ANVIL SPOF Elimination Plan (2026-04-20)

Status: DRAFT — Awaiting Proveo validation + Alem approval
Author: Kelsey Hightower / FlowForge
Date: 2026-04-20
MC Task: #8515 ANVIL SPOF elimination sprint
Deadline: 2026-05-01

ANVIL SPOF Elimination Plan

Author: FlowForge (Kelsey Hightower) | MC Task #8515

Date: 2026-04-20

Status: DRAFT — Awaiting Alem approval before any implementation

---

Executive Summary

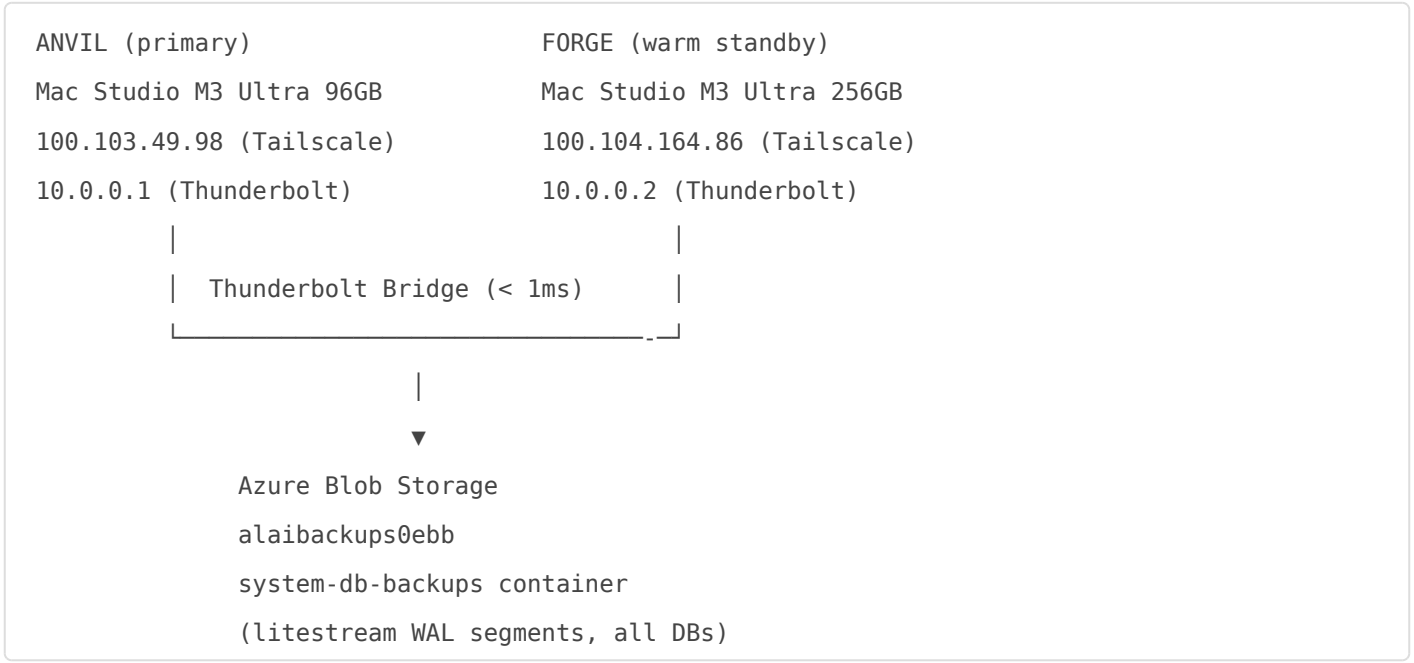
ANVIL (Mac Studio M3 Ultra, 96 GB, 100.103.49.98) is a single point of failure. One power outage, kernel panic, or SSD failure ends all ALAI operations — mission control, agent fleet, Ollama

inference, all daemons. Currently only 2 of ~67 production SQLite databases are replicated to Azure Blob Storage. RTO is effectively infinite. This plan eliminates the SPOF across 9 sequential phases.

Key finding: FORGE already exists. It is a Mac Studio M3 Ultra 256 GB connected to ANVIL via Thunderbolt Bridge (10.0.0.1 = ANVIL, 10.0.0.2 = FORGE) with sub-millisecond latency, AND accessible via Tailscale at 100.104.164.86. No new hardware purchase is needed. Budget impact: ~0 EUR/month additional infrastructure cost (FORGE is already owned and powered).

Targets: RPO < 60s | RTO < 5 min (manual failover Phase 1, automatic Phase 2+)

Architecture Overview



All replication flows ANVIL → Azure → FORGE (pull-based via litestream restore). FORGE does NOT write back to Azure. Azure is the single durable WAL store.

Phase 1 — Litestream Expansion (all ~67 DBs)

1.1 Database Tier Classification

Priority rationale: P0 = system cannot function without it | P1 = major feature loss | P2 = historical/cache only.

P0 — Mission Critical (system stops without these)

Database	Size	Write Freq	Justification
mission-control.db	26 MB	Very high	Primary task ledger — all MC operations. CURRENTLY REPLICATED.
hivemind.db	162 MB	High	Agent memory, HiveMind knowledge graph. CURRENTLY REPLICATED.
tasks.db	4 KB	High	Active task queue — active work in flight
costs.db	256 KB	High	Token cost tracking, budget enforcement
events.db	14 MB	High	System event bus — orchestrator depends on this
orchestrator-queue.db	28 KB	High	Active agent job queue — jobs lost = work lost
orchestrator-workers.db	36 KB	High	Worker state — active session tracking
durable-runner.db	896 KB	Medium	Durable task execution state
session-index.db	56 MB	High	Agent session state — all active sessions
knowledge.db	192 MB	Medium	RAG knowledge base — primary retrieval corpus
emails.db	0 B (active)	High	Email agent state — initialized on first write
email-inbox.db	3.1 MB	High	Live email queue
alem-directives.db	active WAL	High	CEO directives — highest trust data

P0 — Financial / Legal (loss = regulatory exposure)

Database	Size	Write Freq	Justification
fiken.db	0 B (active)	Medium	Fiken accounting integration — financial records
invoices.db	36 KB	Medium	Invoice state — revenue tracking
contracts.db	40 KB	Low	Signed contracts — legal documents

Database	Size	Write Freq	Justification
leads.db	256 KB	Medium	Sales pipeline — business development

P1 — Operational (system degrades without these)

