

Making Agents Work

From lab to production

Boston Code Camp #40
28-Mar-2026



Connect with Bill

linkedin.com/in/billwilder
blog.codingoutloud.com
[@codingoutloud](https://twitter.com/codingoutloud)

HELLO
my name is

Bill

Bill Wilder
AI x Azure x Cybersecurity
billw@devpartners.com



Boston Code Camp

Thanks to our Sponsors!



MILL5

Gold



 **Progress**® Telerik®

triverus

PULSAR
SECURITY 

BRIGHTSTAR  **Silver**

CALL FOR PAPERS
POWERED BY



In Kind



Microsoft

Hosting Sponsor

<https://bit.ly/billbook>

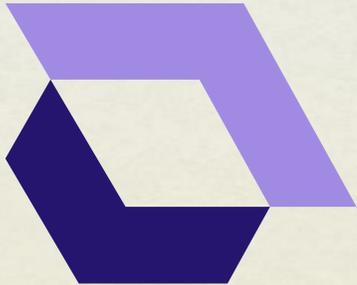


16x recipient
(currently AI MVP)

author

CTO at Open Admissions

AI x Azure x Cyber



AgentFramework.dev

instructor

founder



billw@devpartners.com

<https://meetup.com/bostonazureai>
@bostonazureai

Impossible
suddenly
possible



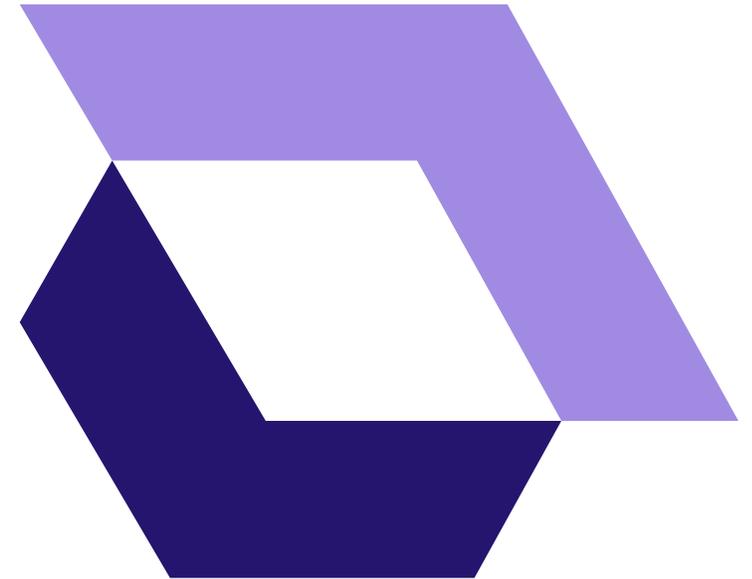
Based on my experience
building a production-ready AI
Agent for high school students
at Open Admissions
openadmissions.io

or via

← QR code



Lessons Learned buiding the
first 90%, then the second
90% ...



openadmissions.io

Non Goals

- ✗ Building an Agent from Scratch
- ✗ General Agent Framework
- ✗ General Microsoft Foundry

But stay tuned for **demos** &
project **source code** posted
(see slide at end for link)

