

Serverless in der AWS

Einstieg in die Serverless-Entwicklung auf Amazon Web Services



htrinter

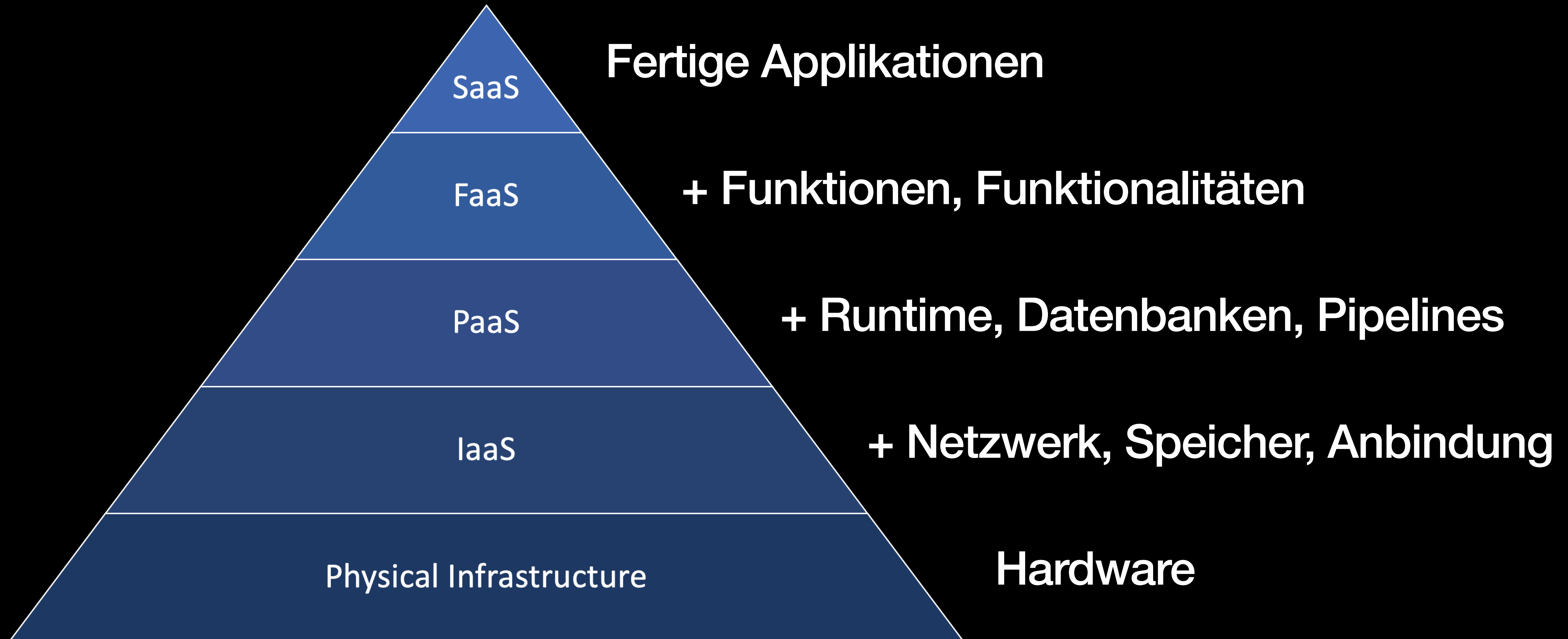
Wer hat "serverless" schonmal gehört?

Wer weiß, was serverless ist?

Wer hat schonmal eine Serverless-Anwendung gebaut?

Was ist serverless?

