



Die Beauftragte
der Bundesregierung
für Informationstechnik



Freie
Hansestadt
Bremen

Workshop

XÖV-Produktionszubehör und Produktionsumgebungen

Lars Hamann, TZI
Mirco Kuhlmann, TZI

Bremen, 26. April 2010

Haus der Bürgerschaft

8:30—12:30 Uhr

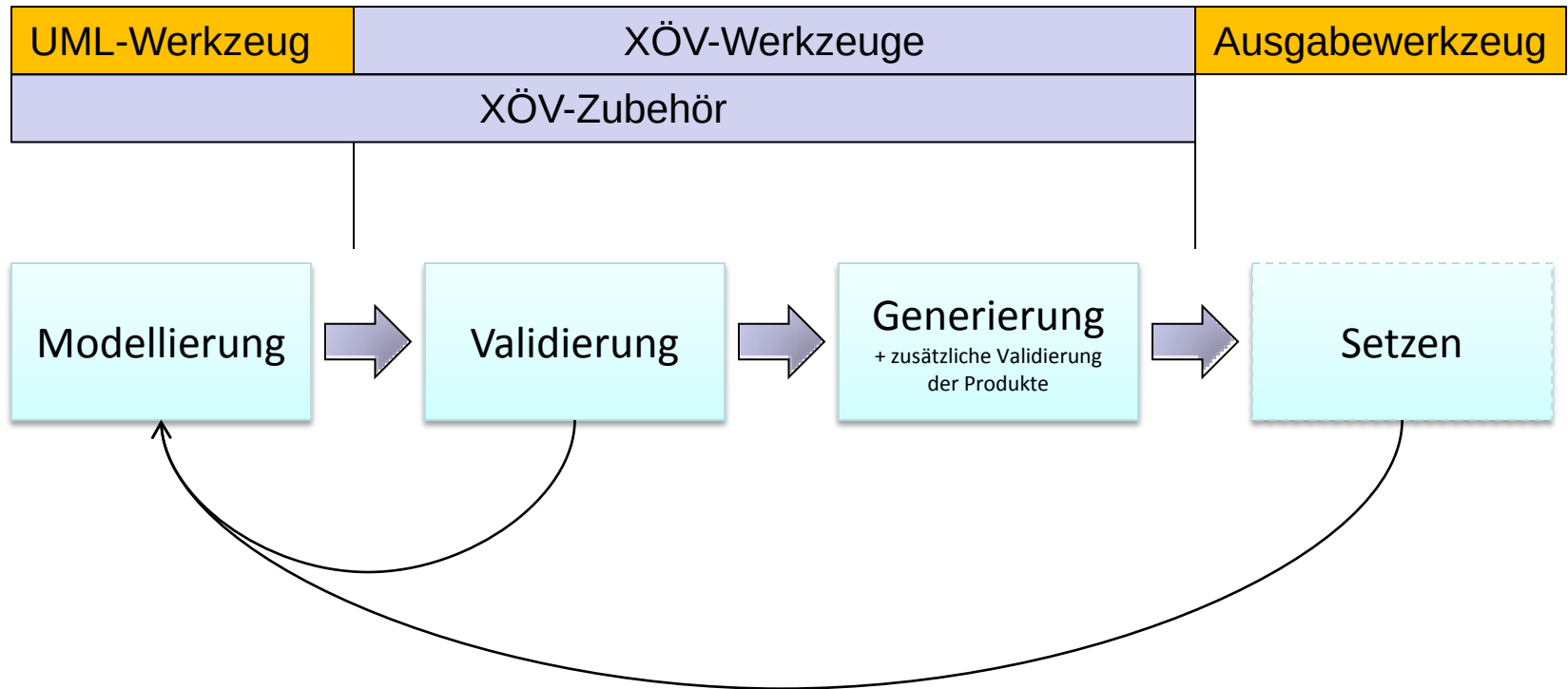
- Kurze Vorstellung der Teilnehmenden & Klärung der Erwartungen
- **Einführung in den XÖV-Produktionsprozess**
- **XÖV-Produktionszubehör und Produktionsumgebungen**
- Pause (9:45 bis 10:05 Uhr)
- **Arbeitsblock**
 - Einsatz des XÖV-Zubehörs in der Praxis*
- Pause (11:15 bis 11:35)
- **Diskussion**
 - Ausgestaltung möglicher Produktionsumgebungen*
- Abschlussrunde

Einführung in den XÖV-Produktionsprozess

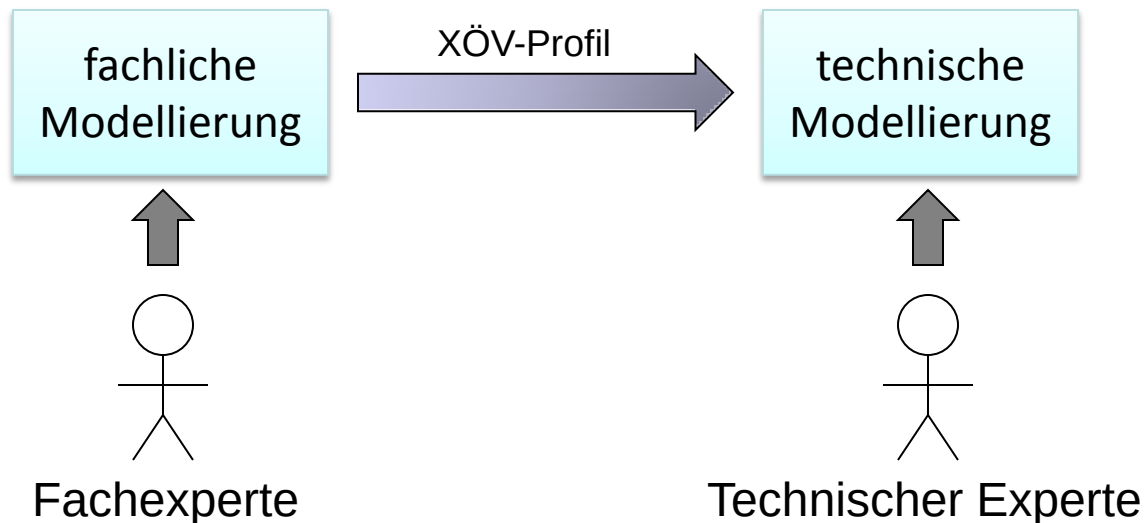
- Grobe Übersicht über den XÖV-Produktionsprozess
- Was sind die Ziele der einzelnen Schritte?
- Wer ist beteiligt?
- Was passiert im Hintergrund?
- Welche Artefakte werden verwendet?
 - Welche davon legt der XÖV-Prozess fest?
 - Wo kann man abweichen?
 - Was muss man bei Abweichungen beachten?



Der XÖV-Produktionsprozess

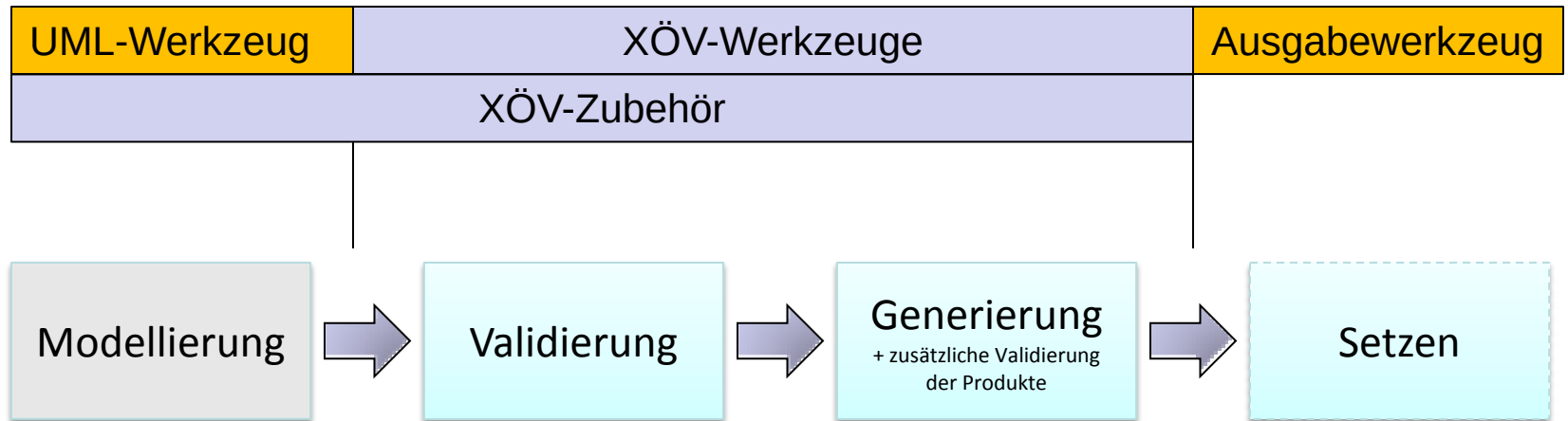


- Überführung der fachlichen Inhalte in ein UML-Modell
 - Grafische Darstellung der Struktur
 - Ausblendung aktuell nicht relevanter Informationen
- Ergänzung um technische Details (XÖV-Profil)
 - Benötigt keine Fachexperten
- Ergebnis der Modellierung ist ein generisches Format (XMI)





Der XÖV-Produktionsprozess



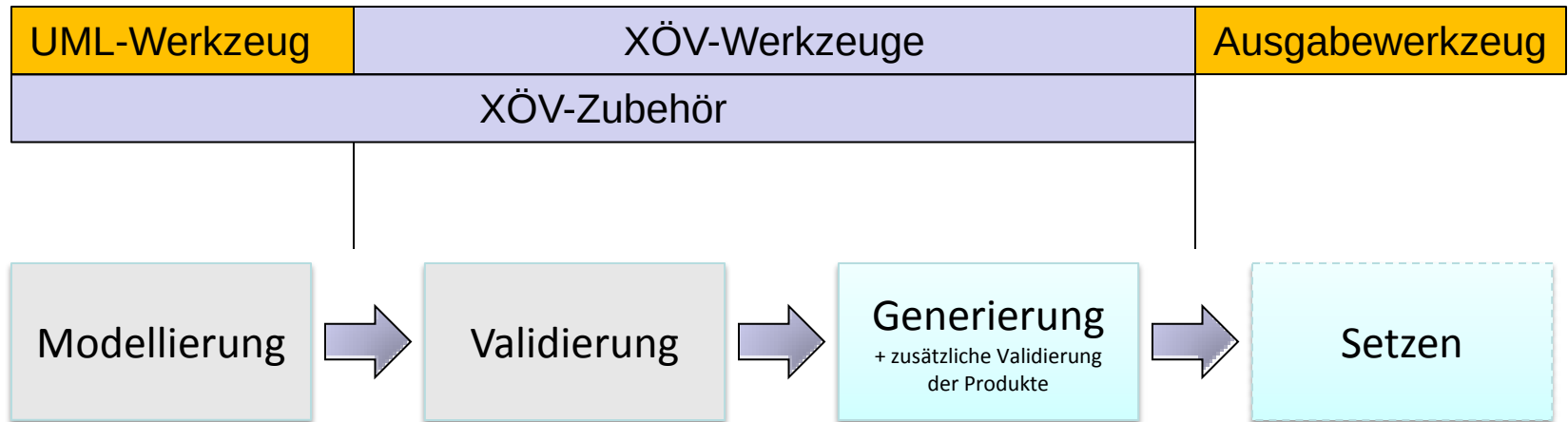
- Für die korrekte Transformation gibt es Anforderungen
 - „Well-Formedness-Rules“
- Weitere Regeln durch das XÖV-Handbuch gegeben
 - Namens- und Entwurfsregeln
- Einhaltung kann ggf. automatisch geprüft werden (OCL):

```
context Class inv SingleInheritance:  
self.hlpIsPartOfProfiledPackage() and  
self.hlpIsPartOfXModel() implies  
self.general->size <= 1
```

- Nur für ein valides Modell lohnen weitere Schritte
- Benutzer erhält Rückmeldung über fehlerhafte Modellelemente



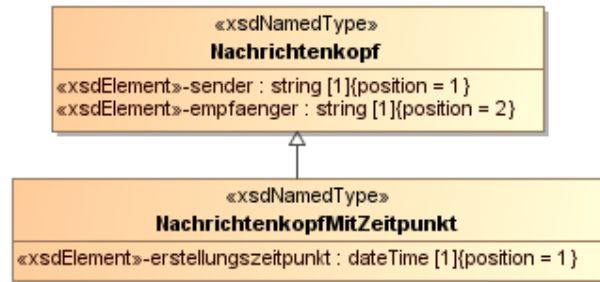
Der XÖV-Produktionsprozess



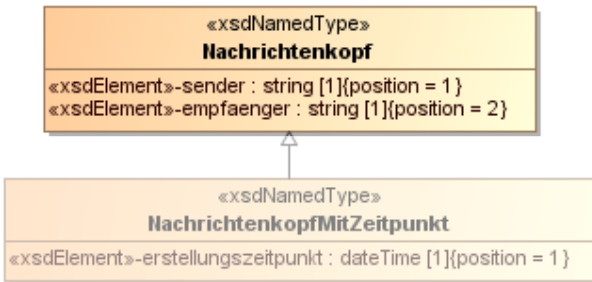
- Textschablonen (Templates) für Elemente des Modells
- Abfrage des Modells mit OCL:

```
<xs:complexType
#if($ocl.eval("n.extension_xsdNamedType.isDefined"))
  name='$ocl.eval("n.name")' #end
#if($ocl.eval("n.isAbstract"))
  abstract='true' #end>
#if($ocl.eval("n.extension_xsdNamedType.isDefined"))
$template.call("annotation.vm", "n")
#end
```

- Template enthält drei verschiedene Sprachen:
 - XML-Schemadefinition
 - Velocity-Anweisungen
 - OCL



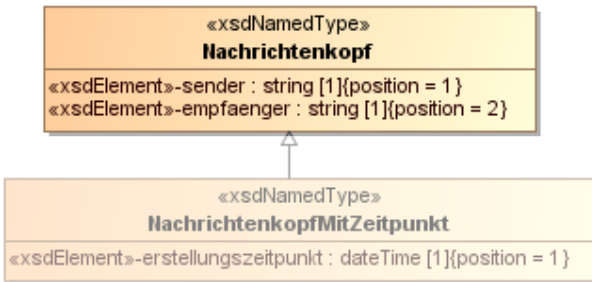
- XML-Schema
- DocBook-Fragmente
- WSDL
- ...



- XML-Schema
- DocBook-Fragmente
- WSDL
- ...

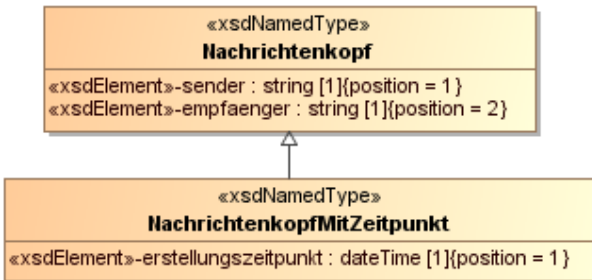
```

<xs:complexType name="Nachrichtenkopf">
  <xs:annotation>
    <xs:documentation>Nachrichtenkopf mit Informationen...</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="sender" type="xs:string">
      <xs:annotation>
        <xs:documentation>Der Sender der Nachricht.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="empfaenger" type="xs:string">
      <xs:annotation>
        <xs:documentation>Der Empfänger der Nachricht.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
    
```



- XML-Schema
- DocBook-Fragmente
- WSDL
- ...

