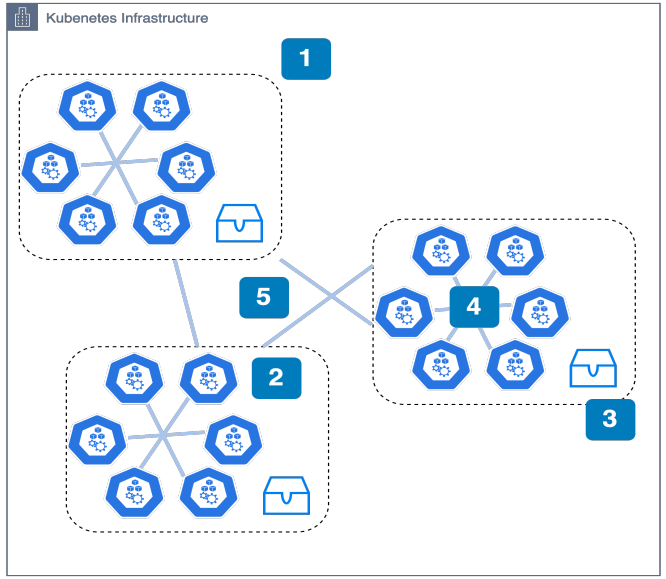
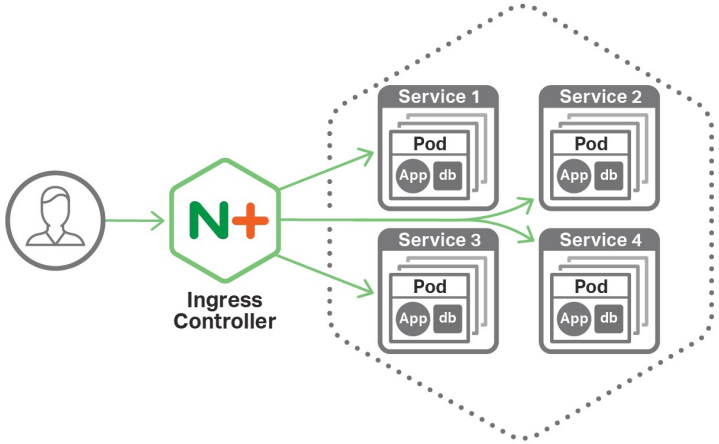
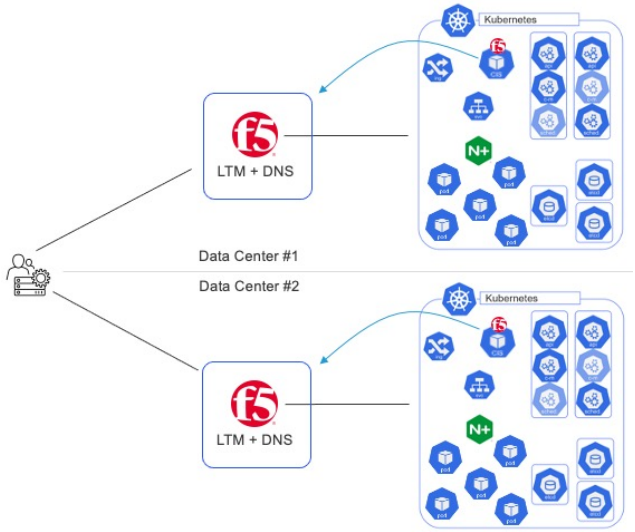
Recap



