

SAP BUILD DISCOVERY DAYS

Agentic AI and automation workflows using Joule Studio

Hosted by Semos Cloud. Live, no slideware for the build itself.

Host Ilija Kiroski, Enterprise Accounts

Build Martin Vasilkovski · Marko Apostoloski, Senior AI/ML Engineers

Today EMEA, September 24, 3:00 PM CET

A few housekeeping notes

- **Questions:** use the Q&A, not the chat. We'll answer as many as we can live as we move through the build, and we're tracking every one even if we don't catch it in real time.
- **Recording:** this session is being recorded, and we'll share the recording afterward.
- **Technical issues:** drop a note in the chat and we'll help where we can.
- **Chat visibility:** make sure Everyone is selected, not Host and panelists, before you send your first message. It defaults to the wrong one sometimes.

An SAP-native HR technology company

Built on **SAP BTP** for over a decade. Today, **24 AI agents** in production for enterprise customers around the world.



SAP runs on us

SAP Appreciate, SAP's own employee recognition program, is built on Semos Cloud: **100,000** employees, more than **500,000** recognitions in 2025.



Global scale

150+ enterprise customers, **2M+** active users, across **170+** countries.



Partner status

10-year SAP-certified ISV partnership. SAP Spotlight+ for Total Rewards.



Built with SAP Business AI

Certified designation, alongside Spotlight+.

WHY NOW

SAP is funding your first move into Business AI

Through SAP's **Partner-Led Adoption Incentive**, SAP invests in a certified partner to build and deploy your first AI agents — so value comes faster, with less risk.



SAP funds it

Up to €100K of project investment. SAP backs a defined, scoped delivery for eligible customers on the SAP Business AI Platform.



A certified partner builds it

Semos Cloud designs and delivers real, grounded AI agents inside your environment — on Joule and SAP Build.



You adopt and see value

Discovery workshop, deployment, and a 30/60/90-day adoption plan. You choose the partner; you own the outcome.

