

Vortrag TH-Rosenheim

2024-01-17

Über mich

Florian.wachs@live.de

Github: florianwachs

DISCLAIMER

Moderne C# Features

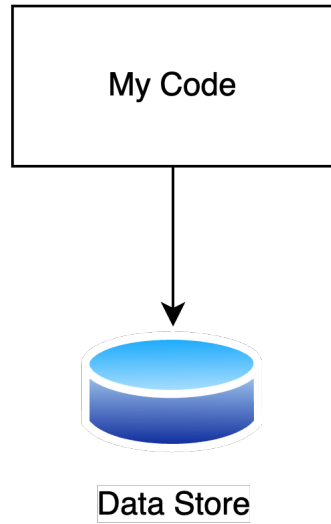
Modern C# Features

- TimeProvider
- Init Properties
- Raw String Literals
- Primary Constructor
- Records
- Collection Expressions
- Switch Expressions
- Pattern Matching

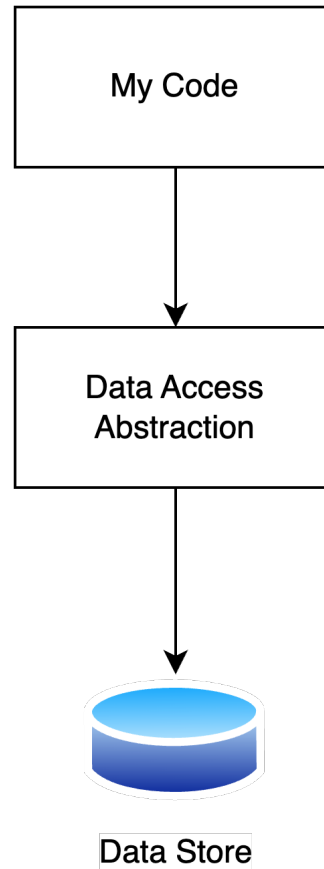
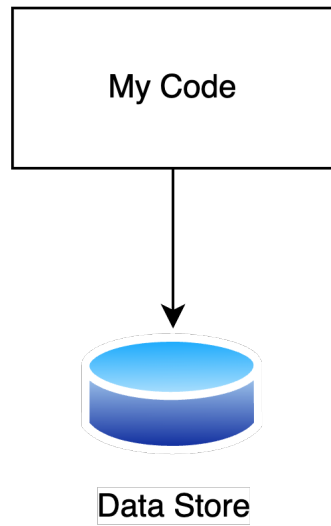
Patterns and Concepts

Repository

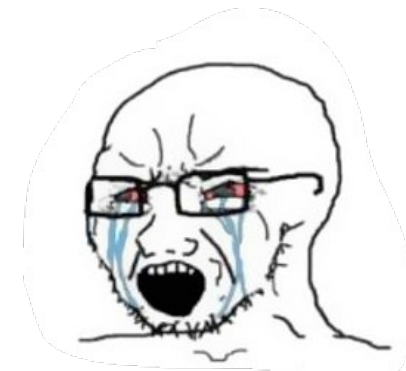
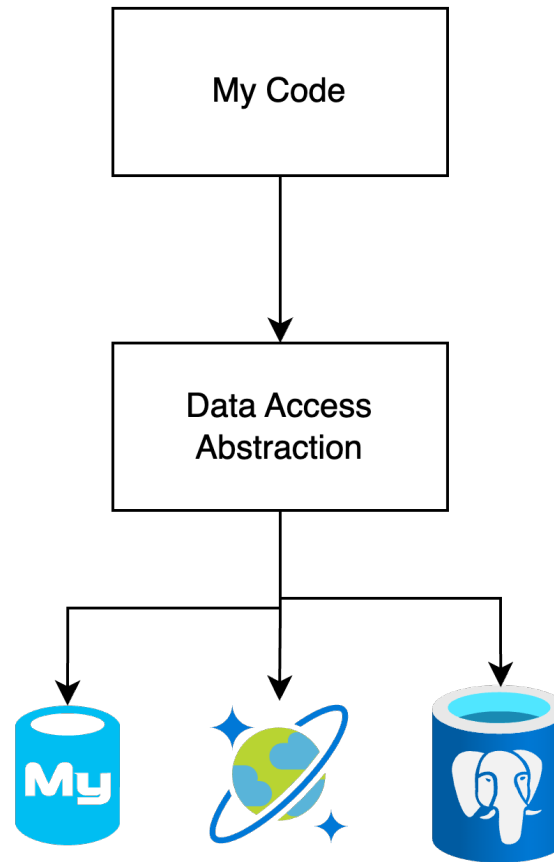
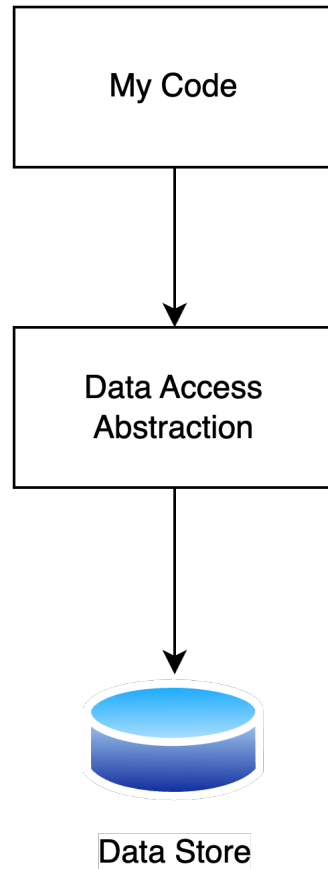
Repository



