



# GREEN WORKFLOW BY DESIGN

Java Forum Stuttgart 2021

Merlin Bögershausen, PSI Software AG

## Rules Engines – Was? Wie? Wo?

Rules Engines in Produkten einsetzen

# Wer spricht da?



Merlin Bögershausen



Senior Software Entwickler – PSI Software AG



MBoegershausen@psi.de

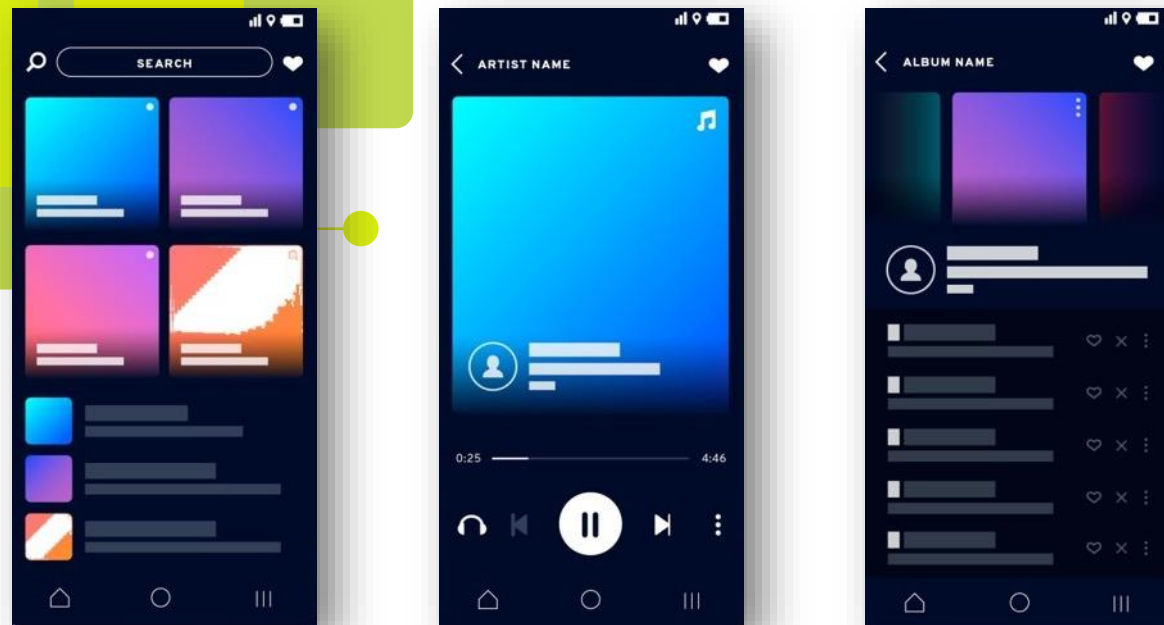


10 Jahre Java EE  
5 Jahre Produktentwicklung

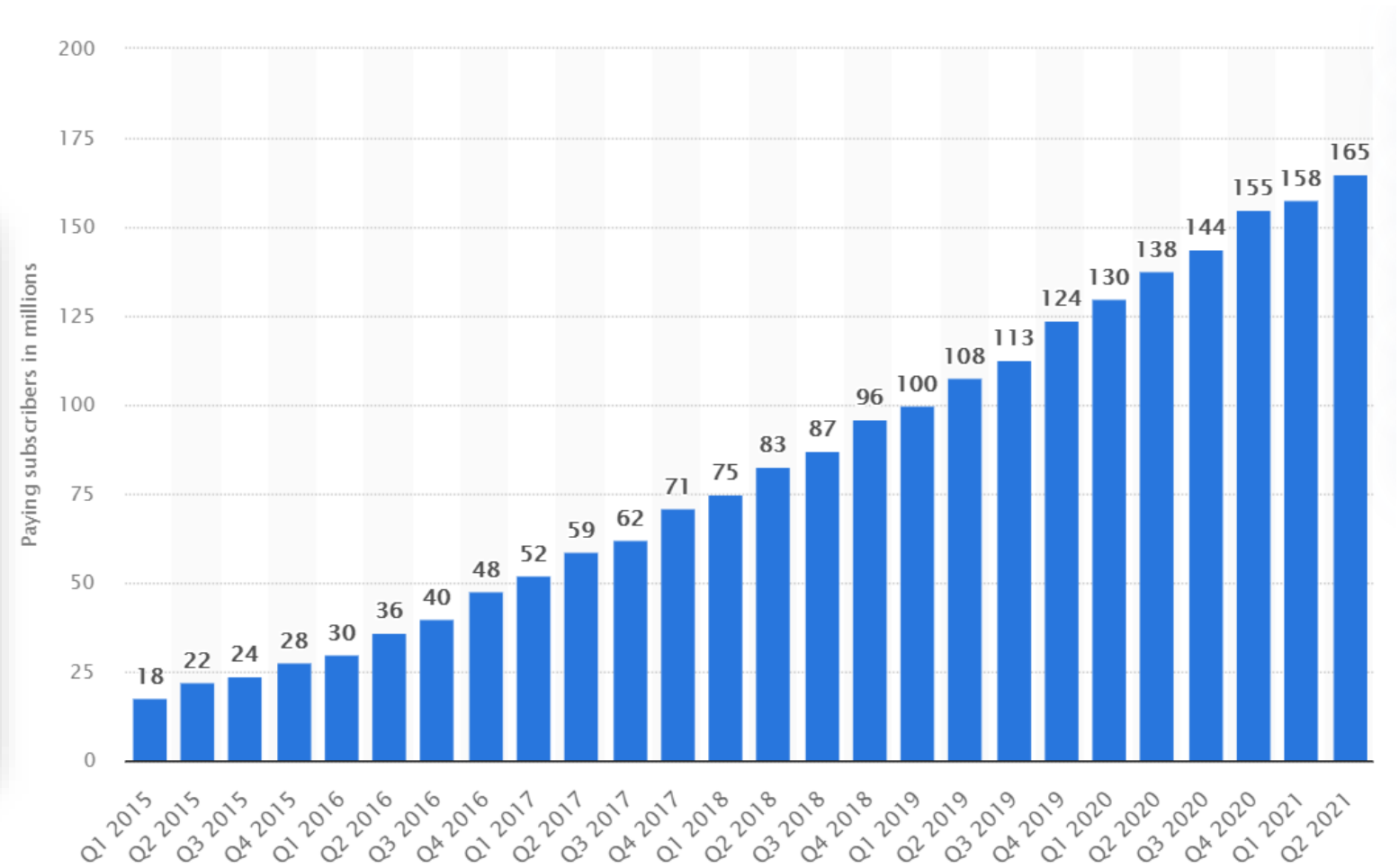
# Welche Ziele haben Firmen?



