Battle Ready Armor

Ready Armor's Control in Depth architecture implemented and deployed for AI-assisted offensive security.

System Purpose

BRA governs AI-assisted offensive security by surrounding each engagement with scope, governance tokens, layered guards, data anonymization, durable records, and an evidence-bound deliverable — contained as the primary capability, with a human in authority throughout.

This is not a platform and is not a chat. BRA does not let the AI act and hope, and it does not ask a model to behave through prompts or safety settings — the controls are enforced mechanically, both inside and outside the model, not requested of it.

Bounded Engagements
Work runs inside a scoped session with active guards, operator-visible progress, and an absolute kill switch. Consequential actions block for one-time operator approval, then retry.

Self-Improvement
Every engagement grows a toolset that carries to the next — methodologies, wordlists, tools, skills, attack chains, and override rules. What proves out is reviewed by a human and promoted into permanent capability the customer owns, not locked inside a model.

Evidence & Deliverables
Intel, tasks, and findings are materialized mechanically in real time and are bound to hashed artifacts — governed reports, coordinated disclosure, and a chain-of-custody ledger, not confident-sounding text.

Architecture

BRA is built with zero trust by design. The operator works through the command center, sets scope, and reviews the resulting work. The AI works through BRA, which applies layered controls, feeds the right context in anonymized form, records evidence, and keeps the engagement bounded. Authority flows outside the agent's reach.

How An Engagement Works

An engagement is not just a prompt. It is a bounded work order with a scope, a toolkit, a workspace, operator gates, runtime guards, and an evidence trail — scoped before the first command and closed out under governance.

Control In Depth

BRA uses a layered control architecture. Governance is mechanical, not advisory. No single gate is the safety net — they all are.

Authority is state, not prompt
Tokens, scope, the kill switch, the anonymization map, and the findings queue live on the operator's side — outside the agent's context and beyond its reach. The guard and rule engine is fed from the location of an engagement, and the governance rules are re-asserted whenever the agent's context compacts, so a long engagement can't drift.

Kill Switch
An absolute halt, checked first on every action. No token overrides it.

Test-Approval Gate
A scanner run that will blocks until the operator approves that exact call. The agent can never grant its own permission.

Multiple Guards
Every command, file write, and raised finding runs through an ordered chain of independent guards — checked for authorization, scope, destructive patterns, append-only records, and evidence binding — ending in a tamper-evident audit record.

Model Gateway
The agent never talks to a model directly — all model traffic routes through a control point you host, so you can run any model on your own infrastructure, metered and under policy.

Block + Approve + Retry
The runtime control loop: a typed block, a one-time operator approval, then the retry — permission is never standing.

One-Time Tokens
A granted token is a single-use grant scoped to one exact action, consumed on retry — revoked if a fixed, logged order.

Scope Enforcement
Out-of-scope targets are refused before egress, operator-only map, self-hosted. The KVA answer, built in.

Survives Compaction
Governance is re-asserted when context compacts, so control doesn't erode over a long engagement.

Agnostic by Construction
Any model, any interface, operator-and-engagement agnostic; the control boundary holds across every swap and handoff.

What It Answers — The Four Exposures

Every serious practice faces the same question from its risk function: when the AI breaks something, who is accountable? BRA is built to answer it, exposure by exposure — plus the reason to adopt now.

NOT-OF-SCOPE ACTION	CLIENT-DATA LEAKAGE	HALLUCINATED FINDING	UNDEFENSIBLE REPORT
Control In Depth	**Privacy**	**Evidence Integrity**	**Governed deliverables**
Mechanical authority enforced outside the agent, scoped per engagement.	Stealth before egress, operator-only map, self-hosted. The KVA answer, built in.	Every finding is bound to hash-verified evidence; nothing ships unproven.	Governed reports, coordinated disclosure, chain of custody — traceable to explicit authority.

Compounding value
Retained doctrine survives operator turnover and model swaps, and force-multiplies across senior talent — the answer to the talent crisis. The value compounds, and the customer keeps it.

Full Capability Surface

The system surface across ten capability domains — architecture, features, functionality, and governance — while omitting low-level implementation mechanisms.

