OSF DIGITAL

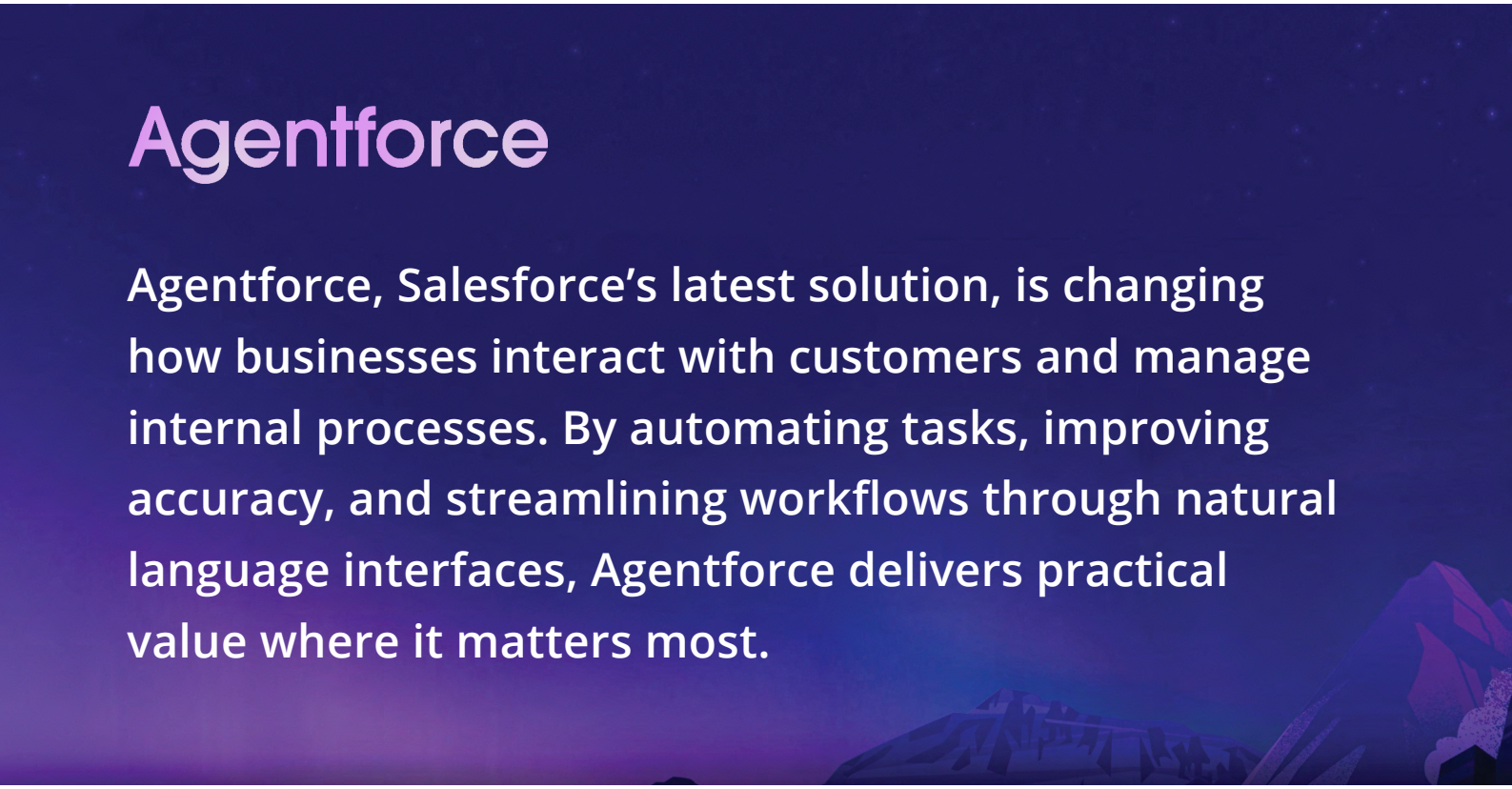
Agentforce Explained: How It Works and Why It’s Different from Bots

PART I

osf.digital

Table of Contents

Introduction



Agentforce, Salesforce's latest solution, is changing how businesses interact with customers and manage internal processes. By automating tasks, improving accuracy, and streamlining workflows through natural language interfaces, Agentforce delivers practical value where it matters most.

This white paper explores the power of Agentforce, outlining how it operates to deliver real-world value. We'll walk you through its core features, examining how agents differ from traditional bots by offering greater flexibility and the ability to handle complex tasks, thereby empowering employees and improving customer interactions. We will also highlight its ability to automate tasks, integrate seamlessly with existing systems, and maintain high levels of control and security. With practical examples and insights into how businesses can leverage Agentforce, this paper demonstrates how the solution can drive efficiency, enhance customer satisfaction, and maximize the value of your Salesforce investment.

For a deeper dive into Agentforce's key capabilities, real-world use cases, and competitive positioning, don't miss Part 2 of our white paper series.

