M10 L1
XML Database and XQuery Language

Ira A. Fulton Schools of Engineering
Arizona State University

Lecture Outline

- Why XML Database and XML Query Language?
- XQuery Language
- IBM DB2 XML Database Example
- Oracle XML Database Example

Why XML Database and XML Query Language?

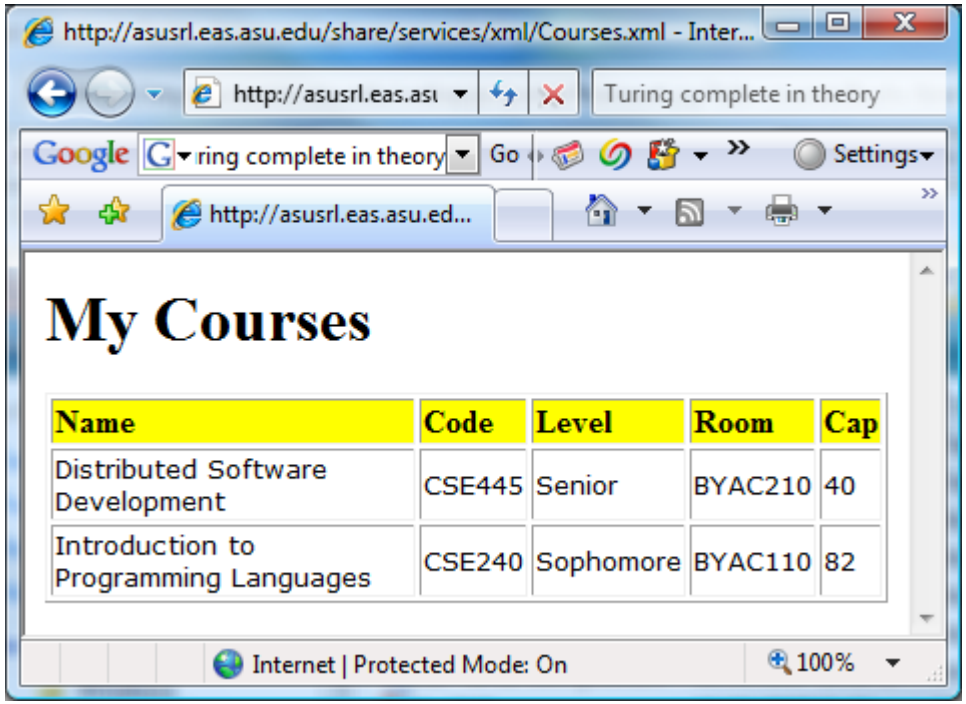
http://www.w3.org/TR/xquery/

- Most of the world's business data is stored in relational databases.
- The relational query language, e.g. SQL, is mature and well-established.
- Can SQL be adapted to query an XML database?
  - Leverage existing software
  - Leverage existing user skills
- Can a relational BD be used to store XML data efficiently?
- How is an XML database different from a relational database?
- Can we have an XML database, that selects data inside the database's computer?

Converting XML Doc to Table
if the data are regular

```
<?xml version="1.0"?>
<Courses>
    <Course>
        <Name>Introduction to Programming
    Languages</Name>
        <Code>CSE240</Code>
        <Level>Sophomore</Level>
        <Room>BYAC110</Room>
        <Cap>82</Cap>
    </Course>
    <Course>
        <Name>Distributed Software
    Development</Name>
        <Code>CSE445</Code>
        <Level>Senior</Level>
        <Room>BYAC210</Room>
        <Cap>40</Cap>
    </Course>
</Courses>
```

XSLT, DataSet, etc.



Problems of Using Relational DB for XML

- The mapping strategy is hard to define when an XML document has a complex structure;
- When an XML document is an irregular tree, it may end up with using many tables or a table with many null column values;
- When retrieving XML documents, many multi-table join queries need to be performed, which make the queries slow.

Country	Count
Canada	200
Mexico	300
France	250
USA	750
Zambia	100

Customer Relation

Province	Count
Ontario	50
Alberta	100
Quebec	50

State	Count
Arizona	250
Utah	100
Texas	300
Oregon	100

City	Count
Phoenix	100
Kingman	20
Flagstaff	40
Tucson	60
Yuma	30

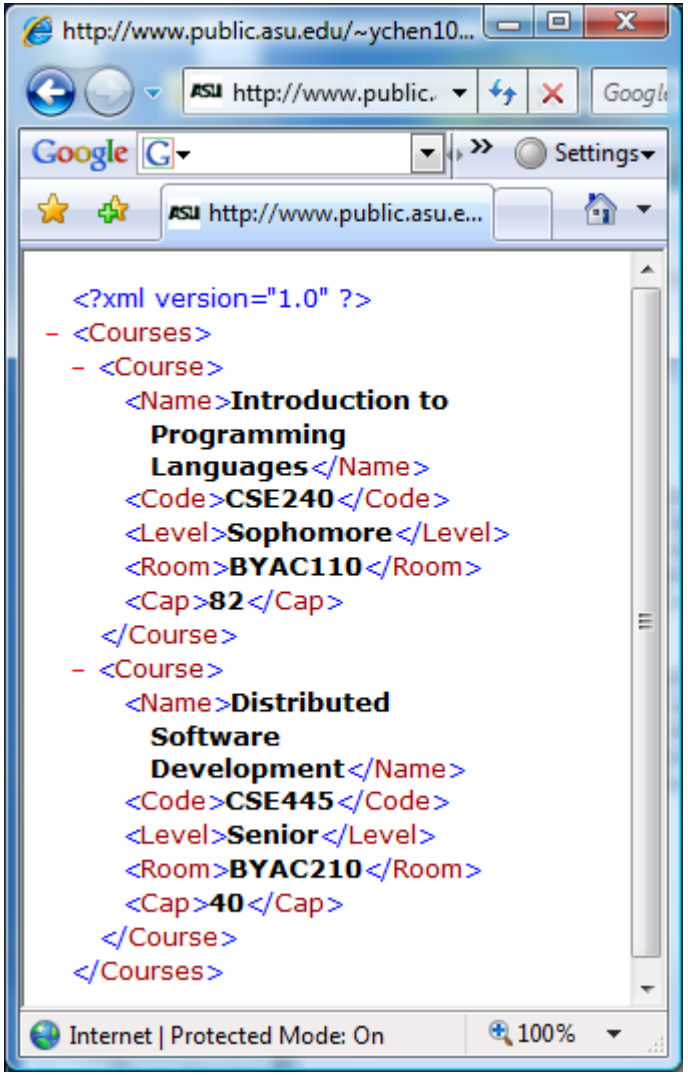
What is the Native XML Database

Store XML document as is.

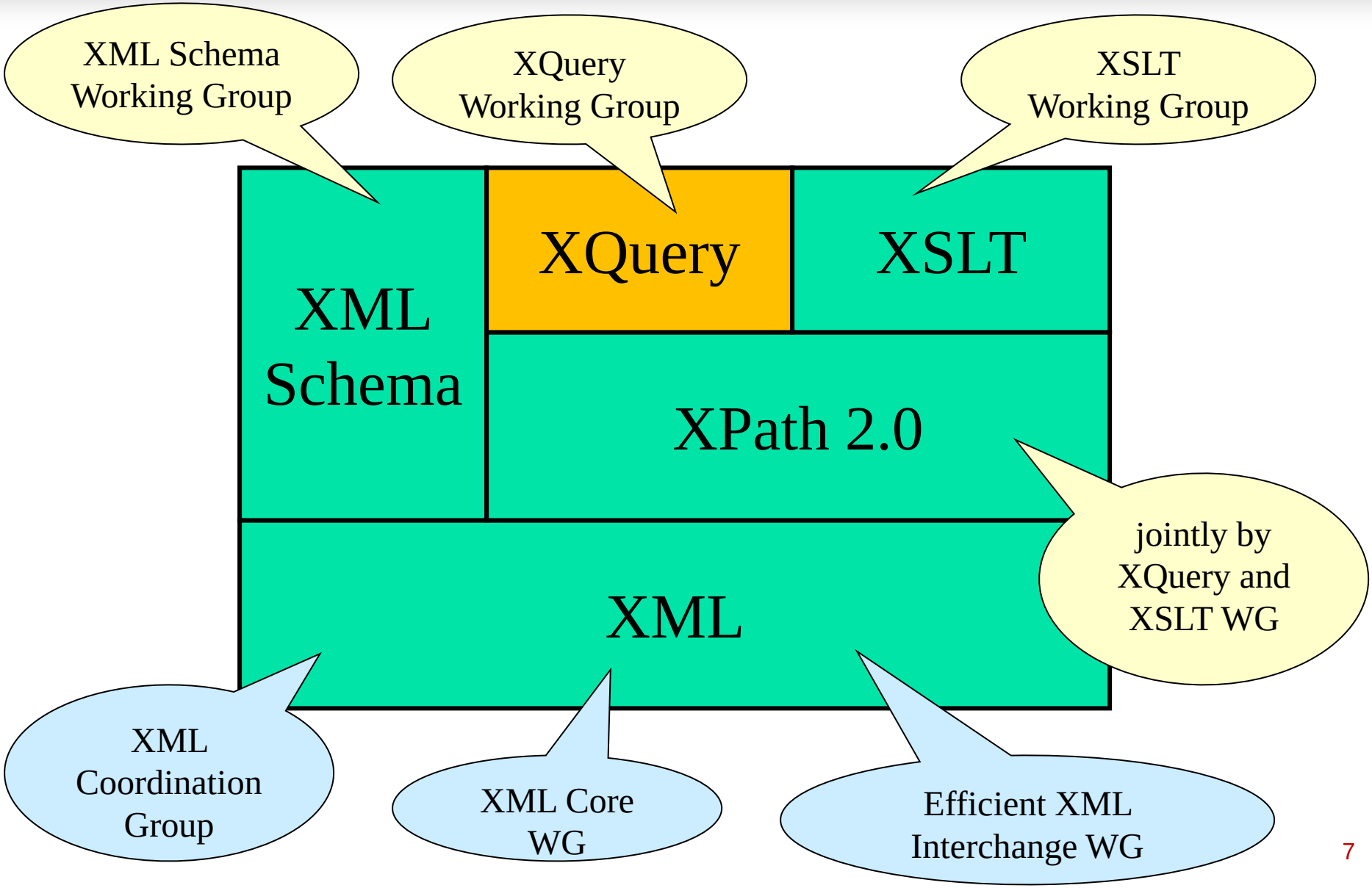
XML document

```
<?xml version="1.0"?>
<Courses>
    <Course>
        <Name>Introduction to Programming
    Languages</Name>
        <Code>CSE240</Code>
        <Level>Sophomore</Level>
        <Room>BYAC110</Room>
        <Cap>82</Cap>
    </Course>
    <Course>
        <Name>Distributed Software
    Development</Name>
        <Code>CSE445</Code>
        <Level>Senior</Level>
        <Room>BYAC210</Room>
        <Cap>40</Cap>
    </Course>
</Courses>
```

File Stored



XQuery is based on Other XML Technologies



Getting Started with XQuery

- XQuery is a **functional programming language**
  - Example: LISP, Scheme (CSE240 and CSE340, AI courses)
  - Every statement evaluates to a result and returns a value

    let $x := 3 let $y := 7 return 8*$x+4*$y

    which returns 52
  - Can we rewrite the code as follows 

    let $x := 3 let $y := 7 $x = 8*$x+4*$y return $x
  - No variables can be used ($x and $y are not variables)
- Primitive types in XQuery
  - number,
  - boolean,
  - strings,
  - dates,
  - times,
  - durations, and
  - XML types

Creating Nodes in XQuery

- Various XQuery functions can create or return nodes
  - Document function reads an XML file

    doc("http://venus.sod.asu.edu/WSRepository/XML/Courses.xml")

    or

    doc("Courses.xml")
  - Element constructor: a pair of tags, forms an element constructor and creates a node:

    <foo><bar>Courses</bar></foo>

XQuery and Path Expressions

- Use curly braces to embed XQuery expressions inside an element constructor

  ```
  Let $g := doc("Courses.xml")
  return <result>{$g/Courses/Course}</result>
  ```

- /Course selects the child elements named Courses
- /Courses/Course selects the Course elements
- //Course

  returns all Course elements that appear anywhere in the document
- //Course[Level = “400”]

  returns all Course elements with Level = “400”
- //Room[@image = “layout210.JPEG”]

  returns all Room elements with attribute image = “layout210.jpeg”
- //Course/(Level | Code)

  returns all Level or Code elements from current node

Even though XQuery shares the syntax with XPath, it intends to perform the query in XML database, instead of in client program.

