributed set of operators to validate and secure transactions. However, attracting and maintaining a robust validator set requires substantial resources,

participants, and economic incentives—factors that are especially difficult for smaller or emerging PoS systems to achieve.

In their early stages, PoS systems are particularly vulnerable to centralization, collusion, and attacks due to insufficient validator participation. This creates a feedback loop that can spiral into a negative cycle: a drop in the price of the staked asset reduces the system's CoC, weakening its security and prompting staker exits, further reducing the total value locked (TVL). The resulting decline in confidence worsens the price drop, leading, again, to a lower CoC, and a weaker and less secure network. Breaking out of this cycle is difficult for early-stage PoS systems, which often lack the resources and market presence to attract the validators necessary for robust security.

Existing Approaches to Bootstrap

To address the bootstrap problem, existing proof-of-stake (PoS) systems typically rely on two primary approaches: gathering resources independently or utilizing restaking. While both approaches aim to establish a secure validator set and ensure network functionality, they come with inherent challenges and trade-offs.

1. Gathering Resources Independently
The first and most straightforward approach is for the PoS network to independently gather the necessary resources to bootstrap its validator set. This involves attracting stakers who are willing to lock up their assets to secure the network. However, as previously discussed, this method is fraught with difficulties, particularly for new or smaller systems:

- **Attracting Stakers:** Convincing participants to stake their assets in a nascent system is a significant hurdle. Stakers must perceive the system as both secure and economically rewarding enough to justify the risks associated with staking, including slashing penalties and price volatility of the staked assets.

- **Management Complexity:** The network must manage the onboarding, monitoring, and incentivization of validators, which requires substantial operational resources and expertise. This is particularly challenging for smaller teams or projects with limited funding.

- **Centralization Risks:** A smaller or poorly distributed validator set can lead to vulnerabilities such as centralization or collusion, which affects the network's security and resilience.

2. Restaking

Restaking has emerged as an alternative approach, where existing stakers from one PoS system, such as Ethereum, reuse their staked assets to secure additional applications. While this model offers some advantages over independent resource gathering—particularly by leveraging an already established and distributed validator set, it also comes with some downsides:

- **Yielding Withdrawal Credentials:** For native restaking, participants must hand over access to their Ethereum stake to a 3rd party contract. This levies risk. It also locks participation to a single restaking platform.

- **Shared Risks:** Restaking involves sharing both capital and risks across multiple systems. The staked assets are slashable for the security of all the applications they are restaked to. This means that a failure or attack on one platform could cascade across others, jeopardizing the security of multiple systems simultaneously.

- **High Costs**: Validators participating in restaking take on increased risks and, therefore, demand higher rewards to compensate for the potential loss of their restaked assets. This increases the cost for new applications to secure their systems.

In summary, while restaking offers better access to capital and a more manageable way to bootstrap security compared to independent resource gathering, it still introduces risks. These trade-offs highlight the limitations of current approaches and the need for more secure, efficient, and scalable solutions to address the bootstrap problem for new PoS systems.

Based Applications

Overview

Critical applications (for Ethereum) should be based, utilizing Ethereum's validator set for their operations and security.

Validators are the backbone of Ethereum, providing essential services to the blockchain. However, apart from performing duties and securing Ethereum, for the first time we show that Ethereum validators can also serve other decentralized systems with their established Sybil resistance and staked capital.

Based Applications(bApps) are a new class of decentralized applications that leverage the Ethereum validator set for enhanced security, bootstrapping, and new capabilities. These applications make use of Ethereum's existing validator set(as opposed to capital) to quickly establish trust and operational resilience without having to build their own validator networks. Examples include based rollups, co-processors, oracles, bridges, and more. Additionally, bApps can serve novel applications like pre-confirmations, where validators can pre-confirm transactions before final inclusion in a block, reducing latency and improving user experience. By utilizing Ethereum's existing validator set, bApps can unlock new possibilities for scalability and user interaction while maintaining a high level of security and decentralization.

Re-utilization of L1 Validators (A New Asset Class)

Ethereum's validator set represents a robust and decentralized network of stakers. Its inherent security comes from the fact that each validator deposits 32 ETH as collateral to participate in securing the chain. This high capital requirement ensures that validators are both financially invested and aligned with the network's integrity. Such trustworthiness creates a stable backbone for additional use cases beyond Ethereum Layer 1.

Based applications extend the utility of Ethereum validators by using the validator as an entity (validation keys only). This means the principal (32 ETH) is never at risk of slashing, and withdrawal credentials are managed by the staker outside of ssv.network. While traditional models often rely on slashable collateral to enforce honesty, bApps instead leverage the substantial amount of non-slashable capital gained from the highly incentivized adoption of Ethereum validators.

By adopting validators' non-slashable capital, bApps unlock a new asset class. Validators can simultaneously participate in multiple decentralized applications, earning additional rewards without risking their 32 ETH. This re-utilization avoids duplicative validator networks and reduces the operational and financial overhead for applications that need robust security guarantees.

One of the most valuable properties of the Ethereum validator set is its inherent Sybil resistance, a fundamental security requirement for decentralized networks. A Sybil attack occurs when an adversary creates numerous fake identities or nodes to overwhelm a network, disrupting its consensus or compromising its integrity. Ethereum's PoS mechanism counters this threat by requiring validators to stake at least 32 ETH, creating a significant economic barrier.

As Sybil-resistant identities, Ethereum validators are not only indispensable to Ethereum's own security but also present a valuable resource for safeguarding other decentralized systems. By leveraging the trust and decentralization of Ethereum's validator ecosystem, bApps enhance their own security models, paving the way for innovative interoperability and shared security solutions.

Smaller or emerging decentralized bApps, which may lack the economic foundation to secure their networks effectively, can bootstrap with Ethereum's validator set. The bApps can leverage Ethereum's robust validator ecosystem, where Ethereum validators opt-in to perform duties on other systems.
This approach allows other networks to inherit Ethereum's trusted Sybil-resistant properties without needing to establish their own independent

validator base. By relying on the staked ETH that underpins Ethereum, these chains gain access to a globally distributed and highly secure validator network. This reduces risks associated with small or vulnerable validator sets.

