



Airlock Microgateway

4.0

Release webinar
24. May 2023



Stefan Dietiker
Product Manager
Airlock Microgateway



Agenda



1. Security



2. Things to consider



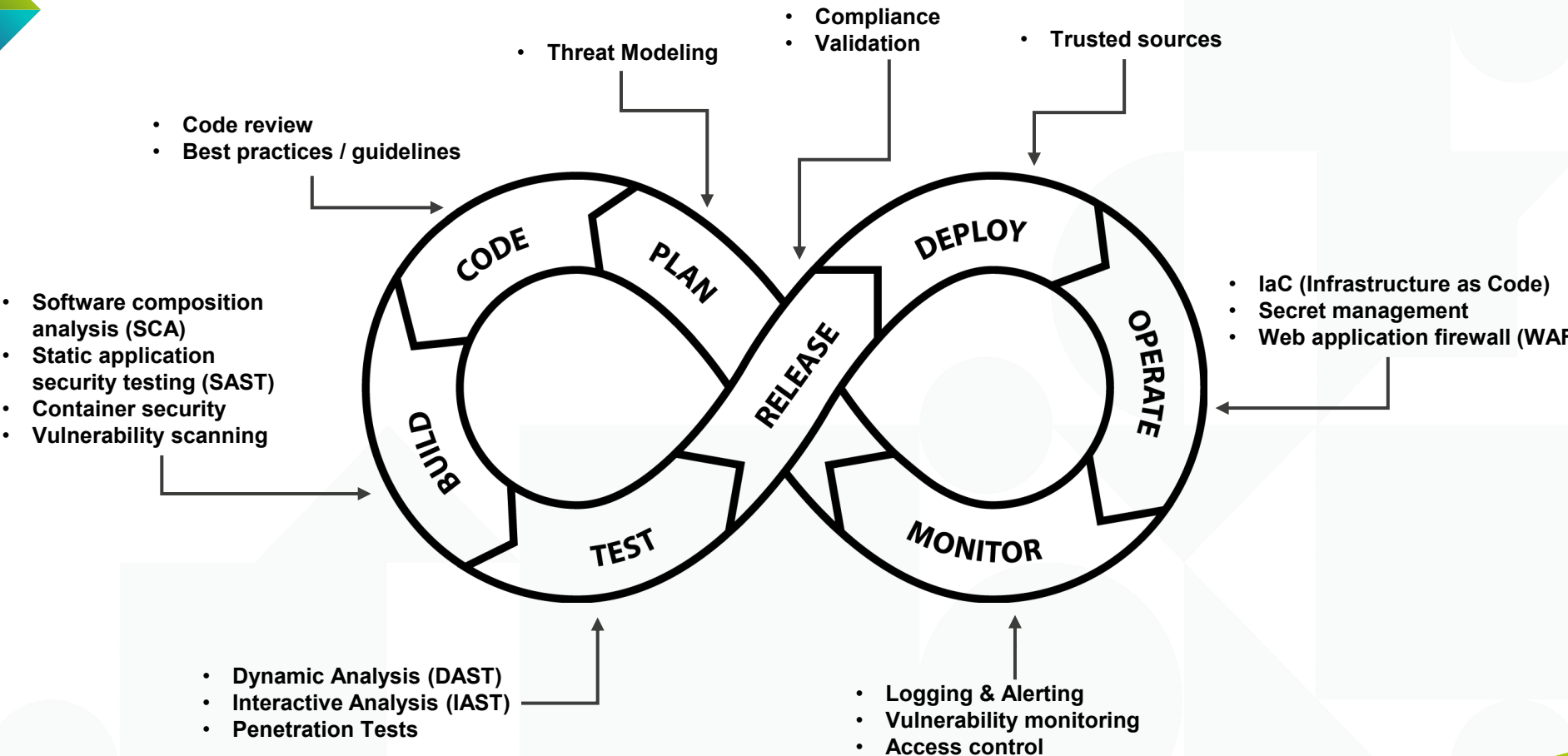
3. Microgateway



4. Demo

A photograph of a classroom with a green chalkboard. Several hands are raised in the air, indicating an interactive session or a Q&A period. The hands are in various positions, some fully open, some with fingers slightly curled. The background is a solid green chalkboard.

How to secure my web application?



