MLX Router — Local Inference Gateway

MLX Router — Local Inference Gateway

MLX Router — Local Inference Gateway

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

Service: mlx-router (com.alai.mlx-router) **Port:** 11500 (127.0.0.1) **Status:** Production (2026-05-01) **Owner:** ALAI System Infrastructure **MC:** #10429

Overview

MLX Router is ALAI’s local inference gateway that routes AI inference requests to zero-cost MLX models running on ANVIL (Mac Studio M3 Ultra, 96GB). It provides tier-fallback routing: tier1 MLX (local, $0 cost) → tier2 FORGE Ollama ($0 cost) → tier3 Anthropic API (metered cost).

Purpose: Reduce inference costs by offloading read-only agent workloads to local MLX models. Anthropic API is reserved for tier3 fallback only.

Cost Savings: All MLX and FORGE requests logged at cost_usd=0. Estimated 95%+ reduction in inference costs for wired agents.

Architecture

```
flowchart LR
    subgraph Caller
        A[Agent/Client]
    end

    subgraph MLX-Router["mlx-router.js :11500"]
        R[Route by model_class]
    end

    subgraph Tier1["Tier 1: MLX Local (ANVIL)"]
        M1[classify → :11437<br/>Qwen3-8B-4bit]
        M2[code → :11438<br/>Qwen2.5-Coder-32B]
        M3[reason → :11435<br/>Gemma-4-26B]
        M4[audit → :11436<br/>Qwen3-32B]
    end

    subgraph Tier2["Tier 2: FORGE Ollama"]
        F[10.0.0.2:11434<br/>qwen3/deepseek/etc]
    end

    subgraph Tier3["Tier 3: Anthropic API"]
        C[claude-haiku/sonnet/opus]
    end

    A -->|POST /v1/chat| R
    R -->|Health: UP| M1
    R -->|Health: UP| M2
    R -->|Health: UP| M3
    R -->|Health: UP| M4
    R -->|Tier1 DOWN| F
    F -->|Tier2 FAIL| C

    M1 -.->|cost_usd=0| CT[cost-tracker.js]
    M2 -.->|cost_usd=0| CT
    M3 -.->|cost_usd=0| CT
    M4 -.->|cost_usd=0| CT
    F -.->|cost_usd=0| CT
    C -.->|metered| CT
```

Service Management

Start

```
launchctl bootstrap gui/$(id -u) ~/Library/LaunchAgents/com.alai.mlx-router.plist
```

Stop

```
launchctl bootout gui/$(id -u)/com.alai.mlx-router
```

Restart

```
launchctl bootout gui/$(id -u)/com.alai.mlx-router
launchctl bootstrap gui/$(id -u) ~/Library/LaunchAgents/com.alai.mlx-router.plist
```

Check Status

```
launchctl print gui/$(id -u)/com.alai.mlx-router
# Look for: state = running
# Get PID from output
```

View Logs

```
# Stdout (health probes, routing decisions)
tail -f /tmp/com.alai.mlx-router.stdout.log

# Stderr (errors)
tail -f /tmp/com.alai.mlx-router.stderr.log
```

Health Check

Endpoint

```
curl -s http://127.0.0.1:11500/health | jq
```

Expected Output

```
{
  "status": "ok",
  "endpoints": {
    "classify": { "available": true, "lastCheck": "2026-05-01T09:00:00.000Z", "latencyMs": 8
},
    "code":     { "available": true, "lastCheck": "2026-05-01T09:00:00.000Z", "latencyMs": 7
},
    "reason":   { "available": true, "lastCheck": "2026-05-01T09:00:00.000Z", "latencyMs": 3
},
    "audit":    { "available": true, "lastCheck": "2026-05-01T09:00:00.000Z", "latencyMs": 5 }
  }
}
```

Healthy state: All four endpoints show `available: true`, latency <50ms.

Unhealthy state: If `available: false`, that model_class will fall to tier2 FORGE on next request.

Model Classes

model_class	Port	Model	RAM (GB)	Use Case	Latency
classify	11437	Qwen3-8B-4bit	5	Classification, routing, QA	~14s
code	11438	Qwen2.5-Coder-32B-Instruct	19	Code generation, review	~117s
reason	11435	Gemma-4-26B (MoE 4B active)	15	Reasoning, synthesis, validation	~94s
audit	11436	Qwen3-32B-4bit	17	Architecture audit, analysis	~120s (est)

Note on latency: MLX inference is sequential and slow. 8B models take ~14s, 32B models take ~94-117s. **Not suitable for synchronous user-facing work.** Use for background/async agent tasks only.

Tier Fallback Chain

1. **Tier 1 — MLX Local (ANVIL):** 127.0.0.1 ports 11435-11438, cost=$0
   - Health-gated: If endpoint `available: false`, skip to tier2
   - Timeout: 120s
2. **Tier 2 — FORGE Ollama:** 10.0.0.2:11434, cost=$0
   - Models: qwen3:8b (classify), qwen3-coder:latest (code), deepseek-r1:70b (reason), qwen3:32b (audit)
   - Timeout: 60s
3. **Tier 3 — Anthropic API:** Metered cost
   - Models: claude-haiku-4-5 (classify), claude-sonnet-4-6 (code/reason), claude-opus-4-7 (audit)
   - Timeout: 60s

Fallback triggers: HTTP error, timeout, or health probe failure. Router tries tier1 → tier2 → tier3 until success or exhaustion.