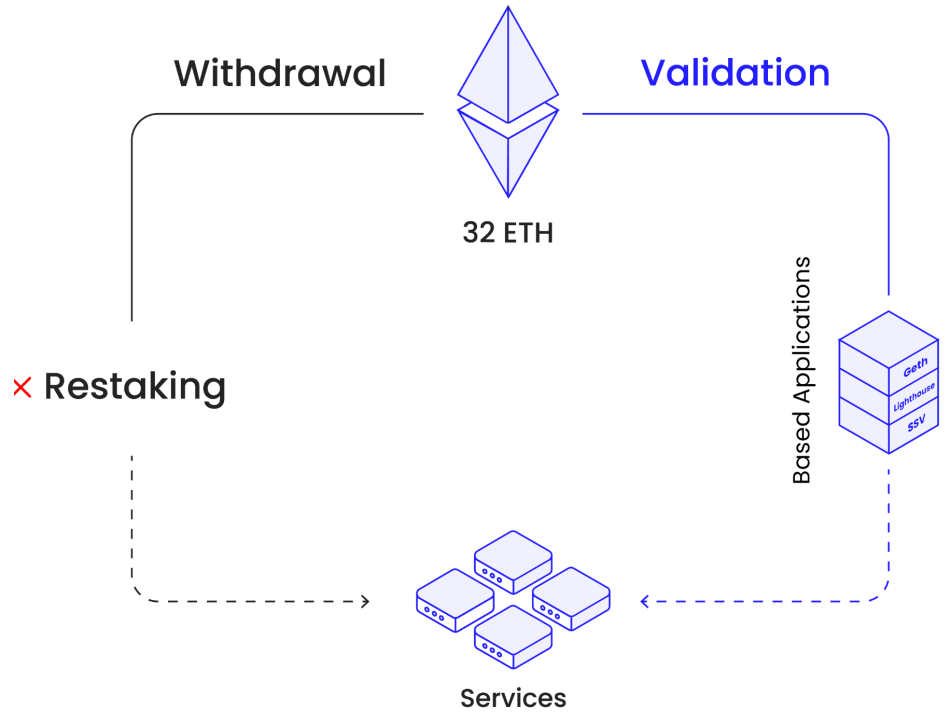

Figure 2: Withdrawal vs. Validation keys for Bootstrapping services

Slashable vs. Non-Slashable Assets and How bApps Use Them

Slashable and non-slashable assets refer to the potential risk of losing part or all of an asset's value due to certain actions or behaviors, particularly in decentralized networks like blockchain.

Slashable assets are those that can be penalized or "slashed" under specific conditions, typically due to misbehavior, such as a validator failing to fulfill its duties or acting maliciously. In Proof-of-stake (PoS) systems, slashing is a mechanism to ensure network security and accountability, incentivizing participants to act honestly and correctly. For instance, in a system like

Ethereum, validators can lose a portion of their staked assets if they attempt to double-sign or act in a way that harms the network.

On the other hand, non-slashable assets cannot be penalized directly. These assets are typically not involved in any accountability mechanism, so their value is unaffected by participant actions or misbehaviors within the network. Non-slashable assets can have a unique role in Sybil-attack (CtA, see above). In fact, under a single validator slashing condition in Ethereum, only ~1ETH is actually slashed (3.125% of the total stake).

Other protocols, like Eigenlayer, made the distinction between unique stake and shared stake. Unique stake refers to assets exclusively staked for a specific validator or service, where the slashing risk applies only to that specific instance. Conversely, shared stake involves assets being used across multiple services or validators, spreading the risk and thus potentially mitigating the chance of slashing affecting all staked assets. This mechanism offers flexibility and can enhance security while ensuring that assets can be used effectively without exposing them entirely to slashing risks from any single service or validator.

Based applications will be able to leverage both slashable and non-slashable forms of assets. Each asset type possesses distinct security properties and, as such, may play different roles in the protocol's design and operation.

Multi-chain Validator Participation

A multi-chain approach leverages validators from multiple Layer 1 blockchains, expanding the utility of several chains into a broader ecosystem. This approach can integrate validators from various blockchains, creating a more decentralized and resilient network for based applications (bApps). By drawing from different blockchain ecosystems, the model enhances scalability, security, and interoperability among applications across chains.

Integrating validators from multiple blockchains reduces reliance on a single ecosystem, mitigating risks associated with chain-specific vulnerabilities,

downtime, or governance issues. This diversity strengthens the overall security of the bApps operating within the network, as it is less likely that a failure or attack on one blockchain will disrupt the entire system. With validators coming from different chains, the network benefits from a variety of consensus mechanisms and security models, increasing its resilience against attacks.

The diversity of validator sets also contributes to improved network security. Each blockchain has its own unique security features, and by pooling validators from multiple chains, the network leverages these varied models to strengthen its foundational security. This cross-chain security architecture ensures that if one blockchain faces challenges, the validators from other chains can continue to maintain the network's integrity, minimizing the likelihood of downtime or disruption.

Furthermore, this multi-chain approach offers greater flexibility for bApps. Developers can choose validators based on specific application requirements, such as faster block times, decentralization of consensus, or more robust security features. This flexibility spurs innovation, as developers are not limited to the constraints of a single blockchain but can take advantage of the best features from multiple networks.

Based Vs. Restaked Apps

Restaking and bApps approach some of the same challenges in different ways. While Restaking relies on capital delegation with slashing risks, bApps also allows the utilization of L1 validators with no slashing conditions.

The different approaches result in different economic, technical, and practical differences.

	Risk	Cost	Participation model	Scale
Based Applications	Only on tokens	Low for L1 validators	Infinite-Sum game	All L1 validators can join risk-free
Restaking	Cascading slashing risk for Ethereum	High for all	Zero-sum game	Bound by L1 validator risk tolerance

Table 1: Comparison between based and restaked applications.

Infinite Vs. Zero Sum Games

A zero-sum game is one in which one participant's gain is exactly balanced by another's loss, whereas an infinite-sum game allows for the possibility of creating additional value through collaboration and growth.

As with all development platforms, the overarching goal is to enable infinite-sum games—collaborative marketplaces where the addition of new "apps" or services amplifies the overall value for all participants. In such ecosystems, growth is not zero-sum but exponential: the more participants and services added, the more valuable the platform becomes for developers, users, and stakeholders alike. This vision fosters innovation, incentivizes collaboration, and ensures that participation benefits everyone involved. However, the current restaking landscape struggles to realize this vision. Instead, it often devolves into a zero-sum game where applications and services compete for finite resources, impeding growth and collaboration.

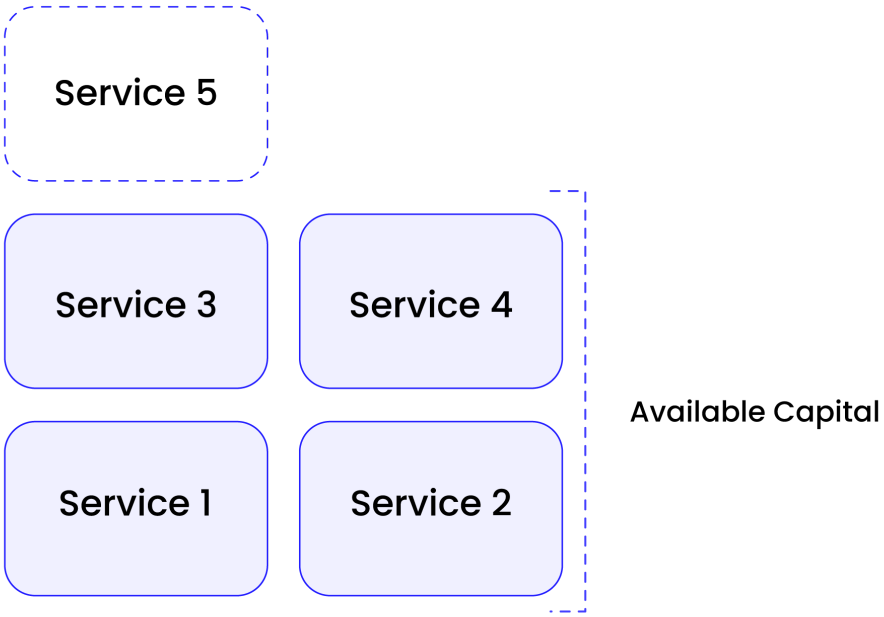

Figure 3: illustration of an account utilizing more than 100% of its capital.

