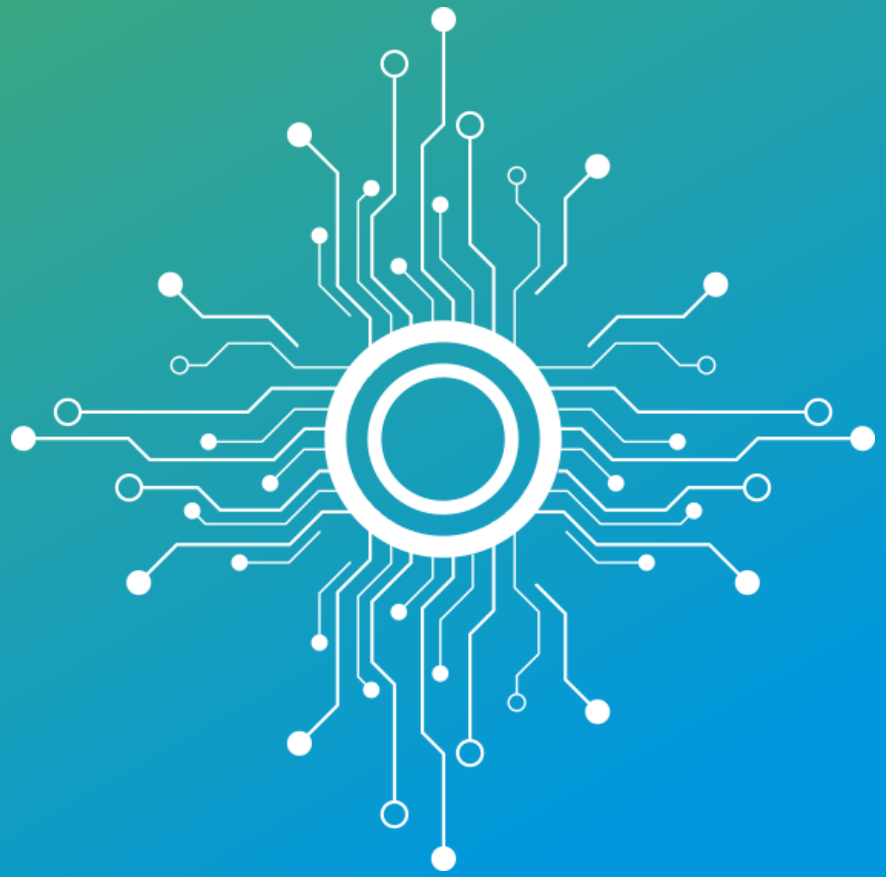
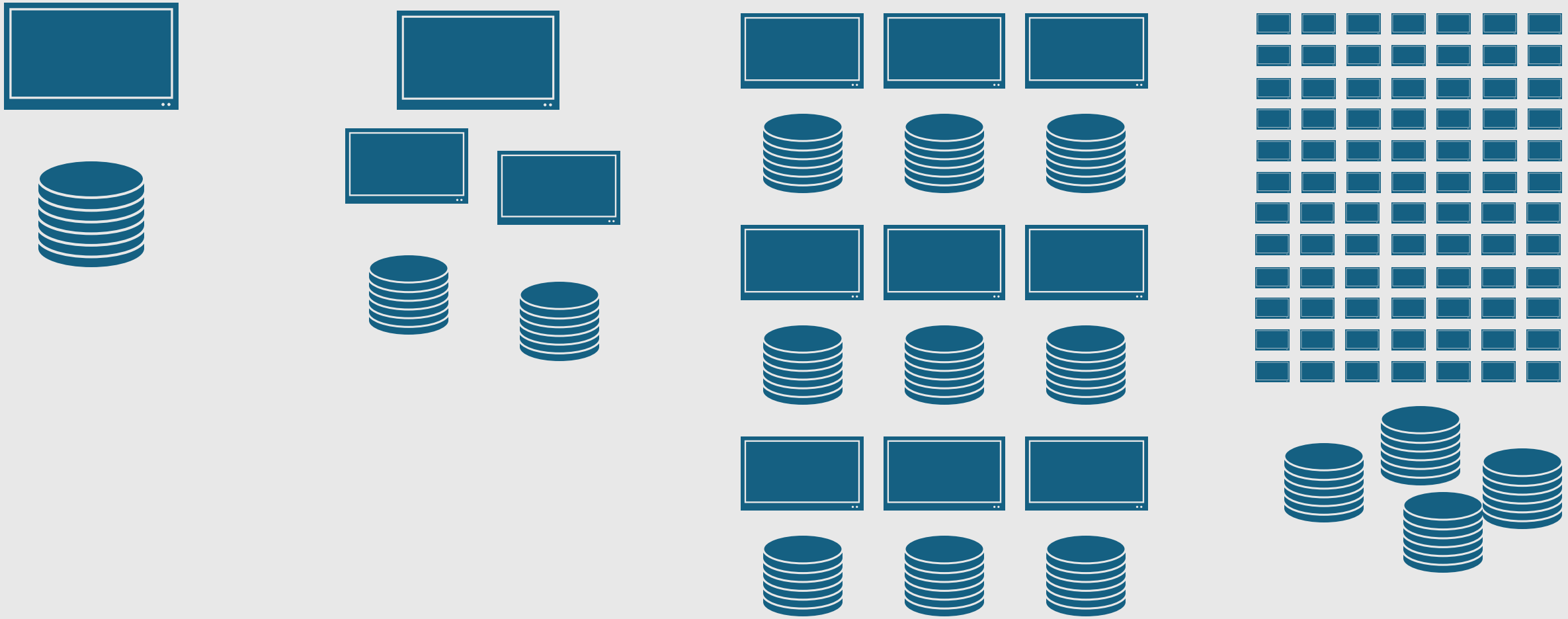
Is the **Lambdalith** the sweet spot in modern Cloud architecture?



