

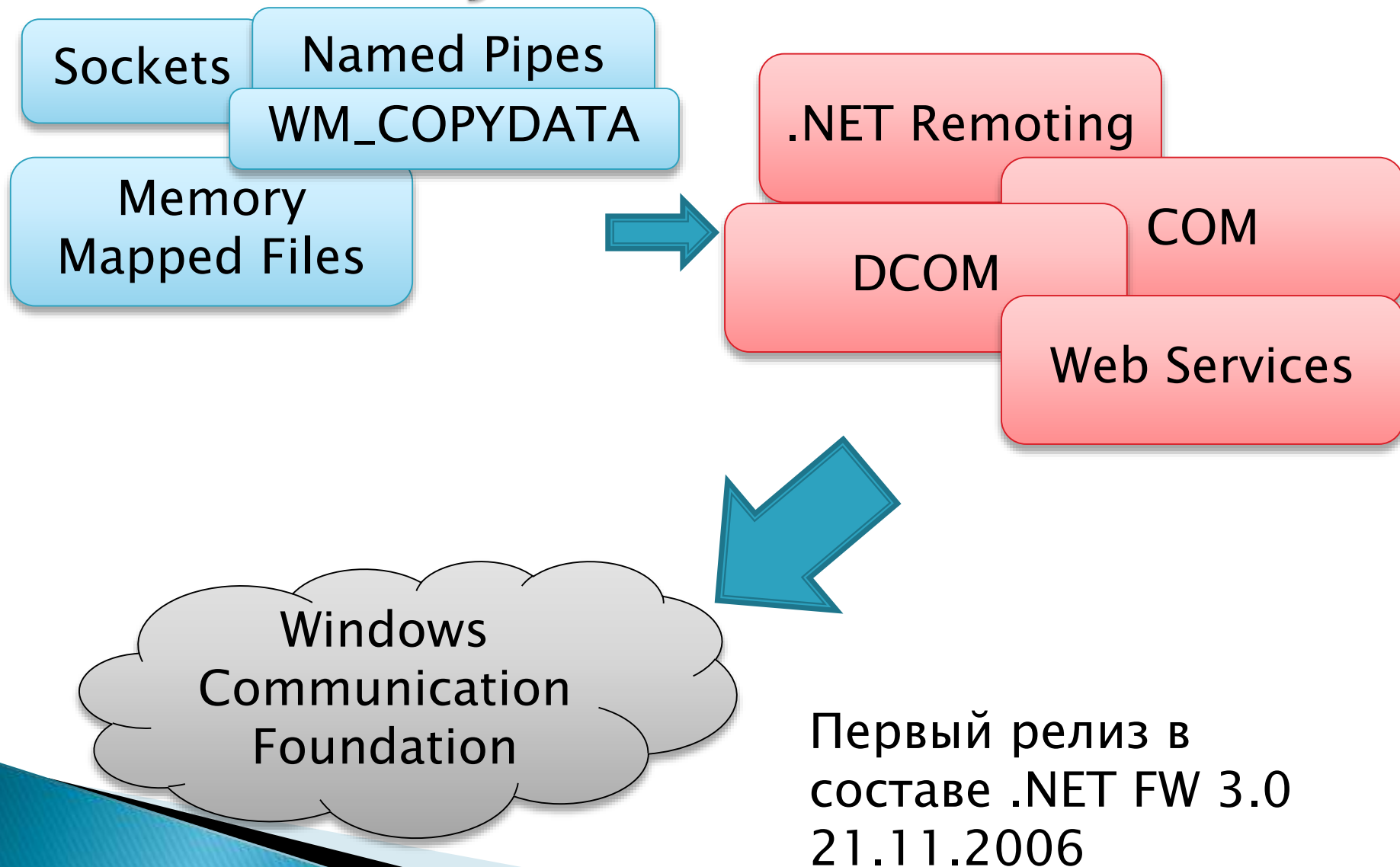
WSF. Легко или проблемно?

