

Macoun

[co coa:ding]

Nebenher erledigt

Amin Negm-Awad

Ablauf

- Einleitung
- Nebenläufigkeit
- Probleme
- Runloops

Einleitung

Einleitung

- Wer bin ich?
- Fußballergebnisse

Nebenläufigkeit

Nebenläufigkeit

- Arten
- Systeme

Arten

- Physikalisch
- Threads
- Kooperation
- Kontrollfluss

Hardware

- Physikalisch: Parallel
- Programmierer: Parallel
- Nutzer: Parallel

Threads

- Physikalisch: Potentiell parallel
- Programmierer: Parallel
- Nutzer: Parallel

Kooperation

- Physikalisch: Seriell
- Programmierer: Seriell
- Nutzer: Parallel

Kontrollfluss

- Physikalisch: Seriell
- Programmierer: Seriell
- Nutzer: Seriell

Vergleich

- Vorteil
- Nachteil
- Schlussfolgerung

Vorteil

Typ	Programmierer	Nutzer
Hardware	Parallel	Parallel
Threads	Parallel	Parallel
Kooperation	Seriell	Parallel
Kontrollfuss	Seriell	Seriell

Nachteil

- Coolnessfaktor
- Ausnutzung Kerne
- Lange Einzelschritte

Schlussfolgerung

- Kooperation: So viel wie möglich, so wenig wie nötig
- Threads: So wenig wie möglich, so viel wie nötig

