

Enterprise AI. Inside Your Organisation.

A governed, Kubernetes-powered platform for enterprise knowledge, automation and AI application delivery.



Deployment	Operating model	Platform scope
Customer-controlled infrastructure	Connected, restricted or air-gapped	Work with AI Build Automate Govern Operate

Technical and solution overview for CIO, CTO, security, infrastructure, procurement and implementation teams.

One governed platform for the enterprise AI journey

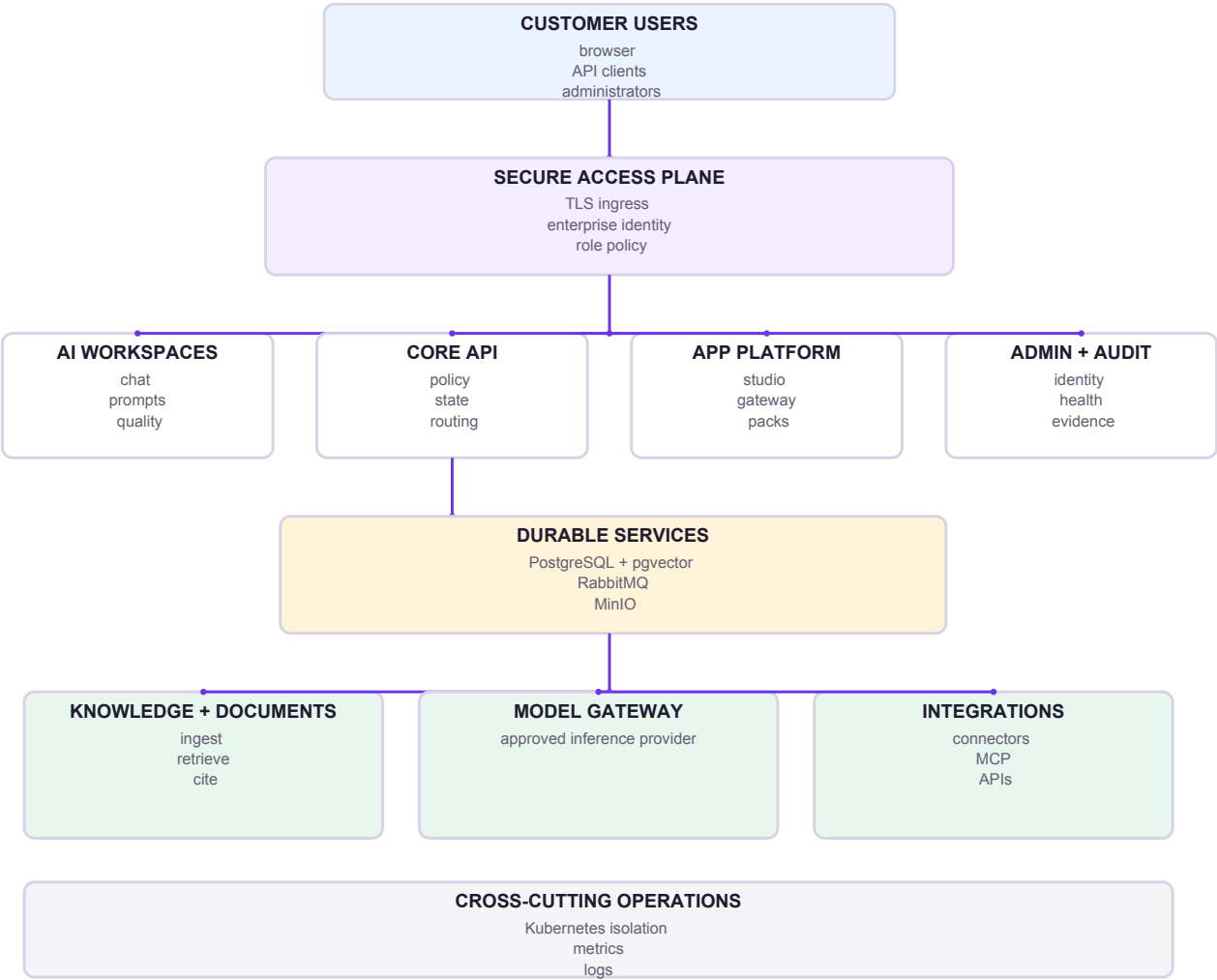
Nexus One AI is deployed on infrastructure controlled by the customer. It combines everyday AI access, enterprise knowledge, document intelligence, evaluation, automation, application creation and operational governance behind one identity and management plane.

