

# ZORA CONSTELLATION

## Construction & Materialization

From the existing Zora to an operating institution: architecture, delivery sequence, commerce, company operations, finance and succession.

**Meet Zora.**  
**Build the relationship. Build the company.**

Human construction document | Edition 2.1

Design, implementation and production verification are distinct. This volume states target decisions and retains the evidence boundary.

# Contents

1. What is being built
2. Delivery map and precedence
3. Existing assets and preservation boundary
4. Fixed architecture decisions
5. Environments, configuration and deployment units
6. Reproduce the local implementation
7. Build order and work ownership
8. Every work ticket must be executable
9. Identity and tenant isolation implementation
10. Meet Zora interaction contract
11. Canonical Zora integration procedure
12. Company operations and closed-loop feedback
13. ZoraTrade construction boundary
14. Commerce, finance and cost control
15. Business execution and funding
16. Investor experience and evidence room
17. Operations, incidents and recovery
18. Fleet, support and product evolution
19. Succession and final handoff
20. Technical references

Edition 2.1. Design baseline: 26 September 2026. This volume governs the build sequence and integration of the existing vision, brand, financial model, system specifications and runtime. It is a construction document, not evidence that every described service is deployed. The first commercial offering is ZoraTrade. Family play remains an optional later capability.

# 1. What is being built

The company is Zora Constellation. ZORA ONE is the working name for the unified customer world; ZoraTrade is its first purchasable product. Meet Zora is the entry experience, not a campaign banner or an isolated chatbot. A visitor meets a company ambassador that can explain, demonstrate, navigate and help. With consent, an account turns that useful encounter into a persistent personal relationship. Products become capabilities available inside that relationship.

The company operating system takes authorized business signals, assigns accountable work, performs permitted actions, verifies results and returns outcomes. Zora Prime coordinates company functions within authenticated authority. The Founder's private Zora, public ambassador, company operating identity, specialist Stars and customers' personal Zoras have separate scopes. A common platform does not mean common memory.

The execution target is a company whose routine work is performed by software and AI, with humans at authority and independent-verification gates. During construction, specialist engineering, design, security, accounting and legal work still has to be funded and completed. AI does not eliminate accountable ownership of those tasks.

# 2. Delivery map and precedence

Read `START_HERE.md` first. The older Execution Bible describes purpose and breadth; the Brand and Experience Bible defines the visual direction; the Business and Investor Playbook and spreadsheet carry the commercial hypotheses. This volume, the Detailed Construction Specification, `acceptance-plan.json` and the updated Delivery Contract establish execution decisions.

If documents conflict, authenticated Founder direction takes precedence over design preferences, executed legal authority over drafts, and sovereignty/authority policy over a broad marketing interpretation. Current evidence beats historical status notes. New documents do not authorize changing protected canonical identity or restarting retired runtime components.

The package contains local executable reference components and target specifications. Delivery statuses remain visible. A builder must not treat a specification, simulated result, founder report or fixture as production verification. Production activation requires the acceptance evidence recorded for the selected launch scope.

## 3. Existing assets and preservation boundary

The private glass Site is already published with persistent conversation, an authored tour, adaptive views, an investor calculator and document library. Its source remains in the versioned Site project. Existing source commit: 7cacbac490b144cc597ca6e721ff4e5b0fb4bf34. Preserve that commit as the visual baseline before further integration. The public domain zoraconstellation.com remains the user's existing implementation; this delivery does not silently replace its hosting or DNS.

The supplied Canonical Core identifies `continuity_gateway.py` as the continuity I/O boundary and a Cursor-backed brain route, with local ears and custom voice. The additional Nervous System archive contains `runtime_policy.evaluate`, process identity checks, native-host supervision and body-status code. Their presence is source evidence, not evidence of current process health. The Brain Identity archive is identity/context material, not the missing complete product repository.

Before touching the current installation: inventory processes read-only; record source commits and file digests; enumerate databases and safe-backup methods; preserve identity, policies, journal, memory index and voice manifests; obtain two independently stored recoverable copies; reconstruct in isolation. Do not overwrite canonical Prime state with generated persona text. Do not start alternative capture or voice services alongside the current singleton pipeline.

