

Datenintegration mit XML

Mario Jeckle

mario.jeckle@daimlerchrysler.com

mario@jeckle.de

www.jeckle.de

DaimlerChrysler Research

dept. product development process chain (FT3/EK)

Inhaltsübersicht

Datenintegration

- I Notwendigkeit
- II Herausforderungen und aktuelle Problemstellungen
- III *klassische* Lösungsansätze
- IV XML als Komponente einer Lösungsarchitektur

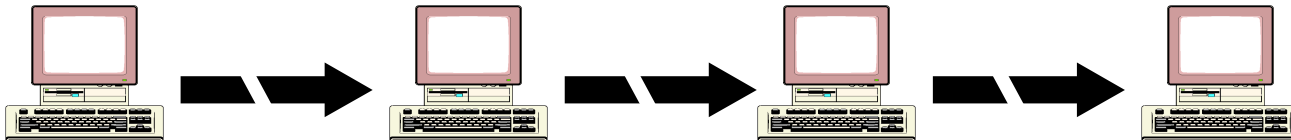
eXtensible Markup Language (XML)

- V Modellbasierter Ansatz
- VI Online Kopplung – *XML und CORBA - Konkurrenz oder Kooperation*
- VII Zusammenschau

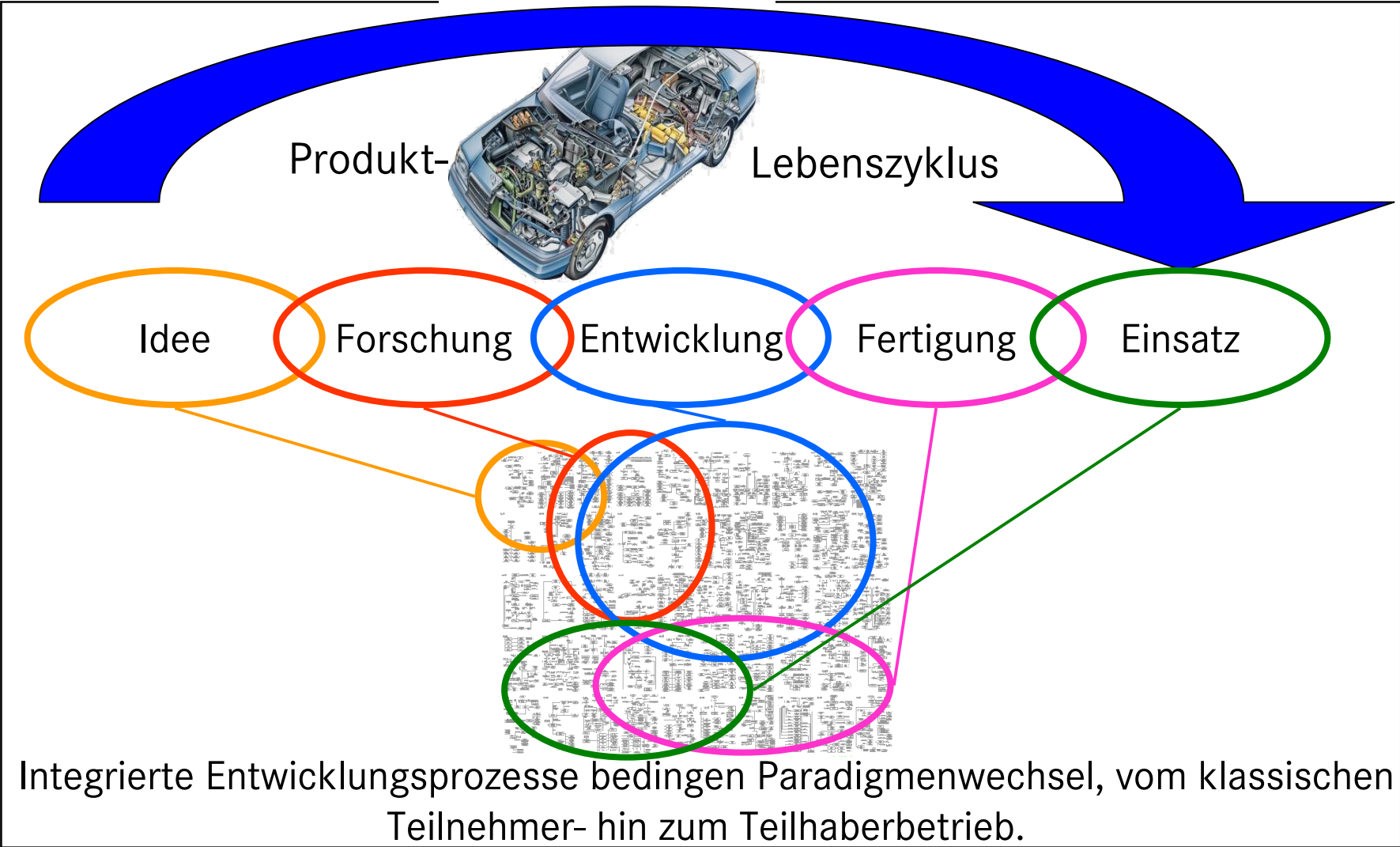


Verteilte virtuelle Prozessketten

Informationsfluß

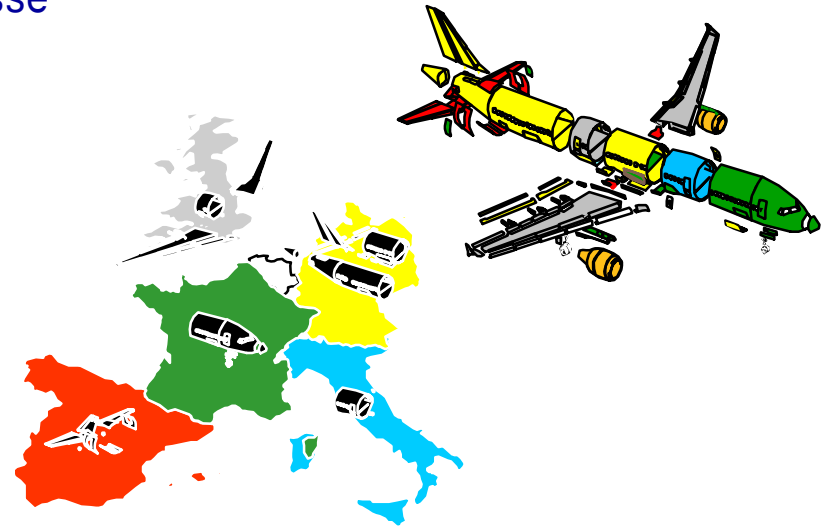


Dynamische verteilte virtuelle Prozesse und Prozeßketten bilden das erfolgskritische Rückrad moderner Unternehmen



Rahmenbedingungen moderner Produktenstehungsprozesse

- Daten sind **partnerweit** (weltweit) **verteilt**
- **Prozesse** sind **partnerübergreifend** (Configuration- / Changemanagement)
- Partner brauchen **unterschiedliche Sichten** auf gleiche Daten
- organisieren die Daten intern entsprechend ihrer Aufgabe unterschiedlich
- haben **unterschiedliche Rechte/Befugnisse**
- setzen **unterschiedliche Systeme** ein
- benutzen **unterschiedliche Sprachen**
- arbeiten in **unterschiedlichen Zeitzonen**
- müssen **flexibel eingebunden** werden
- Arbeiten in **unterschiedlichen Entwicklungsabschnitten**



