Web-Programmierung (WPR)

Vorlesung III.

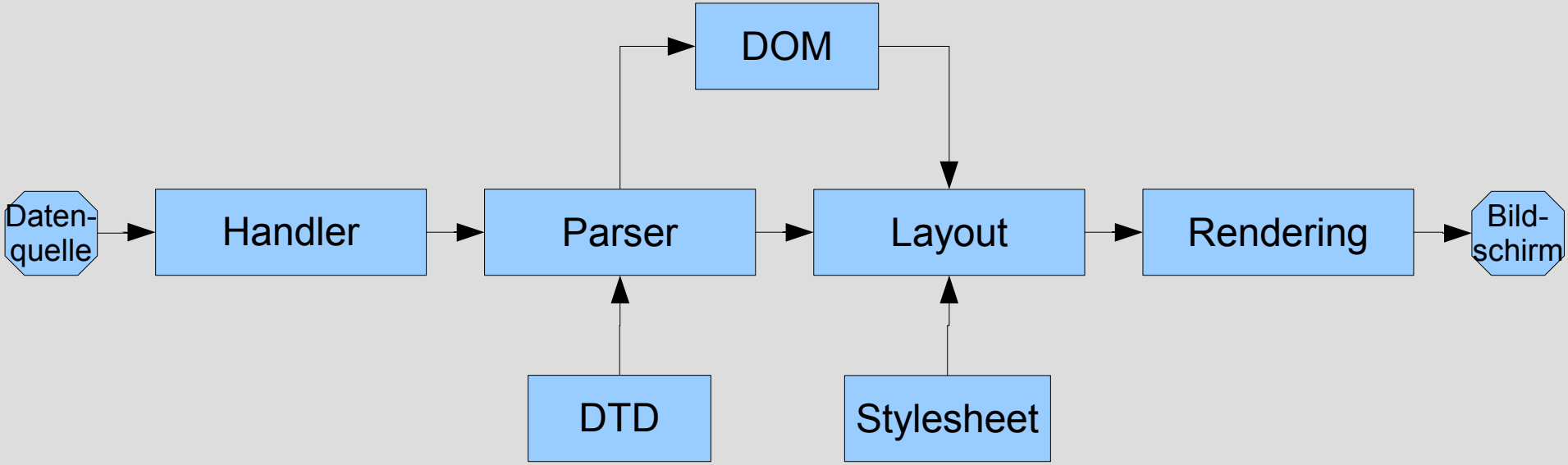
CSS + XSLT

Manfred Gruner

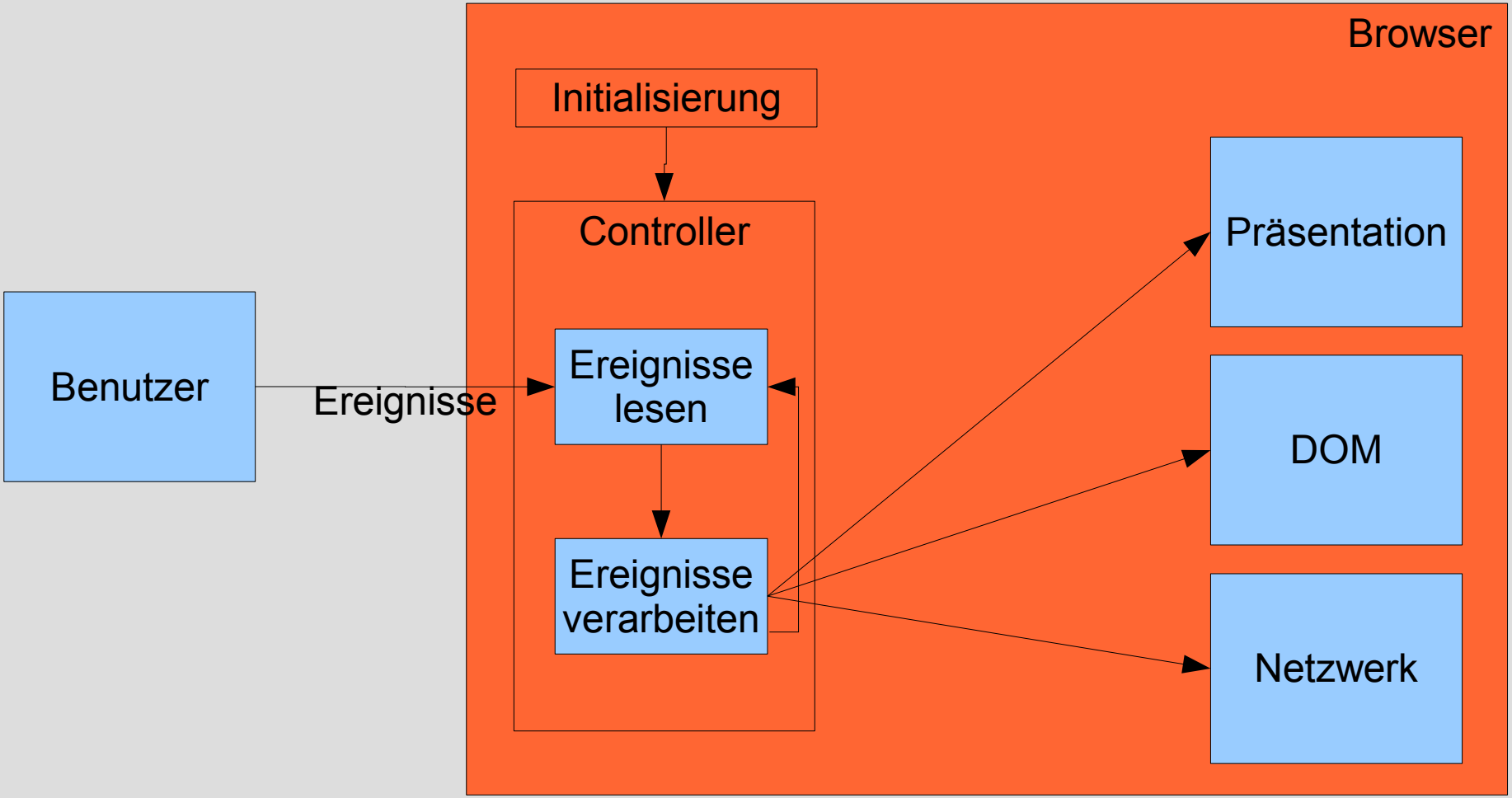
mailto:wpr@gruner.org

4.4 Layout-Prozess

Vereinfachte Darstellung des Layoutprozesses für HTML-Dokumente



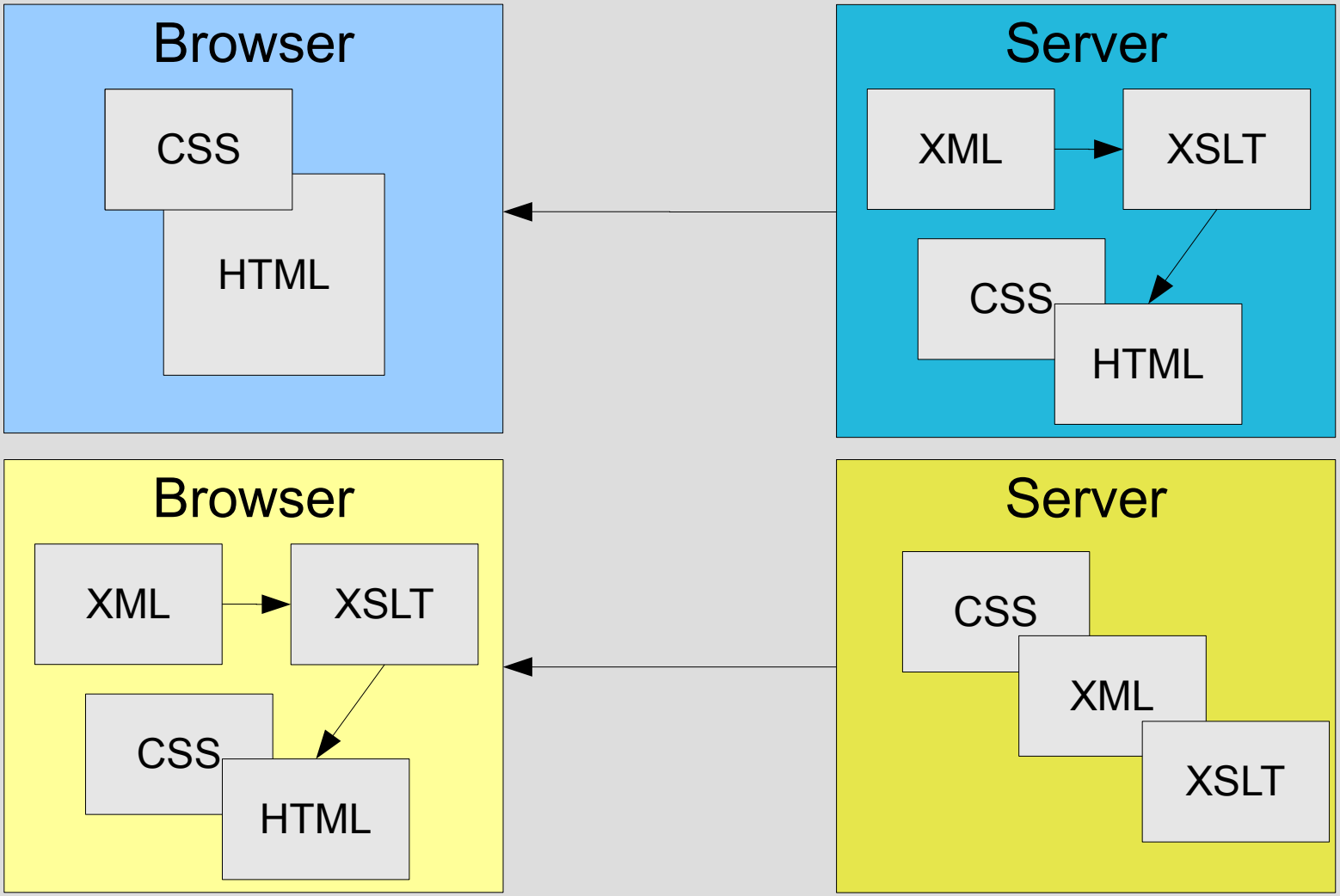
4.5 Ereignisorientierung



5 Präsentation

- Präsentation von HTML u. XML-Dokumenten
- Konsequente Trennung von
  - Struktur u.
  - Präsentation
- Vorteile
  Aufgabentrennung
  - Strukturerstellung (Content)
  - Präsentationerstellung

5 Präsentation



5.2 Cascading Style Sheets (CSS)

- Beschreibung von Layout und Präsentationsfeatures

  HTML und CSS
- Ziele
  - Reduzierung von HTML auf Strukturbeschreibung
  - Trennung von Inhalt und Präsentation
  - Layout-Kontrolle durch Author
  - Zentralisierung der Layout-Konfiguration
  - Reduzierung der Dokumentgröße

