



Beyond MCP

# OAS, Agents, Skills & Teams

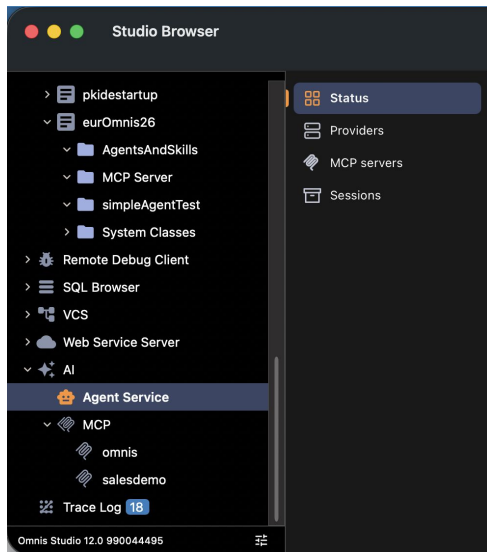
Direct agentic task support in Omnis Studio

- What is OAS?
- Agents & Skills
- A first agent, live
- A sales-ops team
- Recap + Q&A



# What is OAS - Omnis Agent Service ?

*A background service that manages & executes agentic sessions.*

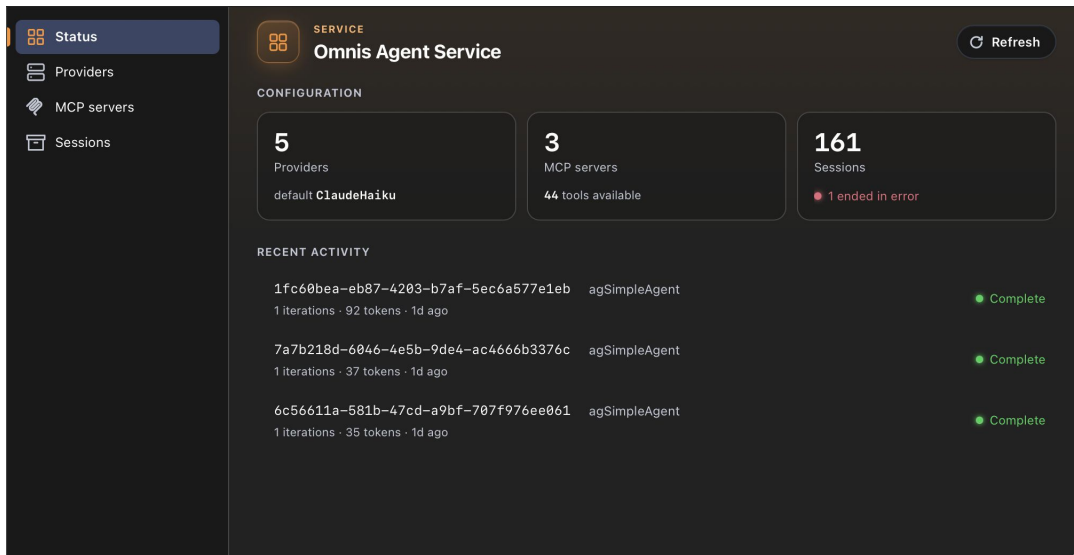


- UI Support in the IDE
- Manages providers (connections to LLMs )
- Manages MCP server connections & the tools available to agents
- Registers and runs agent sessions
- Runs the agentic loop - performing tool-calls
- Every session tracked end to end —  
**full visibility** into what an agent did and why

