

Do you trust your domain objects?

sysart

27.10.2010 Marko Oikarinen & Daniel Wellner

Why are we here?

We have been applying Domain-Driven concepts to large-scale projects in many fields including:

- » Steel industry
- » Electronics industry
- » Health-care
- » Automotive industry

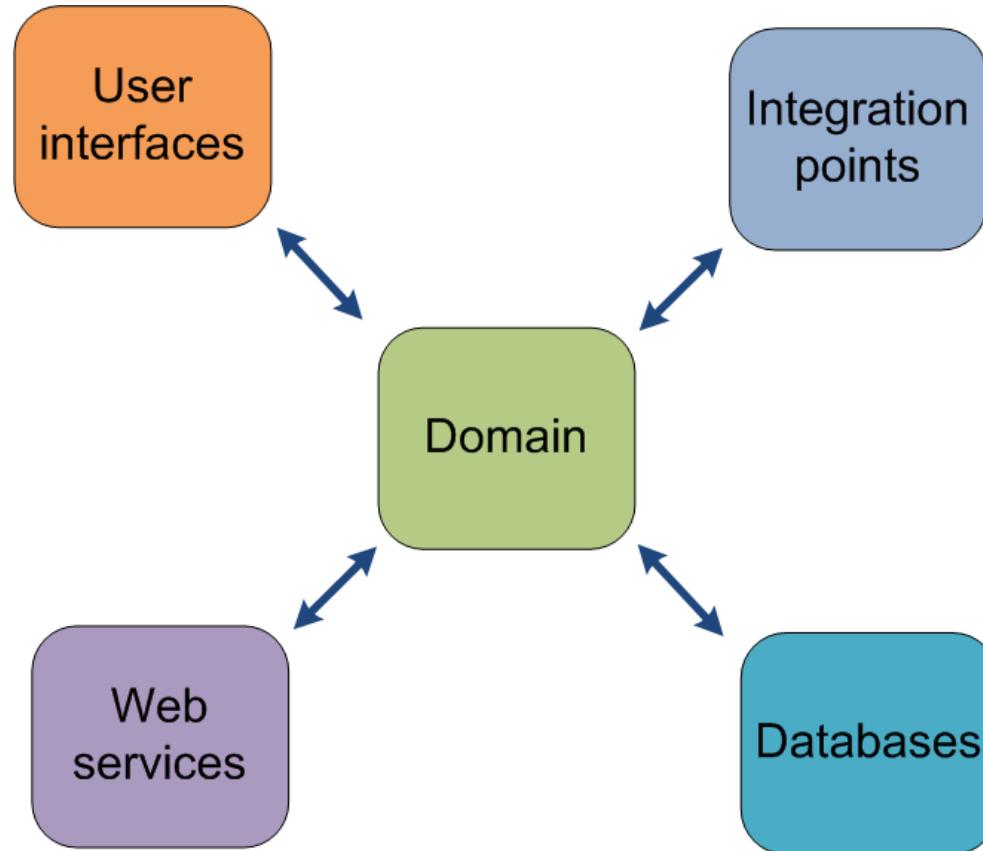
Marko Oikarinen
Object-Oriented Enthusiast



Daniel Wellner
Improvement Evangelist



What are we talking about?



