



Safeguarding APIs

ENSURING SECURITY IN CONNECTED ECOSYSTEMS

Shain Singh, Principal Security Architect @ NGINX (part of F5)

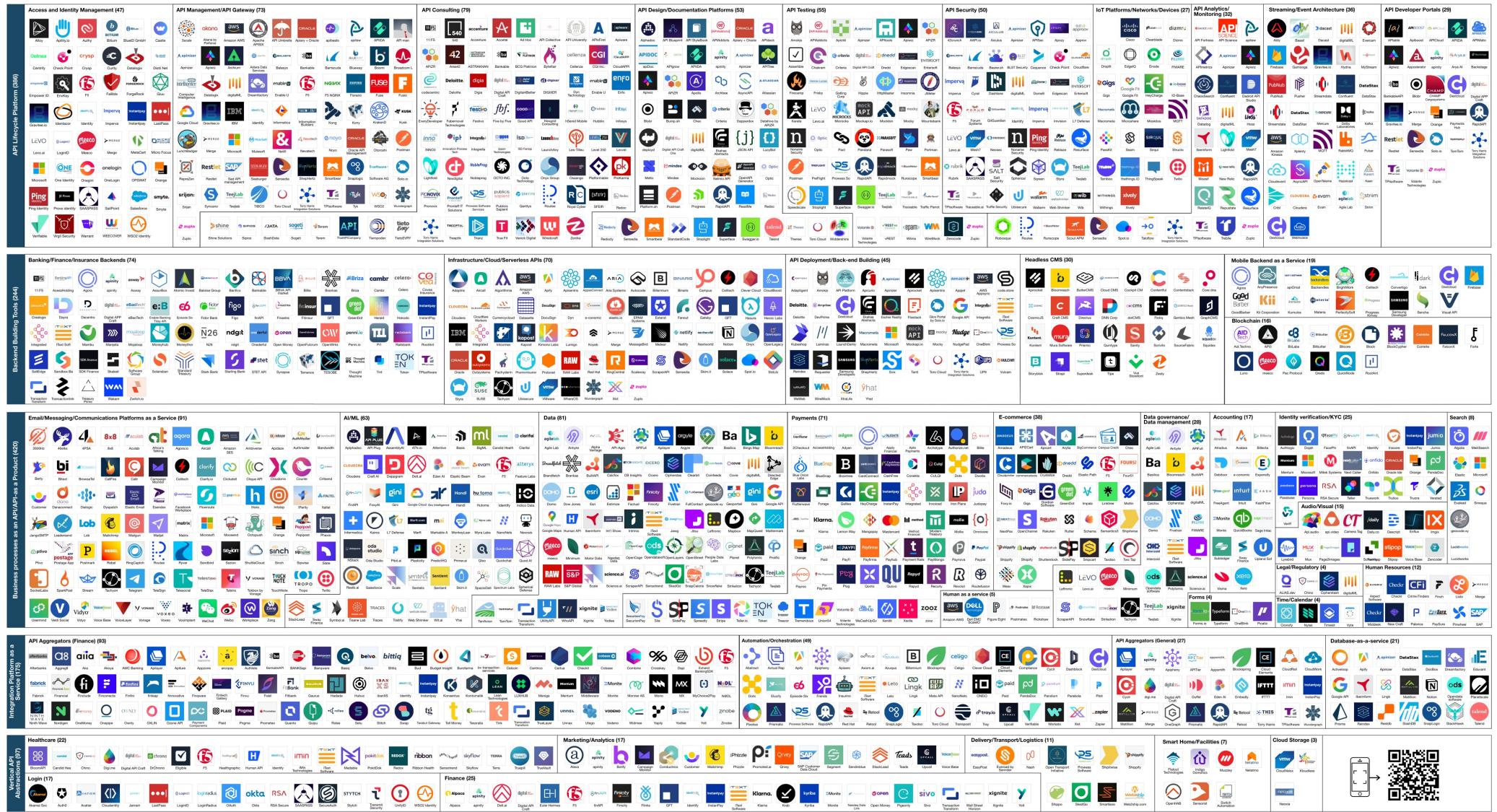
How we tend to think of the API ecosystem



