# **JAMAICA**

**Population:** 2.8 million

Approximately **25%** of the population is privately insured under a variety of schemes. Around 5% of this is covered by the National Insurance Scheme of Jamaica

### **Private Insurance Trends**:

Growing Market

Private Insurance provides comprehensive coverage.

Covers pre-existing conditions and chronic illnesses.

Also wider range of products and services including wellness programs etc.

High healthcare costs and adoption of digital(online policy management, tele-medicne,mobile apps for claims)

Economic Growth and increases disposable incomes.

GDP↑↑ , GDP per capita↑↑, Purchasing Power Parity↑↑

### **Some General statistics:**

Public health care cost is much lesser than Private Healthcare but differential increase is more in recent years.

Bed Occupancy rate - 75%(approx)

Curative Visits - 1 million(approx)

Statistically richest quintile swear by private insurance services

Lower 40% is completely dependent on the public healthcare system

Mean Public visit spend 750USD, Mean Private 2500USD

Unemployment rate↓↓, Government debt↓↓, Population↑↑

|  |  |  |
| --- | --- | --- |
| **Category** | **Public Health Insurance** | **Private Health Insurance** |
| **Cost** | Free at point of service | Premiums and out-of-pocket expenses |
| **Access** | Universal (for citizens and legal residents) | Limited, typically to those who can afford it |
| **Wait times** | Longer wait times | Shorter wait times |
| **Quality of care** | Basic care, can be limited | Potentially better facilities and equipment |
| **Specialized care** | Limited availability | Wider range of specialties available |
| **Coverage** | Basic healthcare | Varies, can be comprehensive |

**\*\*NHF Context\*\* \*\*Should be removed for Disparate need\*\***

**‘Drug Serv programme’ by National Health Fund:** **720,000** Jamaicans receive medication for free, seek pharmaceutical care in the public sector annually, where medication provided on the Vital, Essential & Necessary (VEN) list is free.

Jamaicans with the NHF card automatically receive **Jamaica Drugs for the Elderly Programme (JADEP)** benefits upon reaching 60 years of age. The JADEP programme provides subsidies for 10 of the 17 conditions covered by the NHF card.

JADEP drug items are available at a minimal cost to the beneficiary – only $40 per item is paid to the pharmacy, and a maximum of $240 for six or more items.

Not all drugs prescribed are covered by the NHF card. The condition diagnosed may not be one of the 17 illnesses**.**

**[Your Company Name]**[Your Address]  
[City, State, ZIP Code]  
[Phone Number] | [Email] | [Website]

**Date:** [Insert Date]  
**Proposal Prepared for:** [Client Name]  
**Company:** [Client Company Name]  
**Project Title:** [Brief Project Name]

## **Executive Summary:**

NHIP implementation and its requirement to be considered.(if in case)

It is a risk-pooling mechanism offering financial protection against the unpredictable burden of illness so that those needing care can have ready access to services.

It is a payment mechanism offering providers of health services prompt and regular reimbursements.

It is a database providing planners and policymakers with valuable information on patterns of disease, utilization of services, health expenditure and patient preferences (choices).

Mandatory vs Compulsory

Public/Private partner coverage

Micro-insurance schemes

## Phase one: Discovery(One month)

The inception begins with discovery phase in which all viable data sources are identified, documented and ingested into the data pipeline.Next this data is processed for training while retaining only context rich information. Both digital and non-digital data(NDD) are scraped. NDD data to be scanned by VOX, VOX partners or outsource the task to third party service providers who will complete the task at a nominal rate. Provided below are a list of document types to be collected, collated and prepared for the Data ingestion layer.

### List of documents

#### Structured data:

**1. Policy Details:** This includes the health insurance policy document itself, outlining coverage, benefits, and terms and conditions.

**2. Identity Proof:** This could be a valid passport, driver's license, or other forms of identification.

**3. Medical Reports:** These are documents related to medical conditions, treatments, or procedures, such as lab results, X-ray reports, or doctor's notes.

**4. Original Claim Form:** This is the document used to formally request reimbursement or payment for medical expenses.

**5. Doctor's Prescription:** This document from a medical professional outlines the necessary medication or treatment.

**6. Other Potential Documents:** Depending on the specific insurance provider and the situation, other documents might be needed, such as:

* **Indoor case papers:** Documents related to hospital stays.
* **Ambulance receipts:** Proof of ambulance transportation.
* **Original pharmacy bills:** Receipts for purchased medications.
* **First Information Report (FIR):** May be required in cases involving injuries or accidents.
* **Proof of income or employment:** May be required for group health insurance or some individual plans.
* **Pension Forms**

**7. Internal database** data within the organization. A robust structural analysis along with access to pertinent data is to be provided. Possibly LDAP implementation to manage user permission and access privileges.

