Backend

WcfService (wcf uygulaması) : Elasticsearch client ile iletişime girip index den verileri alan web servis uygulaması.

ServiceLibrary (shared wcf library) : Shared library olarak WcfService uygulaması ile Front end deki ElasticWCFDataAccess katmanı tarafından kullanılır. ElasticWCFDataAccess ChannelFactory aracılığıyla iletişimin sağlanması için bu library nin servis interfacelerini tutar.

Frontend

KarmasisDomain : Entity objeleri burada bulunur, DataAccess katmanından bilgi almak için gerekli servisler burada bulunur, hiçbir library yi referans olarak almaz ancak onu referans olarak alan library ler vardır, böylelikle hiçbir librarye bağımlı (dependent) değildir.

KarmasisPresentation (ViewModels) : ViewModeller buradadır, viewlere bilgileri nasıl bind edeceğinin logici buradadır, KarmasisDomaini referans olarak alır, Portable dır dolayısıyla Xamarin gibi farklı bir platformda library olarak çalışabilir haldedir, bundan dolayı KarmasisDomain de portable olarak tanımlanmıştır, User interface ile alakalı hiçbir bilgi bulunamaz dolayısıyla UserInterface değiştiğinde bu library, değişen UserInterface ile rahatlıkla çalışabilir. Bu yüzden User Interface e bağımlı değildir. DataAccess den verilerin nasıl alınacağı ve User Interface e nasıl sunulacağının logicini taşır. Unit test yapılabilir haldedir. Contructorlarda unity ile dependency injection yapıldığından, Farklı DataAccess (WebService,SqlServer,Azure.. vs) ler Late binding ile modifiye edilebilir, programın tekrardan derlenmesine gerek kalmadan farklı dataaccess ler ile çalışabilir.(RepositoryPattern)

ElasticWCFDataAccess : WCF servisiyle iletişime geçen veri katmanı, data servisiyle nasıl iletişime geçeceği logici burada tutulur, KarmasisDomain de ki abstract belirlenen repository objelerini implemente eder dolayısıyla ViewModellere dependency injection ile eklenenir.

KarmasisDesktopCore : PC Desktop uygulaması olan KarmasisDesktop için temel classları tutar. Bu temel library KarmasiDesktop User Interface katmanına bağımlıdır.

KarmasisDesktop(Views – UI) : ViewModellerin bulunduğu, user intarface dir, ve Composition Root dur. Unity container a istenilen DataAccess library burada eklenir, DataAccess library referans olarak eklenmiş yada reflection ile directory den alınabilir olabilecek şeklinde ayarlanabilir. (Prism-Modularity). Bu user interface projeden çıkarıldığında uygulamanın diğer katmanlarına bağımlı olmadığından proje halen derlenebilir, Xamarin Forms gibi farklı bir user interface uygulamaya eklenilebilir ve uygulama mobil platformlarda çalışabilir.

