

Einführung in CQRS

Command & Query Responsibility Segregation

Ich bin...

Software-Entwickler

- mit CQRS produktiv seit 2010

"Software Craftsman"

- und lerne beständig

Trainer

- Softwareentwicklung, agil, .net, Team-Entwicklung



Philip Jander

[http:// Jander.IT](http://Jander.IT) @ph_j

Erwartungen

- Was bedeutet CQRS?
- Was kann man mit CQRS anfangen?
- (notwendigerweise triviale) Beispielanwendung

CQRS ist...

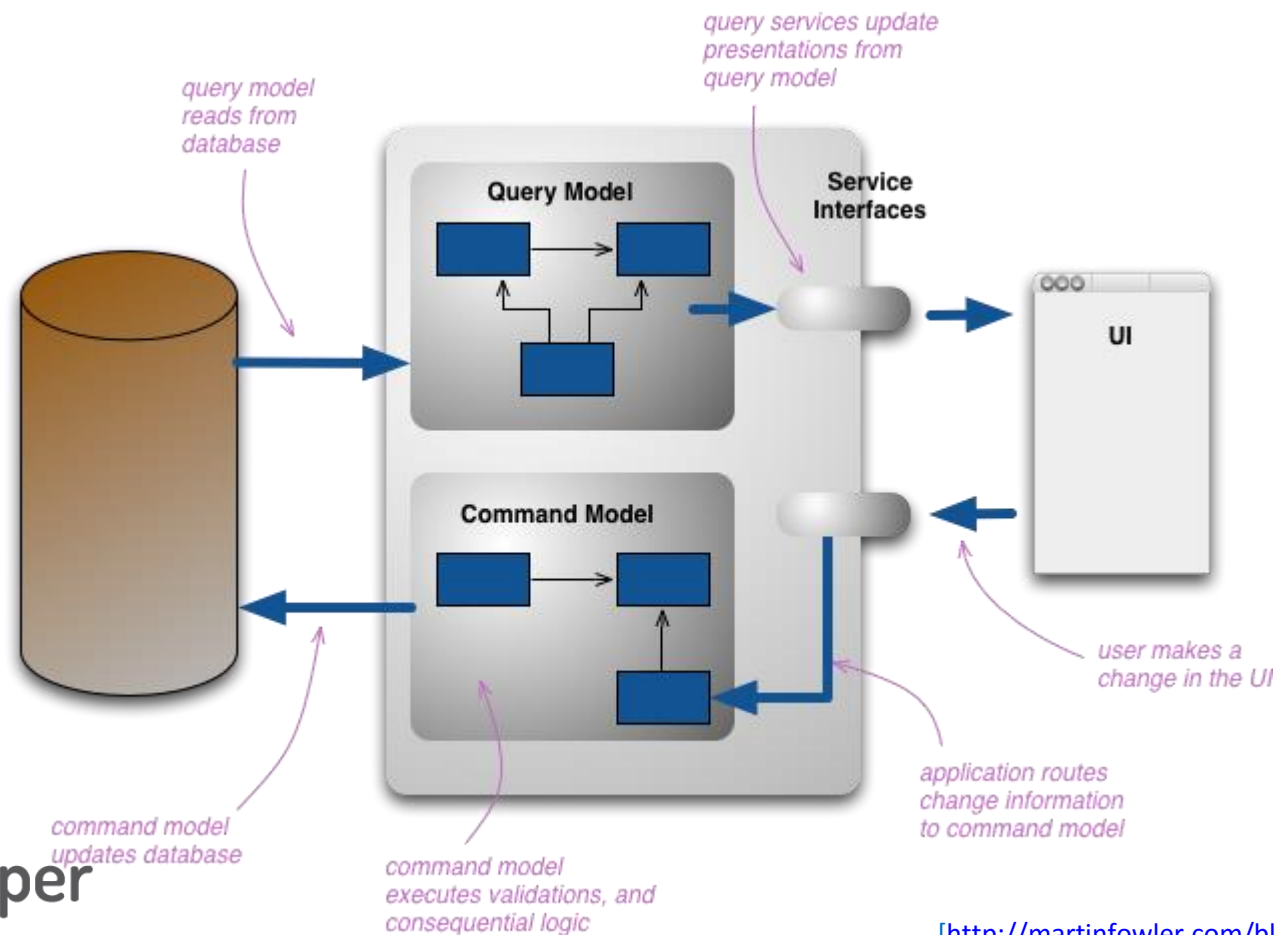
- ▶ ...ein Prinzip

zur Entkopplung



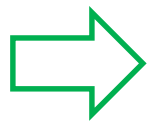
Was bedeutet CQRS?

