# Implementierung von Entscheidungs-Unterstützungs-Systemen auf Komponentenbasis

Forstliche Arbeitswissenschaft
und Angewandte Informatik

Martin Döllerer 04/2002

# Übersicht

- Warum Komponenten?
- Was sind Komponenten?
- Wie geht man vor?
- Was kommt dabei heraus?

# Warum Komponenten?

- Was sind Komponenten?
- Wie geht man vor?
- Was kommt dabei heraus?

# Anforderungen an ein EUS

- Fachliche Anforderungen
  - Komplexer Problemraum
  - Teillösungen aus unterschiedlichen Forschungsrichtungen
- Informationstechnische Anforderungen
  - Integration in bestehende Systeme
  - GIS-Anbindung

# Konsequenzen für das EUS

- Funktionalität
- Schnittstellen
  - Intern
  - Extern
- Performanz / Verteilung

Flexibilität & Komplexität

# Architektur komplexer Software-Systeme

- Trennung von
  - Datenhaltung
  - Applikations-Logik
  - Darstellung / Visualisierung
- Kommunikations-Schnittstellen

## 3-Ebenen-Architektur

# 3-Ebenen-Architektur (3-tier-architecture)

- Datenebene
  - data tier
- Applikationsebene
  - application tier
- Präsentationsebene
  - presentation tier

# Datenebene

- Einheitliche gemeinsame Datenbasis
  - Sachdaten
  - Geodaten
- Gemeinsamer Datenzugriff

Relationales DBMS

# Anwendungsebene

- Definierte, in sich geschlossene Teilbereiche
- Gesamtfunktionalität ergibt sich aus der Summe der Teilfunktionen

Komponenten

- Warum Komponenten?
- **Was sind Komponenten?**
- Wie geht man vor?
- Was kommt dabei heraus?

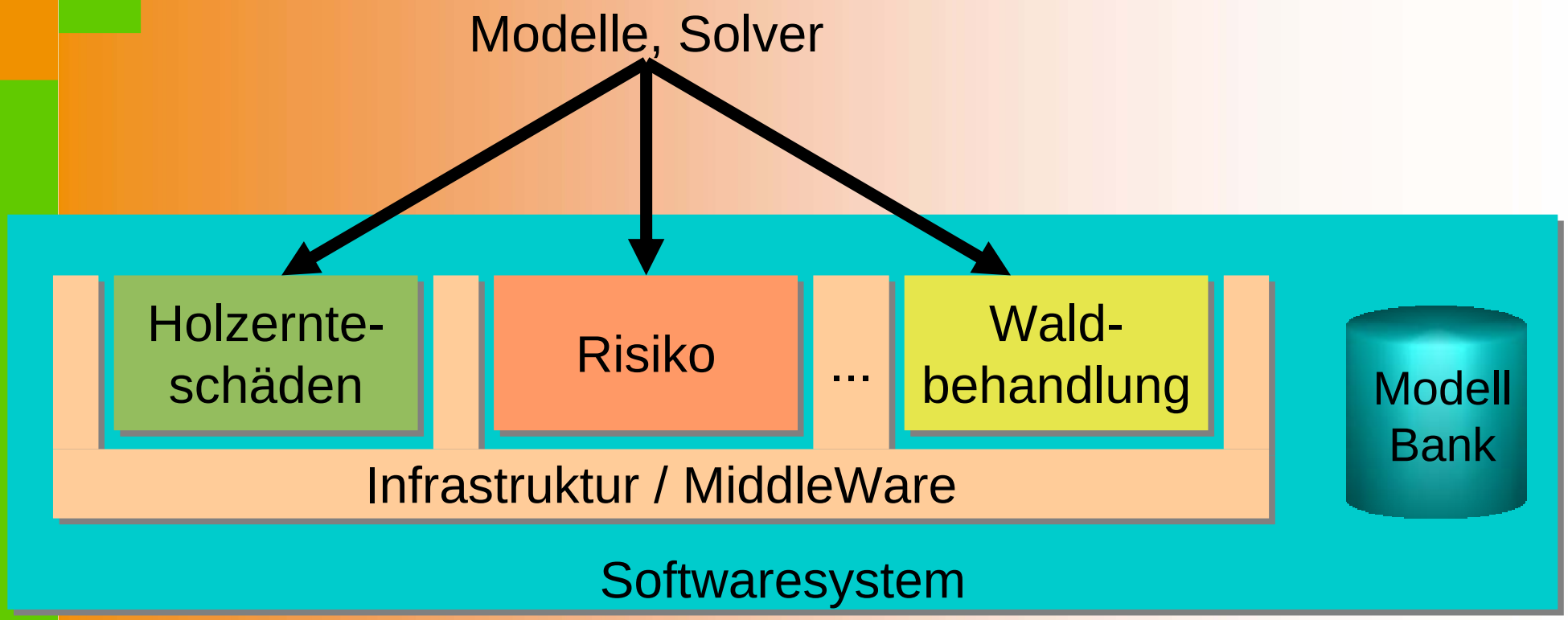
# Eine Komponente ist...

