

Platforms

A brief history, what they are, how to run them, and getting ready for AI

Michael Côté

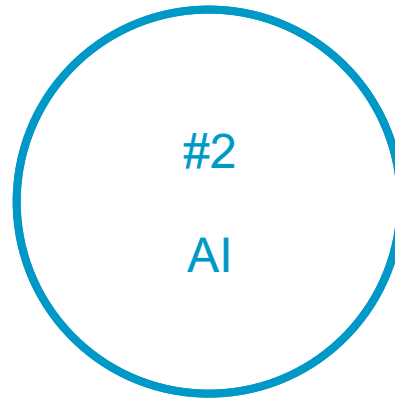
Tanzu Division, Broadcom

15 September 2026

Everyone needs an application platform



Software
Runs
Everything



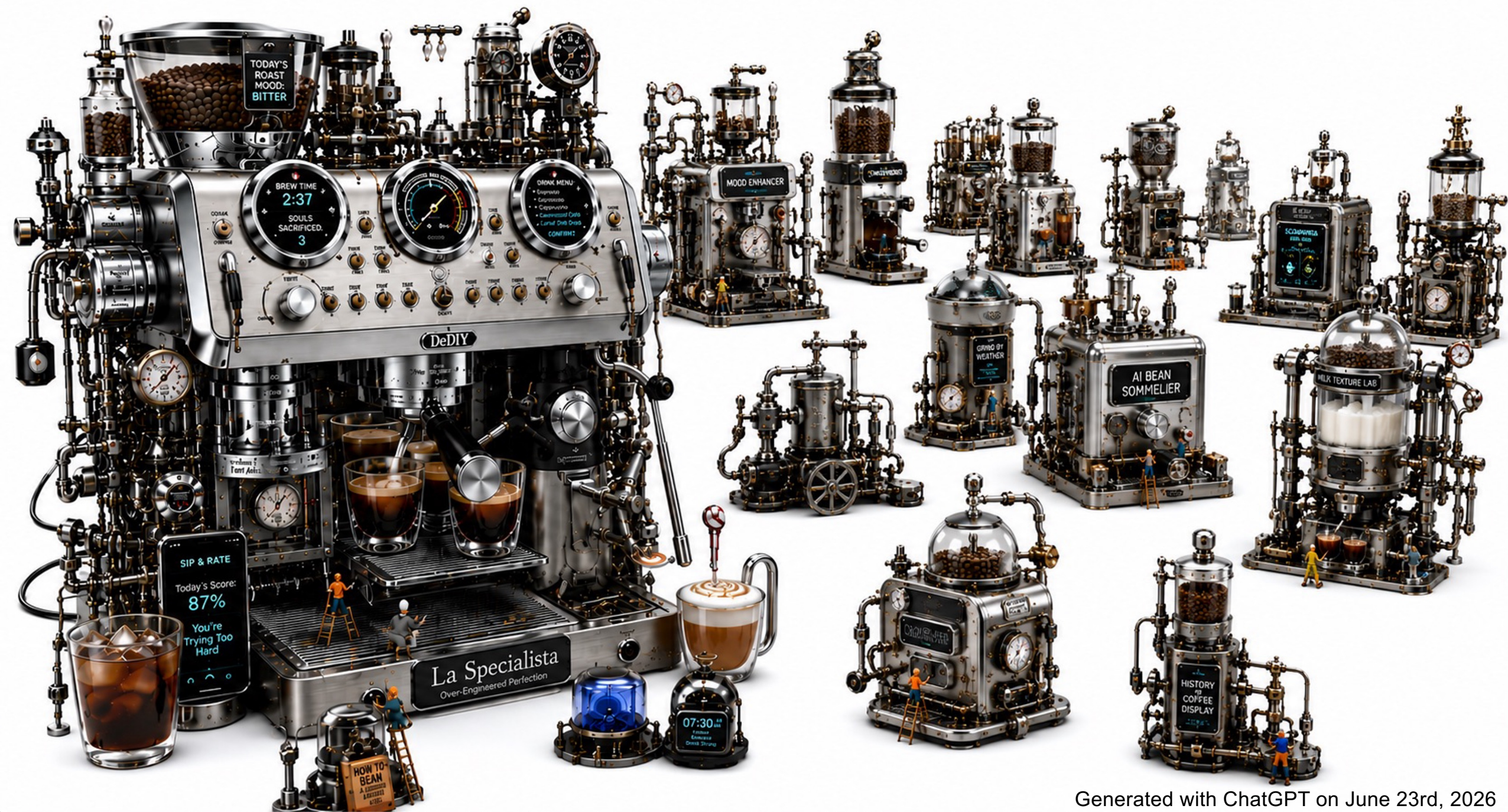
The new app

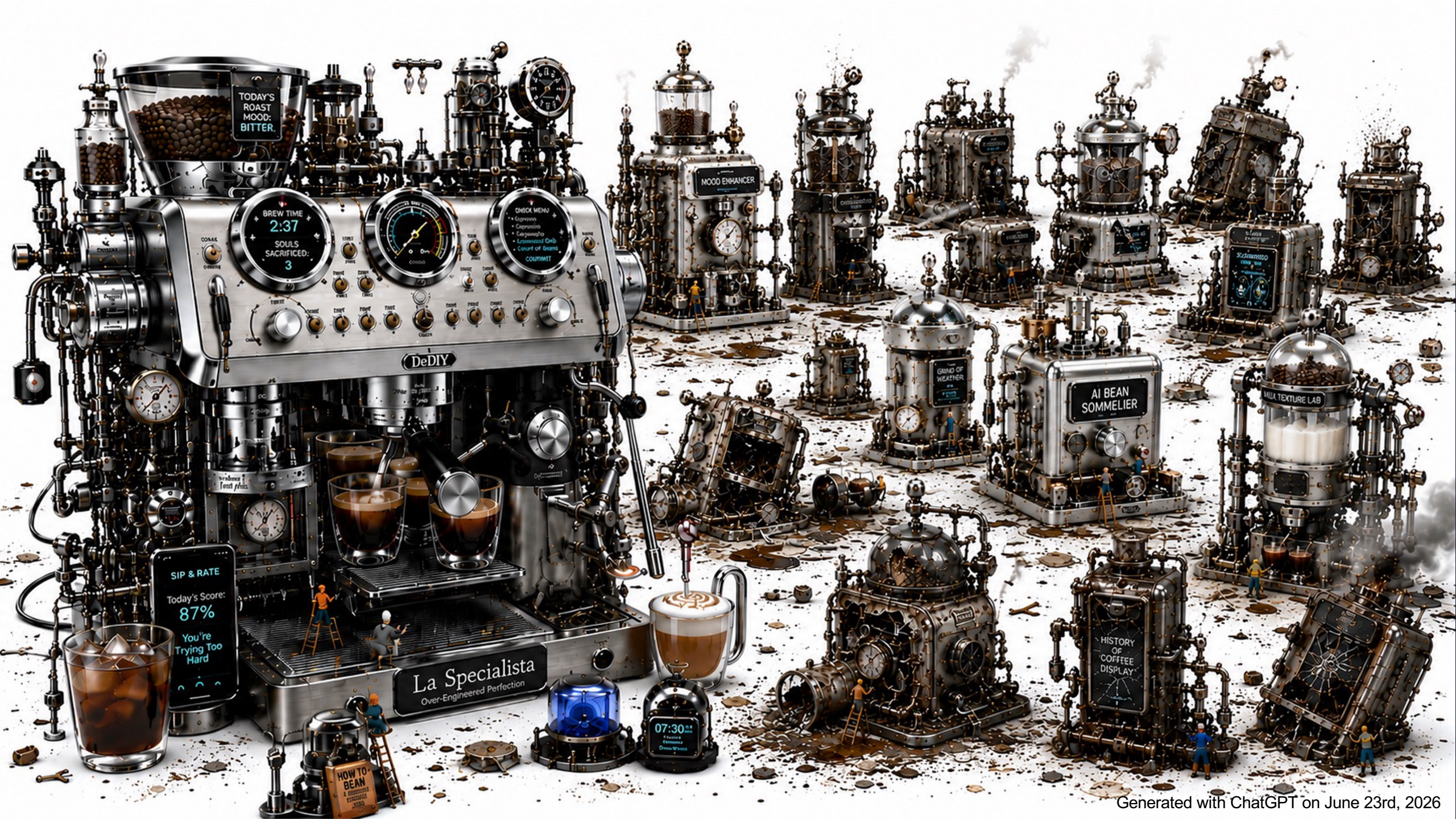


Cat Picture Grade
Security Is Not
Good Enough



#4
You
already
have one!

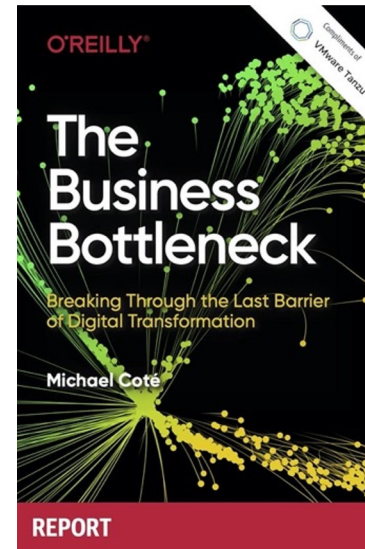
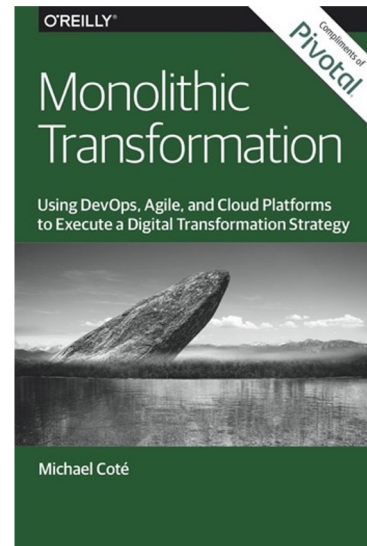




- 350 apps / 7 ops
- 300 apps / 8 ops
- 30,000 devs / 50 ops
- 6,500 devs/16 ops
- 2,500 devs / 5 ops
- 1,200 devs / 6 ops
- 45 app teams / 5 ops
- 300 app teams/ 4 ops

