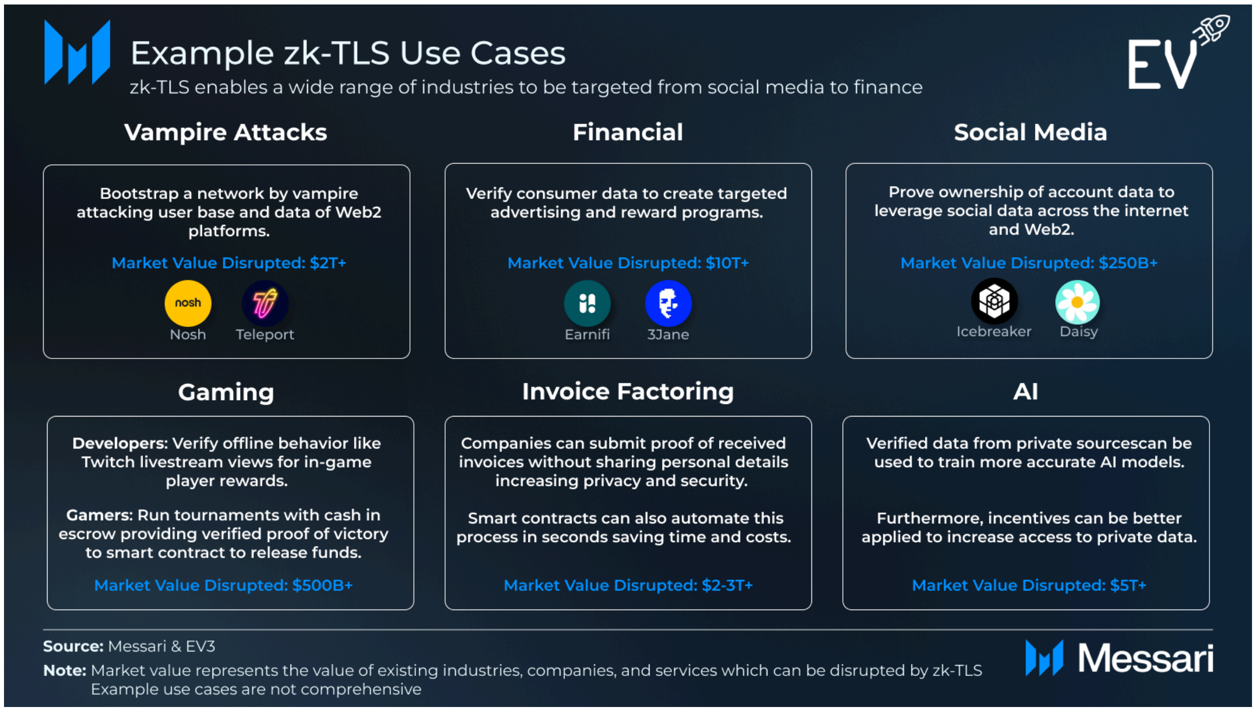
State of zk-TLS

Key Insights

- The internet today is concentrated within the walled gardens of a handful of Web2 platforms.
- zk-TLS makes all user data on Web2 platforms verifiable, enabling users to port their data to any 3rd party or smart contract without permission from the Web2 platform. It puts *the user* back at the center of the value exchange.
- Because the data has been verified, zkTLS enables other companies to build and stack economic incentives directly atop and across previously siloed data stacks with user permission.
- With 95% of internet activity secured by TLS, zk-TLS offers a wide range of potential use cases and is one of the most critical pieces enabling the vision of a sovereign and composable internet.
- We anticipate the first zk-TLS use cases to find product market fit (PMF) in 2025 and widespread adoption to occur in 2028 and beyond.

Why We Need zk-TLS



Example zk-TLS Use Cases

zk-TLS enables a wide range of industries to be targeted from social media to finance

Vampire Attacks

Bootstrap a network by vampire attacking user base and data of Web2 platforms.

Market Value Disrupted: $2T+

Nosh Teleport

Financial

Verify consumer data to create targeted advertising and reward programs.

Market Value Disrupted: $10T+

Earnifi 3Jane

Social Media

Prove ownership of account data to leverage social data across the internet and Web2.

Market Value Disrupted: $250B+

Icebreaker Daisy

Gaming

Developers: Verify offline behavior like Twitch livestream views for in-game player rewards.

Gamers: Run tournaments with cash in escrow providing verified proof of victory to smart contract to release funds.

Market Value Disrupted: $500B+

Invoice Factoring

Companies can submit proof of received invoices without sharing personal details increasing privacy and security.

Smart contracts can also automate this process in seconds saving time and costs.

Market Value Disrupted: $2-3T+

AI

Verified data from private sourcescan be used to train more accurate AI models.

Furthermore, incentives can be better applied to increase access to private data.

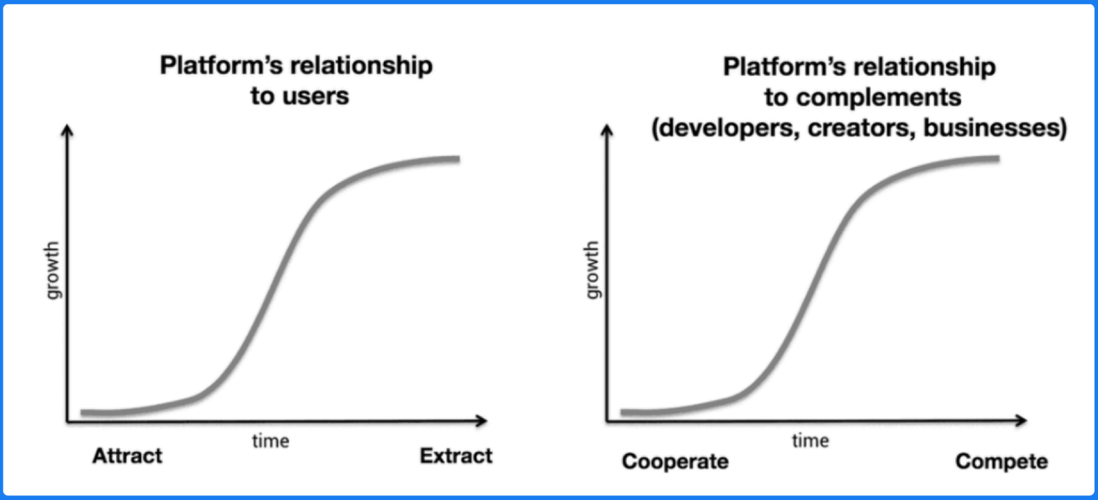
Market Value Disrupted: $5T+

Source: Messari & EV3
Note: Market value represents the value of existing industries, companies, and services which can be disrupted by zk-TLS
Example use cases are not comprehensive

The original vision for the internet was to be open and composable.

Initially, Web2 platforms encouraged composability and offered API access to attract users and developers. However, as network effects solidified, their focus shifted to value extraction, leading to walled gardens that lock in user data. Twitter, once an advocate for composability, restricted API access before its IPO in 2013, cutting off developers overnight. Currently, Facebook blocks migration of friends or messages to other platforms, Uber keeps driver ratings siloed internally, and LinkedIn provides minimal export data of connections.

