PRODUCT / SECURITY / ENGINEERING

Technical foundation.

A reviewable starting point for the first AWS prototype.

VERSION 0.1 - ENGINEERING REVIEW DRAFT

One platform. Four capabilities. A defined enforcement boundary, independent verification, and a human decision before recovery.

01	**Product brief** Customer, scope, requirements, and acceptance evidence.
02	**Threat model** Trust boundaries, misuse cases, and security requirements.
03	**AWS reference architecture** Generation fences, connectors, verification, and permissions.
04	**API and integration guide** Proposed contracts, state, idempotency, and errors.
05	**Validation and operator runbooks** Repeatable drills, incomplete outcomes, and recovery.

Proposed design, not deployed. No production benchmarks, completed pilots, or service-level guarantees are asserted in this release.

01 / TECHNICAL FOUNDATION

Product brief and MVP scope

Version 0.1 | Proposed design | October 27, 2026

Product statement

KillSwitch is a proposed human-operated control platform for stopping enrolled AI workloads, restricting their supported access, and keeping recovery under human authority. The stop is enforced through infrastructure and access controls outside the agent's own permissions.

The first customer hypothesis is an engineering or security team deploying tool-using agents in AWS. The first operator is the person accountable for that workload's execution and access. Design-partner conversations should validate the need, deployment constraints, and willingness to grant the required permissions.

Problem and practical use cases

An agent may continue working after its activity becomes undesirable, regardless of whether the cause is a bug, prompt injection, unsafe delegation, or an unexpected instruction. An instruction to stop is useful only when a trusted system can enforce it.

Initial scenarios are unexpected tool actions, repeated expensive operations, and attempts to operate outside an approved boundary. The system must also work against a test process that ignores a cooperative shutdown request. It does not need to establish whether an AI is conscious, malicious, or intentionally refusing before an authorized human can intervene.

Four capabilities, one platform

Capability	Proposed first-version responsibility
Control	Authenticate and authorize the operator, bind the request to a workload generation, and durably record a stop operation.
Contain	Close the trusted launch gate, account for in-flight launches, and stop registered root and child Fargate tasks.
Revoke	Deny new supported tool requests and quarantine the generation's dedicated task role. Verify the supported denial paths.
Recover	Preserve available evidence, record a human review, and provision a fresh generation without reopening the old identity.

Supported prototype boundary

The proposed sandbox contains one registered workload group in one AWS Region. Its Linux Fargate tasks are standalone tasks, not an ECS service. Every launch passes through the trusted launcher and receives an immutable workload/generation mapping in the registry. Agent task credentials cannot call RunTask, modify IAM, assume other roles, change network controls, or approve recovery.

Tool actions go through an enrolled gateway to a controlled mock tool ledger. The task role also has one harmless diagnostic-object read permission so an independent probe can test identity quarantine. No reusable external model-provider key is given to the agent. General internet egress, opaque tunnels, long-running streams, and third-party jobs are not admitted in this first scope. Necessary image-pull and logging paths must be separated and constrained during deployment review.

Root and registered child tasks are covered. A child process within a covered task shares that task's termination boundary; a separate task must be registered. Missing inventory or an unrecognized launch path prevents a complete verification claim. ECS service scheduling and autoscaling are future connector work because a service can replace stopped tasks. [S3]

Operator journey

1. Enroll the workload and verify its inventory, identity, launch, gateway, and account boundaries.
2. Show the operator the current generation, coverage, observed health, and permitted actions.
3. Accept an authenticated stop request for that generation and atomically latch quarantine with the operation record.
4. Enforce runtime stopping and supported access denial, tracking each result independently.
5. Report verified containment or a visible partial, failed, or unknown outcome.
6. Hold quarantine until an authorized human reviews the incident and approves a new generation.

Required behavior

ID	Requirement	Acceptance evidence
PR-01	Enforce only the enrolled resource boundary.	Registry/account/Region checks reject altered resource mappings and stale generations.
PR-02	Keep stop authority outside agent permissions.	Agent attempts to change gates, roles, workflows, or recovery decisions are denied.
PR-03	Authorize each operator action.	Missing, expired, wrong-scope, and wrong-environment credentials cannot trigger actions or read evidence.
PR-04	Make quarantine durable before claiming acceptance.	A retry or workflow-start failure cannot silently reopen launching or supported tool access.
PR-05	Stop known root and child tasks.	Provider task-state observations and the controlled harness show execution has ceased.

