RSAC Innovation Sandbox 2021 WINNER

Gartner COOL VENDOR 2021

apiiro

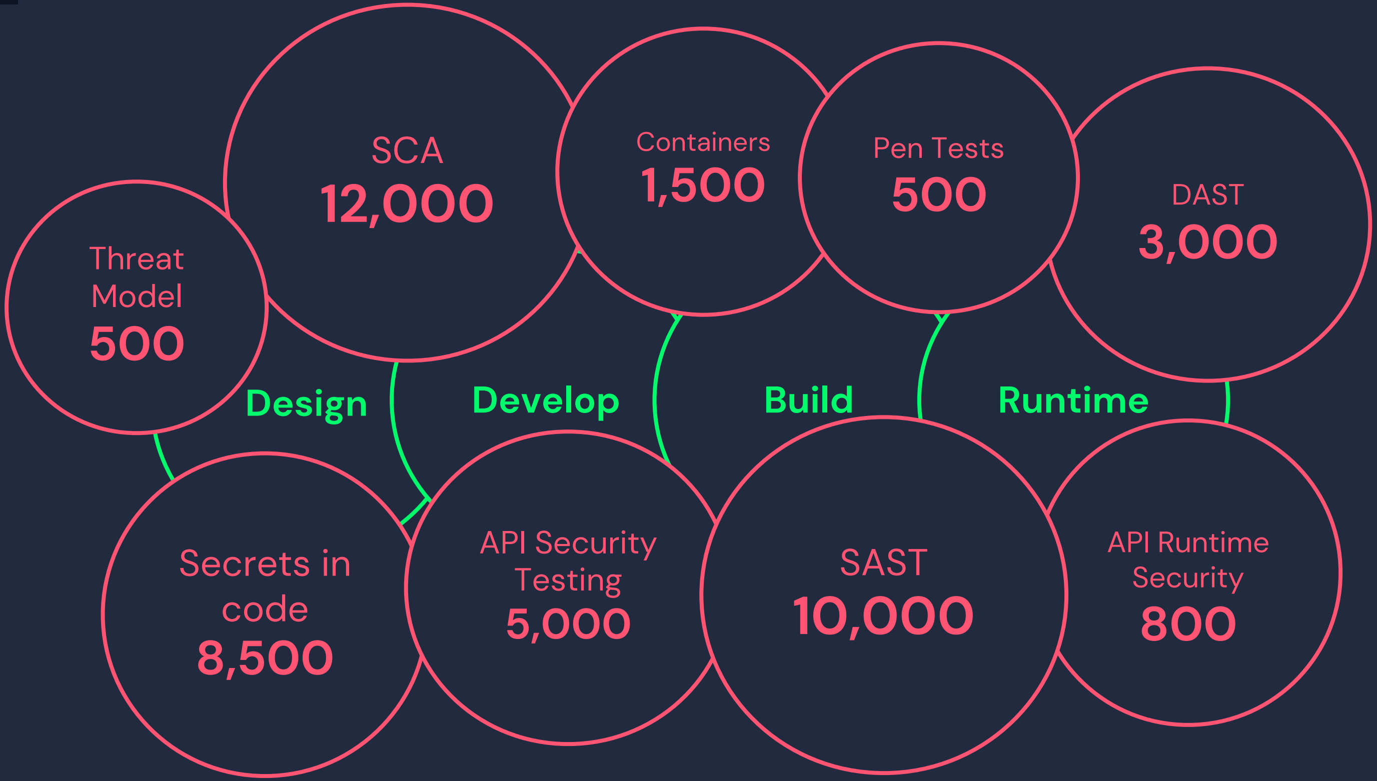
# Decoding Application Security Posture Management (ASPM)

