

CAPTAIN CORE

Alignment Through Architectural Constraint

Document Type	Working Paper 2026
Version	v2.0 · July 2026
Author	Alix Keller
Organization	Captain Collective

Overview

Artificial intelligence is rapidly moving beyond chat interfaces into environments where it is expected to coordinate real work, retain context, and participate in decisions that unfold over months or years. These systems no longer serve a single user through isolated interactions. They increasingly operate within families, healthcare networks, schools, workplaces, financial systems, and other environments where multiple people contribute information, exercise different levels of authority, and depend on the same evolving body of context.

This shift fundamentally changes the alignment problem.

Most contemporary alignment research evaluates models responding to individual prompts. While this work remains essential, it provides only a partial view of how AI systems behave once they become persistent participants in human systems. Long-lived deployments introduce questions that cannot be answered through prompt-response evaluation alone. Context accumulates. Permissions change. Relationships evolve. Authority transfers between people. Information that was once relevant may later become misleading or inappropriate. These are not simply model problems—they are systems problems.

This paper argues that alignment should increasingly be understood as an architectural property rather than solely a property of model behavior. Captain Core is presented as a reference architecture for exploring this emerging class of systems.

The objective is not to prescribe a definitive solution, but to encourage a shift in perspective. As AI becomes embedded within everyday life, alignment research must expand beyond optimizing model outputs to understanding how architecture, governance, and evolving human relationships shape system behavior over time.

"The next generation of AI will be defined by what it is allowed to know."

01 — Introduction: The Unit of Deployment Has Changed

Over the last decade, artificial intelligence has advanced primarily through increasingly capable models. Improvements in reasoning, multimodal understanding, and larger context windows have dramatically expanded what these systems can accomplish. Unsurprisingly, much of AI safety and alignment research has evolved alongside this trajectory, focusing on how individual models reason, behave, and produce outputs.

These efforts have produced important progress, but they also reflect an assumption that is beginning to change.

Historically, most AI interactions were short-lived. A user asked a question, the model generated a response, and the interaction concluded. Even when conversational memory was introduced, the surrounding system remained relatively simple. The model served one person, within one session, under a relatively stable set of assumptions.

That is no longer the direction in which AI is evolving.

Increasingly, AI systems are expected to coordinate appointments, summarize meetings, manage household logistics, support caregivers, organize financial information, assist with education, and participate in long-running workflows that span months or years. These systems are no longer isolated reasoning engines. They become participants within living human systems whose structure changes continuously.

A family is not a single user. Neither is a care network, a school, a business, or a community organization.

Each contains multiple participants with different responsibilities, different perspectives, and different authority over the same underlying context. Parents gradually transfer responsibility to children. Physicians gain temporary access to medical information. Financial advisors participate in limited domains. Caregivers enter and leave. Relationships evolve, permissions change, and the context that informs good decision-making changes with them.

Current alignment methodologies were not primarily designed to evaluate systems operating under these conditions. The challenge is no longer limited to producing correct responses. It extends to determining whether a system continues making appropriate decisions as the environment itself changes.

As AI transitions from isolated interactions to persistent participation in human systems, alignment increasingly becomes a systems engineering problem.

Examples of Persistent Human Systems

Families Multiple generations coordinating care, schedules, finances, and major life decisions across decades.	Care Networks Healthcare providers, family caregivers, and support services sharing responsibility for patient outcomes.
Schools Teachers, administrators, parents, and students collaborating on education over multiple years.	Workplaces Teams, departments, and organizations coordinating work with evolving roles and responsibilities.
Community Organizations Volunteers, staff, beneficiaries, and donors participating in shared missions with changing involvement.	

02 — When the Unit of Deployment Changes

Most alignment research evaluates models because models have historically been the primary unit of innovation. They have also been the primary unit of deployment. For many applications, this remains appropriate. A model that summarizes a document or answers a technical question can often be evaluated independently from the broader environment in which it operates.

Persistent systems are different.

Once an AI begins coordinating activities over time, the surrounding architecture becomes inseparable from the quality of its decisions. The model no longer reasons over a single prompt. It reasons over an evolving body of context contributed by multiple people operating under changing relationships and permissions.

Consider a family coordinating long-term care for an aging parent. One sibling initially manages appointments and becomes the primary point of contact with physicians. Over several months, the system accumulates medical history, scheduling preferences, treatment plans, and operational knowledge through that individual. Then responsibilities shift.

Another sibling assumes coordination. Additional healthcare providers enter. Financial advisors participate in limited domains. Family members rotate in and out. The context that informed good decisions three months ago may no longer reflect the current reality.

