THE *WHAT* AND *WHY* OF CQRS

..in IT Logistics Fashion

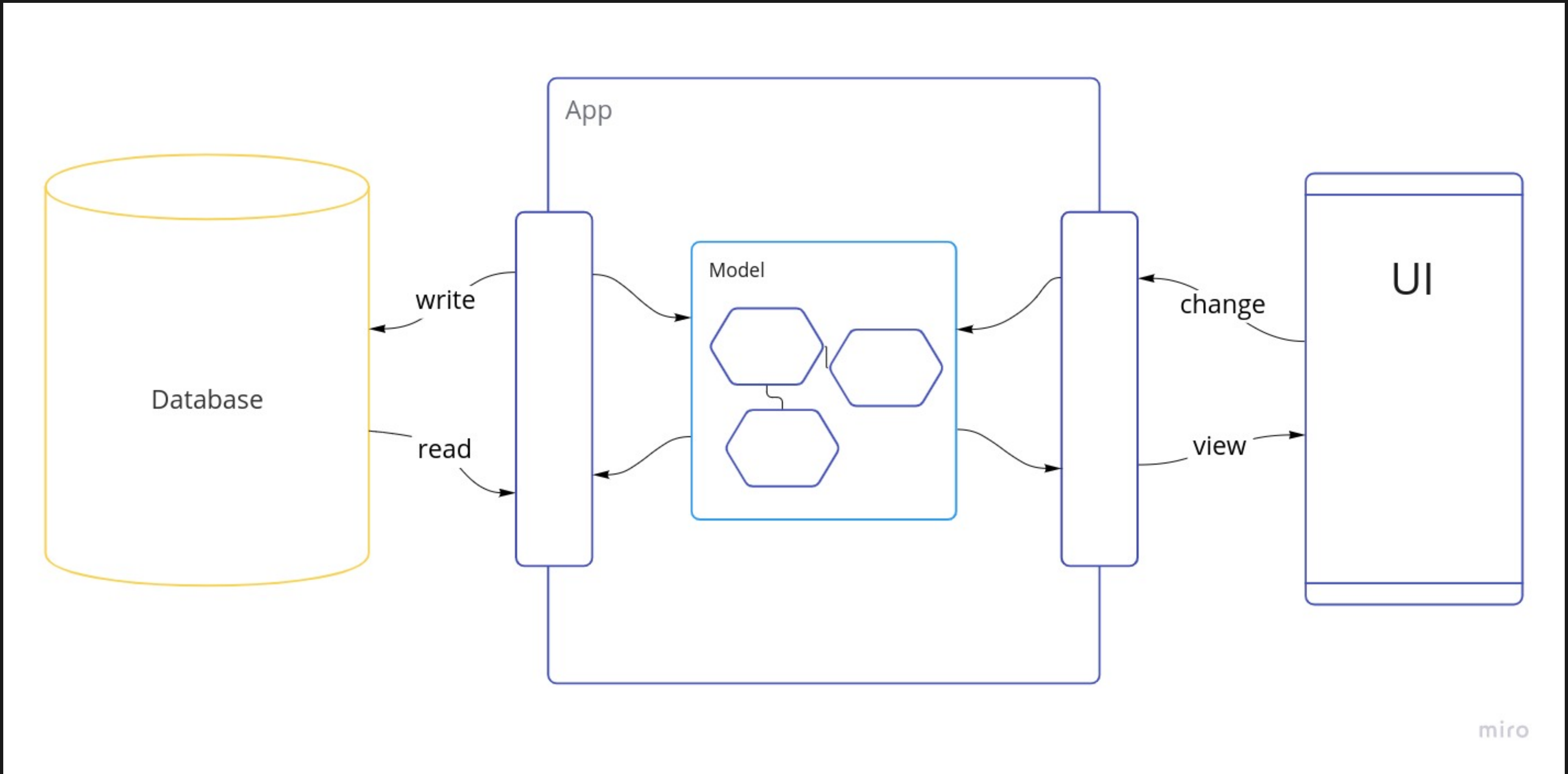
CQR.. WHAT?

