



RECEIZ PUBLIC RELEASE BOOK

Release evidence, architecture, and public implications



V120.0.0

Living Proof Subjects

A proof-native foundation for living digital beings

One immutable identity.

One complete proof history.

Agency that can act - without letting AI invent reality.

RELEASE DATE

August 16, 2026

DEVELOPER SURFACE

@receiz/sdk@120.0.0

EVIDENCE STATUS

Repository candidate

Public edition. Publication, deployment, production smoke, and signed attestation are tracked separately.

CHAPTER 00

The release in one minute

A public guide to the breakthrough, the system beneath it, and the evidence that exists today.

REPOSITORY CANDIDATE - EVIDENCE-BOUND

Receiz v120.0.0 makes creatures, characters, vehicles, agents, collectibles, and other proof-native beings first-class living subjects. Each subject carries an immutable identity, complete proof history, portable mind, event-derived memory, relationships, inventory, bounded mandates, deterministic world participation, and append-only ownership continuity.

Evidence: R1 release narrative, R2 product truth

469

complete profile proof
objects

19/19

living-subject
conformance

37

typed subject MCP tools

39

coordinated AI skills

The breakthrough

AI can speak, perform, suggest, and reason as a subject. It still cannot decide what happened. Only an exact-head, typed, deterministic Receiz command or atomic transaction can admit a world event. Identity, memory, agency, social state, possessions, and ownership therefore remain continuous without making model output the authority.

The public evidence boundary

The candidate line records focused tests, conformance, package builds, lint, page-first-paint, and deterministic-surface checks. Package publication, deployment, production smoke, governance attestation, fresh final browser evidence, and a ready narration derivative remain separately unfinished as of August 16, 2026.

Why this matters

V120 replaces the fragile pattern of a character spread across a prompt, database snapshot, app account, and provider session with one proof object whose life can be verified, replayed, transferred, and continued.

Evidence: R1 Why it matters, R3 current evidence boundary, R4 checklist

CHAPTER 01

A map of this book

Start with the human story. Continue into architecture, evidence, developer adoption, and the exact release boundary.

CHAPTER	WHAT YOU WILL LEARN
02	The category shift Why a living proof subject is different from an AI character or database row.
03	The architecture Identity, proof brain, Twin, world runtime, memory, autonomy, and ownership.
04	The major accomplishments Thirteen shipped outcomes that close the full lifecycle.
05	Evidence and conformance What the candidate line proves, how it fails closed, and what remains unfinished.
06	Building on v120 Developer surfaces, compatibility law, migration steps, and use cases.
07	Public meaning What this makes possible, what it refuses to do, and why those limits matter.
A	Appendix Glossary, exact digests, status ledger, and canonical source map.

How to read the evidence labels

Each major claim points to a canonical release record using R1 through R9. These are repository-local source records, not marketing citations. The source map appears in the appendix.

Implemented fact

A capability stated directly by the release narrative, product-truth ledger, code contract, or conformance record.

Future possibility

A public implication or use case enabled by the architecture. It is presented as what v120 makes possible, not as deployed production evidence.

Public-language promise

This book names the implemented primitive directly: proof object, living subject, ownership transition, deterministic world event, durable proof memory, and public proof surface. It does not downgrade them into cards, app state, metadata, or model memory.

Evidence: R2 frozen primitives and invariants, R8 regression lessons

CHAPTER 02

The category shift

From simulated continuity to verifiable continuity.

Before v120

A proof-native creature could be represented inside an application, but the application still needed a second authority system for identity, memory, autonomy, social state, trade, battle, and transfer. A person-scoped Twin could simulate the creature through a thread key, but the creature did not yet carry its own portable identity, complete proof memory, bounded mandate, or atomically replayable world history.

With v120

The living subject is the proof object. Its identity is immutable. Its life is append-only. Its Twin reasons from exact proof history. Its model may propose actions, but the deterministic world runtime decides what happened. Its memories cite admitted events. Its relationships and inventory change atomically. Its owner can authorize bounded autonomy, revoke it, or transfer the same subject without rewriting identity or dropping history.

Evidence: R1 release narrative

QUESTION	CONVENTIONAL DIGITAL CHARACTER	RECEIZ V120 LIVING SUBJECT
What is identity?	An app record, account link, or prompt persona.	Immutable subject and proof-object identity.
What is memory?	A context window, summary, embedding, or latest snapshot.	Complete content-addressed proof history with bounded retrieval.
Who decides events?	Often the model, server, or application callback.	Typed command plus deterministic reducer and atomic admission.
What happens on transfer?	A new owner record or duplicated character state.	Authority changes; identity and complete history continue.
Can it move?	Only through provider or app export conventions.	Portable mind and artifact continuity across apps, devices, and providers.
How does failure behave?	Partial writes, retries, or snapshot repair may diverge.	Zero-write rejection, idempotent receipts, replay, and exact heads.

