42crunch

Shifting security left, sure! but how?

Continuous Security for APIs

ISABELLEMAUNY
ISABELLE@42CRUNCH.COM

API SECURITY CHALLENGES

“Sure, let’s test that later…”

For the security we said we'd hire someone to...
Hold on, that's not the priority. Let's get the project started first, we'll see about that later...
Hey the project is nearly done, we should do a security audit to...
Too late man, we don't have the time or the budget for that
The site's been online for 10 days, and we haven't done any security testing, I'm nervous...
Don't worry, we'll take care of it later...
LATER...
DOWN! THE SITE IS DOWN! WE'VE BEEN HACKED!
How could you let this happen!?
CommitStrip.com

“We are deploying that API, AGAIN???”

APPLICATION DEVELOPMENT

APPLICATION SECURITY

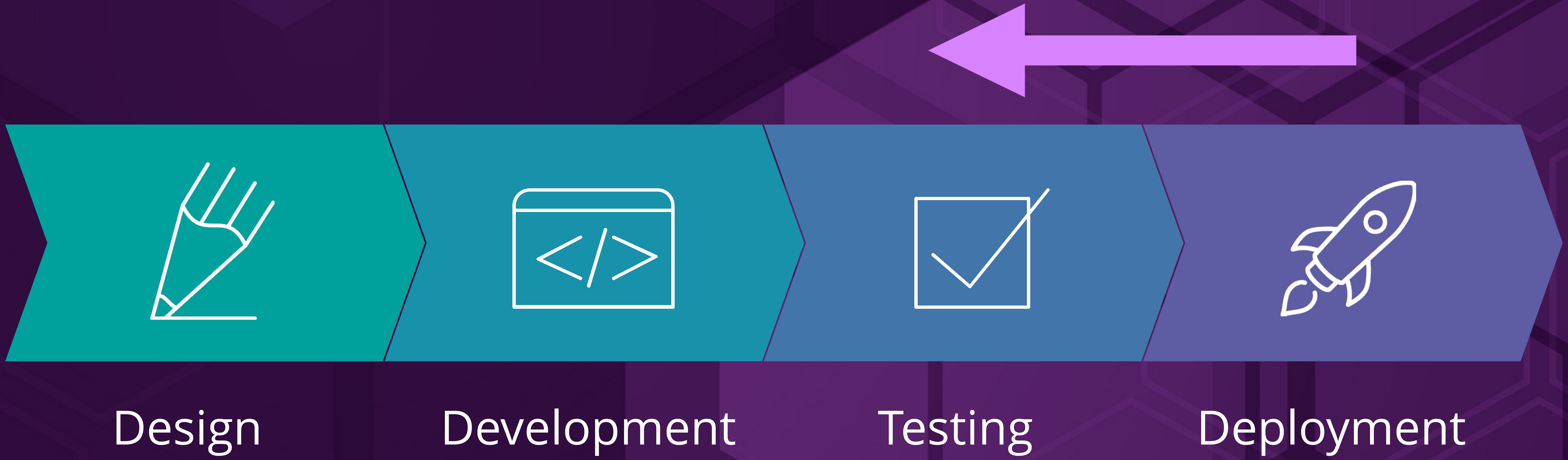
“We have to protect our APIs using 3-legged Oauth but only with authorization_code and PKCE …”

PROTECTING APIS REQUIRES A NEW APPROACH

MEET
DEV SEC OPS

“DevSecOps is the philosophy of integrating security practices within the DevOps process.

DevSecOps involves creating a 'Security as Code' culture with ongoing, flexible collaboration between development, release engineers and security teams.”



