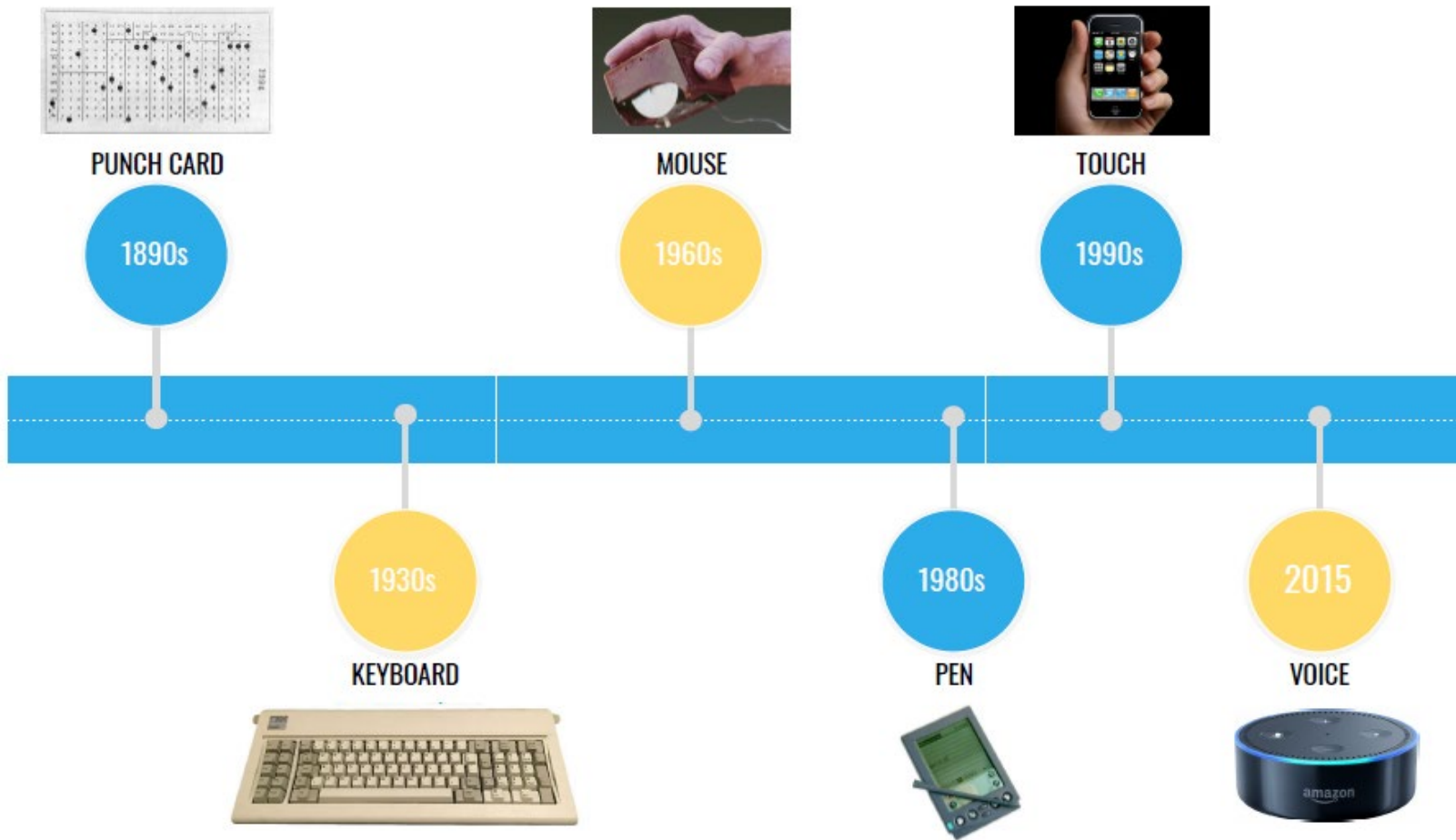


DialogFlow Bot

For Digital Currency

HCI Evolution Timeline

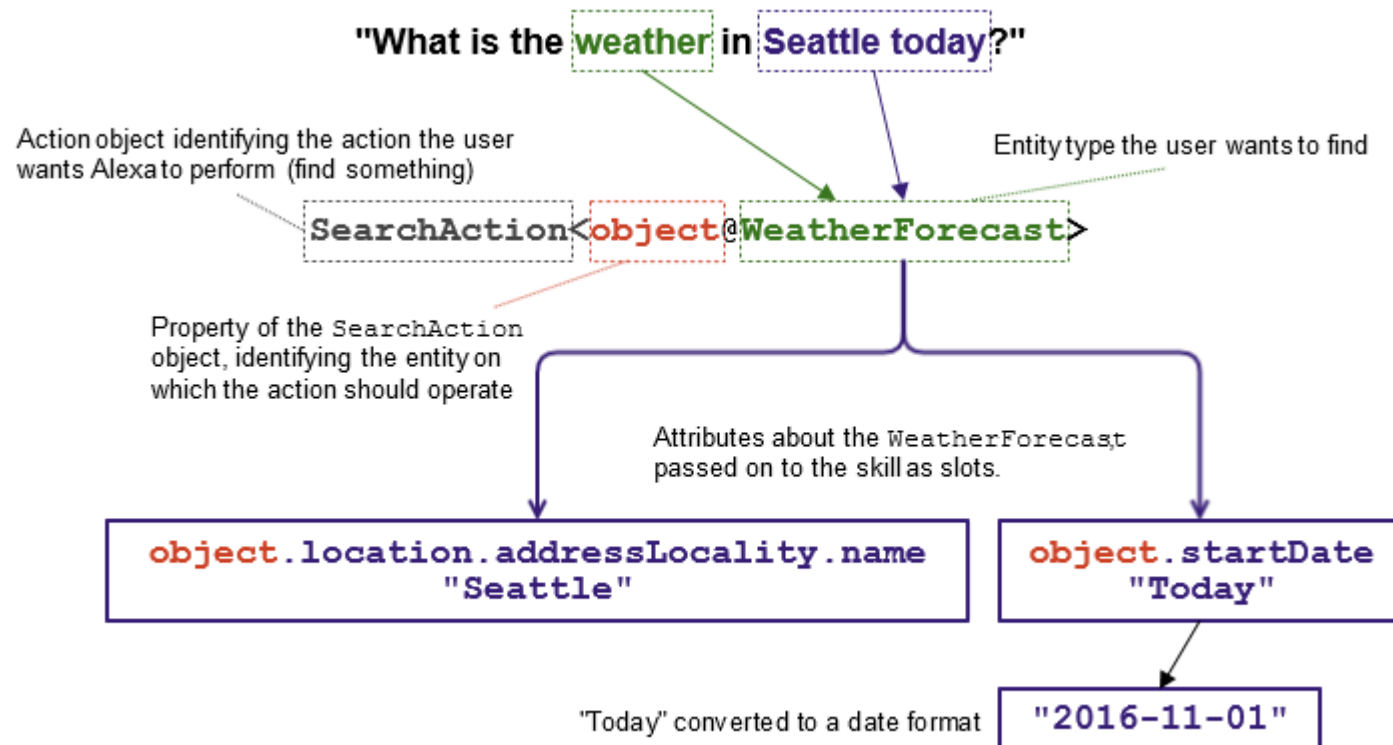


<https://www.youtube.com/watch?v=lwk6KzagxJo&feature=youtu.be>

Recapture

- To understand the request:

Intent represents an action that fulfills a user's spoken request. Intents can optionally have arguments called **slots**. The **sample utterances** are set of likely spoken phrases mapped to the intents.



Intent Components in Dialog AI

Intent Components

Contexts ?

Events ?

Training phrases ?

Action and parameters

Responses ?

Fulfillment ?

Context

- Contexts represent the current context of a user's request.
- This is helpful for differentiating phrases which may be vague or have different meanings depending on the user's preferences, geographic location, the current page in an app, or the topic of conversation.

Event

- Events allow you to **invoke an intent** based on a **non-verbal signal**, such as a **button click**, or the start of a new conversation with a bot.