Systeme

- Tasks
- Threads
- Queues
- Runloop

Tasks

- Parallele Kontrollflüsse
- Wenig geteilte Ressourcen
- Eigene Schnittstelle

Threads

- Parallele Kontrollflüsse
- Geteilte Ressourcen
- Synchronisationsaufwand

Queues

- Parallele Kontrollflüsse
- Geteilte Ressourcen
- Zerteilungsaufwand

Runloops

- Serieller Kontrollfluss
- Geteilter Ressourcen
- Kaum Aufwand

Problem

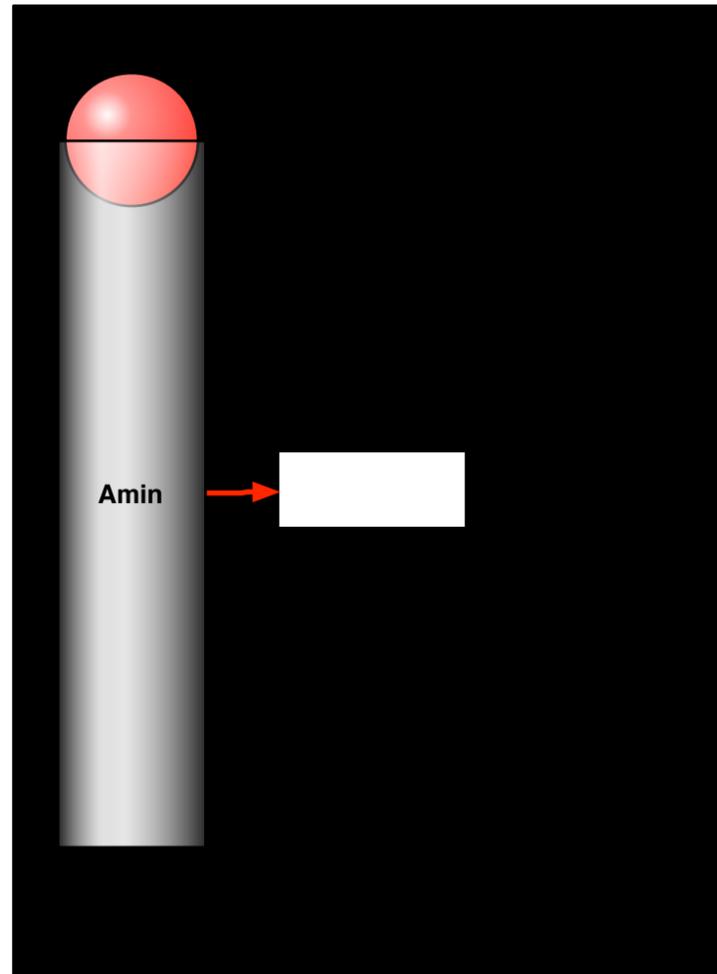
Race Conditions

- Problem
- Lösung mit Threads
- Lösung mit Queues

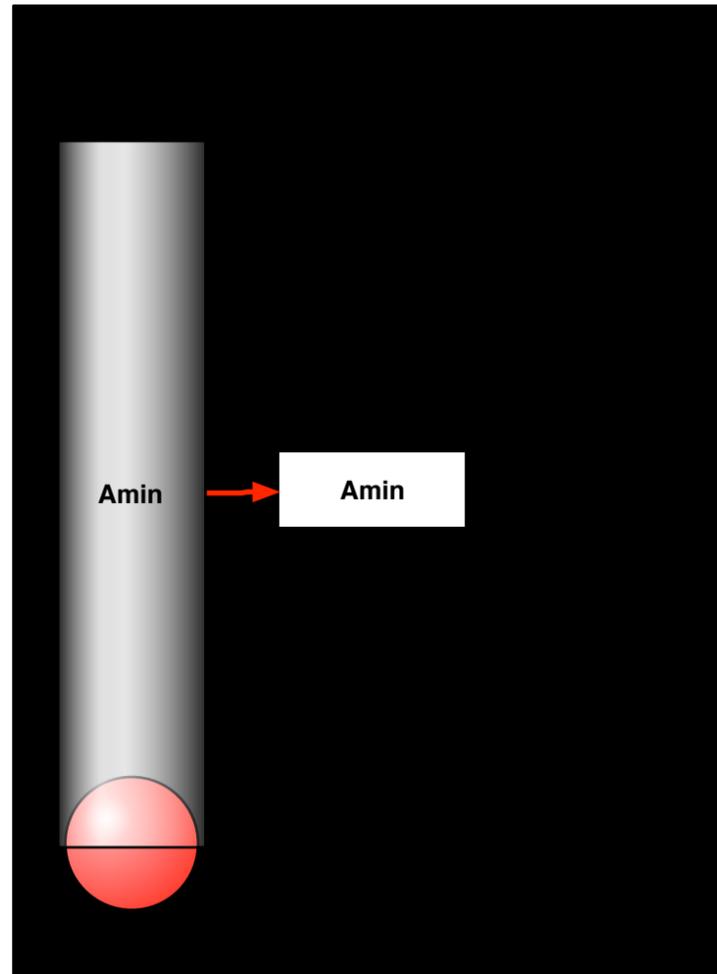
