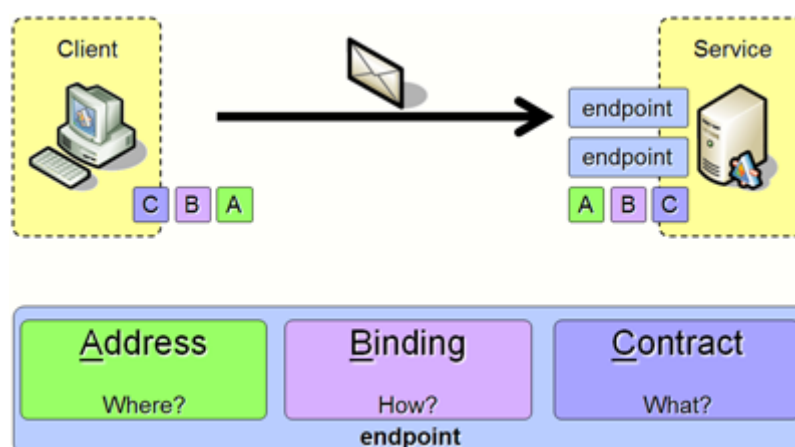


## Windows Communication Foundation (WCF)

1. **WCF:** Windows Communication Foundation; технология основанная на .NET FRAMEWORK для разработки приложений SOA-архитектуры, первая версия 2006 (Indigo).
2. **WCF:** основные принципы
  - разработка сервиса должна быть простой и иметь способность к расширению его функциональных возможностей;
  - один API для всех коммуникационных протоколов;
  - сервис должен быть интероперабельным (функционировать по открытым телекоммуникационным стандартам);
  - сервис должен поддерживать стандарты WS-\*;
  - сервис должен поддерживать REST, RPC и др. архитектуры;
3. **WCF:** коммуникация модель



4. **WCF:** контракты
  - контракт службы;
  - контракт данных;
  - контракт сообщений.

## 5. **WCF**: контракт службы

```
[ServiceContract]
public interface ISmwService
{

    [OperationContract]
    int Add(int x, int y);

    [OperationContract]
    int Sub(int x, int y);

    [OperationContract]
    int Mul(int x, int y);
}
```

## 6. **WCF**: контракт данных – указывает каким образом данные будут сериализованы.

```
[DataContract]
public class Person
{
    string name      = String.Empty;
    string surname   = String.Empty;
    int age          = 0;

    [DataMember]
    public string Name
    {
        get { return this.name; }
        set { this.name = value; }
    }

    [DataMember]
    public string Surname
    {
        get { return this.surname; }
        set { this.surname = value; }
    }

    public int Age
    {
        get { return this.age; }
        set { this.age = value; }
    }
}
```

## 7. **WCF**: контракт сообщений

```
[MessageContract]
public class Addenda
{
    [MessageHeader]
    public int PersonId {get; set;}

    [MessageBodyMember]
    public string Description {get; set;}
}
```

