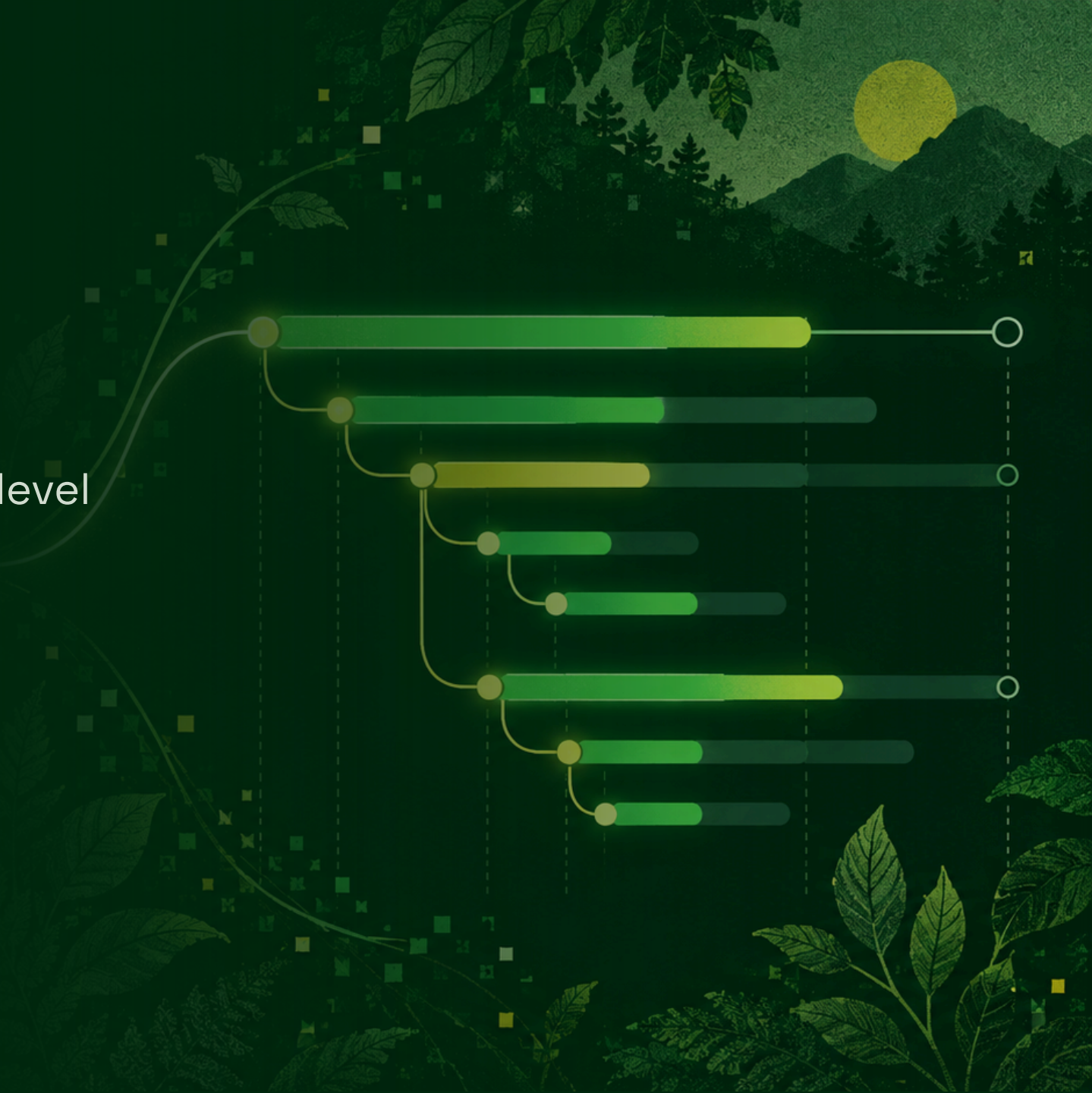


I know OTel

Giving your coding agents expert-level
OpenTelemetry knowledge



Speed At Odds With Quality

- Code generation is fast
 - AI writes code faster than we can read it
- But it is also probabilistic
 - If we can't read it, we can't verify its correctness
 - LLMs are not good at testing software
- **Human comprehension** to balance speed and quality
 - Our ability to understand and make meaning from information


