

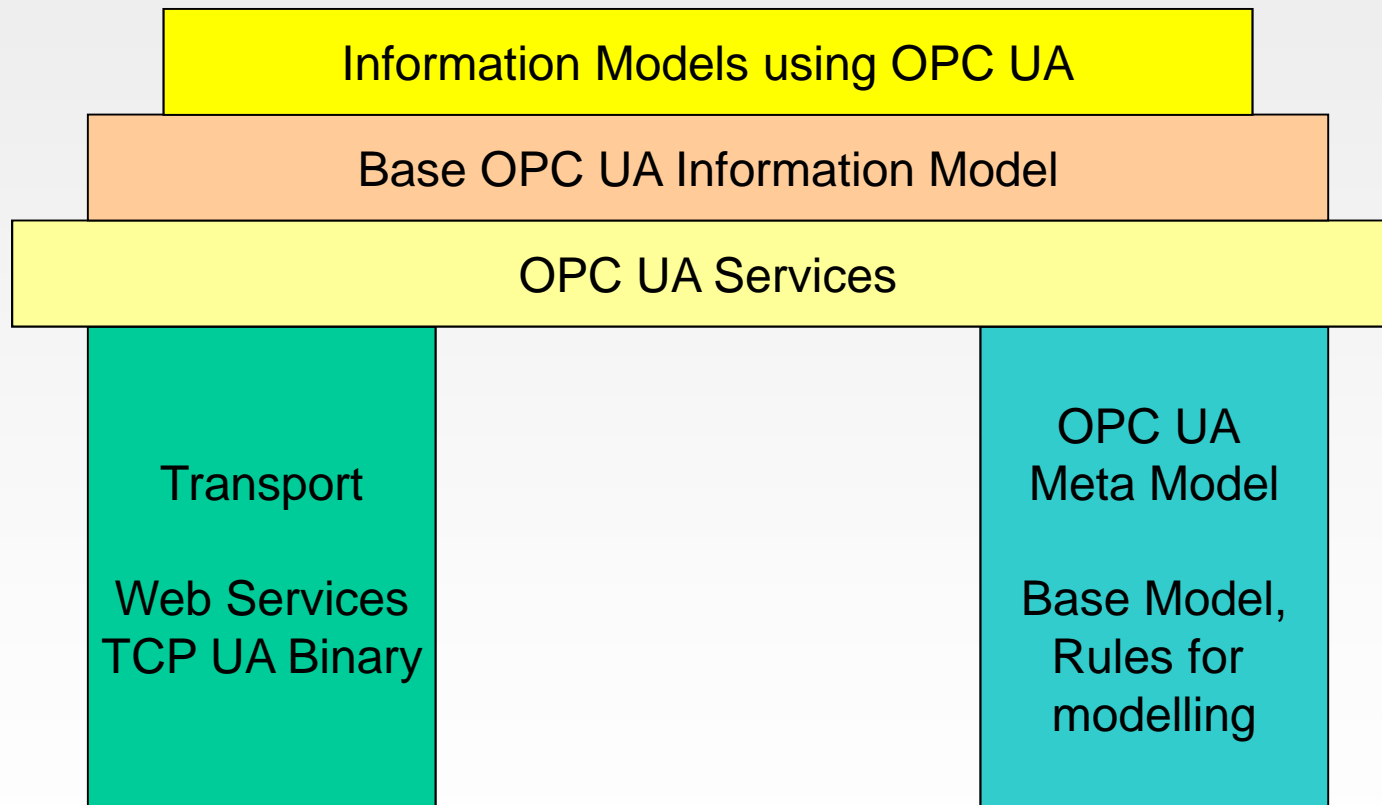
Middleware für verteilte industrielle Umgebungen

