DevSecOps: Moving Beyond SAST in Pipelines

Jon Jarboe
Developer Advocate

About

Jon Jarboe

- Dev Advocacy @ Accurics
- Development, AppSec, IaC
- DevSecOps enthusiast
- Favorite attack technique: Watering hole

accurics™

About

Accurics

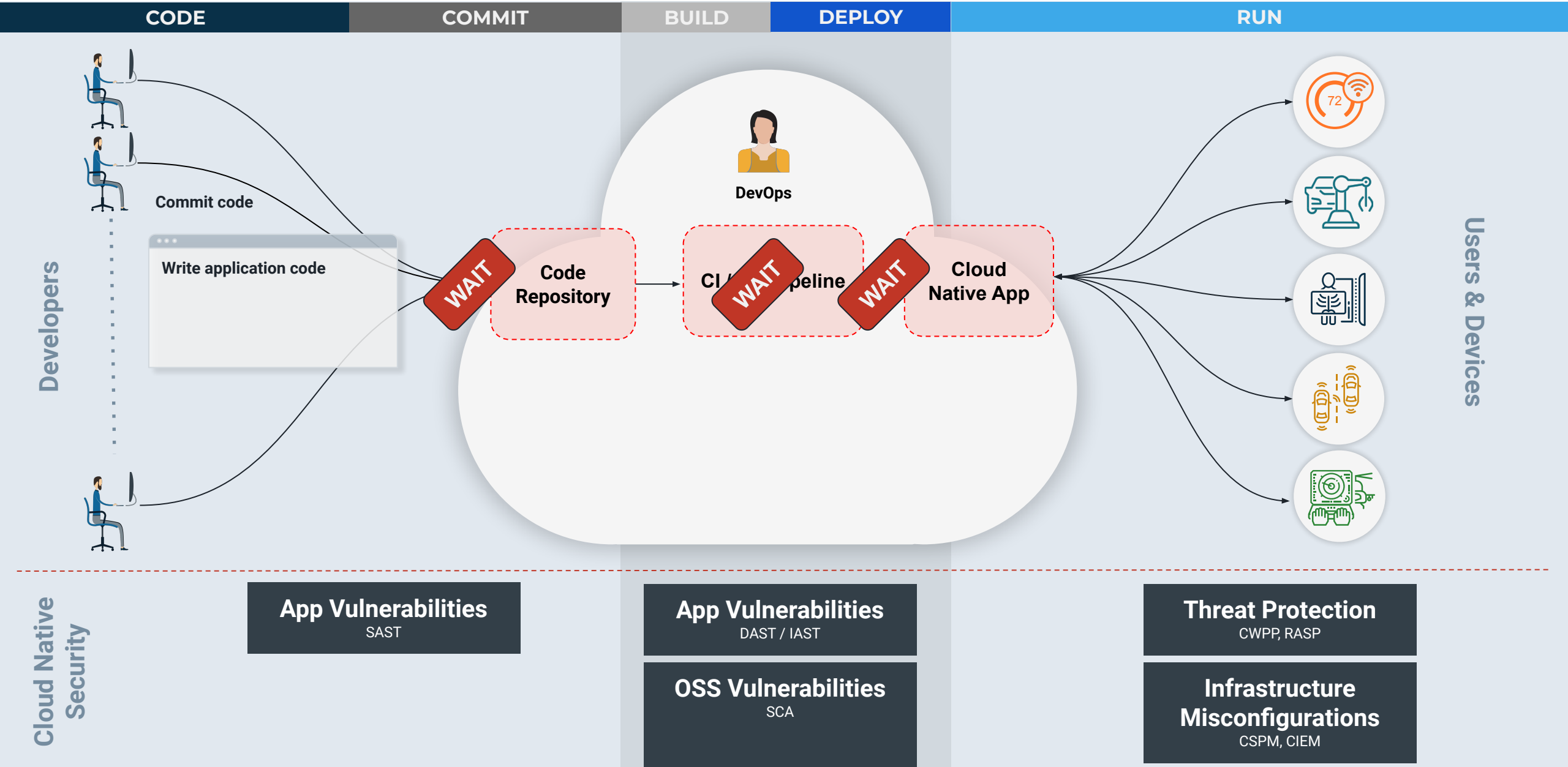
Our mission is to enable cloud cyber resilience with a codified, developer-first approach to security.

We work in automated workflows to self-heal the cloud by enabling developers to programmatically detect and respond to breach paths throughout the development lifecycle.

Agenda: Why DevSecOps?

- DevOps is working!
- Security is not.
- A new approach is needed: DevSecOps.