5.2 Cascading Style Sheets (CSS)

- Anwendungsalternativen
  - Style Attribute

    <body style="background-color: #EEEEEE; font-family: sans-serif">

  - Keine großen Vorteile gegenüber

5.2.2 Anwendungsalternativen

- Style Tag

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
<style type="text/css">
<!--
 p  {  margin-left: 50pt;
       margin-right;    50pt;
       font-family: sans-serif;
       background-color: white;}

 h1   {color:            "#FF0000";
       text-align:  center;}

 body {background-color: "#0022FF";}
-->
</style>
  <title></title>
</head>
<body>
<br>
</body>
</html>
```

5.2.2 Anwendungsalternativen

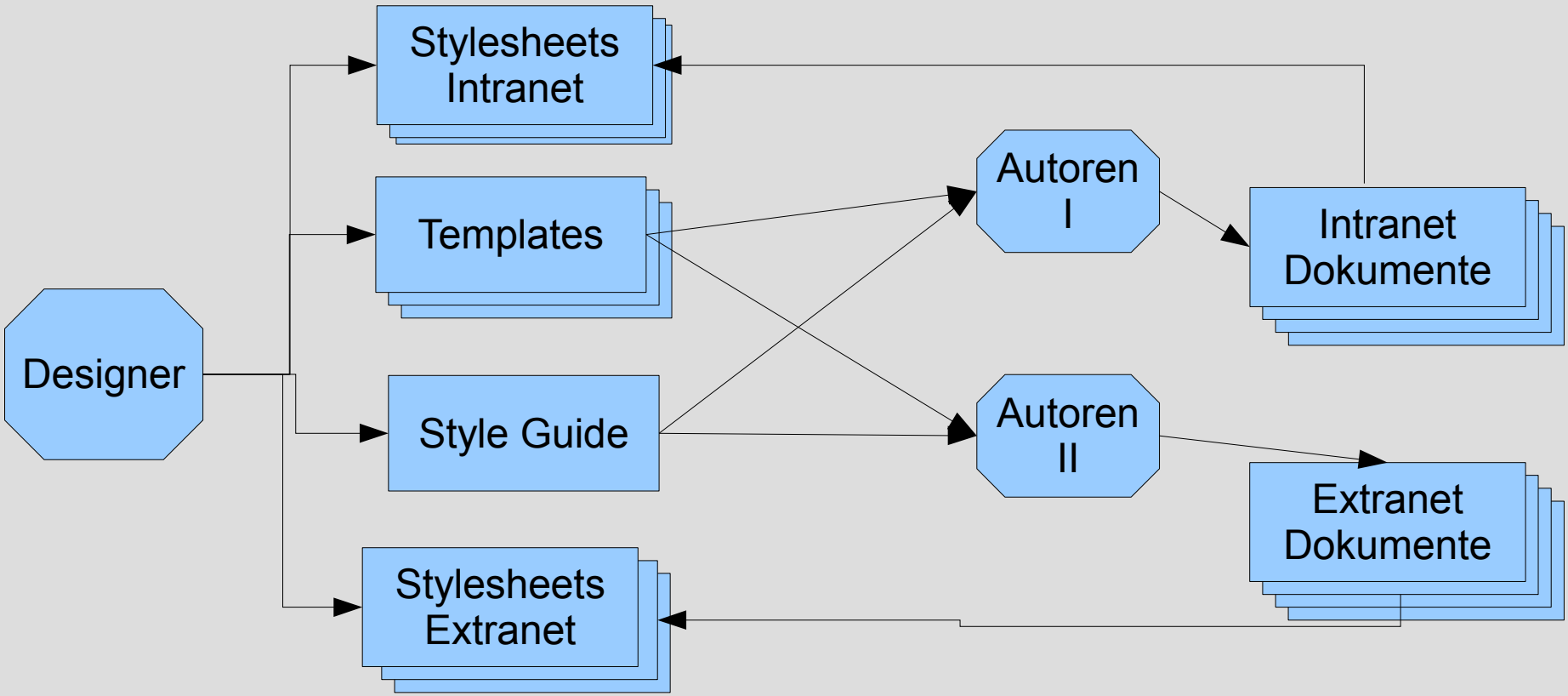
- Externe Stylesheets

```
p   {  margin-left:  50pt;
        margin-right;   50pt;
        font-family: sans-serif;
        background-color: white;}

 h1   {color:          "#FF0000";
        text-align:  center;}

 body {background-color: "#0022FF";}
```

design.css

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
 <link rel="STYLESHEET" href="/css/design.css type="text/css"/>
</head>
<body>
<br>
</body>
</html>
```