OPC UA



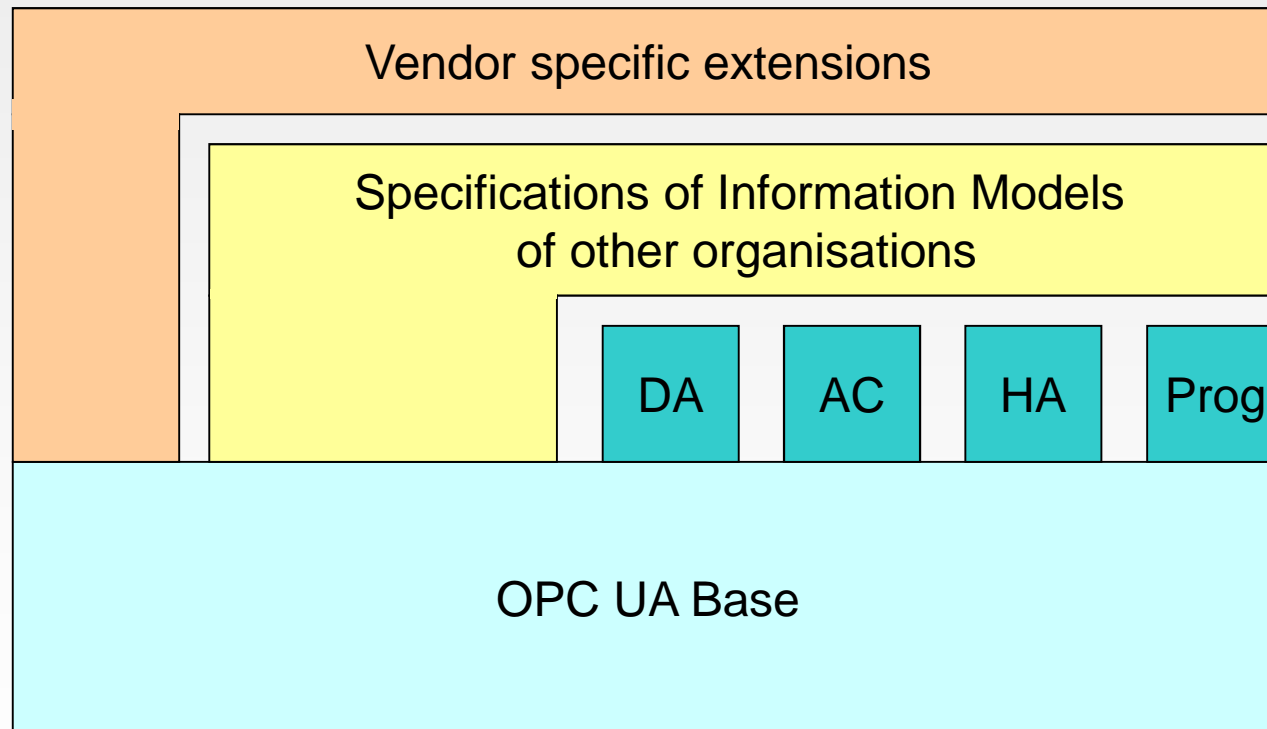
Überblick

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



Architektur Spezifikation (ff)

□ Access Type Specifications

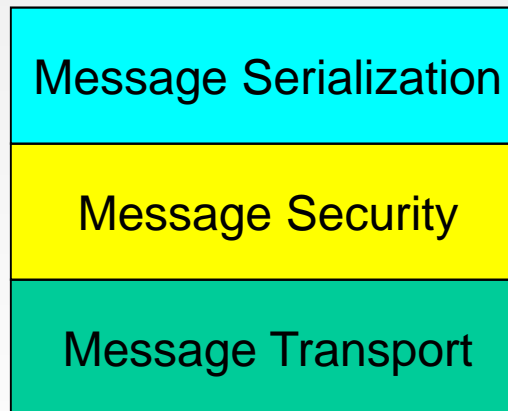
- Part 8 – Data Access
- Part 9 – Alarm and Conditions
- Part 10 – Programs
- Part 11 – Historical Access
- Part 13 – Aggregates