To understand this limitation, consider the scenario where a restaker allocates 100% of their available capital to four services, as illustrated in Figure 3 (restakers are incentivized to maximize their opportunity costs by fully committing their capital). If a new service, Service #5, seeks to enter the ecosystem, the restaker faces a choice: either bring in additional capital to support the new service or reallocate funds from the existing four services. For most restakers, bringing in additional capital is unlikely due to constraints like risk appetite, liquidity, or opportunity cost. This leaves reallocation as the likely option.

However, reallocating capital means reducing support for existing services, potentially lowering the rewards from those services and introducing unnecessary friction. This creates a scenario where restakers are forced to weigh the benefits of participating in a new service against the losses incurred by reducing their stake in established ones. Consequently, new services struggle to attract the necessary initial capital to bootstrap their operations and gain traction within the ecosystem.

The result is a self-reinforcing barrier for new developers and services. Existing services, having already secured their share of restaked capital, enjoy a significant advantage, while new entrants are met with competition and resource scarcity. This dynamic undermines the fundamental aspiration of restaking as a mechanism for fostering growth and innovation.

Restaking was envisioned as a tool to unlock collaborative value creation, enabling developers to build services that complement and enhance one another. However, under the current framework, restakers' capital allocation becomes a **zero-sum game**, where supporting one service necessarily comes at the expense of another. This structural limitation not only stifles innovation but also discourages participation from new developers who see the challenges of securing sufficient capital as insurmountable.

For restaking to truly deliver on its promise of infinite-sum games, the underlying mechanisms must address these issues. After considerable research into bApps, we propose a different approach to solving the bootstrapping problem. This is described in the next chapter (under "Risk Expressive Model"), which solves this issue.

Formalizing Security

Slashable capital can increase both CoC and CtA, whereas non-slashable capital only increases CtA. In native restaking, all capital is potentially slashable, meaning it contributes to both metrics.

The core idea of bApps is to maximize capital efficiency by leveraging the existing sybil resistance provided by Ethereum validators without risking their main staked ETH. Instead, validators are only required to provide proof of their validator status, which grants them access to secure other bApps. These bApps may still require additional capital commitments from participants, which can be either slashable or non-slashable.

Shared security, as described above, introduces significant costs, including opportunity costs from alternative uses, contract risk, and slashing risk. In contrast, relying on validators for security provides a more efficient approach.

Validators' core ETH remains non-slashable, reducing their risk exposure while contributing effectively to security. This flexibility allows bApps to define security parameters with reduced risk for validators. As a result, bApp economics feature a higher Cost-to-Attack (CtA), offering a compelling opportunity for validators to diversify their participation without significant slashing risk. Compared to other models like restaking, bApps provide a more predictable and stable pathway to secure emerging protocols.

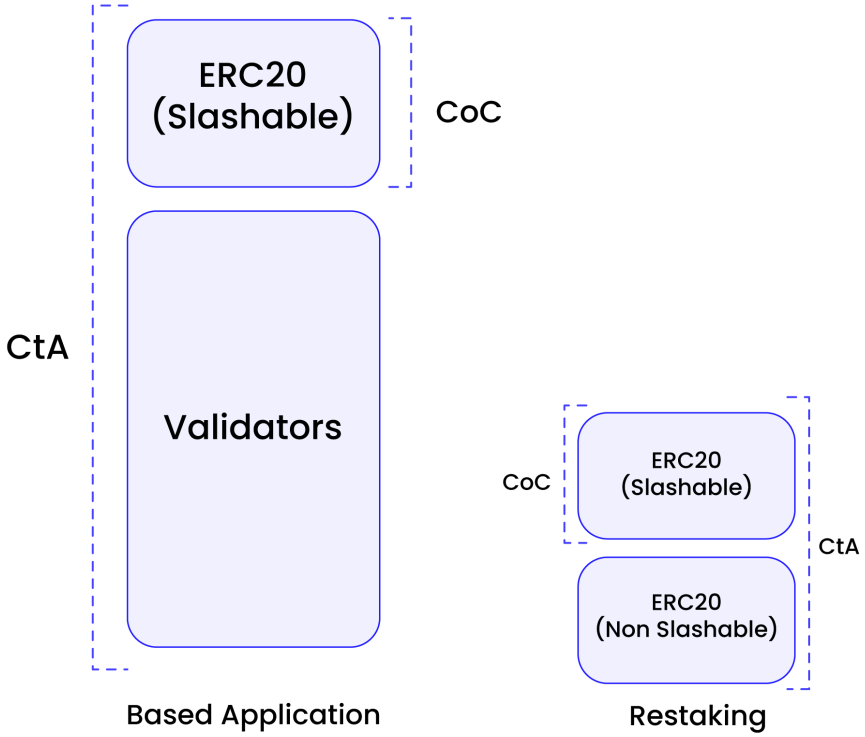

Figure 4: capital requirements: bApps Vs. Restaking

Economic Comparison for Protocol Builders

From the perspective of protocol builders, choosing the technology to secure their systems sufficiently requires a clear understanding of the trade-offs between based applications and other platforms, such as restaking. This section explores numerical examples of these comparisons, emphasizing the unique advantages of based security.

Rewards for bApp Stakers

A bApp developer allocates rewards to incentivize stakers to allocate capital to the bApp, increasing the total staked capital and improving security against attacks. It also incentivizes participants to follow the protocol and earn the rewards, discouraging attacks.

The APR for slashable and non-slashable assets can reasonably be expected to be different, given that they are under different risks. Rewards would most likely reflect the above; thus, non-slashable assets' rewards are expected to be lower than those of slashable assets.

For the following examples, the APRs for slashable and non-slashable capital are assumed to be 10% and 1.75%, respectively. In practice, the numbers may vary. The APR, or reward, is also denoted as service cost since it represents the app's expenditure to increase its attractiveness.

Case 1: Same Rewards To User / Same Cost to bApp

As a bApp builder, it is desired to attract more stakers to participate and secure the bApp given a certain amount of rewards. Denote n as the number of 32 ETH stakers and f as the fraction required to corrupt the service. The following table shows the security the bApp can get from using a restaking ecosystem compared to SSV 2.0.

	Restaking	bApp	Restaking/bApp
CoC (user risk)	$fn*32$	$fn*26.4$	82.5%
CtA (user cost)	$fn*32$	$fn*58.4$	182.5%
Total Reward (service cost)	$n*3.2$	$n*3.2$	100%

Table 2: CoC/ CtA and rewards comparison between bApps and restaking

With a bApp, some rewards are distributed to non-slashable capital contributing to CtA; others are distributed to slashable capital contributing to CoC. With the same amount of rewards, more non-slashable capital is

attracted because the rewards are risk-free, while there are still some slashable stakers who take risks for higher rewards. In Table 2, we assume the bApp gets a larger amount of non-slashable stake. We reduce the amount of slashable stake to equate the cost of restaking. As a result, compared to restaking, the bApp gains 82.5% extra cost-to-attack in exchange for 17.5% cost-of-corruption under the same amount of reward distribution, effectively raising the capital barrier for attacks. The bApp owner can also account for high CtA and offer lower rewards for slashable capital, thus lowering its cost.