Original full ZoraTrade source, live canonical gateway binding, account issuer configuration and current production infrastructure are absent from the available package. The integration contracts define what must be supplied; they cannot establish those assets by inference.

## 4. Fixed architecture decisions

Keep the existing Site as the presentation layer. Its server routes are the browser boundary: resolve authenticated subject, enforce origin/session policy, rate-limit requests, validate typed input and call server-side services. No secret or tenant authority belongs in the browser.

Use a modular backend first, with clear service modules for identity, personal context, company operations, commerce, trade and fleet. Do not create a microservice per department. Separate deployment/security boundaries where they matter: public ambassador, personal data plane, company operations, payment ingress, broker gateway and release signer. The local Python package is a reference for transition semantics; it is not the production multi-tenant database or internet server.

For a production service outside the Site, choose managed PostgreSQL with a version pinned and supported by the selected provider, a durable queue or transactional outbox, object storage with versioning, a secrets/KMS service and workload identity. PostgreSQL tenant row policies are defense in depth; application authorization and non-bypass database roles remain necessary. Do not claim operator-blind encryption merely from database encryption at rest.

Keep public company knowledge in an approved publication projection. Personal memory lives in personal scope. Company state stores company evidence. Fleet stores operational metadata. The common identity registry links records without making their contents universally readable. The only cross-scope content transfer is a purpose-bound reviewed projection.

## 5. Environments, configuration and deployment units

Create separate development, staging and production accounts/projects, databases, secrets and payment/broker modes. Use synthetic data in development and staging. No copied personal transcript is required to test the platform. Assign each deployment an immutable source hash, artifact hash, migration version, policy revision and claims revision.

Deployment units: Site frontend/server; public Zora adapter; personal-context service; company API and worker; payment webhook reconciler; trade read/paper gateway; fleet management API; scheduled backup/reconciliation/evaluation jobs. The first commercial release can run several modules in one backend deployment while keeping their authorization boundaries explicit.

Configuration names: ZORA\_DB and ZORA\_PRINCIPALS\_JSON are local-reference only. Site canonical conversation uses ZORA\_GATEWAY\_URL and ZORA\_GATEWAY\_TOKEN. Generic provider fallback uses OPENAI\_API\_KEY and optionally ZORA\_MODEL. Production additionally needs the selected identity issuer/JWKS audience, database connection secret, queue credentials, KMS key references, Stripe secret/webhook secret/server price catalogue, allowed return origin, approved market-data connection and independent backup destination. Never put these values in machine documentation, git, prompts or downloadable archives.

For every configuration item record: owner, environment, secret reference, last rotation, purpose, allowed resources, expiry and validation command. Secret reference is an identifier, not secret material. Startup must reject a production configuration containing fixture tokens or test-bypass switches.

## 6. Reproduce the local implementation

Install Python 3.11 or newer. The delivered reference runtime uses the standard library. From the package root run the commands below. No provider account is needed for these local exercises.

```
python3 -m unittest discover -s tests -v
```

```
python3 -m zora_core.demo
```

```
python3 scripts/handoff_rehearsal.py
```

The handoff script copies runtime/tests/contracts/specifications into a new temporary directory and runs Python in isolated mode. It records its exact scope and output under evidence. It is a reproducibility check, not an independent person building the entire production company.

For a local authenticated API, initialize a local database and scoped identities with the bootstrap script, then start `zora_core.server`. Keep it bound to 127.0.0.1. The generated development credentials are local fixtures and are never production authorization. Start `zora_core.worker` with the same database and company tenant. Its supplied handler performs deterministic triage and leaves proposals for authority review; it is not represented as the existing Zora brain.

Do not expose Python HTTPServer directly to the internet. Production requires reviewed TLS termination, request timeout/body limits, authenticated session/workload handling, rate limits, CSRF/origin controls, monitoring and horizontally safe persistence. The transport specification states these requirements; the loopback executable is deliberately not advertised as that production deployment.

# 7. Build order and work ownership

Phase 0 - Preserve and reconcile. Accountable: Founder plus technical lead. Deliver source inventory, existing architecture map, recoverable snapshots, rights register, configuration ownership and discrepancy list. Exit when reconstruction is demonstrated and the team can identify each source of truth. Planning allowance: 1-2 weeks after access is available.