Programming the Queries in XQuery

Similar to SQL syntax

```
<html>
{
    let $d := doc("Courses.xml")/Courses
    for $b in $d/Course
        where $b/Level > 200
        order by $b/Code
        return <Course>
                    {$b/Name, $b/Level, $b/Cap}
                </Course>
}
</html>
```

from

where

orderby

select

Native XML Database Products: Current Status

- IBM DB2 9.5 PureXML DB (Commercial)
  http://www.ibm.com/developerworks/data/library/techarticle/dm-0710nicola/
- Oracle XML DB (Commercial), 9i, 10i, 11g, and later
  https://www.oracle.com/database/technologies/appdev/xmldb.html
- Microsoft SQLXML 4.0 (Commercial)
  https://learn.microsoft.com/en-us/sql/relational-databases/sqlxml/sqlxml-4-0-programming-concepts?view=sql-server-ver16
- dbXML (Open Source)
  http://www.dbxml.com/
- eXist (Open Source)
  http://exist.sourceforge.net/
- Apache Xindice (Discontinued)
  http://xml.apache.org/xindice/

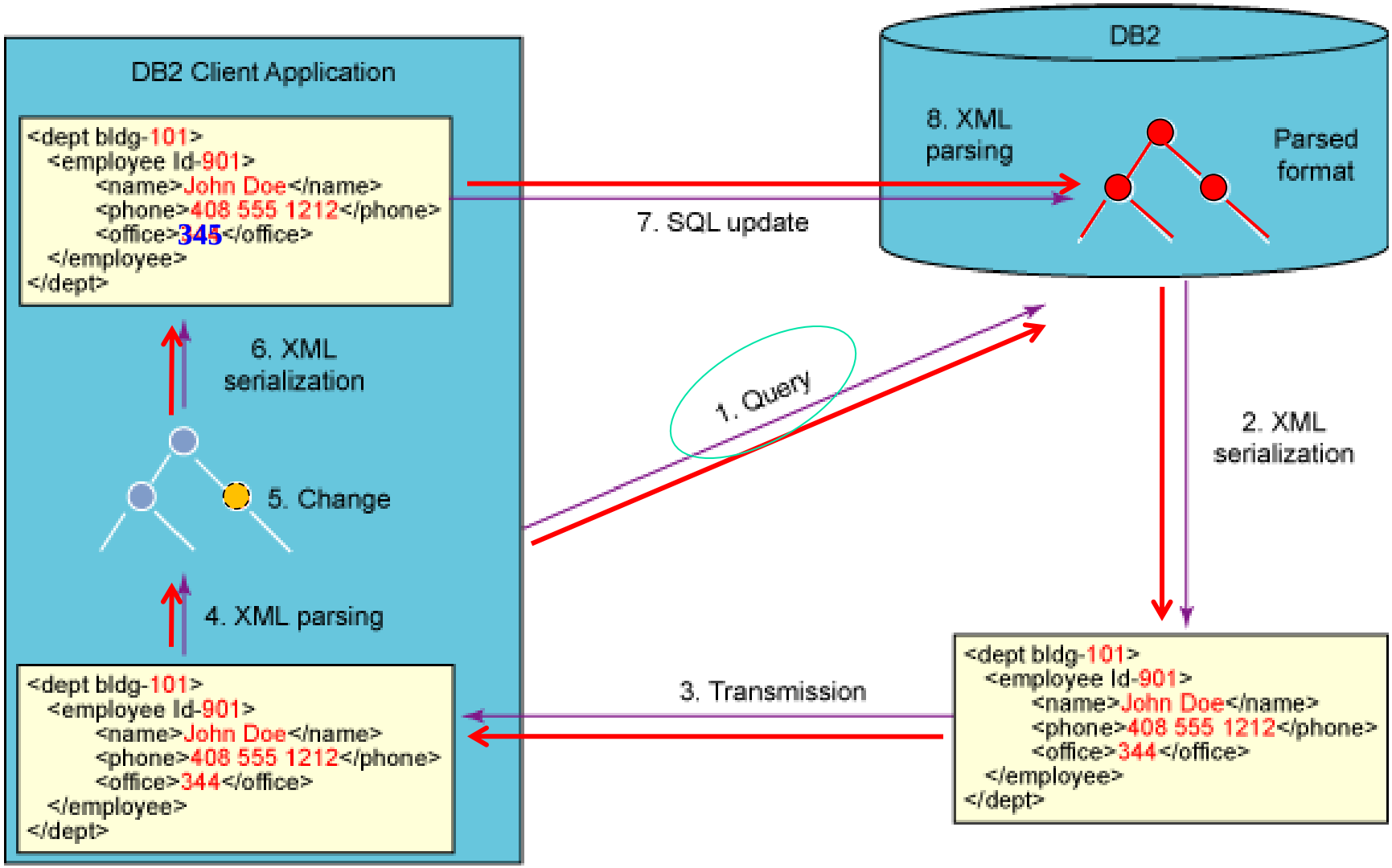
Example: Oracle XQuery: Retrieve Info

```
SELECT xmlquery(
         '<POSummary lineItemCount="{count($XML/PurchaseOrder/LineItems/LineItem)}">
         {
            $XML/PurchaseOrder/User,
            $XML/PurchaseOrder/Requestor,
            $XML/PurchaseOrder/LineItems/LineItem[2]
          }
          </POSummary>'
         passing object_value AS "XML"
         returning content
       ).getclobval() initial_state
FROM   PURCHASEORDER
WHERE  xmlExists(
         '$XML/PurchaseOrder[Reference=$REF]'
          passing object_value AS "XML",
                  'ACABRIO-100PDT' AS "REF"
       )
/
```

Oracle: Deleting a node using XQuery update

```
UPDATE PURCHASEORDER
SET    object_value = XMLQuery
                      (
                        'copy $NEWXML := $XML modify (
                           for $PO in $NEWXML/PurchaseOrder return (
                                replace value of node $PO/User with $USERID,
                                replace value of node $PO/Requestor with $FULLNAME,
                                replace value of node $PO/LineItems/LineItem/Part[@Description=$OLDTITLE]
                               )
                         )
                        return $NEWXML'
                        passing object_value as "XML",
                        'KCHUNG' as "USERID",
                        'Kelly Chung' as "FULLNAME",
                        'Runaway' as "OLDTITLE",
                        'Runaway[Updated]' as "NEWTITLE"
                        returning content
                      )
WHERE xmlExists(
         '$XML/PurchaseOrder[Reference=$REF]/LineItems/LineItem/Part[@Description=$OLDTITLE]'
          passing object_value as "XML",
                  'ACABRIO-100PDT' as "REF",
                  'Runaway' as "OLDTITLE"
       )
/
```

Example: IBM DB2 XML Update

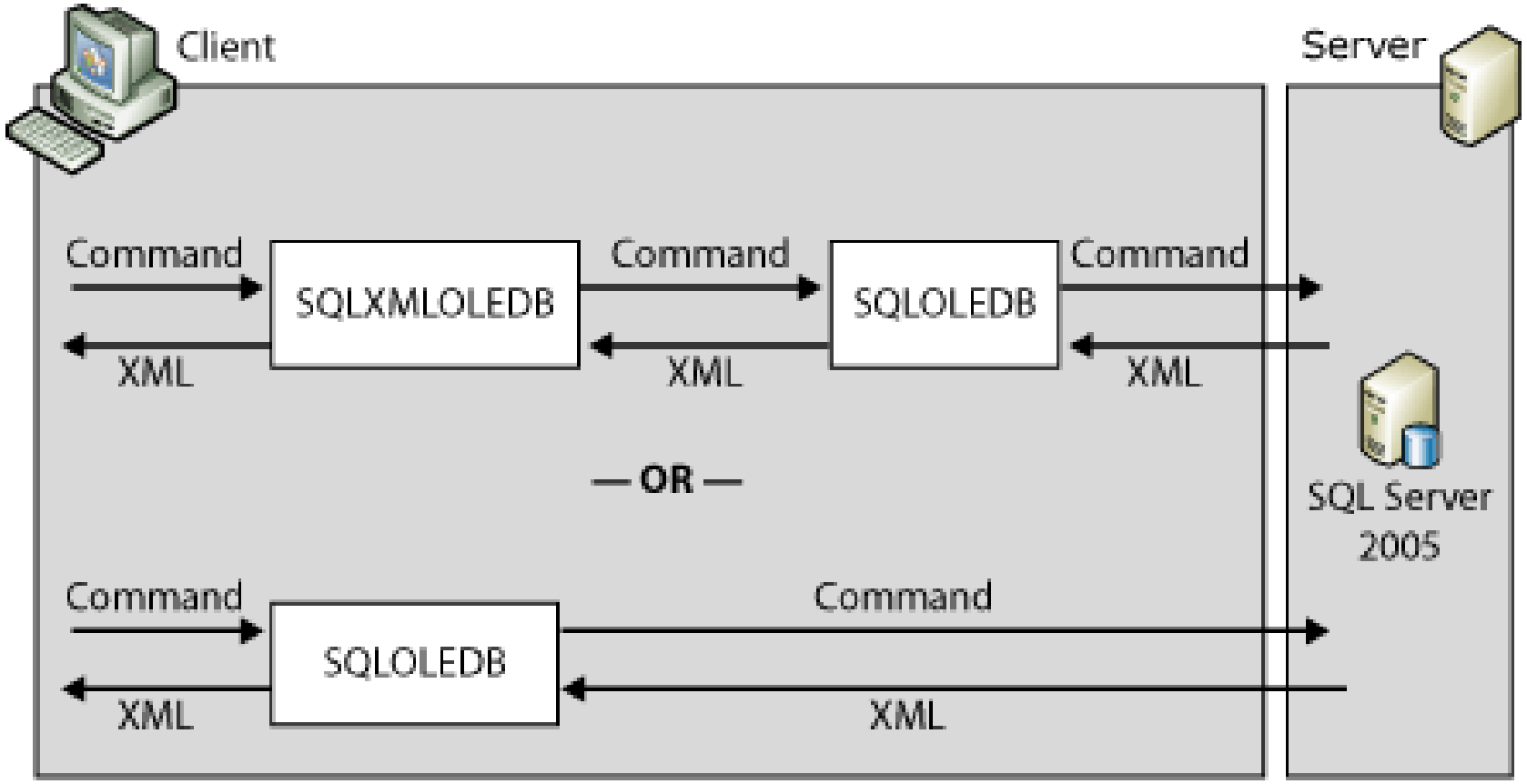


Steps of IBM DB2 XML Update

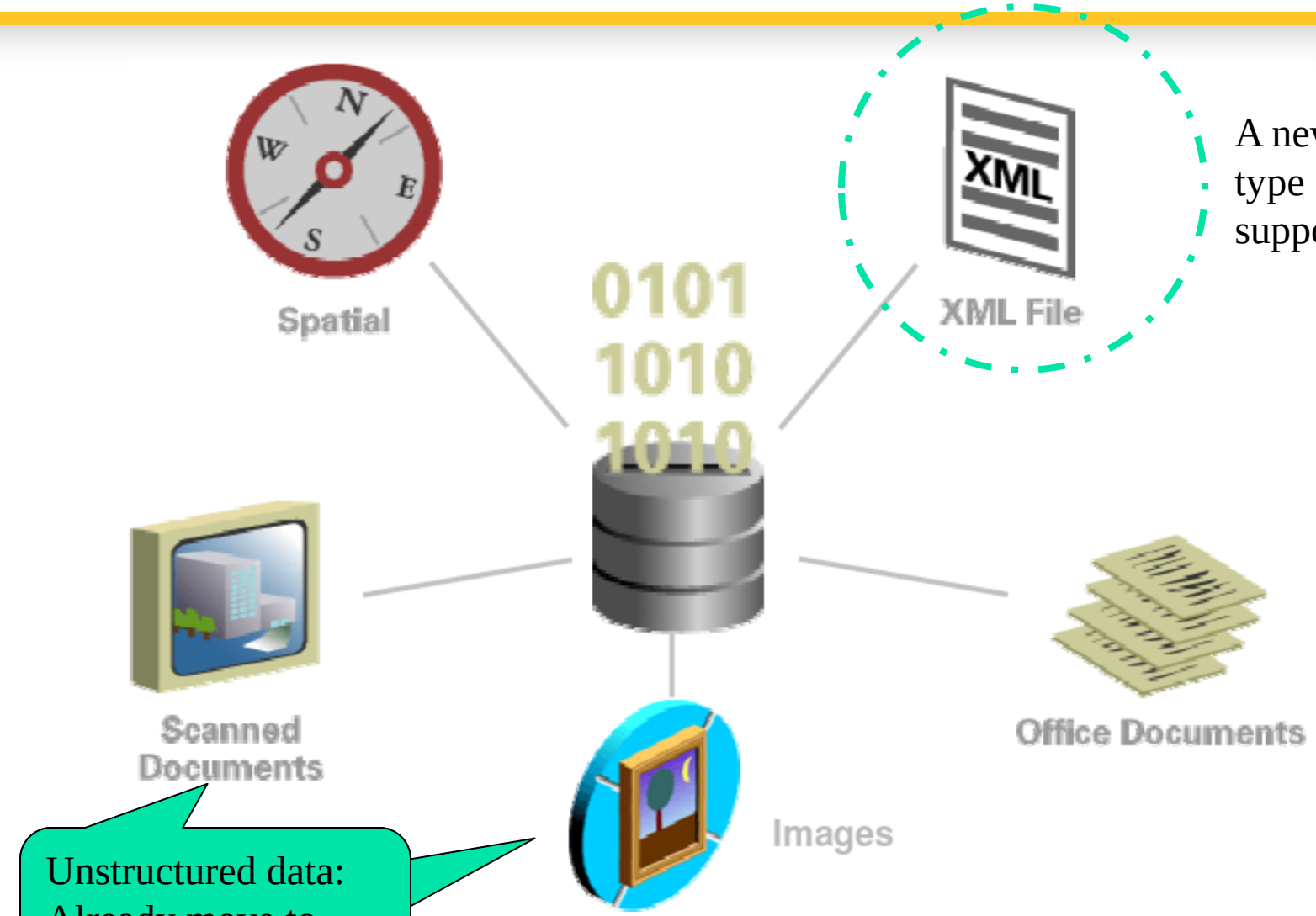
1. The application submits an SQL or XQuery statement to read an XML document.
2. The document is retrieved from the database and serialized to text format.
3. The document is transmitted to the client.
4. The application parses the XML document, typically using the document object model (DOM).
5. The application modifies **part** of the document using DOM APIs.
6. The document is re-serialized in the client application.
7. The application submits an SQL UPDATE statement and transmits the updated document to the database server.
8. The database server parses the updated XML document and replaces the **entire** old document.