- Eigenständig
- Kommunikativ
- Nicht zu klein, nicht zu groß
- Wiederverwendbar
- Anpassbar

# Komponentenorientierte Software-Entwicklung

- Mächtiges Software-Paradigma
  - Erweiterung / Ergänzung der objektorientierten Sicht
- Komponenten-Entwicklung
- Anwendungs-Entwicklung
  - Zusammensetzen von Komponenten zu Anwendungen

# Software aus Komponenten

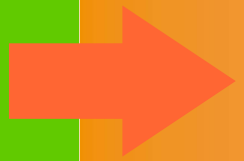


# Infrastruktur (MiddleWare)

- JAVA-Umfeld
  - (Enterprise) Java Beans
  - Proprietäre Lösungen
- Common Object Request Broker Architecture (CORBA)
- (D)COM, ActiveX, .NET

# Vorteile komponenten-orientierter Software

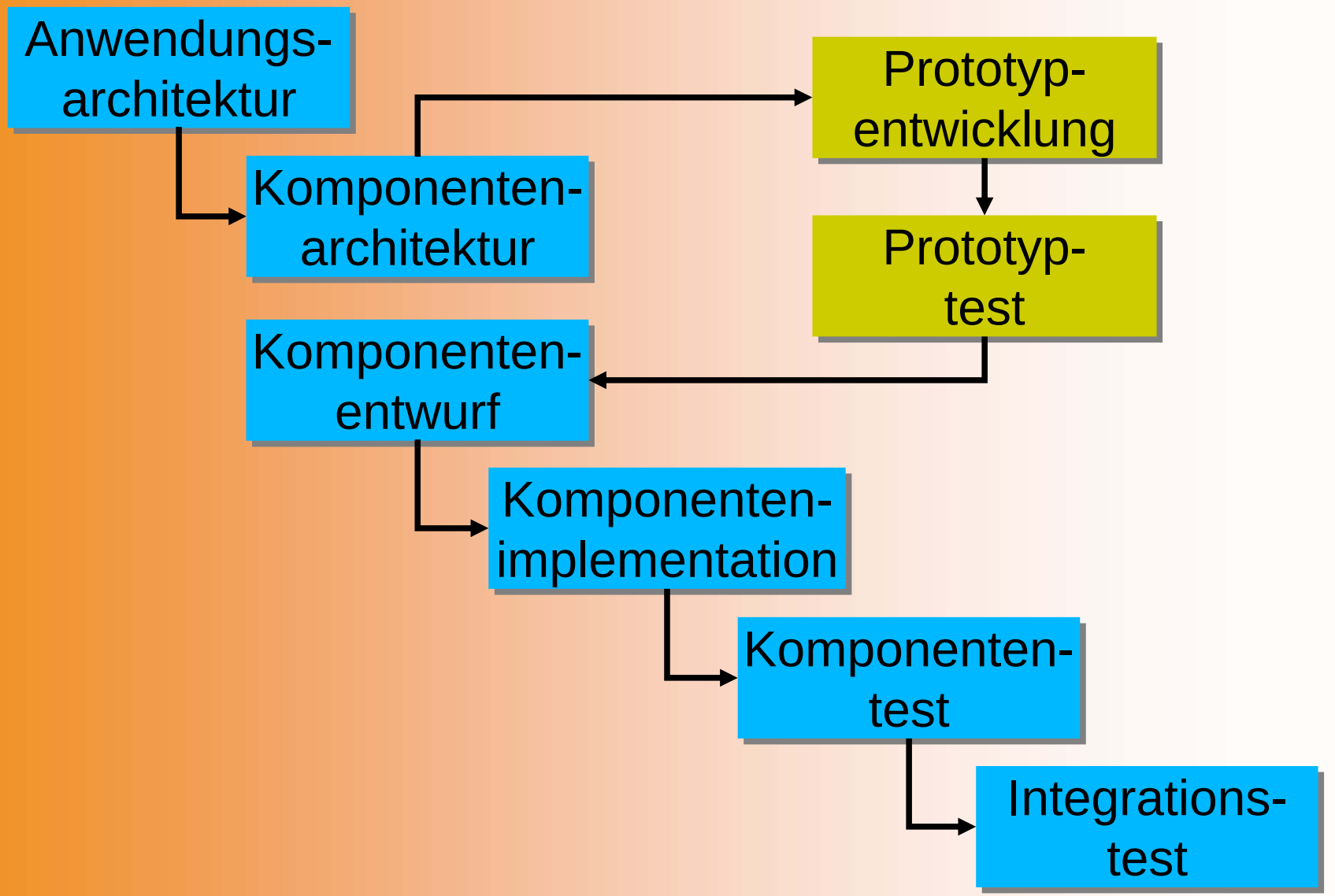
- Der Einsatz austauschbarer, anpassbarer Komponenten verbessert:
  - Erweiterbarkeit
  - Adaptierbarkeit
  - Wartbarkeit