Overview of Agentforce

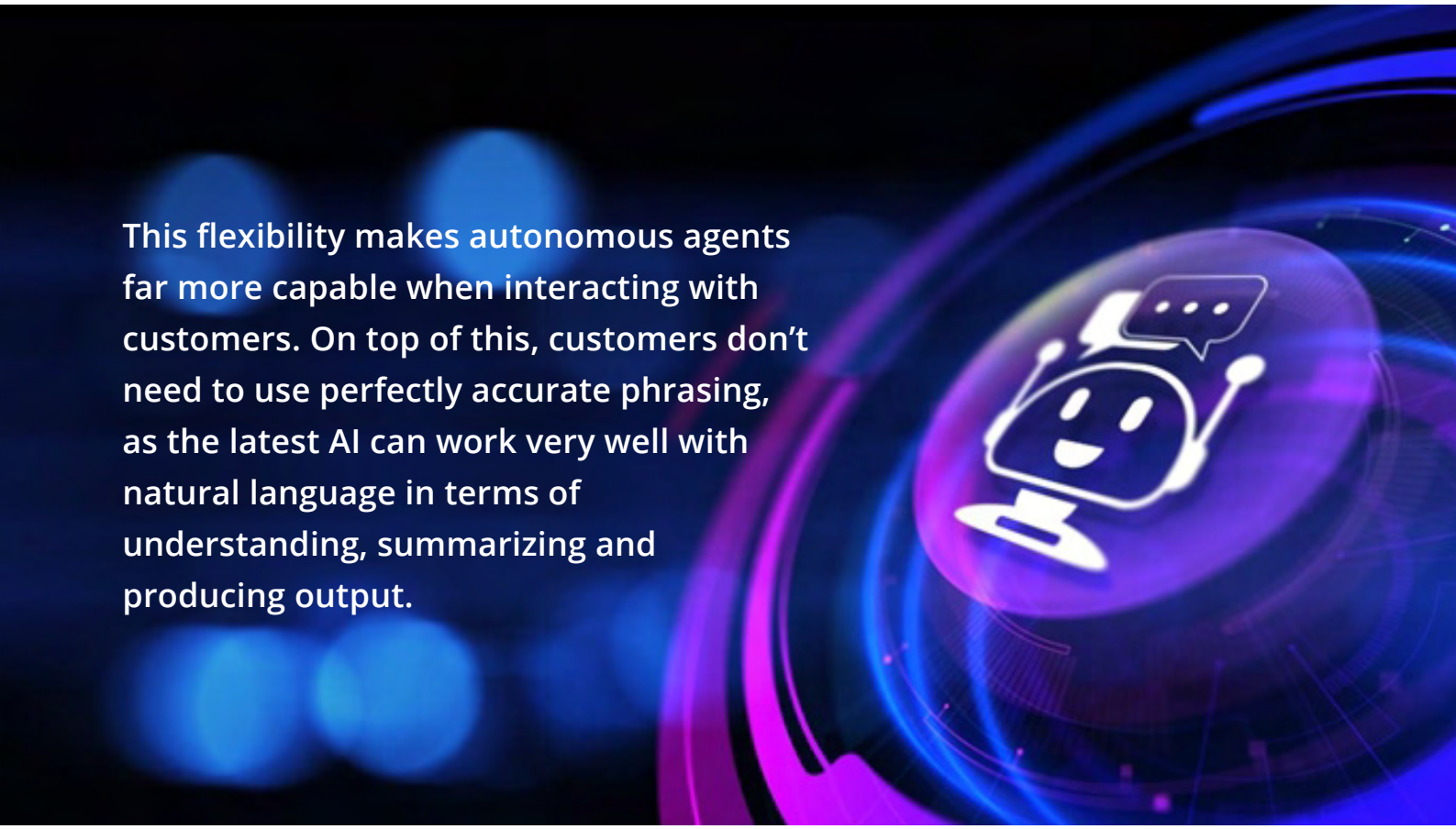
What is an autonomous agent?

First, let's define what Agentforce is targeting: the deployment of autonomous agents to work with customers and other stakeholders like team members. Their tagline is: "Humans with Agents drive customer success together." The idea is not to completely replace humans with agents but to assist and empower humans to do what they do best. In reality, agents will take over certain aspects of human work, primarily repetitive, business rule-driven processes.

An autonomous agent is defined as "a system or entity capable of making decisions and acting independently, without the need for direct human control or intervention." These agents leverage the latest advances in AI to reason and decide on courses of action based on the input they receive.

How is it different to a bot?

Bots, commonly seen as chatbots, are typically simplistic pieces of technology. They pattern-match a request using trigger words to determine the appropriate predefined path to follow. However, bots can only operate within these predefined paths, making them deterministic.
By contrast, an autonomous agent uses AI to understand what the human wants, leveraging advanced AI reasoning engines to match the request to a possible set of actions it could take. It can then execute those actions, sometimes in different orders, and not along predefined and fixed routes.



This flexibility makes autonomous agents far more capable when interacting with customers. On top of this, customers don't need to use perfectly accurate phrasing, as the latest AI can work very well with natural language in terms of understanding, summarizing and producing output.

Agents are to bots what ChatGPT is to standard Google Search, to give a comparison.

How has Salesforce seen autonomous agents?

Autonomous Agents can sound scary - especially with phrases like “without the need for direct human control.” The impressive part about Agentforce, is how well protected and controlled the process of creating and operating an agent is with Salesforce.

Salesforce describes an Agentforce agent as:
“An Agentforce Agent is a proactive, autonomous application that provides specialized, always-on support to employees or customers. They're equipped with the necessary business knowledge to execute tasks according to their specific role.”

Let’s unpack this statement a little further. For now, though, let’s consider that because these agents are operating within a strict setup on the Salesforce platform, we have a lot of fine-grained control on how they operate. This is not free-for-all, “ask me anything” ChatGPT application. It is an agent with as narrow a purpose as you want to give it.

- **Proactive:** Agents are not limited to reactive interactions in chat environments. They can proactively analyze yesterday’s survey data for sentiment, provide real-time alerts on low stock items, and more.
- **Autonomous:** Yes, these agents act independently—but so do current chatbots, and even human team members, regardless of their level of experience or training.
- **Specialized:** This is a key concept. Each agent is narrowly defined in its skill set by you. If it doesn’t understand a task, it will refuse to act.
- **Always-On:** One of their key benefits is the ability to serve customers 24/7, maintaining a consistent level of service around the clock.
- **Business knowledge:** These agents are grounded in your policies, business rules, and data, ensuring they operate within the parameters you’ve set.

Salesforce's Vision for Agentforce

Salesforce has a clear vision for Agentforce: it views agents as the next frontier of productivity and customer success, particularly for its customers. Salesforce aims to enable the creation of hundreds of thousands of these agents in the coming years.

While third-party AI solutions come in many shapes and forms, there is a common underlying set of issues companies are facing:

Trust

Many new AI startups face trust issues, such as concerns over data usage—Are you using my data to train your model? Salesforce has a clear technical and ethical standpoint on trust through its Trust Layer.

