Oracle Private Agent Factory

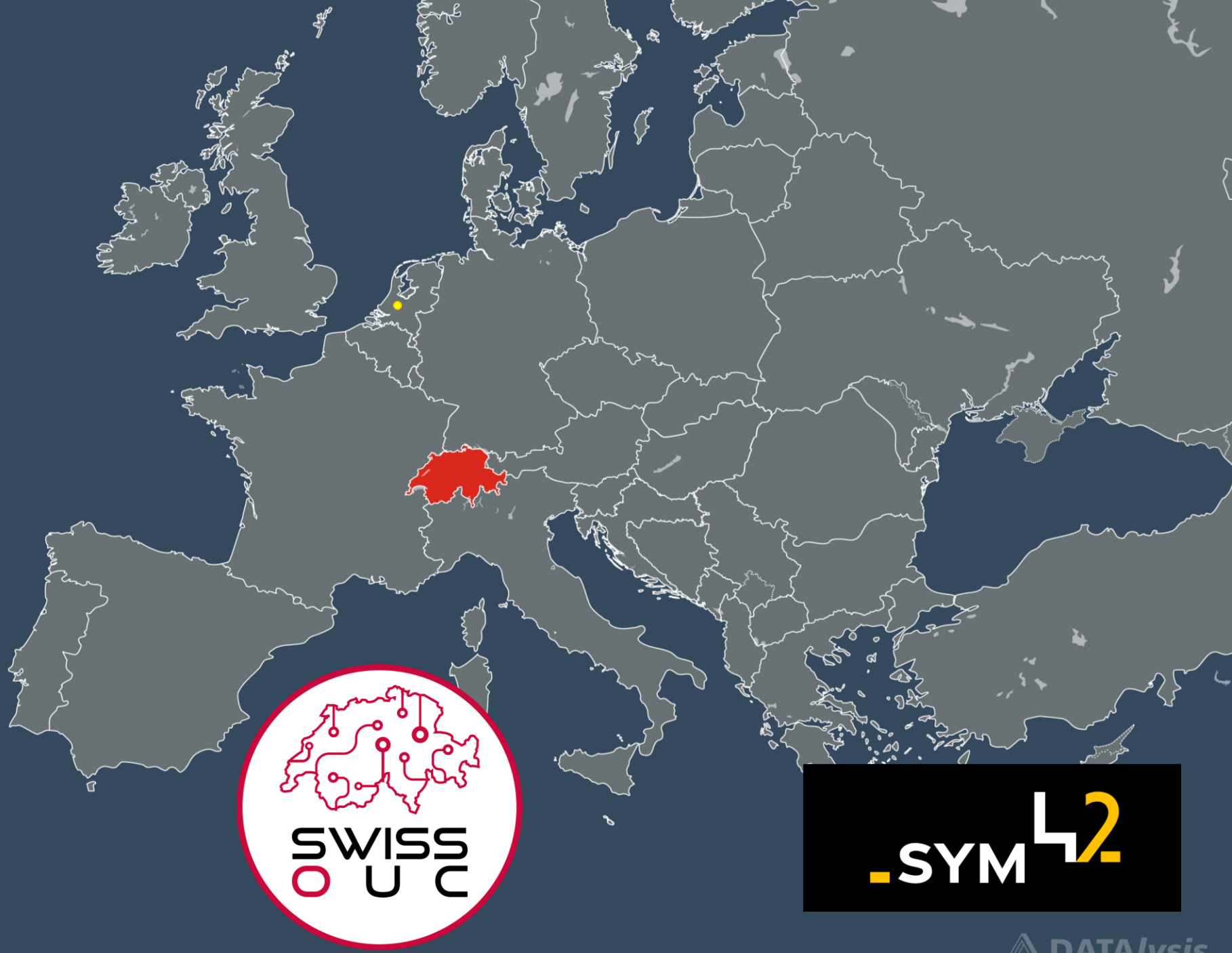
Your no-code, privacy-friendly, secure agents platform

November 13, 2026

@G_Ceresa W: www.datalysis.ch E: info@datalysis.ch @gianniceresa.bsky.social

Gianni Ceresa

Working with ***data***,
Business Analytics
and EPM tools
for more than
15 years



Oracle ACE Director

SWISS OUC

SYM42

@G_Ceresa @gianniceresa.bsky.social

DATAlysis

Oracle ACE

The Oracle ACE Program

1200+ technical experts helping peers globally



- The Oracle ACE Program recognizes and rewards community members for their technical and community contributions to the Oracle community
- 4 membership levels: Director, Pro, Associate, and Apprentice
- Nominate yourself or a colleague at ace.oracle.com/nominate
- Learn more at ace.oracle.com

aceprogram_ww@oracle.com
@oracleace.bsky.social
@oracleace
Oracle ACE Community

DISCLAIMER



(Still) Brand new product, first released in January, every update changes ... a number of things.

New release 2 weeks ago at AI World: installed, no time to look at it...

Oracle Private Agent Factory

Your no-code, privacy-friendly, secure agents platform

@G_Ceresa W: www.datalysis.ch E: info@datalysis.ch @gianniceresa.bsky.social

AI Agents everywhere...

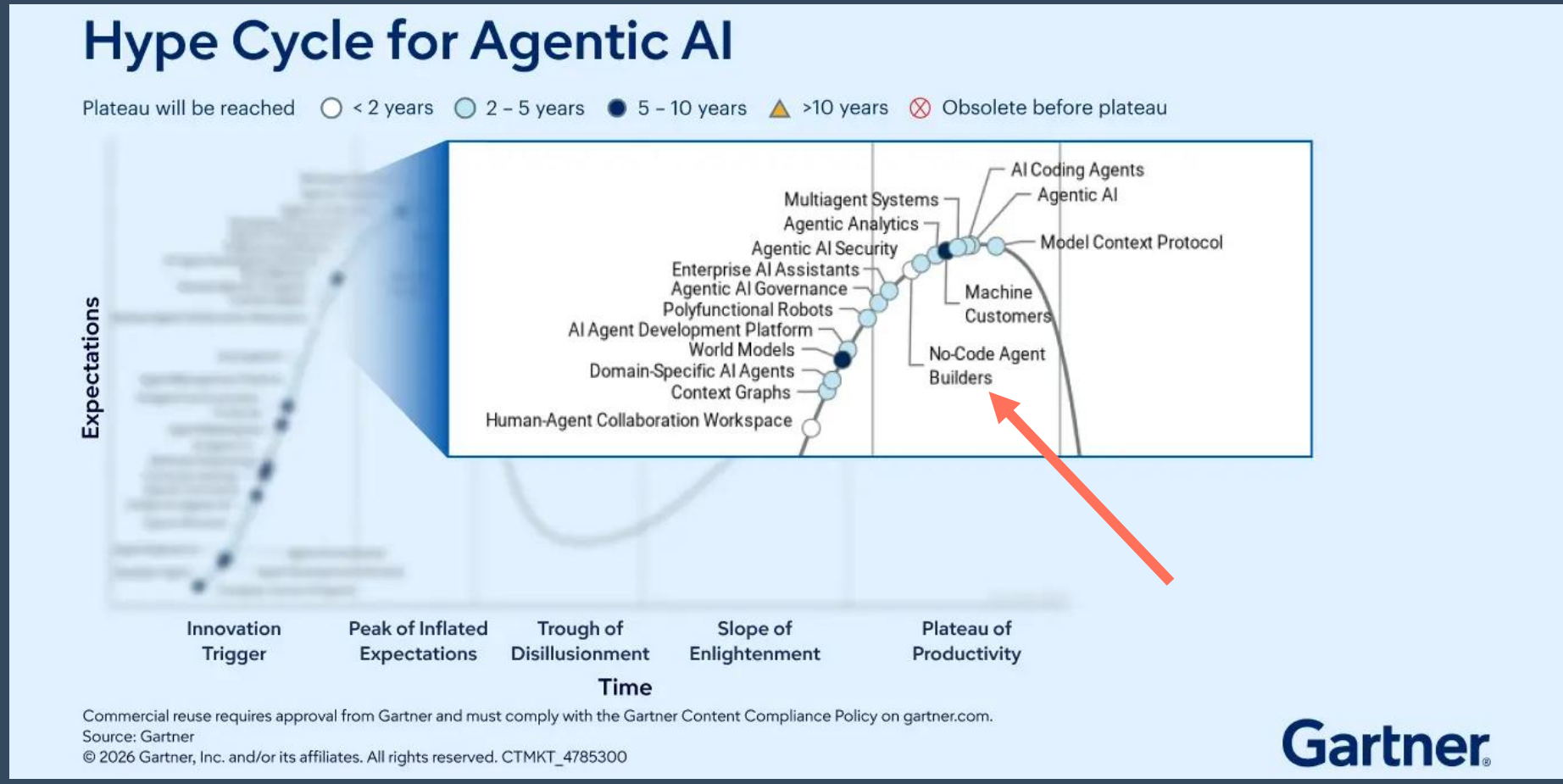
AI Agents everywhere...



AI Agents everywhere...

In this day and age, everybody must have a few AI agents !

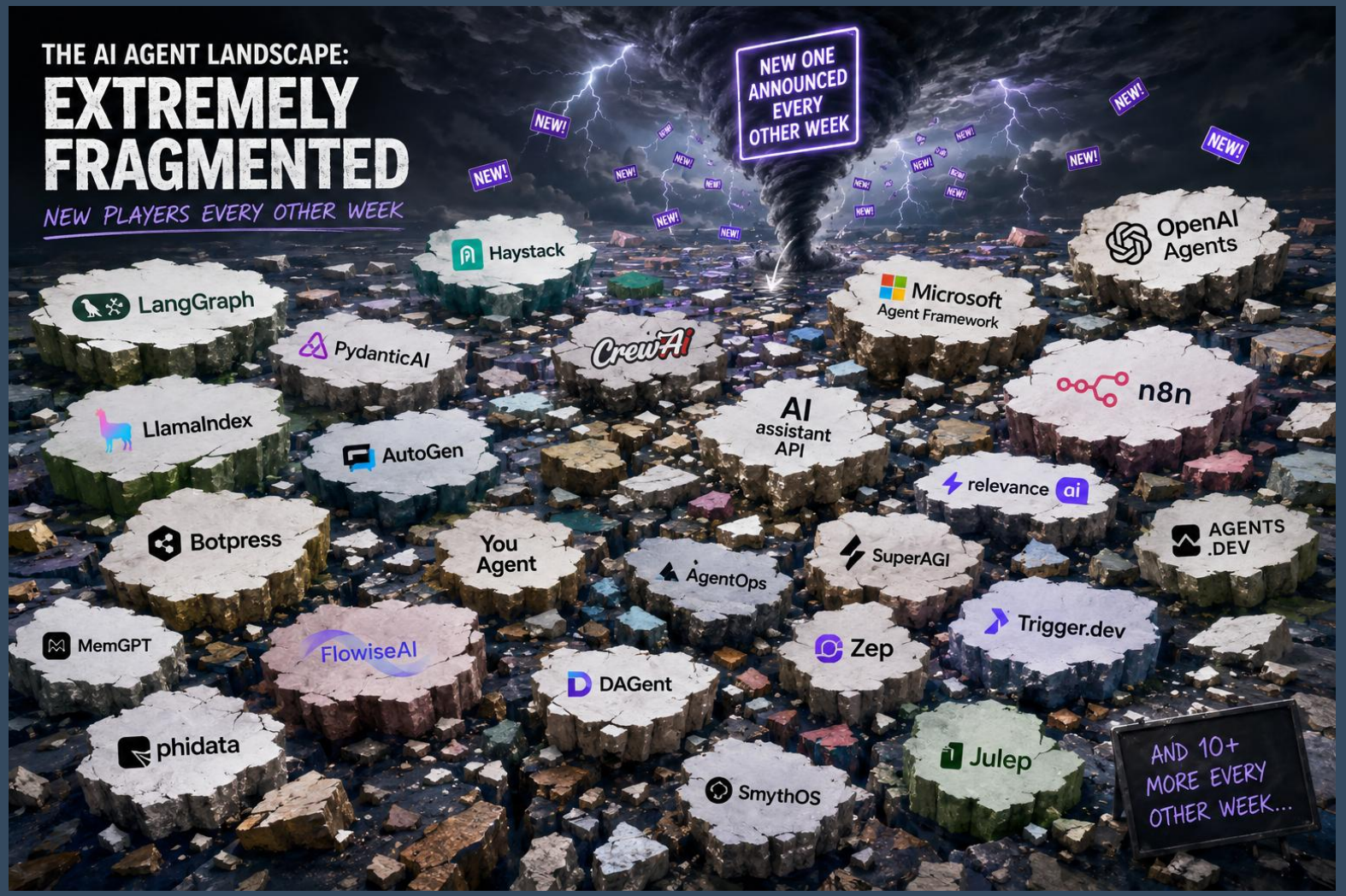
We are near the top of the “Peak of Inflated Expectations” in the Gartner Hype Cycle for Agentic AI.



AI Agents everywhere...

First, packages and libraries to easily code agentic behaving solutions.

Then, no-code AI agents platforms, based on pre-defined building blocks.



AI Agents everywhere...

Oracle couldn't miss the chance to be "yet another" fragment...

A number of products exists already:

- OCI Generative AI
- OCI Generative AI Agents
- Oracle AI Agent Studio (for Fusion Applications)

A number of products have agents embedded, like:

- AI Data Platform
- OAC
- Fusion

And now, another one:

- Oracle AI Database Private Agent Factory

Oracle AI Database Private Agent Factory, or just Private Agent Factory (PAF)

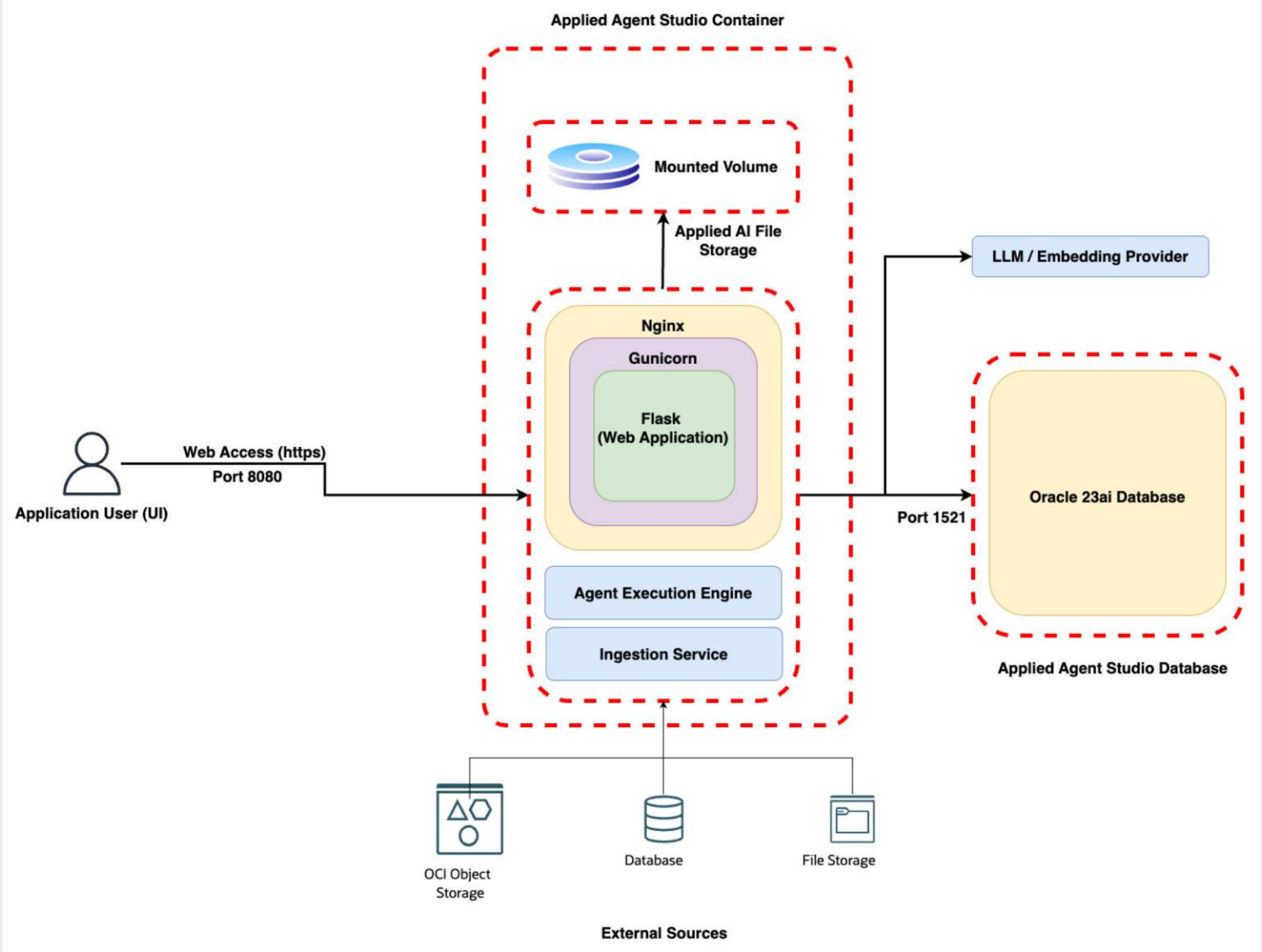
What is it? How to run it? How much does it cost?

High level architecture

It is just a container...

- Oracle Linux 8, 9, RHEL, Ubuntu or macOS
- Podman (rootless container)

Requires an Oracle AI Database 26ai (with extended data types enabled)



Installation

Installation is easy:

Using the Marketplace stack deployment in OCI.

or

Extract an archive and run a script.

Configuration via web interface:

- Create the admin account
- Database connection
- First LLM and embedding models

```
Deleted: 9044ae84fa7489a832148745382ba9a85d7ce57f55da875f634b6e611829a619

Cleaning up proxy environment variables...

+-----------------+---------+--------+-------------------------------+--------------------+-------------------------------------+
| Image           | Status  | Size   | Created                       | Image ID           | Tags                                |
+-----------------+---------+--------+-------------------------------+--------------------+-------------------------------------+
| applied-ai-label | SUCCESS | 6.2 GB | 2026-05-13T14:29:37.247873481Z | 5a9a6e4c9e2c67c5bf53 | localhost/applied-ai-label:25.3.0.0.9 |
+-----------------+---------+--------+-------------------------------+--------------------+-------------------------------------+

[SUCCESS] Step 'Build container images' completed.

[INFO] Starting step: Launch application containers
[INFO] Bringing containers up with 'make up'.
[WARNING] You will be prompted to choose an installation mode. This script assumes 'prod' mode (option 1).
bash deploy.sh; \

   ORACLE
AGENT FACTORY

  ✔ Image 'localhost/applied-ai-label:25.3.0.0.9' found.
  ✔ Image 'localhost/applied-ai-label:25.3.0.0.9' found.
Deploying Oracle AI Database Private Agent Factory in Production mode...
Starting production containers...
Confirming container 'oracle-applied-ai-label' is in a running state...
  - Waiting for container to be ready...
✔ Container is confirmed to be running.
  - Configuring application container 'oracle-applied-ai-label'...
  - Configuring application container: oracle-applied-ai-label...
  ✔ crond service started in oracle-applied-ai-label.
  ✔ Application container oracle-applied-ai-label configured successfully.
Initiating startup for oracle-applied-ai-label container...
  ✔ oracle-applied-ai-label container is UP and RUNNING.
  - Please wait while the application is being set up within oracle-applied-ai-label container...
✔ Application setup successfully within the container 'oracle-applied-ai-label'.
+-----------------------------------------------+------------+
| Stage                                         | Status     |
+-----------------------------------------------+------------+
| Stopping Oracle Private AI Agent agent_factory | Successful |
+-----------------------------------------------+------------+
| Storing App Secret Key                        | Successful |
+-----------------------------------------------+------------+
| Database Migration                            | Successful |
+-----------------------------------------------+------------+
| Starting Oracle Private AI Agent agent_factory | Successful |
+-----------------------------------------------+------------+
The Oracle AI Database Private Agent Factory webapp can be accessed at:
https://private-agent-factory.sub05131423250.datalysisvcn2.oraclevcn.com:8080/agentFactory/
✔ Production deployment complete.
Logs can be found at: ./deploy.log
[SUCCESS] Step 'Launch application containers' completed.

[SUCCESS] Installation script finished!
[INFO] You can now access the application.
[opc@private-agent-factory private_agent_factory]$
```

Management

It comes with a set of scripts for lifecycle management.

It is a podman-compose file, but not made for direct interactions.

Everything is done with one command: `make <target>`

Category	Target	Description
Lifecycle	`up`	Starts the application by running the main deployment script (`deploy.sh`).
Lifecycle	`deploy`	Deploys the application using the active compose file (`podman-compose.yaml`).
Lifecycle	`down`	Stops and removes all application containers.
Lifecycle	`start`	Starts previously stopped containers without recreating them.
Lifecycle	`stop`	Stops running containers without removing them.
Lifecycle	`restart`	Stops and then starts the containers.
Lifecycle	`clean`	Stops containers and removes the main project image.
Lifecycle	`uninstall`	Runs a script to completely uninstall the application.
Build	`build`	Builds all required container images via a script.
Build	`buildaai`	Builds only the AAI service image.
Build	`buildollama`	Builds only the Ollama service image.
Build	`build23ai`	Builds only the Oracle AI Database 26ai Free image (`db23aifree`).
Diagnostics	`logs`	Tails the logs from all running services.
Diagnostics	`logsaai`	Tails the logs from the AAI service container only.
Diagnostics	`diagnose`	Runs a diagnostic script to check the deployment health.

For your sandbox dev environment →

podman-compose

For ***Production*** mode

```
[opc@private-agent-factory private_agent_factory]$ more podman-compose.yaml
version: "3"

services:
  oracle_applied_ai:
    image: localhost/applied-ai-label:${APP_VERSION}
    container_name: oracle-applied-ai-label
    user: "0:0"
    volumes:
      - ${INSTALL_LOC:-.}/applied-ai/dev-shared:/dev-shared
      - ${INSTALL_LOC:-.}/applied-ai/volume:/mount
    command: "/home/aaiuser/install/startup.sh"
    stop_grace_period: 30s
    ports:
      - "8080:8080"
    networks:
      - oracle_network
    # restart: always
    environment:
      - IS_UPGRADE=${IS_UPGRADE:-false}
      - HOST_OS=${HOST_OS}
      - PYTHONPATH=/home/aaiuser/install
      - NO_PROXY=${NO_PROXY}
      - HTTP_PROXY=${HTTP_PROXY}
      - HTTPS_PROXY=${HTTPS_PROXY}
      - no_proxy=${no_proxy}
      - http_proxy=${http_proxy}
      - https_proxy=${https_proxy}
      # Deployment inputs passed from deploy.sh/.env
      - TAKE_SNAPSHOT=${TAKE_SNAPSHOT}
      - USE_WALLET=${USE_WALLET}
      - INCLUDE_KA_DATA=${INCLUDE_KA_DATA}
networks:
  oracle_network:
    driver: bridge
```

For QuickStart mode the compose does define and start an Oracle Free database as well.

You can easily see how to add an Ollama container as well...

Privacy, Security, but with Enterprise in mind

As a podman container, it can run on-premises or your own compute instance on any cloud provider.

No cloud dependencies, fully self-managed and self-contained.

Better when integrate with your enterprise SSO to allow access to more than just the admin user. (Local users available too.)

Can use any local embedding model or LLM (e.g. Ollama or vLLM), or a cloud provider with an OpenAI compatible API, Google Gemini or OCI GenAI.

Fully private, if using only services running on-premises or fully owned by you (including the LLM inference services).

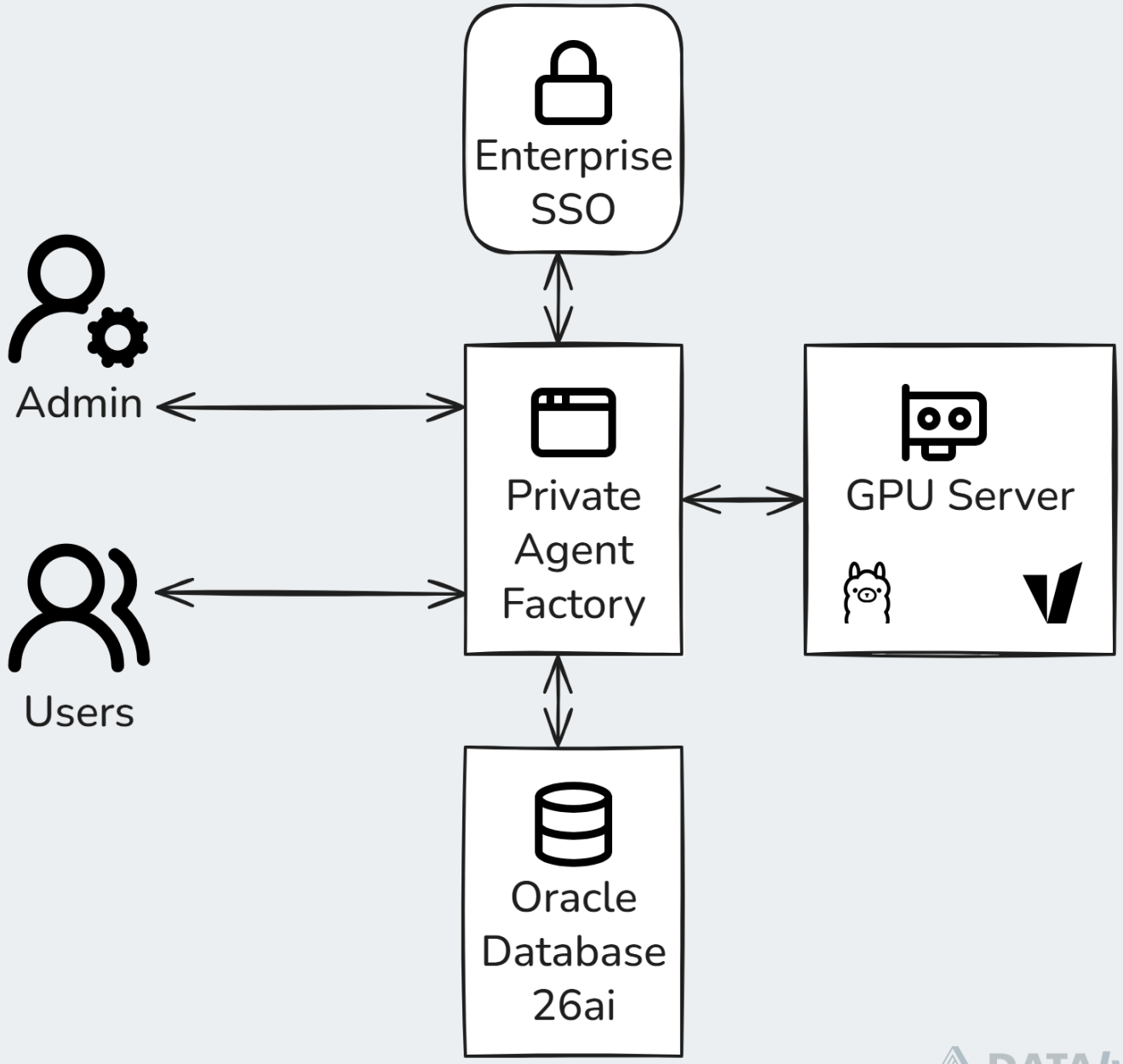
Privacy, Security, but with Enterprise in mind

For full privacy you need:

- LLM inference and embedding on-premises
  - Ollama
  - vLLM

A number of open-source models are very good, but can't always compete with the latest, and greatest, commercial models.

Performance (of LLMs) on-premises can be challenging: usually a lot less hardware available than a cloud service.



Price

Good news on the pricing...

IT IS FREE !

It is Free because you need an Oracle Database 26ai to run it, and therefore you are paying for that already.

One of the many features and tools that Oracle made free once you pay for your database.

Private Agent Factory

What can it do?

No-code workflows for custom agents and pre-built agents

The product, in the current available features, has 2 main usages.

Pre-built agents

- Knowledge: fully automated RAG agent, ingestion, embedding, retrieval.
- Data Analysis: interactive datasets analysis, similar to what Select AI or OAC allows to do.
- Deep Data Research: uses file data sources to prepare a searchable knowledge base for research-style questions.

Build agents/workflows from scratch

- The Agent Builder let you build your own workflows and agents.

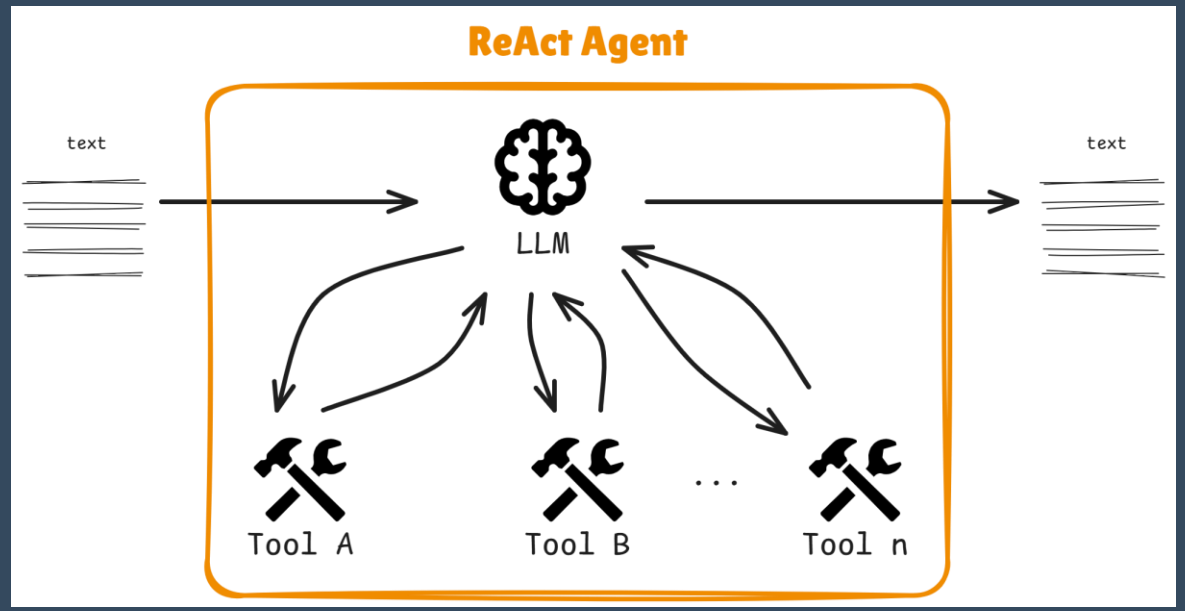
Samples

- Template Gallery: a set of pre-built agents/workflows that you can fully customize if needed, or just use as examples and inspirations to see what you could build.
- Example datasets for the pre-built agents.

Agent vs Workflow: what is the difference?

