

# Kolloquium

## Konzeption eines Prototypen zur automatisierten Abfrage der Preis-API von public Cloud-Anbietern als Grundlage für Sky Computing

Vito D`Elia

Matrikelnummer: 1320417

Referent: Henry-Norbert Cocos

Korreferent: Prof. Dr. Christian Baun

Informatik

Fachbereich 2

Frankfurt University of Applied Sciences

30.08.2024

# Agenda

- 1) Problemstellung und Forschungsfrage
- 2) Stand der Technik
- 3) Design
- 4) Implementierung
- 5) Validierung
- 6) Fazit und Ausblick
- 7) Live-Demo

# Problemstellung und Forschungsfrage

- ❖ Einsatz von Diensten verschiedener Cloud-Anbieter
  - Bei langfristiger Nutzung
    - Schwankender Überblick
    - Dynamische Preisveränderungen
  - Problem: Fehlende Kostenübersicht
    - Keine Überwachungsoption
    - Schwierige Preisvergleichbarkeit

Welche Möglichkeiten zur Sammlung der Preisinformationen von public Cloud-Anbietern existieren und wie kann man diese automatisiert Abfragen und Auswerten?

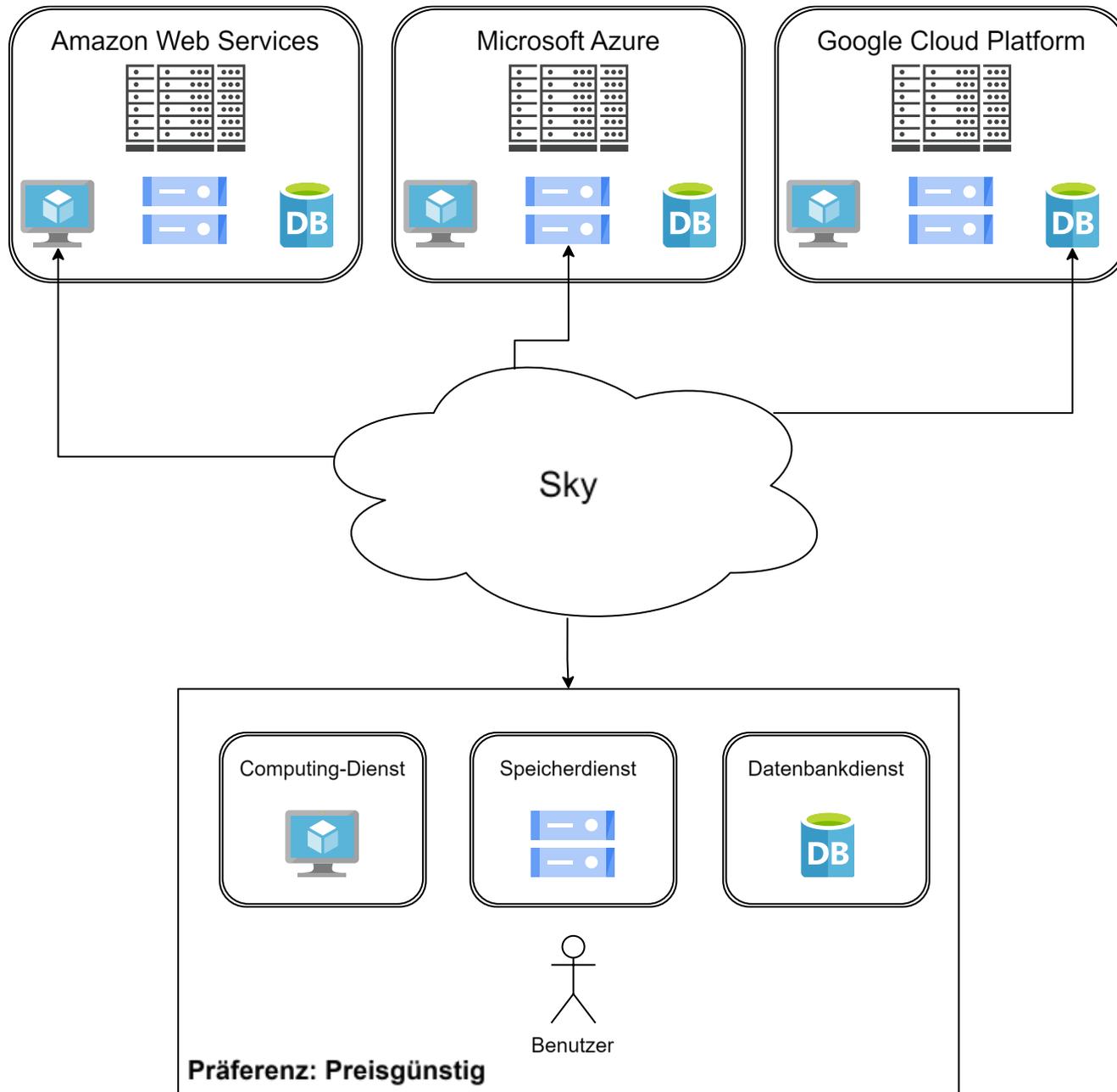
# Stand der Technik

# Multi-Cloud-Umgebung

- Kombinierte Nutzung von Cloud-Ressourcen
  - mindestens zwei öffentliche Clouds
  - mindestens eine private Cloud
- Grund