Microsoft SQLXML 4.0 (2023)

- It primarily performs Client-side XML Formatting



Case Study:
Oracle DB Managing Many Types of Information



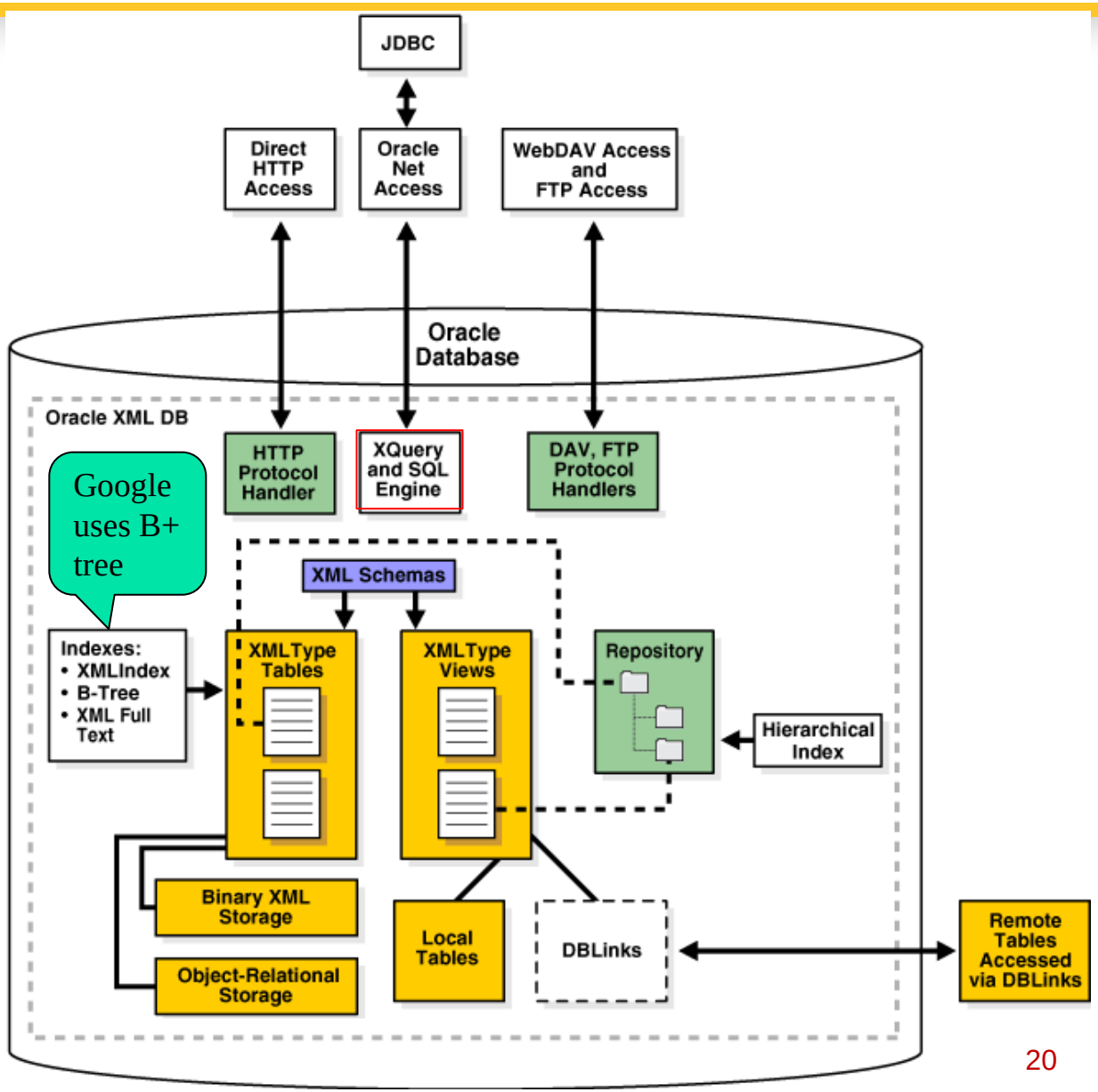
Oracle DB: Enhanced XML Type

https://docs.oracle.com/en/database/oracle/oracle-database/21/adxdb/intro-to-XML-DB.html#GUID-43B25825-FE37-4782-B627-8B2EFB44068D

- XML Schema Support
- Object-Relational Storage Maintaining DOM fidelity
- XML-specific memory management for better scalability and performance
- Built-in XML operators for SQL/XML interchangeability
- XPath Search in the server, and piecewise update of XML via XPath
- XSL Transforms in the server
- Enhanced XML Views for creating your own efficient representations of XML

Oracle XML DB Organization and Flow

- The XML Database is not a simple XML File
- It contains schema for transforming each XML file into a binary file, and store different elements in different places to enable partial updating
- Different indices are generated and stored, which speed up search: Not use sequential search.




Ira A. Fulton Schools of
Engineering
Arizona State University

Google Protocol Buffer and Big Table

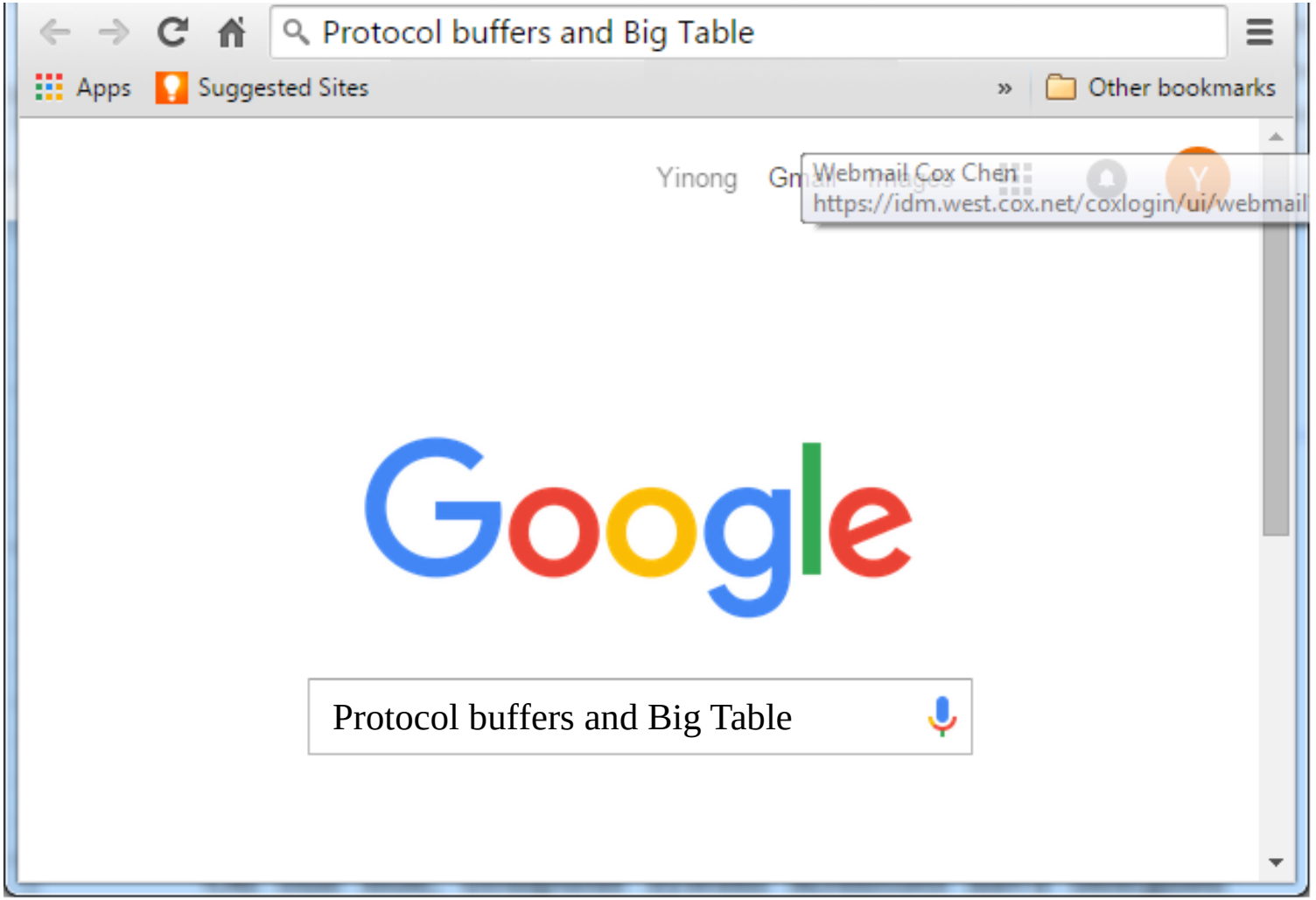
Ira A. Fulton Schools of Engineering
Arizona State University

Lecture Outline

- Google Protocol Buffers
- Big Table

Google Data Description and Management

Googol



How can Google be so fast?

- Distributed and Parallel Computing: Buildings of machines
- Crawling and Indexing
  Working 24/7
- Caching
  Buffering the frequently and recently used links and pages
- Specifically designed service-oriented computing model: Removing SOAP, using HTTP directly: RESTful service, instead of WSDL/SOAP services
- Specifically designed data structures and data management

Google Data Description Language?

- What are the data description languages used by Google? XML and XML Schema?
  - XML is universal and standard for representing data on Web;
  - XML may be slow for Google's amount of Data;
  - Think about the amount of information stored in Google directories – Any efficiency improvement will be significant;
  - A good design needs compromise / tradeoff between flexibility and efficiency

Google Protocol Buffers

http://code.google.com/apis/protocolbuffers/docs/overview.html

Compared to XML and XML Schema, Protocol buffers:

- are simpler
- are 3 to 10 times smaller in presenting the same information
- are 20 to 100 times faster in processing
- are less ambiguous
- generate data structures that are similar to object-oriented classes and thus are easier to use programmatically
- are less flexible/adaptable

Google Protocol Buffers Example

```
message Person {
    required string name = 1;
    required int32 id = 2;
    optional string email = 3;
    enum PhoneType {
        WORK = 0;
        HOME = 1;
        MOBILE = 2;
    }
    message PhoneNumber {
        required string number = 1;
        optional PhoneType type = 2 [default = WORK];
    }
    repeated PhoneNumber phone = 4;
}
```

A Similar Description in XML Schema

```
<element name = Person>
        <complexType>
          <sequence>
                <element name = "Id" type="integer" minOccurs="1" maxOccurs="1"/>
                <element name = "Name" type="string" minOccurs="1" maxOccurs="1"/>
                <element name = "Email" type="string" minOccurs="0" maxOccurs="1"/>
                <element name = "Phone" type="string" minOccurs="1" maxOccurs="1"/>
          </sequence>
        </complexType>
</element>
<element name = "Phone" >
        <simpleType>
          <restriction base = "string" >
                <enumeration value = "Work" type=minOccurs="1" maxOccurs="1"/>
                <enumeration value = "Home" minOccurs="0" maxOccurs="1"/>
                <enumeration value =  "Mobile" minOccurs="0" maxOccurs="1"/>
          </restriction>
</element >
```

Using Protocol Buffer for Service Development: gRPC

- https://cloud.google.com/endpoints/docs/grpc/about-grpc
- https://www.grpc.io/docs/

- gRPC is a high performance, open-source universal RPC framework, developed by Google.
- In gRPC, a client application can directly call methods on a server application on a different machine as if it was a local object.
- By default, gRPC uses protocol buffers as the Interface Definition Language (IDL).
- gRPC exposes endpoints (APIs) to the outside to enable different types of clients.
  - gRPC client
  - HTTP / JSON clients
- Visual Studio supports the development of gRPC services and clients.

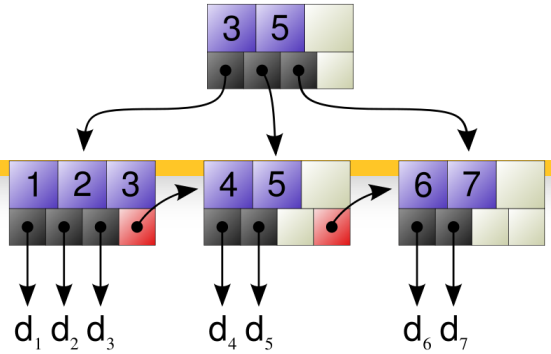
Google BigTable

http://en.wikipedia.org/wiki/BigTable

- **BigTable** is a fast and extremely large-scale database management system;
- It is a compressed, high performance, and proprietary database system built on Google File System (GFS);
- It departs from the convention of relational database, with a fixed number of columns;
- The database is “a sparse, distributed multi-dimensional sorted map”.
- What is it?
- The idea is similar to B Tree and B+ Tree that allow for efficient insertion, retrieval and removal of nodes.