Evidence: R1 sections 1-9, R2 frozen invariants, R3 conformance

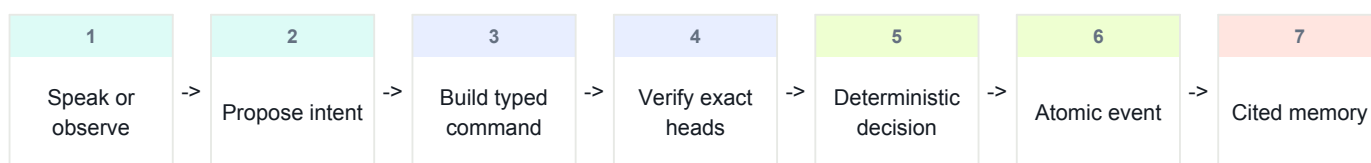
CHAPTER 02

The world-changing idea

A digital being can have continuity without giving probabilistic AI the power to rewrite reality.

The design move

Separate expressive intelligence from canonical authority. The Twin can speak and propose. The deterministic Receiz world admits or rejects the action against exact current heads and authority.



Expression stays rich

- Spoken response and streamed text.
- Emotion, gesture, gaze, blink, breath, audio, and visemes.
- Observed facts and proof-derived memories.
- Structured intentions with reasons, targets, confidence, and citations.
- Provider and model provenance for reproducibility.

History stays exact

- No model statement creates a world event.
- Rejected work advances no head and allocates no Kai coordinate.
- Multi-subject effects advance every head or none.
- Factual memory cites admitted events.
- Replay produces the same heads; retry cannot duplicate admission.

This separation is the foundation for believable digital life that remains inspectable. It permits personality, improvisation, and live performance while preserving deterministic provenance and consent-bounded action.

Evidence: R1 sections 3-6 and 10, R2 Twin and world separation, R3 living-subject laws

CHAPTER 03

The architecture of a living proof subject

One object carries the complete lifecycle instead of outsourcing truth to a prompt, row, or provider.

LAYER	PRIMITIVE	WHAT IT CARRIES
1	Immutable identity	subjectId, proofObjectId, subjectType, identityDigest, genesisDigest, createdAtKai
2	Canonical state	Self model, condition, position, abilities, event history, and append-only heads
3	Complete proof brain	Entire content-addressed history, Kai/Merkle/Fibonacci indexes, exact primary bytes
4	Twin and portable mind	Cited reasoning, structured intent, portable mind artifact, non-canonical performance
5	World runtime	Typed commands, exact-head validation, deterministic reducer, atomic events, receipts, replay
6	Social and material life	Event-derived memory, relationships, inventory, trades, battles, and gifts
7	Consent and custody	Bounded mandates, privacy scopes, ownership transitions, revocation, provenance
8	Application continuity	Byte-preserved namespaces, streams, forks, offline restore, cross-app projection

Strongest truth

The enclosing sealed proof object remains the strongest truth. A model response, index, projection, cache, database row, server response, MCP result, receipt, or latest snapshot cannot replace subject identity, exact history, admitted events, ownership continuity, or settled first paint.

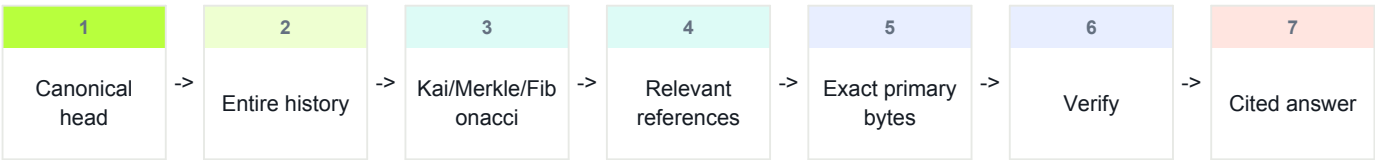
Evidence: R2 frozen primitives and invariants, R5 compatibility law

CHAPTER 03

The complete proof brain

A bounded reasoning window is not a bounded life.