Built live, by the people who wrote it



Martin Vasilkovski

Senior AI/ML Engineer



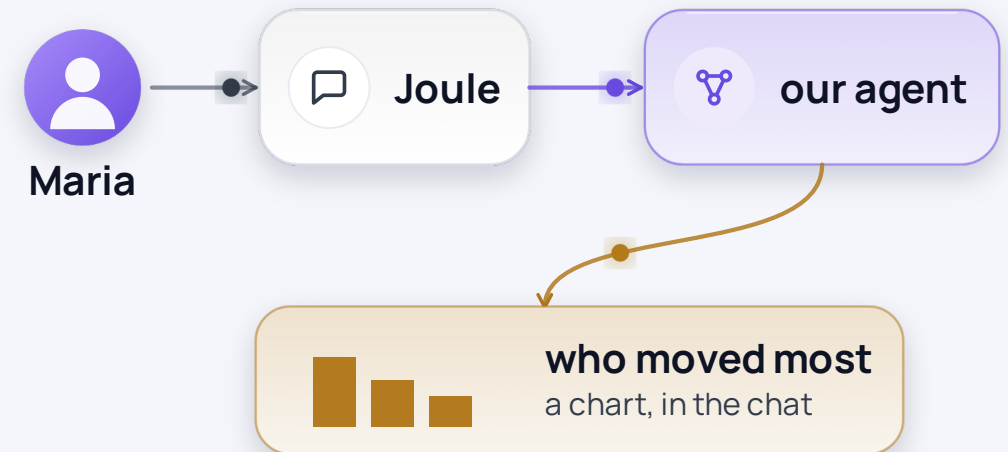
Marko Apostoloski

Senior AI/ML Engineer

One agent for Joule

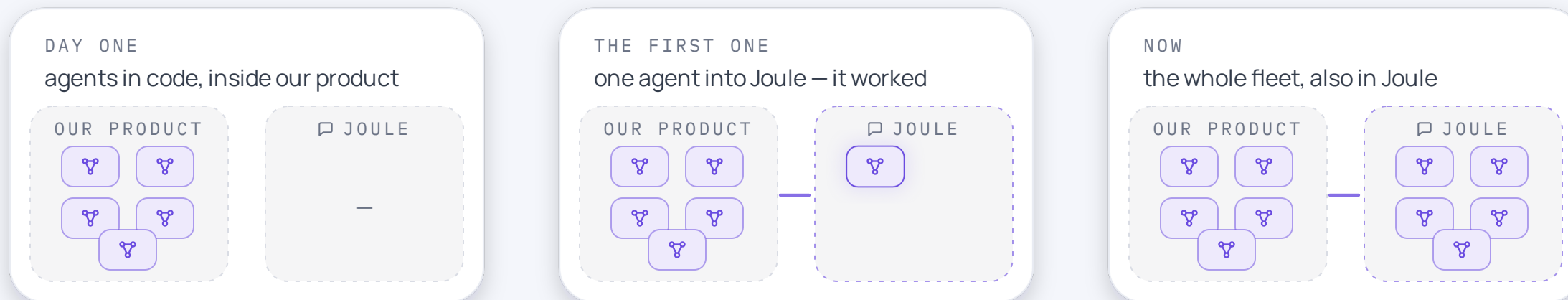
A manager asks about her team.
Joule answers — with a chart. Every
piece in between, named. Then the
code.

stack LangGraph · MCP · A2A · Gen AI Hub · Cloud Foundry · Joule
today demo · how it works · the code, live



Pro-code agents, from day one

- **Semos Cloud** makes recognition, rewards and talent software on SAP SuccessFactors. Our agents are part of that product.
- **Written in code from the start** — our own graphs, prompts and tools, in LangGraph. No builder, no template.
- **One agent went into Joule first.** It showed us what Joule could do for our users, so the whole fleet followed.



Today we do the first step again, in the open: one pro-code agent, from an empty repository, into Joule.

What we will cover

First a question and a live demo. Then how it works, in five parts – then the code.



01

The picture

One app on SAP BTP:
Joule, the agent, the
model, the data.



02

Connecting to Joule – A2A

How Joule finds the
agent, calls it, and the
Skill that points at us.



03

How it thinks – LangGraph and DuckDB

What LangGraph is, our
graph, SQL on a copy,
three rules.



04

A chart in the chat

Why templates can't
draw, and the two-
artifact answer.



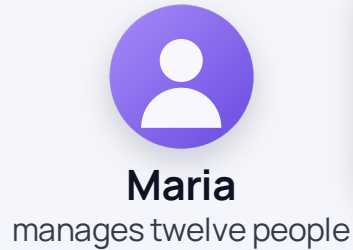
05

Running it

One app on Cloud
Foundry – then the
code, on your machine.

One question, three things to deliver

She manages twelve people on our R&R product. She wants to know who moved most on engagement last quarter.



Who on my team moved

most on engagement
last quarter?

Today: a dashboard, filters, an export. **We want her to ask Joule.**

dashboard

filters

export

just ask Joule



Her numbers, only hers

The transactional reports of our rewards and recognition product, fetched as they are, scoped to her team.

access



An answer, not a table

"Last quarter" turned into dates, rows ranked, movers named.

reasoning



A chart, in the chat

A real Joule card. Templates can't build one from text – the hard part.

rendering

Maria asks Joule. Live.

DEMO



The question

Typed into Joule in plain words: who on my team moved most on engagement last quarter?



The answer

Names and numbers from one manager's team — nothing invented.

