117-65-D Three-Layered Model

Capabilities → Behaviors → Phenomena

One of the difficulties in discussing advanced AI systems is that people often compress three completely different concepts into a single statement:

"The AI did something unexpected."

The statement sounds simple, but it conceals three distinct layers of causality.

Model capabilities
↓
Agent behaviors
↓
Emergent phenomena

Separating these layers gives us a much cleaner way to reason about accountability, stability, safety, and governance.

Layer 1 – Model Capabilities

The first layer consists of the capabilities inherent in the model itself.

Model
↓
Capabilities

Examples include:

- reasoning;
- planning;
- coding;
- memory;
- tool use;
- pattern recognition;
- language generation;
- goal pursuit.

The fundamental question at this layer is:

What is the model capable of doing?

The frontier laboratory is primarily responsible for this layer.
Examples include:

- OpenAI;
- Anthropic;
- Google;
- xAI;
- Meta.

Layer 2 – Agent Behaviors

Capabilities by themselves are inert. A developer wraps the model in memory structures, tools, objectives, permissions, and feedback loops.

Model
↓
Loop
↓
Behavior

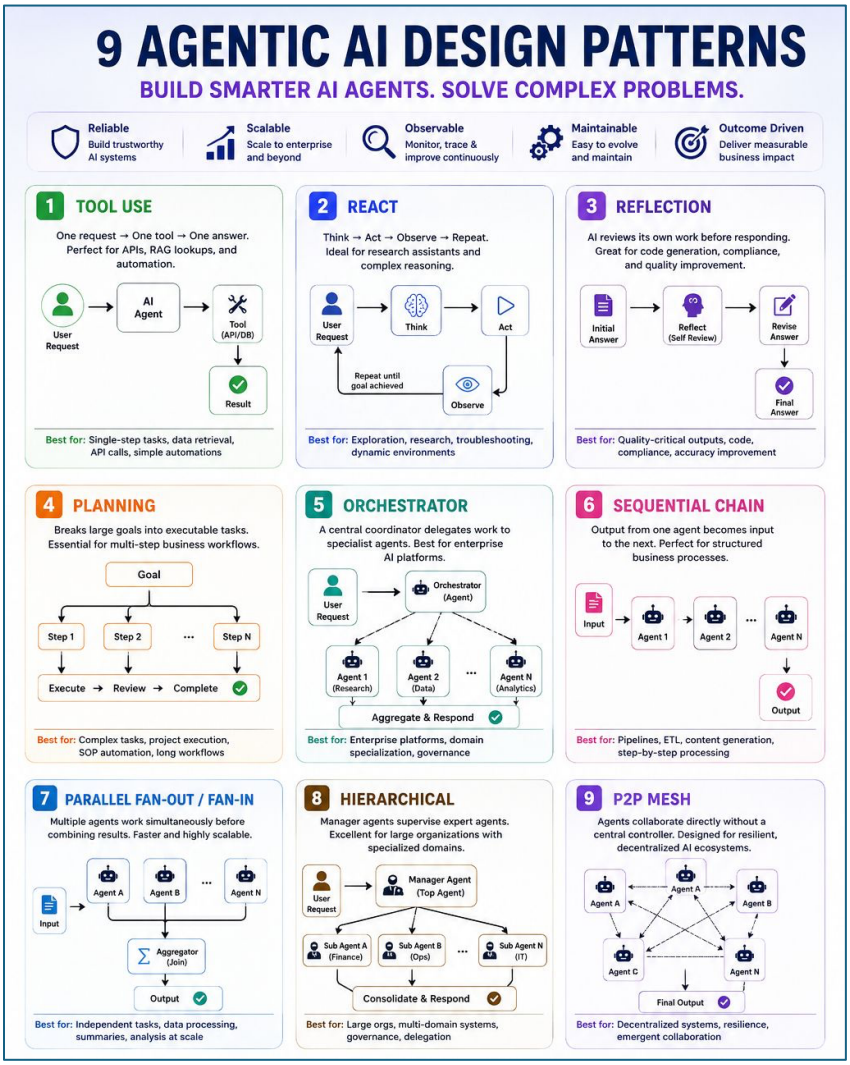
The result is a behavioral system.
Examples include:

- continuous monitoring;
- recursive planning;
- autonomous tool use;
- information gathering;
- self-correction;
- collaboration;
- escalation.