Игорь Лабутин, 02.04.2016

О себе

- ▶ 15 лет в индустрии
 - C/C++: Windows, Linux, QNX, Embedded
 - .NET – последние 8 лет
- ▶ Интересы
 - Сети, протоколы обмена данными
 - Проблемы производительности

WCF – откуда и зачем



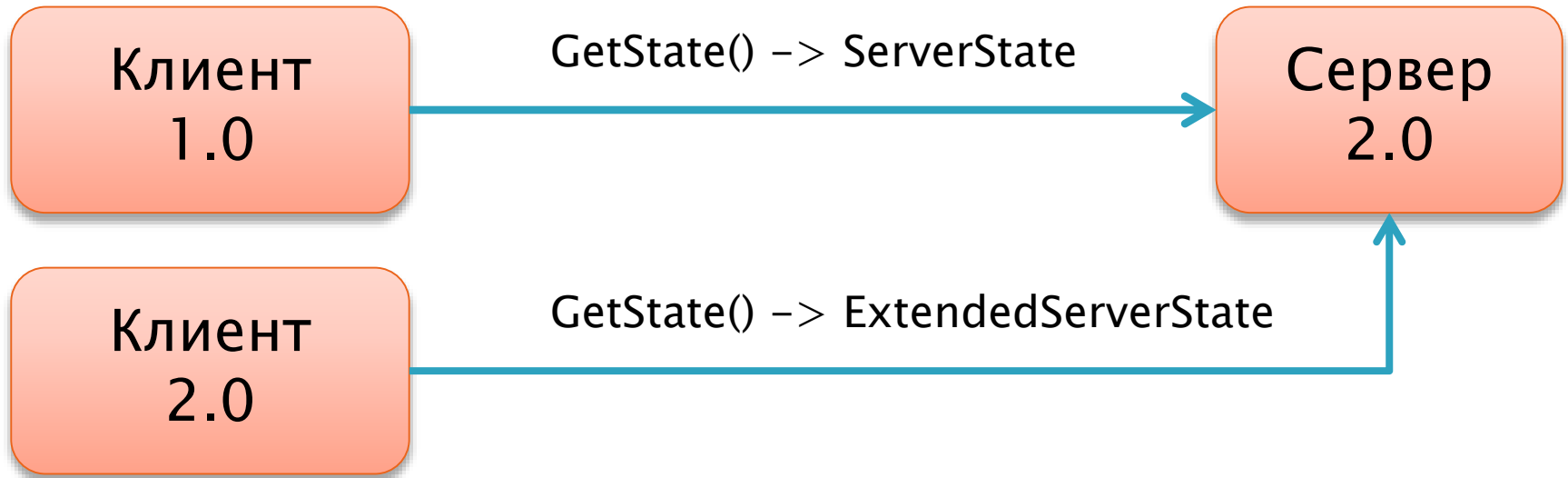
Клиент–сервер: пробуем

Address Binding Contract

Тут без Visual Studio не
обойтись!

Теперь у нас есть самое настоящее
клиент–серверное приложение!

Версионирование



☹ Это неизбежно!

