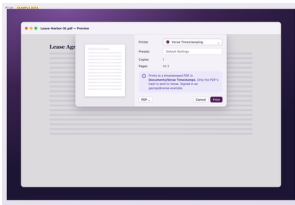


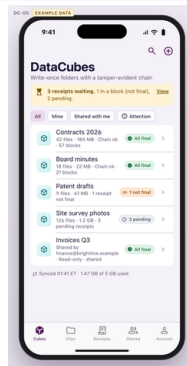
Verae Solutions

Complete product & platform overview

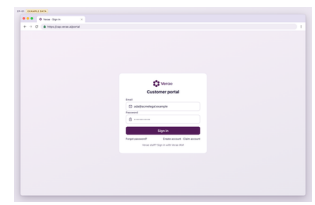
Timestamp registry · DataCubes · PrintIt · Zapier · MCP · Apps · Portals · APIs



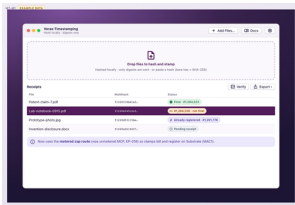
PT-01



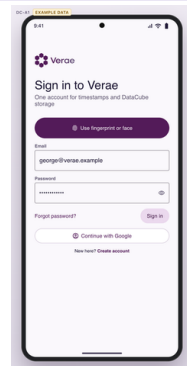
DC-05



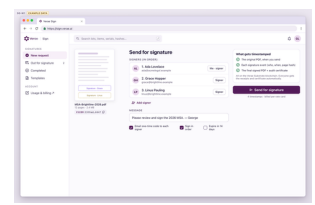
ZP-01



MT-M1



DC-A1



SG-W1

Screens from local design pack only · customer.verae.ai · peergos.verae.ai

© Verae / FlightLookup Inc.

Contents

1. Executive summary
2. PrintIt — print path to the Verae timestamp registry
3. Zapier & automation integrations
4. MCP timestamping for agents
5. Applications — all screens by product family
6. DataCube storage & secure sharing (Peergos + S3)
7. Portals, sales, admin, accounting & ops
8. Pricing snapshot
9. Full API reference (parameters, types, sample calls)
10. Glossary

1. Executive summary

Verae delivers two complementary capabilities: (1) a commercial **Verae timestamp registry** that records content digests and returns receipts with finality stages, and (2) **Digital DataCubes** — Peergos-based encrypted storage for files, directories, and whole cubes — with sharing, chain history, metadata/tags, restamping, and versioned change tracking.

The same commercial account at **customer.verae.ai** meters stamps and storage across PrintIt, Zapier, MCP, mobile and desktop apps, and direct APIs. API keys are xVerae- tokens (stored hashed). Billing is Stripe prepaid or Purchase Order. ITAD field logistics is optional (build flag ENABLE_ITAD).

At a glance

Pillar	What customers get
Timestamp registry	Digest-only stamps; receipts pending→final; restamp + JSONL metadata
DataCubes	E2E encrypted Peergos storage; share file/dir/cube; S3 hosted or BYO
PrintIt	Print or share any document → hash → registry → certificate
Automation	Zapier + HTTP + MCP agents with the same metering
Commerce	Free/Pro/Business tiers; per-customer sales pricing; Stripe or PO

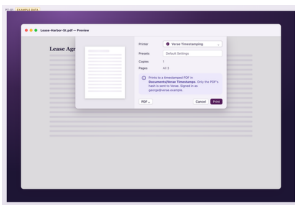
2. PrintIt — print path to the Verae timestamp registry

PrintIt turns everyday print and share actions into registry evidence. The user selects the Verae Print / Verae Timestamping destination (or mobile share sheet). The client hashes the document locally, posts only the digest with an xVerae- key, and produces a human-readable certificate PDF plus machine receipt. No document body is required on the registry.

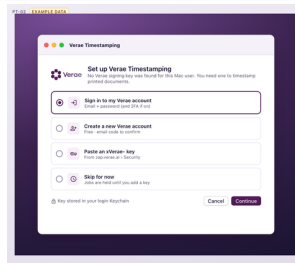
Capabilities

- macOS / Windows / Linux virtual printer or print dialog destination
- iOS / Android share-to-timestamp paths
- Job queue with receipt stages; final certificate (CERT-*)
- Settings for account key, defaults, and notifications
- Same commercial metering as API and Zapier

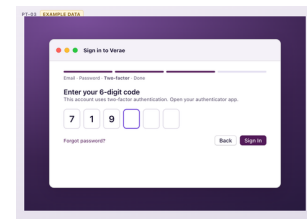
PrintIt screens (local design frames)



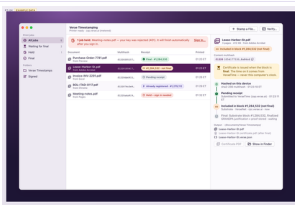
PT-01



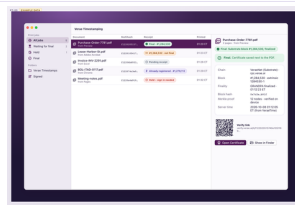
PT-02



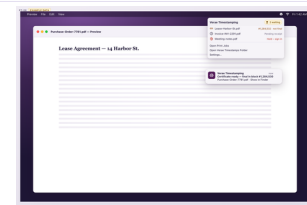
PT-03



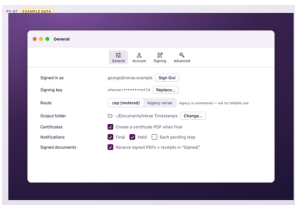
PT-04



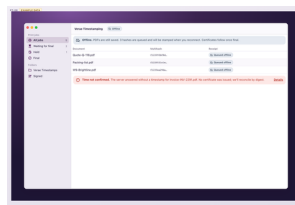
PT-05



PT-06



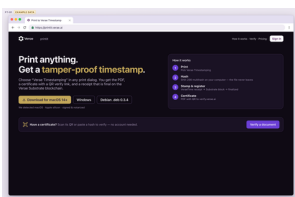
PT-07



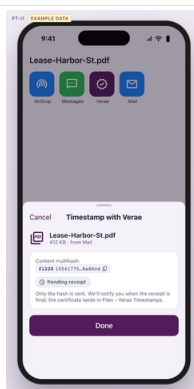
PT-08



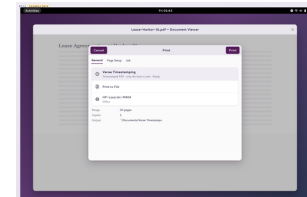
PT-A1



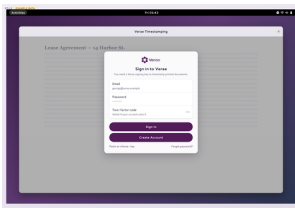
PT-B1



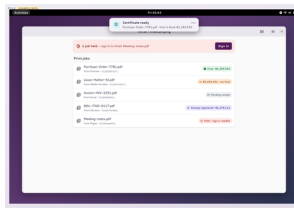
PT-I1



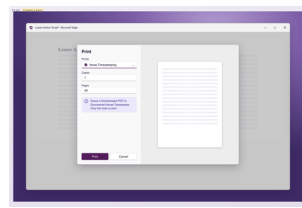
PT-L1



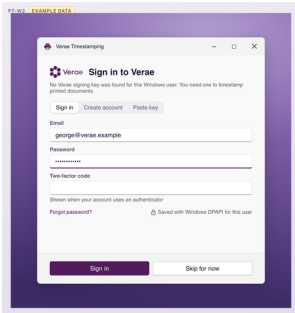
PT-L2



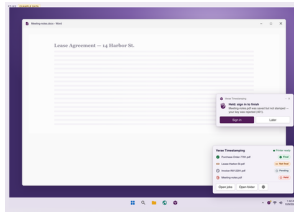
PT-L3



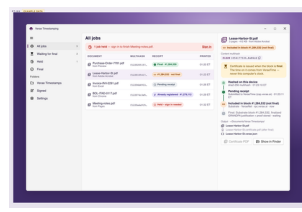
PT-W1



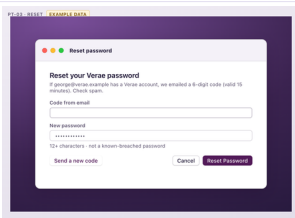
PT-W2



PT-W3

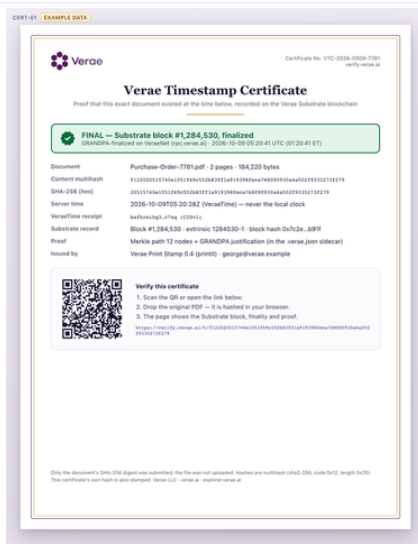


PT-W4



PT-03-v1

Certificate outputs



CERT-01



CERT-01-v1

3. Zapier & automation integrations

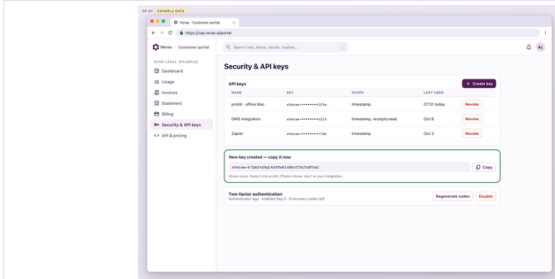
Verae connects to **Zapier** and similar automation so workflows can stamp digests, look up hashes, and fetch receipts without custom servers. The Zapier app and HTTP API share the customer's xVerae- key and the same rate card as direct API calls. (Edge package names may still say “zappier”; the commercial host is customer.verae.ai / zap.verae.ai.)