Rahmenbedingungen moderner Produktenstehungsprozesse

„klassisch“:

- Organisation in zeitlich horizontaler Sicht
- geringer übergreifender Integrationsgrad
- eng umrissene Zulieferer- und Partner Integration

„modern“:

- Nutzung vorhandenen Rationalisierungspotentials durch (massive) Verteilung der Bauanteile
- Synchrone unabhängige Entwicklung der Einzelbaugruppen
- Hoher Interaktionsaufwand in allen Entwicklungsstufen
- dynamische (Unternehmens-)Verbünde und Partnerschaften

klassische Lösungsansätze

“Ein-System”-Strategie



Vorteile:

- (a priori) keinerlei Konvertierungsproblematik durch alleinige Nutzung des nativen Applikationsformates

Nachteile:

- Mögliche Quelle hemmender Inflexibilität
- Verteilungsproblematik

Format Konvertierung

Vorteile:

- Systemflexibilität, hoher vertikaler Optimierungsgrad

Nachteile:

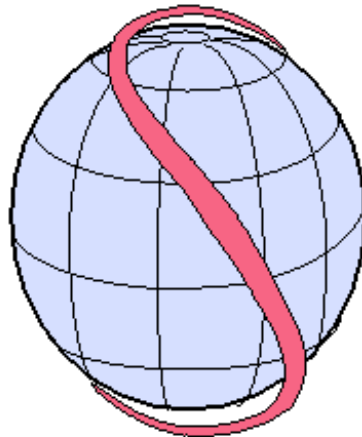
- möglicher Informationsverlust
- Konverterbereitstellung und -Wartung



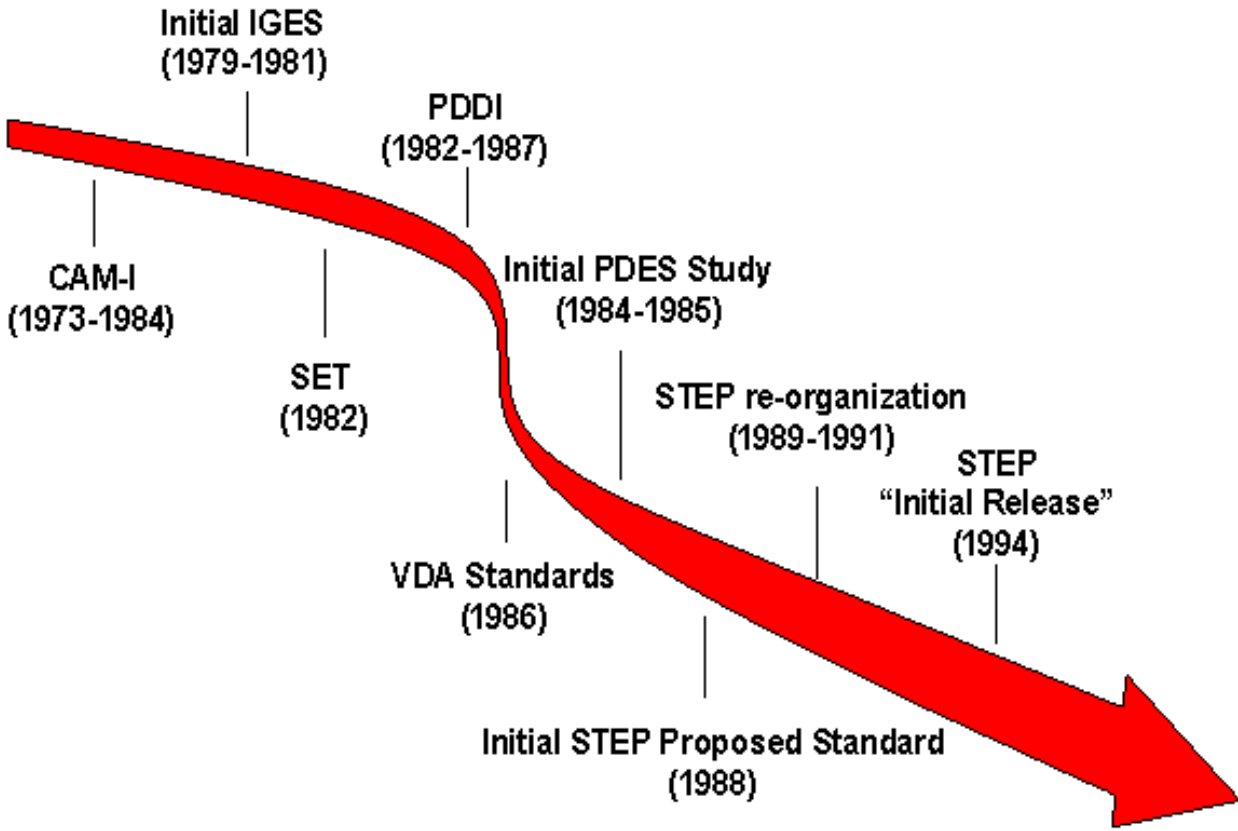
moderne Lösungsansätze – STEP

STEP:

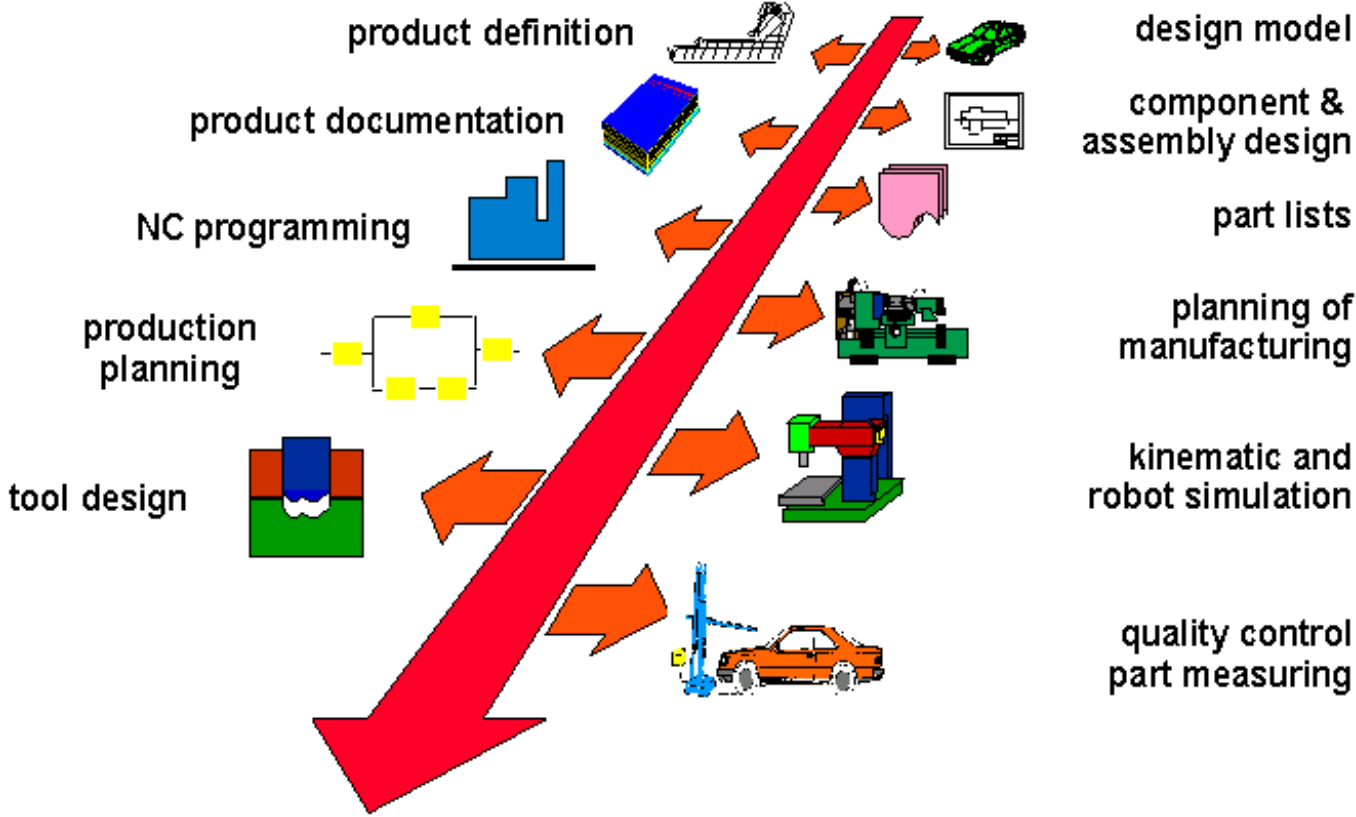
- Standard for the Exchange of Product Model Data
- internationaler Industriestandard entwickelt durch ISO TC184/SC4- “Industrial Data”, unterstützt:
 - Informationsaustausch zwischen engineering Applikationen
 - Langzeitarchivierung von Produktinformation
 - Umsetzung verteilter Produktdatenbanken
- ISO 10303 “Product data representation and exchange”, Part 28



