ORACLE

Visual Flow Designer Fundamentals

V1.2

Session Agenda

1. About dialog flows
2. Visual Flow Designer overview
3. Visual dialog mode architecture
4. Tips

Session Agenda

1. **About dialog flows**
2. Visual Flow Designer overview
3. Visual dialog mode architecture
4. Tips

About dialog flows

Define the narrative of a bot-user conversation in a skill

- Scripts the happy and unhappy paths

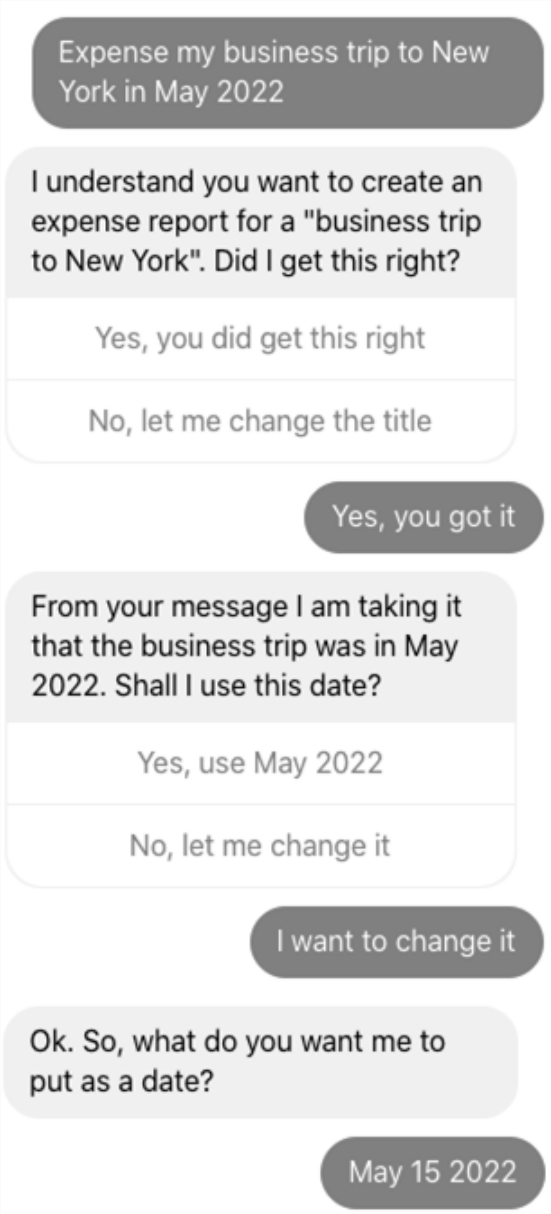
A dialog flow typically begins when a user message resolves to an intent

Use dialogs to prompt for user input or to display messages

Update variables

- User input
- Entity slotting

Remote backend queries and updates



About dialog flows in digital assistant

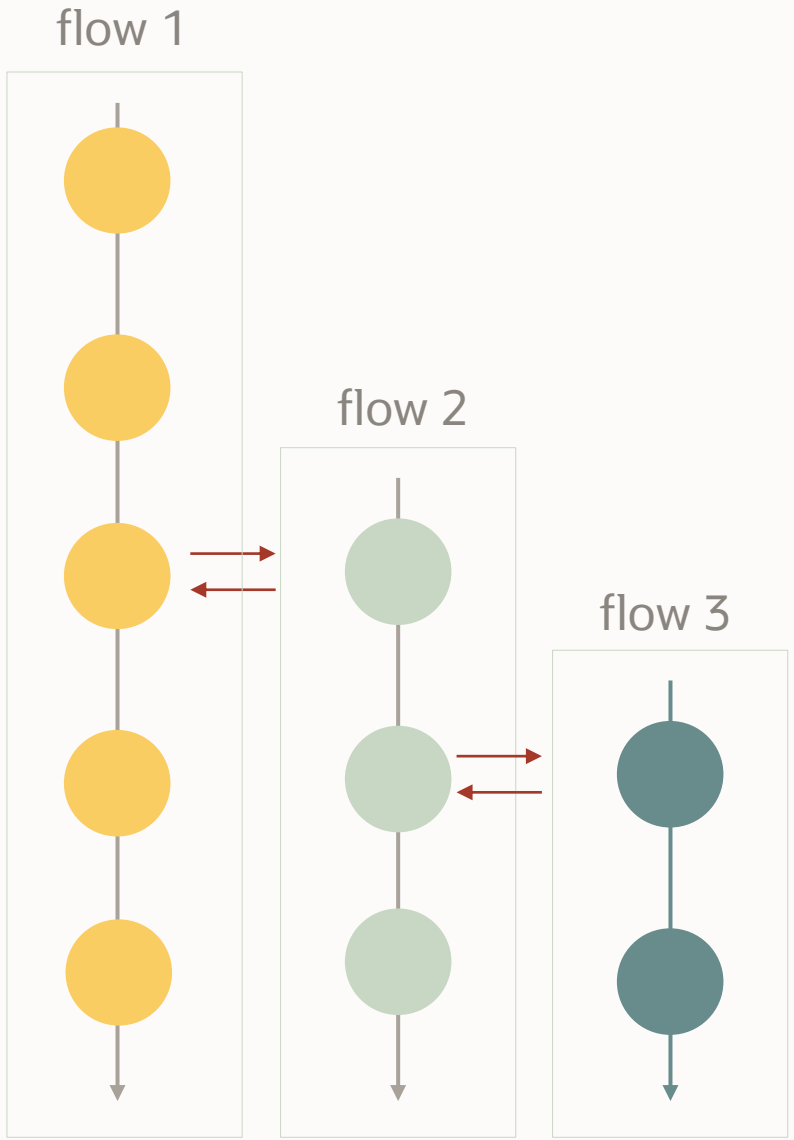
Flows make the conversation

Dialog flows in Oracle Digital Assistant can consist of a single flow or multiple flows chained together through invoke-flow calls