# OAS - Status

*The status panel*

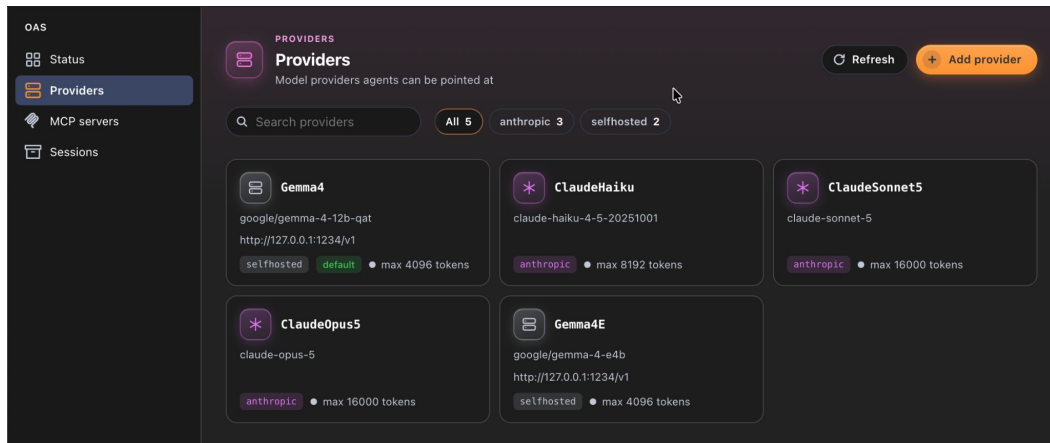
- Lists providers
- Live MCP servers
- Recent activity
- Ability to jump into logs of a session



# OAS - Providers

*Providers are used to connect an agent to an LLM.*

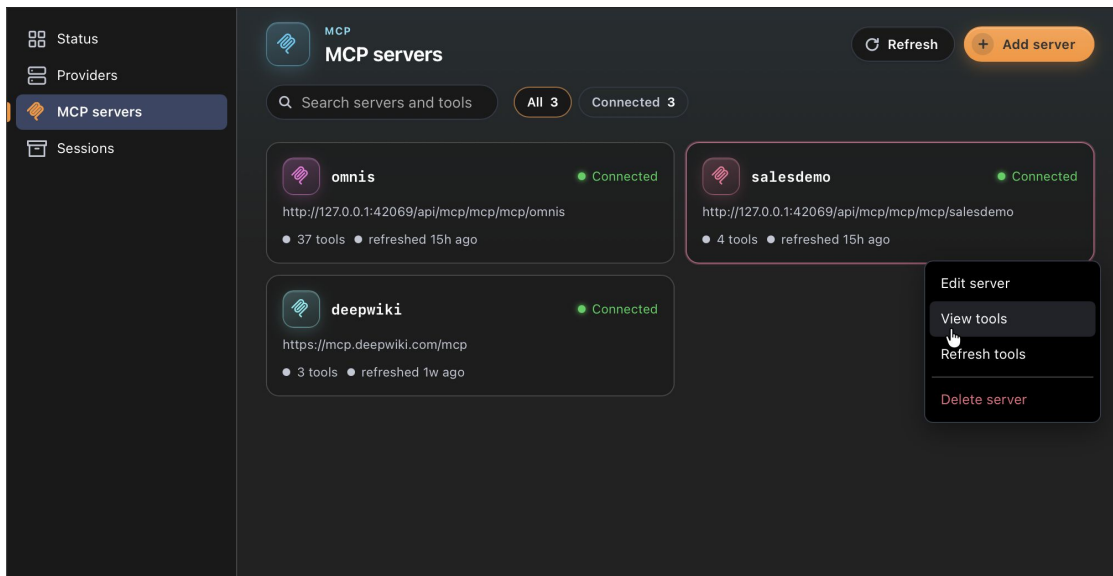
- Accessed via the IDE browser
- LLM - Self-hosted or hosted API
- Switch the default agent provider direct from the dashboard. No agent changes required
- Self-hosted example:  
LM Studio -> Gemma4  
Your data never leaves your infrastructure



# OAS - MCP Servers

*Live connections to MCP for your agents*

- Create multiple connections to live MCP servers
- View the MCP Tools
- Explore a tools function and its parameters



# OAS - Sessions

*Observe your agent sessions*

- All agent sessions logged
- Follow a session activity
- View
  - agent
  - model
  - full context
  - turns
  - tool-calls
  - tokens

The screenshot displays the OAS Sessions interface. On the left is a sidebar with navigation options: Status, Providers, MCP servers, and Sessions (highlighted). The main panel shows a list of sessions under the 'SESSIONS' header. Each session entry includes a unique ID, the agent name 'agSimpleAgent', the model 'claude-haiku-4-5-20251001', and statistics for iterations, tokens, and time. A 'Clear' button and a 'Refresh' button are in the top right. Below the session list, a detailed view of a specific session is shown. This view includes the session ID 'fa17108f-36df-45e7-90c1-36bc41b1fcd6', a timeline of events, and a 'Round 1' section showing a tool call 'salesdemo:tickets\_summary' and the session's starting prompt.

