SERIJALIZACIJA PODATAKA

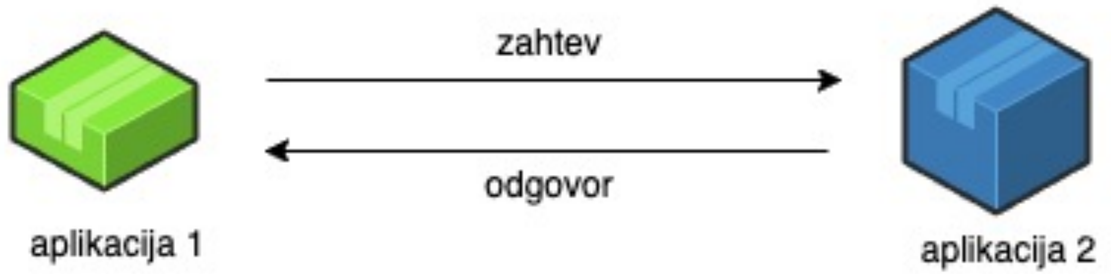
ŠTA JE POTREBNO ZA KOMUNIKACIJU?

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

[1] Interface (computing), https://en.wikipedia.org/wiki/Interface_(computing)

FORMATI ZA SERIJALIZACIJU PORUKA

POSTOJE BAR DVE REPREZENTACIJE PODATAKA

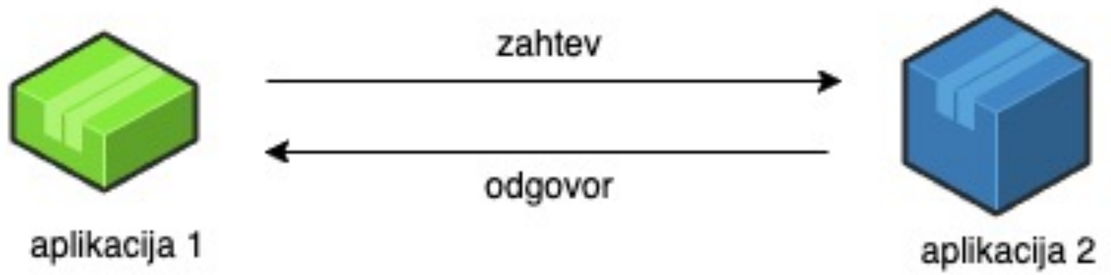
- U memoriji se čuvaju u strukturama, listama, heš tabelama, stablima, objektima, ...
- Za čuvanje u fajl na disku ili slanje preko mreže mora se izvršiti konverzija u nekakav niz bajtova

SERIJALIZACIJA (MARSHALLING)

- Konverzija iz reprezentacije podatka u memoriji u bajt sekvencu

DESERIJALIZACIJA (UNMARSHALLING)

- Konverzija iz bajt sekvence u reprezentaciju podatka u memoriji



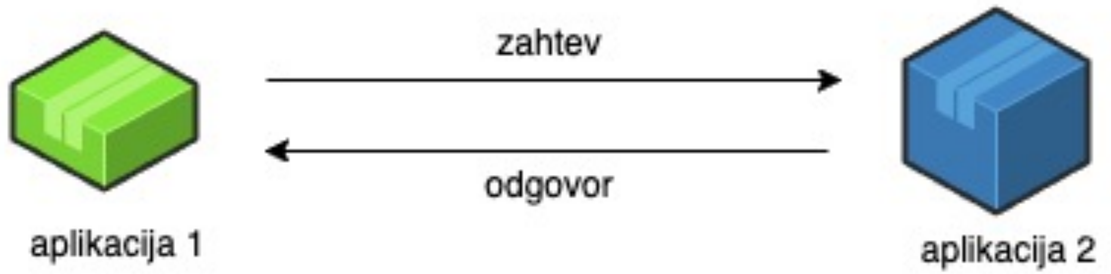
FORMATI SPECIFIČNI ZA JEZIK

PROGRAMSKI JEZICI IMAJU UGRAĐENU PODRŠKU ZA KONVERZIJU

- Java – Serializable
- Python – pickle
- ...