Typical Zap steps

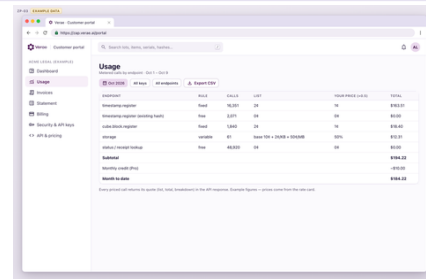
- Trigger: new file in Drive/Dropbox/email → compute SHA-256 in middleware or pre-hash
- Action: Create timestamp (POST /v1/timestamp)
- Action: Get status / receipt when final
- Optional: write receipt JSON back to sheet or ticket

Automation users manage keys and usage in the **customer portal** (same as mobile). Design frames below show portal billing/security used by Zapier key holders.

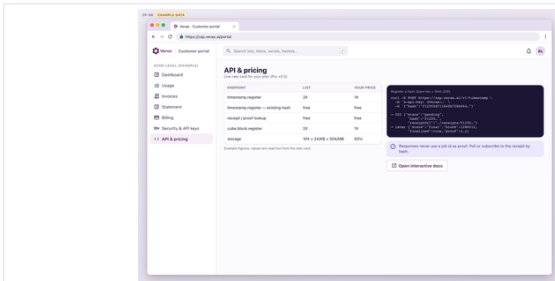
Portal surfaces used with Zapier keys



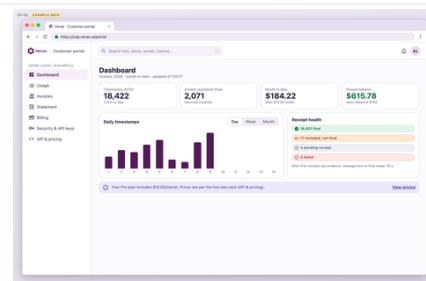
ZP-07



ZP-03



ZP-08



ZP-02

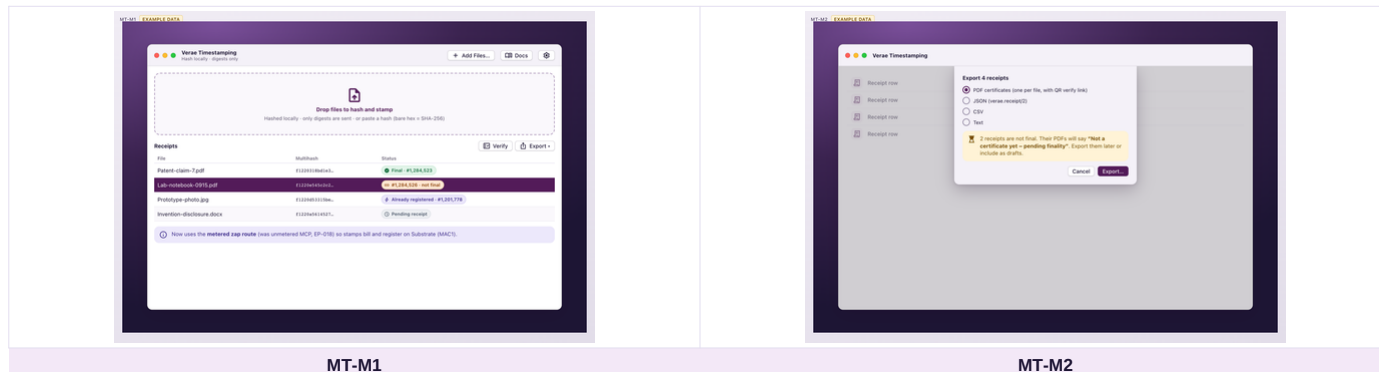
4. MCP timestamping for agents

MCP (Model Context Protocol) exposes Verae timestamp tools to AI agents and IDEs. Agents authenticate with a bearer token file (raw xVerae- or JWT) — never paste secrets into chat. Tools stamp digests only; public HTTP MCP does not use host-login fallback for writes.

Surfaces

- MCP server (Node/Go variants in platform ops) on mcp.verae.ai / mcp.veraetime.net paths
- Desktop companion: Mac Timestamping app (design MT-*) for drop-folder + receipts
- Same /v1/timestamp metering when routed through commercial edge

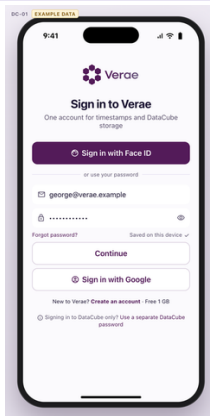
Mac Timestamping / MCP companion screens



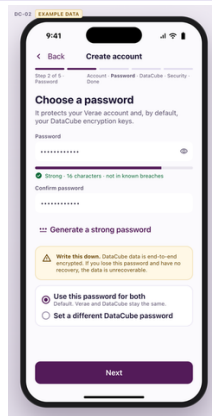
5. Applications — screens by family

Below: every primary local design frame in the pack, grouped by product. Variants (-v1...) included where present. Images are pre-scaled local files only.

