

ORACLE

Answer intent customization

About answer intents in Oracle Digital Assistant

” Answering frequently asked questions is a common requirement for digital assistants

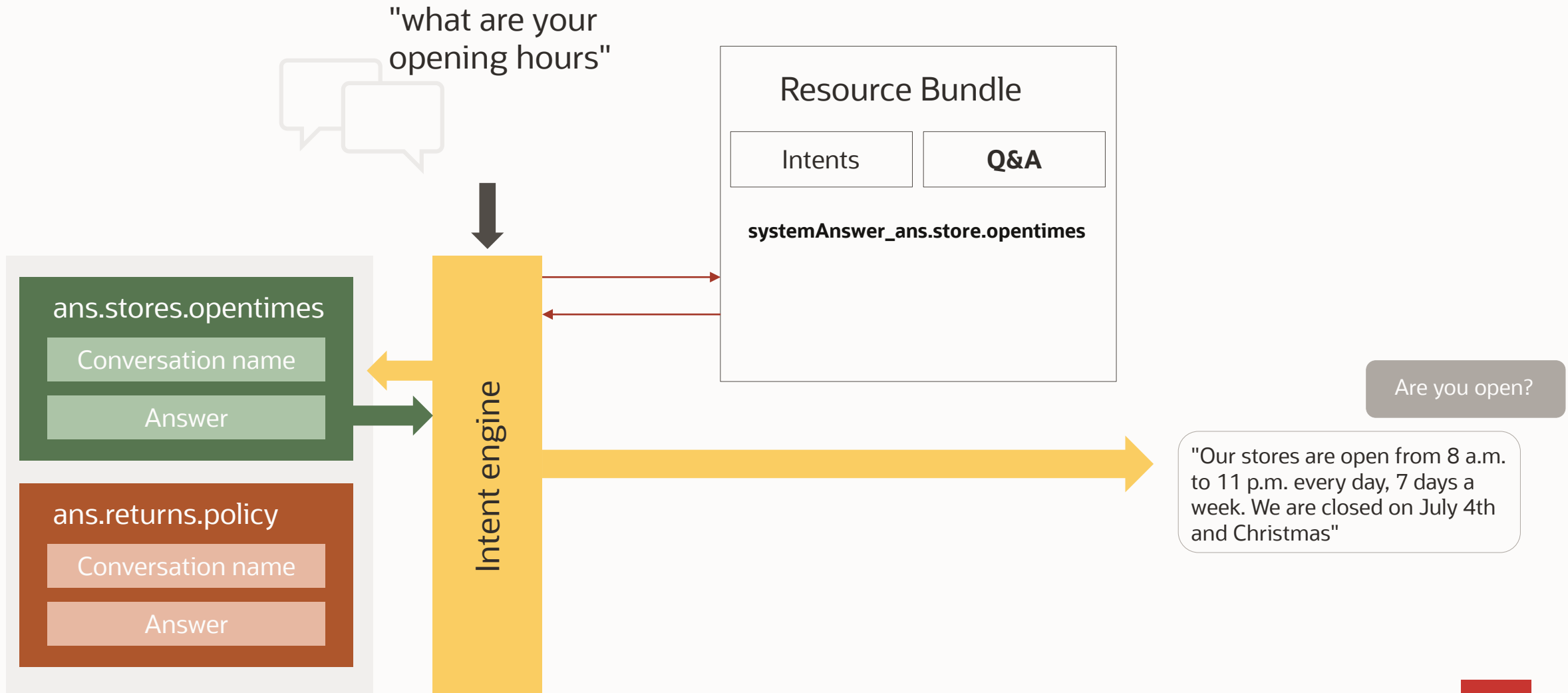
Answer intents use utterances to train the NLP model to resolve and display a pre-defined answer

Answers are saved in resource bundles, allowing multilingual FAQs to be created

No dialog flow is needed to handle frequently asked questions in a conversation

Using dialog flows can be used to improve the user experience for frequently asked questions

Answer intent - runtime architecture



How to ensure good quality answer intents

Make sure the answer intent model is well trained for a specific question

- Use between 70-100 sample utterances for a production bot of how users would ask for an answer
- Use synonyms and different sentence structures
- Create model tests

Curate all answers

- Ensure answers are conversational
- Avoid just copying FAQs from a website

Use the same persona you defined for your digital assistant when answering questions



”

Also, consider using answer intents for in-domain features that the bot doesn't support

Placeholder for features that are missing for a reason

Low cost of implementation

Guide users what to do instead

What you can customize using dialog flow(s)

