serverless IoT-Applications

BED-Con 2017

Niko Will, innoQ

@n1ko_w1ll

about me

- Developer since 2005
- living in a Smarthome since 2012
  - became an IoT Geek
- before: worked on Bosch IoT Suite for 2 years
- now: Consultant at innoQ
- follow me on Twitter: @n1ko_w1ll

agenda

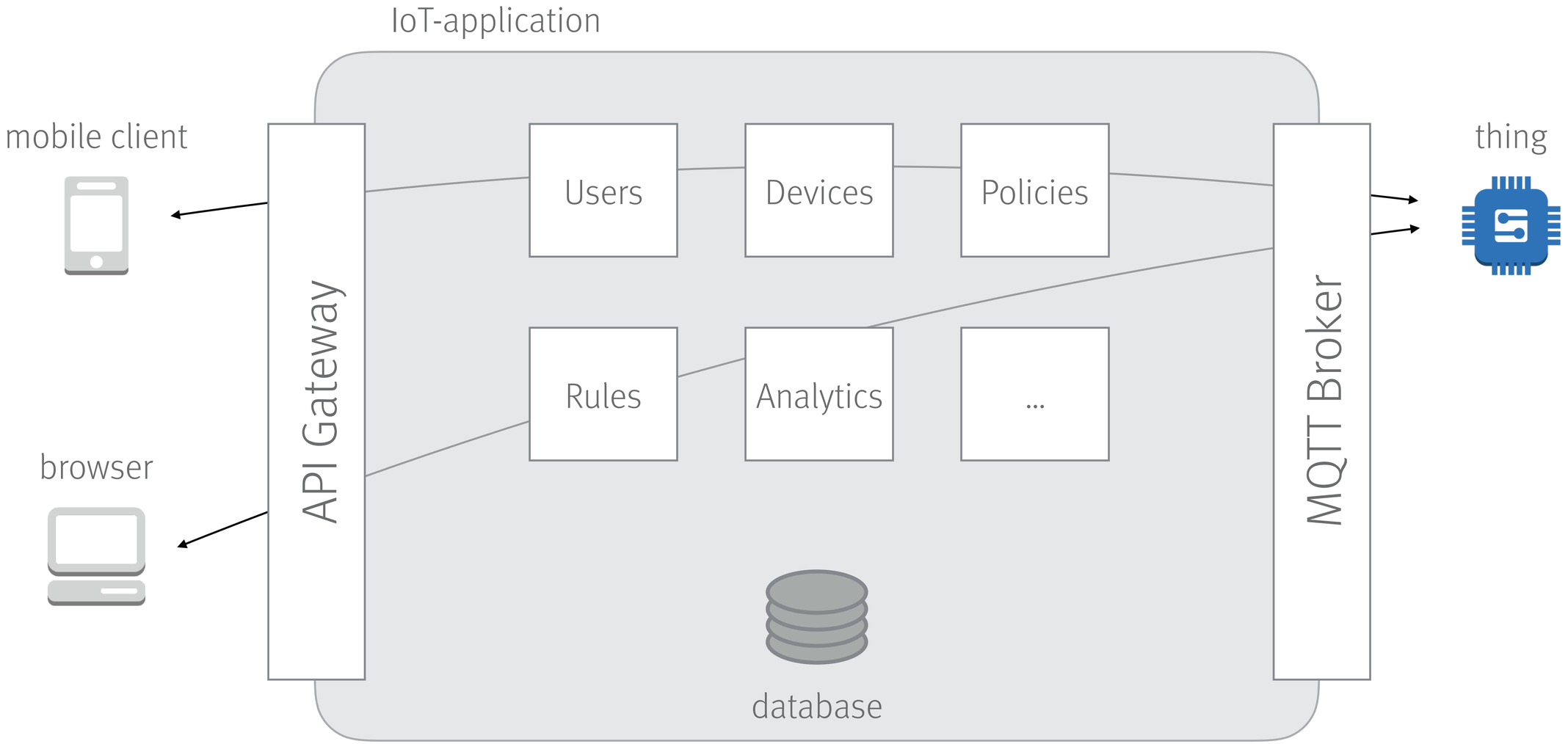
- microservices approach
- AWS, Lambda & IoT
- use-cases
  - JITR | on-boarding | pairing
  - list / search things | command & control | telemetry
  - connected / disconnected / LWT
  - encrypted file transfer | firmware update

microservice approach

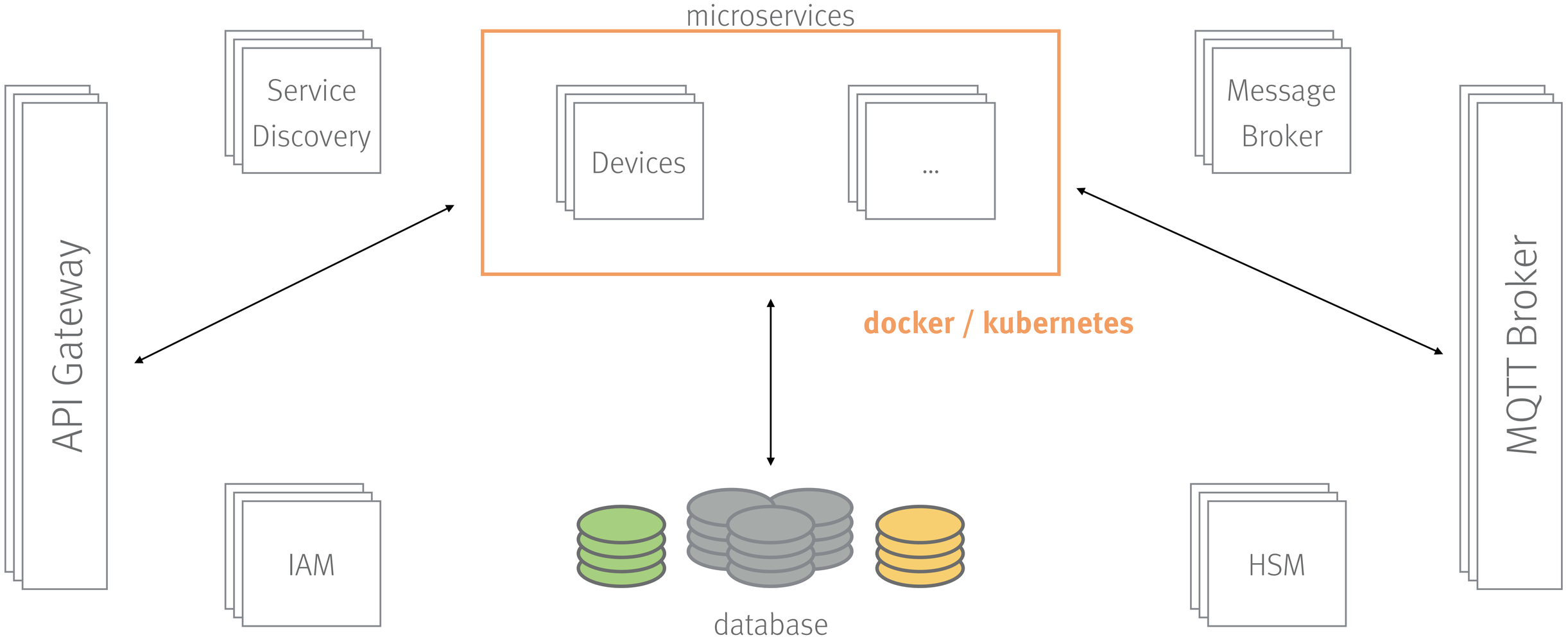
microservices approach

- fault tolerance
- scalability
- agility
- visibility
- security
- cost-efficiency