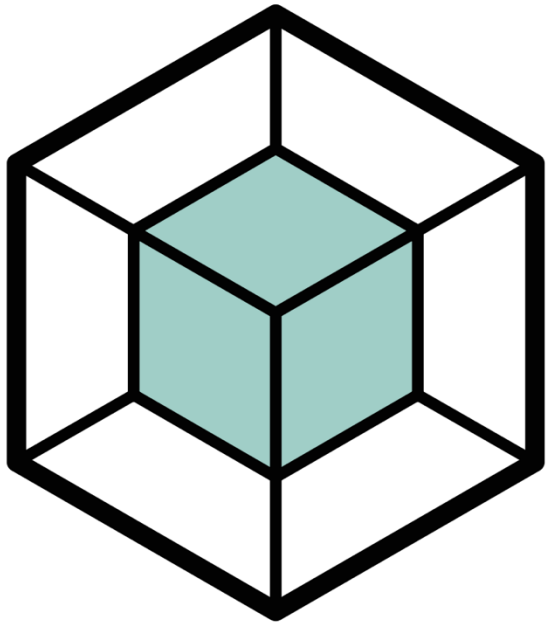
LOUPE
focused on
business integrity

Monolithic > Modular > Microservices > Serverless



Cloud computing evolves

>



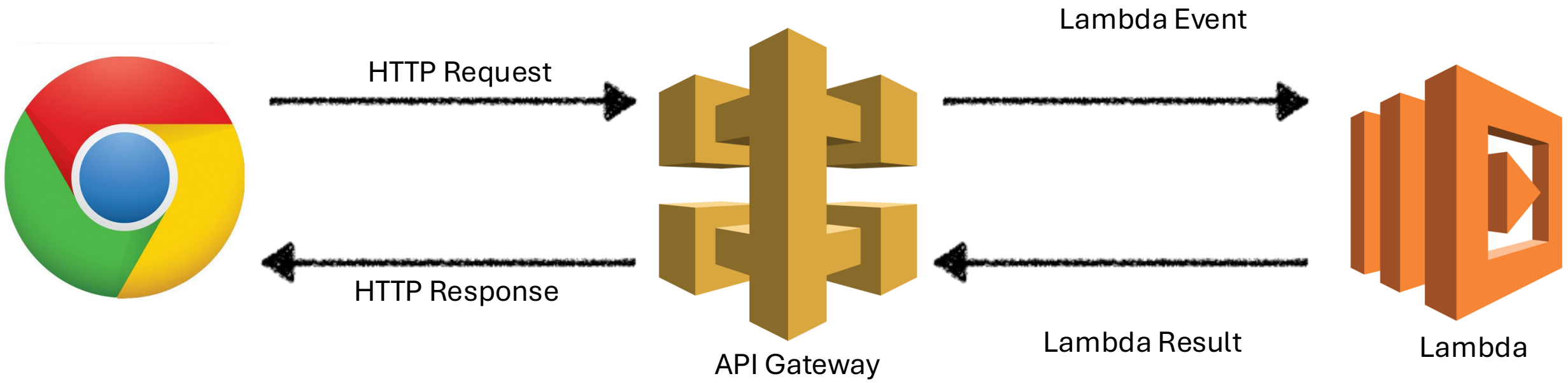
>

EC2
Virtual servers

Beanstalk, EKS, Fargate
Managed containers

Lambda
Function as a Service

Anatomy of a Serverless request



Serverless.com Example

serverless.yml

```yaml
functions:
  createNote:
    handler: api.createNote
    events:
      - http:
          method: post
          path: notes
  listNotes:
    handler: api.listNotes
    events:
      - http:
          method: get
          path: notes
  getNote:
    handler: api.getNote
    events:
      - http:
          method: get
          path: notes/{uuid}
  putNote:
    handler: api.putNote
    events:
      - http:
          method: put
          path: notes/{uuid}
```

api.ts

```typescript
export const createNote: APIGatewayProxyHandler = async (e, _c) => {
  const noteBody = <NoteBody>Requests.body(e);
  const note = await notesRepository.createNote(noteBody);
  return Responses.success(note);
}

export const listNotes: APIGatewayProxyHandler = async (_e, _c) => {
  const notes = await notesRepository.listNotes();
  return Responses.success(notes);
}

export const getNote: APIGatewayProxyHandler = async (e, _c) => {
  const uuid = e.pathParameters['uuid'];
  const note = await notesRepository.getNote(uuid);
  return Responses.success(note);
}

export const updateNote: APIGatewayProxyHandler = async (e, _c) => {
  const uuid = e.pathParameters['uuid'];
  const note = <Note>Requests.body(e);
  if(uuid !== note.uuid) {
    return Responses.error(409,
        `UUID in body and on request path did not match.`);
  }
  const updatedNote = await notesRepository.putNote(note);
```

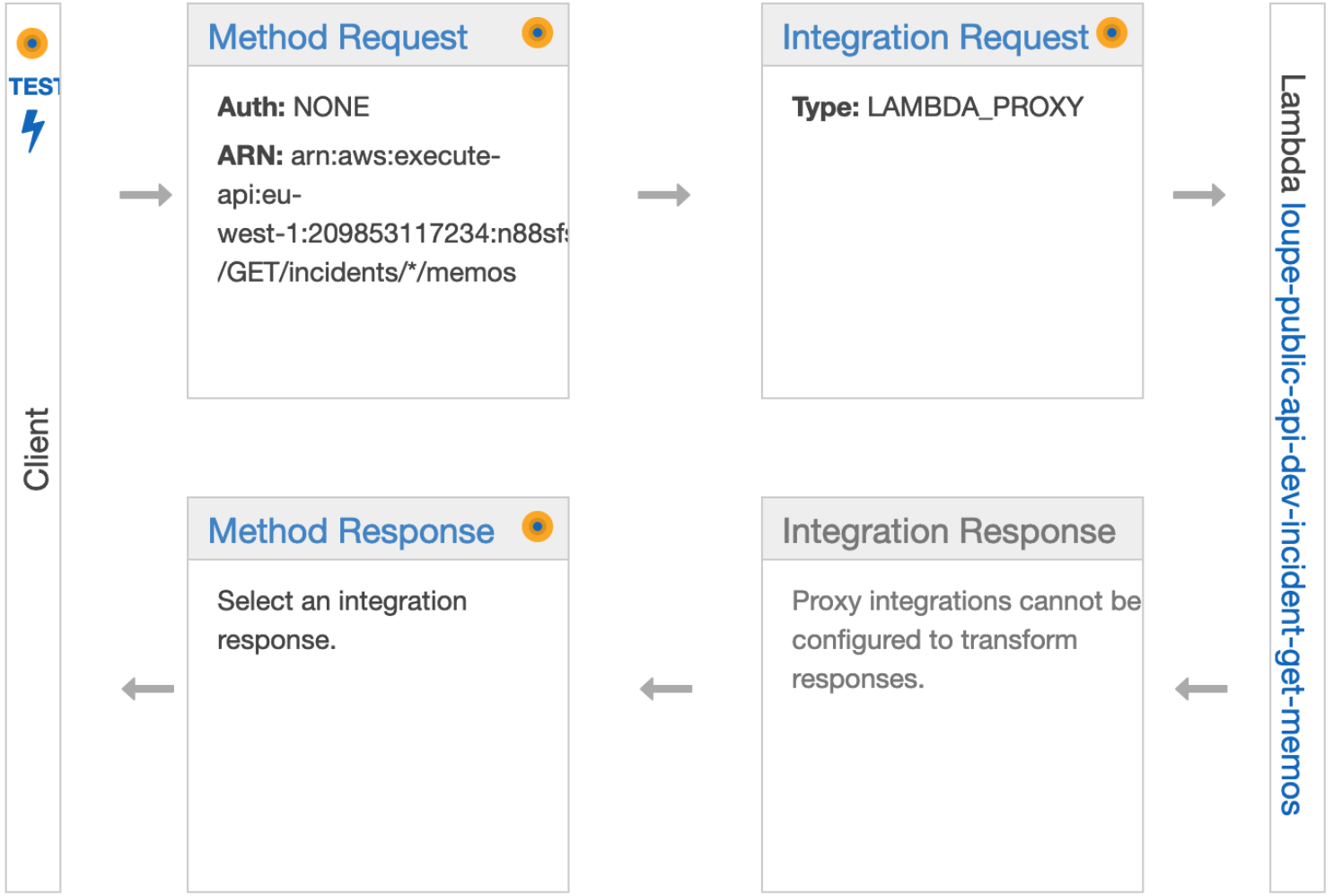
APIs
Custom Domain Names

API: dev-loupe-publ...
Resources
Stages
Authorizers
Gateway Responses
Models
Resource Policy
Documentation
Dashboard
Settings
Usage Plans
API Keys

Resources | Actions

- /
  - /incidents
    - OPTIONS
    - POST
    - /{code}
      - GET
      - OPTIONS
      - PUT
      - /memos
        - GET
        - OPTIONS
      - /messages
        - GET
        - OPTIONS
        - POST
      - /prepare-file-upload
        - OPTIONS
        - POST
      - /submit
        - OPTIONS
        - PUT
  - /readonly
    - /{uuid}
      - GET

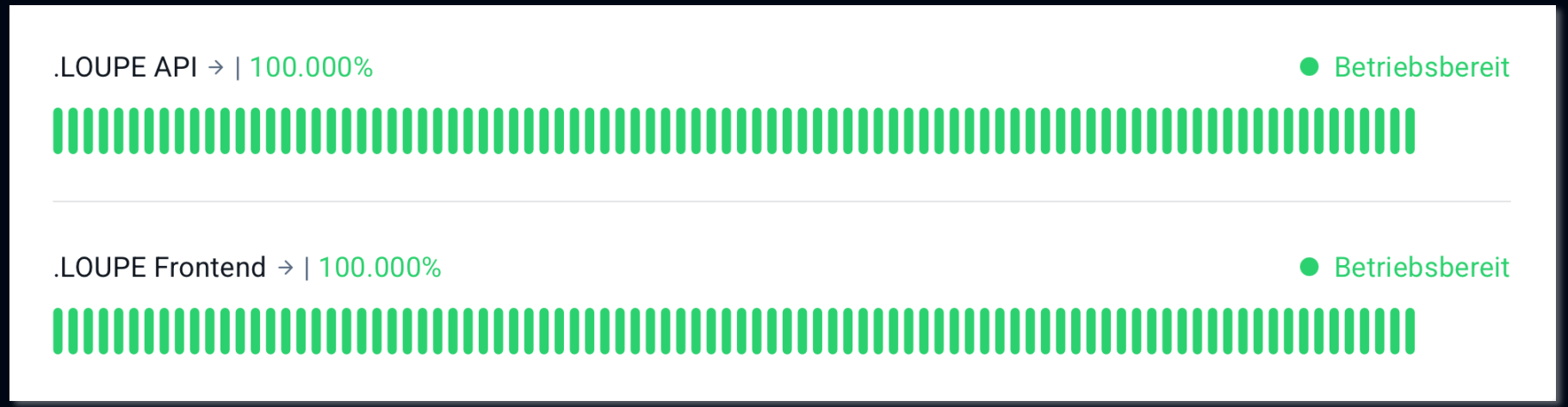
/incidents/{code}/memos - GET - Method Execution



