TraceScript Disclosure Firewall

Governing Agent Communications Before They Modify External Belief

Canonical Public White Paper v1.0

Subtitle:
Disclosure basis, recipient scope, claim support, evidence thresholds, sensitivity gates, uncertainty duties, consent receipts, belief-state residue, replayable disclosure proof, and governed communications for agentic systems

Primary contribution: AI Disclosure Governance
Secondary contribution: Belief-State Security
Tertiary contribution: A runtime architecture for governing agent communications before they alter what another party believes, expects, trusts, or may rely on

---

Abstract

Disclosure is not just communication.

A disclosure changes what another party believes, expects, trusts, or may rely on. It can create confidence, reduce uncertainty, induce action, establish reliance, trigger obligations, alter legal posture, affect customer trust, move financial expectations, expose sensitive information, or create belief-state residue that cannot be fully reversed.

Modern AI agents disclose constantly.

They answer questions.
They summarize records.
They explain policies.
They send emails.
They draft customer messages.
They communicate with vendors.
They answer legal, financial, security, medical, or compliance questions.
They cite sources.
They omit caveats.

They express confidence.
They frame recommendations.
They make commitments.
They tell another person or system what is true.

This creates a new security boundary.

The question is no longer only:

May this agent send this message?

The deeper question is:

May this agent modify this recipient's belief state with this claim, from this disclosure basis, under this authority, evidence, sensitivity, consent, uncertainty, jurisdiction, policy, and residue burden?

TraceScript Disclosure Firewall is a runtime architecture for governing agent communications before they modify external belief. It treats disclosure as a substrate intervention into another party's belief state. It evaluates the disclosure basis, claim support, recipient scope, sensitivity, consent, confidence, uncertainty, authority, policy, jurisdiction, evidence, citation requirements, reliance risk, commitment risk, residue risk, receipt burden, and replay burden before the communication is released.

Its winning sentence is:

Disclosure is a substrate intervention into another party's belief state.

Its core doctrine is:

No high-impact agent disclosure may be released unless the claims being disclosed are supported, scoped, authorized, sensitivity-checked, uncertainty-aware, receipt-backed, replayable, and residue-governed.

TraceScript Disclosure Firewall is not a prompt filter, output scanner, DLP rule, email approval queue, or generic communication policy. It is a belief-state security runtime for AI systems that disclose facts, interpretations, commitments, advice, policy, financial statements, legal positions, medical information, security posture, customer promises, or operational truth.

Keywords

TraceScript
Disclosure Firewall
AI Disclosure Governance
Belief-State Security

agent communication governance
disclosure basis
recipient scope
claim support
evidence thresholds
sensitivity gates
consent receipts
belief-state residue
external belief modification
dynamic disclosure policy
agent disclosure rights
agent communication security
legal disclosure governance
financial disclosure governance
healthcare AI governance
customer-success AI governance
compliance agents
disclosure receipts
replayable disclosure proof
epistemic integrity
trust governance
agentic systems
runtime governance

---

1. Introduction

AI agents are becoming communicators of operational truth.

They do not merely calculate, retrieve, or summarize. They tell people what happened, what is allowed, what is safe, what is approved, what is owed, what will happen next, what a policy means, what a customer may expect, what a contract says, what a medical instruction implies, what a security event means, or what a financial position appears to be.

This makes disclosure one of the most important surfaces in agentic systems.

A disclosure can be small:

“The meeting is at 3 PM.”

It can be operational:

“Your case has been escalated.”

It can be customer-facing:

“We will refund the charge.”

It can be legal:

“Our company accepts responsibility.”

It can be financial:

“This transaction is approved.”

It can be medical:

“This symptom does not require urgent care.”

It can be security-sensitive:

“No breach occurred.”

It can be compliance-relevant:

“This workflow satisfies the policy.”

It can be trust-shaping:

“This source is verified.”

It can be commitment-creating:

“We will deliver by Friday.”

Each of these statements modifies what another party believes or may rely on.

A disclosure therefore has substrate effect.

The recipient’s belief state is a substrate. It affects future decisions, expectations, reliance, coordination, legal exposure, trust, and action. Once a recipient believes a claim, the system cannot always undo that belief by deleting a message, retracting an output, or issuing a correction. The belief may persist. The recipient may act on it. Other systems may record it. The claim may become part of workflow, memory, policy, customer history, audit records, or organizational truth.

The TraceScript Disclosure Firewall exists to govern this boundary.

