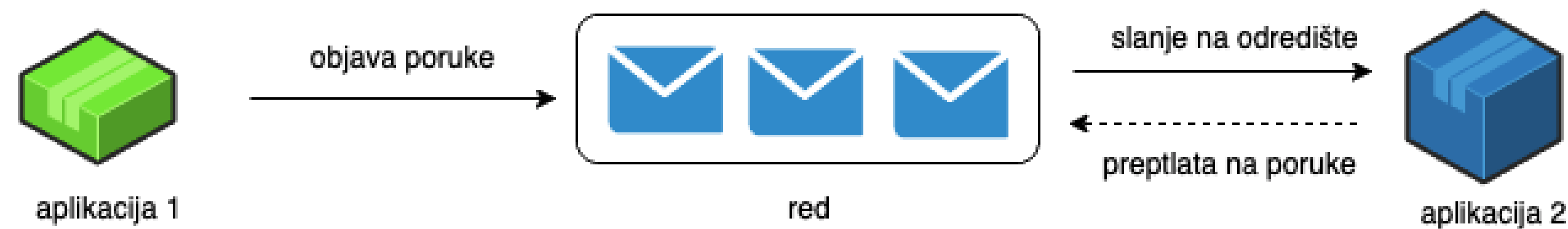
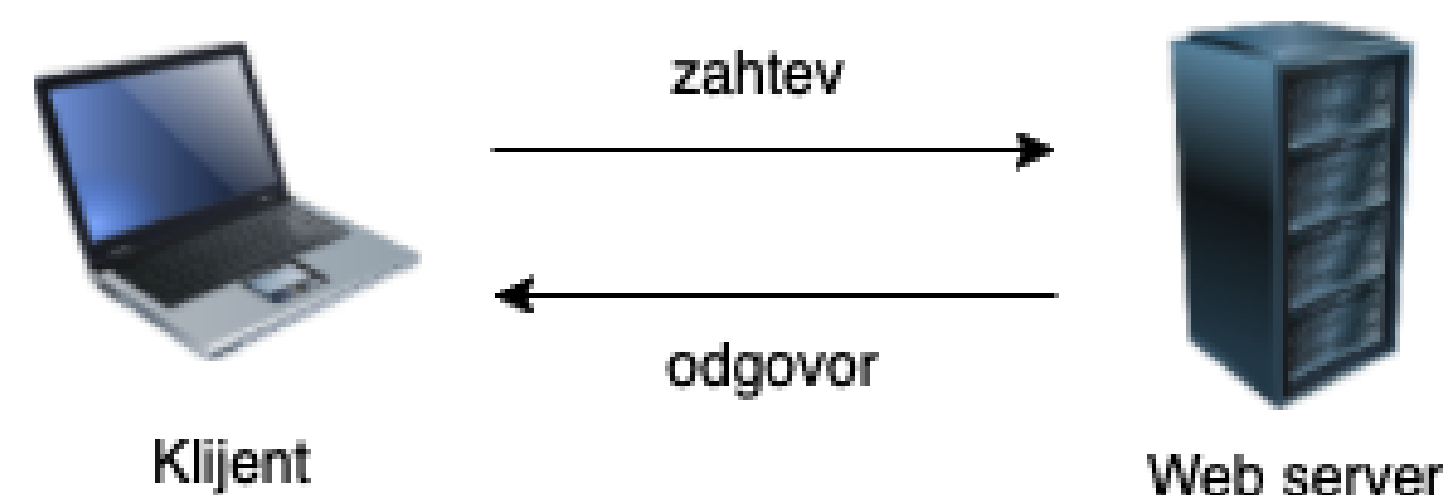
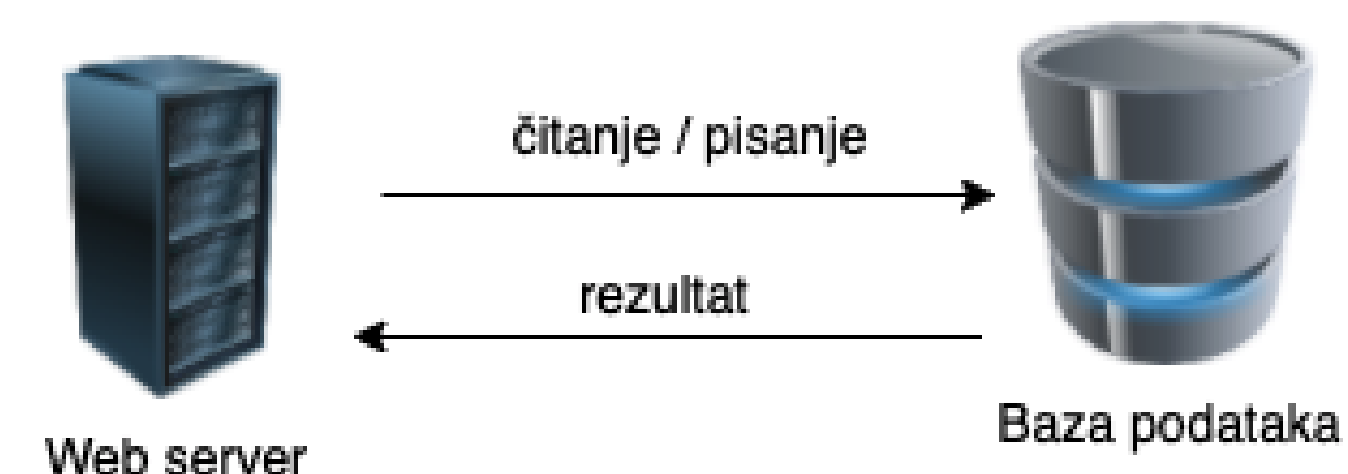