- Part 12 – Discovery



Bindings

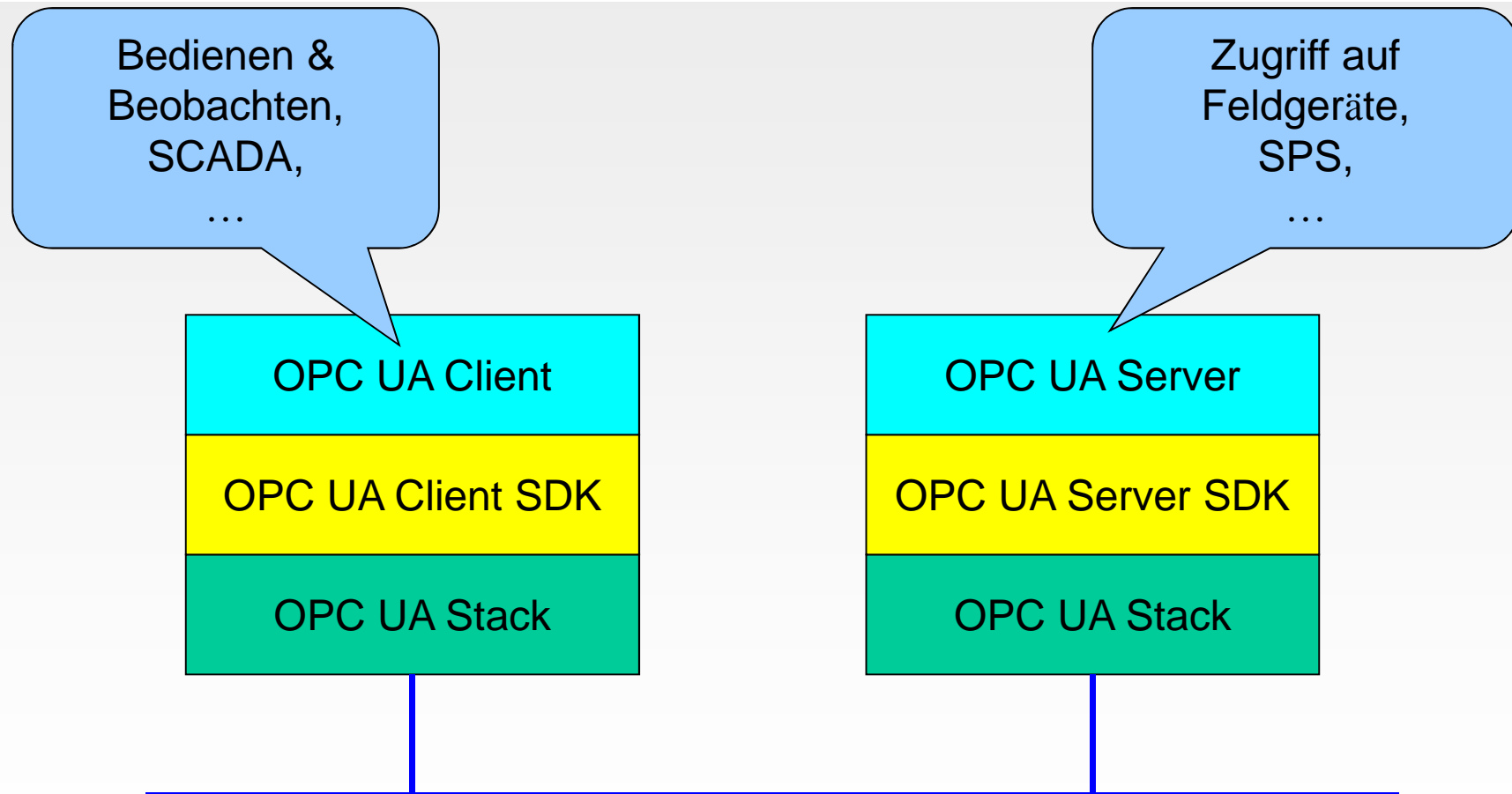
- Part 6
 - Web Service
 - UA TCP Binding



