

Введение в
Managed Extensibility Framework

MEF

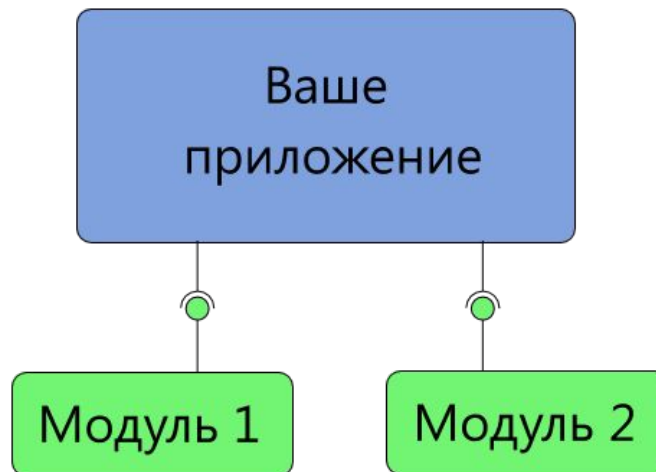
Владимир Лисник

Software Engineer

M. C. Dean

Что такое MEF?

Инструмент позволяющий решать задачи
плагинообразности приложения



Функциональные части



Импорт

// Импорт одного расширения:

```
[Import(typeof(IPlugin))]  
public IPlugin Plugin { get; set; }
```

контракт

тип расширения

точка импорта

// Импорт всех доступных расширений:

```
[ImportMany(typeof(IPlugin))]  
public IEnumerable<IPlugin> Plugins { get; set; }
```

контракт

тип расширения

точка импорта

Экспорт

// Экспорт расширения:

```
[Export(typeof(IPlugin))]  
public class SamplePlugin : IPlugin
```

контракт

экспортируемая
часть

тип расширения

Композиция

// Явная передача элементов композиции:

```
var container = new CompositionContainer();  
container.ComposeParts(part, new SamplePlugin());
```



CompositionPart



CompositionPart

Композиция

// Использование каталога для композиции:

```
var pluginsCatalog = new DirectoryCatalog(Path.GetFullPath("Plugins"));  
var container = new CompositionContainer(pluginsCatalog);  
container.ComposeParts(attributedInstance);
```

объект с точками
импорта

каталог

Каталоги

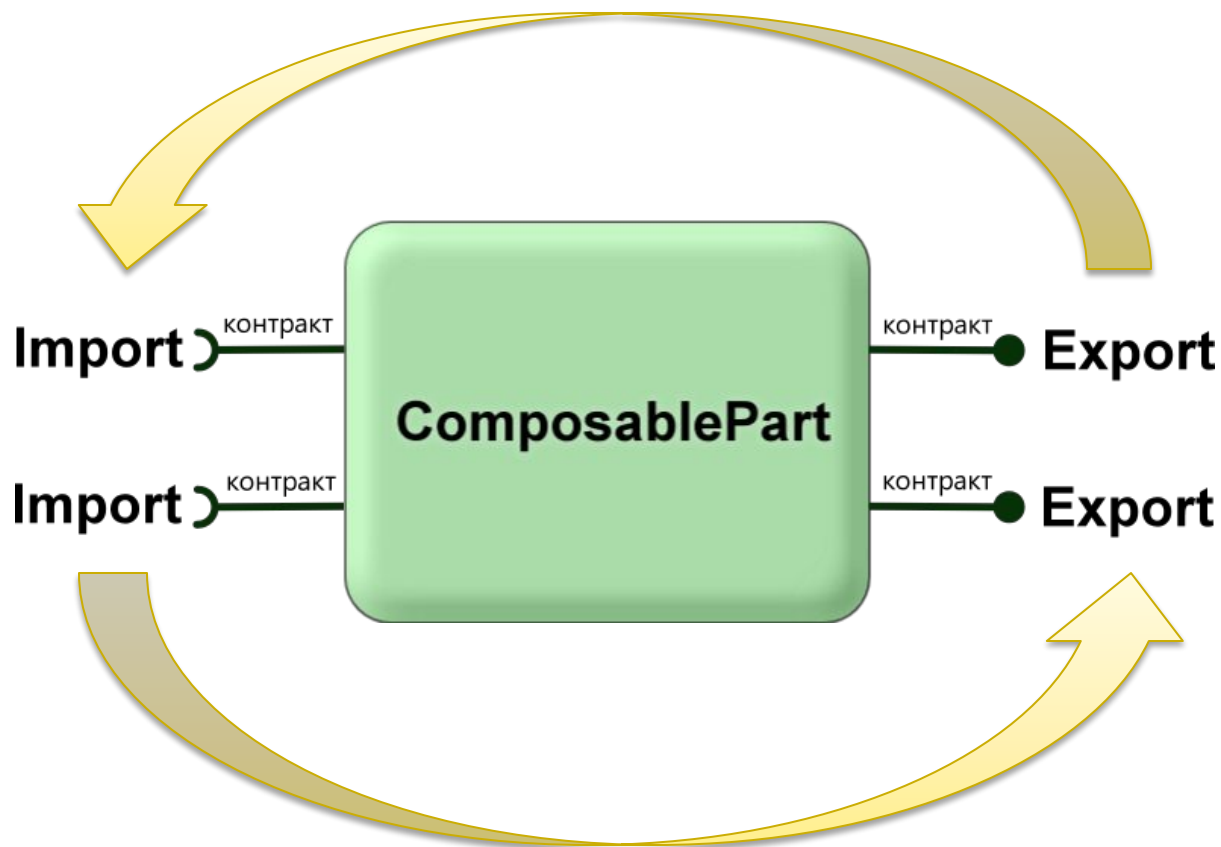
- AggregateCatalog
- AssemblyCatalog
- DirectoryCatalog *
- TypeCatalog
- DeploymentCatalog **