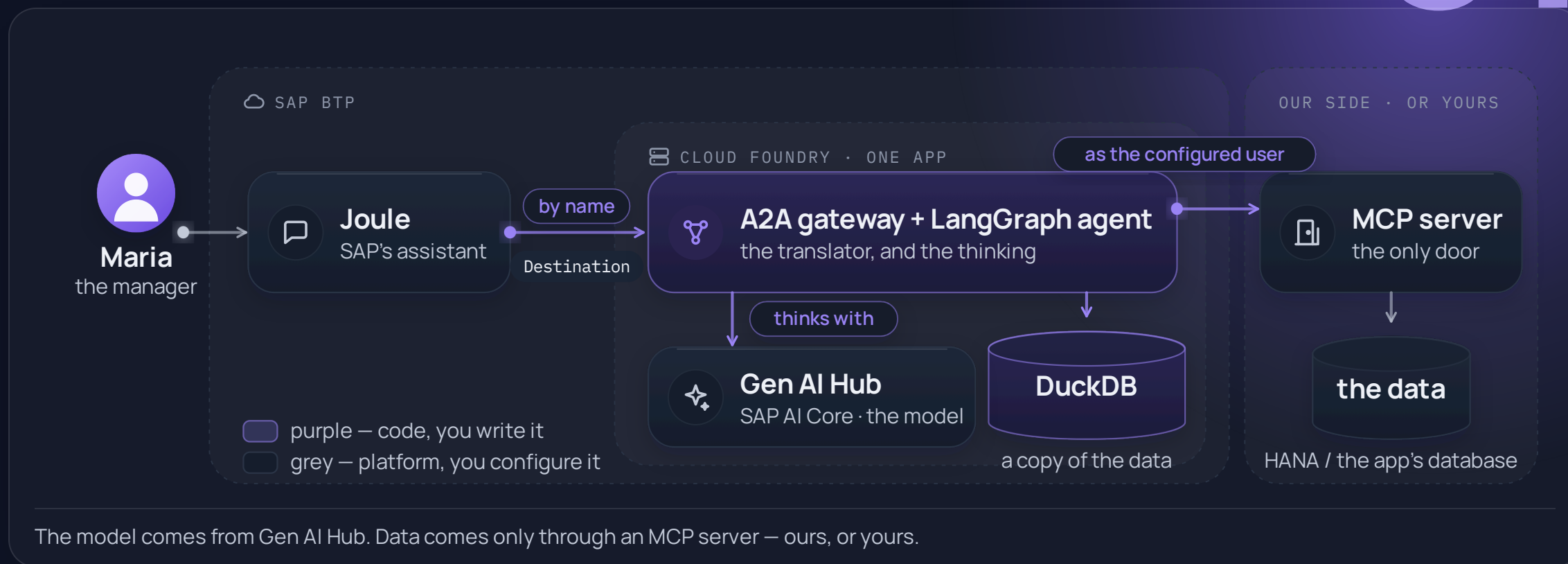


The chart

A real Joule card under the answer, drawn from the same numbers.

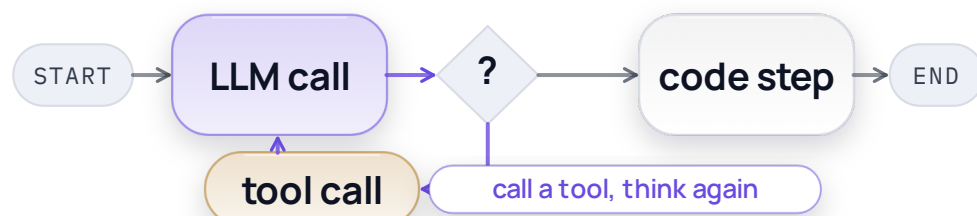
Watch for those three. First the whole picture, the two technologies and the SAP services you need, then how each one happens — then we open the code.

The whole picture



MCP server Model Context Protocol: an open standard for calling tools. The server writes every query.

