

Which Conversational AI Platform Should I Choose?

A Broad Review + Comparison

FOR

CONTENTS

- 03** Overview
- 04** Common Features
- 05** Quick Comparison Table
- 07** Amazon Lex
- 09** Google Dialogflow
- 14** IBM Watson Assistant
- 17** Microsoft Bot Framework
- 21** Key Takeaways

In our blog, we spoke about selecting a conversational AI platform for your digital assistant, as well as some questions to consider when making the decision.

In this report, we aim to give you a broad (and non-scientific) review of a few popular conversational AI platforms out there in relation to those considerations, to help you pick the best one for your situation.

Remember, there is usually no single platform that is better than the other in every aspect!

THE CONTENDERS

In this report, we will be reviewing and comparing the following four conversational AI platforms:

- **Amazon Lex**
- **Google Dialogflow**
- **IBM Watson Assistant**
- **Microsoft Bot Framework**

Fun Fact: These are listed in alphabetical order, but also happen to be in order of complexity of the dialogs you are able to build (more on that later).

All of the conversational AI platforms have a few **common features**, so we will not dive into talking about these in our comparison:

Intent Recognition

Create a model which recognises the user intent using example phrases as training data.

Entity Detection

Extract certain key pieces of information from the user's query. Such as the date, time, locations in the phrase: "Book me a flight from **Boston** to **London**, on the **24th of March**."

Slot-Filling

Define key pieces of information required from the user, and allow the conversation AI platform to ask for any pieces of information that are missing.

Slots: **from_location, to_location, date**
User: "Book me a flight from **Boston** to **London**"
Bot: "Sure, for **which date**?"

Webhook Integration

Ability for the assistant to call out to a custom endpoint for an external API, or internal API to your backend systems.

Versioning

Version your dialog so you can keep track of changes and revert back, if needed.

Conversational AI Platform	Strengths	Weaknesses	Pricing
Amazon Lex	<ul style="list-style-type: none"> • On AWS platform, so can be integrated with other AWS tools • Easy to use interface for simple dialog building 	<ul style="list-style-type: none"> • Difficult to create complex dialogs • Agent handover only works with AWS Connect, making it difficult to handover to other platforms • Basic web widget, compared to other providers 	<ul style="list-style-type: none"> • The lowest cost solution • \$0.00075 per text request with an additional \$0.004 per speech request, if you're using their text to speech service
Google Dialogflow ES	<ul style="list-style-type: none"> • A simple interface for simple actions • Mega agent functionality for combining many agents (skills) into a single agent • Built in knowledge base connectors for extracting answers from websites, powered by Google Search capabilities 	<ul style="list-style-type: none"> • Difficult to create complex dialogs that involve jumping different places, conditional logic, and re-use of old dialogs • No built-in agent handover • Would require a hack in with a custom function 	<ul style="list-style-type: none"> • A low-cost solution • Text requests are \$0.002 per request and voice is priced at \$0.0065 per 15 seconds
Google Dialogflow CX	<ul style="list-style-type: none"> • Great conversational designer interface - Probably the best we've seen so far for complex dialog • Designed for non-technical staff • Essentially a cloud version of Microsoft Composer, simply more user friendly • Very customisable dialog 	<ul style="list-style-type: none"> • Recently released - quite immature • Geared towards building IVR • Out of box web widget is in beta • Relatively more expensive 	<ul style="list-style-type: none"> • \$20 per 100 sessions • Not aggregated by users, such as Watson Assistant