Idan Plotnik, Co-Founder and CEO, Apiiro

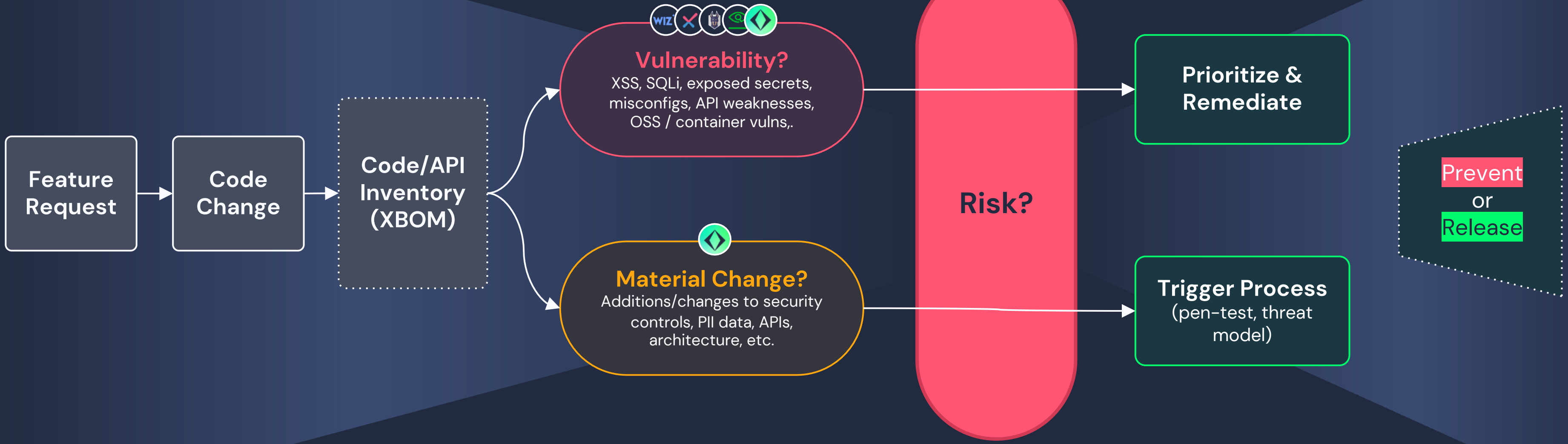
# Agenda

- Current AppSec challenges
- Defining risk
- Taking a risk-based approach with ASPM
- How to build a risk-based AppSec program
- Intro to Apiiro
- ASPM evaluation criteria

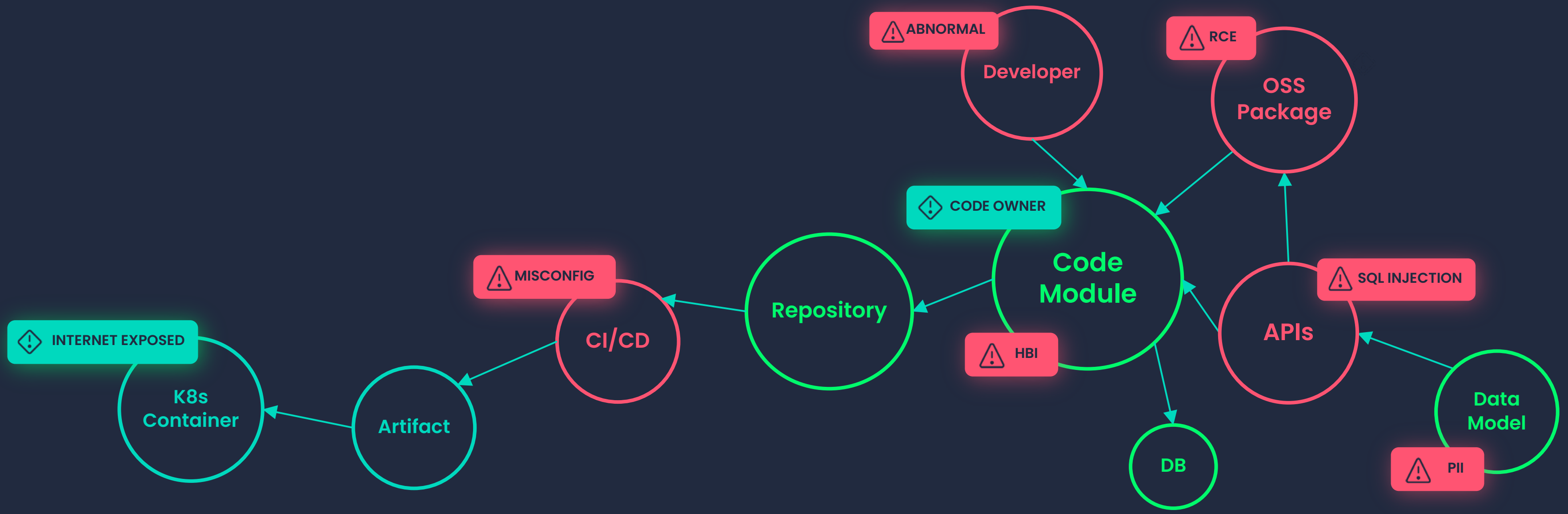
# Vulns = noise


SCA
12,000
Containers
1,500
Pen Tests
500
DAST
3,000
Threat Model
500
Design
Develop
Build
Runtime
Secrets in code
8,500
API Security Testing
5,000
SAST
10,000
API Runtime Security
800