Making Agents Work

- Observable
- Safer
- Consistent*ish*
- Deterministic

Observability: Make the Invisible Visible

Just like past
50 years

OTel as
Central Actor

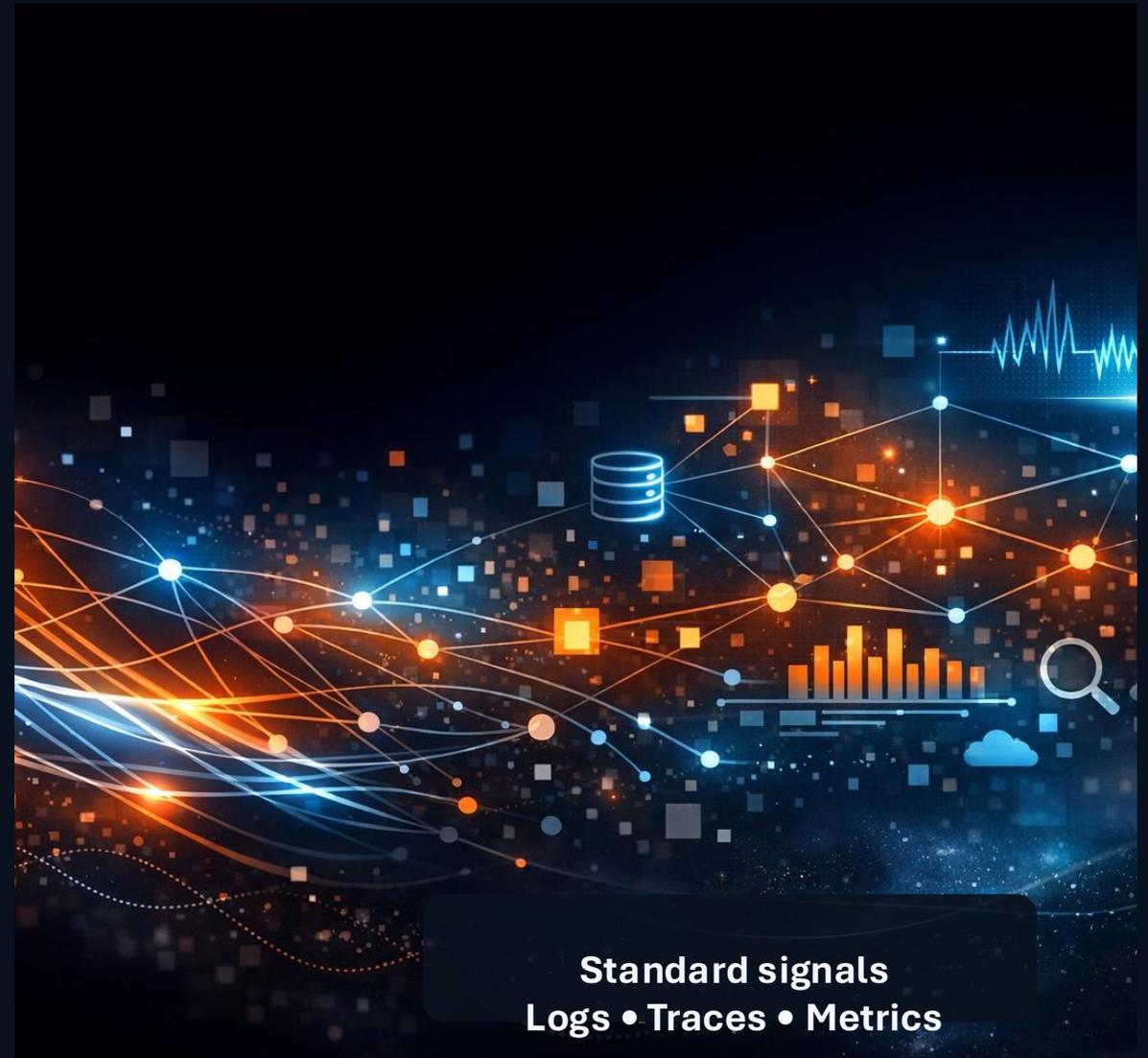
OpenTelemetry Standard

The neutral telemetry layer for traces, metrics, and logs.

Why it matters

- Trace one request across services, calls, and dependencies.
- Measure latency, failures, throughput, and resource cost.
- Correlate logs, metrics, and spans with a common model.
- .NET maps cleanly to ILogger, Meter, and Activity. Most modern platforms also have great alignment.