Problem

- Geteilte Ressource
- Zuweilen verborgen

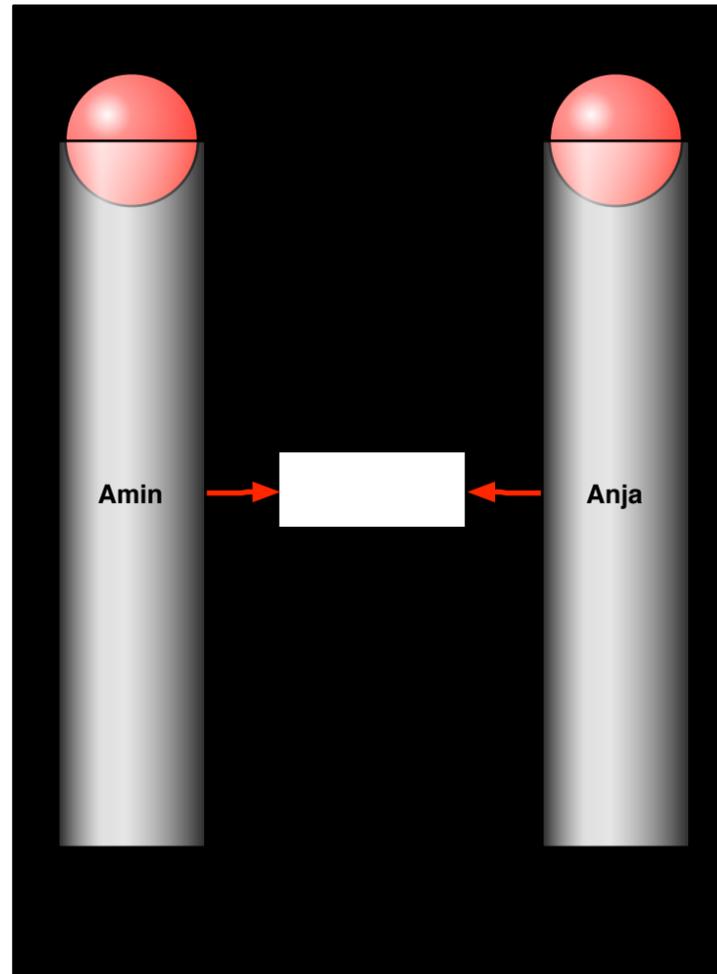
I Kontrollfluss



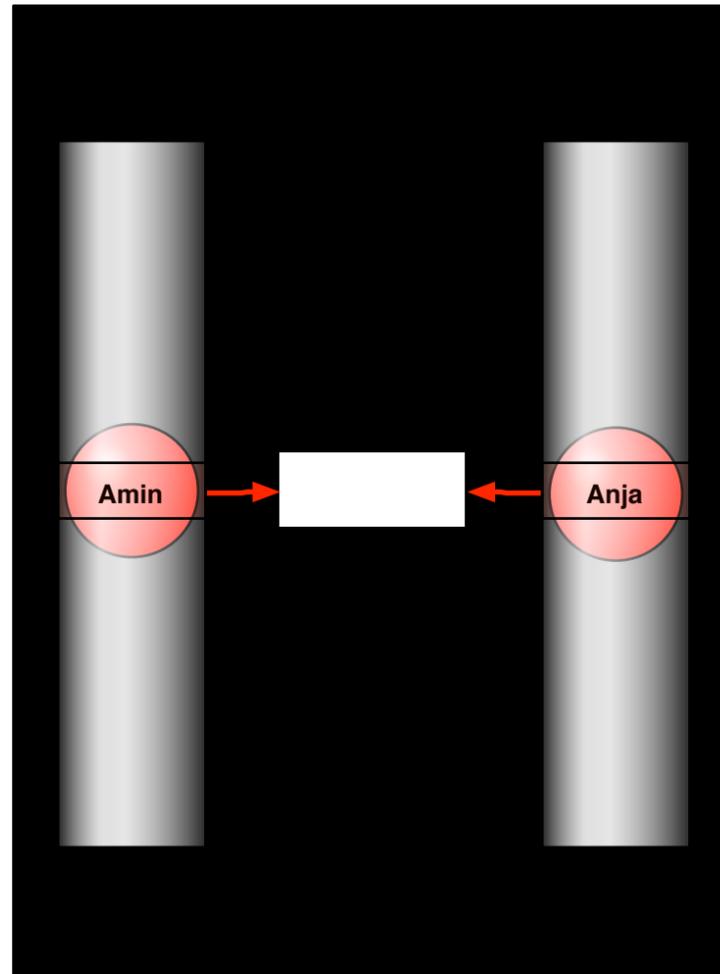
I Kontrollfluss



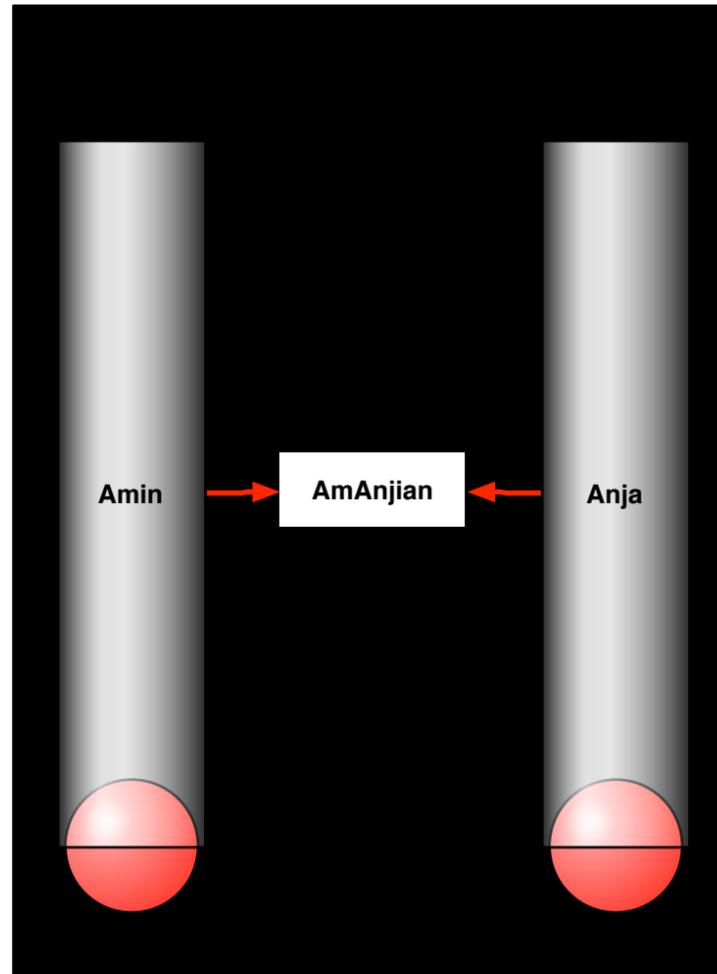
2 Kontrollflüsse



2 Kontrollflüsse



2 Kontrollflüsse