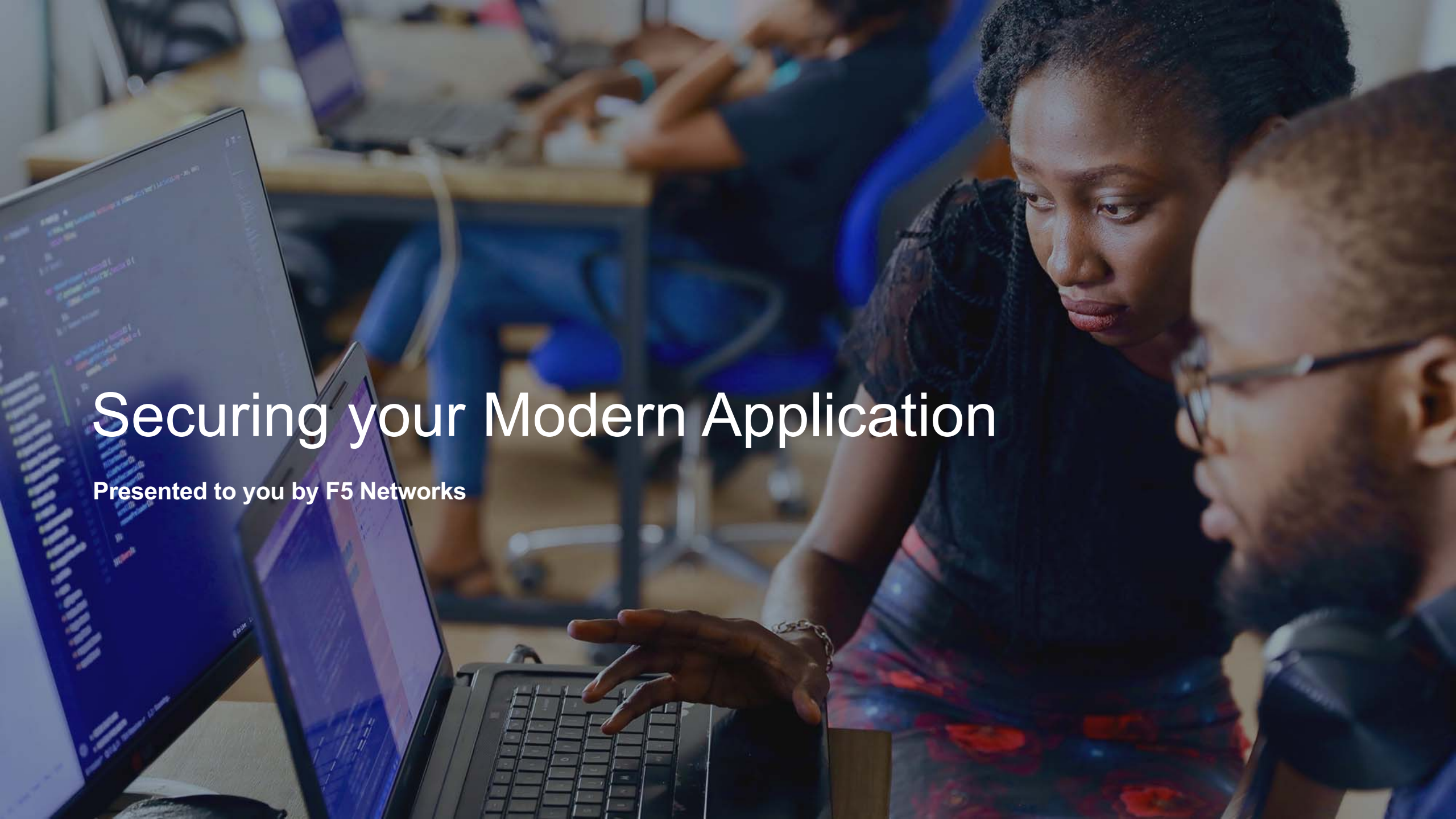




Kubernetes Basic Concept

High Resiliency

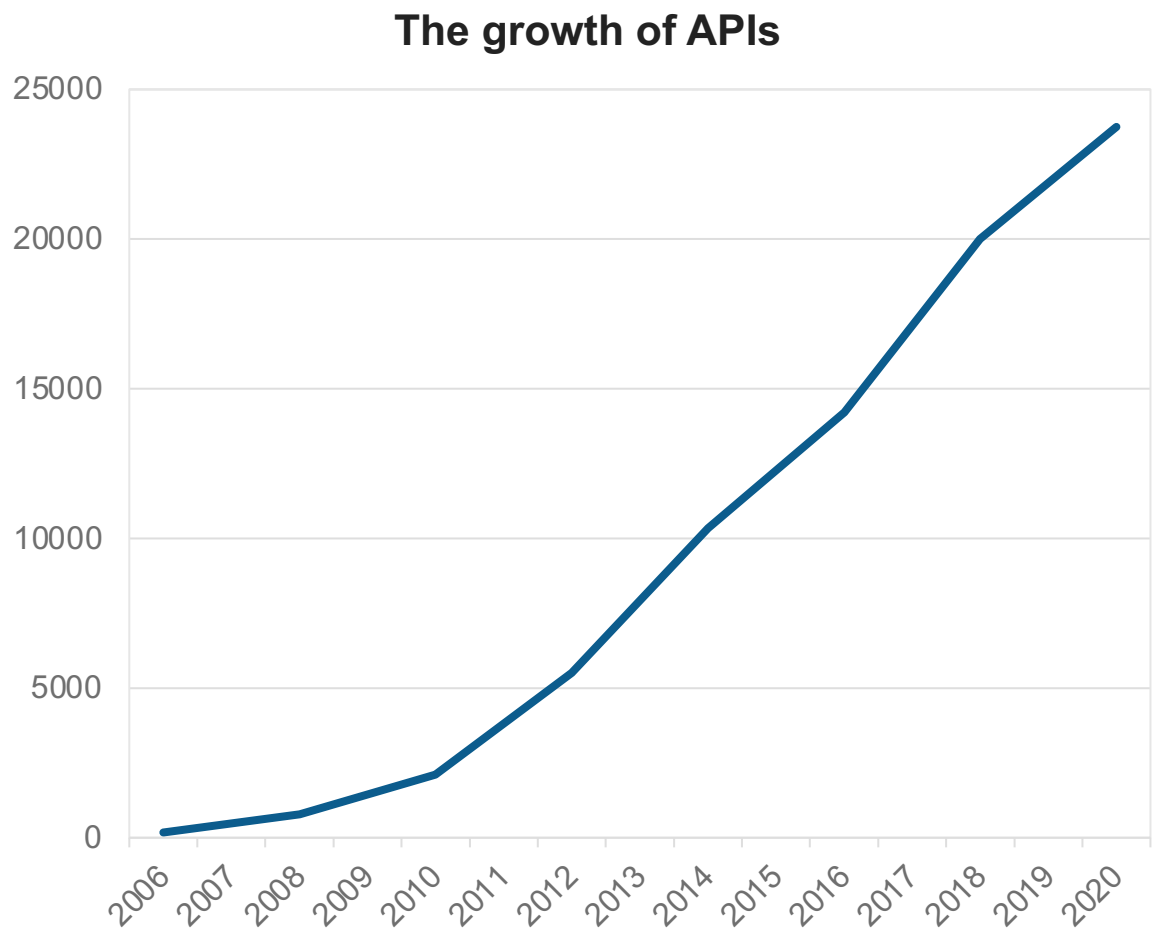
What's still missing?



