

SOLID



SOLID

- Single Responsibility Principle
- Open-Closed-Principle
- Liskov Substitution Principle
- Interface Segregation Principle
- Dependency Inversion Principle

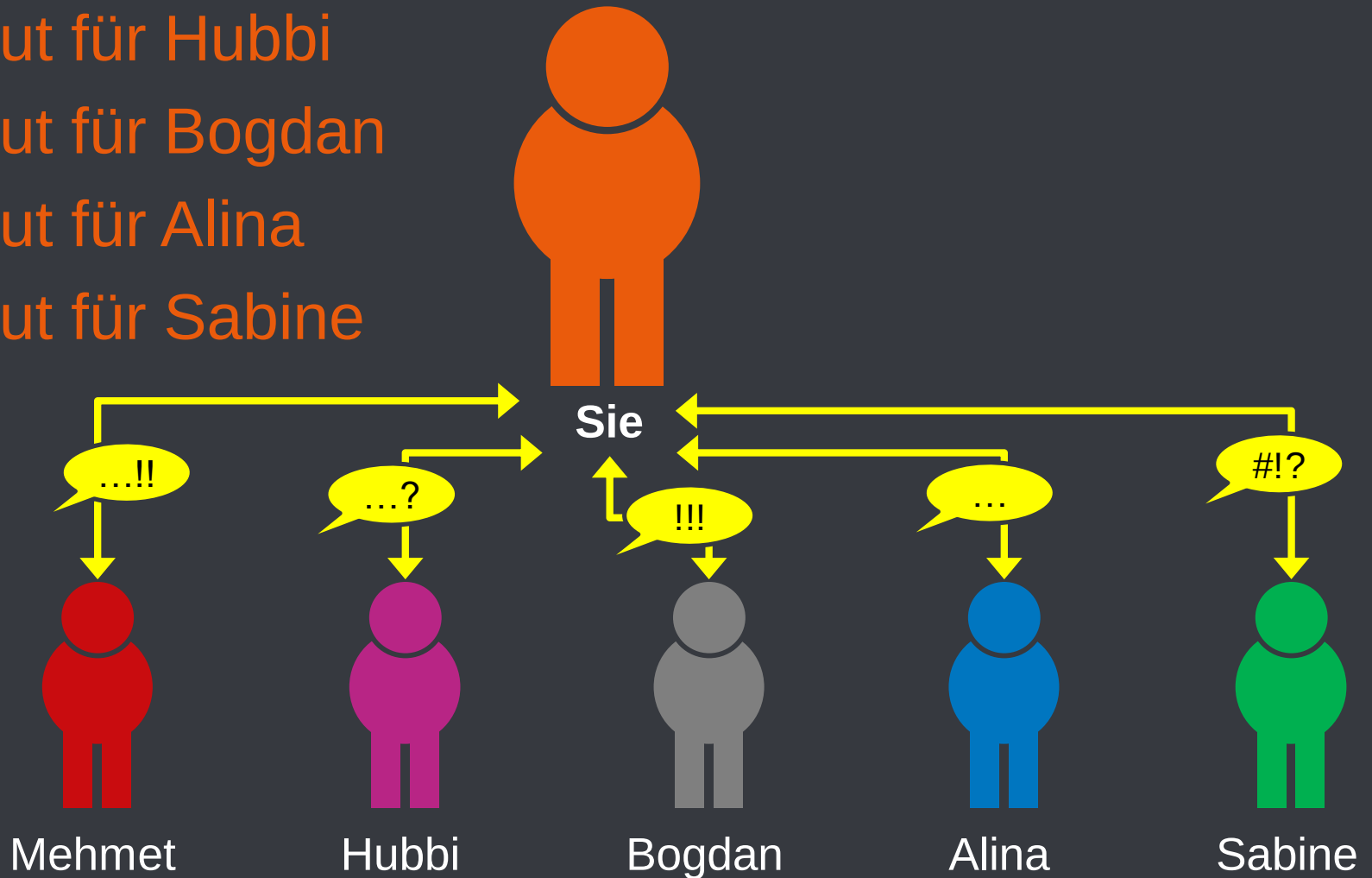
Single Responsibility Principle

(Trennung von Zuständigkeiten)

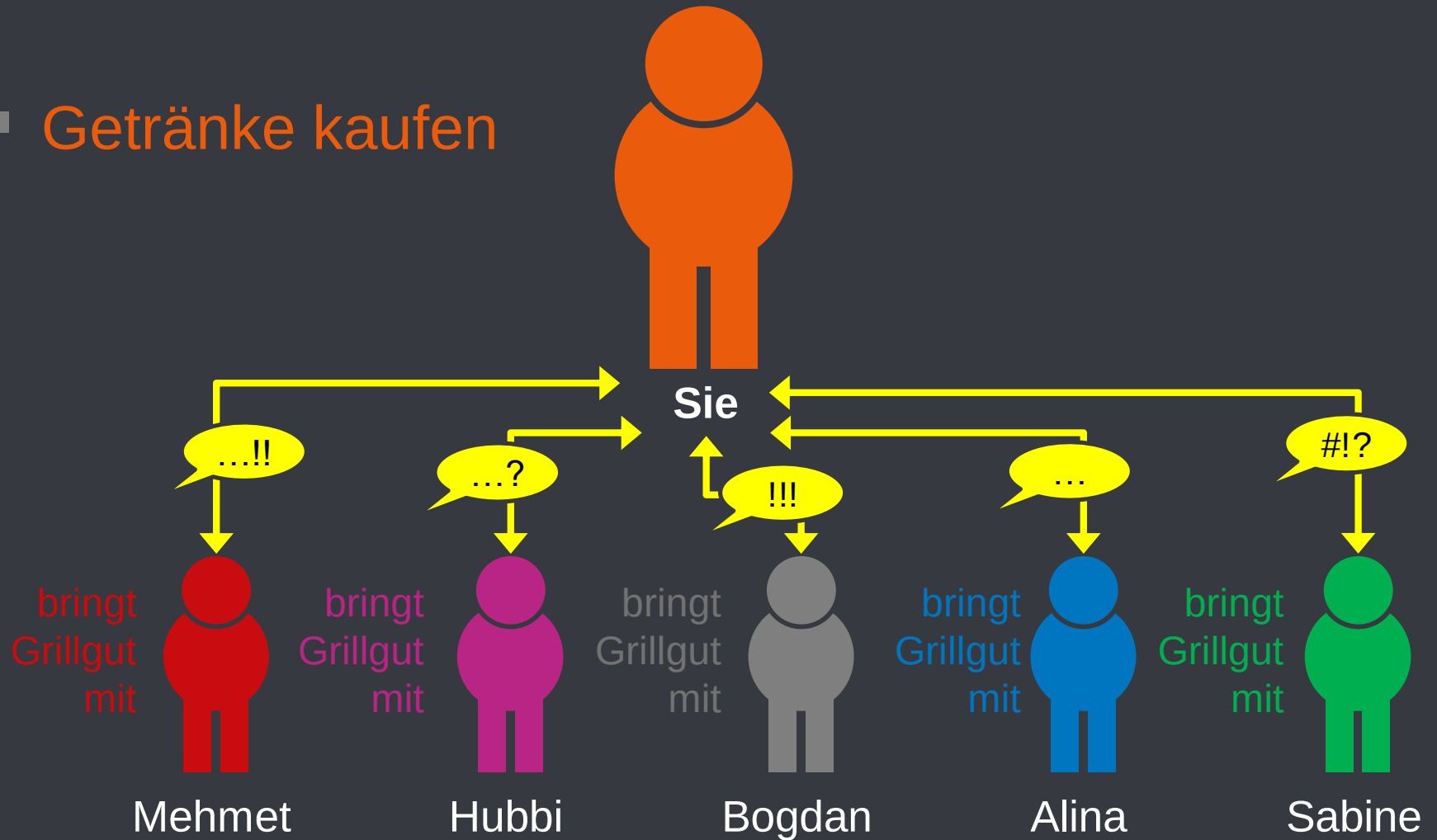


- Mehmet isst Halal
- Hubbi ist Vegetarier
- Bogdan isst nur Biofleisch
- Alina liebt Bratwurst
- Sabine mag nur diese kleinen marinierten Hühnchenfilets ...

