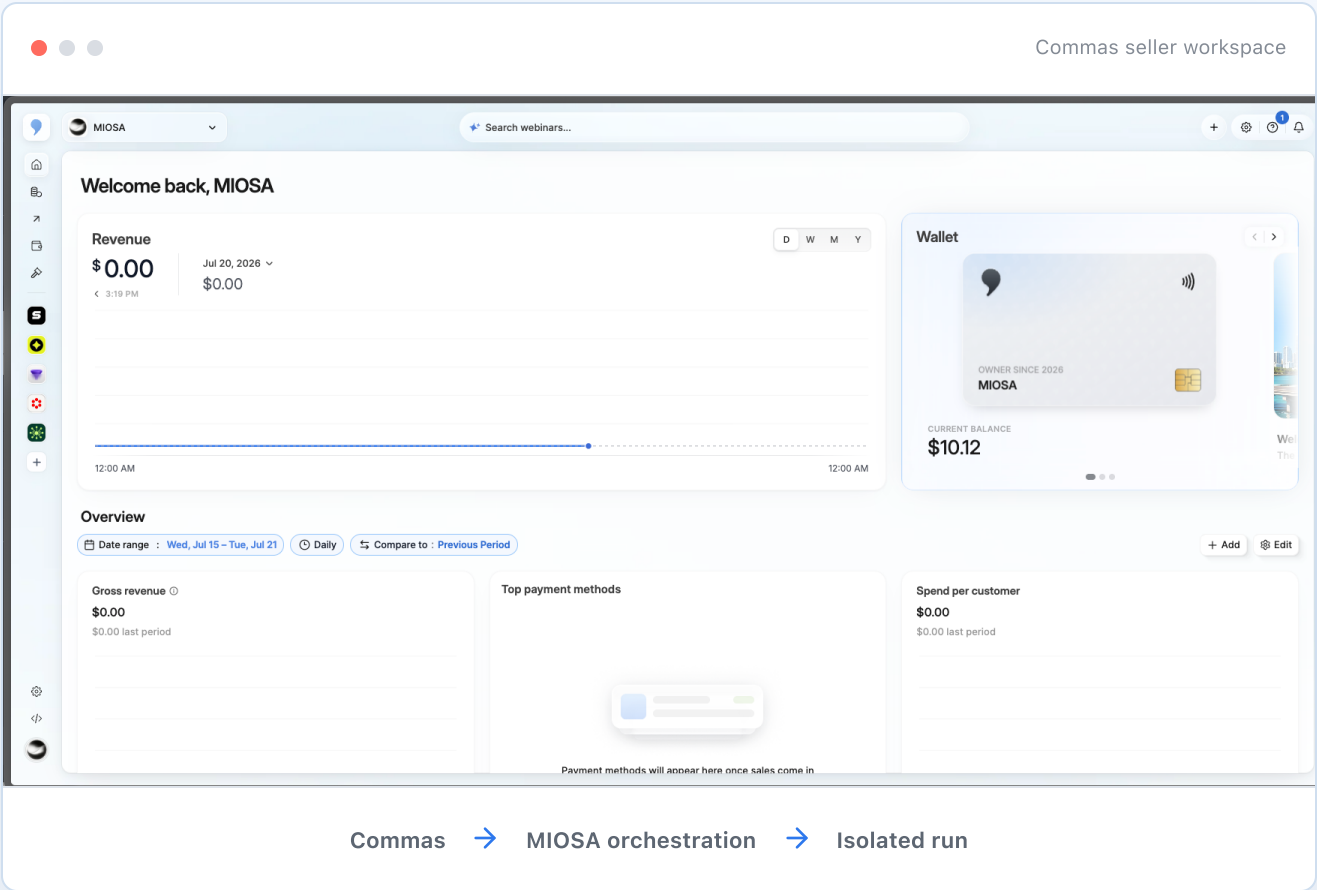


Commas + MIOSA Agent Infrastructure Pilot

A structured proposal for integrating governed agents, organization-scoped context, configurable models, customer-owned keys, and short-lived sandboxes into the Commas product experience.