5.2.2 Anwendungsalternativen



5.2.3 Regeln

- Definition einer Style Regel

```
selektor { Eigenschaft:Wert; }

s1,s2 { Eigenschaft:Wert;
        Eigenschaft2:Wert;
        }
```

Selektoren sind häufig HTML-Tags

5.2.3 Regeln

- Klassifizierung (Tag-Klassen)

  Unterschiedliche Anwendungsfälle
  ==> unterschiedliche Style-Angaben

```
selector.class {styleanweisungen}

p.achtung {font-weight:bold;}
p.beispiel {font-family:monospace}
```

  Beispiel:

```
<p class="achtung">Achtung Absatz</p>
```

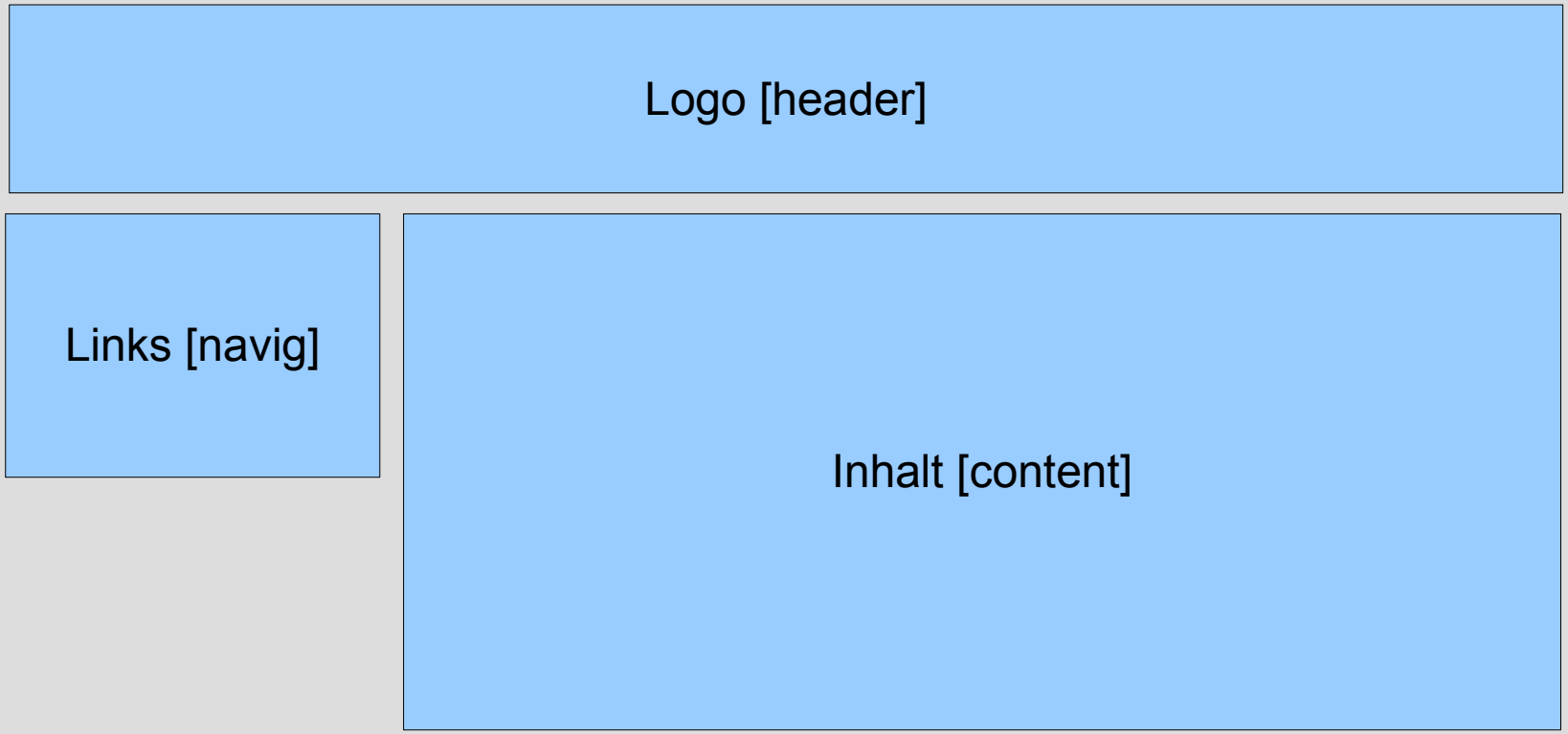
5.2.3 Regeln

- Zusätzliche HTML-Tags bei Anwendung von CSS
  - `<span></span>`
  - `<div></div>`

- span
  - Stil-Definitionen in einem Blockelement
- div
  - Stil-Definition über mehrere Blockelement hinweg

5.2.3 Regeln

- Beispiel



5.2.3 Regeln

- Pseudo-Klassen

  Unterschiedliche Zustände des Elementes
  ==> unterschiedliche Style-Angaben

```
selector:class {styleanweisungen}
```

  Beispiel:

```
a:link {color:red;}
a:visited {color:maroon;}
a:active {color:grey;}
```

5.2.3 Regeln

- IDs
  - Attribut seit HTML 4.0 u. XHTML
  - IDs können in einem URL als Referenz in ein HTML-Dokument benutzt werden.
  - ID kann auch in CSS genutzt werden

    ```
    #kernaussage {font-familyserif;
                  color: red;}
    ```

  - IDs müssen/sollten in einem Dokument eindeutig sein.

5.2.3 Regeln

- Baumstruktur

  Vererbung von CSS-Eigenschaften

- Kontextdefinition

```
ol          {list-style:decimal;}
ol ol       {list-style: upper-alpha;}
ol ol ol    {list-style: lower-alpha;}
```

5.2.8 XML und CSS

- XML enthält keine Präsentation
- CSS fügt diese Informationen bei (nur externes Stylesheet)
- Wichtigste CSS-Eigenschaft ist für XML
  - `display` = {`block` = absatzorientiert
    ,`inline` = zeichenorientiert}
    `display:inline` = default

