Dijets: An Architectural Overview

Fareed S.

Dijets Inc.

Abstract. This paper provides an architectural overview of Dijets as a platform. For details on the economics of its native coin, labeled DJT, we guide the reader to the accompanying coin dynamics paper [3]. Disclosure: The information described in this paper is preliminary and subject to change at any time. Furthermore, this paper may contain “forward-looking statements.”

1 Introduction

Dijets is an extremely lightweight, high-performing, secure, and energy-efficient protocol for building decentralised applications ("dApps"), novel financial primitives, and new interoperable blockchains. Using its innovative DLT approach, the Dijets [2] protocol is creating a new crypto-native economy for frictionless real-world and digital asset exchange, composable financial application primitives and derivatives, Web 3.0 privacy-focused data and social applications.

This paper provides an architectural overview of the Dijets platform with the key focus on three key differentiators of the platform: the engine, the architectural model, and the governance mechanism.

1.1 Dijets Platform Objectives

Dijets is a high-performance, scalable, customisable, and secure blockchain platform. It targets four broad use cases:

- Redefine the utility and accessibility of Web 3.0, making it a viable and user-friendly platform for all.
- Build application-specific Permissioned and Permissionless blockchain deployments.
- Building and launching highly scalable and decentralized applications (Dapps).
- Enabling creation of arbitrarily complex digital assets with custom rules, covenants, and riders (smart assets).

The Dijets platform aspires to serve as a comprehensive framework for the creation, exchange, and management of digital assets.

Dijets architecture possesses unique properties that seamlessly merge innovative design with democratised control for invigorating Web 3.0 - the reimagined Internet experience. Several of the architecture's crucial characteristics are:

1.1.0.1 Unlimited Scalability Dijets employs a highly scalable architecture that is both robust and efficient. Its consensus mechanism is engineered to sustain a global network comprising potentially hundreds of millions of devices—ranging from low to high computational power—ensuring uninterrupted operation characterized by minimal latency and exceptional throughput. This design effectively supports a substantial number of transactions per second.

1.1.0.2 Security The security framework of Dijets is meticulously designed to fortify robustness. Traditional consensus mechanisms are typically vulnerable to attackers just beyond a certain threshold, and protocols like the Nakamoto [4] consensus model lack resilience when 51% of miners exhibit Byzantine behavior. Conversely, Dijets offers a compelling guarantee of safety even under duress, provided the adversarial influence remains beneath a system-adjustable parameter. Remarkably, it maintains safety assurances, albeit compromising liveness, even when adversaries constitute more than 51% of the network, marking it as a pioneering permissionless system with enhanced security assurances.

1.1.0.3 Decentralisation The architecture of Dijets epitomizes decentralization like never before, eschewing any form of centralized authority or influence. The system wholly supports multiple client implementations and cultivates an egalitarian ecosystem devoid of hierarchies among users, ensuring uniformity in treatment across miners, developers, and end-users alike.

1.1.0.4 Interoperability Dijets's design philosophy prioritizes universal and flexible support across various blockchain protocols and digital assets. It employs its native Dijets Token as a security facilitator and a unit of account. Importantly, the platform is agnostic to the value it hosts, facilitating the seamless integration and support of diverse blockchains. Its infrastructure is tailored to facilitate the adaptation of existing blockchains, the transfer of existing balances, the operation of multiple scripting languages and virtual machines, and the accommodation of various deployment scenarios.

1.1.0.5 Governance Dijets champion inclusivity by enabling universal participation in its network's validation and governance processes. Token holders are empowered with voting rights crucial for the determination of essential financial parameters and the strategic trajectory of system evolution, embodying a truly democratic governance model.

This architectural framework makes Dijets an ideal and adaptable platform for the future's diverse and dynamic digital asset ecosystems.

Outline The rest of this paper is broken down into four major sections. Section 2 outlines the details of the engine that powers the platform. Section 3 discusses the architectural model behind the platform, including ECC networks, virtual machines (DVMs), bootstrapping, membership, and staking. Section 4 explores various peripheral topics of interest, including potential optimisations & post-quantum cryptography. Finally, Section 5 concludes the paper by rehashing the key takeaways.

Naming Conventions The name of the platform is Dijets, and is typically referred to as "Dijets platform", and is interchangeable/synonymous with "Dijets Network", or – simply – Dijets. Codebases will be released using three numeric identifiers, labeled "v.[0-9].[0-9].[0-100]", where the first number identifies major releases, the second number identifies minor releases, and the third number identifies patches. The native token of Dijets platform is called "DJT".