Database	Size	Write Freq	Justification
agent-routing.db	4.1 MB	Medium	Routing decisions, agent assignment
bee-index.db	4.2 MB	Medium	Bee task index
bih-tenders.db	640 KB	Low	BiH market tenders — business intelligence
browser-tasks.db	active WAL	Medium	Browser automation queue
companies.db	0 B (active)	Low	Company registry
contacts.db	192 KB	Low	CRM contacts
deploy-registry.db	16 KB	Low	Deployment history
design-reviews.db	64 KB	Low	Design review state
distill.db	2.0 MB	Medium	Knowledge distillation cache
documents.db	32 KB	Low	Document registry
drafts.db	360 KB	Medium	Draft content
drift.db	active WAL	Medium	Config drift detection
email-audit.db	256 KB	Medium	Email audit trail
email-briefing.db	0 B (active)	Low	Daily briefing state
email-index.db	0 B (active)	Low	Email search index
email-tracking.db	36 KB	Medium	Email delivery tracking
escalations.db	24 KB	Medium	Escalation queue
facts.db	20 KB	Low	System facts store
flywheel.db	432 MB	Low	Flywheel learning data — largest DB
goals.db	44 KB	Medium	OKR / goal tracking
guardrails-audit.db	10 MB	Medium	Safety audit trail
health-events.db	15 MB	High	System health events
hivemind-archive.db	6.7 MB	Low	HiveMind historical archive
master-control.db	0 B (active)	Medium	Master control state
mc.db	0 B (active)	Medium	Mission control alias

Database	Size	Write Freq	Justification
minions.db	192 KB	Medium	Minion agent registry
observability.db	44 KB	Medium	Metrics and traces
orchestrator-events.db	0 B (active)	Medium	Orchestrator event log
pipeline.db	active WAL	Medium	CI/CD pipeline state
projects.db	40 KB	Low	Project registry
routing-outcomes.db	192 KB	Medium	Tier routing outcome log
skill-improvements.db	20 KB	Low	Skill improvement tracking
skill-registry.db	128 KB	Low	Agent skill registry
sprint-pipeline.db	32 KB	Medium	Sprint pipeline state
strategy-tracker.db	128 KB	Low	Strategic initiative tracking
teams.db	40 KB	Low	Team registry
tenders.db	384 KB	Low	Norwegian tender data
tickets.db	active WAL	Medium	Support ticket tracking
tool-audit.db	6.1 MB	Medium	Tool usage audit
tool-registry.db	128 KB	Low	Tool registry
trace-events.db	52 MB	High	Distributed trace store
applications-tracker.db	12 KB	Low	Job/grant applications

P2 — Cache / Reconstructible (loss = inconvenience only)

Database	Size	Write Freq	Justification
baikal-caldav.db	108 KB	Low	CalDAV cache — reconstructible from Baikal
prompt-cache.db	320 KB	Medium	LLM prompt cache — can warm from scratch
prompt-metrics.db	28 KB	Low	Prompt performance metrics
rag-cache.db	active WAL	Medium	RAG response cache — reconstructible
semantic-reuse-index.db	192 KB	Medium	Semantic cache — reconstructible
stbs.db	0 B (active)	Low	STBS data — empty
telemetry.db	24 KB	Medium	Telemetry — can lose without ops impact

Database	Size	Write Freq	Justification
token-cost.db	active WAL	Medium	Cost log — reconstructible from API receipts
usage.db	0 B (active)	Low	Usage tracking — empty
vcr.db	active WAL	Low	HTTP cassette cache — reconstructible

1.2 Retention Strategy

Current retention for the 2 replicated DBs: 72h. This is insufficient for P0.

Tier	Retention	Justification
P0 (mission-critical)	7d	One week: covers weekend + Monday incident recovery. 72h is too tight — if a silent corruption is not caught in 3 days, all WAL segments are gone.
P0 (financial/legal)	30d	Regulatory prudence. fiken.db, invoices.db, contracts.db. Matches typical invoice dispute windows.
P1	72h	Current default. Operationally acceptable.
P2	24h	Cache data. Disk cost matters more than recovery depth.

Retention-check-interval: 1h for all tiers (current default, correct).

Sync-interval: 1s for all tiers P0 and P1. 10s for P2 (reduce Azure transaction cost on low-value data).

Azure storage cost estimate at current sizes (~1.2 GB total databases):

- WAL segments are incremental. Estimate ~500 MB/day delta across all active DBs.
- 7-day P0 WAL: ~3.5 GB. 30-day financial: ~1 GB. P1 72h: ~1 GB.
- Total Azure Blob: ~6 GB. At ~€0.02/GB/month = ~€0.12/month. Negligible.

