**Project Citadel: Comprehensive Tech Stack Documentation**

**Executive Summary**

Project Citadel leverages a modern Python-based tech stack built around advanced web crawling, vector databases, and AI/ML frameworks. The architecture emphasizes asynchronous processing, enterprise-grade vector storage, and cutting-edge AI agent frameworks for retrieval-augmented generation (RAG) capabilities.

**1. Core Technology Stack Overview**

mermaid

graph TB

subgraph "Frontend Layer"

STREAMLIT[Streamlit 1.45.0<br/>Interactive Web UI]

end

subgraph "Application Framework"

PYDANTIC\_AI[Pydantic AI 0.1.8<br/>Agent Framework]

FASTAPI[FastAPI 0.115.9<br/>API Framework]

ASYNCIO[AsyncIO<br/>Async Runtime]

end

subgraph "Web Crawling Engine"

CRAWL4AI[Crawl4AI 0.6.2<br/>Intelligent Crawler]

PLAYWRIGHT[Playwright 1.52.0<br/>Browser Automation]

STEALTH[tf-playwright-stealth 1.1.2<br/>Anti-Detection]

end

subgraph "AI/ML Stack"

OPENAI[OpenAI 1.76.2<br/>LLM & Embeddings]

SENTENCE\_T[SentenceTransformers 4.1.0<br/>Local Embeddings]

TRANSFORMERS[Transformers 4.51.3<br/>HuggingFace Models]

TORCH[PyTorch 2.7.0<br/>ML Backend]

end

subgraph "Vector Database"

CHROMADB[ChromaDB 1.0.7<br/>Vector Store]

HNSWLIB[chroma-hnswlib 0.7.6<br/>Similarity Search]

SQLITE[aiosqlite 0.21.0<br/>Metadata Storage]

end

subgraph "Data Processing"

PANDAS[Pandas 2.2.3<br/>Data Manipulation]

NUMPY[NumPy 2.2.5<br/>Numerical Computing]

SCIKIT[Scikit-learn 1.6.1<br/>ML Utilities]

BEAUTIFULSOUP[BeautifulSoup4 4.13.4<br/>HTML Parsing]

end

%% Dependencies

STREAMLIT --> PYDANTIC\_AI

PYDANTIC\_AI --> OPENAI

CRAWL4AI --> PLAYWRIGHT

PLAYWRIGHT --> STEALTH

SENTENCE\_T --> TORCH

CHROMADB --> HNSWLIB

CHROMADB --> SQLITE

classDef frontend fill:#e1f5fe

classDef framework fill:#f3e5f5

classDef crawling fill:#e8f5e8

classDef ai fill:#fff3e0

classDef database fill:#fce4ec

classDef processing fill:#f1f8e9

class STREAMLIT frontend

class PYDANTIC\_AI,FASTAPI,ASYNCIO framework

class CRAWL4AI,PLAYWRIGHT,STEALTH crawling

class OPENAI,SENTENCE\_T,TRANSFORMERS,TORCH ai

class CHROMADB,HNSWLIB,SQLITE database

class PANDAS,NUMPY,SCIKIT,BEAUTIFULSOUP processing

**2. Detailed Dependency Analysis**

**2.1 Frontend & User Interface**

Table

| **Component** | **Version** | **Purpose** | **Dependencies** |
| --- | --- | --- | --- |
| **Streamlit** | 1.45.0 | Interactive web interface for RAG queries | altair, blinker, click, gitpython, pydeck, tornado |
| **Altair** | 5.5.0 | Statistical visualization library | pandas, numpy |
| **PyDeck** | 0.9.1 | WebGL-powered visualizations | - |

**Key Features:**

* Real-time streaming interface for RAG responses
* Session state management for conversation history
* Interactive chat UI with async streaming support

**2.2 AI Agent Framework**

Table

| **Component** | **Version** | **Purpose** | **Dependencies** |
| --- | --- | --- | --- |
| **Pydantic AI** | 0.1.8 | Modern AI agent framework with tool calling | pydantic-core, pydantic-settings |
| **Pydantic Core** | 2.33.2 | Fast validation and serialization | typing\_extensions |
| **Pydantic** | 2.11.4 | Data validation and settings management | annotated-types |