- Events can be used by external services to **trigger** Dialog Flow intents, for example the Google Assistant built-in intents.

Parameters

Parameters allow people to specify **variable** parts of an utterance, for example city or date. Slots are commonly used in task and information-focused skills. There are two classes of slot types:

1. Amazon's **Built-in Entities**, like **@sys.firstname** and **@sys.date**
2. User's **Custom Entities**

Built-in Entities

@sys.location

@sys.number-integer

@sys.geo-county-us

@sys.age

@sys.temperature

@sys.geo-country

@sys.unit-weight-name

@sys.airport

@sys.time

@sys.unit-information-name

@sys.phone-number

@sys.music-genre

@sys.number-sequence

@sys.number

@sys.zip-code

@sys.given-name

@sys.last-name

@sys.unit-speed-name

Intent Components in Dialog AI

Intent Components

Contexts ?

Events ?

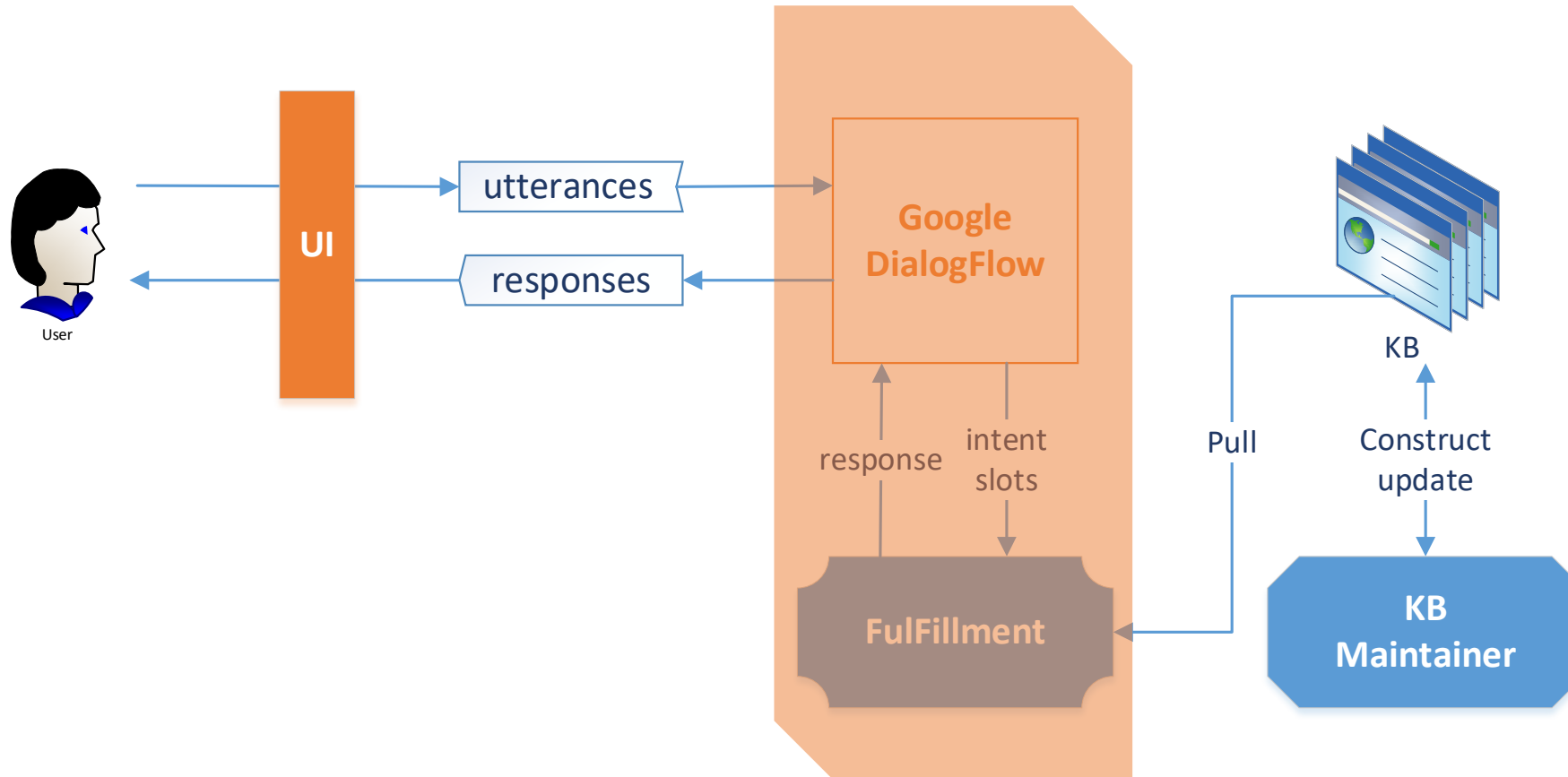
Training phrases ?