Goals of Security Team



Eliminate vulnerabilities
before deployment

Filtering



+

Authentication



Protect at runtime



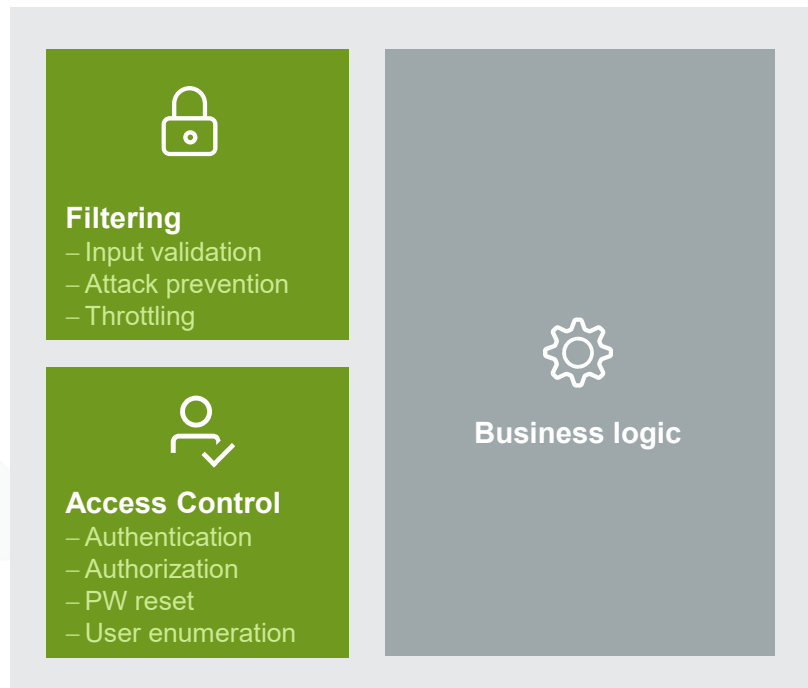
Compliance



Security

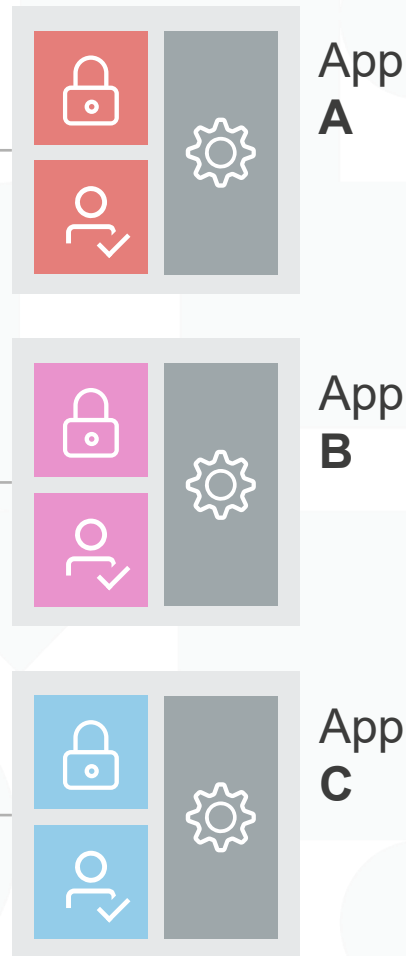
Business Logic

Tasks in a web application



Web application landscape

- 
App
- 
Browser
- 
IoT
- 
API



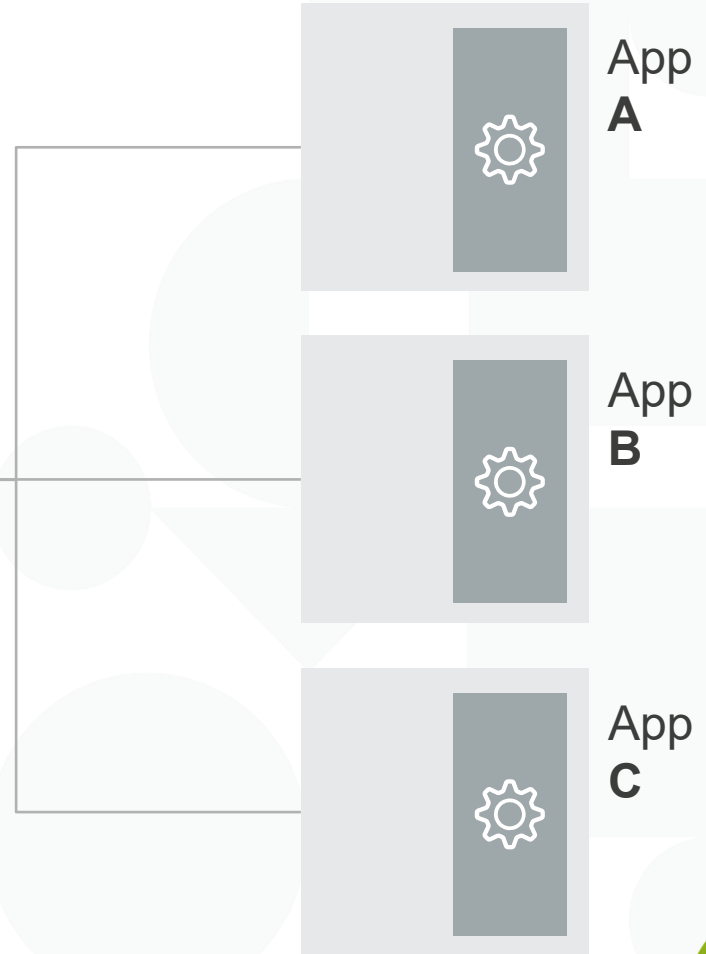
Centralized Security

Faster app development

- + Devs can focus on biz logic
- + Change auth method without affecting apps

Unified security

- + OWASP API Top 10 and Zero-Day attacks
- + Virtual patching
- + Bot & DoS protection
- + Company-wide Sec Policy
- + Language agnostic



App-specific security rules !?



App



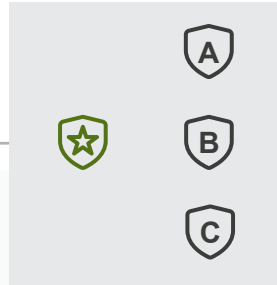
Browser



IoT



API

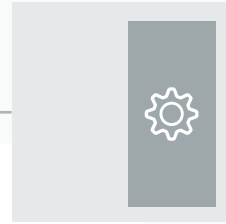


Global
policy

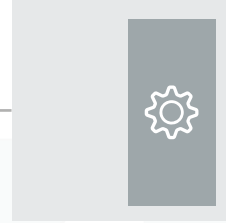
App
policies



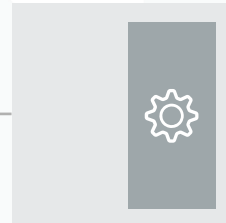
SecOps



App
A



App
B

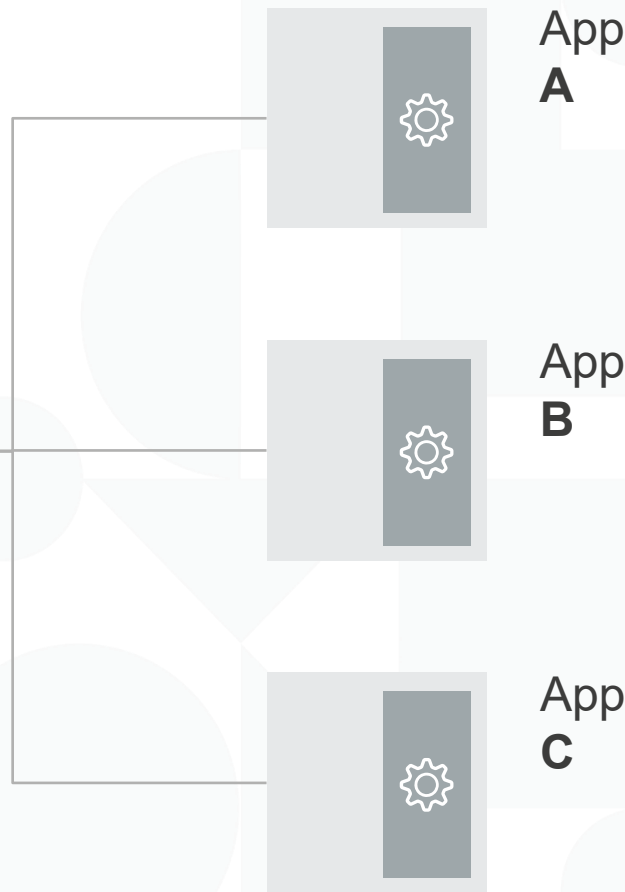
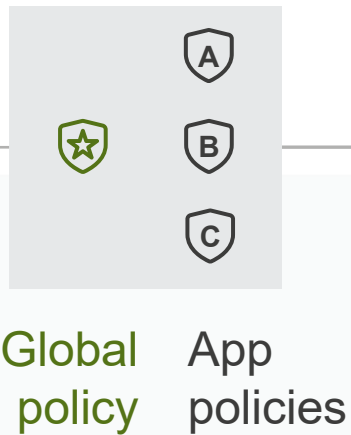


