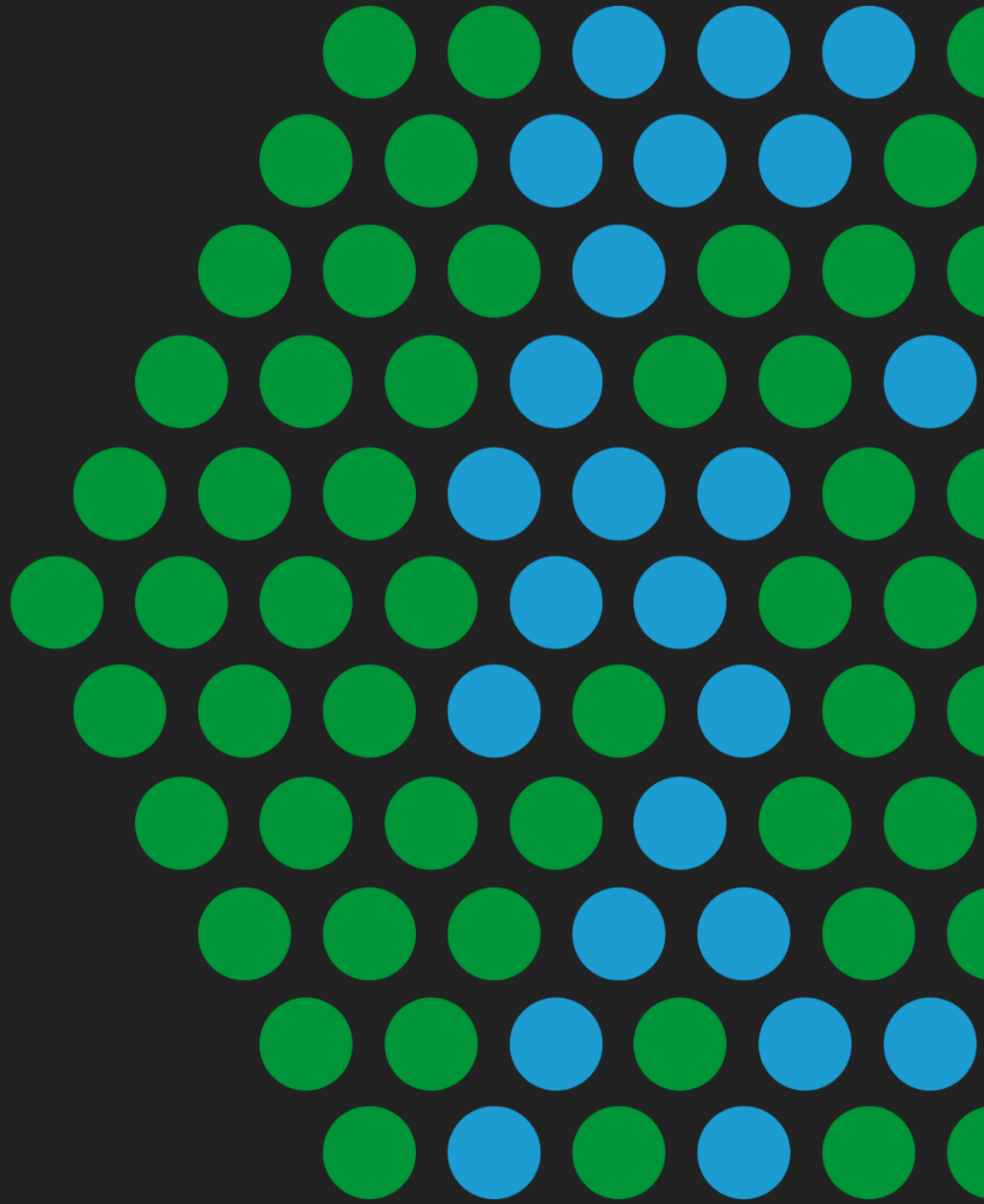


NGINX
Part of F5

Deploy and Secure Your API Gateway with NGINX

FROM ZERO TO HERO

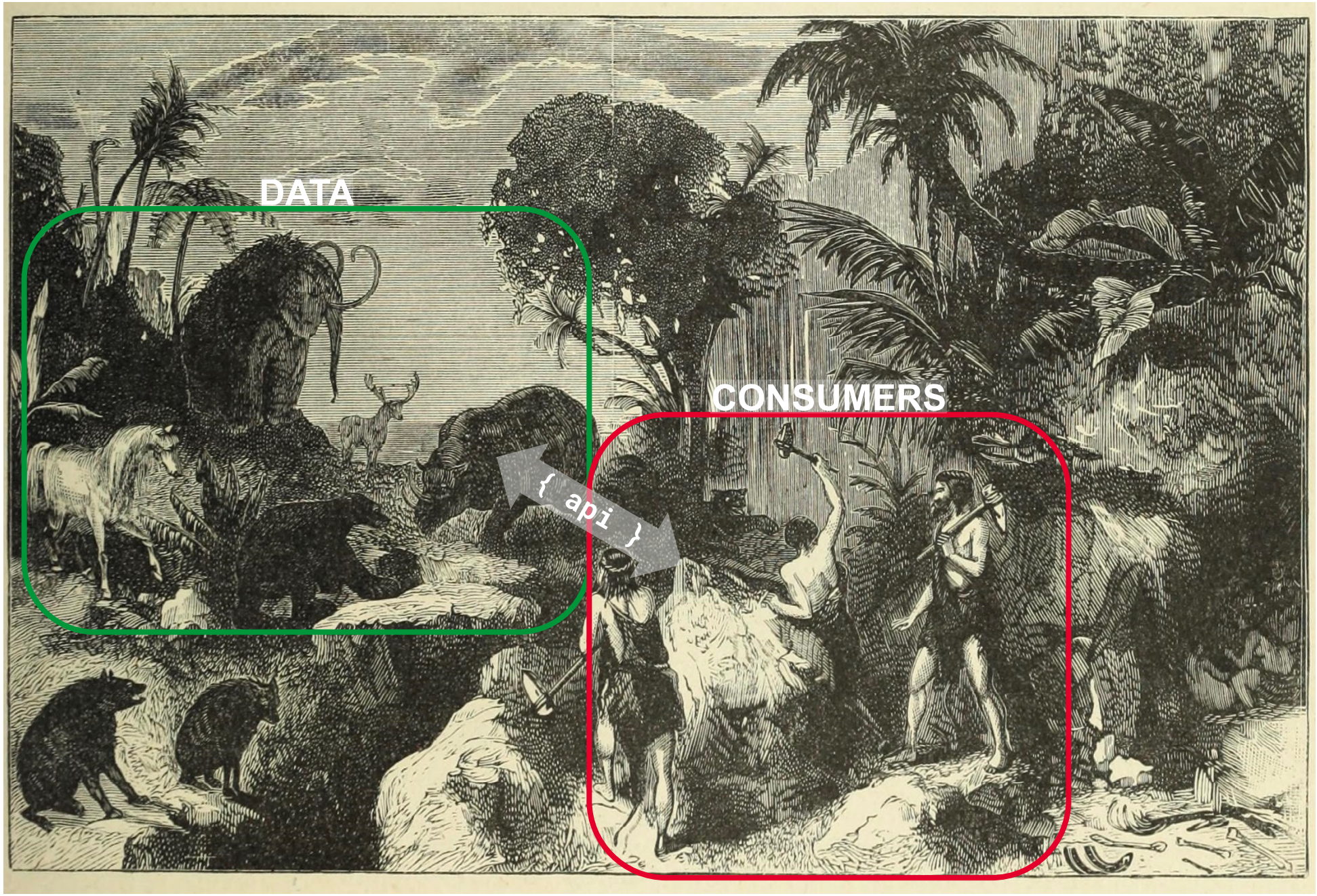
Alessandro Fael Garcia
Technical Marketing Engineer II

November 17, 2021

Agenda

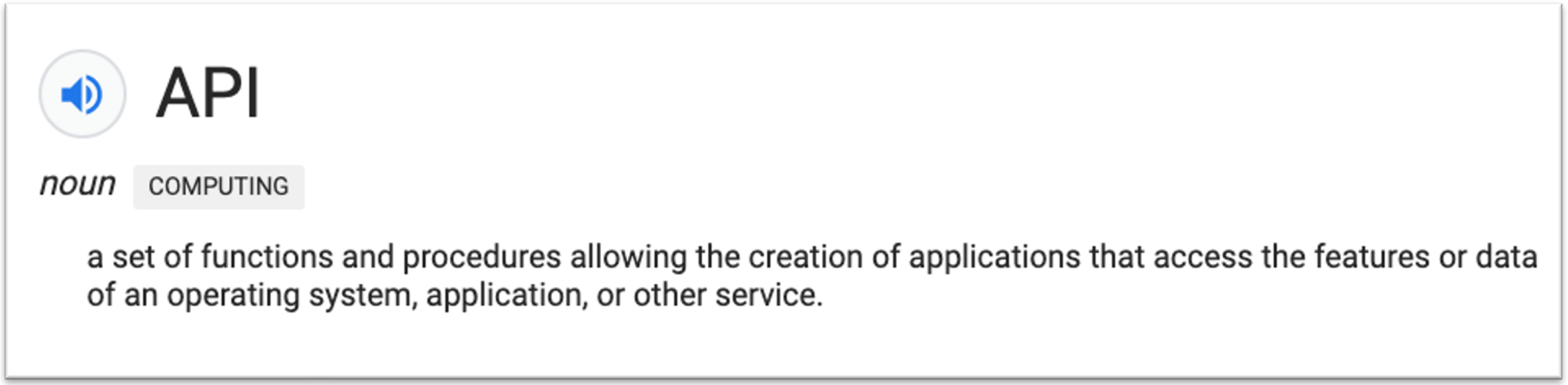
- What is a (REST) API?
- API fundamentals, or why are users & developers exponentially adopting an API first model?
- What are API gateways and how do they help deliver mission-critical digital experiences?
- Deploying NGINX as an API gateway
- Securing an API gateway using NGINX App Protect WAF
- Demo! (*Available at https://github.com/alessfg/nginx-api-gateway-demo*)

What is a (REST) API?


