
Blockchain and Distributed
Computing Group
智能计算研究所
Institute of Intelligent Computing
山东大学
SHANDONG UNIVERSITY

FileDAG: A Multi-Version Decentralized Storage Networks Built on DAG-based Blockchains

Presenter: Minghui Xu

2023/6/8

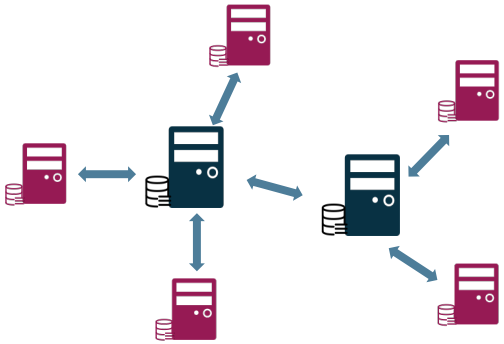
Outline

- Background and Challenges
- FileDAG: A Multi-Version Decentralized Storage Network Built on DAG-based Blockchain
- Evaluation and Discussion

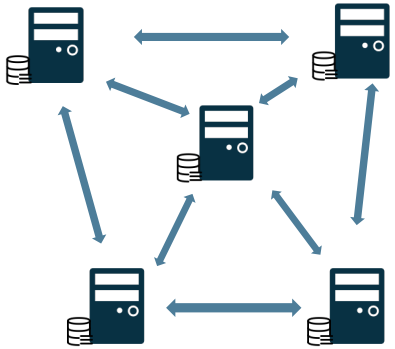
Background

➢ Decentralized Storage Network (DSN)

(1) Centralized Storage:
one node does everything



(2) Distributed Storage:
nodes distribute works to sub nodes



(3) Decentralized Storage Network:
nodes are connected to peers

- DSN uses blockchain to create a decentralized network of storage nodes that collaborate to store and retrieve data.
- Each node contributes storage space and processing power, and is rewarded with cryptocurrency.

Background

➢ Preliminaries

Clients: pay tokens to use storage services

Miners: pledge storage spaces to the DSN and provide storage services to earn tokens

- ❑ **Content Identifier (CID):** a hash-based unique identifier that maps to a data chunk.
- ❑ **Proof-of-Storage (PoS):** a cryptographic protocol that allows a client to efficiently verify the integrity of remotely stored data.

Background

➢ Basic Protocols

❑ A DSN protocol consists of three parts:

- **PUT:** Clients execute the PUT protocol to upload files to miners to store files, and obtain CIDs.
- **MANAGE:** Miners run the MANAGE protocol to control the available storage, audit the service, repair possible faults, etc.
- **GET:** Clients execute the GET protocol to send a CID to miners to request.