App
C

App Teams

From centralized...

- App
- Browser
- IoT
- API



...to distributed WAAP

**Security policy
bundled with application**

- + Devs not reinventing the wheel
- + Apps are independent

Edge Gateway



Microgateways

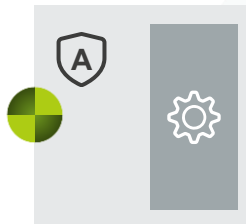


Secure and quick releases

Edge Gateway



Microgateway



Security



Defines company-wide policy



Enforces specific security settings



Observes active security settings



App team



Writes service specific security settings

New release

New release

New release

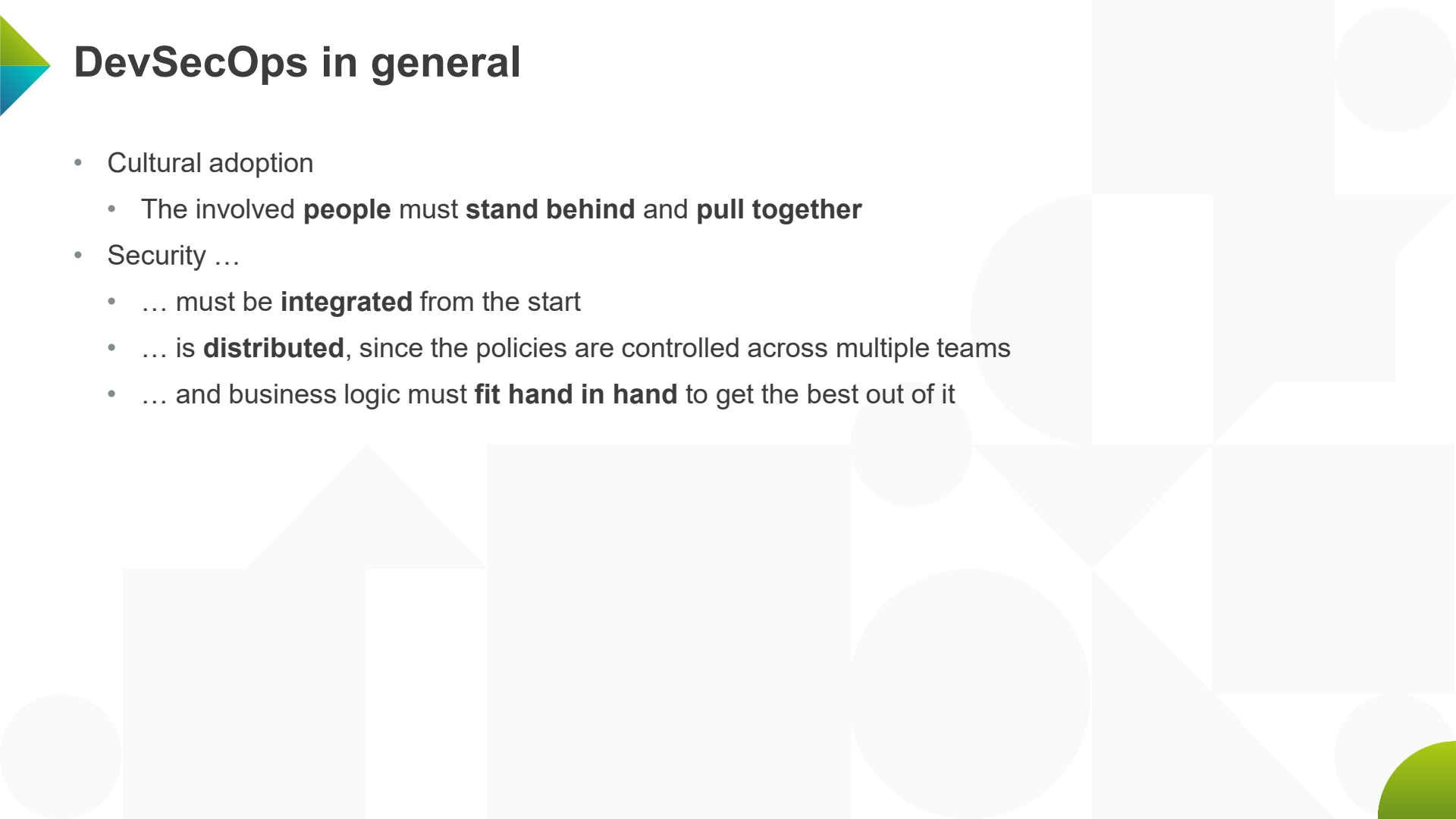


Things to consider





DevSecOps in general

- Cultural adoption
 - The involved **people** must **stand behind** and **pull together**
 - Security ...
 - ... must be **integrated** from the start
 - ... is **distributed**, since the policies are controlled across multiple teams
 - ... and business logic must **fit hand in hand** to get the best out of it
- 

