



# Business Rule Management Systeme und SOA

*Mike Schäfer, IPT*

**JUG'S**  
Java User Group Switzerland

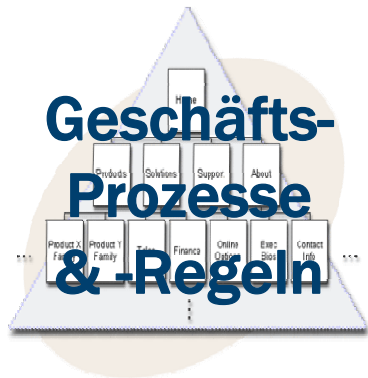
# Agenda

- **Motivation für BRMS**
- **Komponenten & Aspekte von BRMS**
  - **Regel-Editor & Fachsprachen**
  - **Regel-Repository & Verwaltung**
  - **Rule-Engine & Integration**
  - **Test & Analyse**
- **Algorithmen, Marktübersicht, Standards**



# Motivation

# Motivation



**Interne Abläufe**



**Wettbewerb**



**Markt**



**Gesetze und Richtlinien**







**Unvorhersehbares**

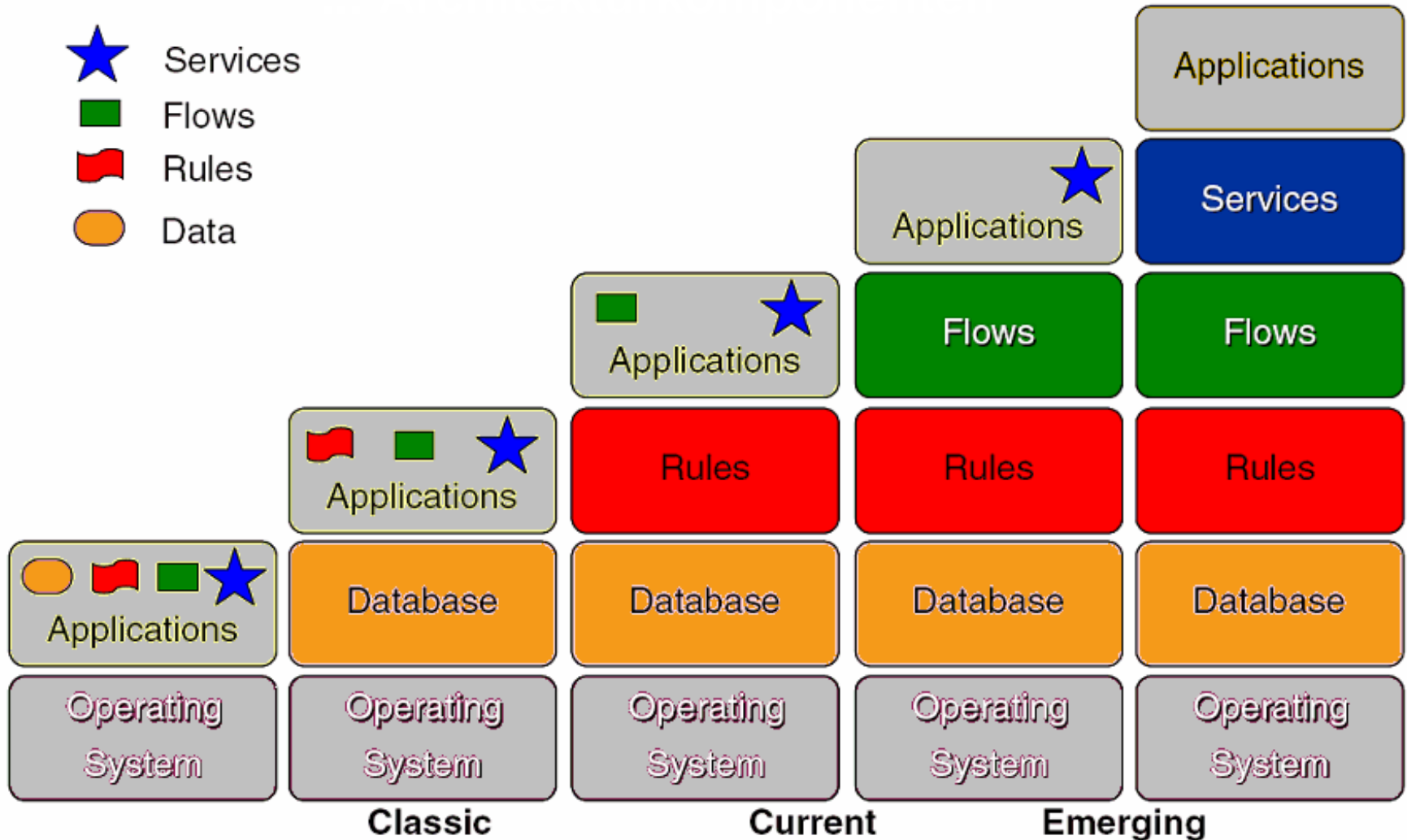
Nürnberg, 2. Januar 2004: Um sechs Uhr morgens liefen heute auf den Rechnern der Nürnberger DATEV die ersten Lohnabrechnungen nach dem neuen Steuertarif. ...

Nachdem dieser Plan den Software-Herstellern am 19. Dezember übermittelt wurde, wuch die Anspannung reger Geschäftigkeit.

... "Hektik zum Jahresende ist für unsere Leute in der Lohn-Entwicklung wohl nichts Neues", sagt Wolfgang Schumacher, Direktor der Software-Entwicklung bei der DATEV. "In diesem Jahr kamen die notwendigen Informationen allerdings extrem spät", räumt er ein.