# Ziel: Sei wie Spotify!

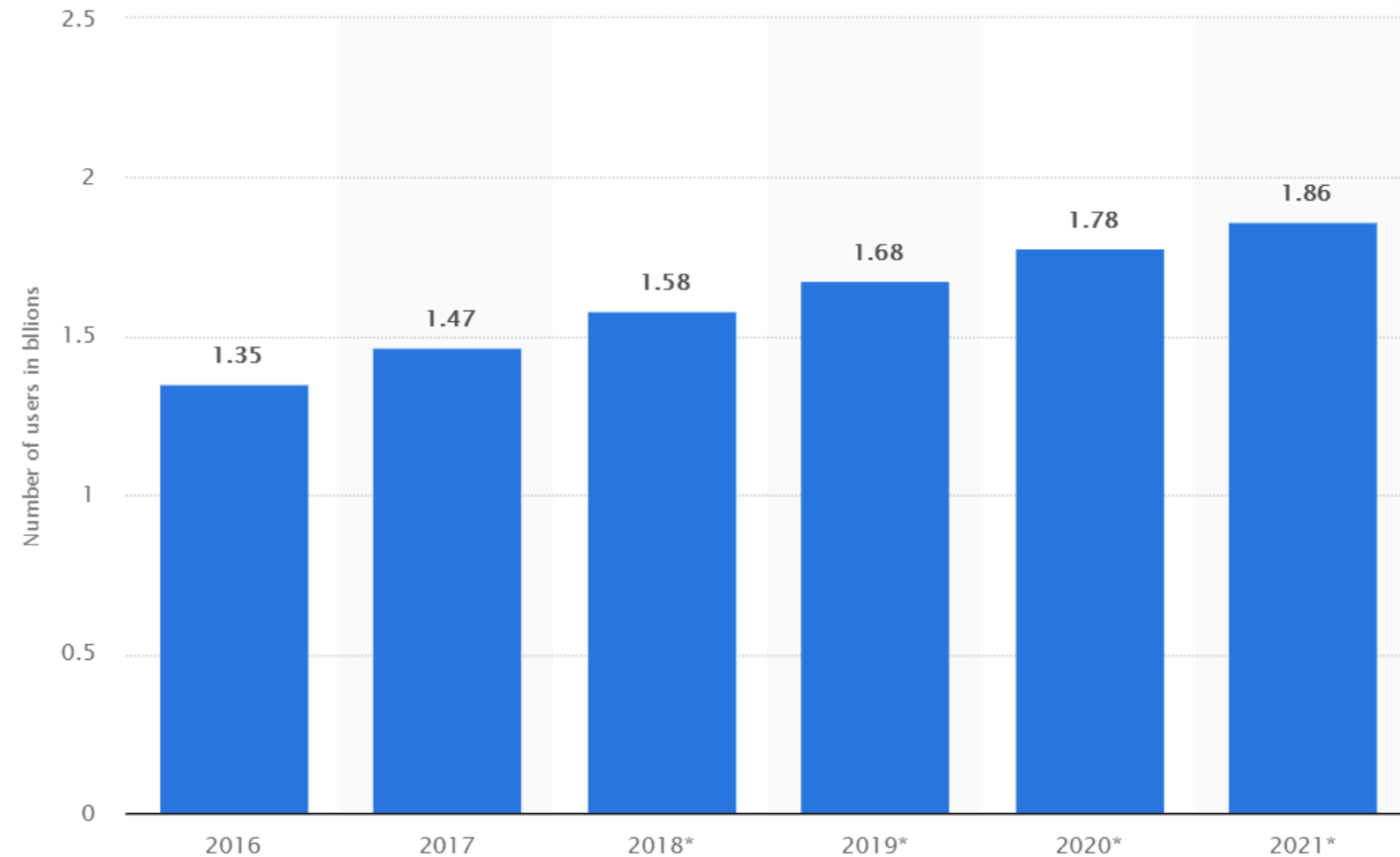
