Maestro.

The AI-native, agentic adaptive learning system

DEPLOYMENT GUIDE

Customer deployment reference — overview, requirements, architecture and preparation

Version 1.1 · Platform release 3.0 · July 2026
maestroed.ai

Document Control

Field	Value
Document	Maestro — Deployment Guide
Version	1.1 (aligned to Maestro platform release 3.0)
Date	July 2026
Audience	Customer IT / infrastructure teams, database administrators, network and security teams, and the institutional project owner
Purpose	Defines what Maestro is, the deployment architecture, the minimum requirements, and everything the customer must prepare before installation
Classification	Public — customer deployment documentation

How to use this guide

Sections 1–2 give decision-makers a concise picture of the system and its deployment architecture. Sections 3–4 are the operational core: the minimum requirements and the customer preparation checklists. Section 5 outlines the deployment procedure performed with the Maestro team, and Sections 6–7 cover security and ongoing operations. Completing every checklist in Section 4 before the scheduled deployment is the single best predictor of a smooth go-live.

Contents

1 System Overview

1.1 What Maestro is

Maestro is an AI-native, agentic adaptive learning system for higher education. It takes a course syllabus and its reference materials and re-engineers them into a governed, node-based adaptive curriculum — then delivers that curriculum to learners as a transparent, mastery-based journey. The work is done by a coordinated system of specialised AI agents (§1.3, §1.5), but human experts remain the authority throughout: every academic decision passes through a faculty or subject-matter-expert (SME) approval gate, and the agents never author or grade on the institution's behalf without governance.

Maestro augments rather than replaces the institution's existing systems. The institution keeps its LMS and student information system as the system of record; Maestro adds an adaptive authoring, delivery, analytics and governance layer on top, and can connect to the LMS through the LTI 1.3 standard.

1.2 The four products

Maestro ships four products on one platform and one data model. A single deployment provides all four.

Product	Role	What it does
Studio	Course authoring	The AI-powered authoring environment. AI agents do the heavy lifting at every stage: extracting and refining the syllabus, researching and grounding content in the institution's own references, generating adaptive nodes, blueprints and learning objects across modalities (text, structured visuals, video, interactive Evidence Checks), and pre-reviewing everything through pedagogy, accuracy, integrity and accessibility lenses. The benefit: a course is re-engineered in a fraction of the traditional time and at higher, more consistent quality — while faculty and SMEs remain the authority who review and approve every artifact before it advances.
Journey	Learner delivery	The learner runtime, with AI working on two levels. An adaptive AI engine interprets every piece of learning evidence in real time, updates each learner's Bayesian learner model, and manages the adaptability of every node while the learner is learning — adjusting depth, pacing, remediation and the next best step for that individual. Alongside it, an always-present AI companion (text and voice) coaches and supports without ever policing. Learners reach Journey by standalone enrollment or LTI 1.3 launch from the institutional LMS.
Pulse	Analytics	The landscape view of learning. Pulse monitors the progress of all learners — the entire learner landscape at once — across the full life cycle of the course, from first enrollment through every node to completion. It shows live cohort mastery by concept, surfaces shared misconceptions the moment they emerge, and drives an intervention life cycle so staff act on evidence, not assumptions.
Compass	Governance	Curriculum governance: full audit trail, node split/merge with record preservation, records views, and improvement suggestions — keeping the AI credible, controlled and aligned to institutional standards.

1.3 How a course moves through Maestro

Maestro is an agentic system: specialised AI agents are invoked at each stage of the pipeline, do their work, and hand the result forward to the next stage — always through a human approval gate. As a course moves through the pipeline:

- **Ingest.** Ingestion agents parse the syllabus (PDF/DOCX or form entry) and the reference corpus (PDF, DOCX, PPTX, EPUB, websites, video sources, pasted text); research agents enrich the material and prepare it for SME review.
- **Architect.** Course-architect agents refine learning outcomes, redesign assessment, apply a weighting rubric and integrity review, and produce the subtopic architecture — each layer gated by academic approval before the next agents are invoked.
- **Generate.** Generation agents turn approved subtopics into adaptive nodes: for every node, blueprint agents design the learning, grounding agents anchor it in the approved references, and production agents create the learning objects in the modality the learning requires. Critic agents review each object before it ever reaches the SME.
- **Publish.** Approved courses are published to Journey for delivery.
- **Deliver and adapt.** Judging agents evaluate learner evidence as it is produced; the adaptive engine folds it into the Bayesian learner model and continuously manages each node's adaptability, so every learner progresses on demonstrated mastery.
- **Observe and govern.** Mining agents cluster misconception patterns across the cohort for Pulse; Compass maintains the audit trail and curriculum quality over time.