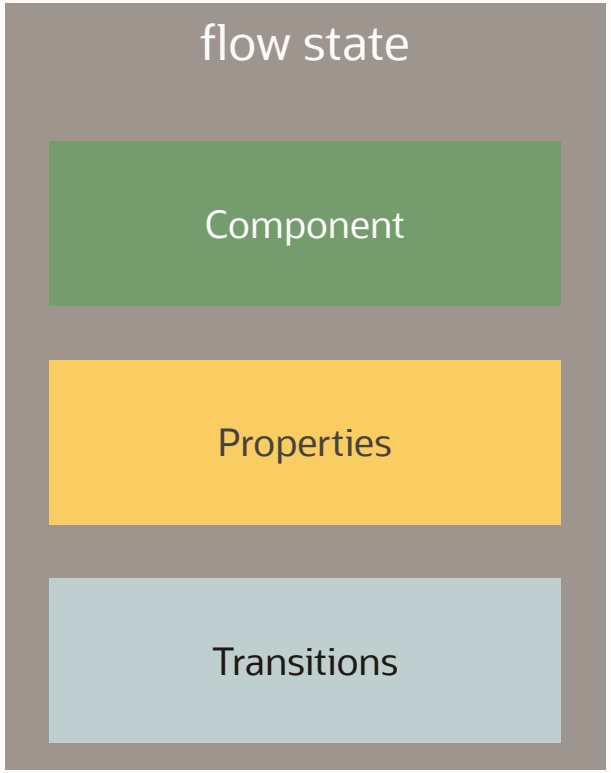
Flows consist of flow states that are visited in a specific order determined by user interaction with a skill

Each flow state refers to a configured component to

- render a user input prompt
- display user messages
- perform logical or functional operations



Flow states



flow states execute logic or render a user interface

Component types

- Built-in
- Custom

Properties

- Pass information into a component
- Used to update variables defined in a flow

Transitions

- next – navigates to a pre-defined next state
- action – conditionally navigates to a next state

About the Visual Flow Designer

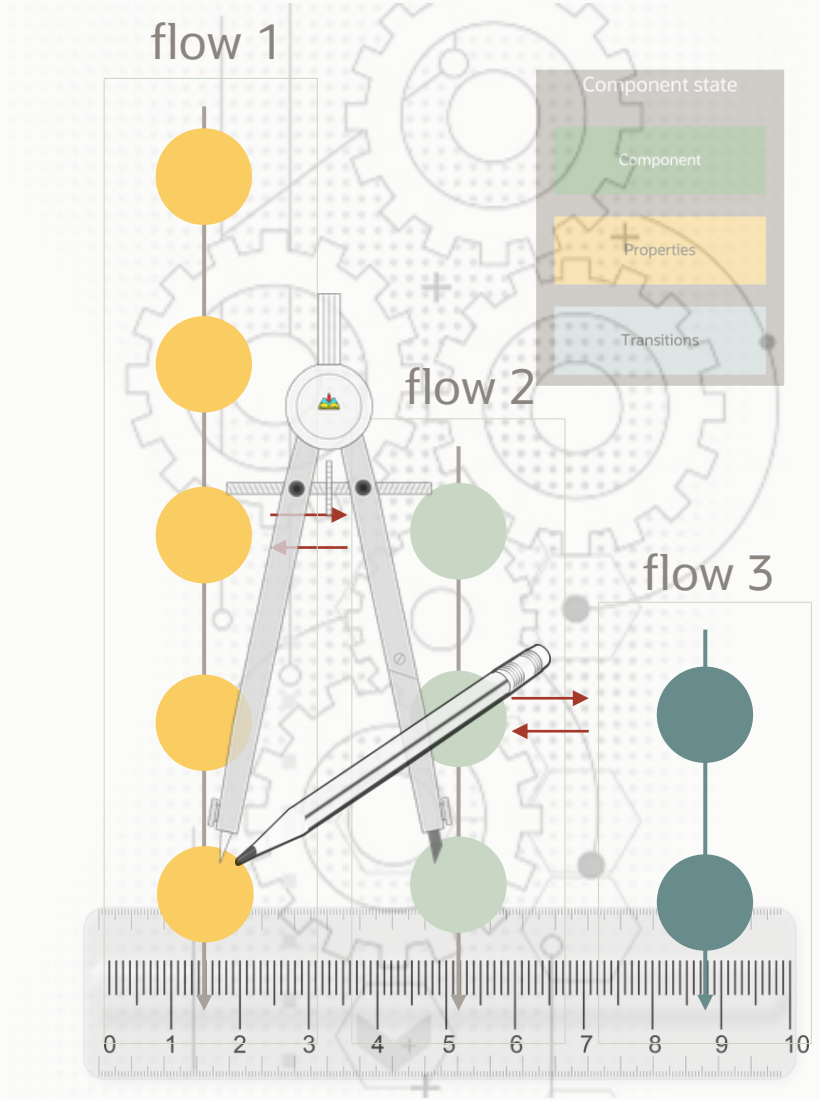
Declarative and visual tool to create and modify flows