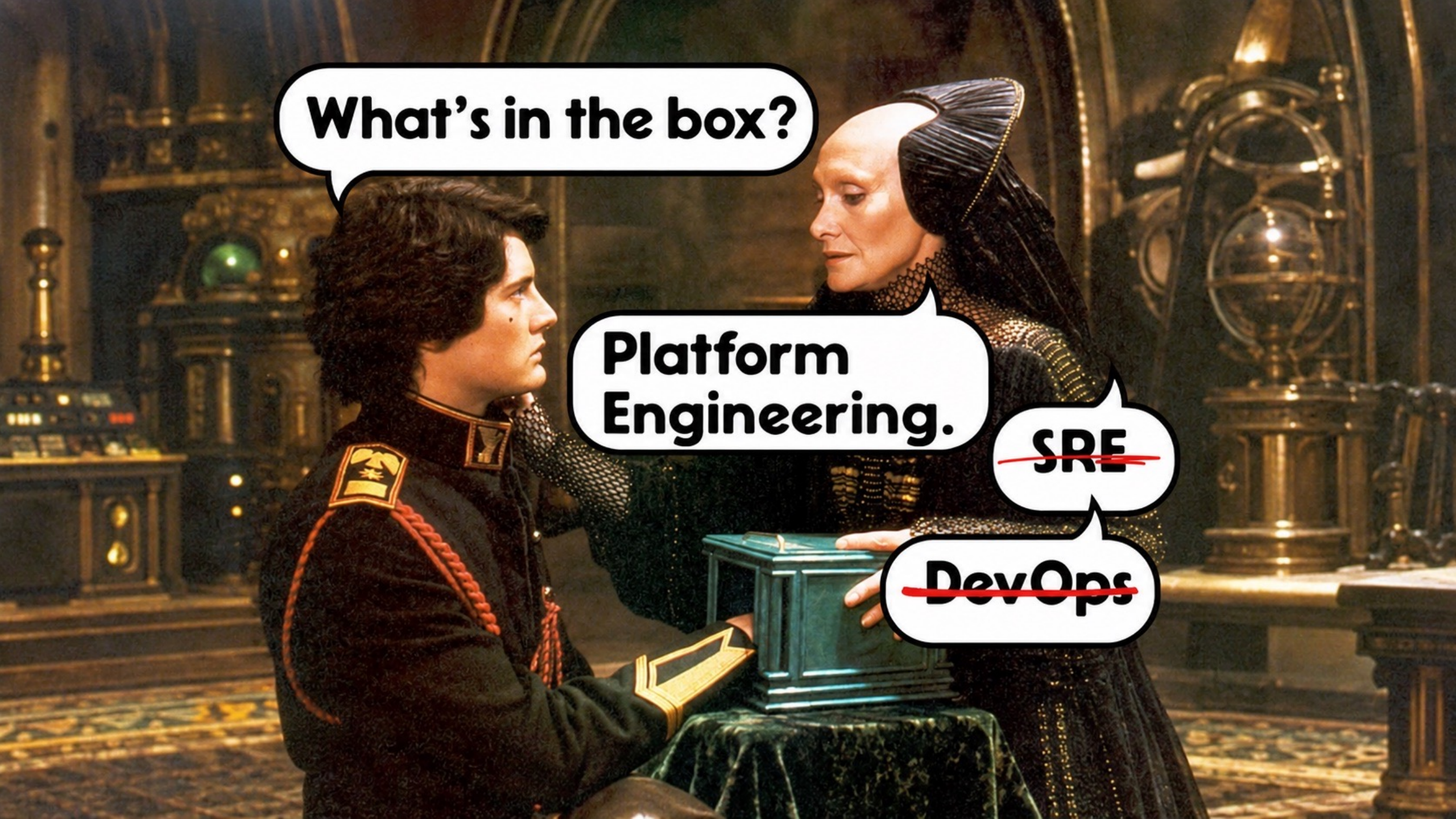
Coté

<https://www.cote.io/> | cote@broadcom.com



Tanzu
Catsup





What's in the box?

**Platform
Engineering.**

~~**SRE**~~

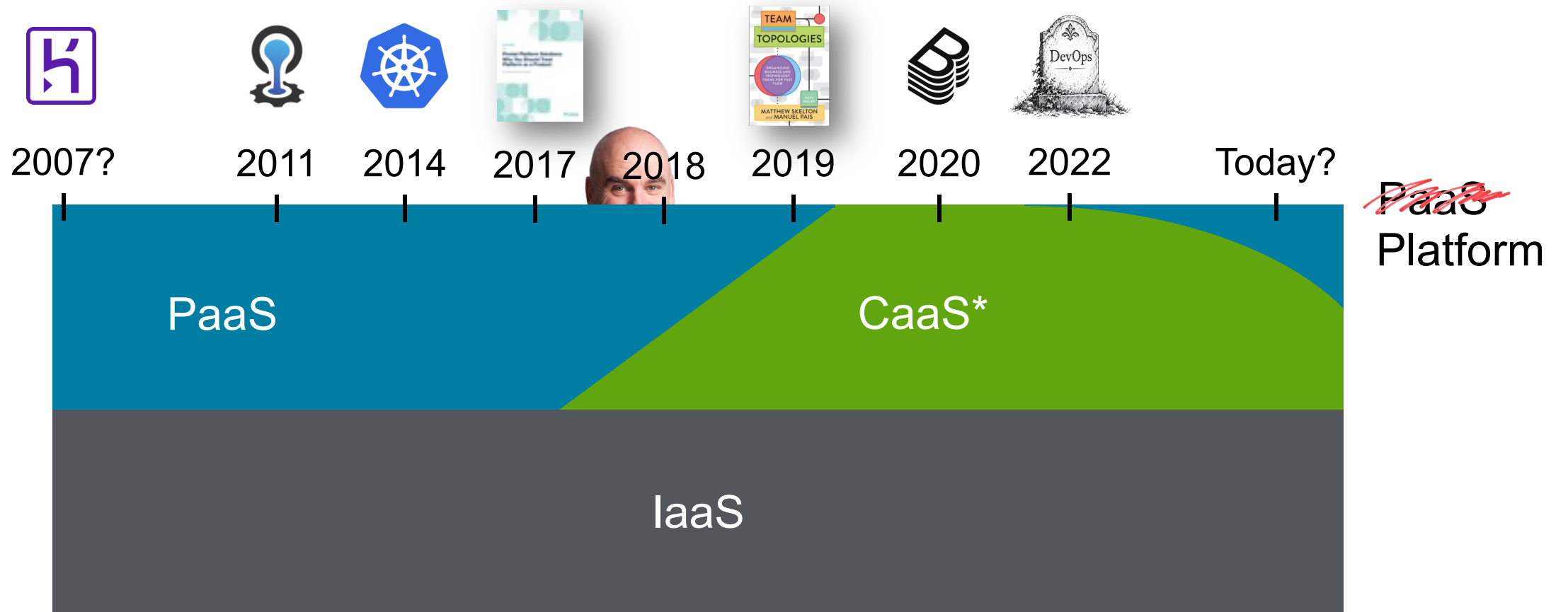
~~**DevOps**~~

“A *digital platform* is a foundation of **self-service APIs**, tools, services, knowledge and support which are arranged as **a compelling internal product**.

“[SO THAT] Autonomous delivery teams can make use of the platform to **deliver product features at a higher pace, with reduced coordination**.”

Evan Bottcher, March, 2018

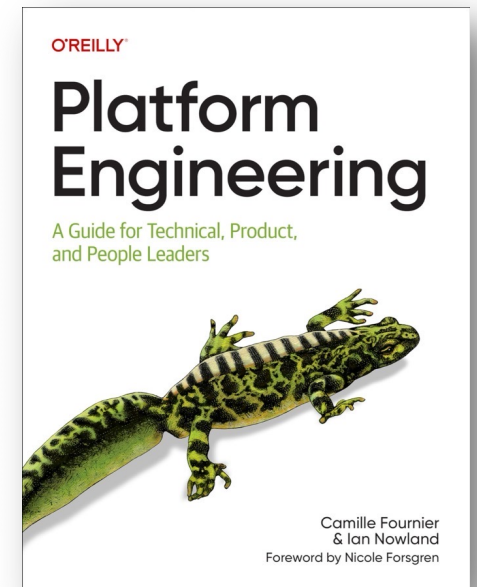




* Code word for "Kubernetes."