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- DocBook content generated by XGenerator. -->
<section id='xhz-nachrichtenkopf'>
  <title>Nachrichtenkopf</title>
  <para>Nachrichtenkopf mit Informationen zum Sender und Empfänger einer Nachricht.</para>
  ...
</section>
<title>Allgemeine Eigenschaften</title>
<informaltable id="tabellen.allgemein.xhz-nachrichtenkopf">
  <tgroup cols="2">
    ...
    <tbody>
      <row>
        <entry>Name im XML-Schema</entry>
        <entry>Nachrichtenkopf</entry>
      </row>
      <row>
```



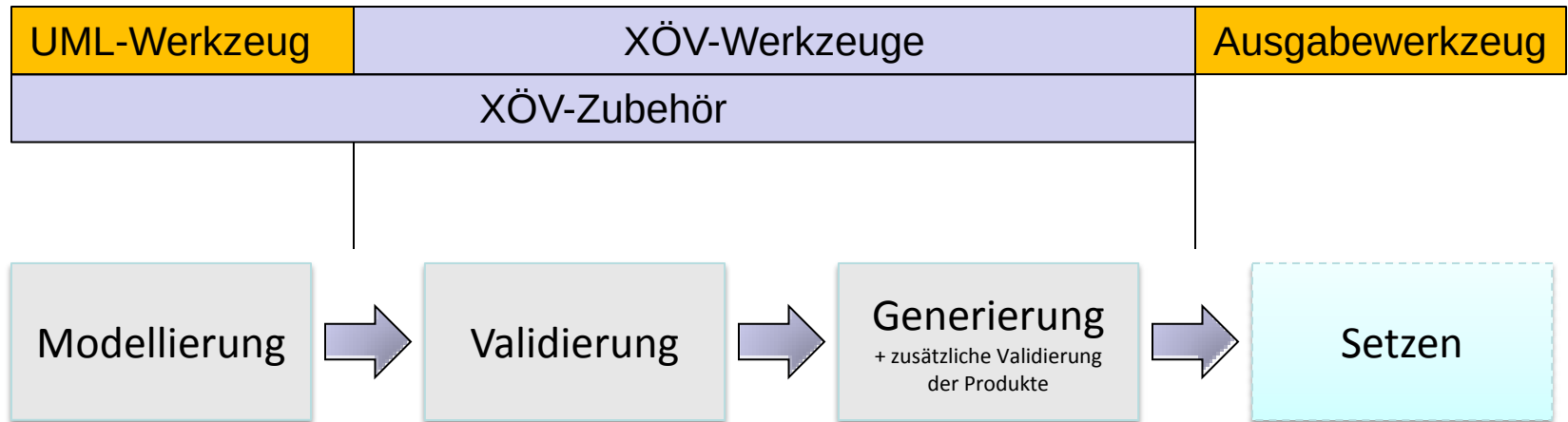
- XML-Schema
- DocBook-Fragmente
- WSDL
- ...

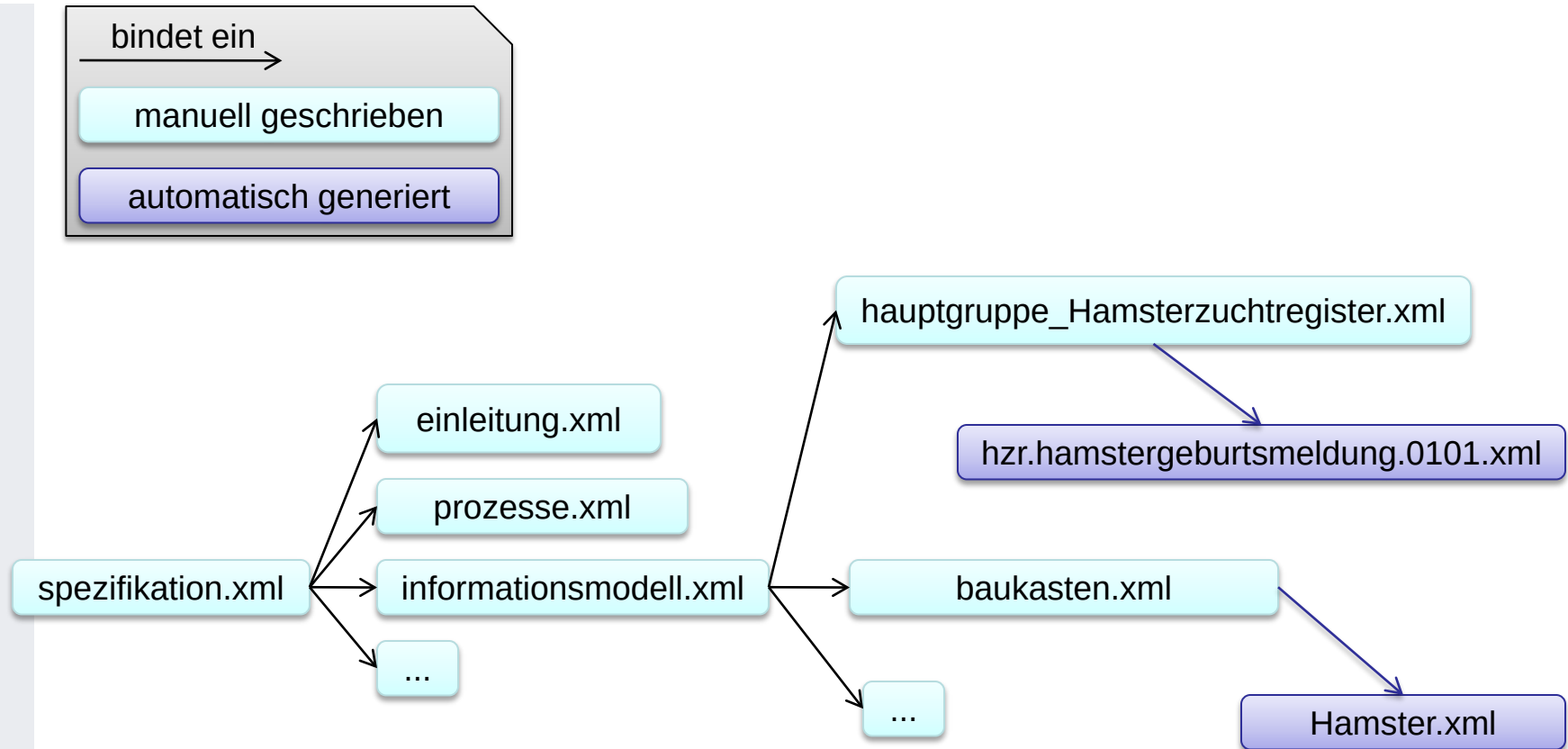
```

<xs:complexType name="NachrichtenkopfMitZeitpunkt">
  <xs:annotation>
    <xs:documentation>Nachrichtenkopf mit Erweiterung...</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="xhz:Nachrichtenkopf">
      <xs:sequence>
        <xs:element name="erstellungszeitpunkt" type="xs:dateTime">
          <xs:annotation>
            <xs:documentation>Der Zeitpunkt, an dem die Nachricht erstellt wurde.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
    