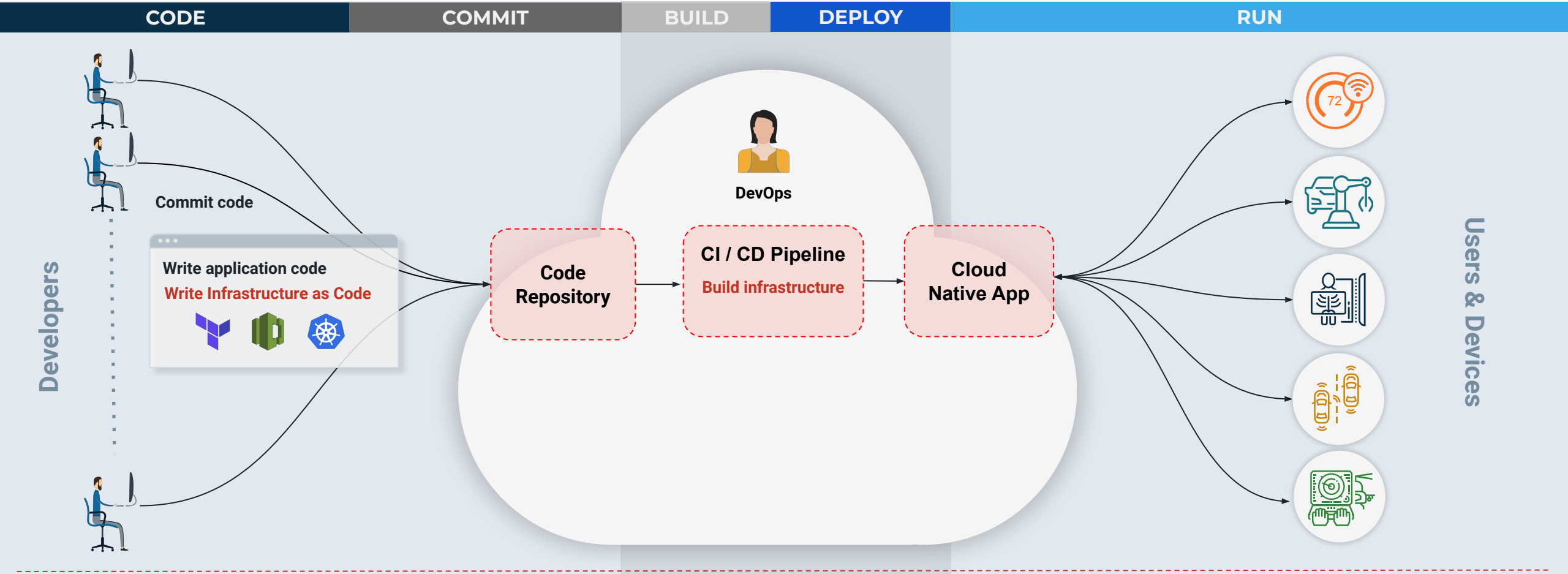
Traditional cloud security tools are **not built for developers**



Automating SAST

- Finds real problems and streamlines some workflows
- Doesn't help with:
  - Manual remediation effort
  - Need for manual review of security findings
  - Risk assessment of findings
  - Prioritization of findings
  - Alert fatigue, or information overload
  - Cultural fit and process inefficiencies

