EigenCloud: Build Powerful Crypto Apps on Any Chain with the Verifiable Cloud

Unleashing the Verifiable Economy

Eigen Labs

1. The Problem: Crypto is verifiable but not fully programmable

Societies thrive when innovation systems, such as free markets, are supported by coordination enforcement systems, such as representative governments. While innovation has raced ahead, our coordination tools (for example, money, double-entry accounting, common law, governance) are rooted in the past. Crypto technologies offer a solution, enabling digital-native coordination systems that are autonomous, enforced by code, and verifiable by anyone. The long-term opportunity for crypto is powering a new global, digital-native, verifiable economy with cheaper, fairer, and more accountable commerce, shifting human effort from solving coordination bottlenecks to creating together.

1.1 Crypto's superpower is verifiability:

Blockchains are unique in their ability to verify and enforce that computations have been performed correctly. Over the years, developers have disrupted entire industries with new classes of applications using this unique capability. We can divide the progression of blockchain applications into three eras:

(1) **Verifiable Money**: Bitcoin made it verifiably certain that there will only ever be 21 million BTC, and the entity holding the private key will control its Bitcoin. These modest commitments made by a single, verifiable application helped create a $2 trillion asset that disrupted money.

(2) **Verifiable Finance**: By generalizing Bitcoin's transaction model, Ethereum expanded crypto programmability with its Turing-complete EVM. Within a decade, this innovation attracted tens of thousands of developers. It kickstarted an era of decentralized finance (DeFi) applications (exchanges, lending, money transfer, prediction markets, stablecoins), which thrived on verifiability as an antidote to counterparty risk. Verifiable lending apps used smart contracts to manage collateral, govern borrowing and lending parameters, and automate adjudication to protect lenders. These teams addressed the key friction in traditional lending - that a borrower will not return assets loaned to them. As these methods reduce the risk of borrower insolvency, the addressable market of verifiable lending increases. This era is just getting started. Ethereum and Solana boast fewer than 25,000 developers, but together, they have grown to more than $500 billion ecosystems disrupting traditional finance.

(3) **Verifiable Economy**: We believe the next era will be the largest. Once we lower the barrier for developers to build powerful verifiable applications, new verifiable applications can disrupt existing marketplaces, address existential risks, and create aligned incentive structures for fundamental societal problems.

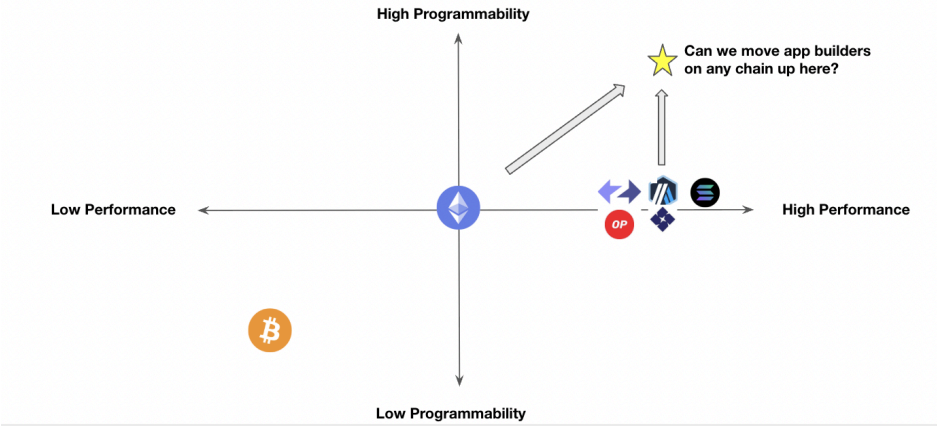
(3a) **Disrupt digital platforms**: Digital marketplaces such as Amazon, Uber, and Airbnb provide value by aggregating the various sides of the marketplace. They also create massive points of centralized, unverifiable control. Blockchains provide a mechanism for any business or consumer application to add verifiability, automated enforcement, user ownership, permissionless access, and immutability to these types of marketplaces.

(3b) **Build trust in the AI era**: As we enter the AI era, it is essential to ensure the outputs of new digital intermediaries, such as centralized AI models and agents, are correct and unbiased. Furthermore, there is no legal infrastructure for AI agents to have property rights or the ability to enter into contracts or accept liability. Blockchains, via the substrate of smart contracts, can ensure that AI agents have digital-native expression with respect to the disposition of property, contracts, and slashing-adjudicated liability.

(3c) **Solve societal-scale problems**: Blockchains' verifiability can also power novel solutions for long-standing societal problems: education (credentials, income shares), financial credit (verifiable credit scores), healthcare (pay-for-outcomes, reputation systems), media (verified sources), and climate initiatives (carbon credits and offsets).

Catalyst	Category	New verifiable guarantee	Design space	Market value
Bitcoin	Cryptocurrency	Money with 21M hard-cap and self-custody	Monetary store-of-value	2T$+ for BTC (May 2025)
Ethereum	Smart contract blockchains	Verifiable execution of deterministic state transition of any program.	Decentralised finance (DEXs, lending, stablecoins, etc.)	1T$+ across ETH + L1s + L2s + tokens created on these platforms (May 2025)
EigenCloud	Verifiable cloud	Verifiable execution of *any* computation, data, or	Disrupt digital platforms Build trust in the AI era	Cloud created 10T$+ of value by making the economy *programmable*.

	commitment that two rational parties can agree on	Solve societal-scale problems	Verifiable cloud will create comparable value by making the economy *verifiable*.



1.2 Crypto's primary constraint is programmability:

Since Ethereum's introduction, blockchain application development has been bound by two structural constraints: (1) scalability and (2) programmability. Enormous research and engineering effort over the past decade has alleviated the first constraint: High-throughput Layer-1 (L1) networks, Layer-2 (L2) rollups and modular stacks, data-availability layers, and succinct-proof systems now provide orders-of-magnitude increases in transaction capacity. However, addressing scalability has not been enough to unlock broad application development or new, non-financial categories of apps and use cases.

Public clouds have set the bar for what "programmable infrastructure" means: they can select any software library, provision specialized hardware, stream data through standard interfaces, and assemble production systems from managed services instead of rebuilding software infrastructure like relational SQL databases. Blockchain architectures, on the other hand, place severe restrictions on various dimensions, making development quite complex.

(a) **Software Limitations:** Developers cannot directly leverage the richness of off-chain open source software libraries, and existing virtual machines (EVM, WASM, zkVMs) require developers to re-implement software. This forces builders into constrained environments and restricts the developer base to the smaller cohort of crypto developers (~20,000) rather than the 20M+ software developers in the world.

(b) **Hardware Limitations**: The blockchain specifies the hardware on which the code is run. Developers cannot request that the blockchain provision specialized resources such asGPUs, high-performance CPUs, trusted execution environments (TEEs), geo-distributed caches, or watchtowers.

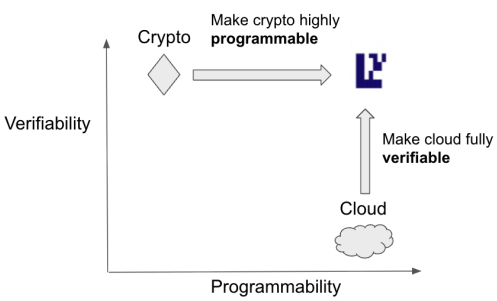
(c) **Interface Limitations**: Blockchains can only trust data that is already on-chain or provided via oracles secured separately from the blockchain. Direct verifiable access to external APIs and real-world state with strong security guarantees remains an unsolved challenge.

