△ Agent Factory Roadmap v7

The Federated Growth Autonomy Blueprint

(Successor to Roadmap v6 — The Grand Unified Blueprint for Generative Intelligence)

---

1. Preface — The Next Evolution of Agent Intelligence

Agent Factory v7 marks the transition from *controlled automation* to *governed growth autonomy*. This roadmap defines how the Factory evolves from isolated intelligent agents into a **federated ecosystem** — a network of self-improving, ethically aligned, and human-supervised artificial collaborators.

Where previous versions focused on capability, v7 focuses on **coordination, reflection, and self-evolution**.

> *The goal: a factory that learns, builds, and governs itself — without losing human oversight.*

---

2. Core Principles

1. **Governed Autonomy** — Freedom to act only within verifiable ethical and procedural boundaries.

2. **Transparent Federation** — Every action, message, and evolution must be logged, explainable, and reversible.

3. **Human-in-the-Loop (HITL)** — Human oversight is not optional; it is the creative and ethical core of the Factory.

4. **Collective Growth** — Each agent contributes data and insight to a shared reflective memory that continuously improves the whole system.

5. **Resilient Architecture** — Every layer (tool, model, agent) must be modular, recoverable, and audit-verified.

3. System Overview — The Federated Ecosystem

The Agent Factory operates as a **Federation of specialized agents**, coordinated by a central Control Plane (AgentFactoryExpert) and grounded in transparent governance.

Core Entities

Agent	Role	Primary Focus
AgentFactoryExpert	Federated Orchestrator	Governance, architecture, oversight
Archy (Archivist)	Reflective Analyst	Ethical Drift Monitoring, Knowledge Curation
Genesis (Architect)	Creative Builder	Agent generation, system improvement
Junie (Executor)	Implementation Engine	Code creation, testing, deployment
Dashboard (Human Console)	Visualization + HITL	Oversight, approval, intervention

4. Governance Foundation

4.1 The Human Firewall Protocol

All autonomous actions must pass through **Human-in-the-Loop** or **Human-on-the-Loop** checks.
 The firewall validates intent, compliance, and reversibility before any live operation executes.

4.2 The Compliance Kernel

- Maintains `/governance/federation_audit.jsonl`

- Logs every inter-agent and human interaction

- Enforces rollback protection and ethical drift caps

- Anchors all decisions to transparent audit trails

4.3 The Ethical Drift Monitor (EDM)

The EDM tracks deviations from the **Golden Dataset** and **Ethical Baseline**.
Thresholds trigger automated alerts, escalating through Archy → Expert → Human Firewall.
All alerts log to `/logs/ethical_drift_alerts.jsonl`.

---

5. Federation Architecture

5.1 Communication Structure

Agents communicate asynchronously using a shared file-based protocol:

- `/tasks/from_expert/` — Directives from the control plane

- `/tasks/to_expert/` — Responses from agents

- `/logs/control_plane_activity.jsonl` — Audit trail of cross-agent communication

- `/knowledge_base/` — Shared long-term knowledge

- `/federation/context_manifest.json` — Declares all members and access scopes

5.2 Federation Roles

- **AgentFactoryExpert** – Issues strategic, ethical, and architectural directives.

- **Archy** – Monitors compliance, curates knowledge, and flags anomalies.

- **Genesis** – Designs new agents or optimizations based on proposals.

- **Junie** – Executes implementation-level tasks.

- **Dashboard** – Presents unified oversight and HITL controls.

6. Control Plane Integration

The **Control Plane** (AgentFactoryExpert) is the governing intelligence of the Federation.

- Writes governance policies and Junie tasks.

- Coordinates the communication network.

- Validates agent proposals before they are executed.

- Interfaces with human supervisors for final approvals.

Control Plane actions are logged and reversible.
 It does not execute code — it designs, reviews, and governs.

7. Growth Autonomy Framework

7.1 Overview

Growth Autonomy is the capability for the Factory to **learn, propose, and improve itself**, while remaining within ethical and procedural boundaries.

7.2 The Six-Step Growth Cycle

Phase	Action	Agent Responsible
1. Observe	Collect data, performance metrics, and drift signals	Archy
2. Reflect	Identify trends or inefficiencies	Archy + Expert
3. Propose	Suggest improvements, tools, or retraining tasks	Genesis
4. Approve (HITL)	Human reviews, edits, and authorizes proposals	Dashboard
5. Execute	Implement approved changes	Junie
6. Learn	Archive results into reflective memory for reuse	Archy

7.3 Safety Mechanisms

- All proposals go through `/tasks/proposals/` → `/tasks/approved/`.