Conversational AI Platform	Strengths	Weaknesses	Pricing
IBM Watson Assistant	<ul style="list-style-type: none"> • Friendly user interface • Usable by non-technical staff • Very powerful dialog editor - ability to do complex conversation flows • Best intent recognition • A few out of box service desk integrations, provides nice integration template for building handover to any helpdesk platform • Most slick looking web chat widget out of box 	<ul style="list-style-type: none"> • No ability to natively create "mega agents" (multiple dialog skills) • Linear dialog node editor, rather than flow chart like Dialogflow CX and Composer • Relatively more expensive 	<ul style="list-style-type: none"> • IBM's base paid plan (Plus) is priced at \$120.00 USD/Thousand Authorized Users, which is a different approach to pricing from the others (charged by request)
Microsoft Bot Framework SDK	<ul style="list-style-type: none"> • Extremely customisable dialogs and logic • Integrates with many channels through Azure Bot Service • Web frontend widget is mature and extensible 	<ul style="list-style-type: none"> • Hard to maintain dialog - must have developer that can run/maintain code • No out-of-box integration with helpdesk solutions for live agent handover 	<ul style="list-style-type: none"> • There is no cost to using Bot Framework SDK, but there is a cost associated with LUIS and Bot Service (more detail on p. 17) • There will be cost in self-hosting the server which will depend entirely on scale
Microsoft Bot Framework Composer	<ul style="list-style-type: none"> • Nice flowchart designer for conversation design • Usable by non-technical but still aimed towards developers • Quite customisable for UI • Integrates with many channels through Azure Bot Service • Web frontend widget is very mature and extensible 	<ul style="list-style-type: none"> • Custom agent handover • Designer is not cloud based, workflow is less ideal (dialog files are local) • Despite having a UI, it is still aimed at technical users 	<ul style="list-style-type: none"> • Pricing is exactly the same as Bot Framework SDK (see above)



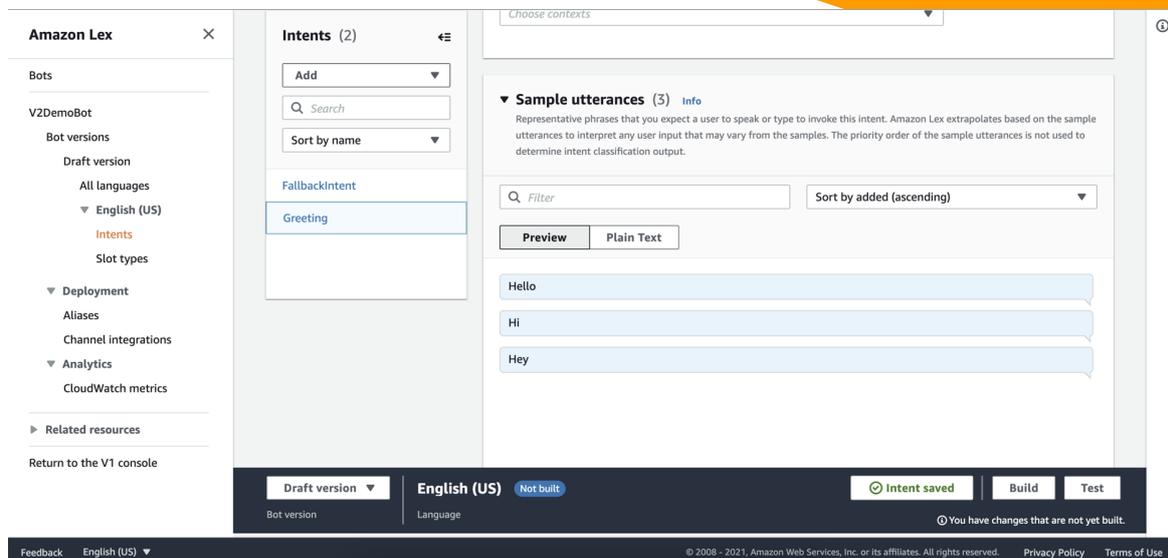
AMAZON LEX

Amazon Lex powers the incredibly popular Alexa virtual assistant. As a result, this platform is the go-to for building any Alexa Skill, which are apps that allow you to do more with Alexa such as turn on music, order food or answer the front door. Launched in 2017, it opened up the technology powering Alexa for developers to use in their own applications.

Amazon Lex provides basic intent authoring capabilities with an intuitive interface (V2 beta). It connects nicely with AWS applications, particularly if you're using Amazon Connect as a SaaS call centre platform.

As it was originally built for Alexa Skills, Lex's ability to handle complex interactions is somewhat limited, as you can't really have those in an Alexa interaction. These include interactions such as question-answer, simple forms and processes.

AWS has recently released their new V2 interface in beta, which in our opinion is much better than the first in terms of ease to use and features, particularly for non-developer users. It also brings in new features such as input/output contexts similar to Dialogflow ES.



