



Making yourself audit-ready

with Verae DataCubes

Software is not a HIPAA, SOC 2, or ISO 27001 certificate.

A technical briefing on what the Verae DataCube Server Solution provides for data in transit and data at rest — and what every organization must still do itself.

Verae Inc

<https://www.verae.com>

Book a call at [verae.com](https://www.verae.com)

app.verae.com

Executive summary

Software alone does not make an organization HIPAA certified, SOC 2 attested, or ISO 27001 certified.

That sentence is the point of this document. It is also the sentence that is most often skipped when a vendor, an integrator, or an internal champion shows encryption, hashing, or a pentest PDF and treats the conversation as finished. It is not finished. Encryption is a control. A pentest is an evaluation of a component. A certificate, an attestation, or a covered-entity determination is a statement about **an organization** — its legal entity, its people, its written policies, its operating procedures, its internal controls, the evidence those controls produce, and the independent party that examined that evidence.

The Verae DataCube Solution gives an organization tools to **store, communicate, timestamp, verify, and audit** for compliance. Those tools are real, they are specific, and they are described in the chapters that follow. They are still only tools. To obtain HIPAA-aligned status as a covered entity or business associate with a defensible program, a SOC 2 Type I or Type II report, or an ISO 27001 certificate, the organization must still:

- write and live by **policies** (what the organization says it will do);
- operate **procedures** (how staff actually do it, every day);
- design and test **internal controls** (the checks that catch failure);
- retain **evidence** (logs, tickets, screenshots, signed approvals, restore tests, training records);
- engage an **independent auditor**, CPA firm, or ISO registrar, as the chosen program requires.

Verae cannot issue those certificates. Verae cannot sit in the organization's chair during an OCR investigation, a SOC 2 fieldwork week, or an ISO Stage 2 audit. What Verae can do — and what this briefing is written to make precise — is provide the **tools**, the **background**, and the **software infrastructure** that make it easier to implement the **technical portion** of those programs.

Contents

Executive summary	2
1 What Verae provides — and what it does not	7
1.1 The boundary	7
1.2 The tools	7
1.3 What the organization must still do	8
1.4 Why the distinction matters in an exam	8
2 The Verae DataCube Server Solution	9
2.1 Purpose	9
2.2 The DataCube as a container	9
2.3 Two information states, one operational picture	10
2.4 What “server” means in practice	10
2.5 What the server solution is not	10
3 Secure communications — data in transit	13
3.1 The problem	13
3.2 Point-to-point encryption	13
3.3 Visible routing	14
3.4 Error handling without leaking content	14
3.5 The public-key directory	14
3.6 What this does, and does not, satisfy	14
4 Encryption at rest — IPFS blocks and Peergos	17
4.1 The problem	17
4.2 Content-addressed blocks	17
4.3 Peergos on top of IPFS	18
4.4 Tamper-evident restore	18
4.5 Append-only at the cube layer	18
4.6 What a disk operator sees	19
4.7 What this does, and does not, satisfy	19

5	Global timestamped receipts	21
5.1	Why hashes are not enough by themselves	21
5.2	What a Verae receipt is	21
5.3	What is registered, and what is not	22
5.4	First registration wins	22
5.5	Sequence	22
5.6	Bundles	22
5.7	What a receipt does not prove	22
6	Peergos security evaluations in Europe	23
6.1	What was evaluated	23
6.2	2019 — Cure53, Berlin, Germany	23
6.3	2024 — Radically Open Security B.V., Amsterdam	24
6.4	How to present these reports to an auditor	24
6.5	Hosted Peergos versus self-hosted DataCubes	25
6.6	What “designed under funding from Cure53 / ROS” is not	25
7	Verae global timestamping — a cross-blockchain receipt	27
7.1	Lineage	27
7.2	What “cross-blockchain” means here	27
7.3	Two deployment patterns	28
7.4	The digital bundle	28
7.5	Proof of existence versus proof of custody	28
7.6	What this does, and does not, satisfy	29
8	Write-once Iceberg archive	31
8.1	The deletion problem	31
8.2	What “write-once Iceberg” means in this solution	31
8.3	How cubes move into the archive	32
8.4	Relation to classical WORM	32
8.5	What this does, and does not, satisfy	33
9	Architecture for an audit interview	35
9.1	The picture	35
9.2	Walkthrough, in the order an examiner usually asks	36
9.3	Identity planes	36
9.4	NATS remains untrusted	36
9.5	What to hand the examiner	36
10	BAAs, DPAs, and ciphertext without host keys	39
10.1	The technical fact	39
10.2	HIPAA — Business Associate	39
10.3	GDPR — processor versus technical measure	40
10.4	NATS operators	40
10.5	Verae as timestamping service	40
10.6	Component assurance versus the organization’s report	40
11	Audit-ready checklist	41
11.1	A. Scope and honesty	41

11.2 B. Data in transit	41
11.3 C. Data at rest	42
11.4 D. Timestamping	42
11.5 E. Write-once archive	42
11.6 F. Access and change	42
11.7 G. Contracts (counsel)	42
11.8 H. Independent examination of <i>this</i> organization	43
12 How to use this briefing	45
13 James H. Garfinkel	47
13.1 Name and sources	47
13.2 Role at Verae	47
13.3 FINRA BrokerCheck (CRD 5052743)	48
13.4 Education (public professional listings)	48
13.5 Why this biography is in the briefing	48
14 Stuart Haber	51
14.1 Role at Verae	51
14.2 The scientific work	51
14.3 Surety, 1994: first commercial blockchain	52
14.4 Other cryptographic publications	52
14.5 Why this biography is in the briefing	52
15 George Lambert	53
15.1 Role	53
15.2 Internet technical architect, from 1994	53
15.3 New Hampshire public office	54
15.4 Why this biography is in the briefing	54
16 Verae Inc — contact	55
16.1 Closing reminder	55

CHAPTER 1

What Verae provides — and what it does not

1.1 The boundary

Verae sells and operates **software infrastructure** and a **timestamping service**. Customers use that infrastructure to store objects, to send messages, to register hashes, to verify receipts, and to produce an audit trail of what their own systems did.

Verae does **not** sell a HIPAA certificate. HIPAA is a United States **statute** (and implementing regulations), not a product seal. A covered entity or business associate becomes “HIPAA compliant” only in the ordinary-language sense that it has performed a risk analysis, implemented the Security Rule’s required and addressable controls as reasonable and appropriate, executed Business Associate Agreements where required, trained its workforce, and can respond to OCR. No software vendor can substitute for that work.

Verae does **not** sell a SOC 2 report covering the customer’s organization. SOC 2 is an **attestation** by a licensed CPA firm against the AICPA Trust Services Criteria, for a named entity, over a named period, with a named system description. The customer’s auditor must still walk the customer’s controls.

Verae does **not** sell an ISO 27001 certificate covering the customer’s organization. ISO 27001 is a **management-system** standard. A registrar certifies that a named organization operates an Information Security Management System (ISMS) over a named scope. Installing a DataCube does not create an ISMS.

1.2 The tools

Within that boundary, the Verae DataCube Solution is built to make the **technical** work of those programs less painful. Concretely, it gives the organization the ability to:

Store. Digital objects — messages, images, documents, logs, model prompts, model outputs, attachments — can be written into an encrypted DataCube. The cube is a customer-controlled container. Verae is designed so that the **contents stay in the customer’s storage**. What leaves the customer’s perimeter for sealing is a **fingerprint** (a cryptographic hash), not the object.

Communicate. Endpoints exchange messages with point-to-point encryption. The body is unreadable to the router. Destination and subject remain visible because a network that cannot see a destination cannot route.