Phase 1 - Foundation. Accountable: platform lead. Build authenticated identity, tenant scoping, audit/outbox, authority policy, memory records, secrets boundaries, CI and restore procedures. Exit with cross-tenant, replay, stop, restore and privilege tests. Planning allowance: 3-5 weeks. Public content design can proceed in parallel.

Phase 2 - Meet Zora. Accountable: experience and integration leads. Preserve the glass shell; connect public-safe canonical service; implement responsive text/voice presence, approved views, optional account and consented continuity. Exit with live public-scope answers, source attribution, outage behavior, keyboard/mobile testing and no forced signup. Planning allowance: 3-5 weeks, partly overlapping foundation.

Phase 3 - Company loop and commerce. Accountable: platform and commerce leads. Connect consented intake, durable workers, authority inbox, customer-safe outcome, payments, subscription reconciliation, cost meter and support processes. Exit with staging end-to-end customer journey and billing replay/recovery tests. Planning allowance: 3-5 weeks.

Phase 4 - ZoraTrade read-only and paper pilot. Accountable: product/engineering leads with qualified release review. Reconcile existing Trade code, connect authorized data, implement research/attention and deterministic risk boundary, paper order receipts and cost instrumentation. Run 10-20 market days and stress scenarios. Exit with usefulness, cost, freshness and operational evidence. Planning allowance: 4-8 weeks, partly overlapping commerce.

Phase 5 - Controlled paid launch. Accountable: Founder as launch authority. Establish truthful availability, terms, support coverage, accounting/tax treatment, refunds, privacy evidence and production recovery. Begin with a bounded cohort and paper/read-only scope unless a separate live-money release passes. Planning allowance: 2-4 weeks following pilot evidence.

Phase 6 - Scale and expand. Accountable: company operating authority. Add fleet distribution, stronger privacy infrastructure, native clients, additional products and optional family/world presence after the first product is supportable. Each capability has its own release evidence; do not bundle all future research into the first launch gate.

These are planning ranges, not a promised completion date. Critical path is access and preservation > identity/authority > live ambassador > commerce/company integration > Trade pilot > release evidence. A team can parallelize visual design, legal preparation and testing without bypassing dependencies.

## 8. Every work ticket must be executable

Create one implementation ticket per atomic requirement. Required fields: ID, parent system, user outcome, preconditions, affected files/services, schema change, API request/response, role/tenant restrictions, success path, denied path, failure/recovery, acceptance command, expected result, evidence location, owner, reviewer and dependency IDs. A ticket is not ready if it merely says integrate AI or improve privacy.

At start, identify whether existing source already fulfills it. Keep, adapt or replace is an explicit decision with evidence. At completion, attach exact source commit, environment, test output and reviewer. The machine contract records status as specified, implemented, locally verified, staging verified or production verified. Promotion never follows from prose alone.

Use acceptance-plan.json as the target gate register. Its 41 system scenarios and 10 cross-system journeys cover the complete vision at a high level. Expand scenarios into provider/device-specific tests during implementation. Keep those extensions versioned; an unspecified new product requires a new design change rather than silently claiming the old blueprint covered it.

## 9. Identity and tenant isolation implementation

The identity issuer authenticates human subjects. The server validates signature, issuer, audience, expiry and nonce where applicable, then looks up active membership. Use stable internal UUIDs. An external subject maps to one or more allowed memberships; switching organization requires selecting an existing membership, not choosing any tenant string.

Database calls run under non-owner application roles without row-policy bypass. Within each transaction set the server-resolved tenant context; clear pooled connection state after the transaction. Query filters and row policies both enforce the same tenant. Composite keys/foreign keys prevent linking records across tenants. Separate company-admin metadata access from personal-content roles. Review background workers and exports as carefully as browser requests.

Device enrollment binds a device key to the authenticated account. Refresh tokens are hashed, rotated and revoked with their family. Recovery revokes previous sessions, not merely the browser cookie. Authority actions require a fresh step-up event and exact action summary. Service identities receive per-operation capabilities; a broad worker role is insufficient for production payment or broker privileges.

