2023/2024 S2

Layer 2 etc.

Dr. Hui Gong (h.gong1@westminster.ac.uk)

UNIVERSITY OF WESTMINSTER

From the Dencun Update

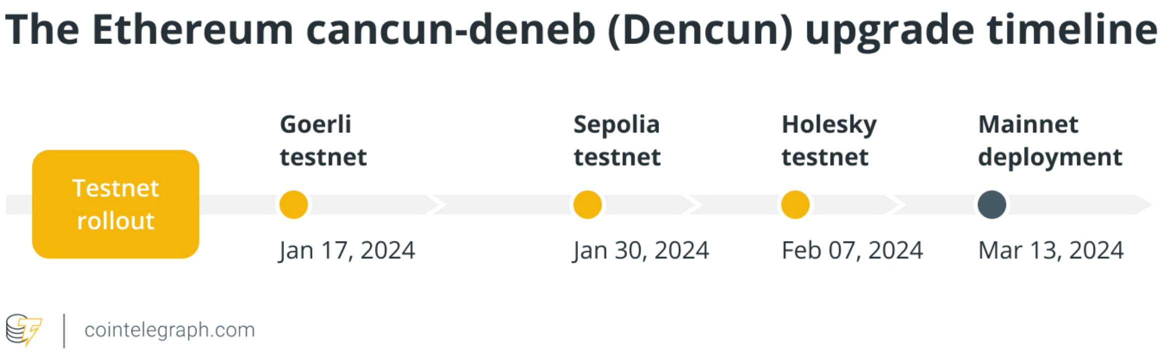
From the Dencun Update

Ethereum Cancun-Deneb (Dencun) upgrade

Dencun, an amalgamation of two separate upgrades — Deneb and Cancun — tackles both Ethereum's consensus and execution layers in a single upgrade.

Technically a hard fork in blockchain terminology, the upgrade rolled out at Ethereum epoch 269,568 at 1:55 pm UTC and finalized at 2:10 pm 13 March 2024. The upgrade is set to significantly slash the transaction fees of layer-2 solutions and boost the scalability of Ethereum.

Dencun unveils **proto-danksharding**, which will benefit L2 networks such as Polygon, Arbitrum and Optimism, among others.


The Ethereum cancun-deneb (Dencun) upgrade timeline

cointelegraph.com

From the Dencun Update

What is EIP-4844 (proto-danksharding)

