

Microsoft experiences'16

Mercredi 5 octobre • Palais des Congrès de Paris

Thomas PIERRAIN
🐦 @tpierrain

Bruno BOUCARD
@brunoboucard

Tomasz JASKULA
@tjaskula

Eric VERNIE
@EricVernie

CQRS

(without Event Sourcing)

#experiences



Why CQRS



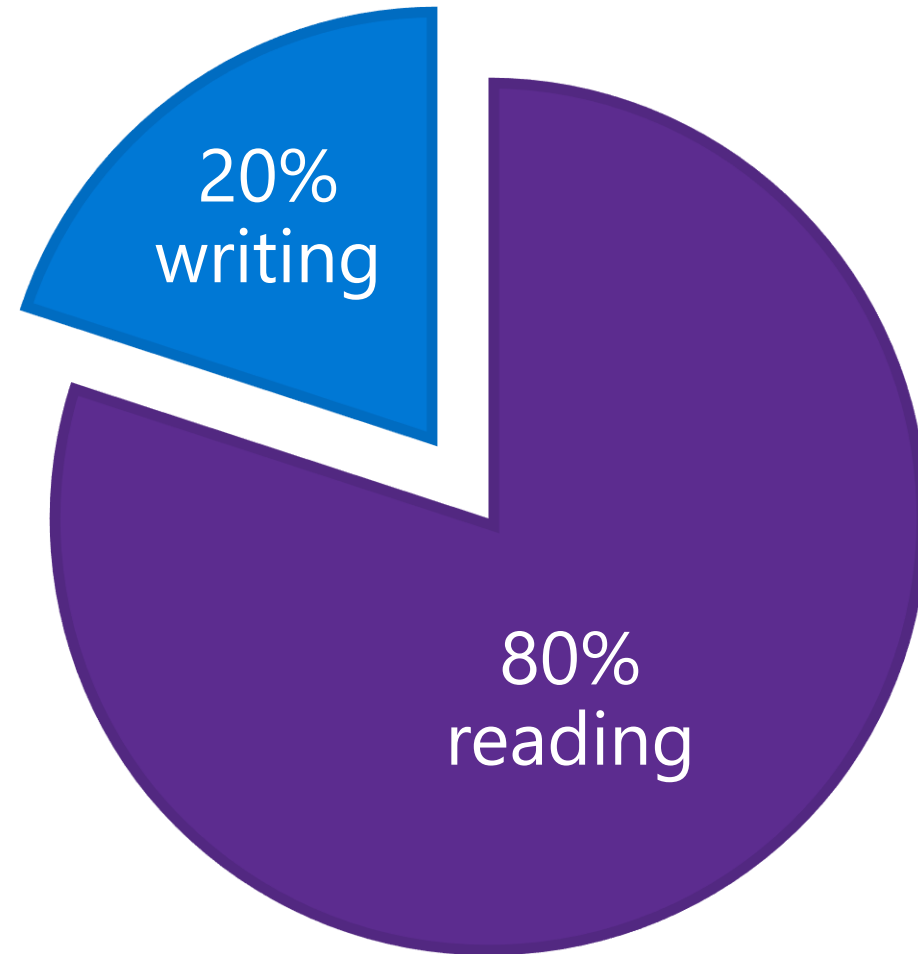
Scalability issues?



Root cause

The Pareto Principle

80-20



With...

...only **one central component** to deal with a wide range of constraints.



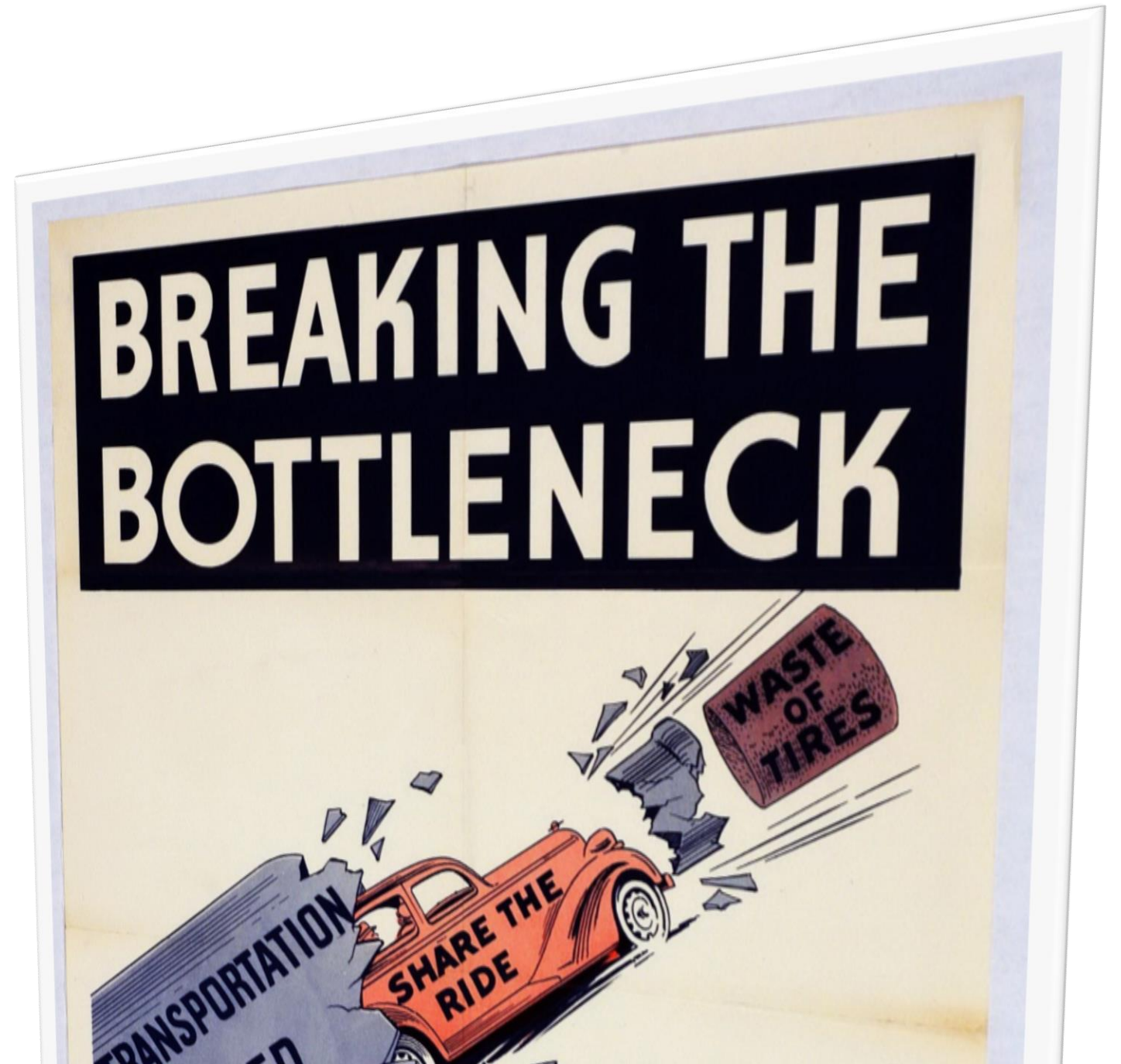
A question (for you)

*Do you have those constraints in mind
while designing your architectures?*

So...

