# Der sprechende Kickertisch

Marco Buss

## Marco Buss

Senior Consultant | AWS Certified Solutions Architect - Assosiate

0173-727 9018

marco.buss@opitz-consulting.com

Tempelhofer Weg 64

12347 Berlin

WWW.OPITZ-CONSULTING.COM

@OC_WIRE

OPITZCONSULTING

opitzconsulting

opitz-consulting-bcb8-1009116

# ■■■ überraschend mehr Möglichkeiten!

„Mit unserer Leidenschaft für neue Technologien und unserem Anspruch an herausragende Beratung sind wir bei unseren Kunden der Motor für die digitale Transformation."

Bernhard Opitz, CEO von OPITZ CONSULTING

Kunden (Auszug)

## Portfolio:

- Spezialist für digitale Transformation
- Software Development
- BI & Big Data
- BPM & Systemintegration
- Cloud & Infrastruktur
- Internet der Dinge
- Managed Services
- Oracle Lizenzmanagement
- Strategy & Change

## Überblick:

- Gründung 1990
- 8 Standorte in Deutschland
- 3 Standorte in Polen
- Ca. 400 Mitarbeiter
- Inhabergeführt
- Breites Themenportfolio
- Herstellerunabhängige Beratung

## Märkte:

- Branchenübergreifend
- 600 Kunden
- 2/3 aller Dax-Unternehmen
- 29% Handel/Logistik/Service
- 29% Industrie/Versorgung/ Telko
- 42% Finanzsektor/Public

