

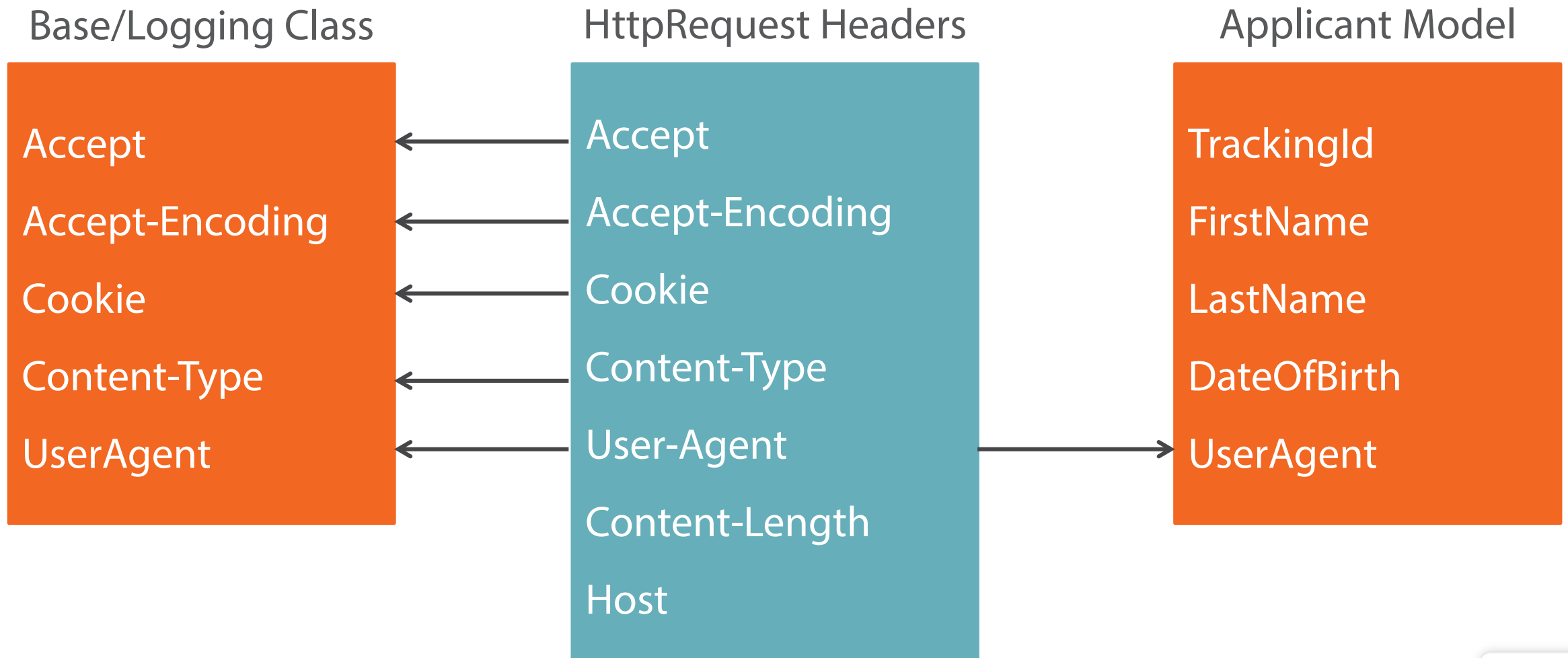
# Improving Data Availability with Custom Value Providers



Alex Wolf


[www.alexwolfthoughts.com](http://www.alexwolfthoughts.com)

# Accessing **Http Header** Information



# Model Binding and Data Availability

```
public ActionResult Index(ApplicantVM applicant)
{
    applicant.UserAgent = HttpContext.Request.UserAgent;
    _context.Applicants.Add(applicant);
    _context.SaveChanges();
}
```





# Understanding Value Providers

# Extending **Model Binding** Through **Inheritance**

Default Model Binder

CreateModel ()  
BindModel ()  
BindProperty ()