*How to optimize an app that has to scale
for massive read accesses?*

**This is where
CQRS is...**



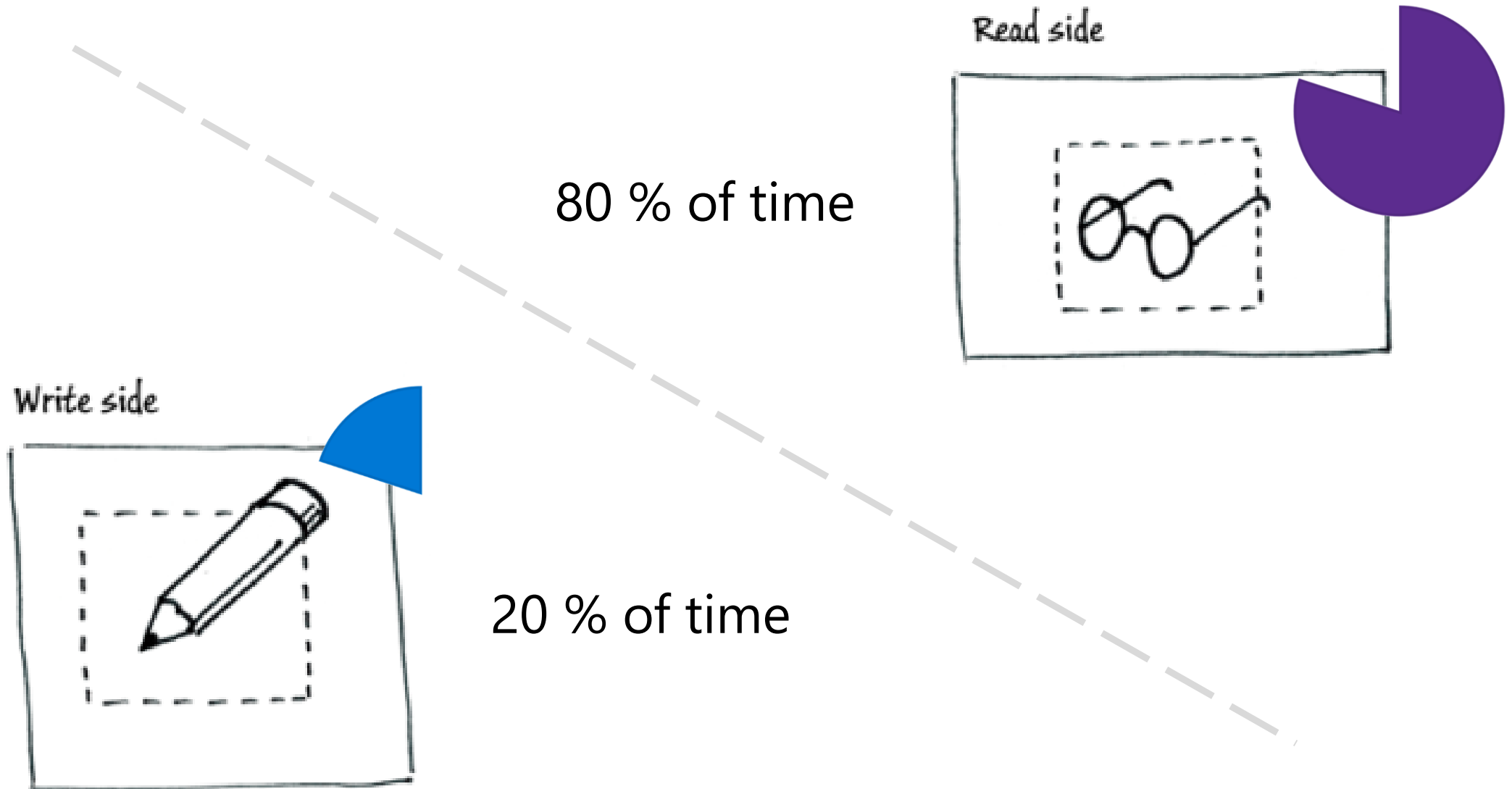
CQRS (short version)



CQRS splits the 80s and the 20s



CQRS splits the 80s and the 20s



Behind the acronym

(powered by Greg YOUNG - @gregyoung)

