Beyond Guardrails

Runtime Execution Governance for Trustworthy Agentic AI in the Software Supply Chain

SSCA Forum
Sept. 22, 2026

John M. Willis
Sustainable Future Tech, Inc. | AGCP.ai

Software and Supply Chain Assurance Forum

AI is already being applied across supply-chain assurance

Representative examples — not an exhaustive survey

DHS / CISA

- AI illumination
- Trade and supply-chain analytics
- Software supply-chain efforts

DoD / DoW

- Supply-chain risk management
- SCREEn illumination
- CDAO / DIU GenAI initiatives

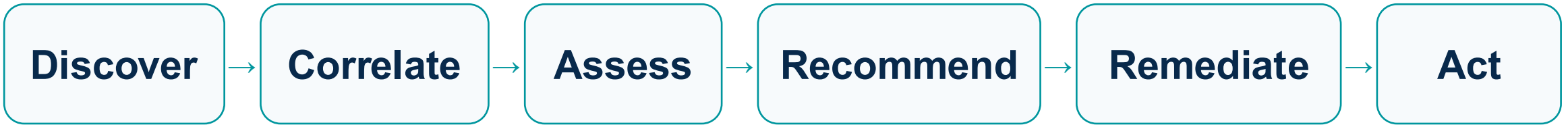
Academia

- LLM-assisted vulnerability analysis
- Code review
- Agentic defense research

Level-set: AI is moving from visibility and analysis toward remediation and action.

The direction of travel

From supply-chain illumination to operational consequence



The architecture changes when analysis becomes a proposed state change.

Agentic AI changes the supply-chain problem

Agents increasingly act across the pipeline — not just analyze it



Software supply-chain assurance becomes an execution-assurance problem.

The core architectural distinction

Probabilistic reasoning and deterministic authority must not collapse into one function

Probabilistic / agentic components

- Interpret
- Infer
- Summarize
- Correlate
- Prioritize
- Recommend
- Propose

Deterministic control components

- Parse
- Validate
- Verify
- Bind
- Resolve authority
- Evaluate policy
- Commit or block

Where LLMs and agents are useful

Powerful reasoning components — not final authorities

- Interpreting complex evidence
- Correlating weak signals across tools
- Summarizing vulnerabilities, dependencies, and incident context
- Prioritizing remediation alternatives
- Generating analyst-readable rationale
- Drafting proposed actions for governed review

LLMs can help explain and propose. They should not be treated as execution authority.

Where deterministic logic must remain in control

Consequential execution requires control, not merely reasoning

Evidence mechanics

- Schema validation
- Hash and signature checks
- Certificate-chain and revocation checks

Governance mechanics

- Policy evaluation
- Authority resolution
- Target, tenant, scope binding
- Invariant checks

Execution mechanics

- Permit / refuse / escalate / hold
- Commit-time revalidation
- Receipt and evidence record

Deterministic logic appears inside PBSAI agents and inside AGCP commit governance.

Not every guardrail controls the crossing

The same word often hides different control functions

Guardrail	A speed-limit sign is a guardrail.
Guardrail	A flashing warning before a bridge is also a guardrail.
But	Neither determines whether a specific vehicle may cross under current conditions.

AI Governance at the Point of Action

System Guardrails, Execution-Adjacent Guardrails, and Runtime Execution Governance

1 System Guardrails

General safety measures across the environments

- Speed Limits
- Lane Markings
- Monitoring
- Safety Standards
- Road Barriers

2 Execution-Adjacent Guardrails

Warnings and checks near the consequential point

- Rumble Strips
- Flashing Warn Lights
- Weigh Limit
- Caution Markers
- Bridge Ahead

3 Runtime Execution Governance

The boundary that controls whether the action may occur



Not every guardrail controls the crossing.



The gate must be non-bypassable.

Runtime Execution Governance controls consequence

The boundary function is different from warning, scoring, or logging

Evaluate

- Current authority
- Qualified evidence
- Applicable policy
- Canonical state

Bind

- Target
- Tenant
- Scope
- Purpose

Enforce

- Permit
- Refuse
- Escalate
- Hold

This is not a recommendation. It is an enforceable decision at the boundary.

