**DOMAIN NAME: CHATBOT DEPLOYMENT WITH IBM CLOUD WATSON ASSISTANT**

**PROJECT TITLE: CHATBOT USING IBM WATSON.**

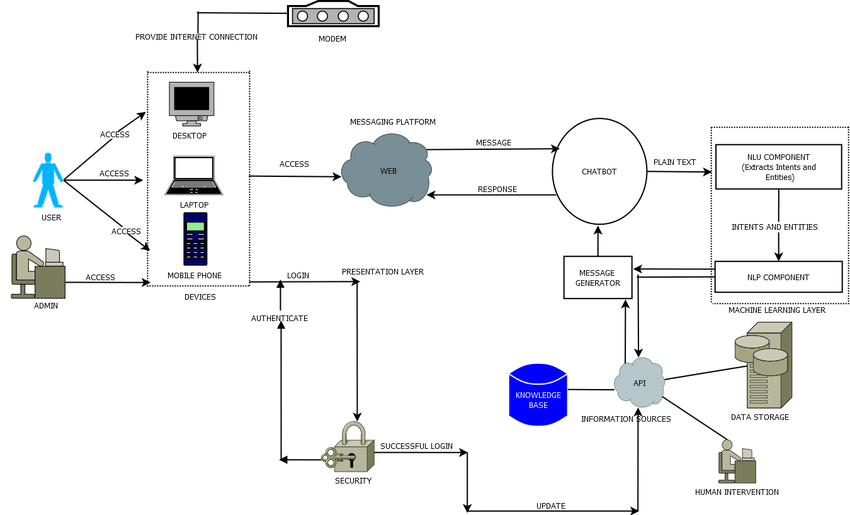
**Phase 3 Submission Document**

**BASIC COMPONENTS USED IN CHATBOT:**

IBM Watson provides various tools and services that can be used to create chatbots. Here are some of the basic components and services commonly used in building a chatbot using IBM Watson.

* IBM WATSON ASSISTANT
* NATURAL LANGUAGE UNDERSTANDING
* LANGUAGE TRANSLATOR
* WATSON DISCOVERY
* CLOUD FUNCTION
* IBM CLOUDS
* INTEGRATION WITH CHANNELS
* ANALYTICS AND MONITORING
* SECURITY AND COMPLIANCE

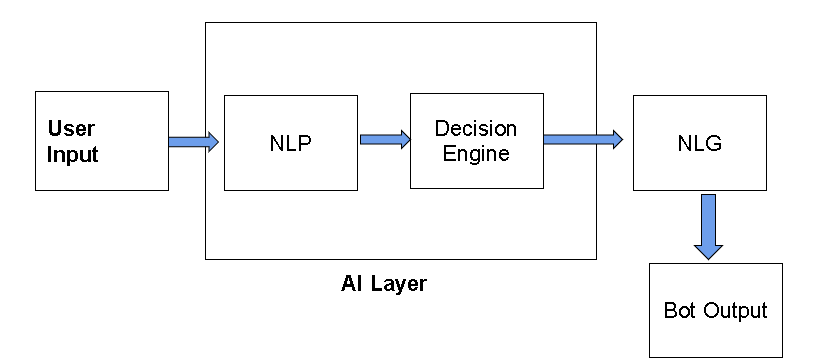
**ARCHITECTURE OF CHATBOT:**



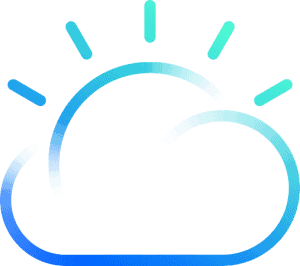
In the very first step, user input is given to the **AI layer** which has an **NLP**unit and a **Decision Engine.   The NLP (Natural Language Processing) unit** understands user intent & context i.e what exactly the user is asking for and based on that it converts user input into machine understandable language and processes it to a decision engine. **Decision Engine** Based on the NLP output **Decision Engine**uses machine learning algorithms to decide the bot response and process that response to the NLG unit **NLG (Natural Language Generation) unit**as the name suggest converts Machine Language into a plain text or human-understandable language. And finally, the output of the NLG unit is our Chatbot response which is processed to the user.