Serverless advantages

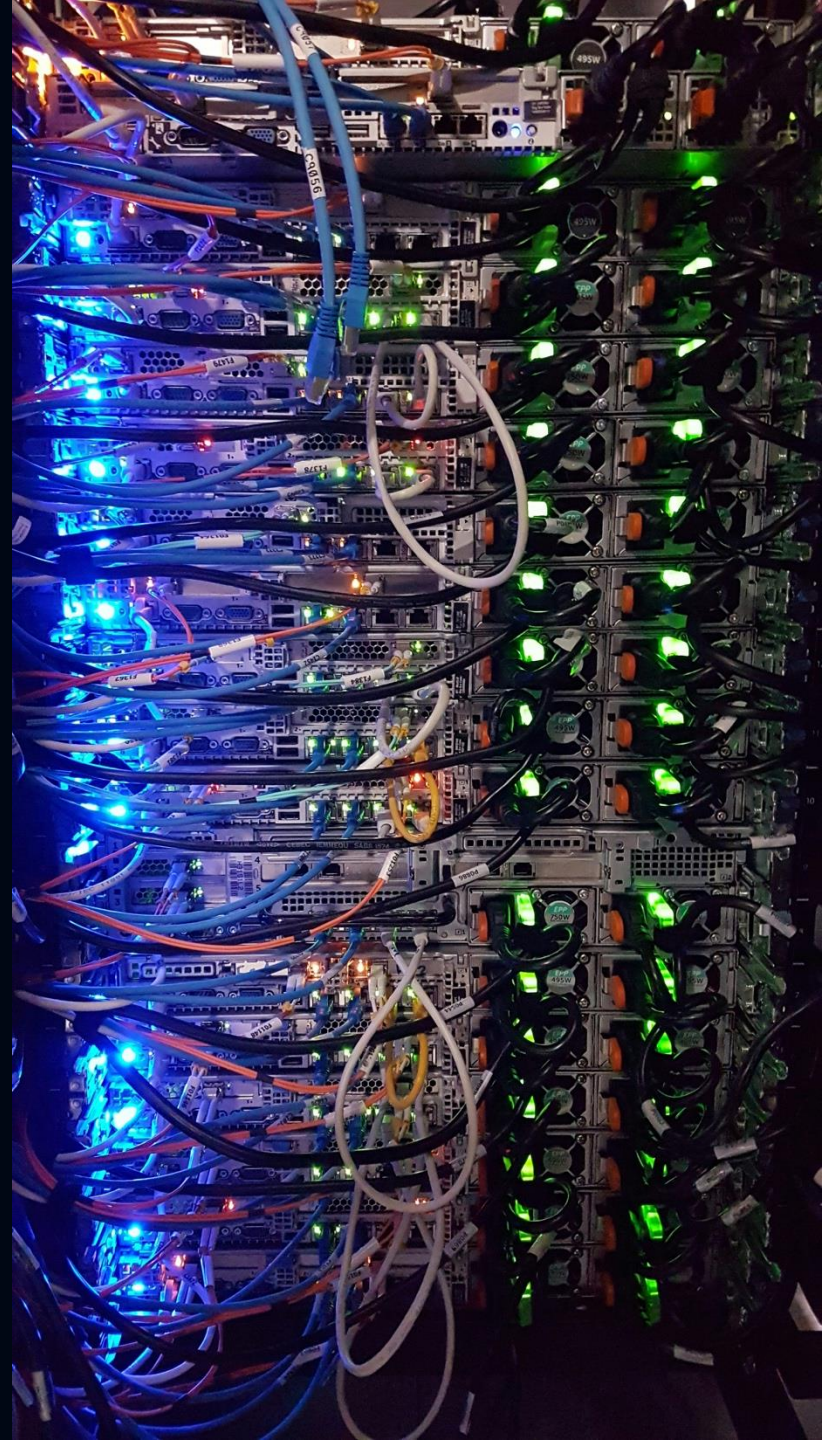
- Small teams can **hand off mundane** and potentially **hard tasks** to the Cloud vendor

Rolling releases, excellent uptime



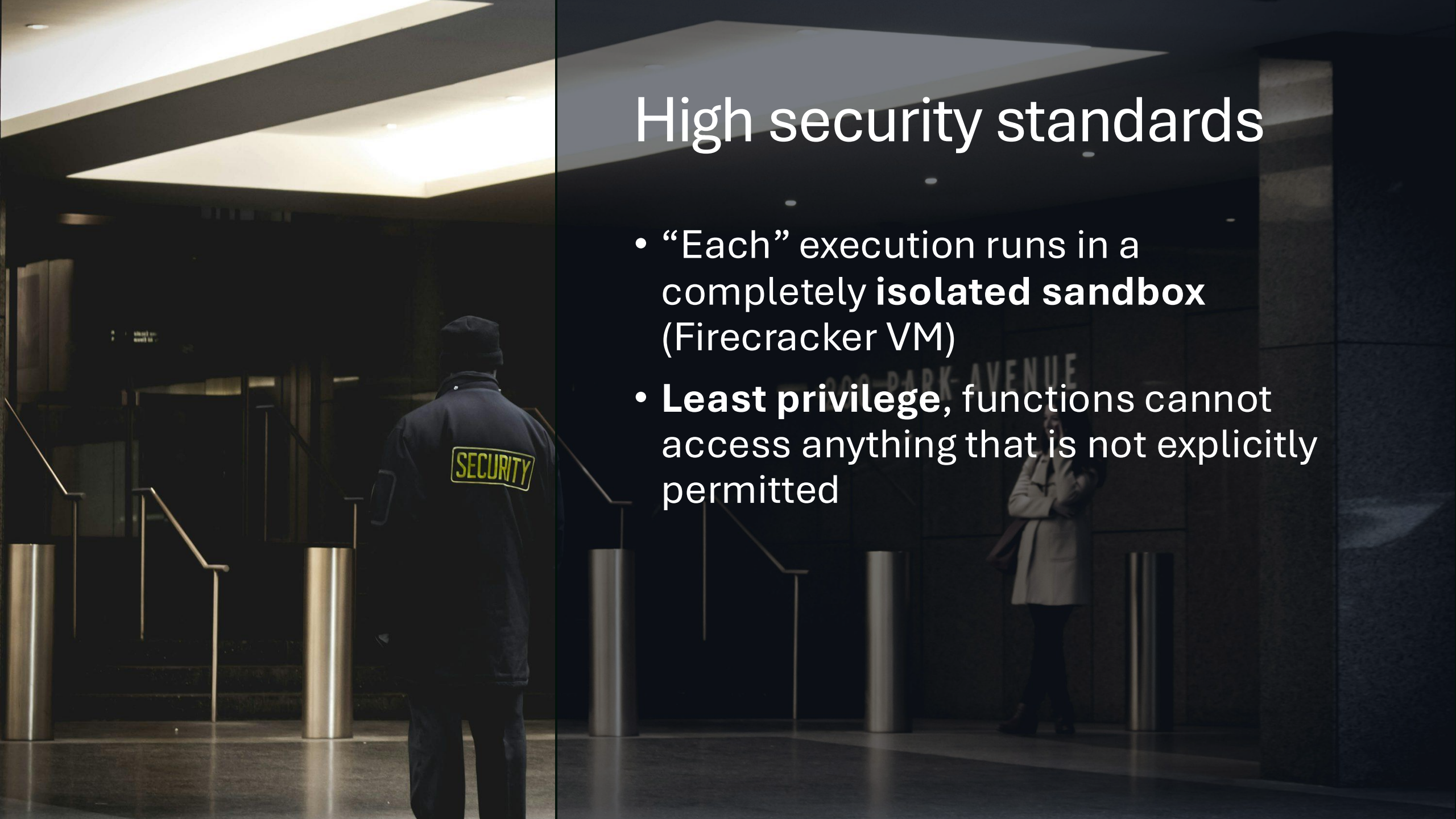
Zero infrastructure management

- Not ”affordable” to a small team
- Unpredictable tasks



Automatic scaling

- Scaling is again a topic that is hard to get right for a small team
- Serverless allows to **scale to 0**