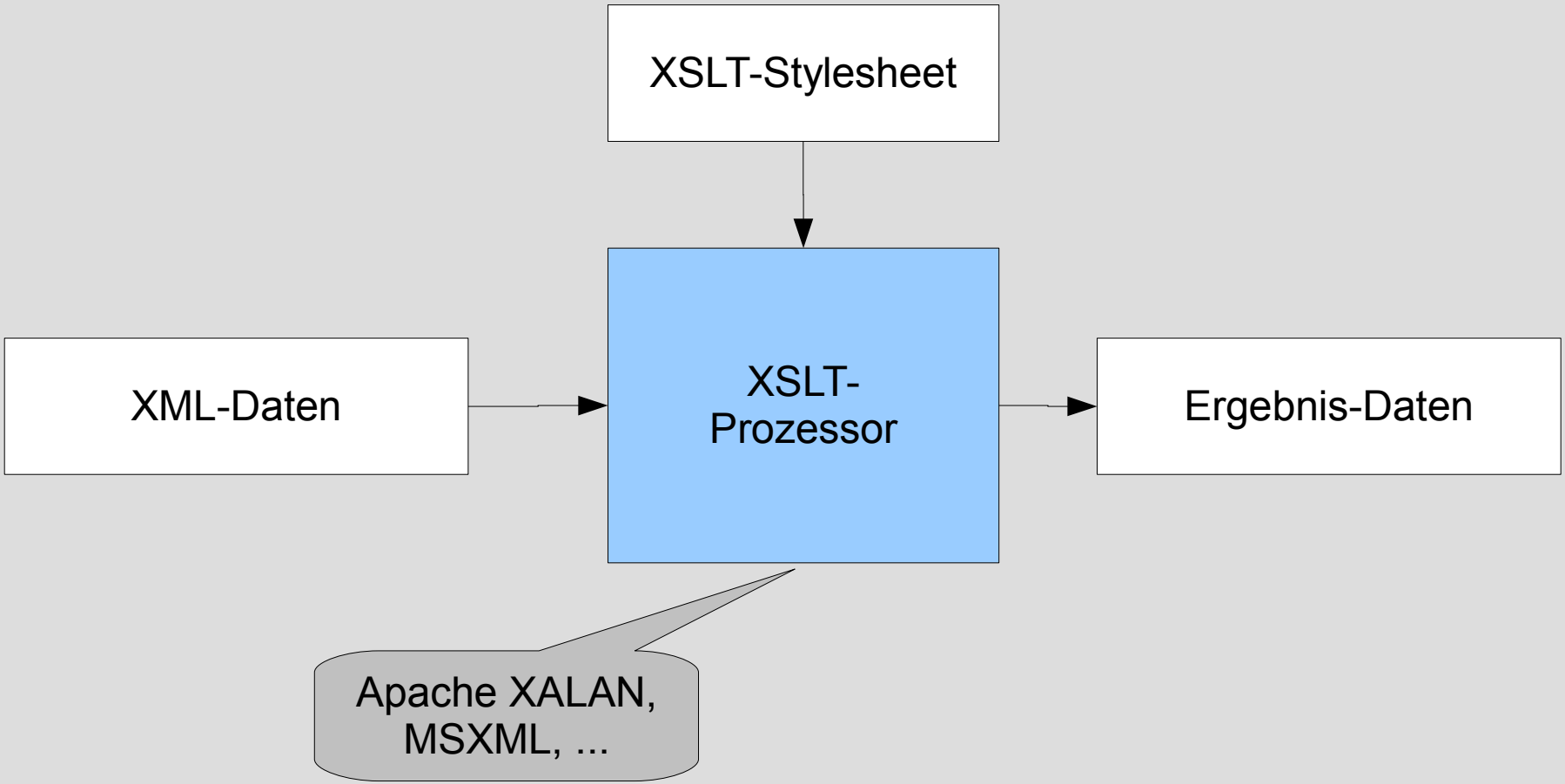
- Verknüpfung

```
<xml-stylesheet type="text/css"
                href="seminar.css"?>
```

