Clonehaus Governance Model — Top-Down Editable Granularity

Core Governance Philosophy

Clonehaus follows a **strict top-down, locally tunable governance model**. Authority, policy, and behavior are defined at the highest appropriate level and may only be **narrowed**, never expanded, as they flow downward.

The system is designed so users never have to guess *where* a decision belongs.

---

1. Organization OS (Constitutional Layer)

Purpose
Defines the non-negotiable foundation of how AI operates inside the organization.

What it sets - Maximum authority ceiling - Allowed categories of action - Global escalation posture - Ethics-Aligned Persona Protocol (EAPP)

Key properties - Applies to *everything below it* - Can be **locked** - Cannot be overridden by domains, agents, approvals, or policies

Current implementation status - The Organization OS exists as a constitutional surface - It defines boundaries only - **No organization-level customization inputs have been enabled yet** - This is intentional to avoid premature complexity

Once locked: - ❌The constitution cannot be changed - ✅Domains and agents may still be configured *within* its limits

---

2. Domains (Operational Policy Layer)

Purpose
Translate organizational rules into practical, team-level policy.

What domains can do - Narrow authority ceilings - Restrict action categories - Tighten escalation requirements

What domains can never do - Exceed Organization OS authority - Enable forbidden actions - Override ethical commitments

Key design principle
Domains are *always editable*, even when the Organization OS is locked.

This is where most real-world governance work happens.

---

3. Agents (Persona & Execution Layer)

Purpose
Define how individual AI teammates behave within their domain.

What agents can do - Fine-tune autonomy - Define persona identity (LPS) - Adjust execution posture

What agents can never do - Exceed domain authority - Escape ethical commitments - Self-authorize actions

This layer enables deep customization **without risk**.

---

4. Runtime (Enforcement Layer)

Purpose
Enforce everything that has been defined — without creating anything new.

What runtime does - Enforces OS → Domain → Agent inheritance - Applies EAPP vetoes - Evaluates approvals and readiness

What runtime never does - Invent rules - Modify policy - Execute actions autonomously

Runtime answers the question:

> "Can this happen?"

Before anything actually happens.

Why This Model Works

This structure provides: - Granularity without chaos - Local control without global risk - Customization without silent authority creep - A clear mental model for users

Users naturally think in this order: - "Is this a company rule?" → **Organization OS** - "Is this how this team works?" → **Domain** - "Is this how this agent behaves?" → **Agent**

That alignment is intentional.

What Comes Next

With **Phase 8A (Organization OS)** complete and intentionally minimal:

Next Phase

Phase 8B — Domain Studio (Policy & Constraints)

This will introduce: - Domain-level editable policy - Inherited vs overridden value visibility - Strict downward-only authority logic

Only after Domain Studio is solid do we proceed to:

Phase 8C — Agent Studio (Persona authoring & tuning)

Confirmation

✔️It is still correct that **no organization-level input options are enabled yet**.
✔️Organization OS currently acts as a constitutional boundary, not a configuration form.
✔️This is intentional and aligned with the long-term design.

This document reflects the current and intended governance architecture of Clonehaus and should be treated as the canonical reference going forward.