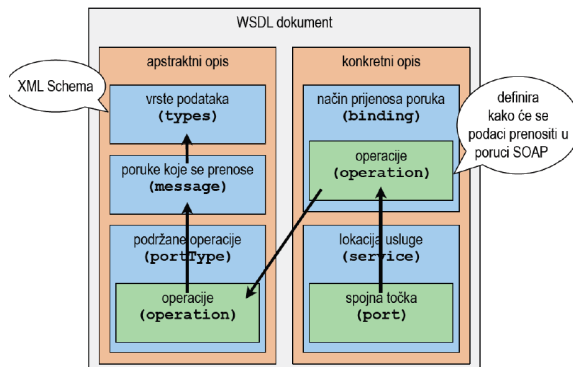
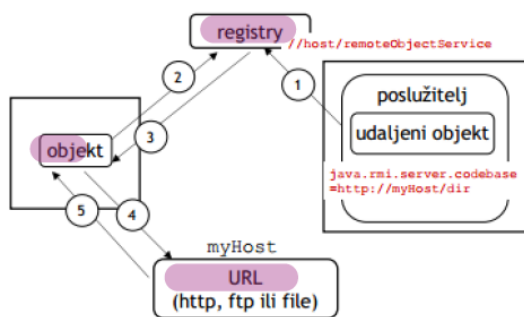


1. Poslužitelj registrira udaljeni objekt pod odabranim imenom.
2. Klijent od registry traži referencu na udaljeni objekt koristeći registrirano ime.
3. Klijent poziva metodu stuba dostupnu na klijentskom računalu.
4. Stub serijalizira parametre i šalje ih skeletonu.
5. Skeleton deserializira parametre i poziva metodu udaljenog objekta.
6. Udaljeni objekt vraća rezultat izvođenja metode skeletonu.
7. Skeleton serijalizira rezultat i šalje ga stubu.
8. Stub deserializira rezultat i dostavlja ga klijentu.



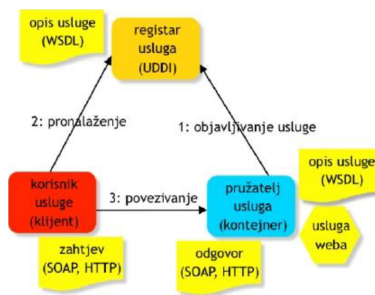
1. types: definira vrste podataka neovisne o platformi i jeziku ,
2. message: definiraju ulazne i izlazne poruke koje se mogu koristiti kao parametri usluge,
3. operation: predstavlja jednu operaciju/metodu/proceduru koja je definirana u usluzi, a sastoji se od definicija ulaznih, izlaznih i iznimnih poruka koje se mogu razmjenjivati korištenjem ove operacije i
4. portType: koristi poruke (pod 2) da bi opisao sve operacije koje pruža usluga.
1. binding: definira kako je konkretna implementacija povezana s operacijama u apstraktnom opisu i definira format u kojem će se poruke prenositi (protokol i elemente) i
2. service: definira URI gdje je usluga isporučena tj. na kojoj adresi se može pozvati usluga (taj URI je definiran u spojnoj točki).



1. Poslužitelj definira codebase udaljenog objekta i registrira taj udaljeni objekt pod odabranim imenom
2. Klijent od registry traži referencu na udaljeni objekt pomoću imena
3. Registry vraća podatke o klasi stuba. Ako se klasa stuba može pronaći na klijentskoj strani učitava se lokalna verzija klase. U suprotnom će klijent učitati klasu koristeći definirani codebase.
4. Klijent traži klasu stuba koristeći codebase.
5. Klasa stuba se dostavlja klijentu. Klijent može pozivati metode udaljenog objekta koristeći primljeni stub

Slaba povezanost - odnosi se na dizajn programske izvedbe, jedna promjena ne zahtijeva promjenu druge, svaka usluga može nesmetano koristiti drugu

## arhitektura web usluge

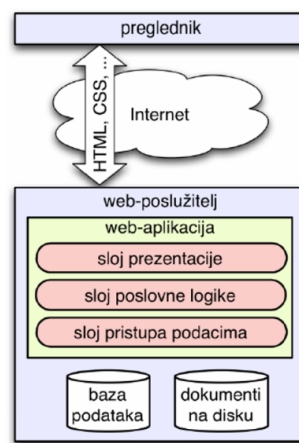


1. Objavljivanje usluge: pružatelj usluge vrši registraciju usluge u Registru usluga koristeći WSDL opis usluge.
2. Pronalaženje usluge: korisnik usluge (klijent) pronalazi odgovarajuću uslugu z registru usluga.
3. Povezivanje: temeljem specifikacije u WSDL opisu usluge, klijent šalje zahtjev za uslugom, te od usluge dobiva odgovor. Ovisno o usluzi, zahtjev i odgovor mogu biti upućeni korištenjem protokola SOAP ili HTTP.

Kategorija	Značajka	KP ili WS
Primjena	Unutar tvrtke	KP
	Između tvrtki	WS
Programski jezici	Neovisnost o programskom jeziku	WS
	Ograničen skup programskih jezika	KP
Komunikacija između komponenti	Proceduralno	KP
	Razmjenoj poruka	WS
Transportni mehanizmi	Različiti transportni mehanizmi	WS
	Jedan transportni mehanizam	KP
Povezanost	Slabo povezane strukture	WS
	Čvrsto povezane strukture	KP
Učinkovitost procesiranja	Velika	KP
	Manja	WS

Potreba za sinkronizacijom procesa  
**UPOTREBA DIJELJENIH SREDSTVA**- izbjeći istovremeni pristup istom sredstvu  
**ALGORITAM ODABIRA VODE** - neophodno da svi procesi budu međusobno sinkronizirani

## Arhitektura web aplikacija



prezentacije - prikaz informacija  
 poslovne logike - obrada podataka od prezentacije  
 pristup podacima - spojen na bazu, pazi na transakcije i podatke