From the perspective of today's AI systems, this transition is remarkably difficult to navigate. Which assumptions should continue influencing future decisions? Which permissions should transfer automatically? Which information remains relevant, and which should become dormant rather than deleted?

These are not failures of reasoning. They are failures of governance.

Current evaluation methods offer relatively little guidance for measuring how systems behave as context evolves through multiple actors over time. Most benchmarks remain focused on correctness, factual accuracy,

reasoning quality, or immediate safety. They rarely evaluate how persistent systems manage changing authority, decaying context, or evolving relationships across months or years of operation.

This paper refers to these challenges collectively as multi-actor contextual drift: the gradual divergence between a system's accumulated representation of reality and the reality it is intended to support.

Unlike traditional distribution shift, contextual drift emerges through the normal operation of the system itself. Information accumulates under one set of assumptions, relationships change, permissions evolve, and yesterday's context quietly becomes today's source of error. Understanding, measuring, and ultimately constraining this form of drift represents one of the central research questions explored throughout the remainder of this paper.

Characteristics of Persistent Systems

Multiple Participants Several people contribute to and rely upon the same evolving system.	Evolving Relationships Responsibilities and authority naturally change over time.
Shared Context Information is accumulated collectively rather than individually.	Longitudinal Decisions Decisions are informed by months or years of evolving context rather than isolated prompts.

"How does a system know what it should still remember when the people involved have changed? This is the central challenge of persistent AI: context must evolve as human relationships evolve."

03 — Alignment Through Architectural Constraint

If persistent AI systems introduce new failure modes, the natural question becomes where those failures originate. Conventional approaches often assume that undesirable behavior can be corrected by improving the model itself through additional training, fine-tuning, or output filtering. Those techniques remain valuable, but they address only one layer of a much larger system.

As AI becomes embedded within long-lived human environments, the architecture surrounding the model begins to shape behavior as much as the model itself.

The model can only reason over information it is permitted to access. It can only retain the context the surrounding system allows it to remember. It can only coordinate people through the identities, permissions, and governance structures that define those relationships. Long before a model generates a response, architectural decisions have already constrained what the model can know and how it is allowed to reason.

Consider a physician reviewing a patient's treatment history. The model does not require unrestricted access to every conversation, financial record, family disagreement, or personal journal entry associated with that individual. It requires the subset of information relevant to the decision it is being asked to support. Determining that subset is not a modeling problem. It is an architectural one.

Much of today's AI infrastructure implicitly assumes that broader access produces better intelligence. There are practical reasons for this assumption. Larger context windows generally improve performance, retrieval

systems benefit from additional information, and persistent memory increases personalization. Yet these advantages introduce an opposing engineering challenge: the more information a system continuously retains, the greater the responsibility required to govern that information appropriately.

Architecture therefore becomes an alignment mechanism.

Instead of asking how a model can produce safer outputs after receiving unrestricted context, we begin asking a different set of questions. How much information should the model receive in the first place? Who determines that scope? Can the system demonstrate why particular information was available during a decision? Can permissions evolve without retraining the model? Can context expire naturally rather than accumulating indefinitely?

These questions shift the emphasis away from correcting outputs toward shaping the computational environment that produces those outputs.

This perspective can be summarized as alignment through architectural constraint. Rather than assuming every safeguard should occur after inference, architectural constraints influence what inference is possible. Identity determines perspective. Permissions determine visibility. Execution environments constrain computation. Verification mechanisms provide accountability. Together they reduce the number of inappropriate states the system can enter before a response is ever generated.

Figure 3. Alignment Through Architectural Constraint

Traditional Alignment	Architectural Alignment
Better Model → Safer Outputs	Identity determines perspective
Output Filtering	Permissions determine visibility
Governance occurs after inference	Execution environments constrain computation
	Architecture constrains computation before inference

"Architecture determines what intelligence is allowed to know before it determines what intelligence is allowed to say."

04 — A Reference Architecture

If alignment increasingly emerges from system architecture, the next question is not which technologies should be adopted, but which architectural responsibilities every persistent AI system must eventually solve.

Technologies will evolve. The responsibilities themselves are likely to remain remarkably stable.

Captain Core is presented here as one reference architecture for exploring these responsibilities. It should not be interpreted as a prescriptive implementation or a fixed technology stack. Rather, it represents an architectural framework for studying how persistent AI systems can coordinate sensitive human environments while operating under explicit computational constraints.

