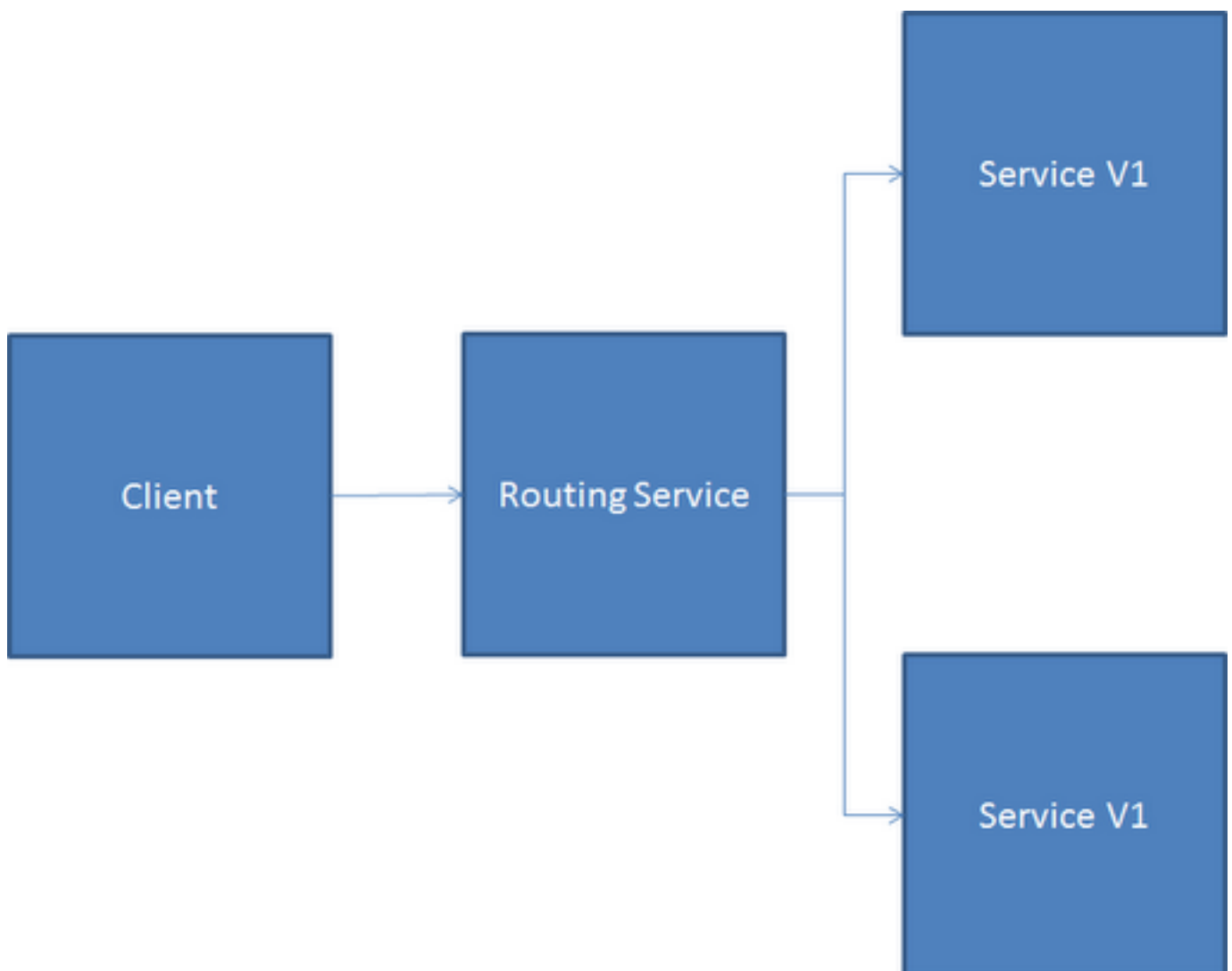


به صورت معمول در سیستم‌های مبتنی بر WCF ارتباط بین سرور و کلاینت در قالب EndpointConfiguration تعریف می‌شوند. یعنی کلاینت برای برقراری ارتباط با سرور نیاز به آدرسی که سرور مورد نظر در آن هاست شده است دارد. این روش هنگامی که فقط یک مقصد وجود داشته باشد روش موثری است. اما اگر سرویس‌های مورد نظر در چند سرور هاست شده باشند نیاز به سیستم مسیریابی خواهیم داشت. خوشبختانه در WCF 4.0 این امکان به خوبی تدارک دیده شده است.

WCF Routing Service چیست؟

Routing Service به عنوان سرویس مسیریابی WCF در دات نت 4 معرفی شد. به وسیله Routing Service می‌توان Endpoint Configuration مقصدهای مختلف را با هم تجمیع کرد و در نتیجه تعداد تنظیمات برای Endpoint در سمت کلاینت کاهش پیدا می‌کند به طوری که کلاینت فقط با یک مقصد در ارتباط است. مقصد کلاینت همان Routing Service می‌باشد که در این سرور درخواست‌های رسیده از کلاینت‌ها با توجه به فیلتر انجام شده به مقصد اصلی ارسال خواهند شد. با استفاده از Routing Service معماری سیستم به صورت تغییر پیدا می‌کند:



اهداف:

موارد زیر اهداف و مزایای استفاده از Routing Service است:

Service versioning «

Content-based routing scenario «

Service partitioning «

Protocol bridging «

هر کدام از موارد بالا در طی پست‌های جداگانه شرح داده خواهند شد.