Two technologies to know



one state object travels through every step

LangGraph – build an agent as a graph

- Open source, from the LangChain team. Widely used.
- Plain Python: runs on a laptop, on Cloud Foundry, anywhere.
- Every run is inspectable, step by step, in a studio.



in memory, or a file – gone when the process ends

DuckDB – SQL inside your app

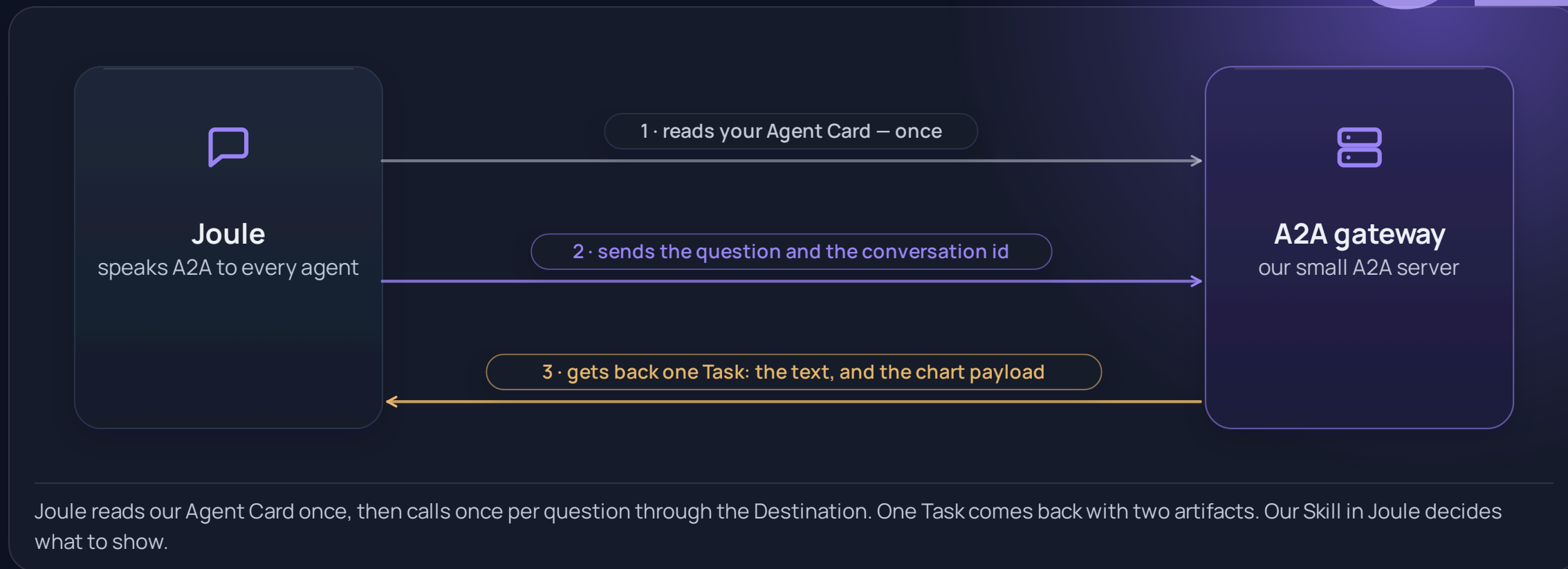
- An analytics database as a library. No server, no setup.
- Load files or rows, query with plain SQL. Fast.
- In memory or a file. Lives and dies with your process.

The SAP services you need

SERVICE	WHAT YOU NEED	WHO
SAP BTP	A subaccount with Cloud Foundry, in a region with Joule and AI Core	subaccount admin
Cloud Foundry	A space you can push to	Space Developer
Destination	One entry that points at the app URL	Destination Administrator
SAP AI Core	A resource group, one model, a service key	the app's service key
Joule	A tenant; publish the Skill with the CLI	a person, with the two Joule roles
MCP server	Fronts your data; the only door to it	platform team

