

# УЕБ-БАЗИРАНО ПРОГРАМИРАНЕ

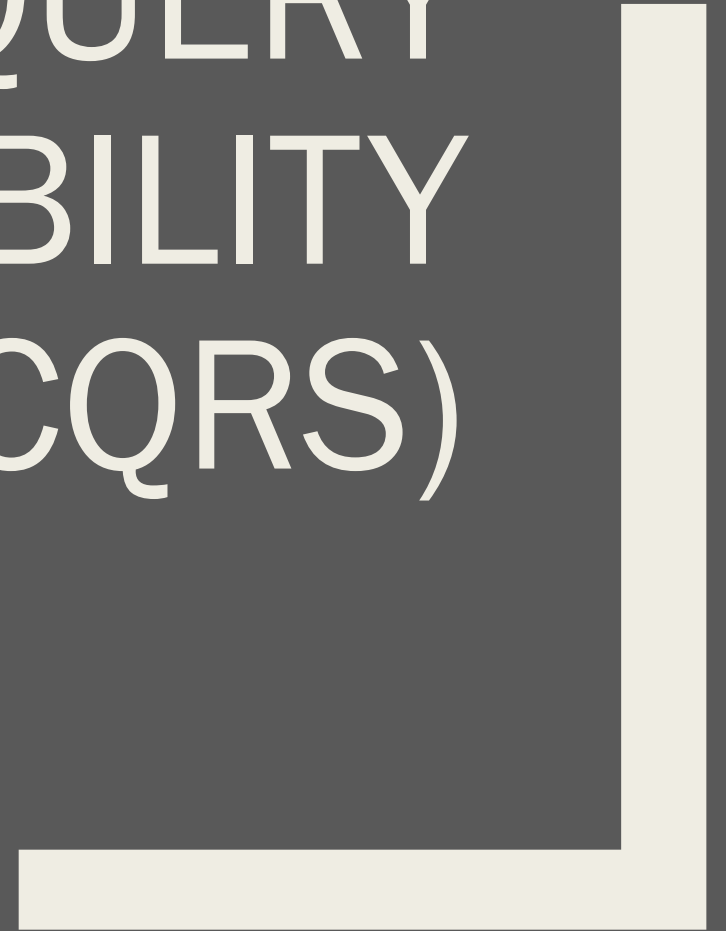
Павел Кюркчиев

Ас. към ПУ „Паисий Хилендарски“

<https://github.com/pkyurkchiev>

@pkyurkchiev

# COMMAND QUERY RESPONSIBILITY SEGREGATION (CQRS)



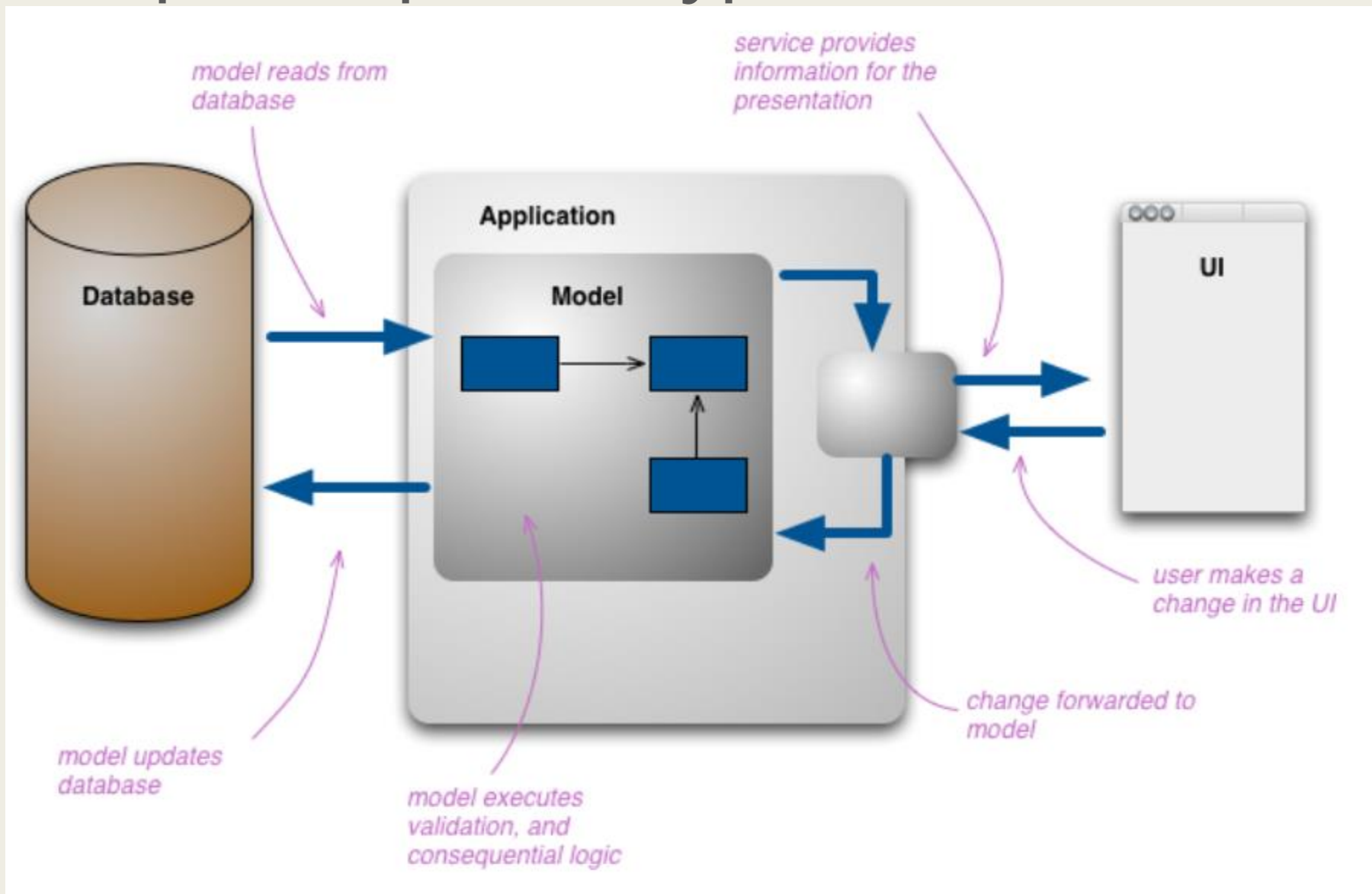
# Какво е Command Query Responsibility Segregation (CQRS)?

- CQRS представлява шаблон за дизайн. Идеята му е разделянето на операциите за работа с информация на две основни групи (за четене и за писане).