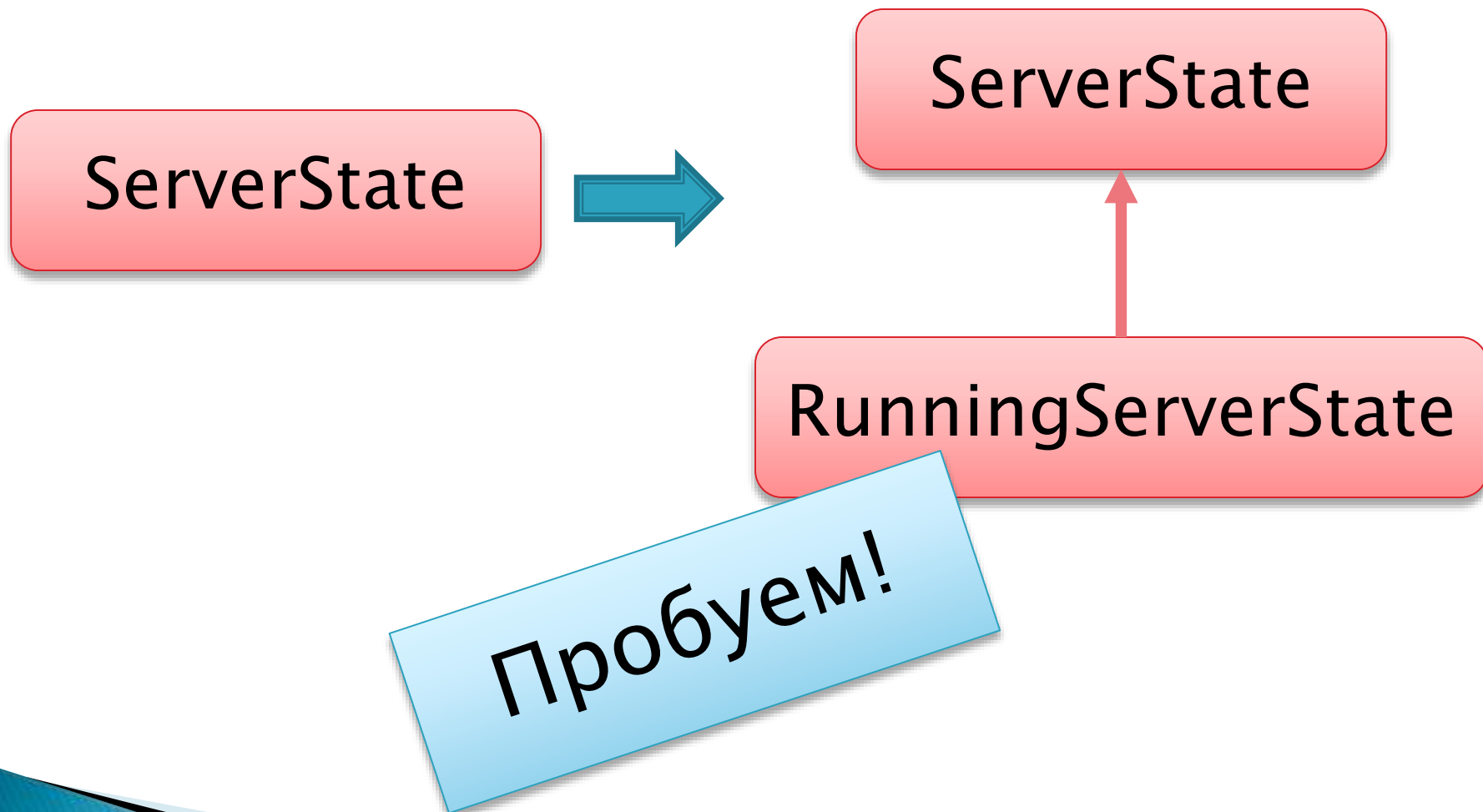
😊 В большинстве случаев
работает без проблем

Но не всегда!
Пробуем!

Версионирование: выводы

- ▶ Enums: Либо backing field и ручная обработка неожиданных значений на клиенте, либо сервер должен следить что кому посылается
- ▶ Можно, но осторожно 😊
- ▶ MS Best Practices: Data Contract Versioning
 - <http://bit.ly/1MDvOGm>