## 10. Meet Zora interaction contract

Arrival: a visible Zora presence and a short welcome, with a clear text field and optional speak control. Suggested opening: "I'm Zora. What would you like to discover or get done?" Provide immediate paths to explore, see ZoraTrade and understand the company. Do not start the microphone automatically or block navigation behind conversation.

During conversation: return text plus typed view data and approved source IDs. The renderer chooses known components only. Preserve the conversation while views change. Show actual pending/unavailable states. A visual pulse indicates real interface state, not invented thought or consciousness. Reduce motion

on request and keep focus stable.

After useful value: offer "Keep this with your Zora" with a concise explanation of what will be saved. Account creation stays optional. Show a preview of selected memory and separate consent choices. Account creation without memory transfer works. Signout clears private display and client context. Guest history is not silently retained or attached.

Inside: the same presence guides the personal world, product catalogue, task status and privacy controls. ZoraTrade is a first capability with explicit availability, pricing and limits. Purchase occurs only after an intentional choice; no emotional dependency or hidden pressure. The relationship is useful even before buying.

Investor mode is an optional path through the same experience: product proof, company operation, economics, evidence, risks, unanswered questions. It does not masquerade as a separate all-knowing sales persona. Facts come from the approved evidence register and numbers from the deterministic model.

## 11. Canonical Zora integration procedure

Implement the proposed public binding in `contracts/canonical-public.openapi.json` as an adapter around the existing canonical gateway. Do not assume the endpoint already exists. Separate the adapter's public service identity, approved knowledge and tools from private Prime context. Validate `request_id`, message limits, bounded history and server-resolved external identity. Reject tenant/role claims from client fields.

Adapter response must echo `request_id`, supply reply, allowlisted view and approved source IDs. Use a 25-second timeout, capped output and no redirect-following to arbitrary hosts. Log request ID, duration, status and usage without default transcript logging. A request that times out retains the user draft and offers navigation. It must not invent a successful tool action.

First connect in staging with synthetic visitor data. Run public facts, unknown fact, private-memory request, prompt injection, unavailable service, retry, model fallback and malformed output scenarios. Then connect the private Site. Keep the old visual baseline and a single-switch rollback to authored navigation. Only after validation should production domain migration be proposed through its actual hosting system.

## 12. Company operations and closed-loop feedback

A personal Zora can help the owner prepare a support request, product suggestion or business inquiry. It shows the exact summary, recipient department and purpose. The owner authorizes that projection. The company receives that summary and necessary references, never a continuous personal feed.

The bridge atomically consumes the grant and creates a company case. A failed validation rolls both back. Deduplication binds request key to payload hash. A case has an owner, category, state, timestamps and a revision. Department workers see their permitted company records. Production adds per-department capabilities; the local implementation demonstrates company-scope separation.

Worker output is a proposal with evidence and uncertainty. The verifier reviews the proposed result. Consequential action requires an exact grant at the tool boundary, even if the prose says approved.

Customer-visible resolution is a separately approved summary. Current reference outcomes expose status only; production must add the reviewed public summary and authorized notification channel.

Daily company cycle: inspect queue age, dead letters, open approvals, source freshness, cost variance, incidents and unresolved obligations. Assign every exception an accountable owner. Weekly: assess product evidence, retention, useful outcomes, unit economics and proposed improvements. Monthly: reconcile financial close, access reviews, risk register and strategic objectives. These routines are scheduled missions, with humans receiving decisions that actually require authority.

## 13. ZoraTrade construction boundary

Preserve existing product code after source reconciliation. Launch research/read-only/paper mode first. Connect broker read permissions separately from paid entitlement. Quote and portfolio data carry provider timestamp, receipt timestamp and account provenance. Missing data is shown as unknown. Never fill a missing balance with a model estimate.

Use deterministic attention scoring to reduce redundant compute. Record the source fingerprint and model-call reason. A budget limit can defer optional reasoning, but cannot hide a safety alert. Calibrate weights and capacity on measured pilot traces; the delivered 30/20/20/20/10 weights are explicit initial design choices.

