Just What is

And How Is It Changing E-Business ?

Klaus Fittges

Tom Donahue

Software AG, Inc.

Overview ....

- **Define Business and Electronic Business**
- **The history of XML**
  - SGML -> HTML -> XML
- **XML in more detail**
- **Where XML suits best**
  - Data Storage / Management
  - Electronic Publishing
  - Electronic Data Exchange
- **Summary**

Business - defined

- **Business = Exchange of goods**
- **Business = Exchange of Documents about goods**
- **e-Business = Exchange of Electronic Documents**
- **Differentiator :**
  - **manage** the documents received - **fast & efficient**
  - **information availability** - **immediate**
  - **reaction on information received** - **immediate**

e-Business =

Storing, **P**ublishing & **E**xchanging **E**lectronic **D**ocuments

Business World Today

- **Many different data formats**
  - Orders, Order Confirmations, Information, Product Information via Phone, Fax, E-Mail, Letters ..
  - Within enterprise and across enterprise boundaries
- **Waste of time due to media changes**
  - Complicated I/O procedures for data coming from different existing IT-Systems
- **Cost intensive special solutions for data integration**
  - Individual adaption of various applications & platforms for access to different types of data

Before: Big Ones Eat The Small Ones

	Traditional Business	Electronic Business
Time To Trade	Weeks	Minutes
Product Design	Company	Customer
Production	Pre Sales	Post Sales
Customer Relationship	Standard	One to One
Strategic Issue	Location	Service

Today: Fast Ones Eat the Slow Ones

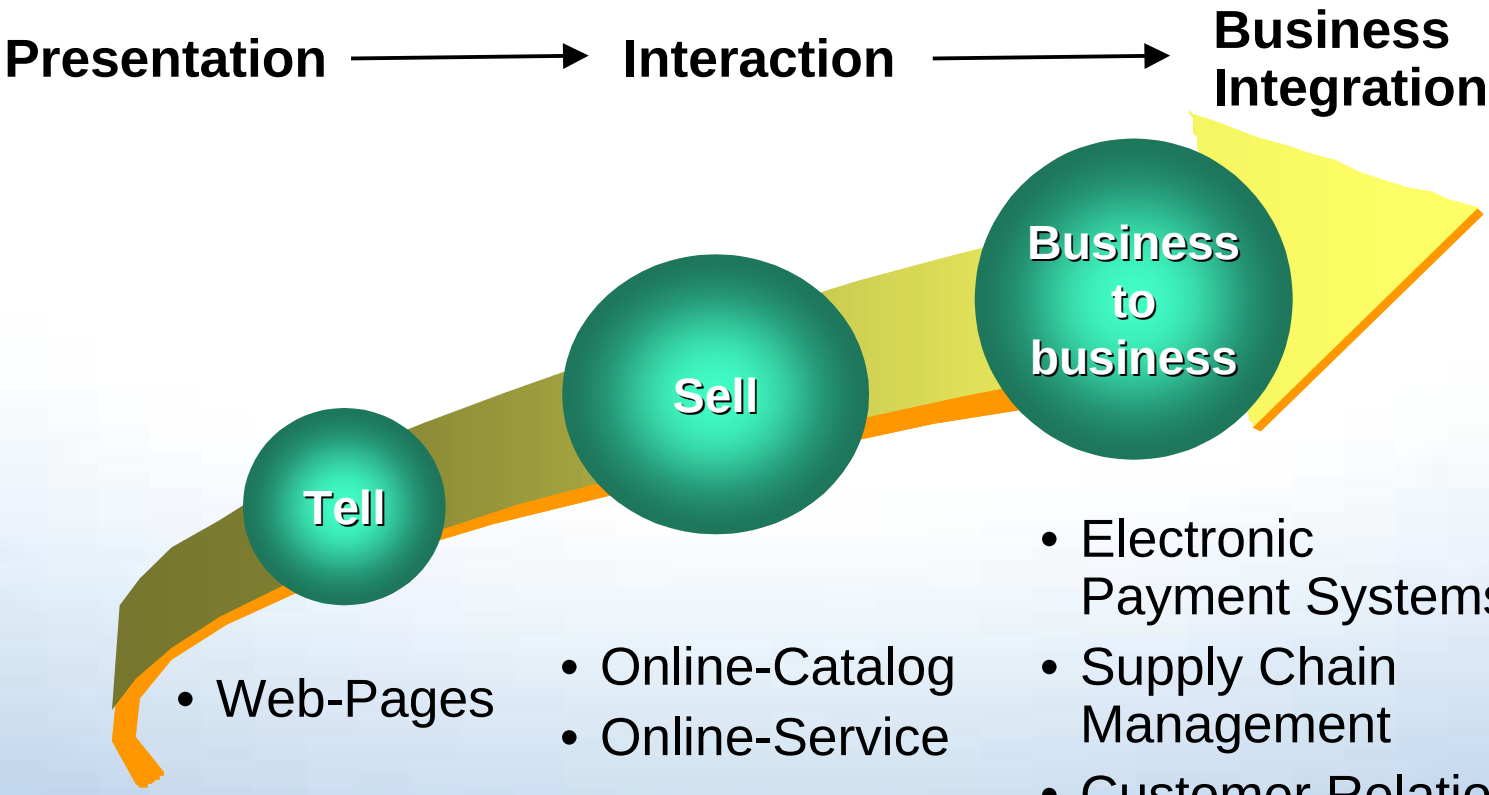
Economics of the E-Generation

	Traditional Business	Electronic Business
Time To Trade	Weeks	Minutes
Product Design	Company	Customer
Production	Pre Sales	Post Sales
Customer Relationship	Standard	One To One
Strategic Issue	Location	Service

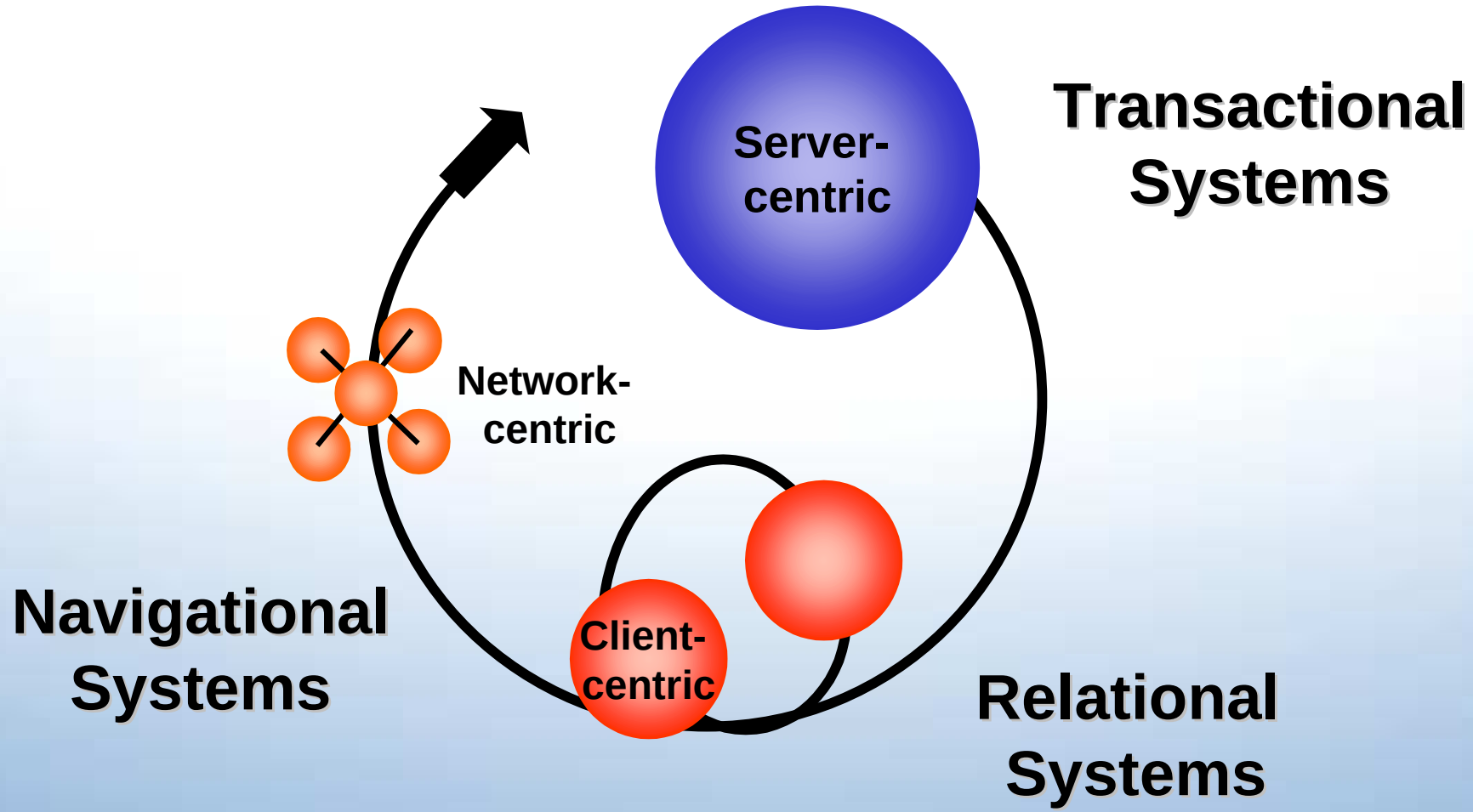
Business World Tomorrow

- **Automated business saves time and money**
  - Standardised architecture for access to data in the enterprise and across enterprise boundaries
  - Highest efficiency due to a minimum of media changes
- **Exchange of platform- and application independent data**
  - Simple integration of existing data via commonly accepted data format
  - Data easily readable and usable / self-describing
  - Long term data integrity and readability guaranteed
- **We are prepared to MEET THAT CHALLENGE**

From the Internet to E-Business



The future of computing architecture …
Navigational Systems



What is this?

The most successful business document in the history of the planet

Self Describing

Issued by trusted authority

Hard to forge

World-wide standard

Convertible

Easy to understand

The most successful business document in the history of the planet

Self Describing

Issued by trusted authority

Hard to forge

World-wide standard

Convertible

Easy to understand

Document Types of a Transaction Workflow

Marketing

- Web Page
- Price List
- Data Sheets

Bid

- Proposal
- Reference Documents

Sales

- Initial Proposal
- Contract
- Project Plan

Fulfillment

- Shipment receipt
- Restocking Order
- Delivery Receipt

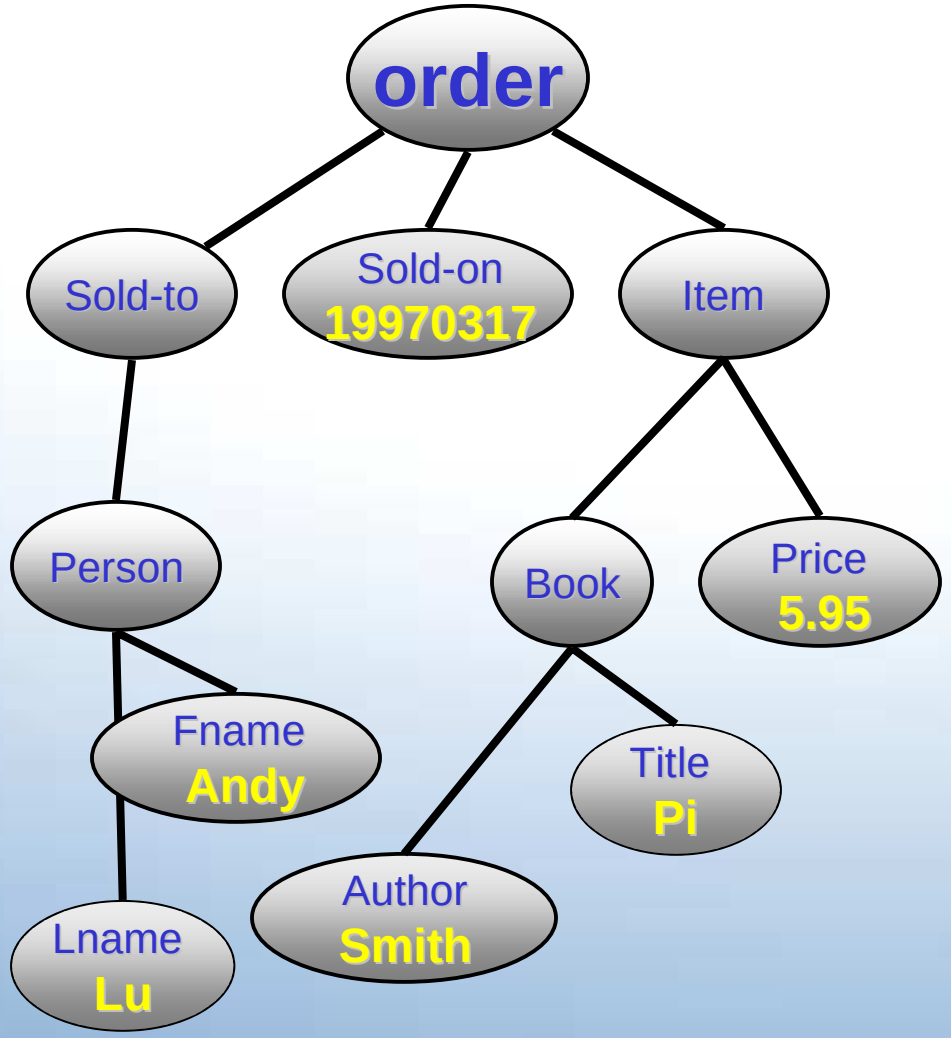
Customer Service

- Warranty Information
- Product Documentation

Maintenance

- Parts Lists
- Repair Manual

XML Documents are Hierarchical



Easy to query and navigate for modification using

XPath-based queries

- **DOM**
  - Document Object Model -

and

- **SAX**
  - Simple API for XML -

Interoperability - The Challenge

You are moving to Electronic Business...

Your data is distributed...

Your users are distributed…

But where is the common denominator?

Interoperability - The Solution

Extensible Markup Language

XML is the only viable option
to manage the diversity of data, applications
and devices
of Electronic Business Applications

XML 1.0 - A Meta-Language!

- **XML separates content from its presentation**
- **XML is simple** *(text based)*
  - easily readable
  - easily understandable
- **XML is extensible**
- **XML is verifiable**
  - validate for correctness / integrity
- **XML is open**
  - platform & application independent

```
<Customer>
    <Name>
        <LastName>
            Smith
        </LastName>
        <FirstName>
            Matthew
        </FirstName>
    </Name>
</Customer>
```

- **W3C-Standard**
- **Self-describing**
- **Hierarchically structured!**

XML Document Structure

```
<?xml version="1.0"?>
<weather-report>
   <date>August 15, 2000</date>
   <time>08:00</time>
   <area>
     <city>Darmstadt</city>
     <region>Hessen</region>
     <country>Germany</country>
   </area>
   <measurements>
      <skies>fine</skies>
      <temperature scale="C">25</temperature>
      <wind>
         <direction>SW</direction>
         <windspeed>6</windspeed>
      </wind>
   </measurements>
</weather-report>
```

Markup

Content

The Prolog


```
<?xml version="1.0"?>
<weather-report>
   <date>August 15, 2000</date>
   <time>08:00</time>
   <area>
     <city>Darmstadt</city>
     <region>Hessen</region>
     <country>Germany</country>
   </area>
   <measurements>
      <skies>fine</skies>
      <temperature scale="C">25</temperature>
      <wind>
          <direction>SW</direction>
          <windspeed>6</windspeed>
      </wind>
   </measurements>
</weather-report>
```

Encoding


```
<?xml version="1.0" encoding="UTF-8"?>
<weather-report>
   <date>August 15, 2000</date>
   <time>08:00</time>
   <area>
     <city>Darmstadt</city>
     <region>Hessen</region>
     <country>Germany</country>
   </area>
   <measurements>
      <skies>fine</skies>
      <temperature scale="C">25</temperature>
      <wind>
          <direction>SW</direction>
          <windspeed>6</windspeed>
      </wind>
   </measurements>
</weather-report>
```

Elements and Attributes

```
<?xml version="1.0"?>
<weather-report>
   <date>August 15, 2000</date>
   <time>08:00</time>
   <area>
     <city>Darmstadt</city>
     <region>Hessen</region>
     <country>Germany</country>
   </area>
   <measurements>
      <skies>fine</skies>
      <temperature scale="C">25</temperature>
      <wind>
         <direction>SW</direction>
         <windspeed>6</windspeed>
      </wind>
   </measurements>
</weather-report>
```

Element

Attribute

Element

Well-Formedness

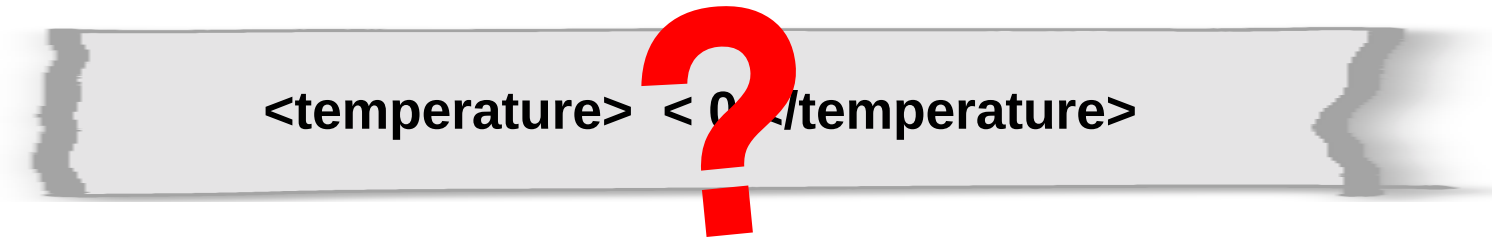
```
<?xml version="1.0"?>
<weather-report>
   <date>August 15, 2000</date>
   <time>08:00</time>
   <area>
     <city>Darmstadt</city>
     <region>Hessen</region>
     <country>Germany</country>
   </area>
   <measurements>
      <skies>fine</skies>
      <temperature scale="C">25</temperature>
      <wind>
          <direction>SW</direction>
          <windspeed>6</windspeed>
      </wind>
   </measurements>
</weather-report>
```

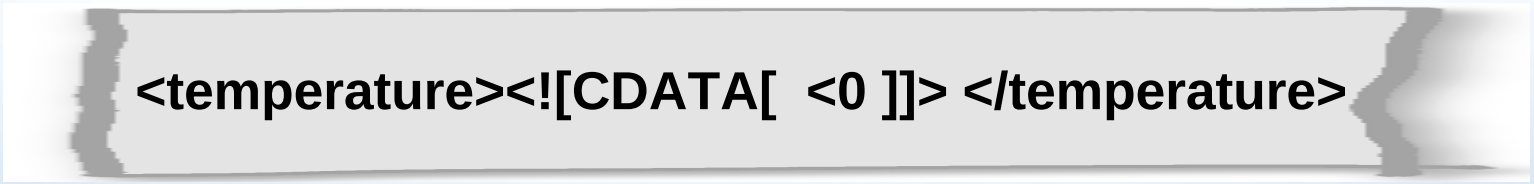
Empty Element

```
<temperature scale="C"></temperature>
```

```
<temperature scale="C"/>
```