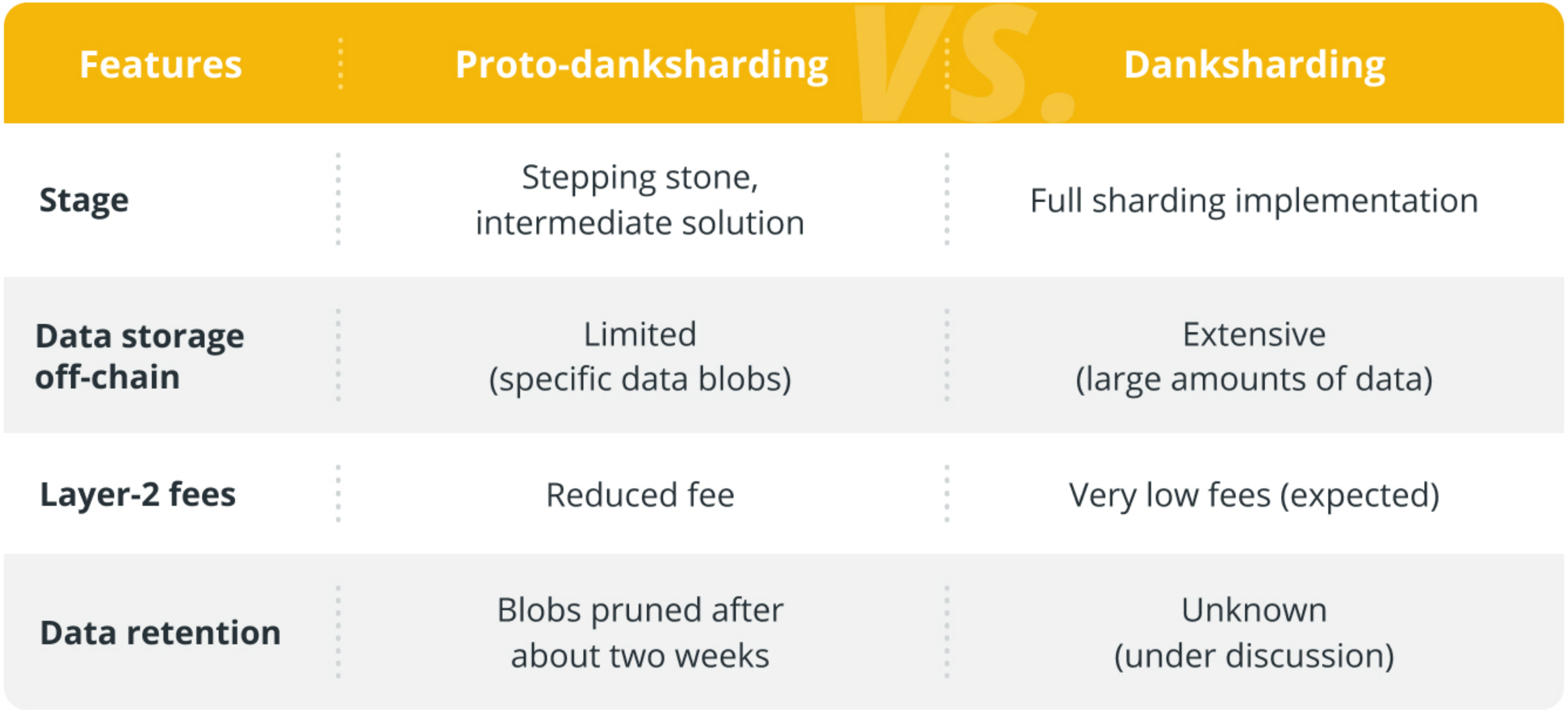
Proto-danksharding, which has introduced data blobs via Ethereum improvement proposal (EIP)-4844, is the most notable feature of the upgrade.

Data blobs are a novel solution intended to improve the efficacy of L2 transaction data storage. Currently, L2 solutions use **transaction calldata**, which the upgrade will replace with **blob data**. Calldata stores limited transaction data on-chain, which needs to be retained by the nodes forever, significantly increasing the burden on validators.

Proto-danksharding is a novel technique that removes the limitations of the present on-chain data storage system, opening up a vastly more effective data management system. By utilizing data blobs specifically designed to manage large volumes of data outside the Ethereum blockchain, rollups benefit from more scalable data storage.