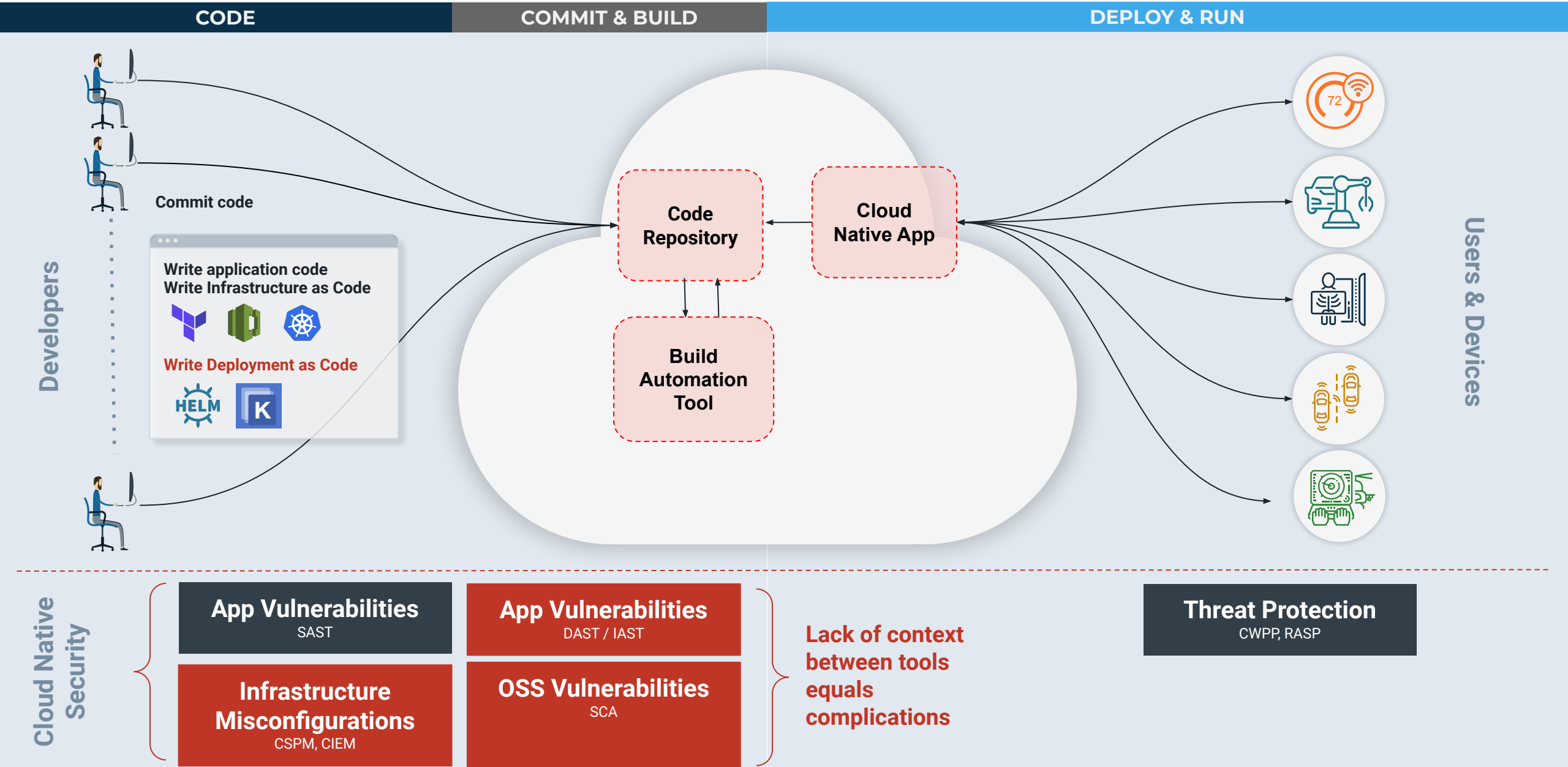
Traditional cloud security tools are **not built for developers**



Cloud Native Security

App Vulnerabilities SAST	**App Vulnerabilities** DAST / IAST	**Threat Protection** CWPP, RASP
Infrastructure Misconfigurations CSPM, CIEM	**OSS Vulnerabilities** SCA	**Infrastructure Misconfigurations** CSPM, CIEM