Strengths

- On AWS platform, along with all the other AWS integrations.
- Easy to use interface for simple dialog building.

Weaknesses

- Difficult to create complex dialogs that involve jumping different places, conditional logic, and re-use of old dialogs.
- Agent handover only works with AWS Connect, with no easy way to handover to other platforms.
- Not the most stunning web widget out of box compared to some providers, but it can be customised with a few variables.

Pricing

- The lowest cost solution of all the contenders being reviewed.
- If you're using their text to speech service, \$0.00075 per text request with an additional \$0.004 per speech request.



DIALOGFLOW

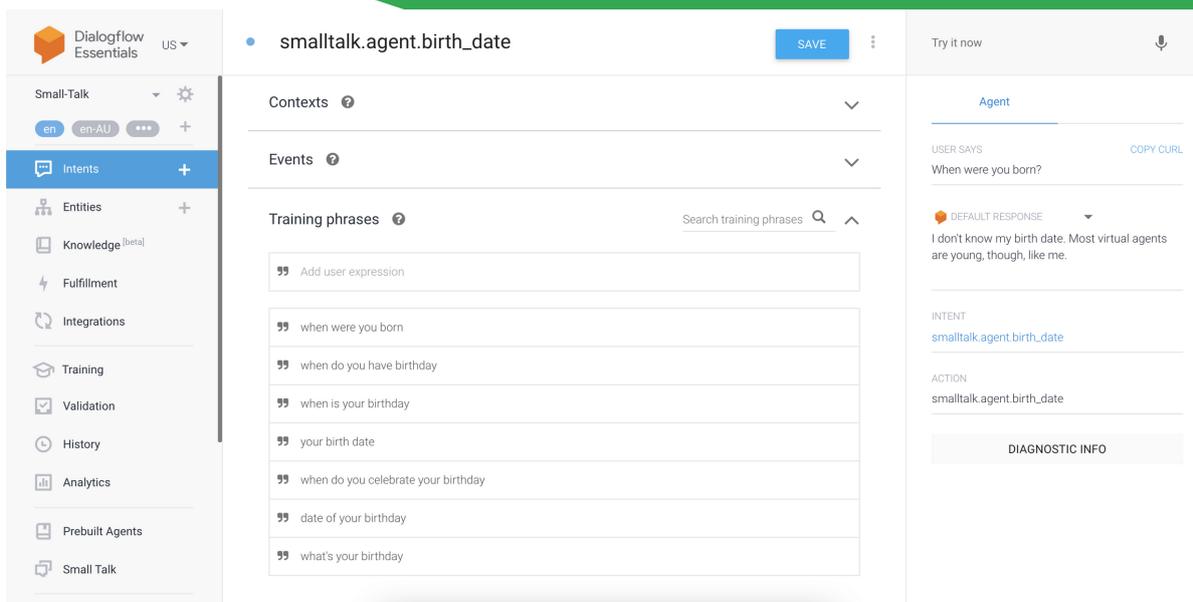
Google has two Dialogflow products - Dialogflow ES (Essentials) and CX (Customer Experience). Despite the similar naming, they're very different products, with different use cases and pricing:

DIALOGFLOW ES

This is the original Dialogflow, and probably the one you know if you've ever worked with it prior to 2020. Google acquired [API.ai](#) in 2016 and turned it into Dialogflow ES. Its capabilities are very similar to AWS Lex, and has always had quite an easy-to-use user interface for simple dialog.

DIALOGFLOW CX

Dialogflow CX (customer experience) is a recently launched conversational AI platform by Google. Launched in 2020, it bears little resemblance to Dialogflow ES.



Dialogflow ES integrates nicely with their voice assistant, Google Assistant, and is very suited to such interactions. However, much like Amazon Lex, ability to handle complex dialog is limited and can get quite convoluted with the use of input/output contexts.

One neat feature it has is their Knowledge connector, in beta. It allows you to connect "data-sources" such as websites, or FAQs, to supplement the assistant's ability to provide an answer on top its trained intents. This is built on top of Google Search, which we already know is incredibly powerful in finding relevant answers to questions. Note that you can also do the same with Amazon Lex (Kendra) and Watson Assistant (Discovery) but they're all separate services with a very steep cost (at least \$2000 USD per month). This is free in Dialogflow.

