

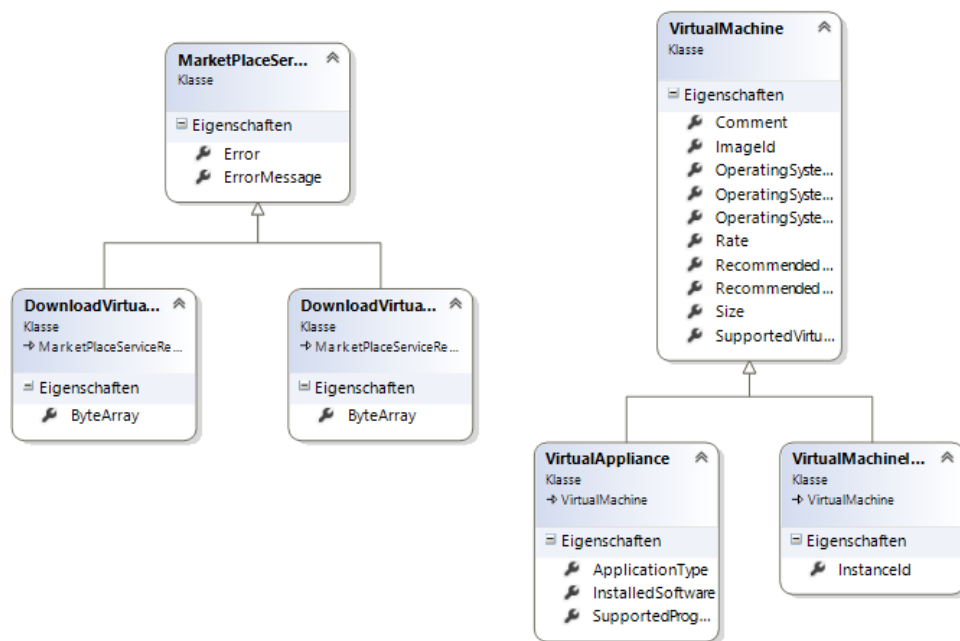
[DOKUMENTATION – ENABLING TECHNOLOGIES]

Patrick Felber Bsc
Hannes Knopf Bsc
Aleksandar Popovic Bsc

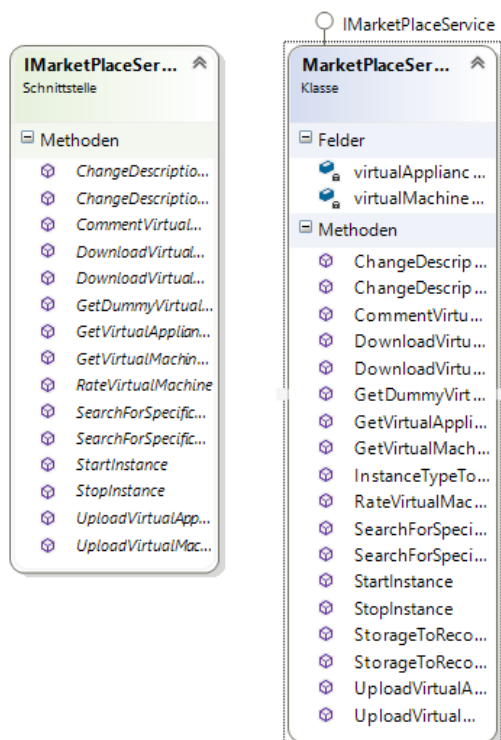
Inhaltsverzeichnis

Klassendiagramm Datenmodell (DataContracts)	2
Klassendiagramm Service	2
Beschreibung Virtuelle Maschine	3
Beschreibung Virtual Appliance	4
Interface Beschreibung - Client Side:	4
Instanzen	4
Virtual Machines	4
Virtual Appliance	5
Soa	6
REST	6

Klassendiagramm Datenmodell (DataContract)