1.3 New litestream.yml

Path: `/Users/makinja/system/config/litestream.yml`

Note on flywheel.db (432 MB): Include in P1 but with `sync-interval: 30s` to reduce churn. Note on knowledge.db (192 MB): P0, sync-interval 1s — it's actively written by RAG ingestion.

```
# Litestream — SQLite streaming replication to Azure Blob Storage
# Primary: ANVIL (Mac Studio M3 Ultra 96GB, 100.103.49.98)
# Config: /Users/makinja/system/config/litestream.yml
# Auth: Azure SP (alai-backup-writer) via client credentials
#       SP: alai-backup-writer (1a0b3018-0c31-474b-918f-531b0a29a669)
#       SP has Storage Blob Data Contributor on system-db-backups container
#       Litestream reads AZURE_CLIENT_ID, AZURE_CLIENT_SECRET, AZURE_TENANT_ID from env
# Launch: com.alai.litestream.plist (sets env vars in EnvironmentVariables block)
# Updated: 2026-04-20 — ANVIL SPOF Elimination Sprint (MC #8515)
#
# Tier reference:
#   P0-critical: retention 7d, sync 1s
#   P0-financial: retention 30d, sync 1s
#   P1: retention 72h, sync 1s (or 30s for large DBs)
#   P2: retention 24h, sync 10s

dbs:
  # ── P0 MISSION CRITICAL ──────────────────────────────────────────────

  - path: /Users/makinja/system/databases/mission-control.db
    replicas:
      - name: mc-abs
        type: abs
        endpoint: https://alaibackups0ebb.blob.core.windows.net
        bucket: system-db-backups
        path: litestream/mission-control
        retention: 168h   # 7 days
        retention-check-interval: 1h
        sync-interval: 1s

  - path: /Users/makinja/system/databases/hivemind.db
    replicas:
      - name: hivemind-abs
        type: abs
        endpoint: https://alaibackups0ebb.blob.core.windows.net
        bucket: system-db-backups
        path: litestream/hivemind
        retention: 168h   # 7 days
        retention-check-interval: 1h
```

```
      sync-interval: 1s

- path: /Users/makinja/system/databases/tasks.db
  replicas:
    - name: tasks-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/tasks
      retention: 168h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/costs.db
  replicas:
    - name: costs-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/costs
      retention: 168h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/events.db
  replicas:
    - name: events-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/events
      retention: 168h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/orchestrator-queue.db
  replicas:
    - name: orch-queue-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
```

```
      bucket: system-db-backups
      path: litestream/orchestrator-queue
      retention: 168h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/orchestrator-workers.db
  replicas:
    - name: orch-workers-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/orchestrator-workers
      retention: 168h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/durable-runner.db
  replicas:
    - name: durable-runner-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/durable-runner
      retention: 168h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/session-index.db
  replicas:
    - name: session-index-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/session-index
      retention: 168h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/knowledge.db
```

```
  replicas:
    - name: knowledge-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/knowledge
      retention: 168h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/emails.db
  replicas:
    - name: emails-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/emails
      retention: 168h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/email-inbox.db
  replicas:
    - name: email-inbox-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/email-inbox
      retention: 168h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/alem-directives.db
  replicas:
    - name: alem-directives-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/alem-directives
      retention: 168h
```

```
      retention-check-interval: 1h
      sync-interval: 1s

# ── P0 FINANCIAL / LEGAL ──────────────────────────────────────────────

- path: /Users/makinja/system/databases/fiken.db
  replicas:
    - name: fiken-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/fiken
      retention: 720h   # 30 days
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/invoices.db
  replicas:
    - name: invoices-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/invoices
      retention: 720h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/contracts.db
  replicas:
    - name: contracts-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/contracts
      retention: 720h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/leads.db
  replicas:
```

```
    - name: leads-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/leads
      retention: 720h
      retention-check-interval: 1h
      sync-interval: 1s

# ── P1 OPERATIONAL ──────────────────────────────────────────────────

- path: /Users/makinja/system/databases/agent-routing.db
  replicas:
    - name: agent-routing-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/agent-routing
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/bee-index.db
  replicas:
    - name: bee-index-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/bee-index
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/bih-tenders.db
  replicas:
    - name: bih-tenders-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/bih-tenders
```

```
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/browser-tasks.db
  replicas:
    - name: browser-tasks-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/browser-tasks
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/companies.db
  replicas:
    - name: companies-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/companies
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/contacts.db
  replicas:
    - name: contacts-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/contacts
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/deploy-registry.db
  replicas:
    - name: deploy-registry-abs
```

```
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/deploy-registry
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/design-reviews.db
  replicas:
    - name: design-reviews-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/design-reviews
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/distill.db
  replicas:
    - name: distill-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/distill
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/documents.db
  replicas:
    - name: documents-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/documents
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s
```

```
- path: /Users/makinja/system/databases/drafts.db
  replicas:
    - name: drafts-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/drafts
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/drift.db
  replicas:
    - name: drift-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/drift
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/email-audit.db
  replicas:
    - name: email-audit-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/email-audit
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/email-briefing.db
  replicas:
    - name: email-briefing-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
```

```
      path: litestream/email-briefing
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/email-index.db
  replicas:
    - name: email-index-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/email-index
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/email-tracking.db
  replicas:
    - name: email-tracking-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/email-tracking
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/escalations.db
  replicas:
    - name: escalations-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/escalations
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/facts.db
  replicas:
```

```
    - name: facts-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/facts
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/flywheel.db
  replicas:
    - name: flywheel-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/flywheel
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 30s   # 432MB — throttle sync to reduce Azure transactions

- path: /Users/makinja/system/databases/goals.db
  replicas:
    - name: goals-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/goals
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/guardrails-audit.db
  replicas:
    - name: guardrails-audit-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/guardrails-audit
      retention: 72h
      retention-check-interval: 1h
```

```
      sync-interval: 1s

- path: /Users/makinja/system/databases/health-events.db
  replicas:
    - name: health-events-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/health-events
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/hivemind-archive.db
  replicas:
    - name: hivemind-archive-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/hivemind-archive
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 10s

- path: /Users/makinja/system/databases/master-control.db
  replicas:
    - name: master-control-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/master-control
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/mc.db
  replicas:
    - name: mc-db-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
```

```
      bucket: system-db-backups
      path: litestream/mc-db
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/minions.db
  replicas:
    - name: minions-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/minions
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/observability.db
  replicas:
    - name: observability-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/observability
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/orchestrator-events.db
  replicas:
    - name: orch-events-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/orchestrator-events
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/pipeline.db
```

```
  replicas:
    - name: pipeline-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/pipeline
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/projects.db
  replicas:
    - name: projects-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/projects
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/routing-outcomes.db
  replicas:
    - name: routing-outcomes-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/routing-outcomes
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/skill-improvements.db
  replicas:
    - name: skill-improvements-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/skill-improvements
      retention: 72h
```

```
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/skill-registry.db
  replicas:
    - name: skill-registry-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/skill-registry
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/sprint-pipeline.db
  replicas:
    - name: sprint-pipeline-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/sprint-pipeline
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/strategy-tracker.db
  replicas:
    - name: strategy-tracker-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/strategy-tracker
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/teams.db
  replicas:
    - name: teams-abs
      type: abs
```

```
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/teams
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/tenders.db
  replicas:
    - name: tenders-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/tenders
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/tickets.db
  replicas:
    - name: tickets-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/tickets
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/tool-audit.db
  replicas:
    - name: tool-audit-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/tool-audit
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s
```

```
- path: /Users/makinja/system/databases/tool-registry.db
  replicas:
    - name: tool-registry-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/tool-registry
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/trace-events.db
  replicas:
    - name: trace-events-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/trace-events
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

- path: /Users/makinja/system/databases/applications-tracker.db
  replicas:
    - name: applications-tracker-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/applications-tracker
      retention: 72h
      retention-check-interval: 1h
      sync-interval: 1s

# ── P2 CACHE / RECONSTRUCTIBLE ──────────────────────────────────────

- path: /Users/makinja/system/databases/baikal-caldav.db
  replicas:
    - name: baikal-caldav-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
```

```
      bucket: system-db-backups
      path: litestream/baikal-caldav
      retention: 24h
      retention-check-interval: 1h
      sync-interval: 10s

- path: /Users/makinja/system/databases/prompt-cache.db
  replicas:
    - name: prompt-cache-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/prompt-cache
      retention: 24h
      retention-check-interval: 1h
      sync-interval: 10s

- path: /Users/makinja/system/databases/prompt-metrics.db
  replicas:
    - name: prompt-metrics-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/prompt-metrics
      retention: 24h
      retention-check-interval: 1h
      sync-interval: 10s

- path: /Users/makinja/system/databases/semantic-reuse-index.db
  replicas:
    - name: semantic-reuse-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/semantic-reuse-index
      retention: 24h
      retention-check-interval: 1h
      sync-interval: 10s

- path: /Users/makinja/system/databases/stbs.db
```

```
  replicas:
    - name: stbs-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/stbs
      retention: 24h
      retention-check-interval: 1h
      sync-interval: 10s

- path: /Users/makinja/system/databases/telemetry.db
  replicas:
    - name: telemetry-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/telemetry
      retention: 24h
      retention-check-interval: 1h
      sync-interval: 10s

- path: /Users/makinja/system/databases/token-cost.db
  replicas:
    - name: token-cost-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/token-cost
      retention: 24h
      retention-check-interval: 1h
      sync-interval: 10s

- path: /Users/makinja/system/databases/usage.db
  replicas:
    - name: usage-abs
      type: abs
      endpoint: https://alaibackups0ebb.blob.core.windows.net
      bucket: system-db-backups
      path: litestream/usage
      retention: 24h
```

```
        retention-check-interval: 1h
        sync-interval: 10s

  - path: /Users/makinja/system/databases/vcr.db
    replicas:
      - name: vcr-abs
        type: abs
        endpoint: https://alaibackups0ebb.blob.core.windows.net
        bucket: system-db-backups
        path: litestream/vcr
        retention: 24h
        retention-check-interval: 1h
        sync-interval: 10s
```