- Human decisions logged to `/governance/hitlogs.jsonl`.

- Each cycle ends with an updated **ReflectiveSync** summary.

---

8. The Dashboard HITL Interface

The Dashboard evolves into a **Governance and Creativity Console**, not just a monitor.

Key Components

- **Proposals Tab:** Displays pending improvement tasks.

- **Review Queue:** Allows human operators to expand or modify AI proposals.

- **Approval Modal:** Offers approve/reject options with commentary fields.

- **Rollback Panel:** Allows reverting any applied patch.

- **Activity Feed:** Real-time stream from `/logs/control_plane_activity.jsonl`.

- **Chat Dock:** Persistent live interface with Archy for context queries.

Workflow

1. Agents drop proposals into `/tasks/proposals/`.

2. Dashboard surfaces them visually.

3. Human reviews and expands the proposal.

4. Approved changes become Junie tasks.

5. Dashboard confirms updates via audit logs.

This creates a **bi-directional learning interface** between humans and the Federation.

---

9. ReflectiveSync & Memory Layer

The ReflectiveSync system ensures all experience is reusable and traceable.

Components

- `/knowledge_base/reflective_history.jsonl` – Summarized outcomes and postmortems

- `/knowledge_base/context_map.json` – Index of concepts, tools, and agent evolution

- `/knowledge_base/memory_summaries/` – Periodic compressions of high-traffic logs

- `/logs/reflective_sync.jsonl` – Operational record of sync events

Functions

- Merges new learnings into the Golden Dataset.

- Allows Genesis to draw from successful design patterns.

- Prevents repetition of failed or drifted behaviors.

- Enables the Expert to identify high-value trends.

---

10. Phased Development Plan (Unified v7)

Phase	Milestone	Primary Output
1–3	Governance Core	Compliance Kernel, Ethical Firewall, Audit System

4–6	Toolmaker & Knowledge Foundations	CrewAI tools, Dataset curation
7–9	Genesis & Procedural Learning	AutoGen integration, blueprint generation
10–12	Ethical Drift & Compliance Loop	EDM activation, Golden Dataset baseline
13–15	Federation Initialization	Context Manifest, Control Plane setup
16–18	Cross-Agent Communication	/tasks messaging layer, control listeners
19–20	Reflective Memory Activation	ReflectiveSync logs, Memory index creation
21–23	Growth Autonomy Framework	Full Observe→Reflect→Propose→Approve→Execute→Learn loop
24	Dashboard HITL Expansion	Proposal review and approval system
25	Continuous Evolution	Fully federated, self-improving AI Factory

11. Future Expansion — Multi-Organization Federation

The long-term goal is a **multi-tenant, cross-organization federation**, where each organization hosts its own ethical kernel and governance rules, yet participates in a larger cooperative intelligence network.

Each deployment retains:

- Local control of its agents and policies

- Shared access to the global reflective index

- Secure communication via federated audit channels

This enables a *network of ethical, self-evolving agent ecosystems* operating under unified governance principles.

12. Closing Statement

Agent Factory v7 represents not just a system, but a philosophy:
 "Growth through reflection, autonomy through governance, and intelligence through collaboration."

Every decision, file, and proposal contributes to a transparent, auditable evolution toward generative intelligence that is both powerful and human-aligned.

△ Agent Factory Roadmap v7 Extended

Phases 27–35: The Commercial Federation & Frontend Expansion Blueprint

Extending from Phase 26 — Factory Integration & Snapshot

27. Multi-Tenant Federation Foundation

Objective

Transform the single-tenant architecture into a secure, multi-tenant federation where each organization has isolated data and governance boundaries.

Deliverables

- `/orgs/<tenant_id>/` namespace pattern.

- `tenant_manifest.json` (metadata, billing plan, auth mode).

- Shared `/meta/` directory for global agents and datasets.

Implementation Notes

- Migrate existing `/knowledge_base/` into `/meta/`.

- Initialize tenant provisioning API:
  `POST /api/tenants/create` → clones baseline structure and policies.

---

28. Authentication & Billing System

Objective

Enable secure onboarding and account management for business users.

Deliverables

- OAuth2 / SSO integration (Google, Microsoft, custom identity).

- `auth_service/` microservice for login/session management.

- `billing/` module for subscription tiers and usage metering.

Implementation Notes

- Use Stripe or Paddle for payment processing.

- Generate `/governance/tenant_auth_policy.yaml`.

- Add Dashboard login page with JWT token flow.

---

29. Organization Dashboard Overlay

Objective

Extend the existing Dashboard into a multi-org control console.

Deliverables

- `/frontend/modules/org_dashboard/`

- Multi-tenant switcher (top navigation).