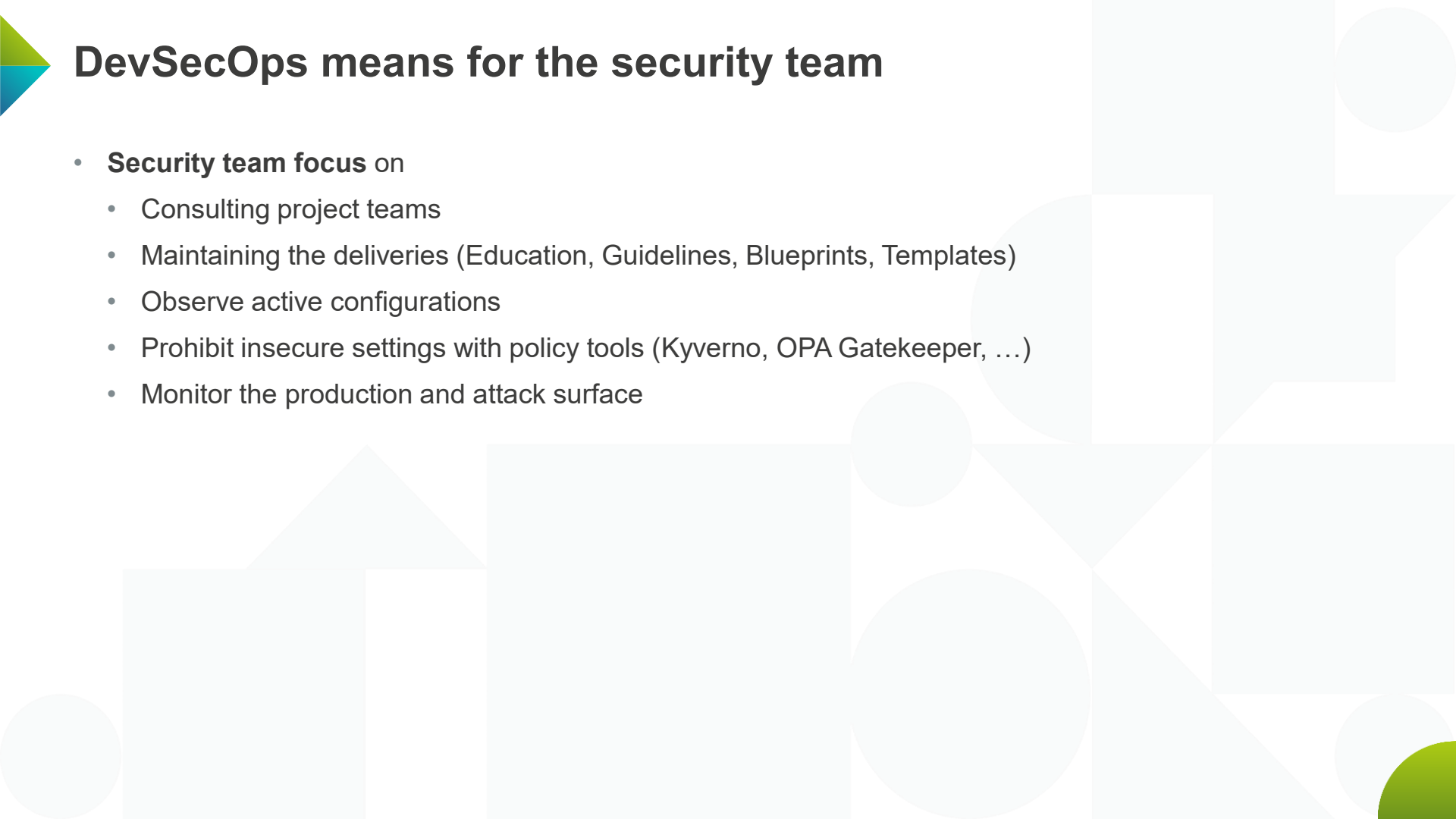


DevSecOps means for the developers

- **Developers** must be **empowered**
 - Education
 - Guidelines
 - Blueprints
 - Templates



DevSecOps means for the security team

- **Security team focus** on
 - Consulting project teams
 - Maintaining the deliveries (Education, Guidelines, Blueprints, Templates)
 - Observe active configurations
 - Prohibit insecure settings with policy tools (Kyverno, OPA Gatekeeper, ...)
 - Monitor the production and attack surface
- 