Timestamp. The first time a hash is registered, Verae records that registration against a time and a sequence. Later, anyone with the receipt and the object (or the object hash) can check that the object existed, in that exact bit pattern, no later than that time.

Verify. Verification does not require trusting Verae’s word, the customer’s archive administrator, or a hosting vendor. It requires recomputing a hash, checking a receipt, and — for stored objects — checking IPFS content identifiers. A regulator can be handed proof rather than a vendor assurance letter.

Audit. Admin history, inspect events, and message-handling summaries are written so that an examiner can see *that* an action occurred, *who* authorized it, and *which* object it touched, without the log itself becoming a second copy of the sensitive payload.

1.3 What the organization must still do

Those five verbs — store, communicate, timestamp, verify, audit — are the technical portion. The rest of a certification or attestation program is organizational:

Policies. Acceptable use, encryption, key custody, access control, retention, legal hold, incident response, vendor management, change management, and (where HIPAA applies) a sanctions policy and a named Privacy Officer and Security Officer.

Procedures. How a new hire is provisioned, how a leaver is deprovisioned, how a restore is tested, how a key is rotated, how a legal hold is placed on a cube, how an inspector is authorized under k-of-n, how a SOC evidence request is answered.

Internal controls. Dual control on inspect. Separation of the author of a change from the officers who approve it. Signed configuration so an unsigned file is rejected. Log-before-reveal so an inspect cannot happen in the dark. Hash-registry “first write wins” so a later rewrite cannot pretend to be the original.

Independent examination. A CPA firm (SOC 2), a registrar (ISO 27001), OCR or a HIPAA assessor, FINRA or SEC examination staff, or an internal audit function that is actually independent. They will ask for evidence that the controls ran during the period, not that a vendor has a nice architecture diagram.

1.4 Why the distinction matters in an exam

Examiners are trained to notice category errors. If a firm says “we are SOC 2 because our storage vendor was pentested in Berlin,” the next hour of the meeting is spent unwinding that claim. If a firm says “we use a component that was independently evaluated in the EU; here is the public report, labeled as a component evaluation; here is *our* system description, *our* key-custody procedure, *our* restore test from last quarter, and *our* Type II report,” the meeting can proceed.

This chapter exists so that nobody using this briefing makes the first speech. Verae provides tools, background, and software infrastructure. The organization provides the program. The auditor provides the opinion. Those three roles do not collapse into one product SKU.

CHAPTER 2

The Verae DataCube Server Solution

2.1 Purpose

A Verae DataCube Server Solution is the on-premises or customer-hosted assembly that gives an organization a place to put sensitive digital objects, a way to move them, a way to prove when they first existed, and a way to show an examiner that the bits have not been silently rewritten.

It is a **server solution** in the sense that an organization runs (or has run for it) a set of cooperating services: encrypted storage, a message fabric, a public-key directory, a timestamping link to Verae's central service, and an archival export path. It is not a single binary, and it is not a cloud folder with a padlock icon.

2.2 The DataCube as a container

A **DataCube** is a customer-controlled container for digital objects and for the metadata that makes those objects examinable. Typical contents include:

- the object bytes (a message body, a document, an image, a log extract, an AI prompt and completion, or any other digital media);
- **private metadata** that the organization needs but does not necessarily share (internal identifiers, matter numbers, legal-hold flags);
- **attached files** that travel with the object;
- an **internal chain** of hashes so that the cube itself has a history — each new write names the previous write.

Verae's public product description is consistent with this split: records stay in **storage the customer controls**; Verae seals a **fingerprint**. The server solution is the machinery that makes that split operational rather than rhetorical.

2.3 Two information states, one operational picture

Classical security training divides information into **data in transit** and **data at rest**. The DataCube Server Solution is built around that division.

In transit, the problem is an untrusted network. Messages must be readable at the destination and nowhere else along the path, yet the path must still be able to deliver them. Chapter 4 treats that problem in full: point-to-point encryption, visible routing, honest-but-curious brokers.

At rest, the problem is an untrusted disk, an untrusted backup operator, and an untrusted replica. Bytes must be stored so that a host who does not hold keys sees opaque blocks, and so that a restore can prove it brought back the same blocks that were written. Chapter 5 treats that problem in full: IPFS content-addressed ciphertext, Peergos hash verification, tamper-evident restore.

A third problem sits beside those two: **time**. Storage and transit prove confidentiality and integrity of *what* was stored or sent. They do not, by themselves, prove *when* it first existed, or that a later rewrite is not being offered as the original. Timestamped receipts (Chapter 6) and the cross-blockchain timestamping architecture (Chapter 8) address that.

2.4 What “server” means in practice

In a typical deployment the organization runs, or links:

- a **Peergos** instance (or equivalent crypttree client) that writes encrypted, content-addressed blocks;
- an **IPFS** layer that stores and replicates those blocks by hash;
- a **message fabric** (NATS in the reference deployment) that carries HPKE-sealed bodies with destinations in the clear;
- a **public-key directory** so every endpoint can find every other endpoint’s encryption key without a private-key leak;
- an **organizational timestamping node** that either syncs with Verae’s central timestamping server or is linked to it;
- an **admin-history** cube that records configuration changes as previous state, new state, and diff;
- an **Iceberg write-once export** that takes cubes out of any single party’s delete path (Chapter 9).

Each of those pieces can be drawn on a whiteboard in an audit interview. Each of them also has a failure mode that the organization’s procedures must name: lost keys, a mis-issued directory entry, a broker that drops messages, a replica that serves a wrong block (detected by hash), a timestamping link that is down, an archive job that did not run.

2.5 What the server solution is not

It is not a substitute for workforce training. It is not a substitute for a Business Associate Agreement analysis. It is not a substitute for access reviews. It is not, by itself, “the HIPAA control set” or “the SOC 2 system.” It is the **technical substrate** on which those controls can be implemented with less faith in honest administrators and more reliance on hashes, receipts, and keys the customer holds.

The following four chapters unpack the substrate: transit, rest, receipts, and the European evaluations of the Peergos storage layer.

CHAPTER 3

Secure communications — data in transit

3.1 The problem

A message that leaves one machine and arrives at another crosses infrastructure the endpoints do not own: routers, load balancers, message brokers, TLS terminators, packet-capture appliances, and people with legitimate operational access to those devices. Any of those parties can copy bits. The honest design question is not “will anyone see the packet” — they will — but “**what** will they see, and **what** will they be able to do with it.”

The Verae DataCube Server Solution answers that question with **point-to-point encryption of content** and an explicit admission that **routing must be visible**.

3.2 Point-to-point encryption

“Best in class” here is not a slogan; it names a concrete choice. Production content is sealed with **HPKE** (Hybrid Public Key Encryption, RFC 9180), in the HPKE-Base mode, using a suite such as X25519-HKDF-SHA256 with ChaCha20-Poly1305. Each endpoint has a key pair. The sender looks up the recipient’s **public** key in a directory that is visible to every E2E service. The sender seals the body to that public key. Only the holder of the matching **private** key can open it.

Consequences that matter in an audit interview:

- The **sender cannot reopen** the ciphertext after it is sealed unless the sender is also a recipient or has retained plaintext. A lookup identifier is enough to talk about the message later without retaining a second copy of the body.
- The **broker cannot open** the body. Possession of the wire image is possession of ciphertext.
- A **lab XOR** construction, if it exists in a codebase for experiments, is not a production algorithm. Production configurations reject it.