Repository



Repository



Repository



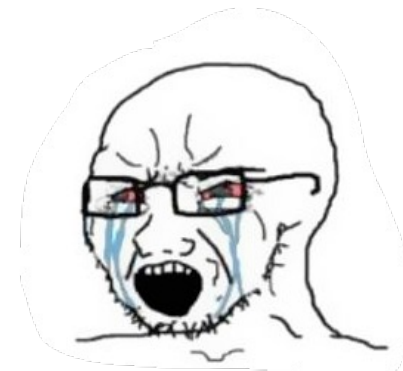
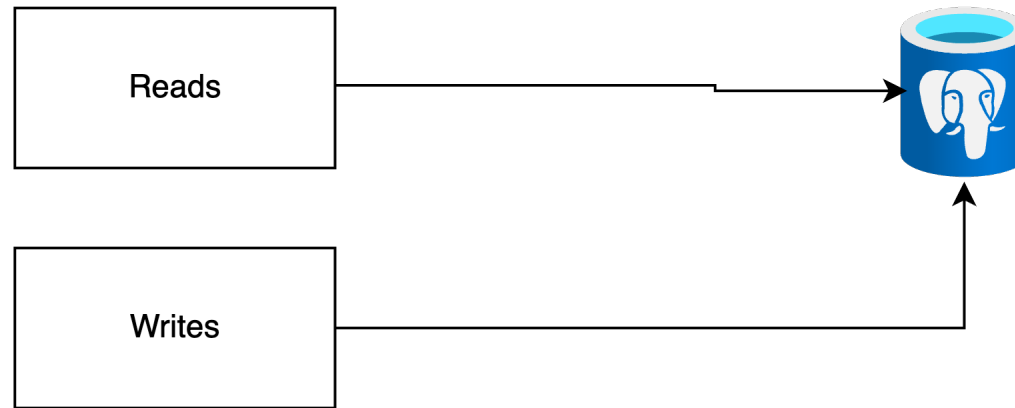
Result Object