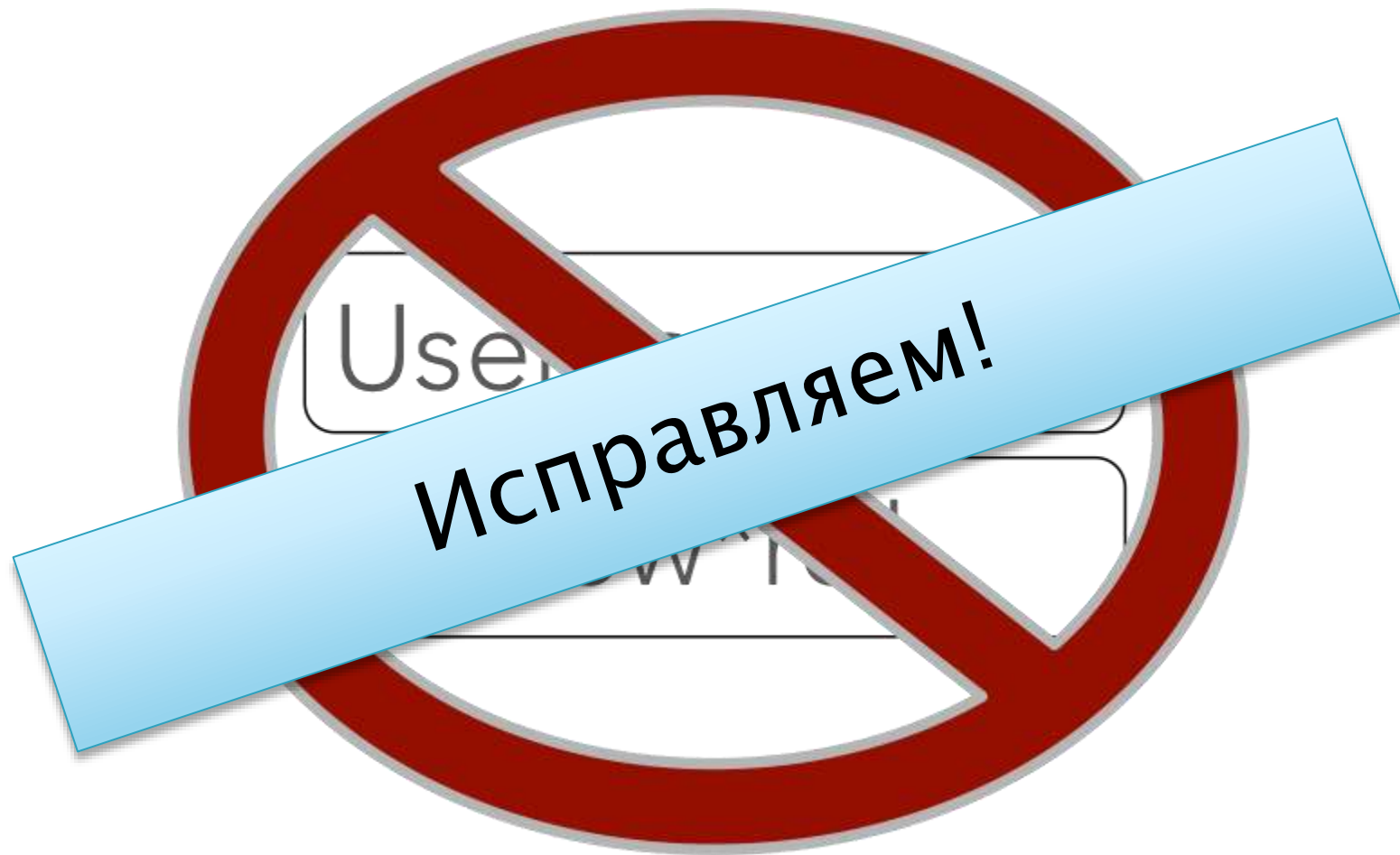
Один класс это скучно. Наследуем



Наследование: выводы

- ▶ Предпочтение атрибуту `KnownType`
- ▶ Но если нужна гибкость и код клиента свой – можно использовать свой `resolver`.

Сервер без авторизации?



Авторизация: результаты ☹

- ▶ 😊 Теперь есть авторизация
- ▶ ☹ Большой расход памяти

	Data Transfer	Memory Traffic
ServiceProxy ctor	20 Mб	420 Mб

Надо что-то
делать!