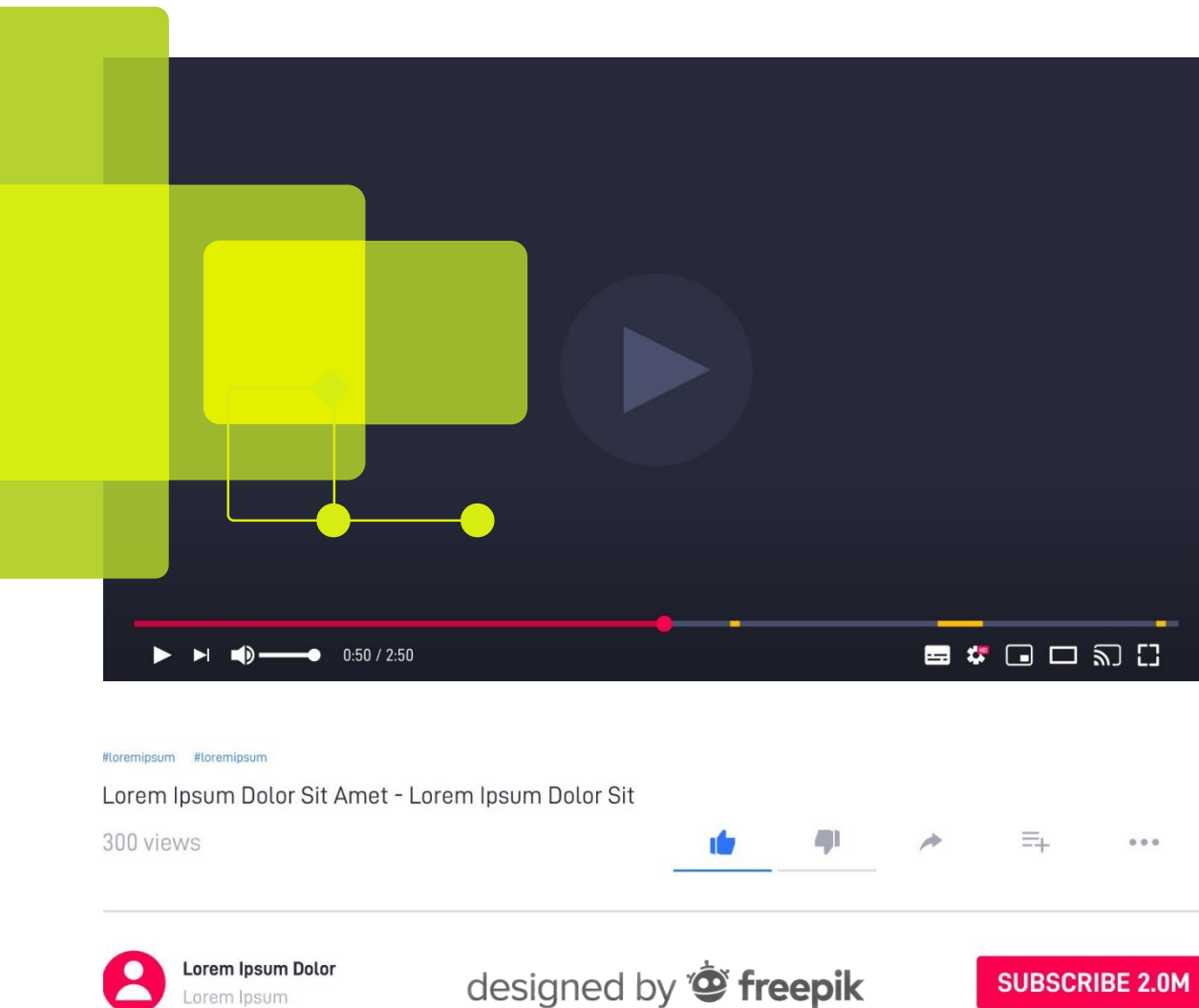


