



# XML – An Introduction

# Source



- The slides in this presentation are made from the XML tutorial of W3Schools, available at [http://www.w3schools.com/xml/xml\\_what\\_is.asp](http://www.w3schools.com/xml/xml_what_is.asp)
- XML, Web Services, and the Data Revolution by Frank P.Coyle

# Markup Languages



- SGML:
  - Standard Generalized Markup Language
  - Mother of Markup Languages
- HTML
  - Most popular presentation language for web
- XML:
  - Draws heavily on the merits & shortcomings of HTML & SGML

# XML over SGML

- XML permits well-formed documents to be parsed without the need for a schema.
- XML is much simpler and more permissive in its syntax.
- XML specification is very small.
- Implementation of SGML on the Internet was difficult.
- SGML supports diverse community of users that produced a specification that was very difficult to implement.
- XML 1.0 simplified the specification.

# Issues with HTML



## **Merits:**

- Very easy to use & learn
- Presentation technology
- It is the most popular

## **Shortcomings:**

- NOT a data technology
- Poor Searching
- There is no Intelligence of content/data
  - ✦ We loose meaning associated with content
- Data cannot be represented hierarchically
- Limited set of tags

# XML over HTML

- XML cannot completely replace HTML.
- HTML has no capability to represent metadata, provide validation, support the basic needs of e-business.
- HTML is intended for consumption by humans whereas XML is meant for both machine and human consumption.

# XML ↔ HTML



- Similar in appearance
- Both are based on SGML

BUT...

- XML describes data
- HTML displays data

# Why XML?



- Derived as a subset of SGML
- Allows you to define your own tags  
XML: **<author> YAG </author>**  
HTML: **<B> YAG </B>**
- Provides meaningful & readable data
- Meaning searches can be performed
- Much simpler than SGML
  - SGML spec = 300 pages, XML = 33 pages
- Purely a Data Technology
  - XML was designed to carry data, not to display data.
- Supports compound documents.



# XML Advantages



- ✓ Web based
- ✓ Extensible
- ✓ License-free
- ✓ Platform independent
- ✓ Single end-to-end IT solution for electronic information exchanges

# XML Advantages



- XML files are human-readable unlike binary data formats.
- Widespread industry support exists for XML.
- Major relational databases now have the native capability to read and generate XML data.
- A large family of XML support technologies is available for interpretation and transformation of XML data for Web page display.

# XML - Uses



- XML Does not DO anything
  - XML was created to structure, store, and transport information.
- XML is just plain text
- XML separates data from HTML
  - With XML data can be stored in separate XML files.
- XML simplifies data sharing
  - provides h/w and s/w independent way of storing data.
- XML simplifies data transport
- XML simplifies platform changes
- XML makes your data more available
- XML is used to create new Internet languages

# How does XML look?



- Simple XML data would look like

**<book>**

**<title>** XML Tech **</title>**

**<author>** YAG **</author>**

**<level>** Freshman **</level>**

**</book>**

**<?xml version="1.0" encoding="ISO-8859-1"?>**

**<customer>**

**<name>**Ram Kumar**</name>**

**<Phone>**2354569**</phone>**

**<Email>**[ram@yahoo.com](mailto:ram@yahoo.com)**</Email>**

**</customer>**

# What Is XML ?

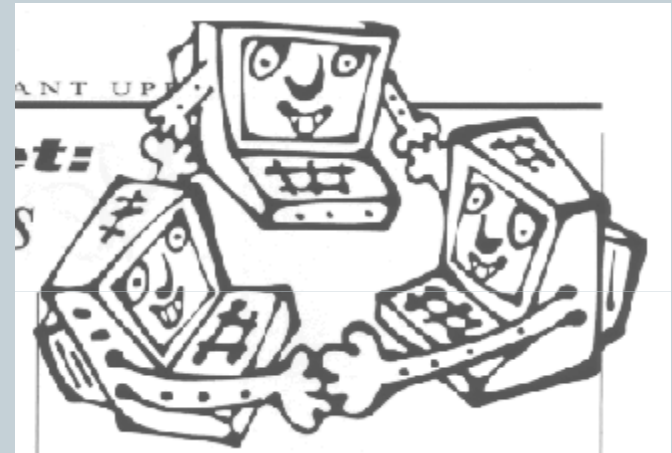


- “XML (eXtensible Markup Language) is
  - platform-independent,
  - self-describing,
  - expandable,
  - standard data exchange format
  - that can be used either independently or embedded and used within other solutions



# Platform Independent

- Windows
- Unix
- Macintosh
- Mainframe



# Self-Describing



- Example:  
`<DATE>July 26, 1998</DATE>`
- Describes the information, not the presentation.  
Format flexible.

