

# Middleware – Cloud Computing

Koordinierungsdienste (Zusatz)  
Nutzung von ZooKeeper zur Anführerwahl

---

Wintersemester 2022/23

Tobias Distler

Friedrich-Alexander-Universität Erlangen-Nürnberg  
Lehrstuhl Informatik 4 (Verteilte Systeme und Betriebssysteme)



Lehrstuhl für Verteilte Systeme  
und Betriebssysteme



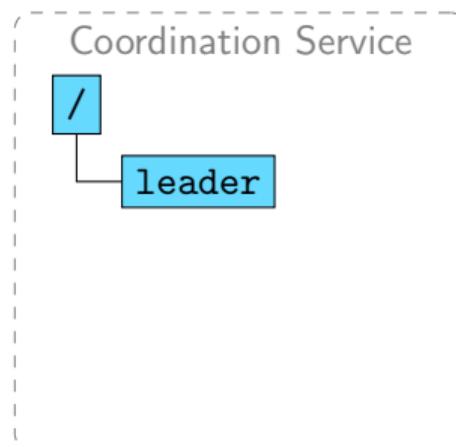
FRIEDRICH-ALEXANDER  
UNIVERSITÄT  
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

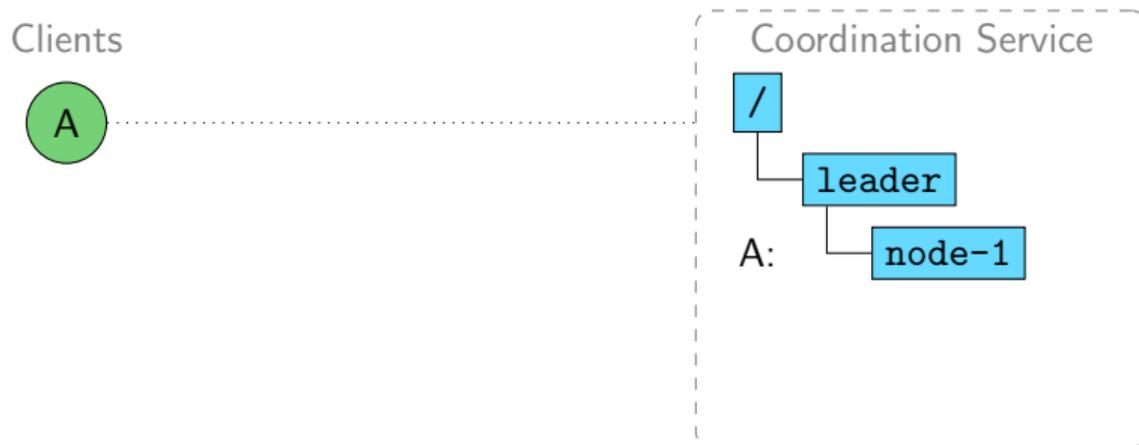
- Problemstellung
  - In einer **Gruppe von Clients** soll ein Anführer gewählt werden
  - Bei Ausfall des Anführers muss ein neuer Anführer bestimmt werden
- Lösungsansatz
  - ZooKeeper-Mechanismus: Dienstunterstützte Knotennamen („Sequenzielle Knoten“)
    - Automatisch generierte **Sequenznummer als Teil des Knotennamens**
    - Client spezifiziert beim Anlegen des Knotens den Namenspräfix
  - Erstellen eines „Verzeichnisknotens“ `/leader` für die Gruppe
  - Vorgehensweise beim Hinzukommen eines neuen Clients
    - Erstellen eines **flüchtigen Kindknotens** `/leader/node-<Sequenznummer>`
    - Suche nach Kindknoten  $k_s$  mit nächstkleinerer Sequenznummer
    - Existiert kein Kindknoten mit kleinerer Sequenznummer → Client ist Anführer
    - Sonst: Client ist kein Anführer → Setzen eines Watch auf den Kindknoten  $k_s$
  - Bei Knotenausfall
    - Automatische Löschung des zugehörigen flüchtigen Knotens
    - **Maximal ein Client** wird per Watch über den Ausfall benachrichtigt