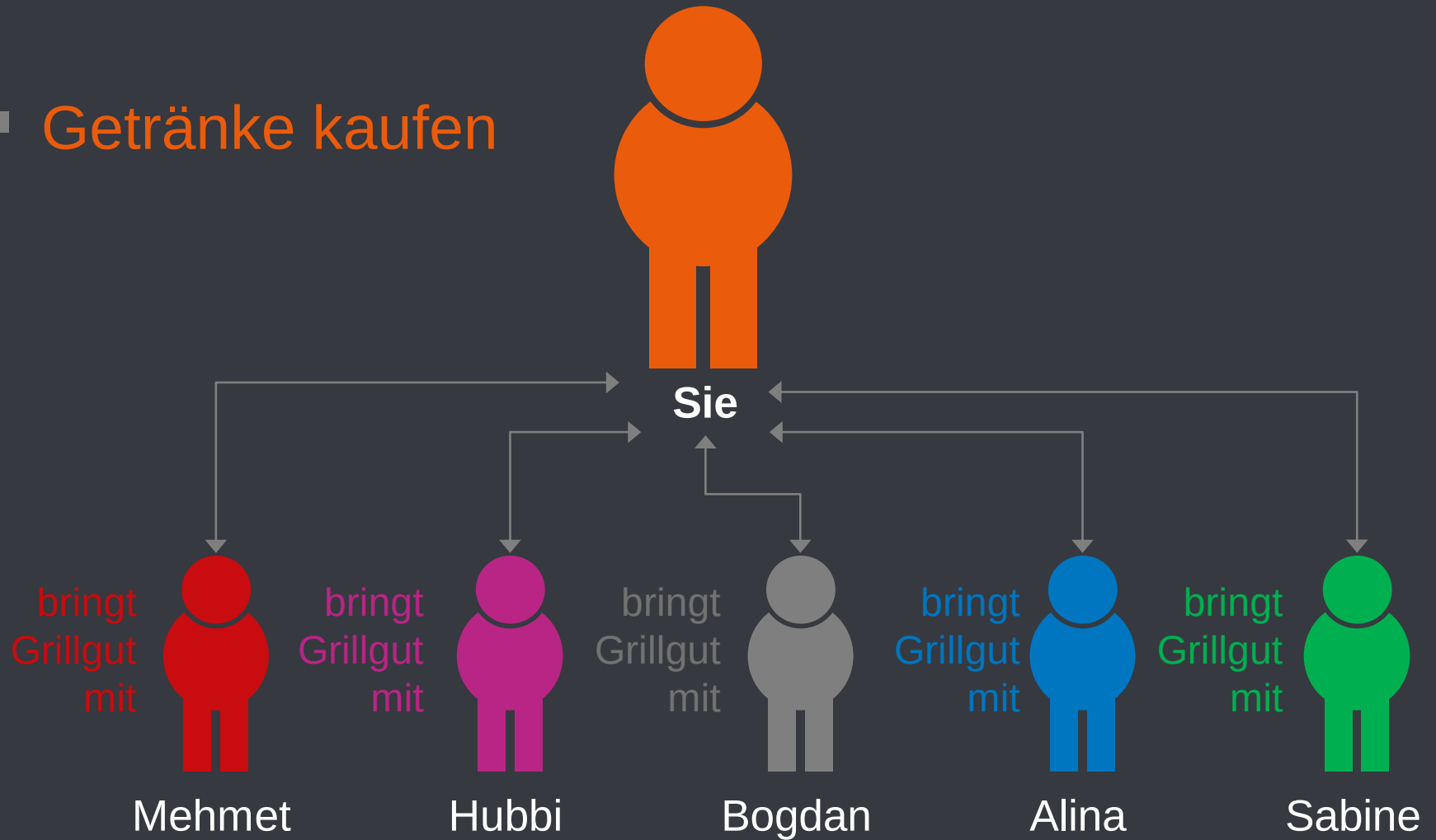
- Getränke kaufen
- Grillgut für Mehmet
- Grillgut für Hubbi
- Grillgut für Bogdan
- Grillgut für Alina
- Grillgut für Sabine



- Getränke kaufen



- Getränke kaufen



Single Responsibility Principle

(a.k.a. Separation of Concerns)



Single Responsibility Principle

(a.k.a. Separation of Concerns)