```

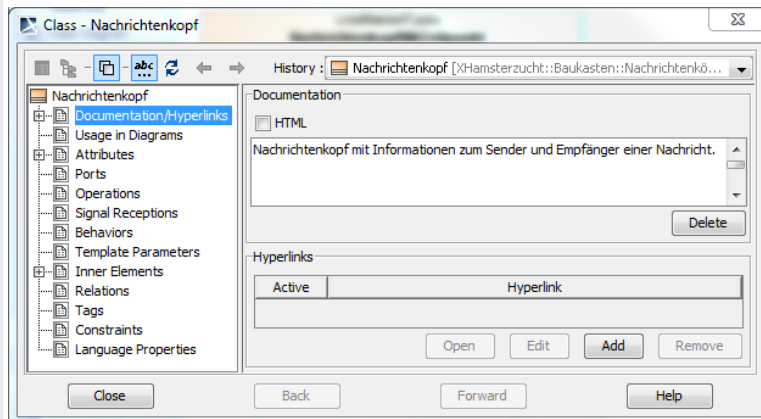
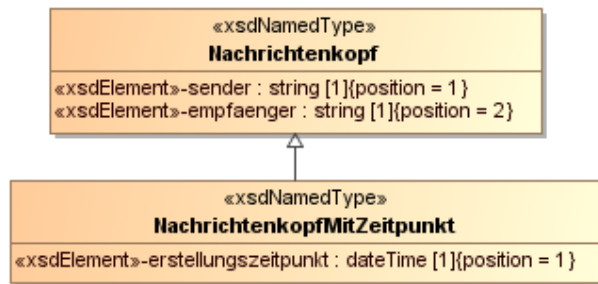


Der XÖV-Produktionsprozess





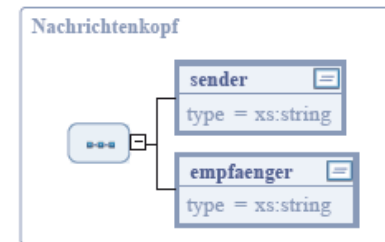
- Werkzeug liest DocBook und erzeugt Ausgabe
- OSCI:
 - Werkzeug: Adobe Framemaker
 - Ausgabe: PDF
 - Vorteil: Endredaktion möglich
 - Nachteil: Preis
- XHamsterzucht
 - Werkzeug: Apache FOP
 - Ausgabe: PDF
 - Vorteil: Open Source
 - Nachteil: Keine Endredaktion
- Ausgabe für PDF im XÖV-Styleguide festgelegt



3.4.1.13. Nachrichtenkopf

Nachrichtenkopf mit Informationen zum Sender und Empfänger einer Nachricht.

Abbildung 3.21. xhz:Nachrichtenkopf



3.4.1.13.1. Allgemeine Eigenschaften

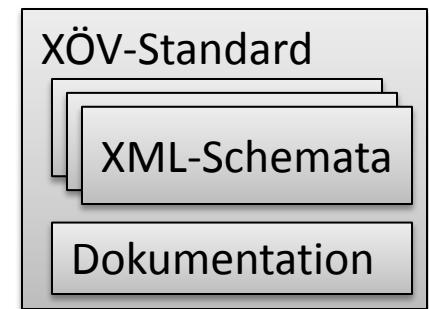
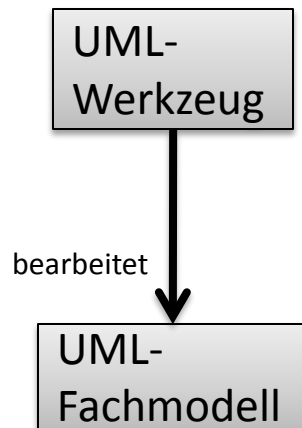
Eigenschaft	Wert
Name im XML-Schema	Nachrichtenkopf
Implementierungshinweis	-
Inhaltsmodell	komplexer Typ