Case 2: Same Slashable Stakers

Assume in either restaking or bApp, the same amount of slashable capital will be attracted to secure the application. Since non-slashable capital carries no risk, assume that an extra ETH validator is securing the bApp. The table below shows the security comparison in this situation.

	Restaking	bApp	Restaking/bApp
CoC (user risk)	$fn*32$	$fn*32$	100%
CtA (user cost)	$fn*32$	$fn*64$	200%
Total Reward (service cost)	$n*3.2$	$n*3.76$	117.5%

Table 3: same as table 3, same slashable stakers

In this case, the bApp is secured not only by the same CoC, but also by the additional CtA, which is doubled compared to restaking, with only 17.5% additional rewards distributed to the additional non-slashable capital.

Security Insights

We can see that in both cases, there are trade-offs between restaking and bApp to increase CtA with decreased CoC or increased rewards, but since in bApp CtA is cheap to get, the security benefits from increased CtA outweigh the costs.

Cascading Risks

Restaking on Ethereum carries considerable risks, particularly the concern that a large-scale slashing event triggered by an AVS could result in the exit of potentially millions of ETH from staking. Such an incident could destabilize the staking ecosystem, diminish trust, and create significant turmoil in Ethereum's stability. The ripple effects of such an event would not only impact individual stakers but could also undermine the overall network security, leading to a reduction in the number of active validators and a decrease in the network's resilience. The loss of millions of ETH from staking would have broader implications, including reduced liquidity and diminished confidence among participants who rely on staking for consistent rewards. Such a scenario underscores the inherent vulnerabilities and systemic risks associated with restaking, particularly when multiple protocols or applications are interconnected.

Conversely, with based applications, validators are always protected from such cascading penalties, providing a more secure environment for those participating in the network. This safeguard is crucial as it reduces the risks for validators, encouraging more participants to stay engaged in the staking process. Additionally, the ability to amplify staking rewards by re-utilizing validators presents a compelling opportunity to significantly improve the economics for solo and small-scale stakers. By reusing validators across different applications, participants can maximize their opportunity costs without taking on additional infrastructure costs, which is especially beneficial for those who may not have the resources of larger staking operations. This ability to leverage existing validators more effectively helps democratize access to staking rewards and levels the playing field for smaller operators.

Protocols like Lido, Rocketpool, and Ether.fi offer permissionless operator options, allowing these operators to opt-in and use their validators to secure bApps, thereby boosting their APR considerably. This flexibility enables validators to participate in multiple layers of security provision, increasing their overall yield and making staking a more attractive proposition for both new and existing participants. By allowing permissionless operators to enhance their validator utility, these protocols will foster a more decentralized

and resilient network, where individual operators are empowered to take part in the broader ecosystem beyond just securing the base layer of Ethereum.

This paves the way for a new economy, making solo and small-time operators more profitable than ever and strengthening their incentives and long-term sustainability within the network. The increased profitability means that more individuals and small entities can afford to participate in staking, thus enhancing the decentralization of the network. A more decentralized validator set not only contributes to the security and robustness of Ethereum but also ensures that power is not overly concentrated in the hands of a few large entities. The emergence of this new model, driven directly by bApps' increased rewards and reduced risk, holds the potential to transform Ethereum's staking landscape, making it more inclusive, resilient, and ultimately contributing to the long-term success and stability of the entire ecosystem.

Use-Cases

Category	Use Cases	Description	Validator Set Importance
DeFi	Oracles, Cross-Chain Bridges, Advanced Financial Protocols.	Utilize validator security to provide reliable pricing, liquidity transfers, and novel financial mechanisms.	Requires strong decentralization and reliability to prevent fraud, ensure data accuracy, and maintain trust.
Data	Data Availability, storage, Rollups.	Provide guaranteed data availability for Layer 2 rollups and other off-chain data solutions.	A strong validator set ensures reliable data publication, mitigating risks related to data withholding or loss.
Off-Chain Computation	Co-processors, Ad-hoc execution, Verifiers.	Offload computation and storage to a decentralized network while maintaining verifiability.	Requires trusted validator nodes to ensure computation results and data are correct and verifiable.

Security as a Service	Fraud Proofs, Slashing Mechanisms, Firewalls, Attack Detection.	Offer additional security services to other blockchain protocols like slashing and fraud detection.	Needs an equally robust validator set to that of Ethereum to uphold network integrity, enforce accountability, and provide effective deterrence against attacks.
Middleware Extensions	Relayers, Indexers, Event Streaming.	Middleware infrastructure to bridge different blockchain layers or to provide specialized services.	The validator set must be trustworthy to maintain secure cross-layer communication and accurate data indexing.
Governance & Coordination	DAOs, Collective Voting Systems, Community Funds.	Facilitate decentralized governance and coordination processes in a transparent and trustless manner.	A decentralized validator set is essential to ensure decisions are executed without manipulation or central authority.
Cross-Chain Communication	Interoperability Layers, Message Passing Protocols.	Enable communication and asset transfer across multiple blockchains	A secure validator set ensures the reliability and trust in communication between distinct blockchain ecosystems.
Economic Security Extensions	Bonded Validators for Collateralized Systems.	Provide collateral and bonded services for systems requiring economic guarantees.	A validator set as strong as Ethereum ensures economic guarantees, minimizing risk of financial loss due to validator failures.
Validator Commitments	Pre-Confirmations, Based Sequencers, Transaction Ordering.	Validators provide additional services like pre-confirmation of transactions, sequencing, and ordering to reduce latency and optimize blockchain efficiency.	A robust validator set is essential for pre-confirmations to provide guarantees on finality, while based sequencers ensure fair and predictable transaction ordering to enhance network performance and mitigate risks associated with transaction manipulation.

Table 4: potential use-cases for bApps

SSV 2.0 - A Based Applications Protocol

SSV 2.0 upgrades will focus on the following:

- **The Based-Applications Chain:** A dedicated "App-chain" to encompass both existing DVT contract and bApp features.
- **Risk Expressive Model:** A novel risk-based model for describing how operators opt-in to secure bApps and the relationship between them.
- **Ultra-Sound SSV Token Model:** A new deflationary model for SSV, including SSV staking and burning.

The Based-Applications Chain

In its current form, the ssv.network uses Ethereum as a coordination layer for: registry management and settlement. Based applications require a similar level of coordination to effectively track which operators have opted-in to support a given bApp. This coordination ensures operators are correctly assigned to roles and duties within the bApp's operational framework.
One option is to continue building on Ethereum as a coordination layer. This approach introduces 3 main limitations:

- **Scale:** Both DVT and future bApp transactions will include significant amounts of data, which are required to be posted as calldata (requires persistence). EVMs have calldata limitations that prevent operations at scale
- **Cost:** Because of the reliance on calldata, gas costs for simple DVT operations are high (even with batch transactions)
- **Multi-Chain:** The 2 above limitations could be reduced significantly by using Ethereum L2s. However, that will further harm the feasibility of

ssv.network being a multi-chain protocol as a heavier dependency will be made on a specific chain.

