

6-STEP MIGRATION FRAMEWORK

# ICP MIGRATION GAME PLAN

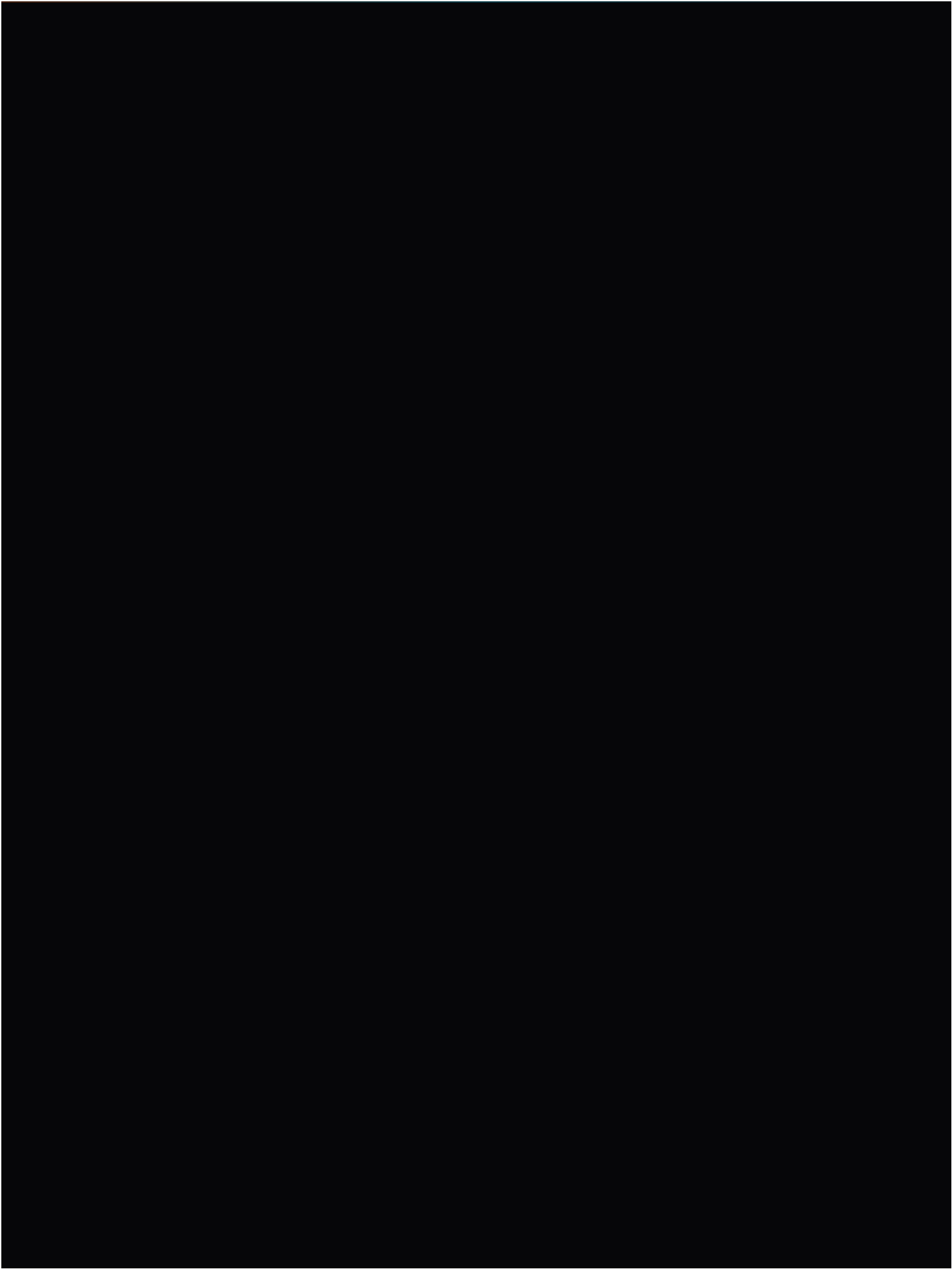
FREE SUMMARY EDITION

---

AUTHOR: DR. RANDY C. SMITH, MBA, WITH THE ASSISTANCE OF AI RESEARCH TOOLS,  
CLAUDE AI AND CAFFEINE AI

THE DEFINITIVE GUIDE TO LEAVING LEGACY INFRASTRUCTURE BEHIND

**AISECRET.AGENCY**



# THE 6-STEP MIGRATION FRAMEWORK

// YOUR COMPLETE PATH FROM LEGACY HOSTING TO THE INTERNET COMPUTER

**01**

## AUDIT YOUR LEGACY STACK

Map every service, API, and data store before you move a single byte.

