Clean TDD

Clean Code & Test Driven Development

sopra steria | ORDINA a Sopra Steria company | TOBANIA a Sopra Steria company

De Reis naar Schone Code

sopra steria | ORDINA | TOBANIA

Intentie-verduidelijkende Namen

sopra steria | ORDINA | TOBANIA

```java
// Don't
int d; // elapsed time in days

List<Account> accountList = new ArrayList<>();

// Do
int elapsedTimeInDays;
int daysSinceCreation;
int fileAgeInDays;

List<Account> accounts = new ArrayList<>();
```

Uitspreekbare Namen

```java
// Don't
record CstmrRcrd(Person person) {}

// Do
record CustomerRecord(Person person) {}
```

Vermijden van Primitieve Obsessie

```
public boolean
isCostHigherThanOtherCost(BigDecimal
product1Cost, BigDecimal product2Cost) {
    return product1Cost.compareTo(product2Cost)
> 0;
}
```

```
public boolean isCostHigherThanOtherCost(Money
product1Cost, Money product2Cost) {
    return
product1Cost.isHigherThan(product2Cost);
}
```

Functies

Minimaliseren van Parameters

Vermijden van Flags

Commentaren

Foutafhandeling

De TDD Uitdaging

sopra steria | ORDINA | TOBANIA

Waarom Tests Schrijven?

Het Schrijven van Tests met JUnit 5

Basis van een JUnit 5 Test

```
import
org.junit.jupiter.api.Test
;
import static
org.junit.jupiter.api.Asse
rtions.assertEquals;

class MathOperationsTest {

    @Test
    void testAdd() {
        assertEquals(2,
Math.add(1, 1), "1 + 1
should equal 2");
    }
}
```

Geavanceerd

Testen met JUnit 5

sopra steria | ORDINA | TOBANIA

Voorbeeld van `@RepeatedTest` en `@BeforeEach`

```
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.RepeatedTest;
import static
org.junit.jupiter.api.Assertions.assertNot
Equals;
import java.util.Random;

class RandomGeneratorTest {

    private Random random;

    @BeforeEach
    void setUp() {
        random = new Random();
    }

    @RepeatedTest(5)
    void testRandom() {
        assertNotEquals(random.nextInt(),
random.nextInt(), "Random numbers should
not be the same");
    }
}
```

AssertJ: Vloeiende Assertions voor Java

sopra steria | ORDINA | TOBANIA

Voorbeeld van AssertJ Gebruik

```
import static
org.assertj.core.api.Assertions.a
ssertThat;

class ListTest {

    @Test
    void testListContains() {

assertThat(Arrays.asList("apple",
"banana", "orange"))
                .as("Check if the
list contains the expected
fruits")
                .contains("banana")

.doesNotContain("grape");
    }
}
```