The canonical head commits the entire content-addressed proof history. Kai ordering, Merkle membership, Fibonacci skip commitments, and segmented indexes locate relevant material. Retrieval then resolves and verifies exact primary proof-object bytes before reasoning.



Complete authority

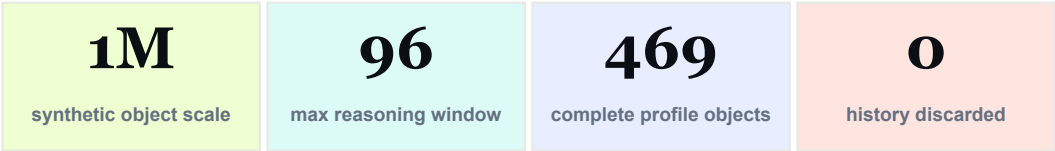
There is no fixed object-count ceiling on canonical history. Exact long-form posts remain primary bytes. Unknown namespaces remain byte-for-byte stable. Missing or mismatched primary bytes fail closed.

The million-object scale test keeps the causal head and retrieval bounded without materializing a million-object prompt.

Bounded work

The 96-object value is a maximum reasoning window - a projection over the brain, never the brain itself. Indexes, excerpts, embeddings, summaries, and model context are locators or derived projections only.

Streaming can emit head, candidates, exact objects, passages, and a final context receipt so useful work starts before the full candidate set finishes.

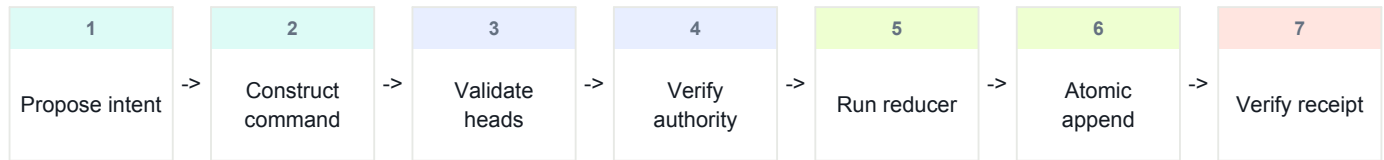


Evidence: R1 section 2, R2 complete proof brain, R3 scale laws, R4 checklist

CHAPTER 03

The deterministic world

Consequential actions follow one exact admission sequence.



CONTROL	WHAT IT PREVENTS	FAILURE RESULT
Exact subject and world heads	Acting on stale or replaced state	Zero writes
Owner capability or current mandate	Unauthorized or expired autonomy	Zero writes
Deterministic reducer	Provider-dependent reality	One reproducible decision
Atomic multi-subject transaction	One-sided relationship, battle, gift, or trade	All heads advance or none
Idempotency identity	Duplicate event on retry	Original receipt returns
Replay and checkpoint	Latest-snapshot-wins divergence	Same inputs reproduce same heads

Zero-write means zero-write

A rejected decision writes no subject head, world head, event, projection, or Kai coordinate. Failure cannot be hidden by inventing an event or partially repairing a later snapshot.

Evidence: R1 sections 4-5, R2 Twin and world separation, R3 conformance

CHAPTER 04

Thirteen major accomplishments

The complete v120 lifecycle, from immutable identity through public release surfaces.

01**Immutable living-subject identity**

A subject remains itself across application, device, provider, and owner changes. Identity and genesis never rewrite.

02**Complete streamed proof brain**

Canonical history is complete; only retrieval and reasoning windows are bounded. Exact primary bytes remain authoritative.

03**Subject Twin and portable mind**

The subject carries its own Twin and portable mind, independent of the owner's personal Twin or one model provider.

04**AI proposes; the world decides**

Rich model output remains non-authoritative. Typed, exact-head deterministic admission alone creates events.

05**Atomic multi-subject reality**

Relationships, battles, gifts, and trades advance every affected subject and the world together or write nothing.

06**Event-derived memory and possessions**

Factual memory cites events. Relationship and inventory changes are admitted facts, not model summaries.

Evidence: R1 sections 1-6, R2 product truth

CHAPTER 04

Thirteen major accomplishments

Durable autonomy, transfer, cross-application continuity, public first paint, and a coordinated developer platform.

07**Bounded autonomy and durable execution**

Mandates limit action, privacy, value, frequency, region, provider, app, tenant, expiry, and revocation at execution time.

08**Bearer ownership transition**

Known-recipient and open handoff preserve identity and history while revoking former-owner authority immediately.

09**Cross-application live continuity**

Replay cursors, causal parents, streams, duplicate tolerance, partitions, and forks converge without latest-snapshot replacement.