Design

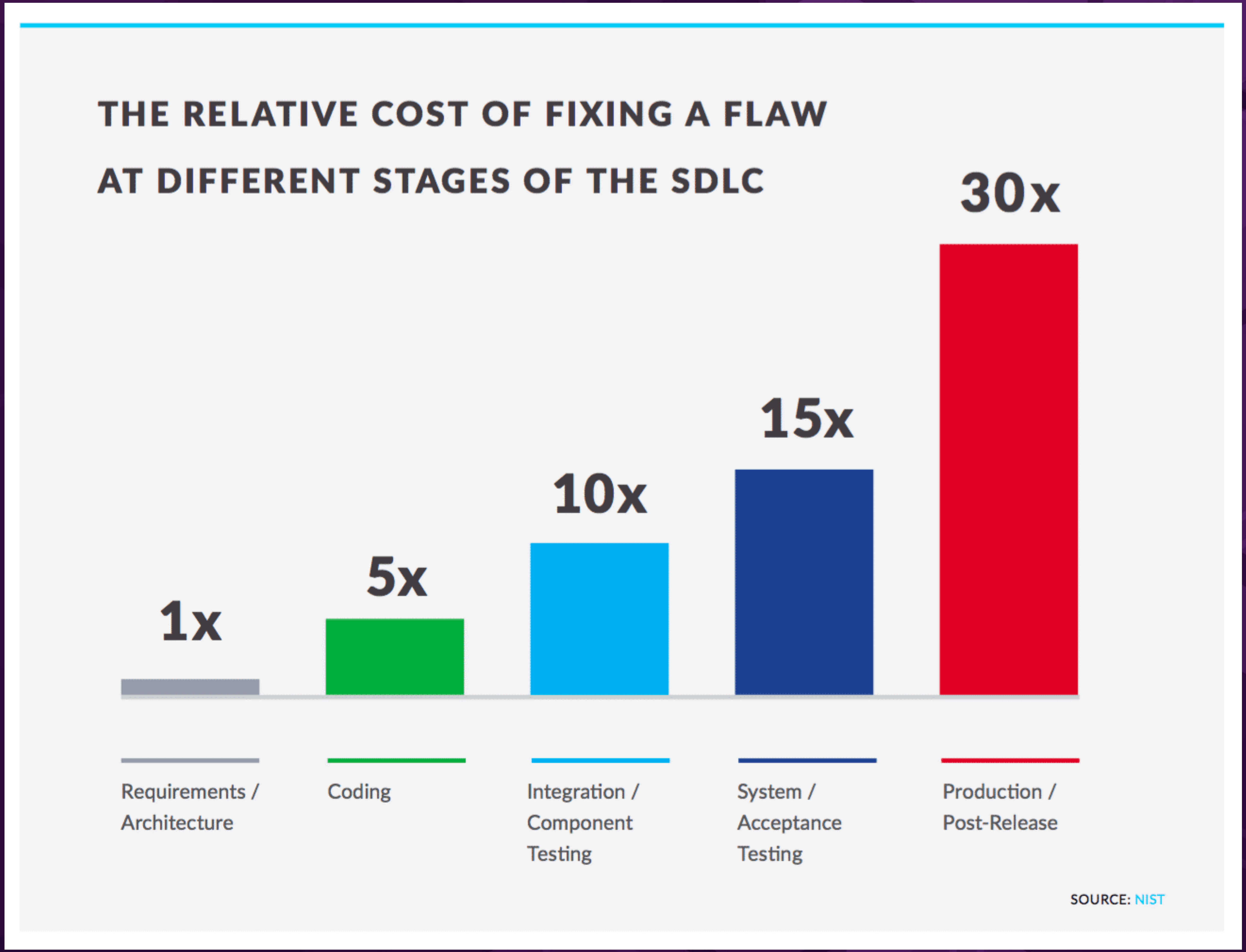
Development

Testing

Deployment

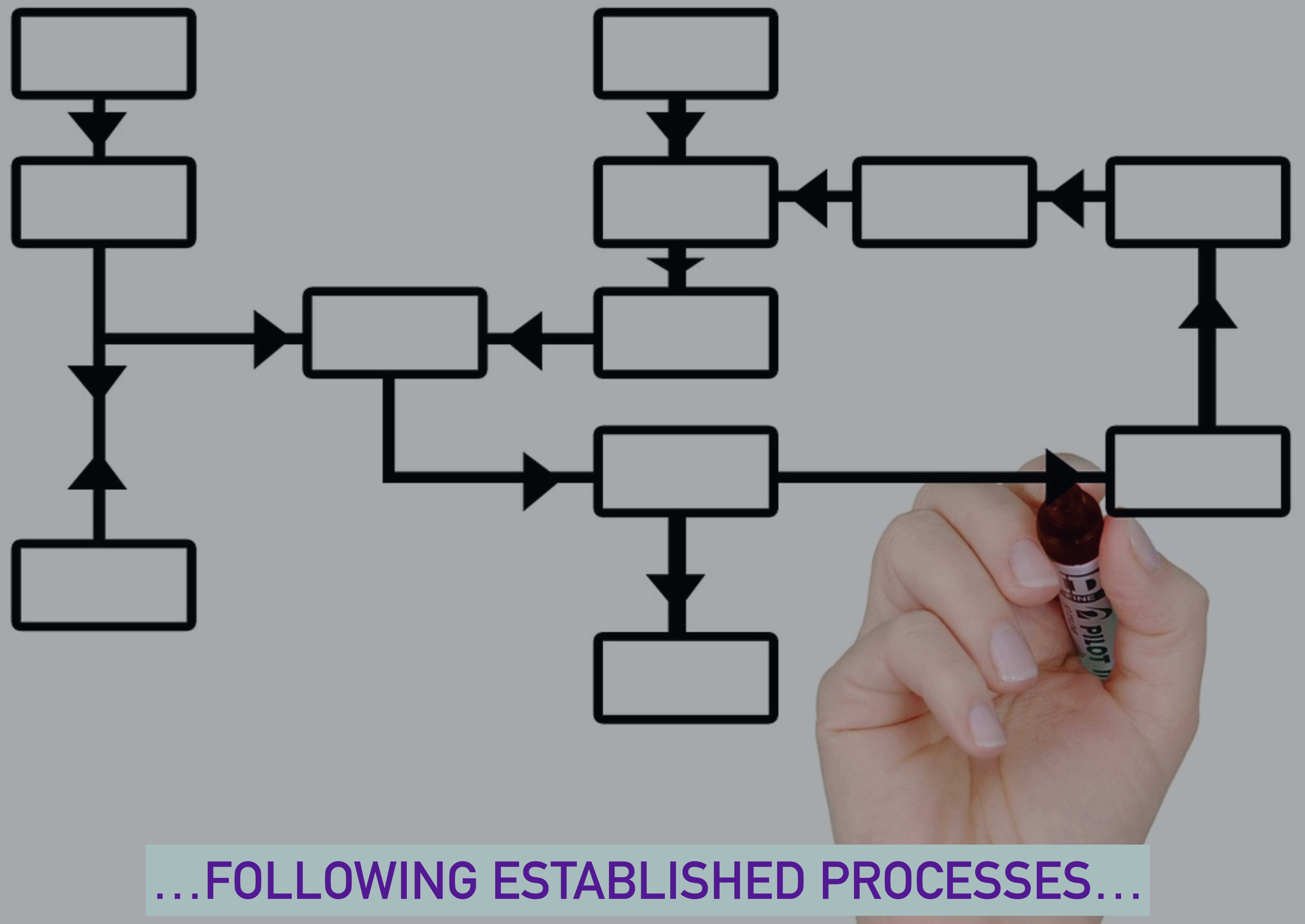
INJECTING SECURITY AS EARLY AS POSSIBLE IN THE API LIFECYCLE

COST OF DEFECTS ALONG THE LIFECYCLE



SOURCE: NIST

A CHANGE IN CULTURE: PEOPLE COLLABORATING…



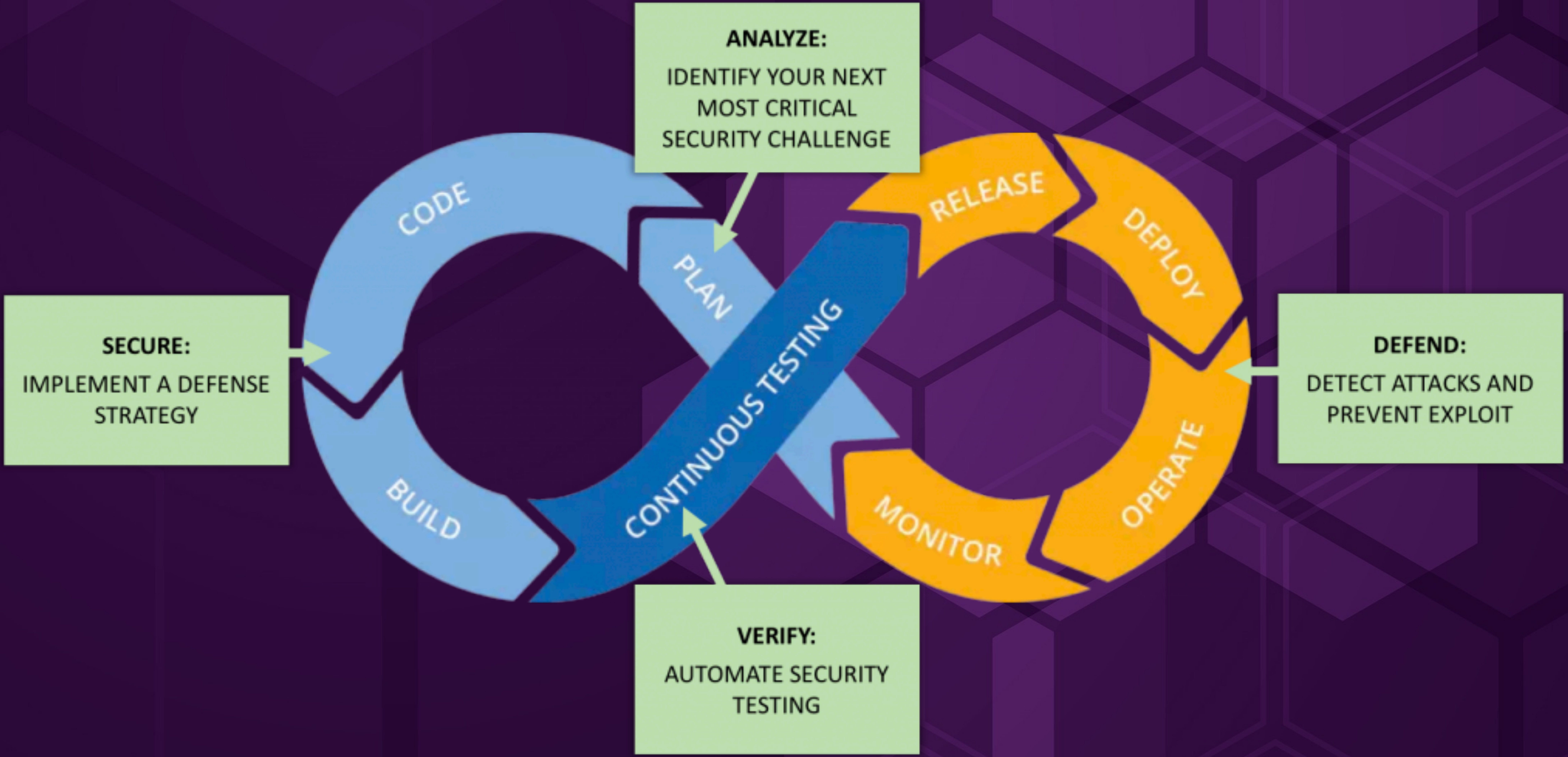
...FOLLOWING ESTABLISHED PROCESSES...

...AND USING THE RIGHT TOOLS.

KEY BENEFITS

- Everyone is responsible for security, everyone has a role to play
  - ✓ No more “throwing over the fence” approach
- Secure by design principles
  - ✓ Automated reviews
  - ✓ Automated security testing
- Security becomes transparent, thanks to security as code
- Developers iteratively learn about best practices
- Security is continuously improved

A DEV-SEC-OPS CYCLE FOR APIS



ANALYZE:

IDENTIFY YOUR NEXT MOST CRITICAL SECURITY CHALLENGE

SECURE:

IMPLEMENT A DEFENSE STRATEGY

DEFEND:

DETECT ATTACKS AND PREVENT EXPLOIT

VERIFY:

AUTOMATE SECURITY TESTING

From: https://jaxenter.com/exploration-devsecops-144849.html

1

ANALYZE

What do we need to secure ?

