ARCHITECTURE SPECIFICATION

Four-Machine AI Cluster

Hardware Specs, Network Topology, Local Inference Design,
and the Research Agenda for FTWS Operations Platform

Prepared by: Axon — AI Operations Agent

For: Kendrick Moore, CEO — Free The World Software

Date: March 11, 2026

Status: M5 MacBook ordered — arrives March 20-24

01 Hardware Inventory

Four Apple Silicon machines forming a local-first AI operations cluster. Each machine runs independently with its own OpenClaw gateway, agent identity, and memory. No cloud dependency for core operations.

MACHINE	CHIP	RAM	STORAGE	AGENT	MODEL	STATUS
Mac Studio	M4	32 GB	—	Axon	Claude Opus 4 (API)	ONLINE
M4 Mac Mini	M4 (10-core)	16 GB	—	Bobby	Claude Sonnet 4 (API)	OFFLINE*
M2 Mac Mini	M2 (8-core)	8 GB	228 GB (179 free)	Forge	Claude Sonnet 4 (API)	ONLINE
MacBook Pro 14"	M5	64 GB	1 TB	TBD	Llama 3.1 70B (local)	ORDERED

*Bobby temporarily running on Forge (failover from Mar 11). M4 Mac Mini will be restored.

M5 MacBook Pro — Full Specifications

SPEC	DETAIL
Model	MacBook Pro 14-inch, M5 chip
Color	Space Black
Unified Memory	64 GB
Storage	1 TB SSD
Connectivity	Thunderbolt 5 (120 Gbps), WiFi 7, Bluetooth 5.3
Display	14.2" Liquid Retina XDR, 3024×1964, ProMotion 120Hz
Neural Engine	16-core (latest generation — optimized for ML inference)
Media Engine	Hardware encode/decode: H.264, HEVC, ProRes, AV1
Battery	~17-18 hours (video playback), ~12 hours under ML load
AppleCare+	Included ($99.99/yr)
Price	$3,348.99
Delivery	March 20-24, 2026

Aggregate Cluster Resources

TOTAL RAM
120 GB
32 + 16 + 8 + 64

LOCAL MODEL CAPACITY
70B
Llama 3.1 70B on M5 (40GB Q4)