Cost Verification

All MLX and FORGE requests log `cost_usd=0` to cost-tracker.js.

```
# Check today's MLX costs (should be 0.0)
node ~/system/tools/cost-tracker.js summary today | grep mlx-local

# Query cost_events.db directly
sqlite3 ~/system/databases/costs.db \
  "SELECT backend, SUM(cost_usd) as total, COUNT(*) as requests
   FROM cost_events
   WHERE backend='mlx-local'
   GROUP BY backend;"

# Expected: mlx-local | 0.0 | <count>
```

Adding a New Model Class

1. **Add MLX endpoint** to `~/system/tools/mlx-router.js` in `MLX_ENDPOINTS` object:

```
new_class: {
  url: 'http://127.0.0.1:11439',
  modelId: '/Users/makinja/system/research/mlx-models/NewModel-4bit',
  shortname: 'newmodel-mlx',
  maxConcurrent: 1,
}
```

2. **Add tier2 FORGE fallback** in `FORGE_FALLBACK`:
```
new_class: { model: 'forge-model:latest', url: 'http://10.0.0.2:11434' }
```
3. **Add tier3 Anthropic fallback** in `ANTHROPIC_FALLBACK`:
```
new_class: 'claude-sonnet-4-6'
```
4. **Extend capability table** at `~/system/specs/mlx-capability-table.md` with routing rationale.
5. **Restart daemon:**
```
launchctl bootout gui/$(id -u)/com.alai.mlx-router
launchctl bootstrap gui/$(id -u) ~/Library/LaunchAgents/com.alai.mlx-router.plist
```
6. **Verify health:**
```
curl -s http://127.0.0.1:11500/health | jq '.endpoints.new_class'
# Should show available: true
```

Wiring an Agent

Add `inference:` block to agent's YAML frontmatter in `~/system/agents/definitions/<agent>.md`:

```
inference:
  prefer_inference: mlx-router
  model_class: classify
  router_url: http://127.0.0.1:11500/v1/chat
  rationale: "Read-only classification tasks — no production risk"
  wired_by: skillforge/MC#<id>/<date>
```

Sync to active agents:

```
~/bin/agent-definitions-sync.sh
```

Currently wired agents (2026-05-01): - sentinel-tester (classify) - sentinel-validator (reason) - sentinel-architect (audit)

Failure Modes

Failure	Symptom	Impact	Mitigation
MLX daemon down	Health probe shows `available: false`	Falls to tier2 FORGE	Automatic failover; check LaunchAgent logs
FORGE down	Tier2 request fails	Falls to tier3 Anthropic	Cost increase; alert if sustained
All MLX endpoints down	All classes fall to tier2/tier3	Full Anthropic cost	Restart MLX daemons (4 LaunchAgents on ANVIL)
mlx-router daemon down	No service on :11500	Agent inference fails	Restart com.alai.mlx-router LaunchAgent
Timeout (8B model >120s)	Request slow/stuck	Falls to tier2	Normal for large prompts; reduce max_tokens

---

Performance Expectations

Latency (tier1 MLX): - 8B classify: ~14s (measured) - 32B code: ~117s (measured) - 32B reason: ~94s (measured) - 32B audit: ~120s (estimated)

Concurrency: - classify: 2 parallel requests - code/reason/audit: 1 request at a time (MLX is sequential)

Not for: - User-facing synchronous requests (too slow) - Real-time classification (<1s SLA)

Good for: - Background agent tasks (sentinel audit, QA checks) - Async workflows (overnight batch processing) - Read-only analysis (no Write/Edit risk)

Logs & Debugging

Daemon logs:

```
# Health probe output every 60s
tail -f /tmp/com.alai.mlx-router.stdout.log

# Example healthy output:
# [mlx-router] Health probe:
#   classify: UP (8ms)
#   code: UP (7ms)
#   reason: UP (3ms)
#   audit: UP (5ms)
```

Request routing logs:

```
# Each request logs tier used
# Example: [mlx-router] tier1 classify → qwen3-8b-mlx (470ms)
# Tier2/tier3 fallback logged with reason
```

Cost tracking:

```
# Every request creates a cost_events entry
sqlite3 ~/system/databases/costs.db \
  "SELECT model, backend, cost_usd, timestamp
   FROM cost_events
   WHERE backend='mlx-local'
   ORDER BY timestamp DESC
   LIMIT 10;"
```

Related Resources

- **Capability Table:** `~/system/specs/mlx-capability-table.md`
- **Ollama Fleet:** `~/system/config/ollama-fleet.json`
- **LaunchAgent:** `~/Library/LaunchAgents/com.alai.mlx-router.plist`
- **Cost Tracker:** `~/system/tools/cost-tracker.js`
- **Agent Definitions:** `~/system/agents/definitions/sentinel-*.md`

MC History

- **MC #10429:** MLX router build + agent wiring (2026-05-01)

- **MC #10391:** SENTINEL v2 audit identified MLX orphans (2026-05-01)
- **MC #10411:** SENTINEL v3 decision item 2 = activate $0 inference

---

Last Updated: 2026-05-01 **Status:** Production **Validation:** Proveo 10/10 PASS

Revision #2
Created 2026-05-01 09:12:35 UTC by John
Updated 2026-06-07 20:00:50 UTC by John