(d) **Consensus limitations**: Blockchains have pre-determined consensus protocols that decide how validator nodes partition work, communicate, and arrive at consensus. These protocols do not allow developers any programmability of these dimensions. This additional programmability is required to build new features such asdata availability (DA), faster finality layers, privacy, multiparty computation, sharded data stores, AI inference protocols, etc.

(e) **Objectivity limitations:** Blockchains can only utilize contracts and agreements on purely objective (or binary) conditions. This massively restricts the categories of contracts that can be moved onchain, as most real-world contracts rely on human-interpretable conditions.

These limitations exist due to fundamental constraints in blockchain technologies. To guarantee tight determinism (i.e., any two nodes will get the same result), blockchains only execute software on a native Virtual Machine (like EVM for Ethereum), and this restricts the (a) software that can be executed, (c) interfaces accessible, and (e) enforce pure objectivity. Furthermore, blockchains are built to operate on consensus of redundant execution, which imposes constraints on the types of (b) hardware accessible and (d) enforces pre-determined consensus protocols.

2. The Promise: Make crypto more programmable to unleash the verifiable economy.



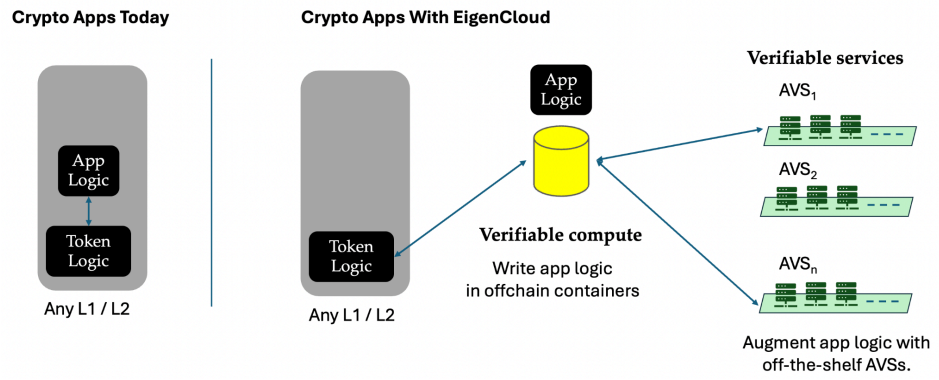
EigenCloud aims to empower developers with a platform that simultaneously offers cloud-grade programmability while maintaining crypto-grade verifiability. EigenCloud enables developers to move expressive application logic into a verifiable off-chain container, while keeping only minimal consensus-enforced logic on-chain and the infrastructure that doesn't need to be verifiable remaining in public clouds (like front ends).

We propose a new application architecture that combines the cloud's programmability with blockchains' verifiability by moving app logic out of smart contracts and into EigenCloud. We factorize the application architecture into two parts:

(1) Token logic. Chains are uniquely suited for token-related logic, i.e., storing tokens, escrowing tokens, and transferring ownership of tokens based on various conditional predicates. Blockchain VMs provide a fully-ordered ledger and an ownership semantic for state that enables these functions really well.

(2) App logic. App logic is the conditional predicate for the token logic. Today, they are built inside the blockchain VM as well. The constraint of fitting the app logic onchain limits the categories of conditions that can be expressed. Thus, while checking for under-collateralization in a lending market is a great fit for blockchain VMs, it is very difficult to build a crypto trading app that moves tokens only when a specific AI model signs off on it. We propose to move this logic into a verifiable off-chain container, and this container can be secured with the desired amount of economic security, with the guarantee that if the container execution is corrupted, the app can slash and redistribute the funds to the app users. EigenCloud also allows for app logic to be expressed using a composition of off-the-shelf verifiable services in addition to building a custom container. Furthermore, EigenCloud will allow for various trust models beyond economic security, including optimistic rollup-grade security, ZK, honest-majority, and TEE, to secure the execution of these verifiable containers.

While the app logic and token logic are executed securely in this architecture, similar to today, portions of the app that do not require this grade of verifiability will continue to run on the public cloud: front ends, analytics, and other aspects of the app that do not require users to trust.



By moving important aspects of logic into the verifiable cloud, developers are freed to build more performant and sophisticated applications that, today, would be difficult or infeasible to implement fully on-chain. EigenCloud thereby dramatically expands the application design space, enabling new classes of verifiable software services to emerge with rich, more powerful logic than is possible with on-chain VMs alone.

Similar to the cloud, which has an array of first and third-party IaaS, PaaS, and SaaS services "on the shelf," EigenCloud has a growing library of first and third-party AVSs available. AVSs are autonomous verifiable services built on top of EigenLayer, which enable powerful application logic to be built without having to do undifferentiated heavy-lifting of building services that are not unique to the application at hand.

EigenCloud enables verifiable apps and agents

Here we give some examples of applications that the EigenCloud uniquely powers:.

App Category	Brief Description	Onchain token logic	Verifiable offchain logic
Verifiable Data Onramp	All data which are inside Web2 walled gardens becomes provable and verifiable	Onchain actions	zkTLS for data import
Verifiable Incentivization for Social Media	If social media incentives are verifiable, they can be incorporated into programmatic inflation. E.g., memecoins that can do fair launch or programmatic incentives for social media.	Token pool reserved for incentives	zkTLS for data import AI for analyzing posts
Verifiable Credit Score	Credit score being verifiable is necessary to lend against it.	Lending pools	zkTLS for data import AI / Compute for data analysis
Verifiable Debate Markets	Can build prediction markets that settle against verifiable judgments.	Prediction market	AI for judging debates DA for writing debate data

Verifiable AI Governance	Governance is a strong trust dependence for protocols/tokens.	Community Rewards + Slashing	AI for evaluating governance actions
Verifiable Preference Aggregation	Governance preferences are aggregated across users.	Governance incentives	DA for storing user preferences Compute / AI for aggregating preferences
Verifiable Community Notes	Community feedback is integrated into concrete onchain consequences like slashing.	Slashing logic for repeated offenders	DA for storing community votes Compute / AI for aggregating communities' notes verifiably
Verifiable DeFi Vaults	The offchain logic is security-critical as it can move users' tokens around.	Hold, transact, escrow tokens	Computation of complex vault logic AI for augmenting intelligence Oracles for data import
Verifiable Pricing Covenants (e.g., onchain Uber, Taskrabbit, Fiverr, Instacart, Airbnb)	Verifiability ensures that take-rate covenants are honored across upgrades. For example, onchain ride-sharing service.	Smart contract upgrade functionality	Intersubjective / AI oracle certifying upgrades satisfy covenants
Verifiable Upgrade Covenants (e.g., onchain Facebook)	For example, any upgrade to onchain Facebook that changes the privacy policy needs a majority of users to agree to the change.	Smart contract upgrade functionality	Intersubjective / AI oracle certifying upgrades satisfy covenants
Verifiable Media	Verifiability can solve fake news and "sensationalist" article titles. Once recorded onchain, anyone can read it without needing to re-verify.	Verified media registry	Sensor integrity from TEE inside cameras Location verified via AVS AI / computation for processing data into articles DA for storing the output media (articles, feeds, etc.).
Verifiable Feed Curation	Verifiability that the feed was curated by an AI with certain preferences, protecting users against adversarial curation.	Verified feed registry	Marketplace of verifiable AI for feed curation.
Verifiable Casinos	Massive trust assumption in online casinos, broken many times.	Token escrow and payment	Verifiable Randomness Private global state Complex computations
Insured AI Agents (e.g., ticket booking)	AI agents are not trustworthy. Solved by agents with payout for erroneous actions.	Money staked and distributed	Verifiable agent execution Verifiable adjudication of errors
Verifiable Medical Insurance	Verifiable inference can solve low trust in medical insurance claim adjudication.	Insurance collateral, payments	Verifiable adjudication of claims using AI
Verifiable Databases	Enterprise blockchain use cases with an easy migration path.	Tokens / payments	Verifiable databases of various types (SQL, noSQL)
Verifiable Datalake	Datalakes made verifiable.	Tokens / payments	zkTLS for Verifiable data import Compute for Verifiable data processing Compute for Verifiable queries
Verifiable Supply	Supply chain for commodities such as food,	Registry,	zkTLS for verifiable data import