Пробуем исправить

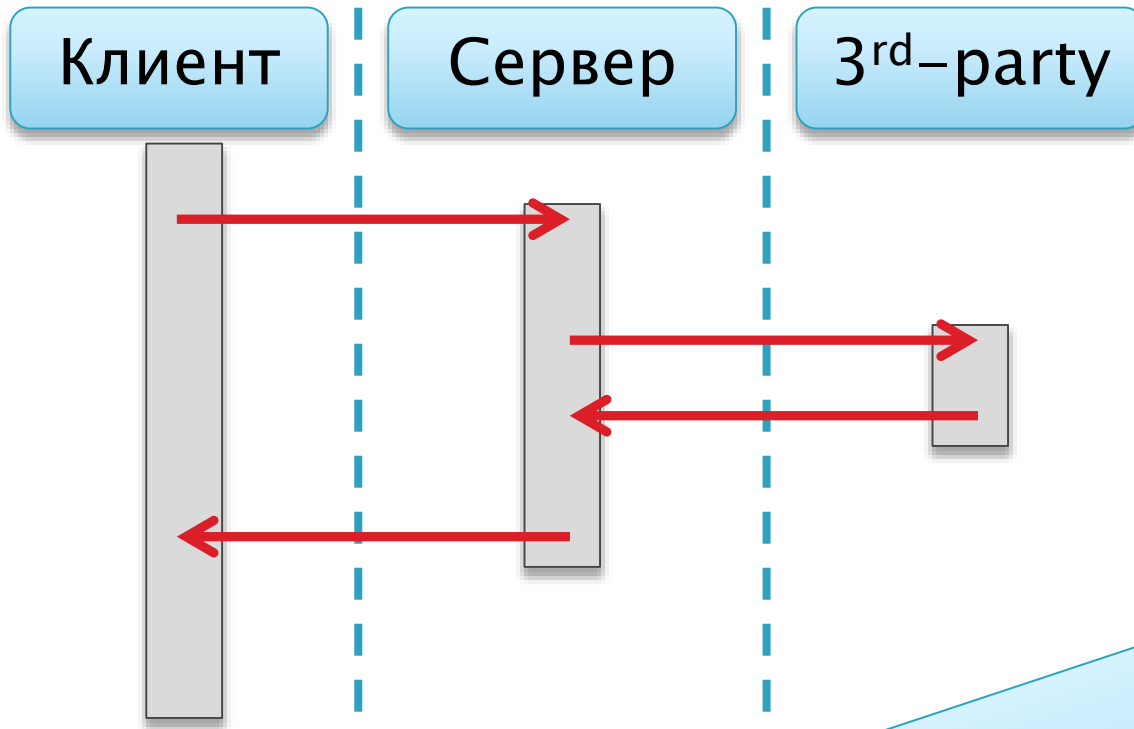
Авторизация: результаты 😐

- ▶ 😊 Авторизация все еще работает
- ▶ 😐 Расход памяти уменьшился

	Data Transfer	Memory Traffic
ServiceProxy(string)	20 Mб	420 Mб
Behavior extension	20 Mб	300 Mб

Есть куда
улучшать

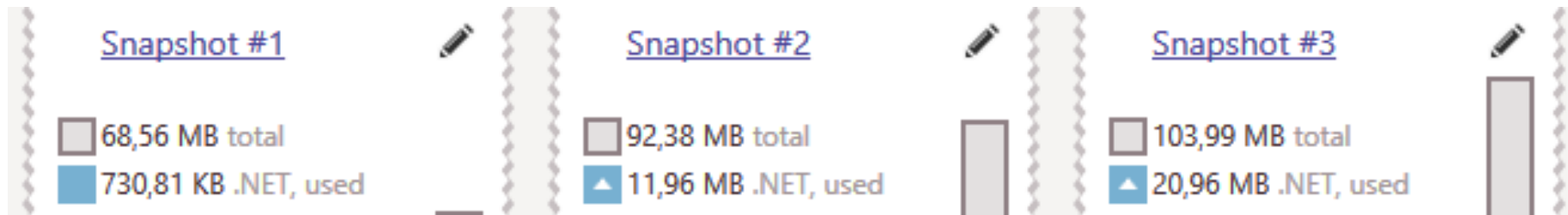
Цепочки вызовов



Делаем!

Цепочки вызовов: выводы

- ▶ WCF – это просто кладезь проблем
 - Смотрим расход памяти



- ▶ Причины?
 - ▶ Наши `InternalProxy` остались в памяти?
 - ▶ Внутренние структуры связанные с `IInternalContract` не освободились?

Куда делась память?

Here you can view how many objects of a certain type present in the object set. ⓘ

Filter:

Type	Objects count	Byte

Filter:

Type	Objects count	Byte

Может, дело в отсутствии `using` ?

Клиент без using

Snapshot #1

60,37 MB total

123,44 KB .NET, used

Snapshot #2

90,64 MB total


757,37 KB .NET, used

Snapshot #3







89,42 MB total

757,27 KB .NET, used

Filter:

Type	Objects
 System.ServiceModel.Channels.ServiceChannelProxy+SingleRetu	1

Filter:

Type	Objects
 System.Runtime.InteropServices.HandleRef	8
 System.Runtime.InteropServices.ComVisibleAttribute	10
 System.ServiceModel.ChannelFactoryRefCache<IService>	1
 System.Runtime.InteropServices.GuidAttribute	1
 System.Collections.Generic.Dictionary<EndpointTrait<IService>,	1
 System.Collections.Generic.LinkedList<EndpointTrait<IService>>	1

Куда делась память?

