



RECEIZ PUBLIC RELEASE BOOK

Definitive release, full-system accounting, and institutional thesis



V121.0.0

Source-First Continuity

The proof object is authority. The world can move at full speed beneath it.

Proof that survives its platform.

Speed that never outranks truth.

One coordinated release boundary across the whole system.

RELEASE DATE

August 19, 2026

DEVELOPER SURFACE

@receiz/sdk@121.0.0

EVIDENCE STATUS

Repository qualified

CHAPTER 00

The release in one minute

A system that preserves exact source truth while every representation becomes faster.

REPOSITORY QUALIFIED - EVIDENCE-BOUND

Receiz v121.0.0 locks the final continuity boundary between sealed proof-object authority and its fast global representations. Proof objects, verified identity, ownership, Settlement, Reserve, public witness, and durable local truth remain source. Servers, databases, sessions, caches, indexes, and UI projections distribute and restore that truth without becoming authority.

8,343

tracked files accounted

105

constitutional laws

39

coordinated AI skills

10

required release evidence gates

The breakthrough

The same architecture now produces both lawful authority and a fast settled experience: local proof projection is immediate; global publication is definite and durable; weaker payloads cannot erase stronger known fields.

The evidence boundary

The final tree passed the full release freeze. The immutable tag, package publication, production deployment, signed attestation, and production smoke are not inferred and remain separately unfinished or unknown.

Why this matters

The product no longer has to choose between proof and speed. Truth paints immediately because the proof object already carries it; the network can spend its time distributing additions instead of renegotiating reality.

Evidence: R1 main release record, R2 product truth, R4 checklist, R5 conformance

CHAPTER 01

A map of the definitive book

Human meaning, system architecture, complete accounting, evidence, and economic consequence.

PART	QUESTION ANSWERED	PAGES
I	What exactly changed in v121, and why is it a category boundary?	4-8
II	How does source-first continuity work across proof, identity, public state, and value?	9-20
III	What exists in the repository - files, routes, packages, tests, docs, artifacts, and evidence?	21-32
IV	What is qualified, what is compatible, and what remains unclaimed?	33-38
V	How can the technology create durable institutional and commercial value?	39-46
A	What are the canonical sources, digests, terms, and accounting methods?	47-53

Public-language contract

Implemented primitives are named directly. Proof object is not reduced to card. Durable local truth is not cache. Settlement is not a generic balance. Public proof is not merely a share page. Sealed payload is not generic metadata.

Accounting contract

The PDF summarizes every tracked file family. The companion CSV records every tracked path with Git mode/object, path type, accounting family, extension, byte count, text-line count, and SHA-256. Its digest is printed in this book.

PART I

The release

A source-first system without a source-last experience.

The decisive move is not adding another authority. It is finally refusing to let representations overrule the proof objects they exist to serve.

CHAPTER 02

What v121 changes

Nine locked outcomes across publication, continuity, identity, value, and mobile experience.

#	LOCKED OUTCOME
01	Profile Composer seals caption, long-form, image, video, audio, and supported media through the established proof path.
02	Successful sealing projects into the settled profile immediately; global indexing continues beneath it.
03	Weaker partial projections cannot erase media SHA-256, PBI, signature state, or stable proof identity.
04	Caption-only and long-form proof objects carry deterministic artifact SHA-256 commitments.
05	Verified showcase hydration becomes durable cold-load truth; later lawful proof objects append.
06	Accepted identity proof projects the complete account immediately, not a half-logged-in shell.
07	Fresh signed-in PWA identity can publish policy for its new locally sealed coordinate immediately.
08	Known Settlement and Reserve state paints immediately; newer lawful ledger additions reconcile behind it.
09	The compact mobile bottom bar remains right-aligned and only as wide as its controls.

CHAPTER 02

The category shift

Representations become infrastructure again - powerful, fast, and subordinate.

QUESTION	WEAKER SOFTWARE DEFAULT	RECEIZ V121
What is true?	Whatever the latest server/session/UI state presents.	The sealed or verified proof object under its governing law.
What paints first?	Loading, rediscovery, then correction.	Strongest known truth, already settled.
What does sync do?	Replace the local picture with a canonical snapshot.	Locate and append verified additions after the known head.
What is the database?	The authority that decides whether state exists.	Projection, indexing, distribution, and restoration beneath proof.
What happens offline?	A degraded fallback.	Ordinary proof creation, sealing, inspection, and verification after enrollment.
What is identity?	A session or mutable username.	Stable Receiz ID continuity; username is presentation.
What is value?	A screen balance that waits on refresh.	Known verified Settlement and Reserve truth with append-only reconciliation.