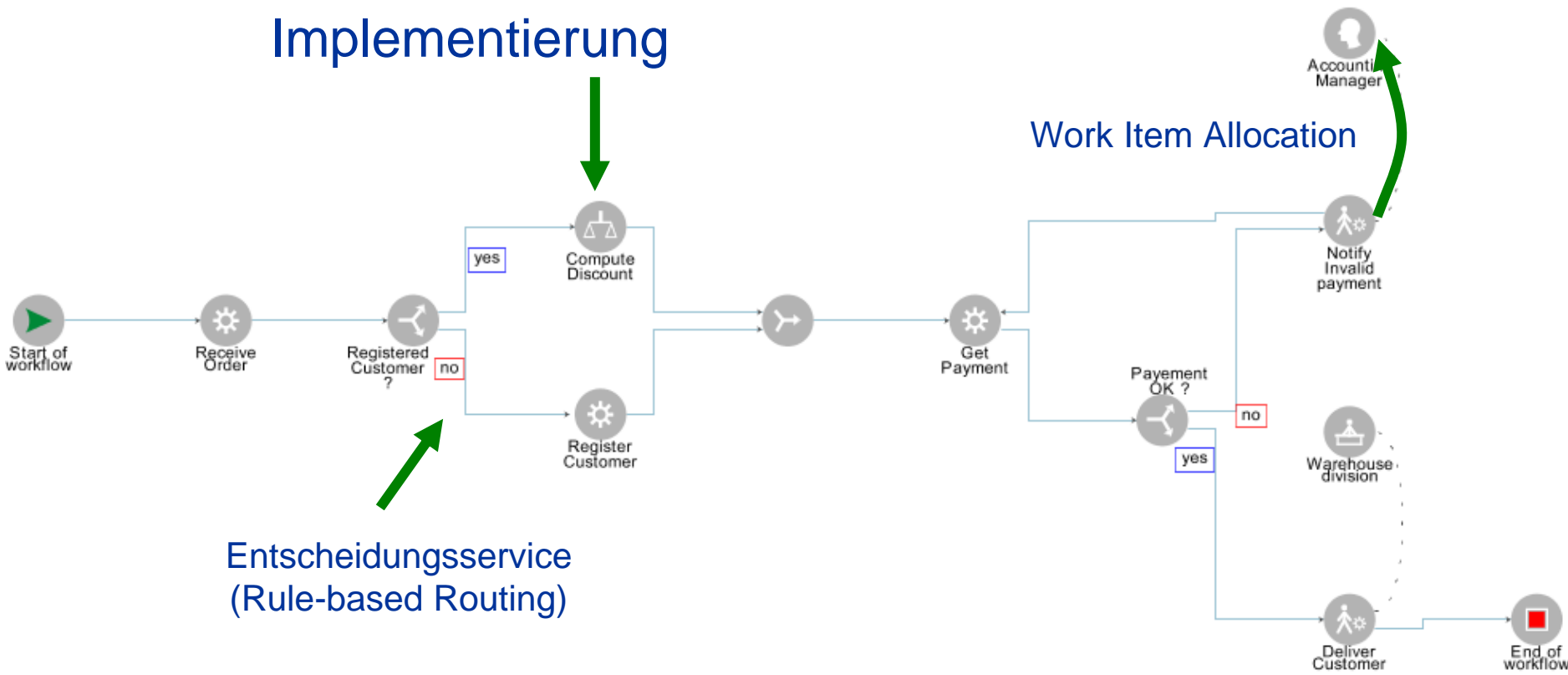
# Explicit Services, Flows & Rules

-  Services
-  Flows
-  Rules
-  Data



# Rules and BPM

## Service Implementierung

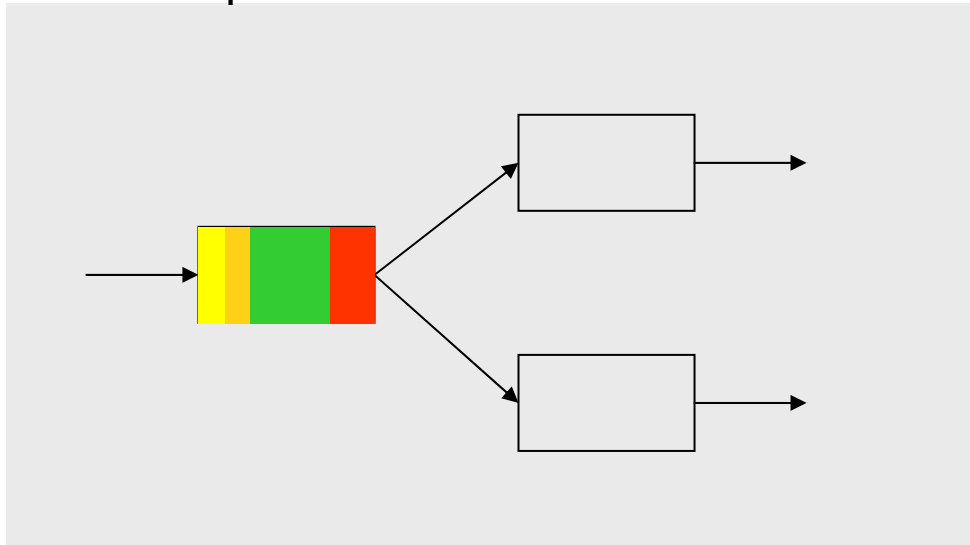


Entscheidungsservice  
(Rule-based Routing)

Work Item Allocation

# Regeltypen: Praxis

## Geschäftsprozess

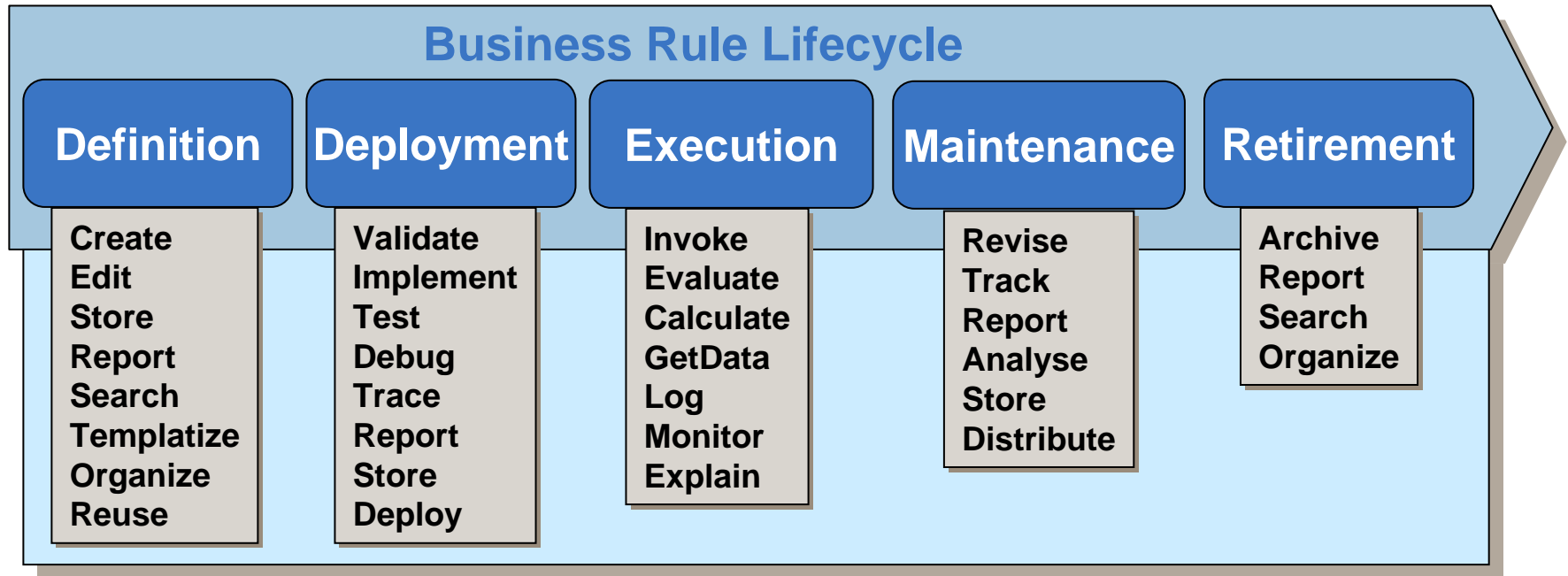


- Validierungsregeln: prüfen Inputdaten auf ihre Korrektheit
- Zuordnungsregeln: Mapping von Inputdaten
- Berechnungsregeln: erzeugen aus Inputdaten ein Ergebnis
- Steuerungsregeln: bestimmen aus Inputdaten und Ergebnis die weiteren Schritte