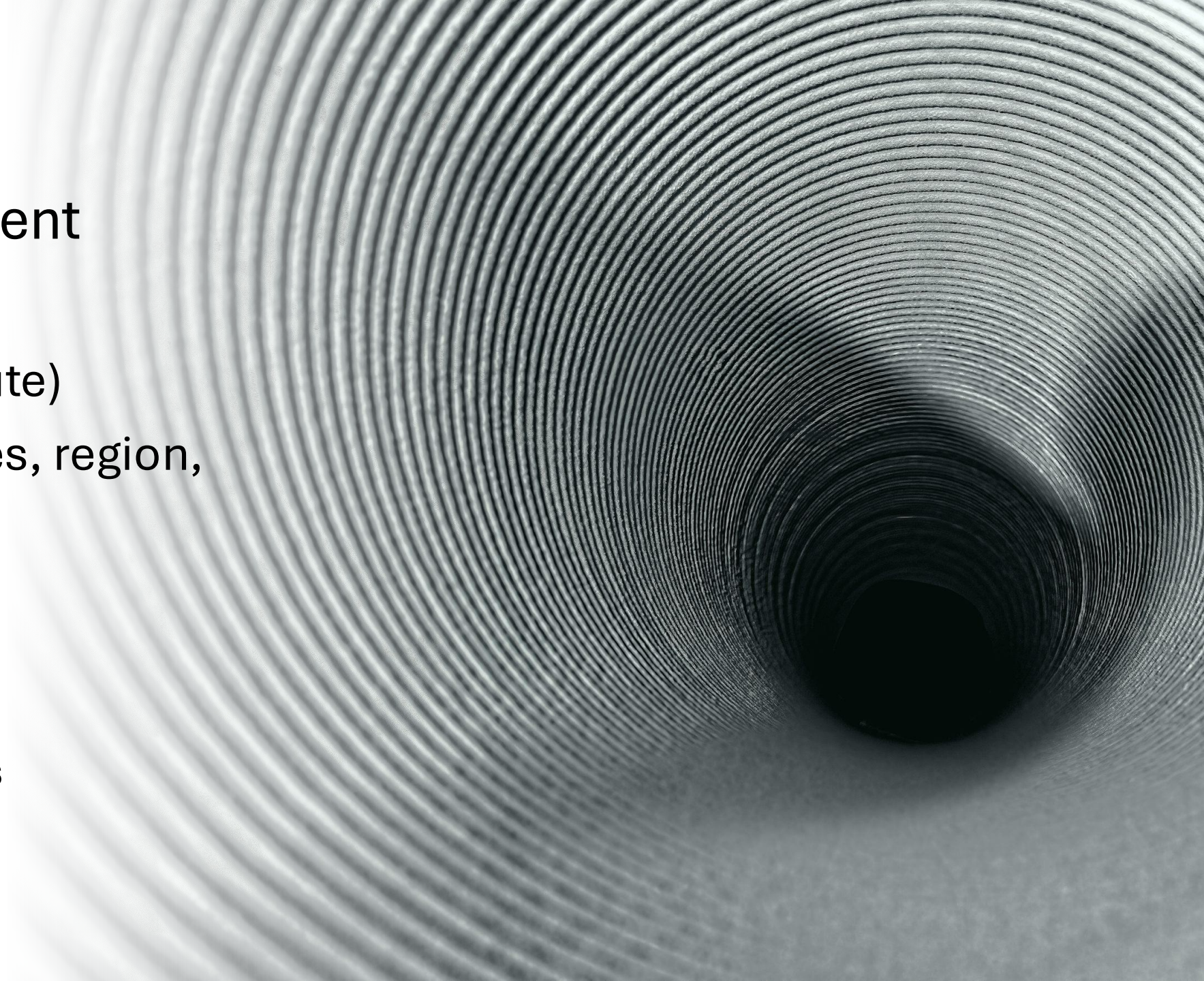


High security standards

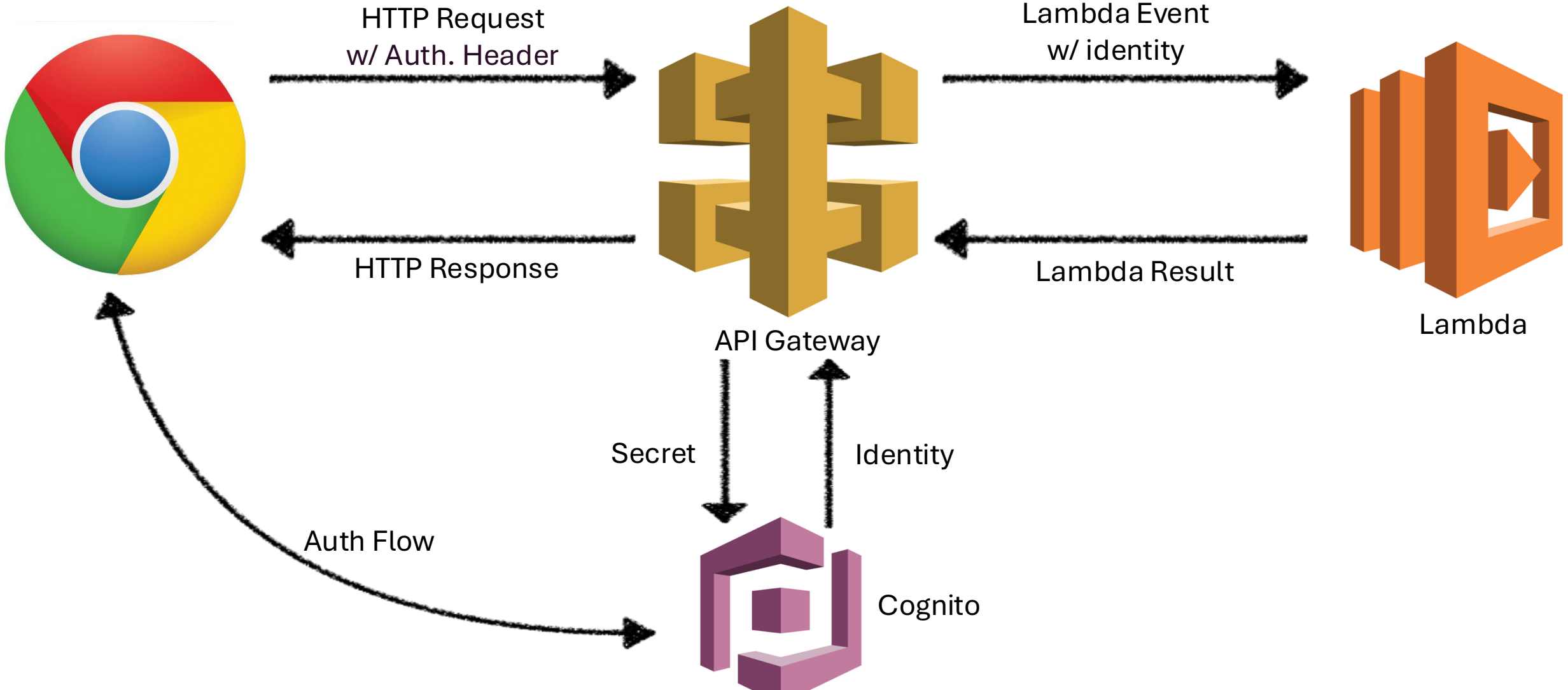
- “Each” execution runs in a completely **isolated sandbox** (Firecracker VM)
- **Least privilege**, functions cannot access anything that is not explicitly permitted

