

# Reasoner for the Semantic Web



KAON & KAON2

Seminar A.I. Tools

Erik Endres

18.1.2007

# Übersicht

- Reasoner
- KAON1
- KAON2 & Protégé

# Reasoner

Ontologien machen Daten für Maschinen  
verarbeitbar.

Reasoner setzen auf dieser  
Wissensrepräsentation auf und leitet neues  
Wissen ab.

# Reasoner

Reasoner = Inferenzer

Inferenz / Schlussfolgern:

Prämissen: „Alle Griechen sind sterblich“,  
„Sokrates ist ein Grieche“

Konklusion: „Sokrates ist sterblich“

# Reasoner

```
<owl:Class rdf:ID="Grieche">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue>
        <Sterblichkeit rdf:ID="Sterblich"/>
      </owl:hasValue>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hat_Lebensspanne"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<Grieche rdf:ID="Sokrates"/>
```

# Reasoner

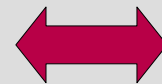
## Aufgaben:

- Klassifikation
- Konsistenzcheck
- Bearbeiten von Anfragen

# Reasoner

## Anforderungen:

- Korrektheit
- Vollständigkeit
- Entscheidbarkeit
- Effizienz



Ausdrucksmächtigkeit  
der  
Repräsentations-  
sprache

# Reasoner

## Logiken:

### Prädikatenlogik

- zu mächtig
- Prolog

### Beschreibungslogik

- Kompromiss zwischen  
Ausdrucksmächtigkeit – Entscheidbarkeit
- Teilmenge der Prädikatenlogik
- OWL-DL

# Reasoner

Anforderungen im Semantik Web:

- Skalierbarkeit
- Adaptive Performance
- Robustheit
- Integration verteilten Wissens
- Verständlichkeit

# Reasoner

## Beispiele:

- FaCT++                      Open Source, C++
- Racer
- Pellet                      Open Source, Java
- **KAON2**                    Java

# KAON(1)

*KArlsruhe ONtology and Semantic Web tool suite*  
a.k.a. **KAON ToolSuite**

Entwickelt von der Universität Karlsruhe und  
dem Forschungszentrum Informatik in Karlsruhe.

1. Version: 2002

Open Source

# KAON ToolSuite

Die KAON ToolSuite besteht aus:

- KAON  
enthält Frontend und Core
- KAON Extensions  
enthält DLP, KAON Server und KAONtoEdit
- TextToOnto

# KAON

- Frontend
  - KAON Workbench
    - besteht aus OI-Modeler, einem Ontologie-Editor
  - KAON Portal
    - ein Tool für Ontologie basierte Webportale
- Core
  - enthält die KAON API und unterschiedliche Standard-Implementierungen, wie den **Engineering Server**