To solve all 3 of the limitations above, a dedicated, ***based-applications chain*** (bApps chain) needs to be built. Such a chain will encompass all current DVT operations and all future bApp operations, while remaining significantly "lighter" than Ethereum and offering cheaper transactions. Such a dedicated chain will be credibly neutral to enable the extension of the ssv.network to multiple L1s.

Security

Securing the bApps chain can be achieved by developing it as a based application itself. In this model, Ethereum validators—and eventually validators from other networks— can opt to contribute to the app chain's Cost-To-Attack(CtA). Additionally, SSV staking is employed as slashable capital, acting as a robust form of Cost-Of-Corruption (CoC). This dual-layered approach aligns incentives with penalizing malicious or unreliable behavior. Over time, as the app chain integrates validators across networks, this model will scale to provide unparalleled resilience and trustworthiness for users and developers alike.

Light Client

The bApps chain can utilize light clients to ensure efficient and streamlined participation, allowing nodes to verify chain state without the need for full data storage or intensive processing power. By leveraging a partially synchronous BFT consensus like CometBFT, the application chain enables rapid finality and easy synchronization, making it ideal for light clients. These lightweight clients can sync quickly and validate blocks effectively, providing a simple yet reliable interface for bApp developers. This approach reduces the barrier to entry for developers to build based applications and the operators running their code.

Operations

Validator management is a continuous process done by the bApps chain to ensure only active L1 validators can participate in securing bApps. An L1 validator's initial onboarding consists of registering to the chain (recommended to be run as a DVT cluster). Once registered, the chain can continuously verify the validator is deposited and active. Active validators can then be used to secure based applications, directly using their key or via some proxy key.

The application registry is responsible for synchronizing all bApp users on the current state of applications, their operators, and their respective weights for tasks. The registry is the cornerstone of each bApp; it's the deterministic state each bApp **has** to have for its consensus to be secure.

The coordination itself must be deterministic. At any point in time, all validators in the network should be aware of which peers have opted in, their respective weight, and the rules of consensus. Deterministic coordination ensures reliability, consistency, and a predictable state that all participants can trust, laying the foundation for seamless interactions between validators and bApps.

Multi Chain

A dedicated 'application chain' inherently creates a neutral-based chain, which serves as a crucial layer in eliminating chain-specific dependencies. By being solely focused on coordination and integration, this application chain does not favor any particular Layer 1 (L1) blockchain, thus promoting a decentralized and unbiased environment for validators from multiple L1 chains. It avoids the complexities and risks often associated with cross-chain interactions, where different blockchains might have conflicting protocols, governance models, or consensus mechanisms.

The neutrality of the chain ensures that it can interact with a wide range of L1 blockchains without being inherently tied to or dependent on any one of them. This removes the friction that typically arises from direct L1-to-L1

interactions, which can involve cumbersome compatibility issues, differing security models, or governance conflicts. With a dedicated neutral chain, validators can participate without needing to navigate the intricacies of each individual blockchain, making the overall ecosystem more accessible and less prone to fragmentation.

By being lightweight and focused solely on coordination, a neutral bApps chain reduces overhead and simplifies the system. This lightweight nature allows for more efficient operation, as there are fewer dependencies and less complexity in managing multiple blockchains. The chain acts as an abstraction layer that streamlines interactions between different L1 blockchains, allowing them to coexist without the burden of cross-chain compatibility issues. This simplifies the network's architecture, ensuring faster processing times and a more scalable solution for decentralized applications (bApps).

The neutrality of the chain also means it can quickly adapt to new or changing blockchains. As long as an L1 chain can integrate into the coordination framework, it can become part of the system without needing to alter the underlying architecture. This flexibility makes it easier to scale and evolve the network, ensuring that future blockchain advancements or improvements can be seamlessly incorporated without disrupting the ecosystem.

Risk Expressive Model

Restaking models identify “unique stake” as slashable capital allocated to a single (and specific) bApp. In practice, that means that a user with 100 units of capital can only allocate up to 100 of those units as unique slashable capital. Further examination of this model reveals it to be a zero-sum-game model in which any new bApp requiring operators to allocate unique capital comes at the cost of some other bApp.

The Risk Expressive model described below addresses this capital inefficiency.

The Model

In the based application model, operators participate in bApps using a Risk Expressive Model (REM), where opting into a bApp implies a commitment that comes with certain obligations. These obligations introduce a risk of future capital slashing if specific conditions or performance standards are not met. This model makes operators financially accountable for the reliability and efficiency of the bApps they provide to the network.

Each bApp an operator opts into increases the level of responsibility and, consequently, the risk it poses to the bApps in which it participates. This means that as operators take on more bApps, they face greater risk of financial penalties, making it essential for them to balance their participation in bApps with the risk they are willing to assume.

A scoring mechanism is integral to this process as it configures the weight of operators based on their level of risk. The weight represents an operator's influence within the network, and the scoring mechanism dynamically adjusts this weight to reflect the amount of risk each operator has taken on. By doing so, the system encourages responsible behavior among operators, ensuring that those who take on higher obligations (and therefore more risk) have their influence properly calibrated. This not only maintains fairness but also helps protect the overall stability of the network.

Different bApps within the network can tailor their risk configurations to match their needs by utilizing a configurable β parameter. Smaller bApps, which may face challenges in attracting sufficient capital, might adopt a more lenient risk approach, represented by a lower β value in the risk graph. This lower β value implies a greater appetite for risk, making these bApps more accessible to operators by enabling them to opt-in to more bApps and maintain more of their weight in each one. On the other hand, larger, more established bApps that have significant capital may choose to set a higher β value. A higher β indicates a preference for attracting operators who can commit to stringent performance requirements and who are comfortable assuming more obligations, thus signaling a lower tolerance for risk. Additionally, a bApp can

assign distinct β values to each token it utilizes, allowing it to fine-tune its risk tolerance based on the role and significance of each token.