API Gateway request enrichment

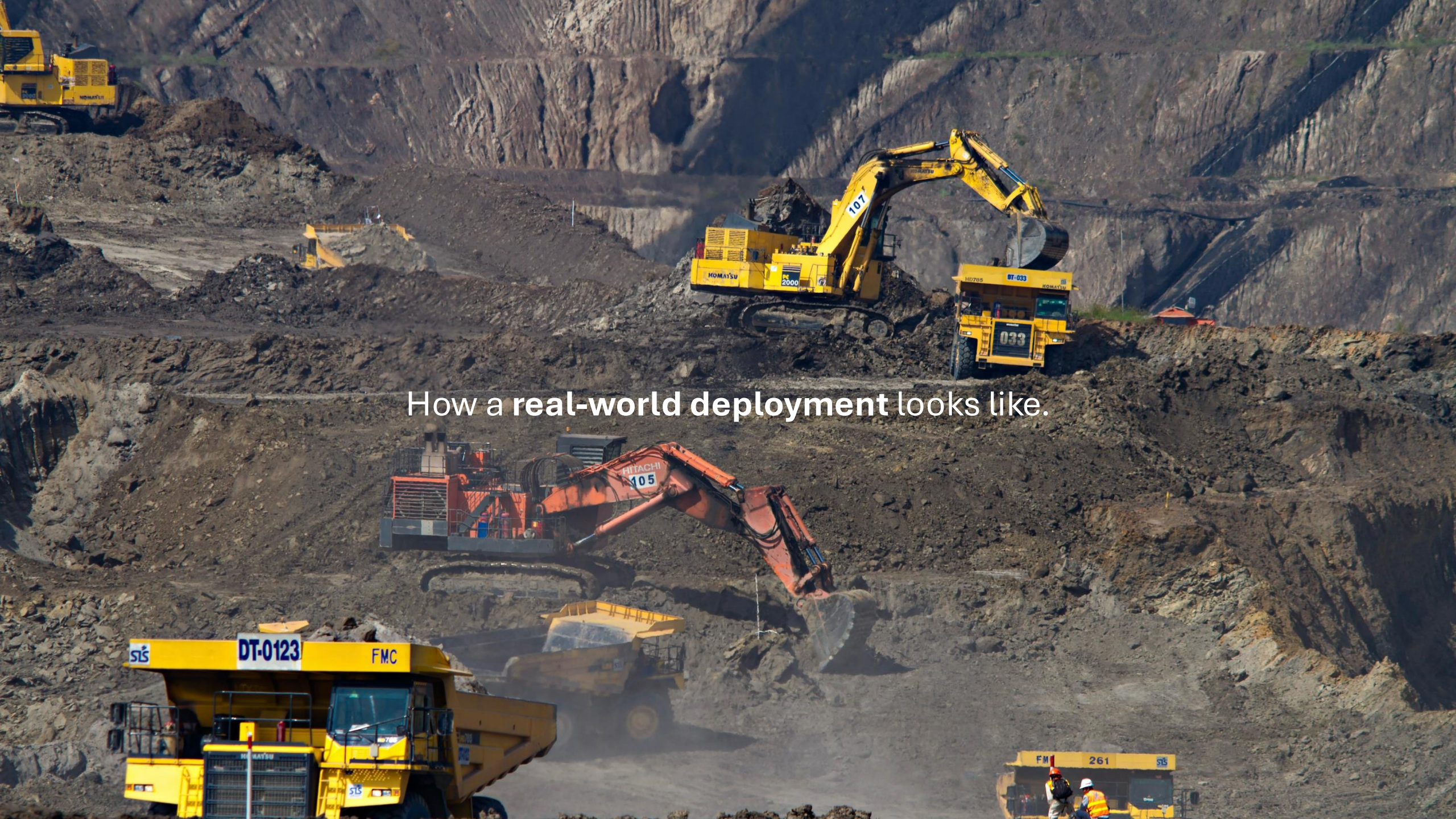
- IP addresses (route)
- Geo Info countries, region, city etc.
- Device types
- WAF information
  - Bot suspects
  - Rate limiting hits



Cognito


HTTP Request
w/ Auth. Header
Lambda Event
w/ identity
HTTP Response
Lambda Result
API Gateway
Lambda
Secret
Identity
Auth Flow
Cognito

How Serverless projects look like in **blogs** and **demos**.


How a **real-world deployment** looks like.
107
KOMATSU
PC 2000
HD785
DT-033
033
HITACHI
105
DT-0123
FMC
261

Vendor lock-in

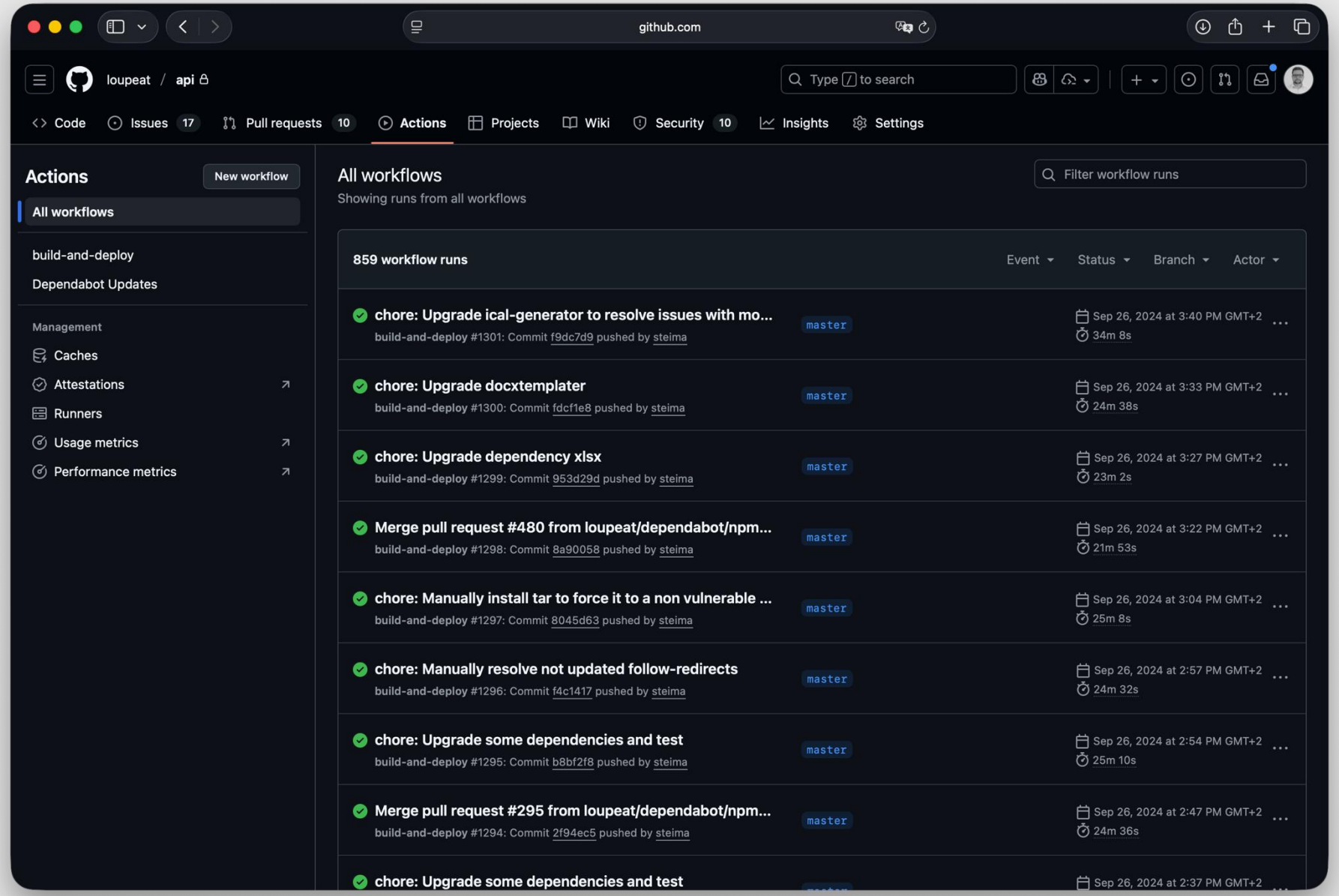
max. 500 Resources

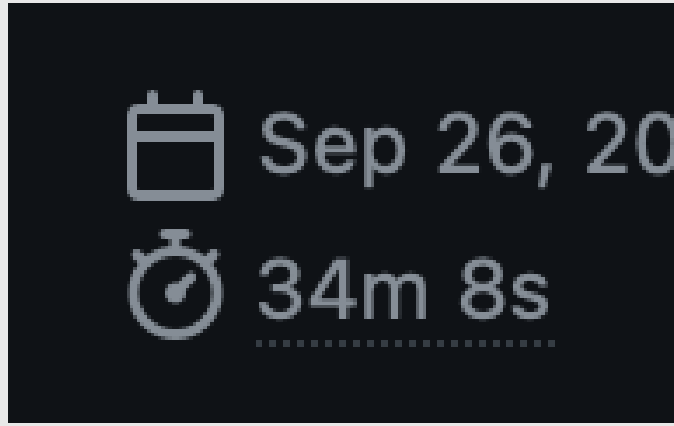
min. 6 R. / Function

~83 Serverless functions

https://aws.amazon.com/about-aws/whats-new/2020/10/aws-cloudformation-now-supports-increased-limits-on-five-service-quotas/