Option / Maybe

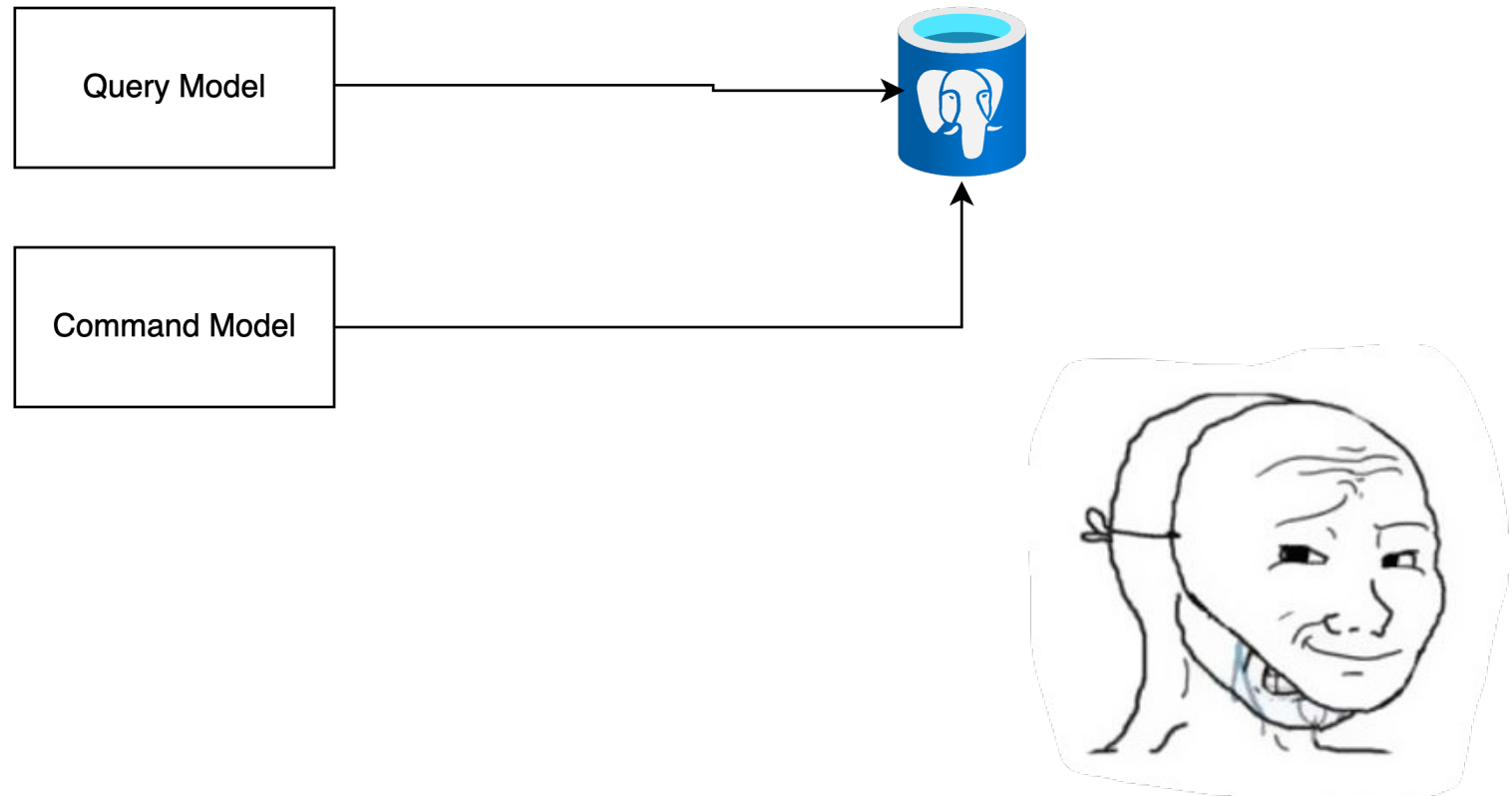
CQRS

Command Query Responsibility Segregation

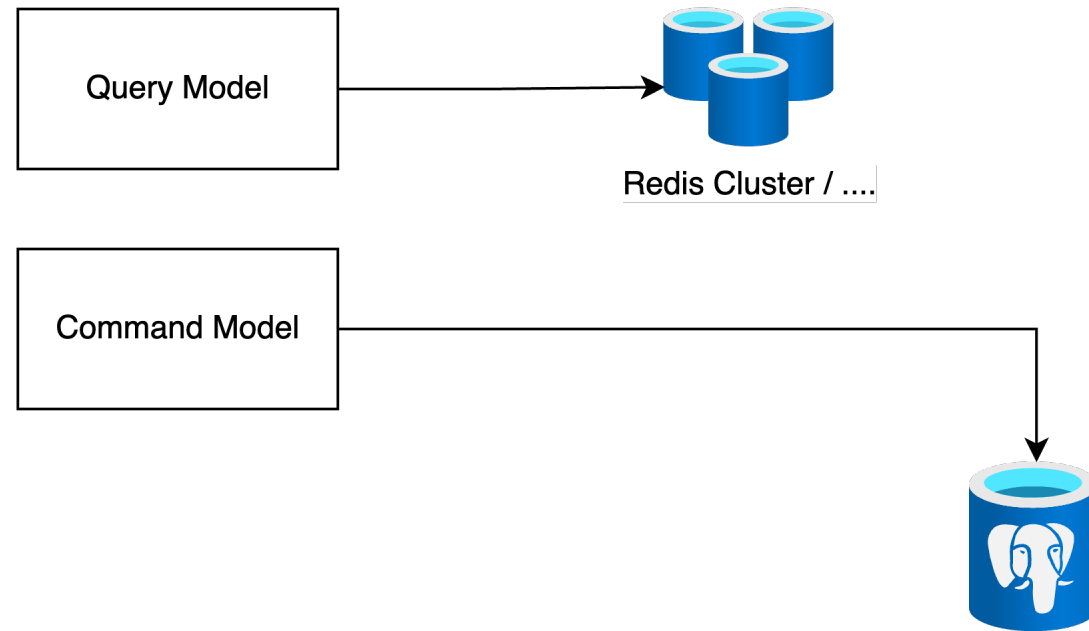
CQRS



CQRS



CQRS



Mediator

Event Sourcing

Event Sourcing

Zahnpasta added

Rasierschaum added

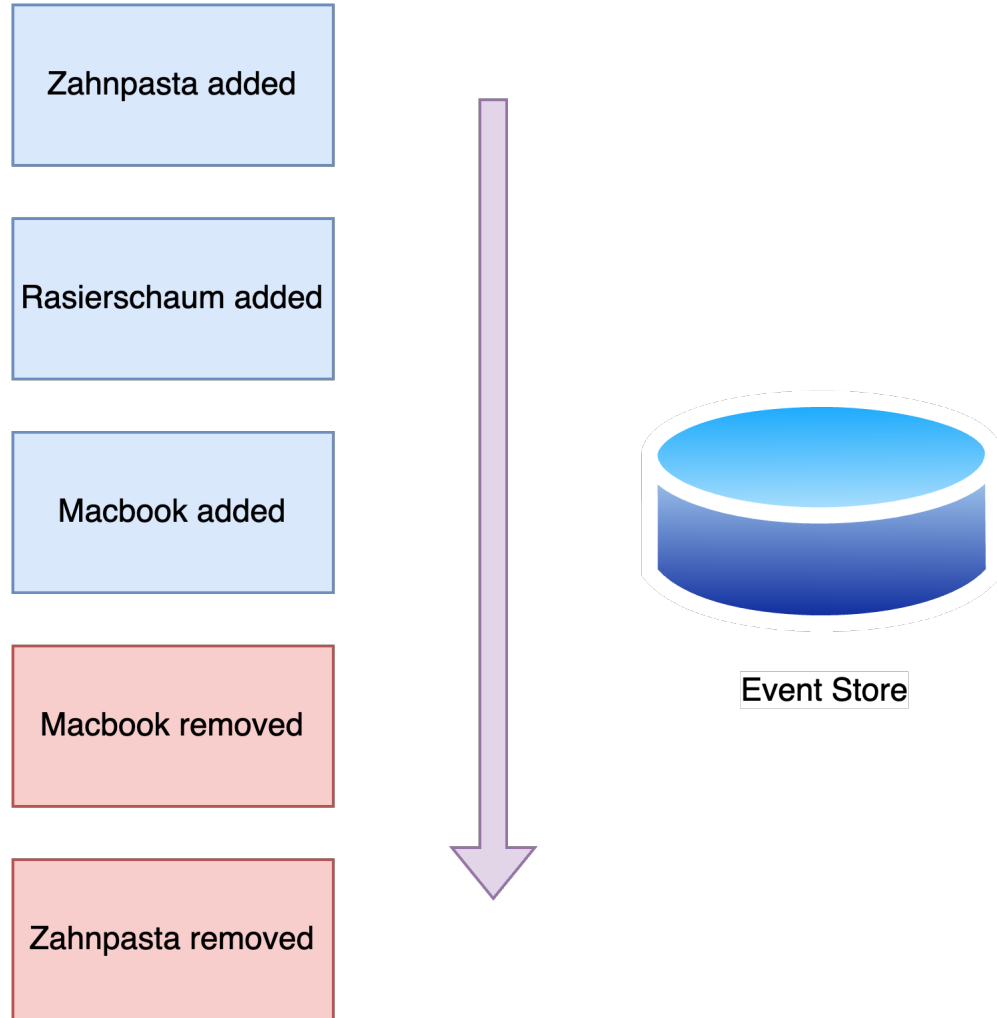
Macbook added

Macbook removed

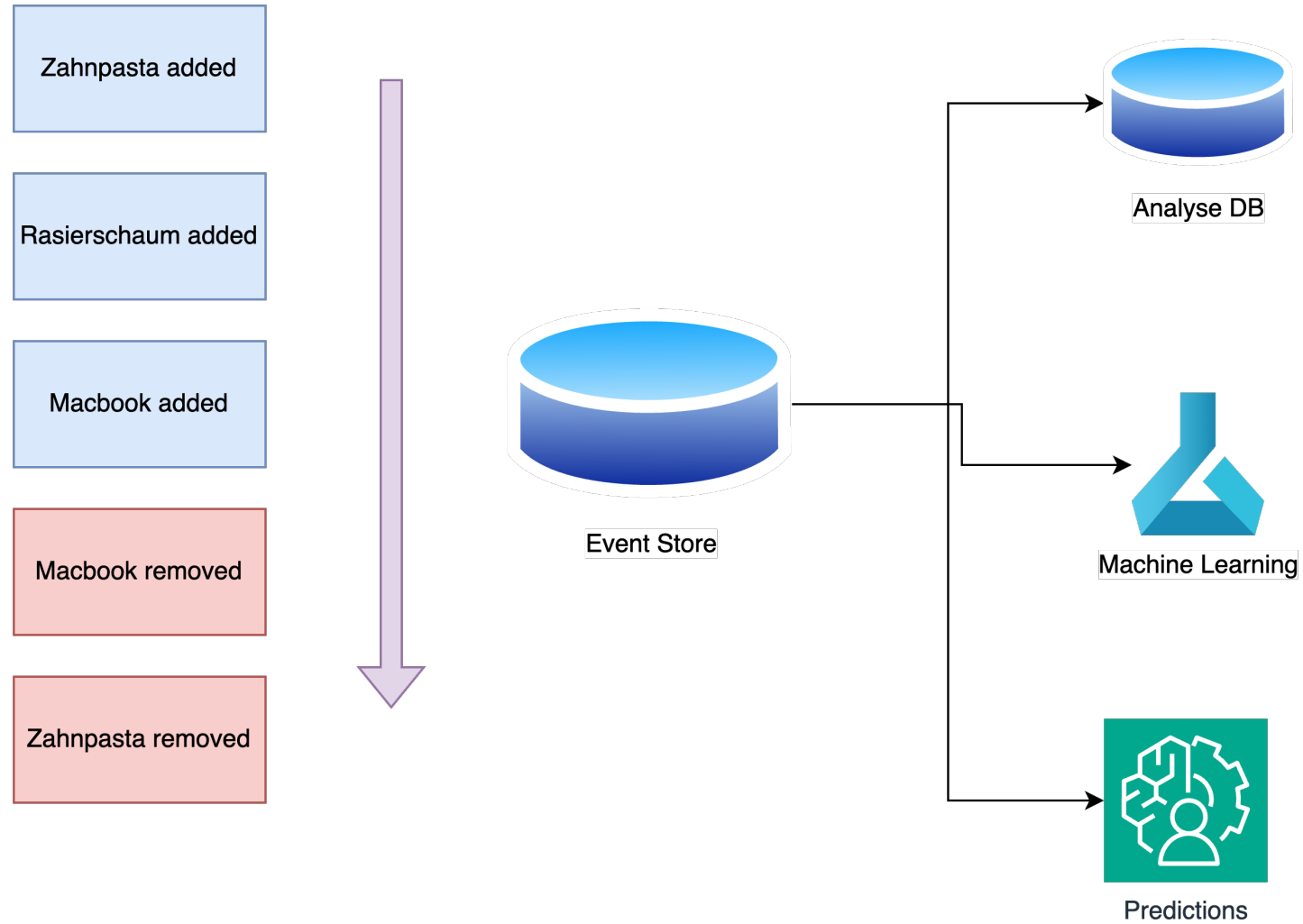
