# Comparative Study of Data Availability Schemes in Various Blockchains

Pranjal Sarode
Chainscore Labs
pranjal@chainscore.finance

Prasad Kumkar
Chainscore Labs
prasad@chainscore.finance

***Abstract*—We make a comparative benchmarking study of currently polkadot's ELVES data availability protocol with emerging data availability solutions which includes Avail, Celestia, Espresso's-Tiramisu and NEAR's sharded DA.**

**This work compares various system parameters—including bandwidth, time, latency, block time, block size, robustness, and cost per megabyte of data availability—and extends the analysis to other emerging factors affecting performance and scalability.**

## I. Introduction

The Blockchain technology is a constantly evolving field that has garnered significant attention and value in recent years. However, despite its growing prominence, the decentralized systems are facing challenges in scaling Blockchains in terms of data availability. Data availability is a critically important aspect of blockchain technology. Data availability refers to the ability of a blockchain network to ensure that all necessary data is accessible and retrievable by all its participants. In a decentralized system, Where multiple nodes work together to validate and store data/transactions, ensuring that the data is avaliable, valid and accessible is quite a important factor for maintaing the integrity and decentralized nature of the network [1].

This study provides a comparative study among the latest available Data availability solutions/models in terms of bandwidth, time and other criteria to provide a bigger overview of pro's and con's of every model.

The Models included in study are as follows:

- Polkadot's ELVES
- Celestia
- Espresso Tiramisu
- NEAR
- Avail

The below is a breif short introduction of the models that will be discussed and analyzed in the study.

### A. Polkadot ELVES

Polkadot is a sharded L0 multichain network the enables cross-chain interoperability and scalablity [2]. It utilizes ELVES (Economic Last Validation Enforcement System) to ensure data availability and validity of the parachains. Polkadot uses a hybrid consensus mechanism which uses BABE for block production and GRANDPA for its block finality. ELVES uses Reed Solomon encoding to split the data into chunks and then disperse it all across the network.

Polkadot uses Nominated Proof of Stake (NPoS) consensus mechanism which allows token holders to nominate entrusted validators to secure the network [3] [4].

### B. Celestia

Celestia is one of the first mover in the modular Data Availability solution space playing a major role in L2 and rollups developments. Celestia as of now utilizes 2D-Reed Solomon Erasure coding to split data into chunks and then encode into a matrix extension Namespaced Merkel Trees are then used to ensure the retirval and validity of the data. The root of the NMT is then stored in the block header [5].

As of now Celestia uses Tendermint consensus mechanism which uses BFT (Byzantine Fault Tolerance) style mechanism to ensure the finality of the blocks [5].

### C. Espresso Tiramisu

Espresso-Tiramisu DA resolves the Data availability scaling issue with a three layred system below is the short overview of these three layers.

- Savoiardi (VID Layer) - Erause codes and stores data across all the nodes [6].
- Mascarpone (DA Committe Layer) - A Small elected Committee stores the full data and guarantees to efficiently recover data [6].
- Cocoa (CDN Layer) - Uploads the data on web2 based CDN solution for seamless and speedy data recovery [6].

Espresso utilizes Hotshot consensus which is a optimistically responsive, communication-efficient consensus protocol in a proof-of-stake setting that is resistant to bribing adversaries and scalable to large number of nodes [7].

### D. NEAR

NEAR provides a high speed DA solution which porvides high transaction volumes with cost-effectivenes. NEAR utilizes the nightshade sharding mechanism, which parallelizes the network into multiple shards.Each shard processes its own transactions allowing the network to handle a higher volume of transactions approx  100,000 TPS [8].

As of now NEAR uses sharding-based proof of stake consensus mechanism.NEAR also implements unique validator elections to ensure security and decentralization of the network

### *E. Avail*

Avail DA helps blockchains scale by providing an abundance of data availability capacity. Its modular design scales data availability capacity with demand, and transaction data can be cryptographically verified quickly by anyone running an Avail light client [9].Avail utilizes Erasure coding, KZG commitments along with light client to ensure its data availability.

As for consensus Avail uses BABE/GRANDPA hybrid consensus used by polkadot for block production and finality. Avail also provides Application Specific Data Retrieval (ASDR) this helps rollups to fetch and decode their own blobs even tho the block might contain many app's data. [10].