  - Redundanz im Falle von Ausfällen
  - Anbieterabhängigkeit meiden
  - Kosteneffizienz
- Probleme
  - Dienste meist proprietär → keine Kompatibilität
  - Kostenunübersichtlichkeit

- Neues Forschungsfeld
  - Aufbau ähnelt dem einer Multi-Cloud
  - Zusätzliche Abstraktionsschichten
- Ziel: Interoperable Cloud-Plattform
  - Einheitliche Gestaltung
    - Anbietern
    - Diensten
    - Ressourcen
  - Anbieterübergreifende Ressourcennutzung

- Kompatibilitätsschicht
  - Sammlung aus freien Software-Lösungen
  - Technische Anbindung
- Intercloud-Schicht
  - Einheitliches Namensschema
  - Verzeichnissystem
  - Accountmanagement
- Peering-Vereinbarung
  - Kostenlose Datenübertragung
- Finanzielle Analyse
  - Aktuelle Preisinformationen
  - Momentane Kosten des Benutzers

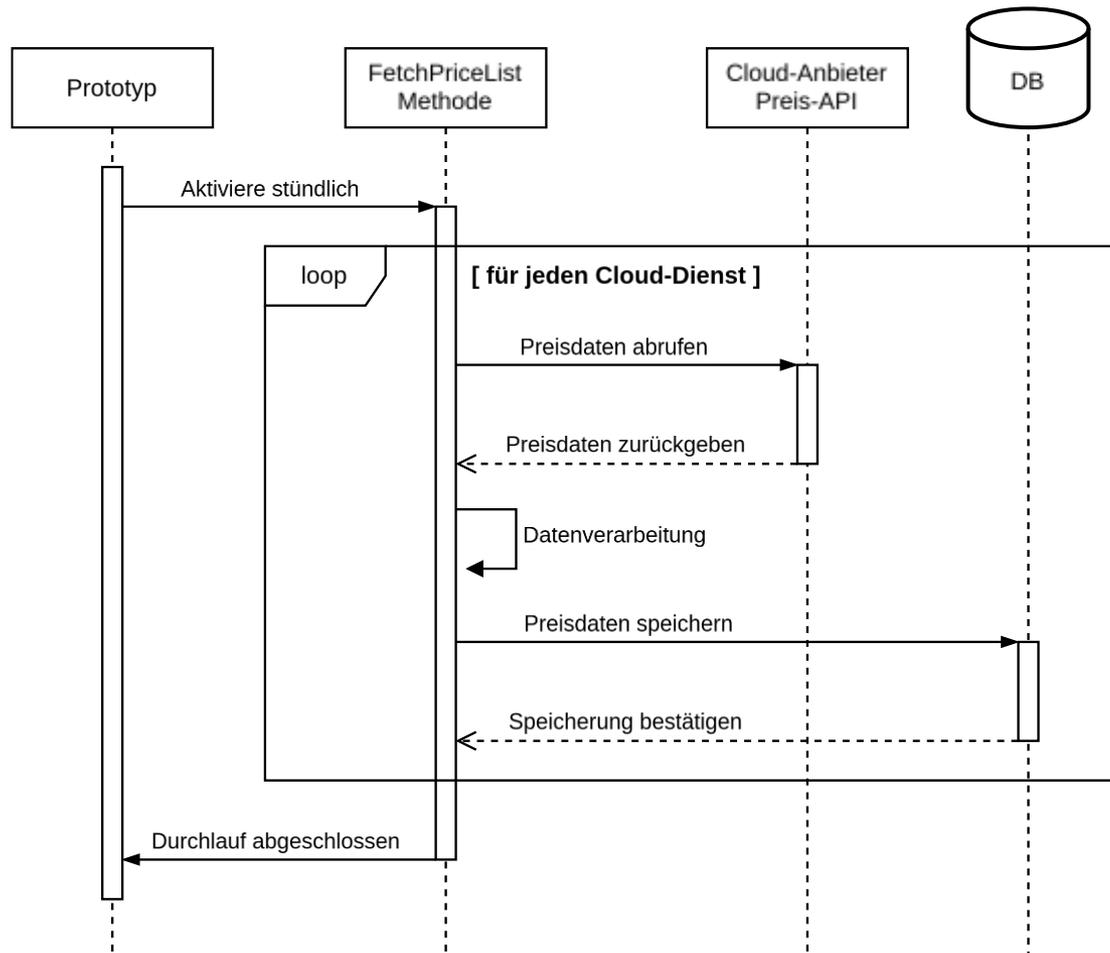


# Design

# Prototyp – Anforderungen

- Sammlung der Preisinformationen
- Kommunikation mit den Preis-APIs
  - Drei Cloud-Anbieter
  - Je drei Cloud-Dienste
    - Computing-, Speicher-, Datenbankdienst
- Automatisierte Abfrage
  - Stündlich
- Verarbeitung in einheitliche Struktur
- Abspeicherung der Preisinformationen
- Tabellarische Darstellung