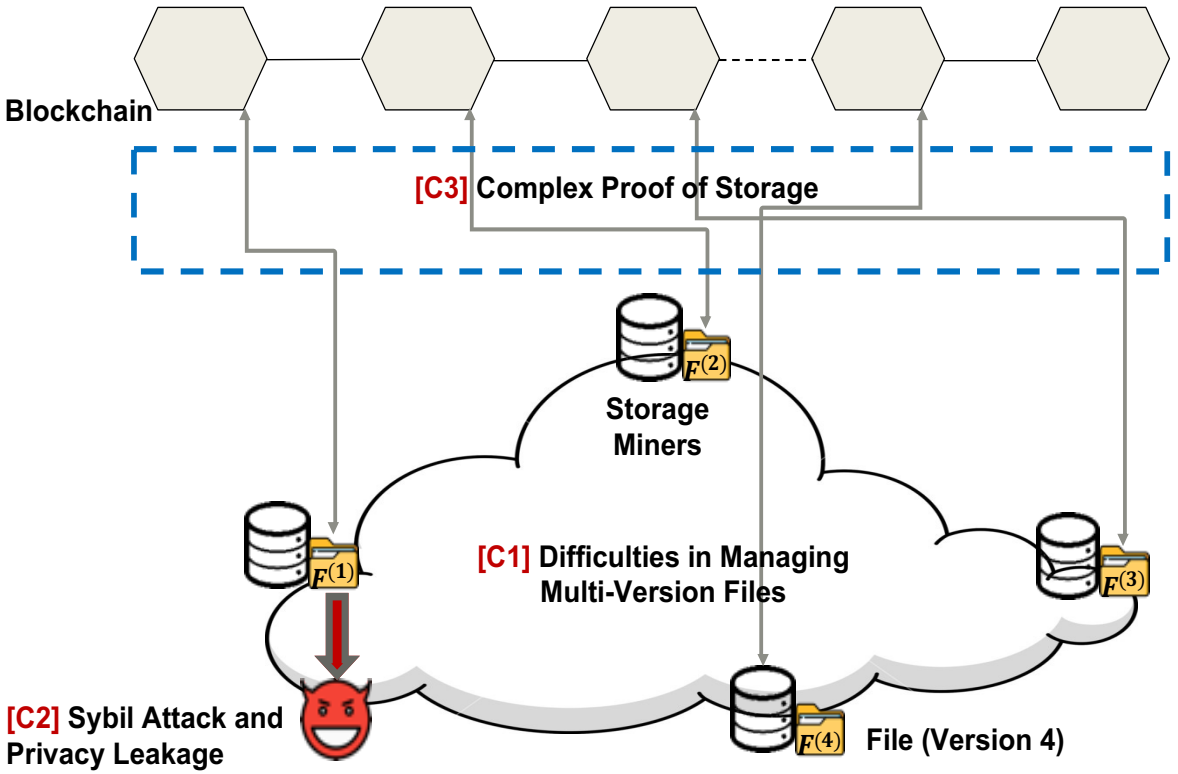
Background

- **Popular DSN Projects**

Challenge Statement

➢ Major challenges faced by DSNs

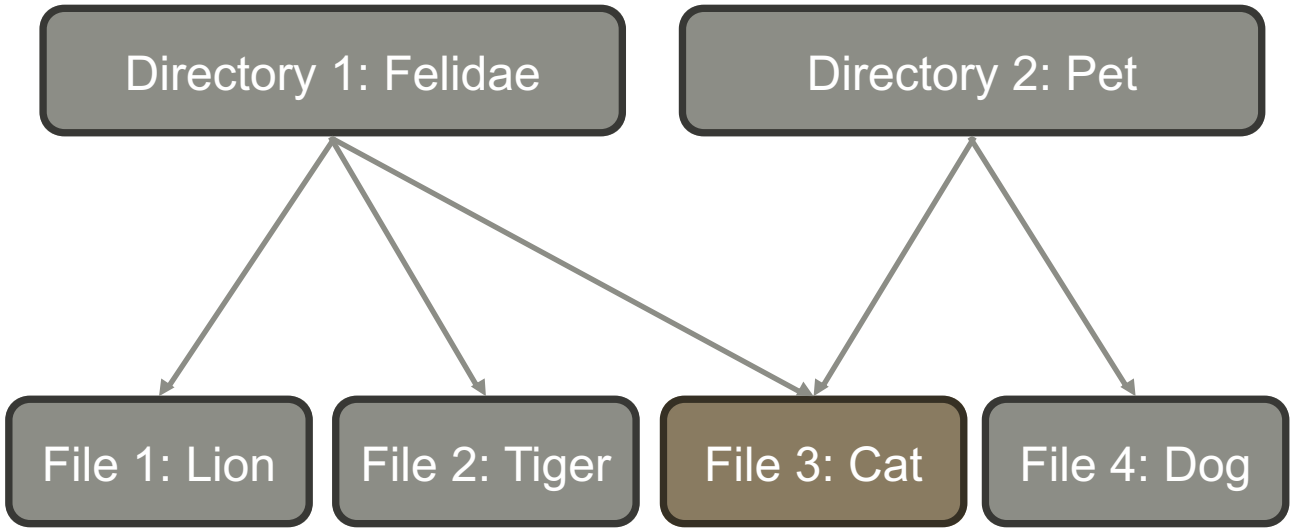
- ❑ **1 Difficulties in managing multi-version files.**
  - ➢ **1.1 Deduplication**
  - ➢ **1.2 Version control**
- ❑ 2. Sybil Attack and Privacy leakage.
- ❑ 3. Inefficient Proof of Storage (PoS).
  - ➢ High cost of generating proofs
  - ➢ High complexity of verifying multi-versioned files



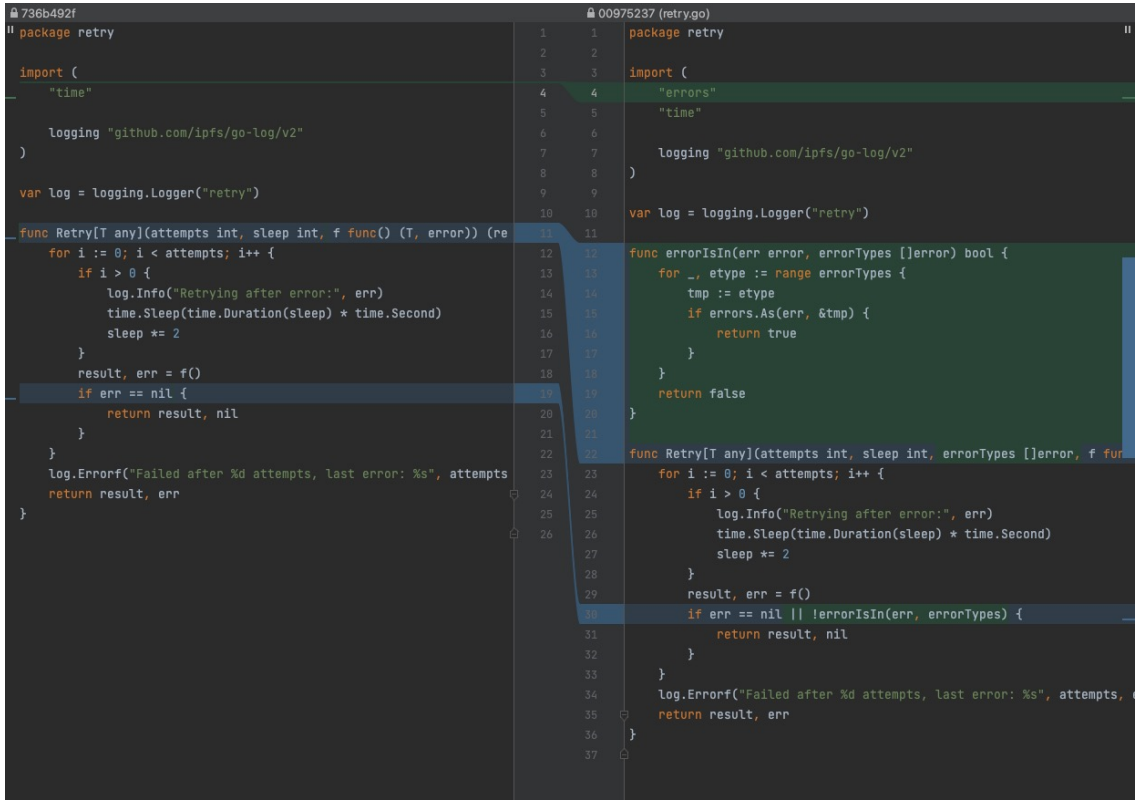
Challenges

➢ Difficulties in Managing Multi-Version Files

- Deduplication of Multi-Versioned Files
  - Directory-level deduplication fails to deduplicate between files



