C I T H O R U M

Jam: How It Works

Plain-English and technical explanation of the Jam data plane, evidence model, and operating envelope.

Buyer · Cithorum · May 2026

Use the April 2026 canonical Cithorum source pack as the base of every claim in this document. Older decks are useful for product history, trial context, and integration roadmap, but aggressive old claims are not repeated as general product promises.

1. The One-Line Story

If your drive can write 1 TB per second, Jam writes 1 TB per second. Jam is engineered to be SSD-bound: no GPU bottleneck, no CPU bottleneck, no RAM bottleneck — only the SSD.

What that means in practice: Jam stops being a separate stage of the infrastructure and starts being a thin layer that lets the storage media you already own deliver more useful service per byte, per watt, per dollar. Most enterprises do not buy faster drives; they buy more drives. Jam removes the reason to buy more drives.

2. One-Page Summary

Jam is a software data plane for large, changing datasets. It sits as a layer below applications and storage services: it compresses, deduplicates, indexes, envelopes, stores, transfers, and reconstructs data while keeping the customer's existing hardware and application layer largely intact.

The canonical April 2026 product documents define Jam Networked Storage (also called RAMOPS) as a deterministic, keyless-at-rest networked storage layer. Its core architecture combines content-addressed immutable objects, an append-only deterministic layout, authenticated transport, quorum replication, and a reproducible benchmark methodology.

For Cithorum's buyers the immediate wedge is lower storage, bandwidth, and I/O cost; the longer-term move is indexed, evidence-aware data infrastructure for AI and knowledge-graph workflows.

3. Layman's Explanation

- Normal infrastructure stores many repeated versions of the same bytes. Jam learns the repeat structure and stores a smaller, lossless representation.
- Normal systems often move full files across the network. Jam can move smaller objects or deltas, then reconstruct the exact original when needed.
- Normal security systems store and manage keys. Jam derives session keys when needed from customer-controlled seed material and session state, so keys are not stored at rest.
- Normal AI and analytics pipelines waste time waiting for data. Jam's storage-layer optimisation and indexed access reduce I/O bottlenecks on compatible workloads.
- Jam's core path is software-first and CPU-sufficient. It does not require GPUs for the storage/compression/transport layer; frontier model training still uses GPUs where required.

4. Technical Model

At a technical level, Jam is not just a compression format. It is a storage and transport layer with deterministic object layout, cryptographic sessioning, and benchmarked compression/retrieval behaviour.

Stage	What happens	Source status
Ingest	Data enters Jam through local storage, block-device, S3/cloud, RAMOPS, SaaS Archiver, TechConnect, or future application adapters.	Source docs mention local storage, AWS S3, VSAN, block devices, TechConnect, and RAMOPS.
Object formation	Data is represented as content-addressed immutable objects, using SHA-256 object IDs and an append-only log.	Canonical in Jam-Networked-Storage.
Reduction	Jam applies lossless compression and deduplication. AI/semantic learning language appears in old decks; exact proprietary internals are not public.	Canonical envelope: up to 100× sparse; 35–75% typical.
Index/retrieve	Objects can be read by ID and reconstructed losslessly. Indexed paths can reduce load time for compatible AI/ML pipelines.	JSL exposes PutObject/GetObject; master reference permits up to 1000× data loading claim.
Security envelope	Session keys are derived on demand; AEAD protects transport; no session keys are stored at rest.	Canonical in MASTER-REFERENCE, MITM, JSL, SLA.
Replication	RAMOPS supports quorum writes, consistent reads within quorum, and eventual consistency across regions.	Canonical in Jam-Networked-Storage.

5. Product Modes

Product	Purpose	Important limits
Jam Networked Storage / RAMOPS	Deterministic, keyless-at-rest networked storage for commercial, sovereign, and air-gapped deployments.	Not a universal 100× compressor; performance depends on corpus.
Jam Standard Library / JSL	Versioned primitives for cryptographic and RAMOPS operations exposed to integrators.	Public surface is primitive/API level; internal algorithms remain proprietary.
M2M TechConnect	Authenticated machine-to-machine transport for OEM, embedded, and unattended workloads.	Managed SLA applies to hosted control plane; air-gapped uptime is customer operational responsibility.
Jam SaaS Archiver	Managed immutable archive with WORM retention, signed audit records, and retrieve APIs.	Cithorum is SOC 2 compliant; ISO 27001 / HIPAA / PCI / RBI / CERT-In / DPDP listed as aligned.
HyperRuntime / streaming runtime	Future concept for on-demand delivery of content as the runtime requests it.	Proof-of-concept / roadmap item, not canonical current capability.
Apache Spark / Atreides integration	Java/Scala API needed to run Jam inside Atreides IO's Apache Spark cluster.	Old deck describes this as a commercial requirement and funding use, not as completed integration.