CQRS: use a different model to update information than the model you use to read information [M. Fowler]



Was bedeutet CQRS?

Der Wert von CQRS liegt darin,
für verschiedene Kanäle
für jedes Teilsystem
unabhängig Entscheidungen treffen zu können



CQRS (und jede Implementierung)
ist immer lokal und niemals global

CQRS ist...

- ▶ ...ein Prinzip.
- ▶ inzwischen ein „Ökosystem“ diverser Pattern
- ▶ Udi Dahan (SOA)
- ▶ Greg Young (DDD)



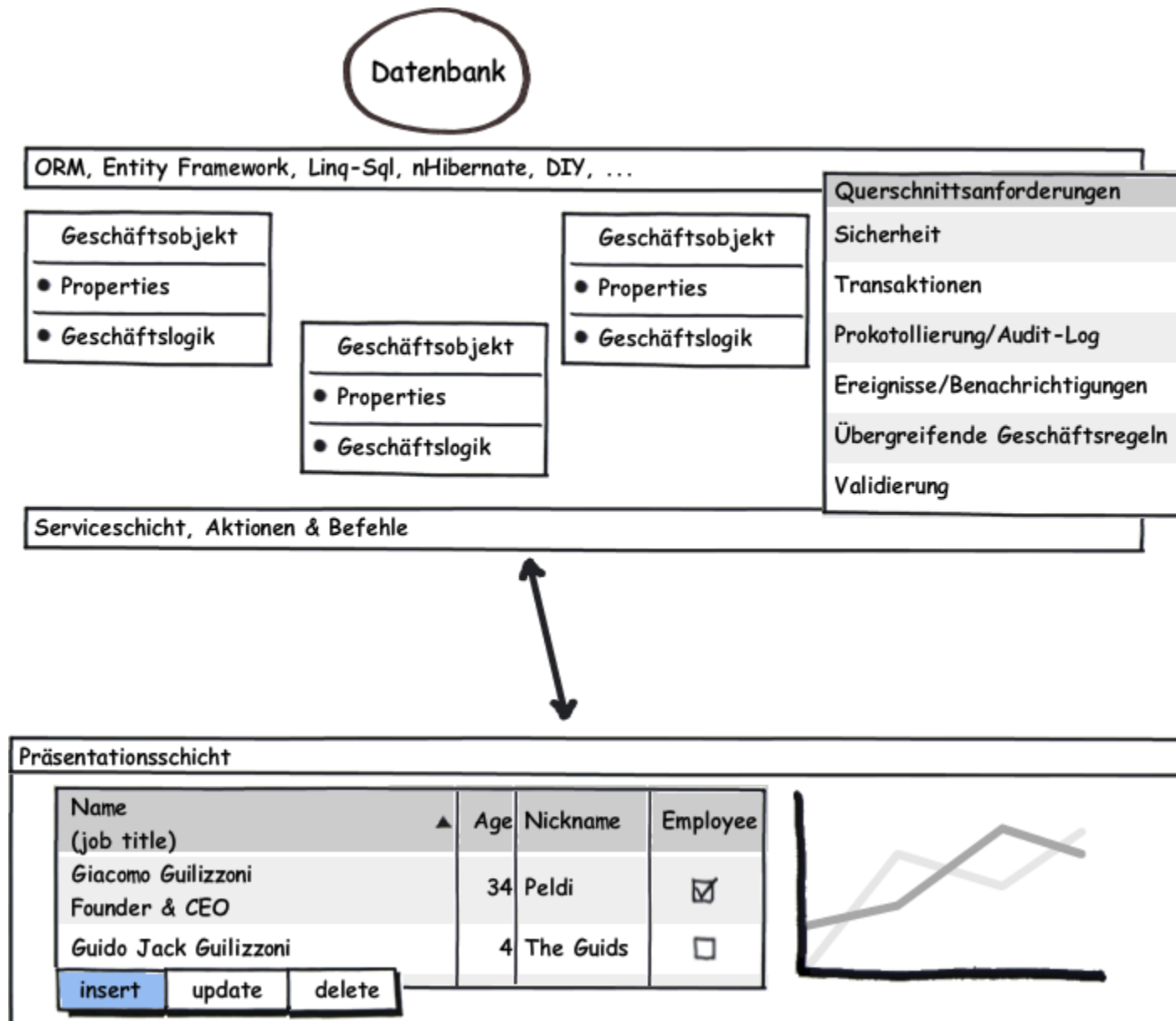
Teilsystem: Domänenmodell mit vertikalem Stack