2 Engine

Dijets core component which powers the platform is its consensus engine.

Background Distributed payments and – more generally – computation, require agreement between a set of machines. Therefore, consensus protocols, which enable a group of nodes to achieve agreement, lie at the heart of blockchains, as well as almost every deployed large-scale industrial distributed system. The topic has received extensive scrutiny for almost five decades, and that effort, to date, has yielded a number of protocols including the classical consensus protocols, which rely on all-to-all communication, Nakamoto consensus, which relies on proof-of-work mining coupled with the longest-chain-rule and the Gossip protocols such as Avalanche [5] which utilise metastability principles to reach consensus. While classical consensus protocols can have low latency and high throughput, they do not scale to large numbers of participants, nor are they robust in the presence of membership changes, which has relegated them mostly to permissioned and static deployments. Nakamoto consensus protocols on the other hand, are robust, but suffer from high confirmation latencies, low throughput, and require constant energy expenditure for their security. Protocols exploiting metastability principles have garnered attention for their unique consensus mechanism but they are not without limitations. The need for further optimisations and resilience against varied network conditions is apparent.

Dijets substitutes Proof of Work (PoW) with a random sampling mechanism that runs at network speed and that has every party adjust its preference to that of a (perceived) majority in the system. Dijets also differs from more traditional blockchains by forming a DAG of transactions instead of a chain. The protocol extends the foundational principles of the aforementioned Gossip protocols. By introducing adaptive sampling, the protocol dynamically adjusts the sampling rate based on real-time network conditions and workload, ensuring optimal performance. Robust sub-sampling incorporates redundancy and error-correction techniques to mitigate the effects of malicious actors and network anomalies. Additionally, enhanced validation techniques are employed to ensure that the consensus process remains both time-efficient and highly reliable. This approach is also supplemented with a finalisation condition that can avoid liveness risks and ensure safety with very low failure probability.

Protocol Properties and Mechanism Dijets is structured around its polling mechanism and operates by repeated sampling of the network. Each node polls a small, constant-sized, randomly chosen set of neighbors, and switches its proposal if a supermajority supports a different value. Samples are repeated until convergence is reached, which happens rapidly in normal operations. In a nutshell, party Pi repeatedly selects a transaction T and sends a *query* about it to k randomly selected parties in the network. If a majority of them send a positive reply, the query is successful and the transaction contributes to the security of other transactions. Otherwise, the transaction is still processed but does not contribute to the security of any other transactions. Then the party selects a new transaction and repeats the procedure. A bounded number of such polls may execute concurrently. Throughout this work the terms “poll” and “query” are interchangeable.

We elucidate the mechanism of operation with an example of a payload transaction: Whenever a user submits a payload transaction tx to the network, the user actually submits it through a party Pi . Then, Pi randomly selects a number of leaf nodes from a part of the DAG known as the virtuous frontier; these are the leaf nodes that are not part of any conflicting set. Party Pi then extends tx with references to the selected nodes and thereby creates a transaction T from the payload transaction tx. Next, Pi sends a Query message with T to k randomly, according to stake, chosen parties in the network and waits for their replies in the form of *Vote* messages. When a party receives a query for T and if T and its ancestors are preferred, then the party replies with a positive vote. The answer to this query depends exclusively on the status of T and its ancestors according to the local view of the party that replies.

While the core mechanism of the operation is quite simple, Dijets protocol leads to highly desirable system dynamics that makes it an excellent choice for large-scale deployment.

- *Permissionless and Robust.* Many blockchain projects employ classical consensus protocols and therefore require full membership knowledge. Knowing the entire set of participants is sufficiently simple in closed, permissioned systems, but becomes increasingly hard in open, decentralized networks. This limitation imposes high security risks to existing incumbents employing such protocols. In contrast, Dijets protocol maintains high safety guarantees even when there are well quantified discrepancies between the network views of any two nodes. Validators of Dijets protocol enjoy the ability to validate without continuous full membership knowledge making Dijets extremely robust and highly suitable for public blockchains.
- *Scalable and Decentralised.* A core feature of Dijets is its ability to scale without incurring fundamental tradeoffs. The protocol can scale to tens of thousands or millions of nodes, without delegation to subsets of validators. Dijets enjoys the best-in-class system decentralisation, allowing every node to fully validate. First-hand continuous participation has deep implications for the security of the system. In almost every proof-of-stake protocol that attempts to scale to a large participant set, the typical mode of operation is

to enable scaling by delegating validation to a subcommittee. Naturally, this implies that the security of the system is now precisely as high as the corruption cost of the subcommittee. Moreover, subcommittees are subject to cartel formation.