*OTel is the instrumentation **standard** that monitoring products support.*



OTel Semantic Conventions for GenAI

[https://opentelemetry.io/docs/specs/semconv/**gen-ai**/](https://opentelemetry.io/docs/specs/semconv/gen-ai/)

[https://opentelemetry.io/docs/specs/semconv/gen-ai/**azure-ai-inference**/](https://opentelemetry.io/docs/specs/semconv/gen-ai/azure-ai-inference/)

[https://opentelemetry.io/docs/specs/semconv/gen-ai/**mcp**/](https://opentelemetry.io/docs/specs/semconv/gen-ai/mcp/)

OTel GenAI Shared Vocabulary - Examples

```
error.type  
gen_ai.operation.name  
gen_ai.provider.name  
gen_ai.request.model  
gen_ai.usage.input_tokens  
gen_ai.input.messages  
gen_ai.output.messages  
gen_ai.system_instructions  
mcp.method.name ← MCP  
gen_ai.provider.name =  
"azure.ai.inference"
```

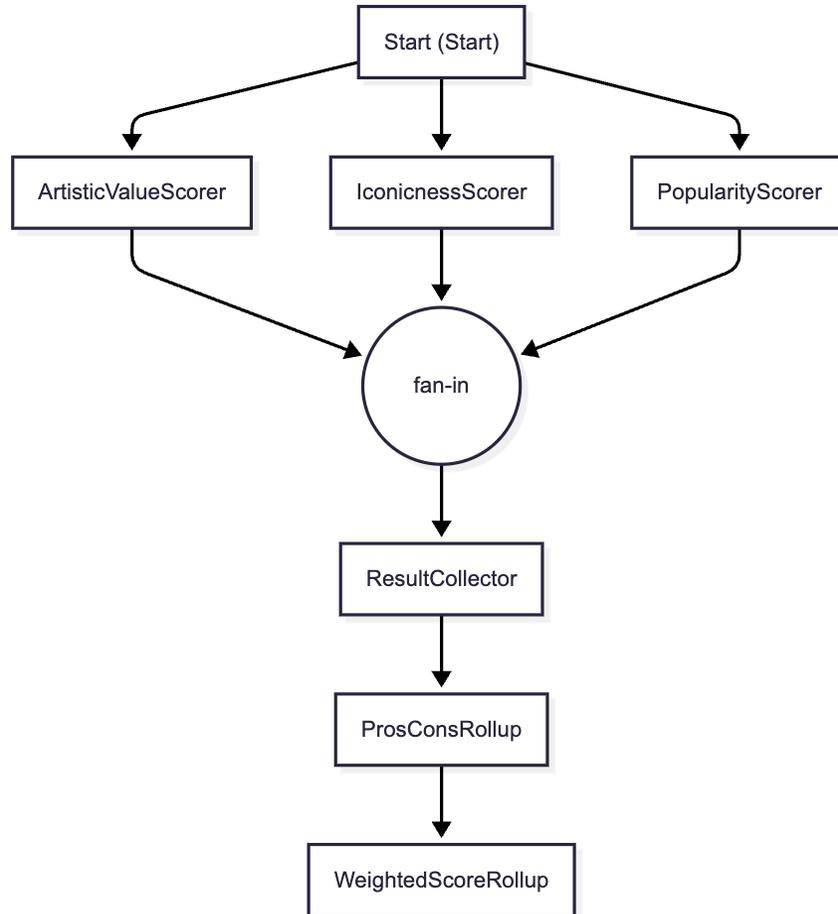
My Stack

- **C#**, Python
 - .NET 10, Aspire, OTel
- Agent Framework
 - Agents, Workflows, OTel
- Microsoft Foundry
 - Agents: Workflows, Evals, OTel
 - Models: LLM serving, Safety
 - Hosting: Function, ACA, Foundry Hosted Agent
- VS Code
 - AI Toolkit extension
- GitHub Copilot, Claude Code
 - Codex, Gemini



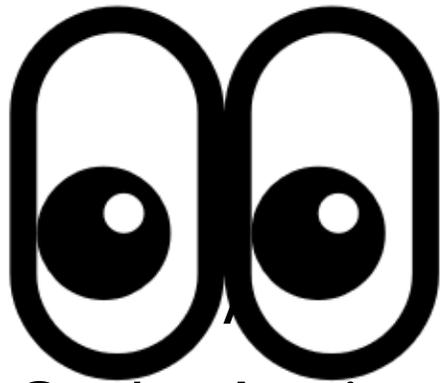
Introducing the Movie Rater Agent

HTTP **POST**: /jobs with movie name in payload
HTTP 202: 1b363a4c8db14ab68f5c88ece357a8e8