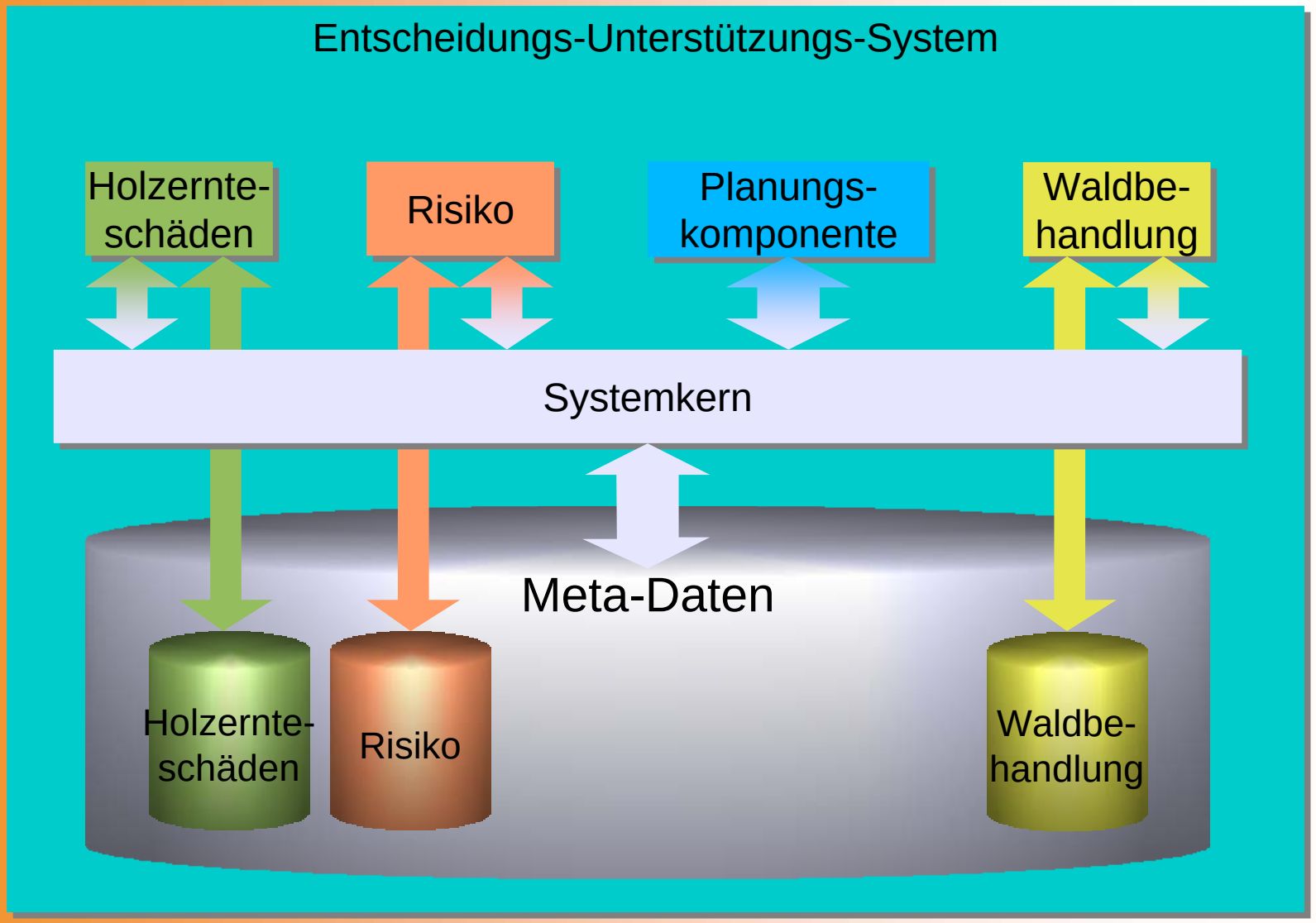
## Flexible Software-Architektur

- Warum Komponenten?
- Was sind Komponenten?
- **Wie geht man vor?**
- Was kommt dabei heraus?