1

2

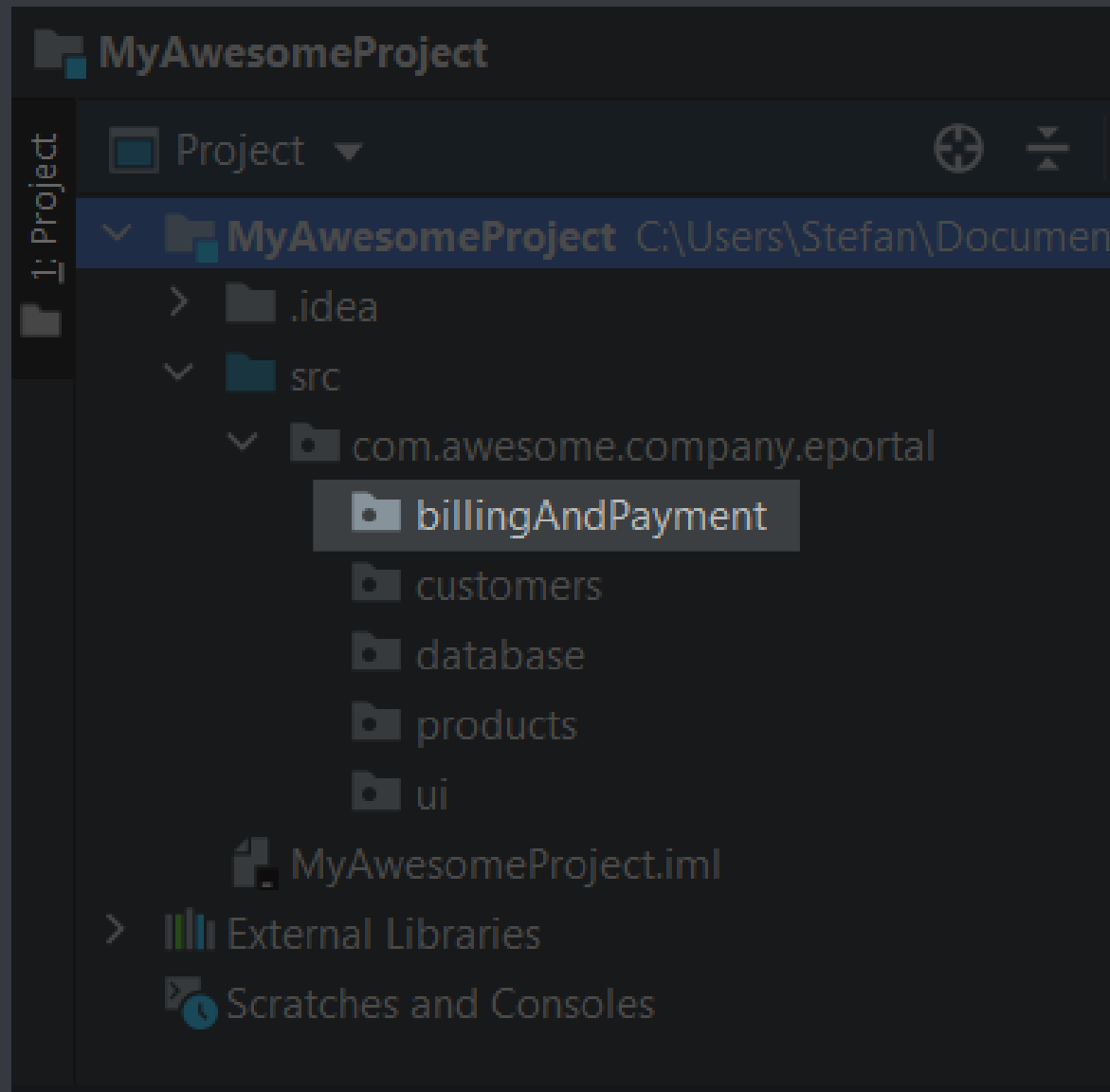
Single Responsibility Principle

(a.k.a. Separation of Concerns)

- Entity / Package / Modul / ... ist für **genau einen Zweck** zuständig
- + weniger Wartung
- + besser verständlich



Bad Smell: “and”, “or”, “common”, ...





Open-Closed-Principle



Process

INTERMEDIATE
BLOCK

Process

TIME ENTERED
TIME EXITED

?



Bitte eine
Steuer-
nummer ...

Gewerbe-
anmeldung?
Gesellschafter-
vertrag? Notar?



Ich möchte
ein Gewerbe
anmelden.

Mitarbeiter?
Berufs-
genossenschaft!
IHK!

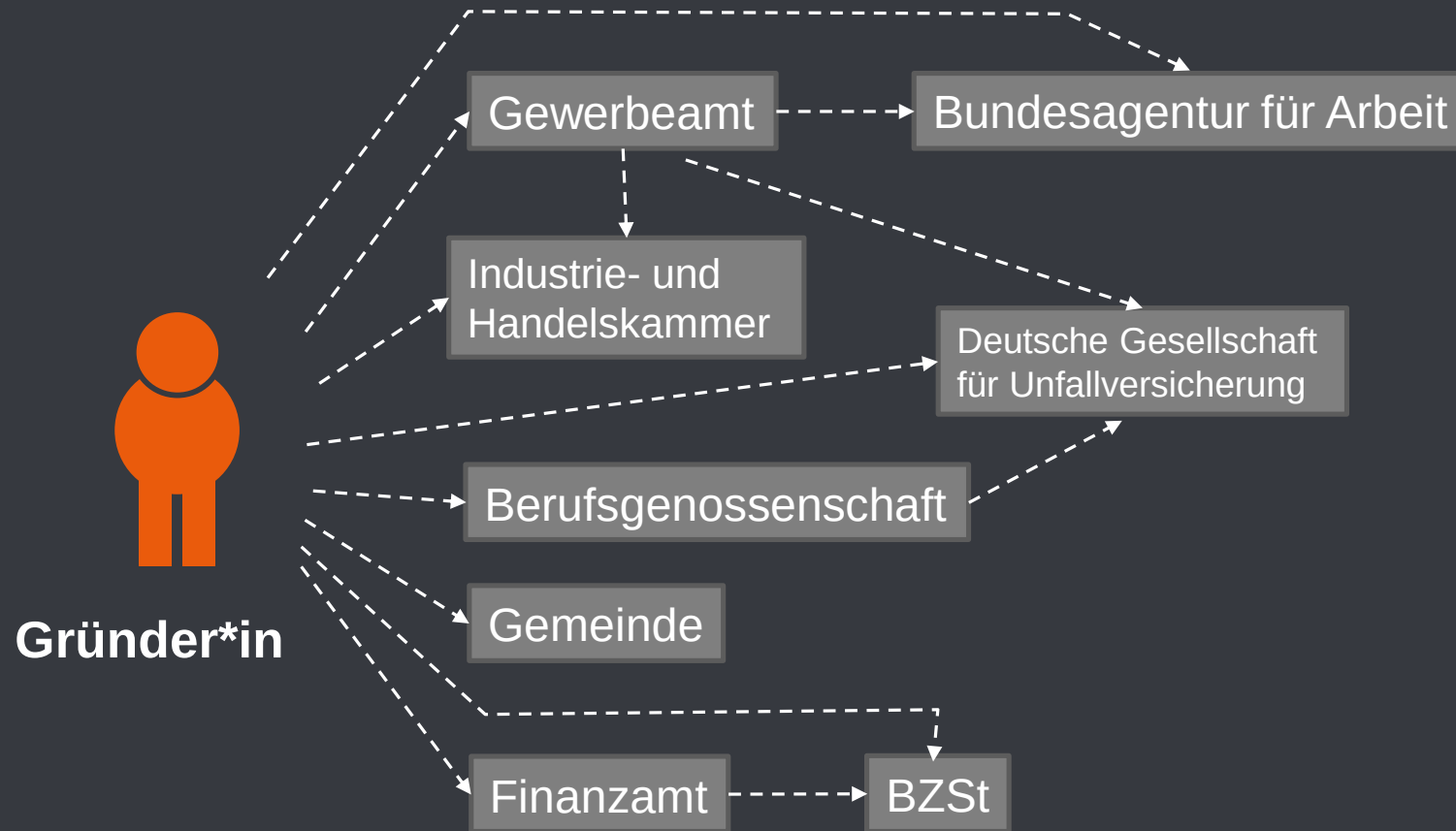
<https://www.youtube.com/watch?v=jm2XC7YmygE>

Ich werde Mitarbeiter und Mitarbeiterinnen einstellen, die wahrscheinlich nicht nur im Büro arbeiten, sondern auch mal Außendienst machen ... wenn die rumfahren ... wie kann ich die versichern? Bin ich da bei Ihnen richtig? Oder wo sollte ich sonst hingehen?