**TABLE I:** Comprehensive Comparison of Data Availability Solutions (Data sourced from network documentation: [2], [11], [6], [8], [10], [12])

| **Feature** | **Polkadot ELVES** | **Celestia** | **Espresso Tiramisu** | **NEAR** | **Avail** |
|---|---|---|---|---|---|
| **Consensus & Architecture** | | | | | |
| Consensus Mechanism | BABE + GRANDPA(PoS) | Tendermint BFT | Permissionless PoS | Nightshade Sharding | BABE + GRANDPA(PoS) |
| DA Model | Reed-Solomon Erasure Coding | 2D Reed-Solomon + Namespaced Merkle Trees | Three-layer Tiramisu System | Sharding-based Parallelization | KZG + 2D Reed-Solomon Erasure Coding |
| Native Token | DOT | TIA | - | NEAR | AVAIL |
| Total Validators | 600 | 100 | 100 | 500 | 105 |
| **Performance Metrics** | | | | | |
| Block Time | $\sim$20s | $\sim$6s | $\sim$5s | $\sim$0.6s | $\sim$20s |
| Block Size | 5 MB | 8 MB | 1 MB | 4 MB | 4 MB |
| TPS | 10 | 0.48 | 0.8 | 53 | 420 |
| Max Throughput | $\sim$40 MB/s | $\sim$6 MB/s | $\sim$5 MB/s | $\sim$16MB/s | $\sim$0.2 MB/s |
| Finality | 6–30s | 5–15s | 1–2s | 1–1.2s | 20–40s |
| **Economic & Security** | | | | | |
| Cost per MB | N/A | $\sim$\$0.08 | N/A | 100kb per NEAR token | $\sim$\$0.0173 |
| Nakamoto Coefficient | 174 | 9 | N/A | 10 | 34 |
| **Key Features** | | | | | |
| Unique Advantages | Cross-chain interoperability via parachains | First modular DA solution for rollups | Web2 CDN integration (Cocoa layer) (High speed retrieval) | High-speed sharding 100k TPS | Application Specific Data Retrieval (ASDR) |

## II. METHODOLOGY

In this section, we detail the methods and approaches we will be using in our research. The methodology is structured as follows:

1) **Benchmarking**
   a) Design
   b) Performance
   c) Worst Case Analysis
   d) Efficiency
2) **Security Assumptions**
3) **Validator Costing**

### A. *Benchmarking*

*1) **Design**:* For benchmarking design a code module will be developed to evaluate the performance of all the data availability solutions under the study.

*a) Polkadot:* For benchmarking of polkadot's ELVES we are using **Polkadot's mainnet**. The current test code implements:

- System remarks for submission of data.
- Measuring the transaction fees
- Time taken for finalization of the block.

[We are also planning to use polkadot's on-chain telemetry (telemetry.polkadot.io) and logs for benchmarking.]

*b) Celestia:* For benchmarking Celestia's data availability solution, we utilize their **RPC** to submit and retrieve data blobs from the DA layer with their namespaces.

[We will also use the **Cosmos SDK** to interact with Celestia nodes and perform various operations such as data submission, retrieval, and verification.]

*c) Espresso's Tiramisu:* For Espresso's Tiramisu, we utilized their three-layered data availability architecture through their **API** endpoints. As of now, our testing framework implements:

- Data availability metrics through **API** endpoints (query.main.net.espresso.network)
- Block analysis including TPS calculations and size measurements

[We are also planning to set up a local testnet node of Espresso and use their SDKs for more in-depth benchmarking.]

*d) NEAR:* For benchmarking NEAR's sharded data availability, we use **NEAR's mainnet** environments. Our benchmarking setup uses near-api and a custom smart contract deployed with near-sdk, which allows storing and retrieving base64 encoded data blobs on-chain. We record the transaction time, retrieval latency, and cost per MB to compare NEAR's blob storage performance with other DA systems.

