

SERIJALIZACIJA PODATAKA

ŠTA JE POTREBNO ZA KOMUNIKACIJU?

◆ INTERFEJS

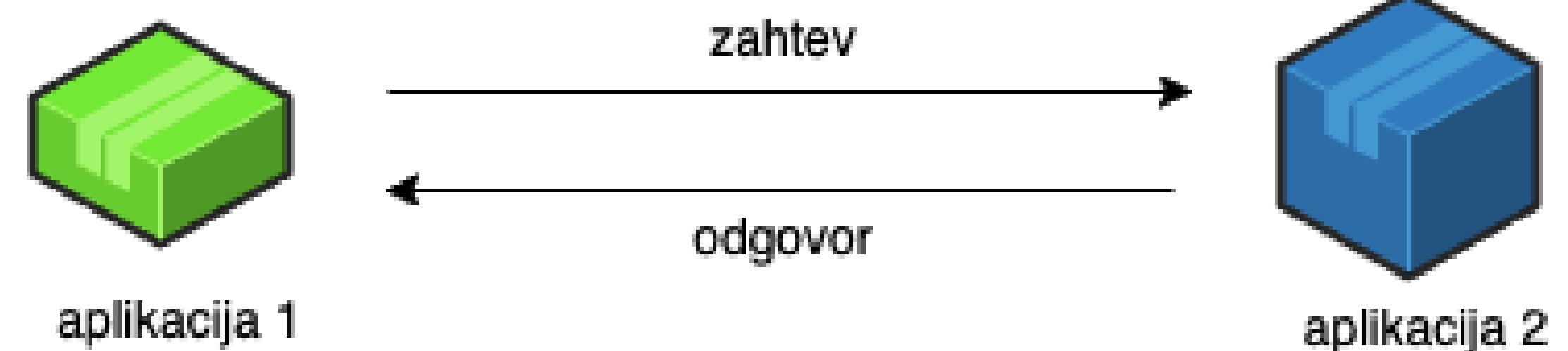
- Deljena granica preko koje dve ili više odvojenih komponenti sistema razmenjuju informacije [1]

◆ PROTOKOL

- HTTP
- MQTT
- SMTP
- ...

◆ FORMAT PORUKE

- Kroz striktno definisanu šemu
- Kroz neformalni dogovor



FORMATI ZA SERIJALIZACIJU PORUKA

◆ POSTOJE BAR DVE REPREZENTACIJE PODATAKA

- U memoriji se čuvaju u strukturama, listama, heš tabelama, stablima, objektima, ...
- Za čuvanje u datoteku na disku ili slanje preko mreže mora se izvršiti konverzija u nekakav niz bajtova

◆ SERIJALIZACIJA (MARSHALLING)

- Konverzija iz reprezentacije podatka u memoriji u bajt sekvencu

◆ DESERIJALIZACIJA (UNMARSHALLING)

- Konverzija iz bajt sekvene u reprezentaciju podatka u memoriji

PROGRAMSKI JEZICI IMAJU UGRAĐENU PODRŠKU

◆ ZA KONVERZIJU

- Java – Serializable
- Python – pickle
- Kryo (3rd party implementacija za Javu)
- ...

◆ PREDNOST

- Sa minimalnom količinom koda se može izvršiti konverzija podatka

◆ MANE

- Čvrsta veza sa programskim jezikom
- Za potrebe deserijalizacije, aplikacija 2 treba da ima mogućnost da konstruiše iste klase kao aplikacija 1 (mimic)
- Lošije performanse od alternativa

PREDNOST

- Mogu ih kreirati i čitati različiti programski jezici
 - JSON
 - XML
 - CSV

MANE

- Problem reprezentacije brojeva
- Nedostatak obavezne validacije šeme
- Nedostatak podrške za binarne stringove [1]

JSON:

```
{  
    "id": "17",  
    "name": "Pera",  
    "lastname": "Perić",  
}
```