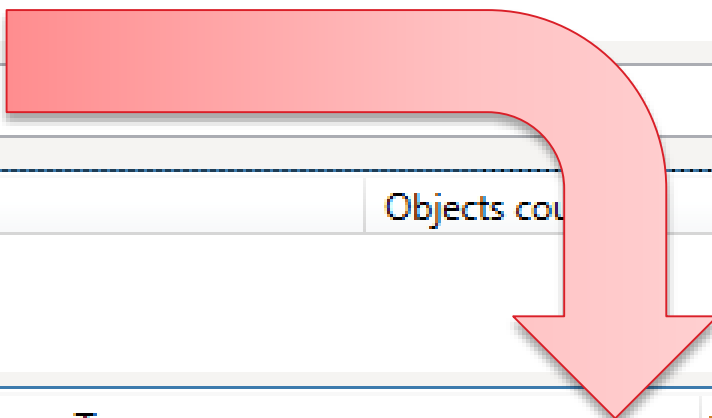
Here you can view how many objects of a certain type present in the object set. 






Filter:

Type	Objects count	Bytes
------	---------------	-------

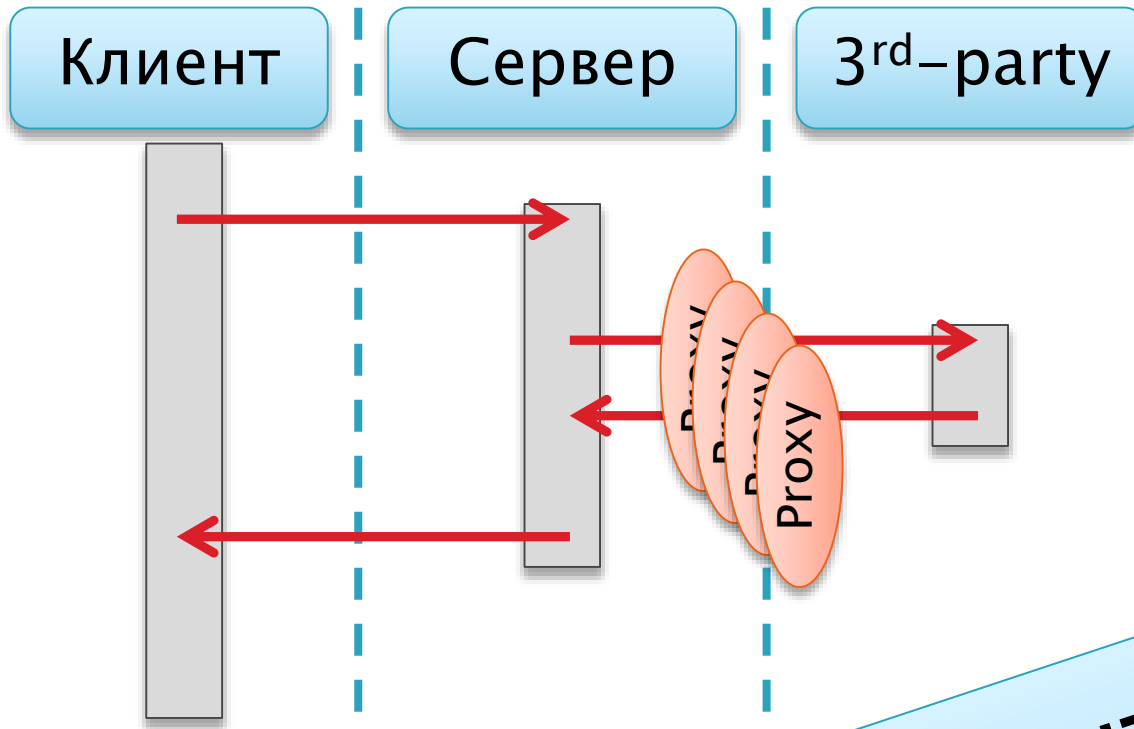
Filter:

Type	Objects count	Bytes
------	---------------	-------



Type	Objects count	Bytes
  System.ServiceModel.InstanceContext	1 600	2 882 400
484  System.Byte[]	484	2 652 976
 160  System.ServiceModel.Channels.BinaryMessa	163	85 412