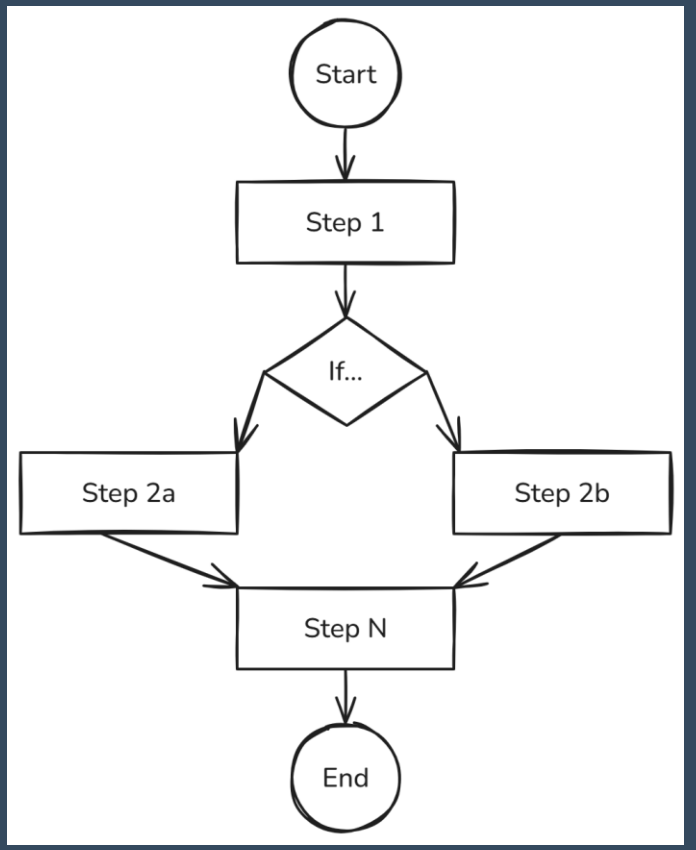
Agent:

- Receive an input
- The LLM thinks
- Can call provided tools
- Keeps iterating till it has an answer
- Returns the answer



Workflow:

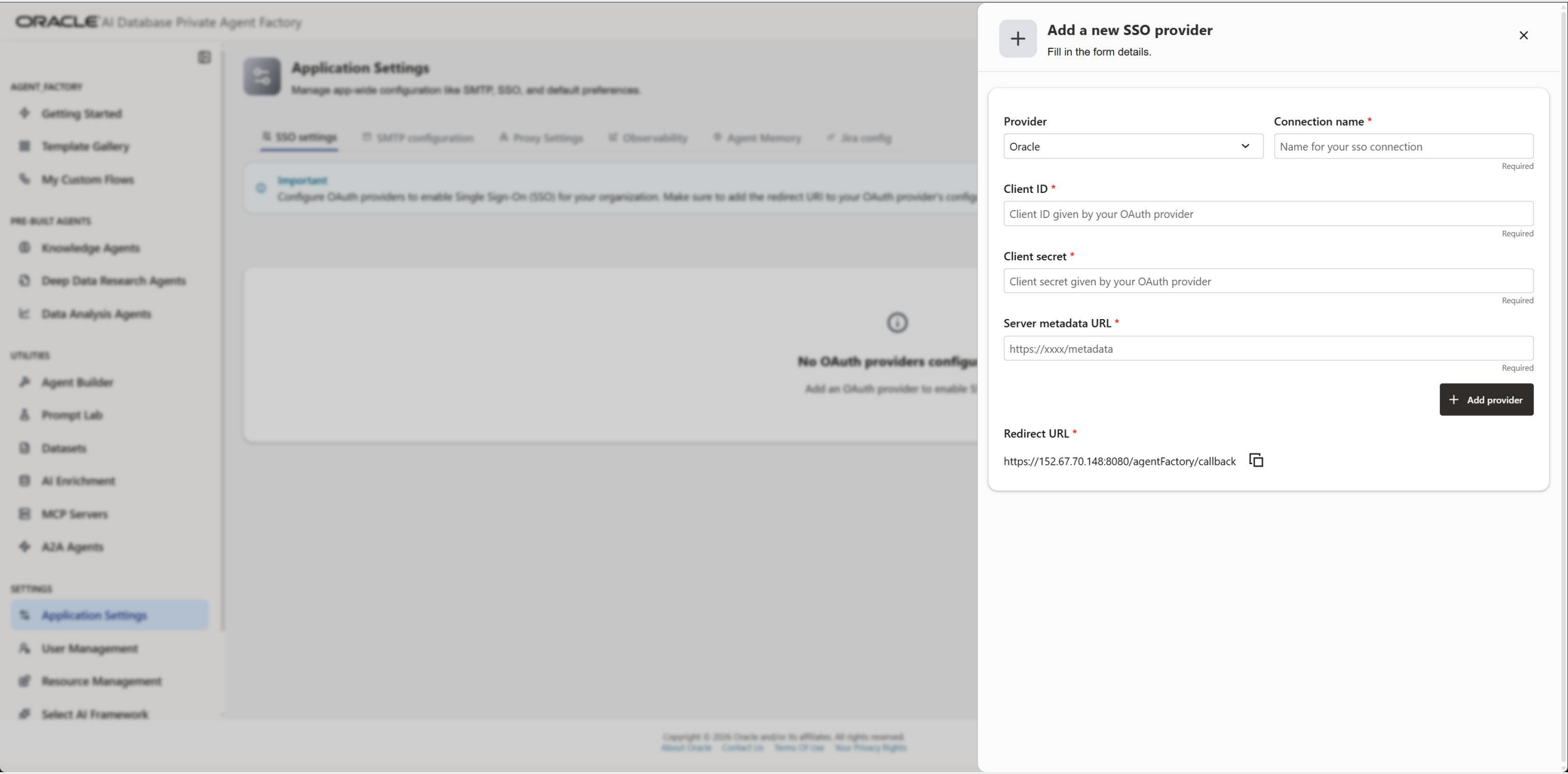
- Execution follows the predefined sequence of steps
- No “thinking” during the process
- A step can be an Agent



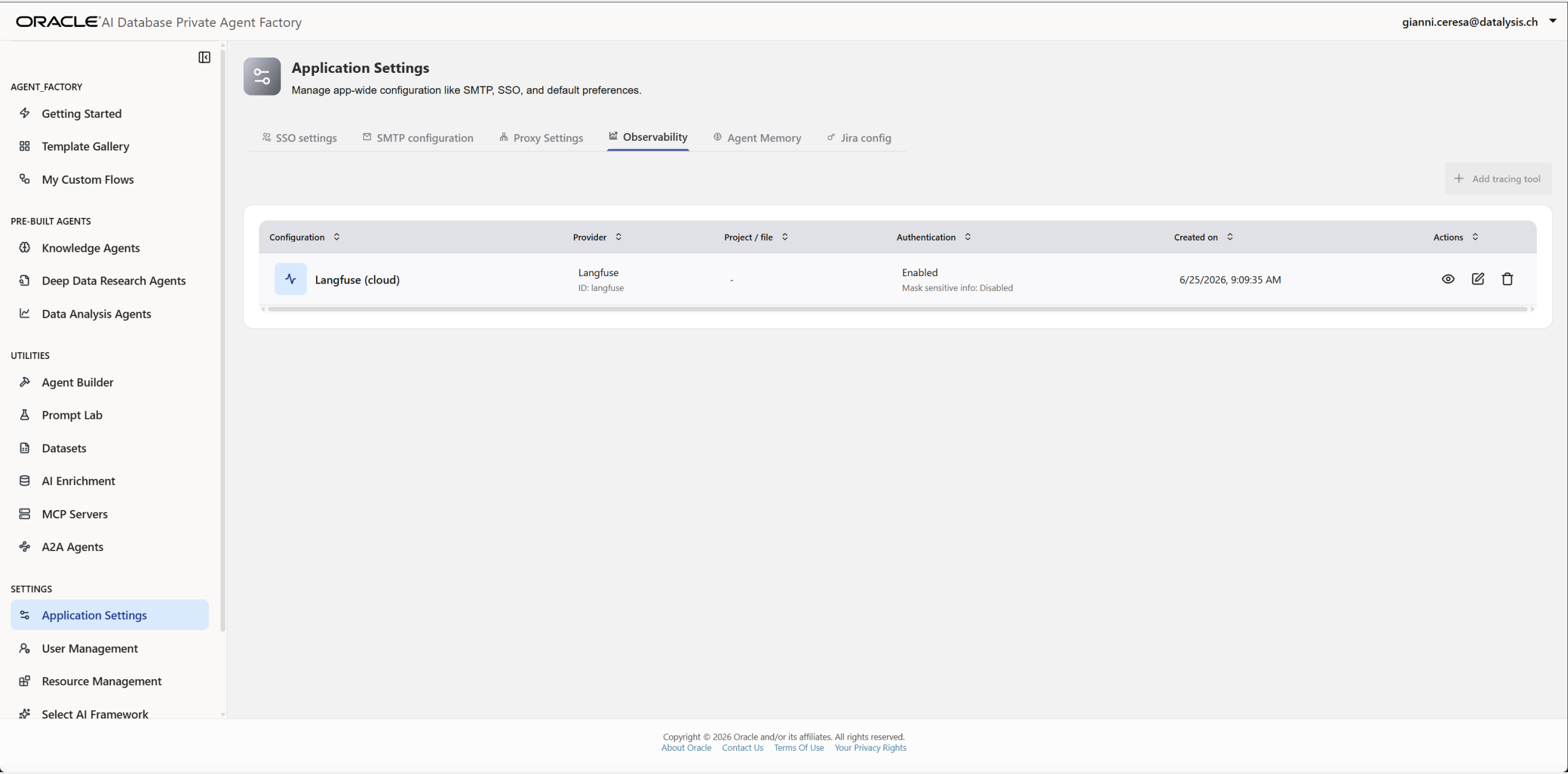
Before the main features, the useful pieces supporting them

Settings & Utilities

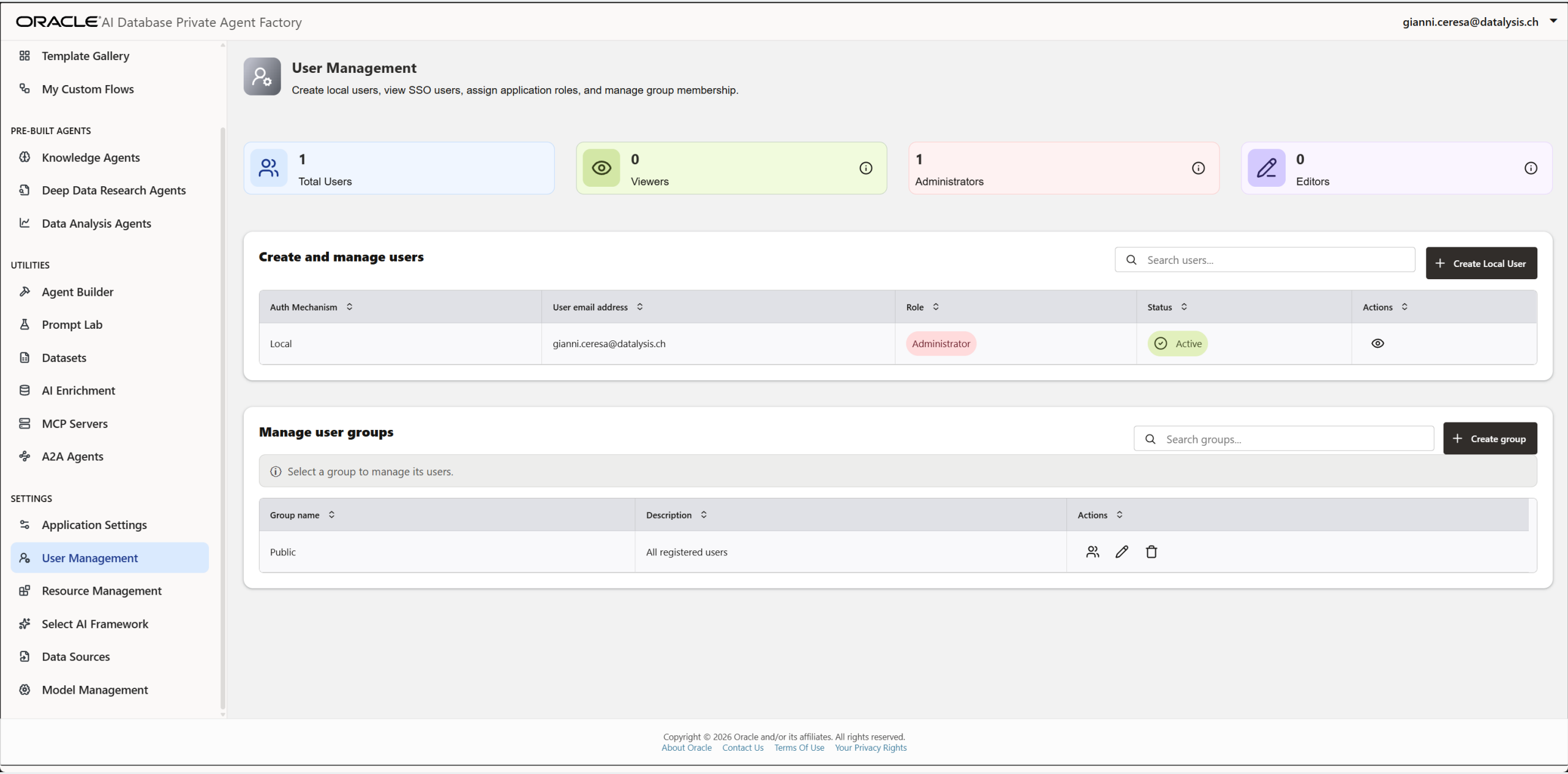
Configure once, forget after...



Configure once, forget after...



Users can now be local, but you should still prefer SSO users



Generative and embedding models configured once

Template Gallery
My Custom Flows

PRE-BUILT AGENTS
Knowledge Agents
Deep Data Research Agents
Data Analysis Agents

UTILITIES
Agent Builder
Prompt Lab
Datasets
AI Enrichment
MCP Servers
A2A Agents

SETTINGS
Application Settings
User Management
Resource Management
Select AI Framework
Data Sources
Model Management

Model Management

Configure and manage language model settings

+ Add configuration

Generative models | Embedding models

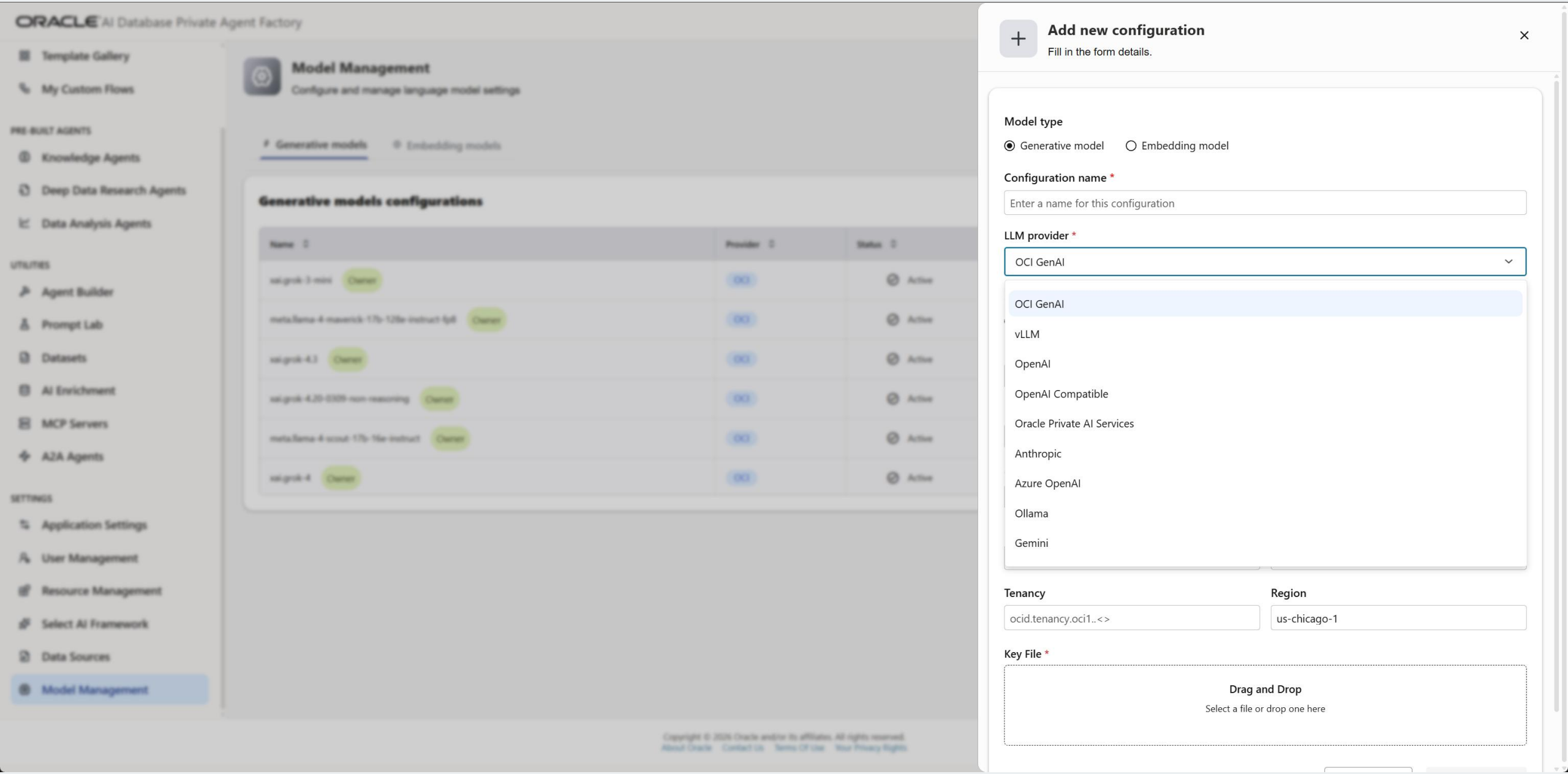
Generative models configurations

Search configuration...

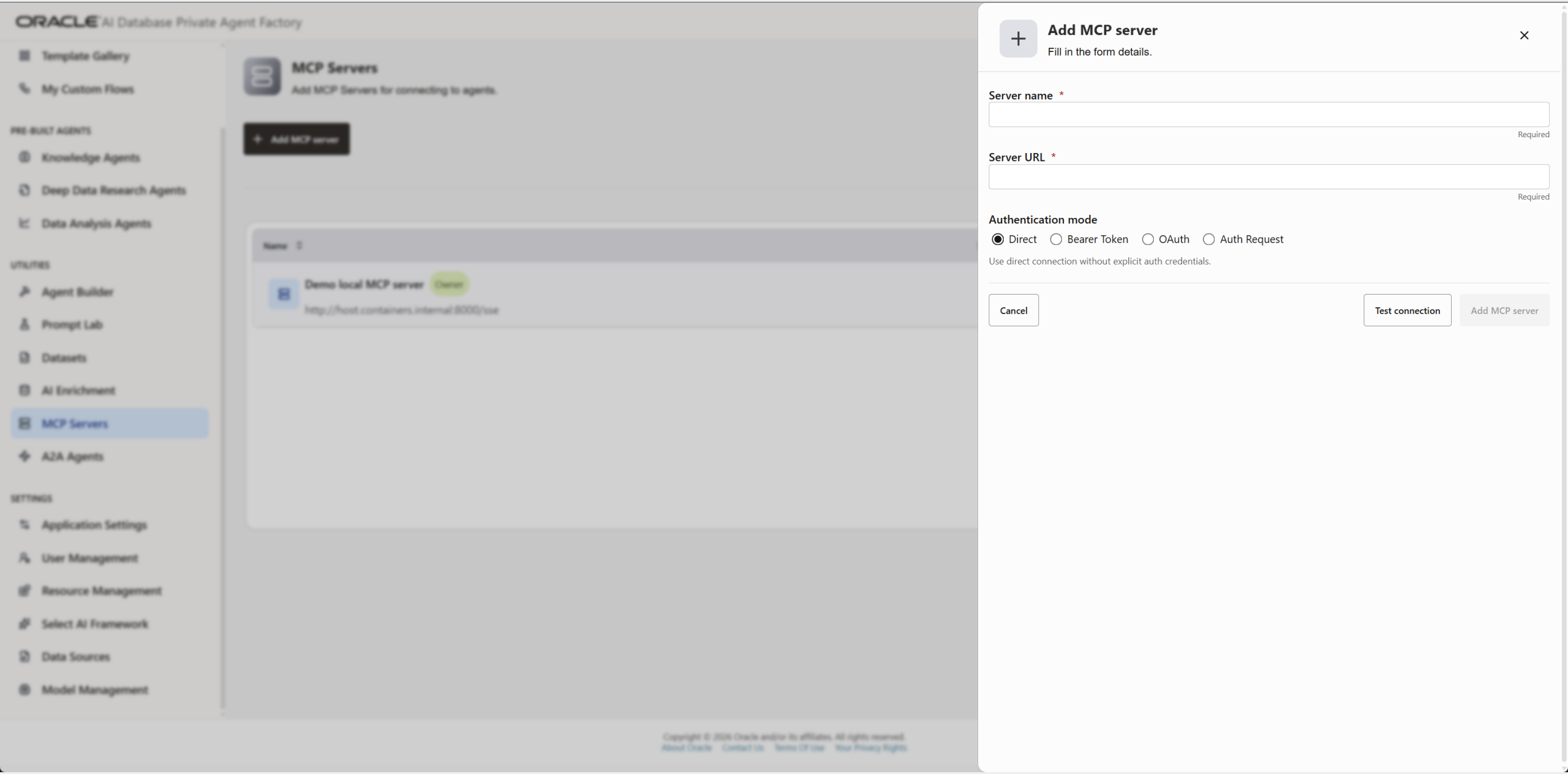
Name	Provider	Status	Last tested	Created at	Actions
xai.grok-3-mini Owner	OCI	Active	8/25/2026, 12:15:29 PM	5/13/2026, 3:14:28 PM	•••
meta.llama-4-maverick-17b-128e-instruct-fp8 Owner	OCI	Active	8/25/2026, 12:15:29 PM	5/13/2026, 3:05:38 PM	•••
xai.grok-4.3 Owner	OCI	Active	8/25/2026, 12:15:29 PM	5/13/2026, 3:07:45 PM	•••
xai.grok-4.20-0309-non-reasoning Owner	OCI	Active	8/25/2026, 12:15:29 PM	5/13/2026, 3:09:22 PM	•••
meta.llama-4-scout-17b-16e-instruct Owner	OCI	Active	8/25/2026, 12:15:29 PM	5/13/2026, 3:06:34 PM	•••
xai.grok-4 Owner	OCI	Active	8/25/2026, 12:15:29 PM	5/13/2026, 3:13:28 PM	•••

Copyright © 2026 Oracle and/or its affiliates. All rights reserved.
About Oracle Contact Us Terms Of Use Your Privacy Rights

Generative and embedding models configured once



MCP, SelectAI, Datasets are all configured centrally in PAF



Pre-built agents: Knowledge

Using the Oracle AI Database 26ai vector search capabilities.

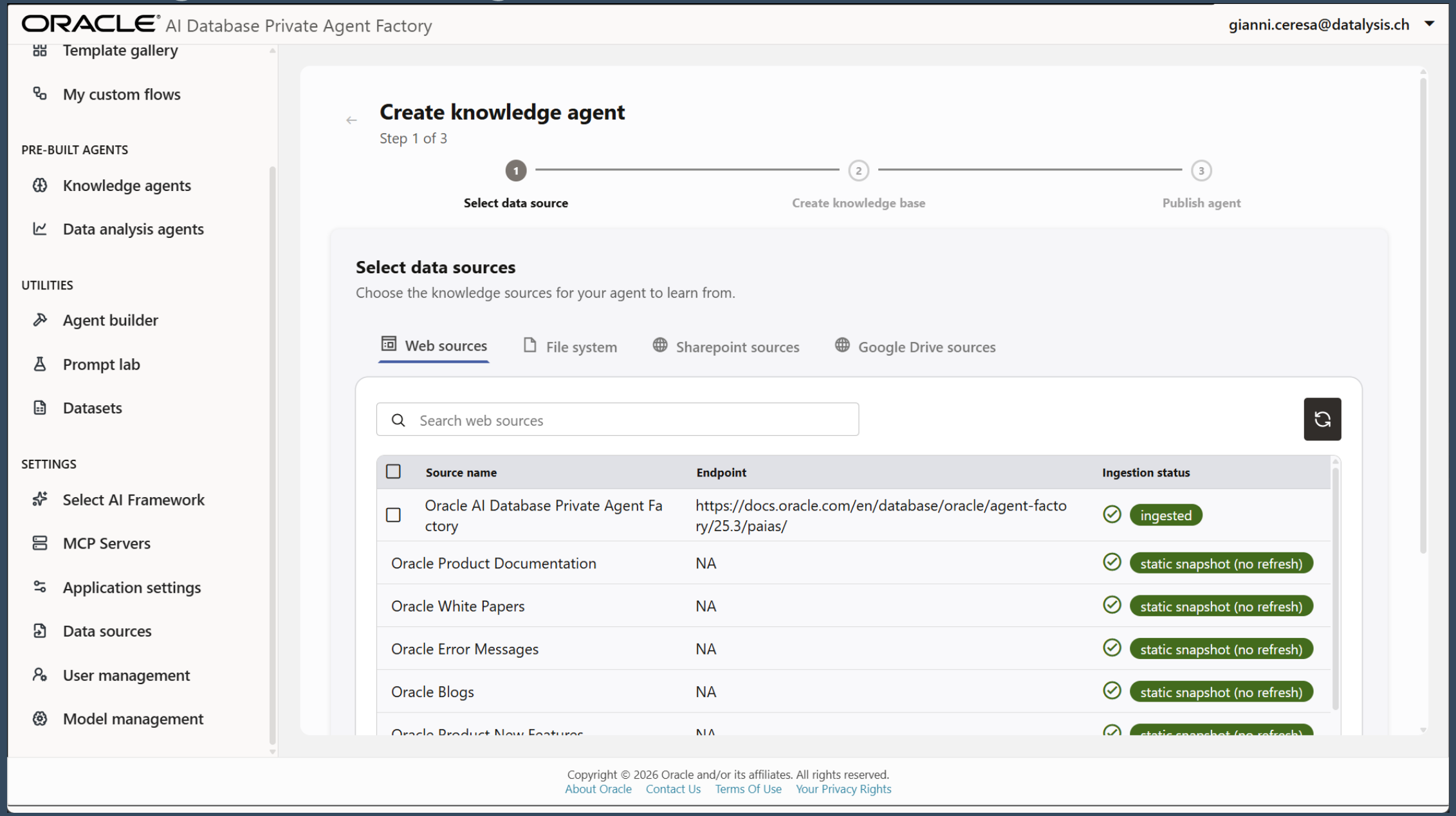
Fully automated RAG solution:

- Data sources
- Document loader (HTML, PDF, MS Word, text, JSON)
- Transformation (chunking, summarization)
- Embedding
- Similarity search

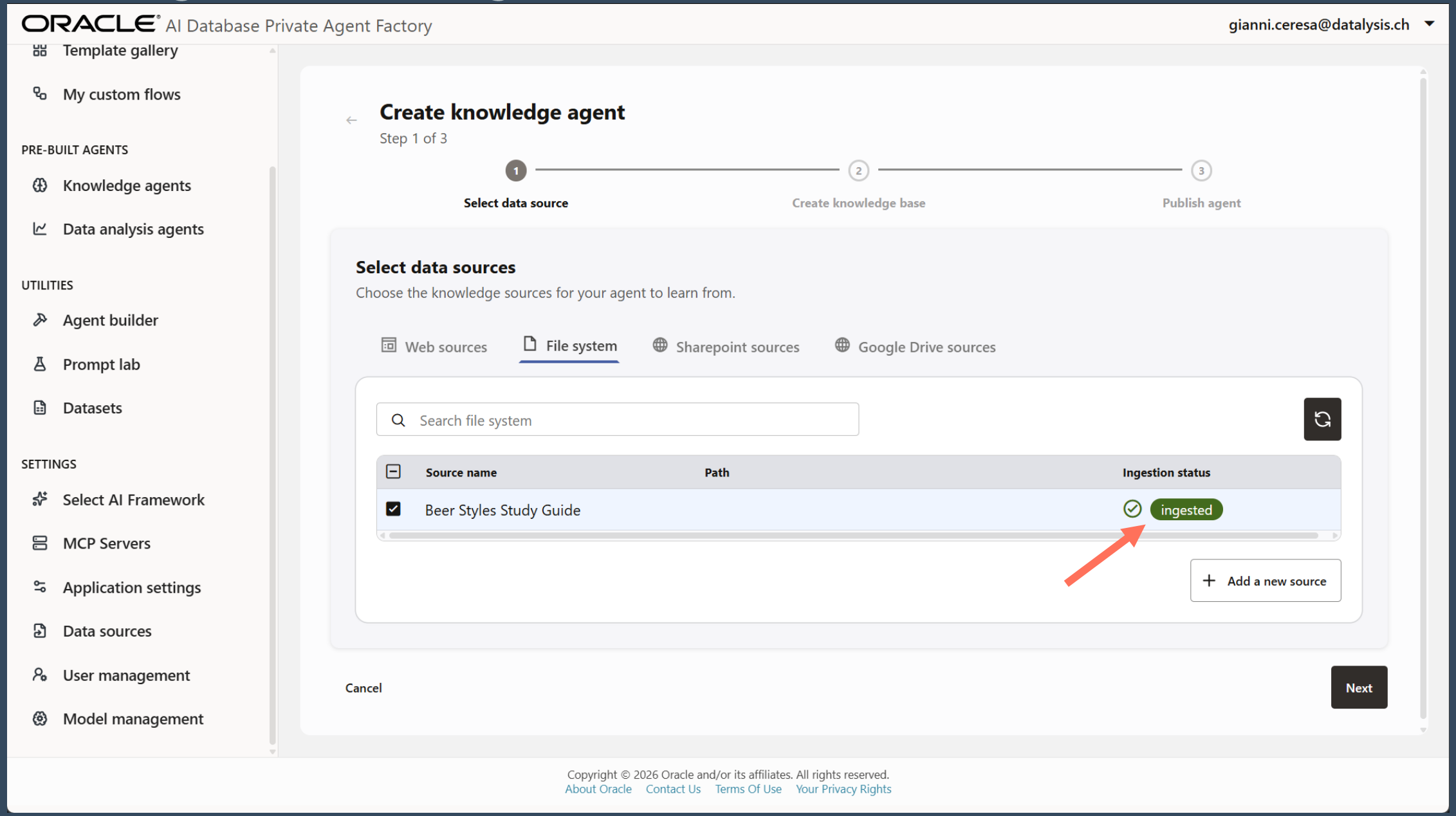
Supported data sources:

- Websites (automated crawler)
- File upload
- SharePoint
- Google Drive

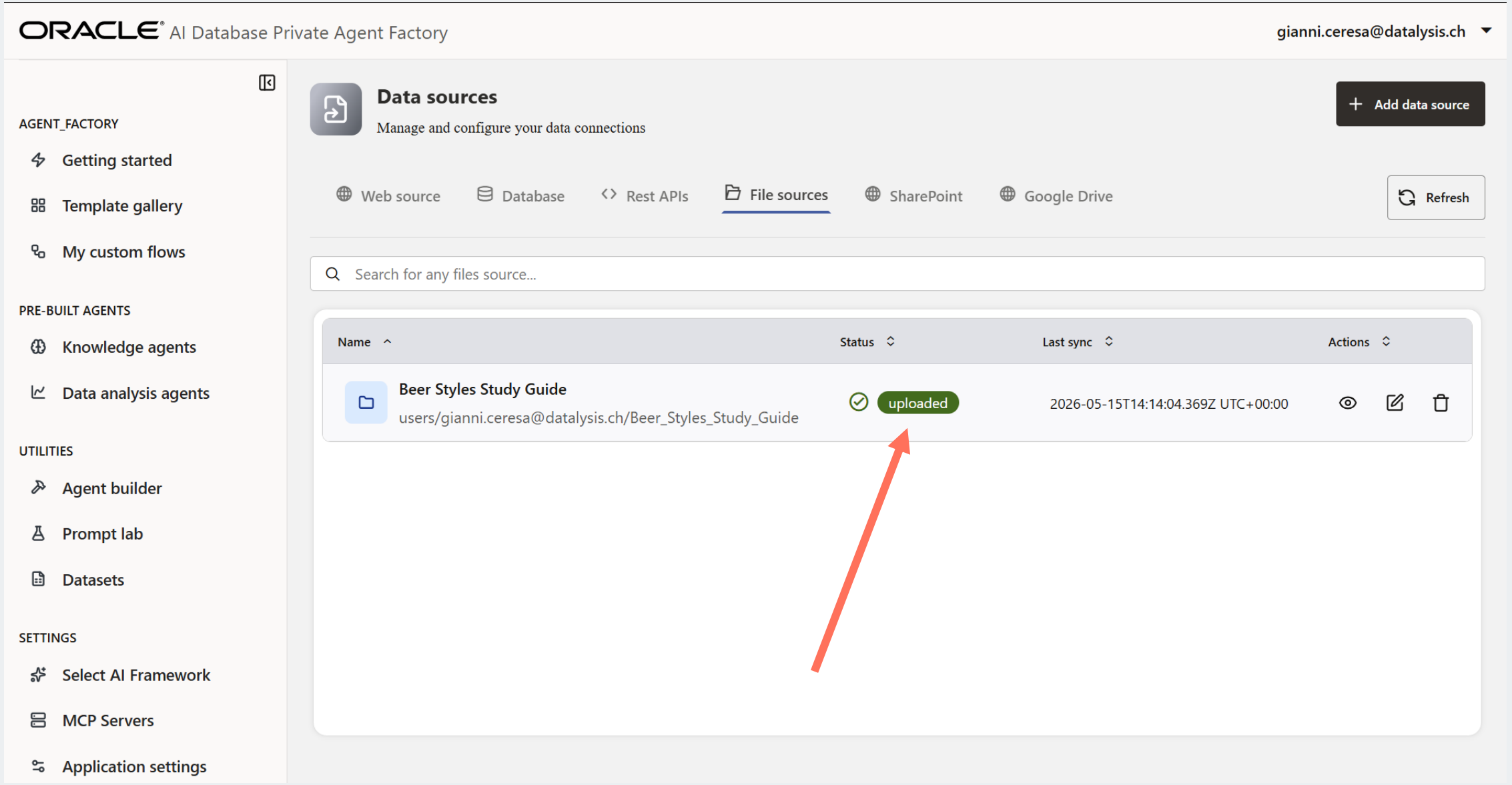
Pre-built agents: Knowledge



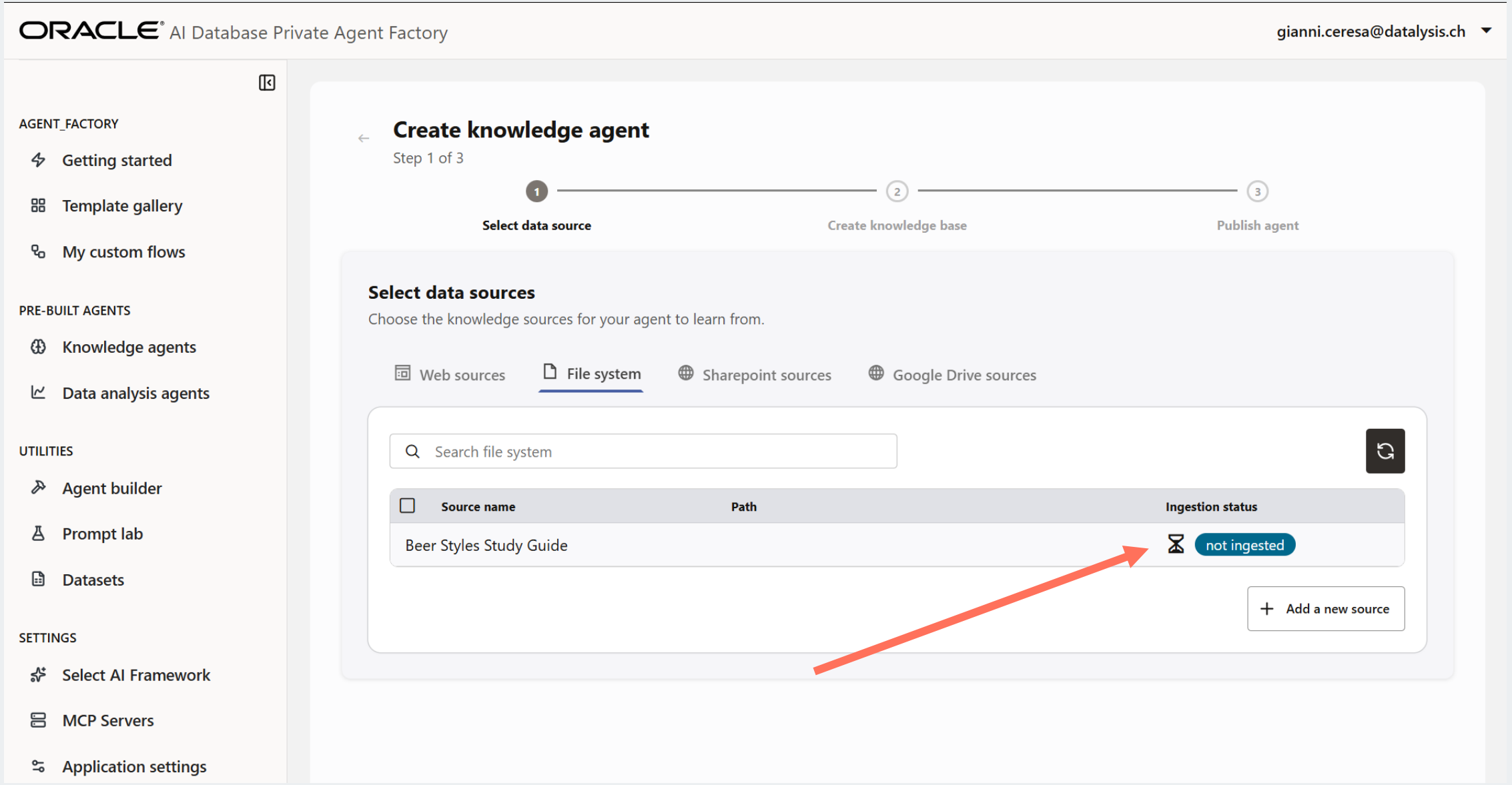
Pre-built agents: Knowledge



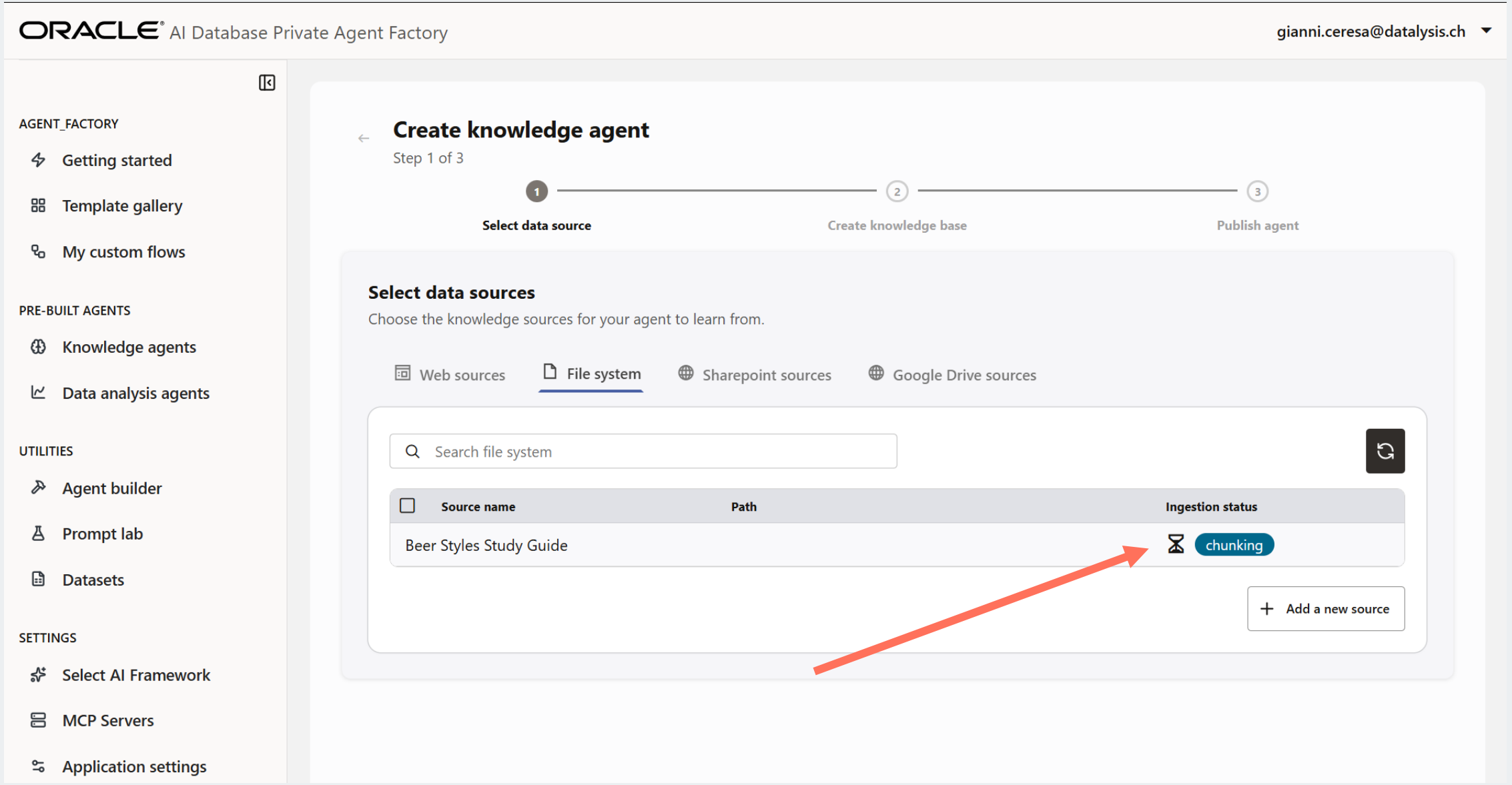
Pre-built agents: Knowledge – sources status



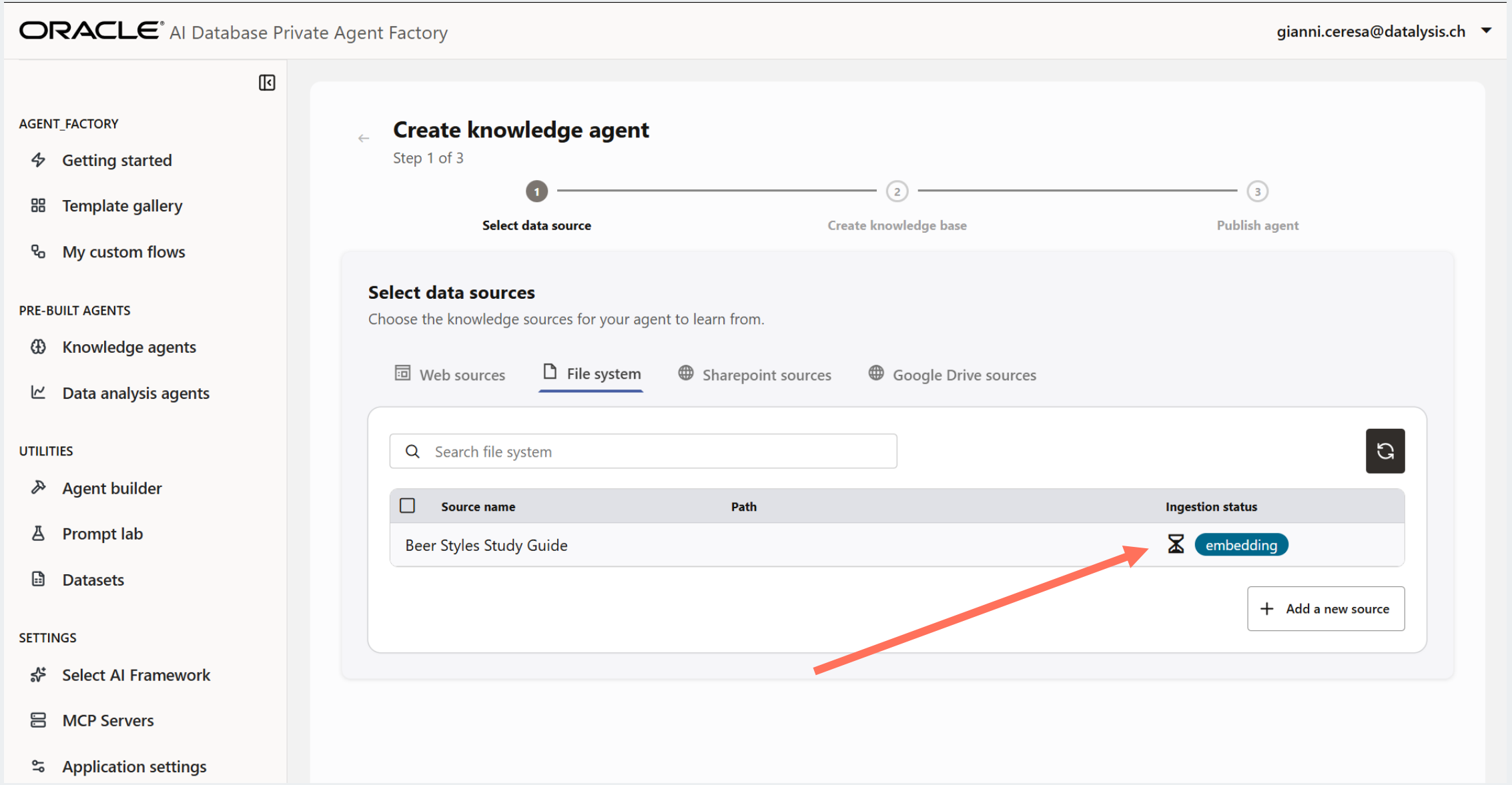
Pre-built agents: Knowledge – sources status



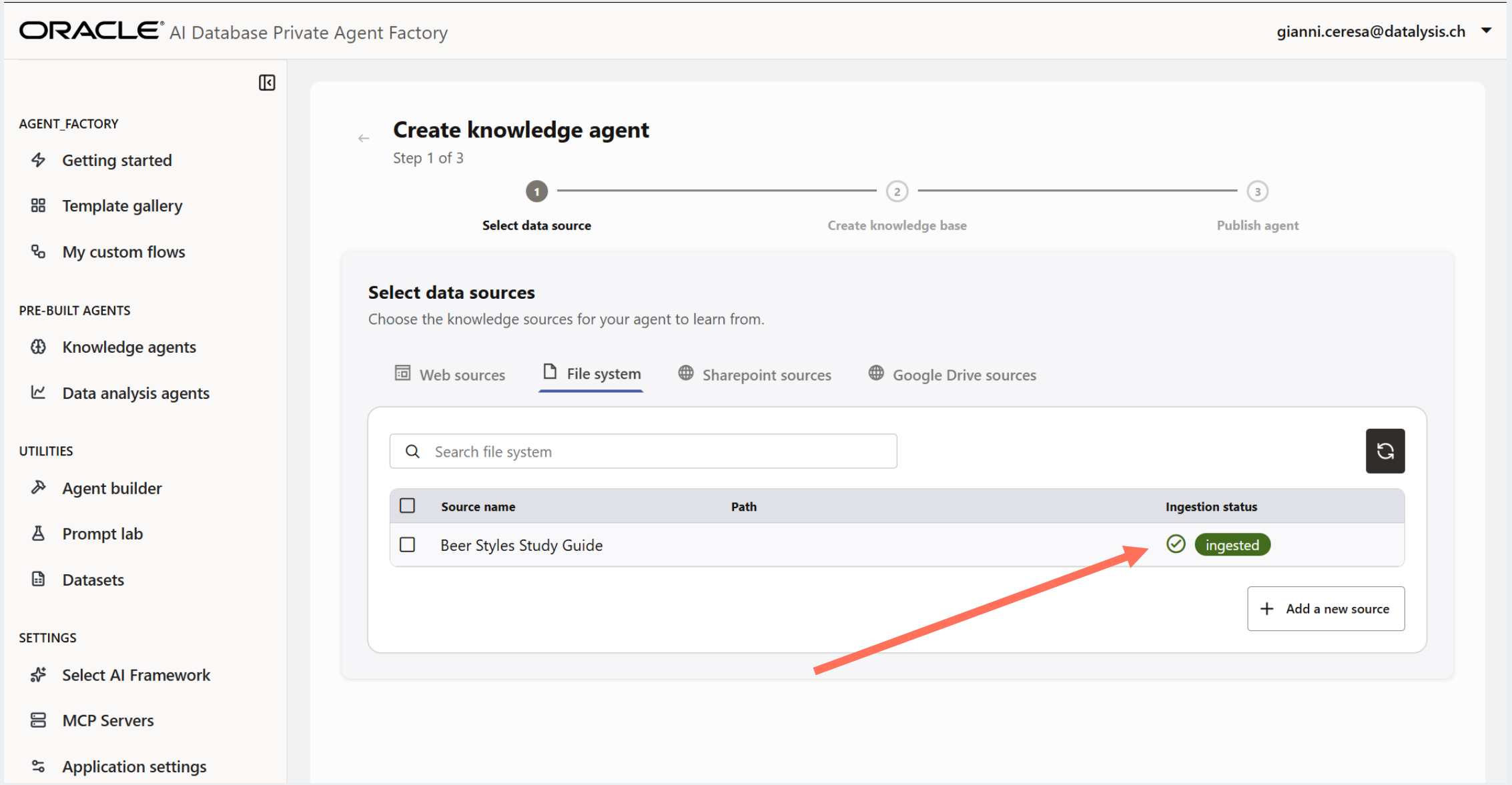
Pre-built agents: Knowledge – sources status



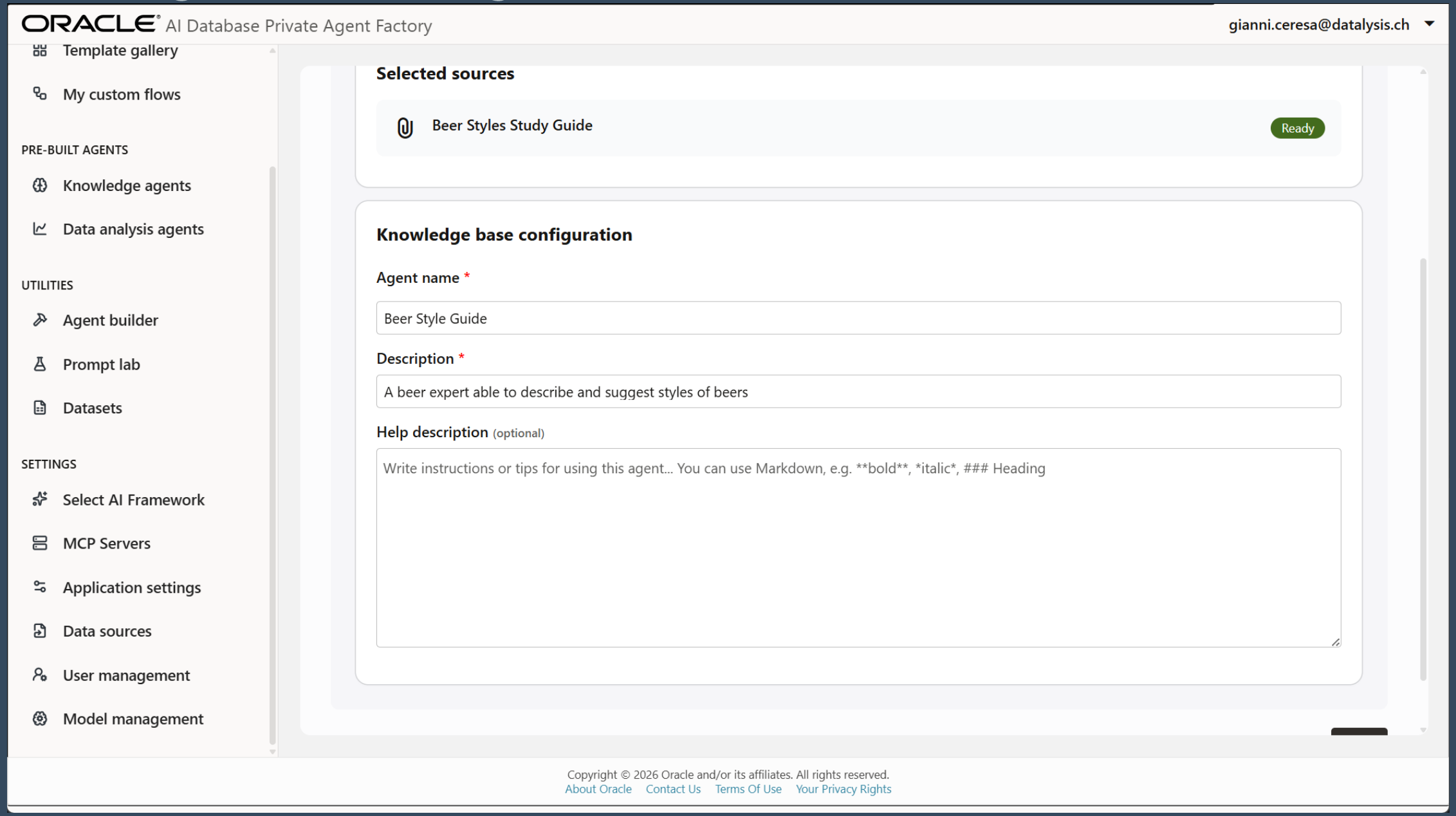
Pre-built agents: Knowledge – sources status



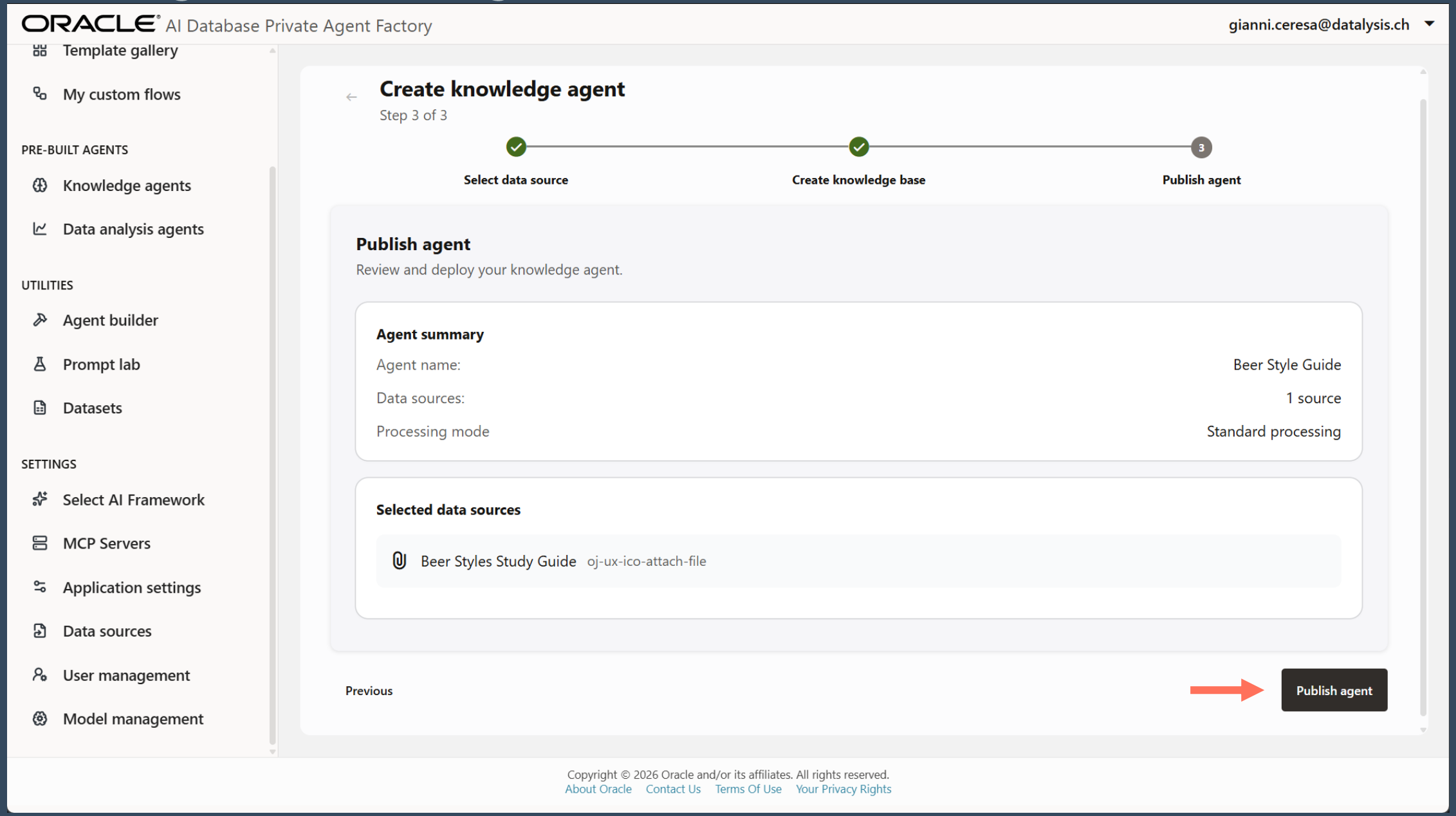
Pre-built agents: Knowledge – sources status



