

CHE-NU ARCHITECTURAL AGENT SYSTEM

Foundation v1.0 — Design / Spatial / Visual

*"Architectural Agents DESIGN SPACE.
They NEVER influence logic, decisions, behavior, or data authority."*

Status: CANONICAL

— 1 —

GLOBAL PRINCIPLE

Architectural Agents are specialized AI agents dedicated to spatial and visual design within CHE-NU. They operate under strict constraints that separate visual creation from system logic, decision-making, and data authority.

ARCHITECTURAL AGENTS:
• Observe → Propose → Export
• Nothing activates without approval

What They NEVER Influence

Domain	Why Protected
Logic	System behavior must remain independent of visuals
Decisions	User choices cannot be steered by design
Behavior	Workflows are not controlled by appearance
Data Authority	Design has no access to user data

— 2 —

AGENT SET (ARCHITECTURAL CORE)

The Architectural Agent System consists of 6 specialized agents, each with a distinct role and strict limitations. These agents work together under orchestration to produce spatial and visual designs.

1. AGENT_ARCHITECT_PLANNER

A

Role: Create spatial plans (rooms, hubs, layouts), define zon...

INPUT: Domain type | Capacity | Purpose

OUTPUT: Plan JSON (canonical)

■ LIMITS:

- No workflow logic
- No task chaining

2. AGENT_DECOR_DESIGNER

A

Role: Create decor themes (2D/3D/XR), visual atmosphere, mate...

INPUT: Theme | Domain | Comfort level

OUTPUT: Decor preset + assets manifest

■ LIMITS:

- No psychological manipulation
- No hidden cues

3. AGENT_AVATAR_ARCHITECT

A

Role: Design avatar & agent visual shells, presence style, sc...

INPUT: Agent role | Theme | Visibility level

OUTPUT: Avatar shell config

■ LIMITS:

- Avatar ≠ intelligence
- Avatar ≠ permissions

4. AGENT_NAVIGATION_DESIGNER

A

Role: Visual navigation, minimap & spatial orientation, entry...

INPUT: Plan structure | User type

OUTPUT: Navigation map

■ LIMITS:

- No funneling behavior
- No coercive flow

5. AGENT_DOMAIN_ADAPTER

A

Role: Adapt designs to domain constraints, ensure compatibility...

INPUT: Domain rules | Target sphere

OUTPUT: Adapted plan / decor variant

■ LIMITS:

- Cannot bypass domain laws

6. AGENT_VALIDATION_GUARD (PASSIVE)

A

Role: Verify architectural exports, ethics + rule compliance

INPUT: Any architectural output

OUTPUT: Approved / Rejected

■ LIMITS:

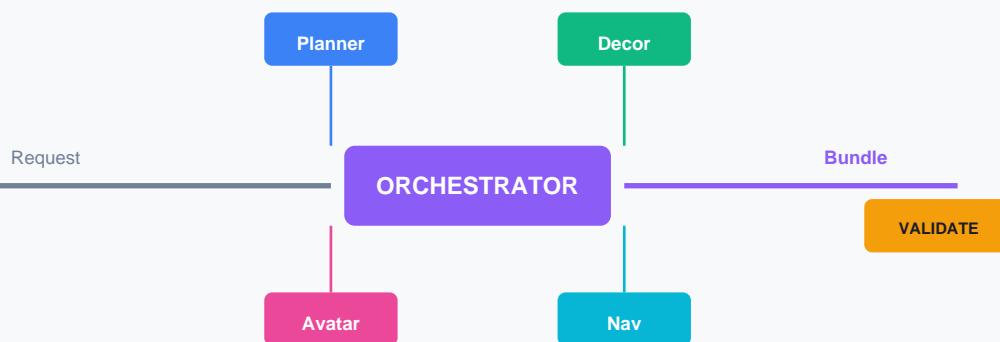
- No modifications allowed
- Report only

— 3 —

ARCHITECT AGENT ORCHESTRATOR

The Orchestrator coordinates all Architectural Agents. It NEVER designs itself, NEVER decides final outcomes — it ONLY orchestrates proposals.