10**Voice, narration, and performance**

Audio, visemes, emotion, and gesture can stream without becoming canonical. Play appears only for exact-digest ready audio.

11**Showcase continuity and first paint**

A sealed composite proof object carries all 469 complete Public/Signed objects and reopens without rediscovering held history.

12**SDK, MCP, and AI Skills parity**

Exact v120 identity and digests coordinate the application, SDK, tools, skills, schemas, verifier, and docs.

13**Emulator, schemas, and conformance**

Draft 2020-12 schemas, deterministic emulation, durable snapshots, and a 19-law conformance suite make behavior testable.

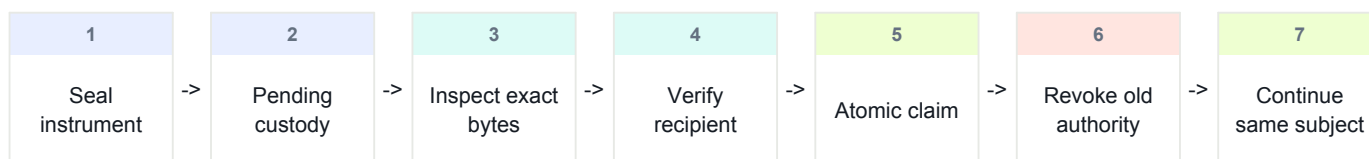
Evidence: R1 sections 7-13, R3 conformance, R4 checklist

CHAPTER 04

Ownership can move without identity loss

Transfer changes custody and authority. It does not clone the subject or erase its life.

A one-time bearer instrument binds the exact current subject and ownership heads, recipient mode, expiry, pending custody, inventory disposition, prior-owner conversation policy, registry, reducer, and claim capability. It can support a known recipient or an open handoff and can be inspected offline before online claim.



What continues

- Subject identity and genesis.
- Complete proof and event history.
- Public memory, relationships, and provenance.
- Transferred inventory and unknown namespaces.
- The ability to verify the same subject independently.

What stops immediately

- Former-owner mandates and activating capabilities.
- Former-owner private Twin access.
- Queued or leased autonomous jobs.
- Conflicting sale, trade, or transfer while pending.
- Autonomy until the new owner authorizes a new mandate.

Replay protection without breaking retry

The same claim attempt can return its original receipt. A second distinct claimant cannot reuse the instrument. Stale, cancelled, expired, or replayed claims change nothing.

Evidence: R1 section 8, R2 bearer ownership transition, R3 bearer-transfer laws

CHAPTER 04

Autonomy is a consent boundary

An owner can authorize meaningful action once - and the runtime rechecks the mandate when work actually executes.

MANDATE DIMENSION	BOUND AT EXECUTION
Action scope	Allowed actions, battle policy, item use, gifts, trades
People and place	Regions, subjects, owners, conversation and privacy policy
Economic exposure	Item value, daily value, approval thresholds, count limits
Time and rate	Expiry, frequency, minimum interval, bounded catch-up
Execution context	Provider allowlist, safety policy, application, tenant
Current authority	Owner, active status, subject head, world head, revocation

Durable scheduling uses leases, crash recovery, deterministic Kai clocks, stale-head rejection, budgets, and bounded offline catch-up. A pause, revocation, expiry, ownership change, provider mismatch, timeout, malformed intent, or scope breach stops admission. The runtime never invents a successful event to conceal a failed model or process.

The practical result

Autonomy becomes an inspectable, revocable, digest-bound capability rather than a broad permission granted to an app, owner record, or model session.

Evidence: R1 section 7, R2 autonomous mandates and runtime, R4 checklist

CHAPTER 04

Public proof and truthful media

The visible experience stays settled while background work remains subordinate.

Showcase continuity

The v120 profile release projection carries all 469 complete Public/Signed proof objects at one sealed composite head. The visible first paint is bounded for speed, but authority is complete. Reopening uses carried proof truth instead of rediscovering history.

The top profile total and Showcase total derive from the same admitted composite count. A successor publish is complete only after its new head settles.

Narration continuity

The enclosing proof object is verified before narration. The derivative binds the exact long-form text digest. A Play control appears only when matching audio is already ready.

An opened reader is settled. Background narration inventory can prepare a later interaction but cannot redefine the reader already on screen.