Business requirement	Nexus One AI response
Keep sensitive information controlled	Prompts, documents, application state and AI workloads remain within the approved customer deployment boundary.
Move beyond isolated pilots	One identity, knowledge, automation, application and operations layer replaces disconnected proof-of-concept stacks.
Generate answers that can be checked	Nexus Notebook retrieves governed source evidence and returns cited answers and exportable outputs.
Automate without losing control	Agents and workflows run through durable execution with permissions, history and human approval boundaries.
Operate AI as enterprise infrastructure	Health, metrics, logs, audit, backup, restore evidence and lifecycle controls are included in the platform architecture.

Designed for enterprise adoption
Start with a controlled workload and expand across teams without replacing the underlying governance and operations model.

Reference architecture

The architecture separates secure access, product services, durable data, model execution, integrations and operational controls. Customer identity, infrastructure, data and policy remain authoritative.



Model-provider independence
Applications use a logical model gateway. Approved local or customer-selected compatible inference services can be changed without redesigning every application.

Kubernetes platform foundation

Nexus One AI is packaged as a Kubernetes application platform. K3s is used as the compact Kubernetes distribution for appliance and single-node deployments; larger designs can adopt an approved multi-node topology. Kubernetes provides a consistent control model for service deployment, health, networking, storage, isolation and upgrades.

Kubernetes building block	How Nexus One AI uses it
Namespaces	Separate platform services, durable data, observability, developer workspaces, capability packs and application previews.
Deployments	Run stateless services such as the portal, Core API, application gateway, workers and document services.
StatefulSets	Run stateful platform services where stable identity and persistent volumes are required.
Services	Provide stable in-cluster addresses for APIs, databases, queues, object storage and applications.
Ingress	Routes approved browser and API traffic through a controlled TLS entry point.
ConfigMaps and Secrets	Separate non-sensitive runtime configuration from protected credentials and keys.
Jobs and CronJobs	Execute controlled one-time, scheduled, maintenance and evidence tasks.
PersistentVolumeClaims	Retain database, object, workspace and operational data across pod restarts.
ServiceAccounts and policies	Bound workload identity and permissions for services, previews and extensions.

Why Kubernetes matters

The same application contract can be installed, inspected, upgraded and operated consistently across supported customer infrastructure profiles.

Namespace and workload architecture

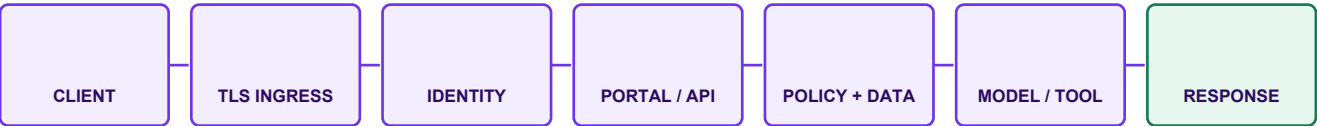
Logical namespace	Principal workloads	Isolation purpose
Platform	Portal, Core API, identity adapter, App Gateway, scheduler, workers, Open WebUI, document services	Customer-facing services, protected APIs and background execution.
Data	PostgreSQL with vector search, RabbitMQ and MinIO	Authoritative data, durable job transport and governed object storage.
Observability	Prometheus, Alertmanager, Grafana, Loki, Alloy and exporters	Metrics, dashboards, alerts, logs and operational evidence.
Workspaces	JupyterHub and persistent developer sessions	Controlled analysis, prototyping and SDK use.
Application previews	App Studio preview Deployments, Services and configuration	Isolation of generated applications before approval and publication.
Capability packs	Pack-specific workloads and dependencies	Bounded extension installation, upgrade, rollback and removal.
Kubernetes system	Ingress controller, DNS, metrics and storage provisioner	Cluster routing, discovery, resource metrics and volume provisioning.