Elementmodell	Sequenz
Verwendet in	- xhz:hamsterzuchtregister.hamstergeburtsmeldung.0101 (<i>globales Element</i>), siehe Seite 10 - xhz:hamsterzuchtregister.hamstertodesmeldung.0103 (<i>globales Element</i>), siehe Seite 15 - xhz:hamsterzuchtregister.zuechterabmeldung.0104 (<i>globales Element</i>), siehe Seite 16 - xhz:hamsterzuchtregister.zuechteranmeldung.0102 (<i>globales Element</i>), siehe Seite 13 - xhz:wettbewerbsmeldungen.einladung.0203 (<i>globales Element</i>), siehe Seite 25 - xhz:wettbewerbsmeldungen.ergebnismeldung.0202 (<i>globales Element</i>), siehe Seite 24
Zur Ableitung genutzt von	- xhz:NachrichtenkopfMitZeitpunkt, siehe Seite 54

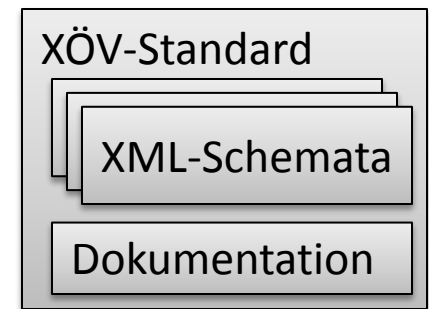
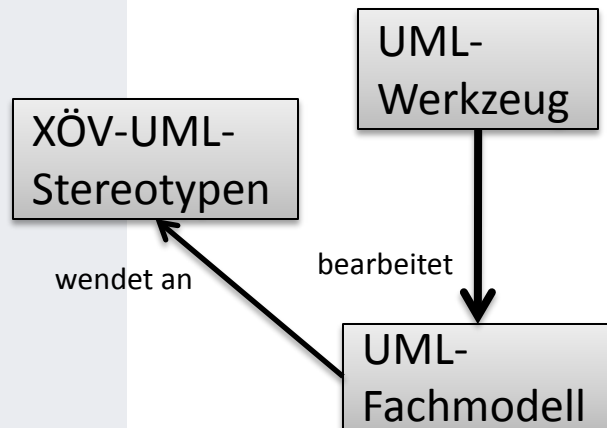
XÖV-Produktionszubehör und Produktionsumgebungen

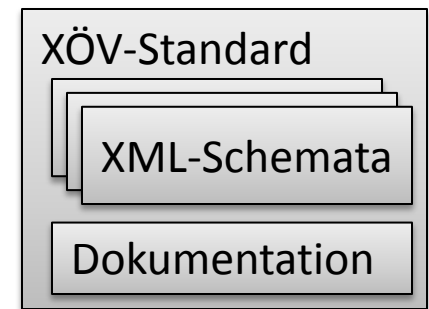