# Основни компоненти при изграждането на приложения с CQRS

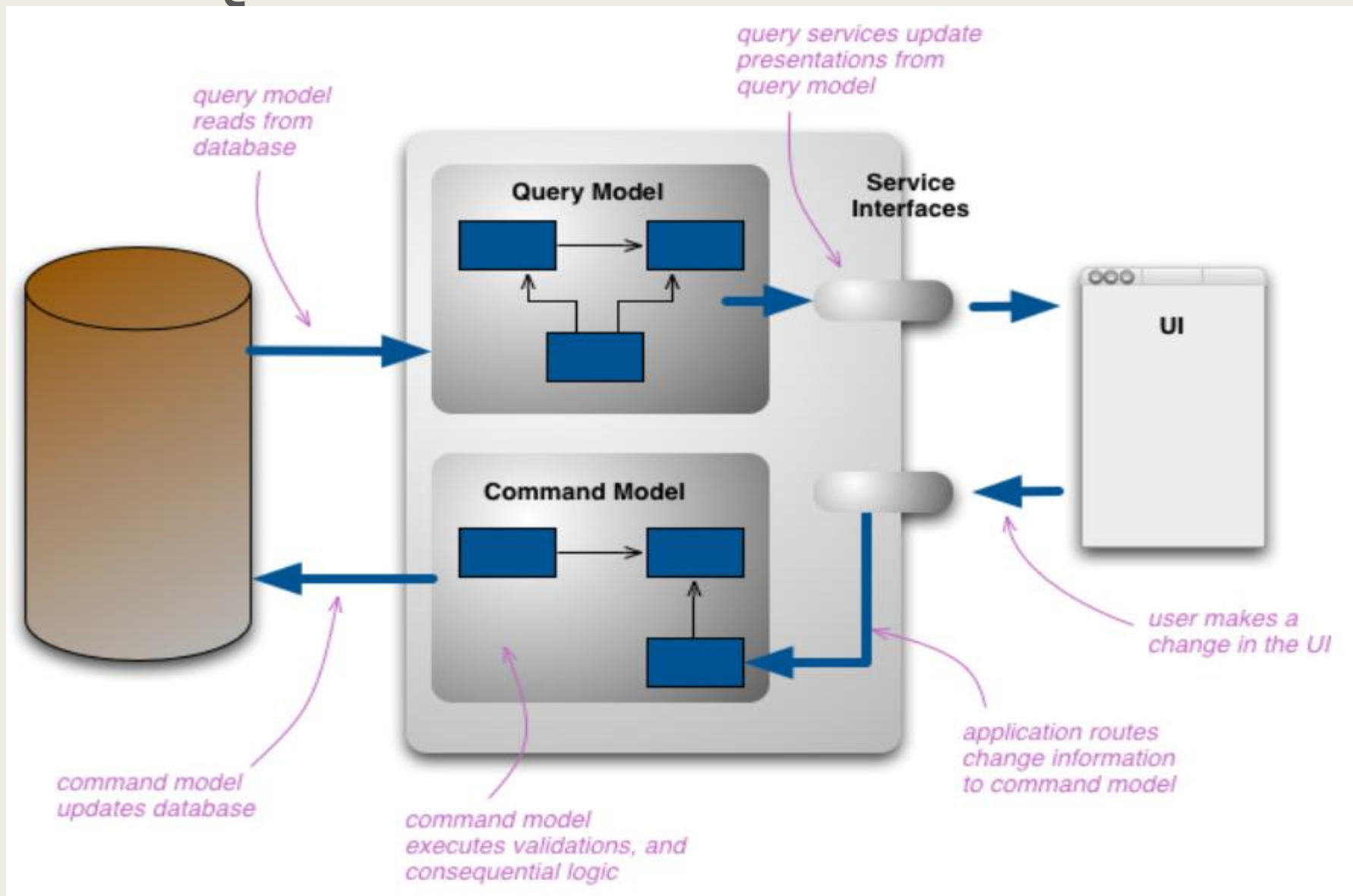
- Command model – модел обработващ операциите за модификация на информацията
- Query model – модел отговарящ за извличането на информация

# Стандартна архитектура



Стандартна архитектура представена  
с шаблона CQRS

# Шаблон CQRS



## CQRS естествено се вписва в някои други архитектури

- Чрез CQRS лесно се измества фокуса от единична имплементация, която работи с CRUD операциите на системата. Това позволява по – лесно преминаване към task-based UI.
- При голяма част от домейните е необходима сложна и комплексна бизнес логика за да бъде актуализирана информацията, в този случай е добре да бъде използвано EagerReadDerivation с цел опростяване модела на запис.

