Hospitality of Chatbot Building Platforms

Saurabh Srivastava, T.V. Prabhakar

Department of Computer Science & Engineering, IIT Kanpur, India

International Workshop on Software Qualities and Their Dependencies (SQUADE)

27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering
Tallinn, Estonia

August 26, 2019

What lies ahead?

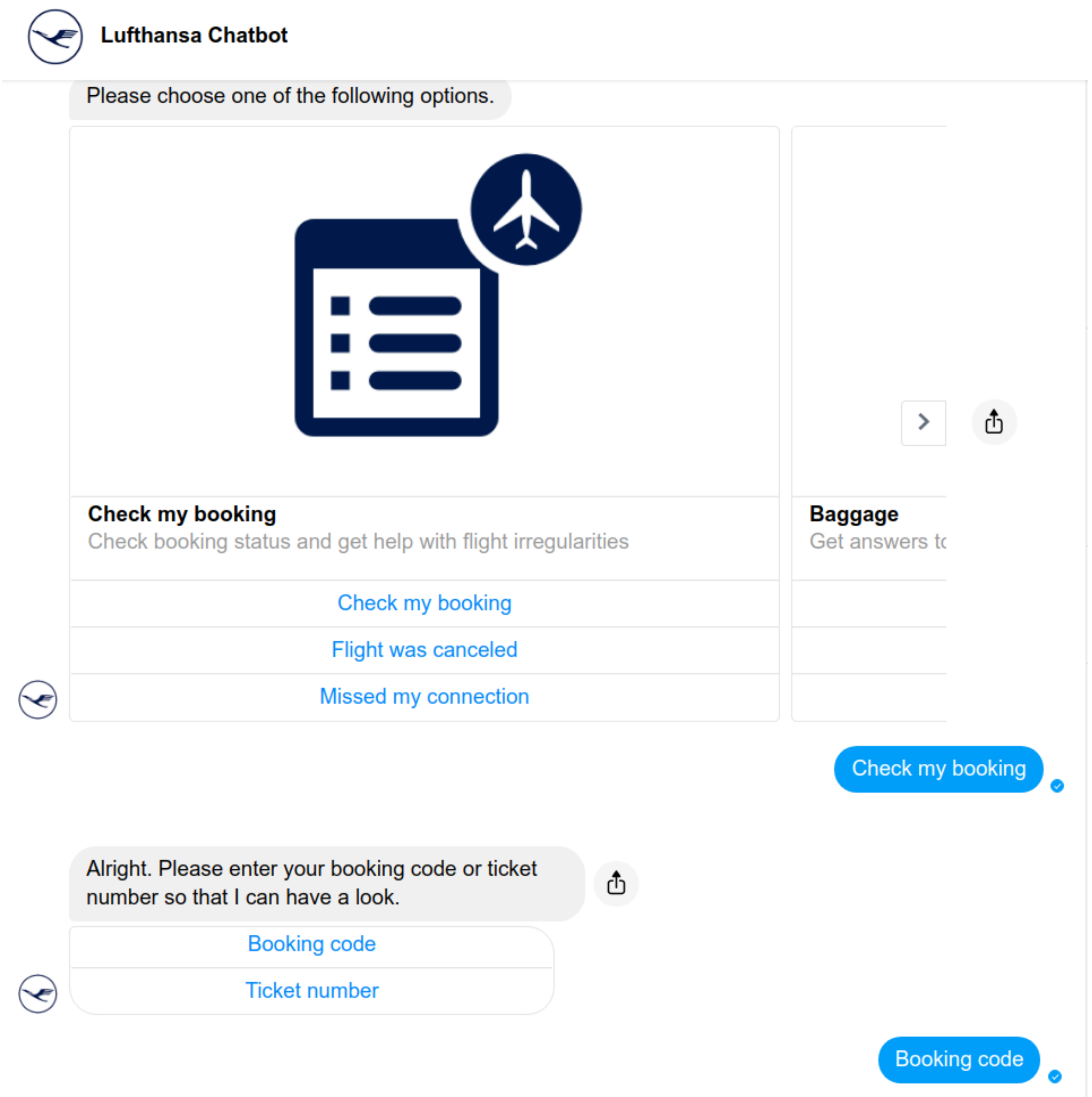
- Motivation
  - Chatbots
  - Chatbot building Platforms
  - Quality Attributes and Architectural Tactics
- The Hospitality Framework
  - Phases of the framework
  - The Hospitality Metric
  - Results
- Discussion

Introduction to Chatbots and Quality Attributes

Part 1

Chatbots – What are they?

- It is a *colloquial* term used to refer to a class of software components, which can interact with users using Natural Languages
- The communication medium can be *text* or *speech*
- The communication could be *flexible* or *constrained*
- The bot (shortened term for "chatbot") could be an *independent* component or *part* of a larger application
- A chatbot can itself be divided in multiple sub-components
    - In the present work, when we say "chatbot", we actually mean "chatbot core"
- Examples – Google Assistant, FB Messenger Bots, E-commerce Bots



Lufthansa Chatbot

Please choose one of the following options.

Check my booking
Check booking status and get help with flight irregularities

Check my booking
Flight was canceled
Missed my connection

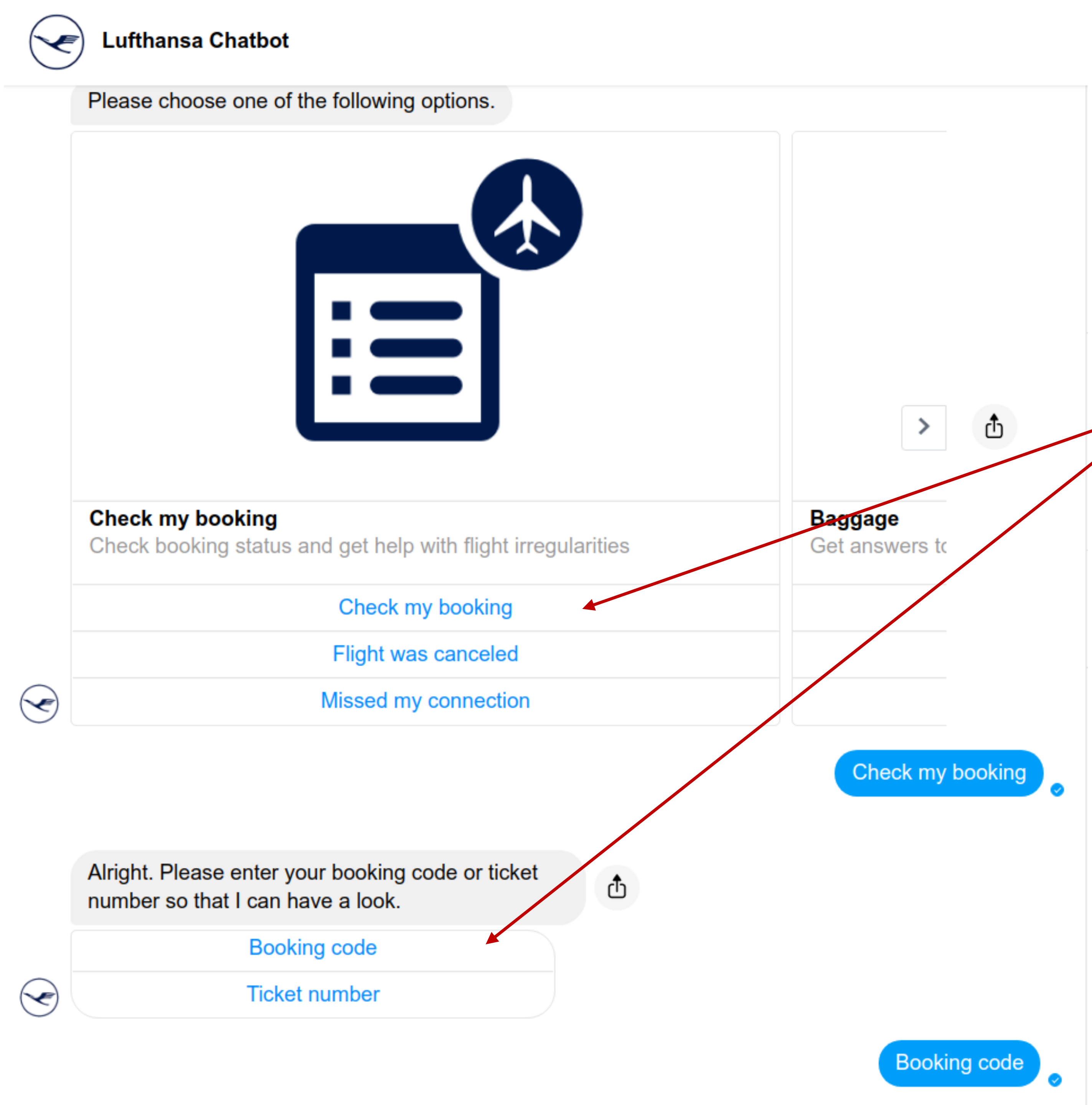
Baggage
Get answers t

Check my booking

Alright. Please enter your booking code or ticket number so that I can have a look.

Booking code
Ticket number

Booking code


Lufthansa Chatbot

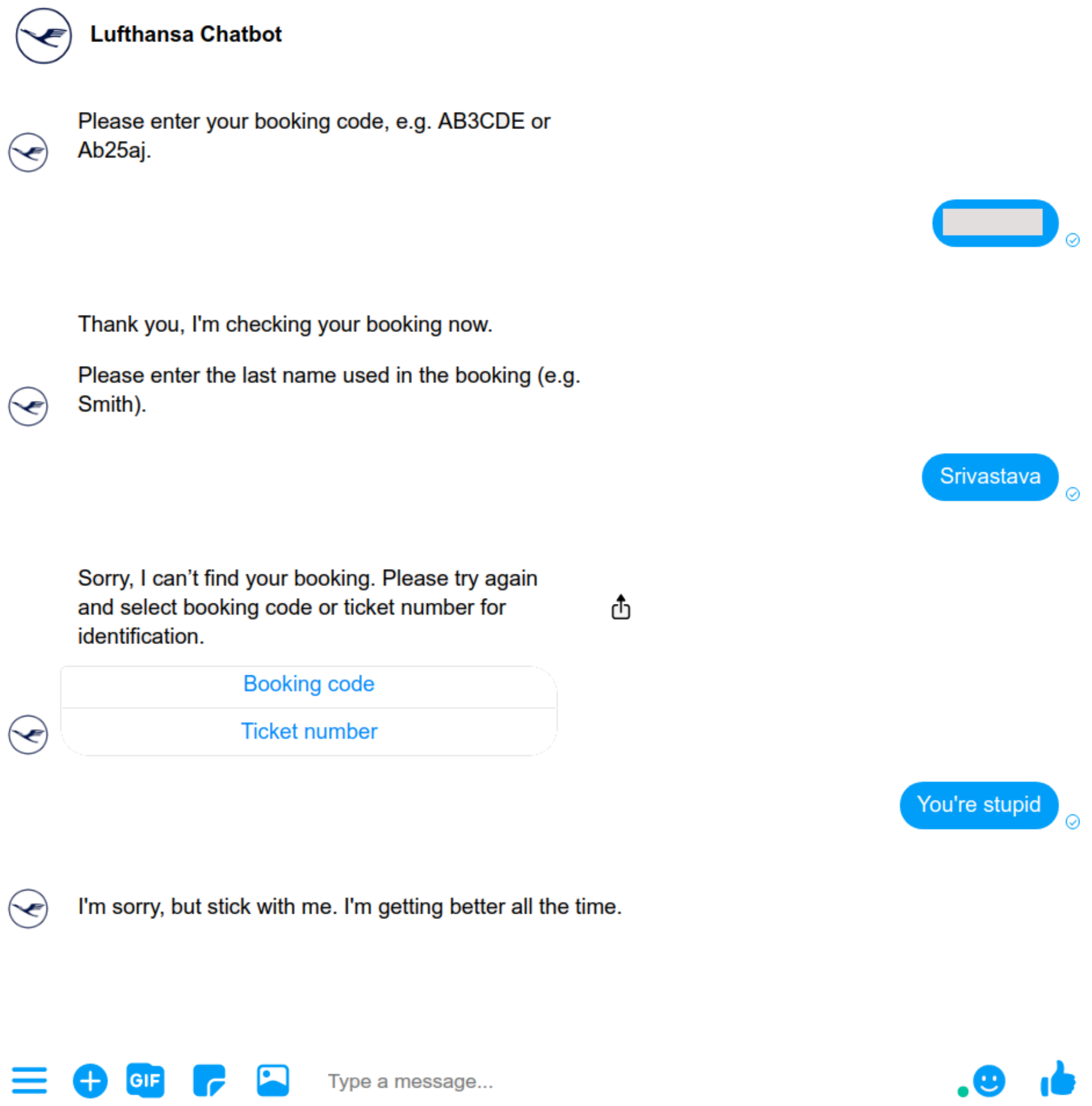
Please choose one of the following options.

Check my booking

Alright. Please enter your booking code or ticket number so that I can have a look.

Booking code

Examples of a *constrained* interface


Lufthansa Chatbot

Please enter your booking code, e.g. AB3CDE or Ab25aj.

[redacted]

Thank you, I'm checking your booking now.

Please enter the last name used in the booking (e.g. Smith).