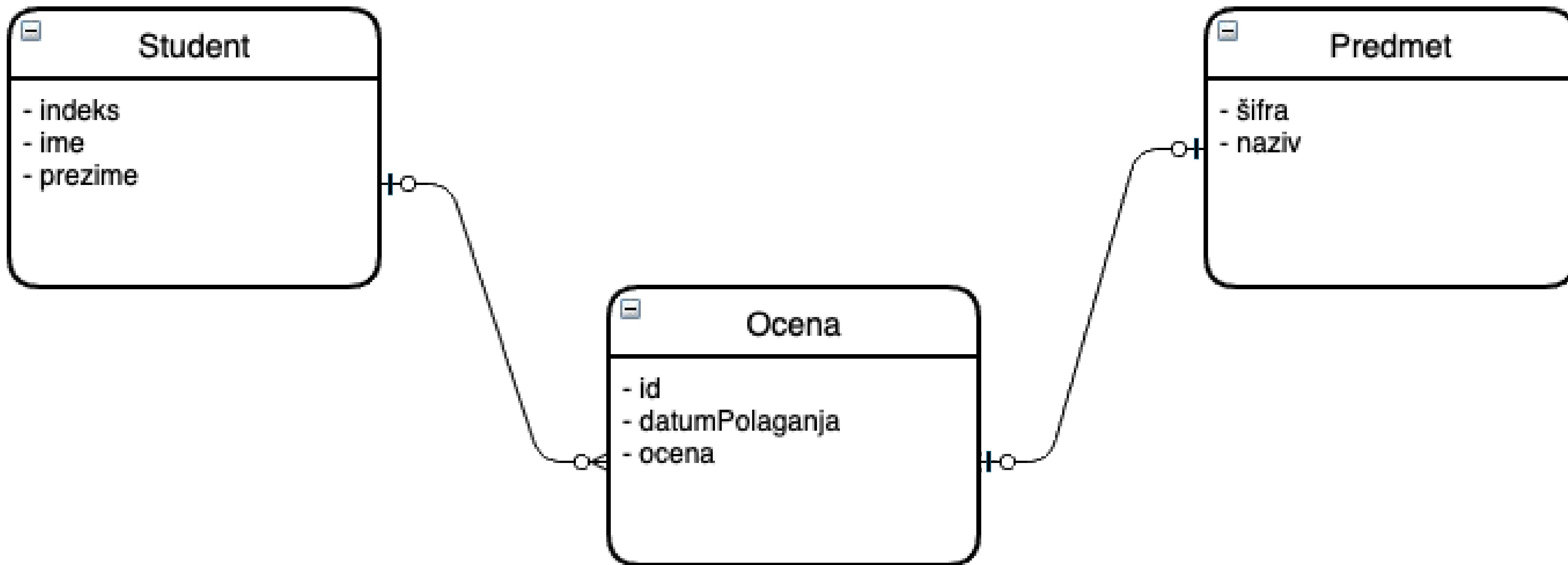
XML:

```
<student>  
    <id>17</id>  
    <name>Pera</name>  
    <lastname>Perić</lastname>  
</student>
```