0:00 / 0:35





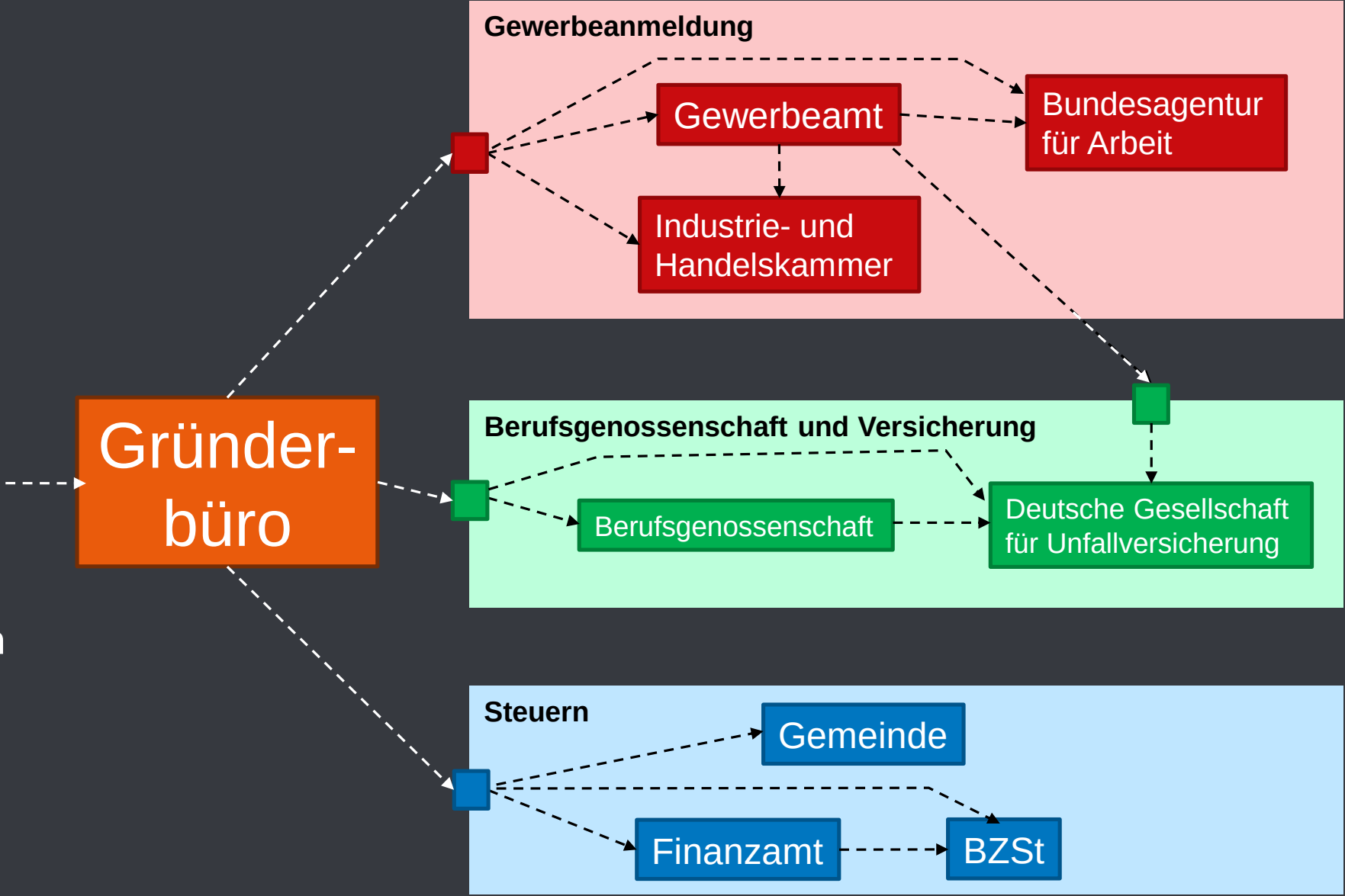
A middle-aged man with short brown hair, wearing a dark grey suit jacket, a light pink shirt, and a blue and white patterned tie. He is looking upwards and to the left with a thoughtful expression. A speech bubble is positioned to his left, and another is at the bottom right.

Startup,
10 Mitarbeiter

...

Verstanden. Ich
kümmere mich.

Gründer*in



Open-Closed-Principle

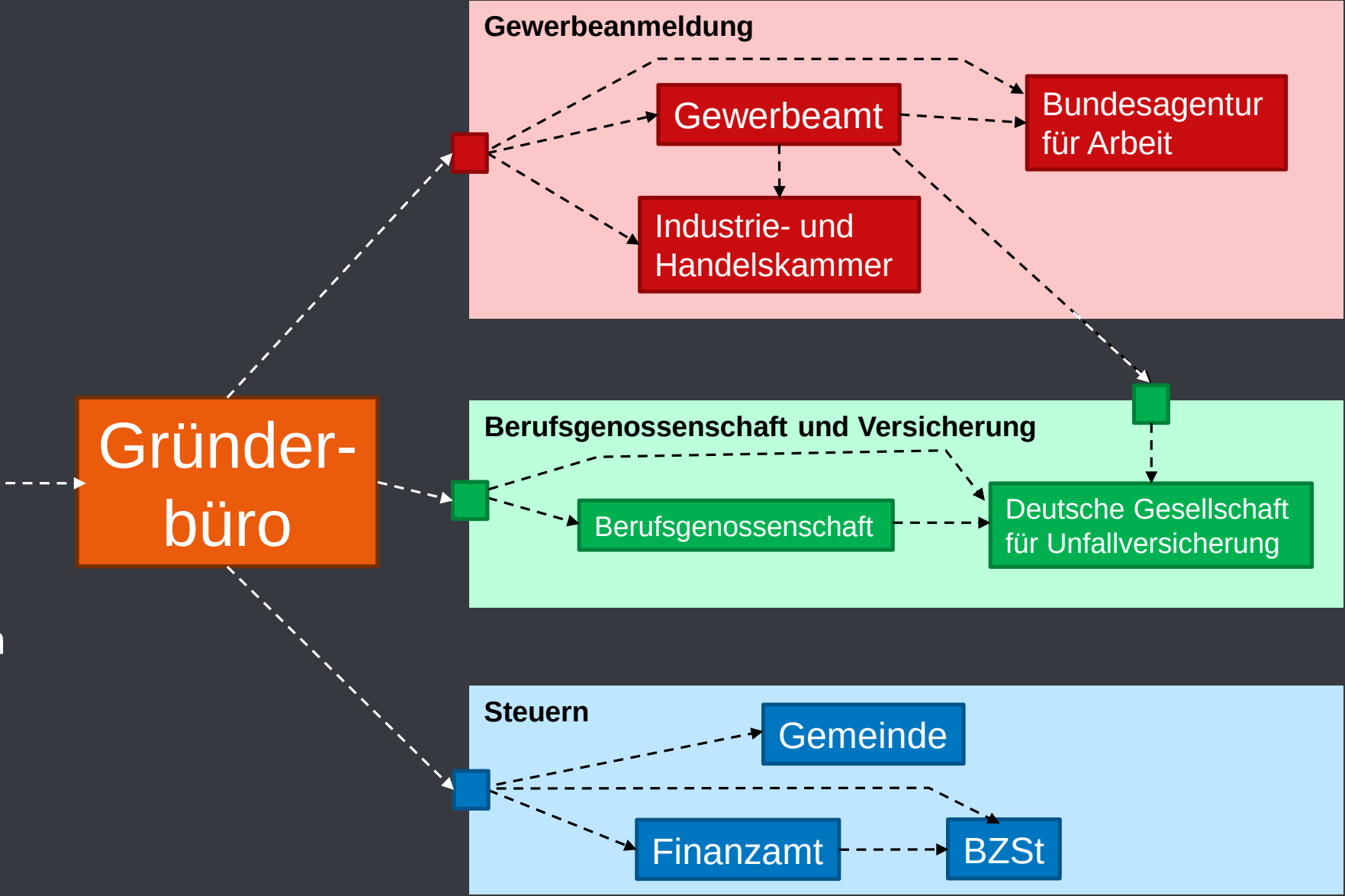
- Open for **extension**
 - leicht zu ändern und zu erweitern
- Closed for **modification**
 - hat kontrollierten Zugang