Srivastava

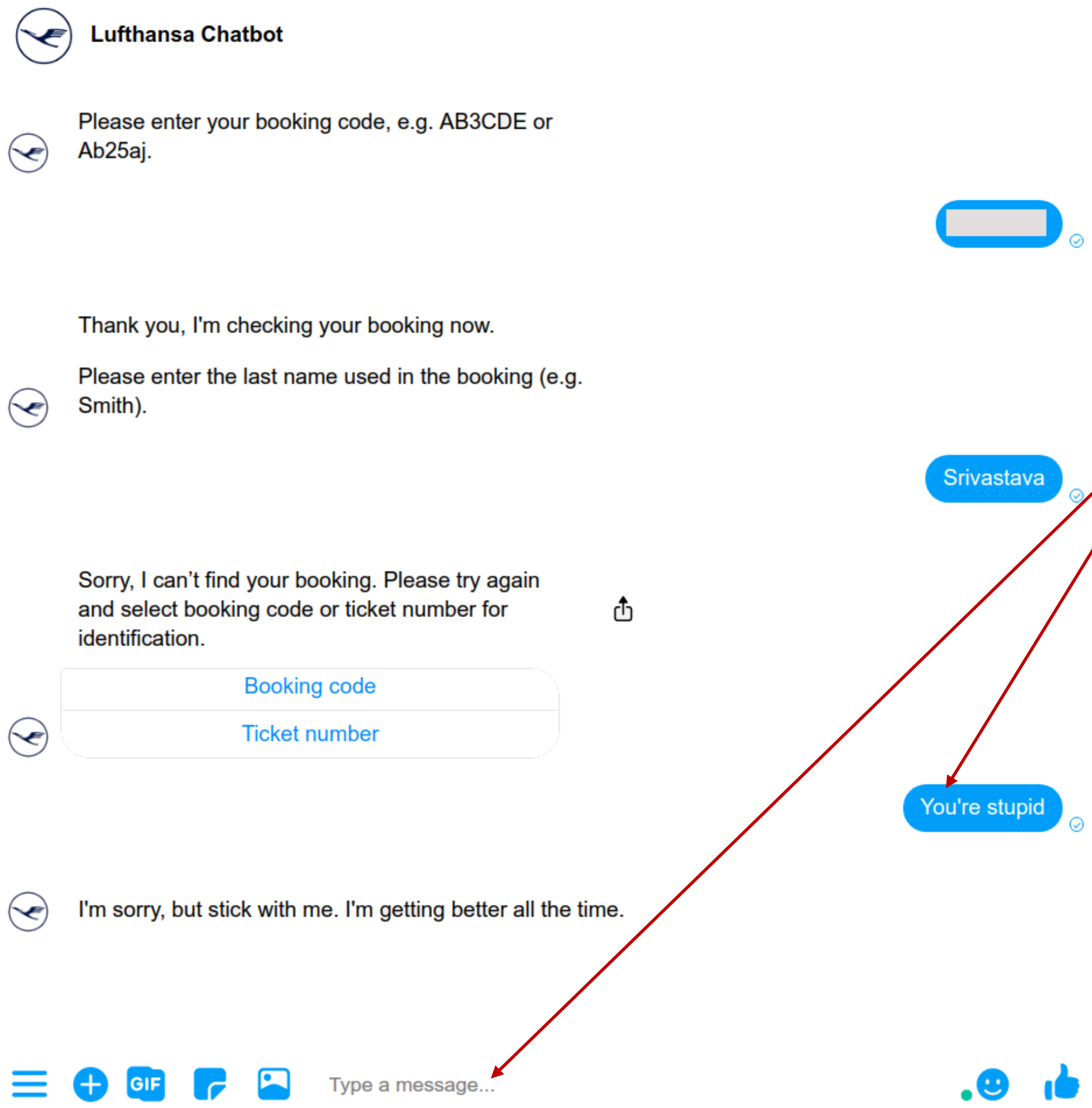
Sorry, I can’t find your booking. Please try again and select booking code or ticket number for identification.

Booking code

Ticket number

You're stupid

I'm sorry, but stick with me. I'm getting better all the time.



Lufthansa Chatbot

Please enter your booking code, e.g. AB3CDE or Ab25aj.

[redacted]

Thank you, I'm checking your booking now.

Please enter the last name used in the booking (e.g. Smith).

Srivastava

Sorry, I can’t find your booking. Please try again and select booking code or ticket number for identification.

Booking code

Ticket number

You're stupid

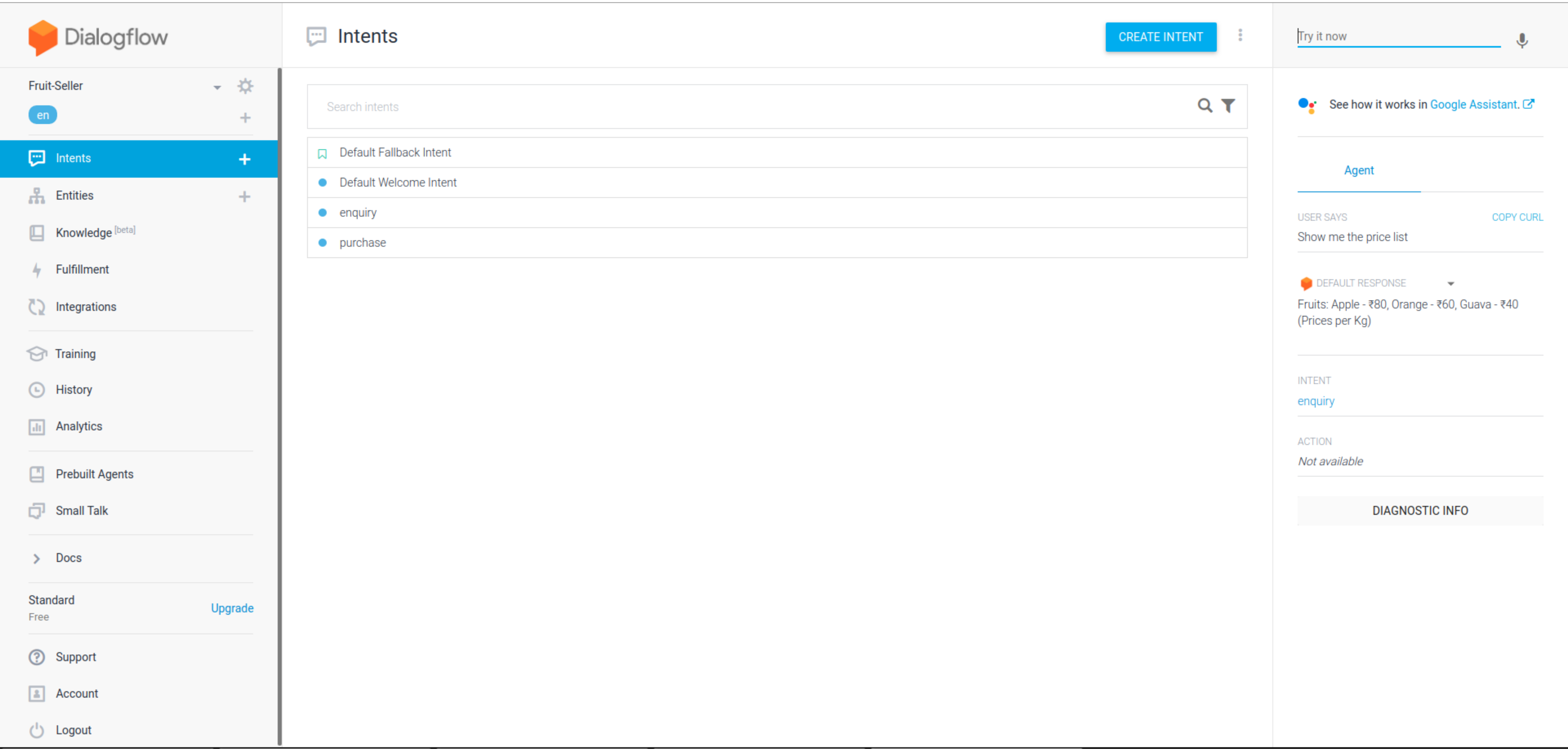
I'm sorry, but stick with me. I'm getting better all the time.

Type a message...

Example of a *flexible* interface

Chatbot building Platfoms – How do they help?

- There are a number of commercial platforms available today, which can aid in development and deployment of chatbots
- These platforms provide a set of features which are useful in
    - Defining the types of queries the bot will cater to (aka <u>Intents</u>)
    - Providing details of each type, including specific pieces of information that the user will supply (often called <u>Entities</u> or Parameters)
    - Supplying response templates or callbacks (usually known as <u>Fulfilments</u>)
    - Orchestrating conversational flow, as close as possible, to a conversation between two human beings (we refer to it as <u>Flow Management</u>)
- Examples – Google Dialogflow, IBM Watson Assistant, Amazon Lex



Dialogflow

Fruit-Seller

en

Intents

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

History

Analytics

Prebuilt Agents

Small Talk

Docs

Standard
Free

Upgrade

Support

Account

Logout

Intents

CREATE INTENT

Search intents

- Default Fallback Intent
- Default Welcome Intent
- enquiry
- purchase

Try it now

See how it works in Google Assistant.

Agent

USER SAYS

COPY CURL

Show me the price list

DEFAULT RESPONSE

Fruits: Apple - ₹80, Orange - ₹60, Guava - ₹40 (Prices per Kg)

INTENT

enquiry

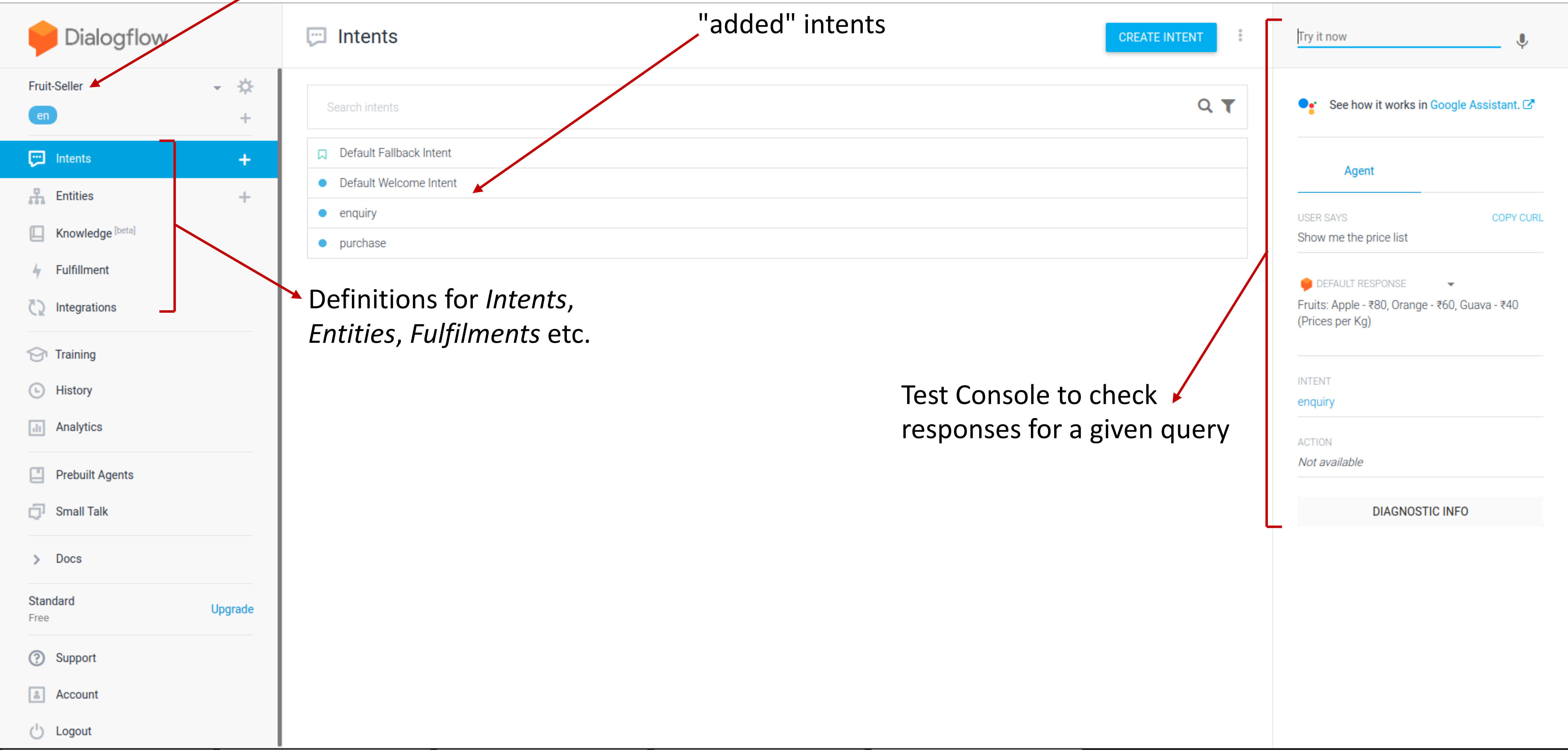
ACTION

Not available

DIAGNOSTIC INFO

Name of the Chatbot

List of defined and "added" intents


Dialogflow

Fruit-Seller

en

Intents

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

History

Analytics

Prebuilt Agents

Small Talk

Docs

Standard
Free

Upgrade

Support

Account

Logout

Intents

CREATE INTENT

Search intents

- Default Fallback Intent
- Default Welcome Intent
- enquiry
- purchase

Try it now

See how it works in Google Assistant.

Agent

USER SAYS
Show me the price list

COPY CURL

DEFAULT RESPONSE
Fruits: Apple - ₹80, Orange - ₹60, Guava - ₹40 (Prices per Kg)

INTENT
enquiry

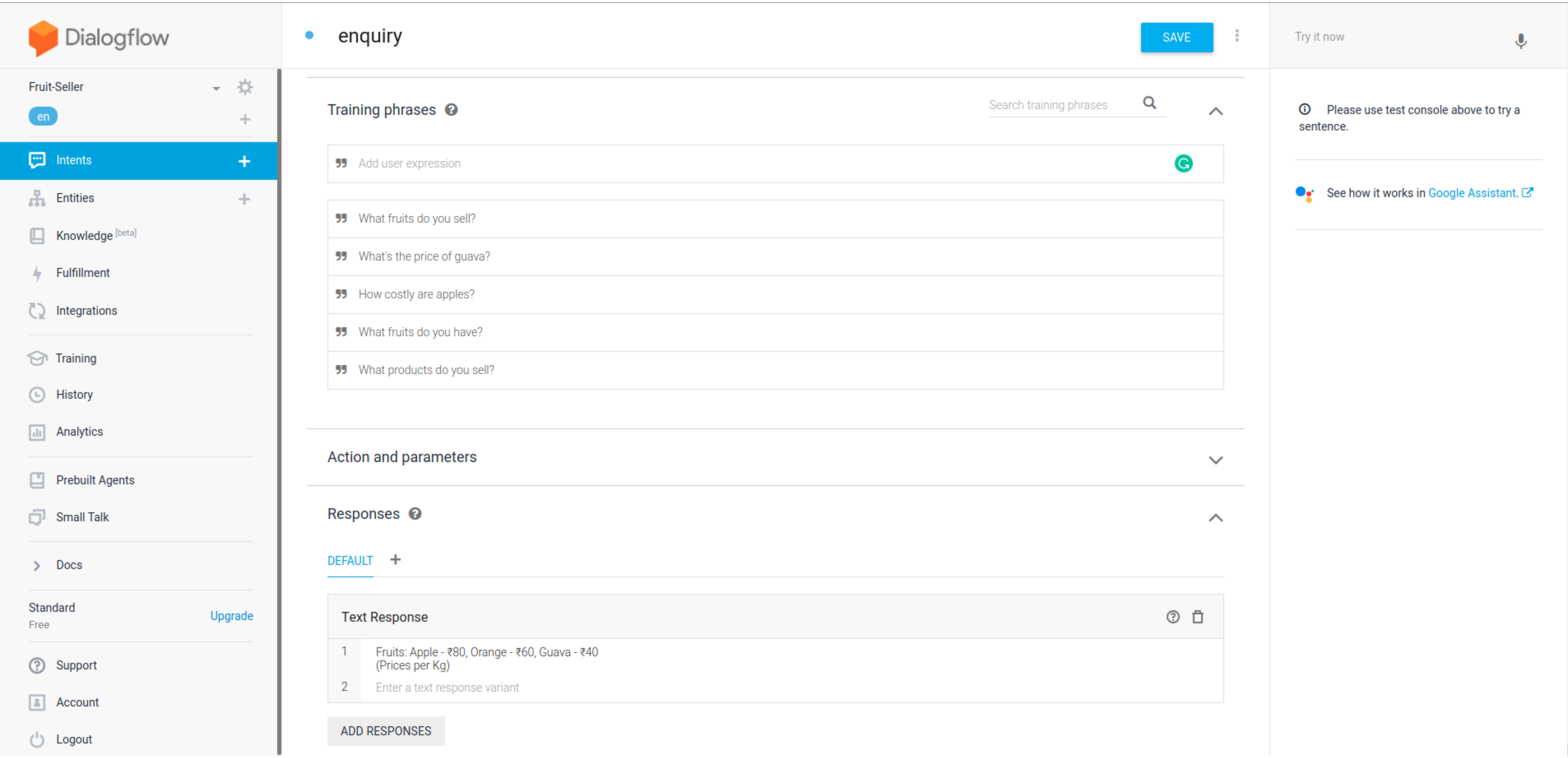
ACTION
Not available

DIAGNOSTIC INFO

Definitions for *Intents*, *Entities*, *Fulfilments* etc.

