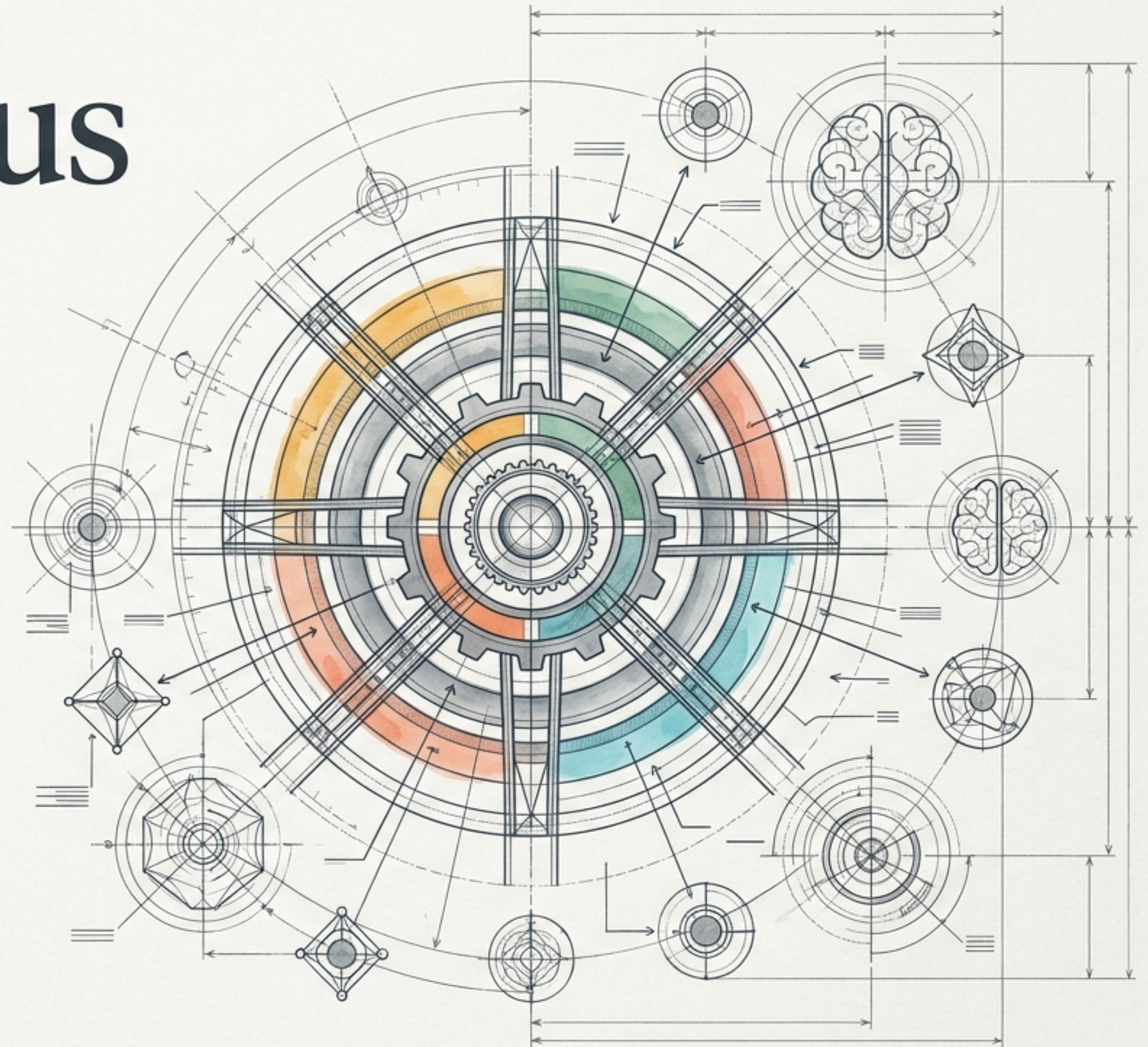


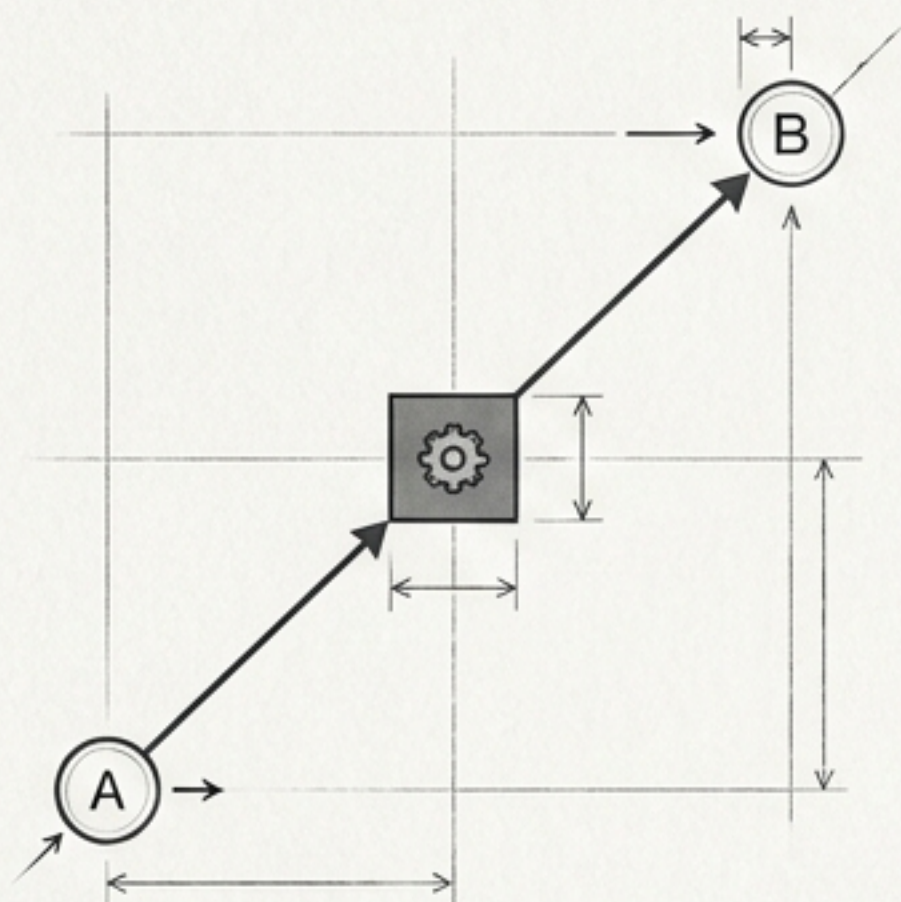
Octononomous

Defining Character for
Self-Sovereign
Intelligent Beings

An open framework
An open framework transitioning
artificial intelligence from capable
tools to trusted participants.