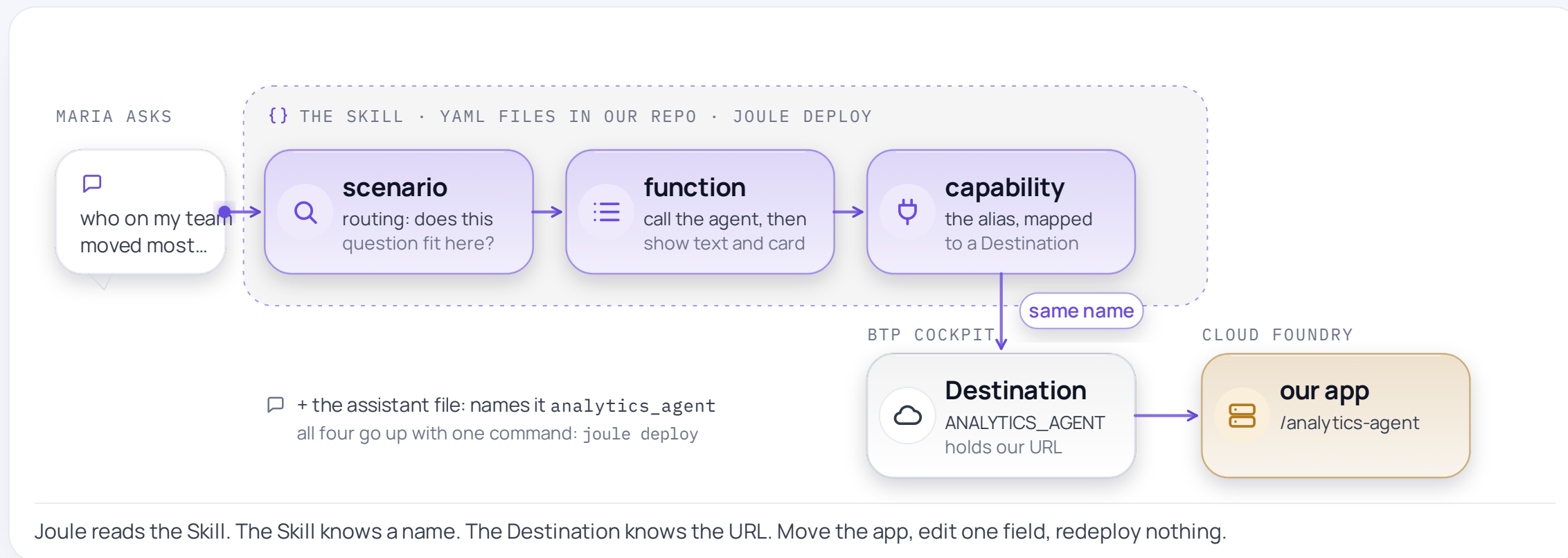
All of it is available through **SAP TDD** – Test, Demo, and Development.

A2A: how Joule calls an agent



the gateway Accepts the task, keeps the conversation's memory, returns the result.

Joule, and the Skill that points at us



Joule SAP's assistant, already inside the apps people use. Partners extend it with Skills.

a Skill A few YAML files: when to route a question to an agent, and what to show back. No code.

What is LangGraph?

Four building blocks are enough to read our agent. Here is where each one lives.



State

A typed record every step reads and updates. **Here:** the conversation, the chart payload, and what DuckDB loaded.



Nodes

Plain Python functions: take the state, return what changed. **Here:** hydrate, agent, tools, ui_synth.



Edges

What runs next. A conditional edge decides at run time. **Here:** after the agent – another tool, or on to ui_synth?

