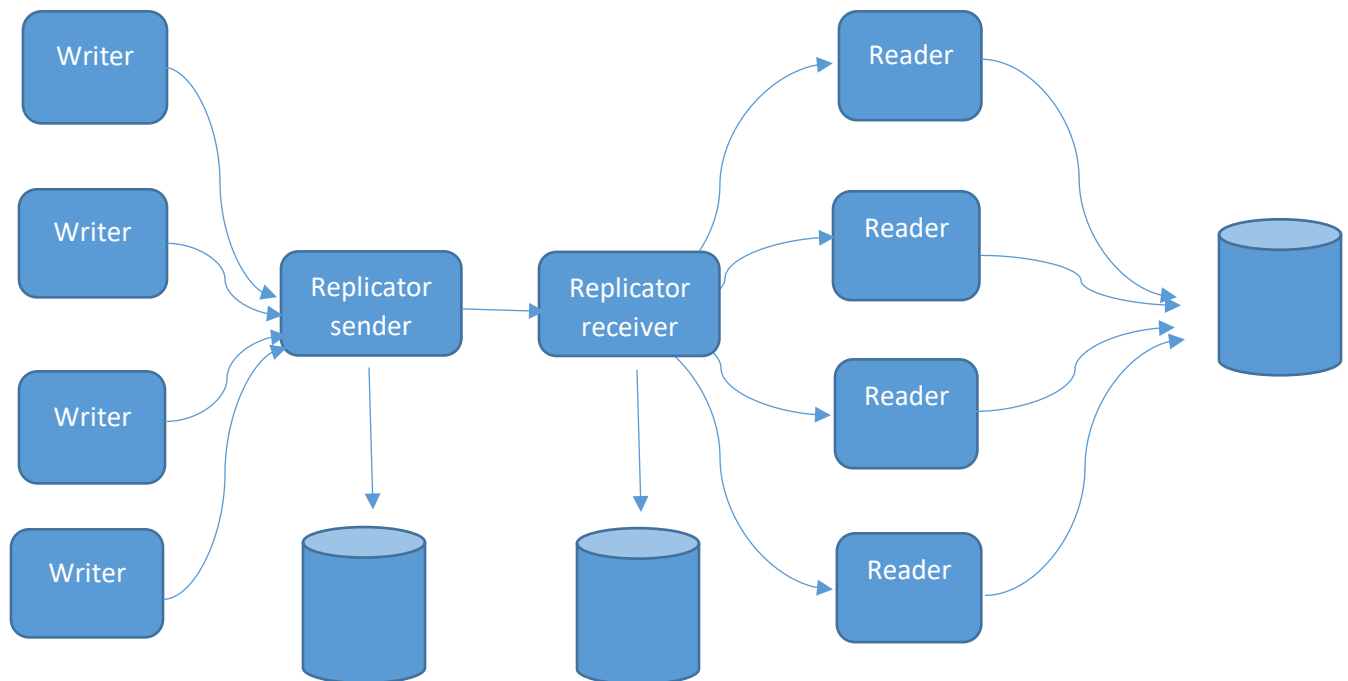


## Projektni Zadatak 4

Potrebno je napraviti dizajn sistema, arhitekturu sistema, implementira i istestirati rešenje koji simulira rad i komunikaciju Replicator modula.

Sistem sadrži 4 komponente:

1. Replicator
2. Writer
3. Reader



### Replicator

Replicator je komponenta koja je sastavljena od 2 komponente, Replicator sender i Replicator receiver komponenti. Služi za repliciranje podataka dobijenih od Writer komponente. Replicator sender prima podatke od Writer komponente i prosleđuje ih Replicator receiver komponenti. Replicator komponenta privremeno podatke smešta kod sebe pre slanja.

Replicator receiver je komponenta koja služi za prijem podataka dobijenih od Replicator sender komponente i slanje tih podataka nekom od Readera u zavisnosti od dataset-a.

