

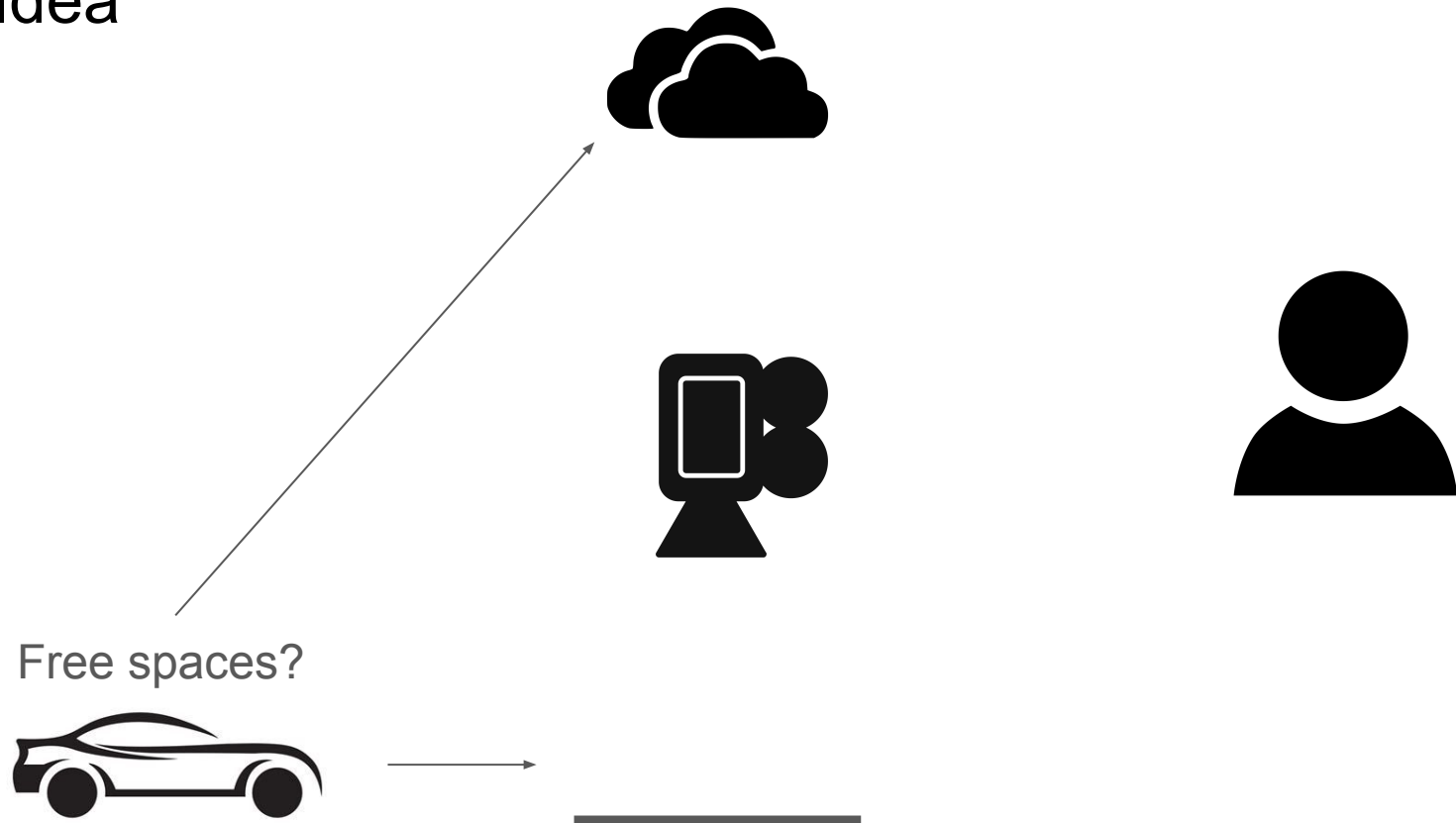
Alpine Parking

Matthias Ebner, Lukas Niederstätter & Paul Prünster

Structure

- Key Idea
- AWS Services
- Workflow
- Lambdas
- Frontend
- DEMO
- Costs

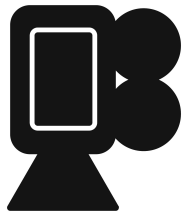
Key Idea



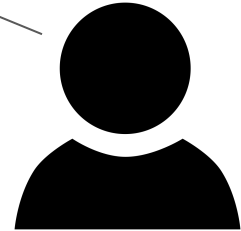
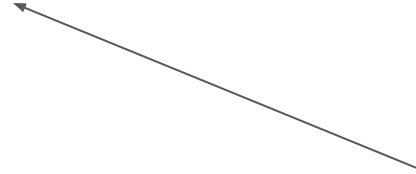
Key Idea