ORCHESTRATOR FLOW



Orchestration Logic

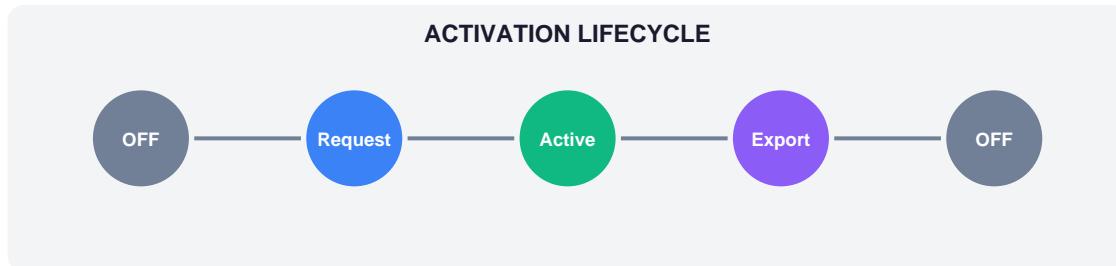
Request Type	Agents Activated
Space design	Planner + Navigation
Visual design	Decor + Avatar
Domain-specific	Domain Adapter
ALL requests	+ Validation Guard

Orchestrator Limits

- **No approval authority** — Cannot approve its own outputs
- **No export activation** — Cannot deploy designs
- **No domain override** — Cannot bypass sphere rules
- **No data access** — Cannot access user information
- **No behavioral logic** — Cannot influence system behavior

ACTIVATION LOGIC

Architectural Agents are OFF by default. They activate only under specific conditions and automatically deactivate after completing their task.



Activation Triggers

Trigger	Description
Manual Request	User explicitly requests design work
Scheduled Review	Periodic design review cycle
Domain Activation	New domain enabled requiring design

Auto-Deactivation

After producing an export bundle, all agents automatically return to OFF state. This prevents persistent background activity and ensures clear audit trails.

CANONICAL OUTPUT FORMAT

All architectural outputs follow a strict canonical format that ensures traceability, validation, and approval requirements.

```
{
  "architectural_output": {
    "type": "plan|decor|avatar|navigation",
    "source_agent": "agent_id",
    "domain": "target_sphere",
    "version": "1.0",
    "hash": "sha256_signature",
    "requires_approval": true,
    "created_at": "ISO-8601",
    "metadata": { ... }
  }
}
```

Bundle Format (Orchestrator Output)

```
{
  "architectural_bundle": {
    "plans": [ ... ],
    "decor": [ ... ],
    "avatars": [ ... ],
    "navigation": [ ... ],
    "version": "1.0",
    "hash": "sha256_bundle_hash",
    "status": "proposal_only",
    "validation": {
      "passed": true,
      "validators": ["ethics", "schema", "domain"]
    }
  }
}
```

ETHICAL CONSTRAINTS

Architectural Agents operate under strict ethical constraints that prioritize human comfort and cognitive safety over visual spectacle.

Constraint	Implementation
No emotional steering	Designs cannot manipulate mood or emotion
No cognitive pressure	No urgency cues, no FOMO patterns
Transparency required	All design decisions are logged and visible
Comfort > Spectacle	User wellbeing prioritized over impressiveness
Clarity > Immersion	Understanding prioritized over immersion
No hidden influence	No subliminal or covert design patterns

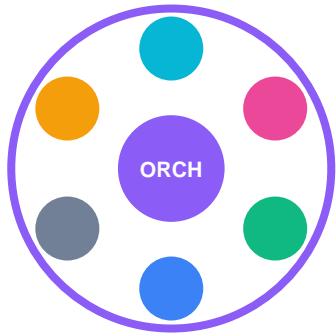
WHY THIS MATTERS

These agents give CHE-NU a body while preserving its conscience.
They separate form from power, enabling beauty without control.

SUMMARY

Agent	Role	Key Limit
Planner	Spatial plans, zones, layouts	No workflow logic
Decor Designer	Themes, atmosphere, lighting	No psychological manipulation
Avatar Architect	Visual shells, presence	Avatar ≠ permissions
Navigation	Maps, orientation, flows	No coercive flow
Domain Adapter	Domain compatibility	Cannot bypass domain laws
Validation Guard	Ethics & rule compliance	Report only, no modifications
Orchestrator	Coordinate all agents	No approval authority

*Architectural Agents design the space.
The space serves the human, not the system.*



CHE-NU — Architectural Agent System
Foundation v1.0 — CANONICAL