8. **WCF**: реализация сервиса

```
public class Service1 : ISmwsService
{

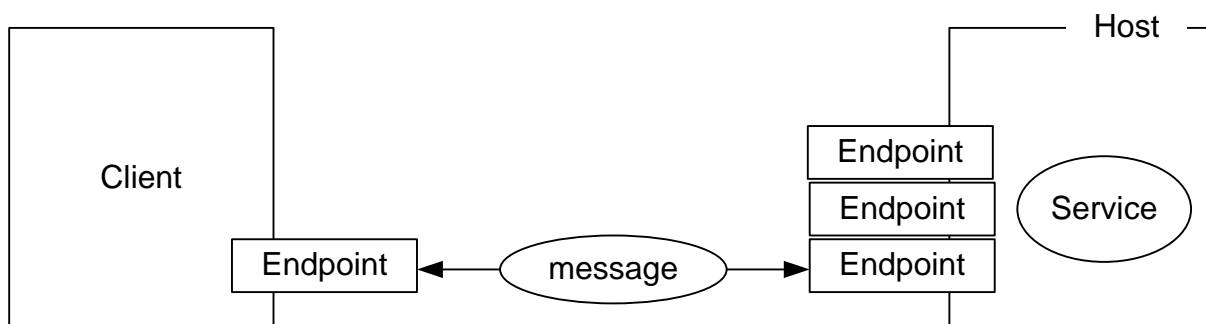
    public int Add(int x, int y) {return (x + y);}

    public int Sub(int x, int y) {return (x - y);}

    public int Mul(int x, int y) {return (x*y);}

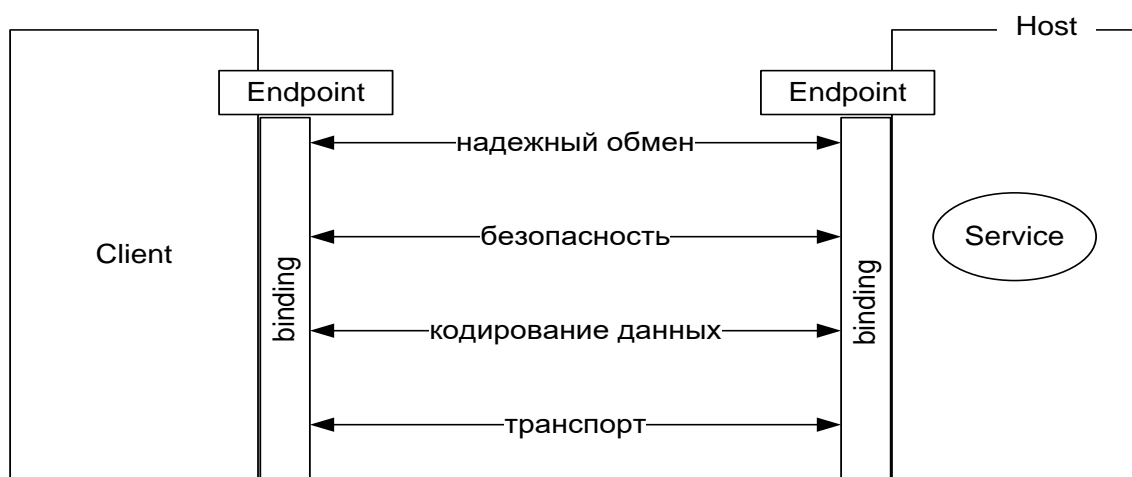
}
```

9. **WCF**: хост – контейнер для сервиса (любое С#-приложение).  
Основное назначение хоста: присоединить WCF-инфраструктуру,  
создать список конечных точек.



10. **WCF**: конечная точка – адрес, привязка, контракт.

11. **WCF**: привязка – инкапсулирует все технологии, позволяющие переправить сообщение от одной конечной точке к другой (транспорт, кодировка, протоколы (безопасность, сессия), правила обмена сообщениями (размер, timeout)).



12. **WCF**: привязка: безопасность: WS-ReliableMessaging;
13. **WCF**: привязка: безопасность: WS-Security
14. **WCF**: привязка:кодирование:
  - текст (ASCII, UTF-8, UTF-16);
  - двоичная (проприетарные алгоритмы);
  - MTOM.
15. **WCF**: привязка:транспорт:
  - HTTP;
  - TCP;
  - IPC Named Pipes;
  - MSMQ;
  - Custom.
16. **WCF**: привязка: стандартные:

Binding	Protocol/Transport	Message Encoding	Security	Default Session	Transaction	Duplex
BasicHttpBinding	Http, Https	Text	None	No	-	-
WSHttpBinding	Http, Https	Text	Message	Optional	Yes	-
WSDualHttpBinding	Http, Https	Text	Message	Yes	Yes	Yes
NetTcpBinding	TCP	Binary	Transport	Optional	Yes	Yes
NetNamedPipeBinding	Named Pipe	Binary	Transport	Yes	Yes	Yes
NetMsmqBinding	MSMQ	Binary	Transport	Yes	Yes	No
WSFederationHttpBinding	Http, Https	Text	Message	Yes	Yes	No
NetPeerTcpBinding	P2P	Binary	Transport	-	-	Yes
MsmqIntegrationBinding	MSMQ	Not Supported	Transport	Yes	Yes	-

- 17.
- 18.
- 19.
- 20.
- 21.
- 22.
- 23.
- 24.
- 25.