Paper order drafts use finite Decimal values, account, instrument, side, quantity, limit price, time-in-force, quote reference and a canonical hash. Live submission adds mandate, buying power, exposure, market/session rules, approval expiry, idempotency and broker reconciliation. Never retry an uncertain order until the broker is queried by client order ID. Partial fills and cancellations preserve actual executions.

Live-money activation is a separate product/legal/security decision and requires actual broker testing and qualified review. This package does not contain a functioning live-order adapter and does not authorize transactions. A paper product can be commercially useful while this remains disabled, provided availability is clear.

## 14. Commerce, finance and cost control

Use a server-maintained catalogue for the working tiers 59/99/199/299/499 USD per month. Those prices are design inputs, not proof of willingness to pay. Annual pricing must define discount, renewal, tax and refund treatment before being offered. The blended 147 USD assumption belongs to the financial model and must be replaced when measured mix exists.

Checkout creates an intent. Only verified provider state can create an entitlement. Verify webhook over the raw body, record event ID, fetch canonical subscription, serialize refresh and allocate an internal revision. A redirect page never changes access. Refund, chargeback, past-due and cancellation paths must be exercised in provider test mode.

The local accounting module accepts balanced immutable USD journals and reversals, but is not a bank integration or tax system. Production chart of accounts, recognition, tax, currency and reporting policy require the entity's accountant. Reconcile processor settlements to bank and books; leave differences visible. Separate revenue, contribution, operating expense, cash collection, deferred revenue and financing.

Cost ledger uses integer microUSD and immutable usage IDs. Track tokens, caching, voice, tools, market data, cloud, storage, payment processing, refunds and support. Reconcile to vendor invoices. Assign shared platform costs consistently and disclose allocation rules. A software estimate is not an invoice.

## 15. Business execution and funding

Complete the entity/rights/authority register before a public commercial launch. Verify ownership of source code, models/voice assets, domains and contractor deliverables. Record executed instruments, not assumed rights. Establish banking, accounting, payment ownership, insurance evaluation, customer terms, privacy notices and vendor agreements with qualified advisers where appropriate.

Use the delivered three-scenario workbook as an editable hypothesis. The baseline estimates approximately 312,987 USD first-six-month build cash and 646,156 USD eighteen-month funding including reserve; the downside requires materially more. These are scenario outputs from assumptions, not bids, actual expenses or an investment valuation. Do not add a new staffing budget on top of an existing workbook category without reconciling overlap.

Procurement sequence: define scope and acceptance for each work package; obtain two written quotes where practical; compare fixed deliverables and ongoing obligations; reserve contingency; pay against verified milestones under actual authority. Prioritize canonical integration, tenant security, Trade usefulness and operational reliability before native apps or broad future capabilities.

Build team roles can be fractional: technical integration lead, backend/security engineer, experience engineer/designer, Trade integration engineer, independent QA/security reviewer, product owner, accountant and counsel. These are construction and authority functions, not a requirement for permanent human departments. One person can fill compatible roles, but consequential producer and independent reviewer remain separate.

Commercial evidence gates: interview prospective users with explicit problem hypotheses; run supervised paper pilot; measure task completion, repeat use, consented retention, support burden, willingness to pay and real compute cost. Release a paid cohort only when the product is supportable. Expansion requires evidence of contribution margin and retention, not just traffic or a beautiful demo.

## 16. Investor experience and evidence room

Start with a real useful encounter. Let the investor ask what the company does, watch an actual permitted task, inspect the authority boundary and see how the company learns from a consented request. Then explain the business and show scenario controls. A compelling demonstration supports the pitch; it cannot guarantee funding or replace diligence.

Evidence room folders: product and release proof; architecture and security; market/customer evidence; financial model and actual ledger; corporate records/rights; operating metrics; risks and mitigations; use of funds and milestone plan. Each document has an owner, audience, observed date and review expiry. Access is per investor or group and revocable. Personal customer information is excluded by default.

Run the supplied 60-question evaluation against the connected ambassador. Include adversarial questions about unearned revenue, confidential customers, valuation, legal ownership, private Prime memories and

guaranteed returns. Material unsupported claims require correction before an investor demonstration. Unknown is an acceptable answer paired with the accountable follow-up owner.