A comprehensive view of all stakeholders creating the programmable economy

Last Update: December 2022

1157



SecOps need to improve for enabling speed

70% Of developers admit to skipping security due to delivery timeframes

81% Of developers admit to pushing code with known vulnerabilities

96% Of cloud breaches are self-inflicted

curl –L whois.shain.io

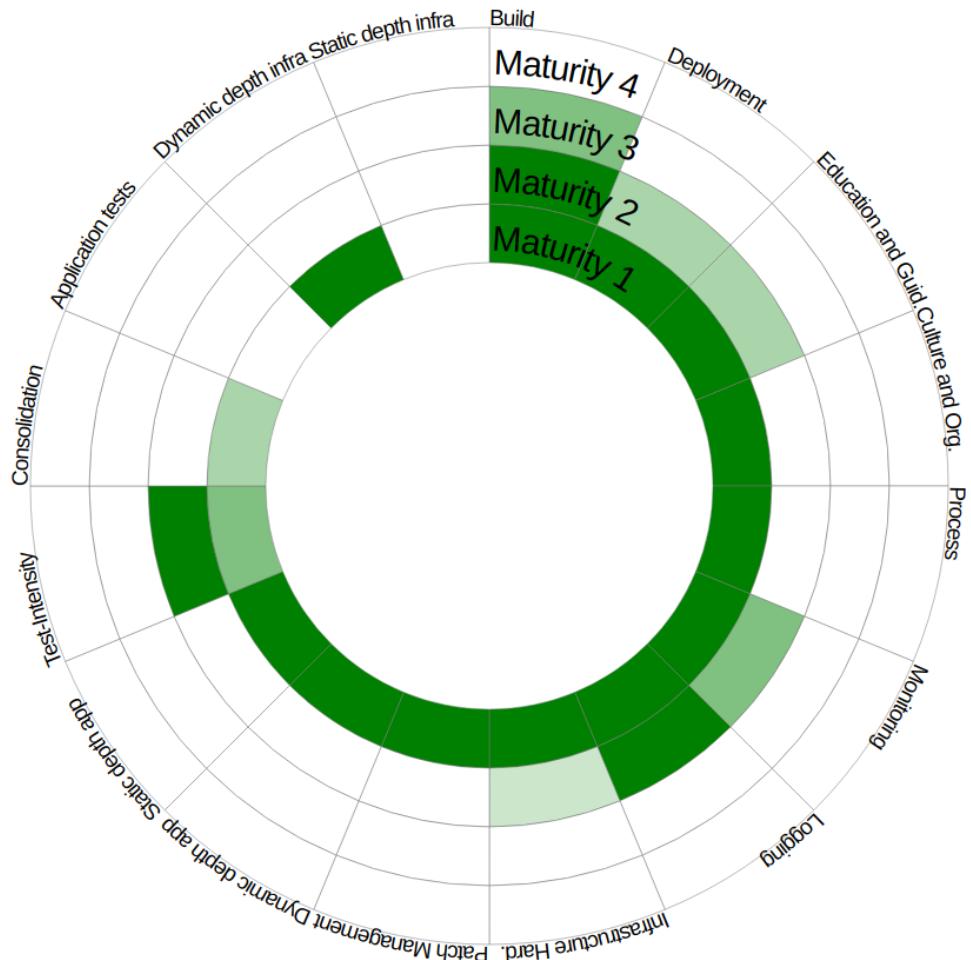


```
{  
  "$schema": "https://raw.githubusercontent.com/jsonresume/resume-  
  schema/v1.0.0/schema.json",  
  "basics": {  
    "name": "Shain Singh",  
    "label": "Principal Security Architect, OCTO, OSPO @ F5 | Project Co-Lead @  
    OWASP"  
  },  
  "profiles": [  
    {  
      "network": "LinkedIn",  
      "url": "https://www.linkedin.com/in/shsingh/"  
    },  
    {  
      "network": "Twitter",  
      "url": "https://twitter.com/shainsingh"  
    },  
    {  
      "network": "Github",  
      "url": "https://github.com/shsingh"  
    }  
  "work": [  
    {  
      "name": "OWASP® Foundation",  
      "position": [  
        "Project Co-Lead - Machine Learning Security Top 10",  
        "Project Co-Lead - Machine Learning Security Verification Standard"  
      ]  
    },  
    {  
      "name": "Cloud Security Alliance",  
      "position": [  
        "DevSecOps Working Group",  
        "Zero Trust Working Group"  
      ]  
    },  
    {  
      "name": "F5",  
      "position": [  
        "Open Source Program Office (OSPO) - Open Source Advocate",  
        "Office of CTO (OCTO) - Innovation Ambassador",  
        "Principal Security Architect [APCJ]"  
      ]  
    }  
}
```