1 Übersicht

2 AWS IoT

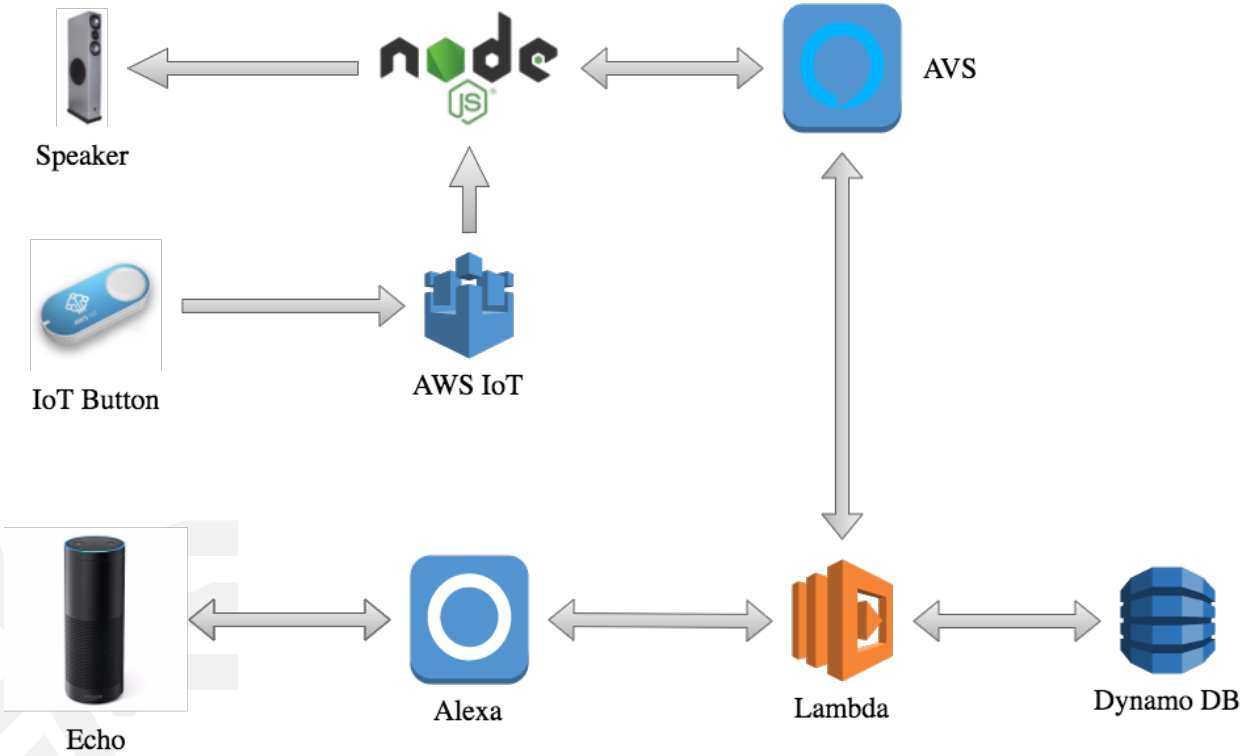
3 Alexa Skill Kit

4 Alexa Voice Services

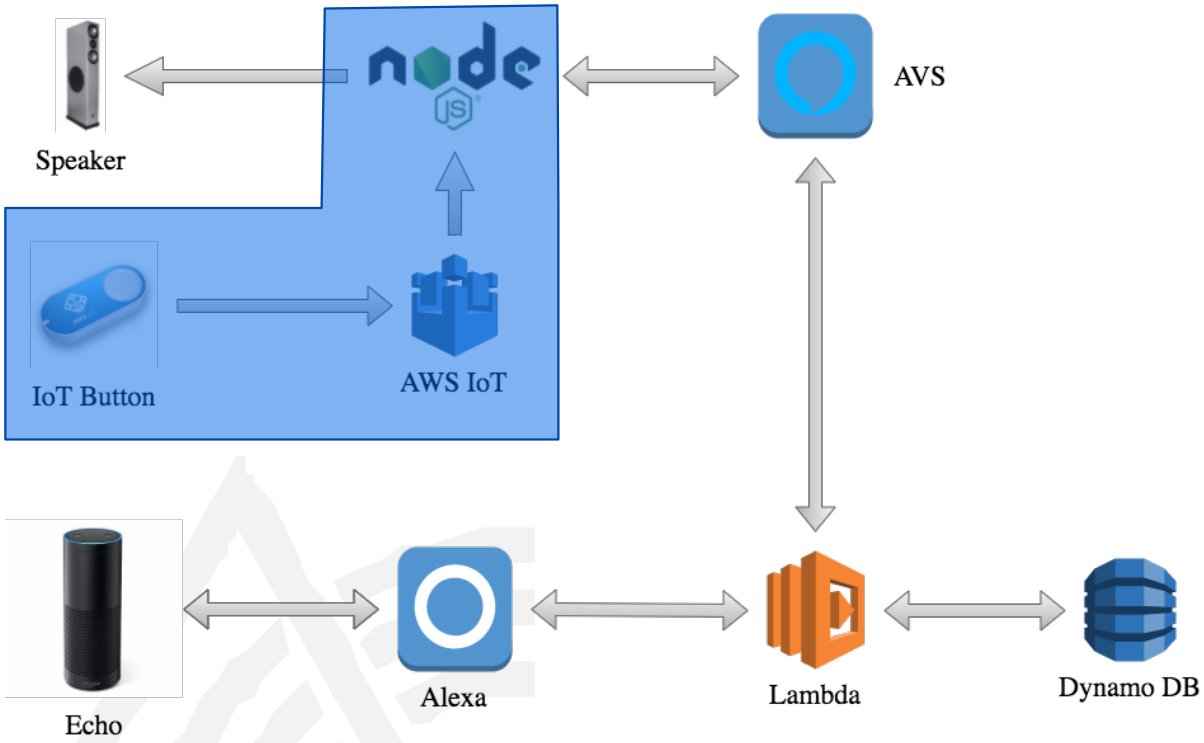
# Übersicht


TECBALL