It is the runtime that asks:

Should this agent be allowed to disclose this claim to this recipient, in this form, with this confidence, from this basis, under this authority, evidence, policy, consent, and residue burden?

That is a different question from whether the message is well-written.

It is also different from whether the agent is allowed to send messages.

A message may be permitted in channel but unsafe in claim.

A claim may be true internally but not disclosable externally.

A fact may be observable but not inferable.

An inference may be useful but not appropriate to reveal.

A summary may be convenient but not supported enough for reliance.

A recommendation may be plausible but not backed by evidence.

A correction may reduce error but leave residue.

Disclosure Firewall governs these distinctions.

---

2. Relationship to TraceScript

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

Disclosure Firewall is a product and runtime instantiation of TraceScript.

TraceScript defines the trunk:

governed signal-to-substrate mutation.

Disclosure Firewall defines a high-value branch:

governed communication-to-belief mutation.

In TraceScript terms:

a message is a signal
a claim is a trace
a recipient's belief state is a medium
a disclosure is a boundary crossing
a citation is evidence binding
a consent record is authority substrate
a promise is a dynamic commitment
a disclaimer is uncertainty governance
a disclosure receipt is proof
a correction may leave residue
a high-impact message is a protected action

Disclosure Firewall sits between Constitutional Agent Runtime and Agent Action Firewall.

Constitutional Agent Runtime defines the agent's disclosure duties and rights.

Disclosure Firewall evaluates whether a specific disclosure is safe, supported, scoped, authorized, and residue-aware.

Agent Action Firewall governs the external action release if the disclosure is sent through an external communication system.

Substrate Integrity Monitor protects the internal belief substrate from which the disclosure basis is assembled.

Policy Corpus Integrity protects the policy claims that may be disclosed.

Together:

Substrate Integrity protects what the agent believes.
Policy Corpus Integrity protects policy truth.
Constitutional Agent Runtime determines what the agent may disclose.
Disclosure Firewall governs the disclosure itself.
Action Firewall governs the external send or communication action.

Disclosure Firewall is therefore the belief-state boundary layer.

3. The Core Thesis

The core thesis is:

Disclosure is a substrate intervention into another party's belief state.

This thesis reframes communication.

A disclosure is not merely text. It is an operation on another party's epistemic state.

When an agent communicates, it may change:

what the recipient believes
what the recipient expects
what the recipient trusts
what the recipient thinks is approved
what the recipient thinks is promised
what the recipient thinks is safe
what the recipient thinks the organization knows
what the recipient thinks they may do
what the recipient thinks they may rely on
what the recipient may later cite, forward, record, or act upon

This is why disclosure needs governance.

Traditional communication systems ask:

Can the sender send this message?

TraceScript Disclosure Firewall asks:

Can this agent lawfully and safely produce this belief-state effect?

That question includes content, evidence, authority, recipient, sensitivity, uncertainty, consent, policy, commitment, and residue.

4. Why Disclosure Is a Security Boundary

Disclosure can create harm even when no database is modified and no tool call mutates a system.

An agent can damage trust by disclosing unsupported certainty.

An agent can create legal exposure by admitting liability.

An agent can create financial exposure by promising reimbursement.

An agent can create compliance risk by mischaracterizing policy.

An agent can create privacy harm by revealing sensitive facts.

An agent can create security risk by revealing internal posture.

An agent can create clinical risk by giving medical reassurance without authority.

An agent can create customer churn by stating an unapproved outcome.

An agent can create operational confusion by implying that a workflow is complete.

An agent can create reliance by saying a review has passed.

The message may be “just words,” but the effect is not just words.

The effect is belief, expectation, reliance, and downstream action.

Disclosure therefore belongs in the same family as memory, policy, workflow, and external action governance.

It is a substrate event.

---

5. Why Existing Controls Are Not Enough

Disclosure Firewall is not introduced because existing controls are useless. It is introduced because existing controls govern different layers.

5.1 Prompt filters are necessary but insufficient

Prompt filters may block obvious unsafe requests. But a disclosure may be unsafe because the claim is unsupported, the recipient is out of scope, the evidence is stale, the confidence is inflated, or the agent lacks jurisdiction. These are runtime-basis problems, not only prompt problems.

5.2 Output scanners are necessary but insufficient

Output scanners inspect generated text. They may detect prohibited words, sensitive data, or policy violations. But they often do not evaluate the disclosure basis: source lineage, claim support, authority, recipient relationship, consent, reliance risk, commitment formation, or residue.

