PC Fleet: Final Target State & Transition Plan

This document outlines the final technical configuration for the unified PC fleet and the step-by-step transition plan to move from the current state to the target architecture.

Part 1: Final Target State Specifications

1. Main Workstation ("Amsterdam")

Role: Primary Productivity & General Computing
Focus: Silence, power efficiency, and stability for non-AI workloads.

Component	Final Specification
GPU	NVIDIA GT 1030 / 1040 (Silent/Low Profile)
CPU	Intel Core i5-8400 (6 Core / 6 Thread)
RAM	32GB DDR4
Storage	2TB NVMe (Boot) + 1TB SATA + 6TB SATA (Archive)
PSU	300W OEM PSU (Optimized for low-draw efficiency)
OS	Windows 11

2. Ollama Server (The Mid-Tier Node)

Role: 24/7 Local LLM Hosting
Focus: Maximizing VRAM compatibility and efficient model serving.

Component	Final Specification
GPU	NVIDIA RTX 3060 (12GB GDDR6)
CPU	AMD Ryzen 5 5500
RAM	16GB DDR4
Storage	512GB NVMe
PSU	Seasonic 550W (80+ Gold)
OS	Windows 11 Pro

3. "ImageBeast" (The AI Flagship)

Role: Primary ComfyUI & Heavy Generative AI Node
Focus: Massive VRAM headroom for Blackwell-era models and dev tools.

Component	Final Specification
GPU	MSI RTX 5090 32GB Vanguard SOC (Blackwell

	Architecture)
CPU	AMD Ryzen 7 9800X3D (8C/16T, 96MB L3 Cache)
Motherboard	MSI X870E-P PRO WIFI (AM5, PCIe 5.0)
RAM	32GB TeamGroup T-Force Delta DDR5-5600 (CL40)
PSU	Corsair RM1200x Shift (1200W, Fully Modular)
Case Req.	Minimum 357mm GPU clearance; 4-slot width support

Part 2: Transition Plan

Phase 1: Preparation (Amsterdam)

- **ComfyUI Backup:** Zip the portable folder or backup models/, custom_nodes/, and user/.
- **Ollama Models:** Copy %USERPROFILE%\.ollama\models to an external drive to save bandwidth during the new build.
- **Open WebUI:** Backup the webui.db to preserve chat history and user accounts.

Phase 2: The Mid-Tier Node (Daily Driver AI)

- **Deployment:** Install the RTX 3060 (12GB) and configure as the always-on assistant.
- **Connection:** Host Open WebUI here on chat.yourdomain.com.
- **Ollama Config:** Set environment variables to allow remote access:
  - OLLAMA_ORIGINS="*"
  - OLLAMA_HOST=0.0.0.0

Phase 3: ImageBeast (Flagship & Dev Lab)

- **Deployment:** Primary heavy generative node and local "Copilot" replacement.
- **Ollama Model Stack:**
  - **Heavy Brain:** ollama pull qwen2.5-coder:32b (for complex refactoring/logic).
  - **Fast Brain:** ollama pull qwen2.5-coder:7b-base (for autocomplete).
- **VS Code Integration (Continue.dev):**
  1. Install **Continue** extension.
  2. In config.json, point apiBase to the ImageBeast IP.
  3. Configure tabAutocompleteModel to use the 7B-base and the models list to include the 32B-instruct.

Phase 4: Amsterdam Cleanup

- **Hardware Swap:** Physically install the GT 1030/1040 and the original 300W PSU.
- **Service Maintenance:** Retain the Cloudflare Tunnel exclusively for existing Flask applications and legacy web services.

Part 3: Troubleshooting & Fine-Tuning

- **Autocomplete Lag:** If suggestions take more than 200ms, switch the autocomplete model to qwen2.5-coder:1.5b-base.
- **Context Window:** For extensive refactoring, increase num_ctx in the Continue config to 16384 or higher (The 5090 has ample VRAM for this).
- **Thermal Monitoring:** Monitor the RTX 5090 under load using MSI Afterburner, ensuring the 4-slot cooler has adequate intake clearance in the selected case.