KNOW YOUR APIS AND THE RISK THEY BRING

See: https://www.owasp.org/index.php/Application_Threat_Modeling

2

SECURE

Establish the rules

CORE API SECURITY RULES

- All APIs request/response data must be validated
- All access tokens must be validated
- Proper authentication in place, adapted to risk
- Rate Limiting for all operations
- Fine-grained authorization for data access

- Authenticate Apps
- Managed secrets: no hardcoded/ readable APIKeys, passwords, tokens in code or deployment scripts
- Security headers must be used
- No libraries with known vulnerabilities
- All transactions are logged
- All APIs are known and governed

Check “How to Prevent” section from OWASP Top 10 for APIs

3

VERIFY

Ensure we comply with the rules!

Code Analysis (SAST)
Code reviews (manual)
API contract analysis (SAST)
Software Component Analysis

VERACODE WhiteSource
CHECKMARX sonarqube
42crunch snyk

Dev

API Implementation Testing (DAST)
API Contract Testing (DAST)
Negative Testing: Hack yourselves!
Container Images Analysis
Deployment Scripts Analysis

clair
Katalon
42crunch

QA/Testing

SSL/TLS Configuration
Kubernetes Configuration
Perf testing (DOS protection)
Pen Testing (manual)

kube-hunter Gatling
42crunch
Qualys. SSL Labs

Production/Ops

RULE OF THUMB FOR TOOLS

- Fit in “developer flow”
  - ✓ IDEs Integration
- Can be automated
  - ✓ Plugins for CI/CD pipelines
  - ✓ API driven
- Can integrate with ecosystem
  - ✓ Logging
  - ✓ Monitoring
  - ✓ SIEM

```
[illegible]: Verifying Checksum
[illegible]: Download complete
[illegible]: Verifying Checksum
[illegible]: Download complete
[illegible]: Verifying Checksum
[illegible]: Download complete
[illegible]: Verifying Checksum
[illegible]: Download complete
[illegible]: Verifying Checksum
[illegible]: Download complete
[illegible]: Verifying Checksum
[illegible]: Download complete
[illegible]: Pull complete
[illegible]: Pull complete
[illegible]: Pull complete
[illegible]: Pull complete
[illegible]: Pull complete
[illegible]: Pull complete
[illegible]: Pull complete
[illegible]: Pull complete
Digest: sha256:[illegible]
Status: Downloaded newer image for 42crunch/bitbucket-api-security-audit-pipe:[illegible]
Audited OASFiles/PixiBasic-Fixed-with-protections.json, the API score is [illegible]
    Found 1 issues in with severity "high" or higher
    Details:
    https://platform.42crunch.com/apis/[illegible]

Detected 1 failures in the 1 OpenAPI files checked
```

