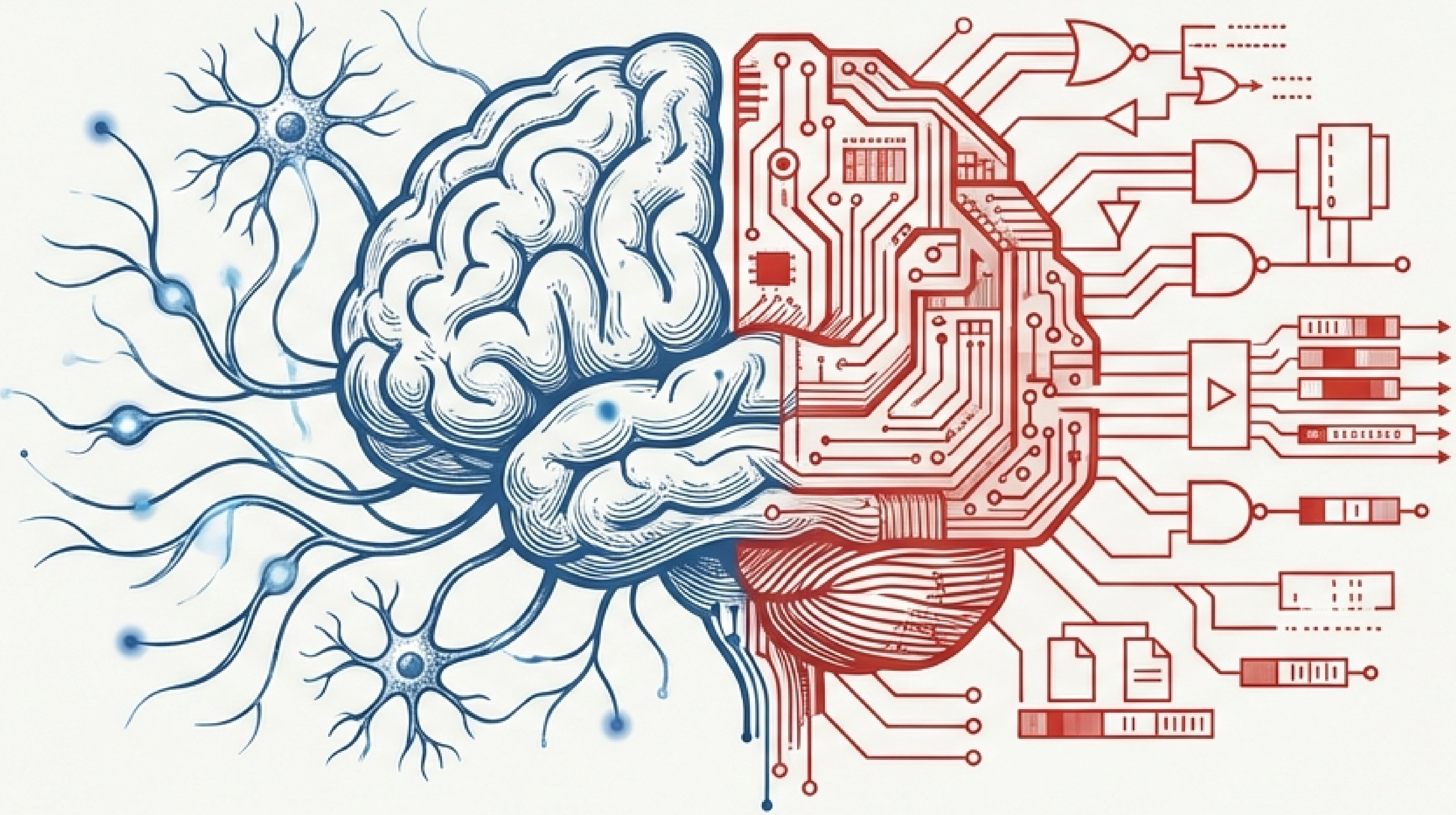


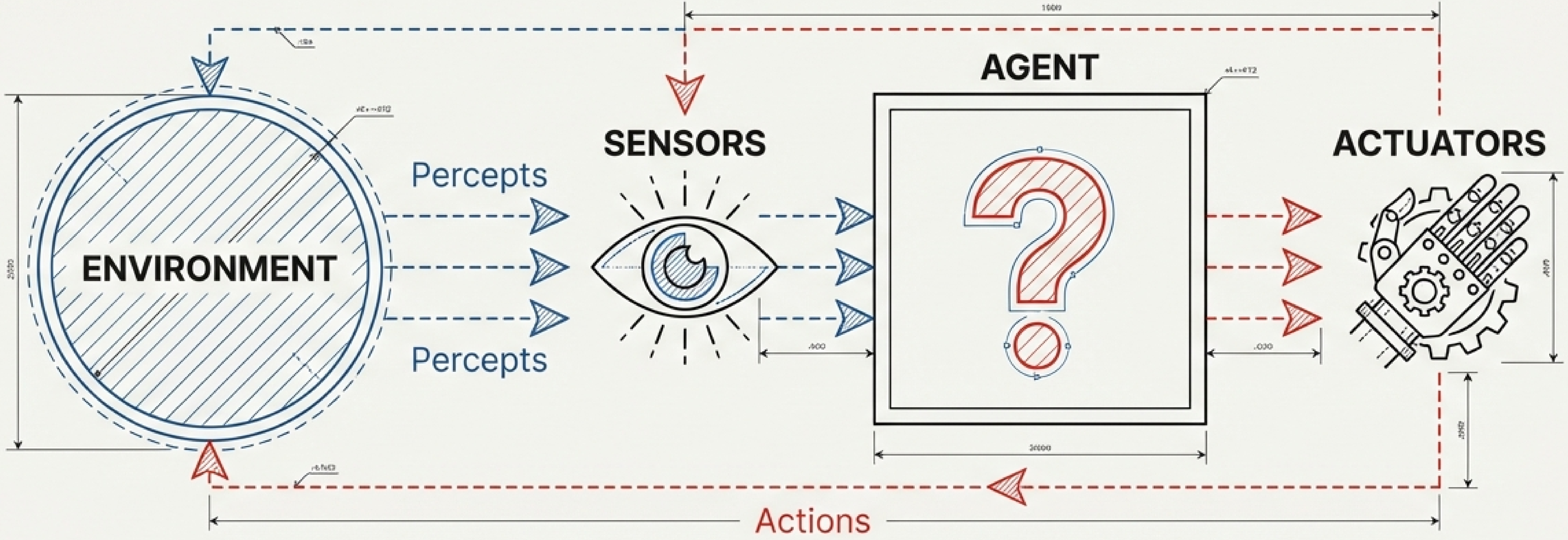
Intelligent Agents: The Anatomy of Autonomy

From environmental constraints to architectural solutions.

Artificial Intelligence is defined by the design of Rational Agents. An agent is not merely a program; it is anything that perceives its environment through sensors and acts upon that environment through actuators. This deck explores how we mathematically define 'doing the right thing' and the architectural blueprints required to achieve it.

MIT
MathInTech

The Fundamental Loop of Interaction



Percept Sequence (P^*)

"The complete history of everything the agent has ever perceived."

Agent Function ($f: P^* \to A$)

'The abstract mathematical description mapping percept histories to actions."

Agent Program

The concrete implementation running on the physical architecture."