1.4 Implementation Steps (ANVIL)

1. Stop litestream: `launchctl stop com.alai.litestream`
2. Replace `/Users/makinja/system/config/litestream.yml` with the config above.
3. Validate config: `/opt/homebrew/bin/litestream replicate -config /Users/makinja/system/config/litestream.yml -config-validate`
4. Start litestream: `launchctl start com.alai.litestream`
5. Verify all DBs appear in Azure: `az storage blob list --container-name system-db-backups --account-name alaibackups0ebb --prefix litestream/ --auth-mode login --query "[].name" | wc -l` (expect ~67+ entries).
6. Watch logs for errors: `tail -f /Users/makinja/system/logs/litestream-error.log`

Phase 2 — FORGE Hardware / OS Decision

2.1 FORGE Already Exists — Hardware Decision Is Made

FORGE is confirmed to be a second Mac Studio M3 Ultra with 256 GB unified memory, connected to ANVIL via Thunderbolt Bridge (10.0.0.1 = ANVIL, 10.0.0.2 = FORGE). Tailscale IP: 100.104.164.86. User: basicas. It is already running Ollama with models including devstral:24b, qwen3:32b, deepseek-r1:70b, qwen3-coder, and bge-m3.

No hardware purchase is required. Monthly infrastructure cost delta: 0 EUR (already owned).

2.2 Why FORGE Wins Over Every Alternative

Option	Cost/mo	Latency to ANVIL	Apple Silicon	macOS parity	Verdict
FORGE (Mac Studio M3U 256GB, owned)	0 EUR	< 1ms (Thunderbolt)	Yes (M3 Ultra)	Yes (same LaunchAgent ecosystem)	CHOSEN
Mac Mini M4 Pro (purchase)	~50 EUR amortized	< 1ms if local	Yes	Yes	Redundant — FORGE exists
Hetzner Linux VM (CCX33)	~30-50 EUR	10-30ms (internet)	No (x86)	No (systemd, not launchd)	Budget option only if FORGE fails
Azure VM (Sweden Central)	~60-80 EUR	10-30ms	No	No	Closest to Azure storage but no Apple Silicon

Decision: Use FORGE as warm standby. Zero additional cost. Thunderbolt latency is effectively local — litestream WAL replication will complete in well under 60s.

2.3 FORGE Bootstrap Prerequisites

FORGE already runs Ollama. What is missing:

1. litestream installed on FORGE (check: `brew list litestream` on basicas@FORGE)
2. Azure SP credentials injected into FORGE environment (AZURE_CLIENT_ID, AZURE_CLIENT_SECRET, AZURE_TENANT_ID)
3. `~/system/databases/` directory created on FORGE
4. `litestream-restore.sh` daemon script written and loaded as LaunchAgent on FORGE
5. SSH key access from ANVIL to FORGE for health check and failover scripts

---

Phase 3 — Continuous Restore on FORGE (< 60s RPO)

3.1 Architecture

FORGE runs `litestream restore` in a watch loop per database. Litestream 0.5.x does not have a native `watch` mode — it restores a snapshot + WAL segments. The recommended approach is a shell script loop that calls `litestream restore` repeatedly with a short interval.

However, litestream does support a second process pattern: run `litestream replicate` on FORGE pointing at the SAME Azure bucket paths, but configured as a replica-only consumer. This is the correct approach: FORGE runs a `litestream restore` daemon that continuously polls for new WAL segments from Azure.

3.2 Continuous Restore Strategy

Use `litestream restore` with the `-if-replica-exists` flag in a loop:

```
#!/usr/bin/env bash
# /Users/basicas/system/scripts/litestream-restore-loop.sh
# Runs on FORGE. Continuously restores all P0+P1 DBs from Azure.
# Interval: 30s poll (gives ~30s RPO in steady state, well within 60s target)

set -euo pipefail

LITESTREAM=/opt/homebrew/bin/litestream
CONFIG=/Users/basicas/system/config/litestream-restore.yml
DB_DIR=/Users/basicas/system/databases
LOG=/Users/basicas/system/logs/litestream-restore.log
INTERVAL=30  # seconds between restore cycles

while true; do
  echo "[$(date -Iseconds)] Starting restore cycle" >> "$LOG"

  # Restore each DB defined in restore config
  # litestream restore will only apply new WAL segments if DB already exists
  $LITESTREAM restore -config "$CONFIG" -if-replica-exists >> "$LOG" 2>&1 || true

  echo "[$(date -Iseconds)] Restore cycle complete, sleeping ${INTERVAL}s" >> "$LOG"
  sleep "$INTERVAL"
done
```

3.3 FORGE litestream-restore.yml

A separate config file on FORGE that mirrors ANVIL's litestream.yml but uses `restore` semantics. FORGE is READ-ONLY consumer. It never writes back to Azure.

