

# Actor Before Agent

Persistent identities, temporary inference, and evidence-bound World  
Patches

Hayden Lindley · Glyphd Labs

2026-07-27

## Contents

|                                     |          |
|-------------------------------------|----------|
| <b>Actor Before Agent</b>           | <b>2</b> |
| Abstract                            | 2        |
| 1. The inversion                    | 2        |
| 2. Actor Capsule                    | 2        |
| Identity core                       | 2        |
| Mutable state                       | 3        |
| Autonomy policy                     | 3        |
| 3. Runtime state machine            | 3        |
| Dormant                             | 4        |
| Eligible                            | 4        |
| Selected                            | 4        |
| Active                              | 4        |
| Committed                           | 4        |
| Cooldown                            | 4        |
| 4. Deterministic selection          | 4        |
| 5. Process types                    | 4        |
| 6. Structured goals                 | 5        |
| 7. Limited knowledge                | 5        |
| 8. World Patch                      | 6        |
| 9. Identity stability and change    | 6        |
| 10. Multiplex World Pass            | 7        |
| 11. Background actors and promotion | 7        |
| 12. Application beyond games        | 7        |
| Personal AI representation          | 7        |
| Organizations                       | 7        |
| Social systems                      | 8        |
| Multi-agent project work            | 8        |
| 13. Evaluation                      | 8        |
| Identity                            | 8        |
| Epistemics                          | 8        |
| Operations                          | 8        |
| Human judgment                      | 8        |
| 14. Security and misuse             | 8        |
| 15. Evidence state                  | 9        |
| 16. Limitations                     | 9        |
| 17. Conclusion                      | 9        |
| Architecture illustration briefs    | 9        |

|                                             |    |
|---------------------------------------------|----|
| Source register . . . . .                   | 10 |
| Public-disclosure and IP boundary . . . . . | 10 |

# Actor Before Agent

## Abstract

Most “agent” architectures begin with a running model and ask how to give it memory, personality, tools, and autonomy. That order makes identity dependent on continuous inference and encourages systems to confuse temporary computation with the person, institution, or character being represented.

The Actor-Agent Runtime reverses the order. An **Actor** is a durable, versioned identity and state capsule that exists when no model is running. An **Agent Process** is a temporary, bounded computation that wakes because the actor is relevant, receives only permitted knowledge, proposes a reaction or plan, and returns a structured World Patch. Deterministic validators check feasibility, knowledge, authority, resource, identity, and policy constraints before any accepted state change increments the actor revision.

The architecture supports persistent people, organizations, game citizens, social proxies, and institutional actors without pretending that thousands of entities are continuously thinking through an LLM. It enables multiplex inference, bounded autonomy, explainable background plans, local-model substitution, replay, and identity continuity.

The source model was developed for the game MAIN CHARACTER, but the distinction generalizes to any system in which a durable person-shaped or institution-shaped state must outlive an inference call.

## 1. The inversion

The conventional pattern is:

agent = model + prompt + memory + tools

The Actor-Agent pattern is:

actor = durable identity + state + policy + lineage

agent process = temporary computation over one bounded situation

This inversion changes the system's economics and semantics.

A persistent actor can remain coherent for weeks while consuming no inference. A scheduler can activate only the actors affected by an event. Several actors can share one multiplex reasoning pass. Different model policies can operate the same actor without replacing identity. Failures can be replayed from actor state, context packet, model reference, and World Patch.

The model is not the person. It is one possible drive process.

## 2. Actor Capsule

An Actor Capsule contains three major layers.

### Identity core

Fields that define durable authored identity:

- immutable actor ID;
- name and pronouns;

- origin and social role;
- public self-description;
- private pressure;
- values and value priorities;
- personality traits;
- style and visual tags;
- speech rhythm and vocabulary constraints;
- content boundaries;
- authored canon facts.

Some identity fields may evolve only through explicit change events. Others are immutable.

### **Mutable state**

Fields that change through events:

- location and schedule;
- mood dimensions;
- resources;
- public reputation;
- attention and stress;
- active goals;
- commitments;
- opportunities;
- relationship edge references;
- memories;
- beliefs and misconceptions;
- promises, debts, secrets, and open loops;
- last activity and cooldown.

### **Autonomy policy**

The bounded authority of the runtime:

- initiative level;
- risk tolerance;
- loyalty bias;
- secrecy preference;
- planning horizon;
- allowed action categories;
- actions requiring confirmation;
- daily or scene compute budget;
- prohibited targets or methods.

The Capsule is versioned and inspectable. It is not one giant prose prompt.

## **3. Runtime state machine**

Actors move through:

```
DORMANT
→ ELIGIBLE
→ SELECTED
→ ACTIVE
→ COMMITTED
→ COOLDOWN
```

### **Dormant**

Only data and deterministic schedules continue. No model call is required.

