

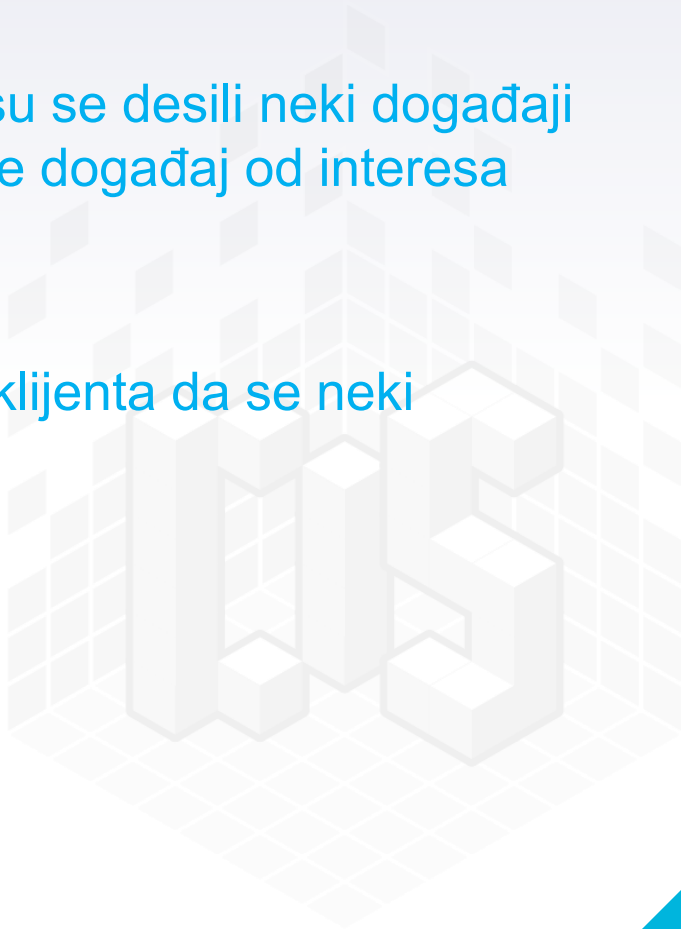
DISTRIBUIRANI SISTEMI

RMI – CALLBACK



Callback

- Ideja: umesto da klijent ispituje stalno da li su se desili neki događaji na serveru, server ih informiše svaki put kad se događaj od interesa desi
- Callback = serverska akcija obaveštavanja klijenta da se neki događaj desio



Callback implementacija

- Klijent kreira udaljeni objekat **Callback objekat** koji implementira interfejs koji sadrži metod koji će server da zove kad se desi događaj
- Server obezbeđuje operaciju koja omogućava klijentima da ga obaveste o remote object referenci na njihov **Callback objekat** (register metod)
- Kad se događaj od interesa desi, server obaveštava zainteresovane klijente o tome pozivom odgovarajućeg metoda

PRIMER: Whiteboard

- Napisati RMI aplikaciju koji omogućava grupi korisnika (klijenata) da dele zajednički pogled na površinu za crtanje koja sadrži grafičke objekte, kao što su pravougaonici, linije i krugovi, od kojih se svaki oblik crta od strane jednog od korisnika.
- Server održava postojeće stanje crteža evidentiranjem svih pristiglih objekata.
- Server takođe obezbeđuje operacije koje omogućavaju vraćanje do tada upisanih objekata.
- Server sadrži broj verzije (integer) koji inkrementira svaki put kada novi oblik stigne i pridružuje ga novom obliku.
- Server obezbeđuje operacije koje omogućavaju klijentima da se raspitaju o najnovijem broju verzije objekata.

PRIMER: Whiteboard

REŠENJE KOJE
NE KORISTI
CALLBACK

VS

REŠENJE KOJE
KORISTI
CALLBACK

+ Interface IWhiteboardCallback

+ register(IWhiteboardCallback cb)

+ unregister(IWhiteboardCallback cb)

Callback interfejs

```
public interface IWhiteboardCallback extends Remote {  
    void callback (int version) throws RemoteException;  
}
```



Callback – implementacija - WhiteboardClient

```
public class WhiteboardClient {  
    ...  
    public class WhiteboardCallback extends UnicastRemoteObject implements  
        IWhiteboardCallback {  
  
        public WhiteboardCallback() throws RemoteException  
        {  
        }  
  
        public void callback(int version) throws RemoteException  
        {  
            showWhiteboard(); /// Metod "roditeljske" klase  
        }  
    }  
    ...  
}
```

Callback – registracija - WhiteboardManager

```
public synchronized void register(IWhiteboardCallback c) throws  
RemoteException {  
    clients.add(c);  
}
```

```
public synchronized void unregister(IWhiteboardCallback c) throws  
RemoteException {  
    clients.remove(c);  
}
```