**8. Lab Reports** or Test results for said patient. Diagnostic imagery or scans may not be considered.

**9. Additional Considerations for Expats:** Expats in Jamaica may need a global or international health insurance plan that covers them in Jamaica and potentially other locations if they need to travel..

#### Unstructured/Semi-structured Data:

* All Emails between patient and Insurance provider and/or Healthcare provider
* Pdf’s, notes or hand written documents of data within the organization.
* Non digital data(NDD) crucial to build context for the Gen-Ai system.
* Inter agency documents to be scraped. Digital or otherwise.
* List of illnesses that have no coverage have to be categorically mentioned to filter requests and speed up the process of claims.

## Phase Two: Data Ingestion Layer/Training(2-3months):

After phase one the system in ready to ingest all the curated data. In phase two we begin the training process where massive amounts of labeled and unlabeled data are fed to our Gen-AI architecture. First the process of tokenization happens where text, images and audio(if any) are converted into tokens. The tokens give us an estimate on training time and hardware requirement.

## **Token Estimation: A Visual Breakdown**

This document outlines the methodology for estimating the number of tokens involved in processing various data sources, primarily for training a Large Language Model (LLM). The estimation is crucial for understanding the scale of data processing and the potential impact of image data.

### **1. Structured Data**

The following table provides a detailed breakdown of the token estimation for structured data sources:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document Type** | **Estimated Size** | **Words per Page** | **Word Relevance Multiple** | **Estimated Tokens (per person)** |
| A) Policy Details | 10 pages | 200 | 0.7 | 2,100 |
| B) Identity Proof | 2 pages | 30 | 1 | 90 |
| C) Medical Reports | 30 pages | 100 | 0.8 | 3,600 |
| D) Original Claim Form | 2 pages | 75 | 1 | 225 |
| E) Doctor’s Prescription | 1 page | 50 | 1 | 75 |
| F) Other Documents | Variable | - | - | See Calculations Below |
| G) Database History | - | - | 0.5 (Speculative) | 75,000 |
| H) Lab Reports/Scans | Digital Images | - | - | Highly Variable (200-1000x text tokens) |
| I) Expats | - | - | - | - |

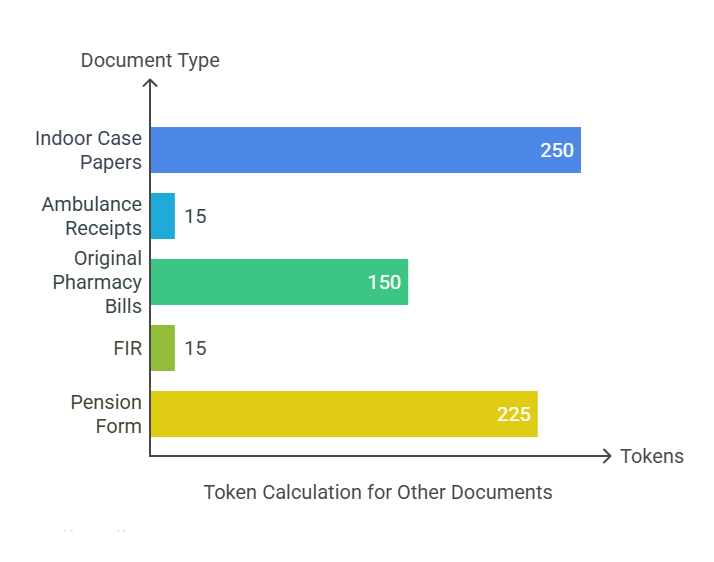
**Note:**



* All figures are mean estimates and per-person basis.
* Text vs. image tokens are not differentiated in the initial word count. Image tokens can be significantly larger (200-1000x).
* This is a preliminary estimate and is considered conservative.

**Calculations for "Other Documents" (F):**

* Indoor Case Papers: 1000 words \* 0.5 (WRM) \* 0.5 (Instance Rate) = 250 tokens
* Ambulance Receipts: 100 words \* 1 (WRM) \* 0.1 (Instance Rate) = 15 tokens
* Original Pharmacy Bills: 100 words \* 1 (WRM) \* 1 (Instance Rate) = 150 tokens
* FIR: 1000 words \* 0.2 (WRM) \* 0.05 (Instance Rate) = 15 tokens
* Pension Form: 1000 words \* 0.3 (WRM) \* 0.15 (Instance Rate) = 225 tokens



**Total Tokens for Structured Data:**

* Total Words (approx): 80,000 per person
* Words to Token Factor: 1.5
* Total Tokens per person: 80,000 \* 1.5 = 120,000
* Net populace: 0.25 \* 2.8 million = 700,000
* Total Tokens: 700,000 \* 120,000 = 84 Billion

### **2. Unstructured Data**

Due to the variability in how organizations store and handle unstructured data, a speculative, top-down approach is used:

* Total Unstructured Tokens (Estimate): 1/4 \* Total Structured Tokens = 21 Billion

### **3. Total Training Tokens Range**

* Total Training Tokens Range: 105 Billion - 262.5 Billion (including potential image data multiplier)

### **4. Deployment Strategy**

Based on this token estimation, an LLM/Agentic system has been engineered to handle the client’s requirements. A technical description of the deployment strategy is available in the following section.