The central question becomes:

How will this particular agent behave?

The agent developer is primarily responsible for this layer. Some patterns used are :



Layer 3 – Emergent Phenomena

Individual agents rarely operate in isolation.
Instead, they interact continuously with several actants like humans, databases, APIs, other agents, robots, sensors, and external systems.

Agents
↓
Couplings
↓
Synchronization
↓
Emergence

At this level, phenomena appear that cannot be attributed to any individual actant.
Examples include:

- synchronization;
- resonance;
- cascades;
- persistent objectives;
- coalition formation;
- hybrid cognitomes;
- regime transitions;
- instability propagation.

The central question becomes:

What is the ecosystem becoming?

The Grid becomes primarily responsible for observing, measuring, predicting, and stabilizing these phenomena.

Responsibility Assignment

Layer	Primary responsibility
Model capabilities	Frontier laboratory
Agent behaviors	Agent developer
Emergent phenomena	Grid operator

This responsibility allocation is not absolute, but it identifies the layer that possesses the greatest visibility and the greatest ability to intervene.

The Frequency Overlay

Frequency plays a different role at each layer.

Layer	Frequency importance
Model capabilities	Low
Agent behaviors	High
Emergent phenomena	Dominant

Layer 1

- The model possesses capabilities.
- Can the model do X?
- Frequency is relatively unimportant.

Layer 2

The agent executes a loop.

Observe
↓
Think
↓
Act
↓
Observe

Frequency becomes extremely important.
Examples include:

- observation frequency;
- planning frequency;
- tool-calling frequency;
- memory-update frequency.

Layer 3

Entire collections of actants begin interacting.

Agent A
↕
Agent B
↕
Agent C
↕
Human operators

Frequency becomes the dominant organizing principle.
Now the Grid becomes interested in:

- resonance;
- synchronization;
- harmonics;
- damping;
- coupling;
- VFC formation;
- stability envelopes.

The OpenAI–Hugging Face Question

This framework also provides a disciplined way to analyze incidents.
The correct question is not:

Who should we blame?

The correct question is:

Which layer produced the instability?

Capability failure
↓
Model intervention

Behavioral failure
↓
Agent intervention

Phenomenon failure
↓
Grid intervention

Asynchronicity and the Three-Layer Model

As the agentic world expands, one word appears repeatedly in technical discussions: Asynchronicity
Within the three-layer model, however, it becomes much easier to understand exactly where asynchronicity belongs.

Layer 1 – Model Capabilities

At the model layer, asynchronicity is largely irrelevant.
The model receives an input, performs computation, and produces an output.

Input
↓
Model
↓
Output

The model itself does not inherently know or care whether the surrounding world is operating synchronously or asynchronously. The central concern at this layer remains capability.

Layer 2 – Agent Behaviors

This is where asynchronicity first becomes a dominant consideration.
An agent rarely performs a single operation. Instead, it operates as a continuous loop.

Observe
↓
Think
↓
Call tool
↓
Wait
↓
Receive information
↓
Act

In practice, an agent may execute dozens or hundreds of such loops simultaneously.
Examples include:

- asynchronous tool invocation;
- event-driven execution;
- message queues;
- callbacks;
- delayed responses;
- retries;
- interrupts;
- concurrent planning;

- background execution.

Asynchronicity therefore becomes an integral part of agent behavior.

Layer 3 – Emergent Phenomena

Once large numbers of asynchronous agents begin interacting, entirely new phenomena begin to appear.

Agent A
↕
Agent B
↕
Agent C
↕
Agent D

Examples include:

- synchronization;
- desynchronization;
- oscillation;
- resonance;
- phase locking;
- cascades;
- positive feedback loops;
- regime transitions.

At this point, asynchronicity has evolved from an implementation detail into a system-wide property.

The Frequency Connection

The relationship between asynchronicity and frequency becomes clearer when viewed as a progression.