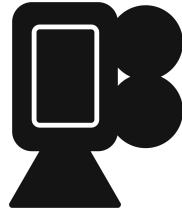
Update DB



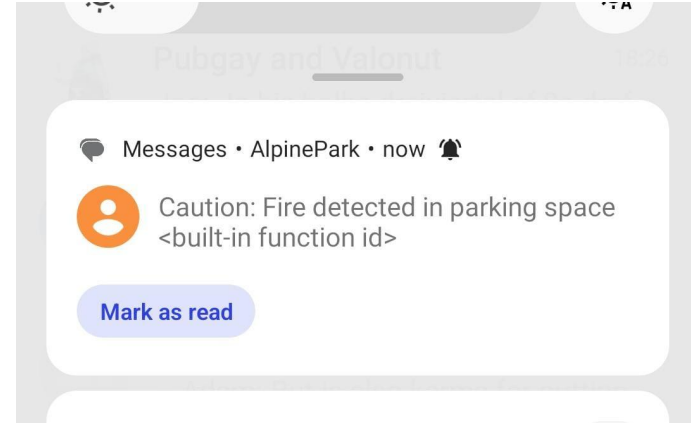
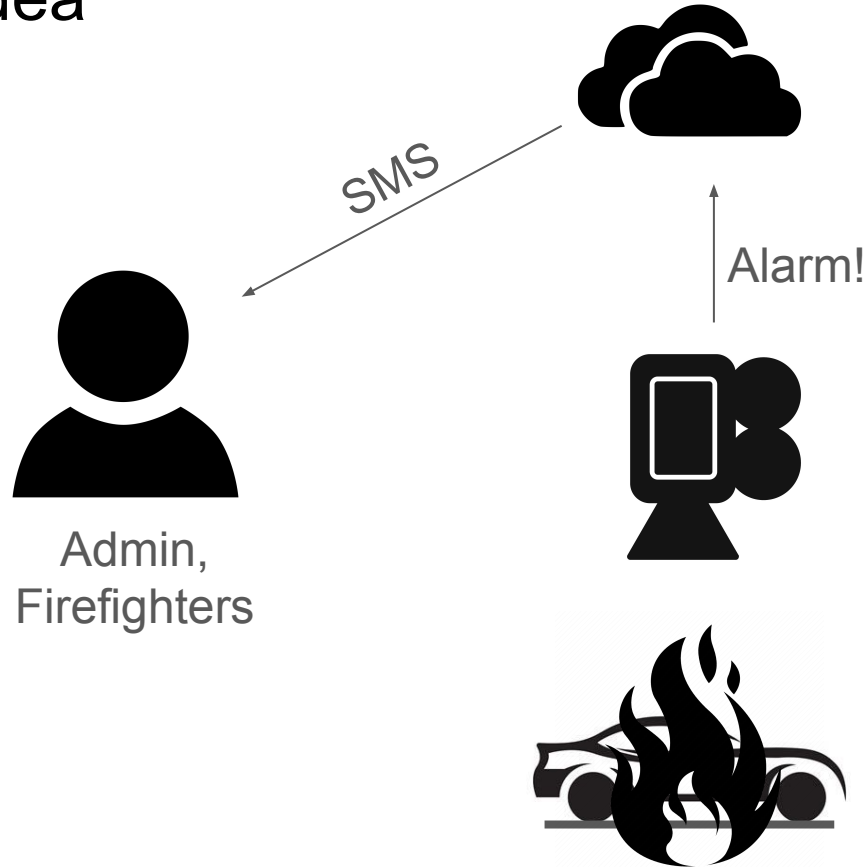
Key Idea