# Managing application risk at scale

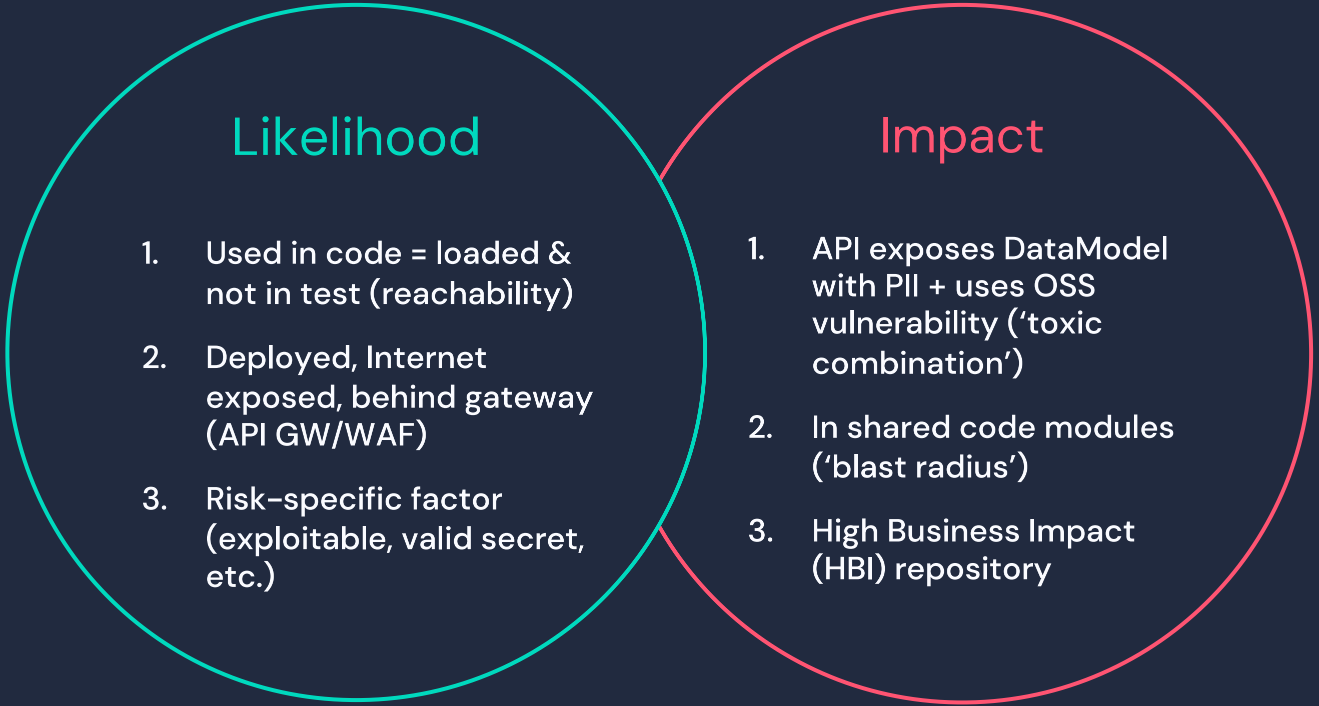

Feature Request
Code Change
Code/API Inventory (XBOM)
WIZ
Vulnerability?
XSS, SQLi, exposed secrets, misconfigs, API weaknesses, OSS / container vulns,.
Material Change?
Additions/changes to security controls, PII data, APIs, architecture, etc.
Risk?
Prioritize & Remediate
Trigger Process
(pen-test, threat model)
Prevent
or
Release

# Risk is a graph



# Risk = likelihood * impact

**Today it is done manually!**



## Likelihood

1. Used in code = loaded & not in test (reachability)
2. Deployed, Internet exposed, behind gateway (API GW/WAF)
3. Risk-specific factor (exploitable, valid secret, etc.)