3.3 Visible routing

A network that cannot see a destination cannot deliver a message. The DataCube Server Solution therefore does **not** claim anonymous, metadata-free messaging. The following remain visible to the transport, by design:

- destination (the handle or address the router needs);
- subject or stream name (so the right service receives the envelope);
- sender handle or lookup identifier, when the protocol carries them for reply and error handling;
- approximate size and timing (any network sees these).

This is the **honest-but-curious broker** model. Curiosity is assumed. Honesty is assumed only in the narrow sense that the broker forwards what it is given; it is **not** trusted with content, and it is **not** trusted not to log destinations.

3.4 Error handling without leaking content

Failures have to be reported. A bounce that includes the original body would undo the encryption. The design therefore returns **error metadata**: an error code, a lookup identifier, a destination class — not the plaintext, and not a replay of the ciphertext into a log aggregator that is a second, weaker store.

Network Error Bundles can carry a sender-visible ciphertext and a system-visible ciphertext so that the right party can diagnose without broadcasting the payload to operators who should never see it.

3.5 The public-key directory

Point-to-point encryption is only as good as the lookup of public keys. The server solution publishes a **directory of public keys** so that availability of those keys is visible to all E2E services. Private keys do not belong in that directory. Private key files are mode **0600**, held on the endpoint or in an HSM, and are never returned by the public listing API.

An examiner can be shown the public listing. An examiner should never be given a private key.

3.6 What this does, and does not, satisfy

For HIPAA Security Rule addressable encryption of ePHI **in transit**, for SOC 2 CC6 cryptographic transmission, and for ISO 27001 Annex A transmission security, this design is the **technical control**: content is encrypted to the recipient, the path is untrusted, keys are endpoint-held.

It does **not** by itself satisfy:

- a rule that requires the organization to **inventory** every channel (personal devices, shadow SaaS, unapproved AI tools);
- a rule that requires **workforce sanctions** when someone bypasses the channel;
- a rule that requires **agreements** with the broker operator covering metadata that may still be personal data.

Those remain policies, procedures, and contracts. The software makes the approved channel strong. It cannot stop a person from using a weak channel instead. That is an organizational control, not a cryptographic one.

CHAPTER 4

Encryption at rest — IPFS blocks and Peergos

4.1 The problem

Bytes that are no longer moving still have to live somewhere: a disk, a replica, a backup, a cold archive. The people who operate those surfaces — hosting providers, backup operators, someone with a snapshot credential — are rarely the same people who are authorized to read the records. A design that stores plaintext on those surfaces is a design that makes every disk operator a reader. A design that stores ciphertext, but lets an operator swap a block without detection, is a design that fails integrity even if it keeps confidentiality.

The Verae DataCube Server Solution stores **data at rest** as **encrypted, content-addressed IPFS blocks**, with **Peergos** verifying hashes on the way in and the way out.

4.2 Content-addressed blocks

IPFS names a block by a cryptographic hash of its contents (a Content Identifier, or CID). Two properties follow immediately:

- If the bits change, the name changes. There is no such thing as “the same CID, different payload” under a collision-resistant hash.
- A client that asked for CID X and received bytes whose hash is not X **knows** it was given the wrong object.

That is the definition of a **tamper-evident** store. It does not prevent a hostile replica from refusing to serve a block (availability). It does prevent a hostile replica from silently serving a substitute and hoping nobody notices (integrity).

4.3 Peergos on top of IPFS

Peergos is not “IPFS with a folder UI.” It is an encrypted filesystem — a **crypttree** — whose nodes and file chunks are stored as IPFS blocks. The properties that matter for an audit-ready DataCube are:

- **Client-side encryption.** Symmetric keys for chunks live with the customer, not on the storage host. A host with disk access sees padded, encrypted chunks and random labels, not filenames, not sibling names, not a social graph.
- **Hash verification.** Peergos **verifies the hash** of every block it reads from or writes to IPFS. A restored cube is not “whatever was on the tape.” It is a set of CIDs re-fetched and re-checked.
- **Signed writes.** Updates are signed. An unsigned rewrite is not a valid Peergos write.
- **Independent chunk keys.** Chunks (up to 5 MiB, padded) are encrypted with random 256-bit keys (Salsa20-Poly1305 in the published design), not with convergent encryption that would leak equality of plaintext across users.
- **CHAMP** places encrypted crypttree nodes under random 32-byte labels, so the storage host cannot walk the directory tree by name.
- **Sharing** uses hybrid encryption (X25519 plus ML-KEM-1024 in current Peergos) so a capability can be given to another user without placing the chunk key on the server.

4.4 Tamper-evident restore

“Backup” in this architecture does not mean a second plaintext copy in a different building. It means:

1. the CIDs that constitute a cube are known;
2. the encrypted blocks for those CIDs exist on more than one replica;
3. a restore **re-fetches** the blocks and **re-verifies** the hashes;
4. the customer, holding keys, decrypts locally.

If step 3 fails, the restore fails closed. That is the opposite of a backup product that “helpfully” returns the last good plaintext it had. Helpfulness of that kind is how silent corruption and silent substitution enter an evidence set.

4.5 Append-only at the cube layer

Peergos itself uses signed updates and immutable blocks. The Verae DataCube adds an **application-level chain**: a JSONL history in which each record hashes the previous record. Dual hashing (for example SHA-256 plus BLAKE2b or BLAKE3) is used so that a future weakness in one function does not silently rewrite history.

These two layers must not be conflated in an audit interview:

- **IPFS / Peergos** prove that the bytes retrieved are the bytes named by the CID, and that names and sizes are not in the clear on the host.
- **The cube chain** proves that the organization’s own history of writes is append-only at the application layer.

Both are needed. Neither is a SOC 2 report.

4.6 What a disk operator sees

If keys never leave the client or the customer's HSM, a disk operator, a VM snapshot operator, and an offsite replica operator see **opaque hashed ciphertext**. They do not see PHI, they do not see document titles, they do not see the graph of who shared what with whom.

That fact **changes the Business Associate / processor conversation**. It does not automatically end it. Usernames, IP addresses, billing identity, and support logs can still be personal data. Chapter 11 takes that up.

4.7 What this does, and does not, satisfy

For HIPAA encryption of ePHI **at rest**, for SOC 2 CC6 encryption of stored data, and for ISO 27001 Annex A cryptography and storage, this is the **technical control**: ciphertext on disk, keys with the customer, hash-checked restore.

It does **not** by itself satisfy:

- a **key-custody policy** (who can unwrap, where the HSM lives, how a lost key is declared);
- a **restore-test procedure** with a dated ticket proving someone actually did it last quarter;
- **retention and legal-hold** rules (those require the write-once archive in Chapter 9, plus lawyers);
- **physical** security of the machines, which still belongs in the organization's ISMS or HIPAA facility controls.

CHAPTER 5

Global timestamped receipts

5.1 Why hashes are not enough by themselves

A cryptographic hash of a document proves that two copies are bit-for-bit the same, or that they are not. It does **not** prove **when** the document first existed. Anyone can hash a file tomorrow and claim they hashed it last year. Anyone can hash a rewritten file and present the new hash as if it were the old one.

Compliance programs that care about books and records, legal holds, intellectual-property precedence, or “produce the original prompt that the model saw” therefore need a second primitive: **a receipt, issued at the time of first registration, bound to the hash, that a later examiner can check without trusting the file’s custodian.**

5.2 What a Verae receipt is

A Verae **global timestamped receipt** is proof of:

- the **hash** of a block of digital information;
- the **time** at which that hash was first registered;
- the **sequence** of that registration relative to other registrations.

The block of digital information may be a message, an image, a document, a log segment, an AI prompt, an AI completion, or any other object that can be stored in digital media. The receipt is about the **bits**, not about the medium they happened to sit on when they were hashed. That is the Haber–Stornetta distinction from 1991, applied here as a product: time-stamp the data, not the disk.