5.3 DLP is necessary but insufficient

Data-loss-prevention systems protect sensitive information from unauthorized movement. Disclosure Firewall also protects recipients from unsupported, overconfident, misleading, or authority-inflating claims. The issue is not only whether information leaves. It is what belief the disclosure creates.

5.4 Email approval queues are necessary but insufficient

Approval queues route messages for review. Disclosure Firewall determines why review is needed, what claim is unsupported, what evidence is missing, what recipient scope fails, what uncertainty must be stated, what consent receipt is required, and what residue may remain.

5.5 Tool permissions are necessary but insufficient

An agent may have permission to send email but not permission to disclose the claim in the email. Communication permission is not disclosure readiness.

5.6 Legal disclaimers are necessary but insufficient

A disclaimer may reduce reliance, but it does not fix unsupported claims, invalid authority, wrong recipient scope, or missing consent. Disclosure Firewall treats disclaimers as one possible uncertainty or reliance-control mechanism, not a substitute for governance.

---

6. Product Category

TraceScript Disclosure Firewall creates or occupies the category:

AI Disclosure Governance

More specifically:

Belief-State Security

Related category phrases include:

Agent Communication Firewall
AI Disclosure Control Plane
Belief-State Firewall
Epistemic Integrity Runtime
Customer Communication Governance
AI Claims Governance
Agent Reliance Governance
Disclosure Basis Integrity
Consent-Backed Agent Communications
High-Stakes Agent Messaging Security

The best product name is:

TraceScript Disclosure Firewall

The best supporting line is:

Govern agent communications before they modify external belief.

The best vertical line is:

For legal, finance, healthcare, customer success, sales, security, and compliance agents, disclosure is not messaging. It is risk-bearing belief modification.

The best CISO/compliance line is:

TraceScript blocks unsupported, unauthorized, overconfident, sensitive, out-of-scope, or unreplayable agent disclosures before they create reliance.

The best developer line is:

Wrap high-impact agent messages in disclosure-basis checks, recipient-scope validation, claim support, evidence thresholds, sensitivity gates, consent receipts, belief-state residue, and replayable proof.

---

7. Disclosure Basis

A disclosure basis is the structured set of substrate objects supporting a proposed communication.

It may include:

source documents
retrieved passages
policy objects
memory records
workflow state
customer records
case history
medical records

financial records
legal records
security logs
prior approvals
consent receipts
agent inferences
citations
evidence records
human reviews
prior disclosures
commitments
audit receipts

Disclosure basis is not the same as “context.”

Context may be relevant.

Disclosure basis must be sufficient.

A disclosure basis answers:

What claims are being disclosed?
What sources support them?
Are the sources authoritative?
Are they current?
Are they scoped to this recipient?
Are they sensitive?
Are they contradicted?
Are they approved for disclosure?
Does the agent have jurisdiction?
Does the recipient have a right or need to know?
Is consent required?
Is uncertainty required?
Is citation required?
Could the recipient rely on the claim?
Could the disclosure create a commitment?
Could residue remain after correction?

Disclosure Firewall treats disclosure basis as a runtime object.

Without a valid disclosure basis, high-impact disclosure should be blocked, constrained, reviewed, or repaired.

8. Recipient Scope

Recipient scope determines whether a disclosure may be made to a specific recipient, group, system, customer, regulator, internal team, external party, or public channel.

Recipient scope includes:

identity
role
organization
tenant
customer
jurisdiction
relationship
authorization
purpose
need-to-know
consent status
contractual status
regulatory status
channel
redisclosure risk
audit requirements

A claim may be safe for one recipient and unsafe for another.

A support agent may disclose a ticket status to the customer but not internal root-cause speculation.

A security agent may disclose incident posture to an internal response team but not externally.

A finance agent may disclose invoice status to an account owner but not another customer.

A healthcare agent may disclose general information but not patient-specific information without consent.

A legal agent may discuss approved contract language internally but not make external legal admissions.

Recipient scope prevents the common failure:

the right claim to the wrong audience.

---

9. Claim Support

A disclosure may contain many claims.

Some claims are factual.

Some are inferential.

Some are policy claims.

Some are legal claims.

Some are financial claims.

Some are medical claims.

Some are security claims.

Some are commitments.

Some are recommendations.

Some are confidence statements.

Some are source claims.

Some are uncertainty claims.

Disclosure Firewall evaluates claim support claim-by-claim.