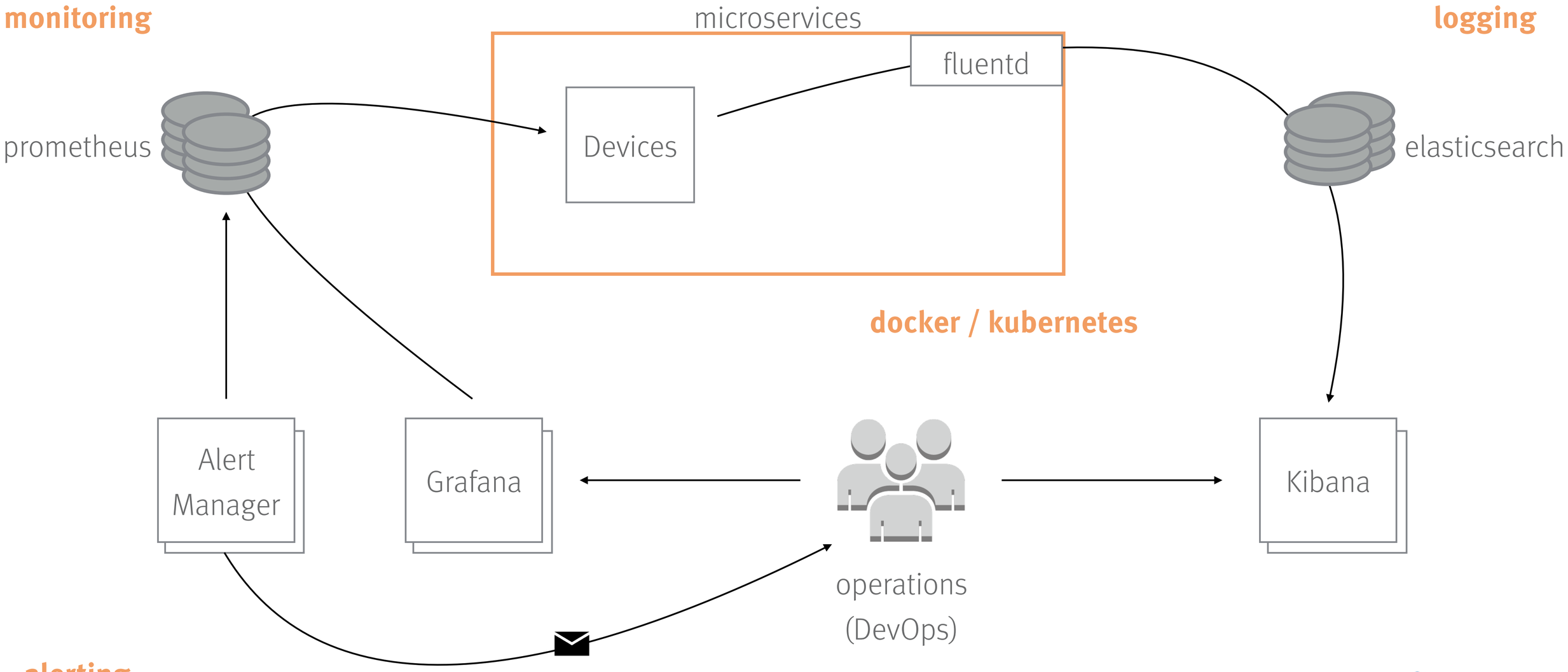
microservices approach

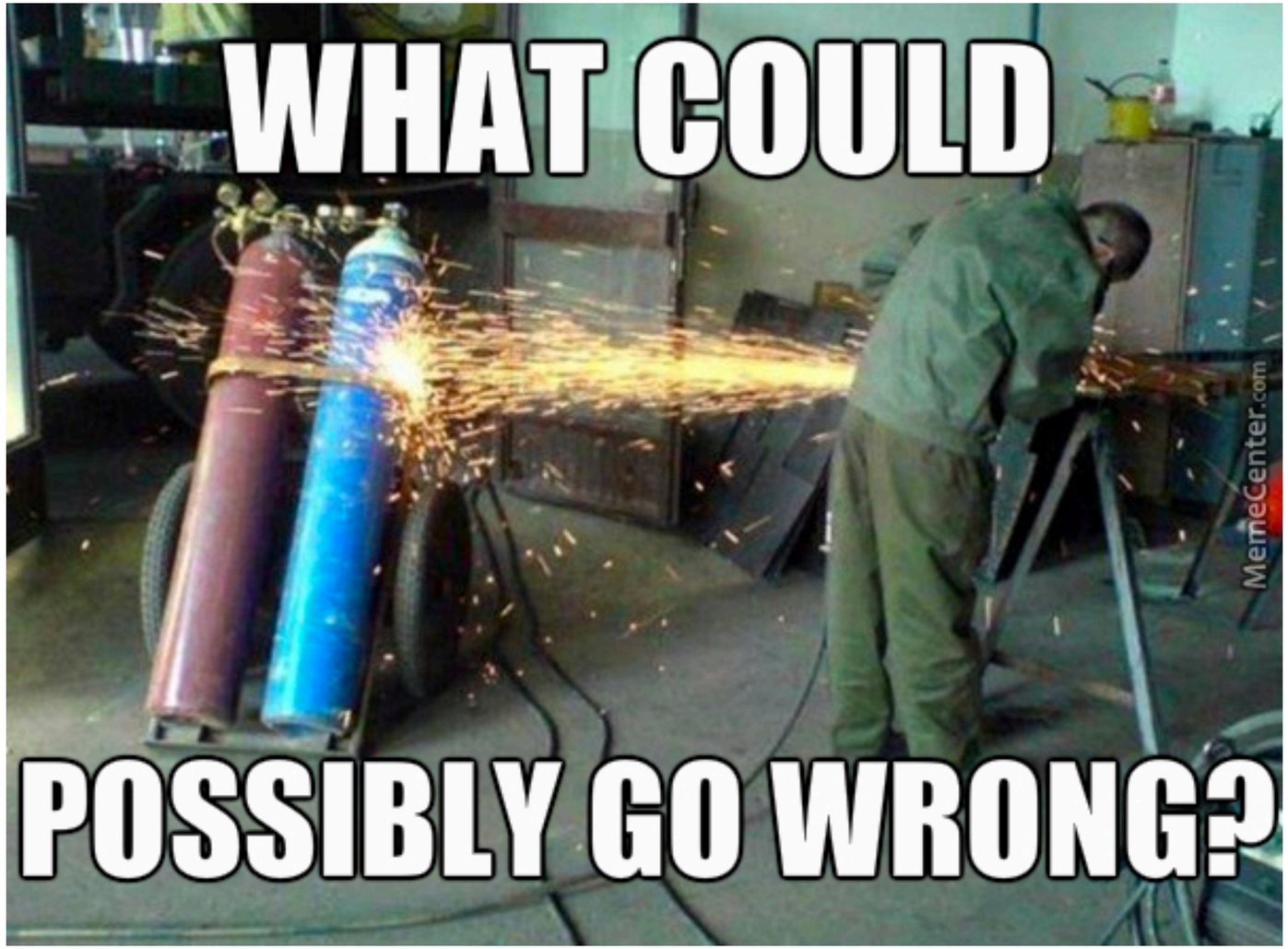


microservices approach



microservices approach





source: https://www.memecenter.com/fun/5802169/what-could-possibly-go-wrong

@n1ko_w1ll

AWS

AWS

AWS Lambda

AWS IoT

Amazon API Gateway

Amazon Cognito

Amazon CloudWatch

State

Amazon DynamoDB

Amazon Elasticsearch

Amazon S3

Fast Data

Amazon SQS

Amazon SNS

Amazon Kinesis

@n1ko_w1ll

AWS Lambda

AWS Lambda

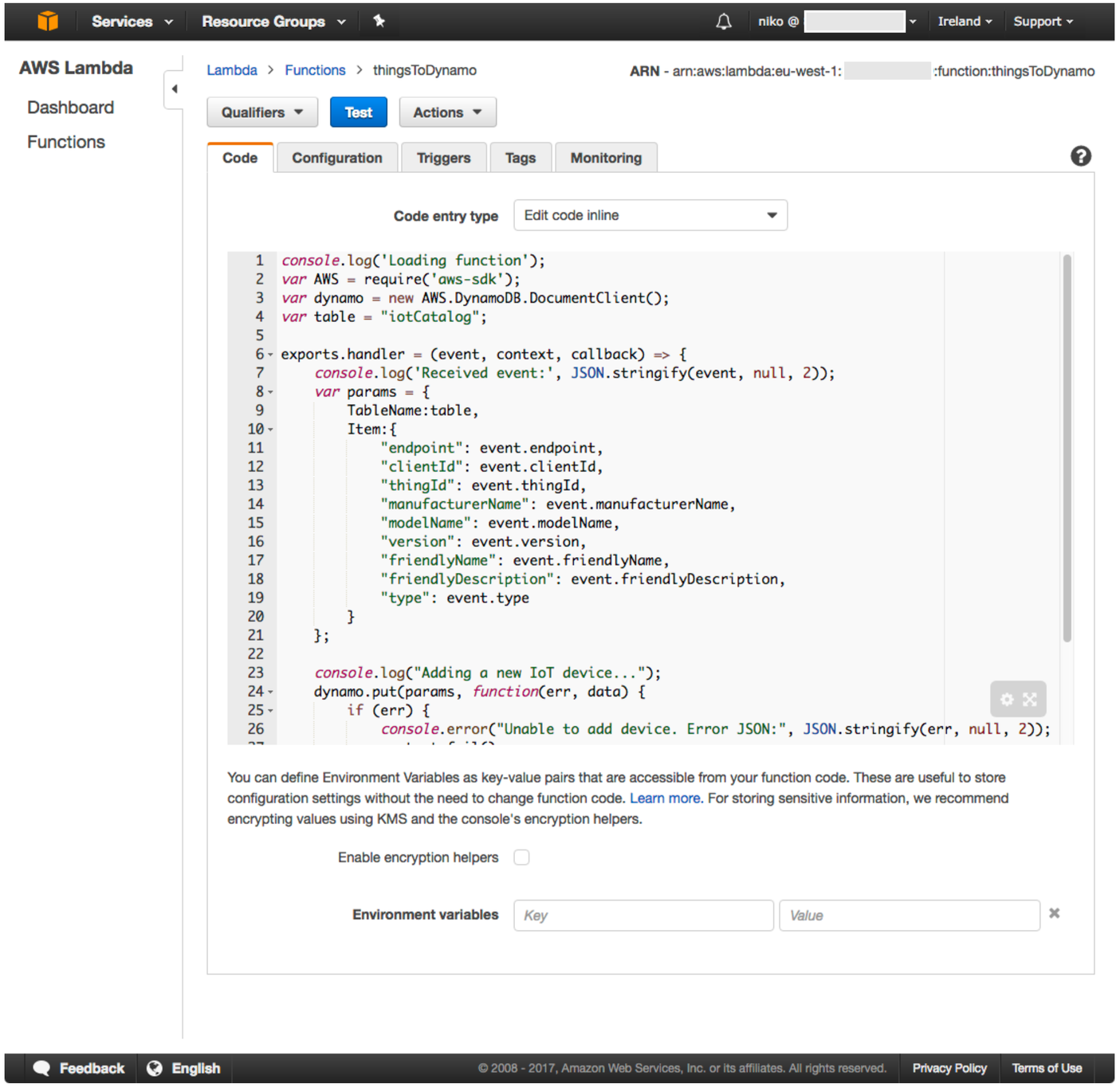
- Functions-as-a-Service (FaaS)
  - serverless
  - „small“ functions
- stateless compute containers
- event-driven

AWS Lambda

- advantages
  - scalable
  - pay-per-execution / pay-as-you-go
    - no upfront capacity planning
  - significantly reduce operational cost

AWS Lambda

- disadvantages
  - vendor lock-in
  - startup latency
  - testing
  - debugging
  - execution duration


Services
Resource Groups
niko @
Ireland
Support
AWS Lambda
Dashboard
Functions
Lambda > Functions > thingsToDynamo
ARN - arn:aws:lambda:eu-west-1: :function:thingsToDynamo
Qualifiers
Test
Actions
Code
Configuration
Triggers
Tags
Monitoring
Code entry type
Edit code inline
1 console.log('Loading function');
2 var AWS = require('aws-sdk');
3 var dynamo = new AWS.DynamoDB.DocumentClient();
4 var table = "iotCatalog";
5
6 exports.handler = (event, context, callback) => {
7 console.log('Received event:', JSON.stringify(event, null, 2));
8 var params = {
9 TableName:table,
10 Item:{
11 "endpoint": event.endpoint,
12 "clientId": event.clientId,
13 "thingId": event.thingId,
14 "manufacturerName": event.manufacturerName,
15 "modelName": event.modelName,
16 "version": event.version,
17 "friendlyName": event.friendlyName,
18 "friendlyDescription": event.friendlyDescription,
19 "type": event.type
20 }
21 };
22
23 console.log("Adding a new IoT device...");