# Prototyp – Design



# Implementierung

- Gewählte Preis-APIs
  - AWS Price List Bulk API
  - Azure Retail Prices REST API
  - Google Cloud Billing API
- Kommunikation über REST
  - GET-Request
    - Basis-URL
    - Filtervariablen
  - HTTP-Response mit Status-Code 200
    - Body enthält Preisliste
    - JSON-Struktur



- Golang
  - Grund
    - Performanz
    - Automatisierung
    - Parallelität
    - Datenbankkommunikation
- } Go-Routine
- } GORM

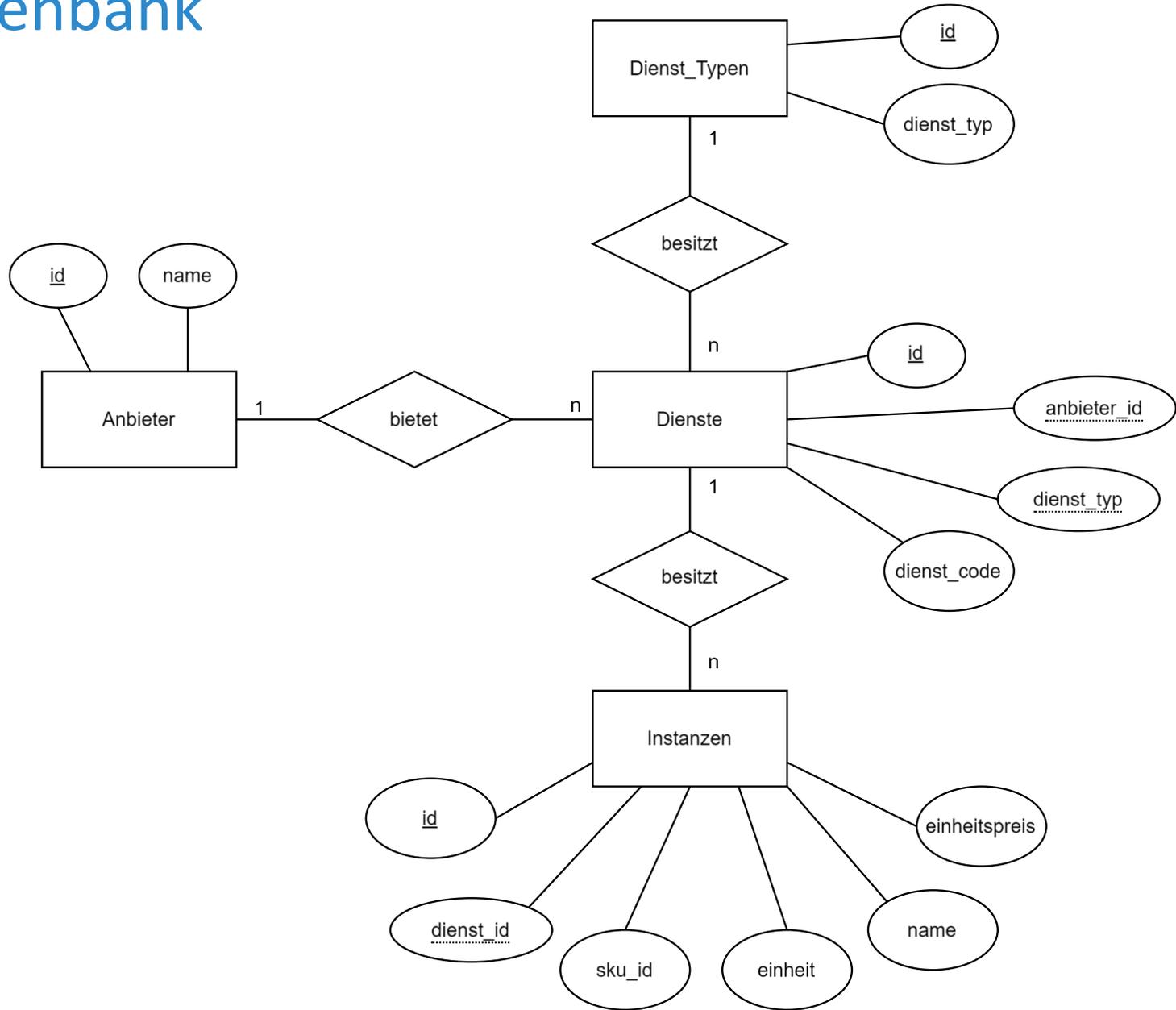
## Go-Struktur

```
type Instances struct {  
    ID          uint  
    ServiceID  uint  
    SkuID       string  
    Name        string  
    UnitPrice   string  
    Unit        float64  
}
```