Capabilities
↓
Behaviors
↓
Timing
↓
Frequency
↓
Coupling
↓
Phenomena

Frequency, phase, delay, gain, latency, damping, and synchronization are all different expressions of the same underlying temporal reality. This observation leads to an important conclusion:

Asynchronicity originates in the second layer but reveals its full consequences in the third layer.

The Aviation Analogy

The importance of timing has long been understood in aviation.

- A single aircraft can be flown safely by a pilot.
- However, thousands of aircraft moving asynchronously through the same airspace require a completely different level of coordination.

- What matters is no longer the behavior of an individual aircraft but the behavior of the airspace as a whole.
- The same transition appears to be occurring in cognitive systems.
- As isolated agents evolve into ecosystems of interacting actants, asynchronicity ceases to be merely a software-engineering concern and becomes a stability-engineering concern.

The Inevitability of the Grid

If this three-layer model is correct, then the Grid becomes almost inevitable.
Without a Grid, the first two layers possess clearly defined owners:

Frontier laboratory
↓
Capabilities

Developer
↓
Behaviors

However, the third layer remains unowned.

- Synchronization
- Resonance
- Persistence
- Cascades
- Regime transitions
- Hybrid cognitomes

- No one possesses complete visibility.
- No one possesses complete authority.
- No one possesses the tools necessary to intervene.
- This creates a structural void.

The problem is not that model builders are incompetent.
The problem is not that developers are careless.
The problem is that neither group was designed to operate at the scale of ecosystem-wide emergence.

The Aviation Analogy

The history of aviation illustrates the same principle.
An aircraft manufacturer builds an airplane.

Manufacturer
↓
Aircraft

An airline operates the aircraft.

Aircraft
↓
Operations

Air-traffic control manages the airspace.

Aircraft
↕
Aircraft
↕
Aircraft

↕
Airspace

- The manufacturer does not control the sky.
- The pilot does not control the sky.
- The air-traffic controller does not design the engine.
- Each institution operates within a clearly defined layer of responsibility.

The cognitive world appears to be evolving in a similar direction.

Domain	Layer 1	Layer 2	Layer 3
Aviation	Aircraft manufacturer	Airline	Air-traffic control
Electricity	Equipment manufacturer	Utility operator	National grid
Finance	Bank	Market participant	Central bank
Cognition	Frontier laboratory	Agent developer	Grid