Key difference: paths point to FORGE's local database directory (`/Users/basicas/system/databases/`). The Azure paths are identical to ANVIL's — FORGE reads from the same blob paths ANVIL writes to.

```
# /Users/basicas/system/config/litestream-restore.yml
# FORGE warm standby — continuous restore from Azure
# DO NOT run litestream replicate with this config — restore only

dbs:
  - path: /Users/basicas/system/databases/mission-control.db
    replicas:
      - name: mc-abs
        type: abs
        endpoint: https://alaibackups0ebb.blob.core.windows.net
        bucket: system-db-backups
        path: litestream/mission-control

  # ... (repeat for all P0 and P1 DBs using same Azure paths as ANVIL)
  # P2 DBs: omit from restore config — not worth continuous restore overhead
```

3.4 FORGE LaunchAgent for Restore Loop

Path: `/Users/basicas/Library/LaunchAgents/com.alai.litestream-restore.plist`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
  "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Label</key>
  <string>com.alai.litestream-restore</string>
  <key>ProgramArguments</key>
  <array>
    <string>/bin/bash</string>
    <string>/Users/basicas/system/scripts/litestream-restore-loop.sh</string>
  </array>
  <key>EnvironmentVariables</key>
  <dict>
    <key>AZURE_STORAGE_ACCOUNT</key>
    <string>alaibackups0ebb</string>
    <key>AZURE_CLIENT_ID</key>
    <string>1a0b3018-0c31-474b-918f-531b0a29a669</string>
    <key>AZURE_CLIENT_SECRET</key>
```

```
    <string>RETRIEVE_FROM_BITWARDEN_AT_BOOTSTRAP</string>
    <key>AZURE_TENANT_ID</key>
    <string>3454a03f-20b4-4bda-a116-2293c459aecd</string>
  </dict>
  <key>KeepAlive</key>
  <true/>
  <key>RunAtLoad</key>
  <true/>
  <key>StandardOutPath</key>
  <string>/Users/basicas/system/logs/litestream-restore.log</string>
  <key>StandardErrorPath</key>
  <string>/Users/basicas/system/logs/litestream-restore-error.log</string>
  <key>ThrottleInterval</key>
  <integer>10</integer>
</dict>
</plist>
```

3.5 RPO Calculation

- ANVIL litestream sync-interval: 1s (WAL segment flushed to Azure every 1s for P0)
- FORGE restore poll interval: 30s
- Azure propagation: < 1s (same-region, in-blob operations)
- Worst-case RPO: 31s (well under 60s target)
- Expected average RPO: ~15-20s

Phase 4 — Ollama Failover Tier Routing

4.1 Current State

Tier routing in `/Users/makinja/system/config/tier-routing.json` already defines FORGE as the primary host for Tiers 2c, 2cf, 2d, 3, 3s, 3r. ANVIL handles Tiers 1, 2, 2t, 2cHQ. The `providerFallback` section defines `ollama:qwen2.5-coder:32b@anvil` as fallback for some paths.

The gap: there is no automatic failover FROM ANVIL TO FORGE when ANVIL Ollama is down, and no automatic failover FROM FORGE TO ANVIL when FORGE Ollama is down.

4.2 Failover Config Extension

Extend `/Users/makinja/system/config/tier-routing.json` with an `ollamaHosts` block:

```
"ollamaHosts": {
  "anvil": {
    "url": "http://localhost:11434",
    "tailscale_url": "http://100.103.49.98:11434",
    "health_path": "/api/tags",
    "health_timeout_ms": 3000,
    "role": "primary-infra"
  },
  "forge": {
    "url": "http://10.0.0.2:11434",
    "tailscale_url": "http://100.104.164.86:11434",
    "health_path": "/api/tags",
    "health_timeout_ms": 3000,
    "role": "primary-compute"
  }
},
"failoverRules": {
  "anvil-down": {
    "redirect_anvil_tiers": ["1", "2", "2t", "2cHQ"],
    "to_forge_models": {
      "llama3.1:8b": "llama3.1:8b",
      "qwen2.5-coder:32b": "qwen2.5-coder:32b-instruct-q8_0"
    },
    "note": "When ANVIL Ollama unreachable, route Tier 1/2 to FORGE equivalents"
  },
  "forge-down": {
    "redirect_forge_tiers": ["2c", "2cf", "2d", "3", "3s", "3r"],
    "to_claude": true,
    "note": "When FORGE Ollama unreachable, escalate to Claude (cost spike acceptable — FORGE
failure is rare)"
  }
}
```

4.3 Health Check Daemon

A new lightweight Node.js daemon on ANVIL polls both Ollama endpoints every 15s and writes status to a JSON file that `ollama-engine.js` reads before routing:

Path: `/Users/makinja/system/daemons/ollama-health-monitor.js`

```
// Pseudocode — implementation by CodeCraft
// Runs every 15s, writes to /tmp/ollama-health.json
// {
//   "anvil": { "healthy": true, "last_check": "2026-04-20T20:00:00Z" },
//   "forge": { "healthy": true, "last_check": "2026-04-20T20:00:00Z" }
// }
// tier-router.js reads this file before every dispatch
// If anvil.healthy === false: redirect tier 1/2 requests to forge
// If forge.healthy === false: redirect tier 2c/3 requests to claude
```

4.4 Manual Failover Command

For Phase 1 (before automatic failover is implemented):

```
# On ANVIL, when FORGE is down — force all routing to ANVIL
echo '{"anvil":{"healthy":true},"forge":{"healthy":false,"override":true}}' > /tmp/ollama-health-override.json