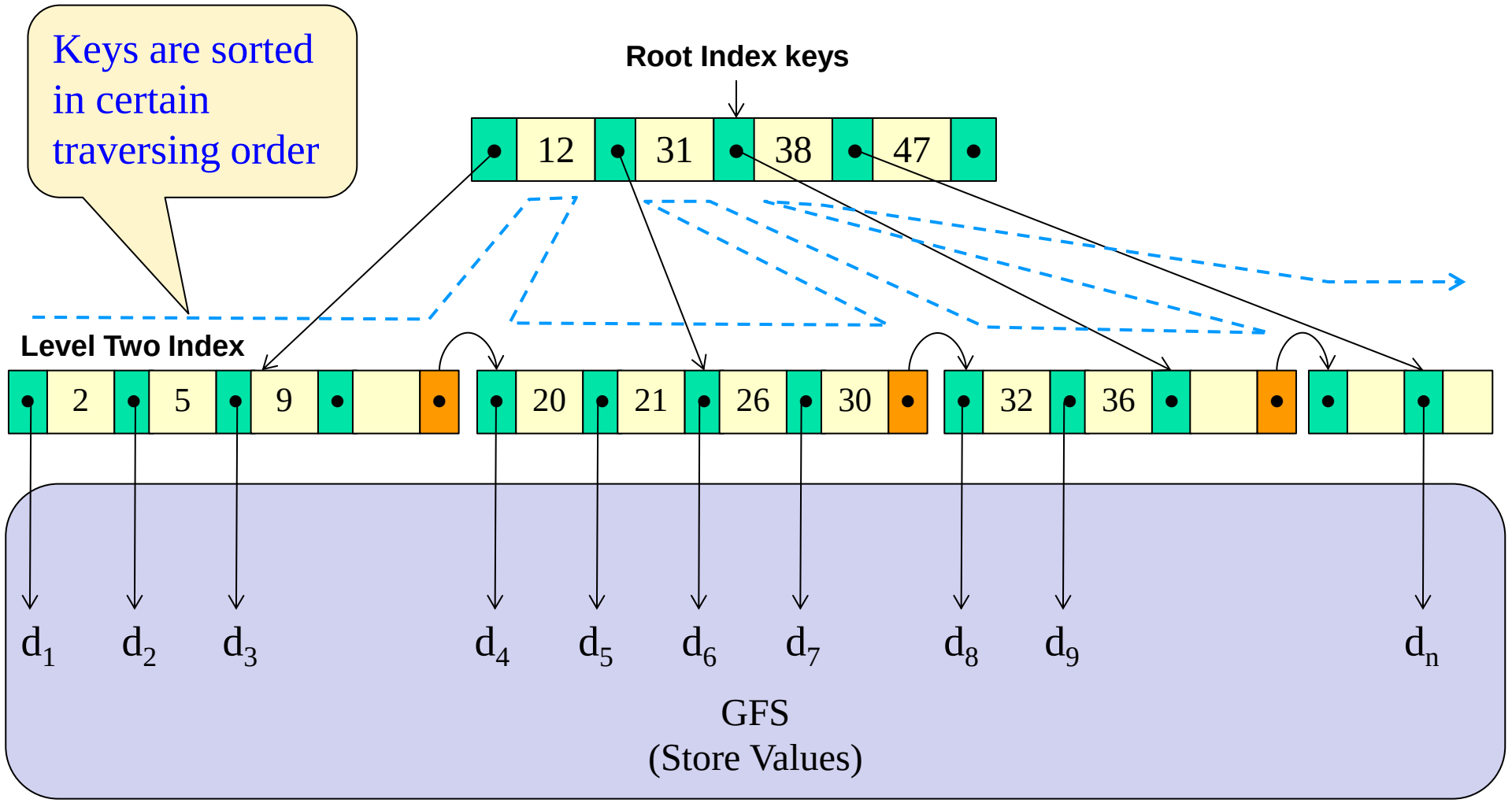
B Tree and B+ Tree

http://en.wikipedia.org/wiki/B%2B_tree



- It extends the binary search tree.
- It is a tree representing sorted data
  in a way that allows for efficient insertion, retrieval and removal of records, each of which is identified by a *key*: key-value database.
- It is a dynamic, multilevel index, with maximum and minimum bounds on the number of keys in each index node.
- In a B+ tree, in contrast to a B Tree, all records (values) are stored at the lowest level of the tree (leaves). Only keys are stored in interior blocks

+ Tree-Based BigTable (Key-Value Store)




Ira A. Fulton Schools of
Engineering
Arizona State University

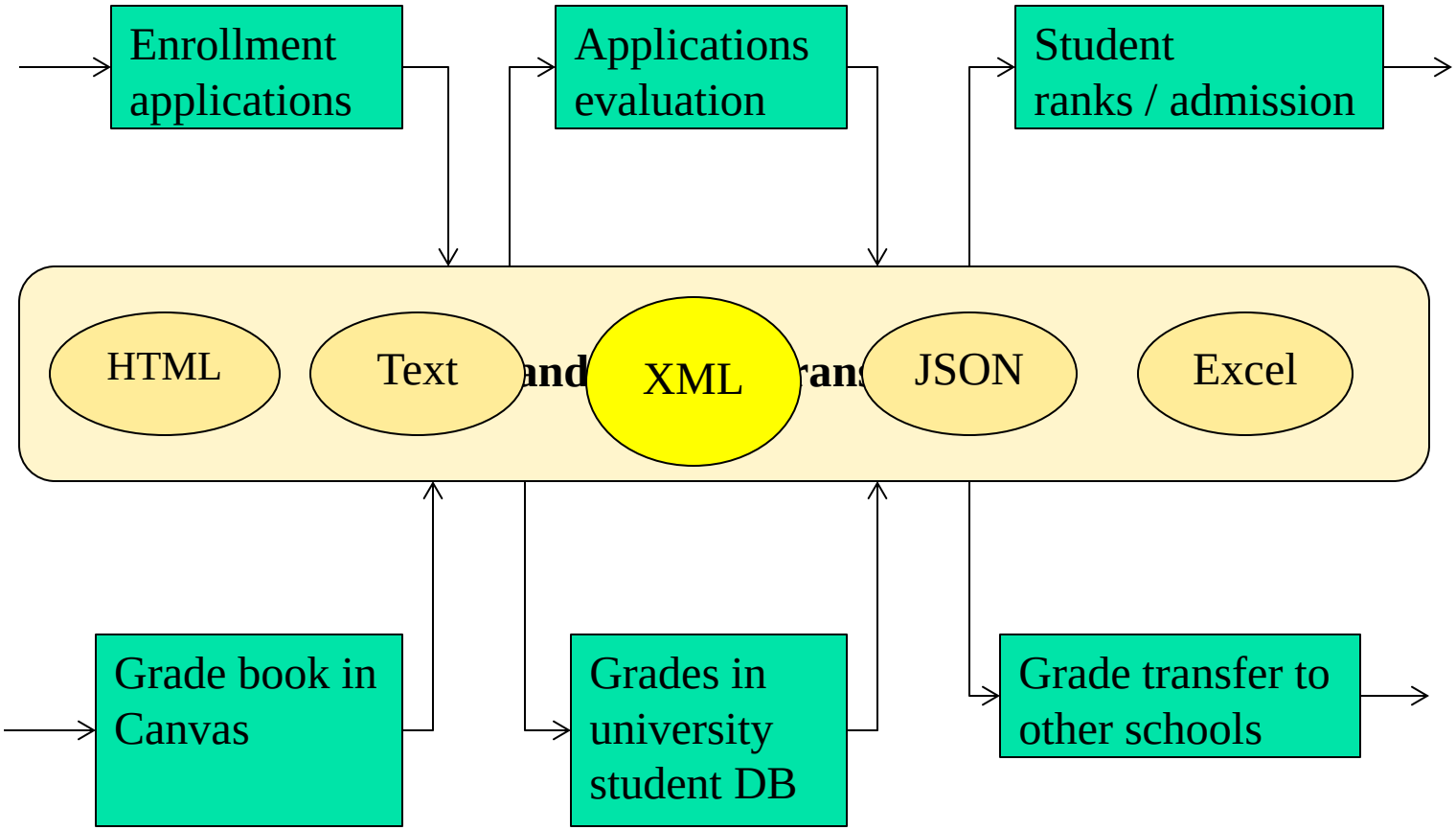
M10 L3
Feed

Ira A. Fulton Schools of Engineering
Arizona State University

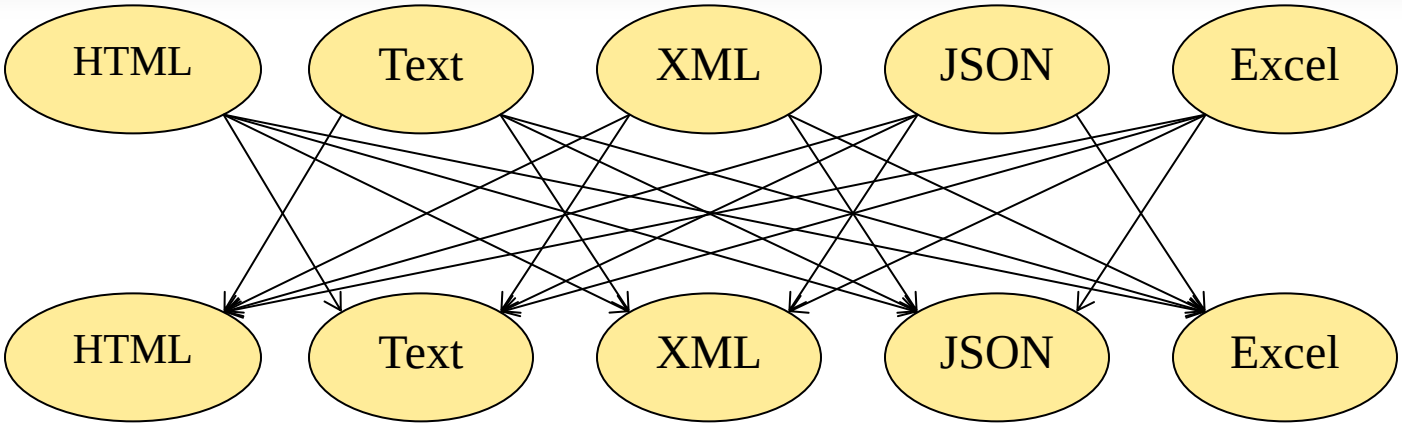
Lecture Outline

- Web Data Representations
- Feed
  - RSS
  - Atom

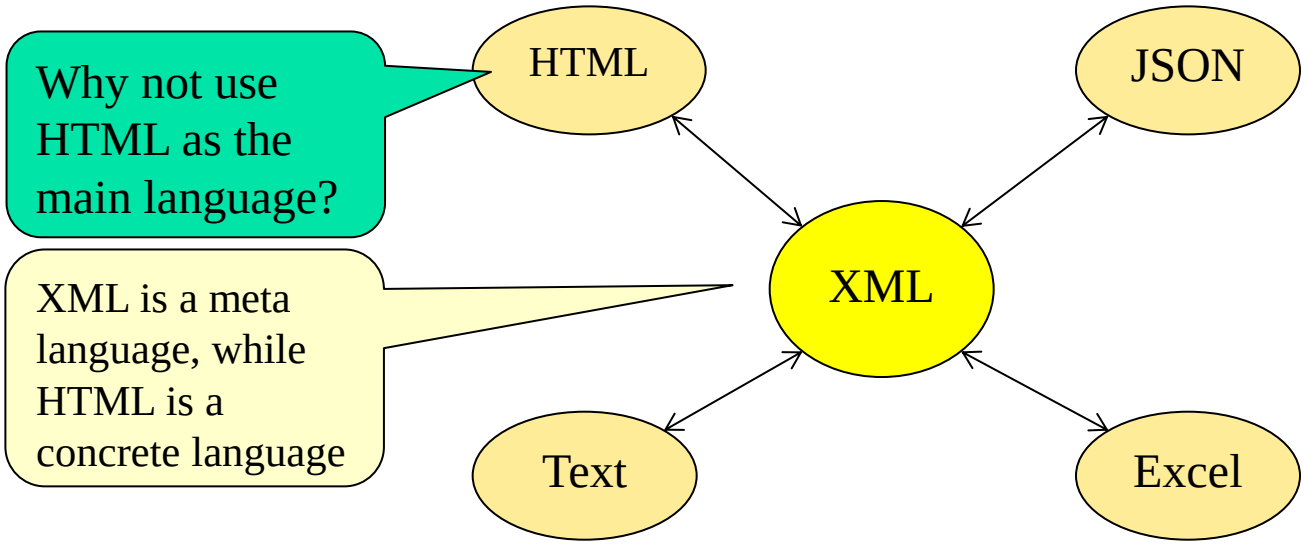
Example: A University Enterprise System



How Do You Implement Conversion?