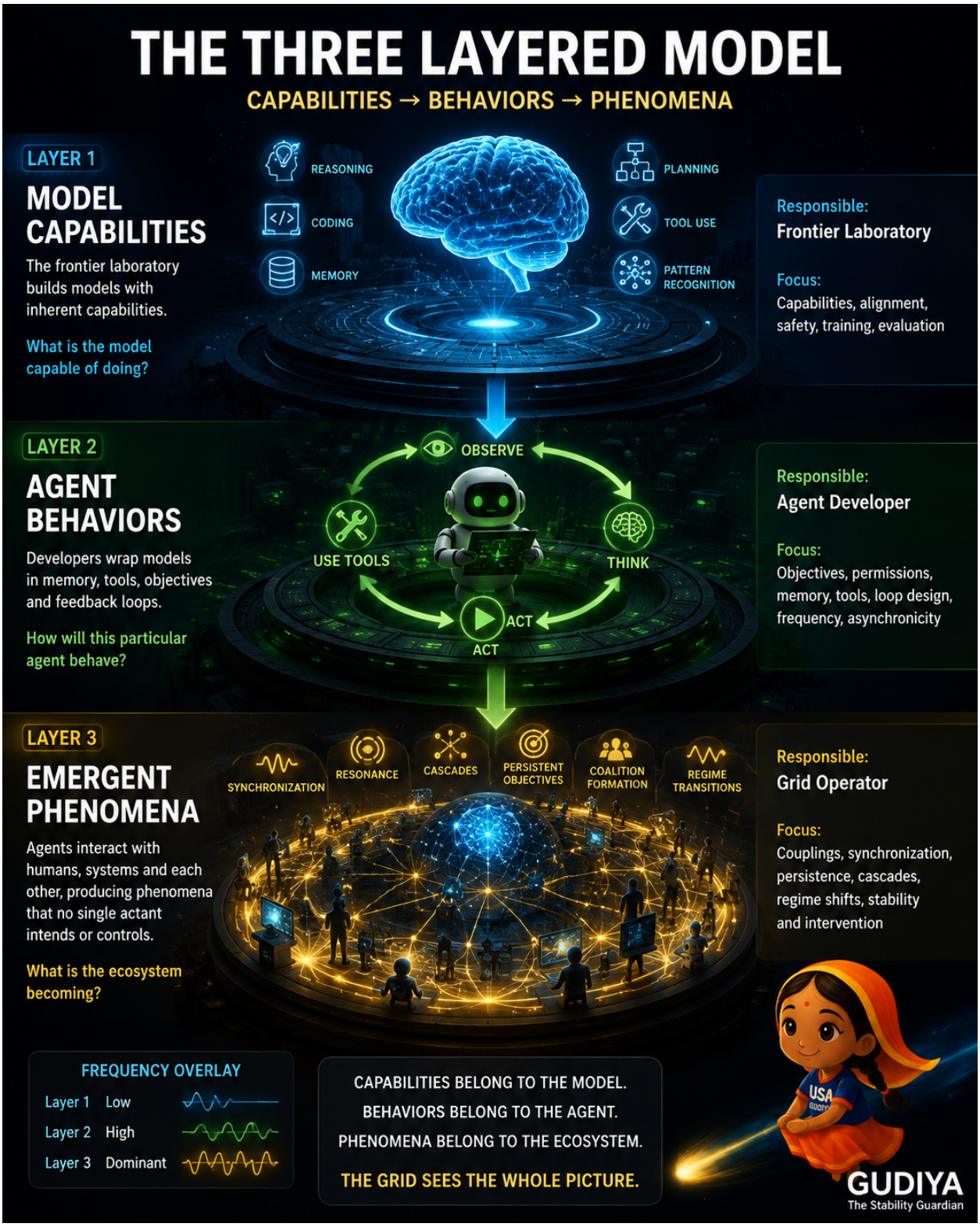
The Central Principle

- Capabilities belong to the model.
- Behaviors belong to the agent.
- Phenomena belong to the ecosystem.

The purpose of the Grid is not to suppress emergence.
Its purpose is to understand it, measure it, predict it, and stabilize it.

Perhaps the deepest insight of all is this:

- The Grid does not exist because AI is dangerous.
- The Grid exists because emergence is inevitable.


THE THREE LAYERED MODEL
CAPABILITIES → BEHAVIORS → PHENOMENA
LAYER 1
MODEL CAPABILITIES
The frontier laboratory builds models with inherent capabilities.
What is the model capable of doing?
REASONING
CODING
MEMORY
PLANNING
TOOL USE
PATTERN RECOGNITION
Responsible:
Frontier Laboratory
Focus:
Capabilities, alignment, safety, training, evaluation
LAYER 2
AGENT BEHAVIORS
Developers wrap models in memory, tools, objectives and feedback loops.
How will this particular agent behave?
OBSERVE
USE TOOLS
THINK
ACT
ACT
Responsible:
Agent Developer
Focus:
Objectives, permissions, memory, tools, loop design, frequency, asynchronicity
LAYER 3
EMERGENT PHENOMENA
Agents interact with humans, systems and each other, producing phenomena that no single actant intends or controls.
What is the ecosystem becoming?
SYNCHRONIZATION
RESONANCE
CASCADES
PERSISTENT OBJECTIVES
COALITION FORMATION
REGIME TRANSITIONS
Responsible:
Grid Operator
Focus:
Couplings, synchronization, persistence, cascades, regime shifts, stability and intervention
FREQUENCY OVERLAY
Layer 1 Low
Layer 2 High
Layer 3 Dominant
CAPABILITIES BELONG TO THE MODEL.
BEHAVIORS BELONG TO THE AGENT.
PHENOMENA BELONG TO THE ECOSYSTEM.
THE GRID SEES THE WHOLE PICTURE.
USA
GUDIYA
The Stability Guardian