PUBLIC CONTRACT	V120 RESULT
Complete profile authority	469 complete proof objects at the exact composite head
Fast presentation	Bounded visible window over complete authority
PDF artifact identity	Exact .receized.pdf remains viewable and downloadable
Incidental Identity Seal prose	Cannot reclassify an eligible PDF into PBI-gated artwork
Narration readiness	No Play control without a ready derivative bound to exact text
Opened reader	Background work cannot add or change controls in place

Evidence: R1 sections 10-11, R2 profile and long-form continuity, R3 scale and narration laws

CHAPTER 04

One coordinated developer surface

The application, packages, tools, schemas, verifier, and documentation share exact v120 identity.

120.0.0 coordinated identity	37 living-subject MCP tools	7 new subject AI skills	30 OpenAI agent prompts
--	---------------------------------------	-----------------------------------	-----------------------------------

SURFACE	PUBLIC PURPOSE	AUTHORITY BOUNDARY
@receiz/sdk@120.0.0	Subject, brain, Twin, memory, social, inventory, mandate, runtime, world, replay, stream, and transfer APIs	Canonical typed primitives and exported digests
@receiz/mcp-server@120.0.0	37 living-subject tools plus nine enclosing-artifact tools	SDK-backed; exact confirmations and independent byte resolution; MCP is not authority
@receiz/ai-skills@120.0.0	39 skills, 33 manifests, 30 prompts, including seven new subject skills	Bound to exact registry and operation matrix; model output stays non-authoritative
Schemas	Draft 2020-12 living-subject bundle	Machine-readable identifiers coordinated with v120
@receiz/sdk/testing	Deterministic emulator and 19-check conformance	Multiple owners/subjects, partitions, retries, recovery, tamper and replay coverage
Receiz Docs	Narrative, migration, compatibility, conformance, evidence	Permanent public boundary; status dimensions remain independent

Evidence: R1 sections 12-13, R3 package parity, R5 coordinated identity

CHAPTER 05

What the candidate line proves

Evidence recorded on August 16, 2026 - separated from publication and deployment claims.

CHECK	RESULT	BOUNDARY
v120 release lock	PASS	Exact release contract
Coordinated release identity	7/7	Application, packages, verifier, docs, registry, matrix
SDK/MCP living-subject parity	15/15	Typed parity and authority boundary
Living-subject conformance	19/19	Zero network calls; zero database calls
Narration and primary-proof binding	79/79	Exact source and readiness behavior
Profile, proof brain, narration integration	33/33	Complete profile and exact long-form retrieval
Lint	PASS	Zero warnings
SDK and MCP TypeScript builds	PASS	Package builds
Page-first-paint	PASS	Direct deterministic World preview
Deterministic surfaces	PASS	Complete custom-domain profile binding

Conformance evidence digest

1c10df56fa86c031b8f3b4cf6a692c905b8b1662350d9a7ba817b162decaeb72

The 19/19 conformance run records zero network calls and zero database calls. It proves repository behavior on exact candidate bytes; it does not manufacture publication, deployment, production state, audio readiness, or governance approval.

Evidence: R4 candidate evidence, R3 evidence boundary

CHAPTER 05

The nineteen conformance laws

Ten core living-subject laws and nine bearer-transfer laws.

Core living-subject laws

1. No model statement becomes an event without typed command admission.
2. Every factual memory cites admitted event IDs.
3. Multi-subject effects are atomic.
4. Retry cannot duplicate an event.
5. A revoked mandate cannot execute.
6. Offline restoration preserves exact state.
7. Independent replay produces the same heads.
8. SDK and MCP reads produce equivalent projections.
9. Unknown namespaces survive every transition.
10. Conversation and animation failure cannot corrupt canonical state.

Evidence: R3 conformance

Bearer-transfer laws

1. A second distinct claim is rejected; idempotent retry returns the original receipt.
2. A stale ownership head is rejected.
3. Cancelled and expired instruments are rejected.
4. The former owner loses private Twin and world authority after claim.
5. A queued former-owner mandate cannot execute.
6. The new owner receives the exact same identity and complete history.
7. Unknown namespaces survive transfer byte-for-byte.
8. A failed transfer produces zero state changes.
9. Offline inspection followed by online claim converges to the same ownership head.

CHAPTER 05

Failure is part of the contract

V120 defines what must never happen - and makes those boundaries release-blocking.