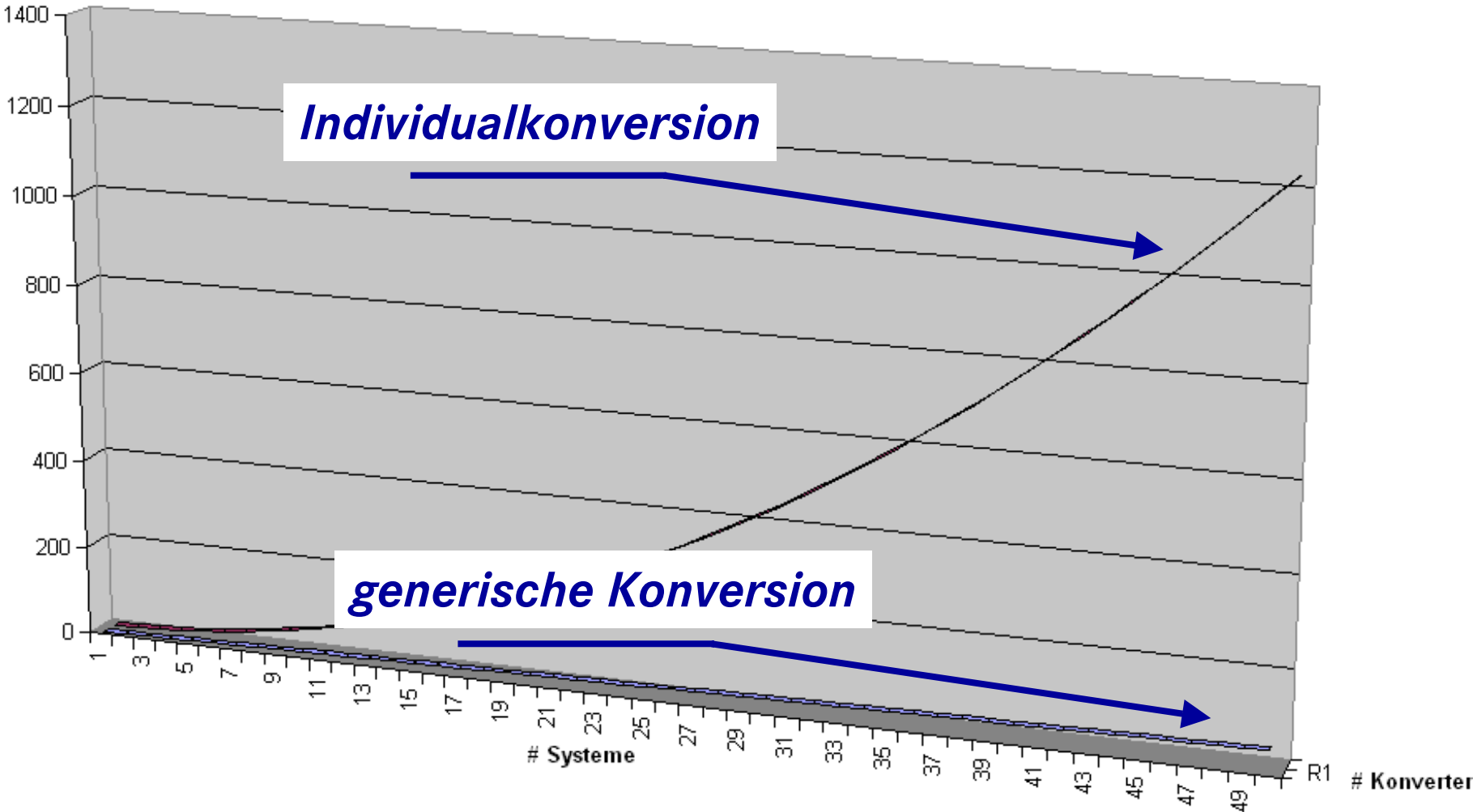
moderne Lösungsansätze – STEP



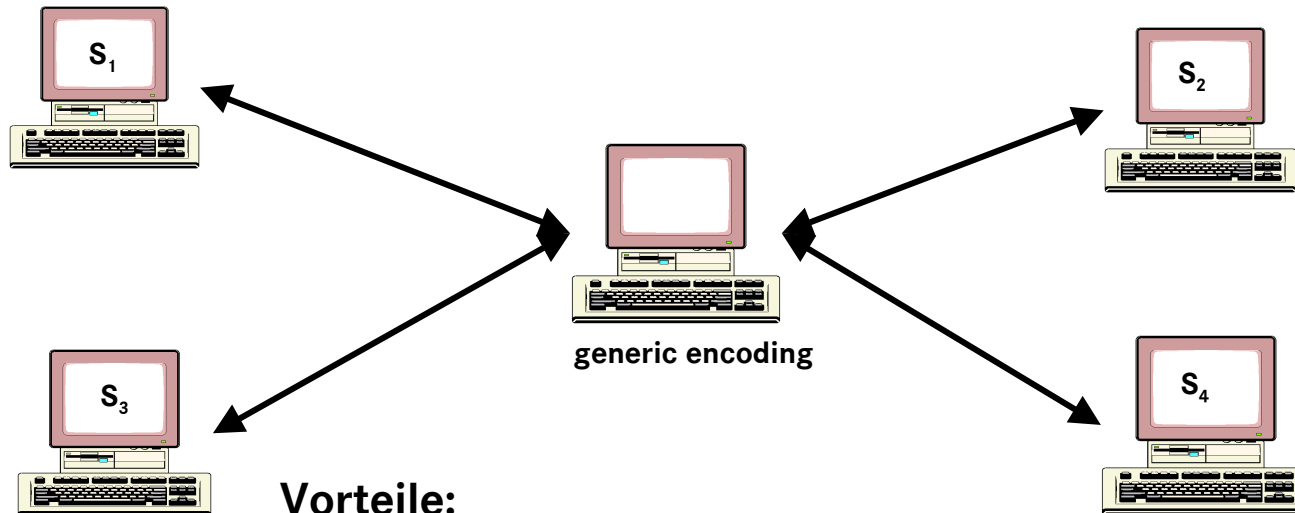
moderne Lösungsansätze – STEP



Konvertierungsproblematik



Generische Inhaltsformate

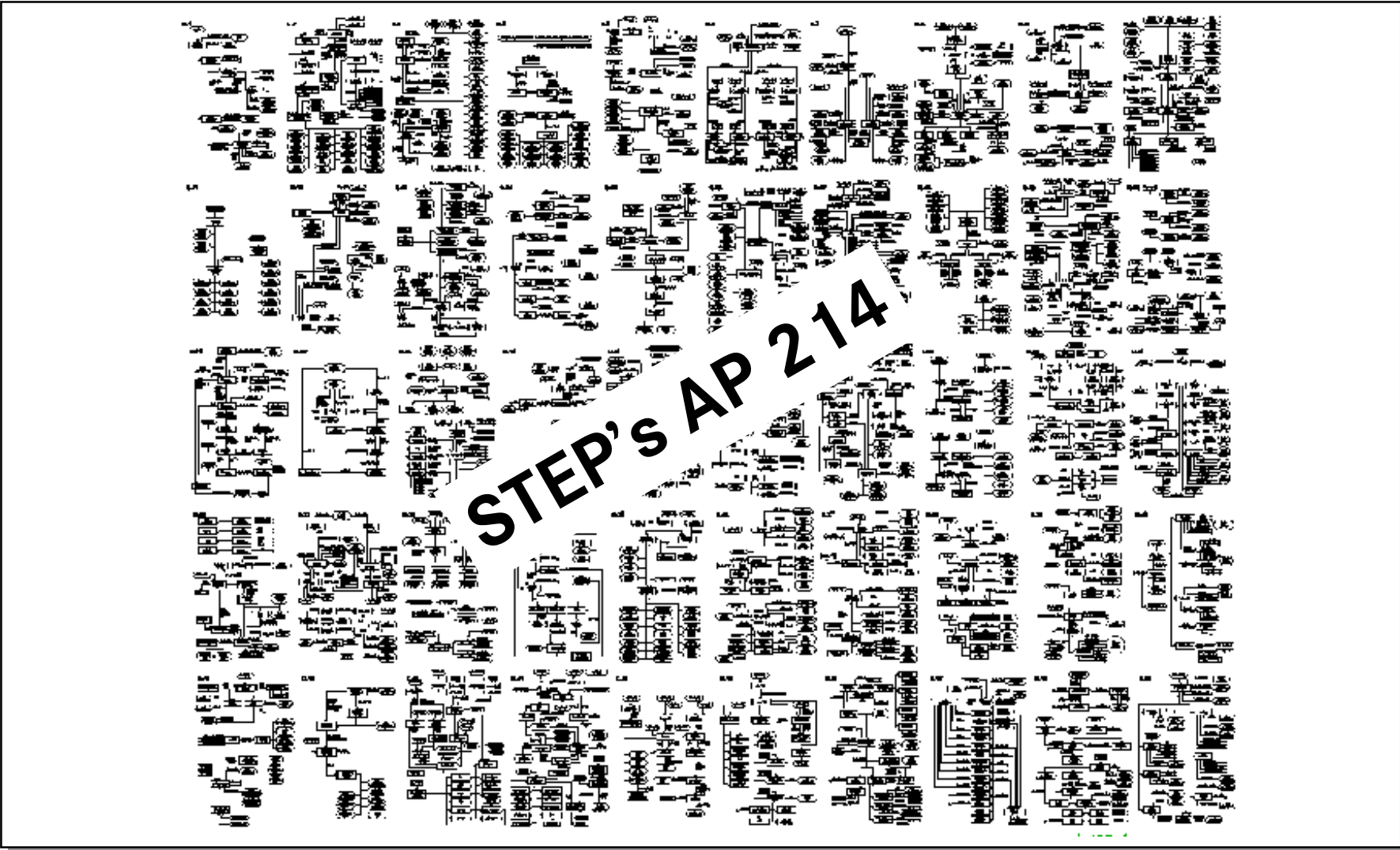


Vorteile:

- massive Reduktion der notwendigen Konvertierungsvorgänge (n^2-n vs. n)

Nachteile:

- Informationsverlust durch Generalisierungsprozess
- Codierungsaufwand der Transformationsprozessoren



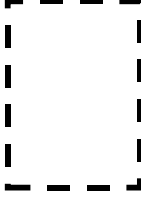
Die eXtensible Markup Language

Clear Text Encoding
of exchange Structures

ISO
10303-28

ISO-STEP
EXPRESS

Other RDF
applications



Platform for Internet
Privacy Preferences

P3P

RDF
(Resource Description Framework)

Platform for Internet
Content Selection

PICS

Schema language
for XML

XML-Schema

Multimedia data

SMIL

Vector Graphics

SVG

Hypertext
Documents

XHTML

Metadata
Interchange

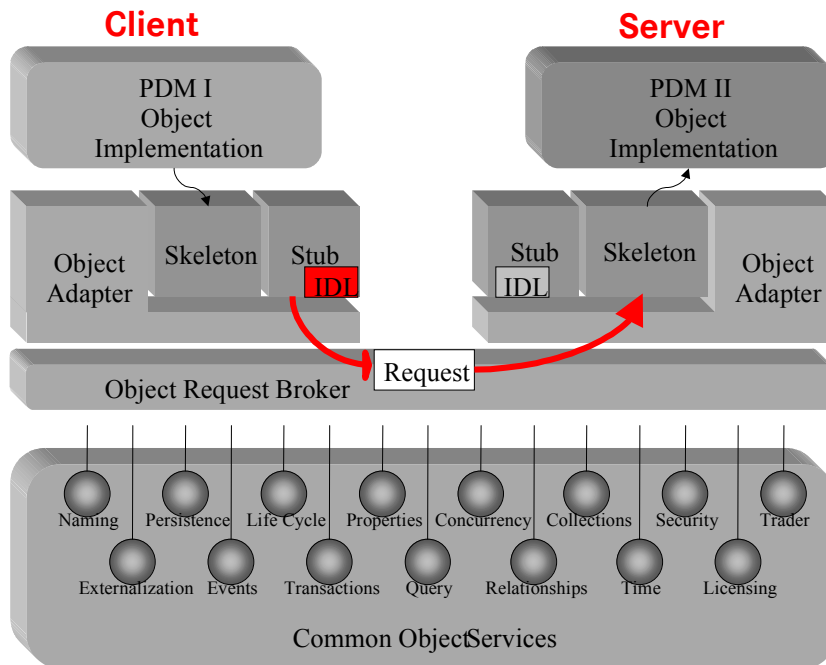
XMI

eXtensible Markup Language

UML

MOF

Die Common Object Request Broker Architecture



- Architektur und Spezifikation zur Erzeugung, Verteilung und Verwaltung verteilt vorliegender Objekte
- Verschiedene Standort kommunizieren über ein einheitliche Vermittlungszentrale, den *Object Request Broker*
- *CORBA* ist programmiersprachen- und netzwerkprotokollunabhängig
- Die Kommunikation wird über ein standardisiertes Protokoll (GIOP/IIOP) abgewickelt; Datenpakete werden Klartext codiert

| | Capability | Description | XML | CORBA |
|---|---|---|---|--------------|
| <p>XML as a Distributed Application Protocol</p> | Plattform | Can the client | Yes | Yes |
| | language independent | and server be on almost any operating system, hardware, and application language? | | |
| | Handle human readable text | Can the document be viewed and edited with low-tech tools? | Yes | No |
| Handle non-string data types | Is there native for integers, floating point numbers, booleans, etc.? | Not natively, needs DCD or XML-Schema | string, short, int, float, double, boolean, byte, char, enums, structs, union | |

