Introducing UI and Persistence Layers

Vladimir Khorikov

@vkhorikov | www.enterprisecraftsmanship.com

In This Module

User Interface

Persistence

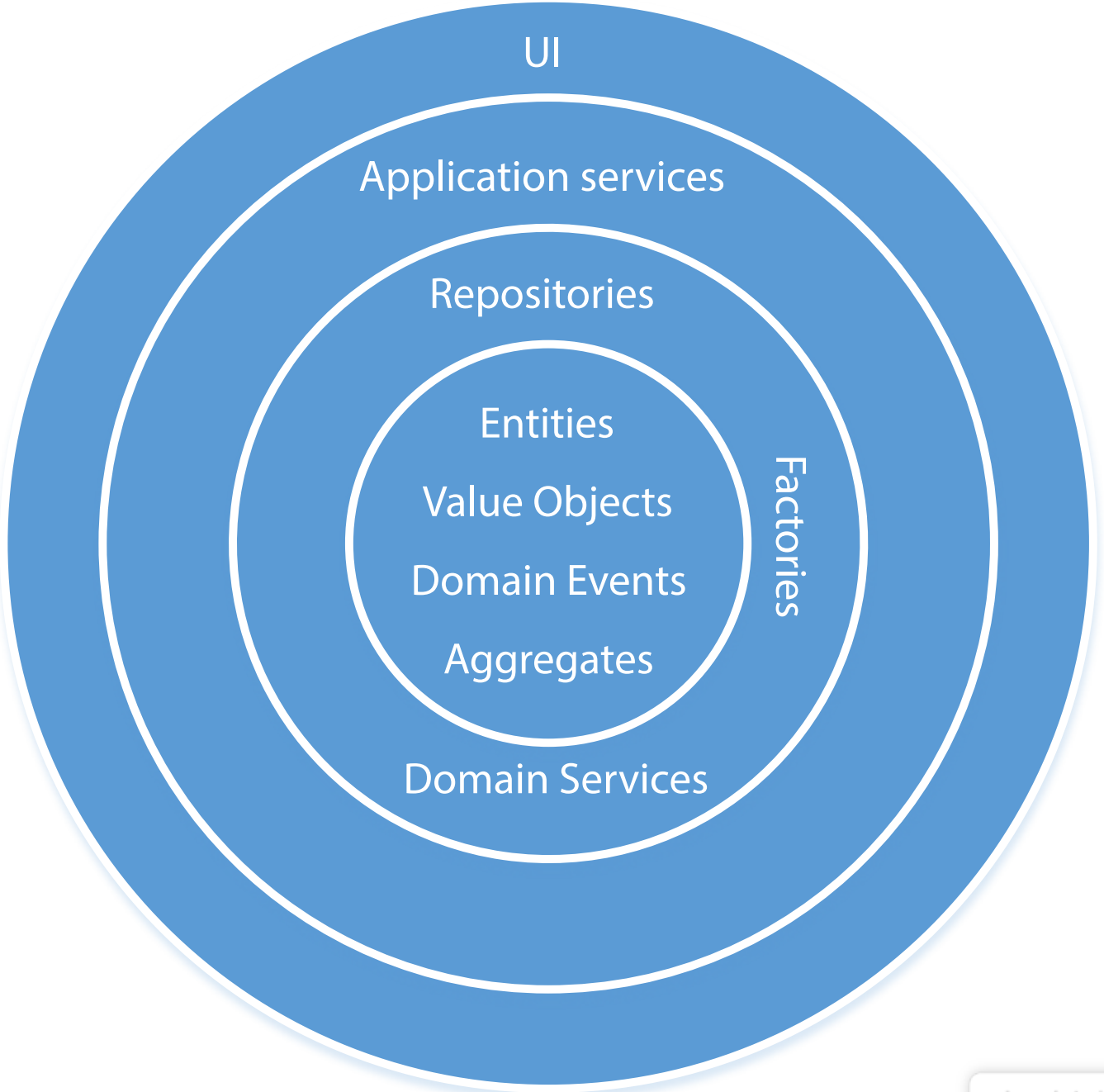
Application Services Layer

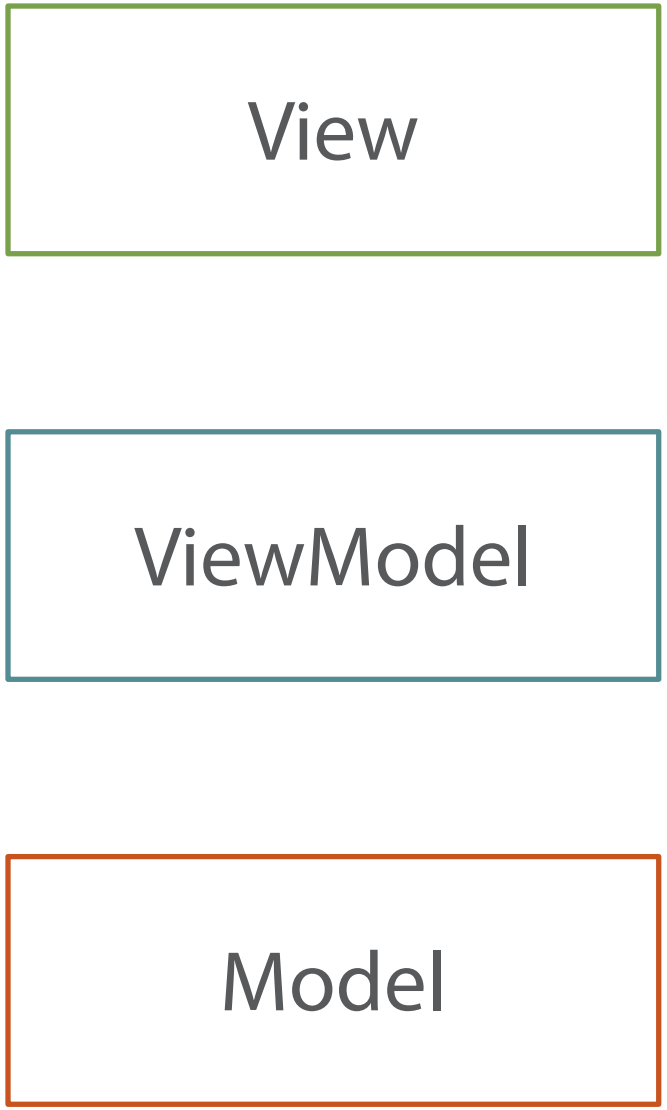
MVVM

View

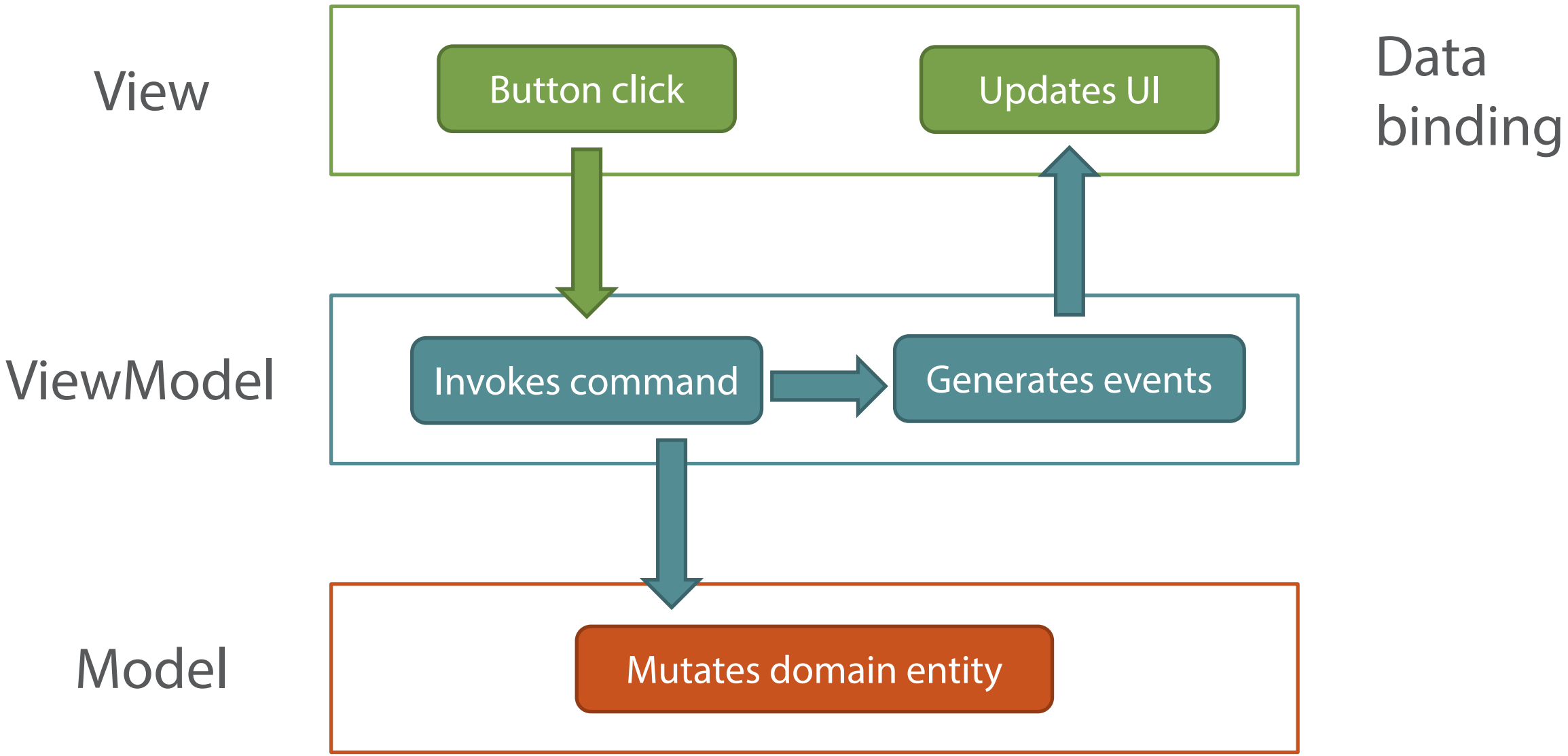
ViewModel