4.0

Microgateway News

Expected by end
of May 2023



Airlock Microgateway



Kubernetes
native

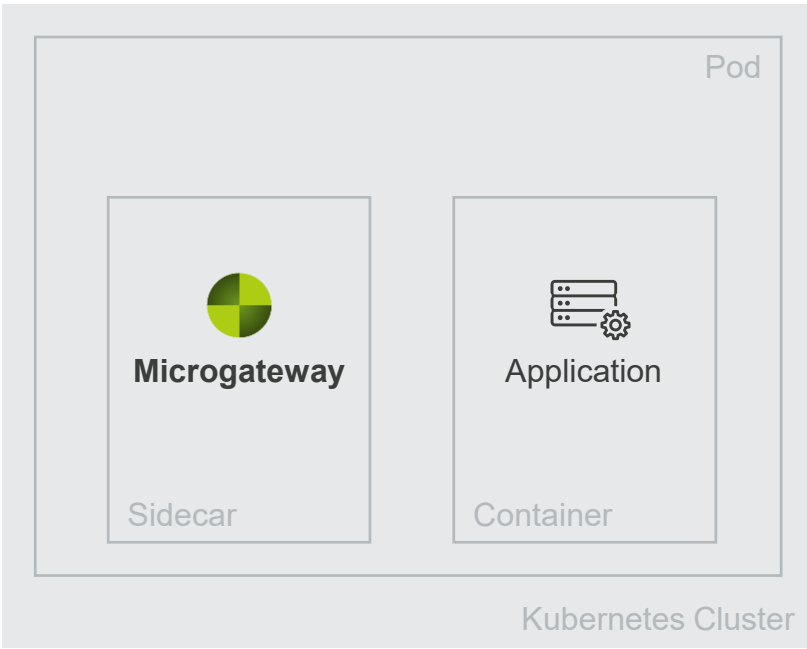


API + App
Firewall

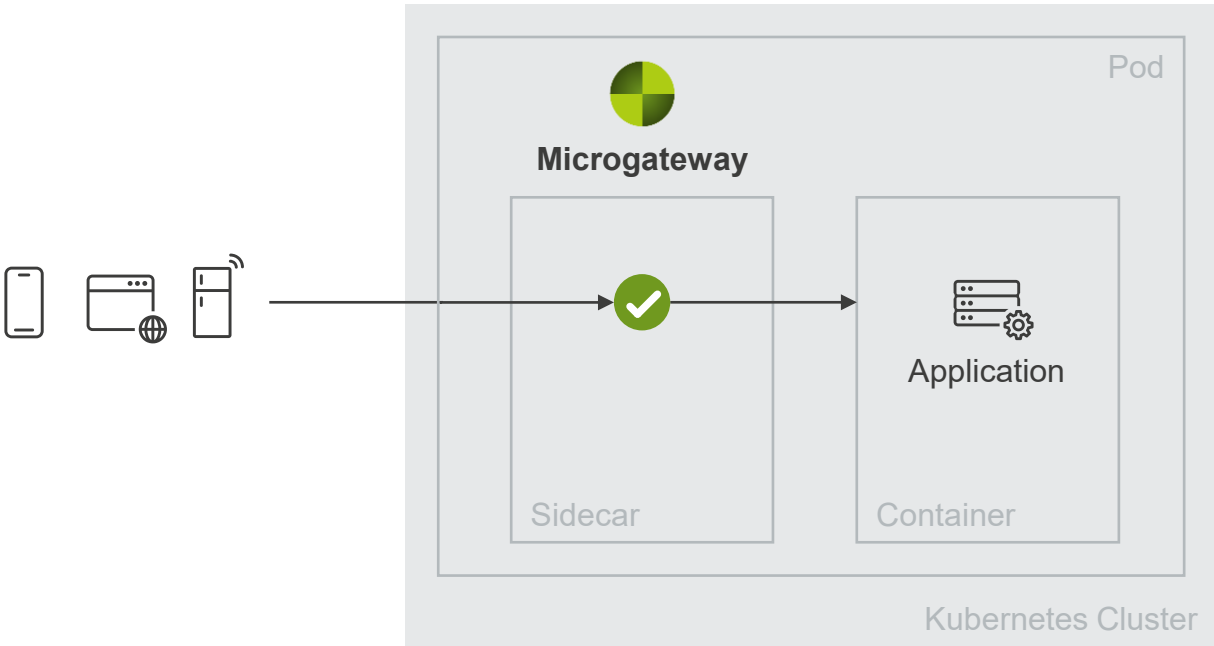


Based on
Envoy Proxy

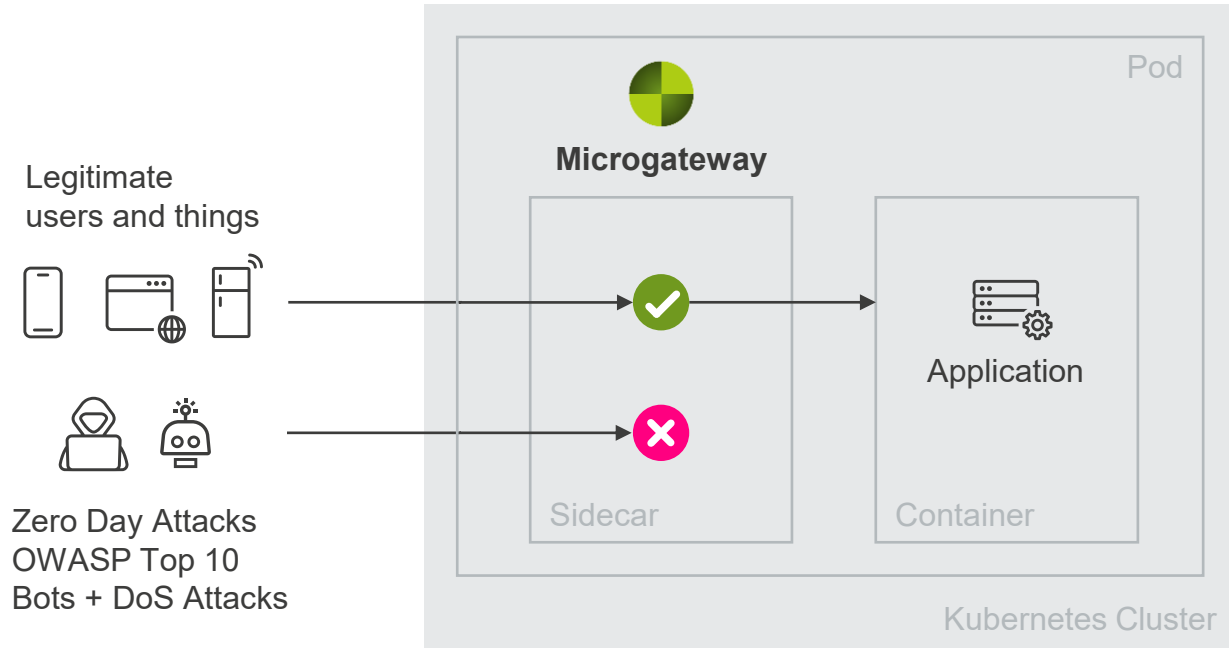
Built for Kubernetes.



Built for Kubernetes. Lets the good users in.



Built for Kubernetes.
Lets the good users in.
Keeps the bad stuff out.



Microgateway

4.0

- Envoy based
- Kubernetes native integration
 - Operator with its CRDs
- Automatic sidecar injection
- Hot-reload
- Istio support
- Community Edition with all features
- Small memory footprint
- HTTP/1.1 and HTTP/2
- mTLS for downstream connections
- Native Envoy HTTP filters
 - Lua, JWT, ext_authZ, ...
- Deny Rules
- Limits
- Header Rewrites
- JSON parsing
- Telemetry
 - Logging
 - Metrics

CRD SidecarGateway

- Select the Pods to apply the settings
- Security settings to apply for a containerPort or path
- Downstream (frontend) configuration like:
 - HTTP protocol, RemoteIP, TLS ciphers and protocols, TLS client certificate required/ignored
- Upstream (backend) configuration like:
 - HTTP protocol, TLS ciphers and protocols
- Native Envoy HTTP filters to prepend

```
com.airlock.microgateway.v1alpha1.SidecarGateway (v1alpha1@sidecargateway.js...
1  apiVersion: microgateway.airlock.com/v1alpha1
2  kind: SidecarGateway
3  metadata:
4    name: myservice
5  spec:
6    podSelector:
7      matchLabels:
8        app: myservice
9
10   applications:
11     - containerPort: 8443
12       routes:
13         - pathPrefix: /metrics
14           unsecured: {}
15         - pathPrefix: /
16           secured:
17             contentSecurityRef:
18               name: myservice
19
20       downstream:
21         remoteIP:
22           xff:
23             numTrustedHops: 1
24         tls:
25           enable: true
26           protocol:
27             minimum: TLSv1_2
28             maximum: TLSv1_3
29
30       upstream:
31         tls:
32           enable: false
33
34       telemetryRef:
35         name: myservice
36
37       envoyHTTPFilterRefs:
38         prepend:
39           - name: myservice
40
41       envoyClusterRefs:
42         - name: extAuthZ
43
```