Directory-level deduplication

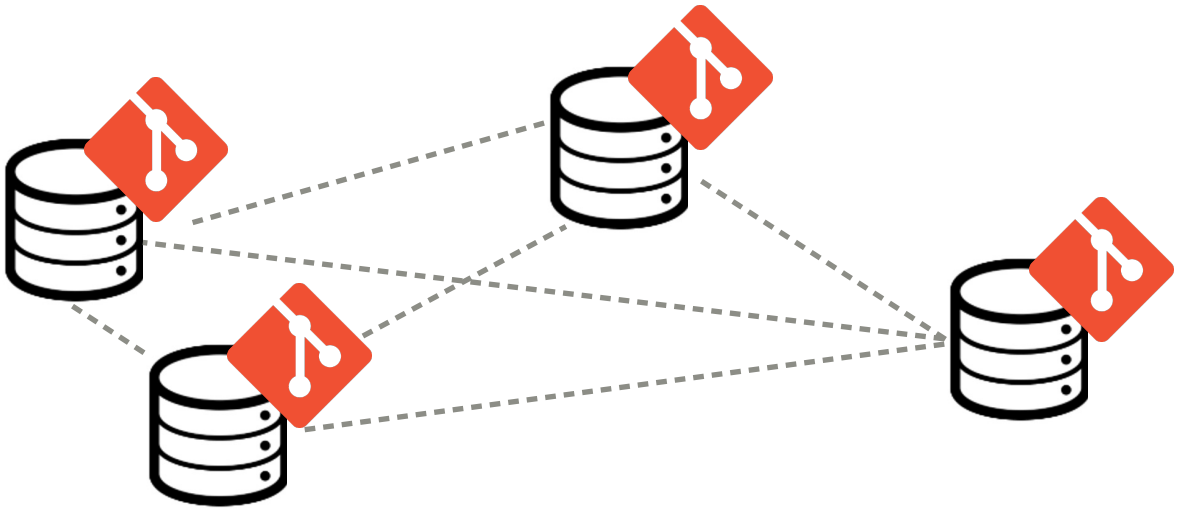


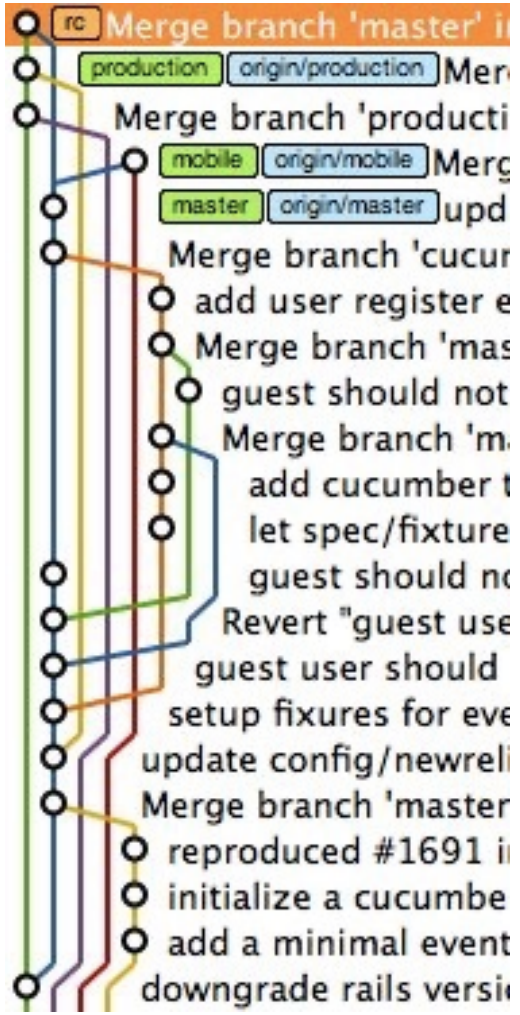
File-level deduplication

Challenges

➢ Difficulties in Managing Multi-Version Files

- Version Control with Blockchain
  - Version control is based on a version graph (also known as a history graph)
  - Version control requires using an additional database that stores redundant information about blockchain.





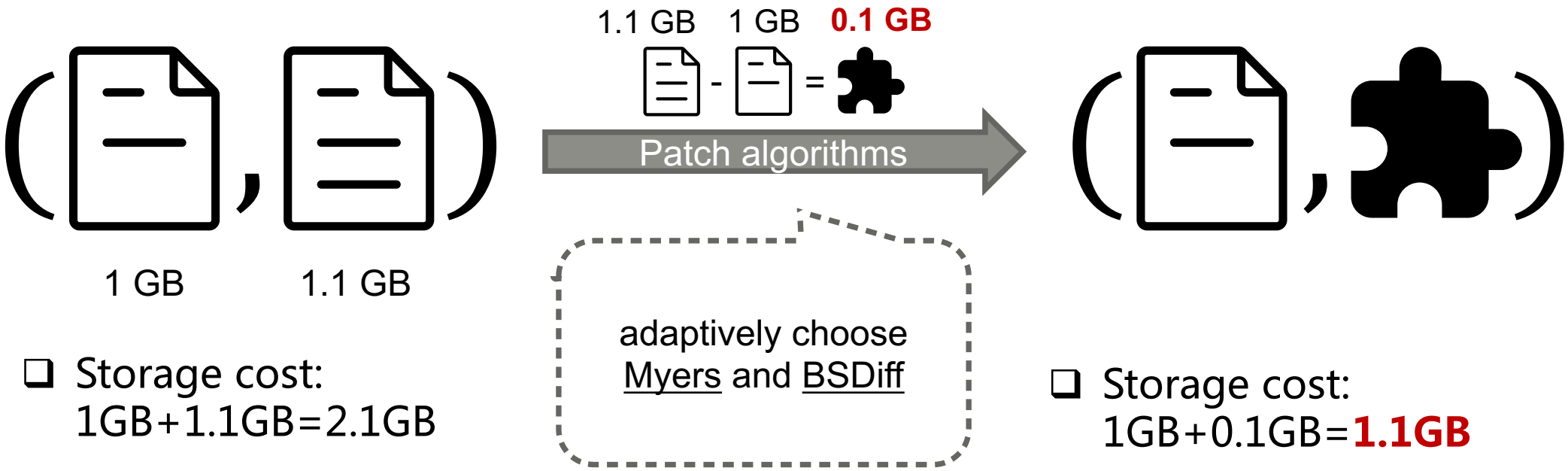
Part of a version graph

FileDAG

➢ Step 1: Increment Generation

Insight:

- Files (e.g., codebases, medical record, and softwares) usually change over time, resulting in multiple versions.
- Neighboring versions usually share some duplicate content.