*e) Avail:* For Avail's benchmarking we have currently utilized avails JavaScript **SDK** and Python implementations. The code framework includes:

- Block bloat testing with automatic payload size adjustment
- Telemetry probing for network metrics
- Data submission through **AppID**-based categorization
- Block retrieval and verification mechanisms

*2) **Performance**:*

*a) **Throughput Analysis**:* This section will analyze the maximum throughput capabilities of different DA solutions under various network conditions/data sizes.

- **Test Parameters:** Time taken, data sizes
- **Metrics:** MB/s

**TABLE II:** Comparative Throughput Analysis of DA Solutions

| Protocol | Throughput (MB/s) |
|---|---|
| Polkadot | X |
| Celestia | X |
| Espresso | X |
| NEAR | X |
| Avail | X |

*b) **TPS**:* This section analyzes the transactions per second (TPS) capabilities of different DA solutions.

- **Test Parameters:** Network conditions, data sizes, transactions count
- **Metrics:** TPS

**TABLE III:** Comparative TPS Analysis of DA Solutions

| Protocol | TPS |
|---|---|
| Polkadot | X |
| Celestia | X |
| Espresso | X |
| NEAR | X |
| Avail | X |

*c) **Cost per MB**:* This section analyzes the cost per megabyte (MB) of data processed by different DA solutions.

- **Test Parameters:** Data sizes, Cost in native token conversion to USD
- **Metrics:** $/MB

**TABLE IV:** Comparative $/MB Analysis of DA Solutions

| Protocol | Cost ($/MB) |
|---|---|
| Polkadot | X |
| Celestia | X |
| Espresso | X |
| NEAR | X |
| Avail | X |

*d) **Latency**:* This section will analyze the maximum time taken by the DA solution to finalize a block.

- **Test Parameters:** Network conditions, data sizes, node count
- **Metrics:** seconds (s)

**TABLE V:** Comparative Latency Analysis of DA Solutions

| Protocol | Latency (s) |
|---|---|
| Polkadot | X |
| Celestia | X |
| Espresso | X |
| NEAR | X |
| Avail | X |

*e) **Max Block Size**:* This section analyzes the maximum block size capabilities of different DA solutions.

- **Test Parameters:** Data sizes.
- **Metrics:** Block size (MB)

**TABLE VI:** Comparative Block Size Analysis of DA Solutions

| Protocol | Block Size (MB) |
|---|---|
| Polkadot | X |
| Celestia | X |
| Espresso | X |
| NEAR | X |
| Avail | X |

*f) **Data Retrieval**:* This section analyzes the maximum data retrieval time capabilities of different DA solutions i.e. the time taken to retrieve a specific size/piece of data from the network.

- **Test Parameters:** Network conditions, data sizes.
- **Metrics:** Retrieval time (s)

**TABLE VII:** Data Retrieval Latency Analysis of DA Solutions

| Protocol | Retrieval Latency (s) |
|---|---|
| Polkadot | X |
| Celestia | X |
| Espresso | X |
| NEAR | X |
| Avail | X |

*3) **Worst Case Analysis**:* In this section, we will evaluate the worst-case performance scenarios for each DA solution. This includes analyzing how each solution handles extreme conditions such as high network latency, large data loads, and potential attacks or failures within the network.

*a) **Cost per MB (Inclusion Phase)**:* This sub-subsection evaluates the cost per megabyte (MB) incurred during worst-case data inclusion scenarios for each DA solution. We measure how gas fee surges, and payload sizes affect the overall inclusion cost.

- **Test Parameters:** Data size (1KB–1MB), gas/fee volatility

**TABLE VIII:** Worst-Case Cost per MB During Data Inclusion

| Protocol | Worst-Case Cost ($/MB) |
|---|---|
| Polkadot | X |
| Celestia | X |
| Espresso | X |
| NEAR | X |
| Avail | X |

- **Metrics:** Cost per MB ($/MB)

*b) **Latency (Worst Case)**:* This measures the maximum inclusion time under stress conditions (e.g., full block utilization).

- **Test Parameters:** Network latency, data size, node count
- **Metrics:** Latency (seconds)
- **Finality Impact:** Evaluates how consensus finality duration contributes to overall latency, including delayed confirmations or resistance under high-load conditions.