Production topology options

- Compact appliance: control plane and data services on one managed node.
- GPU-separated design: control plane remains available while inference runs on an approved accelerator node or provider.
- Enterprise cluster: multiple worker nodes, resilient storage and workload placement according to the customer availability design.

Synchronous request path

Interactive browser and API requests follow a controlled path from identity to policy, data and model execution.

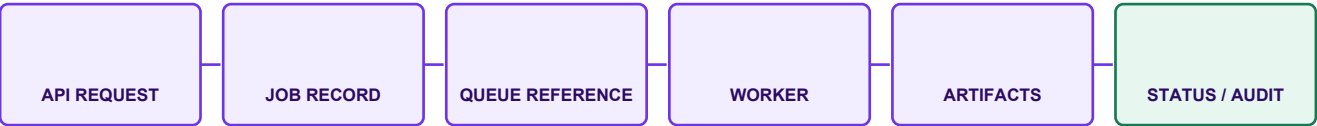


Stage	Background responsibility
Client	Browser, approved application or API client initiates the request.
TLS ingress	Terminates approved TLS, applies host/path routing and forwards only configured routes.
Identity	Validates the user session or API credential and supplies role claims.
Portal / Core API	Validates input, applies entitlement and business policy, and records the request context.
Policy and data	Checks resource access and retrieves authoritative state or approved knowledge.
Model or tool	Invokes the logical model gateway, retrieval service, connector or application capability.
Response and evidence	Returns the result and preserves relevant audit, usage, citation or execution evidence.

No direct browser-to-database access
The browser interacts with protected services. Databases, queues, object stores and connector credentials remain behind the application and policy layers.

Durable background-job architecture

Long-running work is separated from interactive HTTP requests. This prevents document processing, scheduled workflows, evaluation, agent runs and other background tasks from depending on one browser session.



Component	Responsibility
Core API	Validates the request, authorization and inputs; creates an authoritative job record.
PostgreSQL	Stores job state, ownership, parameters, progress, errors and final references.
RabbitMQ	Transports a durable reference to work that is ready for execution.
Scheduler	Creates due work for approved schedules and recurring operations.
Worker	Claims the job, performs bounded steps and updates progress.
MinIO / persistent storage	Stores large inputs, exports, generated artifacts and evidence objects.
Portal and notifications	Read durable state to show running, completed, failed or action-required status.
Audit	Preserves who initiated, approved, changed or stopped consequential activity.

Failure behaviour
A failed worker or pod does not create a simulated success. Durable state allows the platform to report the real outcome and, where policy permits, retry safely.

Data and storage architecture

Data service	What it stores	Operational role
PostgreSQL	Users, roles, collections, documents, prompts, jobs, workflows, applications, approvals, audit and configuration	Authoritative transactional source of truth.
Vector search / pgvector	Embeddings and retrieval indexes associated with governed sources	Semantic retrieval for Notebook and RAG workflows.
MinIO object storage	Original documents, generated artifacts, exports, application packages, evidence and large payloads	Durable object storage with references held in authoritative records.
RabbitMQ	References to queued work and delivery state	Asynchronous transport; not the source of truth.
Persistent volumes	Database files, object data, workspaces and service-specific retained state	Persistence across pod recreation and controlled upgrades.
Backup repository	Encrypted database, object, configuration and platform recovery material	Recovery according to the customer-approved retention and off-host design.