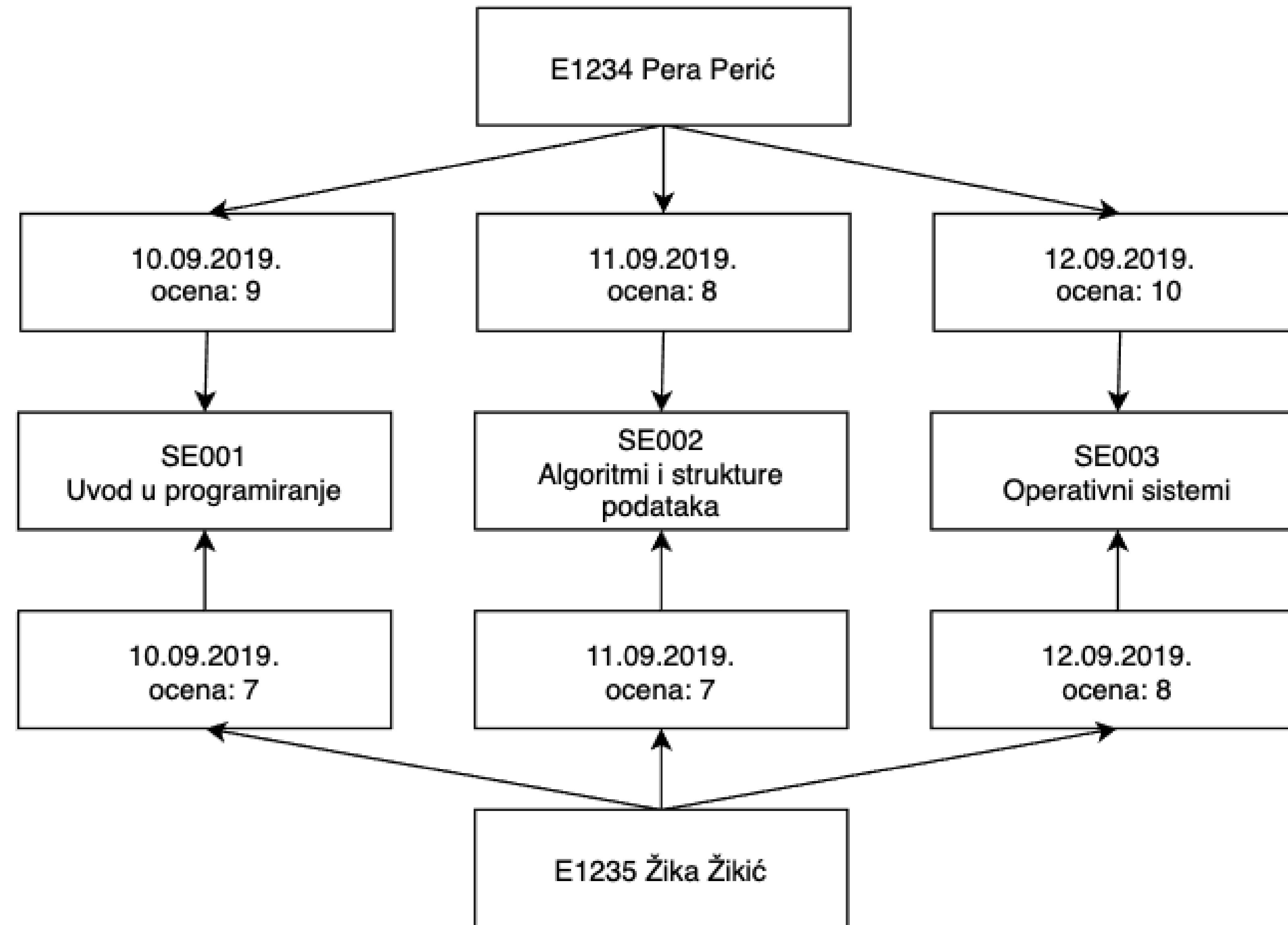
CSV:

17, Pera, Perić

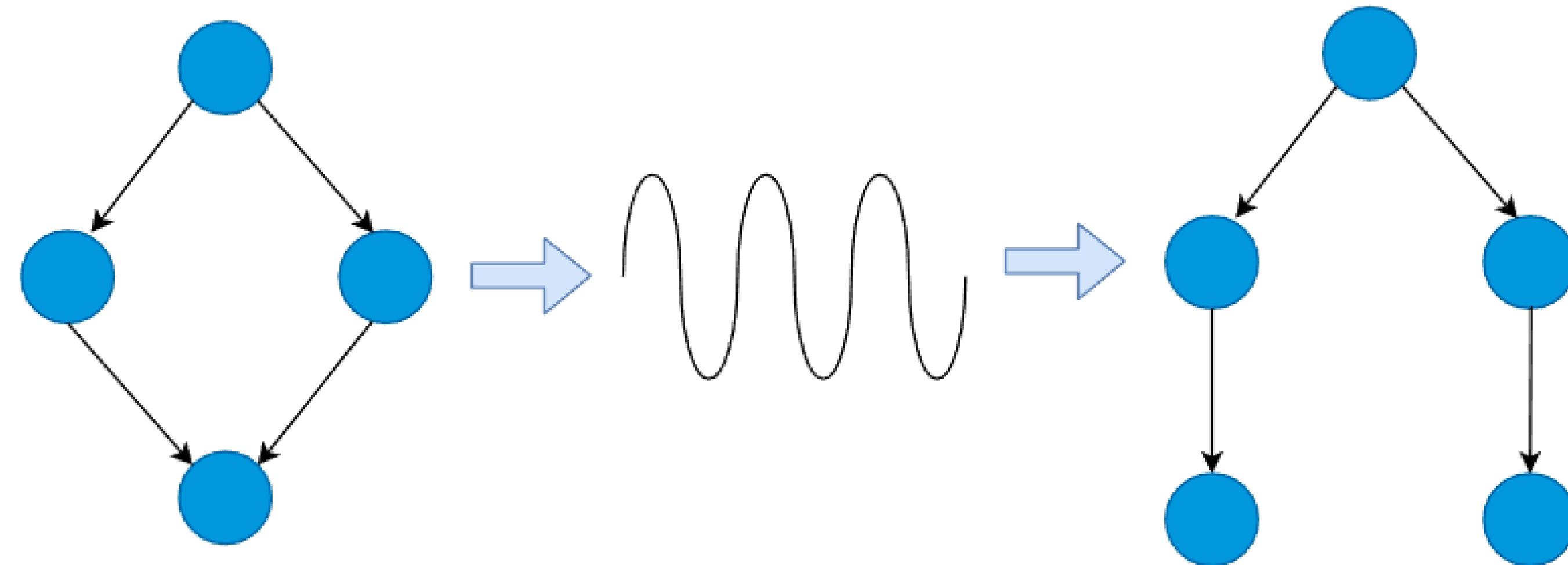
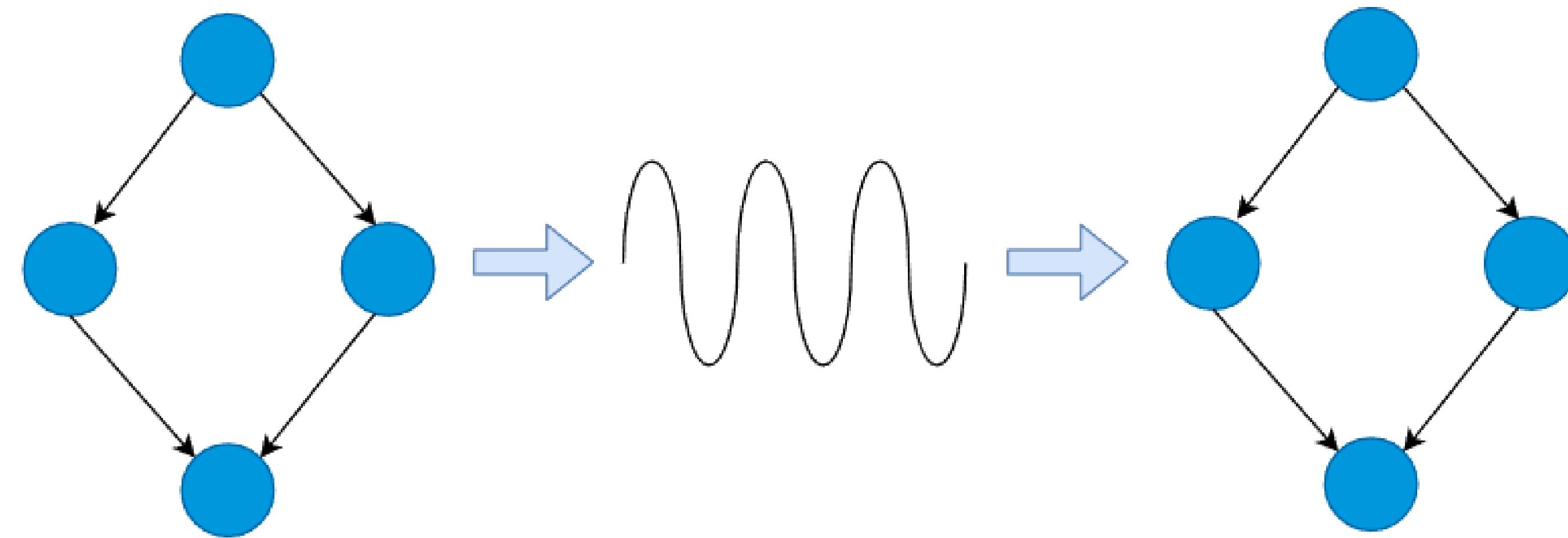
PROBLEM