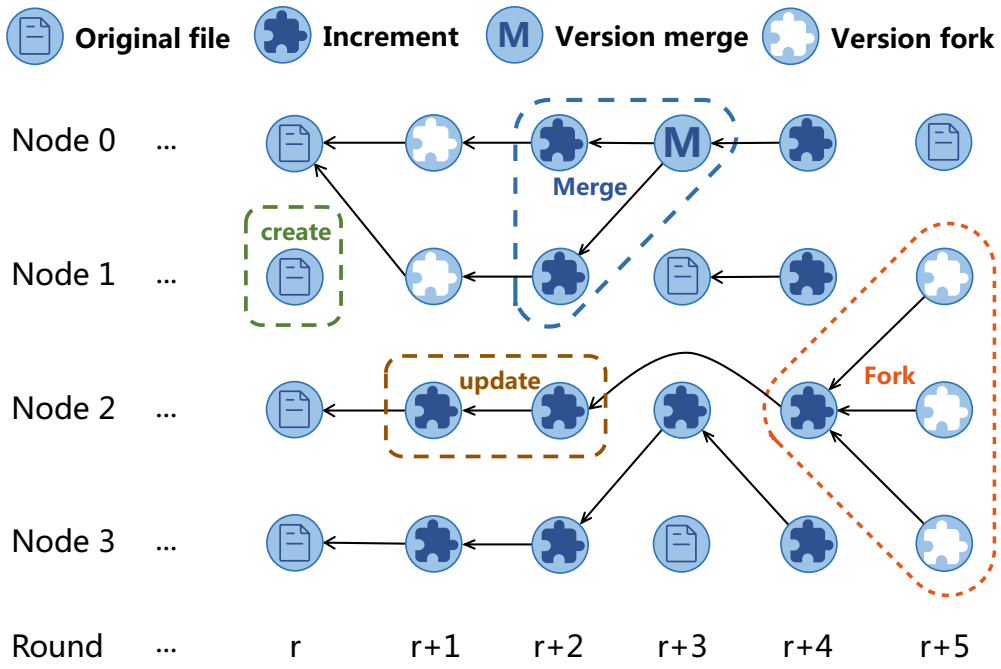
❑ Storage cost:
1GB+1.1GB=2.1GB

❑ Storage cost:
1GB+0.1GB=**1.1GB**

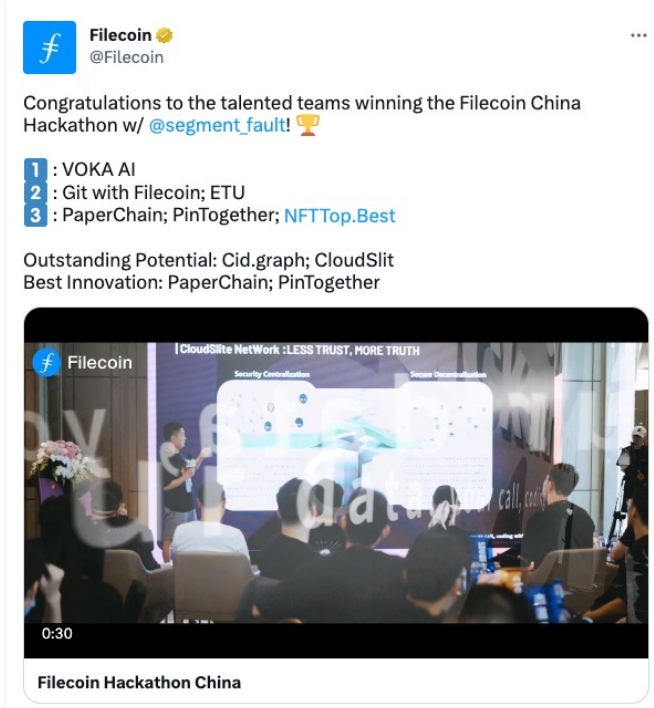
FileDAG

➢ Step 2.1: Build a version graph with a DAG-based Ledger

- ❑ FileDAG uses a DAG-based ledger to record version changes.
- ❑ Support four basic file operations: *create, update, merge, and fork*