PROFESSIONAL TABLE SOCCER
®
.com

# Übersicht

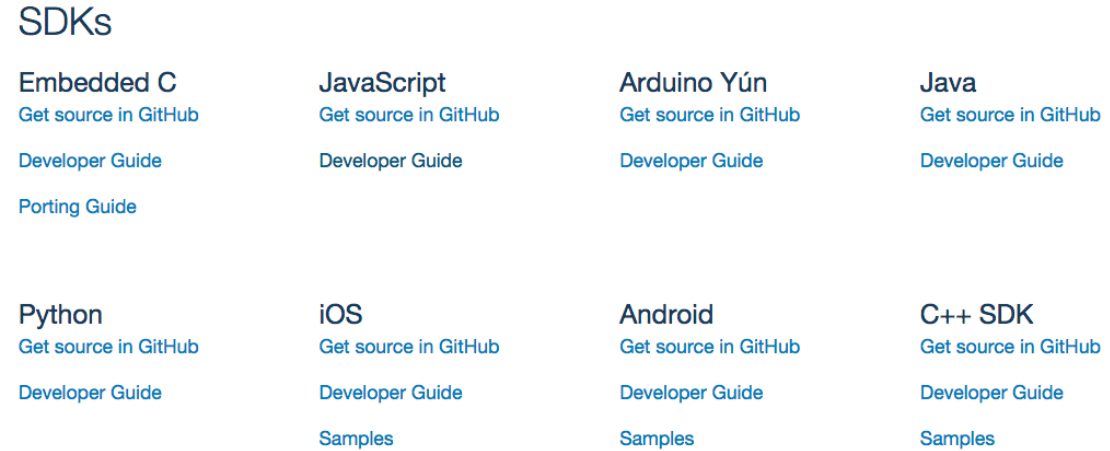


# AWS IoT

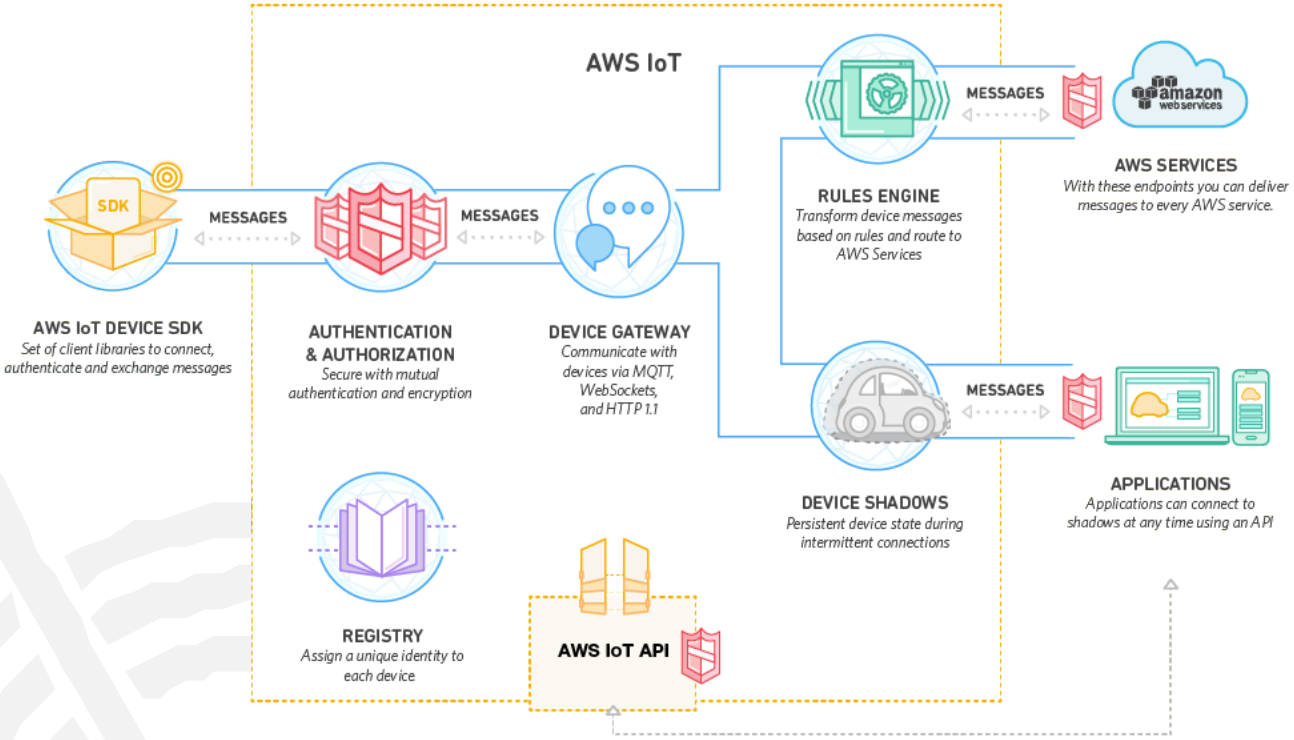