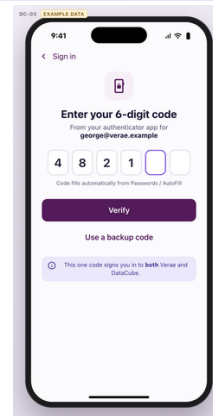
5.1 DataCube — iOS



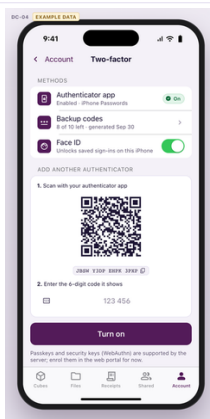
DC-01



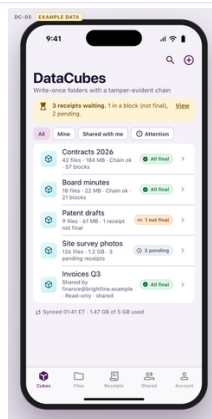
DC-02



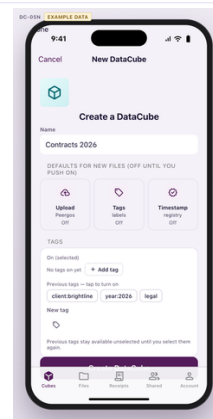
DC-03



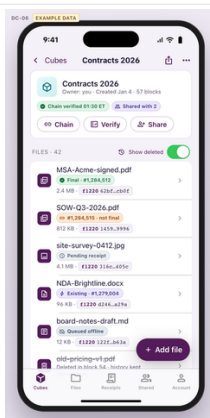
DC-04



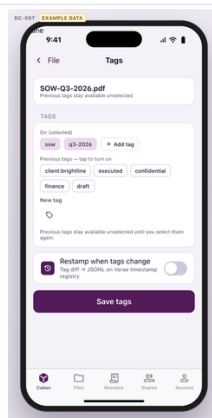
DC-05



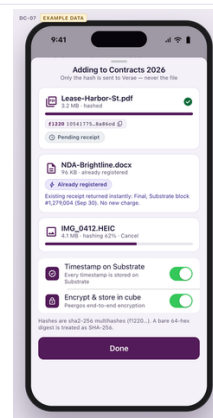
DC-05N



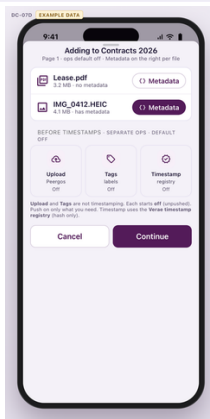
DC-06



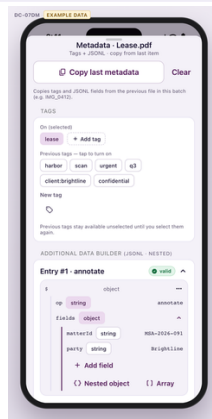
DC-06T



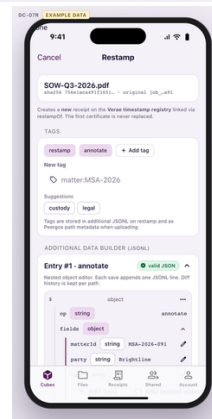
DC-07



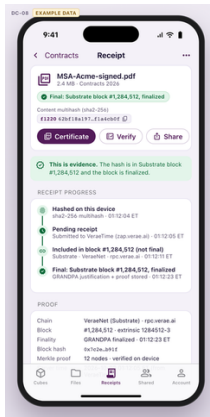
DC-07D



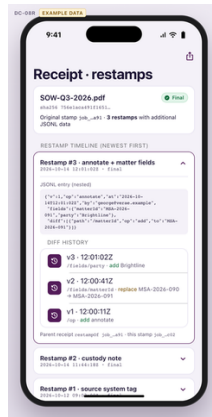
DC-07DM



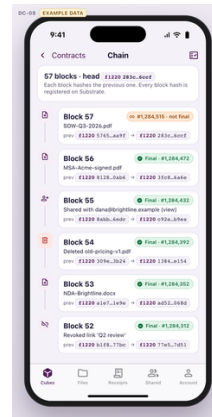
DC-07R



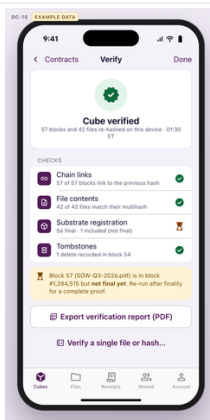
DC-08



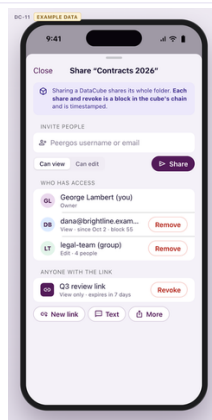
DC-08R



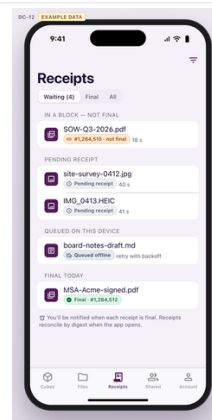
DC-09



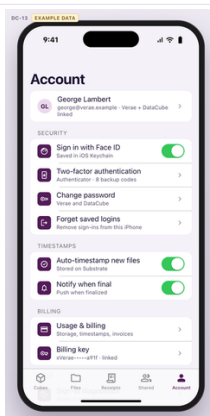
DC-10



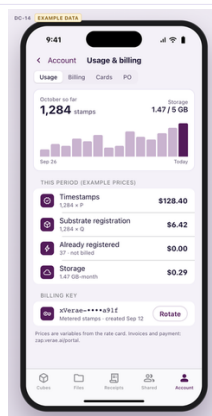
DC-11



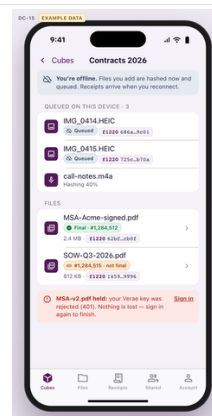
DC-12



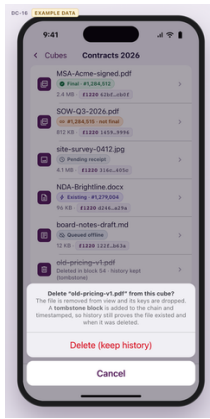
DC-13



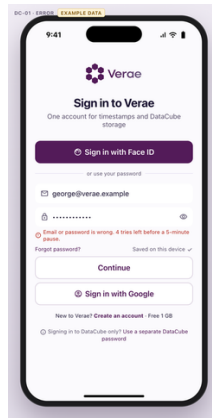
DC-14



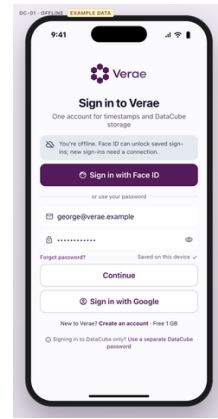
DC-15



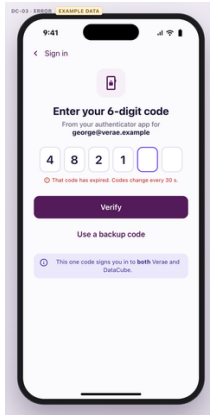
DC-16



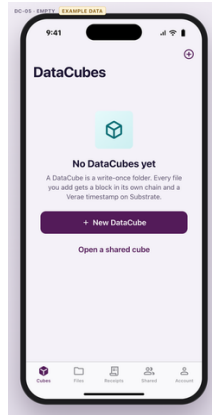
DC-01-v1



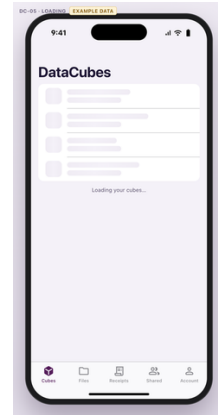
DC-01-v2



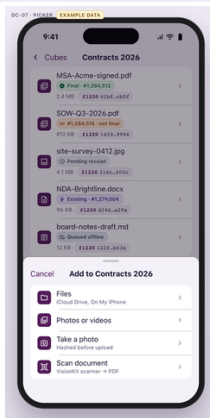
DC-03-v1



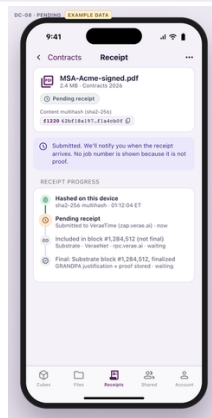
DC-05-v1



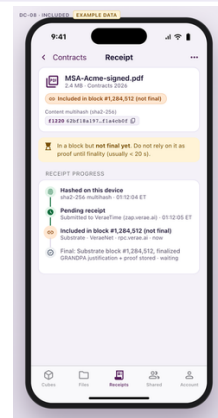
DC-05-v2



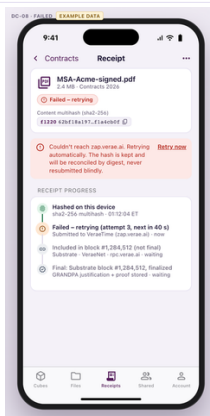
DC-07-v1



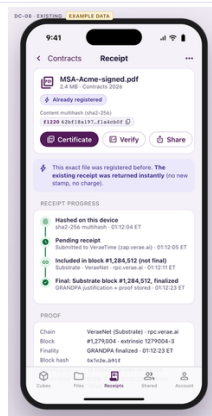
DC-08-v1



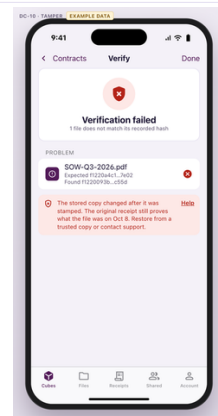
DC-08-v2



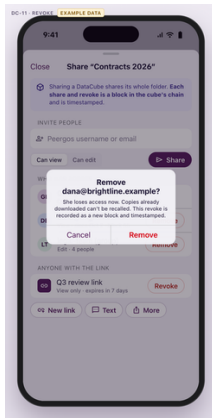
DC-08-v3



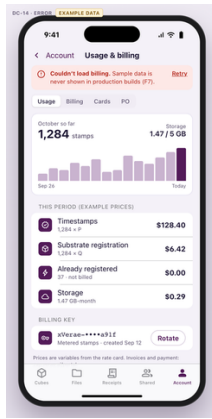
DC-08-v4



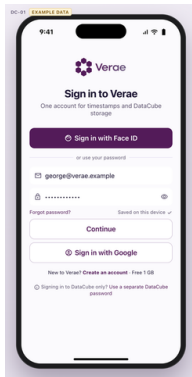
DC-10-v1



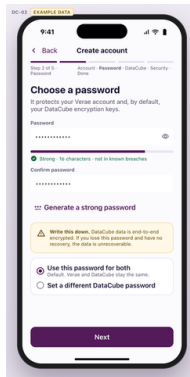
DC-11-v1



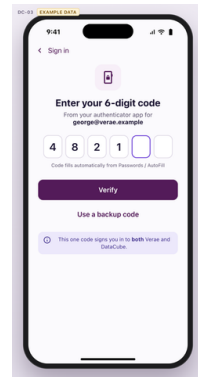
DC-14-v1



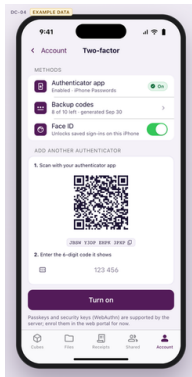
DC-01



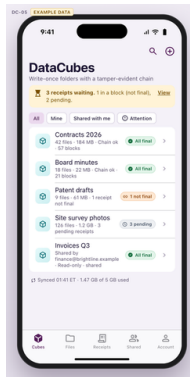
DC-02



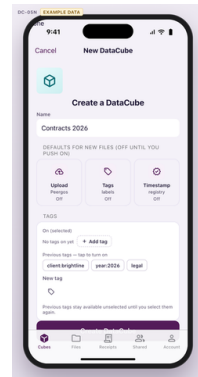
DC-03



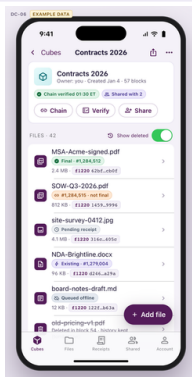
DC-04