De Red-Green-Refactor Cyclus

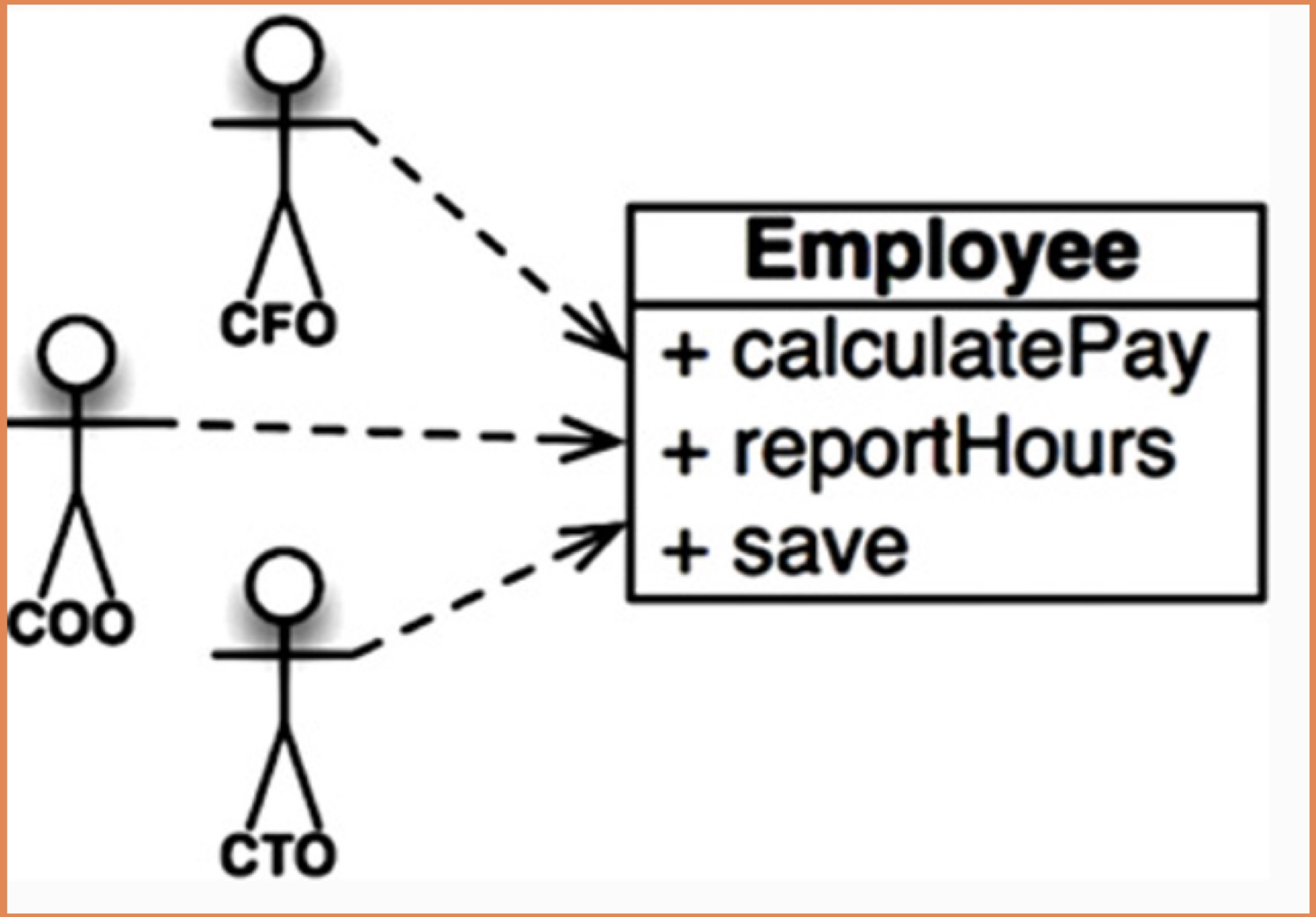
sopra steria | ORDINA | TOBANIA

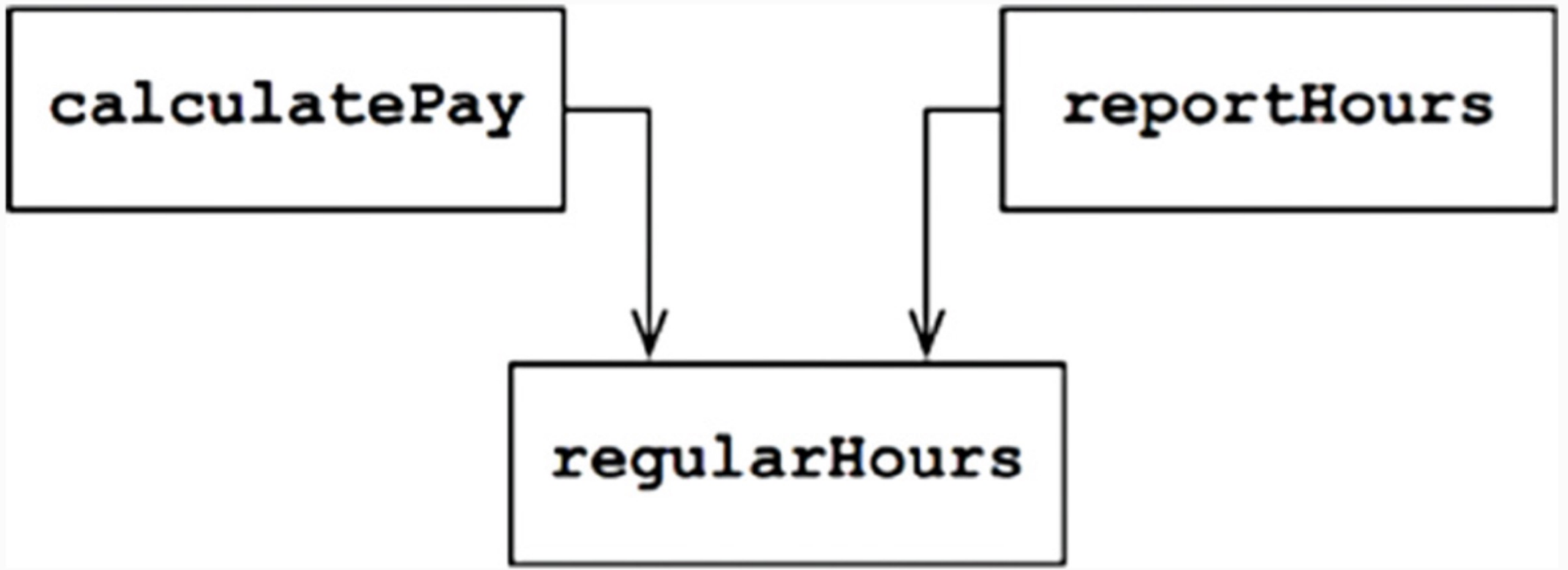
De SOLID Stenen van het Kasteel

sopra steria | ORDINA a Sopra Steria company | TOBANIA a Sopra Steria company

Single Responsibility Principle (SRP)