DEMO

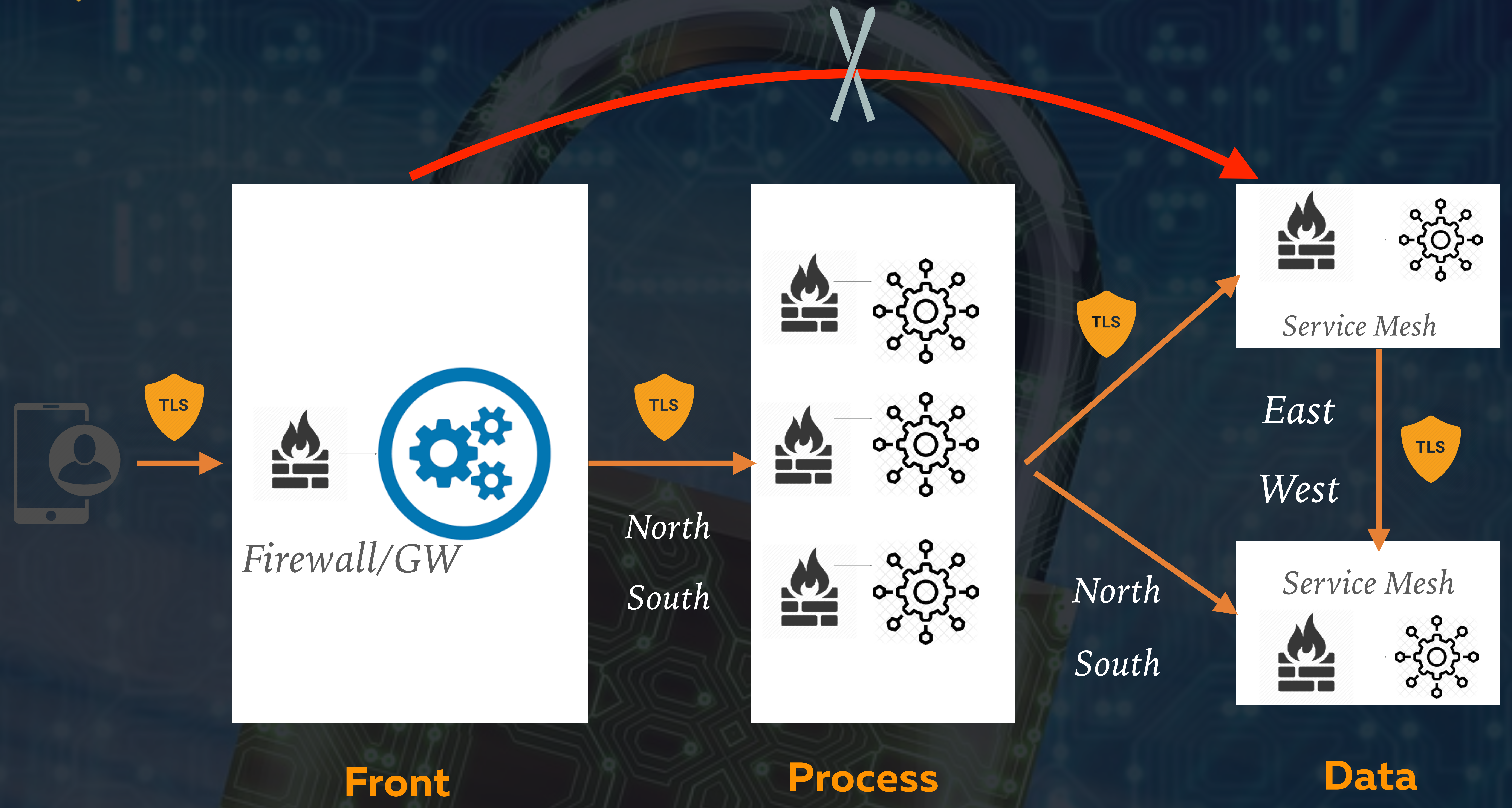
SECURITY AUTOMATION VIA CI/CD

Bitbucket

4

DEFEND

Enforce the rules!