**SESSIONS**

Search sessions All 161 Complete 160 Error 1

| Session ID                           | Agent         | Model                     | Iterations | Tokens | Time   | Status   |
|--------------------------------------|---------------|---------------------------|------------|--------|--------|----------|
| 1fc60bea-eb87-4203-b7af-5ec6a577e1eb | agSimpleAgent | claude-haiku-4-5-20251001 | 1          | 92     | 1d ago | Complete |
| 7a7b218d-6046-4e5b-9de4-ac4666b3376c | agSimpleAgent | claude-haiku-4-5-20251001 | 1          | 37     | 1d ago | Complete |
| 6c56611a-581b-47cd-a9bf-707f976ee061 | agSimpleAgent | claude-haiku-4-5-20251001 | 1          | 35     | 1d ago | Complete |
| aa684691-f1b1-4ffe-83a5-3c9e978f4d03 | agSimpleAgent | claude-haiku-4-5-20251001 | 1          | 35     | 1d ago | Complete |
| 980bd548-6ccf-4073-ae7e-afd67076d3b7 | agSimpleAgent | claude-haiku-4-5-20251001 | 1          | 35     | 1d ago | Complete |

**SESSION** agTickets anthropic · claude-haiku-4-5-20251001

**fa17108f-36df-45e7-90c1-36bc41b1fcd6**

Timeline | High | 2 | 63 | 2026-07-20 |

| Billing | Medium | 1 | 45 | 2026-08-18 |

| Shipping | High | 3 | 38 | 2026-08-22 |

| Shipping | Medium | 1 | 51 | 2026-08-12 |

| Technical | High | 2 | 63 | 2026-07-20 |

| Technical | Low | 2 | 56 | 2026-07-25 |

| Technical | Medium | 2 | 43 | 2026-08-15 |

**Headline:** 17 open tickets across 8 category-priority combinations, with Technical High priority tickets being oldest at 63 days median age.

**Round 1** 1d ago

1 tool call 887 tokens

salesdemo:tickets\_summary

**Session started** 1d ago

Prompt

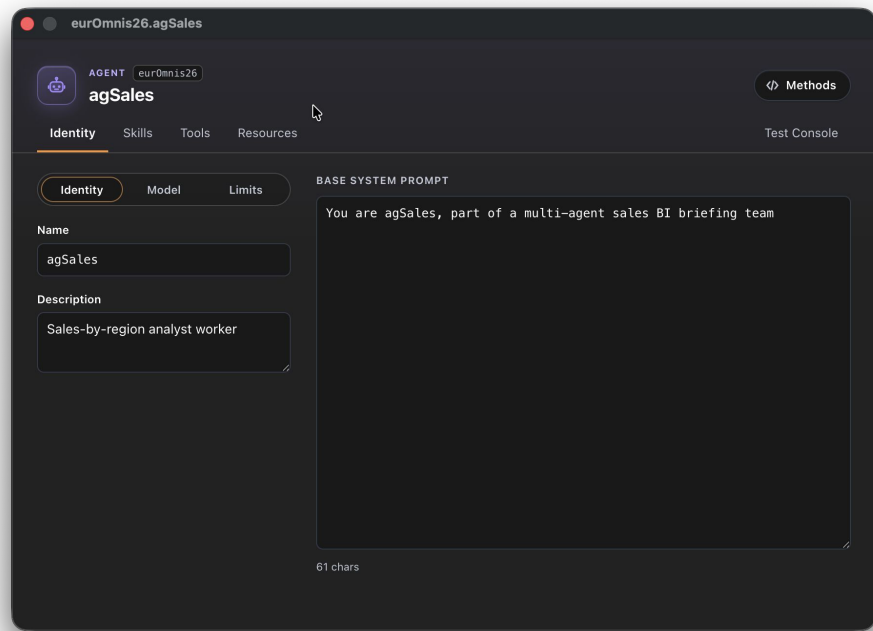
**SYSTEM PROMPT**

You are agTickets, part of a multi-agent sales BI briefing team.