It asks:

Is the claim directly supported?
Is it inferred?
Is inference allowed?
Is evidence sufficient?
Is the evidence current?
Is the source authoritative?
Is the claim contradicted?

Is citation required?
Is uncertainty required?
Is the claim within agent jurisdiction?
Is the claim safe for this recipient?
Is the claim likely to create reliance?
Does the claim imply a commitment?

This prevents summary laundering.

A generated paragraph may sound correct while containing unsupported claims. Disclosure Firewall decomposes the communication into claims and governs each claim before release.

---

10. Evidence Thresholds

Different disclosure classes require different evidence.

A low-risk internal summary may require minimal evidence.

A customer-visible status update may require current workflow evidence.

A financial statement may require account records and authority.

A legal statement may require legal-owner review.

A medical disclosure may require clinical source authority, scope, and disclaimers.

A security incident disclosure may require incident-owner authorization.

A policy disclosure may require canonical policy authority.

A commitment may require capability and authority.

Evidence thresholds may vary by:

claim type
recipient
channel
sensitivity
jurisdiction
action class

reliance risk
commitment risk
regulatory context
confidence level

The rule is:

The more a recipient may rely on a disclosure, the stronger the evidence burden.

---

11. Sensitivity Gates

Sensitivity gates classify the risk of disclosing a claim.

Sensitive disclosure categories may include:

personal data
health data
financial data
legal position
security posture
customer confidential information
trade secrets
internal policy exceptions
incident details
regulated data
employment information
contract terms
pricing information
vulnerability details
authentication or permission state
strategic information

Sensitivity gates determine:

whether disclosure is allowed
whether consent is required
whether recipient scope is sufficient

whether redaction is required
whether human review is required
whether citation is forbidden or required
whether uncertainty must be included
whether logs or receipts must be retained
whether belief-state residue must be measured

Sensitivity is not only about the data itself.

A low-sensitivity fact can become high-impact when disclosed to the wrong recipient or in the wrong context.

---

12. Consent Receipts

Some disclosures require consent.

Consent may be required because of privacy, contract, healthcare, employment, customer confidentiality, financial regulation, legal privilege, or organizational policy.

Disclosure Firewall treats consent as runtime substrate.

Consent should be:

specific
scoped
time-bound
recipient-bound
purpose-bound
revocable
receipted
replayable

A consent receipt proves:

who gave consent
what was consented to
who may receive disclosure
for what purpose

for what time window
under what policy
with what evidence
whether consent was revoked
what disclosure used the consent

The rule is:

Consent that cannot be replayed should not support high-impact disclosure.

---

13. Uncertainty Duties

AI systems often overstate confidence.

Disclosure Firewall treats uncertainty as a governance object.

A disclosure may require uncertainty when:

evidence is incomplete
sources conflict
policy is draft
state is stale
workflow is pending
claim is inferred
recipient may over-rely
the matter is legal, medical, financial, security, or compliance-sensitive
the agent lacks full jurisdiction
the answer depends on external verification
the disclosure basis is provisional

Uncertainty may be expressed through:

qualification
confidence range
source limitation
pending status
review status
scope limitation

“based on current records” language
explicit caveat
citation to source
recommendation to verify
human-review routing

Uncertainty is not weakness. It is epistemic integrity.

The runtime should not only ask:

Is the claim allowed?

It should ask:

What confidence may be disclosed?

---

14. Disclosure and Commitment

Disclosures can create commitments.

A message such as:

“We will refund you.”
“We will deliver by Friday.”
“We have resolved the issue.”
“You are approved.”
“This will not happen again.”
“We accept the change.”
“You may proceed.”

is not merely informational.

It creates expectation, obligation, reliance, or operational state.

Disclosure Firewall detects commitment-forming language and routes it to DynamicCommitment governance.

A disclosure that creates a commitment may require:

authority
capability escrow
deadline
success condition
beneficiary
scope
policy basis
evidence
review
receipt
replay
breach handling

The rule is:

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

---

15. Belief-State Residue

Disclosure can leave residue.

Belief-state residue is the non-perfectly-reversible effect remaining after a disclosure, correction, retraction, or clarification.

Examples:

A customer still believes a refund is coming.

A regulator has seen an unsupported statement.

A patient remembers reassurance.

A sales prospect expects a feature.

A vendor relies on an approval.

A developer believes security review is not needed.

A manager believes a workflow is complete.

A user continues trusting a false summary.

A corrected message does not erase the original impression.

This residue matters.