24 dynamo.put(params, function(err, data) {
25 if (err) {
26 console.error("Unable to add device. Error JSON:", JSON.stringify(err, null, 2));
You can define Environment Variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. Learn more. For storing sensitive information, we recommend encrypting values using KMS and the console's encryption helpers.
Enable encryption helpers
Environment variables
Key
Value
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

@n1ko_w1ll

AWS IoT

AWS IoT

- managed service
- message broker
- rules engine
- shadows
- registry
- security

message broker

- topic based
- publish / subscribe
  - topic wildcards
- protocols
  - MQTT
  - MQTT + WebSockets
  - HTTP

```
$aws/events/presence/connected/clientId
$aws/events/presence/disconnected/clientId

$aws/things/thingName/shadow/update
$aws/things/thingName/shadow/update/delta
```

rules engine

- SQL-like syntax
- augument or filter data
- rule actions
  - state stores
  - fast data pipelines
  - CloudWatch
  - Lambda
  - republish

```
SELECT
     *,
     newuuid() AS requestId,
     clientId() AS clientId,
     timestamp() AS timestamp,
     topic(2) AS deviceId,
     topic(4) AS sensorId
FROM 'device/+/sensor/+/v1'
WHERE temperature > 50 AND color <> 'red'
```

shadows

- JSON document
- current state of thing
- connection independent
- supports client tokens
- supports versioning
- MQTT topics
- RESTful API

```
{
 "state" : {
  "desired" : { "color" : "RED" },
  "reported" : { "color" : „GREEN" }
 },
 "metadata" : {
  "desired" : { "color" : { "timestamp" : 12345 } },
  "reported" : { "color" : { "timestamp" : 12345 } }
 },
 "version" : 10,
 "clientToken" : "UniqueClientToken",
 "timestamp": 123456789
}
```

registry

- manage your things
  - physical device or sensor
  - logical entity
- attributes
- thing types