CDATA Sections





Reserved Characters

<	<
>	>
&	&
‘	'
“	"

```
<temperature>  &lt; 0 </temperature>
```

Character Input

<example>
XML supports international character sets.
This example shows different notations for the number "1":
1 (ASCII),
١ (Devanagari),
१ (Arabic)
൧ (Malayalam)
</example>

Comments and Processing Instructions

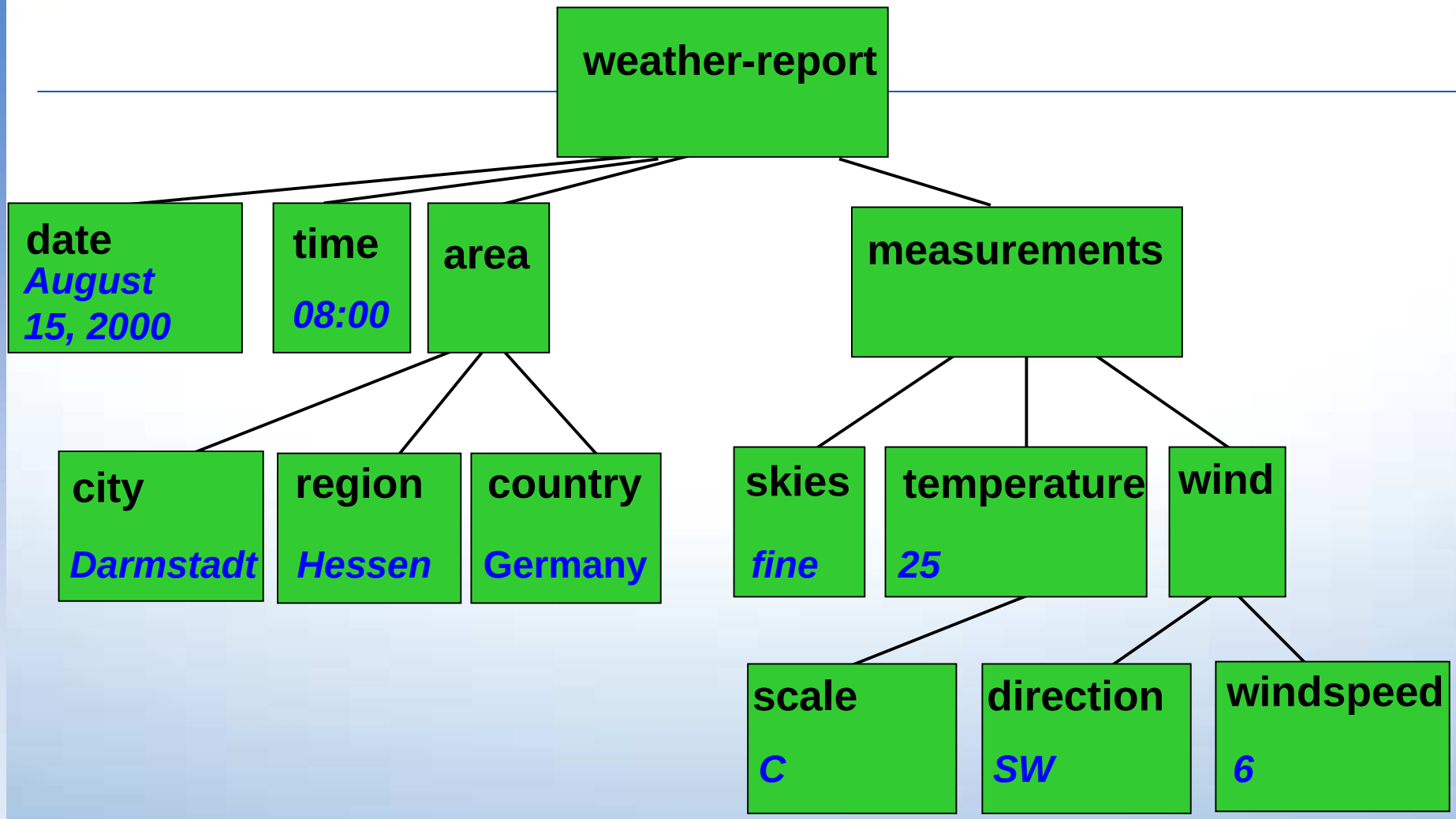
Comment

```
<!-- This is a comment -->
```

Processing Instruction

```
<?app1 fromhere="ignore"?>
```

The Logical Structure of XML Documents



Valid XML Documents

```
<?xml version="1.0"?>
<!DOCTYPE weather-report SYSTEM "weather.dtd">
<weather-report>
   <date>August 15, 2000</date>
   <time>08:00</time>
   <area>
     <city>Darmstadt</city>
     <region>Hessen</region>
     <country>Germany</country>
   </area>
   <measurements>
      <skies>fine</skies>
      <temperature scale="C">25</temperature>
      <wind>
          <direction>SW</direction>
          <windspeed>6</windspeed>
      </wind>
   </measurements>
</weather-report>
```

```
<!ELEMENT weather-report
                (date, time, area,measurements)>
<!ELEMENT  date              (#PCDATA)>
<!ELEMENT  time              (#PCDATA)>
<!ELEMENT  area ((city,region,country) | (x,y))>
<!ELEMENT city               (#PCDATA)>
<!ELEMENT region             (#PCDATA)>
<!ELEMENT country            (#PCDATA)>
<!ELEMENT x                  (#PCDATA)>
<!ELEMENT y                  (#PCDATA)>
<!ELEMENT measurements ((skies | temperature |
                         humidity | visibility | wind)+)>
<!ELEMENT skies              (#PCDATA)>
<!ELEMENT temperature        (#PCDATA)>
<!ELEMENT humidity           (#PCDATA)>
<!ELEMENT visibility         (#PCDATA)>
<!ELEMENT wind               (direction,windspeed)>
<!ELEMENT direction          (#PCDATA)>
<!ELEMENT windspeed          (#PCDATA)>
<!ATTLIST   temperature scale CDATA #IMPLIED>
```

weather.dtd

Document Type Declaration

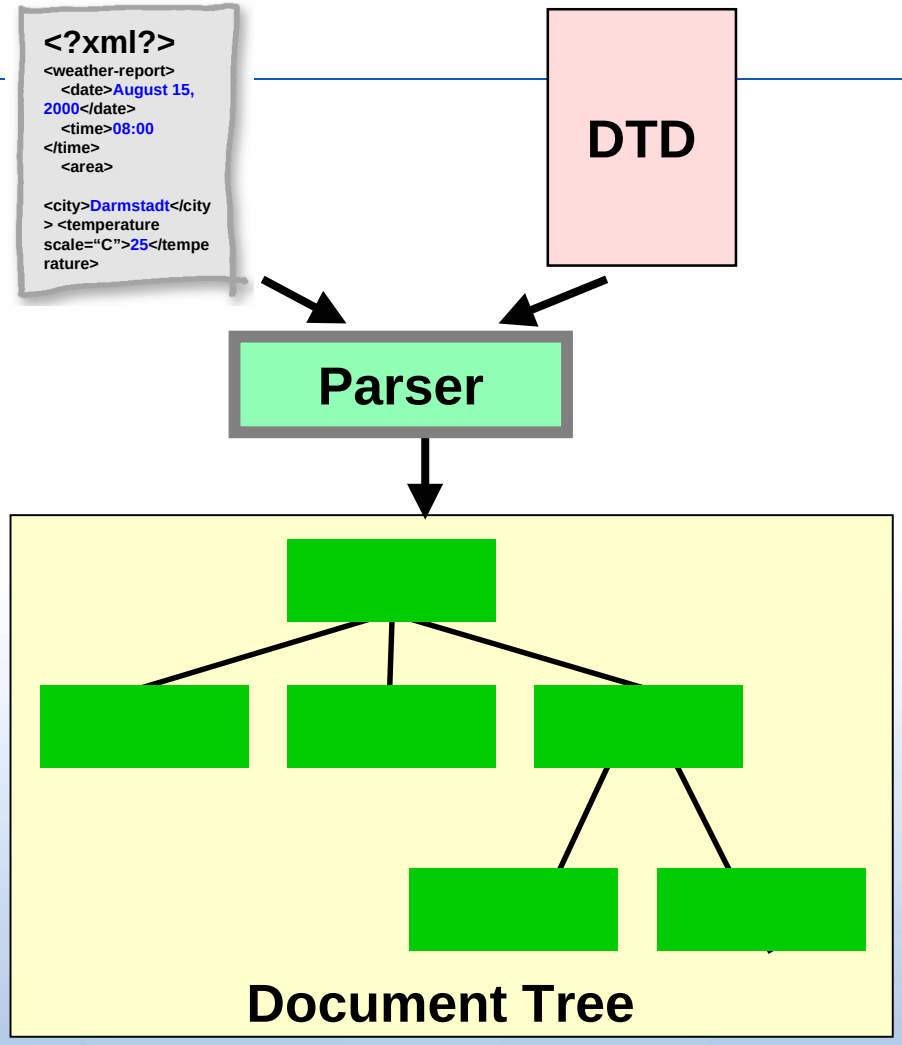
external

```
<?xml version="1.0"?>
<!DOCTYPE weather-report SYSTEM "weather.dtd">
```

Inline DTD

```
<?xml version="1.0"?>
<!DOCTYPE weather-report
[
<!ELEMENT weather-report
                (date,time,area,measurements)>
<!ELEMENT date          (#PCDATA)>    ....
]>
<weather-report>
  <date>   ....
```

Processing XML Documents



Element Type Declaration

<!ELEMENT	weather-report	(date, time, area, measurements)>
<!ELEMENT	date	(#PCDATA)>
<!ELEMENT	time	(#PCDATA)>
<!ELEMENT	area	((city,region,country) | (x,y))>
<!ELEMENT	city	(#PCDATA)>
<!ELEMENT	region	(#PCDATA)>
<!ELEMENT	country	(#PCDATA)>
<!ELEMENT	x	(#PCDATA)>
<!ELEMENT	y	(#PCDATA)>
<!ELEMENT	measurements	((skies | temperature | humidity | visibility | wind)+)>
<!ELEMENT	skies	(#PCDATA)>
<!ELEMENT	temperature	(#PCDATA)>
<!ELEMENT	humidity	(#PCDATA)>
<!ELEMENT	visibility	(#PCDATA)>
	...	

Element Types

Content Models

Element Content Models

EMPTY	**no content**
ANY	**no constraints on content**
|	**choice list**
,	**sequence**
Cardinality	
	exactly one
+	**one or more**
?	**zero or one**
*****	**zero or more**
()	**grouping**
(#PCDATA)	**characters**
(#PCDATA | ...) *	**characters or elements ("mixed content")**

Attribute Declaration

<!ATTLIST	temperature	scale	CDATA	#IMPLIED>
	Element Type	Attribute Name	Attribute Type	Default Value

<!ATTLIST	temperature	scale	CDATA	#REQUIRED>
<!ATTLIST	temperature	scale	(C| F)	#REQUIRED>
<!ATTLIST	temperature	scale	(C| F)	"C" >
<!ATTLIST	temperature	scale	CDATA	#FIXED “C”>

Attribute Type and Default Value

Attribute Type

CDATA	**string value**
ID	**attribute value unique per document**
IDREF(S)	**attribute value matches ID value**
(enumeration)	**restricts to values**

Default Value

#REQUIRED	value must be provided
#IMPLIED	no constraints
"string"	default value
#FIXED	default is only value

Element or Attribute?

```
<measurements>
    <skies>fine</skies>
    <temperature scale="C">25</temperature>
    ....
</measurements
```

```
<measurements>
    <skies>fine</skies>
    <temperature>25</temperature>
    <scale>C</scale>
    ....
</measurements
```

Entities and References

Entity Declaration in the DTD

```
<!DOCTYPE text
[
<!ENTITY sag      "Software AG">
<!ENTITY sagtext SYSTEM
      "http://www.softwareag.com/std.txt">
]>
```

Referencing in the document

```
<text>&sag; is proud to announce ... &sagtext;</text>
```

Structuring Documents

```
<!DOCTYPE  purchase-order
[
<!ENTITY Head                SYSTEM  "Headsection.xml">
<!ENTITY PositionsPC         SYSTEM  "Positions/PC1.xml">
<!ENTITY PositionsMonitor    SYSTEM  "http://monitors.de/m2.xml">
]>
<purchase-order>
   <customer-data> &Head; </customer-data>
   <order>
        <position> &PositionsPC; </position>
        <position> &PositionsMonitor; </position>
    </order>
</purchase-order>
```

Parameter Entities

```
<!DOCTYPE example
[
<!ENTITY % example-entity      "<!ELEMENT example (#PCDATA)>">
%example-entity;
]>
<example>
</example>
```

Entities and Notations

Integration of "Non-XML-data"

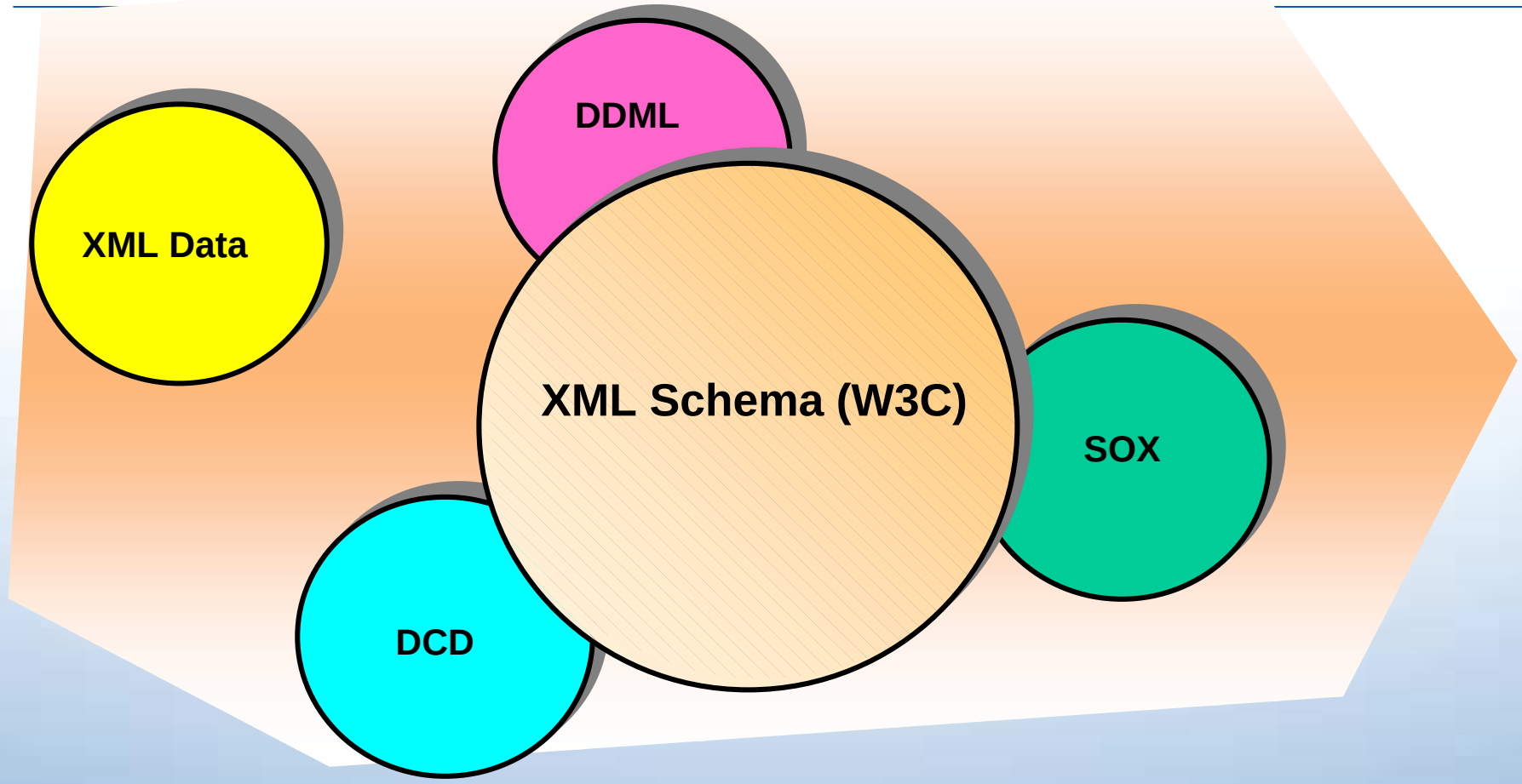
```
<!ELEMENT logo      EMPTY>
<!ATTLIST  logo     image  ENTITY  #REQUIRED>
<!NOTATION GIF      SYSTEM "gifmagic.exe">
<!ENTITY saglogo    SYSTEM "http://www.softwareag.com/logo1.gif"
                    NDATA  GIF>
```

```
<logo   image =  "saglogo"/>
```

Summary

- XML is a meta markup language standardized by the W3C.
- XML documents contain self-descriptive structured data.
- An XML document is *well-formed* if it meets a few formal criteria.
- An XML document is *valid* if it meets the structure rules of a Document Type Definition (DTD).
- The DTD syntax contains grammatical rules for the definition of elements, attributes, and entities.