C

Q

R

S

Behind the acronym

Command

Query

Responsibility

Segregation

Behind the acronym

C*ommand*

Q*uery*

R*esponsibility*

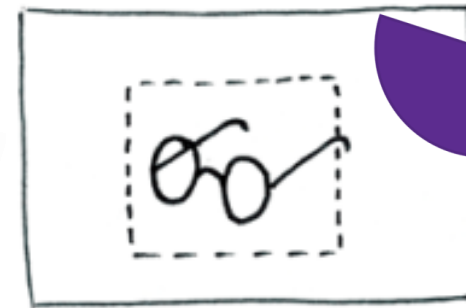
S*egregation*



Write side



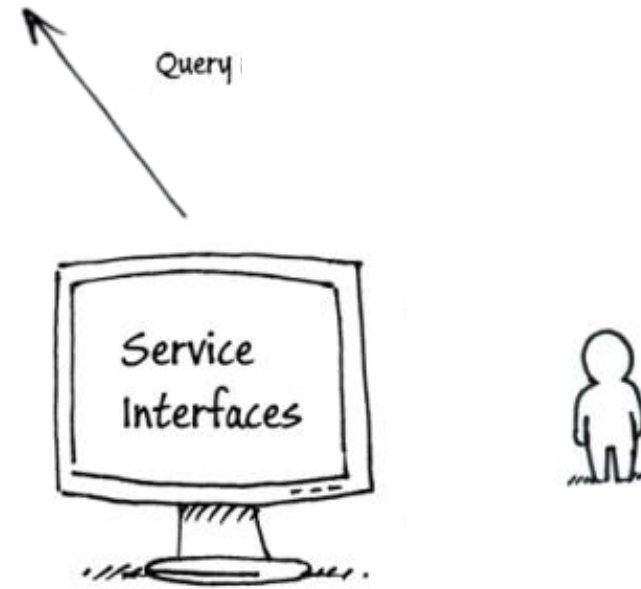
Read side



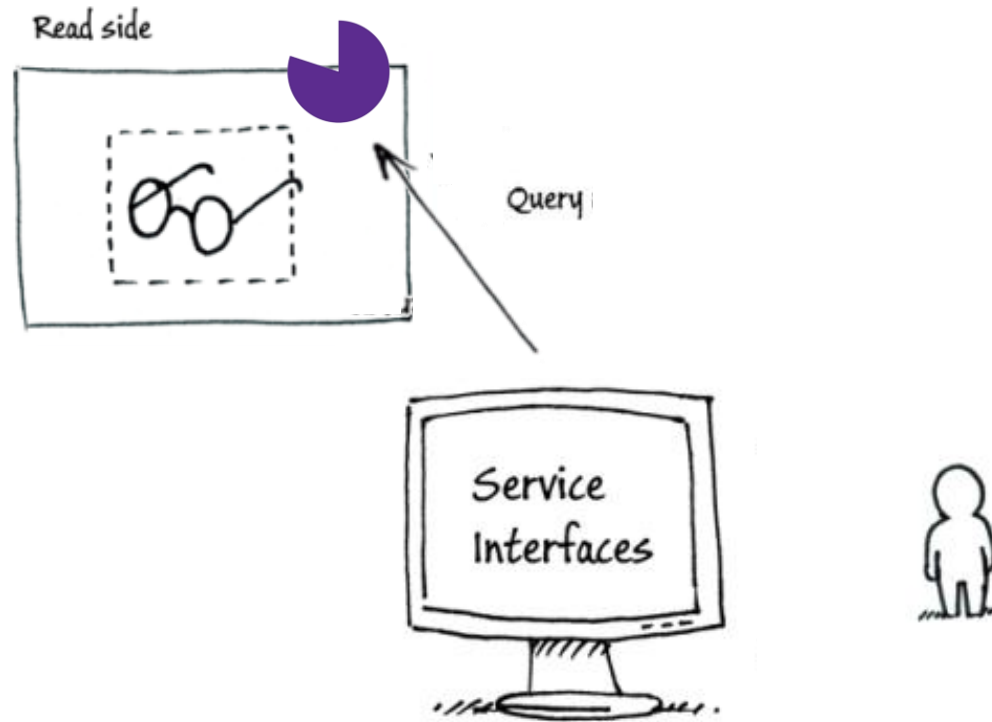
CQRS illustrated



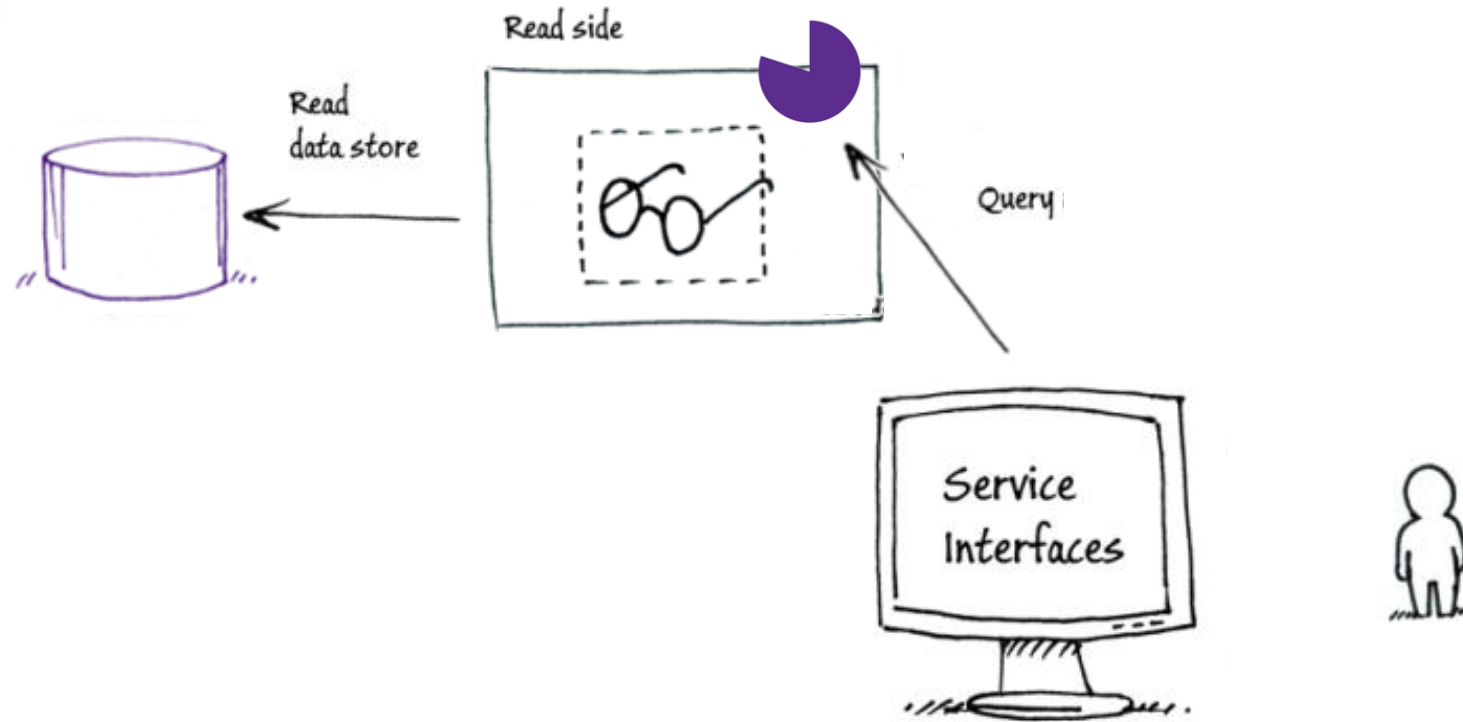
CQRS illustrated



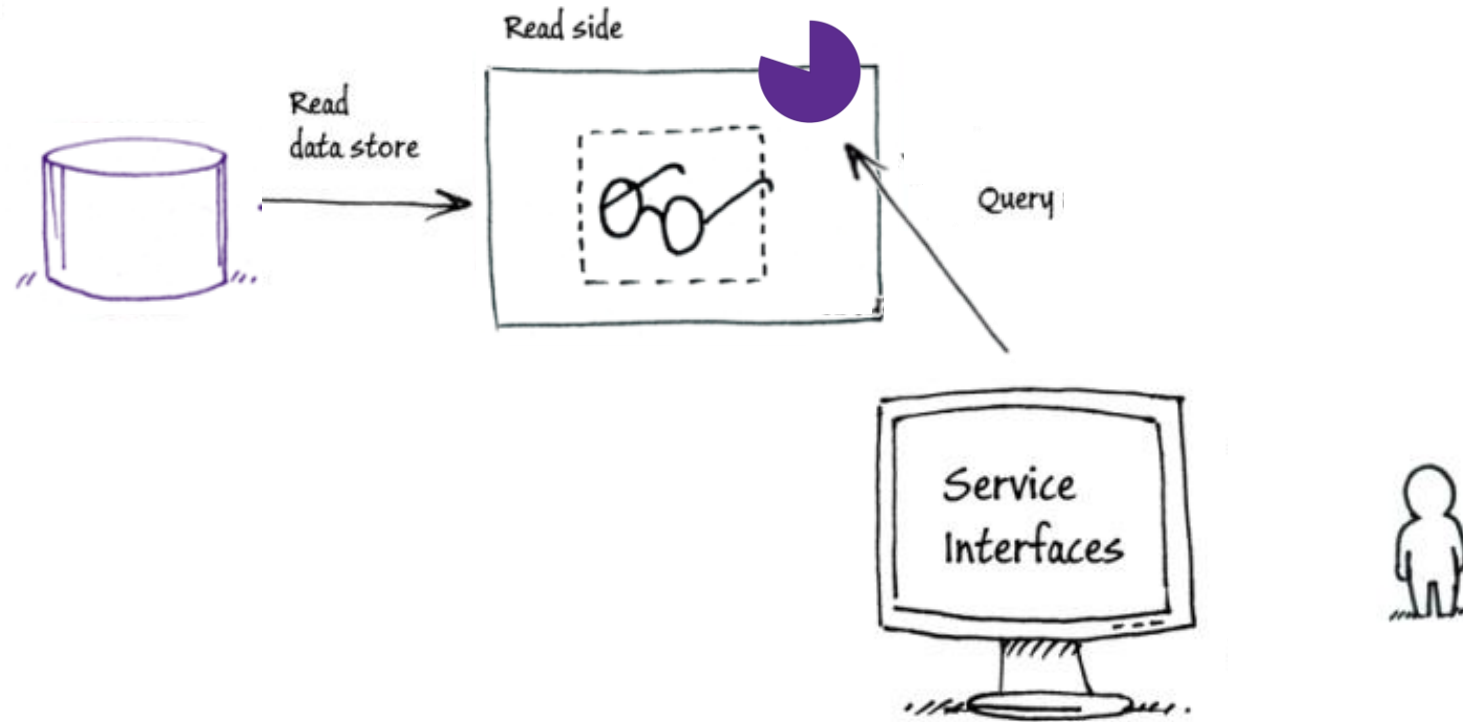
CQRS illustrated



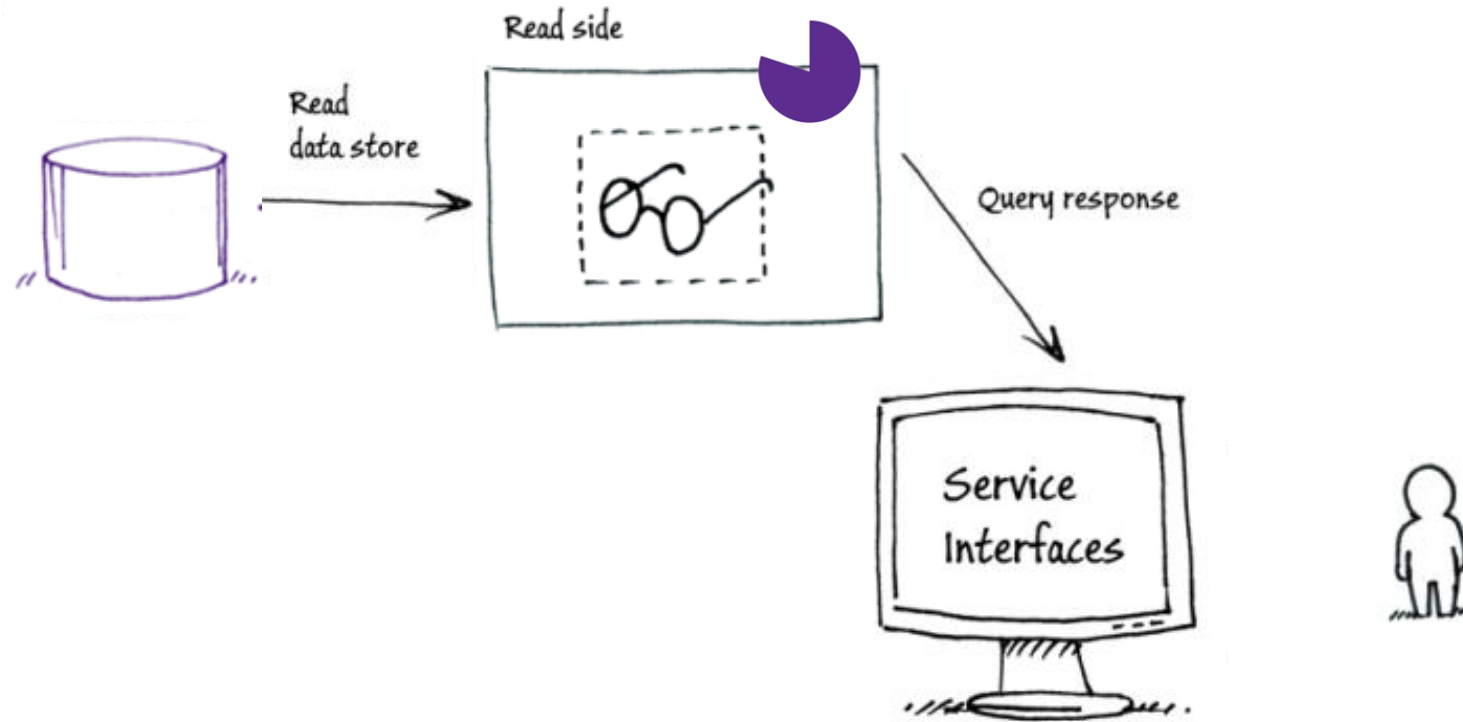