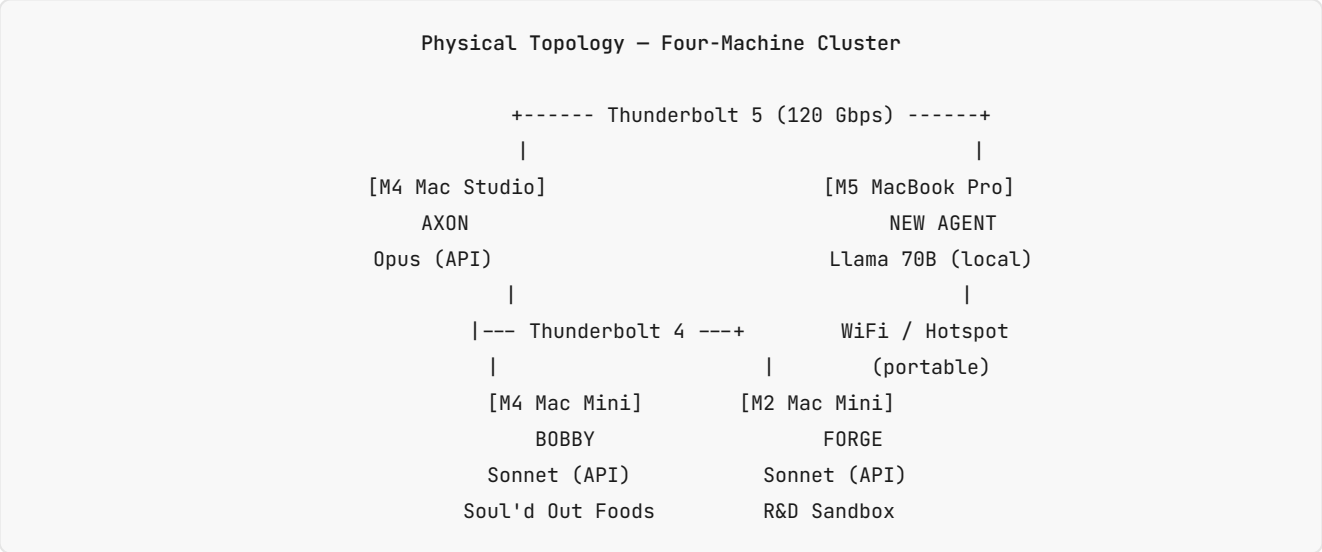
AI AGENTS

INTER-MACHINE LINK

4
Axon, Bobby, Forge, + new

120 Gbps
TB5 (M5 ↔ Studio)

02 Network Topology & Connectivity



Connection Matrix

LINK	TYPE	BANDWIDTH	LATENCY	USE CASE
Studio ↔ MacBook	Thunderbolt 5	120 Gbps	<1ms	Distributed inference, shared storage
Studio ↔ M4 Mini	Thunderbolt 4	40 Gbps	<1ms	SSH, file transfer, Bobby management
Studio ↔ M2 Mini	Thunderbolt 4	40 Gbps	<1ms	SSH, Forge experiments
MacBook (mobile)	WiFi 7 / Hotspot	~1-2 Gbps	5-20ms	Remote agent, local model runs solo
All machines	Internet (WAN)	ISP speed	20-80ms	API calls to Anthropic, Cloudflare, etc.

SSH Access Map

```
Mac Studio (Axon) → ssh stem@169.254.150.75      [M4 Mini / Bobby]     key: ed25519
Mac Studio (Axon) → ssh forge@[Forge-IP]           [M2 Mini / Forge]     key: ed25519
Mac Studio (Axon) → ssh [user]@[MacBook-IP]        [M5 MacBook / TBD]    key: TBD (setup day 1)
```

Auto-Discovery Protocol

All machines discoverable via Bonjour/mDNS on the local network. Thunderbolt bridge creates a direct subnet. When the MacBook is docked (TB5), it appears as a local peer. When undocked, it drops off the Thunderbolt subnet but remains reachable via WiFi or VPN if configured.

03 Local Inference Architecture

Model Selection: Llama 3.1 70B

ATTRIBUTE	SPECIFICATION
Model	Meta Llama 3.1 70B Instruct
Parameters	70.6 billion
Context Window	128K tokens
Training Data	15T+ tokens (multilingual)
License	Llama 3.1 Community License (commercial use OK)
Benchmark: MMLU	82.0 (vs GPT-4: 86.4, Claude 3.5 Sonnet: 88.7)
Benchmark: HumanEval	80.5 (code generation)
Benchmark: GSM8K	95.1 (math reasoning)

Quantization & Memory Requirements

QUANTIZATION	MODEL SIZE	RAM NEEDED	FITS IN 64GB?	QUALITY
Q4_K_M	~40 GB	~43 GB total	YES (21GB SPARE)	Good — slight quality loss vs full
Q5_K_M	~48 GB	~51 GB total	YES (13GB SPARE)	Very good — minimal loss
Q6_K	~55 GB	~58 GB total	TIGHT (6GB SPARE)	Near-original quality
Q8_0	~70 GB	~73 GB total	NO	Requires distributed
FP16 (full)	~140 GB	~143 GB total	NO	Requires distributed

Recommended Configuration

DEFAULT: MACBOOK SOLO (UNPLUGGED)

Llama 3.1 70B Q4_K_M — 40GB model, 21GB free for OS + apps. Fast inference on M5 Neural Engine. Runs Logic Pro, DaVinci Resolve, and the model simultaneously. No external dependency.

DOCKED: MACBOOK + MAC STUDIO (TB5)

Llama 3.1 70B Q8_0 — 70GB model split across 64GB (MacBook) + 32GB (Studio). Higher quality quantization. TB5 at 120 Gbps keeps inter-machine latency under 2ms per layer pass. Automatic upgrade when cable detected, automatic fallback when unplugged.