XML Schema
Existing Schema Languages



Types of XML Documents

Data-oriented

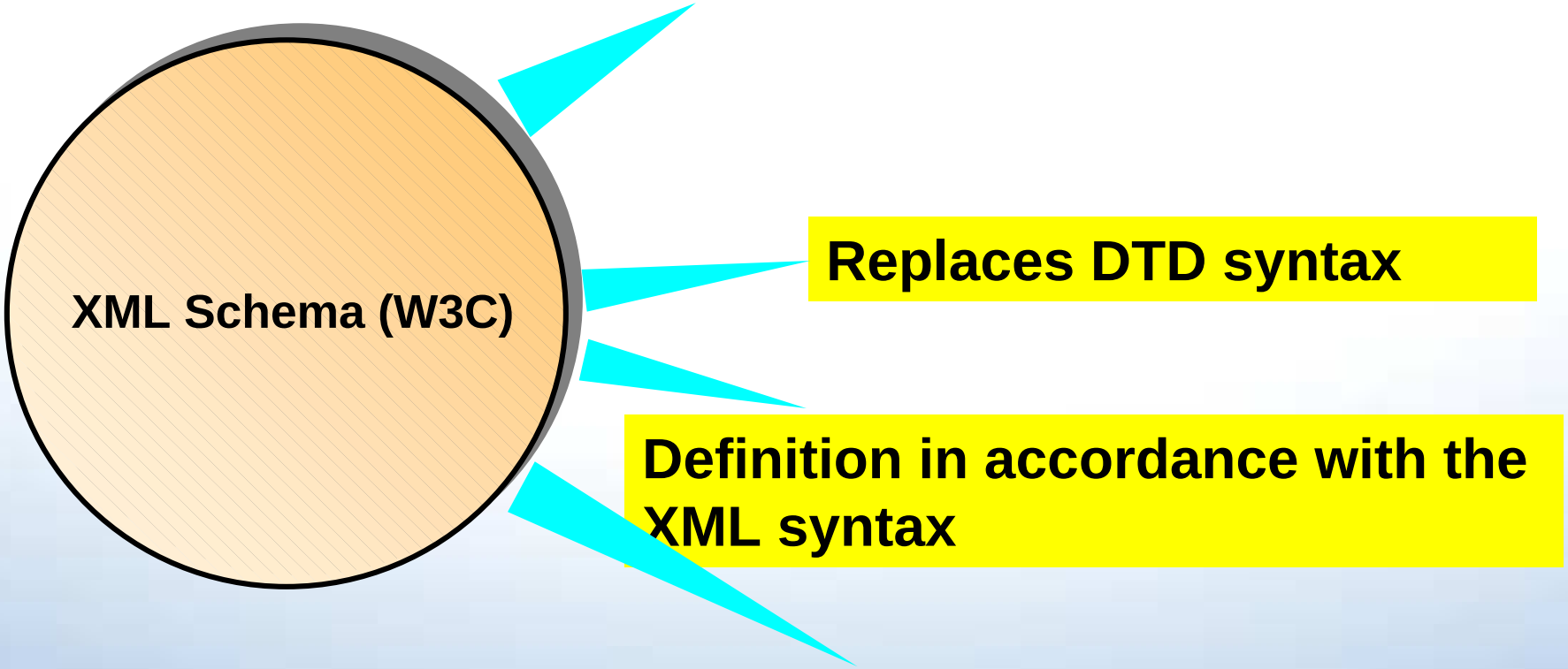
```
<invoice>
  <orderDate>19990121</orderDate>
  <shipDate>19990125</shipDate>
  <billingAddress>
    <name>Ashok Malhotra</name>
    <street>123 IBM Ave.</street>
    <city>Hawthorne</city>
    <state>NY</state>
    <zip>10532-0000</zip>
  </billingAddress>
  <voice>555-1234</voice>
  <fax>555-4321</fax>
</invoice>
```

Document-oriented

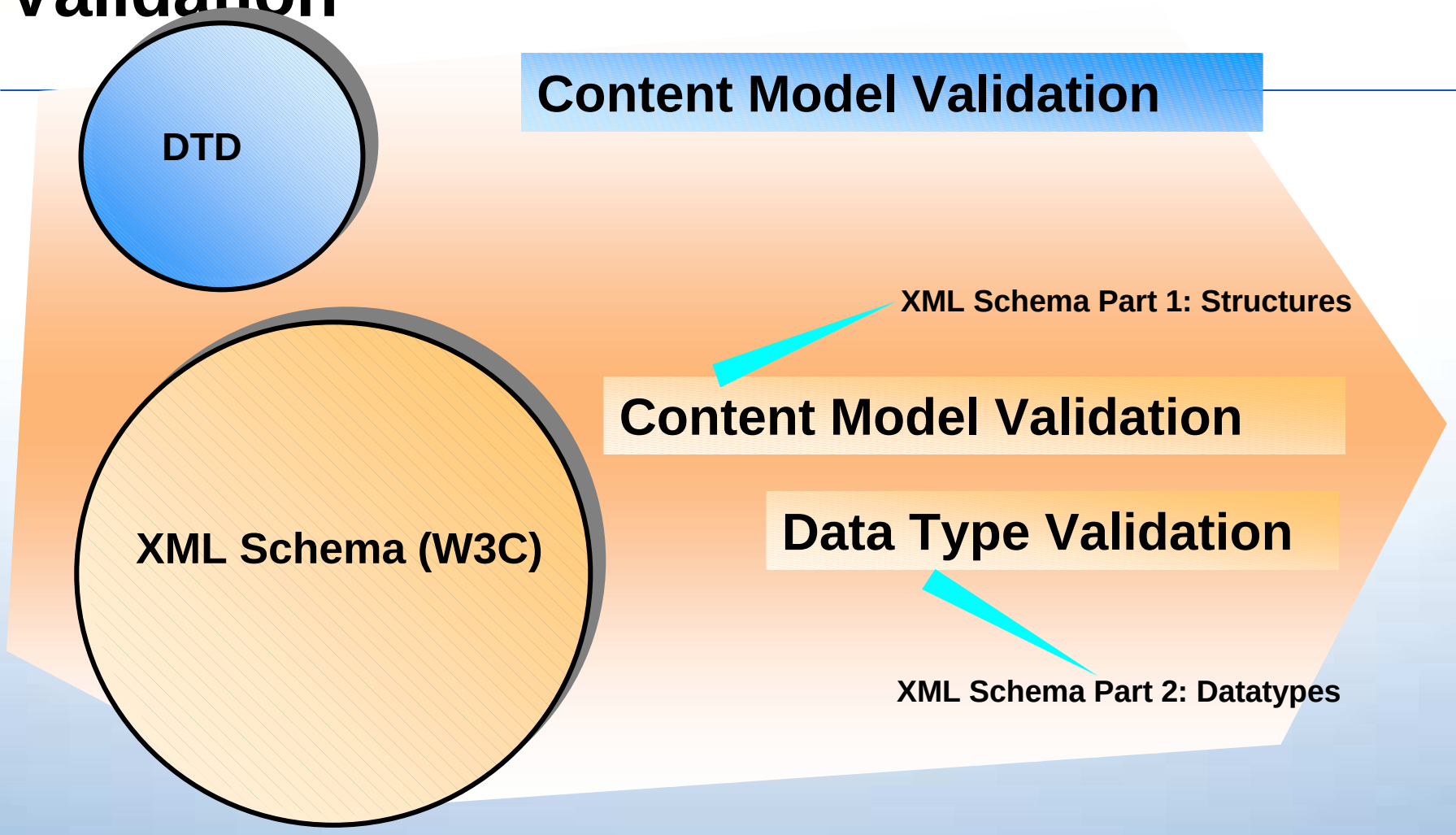
```
<memo importance="high"
    date="19990323">
  <from>Paul V. Biron</from>
  <to>Ashok Malhotra</to>
  <subject>Latest draft</subject>
  <body>
    We need to discuss the latest
    draft <emph>immediately</emph>.
    Either email me at <email>
    mailto:paul.v.biron@kp.org</email>
    or call <phone>555-9876</phone>
  </body>
</memo>
```

XML Schema

Based on XML syntax, tools and technology



Validation


DTD
XML Schema (W3C)
Content Model Validation
XML Schema Part 1: Structures
Content Model Validation
Data Type Validation
XML Schema Part 2: Datatypes

Features

- Complex data types
- User-defined data types (archetypes)
- Attribute grouping
- Refinement of content models, inheritance
- Namespace support

XML Schema vs. DTD

DTD

```
<!ELEMENT    address    (company?, name, street, city, state, zip, phone+)>
<!ELEMENT    company    (#PCDATA)>
<!ELEMENT    name       (#PCDATA)>
<!ELEMENT    street     (#PCDATA)>
<!ELEMENT    city       (#PCDATA)>
<!ELEMENT    state      (#PCDATA)>
<!ELEMENT    zip        (#PCDATA)>
<!ELEMENT    phone      (#PCDATA)>
```

XML Schema

```
<elementType name="address" >
  <sequence>
    <elementType  name="company"  minOccur="0"  maxOccur ="1"/>
    <elementType  name="name"     minOccur="1"  maxOccur ="1"/>
    <elementType  name="street"   minOccur="1"  maxOccur ="1"/>
    <elementType  name="city"     minOccur="1"  maxOccur ="1"/>
    <elementType  name="state"    minOccur="1"  maxOccur ="1"/>
    <elementType  name="zip"      minOccur="1"  maxOccur ="1"/>
    <elementType  name="phone"    minOccur="1"  maxOccur ="5"/>
  <sequence>
</elementType>
```

Data Type Definitions

```
<elementType name="zip">
   <datatypeRef name="zipCode"/>
</elementType>
```

```
<datatype name="zipCode">
   <basetype name="string"/>
   <lexicalRepresentation>
      <lexical>99999</lexical>
      <lexical>99999-9999</lexical>
   </lexicalRepresentation>
</datatype>
```

Data Type Definitions

```
<datatype name="ibmhex32">
  <basetype name="real"/>
  <minAbsoluteValue>
    5.2e-85
  </minAbsoluteValue>
  <maxAbsoluteValue>
    7.2e75
  </maxAbsoluteValue>
</datatype>
```

```
<datatype name="currency">
  <basetype name="decimal"/>
</datatype>
```

```
<datatype name="holidays">
  <basetype name="date"/>
  <enumeration>
    <literal>
      --0101    <!-- New Year -->
    </literal>
     <literal>
      --0501    <!-- May 1 -->
    </literal>
    <literal>
      --0704    <!-- July 4 -->
    </literal>
    <literal>
      --1225    <!-- Christmas -->
    </literal>
  </enumeration>
</datatype>
```

Summary

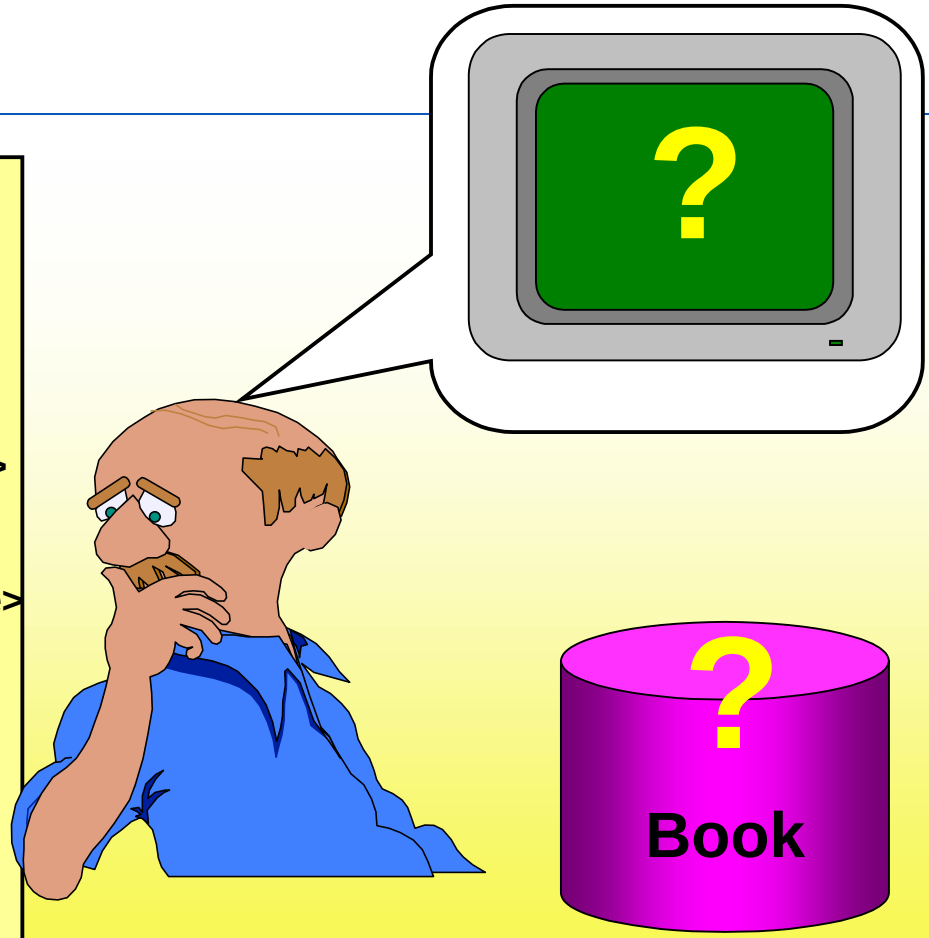
- **Schema languages apply and enhance the concepts of DTDs to a modern, consistent level.**
- **XML Schema is the W3C specification which, in the long run, will replace the DTD syntax.**
- **XML Schema can construct very complex *content models.* Moreover, it supports a complex datatyping concept.**

Processing XML - Overview

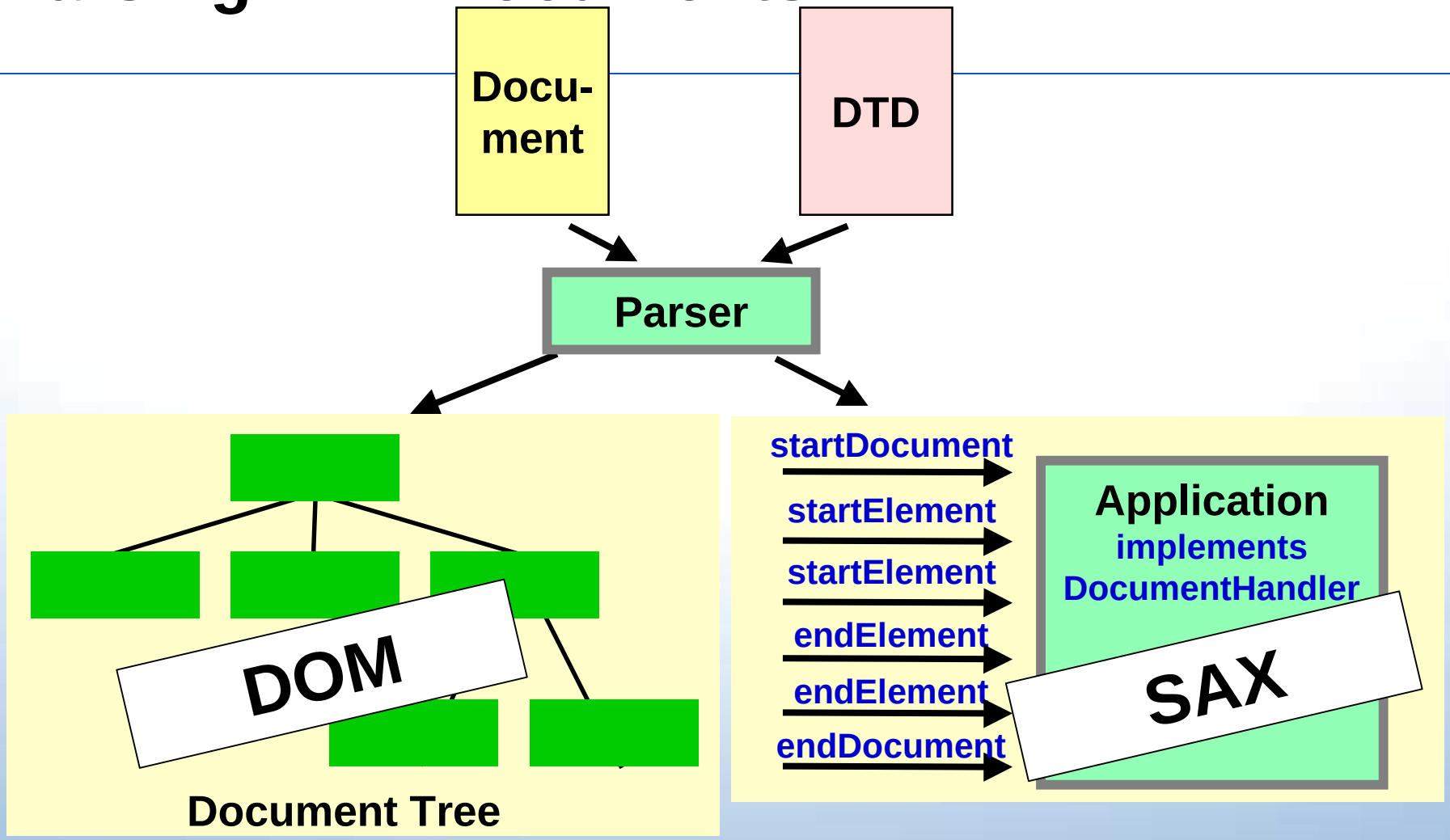
- **Parsing XML documents**
  - Document Object Model (DOM)
  - Simple API for XML (SAX)
- **Class generation**

Processing XML - What's the Problem?

```
<?xml version="1.0"?>
<books>
   <book>
        <title>The XML Handbook</title>
        <author>Goldfarb</author>
        <author>Prescod</author>
        <publisher>Prentice Hall</publisher>
        <pages>688</pages>
        <isbn>0130811521</isbn>
        <price currency="USD">44.95</price>
   </book>
   <book>
        <title>XML Design</title>
        <author>Spencer</author>
        <publisher>Wrox Press</publisher>
        ...
   </book>
</books>
```



Parsing XML Documents



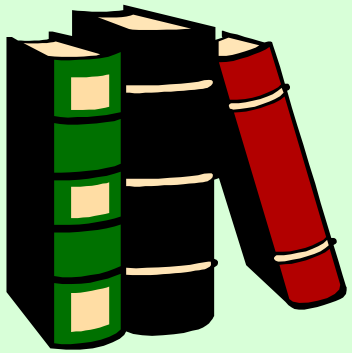
Class Generation

DTD
'books'

Generation

```
<?xml version="1.0"?>
<books>
   <book>
      ...
   </book>
   <book>
      ...
   </book>
</books>
```

Processing



Parser

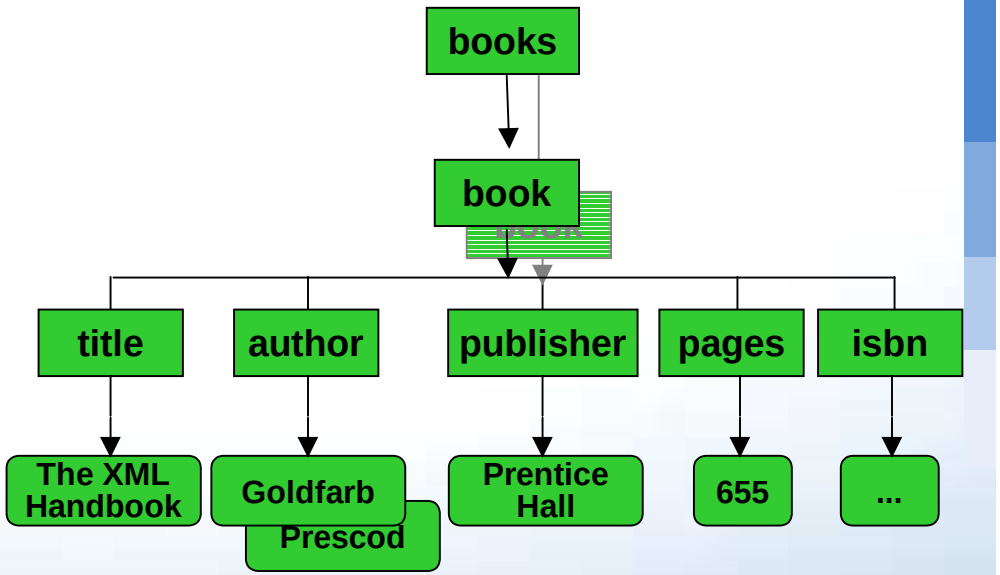
- Project X (Sun Microsystems)
- Ælfred (Microstar Software)
- XML4J (IBM)
- Lark (Tim Bray)
- MSXML (Microsoft)
- XJ (Data Channel)
- Xerces (Apache)
- ...

