



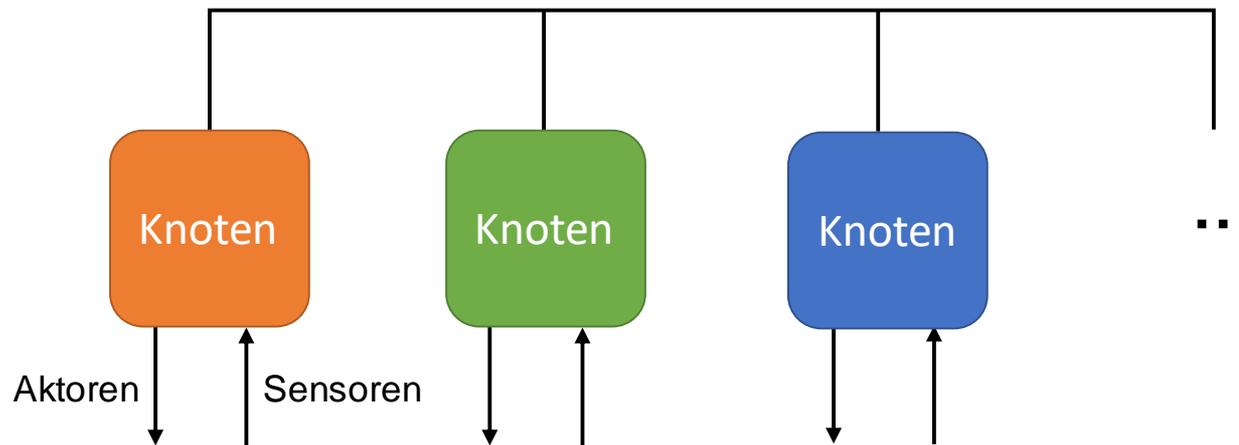
TECHNISCHE UNIVERSITÄT
CHEMNITZ

Mobilität und Verteiltheit in Barrelfish

Mobilität und Verteiltheit in Barrelfish

Motivation

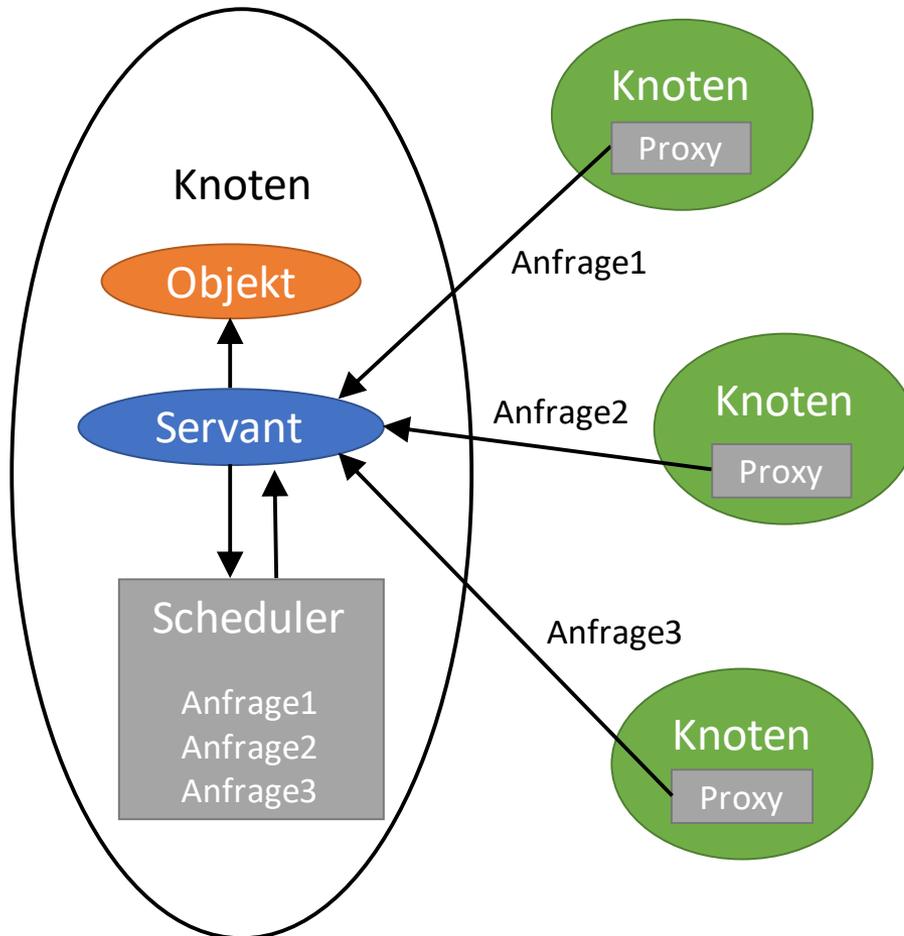
- Verteiltheit, Mobilität, Diversität und Echtzeit
- Explizite Kommunikation fehleranfällig und kompliziert
- Verteilte Aktive Objekte als Lösungsansatz (siehe Frühjahrstreffen)