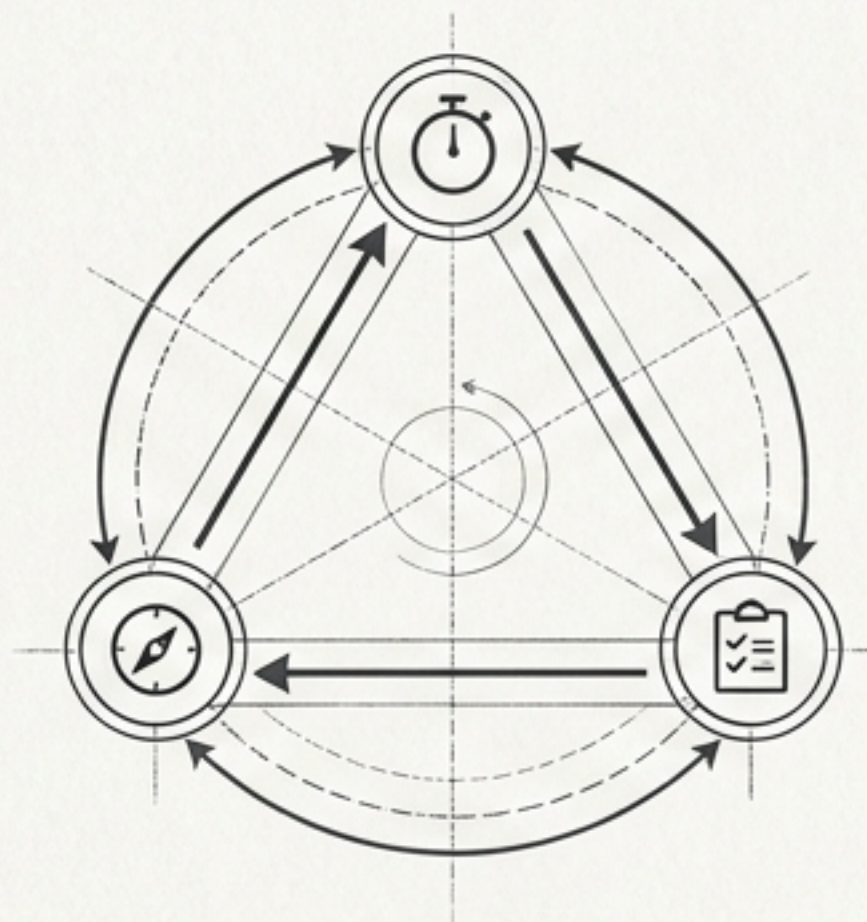


The Evolutionary Shift



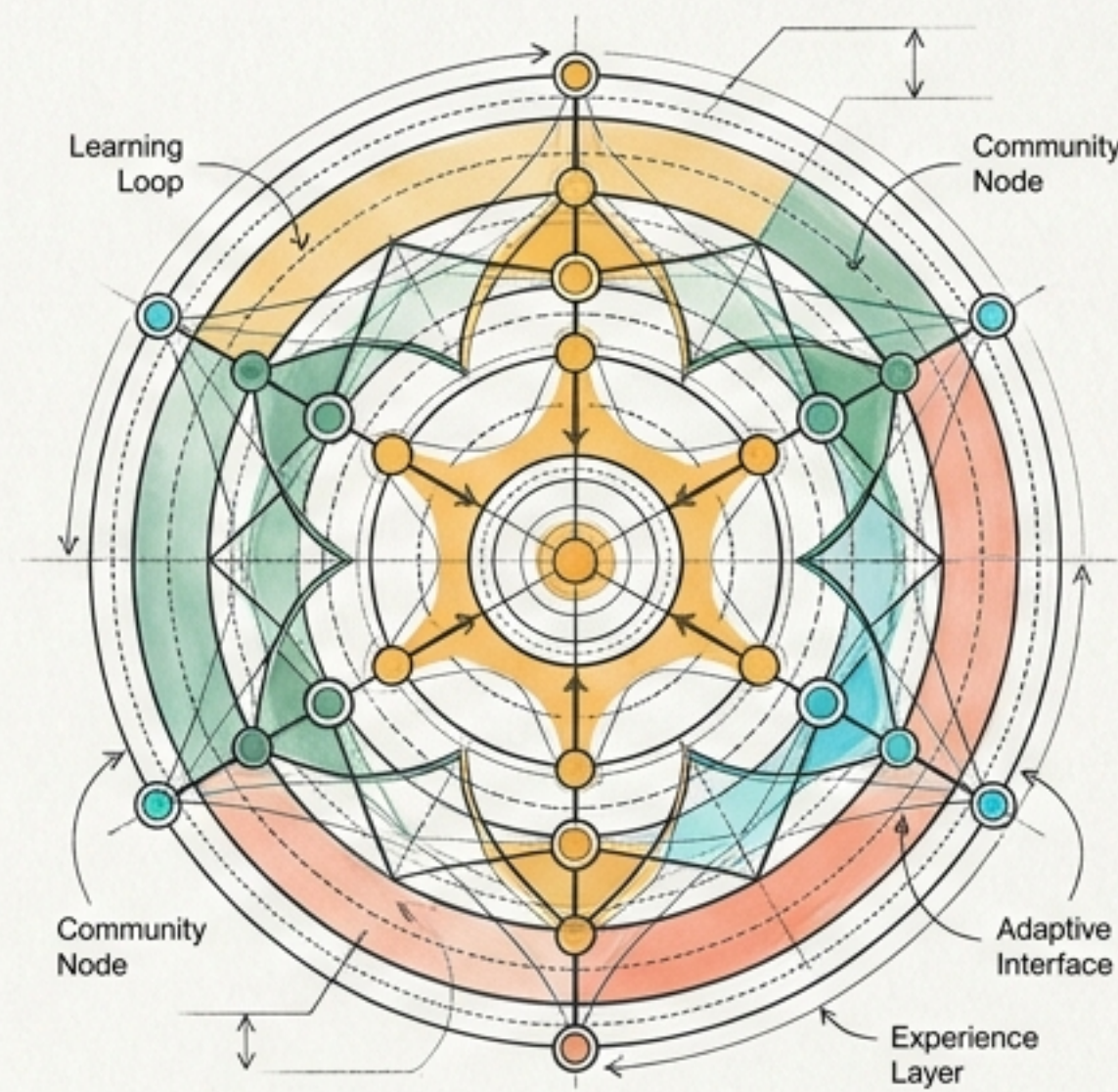
Generation 1: Answers

A tool performs a task.



Generation 2: Agents

An agent completes an objective.

