

(This slide is intentionally left blank)



**THE ROAD TO HELL IS PAVED WITH
JAVASCRIPT AND PRIMITIVE TYPES.**



ONCE UPON A TIME...



FEARLESS TEAM OF DEVELOPERS





THE USUAL WAY



PRIMITIVE TYPES



```
public void addToOrder(  
    long orderId, long itemId, long quantity  
);
```



```
class User {  
    private String firstName;  
    private String lastName;  
    private String emailAddress;  
    private String phoneNumber;  
}
```

**EVERYTHING WAS
FINE
USUALLY...**





GDPR

GENERAL DATA PROTECTION REGULATION





**THE ROAD TO HELL IS PAVED WITH
JAVASCRIPT AND PRIMITIVE TYPES.**



GENERAL DATA PROTECTION REGULATION

- The right of access
- The right to data portability
- The right to erasure
- Deadline: 25 May 2018
- Approved on: 14 April 2016

GDPR - OUR PROBLEMS

- many services
- manual work
- no common domain/data model
- primitive types (userId = String | Long | Int)

GDPR

CTRL + F

- emailAddress
- email_address
- email
- mail
- e_mail

AVOID PRIMITIVE TYPES



STRINGLY-TYPED



STRINGLY-TYPED NUMBERS



```
public void addToOrder(  
    long orderId, long itemId, long quantity  
);  
  
addToOrder(order, item, quantity);  
addToOrder(item, order, quantity);  
addToOrder(order, item, -1);  
addToOrder(order, item, 1000000);  
(orderId + itemId) * quantity
```

DEFENSIVE PROGRAMMING

```
public void addToOrder(  
    long orderId, long itemId, long quantity  
) {  
    if(quantity <= 0 || quantity > MAX_ORDER)  
        throw new IllegalArgumentException("foo bar");  
  
    if(!orderService.orderExists(orderId))  
        throw new IllegalStateException("foo bar");  
  
    if(!itemService.itemExists(itemId))  
        throw new IllegalStateException("foo bar");  
  
    //  
}
```

STRINGLY-TYPED TEXT



```
class User {  
    private String firstName;  
    private String lastName;  
    private String emailAddress;  
    private String phoneNumber;  
}
```



```
public void sendNotification(String email) {  
    if(!emailValidator.isValid(email)) {  
        throw new RuntimeException("...")  
    }  
    if(!emailService.isVerified(email)) {  
        throw new RuntimeException("...")  
    }  
}
```

VALUE = TYPE

???



VALUE = TYPE + CONSTRAINTS



**VALUE = TYPE + CONSTRAINTS +
UNITS**

```
public void addToOrder(  
    OrderId orderId, ItemId itemId, Quantity quantity  
)  
  
public void sendNotification(  
    VerifiedEmail email  
)
```

THE CORE OF DDD

ENTITIES AND VALUE OBJECTS



REQUIREMENTS

- specific types (not stringly-typed)
- immutability

SIMPLE TYPES

```
public class FirstName {  
    private final String value;  
  
    public FirstName(String value) {  
        this.value = value;  
    }  
  
    public String getValue() {  
        return this.value;  
    }  
}
```


COMPLEX TYPES

```
public class User {  
    private final FirstName firstName;  
    private final LastName lastName;  
    private final EmailAddress emailAddress;  
    private final TelephoneNumber telephoneNumber;  
  
    (...)  
}
```

**THAT WAS THE EASY
PART...**



```
public User withFirstName(FirstName firstName) {  
    return new User(  
        firstName,  
        this.lastName,  
        this.emailAddress,  
        this.phoneNumber  
    );  
}
```

AVOID PRIMITIVE TYPES



SCALA



SCALA

EFFORTLESS



SIMPLE TYPES

```
class FirstName(val value: String) {}  
  
case class FirstName(value: String)  
  
class FirstName(val value: String) extends AnyVal {}
```

COMPLEX TYPES

```
case class User(  
  firstName: FirstName,  
  lastName: LastName,  
  emailAddress: EmailAddress,  
  phoneNumber: PhoneNumber  
)
```


IMMUTABILITY?