Status	Working proposal
Audience	Product, engineering, security, leadership
Decision	Technical discovery before scope



What the meeting established

The opportunity is not a generic chatbot. It is a governed execution layer behind the product experience Commas already owns.



Activation means first dollar

The strongest small-business journey removes setup friction and gets a seller to a real, reviewable, monetizable asset.



Isolation is a product primitive

The team identified per-user, short-lived environments as the missing foundation for safe agent execution and cleanup.



The same runtime can serve two markets

Small-business users need a guided path. Enterprise teams need policy, integrations, control, and deployment choice.

NOW

Validate one narrow journey

Do not start by rebuilding the entire agent stack.

NEXT

Measure the controlled pilot

Quality, speed, reliability, cost, and adoption.

THEN

Choose the scale architecture

Managed, Commas-controlled cloud, or hybrid.

One foundation, two deliberately different experiences

The interface and level of control change by customer. The core isolation, orchestration, and evidence model remains consistent.



SMALL BUSINESS

Guided path to first dollar

A constrained agent asks only what is needed, creates the commercial journey, and returns a preview for approval.

- 01 Describe the offer and buyer
- 02 Generate funnel and checkout assets
- 03 Review a live preview
- 04 Publish through Commas

✓ Pilot this lane first



ENTERPRISE

Embedded operating layer

Teams configure agents, data access, policies, providers, and deployment boundaries without leaving the Commas product.

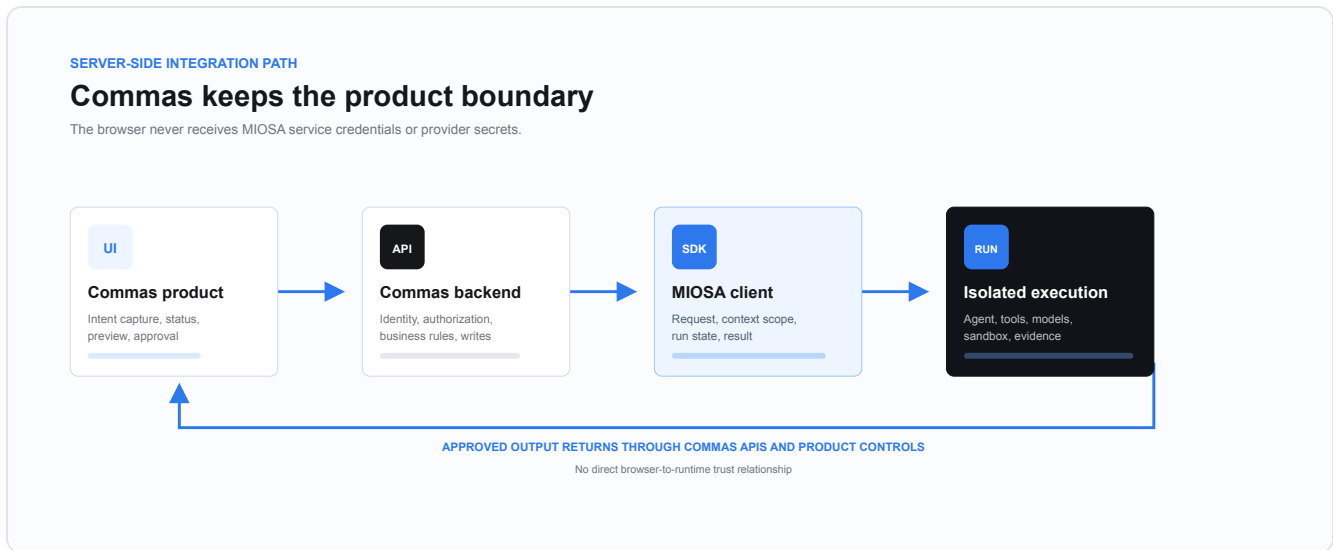
- 01 Connect approved systems and data
- 02 Assign organization and workspace scope
- 03 Run agents under explicit policy
- 04 Audit decisions and actions



Design after pilot evidence

The next conversation starts with the actual Commas stack

MIOSA supports language-specific integration clients and can add adapters where needed. The first technical task is mapping the real codebase, not guessing from a demo.