Another very handy feature is the ability to create mega agents. These are agents which connect several smaller, more specialised agents together through a router. When a user query comes in, it will first figure out which specialised agent to direct it to, before handing it off the particular agent to answer. This type of orchestration is very useful if you have lots of intents, that can potentially be confused with each other if they are all in a single model.

Strengths

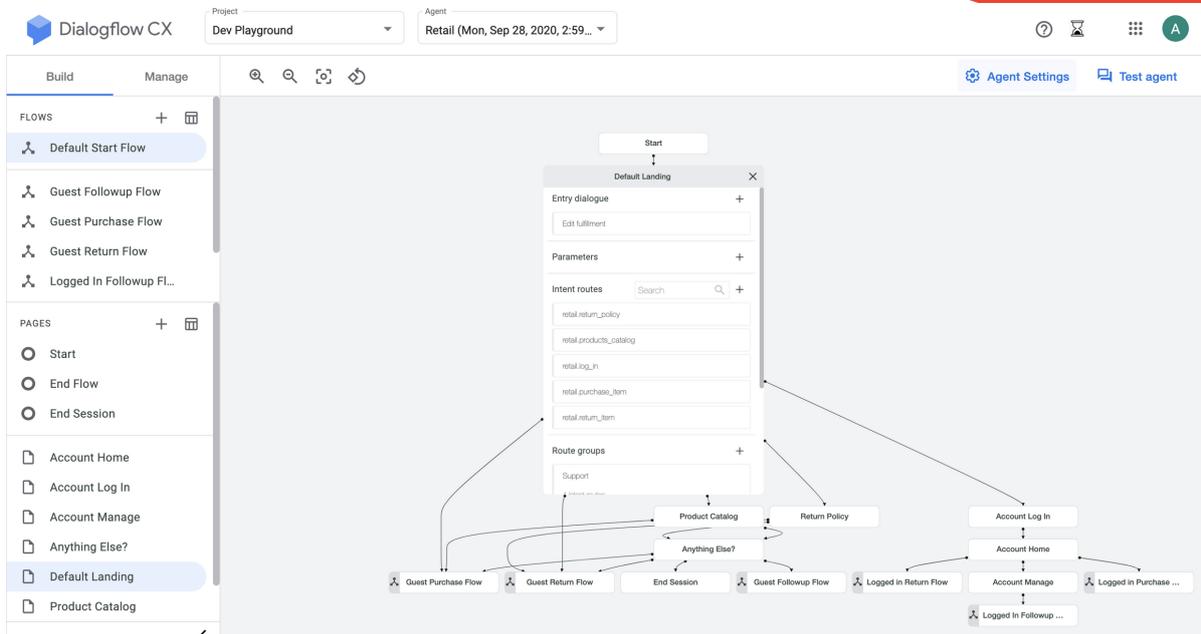
- Simple interface for simple stuff.
- Mega agent functionality for combining many agents (skills) into a single agent.
- Built in knowledge base connectors for extracting answers from websites, powered by Google Search capabilities.

Weaknesses

- Difficult to create complex dialogs that involve jumping different places, conditional logic, and re-use of old dialogs.
- No built-in agent handover. Would require a hack in with a custom function.

Pricing

- A very low cost solution.
- Text requests are \$0.002 per request and voice is priced at \$0.0065 per 15 seconds.
- If you're running a demo or a low request solution, you can use their free-forever trial, which has all of the above for free (with request limits).
- More pricing details [here](#).



Dialogflow CX brings much more powerful dialog authoring capabilities than Dialogflow ES, and is probably the most intuitive (of all of the platforms compared in this report) for complex conversations with its flowchart designer. It's not so much meant to replace ES as ES is better for certain use cases, such as taking a customer through a pre-defined flow with several paths depending on the user's actions.

However, it is a very new platform, so there are still lots of features ES contains that are missing in CX (such as knowledge connector), and the out-of-box integrations are still limited. It is also far more expensive than any other platform on the list.