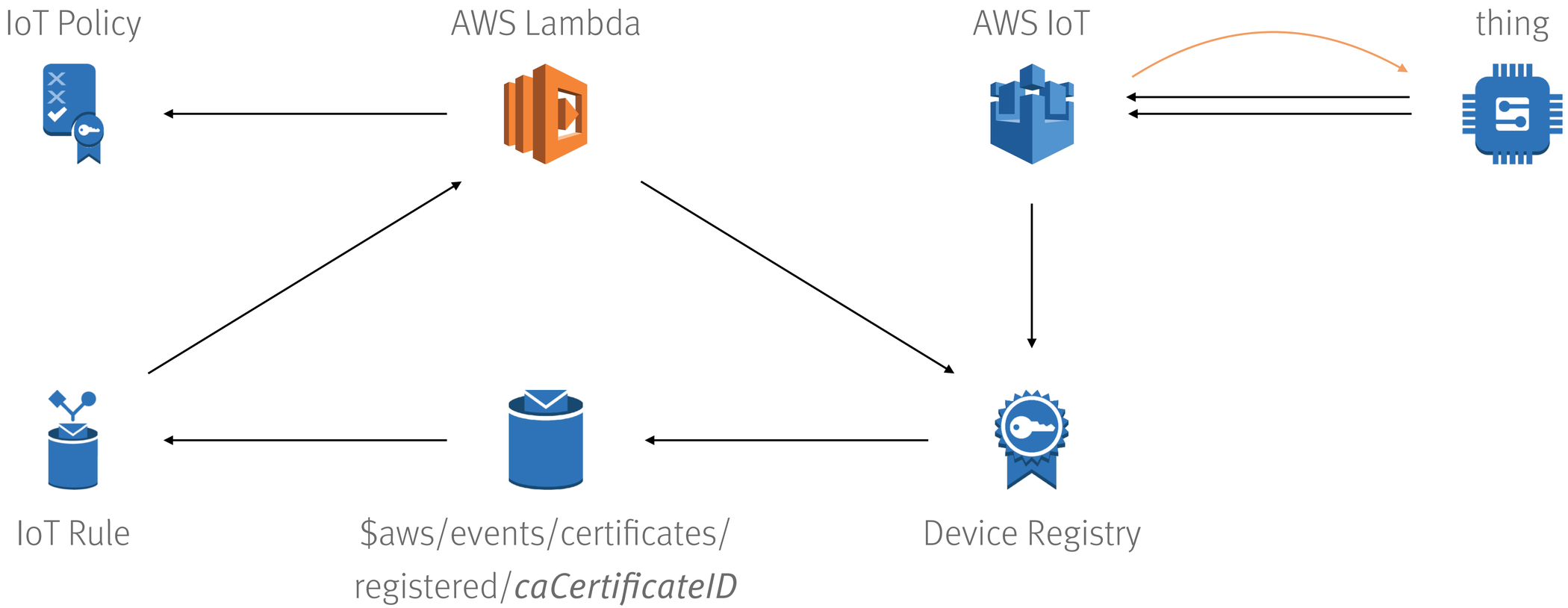
```
{
    "version": 3,
    "thingName": "MyLightBulb",
    "defaultClientId": "MyLightBulb",
    "thingTypeName": "LightBulb",
    "attributes": {
        "model": "123",
        "wattage": "75"
    }
}
```

security

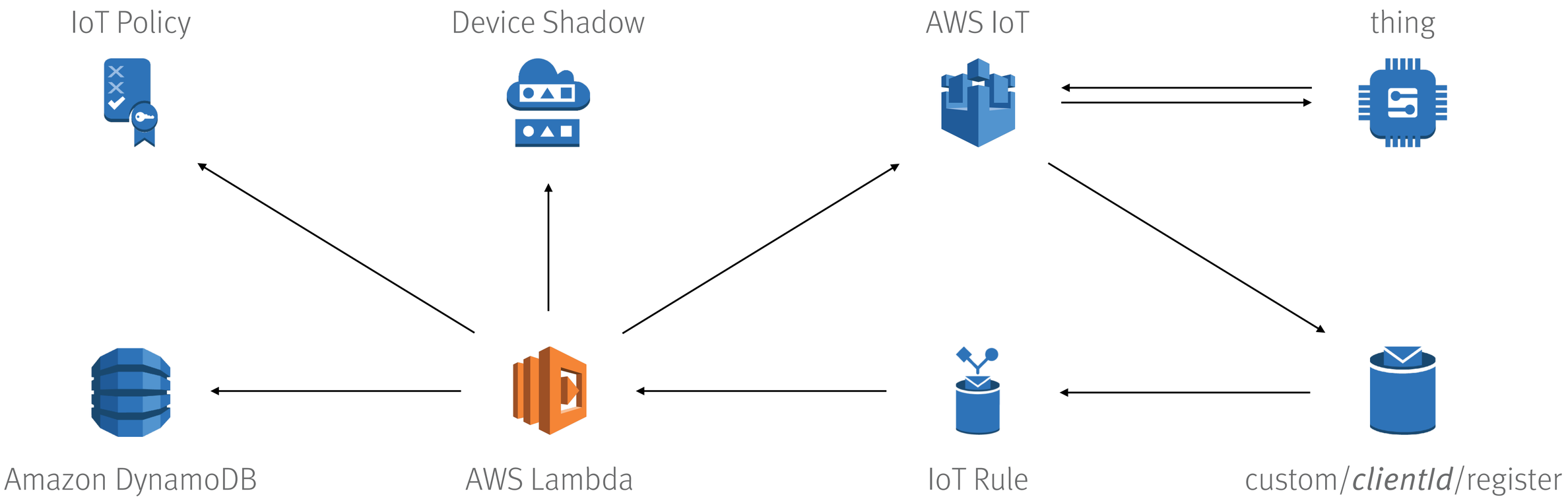
- mutual authentication with X509 certificates + TLS 1.2
  - or SigV4 for HTTPS and WebSockets
- bring your own certificate
  - JITR
  - Atmel ECC508
- policy based access with dynamic values
- role based rules action execution

use-cases

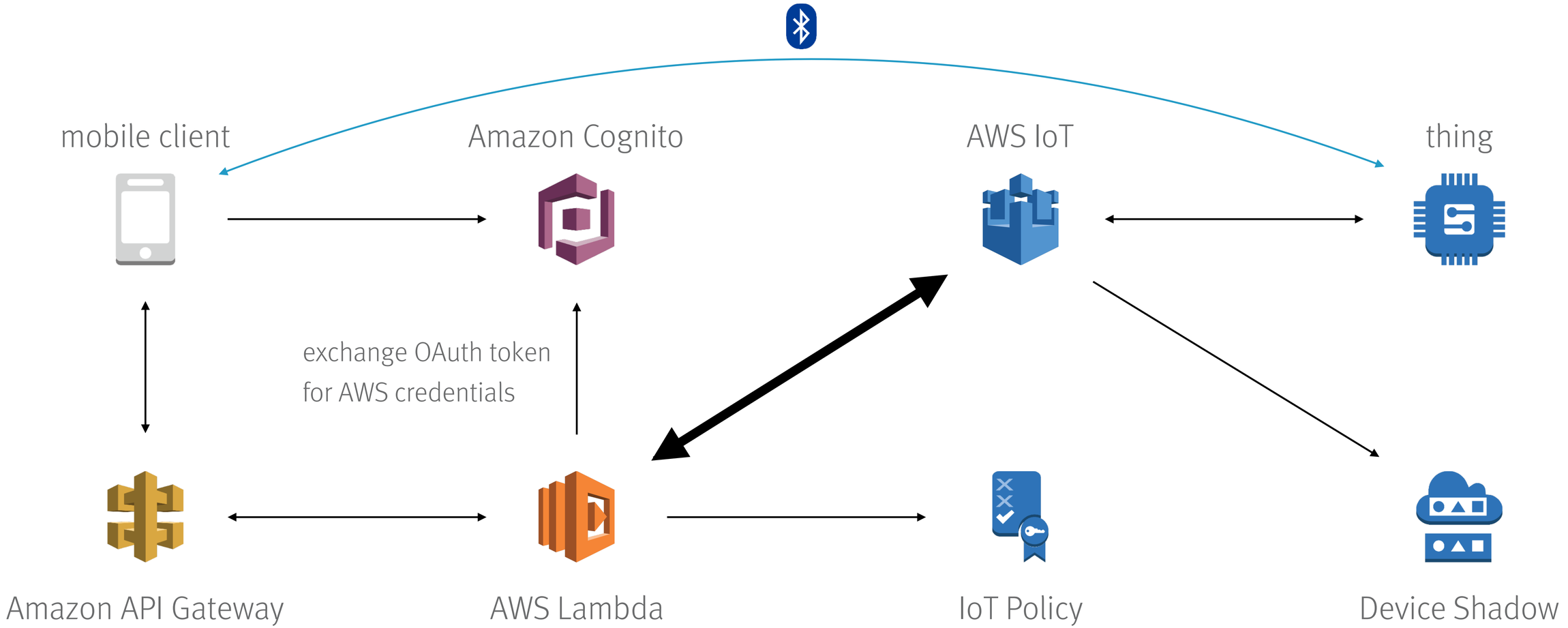
just in time registration (JITR)