Problemstellung

- Vergleich von Barrelfish mit verteilten aktiven Objekten
- Konzepte überschneiden sich teilweise
- Hilft Barrelfish bei der Implementierung?
- Erarbeitung Problemstellung für Masterarbeit

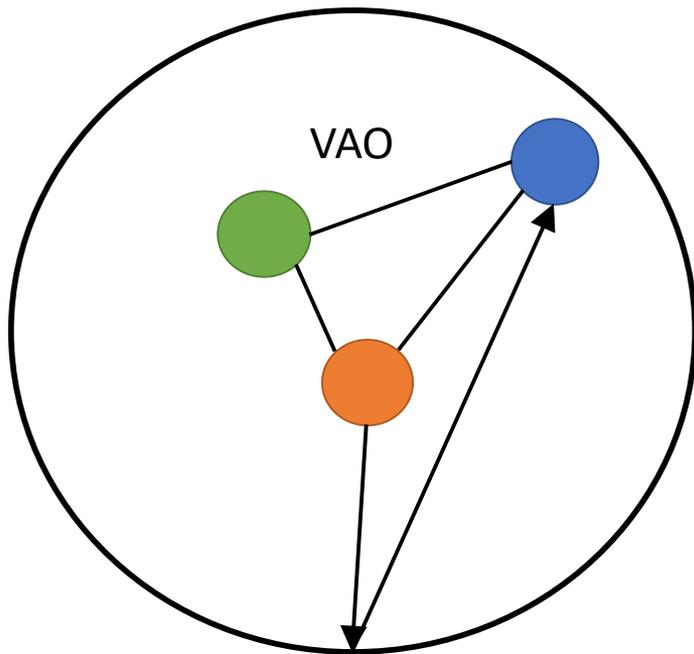
Aktive Objekte



- Mehrere Knoten mit entfernten Objekten
- Trennung von Methodenruf und -ausführung
- Abstraktion von komplizierter Kommunikation
- Nebenläufigkeit und zeitlich abgestimmter Zugriff

Quelle: Greg R. Lavender and Douglas C. Schmidt. Active object - an object behavioral pattern for concurrent programming. 1996.

Verteilte Aktive Objekte – Grundgedanke



- Gleichberechtigte Knoten
- Räumlich verteilte Ausführung des Objektes

Quelle: Daniel Graff, Jan Richling, Tammo M. Stupp, and Matthias Werner. Distributed Active Objects - A Systemic Approach to Distributed Mobile Applications, 2011.