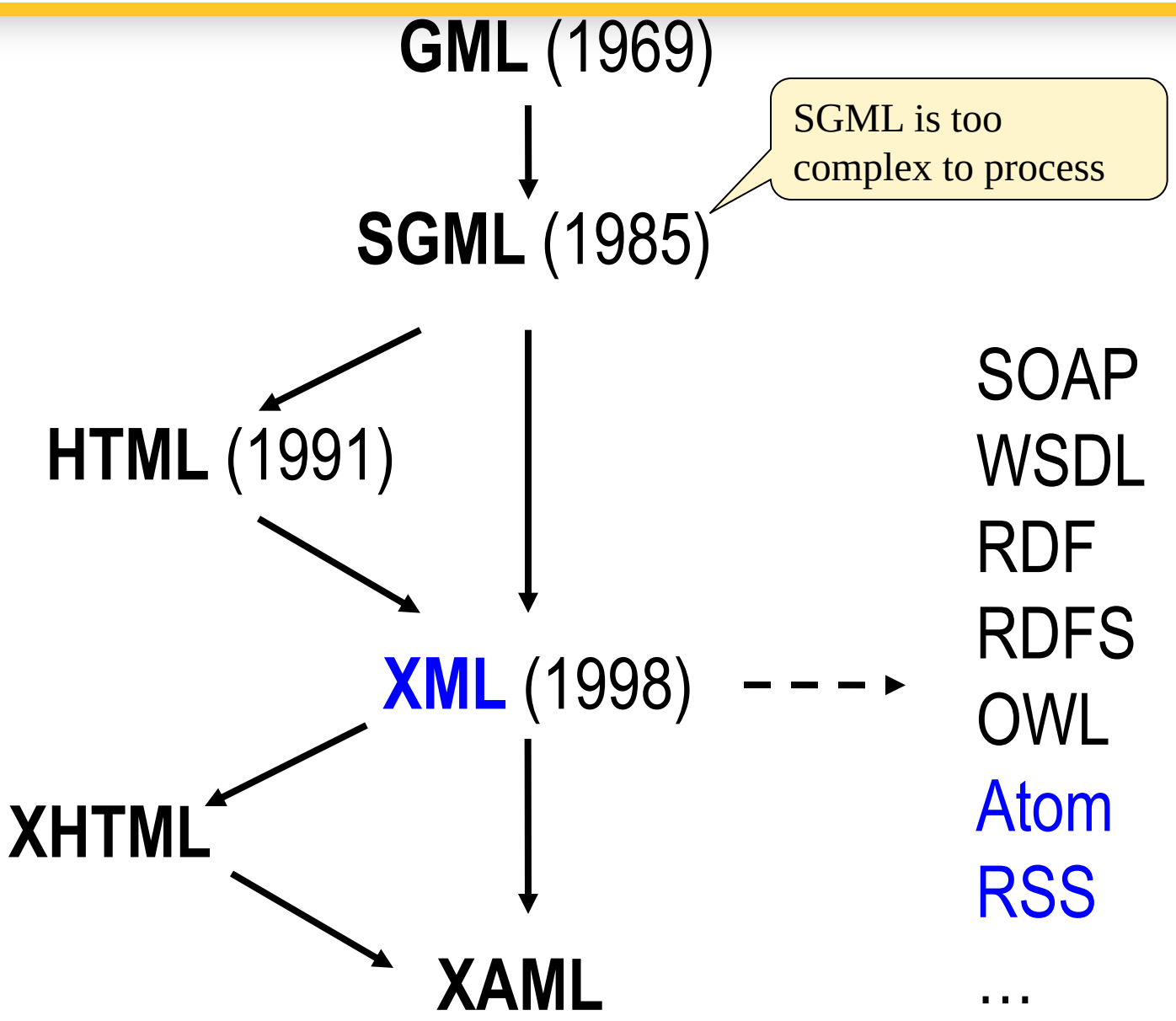


20 data converters



8 data converters

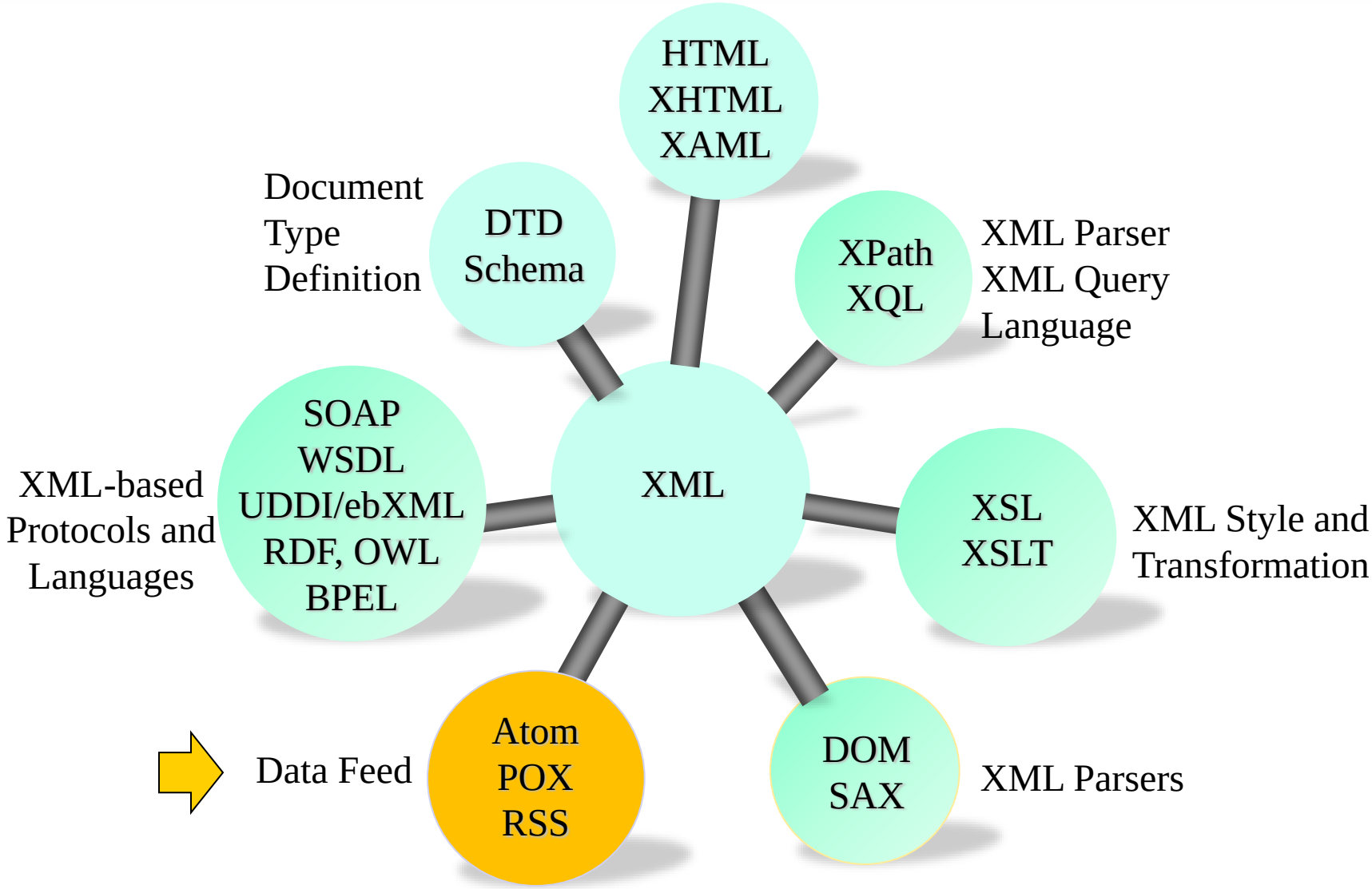
XML and Development



Web Data Representations

Web formats	Description
XML	The premier format for defining data, protocol, and languages
HTML	The traditional format for representing Web data and format
XHTML	Extended HTML 4.01 to conform with the XML format
RSS	RSS (Really Simple Syndication) for feed readers and Web blogs
Atom	Atom extends RSS, and it is also used for representing feeds for feed readers and blog publishing. It has been used in wider context, including the REST architecture.
POX	Plain-Old-XML is used for representing SOAP data, which does not need the header information for complex processing.
JSON	JavaScript Object Notation is efficient for representing data processed or to be processed by a program, such as JavaScript.
Protocol buffers	Google's Web data structure for search engine.
BigTable	Google's data structure for large database management.

XML Related Technologies (Review)



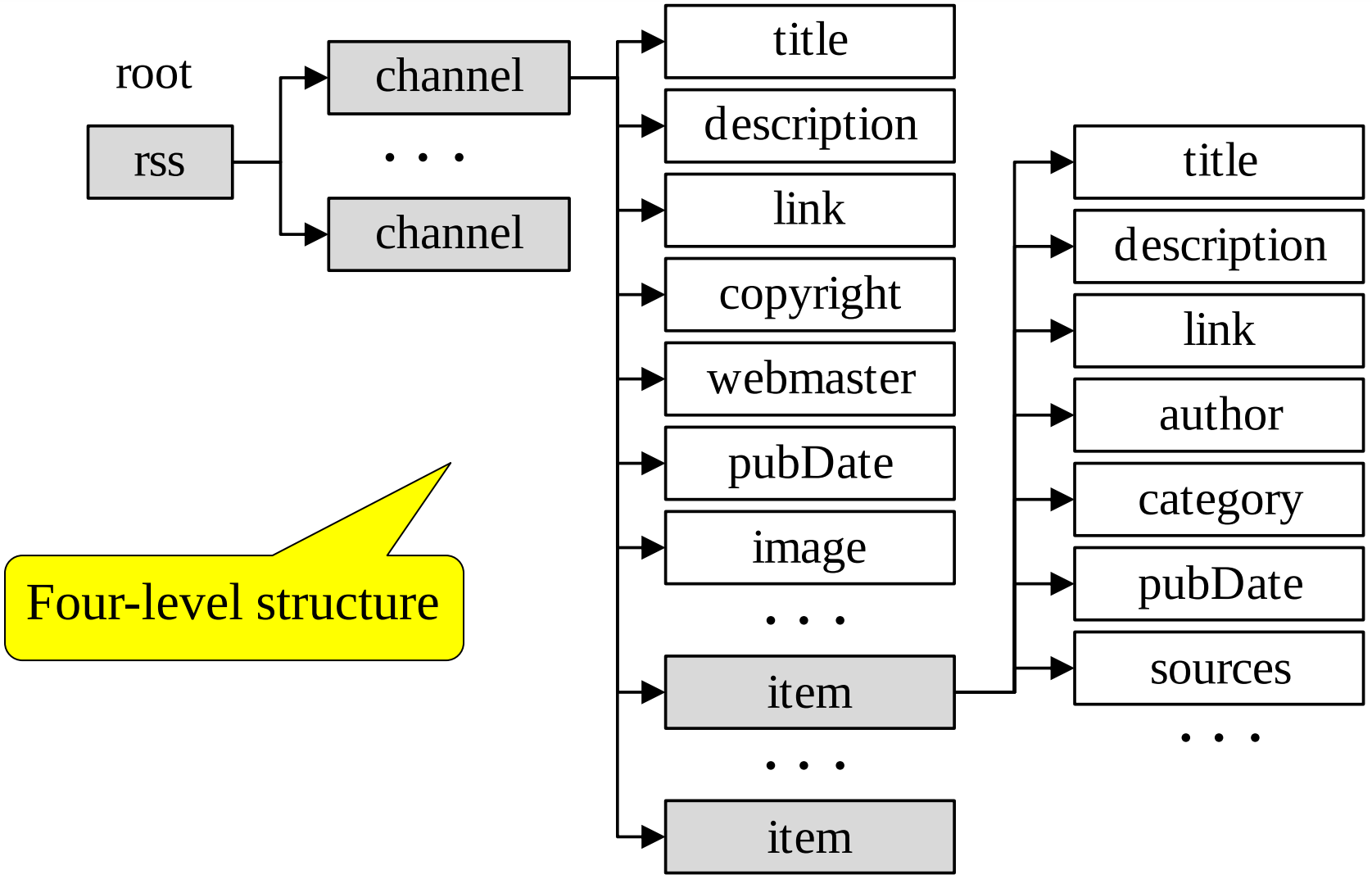
Why Other Data Formats?

- XML is flexible and can be used to define any data structure in a rooted tree.
- For any specific data structure, an XML schema or DTD is needed to define the structure.
- For convenience, a few predefined XML data structures are used, which do not need a schema file:
  - POX
  - RSS
  - ATOM
- They are widely used in Web page feed: Another Web page can conveniently read your Web data with structure, instead of as a string.