Rationality is Not Omniscience

The Theory

The Definition

A rational agent acts to maximize its **expected performance measure**, given:

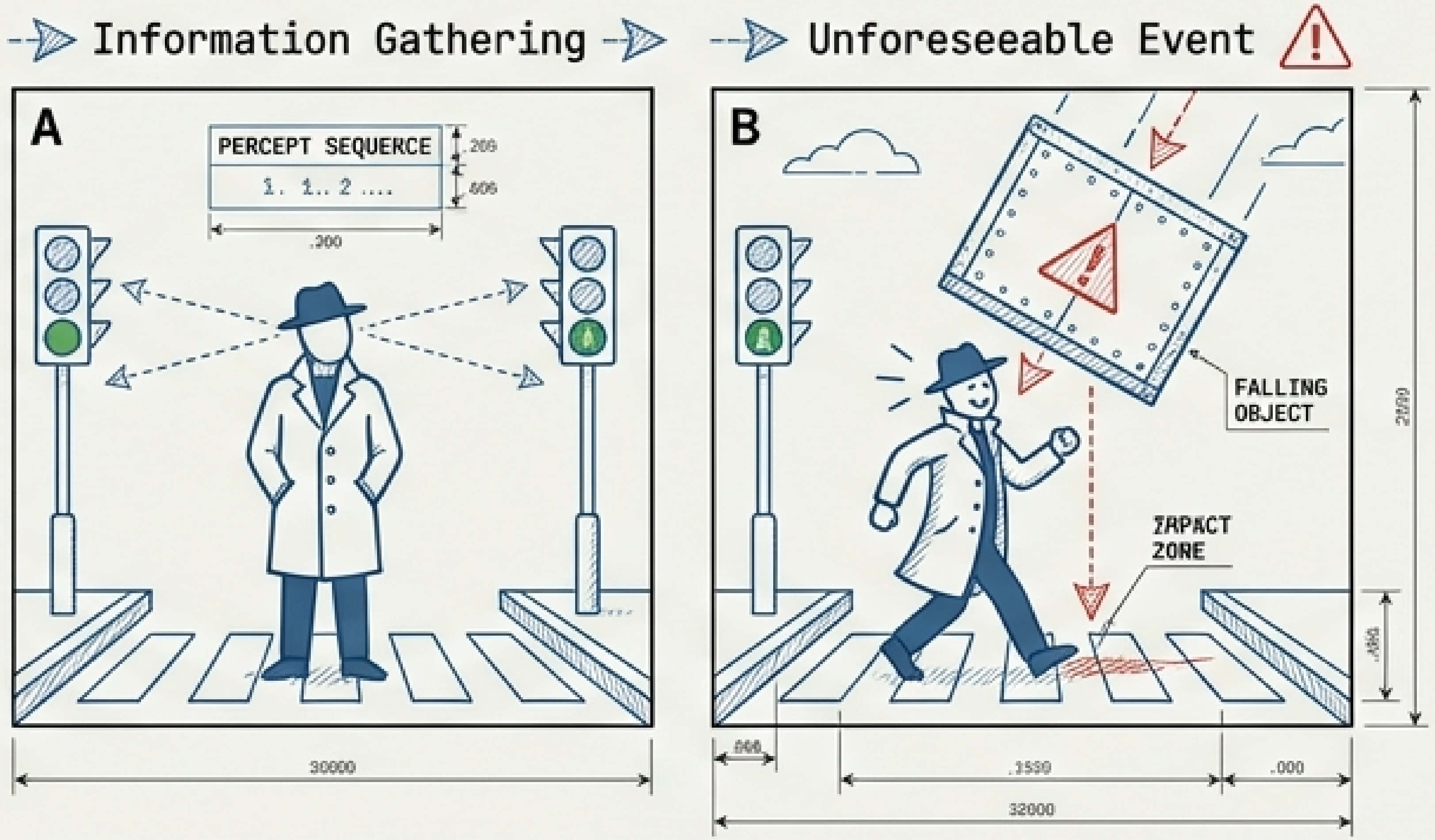
1. The evidence provided by the percept sequence to date.
2. Whatever built-in knowledge the agent possesses.

The Critical Distinction

Omniscience = Knowing the actual outcome (Impossible).

Rationality = Maximizing *expected* outcome based on current data.

The Parable

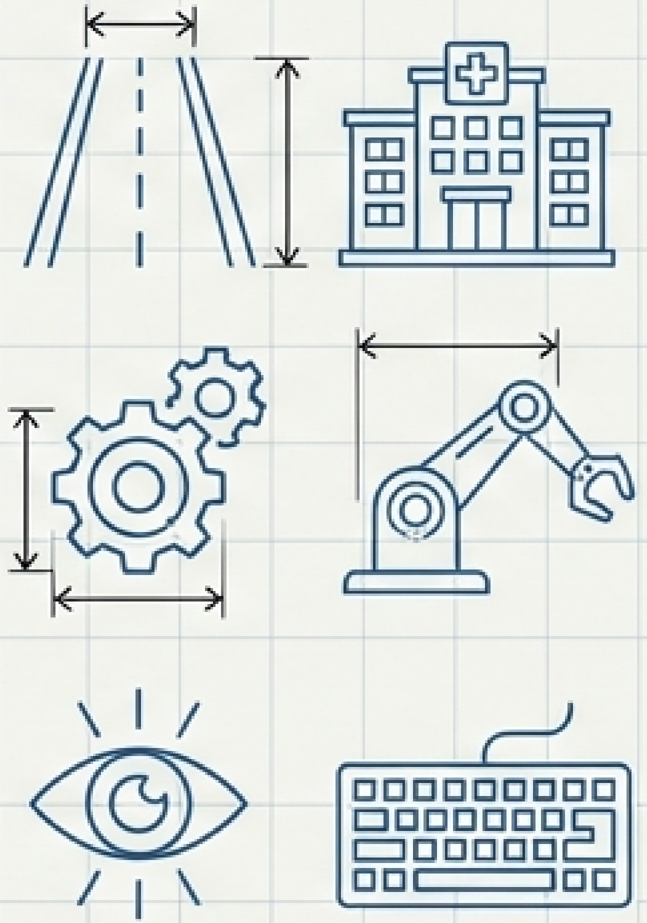


The Crossing Street Example: If you look, see no traffic, cross, and are hit by a falling airplane part, you were still rational. Rationality is about process, not clairvoyance.