Test Console to check responses for a given query

Details of an Intent



enquiry

Training phrases

What fruits do you sell?

What's the price of guava?

How costly are apples?

What fruits do you have?

What products do you sell?

Action and parameters

Responses

Text Response

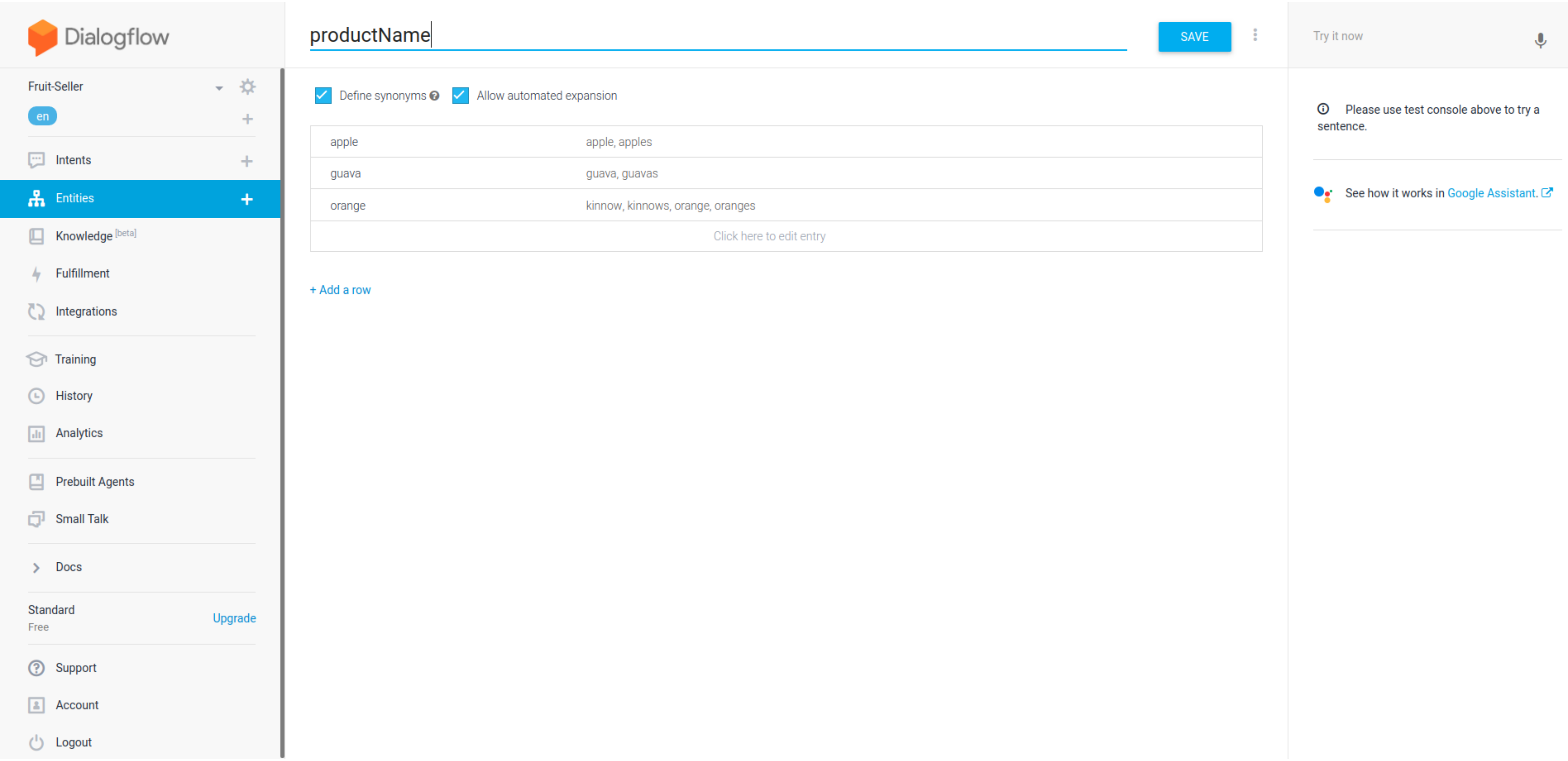
1 Fruits: Apple - ₹80, Orange - ₹60, Guava - ₹40
(Prices per Kg)

2 Enter a text response variant

Please use test console above to try a sentence.

See how it works in Google Assistant.

Detailes of an Entity



productName

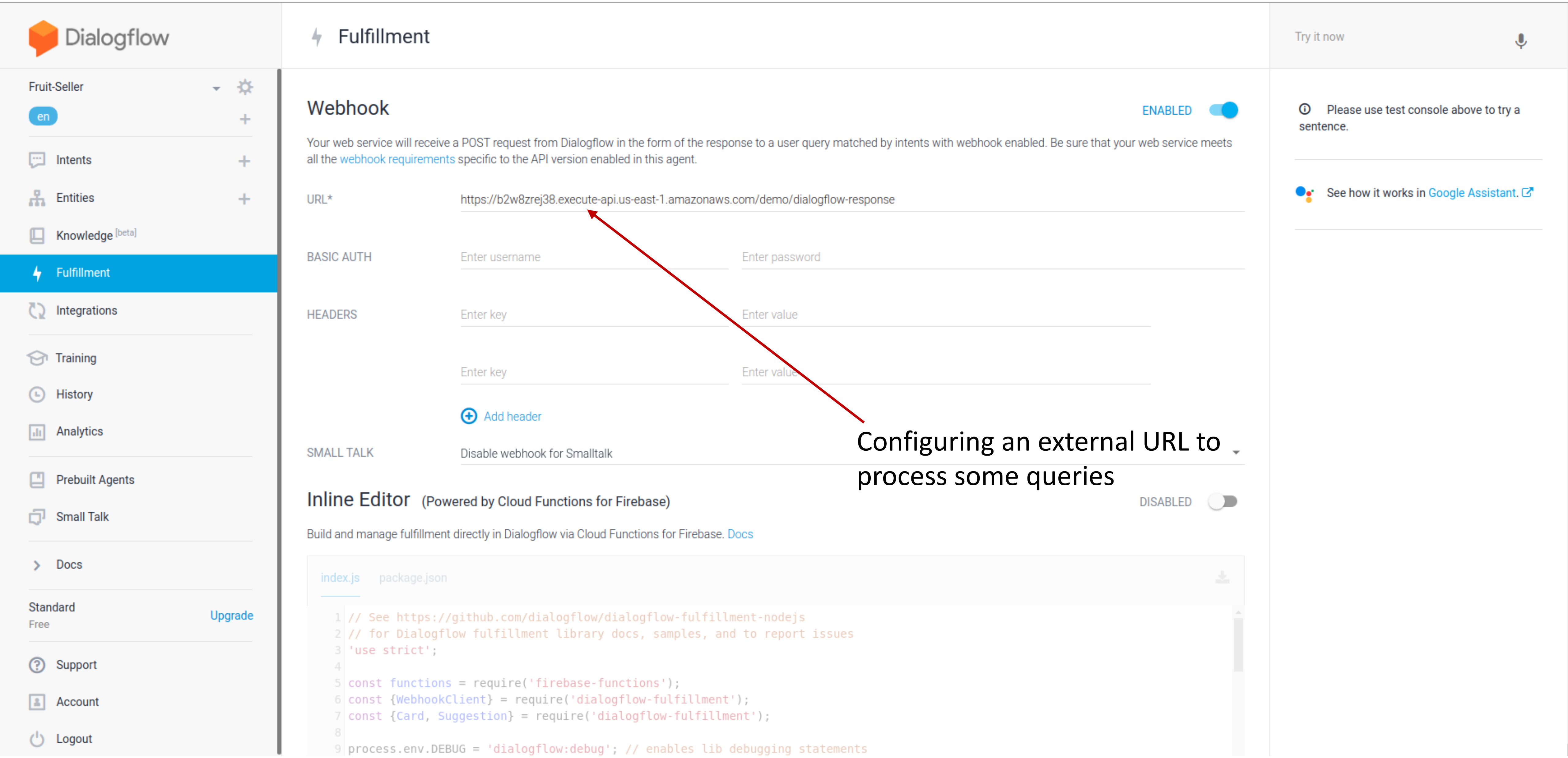
☑ Define synonyms ☑ Allow automated expansion

apple	apple, apples
guava	guava, guavas
orange	kinnow, kinnows, orange, oranges
	Click here to edit entry

+ Add a row

Please use test console above to try a sentence.

See how it works in Google Assistant.



Fulfillment

Webhook

ENABLED

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the webhook requirements specific to the API version enabled in this agent.

URL*	https://b2w8zrej38.execute-api.us-east-1.amazonaws.com/demo/dialogflow-response	
BASIC AUTH	Enter username	Enter password
HEADERS	Enter key	Enter value
	Enter key	Enter value
	Add header	
SMALL TALK	Disable webhook for Smalltalk	

Configuring an external URL to process some queries

Inline Editor (Powered by Cloud Functions for Firebase)

DISABLED

Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. Docs

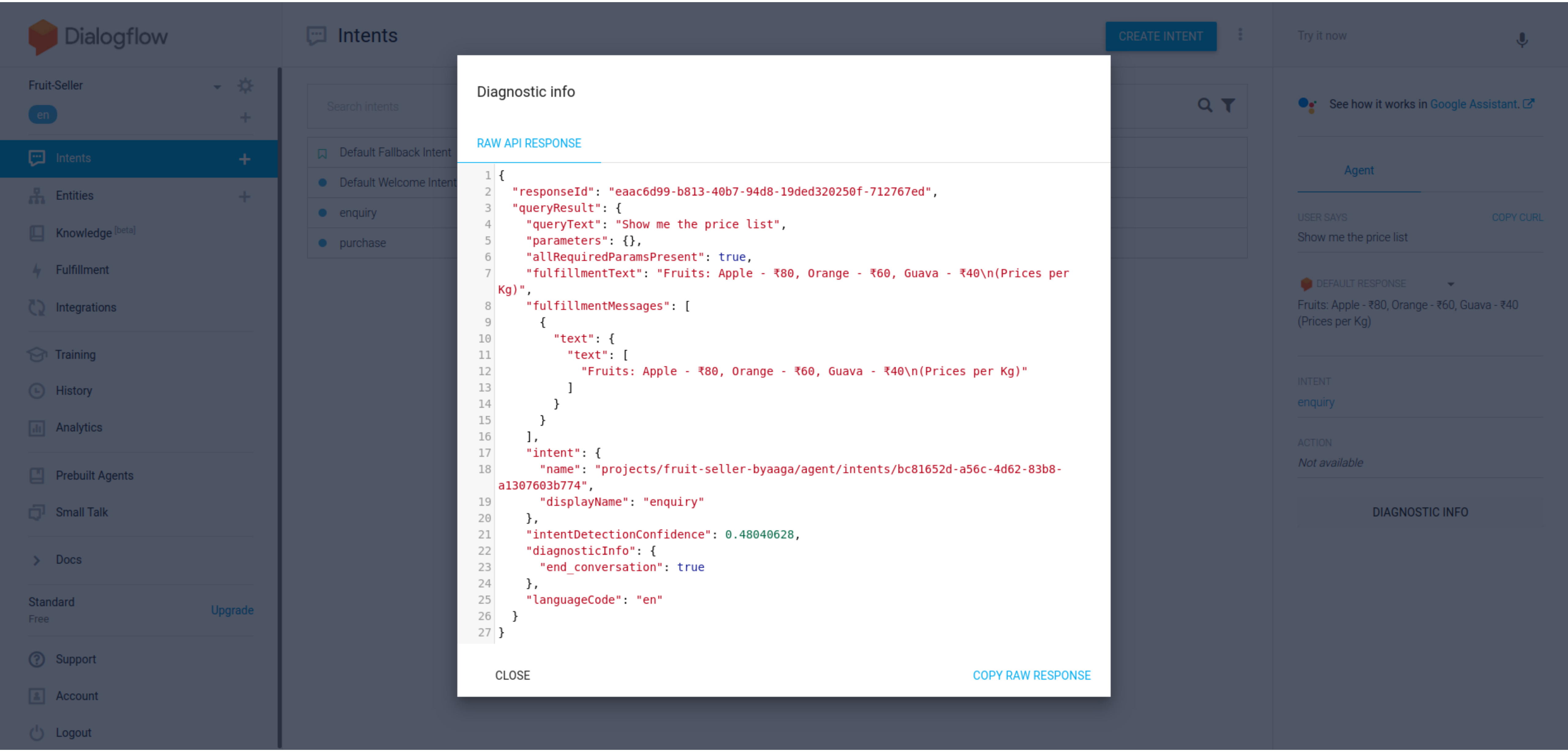
index.js package.json

```
// See https://github.com/dialogflow/dialogflow-fulfillment-nodejs
// for Dialogflow fulfillment library docs, samples, and to report issues
'use strict';

const functions = require('firebase-functions');
const {WebhookClient} = require('dialogflow-fulfillment');
const {Card, Suggestion} = require('dialogflow-fulfillment');

process.env.DEBUG = 'dialogflow:debug'; // enables lib debugging statements
```

Please use test console above to try a sentence.

See how it works in Google Assistant.