Action and parameters

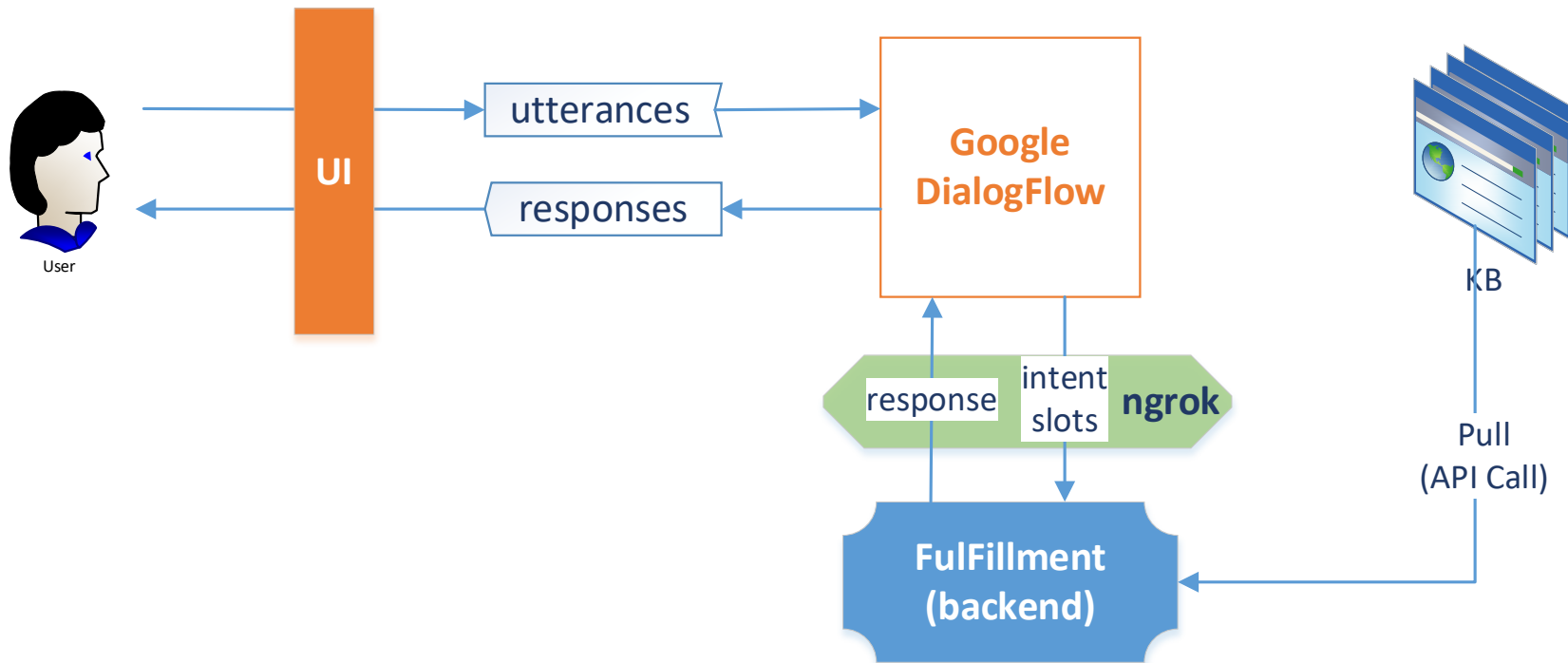
Responses ?

Fulfillment ?

Recapture



Target for Today



Preparation

- Google Dialog API account
- Download source code from Luminus
 - pip install
 - flask
 - Requests
- Download *ngrok.exe* or ngrok binary to your machine from ngrok.com
 - cd into the folder holding *ngrok.exe*
 - ngrok.exe http 5000

```
Select Anaconda Prompt - ngrok.exe http 5000
ngrok by @inconshreveable

Session Status      online
Account             aobo (Plan: Free)
Update              update available (version 2.3.35, Ctrl-U to update)
Version             2.3.18
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://d58ff39c.ngrok.io -> http://localhost:5000
Forwarding           https://d58ff39c.ngrok.io -> http://localhost:5000

Connections         ttl    opn    rt1    rt5    p50    p90
                   0      0      0.00   0.00   0.00   0.00
```

DialogFlow bot for Digital Currency

- Setup the backend
 - `dialogflow_python_backend` in python
 - `pip install`
 - flask
 - requests
- JSON API key has to be specified accordingly
 - <https://pro.coinmarketcap.com/>

The world's cryptocurrency data authority has a professional API made for you.