- *Adaptive.* Unlike other voting-based systems, Dijets protocol achieves higher performance when the adversary is small, and remains highly resilient even under large attacks.
- *Low Latency.* Most blockchains today are unable to support business applications, such as trading or daily retail payments. It is simply unfeasible to wait minutes, or even hours, for confirmation of transactions. Therefore, one of the most important, and yet highly overlooked, properties of consensus protocols is the time to finality. Dijets protocol reaches finality typically in less than 1 second, which is significantly lower than both longest-chain protocols and sharded blockchains, both of which typically span finality minimum to a matter of minutes.
- *Asynchronously Safe.* Dijets protocol, unlike longest-chain protocols, does not require synchronicity to operate safely, and therefore prevent double-spends even in the face of network partitions. In Bitcoin, for example, if synchronicity assumption is violated, it is possible to operate to independent forks of the Bitcoin network for prolonged periods of time, which would invalidate any transactions once the forks heal.
- *High Throughput.* Dijets protocol, capable of building both, a linear chain or a DAG chain, can reach thousands of transactions per second (5400+ tps), while retaining full decentralisation. New blockchain solutions that claim high TPS typically trade off decentralisation and security and opt for more centralised and insecure consensus mechanisms. Some projects report numbers from highly controlled settings, thus misreporting true performance results. The reported numbers for DJT are taken directly from a real, fully implemented Dijets network geo-distributed across the globe on low-end machines. Higher performance results (10,000+) can be achieved through assuming higher bandwidth provisioning for each node and dedicated hardware for signature verification.

3 Dijets Platform - An Overview

This section provides an architectural overview of Dijets platform and discusses various implementation details. The platform separates three concerns: chains (and any assets built on top), execution environments, and deployments.

3.1 Architecture

ECCs / Subnetworks A subnetwork, or an enterprise consortia chain, is a dynamic set of validators working together to achieve consensus on the state of a set of blockchains. Each blockchain is validated by one subnetwork, and each subnetwork can validate arbitrarily many blockchains. A validator may be a member of arbitrarily many subnetworks. A subnet decides who may enter it, and may require that its constituent validators have certain properties. Dijets platform supports the creation and operation of arbitrarily many subnetworks. In order to create a new subnetwork/ECC or to join it, one must pay a fee denominated in DJT.

There are a number of advantages to Dijets Subnetwork/ECC model:

- If a validator is not interested in any of the blockchains in a given subnetwork, it can simply choose not to join that subnetwork. This reduces network traffic, as well as the computational resources required of validators. This is in contrast to other blockchain projects, in which every validator must validate every transaction, even the ones they are not interested in.
- Since a subnetwork or an ECC decides who can join and enter them, one can create private subnetworks which allows each blockchain in the subnetwork to be validated only by a set of trusted validators.
- One can create a subnetwork where each validator has certain properties. For example, one could create a subnetwork where each validator is located in a certain jurisdiction, or where each validator is bound by some real-world contract. This may be benificial for compliance reasons.

Chief Network The default subnetwork for Dijets is called the Chief Network which is validated by all validators. (That is, in order to validate any subnetwork or ECC, one must also validate the Chief Network.) The Chief Network validates a set of pre-defined blockchains, including the blockchain powered by Dijets native coin DJT.

Ternary Chain Ledgers The chief network is comprised of 'Ternary Chains' featuring 3 Primary interoperable Blockchains, each built from the ground up to serve its specific purpose within the Ecosystem.

– *Value Chain:* Dijets Value Chain (VC) acts as the primary conductor of value exchange within the Ecosystem. The VC also functions as a decentralised platform for creating and trading real world assets/resources as smart digital assets (for example, equity stakes, bonds, shareholdings etc). These smart digital assets can be configured with a set of rules that govern their behavior upon creation.
– *Utility Chain:* Dijets Utility Chain allows for the creation of smart contracts using Utility Chain's API. The UC is an instance of Ethereum [6] Virtual Machine powered by Dijets protocol making it interoperable with other active primary chains on Dijets. This means that any asset (ERC20, ERC721 etc) can be bridged to Dijets Network to take advantage of its sub-seecond finality and extremely low and flat transaction fees. Ethereum developers can quickly build on Dijets as Solidity is fully compatible with the Utility Chain.
– *Method Chain:* The Method Chain is the chain management layer for blockchains and their metadata in Dijets. It coordinates validators, keeps track of active primary and child chains, and enables the creation of new enteprise and private blockchains.