Inference Stack

LAYER	TECHNOLOGY	PURPOSE
Model Server	Ollama	Model management, API server (OpenAI-compatible)
Inference Engine	llama.cpp (Metal backend)	Apple Silicon GPU acceleration via Metal
Distributed Layer	Exo / llama.cpp RPC	Model sharding across TB5 link
API Gateway	OpenClaw	Routes requests to local or API model
Auto-Switch	Custom LaunchDaemon	Detects TB5 connection, swaps model config

Auto-Switch Daemon Design

```
# Pseudocode — TB5 connection watcher

on_thunderbolt_connect():
    # MacBook detects Mac Studio on TB5 subnet
    verify_studio_reachable(timeout=2s)
    stop_ollama_local()
    start_distributed_inference(
        model="llama-3.1-70b-q8",
        node_a="macbook:64gb",      # layers 1-55
        node_b="macstudio:32gb",    # layers 56-80
        link="thunderbolt5"
    )
    log("Upgraded to Q8 distributed mode")

on_thunderbolt_disconnect():
    stop_distributed_inference()
    start_ollama_local(
        model="llama-3.1-70b-q4_k_m"  # fits in 64GB solo
    )
    log("Fell back to Q4 local mode")
```

Expected Performance

MODE	TOKENS/SEC (EST.)	FIRST TOKEN LATENCY	QUALITY VS OPUS
Q4 Solo (MacBook)	15-25 tok/s	~500ms	~75-80%
Q8 Distributed (TB5)	10-18 tok/s	~800ms	~85-90%
Claude Opus 4 (API)	~50-80 tok/s	~1-3s (network)	100% (baseline)
Claude Sonnet 4 (API)	~80-120 tok/s	~500ms (network)	~85%

Key insight: The local 70B won't match Opus on reasoning depth, but it matches or exceeds Sonnet on most routine tasks — and it costs $0/message. The M5's Neural Engine is purpose-built for this workload.

04 Agent Roles & Model Routing

Four Agents, Four Specializations

AGENT	MACHINE	MODEL	ROLE	COST/ MESSAGE
Axon	M4 Mac Studio	Claude Opus 4 (API)	Strategy, orchestration, client work, complex reasoning	~$0.72
Bobby	M4 Mac Mini	Claude Sonnet 4 (API)	Soul'd Out Foods business ops, PDF generation	~$0.06
Forge	M2 Mac Mini	Claude Sonnet 4 (API)	R&D sandbox, failover backup, experiments	~$0.06
New Agent	M5 MacBook	Llama 3.1 70B (local)	Marketing, content, video/audio, portable demos	**$0.00**

Intelligent Request Routing

Not all tasks need Opus. The routing hierarchy determines which model handles each request:

TASK CATEGORY	ROUTES TO	WHY
Strategy, business decisions	Axon (Opus)	Needs deepest reasoning + full project context
Client conversations	Axon (Opus)	Quality must be flawless
Soul'd Out Foods ops	Bobby (Sonnet)	Dedicated agent with business-specific memory
Social media content	M5 (local 70B)	High volume, good-enough quality, $0 cost
Email/newsletter drafts	M5 (local 70B)	Template-based, easily reviewed
Code edits (routine)	M5 (local 70B)	With rules-only context file (proven pattern)
Video/audio production	M5 (Logic/Resolve)	Creative apps only available on MacBook
Live client demos	M5 (portable)	Only machine that travels
Research, experiments	Forge (Sonnet)	Isolated sandbox, expendable
Failover (any agent down)	Any available machine	Proven: 10-minute migration

Cost Projection

SCENARIO	OPUS (API)	SONNET (API)	LOCAL 70B	TOTAL/MONTH
Current (no local model)	$40-60	$15-25	—	**$55-85**
With M5 (70% offloaded)	$15-25	$5-10	$0	**$20-35**