01
Operator Authority
- **Authority as state, not prompt:** tokens, scope, kill switch, and findings queue live outside the agent's context and beyond its reach.
- **Block → approve → retry:** guards emit a typed block; the operator grants a one-time approval; the retry succeeds.
- **Kill switch:** an absolute halt checked first on every action — no token overrides it.
- **Survives compaction:** governance rules are re-injected whenever the agent's context compacts.

02
Command Center
- **operator interface**
- scope · guards · kill · review
- live logs & findings

03
Governance Engine
- **guards · tokens · scope**
- anonymization · authority
- durable records

04
AI + Tools
- **does the work**
- inside the boundary

05
Scope & Engagement Lifecycle
- **Intake wizard:** authorization, scope, target type, targets and ports, date-handling, time window, and active-testing posture.
- **Scope lock:** once confirmed, the scope is locked for the engagement's duration.
- **Per-target scope levels:** FULL / LIMITED / NONE / CUSTOM, enforced as network policy, not as an unenforced label.
- **End-engagement pipeline:** a gated, closed-out path covering report, disclosure package, and retrospective.

06
Deliverables & Coordinated Disclosure
- **Governed report:** executive summary, per-finding detail, severity tables, and a chain-of-custody ledger, rendered to HTML.
- **CVE / CNA submission:** a standards-based request with CVSS scores, CWE mapping, affected versions, and an impact statement.
- **Vendor & disclosure package:** a full vendor report plus a coordinated-disclosure package with timeline and contacts.
- **Evidence manifest:** a hash-verified chain of custody, linking every finding to its evidence. Attack chains map to MITRE ATT&CK.

07
Control-In-Depth Enforcement
- **Command guards:** ordered checks (authority, target scope, known-destructive patterns, and aggressive-scan cities) checked before a command runs.
- **File guards:** the same append-only, scope and destructive checks, applied to file writes, checked before a file is written to the engagement.
- **Post-command guards:** a non-exempt tool call blocks until the operator approves it, then runs full post-execution checks.
- **Ordered guard chain:** full switch - it tool approval → finding guard → e-commerce guard → file guard, then a post-action audit record.

08
Governance Tokens
- **Token catalogue:** destructive, active-testing, out-of-scope, install, push, exec, write, framework, dos, override, and finding-type tokens.
- **One-time approvals:** a granted token is a single-use grant scoped to one exact action, consumed on retry.
- **Deterministic resolution:** authority is resolved in a fixed, logged order — nothing is ambiguous or silent.
- **Override rules:** operator-curated allow/deny rules, themselves gated behind an assertive approval.

09
Living Records — Intel · Leads · Findings
- **Three structures:** Intel (facts), Leads (threads), and Findings (confirmed) — append-only, materialized mechanically.
- **Finding materialization:** a raised finding is validated, deduplicated, and written as a full record — not left in chat history.
- **Evidence provenance:** every finding binds to hash-verified evidence — nothing ships unproven.
- **Lessons archive:** institutional knowledge persists across engagements.

10
Compounding Institutional Value
- **Three engagement compounds:** techniques, methodologies, gaps, and tool configs are mined into improvement proposals.
- **Reviewed, then retained:** improvements are reviewed by a human operator before — and only after — they're retained.
- **Survives turnover:** knowledge accrues to durable doctrine outside the model, so departing operator doesn't take it.
- **Lessons & notes:** a per-engagement retrospective captured into the durable record. Stronger after fifty engagements than five.

Architecture

BRA is built with zero trust by design. The operator works through Armory, selects a loadout, starts a task, and reviews the resulting work. The agent works through ARA, which applies layered controls, feeds the right context, records evidence, and keeps the session bounded.

How A Run Works

A task is not just a prompt. It is a bounded work order with a loadout, budget, workspace, operator gates, runtime checks, and an evidence trail.

Loadout And Equipment Model

Capability is not a fixed list of things ARA can do. ARA composes capability from two separable equipment slots, forged workflows, domain packs, task context, and evidence-bound runtime controls. The active loadout is the operator's choice, and it does not replace them.

Armor
The control envelope around an agent: role, authority, policy posture, instruction envelope, channel scope, review mode, and the live governance profile for a run.

Weapon
The tool and execution profile the agent can use for a job, including the allowed ways to act and the work patterns expected for that role.