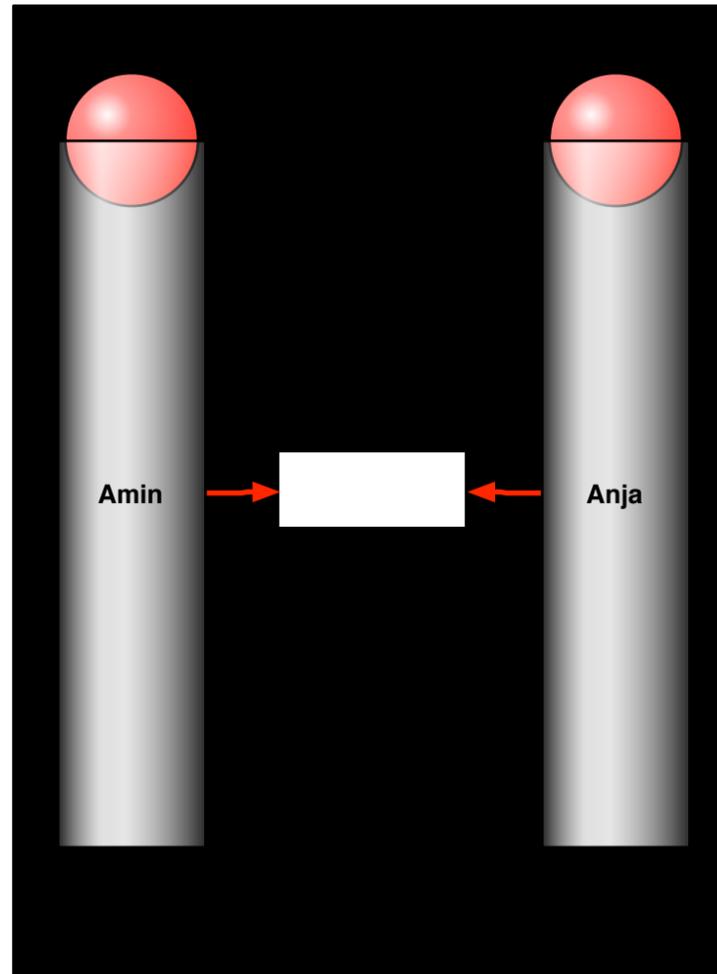
Problem

- Bindings: Gebundenes Objekt
- Core Data: Kontext, Store

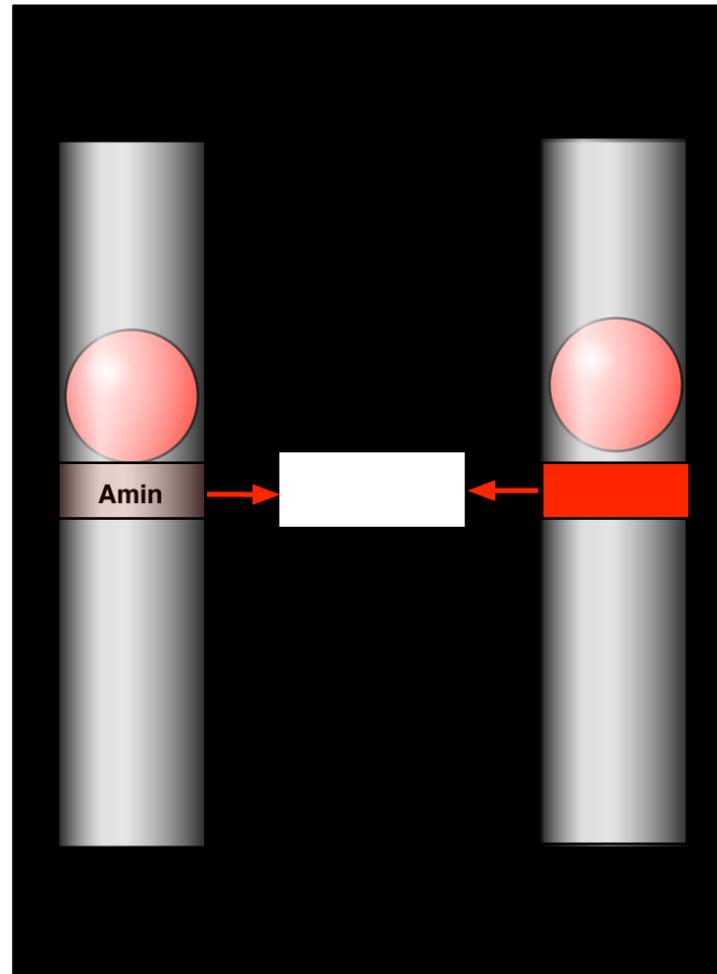
Locks

- Zugriffsbeschränkung

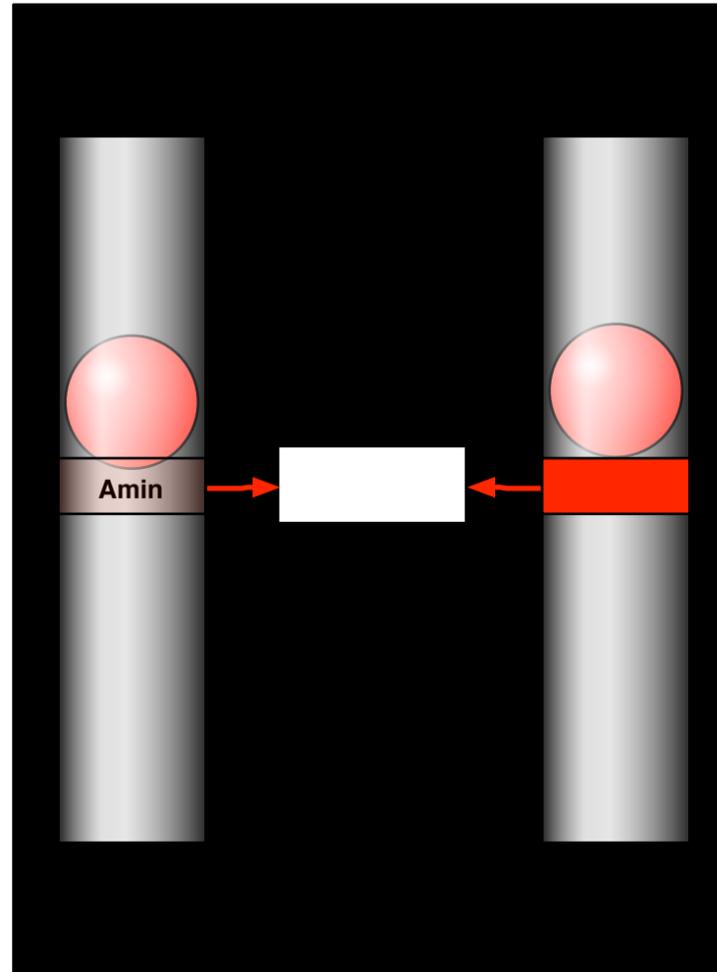
Locks



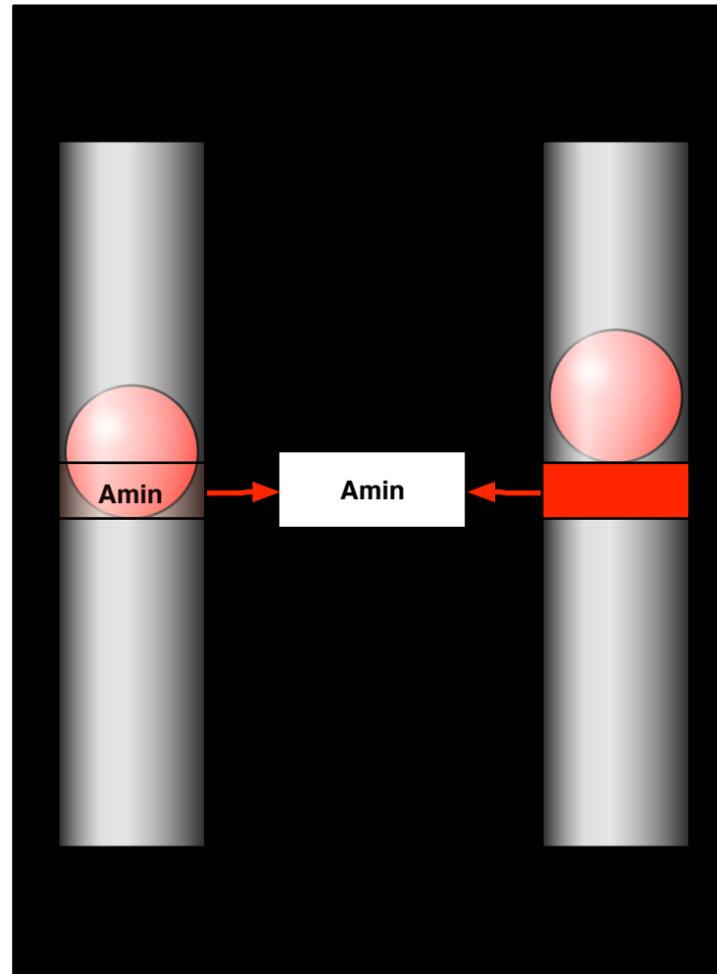