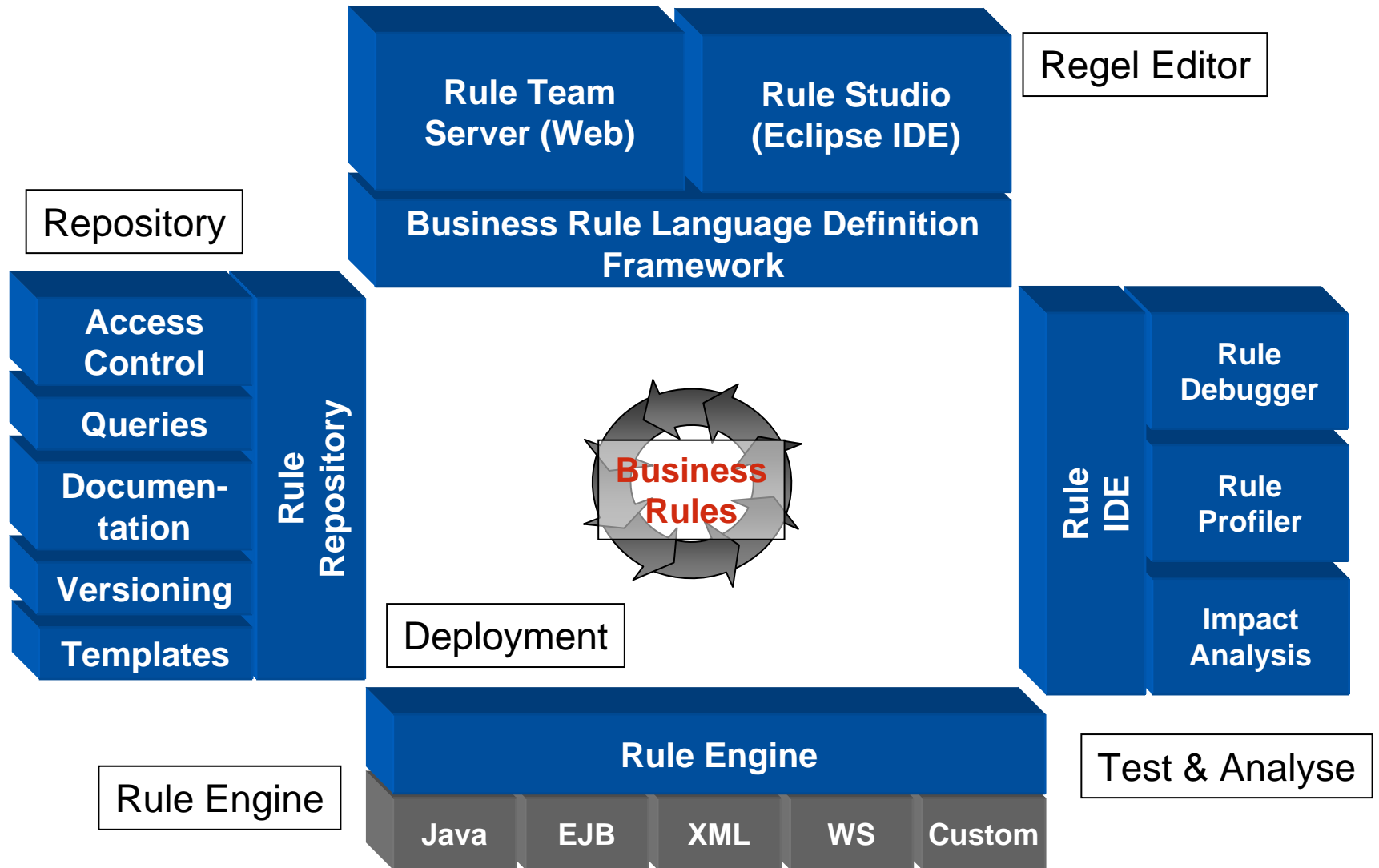


# Komponenten & Aspekte

# Business Rule Lifecycle



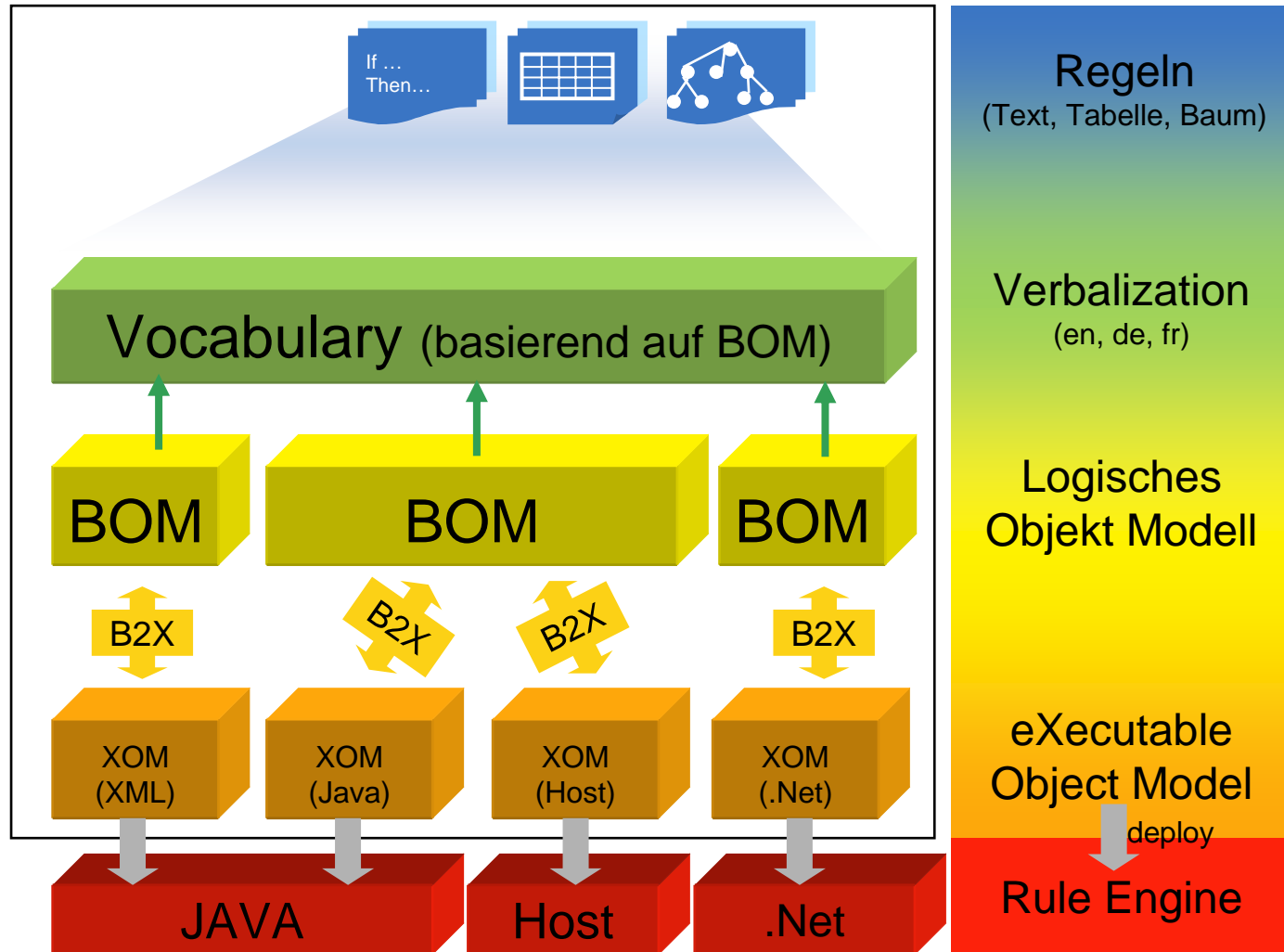
# Komponenten eines BRMS



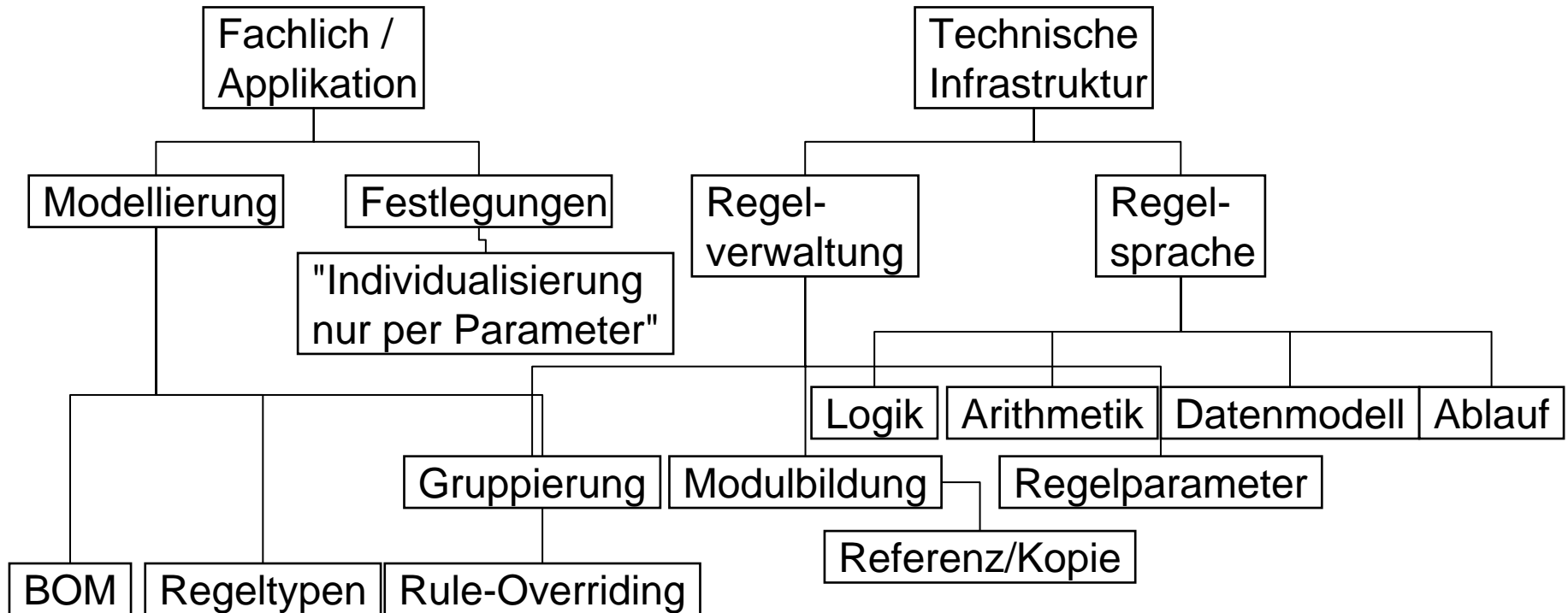


# Fachsprachen

# Fachsprachen (Business Language, Domain Specific Language)



# TRACKS: Fachlich / Technisch





# Verwaltung

# Struktur: Atomare Regeln

Microsoft Access - [Buchungsregel : Table]

