# OllamaNet System Architecture



**Administrator**

System administrator managing the platform

**End User**

A user of the OllamaNet platform

## OllamaNet System
[enterprise]

Administers
[HTTPS]

Uses
[HTTPS]

Uses
[HTTPS]

### Frontend Layer
[system]

**Web Application**
*[React, TypeScript]*

Single-page application providing user interface

**Mobile Application**
*[React Native]*

Mobile interface for OllamaNet

**Developer**

Developer integrating with OllamaNet APIs

Makes API calls to
[HTTPS]

Makes API calls to
[HTTPS]

Integrates with
[HTTPS/API]

### API Gateway Layer
[system]

**API Gateway**
*[.NET Core, Ocelot]*

Routes requests and handles authentication, CORS, etc.

Routes conversation requests to
[REST]

Routes explore requests to
[REST]

Routes auth requests to
[REST]

Routes admin requests to
[REST]

Routes inference requests to
[REST]

### Service Layer
[system]

**Conversation Service**
*[.NET Core Web API]*

Chat and conversation management

**Explore Service**
*[.NET Core Web API]*

Model discovery and browsing

**Auth Service**
*[.NET Core Web API]*

Authentication and authorization

**Admin Service**
*[.NET Core Web API]*

Administration and model operations

Requests completions from
[REST]

Configures
[REST]

Sends emails via
[SMTP]

Reports metrics to
[HTTP]

**Inference Service**
*[Python, Jupyter]*

AI model inference

**Email Service**

For sending notifications

**Monitoring Tools**

For system monitoring

Queries for similar content
[HTTP]

Caches conversations
[Redis Client]

Reads/writes conversation data
[Entity Framework]

Reads/writes model catalog
[Entity Framework]

Reads/writes auth data
[Entity Framework]

Subscribes to service updates
[AMQP]

Reads/writes admin data
[Entity Framework]

Subscribes to service updates
[AMQP]

Publishes URL updates to
[AMQP]

Runs models via
[HTTP]

Integrates with
[HTTP]

### Infrastructure Layer
[system]

**Vector Database**
*[Pinecone]*

Vector embeddings storage

**Cache**
*[Redis]*

Distributed caching

**Shared SQL Database**
*[SQL Server]*

Single database with logical separation for services

**Message Broker**
*[RabbitMQ]*

Asynchronous messaging

### AI Layer
[system]

**Ollama**
*[Go]*

Local LLM runtime

**External APIs**

Third-party integrations

Loads and runs
[File System]

**AI Models**
*[GGUF format]*

Large language models