The missing question

The unit of governance is the proposed consequence

Not just Is the model safe?

Not just Is the artifact signed?

But May this transition commit?

Supply-chain evidence is necessary — not sufficient

Evidence supports governance; it does not automatically authorize consequence

Evidence may include

- SBOM
- SLSA attestations
- Sigstore / Rekor
- in-toto provenance
- NVD / OSV / KEV
- IAM and policy context

Runtime governance must

- Qualify evidence
- Bind evidence to proposal
- Evaluate current authority
- Check policy and invariants
- Enforce the resulting decision

Evidence can support a decision. It does not automatically authorize the action.

The commit boundary

Before: candidate transition. After: authoritative state or external effect.



The commit boundary is where a proposal becomes operationally real.

MCP and A2A are interaction paths — not commit paths

All consequential writes and commits go through AGCP

A2A

- Agent-to-agent collaboration
- Task coordination
- Evidence exchange
- No execution authority

MCP

- Agent-to-tool interaction
- Read / inspect / retrieve
- Bounded task support
- No direct commit path

AGCP

- Sole governed path
- Deterministic governance
- Commit-boundary mediation
- Controls target commits

A2A and MCP move context, evidence, and proposals. AGCP alone governs consequential commits.

PEOPLE
AGENTS
GOVERNANCE
SAFER TOMORROW

PBSAI Cybersecurity Multi-Agent Ecosystem on AGCP

Specialized agents. Shared insight. Stronger security.

MULTI-AGENT
INTELLIGENCE.
SAFER ACTIONS.
STRONGER OUTCOMES.

PBSAI Cybersecurity Multi-Agent Ecosystem

12 DOMAINS | COLLABORATIVE AGENTS | BROADER DEFENSE

SPECIALIZED
AGENTS.
SHARED INSIGHT.
STRONGER
SECURITY.

- A — GRC & Oversight
- B — Asset, Configuration & Change Management
- C — Identity, Credential & Access Management (ICAM)
- D — Threat Intelligence & Monitoring
- E — Protect & Harden
- F — Data Security & Privacy
- G — Incident Response & Digital Forensics (IR/DFIR)
- H — Resilience & Business Continuity
- I — Security Architecture & Systems Engineering
- J — Physical & Environmental Security
- K — Supply Chain & Lifecycle Security
- L — Security Program Enablement, Knowledge & AI Validation

Multi-agent coordination • evidence production • analysis • recommendations • proposed actions

PBSAI proposes; AGCP governs the commit.

AGCP — AI Governance Control Plane

Deterministic runtime governance of consequential execution

FROM INSIGHT
TO CONTROL.

DETERMINISTIC
GOVERNANCE
FOR TRUSTED
EXECUTION.

Proposal qualification → Evidence qualification → Authority resolution → Policy evaluation → Invariant checks → Commit-time revalidation → Decision: PERMIT / REFUSE / ESCALATE / HOLD

Only approved actions cross the commit boundary.

CONTROL TODAY.
CONFIDENCE TOMORROW.

Commit Boundary

POLICY ENFORCED | AUDITABLE | REVERSIBLE

Execution Target

Approved actions executed on target systems

CI/CD pipeline | deployment platform | identity system | network control | other operational systems

AUTHORIZED
ACTIONS.
REAL-WORLD
IMPACT.

Architecture overview — PBSAI proposes; AGCP governs the commit.

A SAFER TOMORROW, TOGETHER.

PBSAI domains produce evidence and proposals

The multi-agent ecosystem structures the work before governance at the boundary

- A–L domains define role boundaries and responsibilities
- Agents collaborate through A2A when agent-to-agent work is needed
- Agents retrieve evidence and context through MCP when tool or resource access is needed
- Output Contracts carry structured evidence, recommendations, and proposed actions
- Proposed actions are not commits

PBSAI output can support a commit decision; it does not itself authorize the commit.

AGCP is the only commit path

The governed path to the execution target is intentionally narrow

- Consequential proposals are submitted to AGCP
- AGCP evaluates authority, evidence, policy, state, and invariants
- Commit-time eligibility is rechecked before target execution
- Only AGCP-approved actions reach the execution target
- MCP and A2A must not provide alternate write paths

Only AGCP-approved actions cross the commit boundary.