The Document Object Model

XML Document

```
<?xml version="1.0"?>
<books>
   <book>
         <title>The XML Handbook</title>
         <author>Goldfarb</author>
         <author>Prescod</author>
         <publisher>Prentice Hall</publisher>
         <pages>688</pages>
         <isbn>0130811521</isbn>
         <price currency="USD">44.95</price>
   </book>
   <book>
         <title>XML Design</title>
         <author>Spencer</author>
         <publisher>Wrox Press</publisher>
         ...
   </book>
</books>
```

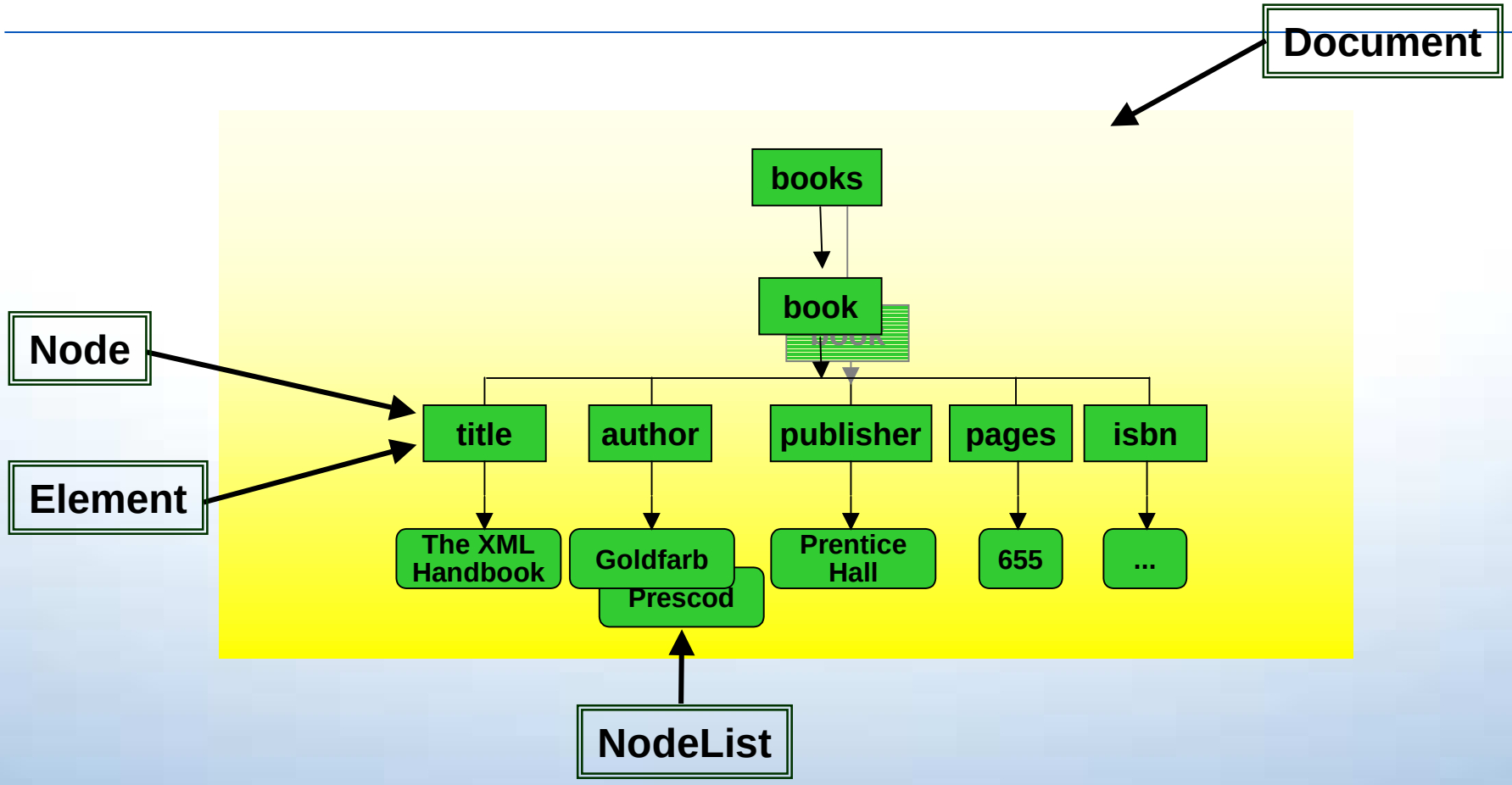
Structure



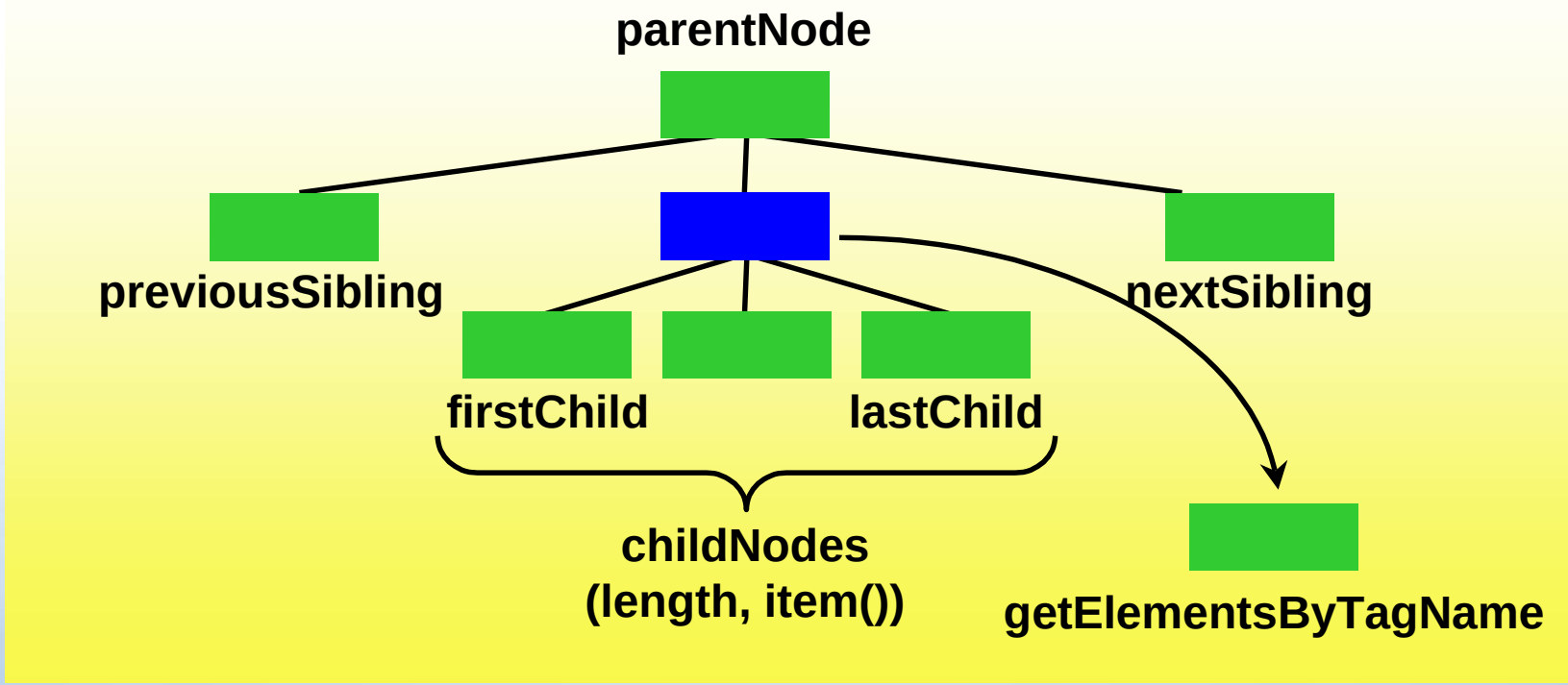
The Document Object Model

- Provides a standard interface for access to and manipulation of XML structures.
- Represents documents in the form of a hierarchy of nodes.
- Is platform- and programming-language-neutral
- Is a recommendation of the W3C (October 1, 1998)
- Is implemented by many parsers

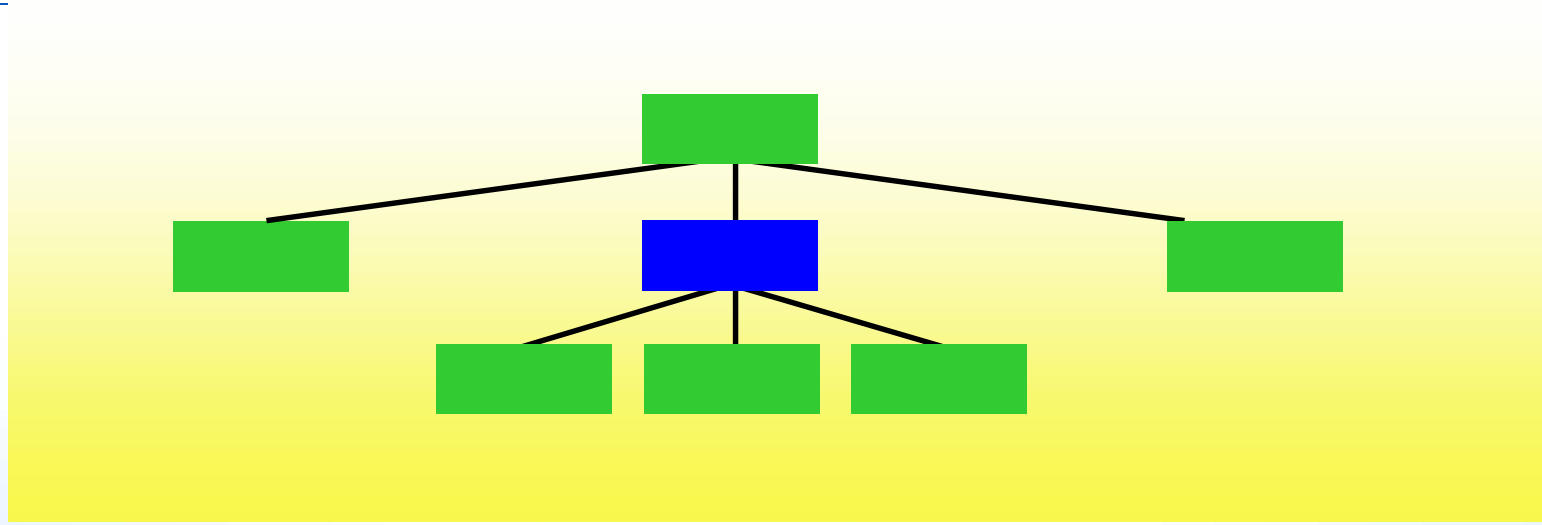
DOM - Structure Model



DOM Methods for Navigation



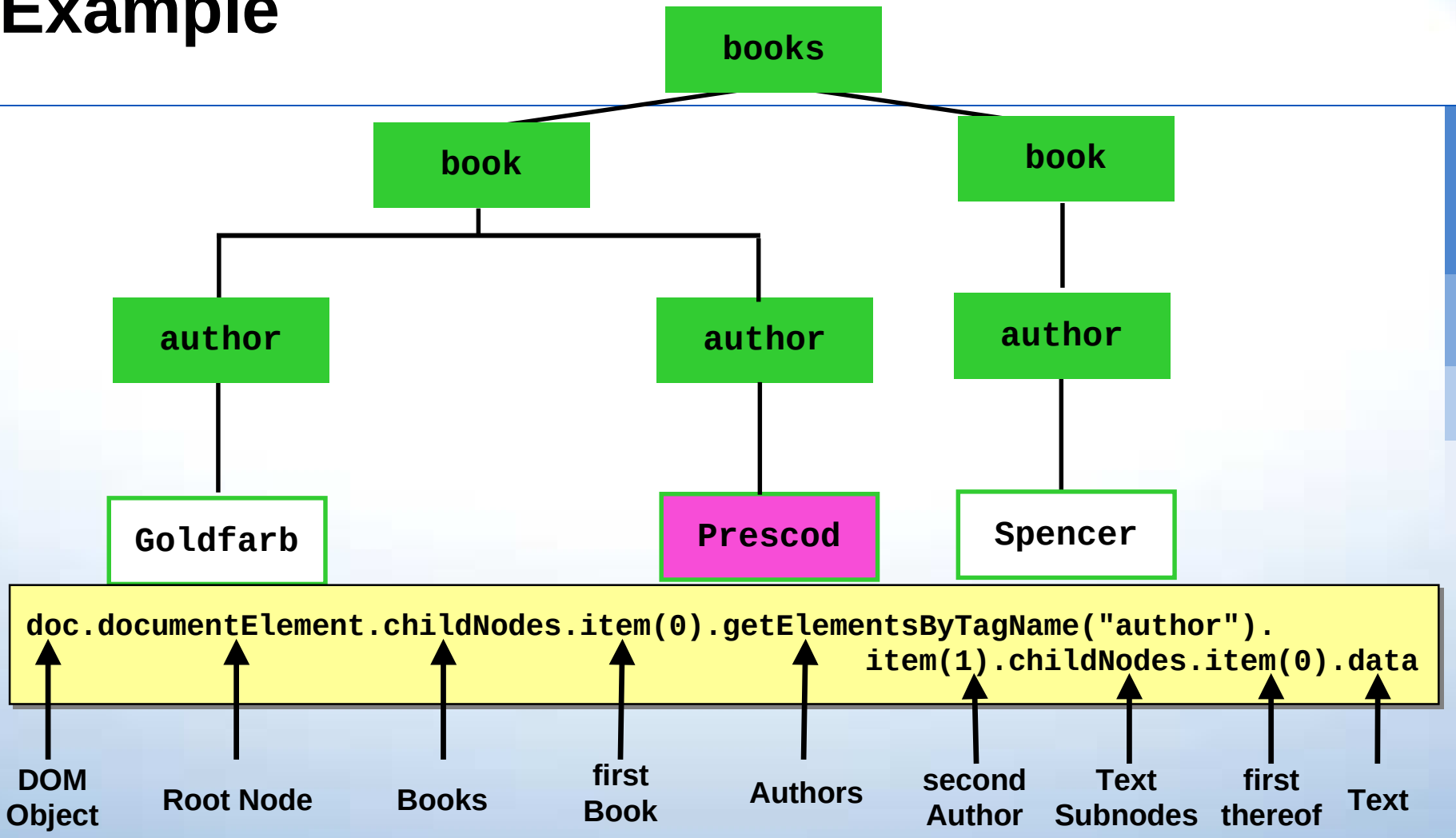
DOM Methods for Manipulation



createElement
createAttribute
createTextNode

appendChild
insertBefore
replaceChild
removeChild

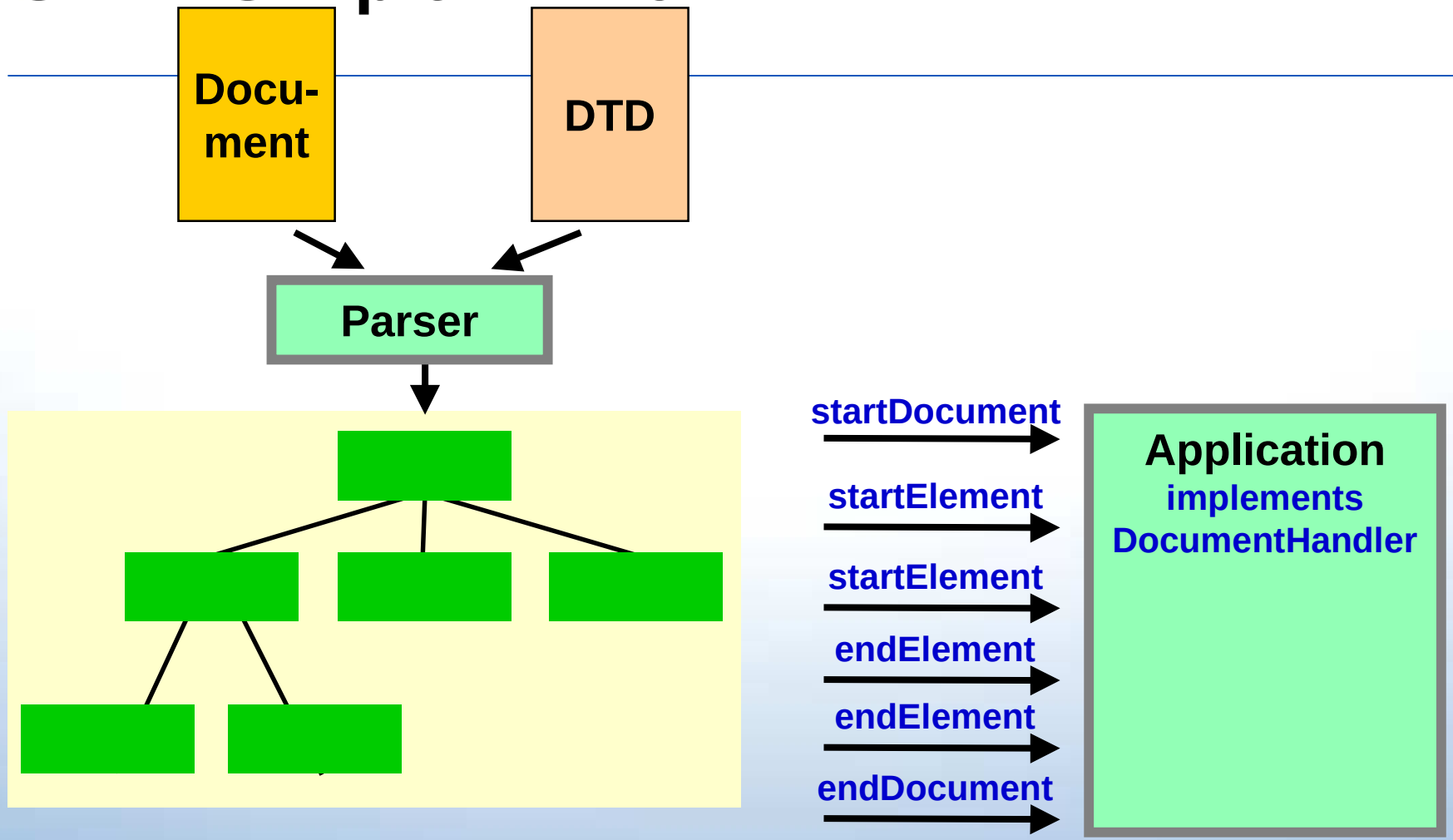
Example


books
book
book
author
author
author
Goldfarb
Prescod
Spencer
doc.documentElement.childNodes.item(0).getElementsByTagName("author").
item(1).childNodes.item(0).data
DOM Object
Root Node
Books
first Book
Authors
second Author
Text Subnodes
first thereof
Text

Script

```
<HTML>
<HEAD><TITLE>DOM Example</TITLE></HEAD>
<BODY>
<H1>DOM Example</H1>
<SCRIPT LANGUAGE="JavaScript">
   var doc, root, book1, authors, author2;
   doc = new ActiveXObject("Microsoft.XMLDOM");
   doc.async = false;
   doc.load("books.xml");
   if (doc.parseError != 0)
      alert(doc.parseError.reason);
   else {
   root = doc.documentElement;
   document.write("Name of Root node: " + root.nodeName + "<BR>");
   document.write("Type of Root node: " + root.nodeType + "<BR>");
   book1 = root.childNodes.item(0);
   authors = book1.getElementsByTagName("author");
   document.write("Number of authors: " + authors.length + "<BR>");
   author2 = authors.item(1);
   document.write("Name of second author: " + author2.childNodes.item(0).data);}
</SCRIPT>
</BODY></HTML>
```

SAX - Simple API for XML



SAX - Simple API for XML

- Event-driven parsing model
- "Don't call the DOM, the parser calls you."
- Developed by the members of the XML-DEV Mailing List
- Released on May 11, 1998
- Supported by many parsers ...
- ... but Ælfred is the saxon king.