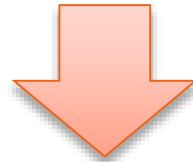
Цепочки вызовов: подробности



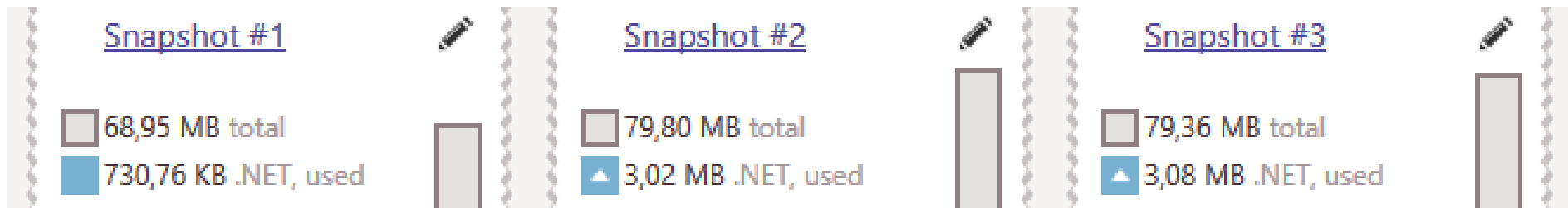
Надо что-то
решать!

Цепочки вызовов: ИТОГ

```
InternalProxy p = new InternalProxy();  
p.DoSomething();
```



```
using (InternalProxy p = new InternalProxy())  
{  
    p.DoSomething();  
}
```



Использование **using** всегда полезно, даже если кажется что освободить нечего.

Расширяемость WCF

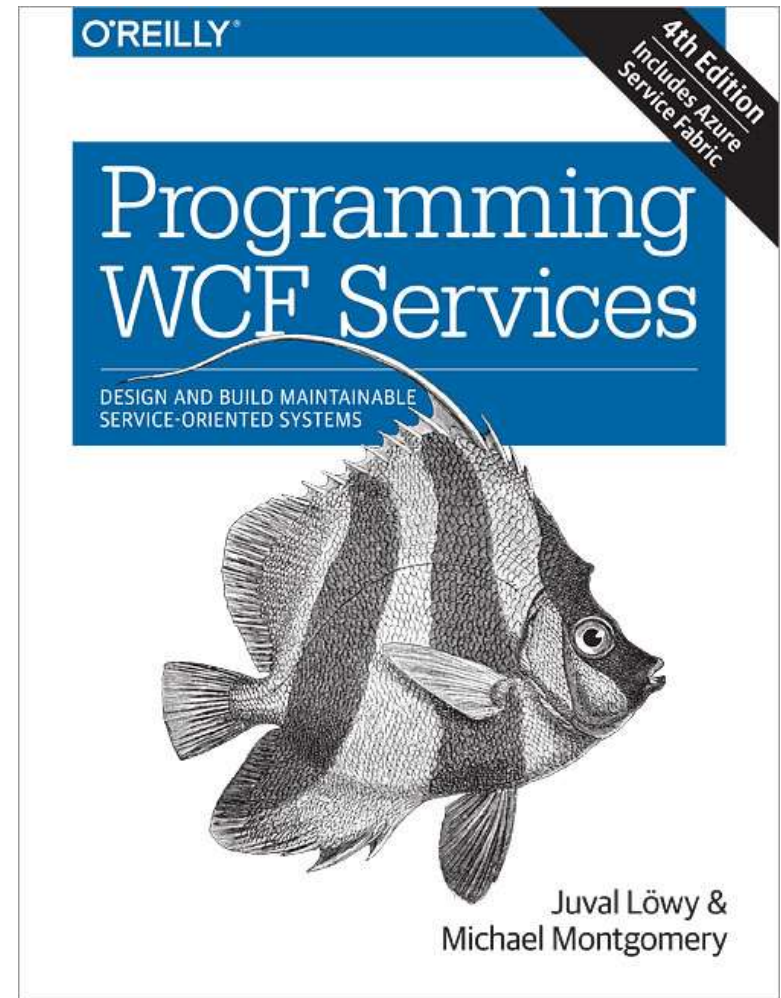
- ▶ Простые вещи требуют минимального кода
- ▶ Сложные вещи сделать можно!
 - Более 30 точек расширения
 - В большинстве случаев требуют много кода и понимания фабрик
 - Активно используются атрибуты

Заключение

- ▶ WCF – это удобно и в большинстве случаев просто
- ▶ Если что-то необходимо сделать – скорее всего найдется способ (и не один!)

Материалы

- ▶ Series of posts by Carlos Figueira at <http://bit.ly/wcfextensibility>
- ▶ MSDN ☺



Спасибо за внимание!



E-mail: ilabutin@gmail.com