DATA
CONSUMERS
{ api }

API (endpoint) -> Application Programming Interface

EASY TO CONSUME DATA



Developers provide dedicated URLs that are optimized to return pure data to requesting clients

Web client → Web server → Request returns HTML

API client → API endpoint → Request returns data

Components of an API request

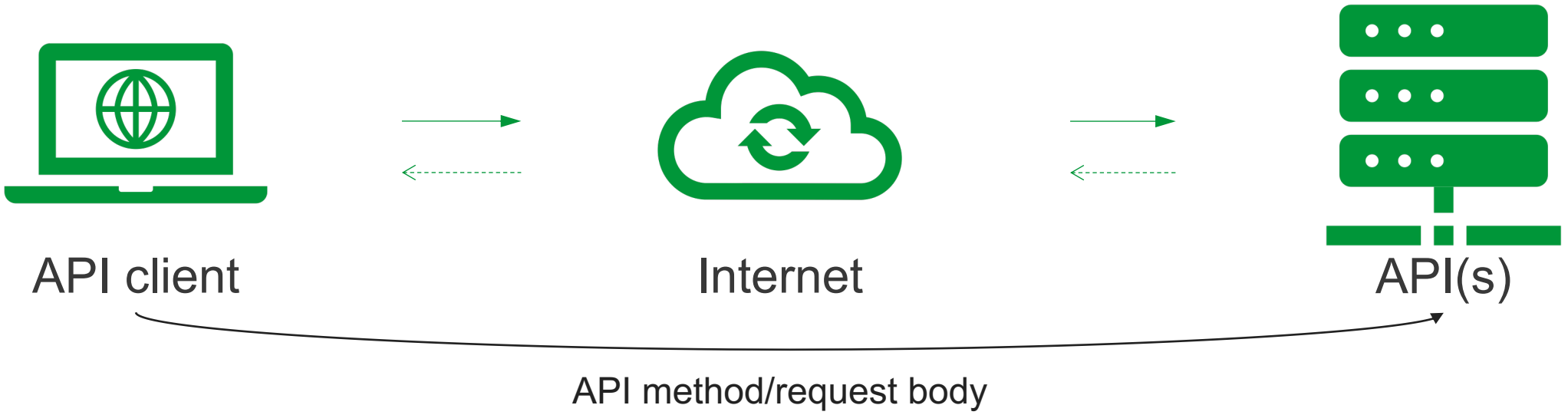
a) API endpoint -> URL

b) API method -> type of operation to conduct on the API endpoint

c) API request body -> if sending data via POST/PUT method

d) API client -> curl, postman, client application, etc…

Lifecycle of an API request

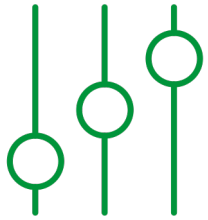


```
$ curl -s -X GET https://pokeapi.co/api/v2/pokemon/pikachu | jq
'.types'
[
  {
    "slot": 1,
    "type": {
      "name": ”electric",
      "url": "https://pokeapi.co/api/v2/type/13/"
    }
  }
]
```

API Fundamentals

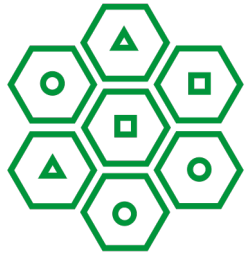
Why are users & developers using more APIs over time?

Drivers behind API adoption



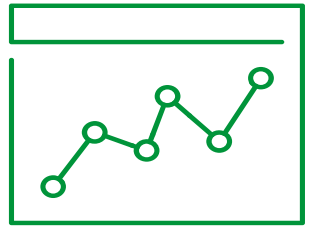
Ease access to information

- Break down siloes and unlock data (within and among organizations)
- Increase collaboration amongst developers



Connect microservices

- Primary interface for communication amongst microservices
- Simple development and maintenance
- Fault isolation and resiliency

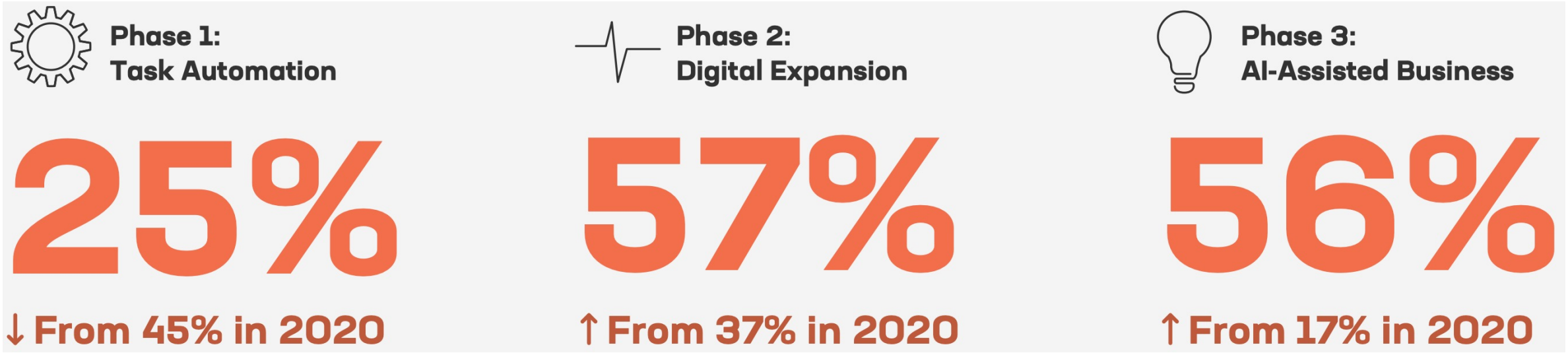


Create new digital revenue streams

- New opportunities to generate revenue
- Build partnerships with third-party developers and business ecosystem

Progress in digital transformation

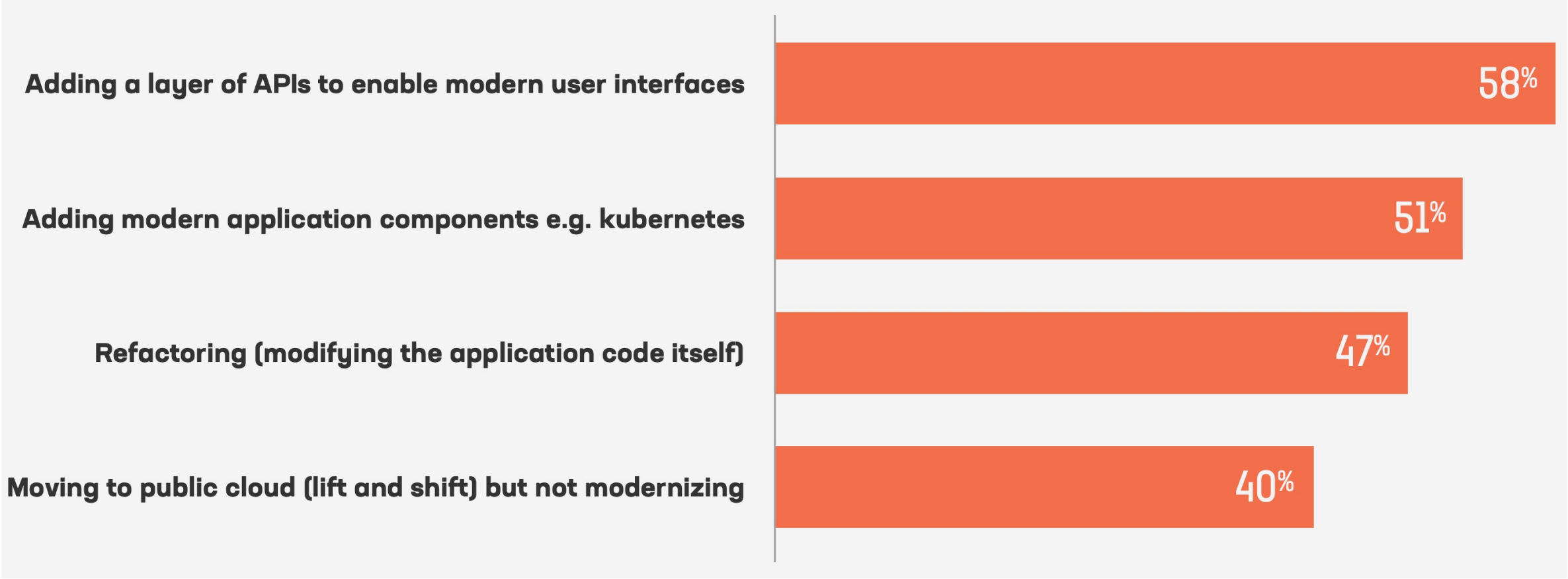
F5'S STATE OF APPLICATION STRATEGY 2021 REPORT