File Edit View Insert Format Records Tools Window Help ILOG JRules

Type a question for help

	kommentar	selektionsKriterien	bc	bc	kontoNummerSoll	bilan	kontoNummerHaben	bilanz	originalBetragRe	kontoBetragRege	bilanzBetragRegel
+	Zahlungsauftrag, indir. Eigene Spesen				kukto	bin	bbb 73 7737200.PP	ere	@originalBetrag	@kontoBetrag	inBilanzWaehrung(@kontoBetrag, @kontoWaehrung)
+	Zahlungsauftrag indir. Fremde Spesen				kukto	bin	bbb 73 7737200.PP	ere	@originalBetrag	@kontoBetrag	inBilanzWaehrung(@kontoBetrag, @kontoWaehrung)
+	Zahlungsauftrag direkte Eigene Spesen Origin	not(devisenTauschNoetig())			kukto	bin	bbb 73 7737200.PP	ere	@originalBetrag	@kontoBetrag	inBilanzWaehrung(@kontoBetrag, @kontoWaehrung)
+	Zahlungsauftrag direkte Eigene Spesen Devisen	devisenTauschNoetig()			bbb 74 7743000.PP	bin	bbb 73 7737200.PP	ere	@originalBetrag	@originalBetrag	inBilanzWaehrung(@originalBetrag, @originalWaehrung)
<b>Element Spesen *</b>											
<b>Wenn</b>											
Element ist in ( DirekteEigeneSpesen , DirekteFremdeSpesen , DirekteEinzelZahlSpesen )											
oder Element ist in ( IndirekteEigeneSpesen , IndirekteFremdeSpesen , IndirekteEinzelZahlSpesen )											
<b>Dann</b>											
setze Sollkonto auf : kukto											
setze Habenkonto auf : bbb 73 7737200.PP											
setze BilanzgruppeHaben auf : ere											
setze BilanzgruppeSoll auf : bin											
+	KundenKontouebertrag, DirekteEinzelZahlSp	devisenTauschNoetig()			kukto	bin	bbb 74 7743000.PP	ere	@originalBetrag	@kontoBetrag	inBilanzWaehrung(@kontoBetrag, @kontoWaehrung)
+	SIC-Zahlungsausgang, Zahlungsdaten, Origin	not(devisenTauschNoetig())			kukto	bin	bbb 06 2060600.PP	bin	@originalBetrag	@kontoBetrag	inBilanzWaehrung(@kontoBetrag, @kontoWaehrung)
+	SIC-Zahlungsausgang, Zahlungsdaten, Devisen	devisenTauschNoetig()			bbb 74 7743000.PP	ere	bbb 06 2060600.PP	bin	@originalBetrag	@originalBetrag	inBilanzWaehrung(@originalBetrag, @originalWaehrung)
+	SIC-Zahlungsausgang, Zahlungsdaten, Devisen	devisenTauschNoetig()			kukto	bin	bbb 74 7743000.PP	ere	@originalBetrag	@kontoBetrag	inBilanzWaehrung(@kontoBetrag, @kontoWaehrung)
+	SIC-Zahlungsausgang, IndirekteEinzelZahlSp				kukto	bin	bbb 73 7737200.PP	ere	@originalBetrag	@kontoBetrag	inBilanzWaehrung(@kontoBetrag, @kontoWaehrung)
+	SIC-Zahlungsausgang, DirekteEinzelZahlSp	not(devisenTauschNoetig())			kukto	bin	bbb 73 7737200.PP	ere	@originalBetrag	@kontoBetrag	inBilanzWaehrung(@kontoBetrag, @kontoWaehrung)
+	SIC-Zahlungsausgang, DirekteEinzelZahlSp	devisenTauschNoetig()			bbb 74 7743000.PP	ere	bbb 73 7737200.PP	ere	@originalBetrag	@originalBetrag	inBilanzWaehrung(@originalBetrag, @originalWaehrung)
+	SIC-Zahlungsausgang, DirekteEinzelZahlSp	devisenTauschNoetig()			kukto	bin	bbb 74 7743000.PP	ere	@originalBetrag	@kontoBetrag	inBilanzWaehrung(@kontoBetrag, @kontoWaehrung)
+	Deckungszahlung, Zahlungsdaten, Original	not(devisenTauschNoetig())			kukto	bin	bbb 07 10707aa.PP	bin	@originalBetrag	@kontoBetrag	inBilanzWaehrung(@kontoBetrag, @kontoWaehrung)
+	Deckungszahlung, Zahlungsdaten, Devisen	devisenTauschNoetig()			bbb 74 7743000.PP	ere	bbb 07 10707aa.PP	bin	@originalBetrag	@originalBetrag	inBilanzWaehrung(@originalBetrag, @originalWaehrung)
+	Deckungszahlung, Zahlungsdaten, Devisen	devisenTauschNoetig()			kukto	bin	bbb 74 7743000.PP	ere	@originalBetrag	@kontoBetrag	inBilanzWaehrung(@kontoBetrag, @kontoWaehrung)
+	Deckungszahlung, IndirekteEinzelZahlSp				kukto	bin	bbb 73 7737200.PP	ere	@originalBetrag	@kontoBetrag	inBilanzWaehrung(@kontoBetrag, @kontoWaehrung)
+	Deckungszahlung, DirekteEinzelZahlSp	not(devisenTauschNoetig())			kukto	bin	bbb 73 7737200.PP	ere	@originalBetrag	@kontoBetrag	inBilanzWaehrung(@kontoBetrag, @kontoWaehrung)
+	Deckungszahlung, DirekteEinzelZahlSp	devisenTauschNoetig()			bbb 74 7743000.PP	ere	bbb 73 7737200.PP	ere	@originalBetrag	@originalBetrag	inBilanzWaehrung(@originalBetrag, @originalWaehrung)
+	Deckungszahlung, DirekteEinzelZahlSp	devisenTauschNoetig()			kukto	bin	bbb 74 7743000.PP	ere	@originalBetrag	@kontoBetrag	inBilanzWaehrung(@kontoBetrag, @kontoWaehrung)
+	ZV Sonderfall, Zahlungsdaten, Original	not(devisenTauschNoetig())			kukto	bin	begkto	xxx	@originalBetrag	@kontoBetrag	inBilanzWaehrung(@kontoBetrag, @kontoWaehrung)
+	ZV Sonderfall Zahlungsdaten Devisen	devisenTauschNoetig()			bbb 74 7743000.PP	ere	heaktu	xxx	@originalBetrag	@originalBetrag	inBilanzWaehrung(@originalBetrag, @originalWaehrung)



# Anti-Pattern: Free Flying Rules

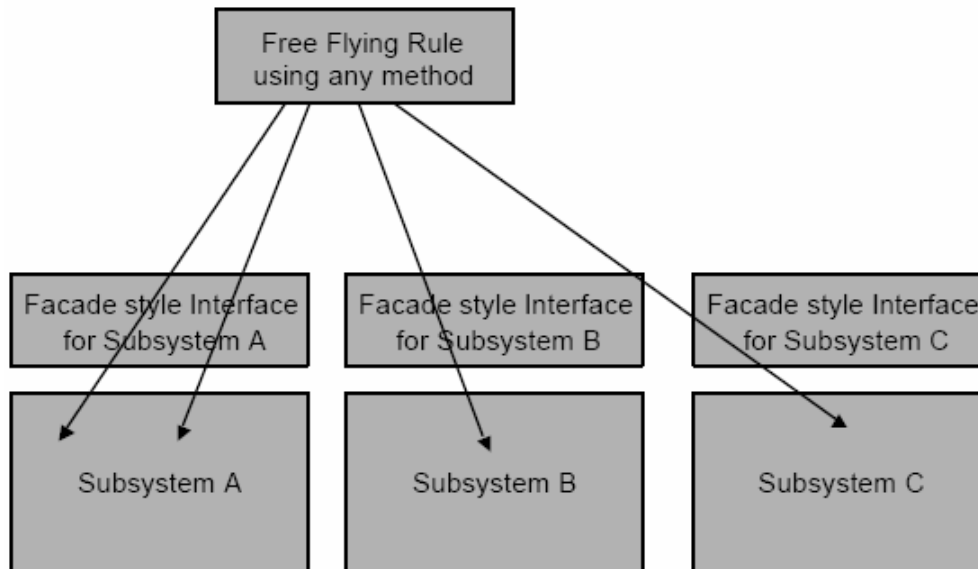


Figure 2: Free Flying Rule – The rule may use any method of the entire object model as a parameter regardless of interfaces, facades, and modularization considerations