OBJEKTNO- RELACIONO MAPIRANJE

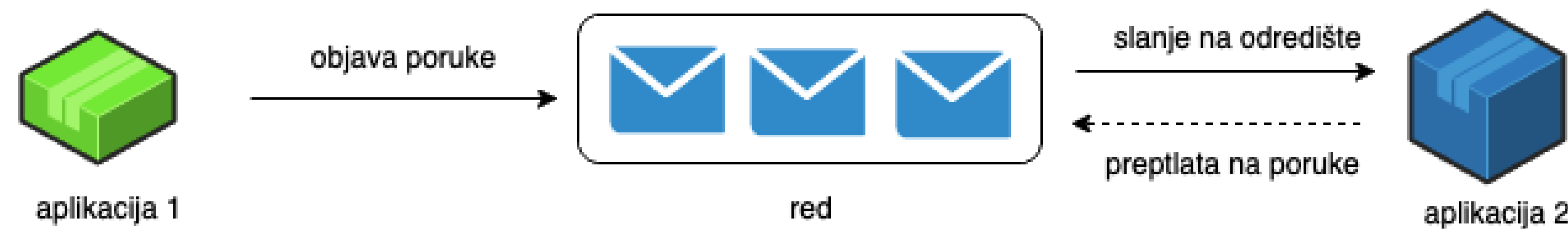
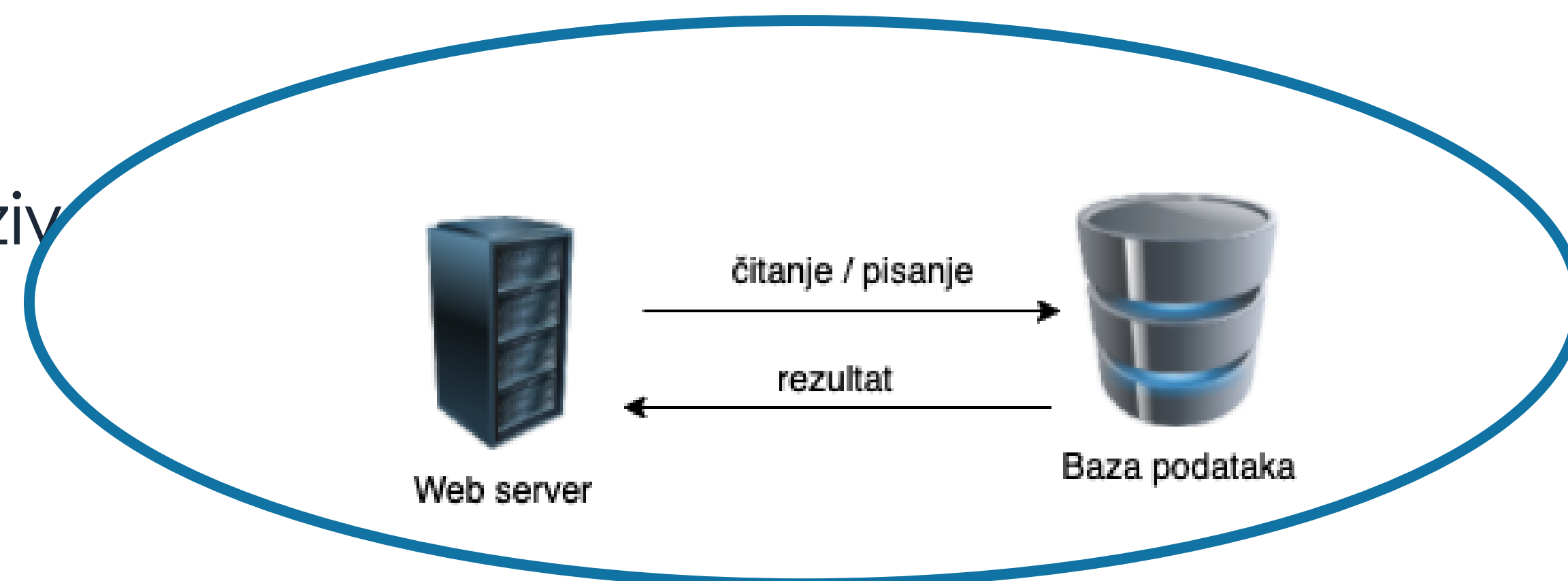
TRI NAJČEŠĆA SCENARIJA