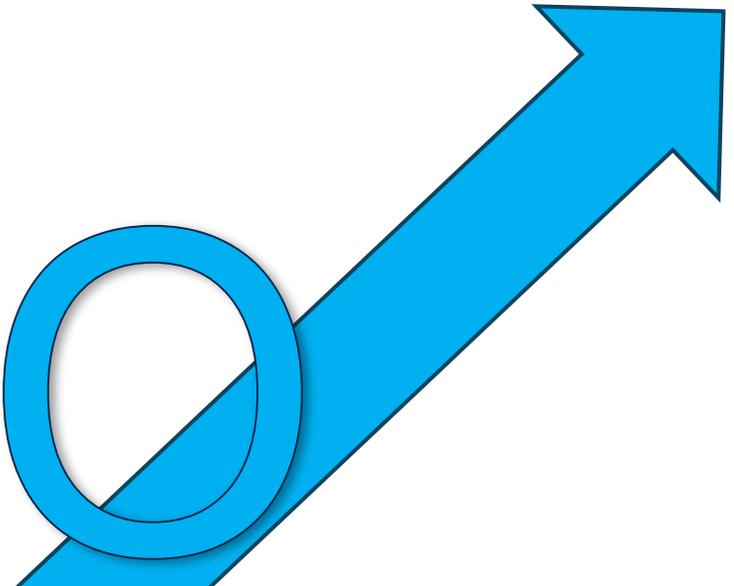


HTTP **GET**: /jobs/1b363a4c8db14ab68f5c88ece357a8e8
HTTP 200: results object with movie rating and summary

Observability via Open Telemetry



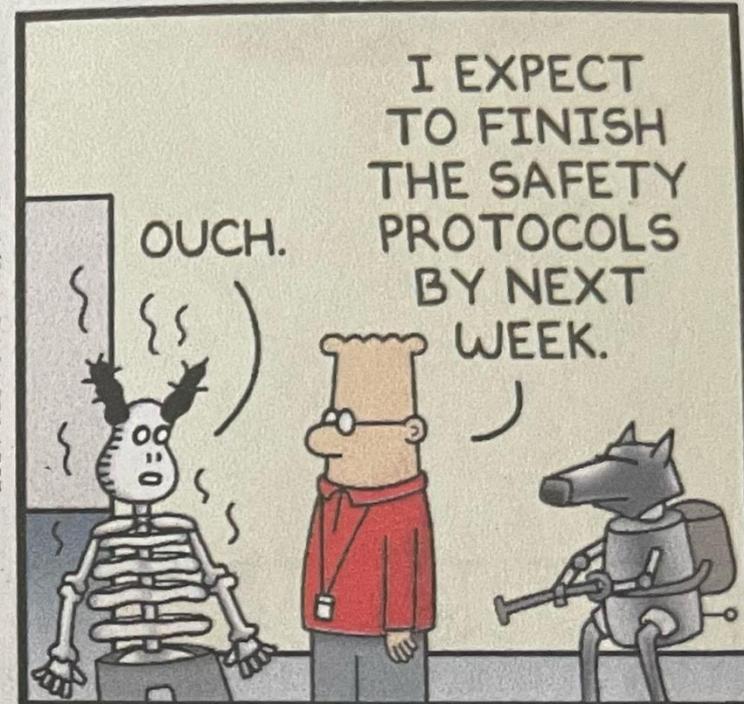
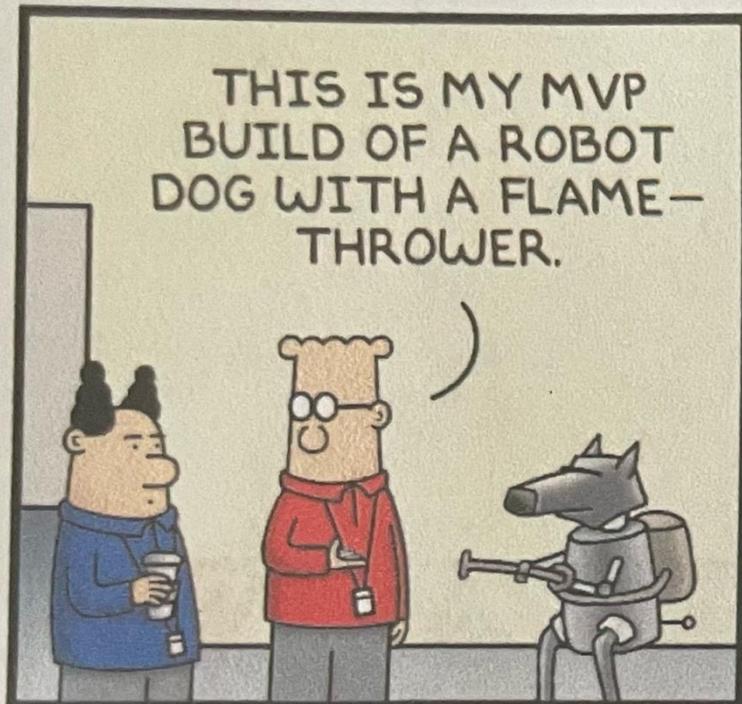
- normal
- Otel + Aspire – errors
- OTel + Microsoft Foundry