- Start von Client A

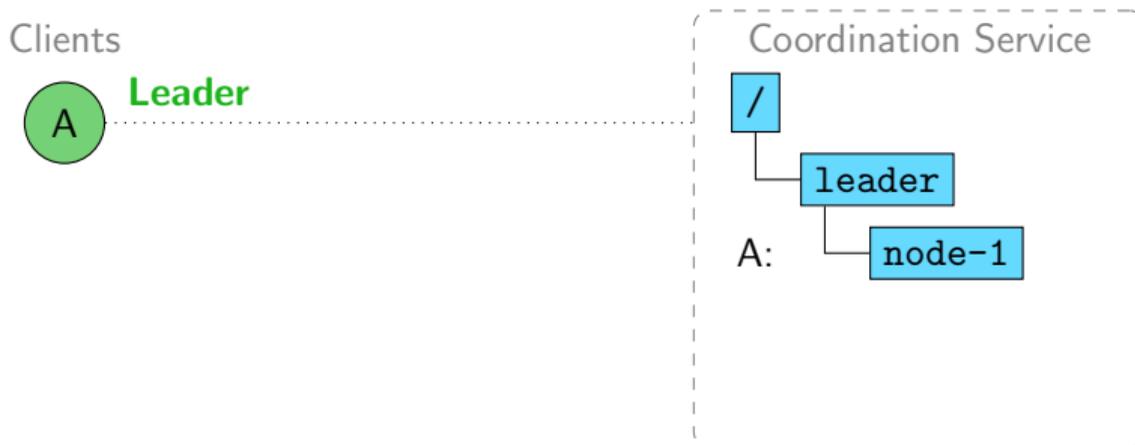
Clients



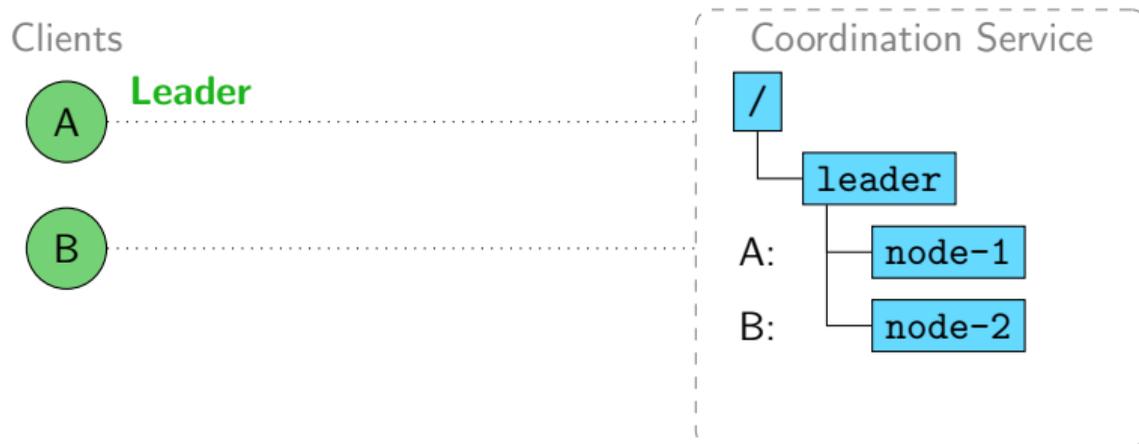
- Start von Client A → Anlegen von node-1