Verteilte Aktive Objekte – Rückblick

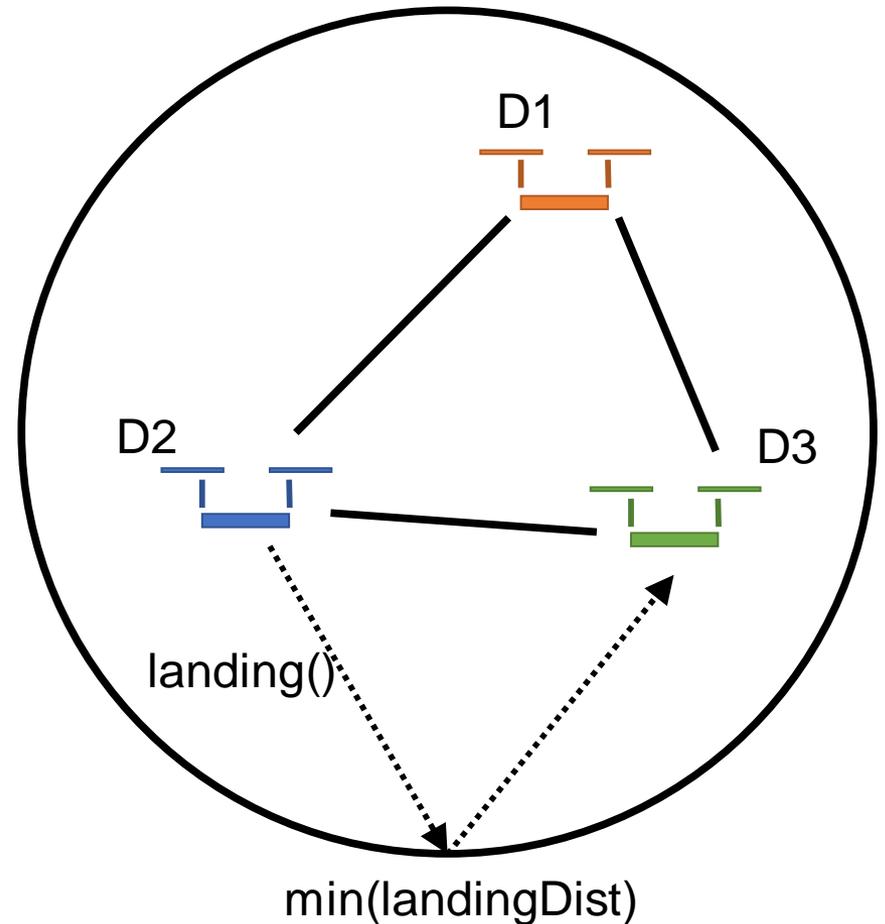
```

class DroneSwarm {
    Float landingDist;
    (Int, Int) landingPos;

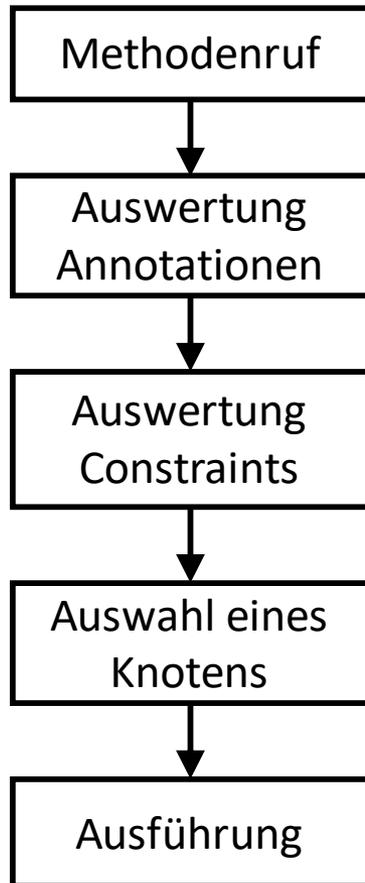
    @Require( min(landingDist) )
    landing() { //lande }

    eventLoop() {
        // ...
        this.landing()
    }

    // ...
}
  