Chain (e.g., food)	diamonds.	payments	TEE for sensor integrity Verifiable location Verifiable Datalakes
Permissionless Prediction Markets	For example, a market can resolve based on a Twitter account tweeting / announcing something.	Prediction market	AI-based resolution of markets zkTLS for data import
Verifiable Location-Based Actions (e.g., Pokemon Go, other AR games)	Bounties are paid to those who verifiably took some location-based actions.	Game bounties	Verifiable location Verifiable sensor integrity Verifiable game execution Verifiable privacy
Verifiable Climate Credits	Solve correct assessment of climate credits, eliminate double counting, enable global trading.	Climate credits registry and trading	Satellite images (forest cover, etc.) made verifiable via TEE sensors. AI and computational processing to translate images into credits is verifiable
Games with Prediction Markets	Enable prediction markets to be built trustlessly.	Game outcomes, prediction markets	Verifiable game execution
Verifiable AI Agents (e.g., trading, investment)	Only if all AI agent dependencies, including tool calls, are verifiable is the AI itself verifiable.	Money stored in a smart contract	Verifiable AI Verifiable database Verifiable privacy Verifiable API access (zkTLS) Verifiable tool calls (computation, etc.)
Verifiable DeSci	Verifiability of the scientist implies that every dollar is used for the cause it supports, ensuring unparalleled transparency and trust needed for crowdfunding. Can also verify IP creation provenance through this method.	Crowdfunding, IP registry	Verifiable AI that produces scientific experiments. zkTLS to verify experiment outcomes via APIs (for example, using cloud labs for bio experiments)
Sovereign AI Agents Funded Via Wealth Shares	Verifiability ensures “integrity of the AI” as well as sovereignty - otherwise some other model could trigger onchain actions and steal the AI’s money. This ensures that the AI agent is fundable via wealth shares, so anyone can now fund the AI agent to receive a portion of its upside.	AI agent stores money in a smart contract, funds itself initially by issuing wealth shares	Verifiable execution of the specific model and dependencies. Verifiable privacy
Transaction Pre-Screening for Security	Protocols get hacked regularly due to lack of transaction pre-screening. Using AI and other complex computation to pre-screen transactions is a huge step forward but needs verifiability to ensure transactions are not censored arbitrarily (and thus lose any benefit of onchain).	Any chain	Verifiable AI for security screening

3. The Product: EigenCloud

We have built EigenCloud to augment any blockchain, introducing new cloud services for builders to launch full-stack crypto apps as easily as building on the cloud, but settled with blockchain-grade trust. EigenCloud simply hosts your offchain logic as an offchain container, proves its results onchain, and charges you for the security actually used. For application developers, EigenCloud provides the following value propositions:

(1) **Build rich, powerful applications:** Deploy containers (e.g., Docker/Nomad/Kubernetes) and use EigenCloud to prove results to an onchain smart contract, so you get cryptoeconomic guarantees for application logic without needing to write blockchain code.

(2) **Use on any chain:** Verified outputs trigger state changes on smart contracts on any L1 or L2, letting you blend cloud-grade developer experience with crypto-grade settlement.

(3) **Consume and compose existing AVSs:** Use off-the-shelf AVS (autonomous verifiable services) in a composable manner with standardized consumption, payment, and trust models.

(4) **Pay only for what you consume:** Choose exactly how much economic security backs each service and pay only for the security you consume. Routine reads can cost pennies, and liquidation engines can get billions in collateral.

3.1 EigenCloud offers cloud-scale programmability with crypto-grade verifiability.

EigenCloud is a suite of developer-focused tools and services to rapidly build, deploy, and scale highly expressive applications that interact verifiably with any blockchain. The core idea of EigenCloud is to augment application logic in smart contracts using offchain computation/ services and ensure that the execution of this container remains verifiable. In order to realize a frictionless experience for developers to build app logic on EigenCloud, we are building three modular primitives on top of EigenLayer that solve specific developer problems.

	What it does	Features	Target audience	Why	When
EigenLayer	Maintain cryptoeconomic guarantees for verifiable services (AVS).	Unique and pooled security, customizable rewards, slashing, and redistribution.	AVSs	Make it possible	Protocol complete. Redistribution on mainnet by Q3.
EigenDA	Ensure that AVS or rollup data remains available for anyone to download.	Hyperscale (50 MB/s), fixed pricing model, DA sampling.	Rollups, AVSs	Make it fast and cheap	Protocol live. V2 brings throughput to 50MB/s by Q2.
EigenVerify	Verify that the AVS or rollup operators ran the computation correctly.	Objective verification via re-execution and intersubjective verification.	Rollups, AVSs	Bring rich verifiability	Devnet live. Mainnet by Q3 2025.
EigenCompute	Abstract all of the above components for an app developer.	Use any cloud container. Automated deployment. Use on any chain. Configure the amount of security needed. Compose AVSs.	Apps	Make it easy	Devnet by Q3. Mainnet by Q4 2025.