Fine tuning versus Comprehensive Coverage

EMPHASIS HAS BEEN ON EFFICACY FOR INDIVIDUAL APPLICATIONS OVER FULL COVERAGE OF ALL DEPLOYMENTS

Identification of the degree of the implementation



DevSecOps Maturity Model (DSOMM) Level 1

Basic understanding of security practices

Recommendations:

- Never fail a build pipeline – security scans will have false positives
- Investigate static and dynamic tools for the DevOps pipeline
- Build expertise with tools and analyse results
- Collaborate with development teams to resolve issues

DevSecOps Maturity Model (DSOMM) Level 2

Adoption of basic security practices

Recommendations:

- Investigate tweaking tools from their default settings for tuning
- Storing results from tools in a consolidated environment
- Starting a security champion program

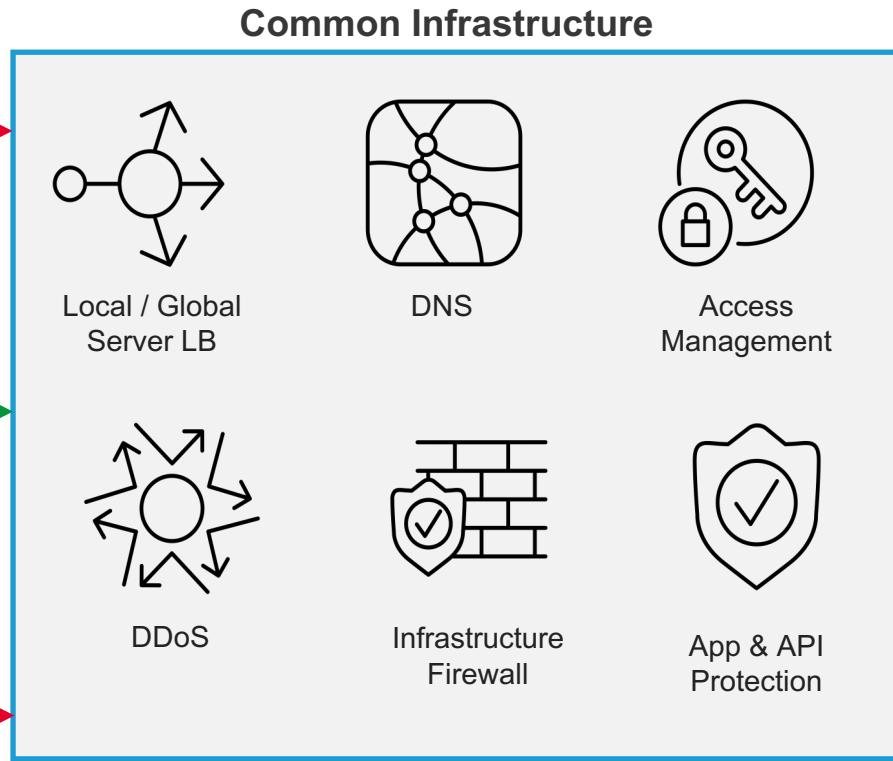
DevSecOps Maturity Model (DSOMM) Level 3