Model

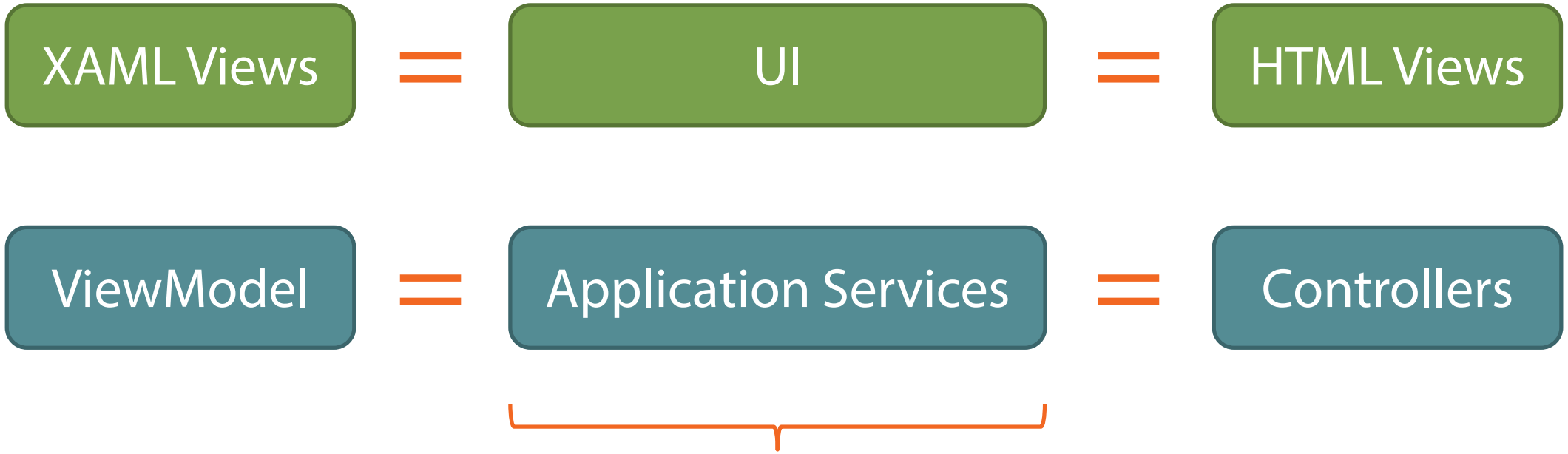




- Mediates collaboration between View and Model
- Transforms the data from Model into digestible form for View