"Layer-1" of DAG-based Ledger



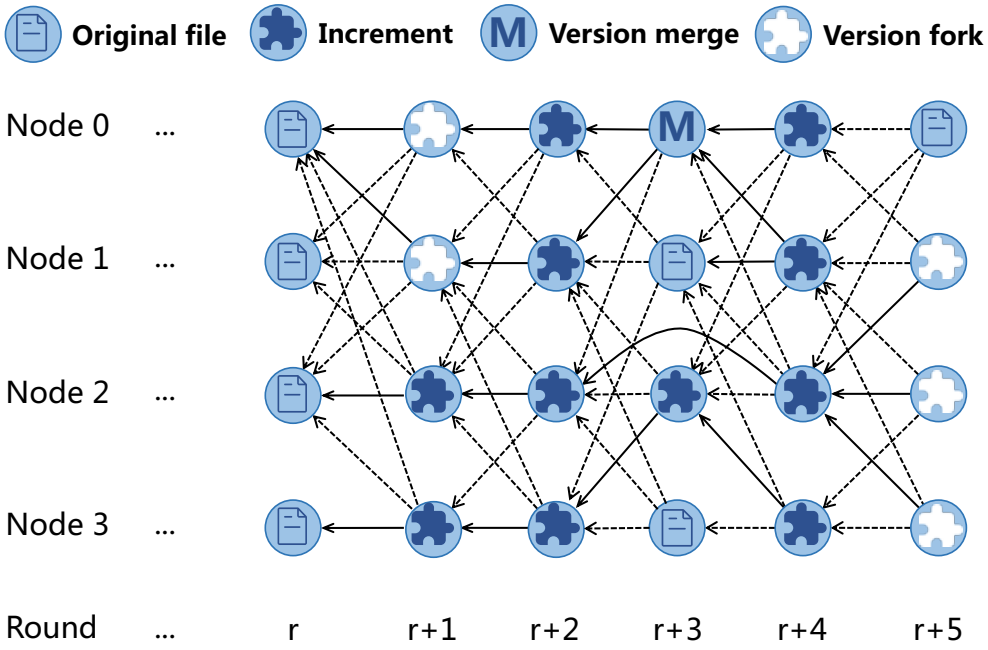
"Git with Filecoin" won the 2nd place in Filecoin China Hackthon

FileDAG

➢ Step 2.2: Make the DAG Ledger Consistent

- ❑ Reinforce the Layer-1 using DAG-Rider (a recent result on atomic broadcast) [1][2][3]
  - ✓ Add Layer-2 edges (dashed arrows) to achieve strong consistency and Byzantine fault tolerance.
- ❑ **Finally, we get a two-layer DAG-based ledger**
  - ✓ Layer-1: version graph (solid arrows)
  - ✓ Layer-2: consistent ledger (dashed arrows)

Version graph and blockchain ledger in one database!



Two-layer DAG-based Ledger

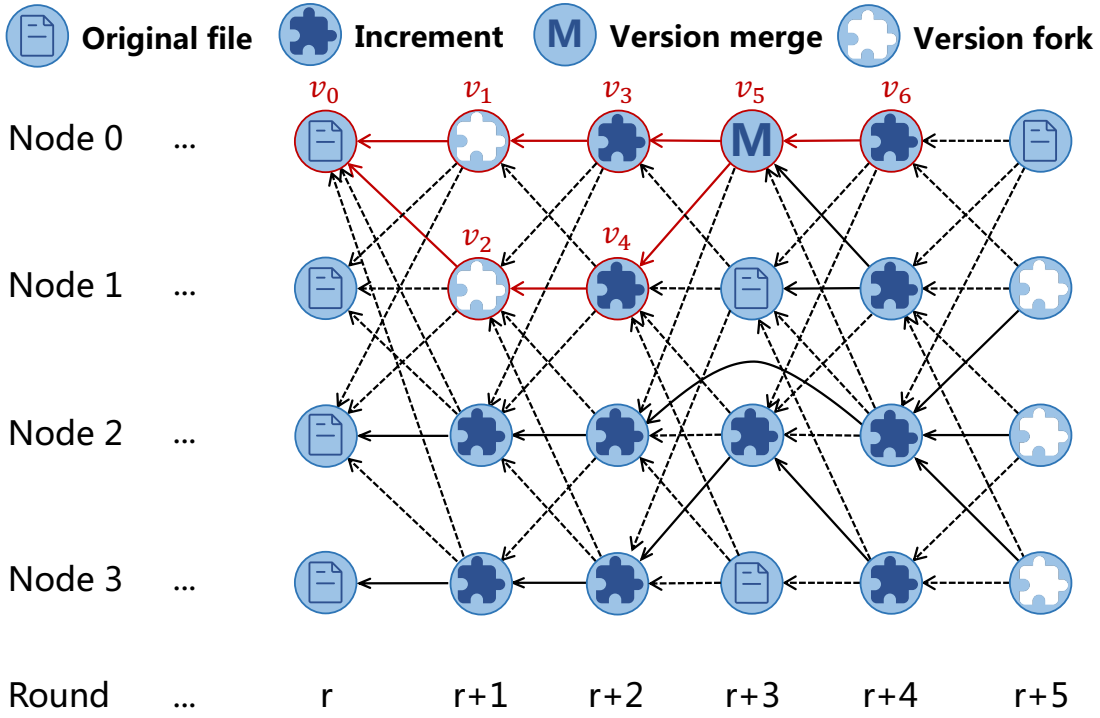
[1] Keidar, Idit, et al. "All You Need Is DAG." *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, ACM, 2021, pp. 165–75. *DOI.org (Crossref)*, https://doi.org/10.1145/3465084.3467905.

[2] Danezis, George, et al. *Narwhal and Tusk: A DAG-Based Mempool and Efficient BFT Consensus*. arXiv:2105.11827, arXiv, 16 Mar. 2022. *arXiv.org*, http://arxiv.org/abs/2105.11827.

[3] Spiegelman, Alexander, et al. *Bullshark: DAG BFT Protocols Made Practical*. arXiv:2201.05677, arXiv, 7 Sept. 2022. *arXiv.org*, http://arxiv.org/abs/2201.05677.