3.2 EigenLayer is the foundation of EigenCloud

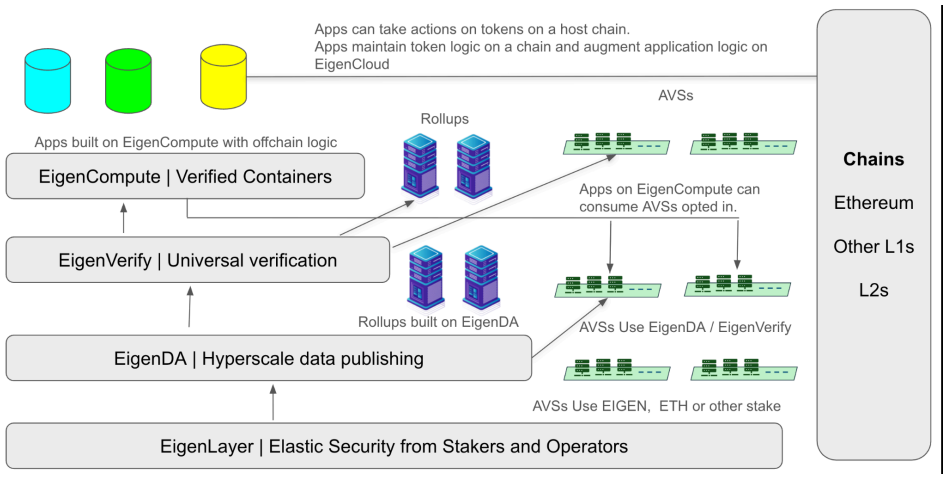
EigenLayer made it possible to get cloud-scale programmability at crypto-grade verifiability by allowing a developer to build offchain services that can be made verifiable. Verifiability in proof-of-stake blockchain networks comes from cryptoeconomic security - validators stake tokens and operate nodes to verify network activity, facing financial penalties (slashing) if they misbehave. With EigenLayer, the cost of bootstrapping economic security is massively reduced, allowing a variety of services to be built. We call these services AVS (autonomous verifiable services) - *autonomous* because anyone can stake and continue running the service (akin to Ethereum) and *verifiable* because the correct operations of these services are enforced by onchain slashing contracts.

Today, EigenLayer is feature-complete and is the largest protocol for shared security with over $10 billion in stake, 2,000 registered operators, and hundreds of AVSs live or in production spanning dozens of distinct categories. Many AVSs express extremely complex infrastructure, and several are as complex as building a new kind of Layer-1 protocol. With critical features like Rewards, Slashing, and Programmatic Incentives launched and live on mainnet, and features like Redistribution going live soon, EigenLayer is a fully featured open-source infrastructure for developers building verifiable services. To take the next step and unlock powerful crypto apps, Eigen Labs will focus its efforts on building expressive offchain services on top of EigenLayer.

3.3 The Eigen Primitives expand programmability and verifiability.

While easier than launching your own L1, it is still complex to build an AVS, which means only teams focusing on building general-purpose infrastructure primitives could build these AVSs. EigenCloud aspires to make it possible for application builders to build cloud-scale applications with crypto-grade verifiability. However, to build an AVS, developers must contend with throughput limitations, the high cost of data publishing on Ethereum, the need to write slashing logic onchain, and the overall complexity of building on top of EigenLayer. Applications can instead choose to utilize an existing AVS if a comparable service already exists. However, consuming existing AVSs in the EigenLayer ecosystem has frictions in discovery, a lack of standardization (across payments, trust models, and supported chains), and composition (when apps need multiple AVSs).

3.3.1 EigenDA is foundational to EigenCloud because it provides a hyperscale, economically secured data availability layer essential for verifying off-chain computations. Without EigenDA, verifiable containers and fraud proofs wouldn't reliably publish their inputs and outputs for independent verification. By offering guaranteed, transparent, and low-cost storage of cryptographic proofs and off-chain data blobs, EigenDA ensures the integrity of EigenCloud's entire trust model, enabling complex applications to inherit blockchain-level trust without sacrificing cloud-like flexibility.



3.3.2 EigenVerify is a set of plug-and-play services that can verify whether an AVS / rollup operator ran a computation correctly. Implementing robust, custom fraud proofs has historically been a complex and resource-intensive task for AVS developers. Currently, only limited forms of dispute resolution, such as double-signing violations or simple ZK proof verification (verifier-as-a-service), have been feasible to implement easily. However, developing expressive on-chain fraud-proof contracts for sophisticated workloads, such as AI inference, large-scale database computations, or custom rollup VMs, is prohibitively difficult or requires significant development time and budget. Additionally, certain faults, like oracle disputes and liveness faults, simply cannot be objectively verified on-chain, making them inherently difficult to adjudicate. EigenVerify addresses these challenges by providing ready-to-use primitives for verifying whether an AVS ran a computation correctly. Whenever a suspected erroneous AVS or rollup execution is detected, anyone can trigger a call to EigenVerify to resolve the dispute. EigenVerify supports two modes with strong security guarantees:

(1) *Objective verification* of whether deterministic code was executed correctly by AVS operators by re-execution of the code.

(2) *Intersubjective verification,* where any task for which any two humans can agree on the outcome can be verified. (Resolution of a prediction market - for example, who won the election.)

EigenVerify is run by an operator set staking their EIGEN tokens across EigenVerify and EigenDA, and requires a majority consensus to respond to a verification request. In case a majority of stake-weighted operators running EigenVerify collude to produce a faulty response, the EIGEN token will be forked to punish the malicious stakers and operators. This ensures that the EigenVerify service remains reliable. The EIGEN token forking is a deterrent for the majority of EIGEN stakers to collude. It imposes a nuclear penalty by burning all their EIGEN tokens and incentivizing the one who initiated the fork appropriately. We point the reader here for details about how the EIGEN token forking works.

3.3.3 EigenCompute provides an abstraction layer for app developers so that the app developer can build using offchain verifiable

computation and autonomous verifiable services (AVSs) easily. EigenCompute leverages EigenLayer, EigenDA, or EigenVerify under the hood to accomplish this objective. It specifically performs two different tasks: verifiable offchain computation

Verifiable Offchain Computation

1. **Containerization and deployment on EigenCompute:** A developer creates a container housing application logic to extend token logic on any blockchain. This container is then deployed to EigenCompute, which simplifies identifying suitable stakers and operators.

2. **Computation secured by EigenLayer:** The computation is run using an EigenLayer operator set and secured by stakers uniquely allocating a portion of their security for this app. The resulting output, accompanied by signed certificates, is delivered to the intended blockchain. The on-chain contract may utilize these results immediately, with unique economic security, or wait until a challenge period has concluded for enhanced security (akin to optimistic rollups).

3. **Data Availability via EigenDA:** The computation data is recorded to Data Availability (DA), allowing any observing node to independently verify the computation using EigenDA.

4. **Challenge Validation via EigenVerify:** If a challenge is initiated, the original computation undergoes validation via EigenVerify, a service designed to re-execute computations for verification purposes. If the initial group of nodes is deemed malicious, they are penalized (slashed), and the forfeited stake is distributed to the application.

5. **EigenDA and EigenVerify secured via Token Forking:** If a majority of the EIGEN stakers securing either EigenDA or EigenVerify act maliciously, corrupting the process, the EIGEN token forks. This action slashes the malicious stakers' and operators' stake, ensuring system integrity without relying solely on the trustworthiness of a majority committee. Anyone can initiate an EIGEN token fork by burning a certain percentage of the original EIGEN token and getting rewarded with a larger fraction of the forked EIGEN token; thus ensuring that there are sufficient incentives for triggering the fork.

6. **Application / chain forking for additional security:** Applications, services, or chains built on EigenCloud have the option to pre-commit and choose to fork in alignment with the EIGEN token. This ensures that, on the correct fork, the results of the application and services maintain full integrity.

Determinism handled by apps rather than the chain: Application developers only get these security guarantees if the application logic is deterministic. In existing blockchains, the correctness or matching of application logic to the intended protocol is left to the developer. In a similar spirit, EigenCompute also extends the responsibility of determinism to the developer. We are able to do this because we ensure, through using EigenVerify, that operators can safely run any container and will not get slashed erroneously because of non-determinism errors.