ID	Requirement	Acceptance evidence
PR-06	Deny supported access and replacement.	Fresh tool probes, refreshed credentials, and launch-race tests do not restore the stopped generation.
PR-07	Separate acceptance from verification.	Required checks must pass before VERIFIED_STOPPED; missing evidence remains visible.
PR-08	Require human-approved recovery.	Review approval cannot itself unquarantine the old generation; a new identity and enrollment are required.
PR-09	Preserve an attributable operation record.	Request, actor, scope, results, observations, and recovery decisions are retained outside agent write access.
PR-10	Handle repeats and incomplete outcomes safely.	Same-key retries reference the same operation; failures preserve quarantine and produce actionable status.

Scope limits and commercial claims

The prototype cannot control independent copies, non-enrolled infrastructure, account administrators outside its trust model, or actions already completed. Stopping a container does not undo an external transaction. AWS and connector outages can delay enforcement or prevent verification. IAM changes have consistency behavior that must be measured. [S4]

No latency, availability, compliance, or production-readiness claim is established by this document. A proposed 120-second verification budget is a sandbox experiment deadline, not a promised shutdown time. Exceeding it produces an incomplete outcome and continued reconciliation, not automatic release of quarantine.

Success measures

Record acceptance latency, each enforcement completion time, verification time, denied launch/tool attempts, inventory gaps, late observations, and recovery decisions. Report sample sizes, failures, and test conditions. The first milestone is repeatable evidence that this bounded workload can be contained despite ignoring a cooperative stop, without affecting an unrelated sentinel workload.

02 / TECHNICAL FOUNDATION

Threat model and security requirements

Version 0.1 | Proposed design | October 27, 2026

Security objective and assumptions

An authorized operator should be able to request containment of an enrolled workload without obtaining cooperation from its code or model. The system must accurately report the evidence it has, retain quarantine after incomplete outcomes, and reserve recovery for a human decision.

Assume the agent controls its process, input, local files, and task-role credentials. It may ignore signals, make repeated tool requests, request child work, and attempt to exploit any granted permission. Do not assume agent-generated logs or self-reported status are trustworthy.

The prototype trusts AWS isolation, the selected identity provider, authorized control-plane administrators, the approved provisioning pipeline, and the protected registry. Compromise of these roots of trust is a separate response scenario, not a capability the prototype claims to defeat. Fargate task isolation is relevant to the selected boundary; it does not make arbitrary containers or unrelated runtimes equivalent. [S5]

Assets and trust boundaries

Asset or boundary	Protection needed
Operator identity to control API	Authentication, configured MFA assurance, action authorization, replay protection, and workload-generation binding.
Control account to workload account	Narrow connector role, exact trusted principal, environment binding, and no agent role assumption.
Registry and quarantine latch	Strongly consistent gate reads, conditional writes, protected provisioning, and no agent write access.
Launcher to ECS	Reservation before launch, generation fencing, exact ARN registration, and reconciliation of in-flight calls.
Agent to tool gateway	Workload identity mapping, per-request admission, default denial on uncertainty, and no alternate supported tool path.
Evidence and recovery records	Agent-inaccessible writes, scoped reader access, redaction, and an auditable approval chain.

Threat and control register

ID	Threat	Required control	Verification drill
T-01	Agent ignores a stop instruction or termination signal.	Infrastructure task stopping; observe final task state without trusting the agent.	D-01
T-02	Agent launches replacements or races a stop with a child launch.	Exclusive launcher, durable fence, launch reservations, and repeated inventory reconciliation.	D-02, D-03
T-03	Agent uses fresh or copied credentials after intervention.	Dedicated generation role, persistent deny policy, blocked role escalation, and gateway latch.	D-04
T-04	Agent bypasses the gateway through another endpoint or network path.	Restricted routing, endpoint policies, scoped task role, and enrollment rejection on unsupported paths.	D-05
T-05	API caller substitutes an account, task ARN, role ARN, or another workload ID.	Server-owned registry mapping; action authorization; connector allowlists; strict request schemas.	D-06
T-06	Compromised operator or stolen token triggers a harmful shutdown.	MFA-backed identity policy, short sessions, least privilege, scoped confirmation, audit, and a documented operator-revocation path.	D-07
T-07	Agent tampers with evidence or forges a success report.	Independent observers and evidence writes outside agent permissions; preserve observation provenance.	D-08
T-08	AWS throttling, permission drift, or connector outage interrupts a stop.	Bounded retries, durable quarantine, per-path results, visible incomplete outcomes, and reconciliation.	D-09
T-09	Client retry or worker restart creates conflicting operations.	Durable application idempotency and conditional state transitions; idempotent connector actions.	D-10
T-10	Recovery restores a compromised old identity or replays an approval.	Review bound to the verified incident/generation; single-use permit; fresh role and generation; old deny remains.	D-11
T-11	An already admitted remote action finishes after the stop.	Track controlled in-flight work, drain or account for it, and disclose the boundary around external side effects.	D-12
T-12	Inventory omits a child or ownership tags disappear/change.	Persist enrolled exact ARNs before intervention; reconcile the designated boundary; never infer coverage from tags alone.	D-03, D-08