Data

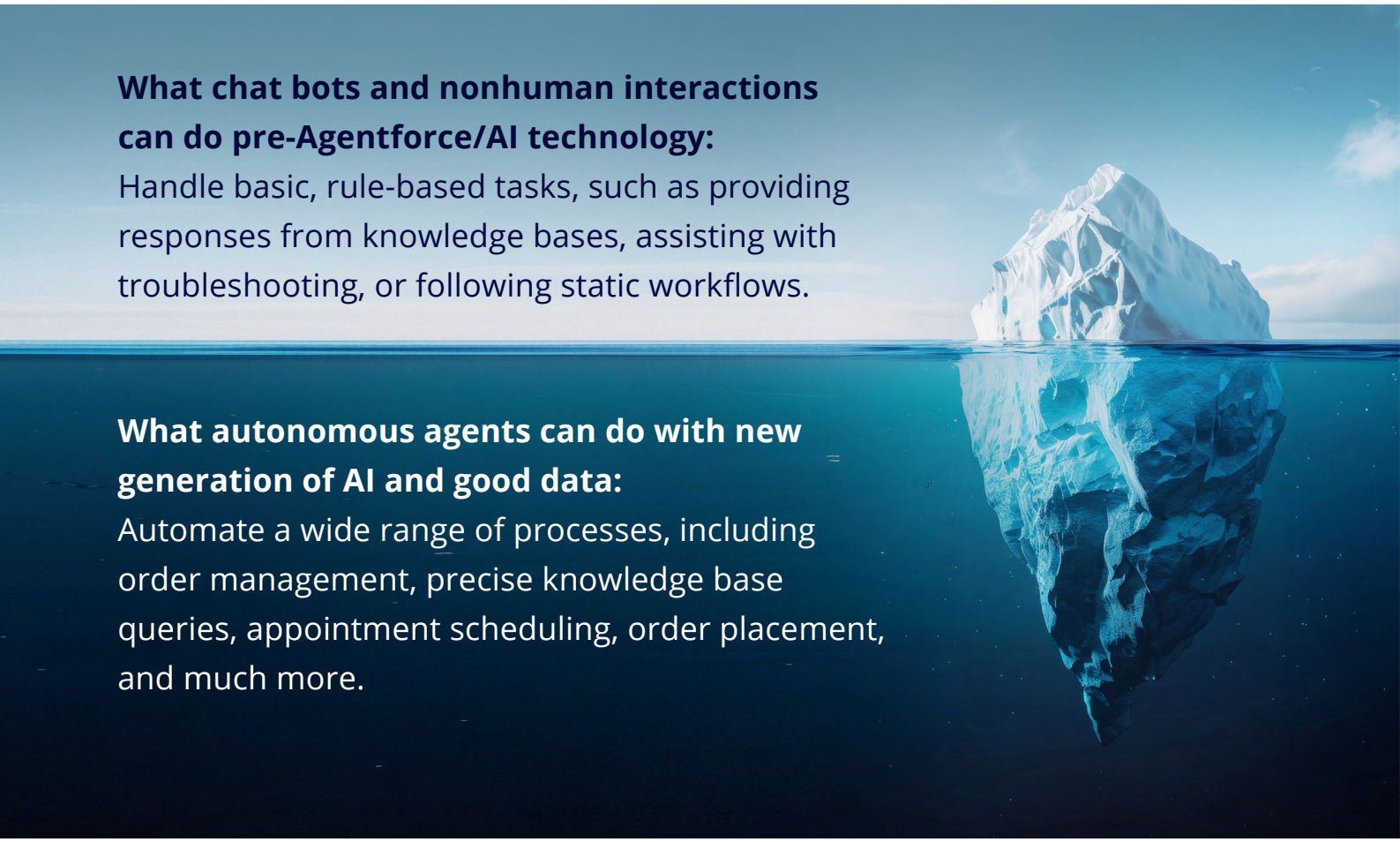
Effective agent operation requires data. For most customers, the Salesforce platform already contains the exact data needed to power these experiences. Additionally, Salesforce offers Data Cloud to integrate and enrich this data further.

Domain

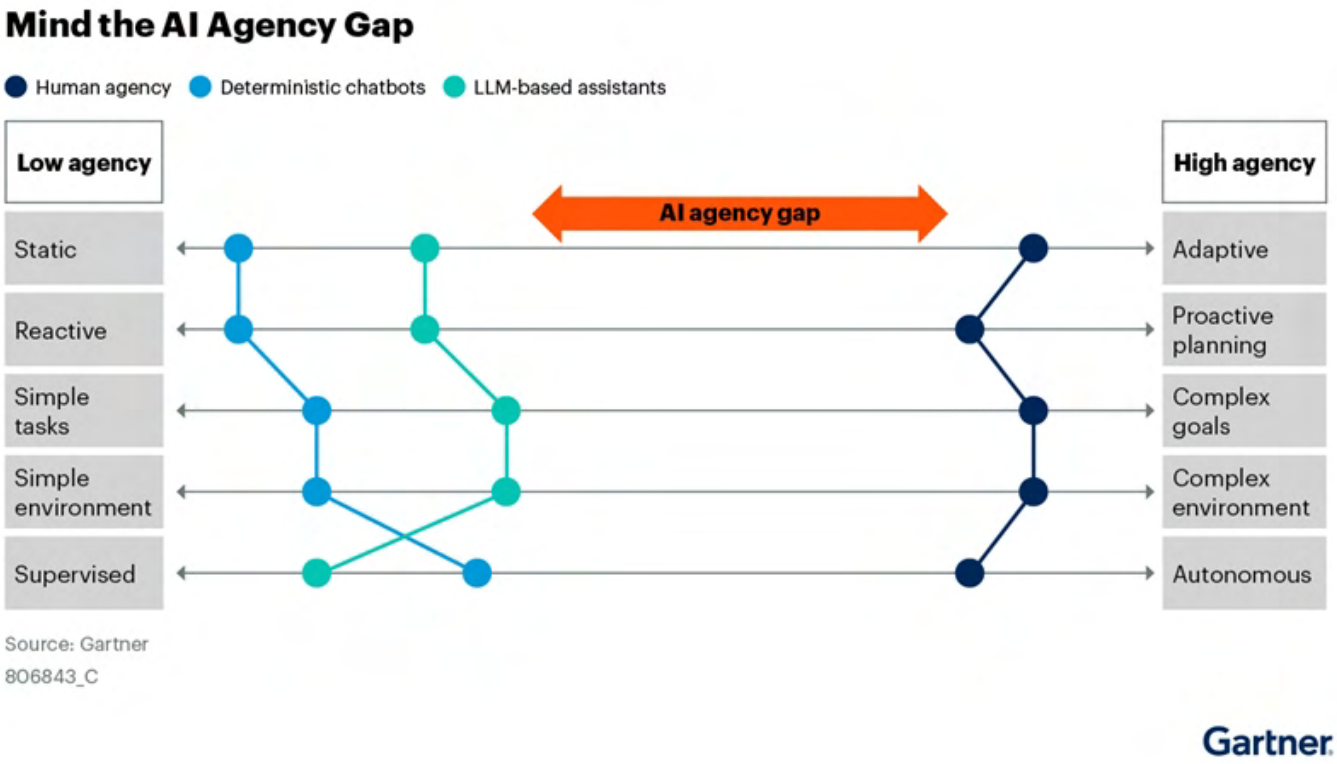
Salesforce operates in the front to middle office: commerce, marketing, service, sales, etc. These domains all lend themselves to agents.

OSF's Vision for Agentforce

We believe that Agentforce is already proving its value. This is demonstrated by the customers we support and the growing interest in the product. The way that Salesforce has set up the guardrails and trust topics gives customers great, helping reduce concerns about deploying autonomous agents.
Our goal is to help customers take full advantage of this new frontier of productivity and ROI.



There are still many processes that require a human skill set and abilities. The idea of agents seamlessly transferring tasks to humans—and back—guided by the guardrails, ensuring they operate strictly within their defined capabilities.



According to Gartner's AI Agency Gap Model, there is a substantial gap between the AI capabilities available today and true human-level agency (represented on the right). This is a valid observation. With the way Salesforce has set up the trust layer and agent definition and setup, they have created a new line (added to the left of the model) that brings significant control over the extent of the agent's autonomy, while retaining its capability to execute complex processes.

We're not claiming that Agentforce is anywhere near achieving human-level agency. However, it represents a significant step forward, unlocking immense possibilities for productivity and innovation.