Lose Kopplung, hohe Kohäsion

1. kohärentes Ganzes mit 1 Aufgabe
(**hohe Kohäsion**)
2. hängt wenig von Anderem ab
(**lose Kopplung**)
3. nach außen: explizite, minimale,
delegierende Kapsel
(**Schnittstelle**)

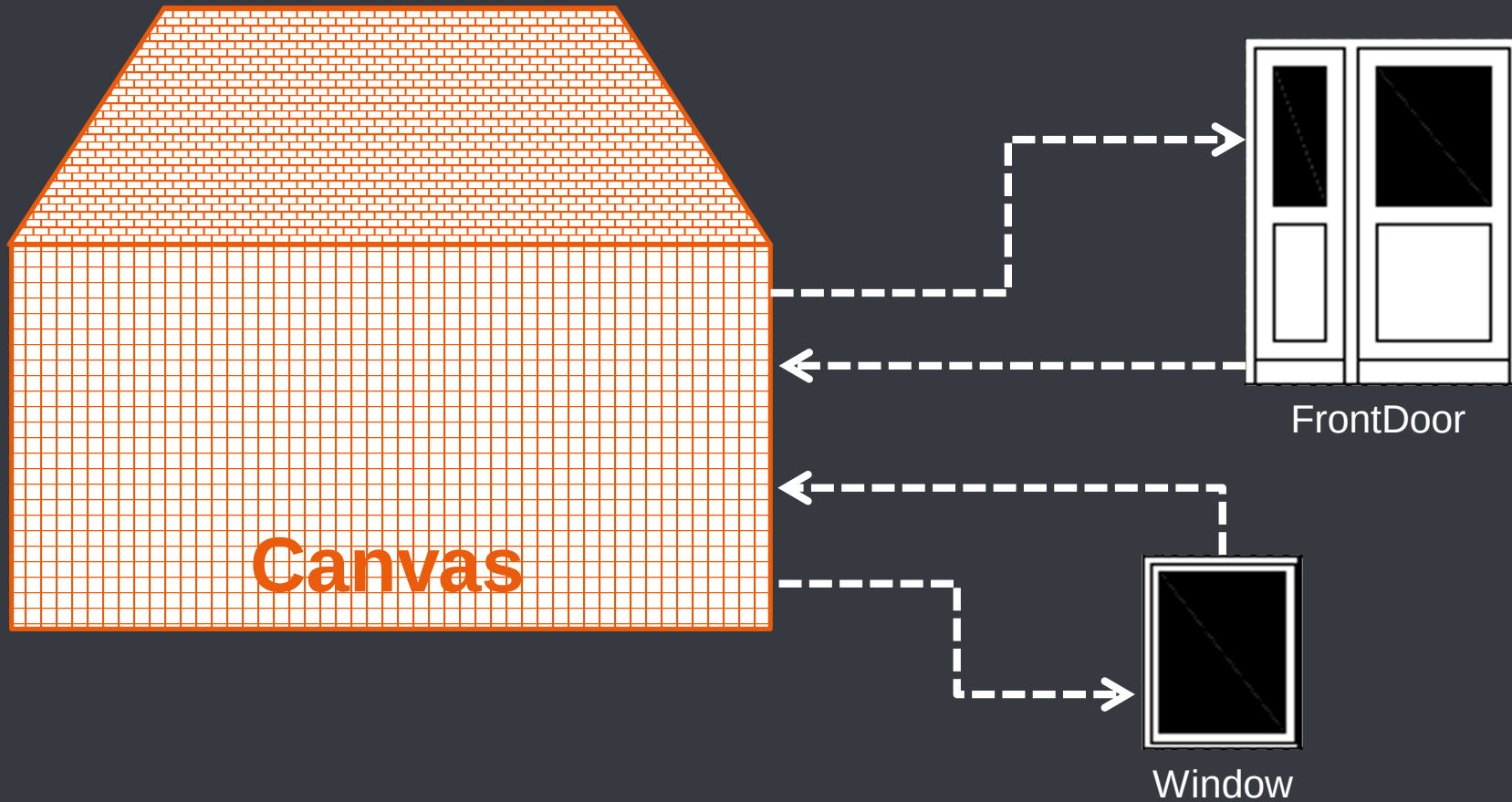
Gründer*in



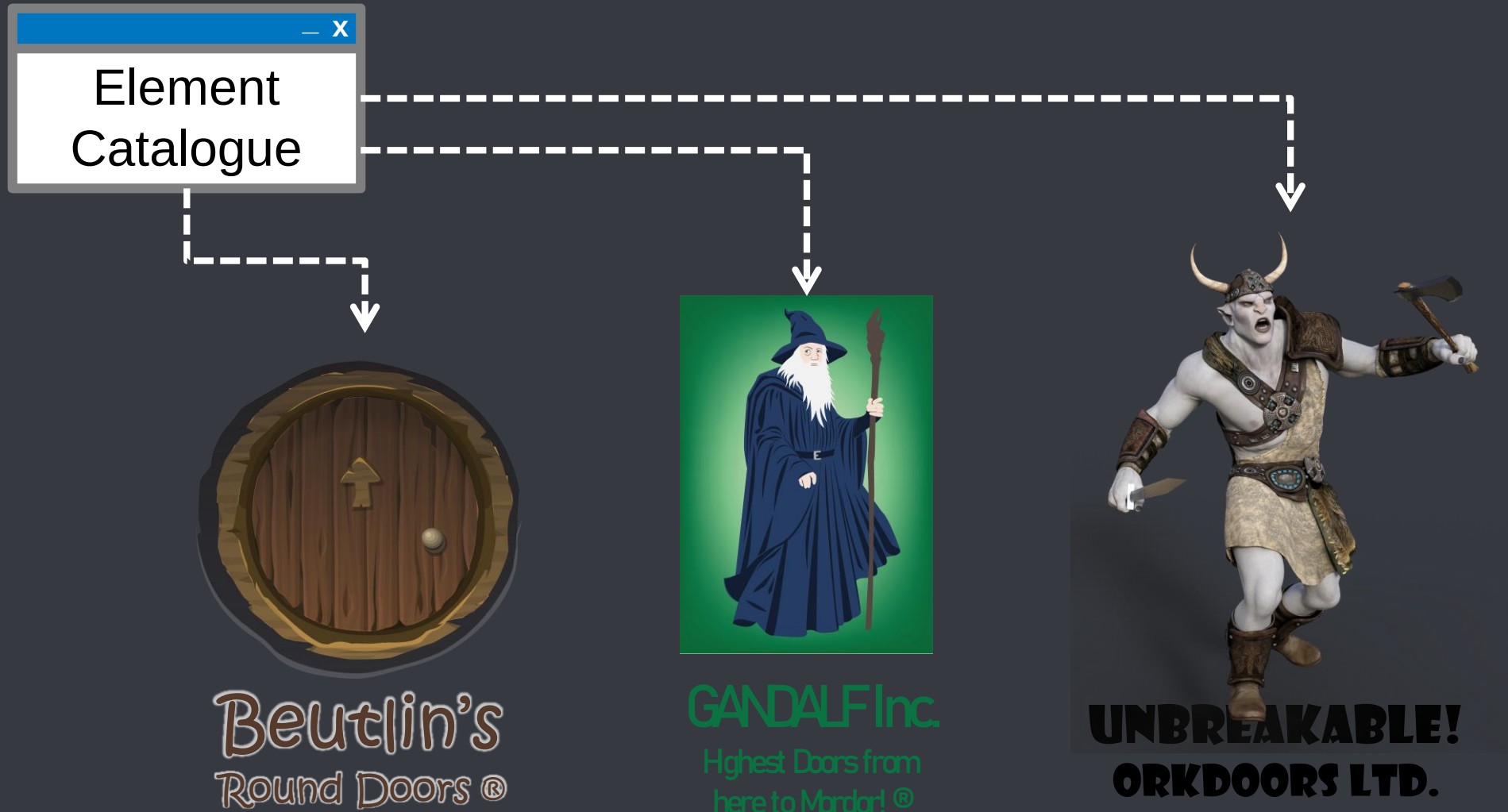
Dependency Inversion Principle



Unerwünschte Dependencies (1)



Unerwünschte Dependencies (2)