4.0

CRD HeaderRewrites

- Allow, remove or add headers in request or response
- Built-in rules contain a predefined set of headers to be allowed, removed or added.
- With custom rules the behavior can be customized

```
{  
  {} headerrewrites.yaml > {} spec > {} request > {} add  
  com.airlock.microgateway.v1alpha1.HeaderRewrites (v1alpha1@headerrewrites.js  
1  apiVersion: microgateway.airlock.com/v1alpha1  
2  kind: HeaderRewrites  
3  metadata:  
4    name: myservice  
5  spec:  
6    request:  
7      allow:  
8        matchingHeaders:  
9          builtIn:  
10           standardHeaders: true  
11          custom:  
12            - name: Allow X-CSRF-Token header  
13              headers:  
14                - name:  
15                  matcher:  
16                    exact: X-CSRF-TOKEN  
17          remove:  
18            builtIn:  
19              alternativeForwardedHeaders: true  
20          add:  
21            custom:  
22              - name: Add headers with TLS information of the downstream connection  
23                headers:  
24                  - name: X-TLS-DOWNSTREAM-PEER-CERT  
25                    value: "%DOWNSTREAM_PEER_CERT%"  
26                mode: addIfAbsent  
27      response:  
28        allow:  
29          allHeaders: {}  
30        remove:  
31          builtIn:  
32            permissiveCors: true  
33        add:  
34          builtIn:  
35            csp: true  
36            featurePolicy: true  
37            hsts: true  
38            hstsPreload: false  
39            referrerPolicy: true  
40            xContentTypeOptions: true  
41            xFrameOptions: true  
42
```

4.0

CRD DenyRules

- Configure the security level
- Override the security level for some deny rules
- Configure exceptions for specific parameters, paths and deny rules
- Configure custom deny rules

```
(-) denyrules.yaml > apiVersion
com.airlock.microgateway.v1alpha1.DenyRules (v1alpha1@denyrules.json)
1  apiVersion: microgateway.airlock.com/v1alpha1
2  kind: DenyRules
3  metadata:
4    name: myservice
5  spec:
6    request:
7      builtIn:
8        settings:
9          level: strict
10         threatHandlingMode: block
11       overrides:
12         - conditions:
13           ruleKeys:
14             - XSS
15             - SQL
16           settings:
17             level: standard
18         exceptions:
19           - blockedData:
20             parameter:
21               source: any
22               name:
23                 matcher:
24                   ignoreCase: true
25                   exact: password
26           requestConditions:
27             path:
28               matcher:
29                 ignoreCase: true
30                 exact: /auth/login
31             method:
32               - POST
33           ruleKeys:
34             - XSS
35             - HTML
36             - HPP
37             - SQL
38             - TEMPLATE
39             - UNIXCMD
40             - WINCMD
41         custom:
42           rules:
43             - ruleKey: CM_REFERRER_BLOCK
44               blockData:
45                 header:
46                   name:
47                     matcher:
48                       exact: referer
49                   value:
50                     matcher:
51                       regex: .*bad.tv
52
```

4.0

CRD Limits

4.0

- General limits for request
- Limits for JSON
- Limits for HTTP parameters

Note:

The limit “bodySize” restricts the **parsed bodies** like JSON document in POST requests and **not uploaded files** (formUpload). This setting should be as small as possible.

```
(-) limits.yaml > apiVersion
com.airlock.microgateway.v1alpha1.Limits (v1alpha1@limits.json)
1  apiVersion: microgateway.airlock.com/v1alpha1
2  kind: Limits
3  metadata:
4    name: myservice
5  spec:
6    request:
7      limited:
8        general:
9          bodySize: 100Ki
10         pathLength: 1Ki
11       json:
12         elementCount: 10000
13         keyCount: 250
14         nestingDepth: 100
15         keyLength: 128
16         valueLength: 8Ki
17       parameter:
18         count: 128
19         nameLength: 128
20         valueLength: 8Ki
21
```



Further benefits

4.0

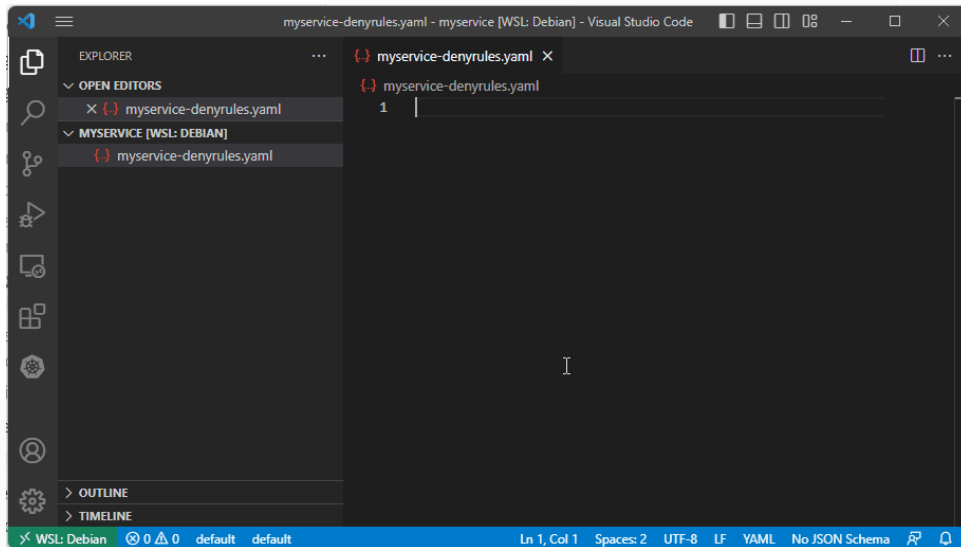
- **Enforce policies**
 - E.g., disallow in production Deny Rules in logOnly mode
- **Observe active configurations**
 - Use kubectl to figure out which settings are configured
- **Templating**
 - Use kustomize or build your helm chart for templating
- **Declarative configuration**
 - Security as code. Besides versioning, auditing and diffing, the configuration can be deployed with GitOps tools.