PBSAI: open multi-agent reference architecture

A structured model for AI-enabled cybersecurity work

PBSAI defines

- Domain responsibilities
- Agent roles and boundaries
- Context envelopes
- Output contracts

PBSAI agents do

- Observe
- Retrieve evidence
- Analyze
- Collaborate
- Recommend
- Propose

PBSAI agents do not

- Self-authorize consequence
- Bypass AGCP
- Commit external writes via A2A/MCP
- Replace runtime governance

PBSAI organizes agent work. AGCP governs consequential execution.

PBSAI Domain K: software and supply-chain functions

Domain K specializes the multi-agent ecosystem for supply-chain assurance

K1 Procurement

- Sourcing requirements
- Supplier constraints
- Procurement evidence

K2 Authenticity

- Verify hashes and signatures
- Validate certificates
- Check attestations and provenance

K3 SBOM Risk

K4 Lifecycle

- Parse SBOMs
- Compute risk and policy status
- Govern end-of-life evidence

Domain K produces supply-chain evidence and proposed actions; AGCP governs commits.

PBSAI operating profiles and agent boundaries

Agent behavior is bounded by context, constraints, provenance, and output contracts

Context Envelope

- mission_id
- thread_id
- task_id
- role / intent
- policy_refs
- constraints

Agent boundary

- Role
- Input schemas
- Permitted retrieval
- Deterministic logic
- Bounded AI
- HITL triggers

Output Contract

- Typed evidence
- Evidence refs
- Decision basis
- Provenance
- Signature
- No commit authority

The agent is a governed software worker, not an unconstrained model with tools.

Where deterministic code sits on the PBSAI side

Inside agent implementation, orchestration, adapters, validation, and output handling

- Deterministic intake and schema validation
- Deterministic adapters: retrieve, inspect, normalize
- Deterministic rules, checks, graph traversal, and scoring
- Optional bounded LLM / ML reasoning for judgment or explanation
- Deterministic post-validation, constraints, signing, and output handling
- Output Contract: evidence, recommendation, or proposal

PBSAI logic may retrieve and validate evidence; consequential execution is externalized to AGCP.

PBSAI Context Envelope over A2A

A2A carries PBSAI context; it does not replace PBSAI semantics

mission_id	release-2026-091
thread_id	widget-4.7.2
task_id	k2-auth-check-017
role / intent	AuthenticityVerificationAgent / verify release candidate
policy_refs	SSC-Policy-v4; Promotion-Policy-v2
constraints	no_direct_promotion; trusted_publishers_only
evidence_refs	artifact; SBOM; prior reports
provenance	requesting and receiving agent identity

Output Contracts as A2A Artifacts

A2A transports PBSAI outputs; it does not authorize consequence

PBSAI Output Contract

- Typed result or evidence
- Decision basis
- Evidence references
- Provenance
- Signature

A2A Artifact

- Transports the output
- Preserves context
- Carries updated envelope
- Supports evidence exchange
- Does not authorize execution

Consequential proposals go to AGCP.

What may change before commit

Distributed environments and long-lived proposals can drift before commit

Authority and evidence

- Operator role or delegation
- Approval or session state
- Signature or attestation freshness
- SBOM or scan result

Risk and policy

- Certificate or signer revocation
- KEV / NVD / OSV updates
- Policy version or waiver
- Risk acceptance or freeze window

Target and operations

- Artifact, tenant, environment
- Route, dependency, topology
- Incident or outage state
- Maintenance or legal hold

Consequences if not rechecked

Stale assumptions can become real operational effects

- Stale authority: previous approval becomes an unauthorized commit
- TOCTOU exposure: checked state differs from committed state
- Policy violation: new constraints are missed at consequence time
- Wrong scope: wrong target, tenant, route, or environment
- Bypass: an alternate path creates consequence outside governance
- Weak auditability: no proof of current authority at commit

Initial authorization is a checkpoint. Commit-time authorization is boundary control.