Recap: Adding UI for the Snack Machine



Try to put logic to a proper layer of abstraction

As part of the domain layer:

```
public override string ToString()
{
    if (Amount < 1)
        return "¢" + (Amount * 100).ToString("0");

    return "$" + Amount.ToString("0.00");
}
```

As part of the application layer:

```
public Money MoneyInside =>
    _snackMachine.MoneyInside
    + _snackMachine.MoneyInTransaction;
```

Designing the Database for the Snack Machine

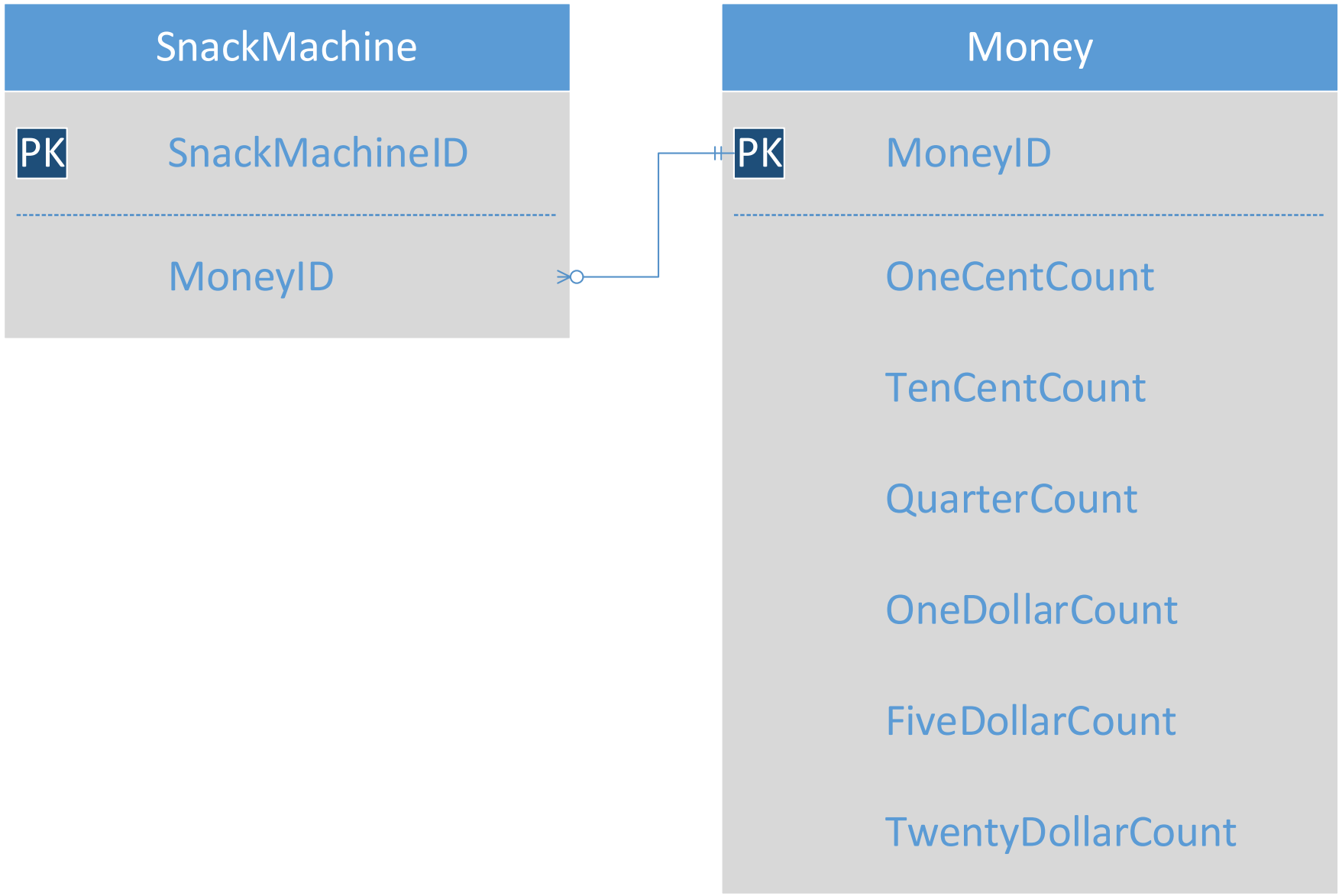
```
public class SnackMachine : Entity
{
    public Money MoneyInside;
    public Money MoneyInTransaction;
}
```