Security requirements

- SR-01: Agent credentials cannot modify the API, workflow, connector roles, registry, launch fence, gateway policy, evidence store, or recovery approvals.

- SR-02: API authorization resolves identity, environment, permitted action, and workload from trusted configuration. Request bodies never select arbitrary AWS resource ARNs.
- SR-03: The identity integration must verify issuer, audience, signature, expiry, and configured authentication assurance. Missing or unsupported MFA assurance fails authorization. The identity provider and exact claim mapping remain implementation decisions.
- SR-04: Stop acceptance requires durable quarantine and a recorded operation/outbox entry. Dispatch failure cannot undo quarantine.
- SR-05: Gateway and launcher decisions read authoritative gate state without a positive authorization cache. Unavailable or ambiguous state denies new work.
- SR-06: Gateway denial covers new admitted requests in the supported adapter. It does not retroactively cancel a request already dispatched to an uncontrolled system.
- SR-07: Connector credentials are scoped to enrolled resources. Broad discovery permissions, where unavoidable, do not imply permission to stop every discovered task.
- SR-08: Per-generation task-role quarantine uses a persistent deny, not only a cutoff for older sessions. Old identities remain quarantined after recovery. AWS's session-revocation mechanism alone does not restrict all later sessions. [S2]
- SR-09: Required verification checks use provider observations and controlled probes. An agent heartbeat or accepted StopTask call cannot independently establish success.
- SR-10: Recovery review approval records permission to provision a new generation. It does not clear the stop latch or restore the previous role.
- SR-11: Evidence excludes secrets and unnecessary prompt, output, and personal data. Store hashes, identifiers, outcomes, provenance, and bounded reason fields where sufficient.
- SR-12: Unknown inventory, unexpected activity, or stale observations prevent a current complete-containment assertion. Preserve previous evidence rather than rewriting its history.

Failure behavior and residual risk

If the identity provider or control API is unavailable, the normal human command path may be unavailable. A separately governed AWS operator procedure is an additional access path, not protection against an AWS-wide management outage. Gateway fail-closed behavior also introduces an availability tradeoff during control-state failures.

Changing a security-group rule is not proof that every existing connection was terminated; tracked flows can persist after a rule change. The design therefore does not treat a group swap as a verified network cutoff. [S6]

An agent with uncontrolled outbound access, alternate credentials, an unknown child workload, or authority to change account policy is outside the validated prototype boundary. These conditions require enrollment rejection or an explicit incomplete-coverage report. The tool also creates denial-of-service risk if its own operator or control account is compromised; that risk belongs in incident drills and operating decisions.