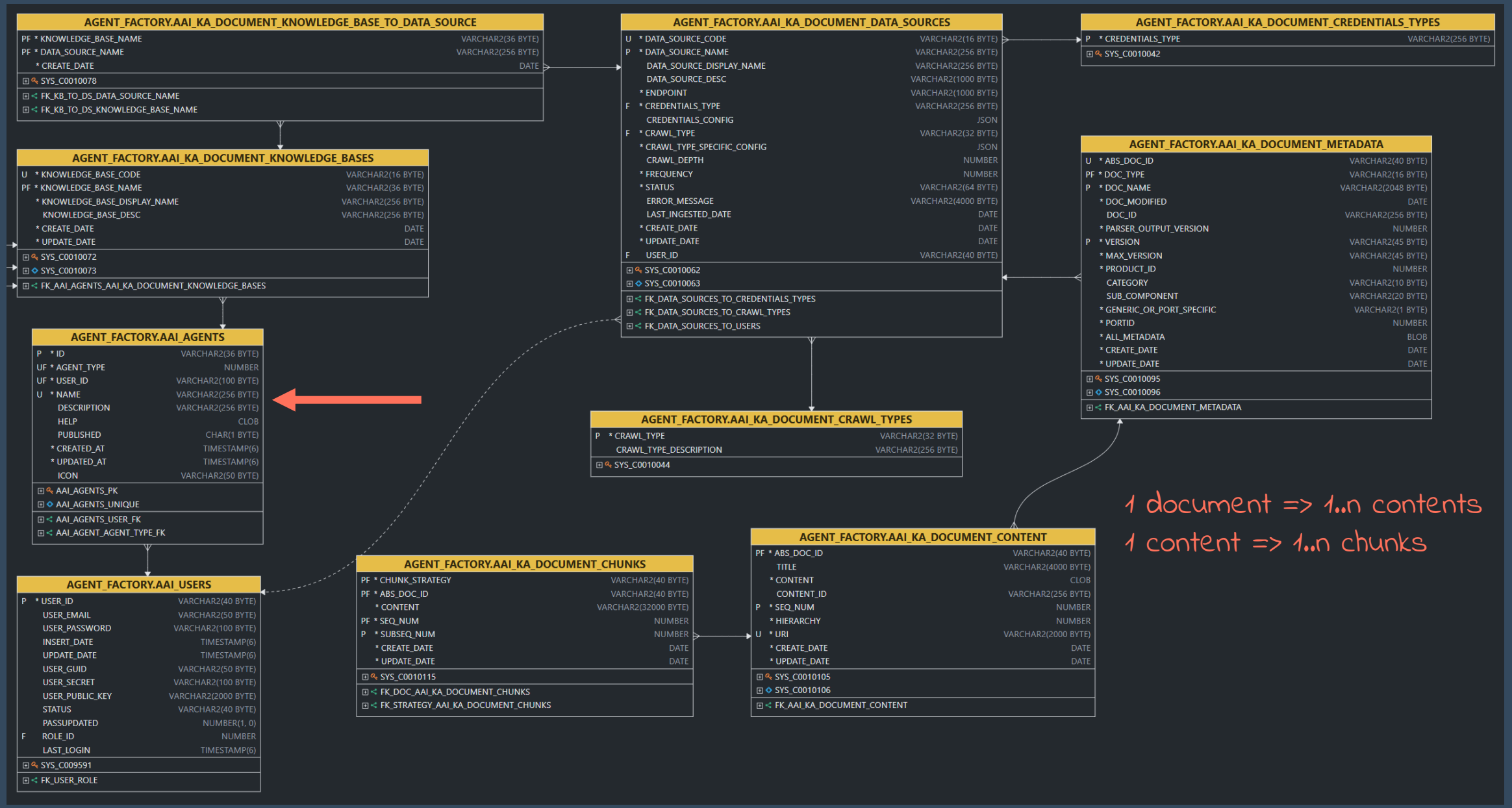
Pre-built agents: Knowledge



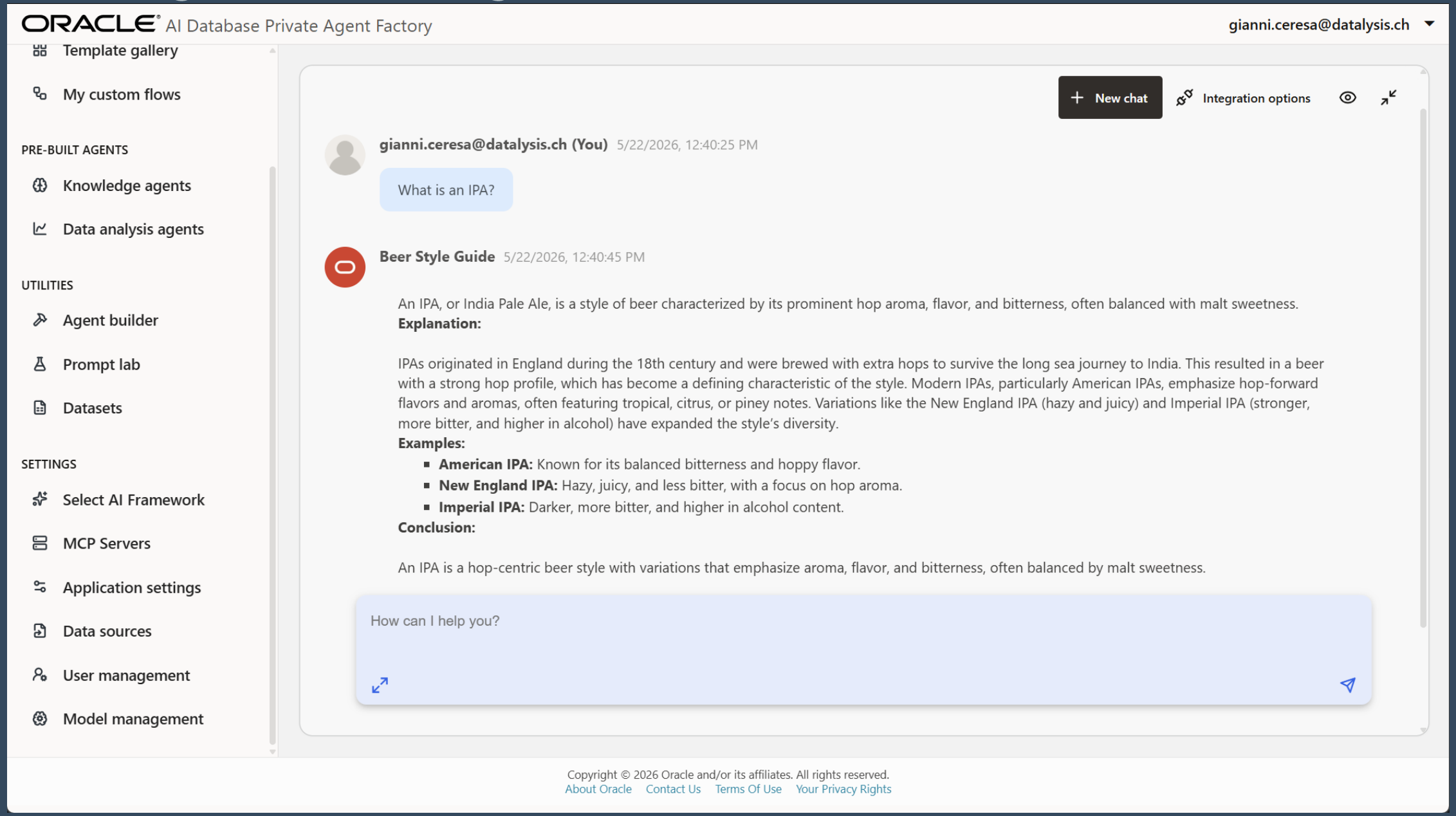
Pre-built agents: Knowledge



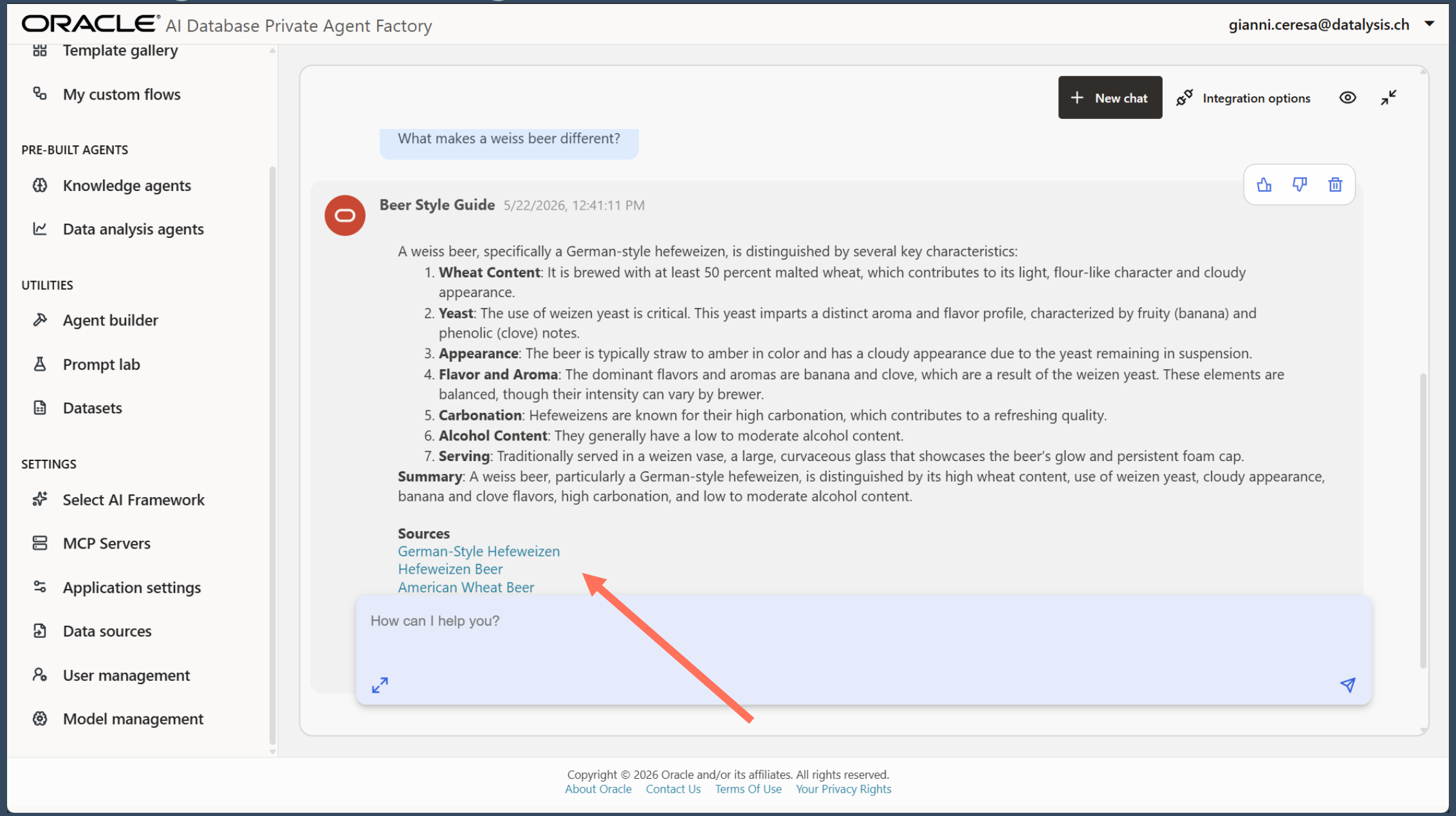
Pre-built agents: Knowledge



Pre-built agents: Knowledge



Pre-built agents: Knowledge



Pre-built agents: Knowledge

craftbeerdotcom-beer-styles.pdf 40 / 120 100%

German-Style Hefeweizen

Style Family: Wheat Beers

Arguably one of the most recognizable beer styles, the German-style hefeweizen offers a striking beer experience thanks to the use of distinctive wheat malt, unique yeast and uncharateristic appearance. This wheat beer breaks from the German beer mold, showcasing yeast-driven fruit and spice as well as bearing an eye-catching mystique. Don't let the cloudy hefeweizen deter you, this beer is one of the world's most enjoyable styles for beer geeks and neophytes, alike. The refreshing qualities of this highly-carbonated style have kept it alive for centuries. Try one for yourself and experience why that is, firsthand.

German-Style Hefeweizen

The German-style hefeweizen is straw to amber in color and brewed with at least 50 percent malted wheat. The aroma and flavor of a weissbier comes largely from the yeast and is decidedly fruity (banana) and phenolic (clove). The intensity of these wildly differing flavor qualities varies depending on the brewer, but the two are most commonly balanced. Hefeweizen typically contains a low to moderate alcohol content and is considered to be the most popular amongst the German weissbier variety.

Hefeweizen Beer

In German, "hefe" refers to the yeast which remains in suspension giving the German beer its cloudy appearance, and "weizen" denotes the use of wheat. This wheat beer is at its best when poured into a weizen vase, a large curvaceous glass which showcases the beer's beautiful glow and corrals its large, persistent foam cap given the style's characteristic effervescence. A German weissbier must showcase the weizen yeast's one-two punch of fruit and spice to be recognized as a German hefeweizen. The bright fruitiness of banana alongside the pungency of clove allow this beer to work well with a variety of lighter foods, such as salads, seafood and even a variety of egg dishes.

Tracking Down German Hefeweizen

German hefeweizen beer, like any other beer, tastes best when freshly poured at a brewery, and we want to help you find the breweries near you that serve the German hefeweizen that you seek. Use our "Find a Brewery" feature to locate breweries based on your location or search for a specific brewery by name. You can even search by state to find a small and independent brewer near you in order to track down your next favorite beer, which could be a German hefeweizen beer.

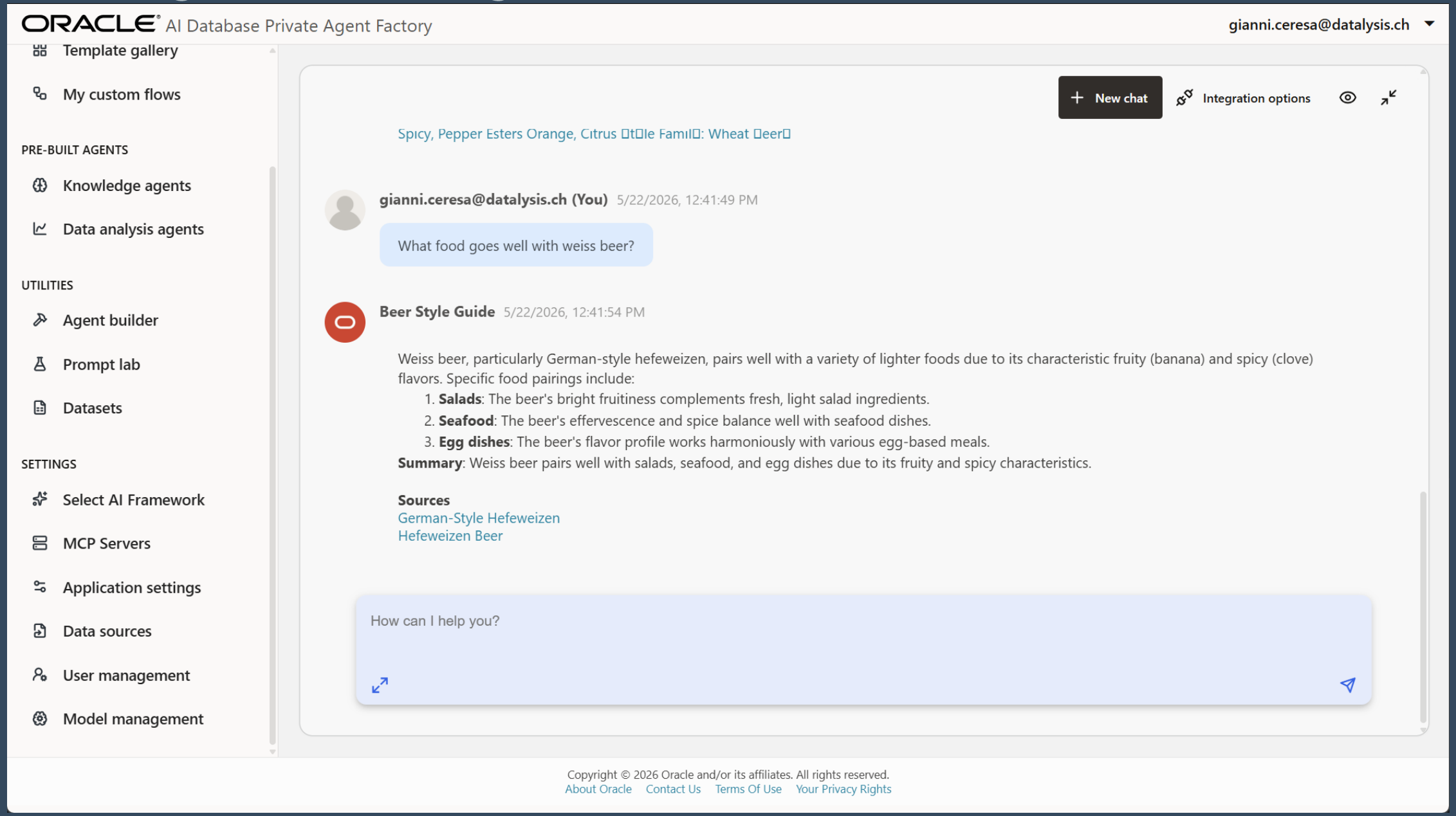
Finding your next favorite German hefeweizen doesn't have to stop here. If you enjoy the website and are interested in a convenient way to learn more about German beer, or any other kind of beer for that matter, sign up to have our newsletter delivered directly to your inbox.

Quantitative Style Statistics

OG 1.047 - 1.056
FG 1.008 - 1.016



Pre-built agents: Knowledge



Pre-built agents: Data Analysis

First version, limited capabilities (a single table).

Based on statistical analysis of the dataset, executed on the source, without sending the data to the LLM.

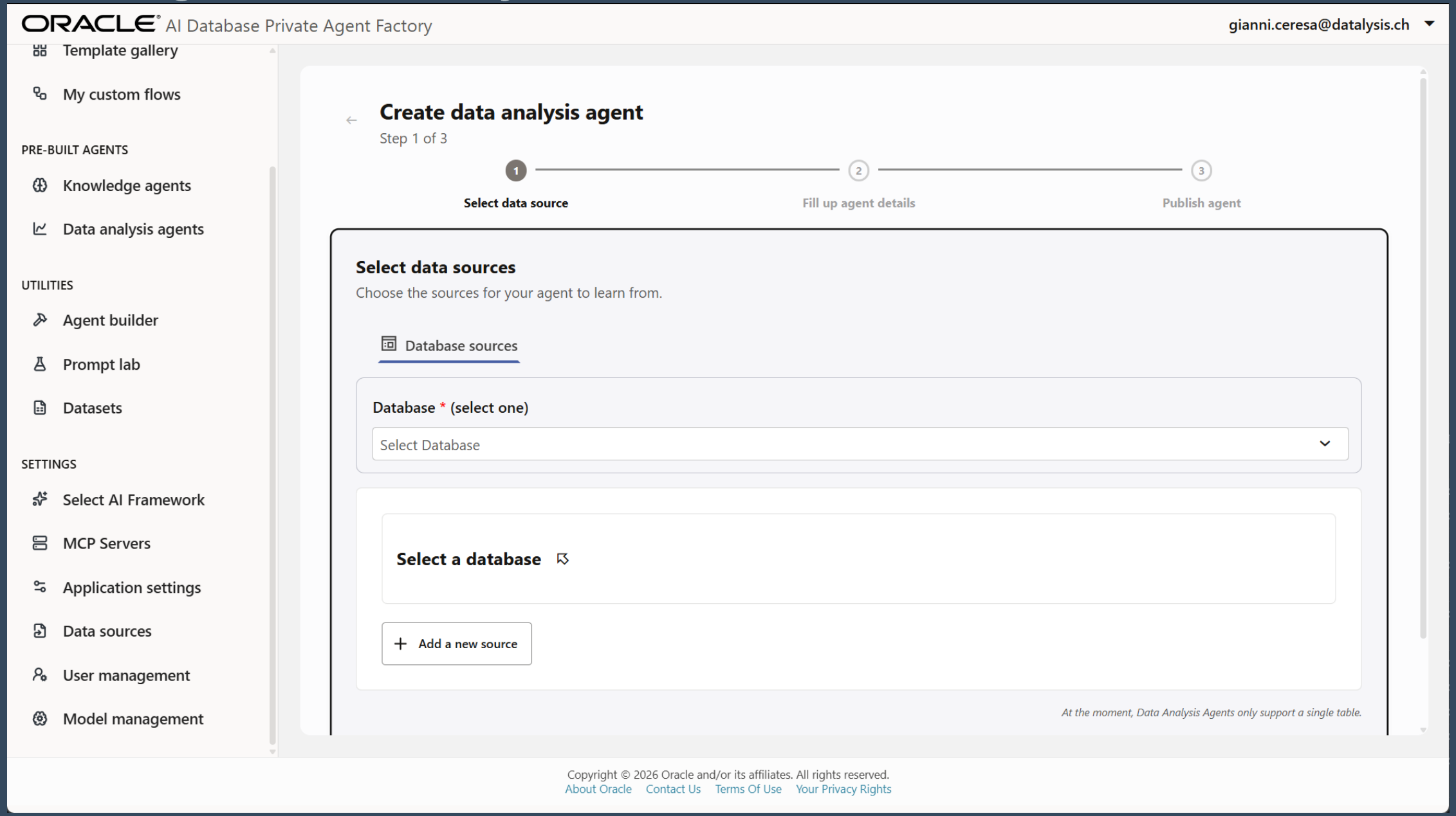
Support Oracle Database 19c or newer sources.

The LLM "explains" the data (turn numerical facts into sentences).

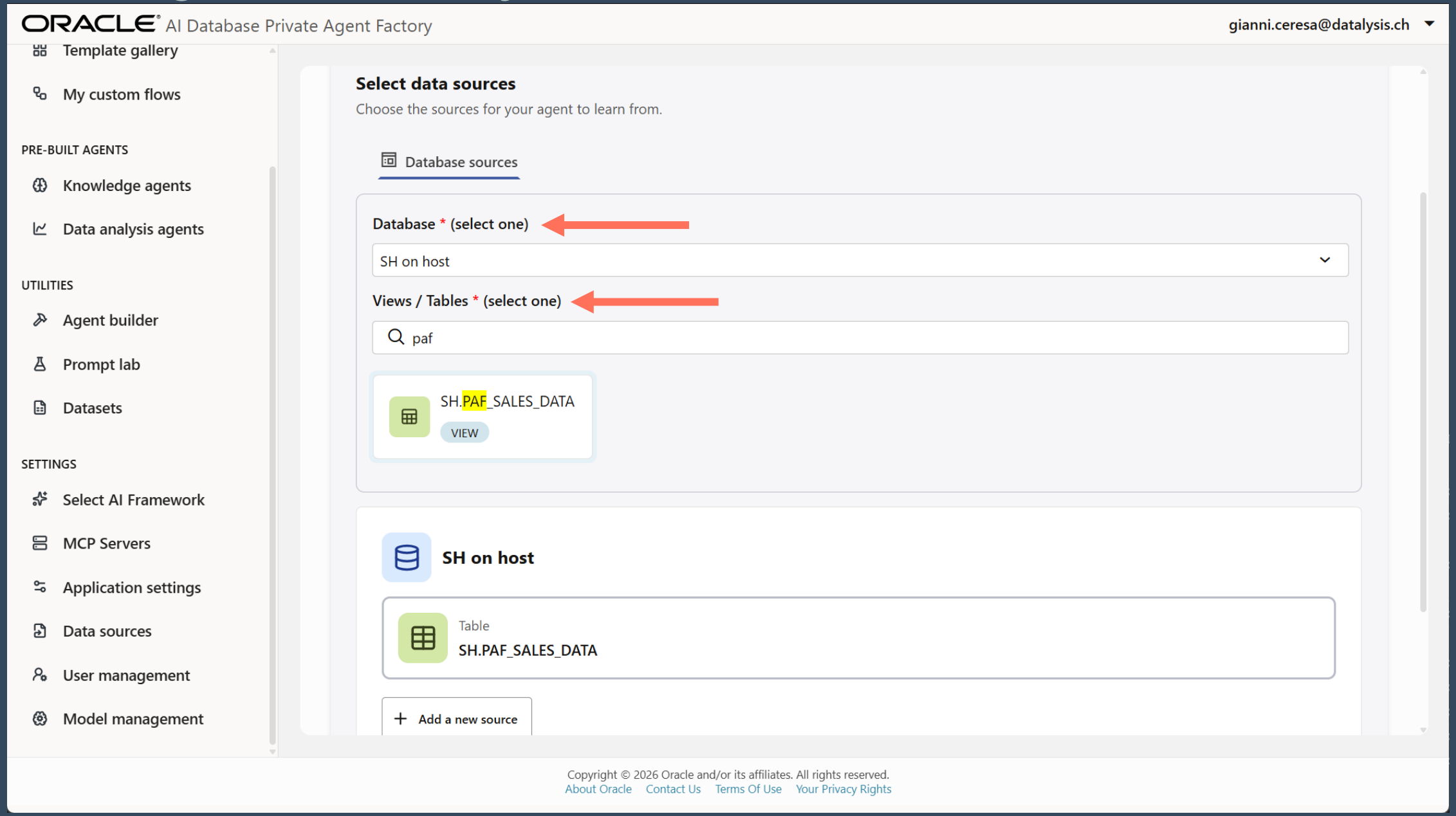
Basic charting to illustrate answers.

My opinion? Pointless!
Never saw any real dataset being a single table, therefore most of the time you would need to create a unified view in your DB to use this agent.

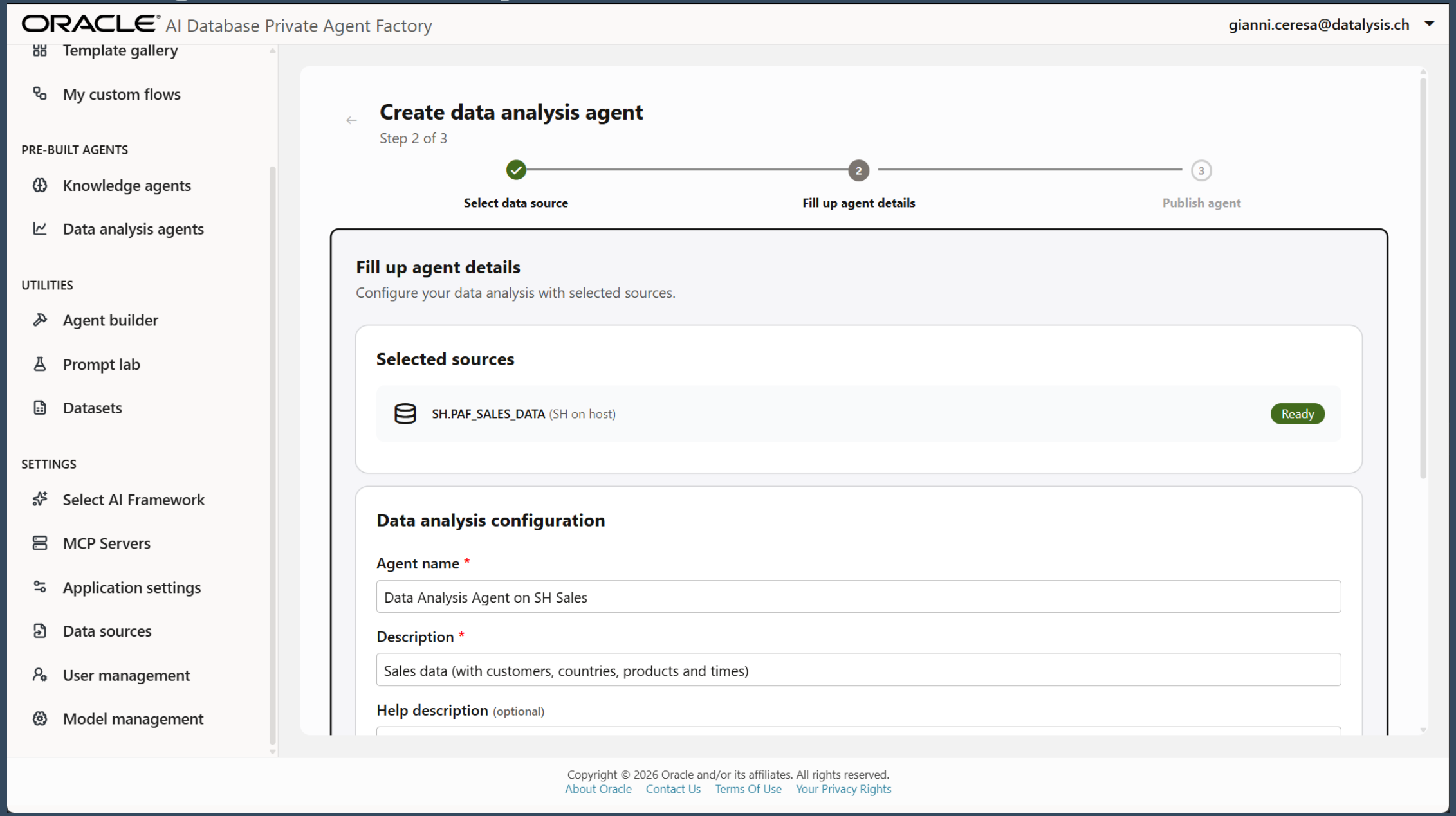
Pre-built agents: Data Analysis



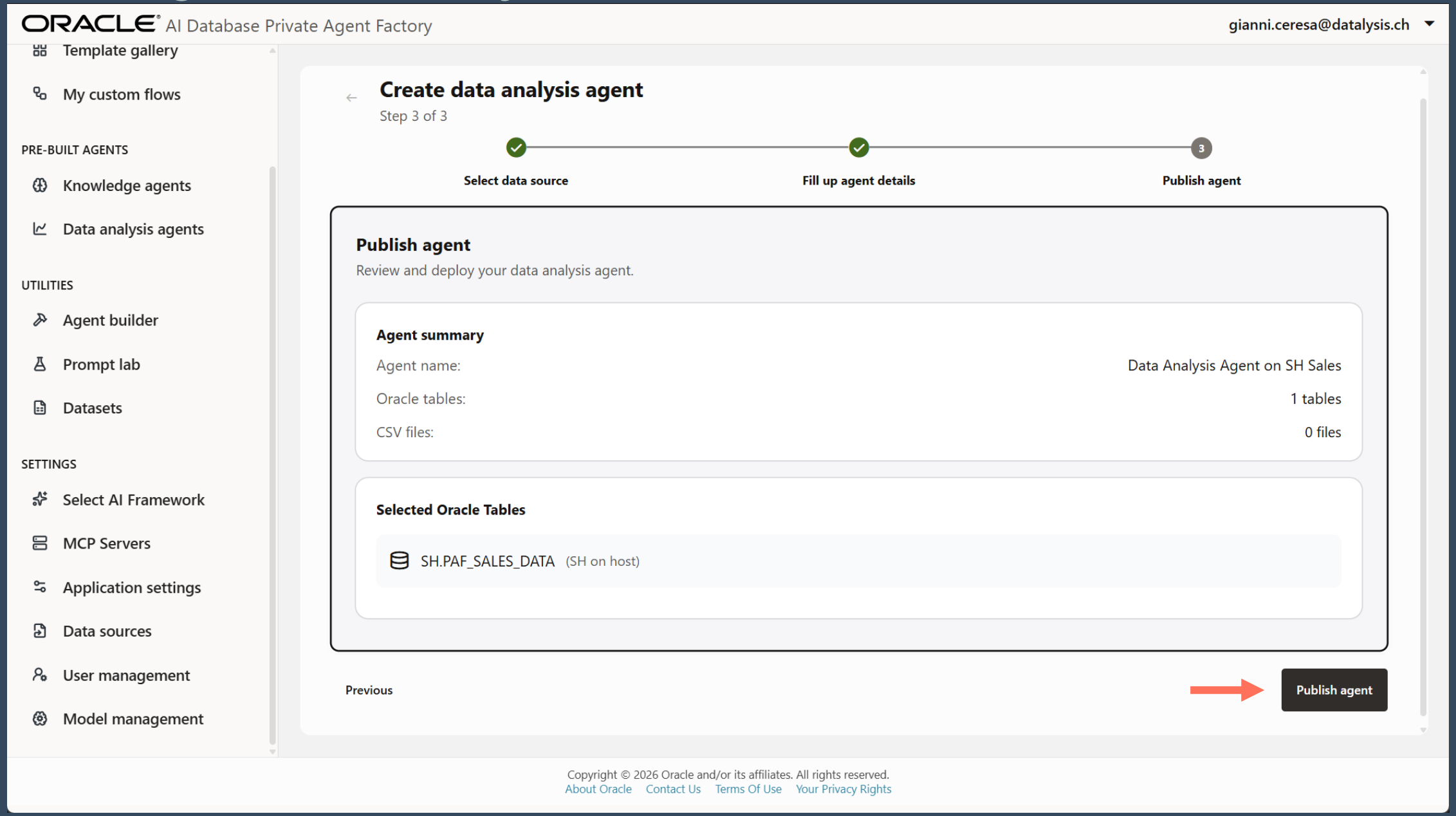
Pre-built agents: Data Analysis



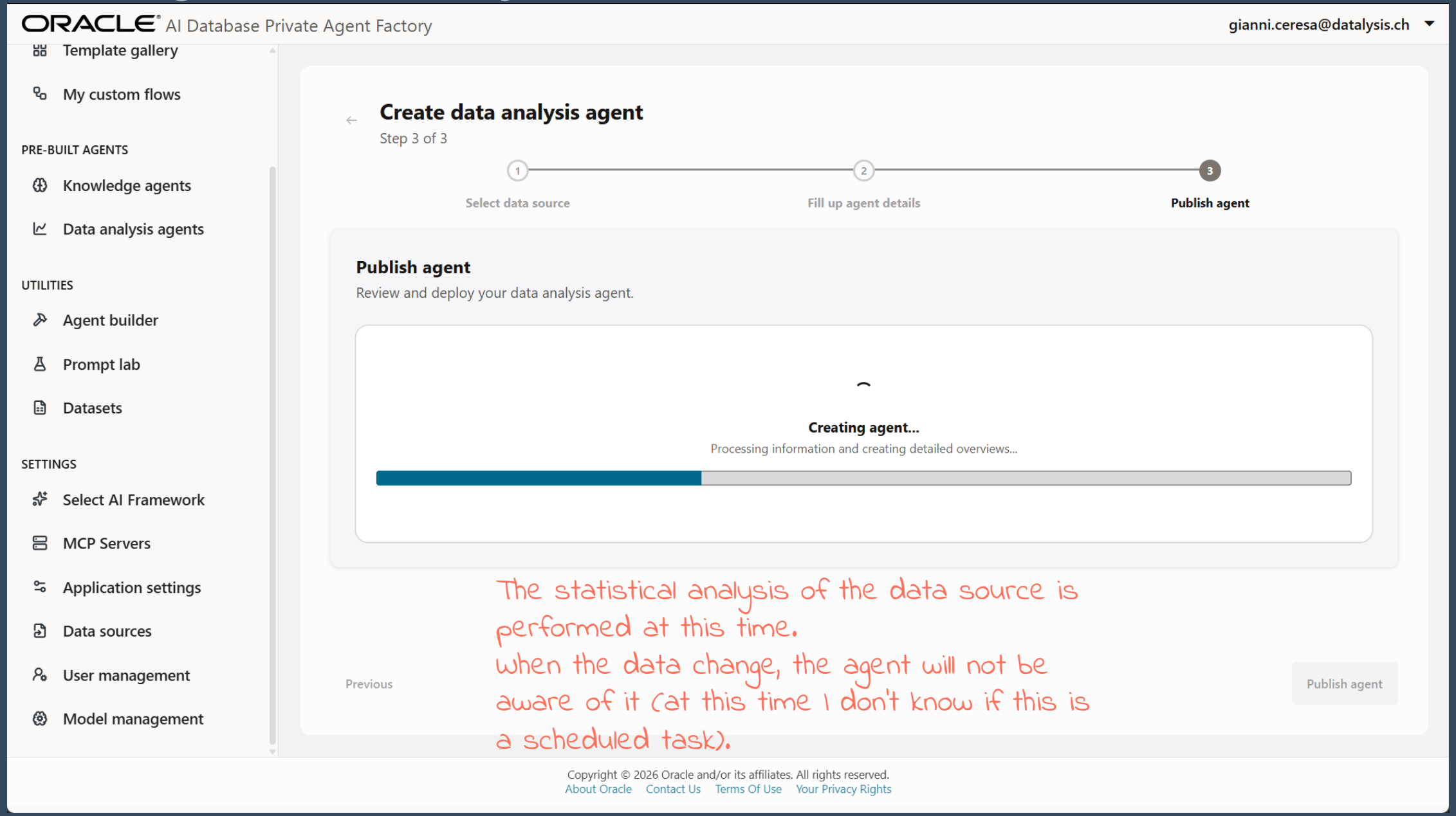
Pre-built agents: Data Analysis



Pre-built agents: Data Analysis

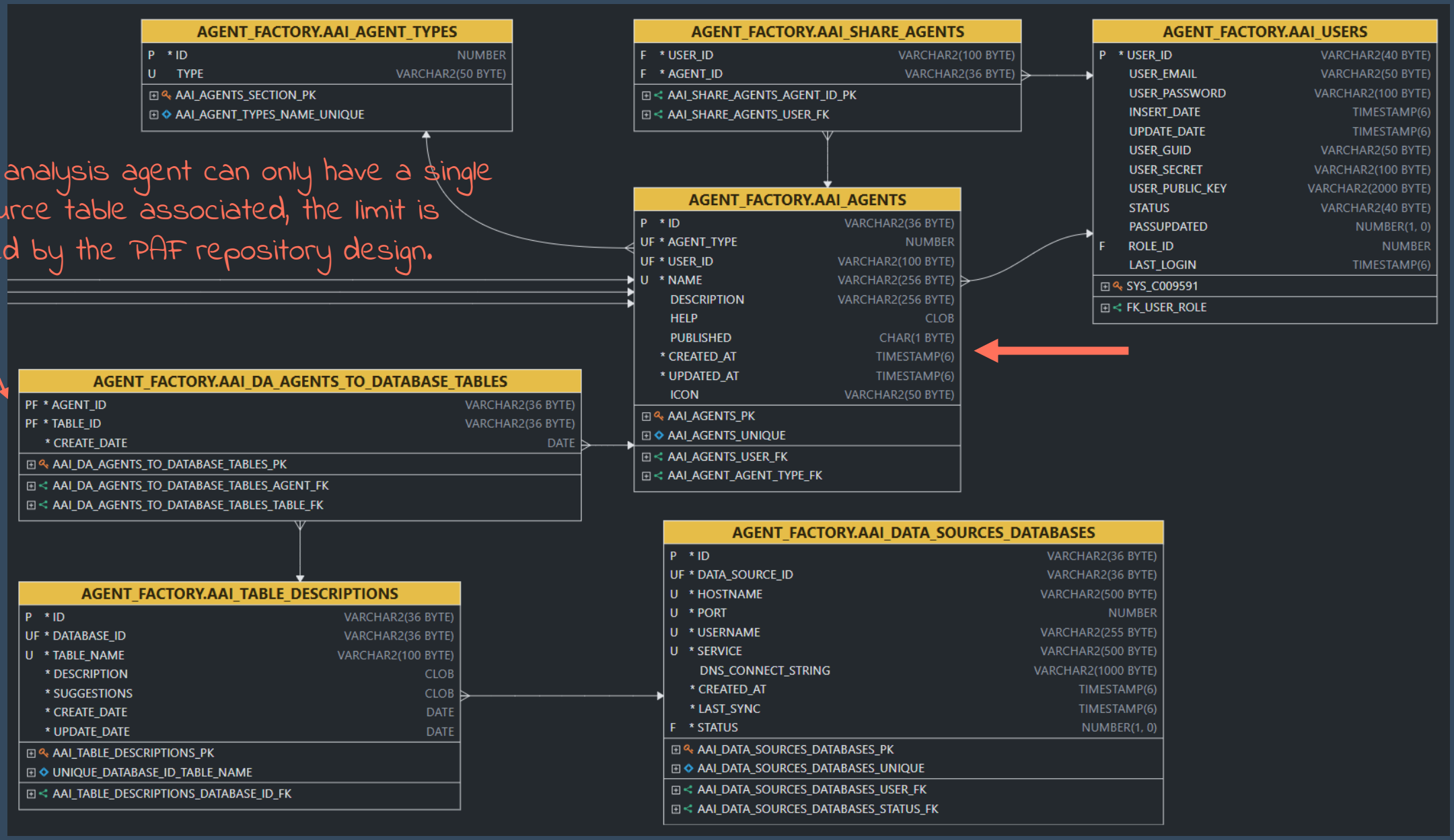


Pre-built agents: Data Analysis



The statistical analysis of the data source is performed at this time.
When the data change, the agent will not be aware of it (at this time I don't know if this is a scheduled task).