A little experiment

Value objects vs. primitive obsession

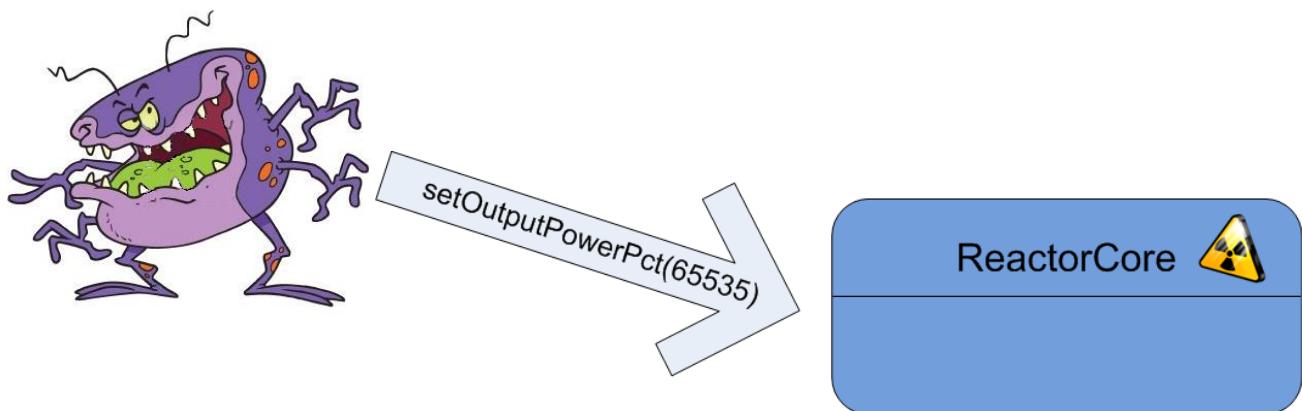
- » Value types remove ambiguity

```
int duration; // What? hours, milliseconds, years???
```

- » Communication of purpose
- » Domain logic
- » What other responsibilities should the value objects have?

Validation

- » Bad data can come in not only though the door but the windows and chimney too (not to mention the sewer).
- » Upper/lower constraints, min/max length, ...
- » Loose domain vs. strict domain (It should be impossible to create invalid values)



Where else would the validation be?

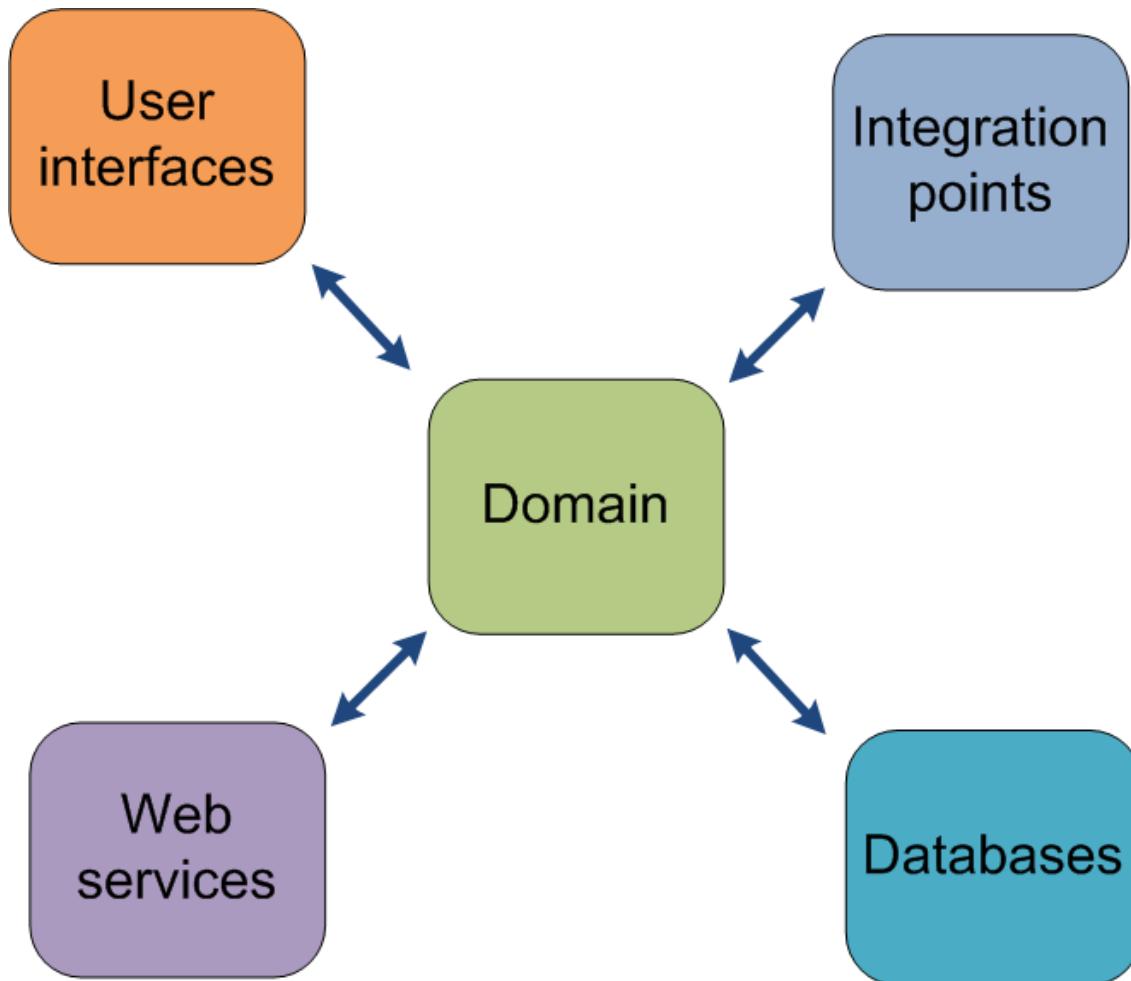
- » UI validation frameworks → do not help with integration databases, messaging etc.
- » Annotation-based validators in entities

```
public class Product : Entity {  
  
    @StringLengthValidator(64)  
    String Description { get; set; }  
  
    ...  
}
```