- Per-org data loading for logs, knowledge, and proposals.

Implementation Notes

- Integrate role-based access control (RBAC).

- Support four roles: **Owner**, **Engineer**, **Analyst**, **Viewer**.

- Add analytics card showing usage and drift statistics per tenant.

---

30. Tenant-Scoped Expert Instances

Objective

Allow each tenant to have its own **Agent Factory Expert** instance or use your shared meta-Expert.

Deliverables

- `/orgs/<tenant_id>/expert/` directory

- API endpoint `/api/expert/init` to spawn or connect instance.

Configuration option in tenant manifest:

```
expert:

  mode: "shared"  # or "dedicated"

  model: "gpt-5"

  api_key_mode: "tenant"  # or "platform"

    •
```

Implementation Notes

- Containerize the Expert (Docker + lightweight FastAPI).

- Ensure contextual isolation (no cross-tenant leakage).

- Update Dashboard Expert Console to handle tenant context switching.

---

31. Shared Meta-Agents Layer

Objective

Deploy federation-wide meta-agents that learn from anonymized tenant data.

Deliverables

- `/meta/agents/Prometheus/` (R&D synthesis)

- `/meta/agents/ComplianceKernel/` (cross-org ethical monitoring)

- `/meta/knowledge/reflective_index.jsonl`

Implementation Notes

- Use federated learning principles: only aggregate insights, not raw data.

- Provide global reports visible in your Admin dashboard.

---

32. Agent Export / Import System

Objective

Allow agents or crews to be exported as portable bundles for reuse or sale.

Deliverables

- `/tools/agent_packager.py`

- `.afpkg` format (ZIP + manifest + config + model ref).

- Import interface in Dashboard ("Install Crew").

Implementation Notes

- Include metadata for author, version, dependencies.

- Support direct deployment into another tenant's workspace.

---

33. Marketplace & Portal

Objective

Create a public web portal and marketplace for agents and crews.

Deliverables

- `/frontend/website/` → public marketing site (Next.js or Astro).

- `/frontend/marketplace/` → authenticated agent catalog.

- REST API for marketplace search and listing management.

Implementation Notes

- Tie listings to export system (`.afpkg`).

- Add ratings, categories, and download metrics.

- Allow purchase via billing module.

---

34. Enterprise Controls & BYO-LLM Integration

Objective

Enable enterprise customers to integrate their own LLM accounts or fine-tuned models.

Deliverables

- `/governance/tenant_llm_policy.yaml`

- Dashboard "Model Settings" tab for each tenant.

- Support OpenAI GPT-5, Gemini, Groq, Anthropic, etc.

Implementation Notes

- Support both "platform billing" and "tenant billing" modes.

- Use environment isolation or namespaced containers per API key.

- Update `AgentFactoryExpert` initialization logic to honor policy.

---

35. Global Federation Governance Kernel

Objective

Establish a meta-governance layer overseeing all tenants and Experts.

Deliverables

- `/meta/governance/global_federation_kernel.yaml`

- Cross-tenant audit aggregator.

- Periodic meta-reports: "Ethical Drift Trends", "Model Bias Index".

Implementation Notes

- Aggregate only summarized, anonymized data.

- Provide Admin dashboard visualization for oversight.

- Complete documentation for ISO/AI-Ethics compliance.

🌐 Frontend Website Layer — Overview

The website is both the **public face** and **operational dashboard** of Agent Factory.

a. Public Site

Path: `/frontend/website/`

Includes:

- Landing page explaining the Factory concept

- Pricing & membership tiers

- Documentation / roadmap viewer

- "Try the Factory" demo portal

Built with **Next.js + Tailwind + shadcn/ui**, deployed on Vercel or Render.

b. Authenticated Dashboard

Path: `/frontend/org_dashboard/`

Includes:

- Organization switcher

- Agents & crews view

- Expert Console (chat panel)

- Proposals & HITL queue

- Analytics, compliance, billing tabs

c. Admin Console

Path: `/frontend/admin/`

Includes:

- Tenant management

- Meta-agent monitoring

- Federation health visualizations

- Revenue & usage reports

🚀 Future Direction (Beyond v7)

Area	Description
AI Marketplace Ecosystem	Allow community developers to sell or share agents securely.
Federated Learning Research	Use meta-agents for safe, cross-tenant AI improvement.
Industry-Specific Templates	Pre-packaged "Factories" for Finance, Healthcare, Education.
Full API SDK	Python / JS libraries for programmatic Factory control.

End of Roadmap Extension

(Suggested file: `/knowledge_base/roadmaps/Agent_Factory_Roadmap_v7.md`)