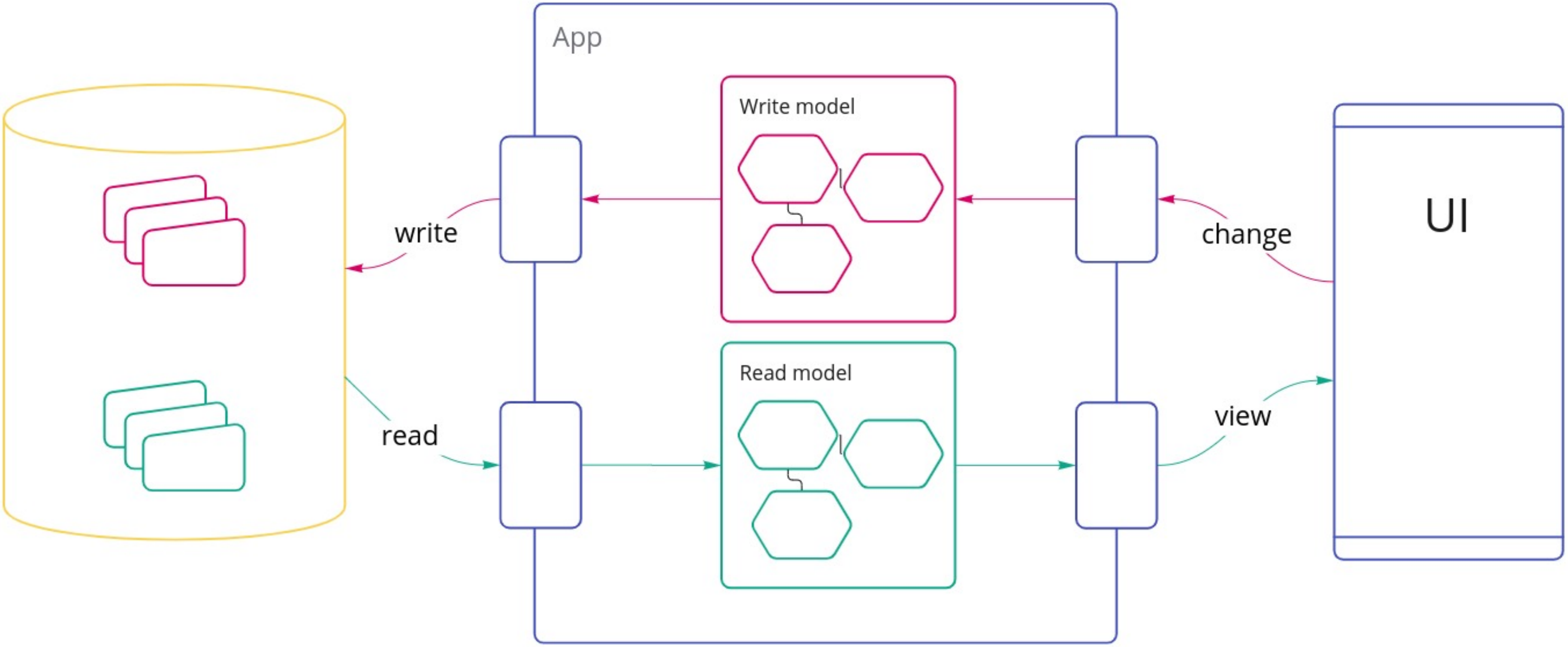
DEFINITION

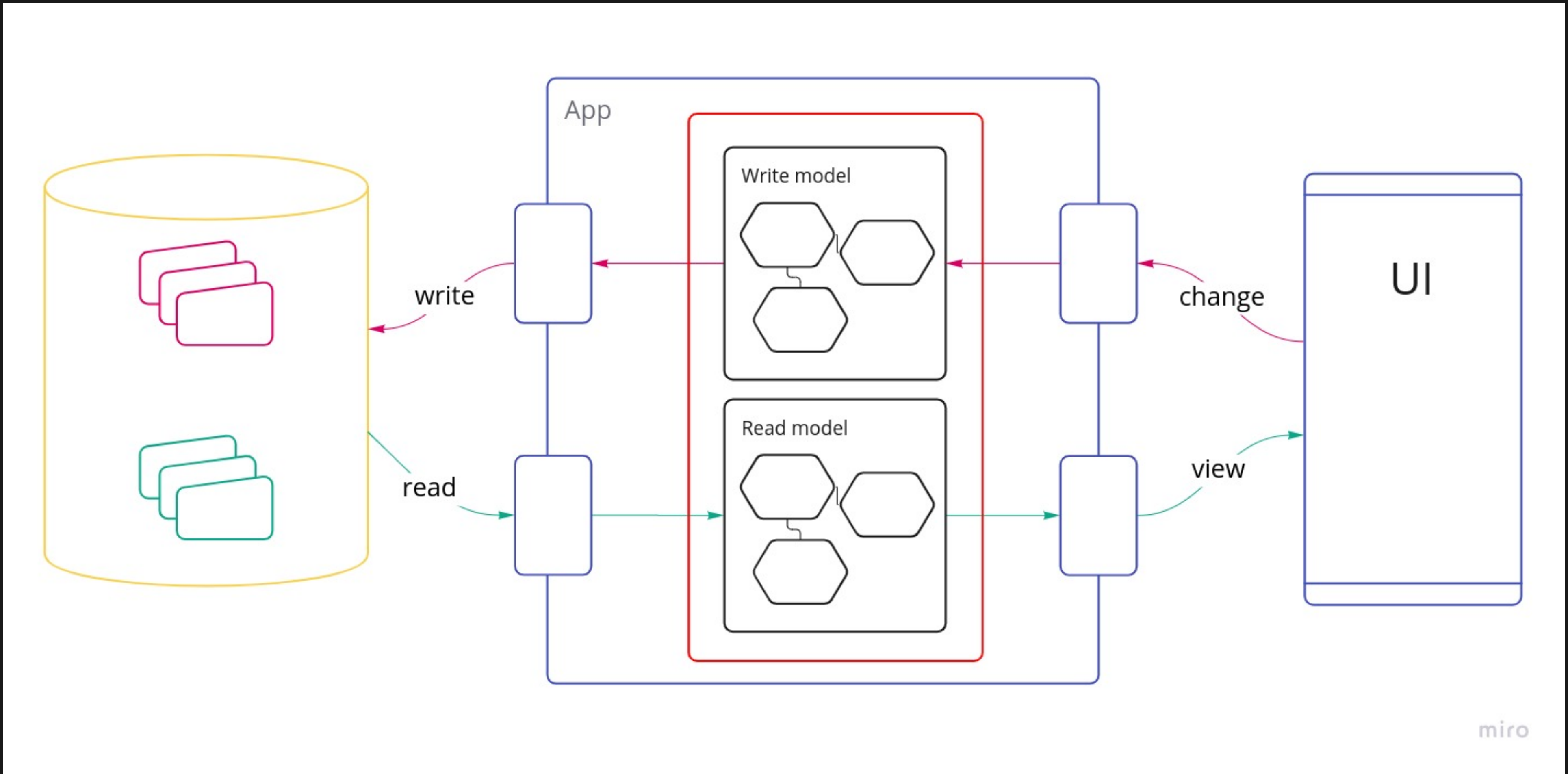
CQRS stands for Command and Query Responsibility Segregation, a pattern that separates read and update operations for a data store. Implementing CQRS in your application can maximize its performance, scalability, and security. The flexibility created by migrating to CQRS allows a system to better evolve over time and prevents update commands from causing merge conflicts at the domain level.