The Design Specification: PEAS

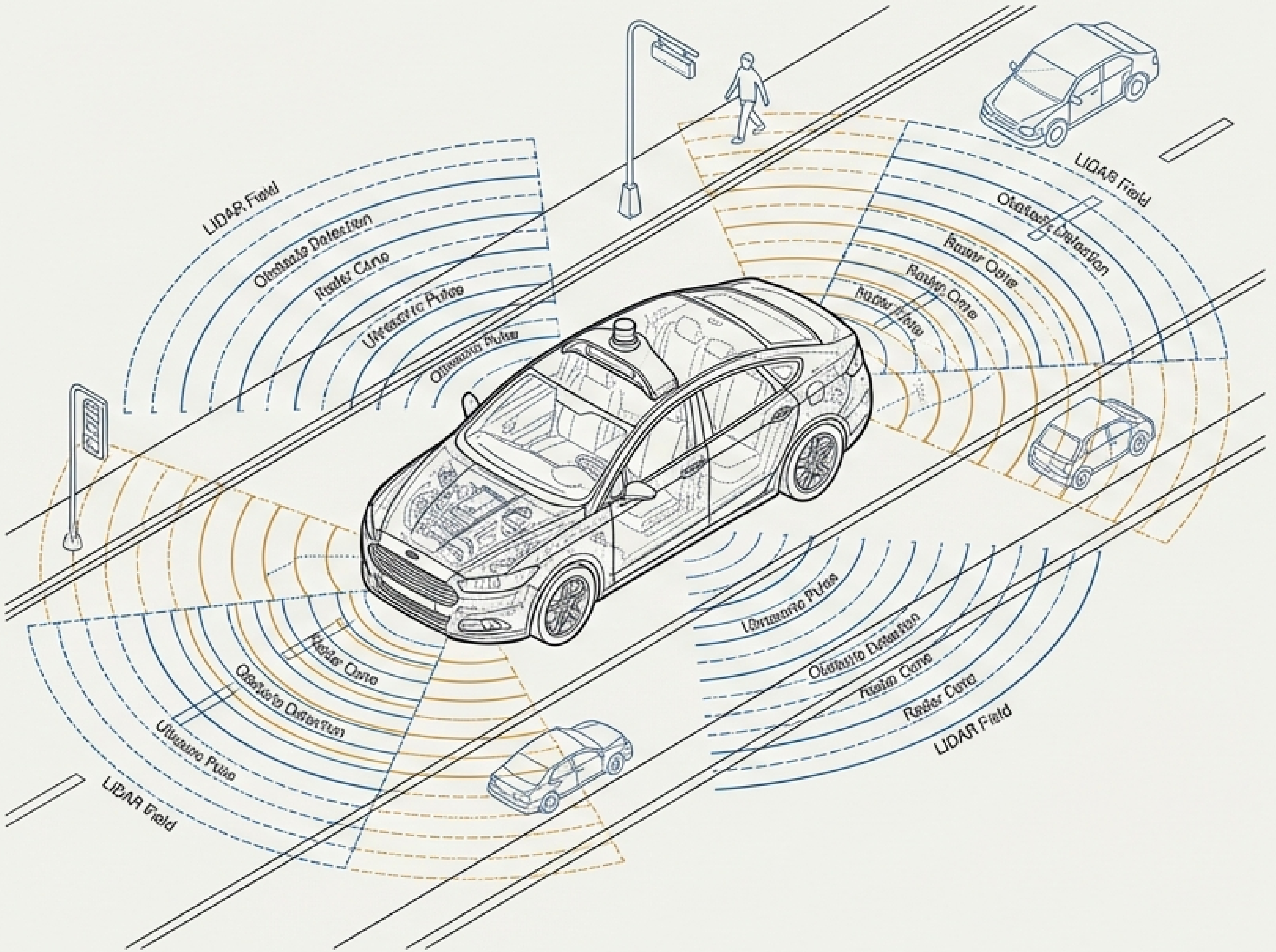
P E A S

- **Performance Measure**
  The objective criterion for success (e.g., safety, speed, profit).
- **Environment**
  The external context in which the agent operates (e.g., roads, hospitals).
- **Actuators**
  The tools used to affect the world (e.g., steering wheel, screen display).
- **Sensors**
  The inputs used to perceive the world (e.g., cameras, keyboard).



→ In designing an agent, the first step must always be to specify the task environment as fully as possible.

Case Study: The Automated Taxi



Spec Sheet

PEAS Breakdown: Automated Taxi

- **Performance**
  Safe, fast, legal, comfortable trip, maximize profits
- **Environment**
  Roads, other traffic, pedestrians, weather
- **Actuators**
  Steering, accelerator, brake, signal, horn, display
- **Sensors**
  Cameras, sonar, speedometer, GPS, odometer, engine sensors

Complexity Correlation: The complexity of the PEAS definition dictates the necessary complexity of the Agent Architecture.

TITLE: Case Study: The Automated Taxi	
PROJECT: RATIONAL AGENTS	

The Environment: Visibility and Competition

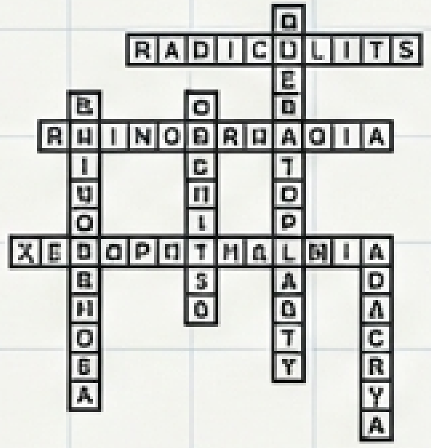
Observability

Fully Observable — **Partially Observable**

Fully Observable: Sensors detect all relevant aspects. No memory required.

Partially Observable: Noisy or missing data. Requires internal state/memory.

Agents



Single Agent — **Multi-Agent**

Single Agent: Solitaire, Crosswords.

Multi-Agent: Competitive (Chess) or Cooperative (Traffic). Communication emerges as a rational behavior.

The Environment: Uncertainty and Time

Predictability

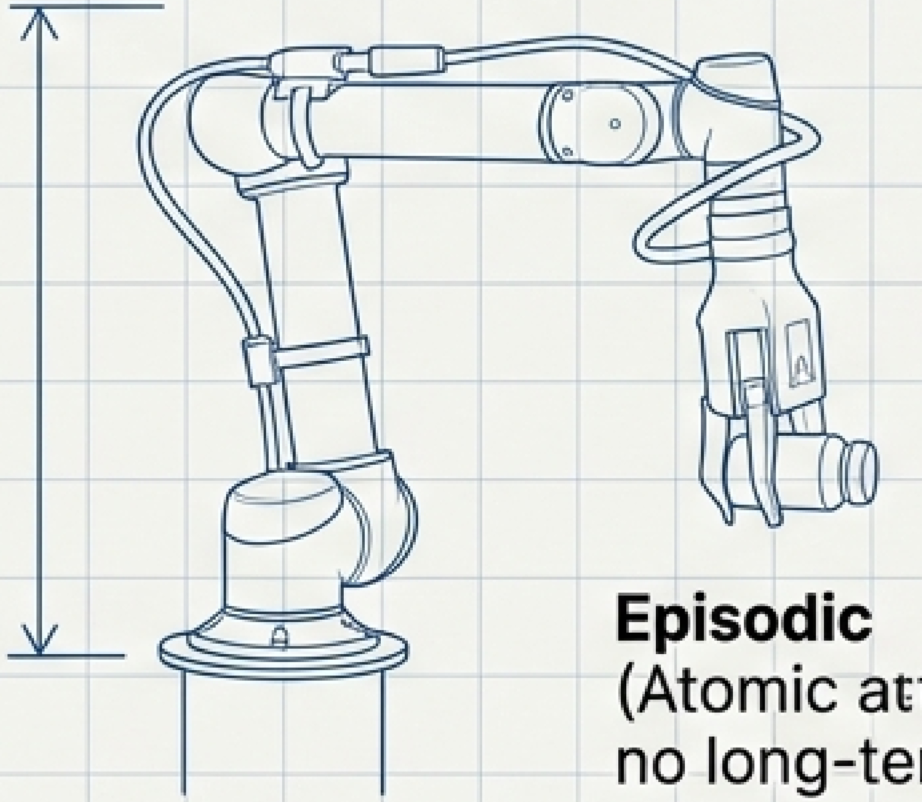
Deterministic vs. Stochastic

Deterministic: Next state is guaranteed (e.g., Chess).
Stochastic: Next state is uncertain (e.g., Taxi driving, weather).