Everything as a Service



serverless



SaaS

Fertige Applikationen

FaaS

+ Funktionen, Funktionalitäten

PaaS

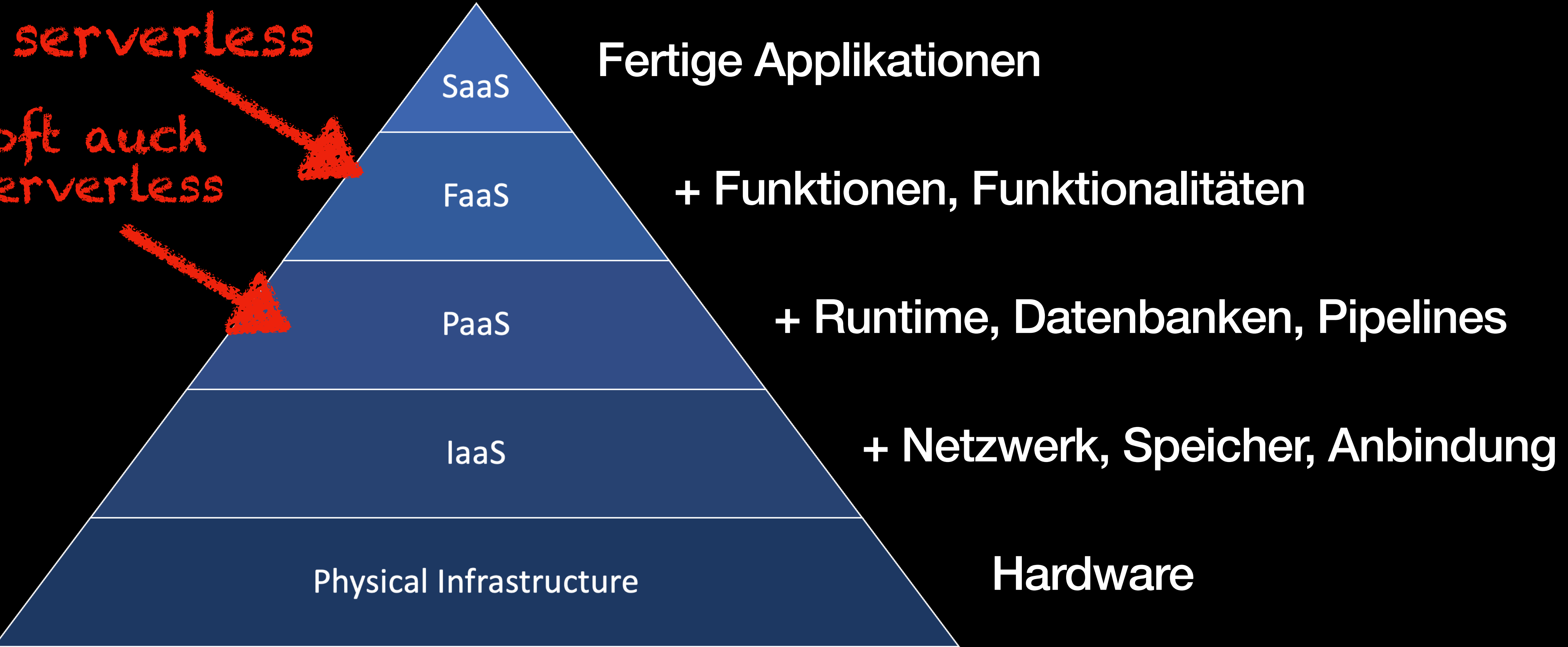
+ Runtime, Datenbanken, Pipelines

IaaS

+ Netzwerk, Speicher, Anbindung

Physical Infrastructure

Hardware



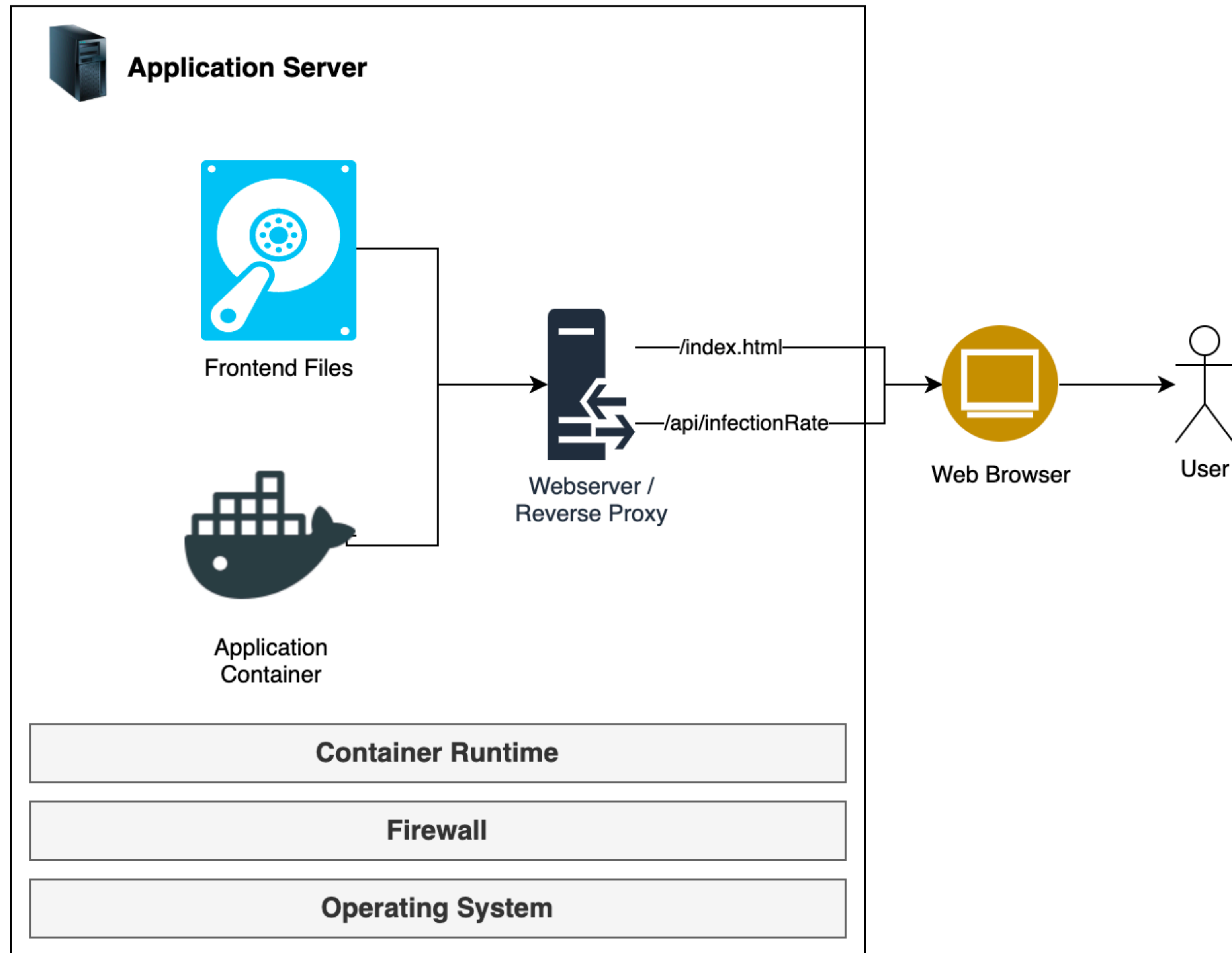
Ausgangsprojekt: allexminuten.de

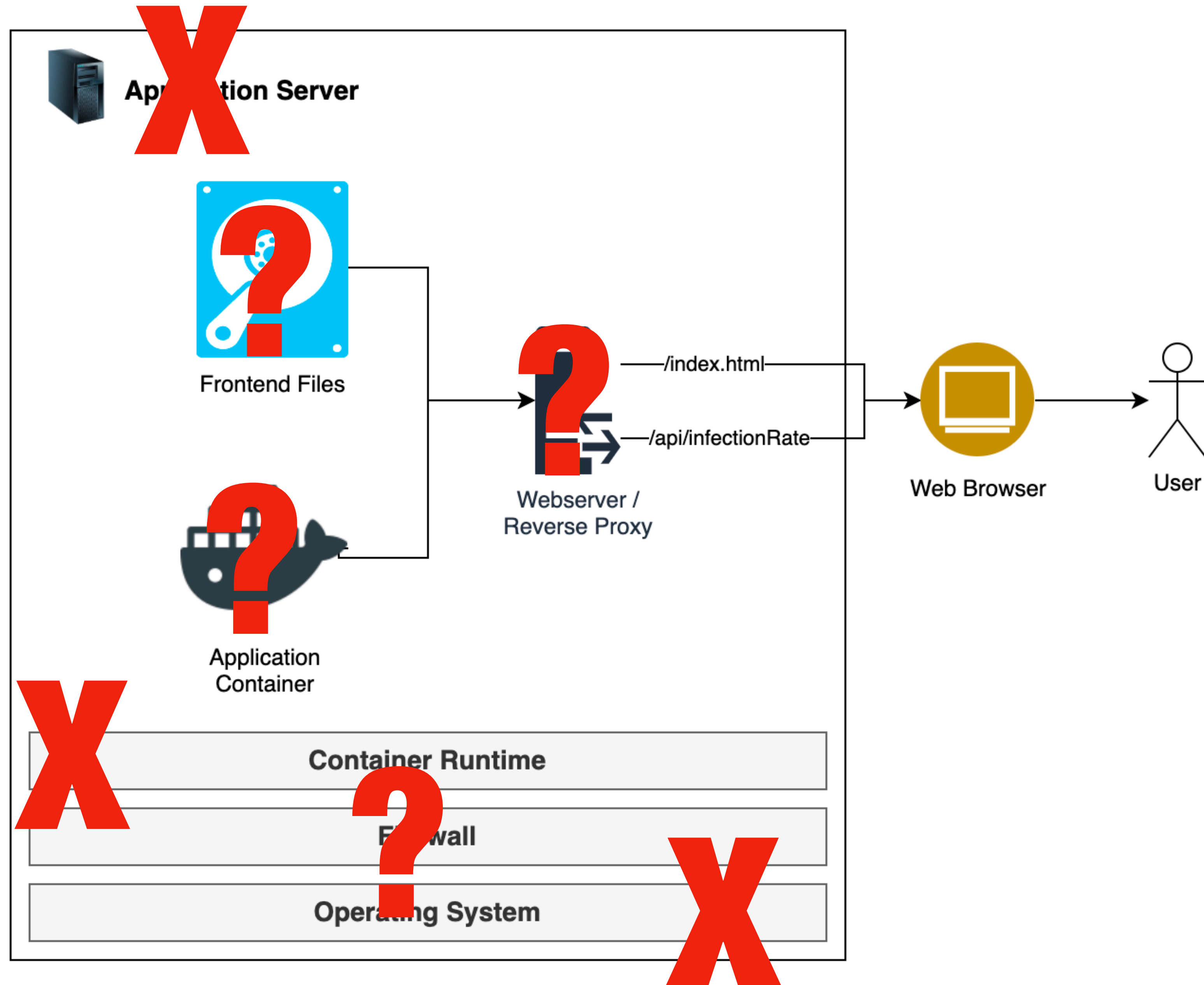
github.com/finreinhard/alleXminuten



finreinhard

**Anforderung:
Hosting + Implementierung einer API**





**Dateien
speichern**



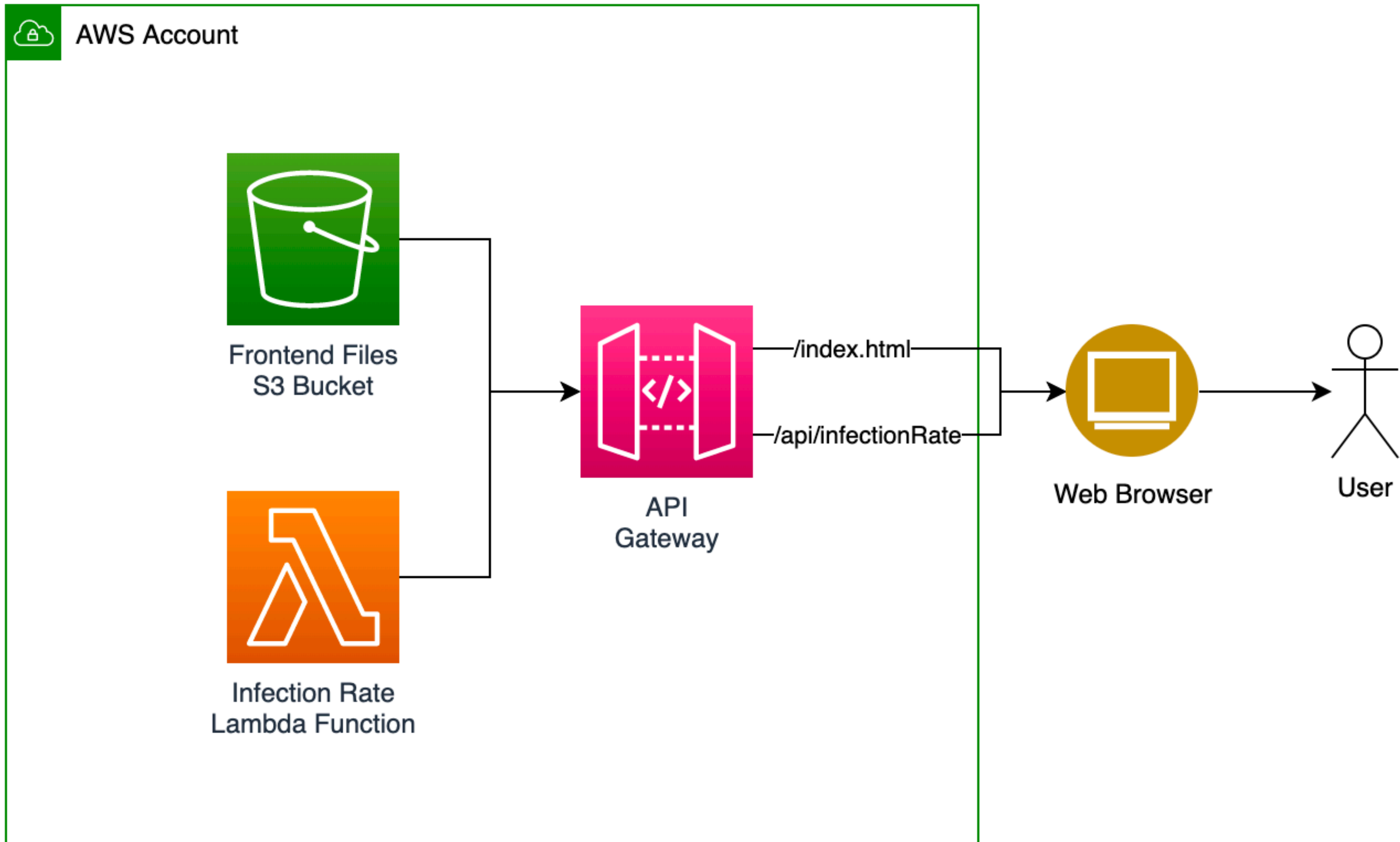
**Code
ausführen**



**Dateien + API
ausliefern**



Services to the rescue!



Demo: AWS Management Console

Infrastructure as Code mit *AWS*

CloudFormation Templates

```
1 Resources:
2   CdksampleQueue3139C8CD:
3     Type: AWS::SQS::Queue
4     Properties:
5       VisibilityTimeout: 300
6       UpdateReplacePolicy: Delete
7       DeletionPolicy: Delete
8     Metadata:
9       aws:cdk:path: CdksampleStack/CdksampleQueue/Resource
10  CdksampleQueuePolicyFA91005A:
11    Type: AWS::SQS::QueuePolicy
12    Properties:
13      PolicyDocument:
14        Statement:
15          - Action: sqs:SendMessage
16            Condition:
17              ArnEquals:
18                aws:SourceArn:
19                  Ref: CdksampleTopic7AD235A4
20            Effect: Allow
21            Principal:
22              Service: sns.amazonaws.com
23            Resource:
24              Fn::GetAtt:
25                - CdksampleQueue3139C8CD
26                - Arn
27            Version: "2012-10-17"
28      Queues:
29        - Ref: CdksampleQueue3139C8CD
30    Metadata:
31      aws:cdk:path: CdksampleStack/CdksampleQueue/Policy/Resource
32  CdksampleQueueCdksampleStackCdksampleTopicCB3FDFDDC0BCF47C:
33    Type: AWS::SNS::Subscription
34    Properties:
35      Protocol: sqs
36      TopicArn:
37        Ref: CdksampleTopic7AD235A4
38      Endpoint:
39        Fn::GetAtt:
40          - CdksampleQueue3139C8CD
41          - Arn
42    Metadata:
43      aws:cdk:path: CdksampleStack/CdksampleQueue/CdksampleStackCdksampleTopi
44  CdksampleTopic7AD235A4:
```

- Definition der Infrastruktur als YAML oder JSON
- Reproduzierbare und versionierbare Infrastruktur
- Anwendung der Definition auf CloudFormation
- Organisation von Ressourcen in CloudFormation in "Stacks"

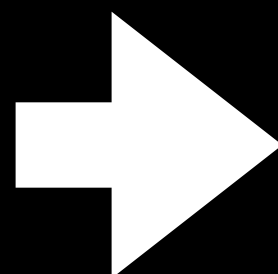
Cloud Development Kit

```
1 import { Duration, Stack, StackProps } from 'aws-cdk-lib';
2 import * as sns from 'aws-cdk-lib/aws-sns';
3 import * as subs from 'aws-cdk-lib/aws-sns-subscriptions';
4 import * as sqs from 'aws-cdk-lib/aws-sqs';
5 import { Construct } from 'constructs';
6
7 export class CdksampleStack extends Stack {
8   constructor(scope: Construct, id: string, props?: StackProps) {
9     super(scope, id, props);
10
11     const queue = new sqs.Queue(this, 'CdksampleQueue', {
12       visibilityTimeout: Duration.seconds(300)
13     });
14
15     const topic = new sns.Topic(this, 'CdksampleTopic');
16
17     topic.addSubscription(new subs.SqsSubscription(queue));
18   }
19 }
20
```

