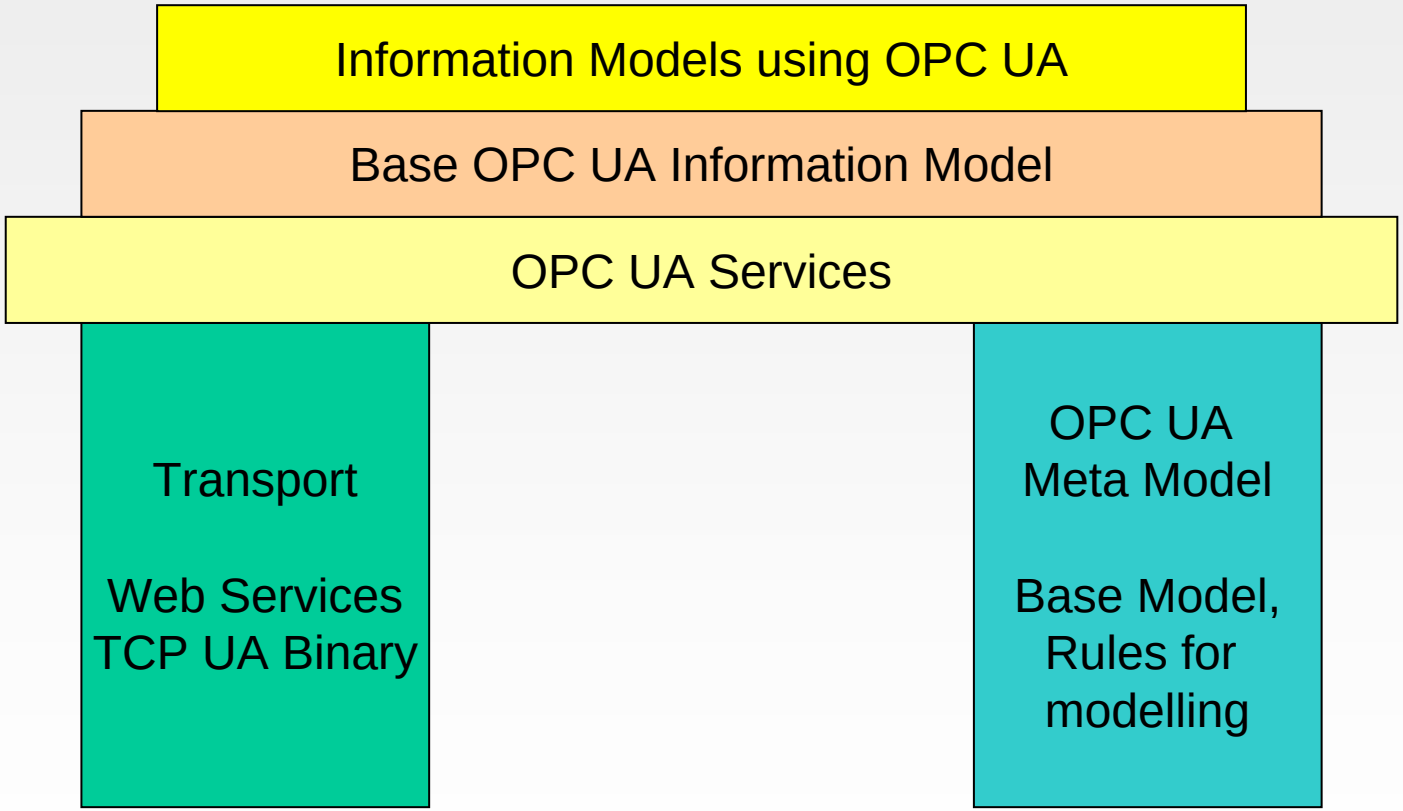
# Middleware für verteilte industrielle Umgebungen