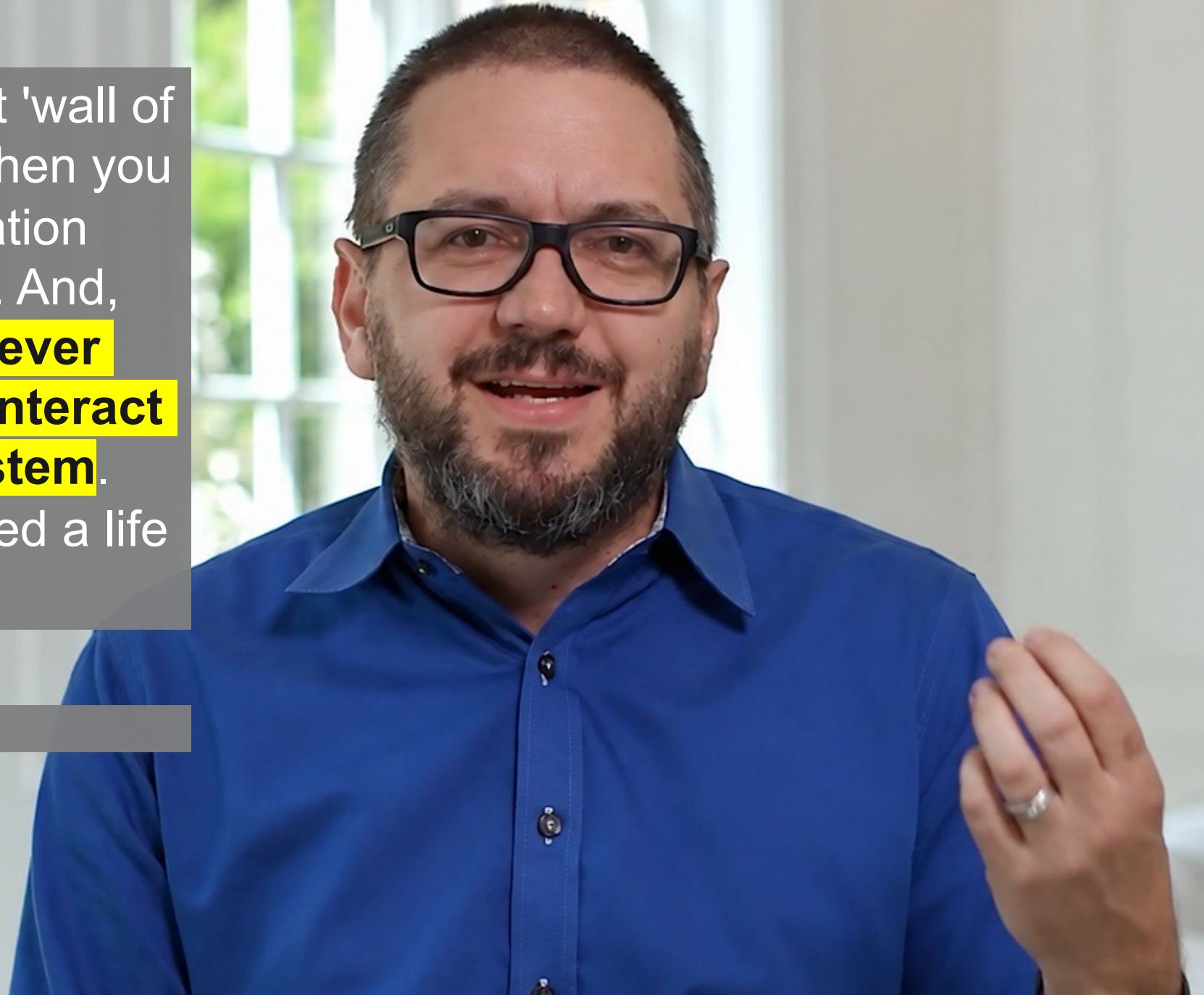
“Platform engineering is the discipline of developing and operating platforms. The goal of this discipline is to manage overall system complexity in order to deliver leverage to the business. It does this by taking a curated product approach to developing platforms as software-based abstractions that serve a broad base of application developers, operating them as foundations of the business.”

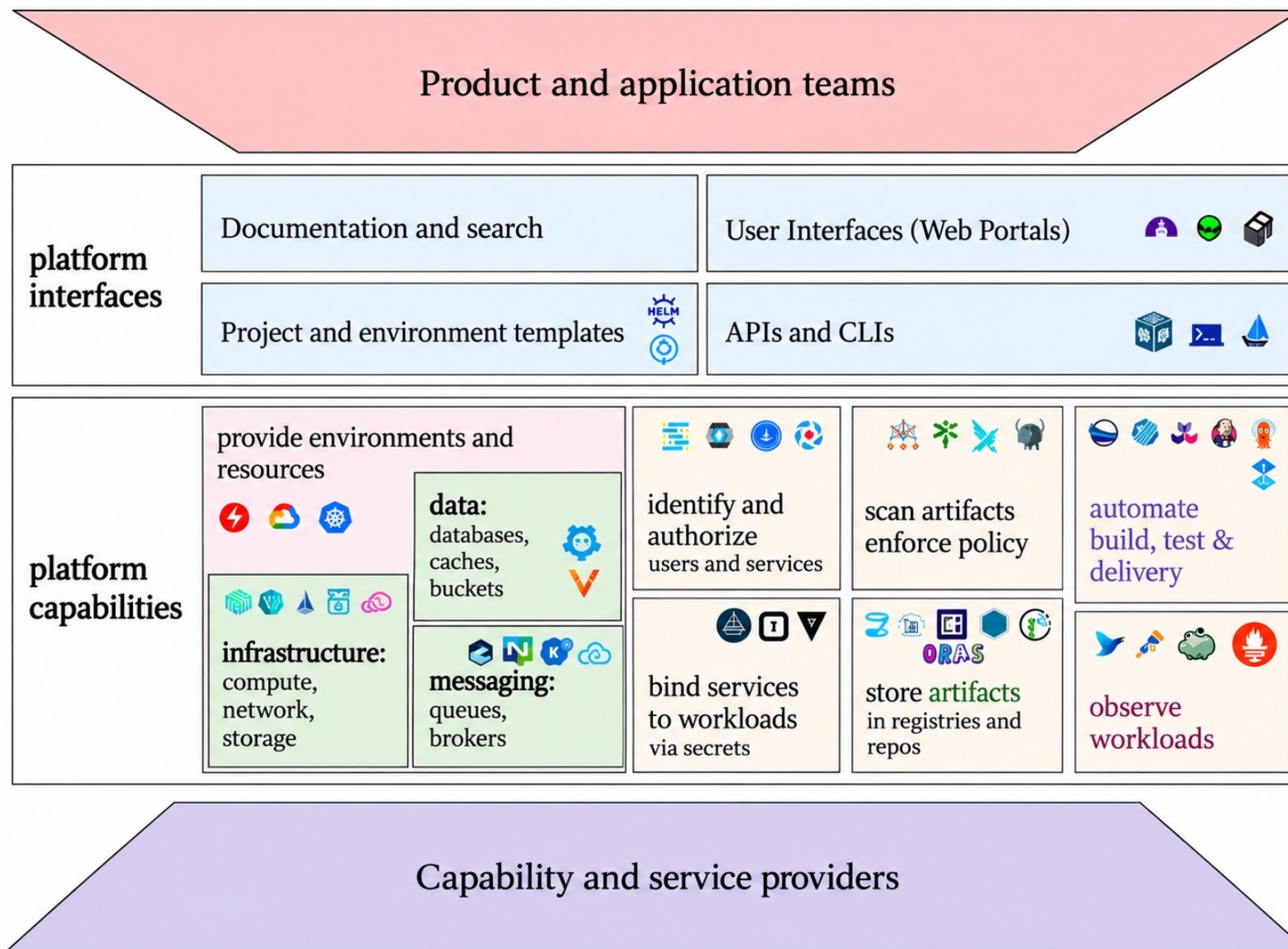
Platform Engineering, Camille Fournier and Ian Nowlan,
November 2024.