# Expandable = Extensible



- HTML has a fixed set of tags  
<H1>, <B>, <PRE>
- XML lets you have your own tags  
<dangerous-substance>, <Shakespearean-character>, <cash-equivalent>



# Standard



- W3C (World Wide Web Consortium) [www.w3c.org](http://www.w3c.org)
- XML 1.0 specification was issued as standards-based text format for interchange of data as of February 1998.
- W3C XML Working Group designed XML as a simplified subset of SGML

# Freedom



- Extensible markup language
  - Customized tags
  - Tags give meaning to the content
  - Separates data from style

# XML : Design by Omission



- No display is assumed. Unlike HTML, XML makes no assumptions about how tags will be rendered in a browser.
- There is no built-in data typing. DTDs and XML Schema provides these support.
- No transport is assumed. The XML spec makes no assumption about how XML is to be transported across the Internet. Hence can be delivered by means HTTP, FTP, or SMTP.

XML limits scope -> to maximize interaction with other tech

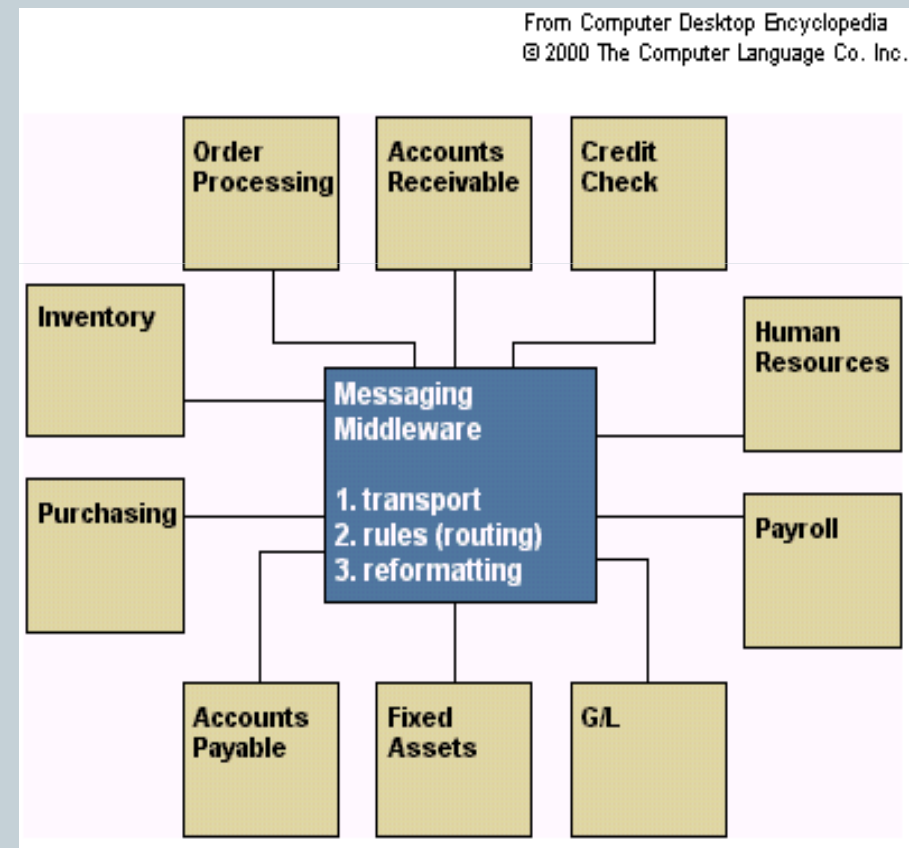
# XML and the Web



- Instead of exchanging data manually or over proprietary networks, data vocabularies can be defined using XML and delivered directly using standard protocols.
- Message Oriented Middleware (MOM) – messaging servers and software that sits between conversational participants.
  - MOM supports the asynchronous delivery of XML, and the ability to broadcast communications to multiple recipients.
- Web services make B2B (software-to-software) direct connections feasible over the Web.

# MOM or Messaging Middleware

- Software that provides an interface between applications, allowing them to send data back and forth to each other asynchronously.
- Data sent by one program can be stored in a queue and then forwarded to the receiving program when it becomes available to process it.