1.4 Platform characteristics relevant to deployment

- **Delivered as a complete product.** Maestro ships as a pre-built installation package containing the application services, the two web applications and all runtime components. The customer provides infrastructure and configuration — never source code, builds or development tooling.
- **Compact application tier.** The platform services — APIs, background workers and the realtime voice gateway — run as a self-contained application tier with a small footprint, deployable on one server or across several for resiliency (§2.4).
- **PostgreSQL is the single source of truth.** All entities, settings, authentication, audit events, pipeline artifacts, reference chunks and vector embeddings live in PostgreSQL with the pgvector extension — on a database instance installed with the platform or on an existing institutional database server (§3.3).
- **Graceful degradation.** The optional graph-visualization database and the background job system degrade gracefully if unavailable; core operation continues.
- **AI runs through the customer's own service accounts.** The platform connects to a supported generative AI service, an embedding and realtime voice service, and optionally an AI video rendering service — all under accounts the customer controls. A locally hosted model endpoint is supported where policy requires it. Browsers never contact AI services directly; all AI traffic is server-mediated.
- **Configurable capabilities.** LTI integration, the AI companion, misconception mining and universal ingestion are individually switchable, so a deployment can start minimal and grow (Appendix A).

1.5 The multi-agent system — how it runs

Under the hood, Maestro is one coordinated multi-agent system. Each agent in §1.3 carries a narrow mandate — extract, research, ground, blueprint, produce, critique, judge, mine — with its own quality gate, and the platform composes them into the pipelines that engineer, deliver and observe a course. Two

architectural choices behind this matter to a deployment. First, the agents run on a durable workflow-orchestration layer: every agent invocation is a checkpointed, resumable, fully audited step in a long-running workflow. If a server restarts, an AI service times out or a pipeline is interrupted mid-course, work resumes from the last completed step — nothing is half-done, duplicated or lost, and every step stays traceable in the audit trail. Second, the platform separates its write path from its read path: learning evidence and academic decisions are captured once, immutably, as the system of record, while purpose-built read views serve delivery and analytics. That separation is why Pulse can render the live landscape of an entire cohort without ever slowing the learners who are producing the evidence.

> *Both layers ship inside the platform package and run within the application tier and the PostgreSQL database — they add no extra infrastructure, servers or middleware to the topologies in §2.4.*

2 High-Level Deployment Architecture

2.1 Architecture diagram

A Maestro deployment is deliberately compact: a self-contained application tier, one required database, the web applications served through an application gateway, and controlled outbound connections to AI services. The same architecture runs in every topology in §2.4.