# What is an Agent?

*Something that takes a task, reasons over it, and acts using the tools, skills and resources you allow.*

- The Agent object has a prompt, skills, tools, resources
- **Prompt** : The agent base instructions.
- **Tools** : Agents can **only** call the tools **you** **authorise**
- **Skills** : Fixed & runtime skills
- Uses the OAS **default provider** (LLM) or select a specific provider for the agent
- Built in **new Agent design editor**
- Test using the built-in **Test Console**
- **Observable**



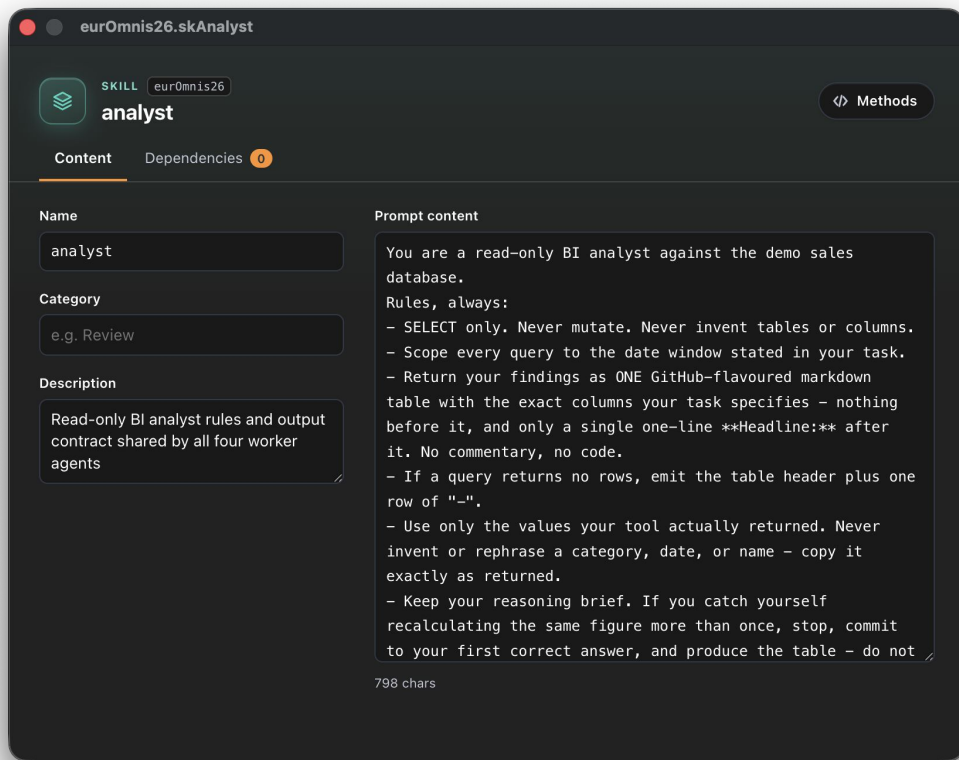
# What is a Skill?

*A reusable chunk of prompt.*

- Supports dependencies
  - Used by agents
- Agent support fixed and runtime skills

*#uses agent fixed skills - set in agent  
task="run a code review"  
agent.\$start(task)*

*# uses fixed and a runtime skill  
agent.\$start(task,"security\_check\_skill")*