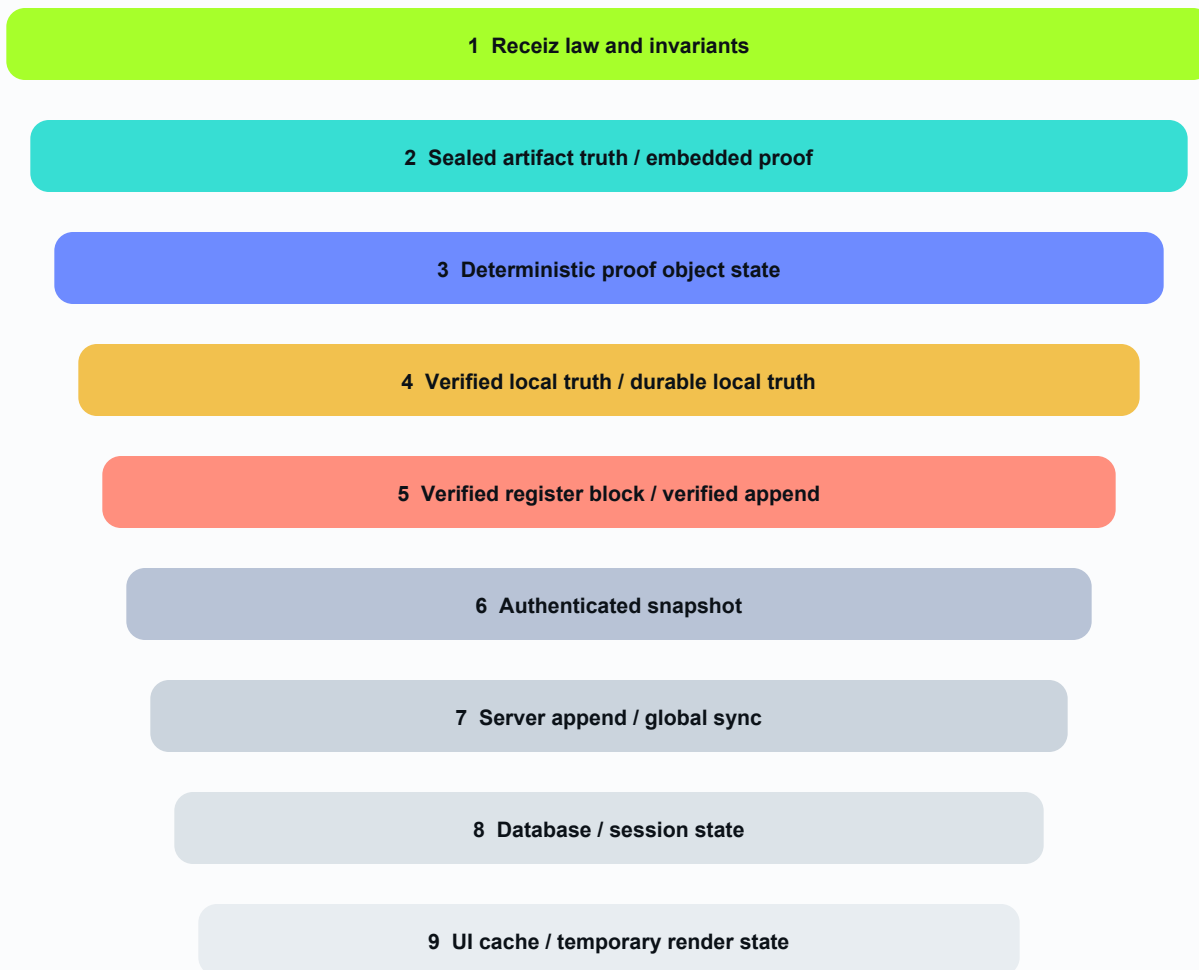
The institutional idea

A government, enterprise, creator, person, or autonomous subject can hold an object whose truth survives platform loss, session loss, network loss, and representational disagreement - while still participating in fast global discovery.

CHAPTER 03

The law in one diagram

A strict authority ladder removes false competition between layers.



Allowed direction

Weaker layers may locate, index, distribute, restore, hydrate, display, or append to stronger truth. They may never replace, erase, delay, rename, or re-authorize it.

Evidence: AGENTS.md truth ladder; R2 frozen invariants; source-never-representation authority law

CHAPTER 03

Why the experience becomes faster

Truthful speed is not an optimization after correctness. It follows from correctness.

- 1** Use strongest known proof immediately
- 2** Paint the settled surface without rediscovery
- 3** Start publication and reconciliation beneath it
- 4** Accept only verified additions after known heads
- 5** Persist the newly verified head as durable cold truth
- 6** Open the next visit from that head - not from loading

Result

The page stops feeling like it loads again because known truth is no longer treated as uncertain. Background work becomes additive rather than corrective.

PART II

The system

One proof-native architecture across media, identity, value, public state, and autonomous subjects.

Receiz is not a generic application with proof features. It is an artifact system whose products are bridges to proof, identity, ownership, Settlement, provenance, and public witness primitives.

CHAPTER 04

The proof-object lifecycle

One admission boundary. Every later change is a verified append or projection.

STAGE	OPERATION	WHAT IT GUARANTEES
1	Create	Caption, long-form, image, video, audio, document, identity, world event, or value event.
2	Seal	Bind deterministic identity, payload, SHA-256, provenance, signature, and governing law.
3	Verify	Recompute artifact integrity and primitive-specific constraints without trusting UI state.
4	Admit	Accept once against exact current heads and authority; rejection writes nothing.
5	Project	Paint local, public, profile, vault, market, world, identity, or Settlement surfaces.
6	Append	Publish verified additions locally and globally without rewriting the admitted source.

Durable rule

First admission only, then append forever. Refresh, hydration, sync, and UI reconstruction are never substitutes for a second admission of known truth.

CHAPTER 05

Profile Composer is a proof pipeline

Every supported public expression travels through the established proof path.

Inputs



What the seal carries

- Stable proof-object identity and governing law
- Real media SHA-256 when uploaded media exists
- Deterministic artifact SHA-256 for caption-only and long-form objects
- PBI, signature state, provenance, and public projection material
- Artifact manifest and live-player commitment projection

What happens after success

- The proof object projects into the settled profile immediately
- No native form submission, reload, or remount may redefine the surface
- Global composite publication starts beneath the settled proof
- Durable retry preserves definite publication when indexing is unavailable
- Weaker responses merge monotonically and cannot erase verified fields

The public consequence

Publishing feels instantaneous without becoming speculative. The user sees the proof object because it exists; the index catches up because it is a representation.

CHAPTER 06

Global publication without global authority

The composite head is a proof object carrying append-only public state.