sopra steria | ORDINA a Sopra Steria company | TOBANIA a Sopra Steria company

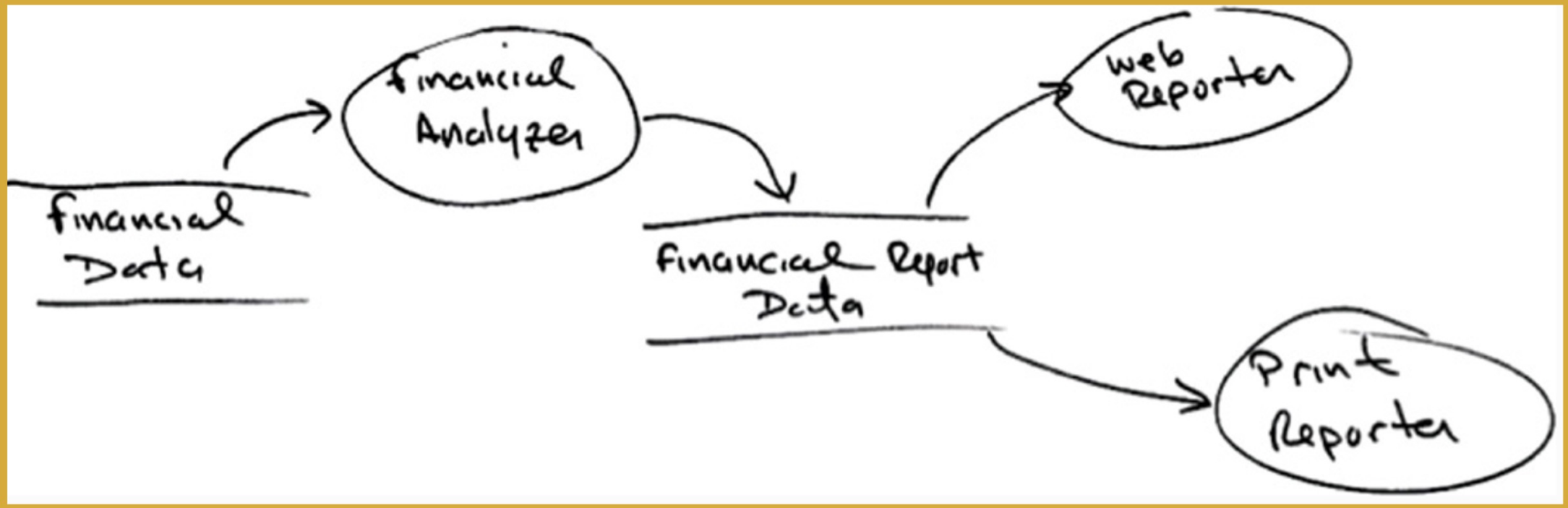

CFO
COO
CTO
Employee
+ calculatePay
+ reportHours
+ save

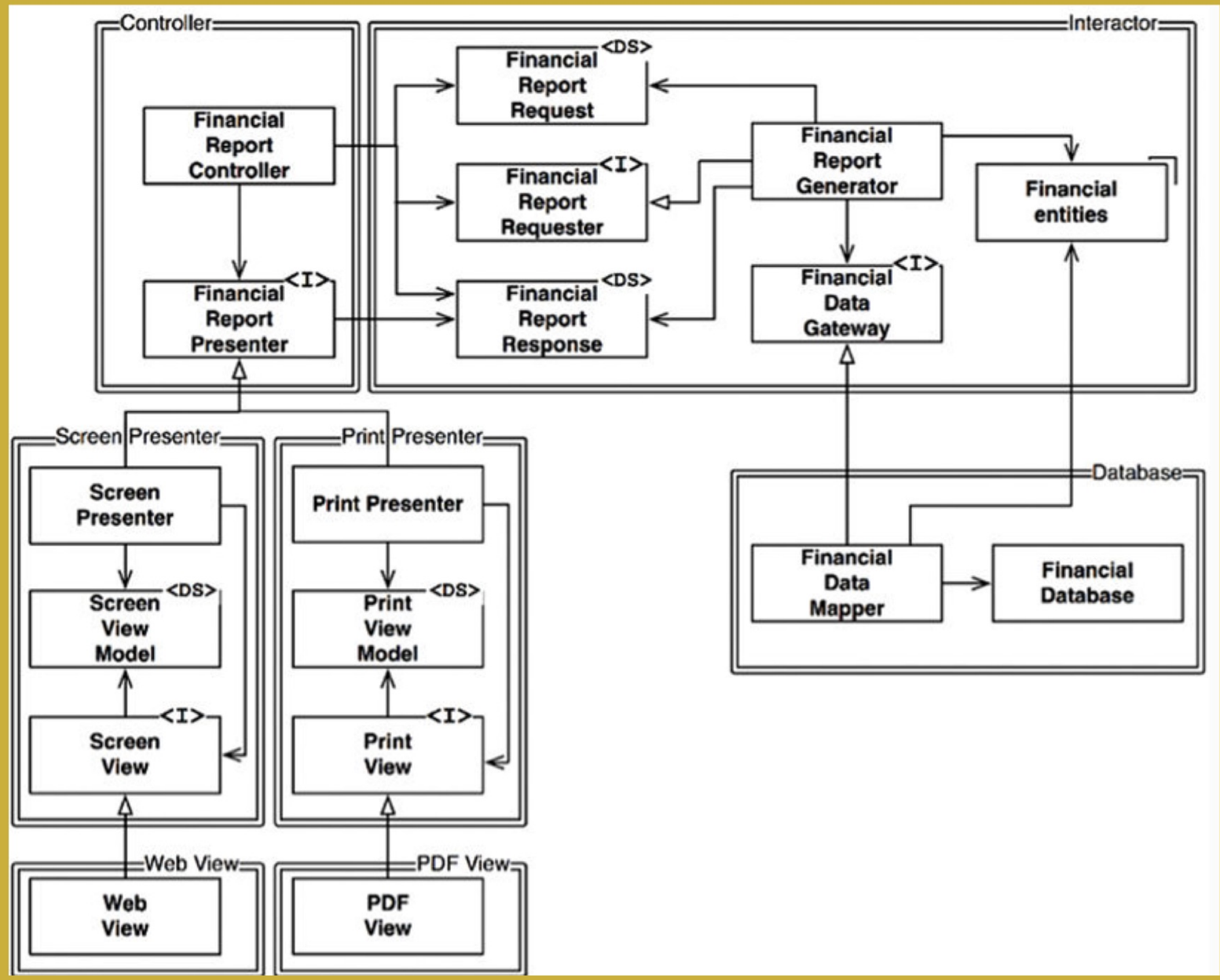

calculatePay
reportHours
regularHours

A good duplication is always cheaper than a bad abstraction.