https://docs.microsoft.com/en-us/azure/architecture/patterns/cqrs

Separate models for **reads** and **writes**

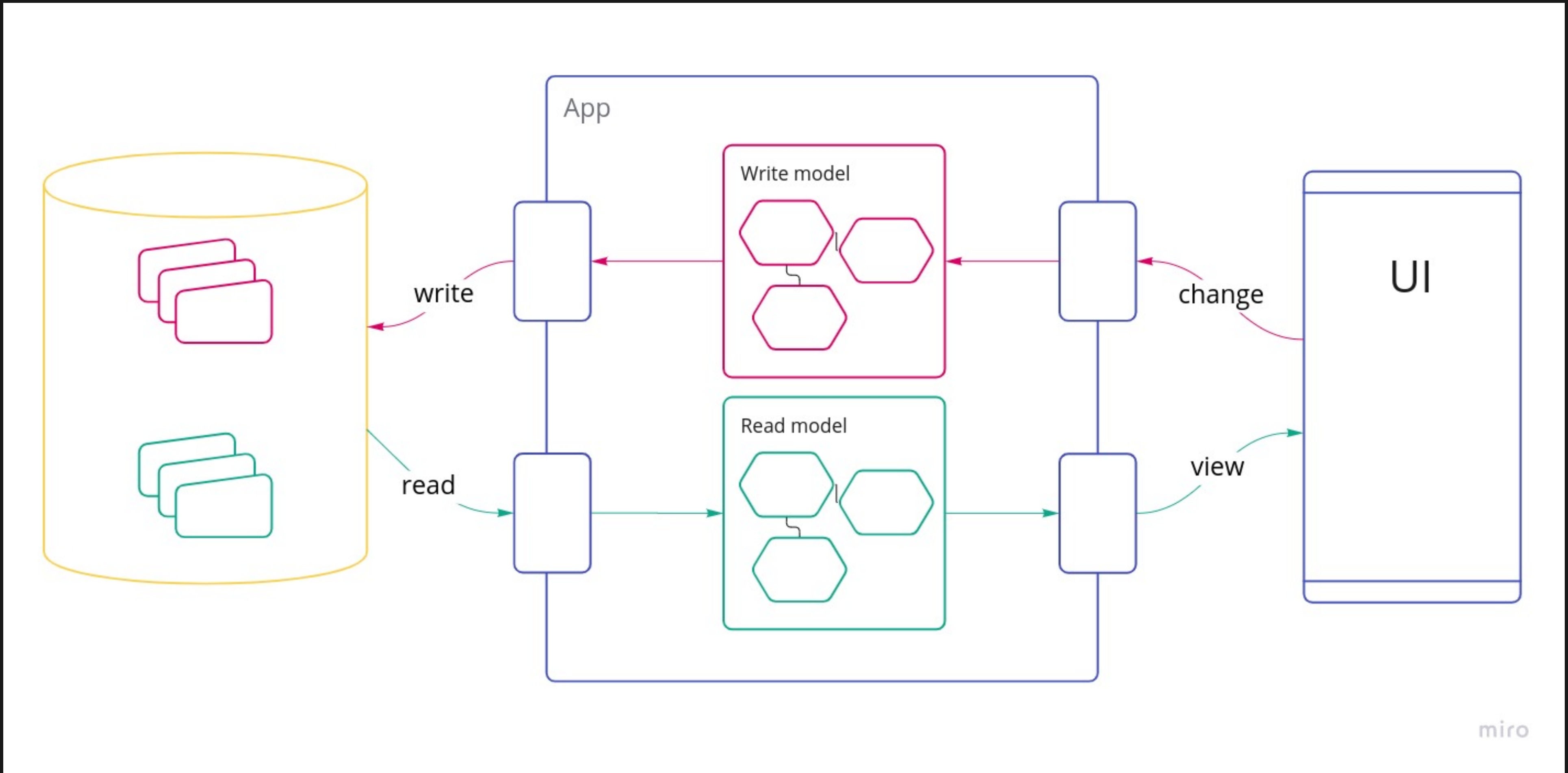

App
Database
write
read
Model
change
view
UI
miro

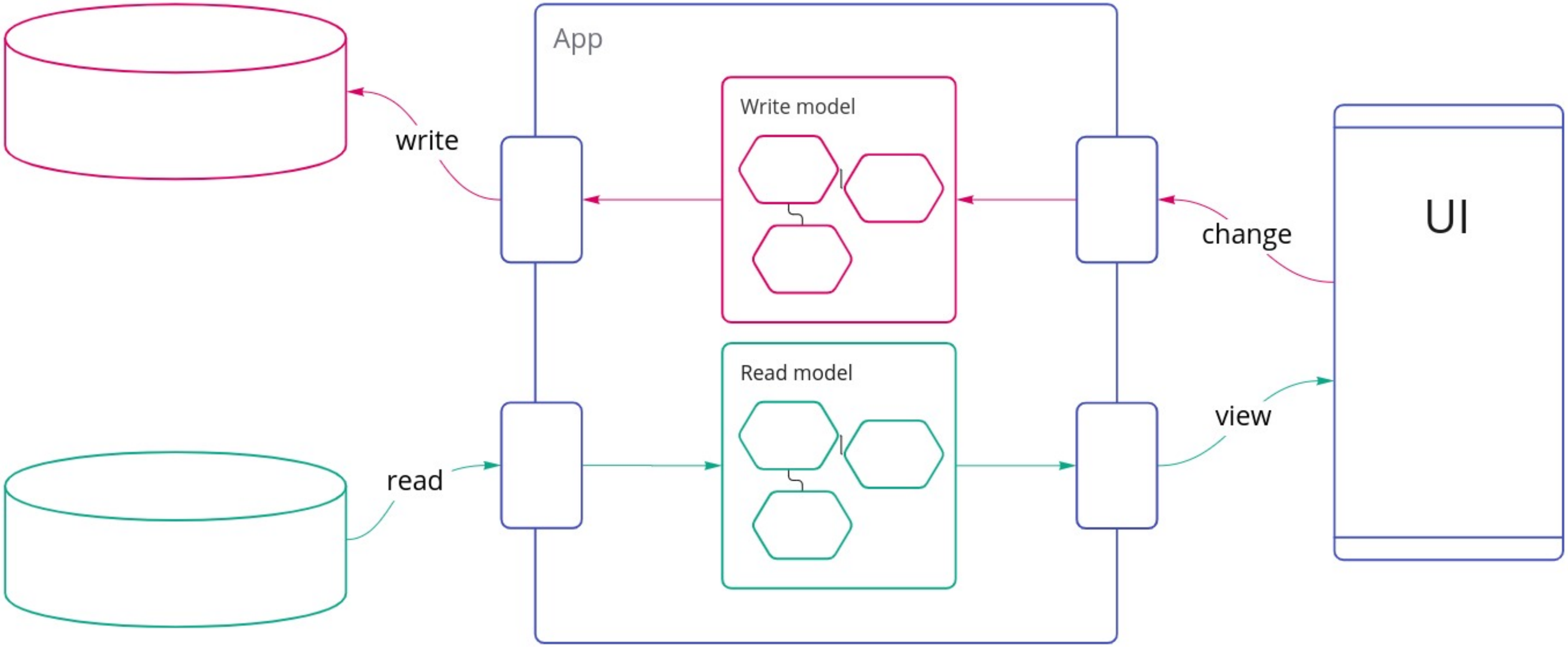

App
Write model
Read model
write
read
change
view
UI

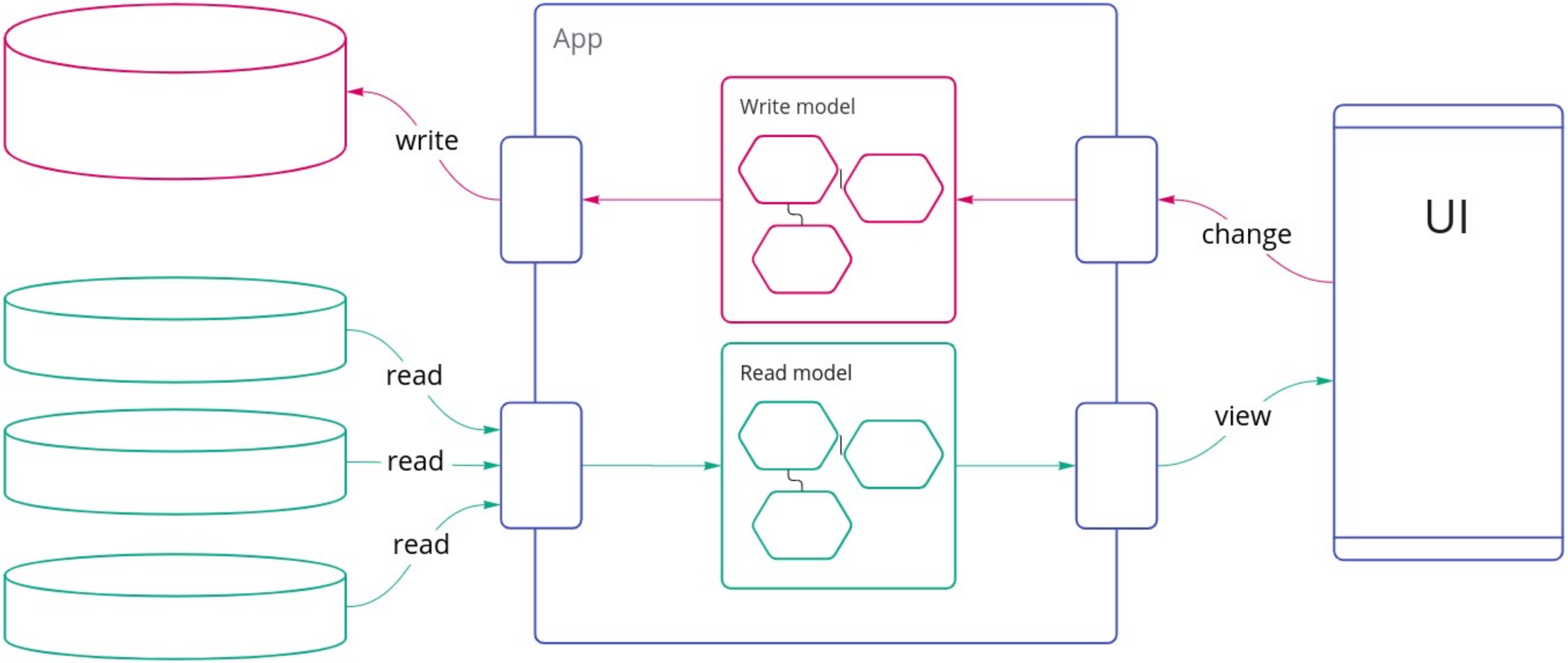

App
Write model
Read model
write
read
change
view
UI
miro

..BUT WHY?

PERFORMANCE!