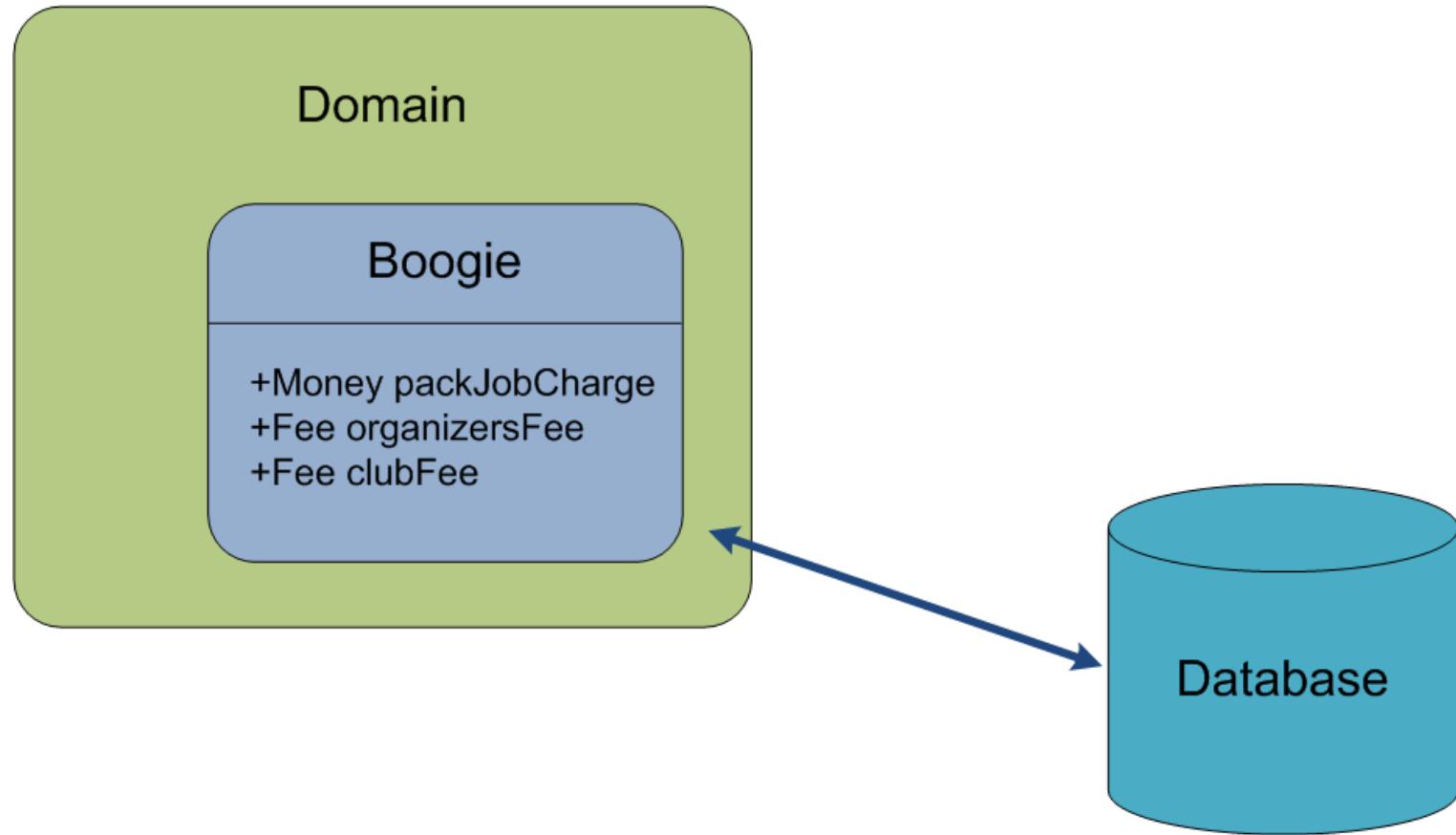
... but what if the entity is not around?