### **Eligible**

An event, goal, relationship, deadline, or player action makes the actor a candidate.

### **Selected**

The scheduler allocates context, tool, and inference budgets.

### **Active**

Rules or a model-assisted pass produce a proposal.

### **Committed**

A validated patch is applied, events are appended, and the actor revision increments.

### **Cooldown**

The actor cannot consume repeated inference until new evidence, a threshold event, or a deadline justifies activation.

The state machine prevents popular or noisy actors from monopolizing compute.

## **4. Deterministic selection**

Eligibility is computed before an LLM call. One reference score is:

```
score =  
    proximity  
  + player_relevance  
  + goal_urgency  
  + relationship_intensity  
  + event_salience  
  + arc_value  
  + deadline_pressure  
  - cooldown  
  - recent_compute
```

Weights are configuration, not hidden model preference. A threshold selects candidates; a hard cap preserves fairness and cost.

The score is not a truth claim about importance. It is a scheduler policy that should be instrumented, configurable, and benchmarked.

## **5. Process types**

The runtime distinguishes process roles.

- **Reactive:** seconds to one time block; responds to a witnessed event.
- **Planning:** chooses short plans for important goals.
- **Relationship:** updates trust, obligations, boundaries, and desired contact.

- **Social propagation:** frames how a rumor or interpretation travels through an allowed graph.
- **Institutional:** reasons from bylaws, dependencies, constituency, and strategy rather than human emotion.
- **Director:** selects existing tensions to converge; cannot invent arbitrary facts or directly mutate state.
- **Memory compiler:** converts event history into episodic summaries, beliefs, relationship evidence, and open loops.

A process has an explicit job. It is not an unconstrained “autonomous agent.”

## 6. Structured goals

A goal includes:

```
target_state
motivation
urgency
deadline
visibility
acceptable_methods
forbidden_methods
dependencies
success_conditions
failure_conditions
plan_steps
last_evidence_update
```

The model may propose a plan. The runtime checks schedule, resources, knowledge, permissions, and causal feasibility.

This separation allows believable failure. An actor may desire an impossible outcome. The goal persists while the proposed plan fails validation.

## 7. Limited knowledge

An agent packet does not receive the whole world. It receives:

- actor canon;
- actor beliefs;
- relevant memories;
- known relationships;
- present scene facts;
- public facts;
- allowed target IDs;
- explicit unknowns.

Facts and beliefs remain distinct. An actor can act on false belief if that belief exists in state. The model cannot use world truth that the actor has never observed merely because the runtime knows it.

A useful typed distinction is:

```
FACT
BELIEF
RUMOR
INFERENCE
```

DESIRE  
LIE  
UNKNOWN

This produces both narrative coherence and security. The same mechanism limits a business agent to the information authorized for its principal.

## 8. World Patch

The process cannot edit the Capsule directly. It returns:

```
WorldPatch {  
  actor_revision_base  
  goal_operations[]  
  belief_operations[]  
  memory_candidates[]  
  mood_deltas[]  
  relationship_deltas[]  
  schedule_proposals[]  
  planned_actions[]  
  presentation_text[]  
  evidence_refs[]  
  confidence  
  unresolved_questions[]  
}
```

Validation checks:

- base revision is current;
- target IDs are allowed;
- actor knew the cited facts;
- deltas remain within bounds;
- resources and schedule permit the action;
- identity constraints are not violated;
- authority permits the operation;
- evidence exists where required;
- simultaneous patches do not conflict.

Accepted operations become events and increment the revision. Rejected operations remain inspectable as proposals.

## 9. Identity stability and change

A persistent actor should not become a new person because one model call sampled a different tone.

Stability mechanisms include:

- immutable canon fields;
- value priority constraints;
- style exemplars;
- contradiction checks;
- bounded trait deltas;
- memory retrieval;
- explicit major-change events.

Sudden reversal requires represented cause: betrayal, conversion, pressure, deception, trauma, new evidence, or deliberate performance.

This does not freeze character growth. It makes growth causal.

## 10. Multiplex World Pass

Several actors can be resolved in one model call when they share a situation.

The packet includes:

- shared scene facts;
- each actor's bounded state;
- relationship edges among included actors;
- separate allowed targets;
- separate output slots;
- conflict rules.

The response contains one patch proposal per actor plus shared scene consequences. Validators apply each independently.

Multiplexing can reduce inference cost and improve consistency because the model sees the same event once. It can also create cross-actor leakage, so the packet compiler must prevent one actor's private state from entering another actor's reasoning slot unless the model is acting as a trusted simulation engine and output validators maintain epistemic separation.

## 11. Background actors and promotion

Not every actor needs a full Capsule. A Tier C background actor may initially have a role archetype and minimal state.