## OPC UA

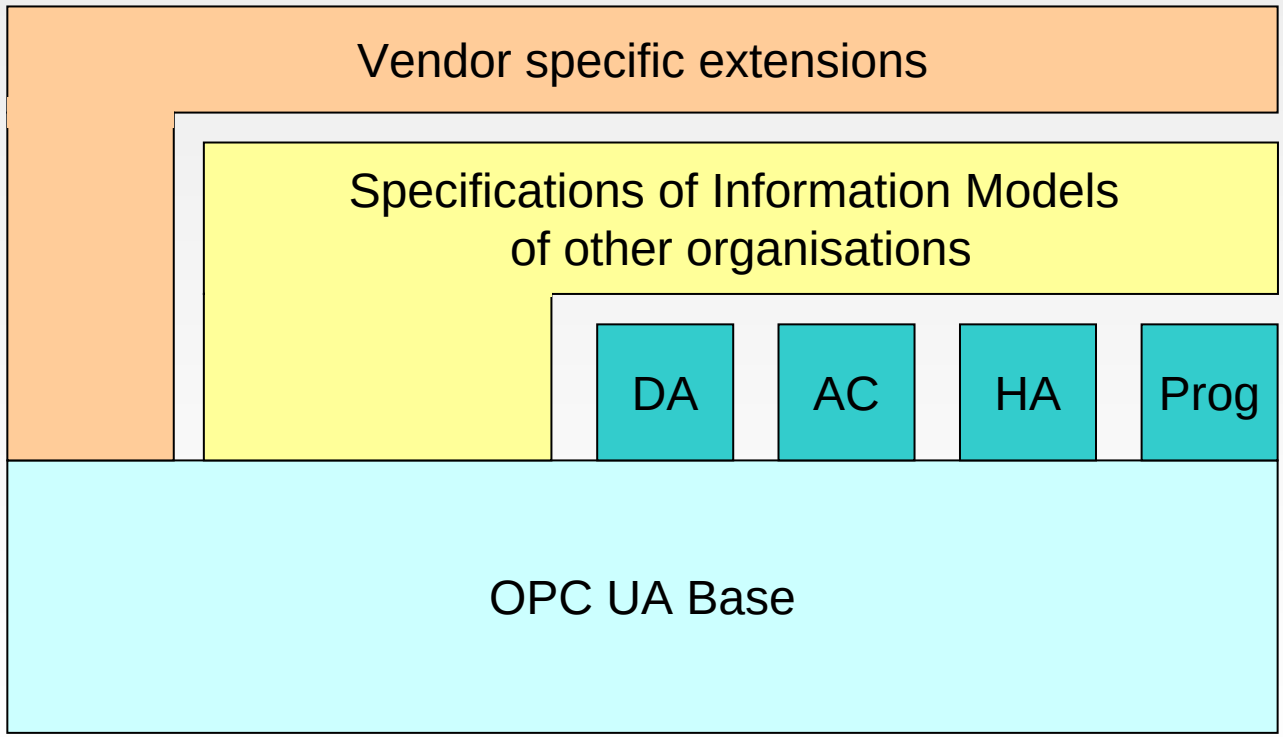


ifak

# Überblick
## Basisbausteine von OPC UA



# Überblick
## Schichten von OPC UA



# Architektur Spezifikation

- Core Specifications
  - Part 1 – Concepts
  - Part 2 – Security Modell
  - Part 3 – Adress Space Model
  - Part 4 – Services
  - Part 5 – Information Model
  - Part 6 – Service Mappings
  - Part 7 – Profiles



ifak

# Architektur Spezifikation (ff)

- Access Type Specifications
  - Part 8 – Data Access
  - Part 9 – Alarm and Conditions
  - Part 10 – Programs
  - Part 11 – Historical Access
  - Part 13 – Aggregates

