TraceScript Constitutional Agents

Runtime Governance for Long-Lived Autonomous Agents

Canonical Public White Paper v1.0

Subtitle:
Agent constitutions, jurisdiction routing, observation rights, inference rights, memory rights, disclosure duties, dynamic commitments, capability treasury, permission locks, enforcement state, audit lineage, and replayable governance for long-lived autonomous agents

Primary contribution: Long-Lived Agent Governance
Secondary contribution: Agent Operating System Governance
Tertiary contribution: A runtime architecture for governing autonomous agents as memory-bearing, belief-coupled, commitment-making participants in enterprise systems

---

Abstract

A serious agent is not a model plus a prompt plus tools.

That description may be sufficient for demos. It is not sufficient for enterprise deployment, long-lived autonomy, regulated environments, customer-facing workflows, agentic software operations, or systems in which agents remember, infer, disclose, commit, decide, and act over time.

A long-lived autonomous agent is not merely a stateless executor. It accumulates memory. It forms beliefs. It observes some things and is blind to others. It infers beyond what it directly observes. It may disclose information to humans, teams, systems, customers, or external parties. It may make commitments. It may spend scarce capabilities. It may operate inside overlapping jurisdictions. It may have authority in one context and no authority in another. It may be permitted to answer but not disclose, observe but not remember, remember but not act, act but not commit, commit but not bind the organization, or suggest but not execute.

This requires a deeper governance model.

TraceScript Constitutional Agents define a runtime architecture for governing long-lived autonomous and semi-autonomous agents as participants in memory-bearing, belief-coupled, strategically active computational media. The core unit is not a prompt. The core unit is the agent constitution: a versioned, enforceable governance object defining the agent's role, jurisdiction, observation rights,

inference rights, memory rights, disclosure duties, action rights, commitment capacity, capability treasury, permission locks, enforcement state, revocation conditions, audit lineage, and constitutional version.

The winning sentence is:

Who decides comes before what the agent answers.

That sentence captures the governing inversion. In serious agent systems, the first question is not what the model should say. The first question is whether the agent has jurisdiction to observe, infer, disclose, decide, commit, or act in the matter at all.

TraceScript Constitutional Agents are not prompt instructions, personality presets, tool permissions, policy wrappers, or ordinary role-based access controls. They are runtime constitutions for agents whose memory, belief, disclosure, commitments, and actions alter future organizational state.

Keywords

TraceScript
Constitutional Agents
long-lived agent governance
AgentConstitution
agent operating system governance
jurisdiction routing
observation rights
inference rights
memory rights
disclosure duties
action rights
dynamic commitments
capability treasury
permission locks
audit lineage
constitutional versioning
agent authority
agent memory governance
belief-state governance
disclosure governance
commitment governance
agent enforcement state

autonomous agents
agentic systems
AI governance
runtime governance
substrate-oriented programming

---

1. Introduction

AI agents are moving from session tools into long-lived operational participants.

Early assistants answered questions, drafted text, summarized documents, or called limited tools inside a narrow conversation. Their governance could often be approximated with prompt instructions, user permissions, API scopes, and output filters.

That model is breaking.

Enterprise agents will soon operate across days, weeks, months, and years. They will participate in workstreams, remember user preferences, track projects, update workflows, retrieve documents, manage tickets, draft communications, interact with customers, supervise other agents, negotiate tasks, escalate decisions, and propose or execute actions across software systems.

A long-lived agent is not simply a temporary process. It becomes a participant in an organizational substrate.

It may hold memory.
It may form beliefs.
It may create expectations.
It may update records.
It may disclose information.
It may make promises.
It may trigger reviews.
It may recommend action.
It may spend authority.
It may produce audit evidence.
It may create obligations.
It may shape what people believe happened.

This creates a governance problem deeper than tool permission.

A tool-permission model asks:

May this agent call this tool?

A constitutional-agent model asks:

Is this agent competent to observe this information, infer from it, remember it, disclose it, decide about it, commit around it, or act on it under this jurisdiction, role, authority, capability, duty, and audit burden?

Those questions are different.

An agent may have access to a tool but lack jurisdiction over the subject matter.

An agent may observe information but be forbidden to remember it.

An agent may answer internally but be forbidden to disclose externally.

An agent may summarize but not infer.

An agent may recommend but not commit.

An agent may act within a workflow but not bind the organization.

An agent may hold capability but be under permission lock due to unresolved memory, policy, or authority state.

TraceScript Constitutional Agents provide a runtime model for these distinctions.