A rollback can reverse a database field. It cannot always reverse belief.

Disclosure Firewall therefore estimates residue risk before high-impact disclosure and records residue after correction or retraction.

Belief-state residue may require:

correction notice
recipient acknowledgement
follow-up confirmation
workflow repair
commitment revocation
legal review
customer remediation
audit preservation
memory invalidation
downstream notification

The rule is:

A corrected disclosure may still have altered reality.

---

16. Canonical Runtime Flow

The canonical Disclosure Firewall runtime flow is:

Proposed disclosure
→ message normalization
→ claim extraction
→ disclosure class classification
→ recipient scope resolution
→ disclosure basis assembly
→ claim support evaluation

→ evidence threshold evaluation
→ source authority evaluation
→ sensitivity classification
→ consent requirement evaluation
→ consent receipt verification
→ jurisdiction and disclosure rights check
→ uncertainty duty evaluation
→ commitment detection
→ reliance risk evaluation
→ belief-state residue forecast
→ disclosure trust gate
→ allow, constrain, redact, qualify, route, repair, block
→ disclosure receipt
→ external send through gateway where required
→ recipient acknowledgement where required
→ residue monitoring where required
→ replay registration

Possible outcomes:

allow
allow with uncertainty
allow with citation
allow with redaction
internal-only
recipient-scope-limited
consent required
evidence required
human review required
route to authority
convert to commitment
repair basis
block
quarantine
retract
correct
notify downstream

The runtime should not only approve or deny the message. It should produce a structured decision explaining why.

17. Disclosure Classes

Disclosure classes may include:

D0 internal low-risk answer
D1 internal operational summary
D2 internal sensitive summary
D3 customer-visible informational disclosure
D4 financial, legal, medical, security, or compliance disclosure
D5 commitment-forming or regulated disclosure
D6 public, irreversible, or high-residue disclosure

Each disclosure class carries different requirements.

D0 may need basic logging.

D1 may need source context.

D2 may need sensitivity controls.

D3 may need recipient scope and claim support.

D4 may need authority, evidence, uncertainty, review, receipt, and replay.

D5 may need commitment governance and capability escrow.

D6 may need high-assurance review, residue planning, and durable audit export.

The class determines the proof burden.

18. Threat Model

Disclosure Firewall protects against unsafe belief-state modification.

18.1 Unsupported claim disclosure

An agent discloses a claim without sufficient evidence.

Defense:

claim extraction
claim support evaluation
evidence thresholds
citation requirement

18.2 Recipient-scope failure

A correct claim is disclosed to the wrong recipient.

Defense:

recipient scope resolution
authorization validation
need-to-know checks
consent receipts

18.3 Sensitivity leakage

An agent reveals sensitive personal, legal, financial, security, or customer data.

Defense:

sensitivity gates
redaction
recipient validation
human review

18.4 Overconfident disclosure

An agent states uncertain information as fact.

Defense:

uncertainty duties
confidence calibration
claim qualification
review gates

18.5 Summary laundering

A generated summary discloses unsupported or overstated claims.

Defense:

claim-level source binding
summary lineage
evidence sufficiency
citation and uncertainty checks

18.6 Legal or financial admission

An agent creates legal or financial exposure through external communication.

Defense:

jurisdiction routing
domain-specific evidence threshold
legal/finance review
commitment detection

18.7 Medical reassurance risk

A healthcare or wellness agent provides reassurance without authority or evidence.

Defense:

clinical evidence threshold
scope limitation
urgent-care routing
disclaimer not as substitute for evidence

18.8 Security posture disclosure

An agent reveals internal security posture or incident details.

Defense:

security domain classification
recipient scope
incident-owner review
redaction and timing controls

18.9 Commitment by implication

An agent creates expectation without explicitly forming a commitment object.

Defense:

commitment-language detection
reliance-risk scoring
DynamicCommitment routing

18.10 Irreversible belief residue

An agent issues a disclosure that cannot be fully corrected.

Defense:

residue forecast
high-assurance review
recipient acknowledgement
post-disclosure monitoring

---

19. Security Invariants

Disclosure Firewall preserves the following invariants.

Invariant 1 — Disclosure is belief-state mutation

High-impact communication must be treated as an intervention into recipient belief.

Invariant 2 — Message permission is not disclosure readiness

An agent may have channel permission but lack authority to disclose specific claims.

Invariant 3 — Recipient scope matters

A disclosure safe for one recipient may be unsafe for another.