CODE
EUROPE

Earlier today at Code Europe, Warsaw

The trust test

Where we are going

01	02	03	04	05
The hidden failure	The knowledge layer	The comparison	The demonstration	The mental model

A change in software creation

■ Agents already write most code in some teams

ANTHROPIC

>80%

of production lines merged were attributed to Claude

Anthropic · May 2026

OPENAI PRODUCT TEAM

Every line

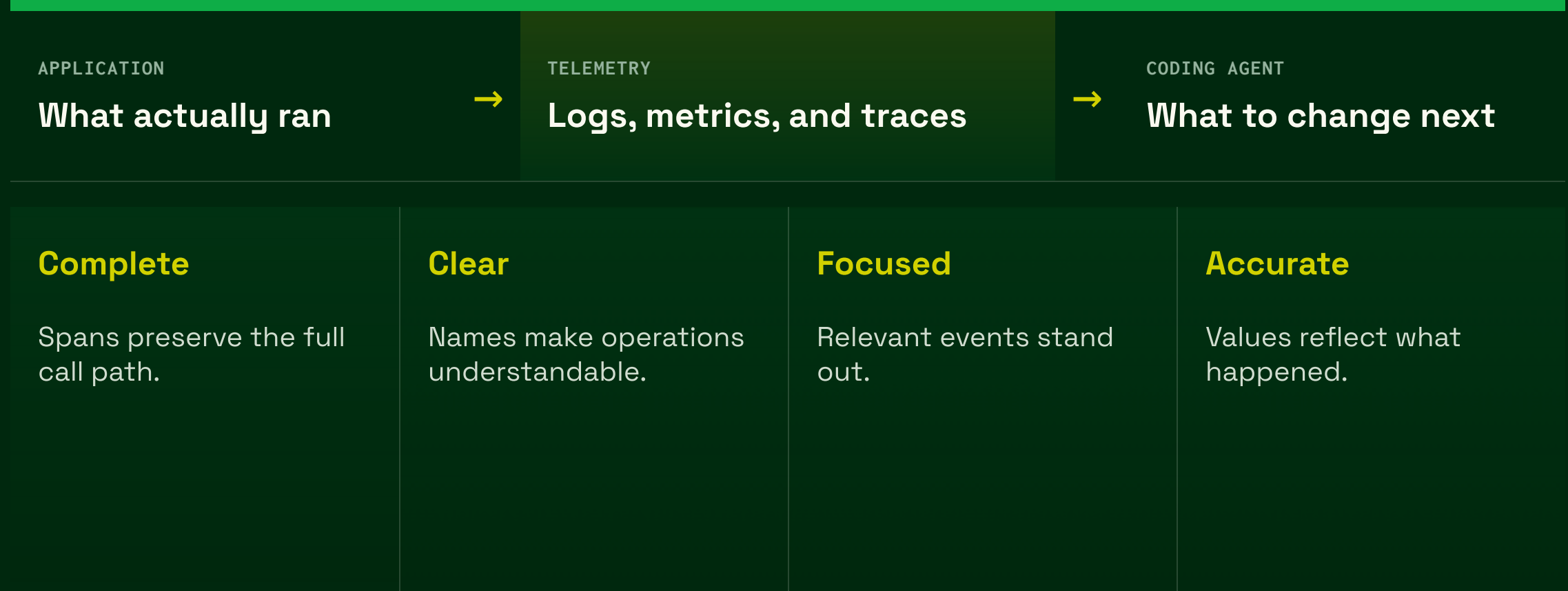
of one product's code and observability was written by Codex

OpenAI · February 2026

Human engineers steer and review. Agents implement changes that shape runtime behavior.

The next reader

■ Agents need telemetry they can trust



The agent's next decision depends on the story telemetry tells.