XML as a Distributed Application Protocol (cont'd)

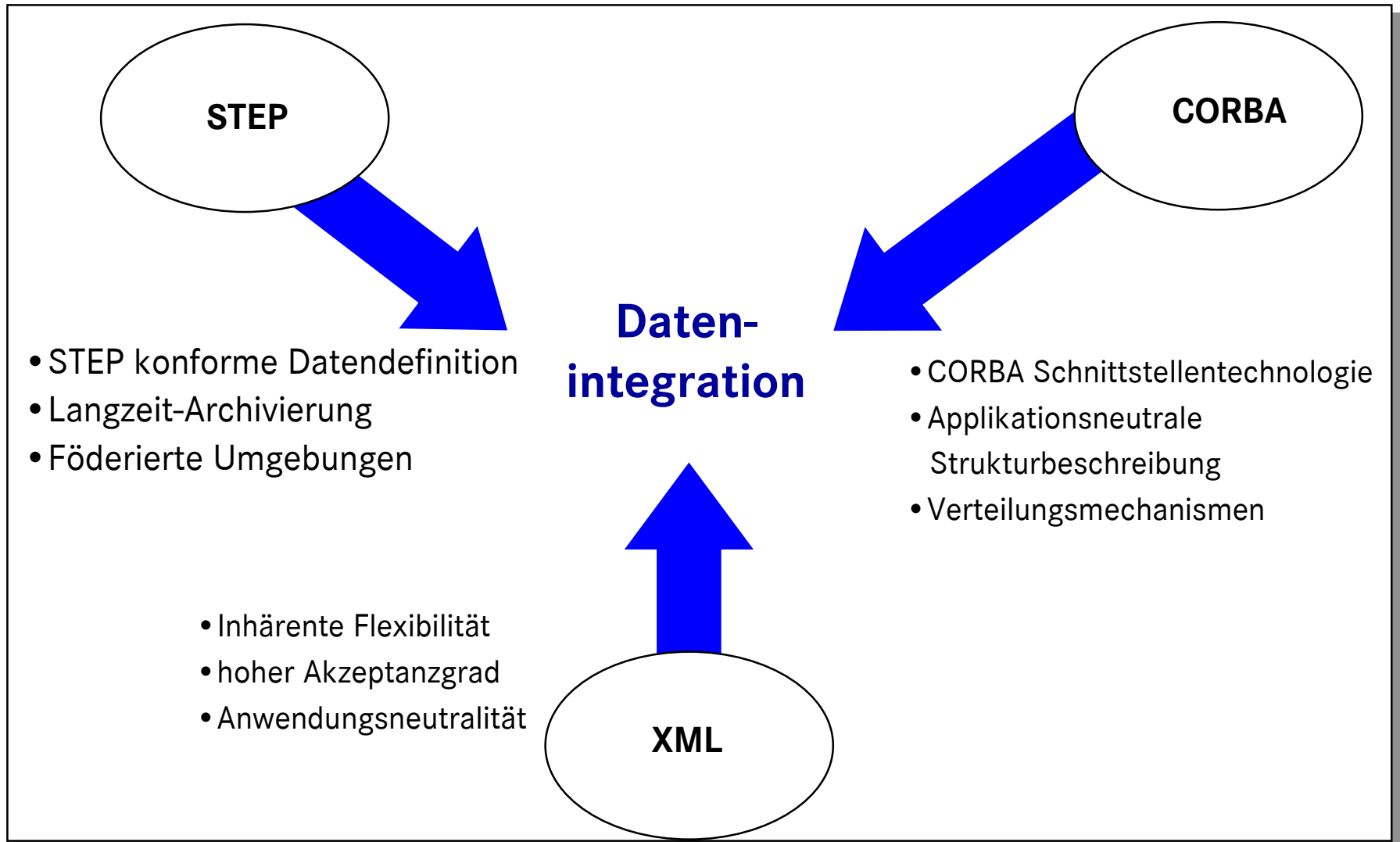
| Capability | Description | XML | CORBA |
|--------------------------|---|--|--------------|
| Huge data set management | Can the data be larger than the allowable application memory? | Yes | No |
| Schema versioning | Can an application handle two data streams that were created using different versions of the schema definition? | Yes | No |
| Distribution support | <ul style="list-style-type: none"> • transparency • remote messaging • lookup facilities • security • transactions, etc. | none (simple messaging can be archived by parsing transmitted XML documents) | full |

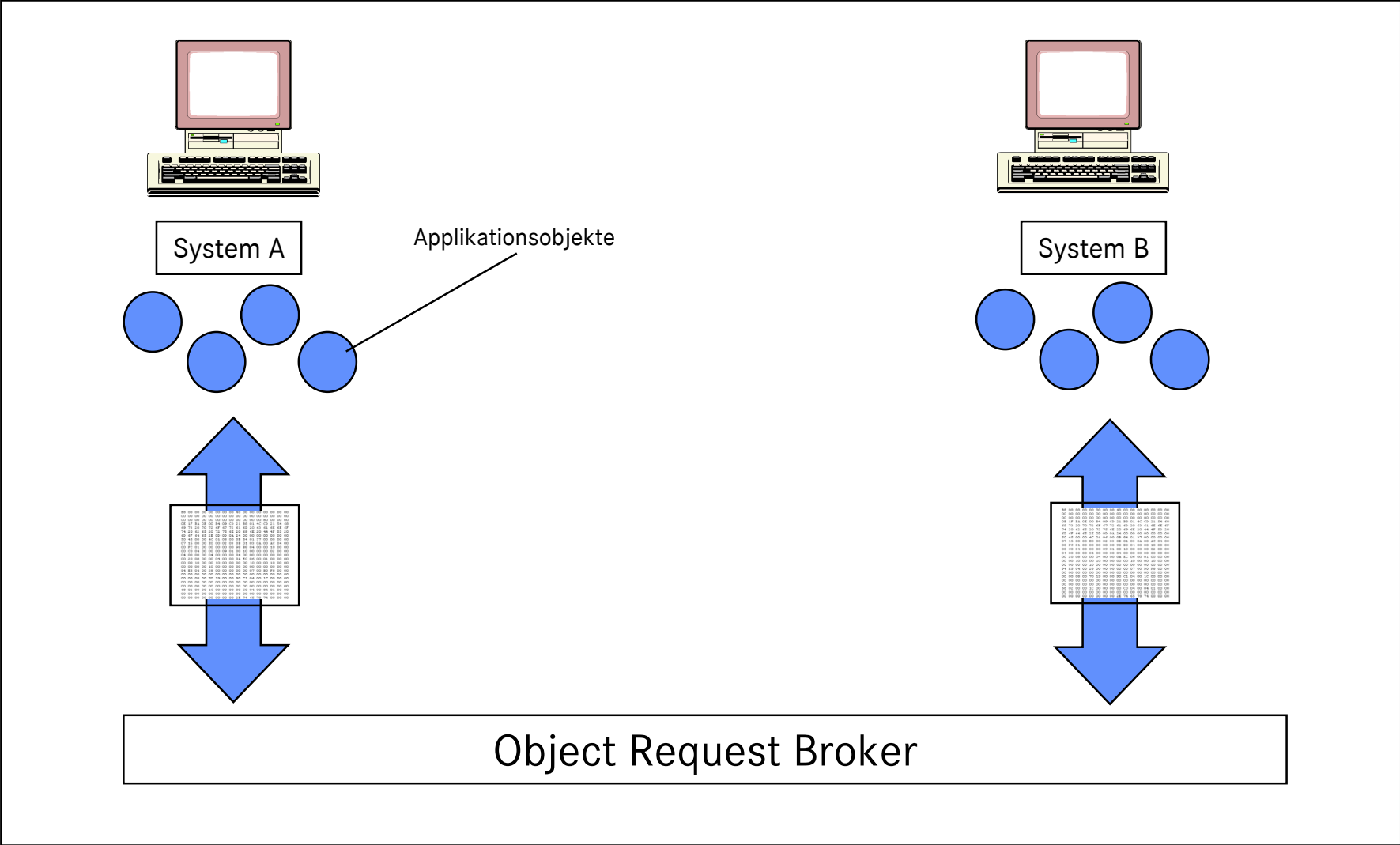
XML as a Distributed Application Protocol (cont'd)

