

DEKLARATIVE APIS

EIN MÄCHTIGES MITTEL IM API DESIGN

WER BIN ICH?

Stefan Tomm

seit 15 Jahren Software-Entwickler

seit > 5 Jahren Tech Lead bei meshcloud 

seit 15 Jahren Software-Entwickler

WAS ERWARTET EUCH HEUTE?

- Was ist eine Deklarative API?
- Beispiel einer Deklarativen API
- Deklarative vs Imperative APIs?
- Demo

WAS IST EINE DEKLARATIVE API?

WIE WIRD EINE API KLASSISCH UMGESETZT?

imperativ mit feingranularen Commands, z.B.

- Erzeuge eine User-Gruppe
- Update eine User-Gruppe
- Lösche eine User-Gruppe
- Update die Mitglieder einer User-Gruppe

WAS IST NUN DER UNTERSCHIED ZU EINER DEKLARATIVEN API?

Deklarative Systeme sind desired-state Systeme, d.h.

- du stellst dem System den vollständigen, gewünschten State zur Verfügung
- das System stellt sicher, dass alle Creations, Updates und Deletions durchgeführt werden

BEKANNTE BEISPIELE FÜR DEKLARATIVE SYSTEME:

- Terraform
- Kubernetes

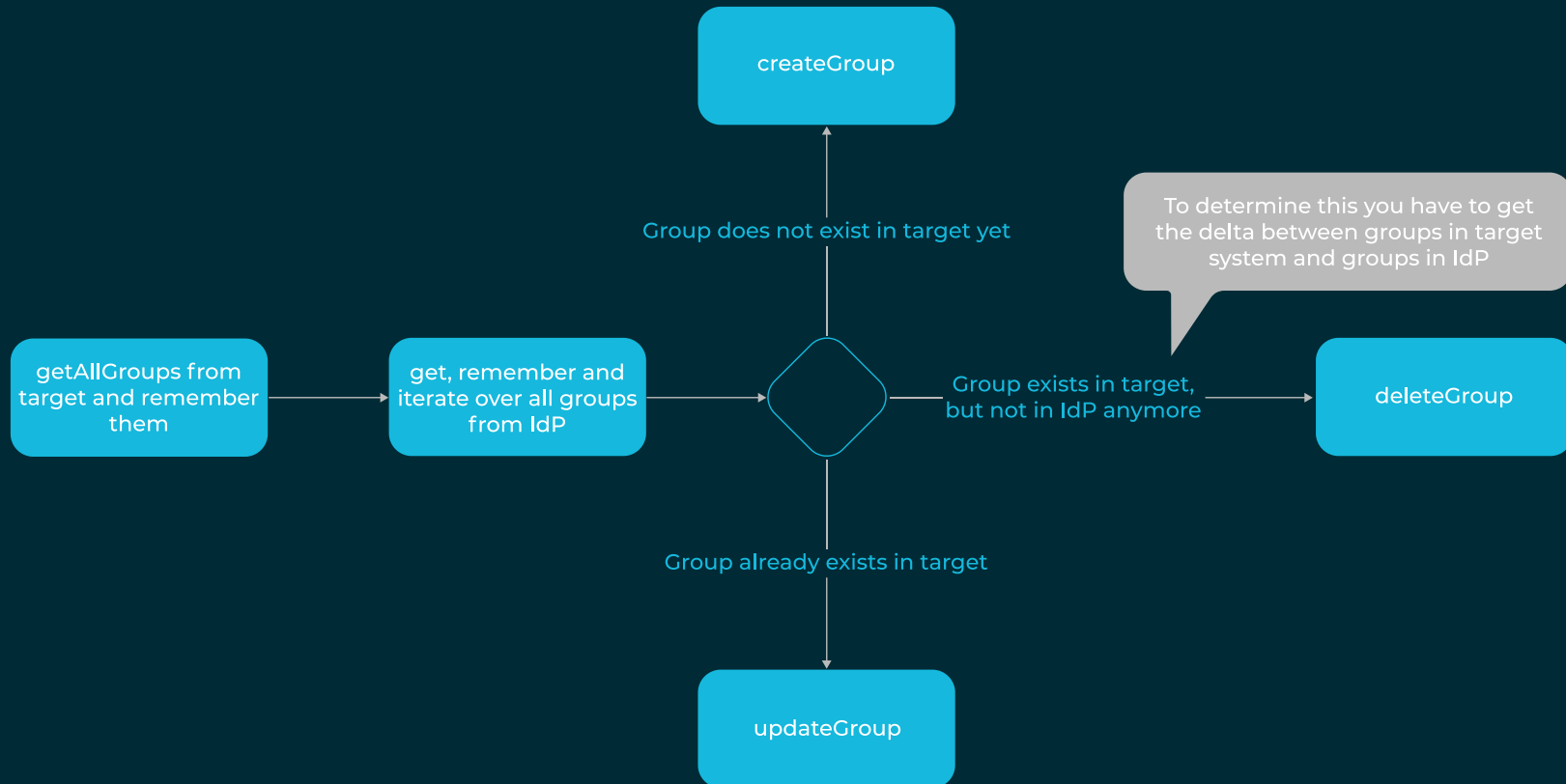
BEISPIEL EINER DEKLARATIVEN API

USER-GRUPPEN SYNCHRONISATION

Gruppen aus zentralem IdP synchronisieren

```
Group: {  
  id: "123-456"  
  displayName: "My Group"  
  members: ["uid1", "uid2"]  
}
```

CLIENT MIT EINER IMPERATIVEN API



REQUEST BEI EINER DEKLARATIVEN API

```
[ // alle User Groups in einem Array
  {
    id: "123-456"
    groupName: "developers",
    members: ["dev1", "dev2"]
  }, {
    id: "456-789"
    groupName: "managers",
    members: ["manager1"]
  }, {
    id: "789-132"
    groupName: "operators",
    members: ["op1"]
  }
]
```

CLIENT MIT EINER DEKLARATIVEN API



UPDATE IN DECLARATIVE API

```
[
  {
    id: "123-456"
    groupName: "devops", // renamed "developers" -> "devops"
    members: ["dev1", "ops1"] // dev2 removed, ops1 added
  },
  {
    id: "456-789"
    groupName: "managers",
    members: ["manager1", "dev2"] // dev2 added
  }
  // id 789-132 deleted
]
```

ABHÄNGIGKEITEN ZWISCHEN OBJEKTEN

```
[
  { // make sure in backend to create users first
    kind: "user"
    username: "dev1"
    lastName: "Schmitt"
  },
  {
    kind: "group"
    id: "123-456"
    groupName: "devops",