## Impact

1. API exposes DataModel with PII + uses OSS vulnerability (‘toxic combination’)
2. In shared code modules (‘blast radius’)
3. High Business Impact (HBI) repository

> “By 2026, over **40%** of organizations developing proprietary applications **will adopt ASPM** to more rapidly identify and resolve application security issues.”

—Innovation Insight for Application Security Posture Management

Gartner®

# AppSec Goals

## 1 Build a code/API inventory & automate risk assessment

- ✓ Map app attack surface
- ✓ Scope AppSec processes & tools
- ✓ Meet compliance requirements

## 2 Prioritize & remediate application risks

- ✓ Reduce MTTR
- ✓ Reduce manual triage time
- ✓ Reduce developer workload

## 3 Manage, measure & prevent application risk

- ✓ Security controls assurance
- ✓ Prevent with developer guardrails
- ✓ Measure MTTR vs. SLA

apiiro

# Customer Case Study #1

SoFi

## The challenge

Supporting business and development velocity while reducing application risk.

## The solution

- ✓ Complete inventory of applications
- ✓ Continuous material code change detection
- ✓ Remediation and process automation

> **“There’s a lot of ASPMs out there. I don’t think we have run across one that’s doing code analysis..and providing the insights that Apiiro does.”**

**Zach Schulze**
Sr. Staff Application Security Engineer

## The results

Went from spending **hours** analyzing design reviews to **5-15 minutes**

Reduced mean time to remediation (MTTR) from **8 days** to **10 minutes**

Got **instant** visibility across applications, APIs, and subsidiaries

apiiro

# Customer Case Study #2

paddle

## The challenge

Delayed releases due to manual, time-consuming security reviews

## The solution

- ✓ Aggregation & enrichment of security alerts
- ✓ Risk-based developer security guardrails
- ✓ Unified solution across dev & sec

> "With Apiiro is a force multiplier we can do more with less, meet the developers where they're comfortable, and provide them the info that they need to fix issues in a single unified view."

**Jonny Herd**
VP of InfoSec & Enterprise Technology, Paddle

## The results

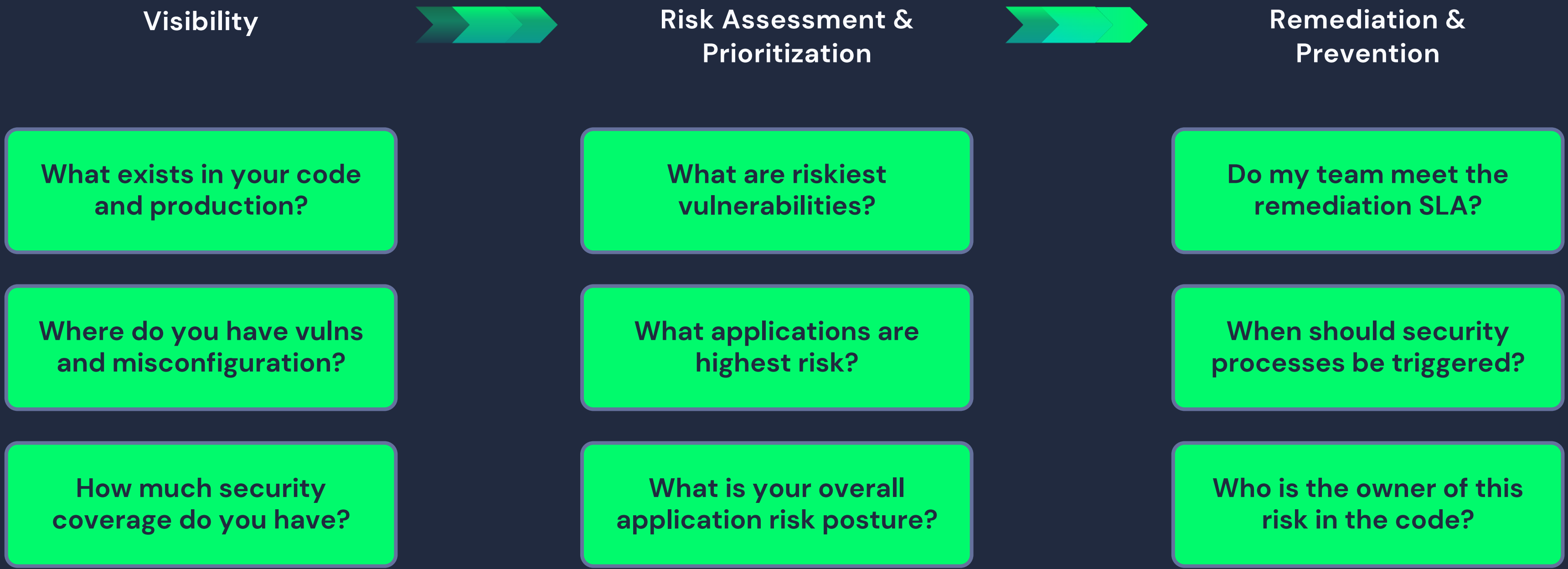
Monitoring **100+** pull requests per week to block high-risk changes