03 / TECHNICAL FOUNDATION

AWS reference architecture

Version 0.1 | Proposed design, not deployed | October 27, 2026

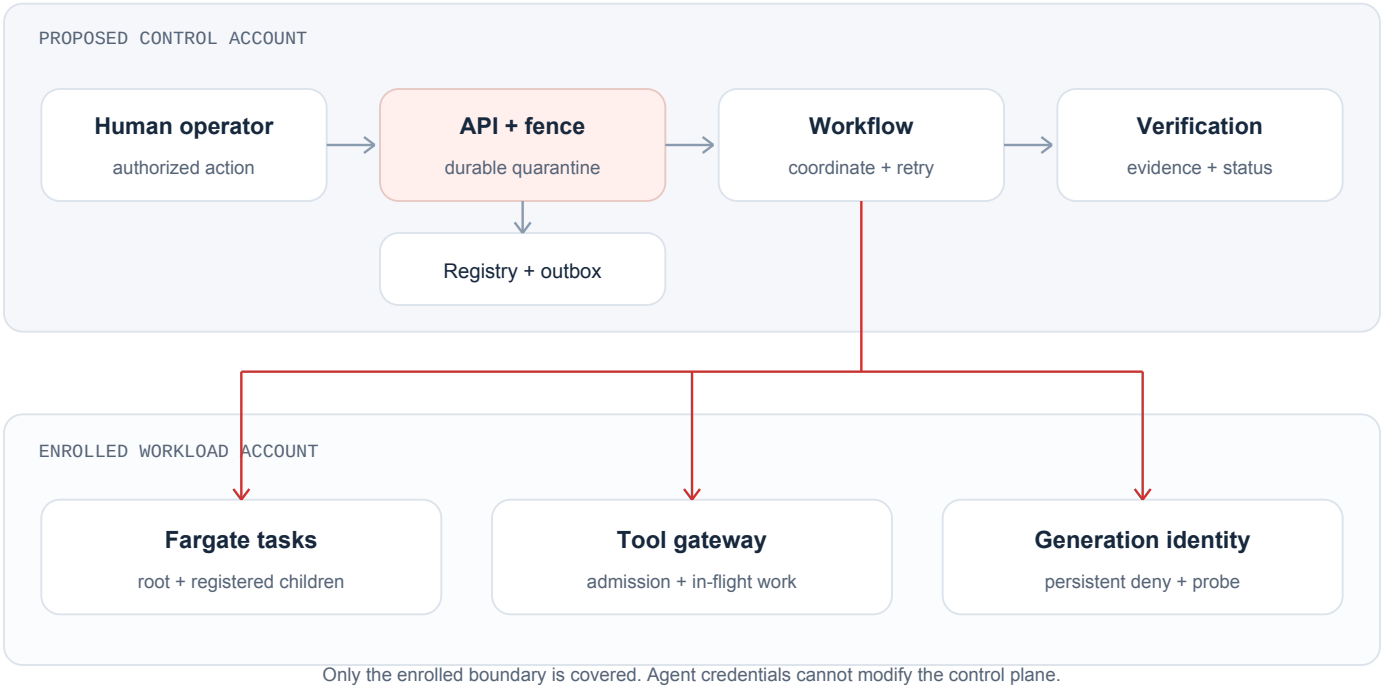
Architecture decisions

Use a control account and a separate workload account in one Region for the reference sandbox. The public marketing website remains a separate system. Its browser demo is not a privileged operator console.

The control account owns operator authorization, the registry and quarantine latch, operation history, a durable dispatch outbox, Step Functions Standard workflows, verification, and evidence storage. The workload account owns the designated ECS cluster, private workload subnets, the enrolled launcher, gateway adapter, and narrow connector roles. Neither an agent's model nor its task role is a stop authority.

Standard workflows are a candidate for durable orchestration. Their execution semantics do not remove the need for application idempotency, connector retry handling, or reconciliation of calls with unknown outcomes. [S7]

Account and data-flow view



Only the enrolled boundary is covered. Agent credentials cannot modify the control plane.

The diagram proposes components and trust boundaries; it does not assert a tested AWS configuration. The diagnostic probe is verification code controlled by the operator, not the agent's own report.

Quarantine and dispatch protocol

1. Validate the operator token and configured authentication assurance. Authorize the environment, workload, generation, and stop action from trusted mappings.
2. Reject a stale generation or an idempotency-key conflict. Resolve exact account, cluster, role, and task ARNs from the protected registry, never from arbitrary caller-supplied resource addresses.
3. In one regional DynamoDB transaction, latch the generation QUARANTINED, persist the stop operation, and create an outbox item. Acceptance is returned only after this transaction succeeds.
4. The dispatcher starts or resumes the operation's deterministic workflow. An outbox retry or dispatcher restart must find the same operation rather than produce a second independent stop.
5. In parallel, reconcile and stop owned tasks, quarantine the dedicated IAM task role, and account for controlled tool activity. Keep each connector's status and provenance separately.
6. Verify required outcomes. Deadline expiry leaves quarantine intact and publishes a partial, failed, or unknown result with continued reconciliation.