- Serverska aplikacija <-> Baza podataka
- Direktna komunikacija klijent <-> server kroz poziv servisa
- Asinhrona komunikacija razmenom poruka preko reda poruka (message queue)



TRI NAJČEŠĆA SCENARIJA

- Serverska aplikacija <-> Baza podataka
- Direktna komunikacija klijent <-> server kroz poziv servisa
- Asinhrona komunikacija razmenom poruka preko reda poruka (message queue)



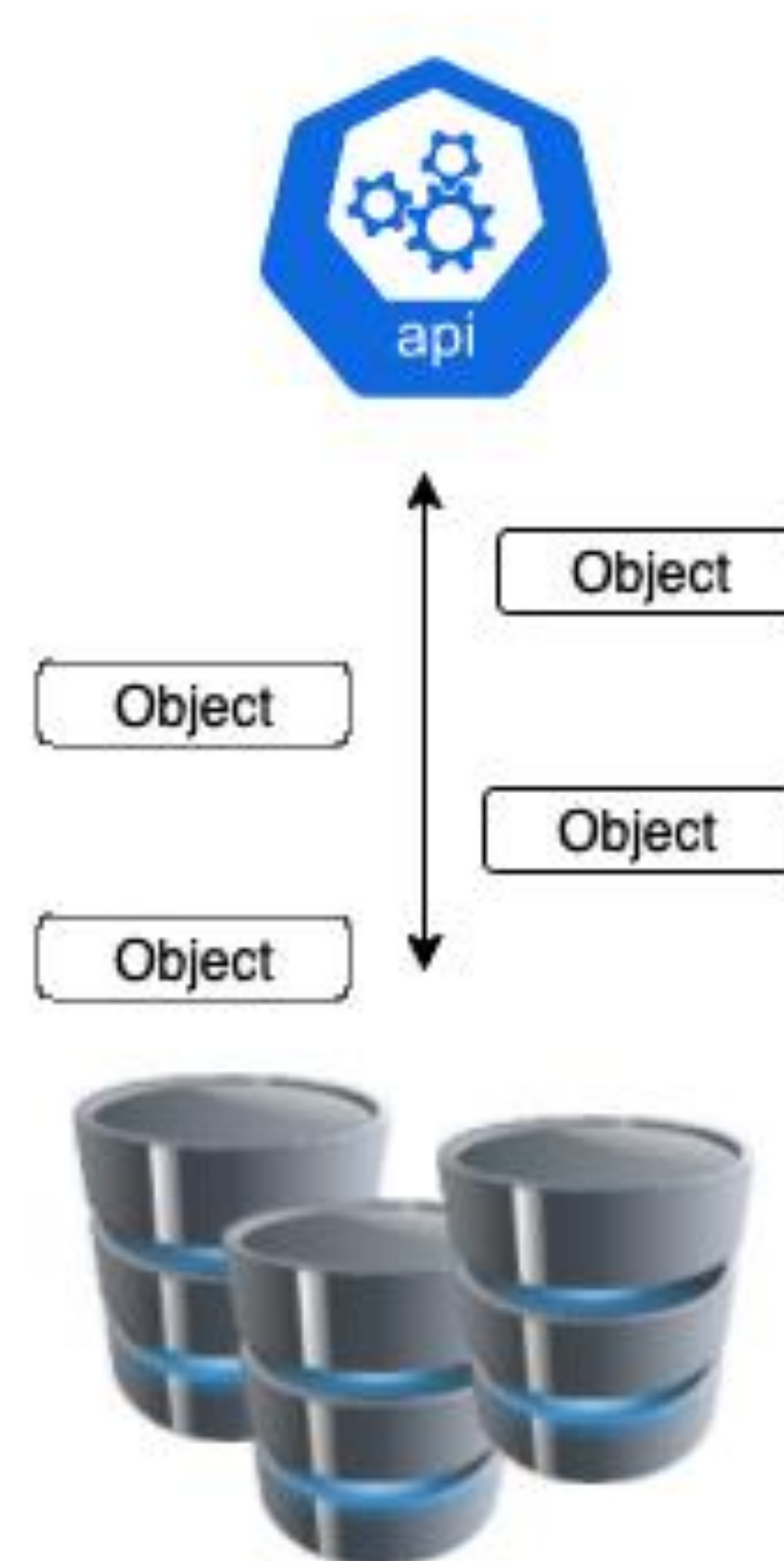


OBJEKTNO-RELACIONO MAPIRANJE (ORM)

4

◆ ŠTA JE CILJ?