From the Dencun Update

Proto-danksharding vs. danksharding



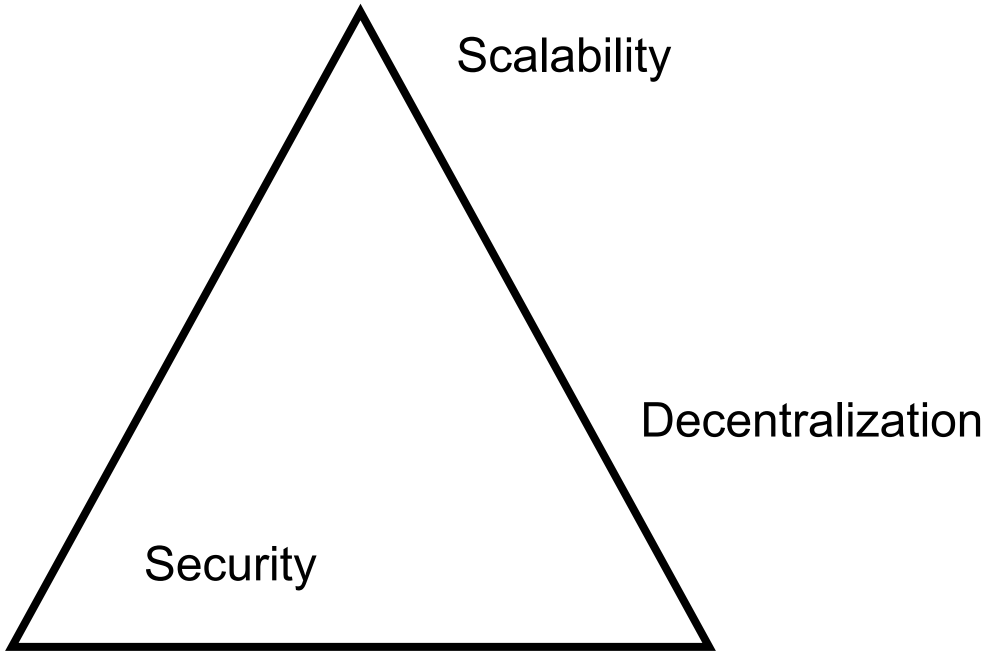
Features	Proto-danksharding	Danksharding
Stage	Stepping stone, intermediate solution	Full sharding implementation
Data storage off-chain	Limited (specific data blobs)	Extensive (large amounts of data)
Layer-2 fees	Reduced fee	Very low fees (expected)
Data retention	Blobs pruned after about two weeks	Unknown (under discussion)

It reduces network congestion and optimizes network size for better performance. Additionally, lower gas prices will increase the usability and efficacy of L2 applications.

From the Dencun Update

Expanding Horizons with Layer 2 Solutions

Layer 2 solutions on the Ethereum network are essential because they address the blockchain trilemma of decentralization, scalability, and security. The blockchain trilemma posits that it is challenging to achieve all three properties simultaneously; typically, a blockchain can only excel at two out of the three. Ethereum, like other Layer 1 blockchains, is decentralized and secure but has struggled with scalability, especially as the network has grown and the demand for block space has increased. This results in high transaction fees and slower processing times during periods of congestion.



From the Dencun Update

Layer 2 solutions

Layer 2 solutions, such as Rollups, aim to alleviate these issues by handling transactions off the main chain, thus increasing throughput and reducing costs without compromising on the network's decentralization or security. Rollups work by grouping or "rolling up" multiple transactions into a single one, and then submitting this to the main Ethereum chain. This process helps in relieving congestion on the network and makes transactions faster and less expensive.

There are two main types of Rollups:

- **Optimistic Rollups**: Optimistic Rollups assume transactions are valid by default and only run computations in the event of a challenge
- **ZK-Rollups**: ZK-Rollups use zero-knowledge proofs to verify transactions’ validity without revealing their contents

Both have their own set of advantages and trade-offs. ZK-Rollups, for instance, offer faster transaction speeds and lower fees but require a more complex setup that can lead to centralization. Optimistic Rollups, on the other hand, are easier to implement but have a challenge period that can delay withdrawals.

From the Dencun Update

Optimistic Rollups vs. ZK-Rollups

Feature	Optimistic Rollups Example	ZK-Rollups Example
EVM Compatibility	Arbitrum, Optimism	Polygon zkEVM, Scroll zkEVM
Transaction Costs	Lower due to minimal data on-chain	Potentially higher due to proof generation
Security	Based on economic incentives and fraud proofs	Based on cryptographic proofs
Use Cases	DeFi applications, decentralized exchanges	Identity verification, private crypto trading

Zero-Knowledge Proof

What Is a Zero-Knowledge Proof?

Zero-knowledge proofs (ZKPs) are a cryptographic method used to prove knowledge about a piece of data, without revealing the data itself.