Web2 Platforms Extract Value Against Users' Interests Over Time



Source: a16z

Messari

Web2 platforms control user data because Transport Layer Security (TLS), the protocol that secures communication between a website and a user, relies on two-way signing that cannot be independently verified.

Zero-knowledge Transport Layer Security (zk-TLS) hacks this system, enabling third parties to verify user data, without requiring the consent of Web2 platforms. With user permission, any exposed data (even under password) can now be verified by third parties.

In other words, **all user data across the internet is now verifiable and portable.**

With 95% of activity on the internet utilizing the TLS protocol, zk-TLS holds the potential to completely upend Web2, breaking trillion dollar monopolies and returning full data ownership to users.

Imagine receiving free Taylor Swift tickets by sharing your Spotify listening history and Starbucks reward points directly with Taylor Swift's team. Currently, this would be bogged down by the coordination between Spotify, Starbucks, and Taylor's team, months of deal-making, and likely millions in royalties. However, what if Taylor could verify you're a top 0.01% superfan and that you've ordered her branded Christmas creamsicle latte (not a real thing) *without the involvement* of either Spotify or Starbucks. Not only that, this entire process, running atop smart contracts, could be entirely automated.

That’s the power of zk-TLS.

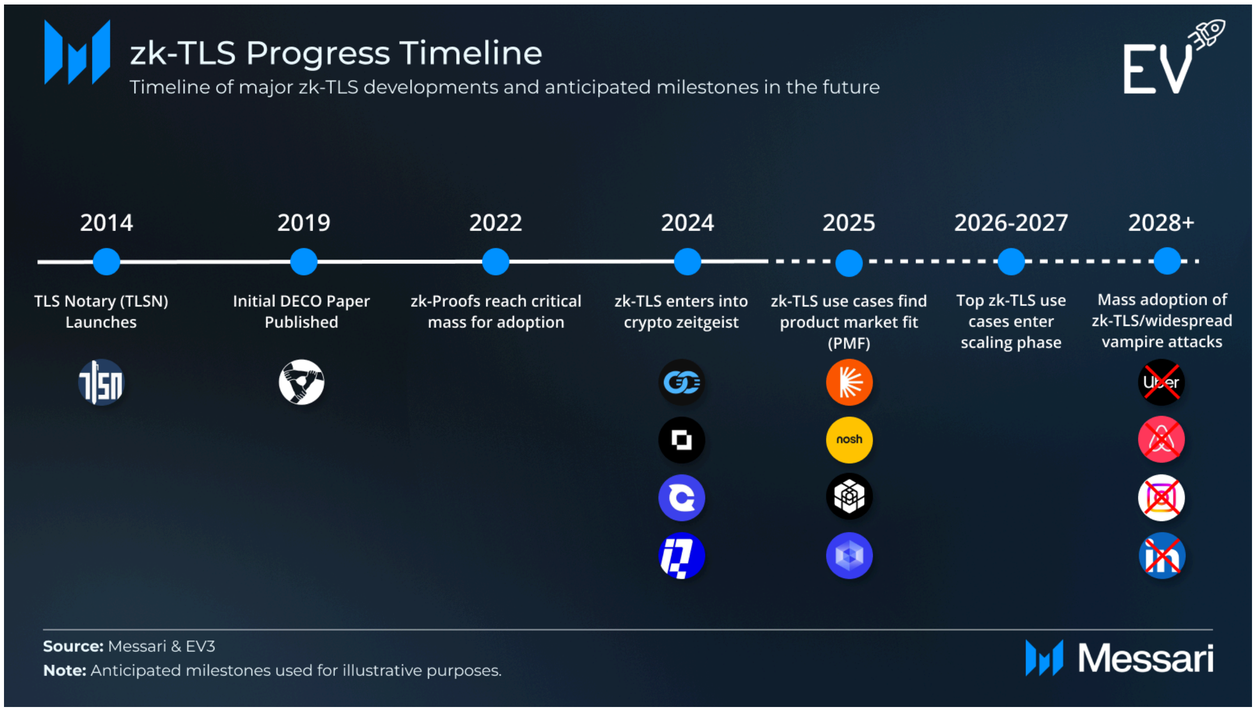
What if you aggregated all of your entertainment data across streaming services (Hulu, Disney+, Netflix), music (Apple Music, Spotify), short-form videos (Youtube Shorts, TikTok), and social media (Instagram, Facebook, VSCO). The verifiability of this data could unlock entirely novel experiences; not just better recommendations, but a user could choose to monetize their aggregate data by selling it to advertising companies or allow companies to generate rewards across all of their activity. Or, what if a game developer could offer in-game perks by verifying you’ve watched their Twitch streams, posted about them on Instagram, and listened to their podcast on Spotify.

Furthermore, using zero-knowledge proofs (ZKPs), users can cryptographically prove that they meet conditions to 3rd parties without sharing underlying data, maximizing privacy and security. For instance, a user could prove that they meet pre-determined conditions for the use cases described above without providing the data itself. From social media, to AI, to invoice factoring, the possibilities of zk-TLS are virtually limitless, all while preserving privacy.

So, how does zk-TLS work precisely and how has the space progressed so far?

In this report, we dive deep into the architecture of zk-TLS, survey the landscape of emerging zk-TLS providers and applications, highlight the most promising zk-TLS use cases, and provide insight into where the space is heading.

Background

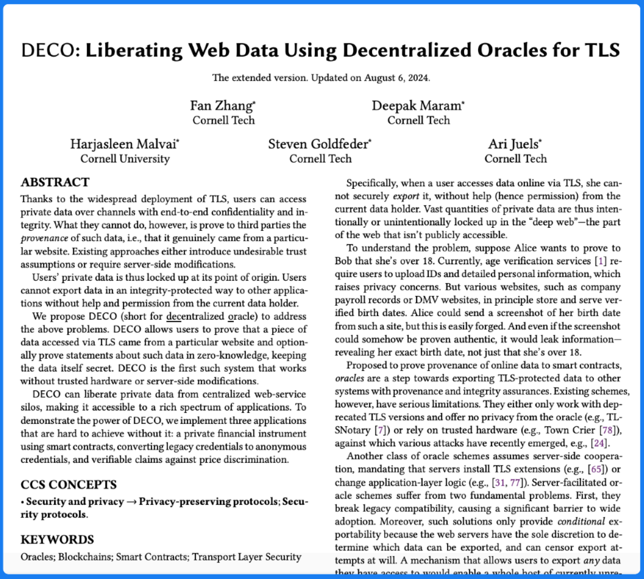


zk-TLS Progress Timeline
Timeline of major zk-TLS developments and anticipated milestones in the future
EV
2014
TLS Notary (TLSN) Launches
2019
Initial DECO Paper Published
2022
zk-Proofs reach critical mass for adoption
2024
zk-TLS enters into crypto zeitgeist
2025
zk-TLS use cases find product market fit (PMF)
2026-2027
Top zk-TLS use cases enter scaling phase
2028+
Mass adoption of zk-TLS/widespread vampire attacks
nosh
Uber
Source: Messari & EV3
Note: Anticipated milestones used for illustrative purposes.
Messari

zk-TLS represents the culmination of decades of cryptographic innovation. Its journey in crypto began in 2014 with TLSNotary (TLSN), developed largely by Ethereum researchers. TLSN successfully verified data provenance (its origin) but faced scalability challenges due to its reliance on server cooperation and lack of privacy guarantees.