- Aplikacije treba trajno da skladište podatke
- Ti podaci mogu biti pohranjeni u DMBS, tekstualne fajlove, itd.
- Ako razvijamo aplikacije u OOP maniru, hoćemo da čuvamo i dobavljamo te podatke kao objekte





◆ ŠTA JE PROBLEM SA RELACIONIM BAZAMA PODATAKA?

- DBMS su (pretežno) orijentisane ka čuvanju podataka u redovima u tabelama (a to ne liči na objekte)
- Asocijacije i kolekcije možemo da simuliramo korišćenjem veza između redova u tabelama
- Postoje konceptualne razlike u paradigmama poznate pod nazivom Object-Relational Impedance Mismatch¹

¹ Object-relational impedance mismatch https://en.wikipedia.org/wiki/Object%E2%80%93relational_impedance_mismatch



◆ KAKO NAM ORM MOŽE POMOĆI U OSTVARIVANJU CILJA?

- Radi konverziju objekata u format koji je u skladu sa API-em za slanje na čuvanje DBMS-u
- Pravi upite ka DBMS-u i kreira od rezultata objekte
- Čuva i rekreira veze između objekata



◆ PREDNOSTI

- Omogućena je konverzija iz bogatog skupa tipova podataka podržanih u OOP jezicima u skromniji skup tipova podataka podržanih od strane DBMS
- Pisanje ispravnih i optimizovanih upita je jednostavnije i brže
- Kod je lakši za ažuriranje, održavanje i ponovno korišćenje jer se sve vreme u aplikaciji koriste objekti
- Konvertuju upite u preporučeni format za specifični DBMS smanjujući prostor za grešku ili napad (SQL Injection)
- Nude koncept "apstrakcije baze" kojim se omogućava laka zamena DBMS-a koji aplikacija koristi



◆ MANE

- Manje kontrole nad kodom jer “neko drugi radi posao za nas”
- Upiti u koje se konvertuje kod mogu biti spori ako se ne poštuju preporuke i principi koje ORM nameće
- Kompleksniji upiti se moraju pisati u nativnom SQL
- Mogu se smatrati primerom apstrakcija koje imaju manjkavosti (*leaky abstractions*¹)

¹ Leaky abstractions <https://www.joelonsoftware.com/2002/11/11/the-law-of-leaky-abstractions/>