And security only gets more challenging as **developers embrace <*>Ops**



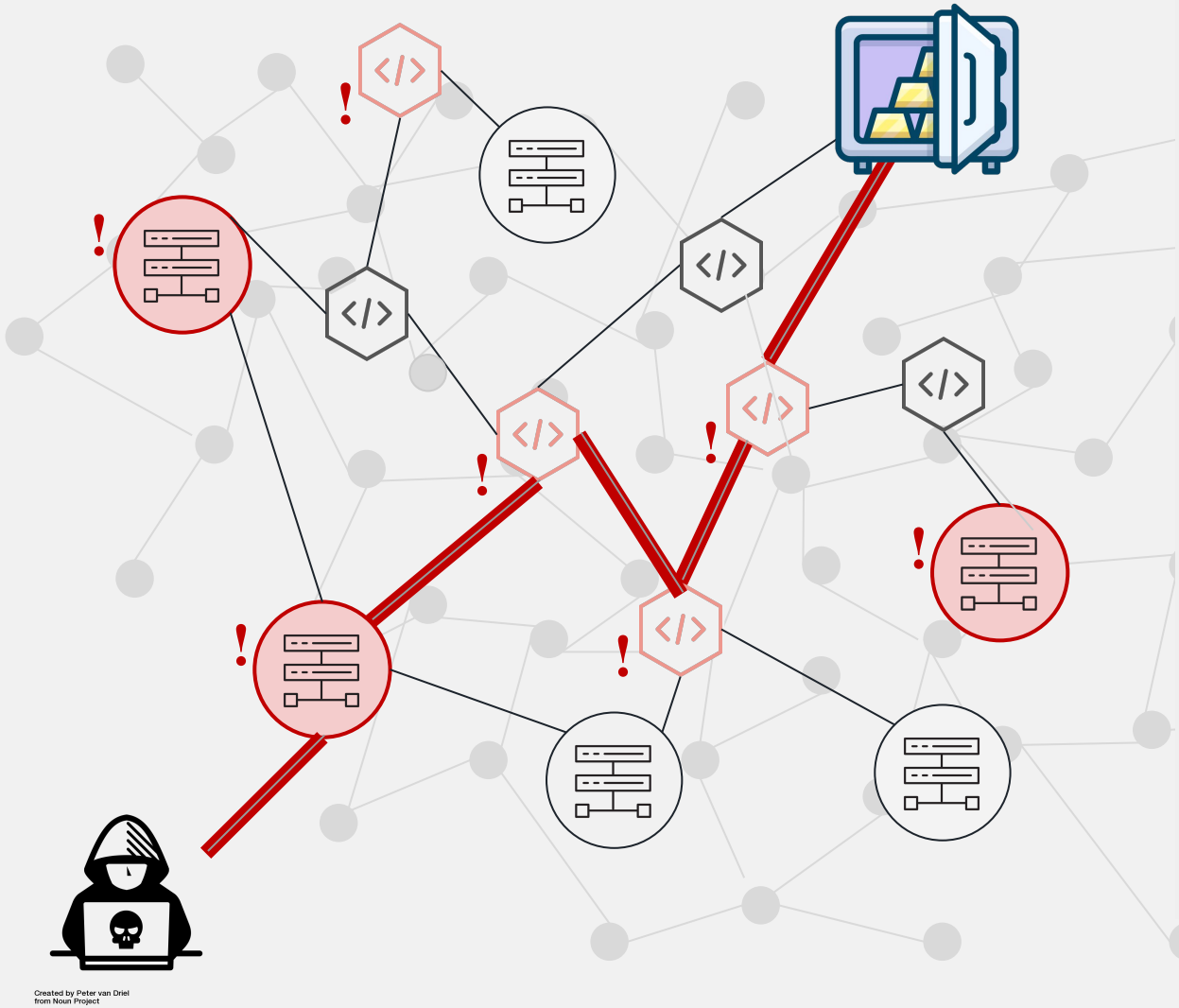
DevSecOps Challenges

- Manual remediation effort
- Need for manual review of security findings
- Risk assessment of findings
- Prioritization of findings
- Alert fatigue, or information overload
- Cultural fit and process inefficiencies

Potential solutions:

- **Automated fixes**
- **Improved accuracy**
- **Risk scoring**

What are breach paths?



Created by Peter van Driel from Noun Project

- Security tools find so many weaknesses
  - Often hard for teams to prioritize according to risk
- Most breaches result from attackers exploiting multiple weaknesses
- Breach path is collection of weaknesses that attacker can exploit to reach objective
- Focusing on breach paths enables risk-based:
  - Automation
  - Prioritization
  - Remediation

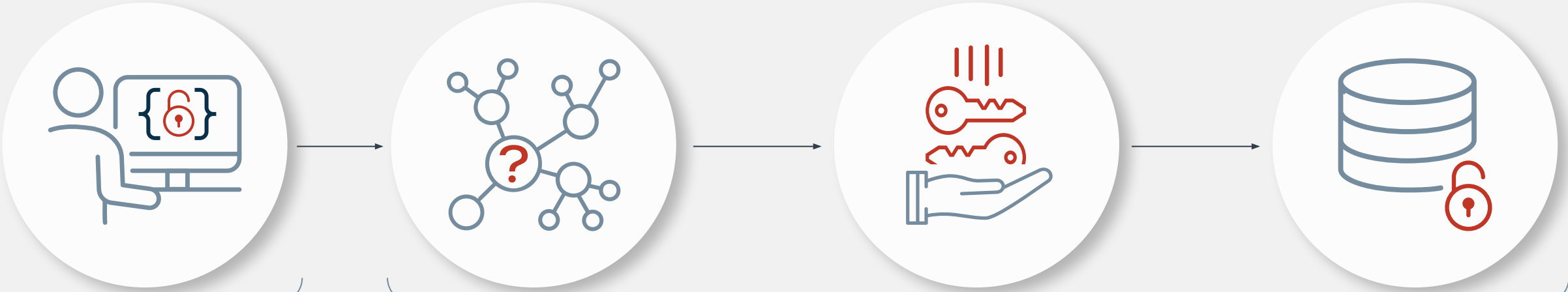
Programmatically Detecting Breach Paths During Development