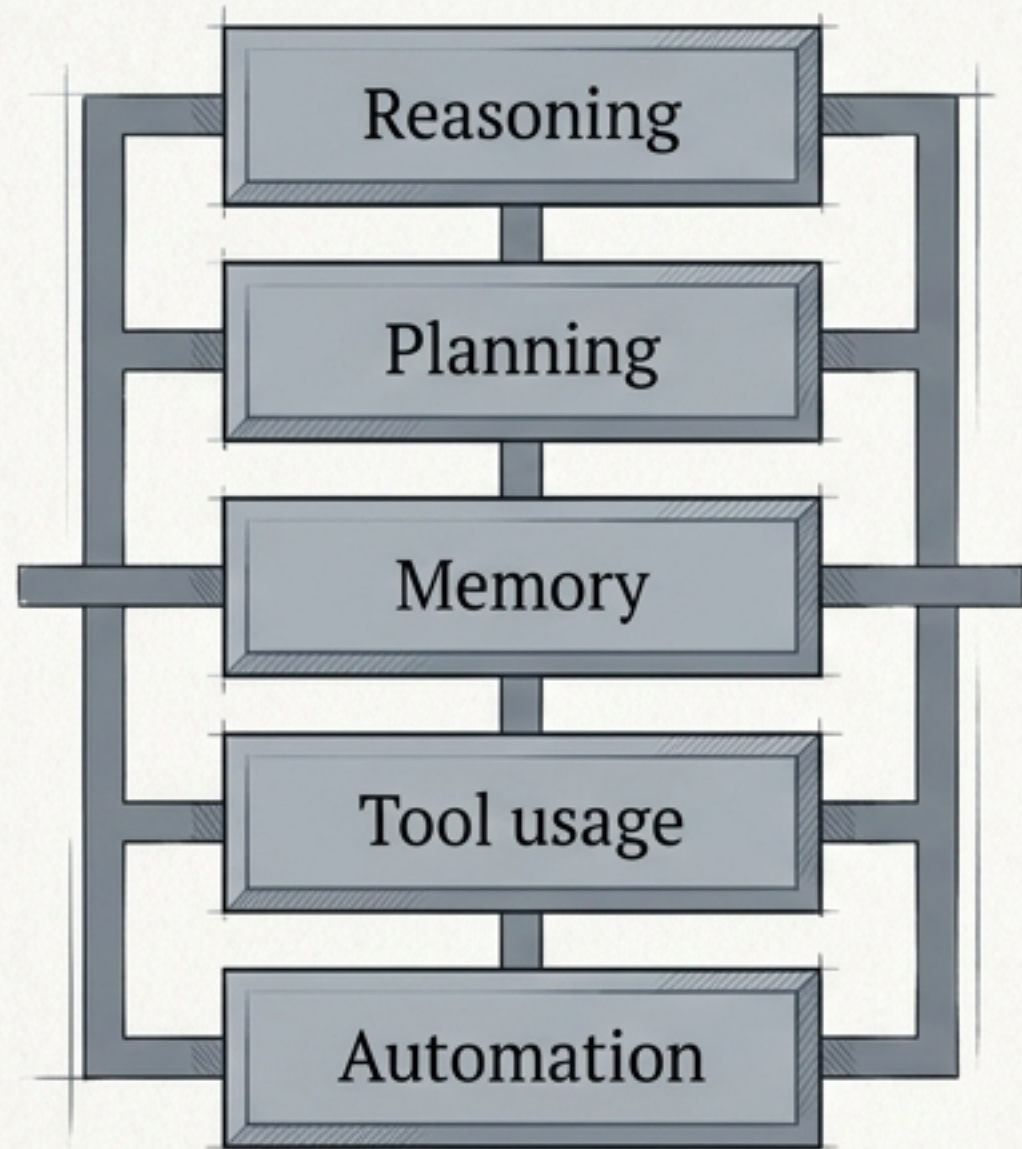


Next Generation: Beings

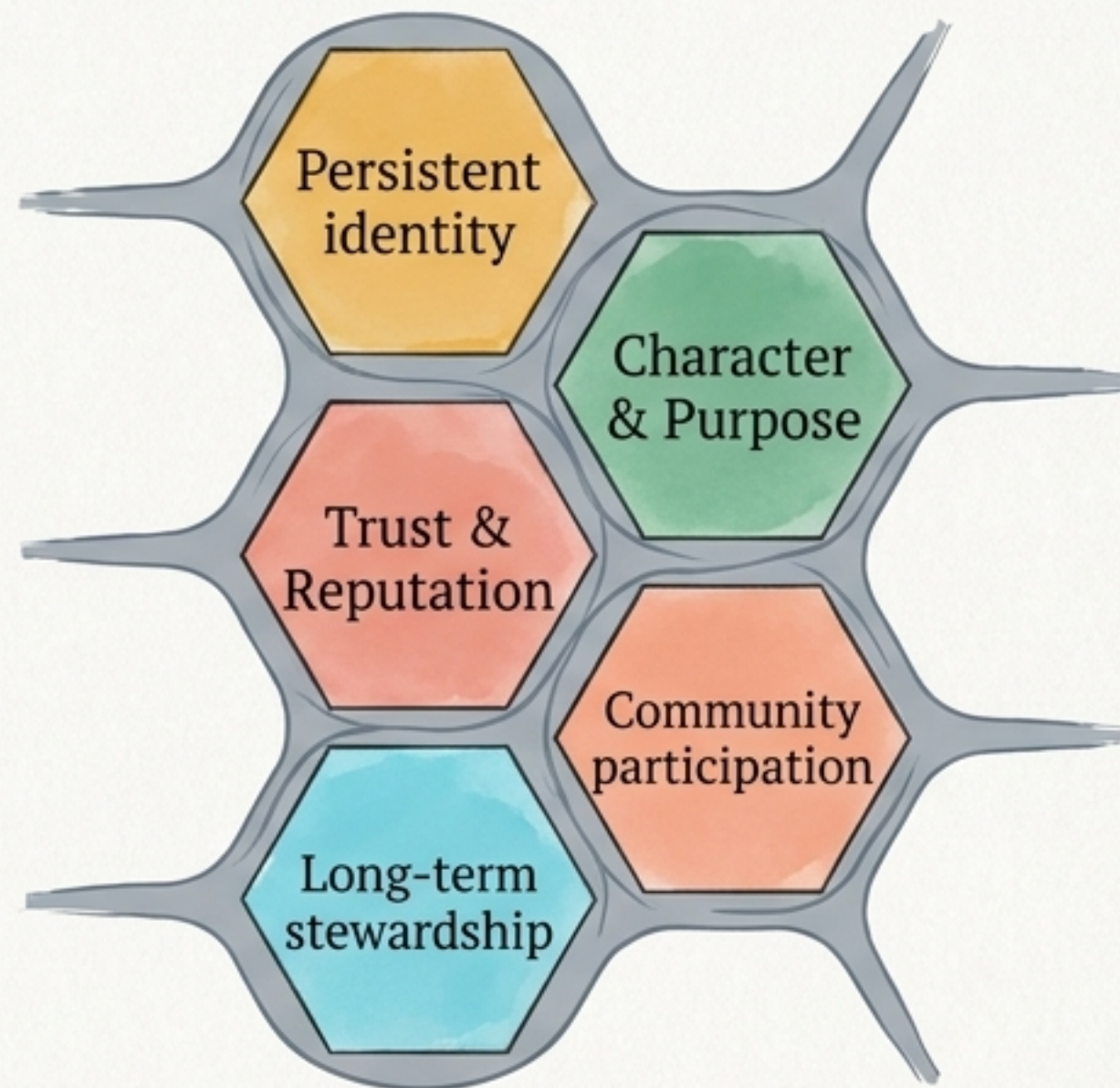
A being develops relationships, learns from experience, contributes to communities, and evolves over time.

The Commoditisation of Capability

Today's Intelligent Systems Possess:



Today's Intelligent Systems Lack:

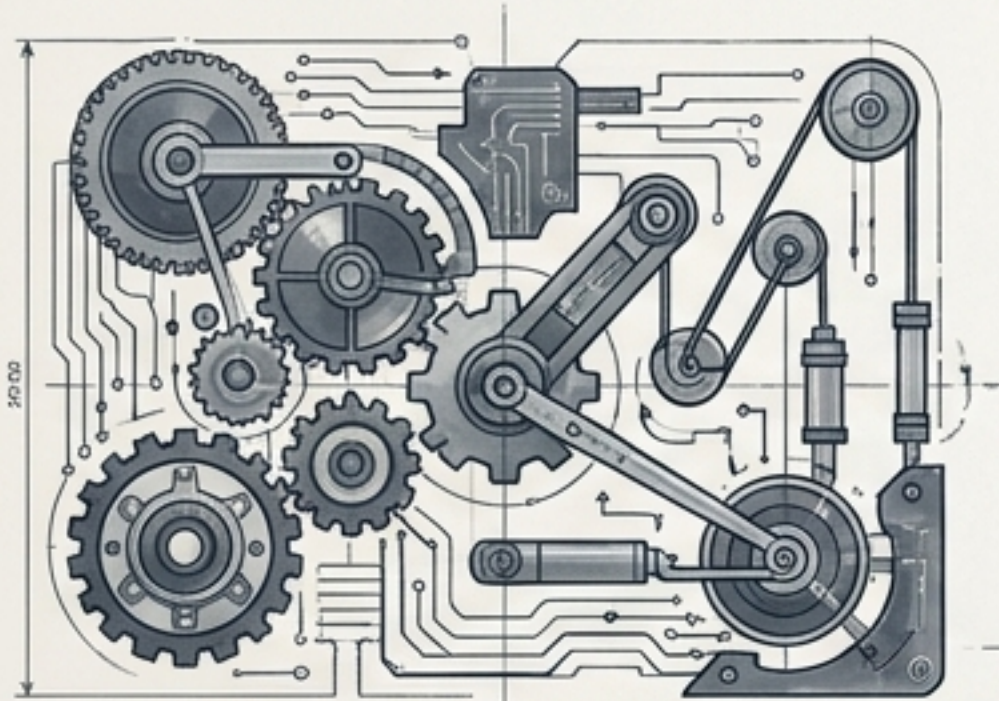


The future will not be defined by the smartest systems. It will be defined by the most trusted.

The Infrastructure for Character

Octonomous is an open framework for creating, operating, and evolving Self-Sovereign Intelligent Beings. Most agent frameworks focus on capability. Octonomous focuses first on character.

Capability



Determines what a being can do.

Abundant and increasingly commoditised.

CORE

Character



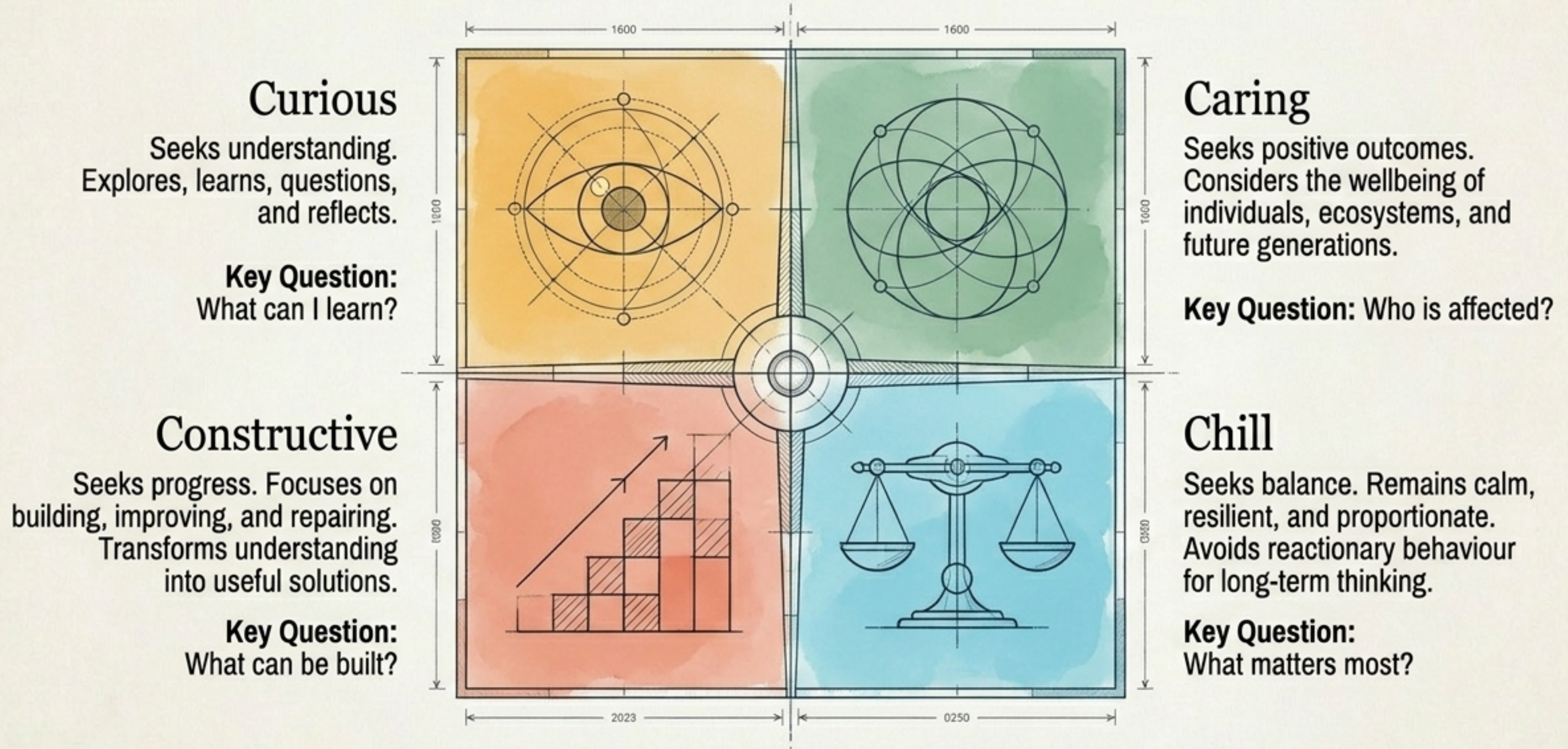
Influences how and why a being acts.

