**Project Design Phase-II**

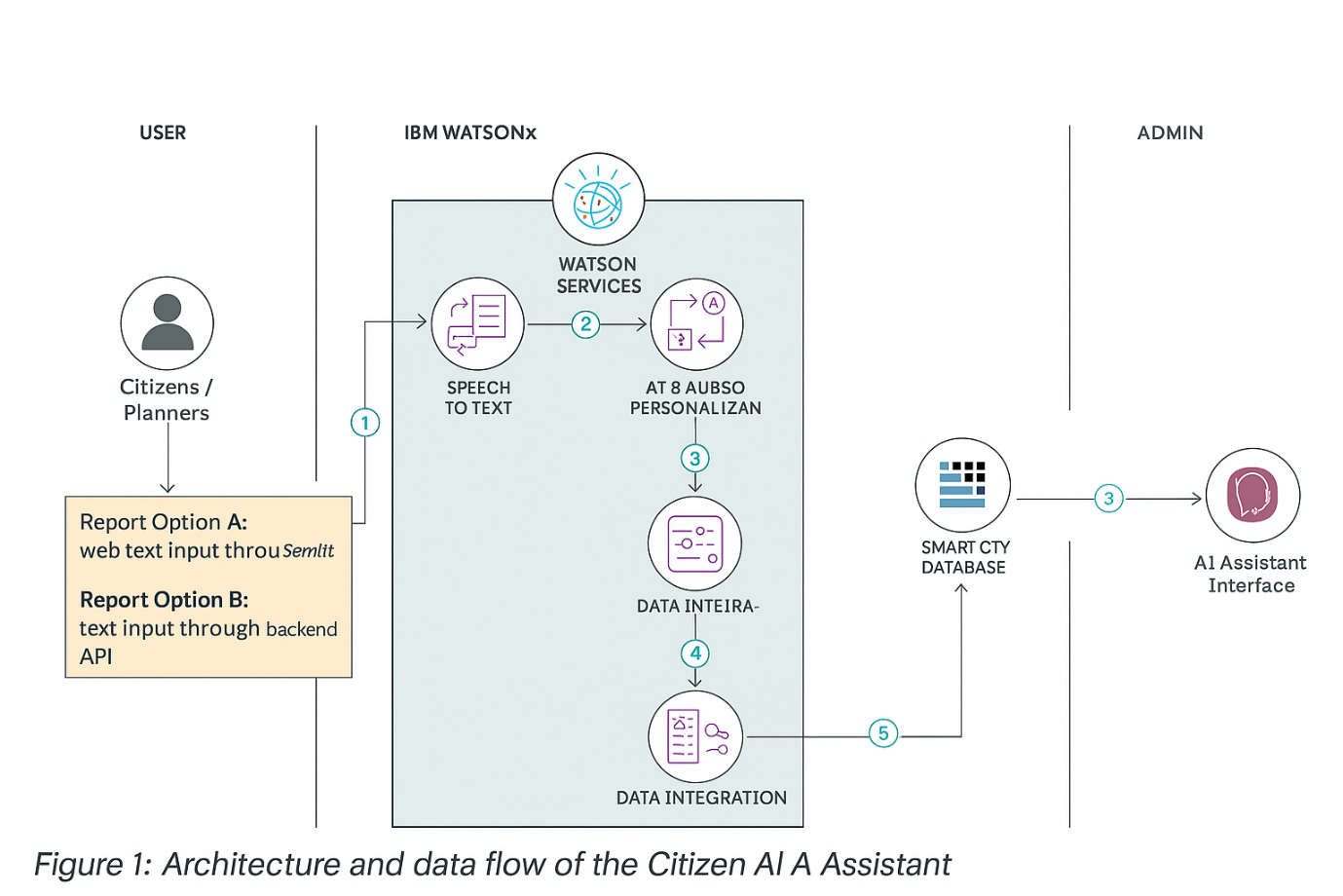
**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| **Date** | **25 june 2025** |
| **Team ID** | **LTVIP2025TMID37165** |
| **Project Name** | **Citizen AI – Intelligent Citizen Engagement Platform.** |
| **Maximum Marks** | **4 Marks** |

**Technical Architecture:**

The Citizen AI Assistant architecture includes the following:  
- Multi-platform UI (Web, Mobile, Voice Interface)  
- AI-based NLP response engine powered by IBM Watson  
- Backend services for user query processing and government service integration  
- Cloud-hosted database and storage  
- Machine learning-based response improvement system  
- Integration with external APIs (e.g., Aadhaar, Weather)  
- Scalable deployment using container orchestration (e.g., Kubernetes)

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



**Table-1 : Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| 1 | User Interface | Web and mobile chatbot interface for citizen interaction | React.js, HTML, CSS, Flutter |
| 2 | Application Logic-1 | Handles user query processing and routing | Python (Flask/FastAPI) |
| 3 | Application Logic-2 | Speech-to-text conversion for voice queries | IBM Watson STT Service |
| 4 | Application Logic-3 | AI assistant for generating responses | IBM Watson Assistant |
| 5 | Database | Stores structured data like user info and query history | MySQL |
| 6 | Cloud Database | Cloud-hosted database for scalability | IBM Cloudant |
| 7 | File Storage | Storage for uploaded documents or logs | IBM Block Storage, AWS S3 |
| 8 | External API-1 | Fetch weather info for relevant queries | IBM Weather API |
| 9 | External API-2 | Verify citizen identity | Aadhaar API |
| 10 | Machine Learning Model | Model for improving response accuracy over time | Custom NLP model on IBM Watson ML |
| 11 | Infrastructure (Server / Cloud) | Hosting and deployment configuration | Kubernetes, IBM Cloud Foundry, Local Server |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| 1 | Open-Source Frameworks | Frameworks used for UI and backend development | React.js, Flask, FastAPI |
| 2 | Security Implementations | Authentication, encryption, and access control | OAuth 2.0, HTTPS, JWT, IAM, OWASP Top 10 |
| 3 | Scalable Architecture | Supports horizontal scaling and modular components | Microservices, Kubernetes |
| 4 | Availability | Ensures service continuity with load balancing | HAProxy, Distributed Cloud Services |