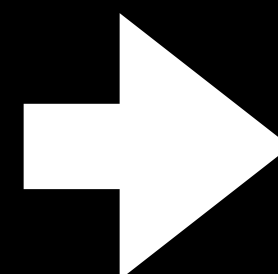
- Definition der Infrastruktur als Code in TypeScript, Python, .NET oder Java
- AWS-Ressourcen als Objekte
- Vorteile einer IDE und Code Inspection
- CDK enthält viele Vorlagen und Regeln, nach denen z.B. benötigte Rollen und Rechte automatisch angelegt werden => weniger Code
- Das Beste: Entwickeln mit dem CDK macht Spaß!

Deployment mit CDK

```
1 import { Duration, Stack, StackProps } from 'aws-cdk-lib';
2 import * as sns from 'aws-cdk-lib/aws-sns';
3 import * as subs from 'aws-cdk-lib/aws-sns-subscriptions';
4 import * as sqs from 'aws-cdk-lib/aws-sqs';
5 import { Construct } from 'constructs';
6
7 export class CdkSampleStack extends Stack {
8   constructor(scope: Construct, id: string, props?: StackProps) {
9     super(scope, id, props);
10
11     const queue = new sqs.Queue(this, 'CdkSampleQueue', {
12       visibilityTimeout: Duration.seconds(300)
13     });
14
15     const topic = new sns.Topic(this, 'CdkSampleTopic');
16
17     topic.addSubscription(new subs.SqsSubscription(queue));
18   }
19 }
20
```



```
1 Resources:
2   CdkSampleQueue3139C8CD:
3     Type: AWS::SQS::Queue
4     Properties:
5       VisibilityTimeout: 300
6       UpdateReplacePolicy: Delete
7       DeletionPolicy: Delete
8     Metadata:
9       aws:cdk:path: CdkSampleStack/CdkSampleQueue/Resource
10   CdkSampleQueuePolicyFA91005A:
11     Type: AWS::SQS::QueuePolicy
12     Properties:
13       PolicyDocument:
14         Statement:
15           - Action: sqs:SendMessage
16             Condition:
17               ArnEquals:
18                 aws:SourceArn:
19                   Ref: CdkSampleTopic7AD235A4
20             Effect: Allow
21             Principal:
22               Service: sns.amazonaws.com
23             Resource:
24               Fn::GetAtt:
25                 - CdkSampleQueue3139C8CD
26                 - Arn
27             Version: "2012-10-17"
28       Queues:
29         - Ref: CdkSampleQueue3139C8CD
30     Metadata:
31       aws:cdk:path: CdkSampleStack/CdkSampleQueue/Policy/Resource
32   CdkSampleQueueCdkSampleStackCdkSampleTopicCB3FDFDDC0BCF47C:
33     Type: AWS::SNS::Subscription
34     Properties:
35       Protocol: sqs
36       TopicArn:
37         Ref: CdkSampleTopic7AD235A4
38       Endpoint:
39         Fn::GetAtt:
40           - CdkSampleQueue3139C8CD
41           - Arn
42     Metadata:
43       aws:cdk:path: CdkSampleStack/CdkSampleQueue/CdkSampleStackCdkSampleTopic7AD235A4:
44
```