Prilikom obrade podataka Replicator receiver komponenta radi sa svojom internom strukturom – CollectionDescription (CD):

CollectionDescription sadrži:

- ID
- Dataset
- HistoricalCollection

HistoricalCollection sadrži:

- Niz ReceiverProperty -a

ReceiverProperty sarži:

- Code
- ReceiverValue

Replikator receiver komponenta salje podatke Reader-u preko DeltaCD komponente:

DeltaCD

- List<CollectionDescription> add;
- List<CollectionDescription> update;

## Writer

Writer je komponenta koja služi za upisivanje novih podataka u Replicator komponentu. Writer komponenti može biti više. Writer komponenta šalje nove podatke svake 2 sekunde Replicator komponenti. Paljenje i gašenje Writera se radi ručno. Podaci koje Writer šalje su:

- Code
- Value

## Reader

Reader je komponenta koja služi za prijem i skladištenje podataka pristiglih od Replicator komponente u bazu. Reader komponenti ima ukupno 4 i svaki radi sa svojim dataset-om. Reader takodje služi i za čitanje podataka iz baze. Reader komponenta podatke dobijene od Replicator komponente snima u bazu. Snimanje u bazu se vrši prema dataset-u. Za svaki dataset treba da postoji posebna tabela/fajl koji će čuvati izmene.

## Scenario rada aplikacije

Writer komponenta radi sledeće:

1. Šalje podatke na svake 2 sekunde Replicator Sender komponenti
2. Inicira paljenje novih Writera
3. Inicira gašenje postojećih Writera

Replicator sender prima podatke od Writera i smešta ih kod sebe u bafer. Podaci se iz bafera skidaju i šalju Replicator receiver komponenti na obradu.

Replicator receiver komponenta prima sadržaj od Replicator sender komponente i pakuje taj sadržaj u svoju internu strukturu – CD. Replicator receiver komponenta treba da vodi računa o dataset-u prilikom obrade, da se podaci pakuju u određeni dataset.

Replicator receiver inicira slanje Reader komponenti samo kad je dataset popunjen, da su obe vrednosti u strukturi pre snimanja u bazu.

Replicator receiver komponenta treba da prepakuje CD koju dobije od Replicator Sender komponente u strukturu DeltaCD koju ce proslediti Reader-u. Slanje ce biti izvršeno ukoliko sledeci uslov bude zadovoljen:

`List<CollectionDescription> add.count + List<CollectionDescription> update.count = 10;`

Reader komponenta treba da proveriti da li dobijeni podaci treba da budu upisani u bazu. Uslov da se podatak upiše u bazu je da izlazi iz Deadband-a.

Deadband predstavlja uslov da li je potrebno podatak upisati u bazu.

Deadband iznosi 2% i to znači da ukoliko pristigli podatak, već postoji u bazi, ukoliko je njegova nova vrednost različita više od 2% od stare vrednosti, tada će biti upisana nova vrednost. Ukoliko nova vrednost ne izlazi iz okvira od 2% od stare vrednosti tada nova vrednost ne treba da bude upisana u bazu.

Jedini izuzetak iz Deadband-a je Code – CODE\_DIGITAL, za ovaj Code se uvek upisuje prosleđena vrednost i ne proverava se Deadband.

Prilikom upisa podatka u bazu, Reader komponenta će generisati timestamp sa vremenom upisa tog podatka, i timestamp takođe treba da bude upisan u bazu zajedno sa podatkom koji se upisuje.

Reader komponenta radi dobavljanje poslednjih vrednosti izabranih Code-ova.

Reader komponenta treba da iščita istorijske vrednosti po vremenskom intervalu za traženi Code.

Implementirati Logger, koji će beležiti sve aktivnosti koje se dešavaju u komponentama.

#### Lista Code-ova:

1. CODE\_ANALOG
2. CODE\_DIGITAL
3. CODE\_CUSTOM
4. CODE\_LIMITSET
5. CODE\_SINGLENODE
6. CODE\_MULTIPLENODE
7. CODE\_CONSUMER
8. CODE\_SOURCE

#### Spisak DataSet-ova po Code-ovima:

- DataSet = 1 – CODE\_ANALOG, CODE\_DIGITAL
- DataSet = 2 – CODE\_CUSTOM, CODE\_LIMITSET

- DataSet = 3 – CODE\_SINGLENODE, CODE\_MULTIPLENODE
- DataSet = 4 – CODE\_CONSUMER, CODE\_SOURCE

## Kriterijum ocenjivanja

1. Dizajn I arhitektura rešenja
2. Korišćenje Scrum metodologije razvoja – definisanje User Story-a i taskova, planiranje i estimacija
3. Implementacija rešenja
4. CI ciklus
  - a. Build
  - b. UnitTestovi
  - c. Pokrivenost koda testovima