Detailed Diagnostic Information for a sample query

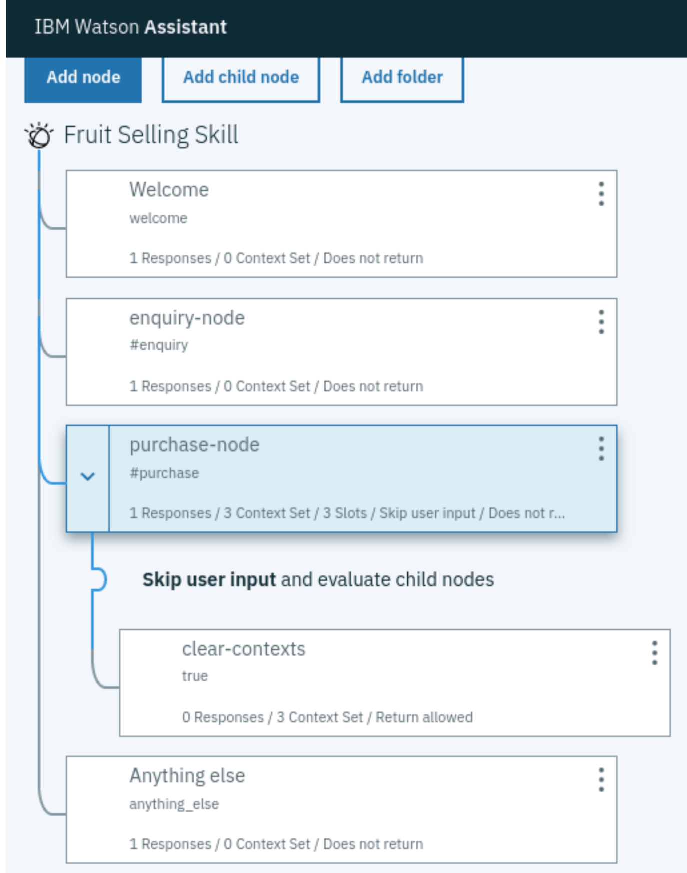


Diagnostic info

RAW API RESPONSE

```
{
  "responseId": "eaac6d99-b813-40b7-94d8-19ded320250f-712767ed",
  "queryResult": {
    "queryText": "Show me the price list",
    "parameters": {},
    "allRequiredParamsPresent": true,
    "fulfillmentText": "Fruits: Apple - ₹80, Orange - ₹60, Guava - ₹40\n(Prices per
Kg)",
    "fulfillmentMessages": [
      {
        "text": {
          "text": [
            "Fruits: Apple - ₹80, Orange - ₹60, Guava - ₹40\n(Prices per Kg)"
          ]
        }
      }
    ],
    "intent": {
      "name": "projects/fruit-seller-byaaga/agent/intents/bc81652d-a56c-4d62-83b8-
a1307603b774",
      "displayName": "enquiry"
    },
    "intentDetectionConfidence": 0.48040628,
    "diagnosticInfo": {
      "end_conversation": true
    },
    "languageCode": "en"
  }
}
```

CLOSE COPY RAW RESPONSE


IBM Watson Assistant

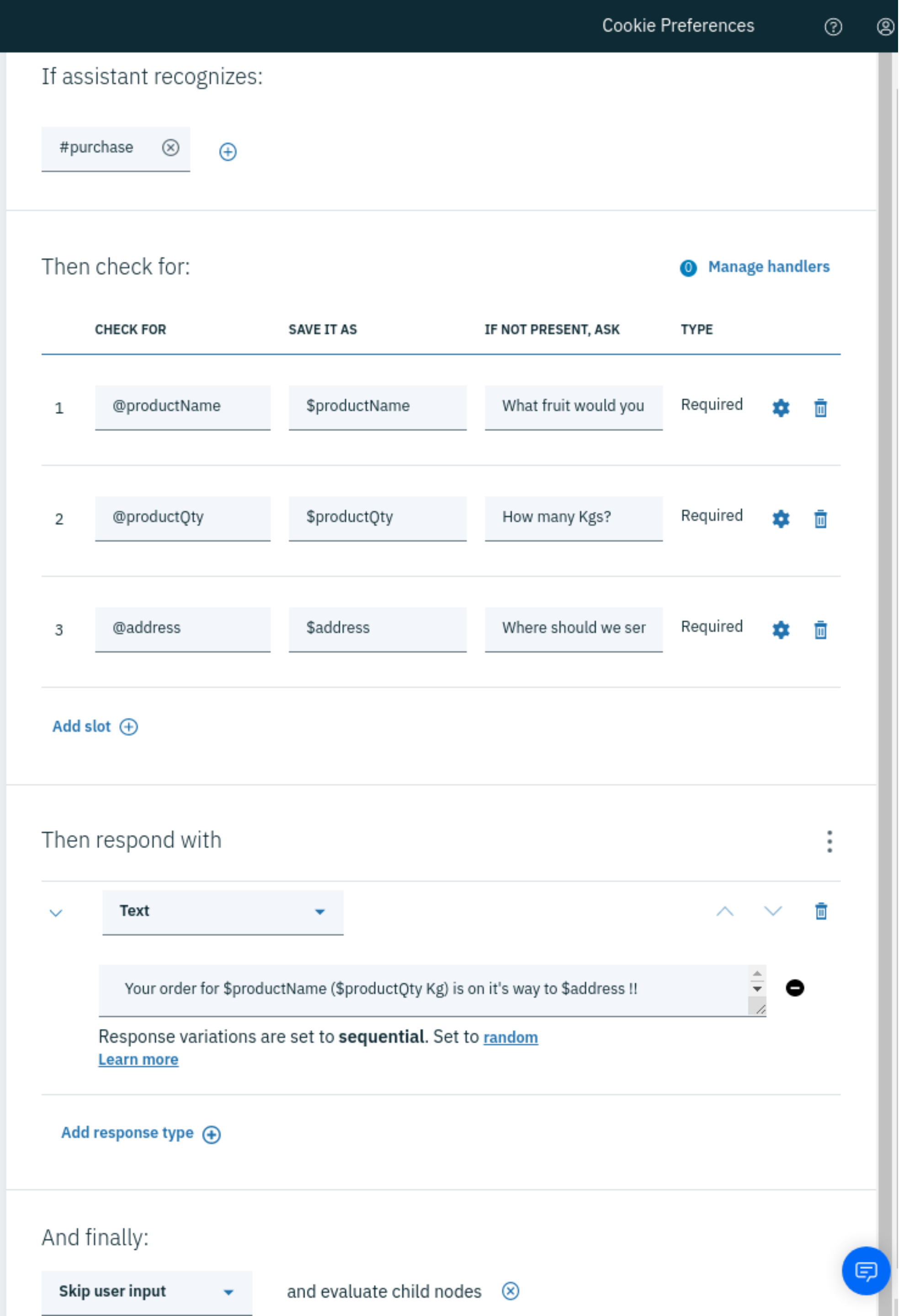
Add node | Add child node | Add folder

Fruit Selling Skill

- Welcome
  welcome
  1 Responses / 0 Context Set / Does not return
- enquiry-node
  #enquiry
  1 Responses / 0 Context Set / Does not return
- purchase-node
  #purchase
  1 Responses / 3 Context Set / 3 Slots / Skip user input / Does not r...
  - **Skip user input** and evaluate child nodes
  - clear-contexts
    true
    0 Responses / 3 Context Set / Return allowed
- Anything else
  anything_else
  1 Responses / 0 Context Set / Does not return

Example of Flow Management –
IBM Watson Assistant's *Dialog Tree*

Definition of a "node"
In the Dialog Tree


Cookie Preferences

If assistant recognizes:

#purchase

Then check for:

Manage handlers

	CHECK FOR	SAVE IT AS	IF NOT PRESENT, ASK	TYPE
1	@productName	$productName	What fruit would you	Required
2	@productQty	$productQty	How many Kgs?	Required
3	@address	$address	Where should we ser	Required

Add slot

Then respond with

Text

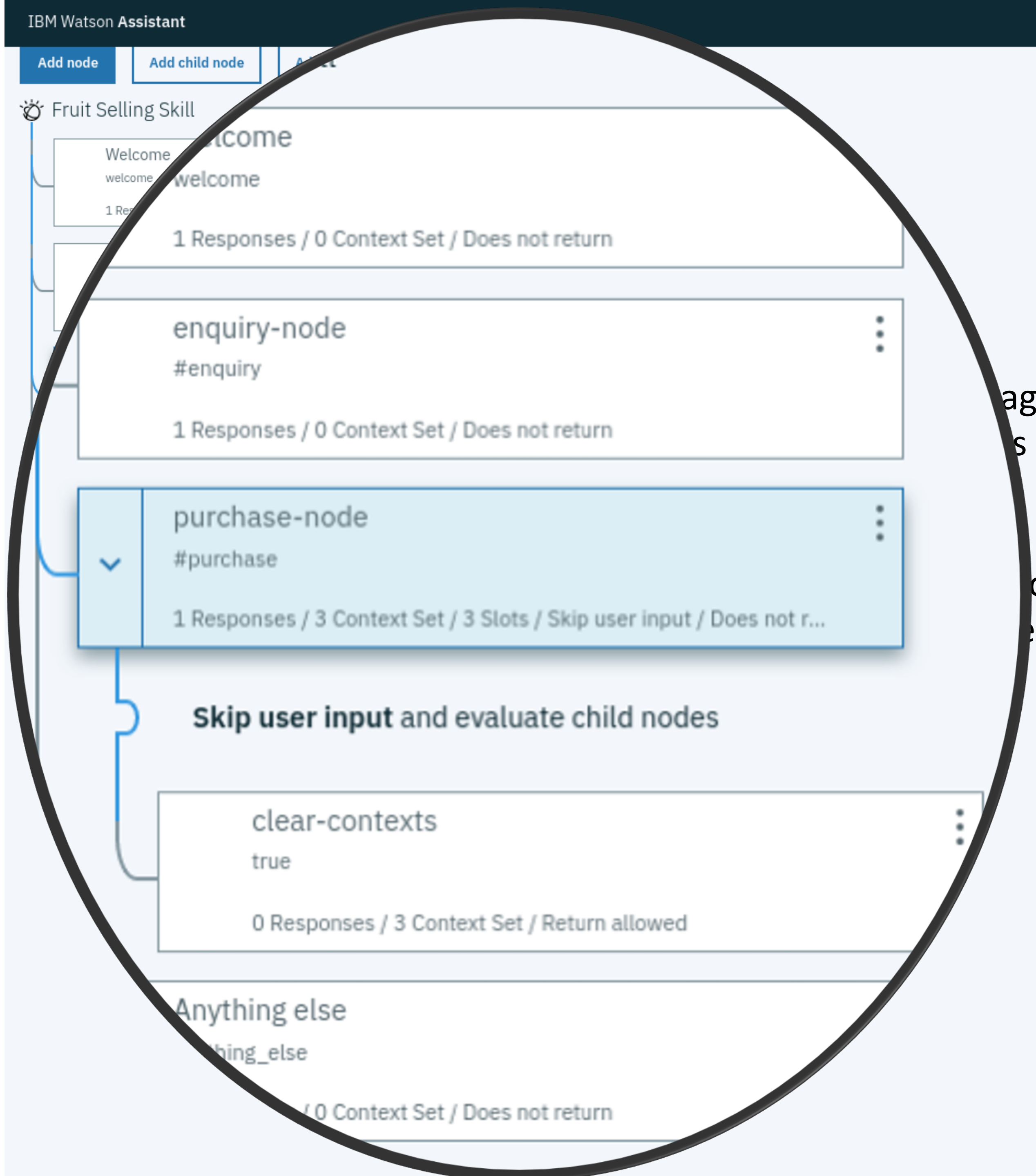
Your order for $productName ($productQty Kg) is on it's way to $address !!

Response variations are set to **sequential**. Set to random
Learn more

Add response type

And finally:

Skip user input and evaluate child nodes


IBM Watson Assistant

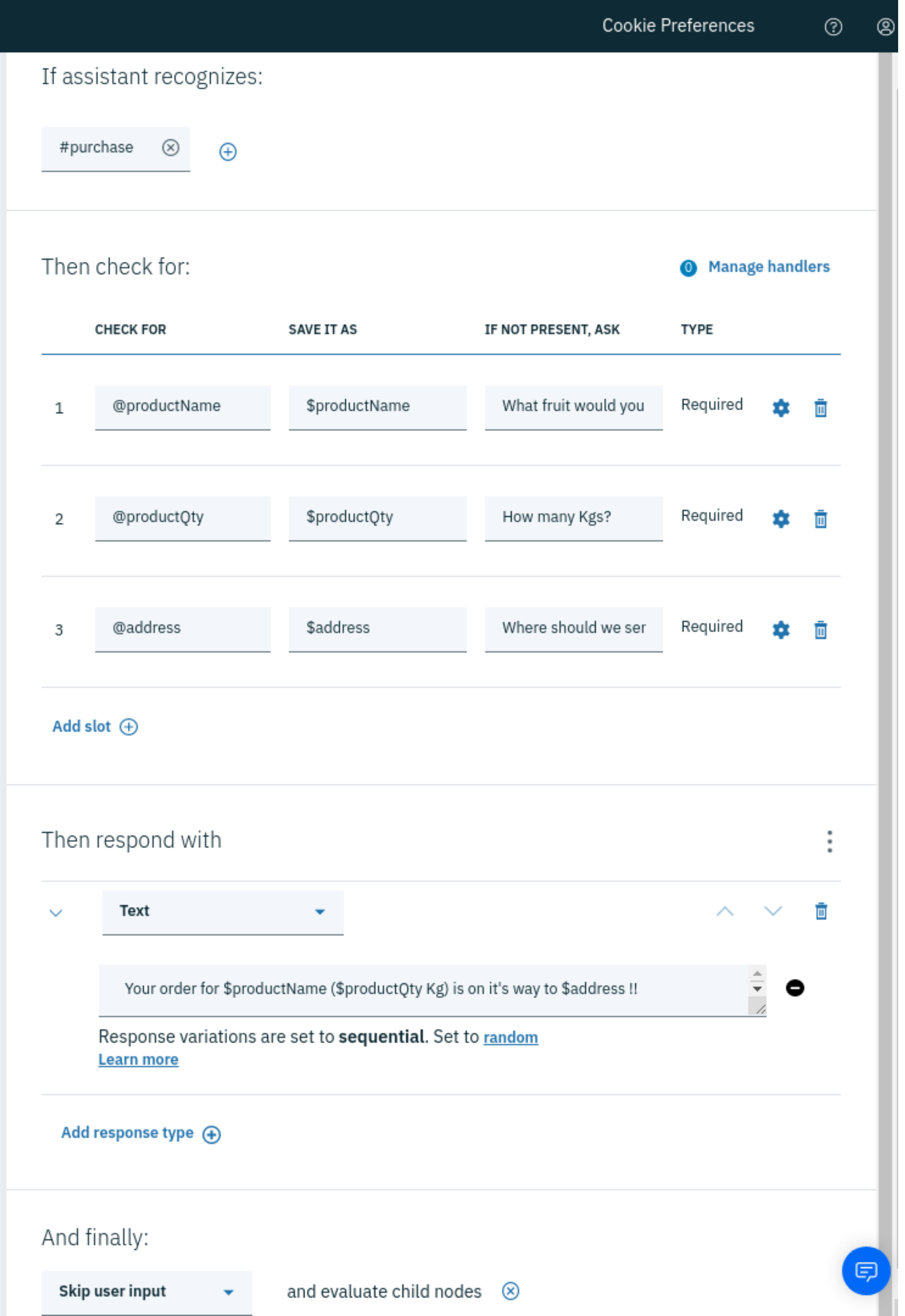
Add node | Add child node

Fruit Selling Skill

- Welcome
  - welcome
  - 1 Responses / 0 Context Set / Does not return
- enquiry-node
  - #enquiry
  - 1 Responses / 0 Context Set / Does not return
- purchase-node
  - #purchase
  - 1 Responses / 3 Context Set / 3 Slots / Skip user input / Does not r...
  - **Skip user input** and evaluate child nodes
  - clear-contexts
    - true
    - 0 Responses / 3 Context Set / Return allowed
- Anything else
  - hing_else
  - / 0 Context Set / Does not return

agement –
s *Dialog Tree*

ode"
e


Cookie Preferences

If assistant recognizes:

#purchase

Then check for: Manage handlers

	CHECK FOR	SAVE IT AS	IF NOT PRESENT, ASK	TYPE
1	@productName	$productName	What fruit would you	Required
2	@productQty	$productQty	How many Kgs?	Required
3	@address	$address	Where should we ser	Required

Add slot

Then respond with

Text

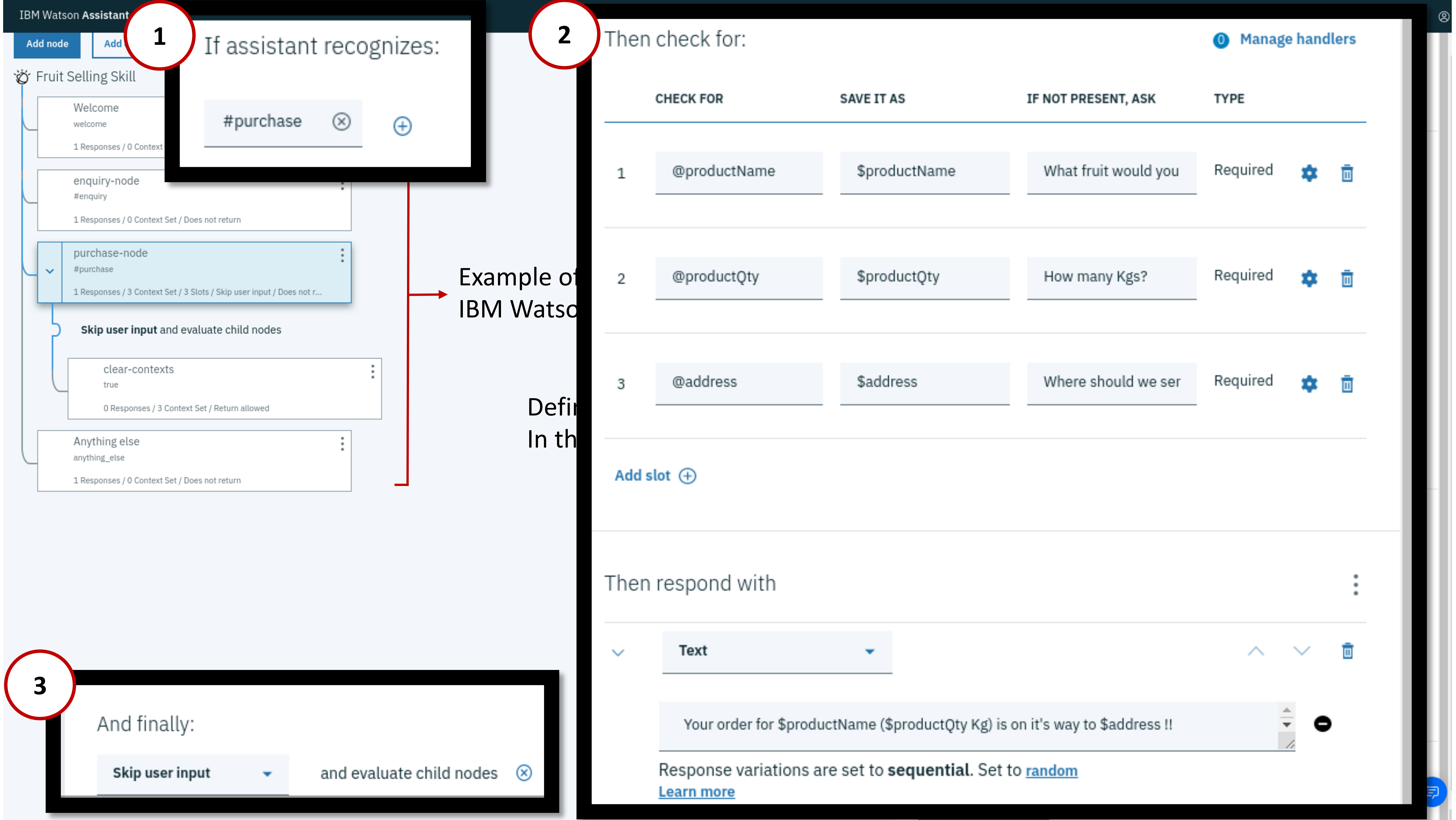
Your order for $productName ($productQty Kg) is on it's way to $address !!

Response variations are set to **sequential**. Set to random
Learn more

Add response type

And finally:

Skip user input and evaluate child nodes



IBM Watson Assistant

Add node | Add

Fruit Selling Skill

Welcome
welcome
1 Responses / 0 Context

enquiry-node
#enquiry
1 Responses / 0 Context Set / Does not return

purchase-node
#purchase
1 Responses / 3 Context Set / 3 Slots / Skip user input / Does not r...

Skip user input and evaluate child nodes

clear-contexts
true
0 Responses / 3 Context Set / Return allowed

Anything else
anything_else
1 Responses / 0 Context Set / Does not return

Example of
IBM Watso

Defi
In th

1

If assistant recognizes:

#purchase

2

Then check for:

0 Manage handlers

	CHECK FOR	SAVE IT AS	IF NOT PRESENT, ASK	TYPE
1	@productName	$productName	What fruit would you	Required
2	@productQty	$productQty	How many Kgs?	Required
3	@address	$address	Where should we ser	Required

Add slot

Then respond with

Text

Your order for $productName ($productQty Kg) is on it's way to $address !!

Response variations are set to **sequential**. Set to random
Learn more

3

And finally:

Skip user input and evaluate child nodes

Quality Attributes – Definition

- **Len Bass**, **Paul Clements** and **Rick Kazman**, in their book titled Software Architecture in Practice, 3rd Edition, Chapter 4, define
  - *A quality attribute (QA) is a measurable or testable property of a system that is used to indicate how well the system satisfies the needs of its stakeholders.*
- They go on to say
  - *You can think of a quality attribute as measuring the "goodness" of a product along some dimension of interest to a stakeholder*.
- Some examples of QAs are ***Reliabiity***, ***Performance***, ***Interoperability***, ***Availability***, ***Modifiability***, ***Usability*** etc.

Architectural Tactics – Definition

- In the <u>same chapter</u>, the authors later explain
    - *We now turn to the techniques an architect can use to achieve the required quality attributes. We call these techniques architectural tactics.*
- They continue as
    - *A tactic is a design decision that influences the achievement of a quality attribute response – tactics directly affect the system's response to some stimulus.*
- Tactics are usually associated with the QA(s) which are dominantly affected by their application
- Example: *Reduce Coupling* and *Increase Cohesion* are common architectural tactics for ***Modifiability***

The Hospitality Framework

Part 2

Achieving quality in (chatbot) applications

- For any software system, achieving quality involves meeting SLAs, keeping up with competition, providing a great user experience etc.
- An independent chatbot, or an application with a chatbot component also have similar concerns
- The architect of the application needs to pen down the non-functional requirements of the application, and relate them to QAs
- However, realising these QAs in an application is not a trivial task, because any efforts to incorporate one, may hamper some other QAs
- There are therefore, critical design trade-offs, that must be evaluated

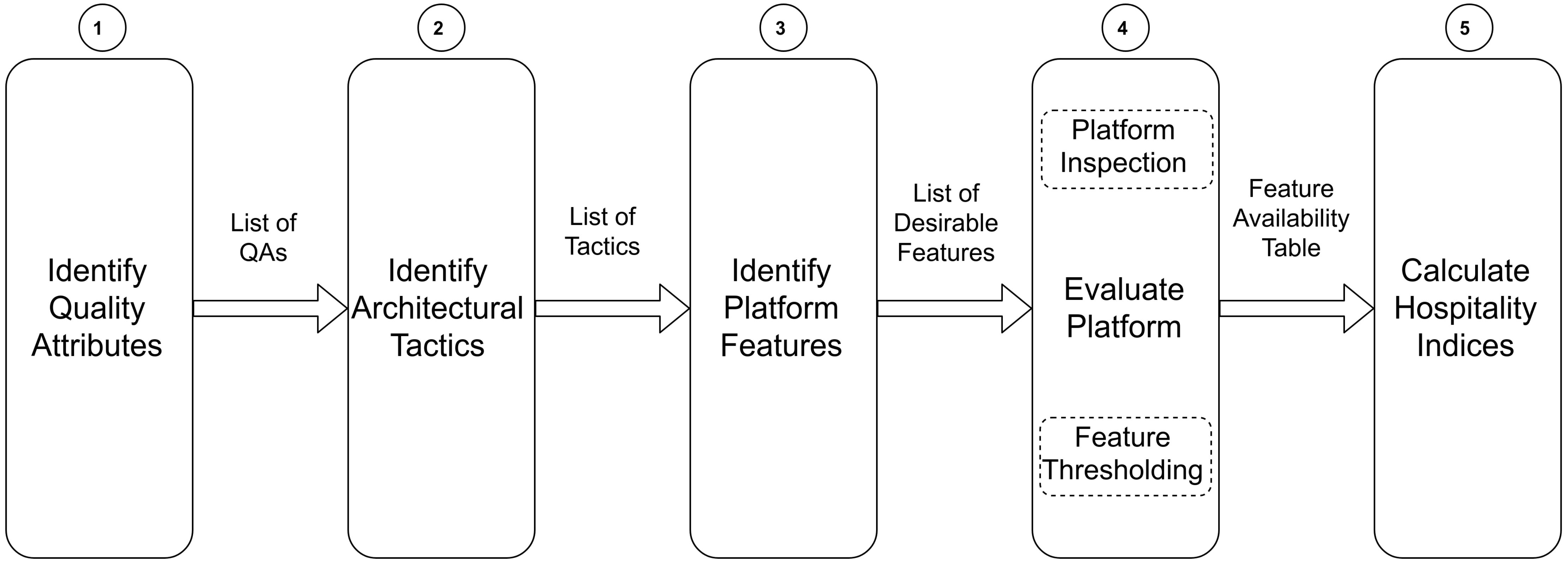
Trade-offs between QAs

- The process of achieving quality in applications may involve understanding the impact of various architectural tactics, for example:
  1. Application of the *Encrypt Data* tactic for ***Security***, would result in additional computation, affecting ***Performance*** negatively
  2. Application of the *Maintain Multiple Copies of Data* tactic for ***Performance***, would result in a partitioned database, making ***Availability*** hard to achieve
- One or more attributes may have to be “prioritised” over others, in case achieving “all” of them is not feasible (usually the case)
- Exactly which attributes are to be priortised, depends on the given use case and the user requirements

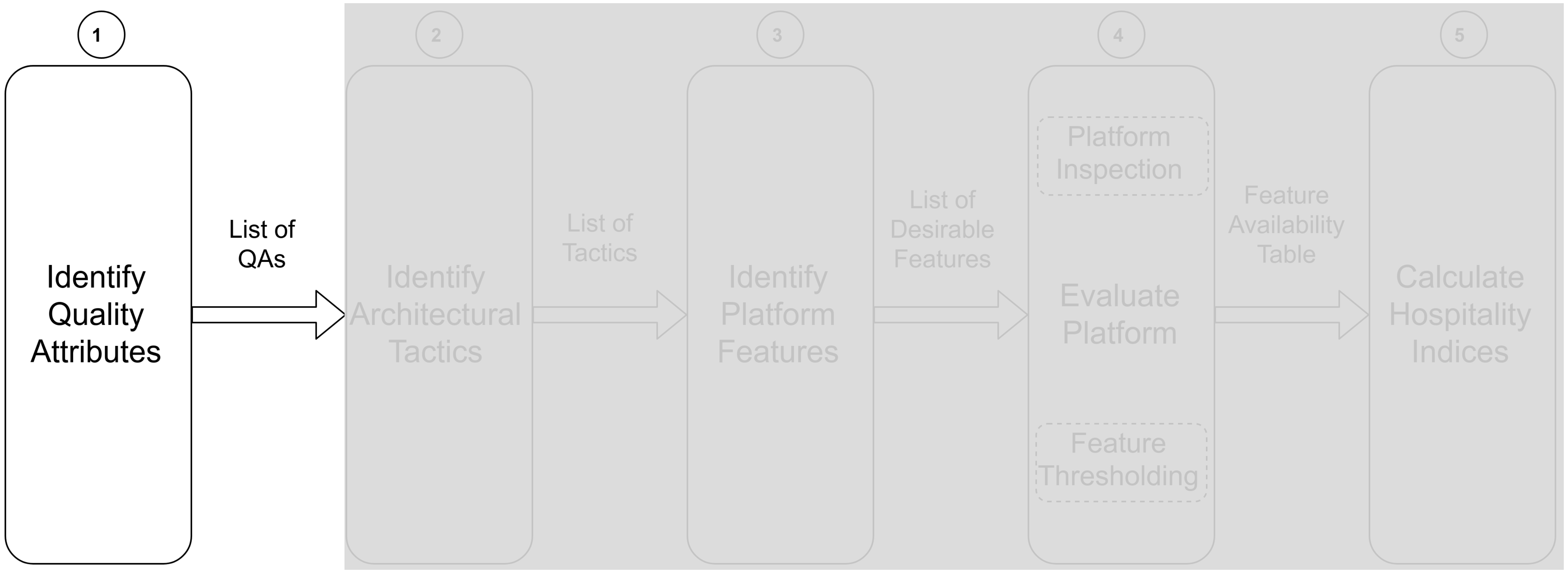
Hospitality Framework

- The *Hospitality Framework* attempts to tackle one part of the problem of achieving quality trade-offs – the role of a *Platform*
- From analysis to development to deployment, software practitioners use a number of commercially available platforms
- The Hospitality Framework attempts to evaluate the usefulness of a platform towards realisation of some “quality goal”
- The goals here could be achieving a quality attribute in general, or encorporation of a particular architectural tactic
- Hospitality framework evaluates a platform's support for achieving them