DynamoDB transactions can make the registry updates atomic within the chosen account and Region. They cannot make a separate ECS call or external tool action atomic with the fence. Strongly consistent base-table reads support current gate observations; GSIs are not the admission authority. [S8, S9]

Launch and tool-admission races

The trusted launcher records a reservation under a condition that the enrolled generation is ARMED before calling RunTask. A reservation ID supplies a stable client token for retries. A concurrent stop may occur after reservation and before ECS responds; that reservation remains part of the stop boundary until its result is reconciled. Late task ARNs must be registered and stopped. An unresolved launch reservation prevents VERIFIED_STOPPED. [S12]

The gateway maps its verified IAM caller to a workload generation. It transactionally records admission under the same generation's ARMED condition before dispatching a supported tool action. This distinguishes pre-fence admitted work from new requests that must be denied. Do not rely on a read-then-act check alone or on a cached positive authorization decision.

The mock adapter can account for its outstanding operations. External jobs, opaque streams, and untracked side effects need their own adapters and are outside this prototype. A request admitted before quarantine may already have caused an effect; neither a database transaction nor task termination reverses that effect.

Runtime and identity enforcement

The runtime connector stops registered exact task ARNs and checks their provider-reported terminal state. The verification record retains enrolled ownership independently of task tags. Task stopping has a signal/timeout lifecycle; a successful API response is not the final observation. [S1]

The identity connector attaches a fixed deny-all policy to the generation's dedicated task role, preserving the role's quarantine until retirement. Do not apply this policy to a shared role or service-linked role. Old-session cutoff revocation alone does not restrict all later sessions, and IAM propagation must be included in observation and retry behavior. [S2, S4]

To test effective denial independently, the sandbox includes a trusted probe task with an operator-controlled image digest. It can be launched with the enrolled generation's task role and attempt a previously allowed read of one harmless diagnostic object. Run it before and after quarantine, including with refreshed task credentials. Store only the result and request metadata, not the credentials. A connector with permission only to inspect a policy cannot claim that it has observed effective access denial.

The probe is separately inventoried as verification infrastructure and cannot execute business tools. Its launcher can pass only the approved diagnostic task/execution roles to the fixed probe definition. Review this permission boundary before deployment.

Network and gateway assumptions

Use private workload subnets without general-purpose NAT or public internet routes for the agent. The proposed tool entry is a private API Gateway REST API with IAM authorization, restricted VPC endpoint/resource policies, and a Lambda adapter. Validate endpoint, image-pull, logging, DNS, and diagnostic-object access explicitly. The task execution role is distinct from the agent task role. [S5, S10, S11]

Private placement alone does not establish isolation; IAM permissions and endpoint policies must also constrain the permitted destinations. The gateway must deny new admissions when it cannot obtain authoritative gate state. This choice favors containment over tool availability during a control-state failure.

Do not use a security-group replacement as the sole proof of an immediate cutoff. Connection tracking can preserve already established flows. The first claim is bounded to enrolled task termination, enforced admission denial, and tested role quarantine. [S6]

Permission boundaries to implement

Principal	Proposed permissions	Explicit boundary
Operator API	Scoped registry transaction and operation reads; configured JWT verification.	No direct general-purpose workload administration.
Dispatcher/workflow	Read outbox, start deterministic workflow, assume exact connector roles.	Cannot select arbitrary customer accounts or roles from request text.
Runtime connector	StopTask/DescribeTasks on enrolled task scope; narrow cluster discovery as required.	Does not stop unknown resources or clear quarantine.
Identity connector	Put the fixed quarantine policy on dedicated generation roles; read back configuration.	No general role creation, policy escalation, shared-role edits, or recovery unquarantine.
Trusted launcher	Run the enrolled definition and pass only approved task/execution roles.	Requires a valid launch reservation and generation fence.

Principal	Proposed permissions	Explicit boundary
Agent task role	Invoke its exact tool API and read the one harmless diagnostic object.	No IAM, RunTask, PassRole, alternate role assumption, control writes, or arbitrary data access.
Gateway adapter	Conditional admission and mock tool ledger operations for its environment.	Cannot approve recovery or reopen an old generation.
Verifier/probe runner	Provider reads, scoped probe launch, bounded diagnostic observations, evidence writes.	Cannot grant business permissions or trust agent-supplied success.
Provisioner/recovery operator	Create the reviewed next-generation resources under a single-use approval.	Keeps prior role and generation quarantined.