5.3 What is registered, and what is not

Verae’s public description of sealing is that **only a fingerprint leaves the customer’s systems**. The object itself can remain in the customer’s DataCube. The central service therefore does not need — and in the intended design does not get — the plaintext in order to issue a receipt.

That split is the difference between a notary who reads the document and a notary who stamps a sealed envelope whose contents they never saw. The second notary can still later confirm that *this envelope, with this unbroken seal, was presented at this time*. They cannot tell you what was inside. For HIPAA, GDPR, and ordinary commercial secrecy, that is the desired shape.

5.4 First registration wins

A hash registry that allowed a later write to overwrite the timestamp of an earlier write would be a forgery machine. The rule is: **the first SHA-256 (and companion hash) and its receipt win**. A second presentation of the same hash returns the original receipt. A different hash is a different object — which is exactly how a revision should be modeled. Revisions get their own receipts. They do not steal the original’s time.

5.5 Sequence

Time on a wall clock is a social convention and a NTP configuration. Sequence inside a registration service is a data-structure fact: this hash was committed after that hash, in this chain, in this Merkle tree, in this cross-chain bundle. Verae receipts carry **sequence** so that an examiner can see order even when two wall-clock stamps are close enough to argue about.

5.6 Bundles

A receipt does not have to travel as a bare timestamp. It can travel inside a **digital bundle** that also holds:

- private metadata the organization needs (matter id, hold flag, classification);
- attached files that should be produced together;
- an internal chain that cross-verifies the organizational server against Verae’s central server.

The bundle is the unit an examiner is handed: “here is the object (or a capability to it), here is the receipt, here is the metadata we claim goes with it, here is the verification path.”

5.7 What a receipt does not prove

A receipt does not prove that the person who registered the hash was authorized to do so. That is an access-control and identity problem.

A receipt does not prove that the object is true, only that **those bits existed at that time**. A false document can be timestamped as honestly as a true one.

A receipt does not prove that the organization retained the object. Proof of existence is not proof of retention. Retention is the write-once archive (Chapter 9) plus the organization’s retention schedule.

A receipt is not a HIPAA, SOC 2, or ISO certificate. It is evidence that a technical control ran.

CHAPTER 6

Peergos security evaluations in Europe

6.1 What was evaluated

The offline storage and replication system used with Verae DataCubes is **Peergos**: an encrypted, peer-to-peer filesystem whose blocks live on IPFS. Peergos was designed as a trust-minimized store — the server is treated as an adversary for content and for most metadata — and that design was submitted to independent European security firms **twice**.

Those engagements are **security evaluations of the Peergos protocol and implementation**. They are pen-tests, source-code audits, and (in 2019) a cryptographic and design review. They are **not**:

- a HIPAA certification of Peergos, Verae, or any customer;
- a SOC 2 Type I or Type II report;
- an ISO 27001 certificate of an ISMS;
- a government “certified filesystem” designation.

Both firms are **EU-based**. Both full reports are **public**. That combination — independent, European, public, repeat — is unusual and is worth attaching to a vendor-assurance file, **labeled correctly**.

6.2 2019 — Cure53, Berlin, Germany

- **Firm:** Cure53
- **Location:** Berlin, Germany
- **When:** May–June 2019
- **Scope:** pentest, source-code audit, and **crypto / design review**
- **Outcome:** no fundamental architectural or cryptographic problems were identified. Issues that were identified were fixed. Cure53 stated that the platform **passed this evaluation**.

A crypto/design review is a stronger statement than a black-box pentest of a web form. It is an expert reading of whether the crypttree, the chunk encryption, the identity model, and the threat model hang together. “Passed” in Cure53’s language is not an ISO mark. It is a specialist firm saying: we looked at the cryptography and the architecture, we did not find a fatal flaw, and the issues we did find were addressed.

Primary sources (public):

- <https://peergos.org/posts/security-audit>
- https://cure53.de/pentest-report_peergos.pdf
- <https://github.com/Peergos/Peergos/tree/master/audits>

6.3 2024 — Radically Open Security B.V., Amsterdam

- **Firm:** Radically Open Security B.V.
- **Location:** Amsterdam, Netherlands
- **When:** September–November 2024
- **Scope:** crystal-box pentest plus code audit of the Peergos **web UI**
- **Funding context:** the 2024 audit post states support from **NLnet** (Netherlands) and refers to European Union Horizon 2020 **NGI-POINTER**, grant **871528**.
- **Outcome:** **0** findings rated extreme, high, or elevated; **2** moderate; **6** low. Peergos states that all were fixed. There was **no data exposure** and **no integrity compromise**. The issues were described as mostly UI crashes.

A crystal-box (clear-box) engagement means the testers had source. A web-UI scope is narrower than the 2019 crypto/design review: it is evidence about the interface that humans actually click, which is where many “encrypted backend” products fail in practice. Zero high-severity findings, no data exposure, no integrity compromise, and a public report are the facts. They are good facts. They are still not a customer’s Type II.

Primary sources (public):

- <https://peergos.org/posts/security-audit-2024>
- <https://github.com/Peergos/Peergos/tree/master/audits>

6.4 How to present these reports to an auditor

Correct:

“Our at-rest layer is Peergos. Peergos was independently evaluated in Berlin in 2019 (crypto and design) and in Amsterdam in 2024 (web UI, crystal box). Both reports are public. We attach them as **component security evaluations**. They are not our SOC 2, not our ISO 27001, and not a HIPAA certification. Our own controls, our own period of examination, and our own auditor are separate.”

Incorrect:

“We are HIPAA certified because Peergos was audited in Europe.”

“Peergos is ISO 27001.”

“The EU certified this filesystem.”

EU funding is not a certification. NGI-POINTER grant 871528 is a research-and-innovation funding fact. It is worth listing under “provenance.” It is not a registrar’s mark.

6.5 Hosted Peergos versus self-hosted DataCubes

Peergos’s hosted privacy notice has stated that peergos.net uses servers in **Germany**. A **self-hosted** organizational DataCube is a **different processing location**. The customer’s Record of Processing, BAA pack, and ISO scope must name *that* location — the customer’s ns1, region, or chosen host — not peergos.net’s Germany, unless the customer actually uses peergos.net.

The evaluations still apply to the **protocol and implementation**. Location of processing is an organizational fact on top.

6.6 What “designed under funding from Cure53 / ROS” is not

The 2019 Cure53 work and the 2024 ROS work are **evaluations** of a system that was designed by the Peergos authors. They are not a claim that Cure53 or Radically Open Security designed Peergos. The accurate statement is: the storage and replication system was **independently audited twice in Europe**, by Cure53 in Berlin (2019) and by Radically Open Security in Amsterdam (2024), with public reports, and the 2019 work included a cryptographic and design review of the architecture that Verae DataCubes rely on for data at rest.

CHAPTER 7

Verae global timestamping — a cross-blockchain receipt

7.1 Lineage

The scientific problem Verae is productizing is not new. In 1991, Stuart Haber and W. Scott Stornetta published “How to Time-Stamp a Digital Document” in the *Journal of Cryptology*. They asked how to certify when a digital document was created or last changed **without trusting the medium** and **without giving the timestamping service the document**. In 1993, with Dave Bayer, they showed how Merkle trees make that efficient. In 1997 they published “Secure names for bit-strings.” Satoshi Nakamoto’s 2008 Bitcoin white paper cites those three papers as references [3], [4], and [5] — three of the eight citations in that document.