In 2019, Cornell researchers introduced DECO, which utilized zero-knowledge proofs (ZKPs) to verify data authenticity without requiring server involvement. DECO also enabled privacy-preserving, selective data sharing. However, to fully scale, further advancements in ZKPs and technologies like multi-party computation were needed—advancements that ultimately paved the way for zk-TLS.

Initial DECO Whitepaper



DECO: **Liberating Web Data Using Decentralized Oracles for TLS**

The extended version. Updated on August 6, 2024.

Fan Zhang* Cornell Tech
Deepak Maram* Cornell Tech
Harjasleen Malvai* Cornell University
Steven Goldfeder* Cornell Tech
Ari Juels* Cornell Tech

ABSTRACT

Thanks to the widespread deployment of TLS, users can access private data over channels with end-to-end confidentiality and integrity. What they cannot do, however, is prove to third parties the *provenance* of such data, i.e., that it genuinely came from a particular website. Existing approaches either introduce undesirable trust assumptions or require server-side modifications.

Users' private data is thus locked up at its point of origin. Users cannot export data in an integrity-protected way to other applications without help and permission from the current data holder.

We propose DECO (short for decentralized oracle) to address the above problems. DECO allows users to prove that a piece of data accessed via TLS came from a particular website and optionally prove statements about such data in zero-knowledge, keeping the data itself secret. DECO is the first such system that works without trusted hardware or server-side modifications.

DECO can liberate private data from centralized web-service silos, making it accessible to a rich spectrum of applications. To demonstrate the power of DECO, we implement three applications that are hard to achieve without it: a private financial instrument using smart contracts, converting legacy credentials to anonymous credentials, and verifiable claims against price discrimination.

CCS CONCEPTS

• **Security and privacy → Privacy-preserving protocols**; **Security protocols**.

KEYWORDS

Oracles; Blockchains; Smart Contracts; Transport Layer Security

Specifically, when a user accesses data online via TLS, she cannot securely *export* it, without help (hence permission) from the current data holder. Vast quantities of private data are thus intentionally or unintentionally locked up in the "deep web"—the part of the web that isn't publicly accessible.

To understand the problem, suppose Alice wants to prove to Bob that she's over 18. Currently, age verification services [1] require users to upload IDs and detailed personal information, which raises privacy concerns. But various websites, such as company payroll records or DMV websites, in principle store and serve verified birth dates. Alice could send a screenshot of her birth date from such a site, but this is easily forged. And even if the screenshot could somehow be proven authentic, it would leak information—revealing her exact birth date, not just that she's over 18.

Proposed to prove provenance of online data to smart contracts, *oracles* are a step towards exporting TLS-protected data to other systems with provenance and integrity assurances. Existing schemes, however, have serious limitations. They either only work with deprecated TLS versions and offer no privacy from the oracle (e.g., TLSNotary [7]) or rely on trusted hardware (e.g., Town Crier [78]), against which various attacks have recently emerged, e.g., [24].

Another class of oracle schemes assumes server-side cooperation, mandating that servers install TLS extensions (e.g., [65]) or change application-layer logic (e.g., [31, 77]). Server-facilitated oracle schemes suffer from two fundamental problems. First, they break legacy compatibility, causing a significant barrier to wide adoption. Moreover, such solutions only provide *conditional* exportability because the web servers have the sole discretion to determine which data can be exported, and can censor export attempts at will. A mechanism that allows users to export *any* data they have access to would enable a whole host of currently unre-

Source: DECO

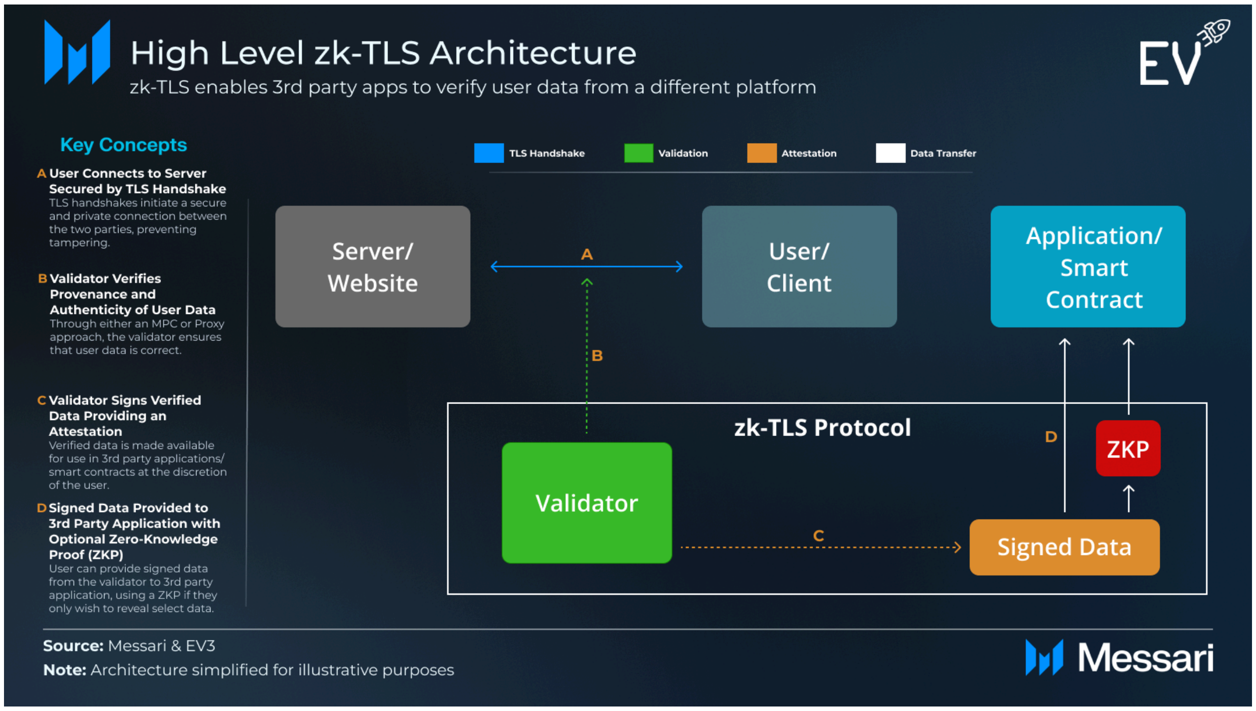
Messari

Recent breakthroughs in zero-knowledge proof technology and cryptography from 2020 to 2023 have made commercial-grade zk-TLS a reality. This has spurred the emergence of numerous zk-TLS companies, each pursuing unique approaches (outlined below) to bring this technology to market.

We anticipate the first zk-TLS use cases to achieve product-market fit (PMF) by 2025, paving the way for widespread adoption of the technology starting in 2028 and beyond.