Template driven state creation

- Driven by what you want to do (e.g., send a message)

For the component used in a state it does

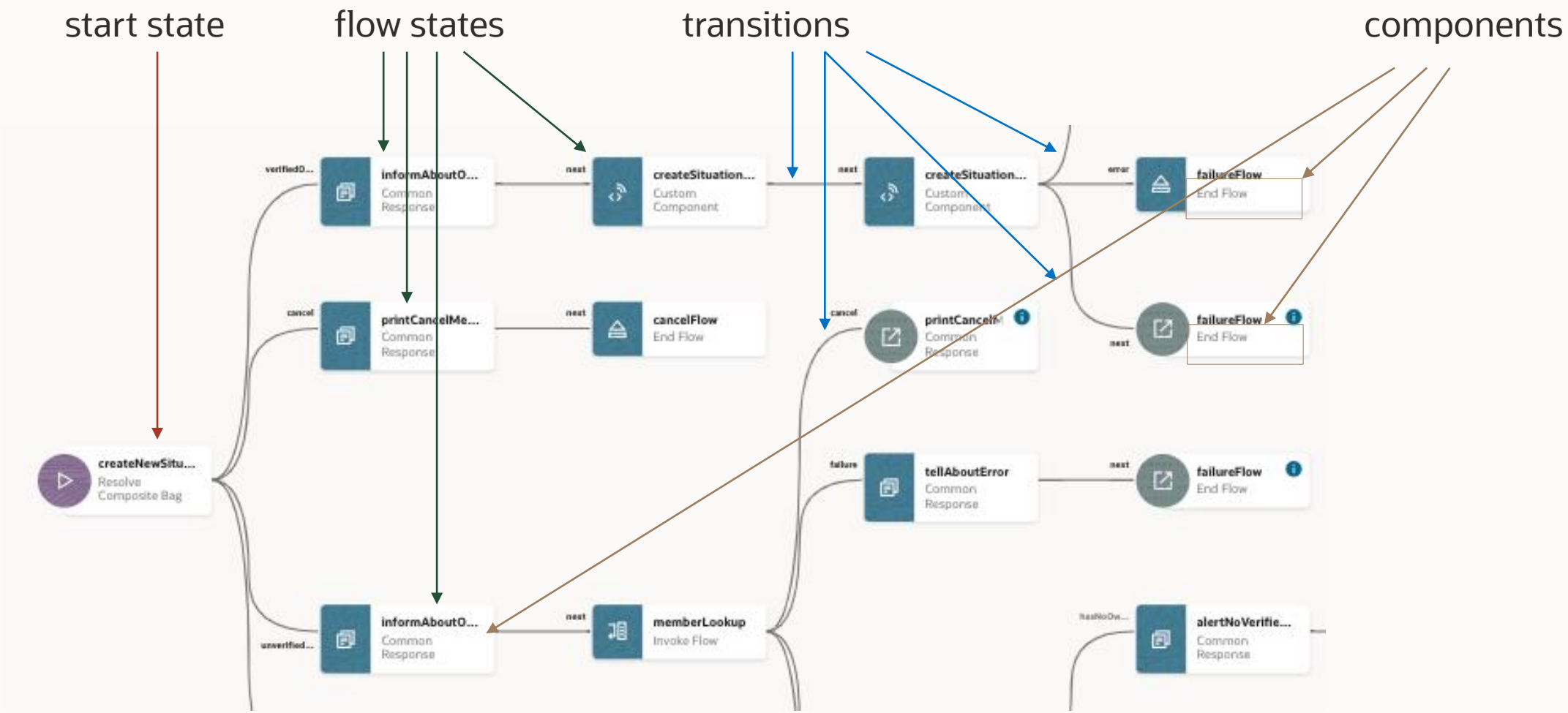
- Introspect and displays component properties
- Supports static values and expressions (Apache FreeMarker)
- Allows creating variables that will be updated by the component