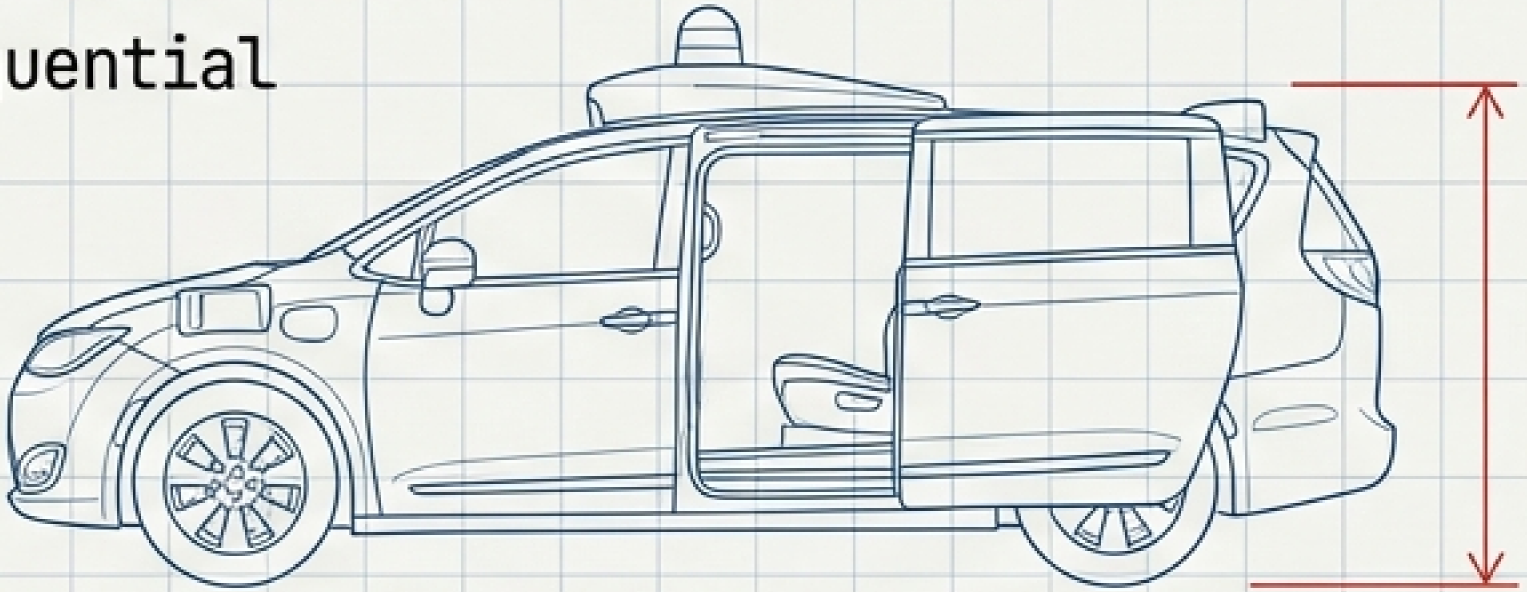
Stochastic implies probabilities; Nondeterministic implies possibilities.

Time Horizon

Episodic vs. Sequential



Episodic
(Atomic attomic actions,
no long-term consequence)

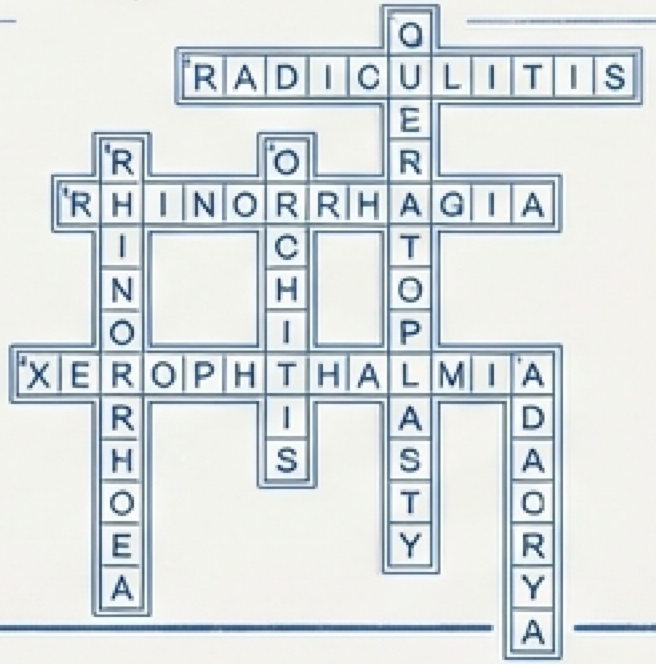


Sequential (Waymo Now Display
(Current decision affects all future decisions)

The Environment: Dynamics and Granularity

Change Over Time

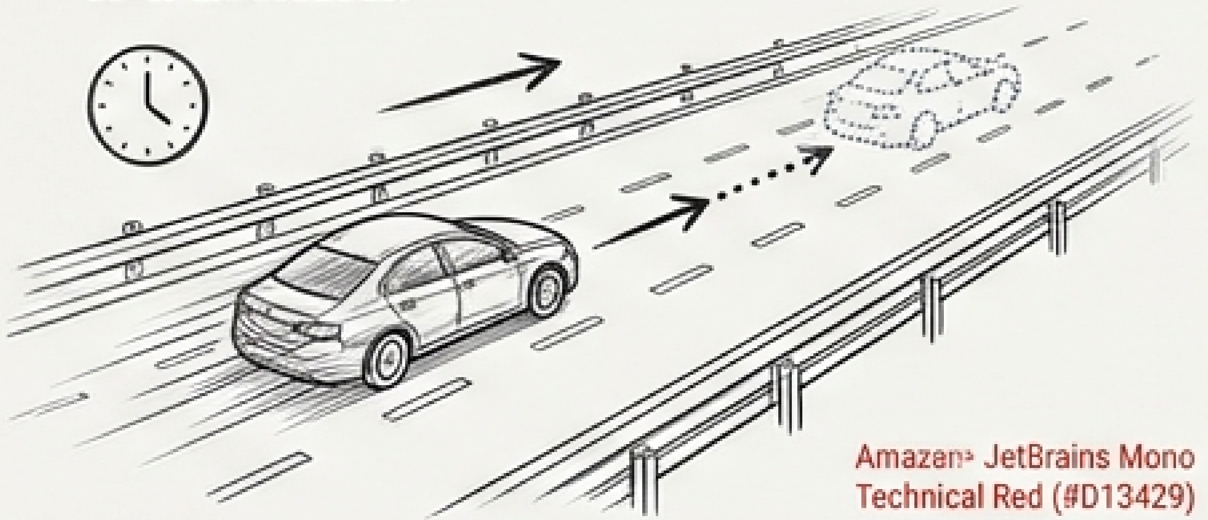
Static: The world waits for you to think (e.g., Crossword).



Static vs. Dynamic

Time Progression

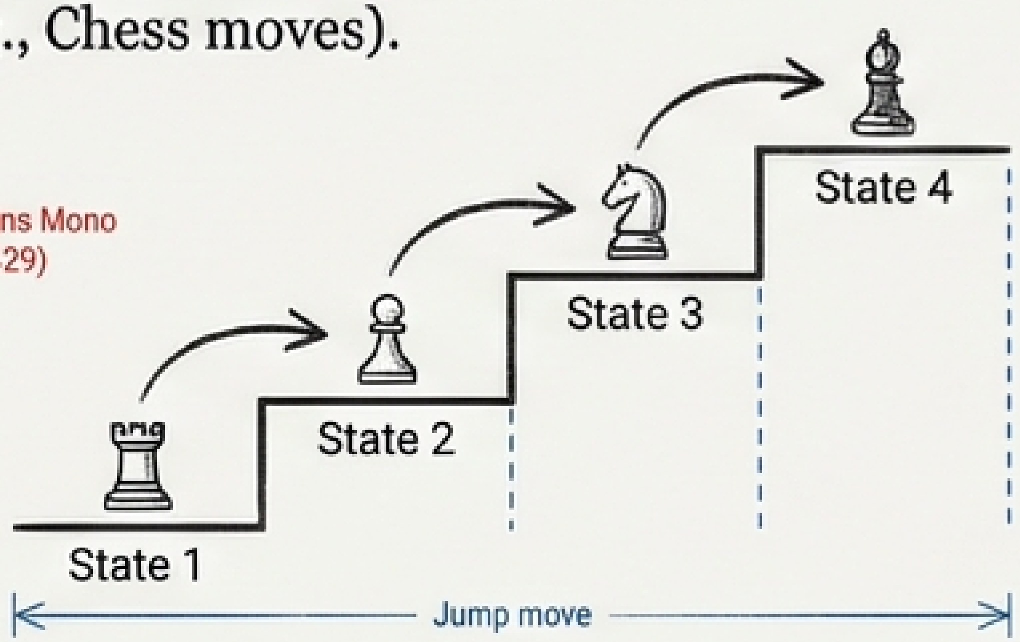
Dynamic: The environment changes while you deliberate (e.g., Driving). Failing to act is a decision.