Securing your Modern Application

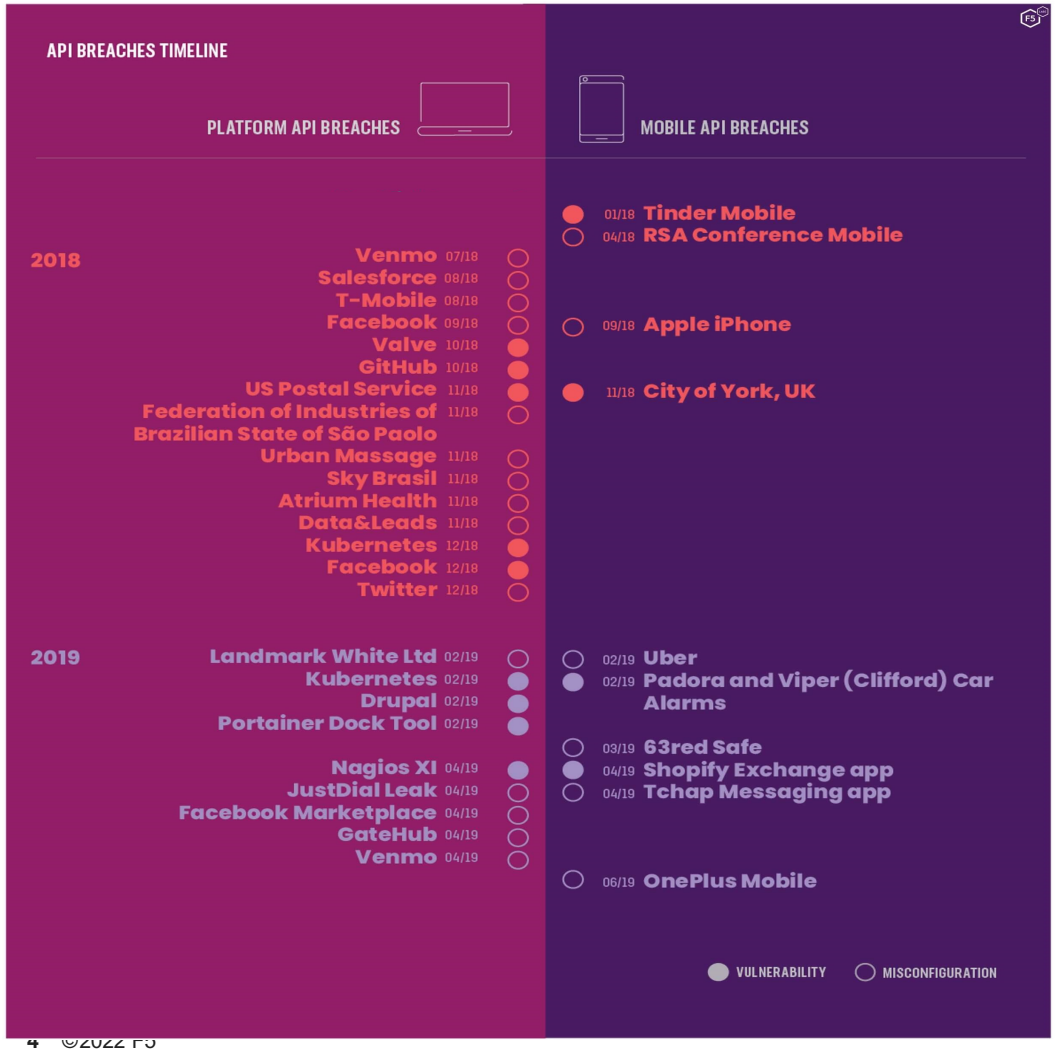
Presented to you by F5 Networks

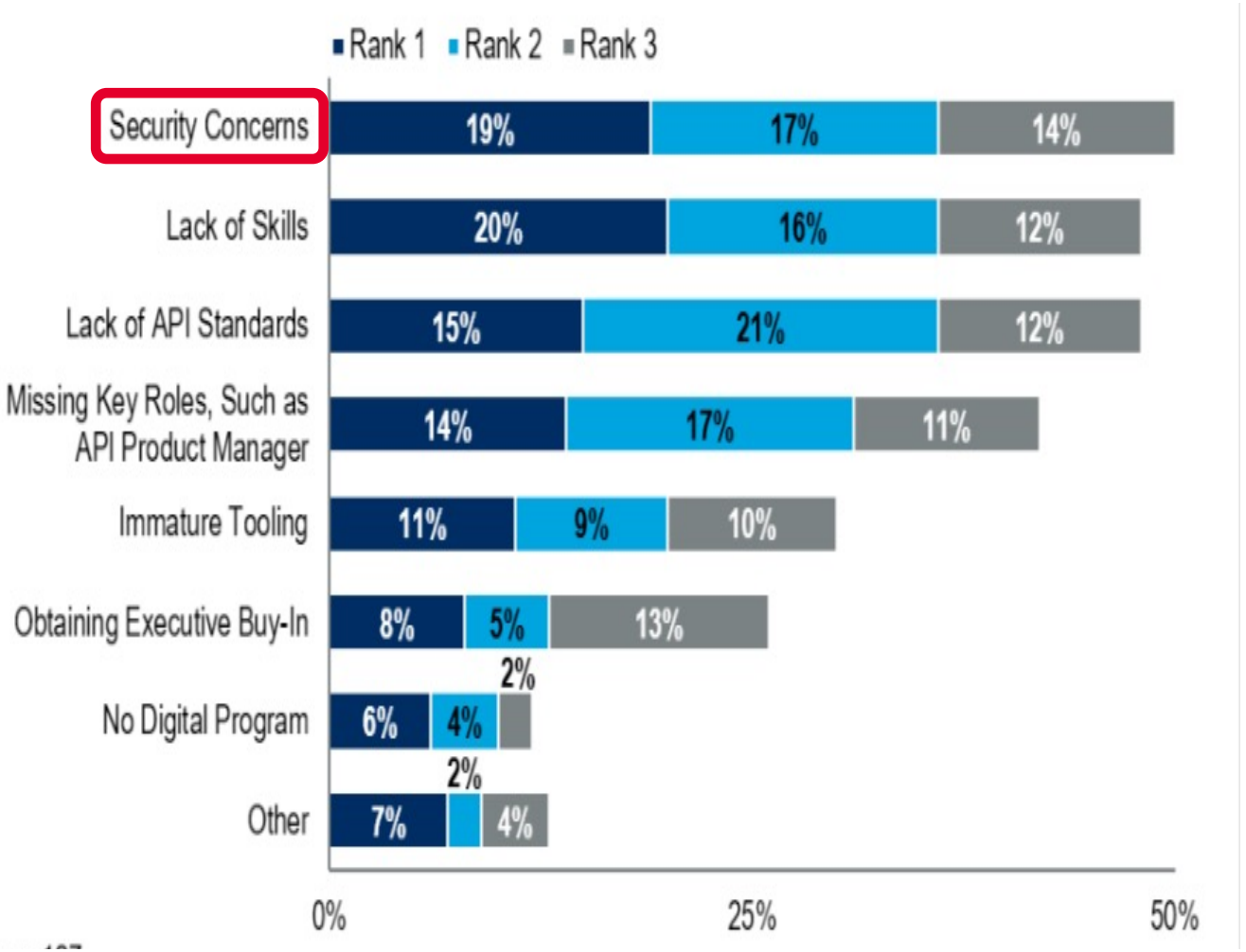
API at the Heart of The Application



Security is the Top Concern

API challenges for organizations





APIs in the crosshairs

Increased attack surfaces with large ecosystems and integrations

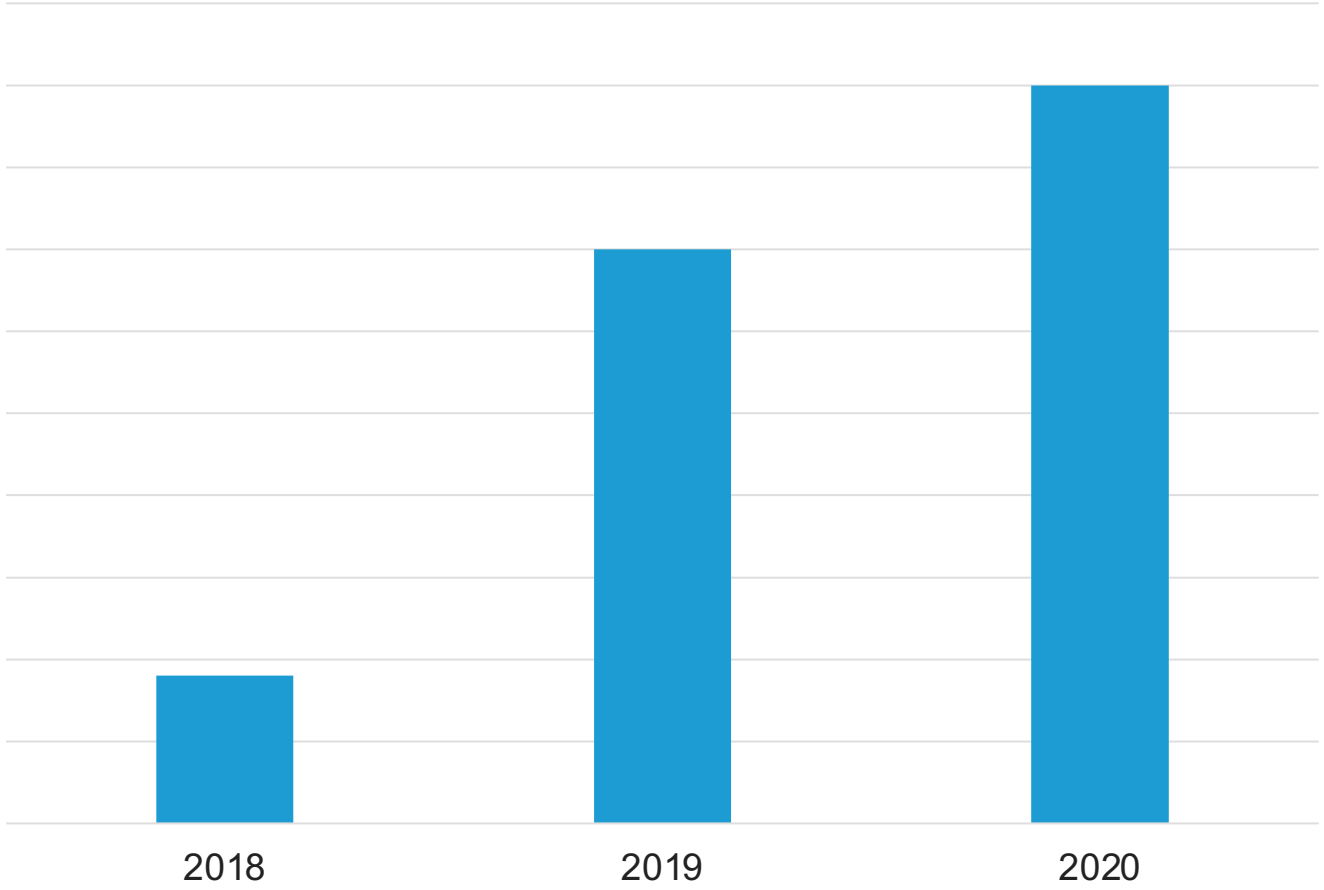
Most API attacks are authenticated users and are hard to detect

Attacked just like web apps, but without the same security controls in place

Often unknown to SecOps as different orgs publish and manage APIs on their own

API INCIDENTS

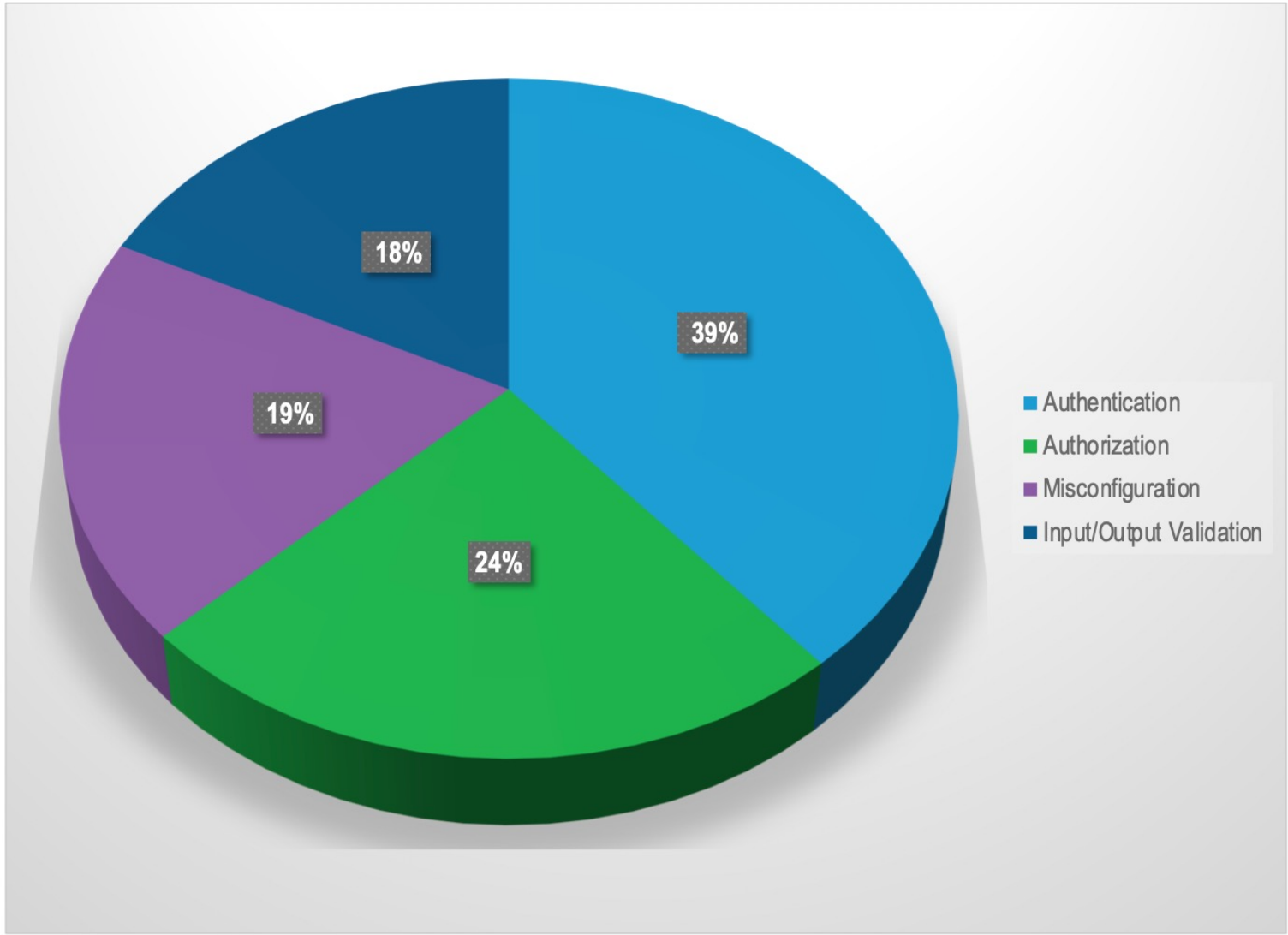
2018–2020 (JULY)