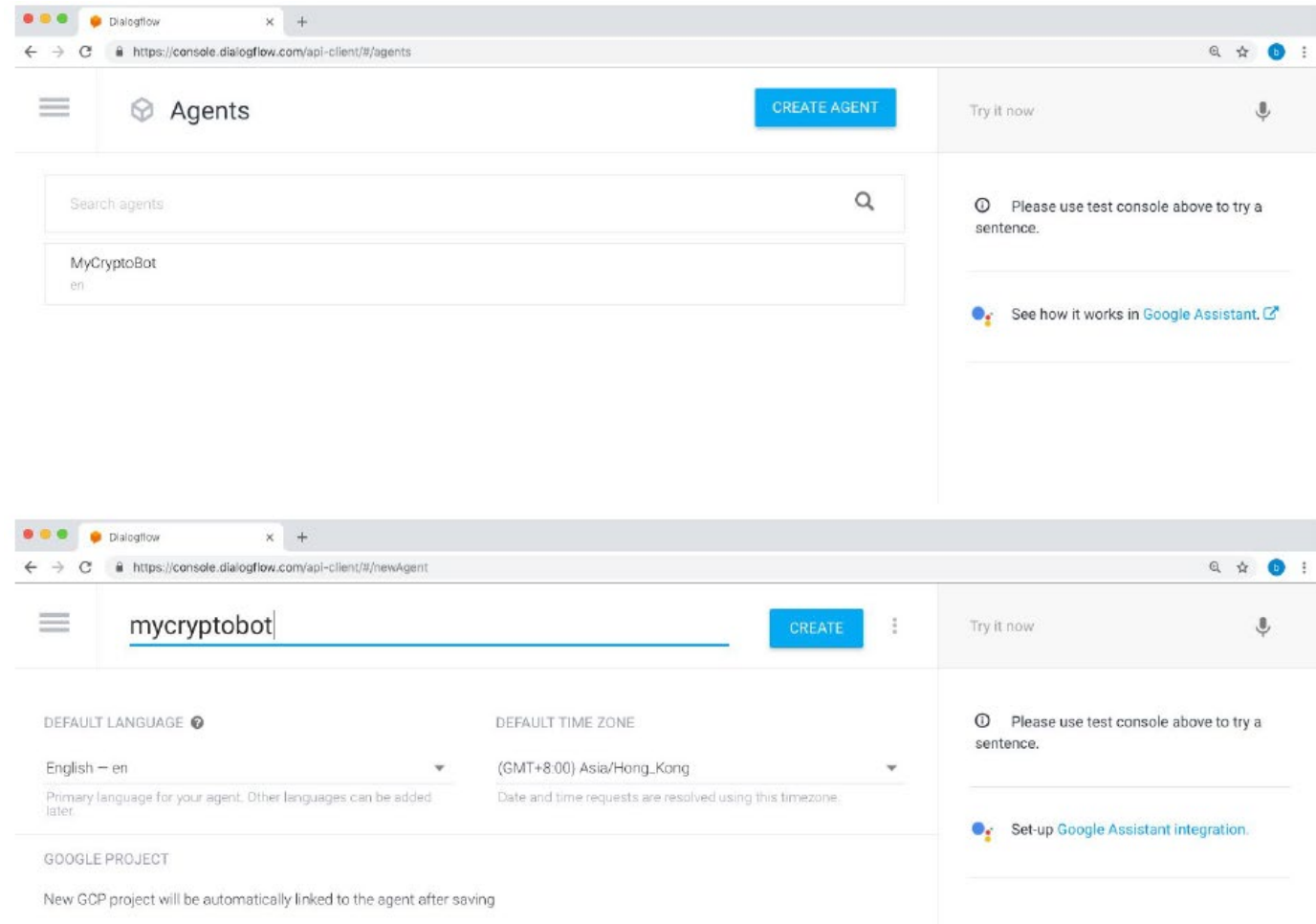
GET YOUR API KEY NOW

Demo DialogFlow Bot

For Digital Currency

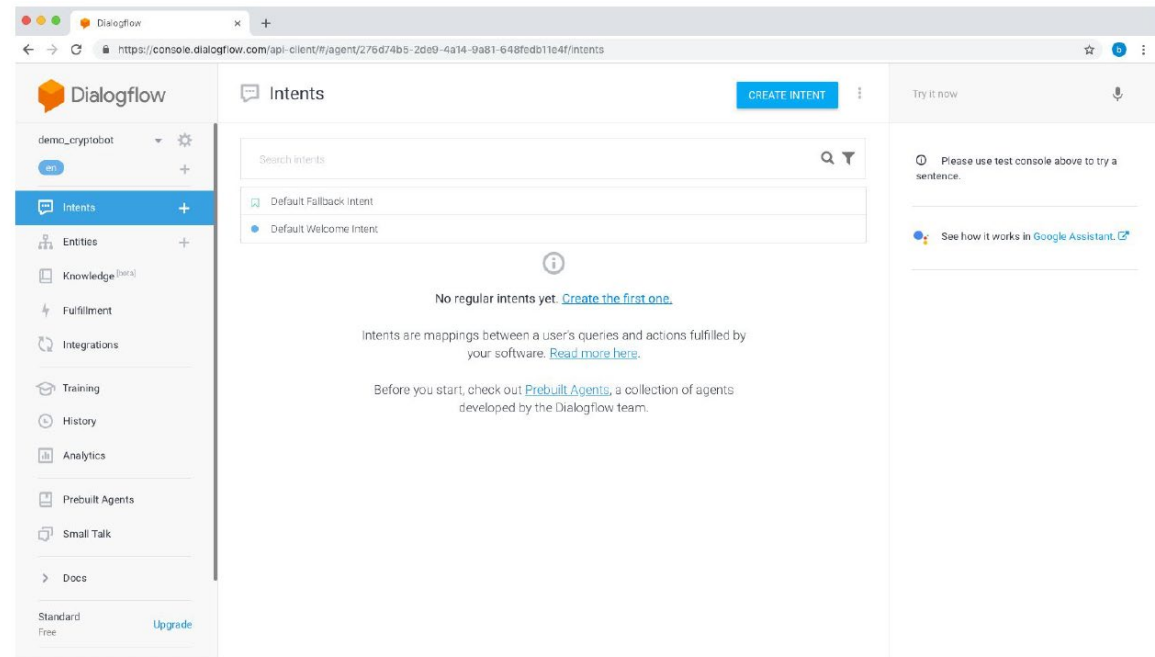
Let's create a simple DialogFlow bot

- Create an Agent



DialogFlow bot for Digital Currency


- Create an Intent
 - *The backend Fulfillment should recognize the name of intent*
 - *(in this case “GetPriceIntent”)*






Click Save to create the “GetPriceIntent”



DialogFlow bot for Digital Currency


- Add Training Utterances & Default Response


Training phrases 


