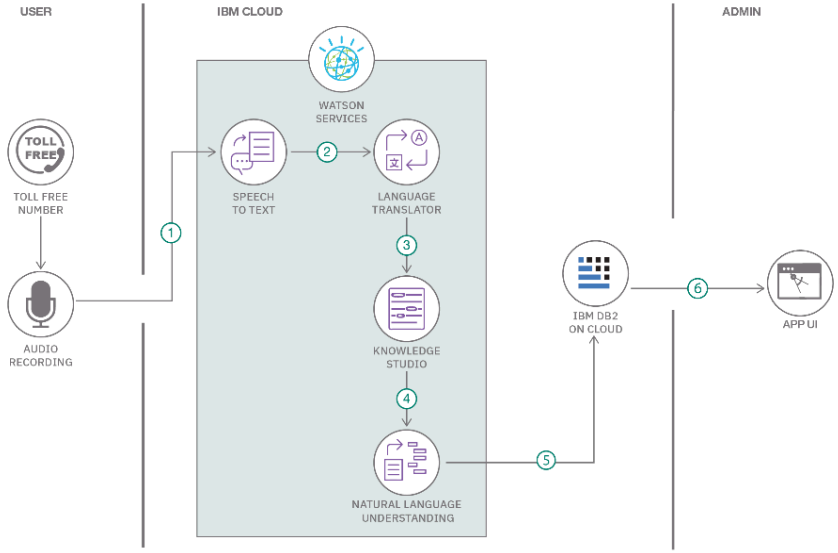
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	31 January 3035
Team ID	SWTID1742930170
Project Name	I-Movies : Movie Ticket Booking System

Technical Architecture:

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



Guidelines:

- Include all the processes (As an application logic / Technology Block)
- Provide infrastructural demarcation (Local / Cloud)
- Indicate external interfaces (third party API's etc.)
- Indicate Data Storage components / services
- Indicate interface to machine learning models (if applicable)

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Core business logic (ticket booking, payment)	Java / Python
3.	Application Logic-2	Voice command processing	IBM Watson STT service
4.	Application Logic-3	Conversational chatbot for support	IBM Watson Assistant
5.	Database	Manages user, movies, booking data	MySQL
6.	Cloud Database	Scalable cloud storage for data	IBM Cloudant
7.	File Storage	Store movie posters, QR codes, logs	IBM Block Storage / Local Filesystem
8.	External API-1	Weather API to suggest movies during weather conditions	IBM Weather API
9.	External API-2	User authentication via Aadhar	Aadhar API
10.	Machine Learning Model	Predict popular movies, user preferences	Recommendation ML Model (scikit-learn)
11.	Infrastructure (Server / Cloud)	Hosting the full app stack:	Cloud Foundry / Kubernetes / Local Server

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	React, Node.js, Flask/Django, scikit-learn	JavaScript, Python, Java
2.	Security Implementations	SHA-256 for passwords, OAuth2, JWT, IAM, OWASP practices	Encryption, Authentication APIs

S.No	Characteristics	Description	Technology
3.	Scalable Architecture	Microservice-based, containerized apps using Docker & Kubernetes	Kubernetes, Docker
4.	Availability	Load balancing, fallback replicas, distributed servers	Nginx, IBM Load Balancer
5.	Performance	CDN for fast content delivery, caching, async processing	Redis Cache, Cloud CDN, Asynchronous APIs