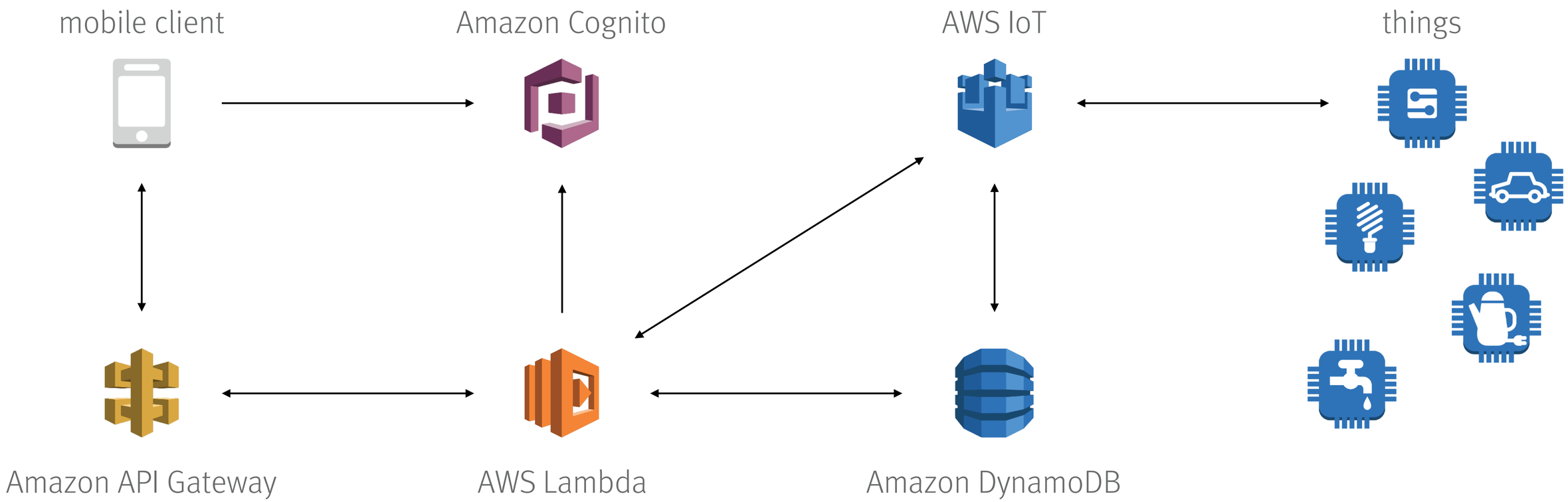
on-boarding


IoT Policy
Device Shadow
AWS IoT
thing
Amazon DynamoDB
AWS Lambda
IoT Rule
custom/*clientId*/register