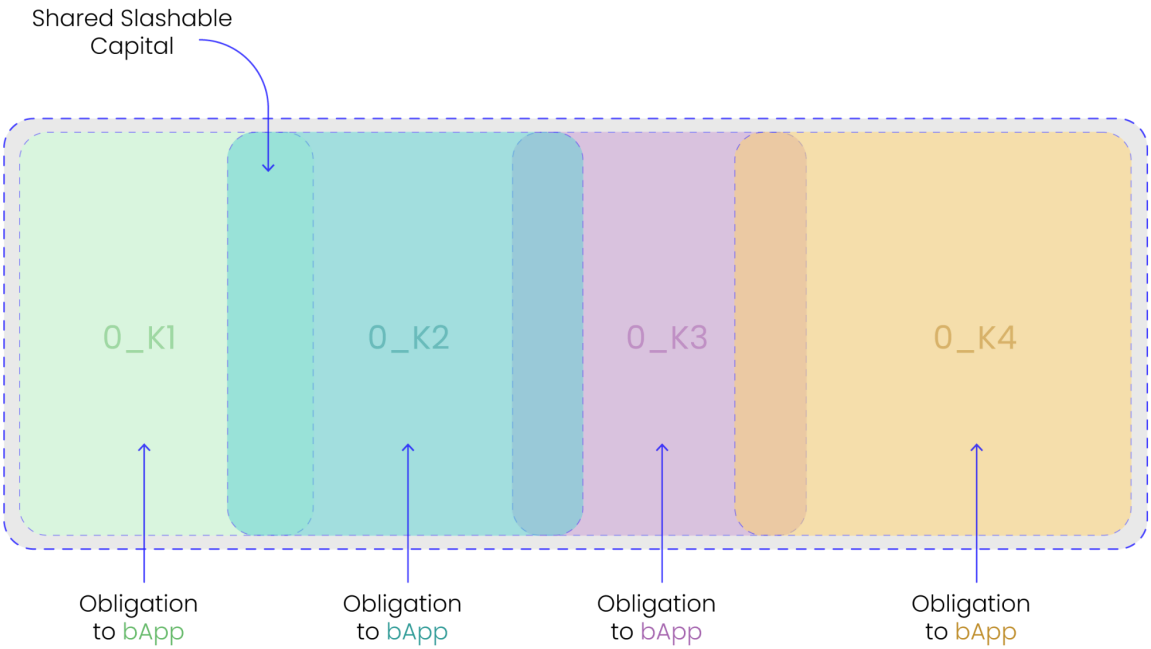

Figure 5: illustration of an account utilizing more than 100% of its capital.

This Risk Expressive Model allows the network to balance capital allocation and security needs effectively across various bApp types. By enabling bApps to adjust their risk levels using β values, the model ensures that both small and large bApps can attract suitable operators while maintaining a consistent level of reliability and accountability across the ecosystem. The Risk Expressive Model, in conjunction with the scoring mechanism, provides a structured yet adaptive approach that motivates operators to engage meaningfully while being mindful of their obligations and the associated risks. This ultimately fosters a robust and resilient environment where operators are encouraged to contribute responsibly, ensuring the long-term stability and efficiency of the network.

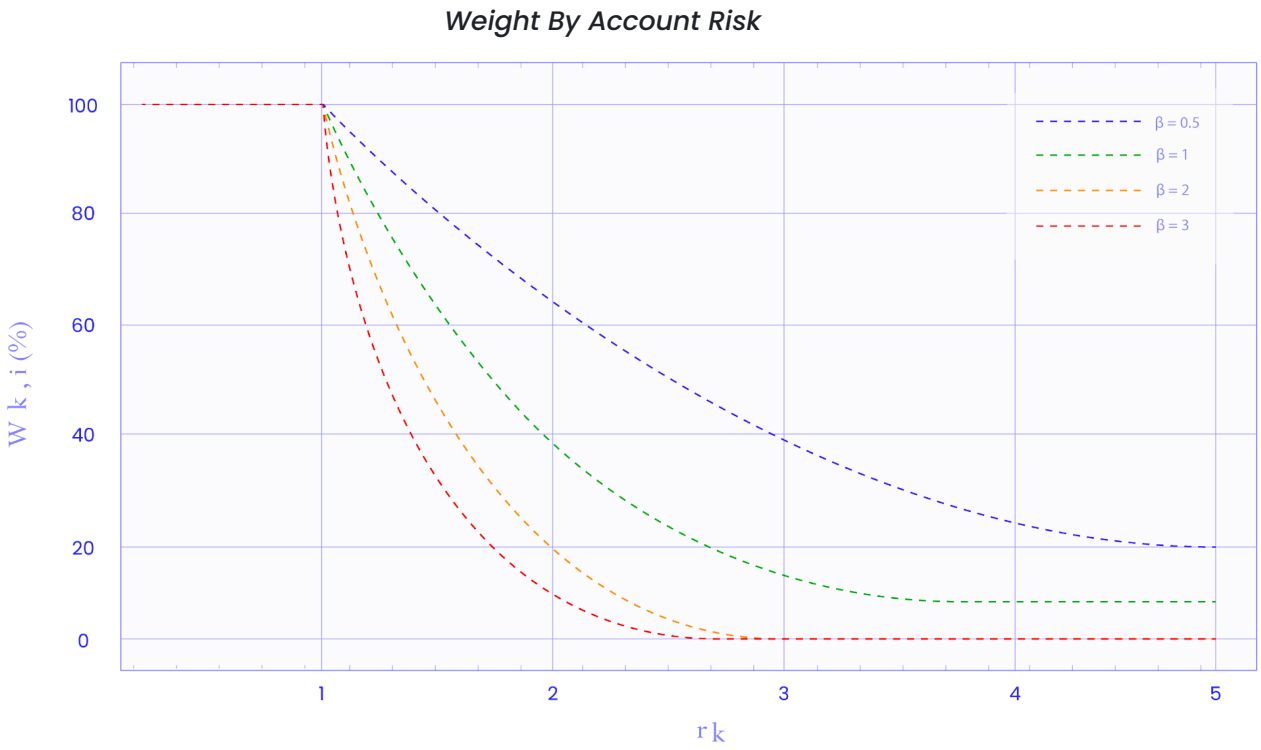

Figure 6: operator weight as a function of its risk, $r(k)$, and β.

Formalizing

Let's define:

- $v_{k,i}$: number of validators assigned to service i by account k.
- v_i: total number of validators in service i.
- $o_{k,i}$: obligation for account k in service i (i.e. the slashable capital from account k available for service i).
- c_i: total obligation (slashable capital) in service i.
- $o_k = \sum_{i \in \text{Services}} o_{k,i}$: total obligation for account k.
- $assets_k$: total slashable assets of account k.
- $r_k = o_k / assets_k$: obligation ratio for account k.

Note that we can have $r_k > 1$, which means that $o_k > assets_k$ and the account has some slashable capital that is associated with more than one service.

The *weight* of an account k in a bApp is denoted by $W_{i,k}$, defined as a value in $[0, 1]$ (*or* $[0\%, 100\%]$). Usually, it would be defined by some *participation ratio,* such as $p_{k,i} = o_{k,i}/c_i$(the fraction of slashable capital participation), or $p_{k,i} = v_{k,i}/v_i$(the fraction of validators). For flexibility, the bApp can define the function that best suits its needs.

However, to take the account's risk into consideration, we can adjust the weight by combining the participation ratio $p_{k,i}$, with the account's risk, r_k, in the following way:

$$W_{i,k} = c \times p_{k,i} \times e^{-\beta \times max(1, r_k)}$$

where β is a hyperparameter that the bApp can adjust according to its security necessities, and c is a normalization constant which can be computed by:

$$c = \left(\sum_{k \in \text{ Accounts}} p_{k,i} \times e^{-\beta \times max(1, r_k)} \right)^{-1}$$

We take $max(1, r_k)$ to avoid the issue in which $r_k \to 0$, artificially increases the voting power of an account with almost no obligation and risk.

A numerical example of this model is provided in Appendix A1.