Cloud Development Kit
Code

CloudFormation
Template

AWS Account

Demo: Umsetzung und Deployment des Projektes mit CDK

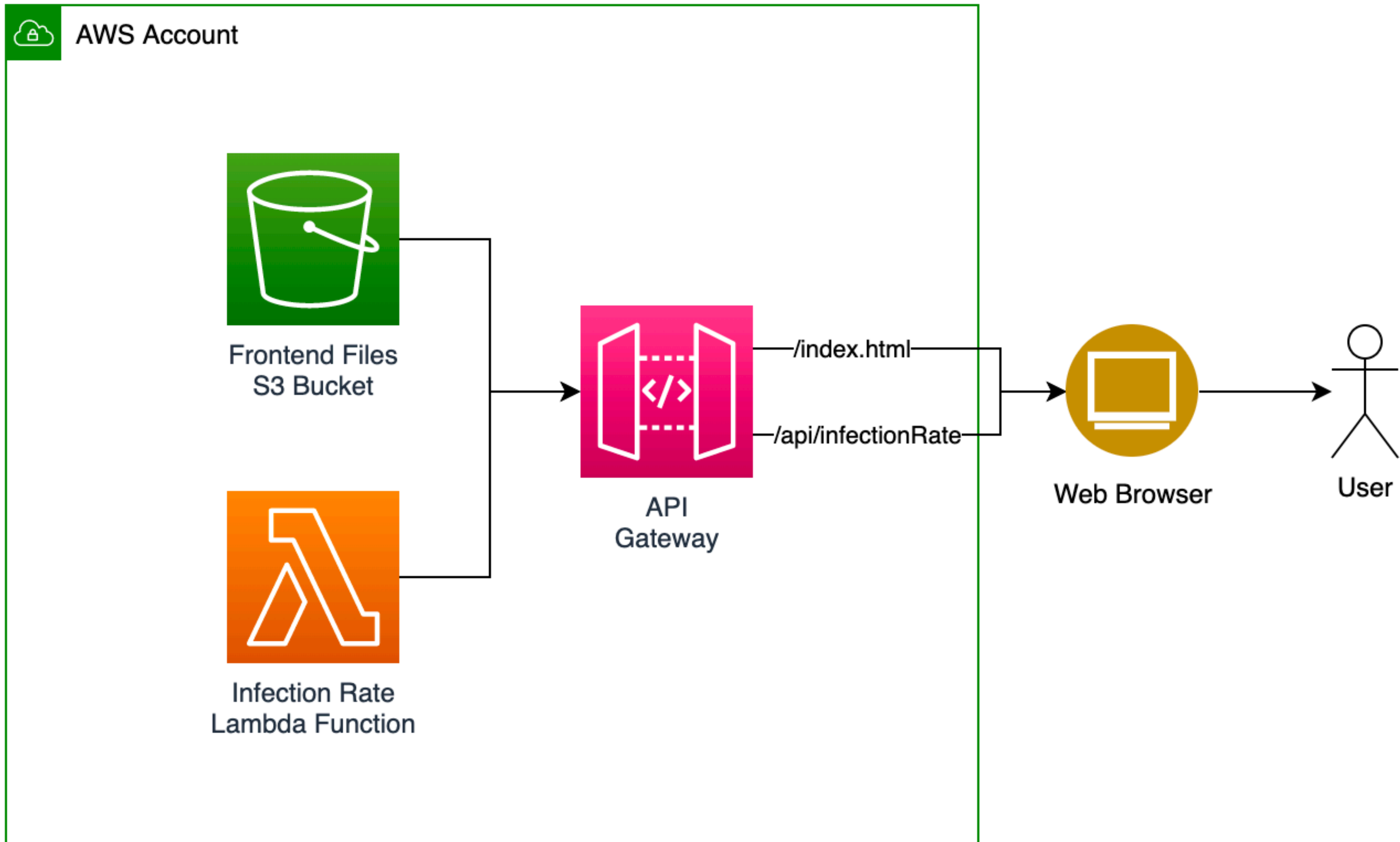
Projektstart mit CDK:

AWS-CLI installieren

```
npm install -g aws-cdk
```

```
cdk init sample-app --language typescript
```

```
cdk deploy
```



alleXminuten:
github.com/finreinhard/alleXminuten



finreinhard

alleXminuten in der AWS:
github.com/htrinter/alleXminuten-aws-serverless



htrinter

Empfehlung: CDK Workshop
<https://cdkworkshop.com/>

Zeit für Fragen!

+

- Geringe Aufwände in Betrieb und Wartung
- Hohe horizontale Skalierbarkeit
- Arbeitsergebnisse schnell online
- Sehr schnelle Provisionierung von Infrastruktur, auch im non-serverless Umfeld
- Per-use pricing

-

- Vendor lock-in auf Cloudanbieter
- Leben mit den Limitationen der Services
- Kompliziertes serverseitiges State-Handling
- Neue Workflows in Entwicklung, komplexeres Debugging
- Einarbeitungs- und Fortbildungsaufwand

alleXminuten:
github.com/finreinhard/alleXminuten



finreinhard

alleXminuten in der AWS:
github.com/htrinter/alleXminuten-aws-serverless



htrinter