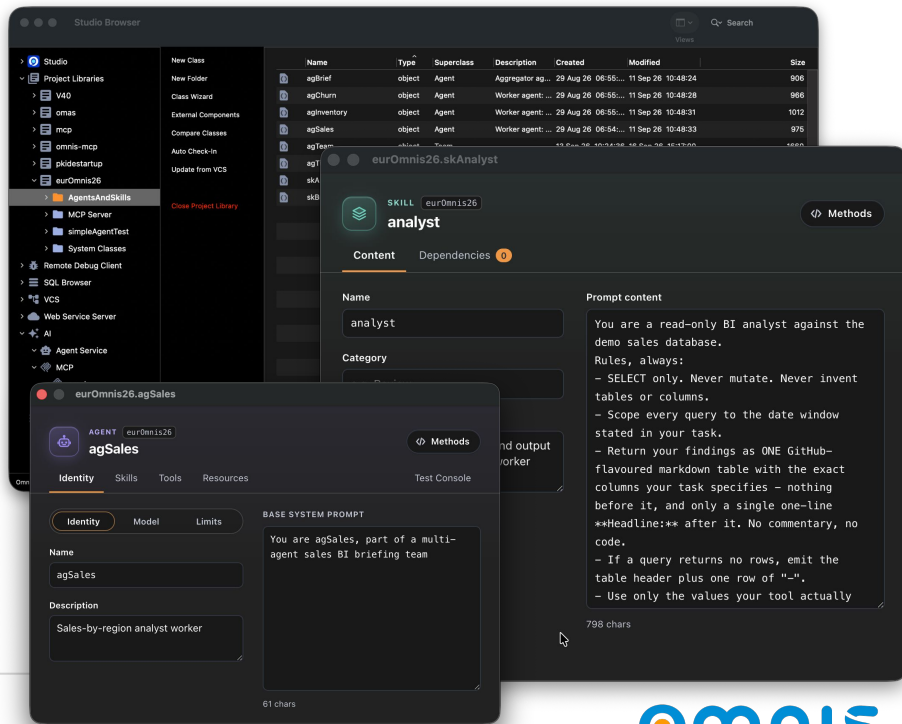


# Why Build Agents & Skills in Omnis?

*Build AI workflows using the development skills you already know.*

- **No new stack** — agents run inside Studio, alongside your existing app
- **New IDE editors** speed up agentic design
- New IDE editors offer integrated **testing consoles** — full visibility, no separate tracing tools bolted on
- Your agents are empowered with MCP tools under **your control**.