How does zk-TLS Work

At the highest level, zk-TLS providers validate data sent between both a user/client and a server/website in a connection that is secured by a TLS Handshake. The validator verifies both the provenance (where the data comes from) and the authenticity of the data (correctness of the data) without having access to sensitive user information like passwords or the knowledge of the actual data itself. High-level generalized architecture of zk-TLS is provided below, with the specific technical implementations explained after.


High Level zk-TLS Architecture

zk-TLS enables 3rd party apps to verify user data from a different platform

Key Concepts

A User Connects to Server Secured by TLS Handshake
TLS handshakes initiate a secure and private connection between the two parties, preventing tampering.

B Validator Verifies Provenance and Authenticity of User Data
Through either an MPC or Proxy approach, the validator ensures that user data is correct.

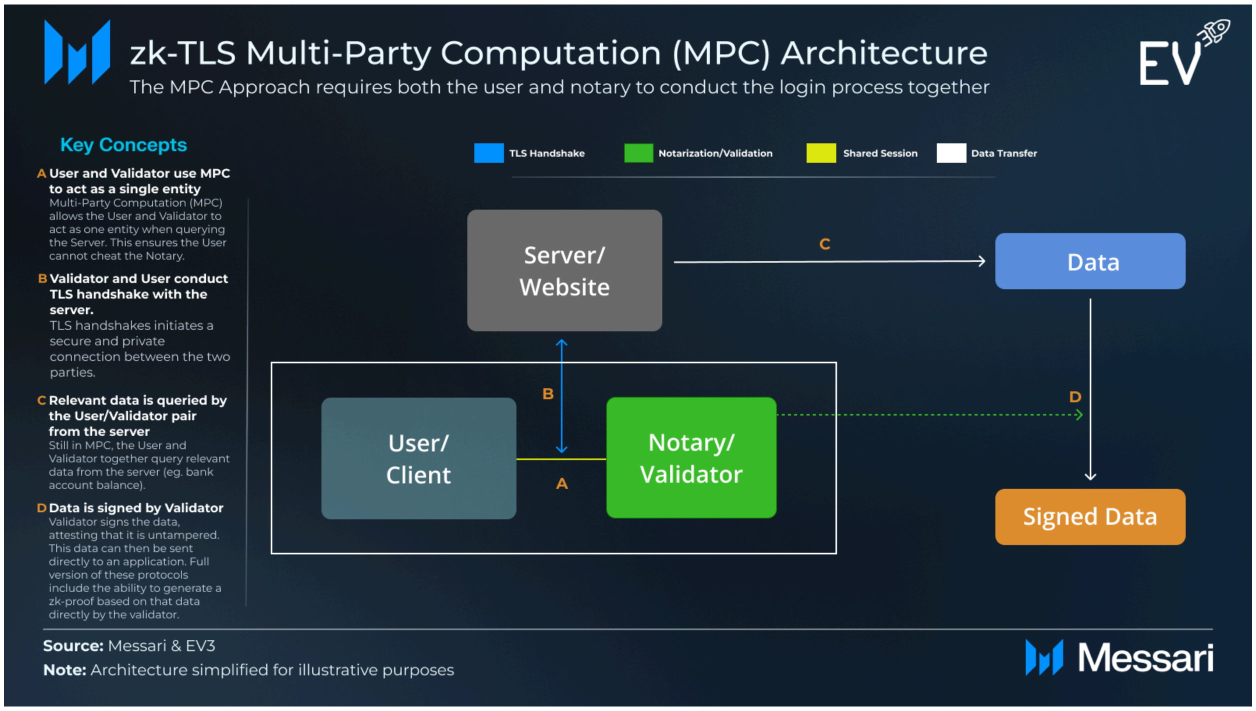
C Validator Signs Verified Data Providing an Attestation
Verified data is made available for use in 3rd party applications/ smart contracts at the discretion of the user.

D Signed Data Provided to 3rd Party Application with Optional Zero-Knowledge Proof (ZKP)
User can provide signed data from the validator to 3rd party application, using a ZKP if they only wish to reveal select data.

Source: Messari & EV3
Note: Architecture simplified for illustrative purposes

The technical implementation of zk-TLS can be broken down into three broad approaches–**Multi-Party Computation (MPC)** schemes like Opacity and Primus, **proxy** based approaches like Reclaim and Pluto, and Trusted Execution Environment (**TEE)** based architecture like Clique.

Remember this–the **goal of zk-TLS is to verify that a user logged in correctly and the server returned the appropriate information.**



zk-TLS Multi-Party Computation (MPC) Architecture

The MPC Approach requires both the user and notary to conduct the login process together

Key Concepts

A **User and Validator use MPC to act as a single entity**
Multi-Party Computation (MPC) allows the User and Validator to act as one entity when querying the Server. This ensures the User cannot cheat the Notary.

B **Validator and User conduct TLS handshake with the server.**
TLS handshakes initiates a secure and private connection between the two parties.

C **Relevant data is queried by the User/Validator pair from the server**
Still in MPC, the User and Validator together query relevant data from the server (eg. bank account balance).

D **Data is signed by Validator**
Validator signs the data, attesting that it is untampered. This data can then be sent directly to an application. Full version of these protocols include the ability to generate a zk-proof based on that data directly by the validator.

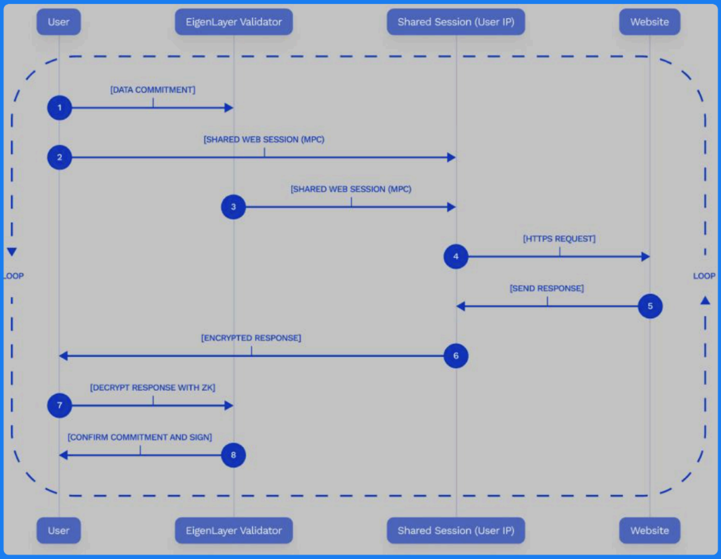
Source: Messari & EV3
Note: Architecture simplified for illustrative purposes

MPC Architecture

In the MPC approach, a trusted node called a *notary* conducts the login with the user. MPC schemes allow multiple parties to conduct a computation without any one party being able to subvert the system. A good mental model for this is a door with two different locks. To unlock the door, both Alice and Bob–who each have different keys–must be present. Without implementing the MPC scheme correctly with the notary, the notary won't attest to the data and the system remains secure.

To increase security, MPC based zk-TLS providers like Opacity leverage Eigenlayer's economic security through an AVS. For example, if a notary colludes with the user to create false data then you fabricate the transcript. If a notary stakes on Eigenlayer and if they submit something that proves that the notary colluded-- like attesting to an impossible claim, then the notary is slashed. The mechanism adds an economic incentive onto the security precautions already taken within a TEE.