Source: https://www.f5.com/state-of-application-strategy-report

Modernization methods

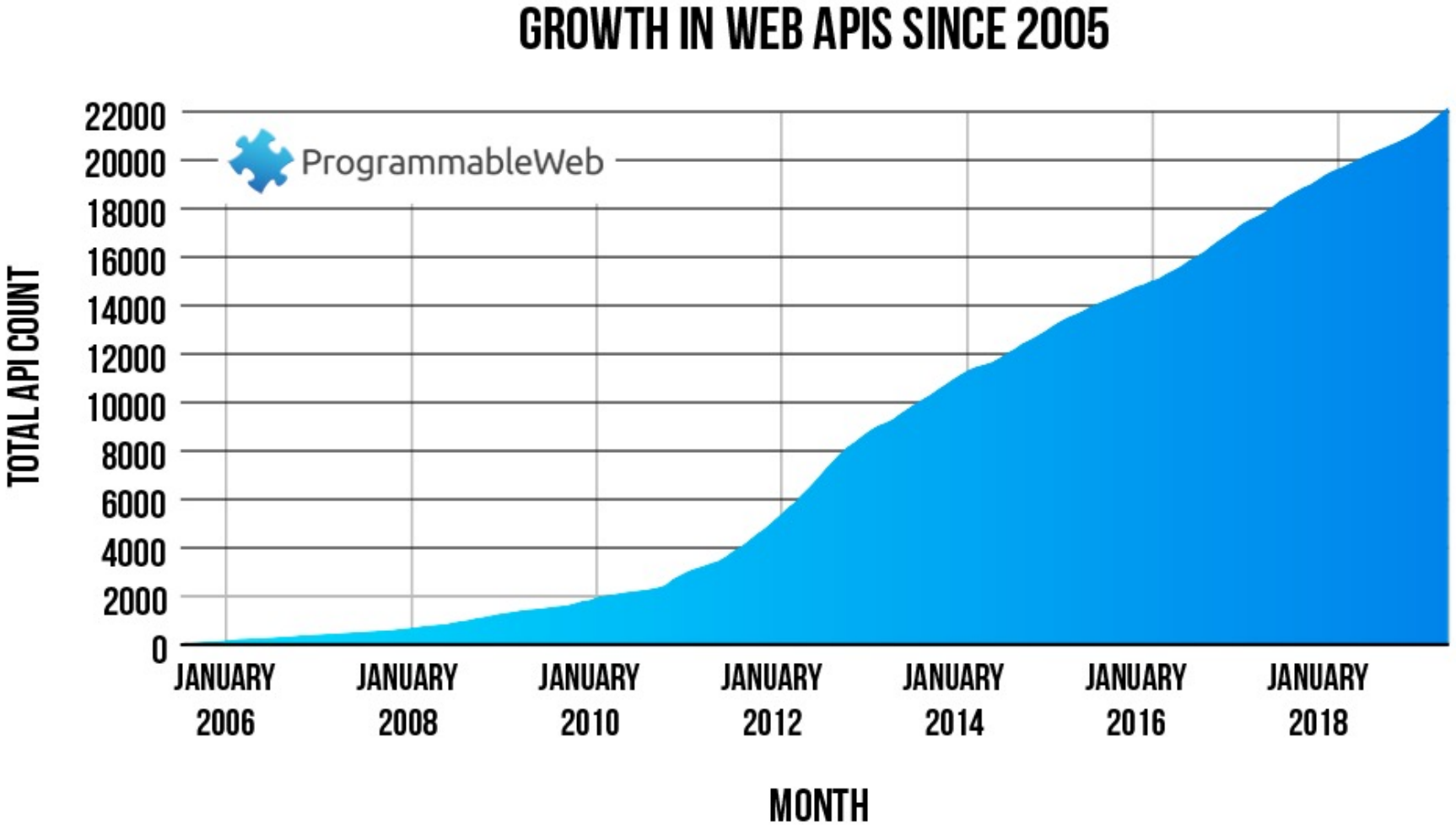
F5'S STATE OF APPLICATION STRATEGY 2021 REPORT



Source: https://www.f5.com/state-of-application-strategy-report

The rise of APIs

APIS ARE EXPERIENCING EXPLOSIVE GROWTH



Source: https://www.programmableweb.com/news/apis-show-faster-growth-rate-2019-previous-years/research/2019/07/17