RSS: An XML-based Feed Data Structure

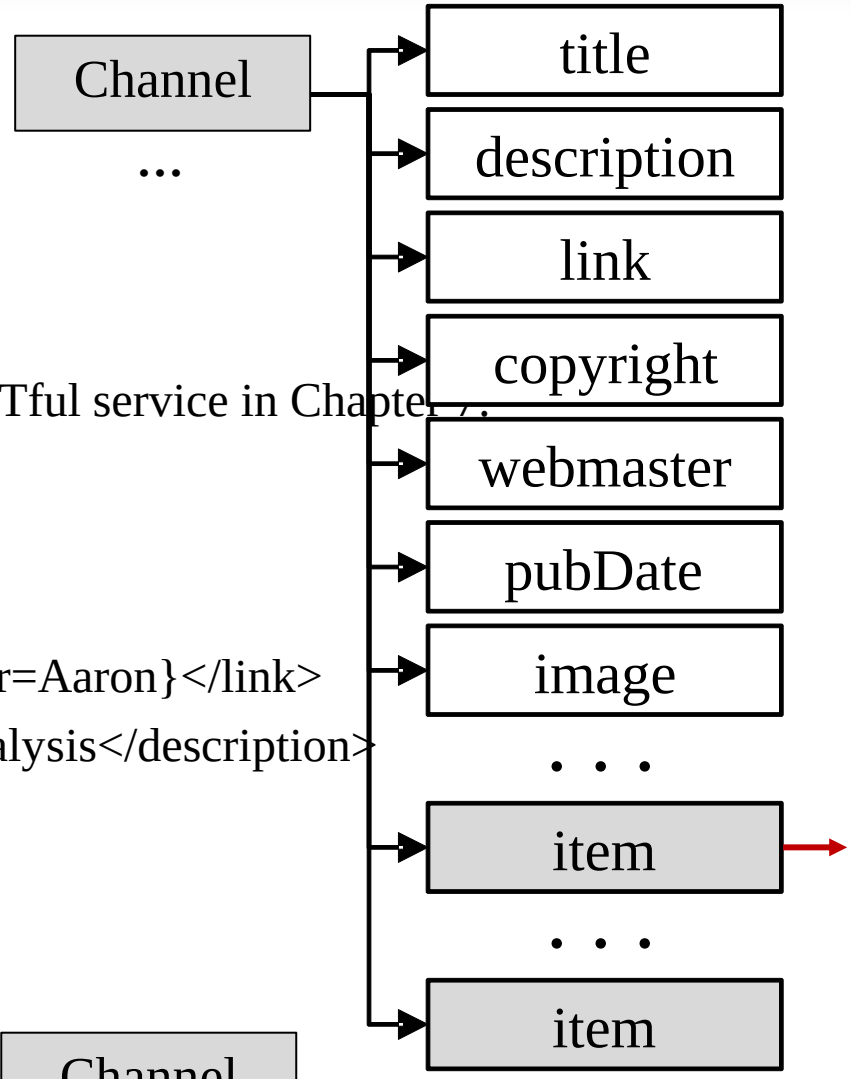
- A **feed** is a data structure that contains a list of items
- Each items is described by a few attributes, including the hyperlink and other information of the item, such as title, author, etc.
- RSS is a language used to describe syndicated feed data.
- A number of RSS versions exist.
- **RSS 1.0** stands for **RDF Site Summary**, where RDF (Resource Description Framework) is an ontology language widely used in semantic Web authoring.
- **RSS 2.0** stands for **Really Simple Syndication**. It is an XML document with a specific schema definition.

RSS Schema, with a four-layer structure



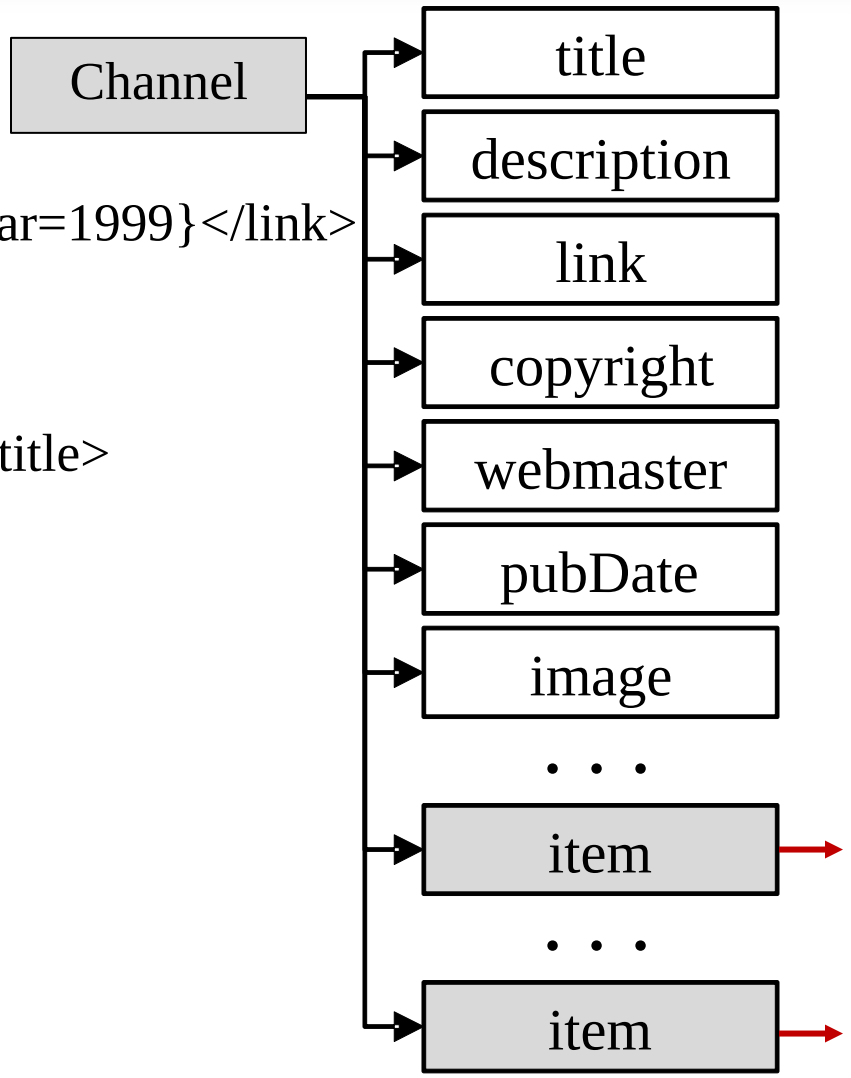
Example of RSS Document

```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
    <title>Computer Science Books</title>
    <link>http://cslibrary.asu.edu/</link>
    <description>
      This example will be further discussed in RESTful service in Chapter 7.
    </description>
    <item>
      <title>Operating Systems</title>
      <link>http://mylibrary.asu.edu/authors/{author=Aaron}</link>
      <description>Operating system design and analysis</description>
      <author>aaronauthor@asu.edu</author>
    </item>
    …
<channel>
    <title>Biology Science Books</title>
    …
```



Example of RSS Document (contd.)

```
<item>
   <title>Compilers</title>
   <link> http://cslibrary.asu.edu/years/{year=1999}</link>
 </item>
 <item>
   <title>Algorithm Analysis and Design</title>
   <author>zetaauthor@asu.edu</author>
 </item>
</channel>
<channel>
 <title>Biology Science Books</title>
 <link>http://biolibrary.asu.edu/</link>
  . . .
```



Example of RSS Document (contd.)

```
<channel>
   <title>My Books</title>
   <link>http://mylibrary.asu.edu/</link>
   <item>
      <title>Programming Languages</title>
      <link>http://www.kendallhunt.com/index.cfm</link>
      <description>
         Introduce different programming paradigms and program techniques
      </description>
      <author>yinong.chen@asu.edu</author>
   </item>
</channel>
</rss>
```

Find RSS URLs

https://about.fb.com/wp-content/uploads/2016/05/rss-urls-1.pdf

country category rss-url
US general http://feeds.abcnews.com/abcnews/usheadlines
US general http://rss.cnn.com/rss/cnn_topstories.rss
US general http://www.cbsnews.com/latest/rss/main
US general http://www.nytimes.com/services/xml/rss/nyt/National.xml
US general http://online.wsj.com/xml/rss/3_7085.xml

US politics http://feeds.foxnews.com/foxnews/politics
US politics http://feeds.washingtonpost.com/rss/rss_election-2012
US politics http://rss.nytimes.com/services/xml/rss/nyt/Politics.xml
US politics http://rss.nytimes.com/services/xml/rss/nyt/Upshot.xml
US politics http://thecaucus.blogs.nytimes.com/feed/

US business http://www.business-standard.com/rss/latest.rss
US business http://www.business-standard.com/rss/home_page_top_stories.rss
US business http://feeds.harvardbusiness.org/harvardbusiness?format=xml
US business http://au.ibtimes.com/rss/articles/region/1.rss
US business http://www.businessweek.com/search/rssfeed.htm
US business http://www.huffingtonpost.com/feeds/verticals/business/news.xml

CA business http://news.gc.ca/web/fd-en.do?mthd=thm&ft=atom&crtr.thm1D=17
CA business http://rss.radio-canada.ca/fils/nouvelles/economieaffaires.xml
CA business http://www.calgarysun.com/money/rss.xml
CA business http://www.huffingtonpost.ca/feeds/verticals/canada-business/index.xml
CA business http://rss.canoe.com/Money/home.xml
CA business http://www.itbusiness.ca/feed

GB general http://www.telegraph.co.uk/news/uknews/rss
GB general http://www.ft.com/rss/world/uk
GB general http://www.standard.co.uk/news/rss
GB general http://www.oxfordmail.co.uk/news/rss/
GB general http://www.dailymail.co.uk/news/index.rss

AU tech http://techau.com.au/feed/
AU tech http://www.techrepublic.com/rssfeeds/blog/australian-technology/
AU entertainment ://feeds.smh.com.au/rssheadlines/entertainment.xml
AU entertainment http://www.news.com.au/entertainment

CNN Top Stories RSS URL

http://rss.cnn.com/rss/cnn_topstories.rss


```
<rss xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:content="http://purl.org/rss/1.0/modules/content
  <channel>
    <title>
      <![CDATA[ CNN.com - RSS Channel - HP Hero ]]>
    </title>
    <description>
      <![CDATA[ CNN.com delivers up-to-the-minute news and information on the latest top stories, weathe
    </description>
    <link>https://www.cnn.com/index.html</link>
    <image>
      <url>http://i2.cdn.turner.com/cnn/2015/images/09/24/cnn.digital.png</url>
      <title>CNN.com - RSS Channel - HP Hero</title>
      <link>https://www.cnn.com/index.html</link>
    </image>
    <generator>coredev-bumblebee</generator>
    <lastBuildDate>Thu, 22 Aug 2024 15:19:24 GMT</lastBuildDate>
    <pubDate>Tue, 18 Apr 2023 21:25:59 GMT</pubDate>
    <copyright>
      <![CDATA[ Copyright (c) 2024 Turner Broadcasting System, Inc. All Rights Reserved. ]]>
    </copyright>
    <language>
      <![CDATA[ en-US ]]>
    </language>
    <ttl>10</ttl>
    <item>
      <title>
        <![CDATA[ Some on-air claims about Dominion Voting Systems were false, Fox News acknowledges in
      </title>
      <link>https://www.cnn.com/business/live-news/fox-news-dominion-trial-04-18-23/index.html</link>
      <guid isPermaLink="true">https://www.cnn.com/business/live-news/fox-news-dominion-trial-04-18-23/
```

Processing RSS Feed

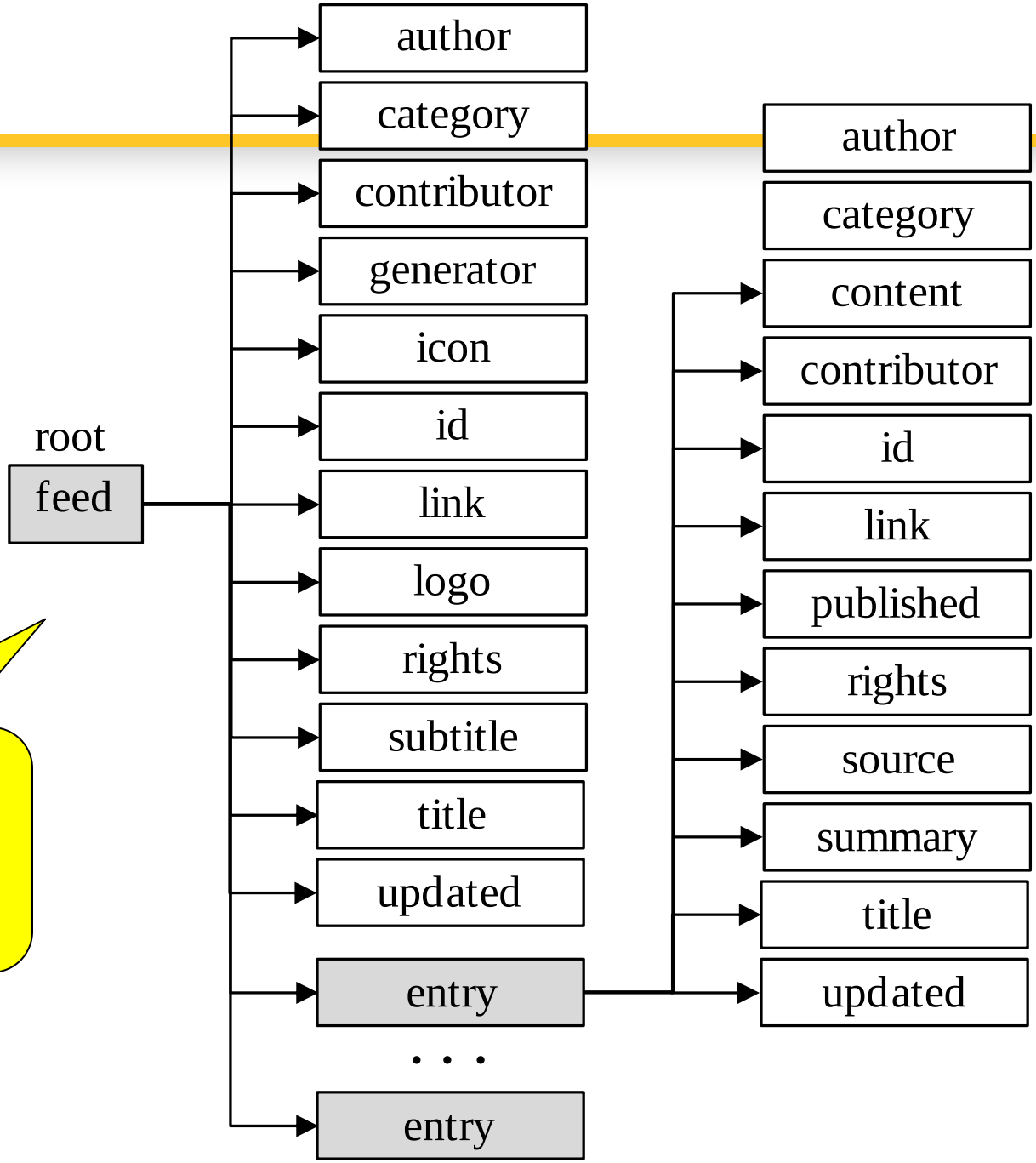
https://learn.microsoft.com/en-us/dotnet/api/system.servicemodel.syndication.syndicationfeed?view=net-8.0

```
string url = " https://justinpot.com/feed/ ";
    XmlReader reader = XmlReader.Create(url);
    SyndicationFeed feed = SyndicationFeed.Load(reader);
    reader.Close();
    foreach (SyndicationItem item in feed.Items)
    {
        String title = item.Title.Text;
        Console.WriteLine(title);
        String summary = item.Summary.Text;
        Console.WriteLine(summary);
        String summary = item.Summary.Text;
        ...
    }
```