Wallet Address Prefixes Wallet address in Dijets Ternary Chain is denoted by the prefix as follows:

Ternary Chain	Mainnet Address	Testnet Address
Value Chain	V-dijets....	V-lothar....
Method Chain	M-dijets....	M-lothar....
Utility Chain	0x......	0x......

Dijets Virtual Machine Each blockchain is an instance of a Dijets Virtual Machine (DVM.) A DVM is a blueprint for a blockchain, much like a class is a blueprint for an object in an object-oriented programming language. The interface, state and behavior of a blockchain is defined by the DVM that the blockchain runs. The following properties of a blockchain, and other, are defined by a DVM:

– The data persisted to disk.
– The APIs exposed by the blockchain and their endpoints.
– The contents of a block.
– The state transition that occurs when a block is accepted.

A blockchain is described as "using" or "operating on" a specific Dijets Virtual Machine (DVM). When establishing a new blockchain, the DVM, it will run is explicitly chosen, along with the initial, or "genesis," state of the blockchain, which sets its starting conditions. This DVM can either be one that already exists, making it possible to leverage pre-built functionality, or it can be newly developed, allowing for customized features to be coded by developers based on unique requirements.

Multiple blockchains can operate using the same DVM, meaning the same foundational system and logic can support numerous separate chains. However, even if they run on the identical DVM, each blockchain remains entirely independent. This independence means that each chain maintains its own state—its distinct data, transactions, and overall history—ensuring that changes or actions on one blockchain do not directly affect others, even if they share the same DVM. This framework allows for diverse applications and ecosystems to be built on similar technical foundations without interdependency between individual blockchains.

3.2 Bootstrapping

Participation in Dijets begins with bootstrapping. When you run a Dijets node, the bootstrapping process occurs in three stages:

1. Connection to Beacon Nodes
2. Network & State Discovery
3. Become a Validator

We elucidate on the three processes below.

Connecting to Beacon Nodes Any networked system of peers that operates without a permissioned or hard-coded set of identities requires some mechanism for peer discovery. In peer-to-peer file sharing networks, a set of trackers are used. In crypto networks, a typical mechanism is the use of DNS seed nodes (which we refer to as beacon nodes), which comprise a set of well-defined seed-IP addresses from which other members of the network can be discovered. The role of DNS seed nodes is to provide useful information about the set of active participants in the system. The same mechanism is employed in Bitcoin Core [1], wherein the src/chainparams.cpp file of the source code holds a list of hard-coded seed nodes. The difference between

BTC and Dijets is that BTC requires just one correct DNS seed node (see Listing 1. below), whereas Dijets requires a simple majority of the beacon nodes to be correct. As an example, a new user may choose to bootstrap the network view through a set of well established and reputable exchanges, any one of which individually are not trusted. We note, however, that the set of bootstrap nodes does not need to be hard-coded or static, and can be provided by the user, though for ease of use, clients may provide a default setting that includes economically important actors, such as exchanges, with which clients wish to share a world view. There is no barrier to become a beacon node, therefore a set of beacon nodes can not dictate whether a node may or may not enter the network, since nodes can discover the latest network of Dijets peers by attaching to any set of beacon nodes.

```
        vSeeds.push_back(CDNSSeedData("bitcoin.sipa.be", "seed.bitcoin.sipa.be", true)); // Pieter Wuille, only supports x1, x5, x9, and xd
        vSeeds.push_back(CDNSSeedData("bluematt.me", "dnsseed.bluematt.me", true)); // Matt Corallo, only supports x9
        vSeeds.push_back(CDNSSeedData("dashjr.org", "dnsseed.bitcoin.dashjr.org")); // Luke Dashjr
        vSeeds.push_back(CDNSSeedData("bitcoinstats.com", "seed.bitcoinstats.com", true)); // Christian Decker, supports x1 - xf
        vSeeds.push_back(CDNSSeedData("xf2.org", "bitseed.xf2.org")); // Jeff Garzik
        vSeeds.push_back(CDNSSeedData("bitcoin.jonasschnelli.ch", "seed.bitcoin.jonasschnelli.ch", true)); // Jonas Schnelli, only supports x1, x5, x9, and xd
```

Listing 1. Hard-coded seed nodes in Bitcoin Core

Network and State Discovery Once connected to the beacon nodes, a node queries for the latest set of state transitions. We call this set of state transitions the accepted frontier. For a chain, the accepted frontier is the last accepted block. For a DAG, the accepted frontier is the set of vertices that are accepted, yet have no accepted children. After collecting the accepted frontiers from the beacon nodes, the state transitions that are accepted by a majority of the beacon nodes is defined to be accepted. The correct state is then extracted by synchronising with the sampled nodes. As long as there is a majority of correct nodes in the beacon node set, then the accepted state transitions must have been marked as accepted by at least one correct node.