Data lifecycle



Customer data classification, retention, deletion, encryption, legal hold and backup-location requirements are implemented through the approved deployment profile and operating procedures.

Model serving and RAG internals

Nexus separates model access from enterprise retrieval. This allows general AI assistance and evidence-grounded knowledge workflows to use different controls while sharing the same identity, audit and operations plane.

Path	Processing sequence
General AI request	User -> AI Workspace -> identity/policy -> logical model gateway -> approved inference provider -> response and usage evidence.
Knowledge ingestion	Source -> type/policy checks -> extraction/OCR -> metadata -> chunks -> embeddings -> vector/index storage.
Grounded question	Question -> collection ACL -> hybrid retrieval -> reranking -> approved evidence -> model generation -> citations and export.
Evaluation	Test case -> selected model or retrieval path -> result -> scoring/judge policy -> persisted quality and safety evidence.
Dataset generation	Approved sources -> grounded examples -> human review -> immutable version -> evaluation or approved training request.
Model training request	Published dataset -> compatibility and entitlement checks -> signed runtime/hardware boundary -> adapter/artifact and recorded outcome.

Inference boundary

The logical model gateway can target an approved local runtime or customer-selected compatible provider. Model size, quantization, context, concurrency and accelerator capacity determine performance.

App Studio runtime isolation

App Studio does not publish generated code directly into the core platform. Each application follows a governed lifecycle and executes inside a bounded Kubernetes runtime.



Lifecycle element	Kubernetes and platform control
Project and workspace	Durable project record, source history and customer ownership.
Sandbox	Temporary isolated execution with resource limits, bounded identity and controlled dependencies.
Preview	Dedicated Deployment, Service and configuration in the preview boundary.
Independent review	A reviewer approves the application version and evidence, separate from the creator.
Release	Approved version is routed through the application gateway under customer access policy.
Operate	Health, logs, usage and lifecycle state are visible without granting cluster-wide access.
Rollback / retire	Routing and version state return to an approved predecessor or remove access while retaining audit history.
Cleanup	Temporary preview resources are stopped and deleted when no longer required.

Exact container images, dependency packs, egress permissions and resource quotas are governed by the customer deployment and pack policy.

Observability data path

Operations are collected independently from user-facing business data. Metrics, logs, alerts and evidence are routed to purpose-built services and exposed through role-controlled dashboards.

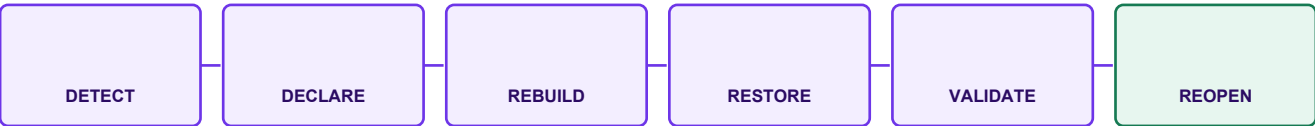
Telemetry source	Collection and destination	Operator use
Kubernetes and workloads	Prometheus metrics through ServiceMonitors/exporters	Readiness, resource saturation, error rate and service trends.
Host and GPU	Node and supported GPU/DCGM exporters	CPU, memory, disk, GPU temperature, utilization and VRAM.
PostgreSQL	Database exporter and platform health probes	Connections, transactions, capacity and database health.
Application logs	Alloy collection into Loki	Searchable troubleshooting and incident context.
Rules and alerts	Prometheus rules into Alertmanager	Warning/critical routing and operational acknowledgement.
Dashboards	Grafana with customer identity integration	Platform, inference, node, GPU and database views.
Evidence timers	Scheduled readiness, security, backup and recovery checks	Repeatable acceptance and operational assurance records.

Customer integration option
Where required, approved metrics, alerts or audit outputs can be integrated with the customer monitoring, SIEM and service-management environment.