Change the display of answers

- E.g., display a question on top of the answer to create context and clarity

Consider application or user context

- E.g., answer the question about opening hours based on a location

Engage with users

- Ask users for feedback
- Display related questions and answers
 - e.g., in a carousel of cards
- Encourage users to ask another question

Answers displayed using a dialog flow

Prints same answer as answer intent without flow

Mapped on the Main flow

- Built-in events | Answer Intent

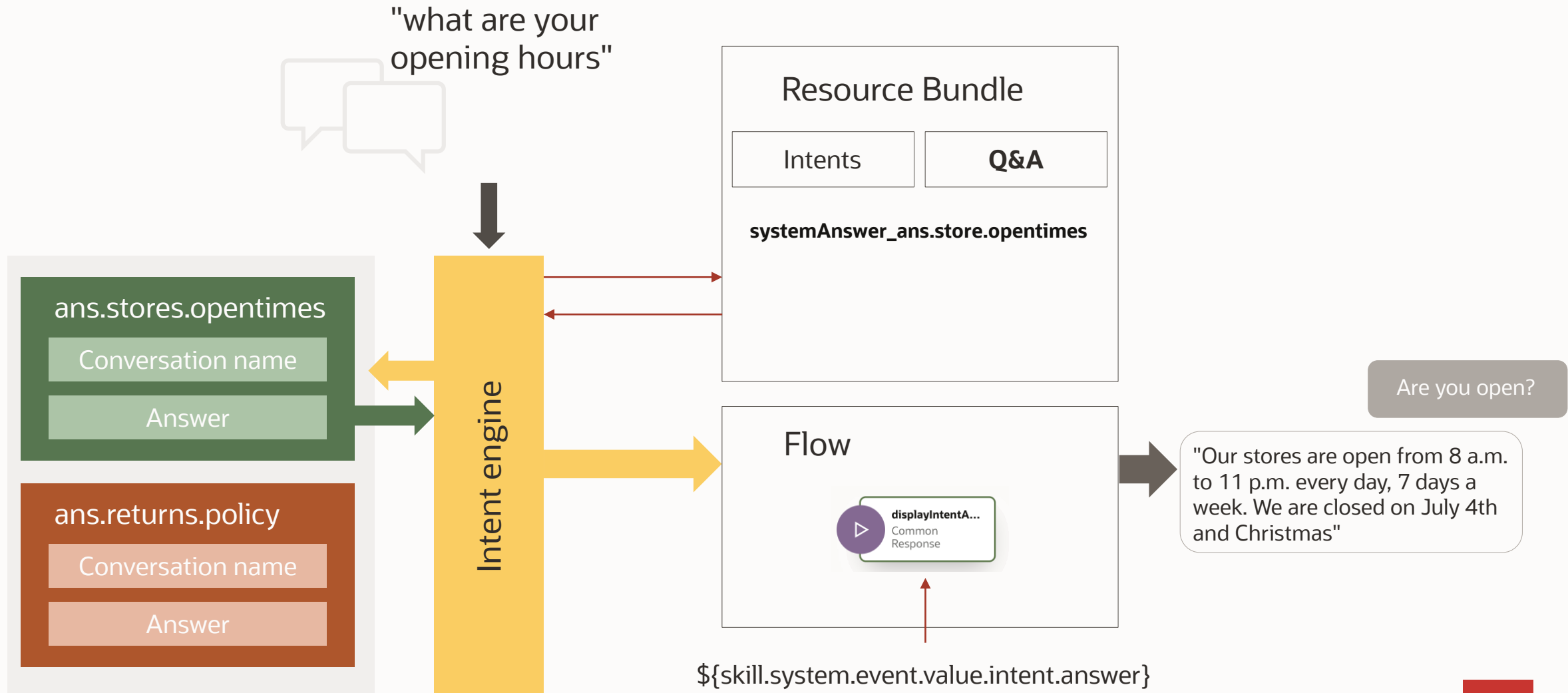
Allows you to add dialog flow states before and after answer is printed

- Prompt for additional information
- Navigate to a follow-up conversation

Component to print answer message added to flow as follows

- No coding required
 - Choose "Add state" menu option
 - In templates, select User Messaging | Display Multimedia Messages | Display Intent Answer

Answers displayed using a dialog flow - runtime architecture



Displaying a question on top of an answer

Add context to the answer

- Users may ask a question not mirrored in the answer
- Adding a question signal what the answer is about