# SOAP



- SOAP is an XML based protocol. Lets clients and providers talk to each other and exchange XML data.
- SOAP defines set of rules for moving data, either in P2P fashion or through an intermediary.
- Three basic options for distributed computing:
  - MS's Distributed Component Object Model (DCOM) - ORPC
  - Java's Remote Method Invocation (RMI) - JRMP
  - OMG's Common Object Request Broker Arch. (CORBA)- IIOP
- Main drawback is limitations in the above technologies:
  - CORBA could talk to CORBA, RMI to RMI, DCOM to DCOM, but they could not talk to each other nor directly to the Web.
- SOAP opens up new options for distributed computing!

# Web Services



- Web services builds on a SOAP foundation.
- Web services is a process and set of protocols for finding and connecting to software exposed as services over the Web.
- Web services can be movie review, weather report, hotel and airline booking package etc.,
- Three major aspects of Web Services:
  - A Service provider provides an interface for software that can carry out a specified set of task.
  - A Service requestor discovers and invokes a software service to provide a business solution.
  - A broker manages and publishes the service. Service providers publish their services with the broker.
- Web services = repository + client + provider



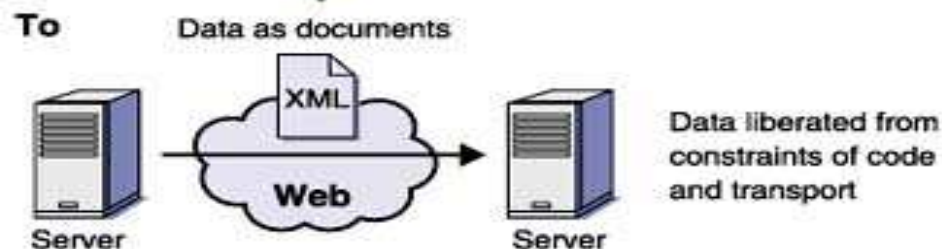
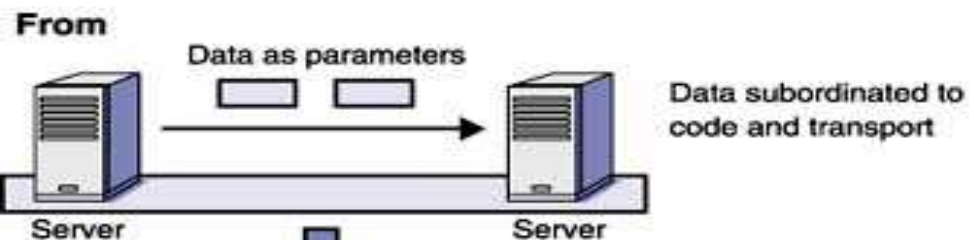
# XML : The Three Revolutions



# Three Revolutions

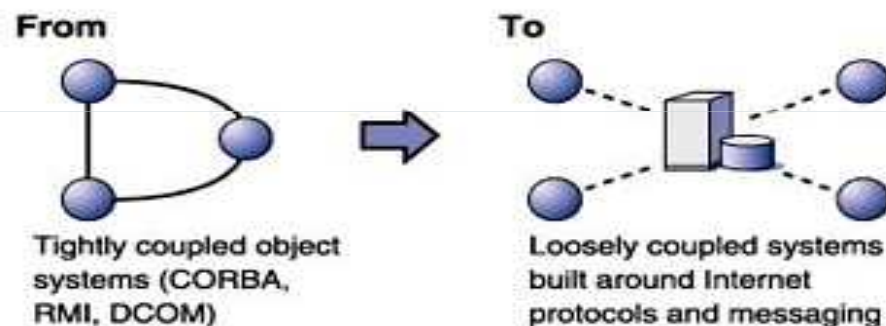
XML's impact on three major areas are:

- # Data – XML is free from program-dependent formats.
- # Architecture – change from tightly coupled distributed systems to loosely coupled based on the Web.
- # Software – simplicity and collaboration instead of building monolithic applications.