Wolfgang Keller: The Pitfalls of Meta-Systems and Business Rules

# Anti-Pattern: Free Flying Rules

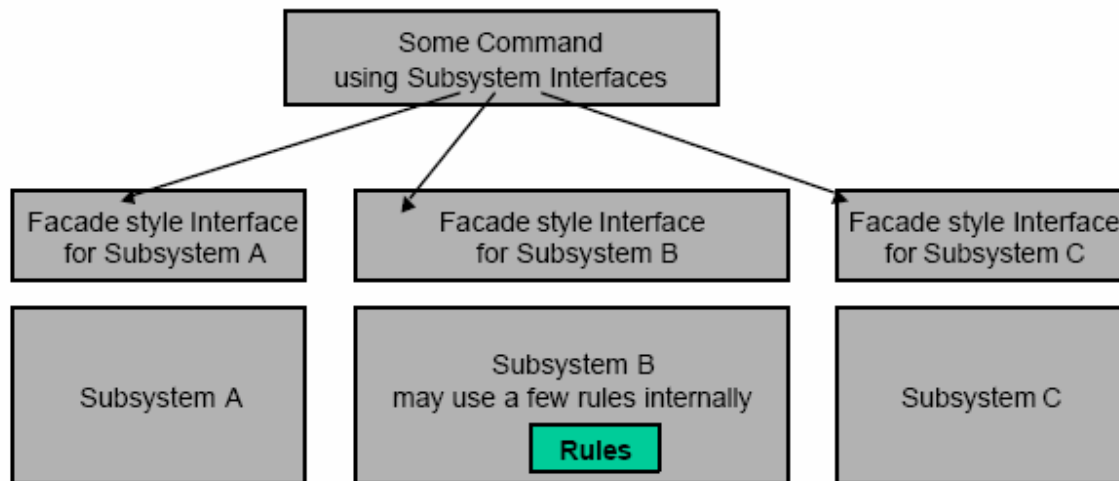


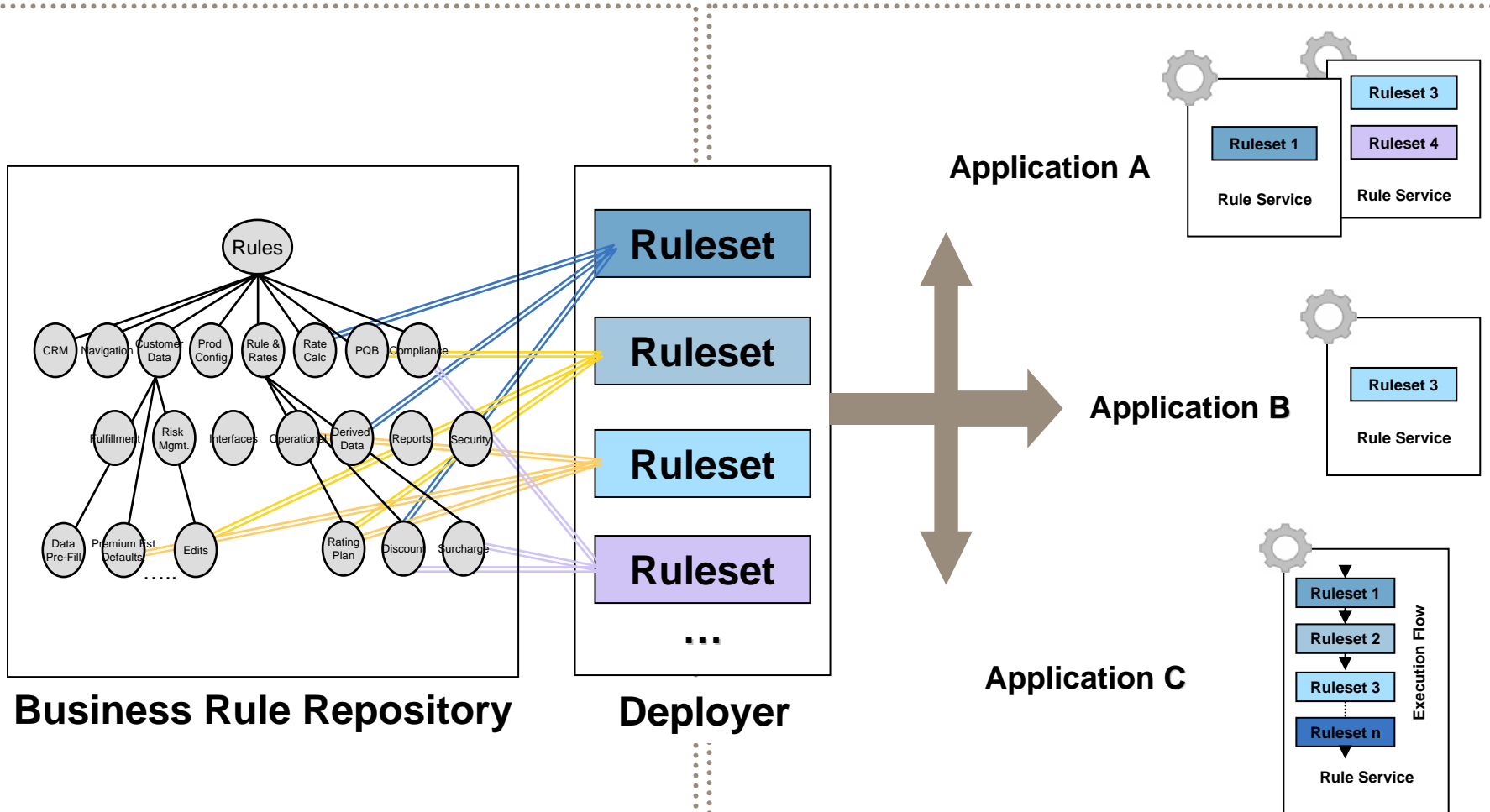
Figure 3: A rule based solution that may be better conforming to good design practices. Rules are confined to a single subsystem. If there is functionality, that needs to span more than one subsystem, it is wrapped into a command [GOF95]. The command uses well defined subsystem interfaces.

