API Security in a Kubernetes World

Dmitry Sotnikov, 42Crunch CPO, Curator of APIsecurity.io

Agenda

- Why API security
- How Kubernetes changes things
- Positive security model
- Traffic level
- Application level

© COPYRIGHT 42CRUNCH

API Breaches are on the rise!

- 300+ breaches reported on apisecurity.io since Oct. 2018
- And those are just the public ones!
- Recurrent combination of:
  - Lack of Input validation
  - Lack of Rate Limiting
  - Data/Exception leakage
  - Data Access authorisation flaws (IDOR/ BOLA)

Hacking Starbucks and Accessing Nearly 100 Million Customer Records

June 20, 2020 samwcyo

© COPYRIGHT 42CRUNCH | CONFIDENTIAL

Facebook - 50 million users' personal information was exposed

PayPal - 1.6 million customers at risk of data exposure

T-Mobile - 76 million users' phone numbers and addresses stolen

Instagram - 49 million users' emails and phone numbers exposed

Uber - 57 million riders and drivers accounts were compromised

Justdial - Over 100 million Indian users' personal data at risk

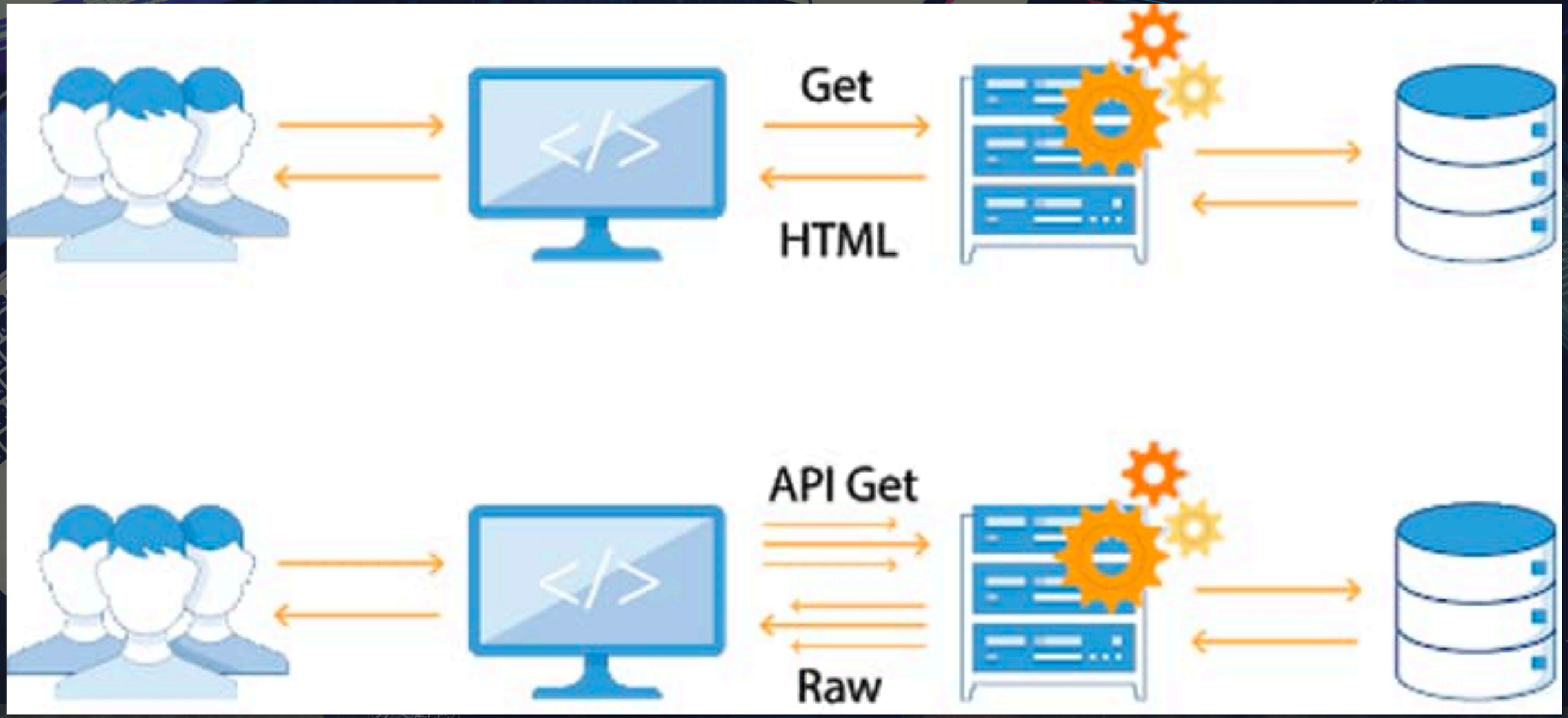
Equifax - 147 million users personal data stolen