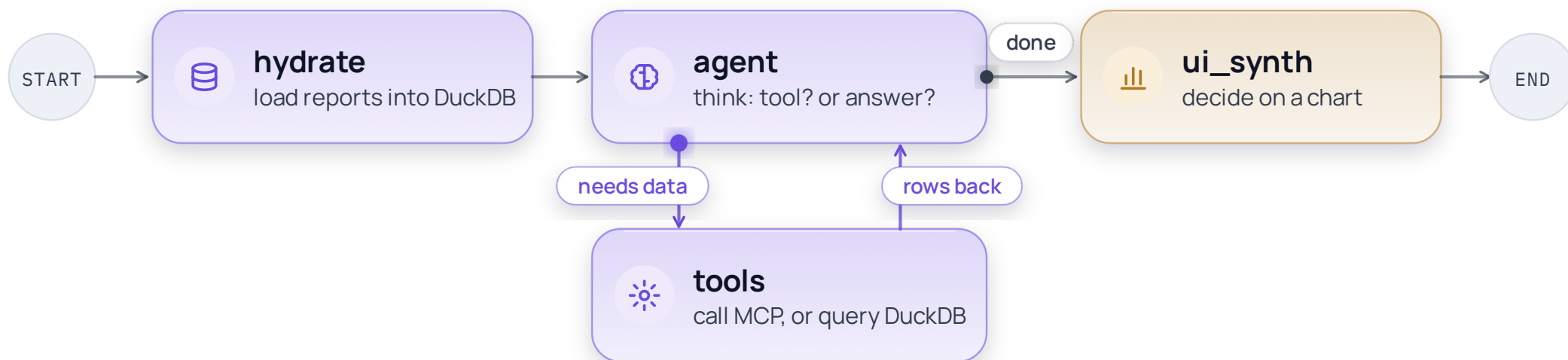


Checkpoint

Saves the state after each step, per conversation. **Here:** why a follow-up question remembers the first.

Why not one prompt with tools? Because the loop is explicit – we can read it, test each node, and cap it.

Our graph: five steps, one file

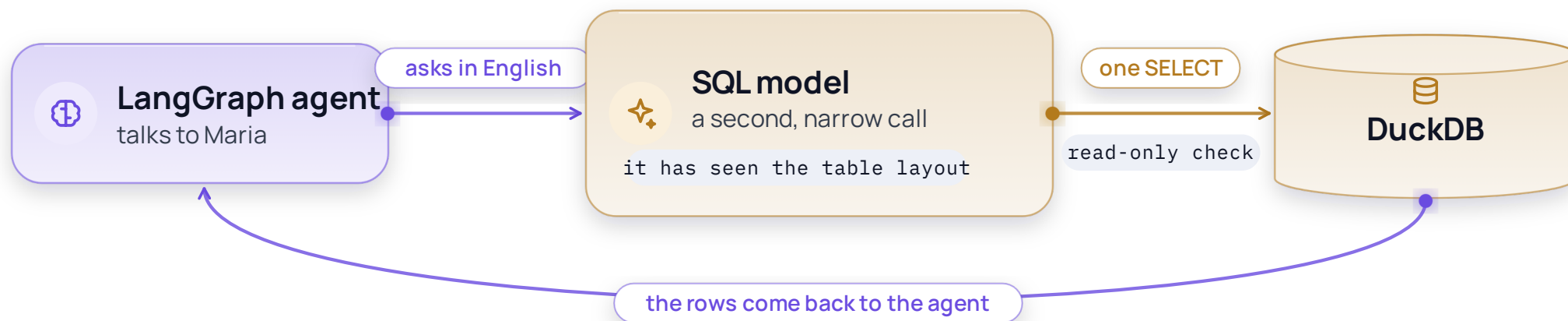


Think, call a tool, look at the rows, think again. At the end, decide on a chart.

LangGraph The agent is a graph of readable steps. Same code in our product and on Cloud Foundry.

DuckDB A small in-memory database. Reports become tables for one conversation, and model-written SQL runs only there.

Why the SQL runs on a copy



The agent never sees a column name. If the SQL is wrong, DuckDB says so and the SQL model tries once more.

The agent asks in English. A second, narrow call that knows the table layout writes one SELECT.

why a copy

Model-written SQL never touches the system of record. It runs only on this conversation's copy.

read-only check

Every query is inspected before it runs: read only, one statement, a row limit.

Three things it is never allowed to do



Guess a number

Every figure comes from a row a tool returned. An empty result is an answer, not a gap to fill.

the system prompt



Show an internal ID

Stripped from SQL results in code, and the prompt forbids showing any identifier. People are named.

in code + prompt



Change anything

Allowlisted MCP tools and a read-only SQL guard. There is no write path to forget to block.

in code

Each rule closes a failure we actually saw.

Joule's templates can only fill in text

1 · THE AGENT ANSWERS

🗨️ **Ana moved most: +12.**

Then Ben, +9, and Chen, +7.