Security model: Depending on the way the output from EigenCompute is utilized, we can get different security models:

A. Instantaneous consumption of computation -> Insured economic security.
B. Lagged consumption of computation results -> Rollup grade security (assuming Eigen majority stake is honest).
C. Lagged consumption with commitment to fork application/chain if EIGEN token forks -> Rollup grade security.

Verifiable Service composition: Apps not only want to use EigenCompute to run offchain components securely, but they also may want to utilize pre-existing AVSs in the EigenLayer ecosystem that have opted in. These AVSs that opt in will have standard consumption patterns, common payment mechanisms, and graded trust models so that app users can consume these AVSs with ease. Furthermore, as we develop the EigenCloud, there will be opportunities to offer fast composition of multiple AVSs, as well as bundled pricing for apps using multiple AVSs. These mechanisms to distribute and charge for AVSs to be consumed by apps create a new shared distribution flywheel, by which AVSs can find easy distribution to Apps. The core mechanism behind EigenLayer was the shared security flywheel, where more stake produced more security, which attracted more AVSs, which paid more rewards back to the stakers. EigenCompute provides new opportunities for apps and agents to consume, compose, and use multiple AVSs, creating a new shared distribution flywheel. In this flywheel, as demand from apps for verifiable services increases, more AVSs show up to distribute to these apps, and the more AVSs that exist, the more verifiable services can be sold to apps.

Why would developers choose to utilize off-the-shelf AVSs rather than relying purely on custom-built containers?

1. **Ease of development and maintenance**: Using an AVS to handle a portion of the application logic is similar to using SaaS services or off-the-shelf APIs, bringing advantages of rapid development, trusted code, and code maintenance.

2. **Different trust models**: Some applications may need a majority-honest trust model, TEE-network-based trust, or other trust models that are not possible in EigenCompute.

3. **Variety of operator nodes**: Applications may require specific kinds of operator nodes, redundancy, geo-distribution of operators, etc., that are not possible in EigenCompute.

4. **AVSs can statistically multiplex compute across Apps**: In EigenCompute, a specific amount of node is provisioned for the specific container. This can be expensive when computation happens only in spurts. Using AVSs that specialize in the particular type of hardware, it is easy for AVSs to charge per query rather than needing to pay persistent fees when using an offchain container.

5. **Determinism handled by AVS**: When consuming an AVS, the AVS may have its own mechanisms to ensure determinism, thus lowering the barrier for app developers.

4. The Token: EIGEN is at the center of the EigenCloud

In the EIGEN token whitepaper, we introduced the EIGEN token as a mechanism to secure a broad range of digital tasks that previously could not be verified in conventional blockchain systems (intersubjective tasks). The key idea underpinning the EIGEN token is forkability: if the majority of the EIGEN token stakers do not perform the task correctly, then the token can be forked and the malicious token stakers slashed. The forkability of the token acts as a deterrent to the malicious behavior of the token holders and keeps the system honest, similar to an L1 token's role today. When the system functions as expected, then forks will be exceedingly rare.

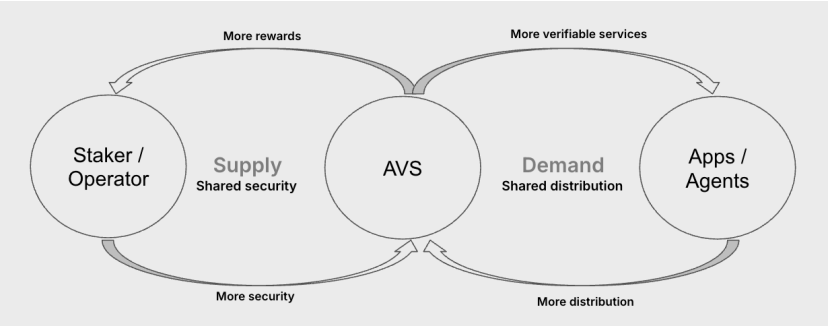
4.1 How EIGEN powers verifiable offchain compute

In the EigenCloud, we use the EIGEN token to ensure the verifiability of our new offchain services, which are designed to permit a fork of the EIGEN token if there is a majority malicious fault on EigenDA or EigenVerify. For EigenCompute, since it uses EigenDA and EigenVerify under the hood, it inherits the security from these services.

We can do this because of the EIGEN token's novel token design, which decouples the general-purpose utility of the unstaked ERC-20 EIGEN token from its forking and social-accountability properties enabled by its staked representation, the bEIGEN token. The EIGEN token functions as a transferable, fungible digital asset that can be used in DeFi applications or held with custody providers. EIGEN tokenholders do not have to participate in, or even be aware of, any forking consensus or controversies. By contrast, the bEIGEN functions as an opt-in representation of staked EIGEN, carrying slashing risks associated with digital work and subject to forking in the event of a fault caused by the dishonest majority.

This functional separation takes on special importance when bEIGEN is unstaked. The bEIGEN reflects a voluntary commitment to secure protocols and applications. When the bEIGEN is unstaked, however, it unwraps into the EIGEN, which, in essence, inherits the fork accepted as canonical by tokenholders and ratified by protocols and applications across the ecosystem. This ensures that the economic alignment between bEIGEN and EIGEN is preserved on the socially recognized fork. The bEIGEN therefore, is both fork-responsive and economically tied to the EIGEN. At the same time, the EIGEN remains a familiar, transferable ERC-20 asset, requiring little or no knowledge of these governance mechanics by the holder.

4.2 The EigenCloud Flywheel and EIGEN value accrual



EigenLayer generalized traditional PoS systems by open-sourcing the foundations of crypto-economic security. Since its inception, it has served as a coordination layer connecting stakers (or their designated operators) with protocols and applications requiring the crypto-economic security assurances provided by active digital work. Applications are built on top of the autonomous verifiable services (AVSs) deployed on EigenLayer. In turn, these AVSs p reward stakers and operators that opt-in to perform computational labor necessary to secure the protocols and applications. With EigenCloud, the flywheel is supercharged by a) expanding the types of applications that can be secured by a blockchain (EigenVerify) and b) making it easier to harness security directly and via AVSs (EigenCompute). In conjunction with these new primitives, the EIGEN token

ensures the eventual verifiability of any task for which any two reasonable humans will agree on its correct execution. As more powerful AVSs get built, novel applications can build on them, and as novel applications are built or imagined, new and innovative AVSs will compete for their business. In this figure, there is no mention of users - this is intentional as we believe that there is an Apps-to-Users flywheel that is outside the scope of EigenCloud (similar to the public cloud, which users do not directly interact with). This is where we will partner with L1s / L2s as well as super-apps which aggregate users across many apps, and perform a role complementary to EigenCloud. While EigenCloud provides tools for building powerful apps, these partners provide apps with user distribution.

This universal verifiability massively expands the design space for app developers and has several important implications for EIGEN:

The EigenEconomy - The EigenCloud uses the EIGEN token to secure its core AVSs – EigenDA, EigenVerify, and EigenCompute – as well as the applications and services built on top of them. EIGEN tokens can be staked to support these services and receive compensatory rewards, similar to how L1 tokens are staked to secure traditional blockchain networks.