CQRS illustrated



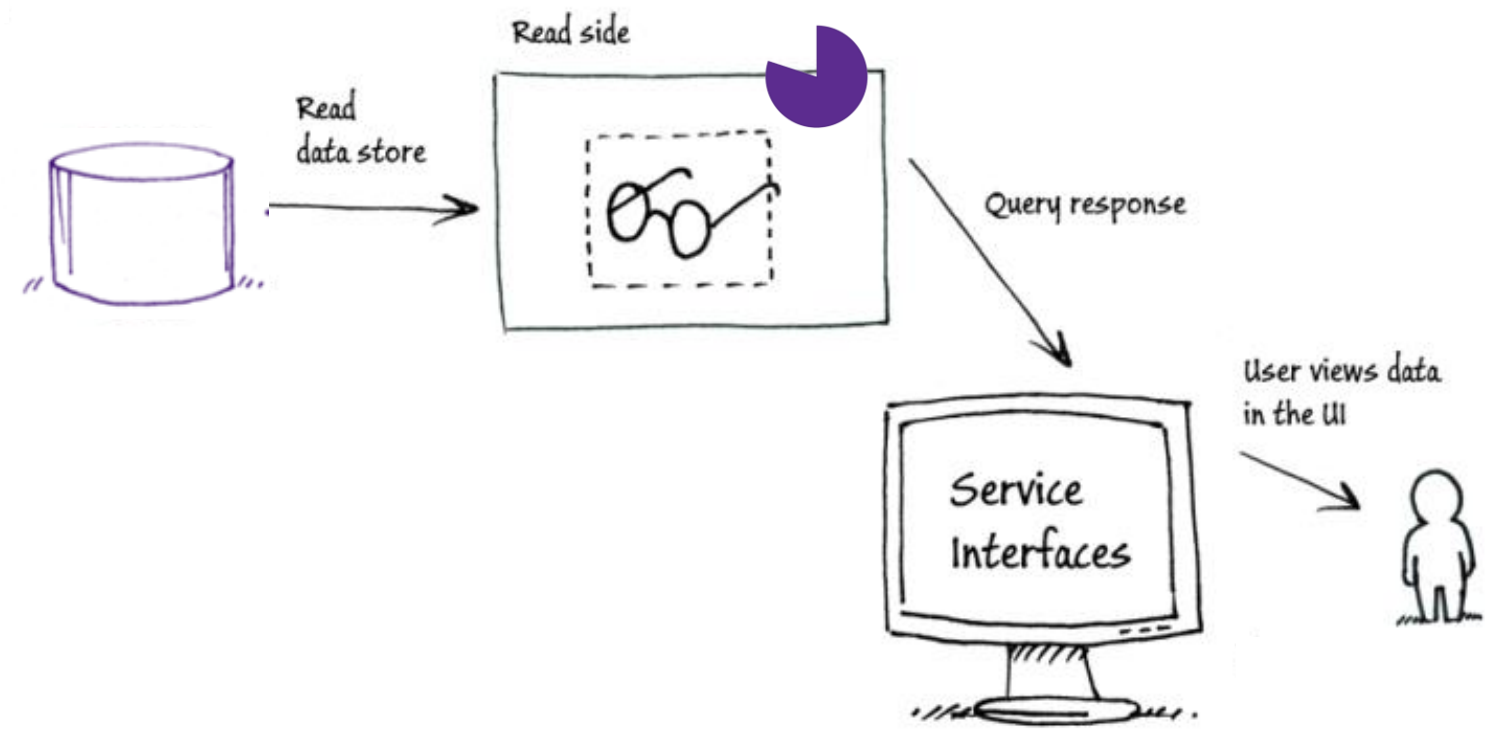
CQRS illustrated



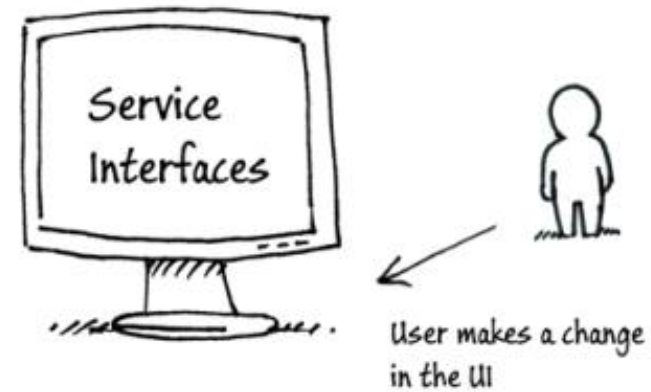
CQRS illustrated



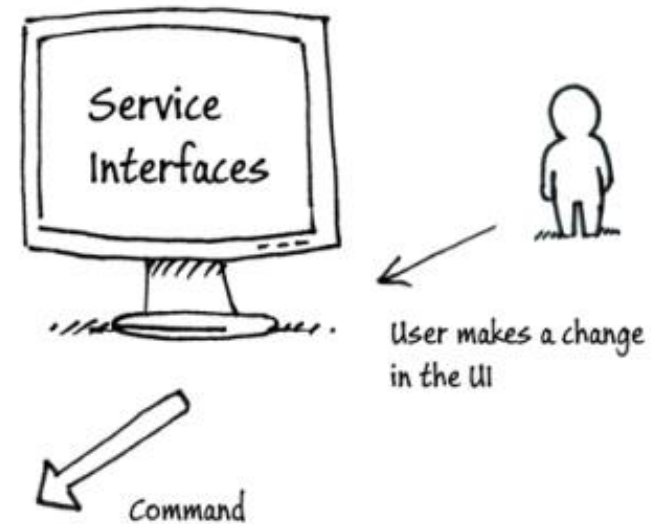
CQRS illustrated



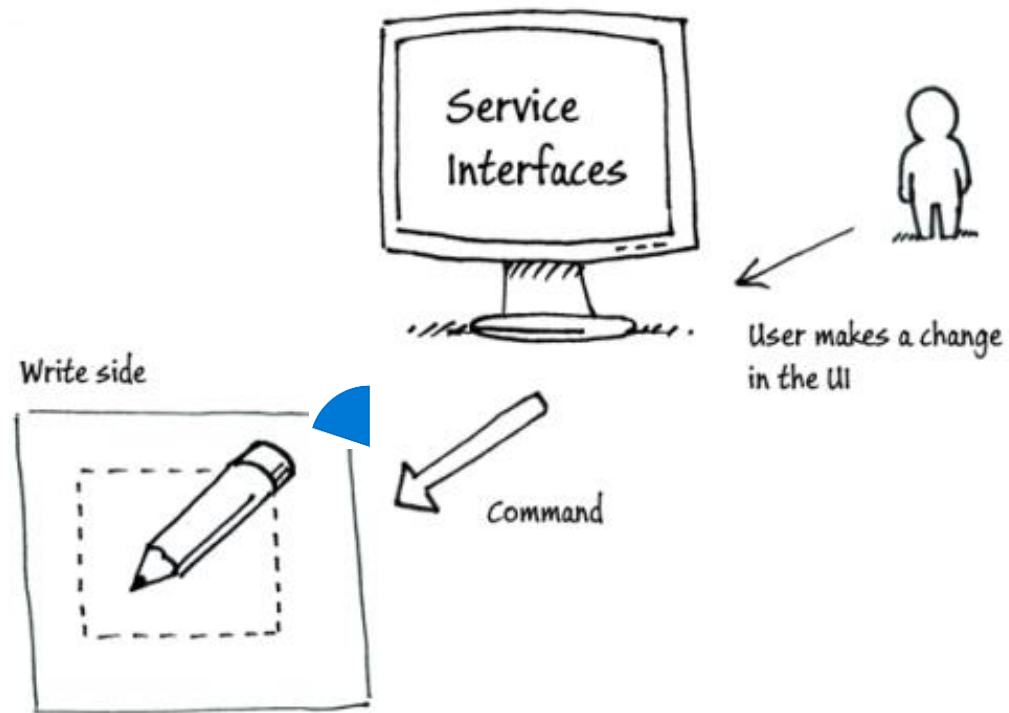
CQRS illustrated



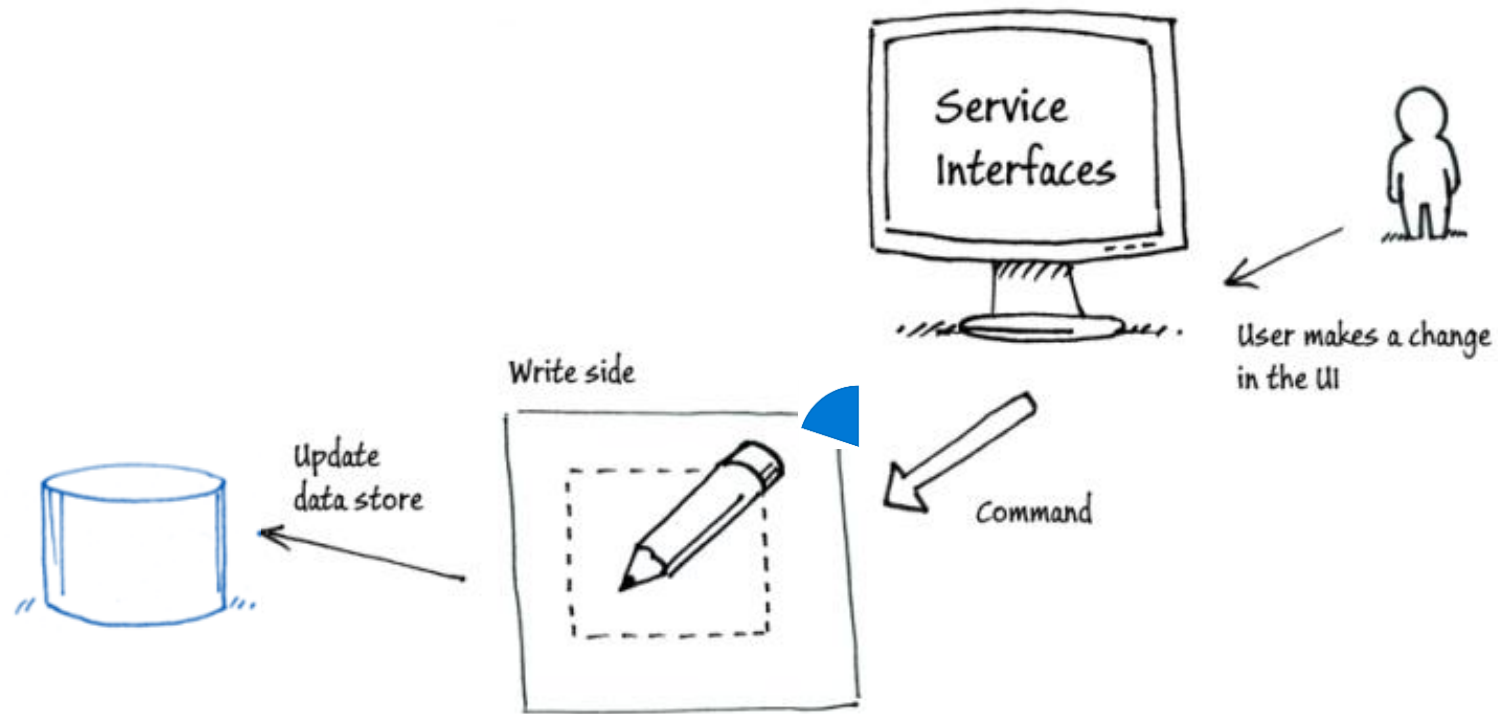
CQRS illustrated



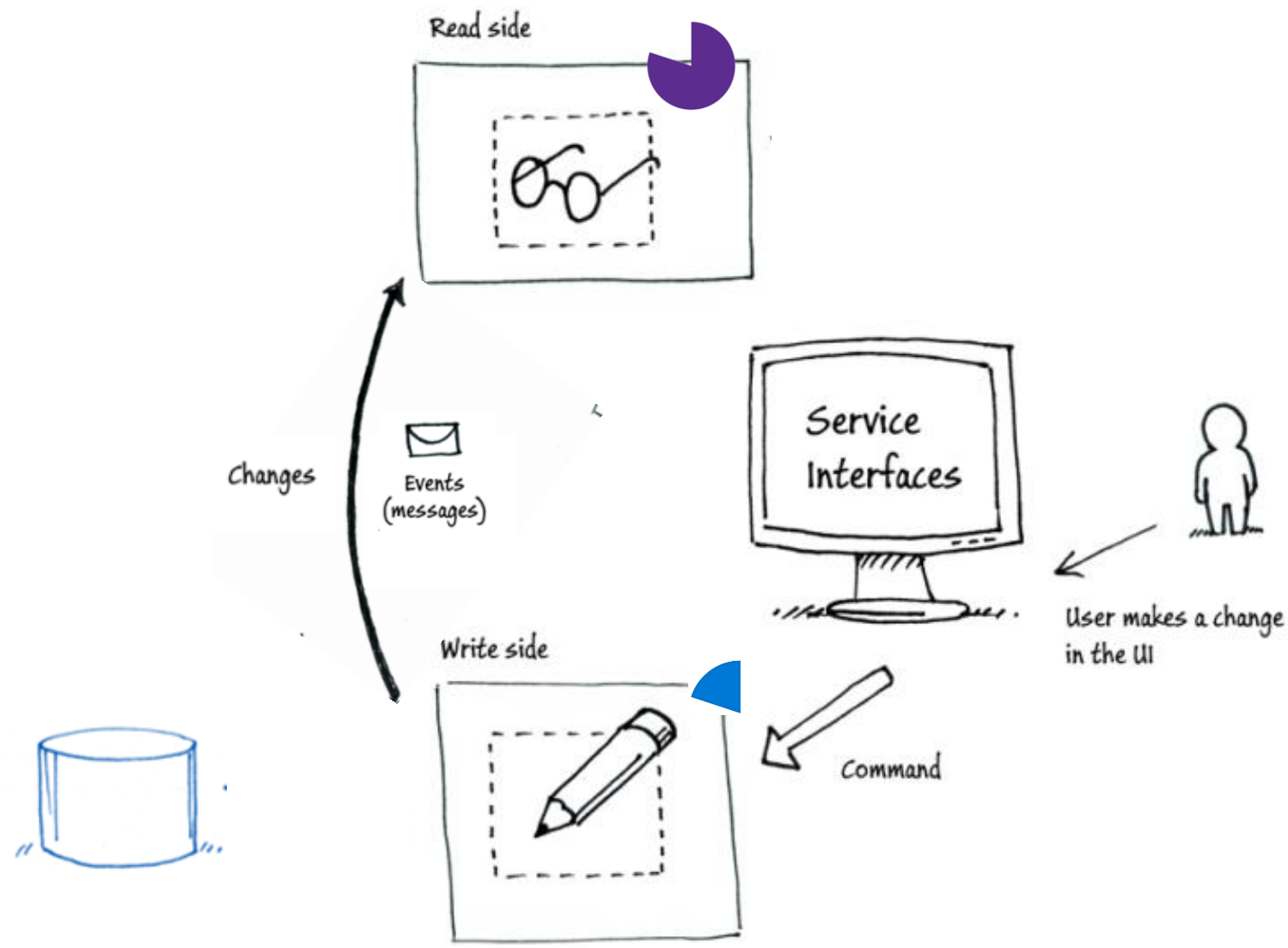
CQRS illustrated



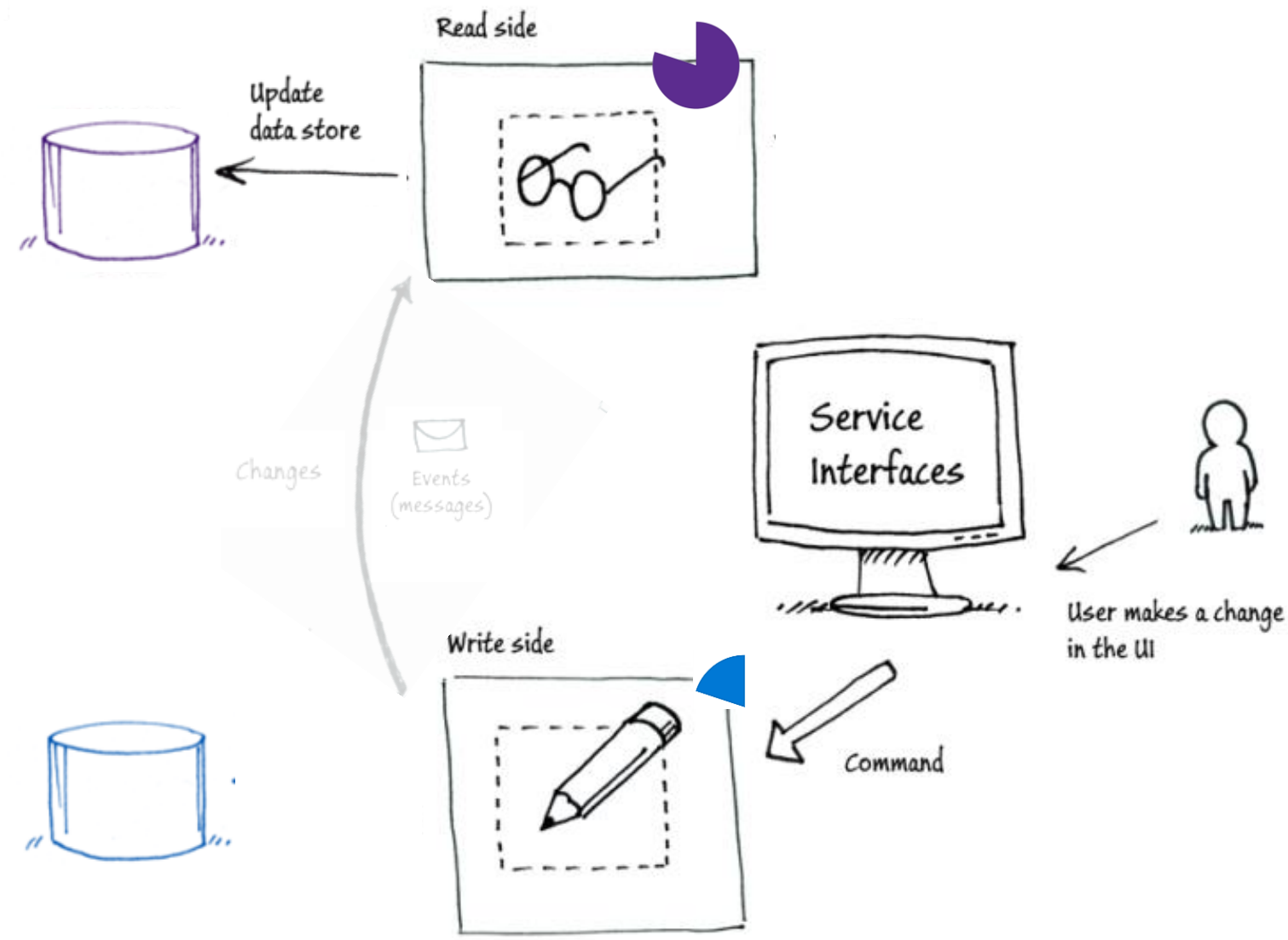
CQRS illustrated