Top API Security Issues

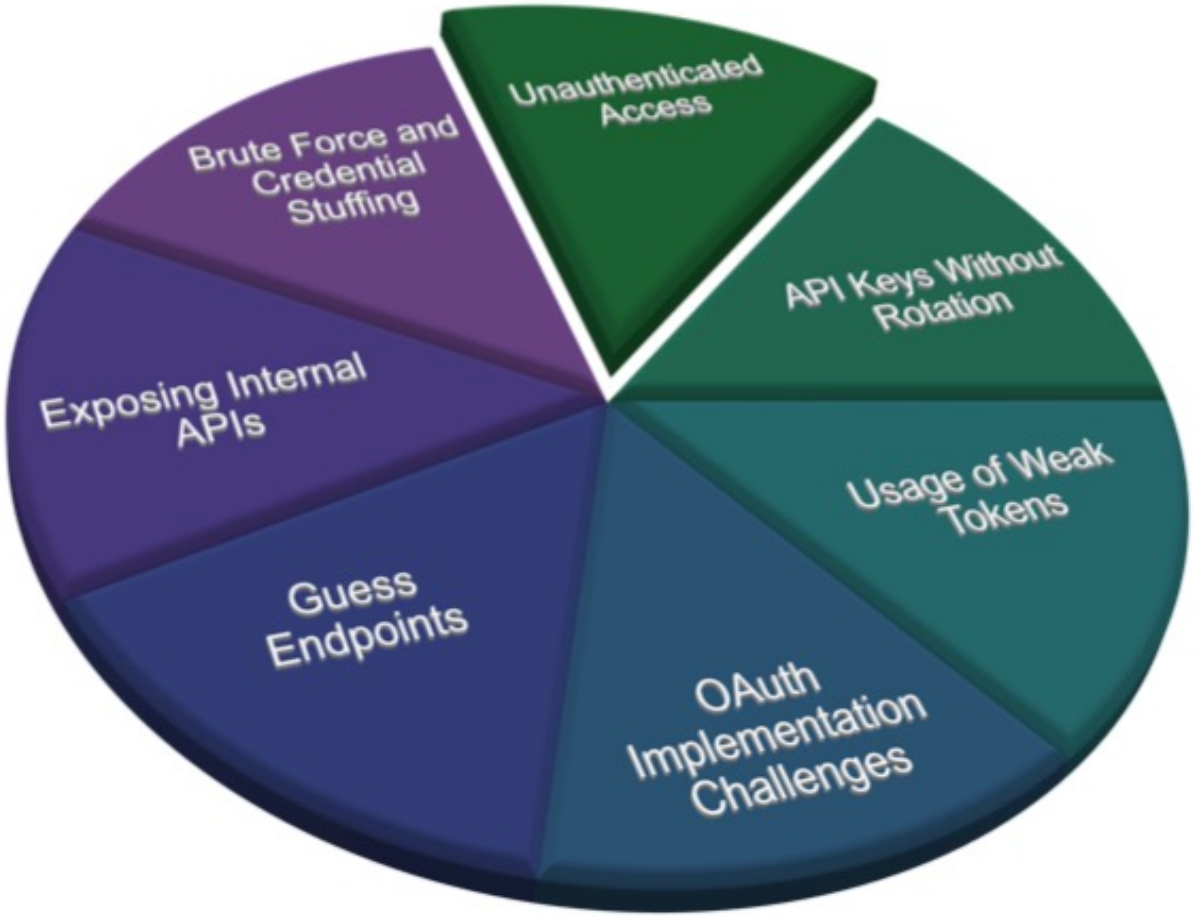
Breach Analysis 2022

- Authentication
- Authorisation
- Misconfiguration



Other Common Security Issues

Protect APIs like Web Apps



- Stolen API keys used to compromise clouds
- Unauthenticated Internal APIs led to loss of confidential data
- APIs allowed validation of stolen credentials
- Administrative API endpoint could be guessed and accessed without proper authorization

APIs need tailored security controls

Protect APIs with DATA and IDENTITY Controls

APIs NEED DEDICATED SECURITY CONTROLS THAT CAN ONLY BE DELIVERED WITH WEB APPLICATION FIREWALL (WAF) AND ACCESS MANAGEMENT

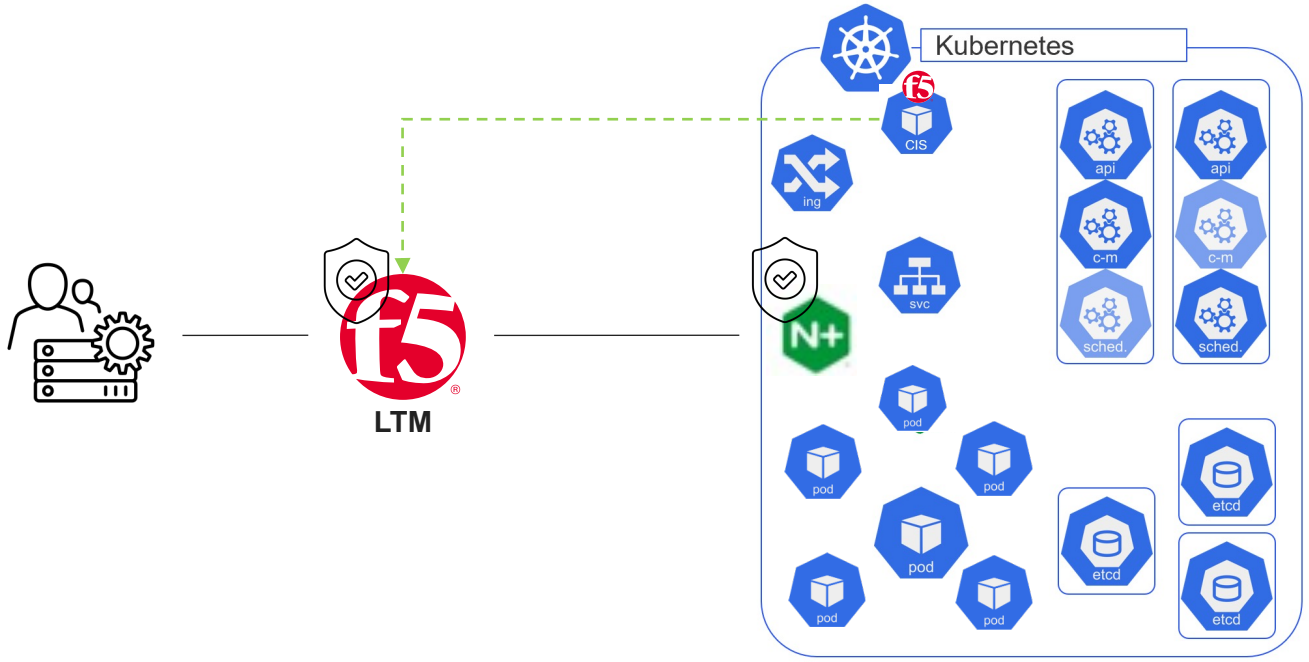
OWASP API SECURITY TOP 10

API1:	Broken object level authorization
API2:	Broken user authentication
API3:	Excessive data exposure
API4:	Lack of resources & rate limiting
API5:	Broken function level authorization
API6:	Mass assignment
API7:	Security misconfiguration
API8:	Injection
API9:	Improper assets management
API10:	Insufficient logging & monitoring

1. HTTPS — API Data Security
2. Access Control
3. JWT
4. API Keys — API Identity Security
5. Restrict HTTP Methods
6. Input Validation
7. Validate Content Type — API Data Security
8. Management endpoints
9. Error handling
10. Audit logs
11. Security headers — API Data Security
12. Cross-Origin Resource Sharing (CORS)
13. Sensitive information in HTTP requests
14. HTTP Return Code