Datenzugriff

Geschäftsmodell

Dienste

Präsentation



Teilsystem: Domänenmodell mit vertikalem Stack

Datenzugriff

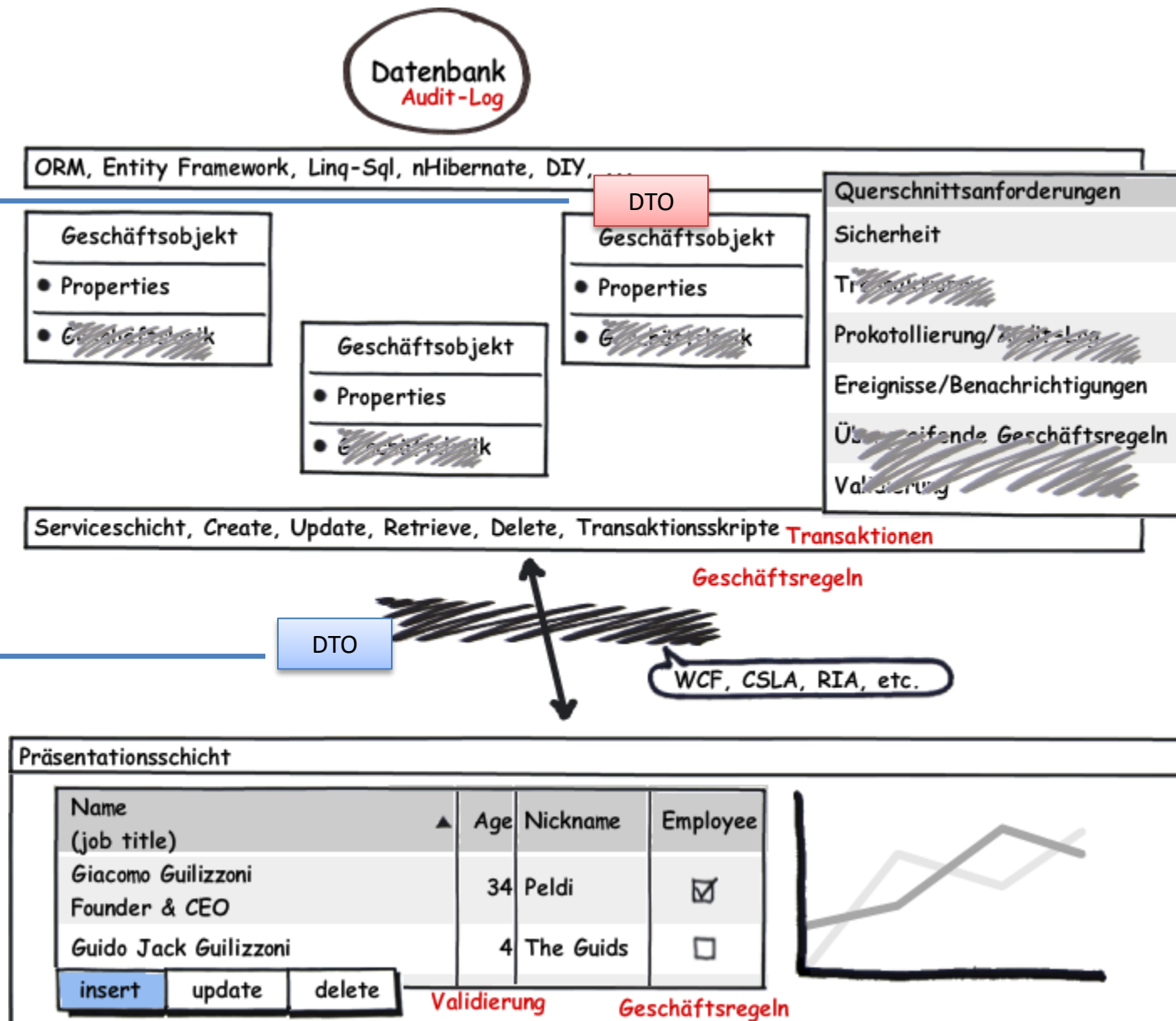