Procedure

- **DOM**
  - Creating a parser instance
  - Parsing the whole document
  - Processing the DOM tree
- **SAX**
  - Creating a parser instance
  - Registrating event handlers with the parser
  - Parser calls the event handler during parsing

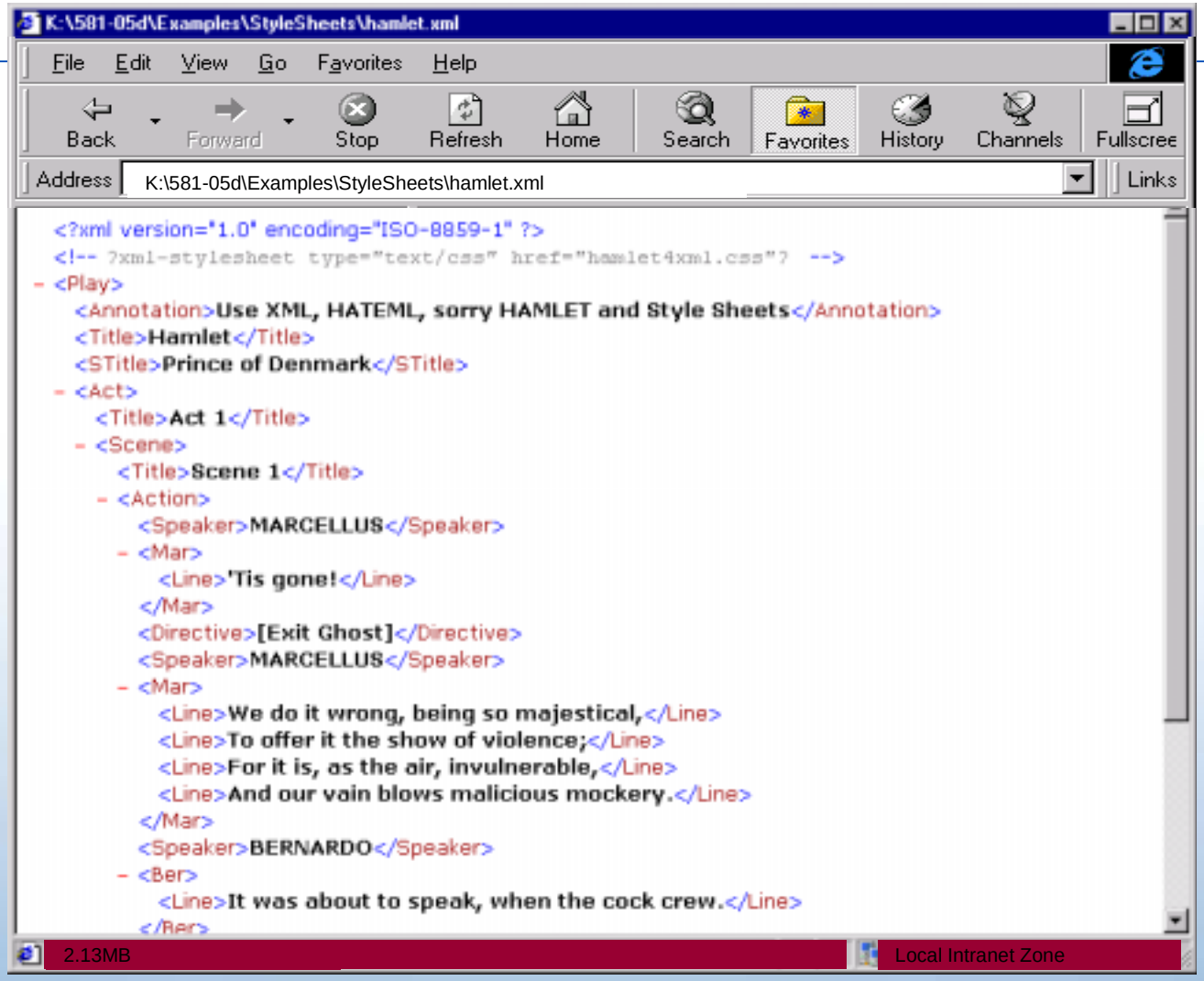
Summary

- To avoid expensive text processing, applications use an XML parser that creates a DOM tree of a document.
- The DOM provides a standardized API to access the content of documents and to manipulate them.
- Alternatively or additionally, applications can work event-based using the SAX interface, which is provided by many parsers.

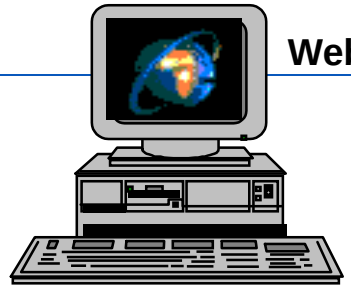
Presenting XML - Overview

- **XML and Web Browser**
  - Displaying XML
  - XML Data Islands
- **Style Sheets**
  - Style Sheets in HTML
  - Cascading Style Sheets
  - XSL - the Extensible Style Language

Displaying XML

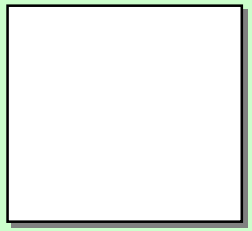


Style Sheets in HTML

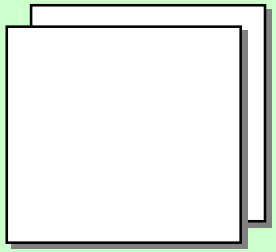


Web Browser

HTML Page



HTML Page with Styles



HTML Page with Style Sheet

HTML Page

```
<HTML>
<head><title>Hamlet in pure HTML</title></head>
<BODY>
<HR COLOR="RED">
<H1 ALIGN="CENTER" COLOR="RED">Hamlet</H1>
<H2 ALIGN="CENTER" COLOR="RED">Prince of Denmark</H2>
<HR COLOR="RED">
<H3>Act 1, Scene 1</H3>
<B>MARCELLUS</B><BR><BR>
<FONT FACE="Times New Roman">'Tis gone!</FONT><BR><BR>
<I>[Exit Ghost]</I><BR><BR>
<FONT FACE="Times New Roman">
We do it wrong, being so majestical,<BR>
To offer it the show of violence;<BR>
For it is, as the air, invulnerable,<BR>
And our vain blows malicious mockery.</FONT>
```

HTML Page with Styles

```
<HTML><head><title>Hamlet with Styles</title>
<STYLE TYPE="text/css">
  H1 {text-align : center; color : red}
  P.Marcellus  {font-family : Times New Roman}
  P.Bernardo   {font-family : Arial}
  .Directive   {font-size: -1; font-style : italic}
</STYLE></head>
<BODY><HR COLOR="RED">
<H1>Hamlet</H1><H2>Prince of Denmark</H2>
<HR COLOR="RED">
<H3>Act 1, Scene 1</H3>
<B>MARCELLUS</B><BR>
<P CLASS="Marcellus">
'Tis gone!<BR><BR>
<P CLASS="Directive">[Exit Ghost]</P>
We do it wrong, being so majestical,<BR>
```

HTML Page with Style Sheet

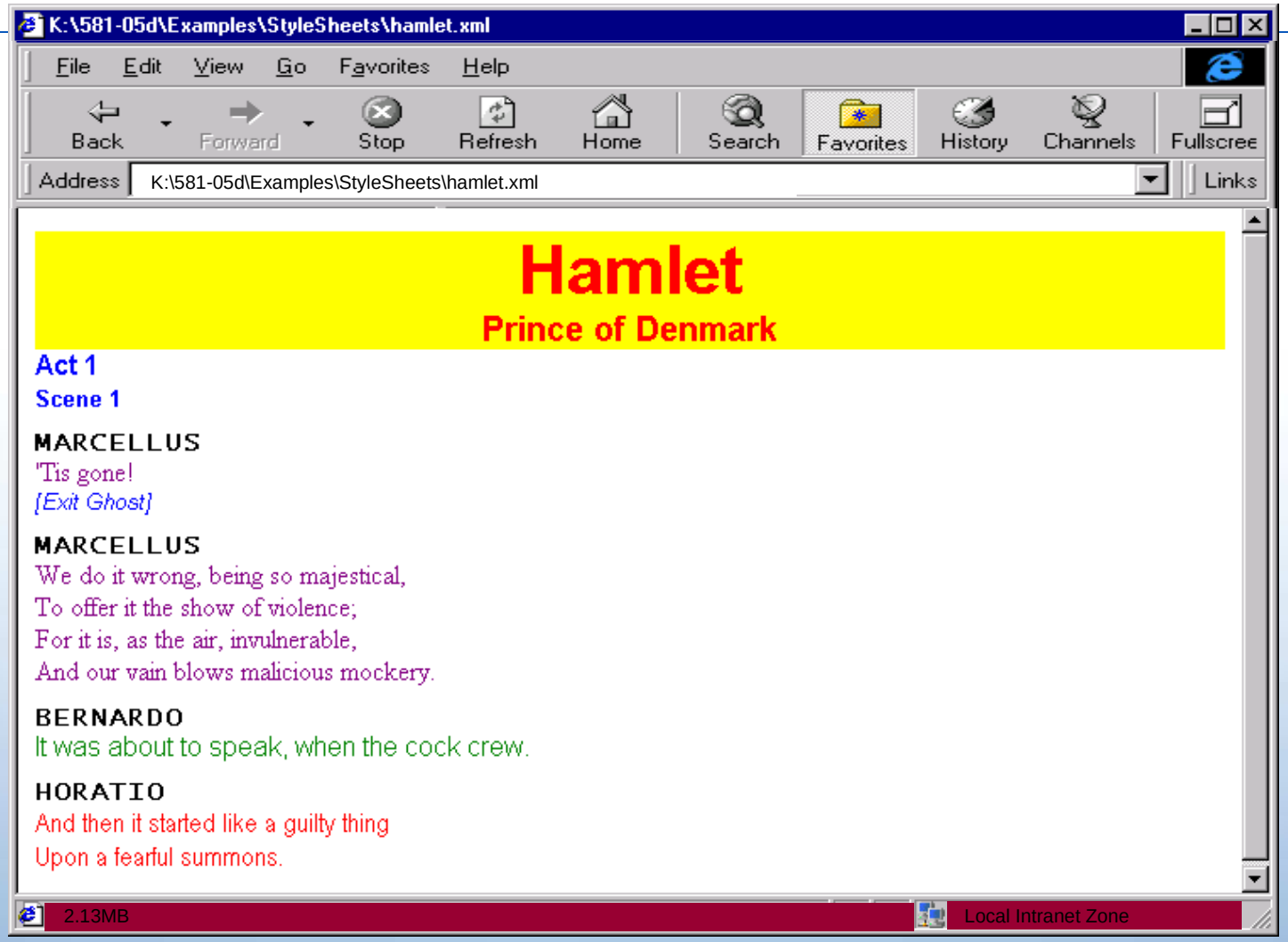
```
<HTML>
<LINK REL=STYLESHEET TYPE="text/css"
      HREF="hamlet.css">

<H1>Hamlet</H1><H2>Prince of Denmark</H2>
<HR COLOR="RED">
<H3>Act 1, Scene 1</H3>
<B>MARCELLUS</B><BR>
<P CLASS="Marcellus">
'Tis gone!<BR><BR>
<P CLASS="Directive">[Exit Ghost]</P>
We do it wrong, being so majestical,<BR>
To offer it the show of violence;<BR>
For it is, as the air, invulnerable,<BR>
And our vain blows malicious mockery.
```

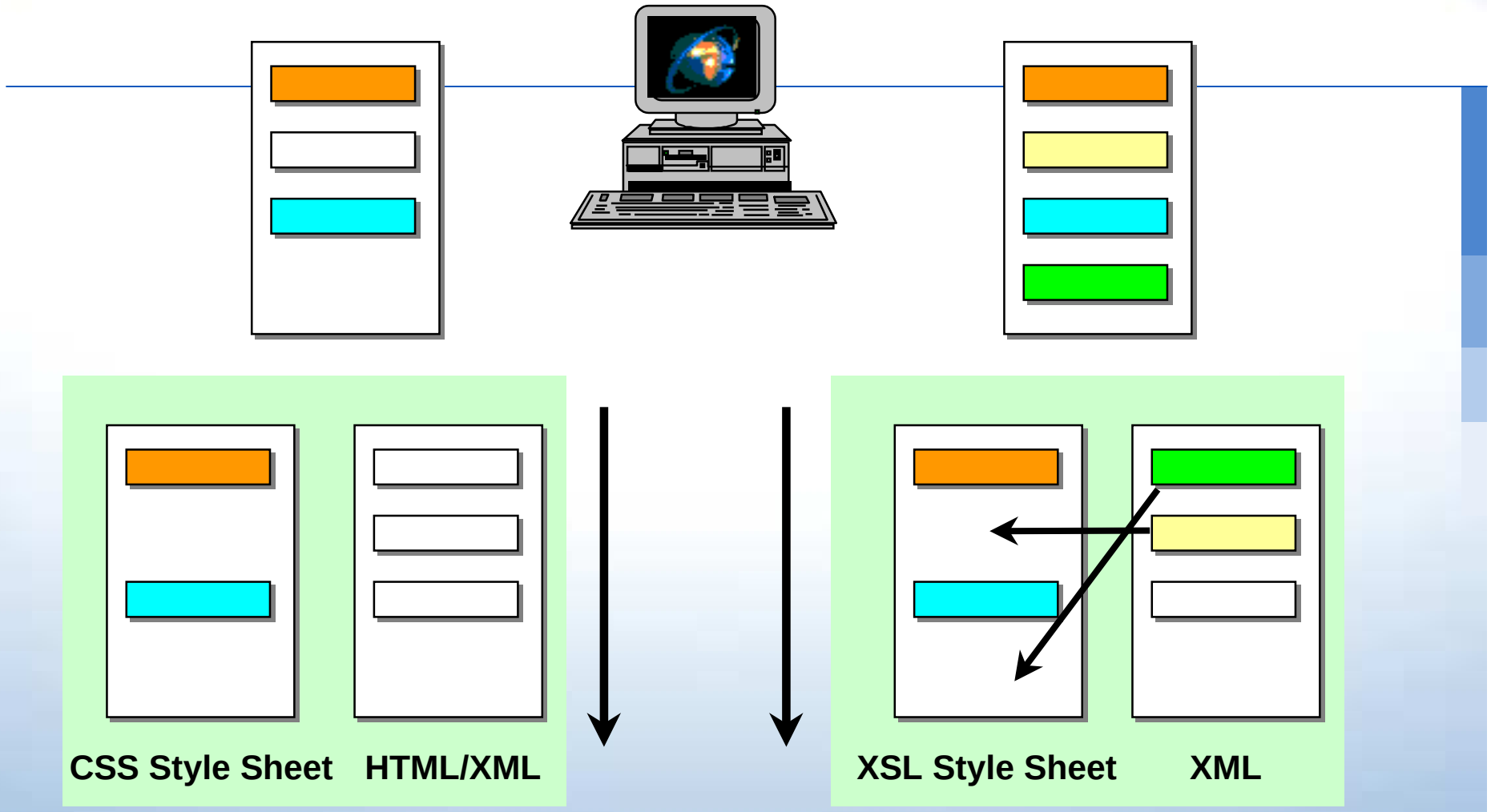
Functionality with Style Sheets (CSS2)

Display	inline, block, list-item, none
Text	style, family, weight, size, ...
Boxes	margin, border, width, height, ...
Media types	aural, braille, print, screen
Paged media	portrait, landscape, left, right, break
Aural styles	voice-family, stress, azimuth, elevation, volume
Internationalization	
Counters	
...	