Slow deployments





Bad developer experience

- Functions cannot be simply run locally
  - Hard to debug
  - Requires slow deployment for tests

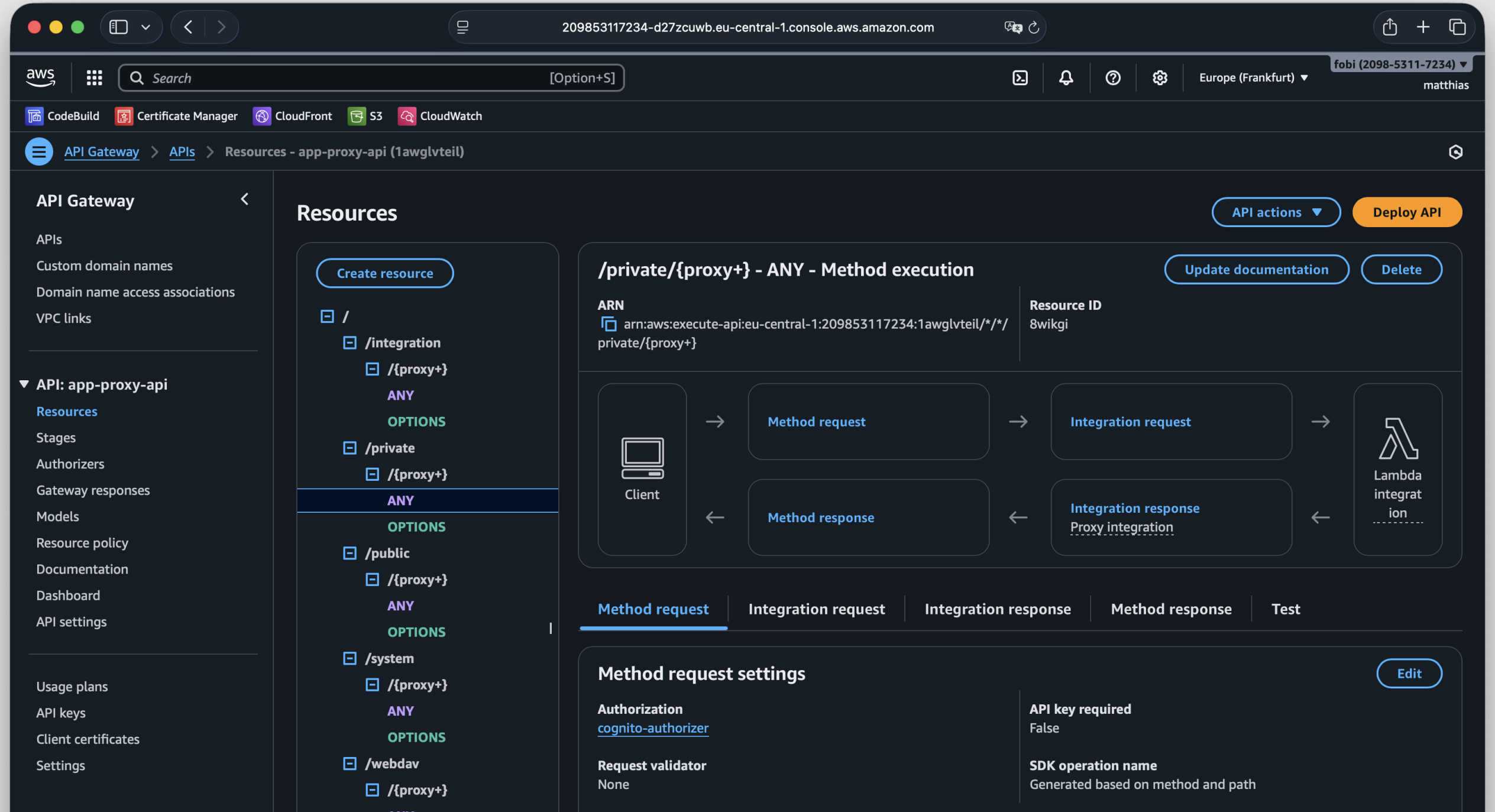
Cold starts

- When a **Lambda** is **not yet loaded** into a Firecracker VM
- Adds anywhere between 600ms and a few seconds onto the request

Enter the LambdaLith

Loupe
focused on business integrity

- A **Lambda** function that is a **monolith**
- Path **routing inside** of **function code** instead of API gateway
- All the advantages of Serverless but with
  - Quicker deployments
  - Better developer productivity
  - Reduced vendor lock-in