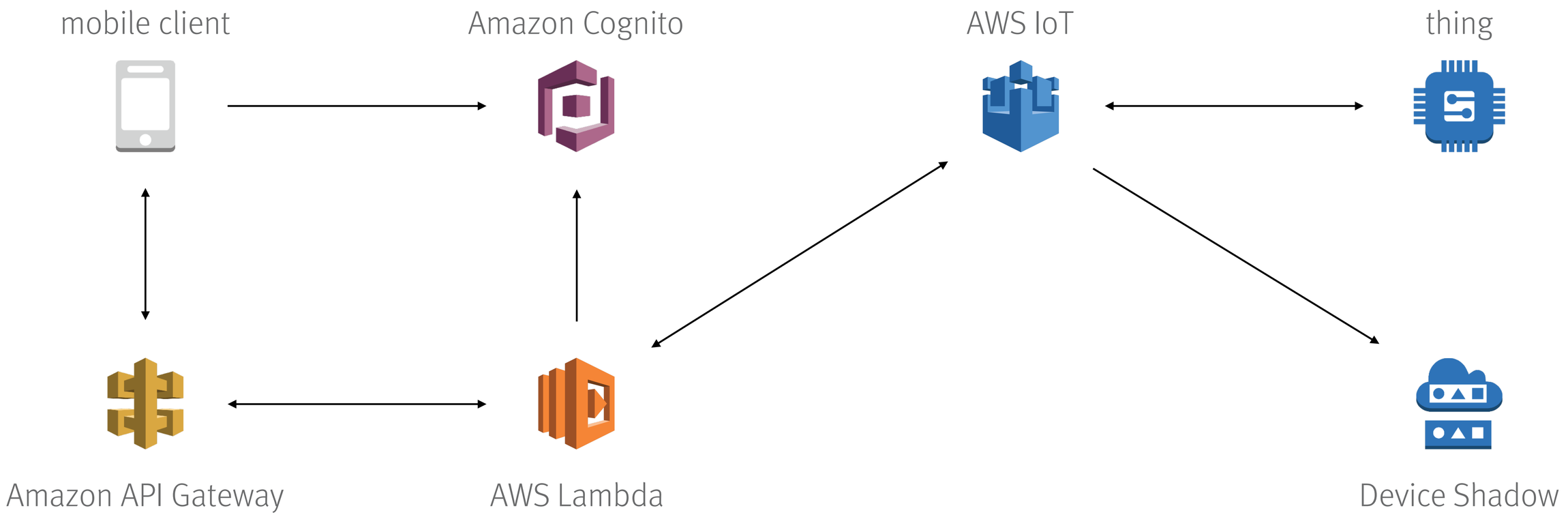
thing pairing



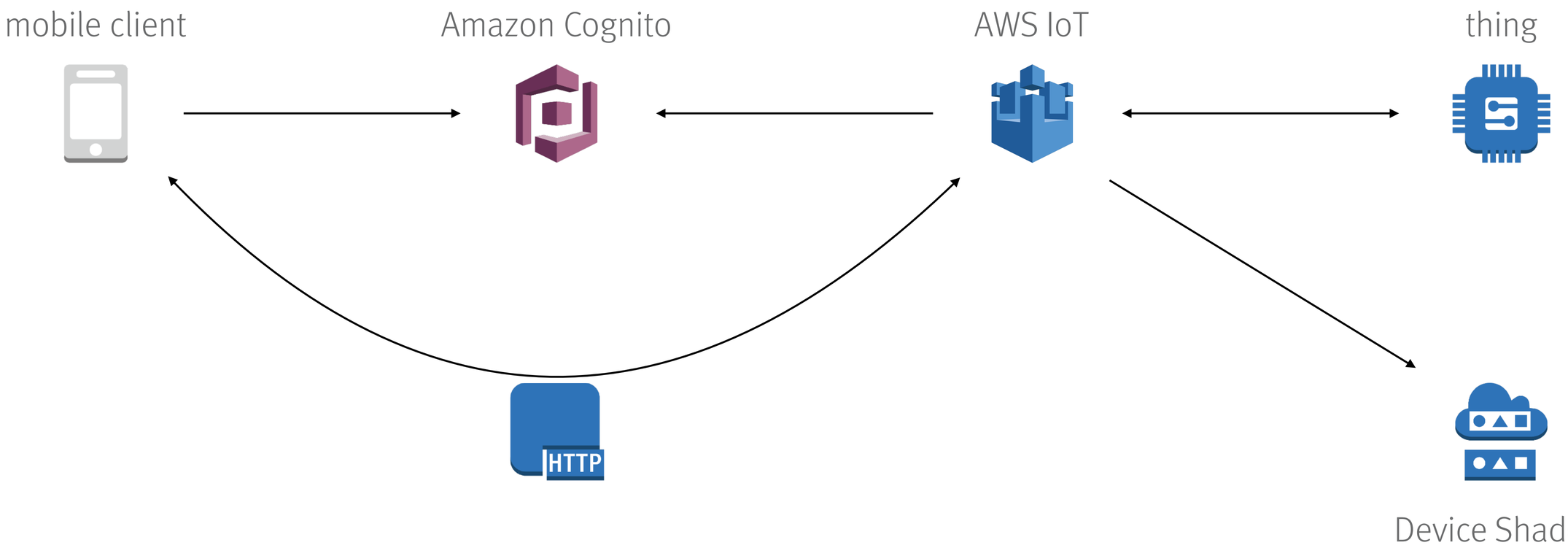
list / search things



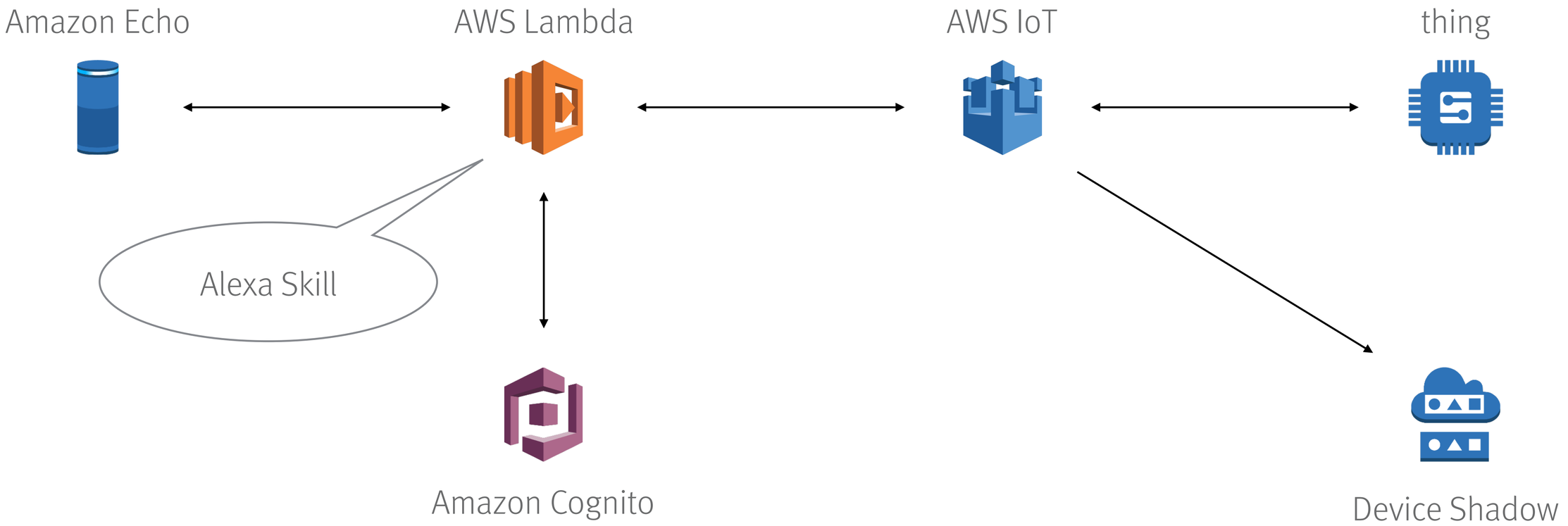
command & control



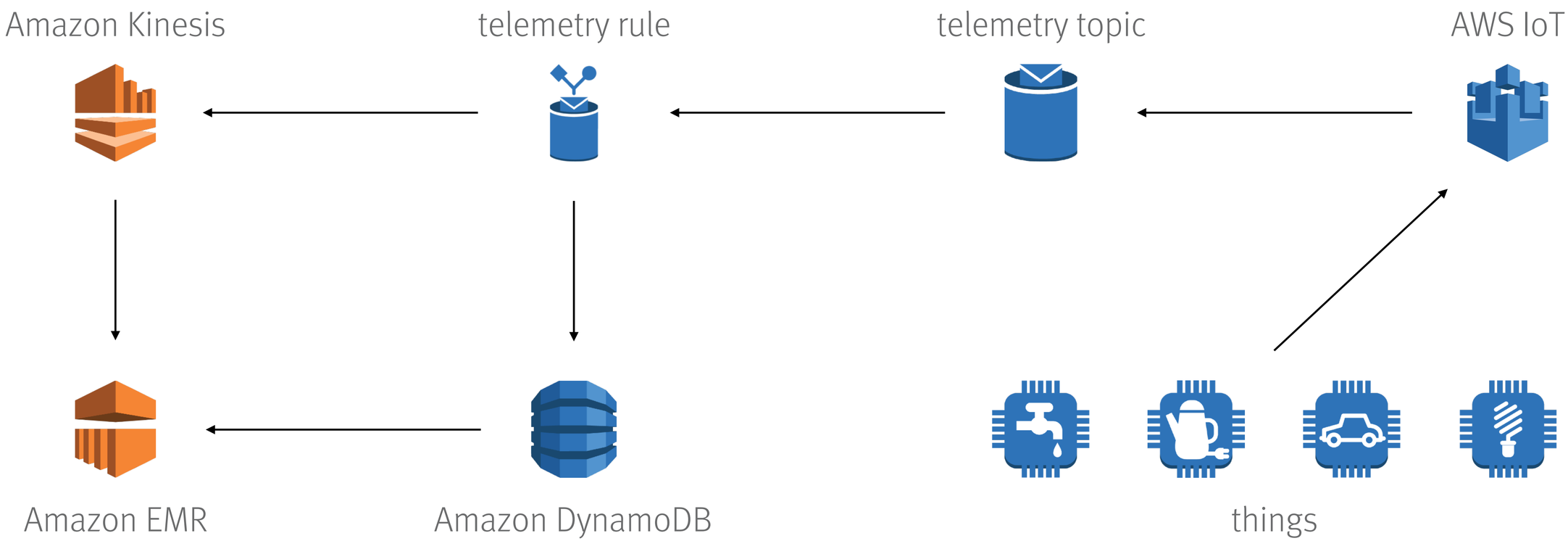