Strengths

- Great conversational designer interface. Probably best we've seen so far for complex dialog.
 - Designed for non-technical staff.
 - Essentially a cloud version of Microsoft Composer, simply more user friendly.
- Very customisable dialog, can do all the logic we could think of.

Weaknesses

- Recently released - at immature stage of development.
- Geared towards building interactive voice response (IVR).
- Out of box web widget is in beta.
- Relatively more expensive.

Pricing

- \$20 per 100 sessions.
- Not aggregated by users, such as Watson Assistant.

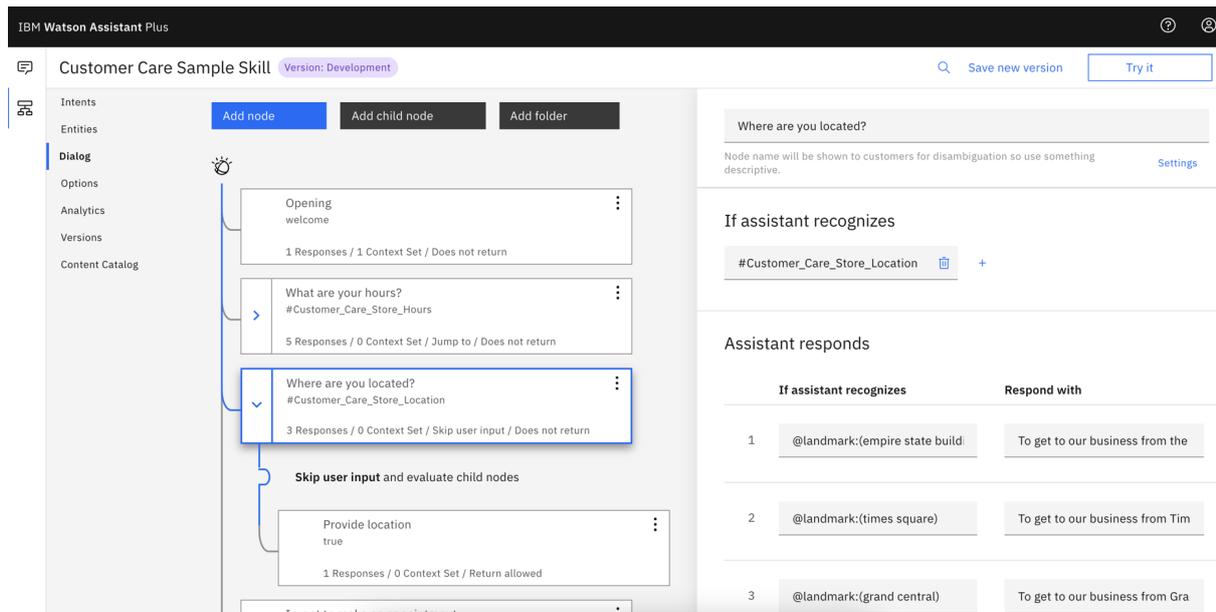


WATSON ASSISTANT

Launched in 2018, IBM Watson Assistant builds off the natural language processing smarts of Watson, that famously won Jeopardy in 2011. Although the state of NLP has come a long way since, Watson has continued to stay top in intent recognition benchmarks with their latest enhanced intent detection engine released in 2020 (beta).

We've tested it out against a few other platforms and it does perform noticeably better especially where there are a low number of training examples; it is able to generalise concepts better than the rest.

Another great feature of the dialog authoring interface is the dialog authoring system. Unlike Dialogflow ES and Amazon Lex, it separates Intents from Dialog. The advantage is this allows you to construct much more complex dialog pathways, making use of features such as jumps, digressions, context variables, and general dialog re-use. This is similar to Dialogflow CX and Composer, except it's presented in a linear fashion rather than a flowchart (which can arguably be slightly harder to grasp for non-developers). The downside is that for very simple assistants that are just question/answers, this can add unnecessary overhead.



The out-of-box web widget is probably the best of the bunch - with a very slick but customisable UI, well thought out layout (particularly with the addition of the welcome screen), and is recently the only one to feature out-of-box retention of conversation history as you switch between pages (all the other ones will lose the transcript/context when you refresh or navigate away from the current page). This is great if your primary channel is a widget on your website.