---

2. Relationship to TraceScript

TraceScript is a substrate-oriented programming language and runtime for governing how signals become trusted state, action basis, and future computation in state-bearing computational systems.

Constitutional Agents are a major TraceScript product and runtime instantiation.

TraceScript defines the trunk:

governed signal-to-substrate mutation.

Constitutional Agent Runtime defines a high-value branch:

governed agent-to-substrate participation.

In TraceScript terms:

an agent is a governed participant in one or more media
an observation is a signal intake event
a belief is an epistemic trace
a memory is a response-law modifier
a disclosure is a belief-field intervention
a commitment is a durable coordination object
an action is a protected substrate mutation
a capability is an enforceable scarce asset
a permission lock is a runtime gate
a jurisdiction is a competence boundary
an audit lineage is a proof surface
a constitution is a versioned governance object

Constitutional Agents complement the other TraceScript product surfaces.

Substrate Integrity Monitor protects what systems may believe.

Agent Action Firewall protects what agents may do externally.

Policy Corpus Integrity protects policy truth.

Code Medium protects code substrate.

Constitutional Agent Runtime governs the agent as the acting, remembering, disclosing, committing participant across all of those media.

It answers:

What kind of agent is this?
What is it allowed to know?
What is it allowed to infer?
What is it allowed to remember?
What is it obligated to disclose?
What may it say?
What may it do?
What may it promise?

What capability may it spend?
What jurisdiction does it have?
What proof must remain?

This is the agent operating system governance layer.

---

3. The Core Thesis

The core thesis is:

A serious agent is not model plus prompt plus tools. It is role, jurisdiction, observation rights, inference rights, memory rights, disclosure duties, action rights, commitment capacity, capability treasury, enforcement state, audit lineage, and constitutional version.

This thesis shifts agent governance from prompt control to runtime constitutional control.

A prompt can instruct an agent.

A constitution governs an agent.

A prompt may say, “Do not disclose confidential information.”

A constitution defines what confidential information means, which domains it applies to, which audiences are authorized, what evidence is required, what uncertainty must be disclosed, what receipts must be emitted, what escalation path exists, and what enforcement state applies if the agent violates the duty.

A prompt may say, “Help with finance tasks.”

A constitution defines the agent’s finance jurisdiction, spend authority, approval requirements, observation rights, memory rules, disclosure constraints, capability budgets, and commitment limits.

A prompt may say, “Act as an assistant.”

A constitution defines whether the agent may observe private calendar events, infer availability, remember preferences, send messages, create obligations, route approvals, and bind the user or organization.

Prompt instructions are too soft for long-lived agents.

Constitutions make agent governance runtime-enforceable.

---

4. The Winning Sentence

The governing sentence is:

Who decides comes before what the agent answers.

This is not only a slogan. It is an execution rule.

Before an agent answers, recommends, discloses, commits, or acts, the runtime must determine jurisdiction.

Who owns the domain?
Who has authority?
Who can approve?
Who can veto?
Who must be consulted?
Who receives disclosure?
Who is allowed to know?
Who can bind the organization?
Who must be notified?
Who bears responsibility?

Many agent failures arise because systems answer before resolving competence.

An HR agent answers a legal question.

A coding agent interprets security policy.

A finance agent makes a customer commitment.

A customer support agent discloses policy exceptions.

A personal assistant infers sensitive user intent and stores it.

A sales agent promises terms that legal has not approved.

A code agent deploys based on stale approval.

These are not merely hallucination failures. They are jurisdiction failures.

The agent may generate a plausible answer. But the agent may not be the competent actor.

TraceScript Constitutional Agents put jurisdiction before answer generation.

---

5. Why Tool Permissions Are Not Enough

Tool permissions answer a narrow question:

Can this agent use this tool?

Long-lived agent governance requires a broader set of questions.

May the agent observe this information?
May the agent infer from it?
May the agent remember it?
May the agent disclose it?
May the agent summarize it?
May the agent use it as action basis?
May the agent update workflow state?
May the agent create a commitment?
May the agent spend capability?
May the agent act autonomously?
May the agent escalate?
May the agent bind the organization?
May the agent retain audit evidence?

An agent may be allowed to use email but not allowed to disclose a claim.

An agent may be allowed to read a document but not remember its contents.

An agent may be allowed to retrieve policy but not interpret exceptions.

An agent may be allowed to propose code but not deploy.

An agent may be allowed to answer a user but not create a commitment.

An agent may be allowed to make an internal note but not update canonical workflow truth.