Phases in application of Hospitality Framework


1
2
3
4
5
Identify Quality Attributes
List of QAs
Identify Architectural Tactics
List of Tactics
Identify Platform Features
List of Desirable Features
Platform Inspection
Evaluate Platform
Feature Thresholding
Feature Availability Table
Calculate Hospitality Indices

1. Identify Quality Attributes


1
Identify Quality Attributes
List of QAs
2
Identify Architectural Tactics
List of Tactics
3
Identify Platform Features
List of Desirable Features
4
Platform Inspection
Evaluate Platform
Feature Thresholding
Feature Availability Table
5
Calculate Hospitality Indices

The Fruit-selling bot – a simple use case

- In order to show the application of the framework, we'll pick the simple use case of an application to be built for a fruit selling shop
- The shop wants a chatbot to be deployed on their website, as well as their app, which can interact with (potential) customers
- It should be able to answer common user queries – like available fruits, their prices, directions to the physical store etc.
- It should also be able to contact the shop's *backend* servers to place orders, generate shipping labels, assign delivery boys etc.
- While the website could be "*text*-only", the app should also have a "*voice*" interface to receive audio inputs and provide audio responses

Requirements – Functional vs Non-functional

Functional Requirements	Non-functional Requirements
• Need an app as well as a website	• Keep chat transcripts onsite (privacy concerns)
• User could browse through fruits available in the inventory	• Chatbot component needs access from multiple locations (app/website)
• Answer user queries about fruit prices, availability, etc.	• Will have access to inventory, require some authentication
• If the user needs directions to the store, provide guidance	• Need voice-to-text/text-to-voice capability, if the chatbot cannot handle it implicitly
• Allow typed text/spoken queries on the app	• Keep the bot simple; don't attempt to answer queries with low confidence
	• Add counter-examples to avoid responding to queries like "How's the weather"?
	• Validate the bot before deployment, check behaviour for common user utterances
	• Response strings and prompts might change/customised

Requirements – Functional vs Non-functional

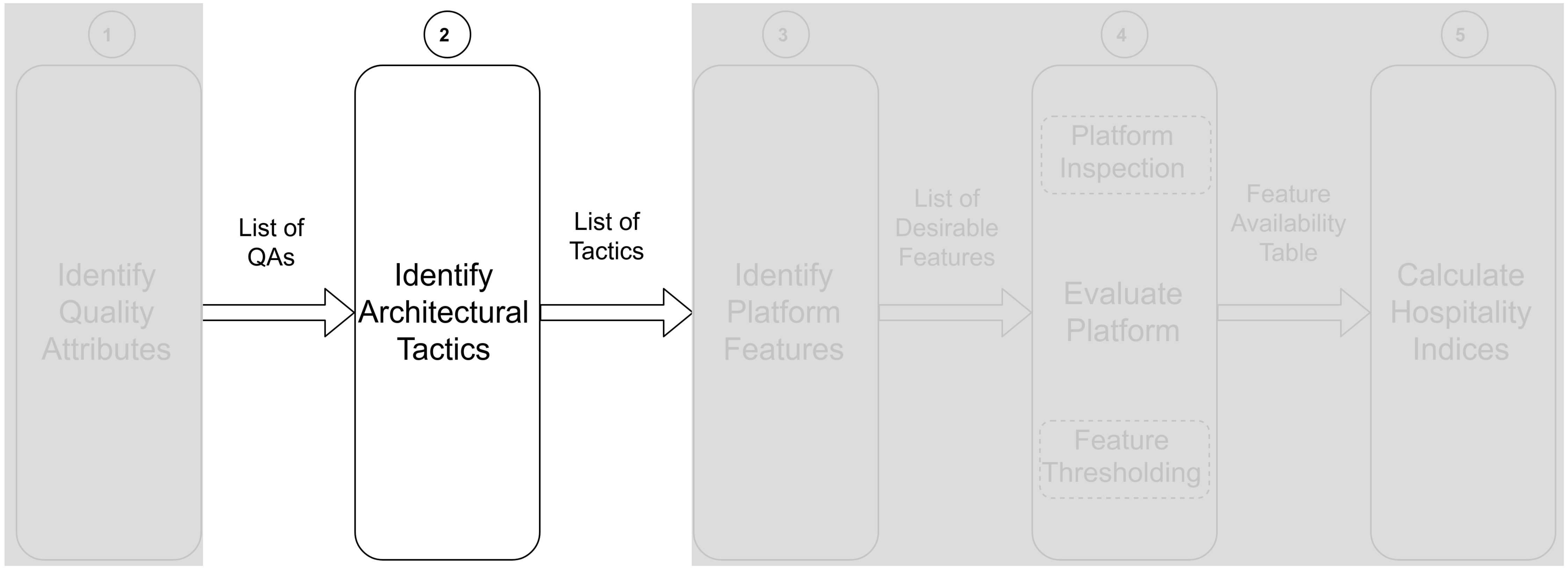
Functional Requirements	Non-functional Requirements
• Need an app as well as a website	• Keep chat transcripts onsite (privacy concerns)
• User could browse through fruits available in the inventory	• Chatbot component needs access from multiple locations (app/website)
• Answer user queries about fruit prices, availability, etc.	• Will have access to inventory, require some authentication
• If the user needs directions to the store, provide guidance	• Need voice-to-text/text-to-voice capability, if the chatbot cannot handle it implicitly
• Allow typed text/spoken queries on the app	• Keep the bot simple; don't attempt to answer queries with low confidence
	• Add counter-examples to avoid responding to queries like "How's the weather"?
	• Validate the bot before deployment, check behaviour for common user utterances
	• Response strings and prompts might change/customised

We will be focusing here for the rest of the phases

Quality Attributes for the sample use case

- ***Modifiability***
  - The shop may wish to customize or update the responses that they show to the user, e.g. "Your fruits are shipped" ⇒ "You'll have your apples soon !!"
- ***Security & Privacy***
  - Usually the tactics for these two attributes are often intertwined
  - Since the bot will interact with user data, keeping it secure as well as away from prying eyes should be a concern
- ***Interoperability***
  - We have two different platforms for deployment, with different I/O needs
- ***Reliability***
  - It is better to cater to less functionality well, than more functionality poorly

2. Identify Architectural Tactics


1
Identify Quality Attributes
List of QAs
2
Identify Architectural Tactics
List of Tactics
3
Identify Platform Features
List of Desirable Features
4
Platform Inspection
Evaluate Platform
Feature Thresholding
Feature Availability Table
5
Calculate Hospitality Indices

Tactics for Modifiability

Tactic	Reason behind choosing this tactic
Abstract Common Services	Keeping intents, parameters and flow logic separate allows adding or modifying them independently.
Defer Binding	Allows tailored responses based on user inputs.
Split Module	Separates the intent matching from business logic.

Tactics for Security & Privacy

Tactic	Reason behind choosing this tactic
Authenticate Communication	Prevents the chatbot from unauthorized access (superfluous calls to platform may incur additional cost).
Protect Data at Rest	Keeps the conversations between users and the store private.
Protect Data in Motion	Prevents breaches due to eavesdropping (e.g. Man-in-the-middle attacks).

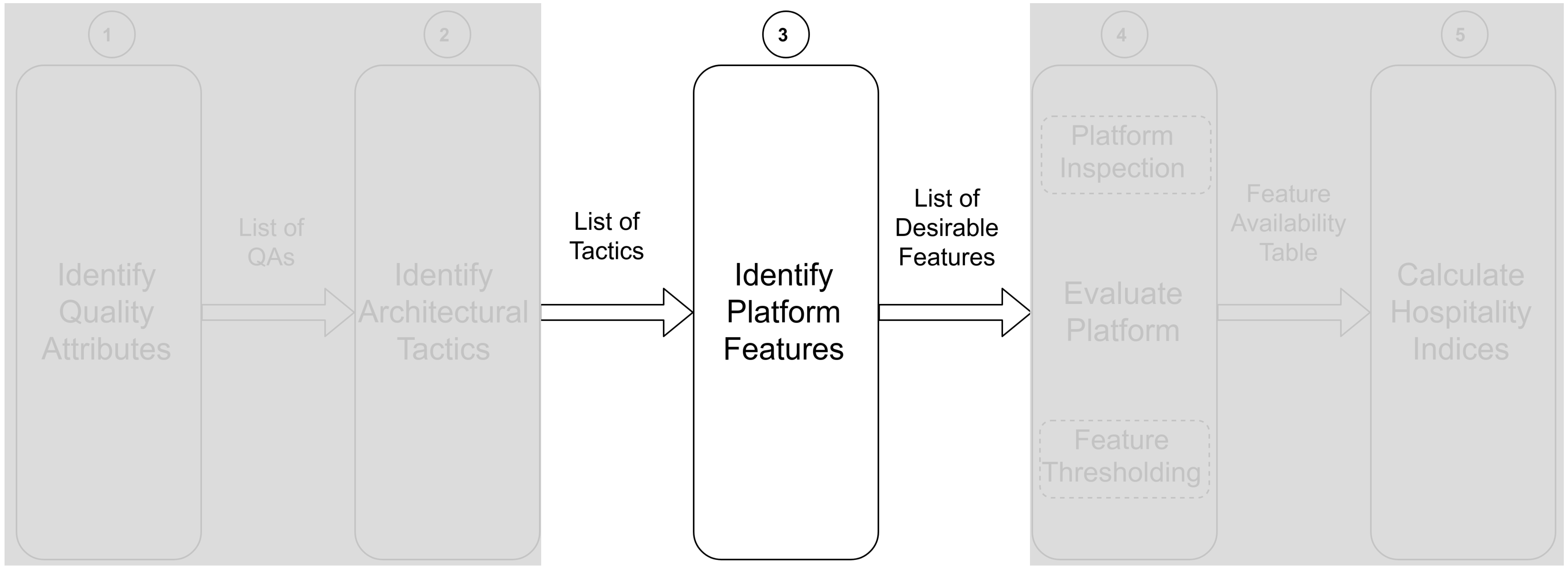
Tactics for Interoperability

Tactic	Reason behind choosing this tactic
Manage Interfaces	Require both ingress and egress capabilities, to and from the chatbot (e.g. API access).
Support multiple Data Formats	Chatbot needs to take queries (and send responses) in both text and audio formats.

Tactics for Reliability

Tactic	Reason behind choosing this tactic
Validate common use cases	Verifies that expected user utterances are properly processed by the chatbot.
Prevent Failures	Restricts the chatbot from responding with low confidence.
Recover from Failures	Handles known nuances of common conversation (e.g. assuming defaults for missing information).

3. Identify Platform Features


1
2
3
4
5
Identify Quality Attributes
List of QAs
Identify Architectural Tactics
List of Tactics
Identify Platform Features
List of Desirable Features
Platform Inspection
Evaluate Platform
Feature Thresholding
Feature Availability Table
Calculate Hospitality Indices

Finding features of a given platform

- This phase involves some "reading"
  - Not in-depth, but good enough to get an idea
- For Chatbot platforms, the best place to read are the numerous blogging websites, which put up articles about latest news
- The Chatbot platforms are still evolving (and documenting changes is usually not a priority), making these articles an even better source
- Reading recent articles that compare two or more platforms may provide a good idea about their offerings
- However, most of the articles do have **biases** (they tend to favour one platform more than the others)

10 Best Chatbot Builders in 2019

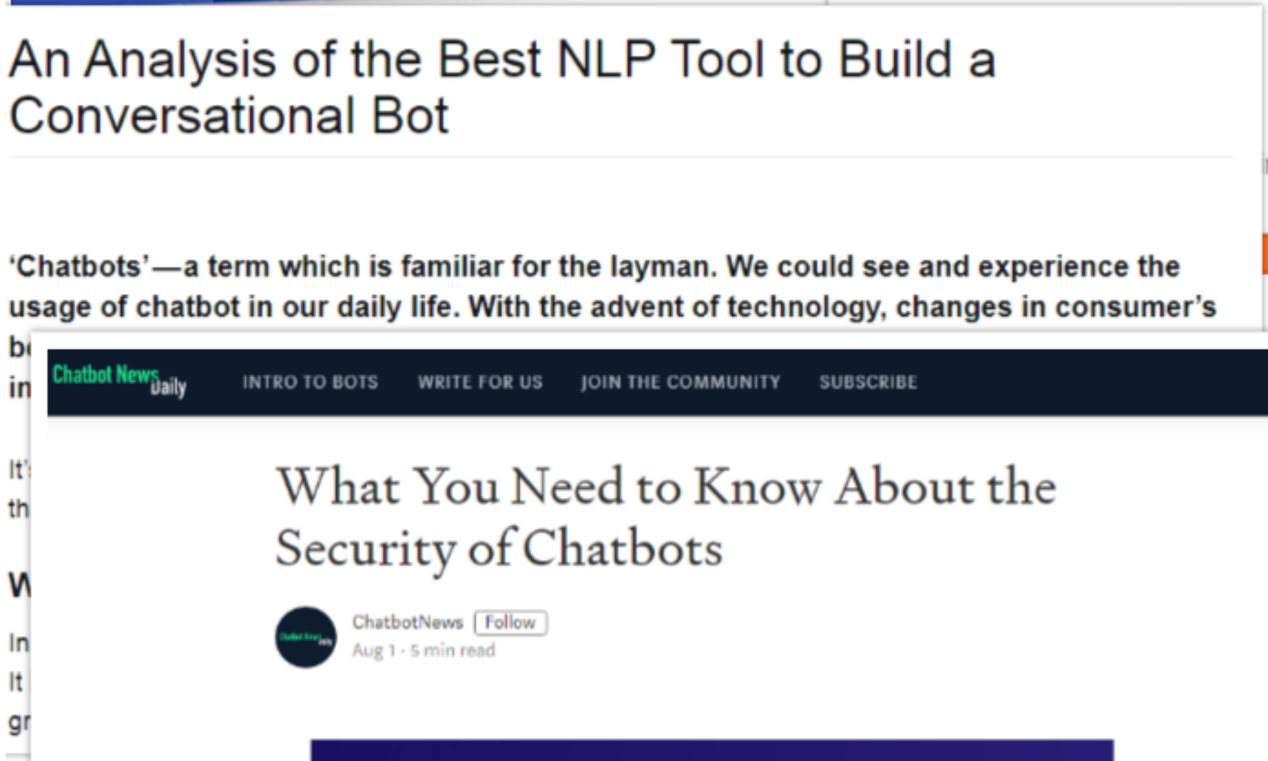
Octo

Dialogflow vs Lex vs LUIS vs Watson vs Chatfuel

I get this question a lot.

"How does Watson compare to Dialogflow?"

nions in this article.