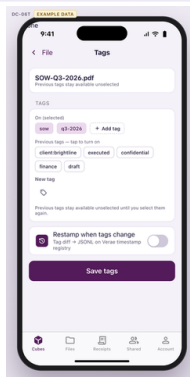
DC-05



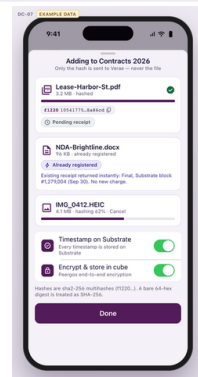
DC-05N



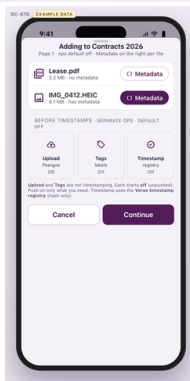
DC-06



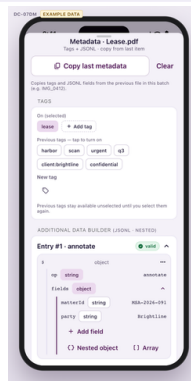
DC-06T



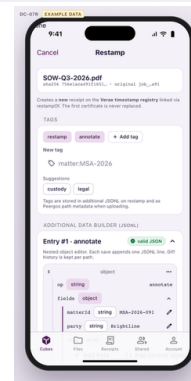
DC-07



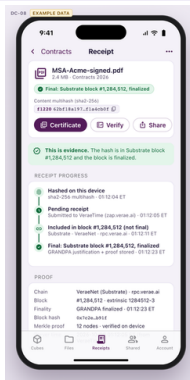
DC-07D



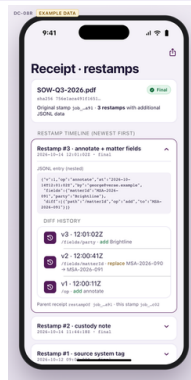
DC-07DM



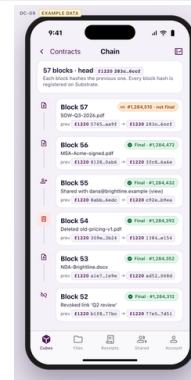
DC-07R



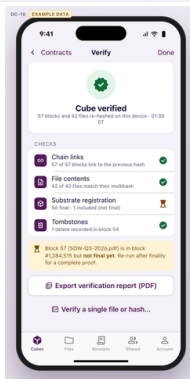
DC-08



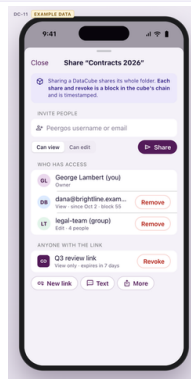
DC-08R



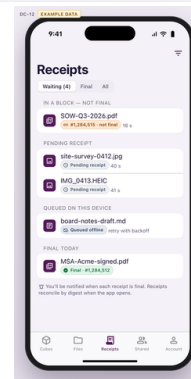
DC-09



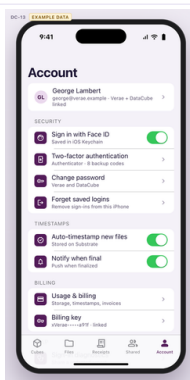
DC-10



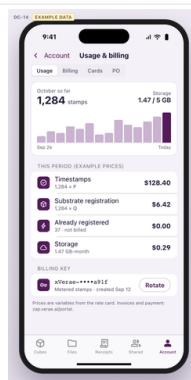
DC-11



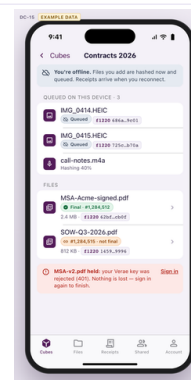
DC-12



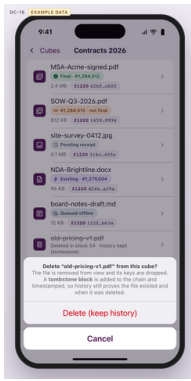
DC-13



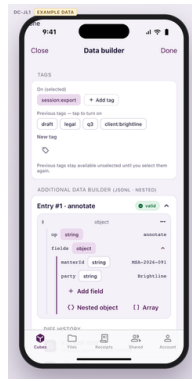
DC-14



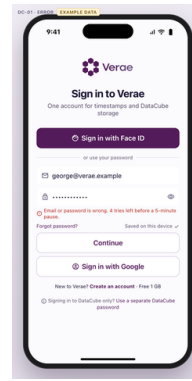
DC-15



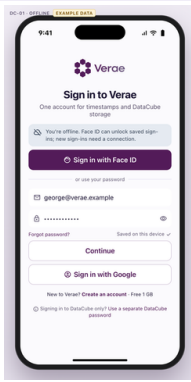
DC-16



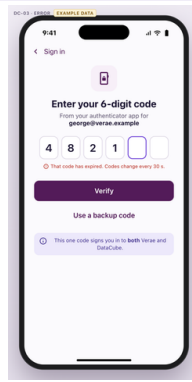
DC-JL1



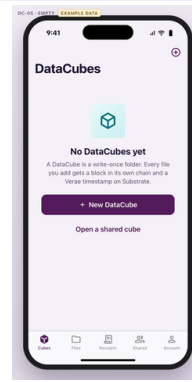
DC-01-v1



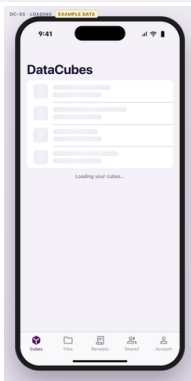
DC-01-v2



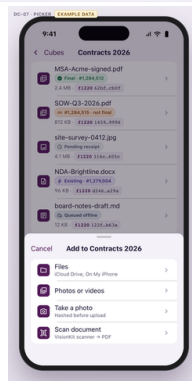
DC-03-v1



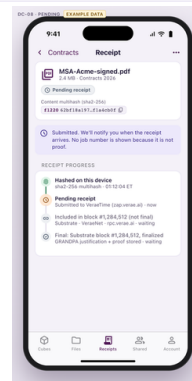
DC-05-v1



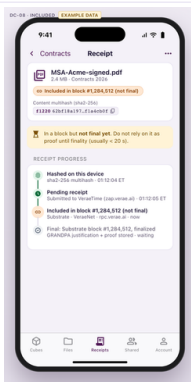
DC-05-v2



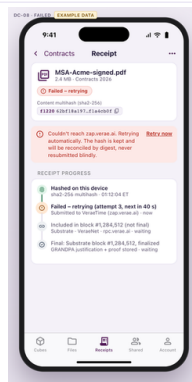
DC-07-v1



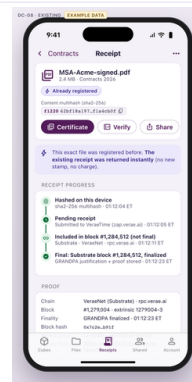
DC-08-v1



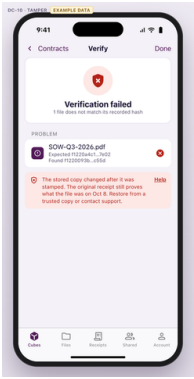
DC-08-v2



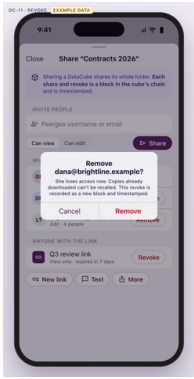
DC-08-v3



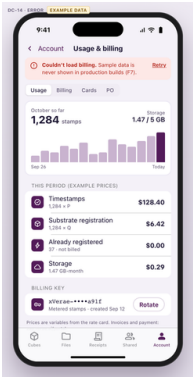
DC-08-v4



DC-10-v1

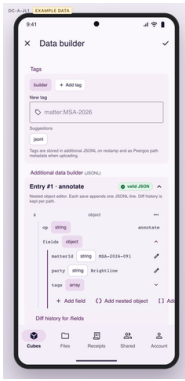


DC-11-v1

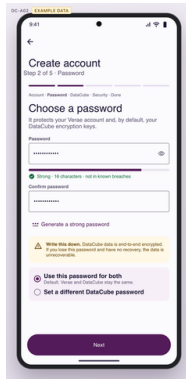


DC-14-v1

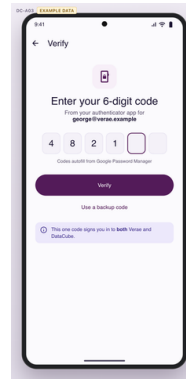
5.2 DataCube — Android



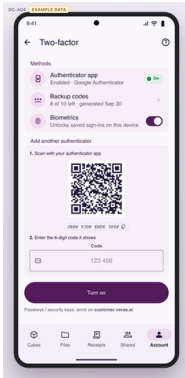
DC-A-JL1



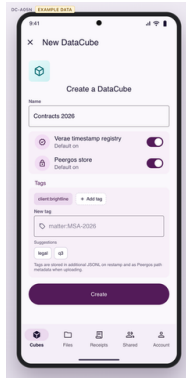
DC-A02



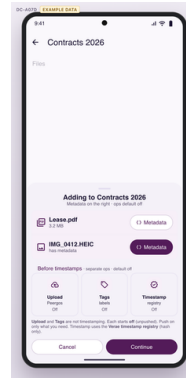
DC-A03



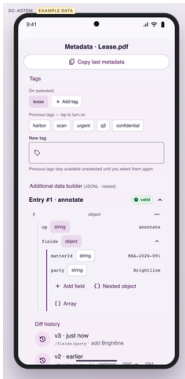
DC-A04



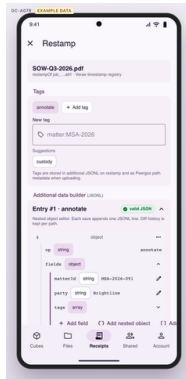
DC-A05N



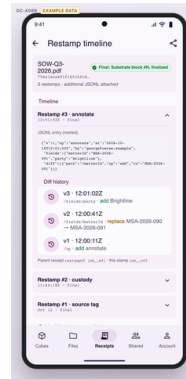
DC-A07D



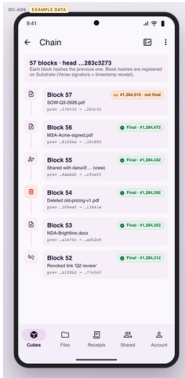
DC-A07DM



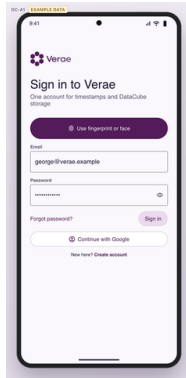
DC-A07R