When the actor becomes important, promotion compiles:

- authored archetype;
- recorded encounters;
- relationship edges;
- location and culture;
- existing role;
- constrained generated details.

Generated details cannot contradict observed history. Promotion is a patch with provenance, not a retroactive rewrite.

## 12. Application beyond games

The architecture maps naturally to:

### Personal AI representation

A durable user-controlled Actor Capsule defines public profile, private state, availability, boundaries, and delegation. Temporary agents answer, schedule, or coordinate within leases.

### Organizations

The actor is an institution with bylaws, constituency, budget, commitments, and authority. The process reasons from institutional state rather than simulated emotion.

## **Social systems**

A user can be represented by static state when unavailable, a live human when present, or a bounded AI proxy. The proxy does not become the human's identity.

## **Multi-agent project work**

Project roles persist while specialized processes wake for research, coding, review, or coordination.

# **13. Evaluation**

## **Identity**

- contradiction rate;
- unexplained trait reversal;
- style drift;
- memory consistency;
- revision lineage.

## **Epistemics**

- use of unavailable facts;
- fact/belief confusion;
- private-state leakage;
- rumor propagation correctness.

## **Operations**

- unauthorized mutation;
- patch validity;
- conflict handling;
- replay;
- rollback;
- cooldown enforcement;
- scheduler fairness;
- cost per meaningful event.

## **Human judgment**

- perceived continuity;
- causal believability;
- relationship coherence;
- inspectability;
- willingness to delegate.

A baseline should compare always-running chat agents, summary-memory agents, rule-only actors, and Actor-Agent capsules under the same scenarios.

# **14. Security and misuse**

A person-shaped capsule can be used to impersonate, manipulate, or surveil. The public architecture requires:

- consent for real-person representation;

- explicit disclosure when an AI proxy speaks;
- bounded audience and duration;
- revocable delegation;
- no hidden expansion of authority;
- no silent training on private actor state;
- separation of public and private fields;
- logs and receipts;
- safe refusal to infer sensitive attributes not represented in state.

The model must not produce private facts for presentation simply because they were present in the internal packet.

## 15. Evidence state

The Actor-Agent model is specified in detail, including Capsule fields, state machine, process taxonomy, eligibility score, goal schema, limited knowledge, World Patch, identity stability, promotion, and mechanical consequences.

A complete production runtime receipt—including persistent storage, scheduler, validator, multiplex pass, conflict resolution, and long-duration identity benchmarks—remains outstanding. The public state is therefore SPECIFIED, with parts implementable as deterministic demos.

## 16. Limitations

The model can still generate shallow or stereotyped behavior. Structured state does not guarantee psychologically rich actors. Capsule design can become too rigid or too large. Multiplex inference can entangle voices. Deterministic schedulers can encode designer bias. A believable identity may require more nuanced memory and causal modeling than simple bounded traits.

For real people, the ethical burden is higher than for fictional characters. A user-controlled proxy requires consent, correction, visibility, and revocation mechanisms.

## 17. Conclusion

An agent is an event. An actor is a continuing identity.

By placing the durable object first, the system can reason only when needed, change models without changing the person, preserve false beliefs without corrupting world truth, and require every state change to pass through a bounded patch.

The canonical sentence is:

**The actor is the memory-bearing identity. The agent is the transient drive.**

## Architecture illustration briefs

- **P06-F01 — Actor before agent:** Persistent Actor Capsule remains across time while short-lived reactive, planning, relationship, and memory processes attach and disappear.
- **P06-F02 — Actor lifecycle:** Dormant → Eligible → Selected → Active → Committed → Cooldown, with deterministic eligibility inputs.
- **P06-F03 — World Patch validation:** Agent proposal enters revision, knowledge, feasibility, authority, identity, and conflict validators before actor/world events are appended.
- **P06-F04 — Multiplex World Pass:** One shared situation feeds bounded actor slots and produces separately validated patches.

## Source register

- **P06-S01** — Actor-Agent Model (2026-07-27). 01\_ACTOR\_AGENT\_MODEL.md
- **P06-S02** — MAIN CHARACTER system specification corpus (2026-07-26/27). Main Character project folder
- **P06-S03** — GlyphCAS Design Report (2026-07-24). How\_we\_can\_make\_CAS\_lightning\_fast\_for\_any\_LL
- **P06-S04** — MCE-1 System Contract (2026-07-25). MCE-1-SYSTEM-CONTRACT-v0.1.md

## Public-disclosure and IP boundary

This paper publishes the public-safe runtime distinction and generalized state contracts. It excludes private character canon, unreleased game narrative, real-person proxy data, proprietary scheduling weights, and confidential implementation details. No claim is made that a production-scale city simulation has been completed.