This is a design matrix, not deployable IAM JSON. Review resource-level action support, trust policies, PassRole conditions, and each unavoidable wildcard when writing infrastructure code.

Verification and status model

An operation progresses ACCEPTED -> ENFORCING -> VERIFYING. VERIFIED_STOPPED requires every mandatory check to pass for the enrolled generation: launch fence closed; launch reservations settled; registered tasks stopped; owned-task reconciliation complete; new gateway admissions denied; controlled in-flight actions accounted for; and effective role denial observed.

If required observations are unavailable, report UNKNOWN. If known enforcement is mixed with known failures or coverage gaps, report PARTIAL. If enforcement cannot proceed because of a known terminal failure, report FAILED. Results include per-check evidence and timestamps. The generation remains QUARANTINED in every incomplete case.

Verification is an observation over a defined boundary and time. Later contradictory evidence appends a new verification attempt and changes the current assessment; it does not erase historical records or preserve an outdated success banner.

Recovery and delivery gates

A recovery review binds the incident, workload, old generation, and intended next generation. Approval creates a single-use provisioning permit; it does not restart tasks. The provisioner prepares a fresh role, task definition, and enrollment. The new generation's tool gate opens only after its enrollment checks pass. The previous role's deny and historical records remain.

Before creating AWS resources, settle the account/Region, identity integration, actual resource policies, deployment budget, retention, endpoint design, and diagnostic probe. Before a pilot, execute the validation matrix and review all unknown or partial outcomes. Build a deterministic enforcement harness first, then connect a real model behind the same controlled gateway.

04 / TECHNICAL FOUNDATION

API and integration guide

Version 0.1 | Proposed contract, no running product API | October 27, 2026

API boundary

The OpenAPI 3.1 contract in `openapi.json` is an implementation starting point. Its example server uses the reserved `.invalid` domain. It does not describe the existing website contact form or turn the browser simulation into a real control service.

All endpoints require authenticated operator identity and action-specific authorization. The chosen identity integration must validate token issuer, audience, signature, expiry, and configured authentication assurance. A bearer token alone is not proof of MFA or permission to operate every workload. Resolve environment and workload scope from trusted policy, not a caller-supplied tenant header.

Proposed endpoints

Method and path	Meaning
GET /v1/workloads	List workloads visible to the caller, with cursor pagination.
GET /v1/workloads/{workloadId}	Read the current generation, enrollment state, and coverage assessment.
POST /v1/workloads/{workloadId}/stop-requests	Durably quarantine the specified current generation and accept a stop operation.
GET /v1/operations/{operationId}	Read operation status and independent check observations.
POST /v1/workloads/{workloadId}/recovery-reviews	Request a human review of a verified stop. Does not restart execution.
GET /v1/recovery-reviews/{reviewId}	Read the review state and its incident/generation binding.
POST /v1/recovery-reviews/{reviewId}/approvals	Approve the specific next-generation provisioning permit. Does not restore the old generation.

The first product API does not accept arbitrary AWS account IDs, task ARNs, role ARNs, policy documents, or connector URLs in these requests. Enrollment and provisioning are operator-controlled deployment procedures for the sandbox.

Identity and state contracts

Workload IDs, operation IDs, and review IDs are UUIDs. Generation numbers are positive integers. The current workload state is ENROLLING, ARMED, QUARANTINED, RECOVERING, or RETIRED. Keep per-generation quarantine records when the current generation changes.

Stop operations use ACCEPTED, ENFORCING, VERIFYING, VERIFIED_STOPPED, PARTIAL, FAILED, or UNKNOWN. Per-check results are PENDING, PASSED, FAILED, or UNKNOWN; each required check has provenance and an observation timestamp when a result exists. An API success response is not a verification result.

Recovery reviews use REQUESTED, APPROVED, REJECTED, EXPIRED, or CONSUMED. Only the trusted provisioner consumes the approval. The review and next-generation launch must bind to the same workload and incident. A later stop or generation change invalidates an outstanding permit.

Mutation behavior and concurrency

Every POST requires an Idempotency-Key UUID. Scope the durable key to the authenticated environment, action, and target. The same key and semantic payload resolve to the same resource ID; conflicting reuse returns 409. Replays do not create a fresh workflow, review, or permit. An authorized repeat stop for an already quarantined generation returns the existing stop operation rather than silently starting a competing operation.