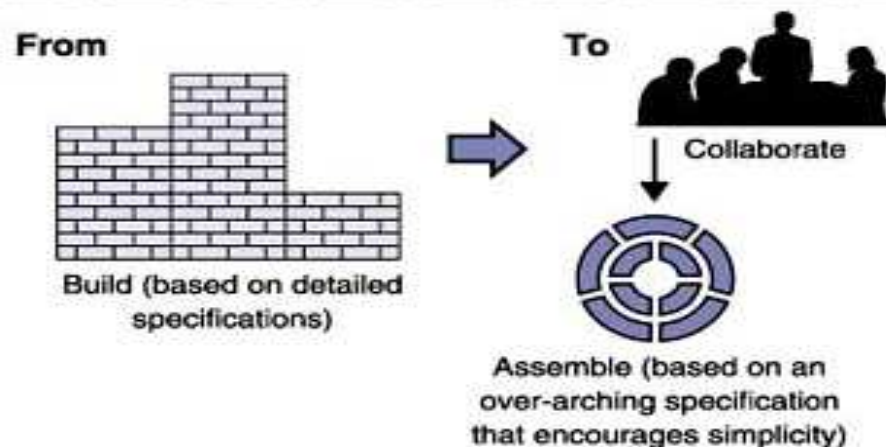
### The Data Revolution

Data isn't tied to transport or language—is free to travel using Web protocols



### The Architecture Revolution

Loosely coupled systems centered around the Web and message-oriented middleware



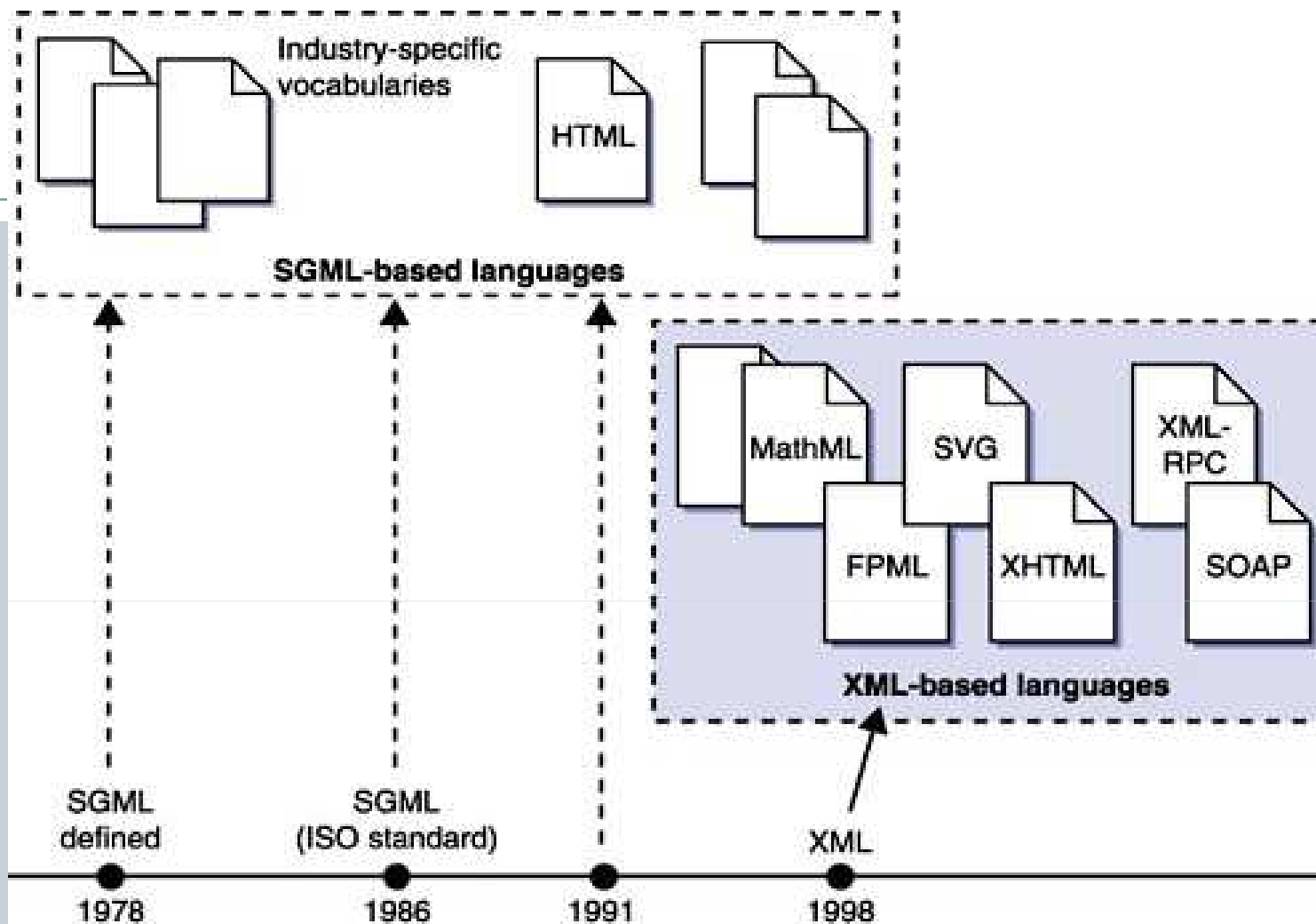
### The Software Revolution

Simplicity of design and the power of combination and collaboration

# The Data Revolution



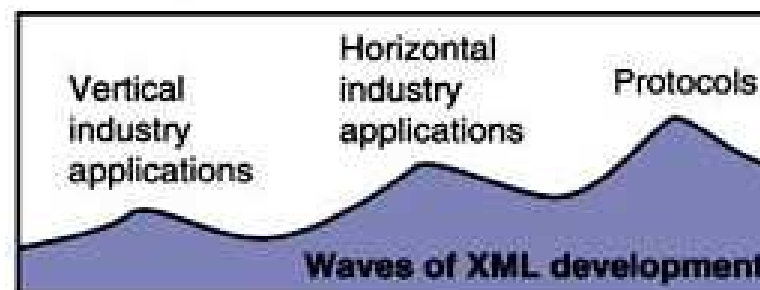
- Prior to XML, data was proprietary, closely associated with applications for format and process.