Opacity’s Multi-Party Compute (MPC) Architecture

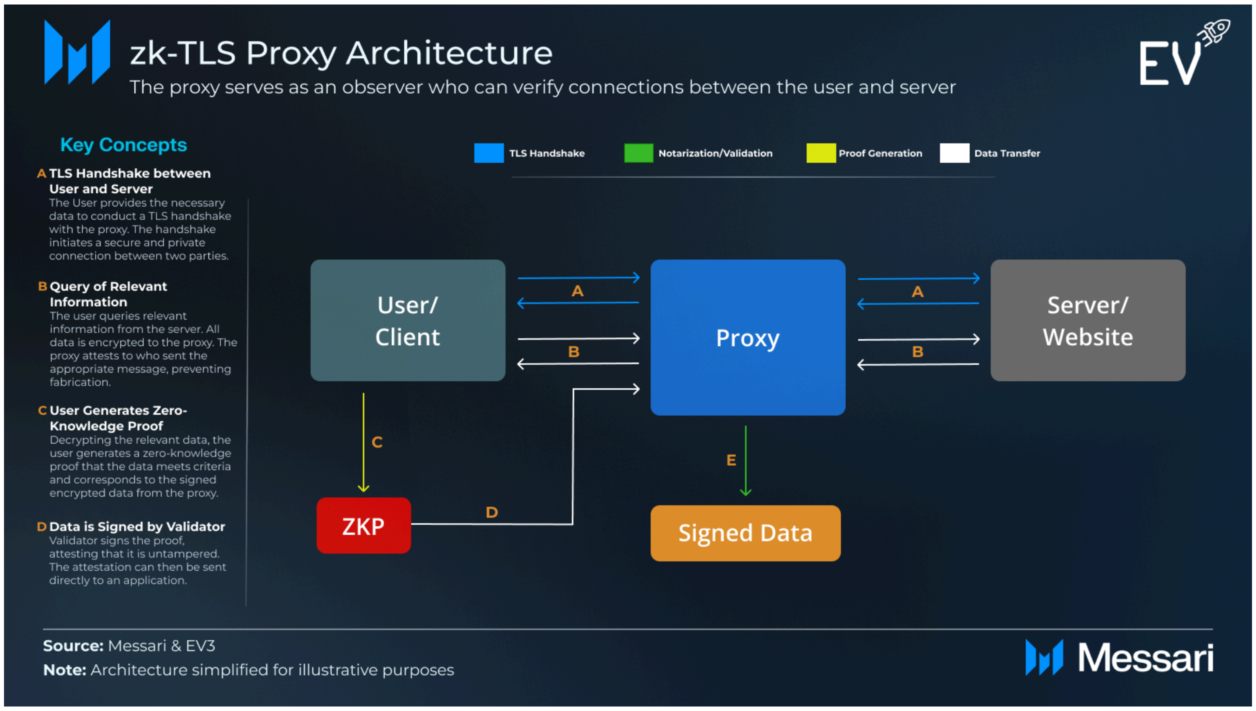


Source: Opacity

Messari

Proxy Architecture

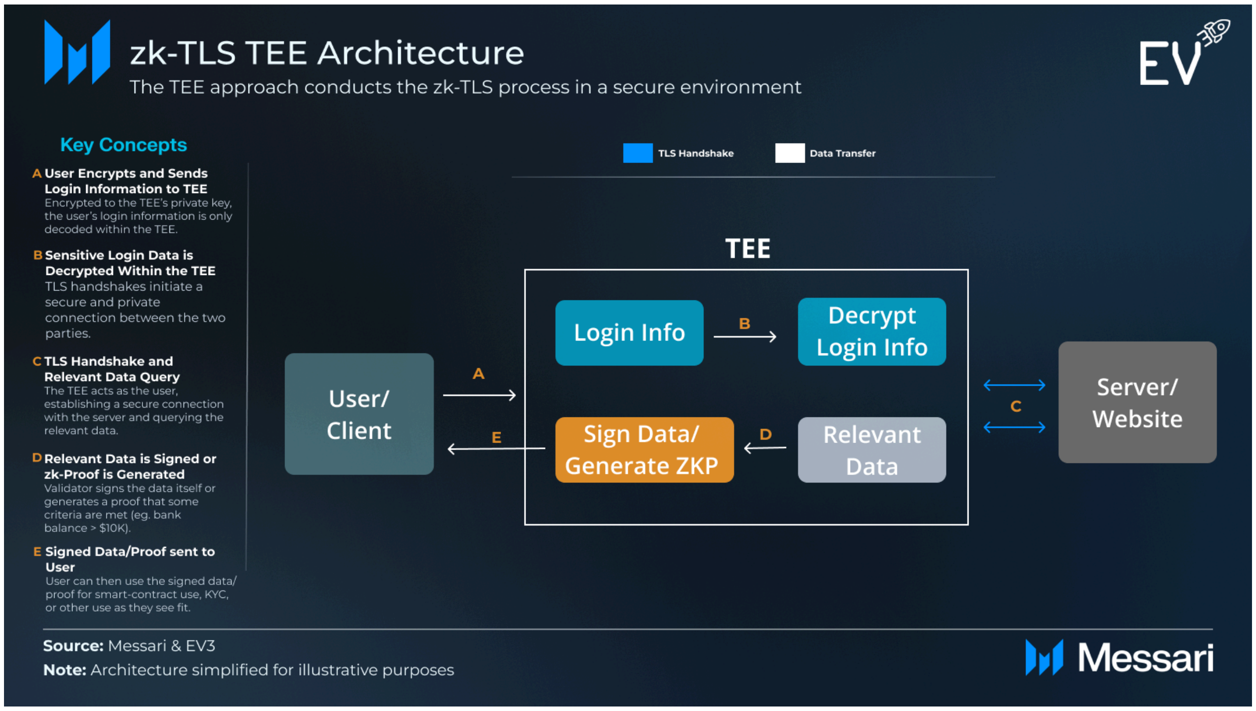
The proxy approach of zk-TLS uses the proxy feature of web browsers, where instead of sending messages directly to a website’s server, the browser sends it via a validator node that acts as an https *proxy*. Since the website will also respond to the server via this proxy, the node sees all encrypted messages via the server and the client. While the contents can’t be decrypted by the proxy, the proxy is able to sign which messages come from the server and client respectively. The client, in-browser, then generates a zero-knowledge proof that the correct information corresponding to the encrypted data was accessed and that it fits a certain criteria (e.g. bank account balance greater than $10,000).


zk-TLS Proxy Architecture
The proxy serves as an observer who can verify connections between the user and server
EV
Key Concepts
A TLS Handshake between User and Server
The User provides the necessary data to conduct a TLS handshake with the proxy. The handshake initiates a secure and private connection between two parties.
B Query of Relevant Information
The user queries relevant information from the server. All data is encrypted to the proxy. The proxy attests to who sent the appropriate message, preventing fabrication.
C User Generates Zero-Knowledge Proof
Decrypting the relevant data, the user generates a zero-knowledge proof that the data meets criteria and corresponds to the signed encrypted data from the proxy.
D Data is Signed by Validator
Validator signs the proof, attesting that it is untampered. The attestation can then be sent directly to an application.
TLS Handshake
Notarization/Validation
Proof Generation
Data Transfer
User/ Client
A
B
Proxy
A
B
Server/ Website
C
ZKP
D
E
Signed Data
Source: Messari & EV3
Note: Architecture simplified for illustrative purposes
Messari

At scale, however, this method can be blocked, especially if using a limited number of proxy nodes. Let's again take the example of proving one's bank account balance. If JP Morgan's servers are receiving hundreds or even thousands of messages from the same proxy node, it's very likely they will flag this node and block its messages. While it is possible to use providers that route requests through residential IPs, this method has challenges surrounding uptime and reliability. Furthermore, it should be noted that the use of residential IP proxies, introduces an attack surface. If the client can position themselves to be the residential IP proxy, they can fake proofs without the attestor knowing.