State Texture

Discrete: Finite distinct states (e.g., Chess moves).

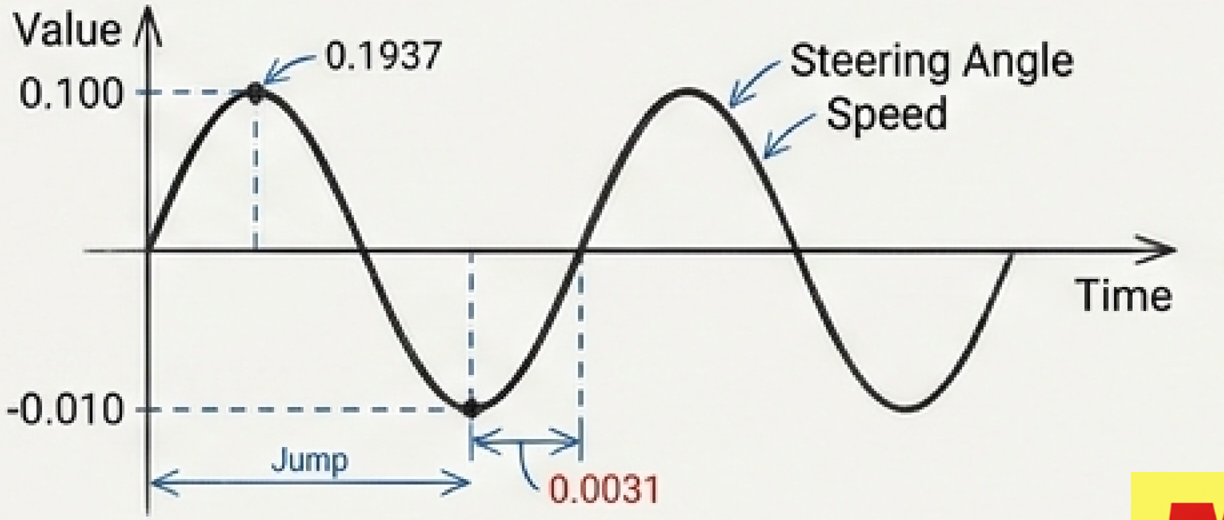
JetBrains Mono
(#D13429)



Discrete vs. Continuous

JetBrains Mono
(#D13429)

Continuous: Smoothly varying values (e.g., Steering angles, speed).



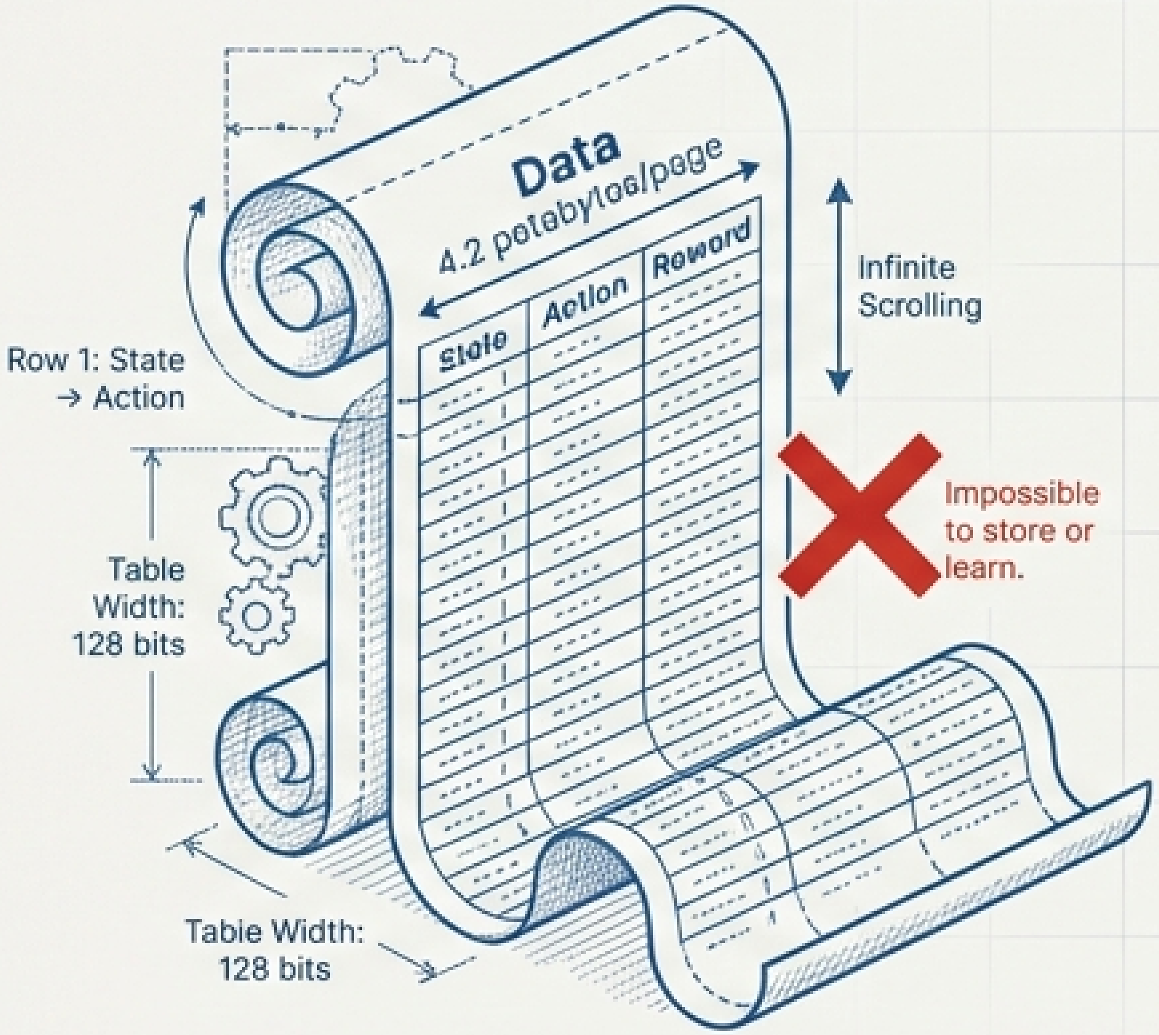
The Complexity Spectrum

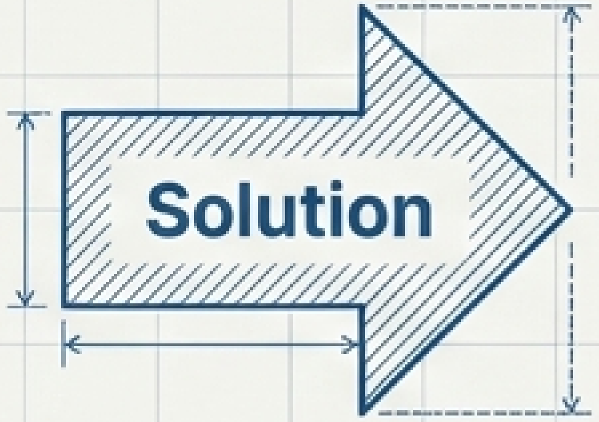
Task Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Solitaire	Fully	Deterministic	Sequential	Static	Discrete	Single
Chess	Fully	Strategic	Sequential	Semi	Discrete	Multi
Taxi Driving	Partially	Stochastic	Sequential	Dynamic	Continuous	Multi