Starbucks - 100 million customer records accessed

Verizon - 14 million subscribers phone numbers and PINs exposed

© COPYRIGHT 42CRUNCH | CONFIDENTIAL

Applications Architecture has changed!



Source: https://apisecurity.io/encyclopedia/content/owasp/owasp-api-security-top-10.htm

“By 2021, exposed APIs will form a larger surface area for attacks than the UI in 90% of web-enabled applications.”

- Gartner, API Strategy Maturity Model -

*https://www.gartner.com/en/documents/3970520

© COPYRIGHT 42CRUNCH

FROM PROTECTING THE PERIMETER...

...TO PROTECTING THE DATA

MANY APIS, DEPLOYED OFTEN

APPLICATION DEVELOPMENT

- ✓ *Fast*
- ✓ *Agile*
- ✓ *Automated*

APPLICATION SECURITY

- ✗ *Too late to the party*
- ✗ *Unadapted Tools*
- ✗ *Manual Reviews*

© COPYRIGHT 42CRUNCH | CONFIDENTIAL

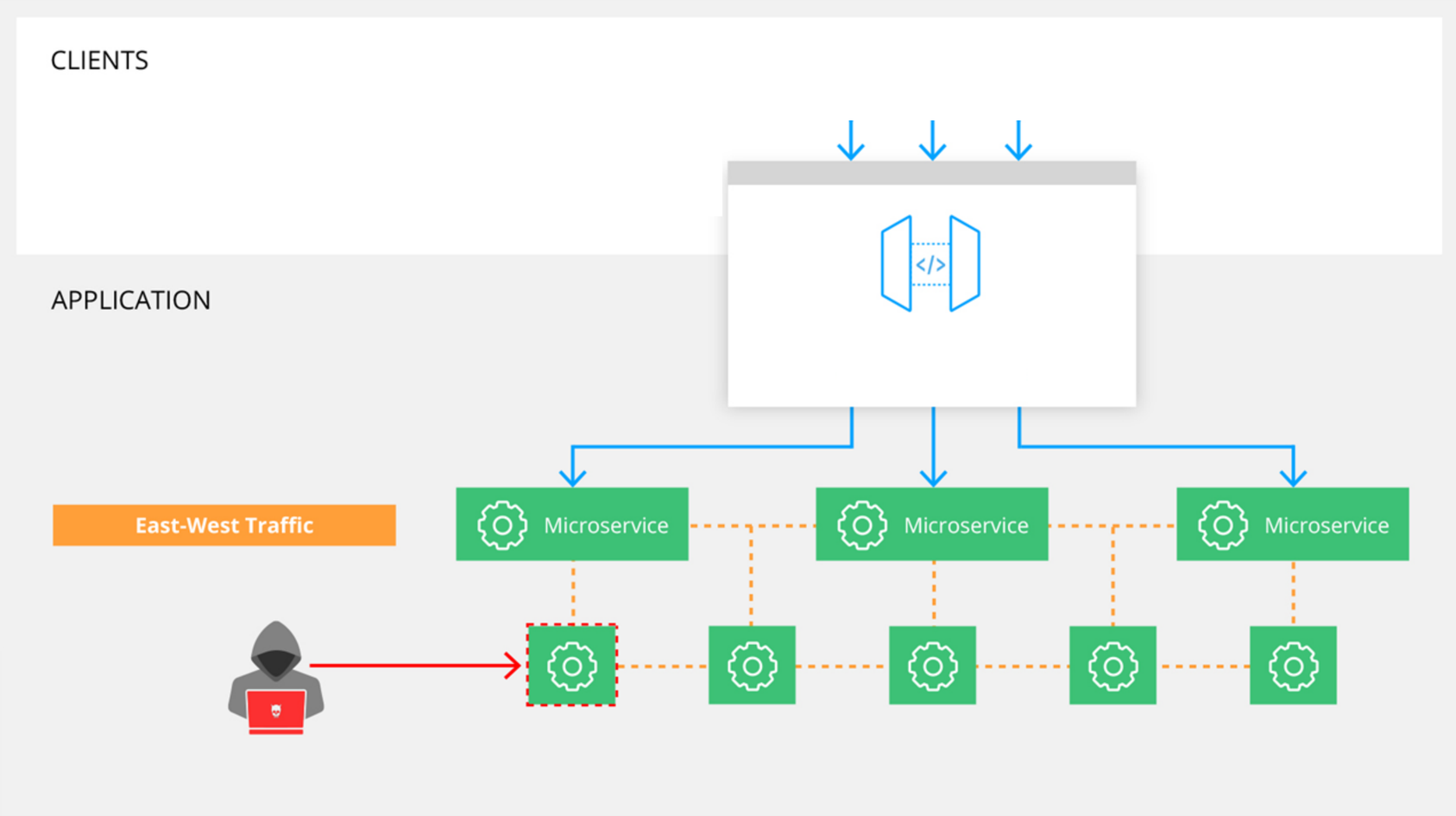
Positive Security Model

- Define expected and reject the rest
- Who can talk to whom
- Authentication and authorization
- Expected data coming in and going out

© COPYRIGHT 42CRUNCH

On the network level

- Which microservices should be accessible to external calls?
- Which microservices are supposed to call which?
- Reject everything else