Use intent's conversation name to define the question

- Conversation name is saved in a resource bundle for optional translation

Resource bundle entry for conversation name can be accessed using an expression

- Key name contains the name of the intent

Add a send message state in front of the answer intent

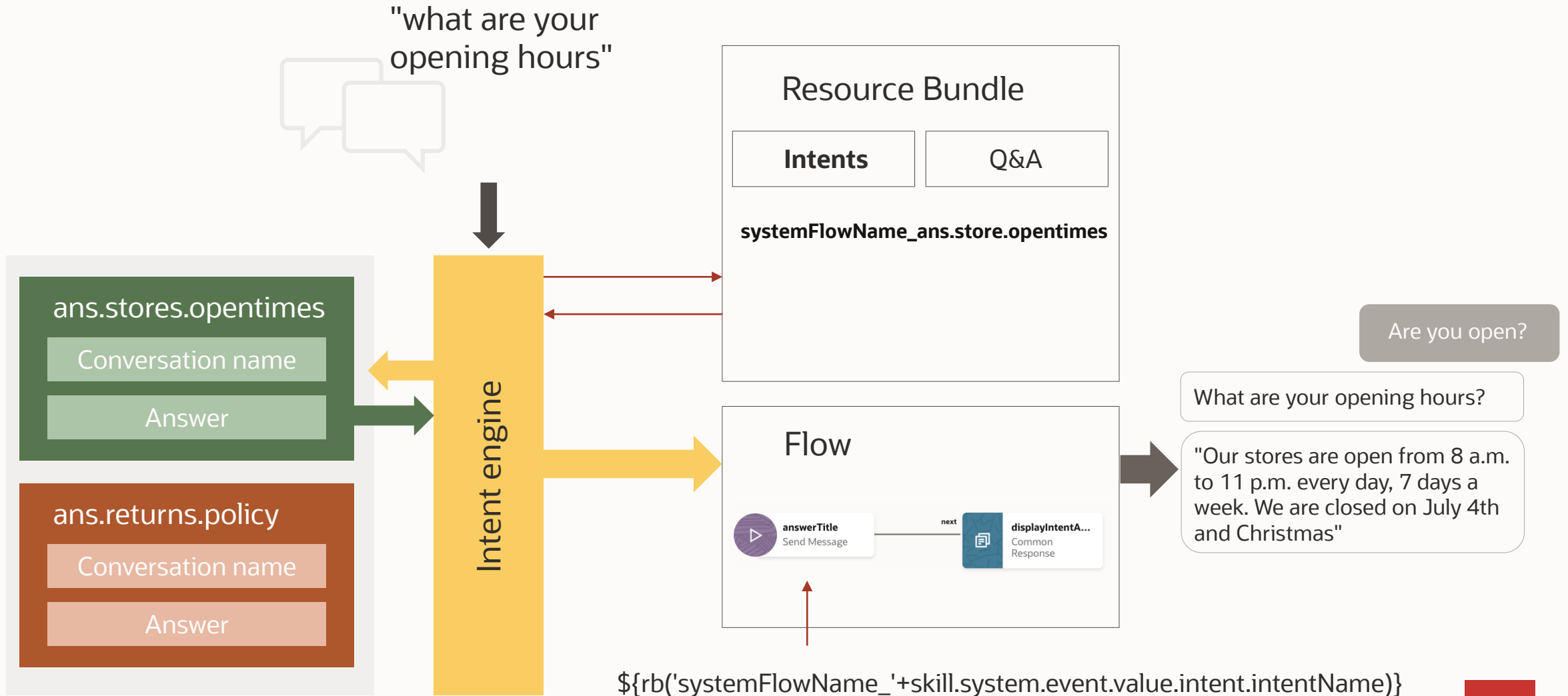
- Reference conversation name in resource bundles

The screenshot shows the 'Intents (2)' management interface. At the top right, there is a link for 'Test Utterances'. Below this, there are two buttons: a dark 'Add Intent' button with a plus icon and a 'More' button with a dropdown arrow. A search bar labeled 'Filter' is positioned below the buttons. A dropdown menu is set to 'Display Name Ascending'. The list of intents includes 'ans.returns.policy' and 'ans.store.openingHours', which is highlighted in green and has a trash icon to its right.

This screenshot shows a form field for 'Conversation Name'. The text 'What are your opening hours?' is displayed above the field. The field itself contains the value 'ans.store.openingHours'.