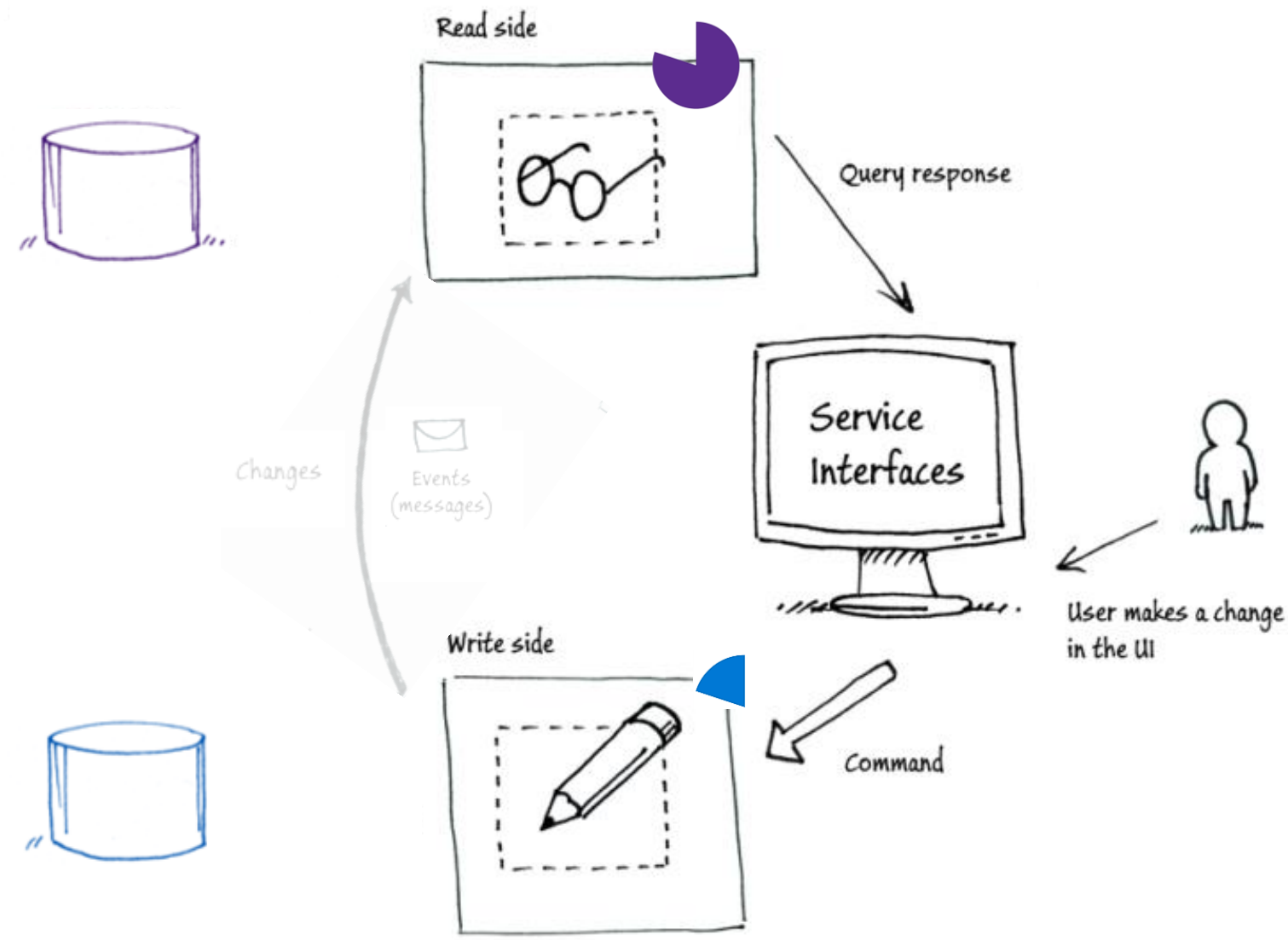
CQRS illustrated



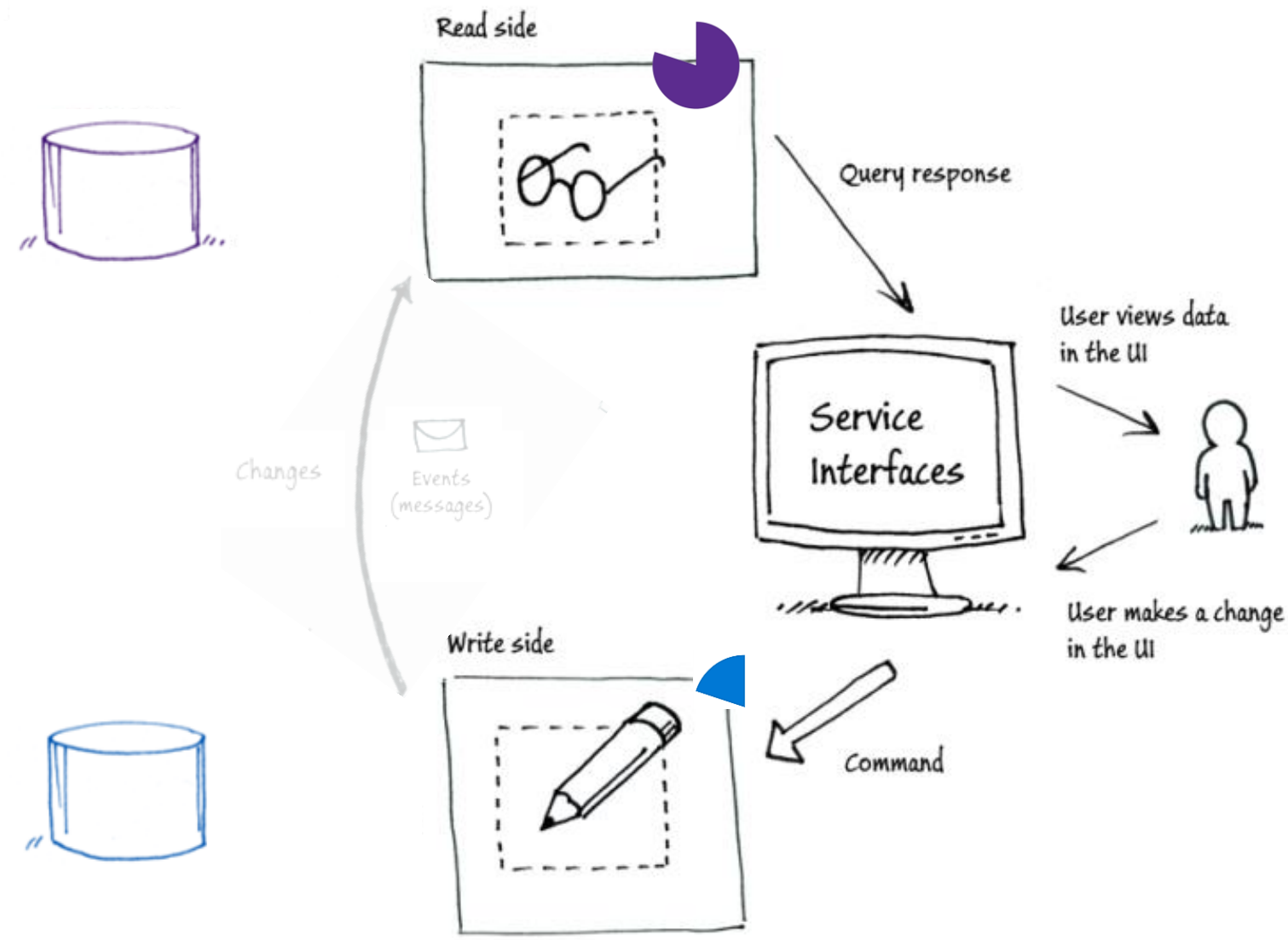
CQRS illustrated



CQRS illustrated



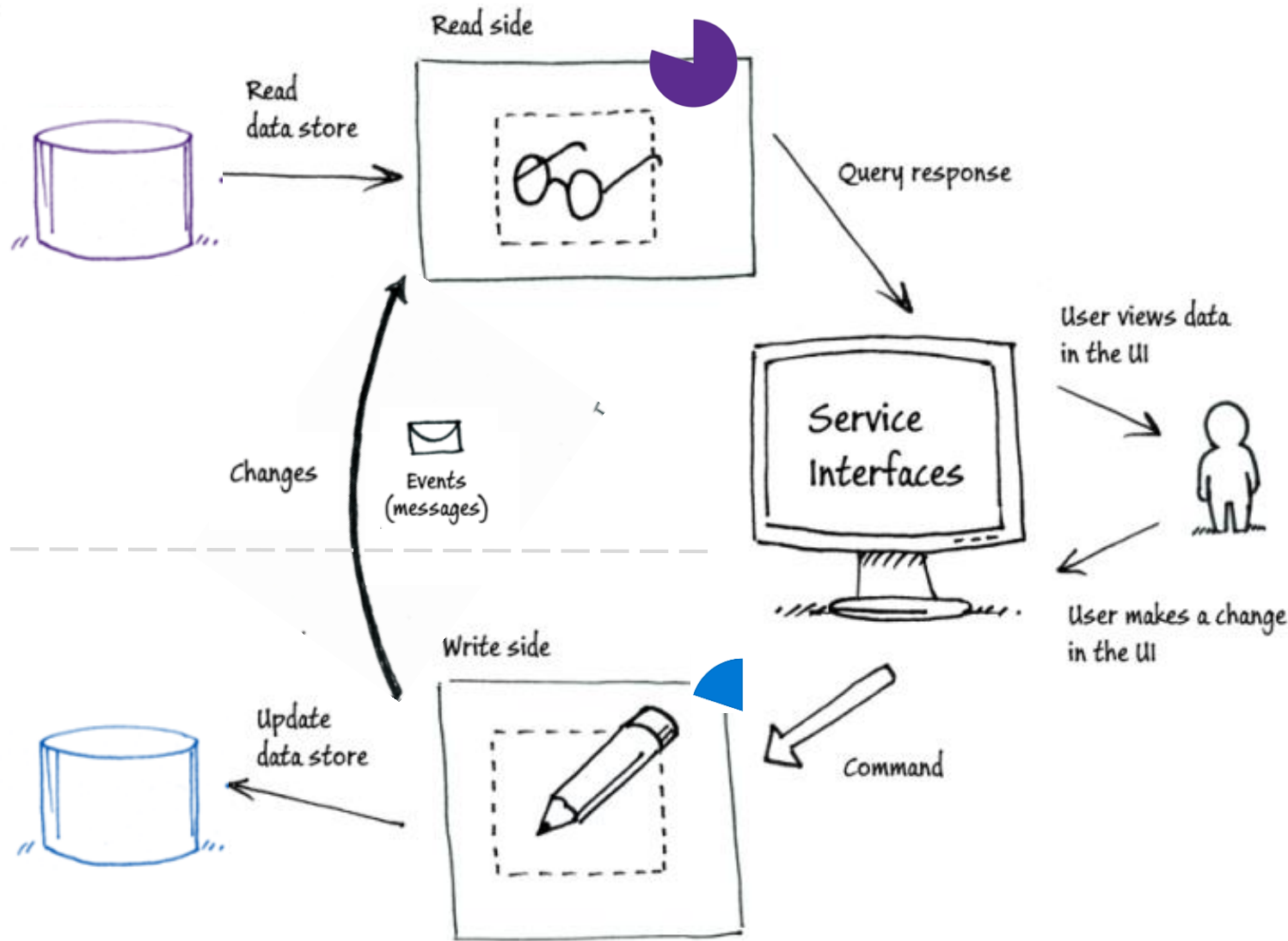
CQRS illustrated



CQRS illustrated

« ready-to-eat data », Caches, non-normalized models, (no SQL)

Classical transactional databases RDBMS (ACID)



Command

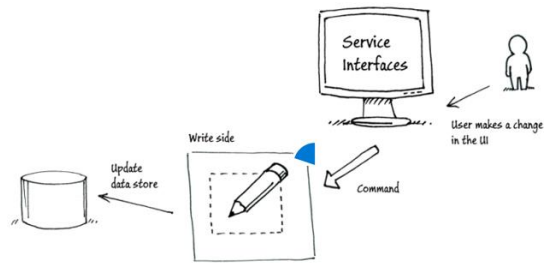
TRIGGERS AN ACTION

Modify the state of the system

>> Don't return data ! <<

Imperative Verb

