Aura: Local-First AI for Knowledge Work

Presentation Handout | January 2026

What is Aura?

Aura is a **local-first, privacy-safe AI foundation** for knowledge work. It runs entirely on your machine—no cloud required, no data leaves your control.

> *"Windows Recall, but local and safe"*

The Privacy Promise

Component	Location
LLM (Ollama)	Your machine
Database (PostgreSQL)	Your machine
RAG Index (pgvector)	Your machine
Your Code & Documents	Your machine

No internet required. No telemetry. No API keys needed.

Why Local-First AI?

Cloud AI Challenges

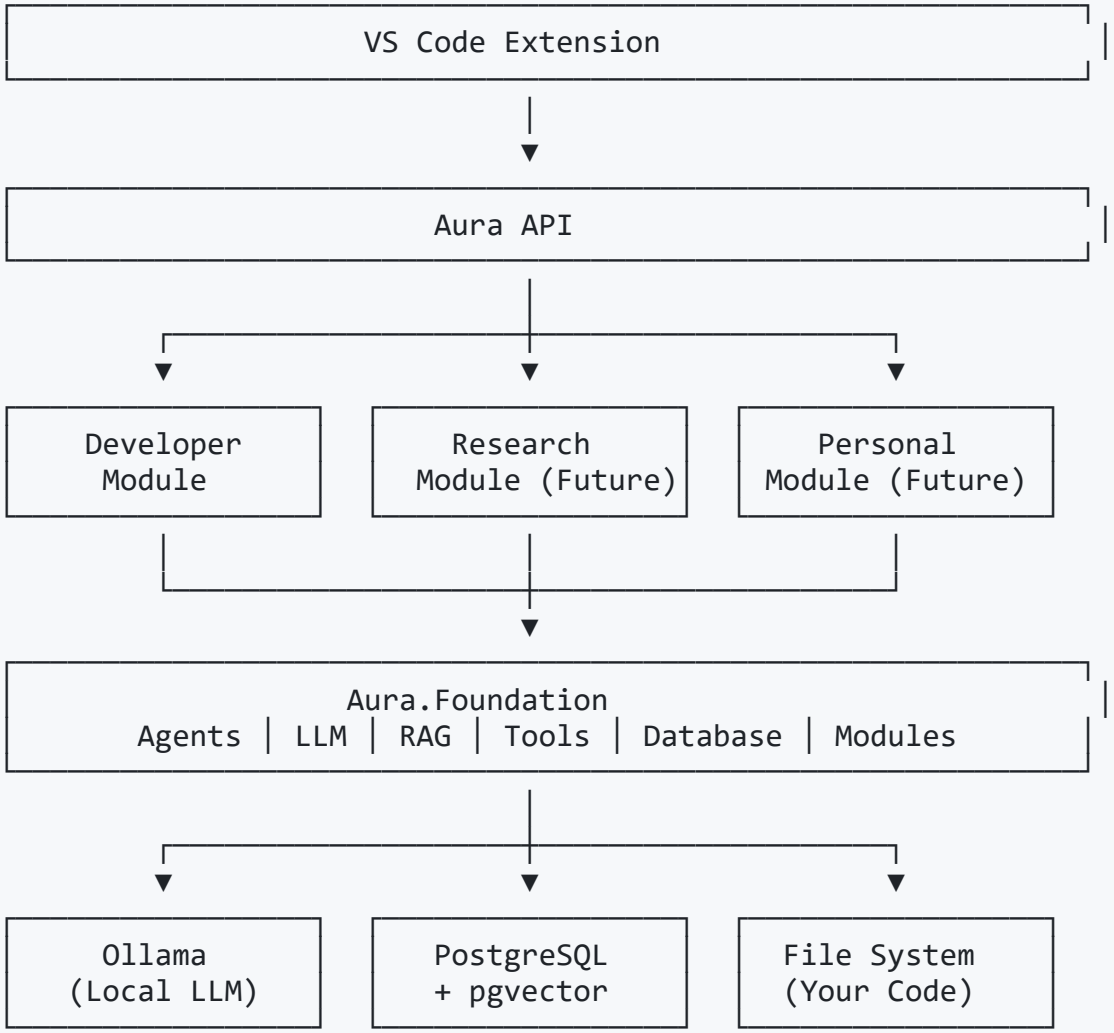
Challenge	Impact
Privacy	Your code goes to third-party servers
Compliance	HIPAA, GDPR, enterprise policies
Cost	$0.01-0.10 per request adds up
Availability	Requires internet, subject to outages
Latency	Network round-trip for every request

The Local Alternative

Modern local LLMs (Llama 3, Qwen 2.5 Coder) run well on consumer GPUs:

- 7B parameter models fit in 8GB VRAM

Architecture Overview



Core Features

1. RAG Pipeline (Semantic Search)

Index your codebase and documents for semantic search:

- **Embeddings**: Local models via Ollama (nomic-embed-text)
- **Storage**: PostgreSQL with pgvector extension
- **Indexing**: HNSW for fast approximate nearest neighbor

2. Code Graph (Structural Search)

Beyond text similarity—understand code structure:

Query	Example
Find by name	`GET /api/graph/find/HttpClient`
Find implementations	`GET /api/graph/implementations/IService`

Developer Workflow (Use Case)

The Developer Module provides a complete workflow for code automation:

```
CREATE → ANALYZE → PLAN → EXECUTE → COMPLETE
  ↓         ↓         ↓         ↓          ↓
 Issue    RAG +     Steps    Human-in    Commit
          Graph               the-loop    + PR
```

Workflow Steps

Phase	What Happens
Create	User describes task, git worktree created
Analyze	RAG finds relevant code, agent enriches requirements
Plan	Business analyst agent generates execution steps
Execute	Each step executed by appropriate agent

Technology Stack

Layer	Technology
Runtime	.NET 9
Orchestration	.NET Aspire
Database	PostgreSQL + pgvector
Vector Search	pgvector HNSW
Local LLM	Ollama
C# Analysis	Roslyn
Multi-Lang Parse	TreeSitter
Extension	VS Code (TypeScript)

Quick Reference

Requirements

- .NET 9.0
- Docker (for PostgreSQL) or native PostgreSQL with pgvector
- Ollama (for local LLM)
- GPU recommended (but CPU works)

Getting Started

```
# Clone
git clone https://github.com/johnazariah/aura.git
cd aura

# Build
dotnet build

# Run
```

Links & Resources

Resource	URL
GitHub	`github.com/johnazariah/aura`
License	MIT
Documentation	`./docs/` in repository

Key Takeaways

1. **Privacy Matters** - Your data should stay on your machine
2. **Local LLMs Are Ready** - 7B models are good enough for most tasks
3. **RAG + Code Graph** - Text similarity + structural queries
4. **Human-in-the-Loop** - AI assists, you decide
5. **Composable Design** - Enable only what you need

> *"The best software is built not by adding features until it works, but by removing complexity until it can't fail."*