Locks



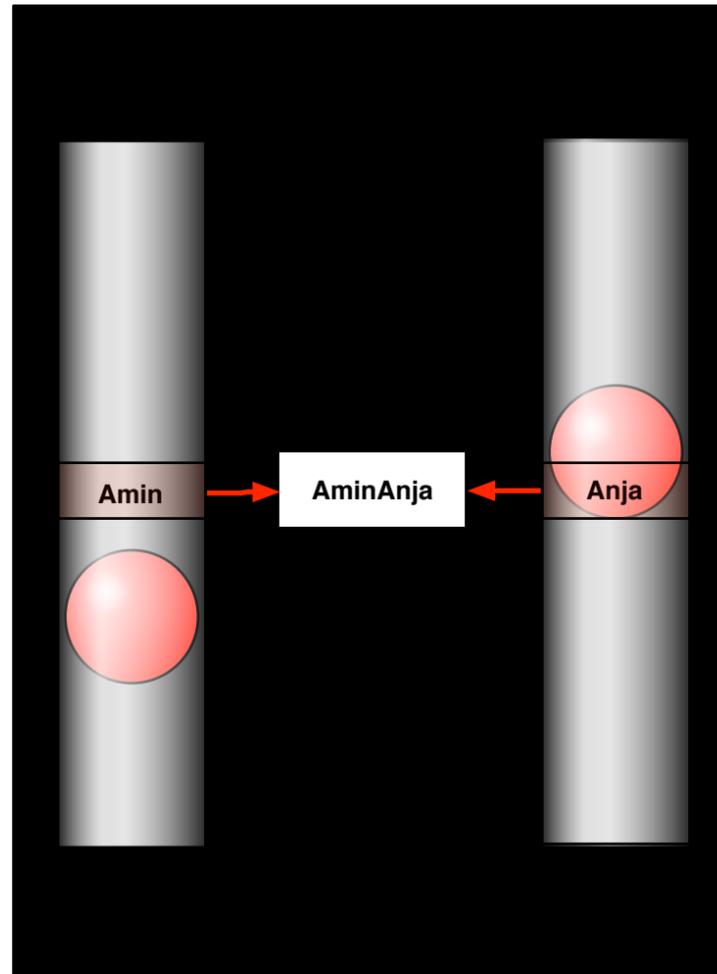
Locks



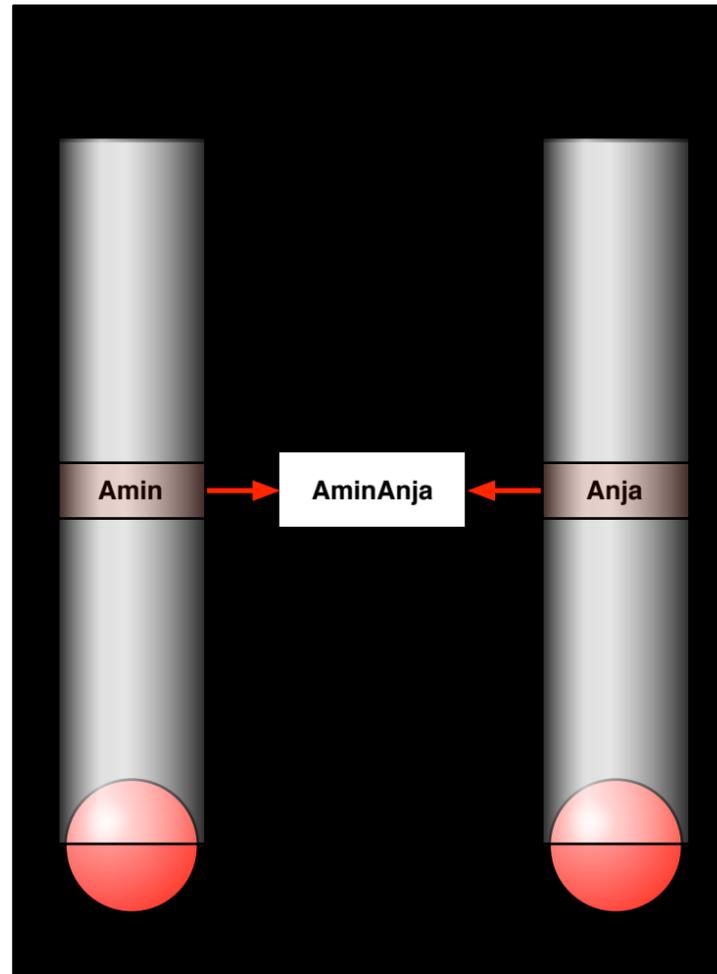
Locks



Locks



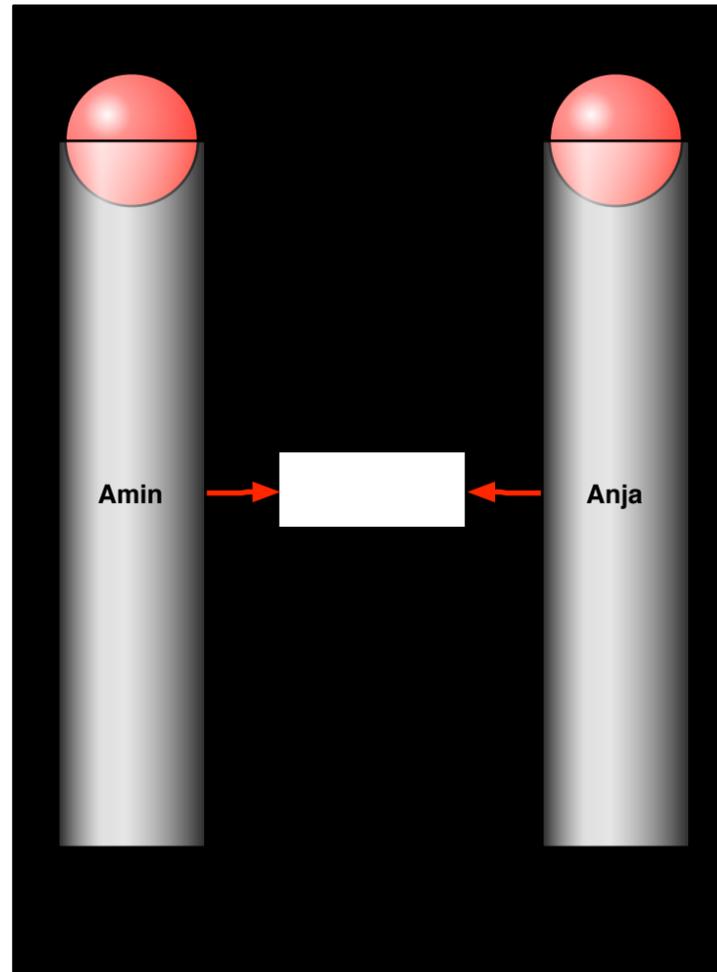
Locks



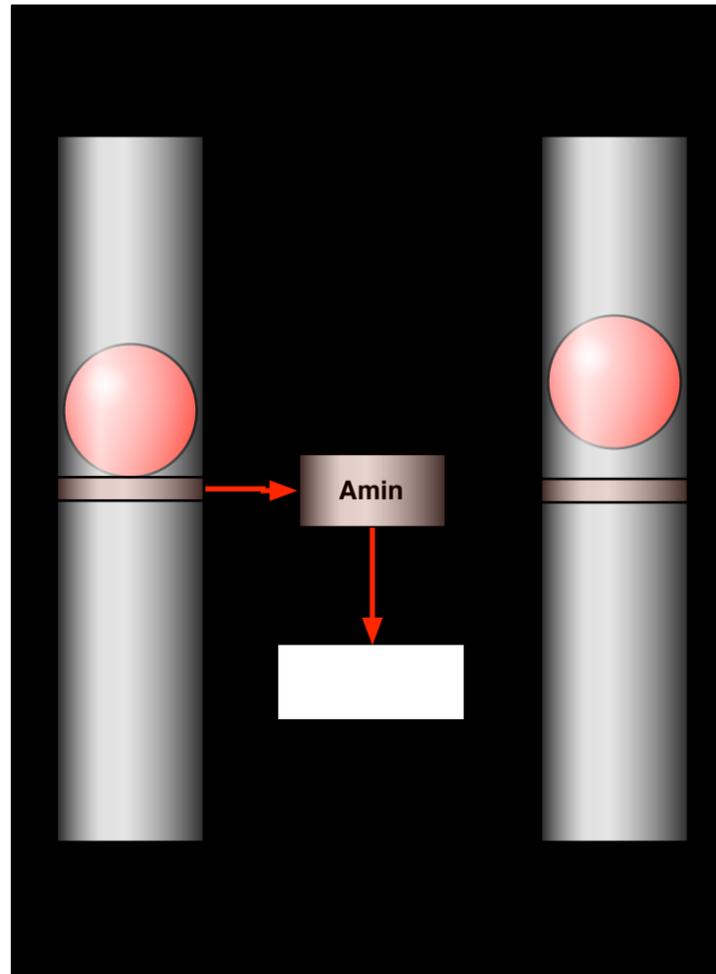
Queues

- Zugriffsserialisierung