– Sandi Metz

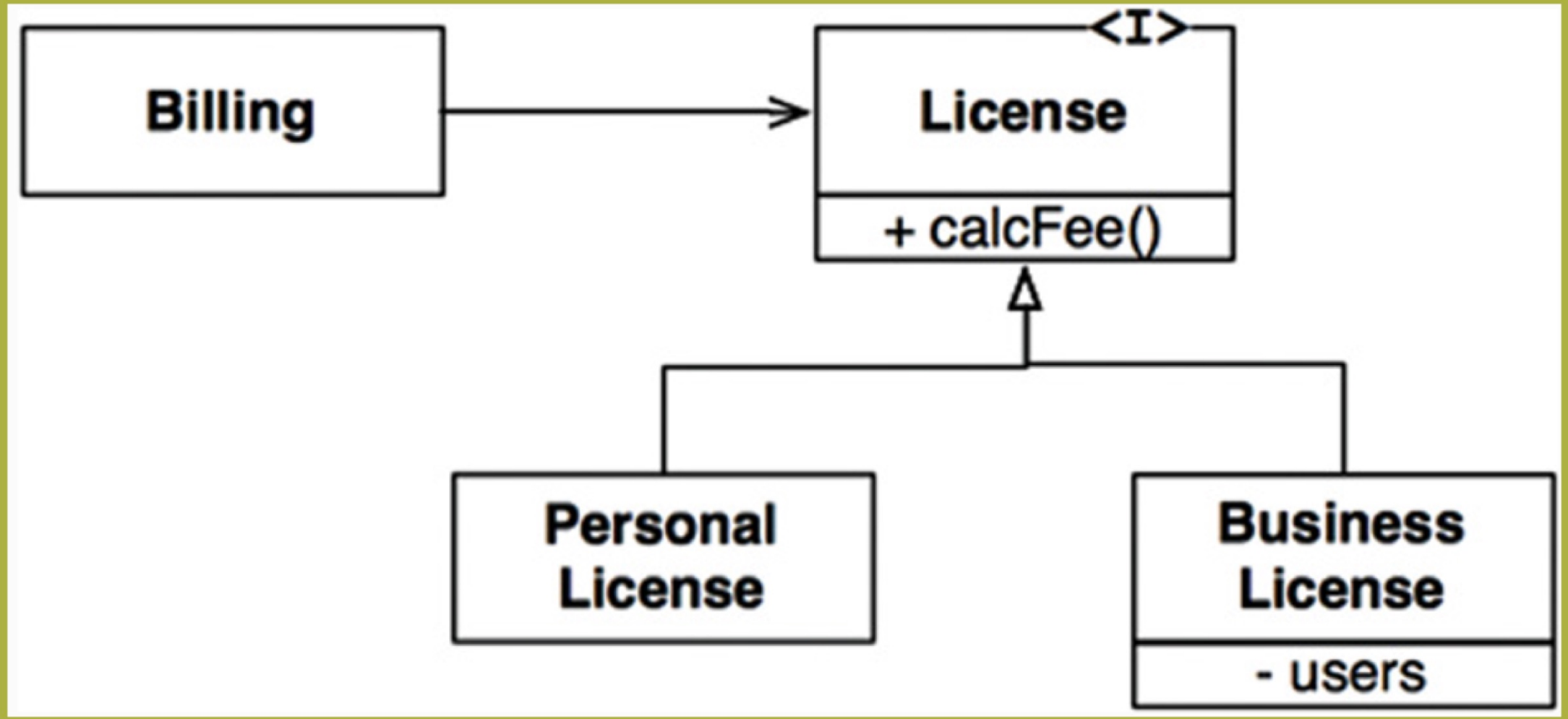
Open-Closed Principle (OCP)


Financial Analyzer
Financial Data
Financial Report Data
Web Reporter
Print Reporter