An Analysis of the Best NLP Tool to Build a Conversational Bot

'Chatbots'—a term which is familiar for the layman. We could see and experience the usage of chatbot in our daily life. With the advent of technology, changes in consumer's b
in

It'
th

W

In
It
gr

What You Need to Know About the Security of Chatbots

ChatbotNews Follow
Aug 1 · 5 min read

ubisend Blog

About Chatbots • August 2, 2019

[Podcast] A Chatbot Is for Life Not Just for Christmas with Sean Clark

Chatbot RO

8 Ways C
(and Roa

Platform Features for Modifiability

Tactic	Desired Platform Features
Abstract Common Services	Ability to create intents independently
	Ability to create parameters independently
	Ability to manage conversation flow independently
Defer Binding	Ability to externalise response generation
	Allow placeholders in response to fill parameter values
	Allow conditional responses
Split Module	Ability to externalise parameter validation
	Ability to externalise response generation

Platform Features for Security & Privacy

Tactic	Desired Platform Features
Authenticate Communication	Ability to create and verify credentials for accessing the chatbot
	Ability to supply credentials to an external source
Protect Data at Rest	Ability to create and verify credentials for accessing chat data
	Ability to keep chat transcripts onsite
Protect Data in Motion	Use secured channels only for communication (e.g. allow `https` and block `http`)

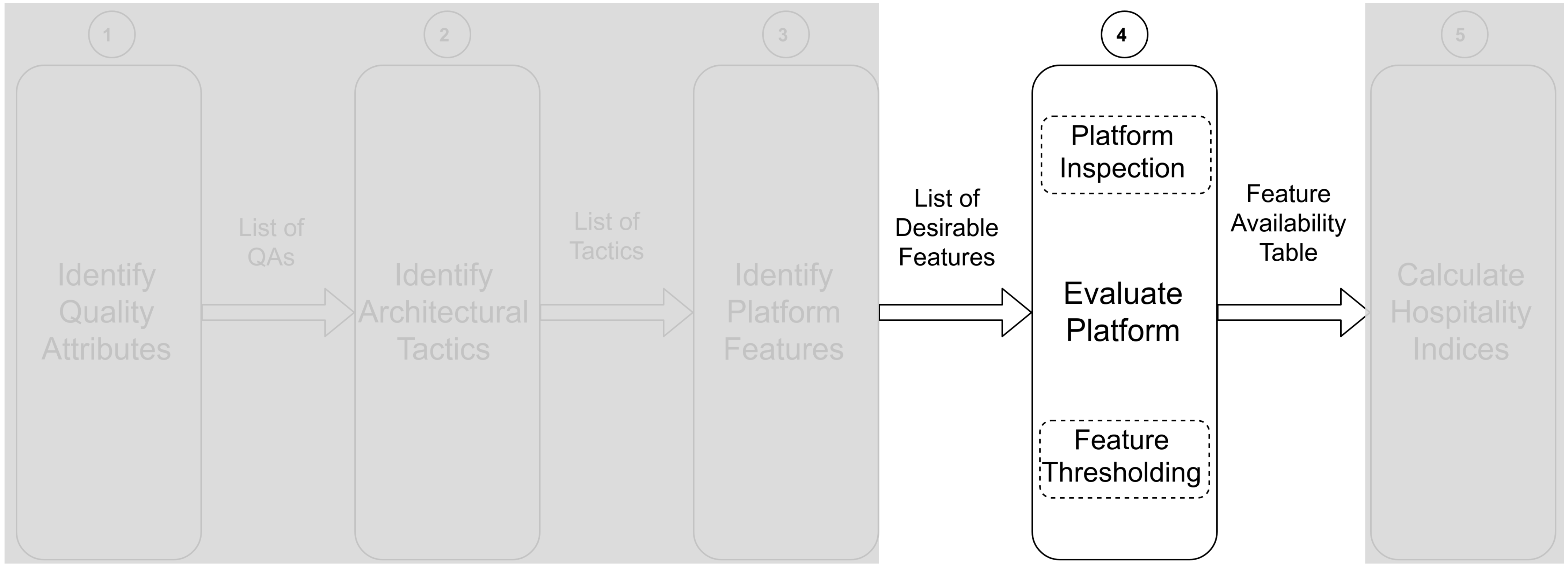
Platform Features for Interoperability

Tactic	Desired Platform Features
Manage Interfaces	Allow API access for intent classification
	Allow API access for slot filling
	Ability to trigger external events
Support multiple Data Formats	Ability to receive voice input
	Provide transcribed text from speech
	Ability to send voice output

Platform Features for Reliability

Tactic	Desired Platform Features
Validate common use-cases	Provide Test Console to observe chatbot response for specific inputs
	Provide Test Console to observe the debug information for specific inputs
Prevent Failures	Ability to set confidence threshold for intent classification
	Ability to provide counter-examples
	Ability to digress and return
Recover from Failures	Ability to provide default conversation flow
	Ability to provide default values for slots

4. Evaluate Platform


1
2
3
4
5
Identify Quality Attributes
List of QAs
Identify Architectural Tactics
List of Tactics
Identify Platform Features
List of Desirable Features
Platform Inspection
Evaluate Platform
Feature Thresholding
Feature Availability Table
Calculate Hospitality Indices

Platform Inspection

- At this stage, the task of inspecting a platform becomes crucial
- This involves searching for particular keywords on the web
  - For example: "dialogflow set confidence" or "lex provide speech input"
  - The results for these searches will usually provide a quick answer to questions like "Can I set a minimum confidence threshold for intent firing in dialogflow?"
- It also involves looking at specific pages in the documentation archives of the particular platform
  - For example: **Test the Bot Using Speech Input (AWS CLI)**
- While in some cases, you may get a straight Yes/No answer to the question, sometimes, the decision may be more complex

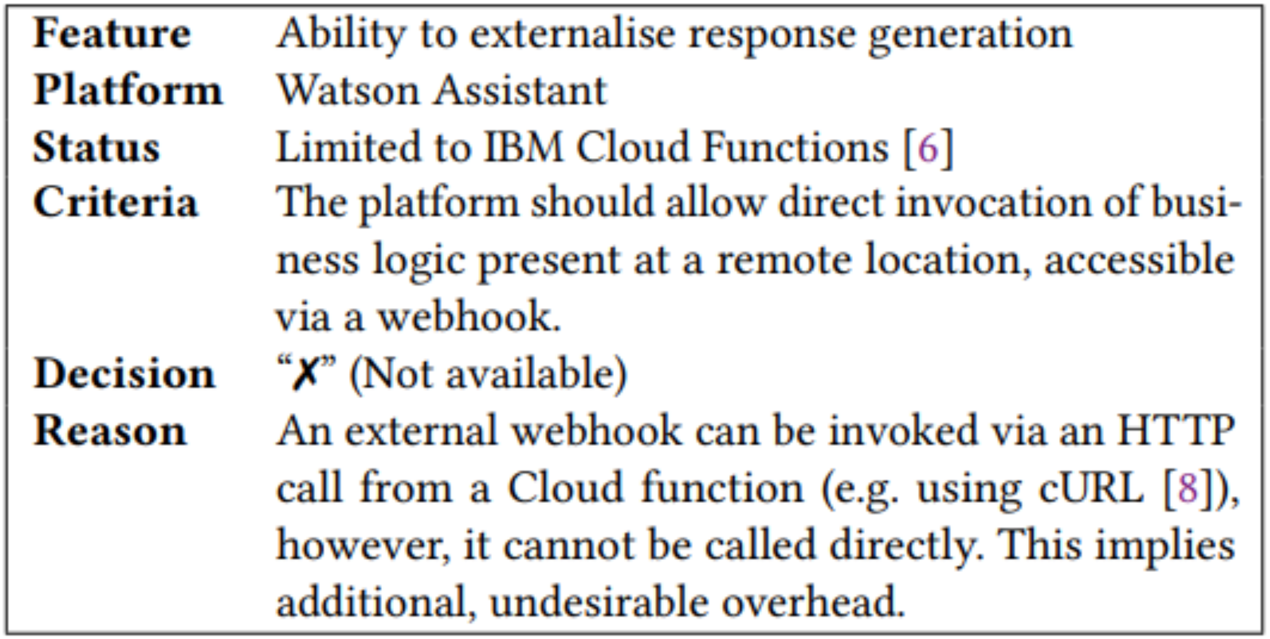
Feature Thresholding

- There can be cases where a feature may only be "partially" supported by a platform
- In such cases, we need to perform what we have termed here as *Feature Thresholding*
- The idea of feature thresholding is that in case a feature is only partially supported, some additional effort will be required
  - The question is – how much work the developer has to do here? If the work is substantial, I count it as a **Nay**, otherwise I term it as **Yay**
- To do so, the architect can create *Feature Cards*, and distribute them among the stakeholders (developers, testers, integrators etc.)

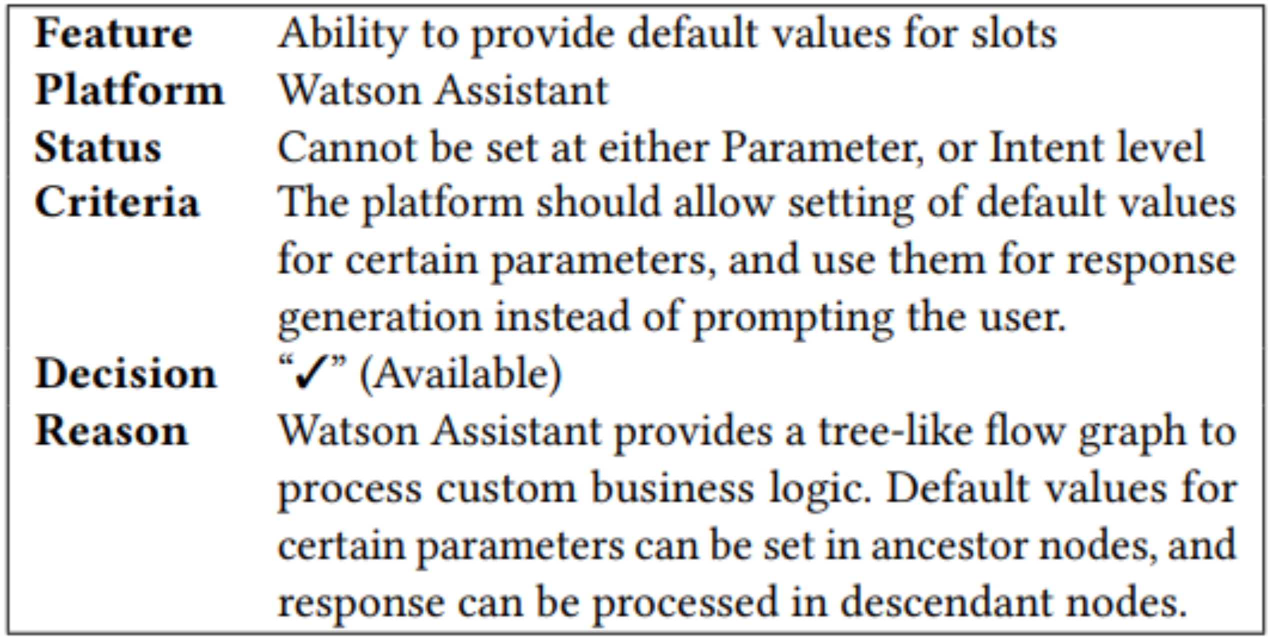
Feature Cards

- Feature Cards can be made on a "per feature, per platform" basis, where the architect is in two minds
- The Feature Cards should mention the platform name, required feature description, and the related offering by the platform
- The stakeholders can opine whether they consider this feature "good enough" (meaning that the custom efforts will be minimal) or not
- Each stakeholder provides a decision – Yes or No – as well as reasons for the decision
- The architect can use these cards before taking a final call

Examples of Feature Cards



Feature	Ability to externalise response generation
Platform	Watson Assistant
Status	Limited to IBM Cloud Functions [6]
Criteria	The platform should allow direct invocation of business logic present at a remote location, accessible via a webhook.
Decision	"✗" (Not available)
Reason	An external webhook can be invoked via an HTTP call from a Cloud function (e.g. using cURL [8]), however, it cannot be called directly. This implies additional, undesirable overhead.



Feature	Ability to provide default values for slots
Platform	Watson Assistant
Status	Cannot be set at either Parameter, or Intent level
Criteria	The platform should allow setting of default values for certain parameters, and use them for response generation instead of prompting the user.
Decision	"✓" (Available)
Reason	Watson Assistant provides a tree-like flow graph to process custom business logic. Default values for certain parameters can be set in ancestor nodes, and response can be processed in descendant nodes.

Snapshot from the paper – **Hospitality of Chatbot building Platforms**, Saurabh Srivastava and T.V. Prabhakar, SQUADE, Tallinn, Aug 26, 2019

Feature Table (1/3)

Desired Platform Feature	Watson Assistant	Dialogflow	Lex
Ability to create intents independently			
Ability to create parameters independently			
Ability to manage conversation flow independently		✗	✗
Ability to externalise response generation	✗		✗
Allow placeholders in response to fill parameter values			
Allow conditional responses		✗	✗
Ability to externalise parameter validation	✗		✗
Ability to externalise response generation	✗		✗

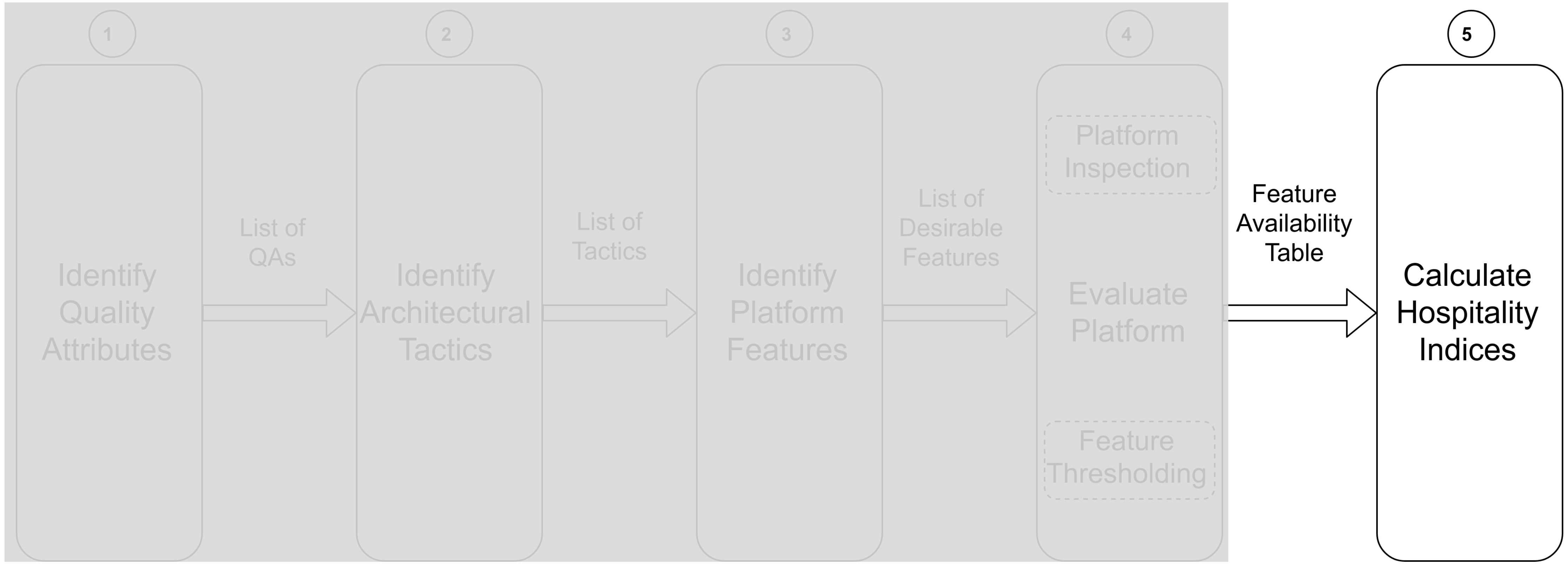
Feature Table (2/3)