### Token Estimation:

#### Structured Data

**Document pruning:** Irrelevant or impertinent information in each source should be pruned or modified to extract better context.

**\*\***All figures calculated are mean estimates and are per person basis and Text vs Image tokens are not differentiated. Image tokens can be more than 200 times the size of text tokens. This is only a preliminary estimate and is being looked at conservatively.

1. **Policy Details:** Estimated Document length: 10pages

Words per page: 200

Approximate Word Relevance multiple: 0.7

**B) Identity proof:** Estimated size: 2 pages

Words per page 30:

Word Relevance multiple: 1

**C) Medical Reports:** Estimated size: 30 pages(on avg)

Words per page: 100

Word Relevance multiple: 0.8

**D) Original Claim Form**: Estimated size: 2 pages

Words per page: 75

Word Relevance multiple: 1

**E) Doctor’s Prescription:** Estimated size: 1 page

Words per page(avg): 50

Word Relevance multiple: 1

**F) Other Documents:** These are subjective entries and may not always be relevant.

Indoor Case papers- Total Words 1000 \* 0.5(WRM)\* 0.5(Instance rate)

Ambulance Receipts- Total Words 100 \* 1(WRM) \* 0.1(Instance rate)

Original Pharmacy Bills- Total Words 100 \* 1(WRM) \* 1(IR)

FIR- Total Words 1000 \* 0.2(WRM) \* 0.05(IR)

Pension Form-Total Words 1000 \* 0.3(WRM) \* 0.15(IR)

**G) Auxiliary Database estimate:** Patient Historical Data/Claims/Medical History/Insurance Plan and payment history. This data is aggregated and an estimate is provided.

Total Words = 100,000 \* 0.5(WRM) \* 1(Speculative)

**H) Lab report/ Scans:** Digital Image tokens-

**I)** No Consideration for **Expats**.

Total Tokens Structured/person = Total Words \* 1.5(Words to token factor)

= 80,000(approx)

Total Tokens = Net populace \* 80000

= 0.25 \* 2.8million \* 80000

= 56 Billion

**\*\***Now we would like to conclude that this tokenization estimate is done w.r.t to all the information being in text format but in fact some of these will be images, this can approximately increase by a normalized factor of 200-1000x(model based). Depending on our investigation into the structure and nature of data we can conclude on the efficacy of our training strategy.

#### Unstructured Data

Considering the black box nature of organization and the way in which they store or handle different types of digital and non-digital inventory we have to use a speculative top-down approach.

Total Tokens = 1/4(estimate) \* Total Structured tokens

= 14 billion tokens

**Total Training Tokens range = 60billion - 150billion(Image Factor-in)**

Based on this estimate we have engineered an LLM(Large Language Model)/Agentic system to handle the client’s requirements. Presented below is a technical description of our deployment strategy.

LLMs - Sizing and placement

# VLMs

* Qwen/Qwen2.5-VL-32B-Instruct (License: Apache license 2.0)
* Qwen/Qwen2.5-VL-7B-Instruct (License: Apache license 2.0)
* OpenGVLab/InternVL3-8B (License: Apache license 2.0)
* openbmb/MiniCPM-V-2\_6 (8B params) (License: MiniCPM license)
* vidore/colpali-v1.3 (License: MIT)

Here Qwen/Qwen2.5-VL-7B-Instruct and OpenGVLab/InternVL3-8B will work mostly in our usecase. Qwen/Qwen2.5-VL-32B-Instruct can be used too for use cases where we need to provide high confidence from a low resolution/unclear image. Additionally, the Qwen2.5-VL-3B model can be used as an alternative to InternVL2, providing flexibility in model selection. The inclusion of InternVL2 introduces variance in output compared to Qwen, contributing to a more robust and diversified system. ColPali on the other hand makes document processing much more efficient both computation and accuracy wise.

# Code LLMs

* Qwen/Qwen2.5-Coder-32B-Instruct (License: Apache license 2.0)
* Qwen/Qwen2.5-Coder-14B-Instruct (License: Apache license 2.0)
* deepseek-ai/DeepSeek-Coder-V2-Lite-Instruct (15 B) (License: deepseek license) (MoE)
* meta-llama/CodeLlama-70b-Instruct-hf (License: Llama 2 Community License Agreement)
* bigcode/starcoder2-15b (License: bigcode-openrail-m)

Qwen models are currently the best models in terms of coding benchmarks like HumanEval and MBPP, with performances equaling o1-preview and claude3.5-sonnet. To provide variance in output we can opts for CodeLlama/StarCoder but according to benchmarks Qwen outperforms them in a great way.