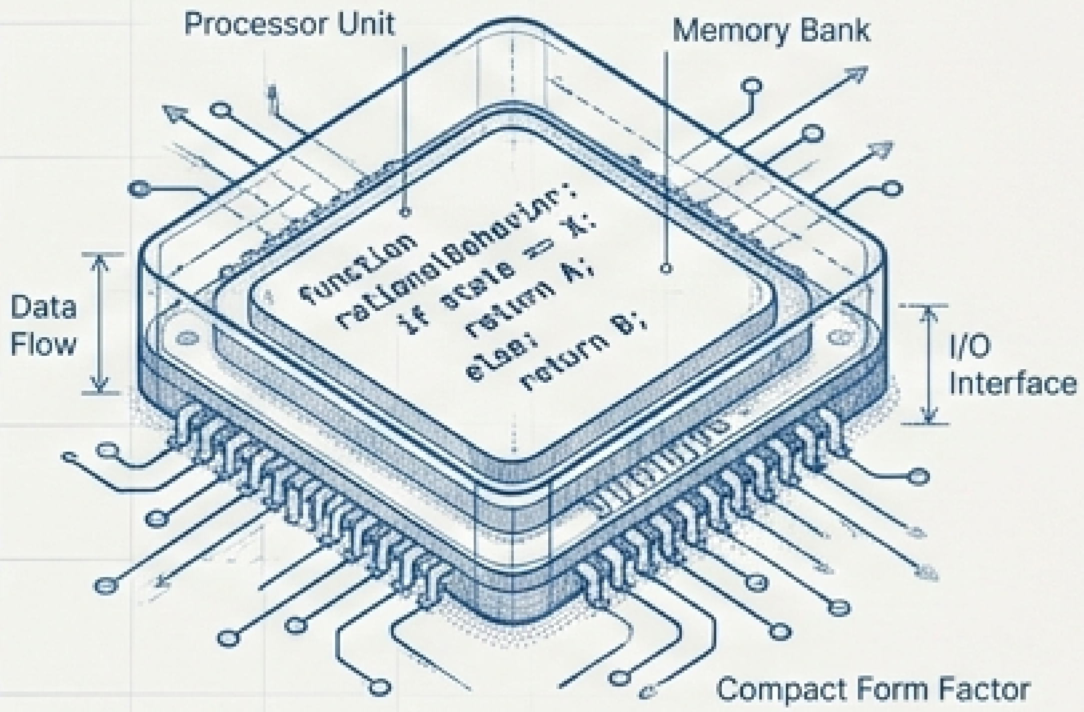
- The "Hardest Case" (Taxi) is Partially Observable, Stochastic, Dynamic, and Continuous. This complexity profile demands the most sophisticated Agent Architectures.

Blueprint: rt (#D13429)

Inside the Agent: From Math to Code







The Lookup Table

The Flaw: A table-driven taxi agent would require 10^{6,000,000,000,000} entries. Impossible to store or learn.

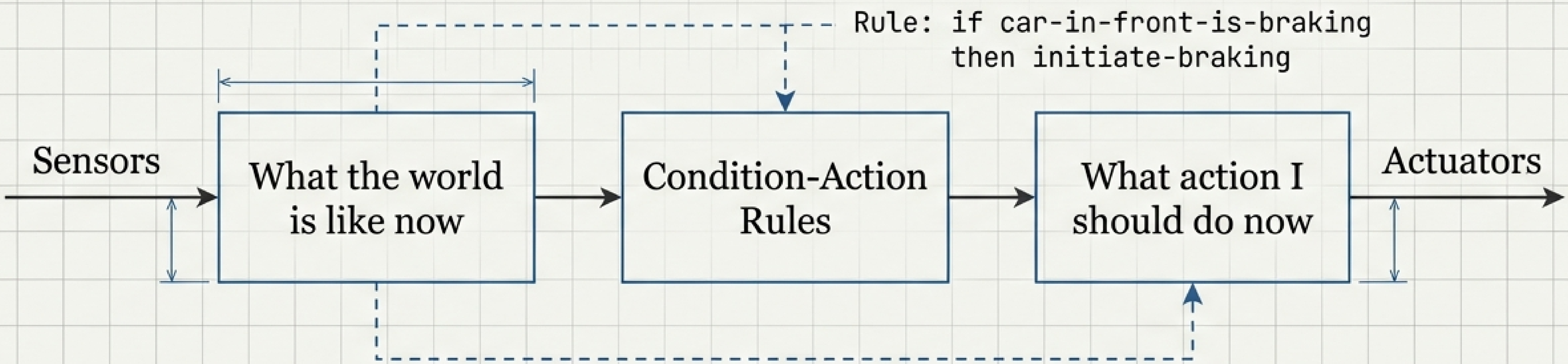
The Agent Program

Goal: Produce rational behavior from a finite program.

Agent = Architecture + Program

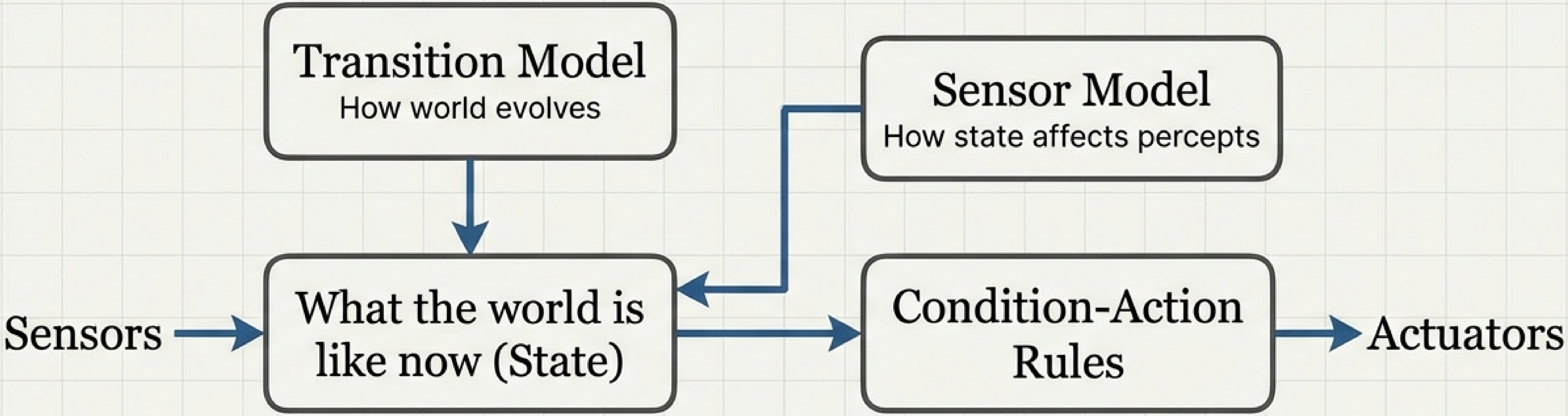
Blueprint: br (#D13429)

Simple Reflex Agents



Limitation:
Only functions correctly in Fully Observable environments.
Vulnerable to infinite loops if sensors are blocked.