OBJEKTNO-RELACIONO MAPIRANJE

9

◆ KLASA

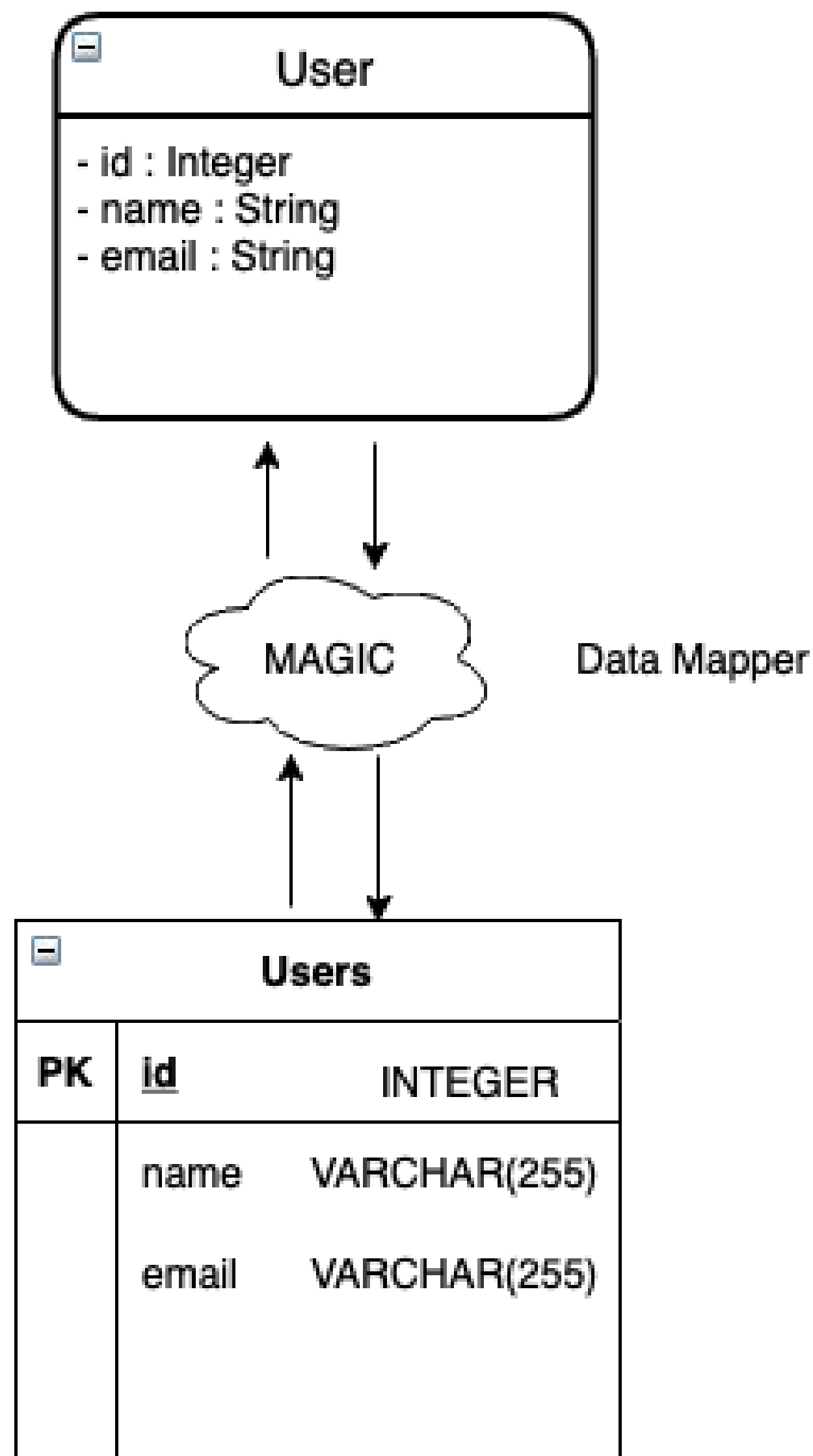
- Trebala bi da ima jedinstveni identifikator kao atribut

◆ DATA MAPPER

- Konvertuje objekat u red tabele, konvertuje tipove podataka, instancira objekte

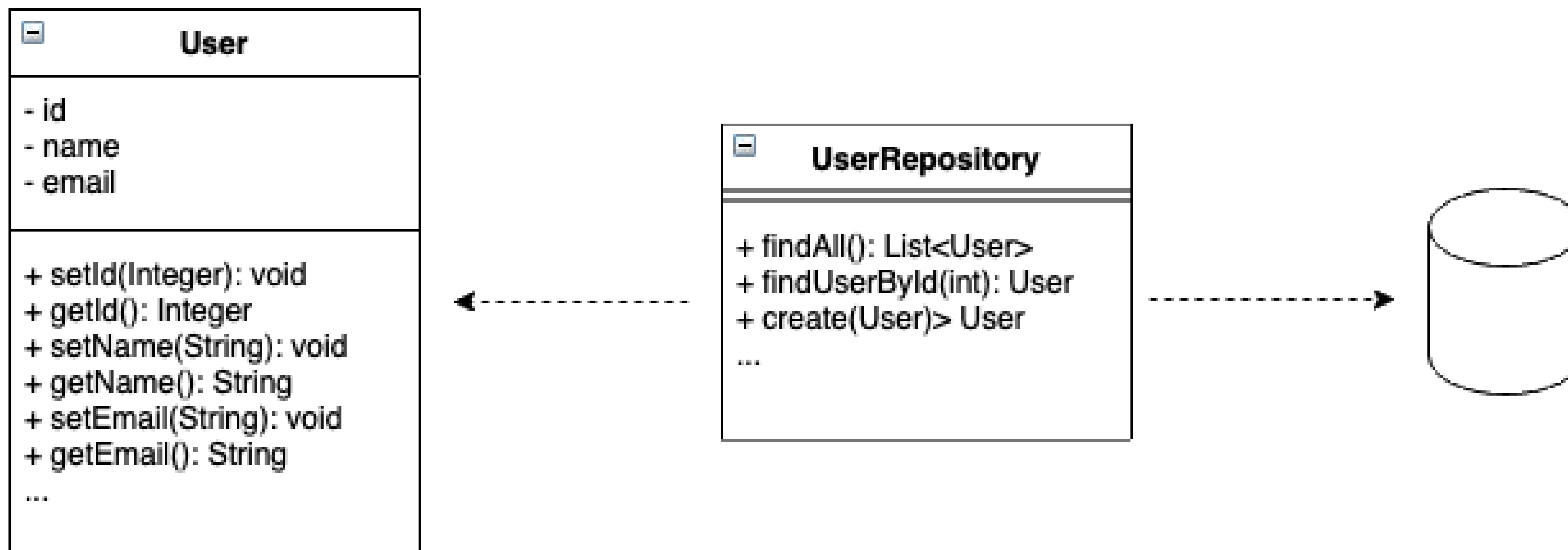
◆ TABELA

- Ima jedinstveni identifikator koji je obično primarni ključ



◆ DATA MAPPER¹ JE ŠABLON KOJI:

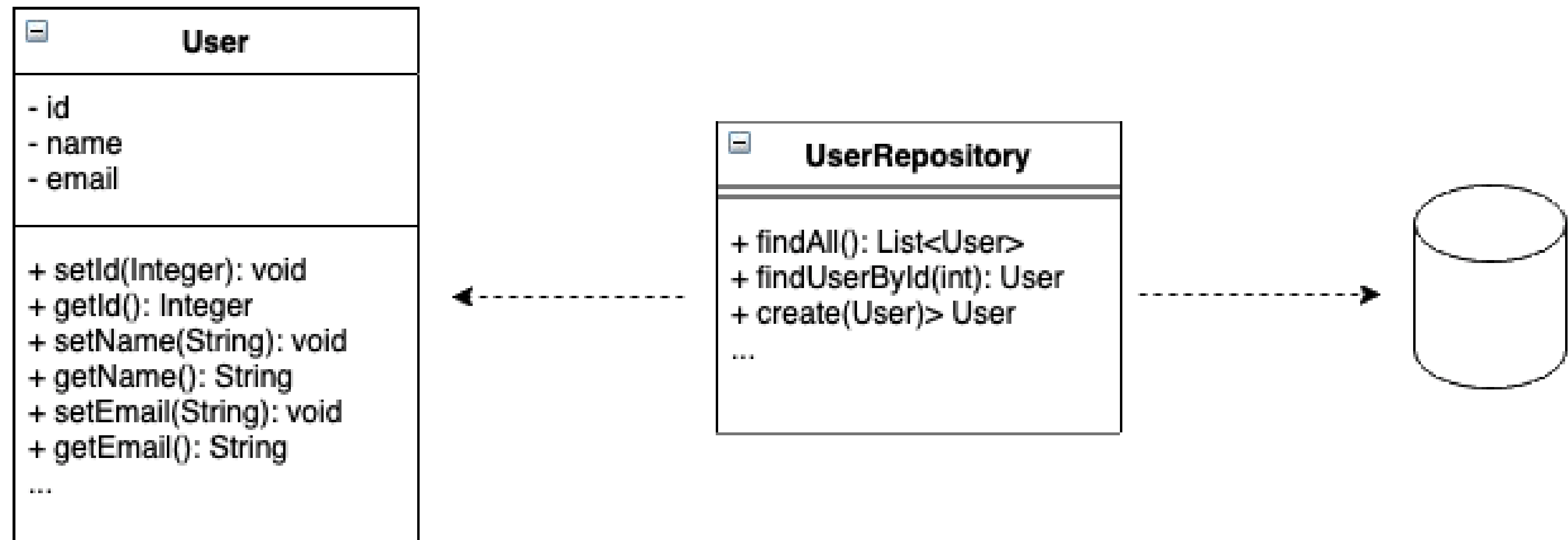
- Postoji u posebnom sloju i razmenjuje podatke između aplikacije i baze držeći ih nezavisne jedno od drugog i od samog mapera