API essentials

USERS & CONSUMERS

Ease of use

Low latency

Security

Documentation

API essentials

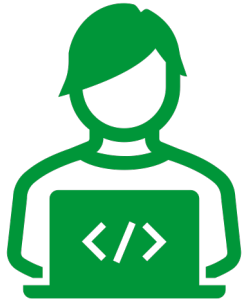
OWNERS & DEVELOPERS

Revenue growth

User experience

Brand protection

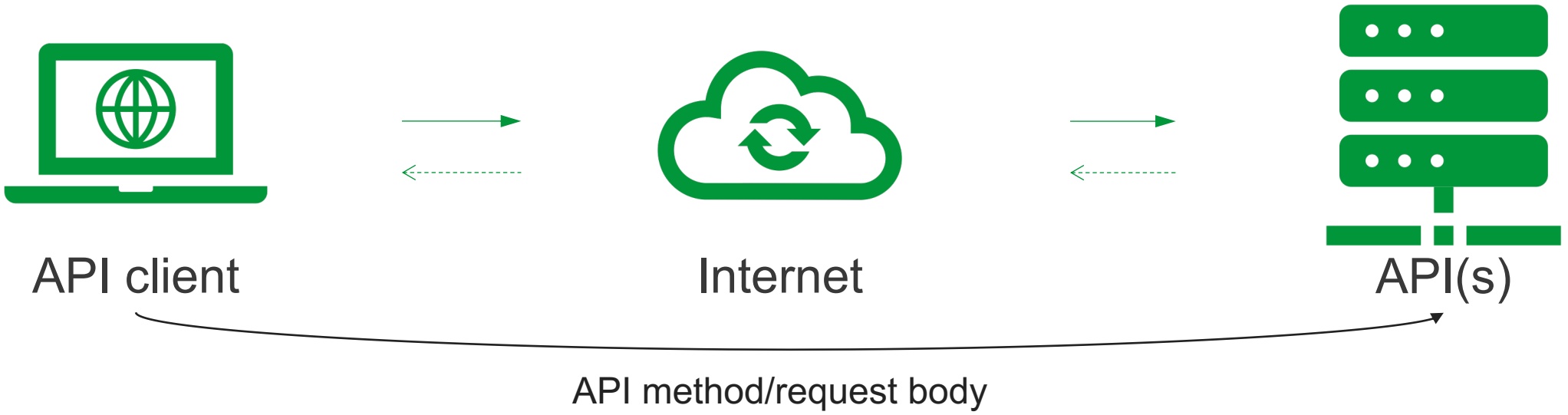
Developer productivity



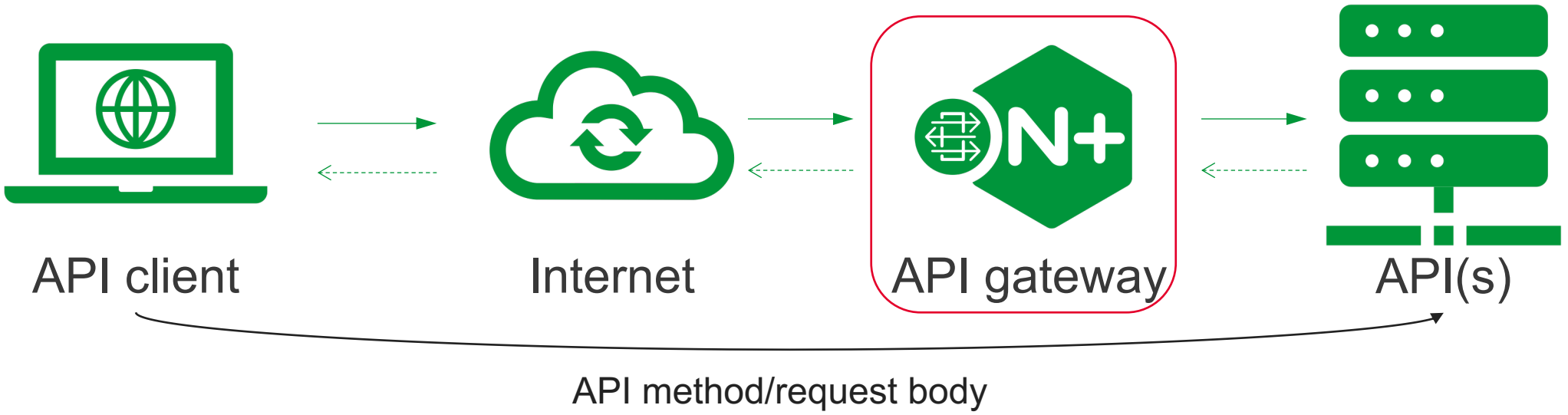
API Gateways

Deliver mission-critical digital experiences

Lifecycle of an API request



Lifecycle of an API request

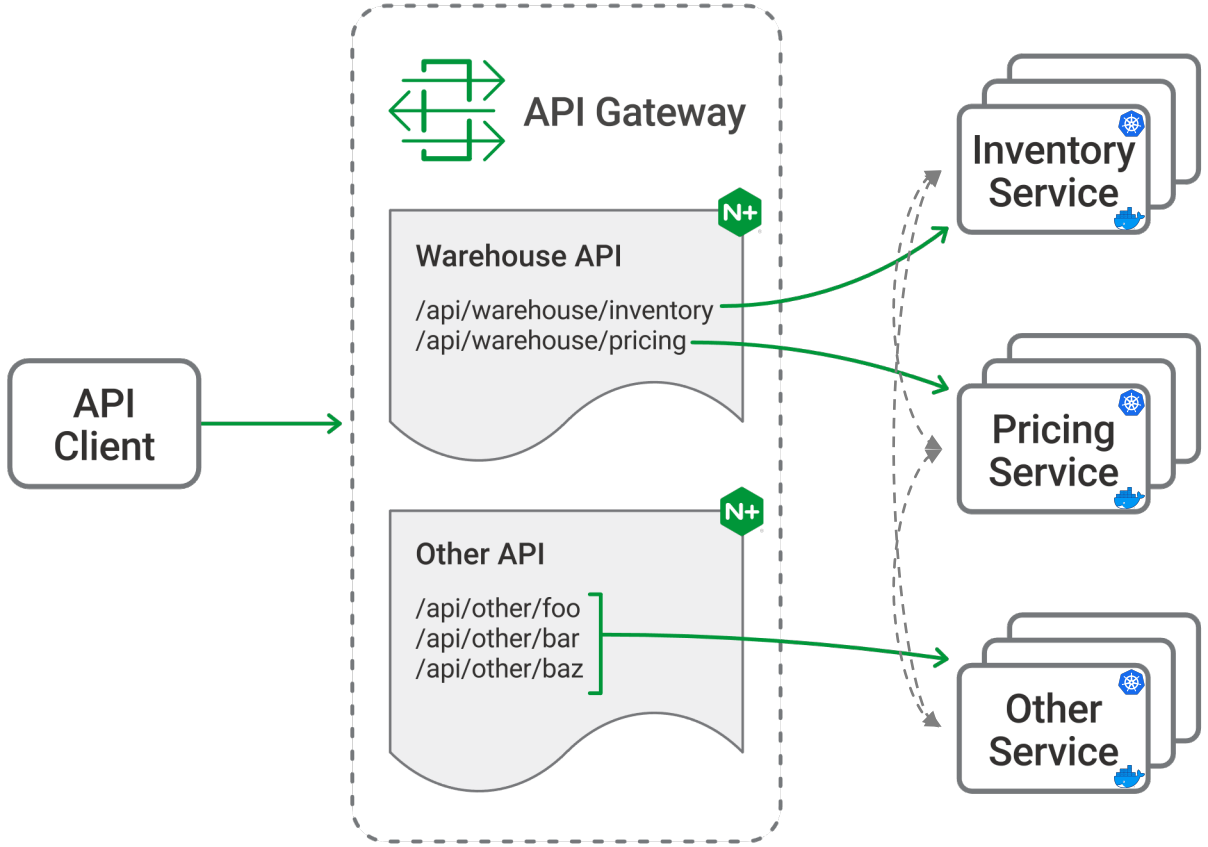



EXIT
EXCEPTION HANDLING
As a courtesy to other guests please wait until this at-bat is complete before returning to your seats.
Thank you
TRAFFIC CONTROL
REQUEST ROUTING
AUTHENTICATION

API gateway essentials

CONTROL ACCESS TO YOUR APIS

- Centralized logging
- Client authentication
- Fine grained access control
- Load balancing
- Rate limiting
- Request routing
- Request/response manipulation
- Service discovery of backends
- TLS termination



API gateway essentials

REAL-TIME PERFORMANCE IS KEY IN DIGITAL EXPERIENCES

30 ms

@p99

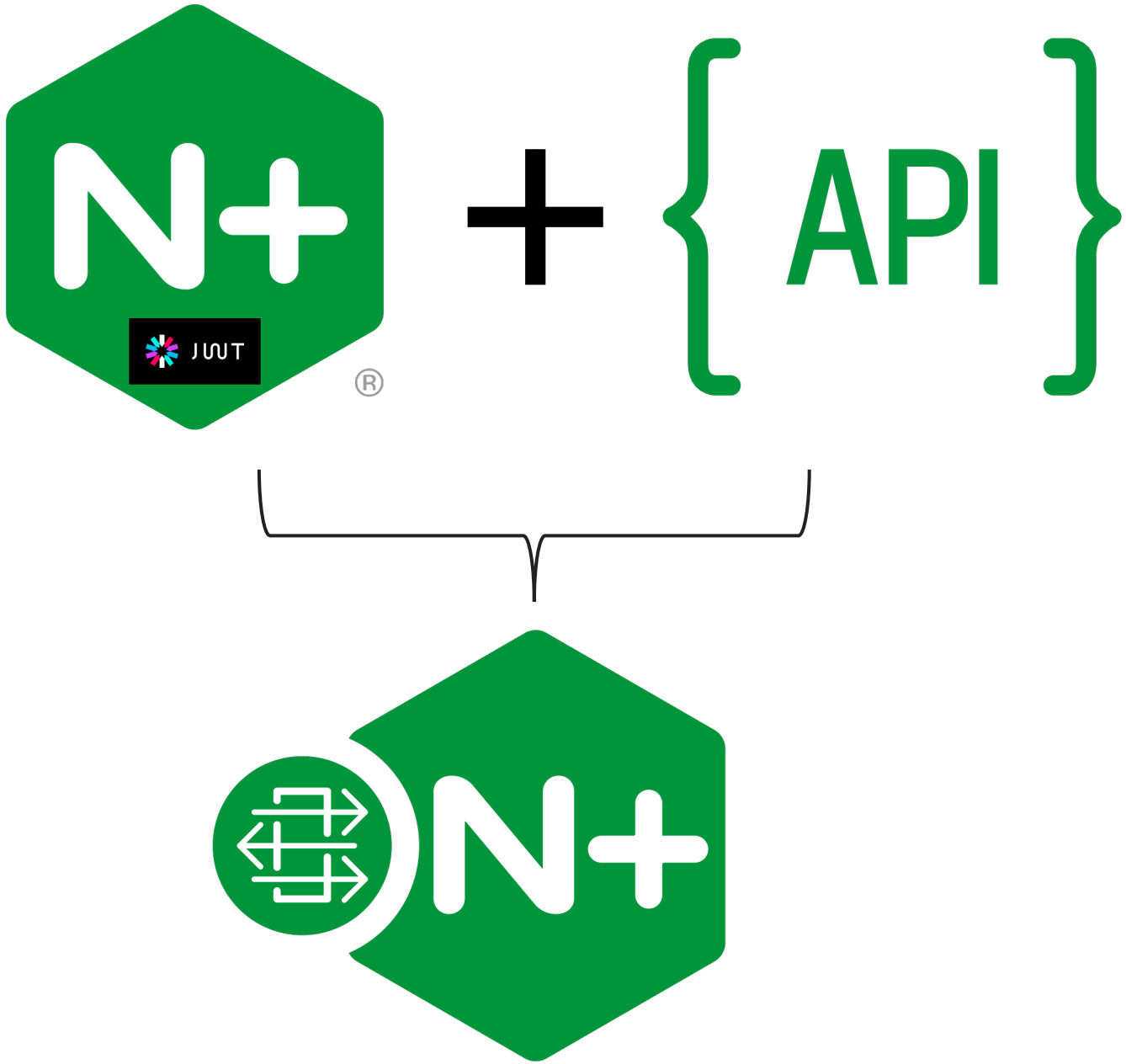
(latency)

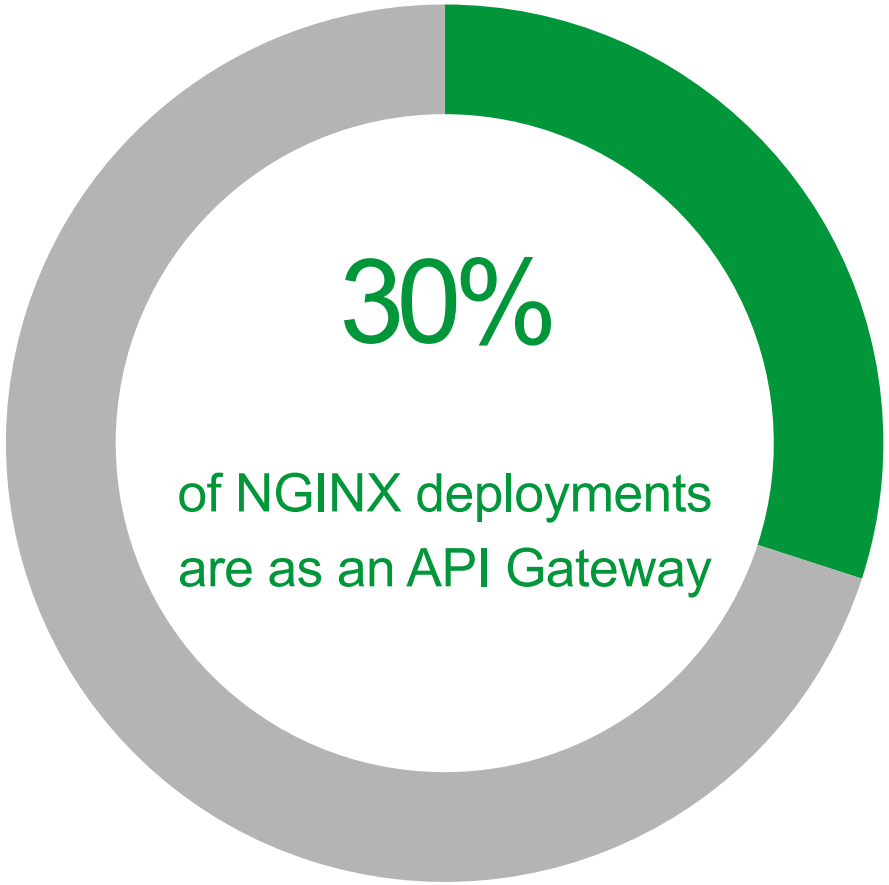
... to process an API request end-to-end