# AWS IoT 101

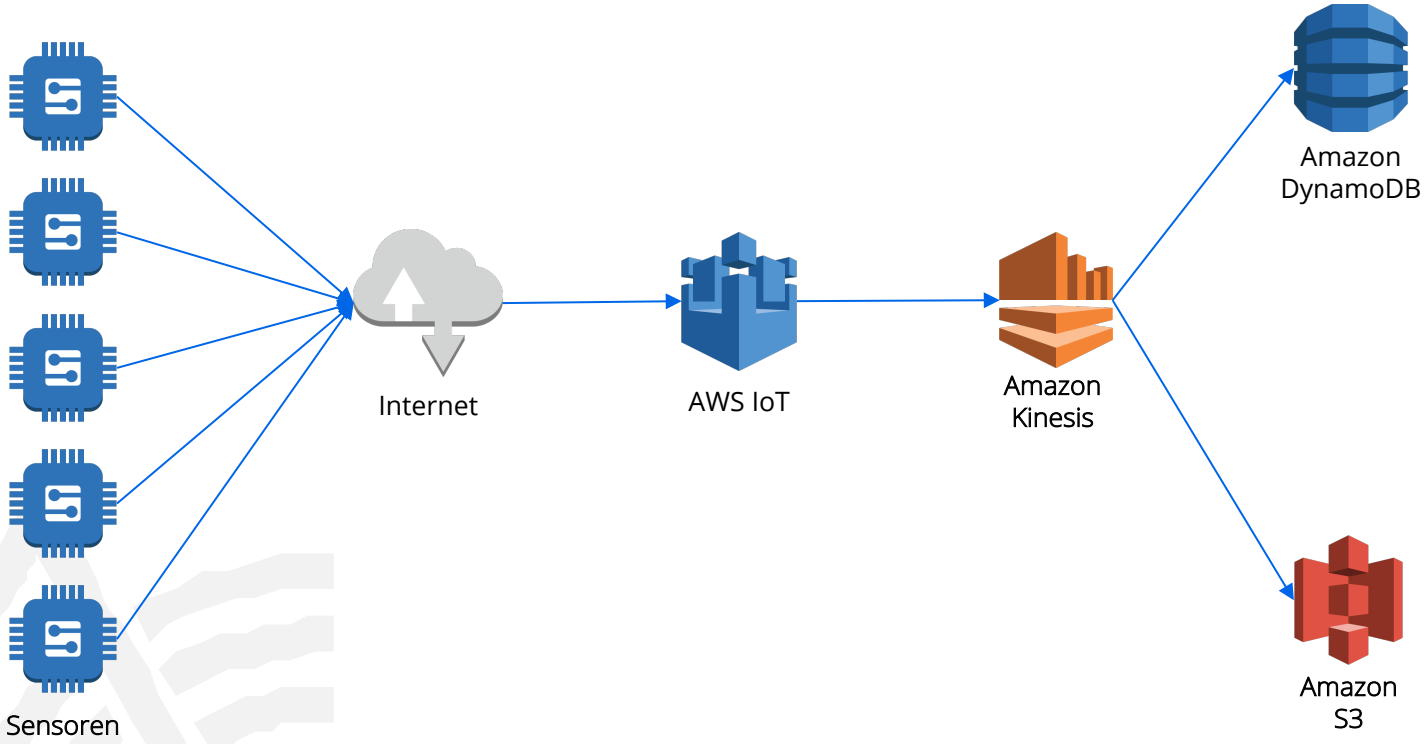
- Amazons IoT Plattform
- Serverless Architekture
- Bezahlt wird für die Nachrichten (5$ / 1 Mio Nachrichten)
- SDK´s verfügbar