Geschäftsmodell

Dienste

Präsentation

Geschäftslogik

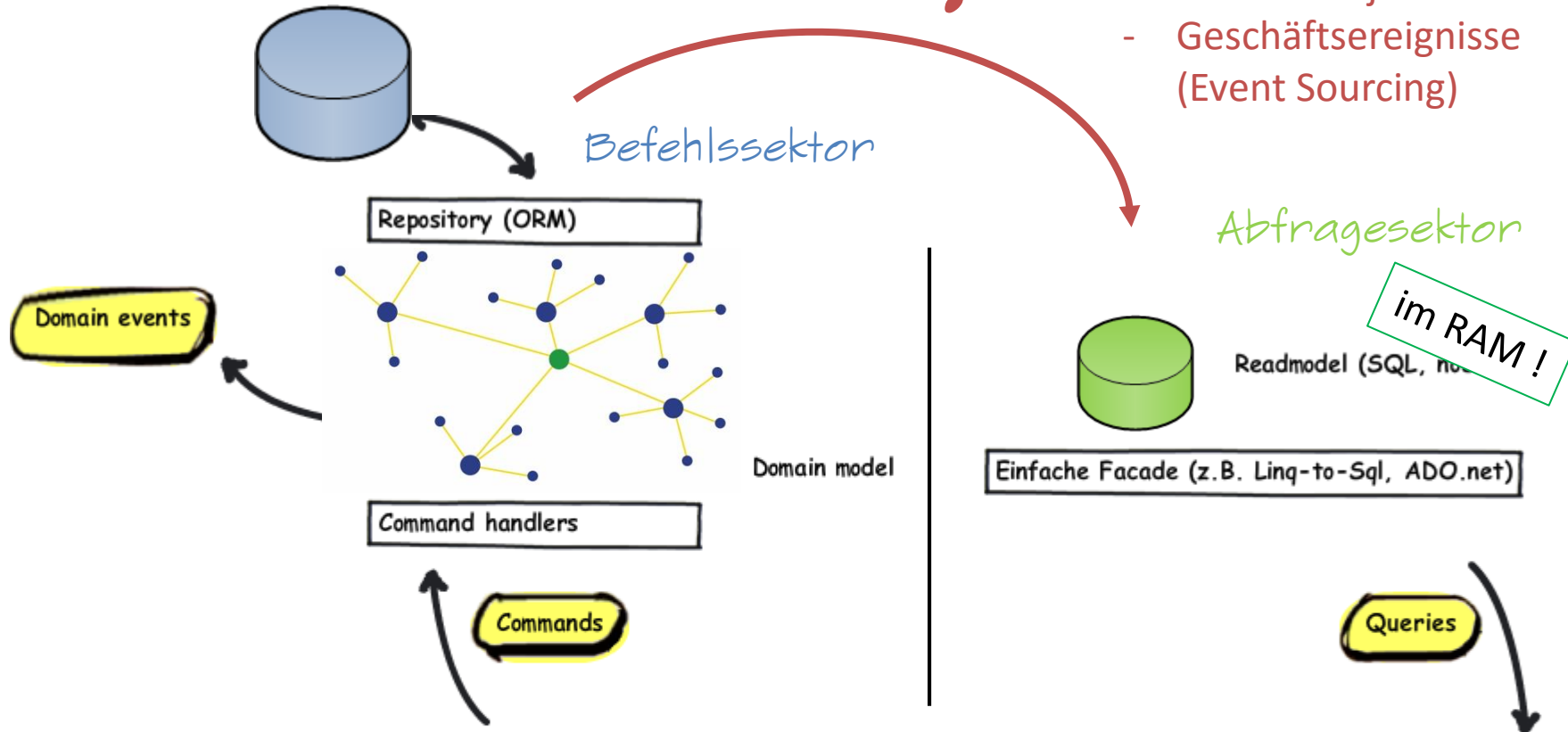
Benutzer