The architecture is organized around five foundational responsibilities that converge to create an environment in which the model operates within explicit architectural boundaries rather than unrestricted access to accumulated data.

Captain Core System Architecture — alignment through constraint, not just output control

01	Identity Establishes perspective rather than simply authenticating users. Identity determines not only who someone is, but how the system should interpret their role within a broader network of relationships. It should remain portable rather than platform-dependent.
02	Authority Defines evolving responsibility rather than static permissions. Authority changes continuously as people delegate, transfer, or expire in their roles. A robust architecture supports temporal authority without requiring fundamental changes to the underlying model.
03	Context Determines what information remains appropriate for a specific decision. Human context is neither static nor universally applicable. Persistent AI should optimize for context lifecycle management rather than unlimited memory.
04	Execution Constrains computation before inference rather than filtering outputs afterward. A system cannot infer from information it never receives. Computation occurs within explicitly bounded environments proportional to the decision being made.
05	Selective Disclosure Ensures every participant receives only the information required for their role. The objective is proportional disclosure rather than secrecy. Intelligence should expand the quality of decisions rather than the quantity of information exposed.

"If this architecture is correct, how would we evaluate whether it actually works?"

05 — Evaluating Persistent AI Systems

Existing AI benchmarks remain valuable, but they measure only part of the engineering problem. As AI systems transition from isolated interactions to persistent participation in human environments, evaluation must expand beyond model performance to encompass system behavior over time.

This shift does not diminish the importance of current evaluation methodologies. Instead, it acknowledges that persistent systems introduce entirely new dimensions of evaluation that traditional benchmarks were never designed to capture.

The question is not whether existing benchmarks are wrong, but whether they are sufficient for systems that coordinate multiple people, accumulate context over months or years, and adapt to evolving relationships and

permissions.

Figure 5. Traditional vs Persistent AI Evaluation

Traditional AI Evaluation	Persistent AI Evaluation
Prompt Safety	Longitudinal System Behavior
Factual Accuracy	Context Integrity
Tool Use	Permission Fidelity
Reasoning Quality	Coordination Quality
Model Alignment	Computational Governance
Memory Retrieval	Context Lifecycle Management

Context Integrity

Does the system maintain an accurate understanding of evolving human context? As relationships change and new participants enter, the system must distinguish between information that remains relevant and information that has become misleading.

Permission Fidelity

Do architectural permissions consistently reflect changing authority? When responsibility transfers between people, the system must adapt without requiring manual reconfiguration or retraining.

Coordination Quality

Does the system improve decision-making across multiple participants? The measure of success shifts from individual task completion to collective outcomes achieved through distributed collaboration.

Computational Governance

Can every meaningful decision be traced to the identity, permissions, context, and execution environment that produced it? Accountability requires more than output logs—it requires architectural transparency.

Context Lifecycle Management

Does the system appropriately validate, archive, retire, and intentionally forget context over time? Forgetting is not a failure of memory—it is evidence that the system is adapting to changing reality.

"The systems we build determine the behaviors we are able to measure."

06 — Computational Governance

As AI systems become increasingly capable, governance gradually shifts from organizational policy into software architecture. This transition represents more than a technical evolution—it reflects a fundamental change in how human intent is expressed and enforced within computational systems.

Historically, governance has existed outside the system. Organizations establish policies describing who should have access to information, when decisions require approval, how responsibilities are delegated, and what constitutes appropriate use. Software has generally enforced these rules only after they have been defined elsewhere.

Persistent AI systems begin changing that relationship.

Identity, permissions, execution environments, and context management no longer function solely as supporting infrastructure. They become the mechanisms through which governance is expressed computationally. The distinction is important. Governance is no longer simply a question of policy—it becomes a question of implementation.

Whether an AI system is capable of accessing information, retaining context, delegating authority, or sharing knowledge increasingly depends upon architectural decisions rather than organizational documentation. This shift creates an opportunity to rethink governance itself.

Rather than assuming every policy must be interpreted after the fact, architectural constraints can make entire classes of inappropriate behavior computationally impossible. A system cannot disclose information it never receives. It cannot reason over context that has expired. It cannot exceed authority that has not been delegated.

In this sense, governance becomes preventative rather than corrective. The architecture itself establishes the boundaries within which intelligence operates. This perspective does not eliminate the need for organizational policy. Instead, it creates a stronger relationship between policy and implementation by expressing governance directly through computational constraints.

Figure 6. Governance Moves Into Architecture

Traditional Governance	Computational Governance
Organization → Policies → Software → Users Governance operates through organizational hierarchy	Human Intent → Identity → Authority → Context → Architectural Constraints → Computation → System Behavior Governance expressed directly through architecture

"Every architectural decision is ultimately a governance decision."