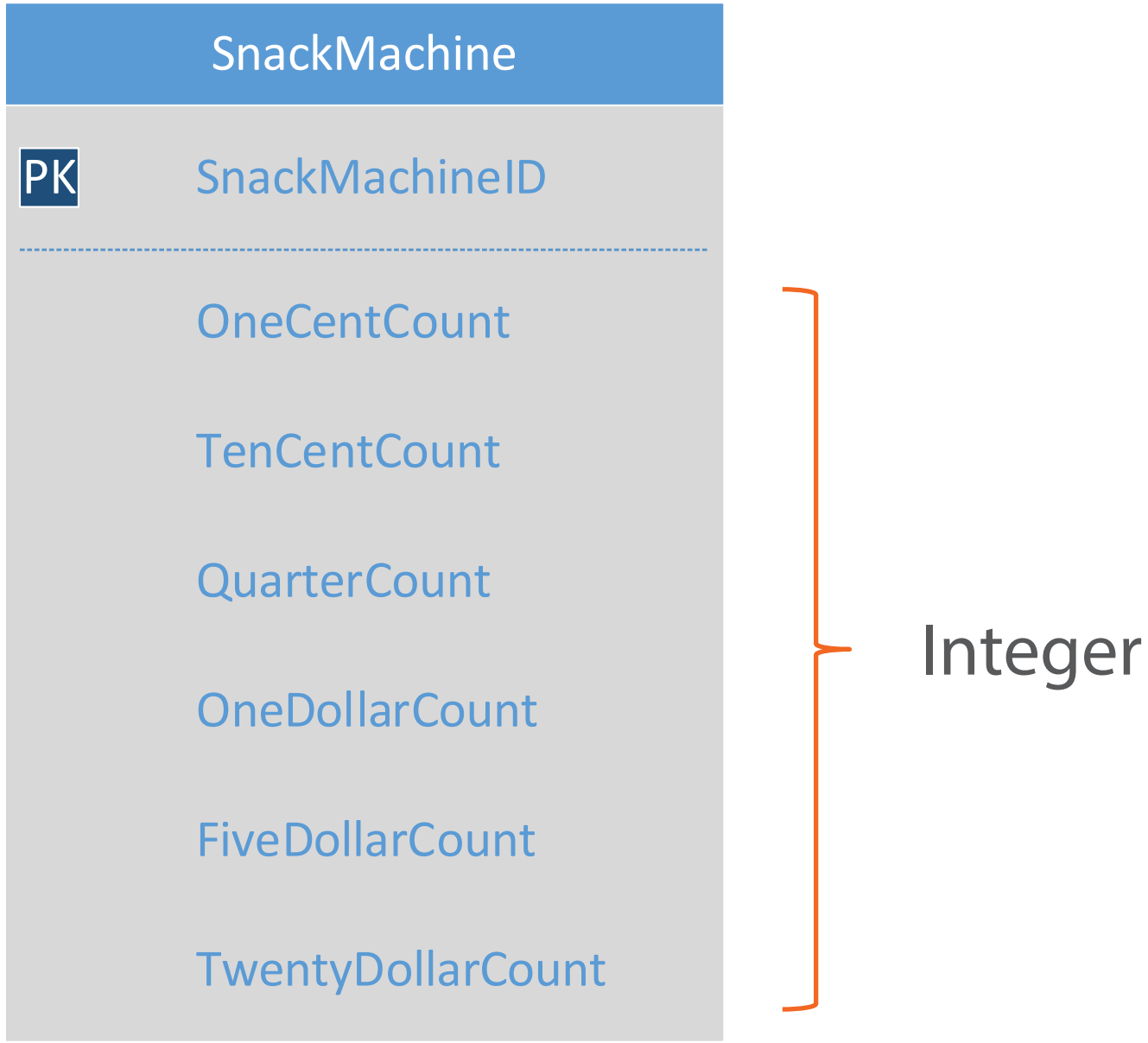
```
public class Money : ValueObject
{
    public int OneCentCount;
    public int TenCentCount;
    public int QuarterCount;
    public int OneDollarCount;
    public int FiveDollarCount;
    public int TwentyDollarCount;
}
```