# Vorgehensmodell

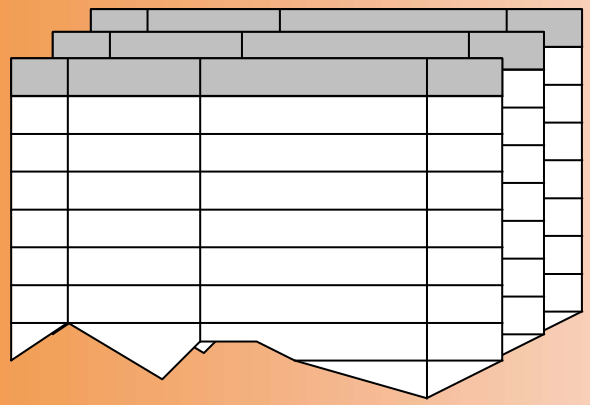


# Anwendungsarchitektur

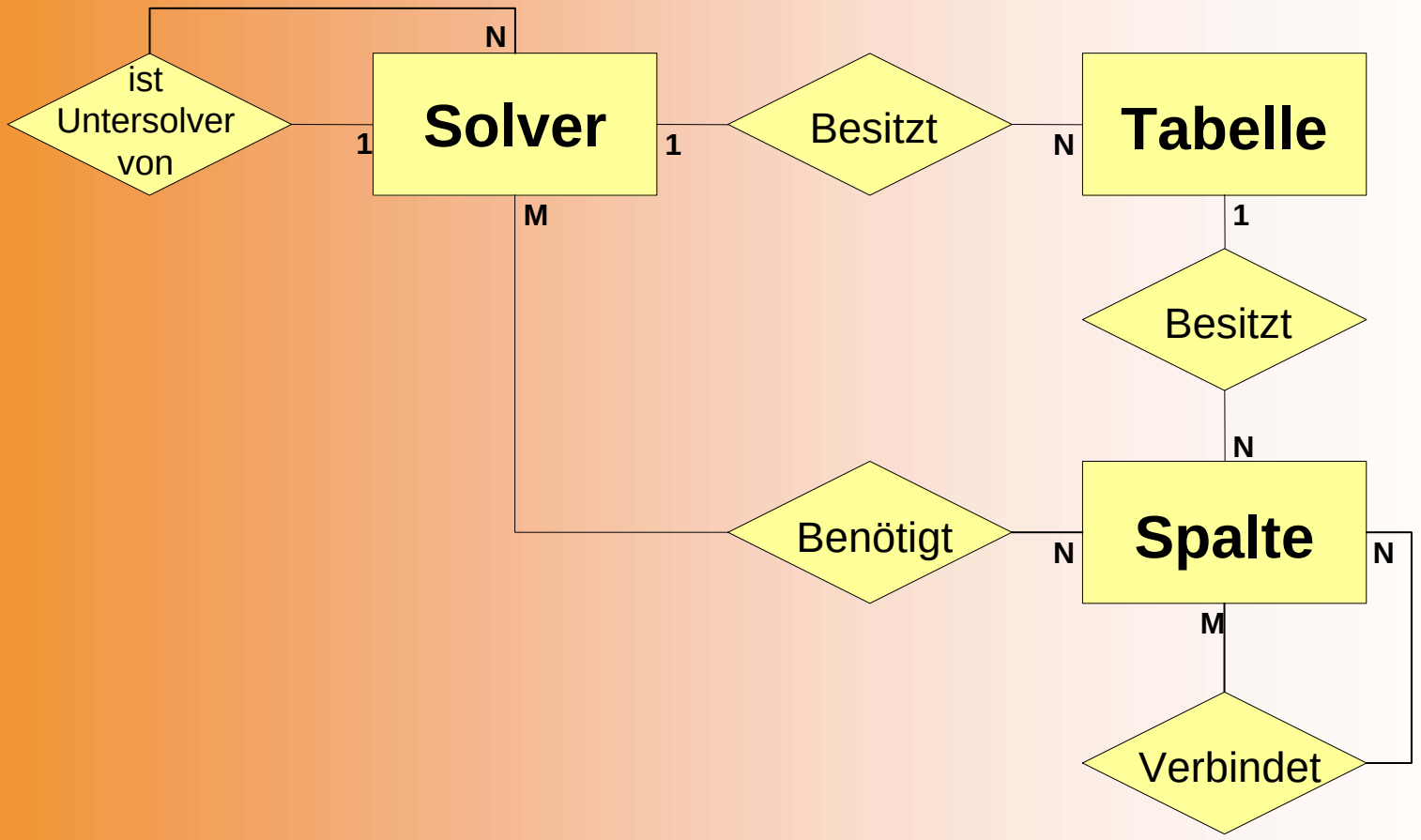


# Meta-Daten

- Datenkatalog
- Abstraktes Datenmodell
  - Logische Sicht auf Tabellen und Spalten