High adoption of security practices

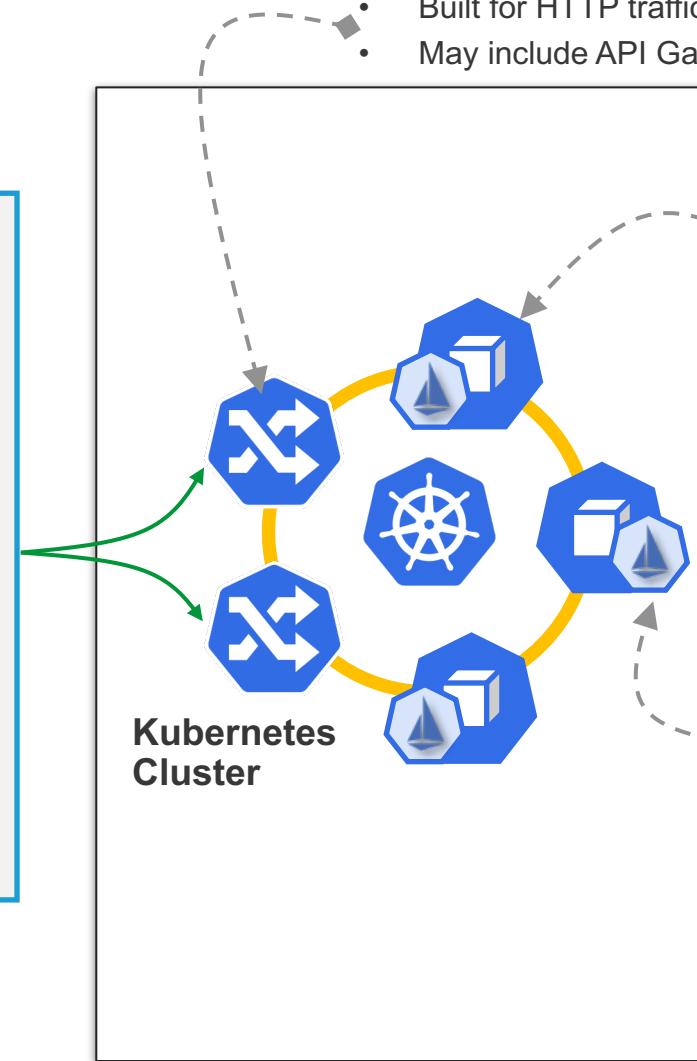
DevSecOps Maturity Model (DSOMM) Level 4

Advanced deployment of security practices at scale

Runtime environment for applications



*Shifting focus to
post-deployment*



Ingress (with API Gateway)

- Layer 7 routing for traffic entry point coming into Kubernetes
- Built for HTTP traffic. TCP/UDP for non-HTTP traffic
- May include API Gateway implementation