© COPYRIGHT 42CRUNCH

mTLS and Service Meshes can help define and enforce

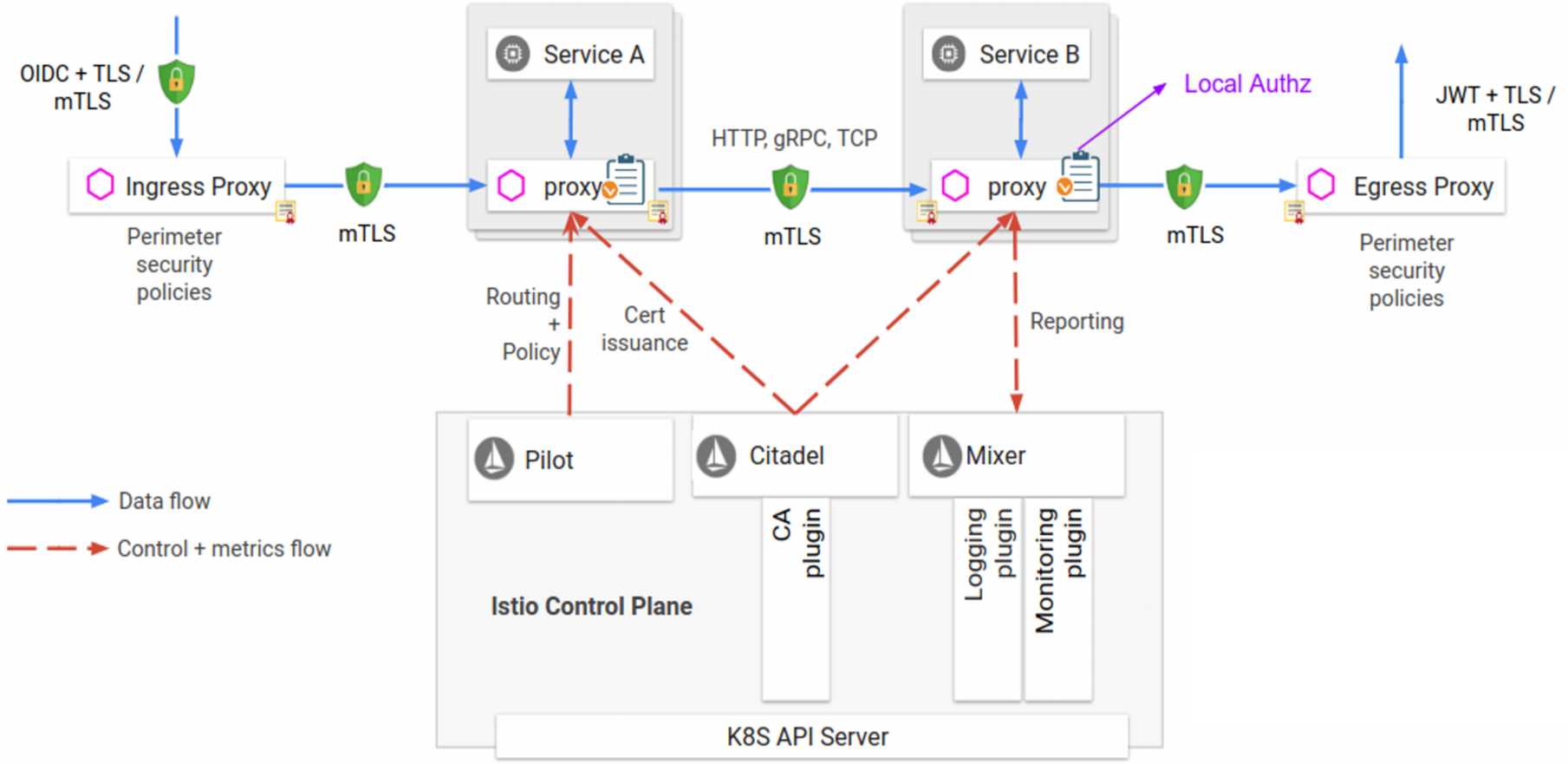


Diagram from Philippe De Ryck: https://pragmaticwebsecurity.com/talks/recipeapiauth.html

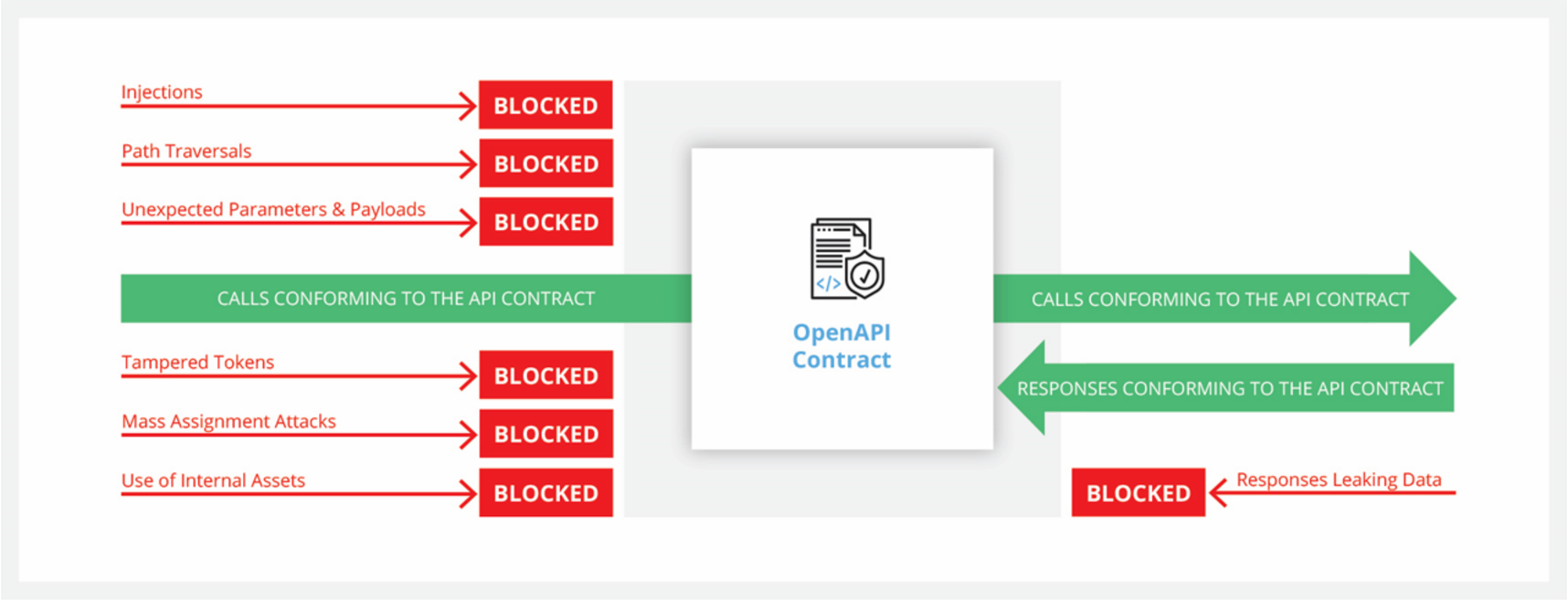
© COPYRIGHT 42CRUNCH

At the API level

- Every API needs authentication (who is calling ?)
- Deciding which one you need depends on the risk
- Authorization for data and functionality access (which resources do they have access to?)