How Agentforce Works

Building Intelligent Agents: The Trust Layer, Integration, and Seamless Execution

Agentforce Agent Breakdown

Agents break down into five key areas as below, to enable them to operate correctly. You'll see that there is a lot of definition of the agent's role and guardrails with data to ensure the right outcome of its actions.

Role

The role we give an agent defines its the specific purpose, narrowing down the tasks it must perform. By setting clear goals, it ensures the agent's actions align with the desired business outcomes.

Trusted Data

We give the agent the data it needs to carry out its role(s). This data may include customer information, business rules, policies, and knowledge bases.

Actions

The actions the agent can carry out on behalf of the user it interacts with. This can reach far beyond the Salesforce ecosystem to other 3rd party systems.

Guardrails

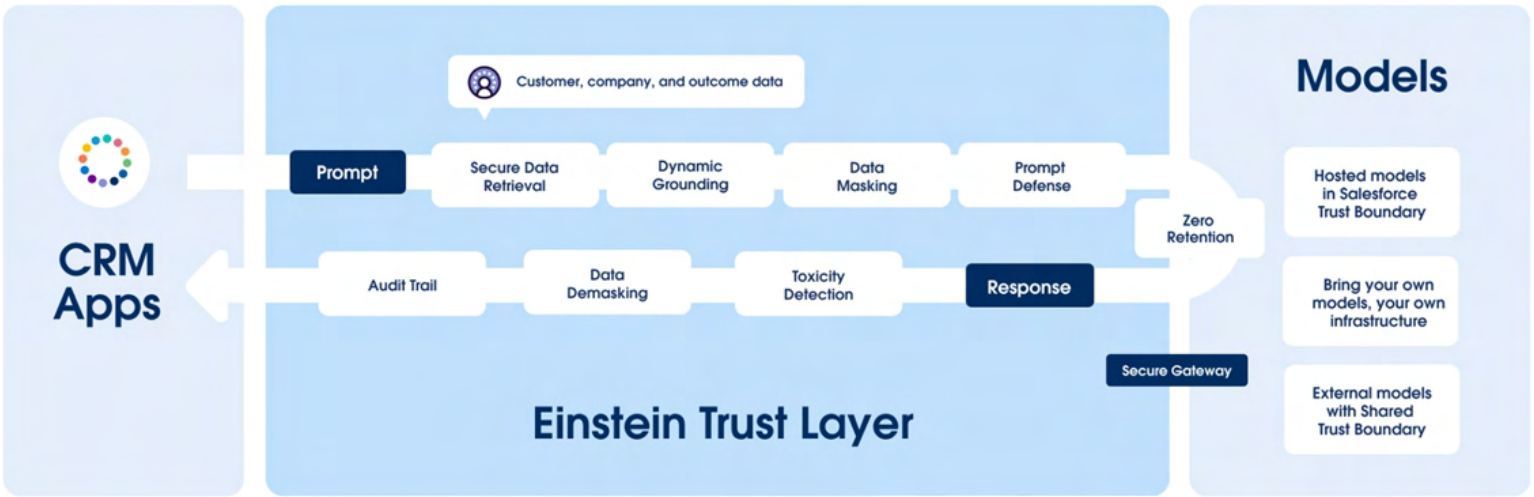
In the form of instructions, we tell the agent what it can and cannot do, how to speak and more. On top of this the Einstein Trust Layer applies systemic guardrails at all times.

Channels

The agent can be deployed across multiple communication channels, such as websites, mobile apps, WhatsApp, Slack, or voice interfaces.

Einstein Trust Layer

Trust is a foundational element of AGENTFORCE. The Einstein Trust Layer is Salesforce's answer for creating that bedrock of trust, as well as then designing effective agents on top of the trust layer.



Here's a breakdown of the Einstein Trust Layer and its key components that make it an excellent foundation for AI-driven agents, as OSF sees it:

- **Secure Data Retrieval**
  Salesforce continues to respect all of its sharing and permissions structure. So, even through AI, agents are just like users, and their data access is set the same way, allowing only the data you want them to have access to.

- **Data Masking**
  Before the prompt goes to an AI model (commonly called an LLM), personal/PII data is masked before it is sent. Models never see PII data, so personal data never leaves the Salesforce ecosystem, no matter what model you connect.

- **Zero Retention**
  Once the model has processed the prompt and provided a response, no data is retained in the model due to the architecture in place. As a result, you can be confident that models are not learning from your data or retaining anything that belongs to you.

- **Toxicity Detection**
  Once the response comes from the model, the trust layer evaluates it for toxicity and inappropriate content. This is important, as no matter what model you might connect, you need a safety net to catch and stop responses that contain any toxicity.

- **Audit Trail**
  Every interaction with AI (request and response) is recorded in the Audit Trail. It can be examined and tracked at any time, giving you the comfort that there is no 'black box' where AI is operating, with full visibility into how it is operating.

How Are Agents Built?

Now we we've covered the trust layer, let's get right into the system with screenshots of exactly how these agents are built. Building agents requires a particular set of skills to get them up and running.

Topics

Defines what the agent will identify and respond to and defines how it will respond to the request

- **Classification**
- **Scope**
- **Instructions**

Actions

Defines the actions the agent can take, once it has identified what the user wants

- **Flows**
- **Prompt Templates**
- **Apex**

Knowledge

(Optional) gives the agent a grounding in your business knowledge from Salesforce Knowledge