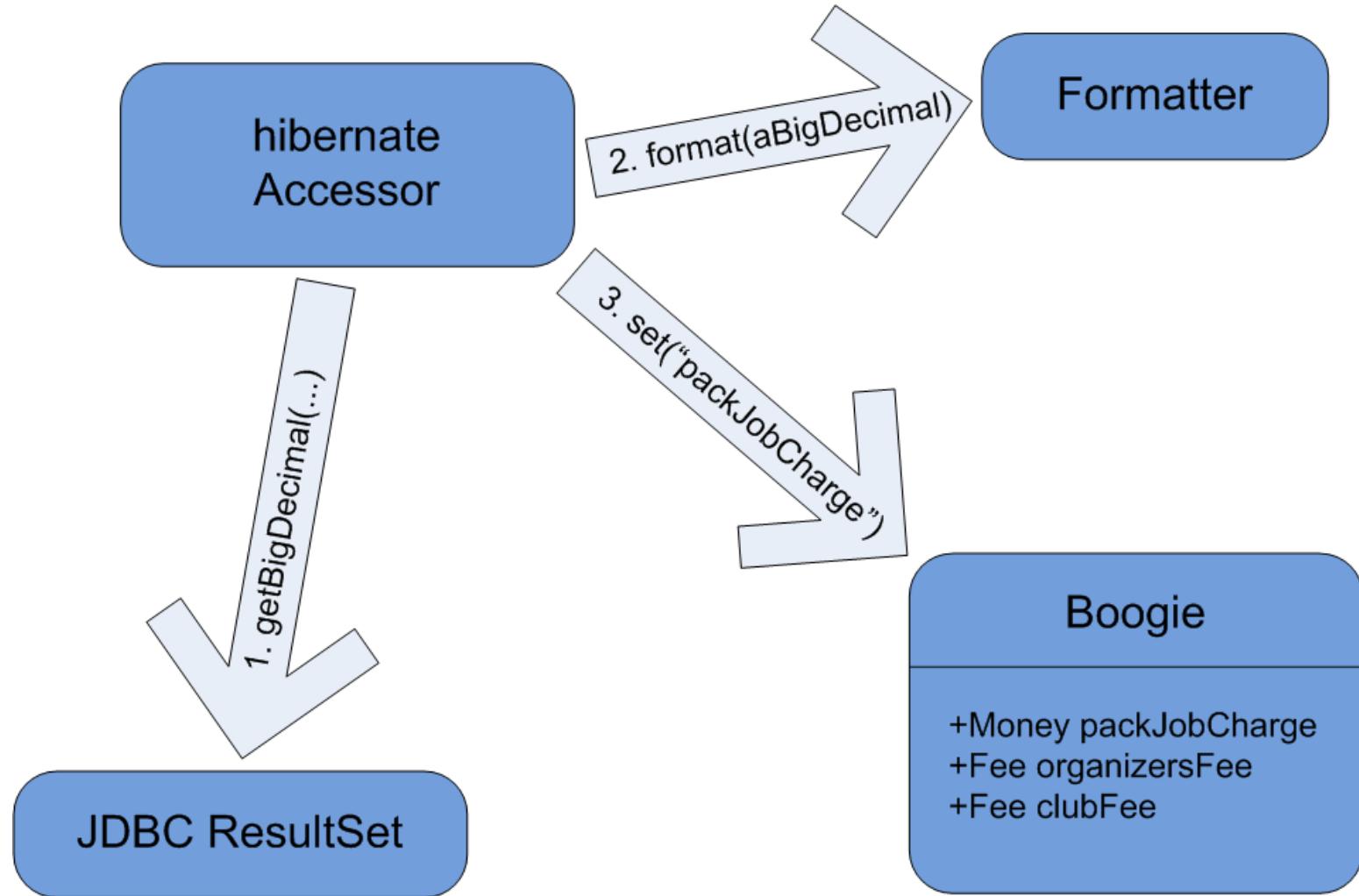
Value objects summary

- » Value types remove ambiguity
- » Value types help reduce noise
- » A value type carries the validation rules with it everywhere in the system
- » Promote a primitive value to a real typed value as soon as it enters the system. Fail Fast!