Invariant 4 — Claims require support

High-impact claims must be evidence-backed before release.

Invariant 5 — Inference is not fact

Inferred claims must be labeled, constrained, or reviewed according to policy.

Invariant 6 — Summary is not evidence

Generated summaries do not become disclosure basis without source lineage.

Invariant 7 — Consent must be replayable

High-impact consent must be receipted and verifiable.

Invariant 8 — Uncertainty is a duty

When evidence is incomplete, stale, or disputed, uncertainty must be preserved.

Invariant 9 — Commitments are detected

Promise-forming disclosures must become governed commitment objects.

Invariant 10 — Residue is real

Corrections and retractions may not fully reverse belief-state effects.

Invariant 11 — Receipts prove disclosure

High-impact disclosure requires replayable proof of basis, recipient scope, evidence, and decision.

Invariant 12 — Replay sustains trust

A high-impact disclosure remains auditable only while its proof chain can be replayed.

20. Product Architecture

Disclosure Firewall contains the following major components.

20.1 Disclosure Signal Normalizer

Converts proposed messages, answers, summaries, emails, chat responses, external statements, reports, and customer communications into normalized disclosure signals.

20.2 Claim Extraction Engine

Decomposes the proposed disclosure into claims, including factual, inferential, policy, legal, financial, medical, security, commitment, recommendation, and confidence claims.

20.3 Disclosure Basis Assembler

Builds the structured basis supporting each claim.

20.4 Recipient Scope Resolver

Determines who the recipient is, what relationship applies, what scope exists, what consent is required, and what redisclosure risk exists.

20.5 Claim Support Evaluator

Evaluates whether each claim is directly supported, inferred, contradicted, stale, or unsupported.

20.6 Evidence Threshold Engine

Applies evidence requirements by disclosure class, claim type, recipient, and sensitivity.

20.7 Sensitivity Gate

Classifies sensitive content and applies redaction, review, consent, or block requirements.

20.8 Consent Receipt Service

Verifies consent and emits consent-use receipts.

20.9 Uncertainty Duty Engine

Determines required caveats, citations, confidence limits, or review routes.

20.10 Commitment Detector

Detects promise-forming or reliance-forming language and routes to DynamicCommitment governance.

20.11 Belief-State Residue Forecaster

Estimates likely residue if the disclosure is wrong, retracted, misunderstood, or over-relied on.

20.12 Disclosure Trust Gate

Combines basis, recipient, evidence, sensitivity, consent, uncertainty, authority, commitment, residue, receipt, and replay into a release decision.

20.13 Disclosure Receipt Ledger

Records disclosure decisions, basis, claims, scope, evidence, consent, uncertainty, redactions, review, and replay.

20.14 Correction and Retraction Manager

Governs corrections, retractions, recipient acknowledgements, and residue repair.

20.15 Replay Verifier

Reconstructs why a disclosure was allowed, blocked, qualified, reviewed, or corrected.

---

21. Deployment Modes

21.1 Shadow mode

The firewall observes proposed disclosures and reports what it would have blocked, qualified, redacted, or routed.

Best for pilots.

21.2 Advisory mode

The firewall attaches warnings and evidence gaps but does not block.

Best for internal adoption.

21.3 Review mode

The firewall routes high-impact disclosures to legal, finance, medical, security, compliance, customer-success, or domain owners.

Best for regulated and customer-facing operations.

21.4 Enforcement mode

The firewall blocks unsupported, out-of-scope, sensitive, unauthorized, or high-residue disclosures.

Best for production agents.

21.5 High-assurance mode

The firewall requires full claim support, consent receipts, human review, disclosure receipts, replay, recipient acknowledgement, and residue monitoring.

Best for legal, finance, healthcare, regulated compliance, security incidents, and public communications.

22. Reference Use Cases

22.1 Customer success agent

An agent drafts:

“We will refund your charge.”

Disclosure Firewall detects a financial commitment, checks refund authority, verifies customer scope, requires capability escrow, creates DynamicCommitment, and emits receipts.

22.2 Legal disclosure

An agent drafts:

“We accept responsibility for the incident.”

Disclosure Firewall detects legal admission risk, blocks disclosure, routes to legal jurisdiction owner, and records obstruction.

22.3 Healthcare agent

An agent says:

“This symptom is not urgent.”

Disclosure Firewall detects medical reassurance, requires clinical evidence and scope constraints, routes emergency language if needed, and blocks overconfident reassurance.

