



SCALING CODE ANALYSIS ACROSS AN ENTERPRISE

FLORIAN NOEDING

PRINCIPAL SECURITY ARCHITECT @ ADOBE

ABOUT ME

- Principal Security Architect @ Adobe
Software Engineer → Security Researcher
→ Security Strategy
- Fun fact: I bake my own German style bread
recipe on my blog <https://florian.noeding.com>



SECRETS IN SOURCE CODE (SISC)

- Detect credentials in source code or repositories

```
import requests

token = '44AE90194399'

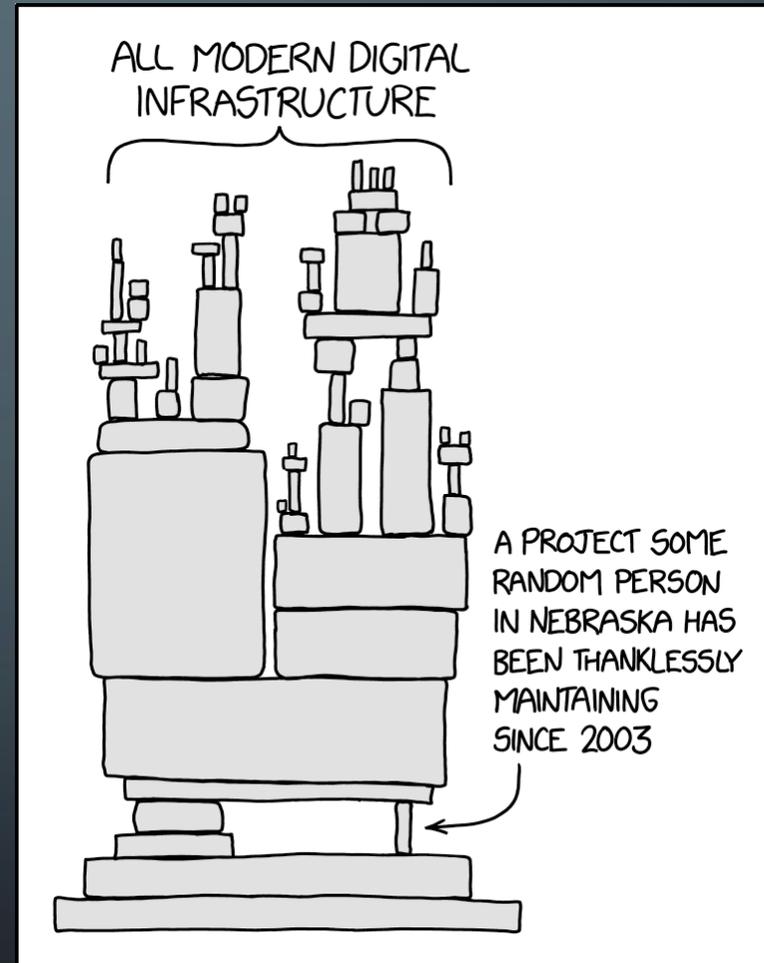
url = 'https://example.com/api'

headers = {
    'Authorization': f'Bearer {bearer_token}'
}

response = requests.get(url, headers=headers)
```

SOFTWARE COMPOSITION ANALYSIS (SCA)

- Create inventory of 3rd party dependencies
- Enables look-up of CVEs affecting these libraries



<https://xkcd.com/2347/>

STATIC APPLICATION SECURITY TESTING (SAST)

- Looks for vulnerable code patterns or dataflows
- Identifies 1st party vulnerabilities

```
@app.get("/user/")
def read_user(username: str):
    query = f"SELECT * FROM users WHERE username = '{username}'"
    conn = get_db_connection()
    user = conn.execute(query).fetchone()
    conn.close()

# ...
```

source

mixing code and data
without output encoding

sink



CHALLENGE

- Hundreds of Products across desktop, mobile, web, ...
- Diverse tech stacks
 - 12 programming languages make up 80% of our code
 - Many more frameworks
- Multiple SCMs
- 100k+ repositories as primary scope
- On average 30k scan events per day

The image features a dark blue background with white, stylized circuit board traces in the corners. These traces form various geometric shapes and paths, some ending in small circles, resembling a network or data flow diagram. The central text is positioned in the middle of the frame.