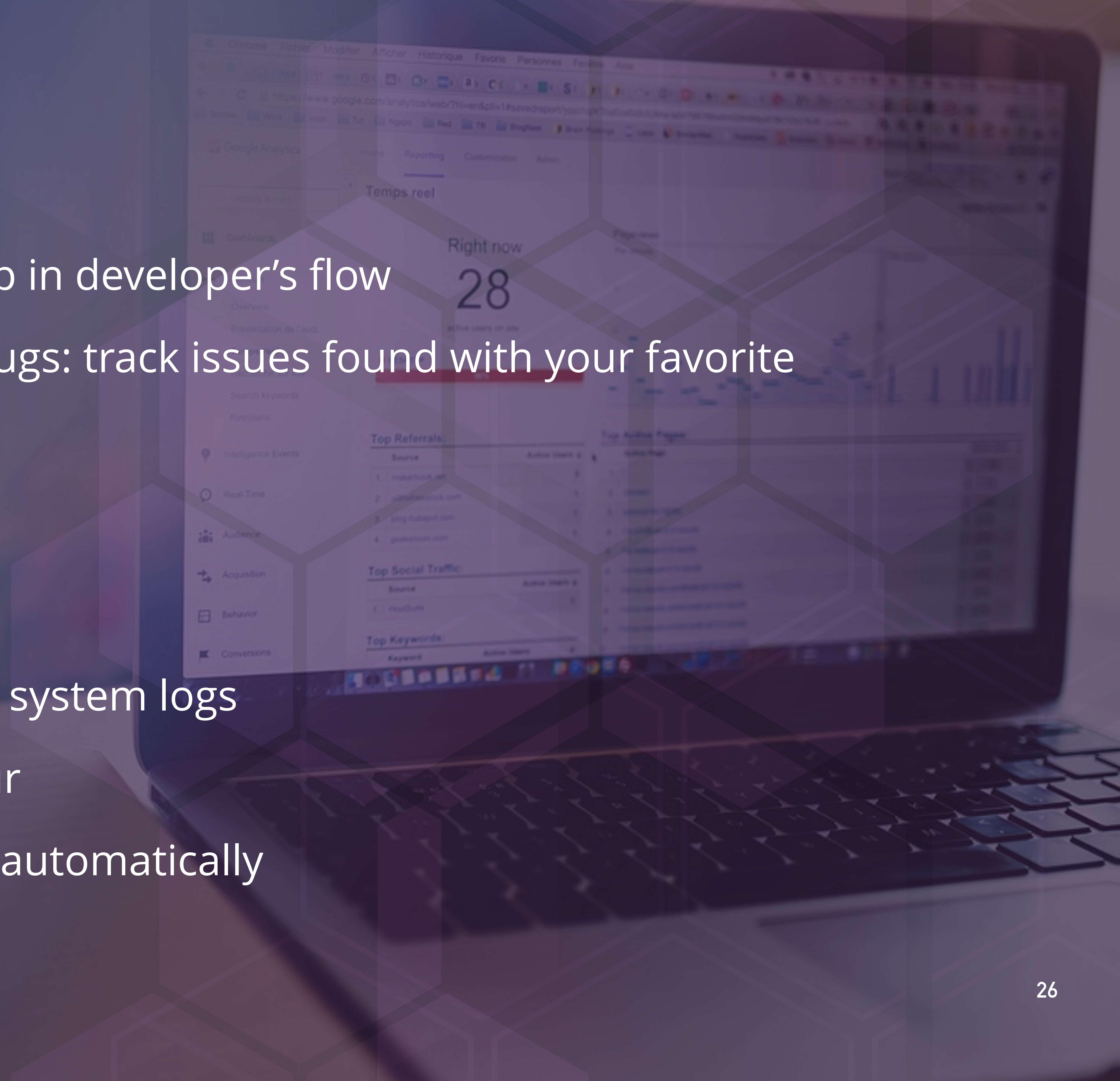
- Automatic Deployment
- Protections as code
- Deployed early

PROTECT *ALL* APIS

App icon made by https://www.flaticon.com/authors/pixel-buddha

MONITOR

Learn and Enhance!



Dev/QA

- ✓ Immediate feedback loop in developer’s flow
- ✓ Treat vulnerabilities as bugs: track issues found with your favorite ticketing system

Production

- ✓ Analyze automatically all system logs
- ✓ Profile runtime behaviour
- ✓ Alert on potential issues automatically

KEY RECOMMENDATIONS

- **Start small and iterate**
  - ✓ Don’t try to address all issues at once!
- **Educate and help developers**
  - ✓ Add security people to development teams
  - ✓ Don’t throw security at them as a new responsibility
  - ✓ Help them by including feedback in their existing development flow
- **Don’t throw too many tools in the pipeline**
  - ✓ Evaluate and choose depending on your needs

RESOURCES

- 42Crunch Website
- Azure DevOps SignUp
- Free OAS Security Audit
- OpenAPI VS Code Extension
- OpenAPI Spec Encyclopedia
- OWASP API Security Top 10
- APIsecurity.io