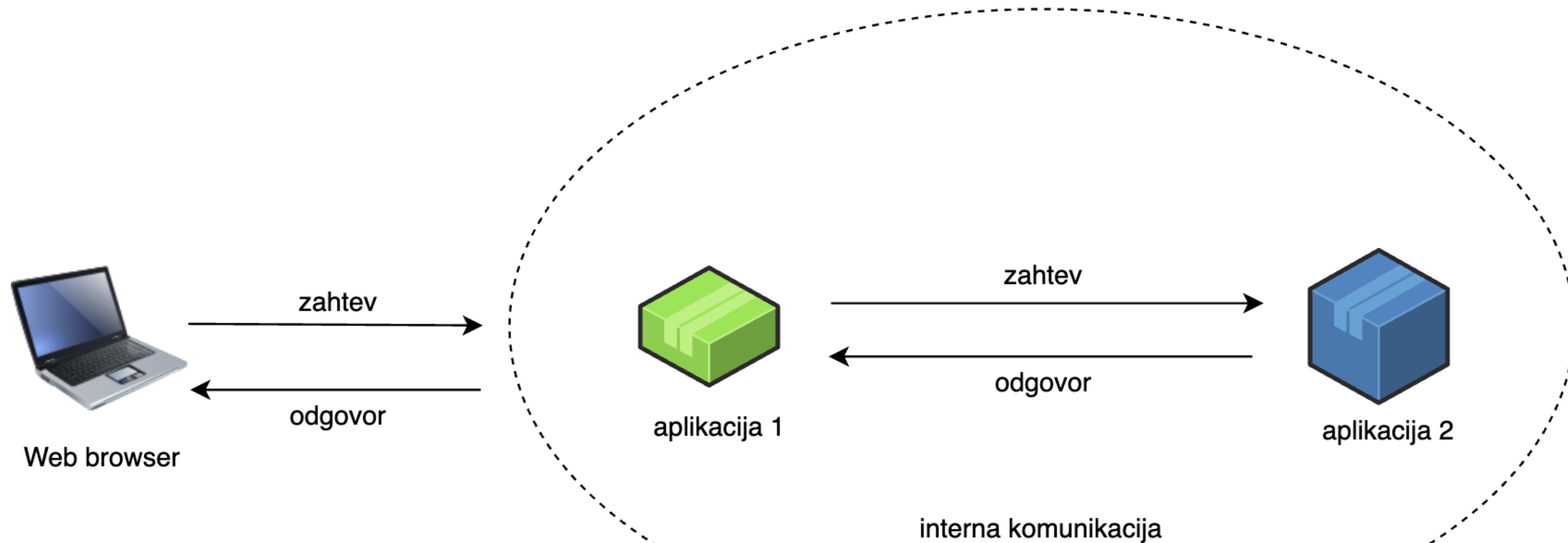
PROBLEM



PROBLEM



BINARNI FORMATI





◆ POSTAJU POPULARNI KAO ALTERNATIVA TEKSTUALNIM

- Protocol Buffers (protobuf)
- Apache Thrift (TBinaryProtocol, TCompactProtocol)
- Apache Avro
- ...