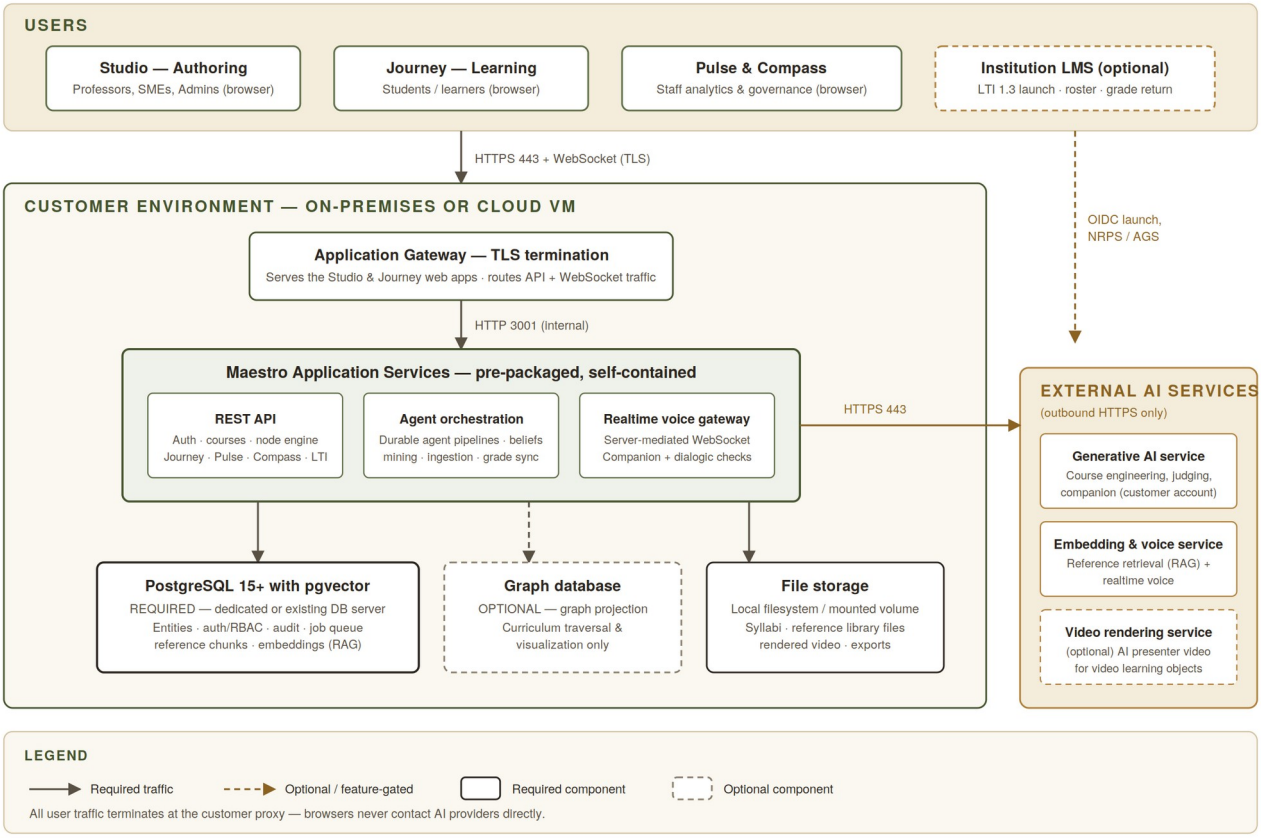

Figure 1 — Maestro high-level deployment architecture

2.2 Components

Component	Required	Description
Application gateway	Yes	Terminates TLS on port 443, serves the Studio and Journey web applications, and routes API plus WebSocket traffic to the application services. Installed with the platform, or fronted by the institution's existing load balancer / gateway. Must support WebSocket pass-through.
Maestro application services	Yes	The pre-packaged application tier (internal port 3001): platform APIs, the agent-orchestration and background workers that run the durable agent pipelines (§1.5 — learner-model updates, mining, ingestion, milestones, LTI grade sync), and the server-mediated realtime voice gateway.
PostgreSQL 15+ with pgvector	Yes	Primary source of truth and job queue; the pgvector extension powers reference retrieval (RAG). Installed with the platform on a dedicated instance, or hosted on an existing institutional database server (§3.3). The application connects as a least-privilege role.
File storage	Yes	Local volume or institutional storage (NFS/SAN) for uploaded syllabi, reference library files, avatars, rendered videos and exports. Metadata lives in PostgreSQL.
Graph database (optional)	Optional	Secondary graph projection used only for curriculum traversal and visualization. The platform starts and operates fully without it.
External AI services	Yes (outbound)	A generative AI service for course engineering and judging; an embedding and realtime voice service; optionally an AI video rendering service. All under the customer's own accounts, outbound HTTPS only. The supported-services list is provided during deployment planning.
Institution LMS	Optional	Any LTI 1.3-compliant LMS for launch, roster sync (NRPS) and grade return (AGS).

2.3 Network ports and traffic flows

Traffic divides cleanly into external flows crossing the environment boundary and internal flows that stay on the private network. Firewall rules should be provisioned exactly as listed; nothing else is required.