**02**

## REDESIGN FOR DECENTRALIZATION

Unlearn centralized thinking. Canisters are server, DB, and API simultaneously.

**03**

## MIGRATE YOUR DATA

Start read-only, use stable memory, plan for orthogonal persistence.

**04**

## DEPLOY YOUR CANISTERS

From local replica to IC mainnet with a fully automated CI/CD pipeline.

**05**

## ADD AI WITH CAFFEINE

Intelligence on the chain — zero external AI bills, data never leaves.

**06**

## GO LIVE & MONITOR

No servers to patch. No 3am wake-up calls. The canister just runs.

// LEGACY DISCOVERY

# 01 AUDIT YOUR LEGACY STACK

*"We thought we understood our system... until we mapped it. Three services were talking to each other in ways nobody remembered why. The audit saved us six months of rework."*

— Migration engineer, Fortune 500 retail

## KEY ACTIONS

- ▶ Map every service, API endpoint, and data store in your current architecture
- ▶ Classify data by sensitivity — public / private / sensitive — before touching a single byte
- ▶ Identify real-time requirements: WebSockets, streaming, live feeds
- ▶ Flag vendor lock-in risks — proprietary databases, cloud-specific managed services
- ▶ Document all authentication flows: OAuth, SSO, session management

## COMMON PROBLEMS & SOLUTIONS

| COMMON PROBLEM                           | ICP SOLUTION                                                                             |
|------------------------------------------|------------------------------------------------------------------------------------------|
| <i>Legacy code with no documentation</i> | Motoko's type system forces explicit interfaces — your contracts become self-documenting |
| <i>Mixed authentication systems</i>      | Internet Identity unifies all auth — one identity, all services, zero vendor dependency  |
| <i>Undocumented API contracts</i>        | Candid IDL auto-generates from Motoko types — documentation that can never go stale      |

// ARCHITECTURE RETHINK

## 02 REDESIGN FOR DECENTRALIZATION

*“The hardest part wasn’t the code — it was unlearning centralized thinking. Once we understood that canisters ARE the server, database, and API layer simultaneously, everything clicked.”*

— Lead architect, DeFi protocol

### KEY ACTIONS

- ▶ Design canister boundaries around business domains — not technical layers
- ▶ Plan for upgrade paths using `stable memory` from day one
- ▶ Design message-passing patterns: async/await vs. one-way fire-and-forget
- ▶ Map HTTP gateway patterns for web2 compatibility and SEO
- ▶ Create a data ownership model — who controls what, and what can never be taken away

### KEY REFERENCES

Motoko Playground

IC Developer Docs

Caffeine AI Platform

// DATA MIGRATION

## 03 MIGRATE YOUR DATA

*“We moved 8 years of customer data on-chain. The query speed surprised everyone — including us.”*

— CTO, SaaS platform

### KEY ACTIONS

- ▶ Start with read-only data — products, content, configs — before touching transactional data
- ▶ Use `stable var` for persistence across canister upgrades
- ▶ Implement pagination for large datasets — canisters have memory limits to plan around
- ▶ Embrace orthogonal persistence: no manual save/load cycles, state is always durable
- ▶ Test every upgrade scenario in your local replica before touching mainnet production

### COMMON PROBLEMS & SOLUTIONS

| CHALLENGE                                  | APPROACH                                                                           |
|--------------------------------------------|------------------------------------------------------------------------------------|
| <i>Large relational datasets</i>           | Normalize into separate canisters per domain — one canister per bounded context    |
| <i>Binary assets (images, video, docs)</i> | Use Caffeine object storage extension — serves directly from the canister via HTTP |
| <i>Real-time data requirements</i>         | Polling or certified query patterns replace WebSocket dependencies                 |

