Chatbots
What, Why and How?

Dorai Thodla @dorait
Bhavani Ravi @geeky_bhavani
Ashish Cherian @Ashcherian

Learning Goals?

1. A Conceptual Understanding
2. How to build a few simple bots

Agenda

Introductory Session - Dorai Thodla (30 mts)

1. Chatbots what why and how
2. Demo of a chatbot
3. Code walkthrough of smbot (a simple minded bot built using NLTK)
4. Lab 1 practice (30 mins)

Advanced Session - Bhavani Ravi (30 mts)

1. Session-2: An API.ai based TechBot (an improved bot) - Bhavani
2. Revisiting architecture and flow
3. Training api.ai with intents/entities
4. Mapping intents to actions
5. Lab 2 practice (60 mins)

Topics

- What is a Chatbot?
- Why do we need chatbots?
- How does a chatbot work?
- How can we build one?
- Current chatbot ecosystem

What is a Chatbot?

- A conversational agent with Intelligence
- Normally lives in a messaging app
- Interacts with users using speech or text messaging
- Uses rules or Machine Learning to learn and improve interactions

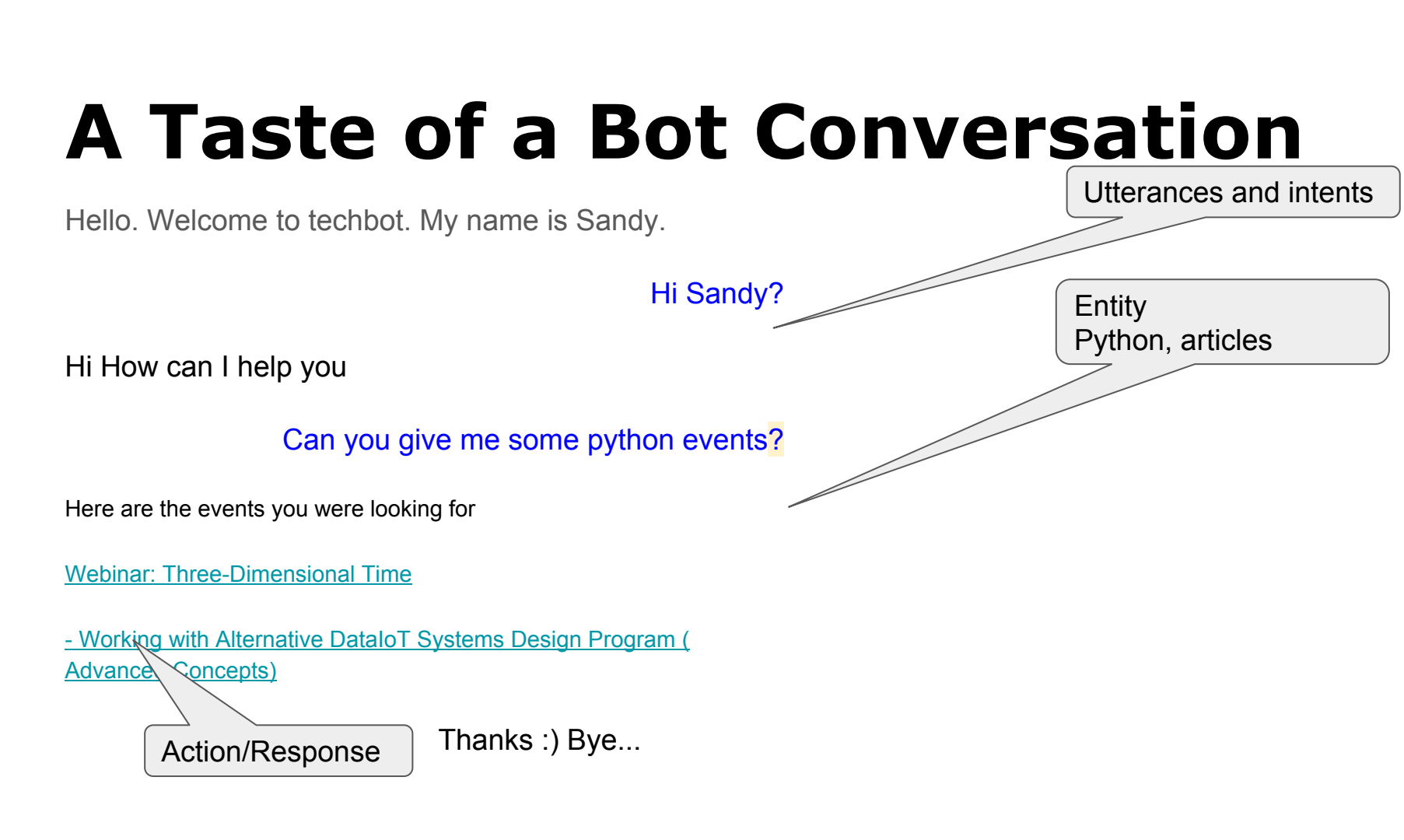
Why Do We Need Chatbots?