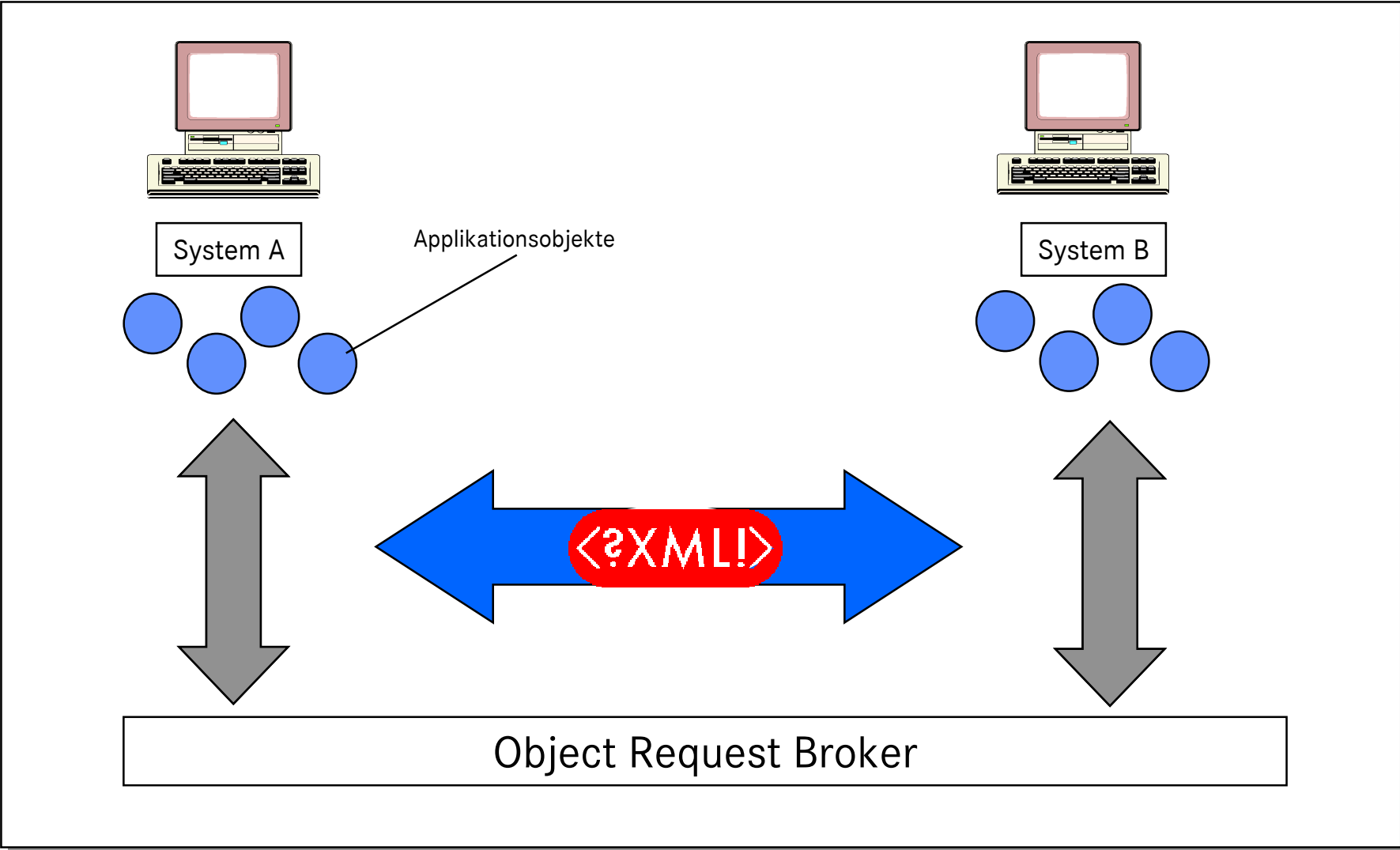
| Capability | Description | XML | CORBA |
|-----------------|--|--|---|
| Object-oriented | Can the data structure have behavior associated with it? Is inheritance supported? | No. (Inheritance is supported by XML-Schema) | partial OO mapping; IDL to implementation language (e.g. Java, C++) generator creates mobile classes without behavior. Only remote objects can inherit. |
| References | <ul style="list-style-type: none"> • synchronized bi-directional references? • multiple references? • lazy pointers | No explicit cyclic support (matter of schema design); Support for shared and lazy references | No |