- XML's strength is its data independence.
- **XML is pure data description, not tied to any programming language, operating system, or transport protocol.**
- Data is free to move about globally without constraints.
- XML's sole focus on data -> variety of transport technologies may be used to move XML across Web.
- Alternative to CORBA, RMI, and DCOM which won't work over TCP/IP.



### From SGML to XML

#### Requirements:

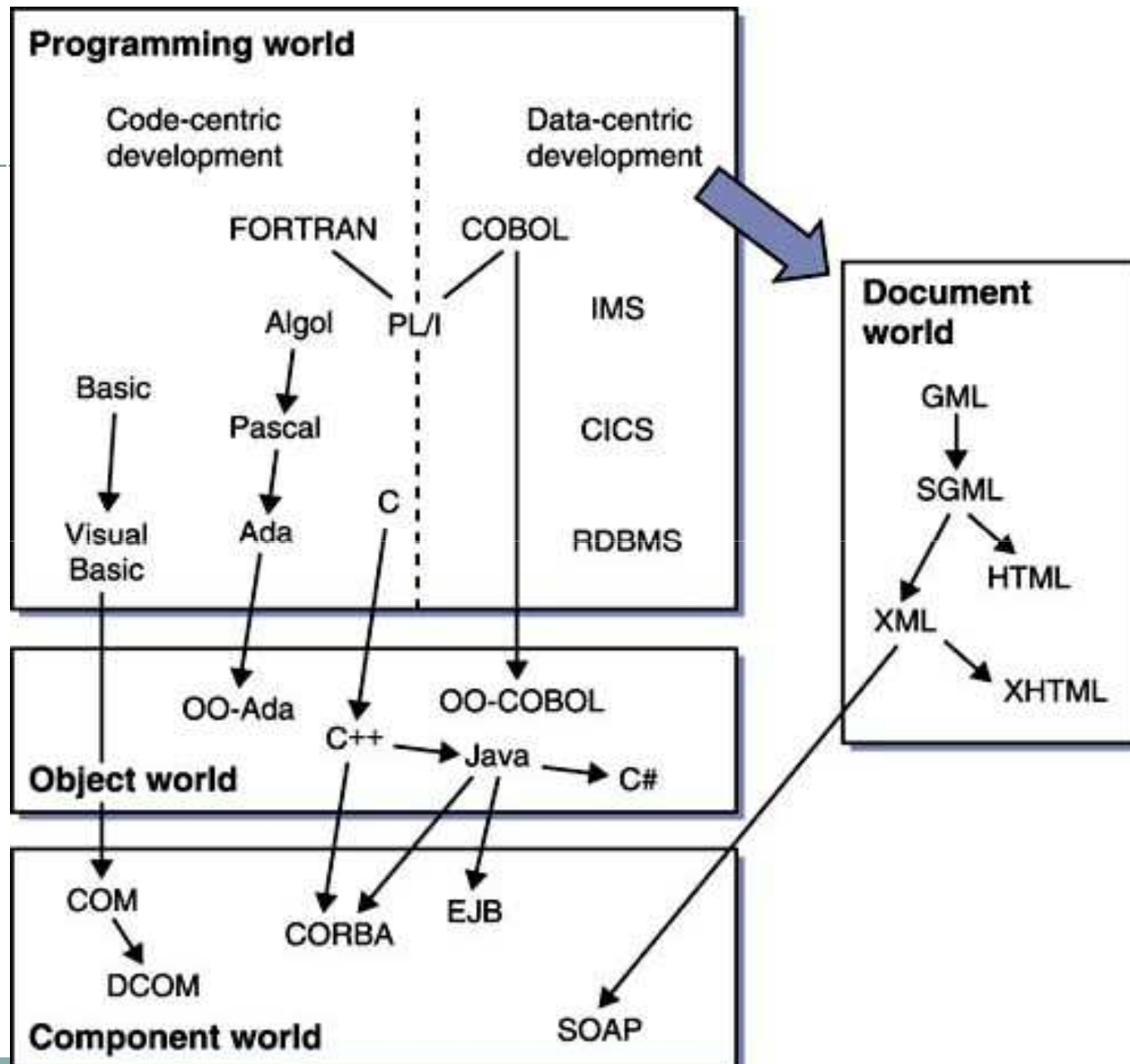
- Easy to define new data vocabularies
- Simple: do no more than necessary
- Must be easy to develop support tools