Search training phrases  


 Add user expression



 price of bitcoin 

 what is the price of bitcoin

Responses 



DEFAULT 

Text response  

1

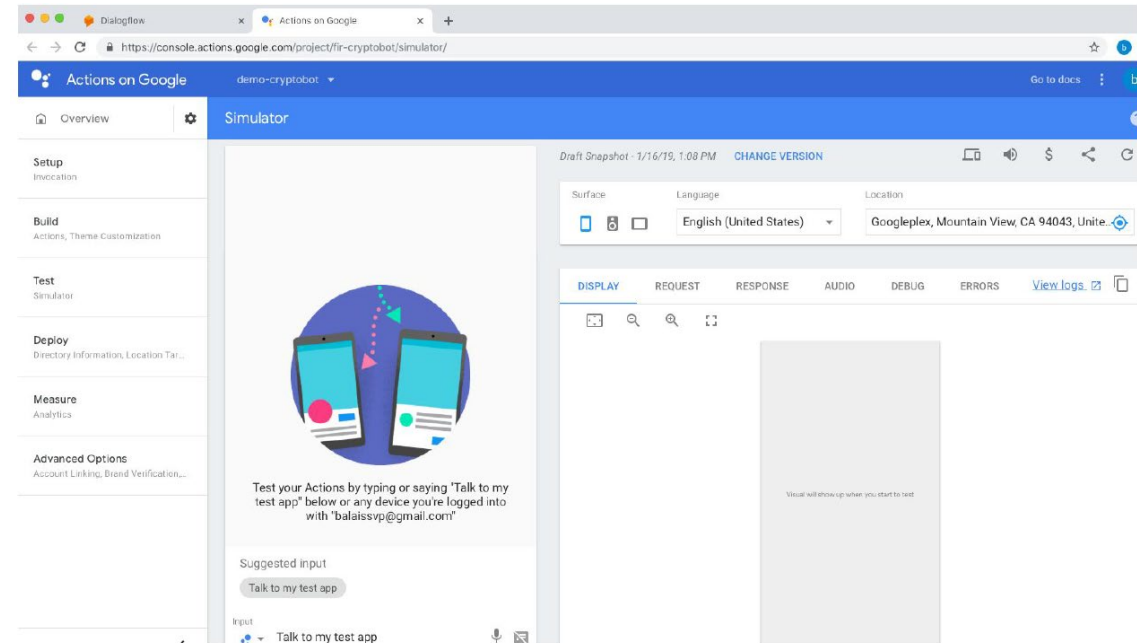
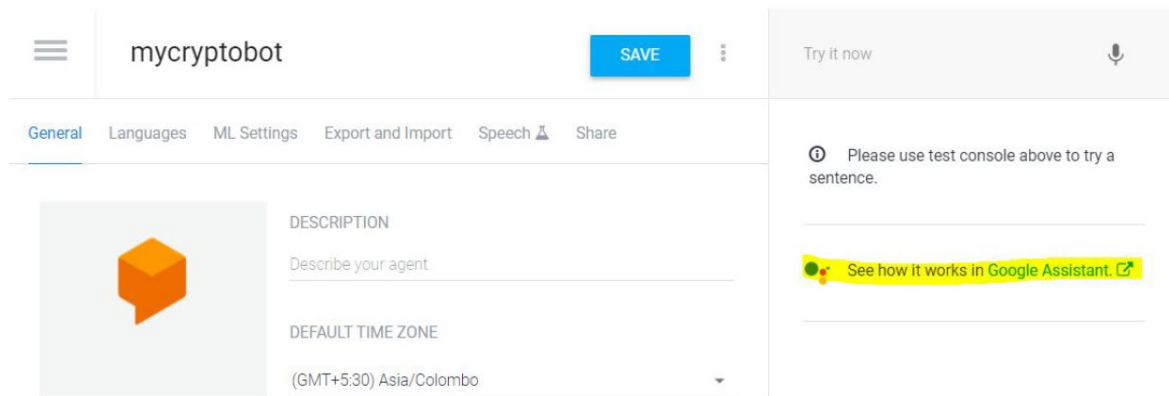
The price of bitcoin is \$3000

2

Enter a text response variant

DialogFlow bot for Digital Currency

- Test your prototype



NLU Pipeline

- Deterministic : *FST* to compile sample *utterances*
- Stochastic : Machine learning models for *entity*, *slot* and *intent* prediction