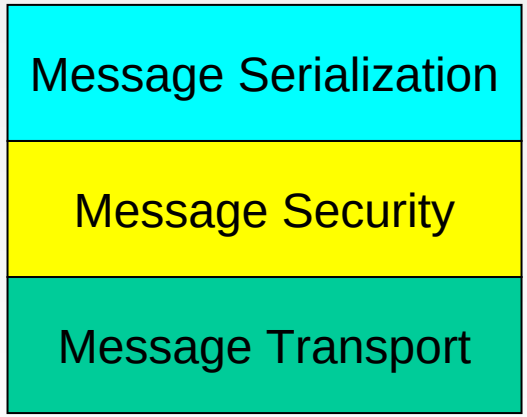
  - Part 12 – Discovery



ifak

# Bindings

- Part 6
  - Web Service
  - UA TCP Binding

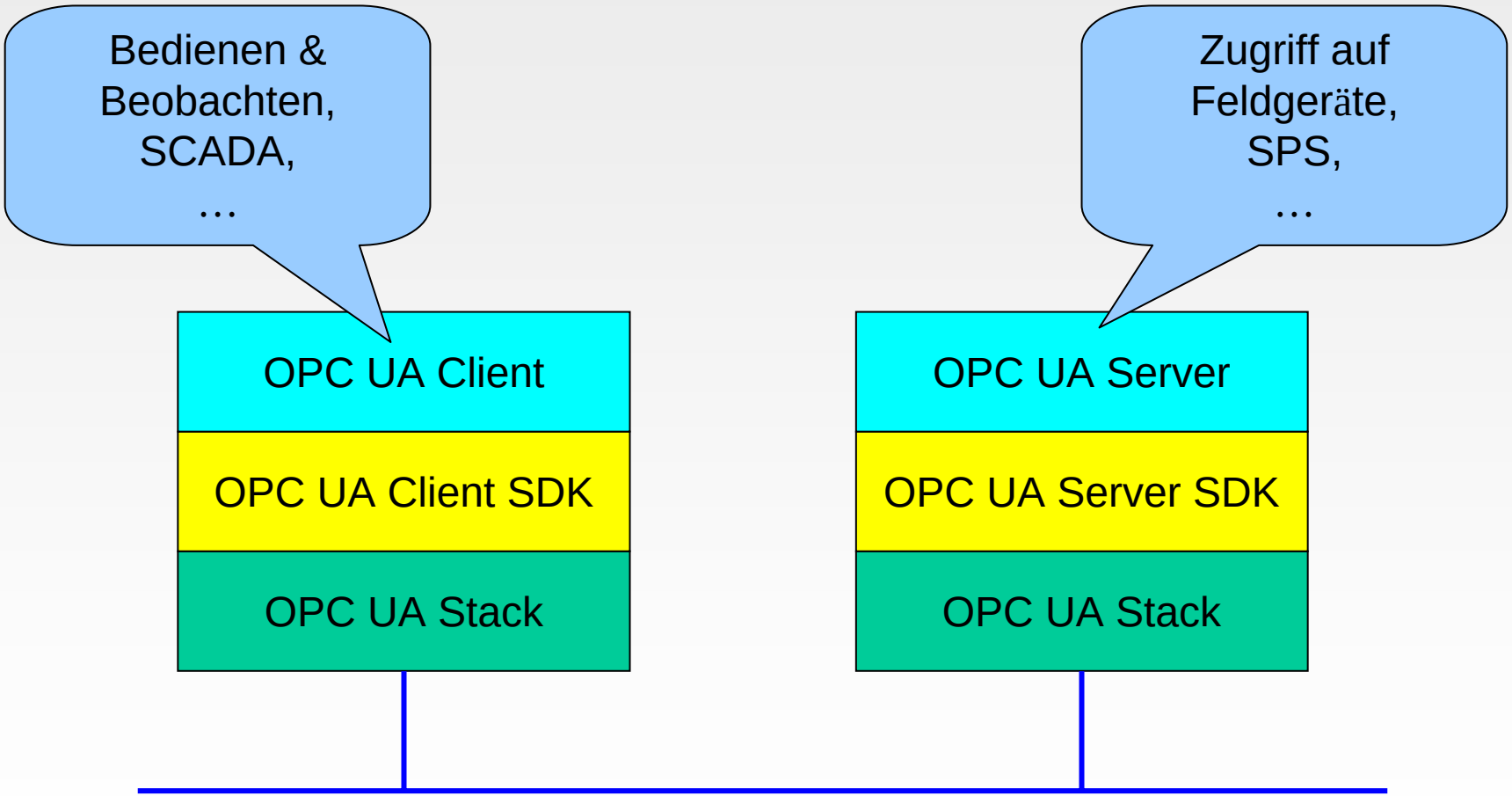



- Part 4
  - Abstract UA Service Specification

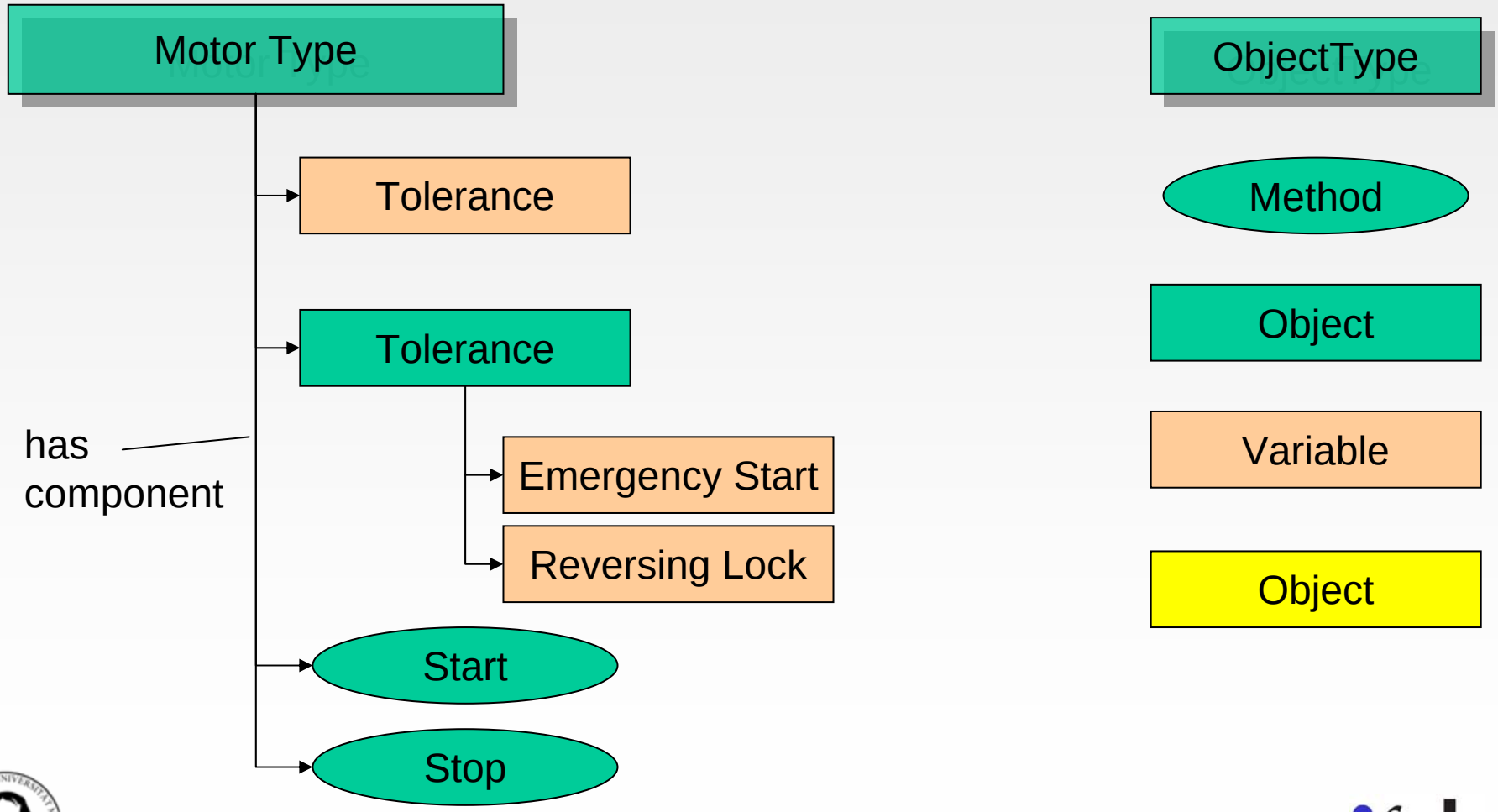


ifak

# Kommunikation: Schichten



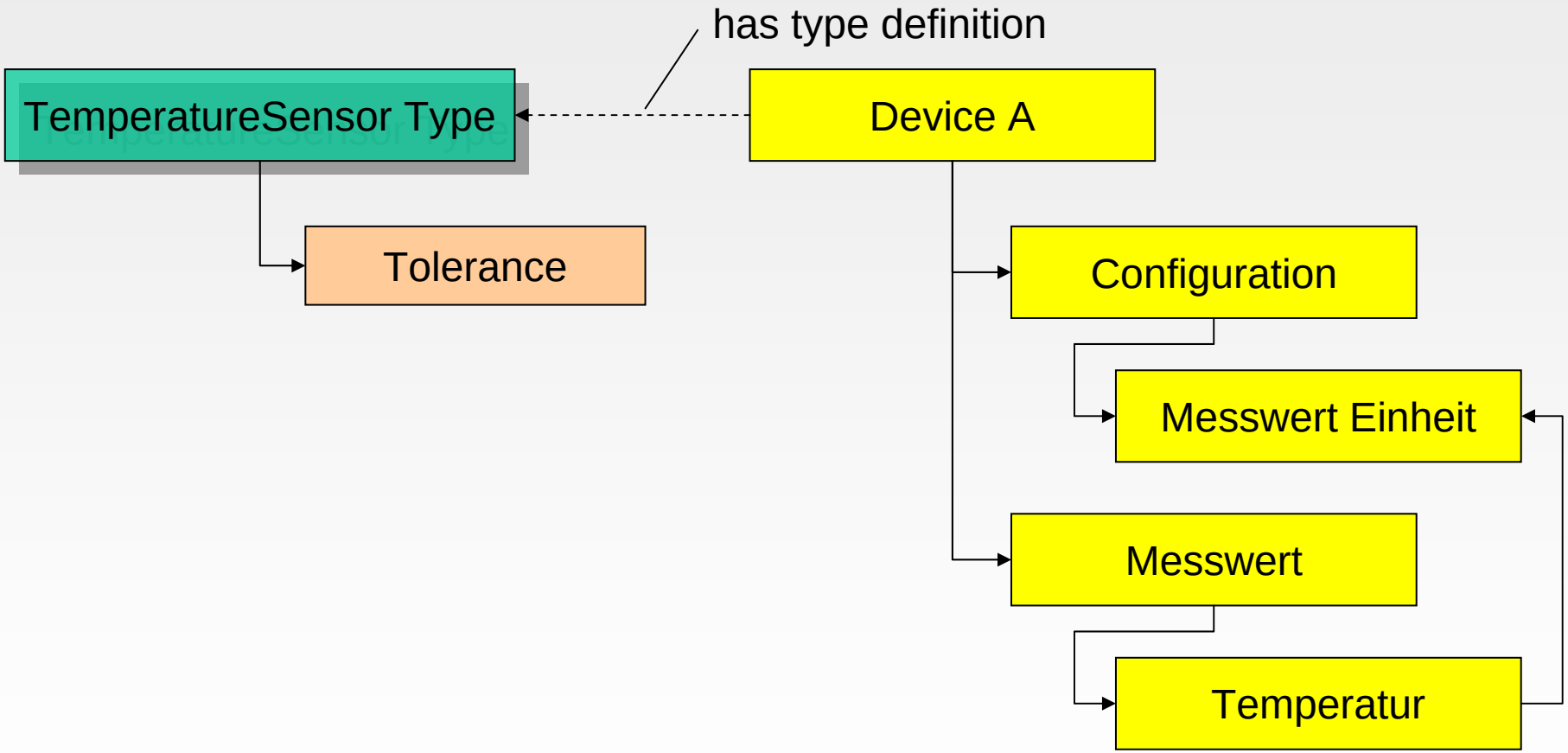
# Complex ObjectType



# Information Modeling
## Beispiel Temperaturtransmitter

- Konfiguration
  - Messwert Einheit
- Messwert
  - Temperatur, Status
- Sonstige Informationen
  - Toleranz
  - Obere / untere Grenze
  - …

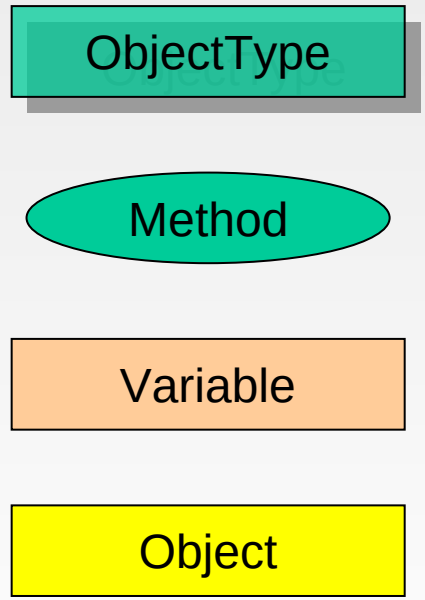
# Information Model
## Typ- und Instanzinformationen



# Mapping OOD – ObjectType

```