FileDAG

➢ Step 3: File Recovery

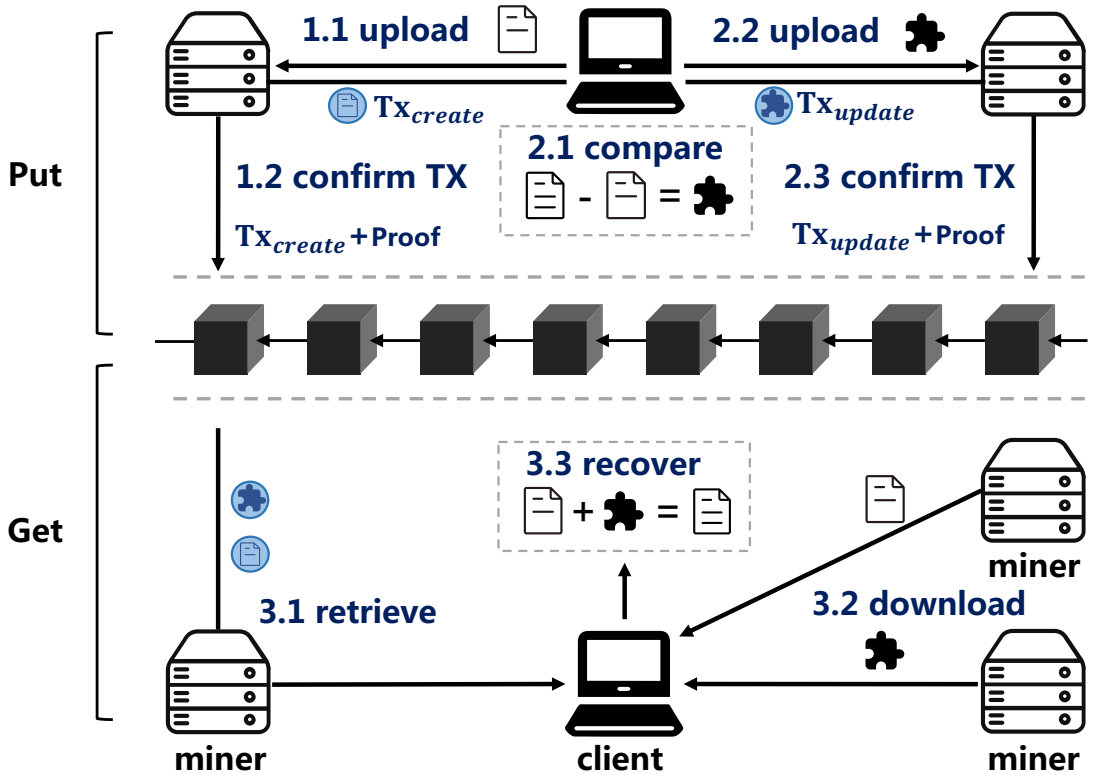


1. Request the CIDs of required increments and the original file.
2. Download increments and original file from the miners.
3. Patch increments together.

FileDAG

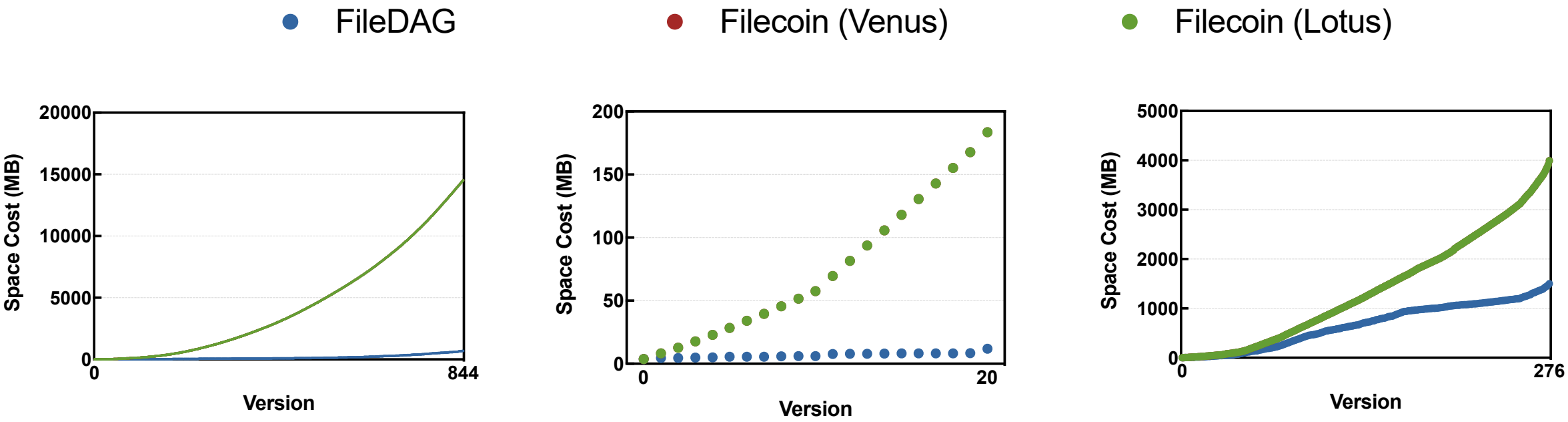
➢ Workflow of FileDAG

- ❑ PUT
  - ❑ Create: upload an original file f to a miner
  - ❑ Update:
    - ❑ Create a new version f' of f
    - ❑ Generate the increment between f and f'
    - ❑ Upload the increment to a miner
- ❑ GET
  - ❑ Retrieve: get the CID list of required increments and the original file
  - ❑ Download: download the increments and the original file from miners
  - ❑ Recover: patch them one-by-one together



FileDAG

➢ Experiment I: Storage Costs



Text (.c)