Pre-built agents: Data Analysis



Pre-built agents: Data Analysis


```
{
    "name": "SH.\"PAF_SALES_DATA\"",
    "type": "TABLE",
    "description": ""
},
{
    "name": "CUST_ID",
    "dataType": "NUMBER(22)",
    "type": "TABLE COLUMN",
    "variationAnalysis": "{\n    \"total_null_count\": 0,\n    \"total_non_null_count\": 918843,\n    \"minimum_value\": 2,\n    \"maximum_value\": 101000,\n    \"mean_value\": 7341.6531,\n    \"stddev_value\": 8899.19231
    "description": "Unique identifier for each customer; likely used as a primary key for customer records."
},
{
    "name": "CUST_LAST_NAME",
    "dataType": "VARCHAR2(40)",
    "type": "TABLE COLUMN",
    "variationAnalysis": "{\n    \"total_null_count\": 0,\n    \"total_non_null_count\": 918843,\n    \"top_5_most_frequent_values_from_sample\": [\n        {\n            \"value\": \"Elkin\",\n            \"frequency\": 6\n
    "description": "Customer's last name; used to identify and categorize customers."
},
{
    "name": "CUST_FIRST_NAME",
    "dataType": "VARCHAR2(20)",
    "type": "TABLE COLUMN",
    "variationAnalysis": "{\n    \"total_null_count\": 0,\n    \"total_non_null_count\": 918843,\n    \"top_5_most_frequent_values_from_sample\": [\n        {\n            \"value\": \"Carlos\",\n            \"frequency\": 7\n
    "description": "Customer's first name; used alongside last name for full customer identification."
},
{
    "name": "CUST_GENDER",
    "dataType": "CHAR(1)",
    "type": "TABLE COLUMN",
    "variationAnalysis": "{\n    \"total_null_count\": 0,\n    \"total_non_null_count\": 918843,\n    \"top_5_most_frequent_values_from_sample\": [\n        {\n            \"value\": \"M\",\n            \"frequency\": 651\n
    "description": "Customer's gender, represented as 'M' for male or 'F' for female; used for demographic analysis."
},
{
    "name": "CUST_MARITAL_STATUS",
    "dataType": "VARCHAR2(20)",
    "type": "TABLE COLUMN",
    "variationAnalysis": "{\n    \"total_null_count\": 301201,\n    \"total_non_null_count\": 617642,\n    \"top_5_most_frequent_values_from_sample\": [\n        {\n            \"value\": \"single\",\n            \"freque
    "description": "Customer's marital status; used for demographic segmentation, with some null values indicating missing data."
```

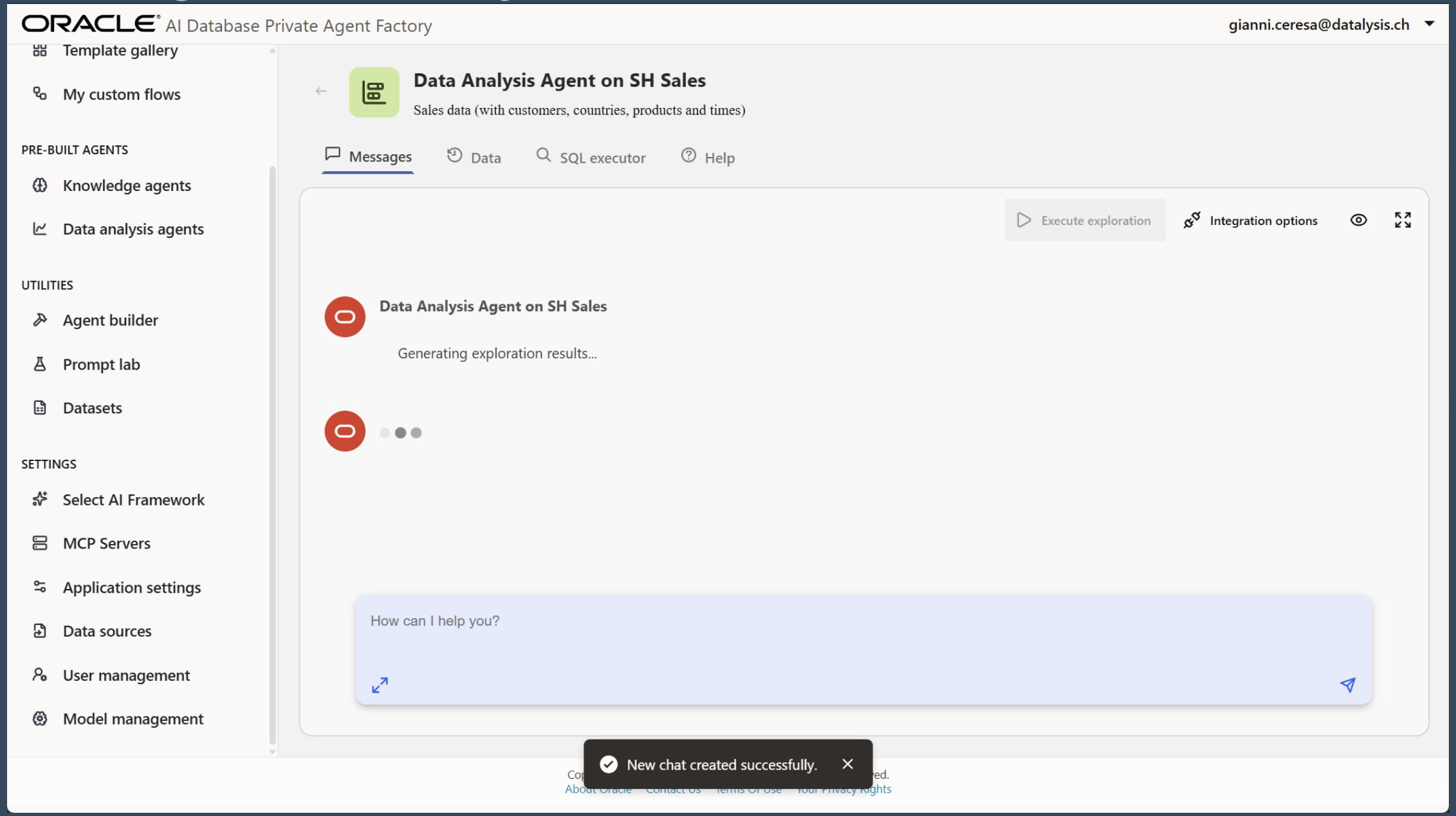
Pre-built agents: Data Analysis


```
{
    "name": "COUNTRY_NAME",
    "dataType": "VARCHAR2(40)",
    "type": "TABLE COLUMN",
    "variationAnalysis": "{\n    \"total_null_count\": 0,\n    \"total_non_null_count\": 918843,\n    \"top_5_most_frequent_values_from_sample\": [\n        {\n            \"value\": \"United States of America\",\n
    "description": "Name of the country where the customer resides; used for international segmentation."
},

variationAnalysis:
{
    "total_null_count": 0,
    "total_non_null_count": 918843,
    "top_5_most_frequent_values_from_sample": [
        {
            "value": "United States of America",
            "frequency": 558
        },
        {
            "value": "Germany",
            "frequency": 84
        },
        {
            "value": "United Kingdom",
            "frequency": 69
        },
        {
            "value": "Japan",
            "frequency": 67
        },
        {
            "value": "Italy",
            "frequency": 49
        }
    ]
}
```


Not sure how the frequency is measured.
Over 918843 rows, with no NULLs, it's impossible that the Top5 only has a total frequency of 827 !

Pre-built agents: Data Analysis



Pre-built agents: Data Analysis

ORACLE AI Database Private Agent Factory

gianni.ceresa@datalysis.ch

- Template gallery
- My custom flows

PRE-BUILT AGENTS

- Knowledge agents
- Data analysis agents

UTILITIES

- Agent builder
- Prompt lab
- Datasets

SETTINGS

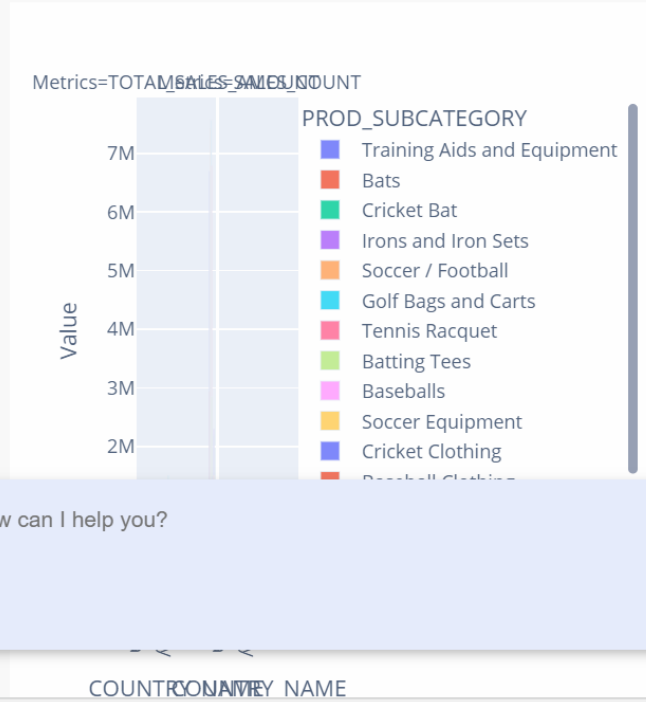
- Select AI Framework
- MCP Servers
- Application settings
- Data sources
- User management
- Model management

Execute exploration | Integration options

Data Analysis Agent on SH Sales 5/22/2026, 12:16:51 PM

What is the distribution of sales across different countries and how does it correlate with the product subcategories sold?

Visualization | Table | SQL



Data Analysis Agent on SH Sales 5/22/2026, 12:17:06 PM

What is the relationship between customer marital status and the total amount spent on purchases, and are there any notable trends?

Message | Visualization | Table | SQL

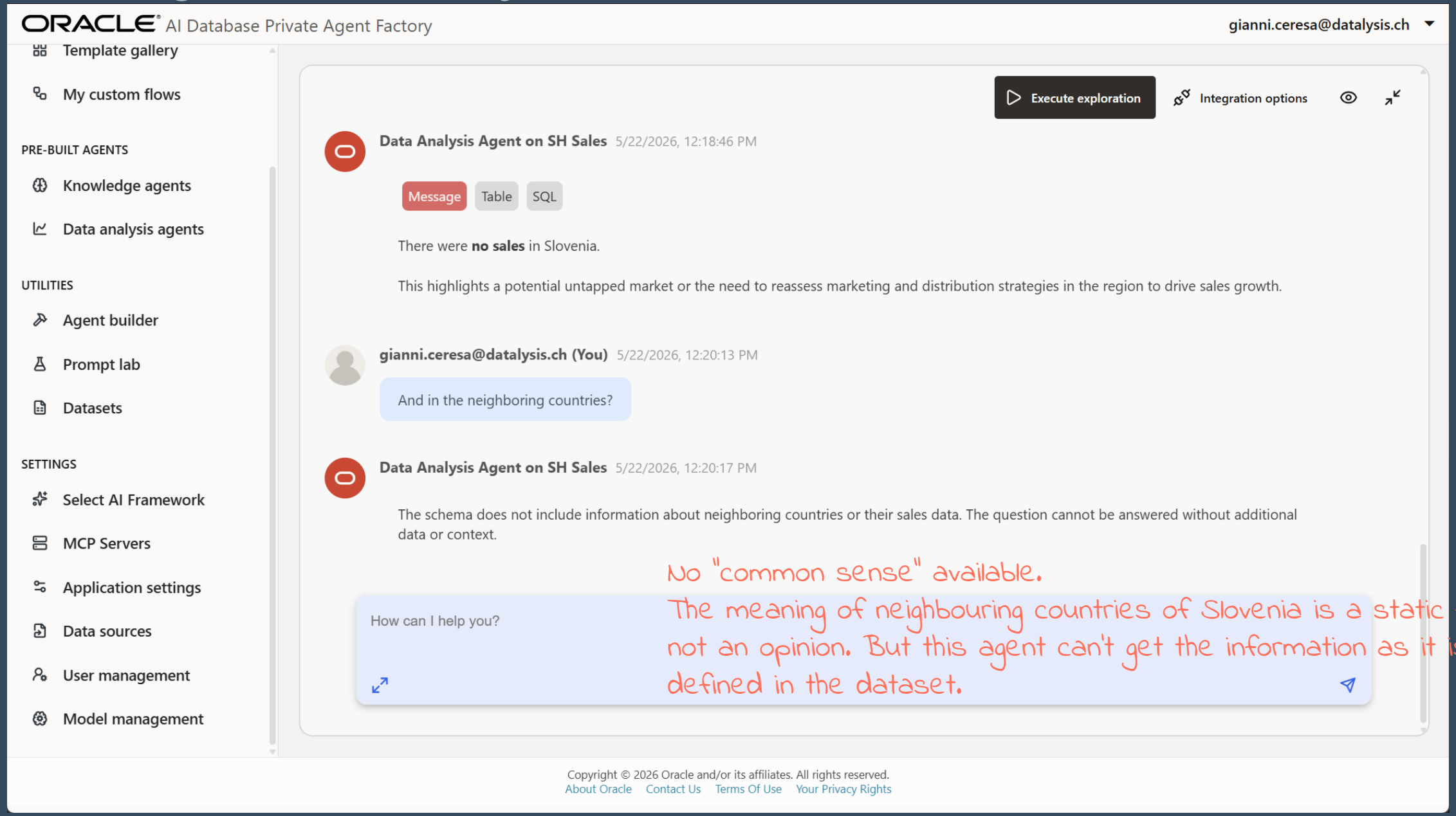
Single customers spent the most overall, with a total of **$34,908,947.22**, followed by those with an unspecified marital status at **$31,718,860.55**. Married customers, despite having the highest count (**2,268**), spent less than single customers, totaling **$29,556,286.97**. Divorced, widowed, and never-married customers spent significantly less, with totals of **$1,242,716.47**, **$642,579.09**, and **$121,361.76**, respectively. Separated customers spent the least, with only **$15,079.15**.

Key insight: Single customers drive the highest spending, while married customers, despite their larger numbers, spend less per capita. This suggests that marital status may influence purchasing behavior, with singles potentially having more disposable income or different spending priorities.

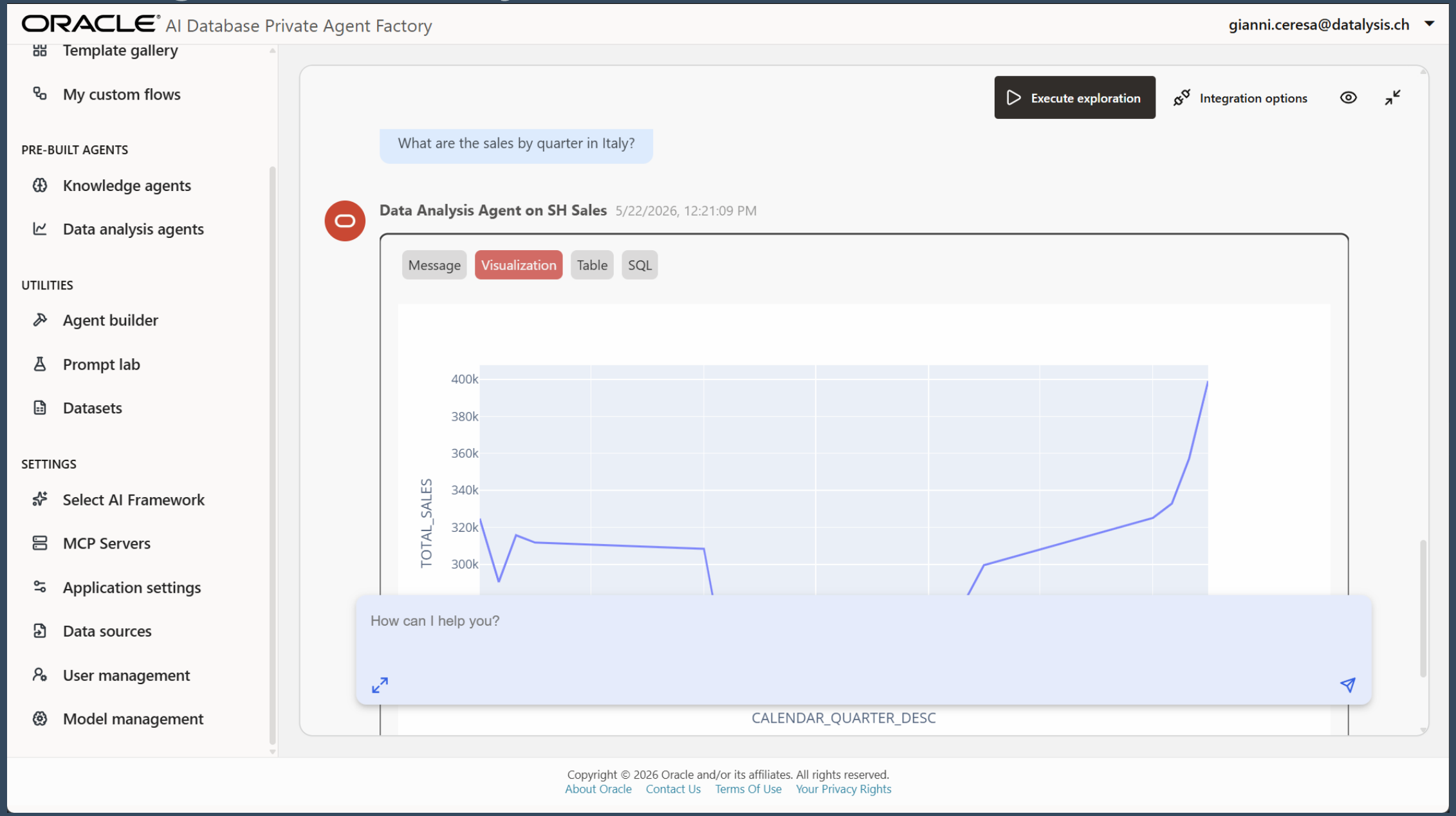
How can I help you?

Copyright © 2026 Oracle and/or its affiliates. All rights reserved.
About Oracle | Contact Us | Terms Of Use | Your Privacy Rights

Pre-built agents: Data Analysis



Pre-built agents: Data Analysis



Pre-built agents: Deep Data Research

Very new, so new that I didn't have time to build an example with sample data etc.

Let's use Oracle example instead: https://blogs.oracle.com/database/deep-data-research-agent-five-alarms-2583-pages-one-decision

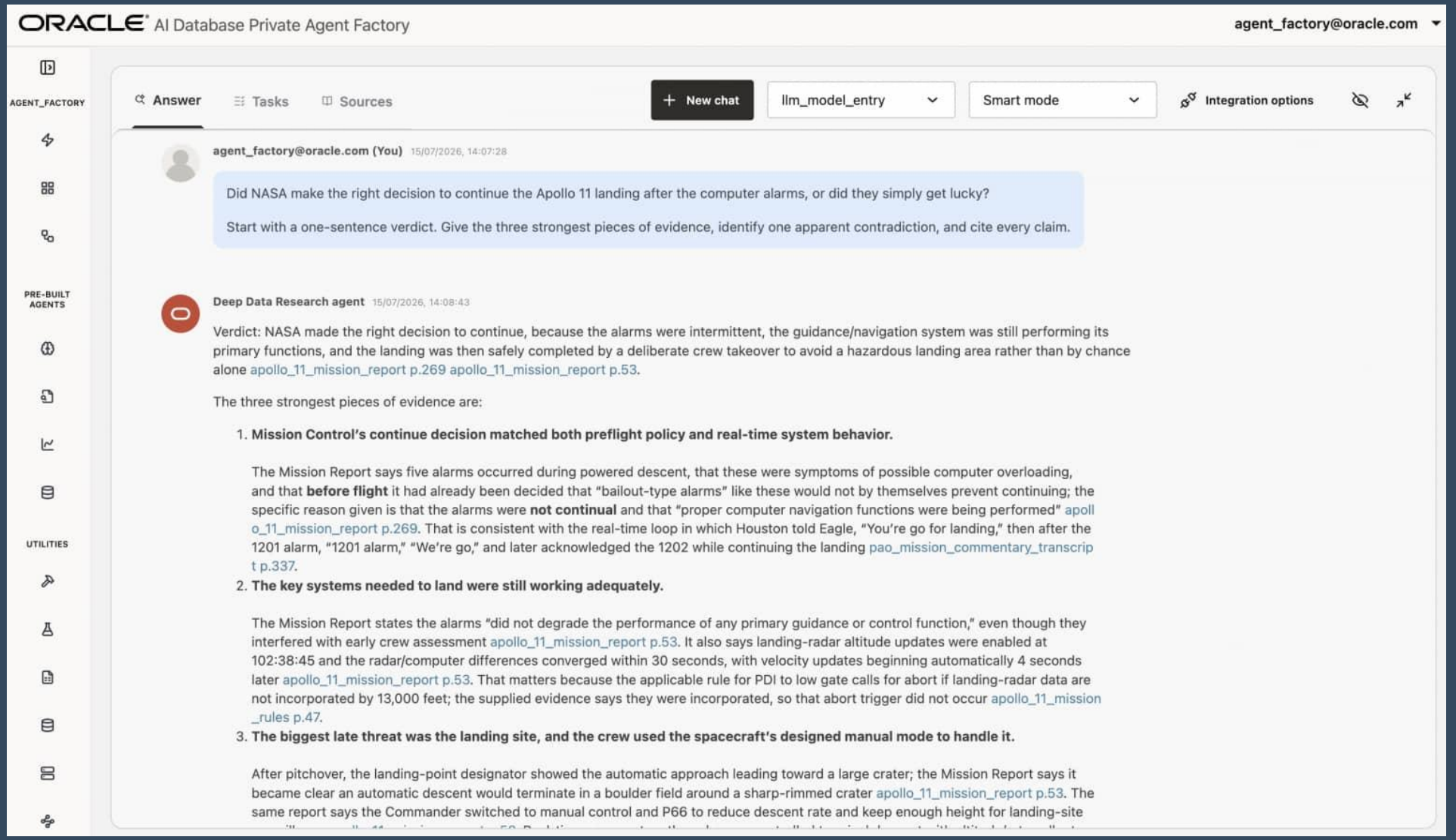
Recreating the Apollo 11 mission investigation

Sources:

- Flight plan
- Mission rules
- Transcripts of air-to-ground communications
- Public-affairs commentary
- Transcript of onboard voice discussions
- Post-flight mission report

2'583 pages of content

Pre-built agents: Deep Data Research



Agents/workflows from scratch:

No-code building

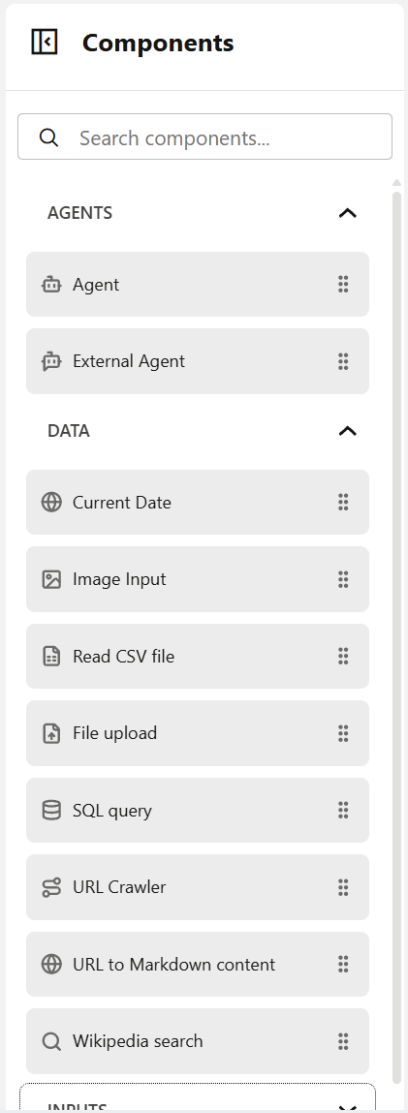
Agents/workflows

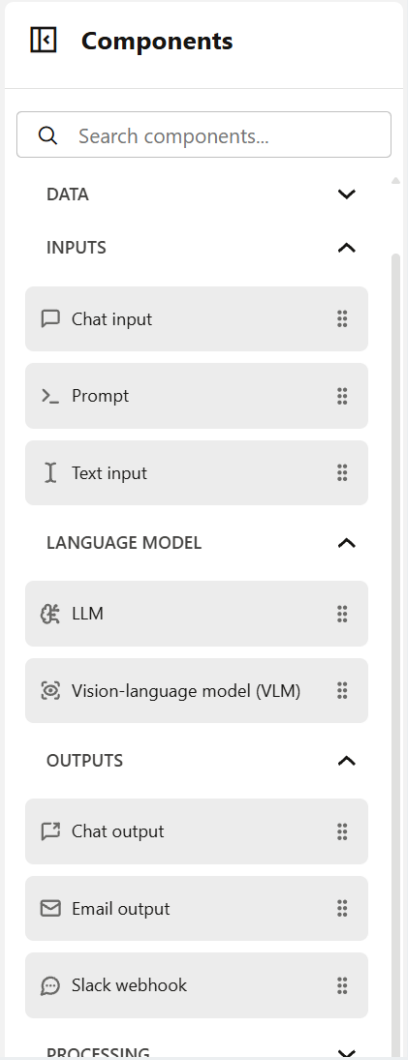
- Limited inputs, outputs supported for now
  - “chat” input
  - “chat” output
  - Email sending
  - Slack webhooks

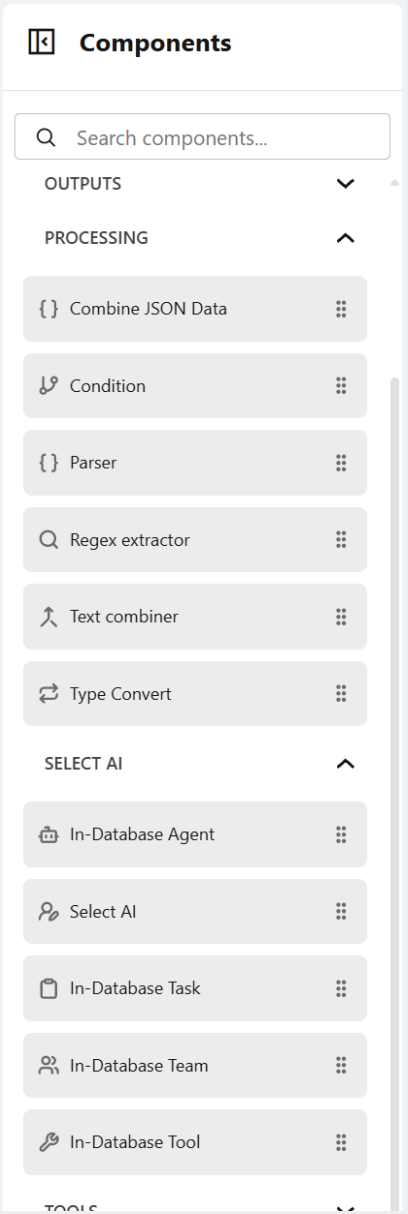
- Difficult to predict future available integrations. Unsure if it’s based on customers requests, Oracle internal needs, or something else...