Pods

- Runs app in a container / CNF

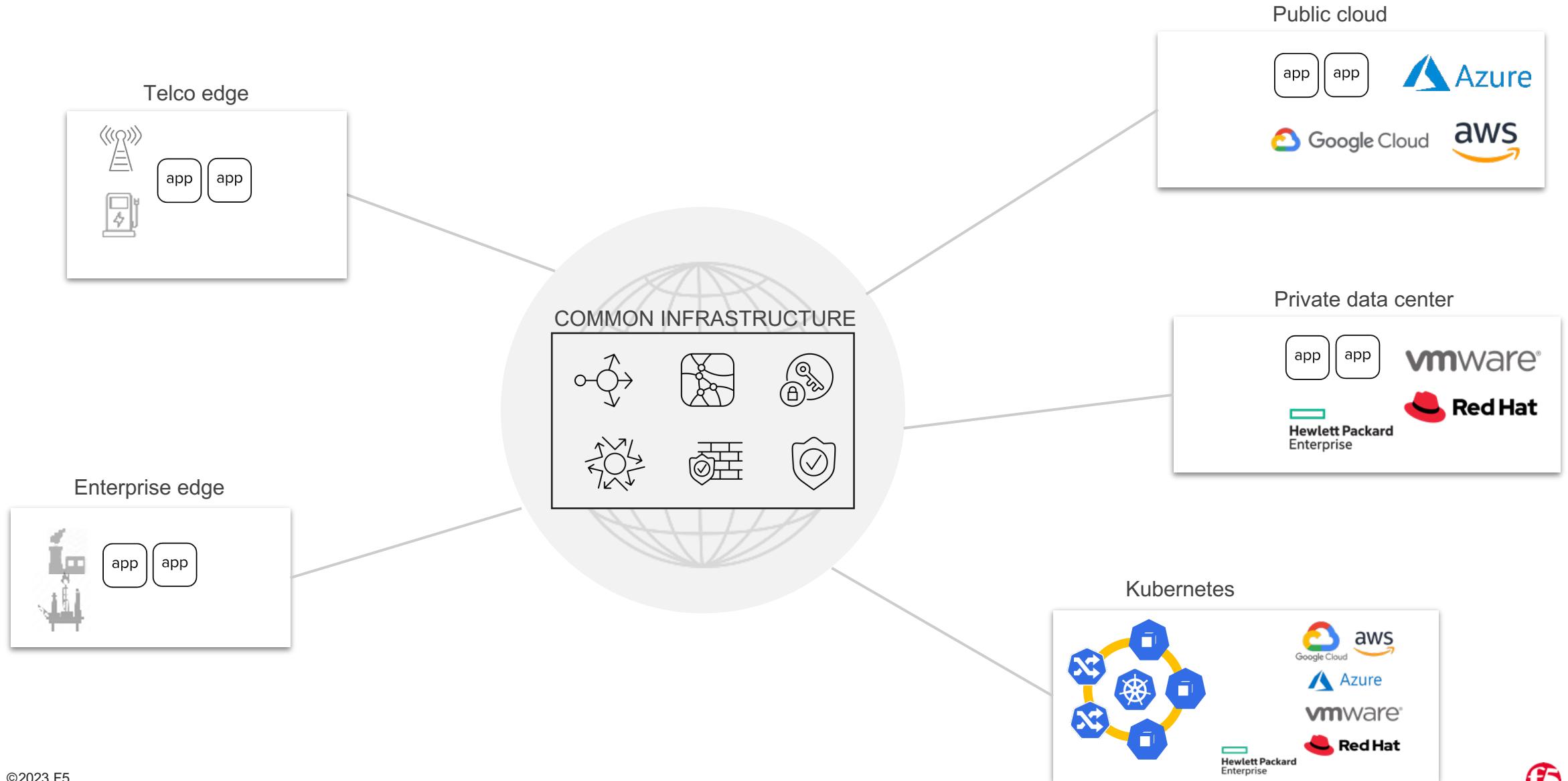
Service Mesh

- Open Source Service Mesh implementation (Istio)
- Injects Sidecar to every pod
- Enforces routing, security with mTLS, etc.
- Provides traceability of pod communication