In the future, EigenCloud primitives may be subject to protocol-level fees designed to support the long-term sustainability of the Eigen ecosystems. If adopted, these fees would be ringfenced for purposes such as staking rewards, protocol incentives, research and development grants, and other public goods. The protocol-level fees could be applied to two EigenCloud value flows:

1. *EigenCloud AVSs to EIGEN Stakers*: EigenCloud AVSs provide compensatory rewards to stakers and operators who perform the computational labor necessary to secure their services.

2. *Applications to EigenCloud AVSs*: EigenCloud AVS rewards may be sourced from the applications that build on them, which may pay services-related fees to the specific AVSs they utilize. Thus, applications may pay fees to primitives, such as EigenCompute, for example, which abstracts away infrastructure complexity and facilitates integrations across multiple AVSs. These application-layer fees would be distinct from the EigenCloud AVS rewards paid to stakers.

In either category, a protocol fee may be applied to EigenCloud-related transactions routed through rewards coordinators or other protocol infrastructure, if approved by a future on-chain governance mechanism. Such fees, if enacted, would re-direct a portion of compensatory flows towards public goods and would not distribute new compensation to passive EIGEN holders or confer any *pro rata* claim on protocol-fee revenue or value.

4.3 The Role of the Ecosystem

Ecosystem participant	Today's benefits	Upcoming upgrades and benefits
ETH stakers	Secure AVSs and earn rewards through EigenLayer.	Slashing with AVS-specified routing enables new categories of AVSs with reward opportunities.
EIGEN stakers	Receive rewards for securing EigenDA.	New Primitives (EigenVerify, EigenCompute) relying on EIGEN staking increase rewards. Other AVSs contribute to further EIGEN staking rewards.
EigenDA Rollups	Benefit from high-throughput data publishing.	EIGEN Slashing for EigenDA provides economic security. New services like EigenVerify augment rollup security.
AVSs	Benefit from crypto-economic security through EigenLayer.	Can access slashing-as-a-service via EigenVerify. Can opt into shared distribution via EigenCompute, minimizing frictions to find distribution to apps.
Ethereum community	Receive rewards for ETH staking on EigenLayer. Benefit from EigenDA enabling new kinds of L2s.	Redistribution enables a new DeFi economy on EigenLayer. New kinds of apps can be built on Ethereum, augmented with EigenCloud.

	AVSs present new services for developers.	
App Developers	Build faster using existing off-the-shelf AVSs.	Rapidly build powerful apps on any chain with augmented application logic run securely offchain. Discover, consume, compose multiple AVSs easily.
Chains: L1 and L2s	N/A	Allow developers to build new categories of applications using EigenCloud.

5. The Possibilities: A world with EigenCloud

5.1 EigenCloud unlocks several novel functionalities

Here we highlight several new unlocks that come with EigenCloud. While several of these features can be built in the free market, some of them require protocol support in EigenCloud primitives. Some of these features are being experimented with in disparate ecosystems, while in EigenCloud they can be integrated.

- Support varied trust models: Cryptoeconomic, optimistic, ZK, majority honest.
    - Anything that two rational parties can agree on can be secured by EIGEN token forking
        - For such services, we can get cryptoeconomic or optimistic guarantees depending on the consumption pattern.
    - Some services will need even stronger guarantees than cryptoeconomic guarantees. They will need ZK verifiability. These services will also be available on EigenCloud. Already EigenLayer has several ZK prover networks which can be leveraged for ZK proving on the EigenCloud.
    - However, some categories cannot be built on ZK or cryptoeconomic trust. These services require majority-honest trust.
        - Secure multiparty computation, secret sharing etc.
        - zkTLS notaries.
        - Internet oracles.
        - Location oracles.
        - Threshold signatories.
        - Data storage.
    - EigenCloud will have a **curated, high-quality, decentralized operator set** to secure these services.

- Contested EIGEN forks create an onchain observable signal.
    - If outcomes from EigenDA or EigenVerify are wrong because a majority is malicious, then anyone can create a fork of the EIGEN token by burning a certain percentage of the original EIGEN token and receiving a larger percentage of the forked EIGEN token as a reward. The burning of the original EIGEN token is observable onchain - thus whenever there is a contested fork, there is a clear onchain observable signal for this purpose. Applications can use this signal to trigger their own governance to then make a choice on the right fork.
    - Example: Rollups with Ethereum security and offchain DA. If an L2 is using EigenDA, it can decide that it will prefer safety over liveness and decide to only confirm state commitments after a lag when there is no observable fork of the EIGEN token.
- Chains with unconditional security.
    - Chains can decide that they will fork together with the EIGEN token. The EIGEN token forks if there is an error in EigenDA / EigenVerify. Thus in the correct fork, EigenDA blobs will remain available and EigenVerify will return the correct answer. This is a very powerful property based on which highly-secure applications can be built.
    - Example: Prediction markets that always resolve correctly. Consider a prediction market that uses EigenVerify to resolve any disputes in its settlement process. If this prediction market is built on a chain that forks with the EIGEN token, then on the correct fork of the chain, EigenVerify returns the right answer, and hence the prediction market resolves correctly. In today's world, no prediction market has this property, as all of them rely on oracles extrinsic to the underlying blockchain, and if the oracle resolves maliciously, then the participants can lose their money.
- Secure composability for chains with Eigen forking.

    - Chains built on EIGEN token forking can interact securely with each other. There is no tradeoff between programmability and interoperability.
    - Example: Prediction market chain <> Defi chain. Suppose a DeFi chain is building financial primitives on prediction market positions, then the DeFi chain must have secure access to the outcomes from the prediction market. If both these chains choose to fork with the EIGEN token, then all these chains resolve together correctly to the same state. This benefit also applies in the reverse direction: the prediction market itself might be betting on how certain prices will resolve, and this secure interoperability benefits this direction also.
- Governance/upgrade covenants.
    - Break the tradeoff between upgradability and verifiability. An app that wants to remain upgradeable cannot make any credible commitments. EigenVerify, using intersubjective resolution, can resolve this.
    - Example: A ride-sharing app that wants to commit to taking a 5% take rate while remaining upgradeable.
    - Example: Beyond Stage-2 rollups. Unruggable rollups commit to upgrades not breaking the safety of existing apps. Extends beyond Stage 2.
- AI as secure as smart contracts.
    - A core limitation in existing blockchains is the requirement of cross-hardware determinism. This is a severe limitation for AI execution, where GPUs have several sources of non-deterministic execution and therefore cannot be verified with the same grade of security as smart contracts.
    - In contrast, on the EigenCloud all facts that can be agreed upon by any two rational observers can be resolved correctly.
    - Examples:
        - AI augmented DeFi
- AI to adjudicate disputes and automate governance.
    - Once we have AI inference and upgrade covenants, we can use that to build and keep models that perform adjudication.
    - Examples
        - Minority-token holder protection
        - AI-governed minting (memecoins)
        - AI debate resolution
        - AI resolved prediction markets
- Composition of services.
    - Services
    - Example: AI agents need multiple services, including inference, etc.

These novel functionalities can be leveraged to create the variety of applications listed earlier.