    members: ["dev1"]
  }
]
```

HERAUSFORDERUNGEN BEI MEHREREN QUELLEN?

- Objekte werden gegenseitig überschrieben
- Objekte die nicht in allen Quellen vorliegen werden immer wieder gelöscht

BEISPIEL MEHRERE QUELLEN

Source 1

```
[
  {
    kind: "user"
    username: "dev1"
    lastName: "Schmitt"
  },
  {
    kind: "user"
    username: "ops1"
    lastName: "Bauer"
  }
]
```

Source 2

```
[
  {
    kind: "user",
    username: "dev1",
    // benennt dev1 um
    lastName: "Meyer"
  }
  // löscht ops1
]
```


WIE BEHANDELT MAN MEHRERE QUELLEN?

- eine Art Container einführen
- Container bei jedem Call als Param mitgeben
- PUT `http://my-host/objects?container=idp1`

BEISPIEL MIT CONTAINER

Source 1

PUT http://my-host/objects?
container=source1

```
[
  {
    kind: "user"
    username: "dev1"
    lastName: "Schmitt"
  },
  {
    kind: "user"
    username: "ops1"
    lastName: "Bauer"
  }
]
```

Source 2

PUT http://my-host/objects?
container=source2

```
[
  { // Conflict - 409
    kind: "user",
    username: "dev1",
    lastName: "Meyer"
  }
  // ops1 unberührt
]
```

DER AS-CODE VORTEIL

- Desired State in VCS speichern & daraus ausführen
- Historie in VCS
- Kooperatives Arbeiten am Desired State

DEKLARATIVE VS IMPERATIVE APIS?

VORTEILE IMPERATIVE APIS

- Fein granularere Steuerung auf Client Seite
- Fire & Forget Szenarien (kein Sync nötig)
- Einfacher um nur einzelne Attribute zu updaten (v.a. wenn verschiedene Quellen das gleiche Objekt manipulieren)

VORTEILE DEKLARATIVE APIS

- Daten Synchronisationen, ein anderes System gibt Desired State vor oder desired-state wird in VCS abgelegt
- Komplexe State Handling Logik nur einmal zentral im Backend umsetzen

DEKLARATIV UND IMPERATIV ERGÄNZT SICH

- APIs zu einer Anwendung haben meistens Use-Cases für beide Ansätze
- Deshalb ergänzen sich beide Ansätze um die Beste Lösung für jeden Use-Case zu finden

DEMO TIME

RESSOURCEN

Slides & Sample Application

<https://github.com/meshcloud/declarative-api-example>

Blog Post

<https://www.meshcloud.io/de/blog/should-i-provide-a-declarative-api-you-probably-should/>