Wolfgang Keller: The Pitfalls of Meta-Systems and Business Rules



# Integration

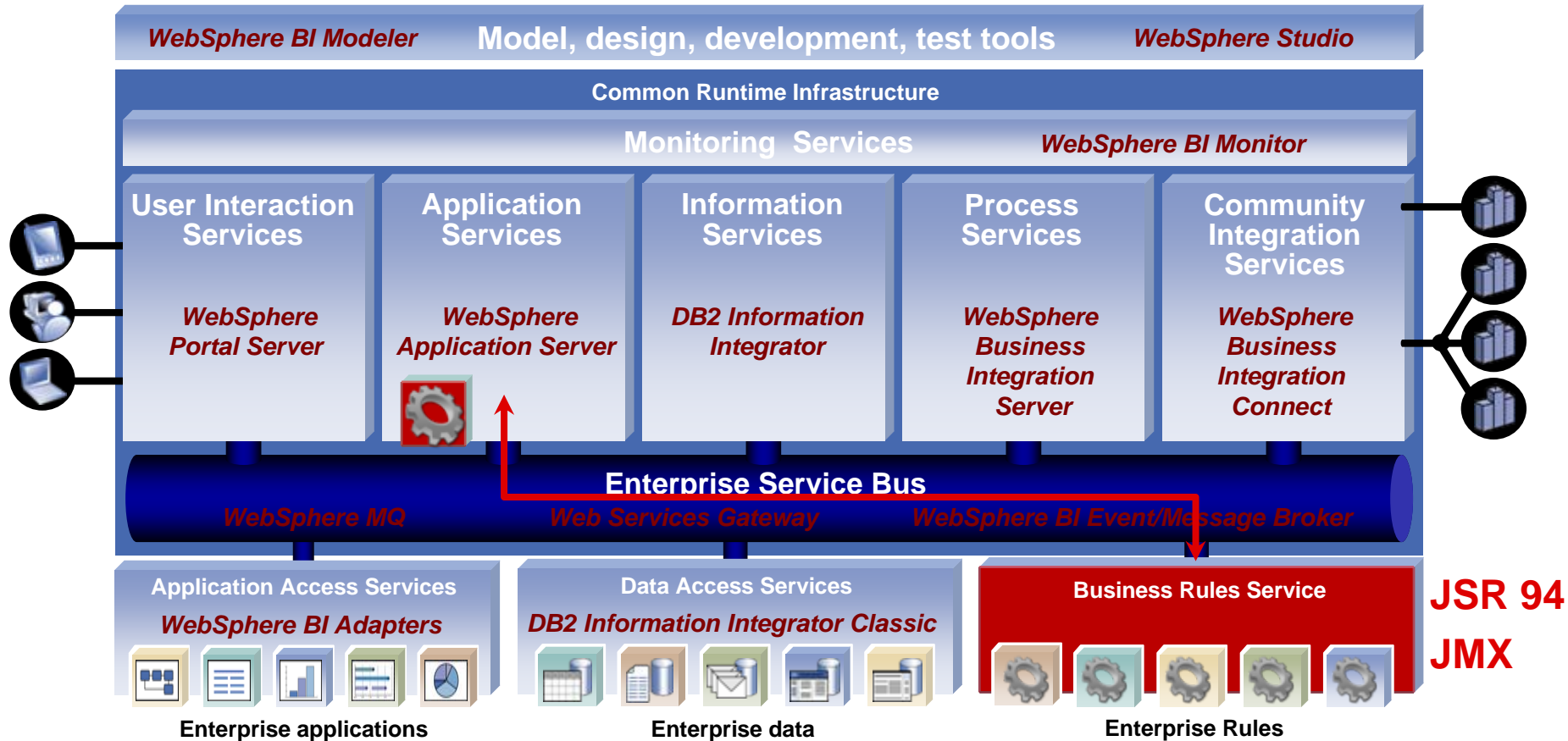
# Administration & Runtime View



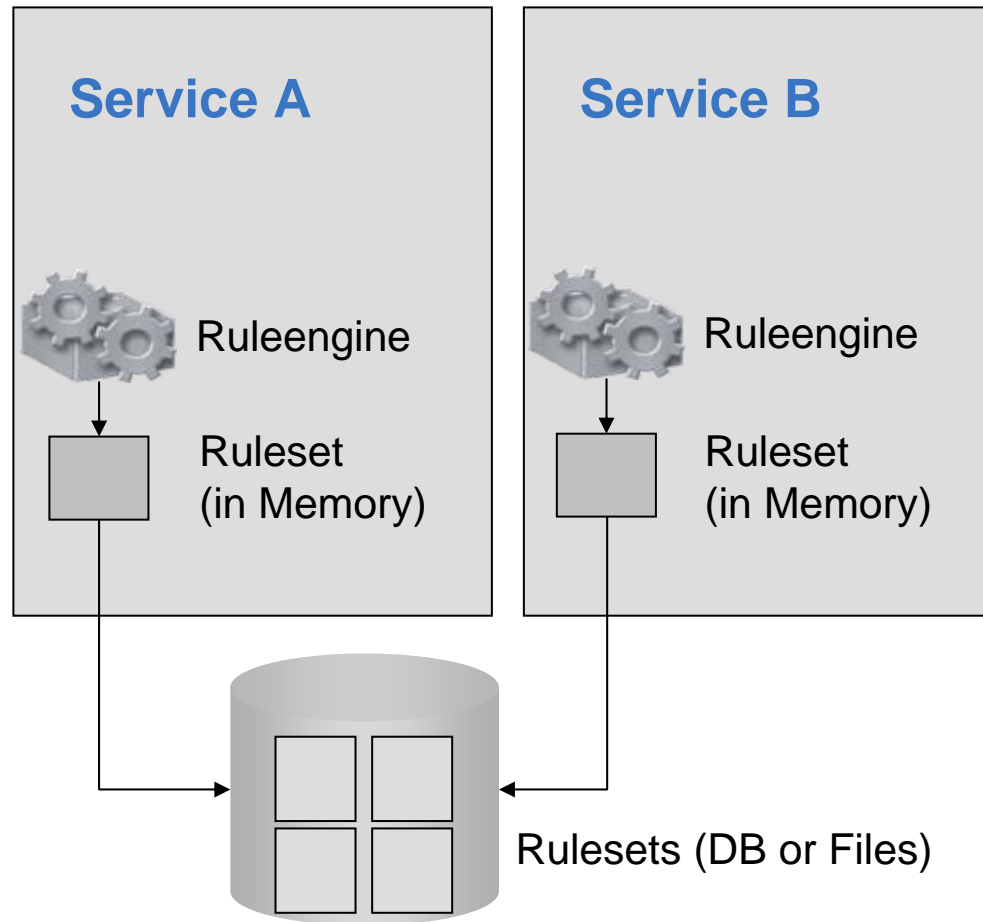
Administration

Deployment

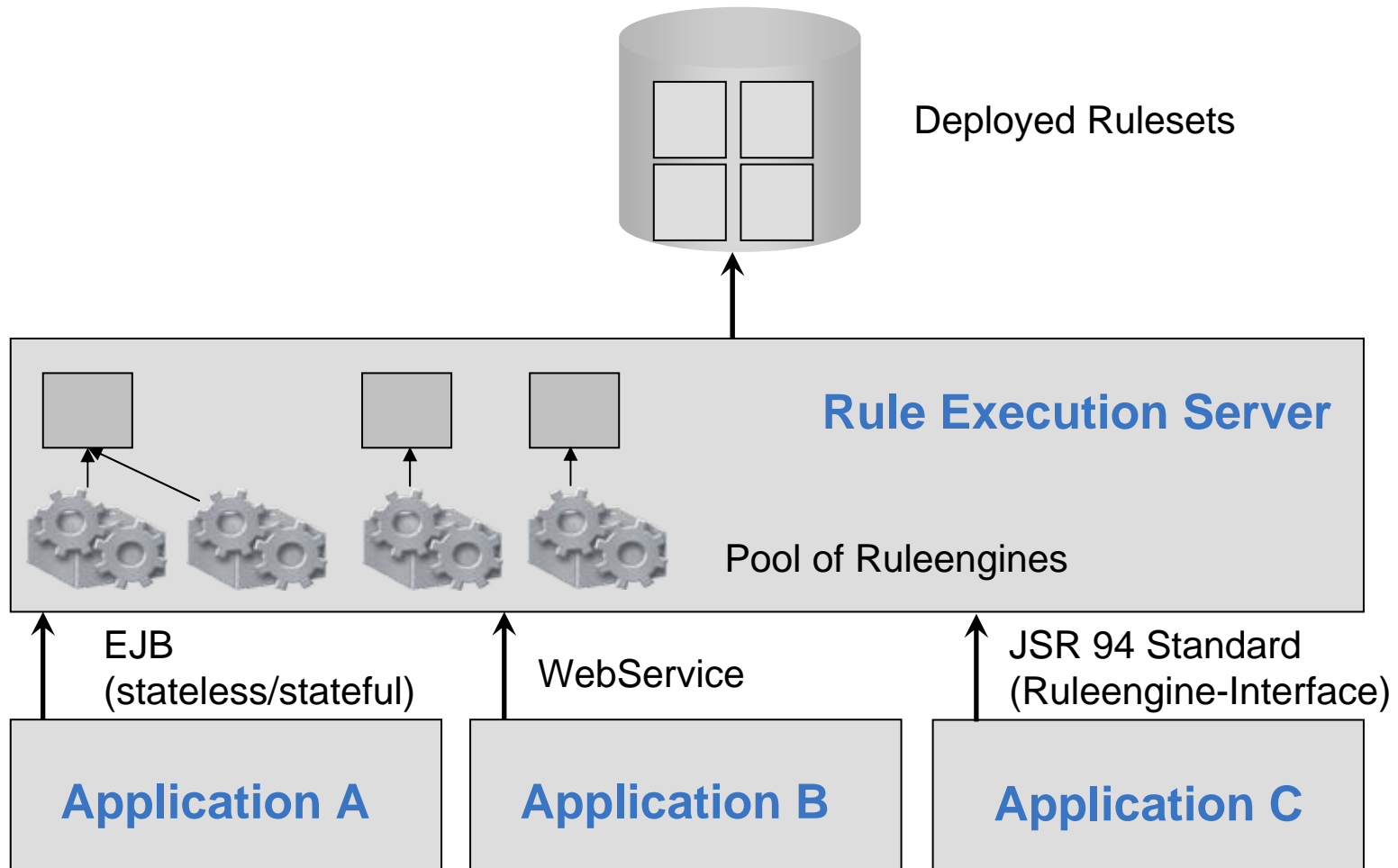
# Business Rule Services



# Embedded Ruleengines (J2SE / J2EE)



# Rule Execution Server (J2EE / Webservice / JSR 94)

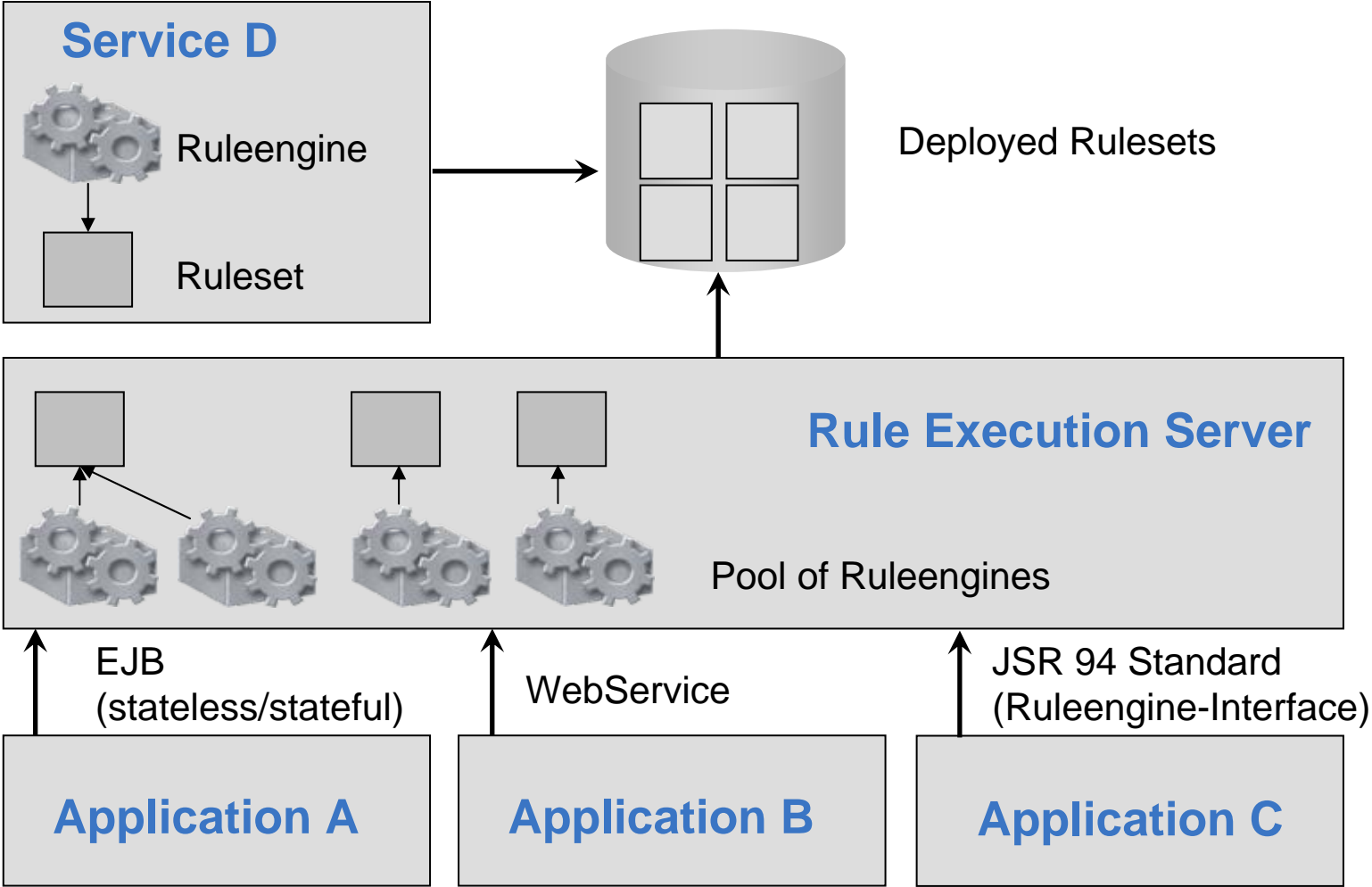


# Pro & Con

	Embedded	Rule Execution Server
Pro	<ul style="list-style-type: none"> <li>• Simplicity</li> <li>• Performance</li> <li>• Functionality (low level APIs)</li> </ul>	<ul style="list-style-type: none"> <li>• Managable (Deployment-Infrastructure, JMX)</li> <li>• Scalable (Engine-Pooling)</li> <li>• Interoperability (EJB / WebService / JSR 94)</li> </ul>