External traffic — inbound (from user networks / internet to the platform)

Flow	Port	Notes
Users → application gateway	443 (HTTPS + WSS)	All Studio, Journey, Pulse and Compass traffic, including WebSocket for voice features. The only port ever exposed to users.
LMS → platform (optional)	443 (HTTPS)	Only when LTI is enabled: the LMS fetches the platform's public keys (JWKS) and browsers are redirected through launch endpoints — all on the same 443.

External traffic — outbound (from the application services to the internet)

Flow	Port	Notes
App → generative AI service	443 (HTTPS)	Course engineering, judging, companion. Exact endpoint hostnames are supplied during deployment planning for allow-listing.
App → embedding & voice service	443 (HTTPS + WSS)	Reference retrieval embeddings and realtime voice sessions.
App → video rendering service	443 (HTTPS)	Only if AI presenter video rendering is enabled.
App → institution LMS (optional)	443 (HTTPS)	LTI token and roster endpoints; grade passback (AGS).
App → web (ingestion)	443 (HTTPS)	URL/website ingestion of reference sources. Built-in SSRF protections block private and link-local ranges; egress can additionally be restricted by policy.

Internal traffic (private network only — never exposed)

Flow	Port	Notes
Gateway → application services	3001 (HTTP/WS)	Restricted to the gateway host(s).
App → PostgreSQL	5432	Restricted to the application server(s).
App → graph database (optional)	7687	Restricted to the application server(s).
App → file storage (if networked)	NFS/SMB as applicable	Only when file storage is mounted from institutional storage rather than a local volume.

2.4 Deployment topologies: on-premises and cloud

Maestro supports a range of topologies, from a compact single-server installation to a resilient multi-server estate. The architecture is identical throughout; the topology decides where each tier runs and what level of availability it provides. The Maestro team sizes and validates the chosen topology with the customer during deployment planning.

Topology 1 — Standard (single server)

All tiers — gateway, application services, database and file storage — on one server or VM. Suitable for evaluation, testing and smaller production cohorts. Recovery relies on VM-level protection (snapshots, hypervisor HA) plus the nightly backups in §7.

Topology 2 — Recommended production (separated data tier)

The application tier (gateway + application services) on one server; PostgreSQL on a separate dedicated server or on an existing institutional database service; file storage on institutional storage (NFS/SAN) or a dedicated volume. This is the recommended production baseline: it isolates the data tier for performance, backup and security, and lets each tier be sized and maintained independently.

Topology 3 — High availability

- Application tier: two or more application servers behind the institution's load balancer in an active–passive arrangement, removing the single server as a point of failure.
- Data tier: PostgreSQL with streaming replication to a standby (or an institutional/managed database service with built-in HA and automated failover).
- File storage: replicated or highly available institutional storage shared by the application servers.
- Recovery objectives: with Topology 3 plus nightly logical backups, typical objectives of RPO ≤ 24 hours (backup-based) or near-zero (replication-based) and RTO under one hour are achievable; exact objectives are agreed during deployment planning.

On-premises deployments

Each server in the chosen topology is specified in §3.1 (hardware) and §3.2 (software). The institution's firewall team opens exactly the flows in §2.3: inbound 443 to the gateway from user networks; the internal ports between tiers restricted to the specific hosts; and outbound 443 from the application server(s) to the allow-listed AI service endpoints, the LMS (if used) and web ingestion. No inbound access to the database or application ports from outside the private network is ever required.

Cloud VM deployments

The same topologies map directly onto any major cloud: VMs in a private VNet/VPC, port 443 published through a cloud load balancer or firewalled public IP, and outbound rules per §2.3. A managed PostgreSQL service may take the place of a self-hosted database server provided it meets the database requirements in §3.3 (PostgreSQL 15+ with the pgvector extension) and sits in the same region and private network. Use a persistent data disk or managed file service for file storage, and cloud-native snapshots alongside the logical database backups.