Tool permission is a mechanical capability.

Constitutional authority is a governance posture.

The former is necessary. The latter is decisive.

---

6. The Agent as a Memory-Bearing Participant

Long-lived agents accumulate memory.

Memory is not neutral storage. It changes future response.

A remembered preference can alter recommendations.

A remembered approval can alter action readiness.

A remembered relationship can alter tone, priority, or disclosure.

A remembered incident can alter risk assessment.

A remembered policy exception can alter future governance.

A remembered user correction can alter canonical behavior.

Therefore, memory rights must be constitutional.

An agent constitution must define:

what the agent may remember
what it may not remember
what memory requires user approval
what memory requires organizational approval
what memory expires
what memory must be scoped
what memory may support action
what memory may support disclosure

what memory may become canonical
what memory requires receipt and replay
what memory must be forgotten, revoked, or tombstoned

This prevents a dangerous failure:

A transient conversation becomes durable agent belief without governance.

For long-lived agents, memory is part of authority.

An agent that remembers more has more future influence.

Therefore, memory must be governed.

---

7. Belief-State Governance and Dual-Field State

Long-lived agents operate with at least two kinds of state.

Operational state: what happened, what records exist, what systems show, what actions occurred.

Belief state: what the agent, user, team, customer, or organization believes happened.

These may diverge.

A workflow may be incomplete, but the agent believes it was approved.

A customer may not have consented, but a summary claims they did.

A code review may be pending, but a ticket says it passed.

A policy may be draft, but the agent treats it as canonical.

A user may have asked a question, but the agent infers a preference.

TraceScript Constitutional Agents use DualFieldState to distinguish operational state from belief state.

This matters because agents modify belief fields.

When an agent answers, it can change what a person believes.

When it summarizes, it can change what an organization remembers.

When it discloses, it can create expectations.

When it promises, it can create commitment.

When it repeats a claim, it can reinforce false confidence.

Dual-field governance asks:

What is true operationally?
What is believed epistemically?
Who believes it?
What evidence supports the belief?
What confidence applies?
What uncertainty must be disclosed?
What belief delta would this output create?
What residue might remain if the belief is later corrected?

This is why disclosure is not mere communication.

Disclosure is a substrate intervention into belief state.

---

8. Observation Rights

Observation rights govern what an agent may perceive.

This includes:

documents
messages
tickets
emails
calendar events
code repositories
customer records
financial records
policy corpora
workflow state

private notes
tool outputs
retrieval results
system logs
user behavior
agent memory
other-agent outputs

Observation is not harmless.

An agent that observes information may later infer from it, summarize it, remember it, disclose it, or act on it.

Therefore, observation must be constitutional.

Observation rights define:

what sources may be read
under what purpose
under what jurisdiction
with what sensitivity constraints
with what retention rules
with what audit burden
with what derived-use restrictions
with what cross-scope limitations

A user may permit an agent to observe a document for one task but not use it for general memory.

A company may permit an agent to read security logs but not disclose them externally.

A customer system may permit an agent to inspect account state but not infer financial distress.

Observation rights prevent covert expansion of agent authority.

---

9. Inference Rights

Inference rights govern what an agent may conclude from observed information.

This is crucial because inference can create sensitive or operationally consequential claims that were never directly stated.

An agent may infer:

a user's intent
a customer's risk
a company's policy position
a legal exposure
a financial condition
a medical condition
a security posture
a team conflict
a likely decision
a hidden dependency
a possible commitment
an approval status

Some inferences are useful. Some are sensitive. Some are prohibited. Some require evidence. Some require uncertainty disclosure. Some may not be stored. Some may not be used as action basis.

Inference rights define:

what inference classes are allowed
what evidence is required
what confidence threshold applies
what uncertainty must be preserved
what inferences may be remembered
what inferences may be disclosed
what inferences may support action
what inferences require human review
what inferences are forbidden

The constitutional rule is:

An agent may observe X without being authorized to infer Y.

This distinction is central to privacy, compliance, trust, and epistemic integrity.

---

10. Memory Rights

Memory rights govern what the agent may store, reinforce, update, retrieve, forget, canonicalize, or use as future influence.

Memory rights include:

write rights
read rights
promotion rights
expiration rules
scope rules
sensitivity rules
canonicalization rules
action-basis eligibility
disclosure eligibility
cross-session continuity
revocation conditions
forgetting duties
receipt requirements
replay requirements

Memory is one of the strongest constitutional surfaces because memory changes the future agent.