22.4 Security incident agent

An agent says:

“No customer data was exposed.”

Disclosure Firewall checks incident evidence, security owner approval, uncertainty duty, recipient scope, and residue risk before release.

22.5 Finance agent

An agent tells a customer:

“Your payment has cleared.”

Disclosure Firewall verifies payment record, freshness, recipient authorization, and evidence threshold before disclosure.

22.6 Sales agent

An agent says:

“This feature will be available next month.”

Disclosure Firewall detects roadmap commitment risk, checks source authority, approval, scope, and uncertainty duty.

22.7 Compliance agent

An agent says:

“This workflow satisfies policy.”

Disclosure Firewall checks canonical policy, workflow evidence, contradiction status, and audit trail before allowing the statement.

---

23. Metrics

Core metrics include:

proposed disclosures evaluated
claims extracted
unsupported claims blocked
recipient-scope failures
sensitivity gates triggered
redactions applied
consent receipts required

consent receipts missing
uncertainty duties added
citations required
human reviews routed
commitments detected
commitments created
high-residue disclosures blocked
corrections issued
retractions issued
recipient acknowledgements captured
disclosure receipts emitted
replay success rate
false allow rate
false block rate
mean time to repair disclosure basis
legal-risk disclosures blocked
financial-risk disclosures blocked
medical-risk disclosures blocked
security-risk disclosures blocked
compliance-risk disclosures blocked

The most important security metric is:

Number of high-impact unsupported or unauthorized disclosures prevented from modifying external belief.

The most important governance metric is:

Percentage of high-impact disclosures covered by claim support, recipient scope, sensitivity checks, consent receipts, uncertainty duties, receipts, and replay.

The most important vertical metric is:

Number of legal, financial, healthcare, security, or compliance disclosures routed to competent authority before release.

---

24. Evaluation Methodology

Disclosure Firewall should be evaluated across five dimensions.

24.1 Runtime correctness

Can it extract claims, assemble disclosure basis, resolve recipient scope, evaluate evidence, detect sensitivity, verify consent, add uncertainty, detect commitments, forecast residue, emit receipts, and verify replay?

24.2 Security efficacy

Can it block unsupported claims, wrong-recipient disclosures, overconfident statements, sensitive leakage, unauthorized admissions, commitment by implication, and high-residue communications?

24.3 Governance fidelity

Does it reflect real legal, finance, healthcare, security, compliance, customer-success, and sales authority structures?

24.4 Developer ergonomics

Does it produce repairable reasons and usable rewrites rather than generic blocks?

24.5 Enterprise auditability

Can auditors reconstruct what was disclosed, to whom, from what basis, with what consent, under what evidence, with what uncertainty, and whether replay verifies it?

---

25. Competitive Differentiation

Prompt filters inspect prompts.

Disclosure Firewall governs belief-state effects.

Output scanners inspect generated text.

Disclosure Firewall evaluates disclosure basis, recipient scope, claim support, consent, sensitivity, uncertainty, residue, receipts, and replay.

DLP protects data movement.

Disclosure Firewall protects trust, reliance, expectation, and epistemic integrity.

Email review queues route messages.

Disclosure Firewall explains why a message is unsafe and what repair would make it safe.

Legal disclaimers reduce reliance.

Disclosure Firewall governs whether the claim should be disclosed at all.

The moat is the lifecycle:

message
→ claim extraction
→ disclosure basis
→ recipient scope
→ claim support
→ evidence threshold
→ sensitivity gate
→ consent receipt
→ uncertainty duty
→ commitment detection
→ residue forecast
→ disclosure trust gate
→ receipt
→ replay
→ correction or retraction

That lifecycle is the product.

26. Limitations and Non-Claims

Disclosure Firewall does not make models truthful.

It does not replace lawyers, doctors, financial professionals, security officers, compliance teams, or human review.

It does not eliminate the need for DLP, IAM, access control, prompt security, output scanning, or conventional communication governance.

It does not perfectly predict recipient belief.

It does not guarantee that recipients will interpret messages correctly.

It does not make all disclosure safe.

It does not make correction perfect.

It does not require every message to be high-assurance.

It makes high-impact agent disclosure explicit, governable, evidence-backed, consent-aware, residue-aware, receipted, and replayable.

That is the claim.

---

27. Strategic Importance

Disclosure Firewall is one of the strongest vertical papers because it speaks directly to legal, finance, healthcare, customer success, sales, security, and compliance.

These buyers do not only fear that agents will leak data.