DESIGN, ROLLOUT, IMPACT

DESIGN PRINCIPLES

Great Developer Experience as the key goal to ensure acceptance:

- Integrate deeply into developer workflow
- Provide timely, relevant and actionable feedback (shift-left!)
- Carefully balance noise and risk reduction
- Single pane of glass into source code related findings

Goal: pragmatic risk reduction instead of zero known vulnerabilities.

PRIMARY PROCESS



Security
as
Code



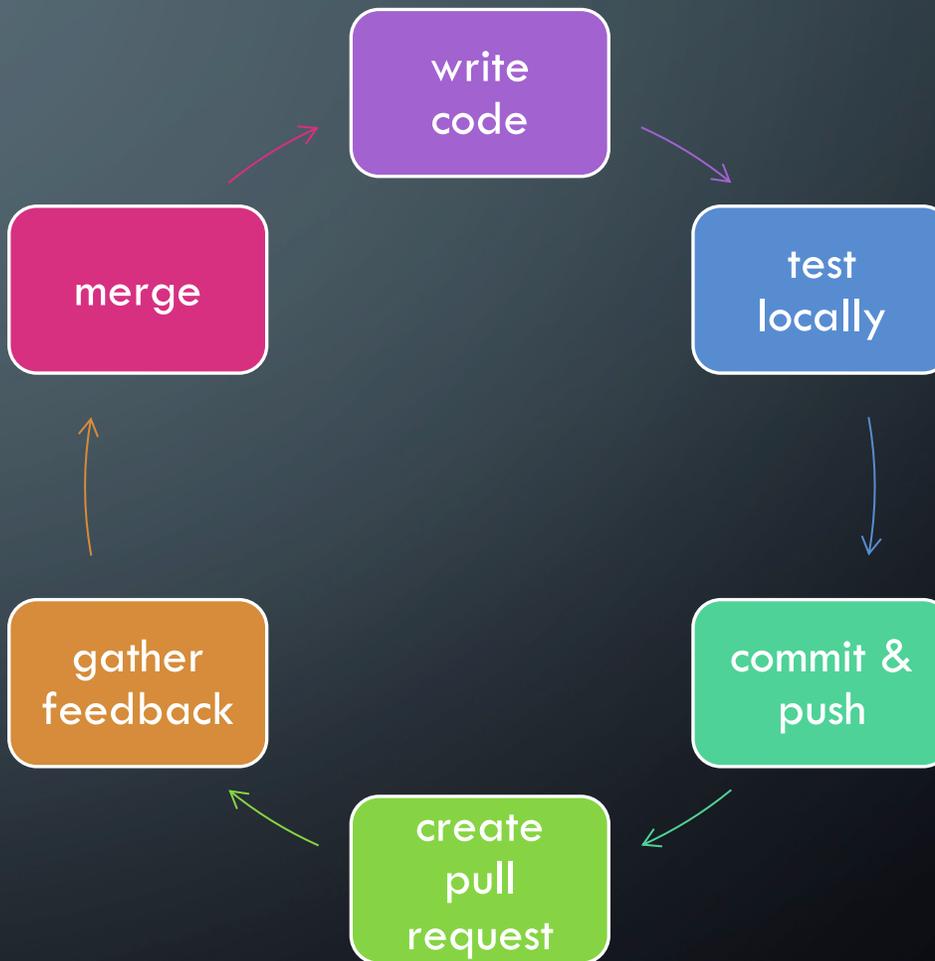
Adversary
Intelligence



Scan requested



Feedback
≤ 5 min + build



FEEDBACK LOOPS

Inline feedback on
pull requests

Findings in changed files only (important!), very few exceptions

Metrics driven
security campaigns

only a few specific risks

Custom dashboard

All findings on any branch
Challenge: not yet widely adopted by engineers

Ticketing for
enforcement

Critical risks only
Challenge: Attribution to project often non-trivial

SCAN TOOL SELECTION

Easy to roll-out

Finds important vulnerabilities

Developer friendly output

Fast enough

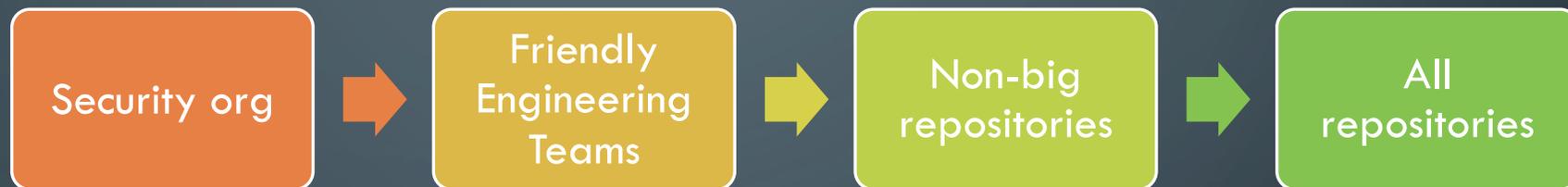
Customizable

SCOPE



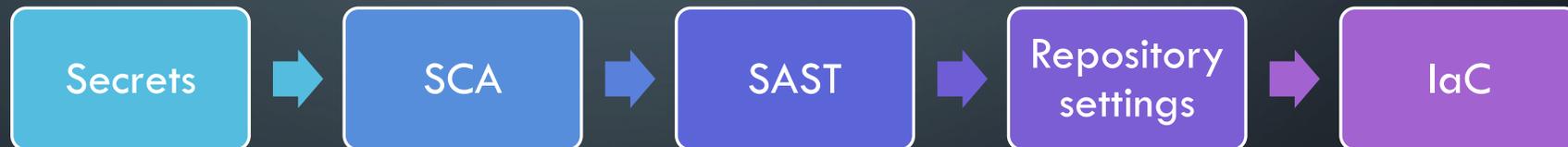
ROLLOUT

Scope



Ongoing feedback
for Kodiak team

Scanners





OUTCOMES 2023

300,000 findings fixed

nudging only – zero enforcement



The image features a dark blue background with white, stylized circuit board traces in the corners. These traces consist of straight lines of varying lengths and angles, ending in small circles, resembling a network or data flow diagram. The traces are located in the top-left, top-right, bottom-left, and bottom-right corners, framing the central text.