**IMo**delBinder

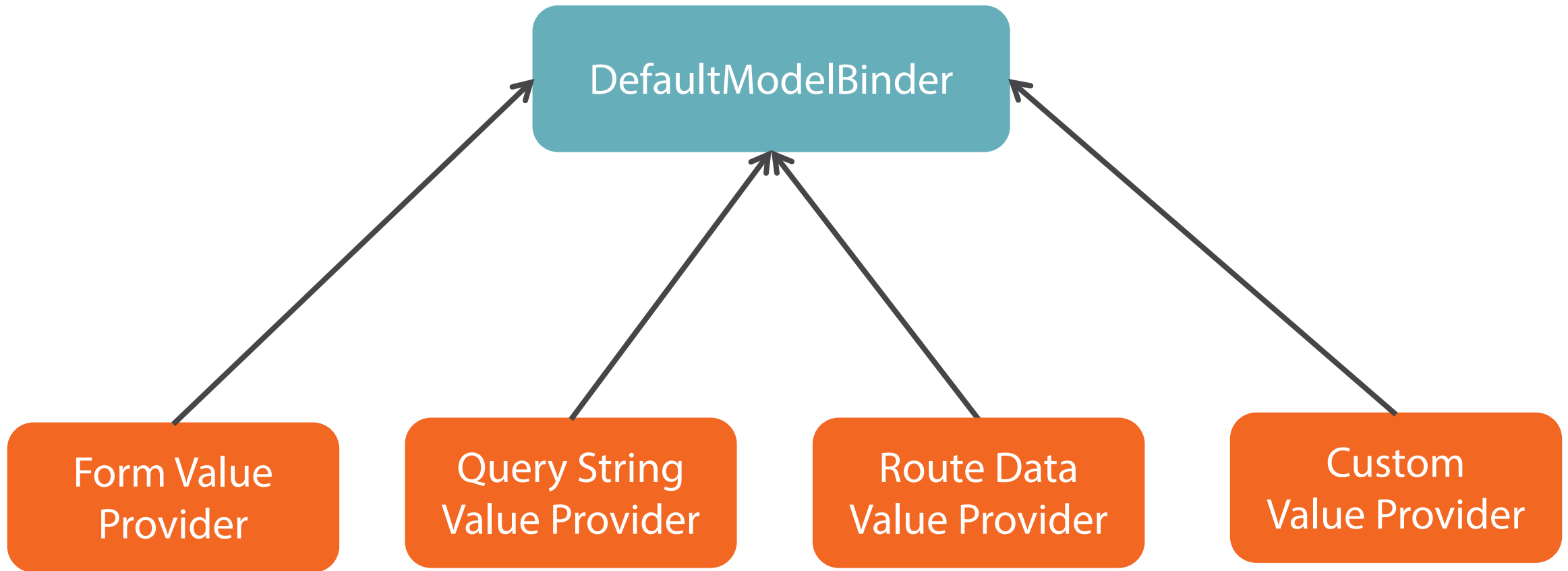
Inherited Custom Binders

Override  
CreateModel ()

Override  
BindModel ()

Override  
All Virtual Methods

# Extending Model Binding Through Value Providers



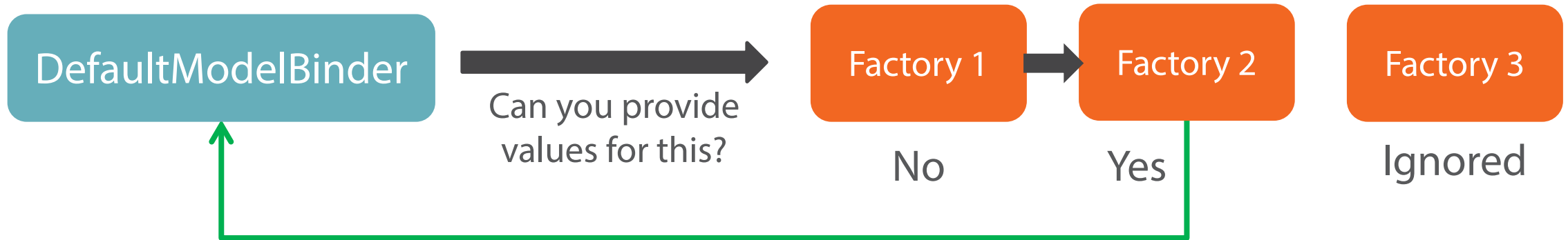
# The **IValueProvider** Interface

```
public interface IValueProvider
{
    bool ContainsPrefix(string prefix);

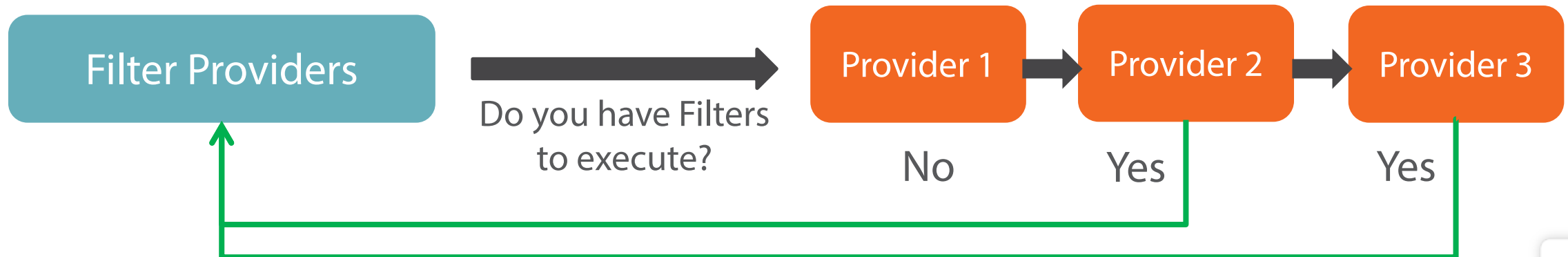
    ValueProviderResult GetValue(string key);
}
```