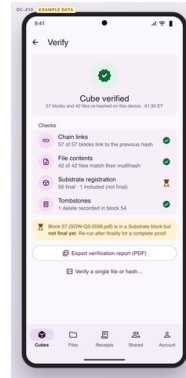
DC-A08R



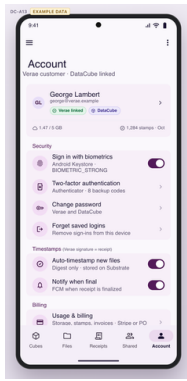
DC-A09



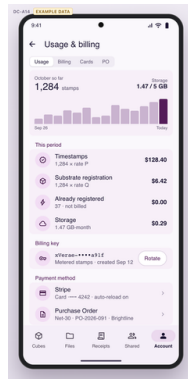
DC-A1



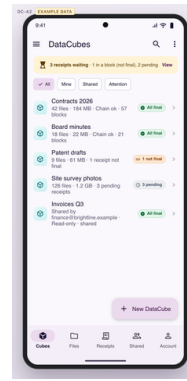
DC-A10



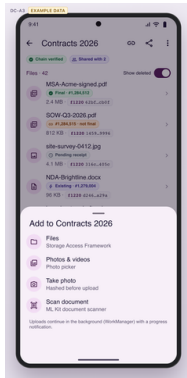
DC-A13



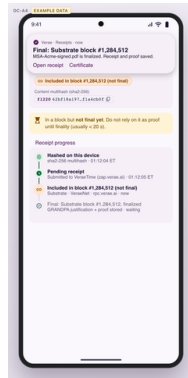
DC-A14



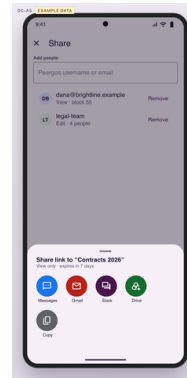
DC-A2



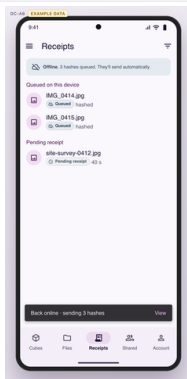
DC-A3



DC-A4

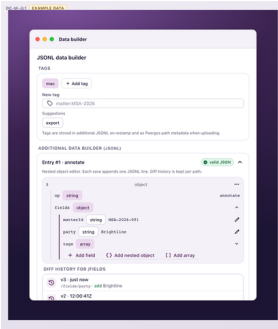


DC-A5

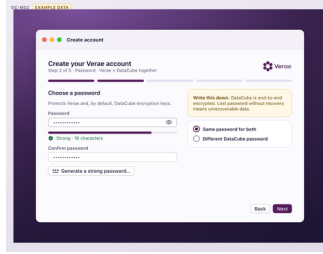


DC-A6

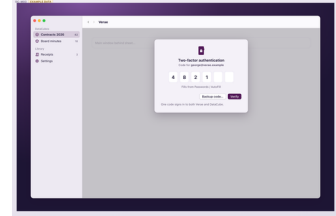
5.3 DataCube — macOS



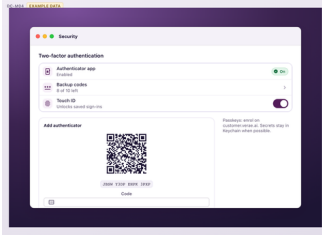
DC-M-JL1



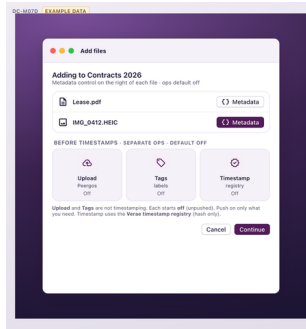
DC-M02



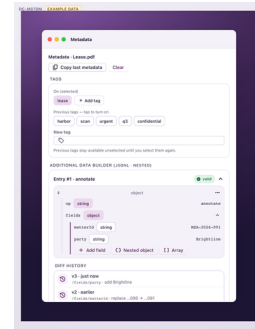
DC-M03



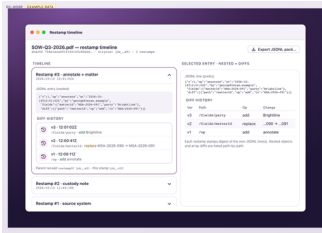
DC-M04



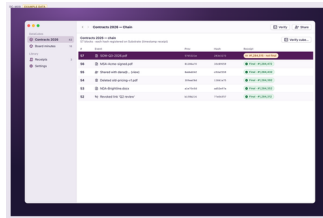
DC-M07D



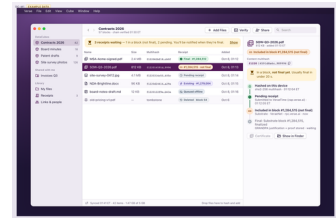
DC-M07DM



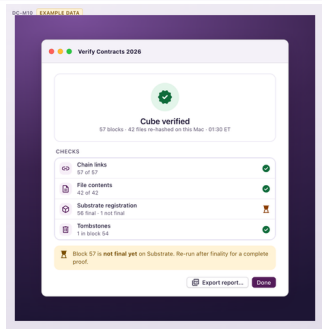
DC-M08R



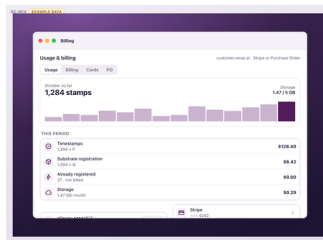
DC-M09



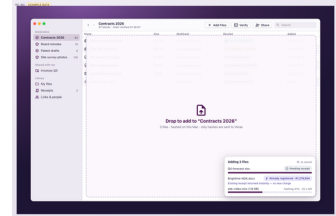
DC-M1



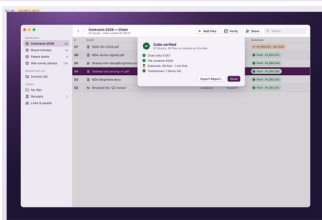
DC-M10



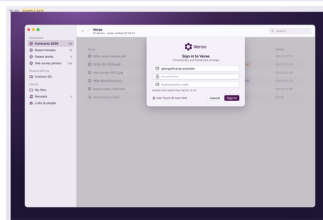
DC-M14



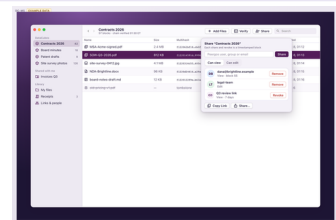
DC-M2



DC-M3

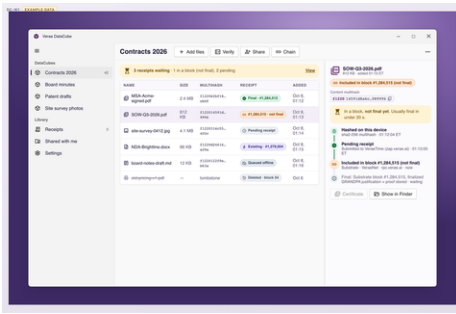


DC-M4

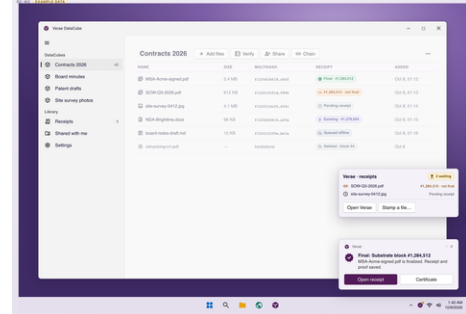


DC-M5

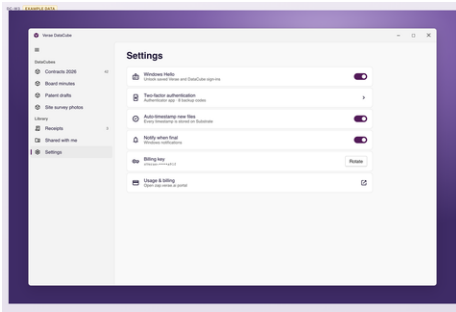
5.4 DataCube — Windows



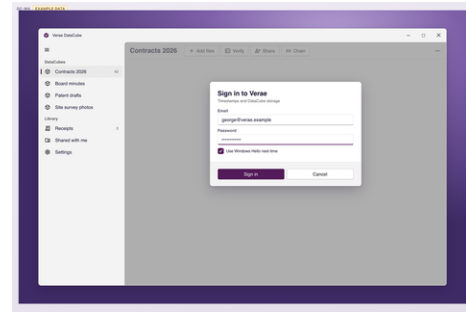
DC-W1



DC-W2

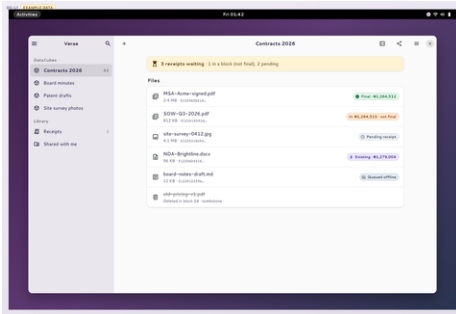


DC-W3

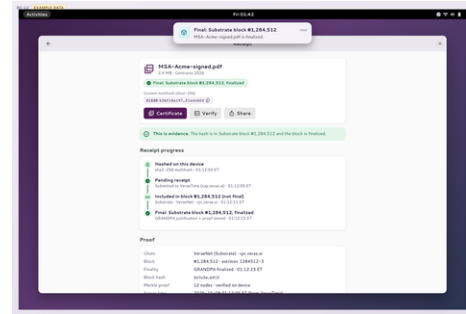


DC-W4

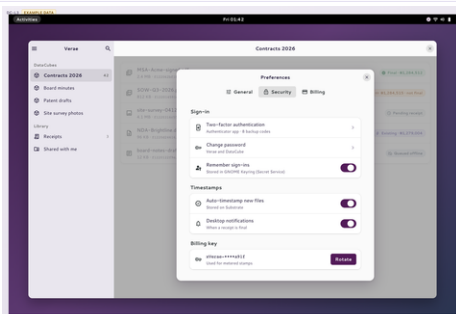
5.5 DataCube — Debian / Linux



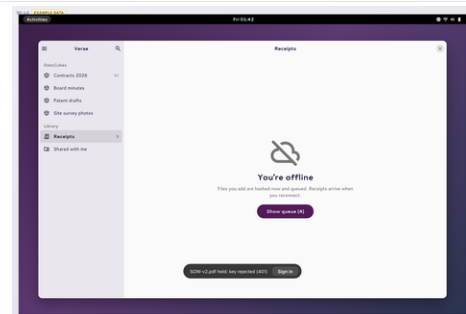
DC-L1



DC-L2

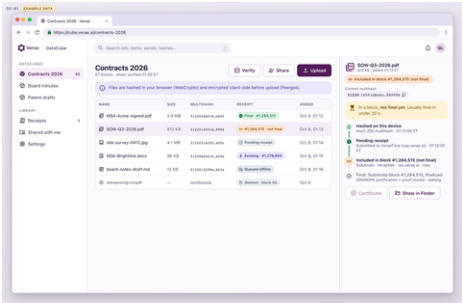


DC-L3

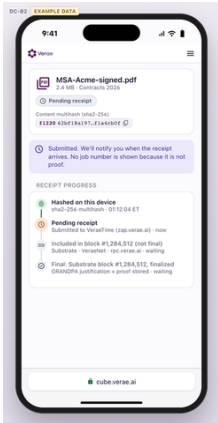


DC-L4

5.6 DataCube — Web proposals



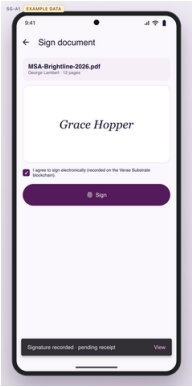
DC-B1



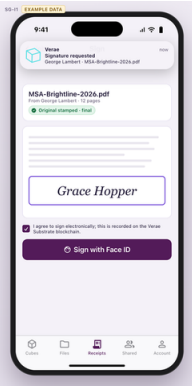
DC-B2

5.7 humanIdentification / Sign

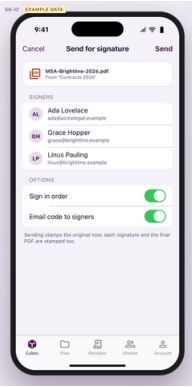
Person attest (draw/type/upload). Digests only on the registry. Distinct from “Verae signature” (= timestamp receipt).