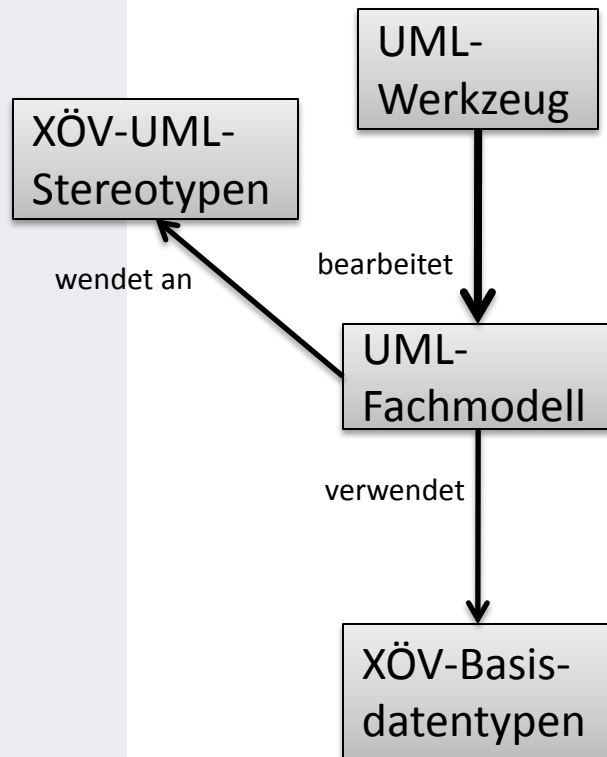
UML-
Fachmodell

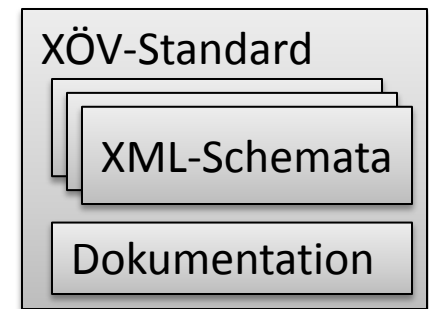
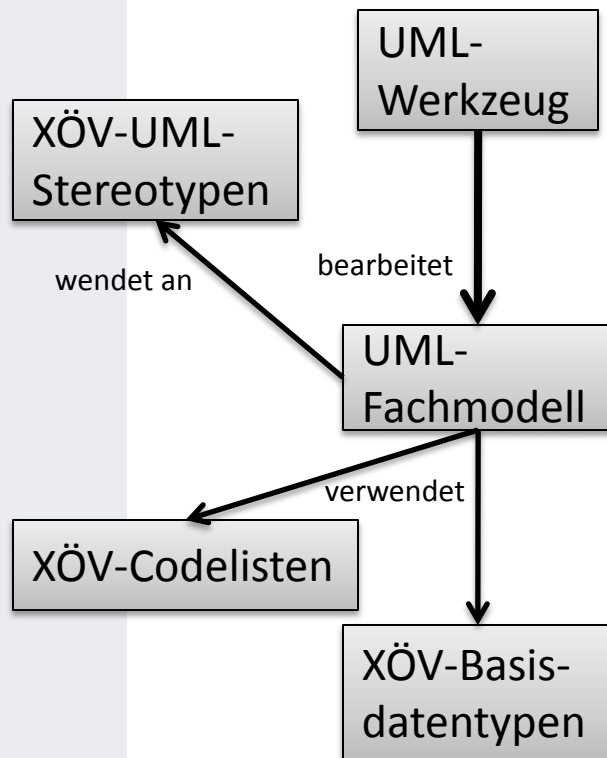
UML-
Fachmodell

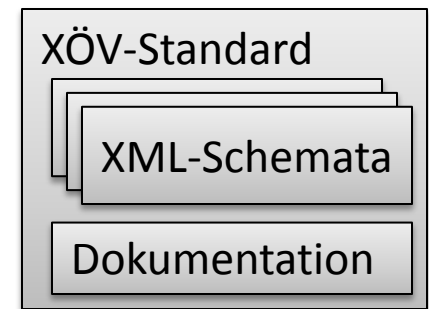
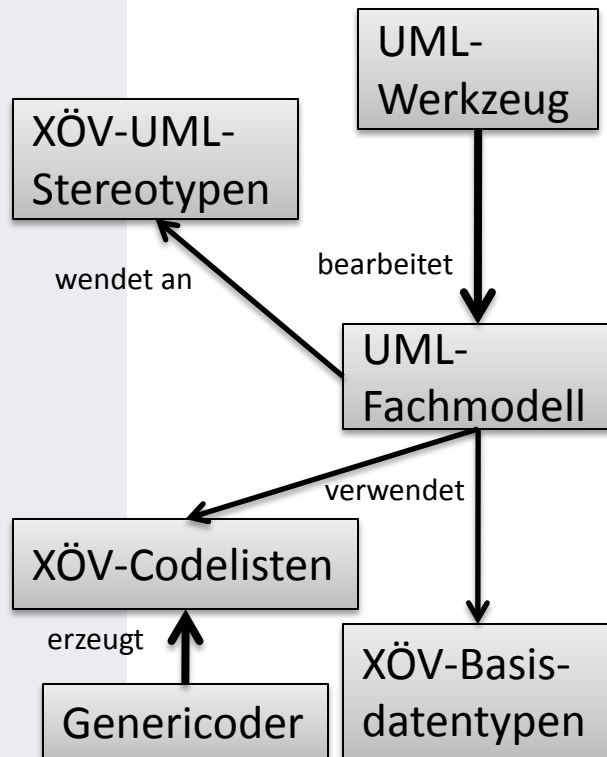
XÖV-Standard
XML-Schemata
Dokumentation

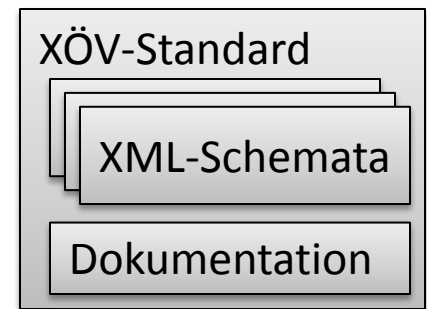
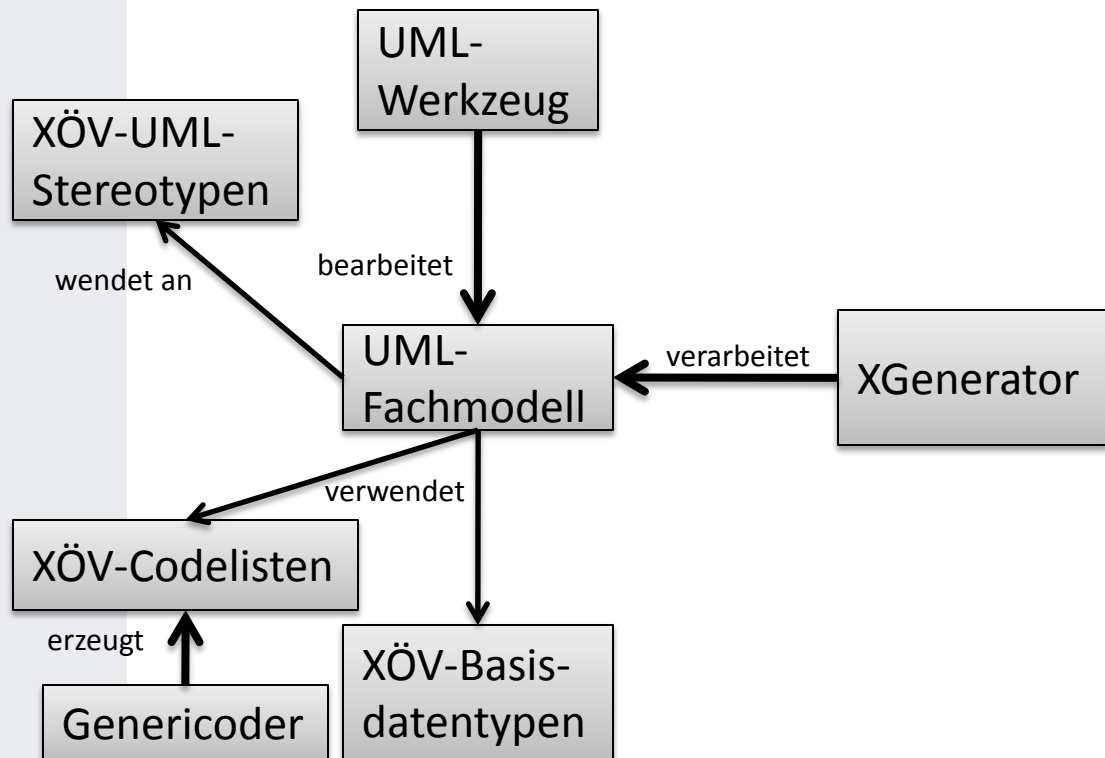


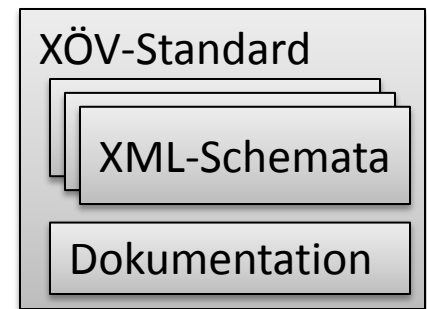
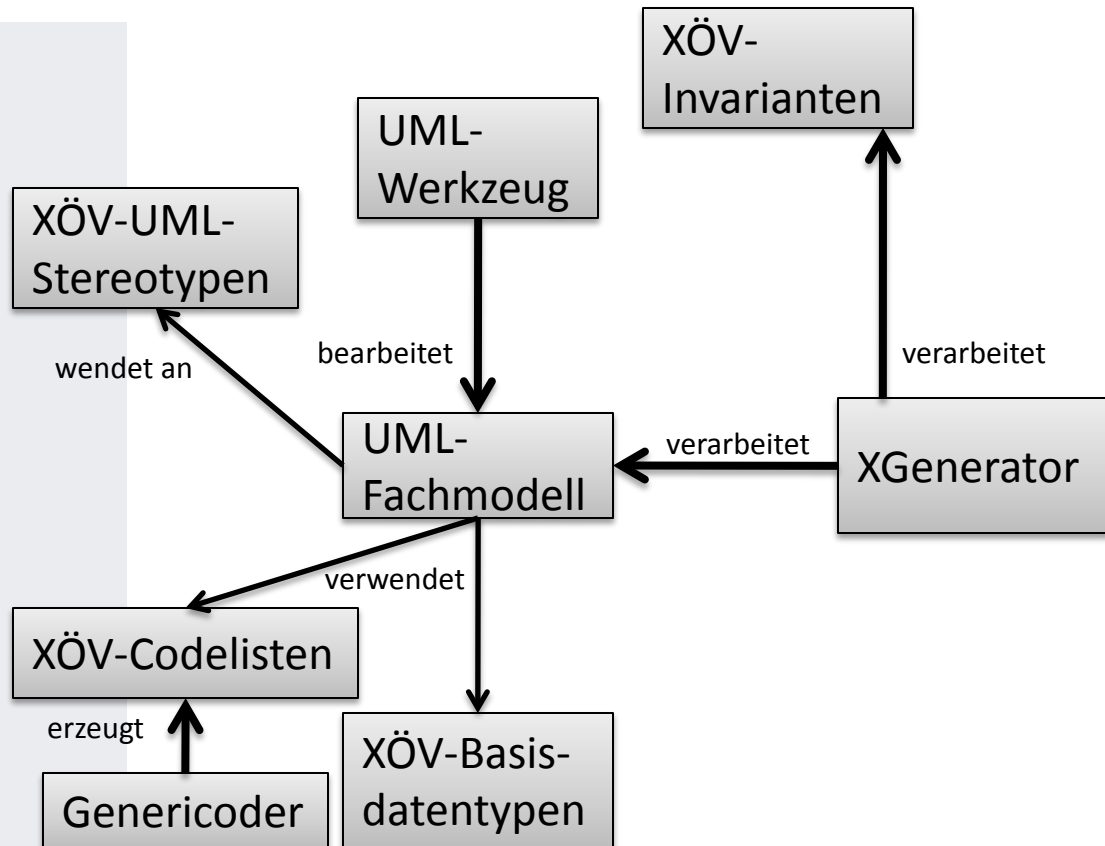


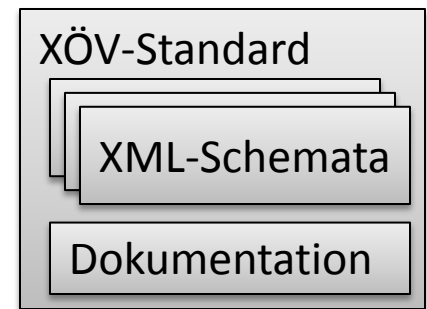
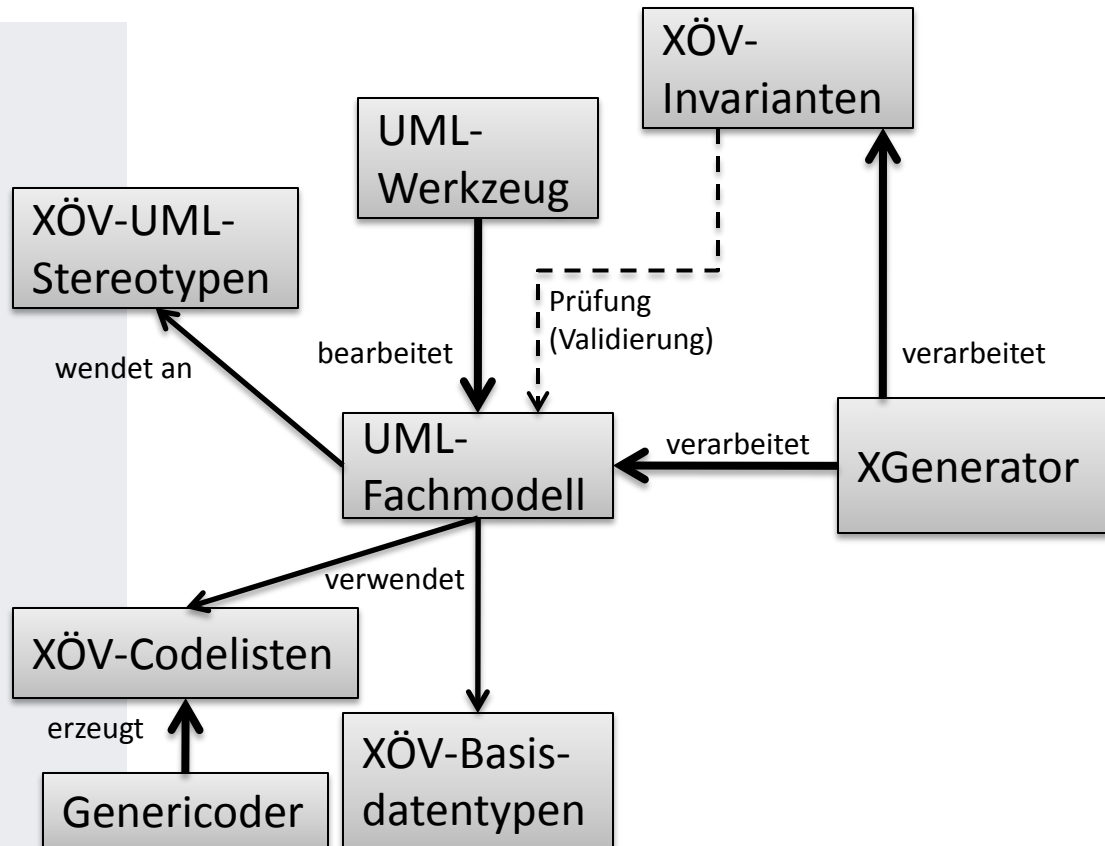


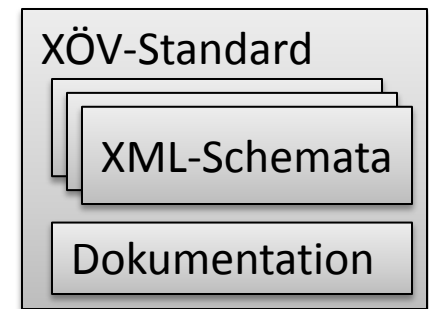
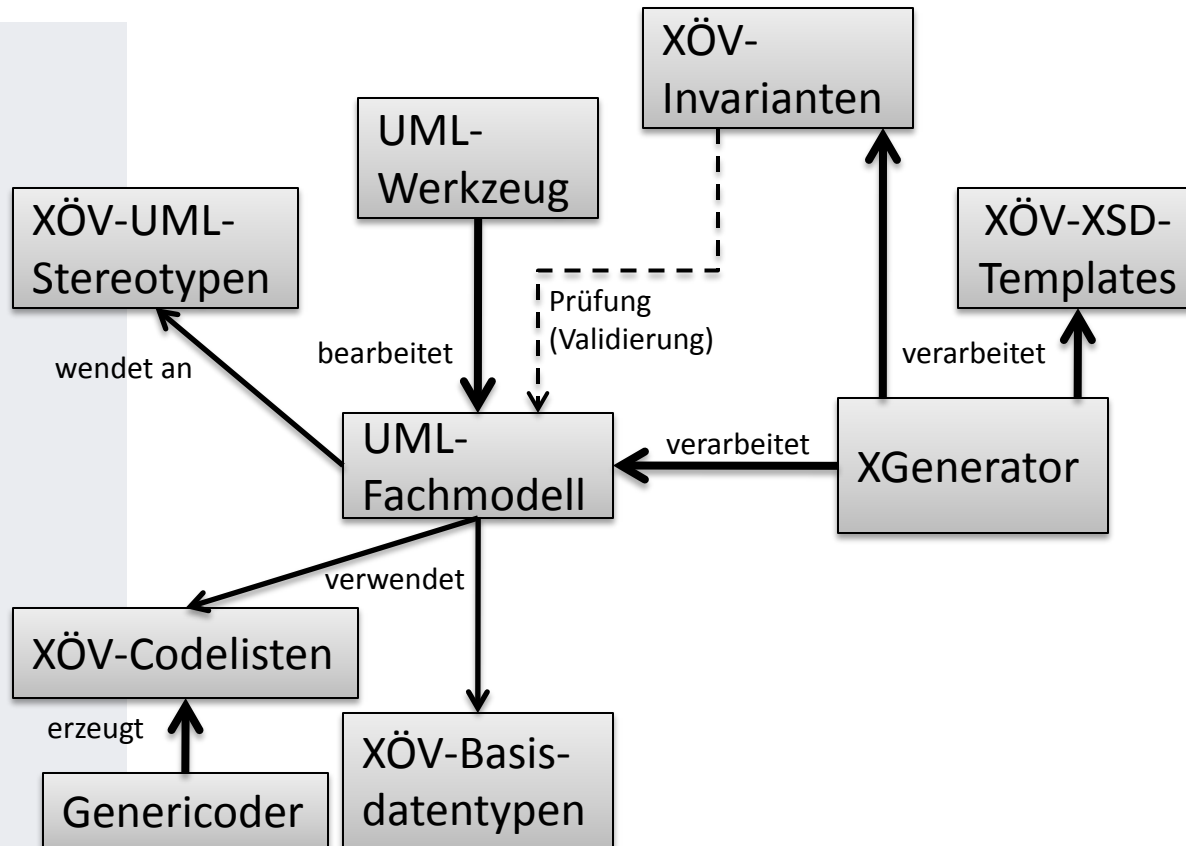




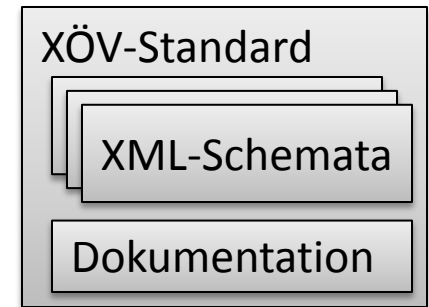
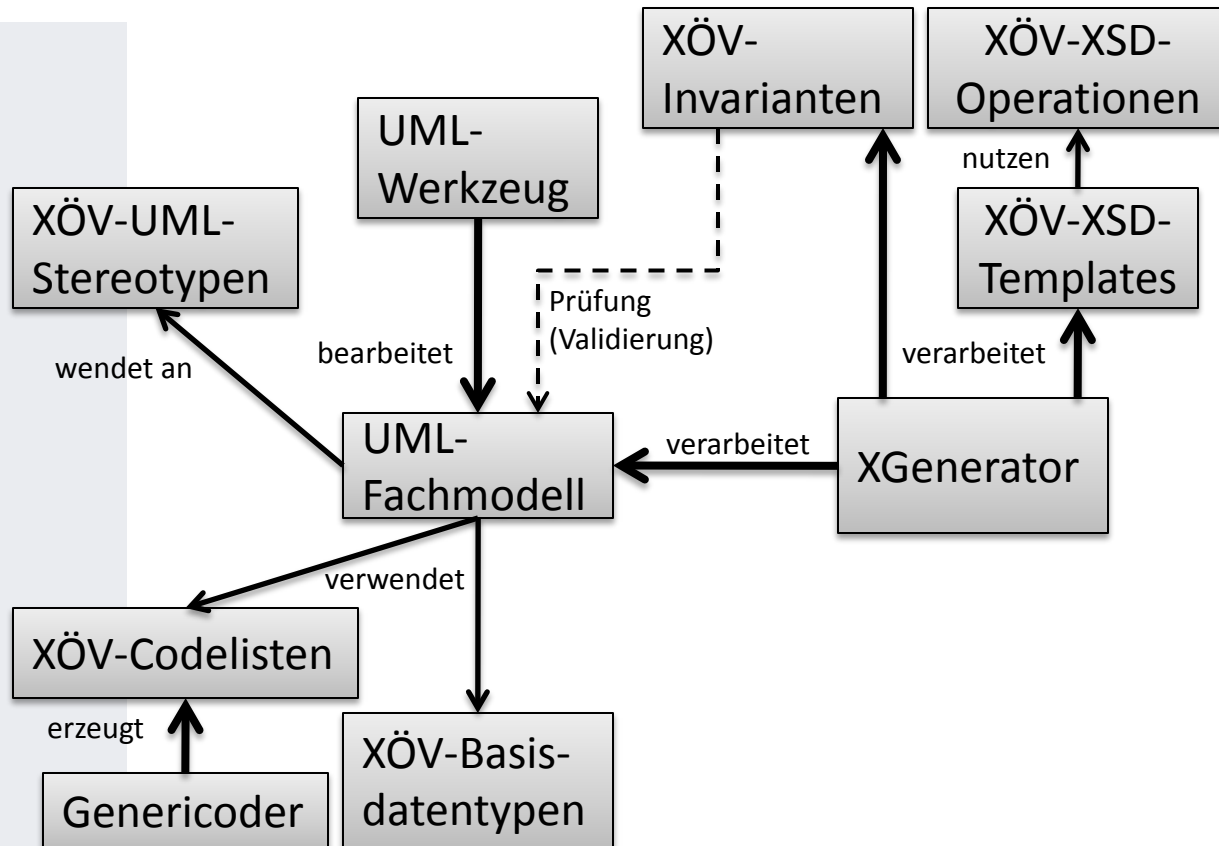




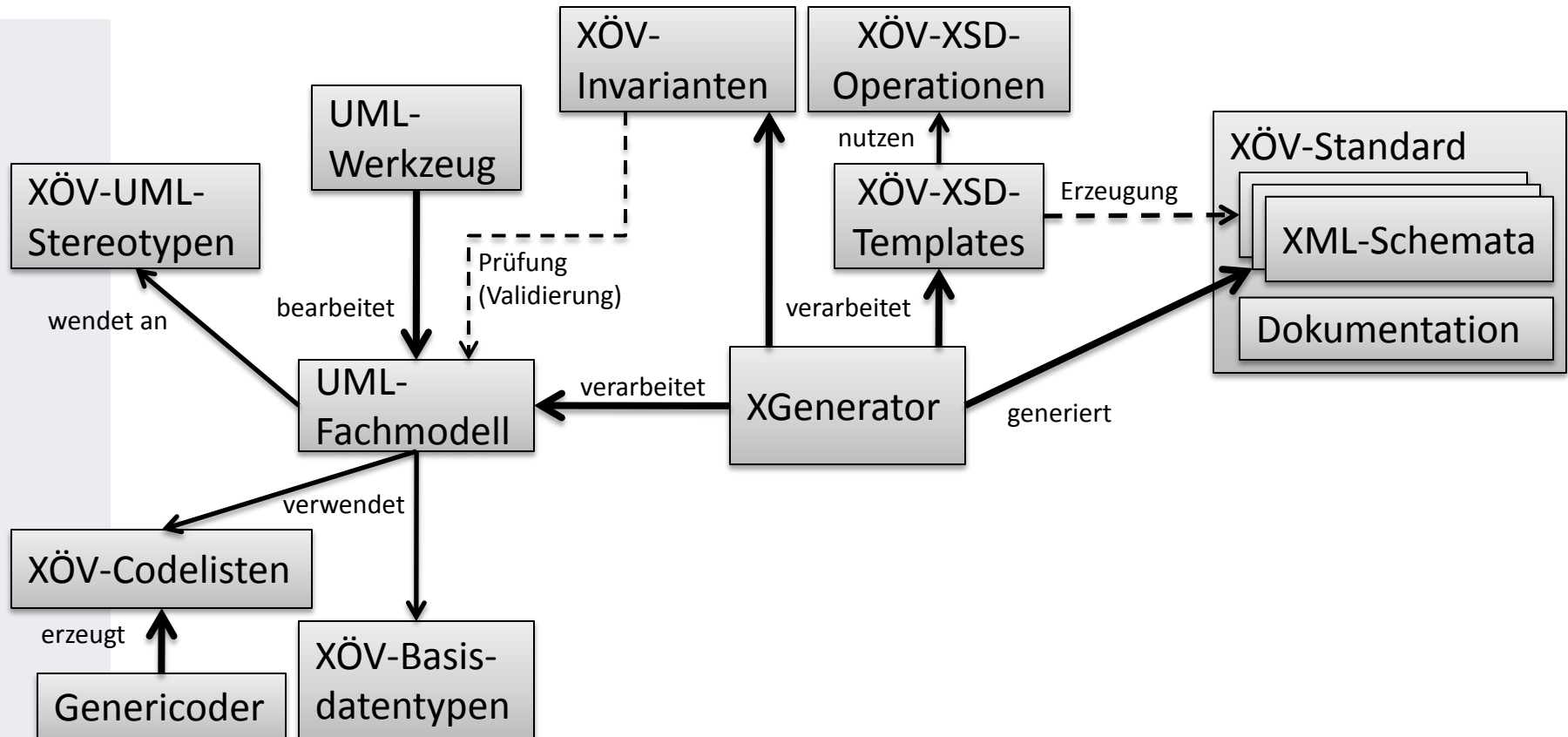




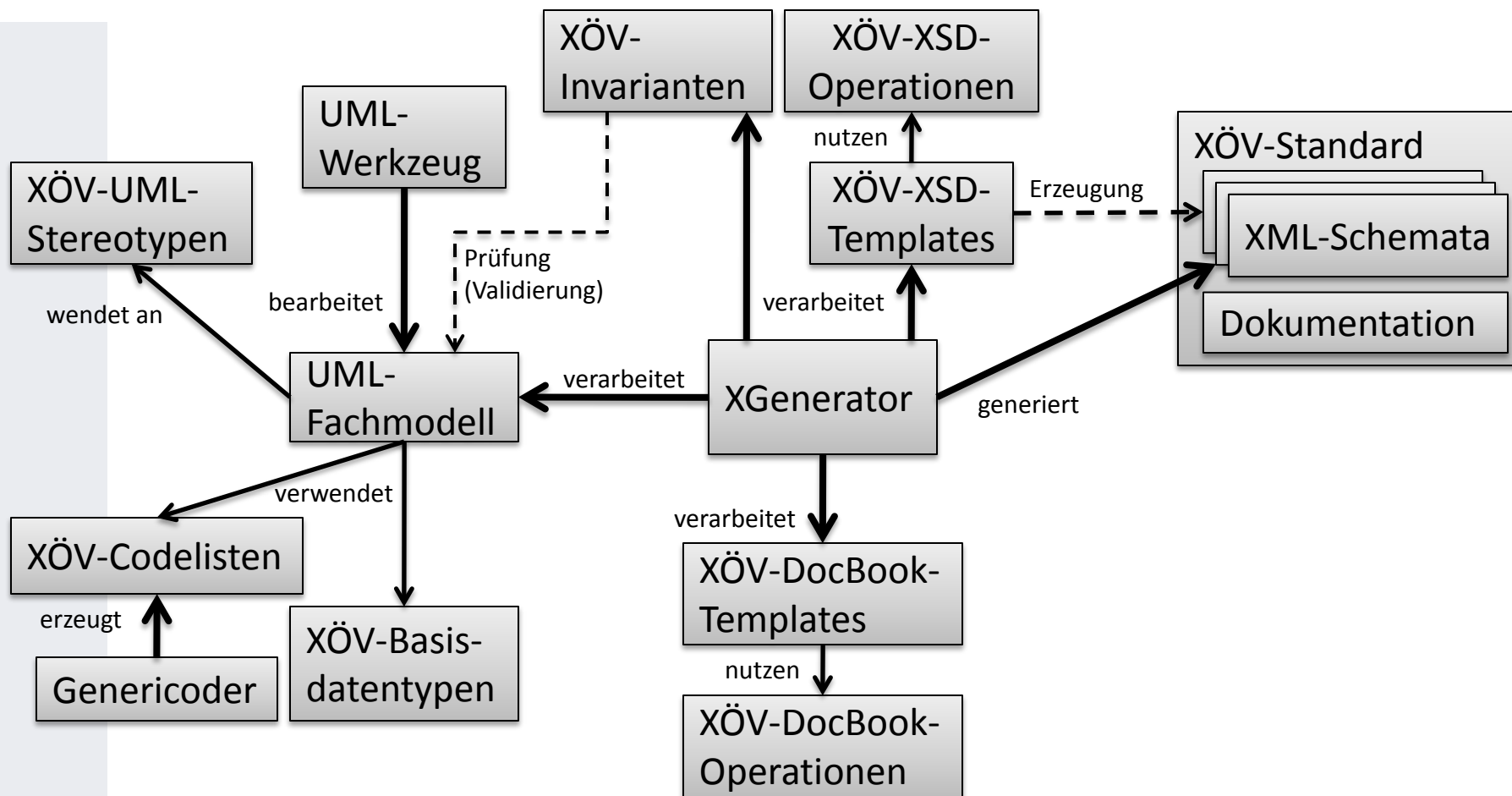
XÖV-Übersicht



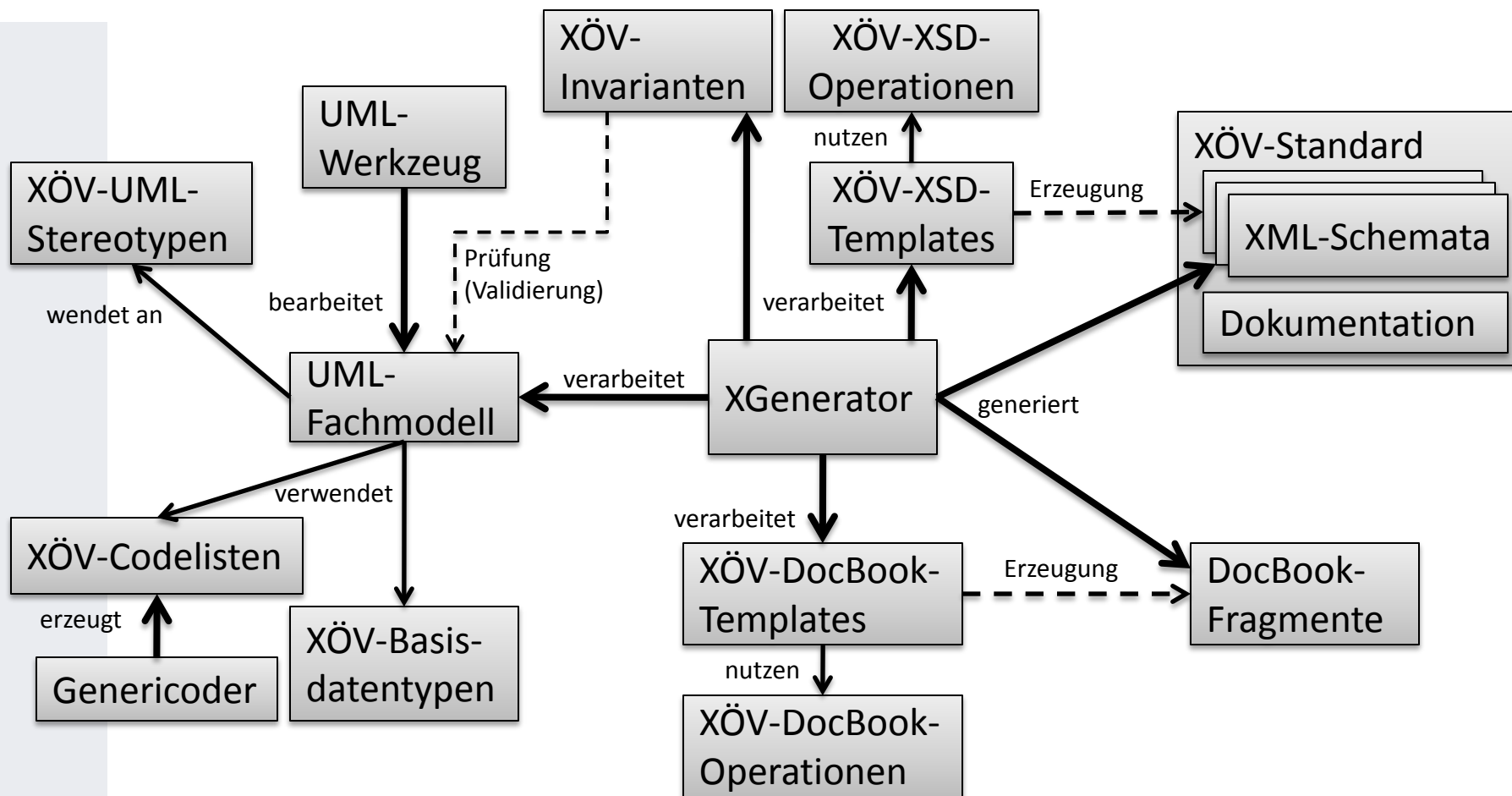