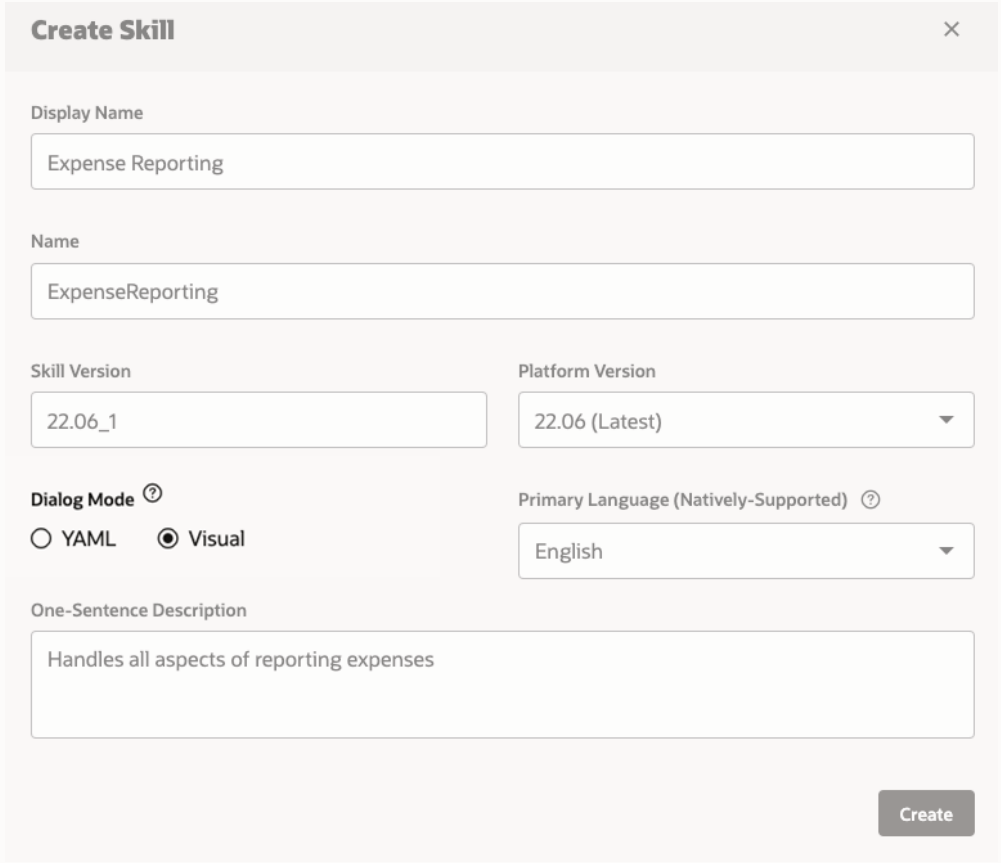
Session Agenda

1. About dialog flows
2. **Visual Flow Designer overview**
3. Visual dialog mode architecture
4. Tips

Visual flow designer example and terminology



Getting started



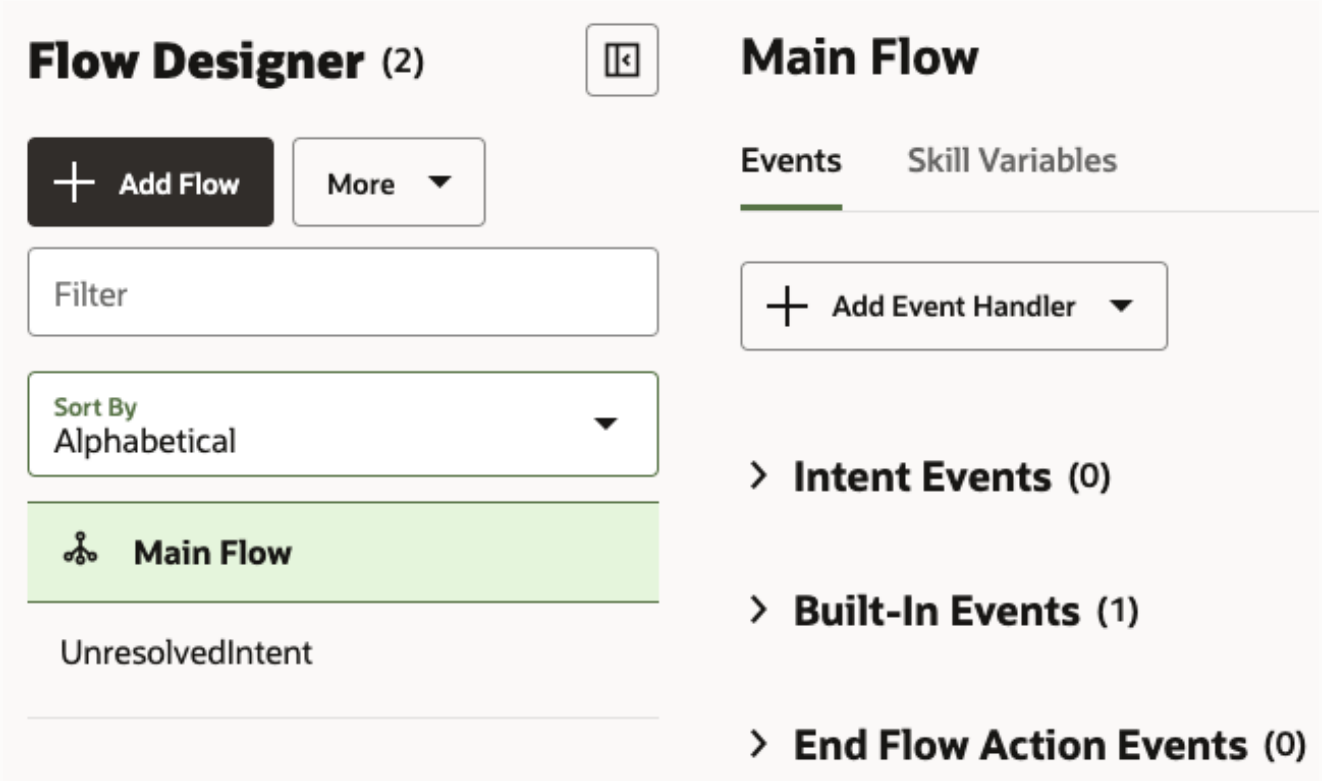
The visual flow designer is available in skills that were created using the **Visual** dialog mode option