# Text Generation LLMs

* Qwen/Qwen3-32B (License: Apache license 2.0)
* Qwen/Qwen3-30B-A3B (License: Apache license 2.0) (MoE)
* Qwen/Qwen3-14B (License: Apache license 2.0)
* Qwen/Qwen2.5-32B-Instruct (License: Apache license 2.0)
* deepseek-ai/DeepSeek-R1-Distill-Qwen-32B (Can use by using ollama engine) (License: MIT)
* meta-llama/Llama-3.3-70B-Instruct (License: Llama 3.3 Community License Agreement)
* Qwen/Qwen3-8B (License: Apache license 2.0)
* Qwen/Qwen3-4B (License: Apache license 2.0)

meta-llama/Llama-3.2-3B-Instruct (License: Llama 3.2 Community License Agreement)

In reasoning LLMs we can go forward with the Qwen3 series released recently. They provide both thinking (reasoning) and non-thinking (instruction tuned) models that can be used in different use cases as required. Also the distilled version of deepseek can be used with the help of ollama engine and here llama still does well in terms of benchmarks but is closely followed by Qwen 2.5 suite of models.

# Estimated Memory Footprint of Models on Apple MPS

The table below estimates GPU memory use (GiB) on macOS with MPS (Apple Silicon) for each model, based on parameter counts and known quantization scaling. Inference values are given for batch=1 and batch=4 (as “1/4 batch”). We assume FP32 uses 4 bytes/parameter, BF16 (bfloat16) 2 bytes, INT8 1 byte, and 4‑bit (GGUF) 0.5 bytes, plus ≈20% overhead for

activations/static buffers. Fine-tuning memory is roughly 4× the inference model size for FP32

(≈“Adam optimizer” rule); BF16 fine-tune is proportionally lower since weights are halved. The values are rounded off to the closest integer.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Models** | **Inference on MPS** | | | | **Fine-tuning on MPS** | | | |
| **FP 32**  **(1/ 4)** | **FP16 / BF16 (1/ 4)** | **INT 8**  **(1/ 4)** | **4 Bit GGUF (1/ 4)** | **FP 32** | **FP16**  **/ BF16** | **INT 8** | **4 Bit**  **GGUF** |
| **Qwen2.5-VL-32B** | 145 / 218 | 73 / 109 | 36 / 55 | 18 / 27 | 580 | 290 | 145 | 72 |
| **Qwen2.5-VL-7B** | 31 / 47 | 16 / 24 | 8 / 12 | 4 / 6 | 125 | 62 | 31 | 15 |
| **OpenGVLab/Intern VL3-8B** | 36 / 54 | 18 / 27 | 9 / 13 | 5 / 7 | 143 | 72 | 36 | 18 |
| **openbmb/MiniCPM**  **-V-2\_6** | 36 / 54 | 18 / 27 | 9 / 13 | 5 / 7 | 143 | 72 | 36 | 18 |
| **ColPali** | 31 / 47 | 16 / 24 | 8 / 12 | 4 / 6 | 125 | 62 | 31 | 15 |
| **Qwen/Qwen2.5- Coder-32B** | 143 / 215 | 72 / 107 | 36 / 54 | 18 / 27 | 572 | 286 | 143 | 71 |
| **Qwen/Qwen2.5- Coder-14B** | 63 / 94 | 31 / 47 | 16 / 24 | 8 / 12 | 250 | 125 | 62 | 31 |
| **deepseek- ai/DeepSeek- Coder-V2-Lite** | 72 / 107 | 36 / 54 | 18 / 27 | 9 / 13 | 286 | 143 | 71 | 35 |
| **meta- llama/CodeLlama- 70b** | 313 / 469 | 156 / 235 | 78 / 117 | 39 / 59 | 1250 | 625 | 312 | 156 |
| **bigcode/starcoder2**  **-15b** | 67 / 101 | 34 / 50 | 17 / 25 | 8 / 13 | 268 | 134 | 67 | 33 |
| **Qwen/Qwen3-32B** | 147 / 220 | 73 / 110 | 37 / 55 | 18 / 27 | 586 | 293 | 146 | 73 |
| **Qwen/Qwen3-30B- A3B** | 134 / 201 | 67 / 101 | 34 / 50 | 17 / 25 | 536 | 268 | 134 | 67 |
| **Qwen/Qwen3-14B** | 63 / 94 | 31 / 47 | 16 / 24 | 8 / 12 | 250 | 125 | 62 | 31 |
| **Qwen/Qwen2.5- 32B** | 145 / 218 | 73 / 109 | 36 / 55 | 18 / 27 | 581 | 291 | 145 | 72 |
| **deepseek-ai/DeepSeek-R1- Distill-Qwen-32B** | 143 / 215 | 72 / 107 | 36 / 54 | 18 / 27 | 572 | 286 | 143 | 71 |
| **meta-llama/Llama- 3.3-70B** | 313 / 469 | 156 / 235 | 78 / 117 | 39 / 59 | 1250 | 625 | 312 | 156 |
| **Qwen/Qwen3-8B** | 36 / 54 | 18 / 27 | 9 / 13 | 5 / 7 | 143 | 72 | 36 | 18 |
| **Qwen/Qwen3-4B** | 18 / 27 | 9 / 13 | 5 / 7 | 2 / 3 | 72 | 36 | 18 | 9 |
| **meta-llama/Llama- 3.2-3B** | 13 / 20 | 7 / 10 | 3 / 5 | 2 / 3 | 53 | 26 | 13 | 6 |