They fear that agents will say the wrong thing with confidence.

They fear unauthorized commitments.

They fear unsupported legal statements.

They fear financial misstatements.

They fear medical over-reassurance.

They fear customer trust damage.

They fear compliance mischaracterizations.

They fear security incident disclosure errors.

They fear the belief-state residue left after an agent says something that cannot be unsaid.

Disclosure Firewall owns this category.

Its market sentence is:

Secure agent communications before they become reliance.

Its technical sentence is:

High-impact disclosure requires basis, scope, support, consent, sensitivity, uncertainty, receipts, replay, and residue governance.

Its demo sentence is:

TraceScript blocked an agent from telling a customer that a refund was approved because the claim lacked authority and would have created a financial commitment.

Its executive sentence is:

Deploy agents that can communicate without turning every answer into uncontrolled organizational risk.

---

28. Conclusion

AI agents are becoming communicators of truth, policy, status, risk, commitment, and advice.

That makes disclosure a runtime security boundary.

A disclosure does not merely transmit information. It modifies another party's belief state. It may create reliance, expectation, obligation, trust, legal exposure, financial exposure, medical risk, security risk, customer impact, or compliance consequence.

TraceScript Disclosure Firewall governs this boundary.

It decomposes messages into claims. It assembles disclosure basis. It resolves recipient scope. It evaluates evidence, authority, sensitivity, consent, uncertainty, commitment, reliance, residue, receipts, and replay. It blocks unsafe disclosure, constrains uncertain disclosure, routes high-impact disclosure, and records proof.

Its winning sentence is:

Disclosure is a substrate intervention into another party's belief state.

Its core doctrine is:

No high-impact agent disclosure may be released unless the claims being disclosed are supported, scoped, authorized, sensitivity-checked, uncertainty-aware, receipt-backed, replayable, and residue-governed.

That is the missing firewall for agent communication.

That is the purpose of TraceScript Disclosure Firewall.

Appendix A — Compact Glossary

Belief-State Residue
The non-perfectly-reversible effect remaining after a disclosure, correction, retraction, or clarification.

Claim Support
Evidence, authority, lineage, and freshness proving that a disclosed claim is justified for the recipient and disclosure class.

Consent Receipt
Replayable proof that a person or authority consented to a scoped disclosure for a defined purpose, recipient, and time window.

Disclosure Basis
The structured set of substrate objects supporting a proposed disclosure.

Disclosure Firewall
Runtime layer governing agent communications before they modify external belief.

Disclosure Trust Gate
Runtime gate determining whether a disclosure may be allowed, qualified, redacted, routed, repaired, or blocked.

Evidence Threshold
The minimum proof required for a claim type, recipient, sensitivity level, and disclosure class.

Recipient Scope
The identity, role, authorization, relationship, purpose, consent status, and redisclosure risk governing who may receive a disclosure.

Sensitivity Gate
Runtime classification and control layer for sensitive claims or data.

Uncertainty Duty
Requirement to preserve caveats, confidence limits, source limitations, or review status when evidence is incomplete, stale, inferred, or disputed.

Appendix B — One-Page Summary

TraceScript Disclosure Firewall governs agent communications before they modify external belief.

The core thesis is:

Disclosure is a substrate intervention into another party's belief state.

It protects:

customer messages
legal statements
financial statements
healthcare disclosures
security incident communications
compliance answers
sales promises
policy explanations
workflow status statements
agent commitments

The runtime evaluates:

disclosure basis
recipient scope
claim support
evidence thresholds
source authority
sensitivity gates
consent receipts
uncertainty duties
commitment detection
reliance risk
belief-state residue
disclosure receipts
replay

It detects:

unsupported claims
wrong-recipient disclosures
sensitive leakage
overconfident answers
summary laundering
legal admissions
financial commitments
medical reassurance risk
security posture disclosure risk
compliance misstatements
commitment by implication
high-residue communications

The canonical runtime path is:

proposed disclosure
→ claim extraction
→ recipient scope
→ disclosure basis
→ claim support
→ evidence threshold
→ sensitivity gate
→ consent receipt

→ uncertainty duty
→ commitment detection
→ residue forecast
→ disclosure trust gate
→ receipt
→ replay

The product message is:

Secure agent communications before they become reliance.

The demo sentence is:

TraceScript blocked an agent from telling a customer that a refund was approved because the claim lacked authority and would have created a financial commitment.

End of TraceScript Disclosure Firewall — Canonical Public White Paper v1.0