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)
- Loše performanse

primeri: serijalizacijaJava; serijalizacijaJavaProto

STANDARDIZOVANI TEKSTUALNI FORMATI

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

JSON:

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

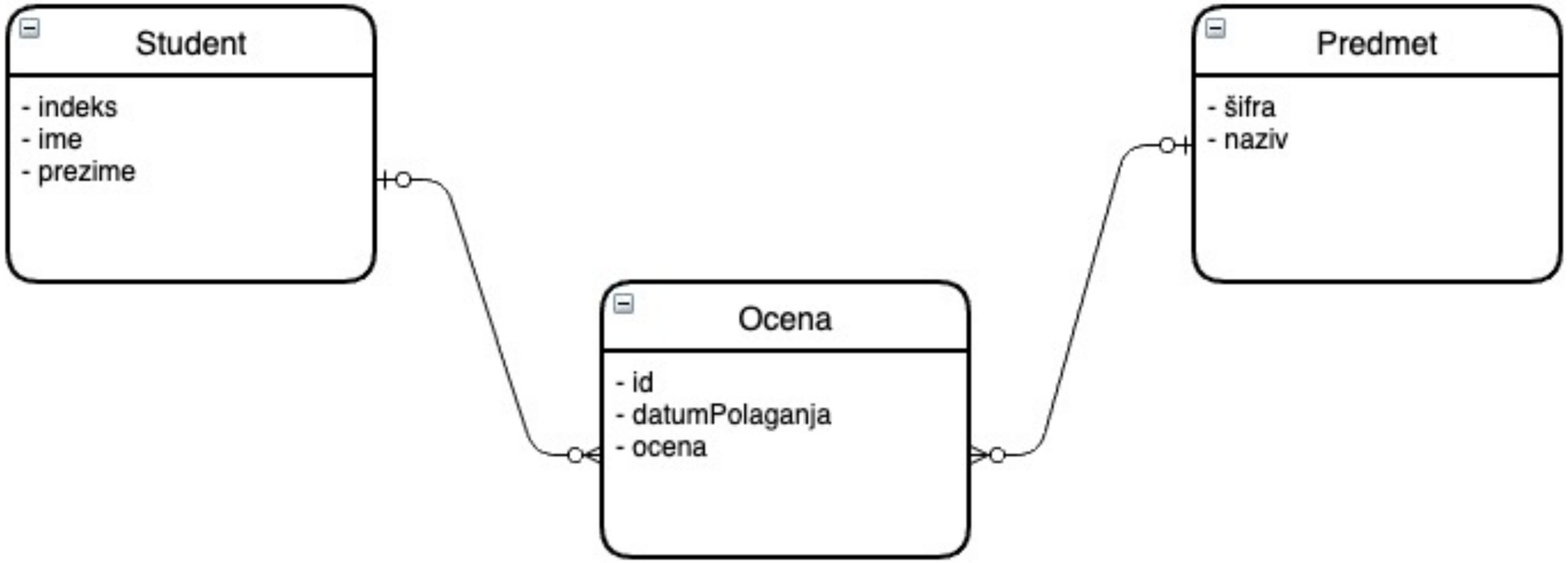
XML:

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

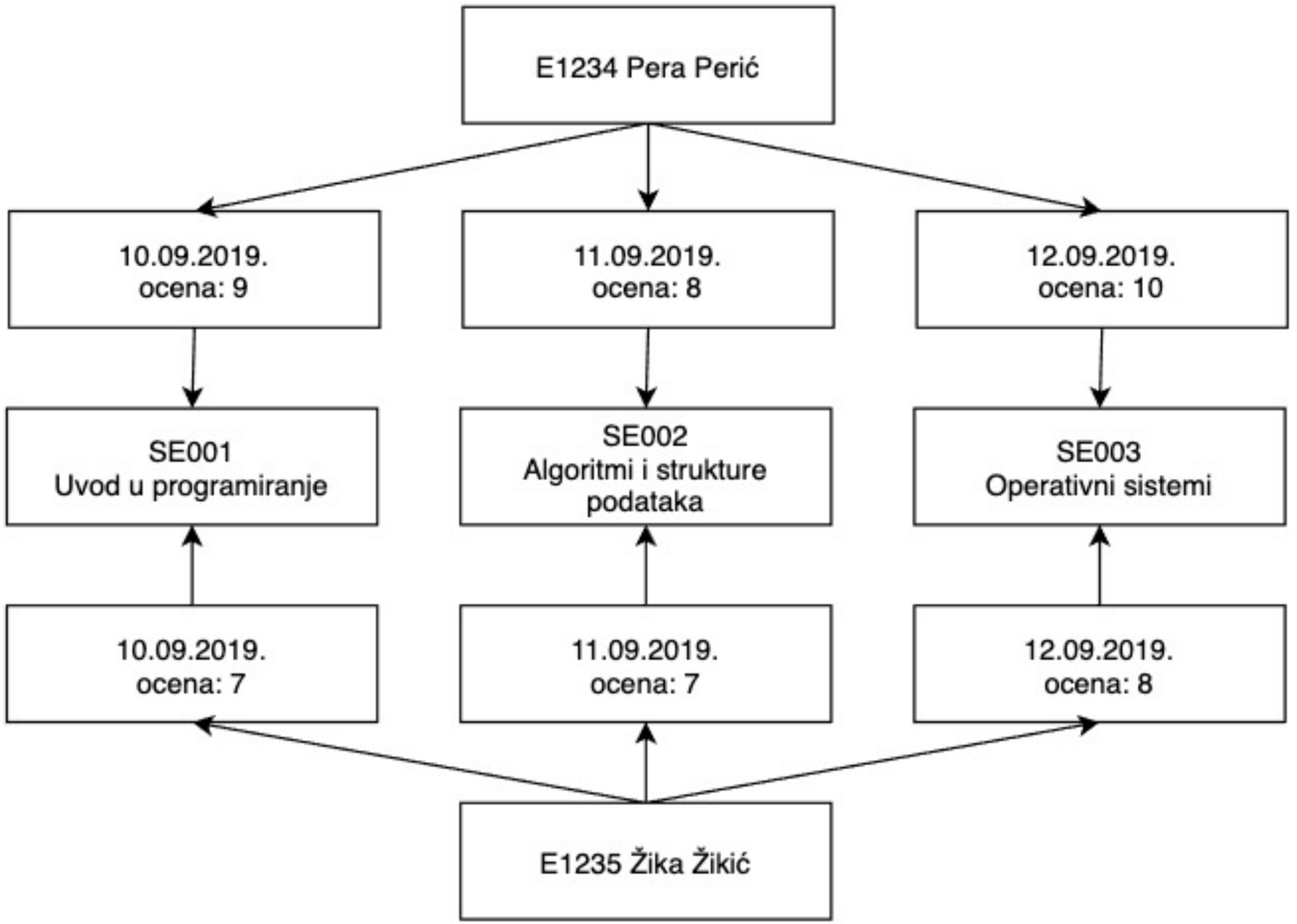
CSV:

```
17,Pera,Perić
```

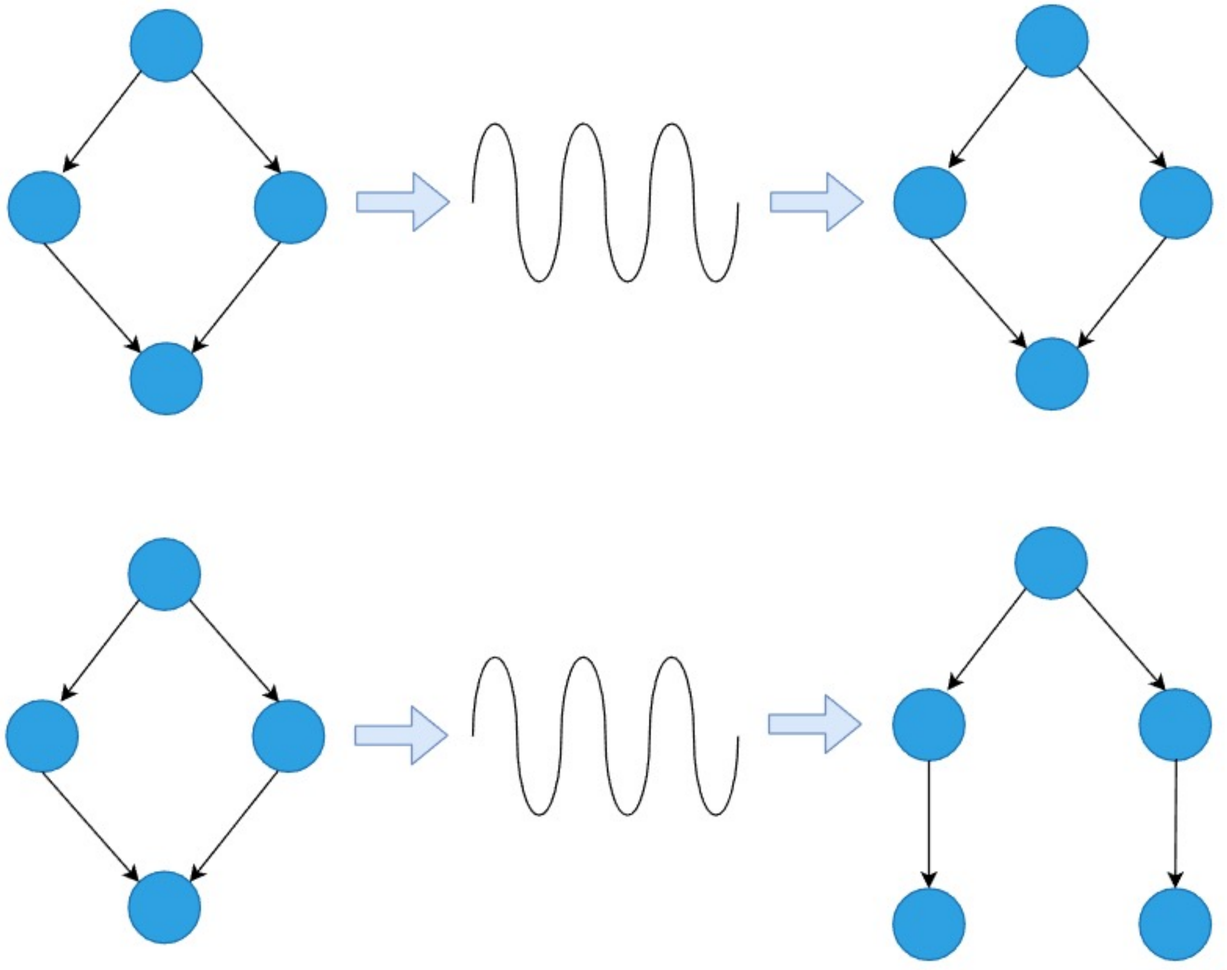
PROBLEM


Student
- indeks
- ime
- prezime
Predmet
- šifra
- naziv
Ocena
- id
- datumPolaganja
- ocena

PROBLEM


E1234 Pera Perić
10.09.2019.
ocena: 9
11.09.2019.
ocena: 8
12.09.2019.
ocena: 10
SE001
Uvod u programiranje
SE002
Algoritmi i strukture podataka
SE003
Operativni sistemi
10.09.2019.
ocena: 7
11.09.2019.
ocena: 7
12.09.2019.
ocena: 8
E1235 Žika Žikić