designed by  freepik



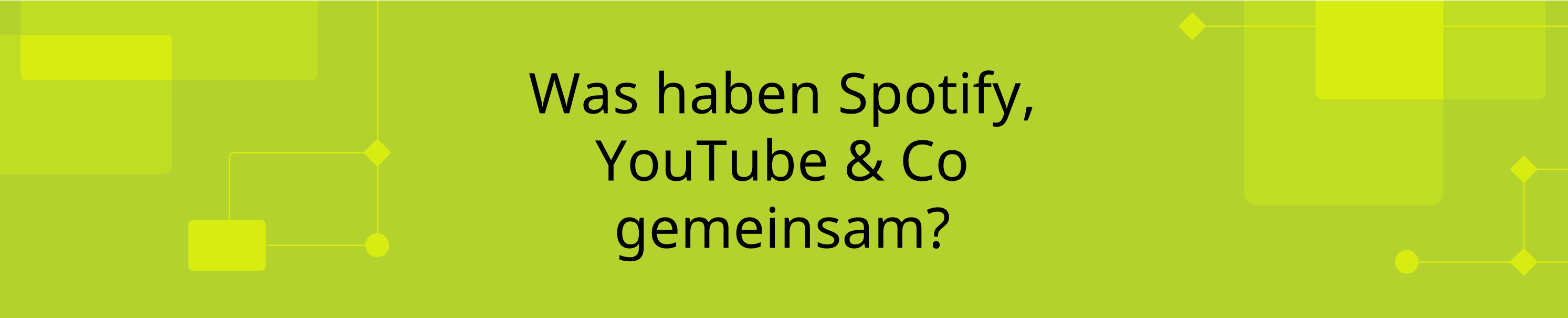
Statista: Number of Spotify premium subscribers worldwide

# Ziel: Sei wie YouTube!



Statista: Number of YouTube users worldwide from 2016 to 2021

Passt das auch für die anderen 95%?



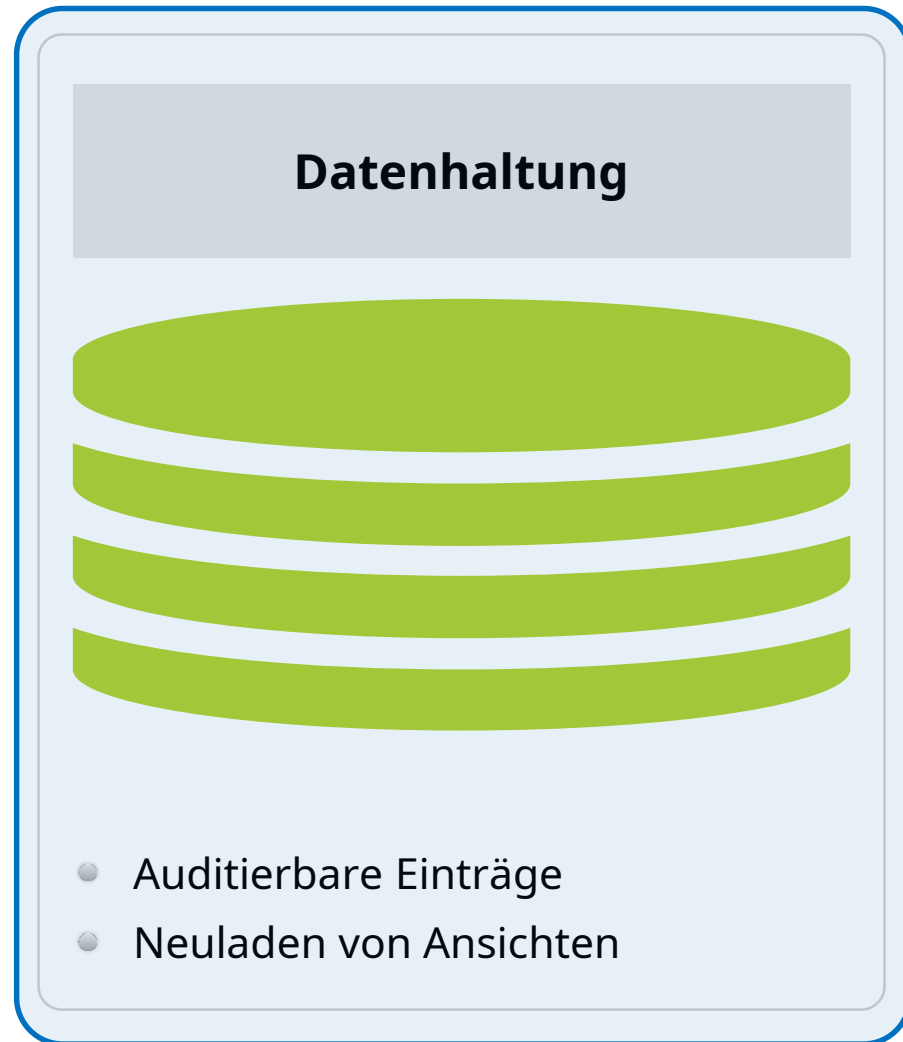
Was haben Spotify,  
YouTube & Co  
gemeinsam?