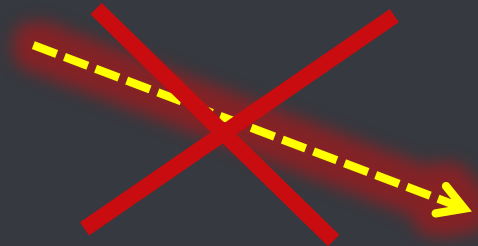
Dependency Inversion Principle

Source code dependencies
refer only to **abstractions**,
not to **concretions**

My
source
code

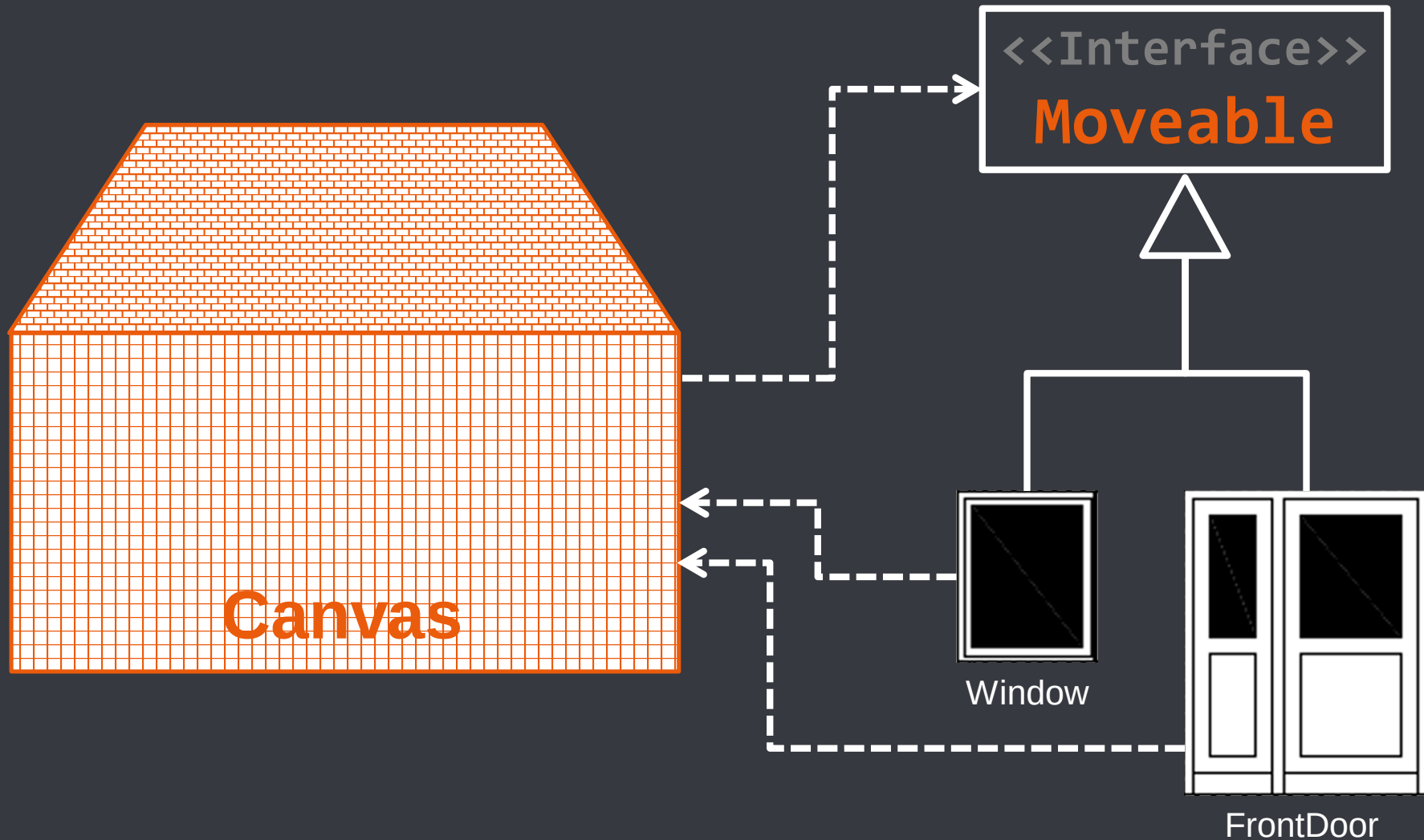


Abstraction

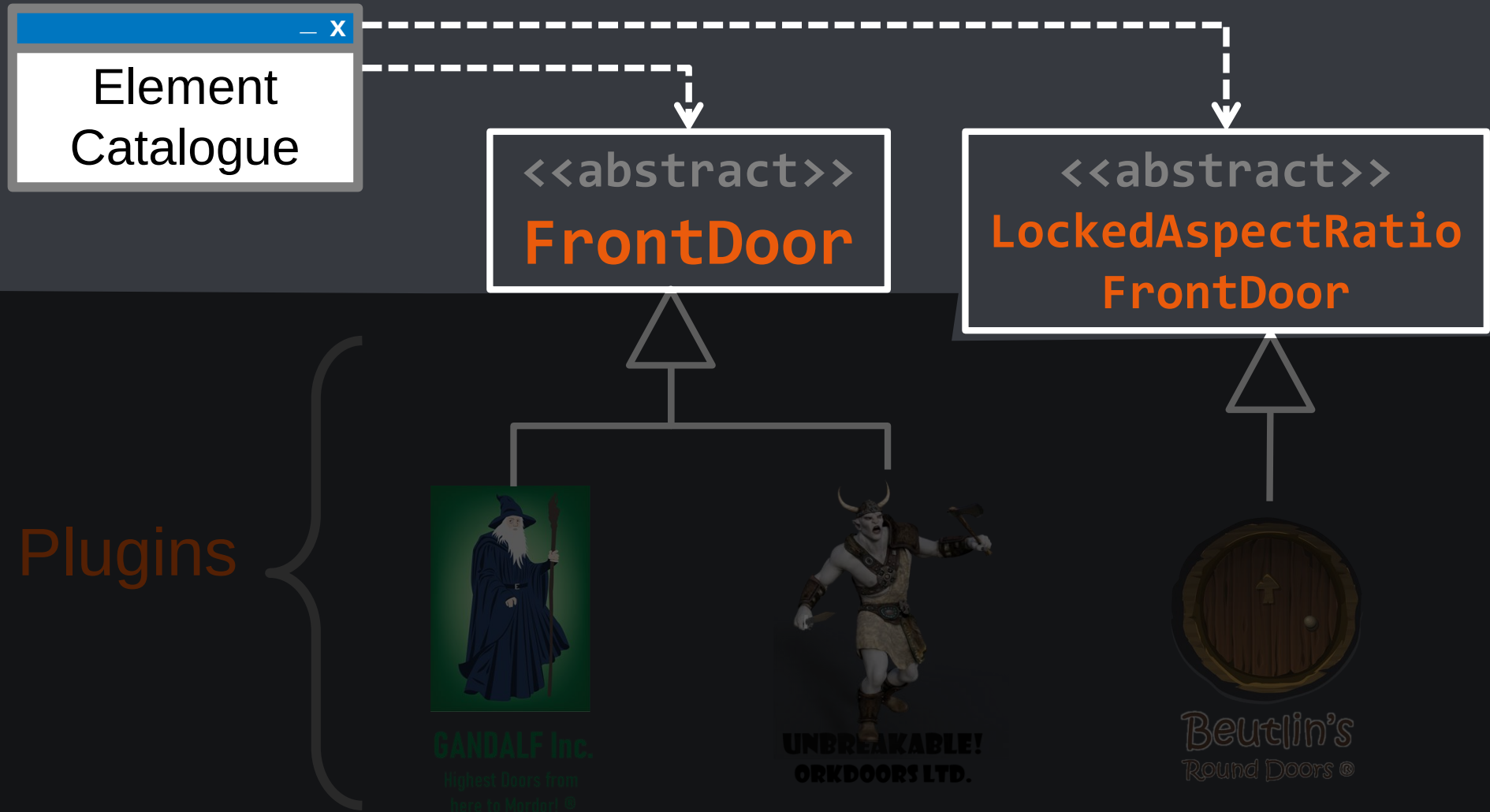


Concretion

Fall 1: Canvas + Elemente