PROBLEM



BINARNI FORMATI

POSTAJU POPULARNI KAO ALTERNATIVA TEKSTUALNIM

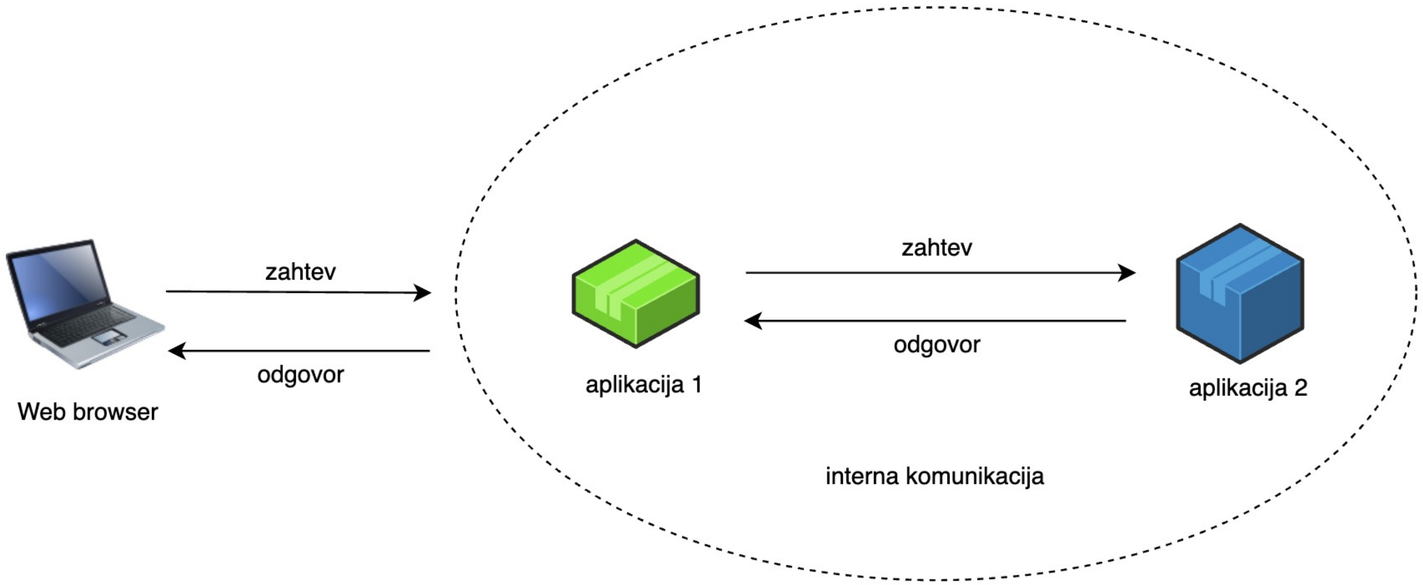
- Protocol Buffers (protobuf)
- Apache Thrift
- Apache Avro
- FlatBuffers

PREDNOSTI

- Manje prostora zauzimaju poruke
- 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

POSTAJU POPULARNI KAO ALTERNATIVA TEKSTUALNIM

- Protocol Buffers (protobuf)
- Apache Thrift
- Apache Avro
- FlatBuffers

PREDNOSTI

- Manje prostora zauzimaju poruke
- 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

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 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 fajlove za željeni jezik
i ne menjaj ih