Other Implementations/TEE

A minimalist approach to solving zk-TLS relies on trusted execution environments or TEEs. TEEs are secure areas within a device's processor that provide an isolated environment for executing code or processing data. Since the data processing within a TEE is shielded from even the owner of the device, a user can simply encrypt their login credentials which is then decrypted within the TEE itself. The TEE acts as the user, handling the TLS handshake and communicating with the server for relevant information, all of which is handled within the TEE itself. In this way, no information should be leaked, and the TEE can return the signed data or a zero-knowledge proof that the data fits specific criteria back to the user.



Key Concepts

A **User Encrypts and Sends Login Information to TEE**
Encrypted to the TEE's private key, the user's login information is only decoded within the TEE.

B **Sensitive Login Data is Decrypted Within the TEE**
TLS handshakes initiate a secure and private connection between the two parties.

C **TLS Handshake and Relevant Data Query**
The TEE acts as the user, establishing a secure connection with the server and querying the relevant data.

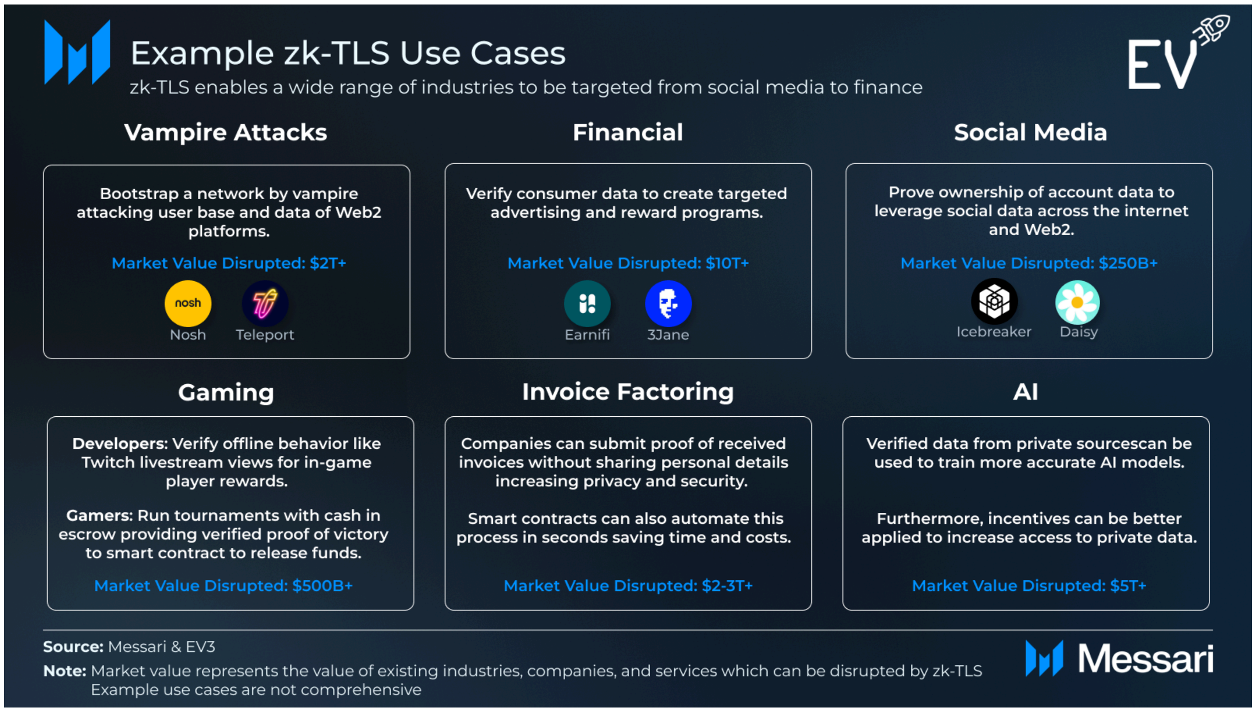
D **Relevant Data is Signed or zk-Proof is Generated**
Validator signs the data itself or generates a proof that some criteria are met (eg. bank balance > $10K).

E **Signed Data/Proof sent to User**
User can then use the signed data/proof for smart-contract use, KYC, or other use as they see fit.

Source: Messari & EV3
Note: Architecture simplified for illustrative purposes

It should be noted that the TEE approach, mainly implemented by the Clique network, entirely relies on the trust assumptions of the hardware itself. If the hardware breaks, sensitive user data like passwords can be laid bare. While schemes to break TEE security have overwhelmingly failed, some researchers have discovered flaws. Furthermore, as TEEs enter the mainstream, with companies like NVIDIA adding them into their recent GPUs, new less-tested offerings can introduce vulnerabilities for solutions solely relying on them. Some zk-TLS providers, like Opacity Network, increase their operational security by running their MPC process within a TEE, but still maintain the security provided by other mechanisms like rotating node selection and the MPC process itself.

zk-TLS use Cases


Example zk-TLS Use Cases

zk-TLS enables a wide range of industries to be targeted from social media to finance

Vampire Attacks

Bootstrap a network by vampire attacking user base and data of Web2 platforms.

Market Value Disrupted: $2T+

Nosh Teleport

Financial

Verify consumer data to create targeted advertising and reward programs.

Market Value Disrupted: $10T+

Earnifi 3Jane

Social Media

Prove ownership of account data to leverage social data across the internet and Web2.

Market Value Disrupted: $250B+

Icebreaker Daisy

Gaming

Developers: Verify offline behavior like Twitch livestream views for in-game player rewards.

Gamers: Run tournaments with cash in escrow providing verified proof of victory to smart contract to release funds.

Market Value Disrupted: $500B+

Invoice Factoring

Companies can submit proof of received invoices without sharing personal details increasing privacy and security.

Smart contracts can also automate this process in seconds saving time and costs.

Market Value Disrupted: $2-3T+

AI

Verified data from private sourcescan be used to train more accurate AI models.

Furthermore, incentives can be better applied to increase access to private data.

Market Value Disrupted: $5T+

Source: Messari & EV3

Note: Market value represents the value of existing industries, companies, and services which can be disrupted by zk-TLS Example use cases are not comprehensive

zk-TLS **puts users at the center of the value exchange,** giving them unfettered access to their data. Most importantly, because the data is verified, zk-TLS **enables developers to layer incentives and economic activity,** fundamentally changing the design space and unlocking novel use-cases across various domains.