... to route, shape, authenticate, secure, and/or cache an API

NGINX Plus

The ultimate API gateway

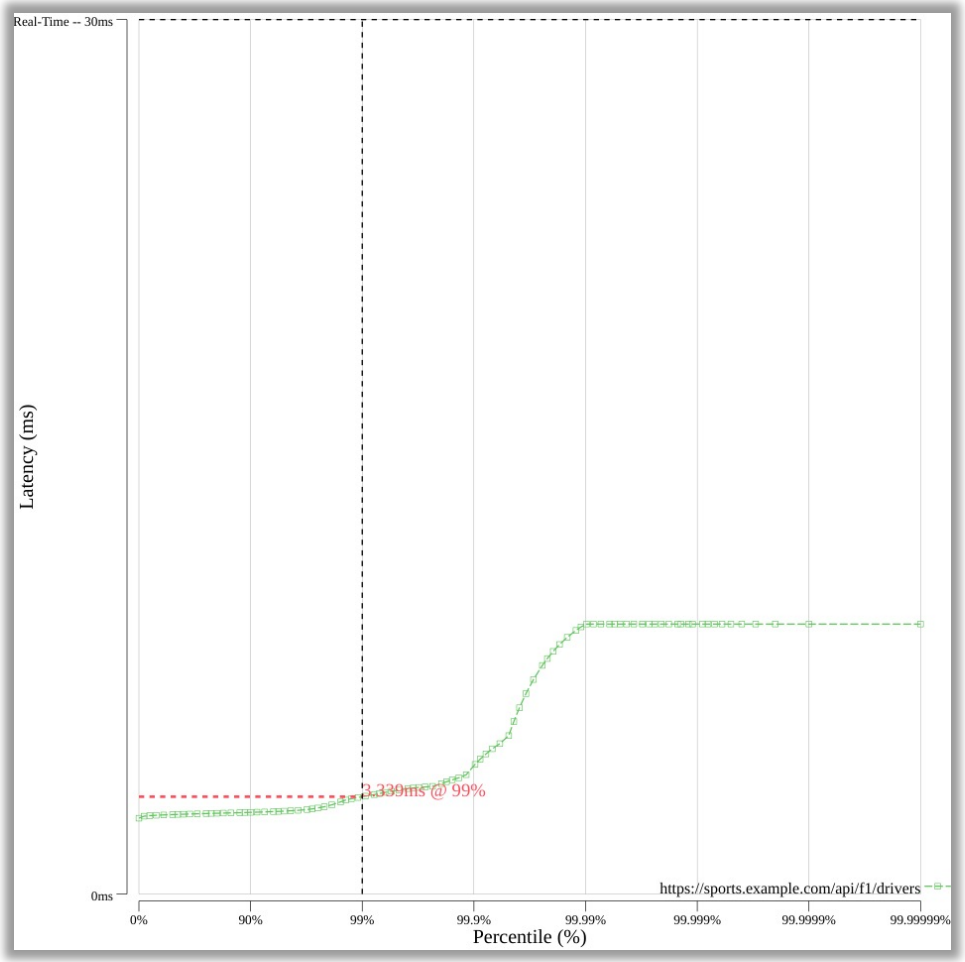

JWT
API



Source: https://www.nginx.com/blog/state-of-modern-app-delivery-2020-results-annual-user-survey/

What makes NGINX API gateways special?

ACCESS YOUR APIS IN LESS THAN 30MS EVEN WHEN USING AN API GATEWAY

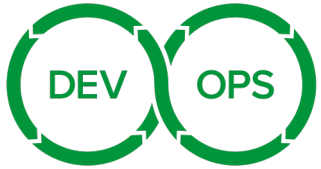



HUBLOT
Santander
SINGHA
SCUDERIA FERRARI
7
V-Power
Santander
MAHLE
Ray-Ban
ups
infor
Experis
SKF
NGK
brembo