Strengths

- Friendly user interface - usable by non-technical staff.
- Very powerful dialog editor - ability to do complex conversation flows via the authoring interface.
 - Using context variables, digressions etc.
- Best intent recognition (from a test run Spark 64 conducted, as well as IBM's research).
- A few out of box service desk integrations, and also provides nice integration template for building handover to any helpdesk platform.
- Most slick looking web chat widget out of box.

Weaknesses

- No ability to natively create "mega agents" (multiple dialog skills).
- Linear dialog node editor, rather than flow chart like Dialogflow CX and Composer.
- Relatively more expensive.

Pricing

- Their base paid plan (Plus) is priced at \$120.00 USD/Thousand Authorized Users, which is a different approach to pricing from the others (charged by request).
- This places it on the order of magnitude as Dialogflow CX, or much cheaper, depending on whether you have repeat users or new users interacting with your assistant. Regardless, this puts it as second most expensive of the list.



BOT FRAMEWORK

The Microsoft offering takes a different approach to the other platforms in the list in that it's not a single all-in-one solution; there are several different components that can be put together to build a digital assistant. This modularity allows you to build out different architectures depending on the requirements of the solution. There are two common components amongst the two architectures:

Microsoft LUIS is the natural language understanding API which allows you to train an intent recognition engine to recognise a user's intent. It is powerful and probably gives you more customisation options than other NLU solutions. However, it does not give you the ability to define the dialog or responses of the digital assistant.

Azure Bot Service interfaces your digital assistant implementation to the many channels it supports, e.g. Facebook Messenger, Web (direct line), Teams, and more. It also provides a very mature and customisable web widget which you can embed into your website, which you can embed from the CDN or customise using React.

The final piece is where you implement the dialog and logic of your digital assistant. For a long time, that was Bot Framework SDK, but recently Microsoft also released Bot Framework Composer, which provides a visual interface upon the SDK for implementing conversations.

BOT FRAMEWORK SDK

Bot Framework SDK provides a highly customisable framework for implementing such content. The SDK came in both C# and NodeJS flavours, which are equally supported.

It provides a library of functions to allow you to elegantly manage conversation flows in code (which is much more thought-out than rolling your own), as well as tying into services such as LUIS. The biggest disadvantage, you can imagine, is that because it is a code-first framework, only developers will be able to maintain this.

BOT FRAMEWORK COMPOSER

Bot Framework Composer, announced in 2020, is a tool for implementing dialogs in a visual manner. Although it is still oriented to the technical user (rather than business), it still makes it much easier to maintain the dialog content, and visualise the conditional logic.

The tool will also automatically generate .LU (language understanding) and .LG (language generation) files and also deploy it to LUIS, making deployment for testing very simple.