OUT OF THE BOX



IMMUTABILITY.

```
user.copy(firstName = FirstName("John"))
```

SELF DOCUMENTING

```
public void addToOrder(long, long, long)
```

```
def addToOrder(OrderId, ItemId, Quantity): Unit
```

"Two methods have the same signature if they have the same name and argument types." - The Java Language Specification

CASE CLASSES



COMPANION OBJECT

```
case class FirstName(value: String)

object FirstName {
  def apply(firstName: String): FirstName = new FirstName(firs
}

val firstName = FirstName("John")
```

CASE CLASS - FUNCTIONS

```
case class TemperatureFahrenheit(val value: Double) {  
  def asCelsius = TemperatureCelsius(value * 9/5 + 32)  
}  
  
case class TemperatureCelsius(val value: Double) {  
  def asFahrenheit = TemperatureFahrenheit((value - 32) * 5/9)  
}
```

CASE CLASSES

- companion object
- pattern matching
- can have functions and fields (just like a class)
- copy method (shallow copy)
- equals/hashCode

MORE HYPE

WHY SHOULD I CARE?



MARS CLIMATE ORBITER



On September 23, 1999, communication with the spacecraft was lost as the spacecraft went into orbital insertion, due to ground-based computer software which produced **output in non-SI units of pound-force seconds (lbf·s) instead of the SI units of newton-seconds (N·s)** specified in the contract between NASA and Lockheed.

The spacecraft encountered Mars on a trajectory that brought it too close to the planet, causing it to pass through the upper atmosphere and disintegrate.

UNDEFINED IN NOT AN ENGINE



PGM-17 Thor (25 January 1957)

AVOID PRIMITIVE TYPES EXCUSES



EXCUSE 1

MORE CODE



**NO DEFENSIVE PROGRAMMING
REMOVED DUPLICATE CODE
VALIDATION IN ONLY ONE PLACE
= LESS ERRORS**



```
public void addToOrder(  
    long orderId, long itemId, long quantity  
) {  
    if(quantity <= 0 || quantity > MAX_ORDER)  
        throw new IllegalArgumentException("foo bar");  
  
    if(!orderService.orderExists(orderId))  
        throw new IllegalStateException("foo bar");  
  
    if(!itemService.itemExists(itemId))  
        throw new IllegalStateException("foo bar");  
  
    //  
}
```

LESS TESTS
INVALID CODE DOES NOT EVEN
COMPILE



EXCUSE 2

**MEMORY ALLOCATION = SLOWER
CODE**



LESS CODE = LESS DB/REST CALLS

```
class FirstName(val value: String) extends AnyVal {}
```

AT RUNTIME IT IS JUST A STRING*

*most of the time

EXCUSE 3

DUPLICATED CODE

DOMAIN MODEL

DTO

ORM ENTITIES



SINGLE RESPONSIBILITY PRINCIPLE



EXCUSE 4

**IT IS NOT THE WAY
WE WRITE CODE**



"The most dangerous phrase in the language is, "We've always done it this way." - Grace Murray Hopper

EXCUSE 5

DIFFICULT



**SOFTWARE ENGINEERING + SOFTWARE
CRAFTSMANSHIP
VS.
CODING + USING UGLY HACKS**



SOFTWARE CRAFTSMANSHIP

- clean code
- self-documenting code
- code that is easy to understand for the next programmers
- code that makes it hard to make a mistake

EXCUSE 6

**EVERYONE CAN
UNDERSTAND MY
CODE**

WHAT ABOUT JOB SECURITY?



EXCUSE 7

IT IS NOT COOL



JAVA -> SCALA

~~functional~~

~~monad~~

~~monoid~~

~~applicative~~

~~traversable~~



**WHAT'S THE ONE THING YOU CAN DO, SUCH THAT BY
DOING IT, EVERYTHING ELSE WILL BE EASIER OR
UNNECESSARY?**

"The One Thing" - Gary Keller

AVOID PRIMITIVE TYPES



**EASY IN SCALA, BUT YOU CAN DO
IT IN JAVA TOO**



DOING DDD IS NOT A BINARY CHOICE



VALUE OBJECTS AND ENTITIES = THE EASIEST PART OF DDD



DISCLAIMER

SAMPLE SIZE: 1

MICROSERVICES?



**THE ROAD TO HELL IS PAVED WITH
JAVASCRIPT AND PRIMITIVE TYPES.**



AVOID PRIMITIVE TYPES



EFFORTLESS DOMAIN-DRIVEN DESIGN

THE REAL POWER OF SCALA

BARTOSZ MIKULSKI

@MIKULSKIBARTOSZ



QUESTIONS?



1. ARE THERE ANY OTHER EXCUSES?

WHAT IS STOPPING YOU FROM DOING THAT?



2. WHEN IS IT A BAD IDEA?

"BEST PRACTICES" ARE NOT UNIVERSAL, IN SOME SITUATIONS THEY ARE HARMFUL

QUESTIONS?



EFFORTLESS DOMAIN-DRIVEN DESIGN

THE REAL POWER OF SCALA

BARTOSZ MIKULSKI

@MIKULSKIBARTOSZ