Vulnerable App Exploited (CWE-918)

Misconfigured Compute Resource

Overly Permissive IAM / RBAC Policies

Unencrypted Database

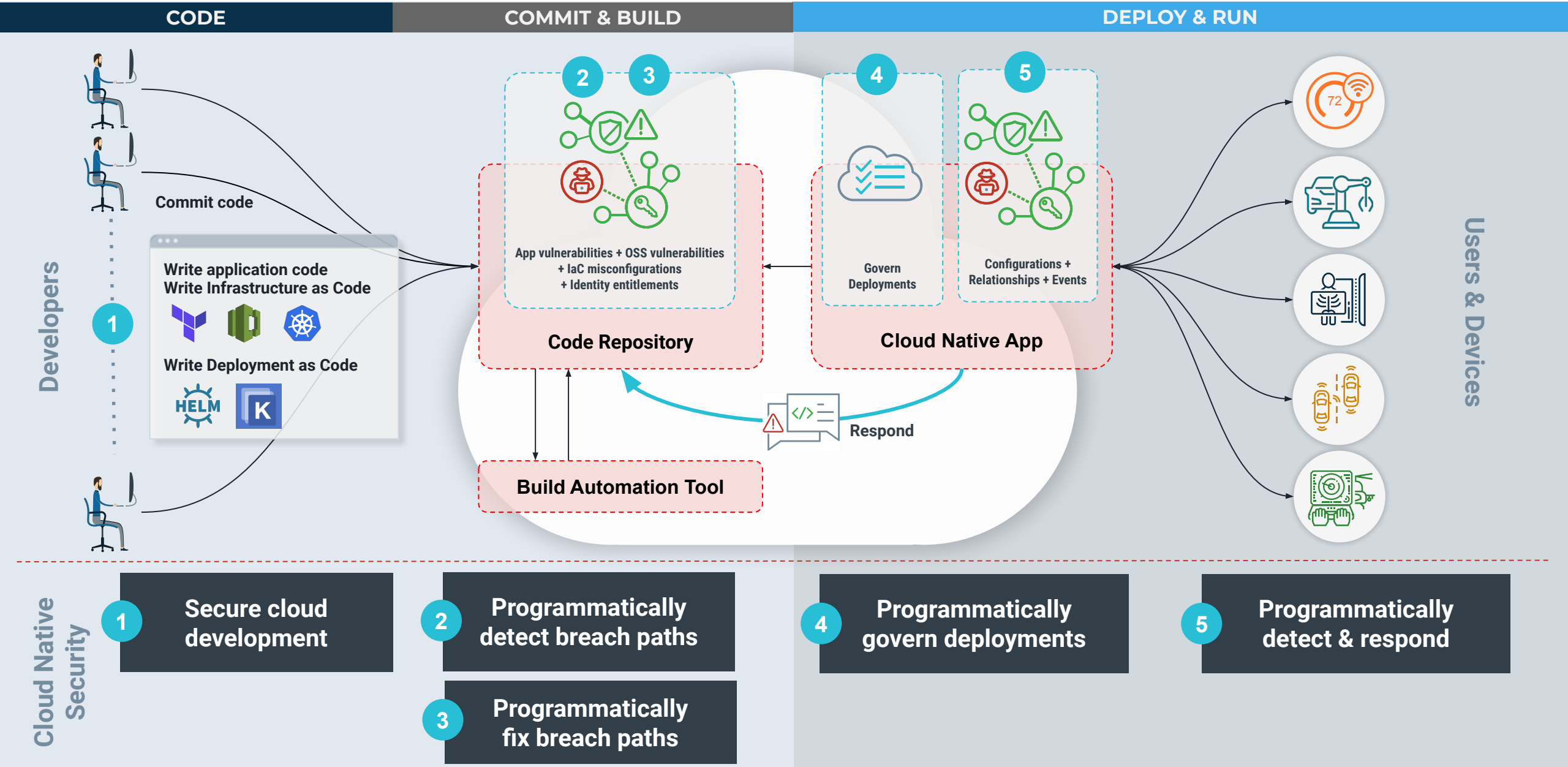


SAST + Infrastructure as Code (IaC) Scanning

PROBLEM : SAST tools do not have context if a vulnerability is exposed, exploitable, and if lateral movement is possible.

SOLUTION: Combining data from SAST & IaC scanning tools will enable end-to-end breach path identification.

A developer-first approach to security enables DevSecOps



Thank You

Let's Chat!

SKILup Day: DevSecOps – "Meet the Speakers" Chat Lounge

Email – jon@accurics.com