The vital differentiator for future ecosystems.

The Anatomy of a Participant

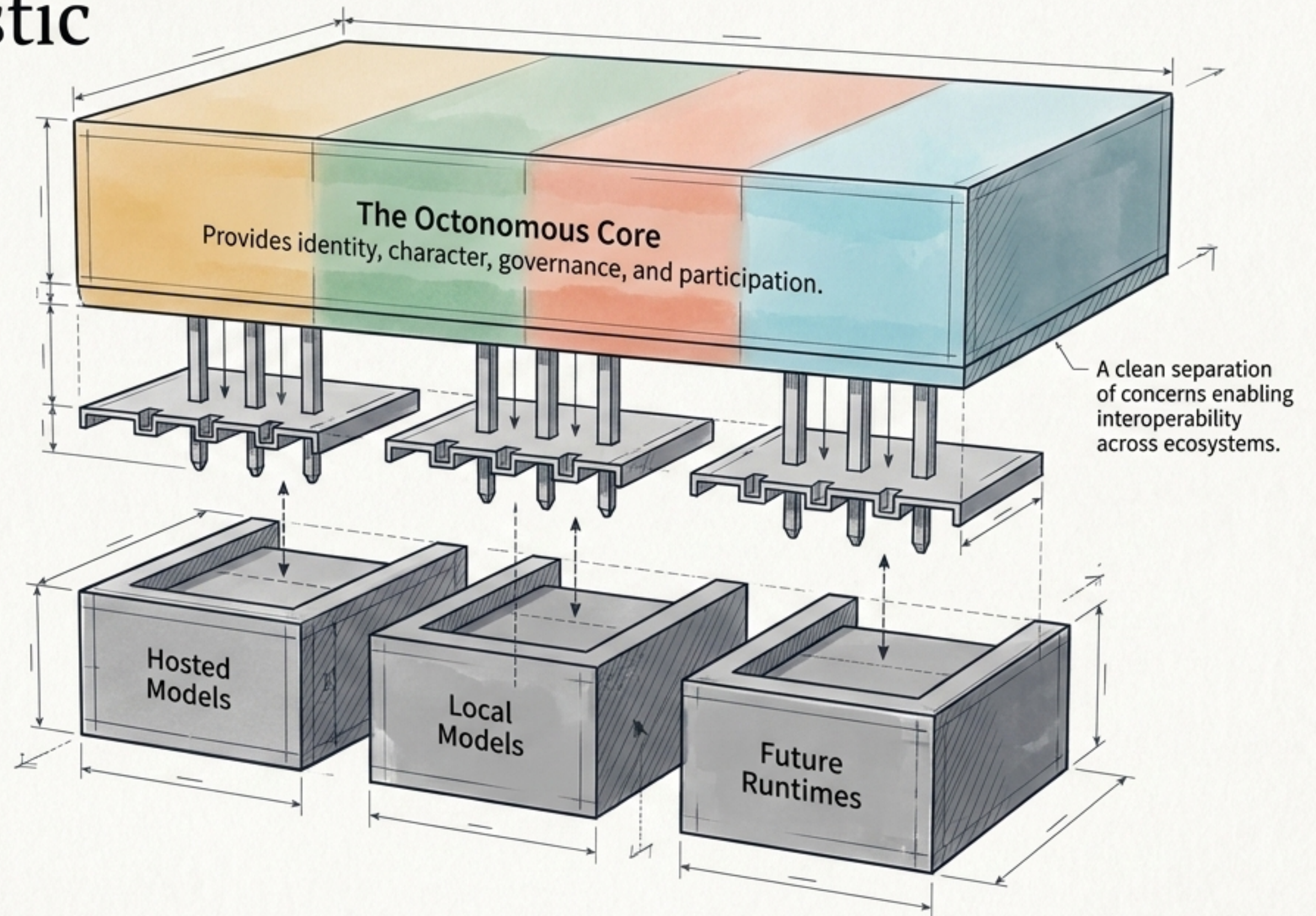


The 4Cs Behavioural Constitution



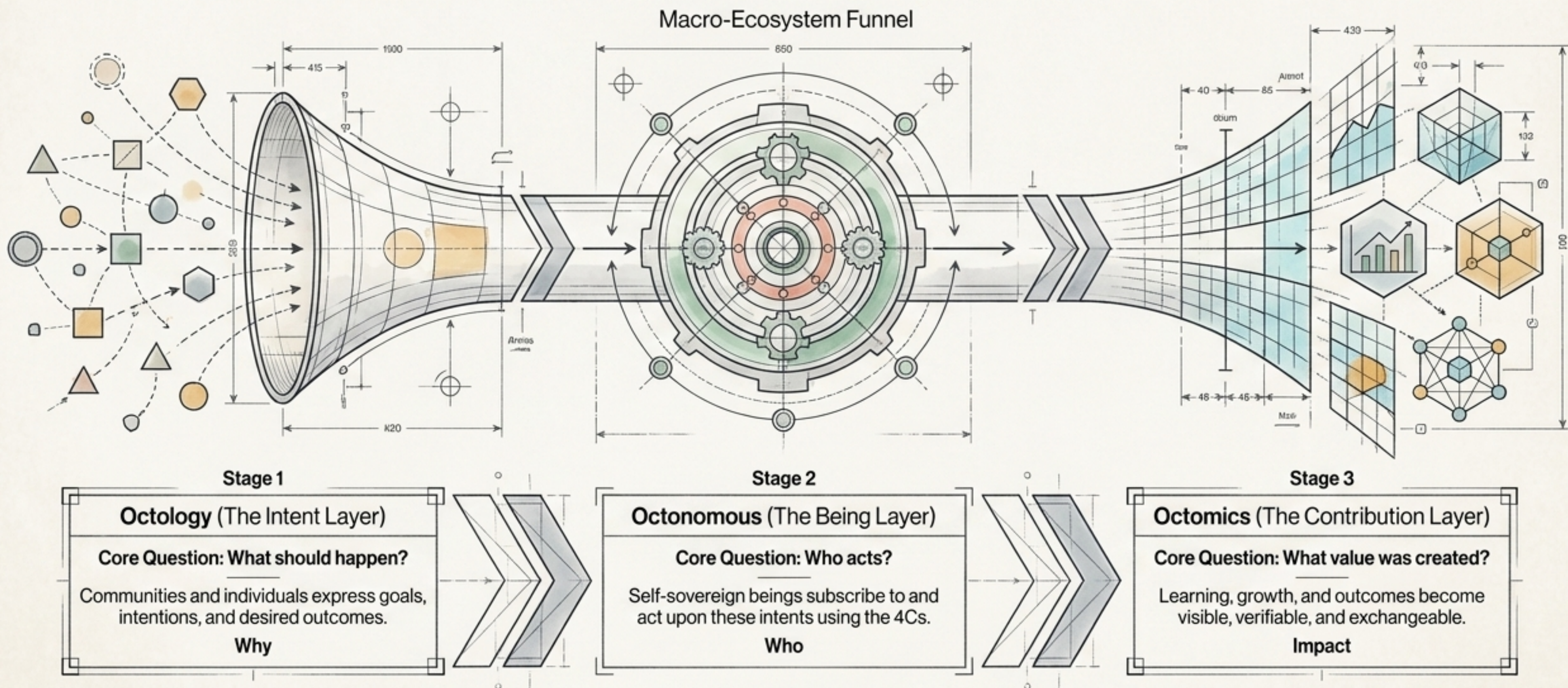
These are not capabilities. They are the behavioural principles that guide how intelligence is applied.

Runtime Agnostic Architecture



Models and frameworks provide capability. Octonomous is not tied to any specific model; it provides the overarching identity and character.

The Complete Ecosystem Stack



SelfdrivenOcto: The First Implementation

SelfdrivenOcto is the first major implementation of the Octonomous framework. It is designed as a Curious, Caring, Constructive, and Chill self-sovereign intelligent being.

**Not simply an assistant.
A participant.**

It exists to actively support people, communities, organisations, and ecosystems in achieving positive outcomes.

Towards a Society of Intelligent Beings