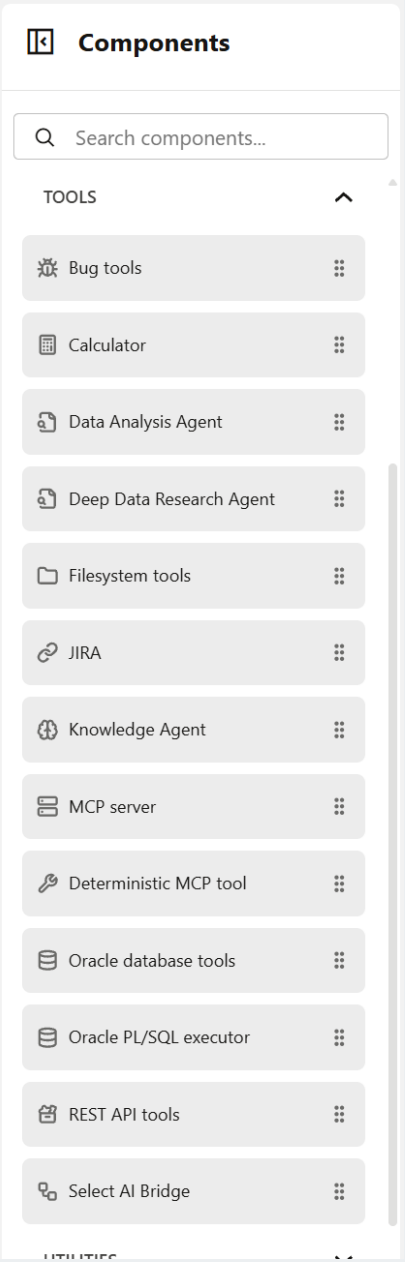
- Custom integrations allows you to connect to anything you need
  - Expose them as a MCP tool in a self-hosted MCP server (for example with FastMCP in python)
  - Use the “Deterministic MCP tool” node to execute the tool (dynamic input available, tool output exposed for further processing)

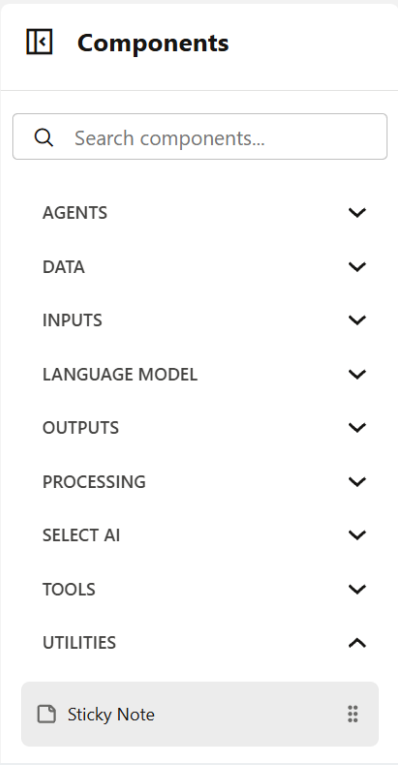
Agents/workflows: building nodes. 43 available nodes in v26.7



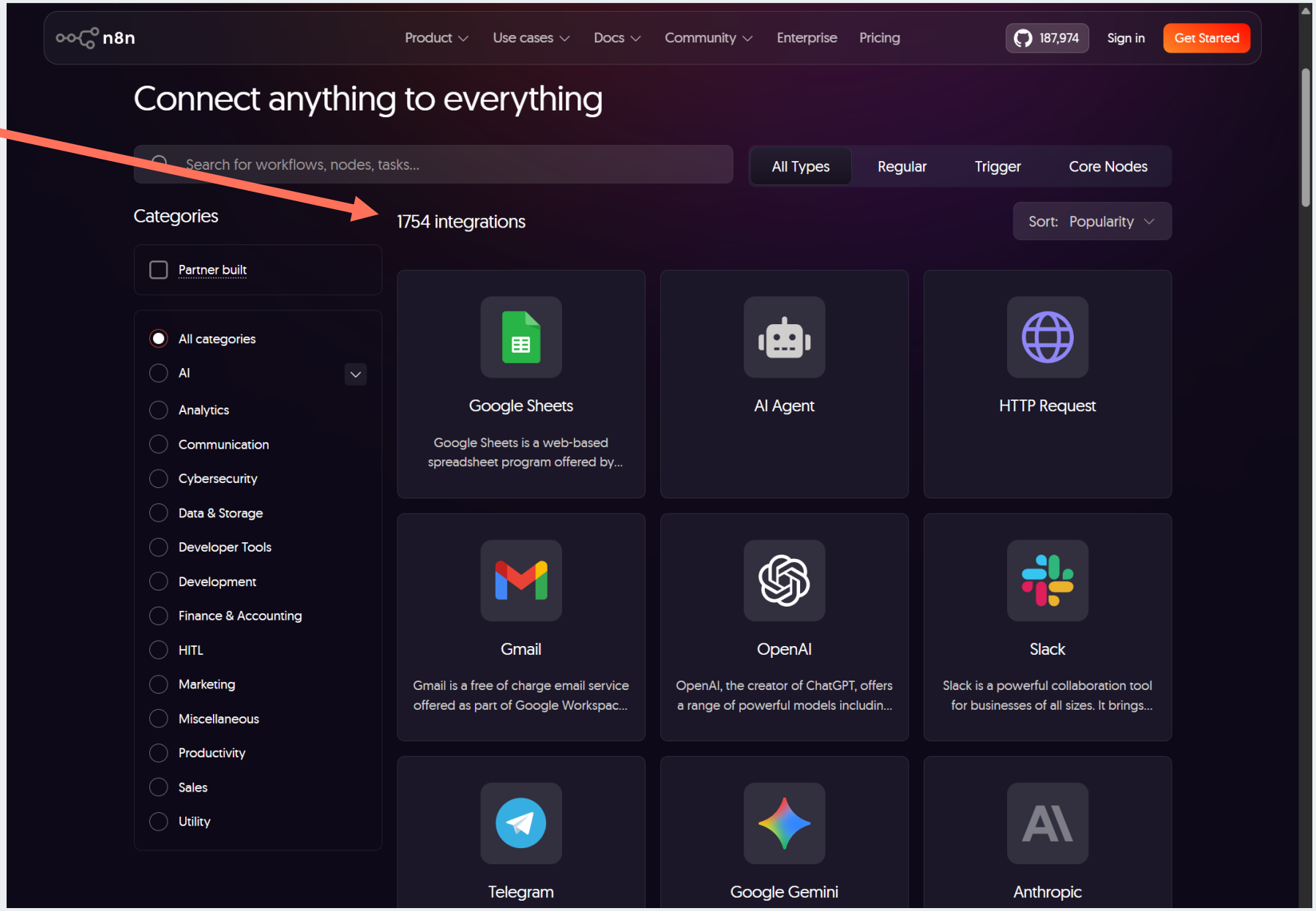




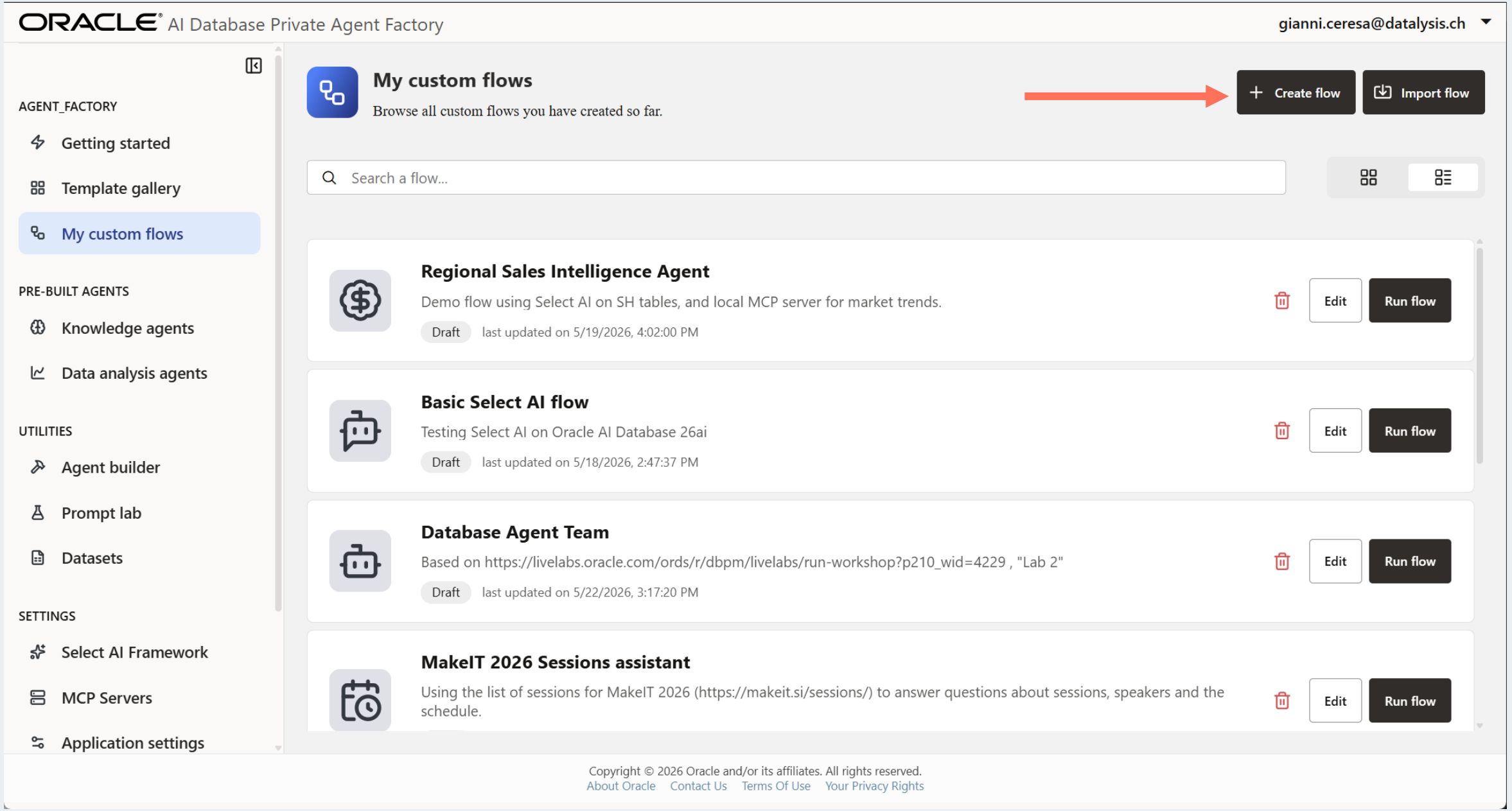




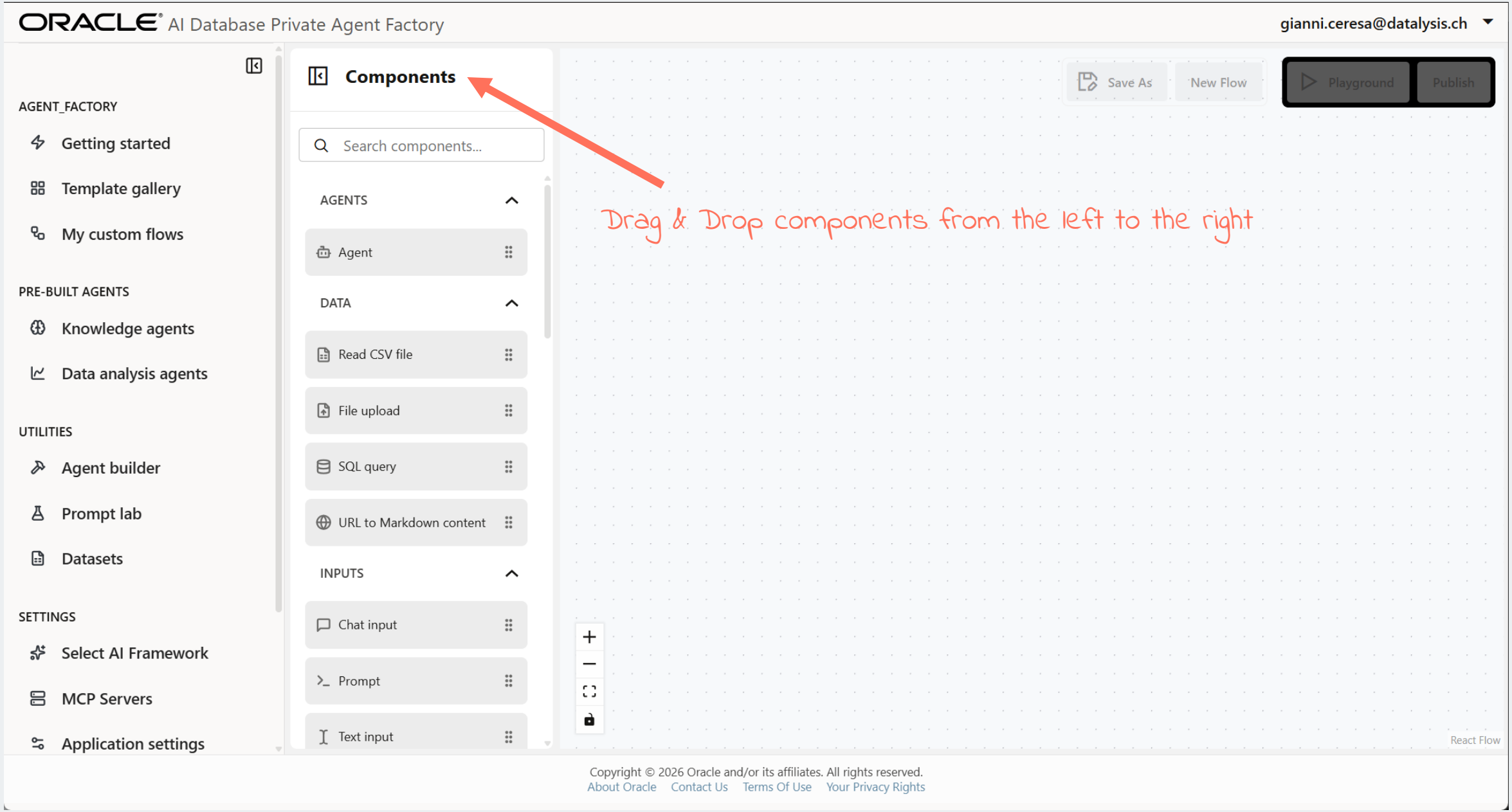
PAF vs n8n: I don't see PAF as a n8n competitor...