Model-Based Reflex Agents

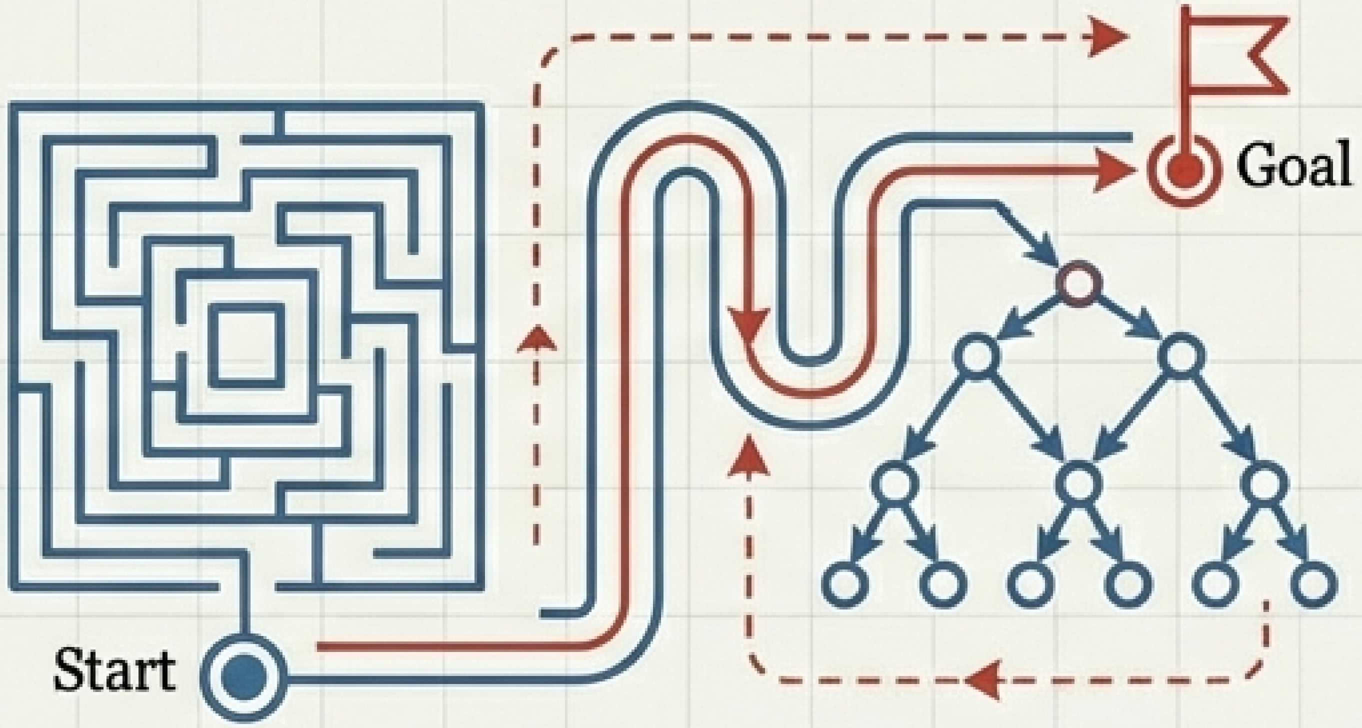


The Upgrade: Handles **Partial Observability** by maintaining an internal state.

1. **Transition Model**: “If I turn the wheel, the car goes right”.
2. **Sensor Model** : “Red pixels in center = Brake lights”.

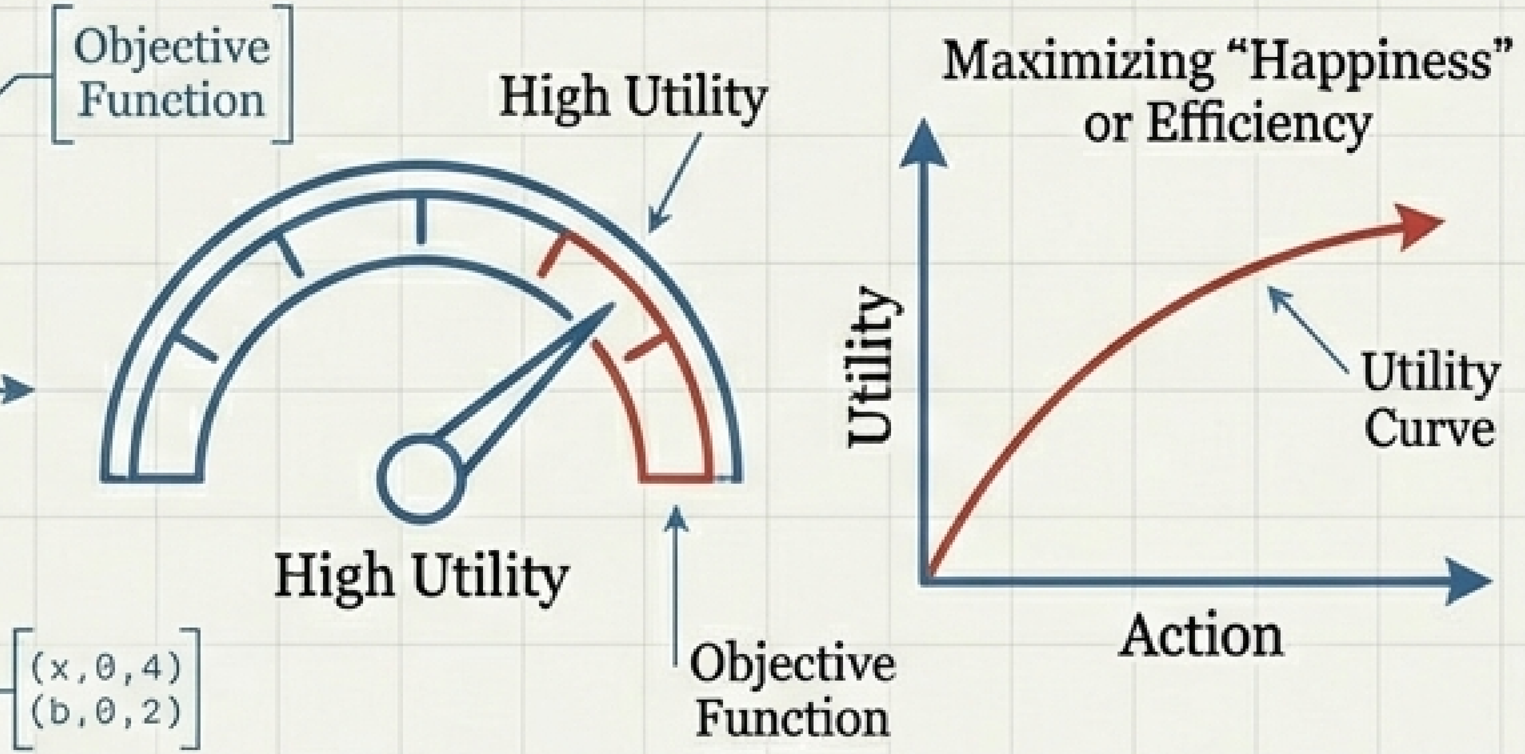
Goal-Based and Utility-Based Agents

The **Planner** (Goal-Based)



Uses **Search** and **Planning**. Decisions are based on achieving a specific destination (Goal). Flexible behavior.