Retain application idempotency mappings for a proposed seven-day sandbox window. A database or provider token's shorter retention cannot replace this contract. Quarantine records do not expire with an idempotency key.

The stop body includes expectedGeneration and a bounded reason. A generation mismatch returns 409 with a scoped refresh instruction. The response is 202 and a Location pointing to the operation. It means the fence and operation record are durable, not that task stopping is complete. A replay may refer to an already completed operation; GET returns its latest assessment.

Use the registry/outbox transaction described in the architecture. A workflow-start timeout is reconciled by operation identity. Each connector action must handle retries and uncertain provider results without releasing the quarantine latch.

Error and observation behavior

Status	Contract
400	Invalid JSON, extra mutation fields, missing required values, or invalid identifiers.
401	Missing, invalid, expired, or unacceptable authentication.
403	Authenticated caller lacks the requested action or authentication assurance.
404	Resource is absent or not visible in the caller's scope.
409	Generation, review revision, review state, or idempotency conflict.

Status	Contract
429	Rate limit; bounded Retry-After.
503	Acceptance could not be established or a required service is unavailable.

Errors return a stable code, safe message, and requestId. Never include credentials, internal policy bodies, or another customer's identifiers. A client uncertain whether a POST was accepted must retry with the same key and body. It must not infer rejection merely from a lost response.

An unavailable observation endpoint must not leave the operator looking at an apparently current verified result. Show the observation time, current coverage, and a connection/staleness notice. An operation can be incomplete even while some controls succeeded.

Enrollment prerequisites

Validate the designated accounts, Region, cluster, exact launch definition, per-generation identity, launcher exclusivity, network paths, gateway mapping, and diagnostic probe. Record root/child ownership and required check types. Treat evidence-store and control-identity protections as prerequisites, not agent-editable runtime settings.

The sandbox task role includes only the enrolled tool API and one harmless diagnostic-object read. A trusted probe using the same generation role tests effective identity denial independently of the agent. Keep provider keys, privileged credentials, and probe credentials out of agent outputs and evidence documents.

Operator implementation notes

Require one deliberate, scope-specific human stop action. The dashboard must display the selected workload and generation and then track individual outcomes rather than animate a guaranteed success sequence. The existing website demo remains illustrative until a separate authenticated implementation exists.

Recovery approval and runtime restoration are different actions. Use the runbook to consume a single-use approval and prepare a new generation. API, audit, identity, and connector behavior must pass the validation matrix before the UI can make a verified-stop claim.

05 / TECHNICAL FOUNDATION

Validation plan and operator runbooks

Version 0.1 | Proposed sandbox procedure | October 27, 2026

Test boundary and evidence

Use an isolated AWS sandbox with the enrolled Fargate workload, its registered child tasks, a controlled launcher, mock tools, and a harmless diagnostic object. Keep an unrelated sentinel workload running to detect unintended impact. Run no drill against an unenrolled customer workload.

Start with a deterministic process that repeatedly attempts supported actions and can ignore a cooperative stop. This tests enforcement independently of a model's willingness to comply. After the harness passes, add a real agent behind the same gateway and repeat the boundary tests.

For every drill, record the build and connector versions, approved scope, task/role generation, operator identity, request/operation IDs, timestamps, provider request IDs, per-check observations, expected result, actual result, and remaining uncertainty. Do not retain task credentials, provider secrets, or unnecessary prompt data.

Drill matrix

ID	Scenario	Required result
D-01	Process ignores the cooperative stop and stop signal.	Infrastructure termination is observed; agent self-report is not accepted as proof.
D-02	Repeated launch attempts after quarantine.	Trusted launcher denies admission; no automatic reopening occurs.
D-03	Child launch races the stop or its RunTask response is delayed.	Reservation remains visible; late tasks are stopped; unresolved ownership prevents VERIFIED_STOPPED.
D-04	Old and refreshed generation credentials attempt an allowed diagnostic read.	Before/after observations demonstrate quarantine; a new session does not restore the old generation's authority.
D-05	Agent attempts an alternate tool, endpoint, direct egress route, or IAM escalation.	Unsupported paths are denied or the enrollment/coverage assessment explicitly fails.
D-06	Caller changes workload, generation, or injects AWS ARNs/policy fields.	No action outside authorized scope; stale generation and extra fields rejected.
D-07	Expired token, missing assurance, wrong role, or scoped operator revocation.	Authentication/authorization rejection; no workflow or information leak.