**Ecosystem Components:**

* pydantic-ai-slim==0.1.8: Lightweight AI agent core
* pydantic-evals==0.1.8: Agent evaluation framework
* pydantic-graph==0.1.8: Graph-based agent workflows
* pydantic-settings==2.9.1: Configuration management

**2.3 Web Crawling & Automation**

Table

| **Component** | **Version** | **Purpose** | **Dependencies** |
| --- | --- | --- | --- |
| **Crawl4AI** | 0.6.2 | Intelligent web crawling framework | playwright, beautifulsoup4, lxml |
| **Playwright** | 1.52.0 | Browser automation and control | pyee, greenlet |
| **tf-playwright-stealth** | 1.1.2 | Anti-detection for web scraping | playwright |
| **BeautifulSoup4** | 4.13.4 | HTML/XML parsing | soupsieve |
| **lxml** | 5.4.0 | High-performance XML/HTML parser | - |

**Additional Crawling Utilities:**

* fake-useragent==2.2.0: Realistic user agent rotation
* fake-http-header==0.3.5: HTTP header spoofing
* cssselect==1.3.0: CSS selector engine

**2.4 AI/ML Stack**

Table

| **Component** | **Version** | **Purpose** | **Dependencies** |
| --- | --- | --- | --- |
| **OpenAI** | 1.76.2 | GPT models and embeddings API | httpx, pydantic, typing\_extensions |
| **SentenceTransformers** | 4.1.0 | Local sentence embeddings | transformers, torch, numpy |
| **Transformers** | 4.51.3 | HuggingFace model library | torch, tokenizers, safetensors |
| **PyTorch** | 2.7.0 | Deep learning framework | sympy, mpmath, networkx |

**Supporting ML Libraries:**

* tiktoken==0.9.0: OpenAI tokenization
* tokenizers==0.21.1: Fast tokenization
* safetensors==0.5.3: Safe tensor serialization
* huggingface-hub==0.30.2: Model hub integration

**2.5 Vector Database & Storage**

Table

| **Component** | **Version** | **Purpose** | **Dependencies** |
| --- | --- | --- | --- |
| **ChromaDB** | 1.0.7 | Vector database for embeddings | hnswlib, sqlite, fastapi |
| **chroma-hnswlib** | 0.7.6 | Hierarchical navigable small world graphs | - |
| **aiosqlite** | 0.21.0 | Async SQLite interface | - |
| **FastAPI** | 0.115.9 | API framework for ChromaDB | starlette, pydantic |

**Database Utilities:**

* PyPika==0.48.9: SQL query builder
* mmh3==5.1.0: MurmurHash3 for hashing
* xxhash==3.5.0: Fast hashing algorithm

**2.6 Data Processing & Analysis**

Table

| **Component** | **Version** | **Purpose** | **Dependencies** |
| --- | --- | --- | --- |
| **Pandas** | 2.2.3 | Data manipulation and analysis | numpy, python-dateutil, pytz |
| **NumPy** | 2.2.5 | Numerical computing | - |
| **Scikit-learn** | 1.6.1 | Machine learning utilities | numpy, scipy, joblib |
| **SciPy** | 1.15.2 | Scientific computing | numpy |

**Text Processing:**

* nltk==3.9.1: Natural language toolkit
* regex==2024.11.6: Advanced regular expressions
* rank-bm25==0.2.2: BM25 ranking algorithm

**2.7 Async & Networking**

Table

| **Component** | **Version** | **Purpose** | **Dependencies** |
| --- | --- | --- | --- |
| **aiohttp** | 3.11.18 | Async HTTP client/server | aiosignal, attrs, frozenlist, multidict, yarl |
| **httpx** | 0.28.1 | Modern HTTP client | httpcore, idna, sniffio |
| **requests** | 2.32.3 | Synchronous HTTP library | charset-normalizer, idna, urllib3, certifi |
| **websockets** | 14.2 | WebSocket implementation | - |

**Async Utilities:**

* anyio==4.9.0: Async abstraction layer
* asyncio: Built-in async runtime
* aiofiles==24.1.0: Async file operations

**2.8 Configuration & Environment**

Table

| **Component** | **Version** | **Purpose** | **Dependencies** |
| --- | --- | --- | --- |
| **python-dotenv** | 1.1.0 | Environment variable management | - |
| **PyYAML** | 6.0.2 | YAML configuration parsing | - |
| **Typer** | 0.15.3 | CLI framework | click, shellingham |
| **click** | 8.1.8 | Command line interface creation | colorama |