- There are several theories - one is that it is the future of interaction
- A single chatbot may replace several apps (especially in mobile)
- Take care of boring repetitive tasks
- The story of WeChat

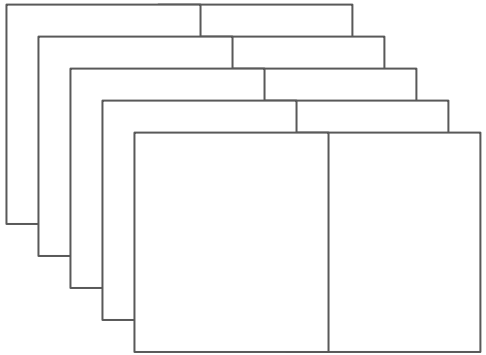
Some Examples of Chatbots

- Customer Service Bots
- A SiteBot - Helps navigate a website
- A teacher bot
- An Interview Bot
- Service Bots - Pizza ordering, travel booking
- More...

A Taste of a Bot Conversation



A Simplified Architecture



Chat Clients
- Webapp
- Facebook
- Slack etc.

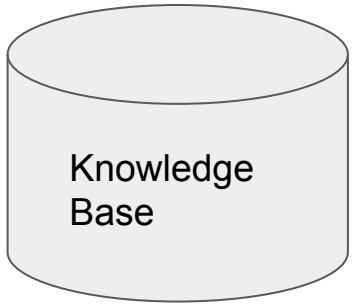
AI Engine
aka Agent

NLU,
NLG and
ML

Extracts intents
Extracts entities
Map intents to actions

Bot Server
App
Access KB
and interact
with AI
Engine

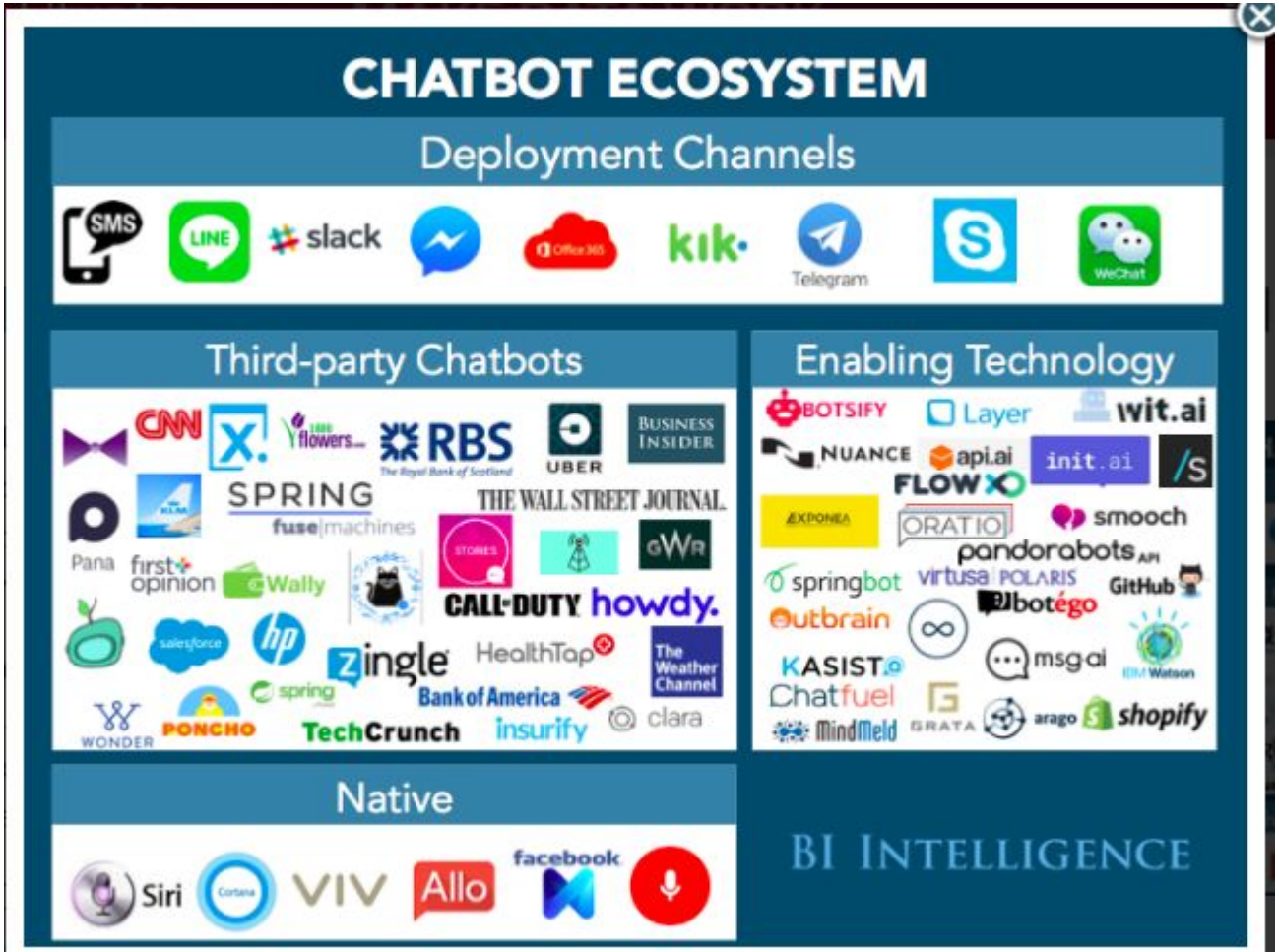
Executes Actions
Uses KB or other web
services



Contains all the
knowledge bot needs

Seven Steps to Building a Chatbot

1. Decides on an application area
2. Design conversations (aka interactions)
3. List intents, entities, actions, responses, contexts
4. Train AI Engine
5. Write Code for Actions
6. Create and update Knowledge Base
7. Test scenarios and incrementally improve



BI Intelligence

Questions?

Join our Slack Channel

buildingchatbots@gmail.com

Entity

What is an Entity?

An entity is a concept we want our personal assistant to understand when it is mentioned by the user in conversation. There are three types of entities in Api.ai:

System – entity types defined by Api.ai such as date, color, email, number and so on which Api.ai already understands. You can find a full list of these entities in Api.ai's documentation on System Entities.

- Developer – Entities which we create for our individual needs — these are what we will be focused on in this article.
- User – These are created for individual users while they use the assistant and can be generated by the Api.ai API to be used within a single session. We won't be covering these in this article but if there's enough reader interest, we might explore this in future!

Api.ai Terms and Concepts

API.AI is built on a number of concepts. It's good to understand what they are before trying anything hands-on.

Agents correspond to applications. Once you train and test an agent, you can integrate it with your app or device.

Entities represent concepts that are often specific to a domain as a way of mapping natural language phrases to canonical phrases that capture their meaning.

Intents represent a mapping between what a user says and what action should be taken by your software.

Actions correspond to the steps your application will take when specific intents are triggered by user inputs. An action may have parameters for specifying detailed information about it.

Contexts are strings that represent the current context of the user expression. This is useful for differentiating phrases which might be vague and have different meaning depending on what was spoken previously.