Designing the Database for the Snack Machine

```
public class SnackMachine : Entity
{
    public Money MoneyInside;
}
```

```
public class Money : ValueObject
{
    public int OneCentCount;
    public int TenCentCount;
    public int QuarterCount;
    public int OneDollarCount;
    public int FiveDollarCount;
    public int TwentyDollarCount;
}
```

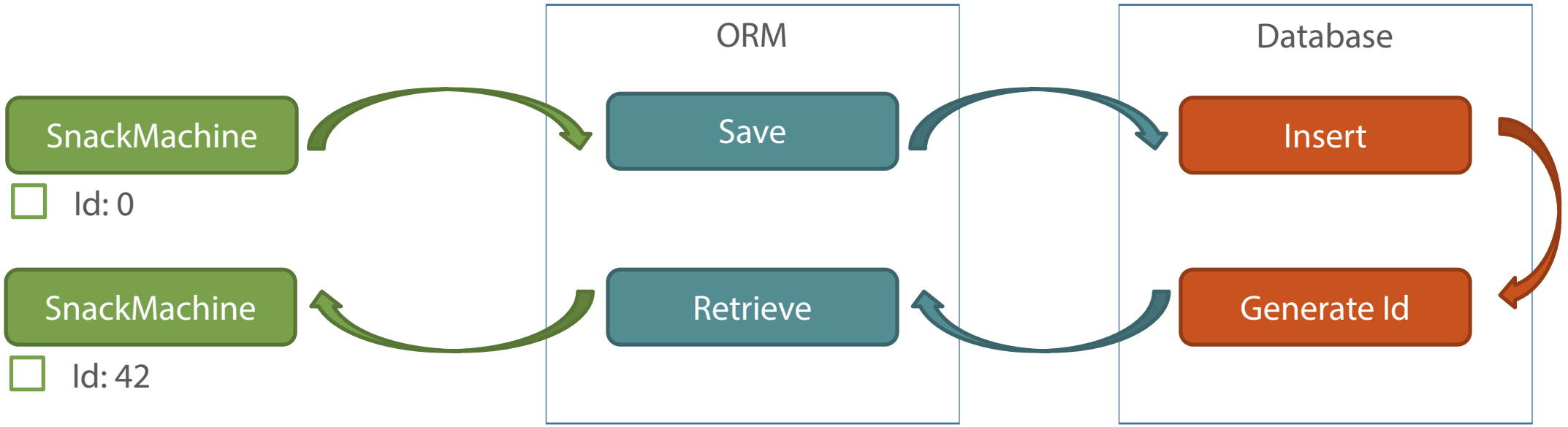




Id Generation Strategies

Identity feature:

```
CREATE TABLE dbo.SnackMachine(
    SnackMachineID bigint PRIMARY KEY IDENTITY(1,1),
    ...)
```



Id Generation Strategies

GUIDs:

```
public abstract class Entity
{
    public Guid Id { get; private set; }

    protected Entity()
    {
        Id = Guid.NewGuid();
    }
}
```