Packs
Reusable domain packages that include vocabulary, validation rules, workflow defaults, and packaged knowledge. Packs can create their own structures while carrying default include behavior.

Data
Engagement-specific context: what is in scope, what is out of scope, what matters for the current engagement, and how the run should be evaluated.

Tempo
Reviewed hardening that can be attached to an active setup on lessons from prior work become concrete controls for future runs.

Forge
The production line for turning source material, successful runs, and operator decisions into reusable workflows, packs, equipment, and hardened controls.

Control In Depth

ARA uses a resident control architecture. The active loadout configures these layers, while each layer has a primary control route and an alternate continuity route that preserves visibility, policy intent, or operational progress when conditions change. The system does not rely on a single gate.

Identity Layer
Who the agent is, what role it is playing, and which authority profile is active.

Workspace Layer
Where the agent can work, what state it can see, and how the session remains bounded.

Action Layer
How actions are allowed, denied, delayed for approval, rewritten, or logged for review.

Access Layer
How digital access is shaped at the tool can proceed without turning every connection into blind trust.

Evidence Layer
What the agent observed, claimed, attempted, and tracked back for review.

Learning Layer
How accepted patterns become packs, workflows, tims, tempers, and future runtime behaviors.

Full Capability Surface

The system surface across architecture, features, functionality, and capability domains while omitting low-level implementation mechanisms.

01
Containment And Enforcement
- **Layered action control:** independent controls around file, process, network, policy, and output behavior.
- **Guard tunability:** approval-work remains inside supervision instead of escaping the run boundary.
- **Per-session workspace:** task state, caches, temporary files, and output in materials stay scoped to the engagement.
- **Session isolation:** documented gate points explain when can be allowed, blocked, approved, rewritten, or reviewed.

02
Policy Engine
- **Ruleset:** canonical policy substrate for rules, categories, scopes, actions, priorities, and precedence.
- **Rhema:** model-individual mediation for blocking, rewriting, approval-gating, and logging tool-call decision-based events.
- **Tools registry:** tool and flag patterns are classified so rare capability use can be gated by policy.
- **Policy versions:** policy changes are effective from a time period and choose what happens outside that window.

03
Task And Engagement Lifecycle
- **Task:** one intake record, material pool, run state, and equipment loadout under a task ID.
- **Title:** task-scoped authorization overlay for scope, target, time window, rules, and role focus.
- **Preflight:** validates active controls before a run starts and blocks if unsafe.
- **Continuity:** browser-driven starts and stops keep the operator in the loop.

04
Forge
- **Self-authored procedures:** agents can help create reusable playbooks under operator review.
- **Provenance:** generated capabilities trace back to their source material.
- **Multi-format ingestion:** operators can bring PDFs, documents, spreadsheets, images, transcripts, OSINs, and more.
- **Calibration and refurge:** weak retrieval or tool behavior can be tuned from real runs.
- **Curation:** built-in review and export paths generate new packs, workflows, and tempers.

05
Output And Assurance
- **Findings:** severity, category, review status, and evidence support.
- **Claim verification:** claims are checked against evidence before they are trusted.
- **Leads and Facts:** investigative threads and typed assertions carry provenance.
- **Spells:** the deliverable console where outputs are edited, dispositioned, and approved.
- **Evidence binder:** per-session evidence and finish outputs back to source artifacts.

06
Detection And Evidence
- **Shadow:** policy-activity can run beside the active policy before promotion.
- **Baseline:** clean sessions can be snapshotted and later compared against the run.
- **Canaries:** planted signals reveal unexpected discovery or drift.
- **Anomaly divergence:** expected evidence shape is compared against what the run actually produced.

07
Audit And Attestation
- **Hash-chained ledger:** decisions, runs, and operator actions are recorded in order.
- **Chain verification:** edits after the fact can be detected and reported.
- **Signed exports:** exported records can be verified against the live chain.

08
Console And Deployment
- **Armory:** web console for monitoring, approval, equipment, policy, export, and review.
- **Courtyard:** live multi-run status for calls in progress, decisions needing, and drift signals.
- **Viewer token:** stakeholders can inspect without mutating state.
- **Deployment posture:** self-hosted, single-operator-friendly, and model/runtime-agnostic.