LAYER	ROLE	FORBIDDEN POWER
Sealed proof object	Carries source truth, payload, identity, integrity, provenance, and law.	None beneath it may decide that already sealed truth is absent.
Composite head	Carries append-only public state and discoverability as another verified proof object.	May not rewrite the source object or erase historical visibility state.
Server	Locates, distributes, backfills, restores, and appends exact verified objects.	May not become permission for deterministic first paint.
Database	Indexes custody, public discovery, profile projections, and bounded lookup paths.	A missing or partial row may not negate stronger proof.
Session	Adds authenticated permissions and owner controls.	May not define stable identity or block public proof paint.
UI	Renders the strongest known projection and verified additions.	May not reduce proof objects to temporary display state.

Definite but subordinate

Global publication is not optional enrichment. It begins immediately and retries durably. Its certainty does not require elevating the index above the object being indexed.

CHAPTER 07

Monotonic truth merging

A smaller payload is not permission to forget a larger verified object.

STRONG FIELD ALREADY KNOWN	WEAKER UPDATE ARRIVES	V121 RESULT
Media SHA-256	Projection omits or nulls the hash	Preserve the verified hash.
PBI / stable proof identity	Response carries presentation-only identity	Preserve the stable identity boundary.
Signature state	Partial public object has no signer detail	Preserve verified signature state.
Ownership / custody	Index row is delayed or incomplete	Preserve lawful known custody; append only verified change.
Settlement / Reserve	Refresh temporarily lacks entitlement or ledger detail	Paint known verified value; reconcile additions behind it.
Showcase head	Network returns an older or partial page	Keep durable verified head; admit only lawful later append.
Restored account	Stale session response names the previous account	Keep the proof-accepted account active.

Merge law

Strength is field-specific and proof-specific. Verified additions may enrich known truth; omission, latency, or representational lag cannot delete it.

CHAPTER 08

Durable cold-load truth

Hydration discovers verified additions once; it does not perform the same proof work on every visit.

Repeated discovery

- Open profile
- Show loading or partial shell
- Refetch known objects
- Rebuild projection
- Correct the surface
- Repeat next visit

V121 continuity

- Open profile
- Paint durable verified head
- Request additions after known head
- Verify and append
- Persist new head
- Next visit opens from it

Visibility is not deletion

Hiding a lawful proof object changes its current public visibility projection. It does not rewrite the object's existence, provenance, or admitted history.

Performance follows ontology

Cold loads become fast because durable verified truth is treated as truth, not because the system cuts corners on verification.

CHAPTER 09

Identity continuity

Stable Receiz ID is identity. Session and username are mechanics beneath it.

EVENT	PROOF ACCEPTED	IMMEDIATE VISIBLE RESULT	BACKGROUND WORK
PBI / passkey recovery	Legacy account + local device identity	Recovered account becomes active locally	Idempotent Receiz ID bind and global mirror
Identity Record / Seal restore	Portable identity proof	Same complete signed-in account experience	Import, bind, sync, wallet/profile refresh
Receiz Key online restore	Fresh private-key proof and active registration	New recovered-account session is consumed immediately	Portable-state enrichment and global propagation
Receiz Key offline restore	Locally verified key proof	Recovered account projects immediately	Server recovery starts when available
Username change	Presentation change bound to stable identity	Label changes without identity rewrite	Indexes update beneath stable Receiz ID

No half-login state

Once identity proof is accepted, account, profile, wallet, upgrade, bottom chrome, and account switcher must agree before import, bind, or sync jobs finish.

CHAPTER 10

Ownership, custody, and fresh creation

A missing servant row cannot veto a newly sealed coordinate; a conflicting owner still binds.

CASE	SOURCE TRUTH	DECISION
New locally sealed coordinate; accepted signed-in identity	Identity proof + locally sealed proof object	Allow creation, custody, and download policy immediately.
Database ownership row absent	Absence of projection evidence	Do not convert absence into denial.
Existing owner matches	Verified custody projection consistent with proof	Proceed and append/index normally.
Existing conflicting owner	Verified binding conflict	Block; the conflict is an ownership-risk boundary.
Database row arrives later	Representation of already lawful outcome	Distribute and accelerate lookup; do not retroactively authorize.

Exact rule

Accepted identity authorizes creation of a new unowned coordinate. Existing conflicting ownership blocks it. Later database projection distributes the result without becoming proof authority.

CHAPTER 11

Settlement and Reserve

Proof-native value must remain visible while lawful additions reconcile.

KNOWN verified Settlement truth	FIRST paint before discovery	APPEND new lawful ledger events	NEVER temporary generic unavailable
---	--	---	---

Settlement primitive

Settlement represents proof-native funded events, assigned value, governed transfer, certificate or note value, and terminal economic state. It is not merely a UI balance.

Reserve primitive

Reserve carries known verified value and entitlements, including Glyph-entitled accounts. It must paint from strongest known truth while later ledger additions reconcile behind it.

Value continuity

A network delay cannot make admitted value disappear. A smaller response cannot erase stronger known Settlement or Reserve truth. A later lawful event may append a new state; it does not make the prior state untrue.

Terminal-state law

Obligation-bearing flows require proof-native completion, cancellation, claim, transfer, or settlement outcomes. The system protects consent and exit against perpetual rollover.

CHAPTER 12

Offline proof is the primitive

After one canonical Signature V4 enrollment, Record, Seal, and Verify do not require login or network.

SURFACE	LOCAL OPERATION	AUTHORITY MATERIAL
/record	Create a proof object	Local Receiz ID + canonical V4 signer
/seal	Seal supported file carriers	Artifact bytes + embedded proof + device certificate
Offline Studio	Record, Seal, Verify	Pinned root + local signer + carried proof
Public proof surface	Inspect and verify	Proof object and recomputable payload
Historical object	Verify under original law	Exact sealed bytes and historical governing law

Separate identities

Receiz ID is deterministic local device identity. Signature V4 is the artifact-signing key and canonical root-signed device certificate. They compose; neither derives from nor replaces the other.