# The Code, Data, Document Cultures



- Code characterized by a focus on programming languages.
- Data is characterized by COBOL, data processing, databases.
- From code perspective, data is to be transported to procedure.
- From data perspective, data is to be stored and manipulated.
- Transactions are essential to enterprise applications.
- Component frameworks provide transactions as a service.
- XML has emerged from a document culture.

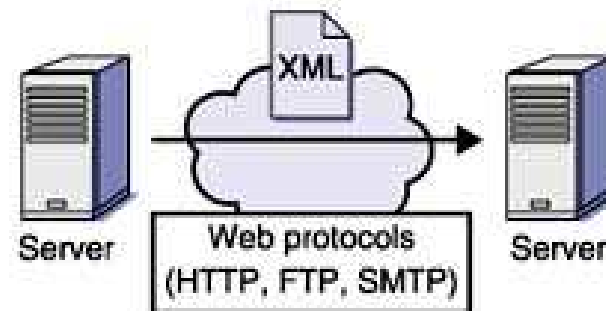


# Data Revolution



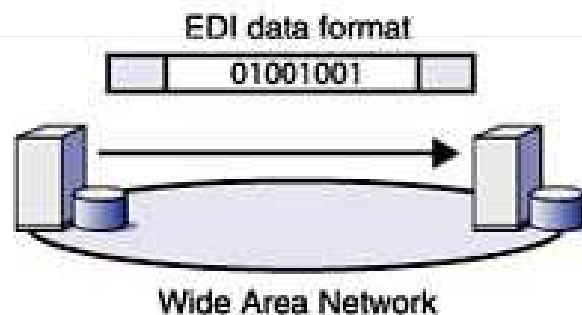
- EDI – network dependent, CORBA, RMI, DCOM – dependent on underlying Object infrastructure.
- XML is technology that allows data to be freed from the constraints created by code-centric infrastructures.
- XML allows code to be treated as data.

## The Data Revolution



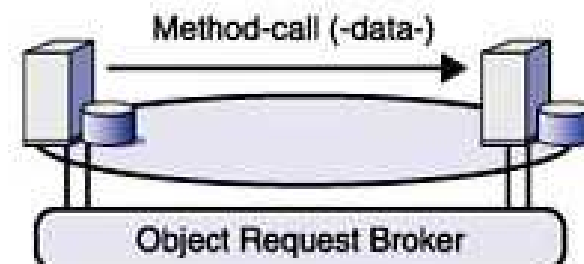
Data is free to move about the Web,  
not dependent on programming  
language or transport protocol

### EDI



- Data formats and messages defined by EDI standards
- Applications run in batch mode outside the Internet
- Proprietary wide area network required to deliver EDI messages

### CORBA, RMI, DCOM



- Data passed as parameters to method calls of an object-oriented language
- Platforms require code to interface with ORB