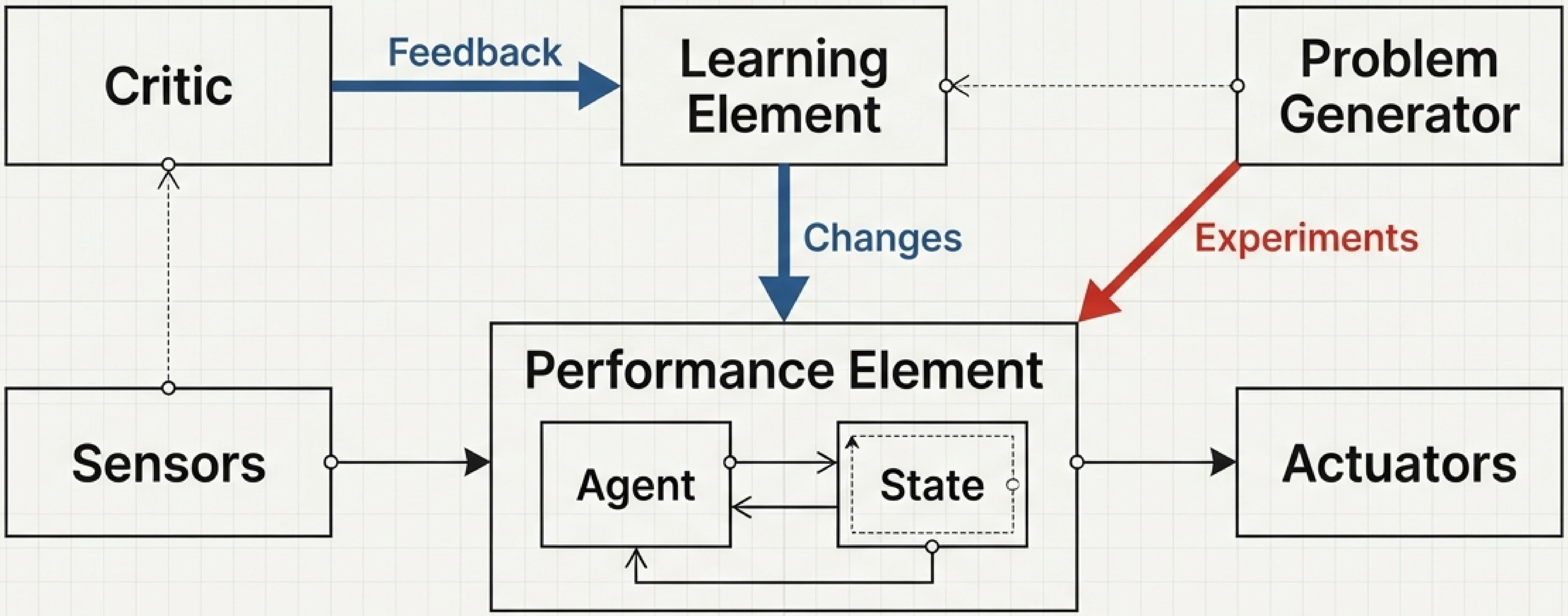
The **Optimizer** (Utility-Based)



Uses **Utility Functions**. Decisions are based on maximizing "happiness" or efficiency. Allows trade-offs (e.g., Speed vs. Safety).

Rationality = Maximizing Expected Utility.

Learning Agents

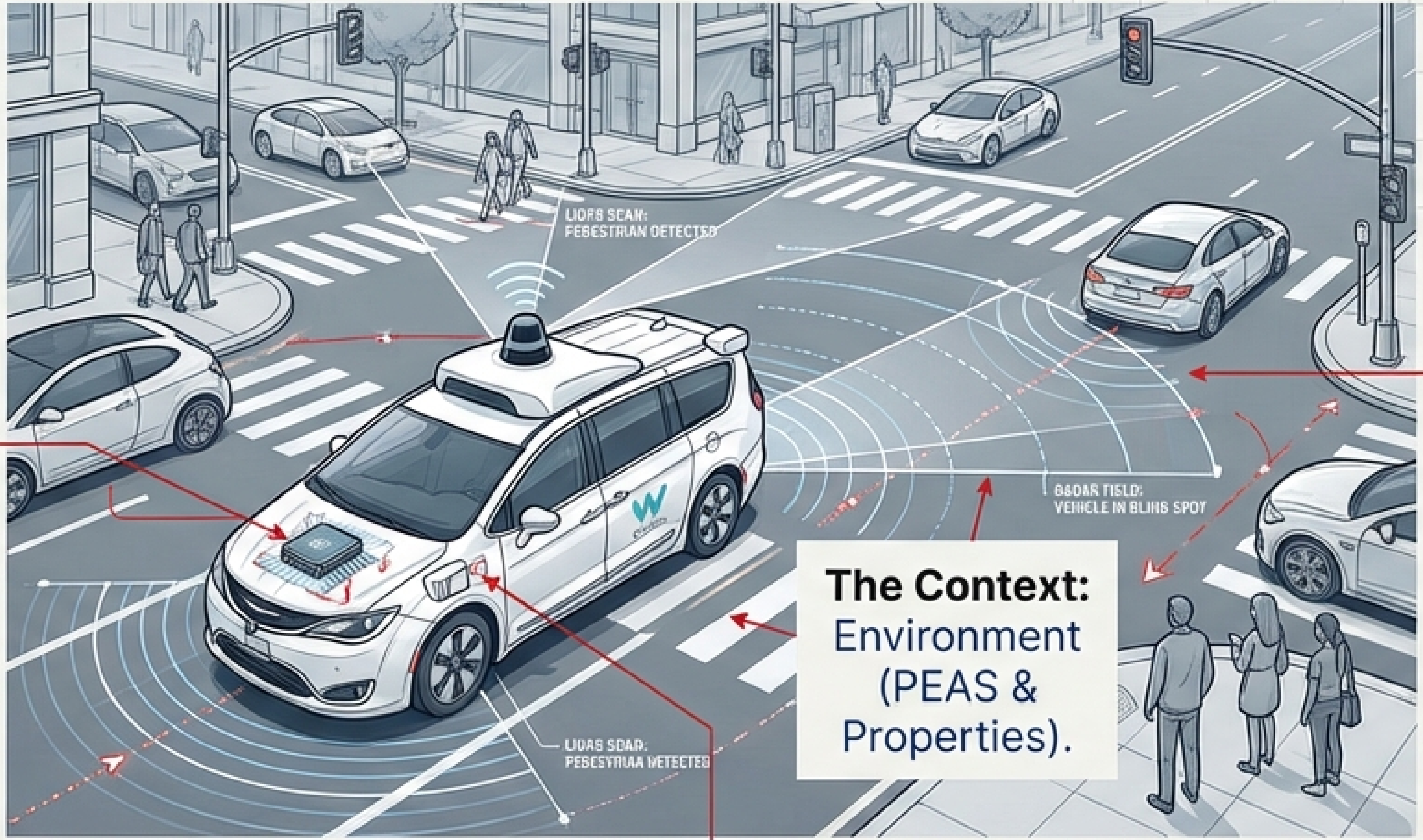


How an agent evolves from ignorant to expert. The **Critic** evaluates performance, and the **Learning Element** rewrites the agent's own code to improve future results.

Reference NO.: 834
(w:800 +37 annotations)

The Anatomy of Autonomy

The Goal:
Rationality (Maximize Expected Performance).



The Context:
Environment (PEAS & Properties).

The Solution:
Agent Architecture.

Simple Environments → **Reflex Agents**
Complex Environments (Dynamic, Stochastic) → **Learning, Utility-Based Agents**

Referen