PREDNOSTI

- Poruke zauzimaju manje prostora
- Obavezna validacija u odnosu na šemu
- Generisanje koda na osnovu šeme za lakše korišćenje
- Mogućnost verzionisanja poruka



MANE

- Mora se definisati šema, tj. izgled poruke, što je podložno greškama
- Generisanje koda zahteva korišćenje generatora, tj. savladavanja nečeg novog



BINARNI FORMATI

11

Protobuf schema:

```
message Student {  
    required int64 id = 1;  
    required string name = 2;  
    required string lastname = 3;  
}
```

Thrift schema:

```
struct Student {  
    1: required i64 id,  
    2: required string name,  
    3: required string lastname  
}
```

Avro IDL (Interface Definition Language) schema:

```
record Student {  
    int id;  
    string name;  
    string lastname;  
}
```

BINARNI FORMATI

1. Kreiraj šemu

.PROTO

```
message Student {  
    required int64 id = 1;  
    required string name = 2;  
    required string lastname = 3;  
}
```

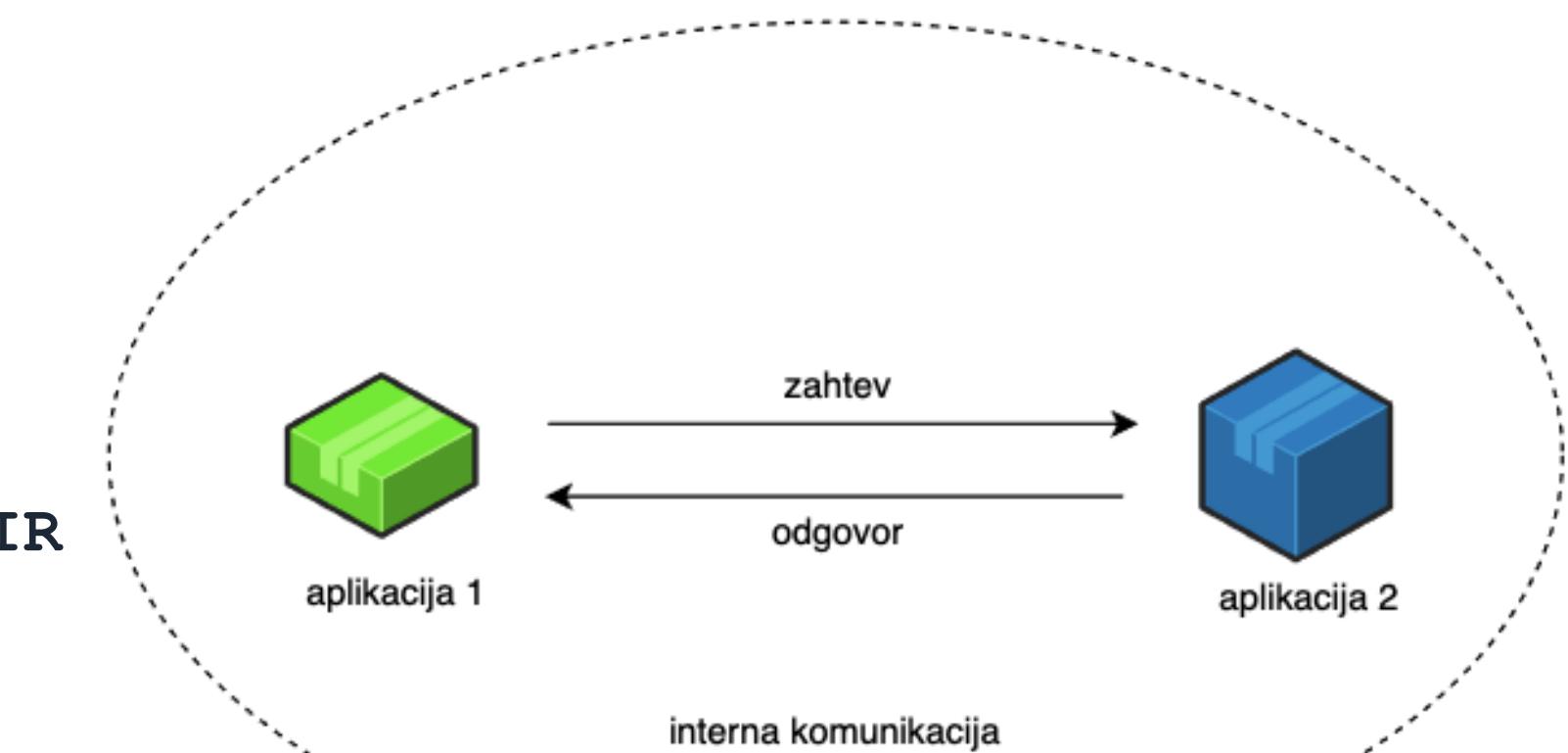
2. Generiši datoteke za željeni jezik i ne menjaj ih