> *In every topology, Maestro remains a customer-controlled deployment: learner data stays in the customer's environment, and only model prompts/requests leave it — over TLS, to the AI services configured under the customer's own accounts.*

3 Minimum Requirements

3.1 Hardware

Sizing is driven mainly by cohort size, the volume of the reference library (vector search), and content-generation activity. The figures below are validated starting points per server role; larger cohorts, additional servers for the high-availability topology (§2.4), or estates beyond 2,000 learners are sized with the Maestro team during deployment planning.

Profile	Minimum — evaluation & testing (up to 200 learners)	Production (up to 2,000 learners)	Notes
CPU	4 vCPU	8+ vCPU	Content generation is I/O-bound on external AI services; delivery is lightweight.
RAM	16 GB	32 GB	Minimum profile includes the database on the same server (Topology 1).
Storage	100 GB SSD	250–500 GB SSD (NVMe recommended)	Reference libraries and rendered video dominate growth; vector indexes benefit from fast storage.
Database server	Co-located on the same server	Separate server or existing institutional database service — 4 vCPU / 16 GB dedicated baseline	See §3.3 for what the database server must provide.
Network	1 Gbps internal	1 Gbps internal	Stable outbound internet ≥ 50 Mbps for AI service traffic.

3.2 Software

Maestro is delivered as a pre-built installation package: the application services, both web applications, the application gateway and all runtime components are included and installed by the package. The customer provides only the base environment below.

Component	Requirement	Provided by
Operating system	Ubuntu 22.04 LTS or equivalent modern 64-bit Linux, patched and NTP-synchronized	Customer
Database engine	PostgreSQL 15+ with the pgvector extension (§3.3)	Installed with the platform, or an existing institutional database server
Application runtime & web tier	All application services, web applications and gateway components	Included in the Maestro installation package
Graph database (optional)	Only for curriculum graph visualization	Included as an optional platform component
TLS certificate	Valid certificate for the platform hostname	Customer (institutional or public CA)
Load balancer (Topology 3 only)	Institutional load balancer with WebSocket support	Customer

3.3 Database requirements

The platform requires PostgreSQL and supports two arrangements: a dedicated instance installed with the platform (default), or an existing institutional database server that the customer operates. If the customer provides the database server, it must meet the following:

- PostgreSQL 15 or later with the pgvector extension available. The extension is enabled once per database by a superuser; the application itself never requires superuser rights.
- A dedicated database for Maestro and a dedicated, least-privilege application role that owns the application schema. Maestro keeps all tables in its own schema, so the server can be shared with other applications.
- Network reachability from the application server(s) on port 5432 (private network only), with capacity for the connection pool of each application server.
- UTF-8 encoding; reliable clock (NTP); standard PostgreSQL tuning for the host's RAM; storage sized per §3.1.
- Backup capability: nightly logical backups with 30-day retention (§7), plus replication/HA options if Topology 3 is chosen (§2.4).

3.4 Network and connectivity

- A DNS name for the platform (e.g. maestro.institution.edu) and a valid TLS certificate for it.
- Inbound: HTTPS 443 only, from user networks to the application gateway. WebSocket upgrades must pass through any intermediate proxies/firewalls (required for voice features).

- Outbound HTTPS (443) from the application server(s) to the configured AI service endpoints — exact hostnames are supplied during deployment planning for firewall allow-listing — plus the institution's LMS endpoints (only if LTI is enabled) and general web access for URL ingestion (may be policy-restricted).
- Internal: the flows and ports listed in §2.3, restricted to the specific hosts of the chosen topology.

3.5 AI service requirements

Maestro consumes external AI capabilities as services, always under accounts the customer owns and controls. The list of supported services for each category is maintained by Maestro and provided during deployment planning.

Service category	Required	Used for
Generative AI service	Yes	All course-engineering and content generation, Evidence Check judging, AI companion chat, rubric grading. Model selection per pipeline stage is configurable in the admin centre.
Embedding & realtime voice service	Yes	Text embeddings for reference retrieval (RAG) and the realtime voice features (companion voice, dialogic Evidence Checks).
AI video rendering service	Optional	Rendering AI presenter videos for video learning objects. Without it, video briefs are still produced for conventional production.
Locally hosted model endpoint	Optional	Where policy requires on-premises inference, a locally hosted model endpoint may serve generation; discovery and health checks are built into the admin centre.