Con	<ul style="list-style-type: none"> <li>• Java-APIs only</li> <li>• Scalability not out of the box</li> <li>• No deployment support</li> </ul>	<ul style="list-style-type: none"> <li>• Communication overhead</li> <li>• Complexity</li> </ul>

# Central Ruleset DB

## Coexistence Embedded & RES





# Test & Analyse

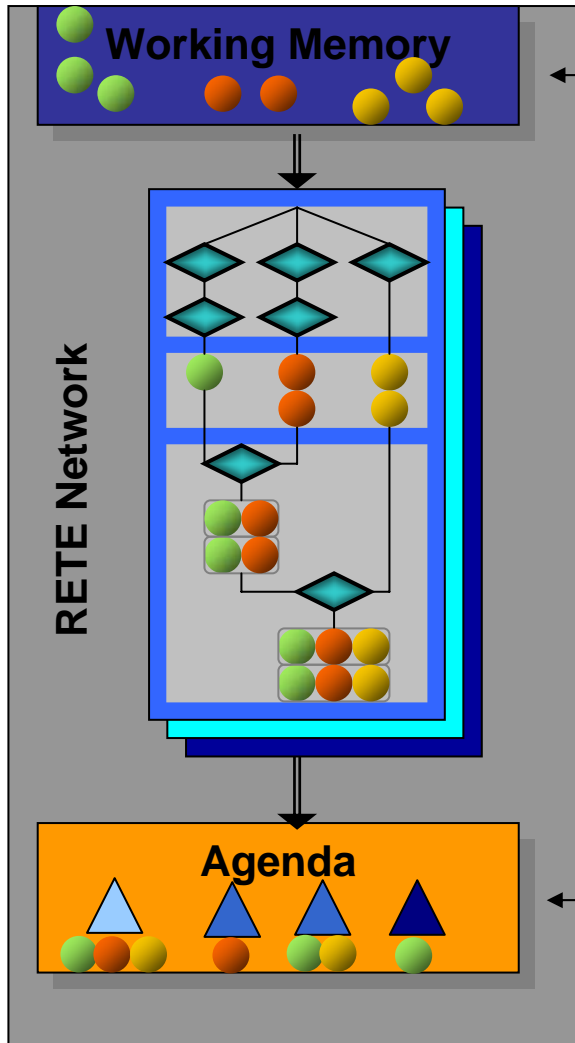
## Test & Analyse

- **Statische Regelanalyse nur begrenzt einsetzbar; Kombination mit Laufzeittests erforderlich**
- **Testumgebung für Fachanwender (vorzugsweise Excel-basiert)**
- **Coverage-Analyse hilfreich**
- **Entwicklungsrepository für IT mit allen Funktionalitäten**