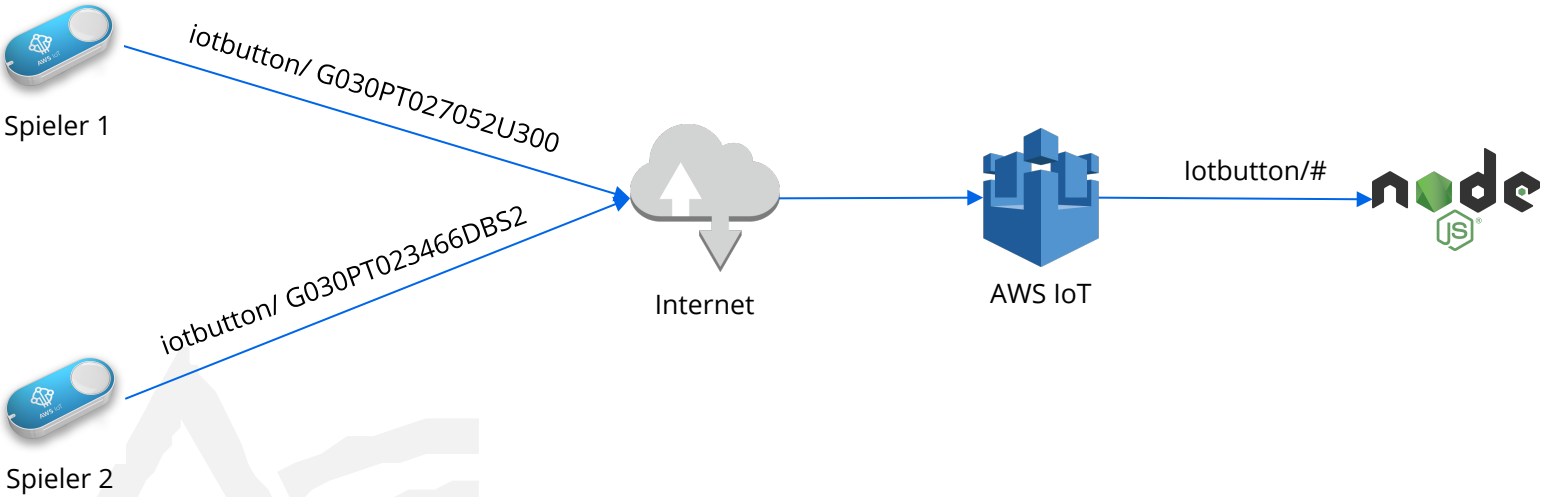
# AWS IoT 101


AWS IoT
SDK
MESSAGES
MESSAGES
MESSAGES
MESSAGES
amazon web services
AWS IoT DEVICE SDK
Set of client libraries to connect, authenticate and exchange messages
AUTHENTICATION & AUTHORIZATION
Secure with mutual authentication and encryption
DEVICE GATEWAY
Communicate with devices via MQTT, WebSockets, and HTTP 1.1
RULES ENGINE
Transform device messages based on rules and route to AWS Services
AWS SERVICES
With these endpoints you can deliver messages to every AWS service.
DEVICE SHADOWS
Persistent device state during intermittent connections
APPLICATIONS
Applications can connect to shadows at any time using an API
REGISTRY
Assign a unique identity to each device
AWS IoT API