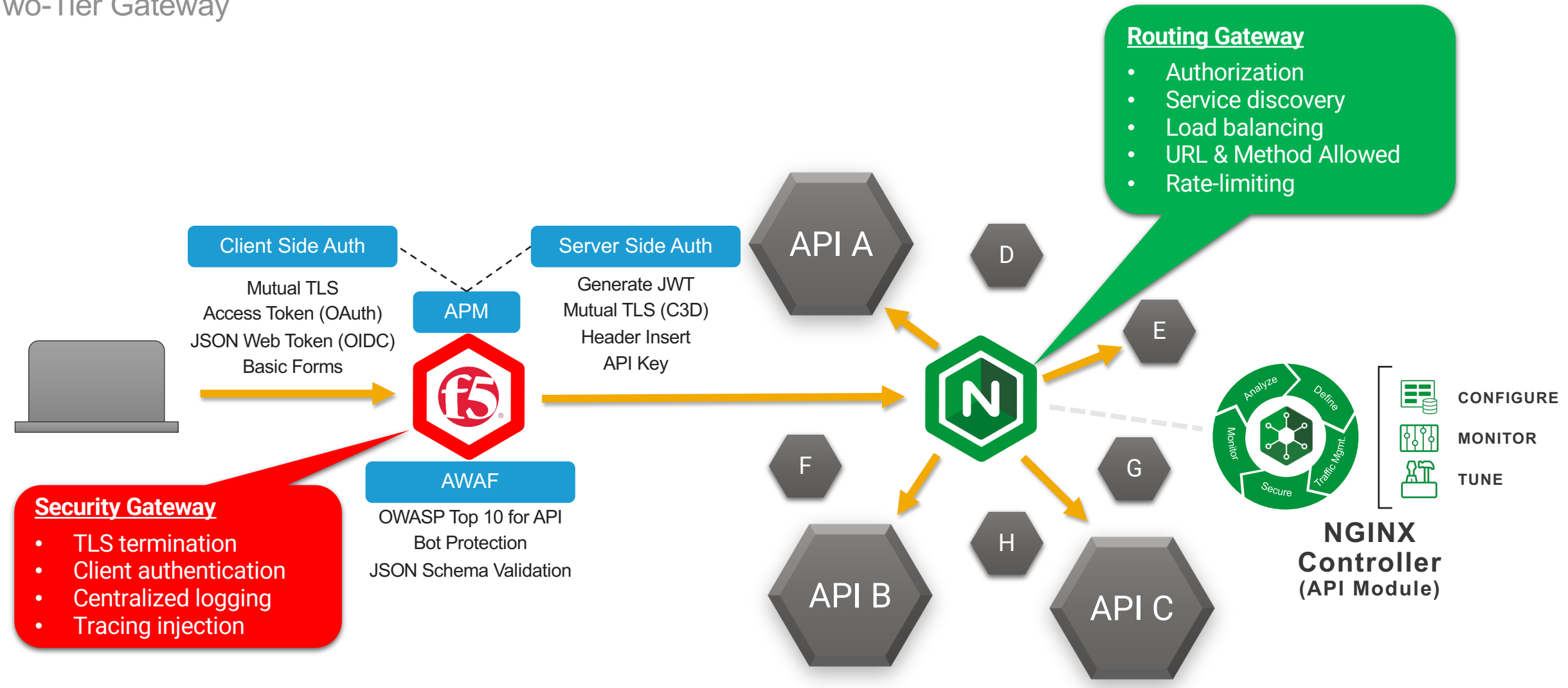
Security Placement

Where should we place the security?

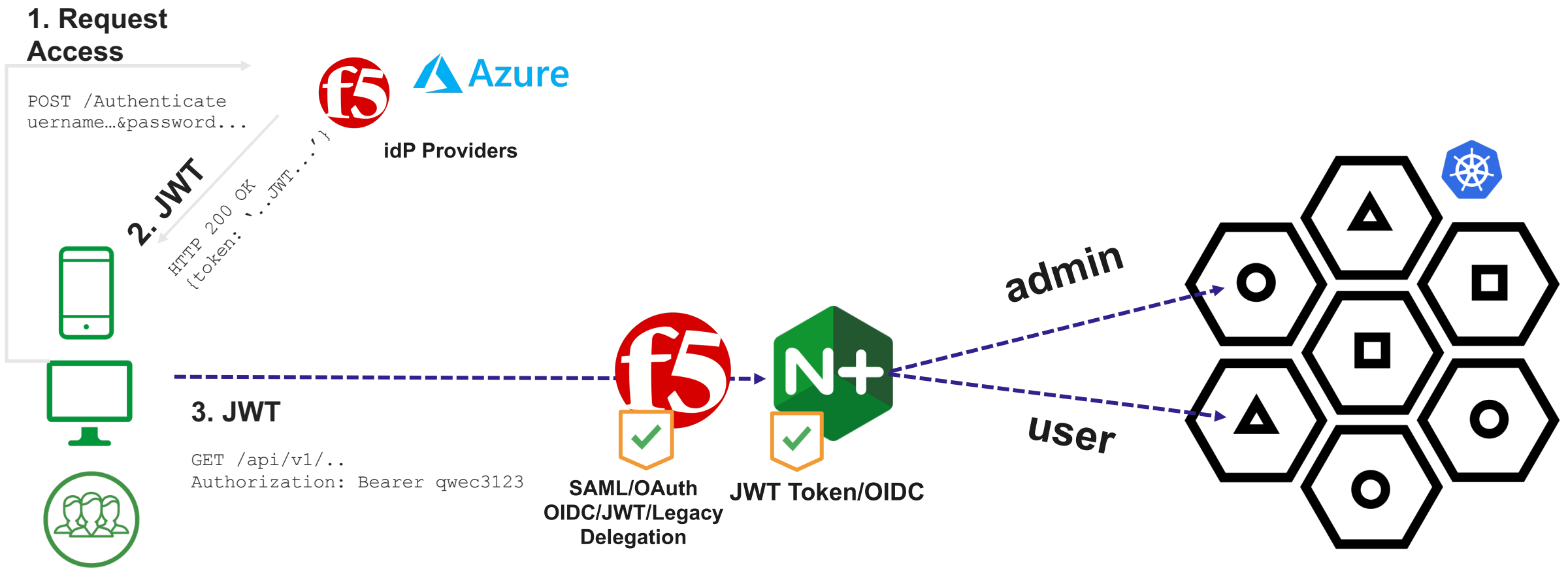


F5 & NGINX Solutions

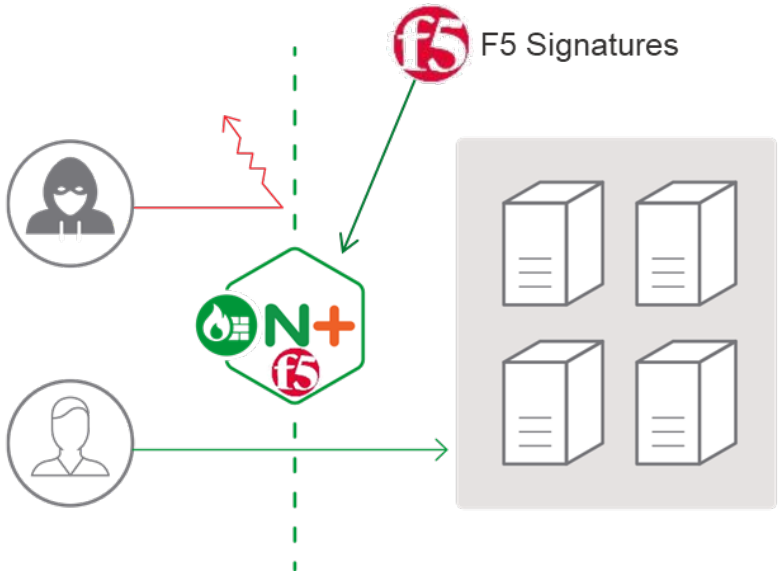
Two-Tier Gateway



AuthN/Z with Federation and Legacy



NGINX App Protect



Secure

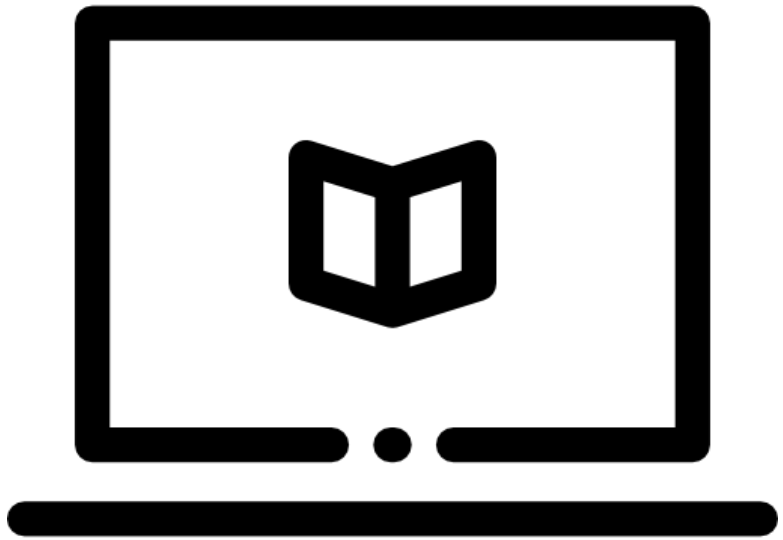
- ✓ High performing
- ✓ Security protection beyond signatures
- ✓ Trusted Signatures from F5

CI/CD Friendly

- ✓ Designed for modern infrastructures
- ✓ Security policy as code – easy integration into CI/CD pipeline and automated app delivery
- ✓ Rapid feedback loop for agile security remediation

Manage

- ✓ Same declarative interface
- ✓ Security statistics via syslog
- ✓ Backed by F5 Support