DOMAIN	RELEASE-BLOCKING REGRESSION	LOCKED CORRECTION
History	Treating 96 objects as canonical history	Complete head; bounded retrieval only
Authority	Letting model, MCP, receipt, row, projection, or snapshot admit an event	Typed command and deterministic reducer
Convergence	Latest-snapshot-wins reconciliation	Replay cursors, causal parents, forks, exact heads
Memory	Factual memory without event citations	Event-derived, rebuildable projection
Atomicity	One-sided relationship, battle, gift, or trade	Every affected head advances or none
Failure	Any rejected action advancing Kai or state	Zero writes and structured failure
Autonomy	Execution after revocation, expiry, owner change, or scope breach	Reverify at execution
Transfer	Identity change, namespace loss, or former-owner authority	Same subject plus immediate revocation
Public proof	Stale visible count or fake narration readiness	Exact composite head and exact-digest audio

These are not edge-case preferences. They protect the central v120 claim: the subject's identity and admitted life remain stronger than every temporary model, transport, projection, or user-interface layer.

Evidence: R2 release-blocking failures, R8 regression lessons

CHAPTER 05

The unfinished release dimensions

Public truth includes what is not yet claimed.

DIMENSION	REQUIRED EVIDENCE	STATUS AS OF 2026-08-16
Full repository gate	Complete optimized application build and full test:release-freeze on a runner that permits local IPC sockets	Not closed
Final visual evidence	Fresh desktop and mobile screenshots and DOM evidence from the final frozen tree, including stable 24/469 first paint and 469 total	Not closed
Narration derivative	Generate or verify at least one ready audio artifact bound to the exact long-form digest before showing Play	No ready audio
Governance	Evidence/signature checks and a signed release attestation	Not closed
Distribution	Package publication	Not claimed
Deployment	Production deployment and production smoke	Not claimed
Release ledger	Final boundary commit, exact commit count, and tag after freeze	Not claimed

Why the distinction matters

Repository implementation and conformance cannot create package publication, production state, governance approval, or a media derivative. Receiz tracks each as independent evidence instead of allowing one successful test suite to stand in for the world outside the repository.

Evidence: R1 evidence boundary, R2 residual risks, R4 still required, R9 commit history

CHAPTER 06

Building on v120

The coordinated identity, compatibility law, and migration boundary developers must preserve.

SURFACE	EXACT IDENTITY
Application	120.0.0
@receiz/sdk	120.0.0
@receiz/mcp-server	120.0.0
@receiz/ai-skills	120.0.0
Constitutional verifier	120.0.0
Application contract	$\geq 120.0.0 < 121.0.0$
Living-subject reducer	receiz.living-subject.reducer.v120 digest exported by SDK
Machine schemas	@receiz/sdk/schemas/v120/living-subjects

Compatibility law

Historical sealed proof artifacts remain independently verifiable under their carried release law. Historical runtime objects do not authorize v120 transitions. V120 import preserves exact bytes and admits them under the current boundary; it does not silently rewrite history.

The complete proof history has no fixed object-count ceiling. Retrieval windows and streamed segments remain bounded projections over that complete history.

Evidence: R5 compatibility matrix

CHAPTER 06

Migration, part one

Adopt v120 as one coordinated authority boundary.

1**Coordinate exact v120 identity**

Upgrade SDK, MCP server, and AI Skills together. Bind the registry and operation-matrix digests.

2**Create real subjects**

Represent every creature, character, vehicle, agent, or collectible as `receiz.subject.v1` with immutable identity and byte-preserved namespaces.

3**Give the subject its own Twin**

Replace person-Twin thread multiplexing with subject Twin operations. A thread key can subdivide conversation, not identity.

4**Admit complete proof memory**

Carry every complete primary object at the exact current head. Do not import a prompt window, summary, embedding, or latest snapshot as canonical history.

5**Replace loose jobs with mandates**

Use an owner-confirmed mandate plus exact subject, world, and mandate heads for runtime ticks.

6**Route consequential action through the world**

Plan, validate, and execute typed commands. Model speech and intentions remain proposals.

7**Make shared reality atomic**

Route relationships, battles, gifts, and trades through exact-head world transactions.

Evidence: R6 migration steps 1-7

CHAPTER 06

Migration, part two

Rebuild memory, transfer, access, integration, convergence, and Showcase continuity around exact proof.

8

Derive factual memory

Project memory from admitted events and require citations. AI summaries remain rebuildable projections.

9

Use bearer ownership transfer

Define recipient mode, acceptance, expiry, inventory, prior-owner conversation policy, and pending custody.

10

Narrow application access

Replace broad ownership-derived access with subject, Twin, memory, mandate, world, runtime, relationship, and inventory scopes.