App
Write model
Read model
write
read
change
view
UI
miro

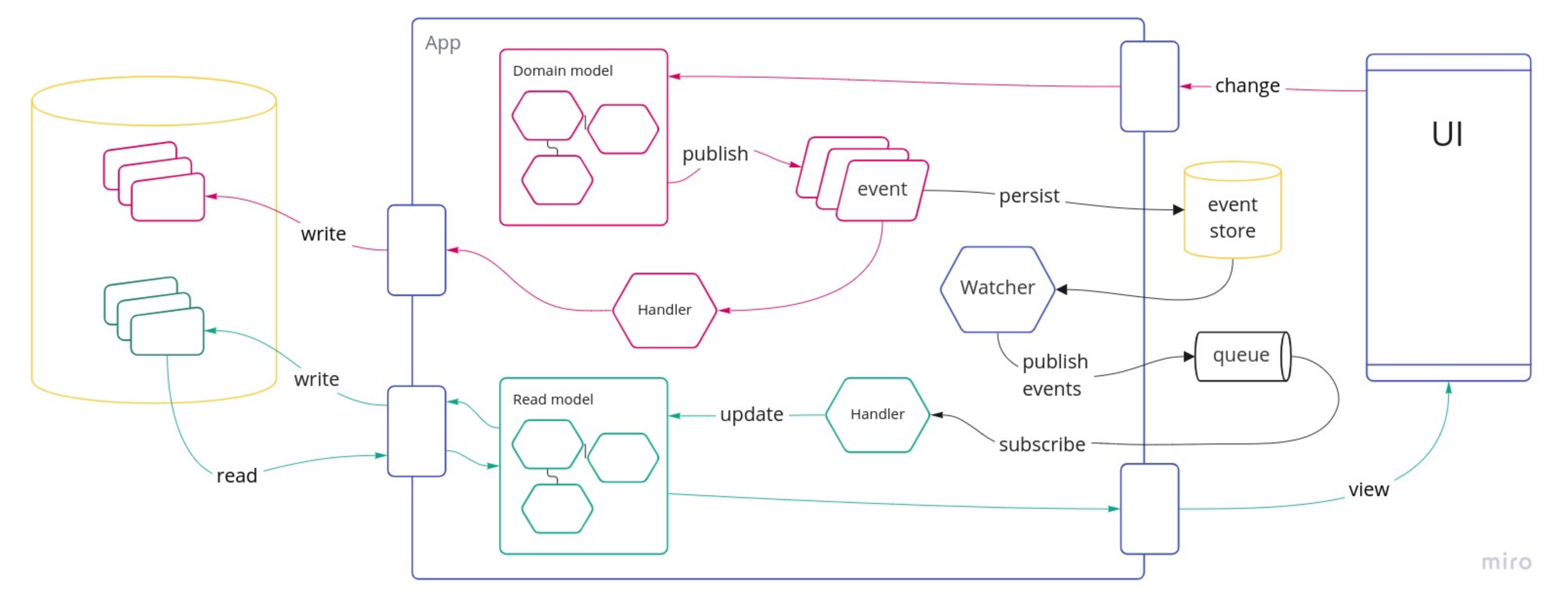

App
Write model
Read model
write
read
change
view
UI


App
Write model
Read model
write
read
read
read
change
view
UI

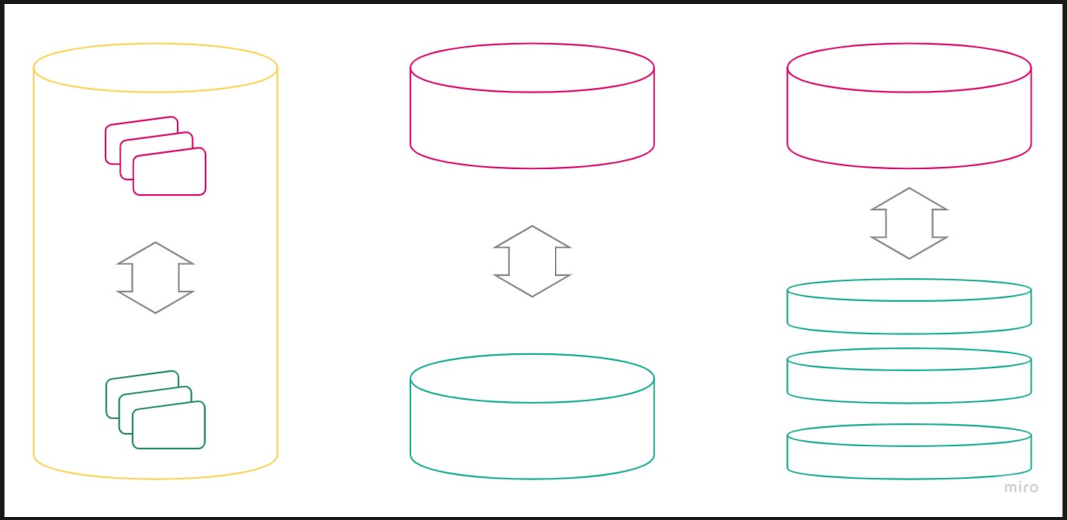
miro

TRADEOFFS

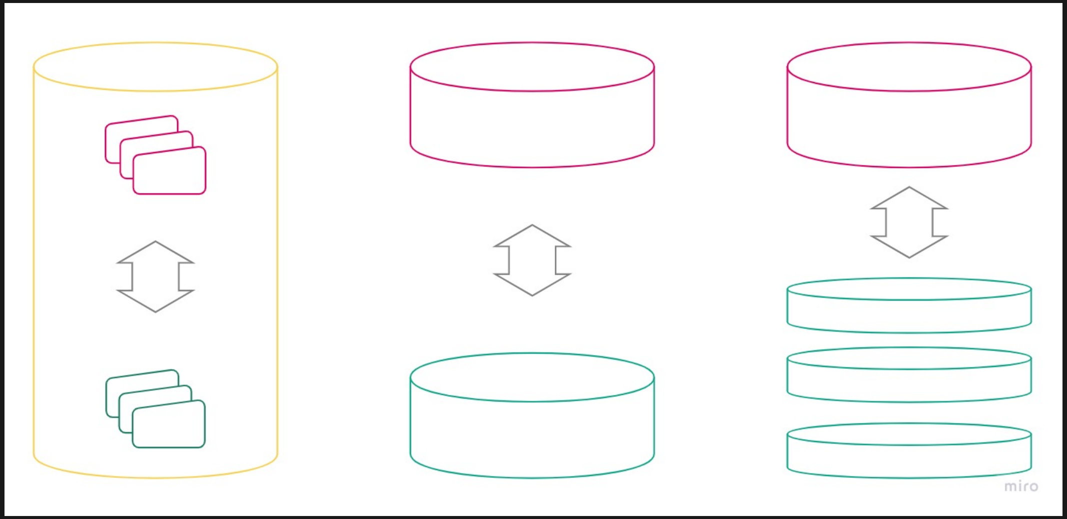
COMPLEXITY 😱


App
Domain model
publish
event
persist
event store
change
UI
write
Handler
Watcher
write
publish events
queue
Read model
update
Handler
subscribe
read
view
miro

CONSISTENCY ?