We illustrate some of the most exciting and compelling zk-TLS use cases.

Entertainment and Media

Zk-TLS enables previously untenable use-cases across entertainment and media. In gaming, for example, users could run a decentralized tournament where they escrow money in a smart contract, run a tournament bracket, and the winner can present a zk-TLS proof to the smart contract that they won, allowing them to withdraw the money.

For game developers, zk-TLS also enables granular rewards for users. Previously, developers could only incentivize influencers and streamers with rewards, early access, and special in-game prizes for promoting their game. Now, they can apply similar incentives to players as well–whether a special character skin for watching a few hours of livestreaming about the game or even commenting during a twitch stream. Additionally, prospective players are able to opt-in, because only with their consent would a proof of stream, for example, be generated by zk-TLSing their twitch activity.

zk-TLS also enables a world with unified media experiences. Not only could users finally have aggregated access to their show history across streaming services, but also benefit from music and audiobook access from Spotify. Currently, streaming services do not optimize to provide users with their most favored content but rather content that keeps them on the platform for longer.

AI, Agents and Intents

The Data Problem

AI companies are running out of accessible training data. Large models are increasingly resorting to synthetic data (the entirety of OpenAI's latest o1 model was trained on synthetic data). The rapid data scaling from the past two years, however, is beginning to slow down. In contrast, small models are poised to grow rapidly and have shown early promise at being better suited for industry-specific tasks. Companies running small models, however, require domain-specific data to train their models. This need has fueled the fabrication of data by malicious actors seeking to take advantage of the situation.

Large players like OpenAI can initiate eight and nine-figure deals directly with platforms for specific data access, helping them overcome the bad actor problem. Smaller players, however, cannot afford such deals, often leaving them with only what is publicly available. Using zk-TLS, data can be easily verified from private sources providing low cost training data for small models. This can significantly increase the competitiveness of small models, pushing AI further.

Agents and Intent Networks

Taking action across the web3 stack requires users to specify and execute a series of transactions across a fragmented stack. Intents provide a different framework for thinking about blockchain actions. Rather than specifying individual transactions, intent-based systems have the user declare *the result* they seek (their intent). In this way, intent based systems abstract away the challenges of multi-step operations that prevent a lot of current widespread adoption.

However, intent-based systems require verification that an action has been correctly completed at the user's behest. For on-chain data, verification is usually possible, but if intent platforms want to include Web2 interactions they require zk-TLS for verification. For example, Axal, a network for verifiable autonomous agents who solve generalized intents, can use zk-TLS to check and verify the cost of placing a bet for a user on FanDuel and take action only if specific criteria are met. In conjunction with oracles, onchain data, and Web2 zk-TLS enables multi-step agentic workflows that were previously inaccessible. By combining data across multiple sources, zkTLS allows agents to link data across multiple platforms (with user permission) and make decisions if those criteria are met. Furthermore, agents can now request permission of others' data and make informed choices based on the moves made by other individuals or agents, knowing this data has been verified.

With the rise of AI agents on blockchain rails, zk-TLS can provide verifiable data enabling these agents to take on increased responsibility and possibly advance faster in their functionality.

Invoice Factoring

Large corporations, like Coca-Cola, employ hundreds of smaller, regional trucking companies to ensure timely delivery of their products. Given their size and the sheer volume of business they provide, these companies often impose a 30-60 day settlement period on their invoices–the trucking company gets paid one to two months *after* the product has been delivered. To pay employees, service repairs, and afford fuel to keep business operations alive, smaller transportation companies rely on *invoice factoring* which allows them to sell their unpaid invoices to a third-party company for a fee. The invoice factoring market was valued at $3 trillion in 2020 and is projected to grow to $5 trillion by 2026.

However, the process for verifying an invoice is entirely manual. Often, it requires the trucking company to submit their *login and password information* so the underwriting bank can login directly and check the validity of a delivered invoice. Not only is this a security concern, but the need for human intervention increases processing times.

zk-TLS fixes both problems, without need for any expensive integrations. Using zk-TLS, a trucking company could submit a proof of invoice delivery without sharing their login information and preserving privacy. Initially, we envision this will be confirmed by the underwriter, but as more financial institutions rely on the flexibility provided on-chain, even this process could be handled by a smart contract. Soon, a trucking company will be able to submit a proof of invoice to a smart contract and receive payment within minutes, all without the need for human oversight.

zk-TLS Unlocks DePIN

DePIN's Verification Problem

A significant challenge with DePIN is the verification problem, particularly for High-P DePINs dependent on location-based physical infrastructure. DePINs need to prevent tampering and fraud with network infrastructure to properly distribute rewards to participants and maintain the DePIN flywheel effect to grow the network.

Take Daylight Energy, an energy DePIN which aggregates distributed energy resources (DERs) to provide higher quality energy data for operating the grid. Households can connect their solar panels, EV chargers, thermostats, batteries, and other DERs to Daylight's network providing real time data of their energy usage in exchange for rewards.

Daylight uses Opacity's MPC model to verify energy expenditure for the month which they can cross-check with their DER to help ensure that people didn't tamper with the sensors. zk-TLS can help solve the verification problem for DePINs accelerating their expansion and adoption.

Web2 Vampires: Nosh and Teleport

Large Web2 gig-economy platforms take a significant cut of user fees, reducing profits for gig workers. For example, rideshare platforms like Uber take an average of 44% of rider fees. In contrast, a DePIN approach to rideshare could increase the share of earnings going to gig workers by eliminating centralized, rent-seeking intermediaries. Emerging protocols like Nosh (for food delivery) and Teleport (for ridesharing) are pioneering DePIN-based solutions to address this imbalance. Notably, Teleport currently charges only 15% of rider fees, significantly boosting driver earnings.

However, scaling DePIN protocols comes with challenges, particularly in ensuring driver quality and minimizing potential incidents. Nosh and Teleport aim to tackle these issues using zk-TLS. This approach enables drivers to securely verify their ratings and other relevant data from Web2 platforms like Uber, Lyft and DoorDash without revealing sensitive information. By leveraging zk-TLS, these DePIN protocols can seamlessly onboard drivers from existing platforms while maintaining quality standards comparable to Web2 services.

Such a strategy offers a cost-effective way to bootstrap supply for DePIN networks, creating a competitive edge against Web2 giants. With lower fees and higher earnings for gig workers, DePIN protocols like Nosh and Teleport have the potential to disrupt established players.