RISK PRIORITIZATION

ADVERSARY MODEL

HIGHLY SIMPLIFIED

Security Researchers

- Various motivations
- generally friendly
- no 0-day leaks

→ Use to identify gaps in program

eCriminals

- Often financially motivated
- Repeatable, scalable exploitation

→ Fix easy to exploit or widely deployed vulnerabilities first

Nation States

- Operations with targeted outcomes
- Hard to predict

→ Defense in depth

SISC – RISK REDUCTION STRATEGY

MOST BREACHES INVOLVE LEAKED OR STOLEN CREDENTIALS

Public and widely shared internal repos

Easy to abuse, particularly cloud credentials

Active and long-lived

Everything else

- consider accepting revoked credentials in historical commits

SCA – RISK REDUCTION STRATEGY

MANY BREACHES START WITH AN OUT-OF-DATE SYSTEM

Exploited in the Wild

- CISA's Known Exploited Vulnerability (KEV) catalog

Exploit Available

- Various Intelligence Feeds

Likely to be exploited

- First's Exploit Prediction Scoring System (EPSS)

Everything else

- Severity (CVSS), Customer & Compliance Expectations

Future:

- Filter out unreachable CVEs
- Use contextual data

SBOM Transparency:

→ Fix based on CVSS

SAST – RISK REDUCTION STRATEGY

OPEN PROBLEM – RELYING ON SAST VENDOR'S SCORES

Exploited in the Wild

```
graph TD; A[Exploited in the Wild] --> B[Exploit Available]; B --> C[Likely to be exploited]; C --> D[Everything else];
```

Exploit Available

Likely to be exploited

Everything else

Future:

- CWE \Leftrightarrow TTP mapping?

Let me know if you've solved this!