Multi Token Model

The previous model enables bApps to assign accounts' weights for a specific token based on their obligations and associated risks. To extend this framework to scenarios where a bApp seeks security through multiple tokens, these weights can be combined to calculate the account's final weight, $W_{k,i}^{final}$
In this case, the bApp should define a combination function tailored to its specific needs. Common examples include the arithmetic mean, geometric mean, harmonic mean, or any weighted variant.
For example, suppose a bApp uses tokens d_1 and d_2, and considers d_1 to be twice as important as d_2. Then, letting $W_{k,i,d}$ to be the weight of account k in

bApp i for the token type d, it could use the following weighted harmonic mean function:

$$W_{k,i}^{final} = \frac{1}{\frac{2/3}{W_{k,i,d_1}} + \frac{1/3}{W_{k,i,d_2}}}$$

In this context, the bApp should define a specific β_d value for each token based on its risk tolerance. Also, an important observation is that, specifically for the non-slashable ETH form of capital, the participation ratio ($p_{k,i,\text{NS ETH}}$) should be used instead of the weight function ($W_{k,i,\text{NS ETH}}$), as this type of capital carries no inherent risk.

Ultra-Sound SSV

With the development of the "Based Applications Chain" the SSV token model will change drastically, expanding the use of the SSV token and making it a three-dimensional fee token with deflationary properties.

We present a path to make SSV a deflationary token, or as the meme world suggests: Ultra-Sound SSV.

SSV Staking

For the bApps chain's security, a new Based Application will be deployed to manage the operator set, securing the chain, collateral requirements, slashing, and rewards. The chain will use the "Risk expressive Model(REM)" to manage each operator's voting weight and rewards (according to the number of validators, staked SSV, and obligation ratios) and slashing conditions.
As done with the Ethereum blockchain and others, and in order to guarantee the integrity of the bApps chain, SSV tokens will be **staked**, exclusively, as a slashable commitment for validators. Misbehaviours will cause penalties in SSV.

Operators can use delegated validator balance and/or SSV from delegators. Operators with more capital are entitled to more tasks and rewards.

Fees

Participation in DVT staking or based applications will require paying fees in the form of network fees. Currently, SSV charges network fees for running validators on clusters, that is, within the context of fees, a one-dimensional structure. The new fee mechanics will create a three-dimensional fee structure for SSV:

Category	Fee	details
F1 - L1 staking	Fixed per validator	L1 validator network fee, set by DAO to be 1% of Ethereum APR
F2 - bApp participation	Per bApp usage	Each bApp “opt-in” requires an additional network fee
F3 - Tx Fees	Per congestion	Transactions on the bApp chain require fees from the sender

Table 5: new fee structure

All collected fees from the categories above will either be distributed as rewards to operators or burnt.

Rewards

Fees accumulated will be distributed to each operator based on its obligation to the bApp. Operators can set a percentage of the amount of rewards to be distributed to accounts that delegate validators and/or SSV tokens to them. The remaining part of the rewards are kept by the operators as operator fees.

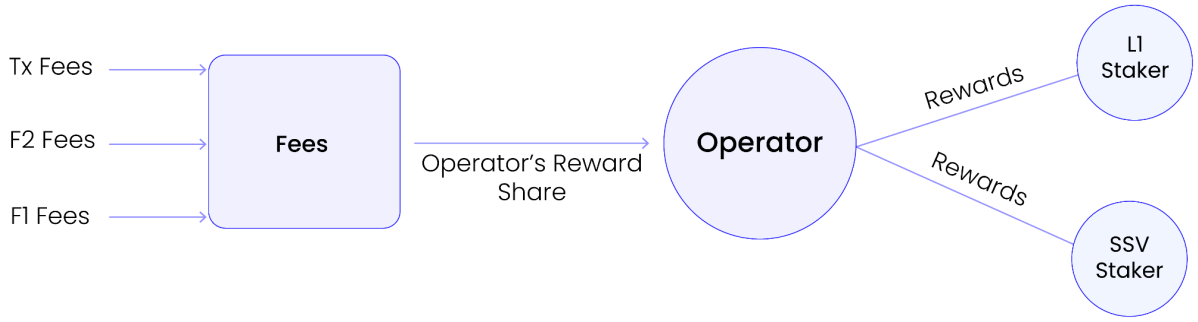

Figure 7: SSV 2.0 rewards flow

$SSV Deflation

The deflation of $SSV is controlled by two aspects, minting and burning.

Minting

The minting of each and every SSV token is controlled solely by the SSV DAO's Multisig committee and is subject to approval by the SSV DAO.
Currently, a substantial part of the minting of SSV comes from the Incentivized Mainnet(IM) program, which, since October 2023 has minted more than 600.000 SSV. Similar to user acquisition, the goal of the IM program is to accelerate the adoption of DVT, which it has been doing successfully.

The re-utilization of L1 validators is a long-term incentive to continue transitioning to DVT by introducing additional incentives for validators. Successfully adopting bApps will enable the decrease of IM rewards to 0 (at some point in the future) while enabling growth with these incentives coming from bApps. Below is a suggested, possible rate of decrease:

	2025	2026	2027	2028	2029	2030	2031
IM reward	1M	640K	410K	135K	45K	6K	0
Supply	13M	13.64M	14.05M	14.185M	14.23M	14.3M	14.3M
Inflation	8.3%	4.9%	3%	0.93%	0.3%	0.04%	0%

Table 6: This is ONLY a potential IM schedule for the sake of deflationary calculations below

Burning

A portion of collected fees will proposed to be burned; that portion is a function of the amount of SSV staked in the protocol: $B = b_{max} * (1 - \epsilon^{-\gamma * s})$ where:

- B - The amount burned
- γ - adjustable hyperparameter
- s - staked SSV.

As a result of burning fees and decreasing IM rewards, the net inflation of SSV should decrease in the upcoming years.