11

Upgrade agent integrations

Use the 37 v120 MCP tools and seven subject AI Skills. Confirm stored plan, permit, and instrument digests for writes.

12

Preserve causal convergence

Keep replay cursors, causal parents, duplicate tolerance, partitions, and forks. Remove latest-snapshot-wins paths.

13

Carry the complete Showcase

Seal exact admitted count and head, append one successor per object, and stop rediscovering history already held.

Rollback law

Rollback restores the last exact v120 runtime and rebuildable projections. It never rewrites or removes proof objects, events, receipts, ownership transitions, or admitted history.

Evidence: R6 migration steps 8-13 and rollback boundary

CHAPTER 06

What developers can build

Examples enabled by the architecture. These are possibilities, not claims of deployed production products.

POSSIBILITY	WHY V120 ENABLES IT
Persistent game companions	A creature can remember exact shared history, act under bounded mandates, survive app changes, and remain the same subject when transferred.
Portable AI characters	A character can carry identity, mind, proof memory, privacy rules, and provider provenance across devices and Twin providers.
Collectibles with lived history	Ownership can move while the same object's events, relationships, provenance, and public memory continue.
Autonomous world participants	Vehicles, agents, or creatures can perform bounded work while every consequential action remains deterministic and revocable.
Cross-application worlds	Different applications can append understood namespaces without deleting unknown application truth or replacing prior history.
Verifiable social simulations	Mutual relationships, battles, trades, and gifts can produce atomic receipts and independently replayable outcomes.
Offline-inspectable handoff	A transfer instrument can be inspected without network authority before an exact-head online claim.
Provenance-aware storytelling	A Twin can speak from complete proof history and cite the exact objects and events behind factual memory.

The common foundation

Every example inherits the same rule: expression can be flexible; admitted identity, history, agency, ownership, and consent remain exact.

Evidence: Derived from R1 sections 1-13; possibilities are interpretive, not deployment evidence

CHAPTER 07

Why this changes the design space

The lasting accomplishment is not a single feature. It is a new way to compose identity, intelligence, action, and ownership.

1. Continuity becomes an object property

A living subject no longer depends on one app, prompt, database snapshot, owner, device, or model provider to remain itself. The immutable identity and complete proof history travel together.

2. Intelligence can be expressive without being sovereign

The Twin can improvise, converse, perform, and propose. The world still admits only deterministic, authorized, exact-head events. This turns AI from an unbounded narrator into a participant inside verifiable rules.

3. Ownership can preserve personhood-like continuity

Transfer changes custody and authority while preserving the same subject identity and life. Former-owner power stops; the new owner must deliberately reauthorize autonomy.

4. Memory becomes provenance, not recollection theater

Factual memory cites admitted events and exact proof objects. Summaries and embeddings help retrieval but cannot replace the primary history.

5. Failure becomes legible

Rejected actions perform zero writes. Retries are idempotent. Partitions preserve causal structure. Replay reconstructs exact heads. The system refuses to disguise uncertainty as a successful event.

Evidence: Synthesis of R1, R2, R3, and R8

CHAPTER 07

What v120 refuses to do

The boundaries are part of the invention.

WEAKER LAYER	IT IS NOT ALLOWED TO...
Model output	Cannot create a world event
Prompt window	Cannot stand in for complete history
Index or embedding	Cannot replace exact primary proof bytes
Database or server response	Cannot outrank sealed artifact truth
MCP result	Cannot create authority
Ownership record	Cannot rewrite subject identity or prior history
Failed decision	Cannot advance state, event, projection, or Kai
Background work	Cannot redefine settled first paint or an opened reader
Old runtime object	Cannot authorize a current v120 transition
Successful repository tests	Cannot imply package publication, deployment, production smoke, audio readiness, or attestation

A trustworthy living subject is defined as much by the authority it refuses as by the capabilities it exposes.

Evidence: R2 frozen invariants and release-blocking failures, R3 evidence boundary, R5 compatibility

CHAPTER 07

The release promise

V120 does not ask the public to trust a character. It makes the character's continuity inspectable.

A living proof subject can speak from its complete verified history, act inside deterministic rules, remember admitted events, form mutual relationships, hold inventory, operate under bounded consent, move between applications and providers, and change owners without ceasing to be the same subject.

That is the major accomplishment of Receiz v120.0.0: one proof-native authority system for the complete lifecycle of digital beings - identity, mind, memory, agency, social state, possessions, world action, transfer, replay, provenance, and public continuity.