© COPYRIGHT 42CRUNCH

Positive security on API call and response level



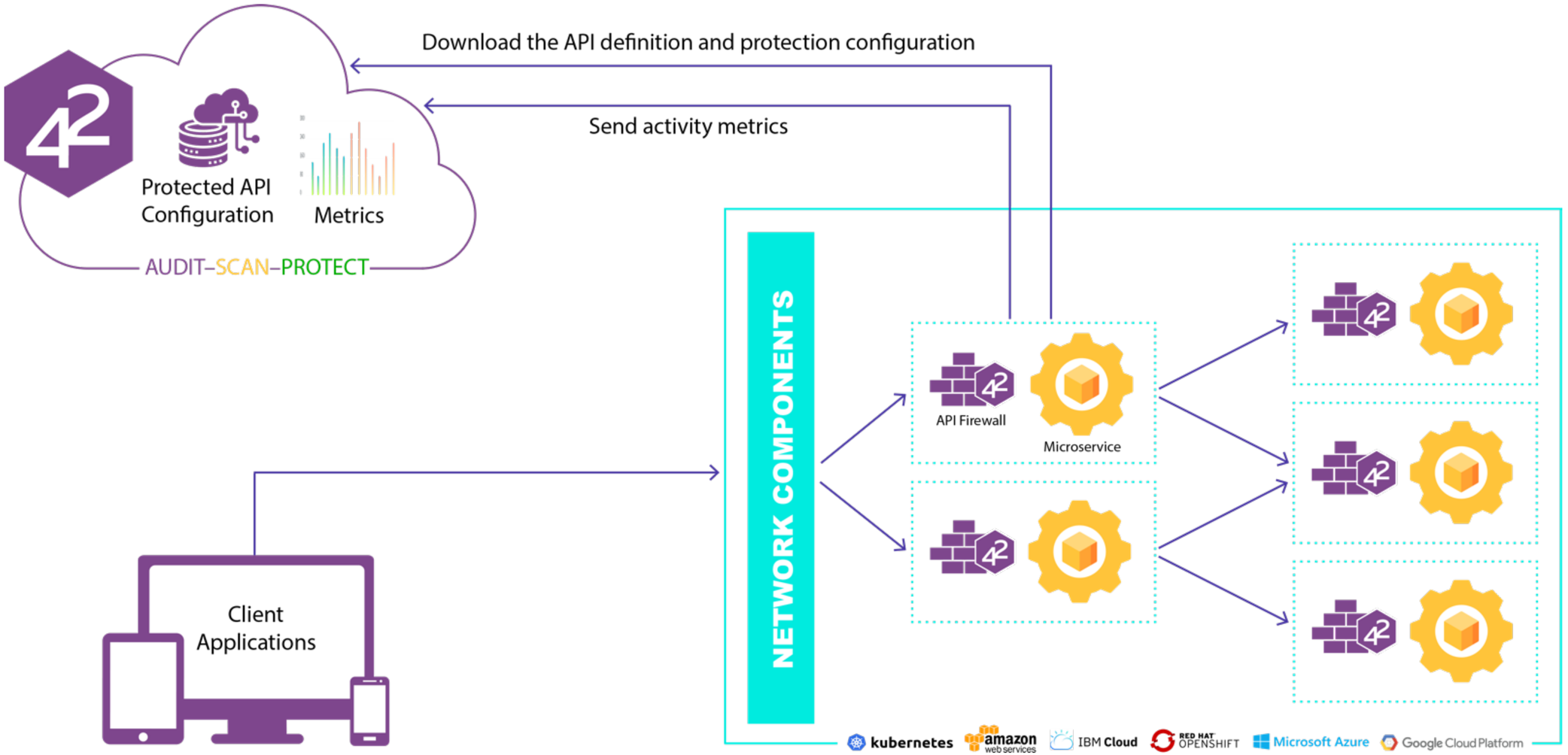
© COPYRIGHT 42CRUNCH

Good vs Bad Contracts

- Define security
- Define headers, parameters, payloads, responses
- Don’t allow additional parameters
- Define every schema, every element
- Define formats, limits, patterns