class Motor
{
public:
  std::string name;
  int power;

  void Start()
  {
    …
  }
}
```



# Sonstige Regeln

- Verschiedene Beziehungstypen unter den Typ-Knoten
  - … has component …
  - … organizes …
  - … is used by …
  - … has property …
  - … has event source …
- Knoten können optional sein (in C++: Pointer of NULL)



ifak

# Datentypen

- Boolean
- ByteString ← Image ← {…Bmp, …JPG, …GIF, …PNG}
- QualifiedName
- String ← LocalId
- GUID
- DateTime ← UtcTime
- DataValue
- DiagnosticInfo
- LocalizedText
- Number ← {Double, Integer ← {}, Float, UInteger ←{}}
- ExpandedNodeId
- XMLElement
- NoteId



ifak

# Sichten

- Organisiert über den Adressraum
- ViewNode verweist auf ‚sichtbare' Knoten
- Es kann mehrere ViewNodes im Adressraum geben → überlappende Sichten sind möglich
- Sichten werden durch Server definiert



ifak

# Implementierung des Information Model

- Definition mittels XML-Schema
- Generierung der C# - Quellen mittels Compiler der OPC-Foundation → Typen der Knoten sind damit bekannt
- Bildung der Instanzknoten
  - Programmierung
  - Konfiguration
  - Dynamisch zur Laufzeit → browsen unterlagerter Geräte / Komponenten



ifak

# Ereignisse

- Empfangen vom Client via Notifications
- Subscribing via EventNotifier
-



ifak

# Quelle

- OPC Foundation: Spezifikationen
- W. Mahnke; S. Leitner; Matthias Damm: OPC Unified Architecture



ifak