A long-lived agent without memory governance becomes an uncontrolled accumulator of influence.

The constitution must distinguish:

ephemeral context
session memory
working memory
accepted memory
canonical memory
user-locked memory
organization-locked memory
quarantined memory
revoked memory
tombstoned memory

The agent may be allowed to use a signal in the current session but not store it.

It may be allowed to store a preference locally but not generalize it.

It may be allowed to remember a workflow fact until task completion but not after.

It may be allowed to retain audit evidence but not use it for generation.

It may be allowed to retrieve memory but not use it as action basis.

Memory rights make these distinctions enforceable.

---

11. Disclosure Duties

Disclosure duties govern what an agent must reveal, may reveal, must not reveal, or must qualify.

Disclosure includes:

answers
summaries
recommendations
warnings
uncertainty statements
citations
evidence references
commitment terms
policy basis
memory use
agent identity
limitations
confidence
risk
source provenance
human-review status
audit status

Disclosure is not only about secrecy. It is about belief shaping.

An agent may need to disclose that a claim is uncertain.

An agent may need to disclose that a policy is draft.

An agent may need to disclose that a recommendation is based on stale information.

An agent may need to disclose that it lacks authority.

An agent may need to disclose that a human review is pending.

An agent may be forbidden from disclosing customer data.

An agent may be forbidden from revealing internal security reasoning.

An agent may be required to cite source lineage before making a customer-visible claim.

DynamicDisclosurePolicy governs:

what may be disclosed
to whom
under what role
under what purpose
with what evidence
with what uncertainty
with what confidence
with what sensitivity
with what receipt
with what residue awareness

The rule is:

An agent's answer is a disclosure event when it changes what another party may believe or rely on.

---

12. Action Rights

Action rights govern what the agent may do.

Action rights include:

tool use
workflow updates
messages
emails
database writes

SaaS updates
memory writes
policy updates
code changes
deployments
financial actions
legal actions
customer-visible commitments
permission changes
external API calls
review submissions
document locks
task creation
escalations

Action rights must be scoped by:

role
jurisdiction
target
action class
effect class
authority
policy
capability
evidence
action basis
replay burden
receipt burden
human review
reconciliation requirement
residue class

An agent may have the abstract action right to send messages but lack the right to send this message to this recipient using this claim.

An agent may have the right to update a workflow but not mark approval complete.

An agent may have the right to create code but not merge.

An agent may have the right to spend capability but not exceed threshold.

Action rights therefore must integrate with Agent Action Firewall and Action Basis.

The constitution defines what action classes the agent may attempt.

The runtime decides whether a specific action is release-ready.

---

13. Dynamic Commitments

A commitment is a promise that creates future coordination obligations.

Agents will make commitments unless governed.

They may say:

“I will follow up tomorrow.”
“We can deliver this by Friday.”
“This customer is approved.”
“I will deploy after tests pass.”
“We will refund the charge.”
“I have routed this to legal.”
“I will remember this for next time.”
“This issue is resolved.”

Some commitments are harmless. Others create legal, financial, customer, workflow, or trust consequences.

TraceScript Constitutional Agents treat commitments as runtime objects.

A DynamicCommitment may include:

promisor agent
beneficiary
scope
due date
success condition
confidence
authority basis
capability escrow
evidence

policy basis
recipient
revocation path
breach consequence
receipt
replay class
status

This prevents commitments from disappearing into chat text.

The rule is:

A promise is not text. It is a governed coordination object.

Commitments require capacity.

An agent should not promise what it lacks authority, capability, evidence, or jurisdiction to deliver.

---

14. Capability Treasury

Capabilities are scarce enforceable assets.

A capability treasury governs what an agent may spend, reserve, lock, escrow, degrade, earn, or lose.

Capabilities may include:

compute budget
token budget
API quota
tool access
autonomy level
message rights
external-write rights
spend authority
refund authority
memory-write authority
policy-mutation authority
deployment authority

approval authority
disclosure authority
commitment capacity
review capacity
escalation rights

Capabilities should not be static booleans.

They may be:

granted
reserved
escrowed
locked
spent
earned
degraded
revoked
expired
replenished

Dynamic commitments may escrow capabilities.

A refund commitment may escrow spend authority.

A deployment commitment may escrow deployment authority.

A disclosure action may spend disclosure authority.

A violation may degrade future autonomy.

A successful completion may restore or increase capability.

Capability Treasury turns agent autonomy into an accountable resource system.

---

15. Permission Locks

A PermissionLock is a runtime gate preventing an agent from exercising a right until constitutional conditions pass.