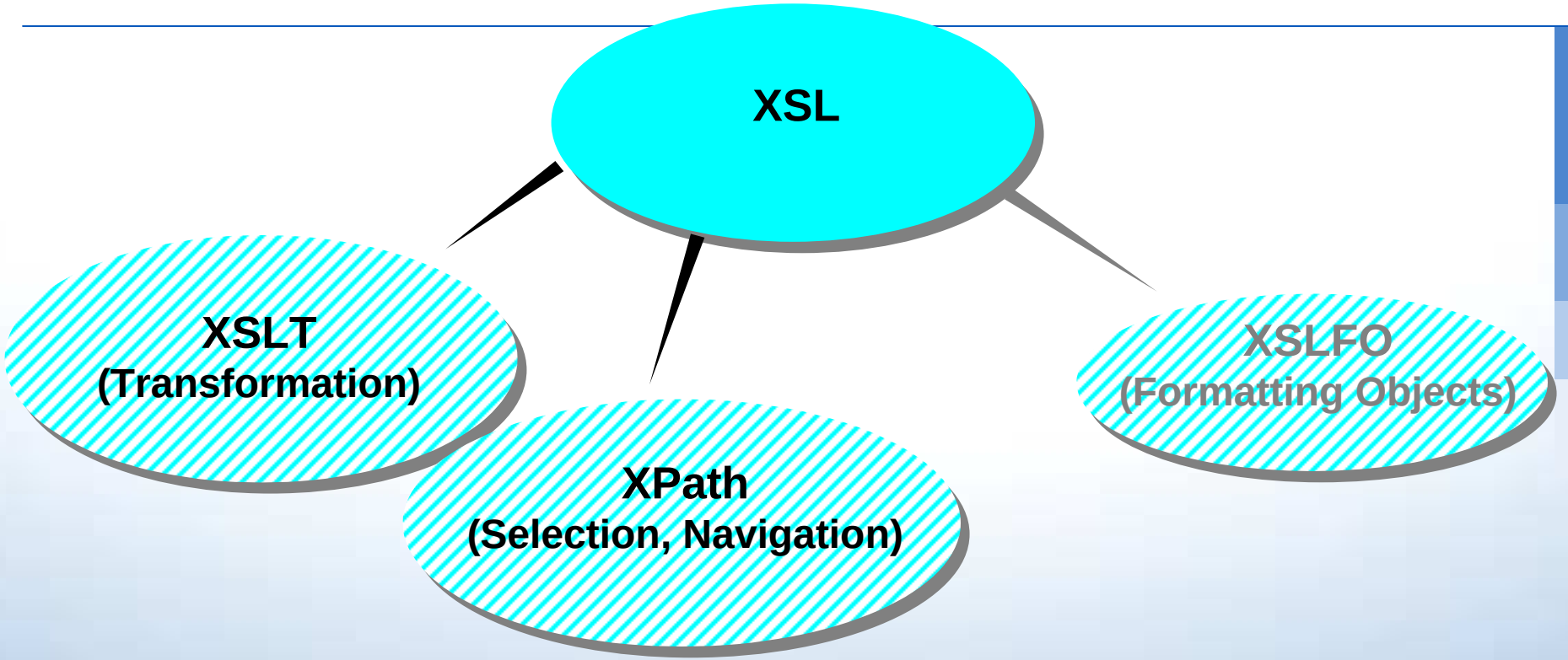
Display


K:\581-05d\Examples\StyleSheets\hamlet.xml
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscree
Address K:\581-05d\Examples\StyleSheets\hamlet.xml
Links
Hamlet
Prince of Denmark
Act 1
Scene 1
MARCELLUS
'Tis gone!
[Exit Ghost]
MARCELLUS
We do it wrong, being so majestical,
To offer it the show of violence;
For it is, as the air, invulnerable,
And our vain blows malicious mockery.
BERNARDO
It was about to speak, when the cock crew.
HORATIO
And then it started like a guilty thing
Upon a fearful summons.
2.13MB
Local Intranet Zone

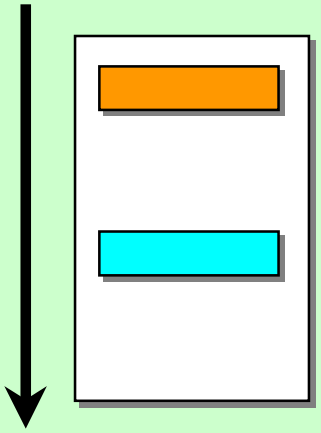
CSS Style Sheet vs. Style Language



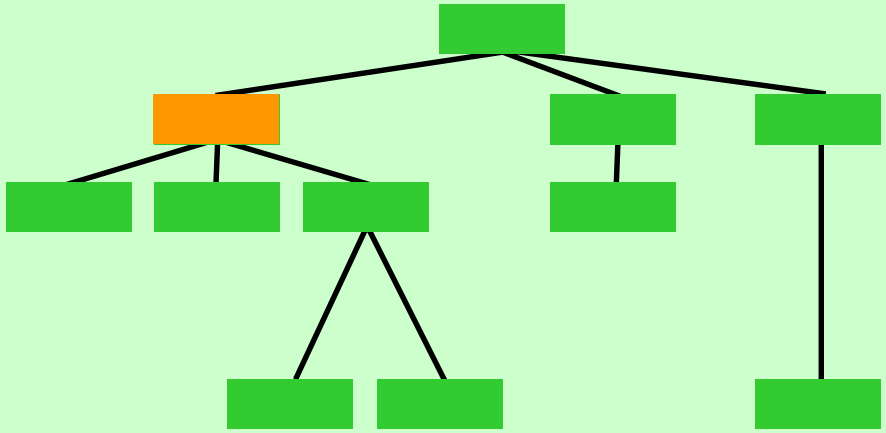
XSL Extensible Style Sheet Language



Anatomy of an XSL Style Sheet



XSL Style Sheet



XML Tree

```
<xsl:template match="pattern">

    HTML Tags
    more Style Sheet processing

</xsl:template>
```

1. Pattern matching
2. Formatting

Pattern Matching (XPath)

```
<xsl:template match="pattern">

    HTML Tags
    further Style Sheet processing

</xsl:template>
```

XPath

XPath Expression	Meaning
/	Child Operator
//	Child Operator without hierarchical restriction; search for specified element on all elements
.	Current context
*	Wildcard; all elements
@	Attribute
[ ]	Using a Filter Pattern

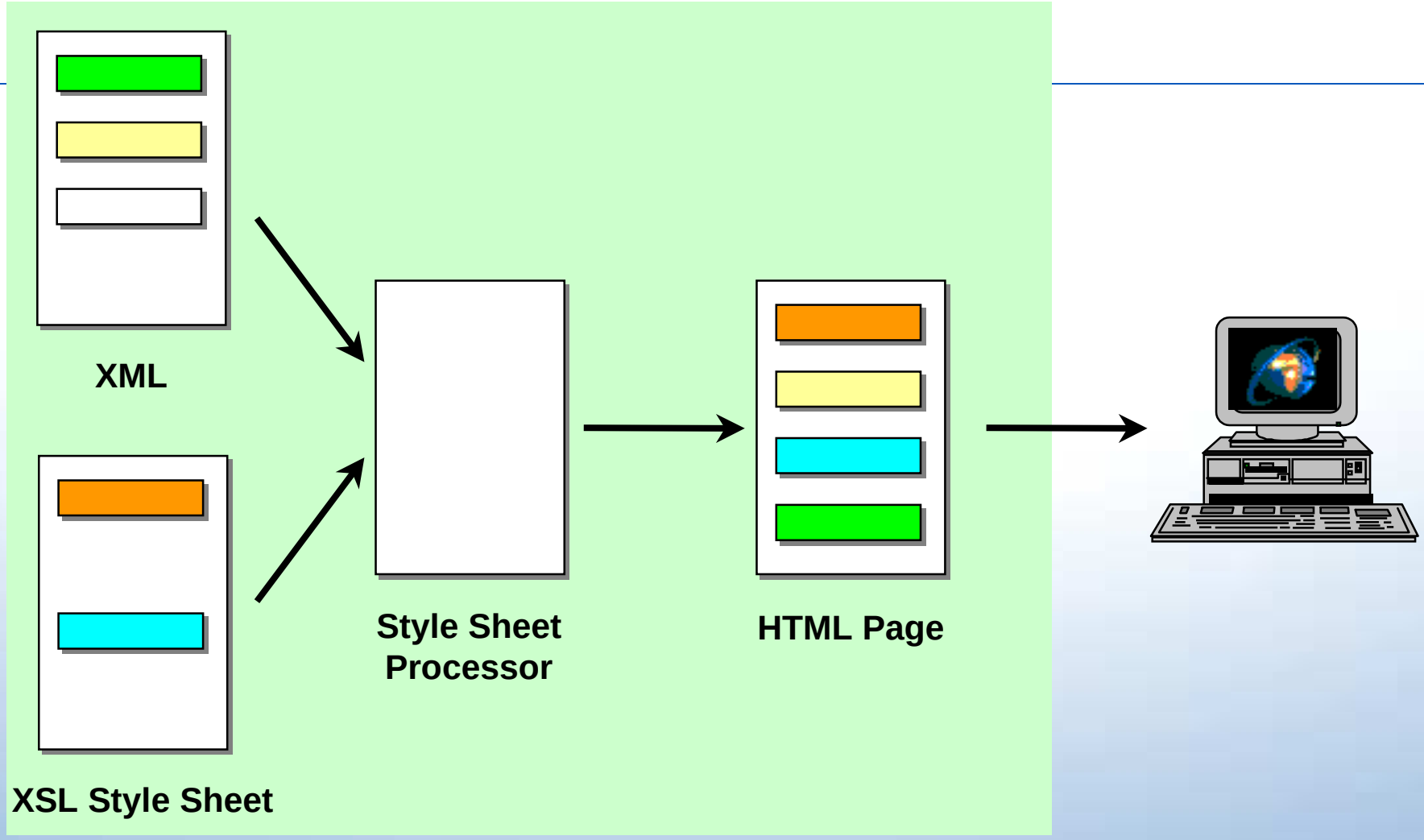
Example :

```
<xsl:template match="yacht[@yachtid="145"]"/>
```

XSLT Syntax

<xsl:apply-templates>	Process subnodes
<xsl:apply-templates select="..">	Process specified nodes
<xsl:for-each select="..">	Iterative processing
<xsl:if test="expression">	Conditional processing
<xsl:choose>	Decide structure
<xsl:when test="expr">	Element of Decide structure
<xsl:otherwise>	"else" processing
<xsl:sort>	Process subnodes in sorted order
<xsl:value-of select="..">	Insert content