Teilsystem: Aufteilung C | Q

zwei Varianten:

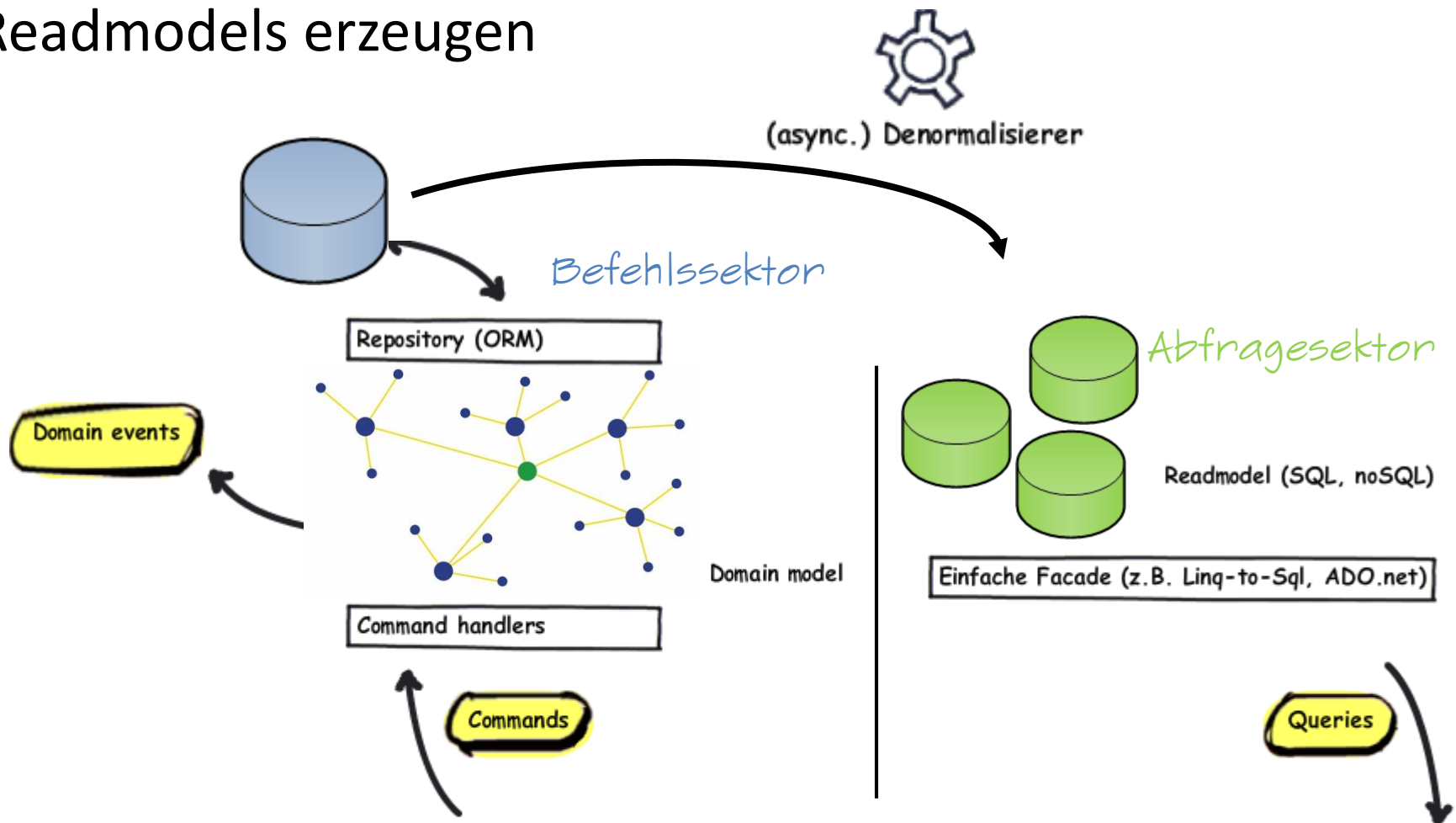
- direkte Projektion
- Geschäftsereignisse (Event Sourcing)