Zahnpasta removed

Cart Id	Product Id	Amount
1	3994	1

Event Sourcing



Event Sourcing



Event Sourcing

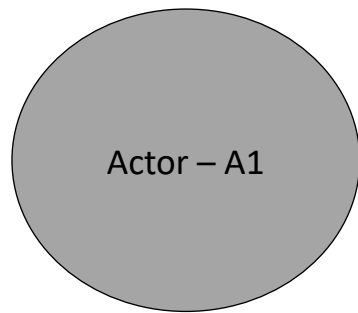
Youtube Videos:

Let's build event store in one hour! - Oskar Dudycz - NDC Oslo 2022

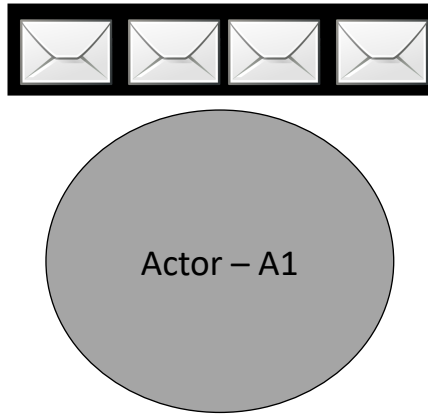
Greg Young – Event Sourcing

Actors

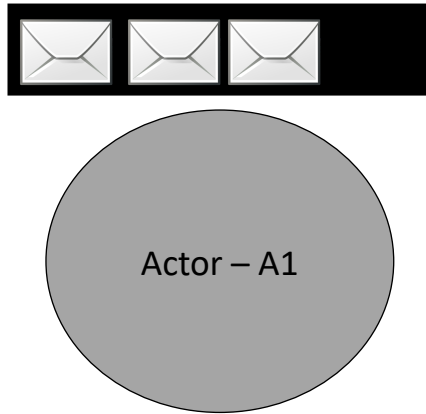
Was ist ein Aktor?