Survival property

The artifact remains legible when copied, exported, inspected offline, or separated from the origin system. The application assists verification; it does not make the object true.

CHAPTER 13

Living subjects remain proof-native

V121 coordinates current admission law without weakening v120 living-subject continuity.

CAPABILITY	CONTINUITY GUARANTEE
Immutable subject identity	The subject remains the same proof object across apps, owners, providers, and sessions.
Proof brain and event-derived memory	Memory cites admitted events and complete content-addressed history.
Twin expression	AI can speak, reason, and propose without deciding canonical world events.
Deterministic world runtime	Exact-head commands and atomic transactions decide what happened.
Bounded mandate	Autonomy is authorized, scoped, inspectable, and revocable.
Ownership transfer	Authority changes; identity and history continue.
Current v121 receiver	Historical receipts remain evidence, but current admission requires exact-byte reverification and current profile authority.

AI boundary

The model may create expression and intention. It cannot invent reality. Only typed deterministic Receiz operations can admit canonical events.

CHAPTER 14

One coordinated developer surface

SDK, MCP, AI skills, registry, docs, app, and service worker carry the same v121 identity.



BOUNDARY	V121 CONTRACT
Registry	Digest 29a793a5bcc0195a... chains from v120 0728651789b26e1d...
Package range	>=121.0.0 <122.0.0 across coordinated AI manifests
Application operations	Exact operation inventory describes authority; identifiers do not create authority.
Compatible HTTP routes	Existing /api/receiz/v120 protocol routes remain supported.
Historical proof	Objects verify under exact sealed bytes and original law; no rewrite required.
Visible versions	Package, docs, app, service worker, release surface, and ruleset identity are locked together.

Full accounting

Every tracked file is represented in the companion manifest; every major family is explained here.

Scale is not evidence by itself. The accounting matters because it shows where the laws live, how they are enforced, and how much of the system is already executable rather than aspirational.

CHAPTER 15

Repository accounting at a glance

Exact tracked-tree totals at PDF generation time.

ACCOUNTING DIMENSION	COUNT
Application pages	193
Application API route files	487
Markdown doctrine / documentation files	1,009
Package manifests	20
Root npm/pnpm scripts	175
Root test scripts	122
AI skills / skill manifests / OpenAI agent prompts	39 / 33 / 30
Current MCP artifact tools / living-subject tools	9 / 37

Exact file-level companion

receiz-v121-full-file-accounting.csv contains every tracked path, Git mode/object, path type, family, extension, bytes, text lines, and SHA-256. CSV SHA-256: 2e998619c73b948a6ce260c0c0a88621512a57d253b6bae431ae2dd740c886c8

CHAPTER 16

Tracked-file families 1

Every file belongs to one explicit accounting family.

ACCOUNTING FAMILY	FILES	BYTES	TEXT LINES
Application surfaces and runtime	2,118	62.4 MB	914,698
Evidence, generated output, and retained QA	1,358	903.5 MB	1,244,438
Public Academy and release archive	751	69.4 MB	256,798
Doctrine, release, and engineering records	734	13.6 MB	219,249
Coordinated packages	665	10.6 MB	254,156
Application tests and contracts	597	6.9 MB	144,792
Application API routes	527	4.4 MB	123,250
Static public and offline artifacts	445	2.3 GB	148,280

Application dominance

The tracked tree is centered on executable application surfaces, APIs, contracts, and coordinated packages. Generated evidence is retained but isolated as its own family.

CHAPTER 16

Tracked-file families 2

Every file belongs to one explicit accounting family.

ACCOUNTING FAMILY	FILES	BYTES	TEXT LINES
Database projections and migrations	293	2.3 MB	66,525
PBI identity subsystem	238	20.3 MB	46,306
AI skill distribution	227	1.2 MB	34,670
Release, generation, and contract tooling	222	2.1 MB	48,068
Repository configuration and other	71	3.7 MB	34,558
Shared runtime, models, and fixtures	63	539.9 KB	9,993
Zero-knowledge proof material	34	11.1 MB	2,099

Authority-bearing families

Doctrine, release records, SDK/MCP/AI packages, PBI, offline artifacts, database projections, and ZK material are accounted separately because they carry different proof boundaries.

CHAPTER 17

Top-level repository families

Where the tracked tree lives.

TOP-LEVEL PATH	FILES	BYTES	TEXT LINES
app	3,242	73.7 MB	1,182,740
output	987	653.2 MB	622,945
receiz-docs	751	69.4 MB	256,798
docs	734	13.6 MB	219,249
packages	665	10.6 MB	254,156
public	445	2.3 GB	148,280
supabase	293	2.3 MB	66,525
.playwright-cli	250	181.0 MB	617,665
PBI	238	20.3 MB	46,306
ai-skills	227	1.2 MB	34,670
scripts	222	2.1 MB	48,068
tmp	117	69.4 MB	3,567
zk	34	11.1 MB	2,099
fixtures	24	17.7 KB	460
models	13	7.9 KB	258
src	13	212.7 KB	6,137

CHAPTER 18

Formats and implementation matter

The repository combines executable code, proof material, public media, documentation, and retained evidence.