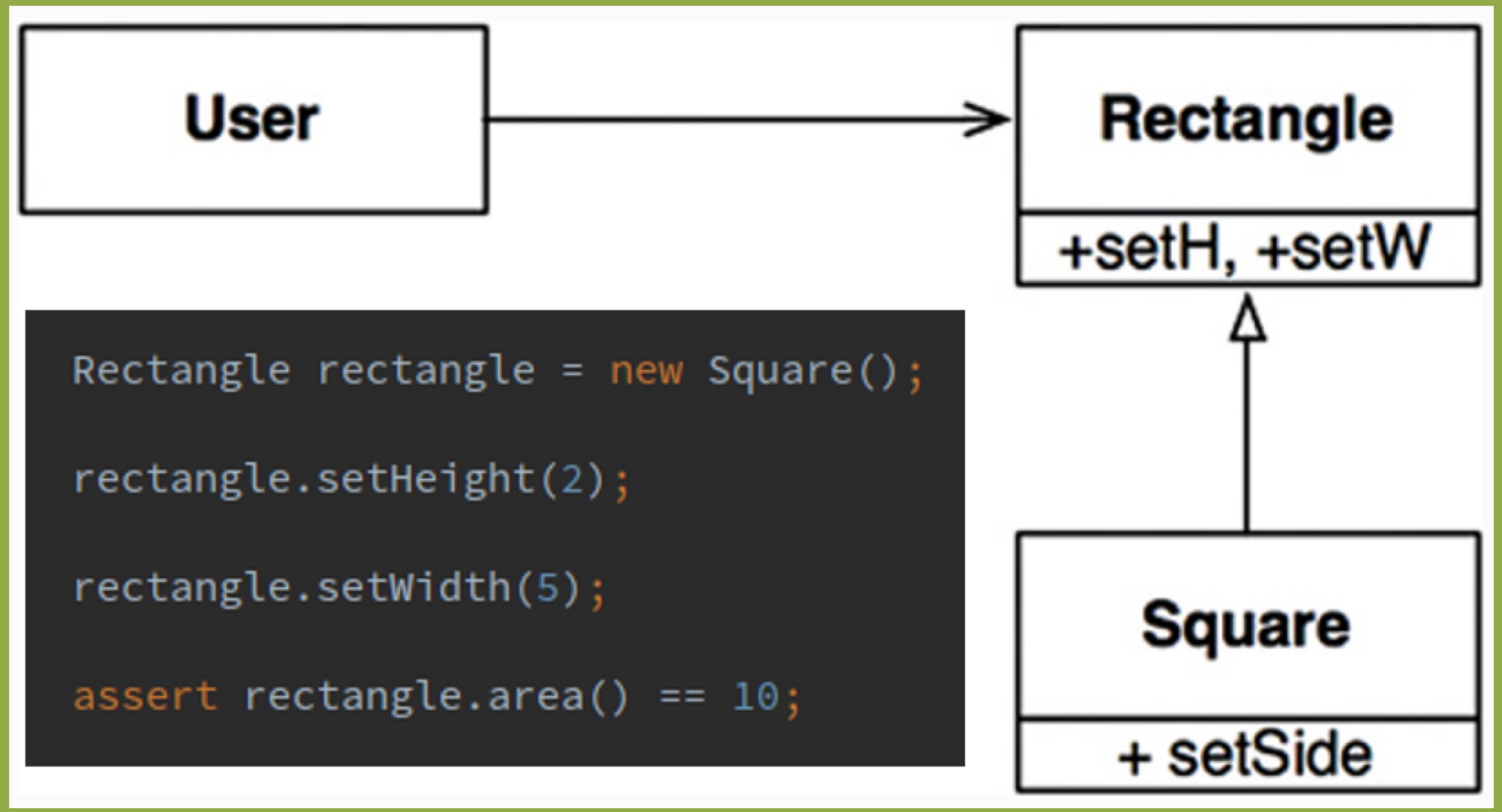

Controller
Interactor
Financial Report Controller
Financial Report Presenter <I>
Financial Report Request <DS>
Financial Report Requester <I>
Financial Report Response <DS>
Financial Report Generator
Financial Data Gateway <I>
Financial entities
Screen Presenter
Screen Presenter
Screen View Model <DS>
Screen View <I>
Print Presenter
Print Presenter
Print View Model <DS>
Print View <I>
Database
Financial Data Mapper
Financial Database
Web View
Web View
PDF View
PDF View

Liskov Substitution Principle (LSP)

