## Technology Stack (Architecture & Stack)

## Technical Architecture

The architecture of Citizen AI Platform integrates real-time generative AI assistance, public sentiment analysis, dynamic dashboards, and personalized policy response simulation. The system uses a Streamlit + Gradio hybrid frontend, a Python backend, and integrates the IBM Granite language model via Hugging Face. Data is visualized using Plotly, and sentiment models may use NLTK or VADER libraries. Future versions aim to migrate to IBM Cloud for scalability and 24/7 uptime.

## Table-1: Components & Technologies

|  |  |  |
| --- | --- | --- |
| S.No | Component Description | Technology |
| 1 | User Interface – Tabs for Assistant, Dashboard, etc. | Streamlit, Gradio |
| 2 | Conversational Logic – AI assistant + prompts | Python, IBM Granite LLM |
| 3 | Sentiment Analysis Module | Python, VADER, NLTK |
| 4 | Visualization & Dashboard | Plotly, Pandas |
| 5 | Authentication (optional for policymakers) | Streamlit session state |
| 6 | Model Hosting & Inference | HuggingFace Transformers, IBM Granite |
| 7 | Future Cloud Hosting | IBM Cloud, Cloud Foundry |

## Table-2: Application Characteristics

|  |  |  |
| --- | --- | --- |
| S.No | Characteristic Description | Technology |
| 1 | Open-Source & Reproducible | Python, Streamlit, HuggingFace |
| 2 | Security & Role Isolation | Session state, local file control |
| 3 | Modular & Cloud-ready Architecture | Python microservices, IBM Cloud (future) |
| 4 | Availability & Access | Local demo + scalable cloud deployment |
| 5 | Performance & Optimization | Efficient prompt processing, caching |