Was ist ein Aktor?

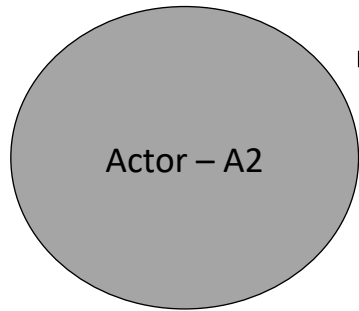


Was ist ein Aktor?

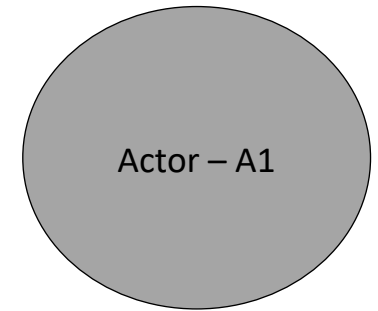


- ▶ Lieber Actor – A1,
Es ist beinahe Dezember und ich vermisse
Dich, es wäre wunderbar wenn du mir auf dies
Nachricht meines von Sehnsucht zerissenen
Herzens antworten würdest.
- ▶ .. Deine Actor – A2

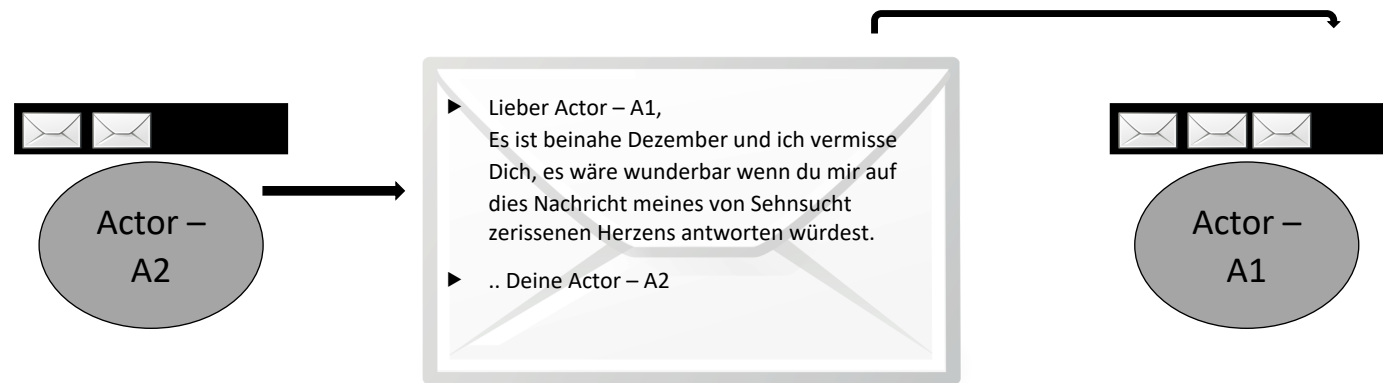
Was ist ein Aktor?



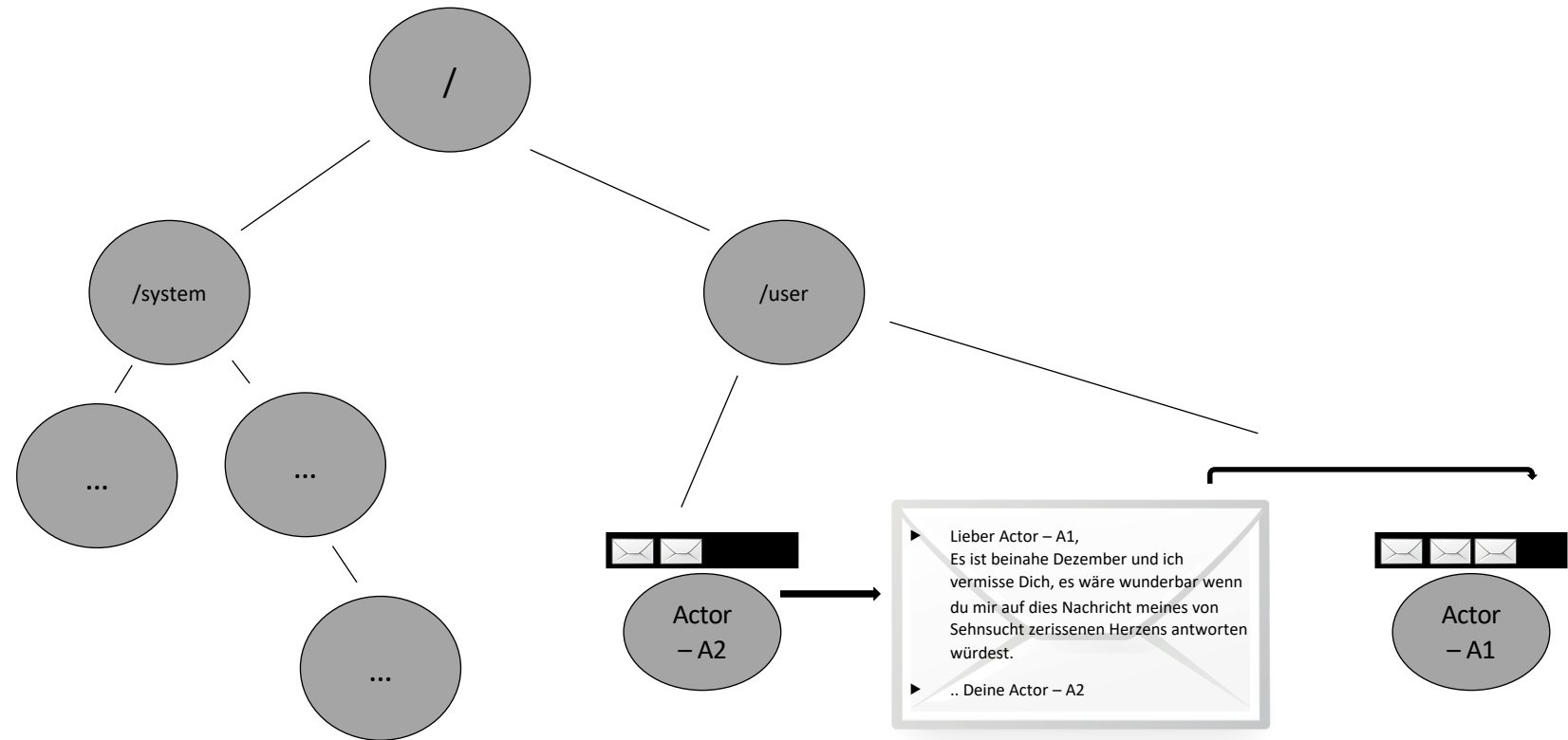
- ▶ Lieber Actor – A1,
Es ist beinahe Dezember und ich vermisse
Dich, es wäre wunderbar wenn du mir auf dies
Nachricht meines von Sehnsucht zerissenen
Herzens antworten würdest.
- ▶ .. Deine Actor – A2