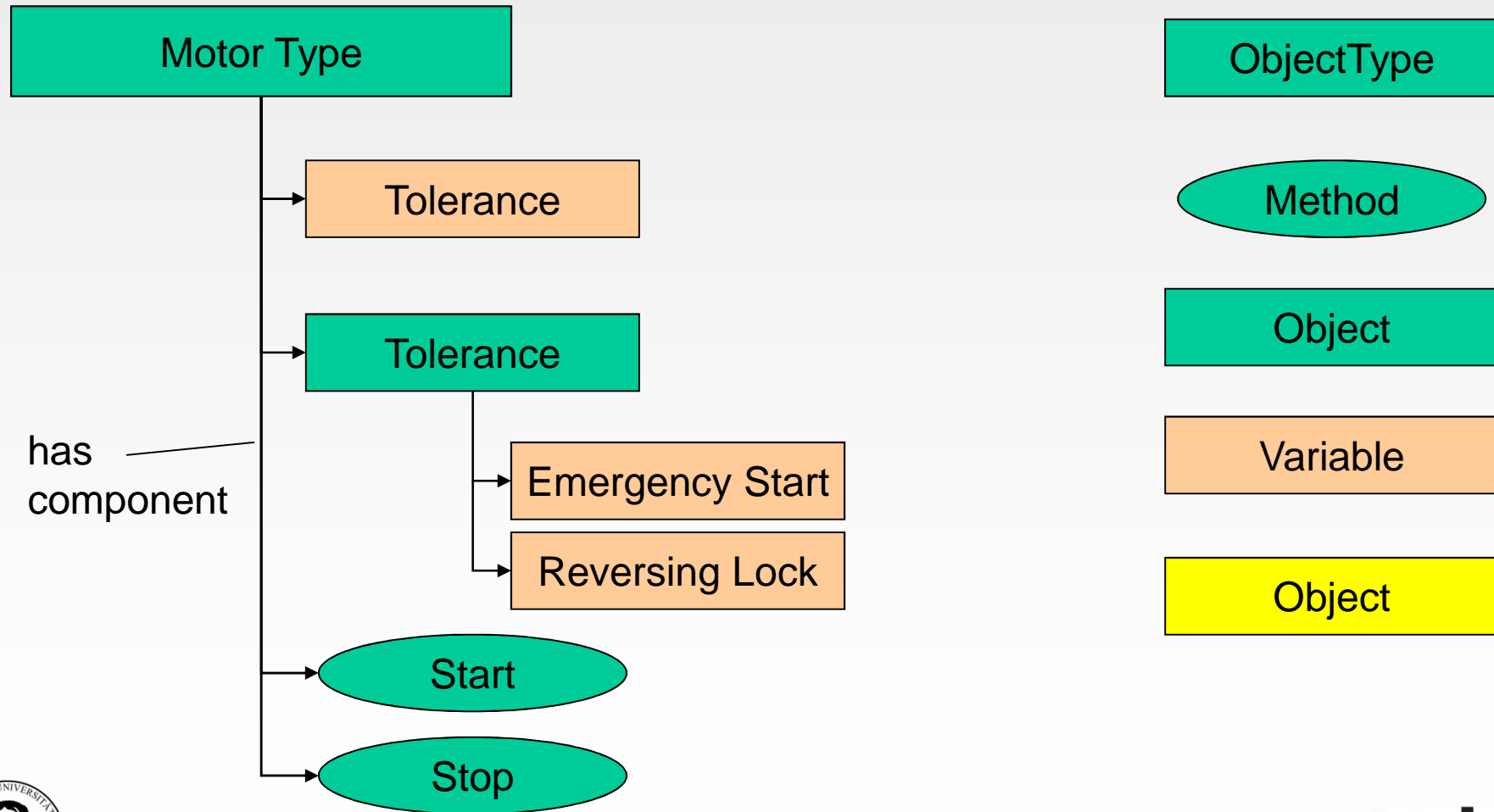
- Part 4
 - Abstract UA Service Specification