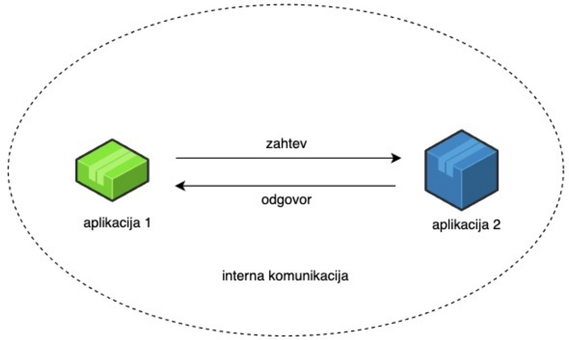
GENERATE

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

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

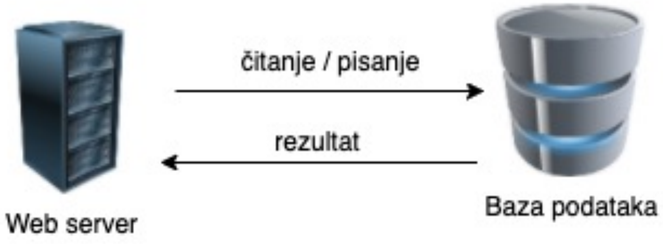
3. Implementiraj klijentsku i serversku
aplikaciju koje će raditi
serijalizaciju/deserijalizaciju

IMPLEMENTATION



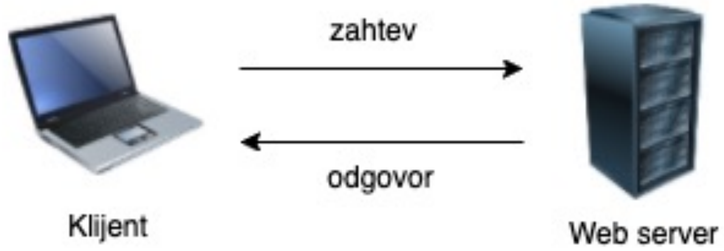
primer: serijalizacijaProto

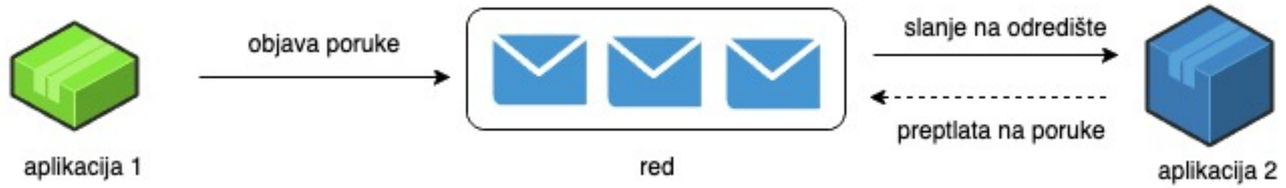
SCENARIJI TOKOVA PODATAKA



TRI NAJČEŠĆA SCENARIJA U KOJIMA SE RAZMENJUJU PODACI

- Serverska aplikacija <-> Baza podataka
- Direktna komunikacija klijent <-> server kroz poziv servisa
- 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://docs.oracle.com/javase/7/docs/platform/serialization/spec/serialTOC.html
- 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/
- DESIGNING DATA-INTENSIVE APPLICATIONS by MARTIN KLEPPMANN https://bit.ly/30gFSz3
- TWITTER API IDs https://developer.twitter.com/en/docs/twitter-ids
- MICROSOFT – MESSAGE ENCODING CONSIDERATIONS https://docs.microsoft.com/en-us/azure/architecture/best-practices/message-encode
- WOLNIKOVSKI ET AL. ZERIALIZER: TOWARDS ZERO-COPY SERIALIZATION https://dl.acm.org/doi/pdf/10.1145/3458336.3465283
- SUMARAY A., KAMI MAKKI S. A COMPARISON OF DATA SERIALIZATION FORMATS FOR OPTIMAL EFFICIENCY ON A MOBILE PLATFORM https://dl.acm.org/doi/pdf/10.1145/2184751.2184810
- POPIC ET AL. PERFORMANCE EVALUATION OF USING PROTOCOL BUFFERS IN THE INTERNET OF THINGS COMMUNICATION https://bit.ly/3GsvbxA

KOJA SU VAŠA PITANJA?