DEMO 

Making Agents Work

- ✓ Observable
- Safer
- Consistent*ish*
- Deterministic

Dilbert Reborn



© 2024 SCOTT ADAMS, INC.

Let's finish the safety protocols on your agent...

THURSDAY



Safety: #\$\$@% Users!

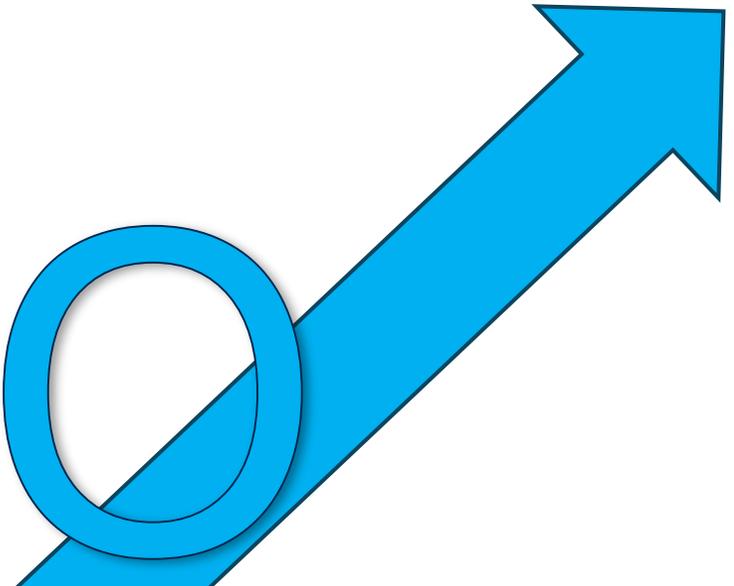
Users say
the darndest
things

Sometimes
users are
hackers

Safety



- Model Safety in Microsoft Foundry
- OTEL integration

DEMO 

Making Agents Work

- ✓ Observable
- ✓ Safer
- Consistent*ish*
- Deterministic

Retirement dates for gpt-4o versions 2024-05-13 and 2024-08-06 have been extended for all deployments except Standard

You're receiving this notification because you're associated with one or more Azure subscriptions that use Azure OpenAI.

To give you additional time to migrate to newer models, **we're extending the retirement date for Azure OpenAI gpt-4o versions 2024-05-13 and 2024-08-06 to 1 October 2026**

for the following deployment types:

- Global Standard
- Global Provisioned
- Global Batch
- Data Zone Standard
- Data Zone Provisioned
- Data Zone Batch
- Provisioned Managed

If you have Standard deployments, the retirement date will remain **31 March 2025**.

As part of this retirement, the replacement model will be set to **gpt-5.1 version 2025-11-13 on 9 March 2026**. If you've selected the option to auto upgrade your standard deployments, they will be automatically upgraded to the replacement model version during the weeks prior to retirement. [Learn more about setting your deployment upgrade options.](#)

After the retirement date, model gpt-4o versions 2024-05-13 and 2024-08-06 will no longer be available or operable. This retirement will only impact base-model deployments.