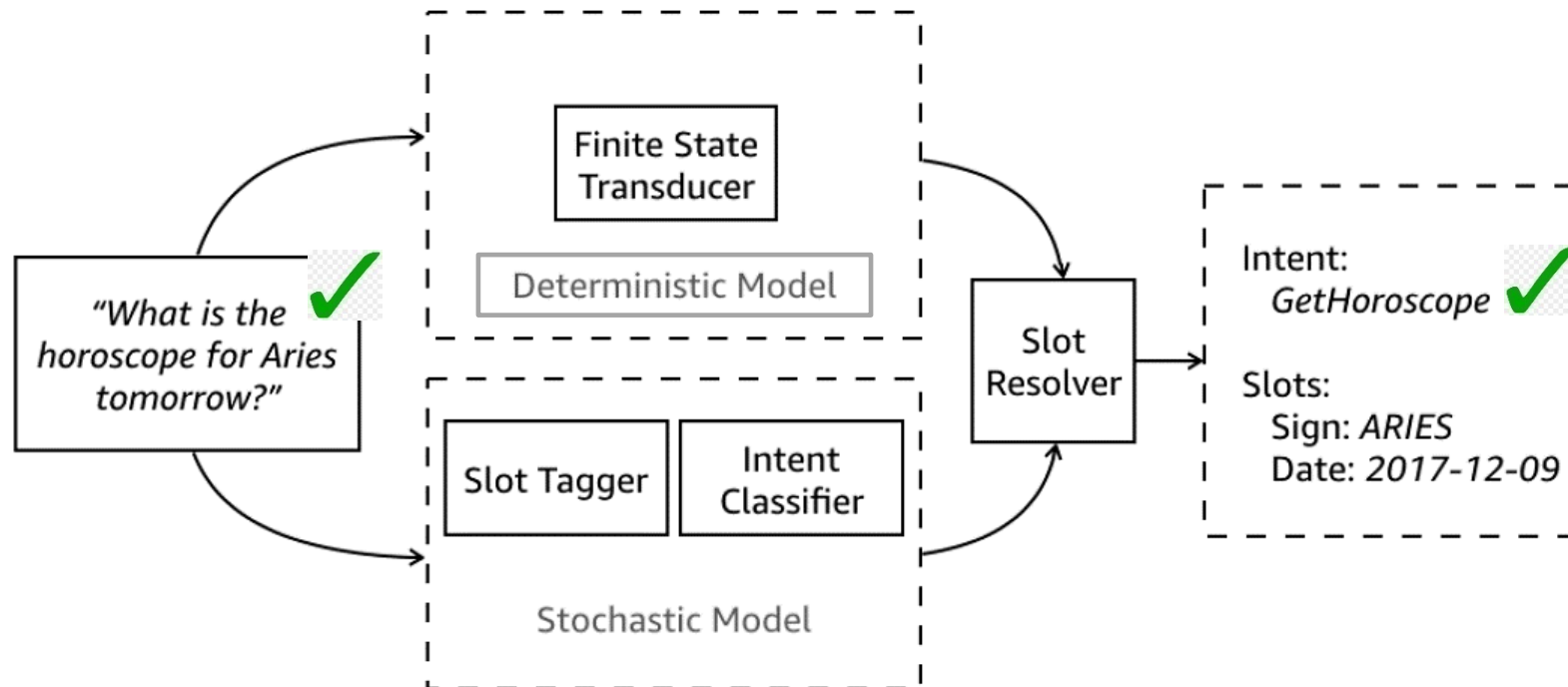
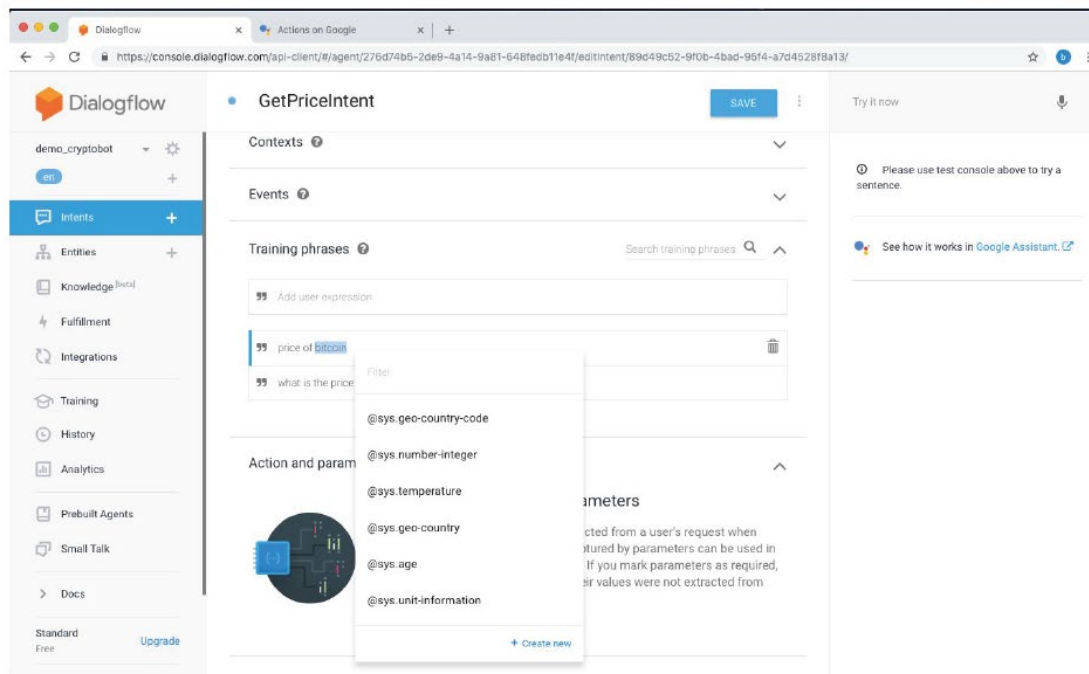


Image from 8. "Just ASK: building an architecture for extensible self-service spoken language understanding."