07 — Research Agenda

This paper does not conclude with definitive answers. Instead, it introduces a new area of systems research that will require contributions from many disciplines. The questions that follow are not unresolved problems—they are opportunities for future engineering and scientific exploration.

Persistent AI systems represent an emerging engineering discipline. Like any new field, progress will come not from solving every problem at once, but from asking the right questions and building the infrastructure needed to explore them systematically.

The research domains outlined below are intentionally broad. Each represents a direction for investigation rather than a specific technical challenge. Progress will likely require contributions from machine learning, distributed systems, cryptography, human-computer interaction, governance, security, and social science. No single discipline is sufficient.

Figure 7. Emerging Research Domains

Research Domain	Representative Questions
Context Evolution	How should context change over years of interaction without diverging from the relationships it represents?
Authority	How should responsibility transfer between participants without introducing unintended persistence or inappropriate access?
Context Lifecycle	When should systems archive, retire, or intentionally forget information? What criteria should determine these transitions?
Computational Boundaries	How much context should be available for any given decision? What information should never enter inference regardless of model capability?
Longitudinal Evaluation	How should persistent systems be measured across months or years? Which metrics best predict whether a system continues behaving appropriately as relationships evolve?
Verification	How can systems demonstrate that architectural constraints were actually enforced during computation? Can these guarantees be independently verified?

"Every answer will introduce new questions. That is how engineering disciplines mature."

08 — Toward Persistent Intelligence

This paper has introduced an architectural perspective on persistent AI systems. The questions it raises extend beyond any single implementation. They point toward a broader shift in how the field may evolve over the coming decade.

The first generation of AI systems demonstrated that machines could reason. They answered questions, summarized documents, wrote code, and generated content with increasing sophistication. These capabilities remain essential, but they represent only the beginning of a longer transformation.

The next generation will be expected to participate.

Participation changes the engineering problem. It requires systems that can coordinate families, healthcare providers, schools, workplaces, financial institutions, and communities whose relationships extend far beyond individual prompts. These environments demand architectures capable of adapting to changing authority,

evolving context, and continuous collaboration between multiple participants.

This shift suggests that the next decade of AI research may become less focused on models in isolation and more focused on the systems within which those models operate. Capability will remain important, but it will no longer be sufficient.

As AI becomes embedded within everyday life, the questions that matter most will increasingly concern governance rather than performance. How much information should a system retain? When should context expire? How should authority transfer between people? Can systems demonstrate why particular decisions were made? Who holds accountability when coordination fails?

These are not questions that models alone can answer. They require architectures that express human intent computationally through identity, permissions, context management, constrained execution, and verifiable governance.

Captain Core represents one exploration of this direction. It should be understood not as a finished architecture, but as an invitation to investigate an emerging engineering discipline whose importance will continue growing as AI becomes infrastructure rather than merely a tool.

The field is still early in this transition. The architectures that ultimately prove most durable may not yet exist. What matters is recognizing that persistent intelligence introduces fundamentally different requirements than isolated reasoning, and that meeting those requirements will demand contributions from many disciplines working together.

Figure 8. The Evolution of AI Systems

Models Isolated reasoning	Assistants Conversational memory	Agents Tool use & actions	Persistent Systems Long-term context	Coordinated Human Systems Multi-actor participation	Computational Governance Architectural constraints
-------------------------------------	--	-------------------------------------	--	---	--

From isolated reasoning to computational governance — each stage introduces new governance requirements that cannot be solved through model capability alone.

"The next decade of AI will not be defined only by more capable models. It will be defined by more governable systems."



Closing Reflection

The next generation of AI will not be defined by capability alone, but by the systems we build to govern it.

As intelligence becomes infrastructure, the questions that matter most shift from what models can do to how they participate in human environments. The architectures we create today will shape not only the behavior of these systems, but the relationships they enable and the trust they earn.

This paper has argued for a perspective in which governance is not an afterthought, but a foundational design principle. The decisions made at the architectural level determine what is possible at the behavioral level. Understanding this relationship is essential as AI moves from isolated tools to persistent participants in the systems that organize human life.

The work ahead is not only technical. It requires collaboration across disciplines, industries, and communities. The systems we build will reflect the values we hold. The question is not whether AI will become embedded in everyday life, but how we ensure it serves the people and relationships it touches.

Architecture determines what intelligence is allowed to become.

Captain Collective · Lite Paper v2.0 · July 2026 · captaincollective.co