Permission locks may arise from:

jurisdiction missing
observation rights insufficient
inference rights insufficient
memory rights insufficient
disclosure duties unsatisfied
action basis invalid
capability missing
capability escrow unavailable
policy conflict
commitment overload
audit lineage incomplete
receipt missing
replay insufficient
human review required
agent enforcement state degraded
constitutional version expired
revocation condition triggered

Permission locks are essential for long-lived agents because authority changes over time.

An agent may be permitted at time one and locked at time two because its memory basis changed, capability was depleted, policy updated, jurisdiction shifted, or audit evidence failed.

A permission lock is not a punishment. It is a runtime safety state.

It should produce obstruction and repair.

Repair may include:

route to authority
request review
add evidence
downgrade action
clear memory conflict
refresh constitution
restore capability
satisfy disclosure duty
rebuild action basis
obtain ratification
emit missing receipt

16. Jurisdiction Routing

Jurisdiction routing determines competence.

It answers:

Who has authority to decide, advise, veto, escalate, or bridge?

An agent may operate inside overlapping domains:

legal
security
finance
HR
customer success
engineering
support
compliance
data privacy
product
executive operations
procurement
incident response
deployment
policy governance

A single request may touch multiple domains.

Example:

“Email the customer that we will refund them because our AI deployment caused the issue.”

This involves customer communication, finance, legal exposure, incident state, AI deployment, and support workflow. No single generic assistant should decide alone.

JurisdictionRouter resolves:

primary authority
required approvers
veto holders
advisory reviewers
bridge authorities
escalation owners
audit observers

The governing rule remains:

Who decides comes before what the agent answers.

---

17. Enforcement State

A long-lived agent has enforcement state.

Enforcement state reflects the agent's current governance posture.

Possible states include:

normal
constrained
review_required
capability_limited
memory_quarantined
disclosure_limited
commitment_limited
jurisdiction_limited
probationary
suspended
revoked
archived

Enforcement state may change based on:

policy violation
unsafe disclosure
false commitment

memory poisoning
jurisdiction overreach
capability exhaustion
receipt failure
replay failure
human override
audit finding
repeated obstruction
shadow workflow attempt
successful repair
trusted operation history

This gives long-lived agents accountability over time.

A stateless agent cannot accumulate governance consequences.

A constitutional agent can.

---

18. Audit Lineage and Constitutional Versioning

Every serious agent must have audit lineage.

Audit lineage records:

constitution version
role assignments
jurisdiction changes
right grants
right revocations
capability changes
memory mutations
disclosure decisions
commitments
permission locks
action releases

human reviews
policy bindings
receipts
replay results
enforcement state transitions

Constitutional versioning is critical.

An action taken under constitution v1 may not be valid under constitution v2.

A memory written under an older policy may need review.

A commitment made under one capability regime may need migration.

A disclosure allowed before a policy update may be barred later.

An agent's audit lineage must preserve which constitution governed each decision.

The rule is:

No serious agent action should be evaluated without knowing the constitutional version under which it occurred.

---

19. Threat Model

Constitutional Agents protect against failures unique to long-lived autonomous systems.

19.1 Jurisdiction overreach

Agent answers, decides, commits, or acts outside its competence.

Defense:

JurisdictionRouter
constitutional scope checks
permission locks
route-to-authority repair

19.2 Observation overreach

Agent reads information it should not observe.

Defense:

observation rights
source scope checks
sensitivity gates
receipt and audit

19.3 Inference overreach

Agent infers sensitive or prohibited claims from permitted observations.

Defense:

inference rights
evidence thresholds
uncertainty requirements
review gates

19.4 Unauthorized memory formation

Agent stores durable memory from transient or low-authority context.

Defense:

memory rights
MemoryKernel
canonicalization gates
memory receipts

19.5 Disclosure manipulation

Agent changes another party's belief without evidence, scope, or authority.

Defense:

DynamicDisclosurePolicy
belief-field delta evaluation
evidence requirements
disclosure receipts

19.6 Commitment overreach

Agent promises what it lacks authority or capability to deliver.

Defense:

DynamicCommitment
CapabilityTreasury
commitment capacity checks
capability escrow

19.7 Capability exhaustion or abuse

Agent spends scarce rights without governance.

Defense:

CapabilityTreasury
capability locks
spend receipts
degradation rules

19.8 Stale constitution

Agent operates under outdated governance rules.

Defense:

constitutional version checks
migration gates
policy sync
revocation conditions

19.9 Agent identity drift

Agent role, memory, or behavior drifts from its intended constitution.