Backup and recovery mechanics

Recovery scope	Mechanism	Design decision
PostgreSQL	Consistent database backup plus non-destructive restore verification	Frequency, encryption, retention and target RPO/RTO.
MinIO / objects	Object backup with integrity and required-artifact checks	Off-host destination, capacity and retention.
Kubernetes configuration	Versioned manifests, release inventory and protected configuration backup	Recovery source and rebuild procedure.
Secrets and trust	Protected secret/trust backup under customer key and access policy	Separation of duties and recovery authorization.
Applications and packs	Versioned release/package artifacts and lifecycle metadata	Which versions must remain recoverable.
Workspaces	Customer-approved persistent workspace backup policy	Scope, retention and exclusion of temporary sandboxes.
Replacement host	Reinstall platform -> restore configuration/data -> validate services -> acceptance checks	Hardware compatibility, DNS/PKI changes and recovery test frequency.

Recovery sequence



The customer solution design defines whether the deployment uses local recovery only, encrypted off-host backup, secondary-site recovery, or a higher-availability architecture.

Network zones and communication paths

Zone	Allowed communication pattern
User access zone	HTTPS to the approved Nexus ingress origin. No direct access to databases, queues or object storage.
Identity zone	OIDC/SAML/enterprise directory communication through approved identity endpoints and trust.
Platform service zone	Mutual in-cluster service communication according to namespaces, service accounts and network policy.
Data zone	Only approved platform services reach PostgreSQL, RabbitMQ, MinIO and persistent storage.
Inference zone	Core/model gateway reaches approved local or remote inference endpoints over defined ports and trust.
Integration zone	Connector workers reach allowlisted customer or provider endpoints using server-side credentials.
Operations zone	Monitoring, backup, support and administrative access restricted to approved operator roles and networks.
Offline update path	Signed release/model/pack media imported through the controlled verification workflow.

Ports and firewall rules
Exact DNS names, certificates, source/destination subnets, ports, proxies and egress rules are documented in the customer low-level design and acceptance plan.

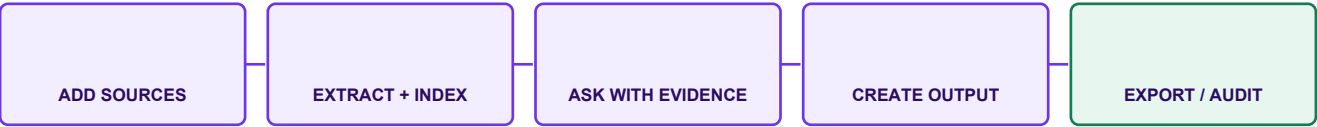
Enterprise capability map

Domain	Capabilities	Business outcome
Work with AI	AI Workspace, Prompt Studio, Prompt Library, model comparison and evaluation	Consistent, governed access to approved AI for individual and team work.
Know	Nexus Notebook, citations, document ingestion, compare, digitization and datasets	Turn enterprise content into searchable evidence, answers and structured outputs.
Automate	Agent Studio, Workflow Designer, schedules, runs, meetings and Chat Rooms	Coordinate repeatable work while retaining permissions, history and human control.
Build	App Studio, applications, templates, Jupyter, SDKs and APIs	Create internal AI applications within governed development and release boundaries.
Govern	Identity, roles, reviews, audit, credentials, guardrails, models and quotas	Apply one policy model across users, models, data, applications and integrations.
Operate	Health, metrics, logs, GPU monitoring, packs, backups and recovery evidence	Operate AI as managed infrastructure with visible service and lifecycle state.

Capability availability depends on the licensed edition, purchased packs, approved integrations and customer infrastructure.

Enterprise knowledge and document intelligence

Nexus Notebook provides a controlled workspace for collections, sources, conversations and generated outputs. Collections are private by default and can be shared through the customer access model.