بررسی یک مثال:

دو Contract به صورت زیر تعریف می‌کنیم:

```
[ServiceContract]
public interface ICalculatorV1
{
    [OperationContract]
    int Add(int a, int b);
}

[ServiceContract]
public interface ICalculatorV2
{
    [OperationContract]
    int Sub(int a, int b);
}
```

پیاده سازی Contract‌های بالا در قالب سرویس به صورت زیر خواهد بود:

```
public class CalculatorV1 : ICalculatorV1
{
    public int Add(int a, int b)
    {
        return a + b;
    }
}

public class CalculatorV2 : ICalculatorV2
{
    public int Sub(int a, int b)
    {
        return a - b;
    }
}
```

تنظیمات Binding سرویس‌ها:

```
system.serviceModel>
<services>
  <service name="WCFRoutingSample.CalculatorV1">
    <host>
      <baseAddresses>
        <add baseAddress = "http://localhost:8732/CalculatorServiceV1/" />
      </baseAddresses>
    </host>

    <endpoint address="" binding="basicHttpBinding" contract="WCFRoutingSample.ICalculatorV1">
```

```

        <identity>
            <dns value="localhost"/>
        </identity>
    </endpoint>

    <endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange"/>
</service>
<service name="WCFRoutingSample.CalculatorV2">
    <host>
        <baseAddresses>
            <add baseAddress = "http://localhost:8733/CalculatorServiceV2/" />
        </baseAddresses>
    </host>

    <endpoint address="" binding="basicHttpBinding" contract="WCFRoutingSample.ICalculatorV2">

        <identity>
            <dns value="localhost"/>
        </identity>
    </endpoint>

    <endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange"/>
</service>
</services>
<behaviors>
    <serviceBehaviors>
        <behavior>
            <serviceMetadata httpGetEnabled="True"/>
            <serviceDebug includeExceptionDetailInFaults="False" />
        </behavior>
    </serviceBehaviors>
</behaviors>
</system.serviceModel>

```

حال باید RoutingService را به صورت زیر هاست نماییم:

```

class Program
{
    static void Main(string[] args)
    {
        var host = new ServiceHost(typeof(RoutingService));
        host.Open();
        Console.WriteLine("Server is running.");
        Console.ReadLine();
        host.Close();
    }
}

```

مهم‌ترین بخش تنظیمات مربوط به Routing Service است:

```

<system.serviceModel>
    <behaviors>
        <serviceBehaviors>
            <behavior name="routingBehv">
                <routing routeOnHeadersOnly="false" filterTableName="filters"/>
                <serviceDebug includeExceptionDetailInFaults="true"/>
                <serviceMetadata httpGetEnabled="true"/>
            </behavior>
        </serviceBehaviors>
    </behaviors>
    <routing>
        <filters>
            <filter name="CalV1ServiceFilter" filterType="EndpointName" filterData="Calv1Service"/>
            <filter name="CalV2ServiceFilter" filterType="EndpointName" filterData="Calv2Service"/>
        </filters>
        <filterTables>
            <filterTable name="filters">
                <add filterName="CalV1ServiceFilter" endpointName="Calv1Service" />
                <add filterName="CalV2ServiceFilter" endpointName="Calv2Service"/>
            </filterTable>
        </filterTables>
    </routing>
    <services>
        <!-- Routing service with endpoint definition -->
    </services>

```

```
<service name="System.ServiceModel.Routing.RoutingService"
  behaviorConfiguration="routingBehv">
  <endpoint
    address="/Calv1"
    binding="basicHttpBinding"
    contract="System.ServiceModel.Routing.IRequestReplyRouter"
    name="Calv1Service"/>
  <endpoint
    address="/Calv2"
    binding="basicHttpBinding"
    contract="System.ServiceModel.Routing.IRequestReplyRouter"
    name="Calv2Service"/>

  <host>
    <baseAddresses>
      <add baseAddress="http://localhost:9000/CalculatorService"/>
    </baseAddresses>
  </host>
</service>
</services>
<client>
  <endpoint address="http://localhost:8732/CalculatorServiceV1"
    binding="basicHttpBinding"
    contract="*"
    name="Calv1Service"/>
  <endpoint address="http://localhost:8733/CalculatorServiceV2"
    binding="basicHttpBinding"
    contract="*"
    name="Calv2Service"/>
</client>
</system.serviceModel>
```

همان طور که مشاهده می‌کنید ابتدا اطلاعات Binding دو سرویس نوشته در بالا را به عنوان Endpoint های مختلف تعریف کردیم و سپس با استفاده از FilterTable نوع درخواست را به Endpoint مورد نظر وصل کردیم (در این مثال فیلتر بر اساس نوع Endpoint است). به وسیله این تعاریف Routing Service می‌داند که هر درخواست را به کدام سرویس ارسال نماید و پاسخ به کجا بازگشت داده شود.