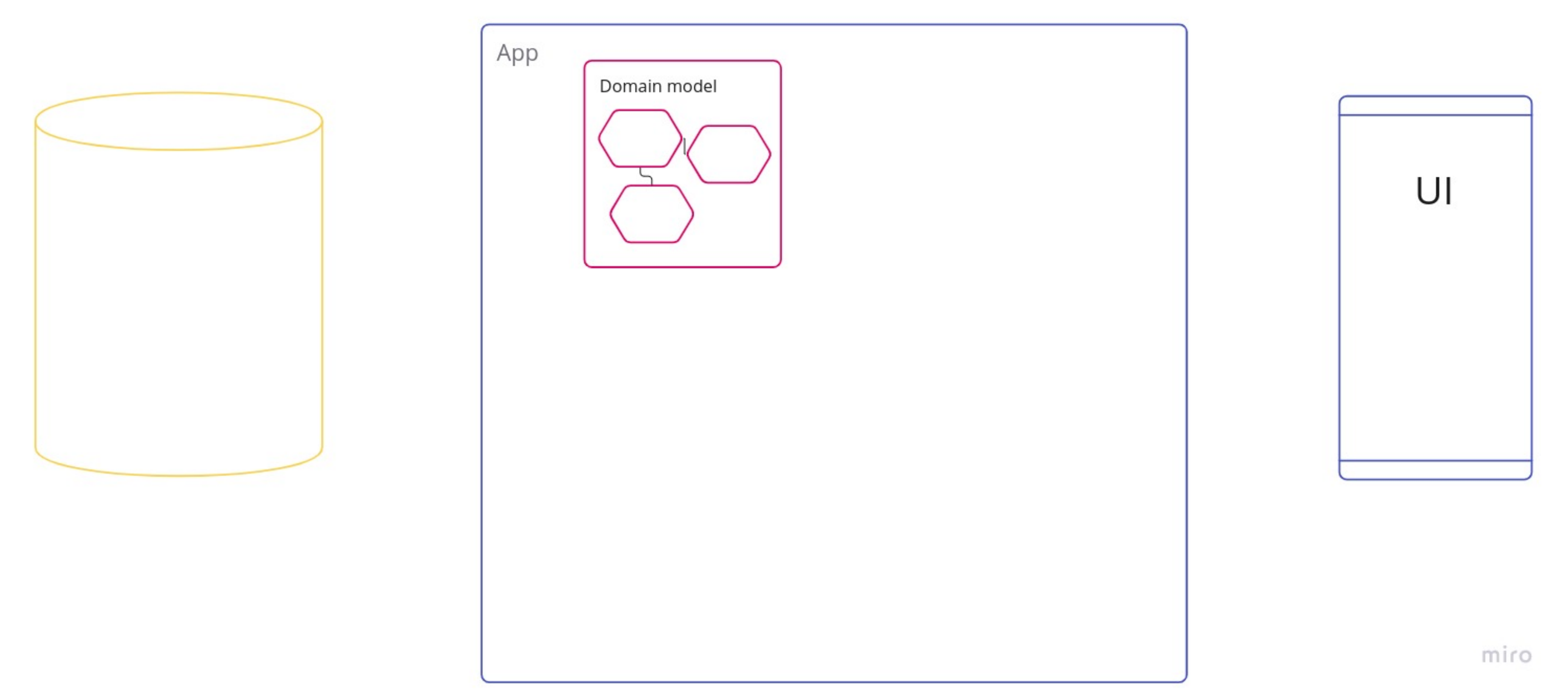


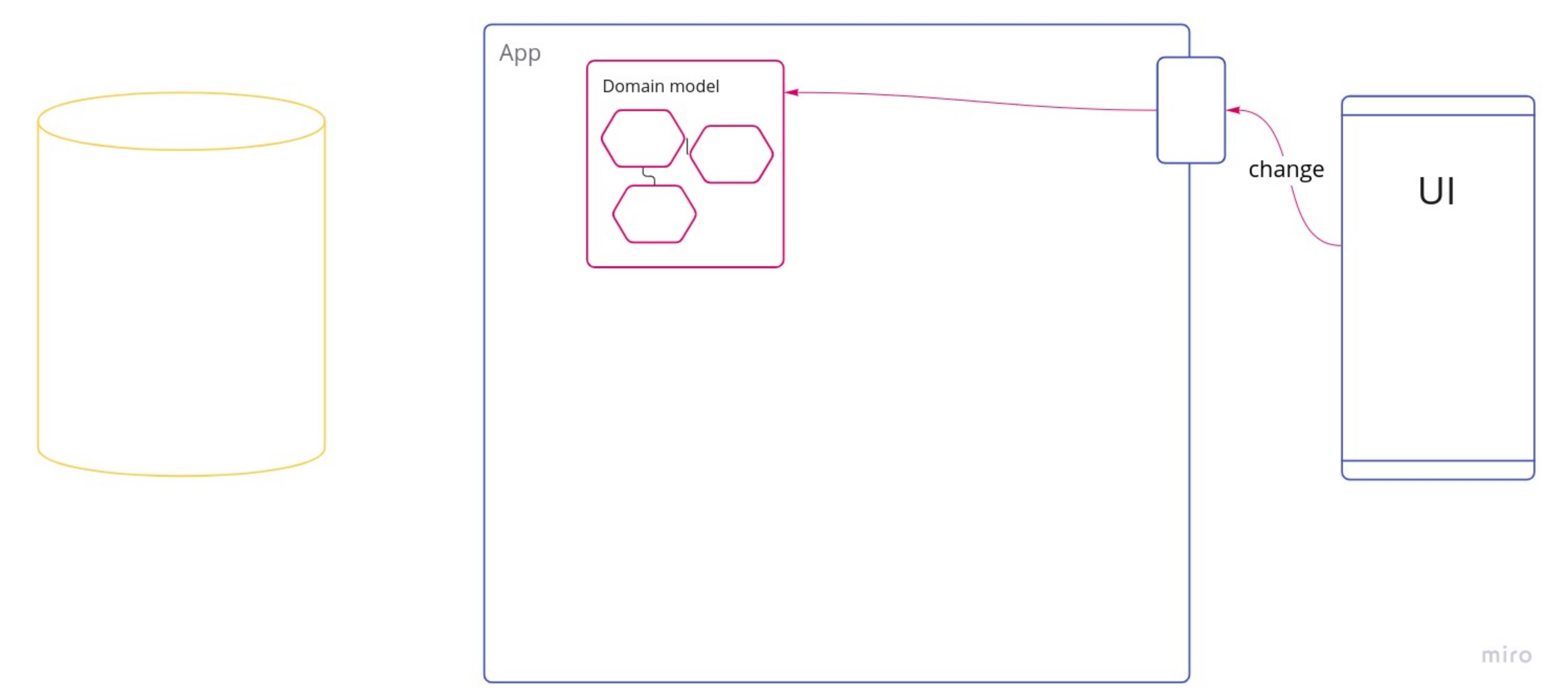
CONSISTENCY ?