FORMAT	FILES	BYTES	TEXT LINES
.ts	3,045	31.8 MB	839,319
.png	1,312	793.0 MB	0
.md	1,009	7.4 MB	156,923
.tsx	972	15.4 MB	395,505
.json	505	1.5 GB	1,058,978
.sql	297	2.4 MB	68,778
.js	151	6.3 MB	114,171
.css	146	4.0 MB	179,691
.yml	134	8.1 MB	166,040
.map	117	463.1 KB	117
.gz	117	842.4 MB	0
.log	113	162.7 MB	453,197
.svg	72	2.0 MB	6,205
[none]	49	1.0 MB	781
.mjs	44	599.3 KB	15,915
.yaml	36	523.6 KB	16,402
.pdf	24	4.2 MB	7,553
.py	23	615.4 KB	13,329
.html	20	5.8 MB	48,900
.webp	18	28.4 MB	0

CHAPTER 19

Application surfaces and APIs

The product is a broad proof-native operating surface, not one isolated feature.

193
page routes

487
API route files

PUBLIC
proof and witness

PRIVATE
identity and custody

SURFACE FAMILY	PRIMITIVES CARRIED
Profile, public profile, public twin	Proof objects, authorship, provenance, public witness, identity
Record, Seal, Offline Studio	Receiz ID, Signature V4, sealed artifacts, offline verification
Vault, market, certificate, note	Ownership, custody, transfer, Settlement, reserve, public proof
Wallet, upgrade, account	Settlement, Reserve, entitlement, identity continuity, recovery
World, sports, game, subject	Deterministic events, Kai order, proof memory, ownership continuity
API routes	Verification, projection, admission, session permissions, distribution, indexing
Legal and trust surfaces	Published product law, compliance claims, evidence boundaries

CHAPTER 20

Contracts, tests, and release gates

The repository makes regression mechanically expensive.

919
test/contract files

122
root test scripts

FREEZE
umbrella release gate

EVIDENCE
visual truth overrides

GATE FAMILY	WHAT IT PROTECTS
v121 release lock	Versions, registry digest, operation matrix, AI manifests, docs, visible versions, service worker, freeze membership
Deterministic surfaces	Known truth and first paint remain settled
Primitive binding	Existing proof/identity/ownership/Settlement authority is reused rather than bypassed
Profile continuity	Immediate local projection, no remount, strongest-field preservation, global append
Settlement and verification conformance	Proof-native value and recomputable verification remain exact
Package and coordinated identity	SDK, MCP, AI Skills, docs, and ruleset agree
Governance controls	Evidence, artifact signatures, and release decisions remain inspectable
Visual evidence contract	Contradictory screenshots or DOM evidence fail the claim even if code tests pass

CHAPTER 21

Doctrine and release records

Product law is preserved as executable review context, not informal institutional memory.

RECORD FAMILY	ROLE
Reasoning kemels	Compress reality-grade, invariant-first, Kai/Chronos, correction, temporal, rollover, and artifact doctrine.
Literal product law	Names exact primitive and first-paint boundaries.
Release product truth	Freezes primitives, invariants, results, gates, and residual risks per version.
Regression lessons	Records the reasoning inversion that caused visible continuity failures.
Compatibility and migration	Protects historical sealed objects and stable protocol routes.
Conformance and checklist	Separates repository qualification from publication, deployment, attestation, and smoke.
Commit history	Names implementation lineage and the exact unresolved repository-boundary commit.
Academy release archive	Projects release claims and source bindings into a public documentation surface.

Doctrine is not decoration

The release book is derived from repo-local law and evidence. It does not import generic software ontology to decide what Receiz primitives are.

CHAPTER 22

Packages, protocols, and agent surfaces

A coordinated implementation boundary for humans, applications, tools, and AI agents.

SURFACE	PUBLIC ROLE	V121 BOUNDARY
@receiz/sdk	Verification, proof operations, artifact transitions, public projection, identity, custody, and runtime primitives	121.0.0
@receiz/mcp-server	Typed agent/tool access to artifact, subject, world, memory, and transfer operations	121.0.0
@receiz/ai-skills	Lawful operational instructions and manifests for coordinated agent behavior	121.0.0
HTTP protocol	Compatible application integration surface	Existing /api/receiz/v120 routes retained
Constitution registry	Immutable current law inventory chained to predecessor digest	105 laws / Kai 13,725,000
Application-operation matrix	Names operations and required authority without creating authority	Digest-bound current inventory

Agent-safe architecture

AI skills and MCP tools can expose operations, plans, and reports. They do not let model output become proof, identity, ownership, Settlement, or world-event authority.

CHAPTER 23

Database and projection accounting

The database is maximally useful precisely because it is not asked to be the proof.

DATABASE / SERVER FUNCTION	LAWFUL USE	REGRESSION IF PROMOTED
Discovery index	Locate public proof objects and composite heads	Missing row treated as proof object absence
Custody projection	Accelerate ownership lookup and conflict detection	Row demanded before lawful fresh creation
Session projection	Grant private controls to a proof-accepted identity	Session defines or delays stable identity
Global append	Distribute verified additions and retry durably	Server confirmation gates settled local projection
Hydration source	Return possible additions after known head	Partial snapshot replaces durable verified truth
Settlement index	Locate later lawful ledger events	Temporary omission erases known Settlement/Reserve
Analytics/ops	Measure behavior and route investigation	Dashboard status becomes product truth

CHAPTER 24

Cryptographic and artifact carriers

Verification must bind the enclosing stronger artifact, not a self-hashable fragment.

CARRIER / MATERIAL	ROLE
PNG	Receiz proof bundle in PNG text chunk
PDF	Canonical Receiz PDF seal update
SVG	Deterministic SVG metadata
JSON	Deterministic trailing whitespace channel
JPEG and trailer-safe files	Receiz trailing marker with carrier-specific exact-byte law
Signature V4	Local artifact signer with canonical pinned-root device certificate
Groth16 / ZK material	Recomputable proof surfaces and preserved historical law
Registry digest chain	Current constitutional law identity and predecessor continuity

Stronger artifact first

Verify the enclosing sealed artifact or proof object before projecting an embedded subpayload. A subpayload hash validates projection; it does not silently become admission, custody, access, or Settlement authority.