## Preisliste

```
"Items": [  
  { [+]  
    {  
      "skuId": "DZH318Z0BPVM/00G1",  
      "productName": "VM Dv2 Series",  
      "unitPrice": 0.0136,  
      "unitOfMeasure": "1 Hour",  
    },  
  ],  
]
```

# Datenbank



# Validierung

- Aktualisierungsdauer abhängig
  - Rechenleistung
  - Internetgeschwindigkeit

## Desktop-Computer mit LAN-Anschluss

Anbieter	Start	Ende	Dauer in Sek
Amazon Web Services	21:34:33	21:35:20	47
Microsoft Azure	21:35:20	21:35:31	11
Google Cloud Platform	21:35:31	21:35:39	8

## Laptop mit WLAN-Verbindung

Anbieter	Start	Ende	Dauer in Sek
Amazon Web Services	21:34:30	21:36:34	94
Microsoft Azure	21:36:34	21:37:24	50
Google Cloud Platform	21:37:24	21:37:47	20

# 24-Stunden Testlauf

- Schwierige Einschätzung
- Mögliche Ursachen
  - AWS Price List Bulk API → Offline
    - Verbindungsabbruch
    - Wartung
    - Aktualisierung

Iteration	Start	Ende	Dauer
1	16:49:26	16:50:31	65
2	17:49:26	17:50:31	65
3	18:49:26	18:50:31	65
4	19:49:26	19:50:26	60
5	20:49:26	20:50:26	60
6	21:49:28	21:50:29	61
7	22:49:28	22:50:29	61
8	23:49:28	23:50:04	36
9	00:49:28	00:50:04	36
10	01:49:28	01:50:04	36
11	02:49:28	02:50:05	37
12	03:49:28	03:50:04	36
13	04:49:28	04:50:31	63
14	05:49:28	05:50:04	36
15	06:49:28	06:50:05	37
16	07:49:28	07:50:04	36
17	08:49:28	08:50:03	35
18	09:49:28	09:50:03	35
19	10:49:28	10:50:04	36
20	11:49:28	11:50:04	36
21	12:49:28	12:50:05	37
22	13:49:28	13:50:04	36
23	14:49:28	14:50:04	36
24	15:49:28	15:50:03	35

# Tabellarische Darstellung

Provider:  Service:  SKU:  Resource Name:

**Service** **SKU** **Resource Name** **Price per unit** **Unit**

Amazon Web Services	Amazon RDS	DR7GUGTJ6VZKFJ76	EUC1-InstanceUsage:db.m6gd.12xl	\$5.3740000000	Hrs
Amazon Web Services	Amazon RDS	EUWKMCTANKSGX74W	EUC1-InstanceUsage:db.r5.2xl	\$1.1600000000	Hrs
Amazon Web Services	Amazon RDS	ZTQXEC247USAQP8N	EUC1-Multi-AZUsage:db.r7g.12xl	\$13.9740000000	Hrs

Provider:  Service:  SKU:  Resource Name:

Provider	Service	SKU	Resource Name	Price per unit	Unit
Microsoft Azure	Azure Virtual Machines	DZH318Z0K9JF/020X	Virtual Machines Basv2 Series Windows	\$0.382000000	1 Hour

Page 1

# Fazit und Ausblick

## Sammlung der Preisinformationen

- Drei public Cloud-Anbieter, je drei Dienste
  - Vereinheitlichung der Daten
  - Speicherung der Daten
  - Automatisierung des Prozesses
  - Zusätzlich: Tabellarische Darstellung der Daten
  - Grundlage für die Preisanalyse geschaffen
- ❖ Eine Lösungsmöglichkeit zur automatisierten Abfrage der Preis-APIs realisiert

- Analyse
  - Export der Preisinformationen in andere Dateiformate
  - Zeitlicher Preisverlauf
  - Detaillierte Datenstruktur
    - Verweis auf identische Ressourcen
    - Weitere Filtermöglichkeiten
- Sky Computing
  - Automatisierte Sammlung der Kosten
  - Analyse zwischen momentanen Kosten und aktuellen Preisen
- ❖ Automatisierte Auswahl der Ressourcen
- ❖ Automatisierte Verlagerung der Workloads

# Live-Demo

Vielen Dank für Ihre  
Aufmerksamkeit!