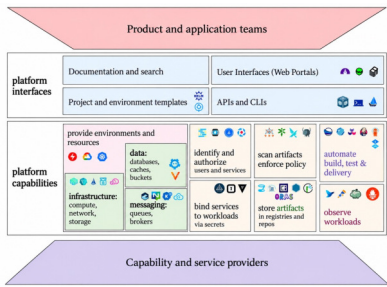


“The initial experience, that 'wall of yaml,' as we like to say, when you configure your first application can be a little bit daunting. And, I'm sorry about that. **We never really intended folks to interact directly with that subsystem.** It's, more or less, developed a life of its own over time.”

Craig McLuckie, SpringOne 2021







Modern applications

GenAI applications

Agentic workflows



VMware Tanzu Platform



Developer Services



Build Services



Data Services



AI Services

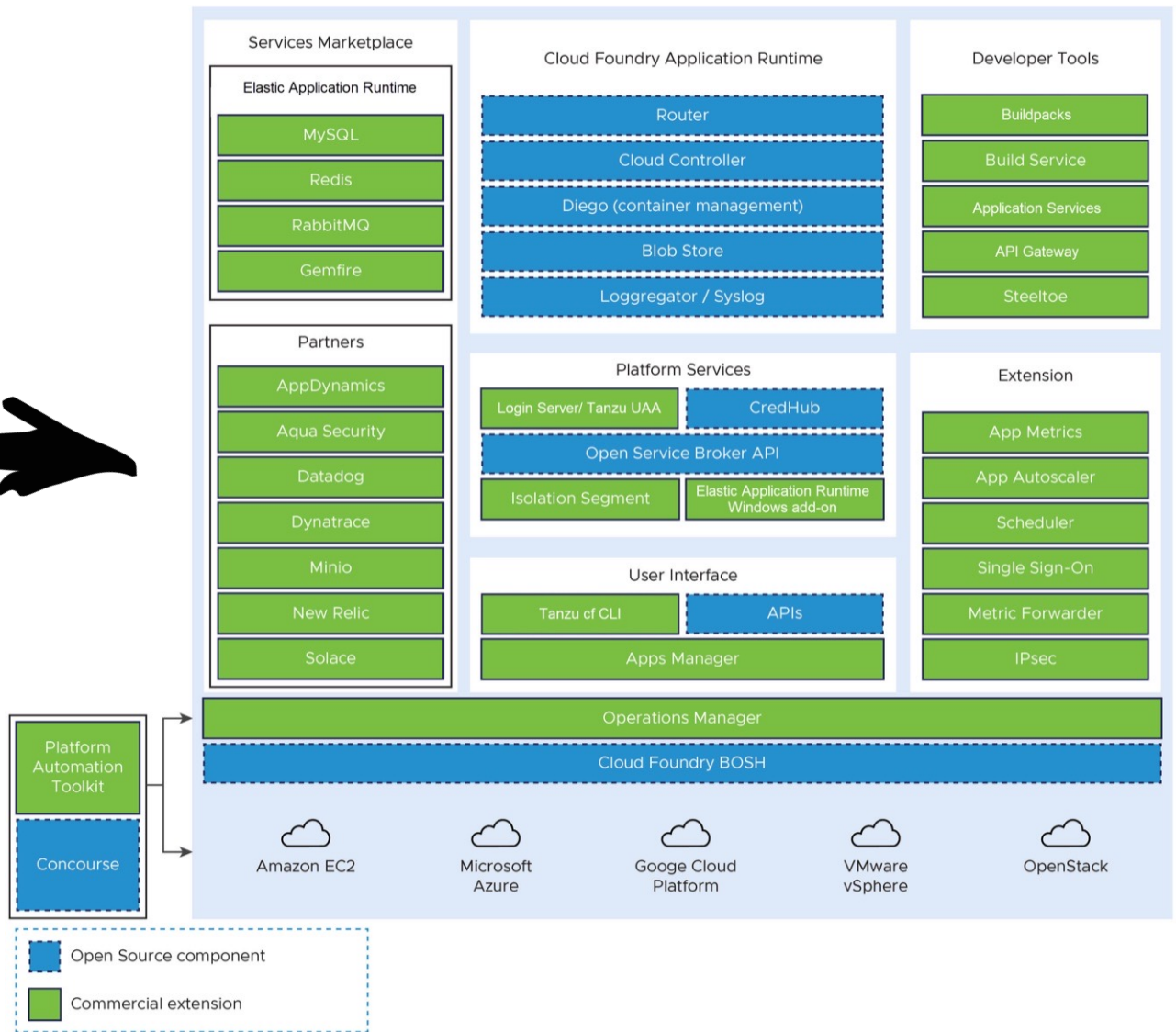
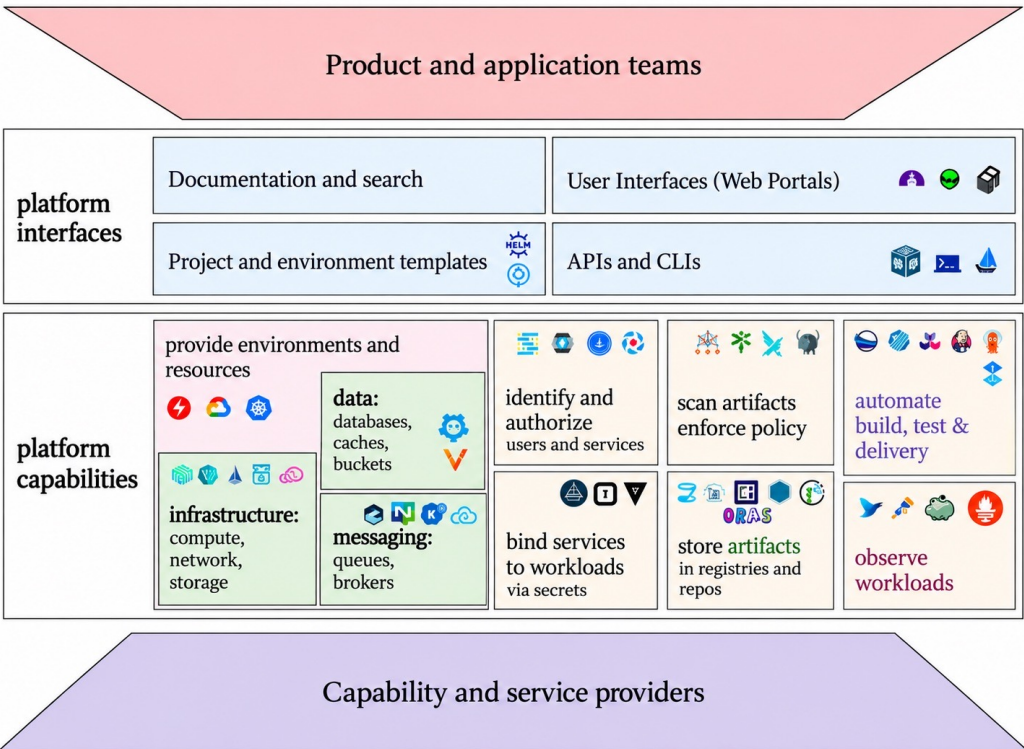