```

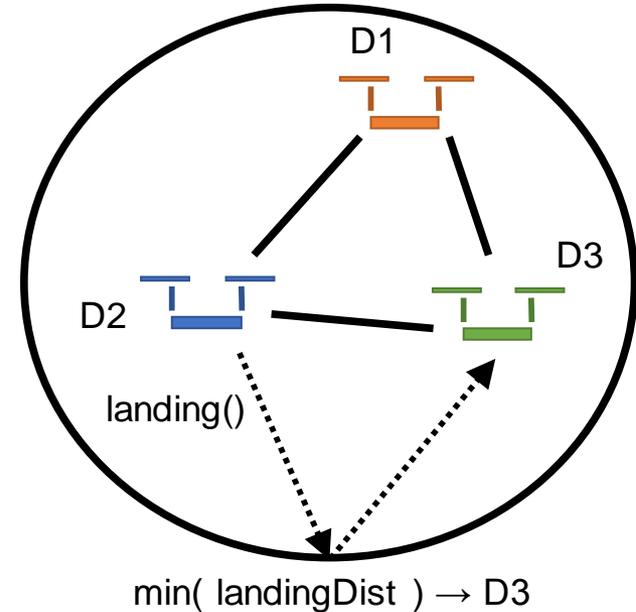


Verteilte Aktive Objekte – Rückblick



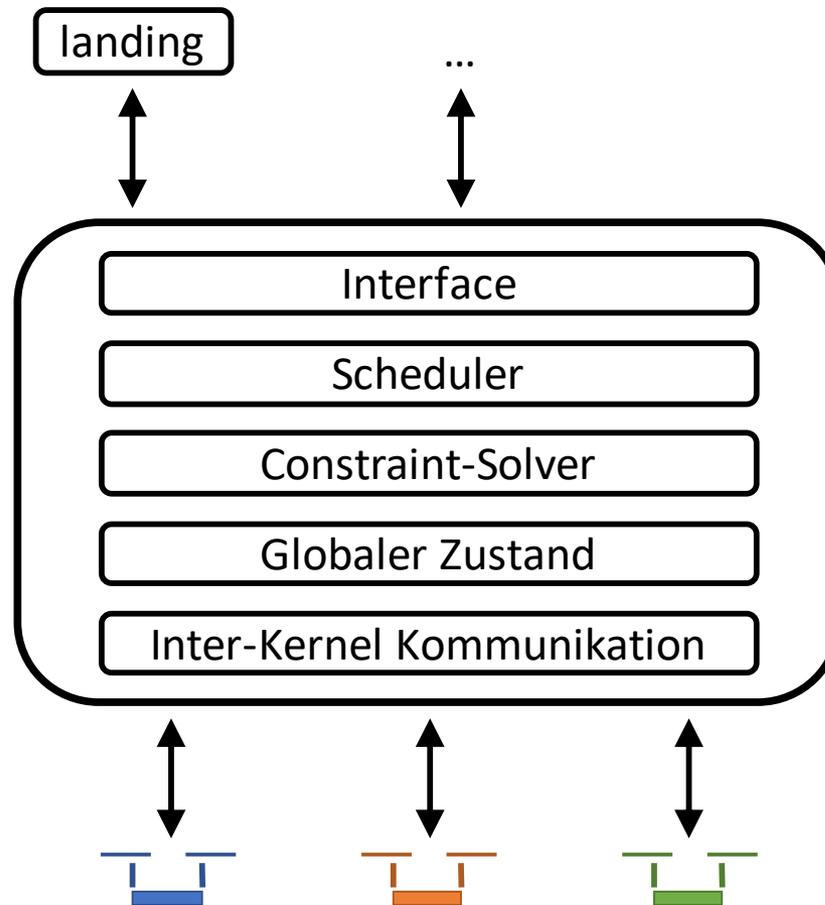
@Require(...)

min(landingDist)



Attribut	Wert	Knoten
landingPos	(3, 4)	{ D1, D2 }
	(8, 1)	{ D3 }
landingDist	4.3	{ D1 }
	3.1	{ D2 }
	1.5	{ D3 }