Strengths

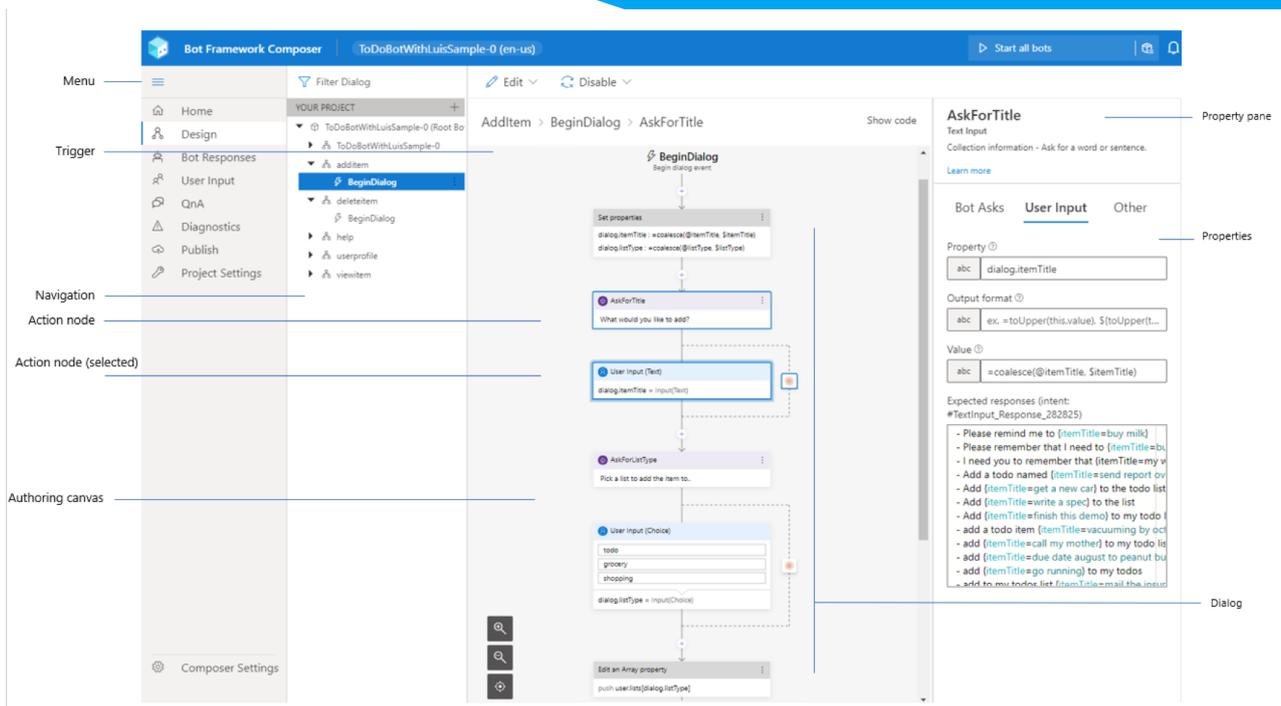
- Extremely customisable dialogs and logic (it's implemented code).
- Integrates with many channels through Azure Bot Service.
- Web frontend widget is mature and extensible.

Weaknesses

- Hard to maintain dialog. Must have developer that can run/maintain code.
- No out-of-box integration with helpdesk solutions for live agent handover (although there are some examples for implementing your own).

Pricing

- There is no cost to using Bot Framework SDK, but there is a cost associated with LUIS and Bot Service.
- Azure Bot Service
 - Free for standard channels (Facebook, Slack, Teams, etc).
 - For premium channels (web chat/direct line), it is priced at \$0.50 USD / thousand messages.
- Microsoft LUIS
 - The standard plan costs \$1.50 USD per thousand transactions, up to 50 transactions per second. Each transaction is equivalent to a message sent to the endpoint for prediction.
- Bot Framework SDK
 - As it is open source, there is no cost in using it. However, there will be cost in self-hosting the server which will depend entirely on scale.



Strengths

- Nice flowchart designer for conversation design - usable by non-technical but still aimed towards developers.
- Quite customisable for UI, as you can implement key logic with jumps and conditionals.
- Integrates with many channels through Azure Bot Service.
- Web frontend widget is very mature and extensible.

Weaknesses

- Custom agent handover.
- Designer is not cloud based, workflow is less ideal (dialog files are local).
- Despite having a UI, it is still aimed at technical users.

Pricing

- Pricing is exactly the same as Bot Framework SDK (see previous page).

There we have it; that's our summary, review and comparison of all of the conversational AI platforms offered by the major cloud vendors.

If we take a look at the list from top to bottom, **there's often a trade-off between ease-of-use and ability to customise dialogs**. Amazon Lex and Dialogflow ES both offer great features for digital assistants that don't involve long, complicated conversations with multiple paths and loops. Platforms such as Microsoft Bot Framework provide endless dialog logic and backend integration capabilities through a code-first framework. Then there are platforms such as IBM Watson and Dialogflow CX strike a balance in the middle, if you need a platform that is usable by non-development staff but also want the ability to serve highly complex dialogs and conversations.

These platforms are constantly changing, with new features and improvements in intent detection capability being released almost on a monthly basis for some.

If you have any questions about any particular platform, or would like us to help you decide based on your specific requirements, please don't hesitate to [get in touch!](#)

We'd love to hear from you!

Signing off,
The Spark 64 Team

enquiries@spark64.com