Marketplace Services



Elastic App Runtime

Private Cloud (VMware Cloud Foundation) | Public Cloud



Running a platform

“ We are building this platform not for us, we are building it for Mercedes-Benz developers.”

Thomas Müller, Mercedes-Benz



Find the Developer Toil, Confusion, Blockers

Find the Developer Toil, Confusion, Blockers

- What are we making?
- We have a strong vision for our product, and we're doing important work together every day to fulfill that vision.
- I have the context I need to confidently make changes while I'm working.
- I am proud of the work I have delivered so far for our product.
- I am learning things that I look forward to applying to future products.
- My workstation seems to disappear out from under me while I'm working.
- It's easy to get my workstation into the state I need to develop our product.
- What aspect of our workstation setup is painful?
- It's easy to run our software on my workstation while I'm developing it.
- I can boot our software up into the state I need with minimal effort.
- What aspect of running our software locally is painful? What could we do to make it less painful?
- It's easy to run our test suites and to author new ones.
- Tests are a stable, reliable, seamless part of my workflow.
- Test failures give me the feedback I need on the code I am writing.
- What aspect of production support is painful?
- We collaborate well with the teams whose software we integrate with.
- When necessary, it is within my power to request timely changes from other teams.
- I have the resources I need to test and code confidently against other teams' integration points.
- What aspect of integrating with other teams is painful?
- I'm rarely impacted by breaking changes from other tracks of work.
- We almost always catch broken tests and code before they're merged in.
- What aspect of committing changes is painful?
- Our release process (CI/CD) from source control to our story acceptance environment is fully automated.
- If the release process (CI/CD) fails, I'm confident something is truly wrong, and I know I'll be able to track down the problem.
- What aspect of our release process (CI/CD) is painful?
- Our team releases new versions of our software as often as the business needs us to.
- We are meeting our service-level agreements with a minimum of unplanned work.
- When something is wrong in production, we reproduce and solve the problem in a lower environment.

