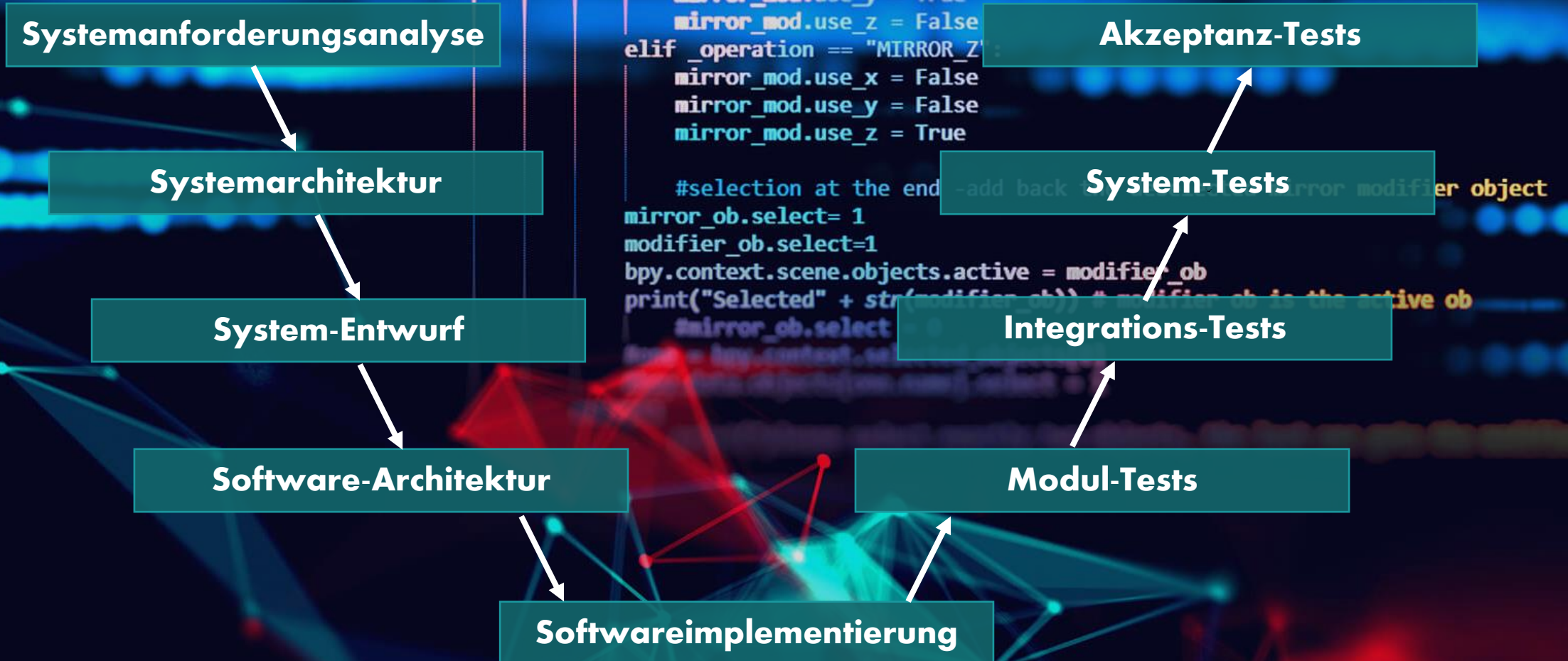


Selenium



Selenium

Selenium IDE



Selenium Webdriver



Protractor - eine Alternative?



```
elif _operation == "MIRROR_X":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
elif _operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True
```