Verteilte Aktive Objekte – Rückblick

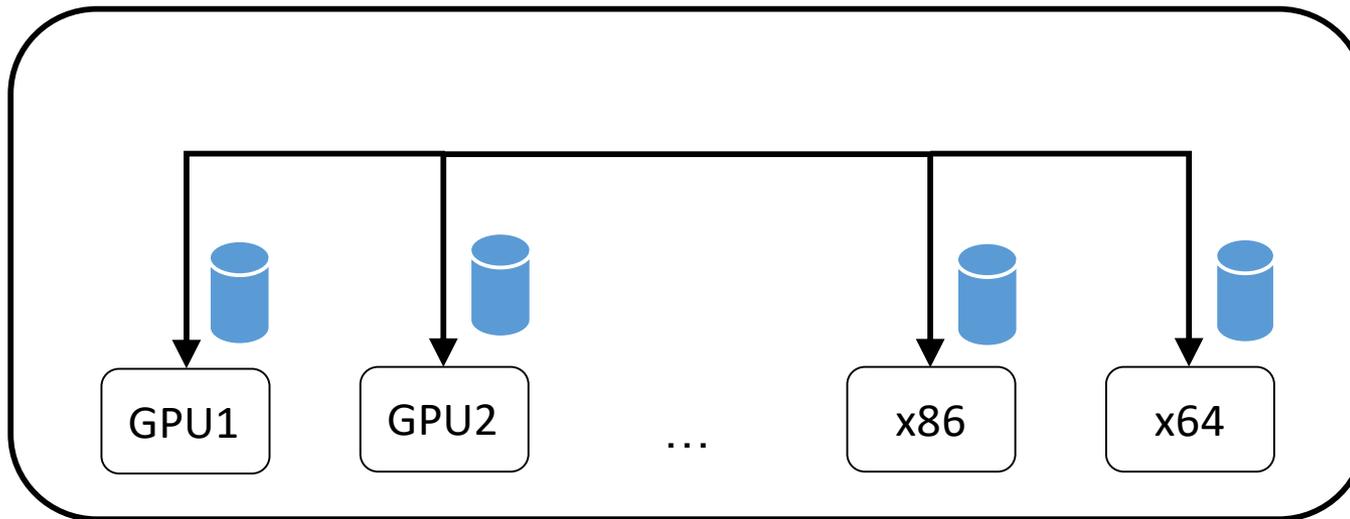


Verteilte Aktive Objekte – Abbildung auf Barrelfish

Verteilte Aktive Objekte	Barrelfish
Heterogenität	?
Netzwerkweite Kommunikation	?
Globaler Zustand	?
Constraint-Solver	?
Verteilter Kontrollfluss	?
Programmiermodell	?

Barrelfish - Konzept

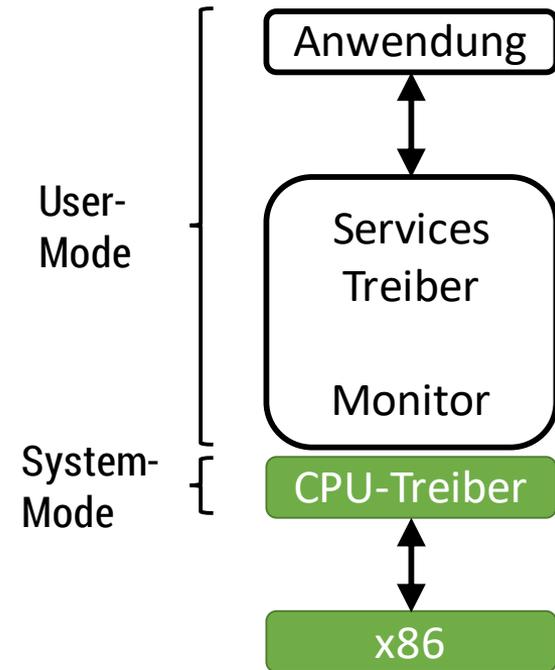
- Lokaler Rechner als verteiltes System
- Immer mehr heterogene Ausführungseinheiten
- Replikation und Kommunikation statt geteiltem Speicher



Quelle: Andrew Baumann, Paul Barham, Rebecca Isaacs, and Tim Harris. The Multikernel: A new OS architecture for scalable multicore systems. In 22nd Symposium on Operating Systems Principles. Association for Computing Machinery, Inc., October 2009.