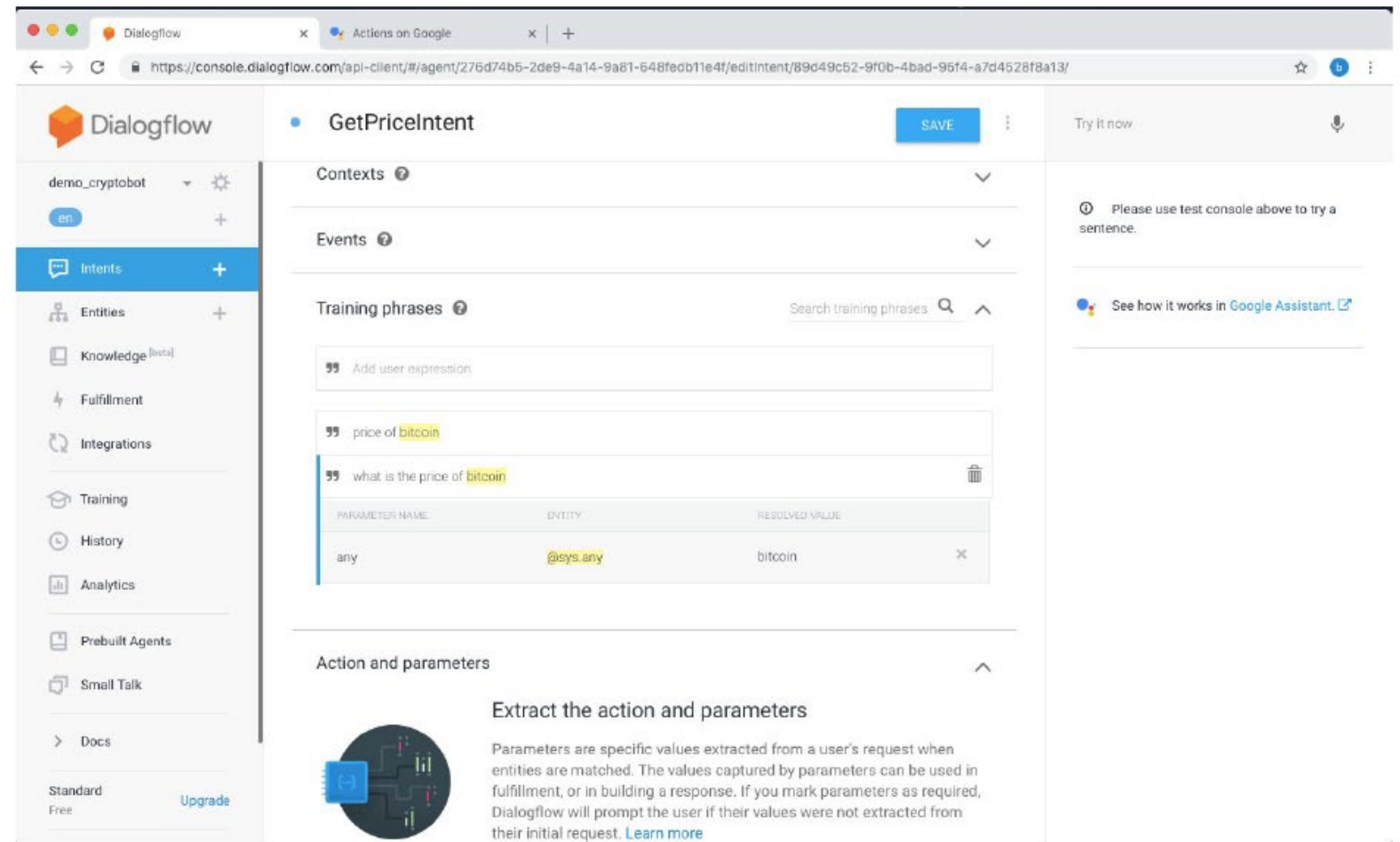
DialogFlow bot for Digital Currency

- Double Click on the portion “*bitcoin*” of Training phrase, to annotate with proper entities
- A pop-up list will appear as shown in the figure



DialogFlow bot for Digital Currency

- Bitcoin Name is not a build-in Entity
- Thus assign “sys.any”



The screenshot shows the DialogFlow console interface. On the left is a sidebar with navigation options: demo_cryptobot, Intents, Entities, Knowledge, Fulfillment, Integrations, Training, History, Analytics, Prebuilt Agents, Small Talk, Docs, and a Standard Free plan with an Upgrade button. The main area is titled 'GetPriceIntent' and includes sections for Contexts, Events, Training phrases, and Action and parameters. The Training phrases section contains two phrases: 'price of bitcoin' and 'what is the price of bitcoin'. Below these is a table showing the parameter extraction for the second phrase.

PARAMETER NAME	ENTITY	RESOLVED VALUE
any	@sys.any	bitcoin

The 'Action and parameters' section at the bottom explains that parameters are specific values extracted from a user's request when entities are matched, and provides a link to learn more.

DialogFlow bot for Digital Currency

- Rename the PARAMETER NAME from to ***coinname*** .
- This ***coinname*** slot should be recognized by backend

Training phrases ?

Search training phrases 🔍 ^

” Add user expression

” how much is litecoin

” pride of bitcoin

PARAMETER NAME

ENTITY

RESOLVED VALUE

coinname

@sys.any

bitcoin



DialogFlow bot for Digital Currency

- Enable the Webhook call for this intent

Responses ?

^

DEFAULT +

Text Response

🗑

1

bad answer

2

Enter a text response variant

⬆⬇⬆

ADD RESPONSES

☐

Set this intent as end of conversation ?