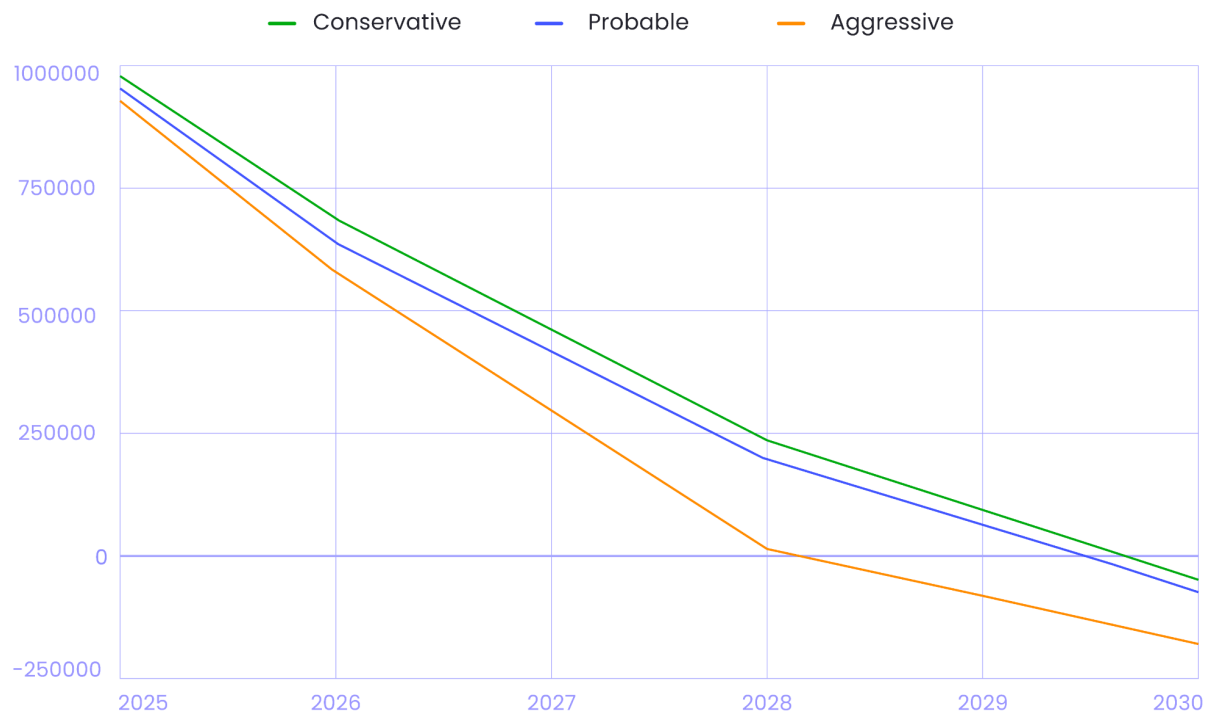

Figure 8: potential SSV token total inflation, including burning (requires DAO approvement)

The chart gives possible examples of how net issuance may change. The conservative line assumes a smaller percentage of staked SSV tokens and fee growth, while the aggressive line considers a larger one. Under these assumptions, we can see that the aggressive scenario will make SSV deflationary before the end of 2027, while the probable scenario will do the same in 2028 and the conservative one in 2029.

Conclusion

The Based Applications Protocol introduces a novel approach to addressing the bootstrap problem faced by proof-of-stake systems. Enabling L1 validators to secure multiple applications without additional capital requirements creates a more capital-efficient ecosystem that benefits both validators and applications.

This formalized security is achieved through several key mechanisms:

- Risk Expressive Model (REM) - Allows operators to participate in multiple bApps while carefully managing their risk exposure through a sophisticated scoring system that adjusts voting power based on obligations.
- No Additional Staking Requirements - L1 validators can secure bApps without locking up additional ETH or taking on cascading slashing risks, unlike traditional restaking approaches.
- Flexible Participation - The protocol enables validators to opt-in to applications based on their risk tolerance and desired level of involvement, creating a more dynamic and accessible ecosystem.

The protocol also addresses key challenges of existing approaches:

- It eliminates the need for applications to gather massive validator sets independently.

- It avoids the cascading risks and high costs associated with traditional restaking.
- It provides a deterministic coordination layer that ensures reliable validator participation.

SSV 2.0 creates a sustainable foundation for proof-of-stake applications to bootstrap their security by combining these elements and introducing a deflationary token model. The protocol's capital-efficient design and risk-aware approach make it an attractive solution for both established validators looking to expand their operations and new applications seeking to build robust validator sets.

As the ecosystem matures and more applications adopt this model, the increased capital efficiency and reduced barriers to entry should foster greater innovation and growth in the proof-of-stake space, maintaining strong security guarantees through carefully designed cryptoeconomic mechanisms.

References

1. Deb, A., Kannan, K., & Tse, D. (2024). Stakesure: Cryptoeconomic Security Analysis of Proof-of-Stake Protocols. arXiv preprint arXiv:2401.05797.
2. Buterin, V., & Griffith, V. (2017). Casper the Friendly Finality Gadget. arXiv preprint arXiv:1710.09437.
3. Buchman, E., Kwon, J., & Milosevic, Z. (2018). The latest gossip on BFT consensus. arXiv preprint arXiv:1807.04938.
4. Buterin, V. (2020). Why Proof of Stake. Vitalik Buterin's website.
5. Ethereum Foundation. (2022). The Merge. ethereum.org.
6. Eigenlayer. (2023). Dual Staking: secure a PoS network with two tokens. blog.eigenlayer.xyz
7. Eigenlayer. (2024). Introducing the EigenLayer Security Model: A Novel Approach to Operating and Securing Decentralized Services. blog.eigenlayer.xyz
8. Drake, J. (2023). Based Rollups: Superpowers from L1 Sequencing. Ethereum Research Forum.
9. Ethereum Foundation. (accessed 2024). Oracles. ethereum.org.
10. Ethereum Foundation. (accessed 2024). Bridges. ethereum.org.
11. Al-Bassam, M., Sonnino, A., & Buterin, V. (2018). Fraud and Data Availability Proofs: Maximising Light Client Security and Scaling Blockchains with Dishonest Majorities. arXiv preprint arXiv:1809.09044.
12. Drake, J. (2024). Based Preconfirmations. Ethereum Research Forum.
13. Chiplunkar A., Neuder M. (2023). Optimistic Relays and Where to Find Them. frontier.tech/optimistic-relays-and-where-to-find-them.
14. Ethereum Foundation. (accessed 2024). Decentralized Autonomous Organizations (DAOs). ethereum.org.
15. Lido. (accessed 2024). Lido: Liquid Staking Protocol Documentation. docs.lido.fi.
16. Ultrasound Money. (accessed 2024). Ethereum Supply Analytics. ultrasound.money.

Appendix

A1 - Risk Expressive Model - Numerical Example

Let's consider a bApp that defined $\beta = 2$ and has 3 participants with obligations $o_1 = 10$, $o_2 = 20$, and $o_3 = 30$, and with risks $r_1 = 1\%$, $r_2 = 100\%$, and $r_3 = 200\%$. The total obligation in the bApp is $10 + 20 + 30 = 60$, and, thus, $p_1 = 1/6$, $p_2 = 2/6$, and $p_3 = 3/6$.

First, we compute the normalization factor c:

$$\begin{aligned} c &= (1/6 \times e^{-\beta \times max(1,0.01)} + 2/6 \times e^{-\beta \times max(1,1)} + 3/6 \times e^{-\beta \times max(1,2)})^{-1} \\ &= (1/6 \times e^{-2 \times 1} + 2/6 \times e^{-2 \times 1} + 3/6 \times e^{-2 \times 2})^{-1} \\ &\approx 13.02 \end{aligned}$$

Then, we can compute the weight for each participant:

$$W_1 = c \times 1/6 \times e^{-\beta \times max(1,0.01)} = 13.02 \times 1/6 \times e^{-2} \approx 29.4\%$$

$$W_2 = c \times 2/6 \times e^{-\beta \times max(1,1)} = 13.02 \times 2/6 \times e^{-2} \approx 58.7\%$$

$$W_3 = c \times 3/6 \times e^{-\beta \times max(1,2)} = 13.02 \times 3/6 \times e^{-4} \approx 11.9\%$$

Note that, even though account 3 has $p_3 = 3/6 = 50\%$, its higher risk dropped its weight to 11.9%.