¹ Data Source Architectural Patterns <https://martinfowler.com/eaCatalog/>

◆ KOJE OPERACIJE TREBA DA PODRŽI

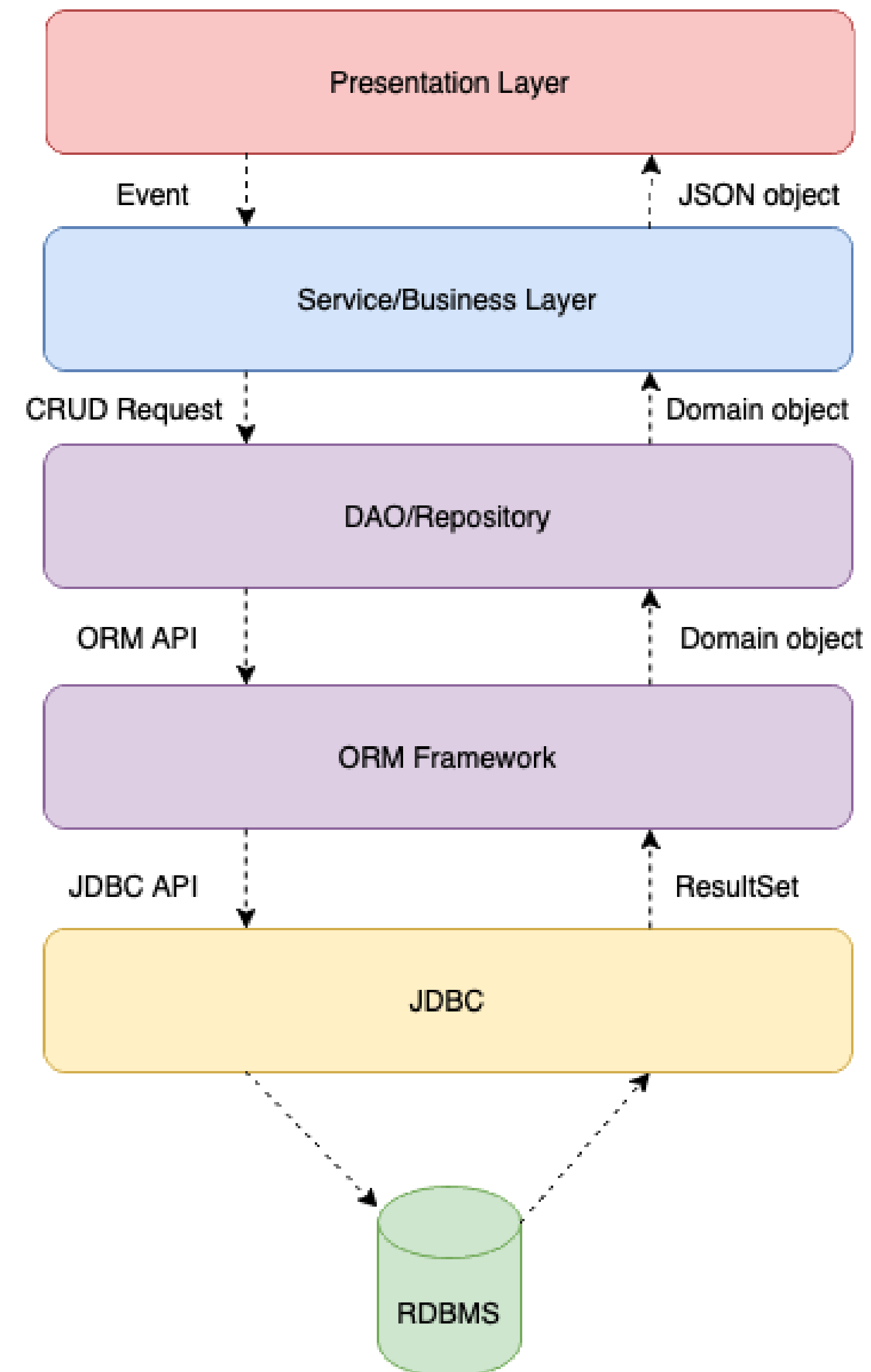
- Create – čuvanje novog objekta u bazu
- Retrieve – dobavljanje objekta iz baze
- Update – ažuriranje podataka o objektu koji se već nalazi u bazi
- Delete – brisanje objekta iz baze
- ...