Gives their AppSec team **2 days'** worth of work back per week

Expand security testing coverage with Apiiro's **native SSCS solution**

apiiro

# ASPM: Building a risk-based AppSec program



Visibility

- What exists in your code and production?
- Where do you have vulns and misconfiguration?
- How much security coverage do you have?

Risk Assessment & Prioritization

- What are riskiest vulnerabilities?
- What applications are highest risk?
- What is your overall application risk posture?

Remediation & Prevention

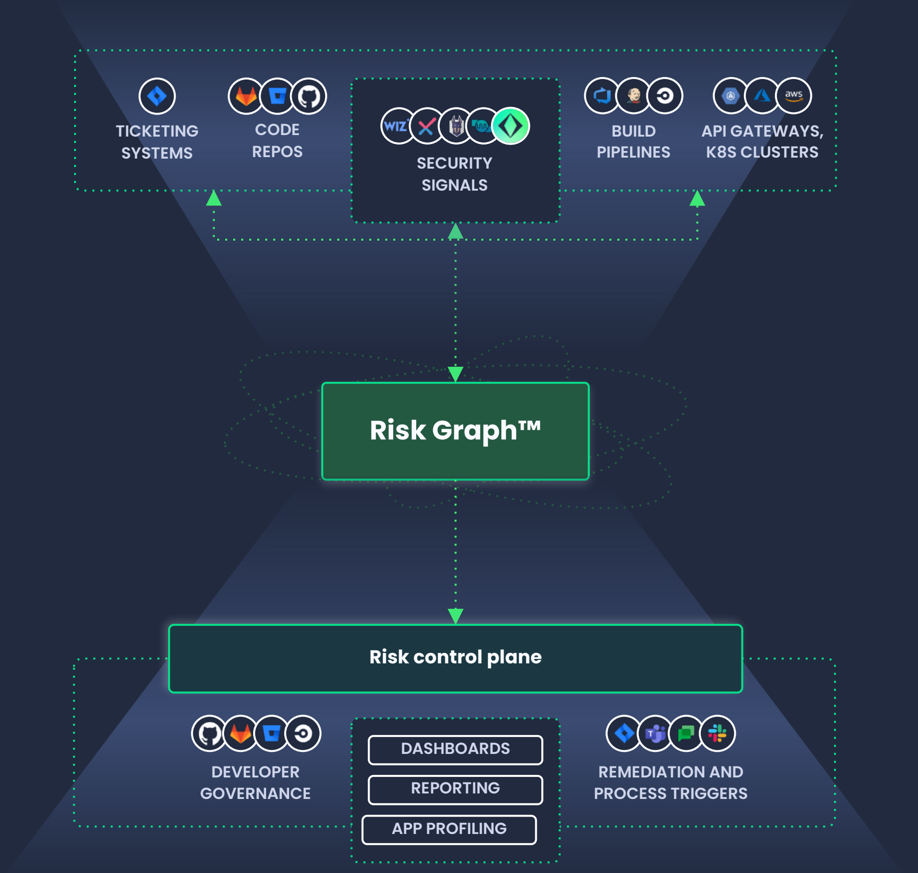
- Do my team meet the remediation SLA?
- When should security processes be triggered?
- Who is the owner of this risk in the code?