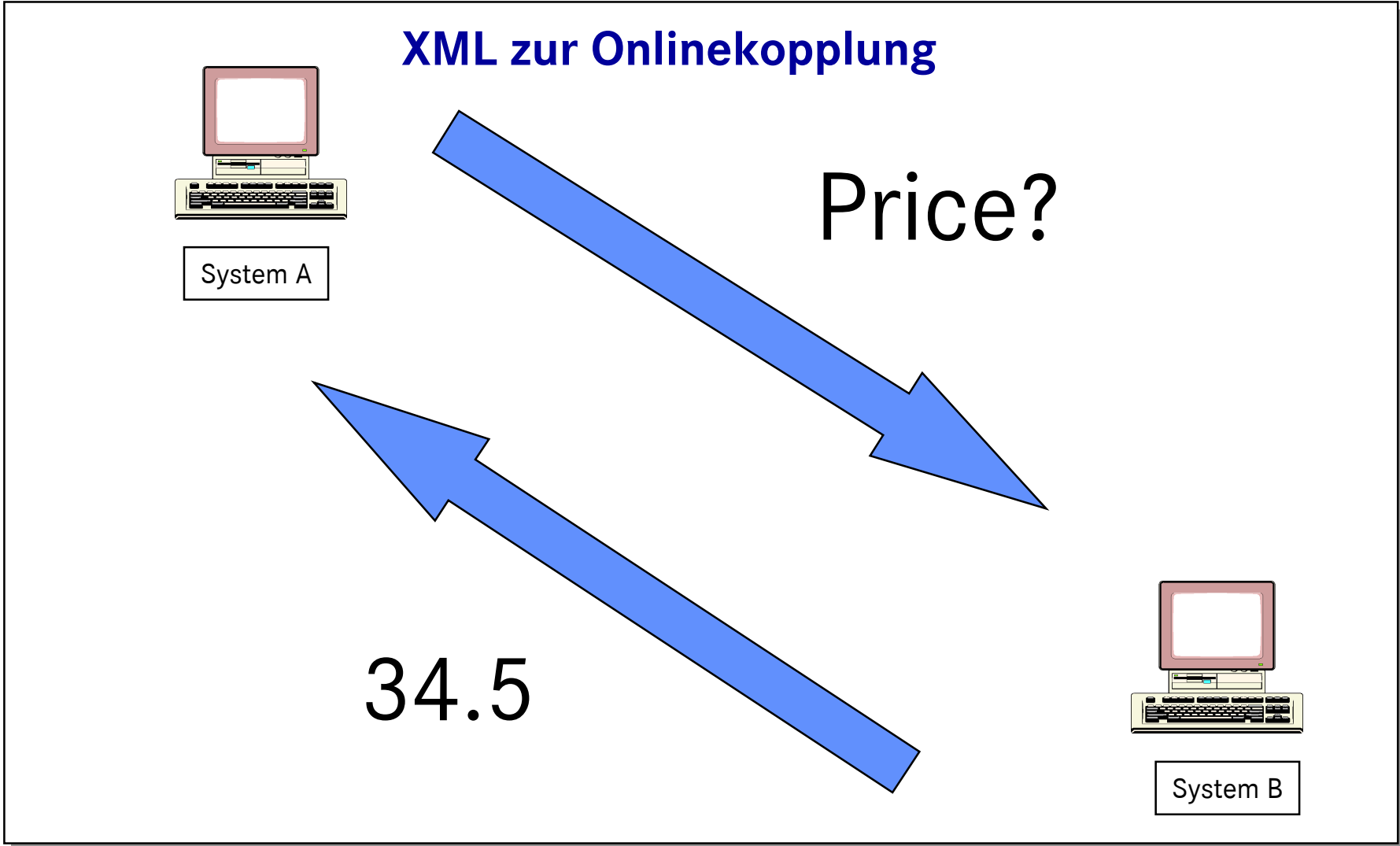
XML as a Distributed Application Protocol (cont'd)

| Capability | Description | XML | CORBA |
|---|--|---|--|
| Integration with application object model | Does the technology allow the application object model to be used as is? | No. XML parsers generally support DOM- or SAX-based object models. Use of application object models requires mapping functions or adapter layer | somewhat (the IDL to xxx generator creates structure classes that must then be used) |









XML zur Onlinekopplung

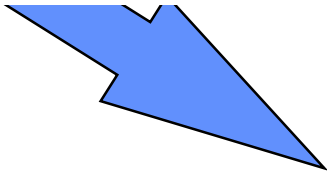
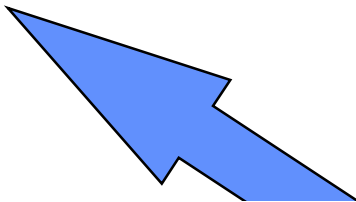


System A

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <symbol>DIS</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```



```

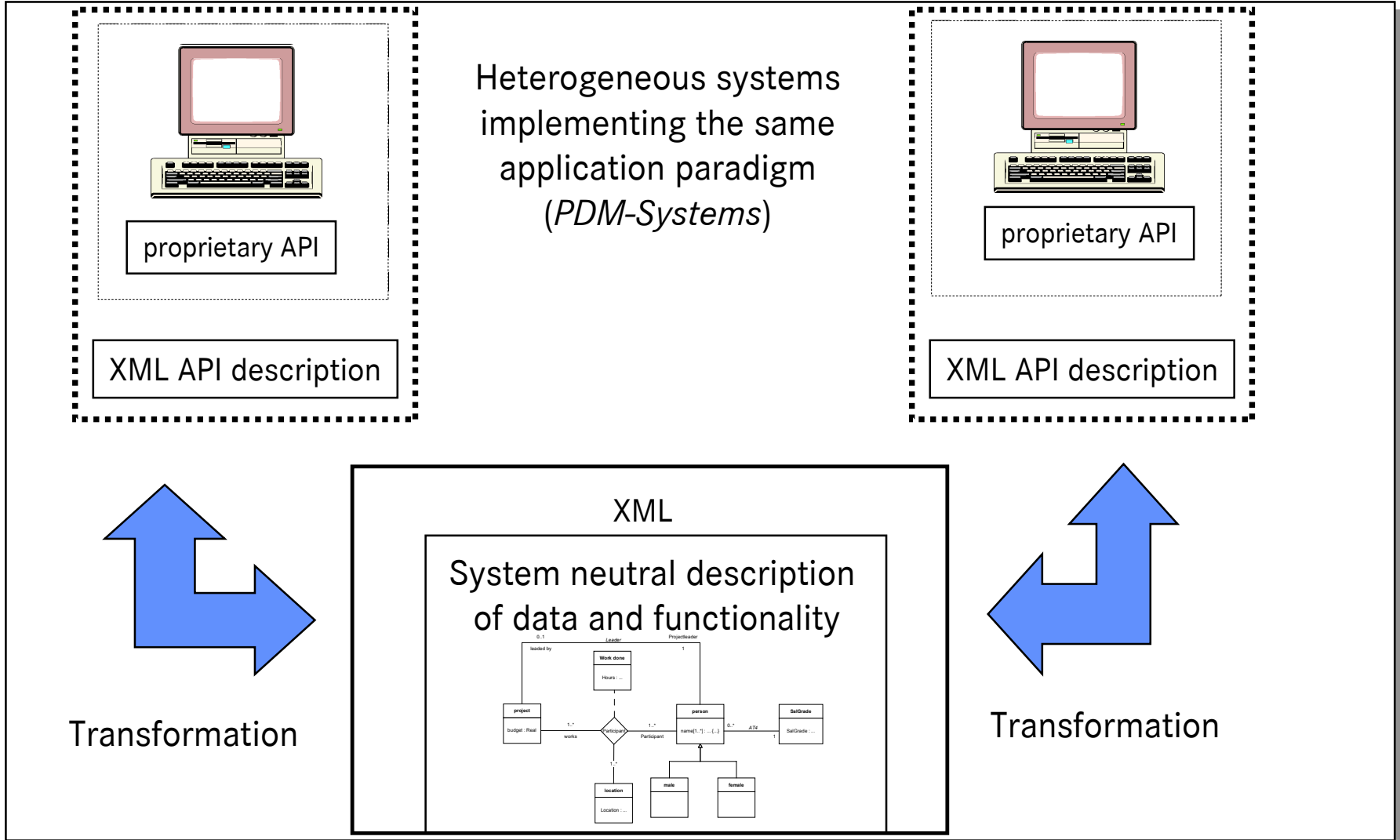
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePriceResponse xmlns:m="Some-URI">
      <Price>34.5</Price>
    </m:GetLastTradePriceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

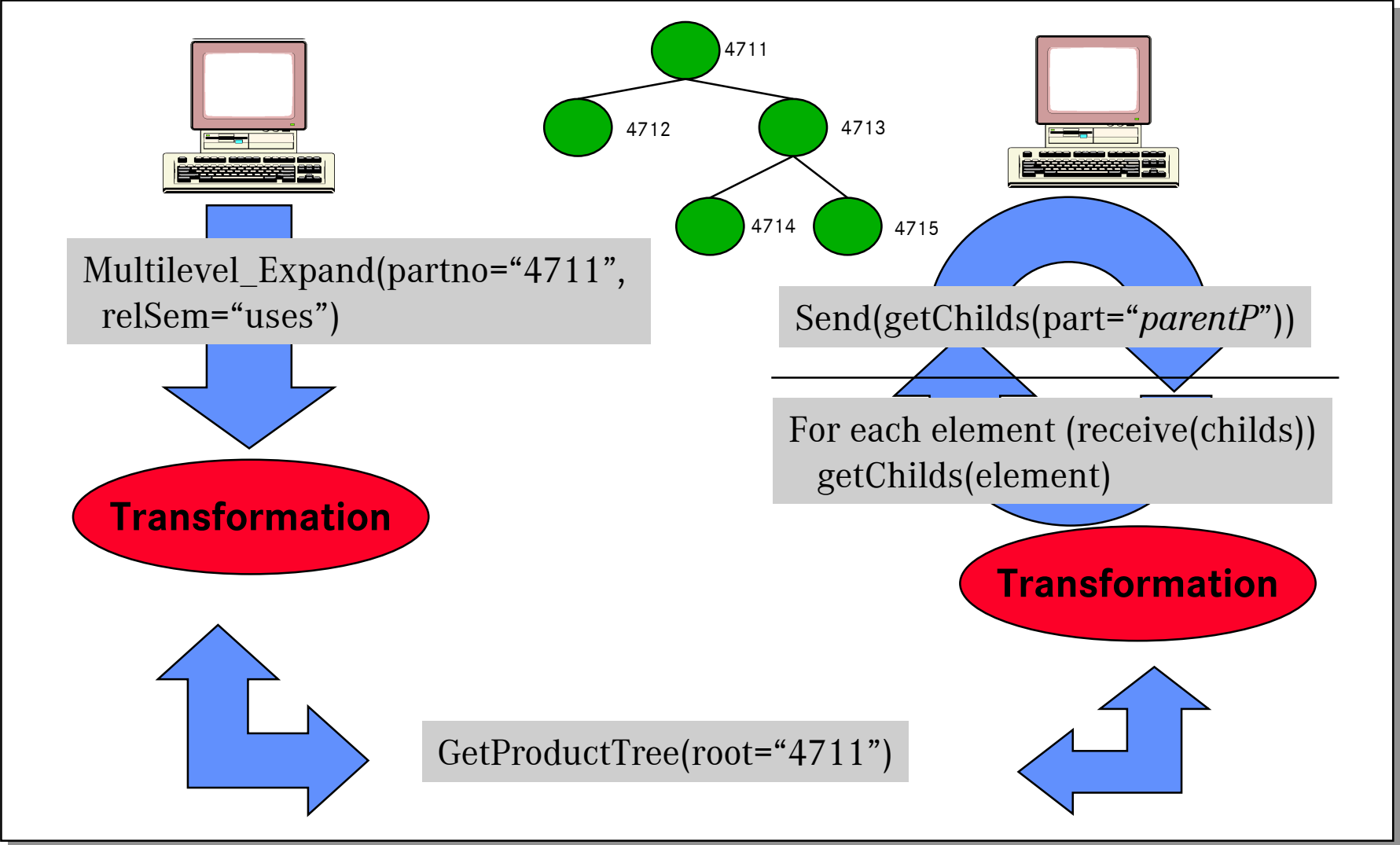
```