Desired Platform Feature	Watson Assistant	Dialogflow	Lex
Ability to create and verify credentials for accessing the chatbot			
Ability to supply credentials to an external source	✗		✗
Ability to create and verify credentials for accessing chat data			
Ability to keep chat transcripts onsite		✗	✗
Use secured channels only for communication (e.g. allow `https` and block `http`)			
Allow API access for intent classification			
Allow API access for slot filling			
Ability to trigger external events	✗		✗
Ability to receive voice input	✗		

Feature Table (3/3)

Desired Platform Feature	Watson Assistant	Dialogflow	Lex
Provide transcribed text from speech			
Ability to send voice output			
Provide Test Console to observe chatbot response for specific inputs			
Provide Test Console to observe the debug information for specific inputs			
Ability to set confidence threshold for intent classification			✗
Ability to provide counter-examples			✗
Ability to digress and return		✗	✗
Ability to provide default conversation flow			
Ability to provide default values for slots			✗

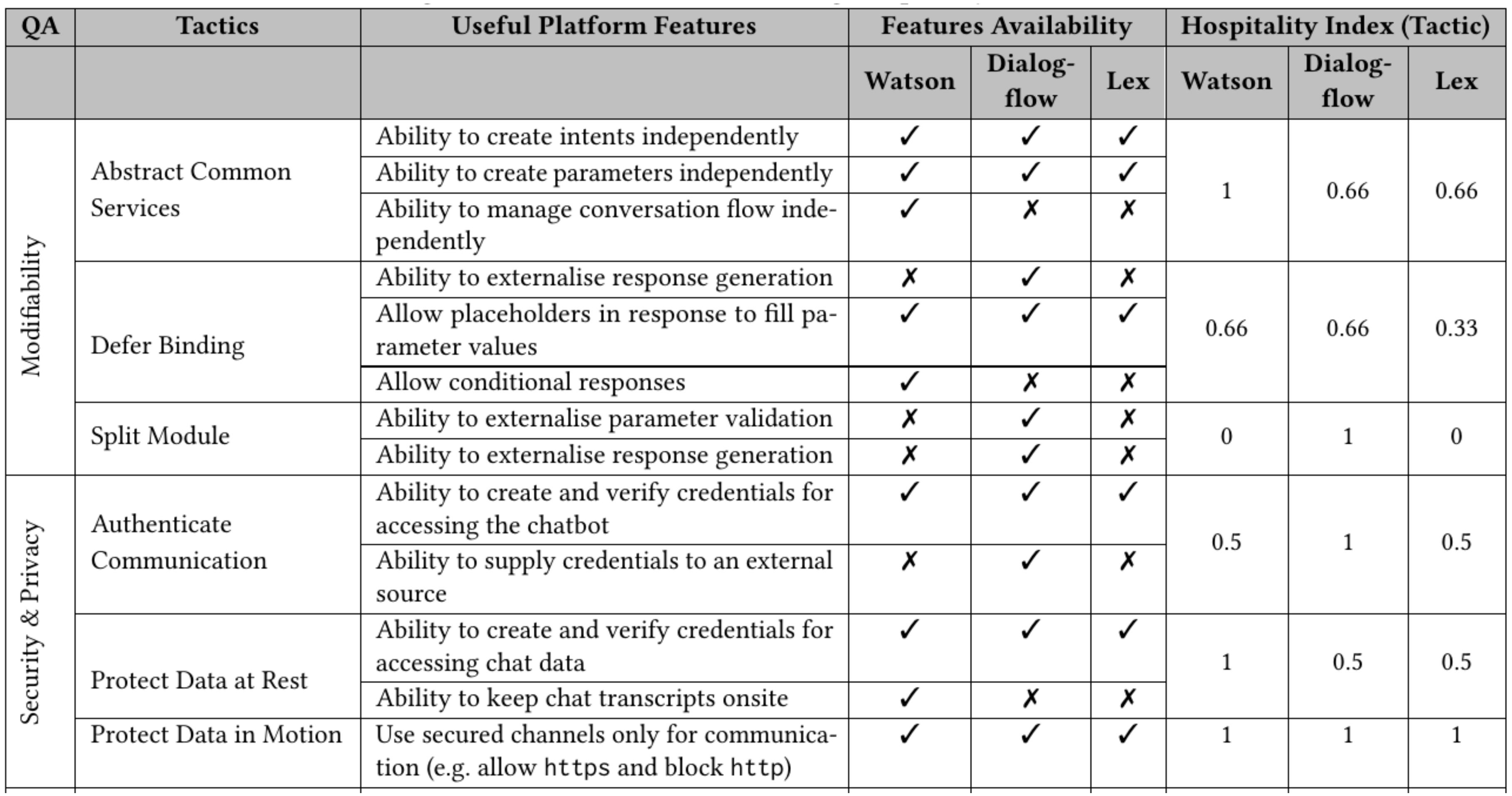
5. Calculate Hospitality Indices


1
Identify Quality Attributes
List of QAs
2
Identify Architectural Tactics
List of Tactics
3
Identify Platform Features
List of Desirable Features
4
Platform Inspection
Evaluate Platform
Feature Thresholding
Feature Availability Table
5
Calculate Hospitality Indices

Hospitality Indices

- The idea behind application of this framework is to be able to "quantify" the "goodness" of a platform for a given use case
- This means, we need a metric to compare the platforms
- Hospitality Index is a measure that provides a number between `0` and `1` (`0` being "bad" and `1` being "good")
- The idea is based on a *weighted-sum analysis* – provide a weight to a given feature or a given tactic, and compute a bottom-up score
- Hospitality Index can be computed at two levels – Tactic or QA
    - Hospitality Index at QA level, uses respective Hospitality Indices at Tactic level

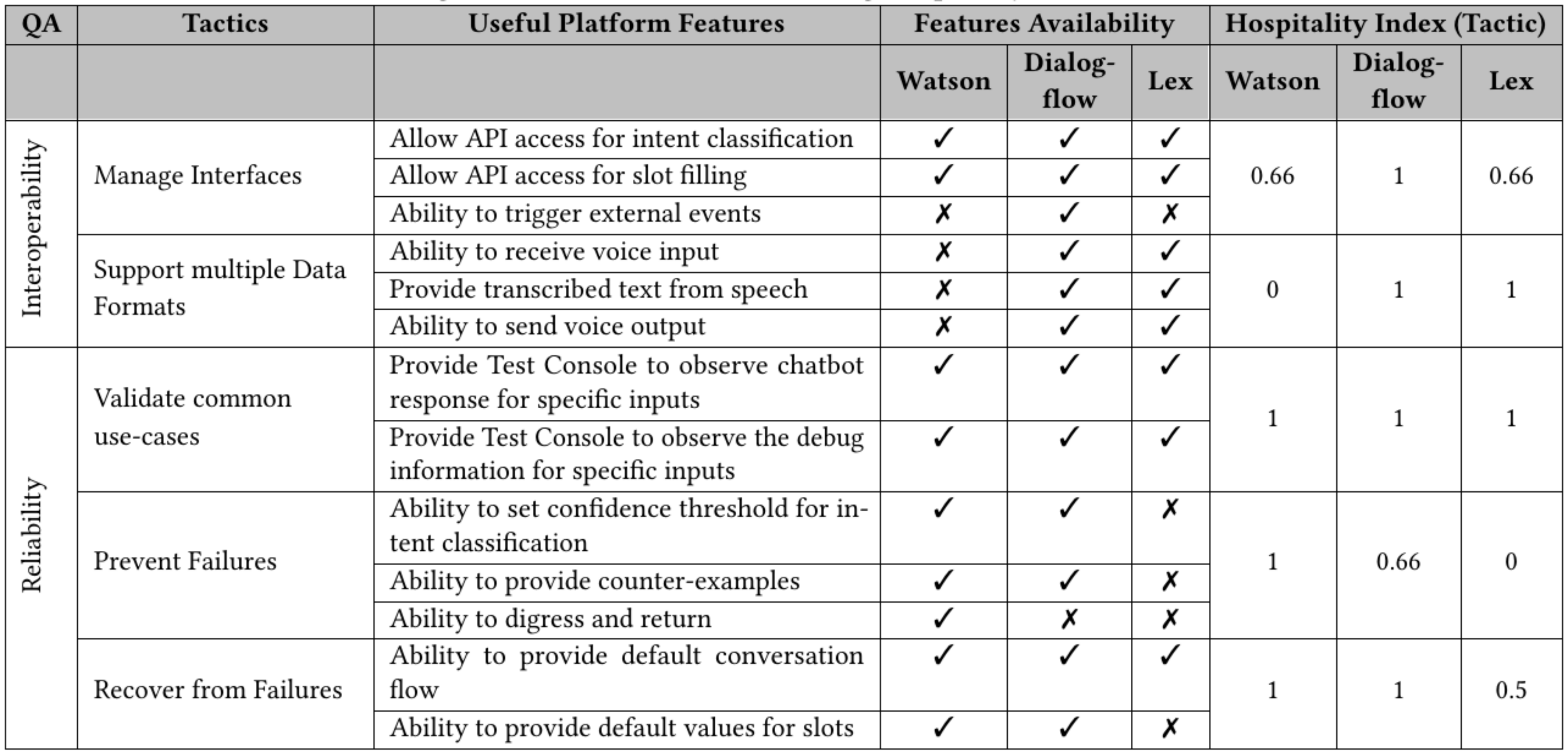
Hospitality Indices at Tactic Level (1/2)



QA	Tactics	Useful Platform Features	Features Availability			Hospitality Index (Tactic)		
			Watson	Dialog-flow	Lex	Watson	Dialog-flow	Lex
Modifiability	Abstract Common Services	Ability to create intents independently	✓	✓	✓	1	0.66	0.66
		Ability to create parameters independently	✓	✓	✓			
		Ability to manage conversation flow independently	✓	✗	✗			
	Defer Binding	Ability to externalise response generation	✗	✓	✗	0.66	0.66	0.33
		Allow placeholders in response to fill parameter values	✓	✓	✓			
		Allow conditional responses	✓	✗	✗			
	Split Module	Ability to externalise parameter validation	✗	✓	✗	0	1	0
		Ability to externalise response generation	✗	✓	✗			
Security & Privacy	Authenticate Communication	Ability to create and verify credentials for accessing the chatbot	✓	✓	✓	0.5	1	0.5
		Ability to supply credentials to an external source	✗	✓	✗			
	Protect Data at Rest	Ability to create and verify credentials for accessing chat data	✓	✓	✓	1	0.5	0.5
		Ability to keep chat transcripts onsite	✓	✗	✗			
	Protect Data in Motion	Use secured channels only for communication (e.g. allow `https` and block `http`)	✓	✓	✓	1	1	1

Snapshot from the paper – **Hospitality of Chatbot building Platforms**, Saurabh Srivastava and T.V. Prabhakar, SQUADE, Tallinn, Aug 26, 2019

Hospitality Indices at Tactic Level (2/2)



QA	Tactics	Useful Platform Features	Features Availability			Hospitality Index (Tactic)		
			Watson	Dialog-flow	Lex	Watson	Dialog-flow	Lex
Interoperability	Manage Interfaces	Allow API access for intent classification	✓	✓	✓	0.66	1	0.66
		Allow API access for slot filling	✓	✓	✓			
		Ability to trigger external events	✗	✓	✗			
	Support multiple Data Formats	Ability to receive voice input	✗	✓	✓	0	1	1
		Provide transcribed text from speech	✗	✓	✓			
		Ability to send voice output	✗	✓	✓			
Reliability	Validate common use-cases	Provide Test Console to observe chatbot response for specific inputs	✓	✓	✓	1	1	1
		Provide Test Console to observe the debug information for specific inputs	✓	✓	✓			
	Prevent Failures	Ability to set confidence threshold for intent classification	✓	✓	✗	1	0.66	0
		Ability to provide counter-examples	✓	✓	✗			
		Ability to digress and return	✓	✗	✗			
	Recover from Failures	Ability to provide default conversation flow	✓	✓	✓	1	1	0.5
		Ability to provide default values for slots	✓	✓	✗			

Snapshot from the paper – **Hospitality of Chatbot building Platforms**, Saurabh Srivastava and T.V. Prabhakar, SQUADE, Tallinn, Aug 26, 2019

Hospitality Indices at Quality Attribute Level

Quality Attribute	Hospitality Index		
	Watson Assistant	Dialogflow	Lex
Modifiability	0.553	0.773	0.330
Security & Privacy	0.833	0.833	0.667
Interoperability	0.330	1.000	0.830
Reliability	1.000	0.887	0.500

Sample Computation of Hospitality Index

- Hospitality Index at the *Defer Binding* tactic
  - Assuming <u>equal</u> weights to all features, we have:
    - Watson Assistant – (`0 + 1 + 1)/3 = 0.66`
    - Dialogflow – (`0 + 1 + 1)/3 = 0.66`
    - Lex - (`0 + 1 + 1)/3 = 0.33`
- Hospitality Index at the ***Modifiability*** QA
  - Assuming equal weights to all tactics, we have:
    - Watson Assistant – `(1 + 0.66 + 0)/3 = 0.553`
    - Dialogflow – `(0.66 + 0.66 + 1) = 0.773`
    - Lex - `(0.66 + 0.33 + 0) = 0.33`

Discussion

Part-3

Uses of the framework

- Selecting a platform
    - We can calculate Hospitality Indices at the QA level for each QA of interest
    - We can then use methods like *Multi-criteria Decision Analysis* to come up with a ranking of the platforms for use
- Selecting other architectural components
    - The analysis provides a great insight into the capabilities and features exposed by the platform
    - This can provide architectural hints for architecting other parts of the system
    - For example, **Watson Assistant** doesn't provide an audio interface, however, by composing solutions using **Watson Speech-to-Text** and **Watson Text-to-Speech,** an application can still provide the "speech" interface

Thank You

That'll be all from my side. Over to you !!