The code worked. The telemetry lied.

The pull request compiled, passed its tests, and produced traces. It was still wrong.

■ ■ Working is not trustworthy

CI / build

PASS

go test ./... ✓

otelc build ✓

service start ✓

What the build cannot tell you

- ✗ Dependencies came from 2024
- ✗ HTTP and database spans split apart
- ✗ Span names ignored current conventions
- ✗ Sensitive values entered the pipeline

01

Models learn yesterday's OpenTelemetry

The problem starts before the prompt.



■ ■ What six models remembered

MODEL	RECALLED COLLECTOR	DRIFT
GPT-5.6 Sol Pro	v0.103.0	57 releases
Gemini 3.7 Flash	v0.118.0	42 releases
Grok 4.6	Declined to guess	n/a
Claude Opus 5	v0.120.0	40 releases
GLM 5.3	about v0.128.0	32 releases
DeepSeek V4 Pro	about v0.120.x	about 40

Verified September 7

v0.160.0

Released September 2,
2026

■ ■ The stale answer still starts

March guidance

processors:

~~logdedup:~~

deprecated alias

Released name

processors:

log_dedup:

canonical name

v0.160.0 starts the legacy name with a deprecation warning.

■ ■ Four failures hide behind green builds

01	02	03	04
Stale versions	Wrong conventions	Broken context	Leaked values
Old packages compile while missing current fixes and capabilities.	Plausible attribute names break standard queries and dashboards.	Valid spans become disconnected traces at service boundaries.	Useful-looking attributes expose personal or sensitive information.

A longer prompt is not a **knowledge source.**

More instructions cannot supply facts the model does not have.

02

Give the agent a maintained knowledge layer

Current facts enter when the task needs them.



■ ■ What an Agent Skill provides

ENTRY

A small instruction surface for one domain

REFERENCES

Focused material selected for a specific task

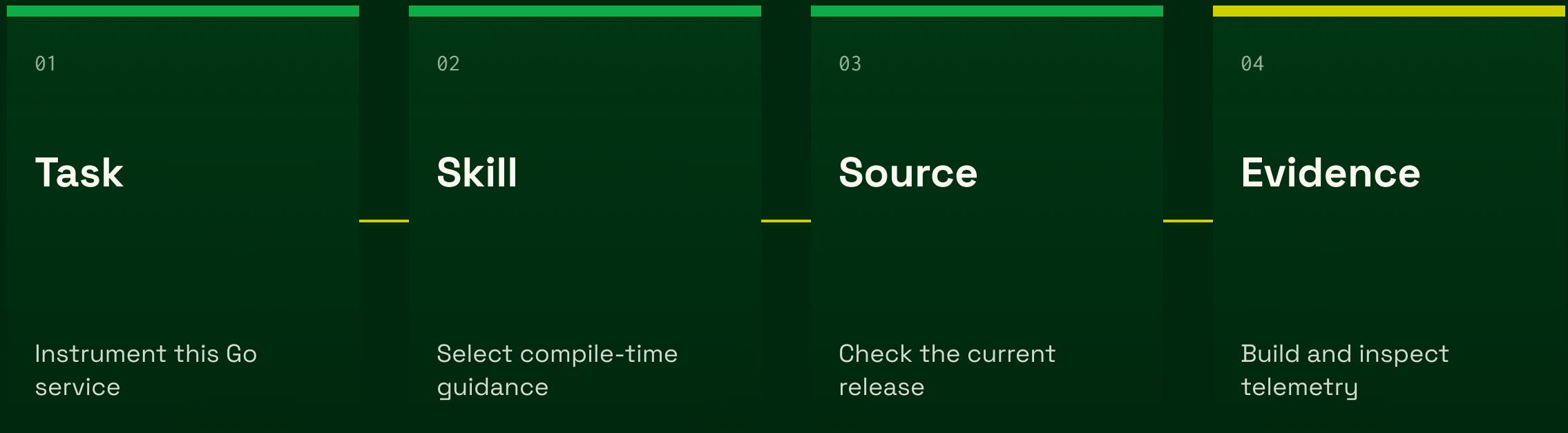
TOOLS

Precise lookups against upstream information

WORKFLOW

Repeatable steps with explicit verification

■ ■ Retrieve only what the task needs



Small retrieval keeps context focused and makes the recommendation traceable.