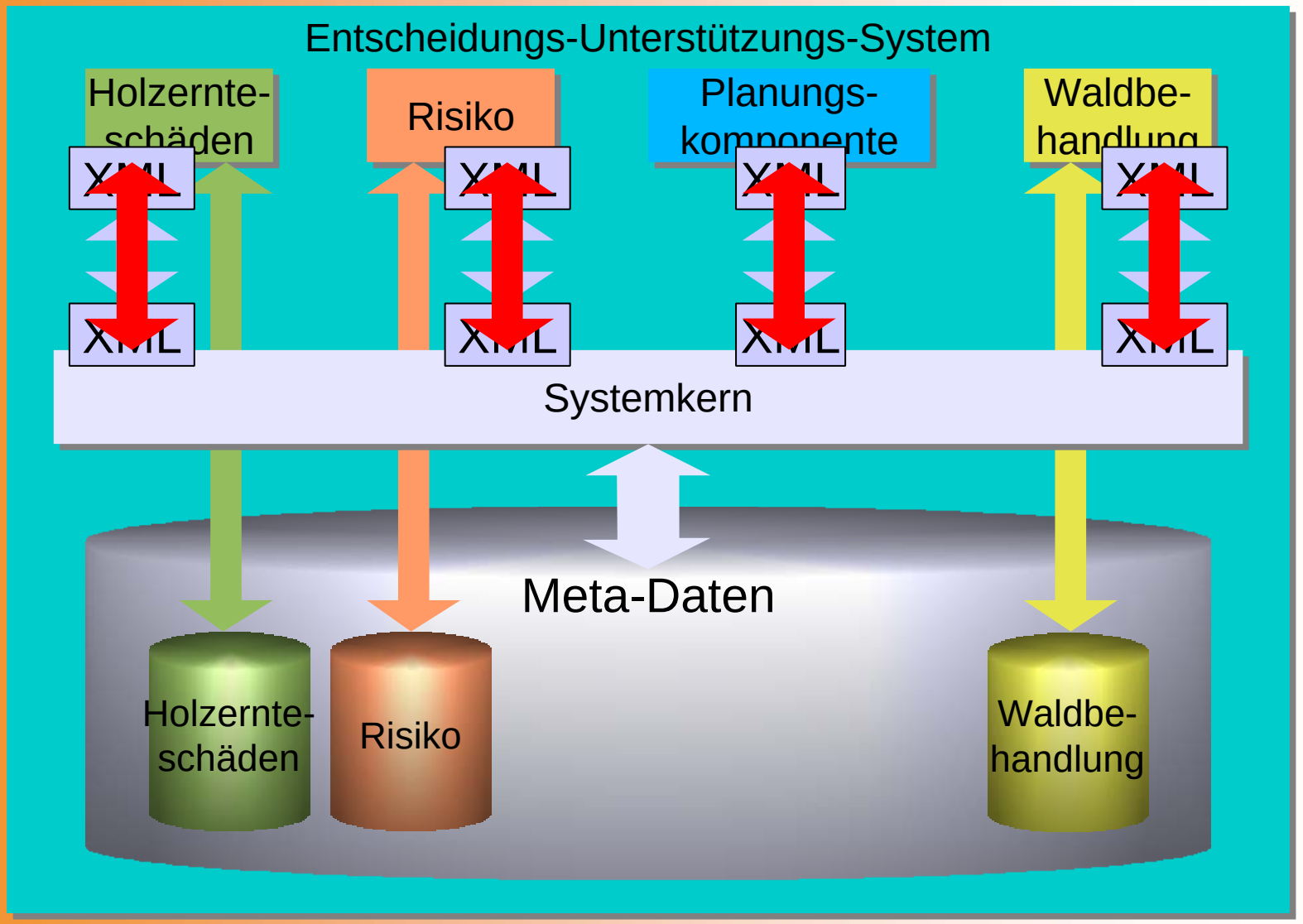
# Datenmodell Metadaten



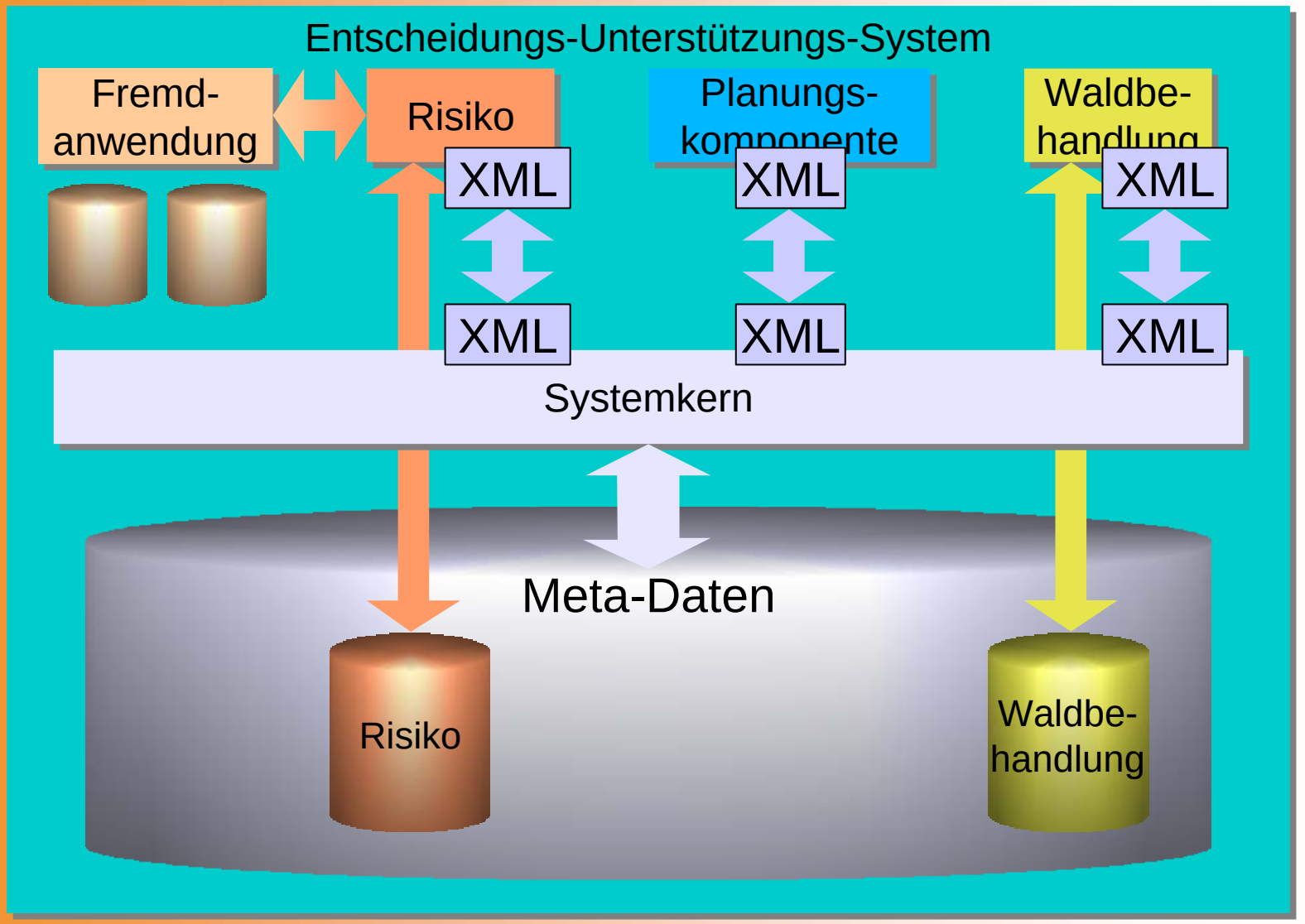
# Metadaten-Dienste

- Solver-Registrierung
- Ergebnis-Präsentation
- Daten-Präsentation
- Solver-Startreihenfolge