Haber co-founded **Surety** in 1994, widely described as the first commercial blockchain: a hash-linked chain of document certificates, with a weekly summary published in the Sunday *New York Times* so that the day’s commitments sat in a public, widely archived analog record.

Haber is a **co-founder of Verae**. Verae’s public site states the aim directly: make that proof simple enough for a compliance team and precise enough for an SEC exam, without forcing the enterprise to operate a public blockchain. The quote on verae.com is: “I co-founded Verae to make blockchain accessible to enterprises without having to deal with the complexity of blockchain.”

7.2 What “cross-blockchain” means here

A single private log, held only by the vendor, is a vendor assurance letter with extra steps. A single public chain may be operationally and commercially the wrong place to put an enterprise’s registration traffic. Verae’s design is **cross-blockchain** in the following sense:

- an **organizational chain** runs at the customer, inside or beside the DataCube, recording hashes, meta-data, and sequence;
- a **central Verae chain** records fingerprints and issues receipts;

- the two are **cross-verified**, so that neither side can rewrite history without the other side's record disagreeing;
- when policy calls for it, commitments can be **anchored further** — additional public or consortium chains — so that proof of existence does not collapse if one operator disappears.

The customer does not have to become a blockchain operator to use this. That is the product claim. The examiner does not have to trust a single vendor's database to verify a receipt. That is the scientific claim, inherited from Haber–Stornetta.

7.3 Two deployment patterns

Linked organizational server. The organization runs a timestamping node next to its DataCube server. That node is linked to the central Verae timestamping server. Local registrations are committed locally (low latency, private metadata stays home) and cross-verified centrally (receipt the rest of the world can check).

Direct sync. An organization that does not want to run the organizational node can sync registrations directly with Verae's central timestamping server. Proof of existence at a specific date and time still issues. Private metadata and attached files, if any, remain the organization's problem to store — typically in the encrypted Peergos DataCube.

Both patterns produce the same external artifact: a receipt bound to a hash, a time, and a sequence. They differ in where the organizational metadata lives and in how much infrastructure the customer operates.

7.4 The digital bundle

The unit of production is a **digital bundle**:

- the **receipt** (hash, time, sequence, verification path);
- **private metadata** (classification, matter, hold, internal identifiers) that need not be on a public chain;
- **attached files** that should be produced together;
- a pointer or capability to the object in the encrypted DataCube, when the organization chose to store the object and not only its fingerprint.

An examiner can be given the bundle, or a redacted bundle, without being given a login to the production console and without Verae ever having seen the object bytes.

7.5 Proof of existence versus proof of custody

Timestamping answers: **did these bits exist by this time?**

The DataCube answers: **does the organization still have them, encrypted, hash-checkable?**

The Iceberg write-once tier answers: **can anyone quietly delete them anyway?**

A complete production to a regulator often needs all three. Timestamping alone is not an archive. An archive without a receipt is a pile of files with a clock on the filesystem that the administrator can set.

7.6 What this does, and does not, satisfy

For SEC Rule 17a-4, FINRA books-and-records, and similar regimes that demand records in a non-rewriteable, non-erasable form **with the ability to produce the original**, timestamped receipts plus write-once archive are the **technical portion** of the answer: you can show when the record was sealed, that the bits match, and that the archive copy is not under ordinary delete.

They do not satisfy:

- the requirement to **capture the channel in the first place** (if the team used an unsealed tool, there is nothing to timestamp);
- the requirement to have **written procedures** for production, legal hold, and supervision;
- the requirement that a **named principal** own the recordkeeping obligation.

Software seals what it is shown. The organization must still show it the right things, on time, under a policy someone will sign.

CHAPTER 8

Write-once Iceberg archive

8.1 The deletion problem

Encryption at rest stops a disk operator from **reading**. Hash verification stops a replica from **substituting**. Timestamped receipts stop a custodian from **back-dating**. None of those, by themselves, stop a person with legitimate administrative power from **deleting**.

Deletion is the failure mode that turns a recordkeeping program into a fine. Off-channel messaging cases at broker-dealers were, at bottom, failures to **produce**. A record that existed and was then removed, “cleaned up,” or rotated out of the only copy is, to an examiner, a record that cannot be produced.

Compliance therefore needs a tier where **ordinary administrative power does not include delete**.

8.2 What “write-once Iceberg” means in this solution

DataCubes — the encrypted, hash-chained containers — are **archived** into an **external, write-once file-storage solution** built on the **Iceberg** table/file model (a layout of immutable data files plus a metadata log that records new snapshots rather than rewriting old ones).

Properties the solution is specified to have:

- **Write once.** A committed archive object is not updated in place. A correction is a **new** object with a **new** receipt. The original remains.
- **External.** The archive is not the same disk as the live Peergos instance. Compromising the live server should not confer delete on the archive.
- **Outside any single party’s deletion control.** The customer operator, Verae, and the hosting vendor are each insufficient, by themselves, to erase a committed archive file. Practical implementations of that requirement use some combination of object-lock / WORM flags, dual-control hold, independent cloud or tape accounts, and contractual prohibition on early delete — plus the technical fact that Iceberg snapshots are new files, not overwrites.

- **For compliance reasons.** This tier exists because retention schedules, legal holds, and examiner production require it — not because it is a convenient backup.

8.3 How cubes move into the archive

A live DataCube is an operational object: it receives new writes, it is replicated as encrypted IPFS blocks, it is readable by holders of the right capabilities. On a schedule and on events (legal hold, period close, exam notice), the server solution **exports** a cube snapshot:

1. freeze a CID set and cube-chain head;
2. confirm Peergos hash verification of every block in the set;
3. register that snapshot's hash with Verae timestamping (first registration of *this* snapshot);
4. write the encrypted snapshot into Iceberg as a new immutable data file (or set of files) under a snapshot id that never reuses a previous id;
5. record, in admin-history, that the archive job ran, who authorized it, and which receipt was issued.

A restore from the archive is the same tamper-evident path as a restore from IPFS: re-fetch, re-verify hashes, decrypt only with customer keys.

What “outside the control of any party for deletion” is and is not _____

It is **not** a claim that physics forbids destruction of every copy in a fire, a court order, or a coordinated malicious act by every holder of every credential. No storage system honestly claims that.

It **is** a claim that the **ordinary** delete path — the button, the `rm`, the lifecycle rule, the “empty trash,” the support ticket to “please remove that customer folder” — is not available to any one party acting alone against a committed archive object. Dual control, object lock, and separation of the archive account from the live-server account are the organizational complements to the file format.

The organization's retention policy still has to name:

- how long a class of record stays in write-once;
- who can place a legal hold that **extends** that period;
- who, under **what dual control**, may allow a record to age out **after** the policy period, if aging out is legally permitted at all.

Software can refuse a delete API. It cannot write the retention schedule. Counsel and the records officer do that.

8.4 Relation to classical WORM

Broker-dealer Rule 17a-4 and similar texts speak of non-rewriteable, non-erasable media, originally meaning optical WORM, later allowing disk with object-lock semantics and a designated third party. Verae's public comparison is that conventional WORM is **immutable by policy** (a vendor setting), whereas Verae **seals** a record with cryptographic proof. The Iceberg write-once tier is where those two ideas meet: the file is not overwritten **and** the snapshot hash is receipted. An examiner can check the receipt even if they do not trust the vendor's “WORM was on” screenshot.

8.5 What this does, and does not, satisfy

This is the **technical portion** of retention and production: committed cubes are not under a single delete key, they are hash-checkable, and they carry a time of archival.

It does not satisfy a retention **policy** that was never written, a legal hold that was never placed, or a production request that the organization answers from a laptop copy instead of from the archive. Those are procedures.