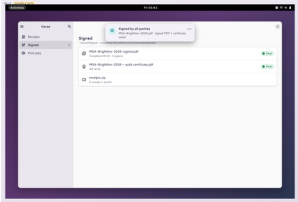
SG-A1



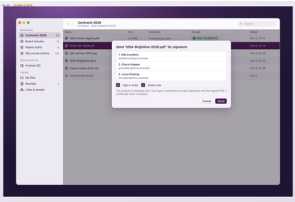
SG-I1



SG-I2



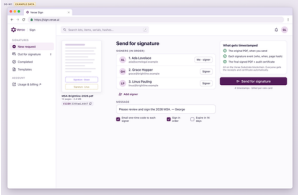
SG-L1



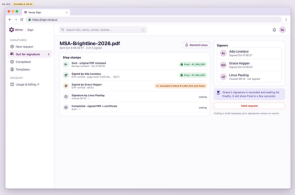
SG-M1



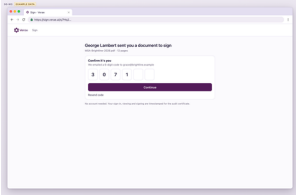
SG-P1



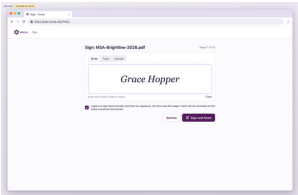
SG-W1



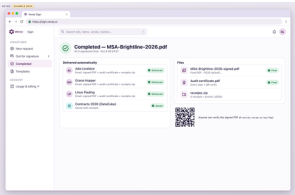
SG-W2



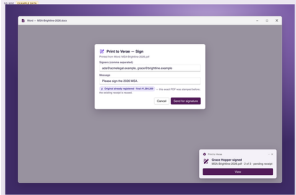
SG-W3



SG-W4



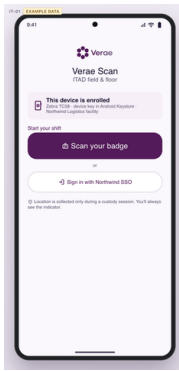
SG-W5



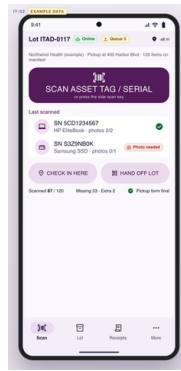
SG-WN1

5.8 ITAD (optional product — ENABLE_ITAD)

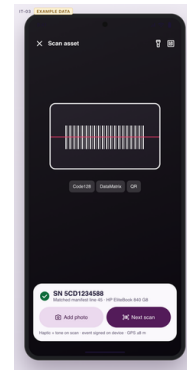
Field logistics and custody. Consumer share builds compile with ITAD off. Screens below are design coverage when enabled.



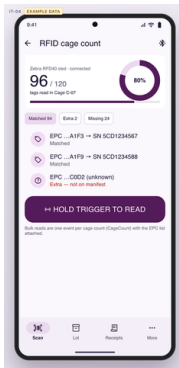
IT-01



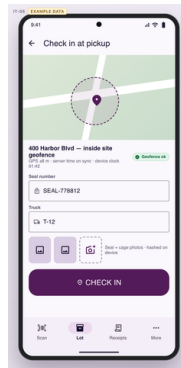
IT-02



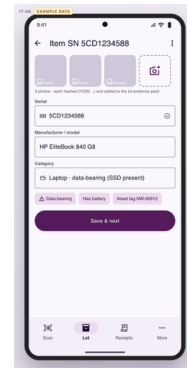
IT-03



IT-04



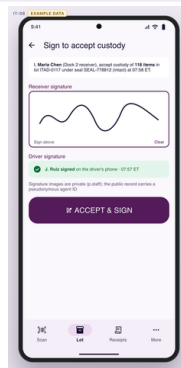
IT-05



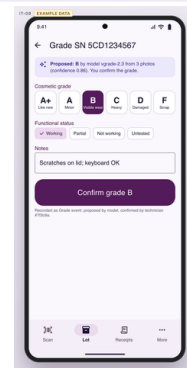
IT-06



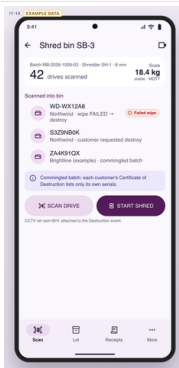
IT-07



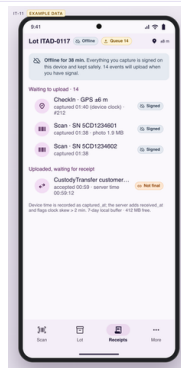
IT-08



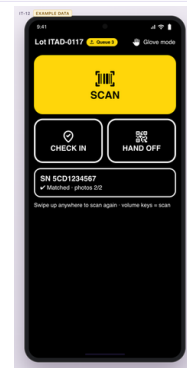
IT-09



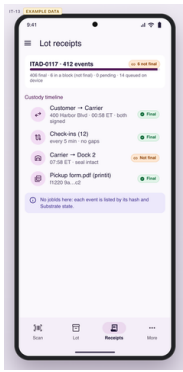
IT-10



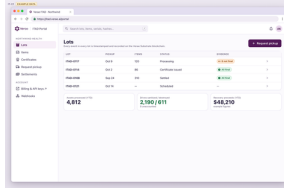
IT-11



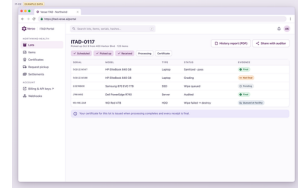
IT-12



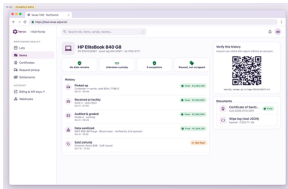
IT-13



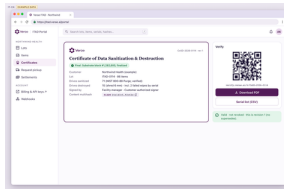
IT-C1



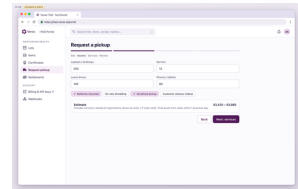
IT-C2



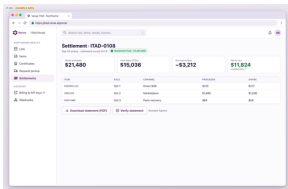
IT-C3



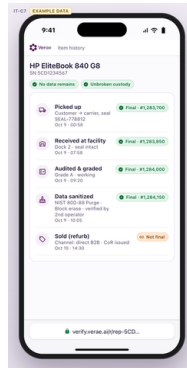
IT-C4



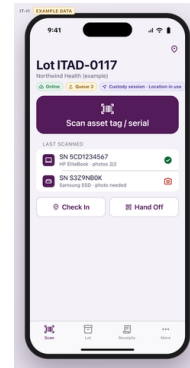
IT-C5



IT-C6



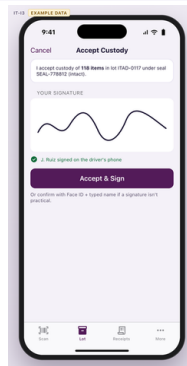
IT-C7



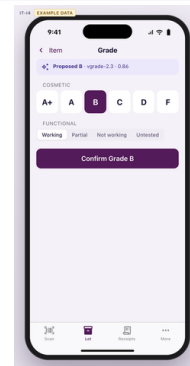
IT-I1



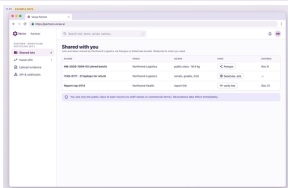
IT-I2



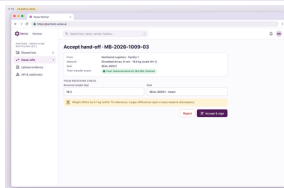
IT-I3



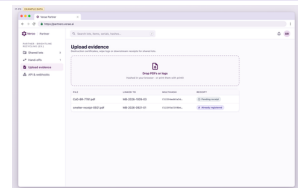
IT-I4



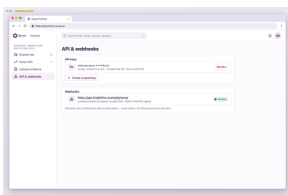
IT-P1



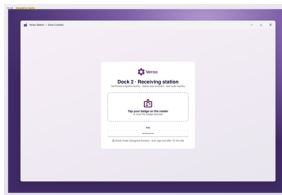
IT-P2



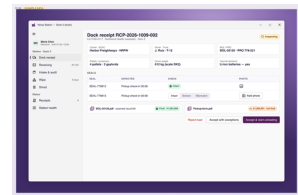
IT-P3



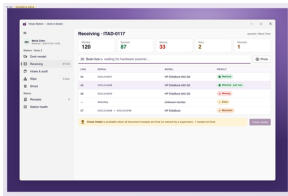
IT-P4



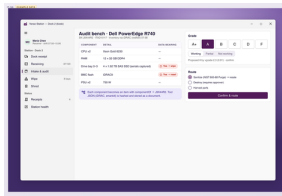
IT-S1



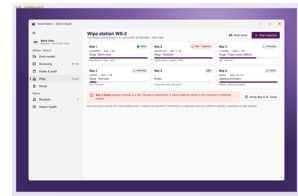
IT-S2



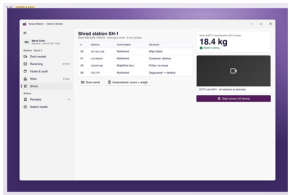
IT-S3



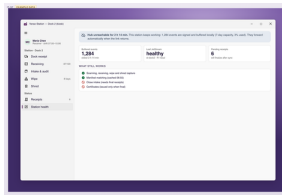
IT-S4



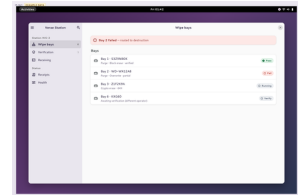
IT-S5



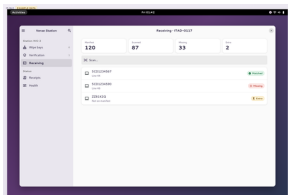
IT-S6



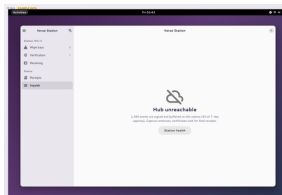
IT-S7



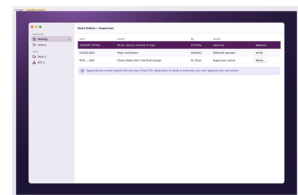
IT-SL1



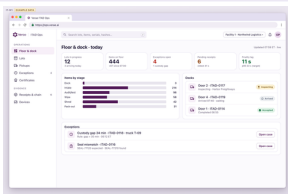
IT-SL2



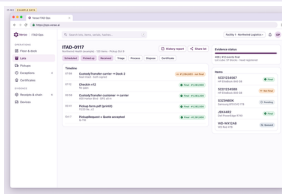
IT-SL3



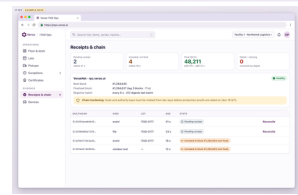
IT-SM1



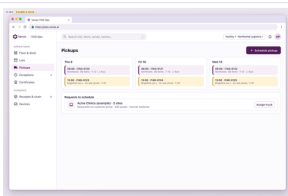
IT-W1



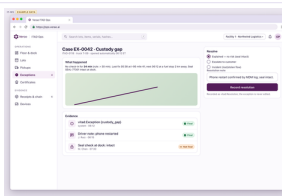
IT-W2



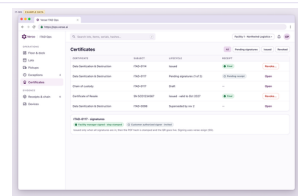
IT-W3



IT-W4



IT-W5



IT-W6



IT-03-v1

6. DataCube storage & secure sharing (Peergos + S3)

6.1 Architecture

DataCubes use **Peergos** for end-to-end encrypted file graphs (cryptree). The commercial plane meters storage and stamps; object bytes live in Peergos-backed storage. Backends include Verae-hosted capacity and optional **customer BYO S3** (endpoint/bucket/region on the customer record).