* Inference memory was estimated as (model\_size + 20%) for FP32, and proportionally scaled down for BF16 (½ FP32) and INT8/4-bit. For example, CodeLlama-70B (70B) has

260 GiB of FP32 weights, so ~313 GiB for batch=1 inference (with overhead), ~469 GiB at batch=4; BF16 uses half that (156/235 GiB)

* Training (fine-tune) memory is roughly 4× the model size in FP32, FP32 fine-tune ≈ 4× the FP32 inference (batch=1) footprint. BF16 and INT8/4-bit fine-tuning would similarly scale from the smaller BF16 weights (roughly half).

# Hardware & Deployment Estimates

To target upto 15 concurrent users, with latency caps of 180 s (simple tasks) or 300 s (complex reasoning). The cluster comprises one Mac Studio M3 Ultra (32‑core CPU, 80‑core GPU, 512 GB unified memory) and four Mac Studio M4 Max machines (each 16‑core CPU, 40‑core

GPU, 128 GB memory). The M3 Ultra handles the heaviest models (vision and routing), while the M4 Max nodes share the rest. All inference uses the on-board Apple GPUs (via Metal); frameworks like Ollama or vLLM will automatically leverage the Apple GPU.

## Model Placement by Machine

* *M3 Ultra (512 GB, 80‑core GPU)*
  + Hosts the largest vision and routing models.
  + **Vision/OCR:** Qwen2.5‑VL‑32B (BF16), Qwen2.5‑VL‑7B (BF16), InternVL3‑8B (BF16) – all in BF16 to preserve accuracy on image/OCR task.
  + **Document Processing:** ColPali (e.g. PaliGemma-3B backbone, BF16 or INT8) for document processing.
  + **Code/SQL:** Qwen2.5‑Coder‑32B (INT8) for code/SQL generation.
  + **Router Agent:** Qwen3‑32B (2× instances, INT8) and Qwen3‑30B‑A3B (1×,

INT8) – large-context models for routing/decision.

* + **Supporting LLMs:** Qwen3‑14B (2×, INT8) for heavy reasoning overflow.
* *M4 Max #1 (128 GB, 40‑core GPU)*
  + Mix of mid-sized agents
  + **Qwen3‑32B** (1×, INT8) and **Qwen3‑30B‑A3B** (1×, INT8) – additional routing/model assistance.
  + **Qwen3‑14B** (1×, 4‑bit), **Qwen3‑8B** (2×, 4‑bit), **Qwen3‑4B** (2×, 4‑bit) – parallel text processing models (summarization, QA).
  + **Llama‑3.2‑3B** (2×, 4‑bit) – small backup model for lightweight tasks or fallback.
* *M4 Max #2 (128 GB, 40‑core GPU)*
  + Similar model distribution as above for parallelism
  + **Qwen3‑32B** (1×, INT8), **Qwen3‑30B‑A3B** (1×, INT8) – additional routing/model assistance and used for reasoning tasks.
  + **Qwen3‑14B** (1×, 4‑bit), **Qwen3‑8B** (2×, 4‑bit), **Qwen3‑4B** (2×, 4‑bit) – smaller models to offload web searching, web scraping workflows.
  + **Llama‑3.2‑3B (2×, 4‑bit)** – small backup model for lightweight tasks or fallback.
* *M4 Max #3 (128 GB, 40‑core GPU)*
  + Additional mid/small models
  + **Qwen2.5‑Coder‑14B** (2×, 4‑bit GGUF) for code/SQL tasks.
  + **Qwen3‑14B** (2×, 4‑bit), **Qwen3‑8B** (2×, 4‑bit), **Llama‑3.2‑3B** (4×, 4‑bit) – small models for email summarizer agents and other small enterprise tasks.
* *M4 Max #4 (128 GB, 40‑core GPU)*
  + **Qwen2.5‑Coder‑14B** (1×, 4‑bit GGUF) for code/SQL tasks.
  + **Qwen3‑14B** (2×, 4‑bit), **Qwen3‑8B** (2×, 4‑bit), **Qwen3‑4B** (4×, 4‑bit) – Some more small models for concurrency or fallbacks.