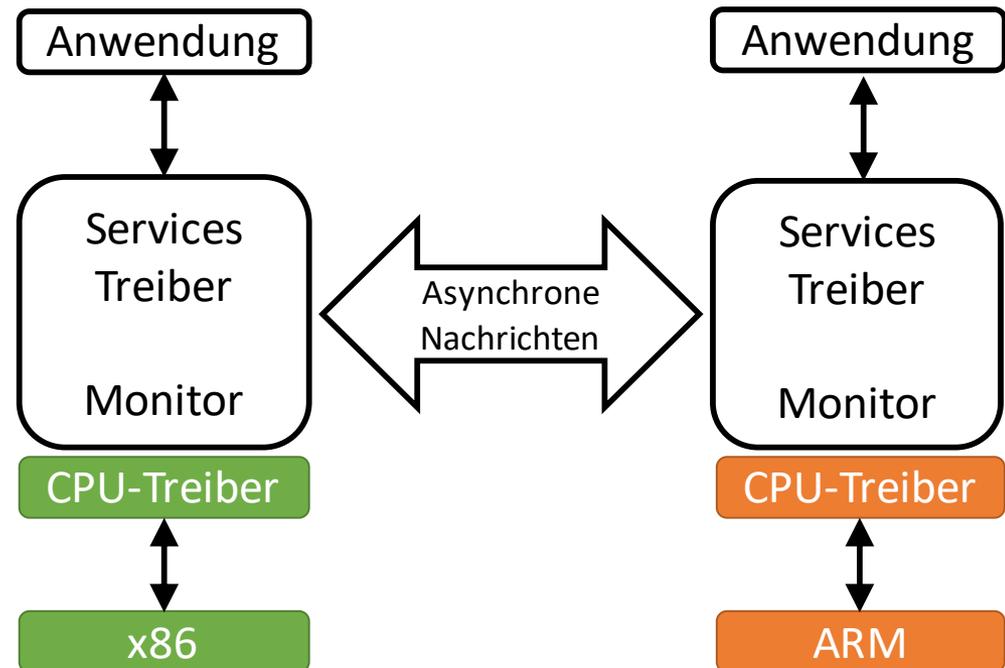
Barrelfish - Architektur

- Heterogene CPU-Treiber
- Alle anderen Prozesse im Nutzermodus
- Monitor-Prozess zur Koordinierung und Konsistenzerhaltung



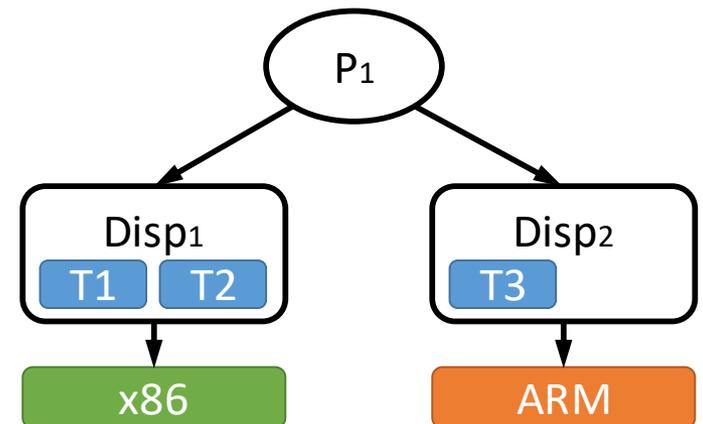
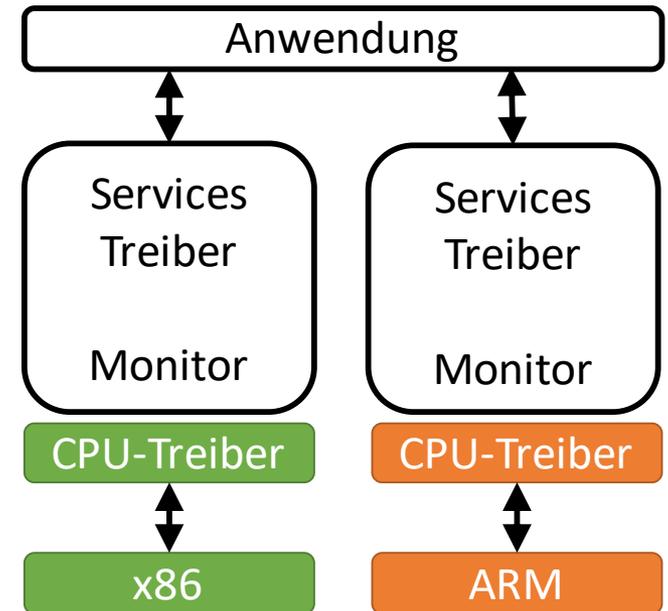
Barrelfish - Architektur

- Kommunikation über Interconnect-Treiber
- Einheitliche Interfaces erlauben Austausch
- Monitor wählt optimalen Mechanismus