01



Inspect the Commas codebase

Identify backend language, deployment model, authentication boundary, API clients, and the current funnel agent implementation.

02



Install the appropriate SDK client

Use the supported client for the Commas backend. Add a custom adapter only where the existing application boundary requires one.

03



Map identity and tenancy

Translate Commas customers, teams, users, and projects into explicit organization and workspace scopes before any agent receives context.

04



Define the agent contract

Set instructions, tools, source access, output schema, model parameters, budgets, review rules, and publishing permissions.

05



Broker secrets and launch isolation

Pass approved credentials through a secret boundary. Create a short-lived sandbox with explicit image, network, time, and resource policies.

06



Review, publish, and close

Return a preview to Commas, require the configured approval, write through Commas APIs, retain evidence, and destroy temporary runtime state.

</> ILLUSTRATIVE SERVER-SIDE REQUEST

```
const run = await miosa.runs.create({
  organization: commasCustomerId,
  workspace: sellerWorkspaceId,
  agent: "first-dollar-builder",
  input: sellerIntent,
  review: "required"
});
```

Exact syntax and language are confirmed against the current SDK and Commas backend during discovery.

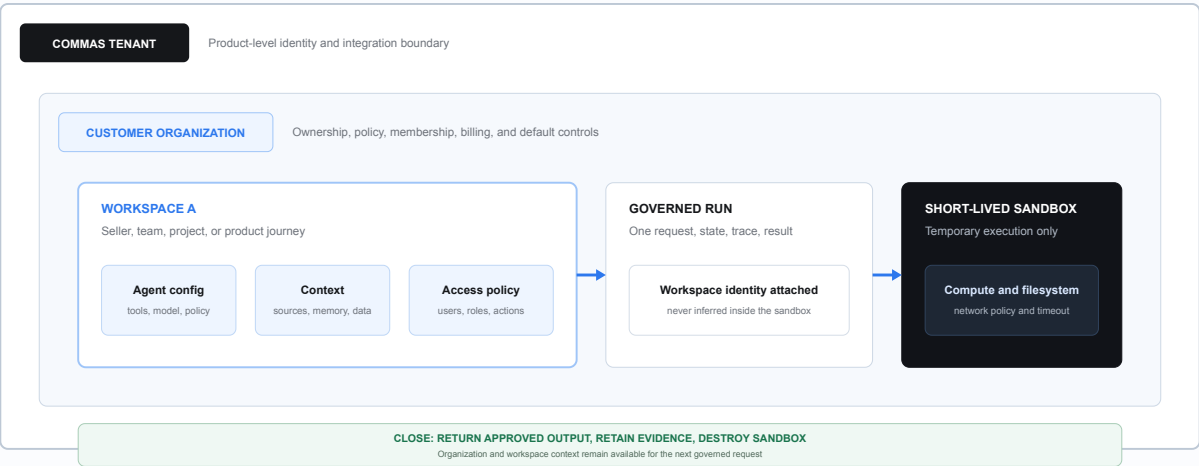
Persistent business context and temporary compute are different things

A workspace is the governed context boundary. A sandbox is the temporary execution environment created for a run. Confusing them creates data leakage and lifecycle ambiguity.

TENANCY AND EXECUTION MODEL

Persistent scopes surround temporary work

Every request carries organization and workspace identity through orchestration, execution, review, and evidence.