**TABLE IX:** Worst-Case Data Latency

| Protocol | Latency (s) |
|---|---|
| Polkadot | X |
| Celestia | X |
| Espresso | X |
| NEAR | X |
| Avail | X |

*c) **Data Retrieval (Worst Case)**:* We measure the time required to fetch data blobs under adverse conditions.

- **Test Parameters:** Network latency, node failures, retrieval requests per second
- **Metrics:** Data Retrieval Time (s)
- **Node Failure Impact:** Evaluates retrieval degradation under partial or complete node unavailability, including re-routing latency and redundancy recovery performance.

**TABLE X:** Worst-Case Data Retrieval Latency

| Protocol | Retrieval Latency (s) |
|---|---|
| Polkadot | X |
| Celestia | X |
| Espresso | X |
| NEAR | X |
| Avail | X |

*4) **Efficiency**:* This section focuses on comparative efficiency based on best and worst metrics of throughput, TPS, and cost per MB results. Efficiency metrics are derived from normalized performance ratios under controlled network environments.

*a) **Data**:* This subsection evaluates data transmission and Latency efficiency across DA (Data Availability) solutions. It compares bandwidth utilization, encoding overhead, and redundancy management under variable network conditions. Key considerations include:

- **Latency:** Reflects the overall efficiency of data flow, capturing how quickly transactions move from submission to availability. Lower latency directly translates to higher throughput and better real-time performance across DA layers.

These metrics provide a foundation for comparing the raw data handling capability of each DA layer independent of consensus or proof generation complexity.

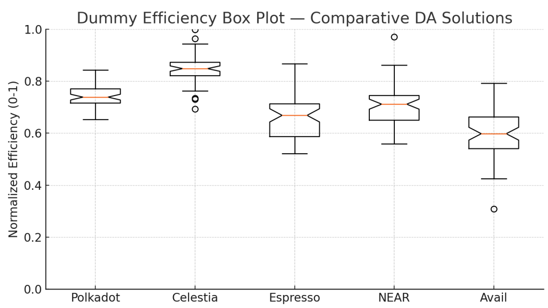


**Fig. 1:** Latency efficiency comparison across DA solutions

*b) **Storage**:* This subsection focuses on storage efficiency and persistence trade-offs once data becomes part of the network. It captures how network-level encoding, redundancy schemes, and node distribution affect the overall data footprint and storage cost.

Key considerations include:

- **Data Size (Post-Inclusion):** Analyzes how the effective data size increases once added to the network due to encoding, sharding, and availability sampling. This metric quantifies real storage growth compared to raw payload size, reflecting true on-network storage overhead.

**TABLE XI:** Data Storage and Transfer Metrics Across DA Solutions

| Protocol | Data Size |
|---|---|
| Polkadot | X |
| Celestia | X |
| Espresso | X |
| NEAR | X |
| Avail | X |

*c) **Proof**:* This subsection focuses on efficiency aspects related to proof generation, verification, and proof size optimization. Proof-related efficiency directly impacts network scalability and client-side verification costs. Key focus points include:

- **Proof Size:** Analysis of proof compactness and its relation to network bandwidth and storage.
- **Proof Computation:** Benchmarks proof generation latency and verification complexity.

**TABLE XII:** Proof Size and Computation Metrics Across DA Solutions

| Protocol | Proof Size (kB) | Proof Computation (ms) |
|---|---|---|
| Polkadot | X | X |
| Celestia | X | X |
| Espresso | X | X |
| NEAR | X | X |
| Avail | X | X |

This evaluation highlights the trade-offs between computational efficiency and proof soundness within different DA architectures.

## *B. Security Assumptions*

In this section, we outline the security assumptions made during the evaluation of the DA solutions. These assumptions are critical for understanding the context in which the solutions are being assessed and the potential vulnerabilities that may arise.

**TABLE XIII:** Security Assumptions of Major Data Availability Solutions

| **Network** | **Honest Majority Requirement** | **Validator Count (Approx.)** |
|---|---|---|
| **Polkadot (ELVES)** | $\geq \frac{2}{3}$ honest validators | 600 |
| **Celestia** | $\geq \frac{2}{3}$ honest for consensus; $\geq 1$ honest full/light node for DA sampling | 100 |
| **Espresso (Tiramisu)** | $\geq \frac{2}{3}$ honest DA committee member or fallback node | 100 |
| **NEAR** | $\geq \frac{2}{3}$ honest validators per shard | 500 |
| **Avail** | $\geq \frac{2}{3}$ honest for consensus; $\geq 1$ honest sampler for DA detection | 105 |

## *C. Validator Costing*

In this section, we will analyze the costs incurred by validators in maintaining and operating each DA solution. This includes hardware costs, bandwidth costs, and any other operational expenses associated with running a validator node.