XÖV-Übersicht



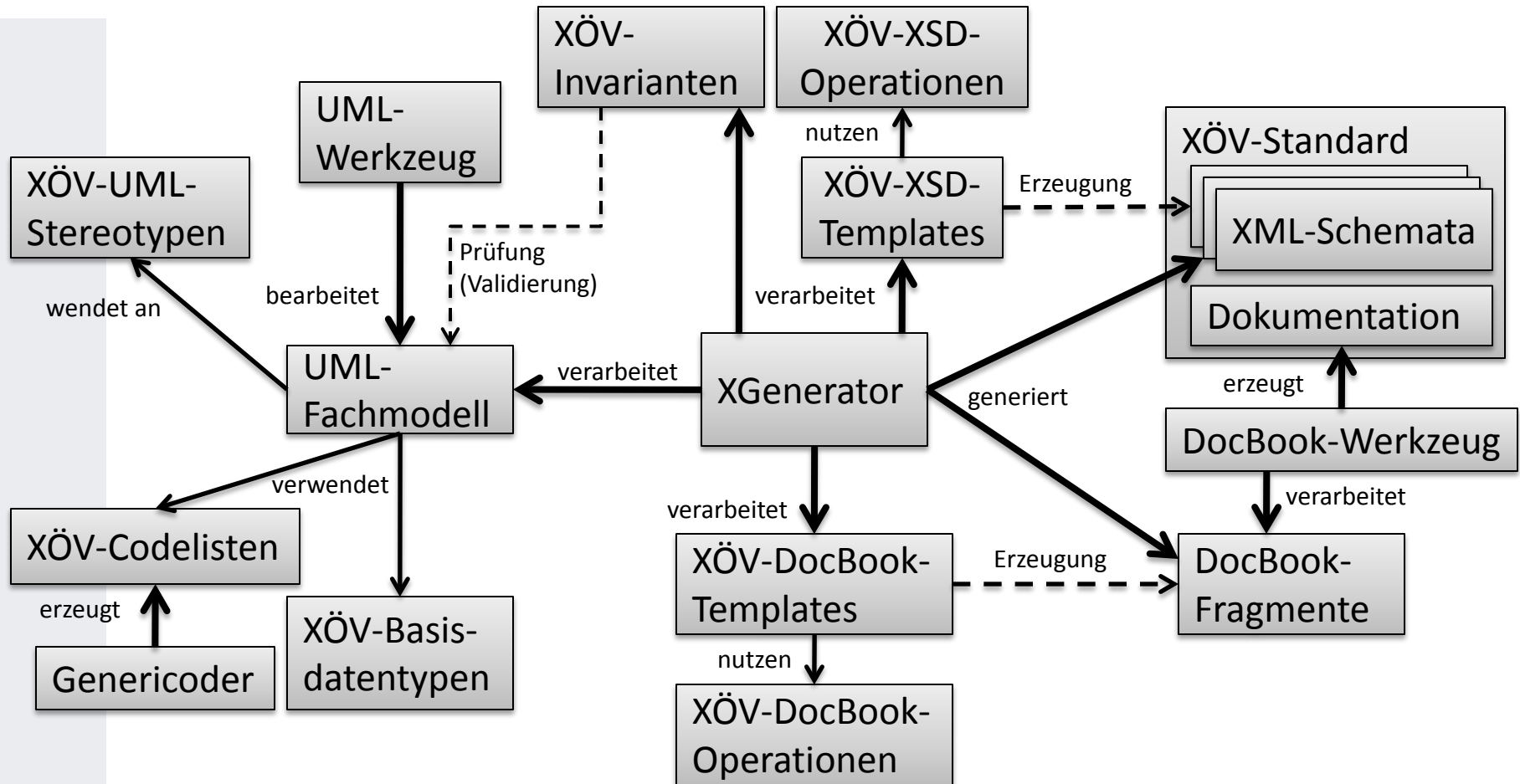
XÖV-Übersicht



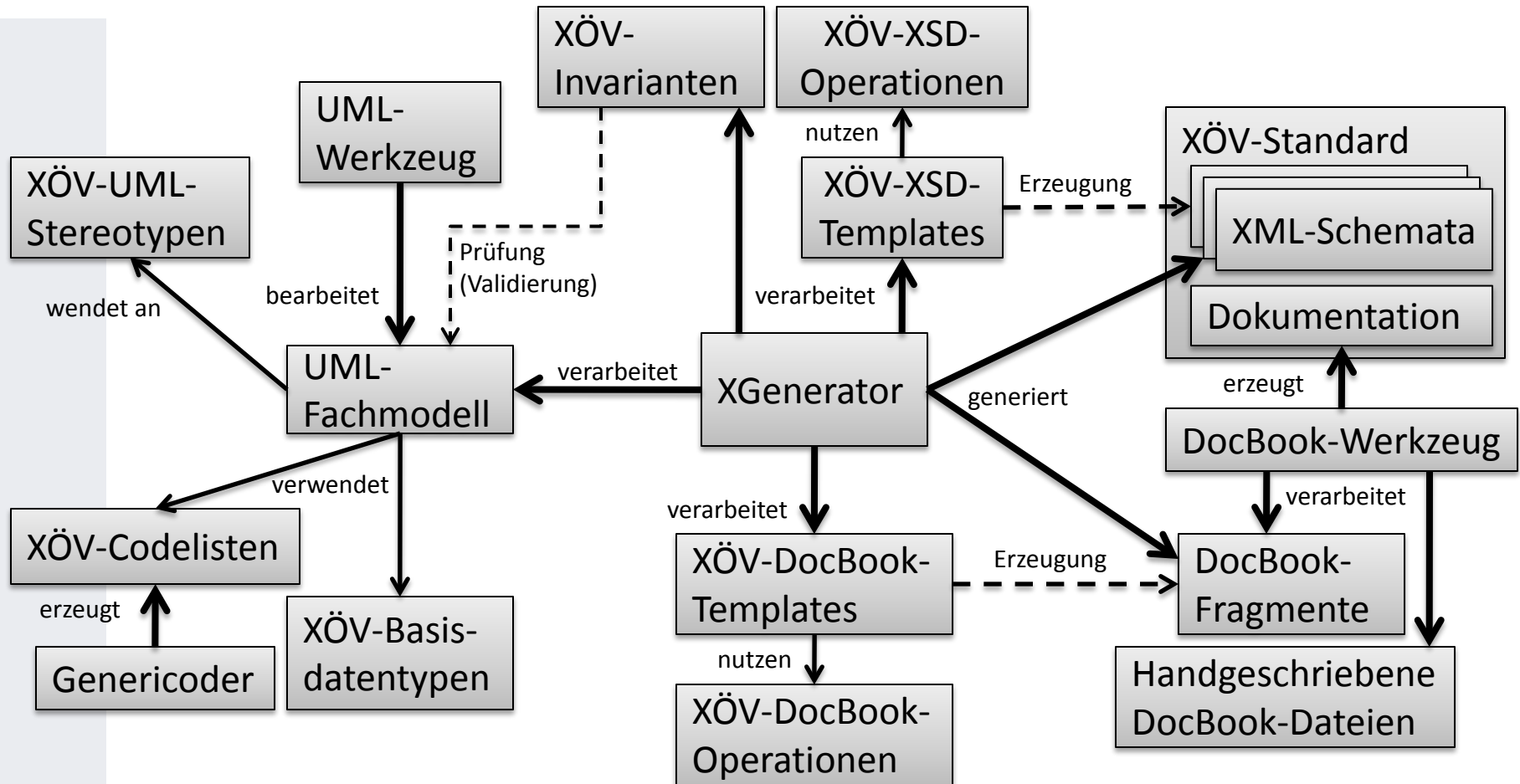
XÖV-Übersicht



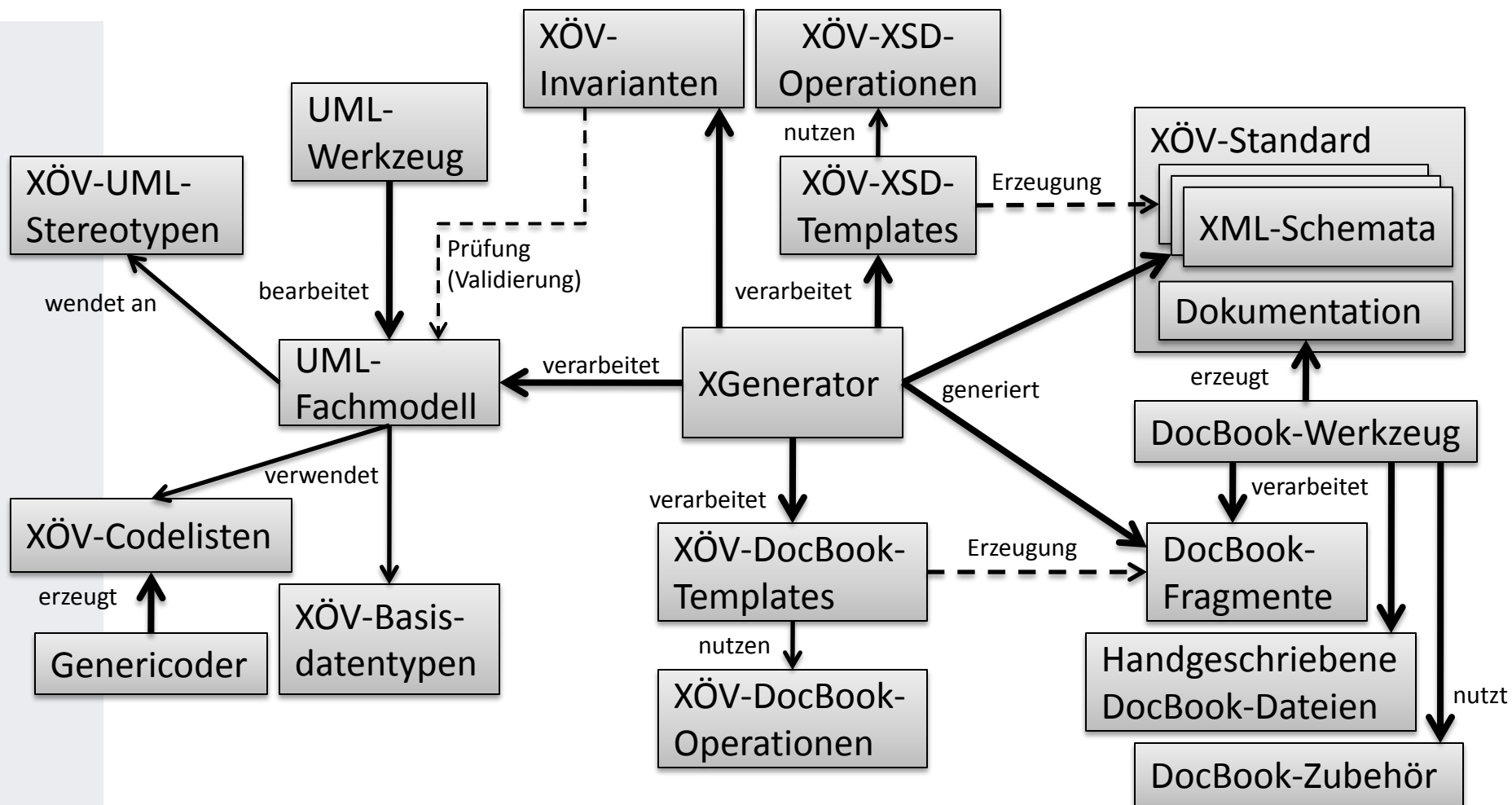
XÖV-Übersicht

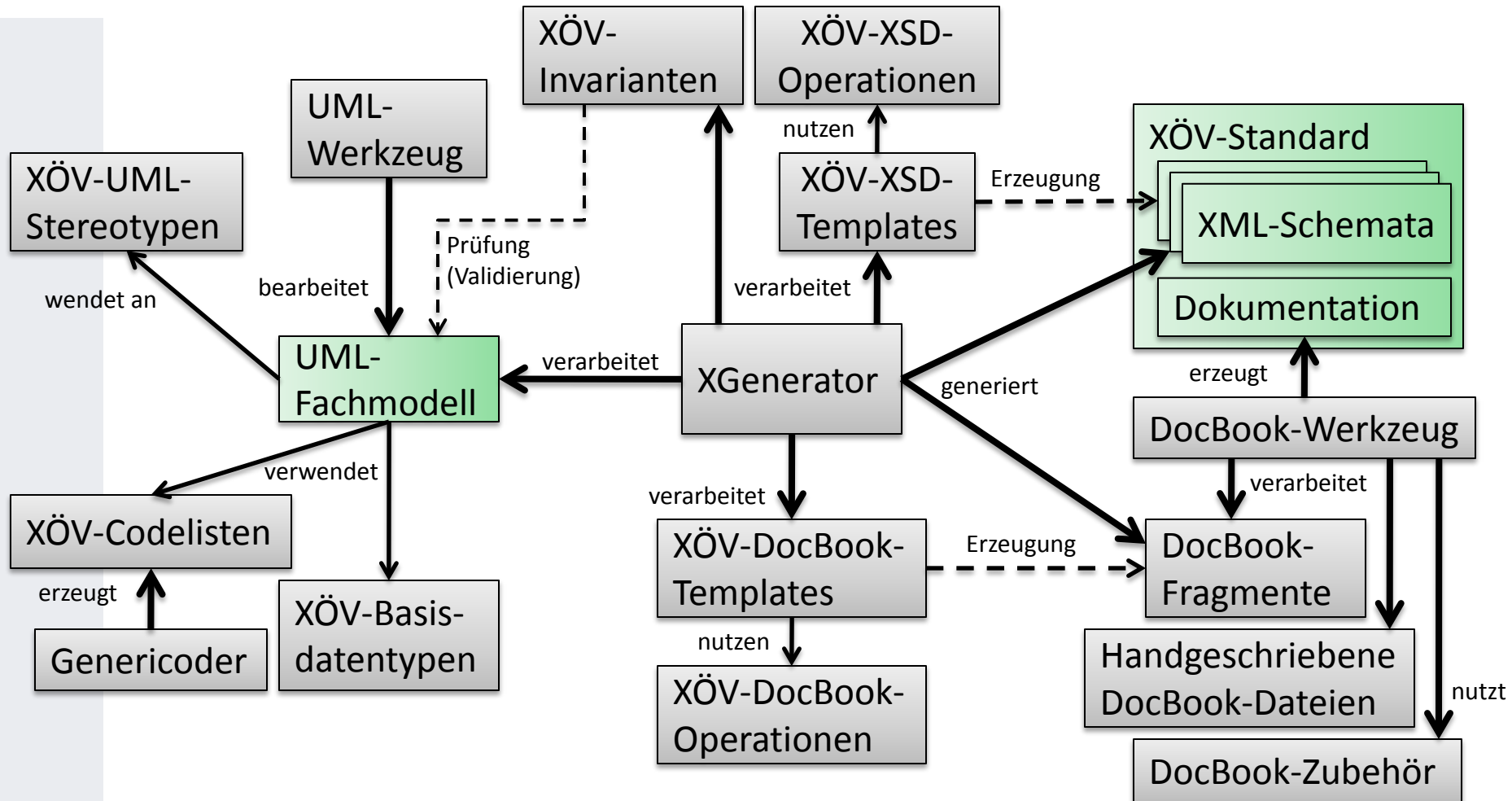


XÖV-Übersicht

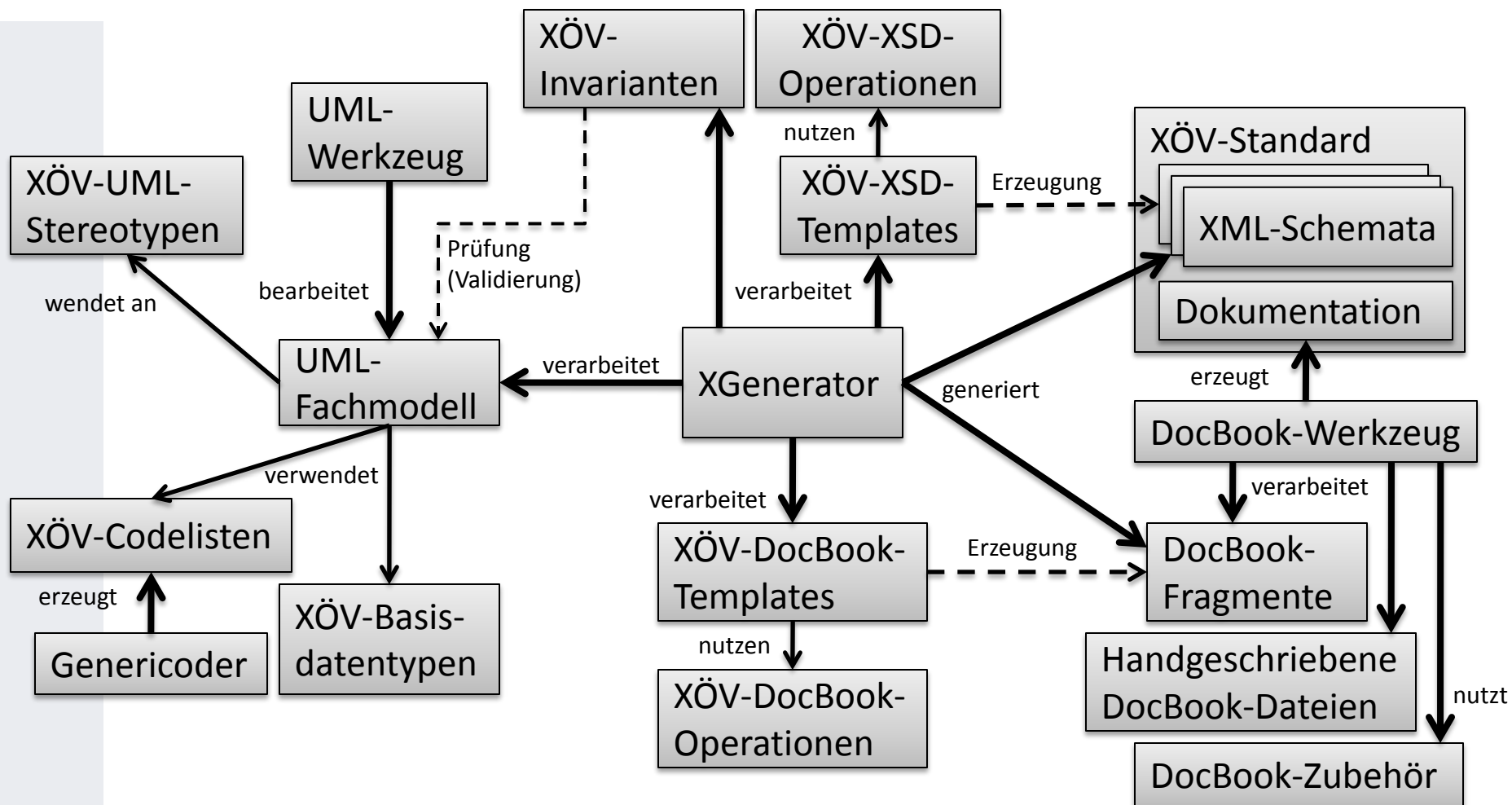


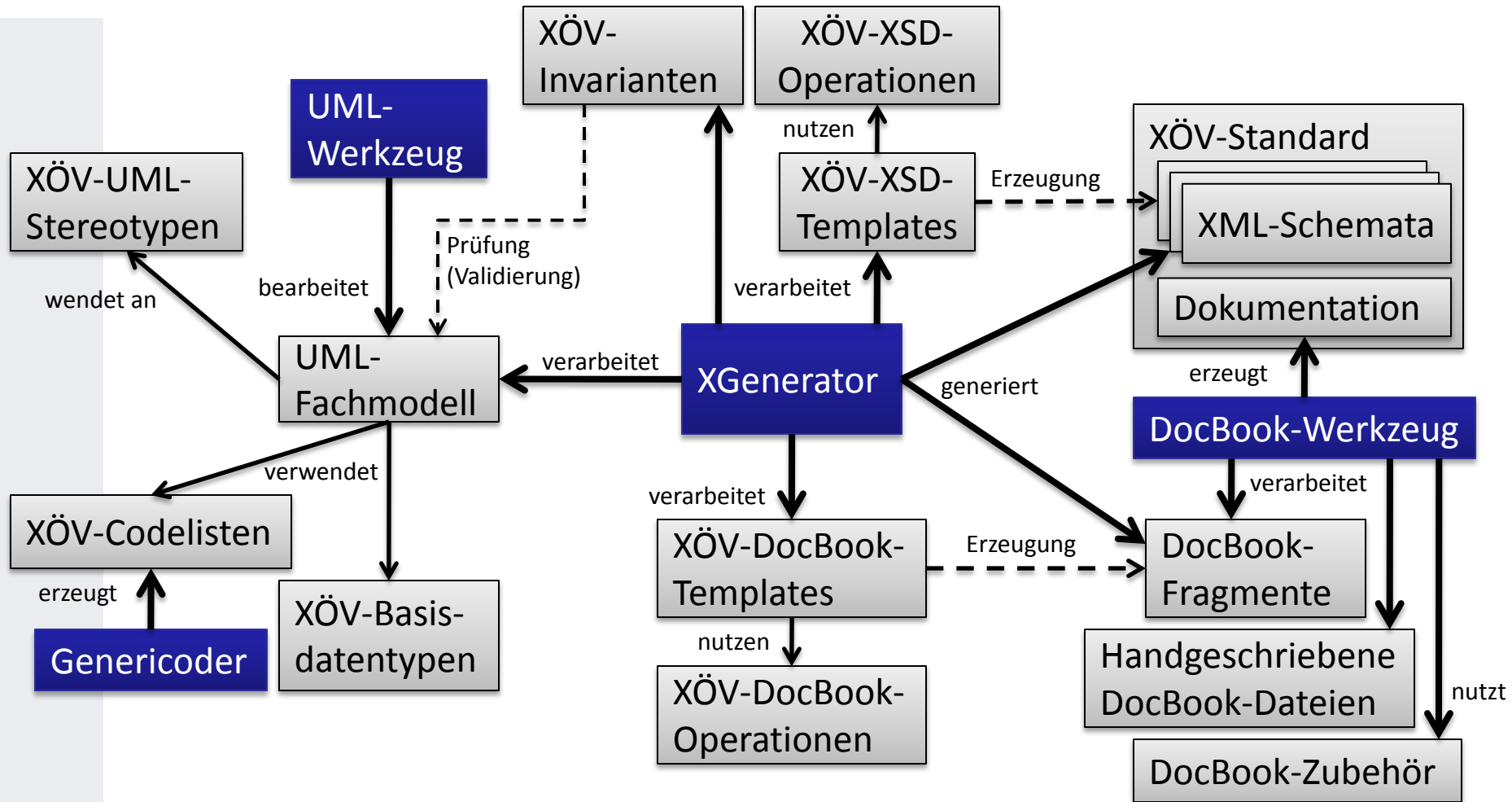
XÖV-Übersicht



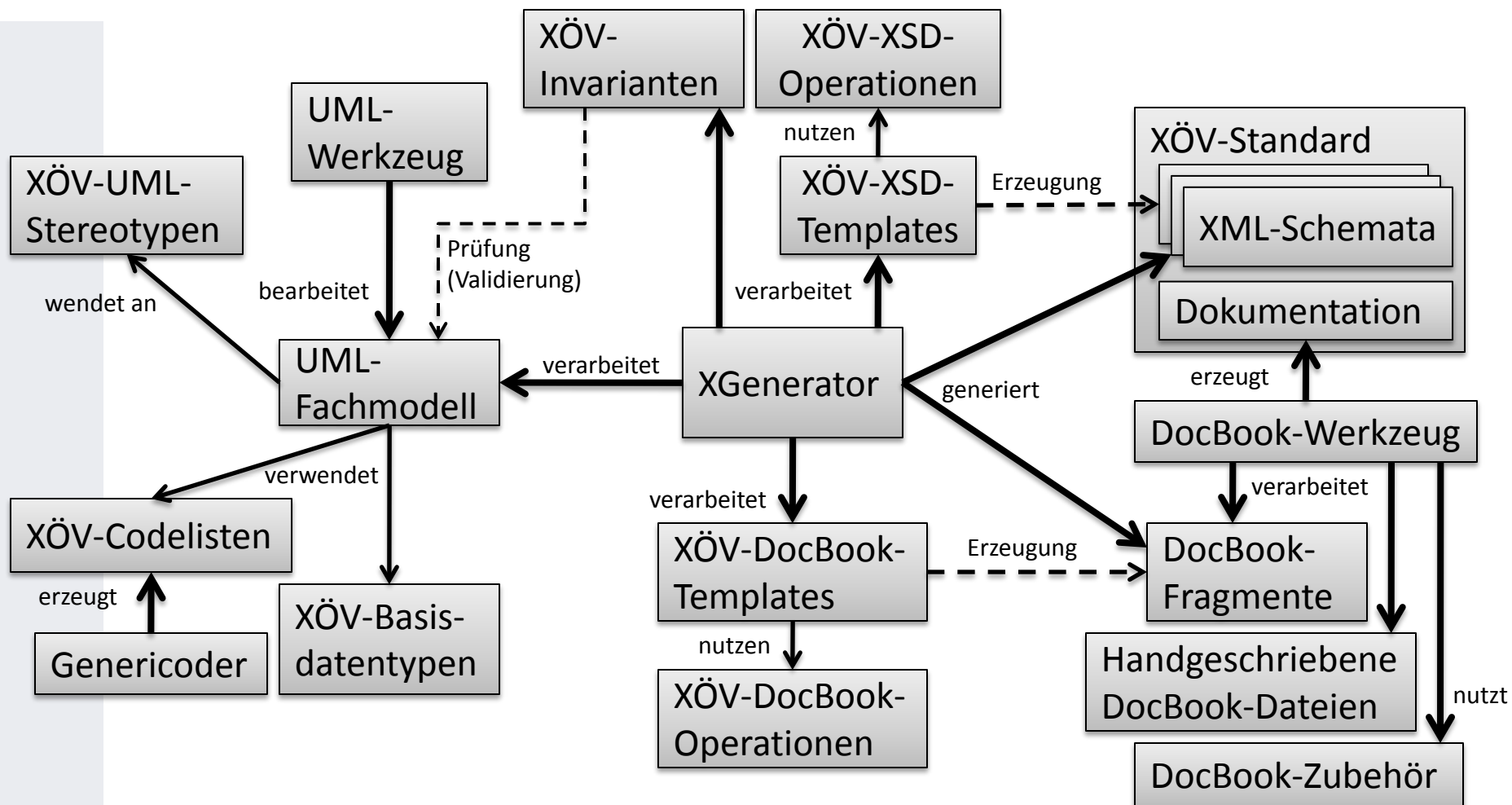


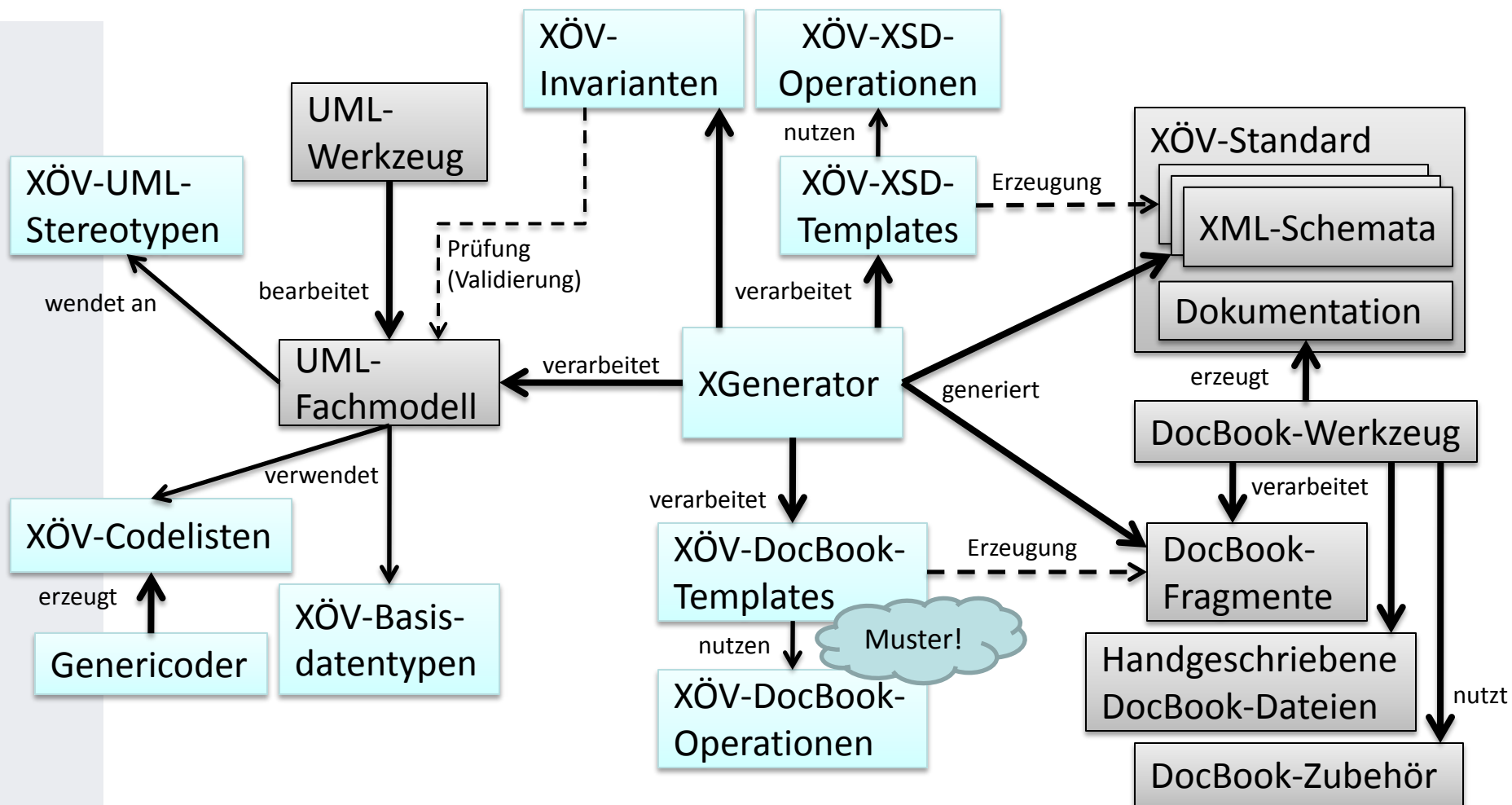
XÖV-Übersicht

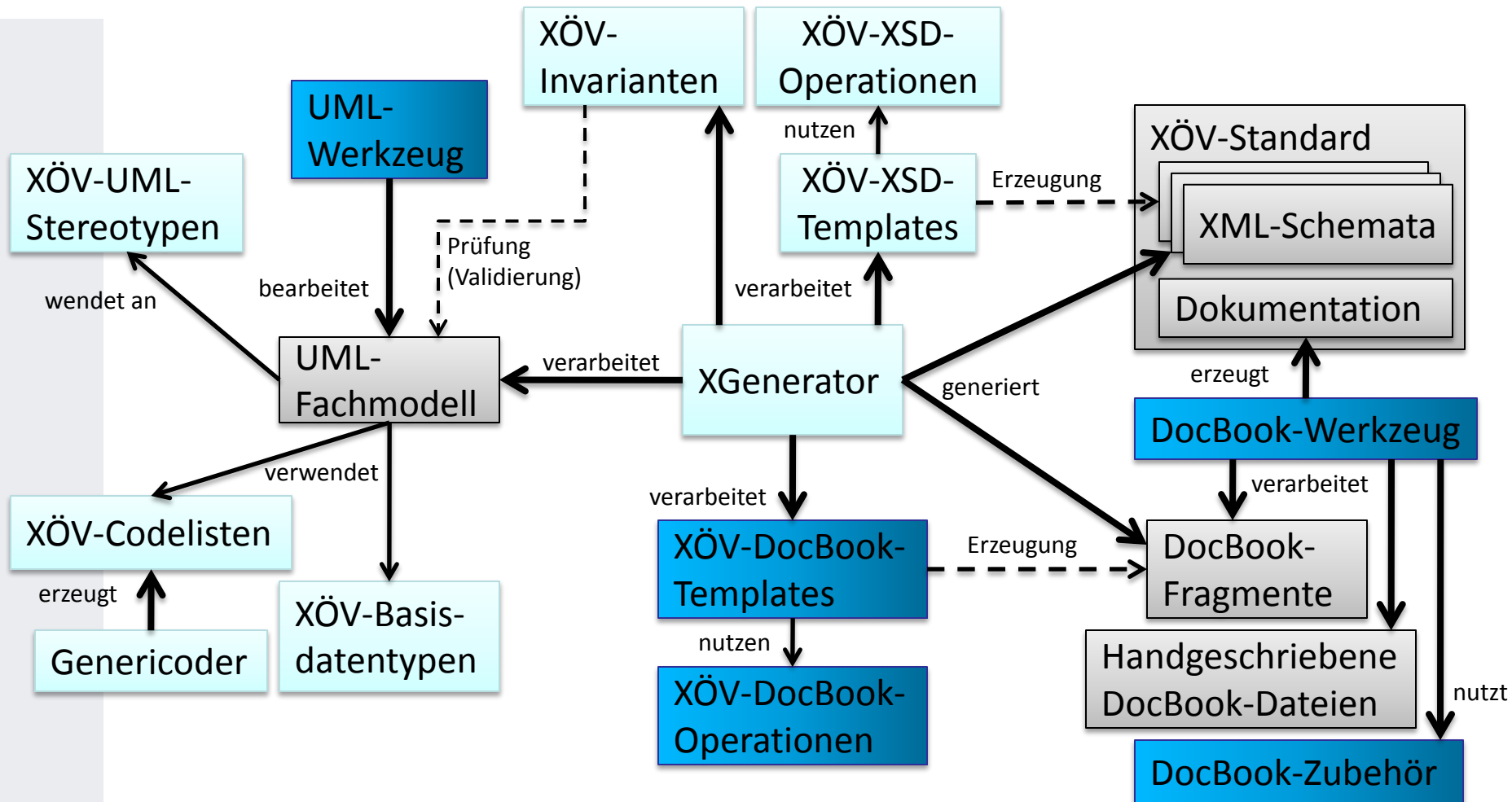


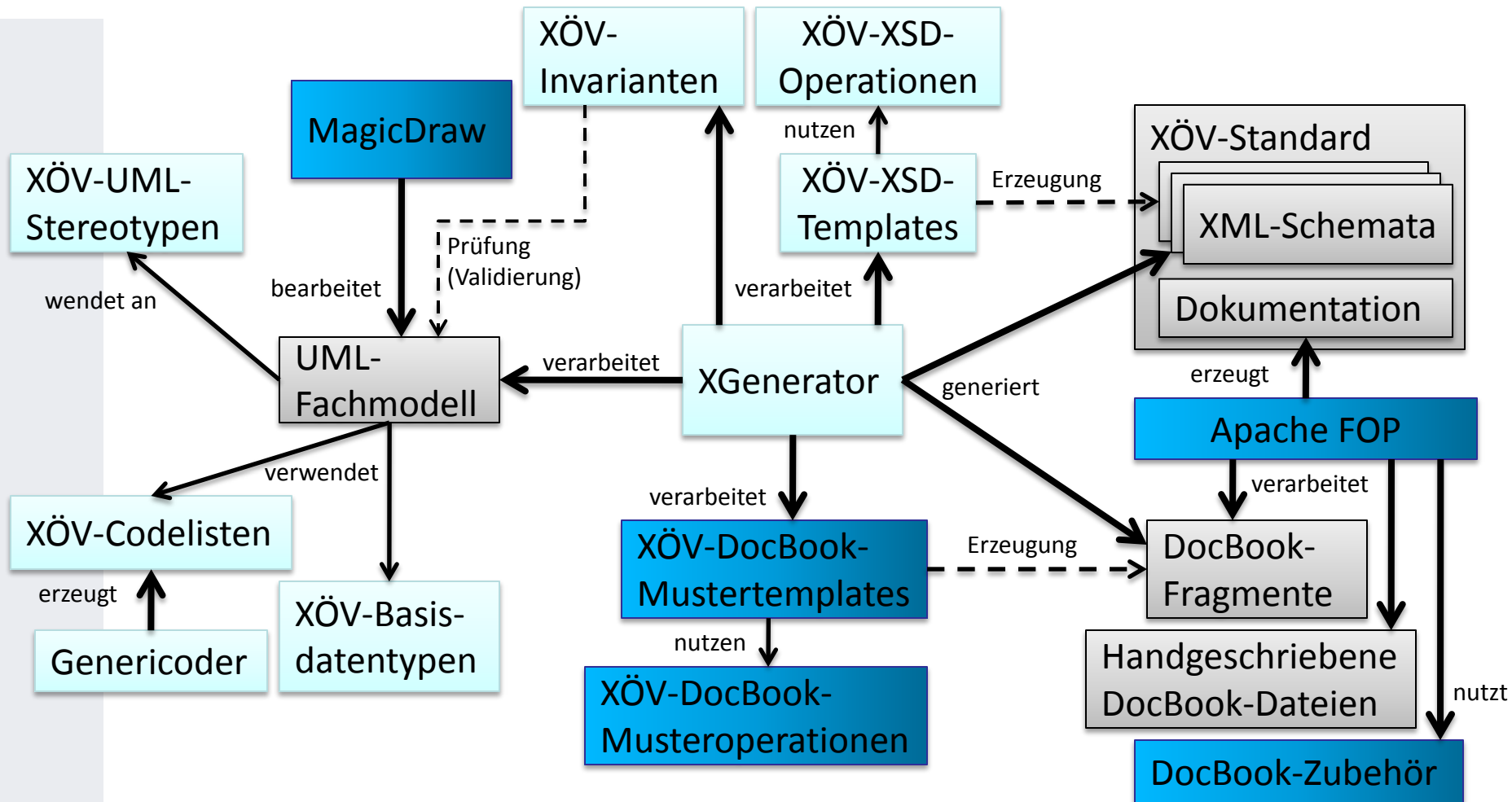


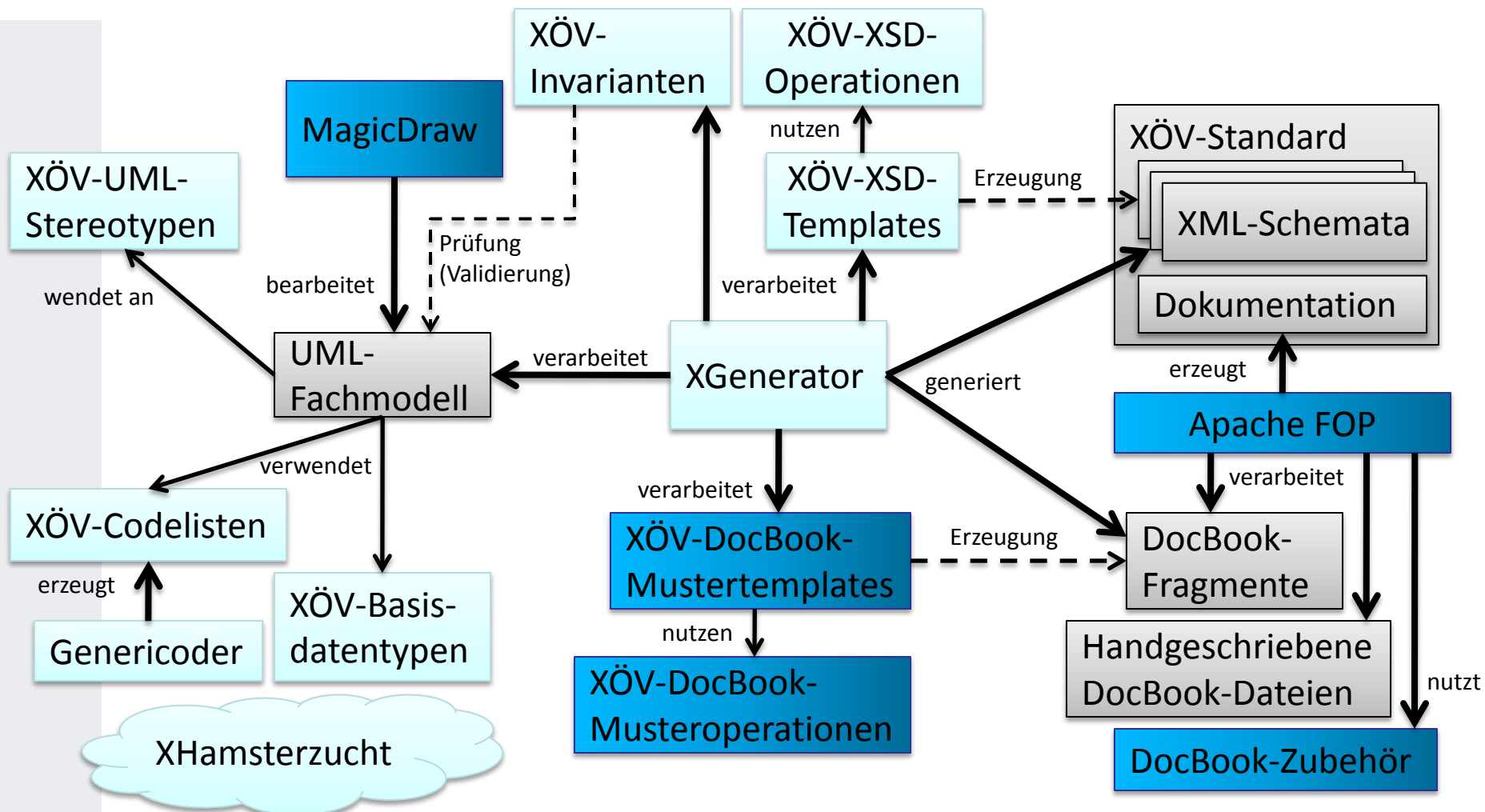
XÖV-Übersicht

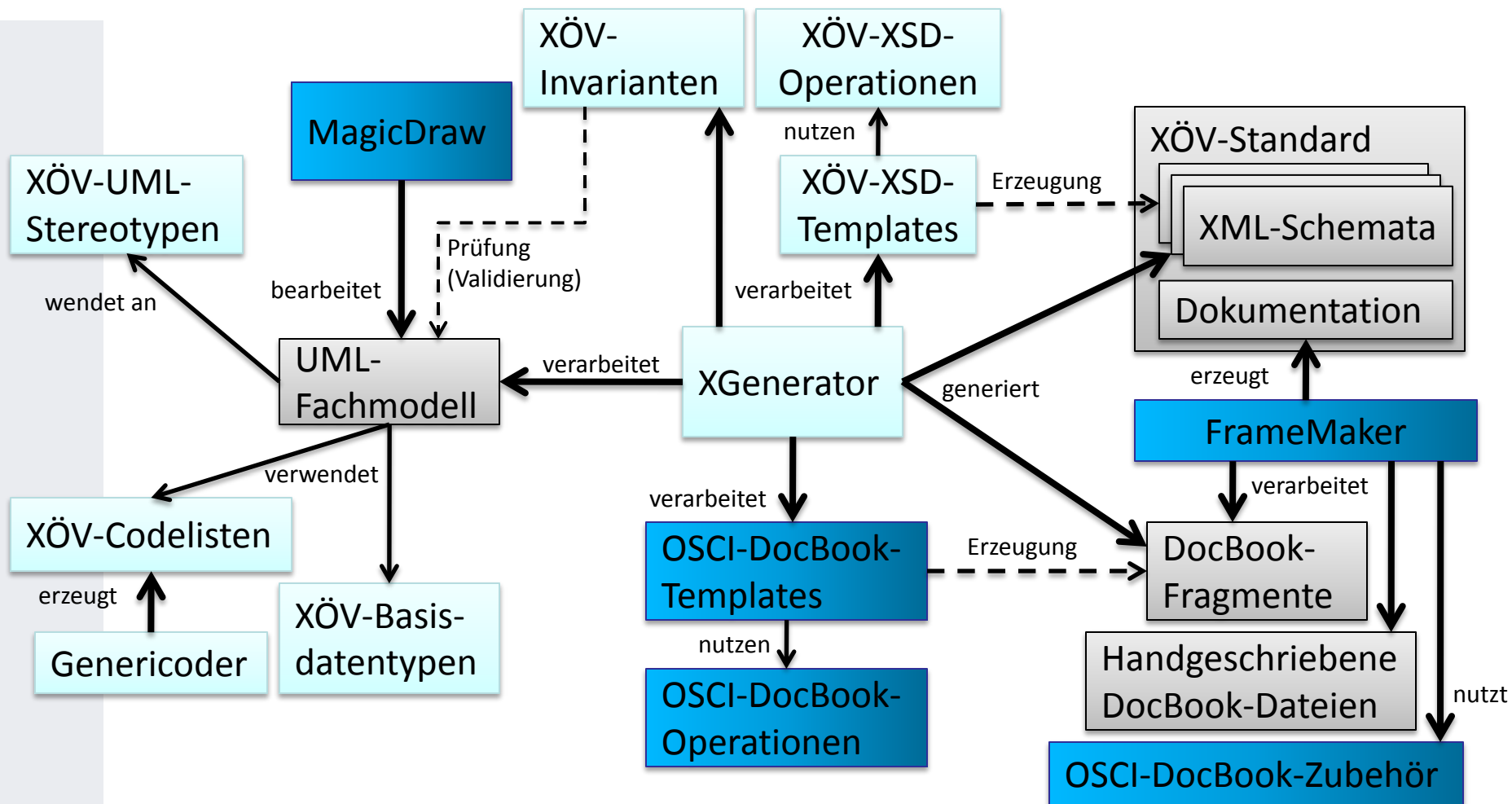


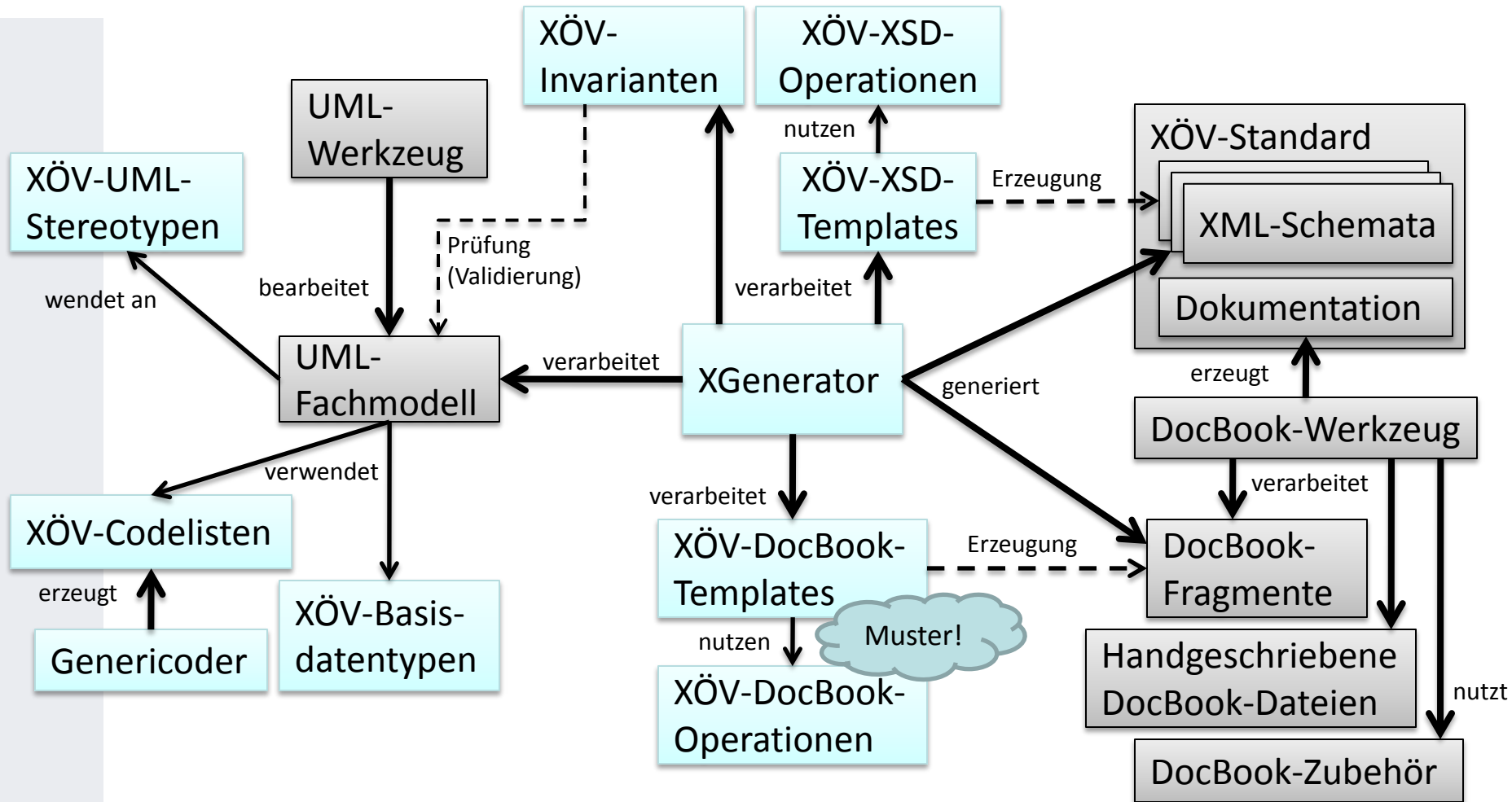












XÖV-UML-
Stereotypen

wendet an

UML-
Fachmodell

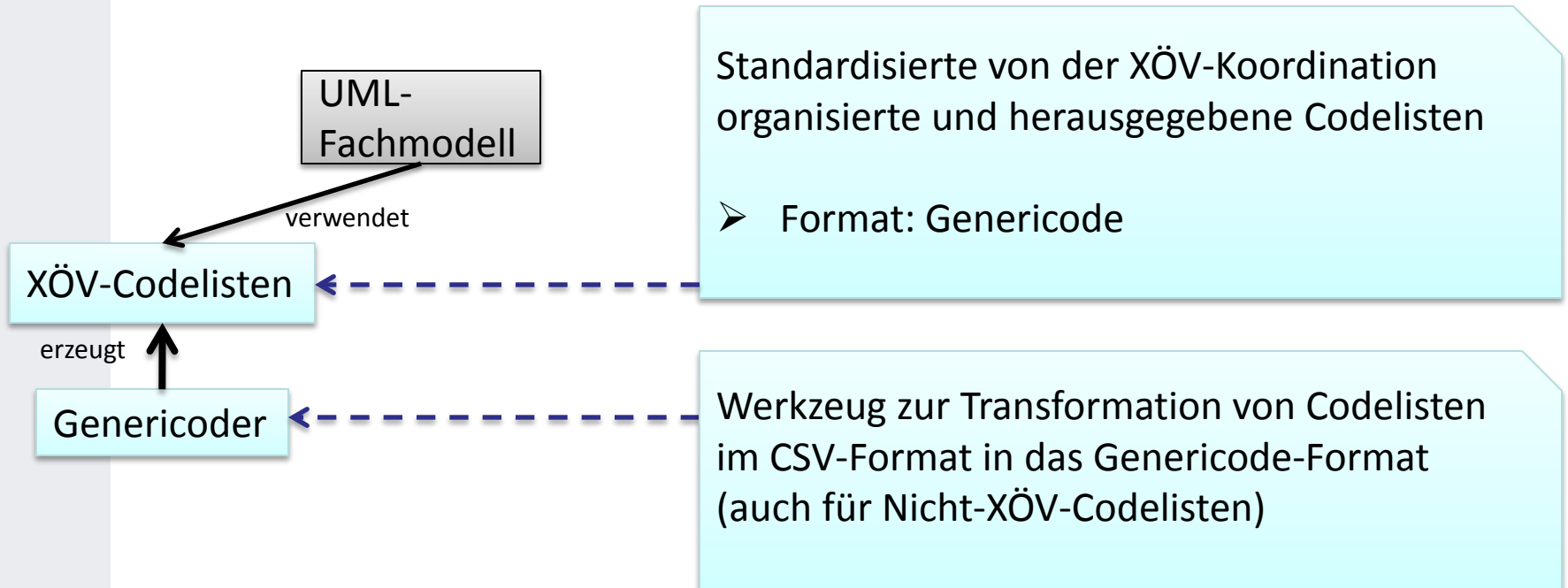