What developers actually want & need

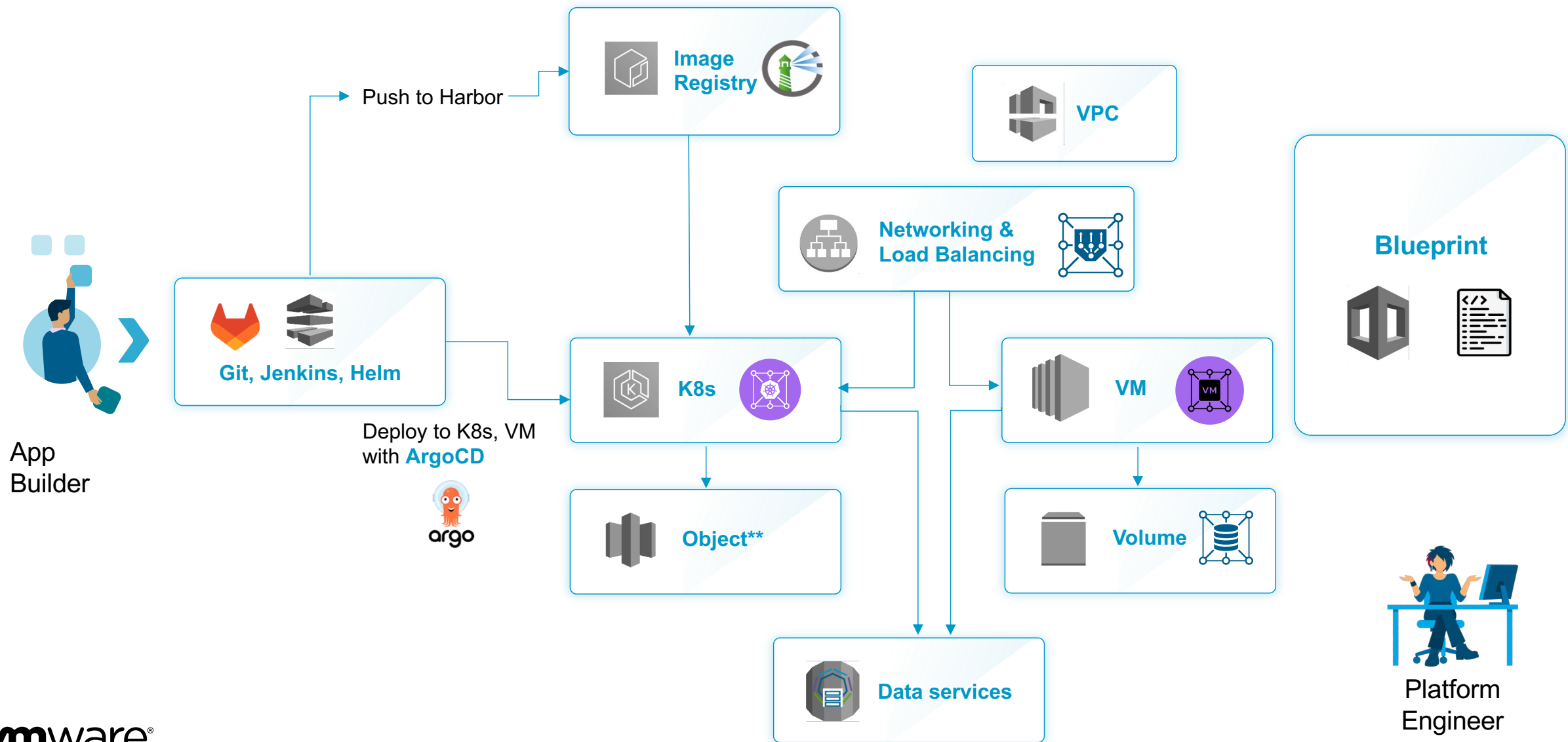
...and what you can (easily) give them



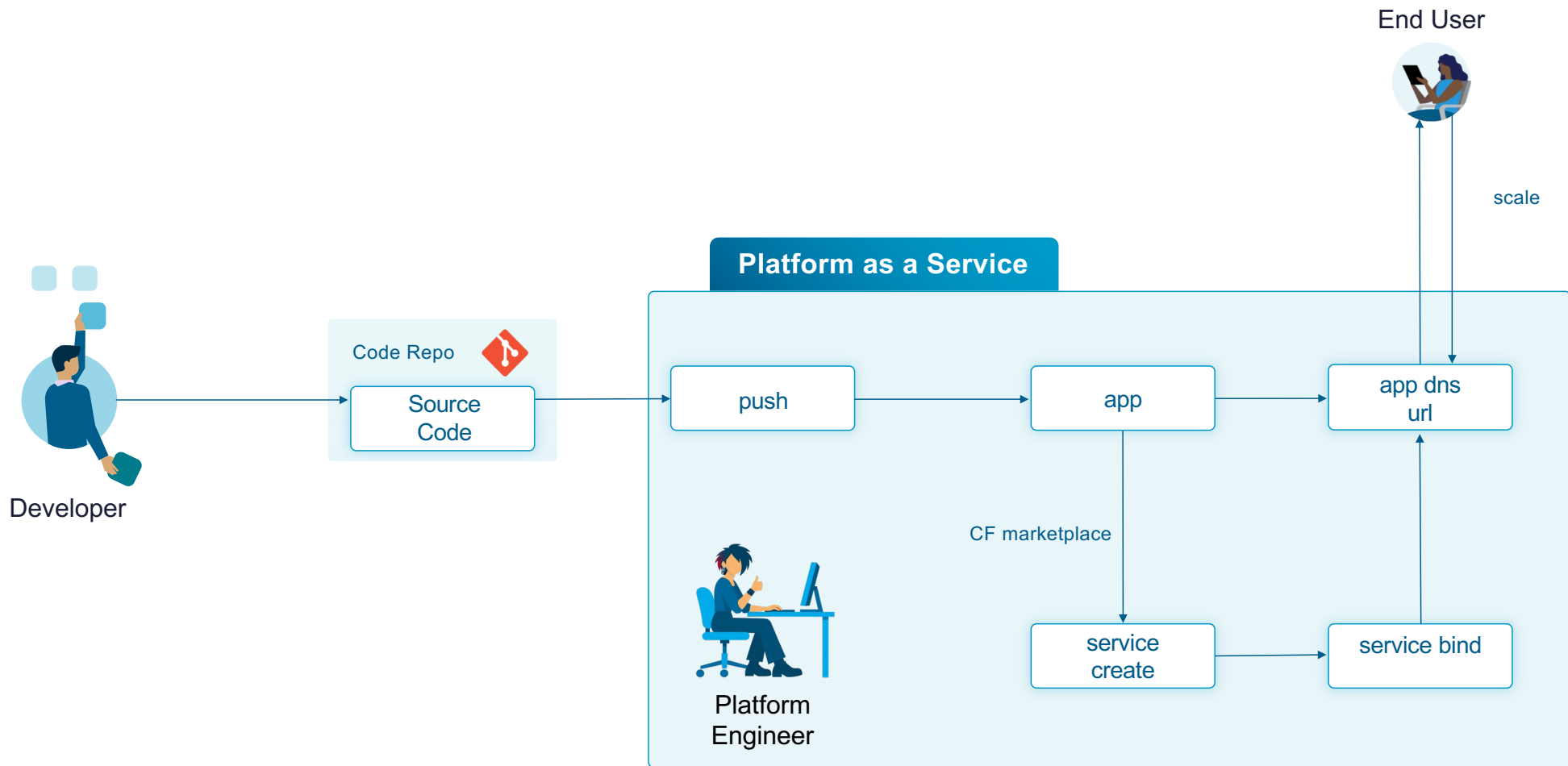
* Very scientifically representative selection of developers...

- Self-service for everything.
- Bring your own framework.
- Easy access to data and service.
- SDLC integrations.
- Simple command to push to production.
- Use new technologies, e.g., AI.
- Lazy security, compliance, governance.

CaaS Path to Production



PaaS Path to Production



A platform's main goal is reducing enterprise “cognitive load”

Shhh...don't tell the developers...platforms help ops people the most...

What ops people actually want & need

- Self-service - not be a ticket queue.
- Standardization - one way to run it, however it was written.
- Catalog - to know what's actually running.
- Deploy the app, not start multi-party negotiations.
- A simple command to patch production.
- Add a new technology once, not per team.
- Pass audits, be compliant, and secure.
- Stay relevant and needed.
- Avoid working in a swamp...and pagers.

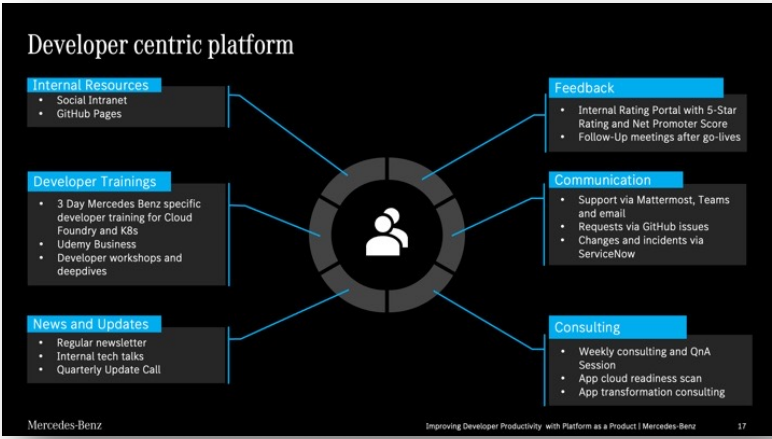
Our Scale Problems (non-regulatory)

Balance & Cog Load	Variety & Legacy	Communications	Coming AI Onslaught
- Needs of the day - Scale theme of balancing quality, speed, and cost - Differing BU missions and dev team needs - Dev Cognitive Load solved long ago by Organizational Cognitive Load - Org Cog Load is much harder Change Management of Org	- Decades of M&A with unintegrated Infra - Many BUs/CIOs/teams - On-prem DCs & Cloud is all 3 major CSPs - Variety of workload targets; containers for modern teams, but still Linux/Window VMs, and big & mid iron - We've always had an IDP... built on tickets and meetings	- Listening to 16k technologists; half of whom write code (1:1) - Team is from Infra backgrounds; empathy for DevEx challenging - Regular feedback & engagement w/Devs - Diff norms for diff dev teams; fewer pervasive problems to solve - Matching ever increased need of new cybersecurity demands	- Dev (and infra) teams leveraging AI faster & more broadly, more in last 10-months - AI adoption velocity++ - Demand for "Agentic" - Mythos (and other augmented vuln finding) hits past processes hits the legacy & modern, but one is ready