NGINX API gateway

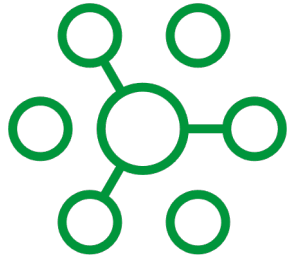
KEY STRENGTHS

High performance



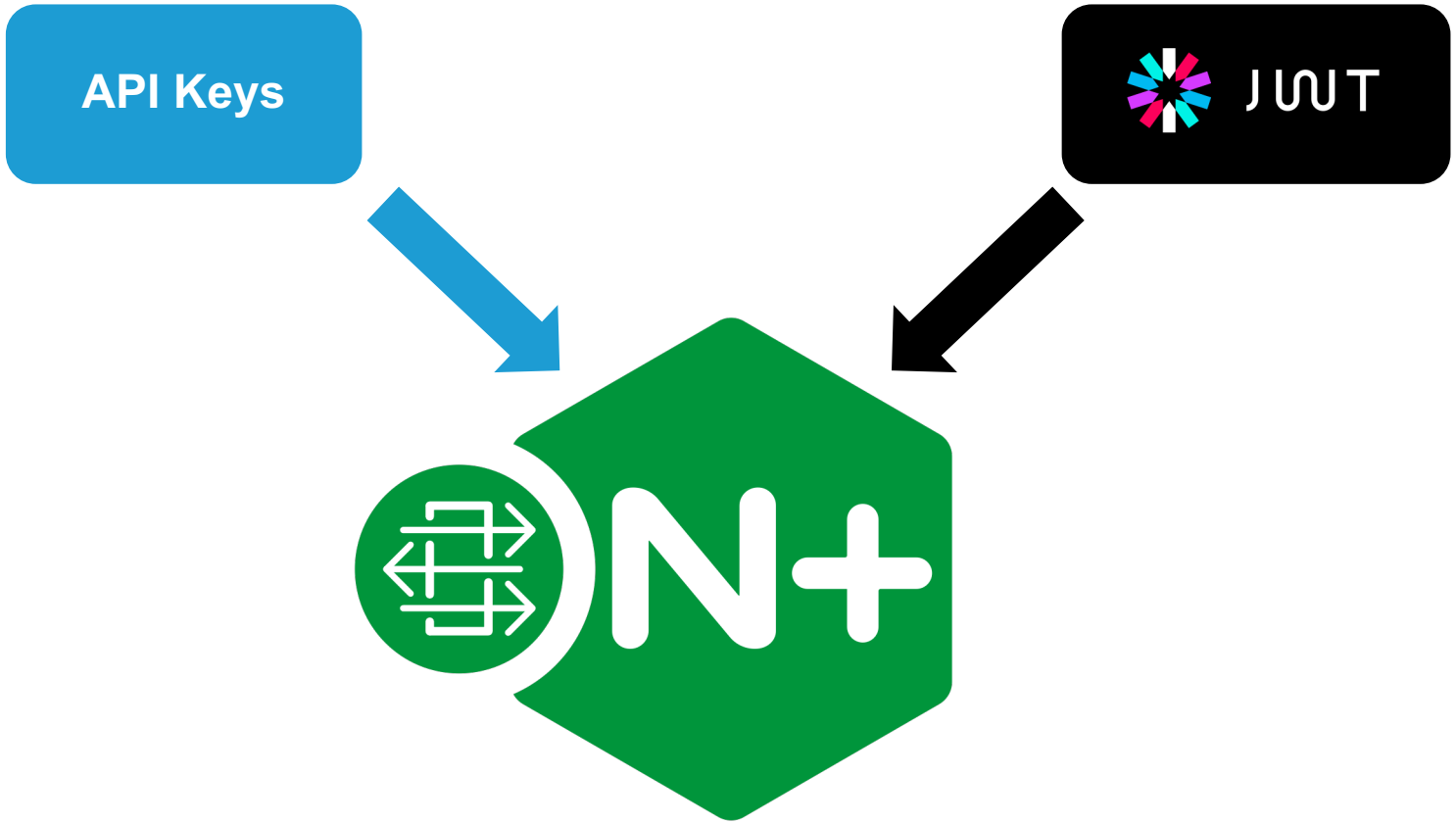
DevOps friendly

Platform flexibility



Distributed environments

Authentication options

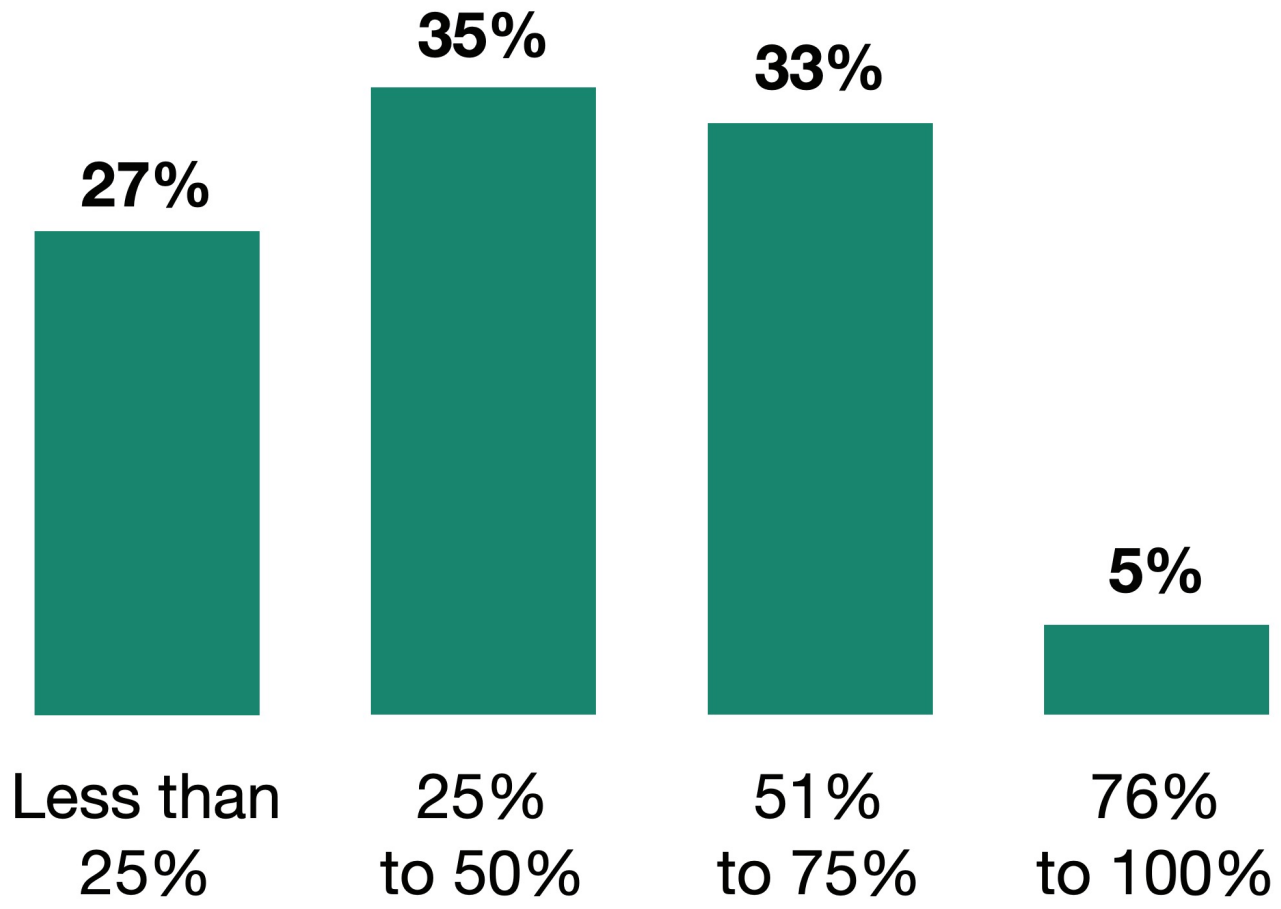


NGINX App Protect

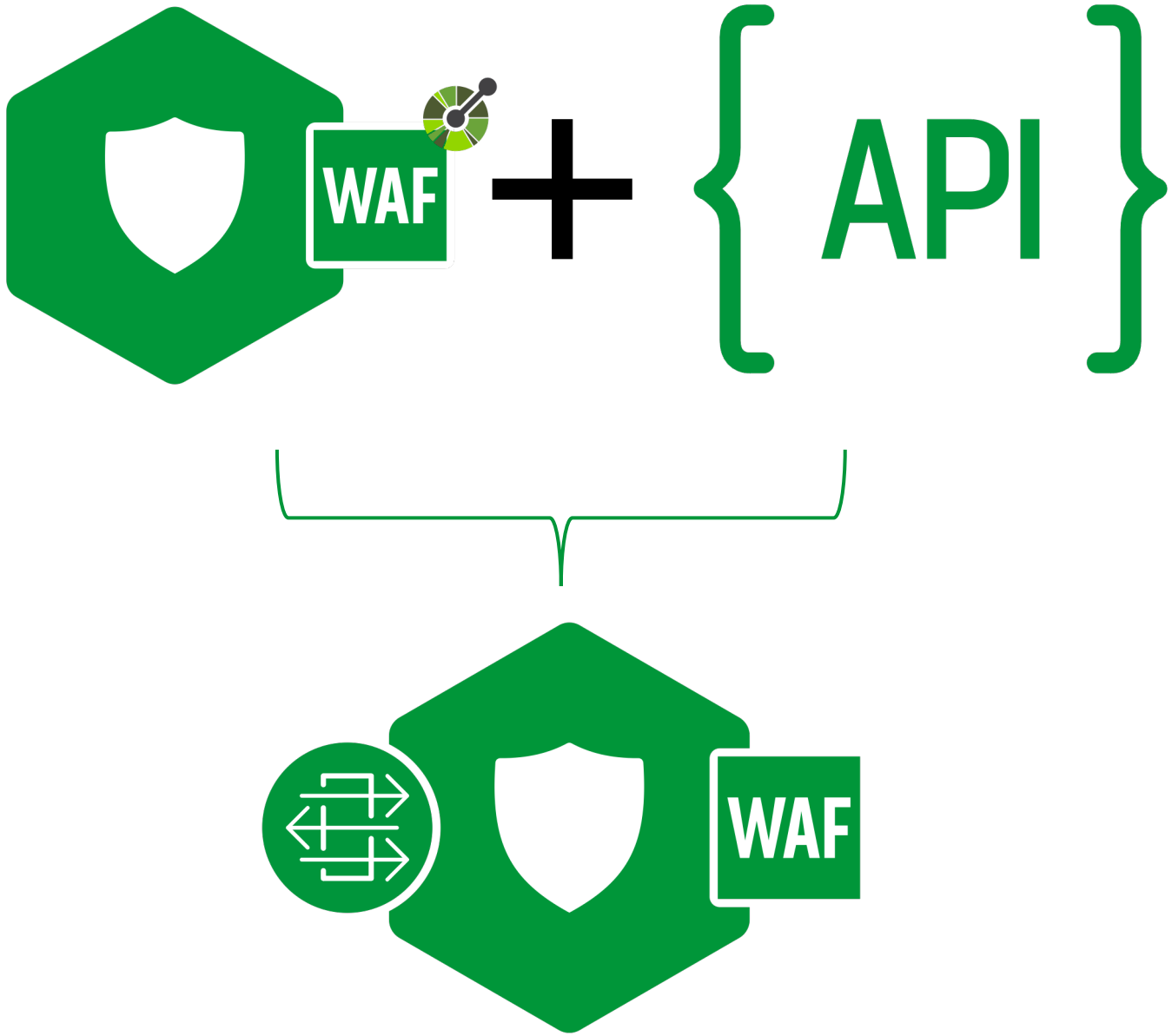
Native API security

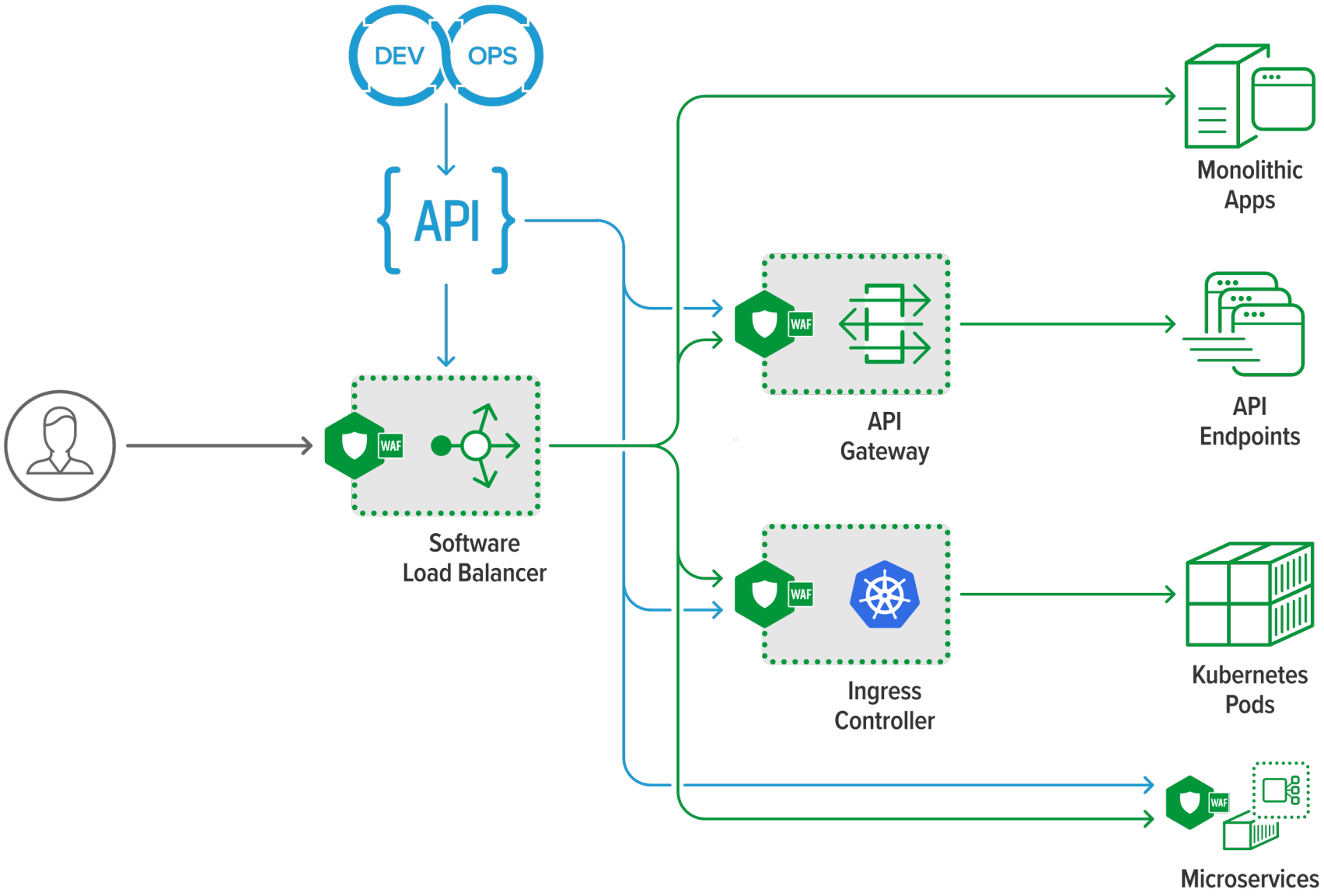
Percentage of apps organizations expose via APIs

FORRESTER'S STATE OF APPLICATION SECURITY 2021 REPORT



Source: https://www.forrester.com/report/the-state-of-application-security-2021/RES164041


WAF
API
WAF


DEV
OPS
API
WAF
Software
Load Balancer
WAF
API
Gateway
WAF
Ingress
Controller
WAF
Monolithic
Apps
API
Endpoints
Kubernetes
Pods
WAF
Microservices

Demo time!

