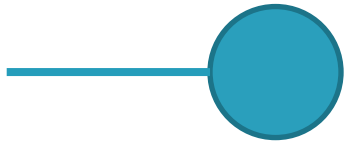
Decoupling the Domain Model from Data Contracts

Vladimir Khorikov

@vkhorikov www.enterprisecraftsmanship.com

Domain Model and Data Contracts



API

/customers

~~**/customer**~~

Breaking change

Domain Model and Data Contracts

```
[{  "name": "James Peterson",
    "email": "james.peterson@gmail.com",
    "status": "Regular",
    "statusExpirationDate": null,
    "moneySpent": 0,
    "purchasedMovies": null,
    "id": 1 }]
```

totalMoneySpent

~~**moneySpent**~~

Breaking change

Domain Model and Data Contracts

Shape of incoming and outcoming data = **Data contracts**

Domain Model and Data Contracts

Serialization of domain entities = **Coupling domain model to data contracts**

No refactoring

Domain Model and Data Contracts

```
[HttpPost]
public IActionResult Create([FromBody] Customer item)
{
    /* ... */
    _customerRepository.Add(item);
    _customerRepository.SaveChanges();
    return Ok();
    /* ... */
}
```

```
{
  "Name": "Some name",
  "Email": "some@email.com"
}
```

Domain Model and Data Contracts

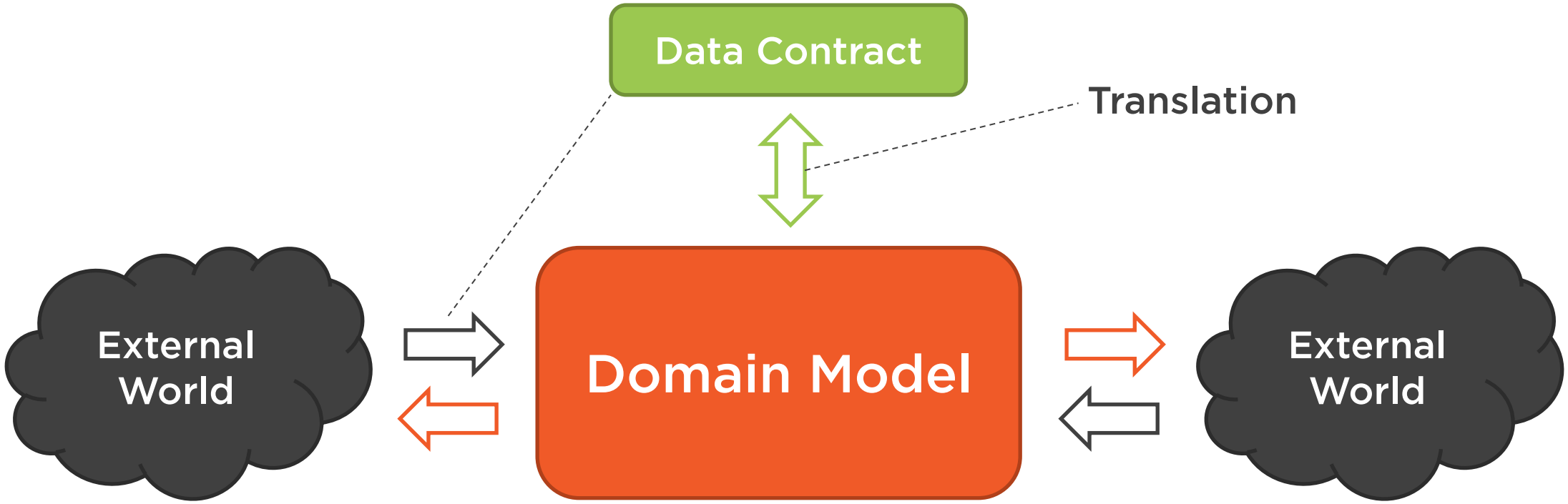
```
public class Customer : Entity {
    [Required, MaxLength(100, ErrorMessage = "Name is too long")]
    public virtual string Name { get; set; }

    [Required, RegularExpression(@"^(.+)@(.+)$", ErrorMessage = "Email is invalid")]
    public virtual string Email { get; set; }

    [JsonConverter(typeof(StringEnumConverter))]
    public virtual CustomerStatus Status { get; set; }
    public virtual DateTime? StatusExpirationDate { get; set; }
    public virtual decimal MoneySpent { get; set; }
    public virtual IList<PurchasedMovie> PurchasedMovies { get; set; }
}
```

Extracting Output Data Contracts

How to implement the decoupling?