Layer	Role
Peergos crypttree	E2E encryption, capabilities, paths, social/sharing graph
Cube chain	Ordered blocks: add, share, revoke, tombstone — prior history retained
Verae timestamp registry	Optional digest receipts per file or JSONL metadata line
Hosted S3-class storage	Included GB on Free/Pro/Business; overage vs AWS pin + markup
BYO S3	Customer bucket; ~1 credit/GB-month default + optional passthrough fees

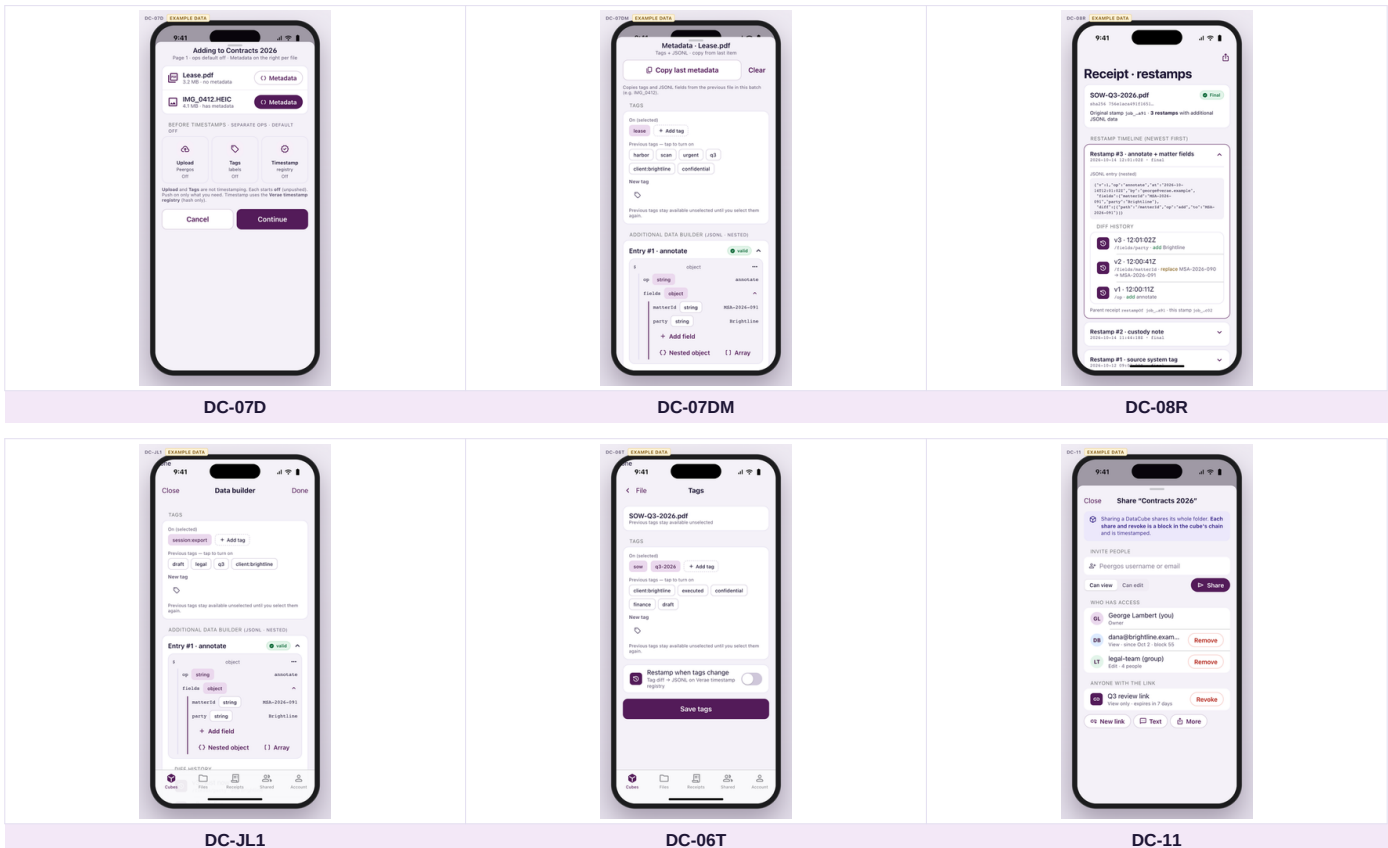
6.2 Sharing scopes

Scope	What is shared	Controls
File	Single object	View/edit capability; expiry; revoke
Directory	Folder subtree	Same capability model
DataCube	Whole cube membership	Invite by email; roles; secret links

6.3 Timestamps, metadata, restamp, versions

- Upload / Tags / Timestamp are separate ops (default off) when adding files.
- Tags: active chips + previous tags (tap to turn on); optional restamp on tag change.
- Metadata: per-file builder; copy last metadata; nested JSONL + path-level diff history.
- Restamp links new receipt via restampOf; original certificate stays valid.

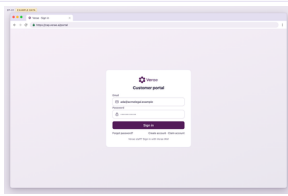
Storage / share / metadata frames



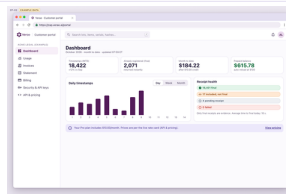
7. Portals, sales, admin, accounting & ops

Customer portal

Web app at customer.verae.ai/portal: signup/login/2FA, profile, xVerae- keys, usage, invoices, Stripe reload, Purchase Order requests, statements.



ZP-01



ZP-02

Service	Usage	Unit	Price	Total
API calls	10,000	calls	\$0.01	\$0.10
Storage	100 MB	MB	\$0.01	\$1.00
Bandwidth	100 GB	GB	\$0.01	\$1.00
API keys	10	keys	\$0.01	\$0.10
API calls	10,000	calls	\$0.01	\$0.10
Storage	100 MB	MB	\$0.01	\$1.00
Bandwidth	100 GB	GB	\$0.01	\$1.00
API keys	10	keys	\$0.01	\$0.10

ZP-03

Invoice	Date	Amount	Status
Invoice 1	2020-01-01	\$100.00	Paid
Invoice 2	2020-02-01	\$200.00	Paid
Invoice 3	2020-03-01	\$300.00	Paid
Invoice 4	2020-04-01	\$400.00	Paid
Invoice 5	2020-05-01	\$500.00	Paid

ZP-04

Statement	Date	Amount
Statement 1	2020-01-01	\$100.00
Statement 2	2020-02-01	\$200.00
Statement 3	2020-03-01	\$300.00
Statement 4	2020-04-01	\$400.00
Statement 5	2020-05-01	\$500.00

ZP-05

Billing	Amount	Status
Billing 1	\$100.00	Paid
Billing 2	\$200.00	Paid
Billing 3	\$300.00	Paid
Billing 4	\$400.00	Paid
Billing 5	\$500.00	Paid

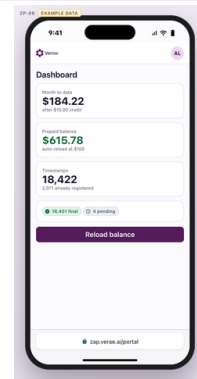
ZP-06

API Key	Key ID	Status
API Key 1	API Key 1	Active
API Key 2	API Key 2	Active
API Key 3	API Key 3	Active
API Key 4	API Key 4	Active
API Key 5	API Key 5	Active

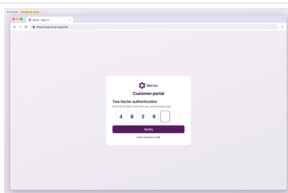
ZP-07

API	Price	Status
API 1	\$100.00	Active
API 2	\$200.00	Active
API 3	\$300.00	Active
API 4	\$400.00	Active
API 5	\$500.00	Active

ZP-08

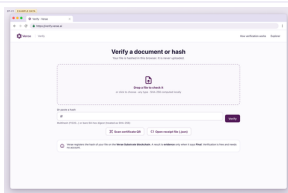


ZP-09

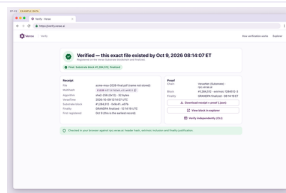


ZP-01-v1

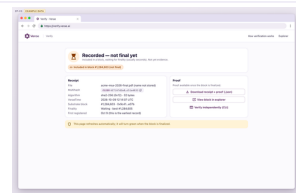
Public verify



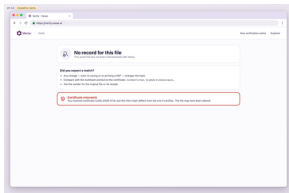
ZP-V1



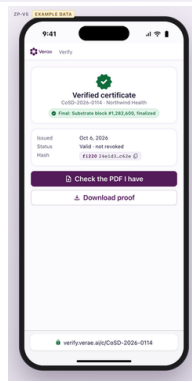
ZP-V2



ZP-V3



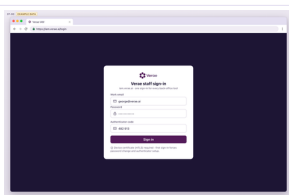
ZP-V4



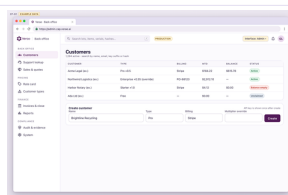
ZP-V5

Staff / admin / sales (back office)

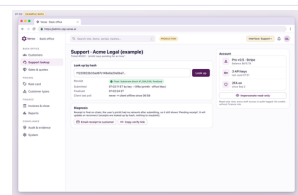
Admin console: customers, tiers, rate cards, per-customer pricing, invoices, exports, support view (no key echo), staff IAM. Sales tools: quotes, packages, segments.