> *Service credentials are institutional accounts and should be provisioned with billing limits and usage monitoring. Costs scale with authoring activity (one-time per course) and learner interaction volume (ongoing).*

3.6 End-user workstation requirements

- A current evergreen browser (Chrome, Edge, Firefox or Safari, latest two major versions). No local installation is required for any role.
- WebSocket connectivity through the institution's network to the platform hostname (port 443).
- A microphone (and permission to use it) for learners using voice interactions; audio output for video and voice content.
- Display 1366×768 or higher recommended. The learner experience meets WCAG 2.2 AA accessibility standards.

4 Customer Preparation Requirements

Everything in this section is the customer's responsibility to have ready before the deployment session. Items marked optional apply only if the corresponding capability is in scope.

4.1 Infrastructure checklist

Item to prepare	Details	Done
Server(s) provisioned	Linux server/VM per §3.1–3.2, with OS patched, NTP synchronized, and SSH access for the deployment engineer.	☐
Database arrangement decided	Either the platform-installed database (default) or an existing institutional PostgreSQL server meeting §3.3, reachable from the app server, with a superuser available for the one-time pgvector extension creation.	☐
DNS record created	Platform hostname (e.g. maestro.institution.edu) resolving to the reverse proxy / load balancer.	☐
TLS certificate issued	Valid certificate and key for the platform hostname (institutional CA or public CA / ACME).	☐
Firewall rules applied	Inbound 443 open to users; outbound 443 open to AI endpoints (§2.3); internal DB ports restricted to the app server.	☐
Storage volume mounted	Data directory for uploads/generated assets with the capacity from §3.1, included in backup scope.	☐
Backup target available	Destination for nightly database dumps and file-storage backups, 30-day retention.	☐

4.2 Accounts, keys and secrets checklist

Item to prepare	Details	Done
Generative AI service credentials	API credentials for a supported generative AI service, provisioned under an institutional account with billing limits (§3.5).	☐
Embedding & voice service credentials	Required for reference retrieval (RAG) and voice features; may be the same account as above depending on the chosen service.	☐
Video rendering credentials (optional)	Only if AI presenter video rendering is in scope.	☐
Database credentials (existing server only)	Only when using an existing institutional database server: a least-privilege application role and password, with database and schema names agreed with the DBA. With the platform-installed database, this is handled during installation.	☐
Initial admin account chosen	Email address and strong password for the first Maestro administrator (created at first start).	☐

4.3 People and roles

Role	Responsibility during and after deployment
Project owner	Institutional decision-maker; approves scope, pilot course selection and go-live.
System administrator	Hosts and operates the server(s); first Maestro admin account; manages users, settings, keys and audit in the admin centre.
Database administrator	Provisions PostgreSQL, enables pgvector, owns backup/restore.
Network/security officer	DNS, TLS, firewall/egress rules; reviews §6 of this guide.
Faculty / SMEs	Author and approve courses in Studio; own every academic gate. At least one professor account holder should be identified for the pilot.
LMS administrator (optional)	Registers Maestro as an LTI 1.3 tool and configures NRPS/AGS (§4.5).

4.4 Academic content inputs

Maestro builds courses from the institution's own material. For each pilot course, prepare:

- The syllabus as PDF or DOCX (or the equivalent details for form entry): course metadata, learning outcomes, assessment plan, weekly/topic structure.
- The reference corpus: textbooks and readings as PDF/DOCX/PPTX/EPUB files, plus any web pages or online video sources to ingest. Confirm the institution holds the rights to use them.
- Assessment philosophy and rubrics, where they exist — grading remains on the institution's standards.
- The names of the SMEs who will review and approve each stage.

4.5 LMS / LTI 1.3 integration prerequisites (optional)

If learners will launch Maestro Journey from the institutional LMS, the LMS administrator must prepare:

- An LTI 1.3-capable LMS reachable over HTTPS from the Maestro server, with clocks NTP-synchronized on both sides.
- Registration of Maestro as an external tool, using the login-initiation, launch and JWKS URLs supplied by the Maestro team at deployment.
- The platform-side values Maestro needs for registration: issuer URL, client ID, deployment ID, and the platform's authorization, token and JWKS endpoints.
- Assignment and Grade Services (AGS) and Names and Role Provisioning Services (NRPS) enabled for the tool, so grades can flow back and rosters can sync.