5.2 Crypto's cloud era

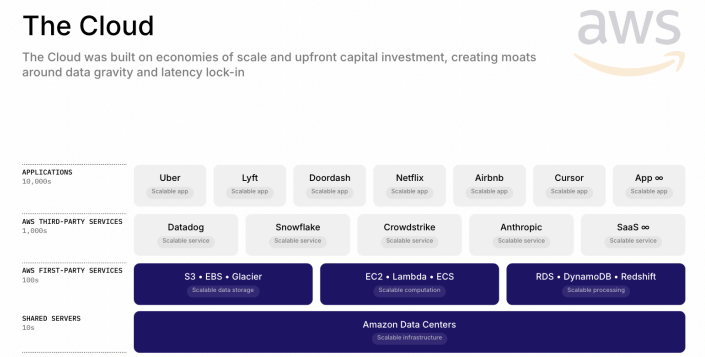
Like the internet before it, blockchain development is maturing from application-specific infrastructure to shared infrastructure to provisionable cloud infrastructure. The evolution of the internet can tell us a lot about how the evolution of blockchains might play out.

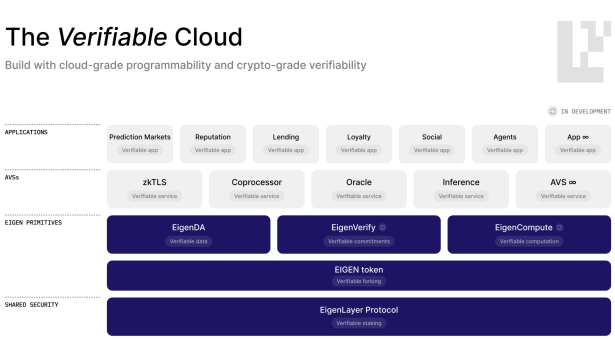
	First Era	Second Era	Third Era
Web	Build your own web server	Simple websites on shared hosts	Powerful apps using cloud and services.
Crypto	Build your own chain	Simple dapps on shared chain	Powerful verifiable apps using verifiable cloud and verifiable services.
Winners	App Tokens (Uniswap, Aave)	Chain Tokens (Ethereum, Solana)	Cloud Tokens (EIGEN, AVSs)

Web Evolution: In the decade before public cloud, application developers shifted from running their own servers to building on shared hosts like Rackspace, Equinix, and Level 3. In the mid-2000s, public cloud ushered in an unprecedented wave of app innovation by democratizing "rentable" performant compute and then arming developers with an expressive set of cloud services for analytics, data, machine learning, networking, identity, and app integration. This combination of readily-available, scalable compute and purpose-built cloud services made it easy to build powerful applications without managing physical hardware. We see that even the most successful applications continue to rely on the public cloud as the obvious choice.

Blockchain Evolution: In the past decade, we've seen two eras of blockchain application development: early app chains, and later explosions of developer activity on big/L1 chains. To usher in the wave of non-financial app innovation - including robust agents - a verifiable cloud is needed to democratize "rentable" verifiable compute and then arm developers with an expressive set of cloud services for data availability,

settlement, execution, and inference. This combination of readily-available, verifiable compute and verifiable offchain services will make it possible to build rich non-financial applications without running your own blockchain. Today, EigenLayer offers developers readily-available, verifiable compute. Rather than launch a chain with their own validator set, AVS teams can access blockchain security on demand and scale instantly across a global network of trusted operators. With EigenCloud, we offer a rich catalog of third-party AVSs and are building a set of expressive offchain services, backed by the EIGEN token. This combination of these two forms a new, verifiable cloud.





Appendix

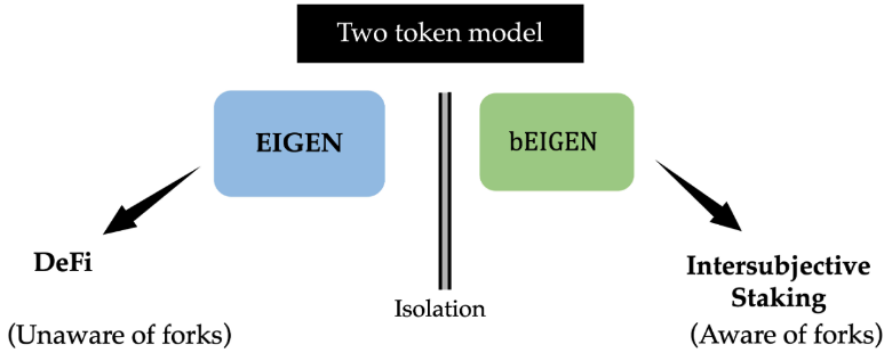
	Verifiability	Scalability	Programmability
BTC	Yes	Low	Low
Ethereum	Yes	Medium	Medium
Solana	Yes	High	Medium
EigenCloud	Adjustable	High	High
Cloud	No	High	High

EigenCloud supports a variety of trust models

EigenCloud supports a variety of trust and verifiability models. Apps using EigenCloud can use the trust model appropriate to the application.

Trust Models Primer: EIGEN Token

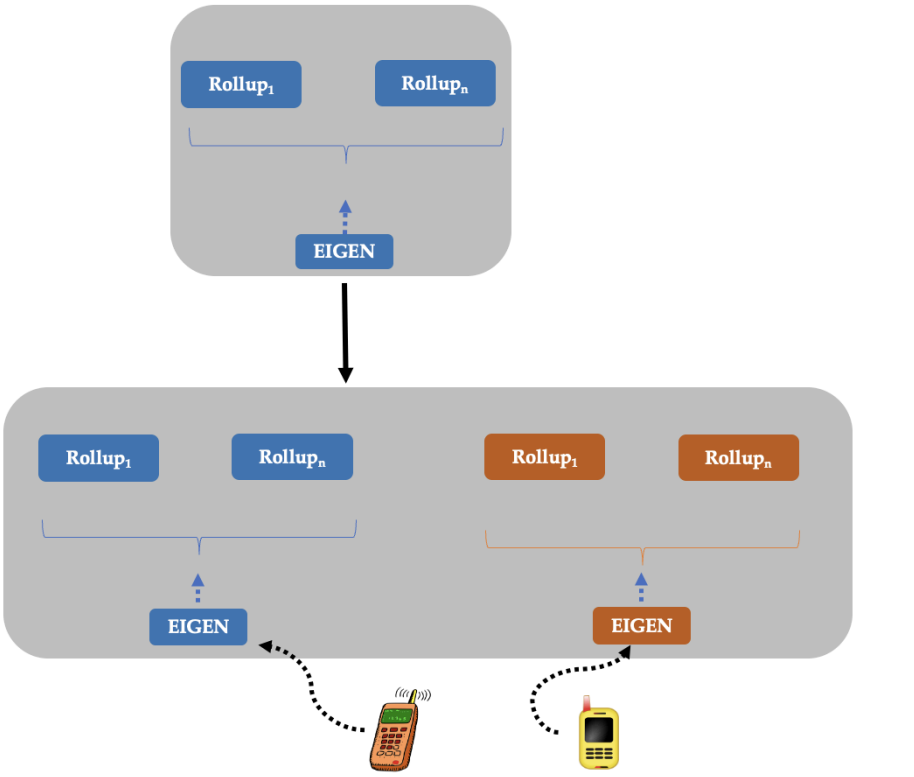
The EIGEN token is designed to fork with the Eigen Primitives: if there is a majority malicious fault on EigenDA or EigenVerify the token forks with it. For EigenCompute since it uses EigenDA and EigenVerify under the hood, it inherits the security from these services. The EIGEN token is built on a two-token model where there is a representation of the token that can be used in DeFi or exchanges without having to pick a fork. There is a bEIGEN token which can be used in staking and is forkable. The EIGEN token is designed to be redeemable on the correct fork. We refer the reader to eigen.eigenlayer.xyz for more details.