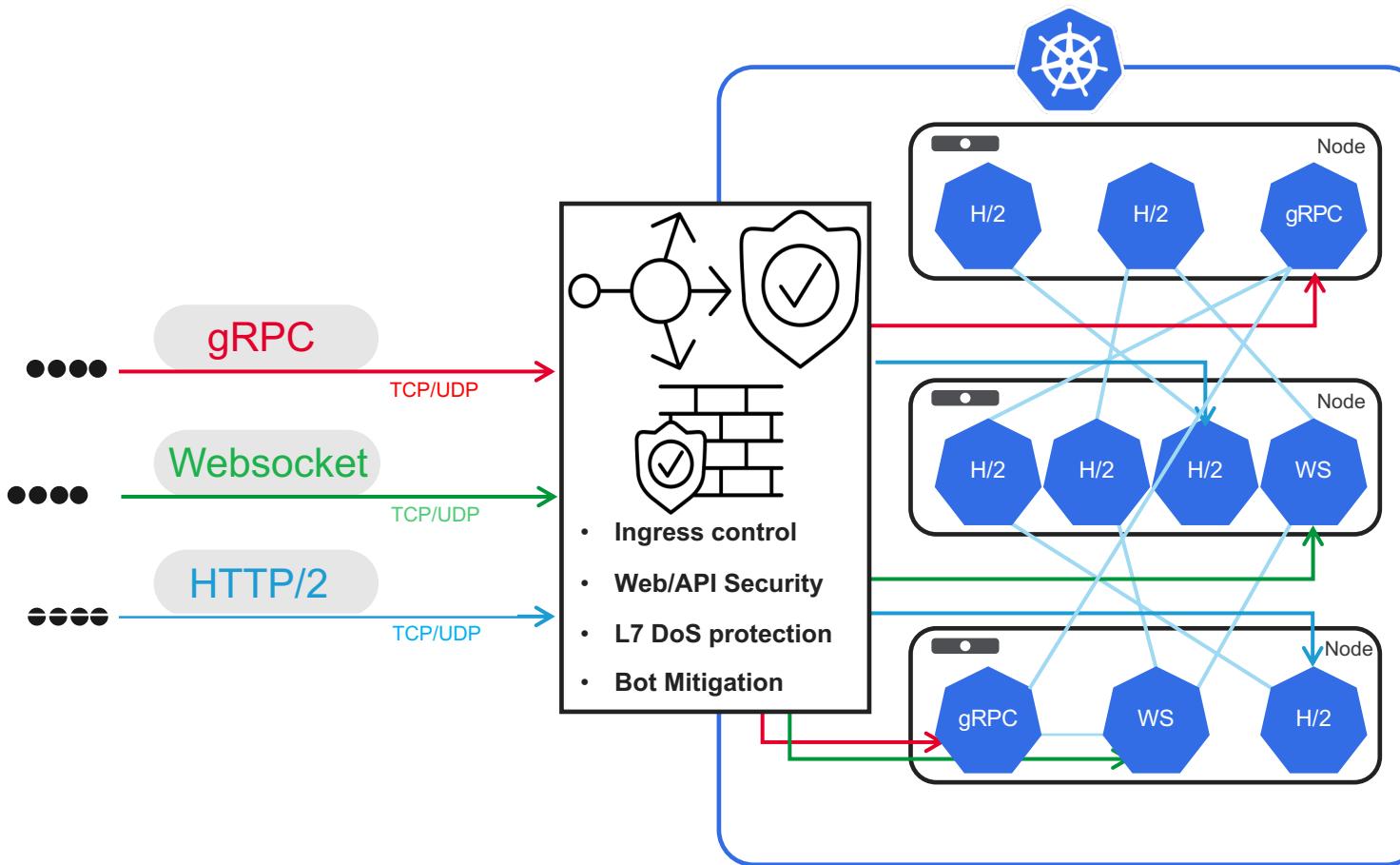
Cloud Microservices PaaS

- On-prem private cloud (e.g: VMware)
- Public cloud (e.g: AWS, Azure, GCP)

App Services at a macro-environment level



App Services at a micro-environment level



Security Controls:

- DoS protection for:
 - HTTP
 - gRPC
 - Websocket
- Web application and API security
- Bot Mitigation
- OpenAPI Spec (Swagger) enforcement
- Attack Signature/Schema Validation inside:
 - HTTP
 - XML
 - JSON
 - gRPC
 - Websockets
 - GraphQL
- TCP SYN flood protection
- AuthN/AuthZ

Security controls that are now mainstream

“Should I create ACLs for non-internet facing apps”

- Docker/Kubernetes service definitions
- Public cloud network ACLs (e.g. AWS security groups)

“Do I need encrypted data at rest and in transit for internal apps”

- Secrets management (e.g Hashicorp Vault, AWS KMS)
- mTLS via service mesh
- LetsEncrypt Certbot for TLS certificates

Can we not implement web application protection if we make deployment simple?

- WAF and L7 DoS configuration via Kubernetes manifests, deployed via Continuous Delivery tooling
- Functional testing of application post WAF deployment removes potential for false positives



Demonstration

<https://github.com/apcj-f5/nap-devsecops-demo>

Links for Demonstration

- GitOps repository
 - <https://github.com/apcj-f5/nap-devsecops-demo>
- Continuous Delivery
 - <https://build.f5labs.dev>
- Monitoring
 - <https://ops.f5labs.dev>
- Apps
 - Health Records API <https://hapi.f5labs.dev>
 - Bank <https://bank.f5labs.dev>
 - GraphQL <https://gql.f5labs.dev>



Repository with Security Controls

Security in CI/CD pipelines with NGINX App Protect

license Apache-2.0 repo status Active build checks passing deploy checks passing commit activity 52/month

 powered by semgrep  pre-commit.ci passed openssf scorecard 8.2 openssf best practices passing

ZAP Baseline Scan	ZAP Full Scan	ZAP API Scan
 hapi.f5labs.dev - ZAP Baseline Scan failing	 hapi.f5labs.dev - ZAP Full Scan failing	 hapi.f5labs.dev - ZAP API Scan failing
 bank.f5labs.dev - ZAP Baseline Scan failing	 bank.f5labs.dev - ZAP Full Scan failing	
 gql.f5labs.dev - ZAP Baseline Scan failing	 gql.f5labs.dev - ZAP Full Scan failing	