> *LTI is disabled by default and enabled per deployment. A self-contained LMS test environment ships with the platform for pre-production round-trip testing.*

4.6 Operational readiness

Item to prepare	Details	Done
Monitoring hook	Institutional monitoring polling GET /api/health (reports database, jobs and graph status) with alerting on non-OK status.	☐

Item to prepare	Details	Done
Backup schedule	Nightly pg_dump of the Maestro database (30-day retention) plus file-storage backup; quarterly restore verification agreed.	☐
Retention policy accepted	Raw telemetry pruned at 180 days; evidence records and the belief ledger are permanent academic records (§6).	☐
Maintenance window	Agreed window for updates; changes to configurable capabilities require a brief service restart.	☐
Support path	Named contacts on both sides and an agreed channel for incident escalation.	☐

5 Deployment Overview

Deployment is performed together with the Maestro team, typically within one working day once Section 4 is complete. The high-level procedure:

5.1 Deployment steps

#	Step	Summary
1	Provision infrastructure	Servers, network, DNS, TLS and firewall rules per the chosen topology (§2.4) and the checklists in §4.
2	Prepare the database	Platform-installed database, or the customer's existing PostgreSQL server prepared per §3.3 (application role, database, one-time pgvector extension).
3	Install the platform	Run the Maestro installation package on the application server(s). It installs the application services, both web applications, the gateway and all runtime components — nothing is built or downloaded from source on site.
4	Configure	Apply the deployment configuration: platform hostname and TLS, database connection, AI service credentials, initial administrator account, and the optional capabilities in scope (§5.2).
5	Initialize	On first start the platform applies and verifies its database schema automatically and creates the initial administrator account.
6	Verify	The platform health endpoint reports all services connected; administrator login works; the end-to-end platform check passes on a sample course (§5.3).
7	Enable options	Switch on LTI, companion voice or video rendering as scoped; register the platform with the institutional LMS if applicable.
8	Handover	Administrator walkthrough of the admin centre, monitoring and backup verification, and sign-off against §5.3.

5.2 Key configuration parameters

All configuration is applied during installation and managed thereafter from the admin centre or the platform configuration. The essential parameters:

Parameter	Required	Purpose
Platform hostname & TLS certificate	Yes	The public name users browse to; certificate installed at the application gateway.
Database connection	Yes	Host, database name and least-privilege application credentials (platform-installed or institutional server, §3.3).
Initial administrator account	Yes	Email and password for the first admin, created at first start.
AI service credentials	Yes	Generative AI service; embedding & voice service; video rendering service if in scope (§3.5).
Optional capabilities	As scoped	LTI integration, AI companion, misconception mining, universal ingestion (Appendix A). Changing a capability requires a service restart.
Graph database connection	Optional	Only when the curriculum graph visualization component is deployed.

5.3 Verification

- Health: the platform health endpoint reports overall status plus database, background-job and optional-component status, and the live capability configuration. The database must show connected for an OK status.
- Functional gate: the built-in end-to-end platform check exercises publish → enroll → deliver → evidence submission → learner-model update → analytics on a sample course.
- Load posture (production): the built-in evidence-pipeline load check must pass with zero server errors and p95 < 1.5 s at the scoped concurrency.
- If LTI is in scope: a full LMS round trip — launch, roster fetch, grade return — against the institutional LMS or the bundled LMS test environment.

6 Security and Compliance Considerations

6.1 Access control

- All platform APIs require authentication (JWT sessions); role-based access control distinguishes admin, professor and student, with course-scoped checks on every authoring and delivery route.
- Admin capabilities (users, settings, keys, prompts, audit) are restricted to the admin role; professor-to-professor review and course access are explicitly assigned.
- A full audit log records administrative and academic actions and is queryable from the admin centre (and surfaced in Compass).