# Algorithmen, Marktübersicht, Standards

# RETE-Algorithmus (1982 by C. Forgy)

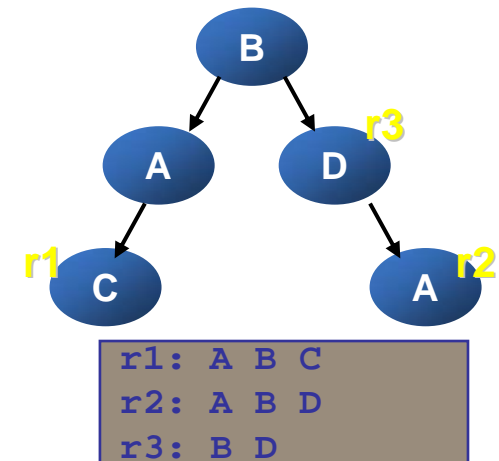
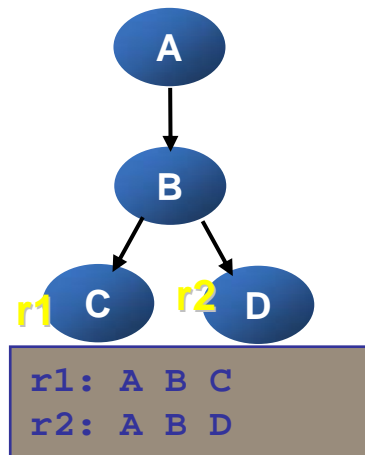
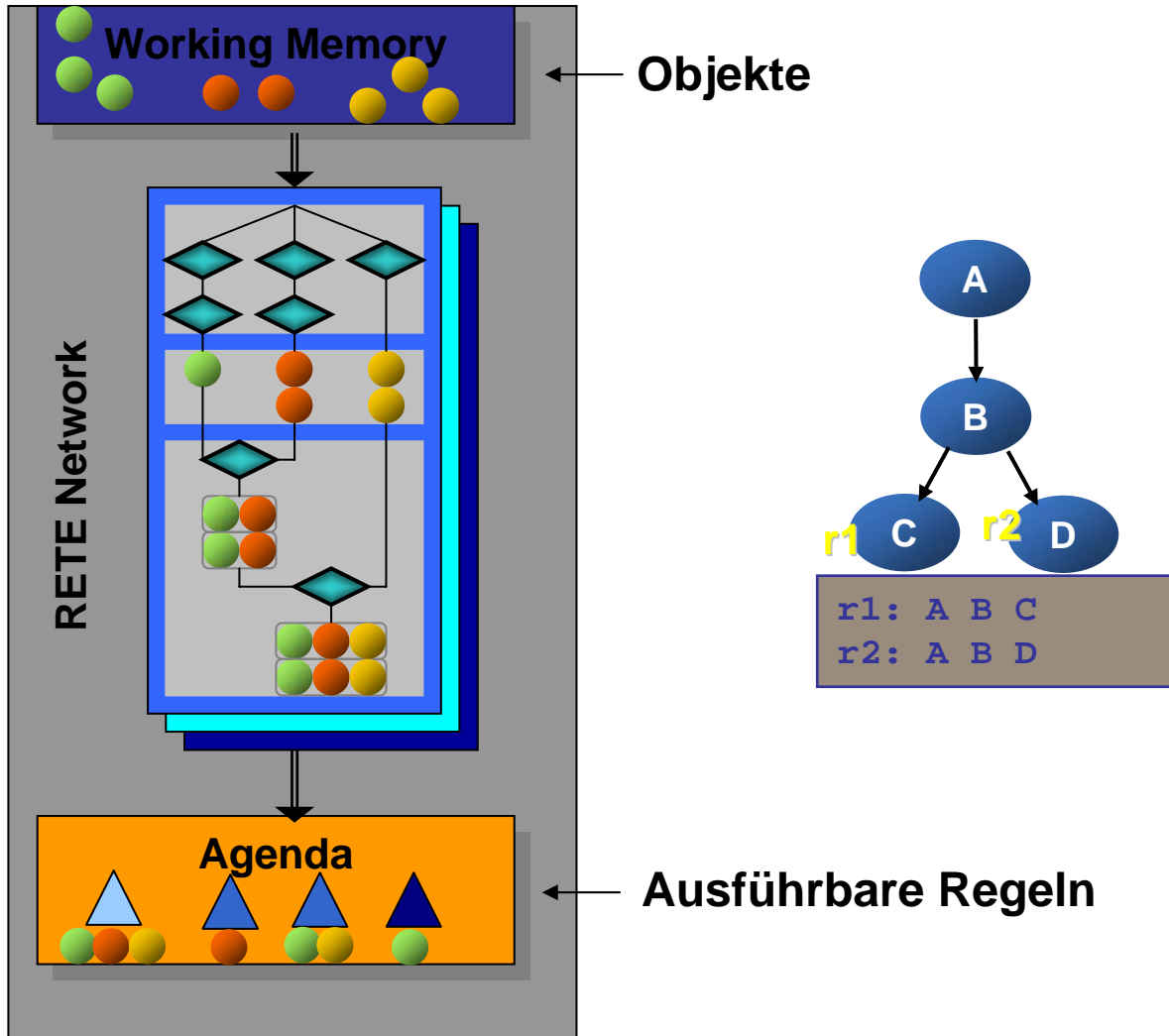


← Objekte

1. Führe "erste" Regel der Agenda aus;  
⇒ Update von Objekten
2. Evaluiere alle Regeln neu  
⇒ Update der Agenda

← Ausführbare Regeln

# RETE-Algorithmus (1982 by C. Forgy)



# Marktübersicht

Magic Quadrant for Business Rule Engines, 2005

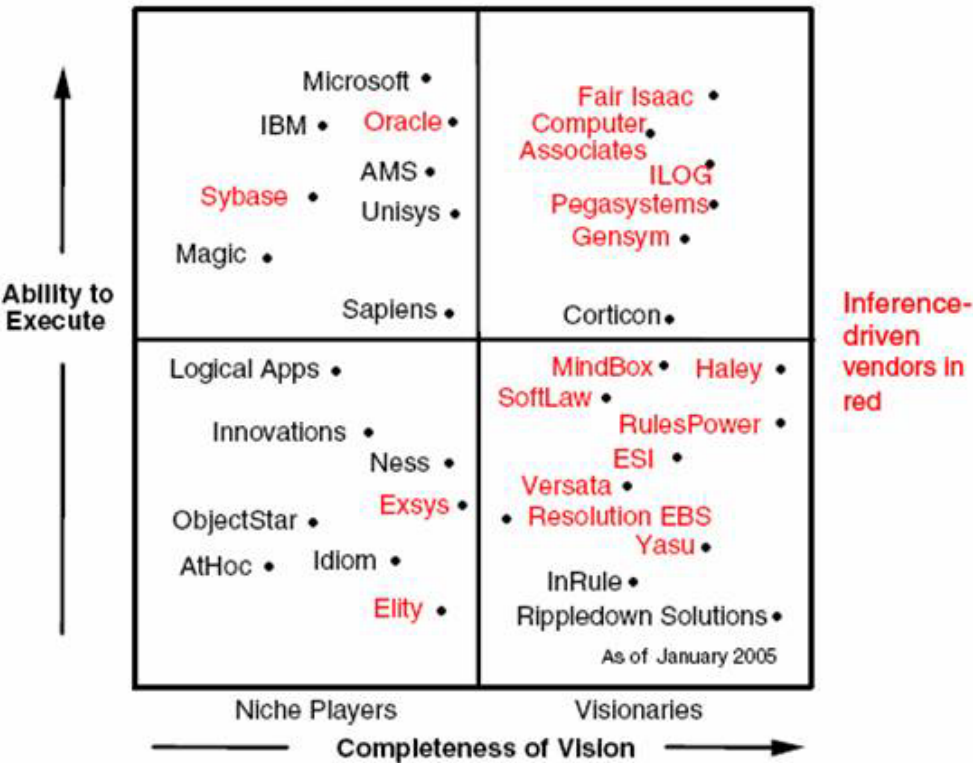
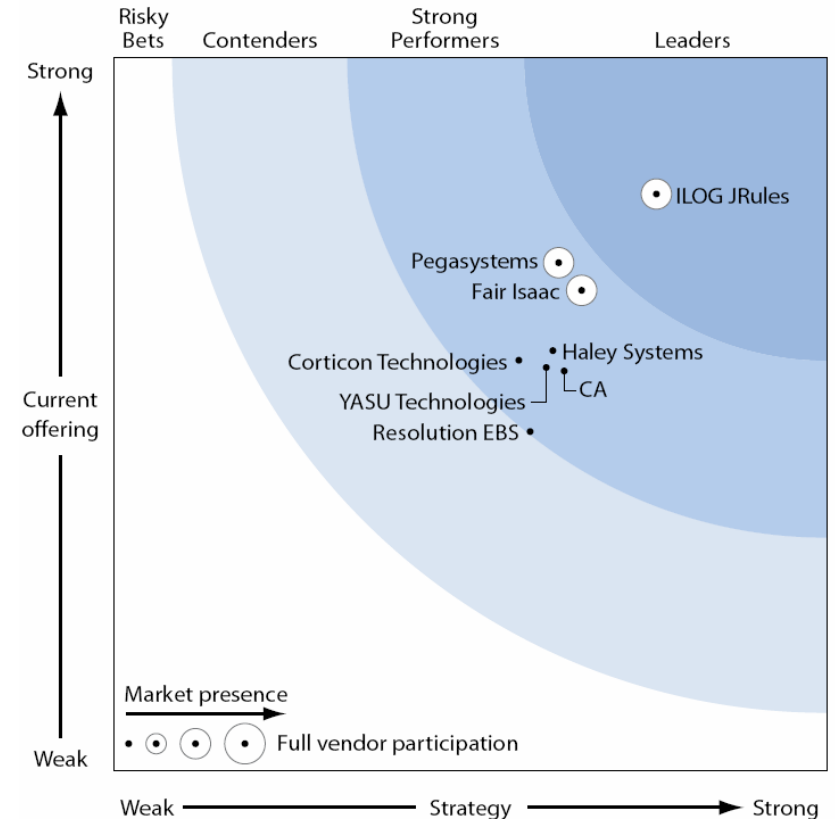


Figure 5 Forrester Wave™: Business Rules Platforms For Java Developers, Q1 '065



# Standards

- ❑ **Java Rule Engine API – JSR 94**  
<http://java.sun.com/developer/technicalArticles/J2SE/JavaRule.html>
- ❑ **Extend J2SE and J2EE with a standard rule engine API**



## Regelaustauschformate:

- ❑ **SRML – Simple Rule Markup Language**  
<http://www.oasis-open.org/cover/srml.html>
- ❑ **RuleML**  
<http://www.oasis-open.org/cover/ruleML.html>
- ❑ **BRML**  
<http://www.oasis-open.org/cover/brml.html>
- ❑ **W3C Rule Interchange Format**  
<http://xml.coverpages.org/W3C-RIF-Announce.html>

# Questions?