6. Security in Plain English

Jam's security story is strongest when described simply: it does not rely on shipping or storing a static secret key next to the data. Instead, session keys are deterministically derived when required from seed material and session-transcript inputs, then discarded.

- Threat model: Dolev–Yao network attacker can observe, modify, inject, delay, and replay messages.

- Current primitives: ECDH/ECDSA over secp256k1, Ed25519 release/identity signing, AES-256-GCM or ChaCha20-Poly1305 AEAD, SHA-256/SHA-3/BLAKE3 hashing depending on purpose, HKDF-SHA-256 key derivation.
- Replay and tamper resistance: transcript binding, per-session nonces, monotonic sequence numbers, and AEAD integrity checks.
- Forward secrecy: session keys are derived per session and destroyed after session close; default rekey interval is configurable and documented as 15 minutes.
- PQC posture: FIPS 140/2 compliant build today; FIPS 140/3 submission planned Q3 2026; migration path to NIST FIPS 203 ML-KEM and FIPS 204 ML-DSA.

7. Performance Envelope

Aspect	Approved phrasing	Why
Compression	Up to 100× reduction in benchmark for sparse / highly compressible payloads; typical workloads 35–75% reduction.	Canonical April 2026 wording.
Backup tier vs rsync	On backup workloads, Jam observes ≈100× compression against rsync as the comparator (April 2026 live test, 123 GB → 1.18 GB; recording videopress.com/v/w4Z0jvUC).	Backup-tier corpus only; do not generalise.
Scale-AI average	5× average compression on the Scale-AI MVP dataset.	In master reference; scope validates reproducible range, not a single extreme number.
Rsync-style speed	Up to 400× faster than classical rsync on the Scale-AI MVP benchmark.	Approved only tied to benchmark conditions.
AI/ML loading	Up to 1000× faster data loading for compatible AI/ML pipelines.	Use with compatibility qualifier.
Storage footprint	Up to 50% reduction in typical enterprise storage footprint.	Canonical outward claim.
Compute / bandwidth cost	Up to 40% reduction in compute and bandwidth cost.	Canonical outward claim.

8. Real-World Trial and Demo Evidence

Source	Truth-based statement	How to use it
Atreides IO	50 GB big-data corpus compressed to 7 MB and reconstructed on a standard MacBook in under 5 minutes. Atreides wanted Apache Spark compatibility before licence execution.	Use as a proof-of-ability and integration roadmap, not as typical compression ratio.
SwissVault	Genome sequencing work: Jam produced 12% smaller FASTQ DNA sequences; user note says 22 minutes to align versus 2.5 hours for the next closest state-of-the-art aligner.	Use as named historical genomics benchmark; attach formal run sheet before scientific publication.
April 2026 backup tier vs rsync	123 GB → 1.18 GB on a backup workload; ≈100× compression vs rsync as the comparator (recording videopress.com/v/w4Z0jvUC).	Use as the canonical backup-tier benchmark.
Ubuntu VM	User note says an Ubuntu VM was stored 45% smaller via Jam versus alternatives.	Use as enterprise workload example; mark as trial-derived until formal benchmark pack is attached.
InterDigital	Legacy deck says InterDigital tested Jam for multimedia and game-engine	Use as historical trial context, not as a signed customer claim.