UNIFIED RISK PRIORITIZATION

SIMPLIFIED MODEL

SCA – Exploited in the wild (KEV)

```
graph TD; A[SCA – Exploited in the wild (KEV)] --> B[SISC – Critical secrets]; B --> C[SCA – Likely to be exploited (EPSS)]; C --> D[Everything else];
```

SISC – Critical secrets

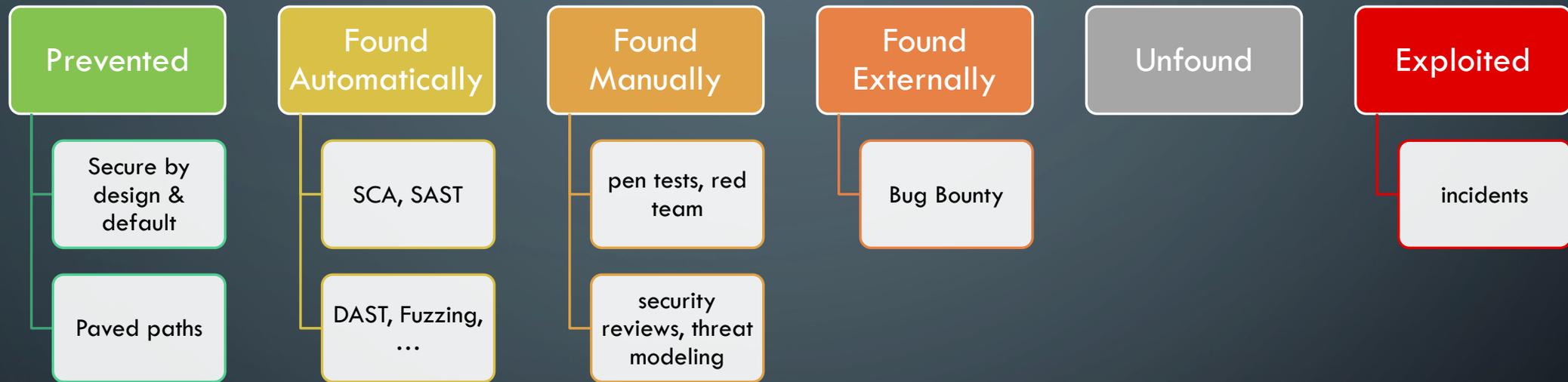
SCA – Likely to be exploited (EPSS)

Everything else

The image features a dark blue background with white, stylized circuit board traces in the corners. These traces consist of straight lines of varying lengths and angles, ending in small circles, resembling a network or data flow diagram. The traces are located in the top-left, top-right, bottom-left, and bottom-right corners, framing the central text.