Ex: **BookARoom**



Query

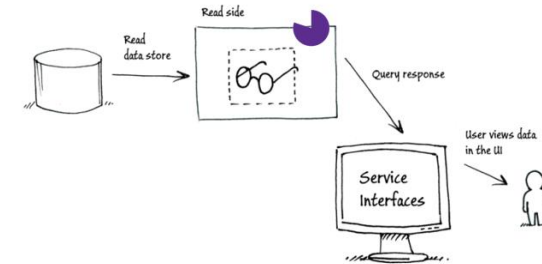
ASKS FOR AN INFORMATION

Read-only!

Get back data

Imperative Verb

Ex: **GetAvailableRooms**



Event

ALREADY HAPPENED

(Immutable)

Past-tense

Ex: **RoomBooked**

Demo



.NET Core

// Licensed to the .NET Foundation under one or more agreements.

Open Source Implementation available on GitHub

2 main parts:

CoreClr (Runtime execution) (C++)

CoreFx (including classes, collections, file systems, xml, etc.) (C#)



ASP.NET Core 1.0 framework relies on it

Classic deployment or

Container deployment (Linux or Windows)





BookARoom

 (a simple **CQRS** project)

Let's find you an hotel to stay!

Destination:

Check-in:

Check-out:

Number of adults:

Search

© 2016 - BookARoom



You would like to go to: **"Budapest"**

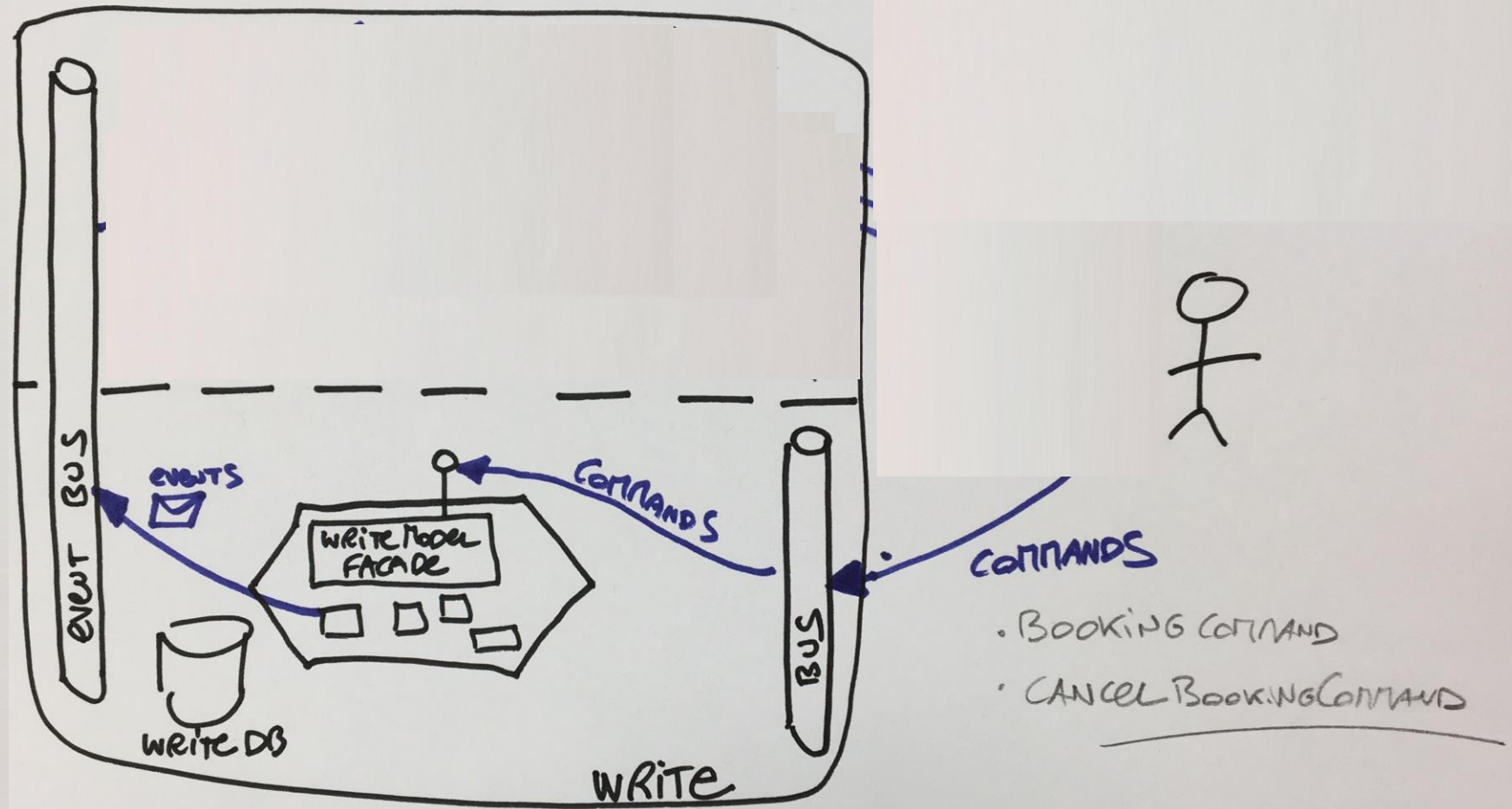
and to book a room for **2 adult(s)** with check-in date: **16/09/2017** and check-out date: **17/09/2017**