CHAPTER 9

Architecture for an audit interview

9.1 The picture

```
Endpoint (private keys stay here / HSM)
  | HPKE content (point-to-point)
  | routing: destination + subject in the clear
  v
Untrusted message fabric (honest-but-curious)
  | cannot read bodies; can see dest / subject / size
  v
Organizational DataCube
  | append-only cube chain (prev + new + diff)
  | private metadata, attachments, internal chain
  v
Peergos cryptree (encrypted names, sizes, graph)
  | chunks -> CID / hash, hashes verified
  v
IPFS (distributed, content-addressed ciphertext)
  |
  |-- Verae timestamping
  |   fingerprint only -> receipt (time + sequence)
  |   org server linked to central Verae, or direct sync
  |
  |-- Iceberg write-once archive
  |   snapshot of cube CIDs, not under single-party delete
```

9.2 Walkthrough, in the order an examiner usually asks

Where are the keys? On the endpoint or in the customer's HSM. Not on the storage host. Not in the public-key directory (that directory holds public keys only). Not in the NATS payload.

What does the network see? Destination, subject, size, timing. Not the body. Production algorithm is HPKE, not a lab construction.

What does the disk see? Padded encrypted chunks under random labels. Peergos verifies hashes. A restore re-fetches CIDs and re-checks them.

How do you prove when it existed? A hash is registered the first time it is seen. A receipt binds hash, time, and sequence. The organizational chain and the central Verae chain cross-verify. Verae is not given the object, only the fingerprint.

How do you stop delete? Live cubes replicate as hashed ciphertext. Committed snapshots go to an external Iceberg write-once tier that is specified to sit outside any one party's delete control.

How do you know the software was not quietly reconfigured? Configuration is Ed25519-signed. Unsigned files are rejected. Admin-history records previous state, new state, and a diff.

How do you inspect without turning inspect into a leak? k-of-n officers. The author of a change is not an officer on that change. Log-before-reveal.

9.3 Identity planes

There are two login planes, and they must not be collapsed in an interview:

- **Drive / Peergos identity** — unwraps cubes. This is the cryptographic identity.
- **Console identity** — TOTP (RFC 6238), session cookie, admin actions. Google SSO, if present, does **not** unwrap cubes.

Mixing those planes is how “we have SSO” becomes “SSO is the master key.” It is not.

9.4 NATS remains untrusted

The reference message fabric is NATS. It is an honest-but-curious broker. Destinations in the clear are **passthrough**, not a bug. Bodies are ciphertext, or the message is rejected. Error responses carry codes and lookup identifiers, not payloads.

A public-edge NATS listener, if enabled, is token-gated and is not the same socket the internal services use.

9.5 What to hand the examiner

- this document (HTML or PDF);
- the two public Peergos reports, labeled as component evaluations;
- a data-flow diagram of *this* instance;
- the public-key listing (public keys only);
- a restore-test ticket;

- an admin-history excerpt for a dated change;
- the organization's policies, not Verae's.

Companion PDFs, same folder as this file (relative PDF hrefs, not Markdown):

- [peergos-for-compliance.pdf](#)
- [nats-service-endpoints.pdf](#)
- [secure-messaging.pdf](#)

CHAPTER 10

BAAs, DPAs, and ciphertext without host keys

This chapter is **guidance for an evidence pack**. It is not legal advice. Counsel has to sign the actual determination.

10.1 The technical fact

Data at rest lives in the **encrypted Peergos crypttree**. Restore uses **distributed, hash-verified, encrypted IPFS blocks**. Hosts and backup media see **opaque blobs** (and random CHAMP labels), not filenames, not PHI, not the share graph — **if** private keys never leave the client or the customer's HSM.

That fact is true of the Peergos design and is the design Cure53 reviewed in Berlin. It **changes** the Business Associate and GDPR-processor conversation. It does **not** automatically delete the need for contracts.

10.2 HIPAA — Business Associate

A Business Associate is a person who **creates, receives, maintains, or transmits ePHI** for a covered entity.

A technical argument the organization can document:

1. ePHI is encrypted **before** it leaves the endpoint.
2. The storage node, IPFS, disk, Iceberg archive, and backup hold **ciphertext plus hashes**.
3. Private keys **do not** reside on the storage host. Hash verification detects substitution.
4. Therefore the host **cannot maintain ePHI in intelligible form**.

HHS has distinguished **conduits** and parties that never have access to ePHI. Many programs still **sign a BAA** whenever a vendor touches infrastructure, because misconfiguration, a support export, or a log could later expose plaintext. OCR will look at **the organization's** risk analysis, not at Verae's architecture diagram.

Practical pack:

- a risk-analysis memo citing crypttree, IPFS hash checks, key custody, and the two public Peergos evaluations;
- a BAA **or** a written determination that a named vendor is **not** a BA because they cannot decrypt;
- a list of every party with disk, VM, backup, IPFS, or Iceberg credentials.

10.3 GDPR — processor versus technical measure

Encrypted data can still be **personal data** if it is reasonably attributable (usernames, IPs, invoice identity). GDPR Article 32 lists encryption as a security measure, not as an exemption from Article 28.

Peergos is designed so the **server is an adversary** for content and most metadata. EU-based independent pentests evaluated that design. Processing location for **self-host** is the customer's region, not peergos.net's Germany, unless the customer uses peergos.net.

A DPA may still be required with anyone who **processes** usernames, logs, or billing. Ciphertext-only storage **narrows** the technical annex. It does not always eliminate the DPA.

10.4 NATS operators

NATS is an untrusted broker. Content is HPKE. Destinations and subjects are in the clear. A NATS operator **cannot** read bodies without endpoint private keys. They **can** see routing metadata. They **can** drop, delay, or copy ciphertext. Treat them like a transit provider: whether a DPA or BAA is needed depends on whether routing metadata is personal data in the relevant jurisdiction.

10.5 Verae as timestamping service

If Verae receives **only fingerprints**, Verae's role for **content** is not “stores the records.” Verae's role is “registers hashes and issues receipts.” That is a narrower processing. Counsel still has to decide whether the fingerprint, plus whatever account data exists (company name, billing, operator emails), requires a DPA or a BAA. Do not let a sales sentence skip that memo.

10.6 Component assurance versus the organization's report

Attaching Cure53 2019 and ROS 2024 is **vendor / component assurance**. It is appropriate. It is **not** the organization's SOC 2, ISO 27001, or HIPAA program. Those still require the organization's scope, Record of Processing, key-custody, inspect k-of-n, admin-history, instance evidence, restore tests, and the contracts that match who can actually see plaintext.

CHAPTER 11

Audit-ready checklist

Use this as a working list. Check an item only when **evidence exists** (screenshot, log export, signed policy, ticket, receipt). This list is not a certificate. Software having been installed does not tick these boxes.