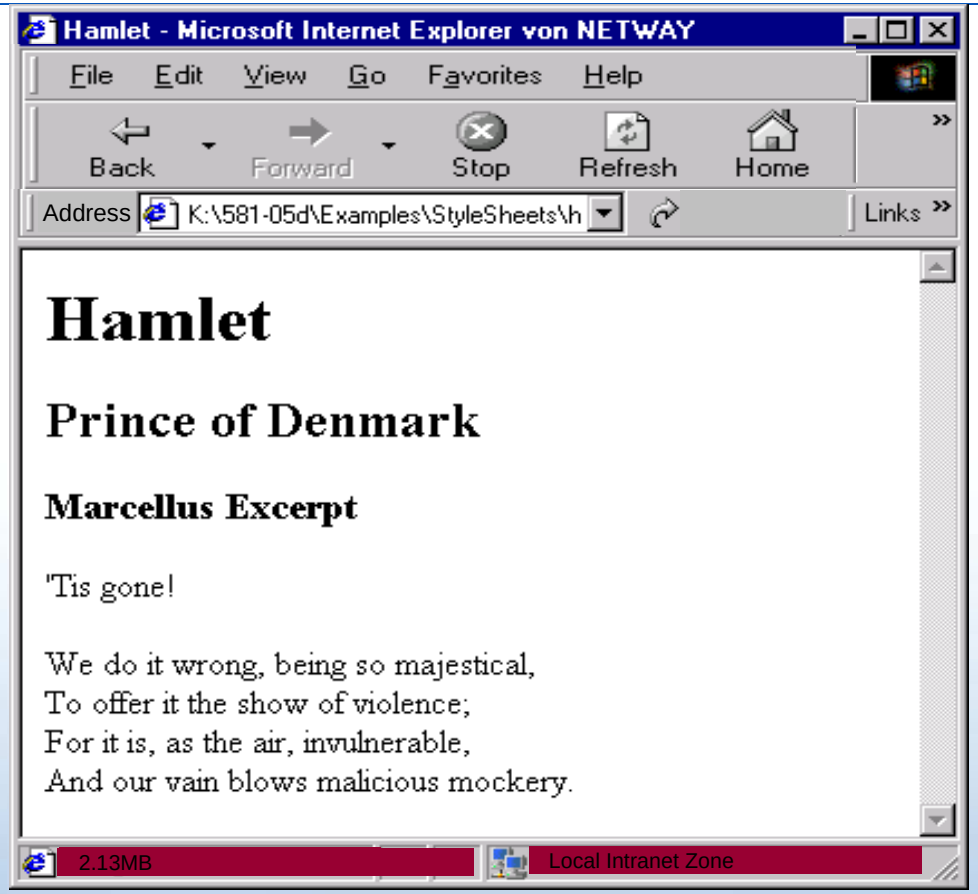
Style Sheet Processing



XSL Style Sheet

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
  <HTML>
    <HEAD><TITLE>Hamlet</TITLE></HEAD>
    <BODY><H1>Hamlet</H1><H2>Prince of Denmark</H2>
                          <H3>Marcellus Excerpt</H3>
         <xsl:for-each select="//Mar">
          <P>
            <xsl:apply-templates>
              <xsl:template match="Line">
                 <xsl:value-of /><BR/>
              </xsl:template>
            </xsl:apply-templates>
          </P>
         </xsl:for-each>
    </BODY>
  </HTML>
</xsl:template>
</xsl:stylesheet>
```

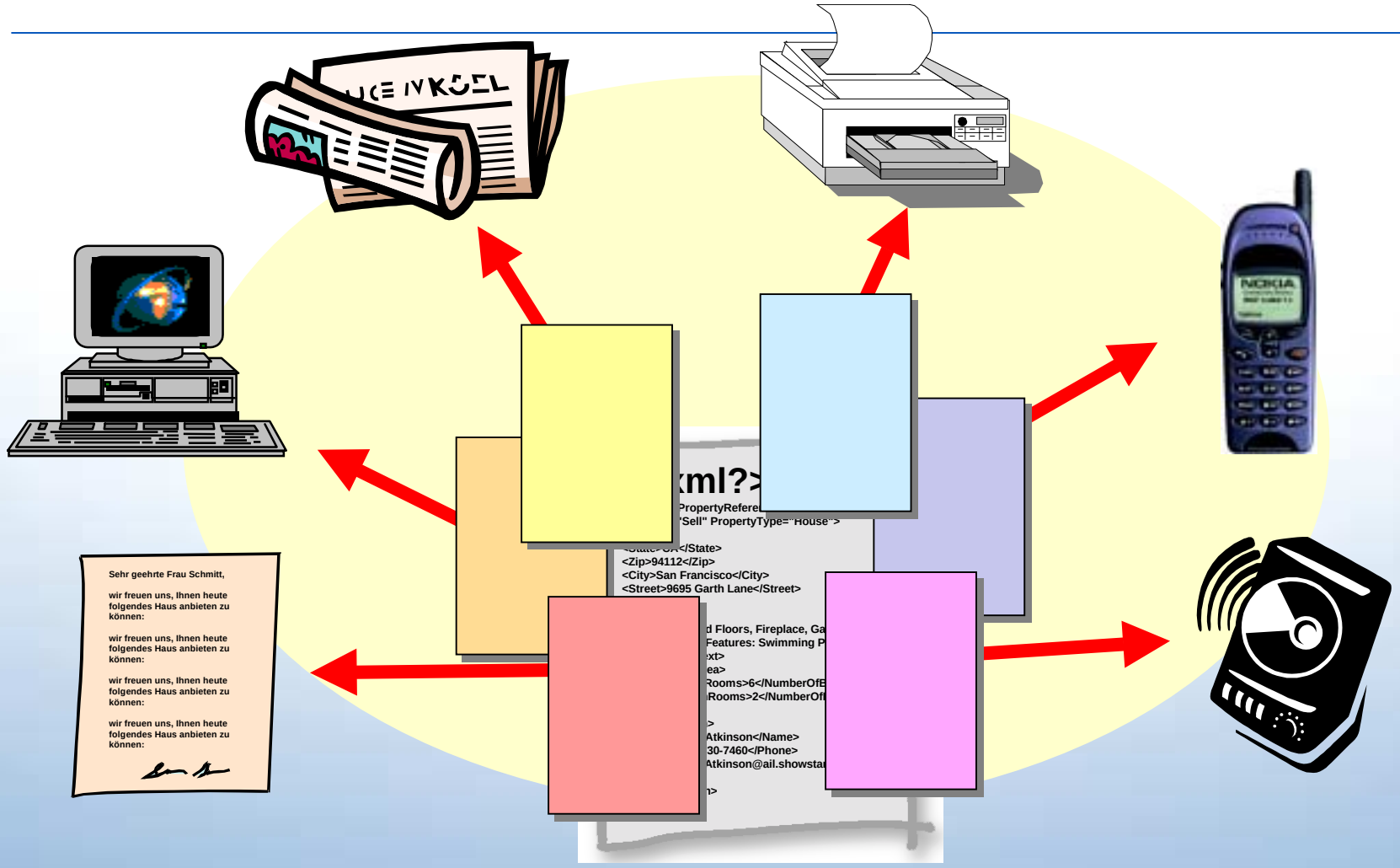
Display


Hamlet - Microsoft Internet Explorer von NETWAY
File Edit View Go Favorites Help
Back Forward Stop Refresh Home
Address K:\581-05d\Examples\StyleSheets\h
Links
Hamlet
Prince of Denmark
Marcellus Excerpt
'Tis gone!
We do it wrong, being so majestical,
To offer it the show of violence;
For it is, as the air, invulnerable,
And our vain blows malicious mockery.
2.13MB
Local Intranet Zone

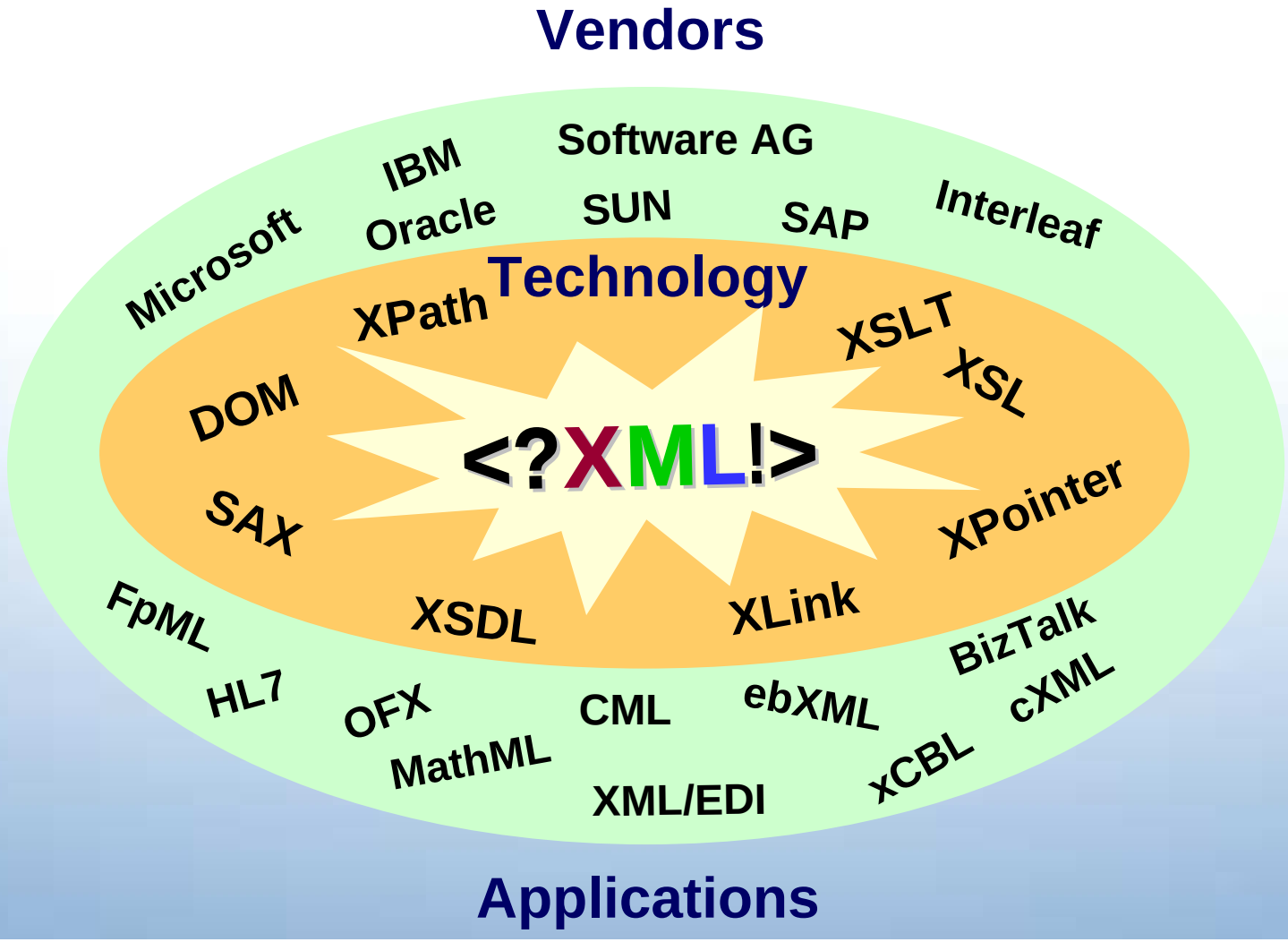
Summary

- XML separates the content of a document from its formatting.
- Presentation is realized using Style Sheets.
- For the presentation of XML data, CSS Style Sheets can be used that provide an extensive support for the various output media.
- The XSL Style Sheet Language has powerful functions to represent particular portions of a document.
- With the help of XSL transformations, documents can be completely reordered before they are presented.

XML Separates Content From Presentation



Broad XML Support!



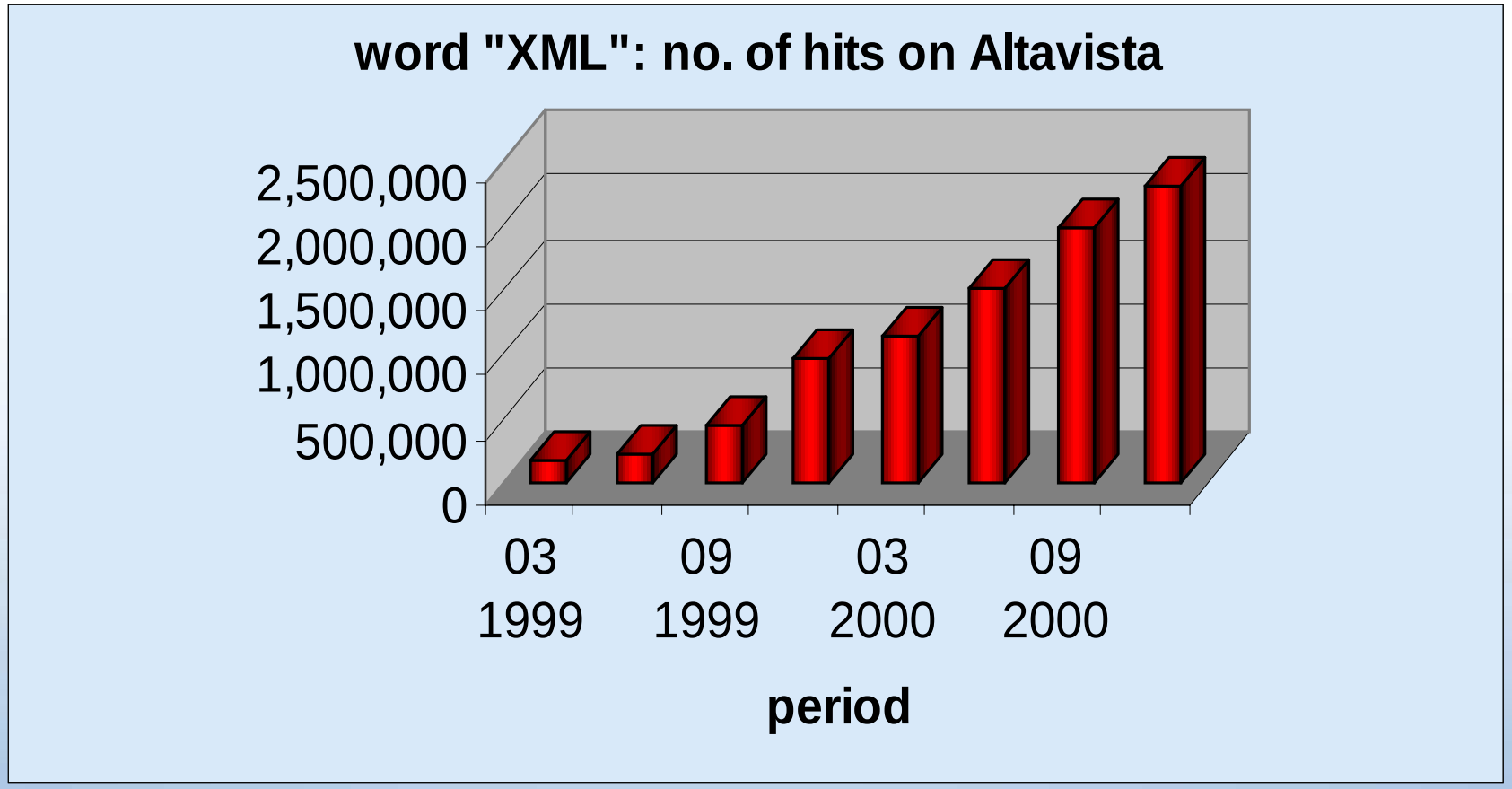
Who Does What?

- **W3C**
  - Defines the XML standard
  - Defines the schema standard
  - Defining XML query languages , XSLT, Xpath, XLink, XPointer...
- **Industry standards groups**
  - Define specific business documents
  - Work with members to gain acceptance
- **Technology vendors**
  - Provide tools and support for standard technologies

Bottom Line - The XML Revolution

- Our early, aggressive forecast about XML and its applications have been proven true.
- By year-end 2003, XML's impact on e-business (in particular) and technology (in general) will be as profound as the Web's impact is today (0.8 probability).
- The bottom line is that enterprises must understand the implications of the emerging XML-enabled web in order to succeed in e-business.

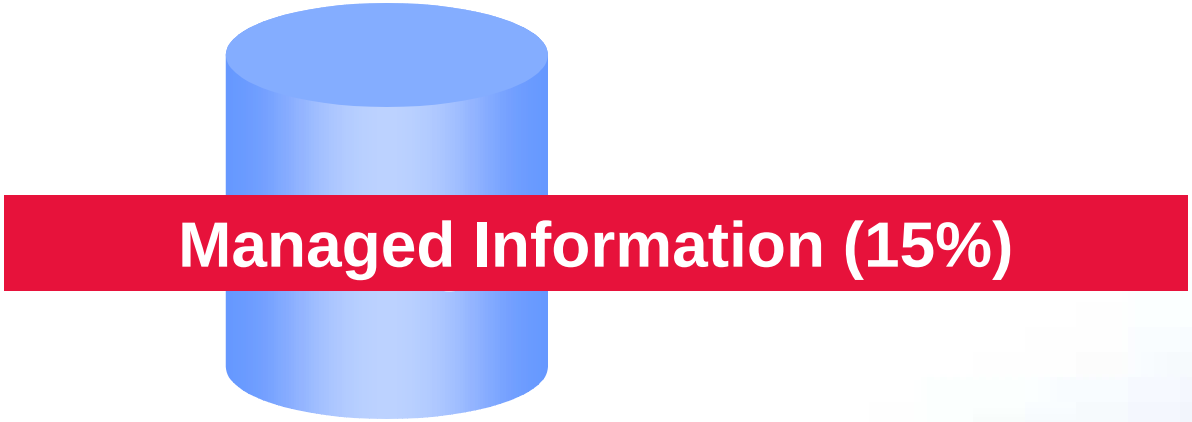
XML: The Interest In The Market



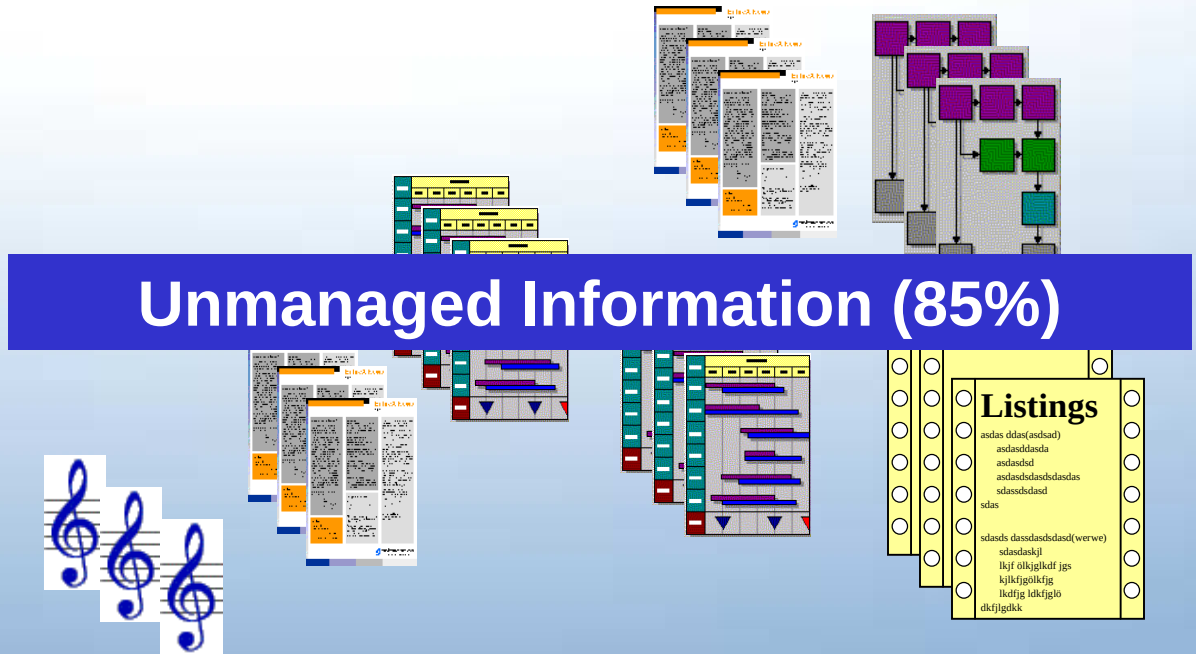
XML Benefits

How Much Data Are We Talking About ?

TeraByte Challenge (15% of data is in DBMS)

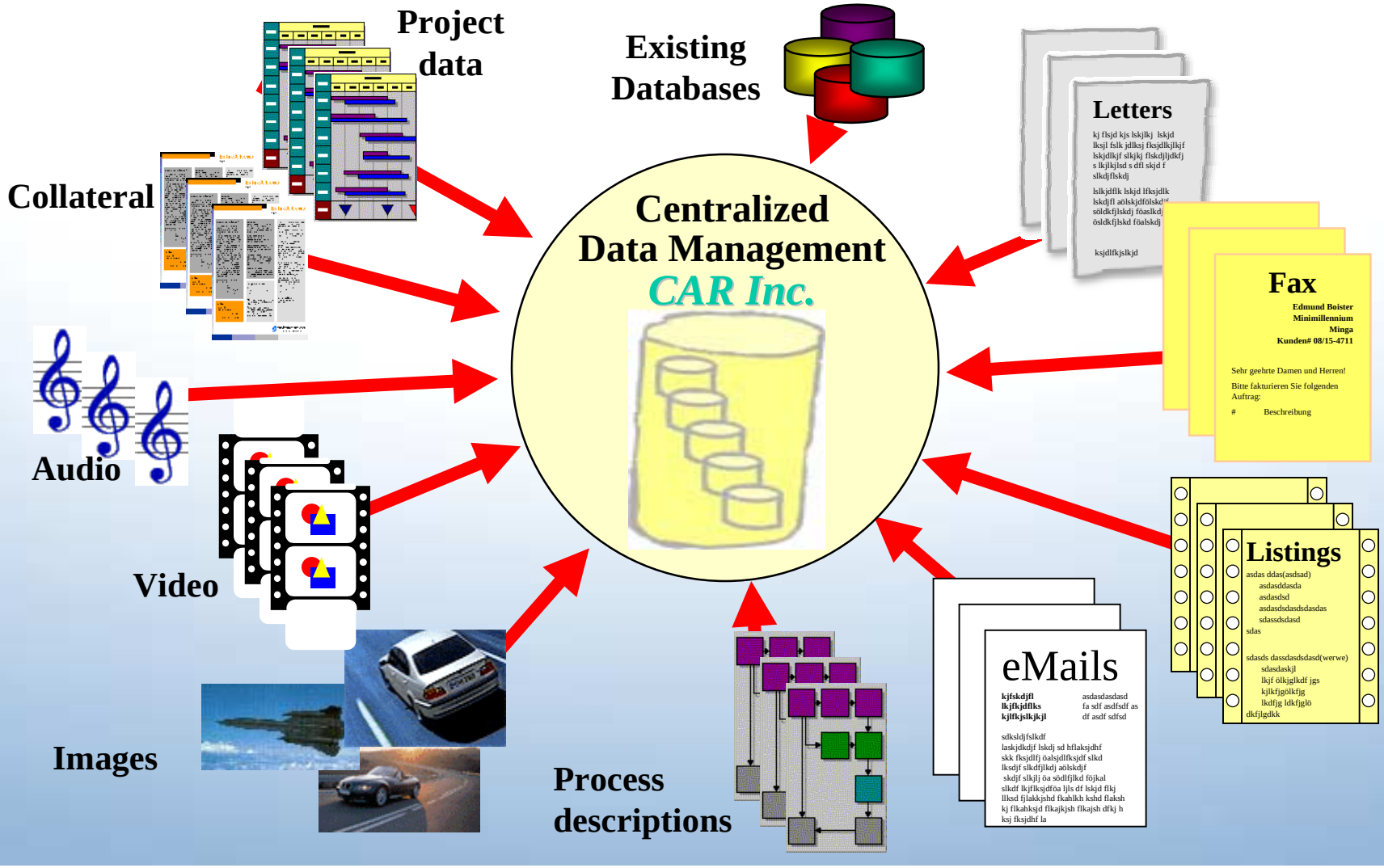


ExaByte Challenge (85% of data is not in DBMS)



Key XML Benefit:

Storage management for any data



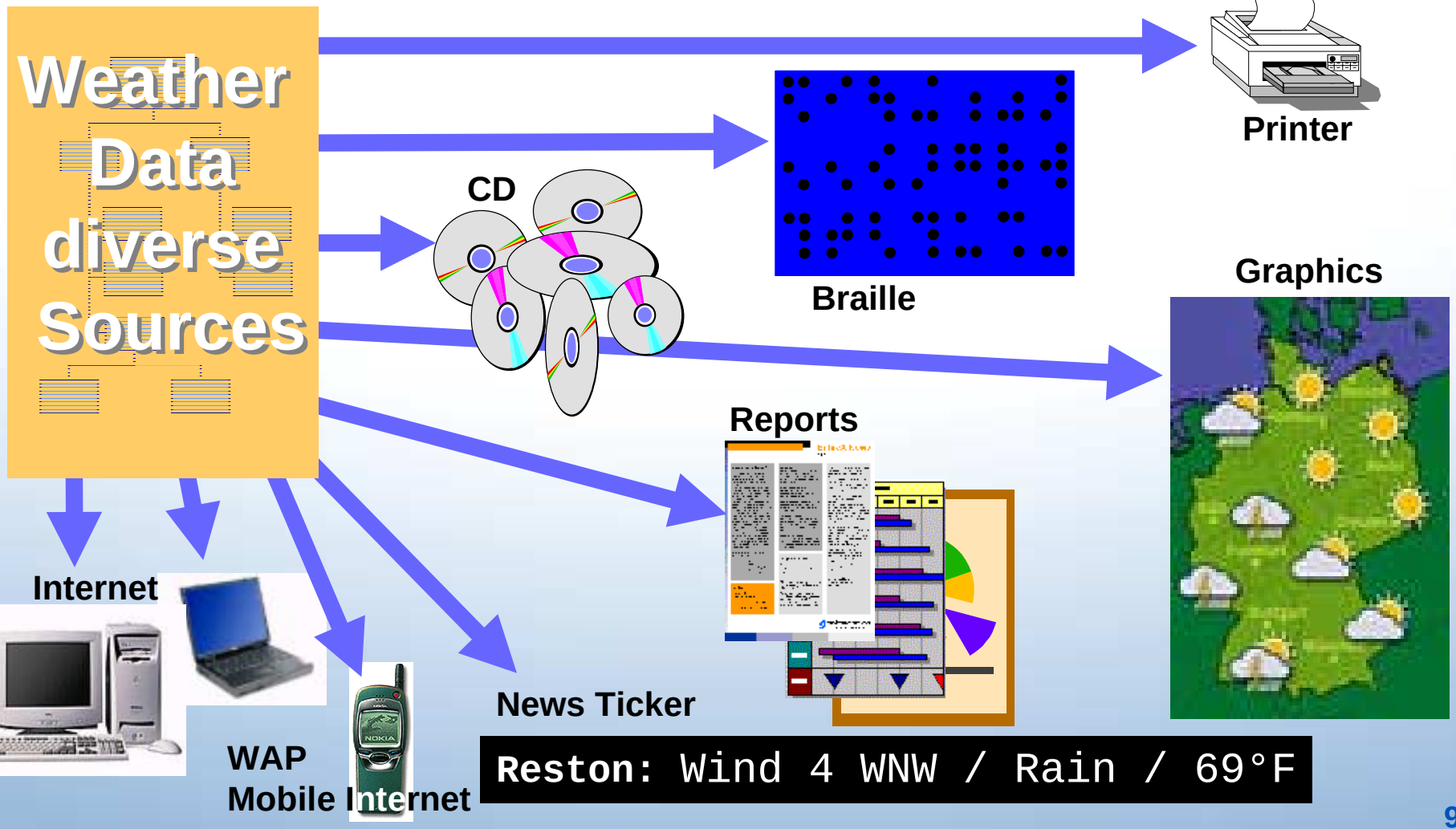
XML Advantages: Data Storage / Management

XML allows to store and manage the other 85% of the yet unmanaged document-based information

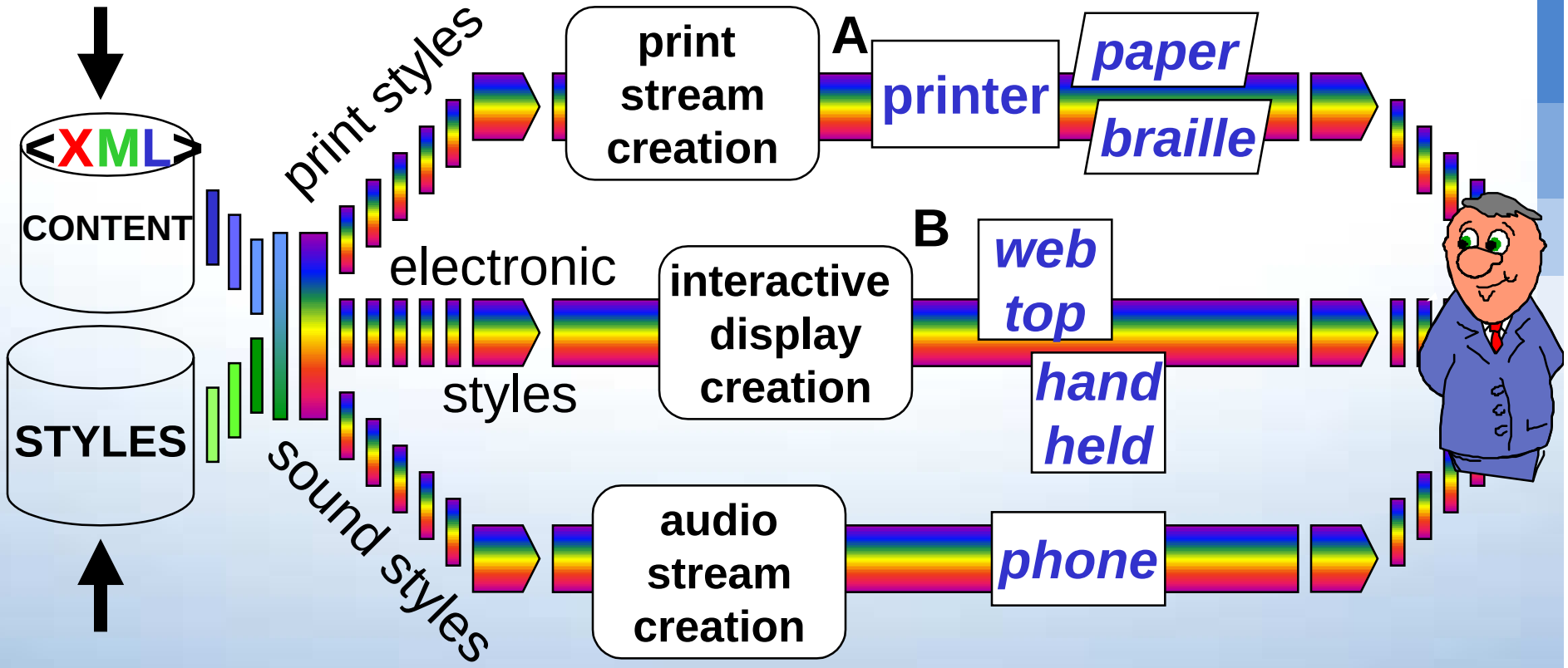
- **Intelligent, simple & structured searches**
  - XML-tags add meaning (meta-data) & structure to the content
- **Long-term availability of text-based data**
- **XML is independent** (prog-languages, apps, platf. or vendors)
- **XML documents can be validated** (reference docs)
- **XML can handle all kinds of data** (text, audio, video ...)
- **XML can be manipulated easily by common tools**

Key XML Benefit:

Push Button Publishing - one source for different devices and layouts



XML is Separation of Form and Content



Source:
GartnerGroup

A: Postscript, Meta code, AFP
B: HTML, XML, Braille, JPEG …

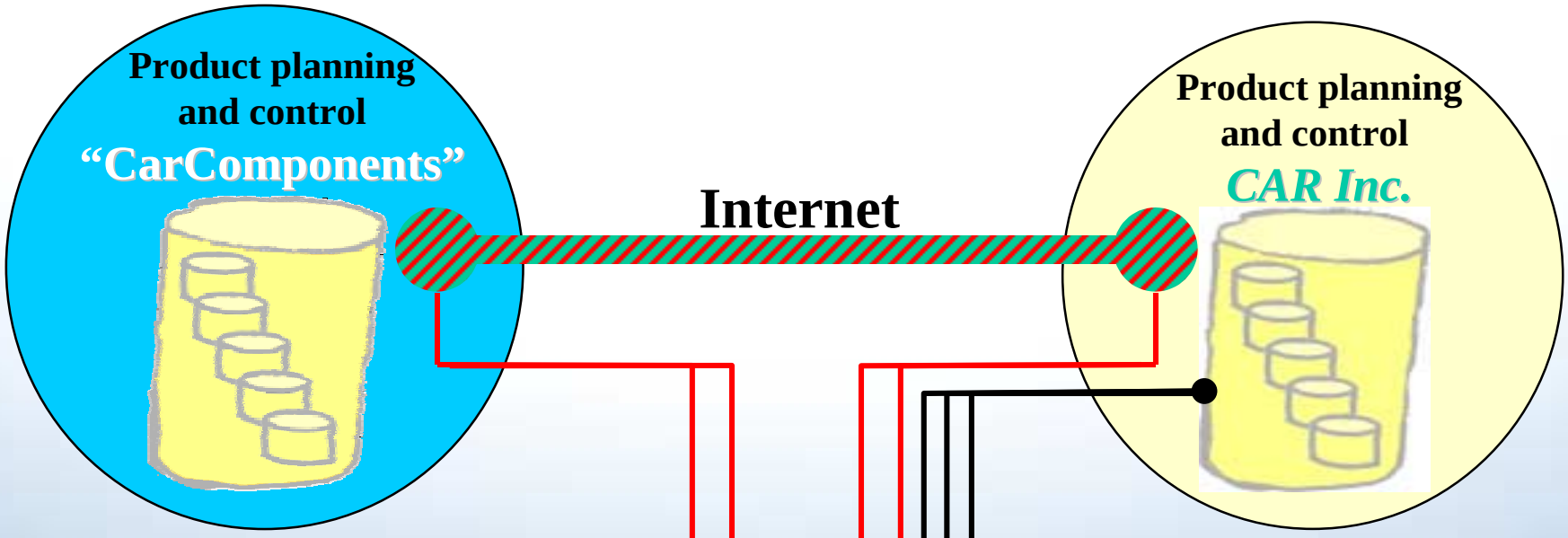
XML Advantages: Electronic Publishing

XML allows to re-use data easily in different formats

- **Separation of document content from its presentation**
- **Dynamic assembly & low-cost content personalization**
  - XSLT for data selection & target display device specific reformatting
  - Examples: WML, (X)HTML, PDF ....
- **Quality improvement for information, delivery & access**