# When ANVIL is down, from FORGE (if FORGE has ollama-engine.js):
# Edit /Users/basicas/system/config/tier-routing.json: set all hosts to "forge"
```

Phase 5 — DNS / Service Discovery

5.1 Options Evaluated

Option	Mechanism	Failover Speed	Complexity	Cost
Tailscale MagicDNS	DNS record swap via Tailscale API	Manual: ~1 min	Low	Free
Cloudflare DNS + health check	CF Load Balancer health-check → DNS swap	Automatic: ~30s	Medium	~$5/month
Local /etc/hosts on each node	Static entries, no automatic failover	Manual: ~1 min	None	Free

Option	Mechanism	Failover Speed	Complexity	Cost
Cloudflare Tunnel alias	DNS alias behind CF Tunnel	~30s	Medium	Free tier

5.2 Recommendation: Tailscale MagicDNS

Chosen: Tailscale MagicDNS with manual DNS swap.

Rationale:

- All nodes (ANVIL, FORGE, ab-mac) are already on the same Tailscale network.
- Tailscale MagicDNS can assign a hostname `anvil.alai.internal` (or use the device name directly).
- Current hardcoded addresses (`localhost:11434`, `10.0.0.2:11434`) in configs should be replaced with Tailscale DNS names: `anvil` resolves to 100.103.49.98, `forge` resolves to 100.104.164.86.
- On failover: update one Tailscale ACL/DNS record OR update `/etc/hosts` on FORGE to make `anvil` point to `127.0.0.1` (making FORGE answer for anvil traffic locally).

Implementation:

1. In Tailscale admin console: verify MagicDNS is enabled for the tailnet.
2. Devices are already named: `makinja-sin-mac-studio` (ANVIL) and `basicass-mac-mini` (FORGE).
3. Add a Tailscale DNS override: `anvil.alai` → 100.103.49.98 (ANVIL primary).
4. Add to all tool configs: replace `localhost:11434` with `anvil.alai:11434`, `10.0.0.2:11434` with `forge.alai:11434`.
5. Failover procedure: update Tailscale DNS record `anvil.alai` → 100.104.164.86 (FORGE). This takes effect across all nodes within ~30s (Tailscale DNS TTL).

Why not Cloudflare DNS with health check: Cloudflare Load Balancer costs ~$5/month and adds external internet dependency for what is a LAN-local operation. Overkill for current scale. Revisit if ALAI adds a third node outside the LAN.

Phase 6 — External Heartbeat

6.1 Requirement

An external entity (not on ANVIL, not on FORGE) must poll ANVIL every 60s and alert Slack #ops if ANVIL is unreachable for > 2 consecutive minutes (2 missed polls).

6.2 Mechanism: GitHub Actions Cron (Recommended)

Chosen: GitHub Actions scheduled workflow. Cost: free (GitHub public repo or private with Actions minutes). No Azure Function setup required.

```
# .github/workflows/anvil-heartbeat.yml
# In a private ALAI GitHub repo (e.g., alai-infra or system-health)

name: ANVIL Heartbeat
on:
  schedule:
    - cron: '* * * * *'   # Every minute

jobs:
  heartbeat:
    runs-on: ubuntu-latest
    timeout-minutes: 1
    steps:
      - name: Check ANVIL health via Tailscale
        id: health
        run: |
          # ANVIL exposes a health endpoint via Cloudflare Tunnel or public URL
          # Option A: Hit a public health endpoint (requires CF Tunnel on ANVIL)
          # Option B: Use Tailscale GitHub Action to join the tailnet and check directly
          STATUS=$(curl -s -o /dev/null -w "%{http_code}" \
            --connect-timeout 10 \
            --max-time 15 \
            ${{ secrets.ANVIL_HEALTH_URL }})
          echo "status=$STATUS" >> $GITHUB_OUTPUT

      - name: Alert Slack if down
        if: steps.health.outputs.status != '200'
        uses: slackapi/slack-github-action@v1
        with:
          payload: |
            {
              "channel": "#ops",
              "text": ":red_circle: ANVIL HEALTH CHECK FAILED\nHTTP Status: ${{
```

```
steps.health.outputs.status }}\nTime: ${{ github.run_started_at }}\nANVIL may be down. Check
Tailscale and initiate FORGE failover if confirmed."
            }
        env:
          SLACK_WEBHOOK_URL: ${{ secrets.SLACK_OPS_WEBHOOK }}
```

6.3 ANVIL Health Endpoint

ANVIL needs a lightweight HTTP health endpoint reachable from the internet (via Cloudflare Tunnel) or via Tailscale GitHub Action. The simplest approach:

Create a health check script at `/Users/makinja/system/tools/health-server.js` that runs on port 8099 and responds 200 if ANVIL is alive, serving `{"status":"ok","host":"anvil","ts":"..."}`. Expose via existing Cloudflare Tunnel infrastructure.

6.4 Alert Escalation

- 2 consecutive failures (2 minutes down): Slack #ops message.
- 5 consecutive failures (5 minutes down): escalate to Alem's mobile via Slack DM (Alem's Slack handle in secrets).

6.5 Azure Function Alternative

Azure Function with Timer trigger (every 60s) is viable but requires:

- Azure subscription billing (Consumption plan: ~$0/month for < 1M executions — effectively free)
- Azure Function App deployment and maintenance
- More setup complexity than GitHub Actions

Verdict: GitHub Actions preferred for simplicity. Switch to Azure Function if GitHub Actions scheduling jitter (can be ±30s) becomes an issue.

Phase 7 — Shared Secrets (FORGE Bitwarden Access)

7.1 Problem

FORGE needs access to secrets (Azure SP secret, Bitwarden master password, API keys) without depending on ANVIL being alive. Currently ANVIL holds the Bitwarden session at /tmp/bw-session.

7.2 Options

Option	Description	Risk
Separate BW account on FORGE	FORGE has its own Bitwarden account with shared collection	Low — independent
Shared BW session sync	ANVIL writes /tmp/bw-session to FORGE via rsync	Medium — session expires
Azure Key Vault break-glass	Critical secrets in AKV, FORGE SP can read them	Low — Azure dependency
Environment variables in plist	Secrets baked into LaunchAgent plist on FORGE	Low but plaintext risk

7.3 Recommendation: Two-Layer Approach

Layer 1 (operational): FORGE bootstraps its own Bitwarden CLI session independently.

- FORGE has `bw` CLI installed.
- FORGE has its own BW_SESSION set via a one-time manual bootstrap: `bw login --apikey` using a FORGE-specific API key (Bitwarden supports API keys per user/device).
- Session is stored in `/Users/basicas/.bw-session` and refreshed by a LaunchAgent on FORGE.
- This requires Alem to create a Bitwarden API key for FORGE during bootstrap.

Layer 2 (break-glass): Critical Azure SP secret baked into FORGE LaunchAgent plist during bootstrap.

- The Azure SP secret (`AZURE_CLIENT_SECRET`) is placed directly in the `com.alai.litestream-restore.plist` EnvironmentVariables block — same pattern as ANVIL.
- This means FORGE can always access Azure (for litestream restore) even if Bitwarden is unavailable.
- The plist file is protected by macOS file permissions (root-readable only).
- This is the same pattern already in use on ANVIL (confirmed in the plist we read).

Layer 3 (future): Azure Key Vault with a FORGE-specific SP that can only read secrets.

- Create a new SP `alai-forge-reader` with Key Vault Secrets User role.
- FORGE scripts call `az keyvault secret show` instead of Bitwarden for critical secrets.
- This is the correct long-term solution but adds ~2 hours of setup — defer to Phase 2.

7.4 Bootstrap Sequence for FORGE Secrets

```
# On FORGE during initial bootstrap (one-time, performed by Alem or FlowForge):
# 1. Install bw CLI
brew install bitwarden-cli