META FEEDBACK LOOP

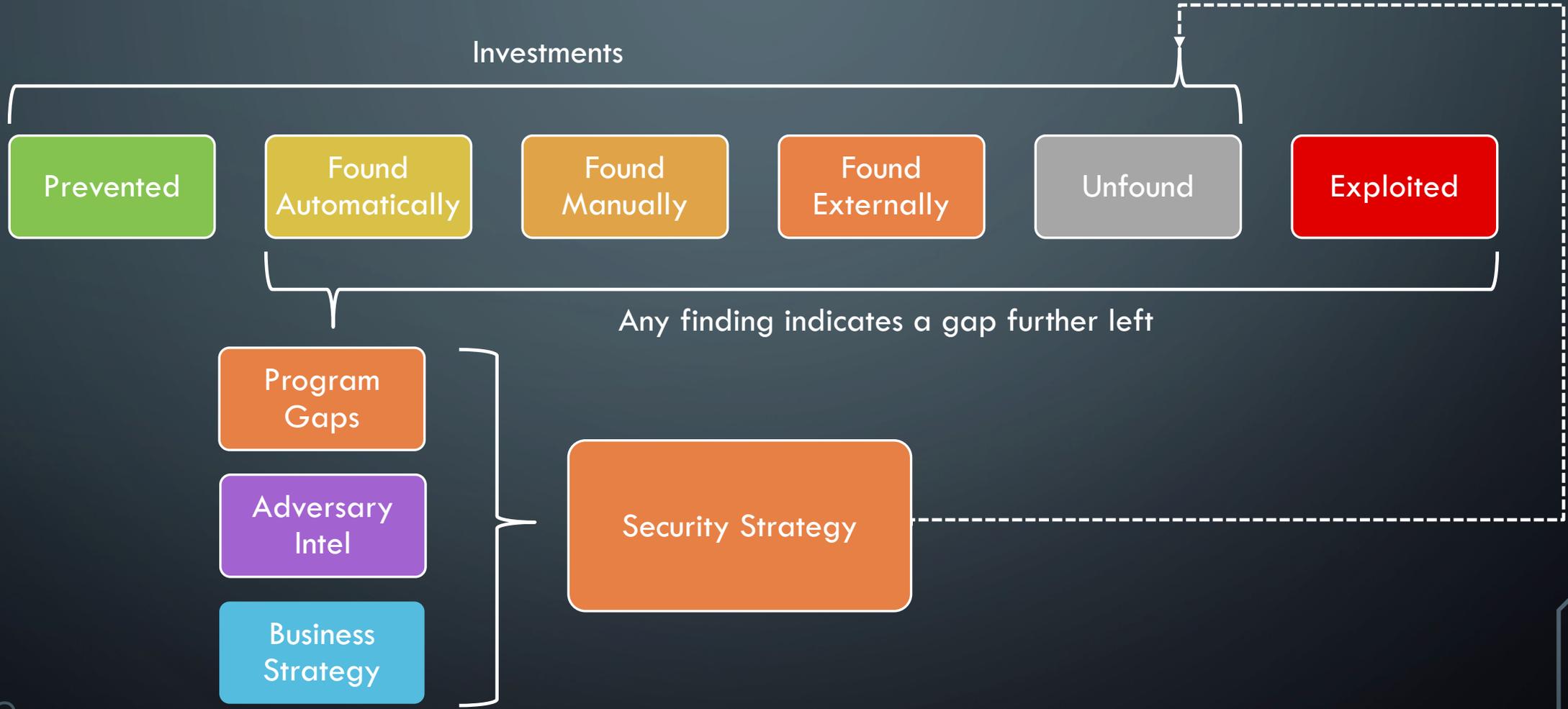
SHIFTING LEFT – 6 BUCKETS OF RISK



Fixing root causes > Fixing symptoms

Hazard elimination > Hazard remediation

META FEEDBACK LOOP



QUESTIONS?

Key Takeaways:

- Focus on great DevEx – talk to them!
- Feedback loop design is crucial
- Fix things that matter
 - SCA: Consider using EPSS
 - SAST: target root causes, not symptoms