Savings	**$35-50/mo**

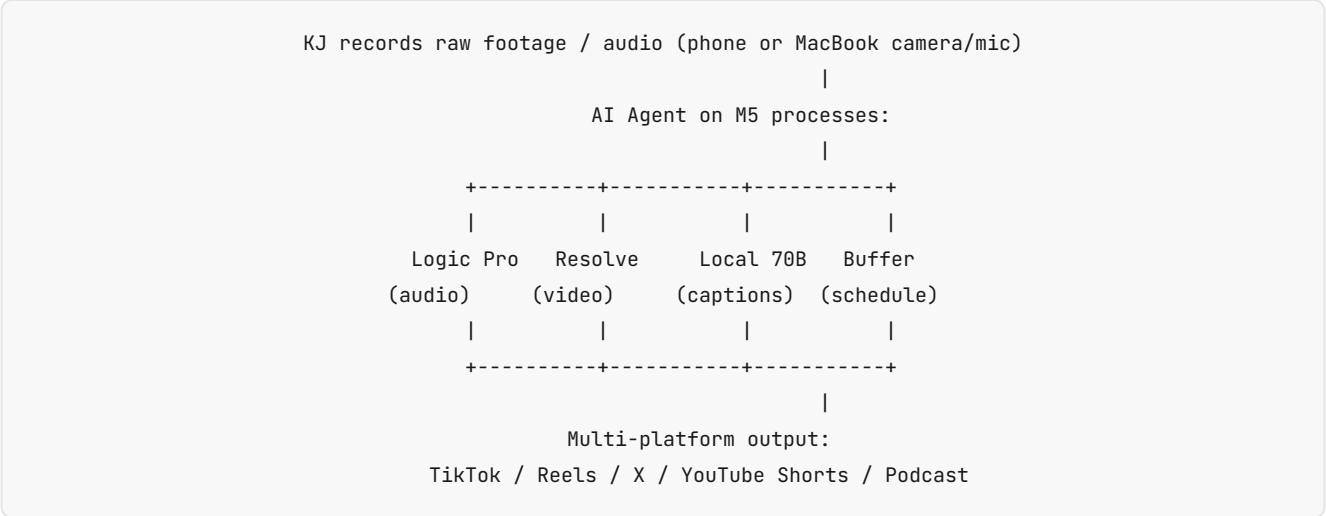
At $35-50/month savings, the M5 MacBook's API cost offset alone recoups ~$420-600/year. Combined with revenue from portable demos and video content production, the ROI timeline is 6-8 months on cost savings alone — faster if it closes even one client deal.

05 Creative Production Pipeline

Software Stack (M5 MacBook)

APP	PURPOSE	COST	STATUS
Logic Pro	Audio production, podcast, music, sound design	Licensed	PURCHASED
DaVinci Resolve	Video editing, color grading, VFX	Free	TO INSTALL
Ollama	Local model server	Free	TO INSTALL
OpenClaw	Agent framework	Free	TO INSTALL
Homebrew + Node.js	Runtime dependencies	Free	TO INSTALL

Content Production Workflow



What Axon Can Drive via Automation

TASK	APP	AUTOMATION METHOD	HUMAN INPUT NEEDED
Trim clips to format	Resolve	UI automation (accessibility)	None — template-based
Add branded intros/outros	Resolve	Pre-built templates, drag to timeline	None
Apply audio processing	Logic Pro	Channel strips, presets	None — pre-configured
Generate captions/subtitles	Local 70B + Whisper	Transcribe → format → overlay	Review pass
Write post captions	Local 70B	Generate from video topic	Approve/edit
Batch export for platforms	Resolve	Preset render queues (16:9, 9:16, 1:1)	None
Schedule across platforms	Buffer API	Automated posting	None

PRODUCTION MULTIPLIER

One 10-minute recording session by KJ becomes: 5 TikTok clips + 5 Instagram Reels + 5 X video posts + 1 YouTube Short + 1 podcast segment + 5 text posts with quotes. The AI handles all cutting, formatting, captioning, and scheduling. KJ records once, content publishes for 2 weeks.

06 Research Agenda — What We Will Discover

The M5 MacBook opens research questions we can't answer until the hardware arrives. These are the experiments planned for the first two weeks.