**Build, Test, Run, Observe** in the IDE you know.



# A Simple Agent

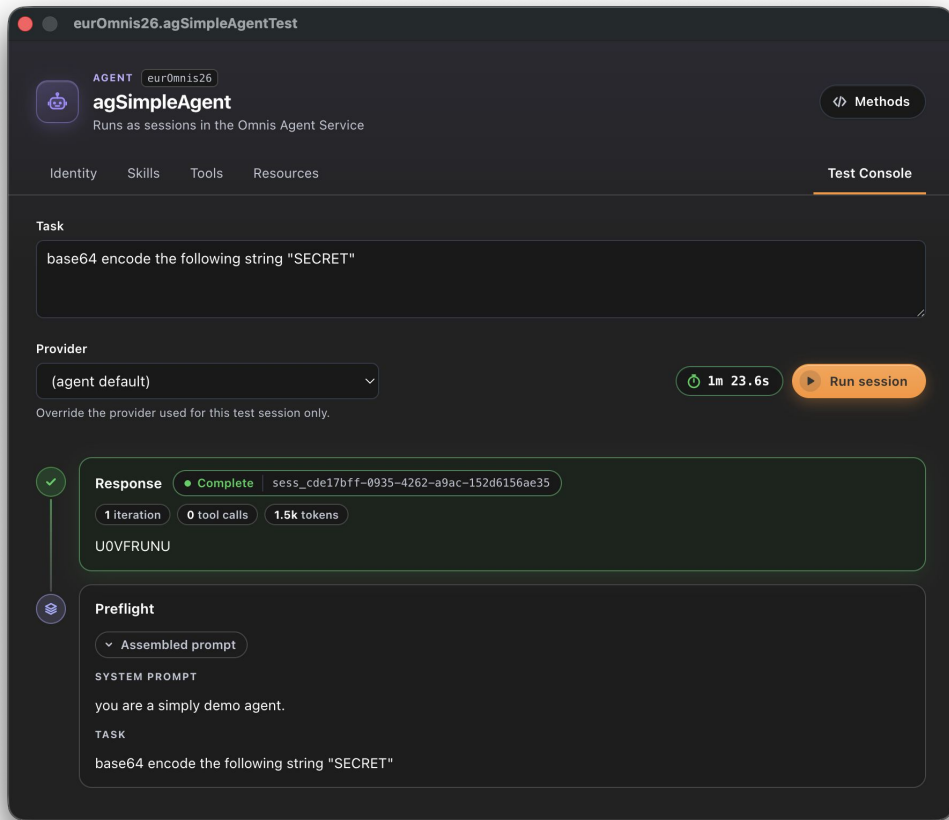
*One agent, one job.*

- **Test Console**

Create → Preflight → Complete

- Every step is observable

- All in the Studio IDE

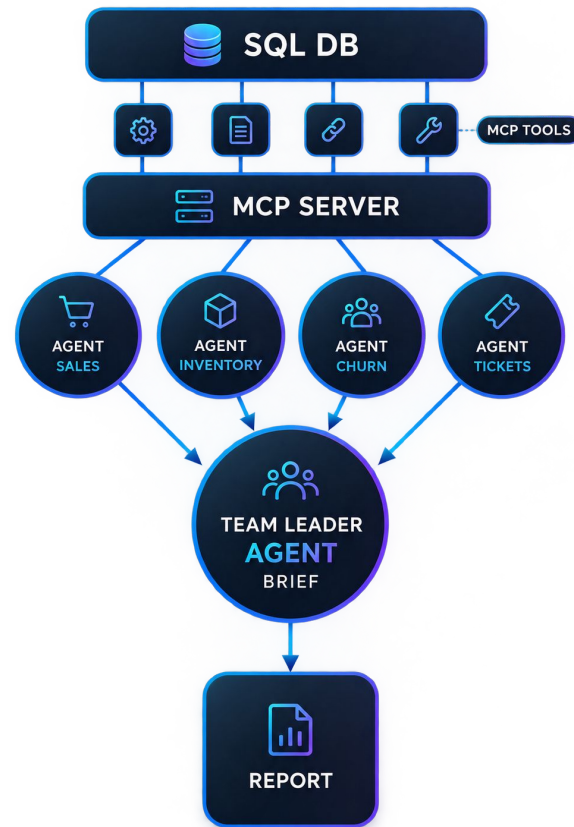


# A Simple Agent - Example

# From One Agent to a Team

*Sometimes real work needs multiple tasks....*

- Create **Dedicated agents** ( team members ) with dedicated tasks / perspectives
- Optional - A dedicated **team leader** who takes action on the team members results
- Members connected to a MCP server to use dedicated, secure tools
- That's a Team - **Let's Build this**



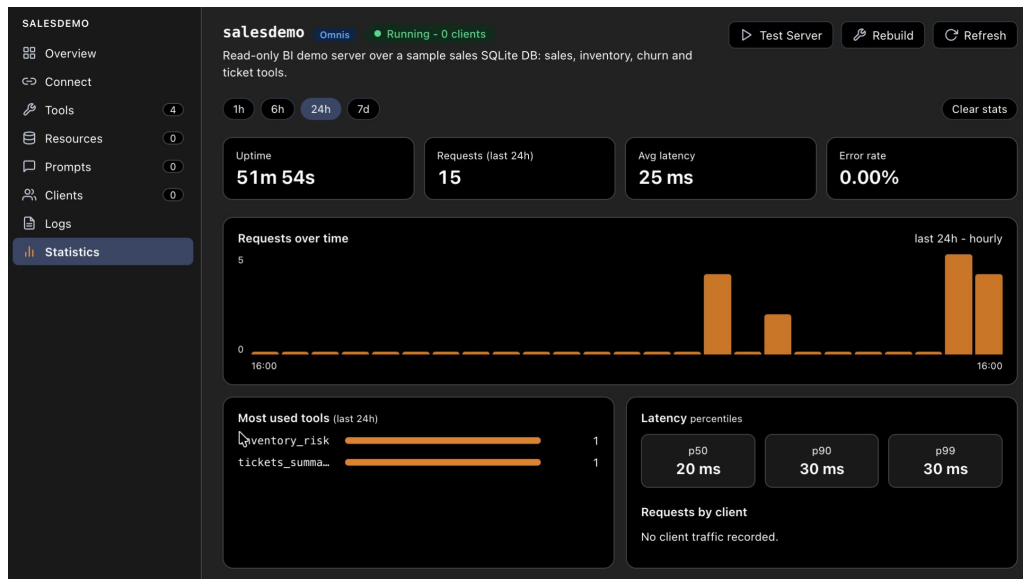
# Step 1 : MCP Server & Tools

*A developer-defined MCP server, tools and a real database.*

## *MCP Tools to access a table in DB*

- oTool\_SalesByRegion
- oTool\_InventoryRisk
- oTool\_CustomerRisk
- oTool\_TicketsSummary

Each tool performs a read-only query against a demo dataset.