// DEPLOYMENT PIPELINE

## 04 DEPLOY YOUR CANISTERS

*“First deployment took 4 hours. Second took 20 minutes. By the fifth, we had a fully automated CI/CD pipeline running on-chain.”*

— DevOps lead, fintech startup

### KEY ACTIONS

- ▶ Configure `dfx.json` with all canister definitions before a single line of code is deployed
- ▶ Set up your local replica for iterative testing: `dfx start`
- ▶ Deploy to IC mainnet when ready: `dfx deploy --network ic`
- ▶ Configure custom domains via the Caffeine dashboard — no DNS wizardry required
- ▶ Set up cycle wallets and monitoring alerts before going public

### DEPLOYMENT CHECKLIST

| ITEM                                   | STATUS SIGNAL                                                            |
|----------------------------------------|--------------------------------------------------------------------------|
| <i>Local replica passing all tests</i> | All <code>dfx canister call</code> responses match expected Candid types |
| <i>Cycle wallet funded</i>             | Minimum 2T cycles per canister recommended for initial deployment        |
| <i>Custom domain verified</i>          | DNS TXT record confirmed; HTTPS certificate auto-provisioned by IC       |

// INTELLIGENCE ON THE CHAIN

## 05 ADD AI WITH CAFFEINE

*“Our OpenAI bill was \$4,200/month. Last month it was zero. And our AI knows things OpenAI never could — because the data never has to leave.”*

— Sarah, community call

### KEY ACTIONS

- ▶ Add Caffeine canister ID to `dfx.json` ; import Candid interface `caffeine_ai.did`
- ▶ Store prompt templates in a dedicated, upgradeable Motoko canister — swap them without redeploying your app
- ▶ Inter-canister inference: `await CaffeineAI.infer({model; input; max_tokens})`
- ▶ Build a prompt caching layer (HashMap on prompt hash) — cuts cycle costs 40–60% for repeated queries
- ▶ Inject user-scoped context from your data canister before inference — personalized AI with zero data leakage

### COST COMPARISON

| SCENARIO                         | TRADITIONAL (OPENAI API) | ON-CHAIN (CAFFEINE AI)              |
|----------------------------------|--------------------------|-------------------------------------|
| <i>Monthly AI inference bill</i> | \$200 – \$4,200+         | Included in cycles                  |
| <i>Data privacy</i>              | Leaves your servers      | Never leaves the chain              |
| <i>Vendor dependency</i>         | OpenAI can revoke access | Permissionless — nobody can stop it |

// PRODUCTION &amp; OPERATIONS

## 06 GO LIVE & MONITOR

*“We went live on a Tuesday. Zero downtime. The canister just... ran. No servers to patch, no load balancers to configure, no 3am wake-up calls.”*

— Founder, B2B SaaS

### KEY ACTIONS

- ▶ Enable canister logging and monitoring from the Caffeine dashboard before you announce
- ▶ Set up cycle alerts — configure minimum threshold warnings so you never hit zero
- ▶ Configure HTTP outcall rate limits to prevent runaway cycle consumption
- ▶ Test every user flow end-to-end on mainnet before your public announcement
- ▶ Document a runbook for common operations: upgrades, top-ups, rollbacks

### OPERATIONS CHECKLIST

| TASK                             | RECOMMENDED CADENCE                                       |
|----------------------------------|-----------------------------------------------------------|
| <i>Check cycle balances</i>      | Weekly — automate alerts at 500B cycle threshold          |
| <i>Review canister logs</i>      | After every deployment and on any user-reported issue     |
| <i>Test upgrade flow</i>         | Before every production upgrade — use local replica first |
| <i>Verify custom domain cert</i> | Quarterly — IC auto-renews, but confirm in dashboard      |

// YOU'VE SEEN THE FRAMEWORK. NOW GO DEEPER.

# READY TO MIGRATE?

This is the summary. The full Game Plan includes complete Motoko code templates, real migration case studies with before/after architecture diagrams, and a step-by-step implementation guide you can follow from day one.

GET THE FULL GAME PLAN — \$19.95

Questions? Join the community at [AiSecret.Agency](https://AiSecret.Agency)