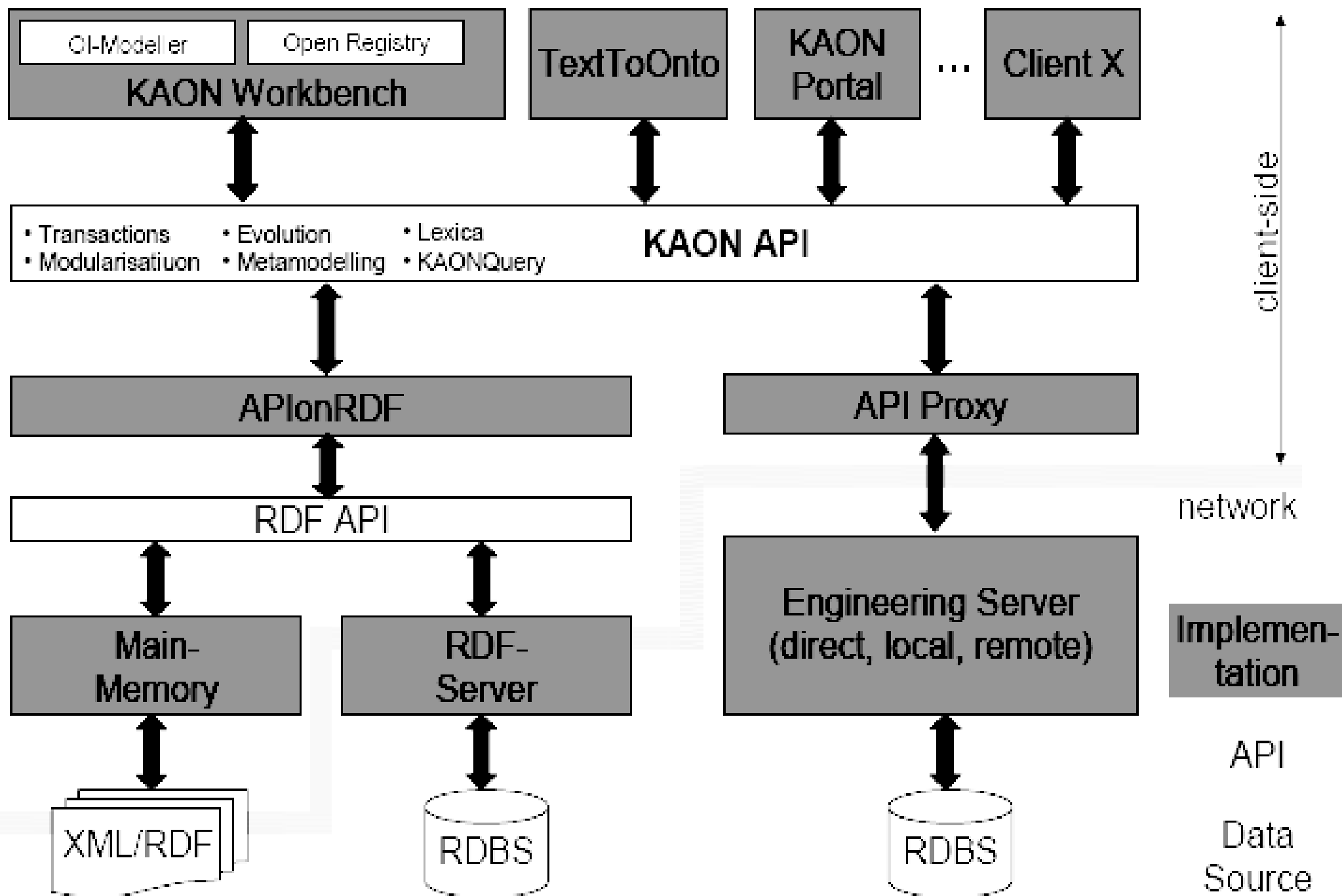


Figure 1.2: KAON Architecture Overview

# KAON2

## Verbesserungen:

- API für OWL-DL, SWRL und F-Logik mit Einschränkungen für das Reasoning

OWL-DL ist auf die SHIQ(D) Teilmenge beschränkt.(Keine Enumeration)

```
<owl:Class>  
  <owl:oneOf rdf:parseType="Collection">  
    <owl:Thing rdf:about="#Sterblich"/>  
    <owl:Thing rdf:about="#Unsterblich"/>  
  </owl:oneOf>  
</owl:Class>
```

# KAON2

- **DIG Interface**

erlaubt Zugriffe über Tools wie Protégé

Standardisiertes XML Interface für  
Beschreibungslogiksysteme

Entwickelt durch die DL Implementation Group (DIG)

- **Inferenz Engine**

für Anfragen in SPARQL Syntax oder  
F-Logik

SPARQL ist eine Abfragensprache für das Semantische Web  
Entwickelt durch das W3C als Nachfolger für RDF Query  
Language, RDQL.

# KAON2

Alle anderen zu Beginn genannten Reasoner verwenden das Baumkalkül (tableaux methods).

- Baumkalküle versuchen zu zeigen, dass die Negation des Arguments ungültig ist. D.h. die Widerlegung der Ungültigkeit.
- Widerlegungskalküle

Anstelle  $\Gamma \models \varphi$  zu zeigen,

wird  $\Gamma \cup \{\neg\varphi\}$  widerlegt.

# KAON2

KAON2 übersetzt die Wissensbasis in ein disjunctive datalog Programm.

- Datalog:  
Datenbank-Programmiersprache, die Prolog ähnelt.

Höherer Aufwand für T-Box  
Geringerer Aufwand für A-Box

=> Effizienter bei großen Instanzenmengen

# KAON2

- Kein OPEN SOURCE!
- Karlsruher Firma Ontoprise GmbH
- 1. Version: 2005
- Für Forschung und Lehrer kostenlos
- Keine Benutzeroberfläche sondern DIG Interface

# KAON2

- Kein Benutzerhandbuch!
- Beispielimplementationen für die API

Example 2:

Ontologie erzeugen, einige Axiome hinzufügen und das Ergebnis in eine Datei speichern.

Example 4:

Regeln zu einer Ontologie hinzufügen und einfache Anfragen stellen.

# KAON2 & Protégé

- KAON2 Server starten:

```
java -cp kaon2.jar
```

```
org.semanticweb.kaon2.server.ServerMain
```

```
-registry -rmi -ontologies server_root -dig
```

```
-digport 8088
```

- Protégé -> OWL -> Preferences -> reasoner URL:  
<http://localhost:8088>

# KAON Tools

## KAON2 OWL Tools

- Entwickelt am AIFB, Universität Karlsruhe
- Toolsammlung die auf KAON2 basieren.

## KAON Web Services Tool Suite

- Entwickelt am AIFB, Universität Karlsruhe
- Ontologien in Web Services

# Quellen

- <http://kaon.semanticweb.org>
- <http://kaon2.semanticweb.org>
- <http://de.wikipedia.org/wiki/Baumkalkül>
- <http://de.wikipedia.org/wiki/Datalog>
- <http://dig.sourceforge.net/>
- <http://semantic.nodix.net>