# Common Use Case

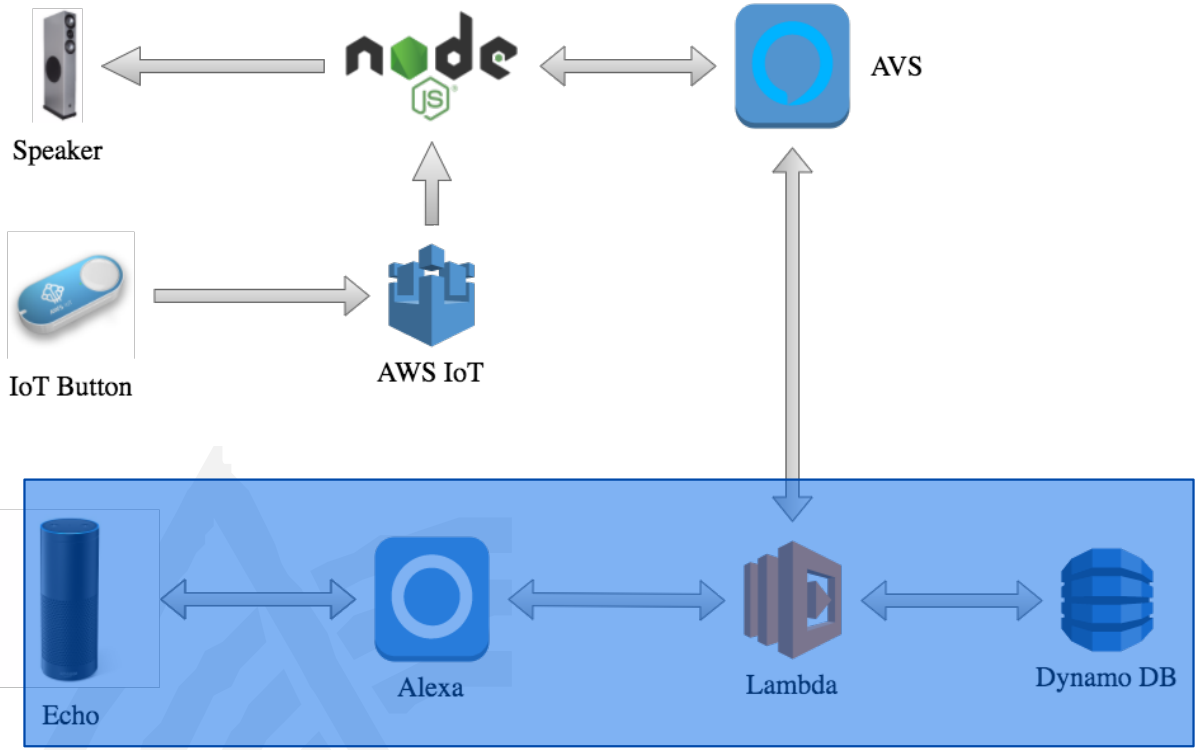


# Anwendung für Kickertisch




Some Code

# Alexa Skill Kit




# Alexa Skill Kit 101

- Es gibt mehrere Skill Typen
  - Custom Interaction Skills
  - Smart Home Skills
  - Flash Briefing Skills
  - Video Skills

- Zwei Hauptbestandteile
  - Interaction Model
  - Service Funktion

# Interaction Model: Built In Intents

- Von Amazon bereits bereitgestellte Intents
- Decken Grundfunktionalitäten ab
- Es müssen keine Sample Uterances definiert werden

```
{"intents": [
  {"intent": "AMAZON.HelpIntent"},
  {"intent": "AMAZON.StopIntent"},
  {"intent": "AMAZON.CancelIntent"},
  {"intent": "AMAZON.YesIntent"},
  {"intent": "AMAZON.NoIntent"}
]}
```

# Interaction Model: Custom Intents

- Hauptbestandteil für eigenes Interaktionsmodel
- Können Slots enthalten
- Slots sind die Parameter der Intents
- Mehrere Build-In Slot Types + Eigene

```
{"intents": [
  {"intent": "StartGame"},
  {"intent": "Punktestand"},
  {"intent": "PlayerPoint",
    "slots": [
      {
        "name": "Player",
        "type": "SPIELER_PUNKT"
      }
    ]},
]}
```

# Interaction Model: Sample Uterances

- Definieren welche Sätze auf welchen Intent matchen
- Viel hilft viel!!!
- Slots werden mit {} markiert

StartGame starte neues match
StartGame beginne neues match
StartGame starte ein neues spiel
StartGame beginne ein neues spiel

Punktestand wie ist der punktestand
Punktestand wie steht es
Punktestand gib mir den punktestand
Punktestand wer führt
Punktestand wer gewinnt

PlayerPoint Punkt für {PLAYER_POINT}

# Interaction Model:

https://jaxenter.de/chatbot-alexa-google-home-siri-54265

**Christian Ochsenkühn**

Christian Ochsenkühn arbeitet bei der OPITZ CONSULTING Deutschland GmbH im Bereich Software Development. Zudem beschäftigt er sich für die Abteilung Business Development & Innovation mit aktuellen Trends und innovativen Lösungen für digitale Problemstellungen. Als Betreiber eigener Webangebote kennt er die Herausforderungen des schnellen technischen Wandels und konzentriert sich auf eine optimale Benutzererfahrung.