XÖV-UML-Stereotypen zur Bestimmung der technischen Ausgestaltung fachlicher Inhalte auf XML-Schema-Ebene, sowie zur Steuerung der DocBook-Ausgabe.

- Modellierte, fachliche Inhalte werden um Stereotypen und ggf. dazugehörige Eigenschaften ergänzt.



-Produktionszubehör im Detail

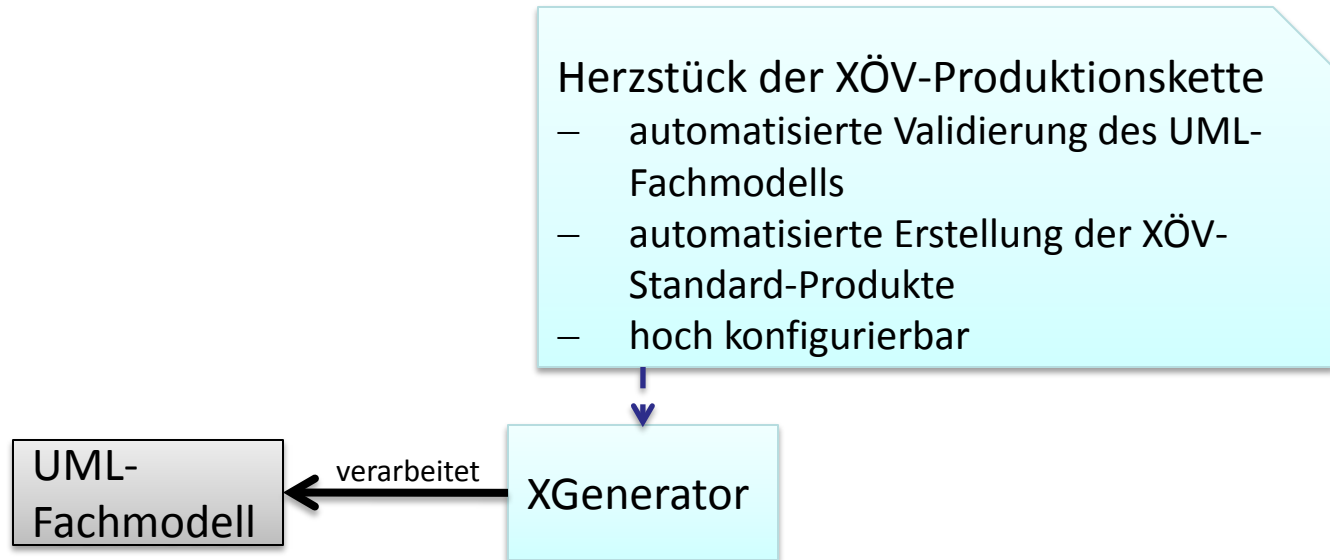
XÖV-Codelisten und Genericoder

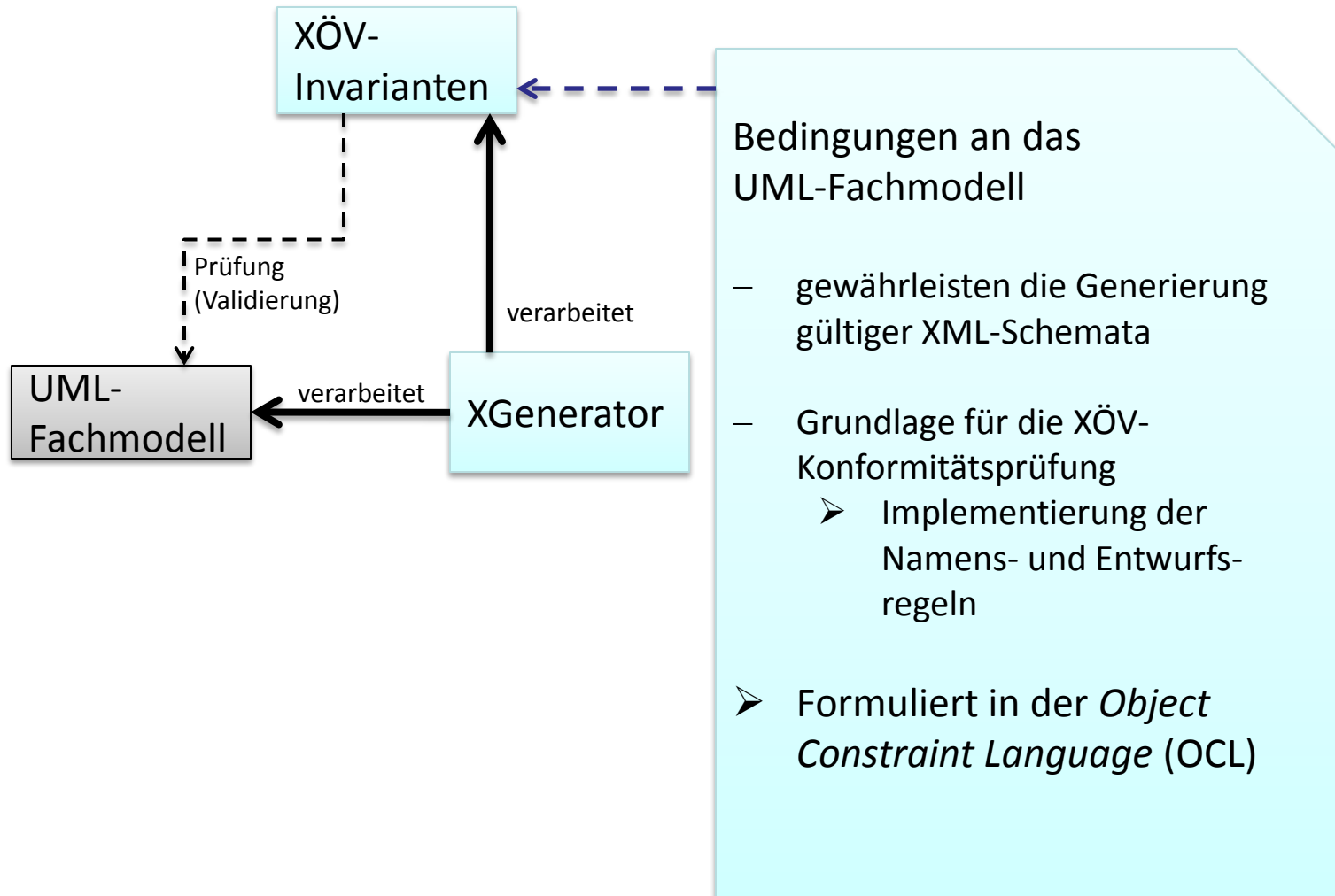




-Produktionszubehör im Detail

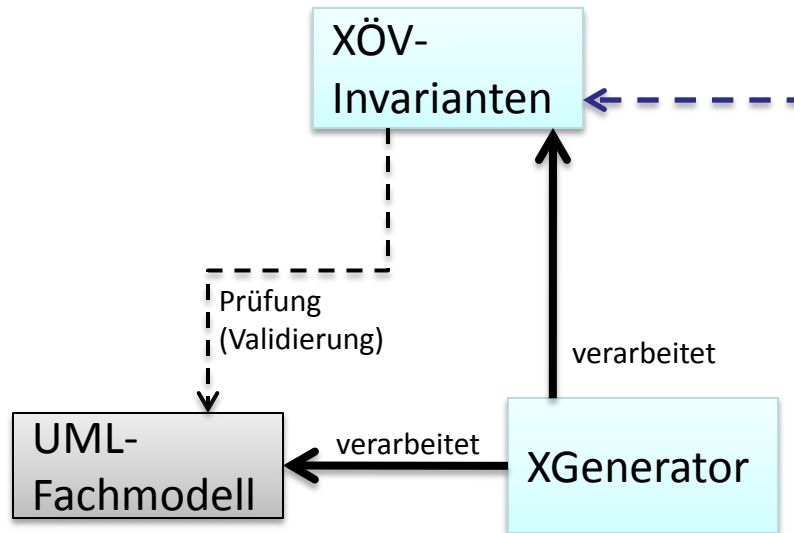
XGenerator







-Produktionszubehör im Detail

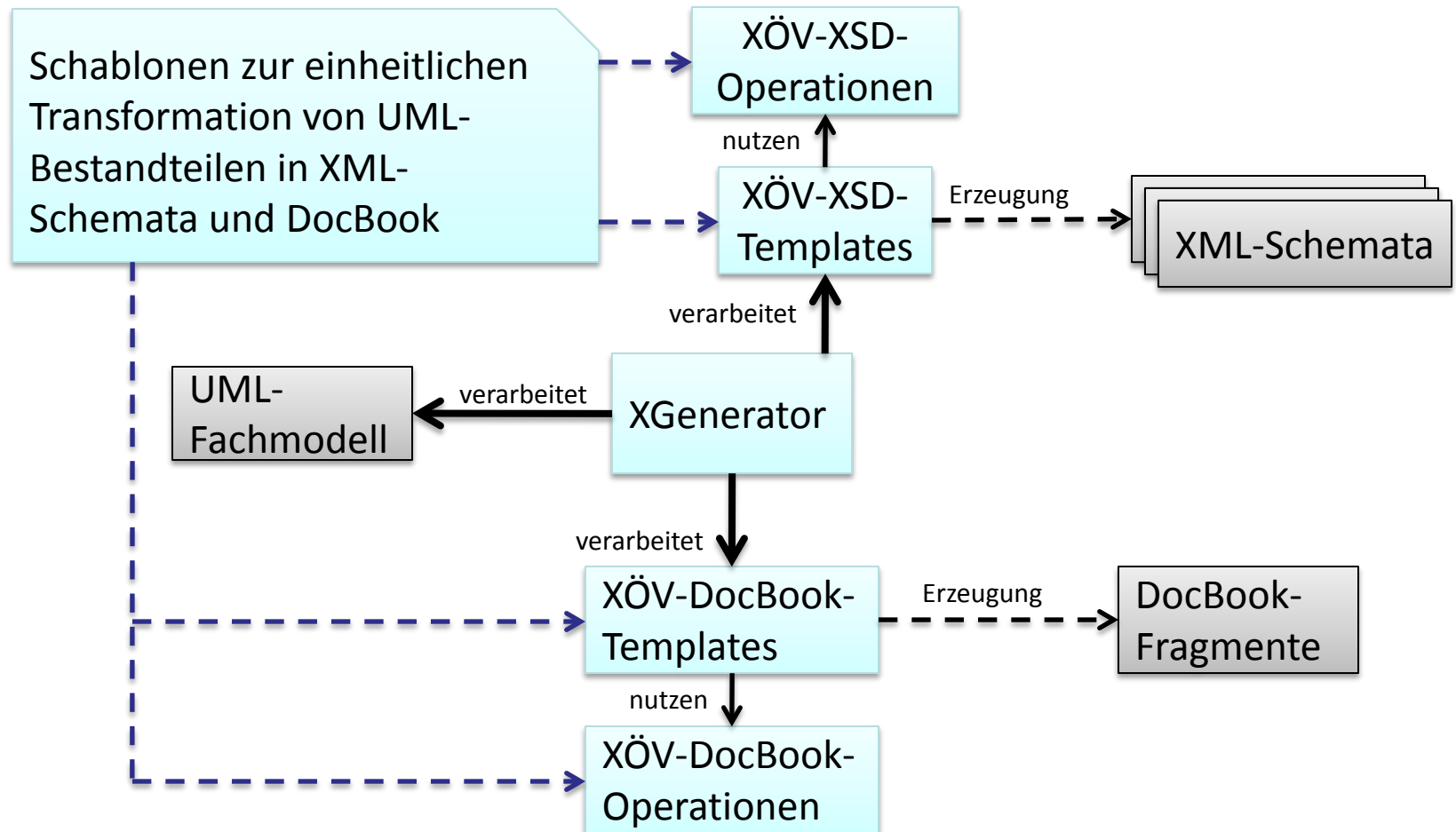


Bedingungen an das UML-Fachmodell

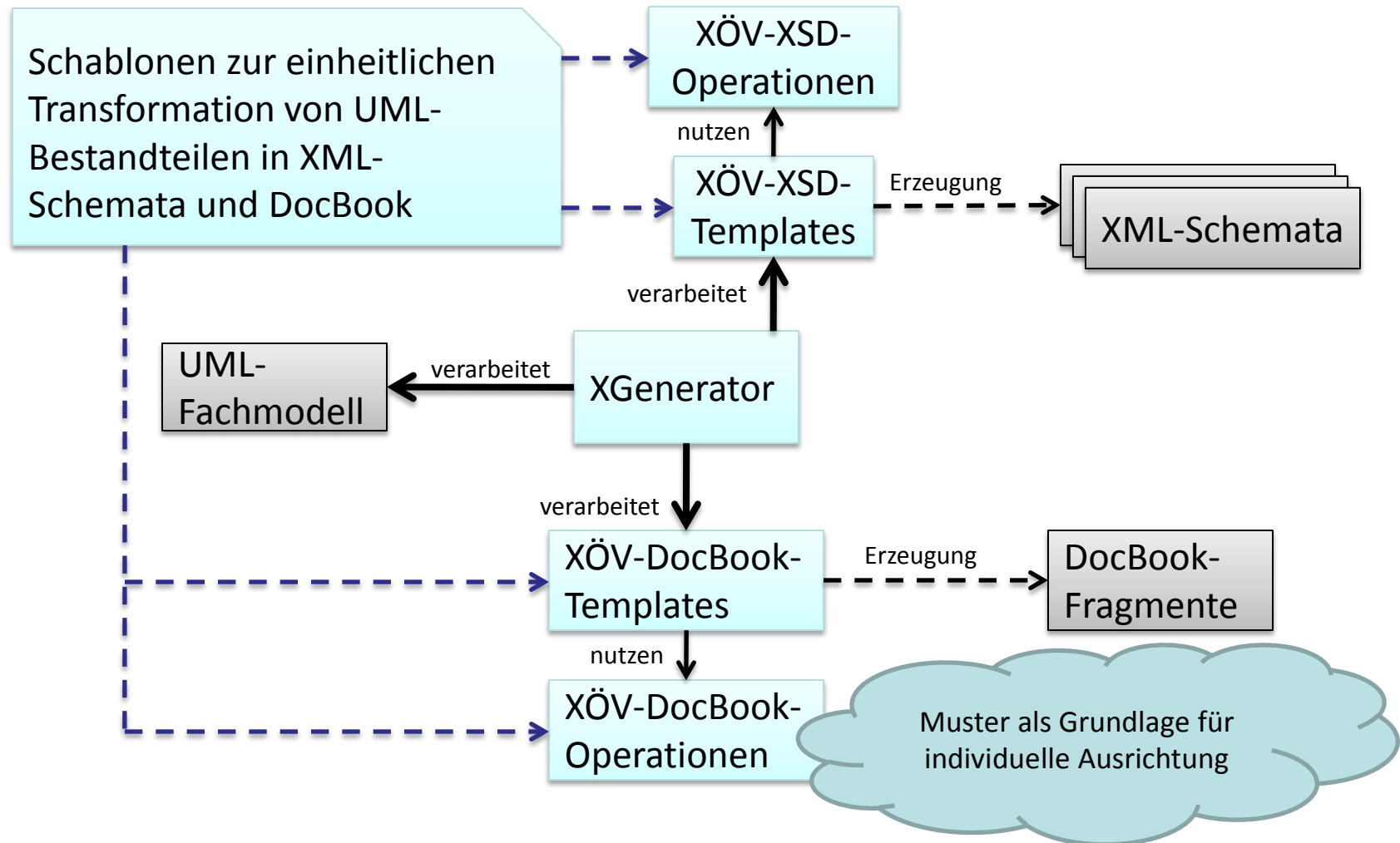
- gewährleisten die Generierung gültiger XML-Schemata
- Grundlage für die XÖV-Konformitätsprüfung
 - Implementierung der Namens- und Entwurfsregeln
- Formuliert in der *Object Constraint Language (OCL)*

NDR-30 (MUSS): Versionierung der Schemata