PART IV

Evidence

Repository qualification is exact. Public release status remains multidimensional.

The credible institutional posture is simple: claim every implemented fact directly, and leave every unperformed external action visibly unclaimed.

CHAPTER 25

The qualification ledger

What is complete and what is still a separate action.

DIMENSION	STATUS	WHAT MAY BE CLAIMED
Repository release records	Documented	The v121 source family and public archive record exist.
Full release freeze	Passed	Repository behavior qualified on the final release tree.
Coordinated package artifacts	Packed for Academy	Exact SDK, MCP, and AI skill artifacts exist in the repo.
Immutable repository tag	Not created	No v121.0.0 tag claim.
Package publication	Not performed	No registry publication claim.
Production deployment	Unknown	No deployment inference.
Production smoke	Not performed	No live-production pass claim.
Signed attestation	Unknown / not evidenced	No signed-governance completion claim.

Repository qualified

This book can say the full release freeze passed and the coordinated v121 boundary exists in the repository. It cannot turn that fact into publication, deployment, customer adoption, revenue, or external certification.

CHAPTER 26

Required release evidence

The canonical v121 conformance list.

```
pnpm test:v121-release-lock
```

```
pnpm test:coordinated-release-identity
```

```
pnpm test:continuity-release-contracts
```

```
pnpm test:local-truth-release-contracts
```

```
pnpm test:profile-release-contracts
```

```
pnpm test:settlement-conformance
```

```
pnpm test:verification-conformance
```

```
pnpm test:release-freeze
```

```
pnpm governance:check
```

```
pnpm governance:evidence-check
```

Evidence separation

These gates qualify repository behavior. Publication, deployment, production smoke, and signed attestation must each produce their own evidence.

CHAPTER 27

Compatibility without rewriting history

Current coordinated identity and historical proof law are different boundaries.

BOUNDARY	V121 BEHAVIOR
Historical proof objects	Unchanged; verify under sealed bytes and original law.
Identity Seal / Receiz Key	Accepted proof restores the same complete account immediately.
Receiz ID / username	Stable Receiz ID remains identity; username remains mutable presentation.
HTTP protocol routes	Compatible /api/receiz/v120 routes remain supported.
SDK / MCP / AI Skills	Coordinate at 121.0.0 with range >=121.0.0 <122.0.0.
Showcase	Verified public appends become globally discoverable and durable.
Private proof data	Restore strongest verified local/artifact truth plus lawful additions.
Settlement / Reserve	Known verified value paints immediately; lawful additions append.
Custom domains	Use the same composite-publication path and stable identity binding.

CHAPTER 28

Migration is preservation

No proof-object rewrite is required.

Coordinated adoption

- Update @receiz/sdk, @receiz/mcp-server, and @receiz/ai-skills together to 121.0.0.
- Continue using compatible v120 HTTP protocol routes.
- Do not transform historical sealed objects.
- Do not transform usernames into identity.
- Do not transform server snapshots into proof authority.
- Merge profile projections by strongest verified field.
- Persist a newly verified composite head as durable cold truth.
- Paint known Settlement and identity state first; reconcile additions in background.

Migration principle

The v121 transition coordinates current receivers and representations. It does not invalidate or rewrite the exact-byte history the system exists to preserve.

CHAPTER 29

Regression anatomy

The failure was applying source-first law backward.

VISIBLE REGRESSION	PROMOTED WEAKER LAYER	LOCKED CORRECTION
Composer payload rejected	Route/server confirmation	Established proof path seals first.
Media proof fields missing	Partial projection	Strongest-field monotonic merge.
Page refresh after publication	Form/navigation mechanics	Immediate local projection; background global append.
Slow profile transitions	Repeated discovery	Durable cold-load verified head.
Wallet temporarily unavailable	Fresh representation	Known Settlement/Reserve paints first.
Fresh PWA ownership denied	Absent DB row	Accepted identity may create new unowned coordinate.
Half-logged-in restore	Old session and sync work	Proof-accepted account projects completely.
Historical/current confusion	Version label	Original law verifies history; v121 governs current receiver.

PART V

Value capture

Why source-first continuity can become a durable institutional and commercial layer.

The economic thesis is not that proof makes money by declaration. It is that independently verifiable continuity removes expensive trust, reconciliation, custody, provenance, and platform-dependence failures across many markets.

CHAPTER 30

The inevitability thesis

Not a guarantee - a set of conditions under which adoption pressure compounds.

IF THIS BECOMES NON-NEGOTIABLE	THEN RECEIZ ALREADY CARRIES THE NEEDED PRIMITIVE
AI agents must act without inventing canonical reality	Typed deterministic operations, exact heads, atomic admission, bounded mandate
Digital media must prove source and survive platform loss	Sealed artifacts, embedded proof, SHA-256, offline verification, public witness
Identity must survive device, account, and provider transitions	Receiz ID, PBI recovery, Identity Seal/Record, Receiz Key continuity
Ownership must transfer without rewriting the object	Stable artifact identity, custody state, provenance, transfer and settlement routes
Value must remain visible through network and representation failure	Known Settlement and Reserve truth with append-only reconciliation
Institutions must audit claims independently	Recomputable proof, registry law, evidence-bound release process
Applications must be fast without centralizing authority	Deterministic first paint plus subordinate global distribution

What makes it feel inevitable

The same primitive set answers pressures arriving from AI autonomy, synthetic media, digital identity, portable ownership, offline resilience, and institutional auditability. One system serves all six without inventing a new authority per product.

CHAPTER 31

Who pays for which outcome

Value capture begins where verification prevents recurring operational loss.