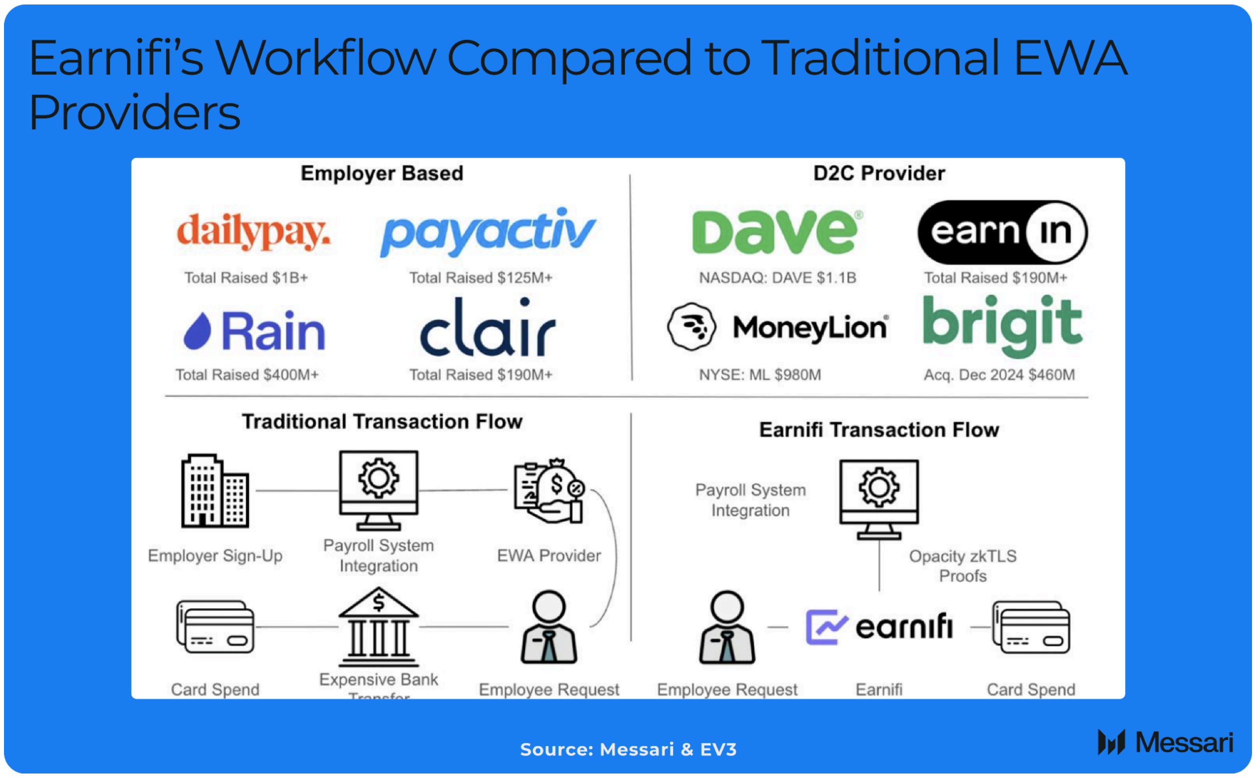
Overall, zk-TLS can help DePIN overcome bootstrapping and scaling constraints, accelerating the build out of such networks.

More Efficient Use-Cases: Earnifi

78% of Americans live paycheck to paycheck, often relying on payday loans, earned-wage access (EWA) providers, or incurring overdraft fees. Payday loans charge $10-$30 per $100 borrowed, with APRs (Annual Percentage Rates) on average exceeding 400%. EWA providers, while offering faster access to wages, still face high costs from traditional financial rails and intermediaries. Although free 3-day ACH transfers are an option, 72% of users choose instant transfers at 5-6%, compounding to over 110% APR for frequent users. Often, these exorbitantly high interest rates can quickly spiral into a cycle of debt, with 4 in 5 users having to renew their loans, usually before their next paycheck.

The industry is split into employer-partnered and direct-to-consumer (D2C) platforms. Employer-partnered models require negotiated integrations, limiting accessibility, while D2C platforms rely on historical payroll data, leading to inaccuracies and higher delinquency rates.

Earnifi, leveraging zk-TLS through Opacity and crypto rails, streamlines the process with real-time wage data. By syncing directly with payroll systems (with user consent), Earnifi eliminates costly integrations, updates wage data daily, and drastically reduces charge-off rates from 6.3% to 0.3%. This inclusive, opt-in solution works for all Americans, regardless of employer size. Future crypto payment advancements and faster zk-TLS generation could enable seamless, real-time payment streaming at a fraction of current costs.



Aggregating Data: Icebreaker

Icebreaker is building an open, decentralized LinkedIn where users control their data. Using Reclaim's proxy approach for zk-TLS, Icebreaker allows users to import their entire network over in a single platform. This synthetic approach gives users full control over how they present their network and how they navigate building relationships. According to Icebreaker CEO Dan Stone, since adding this functionality, Icebreaker users are averaging 1300 contacts imported with over 50,000 imported on the entire platform. Turning on invites for all of these imports would equate to a k-factor of around 100 based on data collected by the team thus-far.

Social Media Megalith: Daisy

In the last decade, social media marketing has exploded as a percentage of companies' adspend and is poised to continue. Brands often partner with influencers to boost exposure to their products. Social media algorithms, however, are known to boost posts with organic engagement (those that are liked, commented on, and shared by other users). Collaboration between influencers, therefore, drives an order of magnitude improvement for both the sponsoring company and the influencer's reach. In fact, in early testing, brands have seen 10-20x higher traffic and 25-50% lower cost-per-acquisition on boosted content vs posts by a single influencer.

The challenge, however, lies in verifying that a social media influencer did in-fact like, share, or comment on another's post.

Daisy uses zk-TLS to turn paid social media influencing from a single-player game into a multi-player one. Influencers can now get paid for boosting each other's sponsored content via likes, comments, and reposts. This boosting creates an echo chamber effect whereby sponsored content is more likely to go viral within the influencers' overlapping social graphs. Using Opacity's zk-TLS method, Daisy is able to enhance attribution of engagement across influencers.

Without zk-TLS, Daisy or a similar company would be only able to tune attribution metrics based on data publicly revealed by social media networks. For example, while anyone can see how many likes an arbitrary Twitter post has, only the creator of the post can view which specific profiles have liked it. zk-TLS allows Daisy to reward influencers based on metrics that are exposed privately to individual users. zk-TLS also enables Daisy to verify posts in exclusive/paid social media groups which would otherwise be inaccessible to a third-party platform.

Challenges

zk-TLS faces a number of challenges threatening its potential and slowing down progress.

- **Scalability Concerns:** The proxy approach in particular could face server denial if too many sessions are initiated. zk-TLS has not been implemented at a significant scale yet, and it remains to be seen whether large scale use cases like vampire attacks work in practice.

- **Well-Funded Web2 Platforms:** zk-TLS could become an existential threat to certain Web2 platforms. These platforms have immense resources to invest in both technological and legal challenges against zk-TLS providers should the threat become credible.
- **Lack of Familiarity with Novel technology:** zk proofs are not well understood outside of blockchain and cryptography, leading to reluctance in integrating the technology among Web2 developers.

Conclusion

zk-TLS holds the promise to upend Web2 by giving users ownership over all their data. Such ownership enables users to port their data to 3rd parties unlocking a wide range of use cases across any space that interacts with the internet. Such technology could prove to be the key in both solving long standing problems in Web2 and making Web3 a reality.

The adoption of zk-TLS is still in its infancy, however, and much must be proven before the technology can make good on its potential. 2025 stands to find the first zk-TLS use case with PMF, opening the door to mass adoption.

zk-TLS is a must watch space in the new year and beyond.