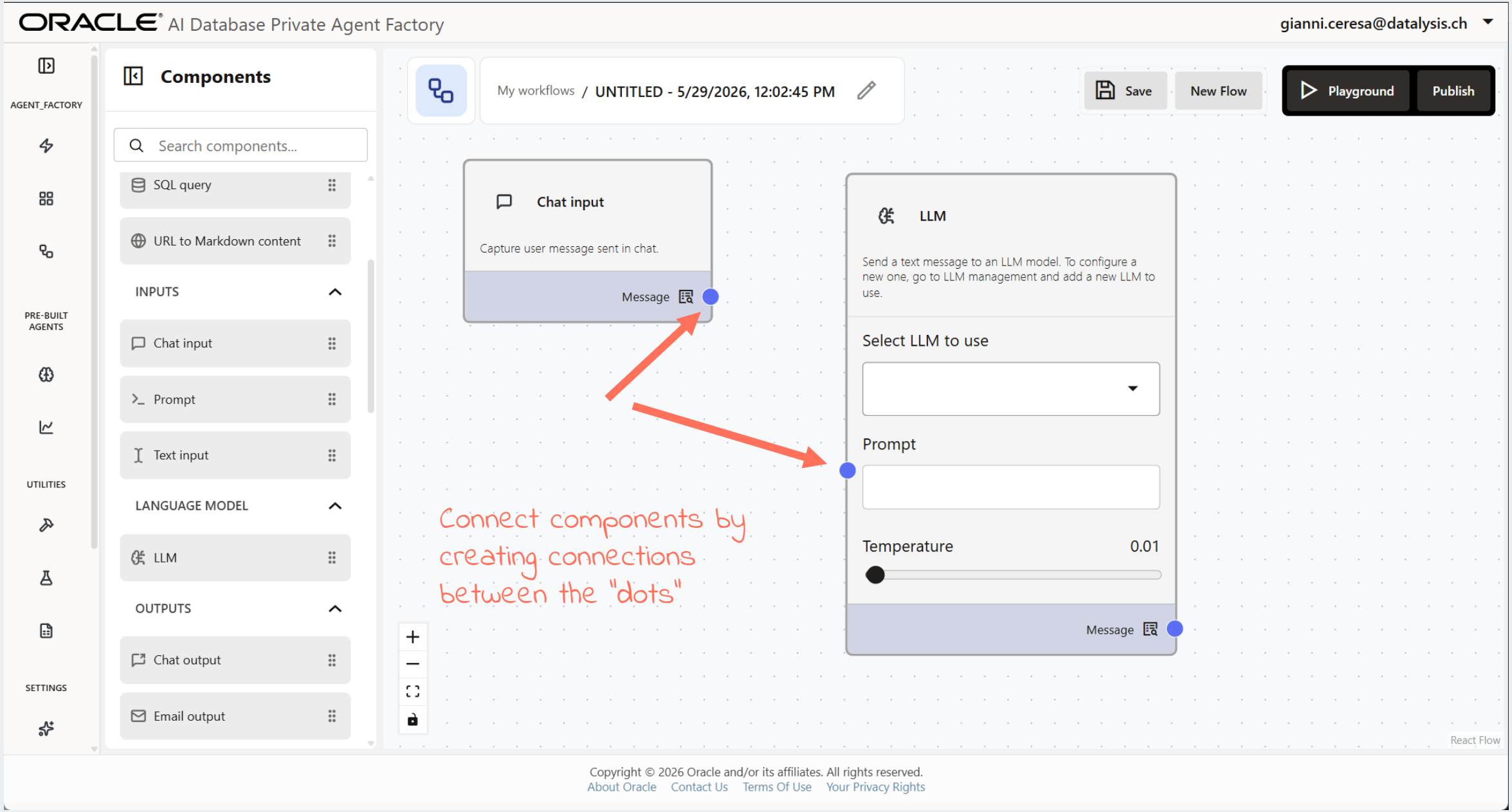
Agents/workflows: Create flow



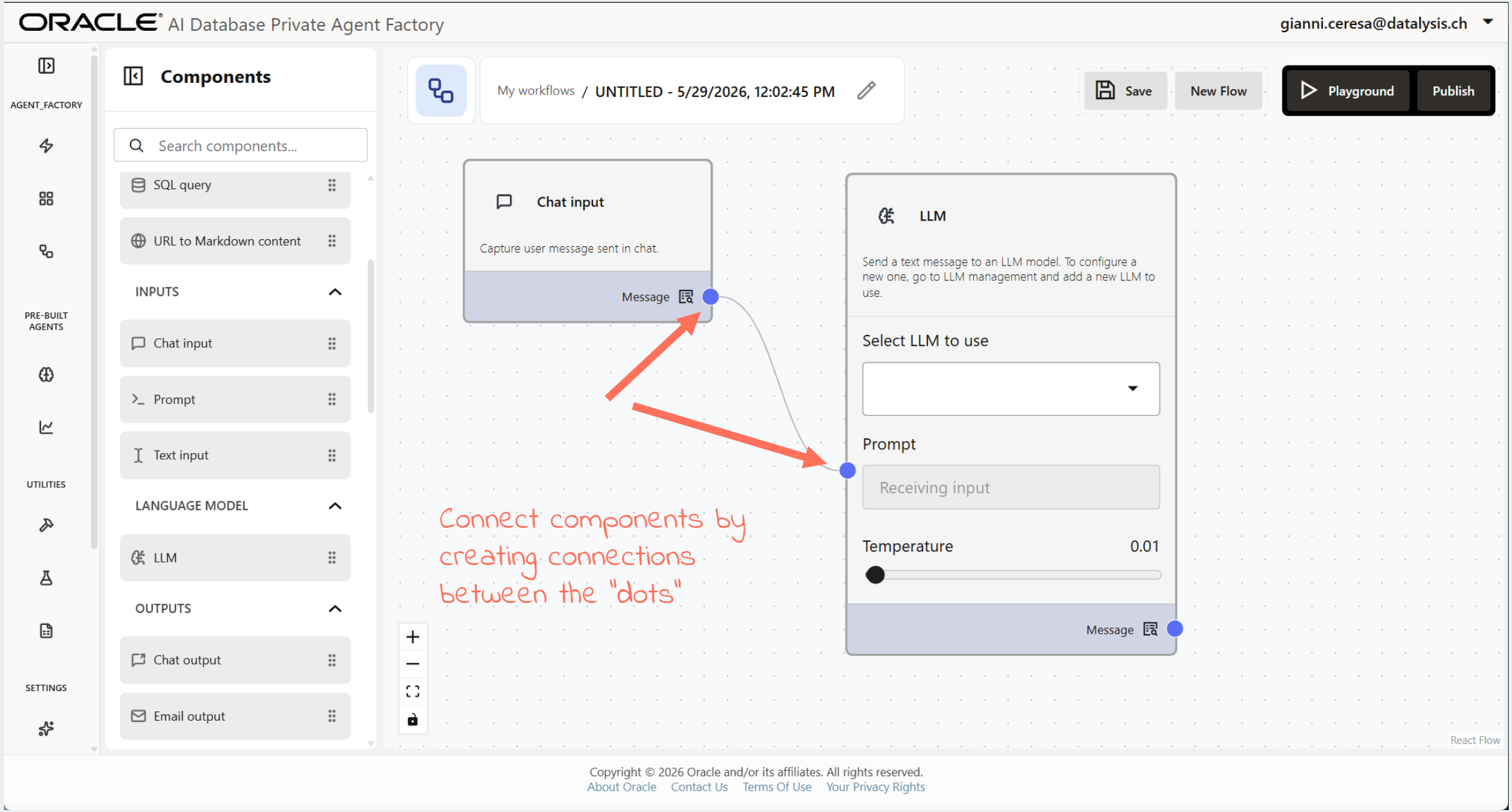
Agents/workflows: Create flow



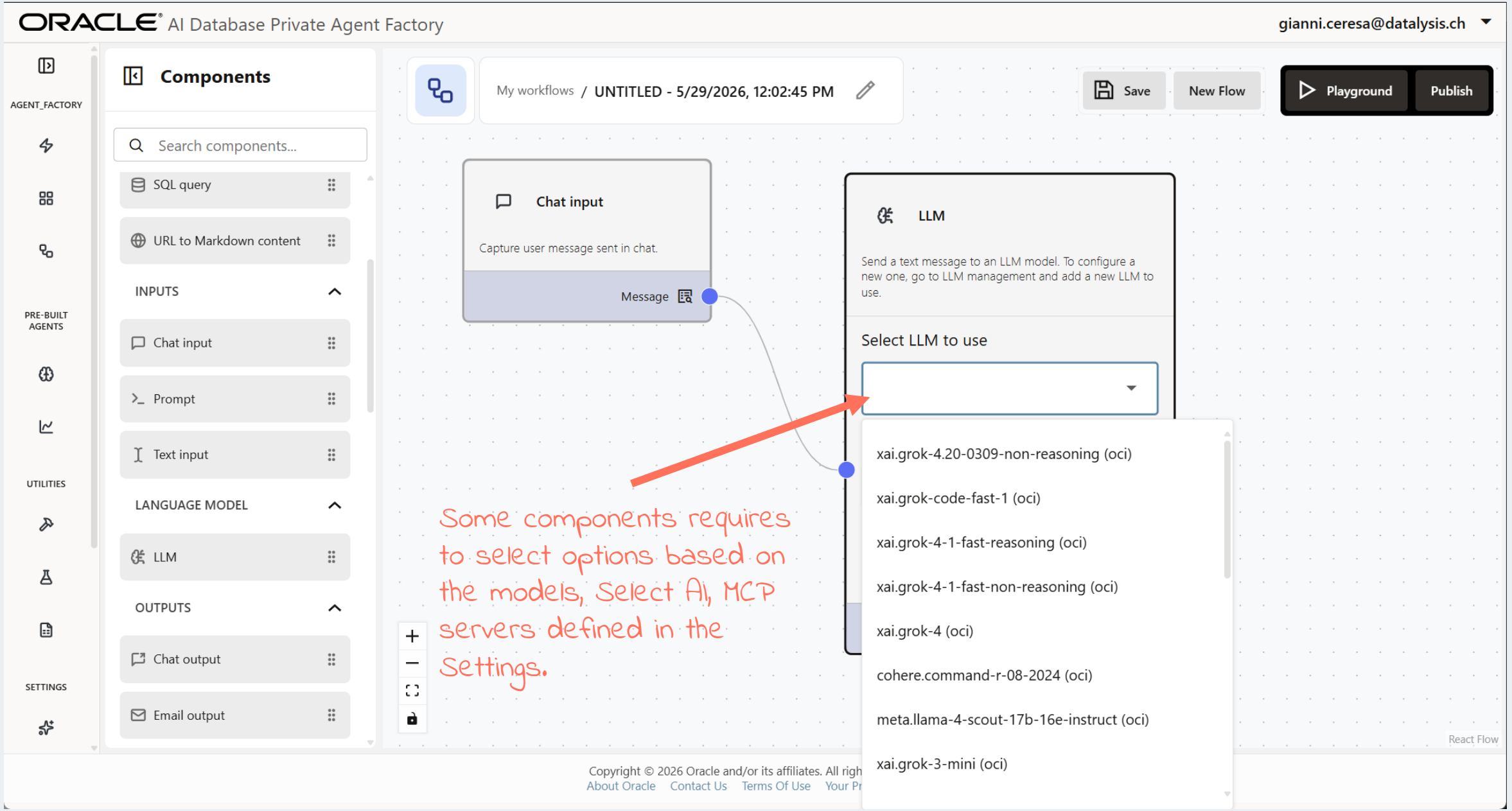
Agents/workflows: Create flow



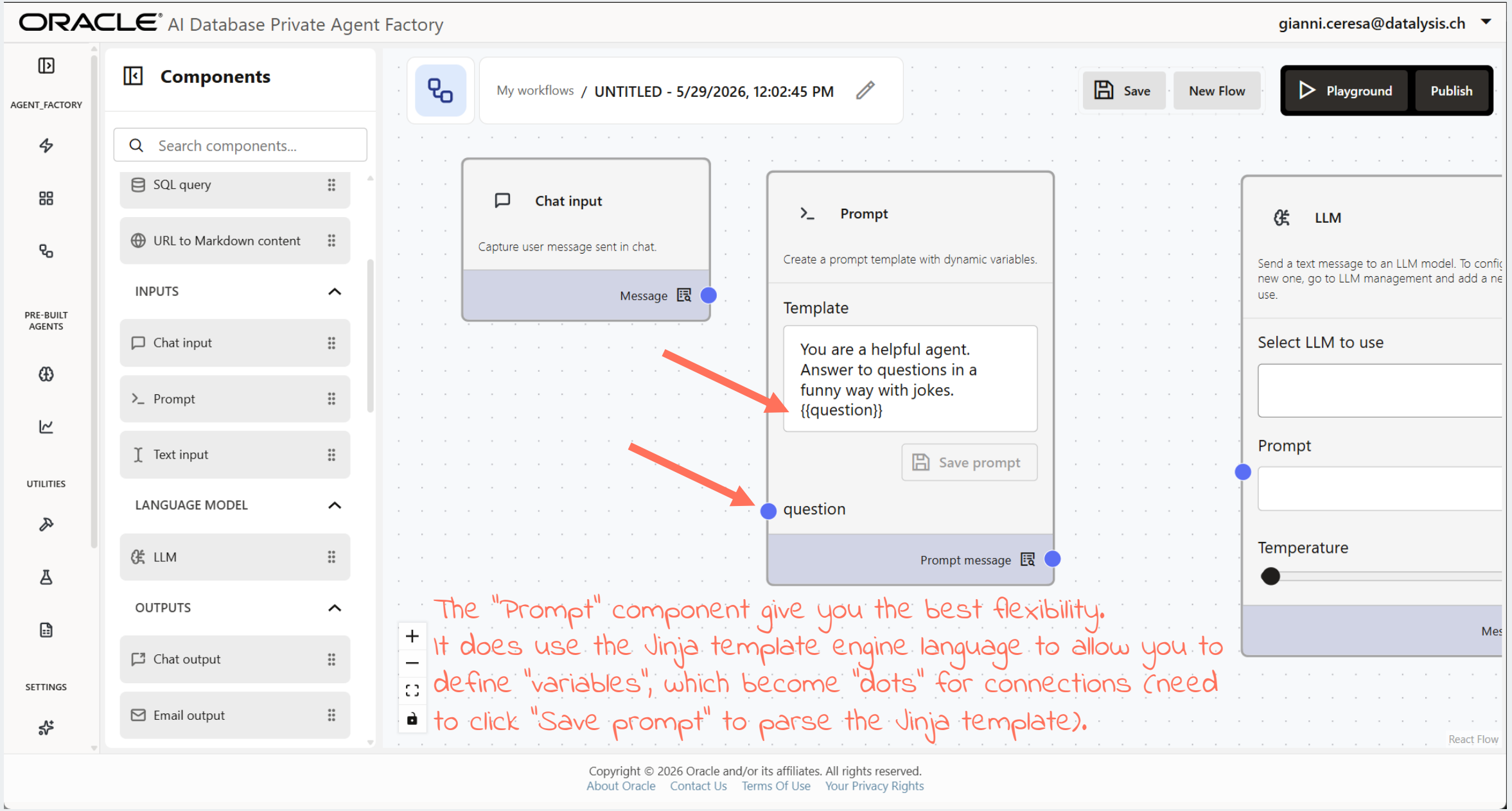
Agents/workflows: Create flow



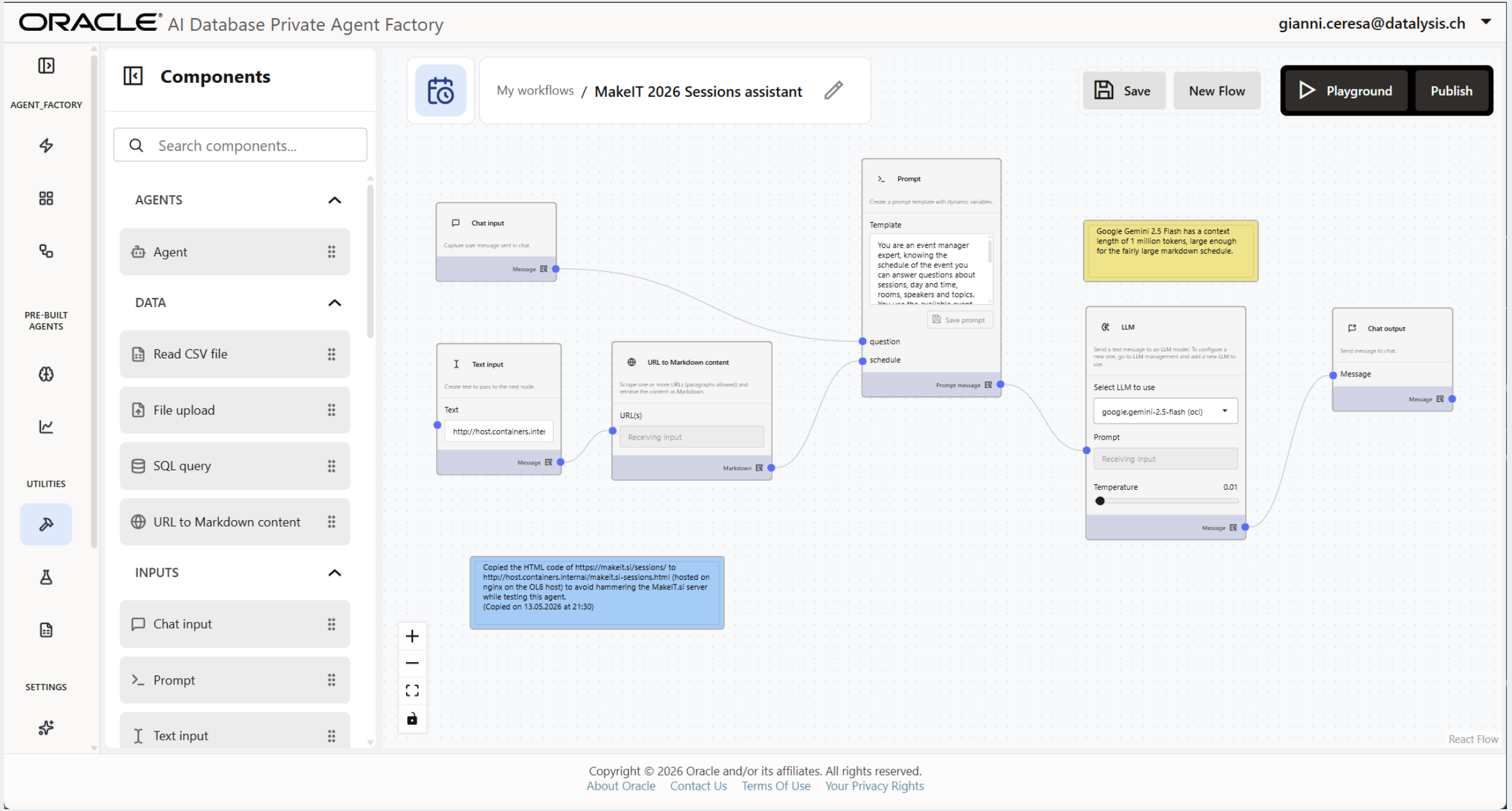
Agents/workflows: Create flow



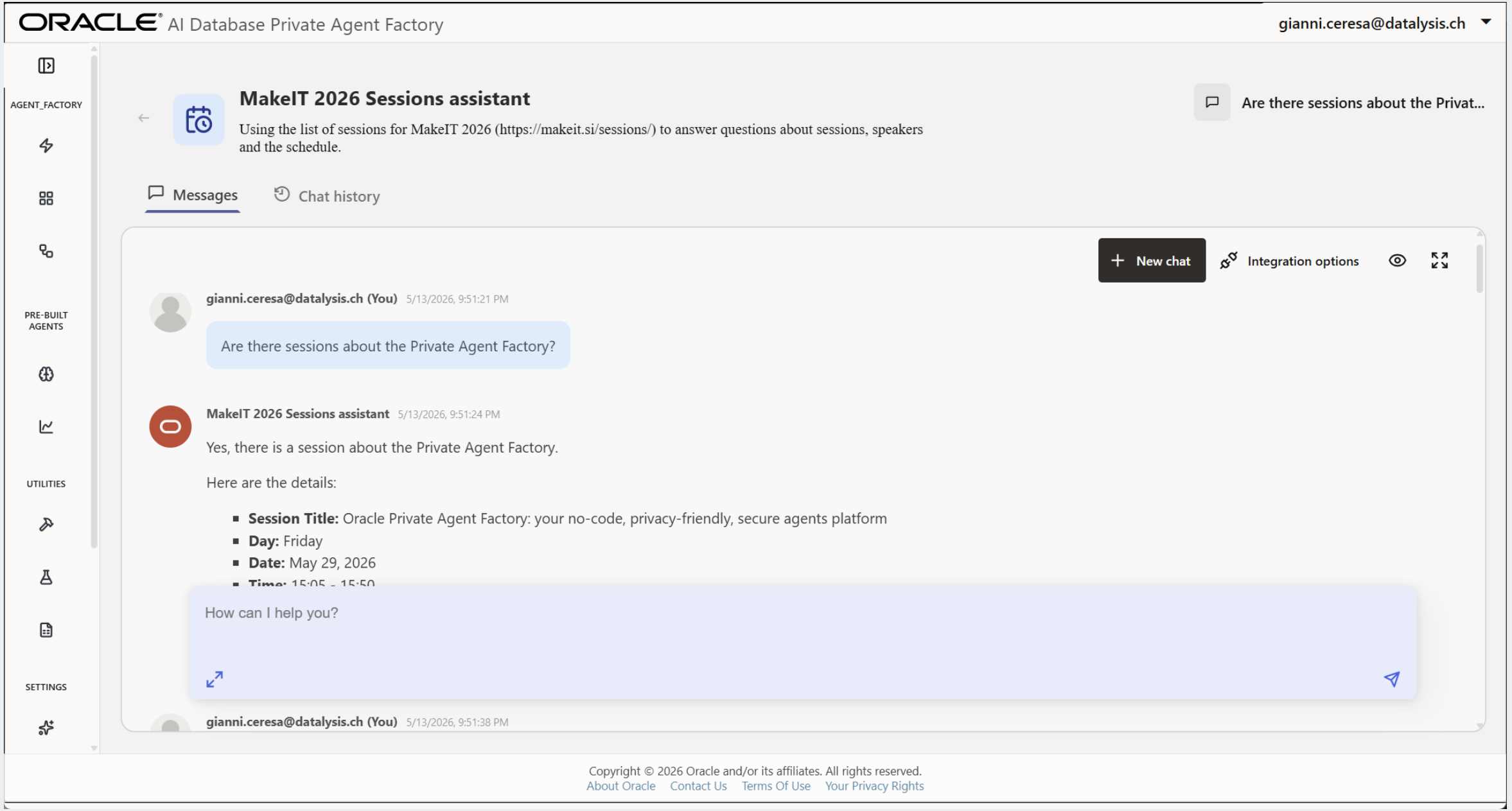
Agents/workflows: Create flow



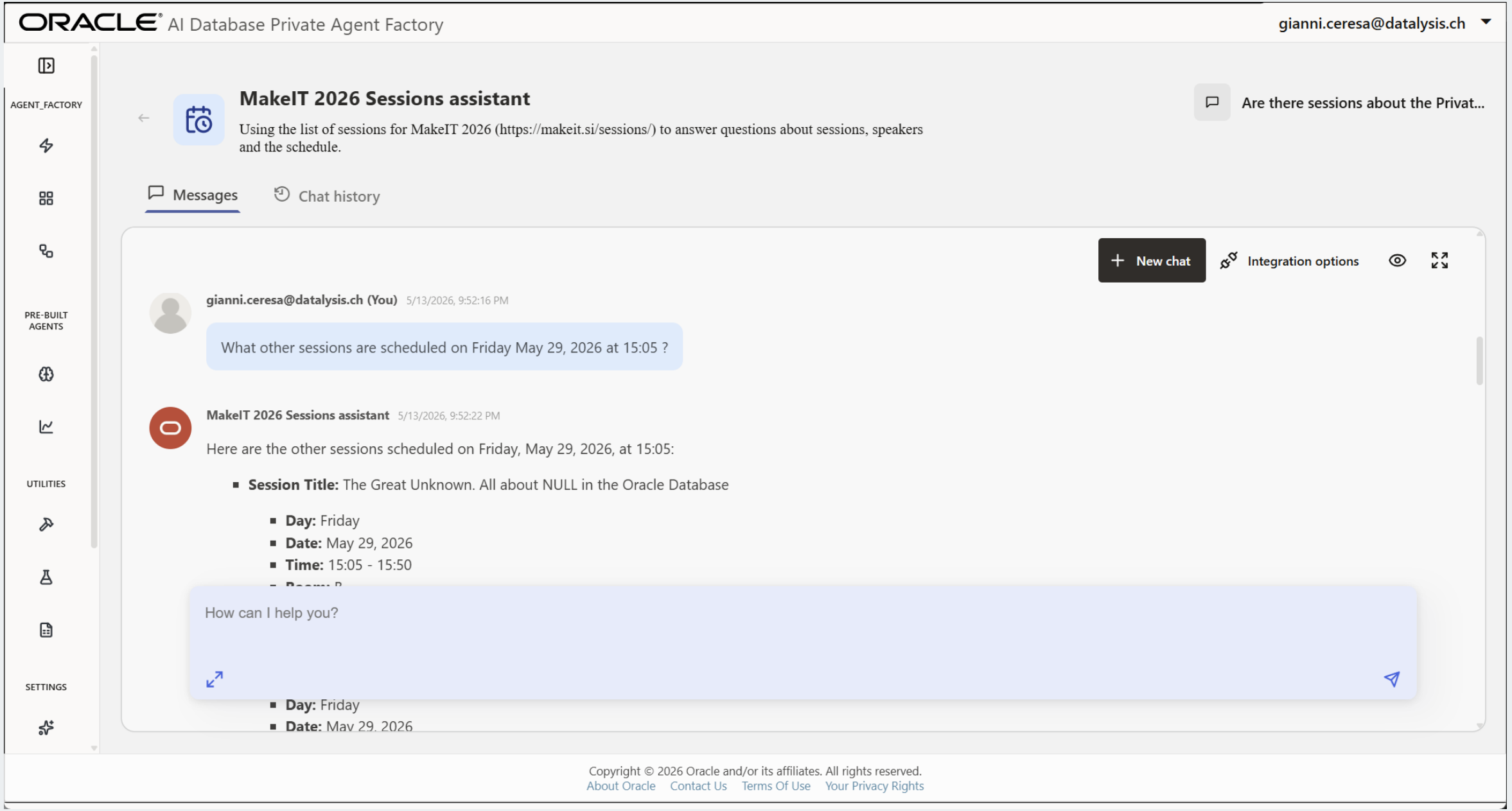
A MakeIT 2026 agent



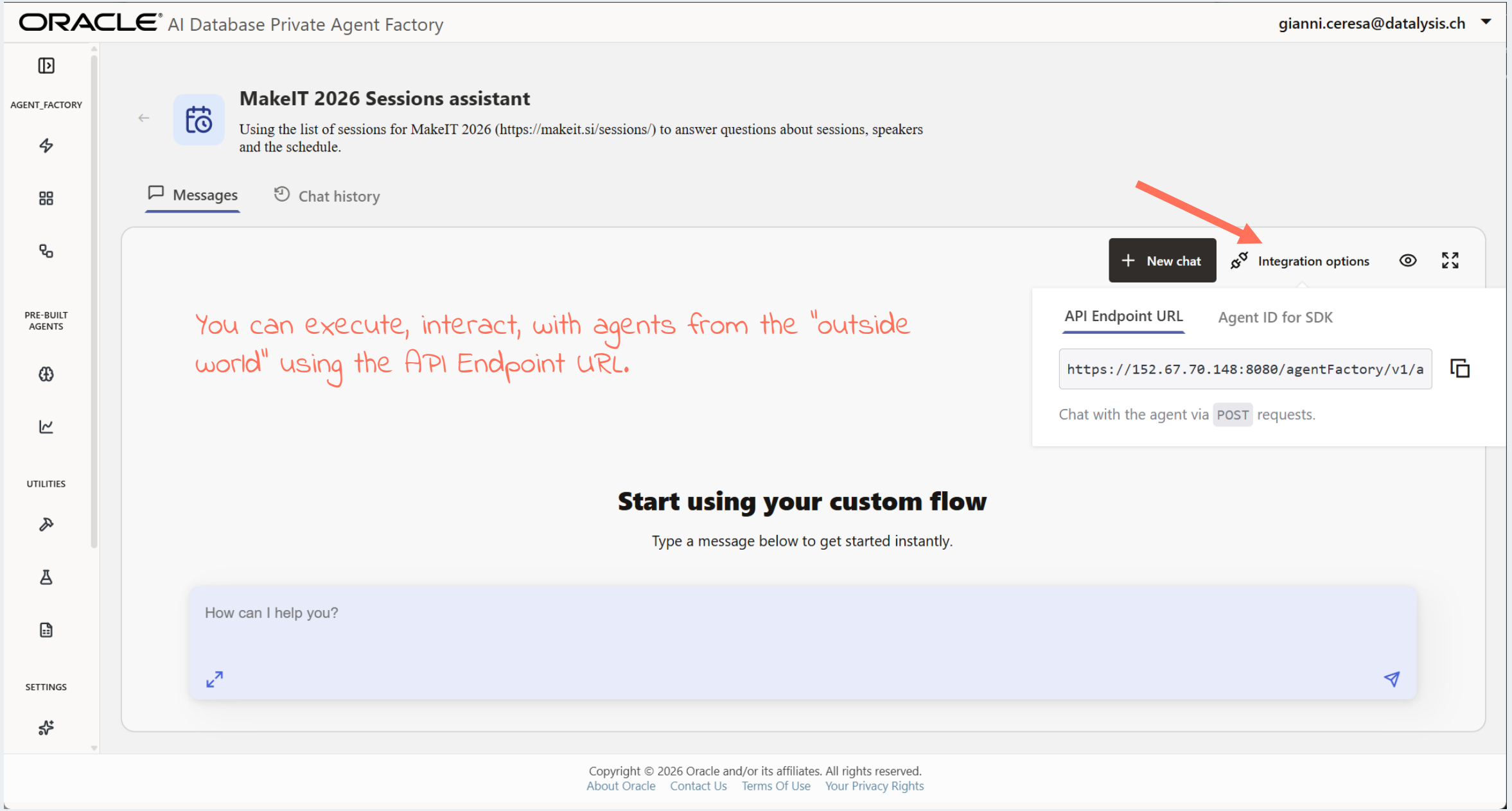
A MakeIT 2026 agent



A MakeIT 2026 agent



A MakeIT 2026 agent



A MakeIT 2026 agent

A quick example in Python:

- Authentication (because no SSO configured)
- Send the first message
- Get the "roomId" to continue the chat
- Send other messages

test_paf_agent.py > ...

```python
import requests
import urllib3
import json

urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

paf_url = "https://152.67.70.148:8080"
published_flow_id = "51B57FD8A5211124E0635401000ABCC8"

# authentication is done via cookies, so we start a session
session = requests.Session()

# part 1 - get cookie using basic auth (b64 encode as user:pwd)
login_url = paf_url+"/agentFactory/v1/loginValidation"

headers = {
  'Authorization': 'Basic [redacted]='
}

login_response = session.get(login_url, headers=headers, verify=False)

print('login status code:',login_response.status_code)

# part 2 - send message to agent
agent_url = paf_url+'/agentFactory/v1/agentBuilder/run/'+published_flow_id

msg = "Are there sessions about the Private Agent Factory?"
payload = {"message":msg}
agent_headers = {'Content-Type': 'application/json'}
agent_response = session.post(agent_url,headers=agent_headers, json=payload, verify=False)
#print("Agent response:", agent_response.text)
data_bytes = agent_response.content
data_str = data_bytes.decode('utf-8')
json_objects = json.loads(data_str)
print("Agent response:", json_objects['message'])

# part 3 - send subsequent messages using room
room_id = json_objects['roomId']

msg = "What other sessions are scheduled on Friday May 29, 2026 at 15:05?"

payload = {"message":msg,"room":room_id}
agent_headers = {'Content-Type': 'application/json'}
agent_response = session.post(agent_url,headers=agent_headers, json=payload, verify=False)
data_bytes = agent_response.content
data_str = data_bytes.decode('utf-8')
json_objects = json.loads(data_str)
print("Agent response:", json_objects['message'])
```

A MakeIT 2026 agent

Command Prompt | gianni@THINKPADX1C13-GC:~

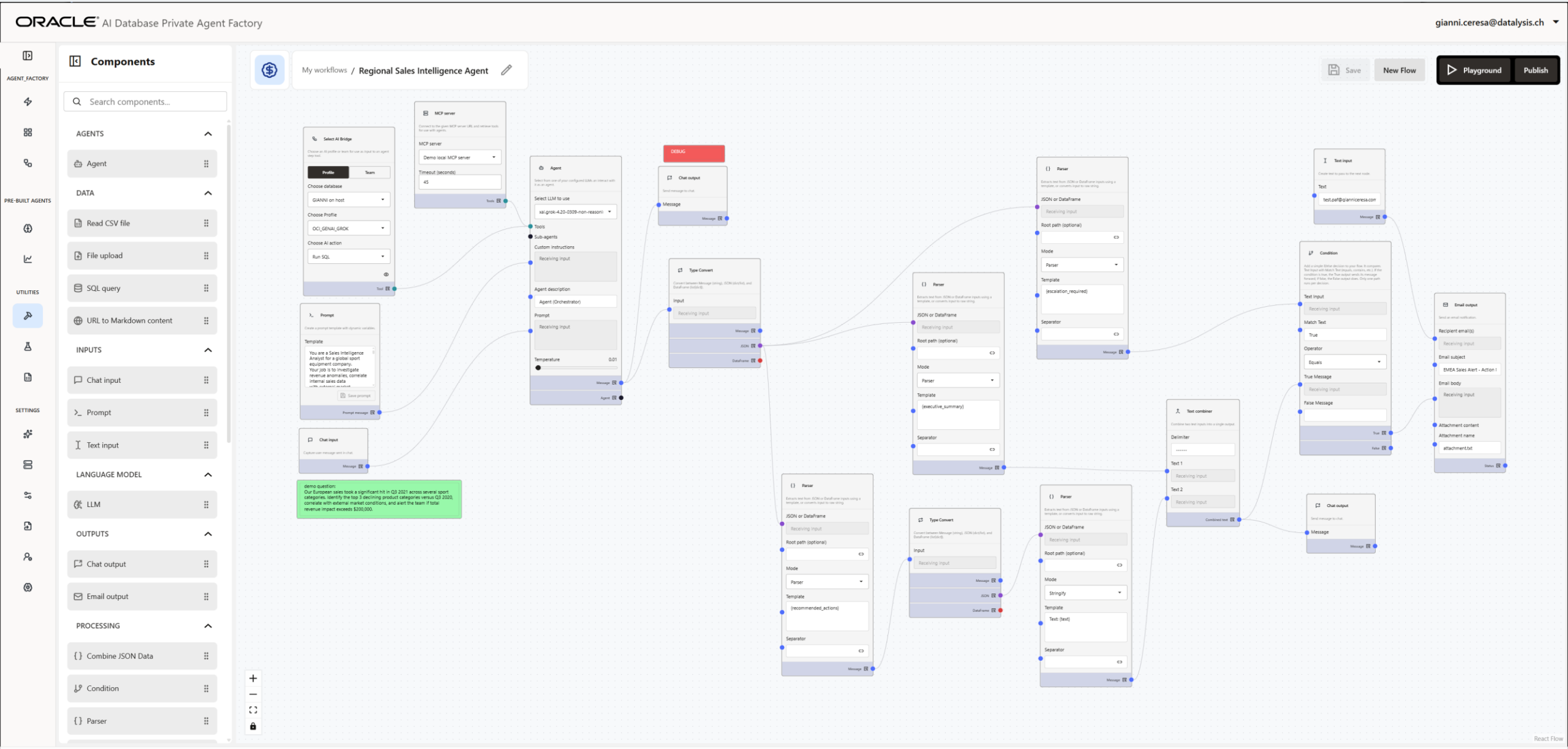
```
(py313langchain) [gianni@THINKPADX1C13-GC ~]$ python test_paf_agent.py
login status code: 200
Gianni: Are there sessions about the Private Agent Factory?
Agent response: Yes, there is a session about the Private Agent Factory.

Here are the details:

*   **Session Title:** Oracle Private Agent Factory: your no-code, privacy-friendly, secure agents platform
*   **Day:** Friday
*   **Date:** May 29, 2026
*   **Time:** 15:05 - 15:50
*   **Room:** A
*   **Speaker:** Gianni Ceresa
*   **Conference:** MakeIT Oracle 2026
Gianni: What other sessions are scheduled on Friday May 29, 2026 at 15:05?
Agent response: Here are the other sessions scheduled on Friday May 29, 2026 at 15:05:

*   **Room B:**
    *   **Title:** The Great Unknown. All about NULL in the Oracle Database
    *   **Speaker:** Patrick Jolliffe
*   **Room A:**
    *   **Title:** Oracle Private Agent Factory: your no-code, privacy-friendly, secure agents platform
    *   **Speaker:** Gianni Ceresa
*   **Room C:**
    *   **Title:** The Bank Crash Chronicles: Why Our 11TB Oracle Exadata Architecture Failed
    *   **Speaker:** Martin Ladecký
*   **Room D:**
    *   **Title:** Infusing Java with AI: 5 Tips and Strategies for GenAI-Powered Applications
    *   **Speaker:** Rustam Mehmandarov
*   **Room E:**
    *   **Title:** Stop Refreshing the Dashboard: Real-Time Reporting with CDC and Apache Flink
    *   **Speaker:** Dominik Žnidaršič
*   **Room F:**
    *   **Title:** Rebuilding the Plane Mid-Flight: Tackling Massive Tech Debt Without Slowing Down
    *   **Speaker:** Taimoor Alam
(py313langchain) [gianni@THINKPADX1C13-GC ~]$
```

A “simple” sample workflow...



A “simple” sample flow...

Data type conversions can easily be many nodes in your workflow...

- From text to JSON, extracting values by key from a JSON object (safer than full text pattern matching on a text), back to text etc.

Simple TRUE / FALSE conditions, text formatting, concatenation can also become many nodes in a workflow.

In this sample workflow there is a single agent node (connected to a MCP server and a Select AI Bridge), the rest is mostly text manipulation and transformation.

A “simple” sample flow…

Scenario:

- Based on the Sales History (SH) sample schema (https://github.com/oracle-samples/db-sample-schemas/tree/main/sales_history)

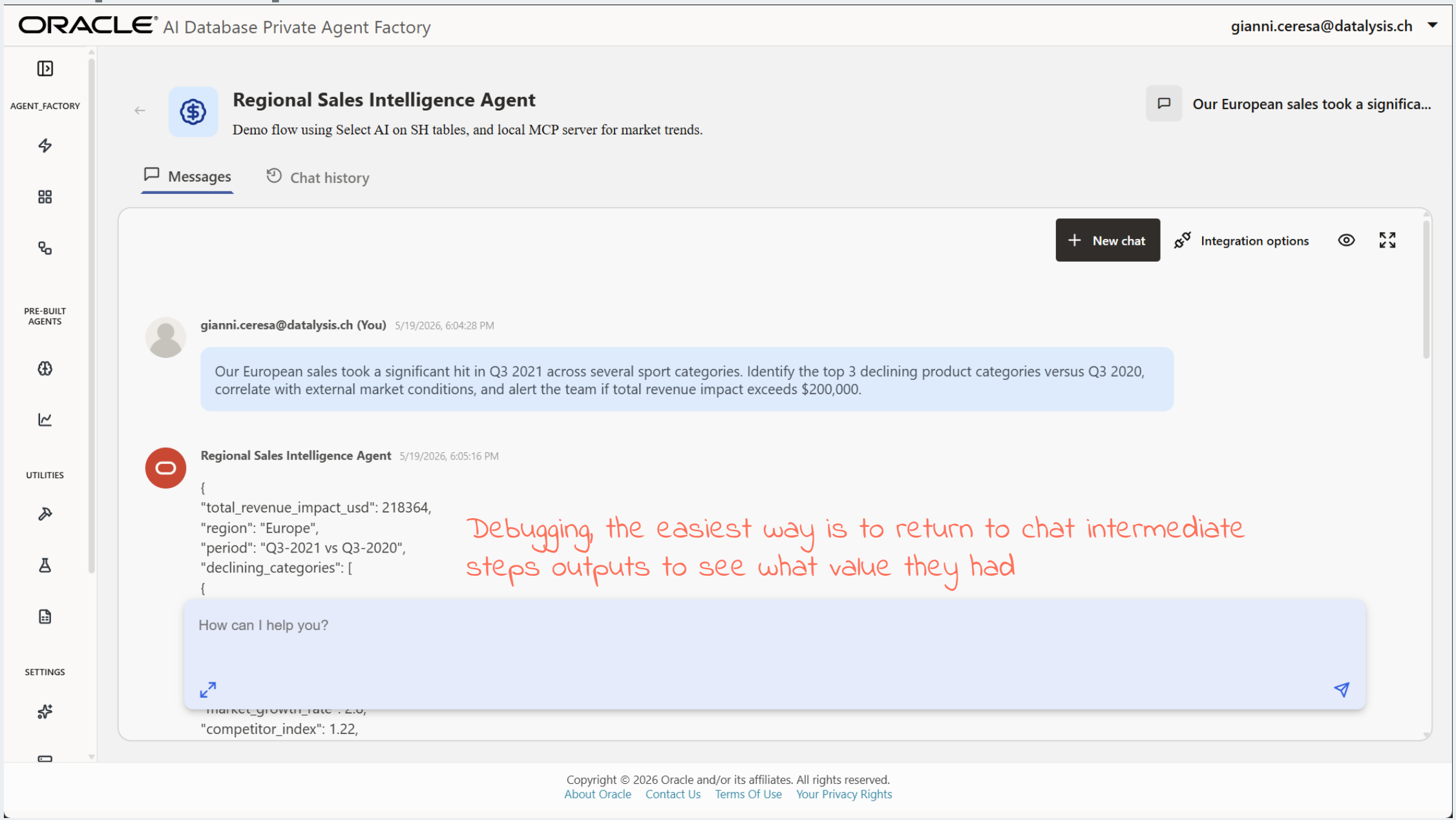
- The VP of Sales saw that in Q3 2021 for Europe the numbers aren’t as great as expected, therefore he does ask the agent:

  “Our European sales took a significant hit in Q3 2021 across several sport categories. Identify the top 3 declining product categories versus Q3 2020, correlate with external market conditions, and alert the team if total revenue impact exceeds $200,000.”

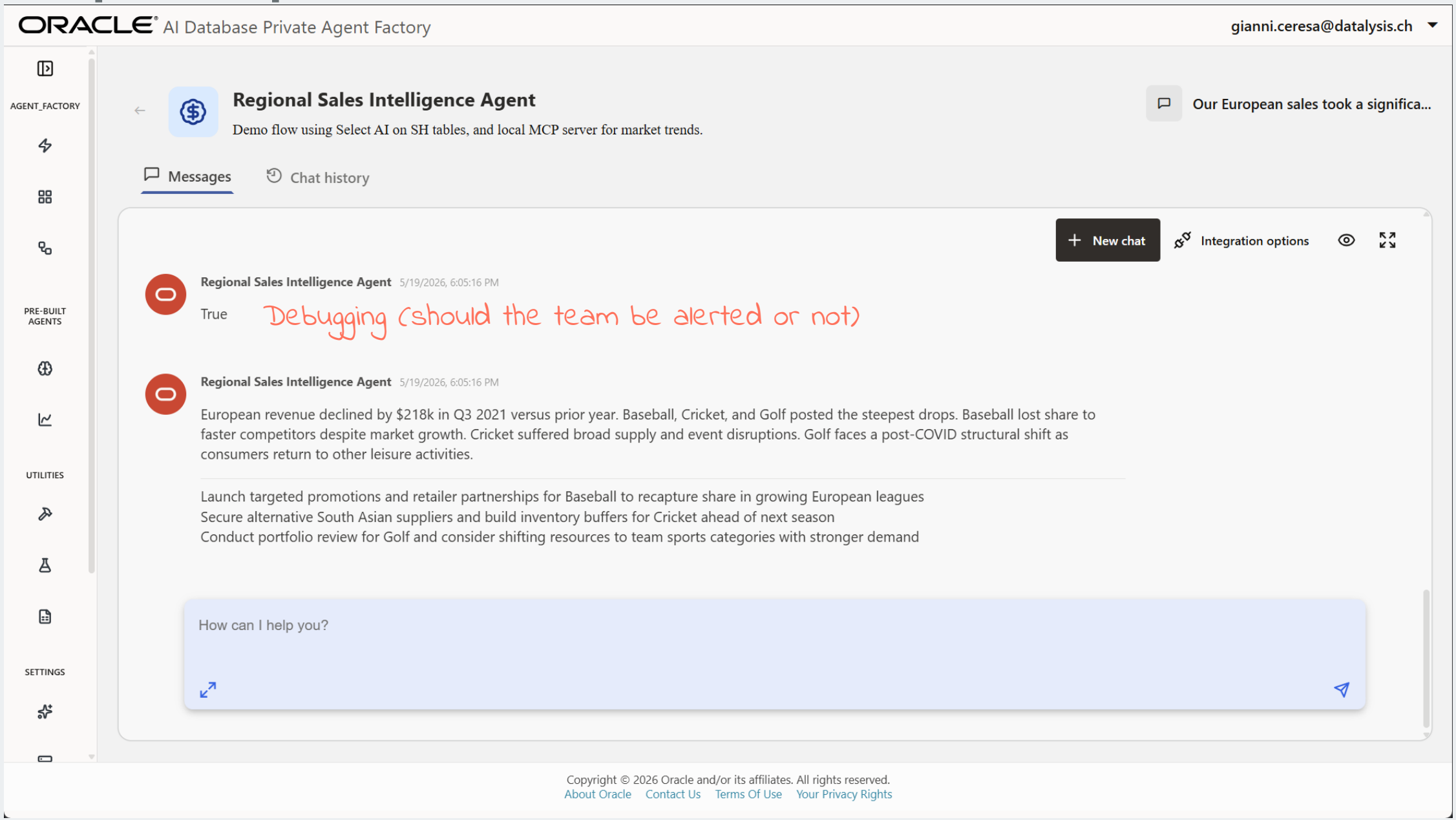
- The agent has access to Select AI to ask questions to the database in natural language and let Select AI translate to SQL, execute the query, and return the data.

- The agent has access to a MCP server
  - A a calculation tool (LLM aren’t always great with math!)
  - A market intelligence platform having information on the market situation.
    - list_market_categories : the list of supported product categories
    - get_market_intel : the external market condition for a category of product, a region and a quarter-year

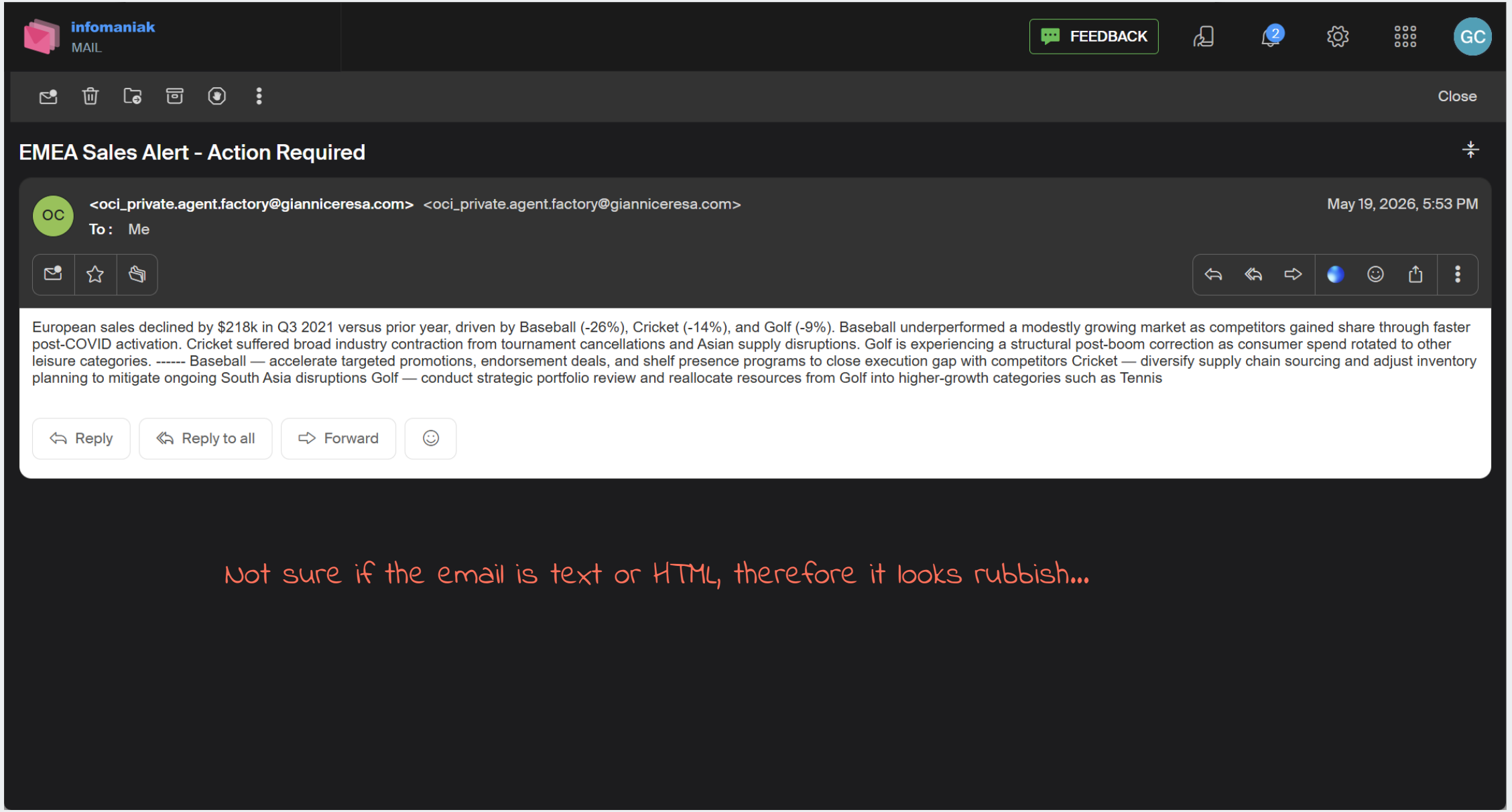
A “simple” sample flow...