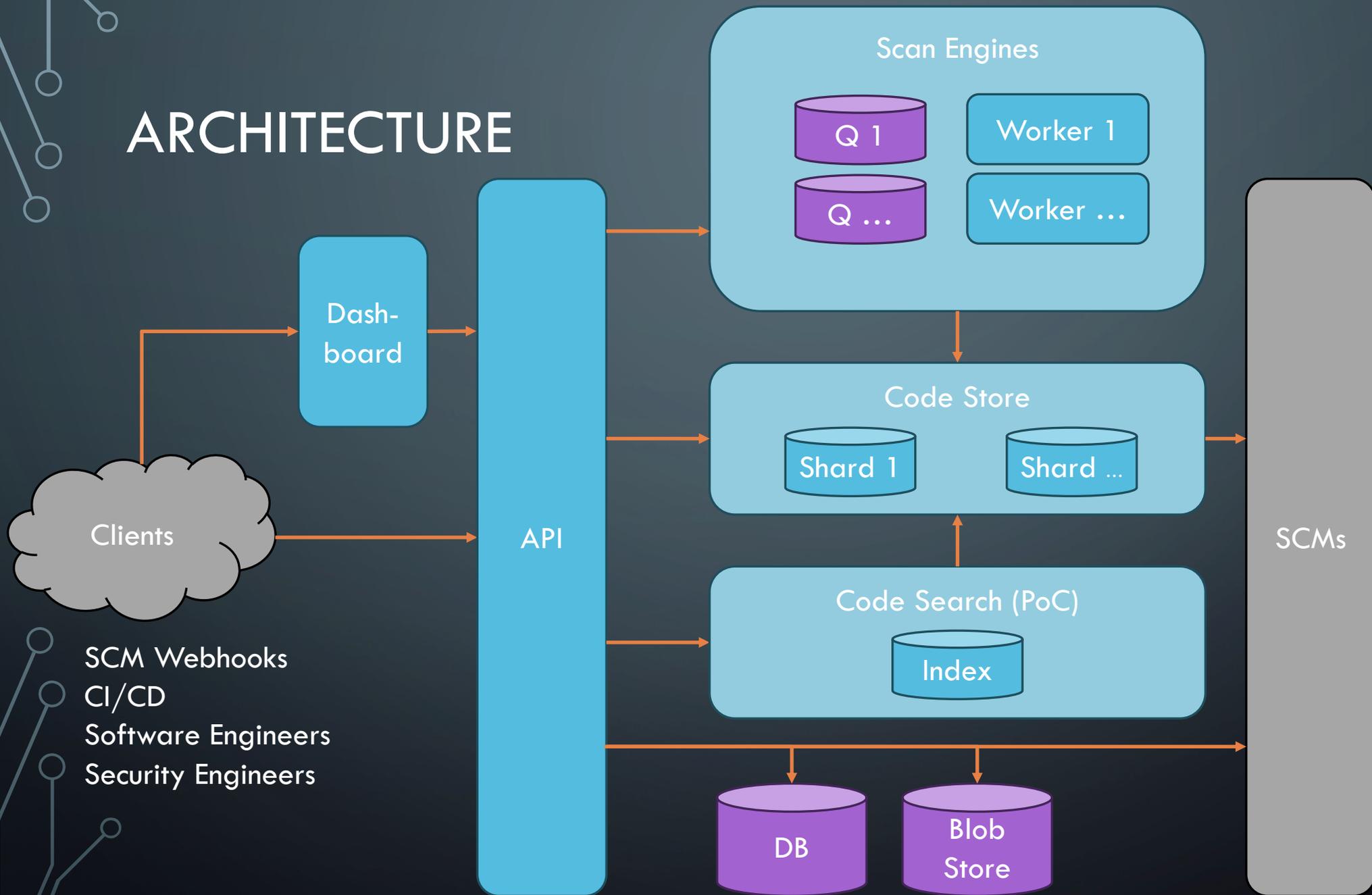


<https://florian.noeding.com>

The background is a dark blue-grey color. In the four corners, there are decorative white line-art patterns that resemble circuit traces or a stylized tree structure. These patterns consist of thin lines that branch out and terminate in small circles, creating a sense of connectivity and technology.

BACKUP SLIDES

ARCHITECTURE

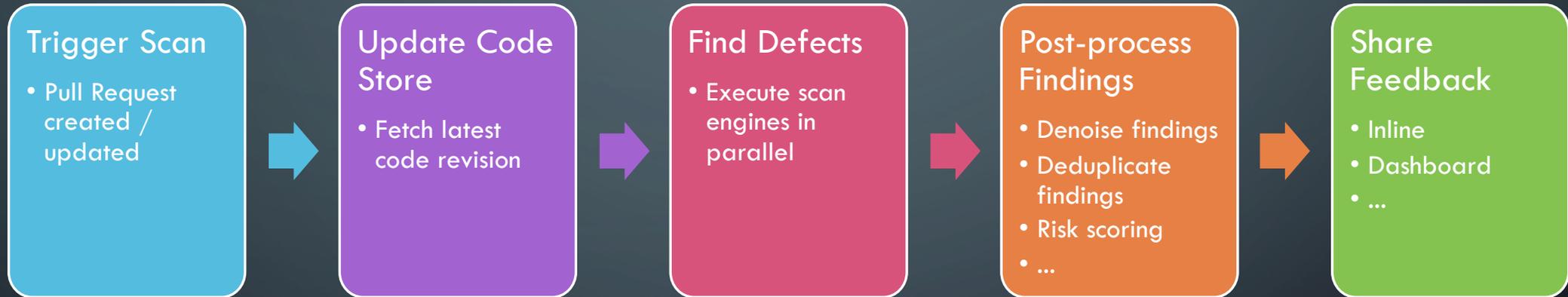


SCM Webhooks
CI/CD
Software Engineers
Security Engineers

Legend

- Kodiak
- Cloud
- External

KODIAK INTERNALS



LINKS

- EPSS: <https://www.first.org/epss/>
- KEV: <https://www.cisa.gov/known-exploited-vulnerabilities-catalog>
- 6 buckets of risk: http://collingreene.com/6_buckets_of_prodsec.html
- My blog: <https://florian.noeding.com>
- More about Kodiak: <https://blog.developer.adobe.com/project-kodiak-shifting-application-security-left-at-enterprise-scale-55f5453d1966>