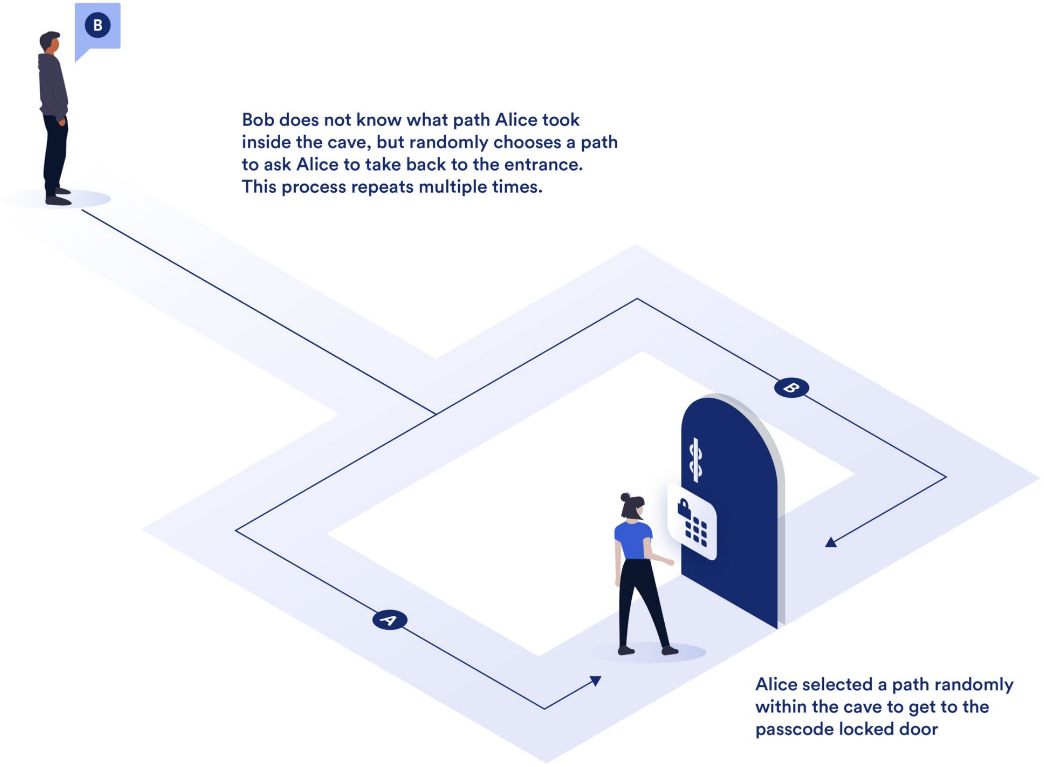
While the inherent transparency of blockchains provides an advantage in many situations, there are also a number of smart contract use cases that require privacy due to various business or legal reasons, such as using proprietary data as inputs to trigger a smart contract's execution. An increasingly common way privacy is achieved on public blockchain networks is through zero-knowledge proofs (ZKPs)—a method for one party to cryptographically prove to another that they possess knowledge about a piece of information without revealing the actual underlying information. In the context of blockchain networks, the only information revealed on-chain by a ZKP is that some piece of hidden information is valid and known by the prover with a high degree of certainty.

Zero-Knowledge Proof

How Do Zero-Knowledge Proofs Work

The three fundamental characteristics that define ZKP include:

- **Completeness**: If a statement is true, then an honest verifier can be convinced by an honest prover that they possess knowledge about the correct input.
- **Soundness**: If a statement is false, then no dishonest prover can unilaterally convince an honest verifier that they possess knowledge about the correct input.
- **Zero-knowledge**: If the state is true, then the verifier learns nothing more from the prover other than the statement is true.



Bob does not know what path Alice took inside the cave, but randomly chooses a path to ask Alice to take back to the entrance. This process repeats multiple times.

Alice selected a path randomly within the cave to get to the passcode locked door

Zero-Knowledge Proof Use Cases

Zero-knowledge proofs unlock exciting use cases across Web3, enhancing security, protecting user privacy, and supporting scaling with layer 2s.