Free spaces?
Where is my car



Key Idea



AWS Services

Lambda Functions

Step functions

Api Gateway

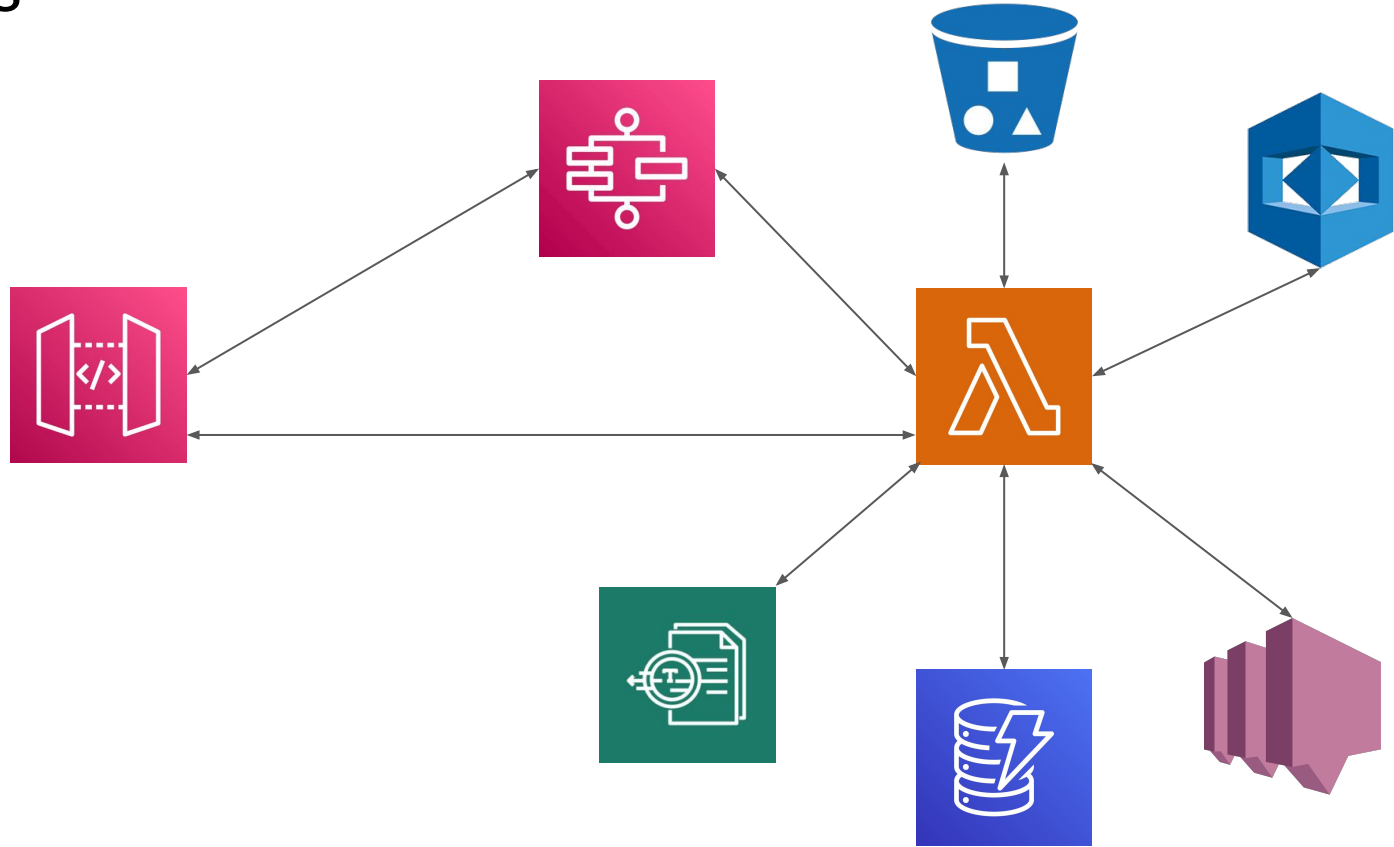
DynamoDB

S3 bucket

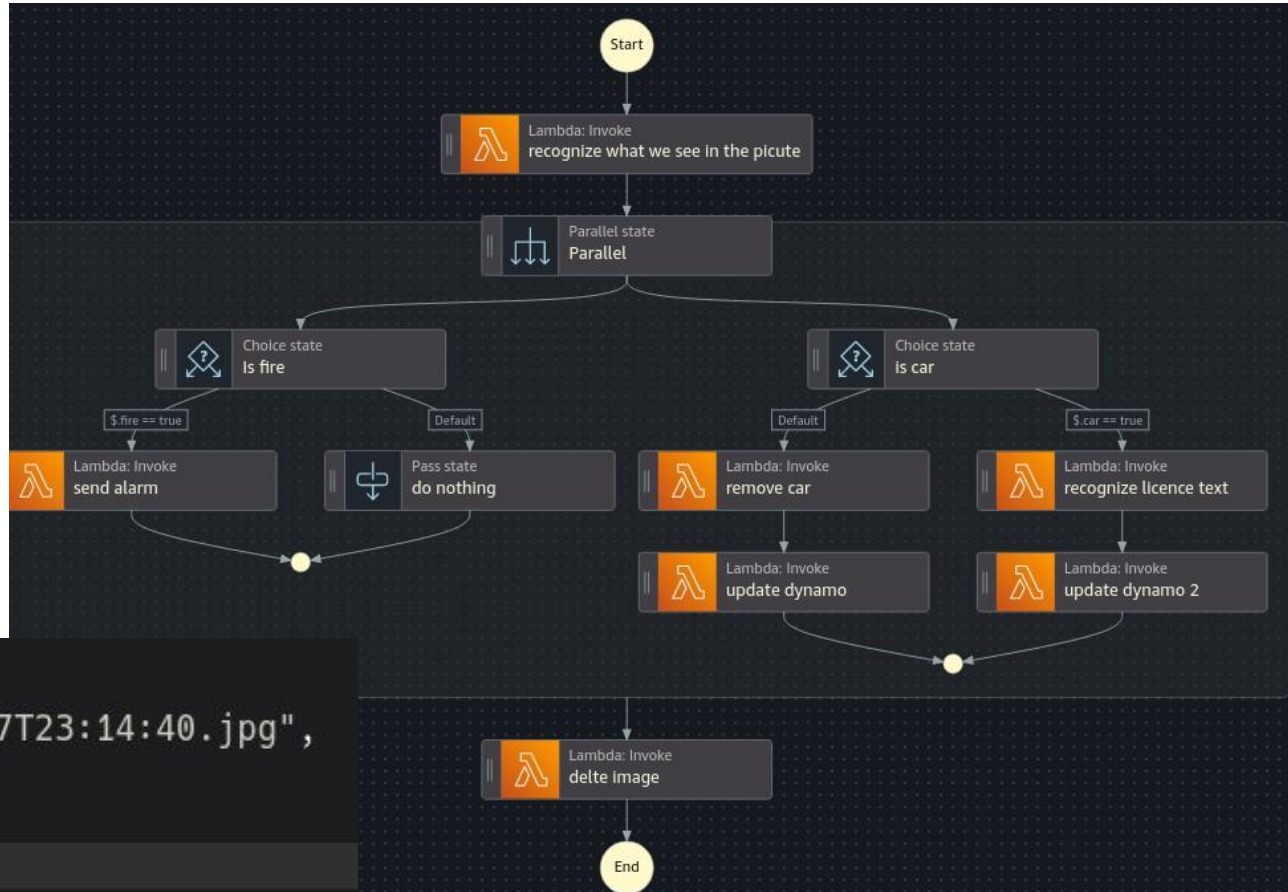
AWS Rekognition

Textract

SNS



Workflow



```
{  
  "image_key": "1_2024-01-17T23:14:40.jpg",  
  "parking_space_id": 1  
}
```


Lambdas

boto3 lib

nice integration of services

```
bucket = 'verteiltssystemeparkingsystem'  
fire_words = ['fire', 'flames']  
car_words = ['car', "license plate", "vehicle", "transportation"]
```

```
def lambda_handler(event, context):  
    s3 = boto3.client('s3')  
    rekognition = boto3.client('rekognition')  
  
    key = event['image_key']  
    parking_space_id = event['parking_space_id']  
  
    obj = s3.get_object(Bucket=bucket, Key=key)  
    image = obj['Body'].read()  
  
    response = rekognition.detect_labels(Image={'Bytes': image})  
  
    fire = False  
    car = False  
  
    labels = []  
    for label in response['Labels']: ...  
  
    output = {  
        "image_key": key,  
        "parking_space_id": parking_space_id,  
        "car": car,  
        "fire": fire,  
        "rekognition_result": labels  
    }  
    return output
```

Frontend

- AngularJS Hello World like 200 mb