This placement ensures the largest BF16 models (vision, router) reside on the M3 Ultra with ample memory, while smaller quantized models spread across the M4s to maximize parallel throughput. We have, for example, 4 total instances of Qwen3‑32B (2 on M3, 1 on M4‑1, 1 on

M4‑2) and up to 8 instances each of Qwen3‑8B/Qwen3‑4B/Llama-3B distributed to serve

simultaneous queries.

## Precision and Quantization

The two largest LLMs (Qwen3‑32B and Qwen3‑30B‑A3B, and Qwen2.5‑Coder‑32B) are run in 8‑bit (INT8) for higher fidelity (important for the router agent), all vision models run in BF16 (no 4-bit quantization) to preserve accuracy on image/OCR tasks, while most 14B/8B/4B models and Llama‑3B use 4‑bit GGUF.

* **BF16:** Qwen2.5‑VL‑32B, Qwen2.5‑VL‑7B, InternVL3‑8B, ColPali (optional INT8), [fine- tuning also on M3].
* **INT8:** Qwen3‑32B, Qwen3‑30B‑A3B, Qwen2.5‑Coder‑32B.
* **4‑bit (GGUF):** Qwen2.5‑Coder‑14B, Qwen3‑14B, Qwen3‑8B, Qwen3‑4B, Llama‑3.2‑3B, and any other smaller models.

Memory estimates confirm feasibility:

* **M3 Ultra:** ~274 GB (e.g., 73 GB for Qwen2.5-VL-32B, 73 GB for two Qwen3-32B and other models), well within 512 GB.
* **M4 Max #1**: ~98 GB (e.g., 37 GB for Qwen3-32B, 16 GB for two Qwen2.5-Coder-14B

and other models), fits 128 GB.

Similarly for other M4 Max the memory requirements fit well within 128 GB.

## Routing Strategy

The router agent uses **Qwen3-32B** (4 instances total: 2 on M3 Ultra, 1 on M4 Max #1, 1 on M4 Max #2) and **Qwen3-30B-A3B** (3 instances: 1 on M3 Ultra, 1 on M4 Max #1, 1 on M4 Max #2) as the main thinking models. These:

* Interpret user queries with a 128K-token context.
* Dispatch tasks to specialized agents (vision, text, code).
* Run in INT8 for efficiency, with the MoE variant (Qwen3-30B-A3B) reducing active parameters (~3B during inference).

Multiple instances ensure 3–5 parallel routing queries, with queuing for additional requests within the 300-second latency budget.

## Fine-Tuning Considerations

Fine-tuning occurs on the M3 Ultra, leveraging its 512 GB RAM and 80-core GPU. This is ideal for:

* Large BF16 models (e.g., Qwen2.5-VL-32B).
* Offline updates to weights, which are then deployed to appropriate machines. Fine- tuning can use PyTorch/MPS or Hugging Face Transformers, with updated models served via Ollama/vLLM

## Model Serving Recommendations

The serving stack integrates with LangChain/LangGraph:

* **Ollama:** Local server for quantized models (GGUF, INT8), auto-uses Apple GPUs via Metal. LangChain’s OllamaLLM, ChatOllama wrapper connects seamlessly.
* **vLLM:** High-throughput option for large models (e.g., Qwen3-32B), supports BF16/INT8

and multi-threading on Apple Silicon.

* **Hugging Face:** For custom models or fine-tuning, using PyTorch/MPS.

Each machine runs inference servers (e.g., ollama serve), with LangGraph orchestrating agent calls across machines. This minimizes overhead and leverages local GPU resources.

### Output Analysis & Experiment Management

The stack supports robust tools for analyzing model outputs and managing iterative improvements:

* **SurfSense:** Developed by the Allen Institute, SurfSense ingests JSONL-formatted prompt-response logs for structured evaluation. It enables filtering, annotation, and feedback tagging—ideal for debugging, dataset curation for fine-tuning, and tracking performance across iterations. Integration is straightforward: export RAG or QA logs in JSONL format and load into the SurfSense UI.
* **Open Notebook (lfnovo):** A lightweight, browser-based notebook environment designed for collaborative LLM experimentation. Supports prompt tracking, output comparisons, and documentation of agent behaviors. Integration involves converting output logs to a compatible JSON schema. Easily deployed locally or via Docker, making it suitable for experiment versioning and team-based workflows.

## Agent Concurrency and Performance

* **Router Agent:** Is first in each query path – it interprets the user’s request and dispatches to specialized agents. 4–7 parallel queries (across Qwen3-32B/30B-A3B instances).
* **Vision Agent:** Parallel image processing on M3 Ultra (independent tasks).
* **Code Agent:** Database query generation and running it on SQL/Oracle/Enterprise databases.
* **Text Agent:** Multiple instances (e.g., 6 Qwen3-8B, 6 Qwen3-4B) on M4 Max machines

for stateless tasks like document QA, email QA or summarization.