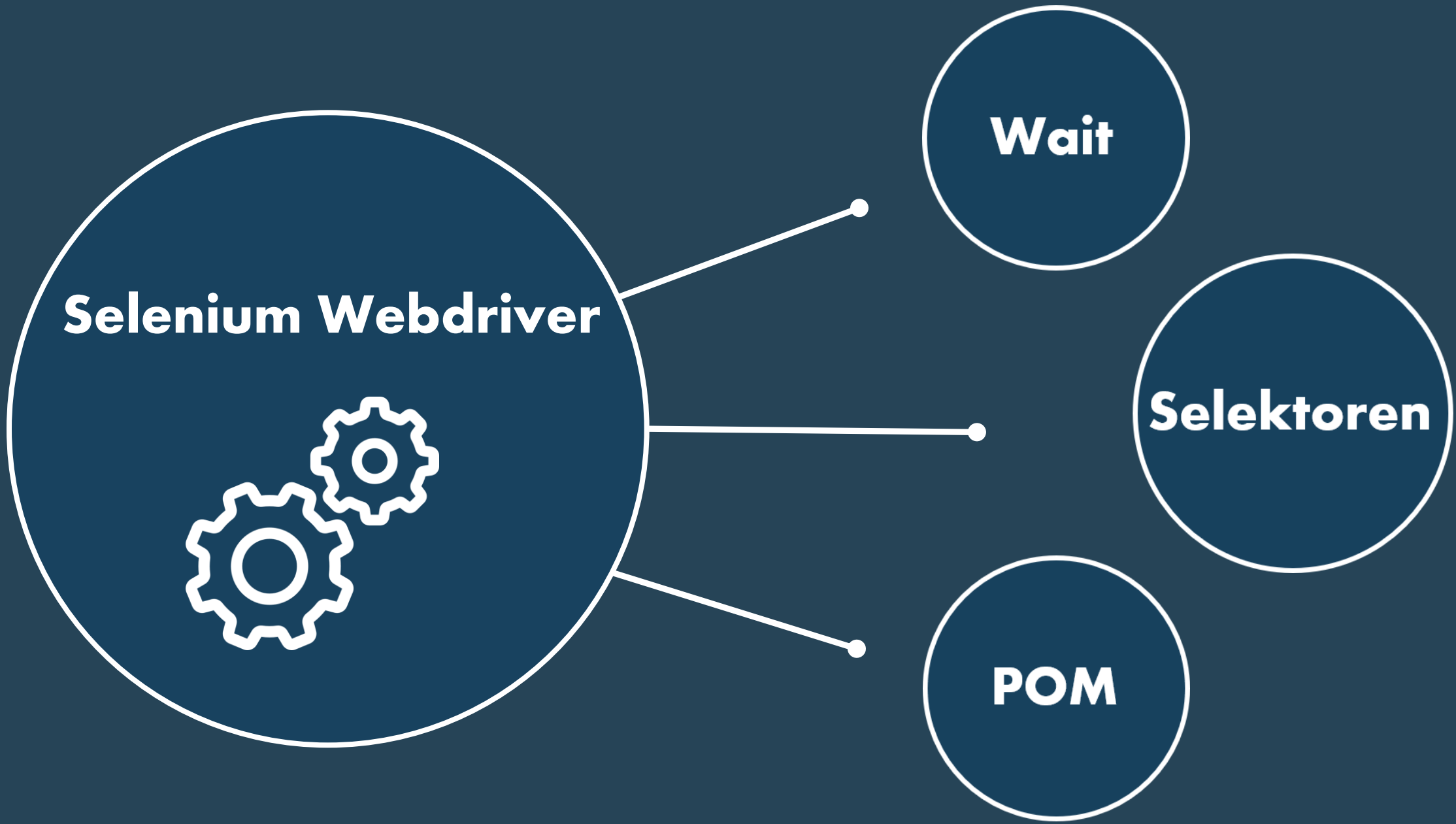
```
#selection at the end -add back the object
mirror_ob.select= 1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) #modifier_ob is active
```



Selenium IDE



- **Firefox Add-on**
- **Einfache Testerstellung**
- **Auf Firefox beschränkt**
- **Keine bedingten Statements**
- **Keine Schleifen**



Wait

- **Warten bis Element im DOM geladen ist**
- **Java-Selenium schneller als Browser**



Selektoren

- **Viele Arten von Selektoren**
- **XPath am wichtigsten**

```
//input[@class='gLfyf gsfi']
```



POM

- **Page Object Model - Designpattern**
- **Seite im Browser → Klasse in Java-Selenium**
- **Funktionen auf der Seite → Methoden in Klassen**



Protractor

- **Spezialisiert auf Angular**
- **Man erspart sich das Wait**
- **Verwendung von Selenium wird empfohlen**





**Danke für Ihre
Aufmerksamkeit**

```
elif _operation == "MIRROR_X":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = True  
    mirror_mod.use_z = False  
elif _operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True  
  
#selection at the end -add back the deselected mirror modifier object  
mirror_ob.select = 1  
modifier_ob.select = 1  
bpy.context.scene.objects.active = modifier_ob  
print("Selected" + str(modifier_ob)) # modifier ob is the active ob  
#mirror_ob.select = 1
```

Quellen

- <https://www.guru99.com/locators-in-selenium-ide.html>
- <https://www.softwaretestingmaterial.com/page-object-model/>
- <https://www.softwaretestingmaterial.com/page-object-model/>
- <https://www.seleniumhq.org/>
- <https://www.protractortest.org/#/>
- <https://www.heise.de/developer/artikel/End-to-End-Tests-mit-Protractor-2461535.html?seite=all>