Similarly the state discovery process is also used for network discovery. The membership set of the network is defined on the validator chain. Therefore, synchronising with the validator chain allows the node to discover the current set of validators.

3.3 Smart Contracts

Dijets supports standard Solidity-based smart contracts through its Ethereum virtual machine (EVM) instance namely The Utility Chain. The platform supports a wide range of powerful smart contract tools, including:

- Smart contracts with off-chain execution and on-chain verification.
- Smart contracts with parallel execution. Any smart contracts that do not operate on the same state in any subnetwork or ECC in Dijets will be able to execute in parallel.
- Solidity++ support. Solidity++ supports versioning, safe mathematics and fixed point arithmetic, an improved type system, compilation to LLVM, and just-in-time execution.
- The state transition that occurs when a block is accepted.

If a developer needs Ethereum Virtual Machine (EVM) compatibility but prefers to deploy smart contracts within a private, isolated environment, they can create a new subnetwork specifically for this purpose. This capability is central to how Dijets Platform supports functionality-driven sharding through the use of its subnetworks and ECCs, allowing developers to tailor environments to specific needs. Additionally, for developers needing to interact with smart contracts currently active on the Ethereum network, they can utilise the Athereum subnet — a dedicated subnetwork that acts as a "spoon" or a replica of Ethereum — enabling seamless integration with existing Ethereum-based smart contracts.

Furthermore Dijets allows developers seeking an alternative to the Ethereum execution environment, the ability and option to deploy their smart contracts through a subnetwork that runs on an entirely different execution model, such as DAML or WASM. This flexibility enables developers to choose environments best suited to their applications. Beyond just supporting virtual machines, subnetworks can also offer additional advanced features. For instance, subnetworks can enforce specific performance standards for validator nodes that need to handle smart contracts with higher storage demands over extended durations. Alternatively, they may include validators that are designed to keep contract states private, enhancing confidentiality. This versatility allows subnetworks to cater to a range of requirements, making them highly adaptable for diverse blockchain applications.

4 Discussion

4.1 Optimisations

Pruning Many blockchain platforms, especially those implementing Nakamoto consensus such as Bitcoin, suffer from perpetual state growth. This is because they have to store the entire history of transactions. However, in order for a blockchain to grow sustainably, it must be able to prune old history. This is especially important for blockchains that support high performance, like Dijets.

Unlike in Bitcoin (and similar protocols), where pruning is not possible per the algorithmic requirements, in comparison, Dijets nodes do not need to maintain parts of the DAG that are deep and highly committed. These nodes do not need to prove any past history to new bootstrapping nodes, and therefore simply have to store active state, i.e. the current balances, as well as uncommitted transactions.

Client Types Dijets can support three different types of clients: archival, full, and light. Archival nodes store the entire history of the Dijets subnetwork, the staking subnetwork, and the smart contract subnetwork, all the way to genesis, meaning that these nodes serve as bootstrapping nodes for new incoming nodes. Additionally these nodes may store the full history of other subnetworks for which they choose to be validators. Archival nodes are typically machines with high storage capabilities that are paid by other nodes when downloading old state. Full nodes, on the other hand, participate in validation, but instead of storing all of the history, they simply store the active state (e.g. current UTXO set). Finally, for those that simply need to interact securely with the network using the most minimal amount of resources, Dijets supports light clients which can prove that some transaction has been committed without needing to download or synchronise history. Light clients engage in the repeated sampling phase of the protocol to ensure safe commitment and network wide consensus. Therefore, light clients in Dijets provide the same security guarantees as full nodes.

Sharding Sharding is the process of partitioning various system resources in order to increase performance and reduce load. There are various types of sharding mechanisms. In network sharding, the set of participants is divided into separate subnetworks as to reduce algorithmic load; in state sharding, participants agree on storing and maintaining only specific subparts of the entire global state; lastly, in transaction sharding, participants agree to separate the processing of incoming transactions.