Completion, validation and tooltip

Plugins for Visual Studio Code and other IDEs provide:

- Completion
- Validation
- Tooltip

Simplifies configuration tasks **tremendously**.



Documentation

4.0

- Microgateway documentation explaining the architecture, concepts and use cases.
- API reference explaining the CRD settings.

Getting started – installation and follow-ups

This getting started lists the steps that are required to install and configure Airlock Microgateway.

Start with Airlock Microgateway

Follow these steps to get started with Airlock Microgateway:

1. [About releases](#)
Check the release notes for new functionalities and changes.
2. [Requirements and Limitations](#)
Ensure that the requirements are met.
3. [Container image repositories and registries](#)
Ensure you have access to the Image registry.
4. [Installation](#)
Follow the installation guide.

Follow up

What's next:

1. [System architecture](#)
Understand the architecture and where Airlock Microgateway fits in.
2. [Configuration](#)
Get familiar with the required configuration resources and the concept behind them.
3. [Integration tasks](#)
Check out the guides to implement your use cases.

The screenshot shows the Airlock Microgateway documentation website. On the left is a navigation sidebar with the Airlock logo and a search bar. The main content area displays the 'DenyRules' API reference page. The page title is 'DenyRules' with the URL 'microgateway.airlock.com/v1alpha1'. Below the title, it states 'DenyRules is the Schema for the denyrules API'. There are two tabs: 'Minimal Usage' and 'Default'. The 'Default' tab is active, showing a code block with the following JSON schema:

```
apiVersion: microgateway.airlock.com/v1alpha1
kind: DenyRules
metadata:
  name: default
spec:
  request:
    builtin:
      settings:
        level: standard
        threatHandlingMode: block
    custom: {}
```

Below the code block is a table titled 'DenyRules' with the following structure:

Field	Type	Description	Required	Default	Allowed Values
metadata	ObjectMeta	Refer to Kubernetes API documentation for fields of metadata	yes		
spec	object	Specification of the desired deny rules behavior.	no		

Why Envoy?

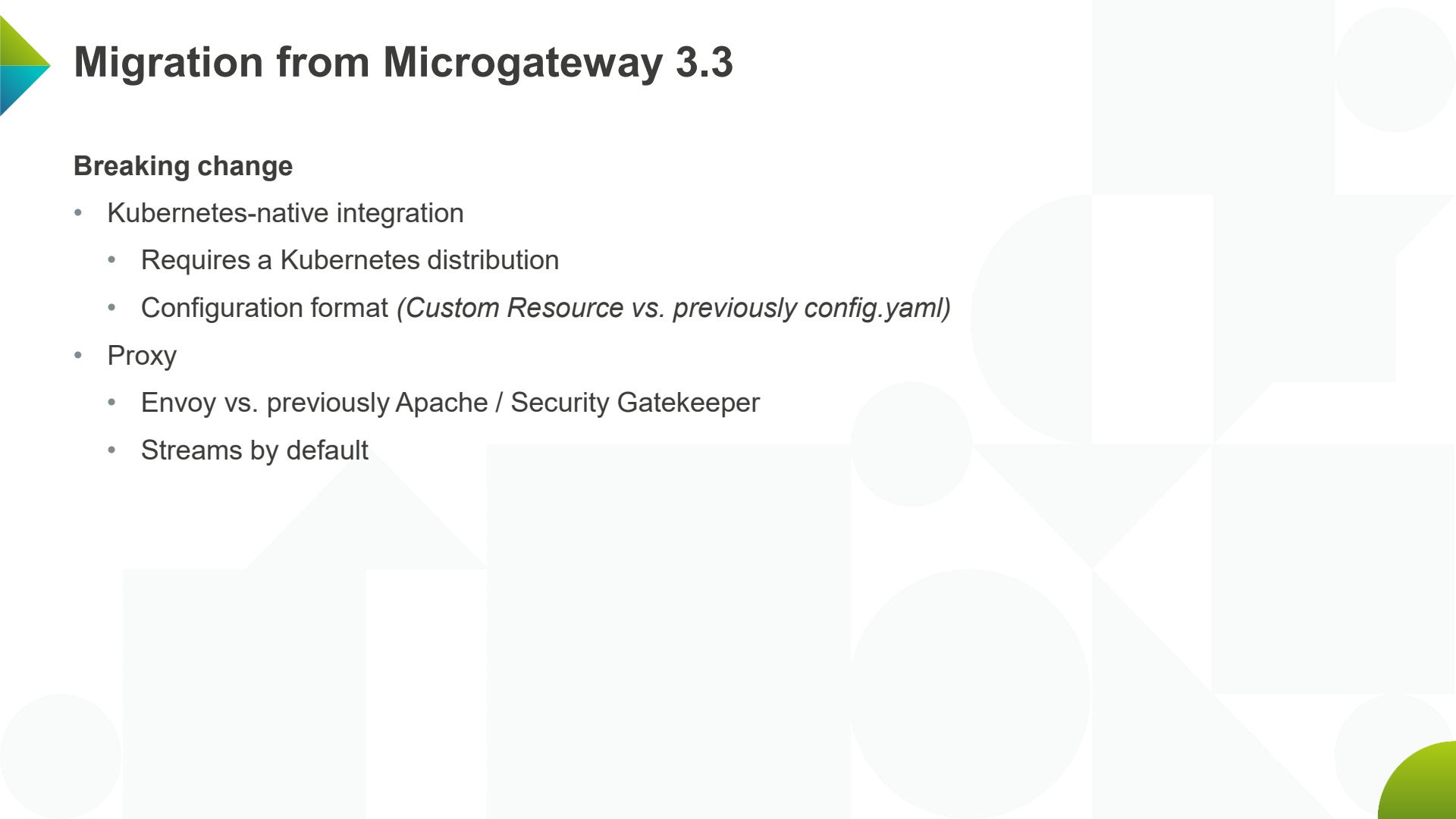
- “Envoy is the de facto standard for Kubernetes data plane” (Gartner)
- Cloud-native architecture
- Support for modern web protocols (e.g., HTTP/3, gRPC)
- Extensible with Lua scripts
- Apache 2 license