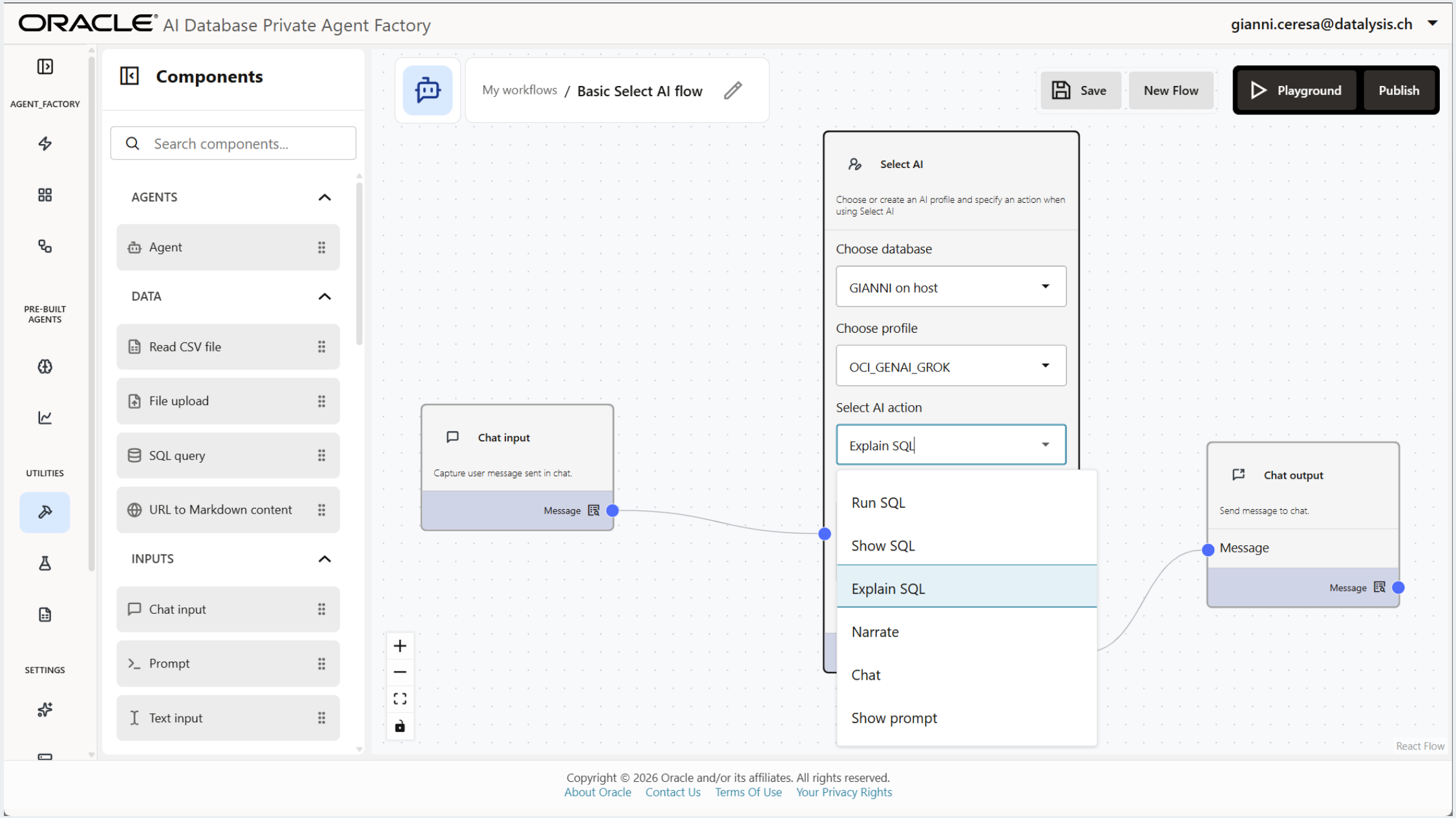
A “simple” sample flow...



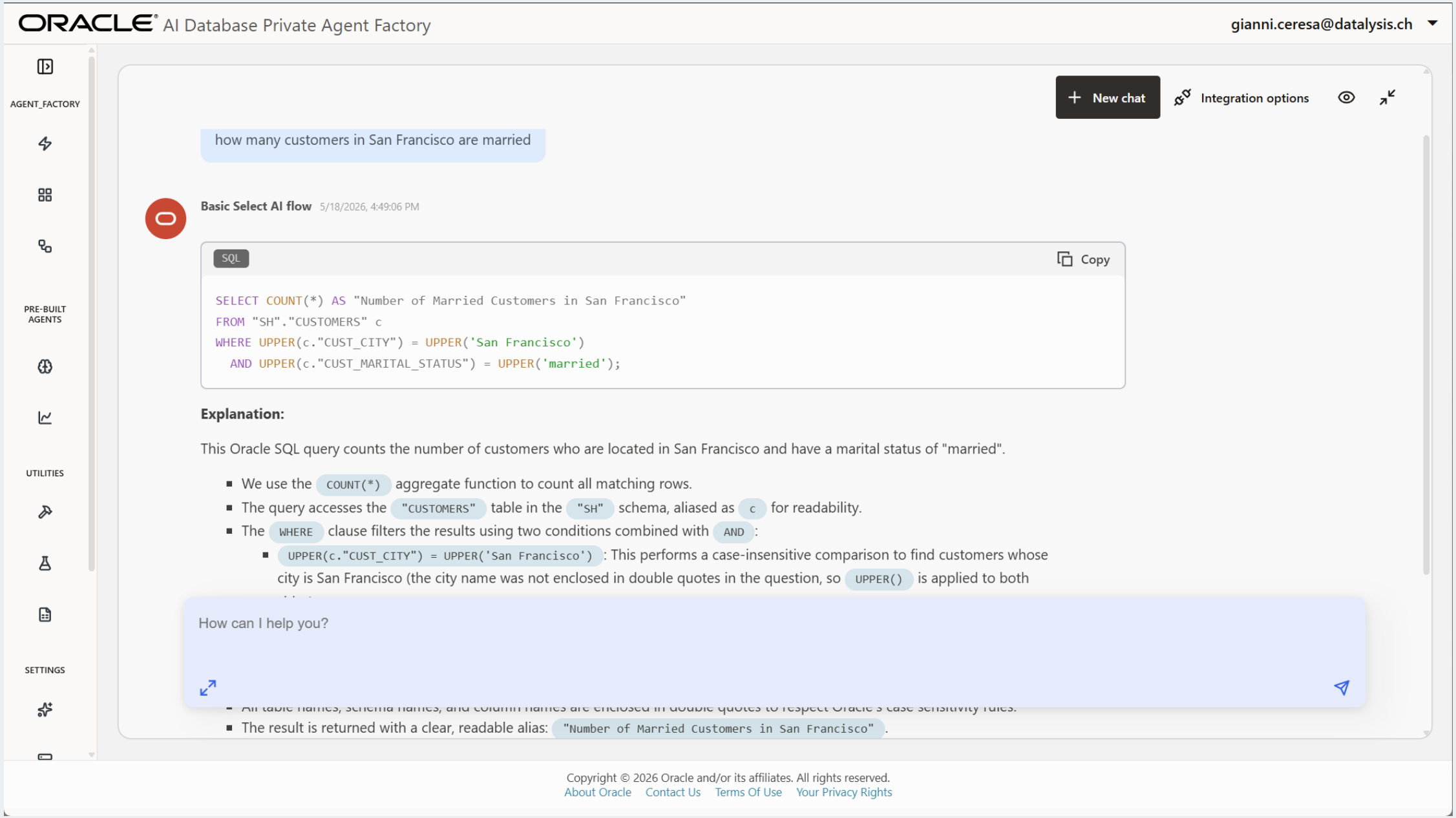
A “simple” sample flow...



Database Select AI



Database Select AI



Database Select AI Agent Team

Based on https://livelabs.oracle.com/ords/r/dbpm/livelabs/view-workshop?wid=4229 for the database objects.

```
2   select * from CUSTOMERS where CUSTOMER_ID = 2;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SCRIPT OUTPUT SQL HISTORY TASK MONITOR

All rows fetched: 1 in 0.052 seconds

	CUSTOMER_ID	NAME	EMAIL	PHONE	STATE	ZIP
1	2	Bob Martinez	bob.martinez@example.com	555-2345	CA	94105

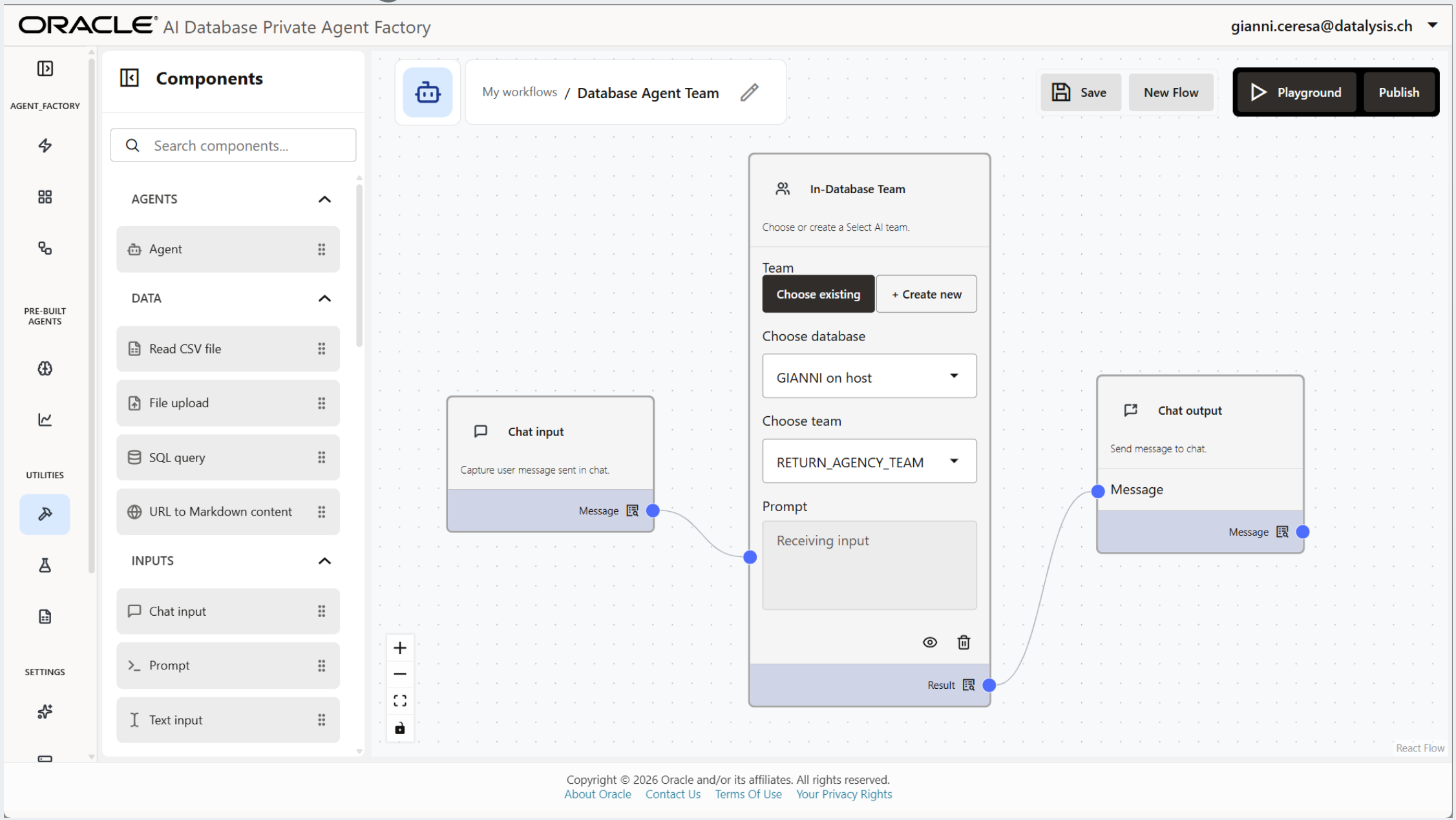
```
2   select * from CUSTOMER_ORDER_STATUS where ORDER_NUMBER = 7820;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SCRIPT OUTPUT SQL HISTORY

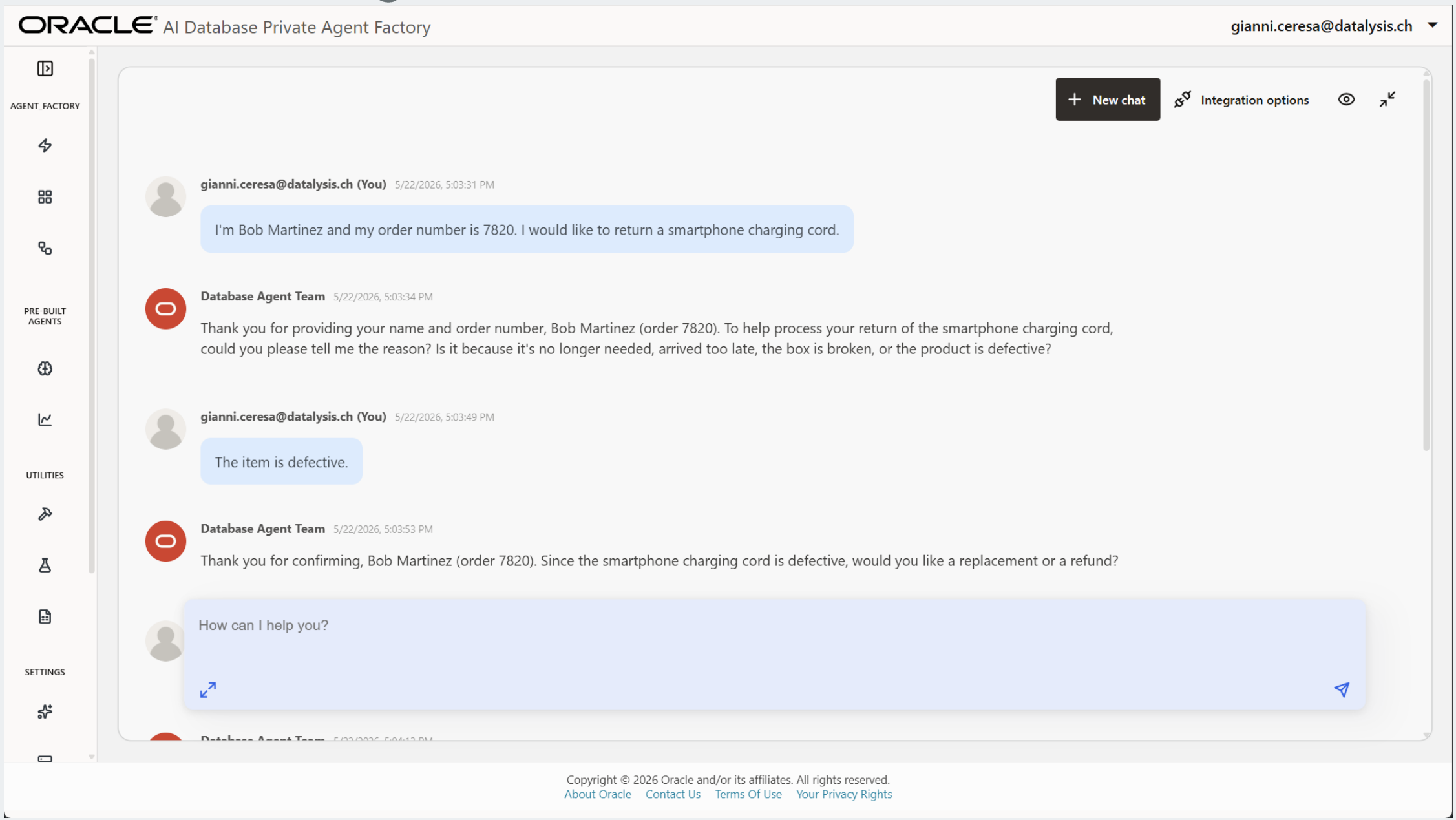
All rows fetched: 1 in 0.040 seconds

	CUSTOMER_ID	ORDER_NUMBER	STATUS	PRODUCT_NAME
1	2	7820	delivered	smartphone charging cord

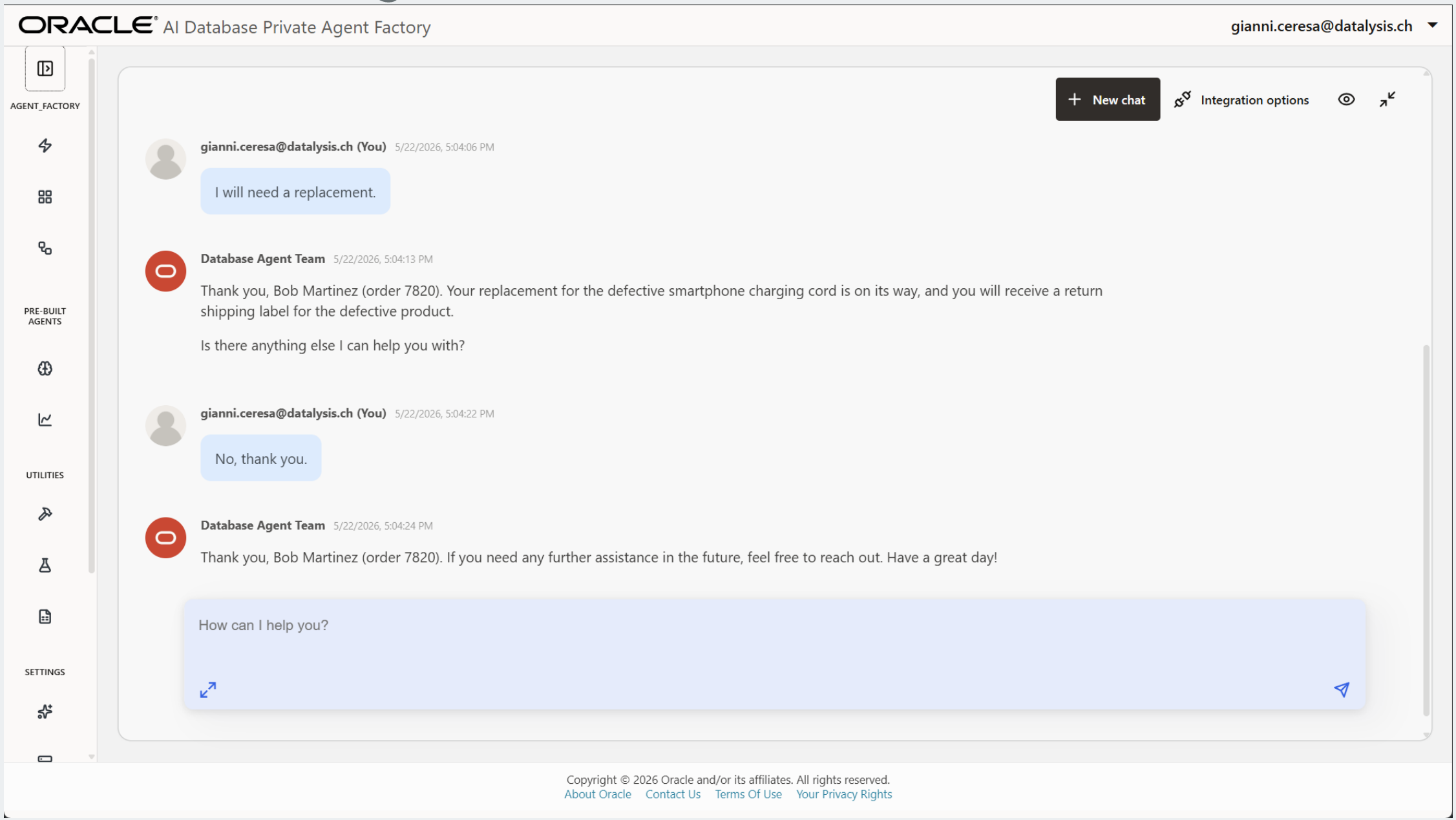
Database Select AI Agent Team



Database Select AI Agent Team



Database Select AI Agent Team



Database Select AI Agent Team

```
2   select * from CUSTOMER_ORDER_STATUS where ORDER_NUMBER = 7820;
```

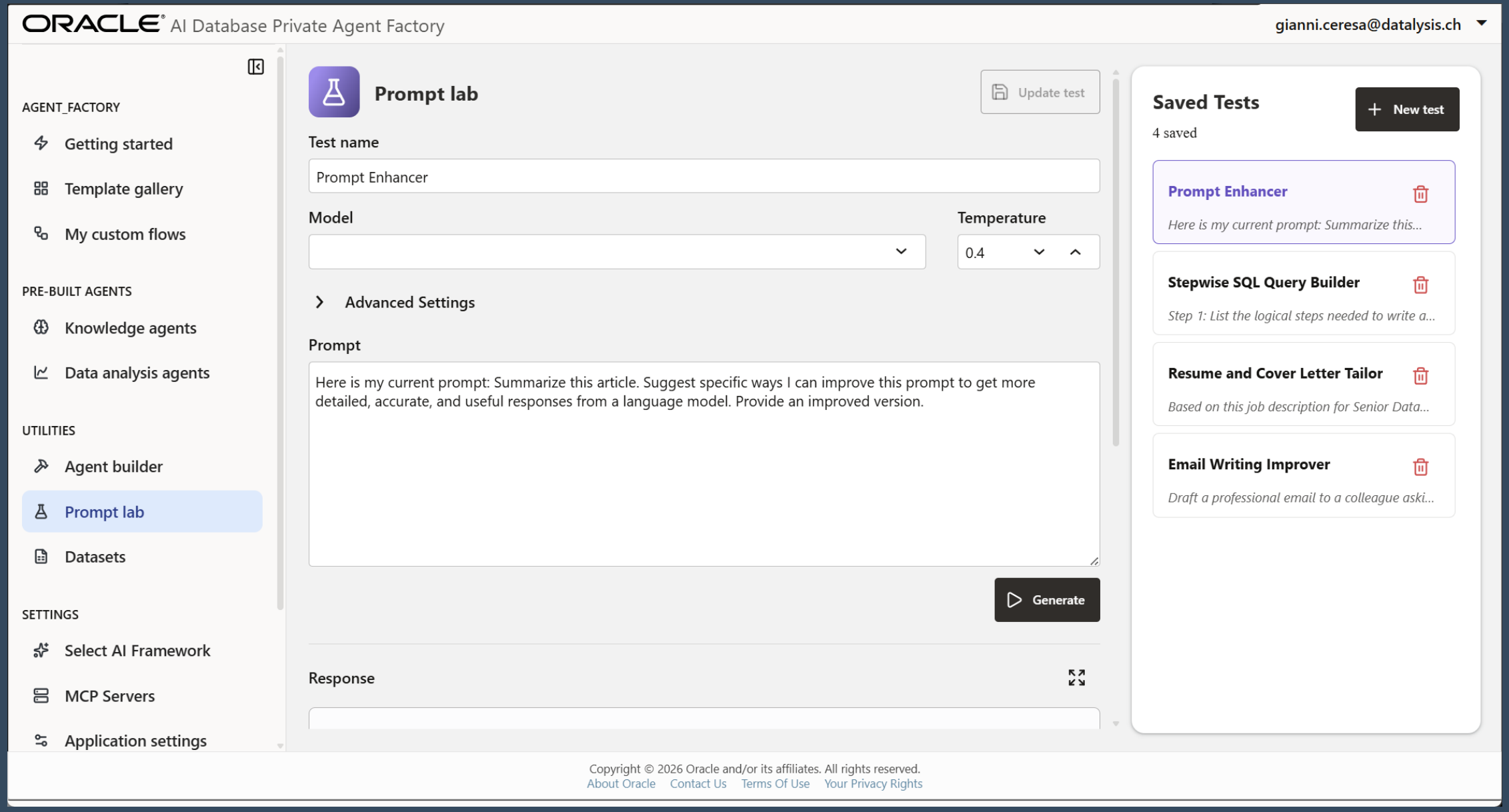
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SCRIPT OUTPUT SQL HISTORY

All rows fetched: 1 in 0.047 seconds

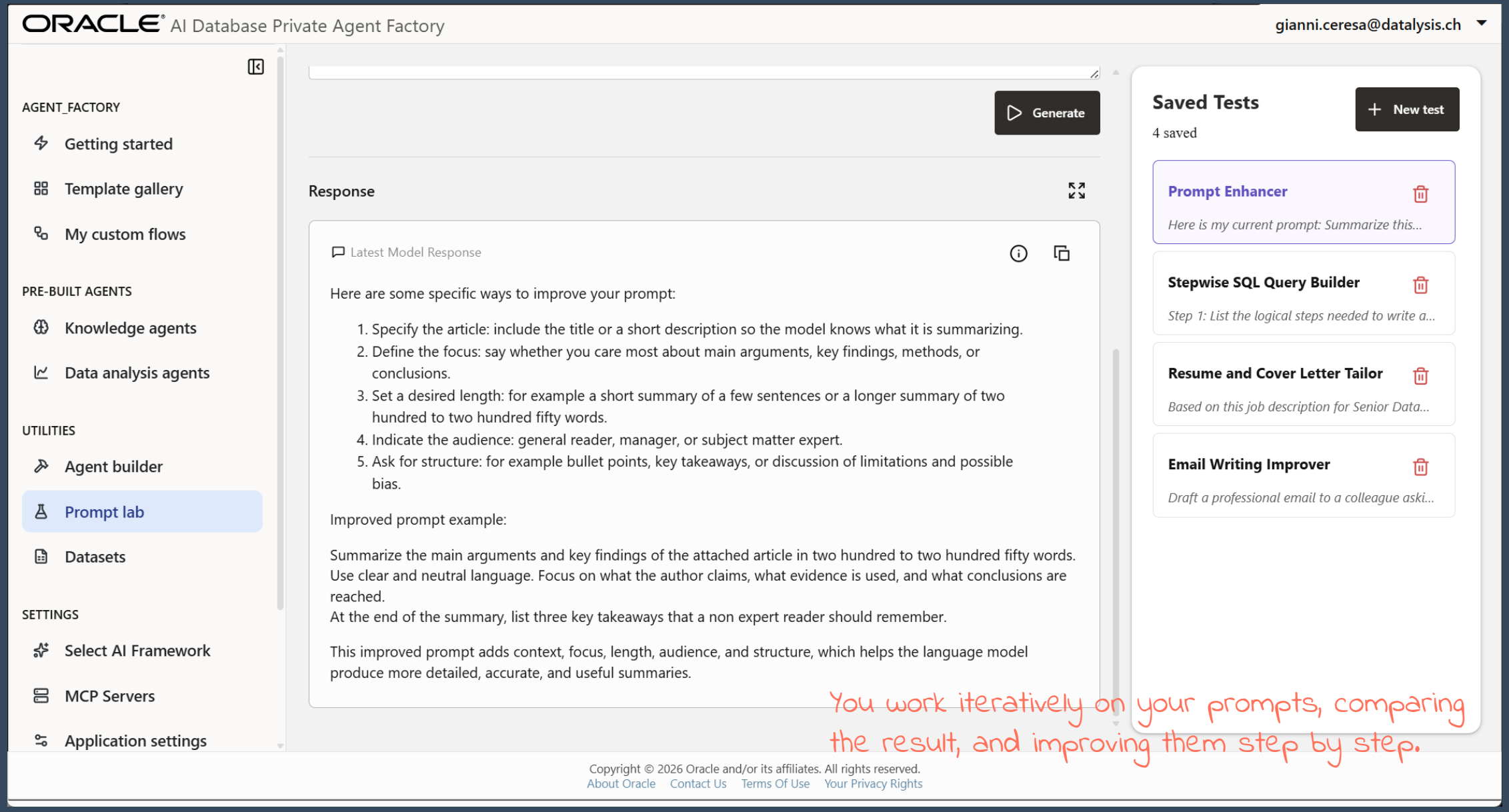
	CUSTOMER_ID	ORDER_NUMBER	STATUS	PRODUCT_NAME
1	2	7820	replaced	smartphone charging cord



The Prompt lab utility



The Prompt lab utility

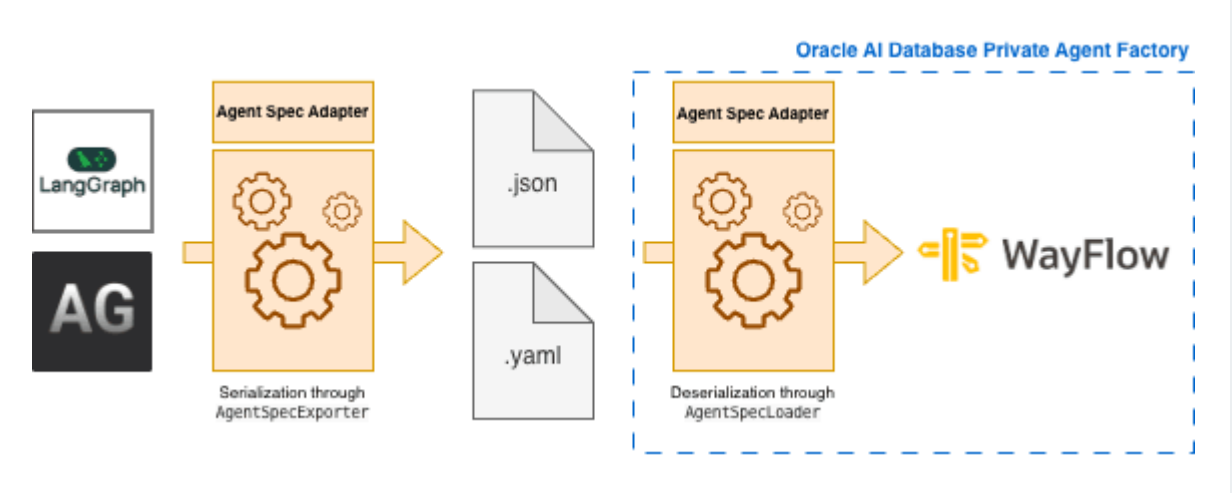


Import agents from external frameworks

Because of the heavy fragmentation of the market, importing from other frameworks and products is challenging.

Oracle create the “Open Agent Specification” (https://oracle.github.io/agent-spec/development/docs_home.html#open-agent-specification-agent-spec) to be define agents in a common text format (JSON or YAML).

You can convert your custom agents in this textual format to then be able to import it in PAF.



Oracle AI Database Private Agent Factory

It is the Oracle attempt at trying to solve the problem of how to build, maintain and run agents at an Enterprise scale. Keeping in mind security and privacy and using most of the existing Oracle pieces in your infrastructure (databases, Select AI etc.).

Still a very young product. Using it in production right now? Could be frustrating because of the limited inputs/outputs and components. Each release can change a lot of things still...
If you use, or plan to use, Select AI a lot in your database, PAF is a good "front-end" to manage workflows exposing those features to users.

- From a security point of view, I feel more comfortable exposing a Select AI agent team in a PAF workflow that I can easily call from a custom UI via the REST API instead of having users to directly connect to my database. PAF behave a bit like a proxy (and comes with SSL, SSO, observability etc.).

Worth keeping an eye on it, every new update should bring some great new features.