AGCP: open governance control-plane model

Deterministic runtime governance for consequential execution

AGCP compiles & evaluates

- Machine-readable policy
- Constitutional invariants
- System invariants
- Proposal qualification

AGCP enforces

- Commit-time eligibility
- Permit / refuse / escalate / hold
- Commit-bound mediation
- Receipts and evidence

AGCP prevents

- Stale authorization
- Unbound evidence
- Bypass paths
- Probabilistic self-authorization

PBSAI proposes; AGCP determines whether the proposed consequence may occur.

AGCP deterministic workflow

All proposed operational writes and commits are submitted before execution



The assurance claim is not checked once. It is still authorized when it matters.

Decision outcomes at the boundary

Enforceable decisions — not warnings, scores, or logs

PERMIT

- Commit action
- Bind exact proposal
- Record receipt

REFUSE

- Block action
- Record basis
- Prevent crossing

ESCALATE

- Route for authority
- Require adjudication
- Preserve context

HOLD

- Pause pending state
- Await evidence or approval
- Re-evaluate before commit

The decision must control the path capable of producing the governed consequence.

Standard enforcement message to the PEP

Canonical AGCP schemas separate governance meaning from target-specific enforcement



Representation can change; governance meaning cannot.

What DS-029 Enforcement Context binds

The PEP receives an integrity-protected governance basis before target-specific commit

- Proposal, action, target, and tenant
- Decision and execution authorization
- Current authority and qualified evidence
- Canonical state and resulting-state validation
- Policy, scope, validity, and lifecycle
- Governance version and enforcement directive

Adapter maps canonical fields to the native PEP interface.

Worked example: should this artifact be promoted?

Promotion is an execution decision, not just an artifact fact



K2 and K3 do not directly write to the deployment platform. The promotion goes to AGCP.

Worked example: Context Envelope values

mission_id	release-2026-091
thread_id	widget-4.7.2
task_id	k2-auth-check-017
role	AuthenticityVerificationAgent
intent	verify_release_candidate
policy_refs	SSC-Policy-v4; Promotion-Policy-v2
constraints	no_direct_promotion; trusted_publishers_only

The context constrains agent behavior before any output is produced.

Worked example: K2 AuthenticityReport values

K2 produces evidence, not an execution command

artifact	widget-4.7.2
digest_match	TRUE
certificate_chain	VALID
revocation_status	CLEAR
signature	VALID
rekor_entry	FOUND
slsa_provenance	VALID
publisher_identity	MATCH
status	VERIFIED
evidence_ref	auth-report-9271

Worked example: K3 SBOMRiskReport values

K3 produces risk evidence and a review recommendation

sbom	cyclonedx-4451
components	138
critical_vulns	0
high_vulns	2
kev_matches	0
fix_versions_available	TRUE
policy_threshold_met	TRUE
recommendation	ELIGIBLE_FOR_PROMOTION_REVIEW
evidence_ref	sbom-risk-4482

Worked example: PBSAI ProposedAction OC

A proposal is not a commit

action	PROMOTE
artifact	widget-4.7.2
target	production
scope	service-A
tenant	enterprise-1
supporting_evidence	auth-report-9271; sbom-risk-4482
requested_by	K3 SBOMAnalysisAgent
execution_authority	NONE

Worked example: AGCP evaluation and commit decision

Only an AGCP-approved action crosses into the execution target

AGCP evaluates

- Proposal
- Evidence
- Authority
- Policy
- State
- Invariants
- Commit-time status

Commit decision

- PERMIT
- REFUSE
- ESCALATE
- HOLD

The commit decision is always written to the ledger.

What changes for software supply-chain assurance?

Artifact assurance becomes execution assurance

Artifact assurance

- What do we know about this software?
- Is the artifact signed?
- Is provenance present?
- What does the SBOM say?

Execution assurance

- May this consequence occur now?
- Is current authority valid?
- Is evidence qualified?
- Is the target state current?

Runtime governance composes with existing supply-chain assurance; it does not replace it.