command & control



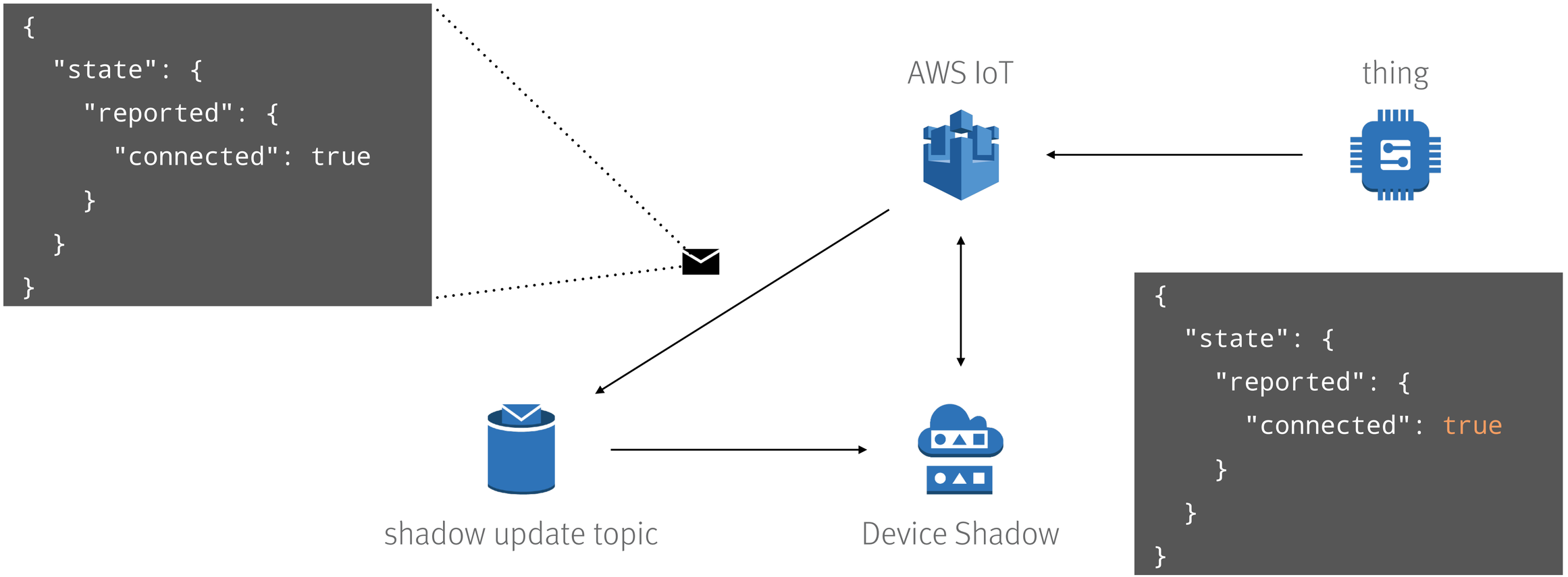
command & control



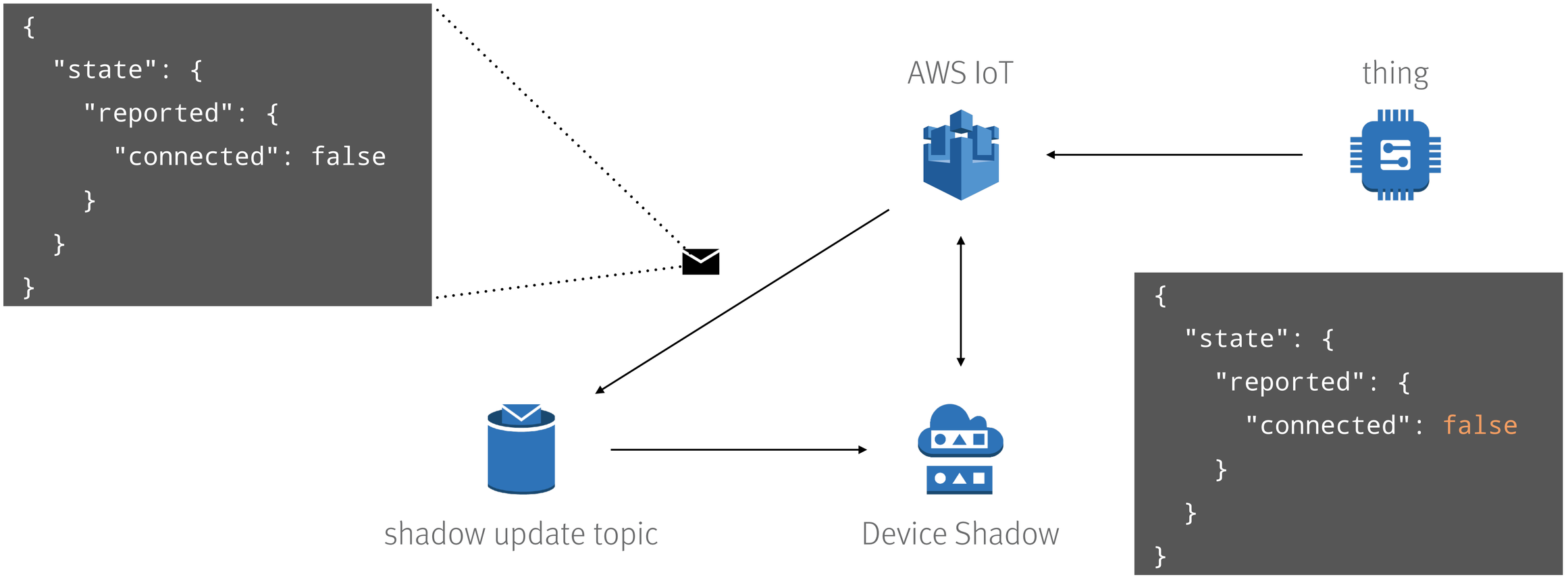
telemetry


Amazon Kinesis
telemetry rule
telemetry topic
AWS IoT
Amazon EMR
Amazon DynamoDB
things