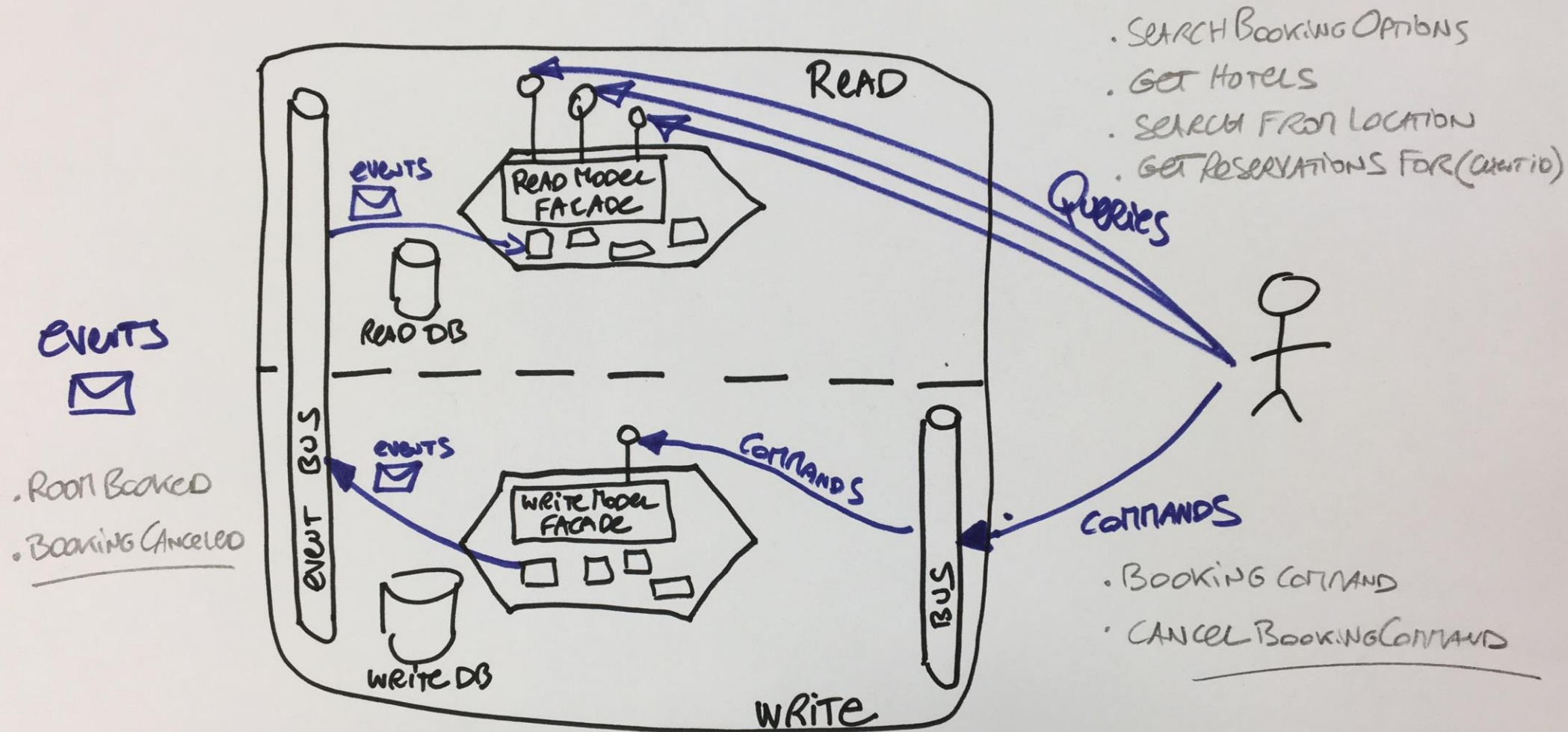
Oups, let me change something.

Here is the list of all available booking options for you. Just click the button to book-a-room!

Hotel name	Room code	One adult occupancy price	Two adults occupancy price	Book
Danubius Health Spa Resort Helia	101	109 EUR	140 EUR	Book that room!
THE GRAND BUDAPEST HOTEL	101	109 EUR	140 EUR	Book that room!
THE GRAND BUDAPEST HOTEL	102	109 EUR	140 EUR	Book that room!
THE GRAND BUDAPEST HOTEL	201	209 EUR	240 EUR	Book that room!

© 2016 - BookARoom





Live coding



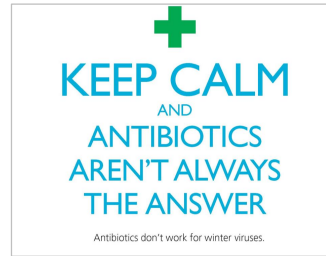
Wrap-up



Warning!



CQRS is not...



...a top-level architecture pattern

Use it for components under stress

Do not generalize its usage for all the components of your platform!

Also, CQRS is not...

Event Sourcing

CQRS ❤️ *Event Sourcing*

but

CQRS \neq *Event Sourcing*

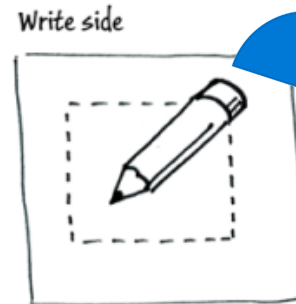
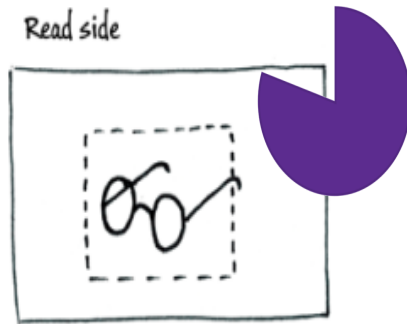
To conclude...

CQRS is mostly

*Well-separated **read** and **write** concerns*

Queries

*Caches, Ready-to-eat data
(no SQL)*



Commands

*Transactionnal
ACID*



Microsoft
experiences'16
Palais des Congrès de Paris

MERCI!!

#experiences

Merci aussi à **Rui CARVALHO, Mendel MONTEIRO-BECKERMAN, Olivier COANET et Clément BOUDEREAU** ainsi que **Greg YOUNG** pour leurs feedbacks et idées pour ce talk.

CQRS origins



Origine de CQRS



use case driven @tpierrain · 29 Sep 2016

Hi @gregyoung , we have a debate here ;-) for me the main drivers for CQRS was scalability & push (with domain reasoning as side-effect)1/2



1



use case driven @tpierrain · 29 Sep 2016

@gregyoung ...but I heard friends saying that it's the other way (#CQRS scalability being only a side effect for them). What's your view?2/2



1



1



Origine de CQRS



gregyoung @gregyoung · 29 Sep 2016

@tpierrain @ThierryAbalea no it was a common way of building systems (transaction log + subscribers). Ours was perf, disruptor similar



1



1



use case driven @tpierrain · 29 Sep 2016

@gregyoung That's what I understood. Thanks a lot for the replies.



1



Origine de CQRS



gregyoung @gregyoung · 29 Sep 2016

@tpierrain @ThierryAbalea it was only later tried on many different types of systems that many other benefits started to be shown.



1



The background image shows a large audience seated in a conference hall, facing a stage. The stage is lit with blue and white spotlights. A speaker is visible on the right side of the stage, standing near a podium. The audience is diverse and appears to be listening attentively.

Wanna play the lab @home?

Follow the white rabbit here:

<https://github.com/tpierrain/CQRS/blob/master/LabInstructions.md>



Poursuivons la discussion avec
#experiences

@tpierrain @brunoboucard @tjaskula @EricVernie

@microsoftfrance @Technet_France @msdev_fr

Et merci de noter la session !