Fulfillment ?


▼

DialogFlow bot for Digital Currency


- Enable the Webhook call for this intent


DEFAULT +


Text Response 

1	bad answer
2	Enter a text response variant 

ADD RESPONSES

☐ Set this intent as end of conversation 


Fulfillment 



Call a web service to connect your back-end

Send intent, parameters and context to your Cloud function or a web service. Execute necessary logic and respond back with a written, spoken or visual response. [Learn more](#)

ENABLE FULFILLMENT



DialogFlow bot for Digital Currency

- Enable the Webhook call for this intent
- SAVE

• GetPriceIntent ? SAVE

Contexts ? ▼

Events ? ▼

Training phrases ? ▼

Action and parameters ▼

Responses ? ▼

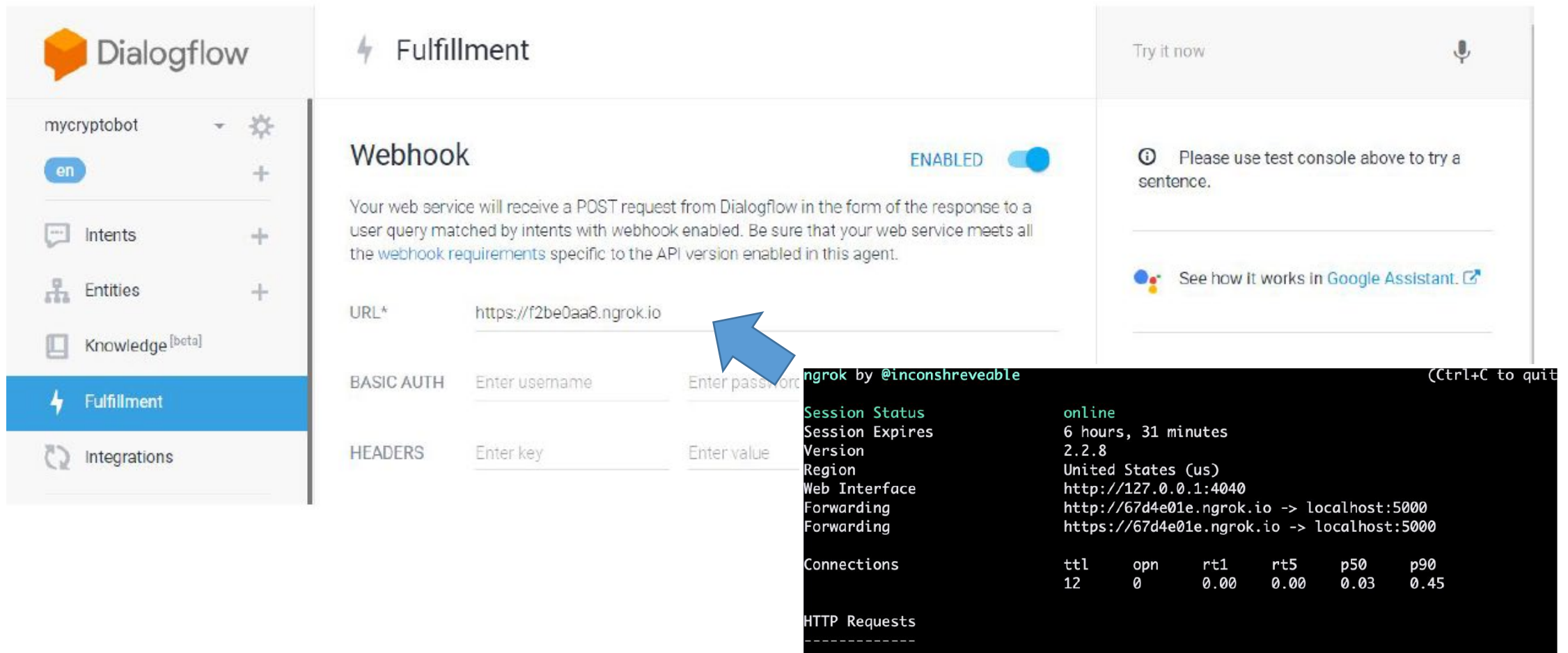
Fulfillment ? ^

☒ Enable webhook call for this intent

☐ Enable webhook call for slot filling

DialogFlow bot for Digital Currency

- Enable WebHook Fulfillment and copy the HTTPs// url



The image shows the Dialogflow Fulfillment settings page. On the left, a sidebar contains the Dialogflow logo and navigation links: mycryptobot, en, Intents, Entities, Knowledge [beta], Fulfillment (highlighted with a blue arrow), and Integrations. The main content area is titled 'Fulfillment' and features a 'Webhook' section with an 'ENABLED' toggle. Below this, there are fields for 'URL*' (https://f2be0aa8.ngrok.io), 'BASIC AUTH' (username and password fields), and 'HEADERS' (key and value fields). A blue arrow points from the 'URL*' field to a terminal window. The terminal window displays the output of the ngrok command, showing session status, session expiration, version, region, web interface, forwarding URLs, and connection statistics.

Dialogflow Fulfillment Settings:

- Webhook:** ENABLED
- URL*:** https://f2be0aa8.ngrok.io
- BASIC AUTH:** Enter username, Enter password
- HEADERS:** Enter key, Enter value

ngrok by @inconshreveable (Ctrl+C to quit)

```
Session Status      online
Session Expires    6 hours, 31 minutes
Version             2.2.8
Region             United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://67d4e01e.ngrok.io -> localhost:5000
Forwarding           https://67d4e01e.ngrok.io -> localhost:5000

Connections         ttl      opn      rt1      rt5      p50      p90
                  12       0       0.00    0.00    0.03    0.45

HTTP Requests
-----
```

Target