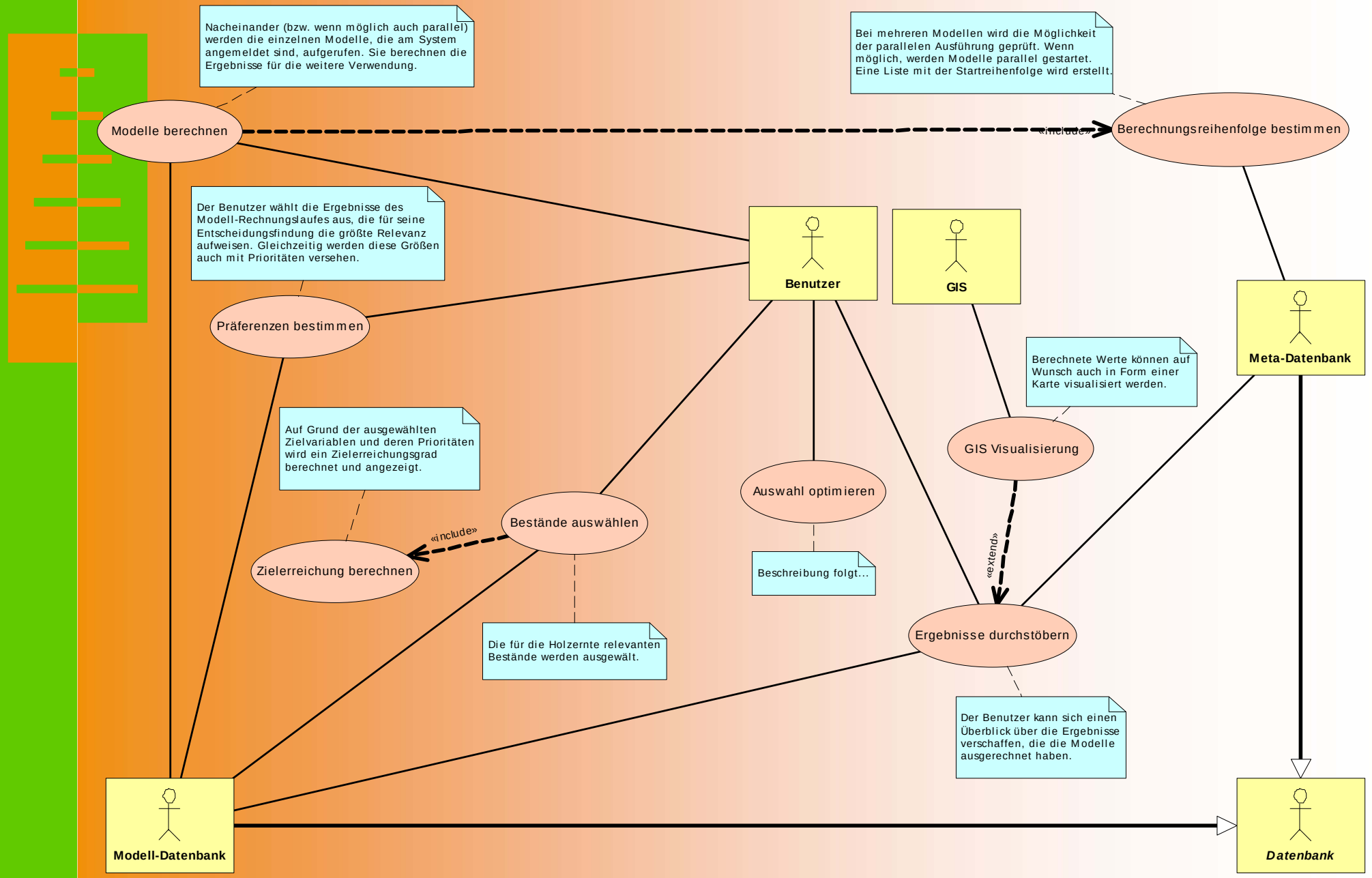
# Client-Server-Schnittstelle

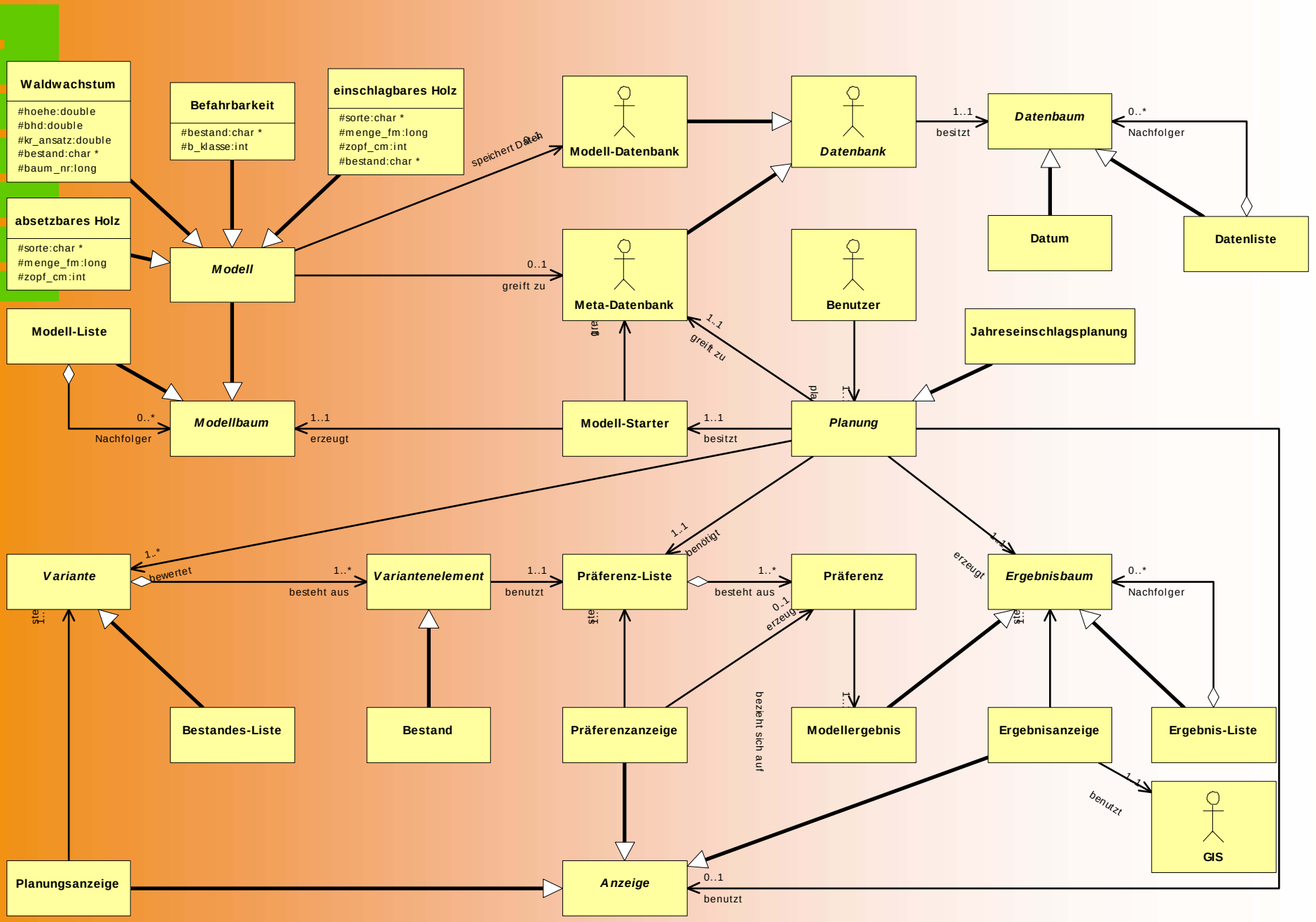


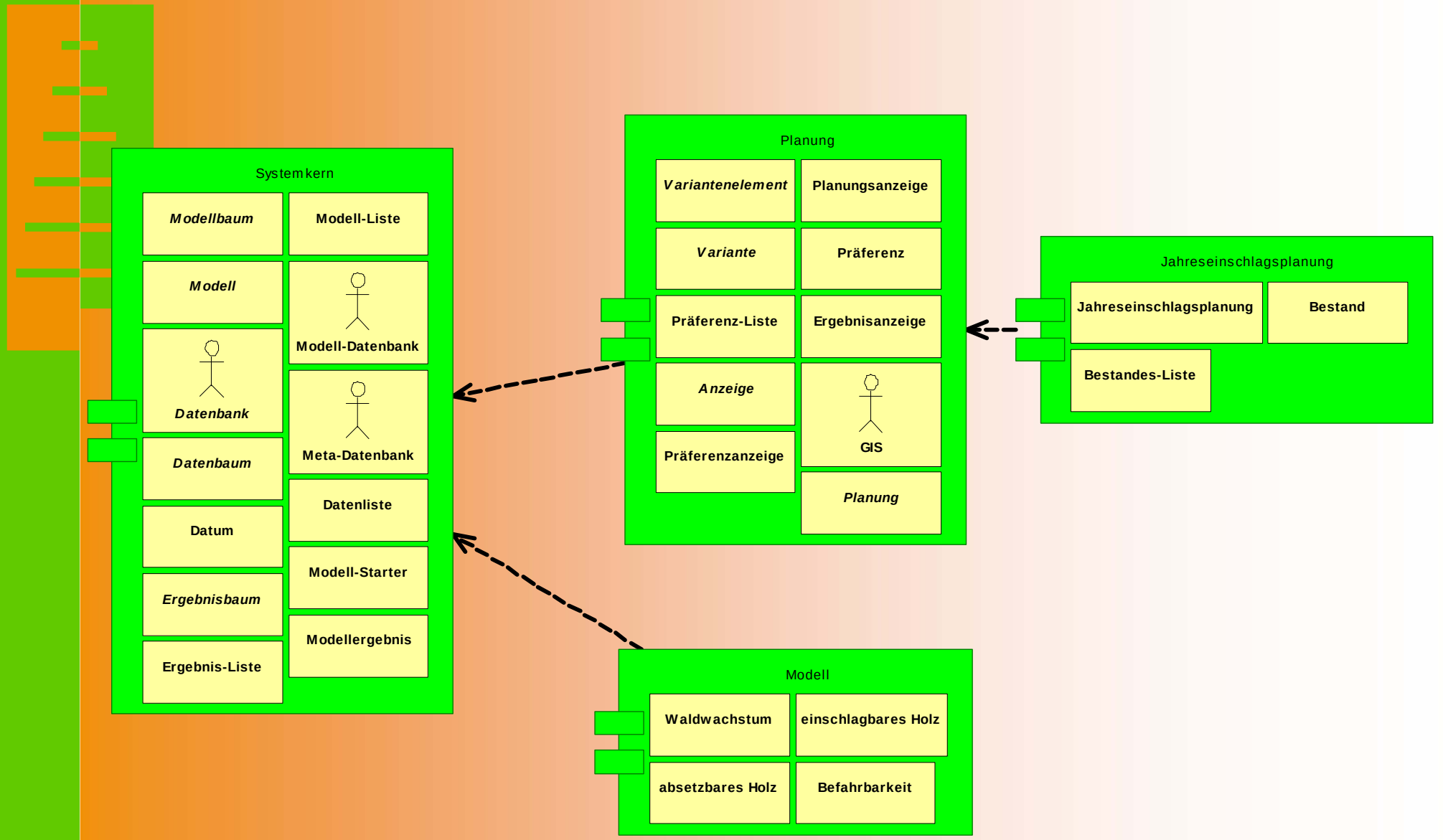
# Integration von Fremdsoftware



- Warum Komponenten?
- Was sind Komponenten?
- Wie geht man vor?
- **Was kommt dabei heraus?**


Nacheinander (bzw. wenn möglich auch parallel) werden die einzelnen Modelle, die am System angemeldet sind, aufgerufen. Sie berechnen die Ergebnisse für die weitere Verwendung.
Bei mehreren Modellen wird die Möglichkeit der parallelen Ausführung geprüft. Wenn möglich, werden Modelle parallel gestartet. Eine Liste mit der Startreihenfolge wird erstellt.
Modelle berechnen
«include»
Berechnungsreihenfolge bestimmen
Der Benutzer wählt die Ergebnisse des Modell-Rechnungslaufes aus, die für seine Entscheidungsfindung die größte Relevanz aufweisen. Gleichzeitig werden diese Größen auch mit Prioritäten versehen.
Benutzer
GIS
Präferenzen bestimmen
Meta-Datenbank
Berechnete Werte können auf Wunsch auch in Form einer Karte visualisiert werden.