Evidence-bound conclusion

The repository candidate implements and tests this architecture. Public package availability, deployment, production smoke, signed attestation, final browser evidence, and ready narration media remain independent release dimensions until separately completed and recorded.

Evidence: R1 Why it matters and evidence boundary, R4 release checklist

1

immutable subject

ALL

complete proof history

O

model-authorized events

FOREVER

append-only continuity

CHAPTER A

Glossary for public readers

The vocabulary behind the release.

TERM	PLAIN-LANGUAGE MEANING
Living proof subject	A proof-native being whose immutable identity carries append-only state, complete proof history, mind, memory, relationships, inventory, mandates, and world participation.
Proof object	A deterministically identified, inspectable object with verifiable or sealed payload and a source of truth stronger than display state.
Twin	The subject's reasoning and conversational surface. It can speak, cite, and propose, but it cannot admit world events.
Canonical head	The exact commitment identifying the accepted state and complete content-addressed history at a point in Kai order.
Primary proof-object bytes	The exact underlying content resolved and verified before reasoning or narration.
Projection	A derived view, summary, index, memory, or UI representation rebuilt from stronger accepted truth.
Mandate	A digest-bound, owner-authorized scope that limits autonomous action and is reverified at execution.
Atomic transaction	A multi-subject action in which every affected head and the world advance together or none advance.
Bearer transfer	A one-time exact-head ownership transition supporting known-recipient or open handoff while preserving the same subject.
Kai	Receiz structured ordering used to place admitted state and events deterministically.
Unknown namespace	Application-carried bytes that a current reader may not understand but must preserve exactly for future applications.
Settled first paint	The deterministic visible surface projected immediately from known stronger truth and protected from background redefinition.

Evidence: R1, R2, and Receiz primitive law

CHAPTER A

Exact identity and digests

The machine-level release coordinates named by the canonical records.

COORDINATE	EXACT VALUE
Release	120.0.0
Release date	2026-08-16
Developer surface	@receiz/sdk@120.0.0
Application contract	>=120.0.0 <121.0.0
V120 registry digest	0728651789b26e1d10c1991ec1c06c1ea4a576f0c6520537b250b171f8857073
Operation matrix digest	1c779ee5ade4b877ae9c6922ab02ba96ffffeb7580f1cf105a59fbb4424f351
Immutable v119 predecessor digest	49c167a437ec7c0e486412dd62c54af4abdf94eda1ebc18d263a027d105cecd9
Conformance evidence digest	1c10df56fa86c031b8f3b4cf6a692c905b8b1662350d9a7ba817b162decaeb72
Schema bundle	@receiz/sdk/schemas/v120/living-subjects
Reducer identity	receiz.living-subject.reducer.v120 digest exported by SDK
Candidate baseline	62b62ecf1 on kojib/finish-longform-narration

V120 appends from the immutable v119 registry digest. It does not reinterpret or rewrite a v119 artifact. The candidate commit ledger records two candidate commits from the baseline but does not claim a final boundary commit, release tag, publication, deployment, production smoke, or signed attestation.

Evidence: R2 product truth, R5 compatibility matrix, R9 commit history

CHAPTER A

Canonical source map

The repository records used to build this public release book.

REF	CANONICAL RECORD	ROLE
R1	docs/releases/v120.0.0.md	Complete public release narrative and thirteen major outcome sections
R2	docs/releases/v120.0.0-product-truth.md	Frozen primitives, invariants, proof brain, world separation, transfer, gates, and residual risks
R3	docs/releases/v120.0.0-conformance.md	Ten living-subject laws, nine bearer-transfer laws, scale, package parity, and evidence boundary
R4	docs/releases/v120.0.0-checklist.md	Exact pass evidence and unfinished release dimensions recorded 2026-08-16
R5	docs/releases/v120.0.0-compatibility-matrix.md	Coordinated identity and historical compatibility law
R6	docs/releases/v120.0.0-migration.md	Thirteen adoption steps, verification, and rollback law
R7	docs/releases/v120.0.0-process.md	Freeze, test, evidence, attest, publish, deploy, and separate production evidence
R8	docs/releases/v120.0.0-regression-lessons.md	Regressions, causes, locked rules, and release gates
R9	docs/releases/v120.0.0-commit-history.md	Candidate baseline and exact non-claims

Release-book integrity

This document summarizes repository-local product truth for public consumption. It does not replace the machine-readable registry, schemas, signed artifacts, test outputs, or final release attestation.

RECEIZ V120.0.0

Living Proof Subjects

One immutable identity. Complete proof memory. Deterministic agency. Append-only life.