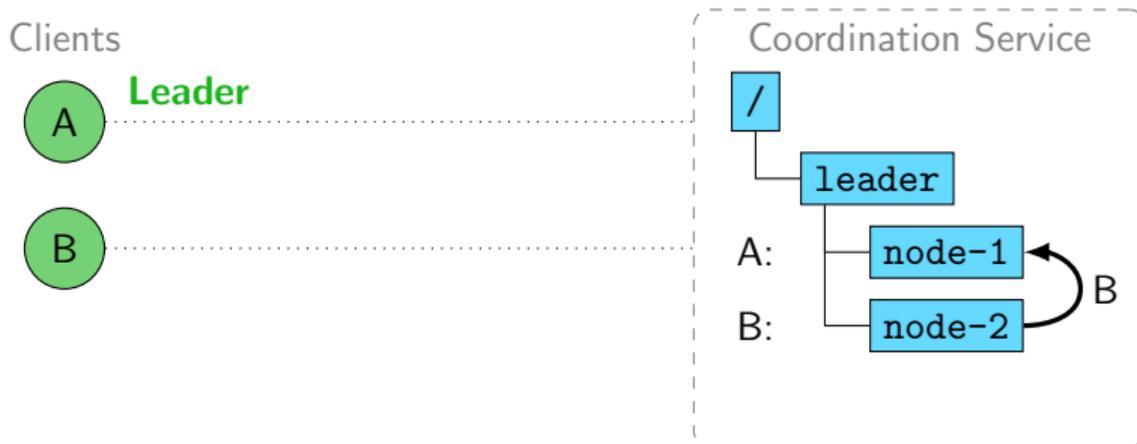
- Start von Client A → Anlegen von node-1 → **Client A wird Anführer**



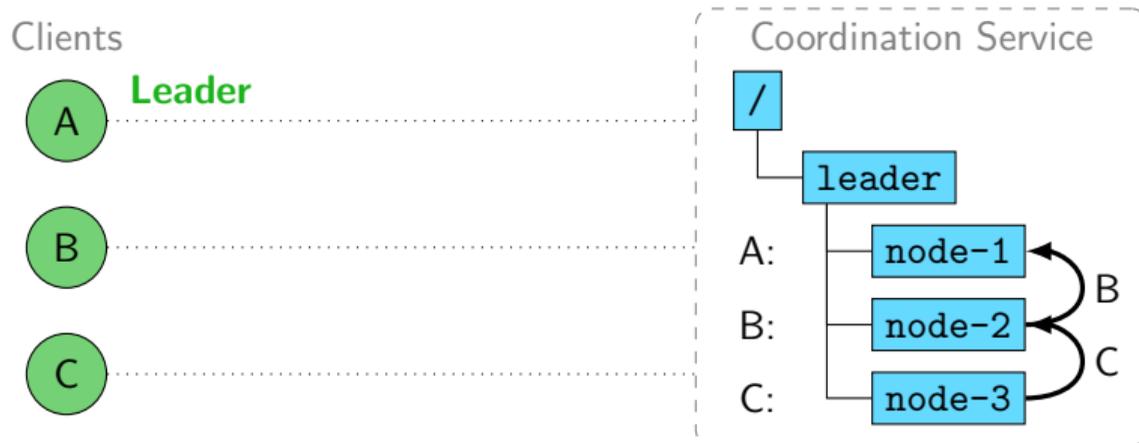
- Start von Client A → Anlegen von node-1 → **Client A wird Anführer**
- Start von Client B → Anlegen von node-2



- Start von Client A → Anlegen von node-1 → **Client A wird Anführer**
- Start von Client B → Anlegen von node-2 → Watch auf node-1



- Start von Client A → Anlegen von node-1 → **Client A wird Anführer**
- Start von Client B → Anlegen von node-2 → Watch auf node-1
- Start von Client C → Anlegen von node-3 → Watch auf node-2



- Ausfall von Client A

Clients



Coordination Service



leader

A:

node-1

B:

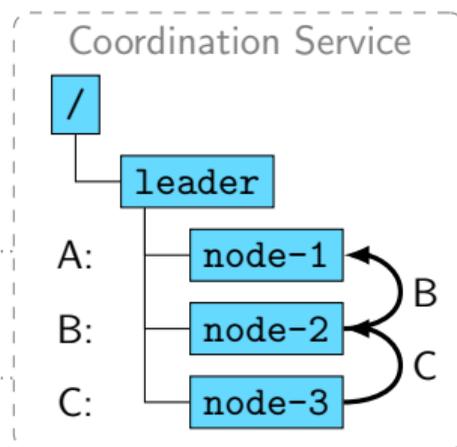
node-2

C:

node-3

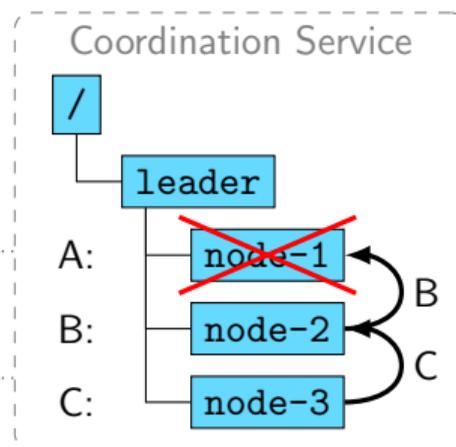
B

C

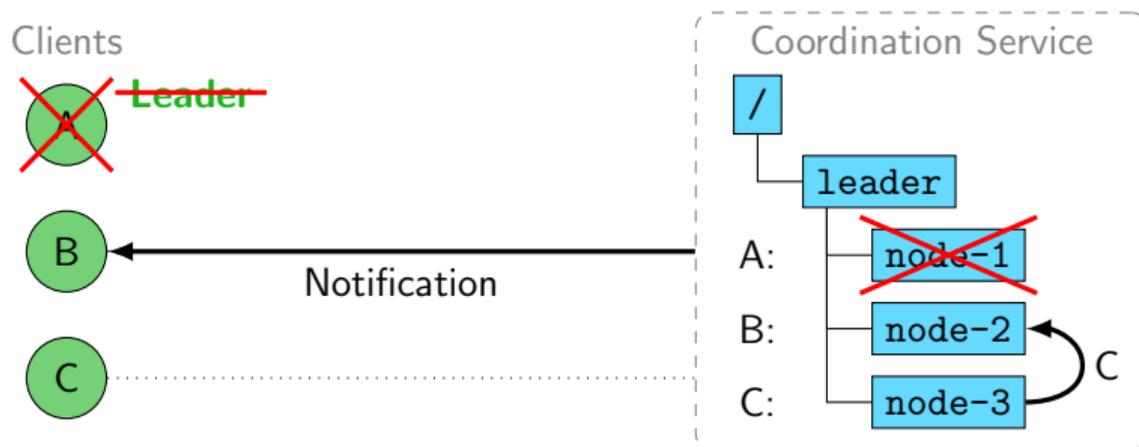


- Ausfall von Client A → Entfernen von node-1

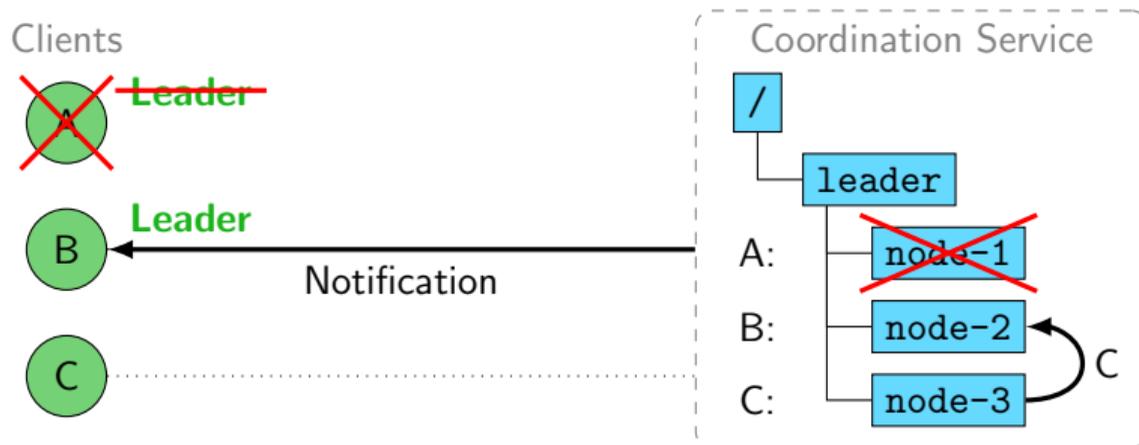
Clients



- Ausfall von Client A → Entfernen von node-1
- Watch von Client B löst aus



- Ausfall von Client A → Entfernen von node-1
- Watch von Client B löst aus → **Client B wird Anführer**



- Ausfall von Client A → Entfernen von node-1
- Watch von Client B löst aus → **Client B wird Anführer**
- Kein Mehraufwand für Client C