Demo available at https://github.com/alessfg/nginx-api-gateway-demo

Define the API gateway

/etc/nginx/conf.d/**api_gateway.conf**

```
1  server {
2      listen 8080;
3
4      # TLS config goes here (for production)
5
6      include conf.d/my_apis/*.conf;
7
8      # Invalid resource
9      location / {
10         return 400;
11     }
12
13     default_type application/json;
14 }
```

```
$ curl -s http://localhost:8080
<html>
<head><title>400 Bad Request</title></head>
<body>
<center><h1>400 Bad Request</h1></center>
<hr><center>nginx/1.19.5</center>
</body>
</html>
```

Define the JSON error responses

/etc/nginx/conf.d/my_apis/**api_json_errors.conf**

```
error_page 400 = @400;
location @400 {
    return 400 '{"status":400,"message":"Bad request"}\n';
}
error_page 401 = @401;
location @401 {
    return 401 '{"status":401,"message":"Unauthorized"}\n';
}
error_page 403 = @403;
location @403 {
    return 403 '{"status":403,"message":"Forbidden"}\n';
}
error_page 405 = @405;
location @405 {
    return 405 '{"status":405,"message":"Method not allowed"}\n';
}
error_page 429 = @429;
location @429 {
    return 429 '{"status":429,"message":"API rate limit exceeded"}\n';
}
```

```
$ curl -s http://localhost:8080
{"status":400,"message":"Bad request"}
```

Define the API endpoints

/etc/nginx/conf.d/my_apis/**f1.conf**

```
location /api/f1/ {
    location = /api/f1/seasons {
        proxy_pass http://f1-admin;
    }

    location ~ /api/f1/[12][0-9]+ {
        proxy_pass http://f1-data;
    }

    location /api/f1/drivers {
        proxy_pass http://f1-data;
    }
}
```

Define the backend/upstream servers

/etc/nginx/conf.d/**api_backends.conf**

```
1  upstream f1-admin {
2      server host.docker.internal;
3  }
4
5  upstream f1-data {
6      server host.docker.internal:8000;
7  }
```

```
$ curl -s http://localhost:8080/api/f1/drivers/hamilton | jq
```


```
{
  "MRData": {
    "xmlns": "http://ergast.com/mrd/1.4",
    "series": "f1",
    "url": "http://ergast.com/api/f1/drivers/hamilton",
    "limit": "30",
    "offset": "0",
    "total": "1",
    "DriverTable": {
      "driverId": "hamilton",
      "Drivers": [
        {
          "driverId": "hamilton",
          "permanentNumber": "44",
          "code": "HAM",
          "url": "http://en.wikipedia.org/wiki/Lewis_Hamilton",
          "givenName": "Lewis",
          "familyName": "Hamilton",
          "dateOfBirth": "1985-01-07",
          "nationality": "British"
        }
      ]
    }
  }
}
```

Rate limiting

/etc/nginx/conf.d/**api_gateway.conf**

```
1   limit_req_zone $remote_addr zone=perip:1m
    rate=2r/s;
2
3   server {
4       listen 8080;
…   ...
```

/etc/nginx/conf.d/my_apis/**f1.conf**

```
 1  location /api/f1/ {
 2      limit_req zone=perip nodelay;
 3      limit_req_status 429;
 4
 5      location = /api/f1/seasons {
 6          proxy_pass http://f1-admin;
 7      }
 8      location ~ /api/f1/[12][0-9]+ {
 9          proxy_pass http://f1-data;
10      }
11      location /api/f1/drivers {
12          proxy_pass http://f1-data;
13      }
14  }
```

```
$ curl -s http://localhost:8080/api/f1/drivers/hamilton
{"MRData":{"xmlns":"http:\/\/ergast.com\/mrd\/1.4","series":"f1","u
rl":http://ergast.com/api/f1/drivers/hamilton...

$ curl -s http://localhost:8080/api/f1/drivers/hamilton
{"MRData":{"xmlns":"http:\/\/ergast.com\/mrd\/1.4","series":"f1","u
rl":http://ergast.com/api/f1/drivers/hamilton...

$ curl -s http://localhost:8080/api/f1/drivers/hamilton
{"MRData":{"xmlns":"http:\/\/ergast.com\/mrd\/1.4","series":"f1","u
rl":http://ergast.com/api/f1/drivers/hamilton...

$ curl -s http://localhost:8080/api/f1/drivers/hamilton
{"status":429,"message":"API rate limit exceeded"}
```

API key authentication

/etc/nginx/conf.d/my_apis/**f1.conf**

```
location /api/f1/ {
    auth_request /_validate_apikey;
…
}

location = /_validate_apikey {
    internal;

    if ($http_apikey = "") {
        return 401; # Unauthorized
    }
    if ($api_client_name = "") {
        return 403; # Forbidden
    }

    return 204; # OK (no content)
}
```

/etc/nginx/conf.d/**api_key.conf**

```
map $http_apikey $api_client_name {
    "7B5zIqmRGXmrJTFmKa99vcit" "client_one";
    "QzVV6y1EmQFbbxOfRCwyJs35" "client_two";
    default "";
}
```

```
$ curl -s http://localhost:8080/api/f1/drivers/hamilton
{"status":401,"message":"Unauthorized"}


$ curl -sH "apikey: thisIsInvalid"
http://localhost:8080/api/f1/drivers/hamilton
{"status":403,"message":"Forbidden"}


$ curl -sH "apikey: 7B5zIqmRGXmrJTFmKa99vcit"
http://localhost:8080/api/f1/drivers/hamilton
{"MRData":{"xmlns":"http://ergast.com/mrd/1.4","series":"f1","url":
"http://ergast.com/api/f1/drivers/hamilton"...
```

JWT authentication

REQUIRES NGINX PLUS

/etc/nginx/conf.d/my_apis/f1.conf

```
location /api/f1/ {
    auth_jwt “F1 API”;
    auth_jwt_key_file /etc/nginx/api_secret.jwk
...
    location /api/f1/drivers {
        if ($admin_permitted_method != "true") {
            return 403; # Method not allowed
        }
        error_page 401 = @401
        error_page 403 = @405

        proxy_pass http://f1-data;
    }
}
```

/etc/nginx/conf.d/api_jwt.conf

```
map $request_method $admin_permitted_method {
    "GET"     true;
    "HEAD"    true;
    "OPTIONS" true;
    default   $jwt_claim_admin;
}
```

/etc/nginx/conf.d/api_secret.jwk

```
{
  "keys": [
    {
      "k": "YWdpbGl0eQ",
      "kty": "oct"
    }
  ]
}
```

```
$ curl -s http://localhost:8080/api/f1/drivers/hamilton

{"status":401,"message":"Unauthorized"}

$ curl -sH "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbiI6dHJ1ZX0.kFplw9Kkg-
6DLFGfVZAPIuWgGPMY9nnMZMQ2iIRN8_s"
http://localhost:8080/api/f1/drivers/hamilton

{"MRData":{"xmlns":"http:\/\/ergast.com\/mrd\/1.4","series":"f1","u
rl":http://ergast.com/api/f1/drivers/hamilton...

$ curl -sH "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbiI6ZmFsc2V9.i7o5c8MEG
ZWD223IWFIs-Qn6f8FBe_DjvZWn-xBzcvI" -X DELETE
http://localhost:8080/api/f1/drivers/hamilton

{"status":405,"message":"Method not allowed"}
```

JSON request validation (1/2)

REQUIRES NGINX JAVASCRIPT MODULE (NJS) -> HTTPS://NGINX.ORG/EN/DOCS/NJS/

/etc/nginx/nginx.conf

```
load_module modules/ngx_http_js_module.so;

user  nginx;
worker_processes  auto;

error_log  /var/log/nginx/error.log notice;
pid        /var/run/nginx.pid;

events {
    worker_connections  1024;
}
...
```

/etc/nginx/conf.d/json_validation.js

```
export default { parseRequestBody };

function parseRequestBody(r) {
    try {
        if (r.variables.request_body) {
            JSON.parse(r.variables.request_body);
        }
        return r.variables.upstream;
    } catch (e) {
        r.error('JSON.parse exception');
        return '127.0.0.1:10415'; // Address for error response
    }
}
```

JSON request validation (2/2)

REQUIRES NGINX JAVASCRIPT MODULE (NJS) -> HTTPS://NGINX.ORG/EN/DOCS/NJS/

/etc/nginx/conf.d/api_gateway.conf

```
...  ...
11       # Request body validation
12       location /_get_request_body {
13           internal;
14           return 204;
15       }
...  ...
```

/etc/nginx/conf.d/my_apis/f1.conf

```
...  ...
 8       location = /api/f1/seasons {
 9           set $upstream f1-admin;
10           mirror /_get_request_body; # Force early-reading of request body
11           proxy_pass http://$json_validated$request_uri;
12       }
...  ...
```