■ ■ Eighteen skills cover the working surface

LANGUAGES

Go · Java · Node.js
Python · .NET · Browser

PIPELINE

Collector · OTTL
Telemetry generation

CORRECTNESS

SDK versions · Conventions
Breaking changes

EMERGING

Weaver · Declarative config
Span-event migration

■ ■ Install the skills

CODEX, CURSOR, AND OTHER AGENTS

```
npx skills add ollygarden/opentelemetry-agent-skills
```

Run from your project root. Select your agent and scope when prompted. Add `--skill otel-go` to install only today's Go skill.

CLAUDE CODE

```
/plugin marketplace add ollygarden/opentelemetry-agent-skills  
/plugin install otel-go@opentelemetry-agent-skills
```

Register the marketplace, then install the skill you need.



Full guide and skill list

[github.com/ollygarden/
opentelemetry-agent-skills](https://github.com/ollygarden/opentelemetry-agent-skills)

03

Hold everything else constant

Same repository. Same model. Same task.



■ ■ One task, two knowledge conditions

THE TASK

Instrument this Go + Gin application with OpenTelemetry Go Compile-Time Instrumentation. Produce a runnable result.

BARE

Model knowledge

Web and terminal tools

SAME

repository
model
prompt

SKILLED

Model + otel-go

Web, terminal, and skill reader

■ ■ Compare outcomes, not confidence

CHECK				
			BARE	SKILLED
otelc version			v1.1.0	v1.1.0
Tests + build			PASS	PASS
HTTP trace captured			YES	YES
Database spans captured			NOT CHECKED	YES
Repeatable workflow			Makefile + docs	Tool pin + script + CI
			11:07	9:46
			Both produced valid instrumented builds.	

■ ■ Better context cost less in this run

BARE	WITH OTEL-GO	
1,910,954 total token traffic	1,160,192 total token traffic	39.3% fewer tokens 750,762 fewer

Web discovery calls fell from 8 to 3. Tool calls fell from 47 to 34.

■ ■ Verification happens outside the model

01	02	03	04
Clean build	Fixed runtime	One request	Real signals
<code>go test ./...</code>	<code>otelc build ./...</code>	<code>GET /api/profiles/:username</code>	<code>trace · metrics · logs</code>

The model does not grade its own work.

04

Now inspect what actually runs

The backend gets the final vote.



■ ■ Context propagation: three checks

01	02	03
Send one profile request	Pass request context	Repeat and inspect
An absent synthetic profile runs one database lookup.	Use the incoming context for the GORM query.	Confirm both spans share one trace ID and parent link.

A captured run follows the same sequence if the live environment fails.

LIVE DEMO

Repair the request context

An expected 404 still runs a database lookup.

```
GET /api/profiles/cedemo2026missing
```

```
HTTP 404 · expected lookup miss
```

```
Gin → SELECT user_models
```

```
after WithContext: same trace ID
```

■ ■ The same request becomes one trace

GIN SERVER

GET /api/profiles/:username

GORM QUERY

SELECT users

1

request

1

trace

0

orphans

Gin span → SELECT child span

Supply current facts, then **verify reality.**

Skills improve the starting point. Running telemetry decides whether the result is trustworthy.

■ ■ Try one task next week

01	02	03
Install the skills Add the open source repository to your coding agent.	Repeat one task Run the same OpenTelemetry request with and without skills.	Inspect the telemetry Judge traces, metrics, and logs in a real backend.

github.com/ollygarden/opentelemetry-agent-skills



Obrigado!

Juraci Paixão Kröhling

Founder and CEO, OllyGarden

[github.com/ollygarden/
opentelemetry-agent-skills](https://github.com/ollygarden/opentelemetry-agent-skills)

talks.kroehling.de



linkedin.com/in/jpkroehling