Liskov Example

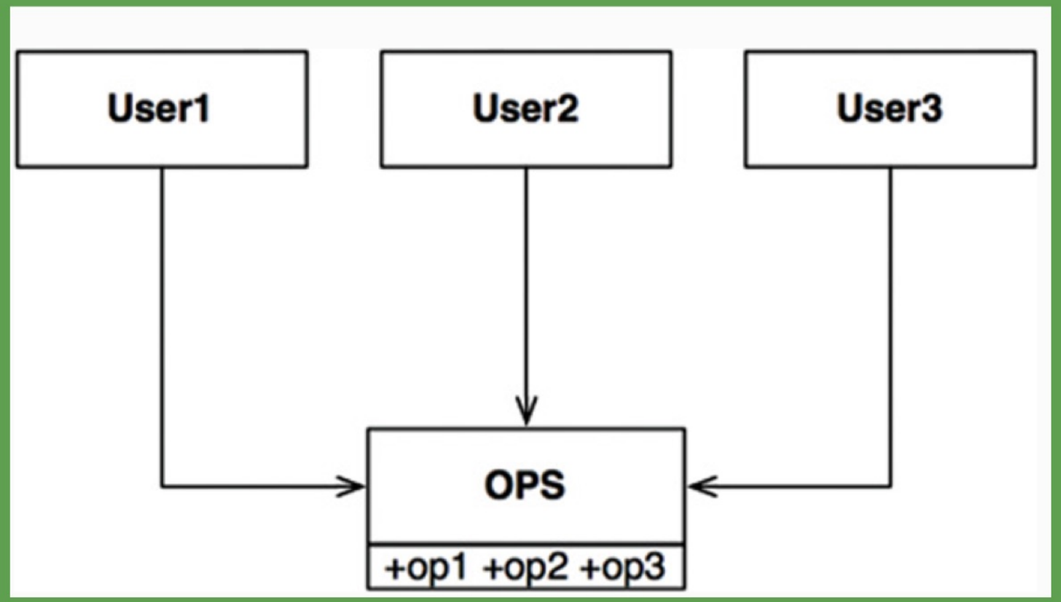


Liskov Violation

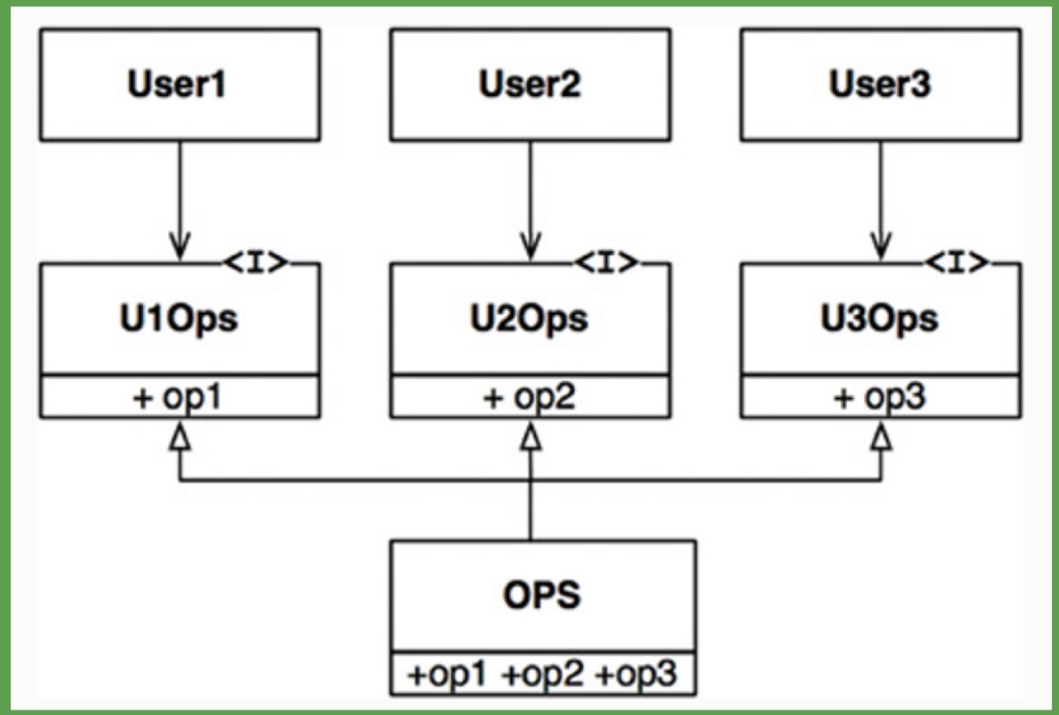



```
Rectangle rectangle = new Square();

rectangle.setHeight(2);

rectangle.setWidth(5);

assert rectangle.area() == 10;
```

Interface Segregation Principle (ISP)