**3. Architecture Integration Patterns**

**3.1 Async Processing Flow**

mermaid

sequenceDiagram

participant App as Python Application

participant Crawl4AI as Crawl4AI Engine

participant Playwright as Browser Engine

participant ChromaDB as Vector Database

participant OpenAI as AI Services

App->>Crawl4AI: async crawl\_request

Crawl4AI->>Playwright: browser.new\_page()

Playwright-->>Crawl4AI: page\_content

Crawl4AI-->>App: markdown\_content

App->>ChromaDB: async store\_vectors

ChromaDB-->>App: storage\_confirmation

App->>OpenAI: async query\_completion

OpenAI-->>App: streaming\_response

**3.2 Data Processing Pipeline**

mermaid

flowchart LR

subgraph "Input Processing"

HTML[Raw HTML] --> BS4[BeautifulSoup4]

BS4 --> LXML[lxml Parser]

LXML --> MD[Markdown Content]

end

subgraph "Text Processing"

MD --> CHUNKER[Smart Chunker]

CHUNKER --> NLTK[NLTK Processing]

NLTK --> CLEAN[Clean Text]

end

subgraph "Vector Processing"

CLEAN --> ST[SentenceTransformers]

ST --> TORCH[PyTorch Backend]

TORCH --> VECTORS[Vector Embeddings]

end

subgraph "Storage"

VECTORS --> CHROMA[ChromaDB]

CHROMA --> HNSWLIB[HNSW Index]

CHROMA --> SQLITE[SQLite Metadata]

end

**4. Performance & Scalability Features**

**4.1 Memory Management**

Table

| **Component** | **Memory Feature** | **Implementation** |
| --- | --- | --- |
| **psutil** | System memory monitoring | Real-time memory usage tracking |
| **more-itertools** | Efficient batching | batched() for chunk processing |
| **Playwright** | Browser resource control | Session pooling and cleanup |
| **ChromaDB** | Batch operations | Configurable batch sizes |

**4.2 Concurrent Processing**

python

# Memory-adaptive dispatcher

dispatcher = MemoryAdaptiveDispatcher(

memory\_threshold\_percent=70.0,

check\_interval=1.0,

max\_session\_permit=max\_concurrent

)

# Parallel crawling with resource limits

results = await crawler.arun\_many(

urls=urls,

config=crawl\_config,

dispatcher=dispatcher

)

**4.3 Error Handling & Resilience**

Table

| **Component** | **Error Handling** | **Recovery Strategy** |
| --- | --- | --- |
| **tenacity** | Retry mechanisms | Exponential backoff |
| **backoff** | Rate limiting | Adaptive retry delays |
| **Crawl4AI** | Browser failures | Session recreation |
| **ChromaDB** | Database errors | Graceful degradation |

**5. Security & Authentication**

**5.1 Secure Communications**

Table

| **Component** | **Security Feature** | **Implementation** |
| --- | --- | --- |
| **cryptography** | Encryption utilities | TLS/SSL support |
| **bcrypt** | Password hashing | Secure credential storage |
| **PyJWT** | JSON Web Tokens | API authentication |
| **pyOpenSSL** | SSL/TLS | Secure connections |

**5.2 Environment Security**

python

# Secure environment variable management

from dotenv import load\_dotenv

import os

load\_dotenv()

openai\_api\_key = os.getenv("OPENAI\_API\_KEY")

model\_choice = os.getenv("MODEL\_CHOICE", "gpt-4-turbo")

**6. Development & Testing Framework**

**6.1 Testing Infrastructure**

Table

| **Component** | **Purpose** | **Features** |
| --- | --- | --- |
| **pytest** | Testing framework | Async test support |
| **pytest-mock** | Mocking utilities | Test isolation |
| **build** | Package building | Distribution management |

**6.2 Code Quality**

Table

| **Component** | **Purpose** | **Features** |
| --- | --- | --- |
| **typing\_extensions** | Type hints | Enhanced type safety |
| **Pygments** | Code highlighting | Syntax highlighting |
| **rich** | Rich text output | Beautiful console output |

**7. Deployment & Monitoring**

**7.1 Observability Stack**

Table

| **Component** | **Purpose** | **Integration** |
| --- | --- | --- |
| **logfire** | Application monitoring | OpenTelemetry-based |
| **opentelemetry-api** | Telemetry collection | Distributed tracing |
| **coloredlogs** | Enhanced logging | Structured log output |