GIS Visualisierung
Auf Grund der ausgewählten Zielvariablen und deren Prioritäten wird ein Zielerreichungsgrad berechnet und angezeigt.
Auswahl optimieren
Bestände auswählen
«include»
«extend»
Zielerreichung berechnen
Beschreibung folgt...
Die für die Holzernte relevanten Bestände werden ausgewält.
Ergebnisse durchstöbern
Der Benutzer kann sich einen Überblick über die Ergebnisse verschaffen, die die Modelle ausgerechnet haben.
Modell-Datenbank
Datenbank


Waldwachstum
#hoehe:double
#bhd:double
#kr_ansatz:double
#bestand:char *
#baum_nr:long
absetzbares Holz
#sorte:char *
#menge_fm:long
#zopf_cm:int
Befahrbarkeit
#bestand:char *
#b_klasse:int
einschlagbares Holz
#sorte:char *
#menge_fm:long
#zopf_cm:int
#bestand:char *
Modell
Modell-Liste
Modellbaum
0..*
Nachfolger
1..1
erzeugt
speichert Daten in
Modell-Datenbank
Datenbank
1..1
besitzt
Datenbaum
0..*
Nachfolger
Datum
Datenliste
0..1
greift zu
Meta-Datenbank
1..1
greift zu
Benutzer
Jahreseinschlagsplanung
Modell-Starter
1..1
besitzt
Planung
1..1
benötigt
1..*
