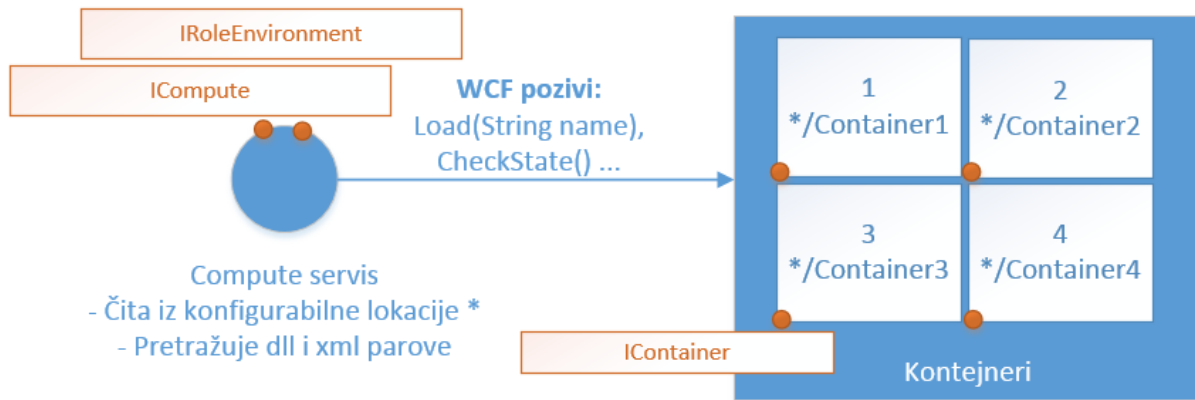


Projektni zadatak B i C – kreiranje PaaS

Definicija zadatka:



Slika 1 - Šematski prikaz zadatka

U skladu sa šematskim prikazom (Slika 1), implementirati *Cloud compute servis*:

- b) Kreirati robustnost kontejnera, tako da ukoliko neki rezultuje otkazom, *ComputeService* servis će to ustanoviti i pokrenuti novi kontejner sa klijentskim programima koji su otkazali zajedno sa kontejnerom. *ComputeService* vrši oporavak na sledeći način:
 1. Periodično proverava stanje kontejnera putem WCF komunikacije i *CheckState* metode, tako što se interfejs iz listinga 1. proširi metodom datom u listingu 2.
 2. Ukoliko je neki od kontejnera rezultovao otkazom, izvršava se oporavak pokretanjem novog kontejnera i pokretanjem paketa koji su bili izvršavani u okviru tog kontejnera. Oporavak se vrši na sledeći način:
 - i. Ukoliko ima slobodnih kontejnera koji već ne izvršavaju klijentski program, prvo pokrenuti novu instancu u okviru slobodnog kontejnera, a zatim krenuti u proceduru startovanja novog kontejnera.
 - ii. Ukoliko su svi kontejneri zauzeti, pokrenuti prvo novi kontejner, a potom učitati klijentski program u novopokrenutom kontejneru.
- c) Obezbediti da *ComputeServis* može da pruža instancama informacije o bratskim instancama putem WCF servera, na sledeći način:
 - a. *ComputeService* sadrži WCF server sa intefejsom iz Listinga 3. Kada se klijentski programi pokreću u okviru kontejnera, oni zatražuju svoju adresu od Compute servisa koristeći metodu *GetAddress()*.
 - b. *ComputeService* servis održava interno gde se koji servis nalazi. Implementira metodu iz listinga 3, *BrotherInstances()* tako što na osnovu imena programa i same instance jedinstveno određene sopstvenom adresom (preporuka je da se koristi samo port kao jedinstveni identifikator), pronalazi preostale portove rezervisane za ostale instance i vraća listu portova kao odgovor.

Listing 1:

```
public interface IContainer
{
    String Load(String assemblyName);
}
```

Listing 2:

```
public interface IContainer
{
    String Load(String assemblyName);

    String CheckState();
}
```

Listing 3:

```
public interface IRoleEnvironment
{
    String GetAddress(String myAssemblyName, String containerId);

    String[] BrotherInstances(String myAssemblyName, String myAddress);
}
```