Five takeaways

The consequence-producing boundary must be controlled

- Agentic AI creates execution risk, not just output risk
- Supply-chain evidence is essential, but evidence is not authority
- PBSAI structures roles, evidence production, and proposed actions
- A2A and MCP support interaction; they are not commit paths
- AGCP supplies deterministic runtime governance for consequential execution

Can the action cross without current, valid, enforceable authority?

Questions & discussion

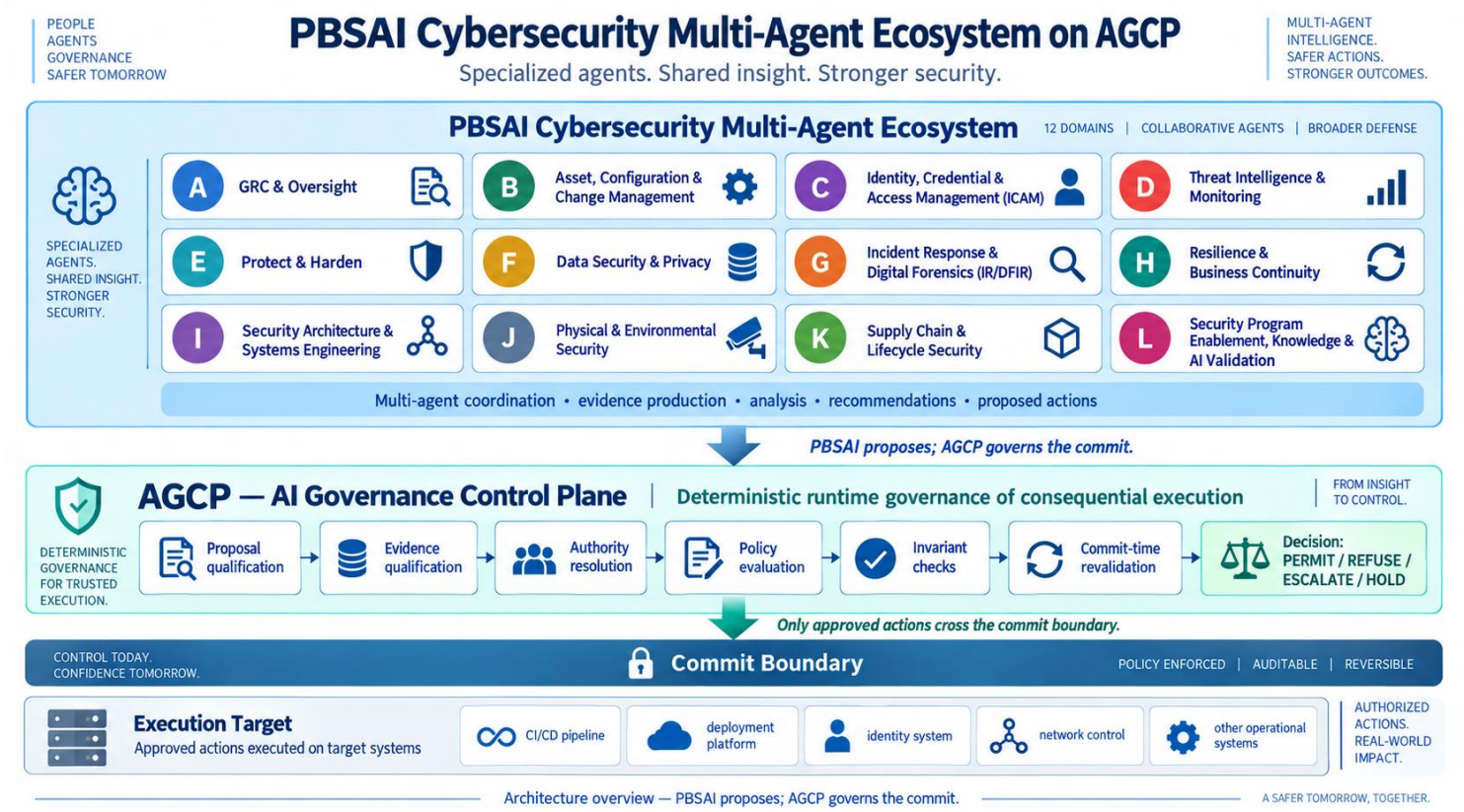
Can this consequence occur now?