Capability	Customer use
Evidence-grounded Q&A;	Ask questions against approved source collections and receive citations back to supporting material.
Document ingestion	Process common enterprise text, PDF and Office formats into governed searchable content.
Document Compare	Identify meaningful differences between versions and preserve an exportable comparison record.
Document Digitization	Convert scans and PDFs into reviewable structured records before publication to enterprise knowledge.
Dataset creation	Create, review and publish grounded examples for evaluation or approved model-training workflows.
Knowledge outputs	Produce summaries, requirement lists, dates, actions and other reusable outputs from governed evidence.

Human-verifiable by design
Source access, extraction quality and model selection affect results. Customer acceptance should include representative documents and explicit quality criteria.

Automation and application delivery

Platform area	Lifecycle	Control model
Agents	Define -> test -> approve -> run -> review	Approved tools, bounded permissions, durable execution history and human checkpoints.
Workflows	Compose -> validate -> schedule -> monitor	Ordered AI, retrieval, transformation and integration steps with visible failures.
App Studio	Requirement -> proposal -> sandbox -> preview -> review -> publish	Isolated Kubernetes execution, independent approval, versioning and rollback.
Applications	Publish -> operate -> update -> rollback/retire	Application gateway, controlled access, lifecycle evidence and operational visibility.
Capability packs	Import -> verify -> accept -> enable -> update/rollback	Signed offline packages, policy validation, quarantine and removal controls.

Developer and integration surfaces

- JupyterHub workspaces for controlled analysis and prototyping.
- Versioned Python and TypeScript SDKs.
- Governed APIs and managed programmatic credentials.
- Reusable templates and offline extension conformance assets.
- Connector and MCP capability projection with server-side secrets and bounded scopes.

Security, identity and governance

Nexus One AI is designed to integrate with the customer security model rather than create a separate unmanaged AI island.

Control area	Platform design
Identity and access	Central identity integration, role-based authorization, team boundaries and reviewer/auditor roles.
Data boundary	Customer-controlled storage and processing; external connectivity is defined by the approved deployment profile.
Secrets	Connector and service credentials remain server-side and are referenced rather than exposed to browser clients.
Audit and review	Administrative and AI activity evidence, assigned reviews and approval boundaries.
Application isolation	Generated applications and extensions execute in bounded Kubernetes namespaces and service accounts.
Supply chain	Digest-pinned runtime images and signed release/pack inventories with fail-closed validation.
Guardrails	Managed API credentials, model/inference policy and customer-specific data/security controls.
Change control	Versioned applications, controlled publication, rollback and signed update paths.

Customer responsibility
Final production design must align with customer PKI, directory services, network segmentation, data classification, retention, DLP, incident response and compliance requirements.

Operations, observability and resilience

Operational area	Included platform capability
Service health	Platform, dependency, application and job status with truthful degraded states.
Infrastructure telemetry	CPU, memory, storage, Kubernetes, database and supported GPU metrics.
Dashboards and alerts	Prometheus, Alertmanager and Grafana operational views.
Central logs	Alloy and Loki collection for platform and workload troubleshooting.
Audit evidence	Administrative actions, governed lifecycle decisions and support evidence.
Backup	Scheduled platform backup with freshness, retention and storage checks.
Recovery verification	Non-destructive restore verification and recoverability evidence.
Supportability	Support-bundle generation, redaction controls and documented operational response.

Production service levels, high availability, recovery point and recovery time objectives are defined for the selected deployment architecture and customer environment.

Deployment and integration architecture

Deployment element	Design options
Infrastructure	Customer appliance, virtual or bare-metal server, GPU-backed server, or custom multi-GPU design.
Network	Connected enterprise network, restricted egress, disconnected/air-gapped operating pattern.
Identity	Customer identity and role model; site-specific directory and SSO integration.
Inference	Approved local model runtime or compatible customer-selected inference provider.
Data services	Platform-managed PostgreSQL/vector search, object storage and durable job transport.
Integrations	REST/API, databases, file/content systems, business applications and governed MCP connectors.
Updates	Controlled online or signed offline release and capability-pack process.
Recovery	Customer-approved local and off-host backup destinations with documented restore procedure.