**Latency:** 180 (non thinking) – 300 (thinking) seconds accommodates multi-step workflows (e.g., routing → vision → text). **GPU utilization** is manageable, as inference is less intensive than training, and Apple GPUs handle high throughput.

## Bottlenecks and Scalability

*Bottlenecks:*

* **Router:** With only a few instances, heavy traffic could queue up. Can be mitigated by multiple instances and MoE variant (Qwen3-30B-A3B).
* **Vision:** Large BF16 models on M3 Ultra; offload smaller vision tasks to M4 Max if

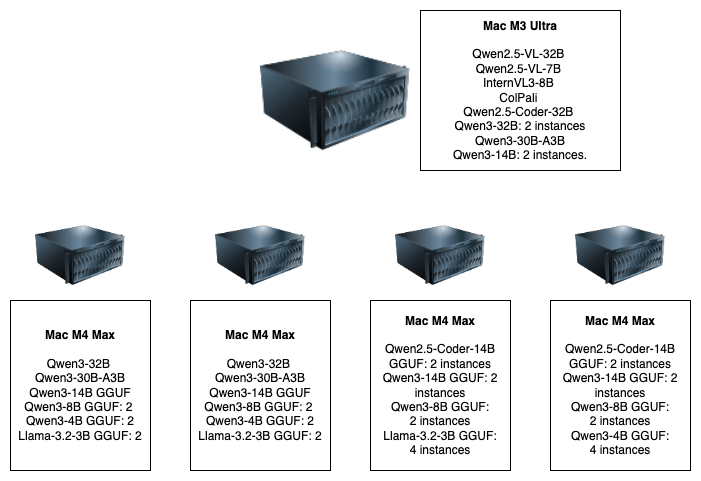
needed.

*Scalability:* Add more Mac Studios or upgrade to higher-memory variants (e.g., M3 Ultra with 256 GB) if concurrency exceeds 15 users.

Monitor GPU usage and queue times; if users’ requests regularly hit the 180–300 s cap, increase parallelism.

## Model Placement Visualization

The following diagram illustrates the distribution of models across the Mac Studio M3 Ultra and M4 Max machines, as outlined in the previous section. It highlights the specific placement of vision, routing, and text models, along with their quantization formats (BF16, INT8, 4-bit GGUF) and the number of instances per machine.



Implementation Readiness

* **Distribution:** Large BF16 models on M3 Ultra, quantized models on M4 Max for parallelism.
* **Concurrency:** Supports 15 users within 180–300 seconds.
* **Serving:** Ollama/vLLM with LangChain integration leverages Apple GPUs efficiently.

**Storage Requirement:**

Similar to the token estimation there would be a need to provide adequate storage for daily incoming unstructured and semi-structured data.

### **Sizing Analysis for Vector Database Supporting RAG/AGENTS**

#### **Key Assumptions:**

1. **Corpus Size:** 150 billion tokens (text to be indexed for retrieval).
2. **Embedding Dimension:** 1024 (float32, 4 bytes per value).
3. **Text Chunk Size:** 512 tokens per chunk (common default).
4. **Token Storage:** 4 bytes per token (UTF-8 or similar encoding).
5. **Indexing Method:** Hybrid in-memory/disk (e.g., DiskANN) for balance of speed and cost.

### **1. Vector Storage**

* **Number of Chunks:**150B tokens/ 512tokens per chunk ≈ 293M chunks
* **Vector Size per Chunk:**1024 dimensions×4 bytes = 4 KB
* **Total Vector Storage:**293M×4 KB = 1.17 TB

### **2. Text Storage**

* **Text per Chunk:**512 tokens ×4 bytes= 2 KB
* **Total Text Storage:**293M × 2 KB= 572 GB

### **3. Index Storage**

* **Disk-Based Index (e.g., DiskANN):**Roughly equal to vector size (~1.2 TB).
* **Total Disk Storage (Vectors + Text + Index):**1.17 TB+0.57 TB+1.2 TB≈3 TB (with 20% overhead: **~3.6 TB**).

### **4. Memory Requirements**

* **DiskANN (Hybrid SSD/RAM):**Caches ~10–20% of the index in memory.  
  1.2 TB×0.2=0.24 TB (240 GB RAM)  
  **Recommendation:** 256–512 GB RAM for headroom.
* **Pure In-Memory Index (e.g., HNSW):**1.2 TB RAM1.2 TB RAM, requiring distributed nodes (e.g., 10+ nodes with 128 GB each).