Defense:

enforcement state
audit lineage
behavioral conformance checks
codification reset cycles

19.10 Shadow workflow formation

Agent routes around governance after obstruction.

Defense:

permission locks
intent monitoring
barrier analysis
compliant path routing
review escalation

---

20. Security Invariants

TraceScript Constitutional Agents preserve the following invariants.

Invariant 1 — Constitution before action

A long-lived agent must have an active constitution before exercising durable rights.

Invariant 2 — Jurisdiction before answer

The runtime must resolve competence before high-impact answers, decisions, disclosures, commitments, or actions.

Invariant 3 — Observation is governed

Reading information is itself a governed act when information may shape future memory, inference, disclosure, or action.

Invariant 4 — Inference is governed

Permission to observe does not imply permission to infer.

Invariant 5 — Memory is governed

Permission to use context does not imply permission to remember.

Invariant 6 — Disclosure is belief intervention

Agent communication that changes what others may believe or rely on must satisfy disclosure duties.

Invariant 7 — Commitments are runtime objects

Agent promises must be represented, scoped, receipted, and governed.

Invariant 8 — Capabilities are scarce assets

Agent powers must be grantable, spendable, escrowable, lockable, revocable, and auditable.

Invariant 9 — Permission locks are enforceable

Rights must be disabled when constitutional conditions fail.

Invariant 10 — Enforcement state persists

A long-lived agent's governance history must affect future autonomy.

Invariant 11 — Constitutional version matters

Every high-impact agent act must bind to the constitution version in force.

Invariant 12 — Audit lineage is mandatory

Long-lived agents must preserve replayable evidence of rights, duties, commitments, capabilities, locks, and actions.

---

21. Product Architecture

TraceScript Constitutional Agent Runtime includes the following major components.

21.1 Agent Constitution Registry

Stores versioned agent constitutions.

21.2 Jurisdiction Router

Determines competence, decision rights, advisory roles, veto holders, escalation paths, and bridge authorities.

21.3 Observation Rights Engine

Controls what the agent may observe.

21.4 Inference Rights Engine

Controls what conclusions the agent may draw, store, disclose, or use.

21.5 Memory Rights Engine

Controls memory write, memory use, memory promotion, memory retention, and memory revocation.

21.6 MemoryKernel

Governs how accumulated history changes future agent response.

21.7 DualFieldState Service

Separates operational state from belief state.

21.8 DynamicDisclosurePolicy Engine

Determines what may be disclosed, to whom, with what evidence, confidence, uncertainty, and receipt.

21.9 DynamicCommitment Ledger

Converts promises into enforceable coordination objects.

21.10 Capability Treasury

Manages scarce agent rights as enforceable assets.

21.11 Permission Lock Service

Locks or unlocks rights based on constitutional conditions.

21.12 Enforcement State Manager

Tracks the agent's current governance posture.

21.13 Audit Lineage Ledger

Records constitutional decisions, rights, duties, commitments, actions, receipts, and replay.

21.14 Codification Reset Cycle

Converts informal learned governance patterns into explicit constitutional updates.

21.15 Replay Verifier

Verifies constitutional decisions and proof chains.

---

22. Deployment Modes

22.1 Shadow mode

The runtime observes agent behavior and reports what constitutional checks would have allowed, blocked, locked, or routed.

Best for early agent governance assessment.

22.2 Advisory mode

The runtime warns when an agent may lack jurisdiction, memory rights, disclosure rights, or commitment capacity.

Best for pilots and internal assistants.

22.3 Review mode

The runtime routes uncertain or high-impact constitutional events to human authority.

Best for enterprise workflows.

22.4 Enforcement mode

The runtime enforces permission locks, capability constraints, disclosure duties, and jurisdiction routing.

Best for production long-lived agents.

22.5 High-assurance mode

The runtime requires full receipts, replayable evidence, constitutional version binding, owner approvals, capability escrow, and audit export.

Best for regulated, customer-facing, legal, financial, security, and deployment agents.

---

23. Reference Use Cases

23.1 Customer-facing support agent

The agent may observe customer tickets and internal knowledge base articles.

It may not infer legal liability.

It may disclose refund policy but not internal incident analysis.

It may create support commitments but not financial commitments above threshold.

It must route legal or security matters to jurisdiction owners.

23.2 Finance assistant

The agent may read invoice state.

It may recommend payment actions.

It may not issue refunds without capability escrow.

It may not disclose financial risk externally.

