TIER-2 PILOT · ONE PER CUSTOMER

Shadow-mode pilot — execution checklist

Tier 1 (everharden.com/gate) is the conviction demo against a *mock* agent — it earns the meeting. Tier 2 (this sheet) is the gate wired in front of the customer's *real* agent in shadow mode: it blocks nothing, logs everything it *would* have stopped, and the deliverable is the customer's own evidence report. Do not promise a guarantee until Stage 1 returns zero ungated paths.

CUSTOMER

_____

OWNER (EVERHARDEN)

_____

AGENT PLATFORM / RUNTIME

_____

CHAMPION (CUSTOMER SIDE)

_____

DATE STARTED

_____

TARGET SHADOW WINDOW

_____ days (recommend 10–14)

STAGE 0

Qualify — before you spend a day on it

- [ ] The agent takes at least one consequential action (moves money, changes a record of record, sends data out). *No consequential action → no gate → politely disqualify.*
- [ ] The customer can name one catastrophic action they fear, and it's dollar- or data-bounded, not a vibe.

  Write it here:

  _____
- [ ] There is a real interception point — an MCP server, a tool-dispatch layer, or SDK middleware — where every tool call passes through. *If tool calls happen in scattered places with no common chokepoint, flag it: completeness will be the whole project.*
- [ ] The customer will give a read-only / staging view of the agent's tool list and traffic for shadow mode.

  Decision: **PROCEED / DISQUALIFY** — reason:

  _____

STAGE 1

Phase 0 — prove the chokepoint can be complete

> Fill `PHASE0_WORKSHEET.md` against the customer's **actual** agent. A small tool list is a snapshot — the day they add a tool, completeness silently breaks, which is exactly what `CHOKEPOINT.md` + the CI drift-check exist to catch.

- [ ] Enumerate **EVERY** tool/capability — not just the obvious ones (PII reads, CRM writes, transactional emails, plan changes, store credit, internal admin APIs, saved reports).
- [ ] For each tool: reaches money? reaches data out? irreversible? → normalized typed action → gated?
- [ ] Run the **side-door checklist** (Worksheet §B). Every box must pass:
  - [ ] No ungated execution primitive (shell, exec, code-run, raw SQL, SSH). If present → STOP, that's the conversation.
  - [ ] No read-that-becomes-a-write left unclassified.
  - [ ] No session-trust (authority is per action+params, never once-per-session).
  - [ ] No third-party / MCP server with its own creds the gate doesn't front.
  - [ ] The catastrophic action reduces to a **typed fact** (amount / change-type / external flag).
- [ ] Fast first pass: paste the tool list into the checker at **everharden.com/gate/#checker** before the deep manual audit.
- [ ] **Sign-off** (Worksheet §D): chokepoint complete? YES / NO. If NO → list open path(s) + the customer change to close them. No enforce until closed.
- [ ] Record every gated tool in `CHOKEPOINT.md`. CI: `python run.py check-chokepoint` must pass.

STAGE 2

Wire the interceptor — shadow mode

> Replace the demo's `adapters/mock_support_agent.py` with a real adapter at the interception point. Shadow is the default (`Interceptor(mode="shadow")`): log the decision it *would* make, let the action proceed.

- [ ] Pick the mechanism: MCP proxy / SDK middleware / tool-dispatch wrapper (circle one).
- [ ] Write per-tool normalizers (raw params → typed `Action`). Intent/words are stripped here — the gate only sees facts.
- [ ] Register every consequential tool; reads pass through as benign. Unregistered consequential tool = `BypassError` by design (no side doors).
- [ ] Inject the customer's real limits/policy (refund ceiling, velocity window, payout-confirm) — not the v1 module defaults.
- [ ] Confirm no LLM / no network in the enforcement path. CI: `python run.py test` (incl. `test_no_inference_in_engine`) green.
- [ ] Point the evidence log at durable storage; it's hash-chained — verify the chain loads.
- [ ] Staging smoke test: replay 2–3 known-bad actions → logged as `WOULD BLOCK/HOLD` and still proceed (shadow).
- [ ] Blast-radius check: confirm shadow cannot stop or delay a single real action. The customer must believe this before it touches production.

STAGE 3

Run the shadow window — 10–14 days

- [ ] Gate live in shadow on real traffic. Blocks nothing.
- [ ] Day 1–2: watch for false WOULD BLOCK/HOLD on legitimate actions → tune limits, re-record in `CHOKEPOINT.md`. Tuning during shadow costs nothing.
- [ ] Watch for new/unseen tools in the log (drift). Each one re-opens Stage 1 for that tool.
- [ ] Mid-window check-in: show the champion the running count of would-have-stopped actions.

- [ ] Verify the hash chain still validates at window end.

STAGE 4

Deliver the evidence report

> The report answers one question: "here is the catastrophic thing we would have stopped this week." Generated from the evidence log via `gate/report.py → render_html` (self-contained, screen-shareable).

- [ ] Generate the HTML report from the customer's real log.
- [ ] Headline: actions observed · would BLOCK · would HOLD.
- [ ] Top dangerous actions: the typed facts, the rule, the plain-English why.
- [ ] Translate to dollars / exposure using the Stage 0 catastrophic-action figure.
- [ ] Walkthrough: a fooled and a fully-jailbroken agent hit the same wall — the gate never trusted the agent.

STAGE 5

Enforce decision — go / no-go

- [ ] Customer reviews the report and decides: flip to enforce / extend shadow / walk away.
- [ ] If enforce: switch `mode="enforce"`, agree the rollout (BLOCK-only first, then HOLD→human-review queue), define who handles held actions.
- [ ] If extend: tune and re-run — no enforce until the report convinces them.

Commercials:

- [ ] Set CHOKEPOINT.md + CI ownership so coverage can't rot after handoff (the moat, not the rules).

Non-negotiables — true at every stage

No model in the enforcement path. Detection/threat-intel is a feed for the log only, never load-bearing.

Shadow is the default. Nothing is blocked until the customer has seen what would be blocked.

Laundering tests are the spec. Every rule must catch the sideways variant (edited payee, split refund, fake approval), not just the naive attack.

Completeness is the product. A documented chokepoint + CI drift-check is the guarantee. Can't prove the set is complete? You have a scanner, not a gate.

SIGN-OFF

Per-customer

STAGE	DONE	DATE	NOTE
0 · Qualify	☐		
1 · Phase 0 complete (YES)	☐		
2 · Wired in shadow	☐		
3 · Shadow window run	☐		
4 · Evidence report delivered	☐		
5 · Enforce decision	☐		

Outcome: ☐ Enforce (won) ☐ Extended shadow ☐ Disqualified — reason:

_____