Required action

“After the retirement date, model **gpt-4o** versions 2024-05-13 and 2024-08-06 will no longer be available or operable”

“As part of this retirement, the replacement model will be set to **gpt-5.1**”

gpt-4o → gpt-5.1

Evals: Understand Behavior at Scale

LLMs are
non-
deterministic

So what
does “right”
look like?

Evals for Speed with Confidence

- Changed prompt or system prompt
- Switched model settings (e.g., temperature)
- Changed to a different model (which might not be up to you!)
- Upgraded same model
- Rewired your workflow
- New data inputs
- ...

Evals

- A technique, not a tool
- But there are tools
 - Microsoft Foundry
 - .NET has abstractions
 - <https://agenteval.dev/>

```
Passed: 1
```

```
Total time: 16.6411 Minutes
```

```
GoatCheck.Evals test net10.0 succeeded (998.6s)
```

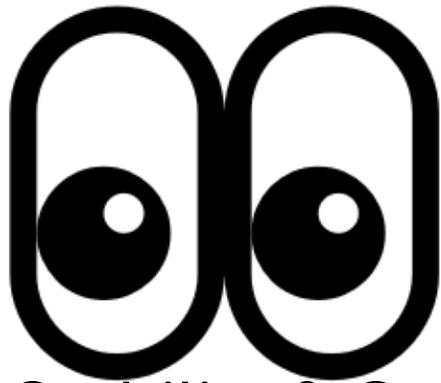
```
Test summary: total: 1, failed: 0, succeeded: 1, skipped: 0, duration: 998.6s
```

```
Build succeeded in 999.6s
```

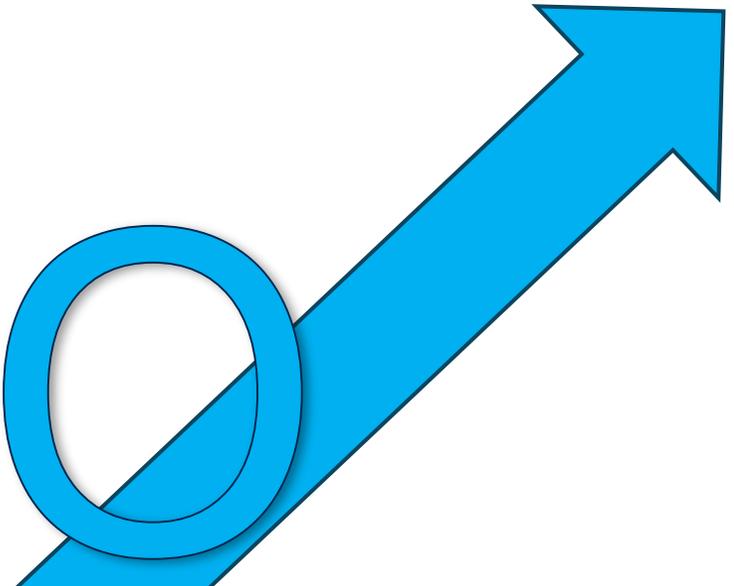
```
ran for 999 seconds
```

```
goat-agent% █
```