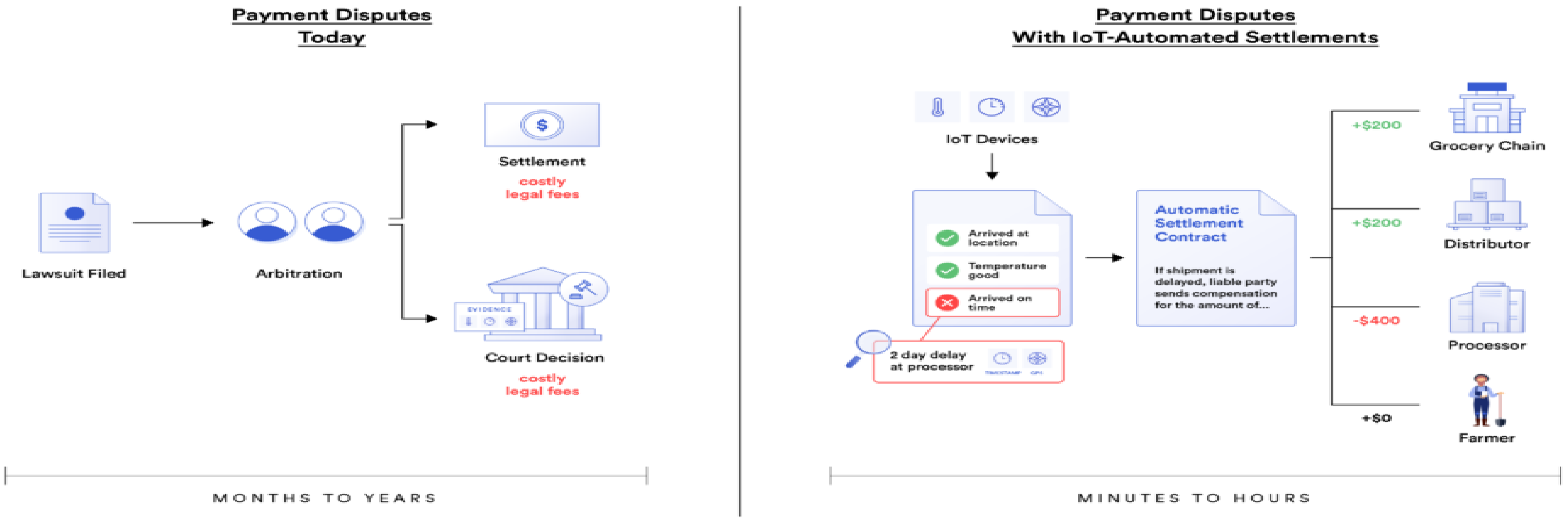
- **Private Transactions**: ZKPs have been used by blockchains such as Zcash to allow users to create privacy-preserving transactions that keep the monetary amount, sender, and receiver addresses private.
- **Verifiable Computations**: Decentralized oracle networks, which provide smart contracts with access to off-chain data and computation, can also leverage ZKPs to prove some fact about an off-chain data point, without revealing the underlying data on-chain.
- **Highly Scalable and Secure Layer 2s**: Verifiable computations through methods such as zk-Rollups, Validiums, and Volitions enable highly secure and scalable layer 2s. Using layer 1s such as Ethereum as a settlement layer, they can provide dApps and users with faster and more efficient transactions.
- **Decentralized Identity and Authentication**: ZKPs can underpin identity management systems that enable users to validate their identity, while protecting their personal information. For example, a ZKP-based identity solution could enable a person to verify that they're a citizen of a country without having to provide their passport details.

Zero-Knowledge Proof Use Cases

Voting Systems

Zero-knowledge proofs can be used to create highly secure and verifiable voting mechanisms that enable individuals to cast votes without compromising their identity or revealing who they voted for.

Internet of Things (IoT)



Zero-Knowledge Proof

Verifiable Random Function (VRF)

A verifiable random function (VRF) is a cryptographic function that takes a series of inputs, computes them, and produces a pseudorandom output and proof of authenticity that can be verified by anyone.

In cryptography, a verifiable random function (VRF) is a random number generator (RNG) that generates an output that can be cryptographically verified as random. Verifiable randomness is essential to many blockchain applications because its tamper-proof unpredictability enables exciting gameplay, rare NFTs, and unbiased outcomes.