- **Data consistency improvement** (write once-publish many)

Key XML Benefit:

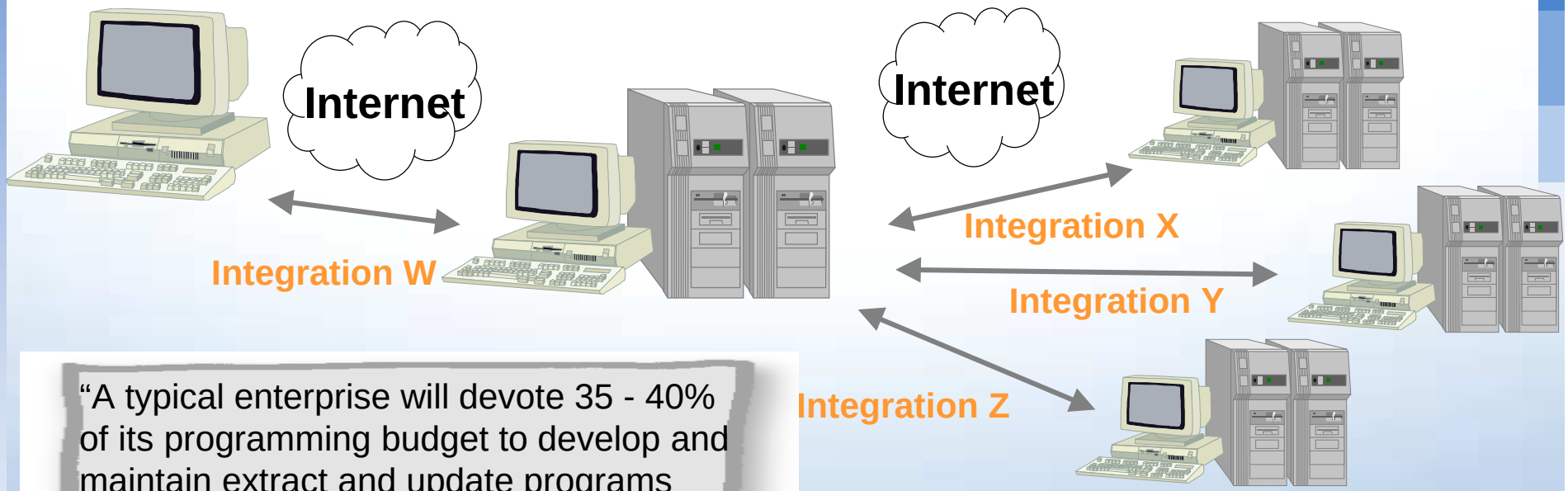
Electronic Document Exchange - Connecting Businesses



Supply-code	Seat Database	
50	Seat, front	Type 555
78	Seat, child-	Type TXc
45	Seat, rear	Type 555
15	Seat, passenger	Type O66
0	Seat, Sport-	Type wmh

Order Code	Car555iA		
13	1x	Engine	5,5 Liter
200	4x	Wheels	235/40
33	1x	Generator	12V
50	2x	**Comfortseat**	**Sport**
45	1x	**Rearseat**	**Luxus**

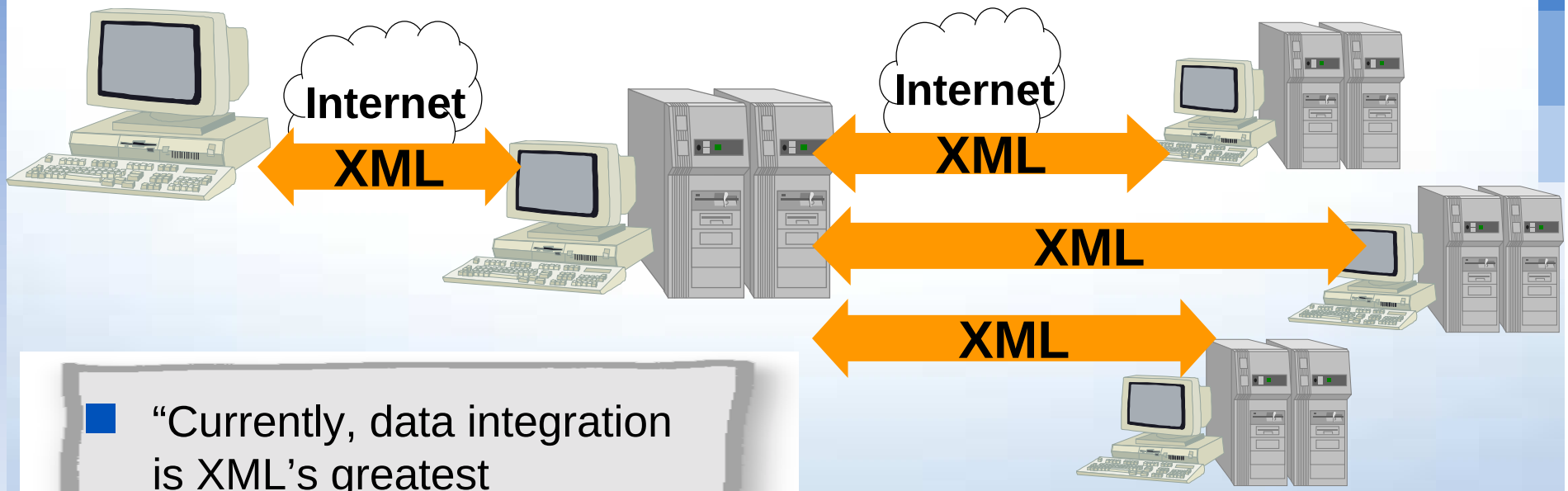
Data Exchange in the Past



"A typical enterprise will devote 35 - 40% of its programming budget to develop and maintain extract and update programs whose purpose is solely to transfer information between different databases and legacy systems"

Deborah Hess, Gartner Group 1999

Today: Business Process Integration



- "Currently, data integration is XML's greatest accelerator"

  Dr. Anthony Picardi,
  IDC, September 2000

XML Advantages: Electronic Data Exchange

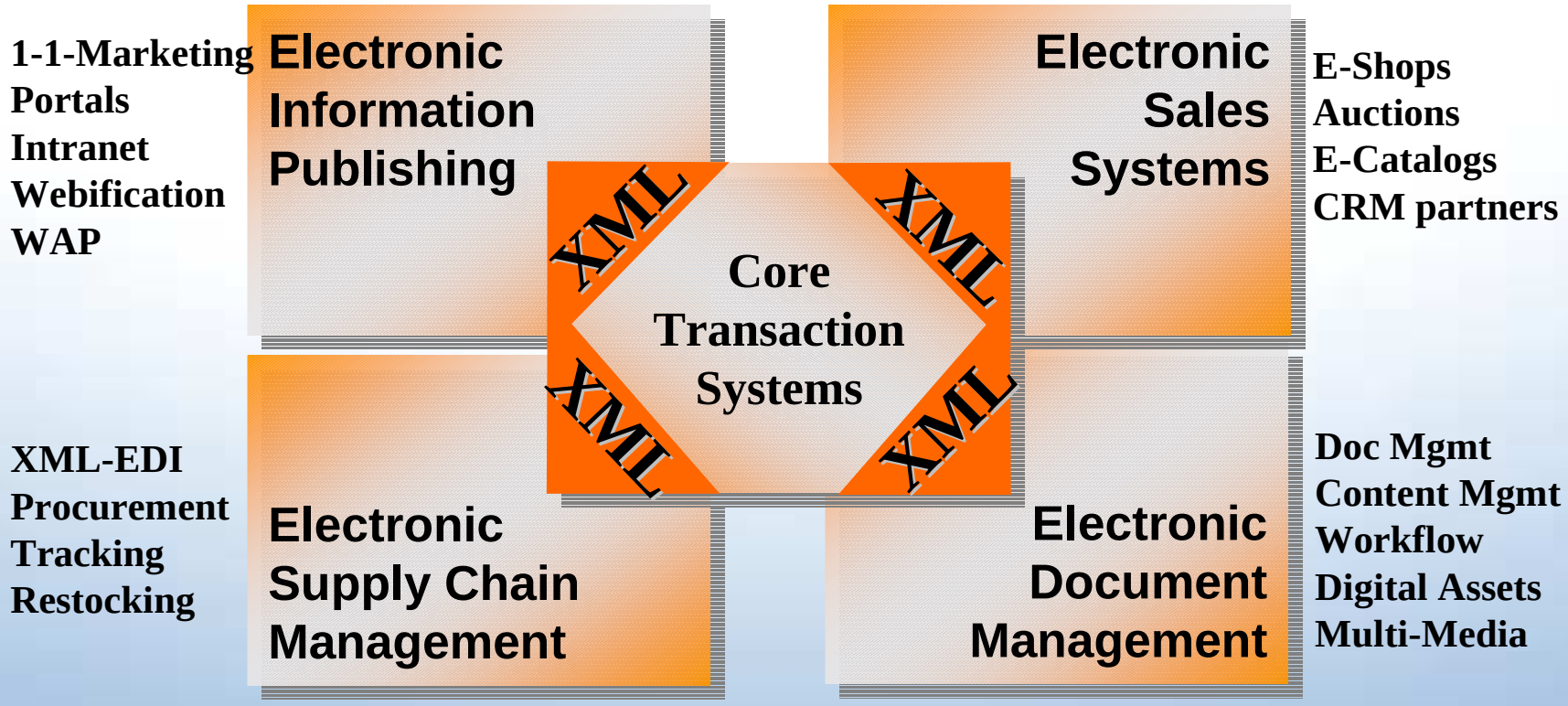
XML removes proprietary message format problems

- **Meta-language to define specific markup-languages**
- **Simple & cheap communication between applications on heterogeneous platforms** (data in medium-neutral format)
- **Simple and much faster implementation of communication interfaces**
- **Usage of standardized & easily extensible data descriptions** (Industries: health, banking, chemical, car...)
- **Usage of standardized communication means und protocols** (HTTP, TCP/IP)

XML-based Data Exchange Standards

- **EDI since 20 years - reserved for large enterprises only**
  - Infrastructure too expensive, inflexible, not robust enough
- **XEDI and XML/EDI are hybrid standards** (EDI+XML)
  - Internet saves cost (X12- or EDIFACT specific DTDs)
  - Integration of small and medium sized Enterprises (SMEs)
- **Various Standards bodies defining content of documents to be used in a business exchange**
  - **Industry specific standards:** RosettaNet (Computer), FpML (Financial).. *(e.g. more than 100 in 40 Branches)*
  - **Application specific standards:** cXML (Ariba), xCBL (CommerceOne)
  - **General purpose standards:** BizTalk, ebXML, XML.org, UDDI.org

Where to Use XML?



Gartner Suggests to Implement XML Now!

Issue	End Users	Vendors
Is XML too risky?	Only if you ignore it	Only if you ignore it
Are we too late?	No, just starting, but you must move	Depends on your space
How do we start?	With partner or customer integrations	Depends on your space
Where should we be in two years?	Starting "long" integrations via e-hubs	Not saying "XML" anymore
How do we win?	Do not quibble about terminology; understand XML process models	Stall; skip the obvious

Gartner

Conclusion

XML is the ideal standard for ...

- **Media-neutral storage of information (future proof)**
- **Flexible usage of data - „Create once - publish many“**
  - Automatic generation and formatting of output dedicated to target device
- **Cost-effective, standardized data exchange via Internet**
  - Integrates information from legacy systems with the Internet business
  - Connects enterprises without specific infrastructure *(e.g. needed for EDI)*
  - Universal general purpose vocabulary expected for e-business doc exchange ebXML?, BizTalk?, xCBL?, cXML? ...
- **XML helps in commercial, technical and scientific applications**
  - Internet Applications / Portals / E-Shops / E-Catalogs ...
  - Content- / Document Management
  - Supply Chain Management

XML is

the enabler for

electronic business

Questions?

The XML Company

Thank you !

References

- **XML**
  - http://www.w3.org/XML
  - http://www.xml.org
  - http://www.xml.com
- **XML Schema**
  - http://www.w3.org/TR/xmlschema-0/
- **XSLT**
  - http://www.w3.org/TR/xslt
  - http://xml.apache.org/xalan/index.html
- **DOM & SAX**
  - http://www.w3.org/DOM
  - http://www.megginson.com/SAX/index.html
- **General**
  - http://www.w3.org
  - http://tamino-platform.software-ag.de/articles/e-XMLCo_0011_FullArticle.doc