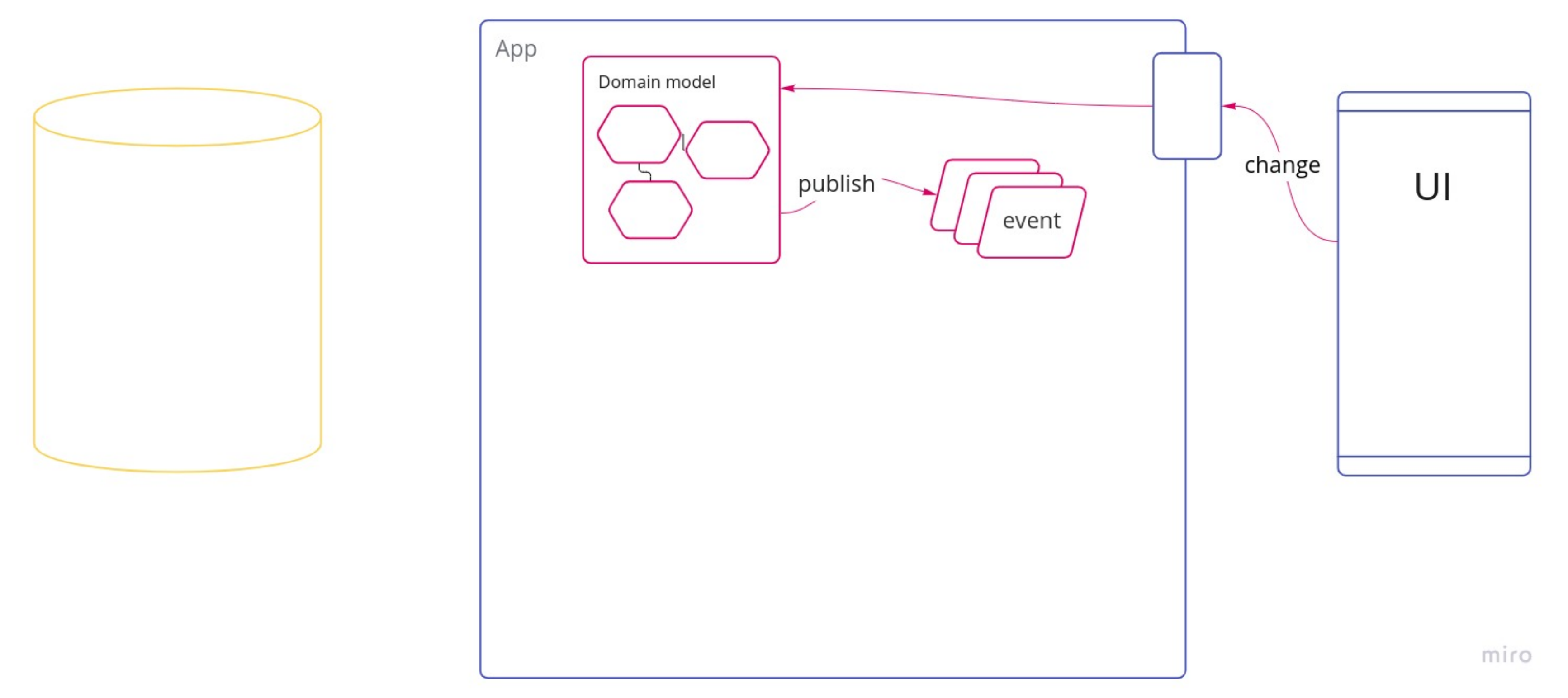


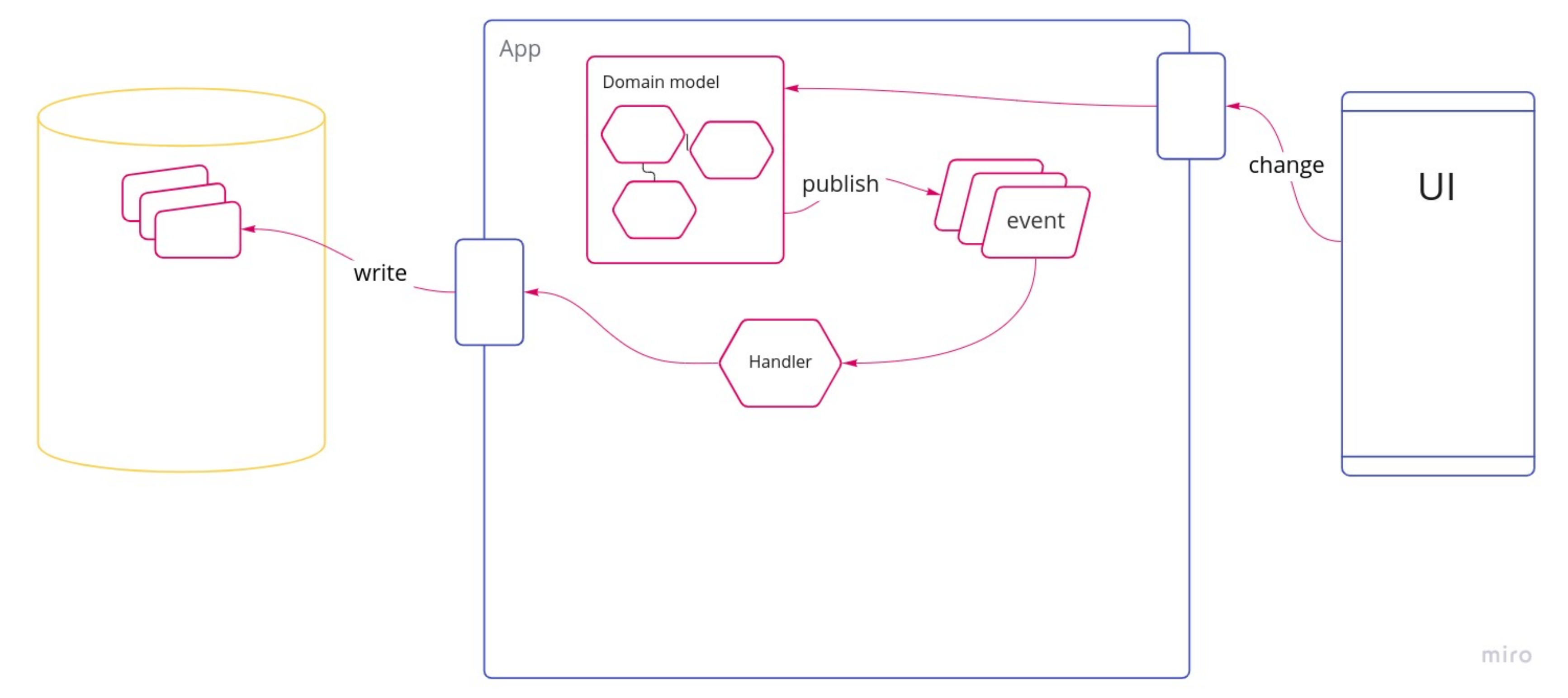
..eventually

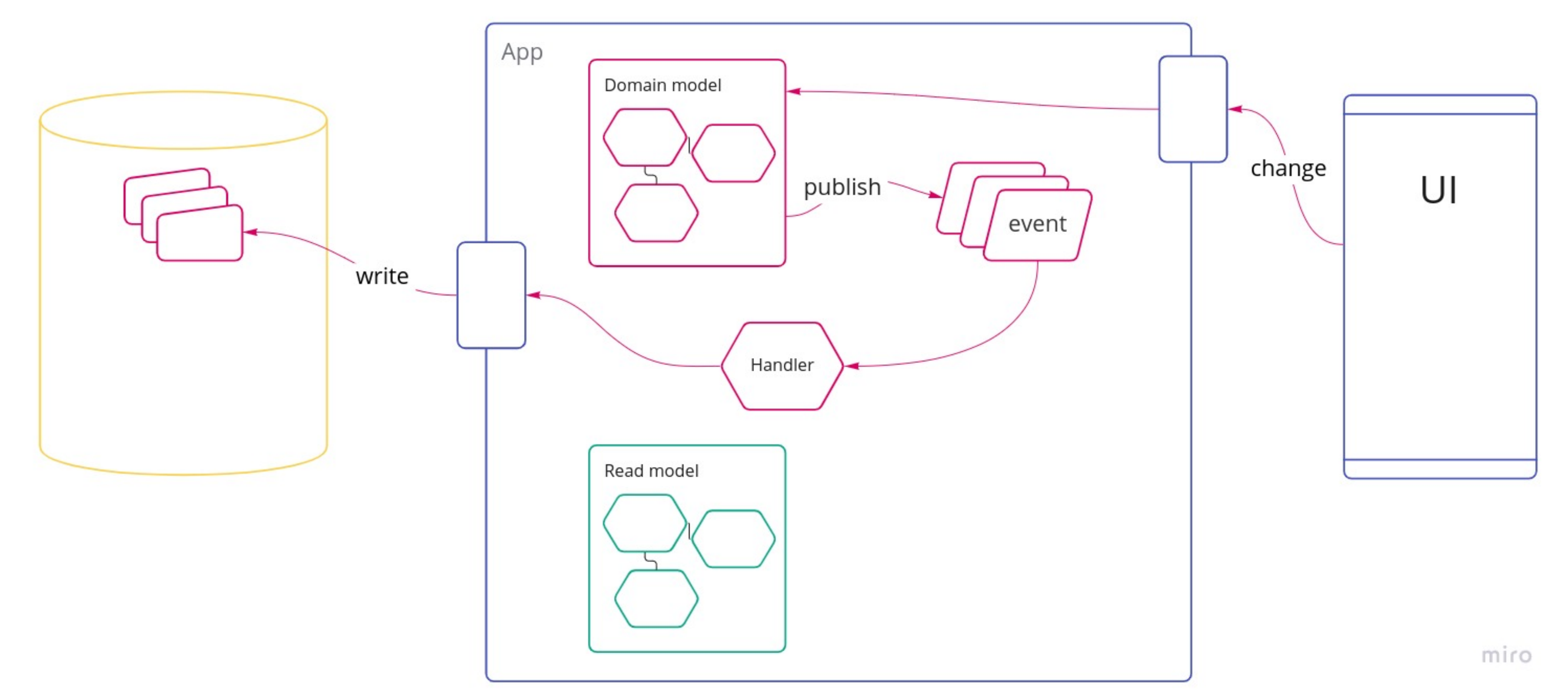
CQRS IN LOGISTICS FASHION

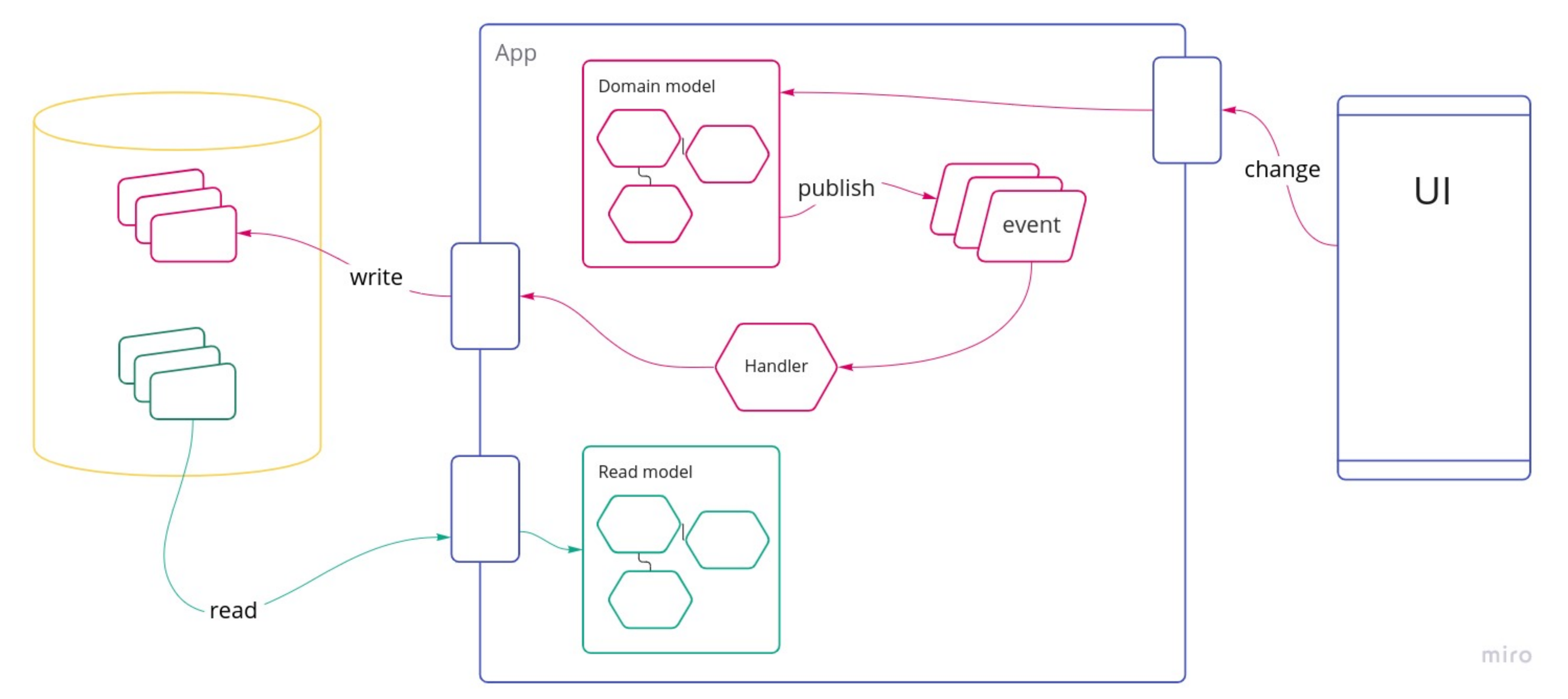

App
Domain model
UI
miro

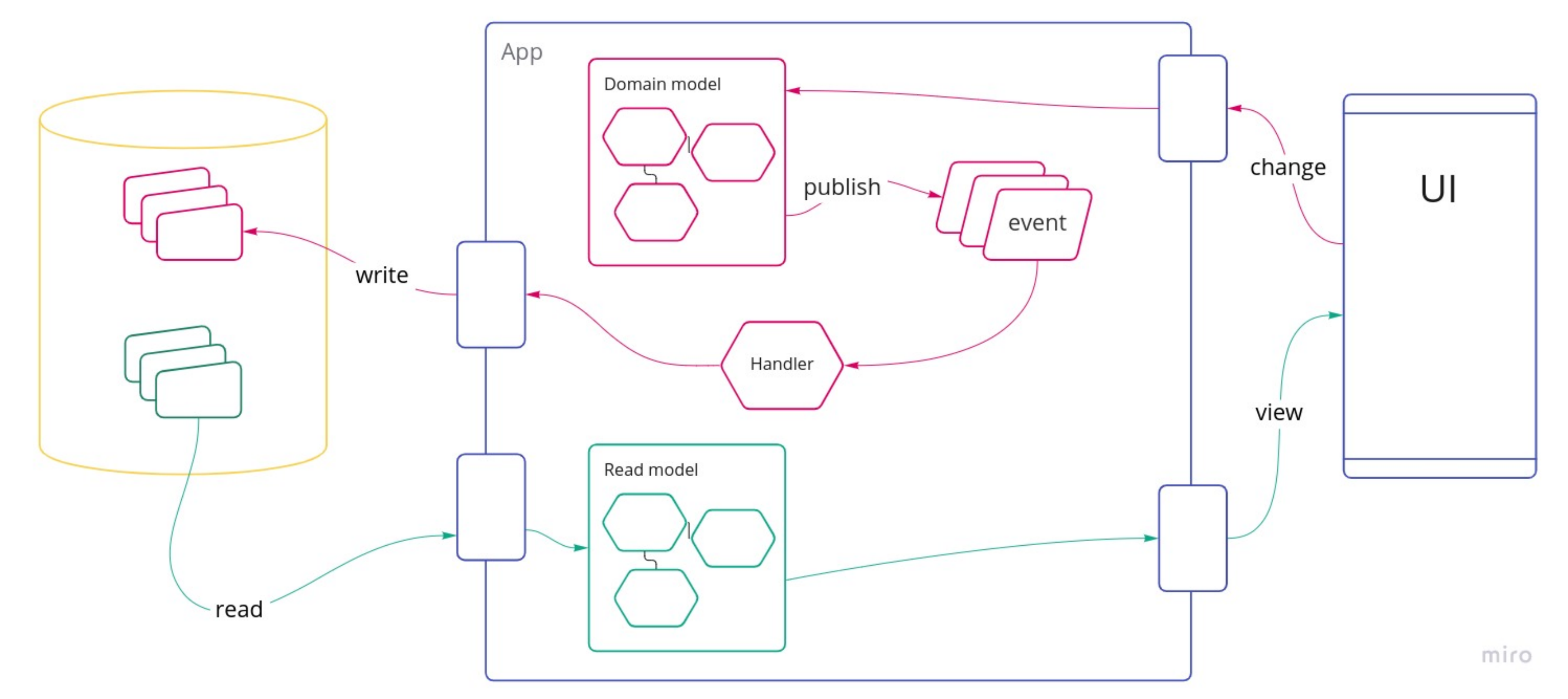

App
Domain model
change
UI
miro

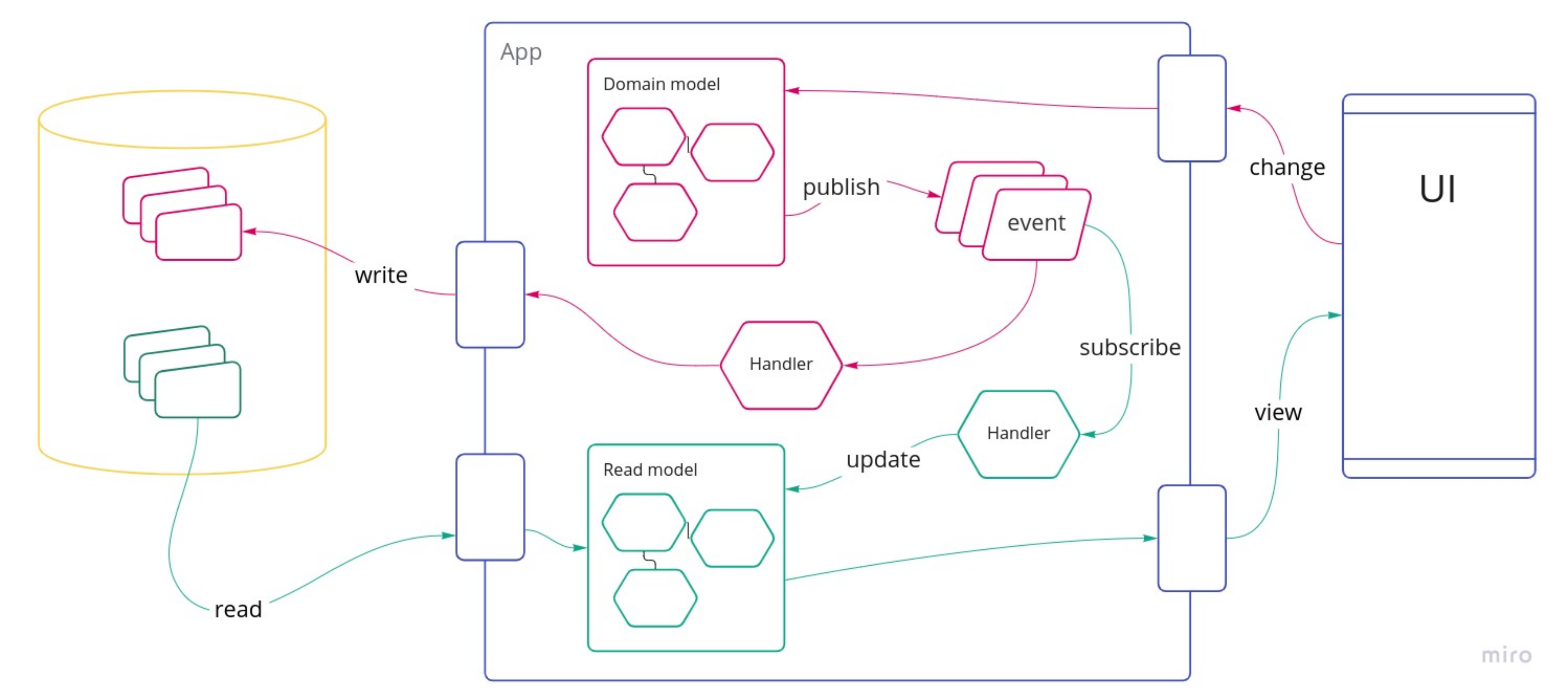

App
Domain model
publish
event
change
UI
miro

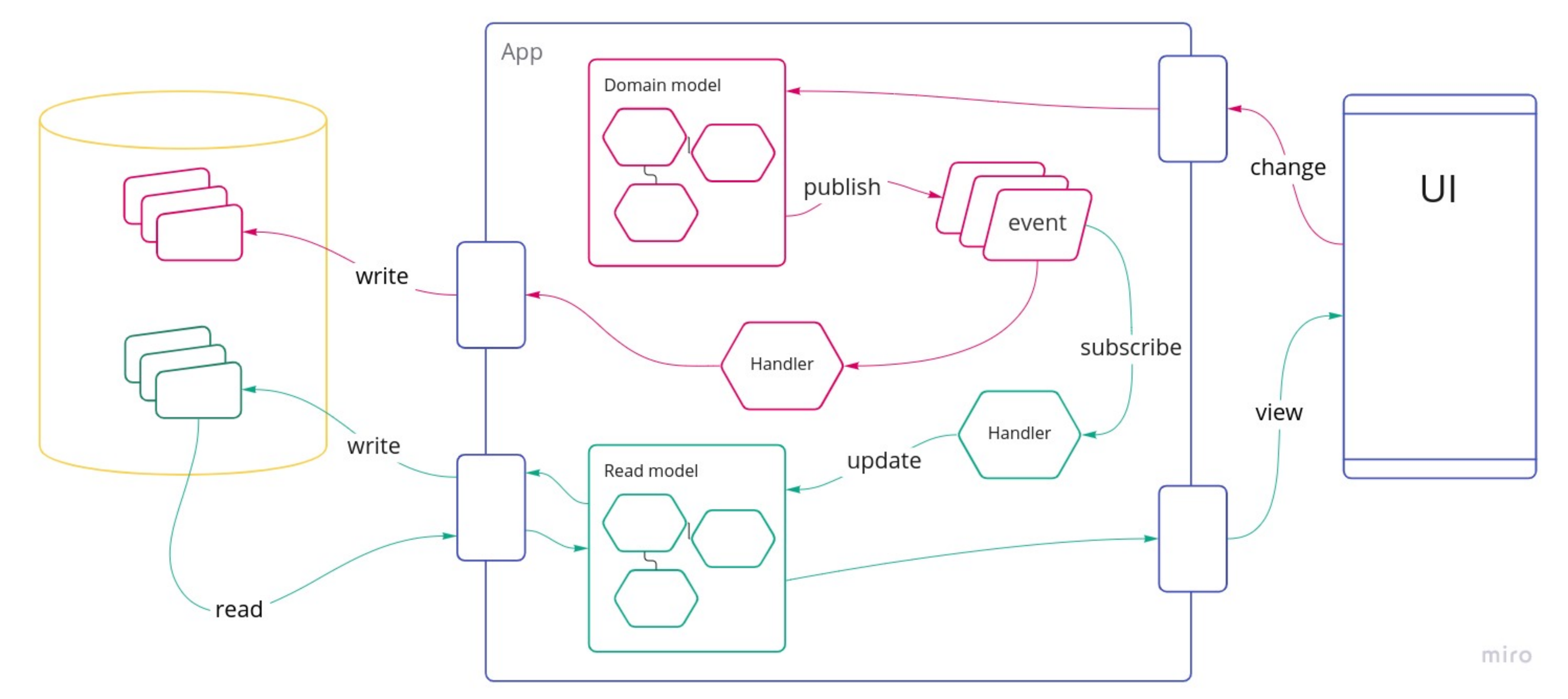

App
Domain model
publish
event
change
UI
write
Handler
miro