Atom

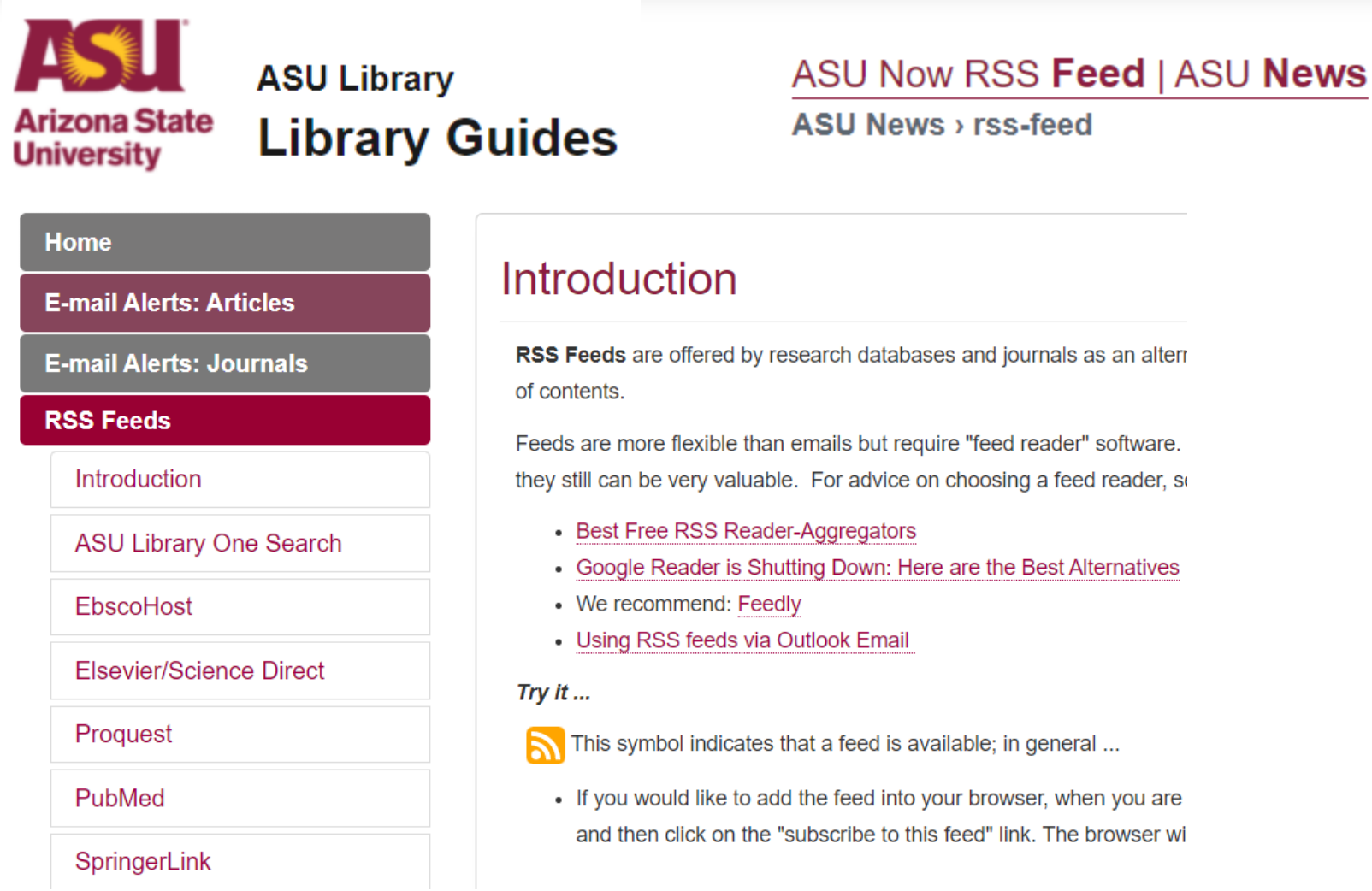
- RSS 2.0 widely used for representing feed data for its simplicity.
- It is a really simple syndication, which does not have many features that today's feed data desired to have.
- There are frequent reports with the interoperability problems with different feed readers and protocols.
- **Atom** is a more recent language designed for describing feed data with more features and with improved structure.

Atom Schema, with a three-layer structure



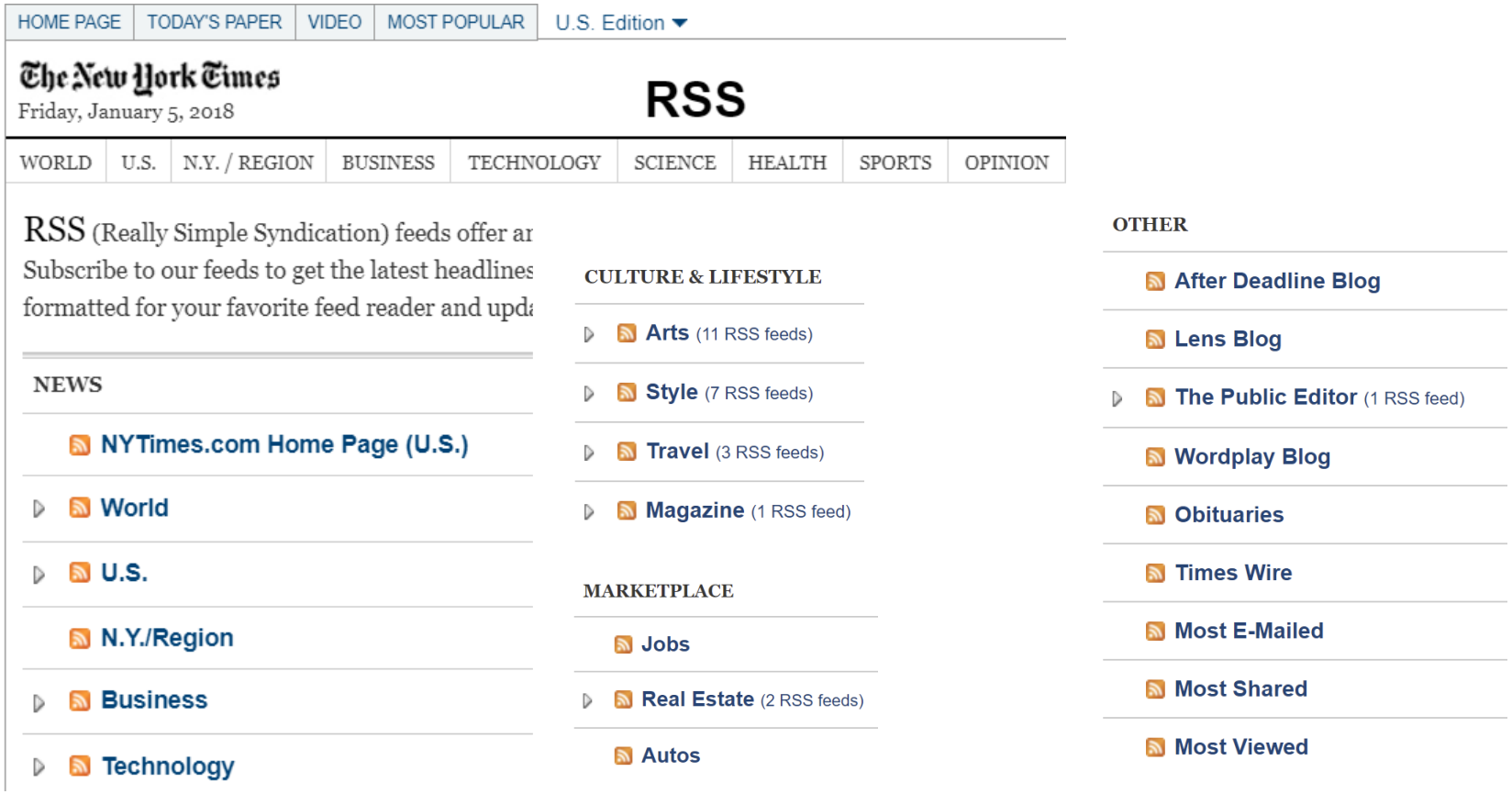
Three-level structure, but with more elements

Feeds Applications: ASU Feed



Feeds Applications: New York Times

- RSS feeds offer another way to get NYTimes.com content.
- You can get the latest headlines, summaries and links back to full articles - formatted for your favorite feed reader and updated throughout the day.




Ira A. Fulton Schools of
Engineering
Arizona State University

M10 L4
JSON

Ira A. Fulton Schools of Engineering
Arizona State University

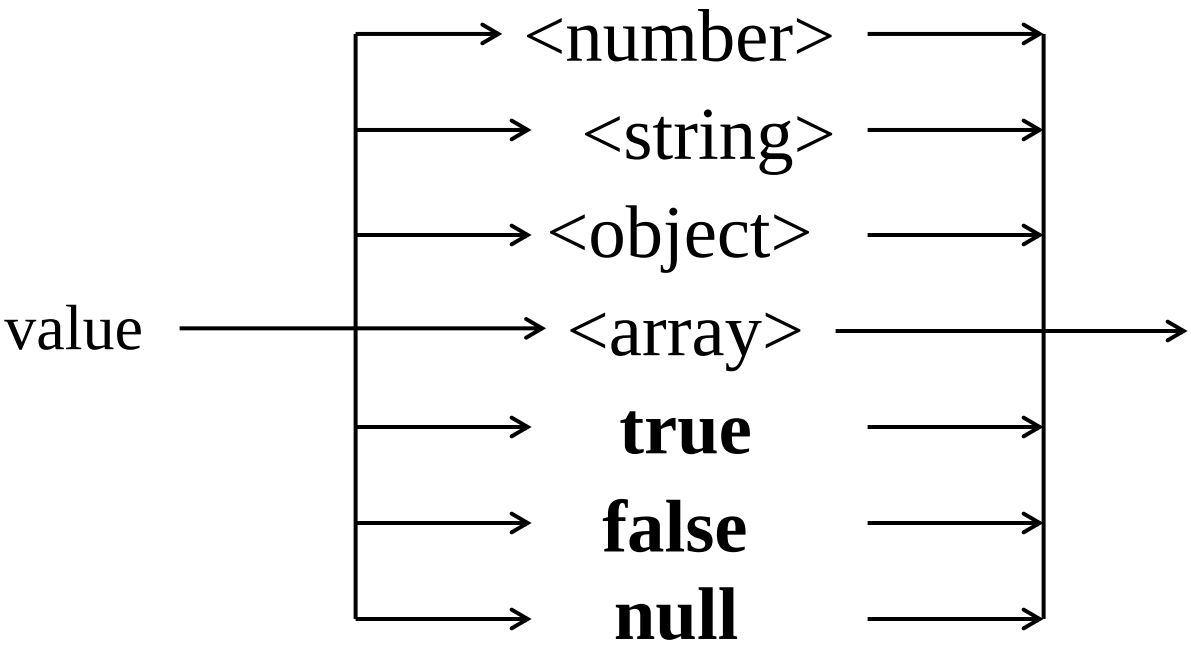
Lecture Outline

- JSON Data Definition
- JSON vs. XML
- JSON vs. Dictionary
- JSON Applications

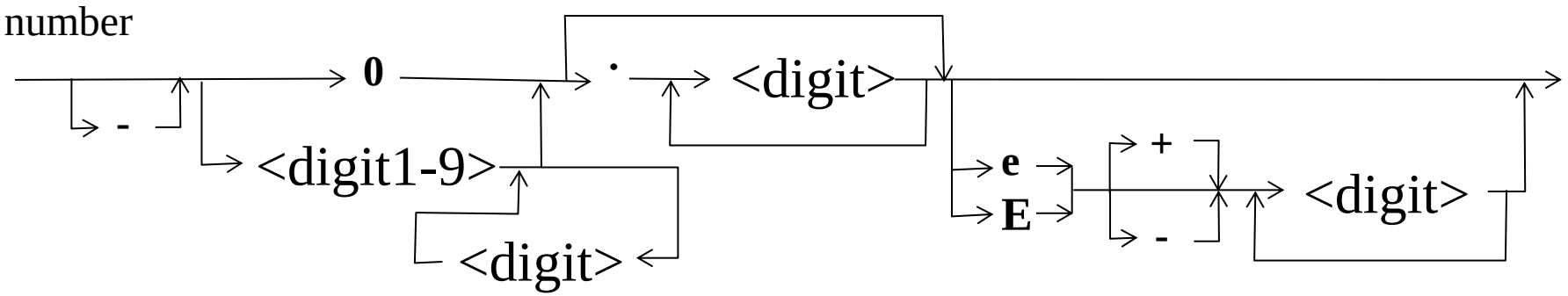
JSON (JavaScript Object Notation)