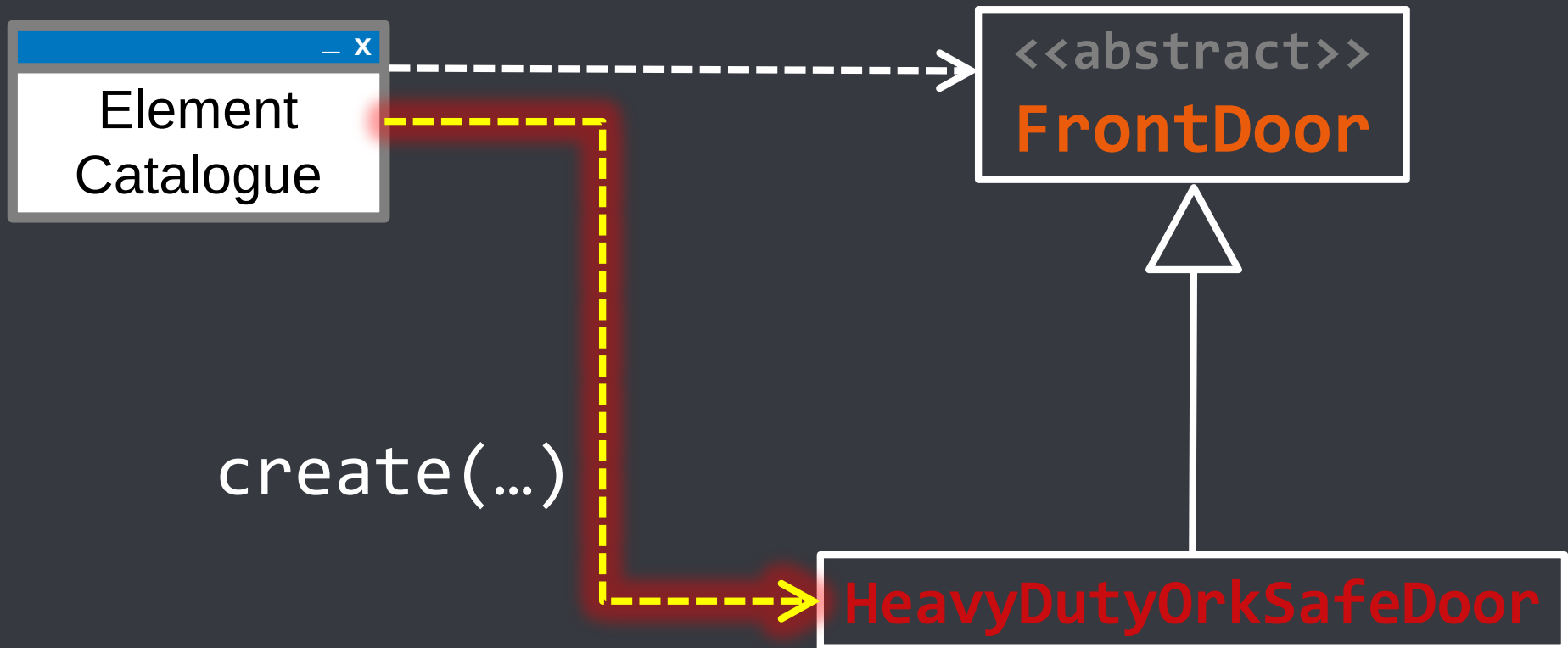


Fall 2: Konkrete Element-Typen



“Never mention the name of anything
concrete and volatile.”





```
public class ElementCatalogue {
```

```
//...
```

```
public void placeFrontDoor() {
```

```
//...
```

```
FrontDoor frontDoor =
```

```
    new HeavyDutyOrkSafeDoor();
```