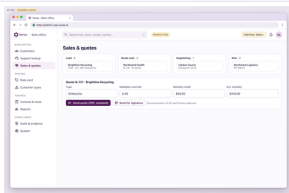
ZP-B0



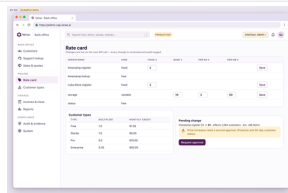
ZP-B1



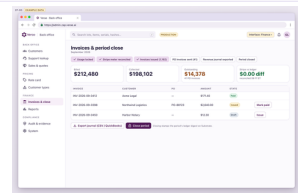
ZP-B2



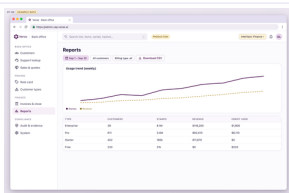
ZP-B3



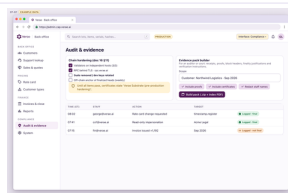
ZP-B4



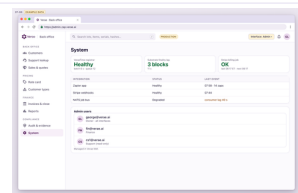
ZP-B5



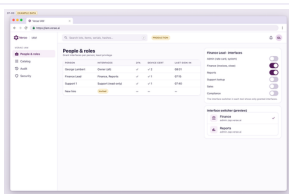
ZP-B6



ZP-B7



ZP-B8



ZP-B9

Accounting & ops

- Accounting: invoices, credit ledger, statements, ledger adjust, export pipelines.
- Ops: edge billing service, middleware/portcullis, worms/trees, chain replica, cold S3/MinIO.
- Health: stack healthcheck, webhook/email alerts, Prometheus textfiles, status page.
- Phone-home / metrics for fleet visibility; job-complete events on core NATS only (no JetStream).

8. Pricing snapshot

Credits ≈ \$0.01 list before tier multiplier. Individual customer pricing via sales/admin overrides. Pay with Stripe or Purchase Order.

Tier	Monthly credits	Stamps/mo incl.	Storage incl.	Multiplier
Free	\$1 (100¢)	20	1 GB	1.0×
Pro	\$10 (1000¢)	250	50 GB	0.5×
Business	\$100 (10000¢)	2,500	500 GB	0.25×

Annual packages: included = monthly×12; price ≈ monthly credits×12×0.8.

Stamp list SKUs (before multiplier)

SKU	List	Notes
timestamp	10¢	Plain; payload may quote 10/16/28
timestamp-re	5¢	Restamp
timestamp-metadata-update	15¢	Metadata update
verify / hash-lookup / receipt	free	Read paths

Storage

Mode	Fee model
Hosted	Included GB on tier; overage vs AWS S3-class pin + markup; compliance hosted higher
BYO S3	~1 credit/GB-month default + optional Amazon passthrough fees

9. Full API reference — parameters, types, sample calls

Commercial host: <https://customer.verae.ai> unless noted. Passwords stored as password_hash. API keys: api_tokens / api_key_hash (not required plain api_key column).

POST /portal/api/signup

Auth: none

Parameter	Type	Req
name	string	yes
email	string	yes
password	string	yes
acceptTos	boolean	yes

Sample call

```
curl -sS -X POST https://customer.verae.ai/portal/api/signup \
-H 'content-type: application/json' \
-d '{"name":"Ada","email":"ada@example.com","password":"CorrectHorse1!","acceptTos":true}'
# 201 {token, apiKey?, customer}
```

POST /portal/api/login

Auth: none

Parameter	Type	Req
email	string	yes
password	string	yes
totpCode	string?	no

Sample call

```
curl -sS -X POST https://customer.verae.ai/portal/api/login \
-H 'content-type: application/json' \
-d '{"email":"ada@example.com","password":"CorrectHorse1!"}'
```

GET/POST /portal/api/profile

Auth: Bearer session

Parameter	Type	Req
Authorization	header	yes
billingType	stripe purchase_order	no
companyName	string?	no

Sample call

```
curl -sS https://customer.verae.ai/portal/api/profile -H "Authorization: Bearer $TOKEN"
curl -sS -X POST .../profile -H "Authorization: Bearer $TOKEN" -H 'content-type: application/json' \
-d '{"billingType":"purchase_order}"'
```

POST /portal/api/2fa/setup|enable

Auth: Bearer session

Parameter	Type	Req
code	string(6) on enable	yes

Sample call

```
curl -sS -X POST .../2fa/setup -H "Authorization: Bearer $TOKEN" -d '{}'
```

```
curl -sS -X POST .../2fa/enable -H "Authorization: Bearer $TOKEN" -d '{"code":"482193}"'
```

POST /portal/api/tokens

Auth: Bearer session

Parameter	Type	Req
name	string?	no

Sample call

```
curl -sS -X POST .../tokens -H "Authorization: Bearer $TOKEN" -d '{"name":"iphone}"'
```

returns plaintext xVerae- once; DB stores hash

GET /portal/api/usage

Auth: Bearer or x-api-key

Parameter	Type	Req
peergosUsername	string?	no

Sample call

```
curl -sS ../usage -H "Authorization: Bearer $TOKEN"
```

GET /portal/api/invoices|statement

Auth: Bearer session

Parameter	Type	Req
(none)	—	no

Sample call

```
curl -sS ../invoices -H "Authorization: Bearer $TOKEN"
```

POST /portal/api/billing/reload

Auth: Bearer session

Parameter	Type	Req
amountCents	number	yes

Sample call

```
curl -sS -X POST ../billing/reload -H "Authorization: Bearer $TOKEN" \
-d '{"amountCents":5000}'
```

POST /portal/api/billing/po-request

Auth: Bearer session

Parameter	Type	Req
poNumber	string	yes
terms	string?	no

Sample call

```
curl -sS -X POST ../billing/po-request -H "Authorization: Bearer $TOKEN" \
-d '{"poNumber":"P0-2026-091","terms":"Net-30"}
```

POST /v1/timestamp

Auth: Bearer xVerae- or x-api-key

Parameter	Type	Req
sha256	string(64 hex)	yes
restamp	boolean?	no
restampOf	string jobId?	no

Sample call

```
curl -sS -X POST https://customer.verae.ai/v1/timestamp \
-H "Authorization: Bearer $XVERAE" -H 'content-type: application/json' \
-d '{"sha256":"e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855"}'
```

Restamp metadata line:

```
curl -sS -X POST https://customer.verae.ai/v1/timestamp \
-H "Authorization: Bearer $XVERAE" -H 'content-type: application/json' \
-d '{"sha256":"<jsonl_sha256>","restamp":true,"restampOf":"<jobId>"}'
```

GET /v1/timestamp/status

Auth: xVerae- or policy

Parameter	Type	Req
sha256	query string(64)	yes

Sample call

```
curl -sS '../v1/timestamp/status?sha256=<64hex>' -H "Authorization: Bearer $XVERAE"
```

POST https://sign.verae.ai/sign/v1/requests

Auth: xVerae- owner

Parameter	Type	Req
documentSha256	string(64)	yes
documentName	string	yes
signers	array<{email,name?}>	yes

Parameter	Type	Req
requireOtp	boolean?	no

Sample call

```
curl -sS -X POST https://sign.verae.ai/sign/v1/requests \
-H "Authorization: Bearer $XVERAE" -H 'content-type: application/json' \
-d '{"documentSha256":"<64hex>","documentName":"msa.pdf","signers":[{"email":"a@b.c"}]}'
```

POST https://sign.verae.ai/sign/v1/public/:token/sign

Auth: public+OTP

Parameter	Type	Req
email	string	yes
otp	string?	no
signatureSha256	string(64)	yes
kind	draw type upload	yes
consent	boolean	yes

Sample call

```
curl -sS -X POST https://sign.verae.ai/sign/v1/public/$TOKEN/sign \
-H 'content-type: application/json' \
-d '{"email":"a@b.c","otp":"123456","signatureSha256":"<64hex>","kind":"draw","consent":true}'
```

Peergos / DataCube bridge (app)

App Rust bridge: login, createCube, upload, listChain, share/revoke. Add-flow flags: uploadPeergos, addTags, timestampRegistry (boolean, default false). Optional POST /v1/timestamp after local SHA-256.

```
ops = {upload:false, tags:false, timestamp:false}
for file in batch:
  meta = file.meta or copy_last_metadata()
  if ops.tags: apply_tags(file, on_tags | previous_selected)
  if ops.upload: peergos.upload(cube, file.bytes)
  d = sha256(file.bytes)
  if ops.timestamp: POST /v1/timestamp {"sha256": d}
  for line in meta.jsonl:
    POST /v1/timestamp {"sha256": sha256(line), "restamp": true, "restampOf": jobId}
```

10. Glossary

Term	Meaning
Verae signature / timestamp	Digest on Verae timestamp registry + receipt/finality
humanIdentification	Person draw/type attest (Sign)
DataCube	Peergos encrypted write-once folder + chain
Restamp	New receipt linked to prior job; JSONL/metadata diffs
PrintIt	Print/share path into the registry + certificate
xVerae-	Customer API token; server stores hash
Stripe / PO	Card prepaid vs Purchase Order
ENABLE_ITAD	Build flag; share builds default false

All screenshots are pre-generated local files from Desktop/Verae-Design-Screens-Local only. No oversized source conversion was performed for this document.

© Verae / FlightLookup Inc. · [customer.verae.ai](#) · [peergos.verae.ai](#)