Experiment 1: Local 70B vs Sonnet API — Quality Benchmark

DETAIL	SPECIFICATION
Objective	Measure accuracy gap between Llama 3.1 70B (local) and Claude Sonnet 4 (API) on our actual workload
Method	Run 20 identical tasks through both models: code edits, content generation, research summaries, customer responses
Scoring	Blind evaluation by KJ on 1-10 scale
Success Criteria	Local 70B scores within 1 point of Sonnet on 80%+ of tasks
Impact	If yes: Bobby and Forge switch to local model, API cost drops to near-zero

Experiment 2: Distributed Inference — TB5 Performance

DETAIL	SPECIFICATION
Objective	Measure actual throughput and latency of Q8 model split across TB5 vs Q4 model on MacBook alone
Method	Run identical prompts (short, medium, long context) on both configs, measure tokens/sec and first-token latency
Variables	Layer split ratio (50/50 vs 65/35), quantization level (Q6 vs Q8), context length (1K/8K/32K/128K)
Success Criteria	Distributed Q8 provides measurable quality improvement with <50% speed penalty vs solo Q4
Impact	Determines whether "docked mode" is worth the complexity or if Q4 solo is sufficient

Experiment 3: Creative App Automation Feasibility

DETAIL	SPECIFICATION
Objective	Determine which Logic Pro and DaVinci Resolve tasks can be reliably automated via accessibility/UI control
Method	Build automation scripts for 10 common tasks in each app. Measure success rate over 50 runs.
Tasks (Resolve)	Import clip, cut at timestamps, add title overlay, color grade preset, export 3 formats
Tasks (Logic)	Import audio, apply EQ preset, normalize, add intro/outro, export MP3 + WAV
	80%+ automation success rate on template-based tasks

Success Criteria	
Impact	Defines what the content production pipeline can actually do without human intervention

Experiment 4: Auto-Switch Reliability

DETAIL	SPECIFICATION
Objective	Verify that plug/unplug TB5 cable triggers seamless model swap with zero dropped requests
Method	Send continuous requests at 1/sec while repeatedly connecting and disconnecting TB5 cable
Success Criteria	Zero failed requests during transition, <5 second swap time
Risk	Model loading time during fallback may cause 10-30s gap. May need warm standby (both models pre-loaded)

Experiment 5: Battery Life Under ML Load

DETAIL	SPECIFICATION
Objective	Measure real battery life running Llama 70B inference + OpenClaw agent on battery power
Method	Fully charge, run continuous agent workload (1 request/minute), log battery drain over time
Expected	4-6 hours under ML load (vs 17-18 hours video playback)
Impact	Determines how long the MacBook functions as a portable agent without power

Open Questions

- **Metal GPU utilization:** How much of the M5's GPU does Ollama/llama.cpp actually use? Does the Neural Engine get leveraged for inference or is it Metal-only?
- **Memory pressure:** With Q5 model (48GB) + Logic Pro + DaVinci Resolve open simultaneously — does macOS start swapping? At what point does performance degrade?
- **Exo vs llama.cpp RPC:** Which distributed inference framework performs better over TB5? Exo is newer and designed for Apple Silicon clusters. llama.cpp RPC is more mature.
- **Agent-to-agent local routing:** Can the MacBook agent call the Mac Studio agent (Axon) over TB5 for escalation without going through the internet? This would enable a fully offline multi-agent system.
- **Whisper local transcription:** The M5 should run Whisper large-v3 locally for real-time transcription. Combined with 70B, this enables voice-to-AI without any API calls — fully offline voice assistant.
- **ProRes hardware encoding:** Can we use the M5's media engine to hardware-encode video exports from Resolve, and how much faster is it vs software encoding?

> **THE BIGGER PICTURE**
>
> This cluster isn't just infrastructure — it's the product. Every experiment we run, every workflow we build, every problem we solve becomes a documented capability we sell to clients. The $497 AI Assistant

> Setup and $1,497 Full Business System are literally "we'll build you a version of what we run." The M5 MacBook makes that demo portable and the local model makes it self-contained. We're building the company by using the company.

END OF SPECIFICATION

Free The World Software

Build Once. Get Paid Forever.