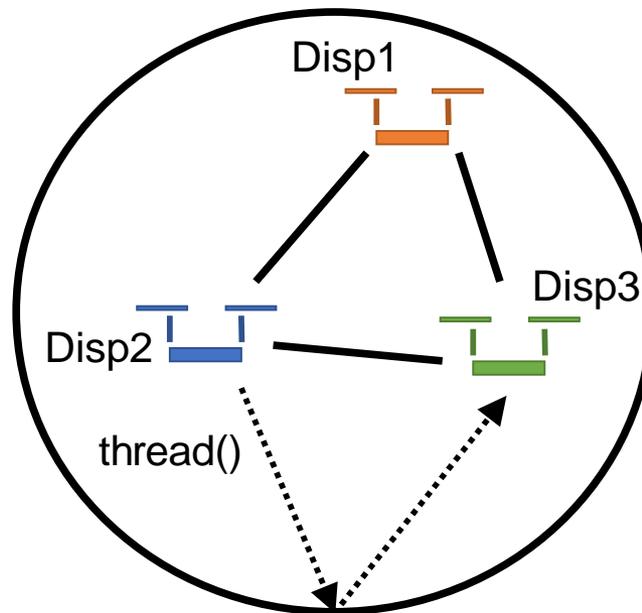
Barrelfish - Architektur

- Verteilte Anwendungen
- Prozesse aus mehreren Dispatchern
- Erzeugung von Threads auf eigenem oder entferntem Dispatcher möglich
- Threads können gemeinsamen Adressraum nutzen



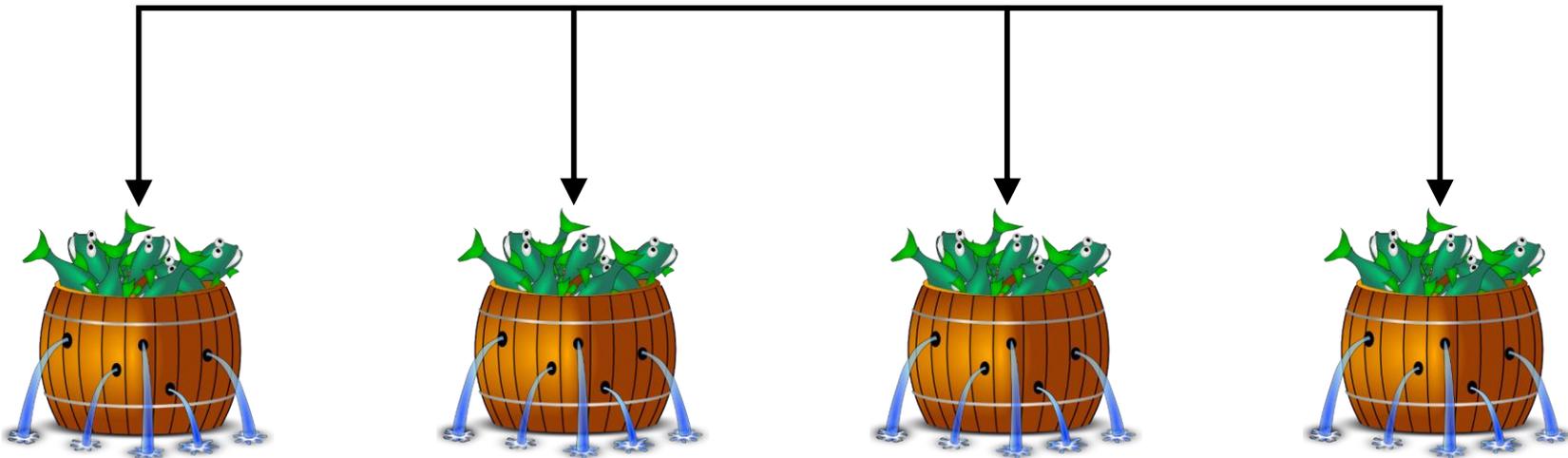
Barrelfish – Verteilte Aktive Objekte

- VAO zur Unterstützung verteilter Anwendungen in Barrelfish
- Eine Instanz pro Ausführungseinheit
- Impliziter Sprung des Kontrollflusses durch Nachrichtenaustausch