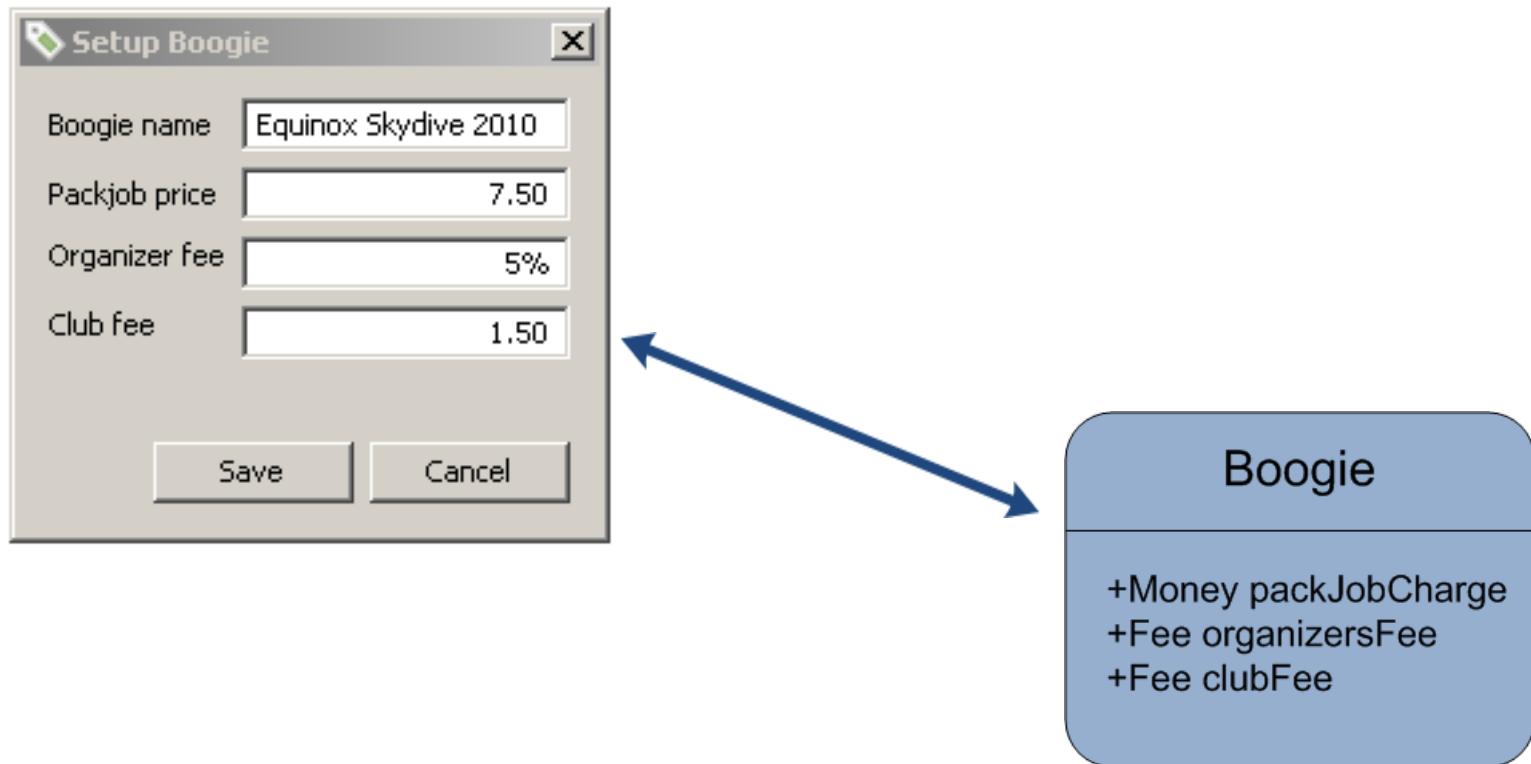


Let's look at some code using hibernate...





How about the UI?

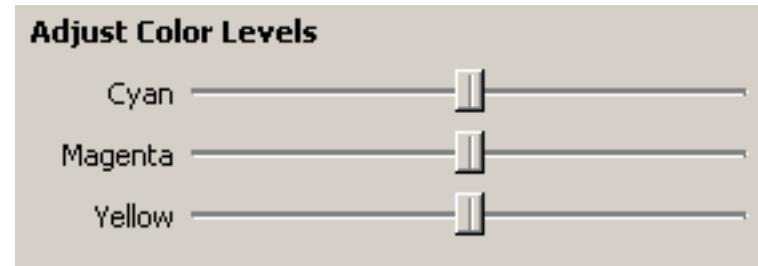


Conversion / Formatting

- » Type conversion is a separate responsibility. A value type should not have to be able to convert itself to/from all possible formats.

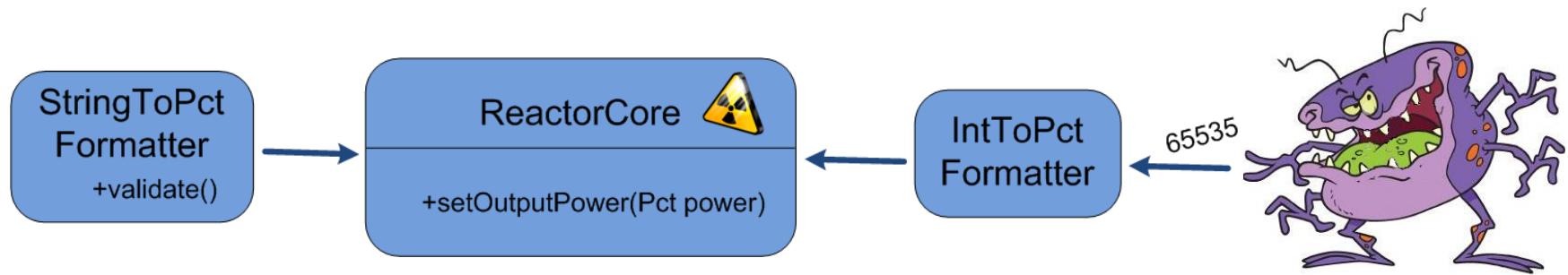
```
return Timestamp.fromEastBrutopianDateExpressedAsLong(eastBrutopianNotation);
```

- » Type conversion is not done in the UI layer only (JSF Converter, ...). It is needed at all edges of the system.
- » JSF Converter and many others convert only to/from a String. We're not always working with strings.



Conversion/Validation responsibilities

- » The conversion layer should not try to be smart and understand business meaning. Ensuring validity belongs to the domain:



- » Value validation: Is this number a valid percentage?
- » Entity validation: does this combination of values make sense?
- » Contextual validation: validateForCheckIn()

Just how far should we go with typing values?

- » Typed keys: CustomerId or Id<Customer>
- » No plain strings: PersonName, RigDescription
- » Can you go too far?

```
Length tableLength = Length.meters(1.8);  
Width tableWidth = Width.meters(1.0);  
Height tableHeight = Height.centimeters(80);
```