Viele Kunden mit den selben Anforderungen



Nach Geeks for Geeks

# Beispiele aus der Logistik



Persistenzlayer



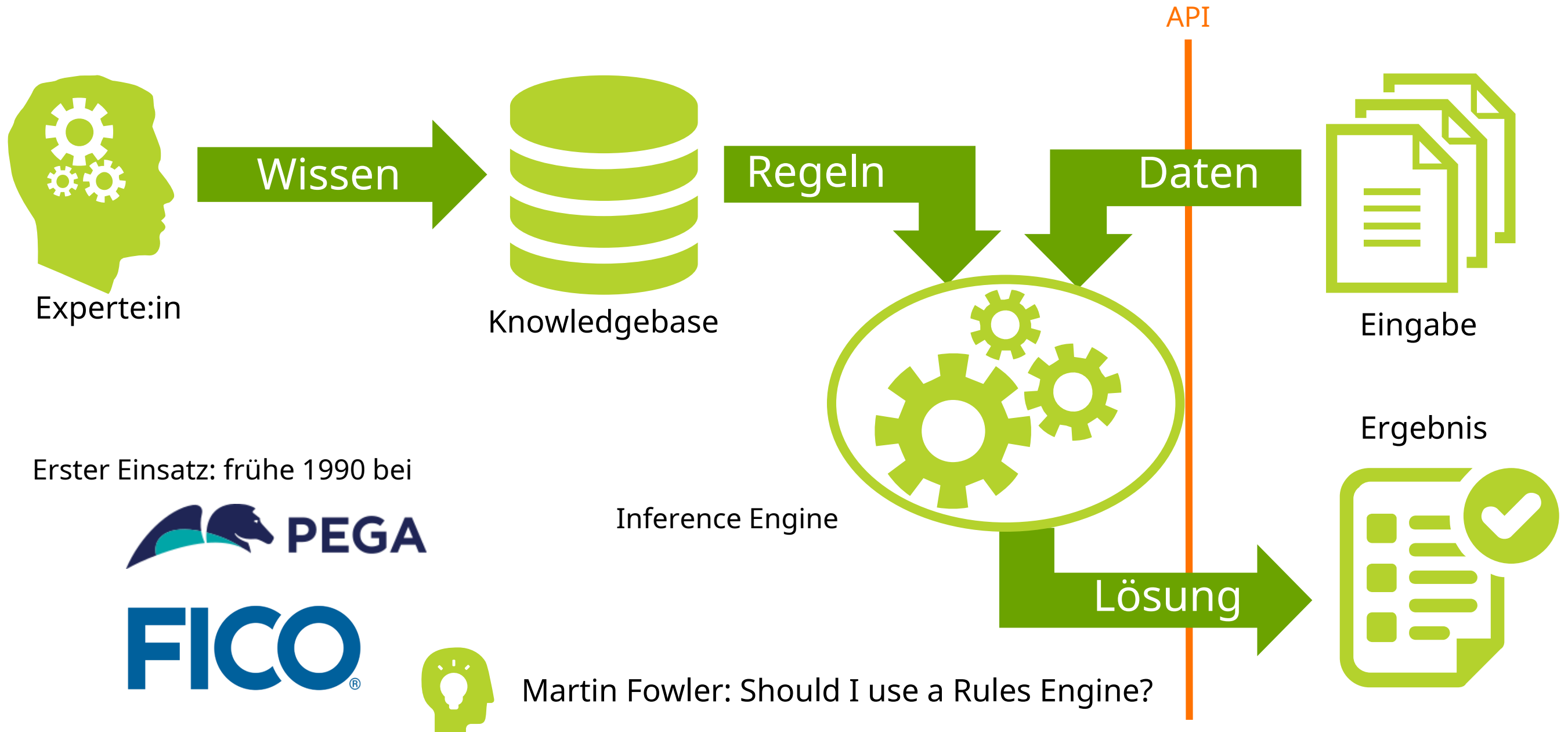
Servicelayer

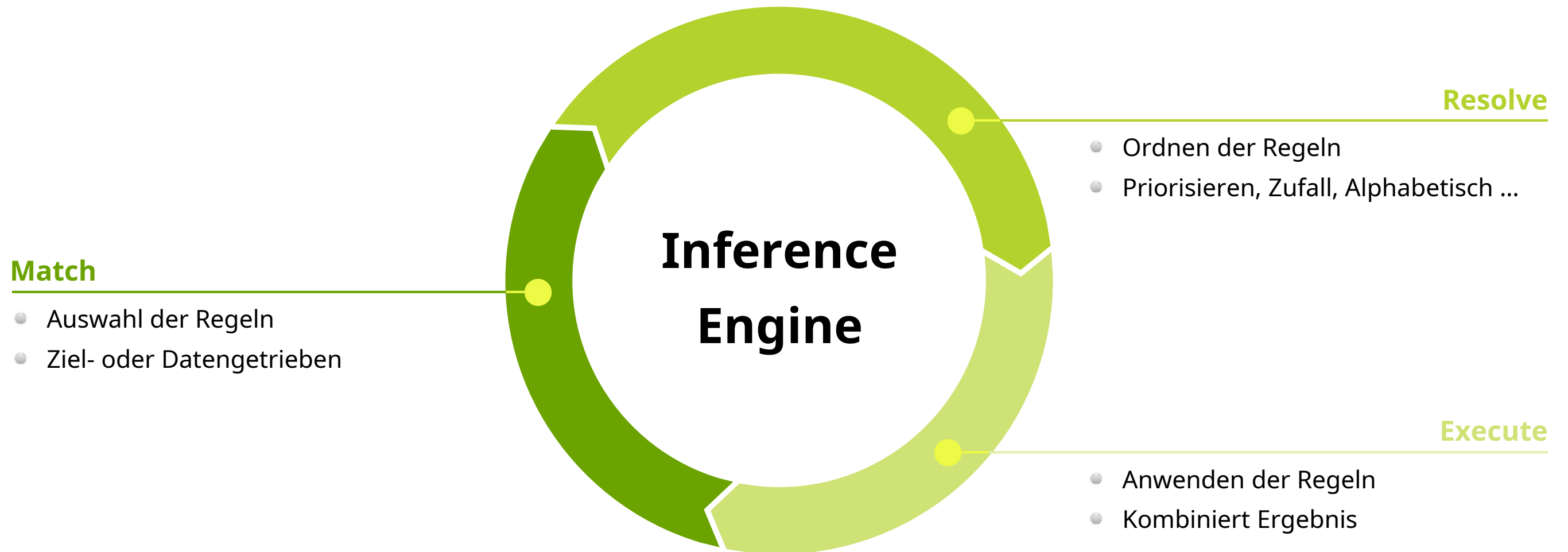


Servicelayer die Zweite???



# Komponenten einer Rules Engine





InfoQ: Real-World Rule Engines







## Red Hat Drools

- Regel Definitionen UI Dialog
- Sehr mächtig und einfach verständlich
- Spring, Aries Blue und Camel Integration
- [www.drools.org](http://www.drools.org)

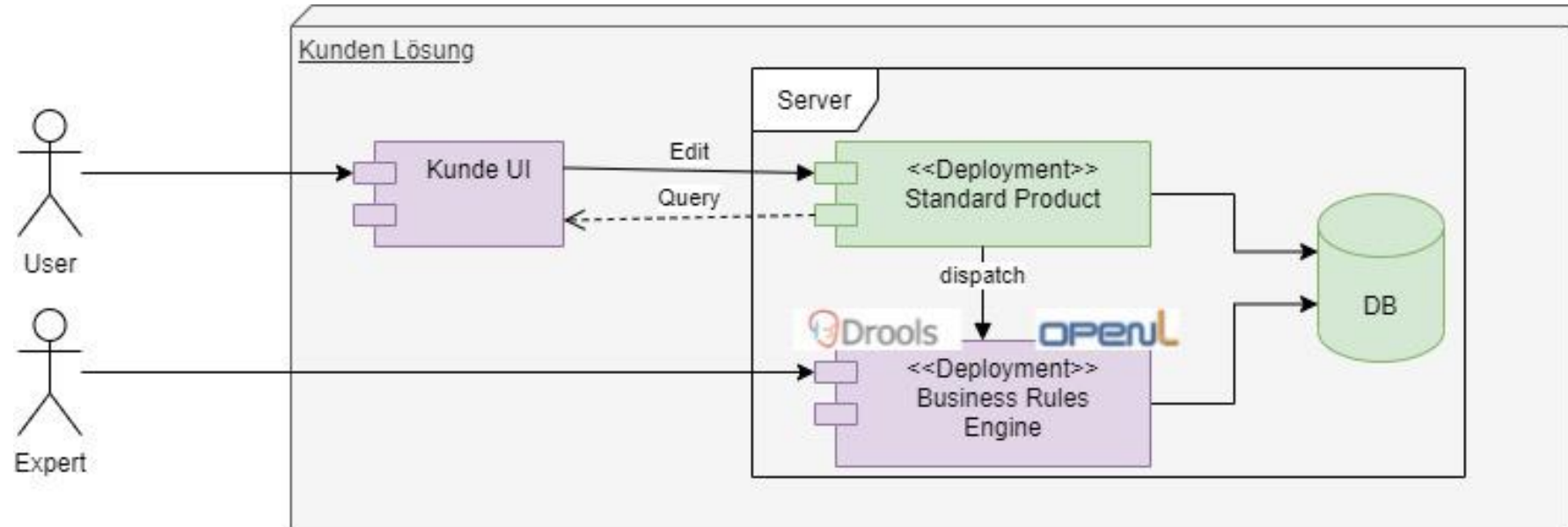
basic_fees				
#	Description	Patient is born		Patient has to pay a fee [\$]
		\$bornAfter	\$bornBefore	\$amount
1			01-Jan-2000	0
2		01-Jan-2010	01-Jan-2018	20
3		01-Jan-2018		50

## OpenL Tablets

- Definition Excel ähnliche Decissionables
- Sehr mächtig und einfach verständlich
- Standalone & NoCode Lösung
- [www.openl-tablets.org](http://www.openl-tablets.org)

 Edit	 Open	 Copy	 Remove	 Run	 Trace	 Benchmark	Target Table <a href="#">AirTicketsPrice</a>
Test AirTicketsPrice AirTicketsPriceTest							
departureCity	destinationCity				_res_		
Departure City	Destination City				Result		
Boston	Paris				850		
Chicago	Madrid				900		
San Diego	Minsk				700		

# Anwendung von Business Rules Engine

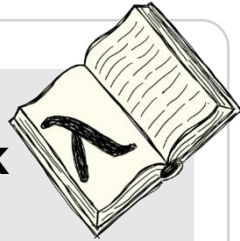


- Experte kann vom Kunden sein
- Seperates Deployment
- Integration über Rest/Camel/Spring JMS
- Produkt separat ausgerollt