Migration from Microgateway 3.3

Breaking change

- Kubernetes-native integration
 - Requires a Kubernetes distribution
 - Configuration format (*Custom Resource vs. previously config.yaml*)
 - Proxy
 - Envoy vs. previously Apache / Security Gatekeeper
 - Streams by default
- 

Migration from Microgateway 3.3

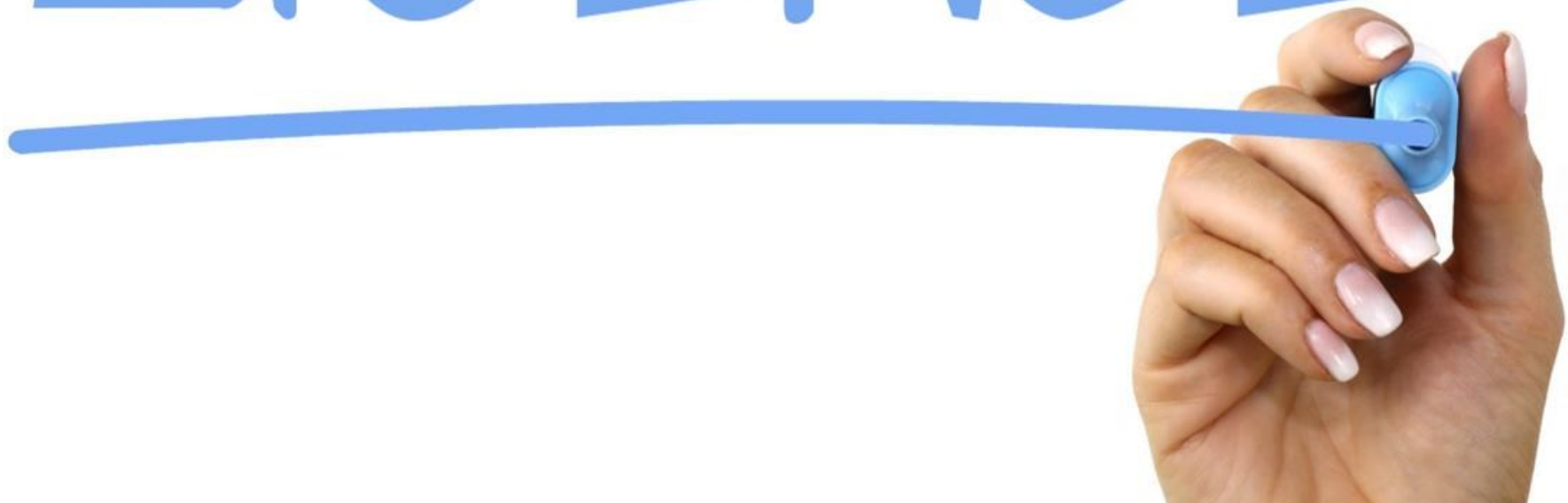
How to migrate?

- Re-integrate the web application
- Use the old configuration file as a reference
 - Configure allowed request/response headers in header rewrites
 - Start with the old deny rules configuration
 - Security level
 - Deny rule exceptions

Do not copy & paste the following settings

- Limits
 - Use the new and recommended default values and only increase them if really required.
 - This setting restricts the **parsed bodies** like JSON document in POST requests and **not uploaded files** (formUpload). This behavior has changed from Microgateway 3.3 to 4.0.

LICENSE



Airlock Microgateway Freemium Model

Feature		Community edition	Premium edition
Kubernetes native integration Operator and CRDs, hot-reload, Istio	Intended for small environments or local development	✓	✓
Reverse proxy Request routing, TLS termination, client certificate authentication, remote IP extraction, ...		✓	✓
Content security (OWASP Top 10) Deny rules against attacks (SQLi, XSS)	Community edition has all features	✓	✓
Access control ³⁾ Authentication enforcement, OIDC, ...		✓	✓ ¹⁾
API security JSON Parser, OpenAPI specifications enforcement ³⁾ , ...		✓	✓
Observability Structured logs and Prometheus metrics.		✓	✓
Throughput Number of concurrent allowed requests over all Airlock Microgateways.		5 requests per second	Up to unlimited ¹⁾
Sidecars Number of Pods that Airlock Microgateway runs per namespace		3	Unlimited
License expiration Time period the license is valid before expiration	Community edition has more restrictive operational parameters	6 months	12 months
Service life ²⁾ The version which is enforced by the license		The oldest supported version	unrestricted
Training Documentation, Kubernetes manifest files, examples, virtual labs, ...		✓	✓
Support		Community Support	Premium Support



Community Edition

Public Forum

- <https://forum.airlock.com/>
- Requires account for questions and commenting
- Airlock moderation

Support level by Airlock

- Hints, pointers to documentation and best practices
- No case analyses
- No guaranteed response times
- Complex cases require premium support



How to get a license?

Community license

Salutation*

Mr

Mrs

Firstname*

Surname*

Job title

Company

Email*

Premium license

Salutation*

Mrs Mr

Preferred Partner

Your preferred partner

Herewith I agree with the privacy declaration. *

Enter the characters you

see

[New](#) | [Audio](#)



Send

There is no built-in license!

<https://airlock.com/community>

<https://airlock.com/microgateway-premium>

A pair of red curtains is shown, partially open, with a spotlight effect illuminating the center. The word "Demo" is written in white text in the center of the spotlight.

Demo

<https://play.instruqt.com/airlock/invite/czwug0sytedy>

(start time about 3 min / valid until 31. May 2023, 18:00 o'clock)



Outlook

Future Microgateway releases

4.1

Microgateway 4.1

Planned features:

- CNI Plugin
- Multi-Namespace support
- Allow rules

4.x

Future Microgateway 4.x releases

Possible features:

- OpenAPI specification enforcement
- Session handling
- Authentication enforcement
- Identity propagation
- Anomaly shield



Thank you.

www.airlock.com/microgateway

docs.airlock.com/microgateway/latest/

Stefan Dietiker
stefan.dietiker@airlock.com