EigenVerify: Trust Models

Any rollup or AVS or app that consumes the outcome of EigenVerify is assuming that the majority of EIGEN stake backing EigenVerify is providing the right answer. If EigenVerify provides a malicious answer which is observable offchain, the token holders will be slashed via token forking. In terms of economic security, the total stake underlying EigenVerify is an upper limit on the amount of economic security that can be consumed when using EigenVerify.

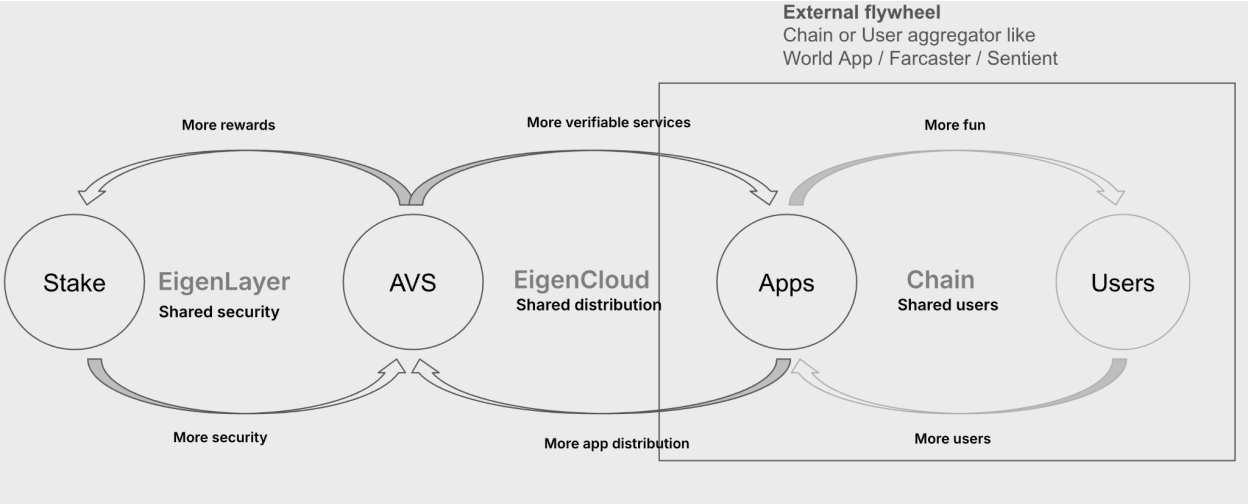
Unconditional security via chain forking

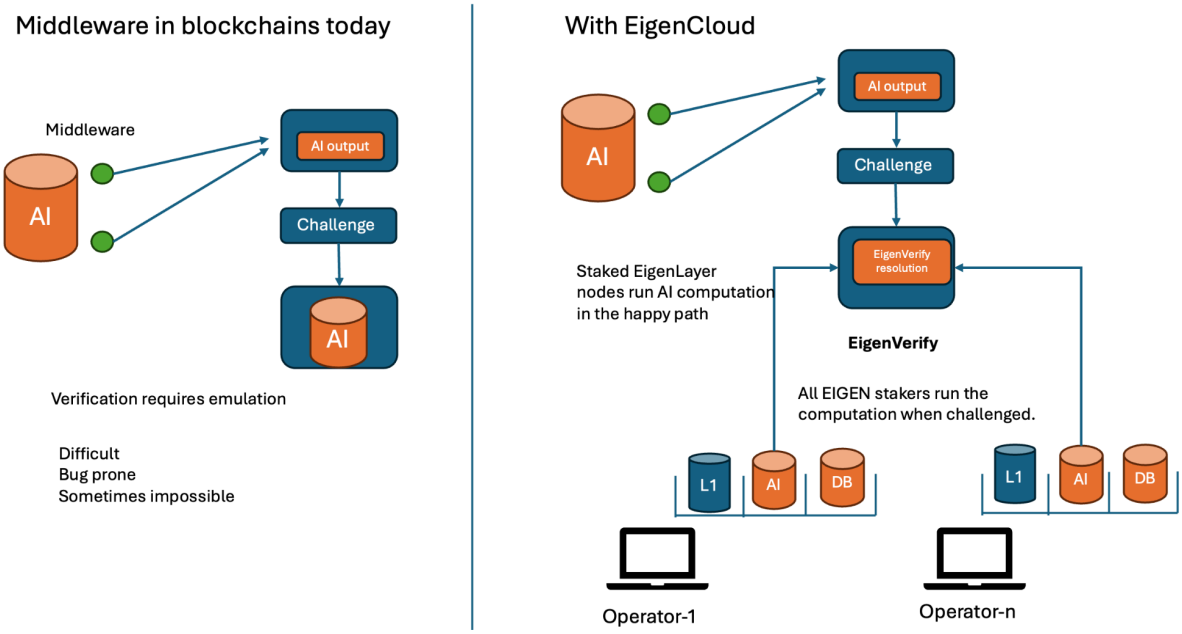


We note that some chains may be built to fork along with the EIGEN token: for example, an L2 built on EigenDA and using EigenVerify for its fraud proof may decide to fork if the EIGEN token forks. We assume that all token forking has to happen within a forkability lag. Any chain state that is either unchallenged beyond the challenge lag, or has been verified by EigenVerify beyond a forkability lag are considered irreversibly final. When the EIGEN token forks, the state of the rollup / chain will be reverted to the last irreversibly–finalized state, and therefore, on the correct side of the fork, the finalized state will always be correct. This means chains that fork with the EIGEN token inherit unconditional correctness of EigenDA and EigenVerify, i.e., blobs written to EigenDA will be available and results obtained from EigenVerify will be correct on the right fork.

Where are users and how do users interact with EigenCloud?

The EigenCloud flywheel does not include users. The relationship between users and apps is usually owned by the token ledger which connects the users to their wallet. This means we can partner with chains to up level their devs to build very powerful apps. The KPI for chains include tokens created, wallets and transactions - which we do not intend to





EigenCompute: Trust Models

	Where	Consumption pattern	Security Model
EigenCompute Immediate	Any chain	After verifying the signature from the EigenCompute operators	Unique economic security attributed to the specific app is slashable and redistributable.
EigenCompute Lagged	Any chain	Accept output if there is no challenge within challenge period, OR after	Answer is correct if there is at least one honest

		EigenVerify outcome if there is a challenge	watcher for the app AND EigenVerify is majority honest. Majority of EIGEN token holders will be forked with EigenVerify is wrong.
EigenCompute Forkable	Chains that fork with EIGEN	Accept output if there is no challenge within the challenge period. Accept EigenVerify output after a forkability lag.	On the right fork, answer is correct if there is at least 1 honest watcher for the app.
EigenCompute Eventual	Any chain	Accept output if there is no challenge within the challenge period. Accept EigenVerify output after a forking lag *if there is no fork triggered*. Use an App governance protocol to pick a fork if forking is triggered.	Answer is correct if there is at least one honest watcher for the app AND App governance protocol is correct. If forking is invoked, at least some amount of EIGEN has been burnt.