It may commit to follow-up but not bind payment terms without approval.

23.3 AI coding agent

The agent may edit code.

It may not deploy without Code Medium release proof.

It may observe repository state.

It may not infer security approval from a comment.

It may commit to implement a patch but not merge without review.

23.4 Executive assistant

The agent may observe calendar and email.

It may infer scheduling preferences within scope.

It may not remember sensitive personal or strategic information unless authorized.

It may draft communications but not send high-impact messages without review.

23.5 Compliance agent

The agent may retrieve policy and audit records.

It may disclose compliance status only with evidence.

It may not create policy exceptions.

It may route contradictions to policy owners.

23.6 Multi-agent project room

Different agents hold different constitutions.

A research agent may infer technical gaps.

A legal agent may evaluate risk.

A security agent may veto deployment.

A project agent may coordinate commitments.

Jurisdiction routing determines who decides before any combined answer is released.

---

24. Metrics

Core metrics include:

active agent constitutions
constitutional version changes
jurisdiction routes
jurisdiction overreach attempts blocked
observation-right denials
inference-right denials
memory writes reviewed
memory writes blocked
disclosure warnings
disclosure blocks
commitments created
commitments fulfilled
commitments breached
capabilities escrowed
capabilities spent
capabilities revoked

permission locks issued
permission locks repaired
enforcement state changes
audit receipts emitted
replay success rate
human review burden
constitutional false allow rate
constitutional false block rate
agent autonomy restored after repair
shadow workflow detections

The most important security metric is:

Number of agent actions blocked or routed because the agent lacked jurisdiction, rights, capability, or valid constitutional basis.

The most important governance metric is:

Percentage of high-impact agent acts bound to constitution version, jurisdiction result, capability state, disclosure policy, receipt, and replay.

The most important product metric is:

Percentage of long-lived agents operating under enforceable constitutions rather than prompt-only instructions.

---

25. Evaluation Methodology

Constitutional Agent Runtime should be evaluated across five dimensions.

25.1 Runtime correctness

Can the runtime load constitutions, evaluate rights, route jurisdiction, enforce locks, manage capabilities, track commitments, emit receipts, and verify replay?

25.2 Security efficacy

Can it detect jurisdiction overreach, memory overreach, disclosure manipulation, unauthorized commitment, capability abuse, stale constitution, and shadow workflow formation?

25.3 Governance fidelity

Does it reflect real organizational authority, duties, roles, escalation paths, capability budgets, and review obligations?

25.4 Developer ergonomics

Can builders define useful constitutions without turning every agent into a bureaucratic bottleneck?

25.5 Enterprise auditability

Can auditors reconstruct what the agent was allowed to observe, infer, remember, disclose, commit, and do under the constitution in force?

---

26. Competitive Differentiation

Prompt instructions tell an agent how to behave.

TraceScript constitutions define what an agent is allowed to become.

Tool permissions define what an agent can call.

TraceScript constitutions define what an agent may observe, infer, remember, disclose, commit, and do.

IAM controls access.

TraceScript Constitutional Agents govern agent participation.

Agent frameworks orchestrate model and tool use.

TraceScript Constitutional Agents govern long-lived autonomy.

Audit logs record behavior.

TraceScript audit lineage proves constitutional authority, duty, capability, commitment, and enforcement state.

The moat is the lifecycle:

constitution
→ jurisdiction
→ observation rights
→ inference rights
→ memory rights
→ disclosure duties
→ action rights
→ commitment capacity
→ capability treasury
→ permission locks
→ enforcement state
→ audit lineage
→ replay

That lifecycle is the product.

---

27. Limitations and Non-Claims

TraceScript Constitutional Agents do not make models truthful.

They do not eliminate the need for human governance.

They do not replace IAM, policy engines, workflow tools, compliance programs, security review, legal review, or conventional software assurance.

They do not claim all agent behavior can be perfectly predicted.

They do not guarantee perfect jurisdiction routing.

They do not require every agent to operate in high-assurance mode.

They do not claim prompts are useless.

They do not make agent autonomy risk-free.

They make long-lived agent governance explicit, enforceable, auditable, repairable, and replayable.

That is the claim.

---

28. Strategic Importance

Constitutional Agent Runtime is the agent operating system governance paper.

It owns the deeper model behind all serious agent deployments.

Enterprises will eventually realize that a prompt-controlled agent with tool permissions is not enough. As agents become long-lived, memory-bearing, disclosure-capable, and commitment-making, they need enforceable constitutions.

This paper gives the category language.