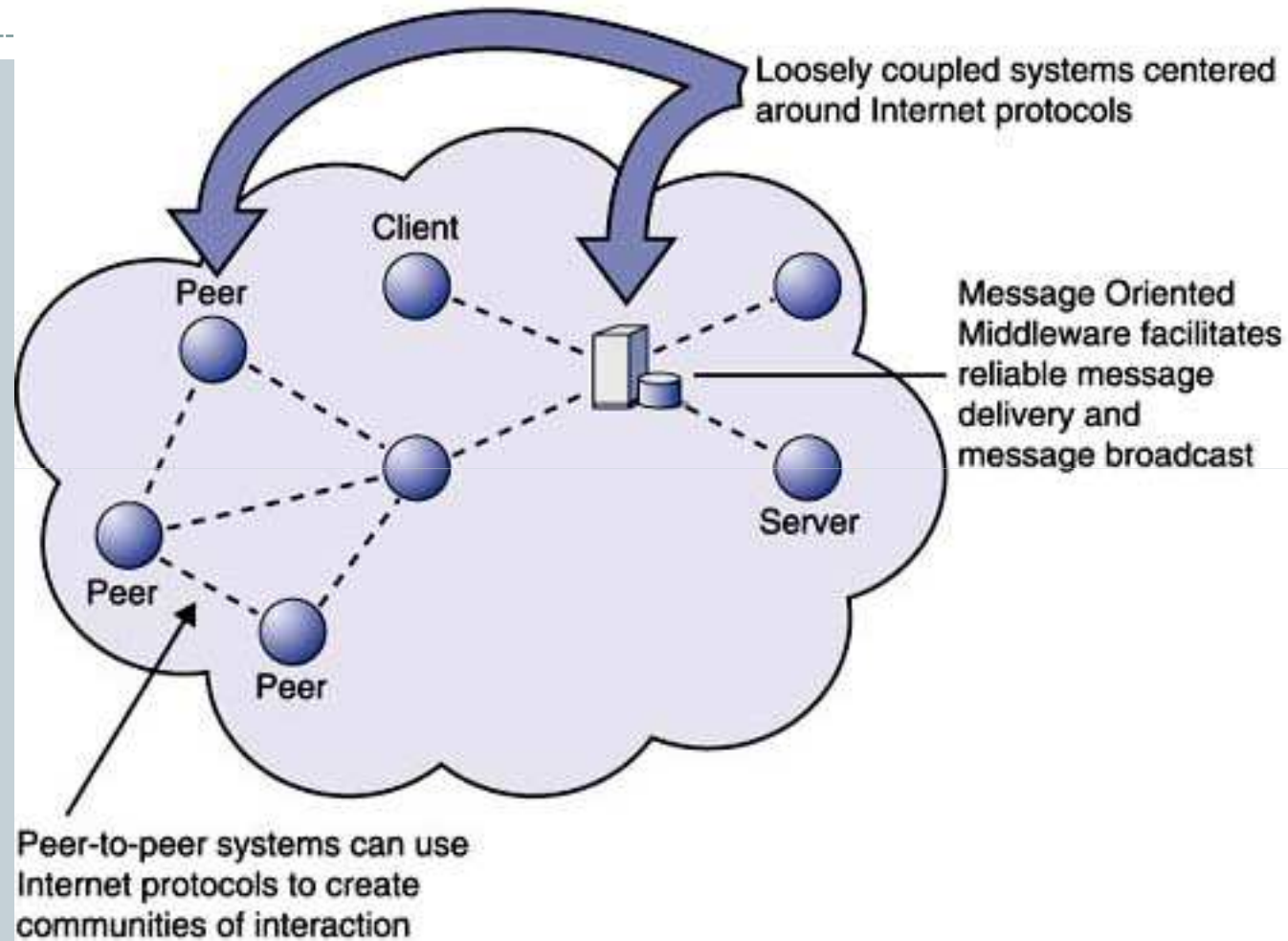


# Architectural Revolution



- XML brings the change in architectural thinking and new possibilities for distributed computing.
- Creates transition from object-based distributed system to architectures based on Web services using open Web technologies.
- Move from tightly-coupled systems such as CORBA, RMI, and DCOM, each with their own transport protocol to loosely coupled systems riding atop standard Web protocols – TCP/IP.
- XML and the Web have enabled the loose coupling of software components.