Klassendiagramm Service



Beschreibung Virtuelle Maschine

```
[DataContract]
public class VirtualMachine
{
    /// <summary>
    /// Gets or sets the operating system type.
    /// </summary>
    [DataMember]
    public string OperatingSystemType { get; set; }

    /// <summary>
    /// Gets or sets the operating system name.
    /// </summary>
    [DataMember]
    public string OperatingSystemName { get; set; }

    /// <summary>
    /// Gets or sets the operating system version.
    /// </summary>
    [DataMember]
    public string OperatingSystemVersion { get; set; }

    /// <summary>
    /// Gets or sets the size.
    /// </summary>
    [DataMember]
    public int Size { get; set; }

    /// <summary>
    /// Gets or sets the supported virtualization platforms.
    /// </summary>
    [DataMember]
    public string SupportedVirtualizationPlatforms { get; set; }

    /// <summary>
    /// Gets or sets the recommended.
    /// </summary>
    [DataMember]
    public int RecommendedCpu { get; set; }

    /// <summary>
    /// Gets or sets the recommended memory.
    /// </summary>
    [DataMember]
    public double RecommendedMemory { get; set; }

    /// <summary>
    /// Gets or sets the rate.
    /// </summary>
    [DataMember]
    public byte Rate { get; set; }

    /// <summary>
    /// Gets or sets the comment.
    /// </summary>
    [DataMember]
    public string Comment { get; set; }

    /// <summary>
    /// Gets or sets the image id.
    /// </summary>
    [DataMember]
    public string ImageId { get; set; }
}
```

Beschreibung Virtual Appliance

```
[DataContract]
public class VirtualAppliance : VirtualMachine
{
    /// <summary>
    /// Gets or sets the application type.
    /// </summary>
    [DataMember]
    public int ApplicationType { get; set; }

    /// <summary>
    /// Gets or sets the installed software.
    /// </summary>
    [DataMember]
    public int InstalledSoftware { get; set; }

    /// <summary>
    /// Gets or sets the supported programming languages.
    /// </summary>
    [DataMember]
    public int SupportedProgrammingLanguages { get; set; }
}
```

Interface Beschreibung - Client Side:

Anlegen der Serviceverbindung:

```
using MarketplaceService;

private MarketplaceServiceClient service;

service = new MarketplaceServiceClient();
```

Instanzen

Starten einer Instanz:

```
service.StartInstance(instance);
```

Stoppen einer Instanz:

```
service.StopInstance(instance);
```

Virtual Machines

Holen aller Virtual Machine:

```
VirtualMachine[] vm = service.GetDummyVirtualMachines();
```

Ändern einer Virtual Machine:

```
MarketPlaceServiceResponse resp =
    service.ChangeDescriptionOfVirtualMachine(machine);
```

Download einer Virtual Machine:

```
DownloadVirtualMachineResponse resp = service.DownloadVirtualMachine(machine);
```

Bewerten einer Virtual Machine:

```
MarketPlaceServiceResponse resp =  
    service.RateVirtualMachine(machine, rate);
```

Kommentieren einer Virtual Machine:

```
service.CommentVirtualMachine(machine, comment);
```

Suchen einer Virtual Machine:

```
VirtualMachine[] vms = service.SearchForSpecificVirtualMachine(machine);
```

Virtual Appliance

Holen aller Virtual Appliances:

```
VirtualAppliance[] va = service.GetVirtualAppliances();
```

Ändern einer Virtual Appliance:

```
service.ChangeDescriptionOfVirtualAppliance(appliance);
```

Download einer Virtual Appliance:

```
DownloadVirtualApplianceResponse resp =  
    service.DownloadVirtualAppliance(appliance);
```

Suchen einer Virtual Appliance:

```
VirtualAppliance[] vas = service.SearchForSpecificVirtualAppliance(appliance);
```

Soa

Vorteile

- Unabhängig von der Umgebung
- XML-Basiert

Nachteile

- Nachrichtengröße – Jeder Aufruf in Envelope Hülle
- Formalität
 - XML-Schema zur Beschreibung sehr stark spezifiziert. Client angepasste Client können die Nachrichten verarbeiten. Änderungen im Service – müssen auch Änderungen im Client gemacht werden.

REST

Vorteile

- Unabhängigkeit von der Umgebung
- REST- Client benötigen keine spezielle Funktionalität – müssen nur http können.
- Skalierbarkeit
 - stateless
- XML- Basiert

Nachteile

- Rest Service sind eher für Client-Server-Verbindungen ausgelegt und nicht für verteilte Umgebungen.

Wir haben uns für SOA entschieden weil jedes Gruppenmitglied mehr Erfahrung als mit REST Services hat.