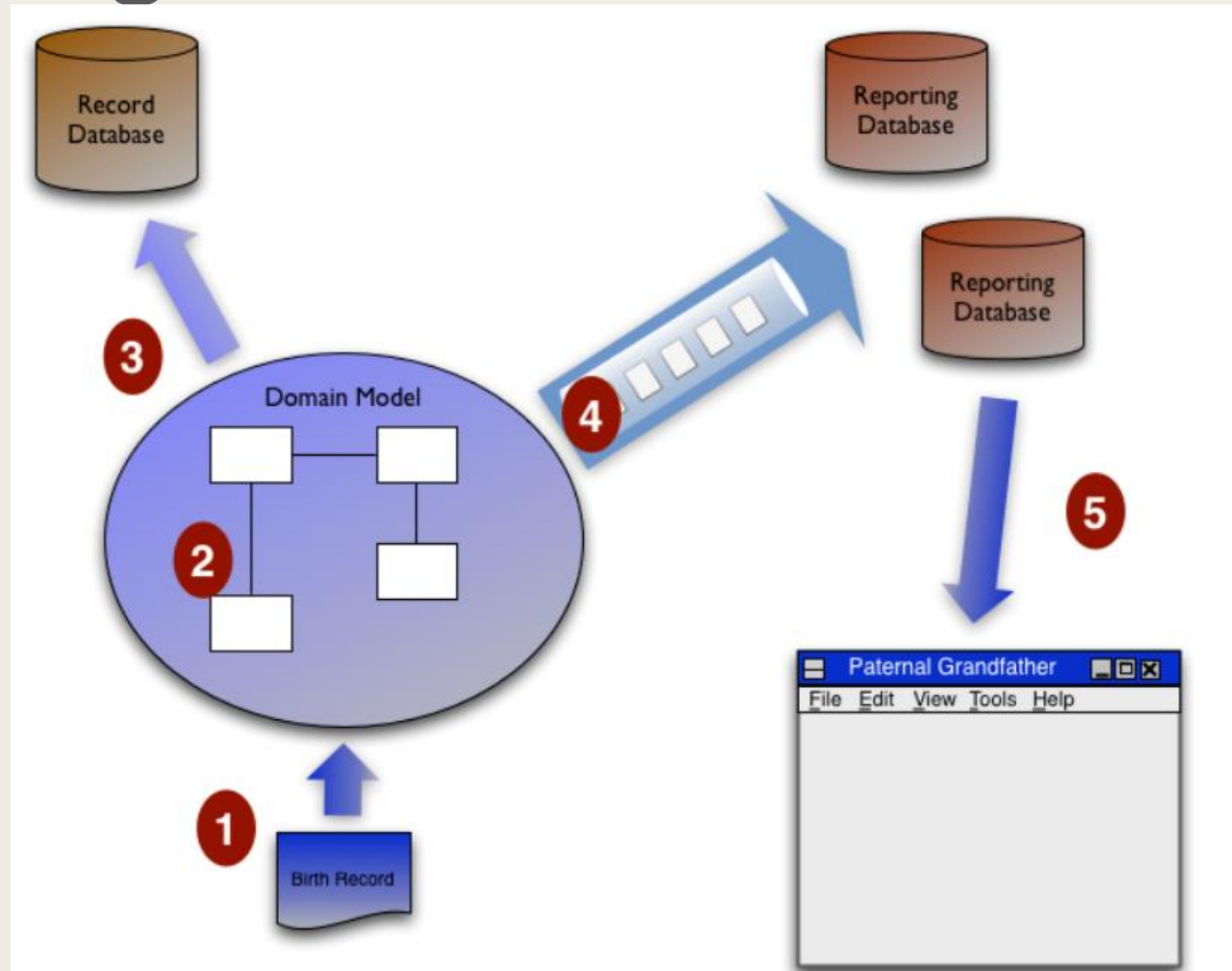


- Ако моделът за запис генерира събития за всички актуализации, можем да структурираме EventPosters при четенето, което позволява да се използват MemoryImages. По този начин може да се избегне честото взаимодействие с база данни.
- CQRS е подходящ за комплексни домейн решения, от които се възползва и DDD.

# Какво представлява EagerReadDerivation?

- Eager read derivation е шаблон, при който четенето не докосва основната база данни. За сметка на това ние четем от една или повече бази данни за отчети (Reporting Databases), като тези бази са структурирани по същия начин, по който са структурирани заявките за четене.

# Cxema EagerReadDerivation



# Какво представлява EventPosters?

- Event Posters са приложения, които в основата си се занимават със справки, подаващи към потребителя информация в реално време за състоянието на нещо (конкретната среда). Това са активни приложения, в които потребителя има контрол върху това каква информация иска да получи.

# Какво представлява MemoryImages?

- Memory Images представляват данни или информация, която е запазена в паметта, а не в конкретна база данни. Най – често се използва в комбинация със събитийни системи.
- Всяка промяна в състоянието на приложението се запазва като събитие, което се съхранява в устойчиво хранилище. Пълното възстановяване на състоянието на приложението представлява извикване на събитията в техния ред на създаване.

# Кога да използваме CQRS

- Имаме сложни домейни от информация, и четенето и писането могат да бъдат разделени логически.
- Подобряване работата на приложенията. Постигане на висока производителност. Позволява индивидуално повишаване на ресурсите (scaling) на отделните части.

# DEMO CQRS

<https://github.com/pkyurkchiev/web-programming-biel/tree/master/examples/ParkingManagement>

ВЪПРОСИ ?