* Только в .NET Framework

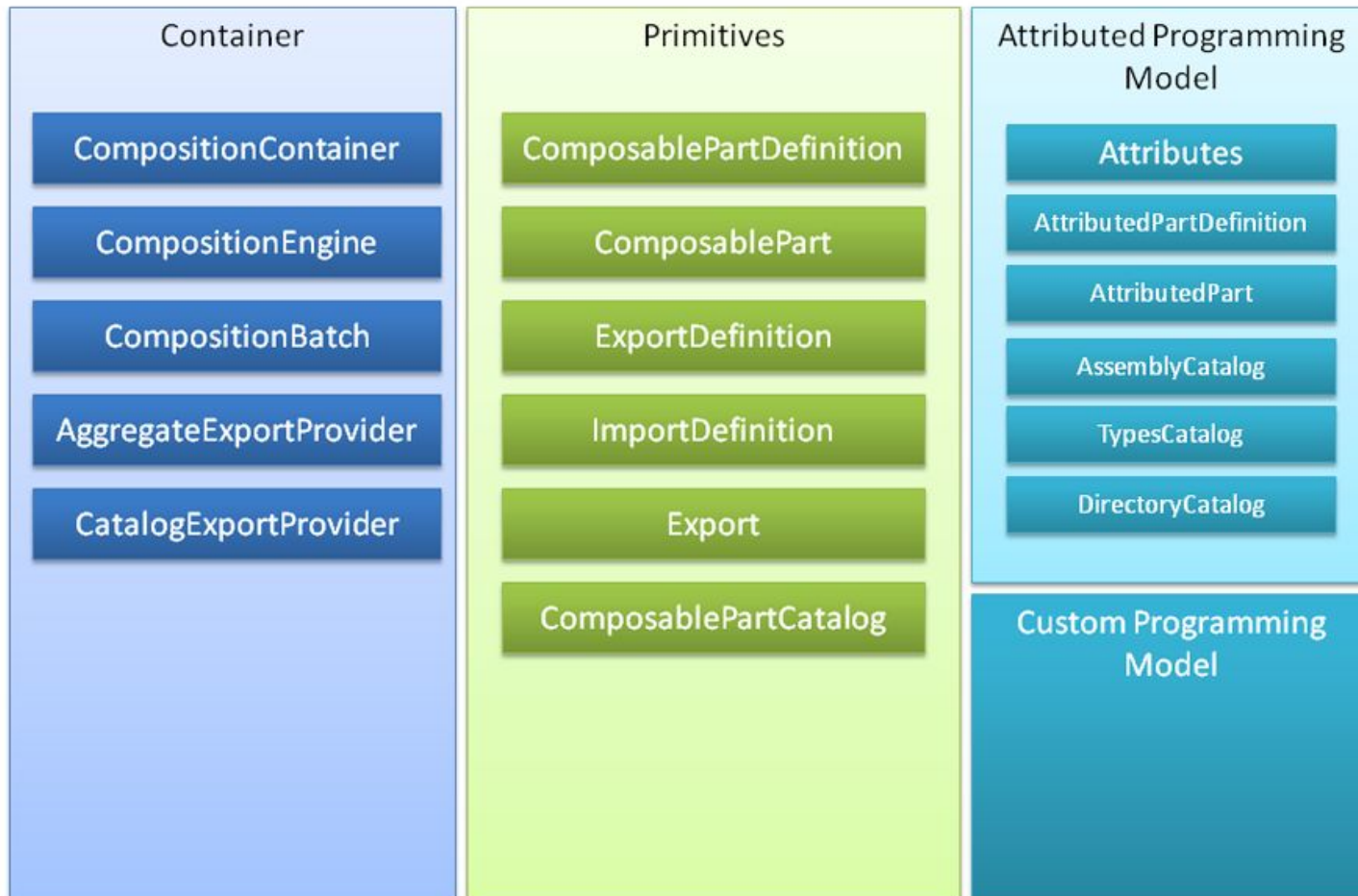
** Только в Silverlight

Обзор архитектуры

// Ключевая концепция:



Обзор архитектуры



Где обитает?

System.ComponentModel.Composition.dll

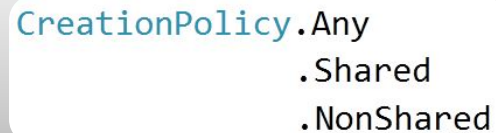
System.ComponentModel.Composition.Initialization.dll*

* Только в Silverlight

PartCreationPolicyAttribute

// Экспорт:

```
[Export(typeof(IPlugin))]  
[PartCreationPolicy(CreationPolicy.Shared)]  
public class FirstPlugin : IPlugin
```



CreationPolicy.Any
.Shared
.NonShared

// Импорт:

```
[Import(typeof(IPlugin),  
    RequiredCreationPolicy = CreationPolicy.Shared)]  
public IPlugin Plugin { get; set; }
```

Метаданные экспортов

// MetadataExportAttribute:

```
[Export(typeof(IPlugin))]  
[ExportMetadata("name", "plugin1")]  
public class FirstPlugin : IPlugin
```

// Пользовательские атрибуты :

1. `public interface IPluginCategoryMetadata`

тип метаданных
(все свойства должны
быть **read-only**)

2. `[MetadataAttribute]
[AttributeUsage(AttributeTargets.Class, AllowMultiple = false)]
public class PluginExport : ExportAttribute`

наш атрибут
(метаданные и экспорт)