The dialog mode option cannot be changed once the skill is created

YAML option no longer recommended

- Code centric development

Main Flow



Always there, cannot be deleted

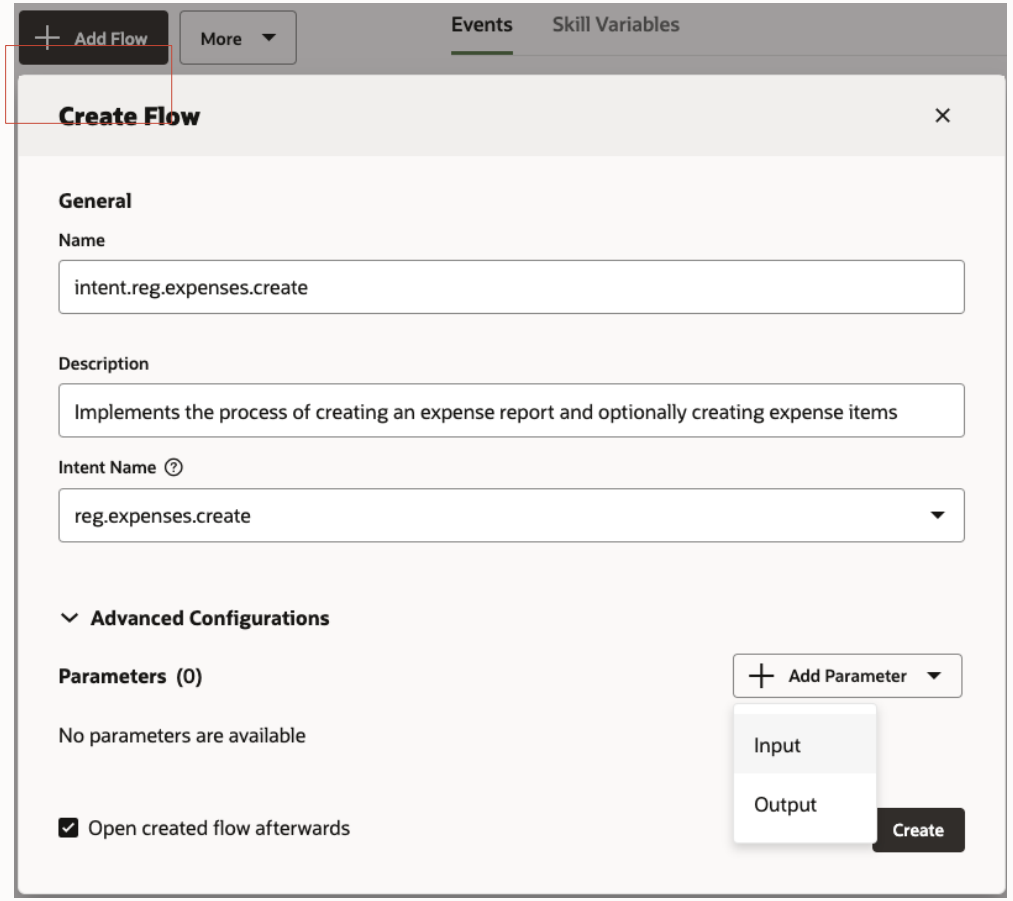
Maps intents and other events to flows

Allows to define global variables

- Variable values are kept until user session expires

Can pass information to mapped flows using input parameters

Creating a flow



There is no limit in the number of flows that can be created within a skill

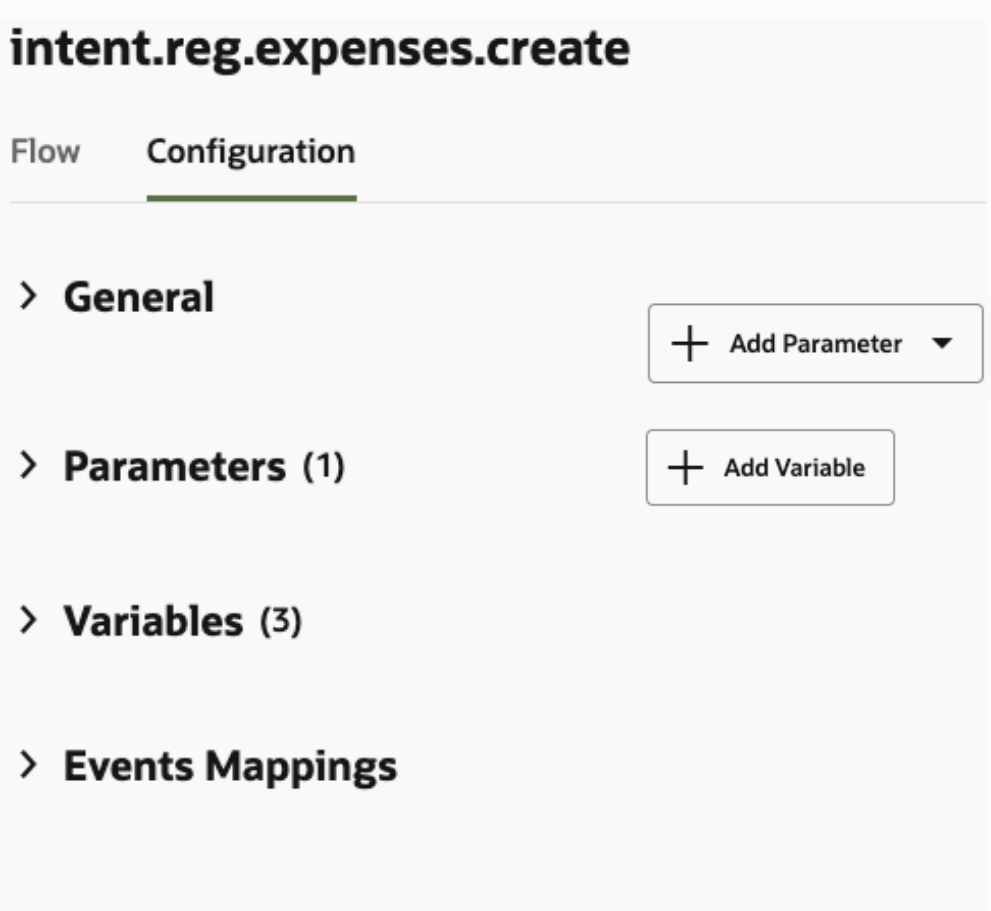
Flows can be mapped to an intent name but don't have to