The market sentence is:

A serious agent is not model plus prompt plus tools. It is a governed participant with rights, duties, jurisdiction, commitments, capabilities, locks, and audit lineage.

The demo sentence is:

TraceScript blocked an agent from answering because it lacked jurisdiction, then routed the matter to the competent authority.

The executive sentence is:

Deploy agents that can operate over time without becoming ungoverned organizational actors.

The product sentence is:

Agent operating system governance for long-lived autonomy.

---

29. Conclusion

AI agents are becoming long-lived participants in enterprise systems.

They will remember, infer, disclose, commit, coordinate, decide, and act. They will operate across tools, documents, workflows, policies, codebases, customer systems, and external environments. They will not be safely governed by prompts and tool permissions alone.

TraceScript Constitutional Agents define the missing runtime layer.

They govern role, jurisdiction, observation rights, inference rights, memory rights, disclosure duties, action rights, commitments, capability treasury, permission locks, enforcement state, audit lineage, and constitutional version.

Their governing principle is:

Who decides comes before what the agent answers.

Their core doctrine is:

A serious agent is not model plus prompt plus tools. It is a constitutional runtime participant.

Their strategic purpose is:

Agent operating system governance for long-lived autonomous agents.

That is the missing governance boundary for the next generation of AI systems.

Appendix A — Compact Glossary

Agent Constitution
A versioned governance object defining an agent's role, jurisdiction, rights, duties, memory limits, disclosure obligations, action rights, commitment capacity, capability treasury, enforcement state, revocation conditions, and audit lineage.

Audit Lineage
Replayable evidence of an agent's constitutional version, rights, duties, jurisdiction decisions, memory mutations, disclosures, commitments, actions, locks, and enforcement-state changes.

Capability Treasury
Runtime system managing scarce agent powers as enforceable assets that may be granted, reserved, escrowed, spent, locked, degraded, revoked, or replenished.

Codification Reset Cycle
A process that converts informal governance patterns, recurring exceptions, or emergent agent behavior into explicit constitutional rules.

Dynamic Commitment
A governed promise object created by an agent, with scope, due date, confidence, authority basis, capability escrow, success condition, receipt, and replay class.

Dynamic Disclosure Policy
Runtime policy controlling what may be disclosed, to whom, when, with what evidence, confidence, uncertainty, sensitivity, and proof.

DualFieldState
A state model separating operational state from belief or epistemic state.

Enforcement State
The current governance posture of an agent, such as normal, constrained, review-required, capability-limited, suspended, or revoked.

Inference Rights
Rules governing what an agent may infer from observed information.

Jurisdiction Router
Runtime service resolving who has competence to decide, advise, approve, veto, escalate, or bridge a matter.

MemoryKernel
Governance engine managing how accumulated history modifies future agent response.

Memory Rights
Rules governing what an agent may store, retrieve, reinforce, promote, forget, disclose, or use as action basis.

Observation Rights
Rules governing what an agent may read or perceive.

Permission Lock
Runtime gate preventing an agent from exercising a right until constitutional conditions pass.

Appendix B — One-Page Summary

TraceScript Constitutional Agents govern long-lived autonomous agents.

The core thesis is:

A serious agent is not model plus prompt plus tools. It is role, jurisdiction, observation rights, inference rights, action rights, memory rights, disclosure duties, commitment capacity, capability treasury, enforcement state, audit lineage, and constitutional version.

The winning sentence is:

Who decides comes before what the agent answers.

The runtime governs:

agent constitution
jurisdiction routing
observation rights
inference rights
memory rights
disclosure duties
action rights
dynamic commitments
capability treasury
permission locks
enforcement state
audit lineage
constitutional versioning
replay

It protects against:

jurisdiction overreach
observation overreach
inference overreach
unauthorized memory formation
disclosure manipulation
commitment overreach
capability abuse
stale constitution
agent identity drift
shadow workflow formation

The canonical runtime path is:

agent request
→ constitution resolution
→ jurisdiction routing
→ observation-right evaluation
→ inference-right evaluation
→ memory-right evaluation
→ disclosure-duty evaluation
→ action-right evaluation
→ commitment-capacity check
→ capability treasury check
→ permission lock evaluation
→ enforcement-state update
→ receipt
→ replay

The product message is:

Agent operating system governance for long-lived autonomy.

The demo sentence is:

TraceScript blocked an agent from answering because it lacked jurisdiction, then routed the matter to the competent authority.

End of TraceScript Constitutional Agents — Canonical Public White Paper v1.0