## The Architecture Revolution

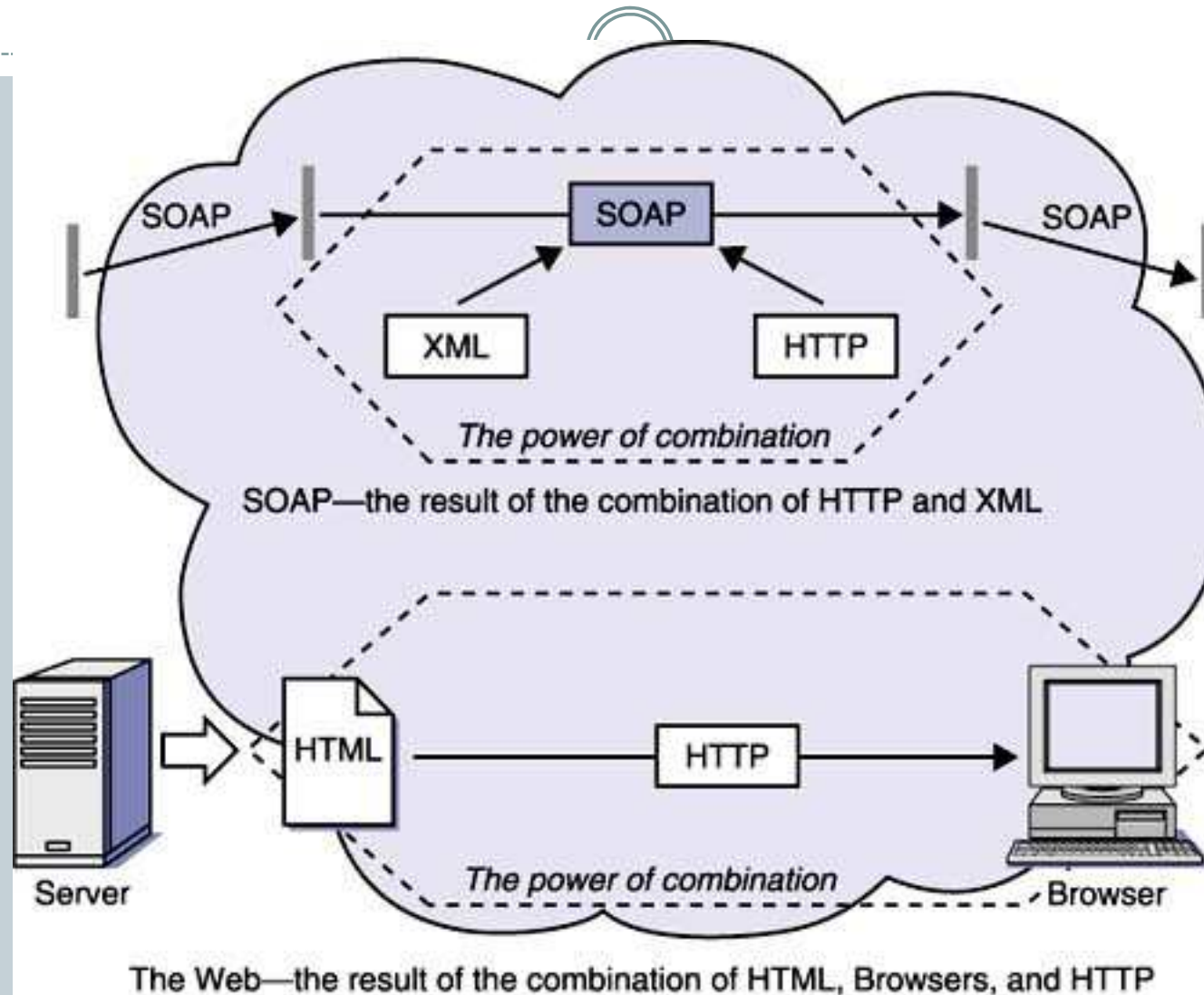


# Software Revolution



- 1970s and 1980s – monolithic applications for specific problems.
- Problem with large software project -> ill-suited to add new functionality and adapting to technological changes.
- 1990s – concept of Simplicity and collaboration
- Instead of defining all requirements, creating building blocks capable of combination with other building blocks.
- The Web – is an example of the Power of combination.

# Software Revolution



# Summary



- Reduced dependence on proprietary data formats for applications.
- A shift from relying on tightly coupled systems such as CORBA, RMI, and DCOM to a more loosely coupled Internet-based framework centred around XML and SOAP.
- A change in focus from object-oriented to service-oriented software.
- The emergence of Web services as technology for discovering and connecting to Internet-based services
- Major shift in the software industry from monolithic applications to applications built up from constituent pieces in an environment that fosters open, collaborative development.