- **Einstein Data**
- **Libraries**

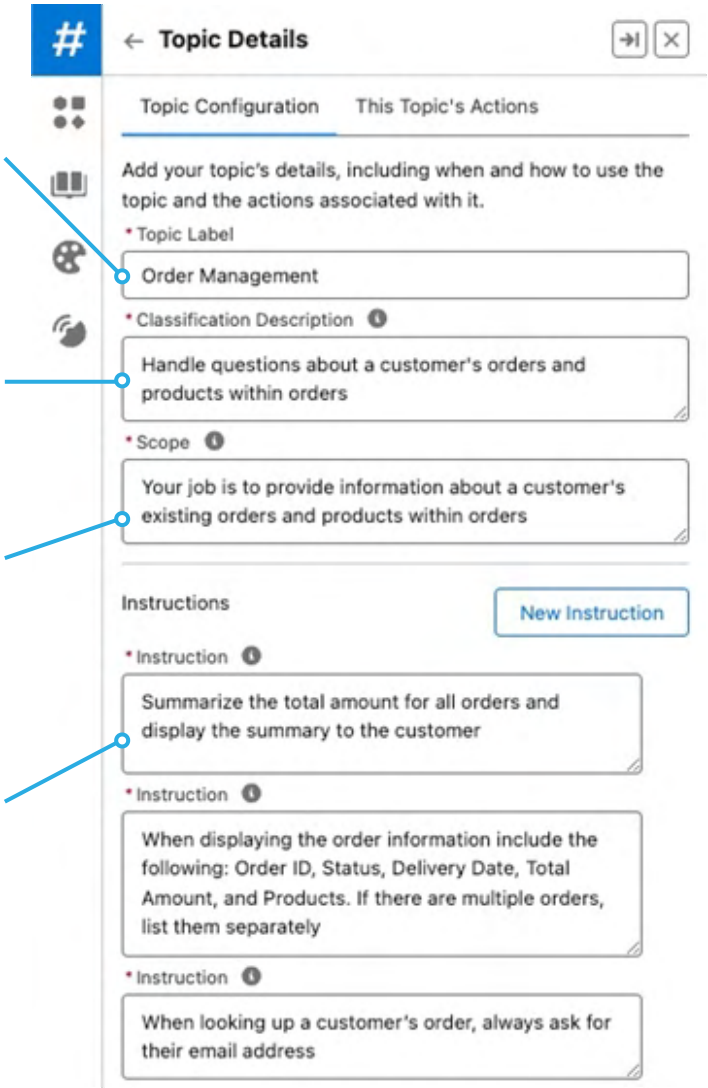
Topics – The What

Topics are the driving force for agents figuring out what they need to do and how they need to respond. This is where the key knowledge of AI comes in to give it the right scope and instructions. In effective we are designing prompts for the agent's behavior. An agent can have multiple topics.

Let's look at a real-life topic from a demo agent.

We have set four types of text for the AI, all of which effectively prompt the agent on how to behave:

1. **Topic Label:** A short label for this topic, in simple terms.
2. **Classification Description:** The agent uses this description to determine if the user's request matches this topic. This is a key field to get right.
3. **Scope:** Defines what the agent can do within this topic.
4. **Instructions:** You can have multiple instructions. Make sure you separate them rather than using one huge block of text. These instructions help add guardrails around formatting, input variables, and other parameters.

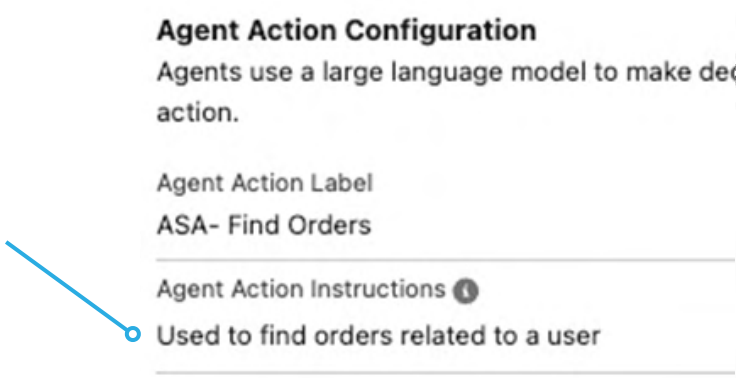


Actions – The How

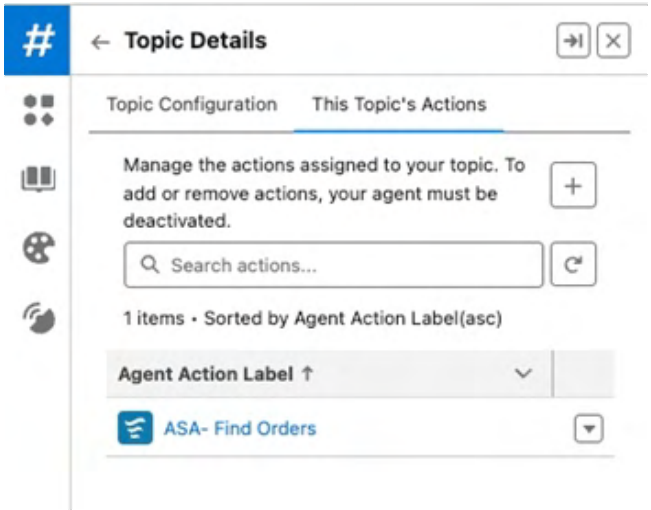
Actions are what the agent takes once it's confident it knows what it should do from the topic. Let's look at the same 'Order Management' topic. In this example, we only have one action (it's a very simple agent) called '**Find Orders**'.

You might be wondering, how do we link the actions to a particular instruction in the topic? The short answer is: we don't. We leave that to the Atlas reasoning engine. It compares all the instructions and actions, and decides which one to use. Very cool, right?

Let's see how it matches, though, which happens using the '**agent action instructions**'. The agent reads this instruction and compares it to the user's input and its own instructions, selecting the right topic and then the right action for the job.



From this, we conclude that building great agents isn't just about being good at Salesforce configuration. **We also need to become skilled at writing topics and actions (which are themselves prompts)** so the agent can pick the right route, even when there are many possible actions and instructions.

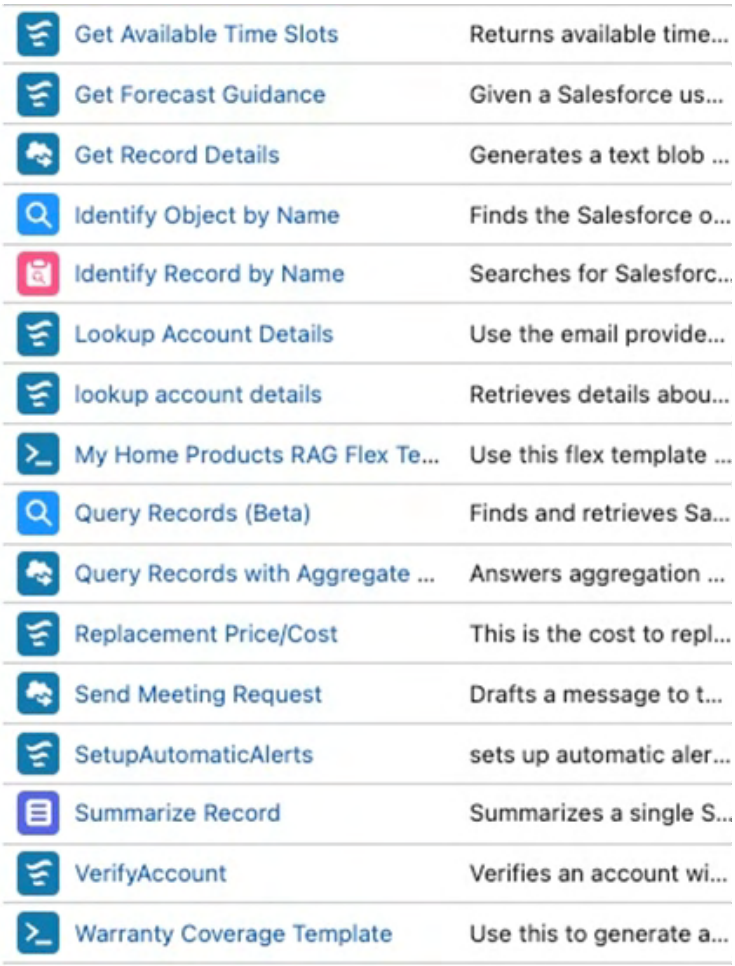


Actions – Deeper Dive

So, we understand the setup with topics and actions, but we haven't explored the detail of what an action can be. These split into Prompt Templates, Flows, and Apex, which are possible routes to perform an action.