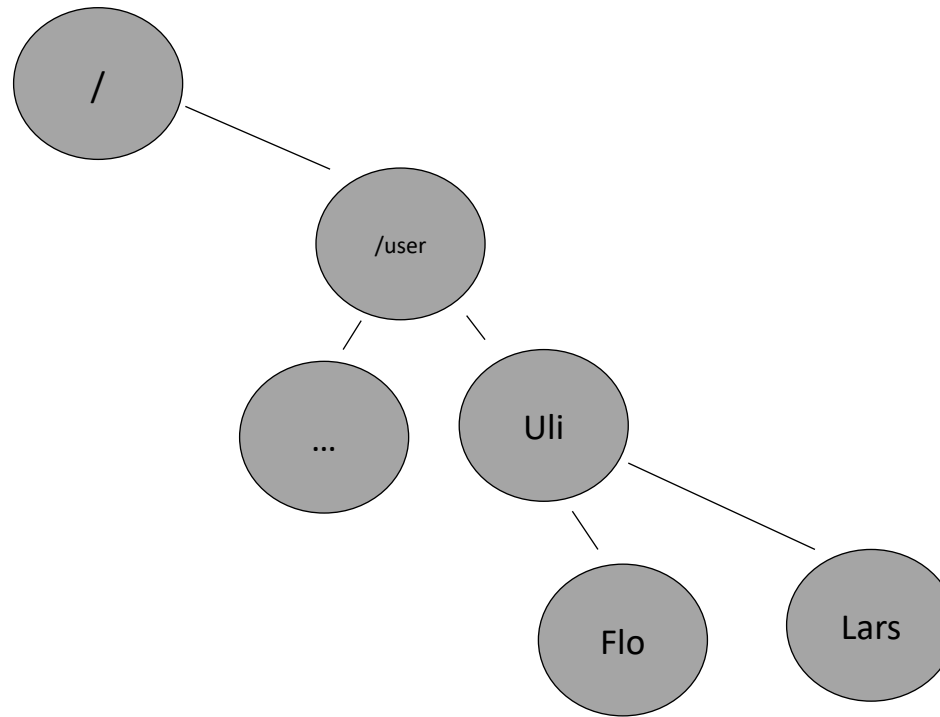
Was ist ein Aktorsystem?



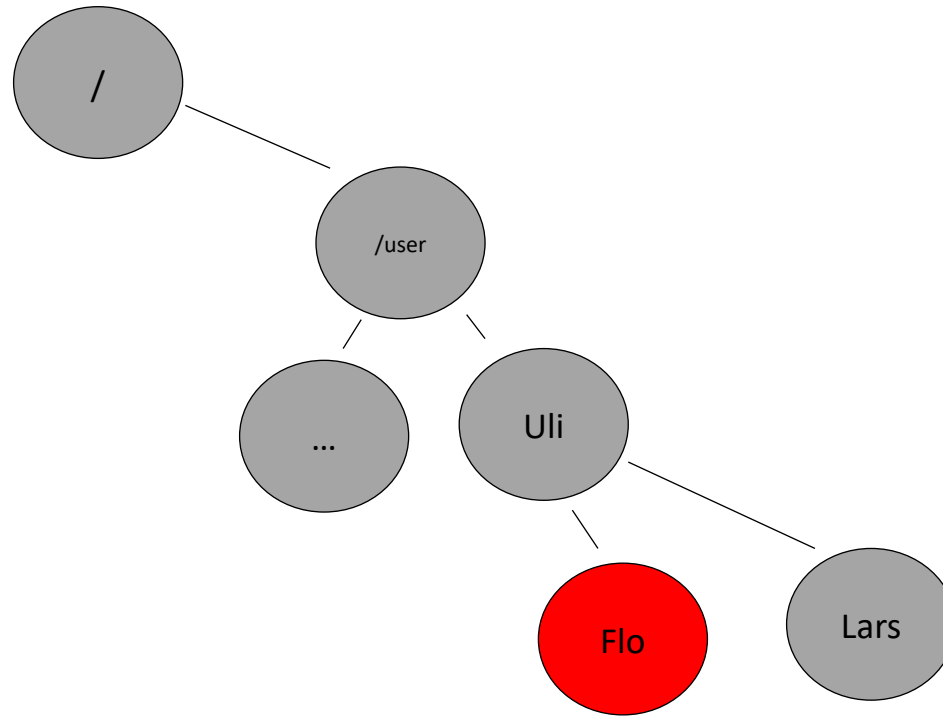
Was ist ein Aktorsystem?