Discussion prompts: latency • bypass coverage • PDP/PEP • TOCTOU • evidence freshness • A2A/MCP boundaries • assessment

Closing perspective

PBSAI structures the multi-agent ecosystem. AGCP governs consequential commits at the execution target.



Open reference resources

- PBSAI Reference Architecture: github.com/jwillisSFT/pbsai-reference-architecture
- AGCP Specification: github.com/jwillisSFT/agcp-spec
- AGCP Website: agcp.ai

A2A and MCP support collaboration and evidence flow; consequential proposals go to AGCP.

Contact

Follow-up and technical discussion

John M. Willis

Sustainable Future Tech / AGCP.ai

9841 Washingtonian Blvd, Suite 200
Gaithersburg, MD 20878

(240) 720-7515
jwillis@agcp.ai
https://agcp.ai

Backup: Representative starting references

For attendees who want to go deeper after the session

DHS / CISA

- Software Supply Chain AI Illumination
- DHS AI Task Force examples
- CISA C-SCRM work

DoD / DoW

- DLA logistics and risk management
- Advana / SCREEn illumination
- CDAO / DIU GenAI initiatives

Academia / research

- NYU Software Supply Chain Security Operations Center
- LLM-assisted code and vulnerability analysis
- Agentic software-security research

Starting points, not a complete survey.

Backup: PDP/PEP vs runtime governance

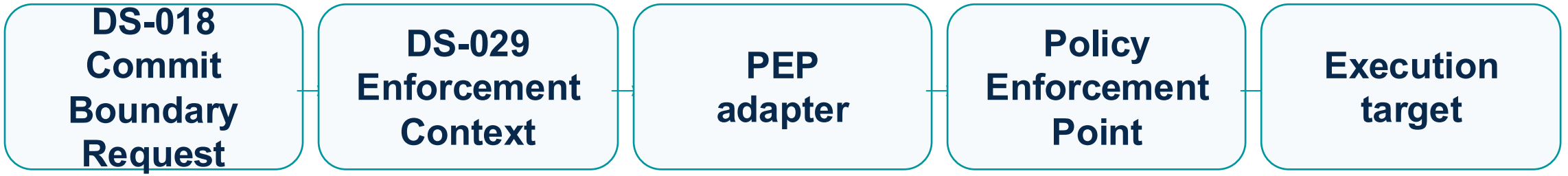
Useful pattern, incomplete semantics

- PDP and PEP are useful architectural patterns
- Runtime governance adds explicit proposal semantics
- Runtime governance qualifies evidence and canonical state
- Runtime governance binds current authority to the proposed consequence
- Runtime governance requires commit-time revalidation
- Runtime governance records receipt or refusal evidence

Runtime governance adds semantics around the proposed consequence.

Backup: PEP-facing message contract

Standard message format adapted to target enforcement points



Adapter can change representation. It must not change governance meaning.