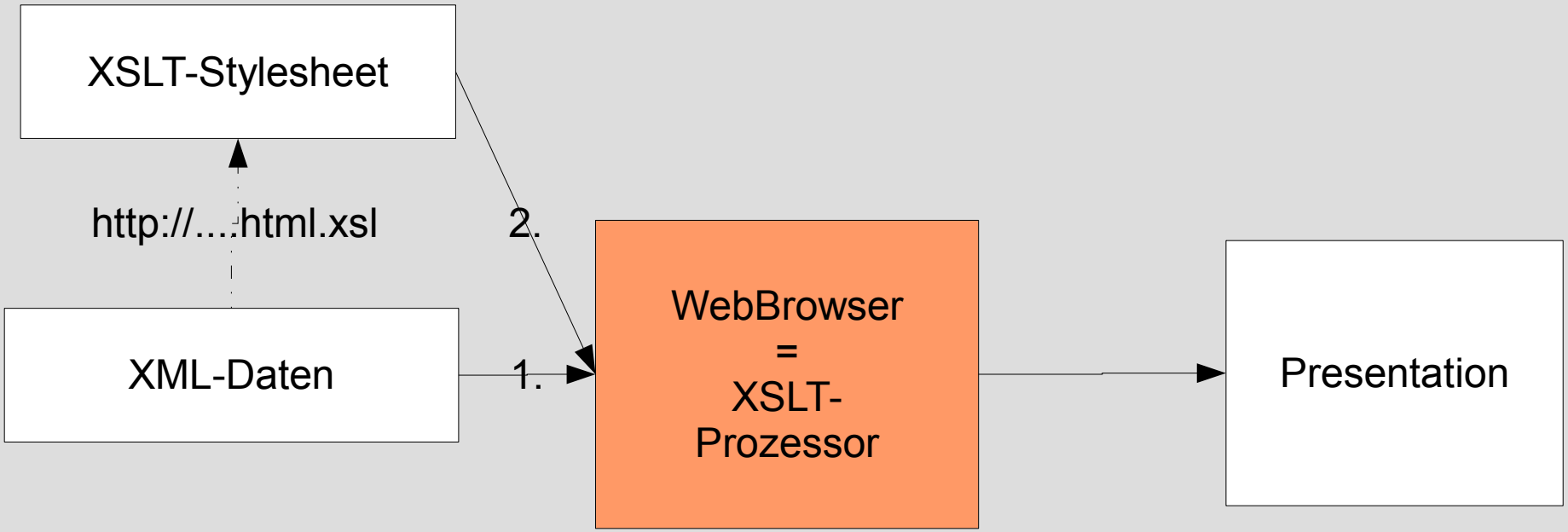
5.3 XSL

- eXtensible Stylesheet Language
- CSS => XML-Struktur bleibt erhalten
- XSL => Andere Dokumentenstruktur wird benötigt
- XSL:FO => Allgemeine Seitenbeschreibungsprache
- XSLT => XSL-Transformation

5.4 XSLT


XSLT-Stylesheet
XML-Daten
XSLT-
Prozessor
Ergebnis-Daten
Apache XALAN,
MSXML, ...

5.4 XSLT


XSLT-Stylesheet
http://....html.xsl
2.
XML-Daten
1.
WebBrowser
=
XSLT-
Prozessor
Presentation

5.4.2 Templates

- XSL-Stylesheet

```
<?xml version='1.0'?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

   <xsl:template match="/">

   </xsl:template>

</xsl:stylesheet>
```

5.4.2 Templates

- Transformation erfolgt regelbasiert
- Eine Regel => xsl:template
- Reihenfolge der Regeldefinition ist belanglos
- Regeldefinition

```
<xsl:template match="">... </xsl:template>
```

match-Attribut definiert Knoten

5.4.2 Templates

- Inhalt von `<xsl:template></xsl:template>` stellt das Template dar.

```
<xsl:template match="Wein">
  <h3><xsl:value-of select="titel"/></h3>
</xsl:template>
```

Inhalt muss xml-konform sein
(siehe CDATA, Entity-Referenzen, ...)

5.4.2 Templates

- Beliebiger Zugriff auf Elemente/Attribute möglich
- Ausgabedokument => andere Struktur

5.4.3 Verarbeitung

- Preorder-Algorithmus
  - Zuerst Knoten selbst
  - Dann Child-Knoten von links nach rechts

  Verarbeitung
  = Ausführung des zugeh. Templates

- Abweichung möglich, durch:
  `<xsl:apply-templates select="`***`elementName`***`"/>`

5.4.4 XPath

- Zugriff auf Knoten und Attribute

Muster	Beschreibung
/	Wurzelknoten
.	Aktuelles Element
inhalt	Alle ***inhalt***-Elemente des Kontextknoten
//inhaltitem	Alle ***inhaltItem***-Elemente des Baumes
seminar/inhalt	Alle ***inhalt***-Elemente die Child-Element von ***seminar*** sind
buch/@isbn	Attribute ***isbn*** des Elementes ***buch***
inhalt/*	Die Child-Elemente des Element ***inhalt***
text()	Textknoten

5.4.5 xslt-Kontrollstrukturen

<xsl:for-each>

<xsl:if>

<xsl:choose>

Übungen

- http://www.digilife.be/quickreferences/quickrefs.htm