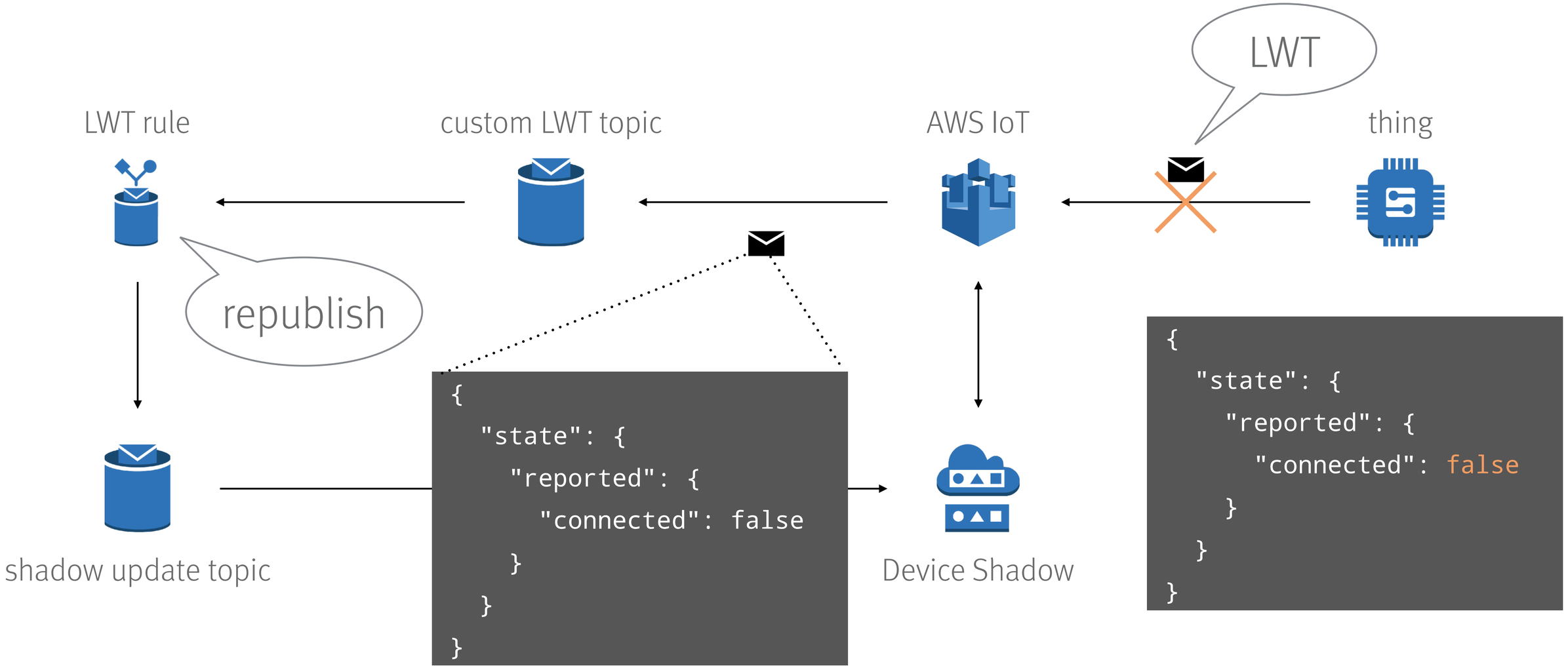
connected / disconnected / LWT



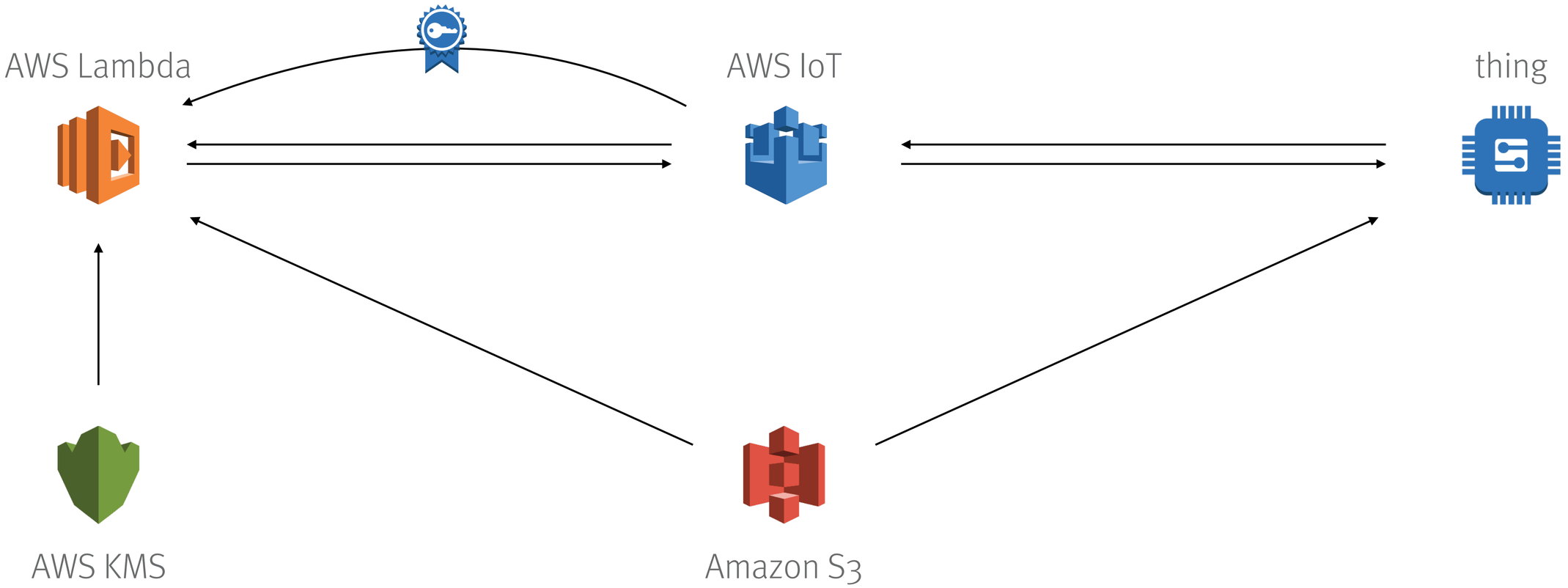
connected / disconnected / LWT



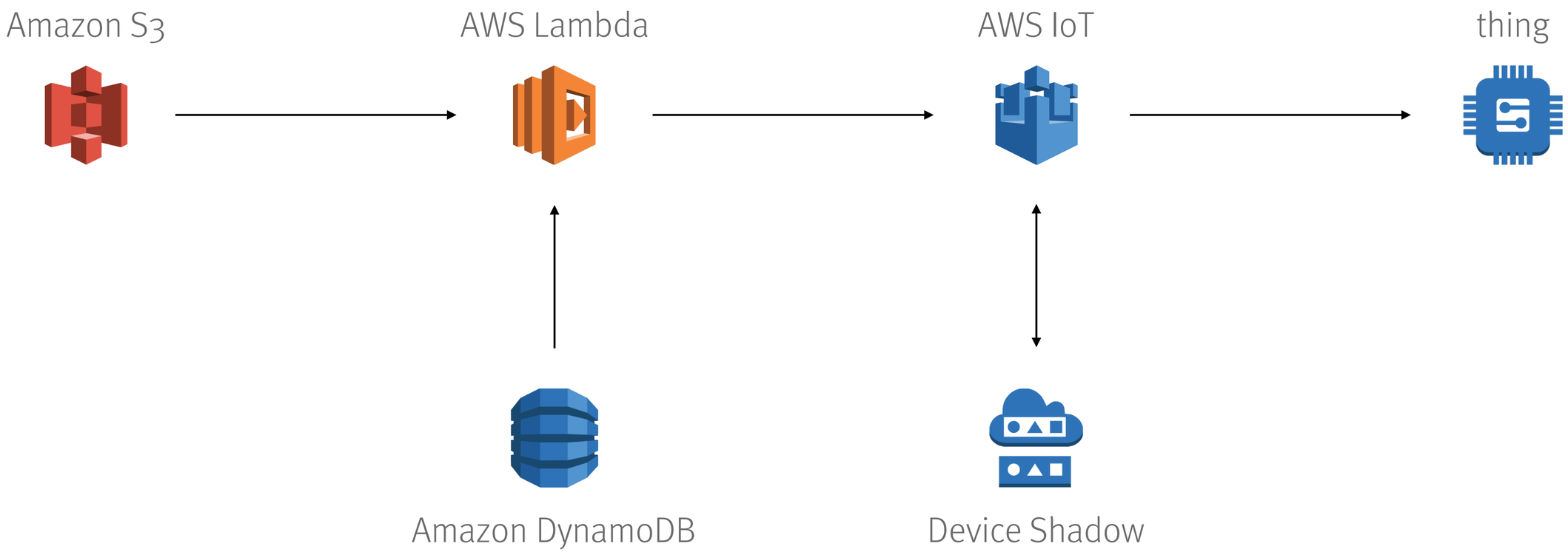
connected / disconnected / LWT



encrypted file transfer



firmware update



summary

- scalable plattform
- common IoT use-cases
- w/o own infrastructure
- w/o upfront capacity planning
- very secure
- very extensible

Thank you.

Questions?

Comments

@n1ko_w1ll

Niko Will

niko.will@innoq.com

innoQ

www.innoq.com

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim am Rhein
Germany
Phone: +49 2173 3366-0

Ohlauer Straße 43
10999 Berlin
Germany
Phone: +49 2173 3366-0

Ludwigstr. 180E
63067 Offenbach
Germany
Phone: +49 2173 3366-0

Kreuzstraße 16
80331 München
Germany
Phone: +49 2173 3366-0

innoQ Schweiz GmbH

Gewerbestr. 11
CH-6330 Cham
Switzerland
Phone: +41 41 743 0116