- Flows mapped to an intent are started when the intent resolves in response to a user message

Flows can call other flows, pass information and receive information

- Input- Output parameters

Creating input, output parameters and variables



Use Configuration tab visible when selecting a flow

Input parameters and variables

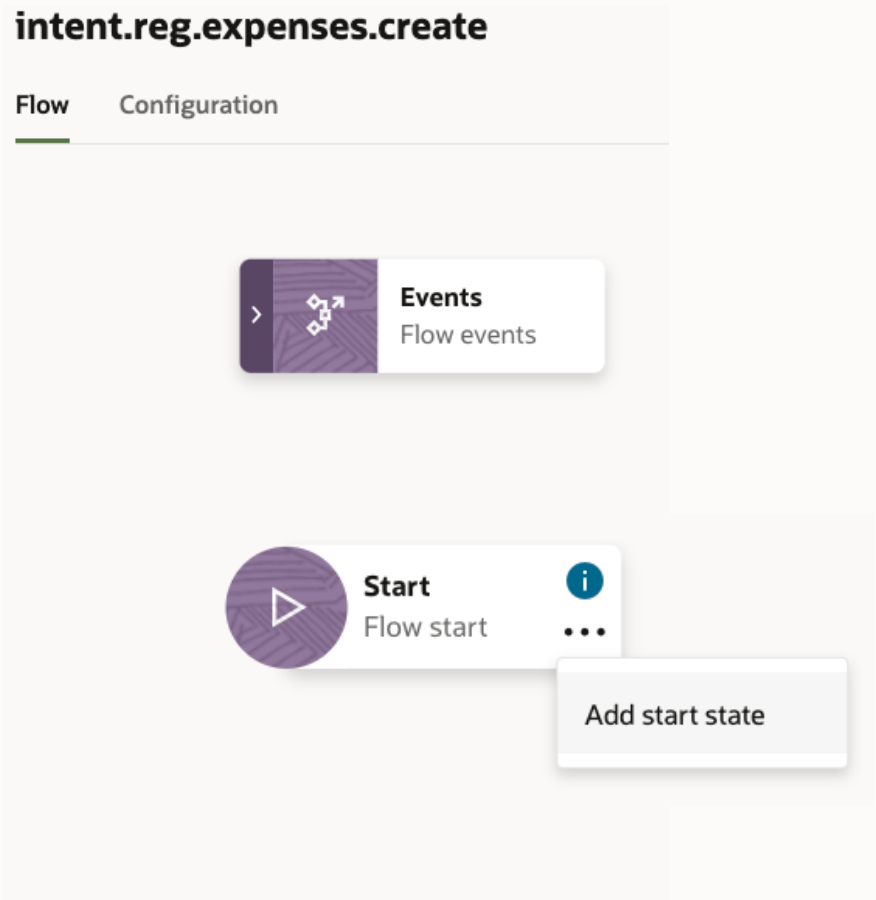
- Must have a unique name within the flow
- Have a type: primitive, complex, entity
- Can have an initial value set

Input parameters and variable values defined for a flow are reset when exiting the flow

Output parameters

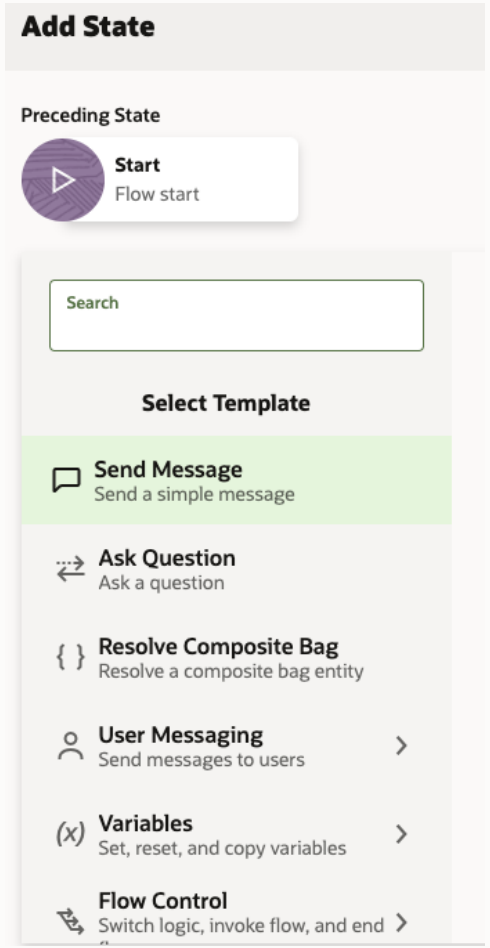
- Have a type: primitive, complex, entity
- Declaratively referenced when ending a flow