Präsentation

Präsentationsschicht				
Taskspezifische Befehle und Aktionen				
Name (job title)	Age	Nickname	Employee	
Giacomo Guilizzoni Founder & CEO	34	Peldi	<input checked="" type="checkbox"/>	
Guido Jack Guilizzoni	4	The Guids	<input type="checkbox"/>	

Readmodels erzeugen

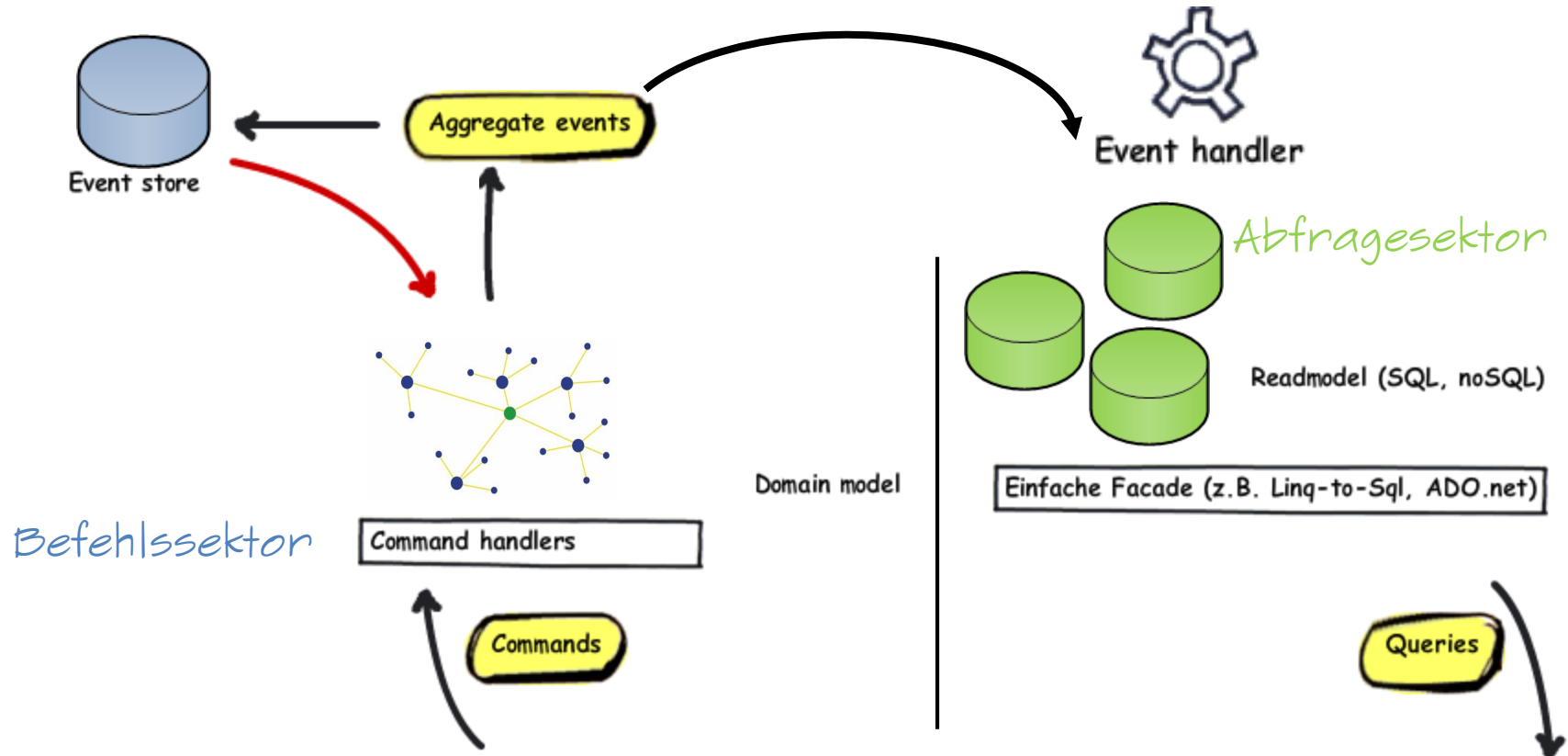


Präsentation

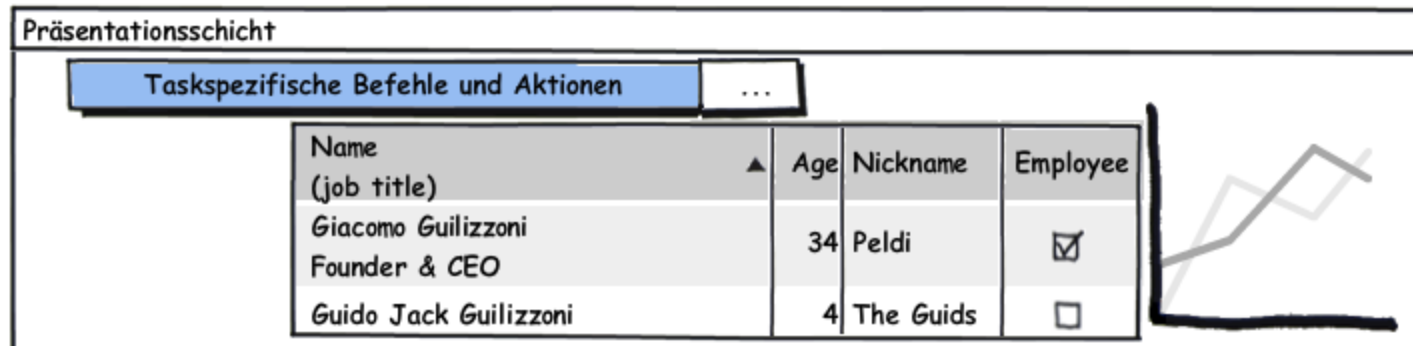
Präsentationsschicht				
Taskspezifische Befehle und Aktionen ...				
Name (job title)	Age	Nickname	Employee	
Giacomo Guilizzoni Founder & CEO	34	Peldi	<input checked="" type="checkbox"/>	
Guido Jack Guilizzoni	4	The Guids	<input type="checkbox"/>	

CQRS + Event Sourcing

Captures all changes to an application state as a sequence of events. [Fowler]



Präsentation



CQRS + Event Sourcing

Formulierung der fachlichen Domäne
ohne Implementierungsbelange
-> Event Storming...



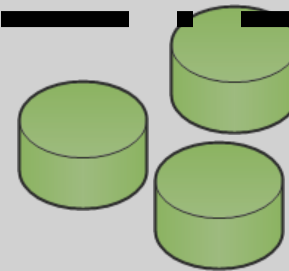
Event store

Aggregate events



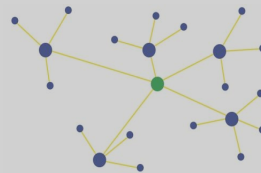
Event handler

Abfragesektor



Readmodel (SQL, noSQL)

Einfache Facade (z.B. Linq-to-Sql, ADO.net)



Domain model

Command handlers

Befehlssektor

Formulierung der Use Cases:

Welche Entscheidungen will
der Nutzer treffen?

Formulierung der Use Cases:

Welche Informationen werden
zur Entscheidungsfindung
benötigt?