Sharding in Dijets exists through its subnetworks and ECCs functionality. Let's consider an example of two subnetworks/ECCs - one being an ECC for precious metals e.g. Gold and the other being a real-estate ECC. Dijets platform employs sharding for the two subnetworks to exist entirely in parallel. The subnetworks interact only when a user wishes to buy real-estate contracts using their gold holdings, at which point Dijets will enable an atomic swap between the two subnetworks.

Quantum Resistant Cryptography Interest in post-quantum cryptography has surged recently, driven by advancements in quantum computing technology and algorithms. The primary concern is that quantum computers possess the potential to compromise some of today's widely used cryptographic protocols, especially those involving digital signatures. In response, the Dijets network model is designed to support a diverse range of Virtual Machines (DVMs), including those that incorporate quantum-resistant cryptographic mechanisms. This flexible architecture allows for the integration of a quantum-resistant virtual machine capable of implementing robust digital signature methods.

As the quantum computing landscape evolves, we expect to see multiple types of digital signature schemes deployed, one of which includes quantum-resistant signatures based on Ring Learning with Errors (RLWE). Notably, the consensus mechanism in Dijets architecture is streamlined and does not rely on complex cryptography for its fundamental operations. This design makes it relatively straightforward to introduce a new virtual machine that can provide quantum-secure cryptographic primitives, enhancing the system's resilience against potential quantum-based security threats.

Egalitarian & Inclusive A frequent issue in permissionless cryptocurrencies is the tendency for wealth to concentrate among the already wealthy—a phenomenon often referred to as the "rich getting richer." This concern is especially relevant in Proof-of-Stake (PoS) systems, where, if implemented ineffectively, the process of wealth generation may disproportionately benefit those with significant initial stakes in the network. For example, in leader-based consensus protocols, a designated leader or a select subcommittee collects the majority of rewards, and the likelihood of being chosen to collect these rewards is often proportional to the amount of stake held. This structure can lead to compounding rewards for large stakeholders, reinforcing their financial position. Similarly, in Bitcoin's mining ecosystem, a "big get bigger" effect is observed, as larger miners benefit from reduced orphaned blocks and less wasted computational work, giving them an advantage over smaller miners.

In contrast, Dijets staking protocol is designed to prioritise fairness through an egalitarian approach to reward distribution. Every participant in the staking process is rewarded proportionally to their stake, ensuring an equitable distribution of rewards across the network. By facilitating broad participation in staking, Dijets system can support millions of participants who can engage in staking on equal terms. To encourage this inclusive environment, the governance council will set the minimum stake required for participation, with an initial low threshold to attract widespread involvement. This design approach eliminates the need for delegation, allowing even those with smaller allocations to participate directly in the staking protocol, further supporting Dijets commitment to inclusion and equality within its network.

5 Conclusion

In this paper, we presented the architecture of the Dijets platform. Unlike many current blockchain platforms, which rely on classical consensus protocols that limit scalability or employ Nakamoto-style consensus, which can be inefficient and costly to maintain, Dijets is designed to be lightweight, fast, scalable, secure, and highly efficient. The platform's native token - DJT, plays a key role in securing the network and covering various infrastructure-related costs, while maintaining a simple, backward-compatible structure. Dijets unique architecture surpasses other solutions in its capacity to achieve high levels of decentralisation, resistance to attacks, and scalability to millions of nodes — all without requiring quorum-based or committee-driven election processes, thus imposing no restrictions on participation.

Dijets introduces significant innovations beyond its consensus engine and across the entire technology stack. It brings forward streamlined approaches to transaction management, governance, ECCs and other components that are often absent in competing platforms. A powerful governance model empowers each participant, giving them a continual voice in shaping the evolution of the protocol. This structure ensures that governance is dynamic and community-driven. Dijets is also built with high customisability in mind, supporting near-instant plug-and-play compatibility with existing blockchains, making integration simple and rapid.

References

1. Bitcoin: bitcoin/bitcoin (October 2018), https://github.com/bitcoin/bitcoin
2. Fareed, S.: Dijets: A refined sub-sampling blockchain protocol (2024), https://dijets.io/DijetsWhitepaper_v1_0.pdf
3. Fareed, S.: Economics of dijets native coin djt (2024), https://dijets.io/Dijets_Tokenomics.pdf
4. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008), https://bitcoin.org/bitcoin.pdf
5. Team, R.: Snowflake to avalanche: A novel metastable consensus protocol family for cryptocurrencies. (2018), https://ipfs.io/ipfs/QmUy4jh5mGNZvLkjies1RWM4YuvJh5o2FYopNPVYwrRVGV
6. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger (2014), https://ethereum.github.io/yellowpaper/paper.pdf