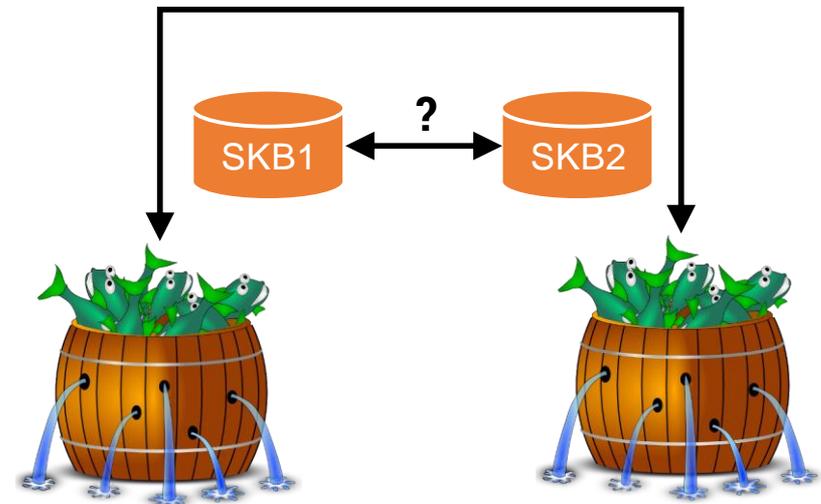
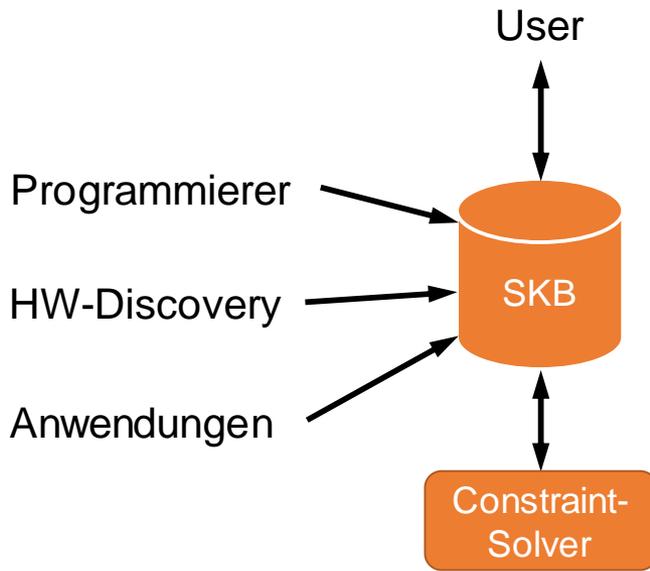
Barrelfish – Echte Verteiltheit

- Mehrere Barrelfish-Instanzen
- Bekanntmachung der Knoten untereinander
- Kommunikation durch neue Interconnect-Treiber



Barrelfish – Heterogenität und Mobilität

- Unterschiedliche CPU-Treiber
- SKB für Verwaltung von Informationen (z. B. Positionen)
- Problem: Konsistenz



Verteilte Aktive Objekte – Fazit

Verteilte Aktive Objekte	Barrelfish
Heterogenität	Heterogene CPU-Treiber
Netzwerkweite Kommunikation	Interconnect-Treiber
Globaler Zustand	System Knowledge Base
Constraint-Solver	System Knowledge Base
Verteilter Kontrollfluss	TODO
Programmiermodell	TODO

Verteilte Aktive Objekte – Aufgabenstellung für Masterarbeit

- Herstellung echter Verteiltheit
- Objekt-Deployment (Instanzengeneration)
- Koordinierung der verschiedenen Instanzen
- Gewährung von Konsistenz
- Unterstützung Programmiermodell

Quellen

- Daniel Graff, Jan Richling, Tammo M. Stupp, and Matthias Werner. 8th iee international conference and workshops on engineering of autonomic and autonomous systems. In Distributed Active Objects A Systemic Approach to Distributed Mobile Applications, 2011.
- Andrew Baumann, Paul Barham, Rebecca Isaacs, and Tim Harris. The Multikernel: A new OS architecture for scalable multicore systems. In 22nd Symposium on Operating Systems Principles. Association for Computing Machinery, Inc., October 2009.
- Andrew Baumann, Paul Barham, Tim Harris, and Rebecca Isaacs. Embracing diversity in the Barrelfish manycore operating system. In Proceedings of the Workshop on Managed Many-Core Systems. Association for Computing Machinery, Inc., June 2008.
- Roscoe Timothy and Rik Farrow. The Barrelsh Multikernel: an interview with Timothy Roscoe. <https://www.usenix.org/system/files/login/articles/1906-roscoe.pdf>, 2010.
- Greg R. Lavender and Douglas C. Schmidt. Active object - an object behavioral pattern for concurrent programming. 1996.