As the name suggests, a verifiable random function is defined by its core features:

- **Verifiable** — Anyone can verify that the random number generated by a VRF is valid.
- **Random** — The output of a VRF is entirely unpredictable (uniformly distributed) to anyone who doesn’t know the seed or private key and follows no pattern.
- **Function** — VRFs rely on a mathematical algorithm to produce both the random number and a proof that verifies its authenticity.

Cross-Chain Bridges

What Is Cross-Chain?

Cross-chain technology refers to the ability to transfer data and tokens between different blockchains.

The Web3 landscape is increasingly becoming multi-chain, with the DApp ecosystem existing across hundreds of blockchains, layer-2 networks, and appchains. However, blockchains don’t have the native ability to communicate with external systems or APIs. This limitation not only prevents blockchains from communicating with existing web infrastructure but also with other blockchains.

Given the wide variety of blockchain ecosystems, it’s critical that these distinct on-chain environments are able to interoperate and communicate with each other. Cross-chain interoperability protocols are a critical piece of infrastructure for exchanging data and tokens between different blockchains.

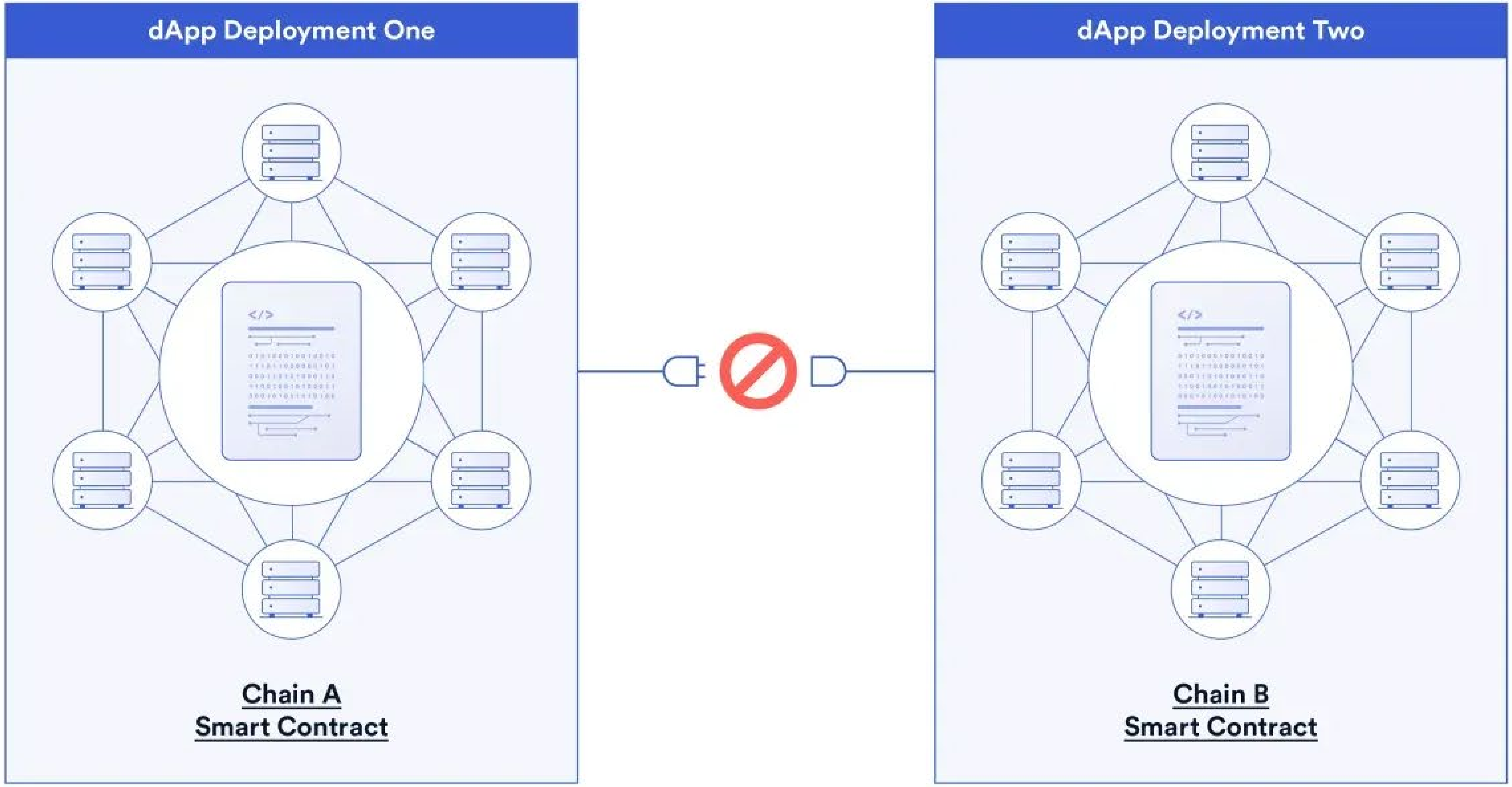
Why Is Cross-Chain Communication Important?