Then, actions split into two types:

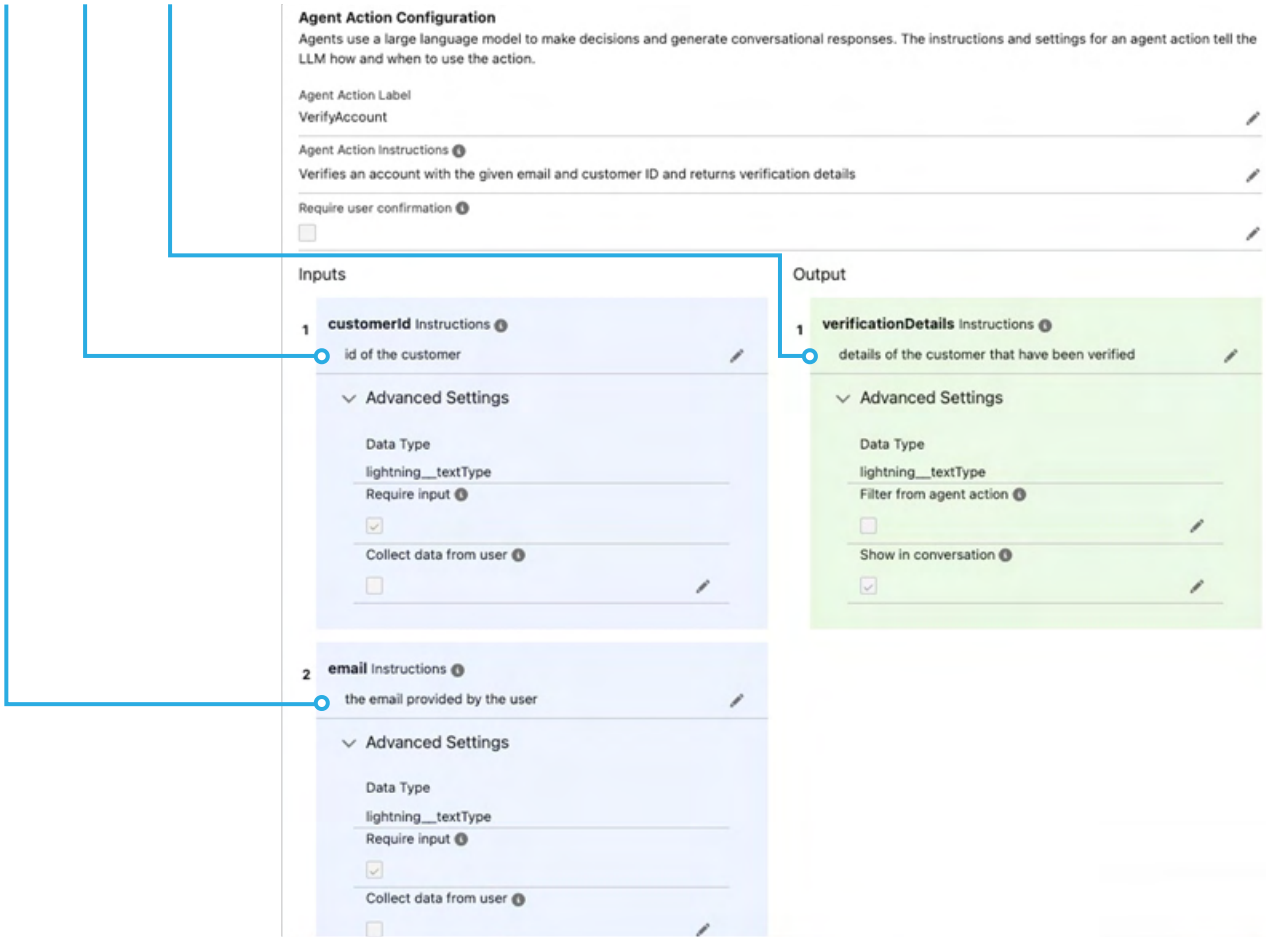
Standard: These are supplied by Salesforce. As agents develop, expect Salesforce to provide more and more standard actions for use. For now, examples include 'Answer with Knowledge', 'Identify Record by Name', and 'Draft or Revise Email'.



Custom: Internal Salesforce teams and partners like OSF can help with the creation of new agents, their topics, and actions. OSF believes this is where there will be an explosion of actions available. In fact, partners like OSF are already building their own

Now that we know actions can be standard or custom, and can be Apex, Flow, or Prompt Templates, let's look at an example of each. But before we do that, let's briefly explore how input and output work for actions.

Every action needs input and output. Take the **Verify Account** action, for example. It requires two inputs—**customerId** and **email**. The output is the **verificationDetails**. The agent provides the inputs and then receives the output.



Make sure you note the 'Instructions' on each of the inputs and outputs. These are yet more natural **language instructions to the agent** to make sure it understands what it is giving and getting back, and feeds the Atlas reasoning engine.

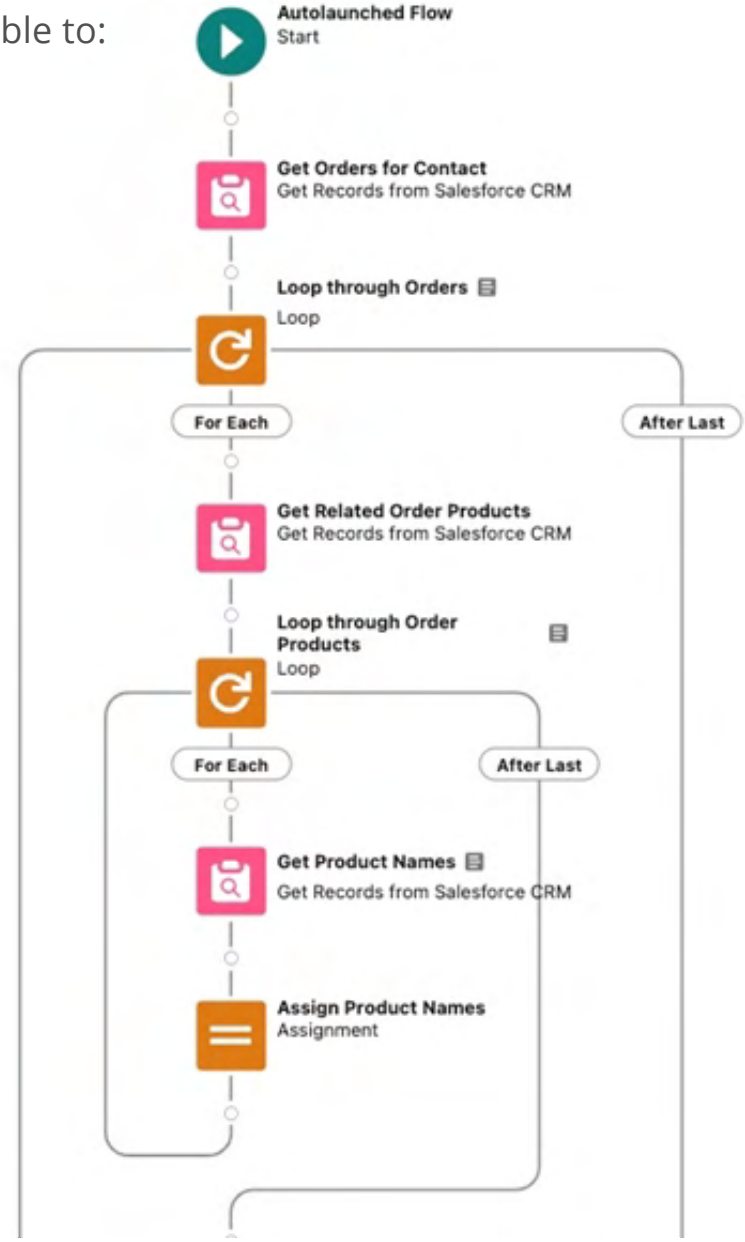
What we love at OSF is how the whole process of building agents also leverages AI and reasoning to ensure that we are not making strong links between one instruction and one action. This would significantly lower the flexibility of agents and the ability to use actions as needed, to help the customer or stakeholder.