/etc/nginx/conf.d/json_validation.conf

```
1  js_import conf.d/json_validation.js;
2  js_set $json_validated json_validation.parseRequestBody;
3
4  server {
5      listen 127.0.0.1:10415; # Error response of json_validator()
6      return 415; # Unsupported media type
7      include conf.d/my_apis/api_json_erors.conf;
8  }
```

```
$ curl -sH "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbiI6dHJ1ZX0.kFplw9Kkg-
6DLFGfVZAPIuWgGPMY9nnMZMQ2iIRN8_s" -i -X POST -d 'garbage123'
http://localhost:8080/api/f1/seasons

HTTP/1.1 415 Unsupported Media Type
```

```
$ curl -sH "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbiI6dHJ1ZX0.kFplw9Kkg-
6DLFGfVZAPIuWgGPMY9nnMZMQ2iIRN8_s" -i -X POST -d
'{"season":"2020"}' http://localhost:8080/api/f1/seasons

HTTP/1.1 200 OK
```

NGINX App Protect OpenAPI security (1/2)

REQUIRES NGINX APP PROTECT WAF

/etc/nginx/**nginx.conf**

```
      1  load_module modules/ngx_http_app_protect_module.so;
    ...  ...
     14  http {
     15      app_protect_enable on;
     16      app_protect_policy_file "/etc/nginx/nginx_api_security_policy.yml";
     17      app_protect_security_log_enable on;
     18      app_protect_security_log "/etc/app_protect/conf/log_default.json" /var/log/app_protect/security.log;
    ...  ...
```

NGINX App Protect OpenAPI security (2/2)

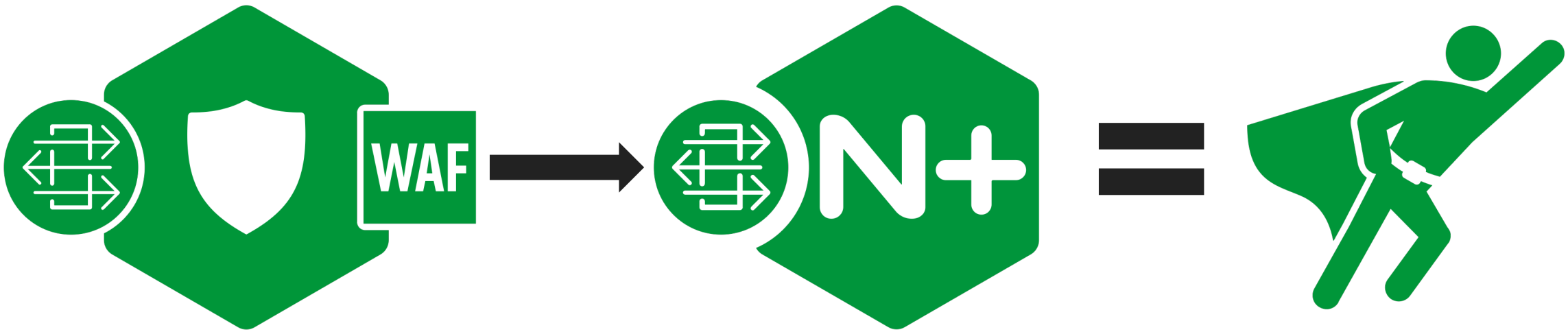
REQUIRES NGINX APP PROTECT WAF

/etc/nginx/nginx_api_security_policy.json

```
{
  "policy": {
    "name": "app_protect_api_security_policy",
    "description": "NGINX App Protect API Security Policy",
    "template": {
      "name": ”API_POLICY_NGINX"
    },
    "open-api-files": [
      {
        "link": "file:///etc/nginx/ergast_f1_oas.yml"
      }
    ],
    "blocking-settings": {
      "violations": [
        {
          "block": true,
          "description": "Mandatory request body is missing",
          "name": "VIOL_MANDATORY_REQUEST_BODY"
        },
...
```

```
$ curl -sH "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbiI6dHJ1ZX0.kFplw9Kkg-
6DLFGfVZAPIuWgGPMY9nnMZMQ2iIRN8_s" -i -X POST -d 'garbage123'
http://localhost:8080/api/f1/seasons

HTTP/1.1 403 Forbidden
{"supportID": "4839869788531770938"}




$ sudo cat /var/log/app_protect/security.log

attack_type="HTTP Parser Attack”...
support_id="4839869788531771448”...
```

In conclusion…


WAF
N+

Resources

FIND OUT MORE!

https://www.nginx.com/resources/library/nginx-api-gateway-deployment

BLOG TECH Liam Crilly of F5 January 20, 2021

Deploying NGINX as an API Gateway, Part 1

API gateway

TWITTER LINKEDIN Y

This is the first blog post in our series on deploying NGINX Open Source and NGINX Plus as an API gateway:

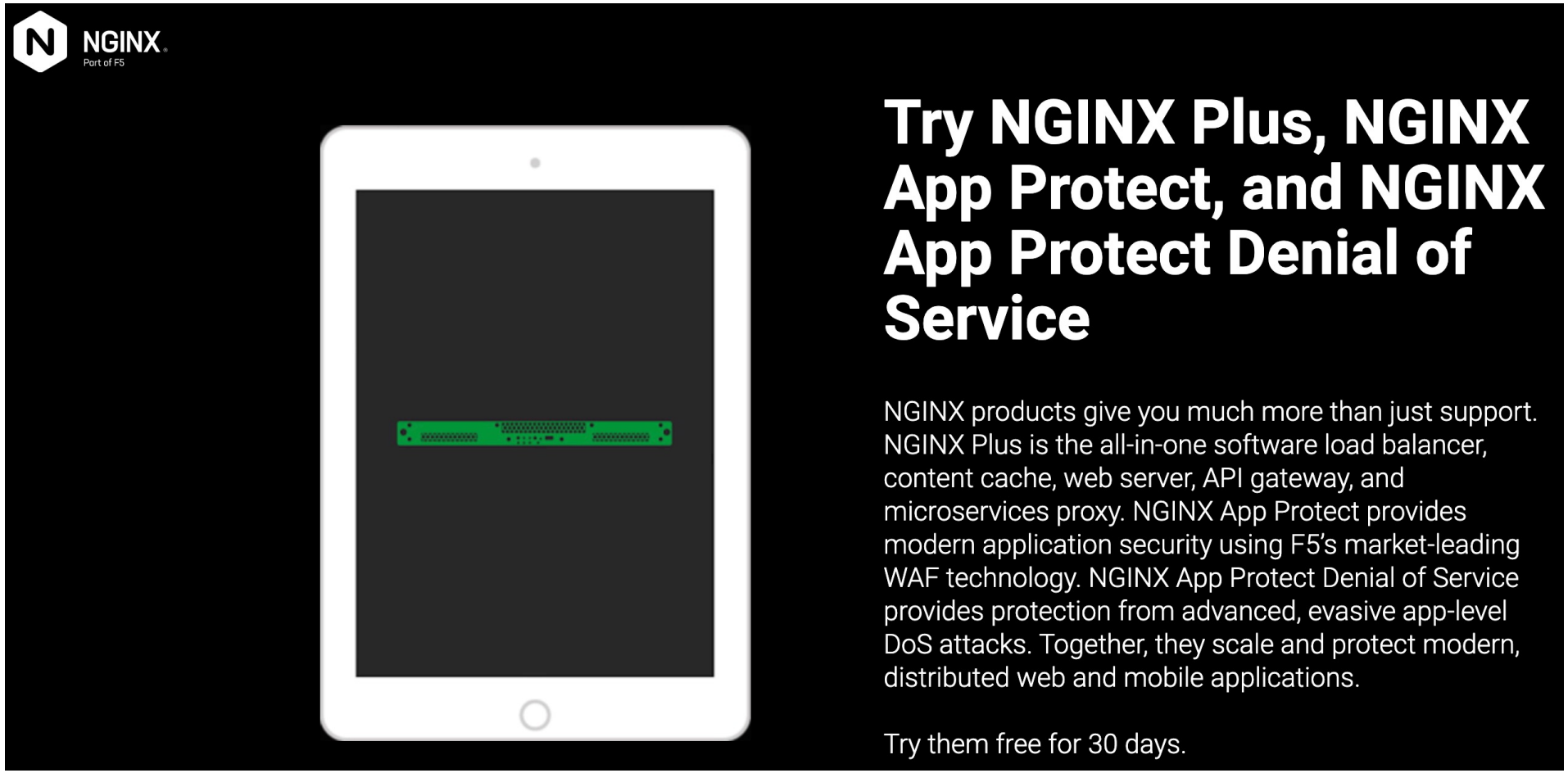
- *This post provides detailed configuration instructions for several use cases. Originally published in 2018, it has been updated to reflect current best practice for API configuration, using nested* `location` *blocks to route requests, instead of rewrite rules.*
- *Part 2 extends those use cases and looks at a range of safeguards that can be applied to protect and secure backend API services in production.*
- *Part 3 explains how to deploy NGINX Open Source and NGINX Plus as an API gateway for gRPC services.*

Note: Except as noted, all information in this post applies to both NGINX Open Source and NGINX Plus. For ease of reading, the rest of the blog refers simply to "NGINX".

https://www.nginx.com/blog/deploying-nginx-plus-as-an-api-gateway-part-1/

Check out NGINX Plus and NGINX App Protect!

DEPLOY AND SECURE YOUR API GATEWAY WITH NGINX'S FREE TRIAL -> HTTPS://WWW.NGINX.COM/FREE-TRIAL-REQUEST/



Thank you for attending!


a.faelgarcia@f5.com


alessfg


@alessfg

Q&A time!