	file storage.	
NVIDIA Inception	NVIDIA Inception membership is supported by the Inception reception invitation, which states the email was obtained via the Inception membership list.	Use as programme membership / ecosystem validation, not as NVIDIA product certification.
NATO / UNITE-Brave	UNITE-Brave NATO portal access is supported by the NATO Communications and Information Agency email thread announcing the live portal and registration/matchmaking path.	Use as NATO/UNITE access or portal approval; avoid implying NATO technical endorsement.
RTX / Raytheon ecosystem	Supplier.io email says the profile was reviewed, approved, and added to the RTX ecosystem; the user note identifies this as Raytheon's partner-channel route.	Use as approved supplier/profile access to the RTX buyer ecosystem, not as an RTX purchase order.
M2M formal MRR	M2M Tech Connect is a formal paying client at $7K/month MRR. Lucas message confirms first invoice sent, Jam for Linux delivered, MacOS access given, and heartbeat/dashboard operation visible.	Use as commercial validation and recurring revenue proof.
NVIDIA / ZenLaunchpad / Atreides quotes	Quotes exist in Cithorum's proof register and old/company deck materials.	Use as credibility, not technical performance proof.

9. What Jam Is Not

- Not a new database. It exposes storage and object primitives but does not replace application databases by itself.
- Not a universal 100× compressor. The canonical typical workload range is 35–75%, with sparse / highly compressible payloads reaching the upper benchmark case. Backup-tier 100× requires a backup-tier corpus.
- Not a magic GPU replacement. The core data plane is GPU-free; frontier model training and high-end GPU workloads remain GPU workloads.
- Not a finished Apache Spark adapter today unless a separate build artefact says so. The Atreides path requires Java/Scala integration work.
- Not currently FIPS 140/3 certified. The source pack says FIPS 140/2 compliant today and FIPS 140/3 planned/submitted on a roadmap.

10. Funding-Relevant Roadmap

The cleanest technical roadmap from the source material is not to chase a bigger headline ratio. It is to productise integrations and validate benchmark reproducibility.

- Complete the independent 64 TB benchmark pack: five workload classes, reference rig, five runs per class, median and IQR, raw results preserved.
- Republish the April 2026 100× rsync backup-tier benchmark on a public corpus with raw logs.
- Build the Java/Scala adapter required for Apache Spark so Atreides-style cluster workloads can run Jam inside their existing analytics environment.
- Formalise SwissVault-style genomics evidence with reproducible FASTQ compression and alignment run sheets.
- Convert old HyperRuntime concept into a staged roadmap: streaming retrieval, runtime-requested content delivery, then application-facing SDKs.

- Continue security hardening: release-signing fingerprint publication, seed ceremony runbooks, HSM/TPM identity-key storage, and FIPS 140/3 process.

Source Boundary

These documents draw on the canonical Jam source archive maintained by Cithorum. They do not invent new performance numbers. Where a claim comes from an older deck rather than the April 2026 canonical source pack, it is labelled as historical, trial-derived, or requiring formal benchmark attachment before external publication.

- Cithorum Reworked-v2 MASTER-REFERENCE.md (April 17, 2026)
- Jam-Networked-Storage.docx, v2.1, April 2026
- Jam-Standard-Library.docx, v2.1, April 2026
- Jam-MITM-Hardness.docx, v2.1, April 2026
- Jam-Installation-Guide.docx, v2.1, April 2026
- Jam-SaaS-Archiver.docx, v2.1, April 2026
- Cithorum-Scale-AI-Scope-Requirements.docx, v2.1, April 2026
- Cithorum-Licensing-Proposal.docx, v2.1, April 2026
- Cithorum-M2M-TechConnect-SLA.docx, v2.1, April 2026
- Jam-Showcase.html, Reworked-v2 commercial showcase
- Cithorum Jam Company Deck.pdf, May 2025
- JAM - Legacy Deck.pdf, November 2024
- Direct user-supplied SwissVault/Ubuntu benchmark note, April 25, 2026
- M2M Partnership Proposal email thread PDF, April 28-29, 2026
- M2M Tech Connect formal $7K/month MRR confirmed by user, April 29, 2026; Lucas screenshot confirms first invoice sent and Jam for Linux delivered
- RTX / Supplier.io profile acceptance email PDF, April 29, 2026
- UNITE-Brave NATO portal email from NATO Communications and Information Agency PDF, April 29, 2026
- NVIDIA Inception reception invitation email PDF, April 29, 2026
- Cithorum + Jam backup-tier 100× rsync benchmark, April 2026 (recording: videopress.com/v/w4Z0jvUC)