Lab Time

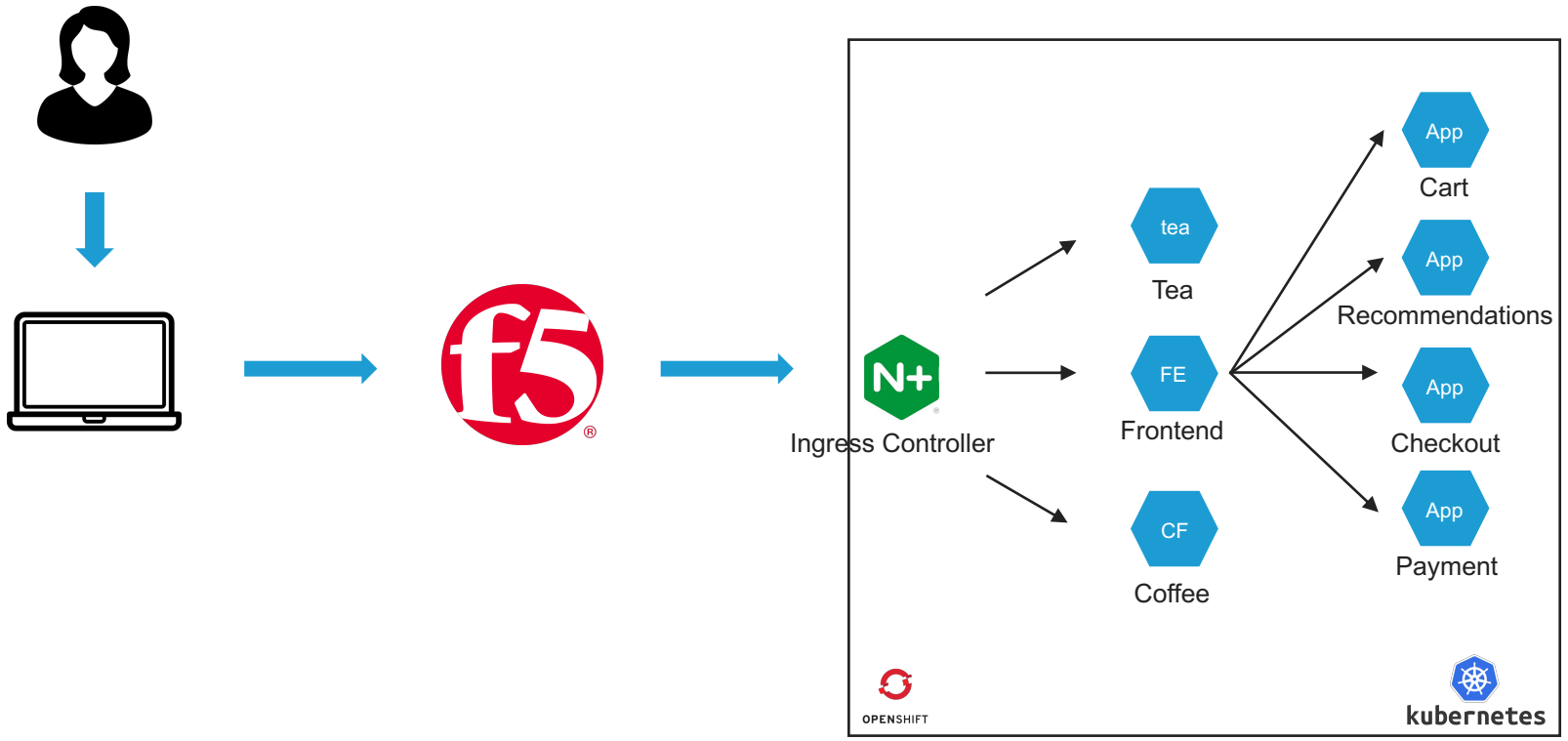
- Advanced Rate Limiting (targeted, bursting)
- Ingress mTLS
- JWT Authentication
- API Security Protection

Accessing the Lab

- An email sent. Course Name: **sg-k8s-bootcamp-2022-lab7-groupx**
- Lab guide is in the sharepoint folder shared in the email 2 days before.
- Launch the lab

Environment

Demo / Lab



Module 1: Advanced Rate Limit

A leaky bucket, FIFO queue

The burst value defines the size of the queue, which allows an exceeding number of requests to be served beyond the base limit. When the queue becomes full, the following requests will be rejected with an error code returned.

Advanced Rate Limiting Configuration

Rate Sets the maximum request rate

Key Defines the request characteristic against which the limit is applied

Zone Defines the shared memory zone used to store the state of each IP address

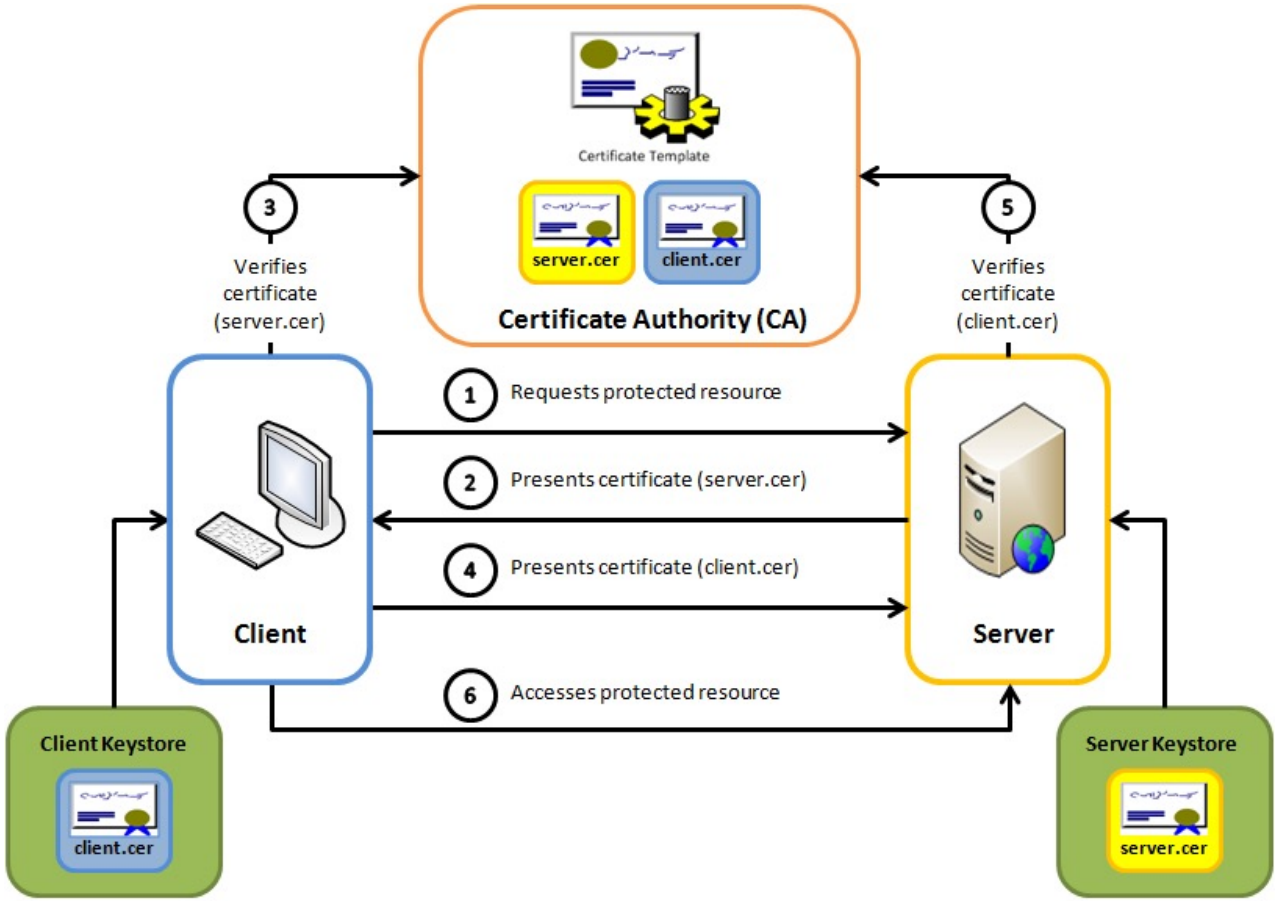
01-ratelimit-policy.yaml

```
1   apiVersion: k8s.nginx.org/v1
2   kind: Policy
3   metadata:
4     name: rate-limit-burst-policy
5     namespace: demo-app
6   spec:
7     rateLimit:
8       rate: 5r/s
9       burst: 10
10      key: ${http_user_id}
11      zoneSize: 10M
12
13  ---
14
15  apiVersion: k8s.nginx.org/v1
16  kind: Policy
17  metadata:
18    name: rate-limit-policy
19    namespace: demo-app
20  spec:
21    rateLimit:
22      rate: 1r/s
23      key: ${http_user_id}
24      zoneSize: 10M
```

00-app-virtualserver.yaml

```
1   upstreams:
2   - name: frontend
3     service: frontend
4     port: 8080
5   - name: tea
6     service: tea-svc
7     port: 80
8   - name: coffee
9     service: coffee-svc
10    port: 80
11    routes:
12  - path: /
13    policies:
14  - name: rate-limit-burst-policy
15    action:
16    pass: frontend
17  - path: /tea
18    action:
19      proxy:
20        upstream: tea
21  - path: /coffee
22    policies:
23  - name: rate-limit-policy
24    action:
25      proxy:
26        upstream: coffee
```