11.1 A. Scope and honesty

- Named legal entity and systems in scope (console, Drive, message fabric, IPFS, timestamping link, Iceberg archive)
- Written sentence in the system description: **software is not our HIPAA / SOC 2 / ISO 27001 certificate**
- Data-flow diagram: endpoint → HPKE → DataCube → Peergos/IPFS → receipt → Iceberg
- PHI / personal data vs ciphertext vs routing metadata, named
- Peergos evaluation chapter attached, labeled **component security evaluation**

11.2 B. Data in transit

- Production algorithm is HPKE (or documented successor), not a lab construction
- Destinations and subjects documented as **visible by design**
- Error paths carry codes and lookup ids, not bodies
- Public-key directory lists public keys only; private keys mode 0600 or HSM
- Broker operator named; metadata vs content in the annex

11.3 C. Data at rest

- Customer holds Peergos / Drive keys; not on storage host
- Peergos hash verification on write and on read, evidenced
- Restore test: recover a cube from content-addressed ciphertext **without** a plaintext tape
- Dual-hash cube chain; first-hash-wins registry

11.4 D. Timestamping

- First-registration rule documented and tested (second submit returns original receipt)
- Organizational node **linked** to central Verae, **or** direct sync, named in the diagram
- Sample bundle: receipt + private metadata + attachment pointer, redacted for the evidence pack
- Verae receives fingerprints, not objects — stated in the data map

11.5 E. Write-once archive

- Iceberg (or equivalent) export job exists and has a dated last-run
- Object-lock / dual-control / separate account — evidence that no **single** party can delete a committed snapshot
- Retention schedule written by records/counsel, not by engineering folklore
- Legal-hold procedure extends retention; ticket example

11.6 F. Access and change

- Console requires authentication (TOTP or equivalent); Drive login is a separate plane
- Inspect is k-of-n; author is not an officer; log-before-reveal
- Joiner / mover / leaver for console users
- Signed configuration; unsigned rejected; admin-history prev + new + diff
- HSM or a dated plan to move lab keys to HSM

11.7 G. Contracts (counsel)

- Written BA / not-a-BA determination for disk, VM, backup, IPFS, Iceberg
- DPA Article 28 where usernames, logs, or IPs are processed
- NATS operator: metadata vs content
- Verae timestamping: fingerprint-only processing described
- Peergos PDFs attached and labeled “not our Type II / ISO”

11.8 H. Independent examination of *this* organization

- SOC 2 Type I/II engagement, **or** ISO 27001 registrar, **or** HIPAA risk analysis plus policies — **the program actually chosen**, not all three as wallpaper
- Evidence window (Type II / surveillance) if applicable
- Named internal owner who will sit in the meeting and not claim the software is the certificate

CHAPTER 12

How to use this briefing

1. Read the **executive summary** aloud in the first five minutes of any vendor, board, or auditor meeting that touches this system. If anyone says “so we are certified,” stop and reread Chapter 1.
2. Read **What Verae provides** so the five verbs (store, communicate, timestamp, verify, audit) are not confused with an ISMS, a Type II, or a HIPAA program.
3. Read the **transit, rest, receipts, Peergos evaluations, timestamping, and Iceberg** chapters in that order. They are the technical portion, in the order an examiner usually probes: “can the wire read it, can the disk read it, can you prove when, who looked at the crypto, can you produce it later.”
4. Fill the **checklist** with **this instance’s** evidence. Empty checkboxes are not a moral failing; they are the work remaining.
5. Give **BAAs and DPAs** to counsel with the architecture diagram. Do not let engineering declare a vendor “not a BA.”
6. Attach the two **public** Peergos reports as **vendor security evaluations**, with a cover slip that says they are not the organization’s SOC 2, ISO 27001, or HIPAA certification.
7. Point auditors at **live technical surfaces** (health, public-key listing, Drive). Do not give them private keys. Do not give them a story that the pentest PDF is the Type II.
8. Keep the **biographies** at the back of the PDF for provenance — who built the timestamping science, who is building the product, who architected the internet integration and the DataCube server side — without substituting biography for controls.

Companion system PDFs in the same folder:

- [peergos-for-compliance.pdf](#)
- [nats-service-endpoints.pdf](#)
- [secure-messaging.pdf](#)

CHAPTER 13

James H. Garfinkel

13.1 Name and sources

This chapter summarizes **James H. Garfinkel** (the spelling on the FINRA BrokerCheck individual report; also styled Garfinkle in some informal references). Two public sources were used, as requested:

- **FINRA BrokerCheck**, CRD number **5052743**, https://files.brokercheck.finra.org/individual/individual_5052743.pdf
- **verae.com**, which names Stuart Haber as co-founder and describes the DataCube product. As of this writing the public marketing site does **not** carry a dedicated biography page for Garfinkel. The product and company description below is therefore taken from verae.com’s company narrative plus Garfinkel’s own public statement of role, not from a fabricated “about the CEO” page.

13.2 Role at Verae

Public professional listings (including a long-standing Facebook work listing) identify Garfinkel as **CEO of Verae LLC**, from **January 2020**, based in **New York, New York**. In that listing he describes Verae’s technology as blockchain-authenticated containers called **DataCubes**, “which conveniently and securely ensure integrity and enable selective sharing of digital information and records of all sorts,” and he quotes Stuart Haber on the original motivation for timestamping digital objects.

That description matches the public verae.com product: records remain in customer-controlled storage; Verae seals a fingerprint; the scientific core is Haber-style cryptographic timestamping, aimed at compliance and examinations rather than at operating a public chain.

13.3 FINRA BrokerCheck (CRD 5052743)

The following facts are from the public BrokerCheck report. They are **registration history**, not a Verae claim and not an endorsement by FINRA.

- **Name on the report:** James H. Garfinkel
- **CRD:** 5052743
- **Current registration:** this broker is **not currently registered**
- **Disclosed events:** the report summary states **no** events disclosed about this broker
- **Examinations recorded on the summary:** 0 principal/supervisory exams; 3 general industry/product exams; 3 state securities law exams
- **Investment adviser:** BrokerCheck points readers to the SEC IAPD site for any IAR record

Prior broker registrations listed on the report:

- **LPL Financial LLC**, CRD 6413, New York, NY, **January 2013 – March 2024**, registered representative
- **Hornor, Townsend & Kent, Inc.**, CRD 4031, New York, NY, **November 2011 – December 2012**
- **Park Avenue Securities LLC**, CRD 46173, New York, NY, **April 2008 – October 2009**

Other employment lines that appear in the same public report family include **Abbott Solutions Inc.** (insurance advisor, from November 2009) and **MPWM Advisory Solutions LLC** (investment adviser representative, January 2014 – November 2020). Readers should treat the PDF as the living source; registration status changes.

13.4 Education (public professional listings)

Public professional listings state:

- **Hamilton College**, B.A. Economics, class of **1980**
- **United Nations International School**, class of **1975**

Other ventures listed on professional-directory sites (not FINRA, not verae.com) have included Beryl Consulting, Intaglio Blockchain, and earlier broker-dealer roles. Those directory sites are secondary; they are noted here only so that a reader who searches the name is not surprised. They are not used as primary evidence of Verae corporate structure.

13.5 Why this biography is in the briefing

A compliance briefing is not a pitch deck. Garfinkel's biography is here because the user of this system is entitled to know **who is on the commercial and regulatory-facing side of Verae**, and because a broker-dealer or RIA CCO will **look the name up on BrokerCheck anyway**. The honest presentation is:

- CEO of Verae LLC (public listing, from 2020);
- a long New York securities-industry registration history ending March 2024 at LPL, **not currently registered**;

- **no** BrokerCheck disclosed events on the current report;
- **not** given a biography block on verae.com at the time this document was compiled — Haber is the scientist the site puts on the homepage.

Do not treat a CRD number as a product certification. Do not treat a CEO biography as a control.

CHAPTER 14

Stuart Haber

14.1 Role at Verae

Stuart Haber is a **co-founder of Verae**. The company’s public site introduces him as one of the creators of blockchain and as the inventor, in 1991, of cryptographic timestamping — “the science that lets you prove a record hasn’t been altered, and the foundation under every blockchain since.” The site quotes him:

“I co-founded Verae to make blockchain accessible to enterprises without having to deal with the complexity of blockchain.”