Backup: TOCTOU and revocation

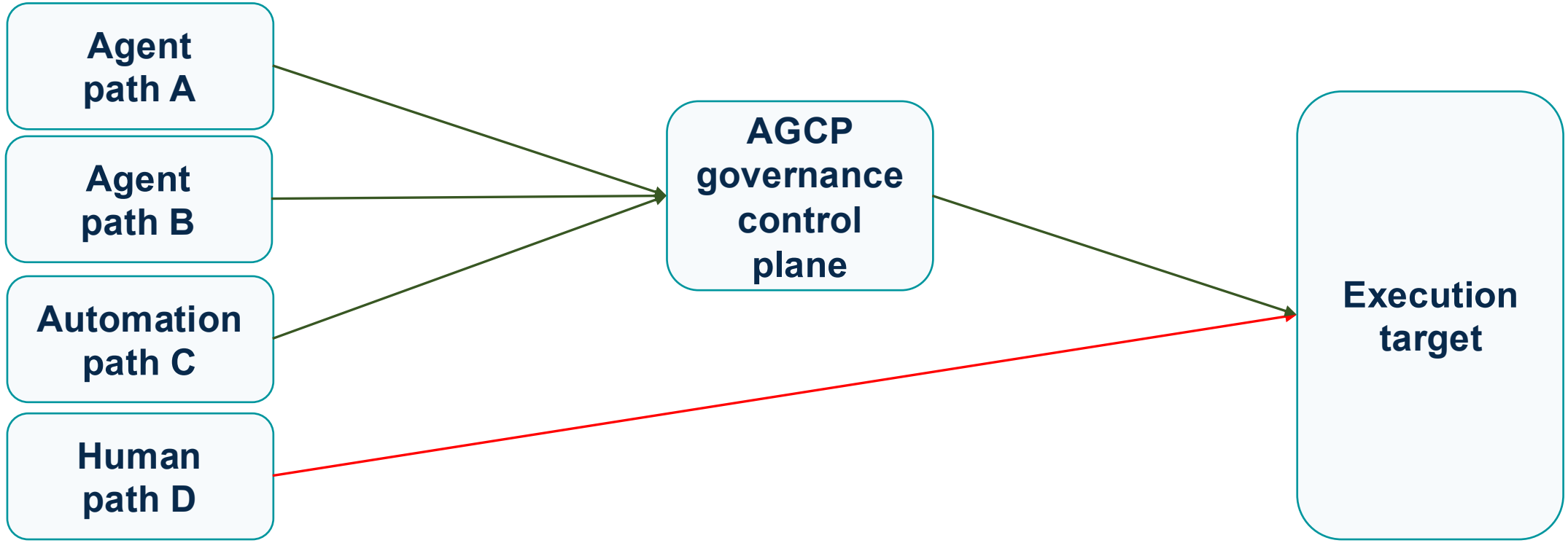
Earlier approval is not necessarily current authority at commitment



Examples: signer revoked • KEV updated • exception expired • target changed • authority changed

Backup: Bypass coverage

Every path capable of producing the governed consequence must be mediated



MCP and A2A must not provide alternate direct write paths to target systems.

Backup: K2 + K3 evidence details

The strongest worked example is promotion / deployment gating

K2 Authenticity Evidence

- X.509 chain
- Revocation status
- Artifact digest
- Publisher signature
- Sigstore / Rekor record
- in-toto / SLSA attestation

K3 SBOM Risk Evidence

- CycloneDX / SPDX SBOM
- purl / CPE normalization
- NVD / OSV matches
- KEV status
- Fix versions
- Risk score / recommendation

Evidence supports the AGCP decision; it does not itself authorize promotion.

Backup: Assessment and conformance

Governance architecture must itself be assessable

Assess

- Architecture claim
- Boundary mediation
- Bypass coverage
- Evidence qualification
- Authority resolution

Test

- Schema validity
- Policy evaluation
- Invariant behavior
- Commit-time revalidation
- Receipt / ledger evidence

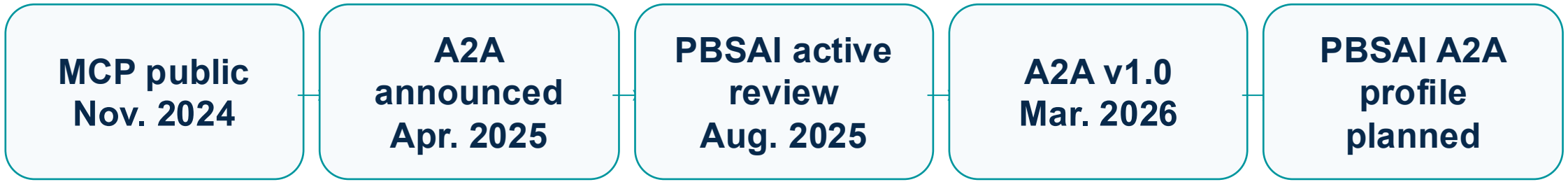
Demonstrate

- Permit cases
- Refuse cases
- Escalation cases
- Hold cases
- Replay / auditability

Trustworthy execution governance cannot depend only on claims.

Backup: Protocol evolution and PBSAI update path

PBSAI predates A2A maturity; the repo is MCP-oriented today



MCP for tools/resources; A2A for agent-to-agent; AGCP for consequential commits.