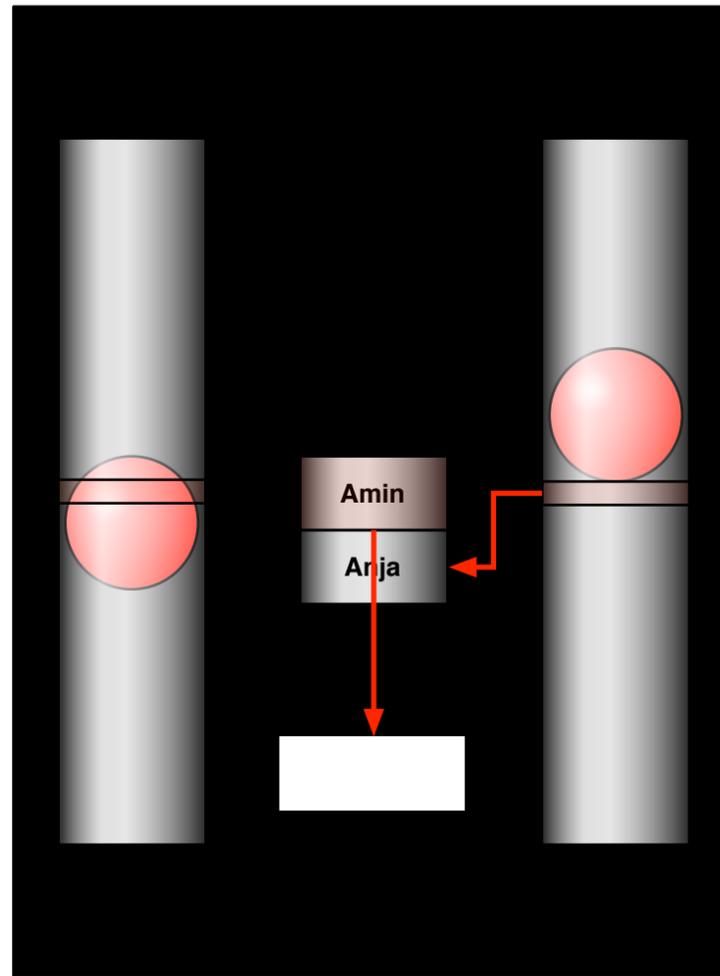
Queues



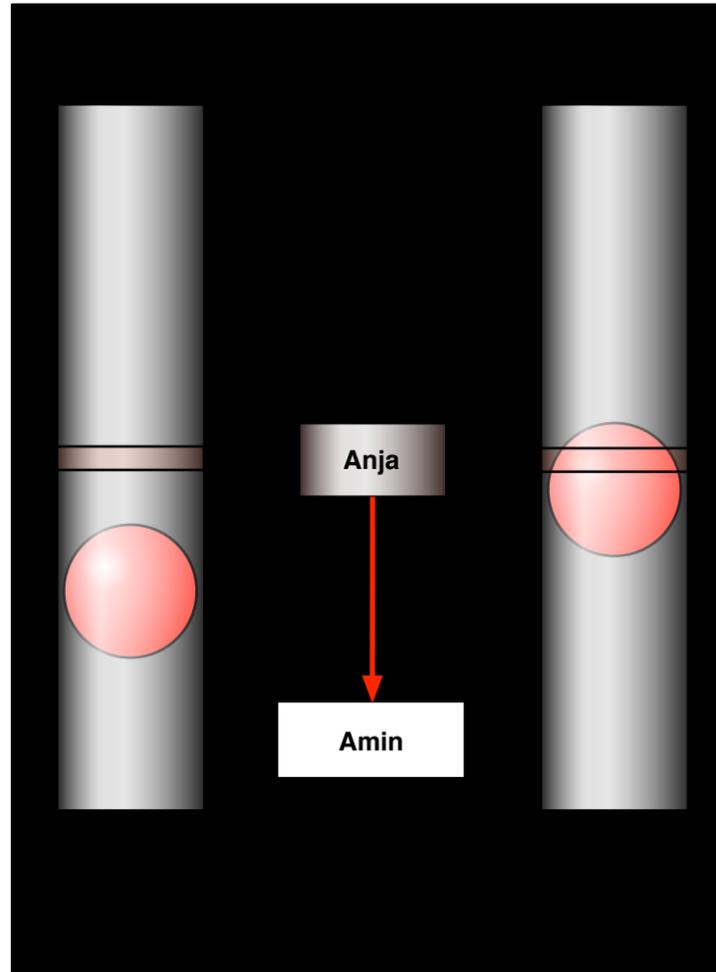
Queues



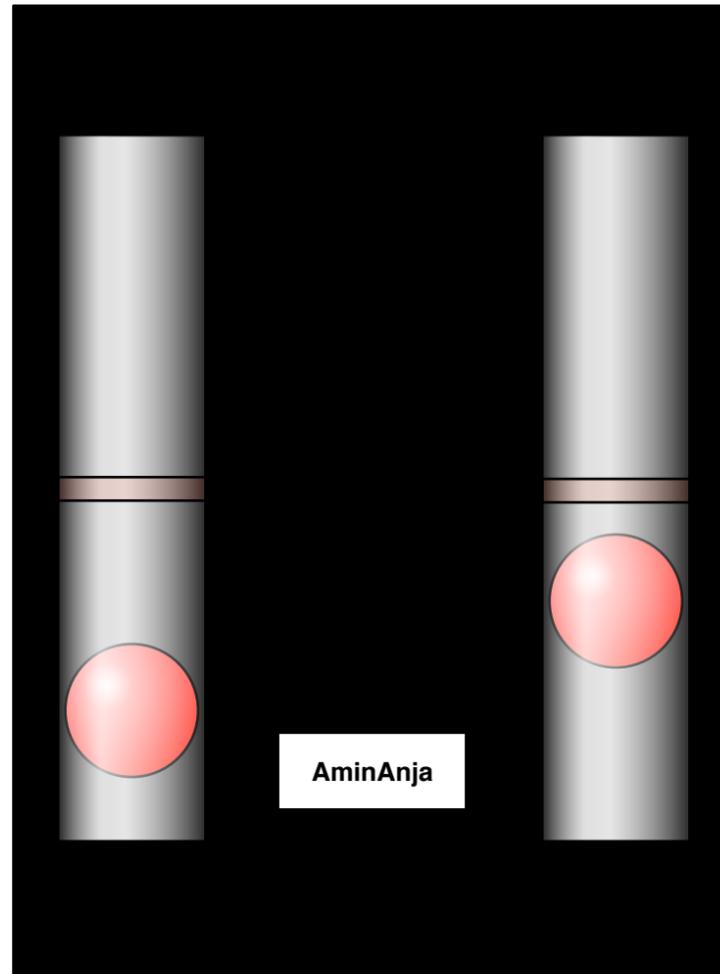
Queues



Queues



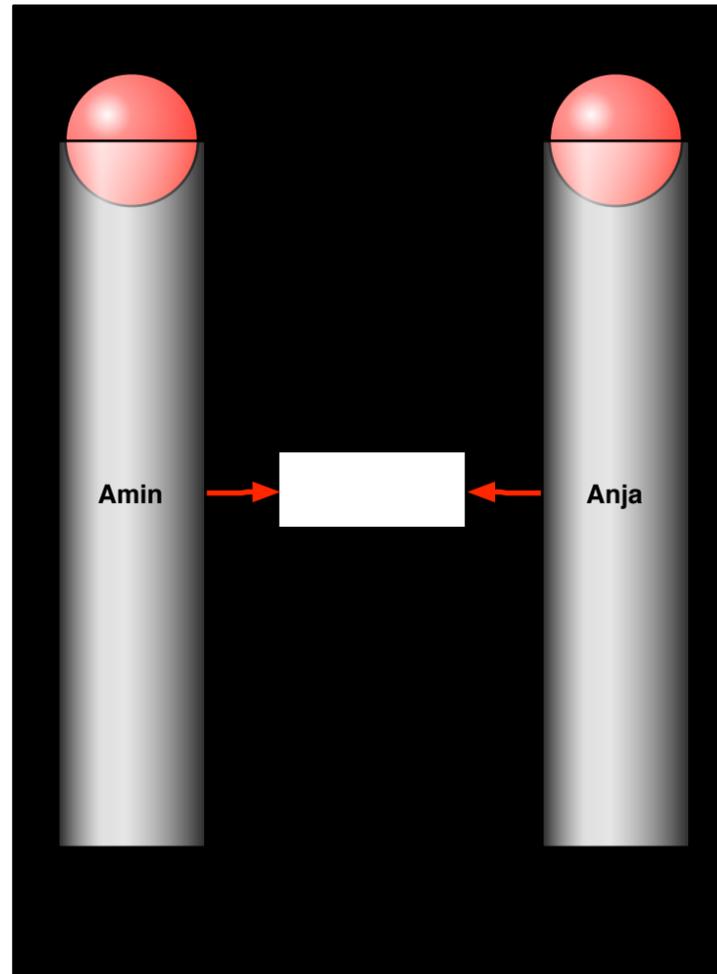
Queues



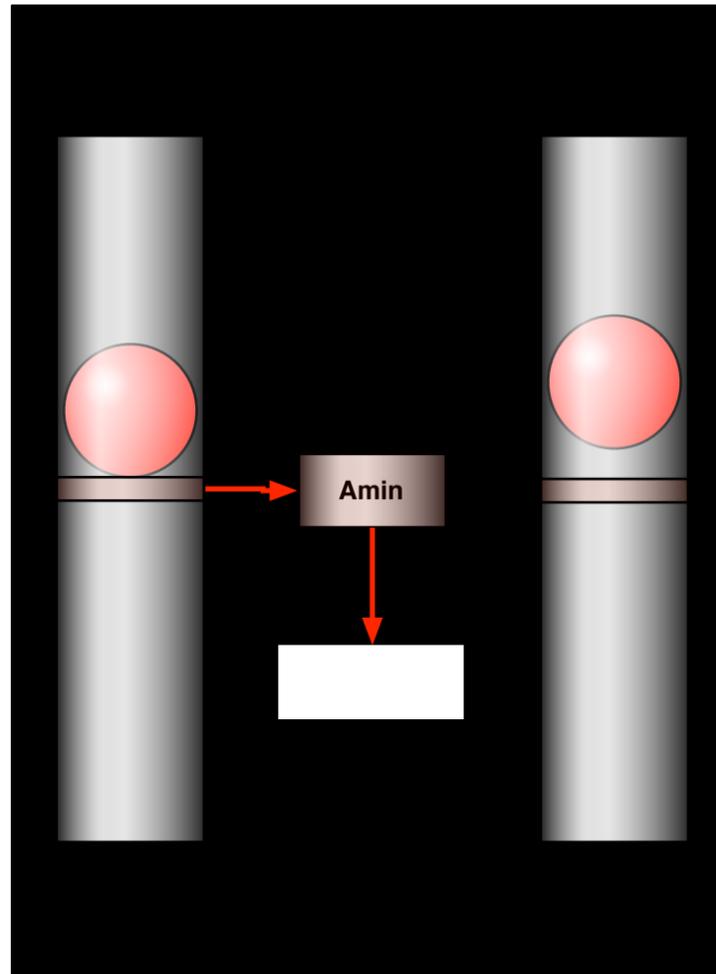
Queues (Synch)