Answers displayed with question on top - runtime architecture



``${rb('systemFlowName_'+skill.system.event.value.intent.intentName)}``



Passing context information to an answer

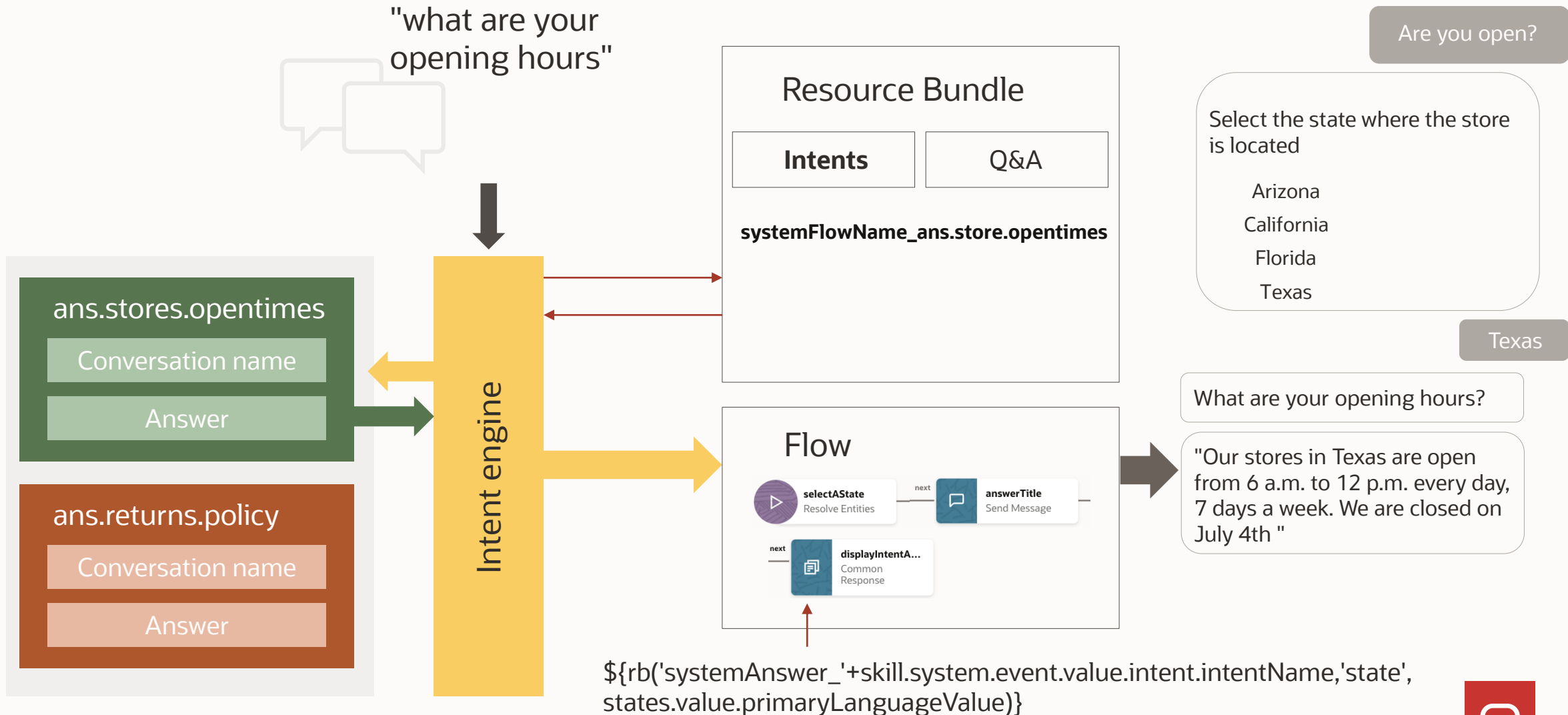
Answers to a question may depend on context

- The answer for a premium member may differ from the answer for others
 - "How much is the discount you offer on electronics?"
 - "Is shipping and handling free?"

Resource bundles can be parameterized

- To reference data in a message
 - "Your order is scheduled to ship on {shippingDate}"
- To conditionally print messages
 - {membership, select,
platinum { ... platinum member text ...}
premium { ... premium member text ...}
other { ... text ...}
}

Answers displayed based on context - runtime architecture



An in depth look at the resource bundle

Expression used in answer dialog flow state

```
`${rb('systemAnswer_'+skill.system.event.value.intent.intentName,'state',states.value.primaryLanguageValue)}
```

Reads state name from flow variable



Customized resource bundle message

User-Defined Intents **Q&A** Configuration

Keys (2)