Introduce Data Transfer Objects (DTOs)

Extracting Output Data Contracts



Extracting Output Data Contracts

Output data contracts = DTO

Input data contracts = ...Model, ...Request } DTO

Domain-Driven Design in Practice

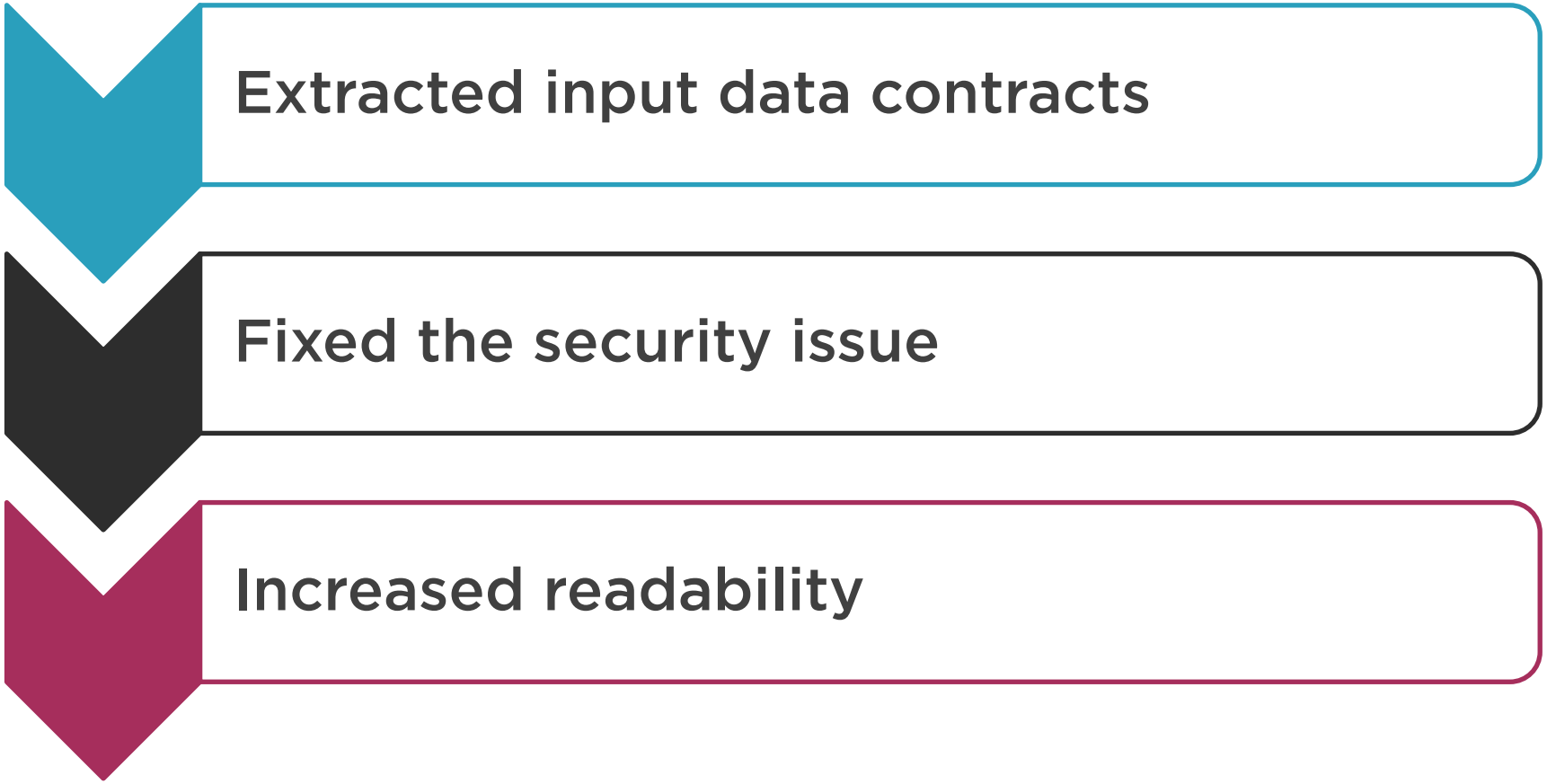
by Vladimir Khorikov

A descriptive, in-depth walk-through for applying Domain-Driven Design principles in practice.

▶ Resume Course

Table of contents | Description | Transcript | Exercise files | Discussion | Learning Check | Recommended

Recap: Extracting Input Data Contracts



Summary

Decoupled the domain model from the application data contracts

Data contract is the shape of the data that comes in and comes out of your application

Avoid coupling data contracts to the domain model

- You won't be able to refactor the domain model
- Such coupling poses security risks

Extracted data contracts in the form of DTOs

Validation and serialization attributes on domain classes is a sign of coupling

In the Next Module

Using Value Objects as Domain Model Building Blocks