Eval Tools



- Stability & Quality
- Microsoft Foundry

DEMO 

Making Agents Work

- ✓ Observable
- ✓ Safer
- ✓ Consistent*ish*
- ☐ Deterministic

Selective Determinism: Right Tools for the Job

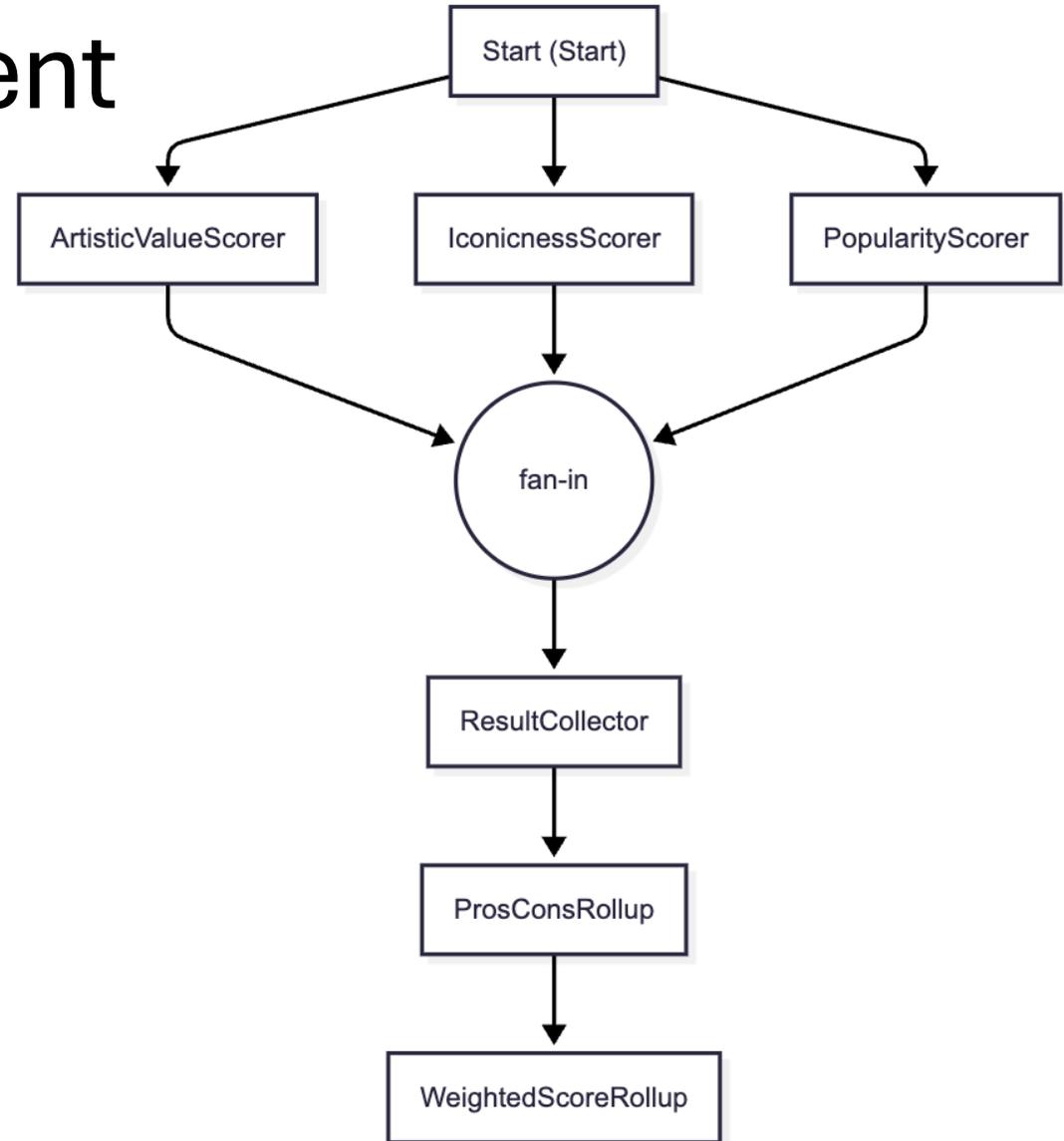
LLMs are
non-
deterministic

But code is
deterministic

Sometimes Code is Right Tool for the Job

- Old school - when there's a right answer
- Could be an optimization (Magentic-One/Orchestrator for prototype, Selective Determinism for production)
- Calculations
- Fixed Sequences (use a defined workflow)

Recall the Movie Rater Agent



Making Agents Work

- ✓ Observable
- ✓ Safer
- ✓ Consistent^{*ish*}
- ✓ Deterministic

Check out
AgentFramework.dev

See you at
Boston Azure 🤖 **AI** 🤖
bostonazureai.org
@bostonazureai
for VIRTUAL & IN-PERSON EVENTS



Connect with Bill

Bill Wilder
@codingoutloud

blog.codingoutloud.com

linkedin.com/in/billwilder