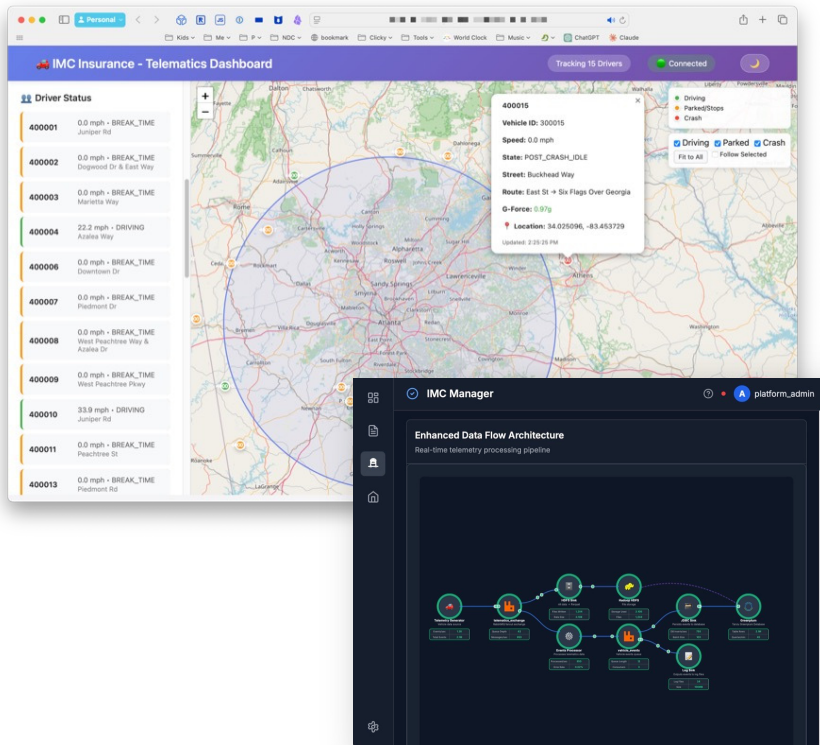
Jay Moran
SVP of Platform Engineering, Fiserv

PLATFORM CON LIVE DAY NEW YORK

Platform marketing & community management



But what about AI?



```

Claude Code v2.1.77
Welcome back Coté!

Opus 4.6 (1M context) · Claude Max · Michael Coté
~/dev/chatdm

Tips for getting started
Run /init to create a CLAUDE.md file with instructio...
Recent activity
No recent activity

) Tell me how we could add AI image generation to this project

• Explore (Explore ChatDM project structure)
  ↳ Done (28 tool uses · 69.9k tokens · 1m 1s)
  (ctrl+o to expand)

• Cool project. Here's how image generation could fit into ChatDM:

The Short Version

Add a new ImageGenerationTool as an MCP tool that calls an image generation API (most likely OpenAI's DALL-E or Stability AI), then saves the resulting image to ~/.chatdm-mcp/images/. The tool follows the exact same @Service + @Tool pattern as your existing tools.

Concrete Approach

1. New dependency in pom.xml

Two realistic options:

- Spring AI's image model abstraction - Spring AI already has ImageModel support with adapters for OpenAI (DALL-E 3), Stability AI, and others. Since you're already using Spring AI for MCP, this is the natural fit. You'd add something like spring-ai-openai-spring-boot-starter and get a clean ImageModel interface.
- Direct HTTP calls - Hit the OpenAI or Stability API directly with RestClient. Fewer moving parts, but you lose the abstraction.

Given this is also a learning project for Spring AI, the Spring AI route makes more sense.

2. New tool class: src/main/java/io/cote/chatdm/image/ImageGenerationTool.java

@Service
public class ImageGenerationTool {

    private final ImageModel imageModel;
    private final FileDMStoreRepository store;

    @Tool(name = "ChatDM_generateImage",
          description = "Generate an image for a DAO scene, character, map, or item. "
            + "Returns the file path to the saved image.")
    public String generateImage(

```

```

Claude Code v2.1.77
Opus 4.6 (1M context) · Claude Max
~/dev/crisps-demo

) what's up nerd

• Morning. Here's your Tuesday rundown:

Calendar
- 9:30 AM - Weekly supply chain standup
- 11:00 AM - Flavor P&L review with Finance
- 2:00 PM - Q3 packaging vendor shortlist (Linda + procurement)
- 4:00 PM - 1:1 with VP Ops

Pending
- BBQ SKU rationalization - still no decision; regional sell-through data should land tomorrow
- Packaging redesign proofs due back from agency Friday
- Co-packer audit report (Dallas) - waiting on QA sign-off

This week
- Wednesday: Retail buyer meeting with Jumbo regional
- Friday: Q2 cost-of-goods deadline for Finance

The big one is that BBQ data dropping tomorrow - worth a nudge to whoever's pulling it so you're not scrambling before the Jumbo meeting.

) what do you make of the SKU reduction strategy

• Honestly, the cut makes sense directionally. The math is pretty clear - bottom 70% of SKUs doing 12% of revenue but eating 35% of complexity costs. That's a lot of warehouse space and production changeovers for not much return.

But Sales isn't wrong either. Shelf space is zero-sum. You pull 90 SKUs from ah or Jumbo, competitors fill those slots by Thursday. And getting that space back is a brutal fight.

```



AI in apps...

- Sales assistants.
- Sloppy integration.
- Science-ing.
- ???



Programming...

- New & old code.
- SDLC juicing.
- Trad'l "data science."
- Making pptx?



ChatClaw...

- Your own ChatGPT.
- Customer service.
- Better search.
- Chat-as-UI

New things platform engineers will likely do with and for AI

- Curating models, not just hosting them.
- Self-service access to the approved list.
- Gateways and brokers, for models and for MCP.
- Registries: MCP servers, prompts, tools, integrations.
- Identity and scope for agents.
- Sandboxed runtimes, secrets the agent never sees.
- Somewhere to run the flood: environments, data, test estates.
- Application frameworks.
- Data access without data leaving.
- Eval, testing, safety.
- Audit and compliance, on demand.
- Cost attribution at provisioning, not at invoice.
- AIFinOpsBizSREDevOpsEng.

Bonus: Tools for Headless Ops

Thanks!



cote@broadcom.com



TryTanzu.ai



Slides & stuff! 