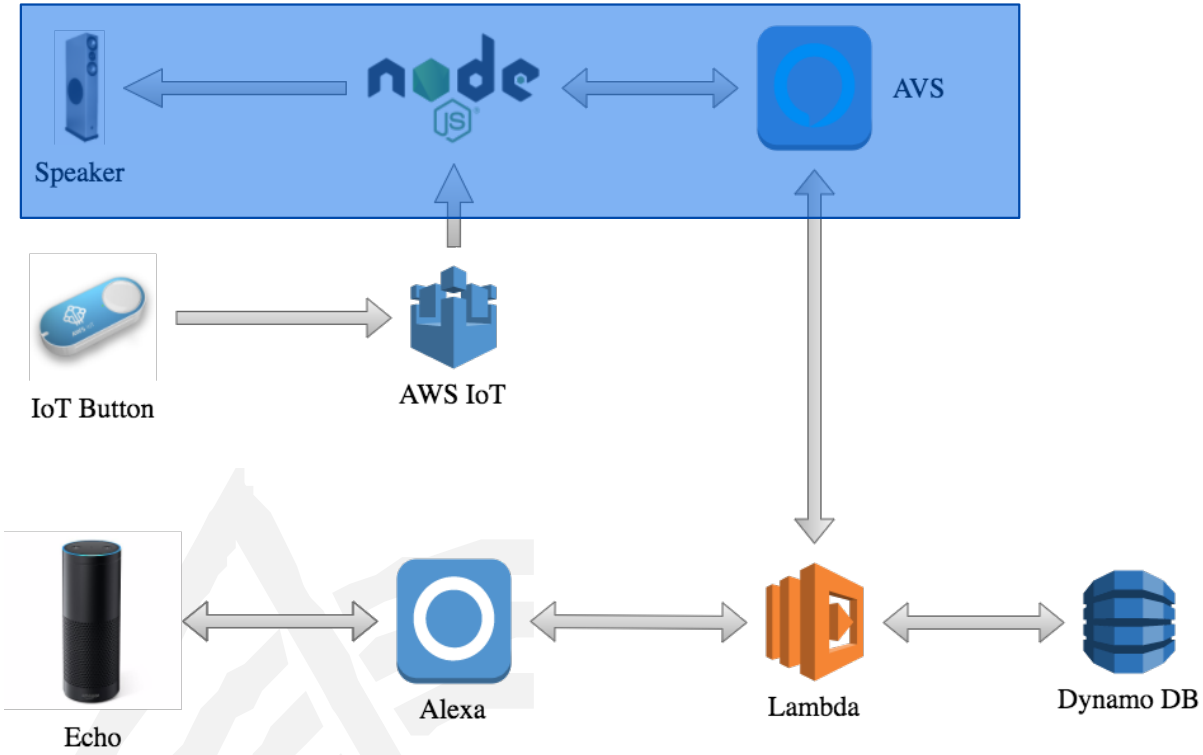
**Alle Beiträge von Christian Ochsenkühn**

# Service Function

- Werden entweder als AWS Lambda Function oder Webservice implementiert
- Für Lambda existieren mehrere SDK´s
- NodeJS und Python haben die größte Verbreitung