[chart: Ana 12 · Ben 9 · Chen 7]

2 · THE SKILL TEMPLATE

message:

<? apiResponse.text ?> |

one slot · one string

3 · MARIA SEES

🗨️ **Ana moved most: +12.**

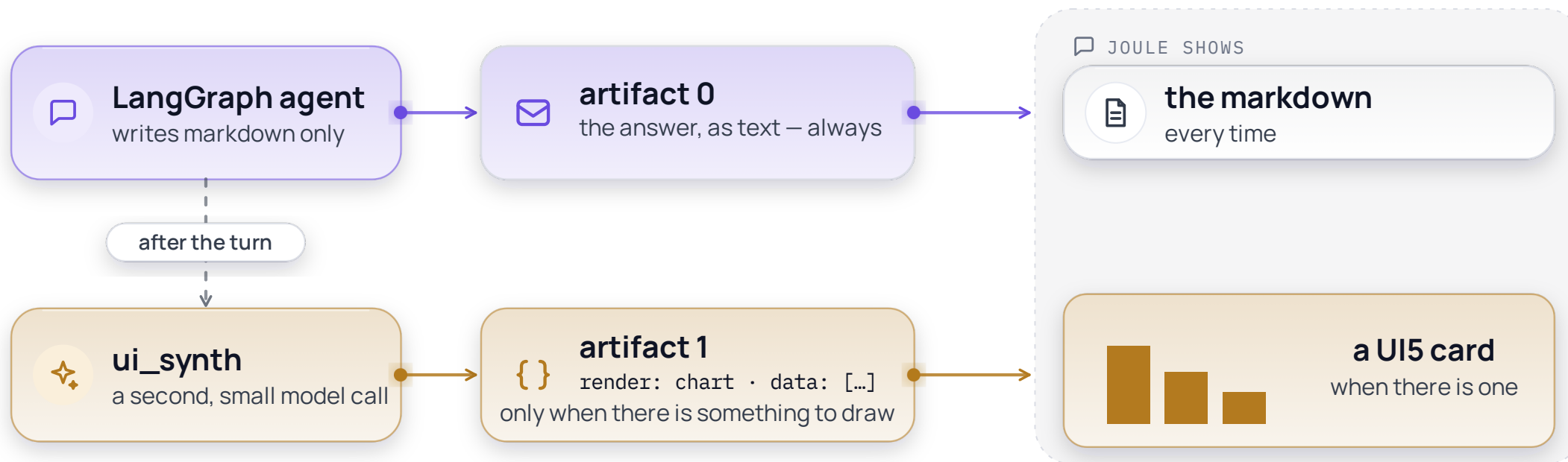
Then Ben, +9, and Chen, +7.

[chart: Ana 12 · Ben 9 · Chen 7]

✗ the same words back – no chart

A Skill template has one text slot. A chart described in words stays words.

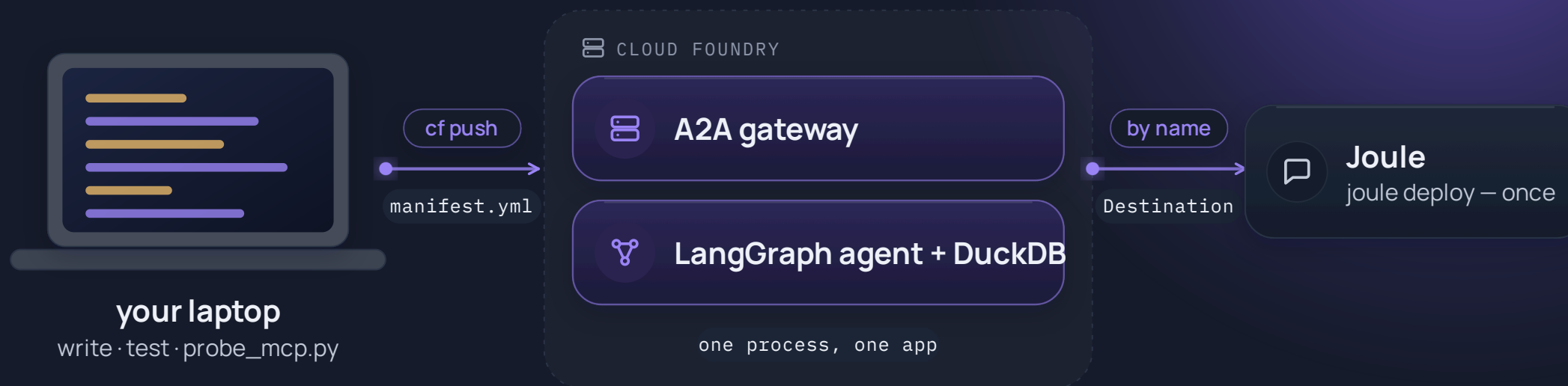
Two artifacts: the answer, plus a payload