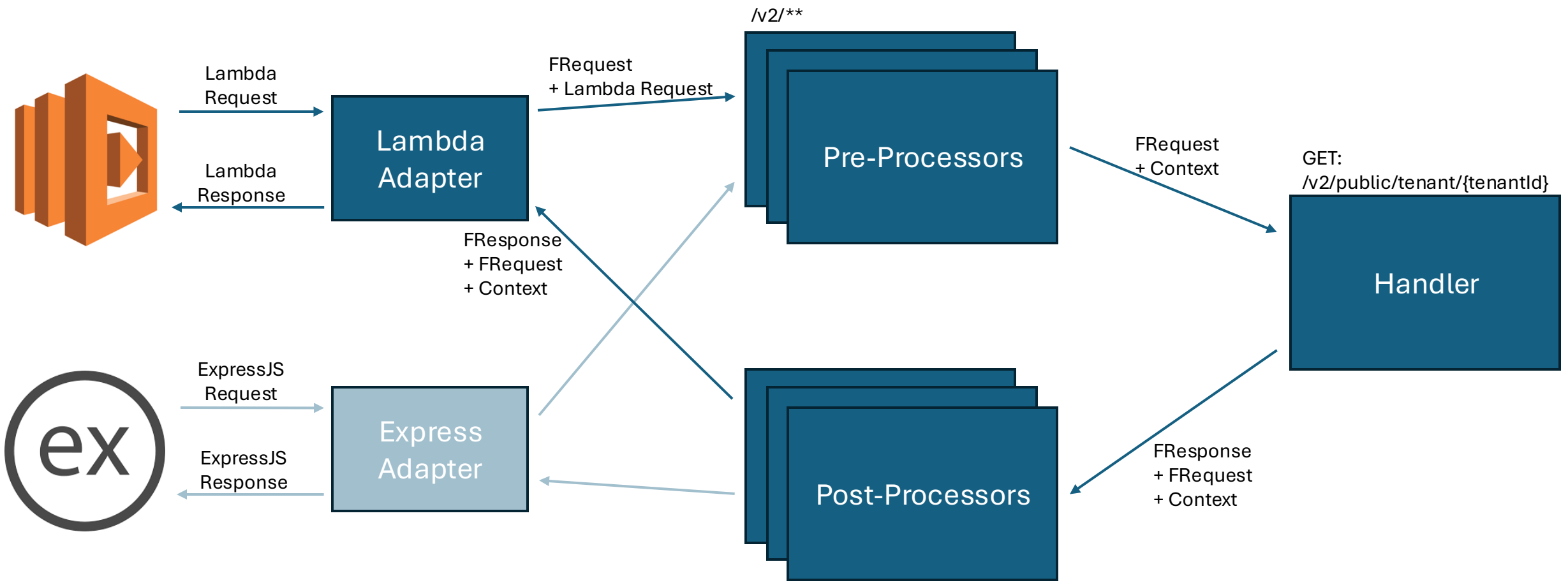

209853117234-d27zcuwb.eu-central-1.console.aws.amazon.com
aws
Search
[Option+S]
Europe (Frankfurt)
fobi (2098-5311-7234)
matthias
CodeBuild
Certificate Manager
CloudFront
S3
CloudWatch
API Gateway
APIs
Resources - app-proxy-api (1awglvteil)
API Gateway
APIs
Custom domain names
Domain name access associations
VPC links
API: app-proxy-api
Resources
Stages
Authorizers
Gateway responses
Models
Resource policy
Documentation
Dashboard
API settings
Usage plans
API keys
Client certificates
Settings
Resources
API actions
Deploy API
Create resource
/
/integration
/{proxy+}
ANY
OPTIONS
/private
/{proxy+}
ANY
OPTIONS
/public
/{proxy+}
ANY
OPTIONS
/system
/{proxy+}
ANY
OPTIONS
/webdav
/{proxy+}
ANY
/private/{proxy+} - ANY - Method execution
Update documentation
Delete
ARN
arn:aws:execute-api:eu-central-1:209853117234:1awglvteil/*/*/private/{proxy+}
Resource ID
8wikgi
Client
Method request
Integration request
Lambda integration
Method response
Integration response
Proxy integration
Method request
Integration request
Integration response
Method response
Test
Method request settings
Edit
Authorization
cognito-authorizer
API key required
False
Request validator
None
SDK operation name
Generated based on method and path

Meet FMiddleware

Loupe
focused on
business integrity

- **Takes inspiration** from other web-frameworks such as Spring
- At the core it is a **REST router** that can be run via ExpressJS or AWS Lambda or …
- Allows for request **pre-processing** and **post-processing**
- Pre- and Post-processing can be filtered by request path and runtime environment

FMiddleware Request Routing



Function handler

```
api.put(buildPath("settings"), async
  (request: FRequest<any, NotificationSettings>) => {
    assertCompanyAccess(api, request);
    const auth = api.context<Auth>(request, "auth");
    const updatedSettings = await
    SetNotificationSettingsUseCase.setNotificationSettings(
        auth,
        request.body
    );
    return api.responses.OK(request, updatedSettings);
}, NotificationSettingsSchema);
```

Pre-processor

```
export const CountryPreProcessorAWSLambda: RequestPreProcessor = {
    name: ".LOUPE viewer country pre-processor for AWS Lambda",
    pathPatterns: [
        "/v2/**"
    ],
    requestSource: "aws-lambda",
    process: async (_api, request, _handler) => {
        const viewerCountry = request.headers["cloudfront-viewer-country"];
        request.context["country"] = viewerCountry;
    }
};

export const CountryPreProcessorExpressLocal: RequestPreProcessor = {
    name: ".LOUPE set a fixed country for express local dev",
    pathPatterns: [
        "/v2/**"
    ],
    requestSource: "express",
    process: async (_api, request, _handler) => {
```

Results

- Stack can be fully run locally
  - Convenient testing and debugging
  - Prepared for execution in Docker
- Very quick AWS deployments 5-6 minutes
- Reduced cold starts
  - Usually, 3-4 warm instances up and running
  - High instance re-use makes caches more relevant

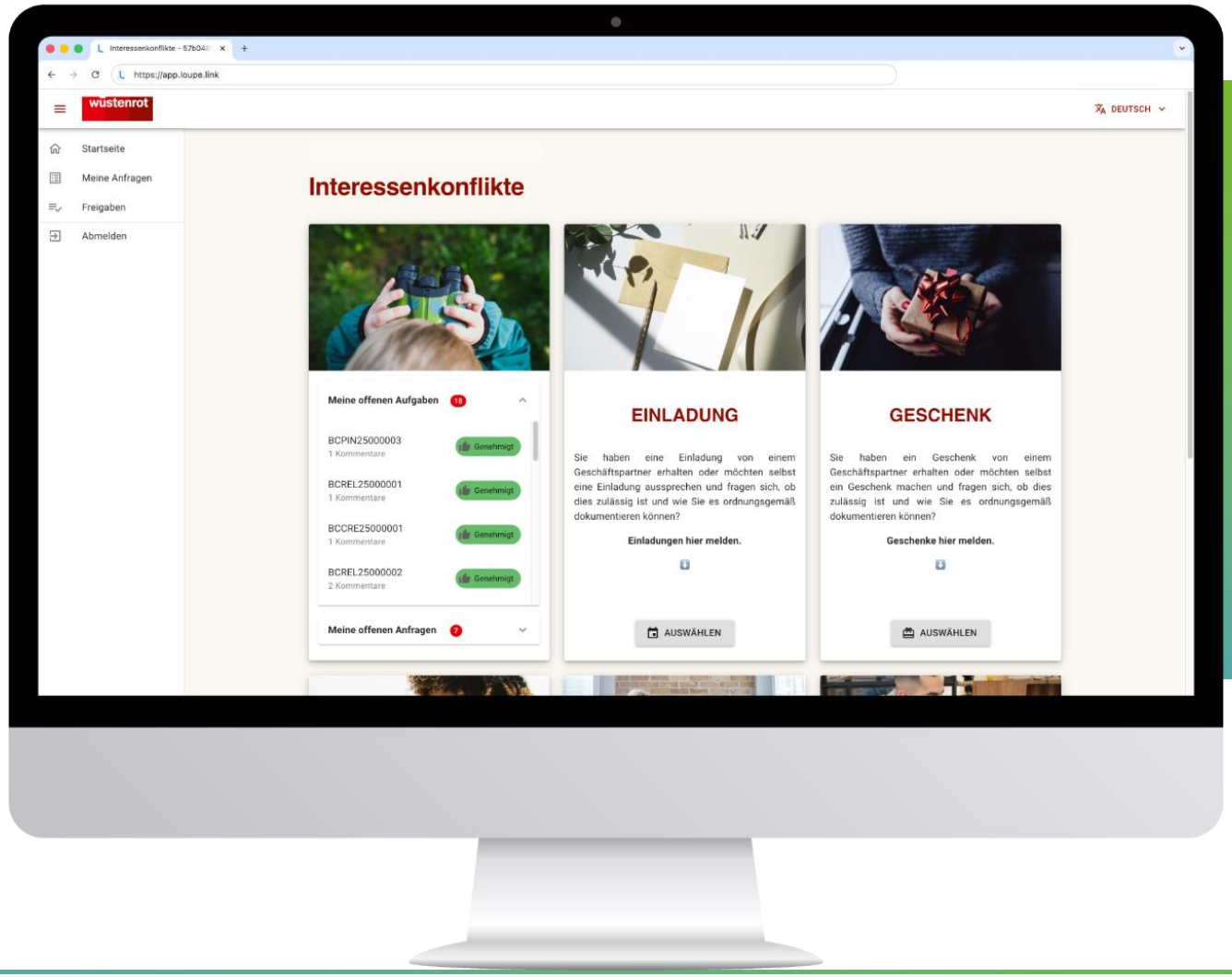
Outlook

- Refactor FMiddleware into a **TypeScript** / JavaScript **library**
- Support **Docker deployment** for on-premises installations
- OpenAPI spec generation for more advanced use-cases

Q&A

All stock images Creative Commons licensed via unsplash.com or directly licensed via iStockphoto.com