```
<invariant context='Package' name='SchemaMustBeVersioned'>
  <body><![CDATA[
    let schemaVersion:String = self.hlpPackageVersion() in
    self.extension_xsdSchema.isDefined and
    self.hlpIsPartOfXModel() implies
      schemaVersion.isDefined() and schemaVersion <> ''
  ]]></body>
</documentation>
  Die Eigenschaft "version" muss im Stereotyp «xsdSchema»
  oder im «xsdXModel» definiert sein.
</documentation>
</invariant>
```



XÖV -Produktionszubehör im Detail



Arbeitsblock

Einsatz des XÖV-Zubehörs in der Praxis

- Wie wird das XÖV-Zubehör eingesetzt?
- Wie tritt das Zubehör auf Dateiebene in Erscheinung?
 - Welche Dateien gehören zum Zubehör?
 - Welche Rolle spielen sie?
 - Wie werden sie genutzt?
- Wie tritt das Zubehör im UML-Fachmodell in Erscheinung?
 - Welche Bestandteile sind direkt im Modell sichtbar?
 - Welche Bestandteile werden täglich während der Modellierung genutzt?
- Welchen Nutzen bringt das Zubehör sonst noch?
 - Namens- und Entwurfsregeln
 - XSD- und DocBook-Templates
- Was ändert sich für bestehende XÖV-Projekte?



Die Beauftragte
der Bundesregierung
für Informationstechnik



Freie
Hansestadt
Bremen

Vielen Dank für Ihre Teilnahme!

Lars Hamann, TZI

(E-Mail: lhamann@tzi.de)

Mirco Kuhlmann, TZI

(E-Mail: mk@tzi.de)



Die Beauftragte
der Bundesregierung
für Informationstechnik



Freie
Hansestadt
Bremen

Ergebnisse Workshop 1: XÖV-Produktionszubehör und -Produktionsumgebung



- Plugins:
 - Verfügbarkeit? / Preis?
- Templates für FrameMaker
 - Können diese bezogen werden?
- Beispielnachrichten XHamsterzucht
- XÖV-Konformität bei Weiterentwicklung XÖV?
 - Standard vorher XÖV –konform danach nicht mehr
 - Vorschlag: Label + Version
 - Voraussichtliche Entwicklung von Regeln
- Versionsmanagement in Produktionsumgebung aufnehmen bzw. Vorschläge machen?
- PU automatisierbar?
- XPfleger Schulung (Preis?)

- XGenerator
 - Regeln flexibel an-/abschaltbar
 - Kontextsensitive Informationen bei Fehlern
 - Kategorien für Fehler (Fehler, Warnung, Hinweis)
 - XMI Format
 - Alle Werkzeuge wird nicht funktionieren
 - Was sind die relevanten Werkzeuge?
 - Kopplung mit Eclipse basiertem UML-Werkzeug (Link auf Fehler)
 - Testkonzept für Werkzeuge
- Schematron für Geschäftsprozesse
- DocBook Struktur mit Entities kompliziert
 - Konfigurationsmöglichkeit für die Einbindung
- Ausgabe von ODF (Open Document Format)

- Synergieeffekte mit anderen PUs (V-Modell) nutzen
 - Workshop geplant
- Details zu Arbeit mit der PU im XÖV-Handbuch
- Formulare generieren