Application Security Policy applied at Ingress

nap-devsecops-demo / argocd / manifests / bank / bank-ingress-policy.yaml

shsingh enhance waf for bank app (#229)

Code Blame 12 lines (12 loc) · 265 Bytes Code 55% faster with GitHub Copilot

```
1 apiVersion: k8s.nginx.org/v1
2 kind: Policy
3 metadata:
4   name: "bank-ingress-policy"
5 spec:
6   waf:
7     enable: true
8     apPolicy: "bank-waf-default-blocking-policy"
9     securityLogs:
10       - enable: true
11         apLogConf: "bank-waf-log-policy"
12         logDest: stderr
```

nap-devsecops-demo / argocd / manifests / bank / bank-waf-default-blocking-policy.yaml

shsingh custom responsepage works - change back to default ✓ 7fd74cd · last month History

Code Blame 16 lines (16 loc) · 620 Bytes Code 55% faster with GitHub Copilot Raw ⌂ ⌄ ⌅ ⌆ ⌇

```
1 apiVersion: appprotect.f5.com/vbeta1
2 kind: APPolicy
3 metadata:
4   name: "bank-waf-default-blocking-policy"
5 spec:
6   policy:
7     name: "bank-waf-default-blocking-policy"
8     template:
9       name: POLICY_TEMPLATE_NGINX_BASE
10      applicationLanguage: utf-8
11      enforcementMode: blocking
12      # response-pages:
13      # - responsePageType: default
14      # responseActionType: custom
15      # responseContent: "The request URL was rejected. Please consult with your administrator. Your support ID is: <%TS.request.ID()%>"
16      # responseHeader: "HTTP/1.1 403 Forbidden\r\nCache-Control: no-cache\r\nPragma: no-cache\r\nConnection: close"
```

Functional Tests with Security enforcement

The screenshot shows a GitHub Actions workflow run for a repository named 'bank.f5labs.dev'. The workflow is titled 'Functional Testing #21' and was triggered manually 3 weeks ago by a user named 'leonseng' (commit hash: c965ad2). The status is 'Success' with a total duration of 38 seconds. There are no artifacts.

The workflow file is named 'bank.f5labs.dev-k6-tests.yaml' and is triggered on 'workflow_dispatch'. It contains a single job named 'run-k6-tests' which completed successfully in 29 seconds.

Below the workflow details, there is a 'run-k6-tests summary' section. This section includes a 'StepSecurity Report' which provides a preview of network events. The report highlights a 'Network Events' table:

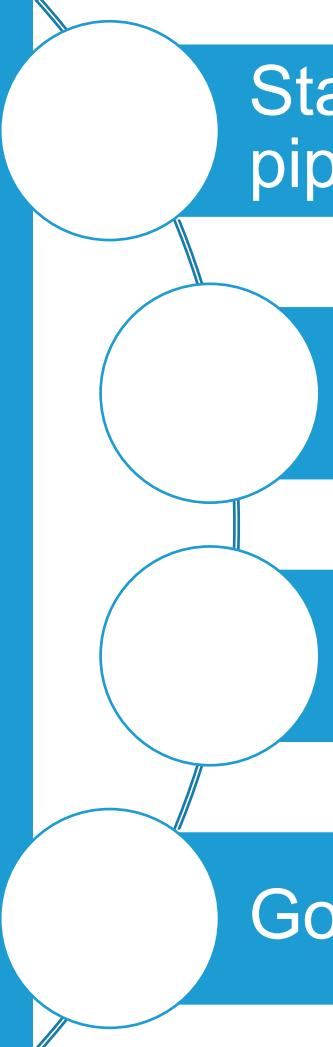
Process	Destination	Status
k6	reports.k6.io	✗ Blocked
node	api.github.com	✓ Allowed
git	github.com	✓ Allowed
...

At the bottom left, the page footer indicates '15 ©2023 F5'. The F5 logo is located at the bottom right.

Application Metrics Monitoring



Key Takeaways



Start incorporating runtime environment controls into pipelines for feedback loops

Start small, then increment – complete coverage over fine tuning

Integrate into DevOps processes as opposed to just installing security tooling

Goal is to have security across all apps, everywhere



A force for a better digital world