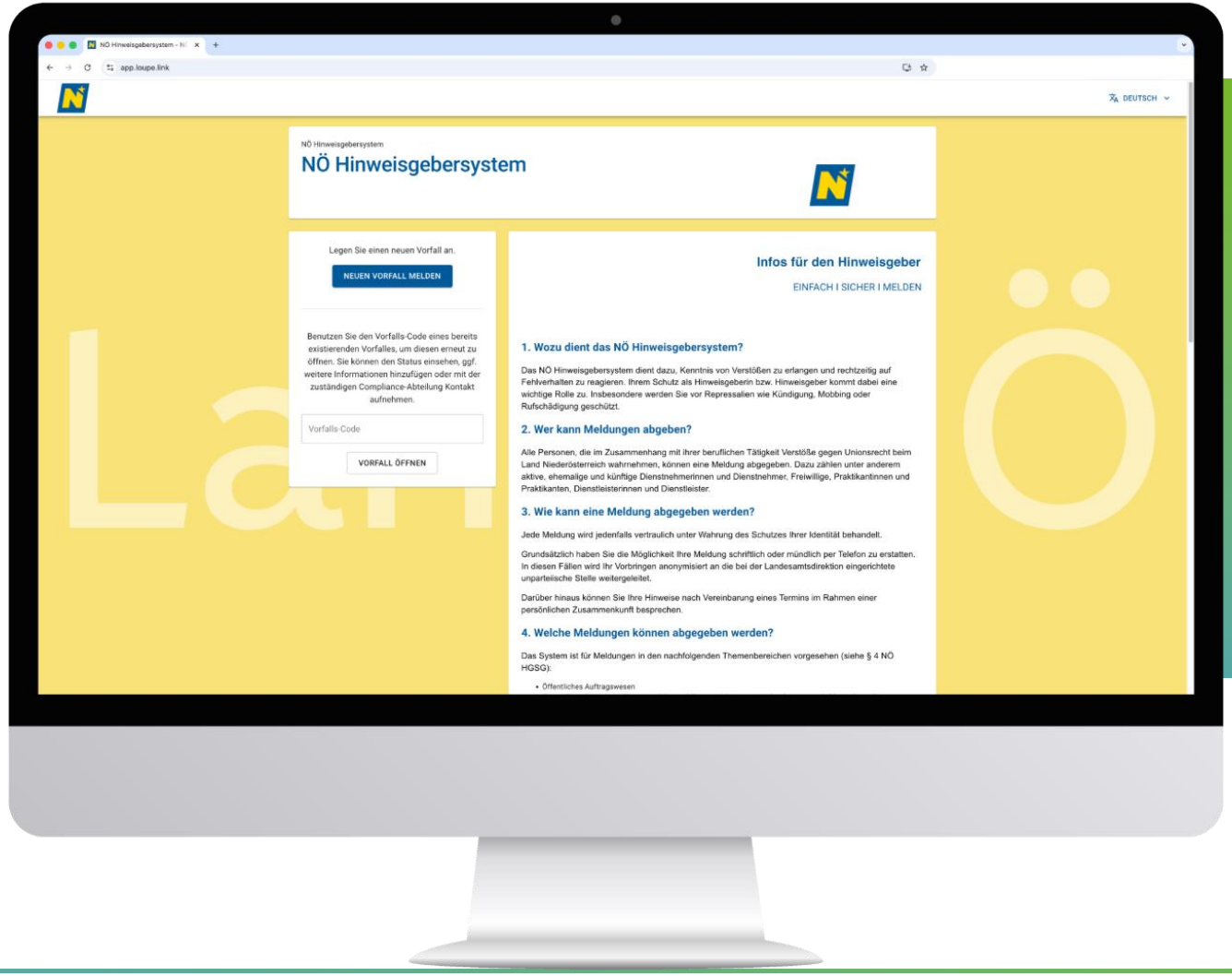
Integrity Check



Auf der Startseite können Mitarbeitende schnell die passende Kategorie auswählen und eine Meldung an Compliance übermitteln.

Die verfügbaren Kategorien lassen sich individuell an die Anforderungen Ihres Unternehmens anpassen.

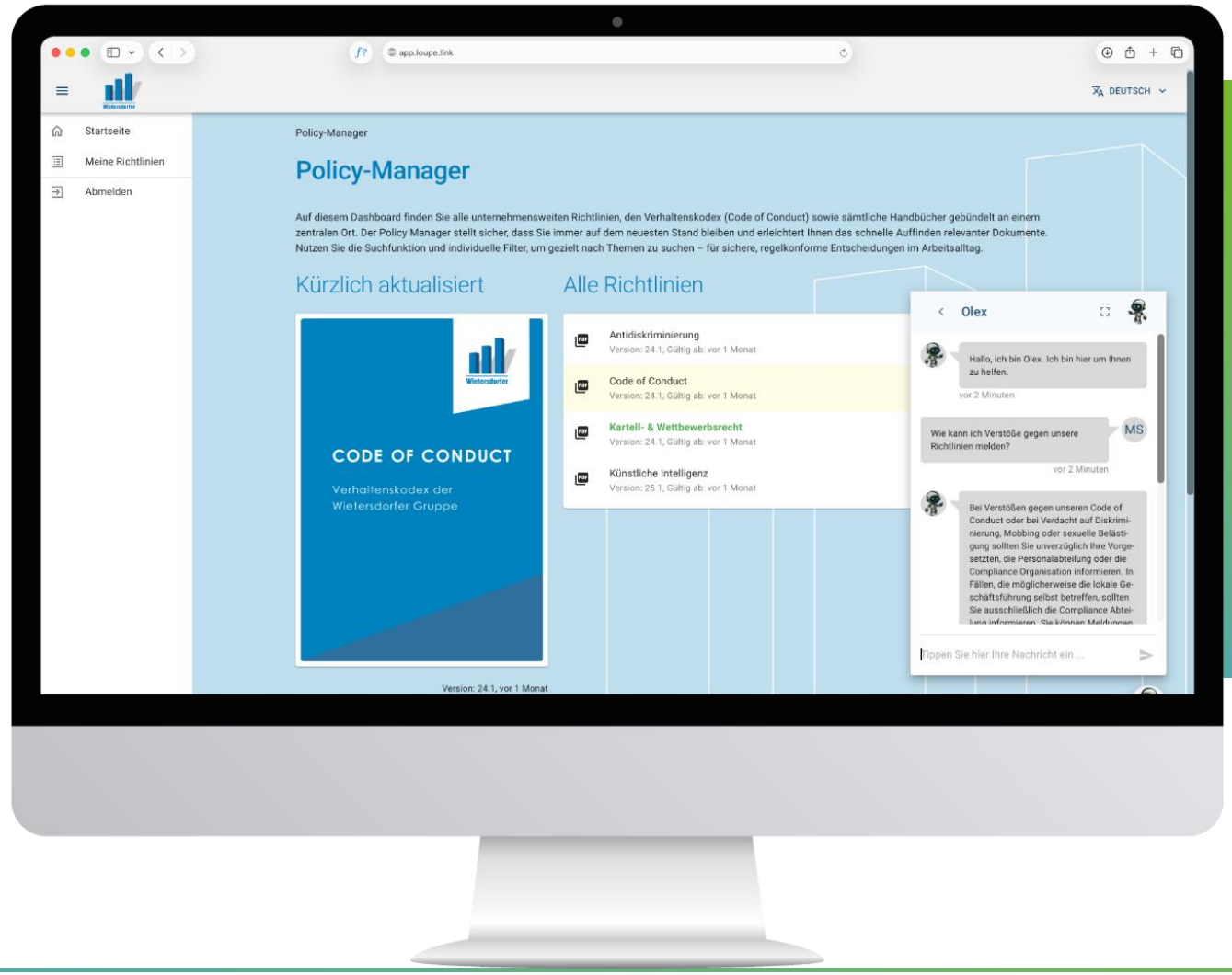
Whistleblowing



Auf der Startseite können Hinweisgeber:innen einen neuen Vorfall melden oder mit einem Vorfallscode eine bestehende Meldung wieder aufrufen.

Die angezeigten Informationen lassen sich individuell gestalten und an die Bedürfnisse Ihrer Organisation anpassen.

Policy Navigator



Die Richtlinienübersicht zeigt alle gültigen Dokumente sowie kürzlich aktualisierte Versionen.

Zusätzlich können auch Schulungsvideos oder kurze Learning Nuggets, die Inhalte aus den Richtlinien vermitteln, eingebunden werden.

Happy Customers

Loupe
focused on business integrity

Einige von ihnen ...

STRABAG

wustenrot

iSi GROUP

WIENER STÄDTISCHE
VIENNA INSURANCE GROUP

Silhouette®

nöm

AUVA

Wietersdorfer

Lagermax

LOACKER

EVVA
access to security

TGW
LOGISTICS

SOS KINDERDORF

Haubis
In der Familie. Seit 1902.

Compliance gibt’s (nicht) im Abo.

Integrität ist nicht verhandelbar.

Menschenrechtsverletzungen sind kein Business-Case.