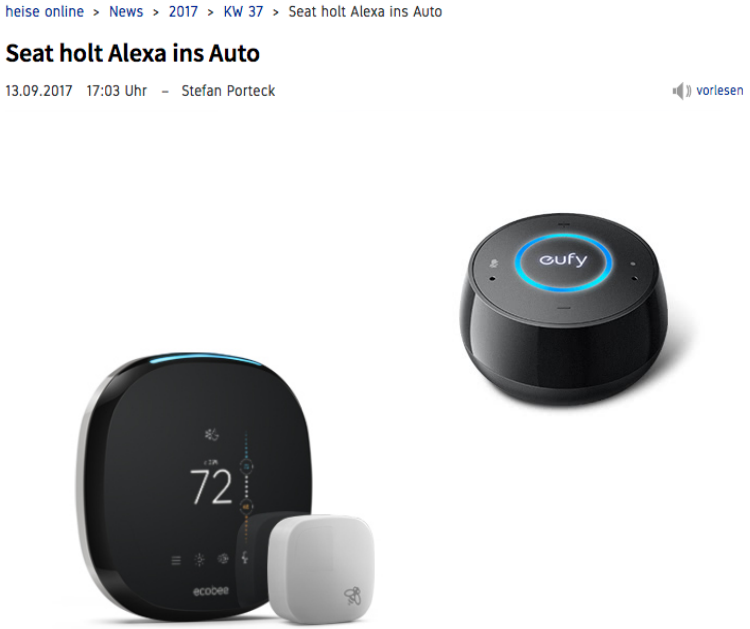

Some Code

# Alexa Voice Service




# Alexa Voice Service 101

- Integration von Alexa in eigenes Device
- SDK für C++ verfügbar
- HTTP/2 API



# Warum in NodeJS?

# Weil es geht!!!

Und ich kein C++ kann.

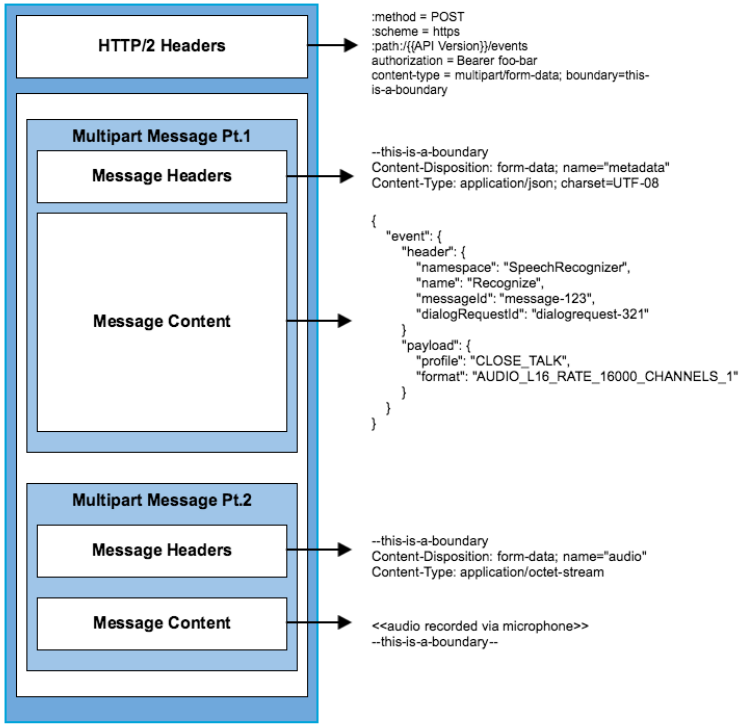
# AVS API v2

- Umstieg auf HTTP/2
- Erfordert Erzeugung eines „Downstream“ Kanals
- Ping alle 5 Minuten

# Authorization

- Login with Amazon Access Token wird benötigt
- Authorisierung per CompanionApp
- Authorisierung per Companion Site
- Authorizierung AVS Produkt

# Aufbau HTTP/2 Multipart Message



# SpeechRecognizer

- Einziger API Endpoint der im Projekt genutzt wurde
- Audio Sample mit dem Kommando wird an AVS gesendet
- Das Sample wird ausgewertet und der entsprechende Alexa Skill aktiviert
- Die Antwort des Skills wird über den Lautsprecher ausgegeben

# Probleme bei der Benutzung von NodeJS und AVS

- HTTP/2
  - HTTP/2 in NodeJS 8.4 funktionierte nicht für die Authorisierung
  - HTTP/2 (https://github.com/molnarg/node-http2) für Node 6.10 hatte Probleme bei größeren Audio Files
  - HTTP/2 in NodeJS 6.2.2 funktioniert
- Downchannel über längeren Zeitraum offen halten hat lange gedauert
- Erstes Modul für Audio Ausgabe war nicht gut

# Fragen?