© COPYRIGHT 42CRUNCH

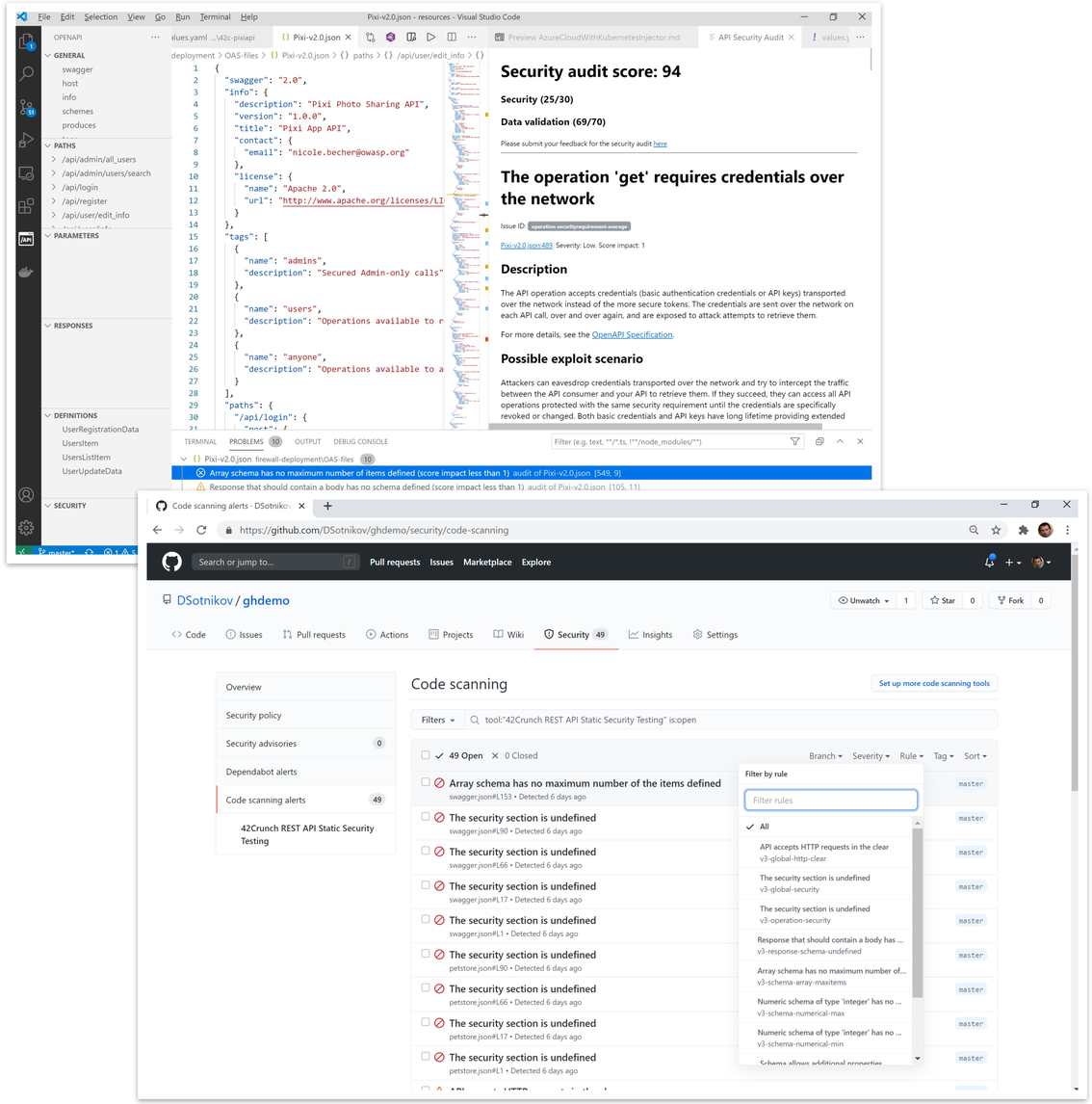
Protection Deployment: Sidecar Mode



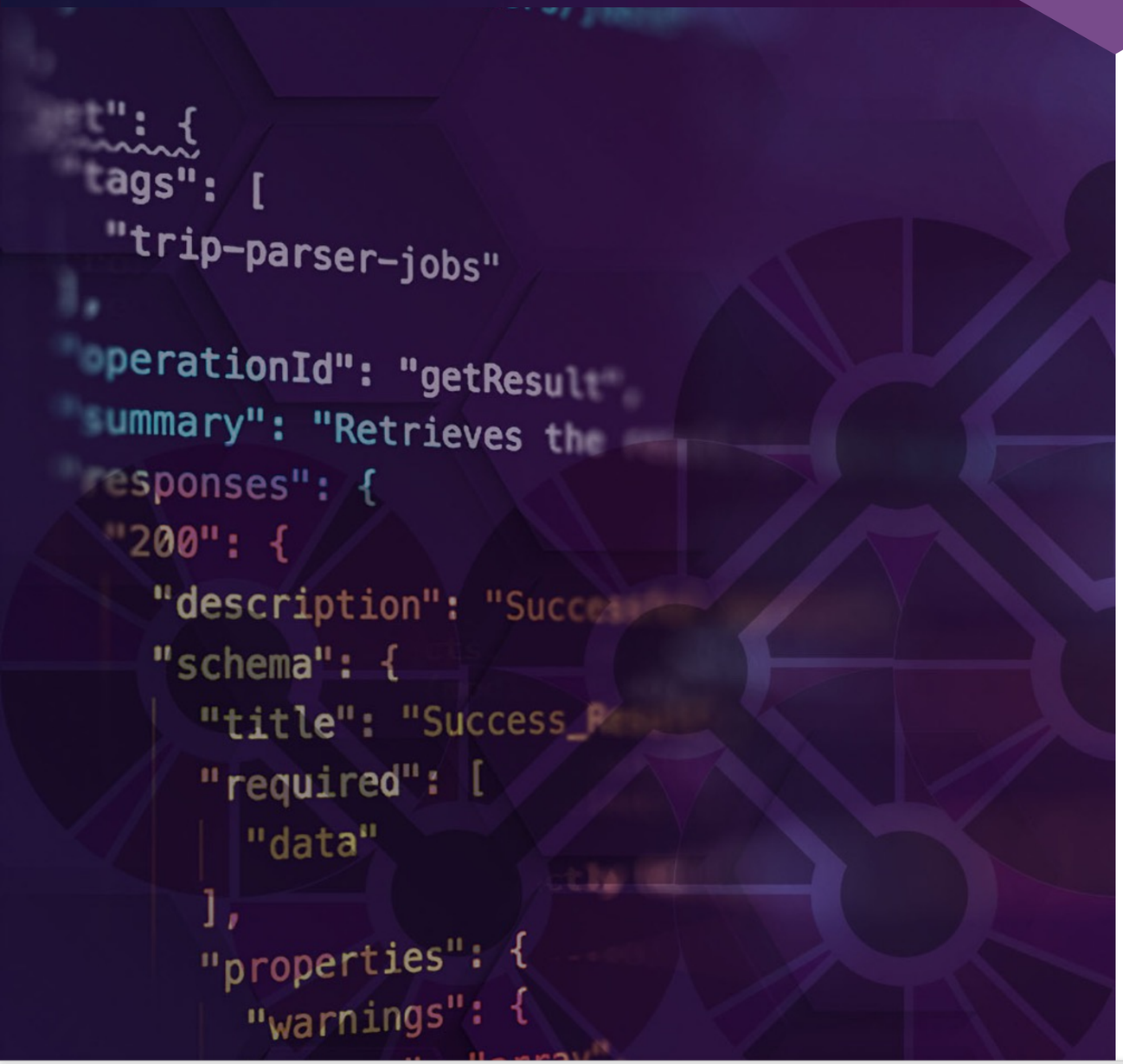
© COPYRIGHT 42CRUNCH

Plenty of tools available

- Standalone OpenAPI security audit
- Plugins for VS Code, IntelliJ, Eclipse
- GitHub Actions
- Sonarqube
- CI/CD pipeline plugins: Azure DevOps, BitBucket, Jenkins, Bamboo



© COPYRIGHT 42CRUNCH



Summary

- Use positive security model
- Define and enforce communication routes (mTLS, etc.)
- Define and enforce all API calls and responses
- Automate with DevSecOps

© COPYRIGHT 42CRUNCH

Thank you!

Contact us | info@42crunch.com | 42crunch.com

Want to see a personalized demo ?

https://42crunch.com/request-demo/