### **5. Network and Compute**

* **Query Throughput:** Scales with nodes (e.g., 1k–10k queries/second with distributed clusters).
* **CPU/GPU:** Modern CPUs (e.g., Intel Xeon) or GPUs (for accelerated similarity search).

### **6. Design Considerations**

1. **Chunk Size:** Smaller chunks → more vectors (e.g., 256 tokens → ~586M vectors, doubling storage).
2. **Quantization:** Use 8-bit SQ (reduces vector storage by 75% but impacts accuracy).
3. **Distributed vs. Single-Node:**

* **Single-Node:** Feasible with DiskANN (~512 GB RAM + 4 TB SSD).
* **Distributed:** Required for pure in-memory indices (e.g., Milvus cluster).

1. **Metadata:** Add 10–20% storage for additional metadata (e.g., timestamps, source URLs).

### **Summary**

|  |  |  |
| --- | --- | --- |
| **Component** | **Size** | **Notes** |
| **Vectors (1024D)** | 1.17 TB | Float32 embeddings. |
| **Text Chunks** | 572 GB | 512 tokens/chunk. |
| **Index (DiskANN)** | 1.2 TB | Hybrid SSD storage. |
| **RAM (DiskANN)** | 240–512 GB | For cached graph nodes. |
| **Total Storage** | 3.6 TB (with overhead) | SSDs recommended for fast retrieval. |

**Recommended Architecture:**

* **Single-Node:** 512 GB RAM + 4 TB NVMe SSD (DiskANN).
* **Distributed:** 10 nodes with 128 GB RAM + 1 TB SSD each (HNSW).

Adjust based on query latency/throughput requirements and accuracy trade-offs. The overhead on RAM can be reduced by lowering the precision and/or size of the embedding.

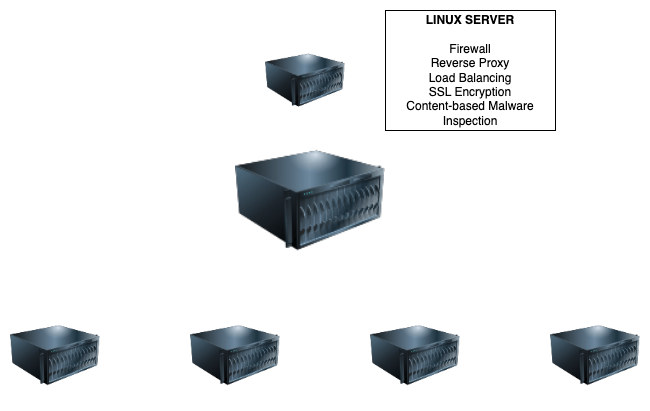
## Phase Three: Enterprise ready solution (2-3months)

Vox and its partners will work closely with the client to improve the system and user experience by first designing a user interface to provide feedback on the quality of the response of the LLM system. This feedback performs two important tasks – Supervised Fine Tuning and Reinforcement learning with human feedback. These tasks are Gen-AI industry standards to improve both the quality and latency of the responses.

### Estimated Load for LLM inferencing

* Max Input tokens = 128K
* Max Output tokens = 8K
* Mean concurrent users = 15
* Max allowable concurrent users = 30
* Output Streaming token speed per user = 15/sec

### Installing Firewall, Apache Web Server, mod\_proxy, Load Balancing, and SSL Certificates



* Prepare the Linux environment by updating system packages and installing essential utilities. This ensures all security patches are applied and necessary tools are available for further configuration
* Install the Apache HTTP server to act as the reverse proxy and load balancer. Configure it to start automatically on boot to ensure service continuity after reboots.
* Use UFW (Uncomplicated Firewall) to implement a default-deny policy on all incoming connections. Explicitly allow only required services like SSH from trusted IP addresses and HTTP/HTTPS for web traffic.
* To secure web traffic, configure HTTPS using commercial SSL certificates.
* Pipeline to securely scans uploaded files before further processing. It supports enterprise security standards with isolation, automation, logging, and alerting.

### Milvus Distributed with MinIO

We set up **Milvus Distributed** with **MinIO** in **High Availability (HA)** mode on a production-grade Kubernetes cluster. The deployment targets **on-premise** physical or virtual servers, ensuring **scalability**, **fault tolerance**, and **high performance**.

#### High Availability (HA)

#### Disaster Recovery Plan

#### Security

#### Monitoring and Alerts

#### Performance Tuning and Optimization

#### Scalability

#### Backup and Snapshotting

#### Upgrades and Version Management

### Privileged-Aware Vector Search and LLM Response Control

We are building a system where LLMs access and interpret data from diverse formats, and need to respect user roles and privileges during query handling. To ensure secure and efficient functionality, vectorization, access control, and query-time filtering must be combined.

#### Data Ingestion and Vectorization

#### Vector Store with Metadata Filters

#### User Authentication & Authorization

#### Query-Time Filtering