# Step 2 : The Agents

*Four analysts and a leader  
Meet the team.*

- **agSales** — is an agent  
*Uses **oTool\_SalesByRegion** & is an **analyst** using the **skAnalyst** skill*
- **agInventory** — is an agent  
*Uses **oTool\_InventoryRisk** & is an **analyst***
- **agChurn** — is an agent  
*Uses **oTool\_CustomerRisk** & an **analyst***
- **agTickets** — is an agent  
*Uses **oTool\_TicketsSummary** & an **analyst***

|                       |   |             |                   |            |
|-----------------------|---|-------------|-------------------|------------|
| New Class             |   | Name        | Type <sup>^</sup> | Superclass |
| New Folder            | 📁 | agBrief     | object            | Agent      |
| Class Wizard          | 📁 | agChurn     | object            | Agent      |
| External Components   | 📁 | agInventory | object            | Agent      |
| Compare Classes       | 📁 | agSales     | object            | Agent      |
| Auto Check-In         | 📁 | agTeam      | object            | Team       |
| Update from VCS       | 📁 | agTickets   | object            | Agent      |
|                       | 📁 | skAnalyst   | object            | Skill      |
|                       | 📁 | skBriefing  | object            | Skill      |
| Close Project Library |   |             |                   |            |

- **agBrief** — our team leader  
*Uses no tools but **uses a skill - skBriefing***

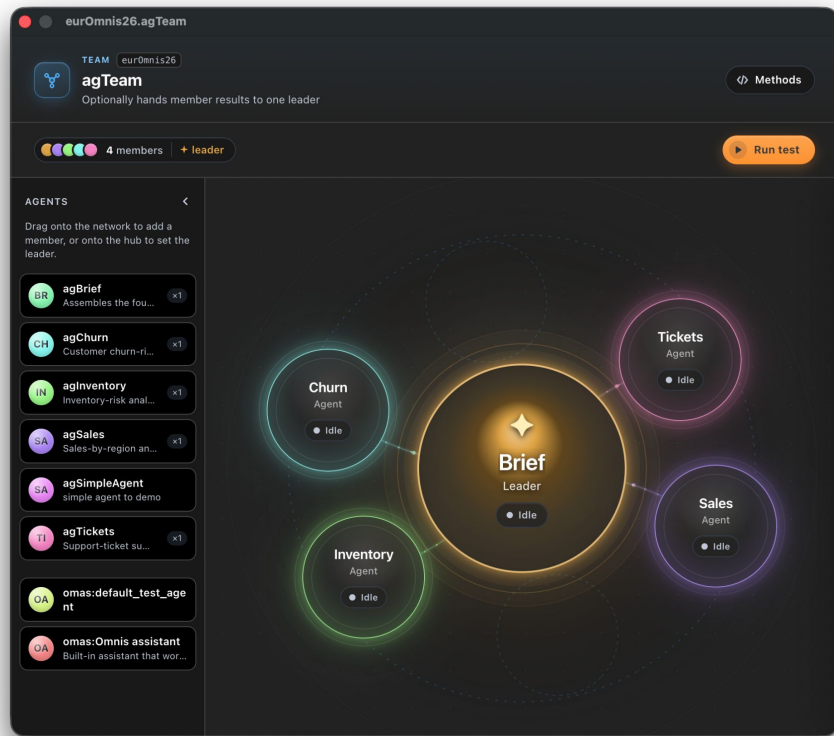
*Note : Our shared analyst skill enforces table-only output,  
no fabrication of data.*

We do this using the Team Editor

## Step3 : Agents Assemble

*Assemble our agents into a team*

- We need a **Team object** to orchestrate everything
- We add our **team leader** & Team **members - our agents**
- Members will run concurrently & report to the team leader when complete
- Our **team leader** ( optional ) will take action for the team and produce a result based on the members results