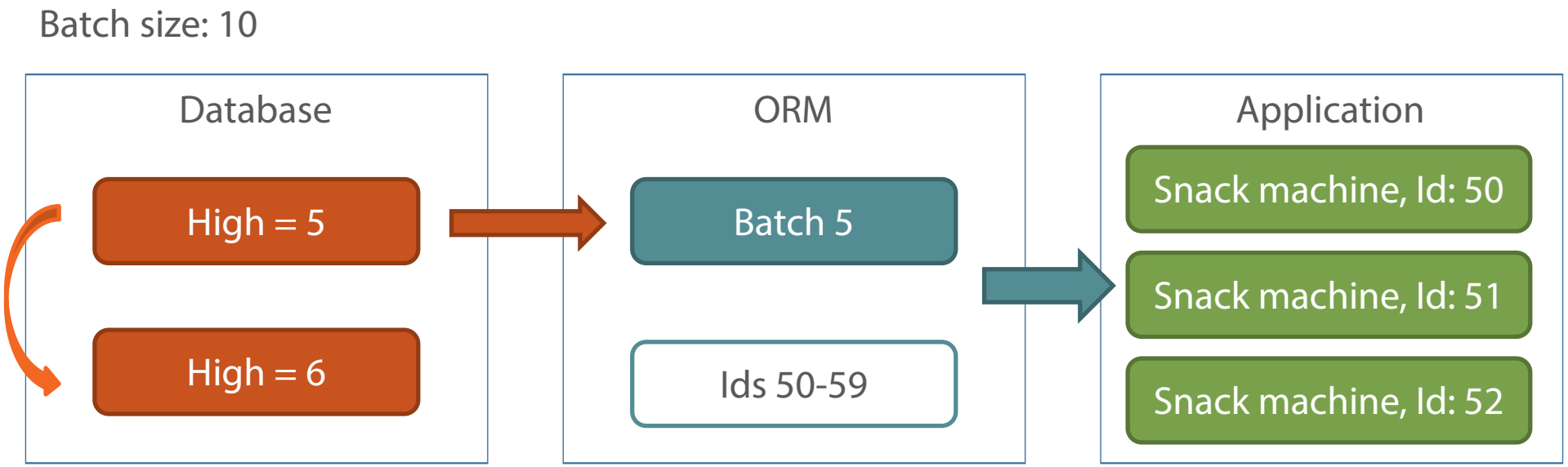
0CC287B7-1E21-4DB4-BEE6-6B53C68C6052

Id Generation Strategies

Hi/Lo:

High = Number of batch

Low = Number of Id in the batch



Id Generation Strategies

Table name: dbo.Ids

	EntityName	NextHigh
	SnackMachine	0
▶*	*NULL*	*NULL*

Id Generation Strategies

```
public class HiLoConvention : IIdConvention
{
    public void Apply(IIdentityInstance instance)
    {
        instance.Column(instance.EntityType.Name + "ID");
        instance.GeneratedBy.HiLo(
            "[dbo].[Ids]",
            "NextHigh",
            "9",
            "EntityName = '" + instance.EntityType.Name + "'");
    }
}
```

Mapping Strategies

XML Files | Attributes | Fluent mappings

Mapping Strategies

XML Files:

```
<class name="SomeEntity" table="[dbo].[SomeEntity]">
  <property name="Property1">
    <column name="Property1" not-null="true" />
  </property>
  <property name="Property2">
    <column name="Property2" not-null="true" />
  </property>
</class>
```

Mapping Strategies

Attributes

```
[Class]
public class SomeEntity
{
    [Property]
    public string Property1 { get; private set; }

    [Property]
    public string Property2 { get; private set; }
}
```

Mapping Strategies

Fluent mappings:

```
public class SomeEntityMap : ClassMap<SomeEntity>
{
    public SomeEntityMap()
    {
        Map(x => x.Property1);
        Map(x => x.Property2);
    }
}
```

Summary

- MVVM pattern in Onion Architecture
- Try to put logic to a proper layer of abstraction
- Value Objects shouldn't have their own tables
- Purity trade-offs when using an ORM
- Id generation strategies
- Mapping strategies

In the Next Module

Aggregates and repositories