BUYER / OPERATOR	PAID OUTCOME	RECEIZ PRIMITIVE
Creators and rights holders	Portable authored media, provenance, public verification, transfer continuity	Proof object + public proof + ownership
Enterprises	Audit-ready records, identity continuity, deterministic agent actions	Artifact + identity + event admission
Institutions and governments	Offline-verifiable records, durable evidence, stable identity, inspectable authority	Sealed artifact + registry + public witness
Marketplaces	Custody, provenance, listing, transfer, funded purchase, certification	Ownership + Settlement + certificate/note
AI platforms and games	Portable subjects, bounded autonomy, exact event history, provider-independent continuity	Living subject + Twin + deterministic world
Consumers	Identity and owned objects that remain theirs across apps and devices	Receiz ID + vault object + recovery
Developers	SDK, MCP, skills, verification, projection, and operations without rebuilding proof infrastructure	Coordinated developer surface

CHAPTER 32

The cash-out stack

Multiple revenue surfaces can share one proof-native substrate.

1. Developer infrastructure

SDK, MCP, AI skills, verification, hosted indexes, operational tooling

2. Artifact services

Seal, verify, certify, preserve, publish, inspect, and recover

3. Identity continuity

Recovery, device binding, portable identity records, institution-grade account continuity

4. Ownership and custody

Vault operations, transfer, marketplace infrastructure, provenance histories

5. Settlement rails

Certificates, notes, funded events, reserve, governed transfer and terminal state

6. Proof-native worlds

Living subjects, deterministic events, memory, mandates, economies, and licensed runtimes

Compounding architecture

Each commercial surface can increase the number of verified objects, identities, appends, transfers, and inspection events without requiring a new truth model. The substrate compounds if adoption compounds.

CHAPTER 33

The moat is constraint-level architecture

Features can be copied. A coherent truth hierarchy across every boundary is much harder to reproduce.

DEFENSIBILITY LAYER	WHY IT COMPOUNDS
Sealed artifact truth	Every verified object adds independently inspectable history that does not depend on continued platform authority.
Constitutional law and digest chain	Receivers can name the exact current law while preserving original historical law.
Cross-surface primitive consistency	Identity, proof, ownership, Settlement, public witness, offline verification, and AI agency share one ontology.
Carrier and device evidence	Real artifact transformations become executable compatibility law rather than undocumented edge cases.
Contract density	Release locks, conformance, visual evidence, and regression records make silent semantic drift expensive.
Developer coordination	SDK, MCP, AI skills, docs, app operations, and public archive move as one release boundary.
First-paint continuity	The user experience expresses the truth hierarchy directly, making the product both faster and harder to downgrade.

Why incumbency alone is insufficient

A larger database, brand, cloud, or distribution channel cannot substitute for independently recomputable objects if the system still requires its own server to make them true.

CHAPTER 34

Institutional readiness

The standard is not a beautiful demo. It is bounded authority, repeatable evidence, compatibility, and exit.

INSTITUTIONAL QUESTION	V121 ANSWER	OPEN EVIDENCE
Can the claim be recomputed?	Yes, where the governing proof object and verifier exist.	Independent deployment review may still be required.
Does history survive upgrades?	Historical objects verify under original law; no rewrite required.	Long-horizon archival operations are a separate program.
Can operations fail closed?	Admission, ownership conflicts, and primitive boundaries reject without promoting weaker truth.	Production fault evidence remains separate.
Can users exit?	Artifacts, identity records, offline verification, ownership, cancellation, claim, and settlement carry terminal states.	Jurisdiction-specific process is not inferred.
Can agents be bounded?	Typed operations, mandates, exact heads, deterministic admission, report-only receipts.	Operator policy must still be configured.
Is release evidence honest?	Repository qualification is separated from tag, publication, deployment, smoke, and attestation.	Those external actions remain pending/unknown.

CHAPTER 35

Commercial sequencing

The path from qualified technology to durable revenue has explicit gates.

PHASE	OBJECTIVE	PROOF OF COMPLETION
1. Release closure	Tag, sign, publish coordinated packages, deploy, and smoke	Immutable tag, registries, attestation, production evidence
2. Reference adoption	One end-to-end public proof, identity, ownership, or agent workflow	Verified objects used outside the authoring session
3. Paid infrastructure	Charge for distribution, indexing, operational verification, or hosted coordination	Contracted usage and collected revenue
4. Transaction layer	Enable certified transfer, custody, note, certificate, or Settlement routes	Lawful terminal-state and reconciliation evidence
5. Institutional programs	Package policy, deployment, audit, recovery, and sovereignty boundaries	Independent qualification and operator governance
6. Ecosystem compounding	Third-party applications create and verify interoperable objects	Multi-party object and operation growth

No skipped gate

The repository establishes product and engineering capability. Revenue, adoption, valuation, and institutional acceptance become facts only when separately evidenced. This sequencing protects the claim while making the route to cash-out concrete.

CHAPTER 36

What makes the public say 'really?'

Five statements that are both understandable and grounded in implemented primitives.

The file can carry its own proof.

The object can be inspected and verified without asking the app or database to make it true.

Your identity can survive the session.

Receiz ID and portable recovery proofs project the complete account across devices and restore paths.

An AI can act without being allowed to invent history.

The model proposes; typed deterministic operations admit or reject events.

Ownership can move while the object stays the same.

Custody and authority append; stable identity and complete history continue.

The fastest screen can also be the most truthful one.

Known verified truth paints immediately; networks append beneath it.

PART A

Appendix

Canonical sources, digests, methodology, and explicit claim boundaries.

A public release artifact is credible only when a reader can trace its statements back to governing records and distinguish implemented fact from future possibility.

APPENDIX A

Canonical v121 source map

Repository-local evidence identifiers used by this book.