Adding states to a flow



Choose "Add state" on the "Start" state to begin composing the conversation flow

Adding states to a flow

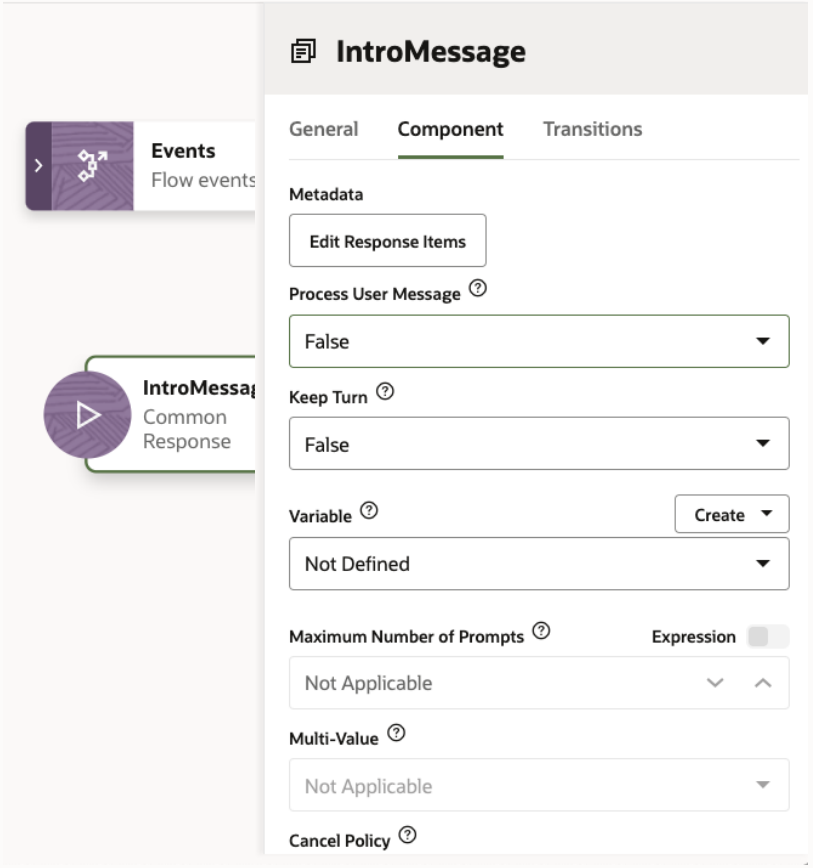


Choose "Add state" on the "Start" state to begin composing the conversation flow

A library of state template allows developers to select the functionality to add

- E.g., display a message, set variables, invoke a subflow

Adding states to a flow



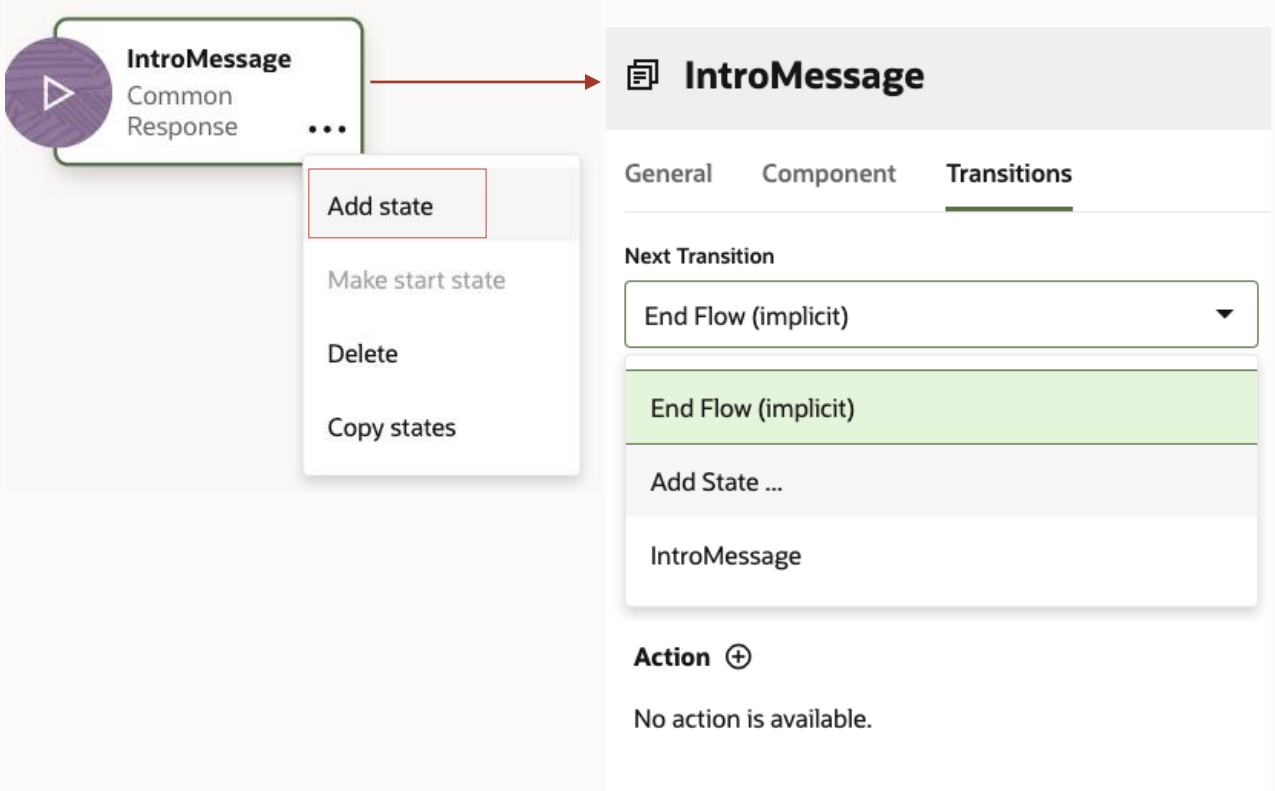
Choose "Add state" on the "Start" state to begin composing the conversation flow