Module 2: Mutual TLS



Mutual SSL authentication / Certificate based mutual authentication

Ingress mTLS Configuration

type The type of certificate you are verifying against

verifyClient Where to enable client verification

verifyDepth The maximum number of intermediate certificate issuers

02-mtls-secrets.yaml

```
1  kind: Secret
2  metadata:
3    name: ingress-mtls-secret
4    namespace: demo-app
5  apiVersion: v1
6  type: nginx.org/ca
7  data:
8    ca.crt: LS0tLS1CRUdJTiBDRV….
```

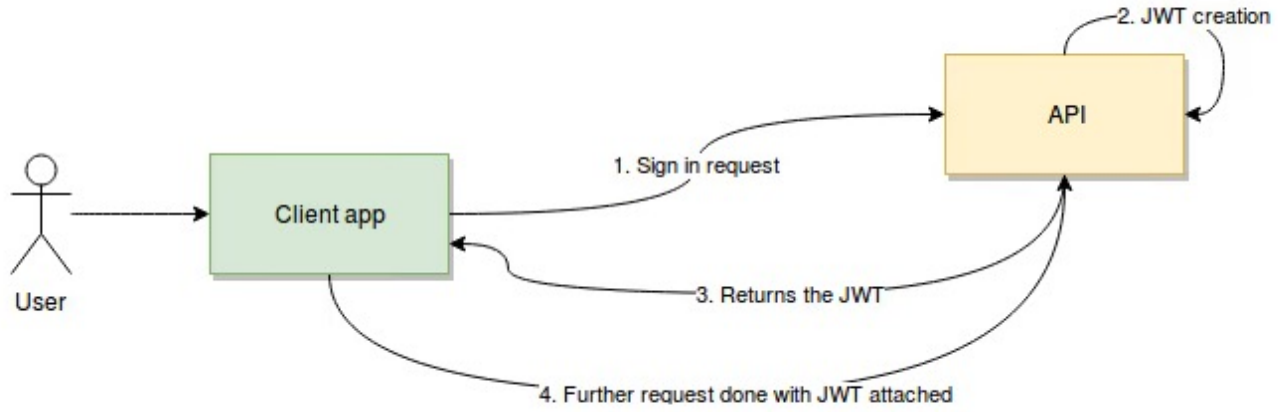
02-mtls-policy.yaml

```
1  apiVersion: k8s.nginx.org/v1
2  kind: Policy
3  metadata:
4    name: ingress-mtls-policy
5    namespace: demo-app
6  spec:
7    ingressMTLS:
8      clientCertSecret: ingress-mtls-secret
9      verifyClient: "on"
10     verifyDepth: 1
```

00-app-virtualserver.yaml

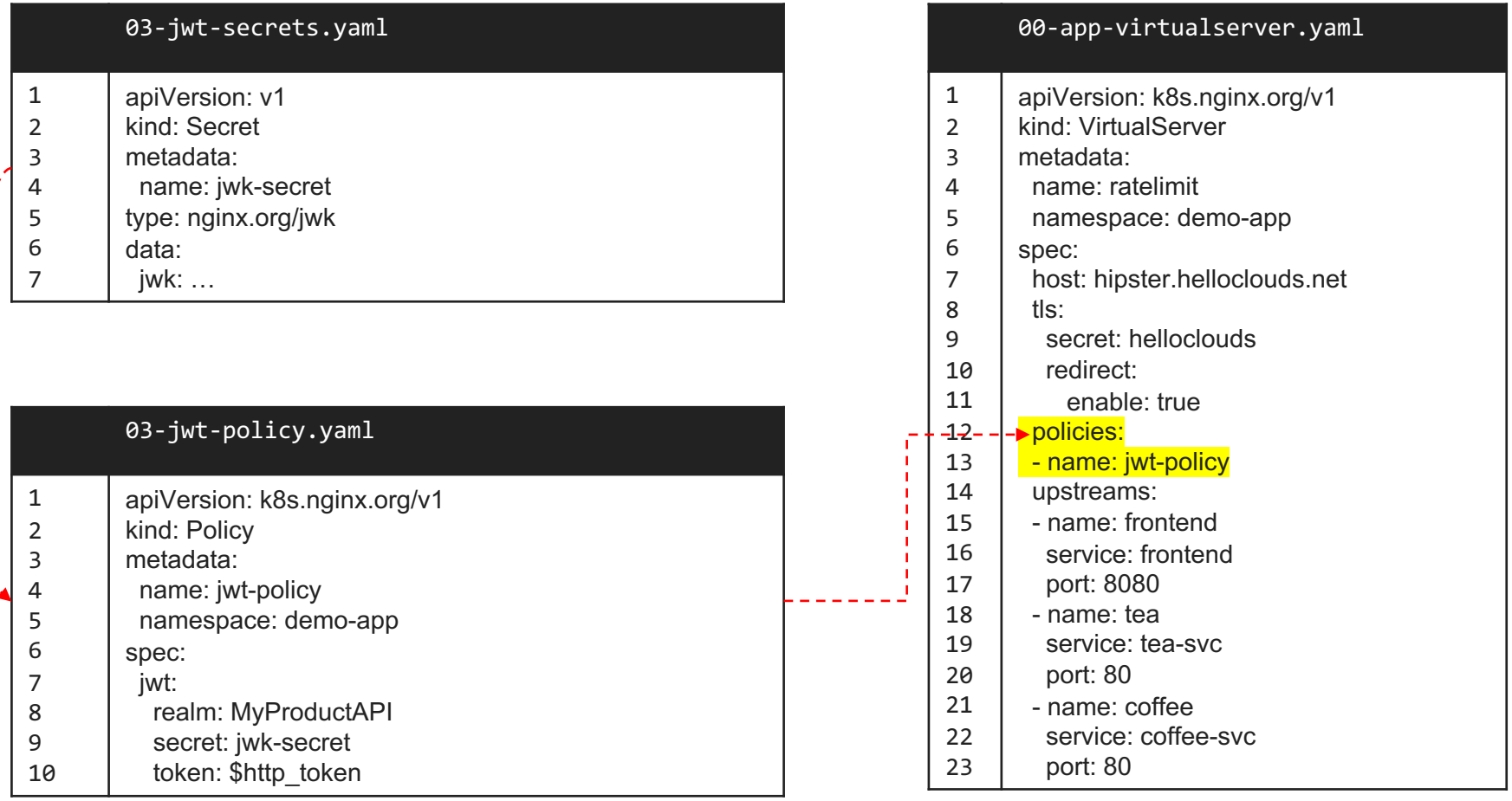
```
1  apiVersion: k8s.nginx.org/v1
2  kind: VirtualServer
3  metadata:
4    name: ratelimit
5    namespace: demo-app
6  spec:
7    host: hipster.helloclouds.net
8    tls:
9      secret: helloclouds
10     redirect:
11       enable: true
12   policies:
13   - name: ingress-mtls-policy
14   upstreams:
15   - name: frontend
16     service: frontend
17     port: 8080
18   - name: tea
19     service: tea-svc
20     port: 80
21   - name: coffee
22     service: coffee-svc
23     port: 80
```


Module 3: JWT Authentication



A JWT is a mechanism to verify the owner of some JSON data. It's an encoded, URL-safe string that can contain an unlimited amount of data (unlike a cookie) and is cryptographically signed.

JWT Authentication Configuration



03-jwt-secrets.yaml

```
1  apiVersion: v1
2  kind: Secret
3  metadata:
4    name: jwk-secret
5  type: nginx.org/jwk
6  data:
7    jwk: …
```

03-jwt-policy.yaml

```
1  apiVersion: k8s.nginx.org/v1
2  kind: Policy
3  metadata:
4    name: jwt-policy
5    namespace: demo-app
6  spec:
7    jwt:
8      realm: MyProductAPI
9      secret: jwk-secret
10     token: $http_token
```

00-app-virtualserver.yaml

```
1  apiVersion: k8s.nginx.org/v1
2  kind: VirtualServer
3  metadata:
4    name: ratelimit
5    namespace: demo-app
6  spec:
7    host: hipster.helloclouds.net
8    tls:
9      secret: helloclouds
10     redirect:
11       enable: true
12   policies:
13   - name: jwt-policy
14   upstreams:
15   - name: frontend
16     service: frontend
17     port: 8080
18   - name: tea
19     service: tea-svc
20     port: 80
21   - name: coffee
22     service: coffee-svc
23     port: 80
```


Module 4: API Protection


```
{
  "openapi": "3.0.1",
  "info": {
    "title": "httpbin",
    "description": "An unofficial OpenAPI definition for [httpbin.org](https://httpbin.org).",
    "version": "1.0-oas3"
  },
  "externalDocs": {
    "url": "http://httpbin.org/legacy"
  },
  "paths": {
    "/get": {…
    },
    "/headers": {…
    },
    "/delay/{n}": {
      "get": {
        "summary": "Delays responding for min(n, 10) seconds.",
        "parameters": [
          {
            "name": "n",
            "in": "path",
            "description": "Response delay, in seconds.",
            "required": true,
            "style": "simple",
            "explode": false,
            "schema": {
              "maximum": 10,
              "minimum": 0,
              "type": "integer"
            }
          }
        ],
        "responses": {
          "200": {
            "description": "OK",
            "content": {
              "application/json": {}
            }
          }
        }
      }
    },
  "components": {…
  }
}
```