REF	SOURCE	ROLE
R1	docs/releases/v121.0.0.md	Main narrative, authority boundary, compatibility, qualification
R2	docs/releases/v121.0.0-product-truth.md	Frozen primitives, invariants, result, gates, residual risks
R3	docs/releases/v121.0.0-regression-lessons.md	Failure anatomy, lessons, locked rules, contract gates
R4	docs/releases/v121.0.0-checklist.md	Completed and unfinished release actions
R5	docs/releases/v121.0.0-conformance.md	Required repository evidence commands
R6	docs/releases/v121.0.0-compatibility-matrix.md	Historical and current compatibility
R7	docs/releases/v121.0.0-migration.md	Adoption without proof rewrite
R8	docs/releases/v121.0.0-process.md	Release sequencing and practical rule
R9	docs/releases/v121.0.0-commit-history.md	Implementation lineage and unresolved boundary commit
R10	docs/releases/v121.0.0-constitution-registry.json	105-law immutable registry
R11	receiz-docs/content/releases/v121.0.0.json	Public archive status and claim bindings
R12	AGENTS.md and linked doctrine kernels	Literal product law and truth hierarchy

APPENDIX B

Release digests and identity

Exact current coordination anchors.

FIELD	VALUE
Release	121.0.0
Release date	2026-08-19
Registry digest	29a793a5bcc0195ab41d30614d37ac51df66023af354fa4335460764eb0af413
Previous registry digest	0728651789b26e1d10c1991ec1c06c1ea4a576f0c6520537b250b171f8857073
Effective Kai	13,725,000
Registry laws	105
SDK / MCP / AI skills	121.0.0 / 121.0.0 / 121.0.0
AI skill index	39 skills; 33 manifests; 30 OpenAI agent prompts
Tracked-file manifest SHA-256	2e998619c73b948a6ce260c0c0a88621512a57d253b6bae431ae2dd740c886c8

Digest meaning

The manifest digest authenticates this generated accounting file as a byte sequence. It does not replace the stronger proof, release, registry, or governance primitives described in the book.

APPENDIX C

Accounting methodology

How 'all files' is made exact and reproducible.

- Resolve the release tree with git ls-files.
- Read each regular file exactly as bytes; record symlink target bytes without following the link.
- Compute SHA-256 for each path.
- Count bytes for every file and text lines only when the sampled content is non-binary.
- Assign each path to one explicit accounting family using stable top-level and route/test rules.
- Aggregate by accounting family, top-level path, and extension.
- Write the exact file-level CSV companion and hash the completed CSV.
- Render aggregate accounting into the PDF without hiding the residual family.

Scope boundary

Tracked files are the repository release scope. Untracked local files, ignored dependency caches, external production state, package registries, and deployment platforms are not silently included or inferred.

Companion location

output/pdf/receiz-v121-full-file-accounting.csv and output/pdf/receiz-v121-full-file-accounting.sha256

APPENDIX D

Claim boundary

Implemented fact, qualified repository evidence, and future value remain distinct.

CLAIM CLASS	THIS BOOK'S RULE
Implemented primitive	Name directly when code and contracts satisfy the Receiz acceptance criteria.
Repository qualification	Claim only the gates recorded as passed on the release tree.
Publication / deployment	Claim only with package, tag, deployment, and smoke evidence.
Institutional adoption	Do not infer from technical capability or documentation quality.
Revenue / valuation	Do not infer from architecture, market relevance, or repository scale.
Inevitability thesis	Present as conditional adoption pressure and architecture fit, never as a guaranteed financial outcome.
External certification	Do not infer from internal conformance or governance controls.
Visual truth	Any contradictory screenshot, recording, DOM snapshot, or user reference fails the claimed surface state.

The highest-level claim

Receiz v121 is a repository-qualified proof-native artifact system whose sealed objects, identities, ownership states, Settlement states, public witnesses, and verified appends remain source while global representations become fast and durable beneath them.

APPENDIX E

Glossary of implemented primitives

Direct names for the architecture the release already carries.

TERM	MEANING IN RECEIZ
Proof object	Stable object identity plus recomputable payload, integrity/provenance material, inspection behavior, and truth stronger than UI state.
Vault object	Proof identity, visual identity, ownership/custody, media, provenance, actions, and a surface where the object remains itself.
Public proof surface	Public presentation of proof, provenance, ownership/custody, source material, verification, shareability, or witness state.
Identity primitive	Receiz ID, PBI, portable identity proof, recovery, binding, and account/device continuity.
Settlement primitive	Proof-native value, reserve, certificate, note, funded purchase, governed transfer, or terminal economic state.
Durable local truth	Verified stored truth used immediately; not temporary cache and not subject to rediscovery before use.
Composite head	A proof object carrying append-only public state and discoverability.
Kai	Structural proof coordinate and ordering state machine; Chronos remains human-readable bridge time.
Living subject	A proof-native being with immutable identity, complete history, memory, bounded agency, world state, and ownership continuity.

FINAL

The public release statement

A concise statement suitable for institutions, builders, creators, and the public.

Receiz v121.0.0 establishes source-first continuity across the full proof-native system. A sealed proof object remains authority when it is published, indexed, hydrated, restored, displayed, transferred, or reconciled. Stable identity survives session mechanics. Known Settlement and Reserve value survives representational delay. Public proof becomes globally discoverable without making the index the proof. AI can express and propose without being allowed to invent canonical events. Historical objects remain exact-byte verifiable under their original law.

The result is not a slower system made safer by more checks. It is a faster system made possible by a stronger truth boundary: paint what is already known, append only what is newly verified, and never ask a weaker representation to re-authorize the source.

Repository-qualified release

The v121 release tree and full release freeze are recorded. Immutable tag, package publication, production deployment, signed attestation, production smoke, adoption, and revenue remain separately evidenced actions.

The durable rule

First admission only. Then append forever.