GENERIŠI

```
protoc -I=$SRC_DIR --java_out=$DST_DIR  
$SRC_DIR/student.proto  
  
protoc -I=$SRC_DIR --python_out=$DST_DIR  
$SRC_DIR/student.proto
```

3. Implementiraj klijentsku i serversku aplikaciju koje će raditi serijalizaciju/deserijalizaciju

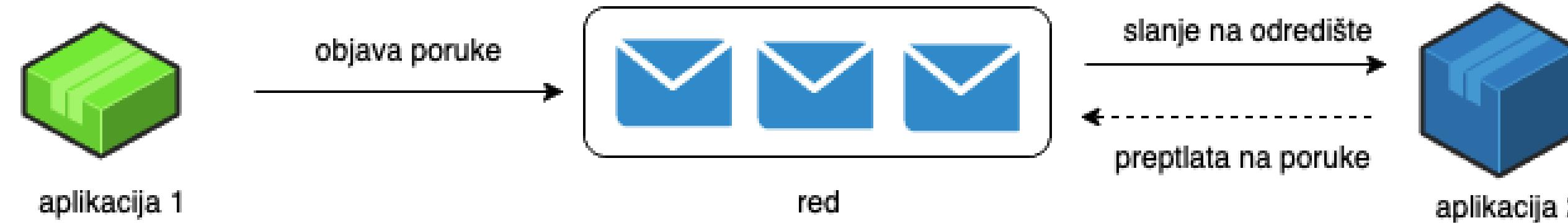
IMPLEMENTIRAJ



SCENARIJI TOKOVA PODATAKA

◆ TRI NAJČEŠĆA SCENARIJA

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





REFERENCE

- ◆ PRIMERI PO UZORU NA <https://github.com/mbranko/isa19/tree/master/02-serialization>
- ◆ JAVA OBJECT SERIALIZATION SPECIFICATION <https://bit.ly/3tJnz3W>
- ◆ PYTHON PICKLE <https://docs.python.org/3/library/pickle.html>
- ◆ GOOGLE PROTOCOL BUFFERS <https://developers.google.com/protocol-buffers>
- ◆ APACHE THRIFT PROTOCOL <https://thrift.apache.org/>
- ◆ APACHE AVRO <https://avro.apache.org/docs/current/>
- ◆ MARTIN KLEPPMANN. DESIGNING DATA-INTENSIVE APPLICATIONS <https://bit.ly/30gFSz3>
- ◆ X IDS <https://docs.x.com/fundamentals/x-ids>
- ◆ NETFLIX FALCOR JSON GRAPH <https://netflix.github.io/falcor/documentation/jsongraph.html>
- ◆ BENCHMARK SERIALIZERS <https://github.com/eishay/jvm-serializers/wiki>

**KOJA SU VAŠA
PITANJA?**