A library of state template allows developers to select the functionality to add

- E.g., display a message, set variables, invoke a subflow

Selecting the added state displays a properties pane to configure the state's component properties and next navigation target (transition)

Adding transitions



Transitions determine the next state to visit within a flow

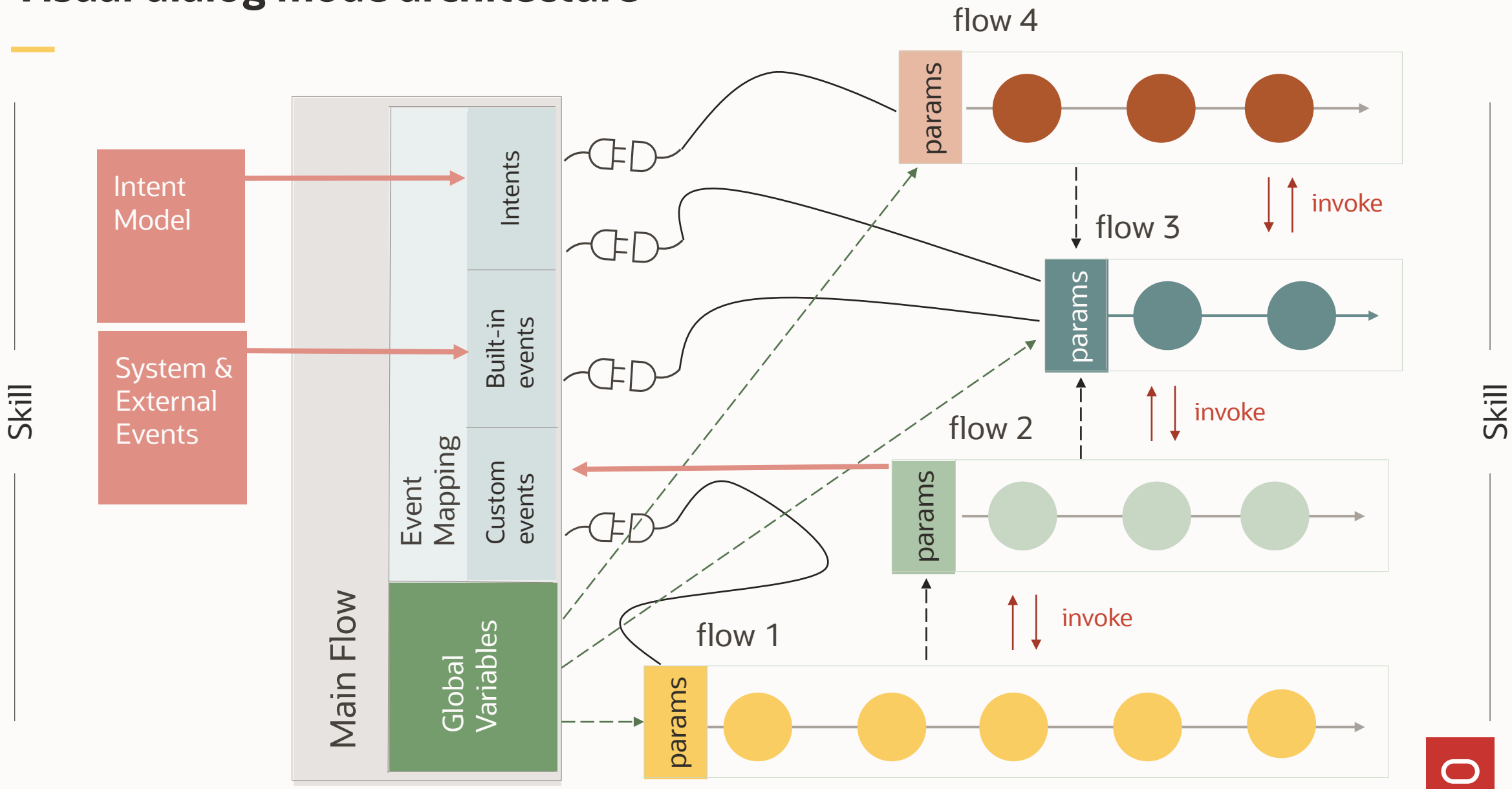
- Static navigation using "Next Transition"
- Conditional navigation using "Actions"

Components associated with a state determine the available actions

Session Agenda

1. About dialog flows
2. Visual Flow Designer overview
3. **Visual dialog mode architecture**
4. Tips

Visual dialog mode architecture



Session Agenda

1. About dialog flows
2. Visual Flow Designer overview
3. **Visual dialog mode architecture**
4. Tips

Recommendation for getting started

Build conversation flows incrementally

Select a conversation to build

- E.g., file an expense, check orders

Create the intent for this conversation

Create entities needed for and in the conversation

Create a flow for the intent

Create and call subflows for conversation logic that can be reused from other flows

Build self contained flows

> Think 'modularization' when building flows

Use input parameters and output parameters for communication between a parent and the subflow

Use action strings in end flows to indicate to the parent flow what to do next


ORACLE