bewertet
Variante
1..*
besteht aus
Variantenelement
1..1
benutzt
Präferenz-Liste
1..*
besteht aus
Präferenz
0..1
erzeugt
1..1
erzeugt
Ergebnisbaum
0..*
Nachfolger
Bestandes-Liste
Bestand
Präferenzanzeige
bezieht sich auf
Modellergebnis
Ergebnisanzeige
Ergebnis-Liste
1..1
benutzt
GIS
Planungsanzeige
Anzeige
0..1
benutzt


Systemkern
Modellbaum
Modell-Liste
Modell
Modell-Datenbank
Datenbank
Meta-Datenbank
Datenbaum
Datenliste
Datum
Modell-Starter
Ergebnisbaum
Modellergebnis
Ergebnis-Liste
Planung
Variantenelement
Planungsanzeige
Variante
Präferenz
Präferenz-Liste
Ergebnisanzeige
Anzeige
GIS
Präferenzanzeige
Planung
Jahreseinschlagsplanung
Jahreseinschlagsplanung
Bestand
Bestandes-Liste
Modell
Waldwachstum
einschlagbares Holz
absetzbares Holz
Befahrbarkeit

# Integrationsplattform

- Flexible, Client/Server-fähige Software-Architektur
- Einheitliche Datenbasis
- Möglichkeit der Parallel-Verarbeitung
- Austauschbarkeit von Komponenten, die zu den Spezifikationen konform sind

# Danke...

... für Ihre Aufmerksamkeit!