Präsentation

Taskspezifische Befehle und Aktionen

Name (job title)
Giacomo Guilizzoni Founder & CEO
Guido Jack Guilizzoni

34	Peldi	<input checked="" type="checkbox"/>
4	The Guids	<input type="checkbox"/>

Queries

Gewinn

- Gekapseltes, verhaltensorientiertes Domänenmodell
- Modell drückt Intention des Anwenders aus
- Garantiertes Audit-Log mit vollständiger Historie
- Umgehen des Objekt-Relationalen Impedanz-Mismatches
- Frei optimierbare und verteilte Readmodels
- Testbarkeit in der Sprache der Domäne
- Leicht integrierbar in externe Systeme
- Kombinierbar: Domänenmodell + Transaktionsskript

Risiken und Nebenwirkungen

- Für viele Teams neue Funktionsprinzipien
- Gefühlte steile Lernkurve (auch wenns eigentlich einfach ist)
- Hoher Basisaufwand
überzogen für einfache CRUD-Anwendungen
- Duplikation von Daten im Readmodel
- Verleitet zu Asynchronität
- Andererseits: Google, Amazon, Netflix, Twitter, ... basieren auf vergleichbaren Architekturen

Zusammenfassung

- CQRS trennt Ausführung von Aktionen und Datenaufbereitung
- Ermöglicht ein fokussiertes Domänenmodell
- Legt den Fokus auf Befehle, Ereignisse und Zustandsänderungen

- CQRS+Event sourcing abstrahiert Verhalten vom Zustand
- ersetzt die Zustandsdatenbank durch eine Datenbank der Zustandsänderungen (mit Intention)
- Das Protokoll ist die Datenquelle

- Optimal für verteilte Systeme
und/oder komplexe volatile Anwendungsdomänen

Quellen und Verweise

- CQRS
 - Microsoft Patterns&Practices Group <http://aka.ms/cqrs>
 - Udi Dahan <http://www.udidahan.com/>
 - Greg Young <http://goodenoughsoftware.net/>
 - Jonathan Oliver, et al. <http://distributedpodcast.com/>
- <http://groups.google.com/group/dddcqrs>
- dotnetpro 5/2012
- Heise Developer - Architektur: „Getrennt sind wir stark, CQRS“ <http://bit.ly/10fY31m>
- DDDx Webcasts: www.skillsmatter.com

CQRS – Architektur in der Praxis

Trainer:	Philip Jander
Datum:	01.06.2016 - 02.06.2016
Uhrzeit:	09:00 - 18:00 Uhr
Preis:	Frühbucher: EUR 1.799,- zzgl. MwSt. Normalpreis: EUR 1.999,- zzgl. MwSt.
Ort:	Köln
Stadt:	Köln

[JETZT ANMELDEN!](#)[IN KALENDER SPEICHERN](#)

Beschreibung

Commands (Ausführen von Geschäftslogik) und Queries (Datenabfragen) haben grundlegend verschiedene nichtfunktionale Anforderungen. Daher erfordern sie getrennte Implementierungsansätze. Das Architekturmuster CQRS (Command/Query Responsibility Segregation) setzt diese Erkenntnis konsequent um. Durch die Aufteilung der Verantwortlichkeit für beide Hälften eines Systems gewinnen Sie die Freiheit, diese unabhängig zu entwerfen und auf unterschiedliche Ziele hin zu optimieren.

Das Training führt Sie durch CQRS, von der anfänglichen Grobskizze bis zu Optionen für die Implementierung der Geschäftslogik. Ausgerichtet auf C# 5 und .NET 4.5, entwickelt Philip Jander gemeinsam mit den Teilnehmern im Laufe des Trainings den Rahmen für die Überführung eines



Über Philip Jander

Philip Jander ist freier Software-Entwickler, IT-Berater und Trainer mit Schwerpunkt .NET. Als bekennender Agilist befasst er sich gerne mit Softwarearchitekturen für komplexe Anwendungsszenarien und mit funktionaler Programmierung. Die Erfahrungen aus der Projektarbeit gibt er in Vorträgen und Trainings weiter und begleitet Entwicklerteams in Softwareunternehmen bei Projekten und