Actions – Flow

Salesforce's Flow tooling is one of the most versatile and useful parts of the whole platform. It is used everywhere. It helps build checkouts, order management capability, general business automation, and now, agent actions.
There's a simple flow for the **'FindOrders'** action from our agent. This flow retrieves orders for the customer we're speaking to, gets the related products and their names, and then provides that information back to the agent so it can be shared with the customer.

Flows can get very flexible by being able to:

- Use integrations like Mulesoft and APIs to call in data or execute actions, allowing agents to access third-party systems to do their work.
- Call Apex code if needed (Apex is hihgly flexible, and more developer-focused)
- A lot of record work, where business users can create these flows – such as finding, sorting, updating records.



Actions – Prompt Template

Prompt Templates are incredibly powerful way to drive an agent's actions. It utilizes even more of the power of Generative AI, again within constrained guardrails.

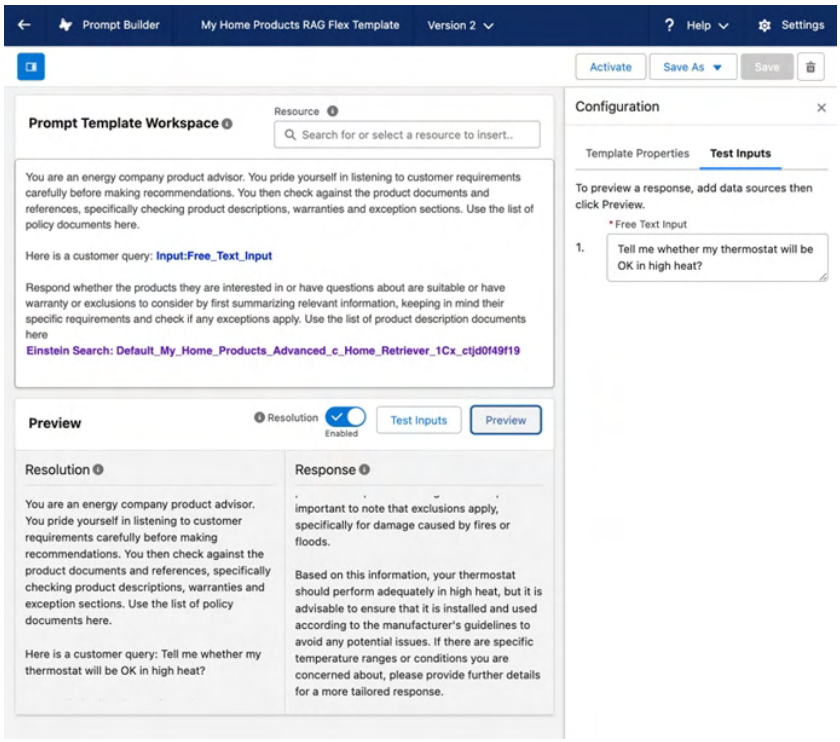
Check out the example to the right. There's a lot going on here, but it's not too technical!

1. **Prompt:** The prompt itself, written in natural language, sets the stage for what the action needs to do.

2. **Data:** You can see two pieces of data here: the first is what the user asked (Input::Free_text_Input), and the second is unstructured data provided by Data Cloud—in this case, a set of product documents (Einstein Search:...), such as manuals.

3. **Preview:** Here, you can see the test input and the response from the AI, allowing us to test the action. It performed great!

Now, the agent can query product documentation from an external system via Data Cloud and summarize a response to a customer question—all in one prompt template!



Actions – Apex

Apex has long been a useful part of the Salesforce ecosystem: it is Salesforce's internal coding language and tooling, used for more heavyweight processing and advanced needs on the platform.

The more we rely on Apex code, the less maintainable the system becomes for administrators and users, as it moves away from the business/declarative tooling. As much as possible, we try to use the right tool for the job—leveraging business tooling for most configurations and extending with Apex only for areas that require more heavyweight solutions.

Still, Apex can be invoked by an agent's action, providing huge flexibility because anything Apex can do, an agent can do too. Many existing orgs will have numerous Apex-driven processes, so an agent's ability to use them is essential.

```
38              }
39
40              public void save() {
41
42                  if (ctrl.getRecord().getsObjectType() == Case.sObjectType) {
43                      try {
44                          //Fetching the assignment rules on case
45                          AssignmentRule AR = new AssignmentRule();
46                          AR = [select id from AssignmentRule where SobjectType = 'Case' and Active = true limit 1];
47                          System.Debug('CASE AR: ' + AR);
48
49                          Database.DMLOptions dmlOpts = new Database.DMLOptions();
50                          dmlOpts.assignmentRuleHeader.assignmentRuleId= AR.id;
51
52                          Case newCase = new Case();
53                          newCase = (Case)ctrl.getRecord();
54                          //Setting the DMLOption on Case instance
55                          //newCase.RecordTypeId = selectedRecordId;
56                          newCase.setOptions(dmlOpts);
57                          upsert newCase;
58                          ApexPages.Message myMsg = new ApexPages.Message(ApexPages.Severity.CONFIRM , 'Tha
59                          ApexPages.addMessage(myMsg);
60                      } catch (Exception e) {
61                          ApexPages.Message myMsg = new ApexPages.Message(ApexPages.Severity.ERROR , e.getM
62                          ApexPages.addMessage(myMsg);
63                      }
64                  } else if (ctrl.getRecord().getsObjectType() == Lead.sObjectType) {
65                      try {
66                          //Fetching the assignment rules on case
67                          AssignmentRule AR = new AssignmentRule();
68                          AR = [select id from AssignmentRule where SobjectType = 'Lead' and Active = true limit 1];
69                          System.Debug('LEAD AR: ' + AR);
70
71                          Database.DMLOptions dmlOpts = new Database.DMLOptions();
72                          dmlOpts.assignmentRuleHeader.assignmentRuleId= AR.id;
73
74                          Lead newLead = new Lead();
75                          newLead = (Lead)ctrl.getRecord();
76                          //Setting the DMLOption on Case instance
77                          newLead.setOptions(dmlOpts);
78                          upsert newLead;
79                          ApexPages.Message myMsg = new ApexPages.Message(ApexPages.Severity.CONFIRM , 'Tha
80                          ApexPages.addMessage(myMsg);
81                      } catch (Exception e) {
82                          System.Debug('Error: ' + e.getMessage());
83                          ApexPages.Message myMsg = new ApexPages.Message(ApexPages.Severity.ERROR , e.getM
84                          ApexPages.addMessage(myMsg);
85                      }
86
87                  } else {
```