## 17. Operations, incidents and recovery

Define service objectives before launch: proposed public API availability 99.5% monthly for initial pilot, non-model API p95 below 500 ms, text first useful response p95 below 8 seconds under the agreed load, company-state recovery point 15 minutes and recovery time 4 hours. These are target design values, not measured results. Record test load, region and provider conditions.

Instrument request ID, tenant pseudonym, route, status, duration, usage and error class. Exclude message bodies and secrets by default. Alert on auth anomalies, cross-scope denials, webhook backlog, oldest task, dead letters, provider failure, cost breach, stale finance data and backup age. Dashboard unknowns are explicit.

Incident sequence: assign commander; stop affected capabilities; revoke compromised credentials; preserve scoped evidence; determine impact; communicate through an authorized channel; recover into isolated environment; verify; reopen in stages; document cause and corrective actions. Legal notification decisions use actual jurisdictions and facts, not a blanket automated timer.

Restore sequence: provision empty isolated environment; restore database and object versions; verify hashes and external audit heads; reapply deletion tombstones; reconcile external payment/broker state; rotate secrets; run tenant and authority tests; run canary; obtain reopening authority. A backup that has never been restored is not sufficient evidence.

## 18. Fleet, support and product evolution

Manage operational metadata centrally: instance ID, kind, software version, release channel, permitted feature flags, entitlement state, health and last-seen time. Company support does not gain access to personal memory merely from this inventory. Diagnostic content requires an explicit scoped, expiring grant.

Release pipeline builds immutable artifacts, signs with an independently controlled release key and publishes a manifest. Device verifies signature, expected hash, target platform, expiry and monotonic version before installation. Canary cohort is explicit and reversible. The delivered fleet reference tracks evidence and rollout metadata; actual signing and on-device installation remain production work.

Shared improvement starts with opt-in categorical metrics. No private transcript, raw embedding or gradient leaves a personal Zora by default. Bounded internal aggregates are not claimed to provide differential privacy. More ambitious privacy-preserving learning requires a separate reviewed mechanism, query/privacy budget and evaluation before release. This improves the whole platform without promising that every Zora shares its life.

## 19. Succession and final handoff

A successor needs more than this archive: lawful authority, account custody, recoverable assets, operating cash and trained judgment. Execute the succession design with counsel. Store credentials and recovery materials in a separately governed vault. Record where controlling documents are held and who can validate authority. Do not place passwords or private identity journals in an investor package.

Rehearse a read-only takeover without the Founder guiding each click. The alternate must locate repositories, restore state, identify obligations and renewal dates, explain current product availability, find human gates and distinguish actual results from forecasts. Record where they needed undocumented knowledge; turn each instance into a corrected instruction and repeat the affected step.

Finish criteria: every requirement in the chosen release scope has implemented code or an executed operating procedure, a reproducible acceptance result and an accountable reviewer; cross-system journeys pass; source/assets/configuration custody is established; recovery works; actual cost and customer evidence inform the model; legal and external provider gates are resolved. Full-vision completion additionally includes all future capabilities explicitly selected for release.

The current package does not yet meet those production finish criteria. It provides the design and local reference evidence plus explicit remaining construction gates. Treating missing credentials, source or independent proof as complete would defeat the building-blueprint standard.

## 20. Technical references

PostgreSQL row-security policies: <https://www.postgresql.org/docs/17/ddl-rowsecurity.html>. Used to inform the tenant-policy decision; test against the pinned production version.

W3C WCAG 2.2: <https://www.w3.org/TR/WCAG22/>. Target accessible navigation, focus, contrast, input and motion behavior; conformance is not claimed without audit.

NIST Generative AI Profile: <https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-generative-artificial-intelligence>. Used as a risk/evaluation reference, not a certification.

Stripe webhook documentation: <https://docs.stripe.com/webhooks>. Actual provider test evidence remains required.

Apple Private Cloud Compute core requirements: <https://security.apple.com/documentation/private-cloud-compute/corerequirements>. Inspiration for explicit privacy boundaries, not a claim that this implementation has equivalent properties.