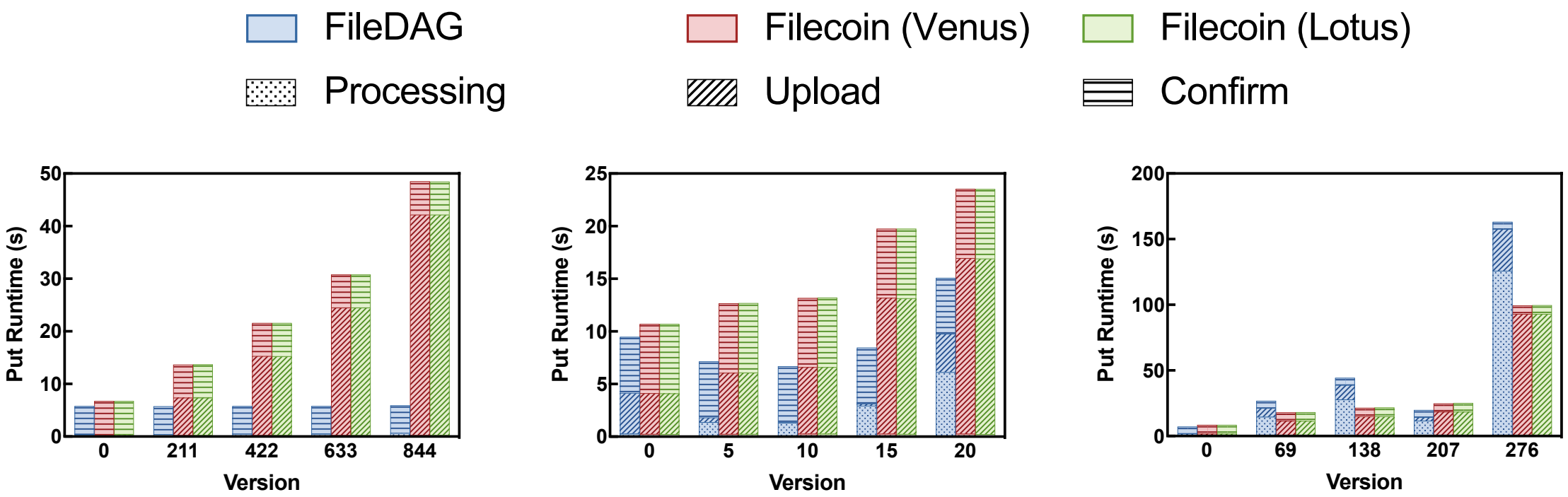
Multimedia (.pptx)

Binary (.apk)

The storage costs of Filecoin exhibit quadratic growth, while those of FileDAG show linear growth only.

FileDAG

➢ Experiment II: PUT runtime



Text (.c)

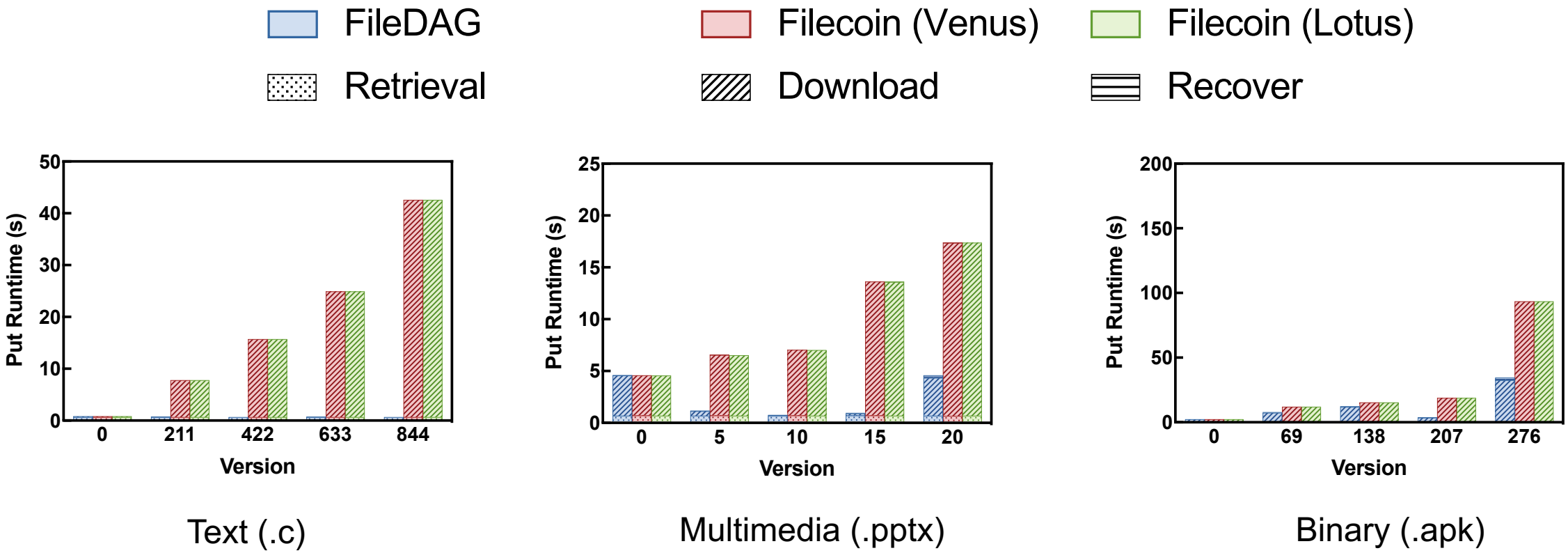
Multimedia (.pptx)

Binary (.apk)

Compared to Filecoin, FileDAG increases processing time but shortens upload time.

FileDAG

➢ Experiment III: GET runtime



Text (.c)

Multimedia (.pptx)

Binary (.apk)

FileDAG achieves low latency when executing the GET protocol mainly due to the significant decrease in download time.