# Cooperative and Competitive Services

## Competitive

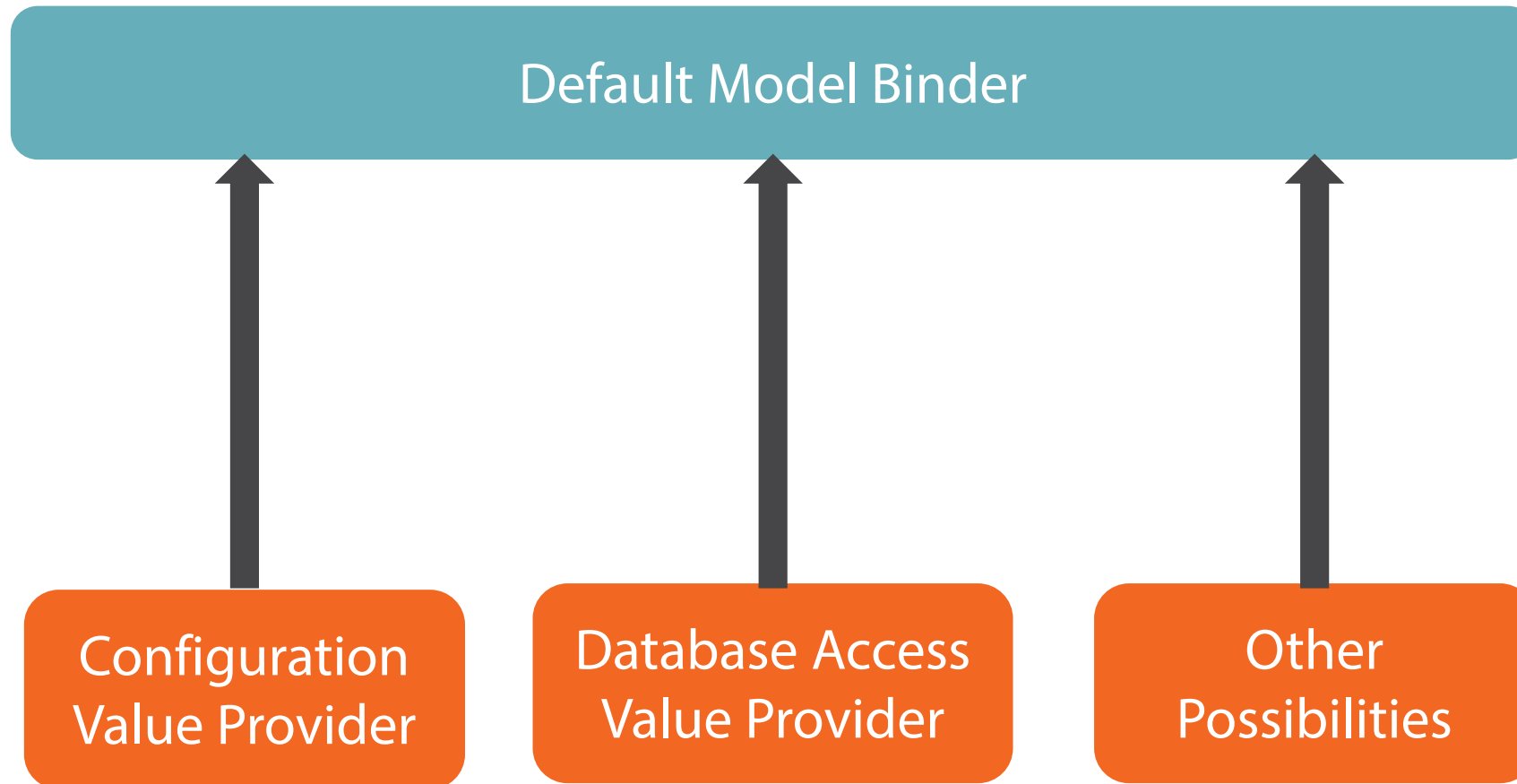


## Cooperative



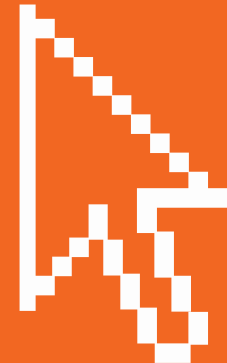


# Value Provider Possibilities



# Demo

## Creating a Custom Value Provider



# Summary



- Value Providers supply data to Model Binders
- Implemented through IValueProvider interface
- Supplied to the framework using Value Provider Factories
- Competitive service – the first Value Provider that can supply a value will be used