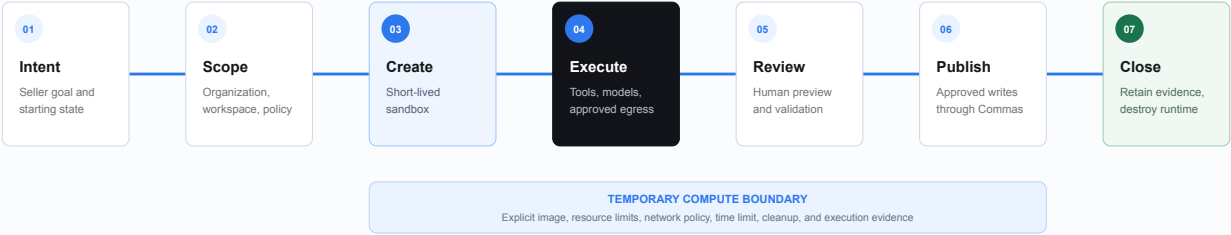


Organization Ownership and policy boundary for a Commas customer, business, or enterprise account.	Workspace Scoped context for a seller, team, product journey, client, or project.	Agent Configured role with instructions, tools, context access, model policy, and output contract.
Run One governed request with an identity, state, trace, cost, result, and review status.	Sandbox Temporary compute, filesystem, process, and network boundary used by the run.	Evidence Durable record of what ran, what it accessed, what it returned, and whether cleanup succeeded.

ONE GOVERNED RUN

From seller intent to approved Commas output

Workspace context persists. Runtime state is temporary. Publishing always returns through Commas controls.



Commas controls the experience. Each journey controls its agent configuration.

The pilot should expose the minimum configuration needed by the team while keeping secrets, infrastructure controls, and provider access behind appropriate boundaries.

Control	Examples	Scope
Agent identity	Name, purpose, instructions, owner	Product journey or workspace
Context	Approved sources, memory, user inputs	Organization and workspace
Tools	Commas APIs, providers, internal services	Allowlist per agent
Model	Provider, model, temperature, limits	Agent or task type
Credentials	Commas-owned or customer-owned keys	Secret broker, never prompt text
Execution	Image, CPU, memory, time, egress	Run and sandbox policy
Output	Schema, preview, approval, write target	Product action
Evidence	Trace, cost, result, cleanup status	Retained execution record



BRING YOUR OWN KEYS

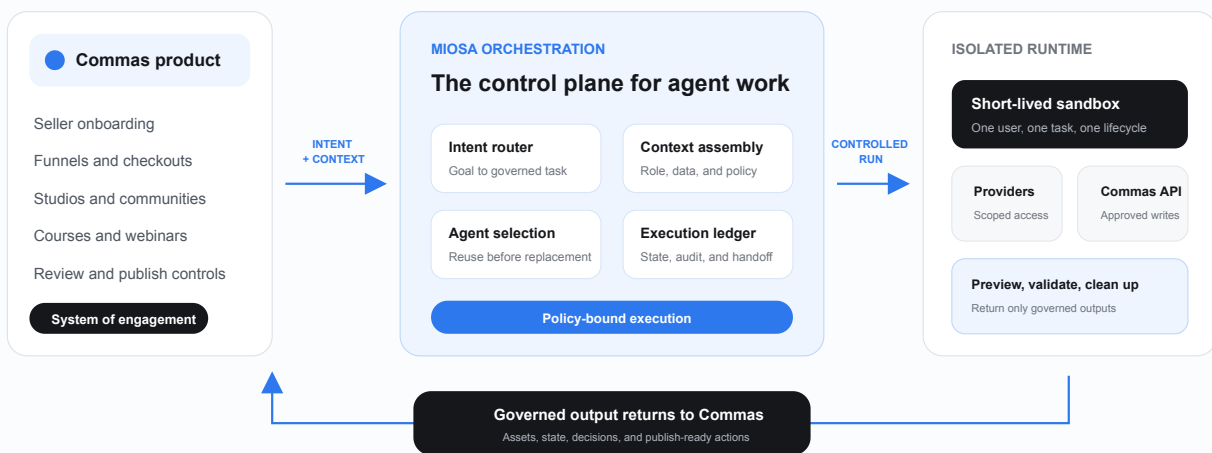
Provider choice without exposing credentials

Commas or its customers can supply supported provider keys. Credentials are resolved at execution time and never embedded in prompts, repositories, or generated assets.