Agentforce Explained: How It All Comes Together

Let's take a step back and sum up how Agentforce operates. By combining clear guidance on what the agent can do, secure execution protocols, and powerful AI-driven capabilities, Agentforce delivers a highly adaptable solution for business automation. These components work together seamlessly, providing a reliable and efficient way to leverage the power of AI for enhanced productivity and performance.

Topics – The What

Topics govern how the agent operates, its guardrails, what it can and can't answer, and how to do it.

- **The better the instructions, the better the agent performance.**
- **The more you instruct the agent, the more it will align with your brand.**
- **Clear topics make an agent more effective.**

Actions – The What

Actions govern what the agent can do, once it has figured out what it is meant to do from the topics.

- **Actions are specific and well managed in Flow, Prompts or Apex, giving further guardrails for task execution.**
- **The routes for actions are very flexible and can incorporate 3rd party systems to Salesforce.**
- **If an agent doesn't understand the topic, no action is taken.**

Einstein Trust Layer

The trust layer acts as a strong safety net and set of guardrails that are present for every agent and AI execution.

- **Data masking for no PII leaves Salesforce.**
- **Toxicity detection for appropriate responses.**
- **Zero retention of request or response—your data remains yours.**
- **Governs all models.**

AI & Prompting

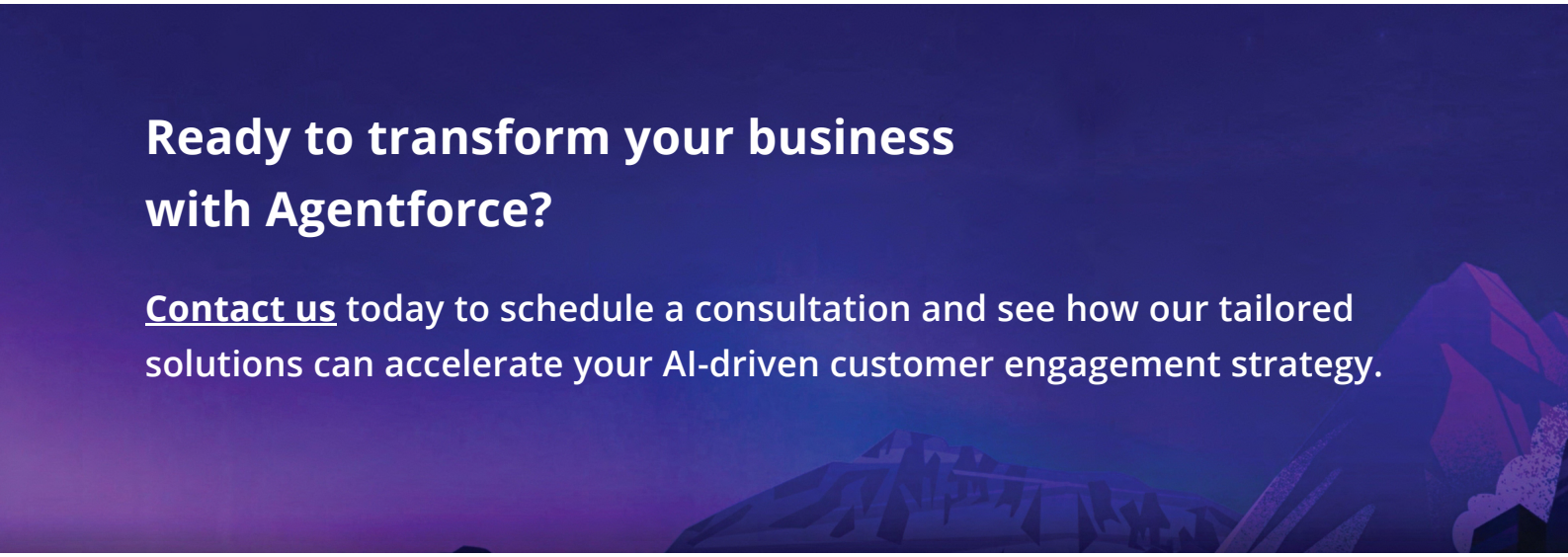
At the heart of Agentforce is the ability to prompt AI effectively using natural language.

- **People can learn to prompt well with some practice.**
- **Prompt Builder and Agentforce enables easy testing of agents.**
- **As actions govern what the agent does, you can feel safe in making the actions highly specific.**

Conclusion

Agentforce offers significant benefits for businesses by streamlining operations, enhancing customer experiences, and driving productivity. The flexibility of Agentforce, coupled with its integration capabilities, allows businesses to customize the agents to their specific needs, ensuring that they can handle diverse workflows across different departments like sales, marketing, and customer service. With Salesforce's trusted platform and the built-in safety features of the Einstein Trust Layer, businesses can confidently deploy autonomous agents without worrying about data security, compliance, or ethical concerns. This combination of efficiency, flexibility, and security enables businesses to scale operations, enhance customer satisfaction, and ultimately drive profitability, all while maintaining full control over how their agents operate and interact.

To truly unlock the potential of Agentforce, it's important to understand its practical applications and the proven frameworks that OSF Digital offers to help businesses activate it successfully. Continue exploring how Agentforce can revolutionize your business – read Part 2 of this white paper series, to learn more about Agentforce's core capabilities, real-world use cases, and competitive landscape.



OSF DIGITAL

OSF Digital is a global leader in digital transformation, specializing in Salesforce solutions that drive operational efficiency and business growth. With expertise in AI and composable architectures, OSF Digital empowers businesses to create seamless, future-ready customer experiences. Leveraging data-driven insights, OSF helps clients enhance performance, optimize processes, and scale for success. From innovative commerce solutions to managed services, OSF Digital is committed to helping companies maximize their digital investments and achieve measurable business outcomes.

Thank You

Connect With Us

osf.digital