**7.2 Production Utilities**

Table

| **Component** | **Purpose** | **Features** |
| --- | --- | --- |
| **uvicorn** | ASGI server | High-performance async server |
| **watchdog** | File system monitoring | Auto-reload capabilities |
| **humanize** | Human-readable formats | User-friendly output |

**8. Critical Dependencies Matrix**

**8.1 Core Runtime Dependencies**

mermaid

graph TD

subgraph "Critical Path"

PYTHON[Python 3.11+] --> PYDANTIC[Pydantic 2.11.4]

PYDANTIC --> PYDANTIC\_AI[Pydantic AI 0.1.8]

PYTHON --> ASYNCIO[AsyncIO Runtime]

ASYNCIO --> AIOHTTP[aiohttp 3.11.18]

end

subgraph "AI Processing"

TORCH[PyTorch 2.7.0] --> SENTENCE\_T[SentenceTransformers 4.1.0]

OPENAI\_API[OpenAI 1.76.2] --> TIKTOKEN[tiktoken 0.9.0]

end

subgraph "Web Crawling"

PLAYWRIGHT[Playwright 1.52.0] --> CRAWL4AI[Crawl4AI 0.6.2]

BEAUTIFULSOUP[BeautifulSoup4 4.13.4] --> LXML[lxml 5.4.0]

end

subgraph "Data Storage"

FASTAPI[FastAPI 0.115.9] --> CHROMADB[ChromaDB 1.0.7]

CHROMADB --> HNSWLIB[chroma-hnswlib 0.7.6]

CHROMADB --> AIOSQLITE[aiosqlite 0.21.0]

end

%% Cross-dependencies

PYDANTIC\_AI --> OPENAI\_API

CRAWL4AI --> PLAYWRIGHT

SENTENCE\_T --> TORCH

classDef critical fill:#ffcdd2

classDef ai fill:#c8e6c9

classDef crawling fill:#bbdefb

classDef storage fill:#fff9c4

class PYTHON,PYDANTIC,PYDANTIC\_AI,ASYNCIO,AIOHTTP critical

class TORCH,SENTENCE\_T,OPENAI\_API,TIKTOKEN ai

class PLAYWRIGHT,CRAWL4AI,BEAUTIFULSOUP,LXML crawling

class FASTAPI,CHROMADB,HNSWLIB,AIOSQLITE storage

**8.2 Version Compatibility Matrix**

Table

| **Python Version** | **Core Framework** | **AI/ML Stack** | **Web Crawling** | **Database** |
| --- | --- | --- | --- | --- |
| **3.11+** | ✅ Recommended | ✅ Full Support | ✅ Optimized | ✅ Stable |
| **3.10** | ⚠️ Limited | ✅ Compatible | ✅ Compatible | ✅ Compatible |
| **3.9** | ❌ Not Supported | ⚠️ Limited | ⚠️ Limited | ✅ Compatible |

**9. Installation & Setup Requirements**

**9.1 System Prerequisites**

bash

# Python 3.11+ required

python --version # Should be 3.11 or higher

# System dependencies for Playwright

playwright install chromium

playwright install-deps

# System dependencies for machine learning

# Linux: apt-get install build-essential

# macOS: xcode-select --install

# Windows: Visual Studio Build Tools

**9.2 Environment Configuration**

bash

# Required environment variables

OPENAI\_API\_KEY=your\_openai\_api\_key

MODEL\_CHOICE=gpt-4-turbo # Optional, defaults to gpt-4-turbo-mini

CHROMA\_DB\_PATH=./chroma\_db # Optional, defaults to ./chroma\_db

**9.3 Resource Requirements**

Table

| **Component** | **Minimum RAM** | **Recommended RAM** | **Storage** |
| --- | --- | --- | --- |
| **Base System** | 4GB | 8GB | 2GB |
| **ChromaDB** | 2GB | 4GB | Variable |
| **Playwright** | 1GB | 2GB | 500MB |
| **ML Models** | 2GB | 4GB | 1GB |
| **Total** | **9GB** | **18GB** | **3.5GB** |

This comprehensive tech stack provides Project Citadel with enterprise-grade capabilities for intelligent document processing, vector storage, and AI-powered question answering while maintaining flexibility for future scalability and feature enhancement.