User1
User2
User3
OPS
+op1 +op2 +op3

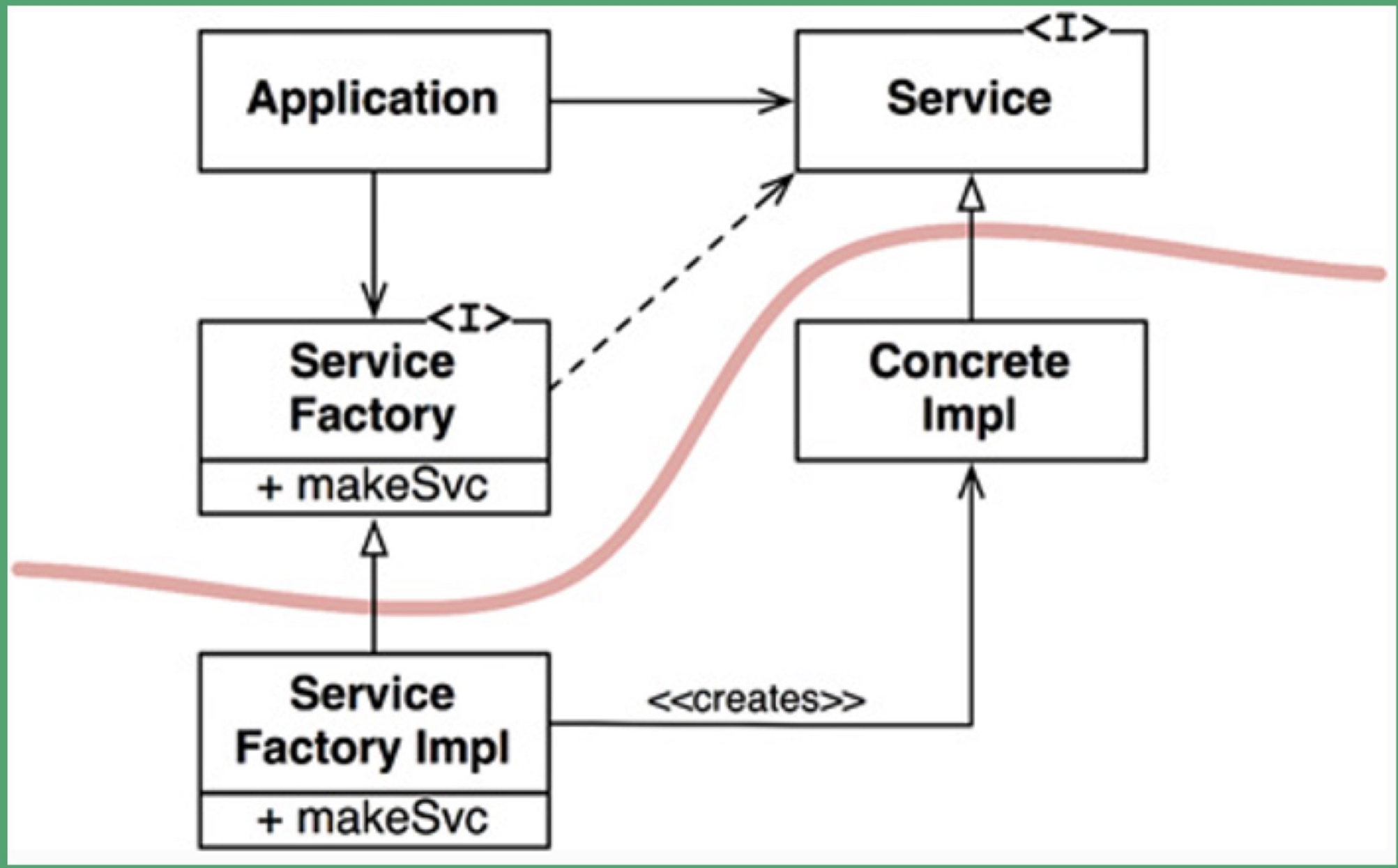
voor


User1
User2
User3
<I>
U1Ops
+ op1
<I>
U2Ops
+ op2
<I>
U3Ops
+ op3
OPS
+op1 +op2 +op3

Na

Dependency Inversion Principle (DIP)

sopra steria | ORDINA a Sopra Steria company | TOBANIA a Sopra Steria company


Application
<I>
Service
<I>
Service
Factory
+ makeSvc
Concrete
Impl
Service
Factory Impl
+ makeSvc
<<creates>>

Epiloog