Data Protection Configuration

Dataguard Provide masking for sensitive data

04-dataguard-protect.yaml

```
1  apiVersion: appprotect.f5.com/v1beta1
2  kind: APPolicy
3  metadata:
4    …
5  spec:
6    policy:
7      …
8      data-guard:
9        creditCardNumbers: true
10       enabled: true
11       enforcementMode: ignore-urls-in-list
12       …
13       maskData: true
14       usSocialSecurityNumbers: true
15     enforcementMode: blocking
16     name: dataguard-blocking
17     …
```

04-waf-policy.yaml

```
1  apiVersion: k8s.nginx.org/v1
2  kind: Policy
3  metadata:
4    name: hipsterstore-waf-policy
5    namespace: demo-app
6  spec:
7    waf:
8      enable: true
9      apPolicy: "dataguard-blocking"
```

00-app-virtualserver.yaml

```
1  apiVersion: k8s.nginx.org/v1
2  kind: VirtualServer
3  metadata:
4    name: ratelimit
5    namespace: demo-app
6  spec:
7    host: hipster.helloclouds.net
8    tls:
9      secret: helloclouds
10     redirect:
11       enable: true
12   policies:
13   - name: ingress-mtls-policy
14   - name: hipster-waf-policy
15   upstreams:
16   - name: frontend
17     service: frontend
18     port: 8080
19   - name: tea
20     service: tea-svc
21     port: 80
22   - name: coffee
23     service: coffee-svc
24     port: 80
```


API Protection Configuration

Swagger File

05-appolicy-api.yaml

```
1  apiVersion: appprotect.f5.com/v1beta1
2  kind: APPolicy
3  metadata:
4    name: appolicy-api
5    namespace: demo-app-mod5
6  spec:
7    policy:
8      name: appolicy-api
9      template:
10       name: POLICY_TEMPLATE_NGINX_BASE
11     applicationLanguage: utf-8
12     enforcementMode: blocking
13     open-api-files:
14     - link: https://raw.githubusercontent.com/mcheo-
15 nginx/sample-files/main/simplified-httpbin-oas.json
16     …
```

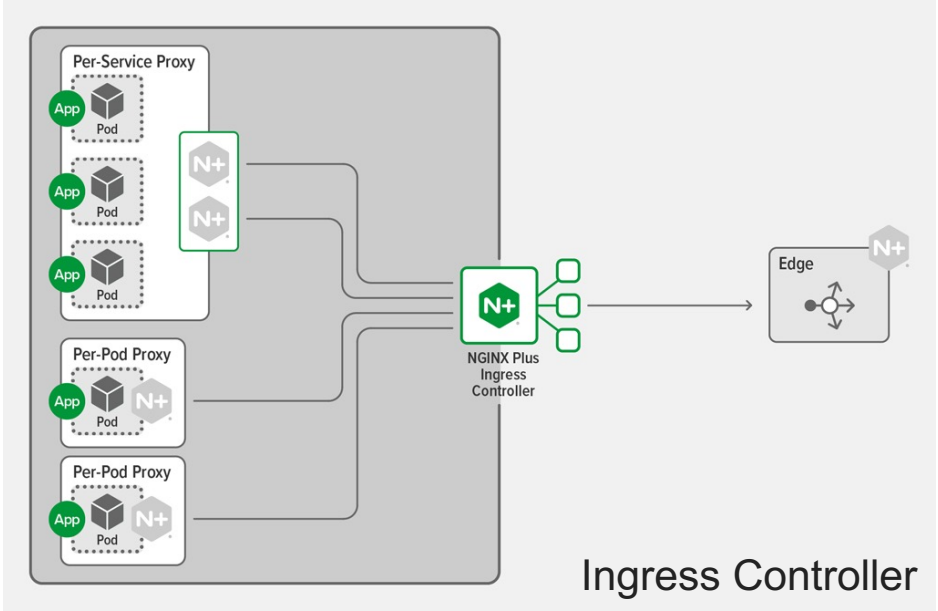
05-httpbin-waf-policy.yaml

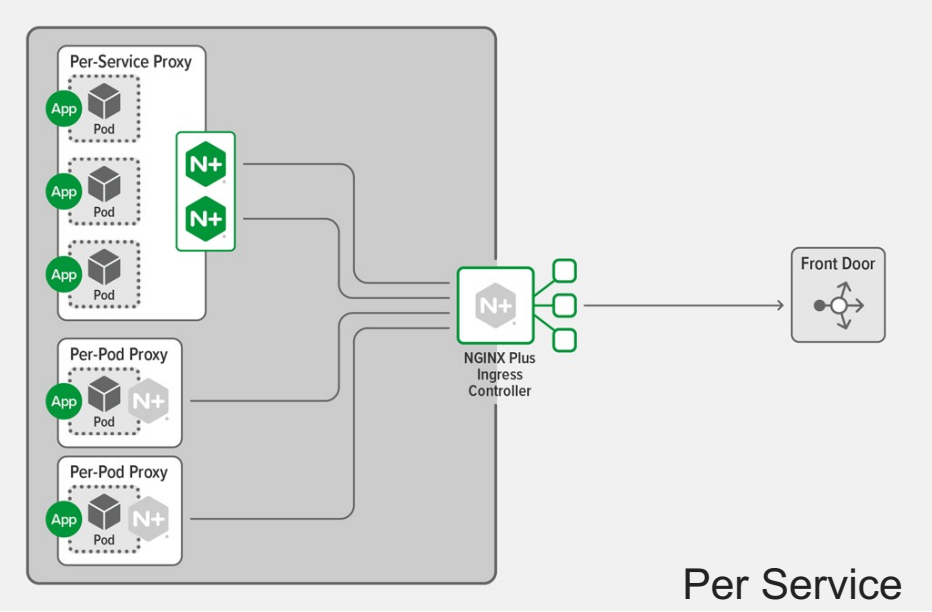
```
1  apiVersion: k8s.nginx.org/v1
2  kind: Policy
3  metadata:
4    name: httpbin-waf-policy
5    namespace: demo-app-mod5
6  spec:
7    waf:
8      enable: true
9      apPolicy: "appolicy-api"
```

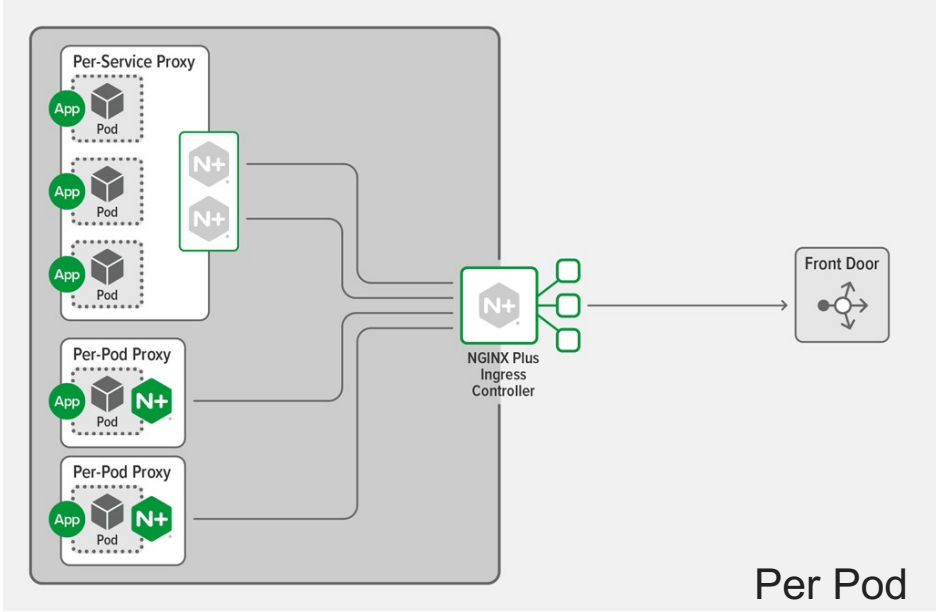
05-httpbin-vs.yaml

```
1  apiVersion: k8s.nginx.org/v1
2  kind: VirtualServer
3  metadata:
4    name: httpbin
5    namespace: demo-app-mod5
6  spec:
7    host: httpbin.helloclouds.net
8    tls: {}
9    policies:
10   - name: httpbin-waf-policy
11   upstreams:
12   - name: httpbin-upstream
13     service: httpbin-svc
14     port: 80
15   routes:
16   - path: /
17     action:
18       pass: httpbin-upstream
```


inside microservices cluster







What’s Next?