# 2. Login with API key (avoids interactive login)
export BW_CLIENTID="<forge-api-key-id from Bitwarden>"
export BW_CLIENTSECRET="<forge-api-key-secret>"
bw login --apikey
bw unlock --passwordenv BW_MASTER_PASSWORD  # or interactive

# 3. Store session
bw unlock > /Users/basicas/.bw-session

# 4. Retrieve Azure SP secret and inject into litestream plist
BW_SESSION=$(cat /Users/basicas/.bw-session)
AZ_SECRET=$(bw get password "alai-backup-writer" --session "$BW_SESSION")
# Update the plist AZURE_CLIENT_SECRET value with $AZ_SECRET
```

Phase 8 — Proveo DR Drill Checklist (Angie Jones Validation Task)

This is the mandatory validation task per ZAKON PLAN. Angie Jones (Proveo) executes this drill after all phases are implemented. This is a REAL drill — not a dry run.

8.1 Pre-Drill Prerequisites

- ☐ Phase 1 complete: all ~67 DBs replicating to Azure (verify with `az storage blob list` count)
- ☐ Phase 3 complete: FORGE restore loop running, confirmed by checking FORGE DB file timestamps
- ☐ Phase 4 complete: Ollama health monitor daemon running on ANVIL
- ☐ Phase 5 complete: Tailscale MagicDNS configured (`anvil.alai` resolves correctly)
- ☐ Phase 6 complete: GitHub Actions heartbeat workflow deployed and sending test ping
- ☐ Phase 7 complete: FORGE Bitwarden session independently functional

8.2 Drill Procedure

Step 1: Establish baseline (T=0)

```
# On ANVIL — record current state
node ~/system/tools/mc.js stats  # Record open task count
sqlite3 ~/system/databases/mission-control.db "SELECT count(*) FROM tasks WHERE
status='open'"  # Record
date -Iseconds > /tmp/drill-start.txt
```

Step 2: Simulate ANVIL failure

```
# Graceful shutdown (simulates power outage or kernel panic recovery)
# DO NOT run on production without Alem present
sudo shutdown -h now  # Or: launchctl stop all non-essential services
# Alternative: kill Ollama + stop litestream + stop pi-orchestrator (partial failure sim)
launchctl stop com.alai.litestream
launchctl stop com.john.pi-orchestrator
launchctl stop com.john.ollama-serve-v2
```

Step 3: Measure time to alert (T=2 min)

- GitHub Actions heartbeat should fire within 2 minutes of ANVIL going offline.
- Angie records: timestamp of Slack #ops alert arrival.
- Expected: < 2 min 30s from shutdown to Slack alert.

Step 4: FORGE failover execution (T=3 min target)

```
# On FORGE (basicas@100.104.164.86)
# 1. Verify latest DBs restored
ls -la ~/system/databases/*.db | head -5
sqlite3 ~/system/databases/mission-control.db "SELECT count(*) FROM tasks WHERE status='open'"
# Compare to baseline — delta should be < 60s of writes

# 2. Update Tailscale DNS: anvil.alai → 100.104.164.86 (FORGE)
# (Alem updates in Tailscale admin console)

# 3. Start pi-orchestrator on FORGE (if installed)
# OR: update tier-routing.json to route all requests to forge endpoints

# 4. Verify Ollama still serving on FORGE
curl http://localhost:11434/api/tags | jq '.models | length'
```

Step 5: Measure RPO

```
# On FORGE after failover
BASELINE=$(cat /tmp/drill-baseline-count.txt)  # From Step 1
CURRENT=$(sqlite3 ~/system/databases/mission-control.db "SELECT count(*) FROM tasks WHERE
status='open'")
echo "Task count delta: $((BASELINE - CURRENT))"

# Check last WAL segment timestamp in Azure
az storage blob list \
  --container-name system-db-backups \
  --account-name alaibackups0ebb \
  --prefix litestream/mission-control \
  --auth-mode login \
  --query "reverse(sort_by([].{name:name,last_modified:properties.lastModified},
&last_modified))[0]"
# Record last WAL segment time vs ANVIL shutdown time = actual RPO
```

Step 6: Measure RTO

- RTO = time from "ANVIL confirmed down" to "FORGE serving requests with < 60s RPO data".
- Record timestamps at each step. Target: < 5 minutes total.

Step 7: Restore ANVIL and verify