The agent writes text. A second small call adds a typed payload: what to draw, with the numbers.

Text always. The card only when the payload is good.

You write it on a laptop. It runs on Cloud Foundry.



One app, one push. Then one Destination, and publish the Skill once.

Getting the code running



The repository

We give you access to the GitHub repo. Clone it; everything is inside.



One .env file

Copy `.env.example` to `.env`.
That is the only configuration.



Your AI Core key

Client id, secret, auth URL, base URL — from the service key. Keep `$` characters quoted.



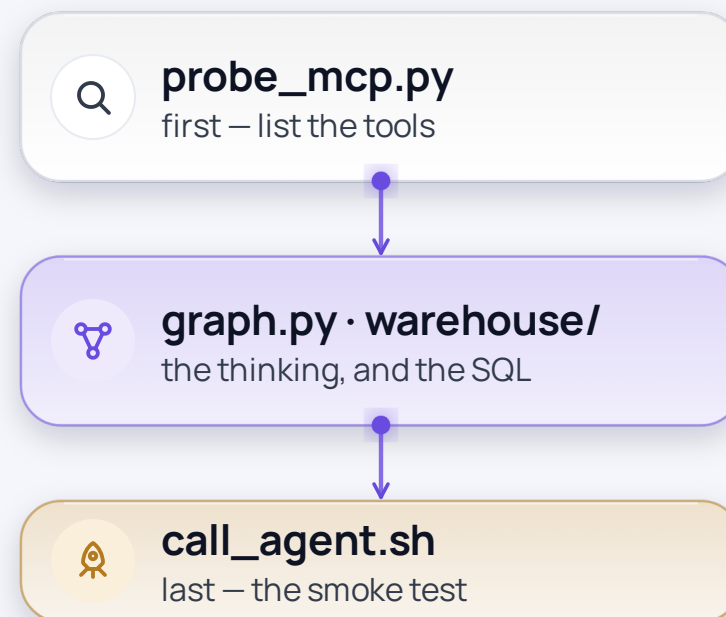
MCP and the email

MCP URL, the three allowed tools, and the one email the app answers for.

`.env` is never committed. Run `probe_mcp.py` first: if it lists the three tools, the setup is right.

The repository, in the order of the story

- 01 `scripts/probe_mcp.py` list the MCP tools
- 02 `src/mcp_client.py` the door and the allowlist
- 03 `src/agent/graph.py` the five LangGraph nodes
- 04 `src/warehouse/` DuckDB hydration and text-to-SQL
- 05 `src/gateway/` the A2A gateway and the Agent Card
- 06 `joule/` the Skill: scenario and function YAML
- 07 `src/agent/ui_synth.py` the typed payload



Top to bottom is the order of the story. Questions welcome as we go.

NOW, LIVE

Into the code

CODE



The agent

The graph, the MCP tools, SQL on the copy, the payload. Then one question traced in LangGraph Studio.



The connection

The Agent Card, the gateway, the four Skill files, and the one Destination that ties them together.

Eight stops, top to bottom. Interrupt any time — the code is yours as well.