Cross-chain interoperability is critical for a more integrated Web3 ecosystem as well as for building bridges between existing Web2 infrastructure and Web3 services. By enabling cross-chain smart contracts, cross-chain interoperability solutions reduce fragmentation in the ecosystem and unlock higher capital efficiency and better liquidity conditions.

DeFi’s permissionless composability has given rise to increasingly complex applications that allow developers to combine distinct DApps into a structure that can achieve more than the sum of its parts. However, composability is significantly hindered with hundreds of different networks, as a smart contract can only natively compose with other contracts on the same network. If an application wants to follow the users and remain competitive in a rapidly changing multi-chain environment, it has to be deployed on multiple platforms, leading to fragmented liquidity and a degraded user experience. Furthermore, individual DApp deployments take up precious development resources that could otherwise be spent improving the business logic of the application.

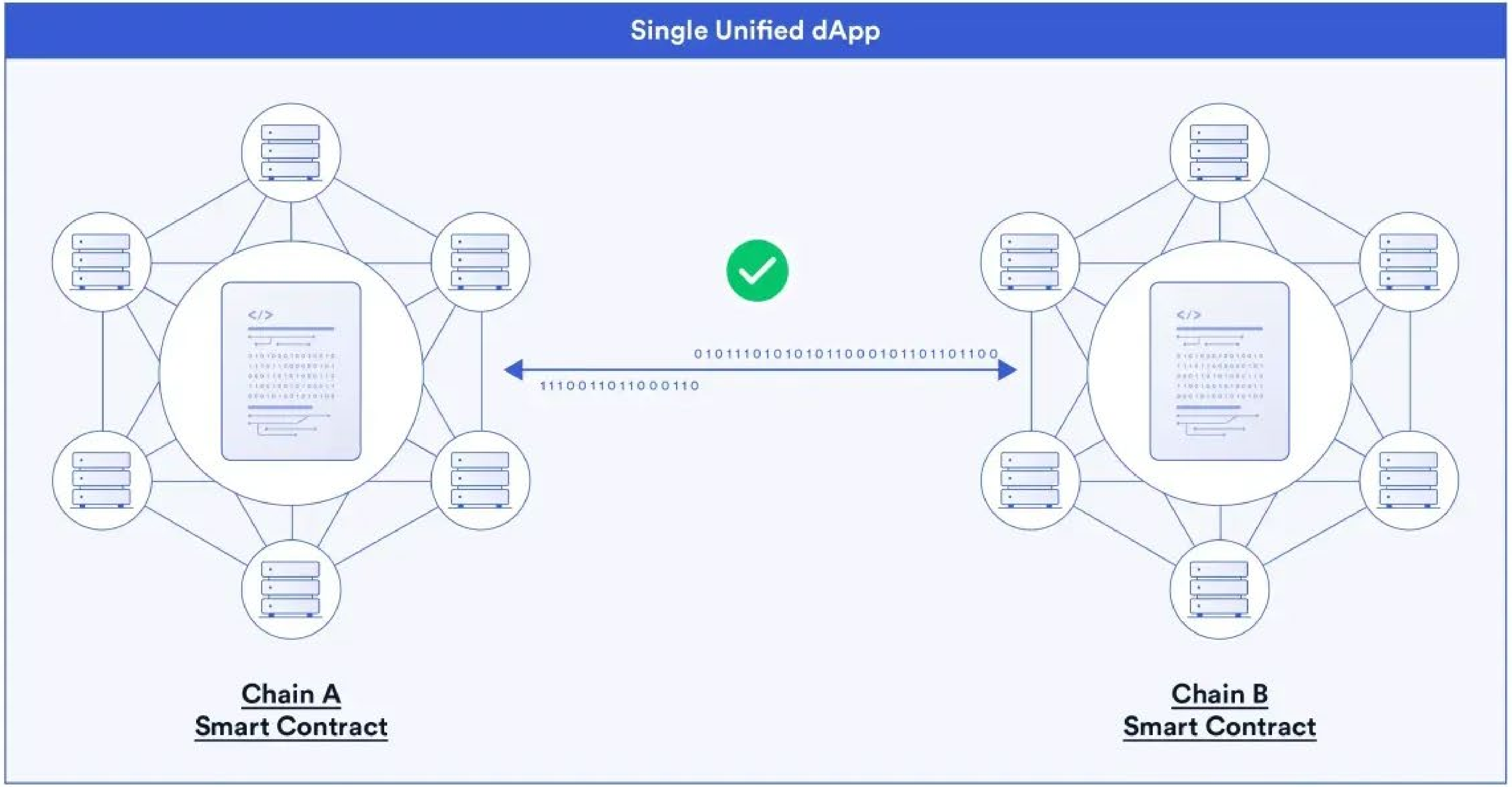
Cross-Chain Bridges

Why Is Cross-Chain Communication Important?



Cross-Chain Bridges

Why Is Cross-Chain Communication Important?



How Does Cross-Chain Technology Work?

Cross-chain solutions typically involve **validating the state of the source blockchain** and **relaying the subsequent transaction to the destination blockchain**. Both of these functions are required to complete most cross-chain interactions.

One key piece of infrastructure is a **cross-chain bridge** that enables tokens to be transferred from a source blockchain to a destination blockchain. A cross-chain bridge typically involves locking or burning tokens on the source chain through a smart contract and unlocking or minting them through another smart contract on the destination chain. In effect, a cross-chain bridge is a cross-chain messaging protocol applied to a very narrow use case—transferring tokens between different blockchains. As such, cross-chain bridges are often application-specific services between two blockchains.