- Zugriffsserialisierung

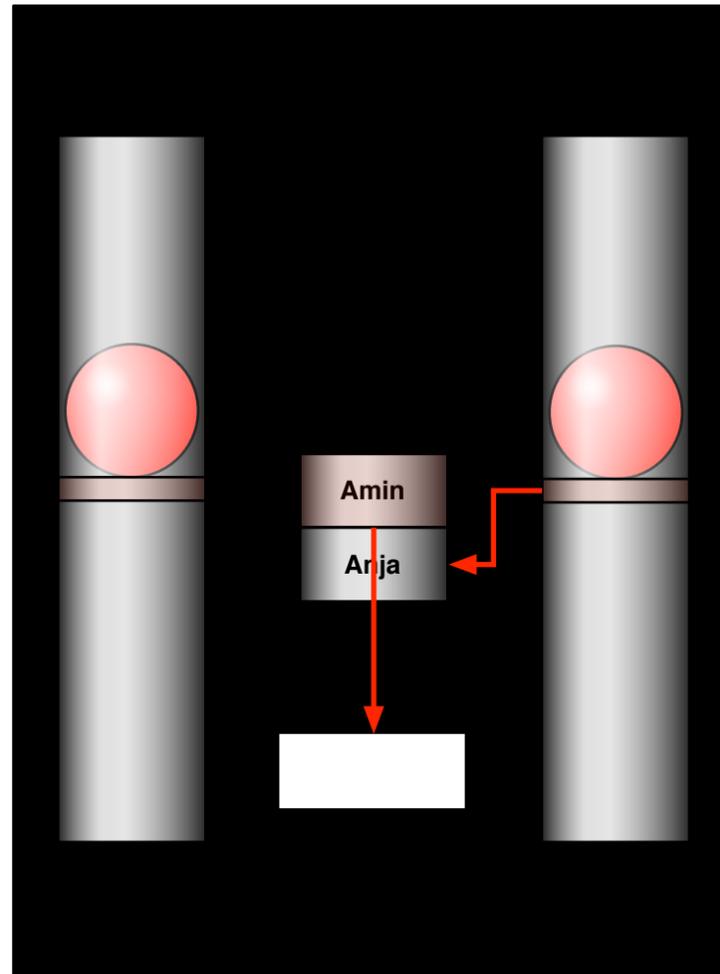
Queues



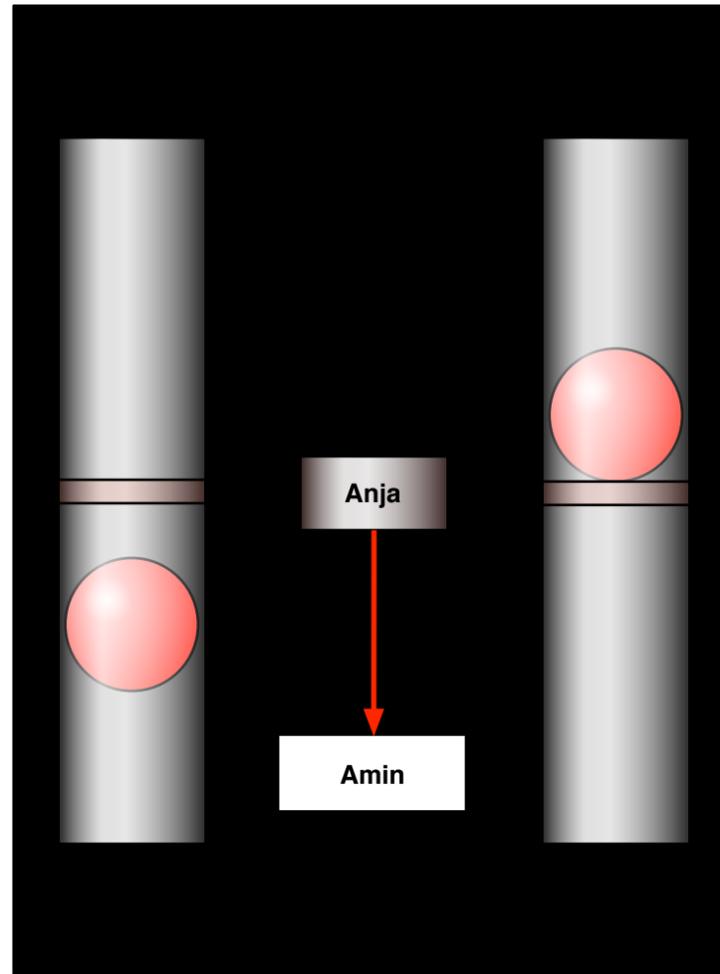
Queues



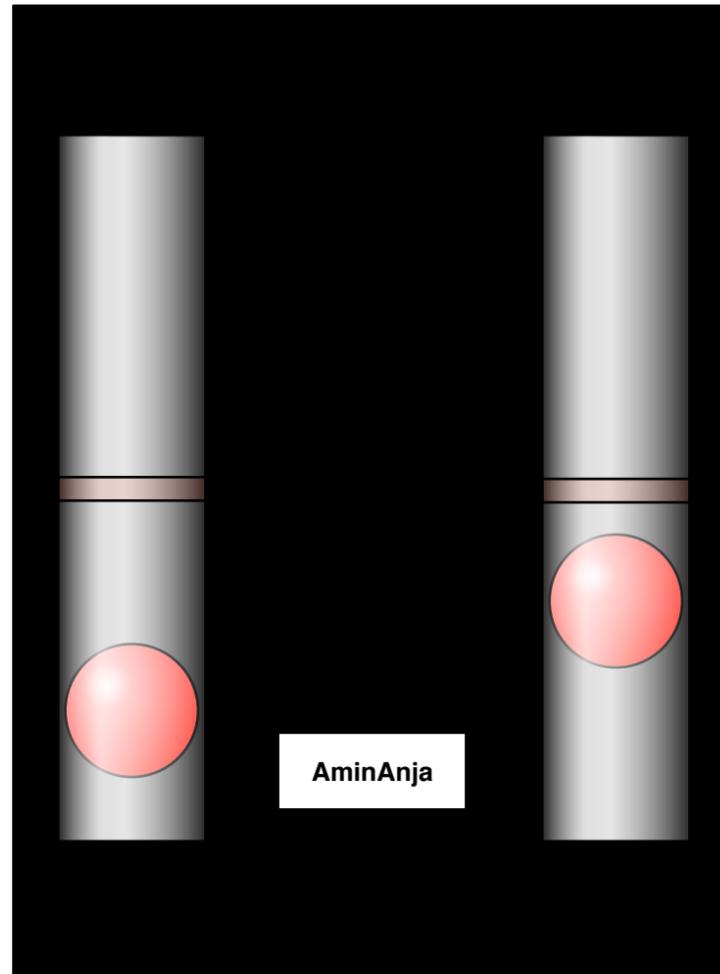
Queues



Queues



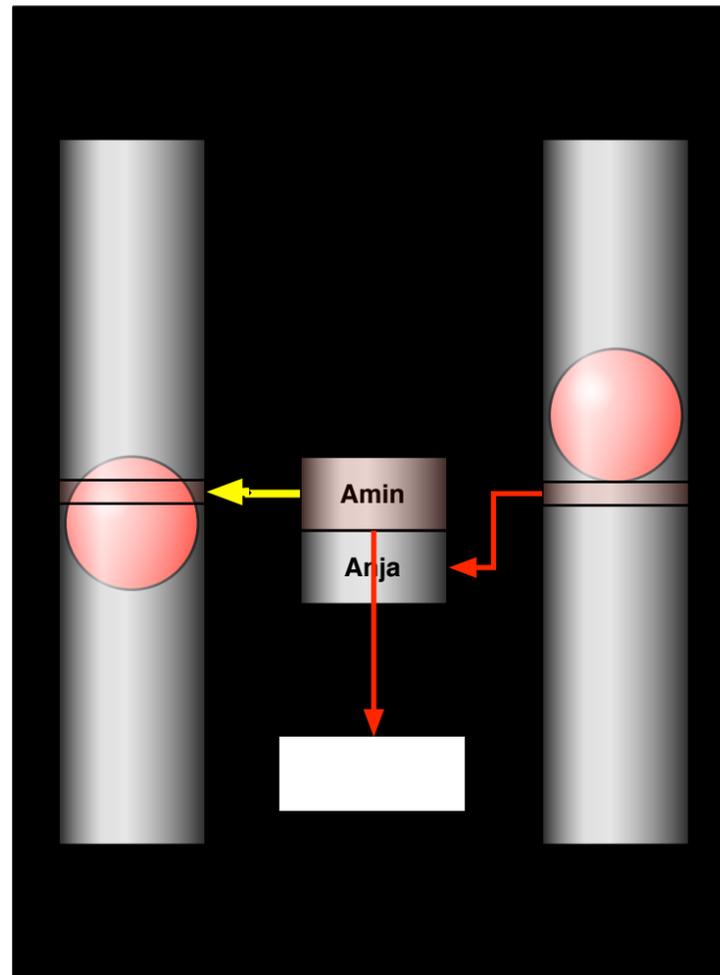
Queues



Blocks

- Kein eigenes System
- Hilfsmittel (Closure)
- State-Fixierung

Blocks



Deadlock

- Lock im Lock (Threads)
- Synch im Synch (Queue)

Run-Loops

Run-Loops

- Seriell
- Input-Sources
- Perform-Selector
- Timer
- Observers
- Modes

Seriell

- Thread der Runloop
- Thread-Shifting

Input-Sources

- Port-based
- Custom

Perform-Selector

- Gesicherte Nachrichten
- Alle
- Pro Thread geordnet
- Thread-Shifting

Timer

- Gesicherte Nachrichten
- Alle
- Pro Thread geordnet
- Thread-Shifting

Oberservers

- Schrittweise Mitteilungen
- Abarbeitungsreihenfolge

Modes

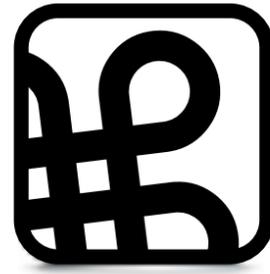
- Auswahl an Sources
- Beispiel: Timer
- Common-Mode

Modal-Sessions

- Kein Beachball
- Operation während Modal-Window
- Keine Nutzereingabe
- Abbruch

Modal-Sessions

- `-beginModalSessionForWindow:`
- `{`
`-runModalSession: // Abbruch`
`... // Lahmer Kram`
`}`
- `-endModalSession:`



Macoun