FileDAG

➢ Experiment IV: Comprehensive Comparisons

	Text				Multimedia	APK (Binary)		
	IPLD	go-ipfs	ccf-deadlines	Git	PPT	Minecraft	WeChat	Netflix
# of versions	212	129	244	845	21	277	18	20
Average size (MB)	3.3	7.3	1.8	50.9	8.7	14.4	228.4	99.1
Storage (FileDAG) (MB)	**7.1**	**16.3**	**2.9**	**104.8**	**11.8**	**1503.0**	**3052.3**	**356.8**
Storage (Filecoin (Venus)) (MB)	712.8	985.3	483.4	43035.7	183.7	3994.4	4110.4	1981.7
Storage (Filecoin (Lotus)) (MB)	712.8	985.3	483.4	43035.7	183.7	3994.4	4110.4	1981.7
Storage (Sia) (MB)	642005.1	47889.1	11850.5	2592801.5	391.6	12839.2	78046.4	47644.7
Put runtime (FileDAG) (s)	**5.1**	**5.3**	**5.0**	**5.4**	**7.7**	22.2	760.5	**92.1**
Put runtime (Filecoin (Venus)) (s)	9.4	13.7	7.9	59.2	14.5	20.9	**232.3**	103.3
Put runtime (Filecoin (Lotus)) (s)	9.3	13.8	8.2	59.0	14.7	**20.3**	234.2	106.3
Put runtime (Sia) (s)	112.6	407.1	51.1	500.9	74.1	33.7	534.6	227.7
Get runtime (FileDAG) (s)	**0.8**	**1.0**	**0.7**	**1.0**	**1.3**	**6.6**	**182.8**	**19.2**
Get runtime (Filecoin (Venus)) (s)	4.2	8.3	2.6	53.5	9.9	15.7	232.7	103.2
Get runtime (Filecoin (Lotus)) (s)	4.0	8.4	2.7	53.9	9.6	15.5	230.9	104.1
Get runtime (Sia) (s)	8.3	24.1	5.0	106.6	20.1	43.8	671.2	313.2

FileDAG significantly reduces storage costs and GET runtime, while in most cases also reducing PUT runtime.

Thank you!
Q & A

Appendix

TABLE 1: Comparison of FileDAG with Existing DSNs

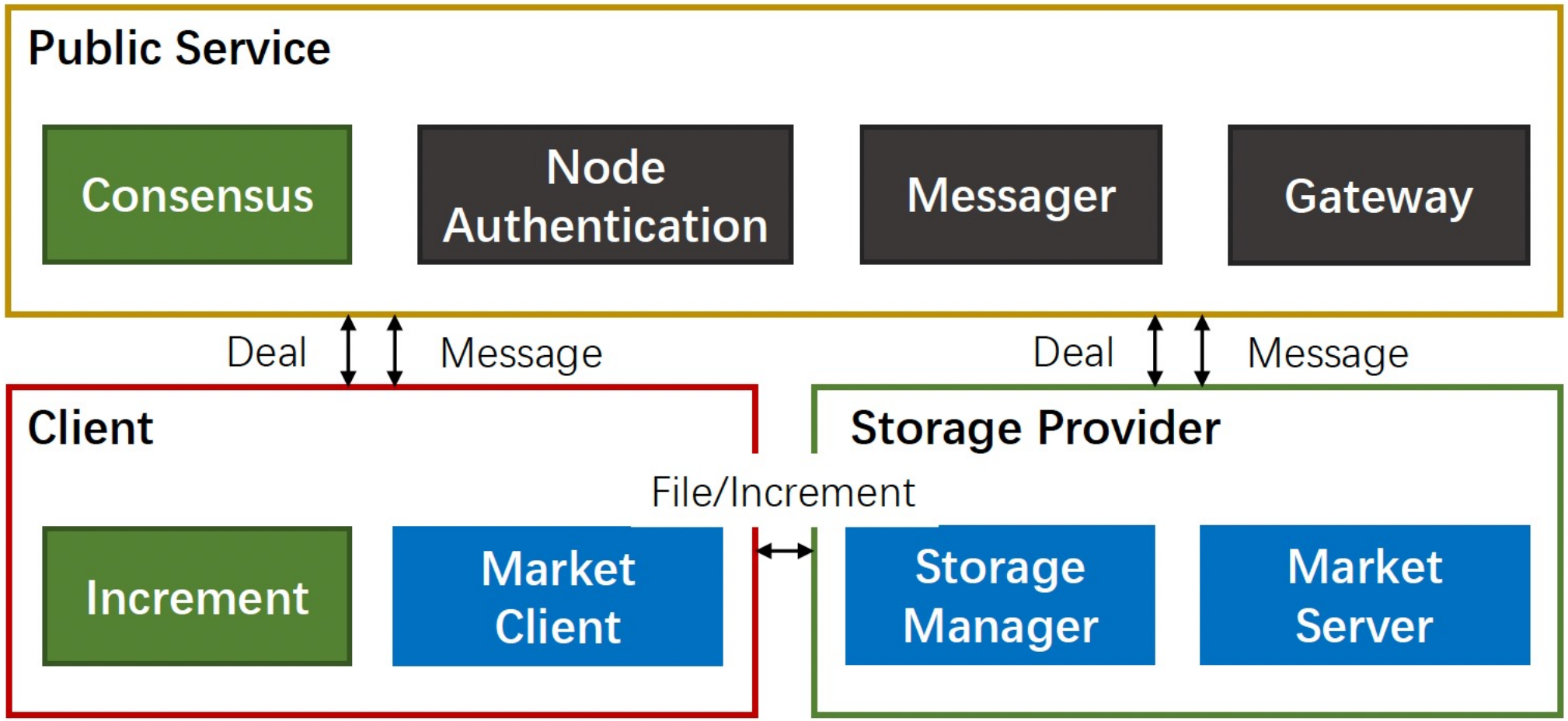
	Consensus Algorithm	Ledger	On-Chain DR	Deduplication Level
Filecoin [6]	Expected consensus	DAG (tipset)	✗	Directory
Storj [7]	PoW	Chain	✗	Directory
Sia [8]	PoW	Chain	✗	Directory
Swarm [9]	PoW	Chain	✗	Directory
FileDAG	DAG-Rider†	DAG	✓	File

DR Derivative Relationship
† Modified

Appendix

Definition 4.1. *(Consistency of multi-version DSN). For any version v of a file, an honest node can be convinced by the PoS proof that v is available in the FileDAG network only when v is indeed available; and if an honest node claims that v is available, then all other honest nodes claim the same.*

Appendix


Public Service
Consensus
Node Authentication
Messager
Gateway
Deal
Message
Deal
Message
Client
Storage Provider
File/Increment
Increment
Market Client
Storage Manager
Market Server