ORM I ARHITEKTURA APLIKACIJE

12





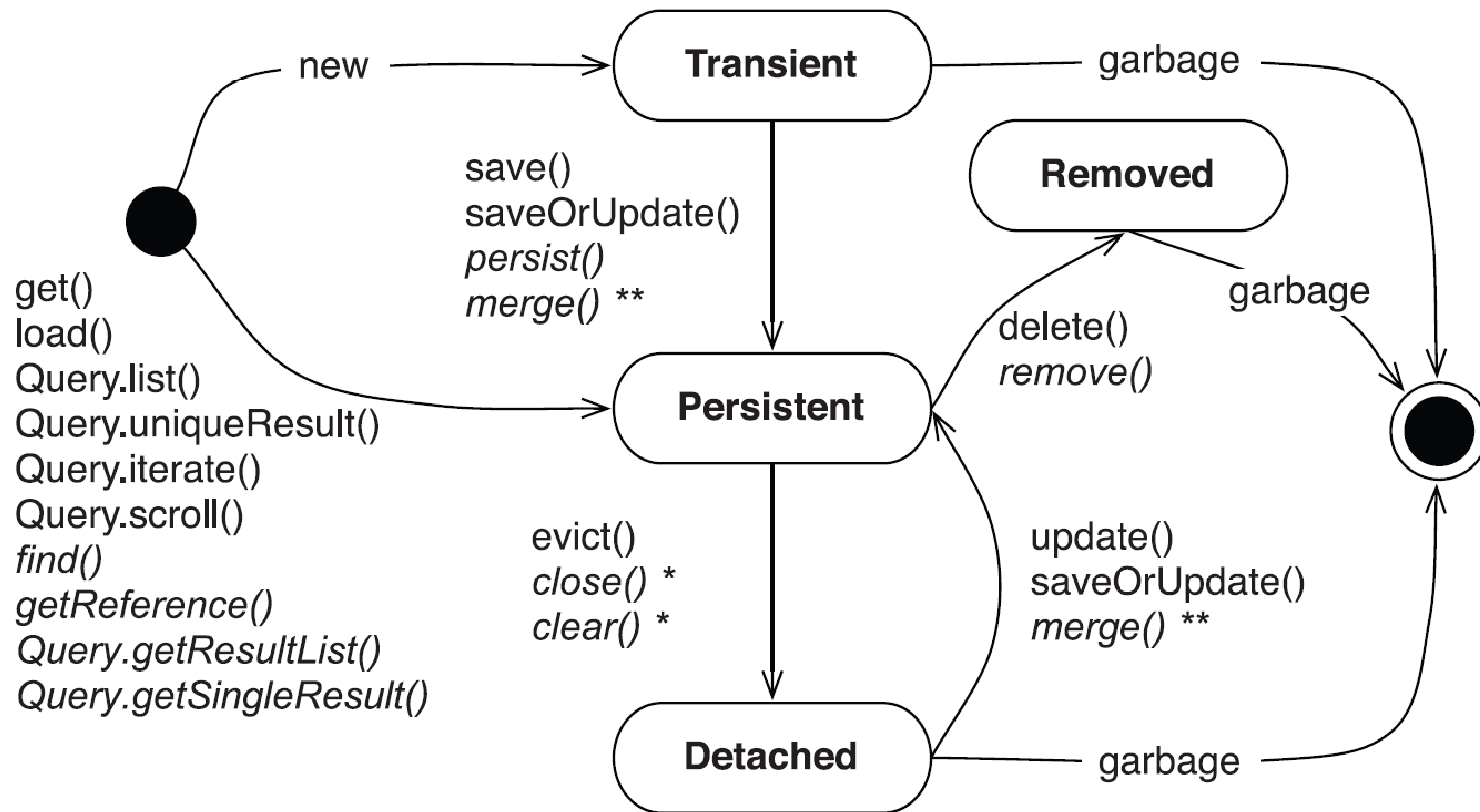
◆ ŠTA JE JPA?

- Java standardna specifikacija za upravljanje relacionim podacima u enterprise Java aplikacijama (praktično specifikacija za ORM)
- Ranije poznatija kao Java Persistence API
- Entitet je objekat koji se čuva u bazi
- Koriste se anotacije da se označe različite komponente koda kako bi se odradilo mapiranje na DMBS
- Postoje različite implementacije JPA
 - Hibernate (najpopularniji, JBoss)
 - EclipseLink (referentna implementacija, Oracle)
 - ObjectDB
 - ...



JPA BEANS: ŽIVOTNI CIKLUS

14





REFERENCE

- ◆ **PRIMERI PO UZORU NA** <https://github.com/mbranko/isa19/tree/master/07-orm>
- ◆ **FOWLER M. ORMHATE** <https://martinfowler.com/bliki/OrmHate.html>
- ◆ **DREWSKY, HIGHSCALABILITY.COM. THE CASE AGAINST ORM FRAMEWORKS IN HIGH SCALABILITY ARCHITECTURES**
<http://highscalability.com/blog/2008/2/2/the-case-against-orm-frameworks-in-high-scalability-architec.html>
- ◆ **IRELAND C. ET AL. A CLASSIFICATION OF OBJECT-RELATIONAL IMPEDANCE MISMATCH. FIRST INTERNATIONAL CONFERENCE ON ADVANCES IN DATABASES, KNOWLEDGE, AND DATA APPLICATIONS.** <https://ieeexplore.ieee.org/document/5071809>
- ◆ **TORRES A. ET AL. TWENTY YEARS OF OBJECT-RELATIONAL MAPPING: A SURVEY ON PATTERNS, SOLUTIONS, AND THEIR IMPLICATIONS ON APPLICATION DESIGN**
<https://www.sciencedirect.com/science/article/abs/pii/S0950584916301859>
- ◆ **CHEN T. P. ET AL. DETECTING PERFORMANCE ANTI-PATTERNS FOR APPLICATIONS DEVELOPED USING OBJECT-RELATIONAL MAPPING**
<https://bit.ly/2Pbh6hX>
- ◆ **FOWLER M. PATTERNS OF ENTERPRISE APPLICATION ARCHITECTURE**
<https://martinfowler.com/eaCatalog/>

**KOJA SU VAŠA
PITANJA?**