**TABLE XIV:** Minimum or Recommended Hardware Requirements for DA/Validator Nodes (Data sourced from network documentation: [13], [14], [12], [15], [16])

| **Network** | **CPU** | **RAM** | **Storage** | **Bandwidth** | **Entry Fees** | **Cost** |
|---|---|---|---|---|---|---|
| Polkadot | 8 cores | 32 GB | 2 TB NVMe | 500 Mbps | $XXX | $XXX |
| Celestia | 8 cores | 64 GB | 8 TB NVMe | 1 Gbps | $XXX | $XXX |
| Espresso | 4 cores | 8 GB | 1 TB SSD | 100 Mbps | $XXX | $XXX |
| NEAR | 8 cores | 32 GB | 3 TB NVMe | - | $XXX | $XXX |
| Avail | 4 cores | 8 GB | 20 GB SSD | 50 Mbps | $XXX | $XXX |

# REFERENCES


[1] M. B. Saif, S. Migliorini, and F. Spoto, "A survey on data availability in layer 2 blockchain rollups: Open challenges and future improvements," *Future Internet*, vol. 16, 2024.

[2] Polkadot, "Polkadot protocol architecture," https://docs.polkadot.com/polkadot-protocol/architecture/polkadot-chain/overview., 2025, accessed: 2025-09-22.

[3] J. Burdges, A. Cevallos, P. Czaban, R. Habermeier, S. Hosseini, F. Lama, H. Kilinc, X. Luo, F. Shirazi, A. Stewart, and G. Wood, "Overview of polkadot and its design considerations," 2020.

[4] J. Burdges and A. Cevallos, "Efficient execution auditing for blockchains under byzantine assumptions," https://eprint.iacr.org/2024/961, 2019, https://eprint.iacr.org/2024/961.

[5] M. Al-Bassam, "Lazyledger: A distributed data availability ledger with client-side smart contracts," arXiv preprint arXiv:1905.09274, 2019, https://arxiv.org/abs/1905.09274.

[6] Espresso-Systems, "The espresso sequencer: Hotshot consensus and triamisu data availability," *Avail Project Documentation*, vol. 1, no. 1, 2023.

[7] E. Documentation, "Espresso documentation," https://docs.espressosys.com/network/, 2025, accessed: 2025-10-11.

[8] N. Protocol, "The near white paper," https://pages.near.org/papers/the-official-near-white-paper/, 2020, accessed: 2025-09-16.

[9] Avail-Team, "Avail: A unifying blockchain network," *Avail Project Documentation*, vol. 1, no. 1, 2024.

[10] Avail, "Avail: The data availability blockchain," *Avail Project Documentation*, vol. 1, no. 2.1, 2024.

[11] Gate.io, "A comparative study of celestia and near," https://www.gate.com/learn/articles/a-comparative-study-of-celestia-and-near/6143, 2024, accessed: 2025-09-22.

[12] E. S. Documentation, "Running a mainnet-1 node," https://docs.espressosys.com/network/releases/mainnet-1/running-a-mainnet-1-node, 2025, accessed: 2025-10-11.

[13] P. Documentation, "Validator requirements and vps provider list," https://docs.polkadot.com/infrastructure/running-a-validator/requirements/#vps-provider-list, 2025, accessed: 2025-10-11.

[14] C. Docs, "Nodes overview," https://docs.celestia.org/how-to-guides/nodes-overview, 2025, accessed: 2025-10-11.

[15] N. Nodes, "Hardware requirements for a near validator," https://near-nodes.io/validator/hardware-validator, 2025, accessed: 2025-10-11.

[16] A. Project, "Overview: Run an avail full node," https://docs.availproject.org/da/operate-a-node/run-a-full-node/overview, 2025, accessed: 2025-10-11.