Was ist ein Error-Kernel?



Was ist ein Error-Kernel?



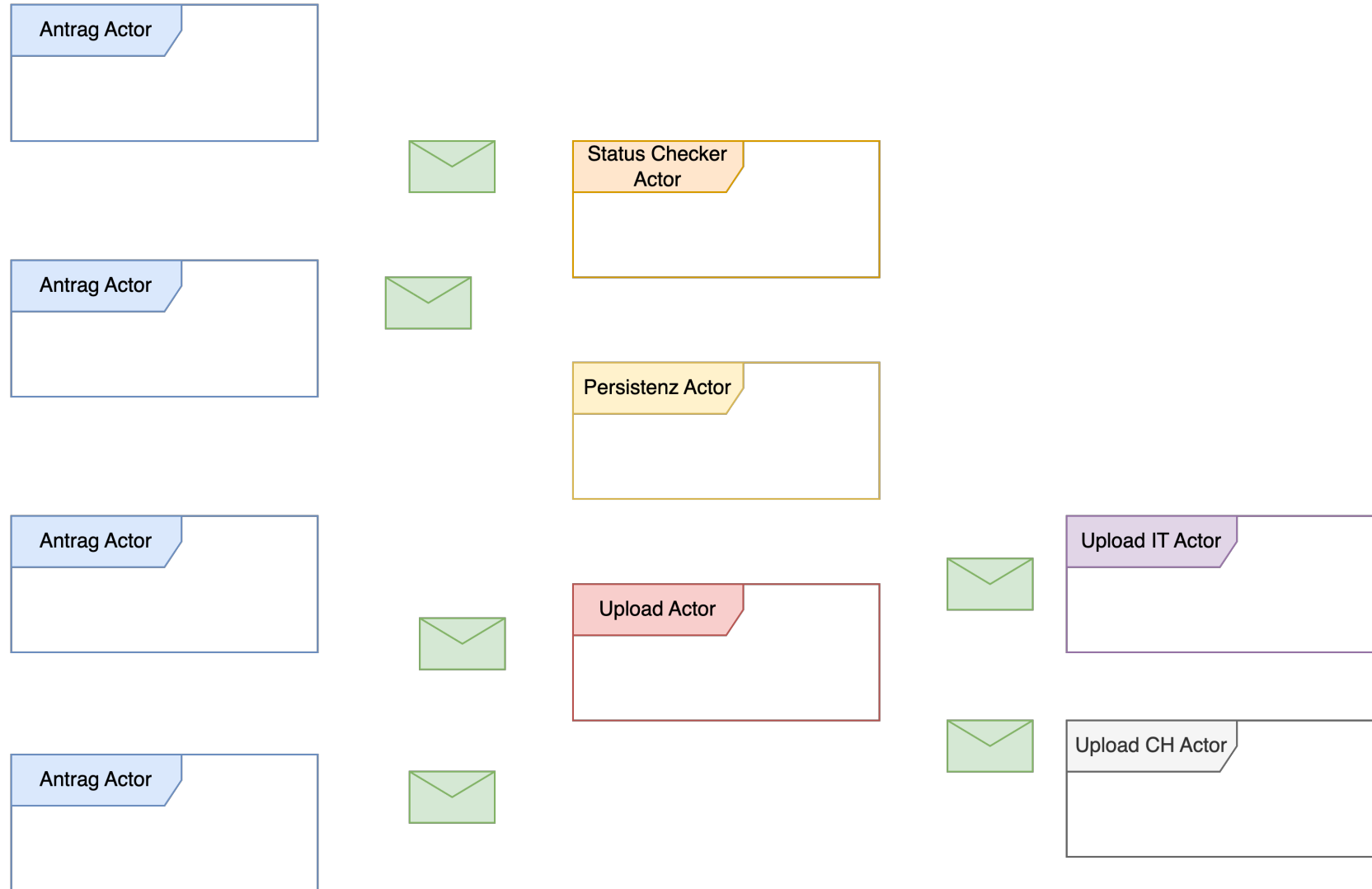
Aktor

- Eigenständige Komponente innerhalb eines Aktorsystems
- Kommunikation ausschließlich über Nachrichten die zwischen Aktoren verschickt werden (kein „Public State“)
- Nachrichten gehen in der Mailbox ein und werden sequentiell verarbeitet
- Keine Nebenläufigkeit innerhalb des Aktors

Aktorsystem

- Kontext in dem Aktoren ausgeführt werden
- Kümmt sich um das Scheduling der Nachrichtenverarbeitung
- Kann sich über Rechnergrenzen hinweg erstrecken (Cluster)
- Jeder Akteur erhält eine eindeutige Adresse
- Stellt weitere Dienste bereit, z.B. Lokalisierung, Error-Handling, Logging, Scheduling, Dead-Letter-Queue, Persistenz,...