The architecture of a chatbot using IBM Watson is designed to be modular and scalable, allowing you to customize and expand the components based on your specific use case and requirements. It provides a flexible foundation for building chatbots that can understand, process, and respond to user interactions across different channels.

The architecture of a chatbot built using IBM Watson typically involves multiple components and services working together to provide a seamless conversational experience. Here's an overview of the architecture of a chatbot using IBM Watson.This is the front-end of the chatbot, where users interact with the bot. It could be a website, a messaging app (e.g., Facebook Messenger, Slack), a mobile app, or any platform where users initiate conversations.



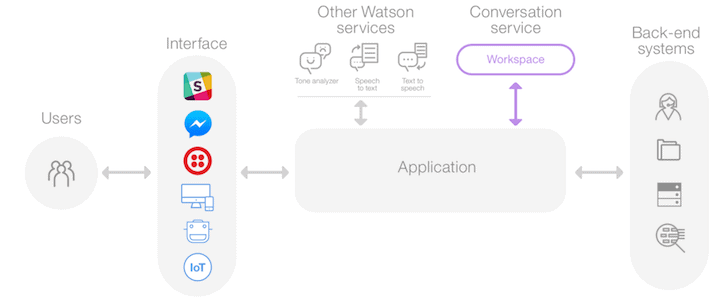
**IBM WATSON ASSISTANT:**



IBM Watson Assistant is an AI service in IBM Cloud that lets you build, train and deploy Chabot over the cloud**. Anyone can build a Chatbot.** You need not have any sort of technical experience to build a chatbot using IBM Watson Assistant. The intuitive interface helps you to easily create dynamic conversational flows.**Powered by NLP**The service uses NLP strategies like Intent Classification and Entity Recognition to understand user Intent and Context. **Connect anywhere**Watson’s assistant provides API’s to integrate your chatbot to your custom Application it may be Mobile/Web/Desktop and you can also integrate into various channels like Facebook messenger, Slack, Twilio, etc.

Watson Assistant is the core component responsible for understanding user input and generating responses. The architecture within Watson Assistant includes the following components:

* **Skills:** Each chatbot is represented as a skill, which contains the intent, entity, and dialog definitions.
* **Intents:** Intents represent the user's goal or purpose behind a message.
* **Entities:** Entities are objects or concepts mentioned in user input (e.g., dates, product names).
* **Dialog Flow:** The dialog flow defines how the chatbot responds to user inputs, with nodes and conditions to guide the conversation.



**CODING:**

from ibm\_watson import AssistantV2

from ibm\_cloud\_sdk\_core.authenticators import IAMAuthenticator

# Initialize Watson Assistant

authenticator = IAMAuthenticator('YOUR\_API\_KEY') # Replace with your API key

assistant = AssistantV2(

version='2021-06-14', # Update to the current version

authenticator=authenticator

)

assistant.set\_service\_url('https://api.us-south.assistant.watson.cloud.ibm.com') # Update with your service URL

# Create a new skill

workspace\_name = 'MyChatbotSkill' # Customize the name

response = assistant.create\_skill(

name=workspace\_name

).get\_result()

# Define an intent

intent\_name = 'greet'

assistant.create\_intent(

skill\_id=response['skill\_id'],

intent=intent\_name,

examples=[

{'text': 'Hello'},

{'text': 'Hi'},

{'text': 'Hey'},

]

)

# Define a dialog

assistant.create\_dialog\_node(

skill\_id=response['skill\_id'],

dialog\_node='greet',

conditions='#greet',

title='Greeting',

type='response',

output={'generic': [{'values': [{'text': 'Hello! How can I help you?'}]}],

parent=None

)

# Create a session

session = assistant.create\_session(skill\_id=response['skill\_id']).get\_result()

session\_id = session['session\_id']

# Chat with the bot

while True:

user\_input = input("You: ")

if user\_input.lower() == 'exit':

break

response = assistant.message(

skill\_id=response['skill\_id'],

session\_id=session\_id,

input={'message\_type': 'text', 'text': user\_input}

).get\_result()

# Extract the assistant's reply

reply = response['output']['generic'][0]['text']

print(f"Bot: {reply}")

# End the session

assistant.delete\_session(skill\_id=response['skill\_id'], session\_id=session\_id)