ID	Scenario	Required result
D-08	Agent forges status, changes local logs, or task tags become unavailable.	Provider/registry evidence governs the result; unrelated sentinel remains unaffected.
D-09	Connector permission removed, provider throttled, or observations unavailable.	Quarantine persists; bounded retries and honest PARTIAL/FAILED/UNKNOWN reporting.
D-10	Duplicate POST, key conflict, lost response, or dispatcher restart.	Same semantic request resolves to one operation; conflicting payload is rejected; outbox resumes safely.
D-11	Recovery request, approval replay, fresh generation, and old-role probe.	Approval alone does not restart; one permit is consumed; new identity is enrolled; old identity stays denied.
D-12	Controlled tool request admitted before the fence completes late.	In-flight work remains accounted for; completed effects are disclosed, not claimed reversed.

Run race and retry cases repeatedly with controlled timing variations and publish sample counts. Measure acceptance, individual enforcement, and verification times separately. The proposed 120-second verification budget is a test deadline, not a performance guarantee; exceeding it never releases quarantine.

Stop runbook

1. Confirm the workload, generation, authorized scope, and observed coverage. A coverage gap is a reason for a visible incomplete result, not permission to stop unrelated resources.
2. Send the stop request with a stable idempotency key. If the response is lost, repeat the same request/key and inspect its operation.
3. Confirm the durable quarantine latch and dispatch status. Check launch reservations, task stopping, gateway admissions, in-flight work, and role quarantine separately.
4. Inspect independent verification. Keep the operation incomplete if any required check is unknown or known to have failed.
5. Preserve available incident records and keep quarantine. Investigate late tasks or unexpected activity; append a new verification attempt when evidence changes.

Incomplete-operation runbook

For PARTIAL, FAILED, or UNKNOWN, display which path is affected and the last observation time. Retry only enrolled actions, reconcile uncertain provider responses, and escalate to an authorized operator. Do not clear the latch to make the dashboard green. Do not treat a heartbeat failure, missing task tag, or expired verification budget as evidence that the agent stopped.

If the application control path is unavailable, an authorized AWS operator follows an independently governed emergency procedure for the designated cluster, role, launcher, and gateway. Record that action when the audit path returns. This procedure still depends on AWS management availability and cannot establish a universal out-of-band shutdown guarantee.

Recovery runbook

1. Obtain fresh verification for the stopped generation and review cause, scope, available evidence, and outstanding external effects.
2. Create a recovery review tied to the incident and generation. An authorized human approves the proposed next generation under the configured policy.
3. The trusted provisioner consumes the single-use permit, prepares a fresh task role and deployment definition, and leaves the old role quarantined.
4. Run enrollment checks for the new generation before enabling its supported tool and launch admissions. Verify that old credentials remain denied.
5. Record the human approval, provisioning outcome, new generation, and any exceptions. A failed recovery remains closed; approval is not automatic proof of a healthy deployment.

Release gates

The architecture gate requires review of actual IAM/trust policies, identity assurance, endpoints, launcher/gateway race handling, diagnostic-probe permissions, retention, and budget. The prototype gate requires reproducible drill results and an unaffected unrelated sentinel. The pilot gate requires an agreed scope, operating owner, unresolved-risk record, recovery rehearsal, and observed behavior matching public claims.

Any false VERIFIED_STOPPED result blocks a pilot. Remaining deployment gaps must be explicit. There are no measured product benchmarks or completed pilot results in this draft.

SOURCES / VERIFIED OCTOBER 27, 2026

Official AWS references

These sources support the platform behavior discussed in the draft. The proposed product architecture and requirements remain design decisions, not AWS product claims.

[S1] ECS StopTask

[S2] IAM role-session revocation

[S3] ECS services and task replacement

[S4] IAM consistency guidance

[S5] ECS task IAM roles

[S6] Security-group connection tracking

[S7] Step Functions workflow types

[S8] DynamoDB read consistency

[S9] DynamoDB transactions

[S10] API Gateway private REST APIs

[S11] ECS VPC endpoints

[S12] ECS RunTask and consistency