- ✓ Keys scoped by organization or workspace
- ✓ Provider and model chosen by policy
- ✓ Budgets and limits enforced per run

REFERENCE ARCHITECTURE

Commas controls the experience. MIOSA controls the work boundary.



Working model for technical discovery. Exact controls, ownership, and guarantees require joint validation.



Security claims require joint validation.

The pilot must produce a reviewed threat model, ownership matrix, data flow, retention policy, and testable acceptance criteria before production.

Start with 3-5 selected users and one measurable journey

Roberto works directly with product and engineering to map the codebase, configure the integration, instrument the pilot, and transfer the operating model.

PHASE 1

Technical discovery

Codebase, SDK, agent code, identity, APIs, data, security, deployment, baseline.

Output: decision record

PHASE 2

Controlled pilot

One first-dollar journey, 3-5 users, isolated runs, review, metrics, support.

Output: measured integration

PHASE 3

Scale decision

Assess quality, economics, reliability, placement, ownership, and enterprise path.

Output: go, revise, or stop



Codebase and SDK mapping

Select the integration boundary and implement required adapters.



Tenancy design

Map users, teams, organizations, workspaces, agents, and data access.



Agent configuration

Instructions, tools, models, parameters, keys, budgets, and outputs.



Controls and transfer

Isolation tests, instrumentation, documentation, and team enablement.

Selected user reaches a live, reviewable asset

Minutes from intent capture to usable preview

Human changes required before publishing

Reliability

Completion, retry, failure, and cleanup rates

Economics

Model and compute cost per completed journey

Adoption

Users who complete the journey without intervention

PILOT OWNERSHIP

Clear ownership prevents hidden assumptions

The technical discovery converts each item below into an owner, artifact, and acceptance criterion.



Commas owns

Product and customer boundary

- **Seller experience**
Onboarding, review, publishing, support
- **Product logic and APIs**
Business rules, approved writes, UI state
- **User consent and permissions**
Who can request, review, and publish
- **Success definition**
Journey baseline and desired outcome



MIOSA owns

Proposed execution boundary

- **SDK and orchestration**
Request, state, routing, and handoff
- **Workspace context**
Scoped agents, sources, policy, memory
- **Sandbox lifecycle**
Create, constrain, observe, destroy
- **Execution evidence**
Trace, result, cost, cleanup status



Joint decisions

Validated before production

- **Threat model**
Data, secrets, egress, retention
- **Deployment placement**
Managed, BYOC, or hybrid
- **Acceptance criteria**
Quality, reliability, economics
- **Commercial scope**
Timeline, support, ownership, price

Technical discovery and pilot selection

Reply to the existing email thread, confirm the time, and include the product and engineering owners who can answer implementation questions.

10 min	Choose the first-dollar journey	User, starting state, publish event, and successful result.
15 min	Inspect the current code and SDK boundary	Backend language, agent code, APIs, test tenant, and shared sandbox.
15 min	Map identity, data, and isolation	Organizations, workspaces, credentials, network, retention, audit, cleanup.
10 min	Select pilot placement	MIOSA managed, Commas-controlled cloud, or a limited hybrid.
10 min	Lock owners and acceptance criteria	Scope, success metrics, exclusions, timeline, and next commercial step.

BRING TO THE SESSION

Required inputs

- ✓ API documentation and test credentials
- ✓ Non-production seller tenant
- ✓ Existing funnel agent code
- ✓ Current shared-sandbox workflow
- ✓ First-dollar baseline and failure points
- ✓ Product, engineering, and security owners

Reply in the email thread



This is a non-binding working proposal for technical discovery. Exact scope, controls, guarantees, ownership, timeline, and commercial terms require joint validation.