- Low Code
- Kundenaufwand

## Delivered Technologies Rule Book



- Pure Java, Lambda oder Annotation
- Spring Integration
- Given-When-Then Pattern

```
RuleBuilder.create()
    .withResultType(String.class)
    .withFactType(String.class)
    .when( fact -> "Rain".equals(
        fact.getStrVal( key: "Forecast")))
    .then( (fact, result) ->
        result.setValue("Umbrella"))
    .build();

@Rule
public class ItRains {
    @Given("Forecast")
    private String forecast;
    @Result
    private String accessory;
    @When
    public boolean when() {
        return "Rain".equals(forecast);
    }
    @Then
    public RuleState then() {
        accessory = "Umbrella";
        return RuleState.BREAK;
    }
}
```

<https://github.com/deliveredtechnologies/rulebook>

## JEasy Easy Rules

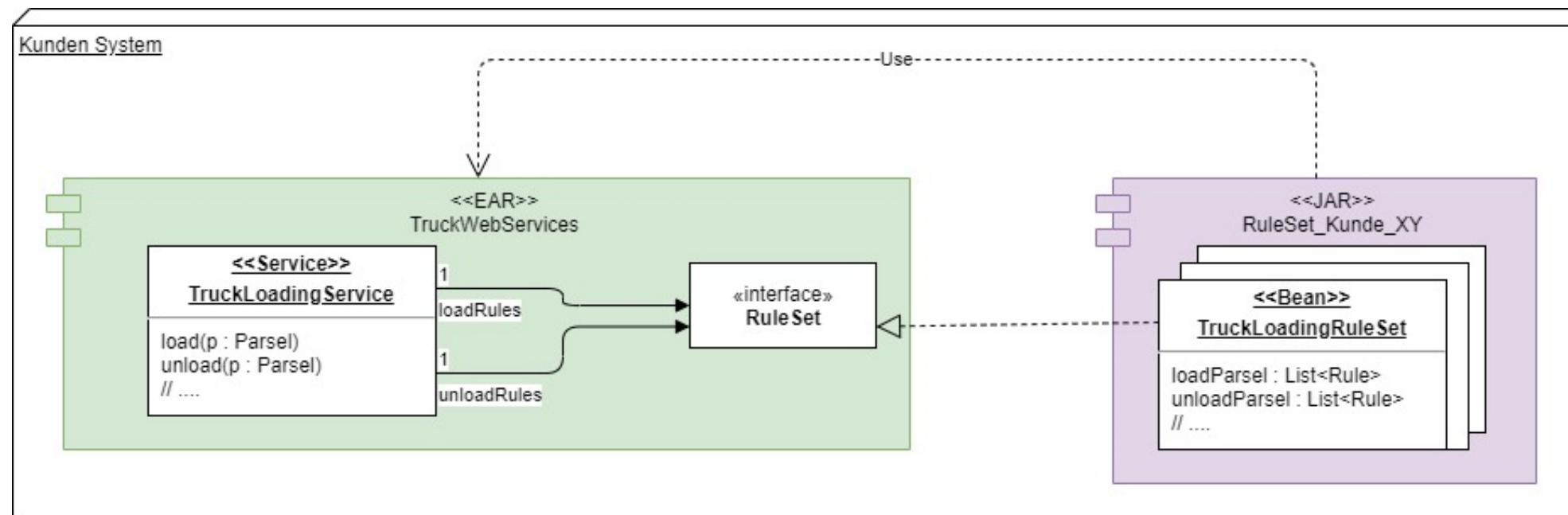
- Java Lambda oder Annotation
- Rule Descriptor Language
- If-Then Pattern

```
@Rule(name = "weather rule",
    description = "if it rains then take an umbrella")
public class WeatherRule {
    @Condition
    public boolean itRains(@Fact("rain") boolean rain) {
        return rain;
    }
    @Action
    public void takeAnUmbrella() {
        ake an umbrella!");
    }
}

name: "weather rule"
description: "if it rains then take an umbrella"
condition: "rain == true"
actions:
- "System.out.println(\"It rains, take an umbrella!\");"
```

<https://github.com/j-easy/easy-rules>

# Anwendung von Business Rules Engine



- Kundenspezifika ausgelagert
- Verschmelzen im Build
- Integration über POJO/Spring
- Entwickler schreiben Regeln



