

Two ways to ship an app with Claude. Only one is production-ready.

```
$ compile --target=production --check=rbac,hallucination,injection
```

THE CASE

Claude writes excellent frontends — and, increasingly, backend code too. But **“Claude wrote it”** and **“a runtime enforces what it's allowed to do”** are different claims. A Magic Cloudlet closes that gap: it compiles a plain-language description into a validated backend where invalid or unauthorized instructions can't run, whether Claude built the frontend alone or the whole stack free-form.

Cost

Reference backend build: **~80% fewer AI tokens** on a cloudlet than hand-built (~25K vs ~140K) — auth & hosting collapse into the runtime.

Quality

Hallucinated or invalid logic fails **before** it runs — a named, pre-execution error, not a production incident.

Security

RBAC is enforced by the **runtime itself**, per function, per role — not by application code a model happened to write correctly.

TRUST BOUNDARY

Claude alone: model → free-form code → your CI → production. *Boundary = the prompt.*

Magic Cloudlet: model → execution tree → per-role whitelist → production. *Boundary = the runtime.*

LEGEND

HANDLED

PARTIAL

MANUAL

THE PRODUCTION-READINESS CHECKLIST — WHO OWNS EACH STEP

REQUIREMENT		CLAUDE ALONE — FRONTEND ONLY, OR FULL STACK, FREE-FORM	MAGIC CLOUDLET + CLAUDE
1	Database schema	MANUAL Designed by hand, from scratch.	HANDLED Generated, or wraps your existing DB — zero migration.
2	Hosting & infra	MANUAL Containers, secrets, CI/CD — a DevOps project.	HANDLED Managed cloudlet, or self-host / on-prem / air-gapped.
3	API / route layer	MANUAL Backend sprint — endpoints, contracts, versioning.	HANDLED Generated per table/module, with an OpenAPI spec.
4	Authentication	MANUAL SSO / OAuth / SAML, built and maintained.	HANDLED Runtime-native tokens; SSO scoped as needed.
5	Authorization / RBAC	MANUAL Written and maintained in app code by hand.	HANDLED Enforced by the runtime, per function, per role.
6	Injection defense	MANUAL Full audit — free-form code accepts any call.	HANDLED Closed vocabulary — unauthorized calls can't be expressed.
7	Rate limiting / abuse	MANUAL Designed and implemented separately.	HANDLED Capability grants checked at every dispatch.
8	Logging / audit trail	MANUAL Designed, instrumented, wired to a log store.	HANDLED Built-in, filterable, severity-tagged change log.
9	Encryption at rest/ transit	MANUAL Configured as part of the infra build.	HANDLED Runtime-managed; air-gap-ready for sensitive data.
10	Security review	MANUAL Full human review of every generated line.	PARTIAL Structural checks handle validity; logic still reviewed.
11	Compliance mapping	MANUAL Ongoing, specific to your framework.	PARTIAL Self-host/air-gap keeps infra in your boundary.
12	Deploy pipeline & rollback	MANUAL Built once a backend and infra exist.	HANDLED Included with a managed cloudlet.
13	Ongoing patching	MANUAL Your team, indefinitely.	PARTIAL Maintained by Nuiity (managed), or self-run under MIT.

13/13 manual when Claude builds it alone — either the frontend only, or the full stack free-form. 9/13 handled and 4/13 reduced to partial when a Magic Cloudlet builds the backend alongside Claude.

Five security layers, side by side

LAYER	CLAUDE, FREE-FORM BACKEND	MAGIC CLOUDLET + CLAUDE
Instruction vocabulary	Open — any identifier the host language can resolve.	Closed — finite, registered capability set.
Pre-execution check	A human reads every generated line before it ships.	Static proof vs. the live capability registry.
Capability grants	Not native — hand-built and hand-maintained.	Argument-bound, checked at every dispatch.
Identity & RBAC	Middleware each new build may re-implement.	Runtime object — one role, token to admin screen.
Sandboxing & audit	Whatever your team designs and maintains.	Function-level sandbox + severity-tagged log.

WHERE WE'RE HONEST ABOUT TRADE-OFFS

Claude alone is faster to a disposable prototype and needs no added platform relationship. If your build is a throwaway demo nobody will ever put real customer data through, that may be the right call. What doesn't change: a prompt is negotiable, a runtime boundary isn't — and free-form code needs a human review pass regardless of how good the model that wrote it is.

What this actually costs, both ways

	CLAUDE ALONE + YOUR TEAM	MAGIC CLOUDLET + CLAUDE
Platform cost	No fee — cost shows up in eng. time.	Self-host free (MIT). Managed: flat \$100/mo.
Adding people	Backend engineer(s), indefinitely.	+\$100/mo user · +\$200/mo developer.
End users	N/A — your own build.	Unlimited — never metered, never billed.
Time to secured backend	Days–weeks of review, after the demo ships.	Minutes to a validated, RBAC-locked API.

~140K

tokens: backend built & secured from scratch (ref. build)

~25K

tokens: same backend, generated & validated on a cloudlet

~80%

fewer tokens on the backend/integration slice of the work

FOR THE APPROVAL MEMO

\$100/month buys a hosted, secured, RBAC-enforced backend runtime with one developer seat — less than most orgs spend on a single hour reviewing free-form generated code before it touches production data.

Twenty minutes. Bring your real use case.

Nuity AI's engineers will walk your team's own backend or internal-tool backlog against this comparison — live, no sales deck.

SELF-HOSTED

Free

MANAGED CLOUDLET

From \$100/mo

MANAGED SERVICES

From \$3,000/mo

Nuity AI

Dallas–Fort Worth, TX
clayton.redmon@nuity.ai · nuity.ai