3. `[PluginExport(Category = PluginCategory.Adv)]
public class SecondPlugin : IPlugin`

Доступ к метаданным



Lazy<T, TMetadata>

// Точка импорта:

```
[Import(typeof(IPlugin))]  
public Lazy<IPlugin, IPluginCategoryMetadata> Plugin { get; set; }
```

// Работа с метаданными:

```
if (Plugin.Metadata.Category == PluginCategory.None)  
{  
    name = Plugin.Value.Name;  
}
```

Рекомпозиция

по умолчанию = **false**

```
[ImportMany(typeof(IPlugin), AllowRecomposition = true)]  
public IEnumerable<IPlugin> Plugins
```

Могут производить рекомпозицию:

- AggregateCatalog (автоматически)
- DirectoryCatalog (вызвать Refresh)
- DeploymentCatalog (автоматически)

Доступность

- .NET Framework
 - Как отдельная библиотека в .NET 3.5
 - Поставляется в составе .NET 4.0
 - Доступен в .NET Framework 4 Client Profile
- Silverlight
 - Как отдельная библиотека в Silverlight 3
 - Поставляется в составе Silverlight 4
- Silverlight for Windows Phone
 - Не доступно

Лицензия

MS Public License

(Проект с открытым исходным кодом)

Ресурсы

- <http://mef.codeplex.com/>
- <http://blogs.msdn.com/b/gblock/archive/tags/mef/>
- <http://blogs.msdn.com/b/nblumhardt/archive/tags/mef/>

Спасибо!