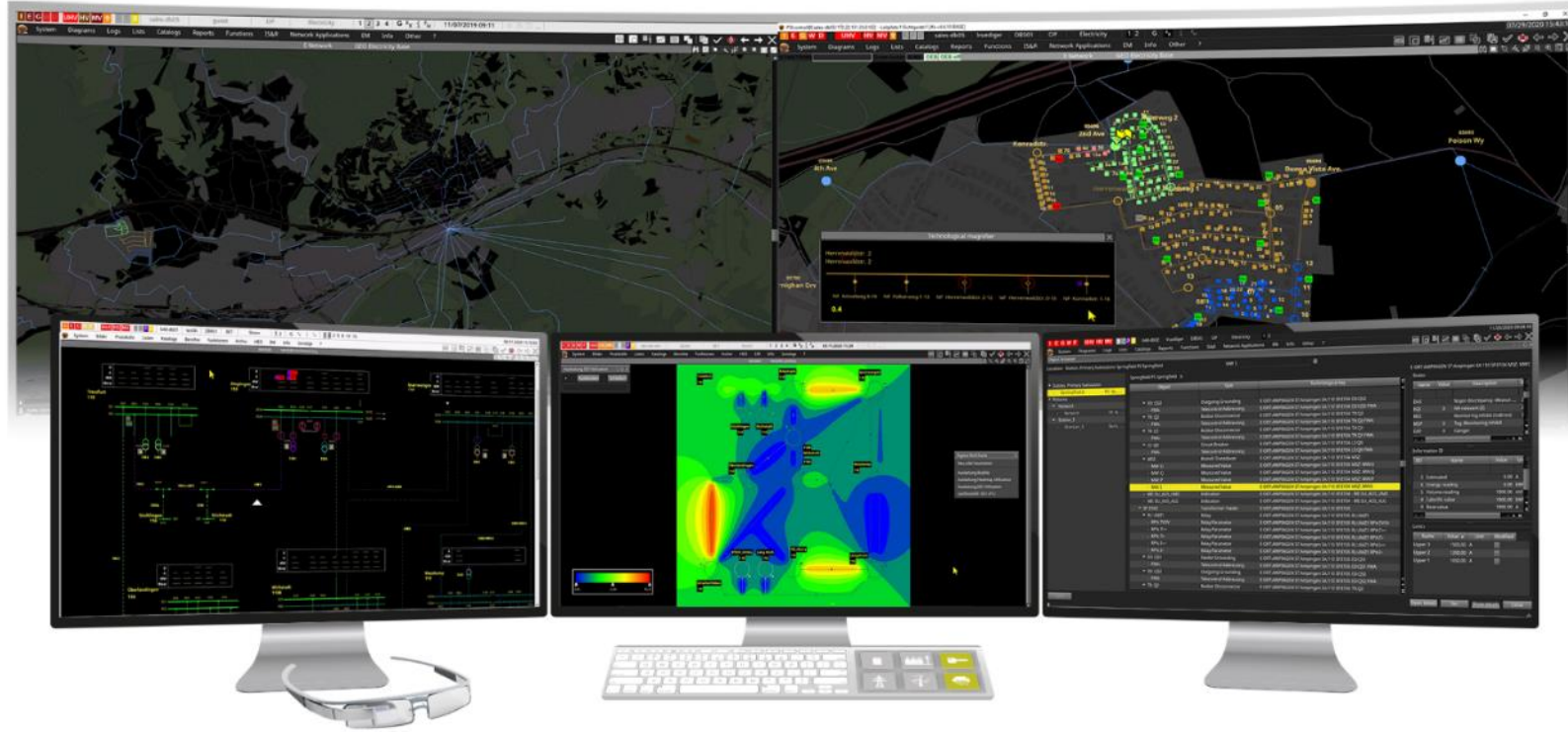
- Wartbarkeit
- Wiederverwendbarkeit
- Verständlichkeit

# Rules Engines zusammengefasst



- Separation
- Spezifische Anforderungen
- List von Regeln
- If-Then bzw. Given-When-Then
- Business Rules Engine
- Java Rules Engine
- Produktanpassungen
- Regel basierten Systemen





## Lust auf Produkte in Energieversorgung, Produktion und Infrastruktur?

Wir suchen [psi.de/psi-career](https://psi.de/psi-career)

- Softwareentwickler für unsere Produkte
- Berater für unsere Kunden
- Projektleiter bei der Implementierung





# Referenzen

- Martin Fowler: Should I use a Rules Engine? <https://martinfowler.com/bliki/RulesEngine.html>
- InfoQ: Real-World Rule Engines <https://www.infoq.com/articles/Rule-Engines/>
- Baeldung: List of Rules Engines in Java <https://www.baeldung.com/java-rule-engines>
- Oracle: Business Rules User's Guide [https://docs.oracle.com/cd/E12483\\_01/web.1013/b28965/jsr94.htm](https://docs.oracle.com/cd/E12483_01/web.1013/b28965/jsr94.htm)
- Geeks for Geeks: Software Engineering | Classification of Software Requirements <https://www.geeksforgeeks.org/software-engineering-classification-of-software-requirements/>
- Statista: Number of YouTube users worldwide from 2016 to 2021 <https://www.statista.com/statistics/805656/number-youtube-viewers-worldwide/>
- Number of Spotify premium subscribers worldwide from 1st quarter 2015 to 2nd quarter 2021 <https://www.statista.com/statistics/244995/number-of-paying-spotify-subscribers/>
- Drools Business Rules Engine <https://www.drools.org/>
- OpenL Tables Business Rules Engine <https://openl-tablets.org/>
- Jeasy Easy Rules <https://github.com/j-easy/easy-rules>
- Rulebook <https://github.com/deliveredtechnologies/rulebook>



Direkter Kontakt



@MBoegie



mboegers



merlin-boegershausen

# VIELEN DANK

## FÜR IHRE AUFMERKSAMKEIT!