6.2 Learner data protection

- Assessment answer keys, traps and grading maps never reach the learner's browser — published payloads are sanitized server-side.
- Staff-only integrity signals are structurally separated from anything learner-facing; the AI companion's grounding excludes them by design, in line with FERPA-style data boundaries.
- Voice interactions are server-mediated: the browser never connects to the AI provider, session instructions stay server-side, and voice sockets are authorized by single-use, 60-second, purpose-typed tickets. Voice audio is never stored — only transcripts.
- Learner evidence records and the belief ledger constitute the academic record and are retained permanently; raw telemetry is pruned at 180 days.

6.3 Platform hardening

- URL/website ingestion is SSRF-hardened: private, link-local and metadata address ranges are blocked, redirects are capped and re-validated, and file types are detected by content, not extension.
- For split-origin deployments (frontends on a different origin than the API), CORS must be tightened to the exact origins during configuration.
- Secrets (database credentials, session-signing secret, AI service credentials) are managed by the platform in its server-side configuration with restricted permissions — generated or applied at installation, and never present in anything delivered to the browser.
- Run the application as an unprivileged system user; expose only port 443; keep OS and runtime patched.

6.4 What leaves the customer environment

Only AI-provider traffic leaves the deployment: prompts and content fragments required for generation, judging, embeddings and voice, sent over TLS to the configured providers under the customer's own API accounts. No learner data is sent to Maestro (the vendor) and there is no telemetry callback — the customer's instance is self-contained.

7 Operations and Maintenance

7.1 Routine operations

Activity	Frequency	Detail
Health monitoring	Continuous	Poll GET /api/health; alert when status is not ok or background jobs report failure.
Database backup	Nightly	Logical dump (pg_dump, custom format) of the Maestro database; retain 30 days; back up the file-storage directory alongside.
Restore verification	Quarterly	Restore the latest dump into a scratch database and verify.
Telemetry pruning	Monthly	Delete raw telemetry events older than 180 days (provided SQL); evidence and the belief ledger are never pruned.
Vector index rebuild	After bulk ingestion	Rebuild the pgvector HNSW index after large reference-library loads.
pgvector upgrades	As released	A provided script handles extension version bumps.

7.2 Behaviour under partial failure

- PostgreSQL unreachable → the server fails fast at startup by design (it is the source of truth). Restore the database connection and restart.
- Background jobs fail to start → the API continues serving; an hourly evidence sweep folds any records submitted while workers were down (idempotent), so learner evidence is not lost.
- Graph database down → curriculum graph visualization degrades; all entity reads and writes continue.
- AI service outage → authoring/generation pauses; published course delivery of already-produced content continues.
- Interrupted agent pipelines → durable workflow orchestration (§1.5) resumes each pipeline from its last completed step once service is restored; no partial or duplicated artifacts are produced.

7.3 Incident quick reference

Symptom	First checks
Platform unreachable	Backend service status and logs; startup fails fast if PostgreSQL is unreachable.
Learner progress not updating	Job status on /api/health; the hourly sweep self-heals once jobs run.
Voice not connecting	AI service credential validity; WebSocket pass-through on proxies; server logs for upstream errors.
LTI launch fails	Platform registration values (issuer / client ID / deployment ID), clock skew, JWKS reachability.
Ingestion stuck	Ingestion queue depth; per-source status and error columns in the admin view.

A Appendix — Configurable Capabilities

Platform capabilities are individually switchable per deployment, so an institution can start minimal and grow. Capabilities are set during installation and can be changed later; a change requires a brief service restart.

Capability	Default	Controls
Journey runtime	On	Learner delivery: enrollment, sessions, adaptive path delivery.
Background processing	On	Learner-model updates, scheduled re-checks, mining and sync jobs.
LTI 1.3 integration	Off	LMS launch, roster sync (NRPS) and grade return (AGS). Requires LMS registration (§4.5).
AI companion	On	Learner chat, proactive support triggers, and voice.
Misconception mining	On	Cohort misconception pattern mining and the SME review queue feeding Pulse.
Universal ingestion & research	On	Drop-anything reference ingestion and the AI research pass.

Maestro — learning where mastery is real, and everyone stays in the lead.