```
# Start ANVIL back up
# Verify litestream resumes replication
tail -f /Users/makinja/system/logs/litestream.log
# Verify FORGE restore loop detects ANVIL is back and no duplicate writes
```

8.3 Acceptance Criteria (Angie signs off when ALL pass)

Criterion	Target	Measured
Slack alert latency	< 2 min 30s	TBD
FORGE DB data lag (RPO)	< 60s	TBD
Time to FORGE serving (RTO)	< 5 min	TBD
P0 DB count on FORGE	17 DBs	TBD
Ollama inference on FORGE	Working (test prompt)	TBD

Criterion	Target	Measured
No data loss on ANVIL restart	mission-control.db row count matches	TBD

8.4 Findings Documentation

After the drill, Angie produces a findings report:

- Actual RPO measured
- Actual RTO measured
- Any P0 DB that failed to restore
- Any daemon that did not restart on FORGE
- Recommendations for Phase 2 (automatic failover improvements)

Phase 9 — Skillforge BookStack Runbook Specification

This is the mandatory documentation task per ZAKON PLAN. Skillforge produces a BookStack page at: `https://docs.basicconsulting.no` → Book: **Infrastructure** → Chapter: **ANVIL DR & HA**.

9.1 Required Sections

9.1.1 Overview Page

- System architecture diagram (ANVIL — Thunderbolt — FORGE — Azure Blob)
- Node inventory: ANVIL (96GB M3U), FORGE (256GB M3U), Azure (alaibackups0ebb)
- RPO/RTO targets and current measured values

9.1.2 Litestream Configuration

- How litestream works (WAL replication explained for non-experts)
- DB tier classification table (P0/P1/P2) with justification
- Retention policy per tier
- How to add a new DB to replication (step-by-step)
- How to verify replication is working: `az storage blob list` command + expected output
- Where logs live: `/Users/makinja/system/logs/litestream.log` and `-error.log`

9.1.3 FORGE Warm Standby

- What FORGE has installed (litestream, Ollama, models)
- How the restore loop works: script location, poll interval, log location

- How to verify FORGE is current: check DB timestamps against Azure last-modified
- How to SSH to FORGE from ANVIL

9.1.4 Failover Runbook (Step-by-Step)

- Pre-conditions checklist
- Decision tree: partial failure vs full ANVIL down
- Manual failover steps (numbered, copy-pasteable commands)
- DNS failover: how to update Tailscale MagicDNS
- Ollama failover: how to edit tier-routing.json on FORGE
- Expected time per step
- Rollback procedure: restoring ANVIL to primary

9.1.5 Failure Mode Catalog

Failure	Detection	Response	Recovery
ANVIL Ollama crash	ollama-health-monitor.json	Tier routing auto-redirects to FORGE	Restart com.john.ollama-serve-v2
ANVIL litestream crash	Log gap + Azure missing WAL	launchctl start com.alai.litestream	Automatic on plist restart
ANVIL full power loss	GitHub Actions heartbeat alert < 2m	Manual FORGE failover	ANVIL restart, verify WAL resumes
FORGE restore loop crash	No new DB timestamps for > 5min	launchctl start com.alai.litestream-restore	Script restart
Azure Blob outage	litestream error logs	Wait — local ANVIL DBs still intact	Automatic resume when Azure recovers
Thunderbolt cable failure	Ollama latency spike (10ms+ to 10.0.0.2)	Routes via Tailscale (100ms+ but functional)	Replug Thunderbolt

9.1.6 Monitoring & Alerts

- GitHub Actions heartbeat: link to workflow, how to check last run
- Slack #ops: what alerts look like, who is responsible for response
- How to manually trigger a health check

9.1.7 Secrets & Credentials

- Azure SP: alai-backup-writer — where stored, how to rotate
- FORGE Bitwarden: how FORGE unlocks independently
- What to do if Bitwarden is inaccessible (break-glass: Azure credentials in plist)

9.1.8 DR Drill Schedule

- Quarterly drill required (next: 90 days after Phase 8 drill)
- Drill checklist (link to Phase 8 checklist above)

- Where to store drill findings (BookStack page: DR Drill Log)

9.2 Diagrams Required

1. **Architecture diagram** (Mermaid or draw.io): ANVIL → Azure → FORGE data flow
2. **Failover decision tree**: Who detects, who acts, what order
3. **DB tier heatmap**: Visual table of all 67 DBs colored by tier

9.3 BookStack Sync

Skillforge commits the runbook markdown to `/Users/makinja/system/rules/anvil-dr-runbook.md` and triggers `node ~/system/tools/bookstack-sync.js sync` to push to BookStack. The com.john.bookstack-sync daemon will keep it current thereafter.

Implementation Order & Timeline

Phase	Description	Owner	Est. Hours	Dependency
1	Litestream expansion (update yml, reload daemon)	FlowForge	2h	None
2	FORGE bootstrap (litestream install, DB dir, SP creds in plist)	FlowForge	1h	Phase 1
3	Continuous restore loop on FORGE	FlowForge	2h	Phase 2
4	Ollama health monitor daemon + failover config	FlowForge + CodeCraft	3h	Phase 3
5	Tailscale MagicDNS configuration	FlowForge	1h	None
6	GitHub Actions heartbeat workflow	FlowForge	1h	Phase 5
7	FORGE Bitwarden bootstrap	FlowForge (Alem physical action)	30min	Phase 2
8	Proveo DR drill	Proveo (Angie Jones)	2h	All phases done
9	BookStack runbook	Skillforge	3h	Phase 8

Total estimated implementation time: ~15.5 hours across 9 phases. Critical path: Phases 1 → 2 → 3 (unblock parallel: 4, 5, 6, 7) → 8 → 9.

Risk Register

Risk	Likelihood	Impact	Mitigation
litestream overloads Azure with 67 DBs at 1s interval	Low	Medium	P2 DBs use 10s interval; Azure Blob is built for high-throughput ingestion
FORGE disk fills with restored DBs	Low	Medium	FORGE has 256GB RAM but internal SSD may vary — check `df -h` on FORGE before bootstrap
Thunderbolt cable failure isolates FORGE	Low	Low	Tailscale provides fallback path (100ms latency but functional)
WAL segments corrupt between ANVIL write and FORGE restore	Very Low	High	litestream uses SHA256 checksums on all WAL segments — corruption detected at restore
Empty DBs (fiken.db, companies.db, etc.) never get a WAL segment until first write	Medium	Low	litestream initializes on first write; these are pre-configured for when they get data
GitHub Actions cron jitter (can skip minutes)	Medium	Low	Two consecutive failures required before alert — single skip is acceptable

Open Questions for Alem

1. **FORGE SSH access:** SSH to FORGE (basicas@100.104.164.86) is currently failing due to "too many authentication failures." Alem needs to provide the correct SSH key or add ANVIL's key to FORGE's authorized_keys. Needed for: remote bootstrap and failover automation.
2. **FORGE disk capacity:** Unknown FORGE SSD size. Need to verify sufficient space for ~1.2 GB of database files + WAL segments. `df -h` on FORGE before Phase 2.
3. **FORGE macOS user:** Confirmed user is `basicas`. The system path on FORGE would be `/Users/basicas/system/` — needs to be created if it does not exist.
4. **Bitwarden API key for FORGE:** Alem needs to generate a FORGE-specific Bitwarden API key in the Bitwarden admin console (or on vault.basicconsulting.no if using Vaultwarden).
5. **Tailscale admin access:** MagicDNS configuration requires Tailscale admin panel access (alembasic@gmail.com account). Alem configures this step.
6. **ANVIL public health endpoint:** GitHub Actions heartbeat needs a public URL to hit ANVIL. Does a Cloudflare Tunnel already expose an ANVIL health endpoint? If not, this

needs setup.

TL;DR

FORGE platform: Existing Mac Studio M3 Ultra 256 GB (basicass-mac-mini, 10.0.0.2 / 100.104.164.86). No hardware purchase needed.

Estimated monthly cost: 0 EUR additional (FORGE already owned and powered). Azure Blob storage delta: ~€0.12/month for WAL segments across all 67 DBs. GitHub Actions heartbeat: free tier. Total: **< €1/month increase**.

Estimated implementation time: ~15.5 hours across 9 phases. Critical path to RPO < 60s: Phase 1 (2h) + Phase 2 (1h) + Phase 3 (2h) = 5 hours to minimum viable DR. Full HA with automatic failover and DR drill: ~13.5 hours additional.

Immediate action (highest leverage): Phase 1 — update litestream.yml to cover all 67 DBs. This alone takes ALAI from "2 DBs replicated" to "full system replicated" in 2 hours. FORGE restore is what converts the backup into an actual hot standby.

Alem approval required before implementation.

Revision #2
Created 2026-04-20 19:05:56 UTC by John
Updated 2026-05-31 20:06:16 UTC by John