—Stuart Haber, Co-Founder of Verae; co-inventor of blockchain timestamping, 1991

Verae’s product claim is that this is the proof Haber has been building toward: simple enough for a compliance team, precise enough for an SEC exam, with the customer’s records remaining in the customer’s storage and only a fingerprint being sealed.

14.2 The scientific work

Haber is a cryptographer. He worked at **Bellcore** (Bell Communications Research) in Morristown, New Jersey, in the late 1980s and 1990s, in the research culture descended from Bell Labs. With physicist **W. Scott Stornetta** he posed a problem that looks obvious only after it is solved: digital documents live on media that can be altered without a mark; how do you certify **when the bits existed**, without trusting the disk, and without handing the document to the timestamping service?

1991. Stuart Haber and W. Scott Stornetta, “How to Time-Stamp a Digital Document,” *Journal of Cryptology*, volume 3, number 2, pages 99–111. This is the paper that introduces computationally practical digital timestamping so that a user cannot back-date or forward-date a document even with the collusion of a timestamping service, while keeping the document itself private. Bitcoin’s white paper cites it as reference [3].

1993. Dave Bayer, Stuart Haber, and W. Scott Stornetta, “Improving the Efficiency and Reliability of Digital Time-Stamping,” in *Sequences II: Methods in Communication, Security and Computer Science*, pages

329–334. Merkle trees, efficient certificates, publication of a compact root. Bitcoin’s white paper cites it as reference [4].

1997. Stuart Haber and W. Scott Stornetta, “Secure names for bit-strings,” *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 28–35. Bitcoin’s white paper cites it as reference [5].

Those three papers are **three of the eight** citations in Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System” (2008). Haber has remarked that this is a .375 batting average, and that Bitcoin is an application built on top of the early blockchain rather than the invention of the data structure itself. That is a historical statement, not a claim that Verae is Bitcoin.

14.3 Surety, 1994: first commercial blockchain

Haber and Stornetta took the research into production as **Surety**. Surety hashed customer documents, linked those hashes in a chain of certificates, and published a weekly summary hash in the classified section of the *Sunday New York Times*. That analog publication is widely described as the oldest surviving public blockchain: a commitment that does not depend on Surety’s servers remaining honest or online, because the *Times* is independently archived. The design idea — **commit a compact fingerprint to a widely witnessed record, without revealing the documents** — is the same idea Verae productizes for enterprise compliance.

14.4 Other cryptographic publications

Haber’s research record is broader than timestamping. It includes work on minimum-knowledge interactive proofs, symmetric public-key encryption, and secure multi-party protocols (including “Cryptographic Computation: Secure Fault-Tolerant Protocols and the Public-Key Model”). A reader who wants the academic trail should start with the three Bitcoin-cited papers and the 1991 *Journal of Cryptology* article, then the ACM CCS 1997 paper.

14.5 Why this biography is in the briefing

Timestamping in this document is not a metaphor. It is a specific scientific object — hash, time, sequence, privacy of the document from the notary — that Haber defined in print in 1991, commercialized in 1994, and is now building into Verae so that a CCO can produce a receipt instead of a vendor letter. The biography is here as **provenance of the receipt**, not as a substitute for the organization’s controls.

The same honesty that applies to Peergos applies here: Haber co-invented the timestamping chain; he did not issue the customer’s SOC 2.

CHAPTER 15

George Lambert

15.1 Role

George Lambert is an Internet technical architect and a New Hampshire public official. Public candidate listings (BallotReady) identify him as **CTO / Chief Architect, Verae LLC**, from **2019**. He is the architect of the DataCube **server** side of the solution described in this briefing: the assembly of encrypted storage, point-to-point messaging, public-key directory, organizational timestamping link, and write-once archive that an organization actually runs.

15.2 Internet technical architect, from 1994

Lambert's public account of his engineering work begins in the **earliest days of the commercial Internet**, when shipping an application that could speak TCP/IP was still a specialist task.

In **1994–1995** he was moving through senior application engineering and sales roles (Terralogics; Serif) at the moment Borland's developer tools were being connected to the public network. He then founded **NetMasters LLC** (New Hampshire, formed 1996; he describes the product work from 1995). At NetMasters he **personally designed and built FastNet Tools** — the **Internet integration technology that shipped with Borland's Delphi and C++ Builder** products.

Those tools are the historically important fact. They enabled a generation of Delphi and C++ Builder developers to Internet-enable applications as a shipped part of Borland's commercial toolchain, not as a research demo. Lambert's own summary is that FastNet Tools “enabled millions of developers to internet-enable their applications in minutes.” That is the beginning of the through-line that ends, in this briefing, at a DataCube server: **make the hard network problem a product that other people can operate**.

He continued as **Vice President of Engineering** at GoldenWare / **FlightLookup** (from 1995, for two decades on the public listings), building airline-information systems, and later as founder of PodKey Solutions (2004–2010) and as an interim/contract CTO.

15.3 New Hampshire public office

Lambert is a **Republican** from **Litchfield, New Hampshire**, born **4 September 1968** in Sanford, Maine (Wikipedia; Vote Smart).

- **Litchfield Board of Selectmen**, from 2007 (Vote Smart lists council of town selectmen, Town of Litchfield, 2007–present on the biography page then current).
- **New Hampshire House of Representatives**: Hillsborough 27th district, December 2010 – December 2012; Hillsborough 44th district, December 2012 – December 2014.
- Subsequent campaigns for the New Hampshire Senate, District 18 (2014, 2018, 2020, 2022 Republican primary winner in several of those cycles).

Vote Smart records a diploma from **Pelham High School, 1986**; family (wife Rhonda, children); home city Litchfield. He has also served on the **National Grange Strategic Planning Task Force** (1999–2004).

His public technical site, <https://georgelambert.org/>, describes three decades at the intersection of technology, governance, and sound money, and repeats the FastNet / Borland fact as the credential for knowing what scalable architecture looks like.

15.4 Why this biography is in the briefing

The DataCube Server Solution is not only a timestamping API. It is a **running system**: keys, brokers, crypt-tree, replicas, admin-history, Iceberg export. Lambert is the person whose public career is the **implementation** of Internet integration from the Borland era through to this server stack, and whose public service is New Hampshire legislative and municipal office. The biography is here so that a reader who asks “who architected the server side, and on what history?” has a sourced answer.

It is not a substitute for a pentest, a Type II, or a restore test. Code either runs or it does not; Lambert’s own stated ethic. This briefing is the description of what the code is for, and of what it cannot certify by existing.

CHAPTER 16

Verae Inc — contact

This page is the contact block that also appears in the footer of every page of the PDF edition and at the bottom of every HTML chapter.

Verae Inc

Website: <https://www.verae.com>

Product / activity console: <https://app.verae.com>

Book a conversation: use the “Book a call” path on <https://www.verae.com> (the public site states that the company will be in touch within one business day).

API (fingerprint seal, as described on the public site): <https://api.verae.com/v1/seal>

This briefing (HTML and PDF):

- <https://docs.pfc.georgelambert.org/audit-ready/>
- <https://git.georgelambert.org/marchon/peergos-making-yourself-audit-ready-with-verae-datacubes>

A street address and telephone number are **not invented here**. If a procurement file requires them, take them from a current Verae engagement letter, W-9, or the form on [verae.com](https://www.verae.com), not from this paragraph.

16.1 Closing reminder

Software alone does not make an organization HIPAA certified, SOC 2 attested, or ISO 27001 certified.

The Verae DataCube Solution gives you tools to store, communicate, timestamp, verify, and audit. Your organization still has to write the policies, run the procedures, operate the internal controls, and sit with an independent auditor. Verae provides the tools, the background, and the software infrastructure for the **technical portion** of that work.

Verae Inc — <https://www.verae.com>