Integration lifecycle



No implicit external access
Customer integrations are provisioned explicitly with approved credentials, scopes, hosts, retention and data-boundary terms.

Deployment profiles and sizing inputs

Profile	Typical deployment intent	Sizing focus
Nexus One AI Workstation	Individual specialist, developer, researcher or compact local deployment.	Selected local models, memory capacity, storage and single-user responsiveness.
Server S	Contained team, departmental introduction or focused use case.	Concurrency, document volume, model size and growth headroom.
Server M	Sustained departmental operations and broader automation.	Higher utilization, job throughput, retention and integration load.
Server L	Multi-team use, larger knowledge workloads and advanced governance.	Multi-GPU options, isolation, service continuity and operational capacity.
Server Max	Custom enterprise, specialized or multi-node programme.	Workload separation, resilience, storage, networking and enterprise service levels.

Sizing is workload-led

- Selected model and quantization.
- Context length and expected response time.
- Active and concurrent users.
- Document count, size, ingestion rate and retention.
- Agent/workflow frequency and integration latency.
- GPU/VRAM, CPU, memory, storage and network performance.
- Availability, backup, recovery and growth requirements.

Representative enterprise solutions

Business area	Example solution pattern	Platform capabilities
Procurement	Tender and contract intelligence	Notebook, document compare, prompts, workflows and approval.
Legal and compliance	Policy and obligation assistant	Cited knowledge, document comparison, audit and controlled outputs.
Operations	Daily operations brief and exception handling	Scheduled workflows, agents, integrations and governed reporting.
Service management	Internal support knowledge assistant	Notebook, AI Workspace, team access and feedback/evaluation.
Finance	Reconciliation and anomaly-review assistant	Document/data ingestion, workflows, human review and audit.
HR	Employee policy and onboarding workspace	Private collections, controlled access, applications and automation.
Engineering	Technical knowledge and drawing/document search	Multi-format ingestion, retrieval, citations and developer tools.
Management	Meeting, decision and reporting assistant	Meeting pipeline, prompts, schedules and evidence-backed outputs.

Each customer solution is scoped against the available data, approval model, security classification and measurable acceptance criteria.

Customer evaluation checklist

Evaluation area	Questions to confirm
Business scope	Which decisions, documents, workflows and users are in the first release?
Data	What sources, classifications, access rules, retention and quality constraints apply?
Models	Which models are approved, and what quality, latency and safety criteria apply?
Identity	Which directory, SSO, MFA, roles and approval responsibilities are required?
Infrastructure	What compute, GPU, storage, network and availability profile is approved?
Security	What PKI, segmentation, secrets, DLP, audit and incident-response controls are required?
Integrations	Which systems, credentials, scopes and external data boundaries are approved?
Operations	Who owns monitoring, backup, restoration, capacity, updates and support escalation?
Acceptance	What functional, quality, security, performance and recovery evidence is required?

Recommended evaluation output

A signed solution design containing the deployment diagram, responsibility matrix, capability scope, model/data profile, sizing assumptions, security controls and acceptance plan.

Implementation approach



Phase	Primary output
Discover	Prioritized use cases, users, outcomes and exclusions.
Assess	Infrastructure, identity, security, data and integration readiness.
Design	Approved reference architecture, sizing, responsibilities and implementation plan.
Deploy	Installed and configured platform in the customer environment.
Accept	Functional, quality, security, performance and recovery evidence.
Adopt	Training, operating procedures, support handover and initial workload onboarding.
Expand	Controlled addition of data, models, integrations, applications and teams.

Customer-ready next step

Begin with a solution-design workshop. Cezen and the customer jointly define the first workload, deployment boundary, infrastructure profile, identity model, data sources, security controls and acceptance criteria.

Important commercial and technical note
Specific capability availability depends on the licensed edition, purchased packs, customer infrastructure and approved integration scope. Final architecture, sizing, security configuration and service levels are confirmed during solution design.