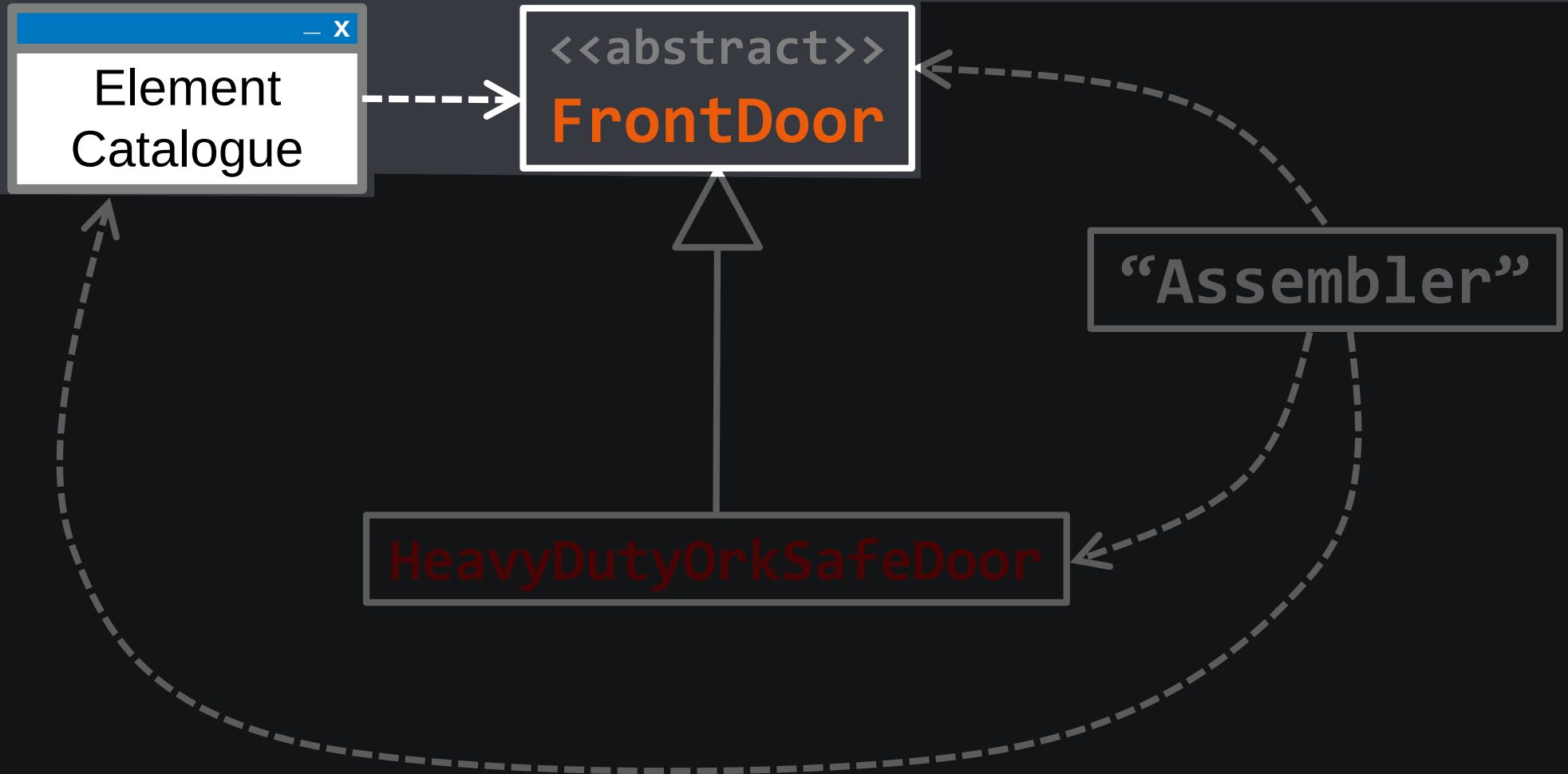
```
    mainCanvas.addElement( frontDoor );
```

```
}
```

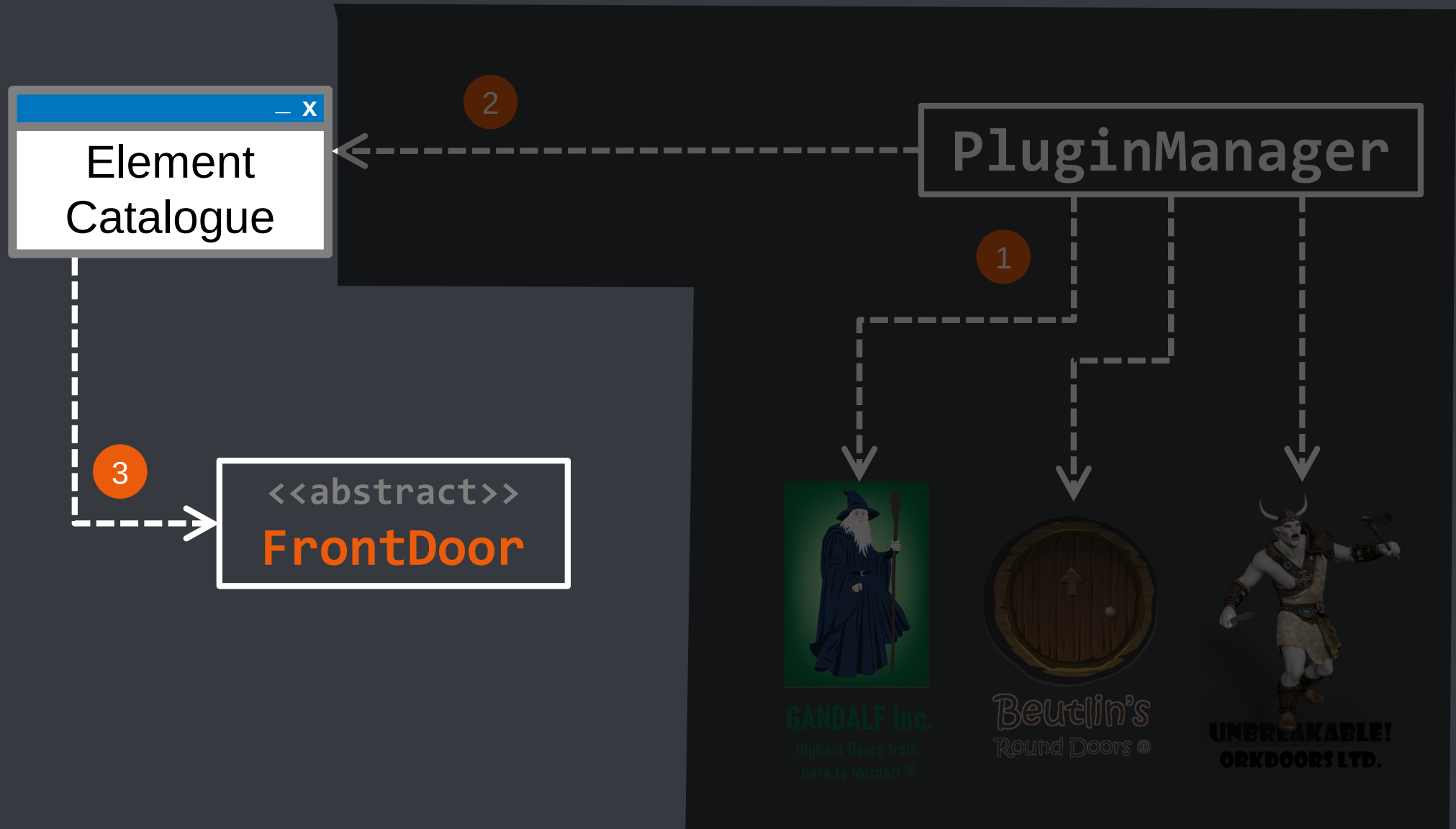
```
//...
```

```
}
```

Dependency Injection Pattern



Hier eher: Plugin-Manager als Assembler



Field Injection

```
//...
```

```
@Autowired
```

```
FrontDoor frontDoor;
```

```
public void placeFrontDoor_betterVersion() {
```

```
    //...
```

```
    mainCanvas.addElement( frontDoor );
```

```
}
```

```
//...
```



ArchiLab

www.archi-lab.io

© 2022 Prof. Dr. Stefan Bente