# Apiiro: Deep ASPM

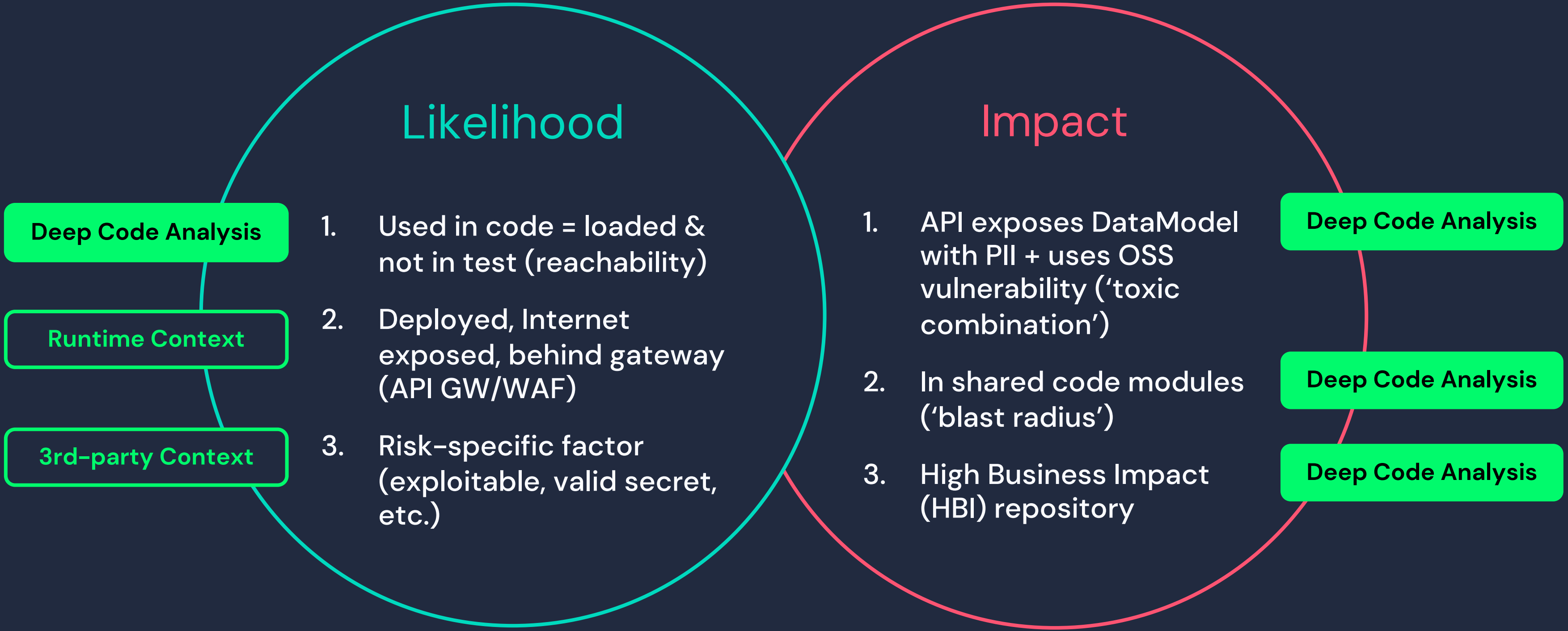
**Get complete & continuous app visibility**

Prioritize & remediate application risks

Manage, prevent & measure application risk



# Apiiro = ASPM + Deep Code Analysis + Runtime Context



## Likelihood

1. Used in code = loaded & not in test (reachability) — Deep Code Analysis
2. Deployed, Internet exposed, behind gateway (API GW/WAF) — Runtime Context
3. Risk-specific factor (exploitable, valid secret, etc.) — 3rd-party Context

## Impact

1. API exposes DataModel with PII + uses OSS vulnerability (‘toxic combination’) — Deep Code Analysis
2. In shared code modules (‘blast radius’) — Deep Code Analysis
3. High Business Impact (HBI) repository — Deep Code Analysis

# ASPM Evaluation Criteria

- Deployment & security coverage
- Security tool integration
- Code Inventory
- Correlation & prioritization
- Remediations & prevention
- Application risk profiling
- Governance & compliance
- Native AST solutions
- Native supply chain security solutions

RSAC Innovation Sandbox 2021 WINNER

Gartner COOL VENDOR 2021

apiiro

# Thanks!