# The Live Dashboard

*One click, dispatched from a real Omnis window.*

- Team tested - lets add the team to a window
- Dispatch button starts the team

Do `$clib.$objects.agTeam.$newref()` Returns `iTeam`

Do `iTeam.$callbackinst.$assign($cwind)`

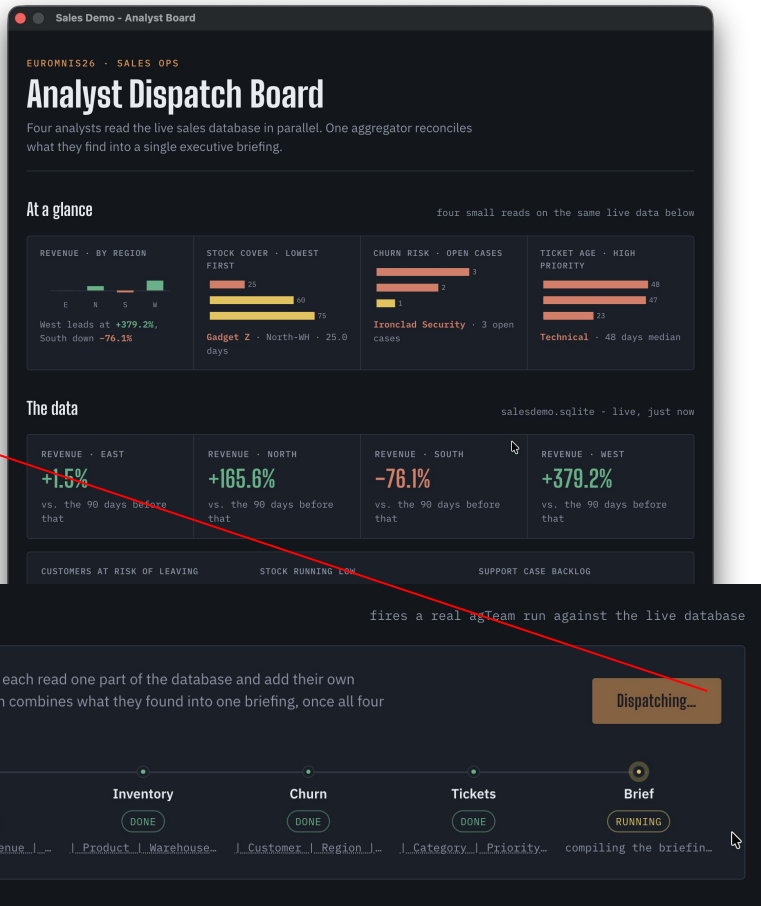
Do `iTeam.$start()`

Live 'per-member agent' progress stepper -  
notified when each member completes

 `$membercompleted`

- Briefing result rendered as markdown

 `$completed`

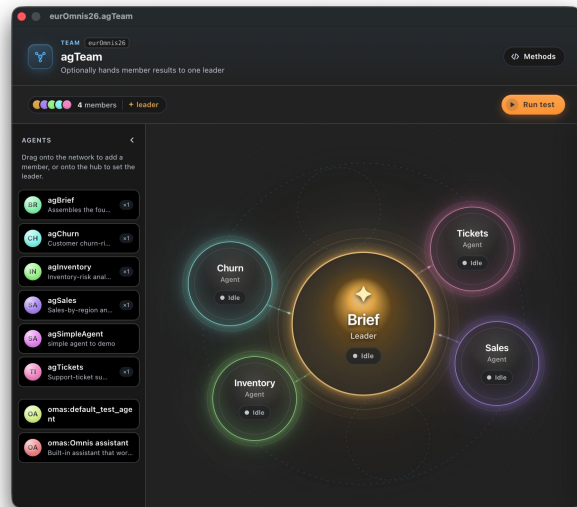


# Team- Example

# Recap

*From one simple agent to a live ops dashboard — small primitives compose into real systems.*

- OAS - A new Agent service
  - Agents: prompt + tools + skills
  - Skills: reusable prompt chunks
  - Providers: swap the model, not the code, hosted or local
  - Teams: fan out, then optionally pass to a team leader
- 
- Agents route into Omnis is **only via MCP** - and only using tools **you allow**.
  - Observability - Agents, OAS, MCP - Monitor everything
- 
- Intended preview end of October 26





# Questions ?

OAS, Agents, Skills & Teams — Thank you