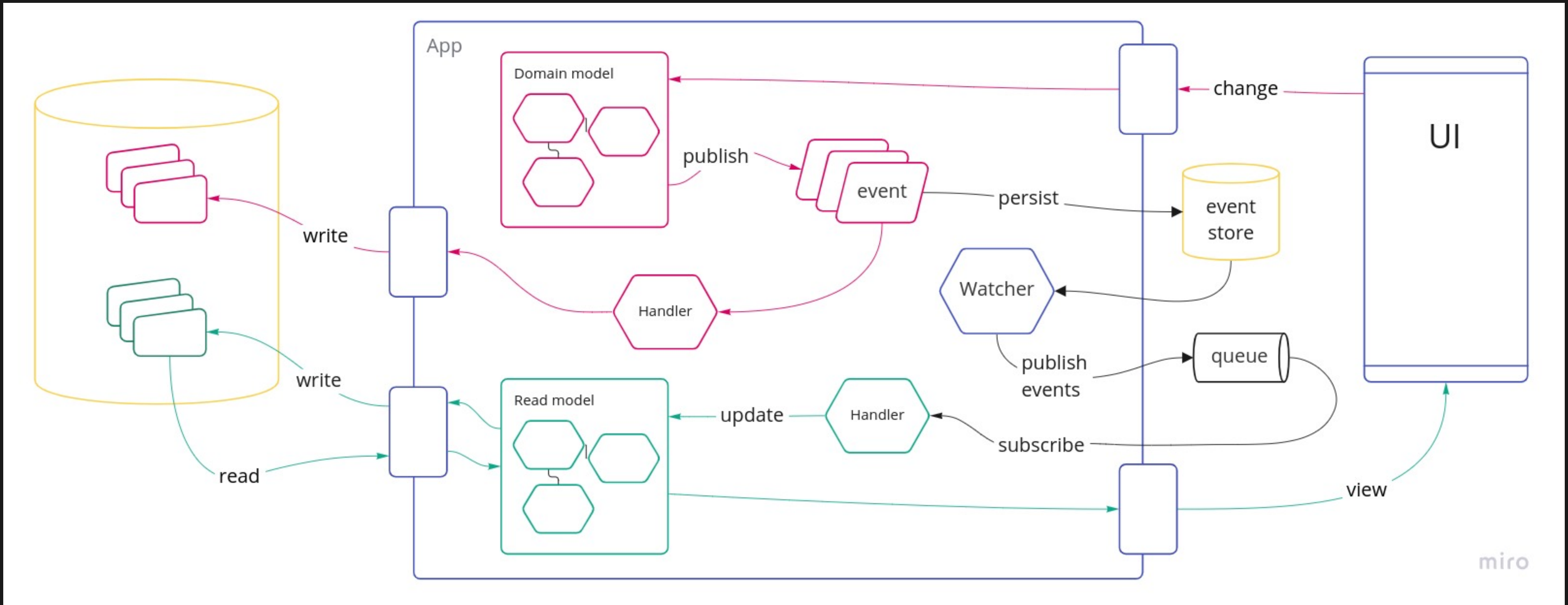

App
Domain model
publish
event
change
UI
write
Handler
Read model
miro


App
Domain model
publish
event
change
UI
write
Handler
Read model
read
miro


App
Domain model
publish
event
Handler
write
read
Read model
change
view
UI
miro


App
Domain model
publish
event
change
UI
write
Handler
subscribe
Handler
view
update
Read model
read
miro


App
Domain model
publish
event
write
Handler
subscribe
Handler
write
update
Read model
read
change
UI
view
miro


App
Domain model
publish
event
persist
event store
change
UI
write
Handler
Watcher
write
publish events
queue
Read model
update
Handler
subscribe
read
view

👍

- domain model not compromised for reads
- scale reads and writes independently

🤔

- complexity 📈
- (eventual) consistency

RESOURCES

- Martin Fowler: CQRS (blog post)
- Architecture Patterns with Python (book)
- Implementing Domain-Driven Design (book)
- Designing data-intensive applications (book)

QUESTIONS?

philipp.albrecht@momox.biz

@lens on Slack