=> let's learn a new lightweight framework
- Alpine JS
- Material Design Lite



```
<html>
  <head>
    ...

    <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x.x/dist/cdn.min.js"></script>
  </head>
  ...
</html>
```



```
<div x-data="{isOpen : false}">
  <button @click="isOpen=!isOpen">Show/Hide</button>
  <h1 x-show="isOpen">Alpine Tutorial</h1>
</div>
```

```
<div x-data="{ inputValue : '' }" >
  <input type="text" x-model=inputValue />
  <p x-text="inputValue"></p>
</div>
```

LIVE DEMO

Transactional text message delivery rate

46.15%

Transactional text messages



Costs: car in parking spot

- Assumptions: 1 Parking space
 - Critical Path: \$0,001505851 per Execution
 - \$1,505851 per 1000 Executions
-
- Assumption: 100 Parking spaces
 - \$150,5851 per 1000 Executions
 - Optimization is possible:
 - More parking spaces per camera
 - Motion detection per parking space



Costs: case of fire

- Assumptions: 1 Parking space
 - Critical Path: \$0,000406351 per Execution
 - \$0,406351 per 1000 Executions
-
- Assumption: 100 Parking spaces
 - \$40,6351 per 1000 Executions
 - Optimization is possible:
 - More parking spaces per camera
 - Motion detection per parking space



Thank you for your attention