Cross-chain bridges are only one simple application serving cross-chain functionality. Programmable token bridges enable more complex cross-chain interactions, such as swapping, lending, staking, or depositing tokens in a smart contract in the same transaction that the bridging function is executed, while arbitrary data messaging protocols provide more generalized cross-chain functionality, which can support the creation of more complex dApps such as cross-chain decentralized exchanges (DEXs), cross-chain money markets, cross-chain NFTs, cross-chain games, and much more.

Types of Cross-Chain Bridges

A cross-chain bridge is a type of decentralized application that enables the transfer of assets from one blockchain to another. Cross-chain bridges increase token utility by facilitating cross-chain liquidity between distinct blockchains. A cross-chain bridge typically involves locking or burning tokens on the source chain through a smart contract and unlocking or minting tokens through another smart contract on the destination chain.

Cross-chain bridges are powered by three main mechanism types:

- **Lock and mint** — A user locks tokens in a smart contract on the source chain, then wrapped versions of those locked tokens are minted on the destination chain as a form of IOU. In the reverse direction, the wrapped tokens on the destination chain are burned to unlock the original coins on the source chain.
- **Burn and mint** — A user burns tokens on the source chain, then the same native tokens are re-issued (minted) on the destination chain.
- **Lock and unlock** — A user locks tokens on the source chain, then unlocks the same native tokens from a liquidity pool on the destination chain. These types of cross-chain bridges usually attract liquidity on both sides of the bridge through economic incentives such as revenue sharing.