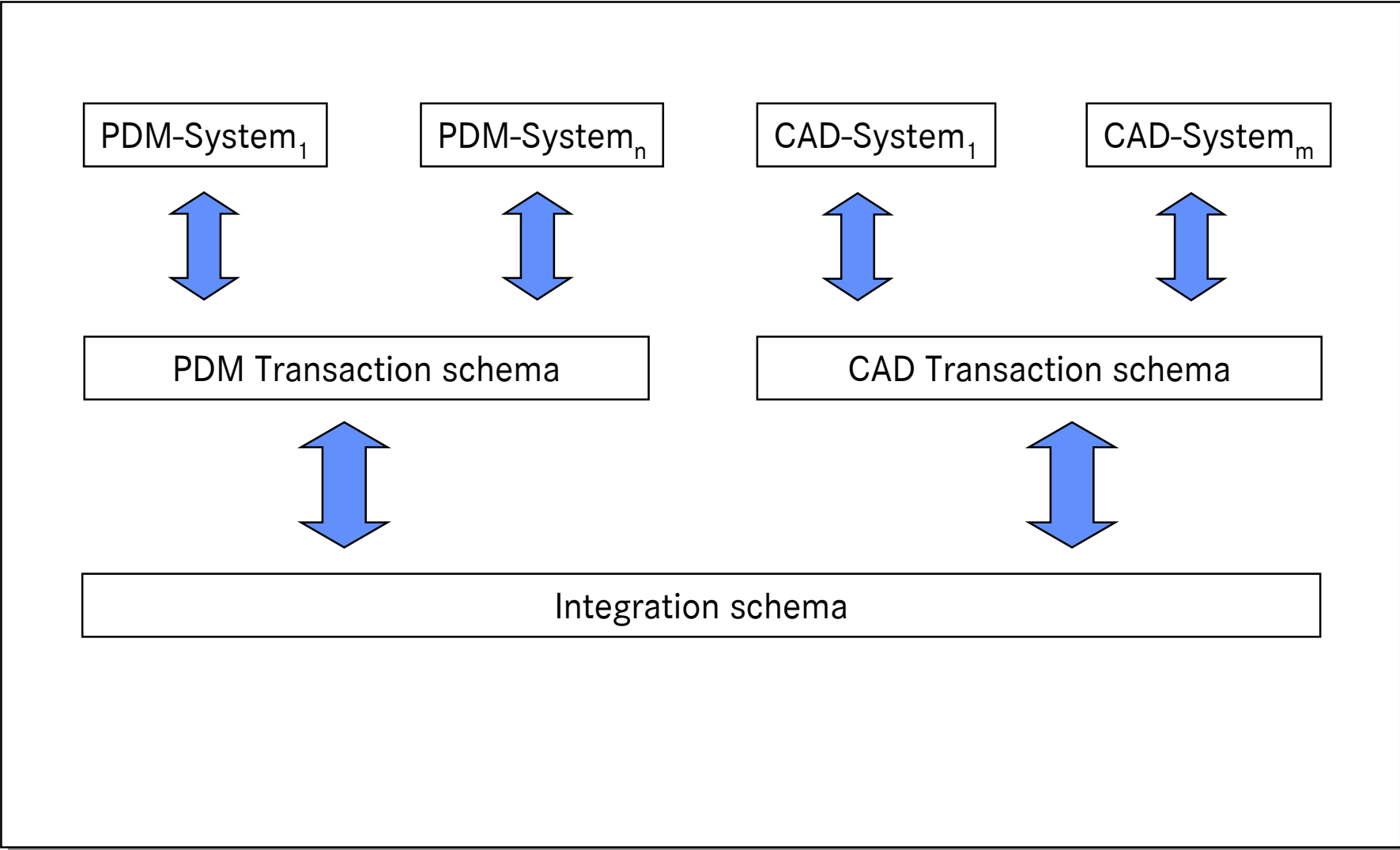


System B









References...

eXtensible Markup Language:

- www.w3.org/XML
- www.xml.com
- www.oasis-open.org/cover

CORBA:

- www.omg.org

STEP:

- www.nist.gov/sc4/www/stepdocs.htm
- www.pdm-if.org/pdm_schema

www.jeckle.de