Filter

- ans.returns.policy
- ans.store.openingHours**

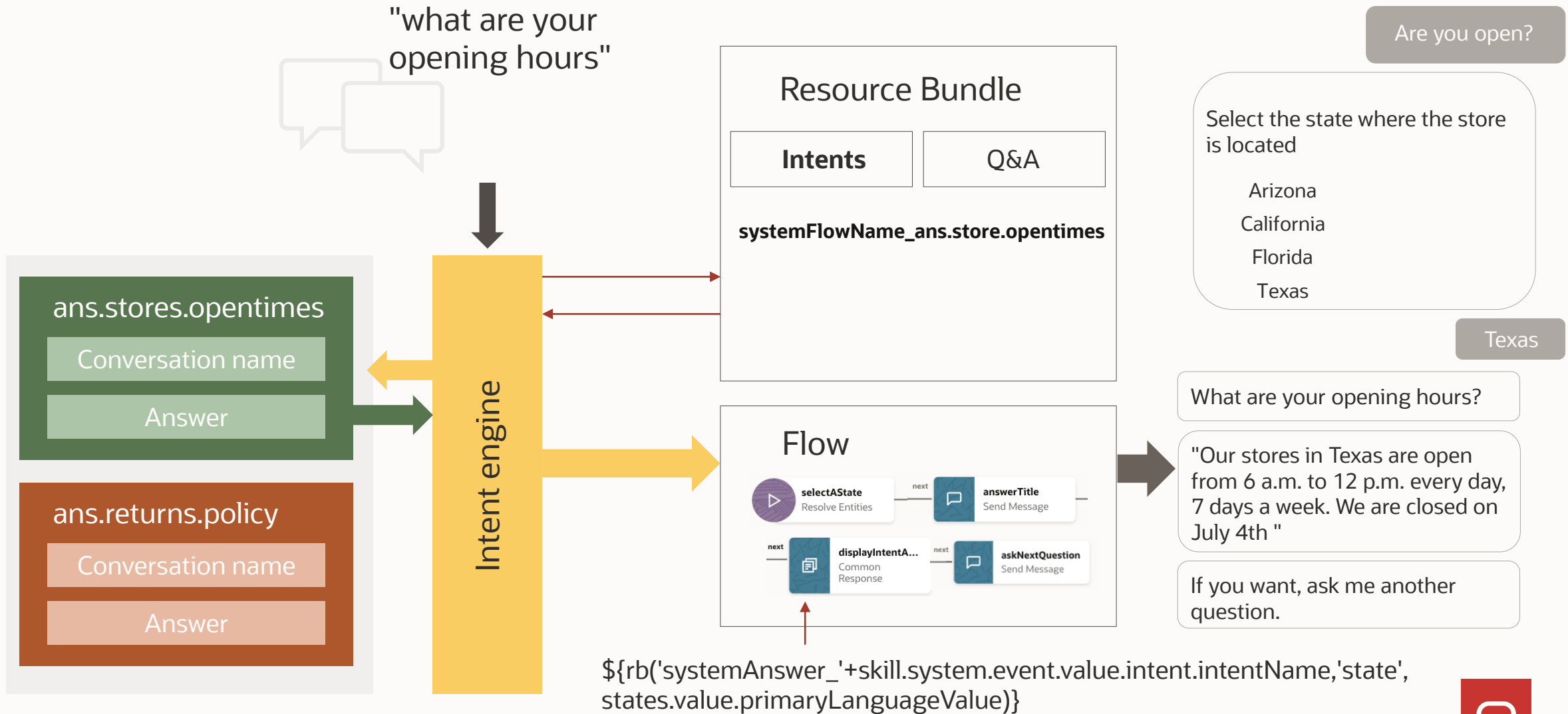
systemAnswer_ans.store.openingHours

+ Add Language

Language	Message
default	<code>{state, select, Texas {Our stores in {state} are open from 6 a.m. to 12 p.m. every day, 7 days a week. We are closed Arizona {Our stores in {state} are open from 10 a.m. to 10 p.m. from Monday to Saturday. We are closed other {Our stores in {state} are open from 8 a.m. to 11 p.m. every day, 7 days a week. We are closed Christmas}</code>



Answers display with follow up - runtime architecture



Answer intent customization best practices

If you have a need for answer intent display customization, create a flow to handle all answer intents

- In Main Flow, select built-in events | Answer Intent to map the flow as the handler for all answer intents

If you need to customize an answer different than others, create a separate flow for it

- In Main Flow, map answer intent name to flow name



Best practices in action

Filter

Display Name Ascending

Main Flow

UnresolvedIntent

ans.store.openingHours

global.answerFlow

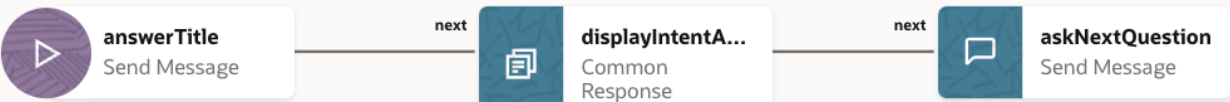
ans.store.openingHours

Flow Configuration



global.answerFlow

Flow Configuration



ORACLE