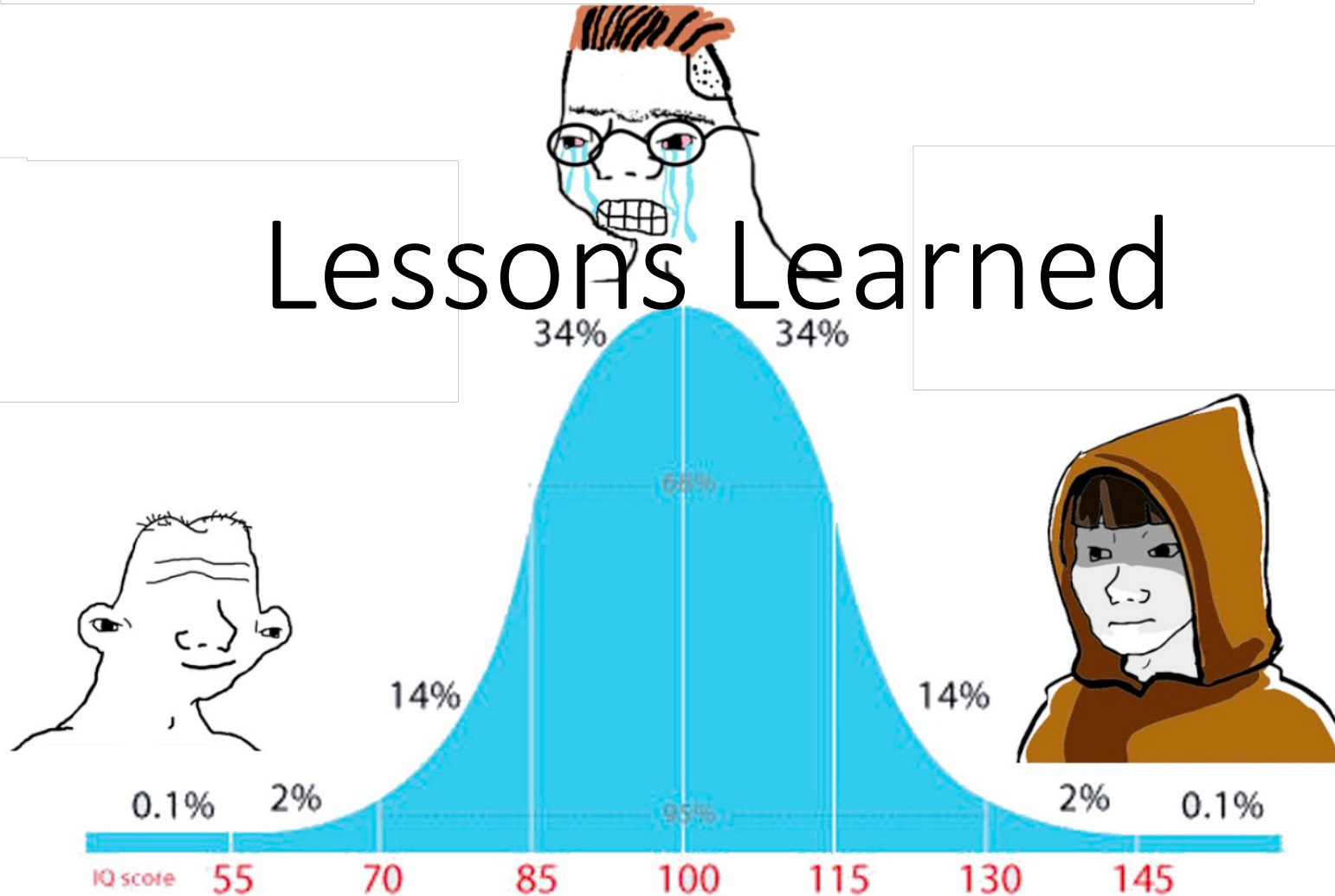
Aktoren



Aktoren mit dotnet

- [Akka.NET Documentation | Akka.NET Documentation \(getakka.net\)](#)
- [Proto.Actor](#)
- [dotnet/orleans: Cloud Native application framework for .NET \(github.com\)](#) (Virtual Actors)
- [Dapr - Distributed Application Runtime](#)

Lessons Learned



Testing

Testing

When I look around now, I see a lot of people using **mocks to replace all their dependencies**. My concern is that they will begin to hit the **Fragile Test** issues that mocks present. Gerard Meszaros identifies the issues we hit as two specific smells: **Overspecified Software** and **Behavior Sensitivity**.

<http://codebetter.com/iancooper/2007/12/19/mocks-and-the-dangers-of-overspecified-software/>

Testing

Refactoring (noun): a **change made to the internal structure** of software to make it easier to understand and cheaper to modify **without changing its observable behavior**.

<https://martinfowler.com/bliki/DefinitionOfRefactoring.html>

Testing

If the program's **behavior is stable** from an observer's perspective, **no tests should change**.

https://medium.com/@kentbeck_7670/programmer-test-principles-d01c064d7934

Testing

- „The Trigger for a new test is new behavior“

Diagrams as Code

Diagrams as Code

- [Mermaid | Diagramming and charting tool](#)
- <https://www.plantuml.com/>
- [The C4 model for visualising software architecture](#)
 - <https://structurizr.com/>

Lessons Learned

- Soft Skills
- PKIM
- „echte“ DB für Integrationstests
- Es ist Ok JSON in einer SQL DB zu speichern
- Testing erlaubt schnellere Softwareentwicklung
- eine funktionale und eine objektorientierte Sprache lernen
- Vom Code anderer Lernen
- API für UI und Automations sind verschieden
- Es muss nicht immer eine SPA sein (HTMX...)
- Nicht jedem Trend „blind“ nachlaufen (GraphQL, Microservices, ...)

Vielen Dank!