Kommunikation: Schichten



Complex ObjectType



Information Modeling

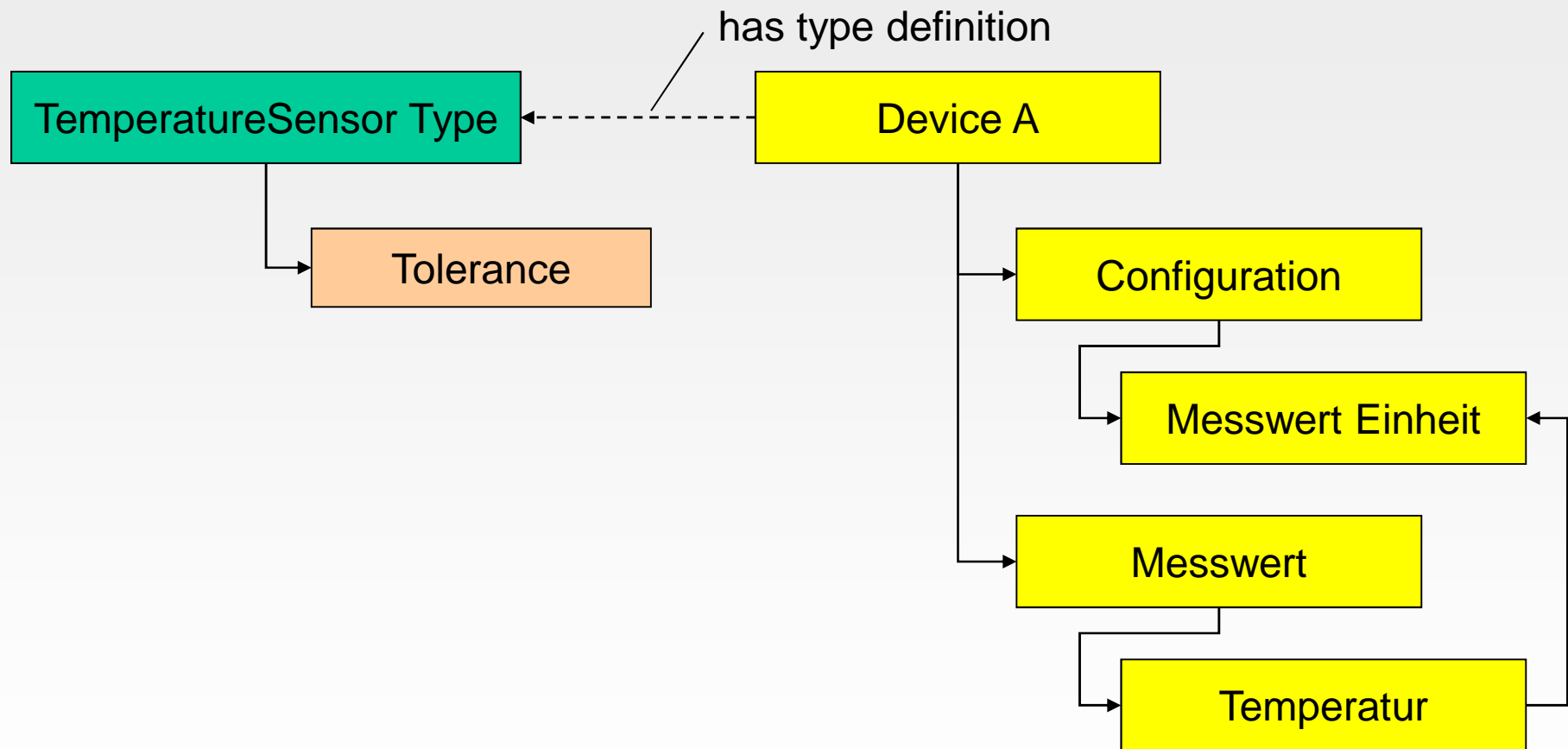
Beispiel Temperaturtransmitter

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



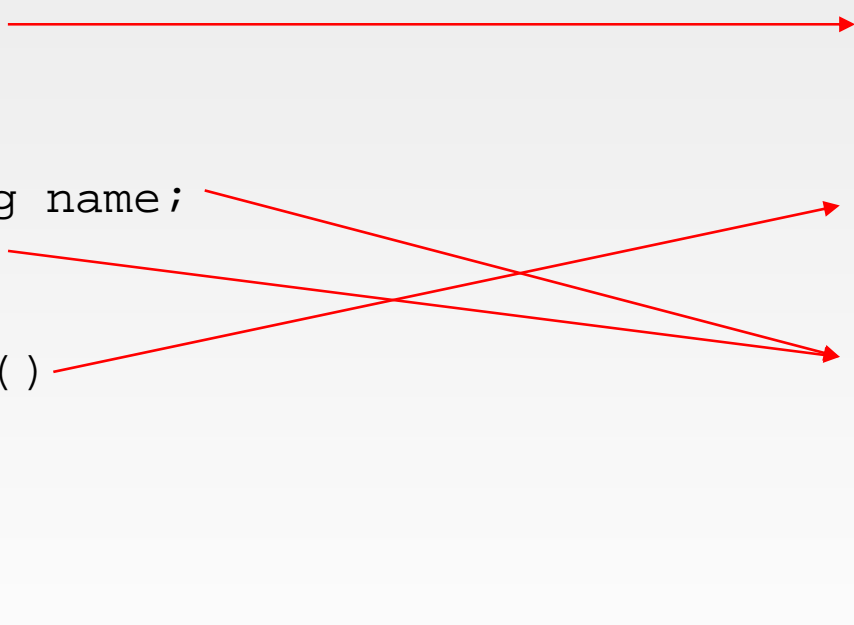
Information Model

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)



Datentypen

- Boolean
- ByteString \leftarrow Image \leftarrow {...Bmp, ...JPG, ...GIF, ...PNG}
- QualifiedName
- String \leftarrow LocalId
- GUID
- DateTime \leftarrow UtcTime
- DataValue
- DiagnosticInfo
- LocalizedText
- Number \leftarrow {Double, Integer \leftarrow {}, Float, UInteger \leftarrow {}}
- ExpandedNodeId
- XMLElement
- NotId



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



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



Ereignisse

- Empfangen vom Client via Notifications
- Subscribing via EventNotifier
-



Quelle

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