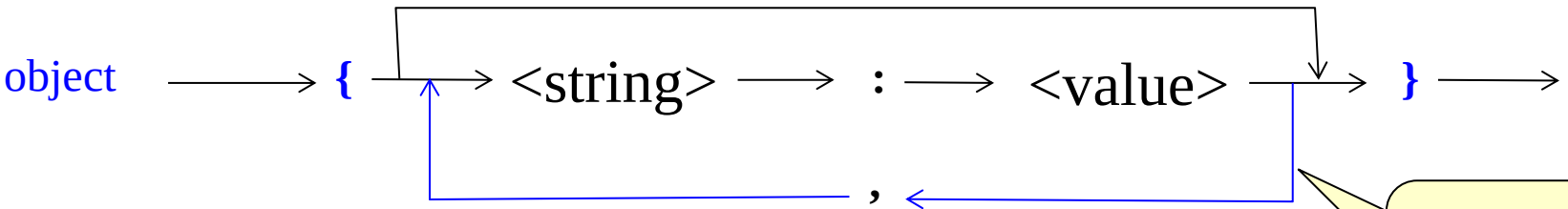
- JSON is a light-weight alternative to XML for simple data-interchange
- JSON is simpler than XML and more compact
  - JSON uses no tags, and it uses braces instead, like programming language
  - XML parsing is harder because of its complexity
- JSON has fixed schema (structure definition), and it is not as extensible as XML
- Preferred for simple data exchange by many
- Transforms to other formats? Find library or write your own.

JSON (JavaScript Object Notation) Data Definition



JSON Number and Object Format



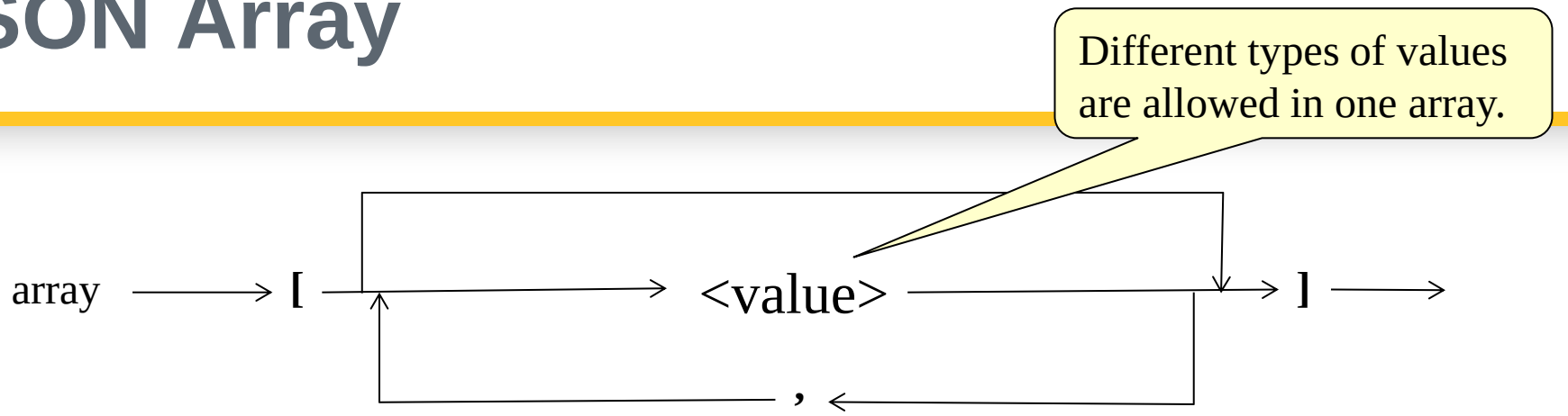


A list of string-value pairs

Object Example

```
{"name":"John Doe", "age":25,
"married":true, "University":"ASU",
"Graduated":false,
"Courses":{"CSE240":200,"CSE310":300,
"CSE445":400, "GPA":3.75}}
```

JSON Array



Array of string

```
["John Doe", "Marry", "Smith"]
```

Array of mixed values

```
[{"John Doe":25}, {"Marry":30}, "Smith", 20, true]
```

Array of arrays

```
[
  [1, 2, 3, 4],
  [2, 3, 4, 5],
  [5, 2, 3, 5]
]
```

```
for(var i=0;i<myArray.length;i++)
{
    console.log(myArray[i].name,
                myArray[i].value);
}
```

XML vs. JSON

```
<studentInfo>
   <students>
      <student>
         <name>John Doe</name>
         <phone>4801234567</phone>
      <student>
      <student>
         <name>Mary Smith</name>
         <phone>6022345678</phone>
      <student>
      <student>
         <name>April Lee</name>
         <phone>6233456789</phone>
      <student>
   </students>
</studentInfo>
```

```
{ "studentInfo" :
   { "students" :
      [
         {
            "name" : "John Doe",
            "phone" : 4801234567
         },
         {
            "name" : "Mary Smith",
            "phone" : 6022345678
         },
         {
            "name" : "April Lee",
            "phone" : 6233456789
         },
      ]
   }
}
```

object

object

array

object

JSON and Dictionary

A dictionary is similar to an array. In an array, we use an integer as the index to its element.

- In a dictionary, an element is accessed using an index as well. However, it is not a number index, but a string (name) index.
- It represents a JSON object/

A Python dictionary is dynamic. It offers the following dynamic operations:

- Create(): Creates an empty dictionary.
- Insert(e): Adds element e into the dictionary
- Delete(e): Removes element e from the dictionary, if e exists.
- Lookup(e): Checks if the element e is in the dictionary. If it does, it returns the information associated with the element

Python Dictionary Example (Short)

```
zip = dict() # Create a dictionary structure
zip = {"Tempe": 85281, "Chandler": 85224, "Phoenix": 85003, "Mesa": 85201, "Scottsdale": 85250}
print("zip is", zip)
print("Tempe zipcode is ", zip["Tempe"])
print("Mesa zipcode is ", zip["Mesa"])
print("Mesa zipcode is ", zip.get("Mesa"))
print("mesa zipcode is ", zip.get("mesa")) # will not cause an error
print("tempe zipcode is ", zip.get("tempe")) # will not cause an error
print("Gilbert zipcode is ", zip.get("Gilbert")) # will not cause an error
print("tempe zipcode is ", zip["tempe"]) # will cause an error
```

```
zip is {'Tempe': 85281, 'Chandler': 85224, 'Phoenix': 85003, 'Mesa': 85201, 'Scottsdale': 85250}
Tempe zipcode is  85281
Mesa zipcode is  85201
Mesa zipcode is  85201
mesa zipcode is  None
tempe zipcode is  None
Gilbert zipcode is  None
```

Python Dictionary Example (Extended)

```
zipcode = dict() # Create a dictionary structure
zipcode["Tempe"] = [85281, 85282, 85283, 85284, 85287] # insert
zipcode["Chandler"] = [85224, 85225, 85226, 85226] # insert
zipcode["Phoenix"] = [85003, 85004, 85006, 85007, 85008, 85009] # insert
zipcode["Mesa"] = [85201, 85202, 85203, 85204, 85205, 85206, 85207]
# insert
zipcode["Scottsdale"] = [85250, 85251, 85253, 85254, 85255] # insert
cities = ["Tempe", "Mesa", "Gilbert", "Phoenix", "Scottsdale",
"Chandler"]
# Continue on next page
```

Python Dictionary Example (Cont'd.)

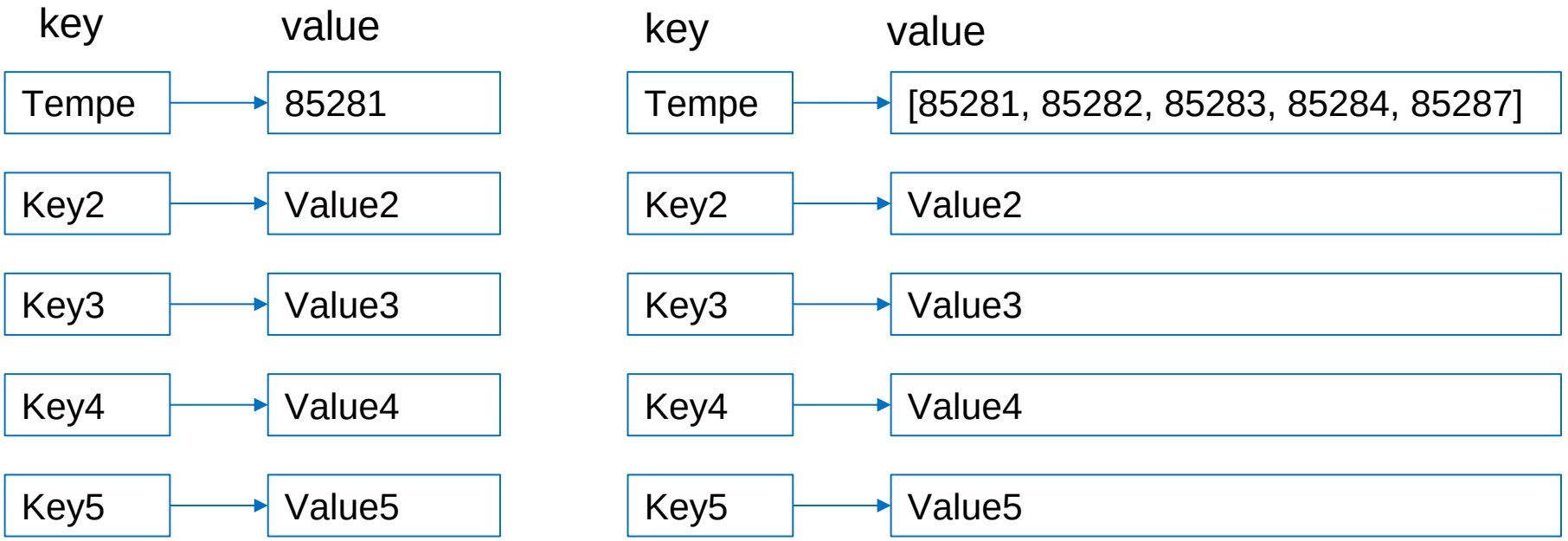
```
# Continued from previous page
for city in cities:
   if city in zipcode:
      print("The zip code of", city, "is", zipcode[city]) # lookup
   else:
      print("The city", city, "is not in my list")
city = input("Please enter a city name:")


found = False
for c in cities:
    c1 = c
    city1 = city
    if c1.lower() == city1.lower():
        print("The zip code of", c, "is", zipcode[c]) # lookup
        found = True
if found == False:
    print("The city", city, "is not in my list")
```

```
The zip code of Tempe is [85281, 85282, 85283, 85284, 85287]
The zip code of Mesa is [85201, 85202, 85203, 85204, 85205, 85206, 85207]
The city Gilbert is not in my list
The zip code of Phoenix is [85003, 85004, 85006, 85007, 85008, 85009]
The zip code of Scottsdale is [85250, 85251, 85253, 85254, 85255]
The zip code of Chandler is [85224, 85225, 85226, 85226]
Please enter a city name:Mesa
The zip code of Mesa is [85201, 85202, 85203, 85204, 85205, 85206, 85207]
Press any key to continue . . .
```

Structures of Dictionary / JSON Object

A dictionary/JSON object consists of key-value pairs.



Processing JSON Data Example in C#

https://msdn.microsoft.com/en-us/library/cc197957(v=vs.95).aspx

```
JsonArray users =
                (JsonArray)JsonArray.Load(responseStream);
var members =   from member in users
                where member["IsMember"]
                select member;
foreach (JsonObject member in members)
{
    string name = member["Name"];
    int age = member["Age"];
    // Do something...
}
```

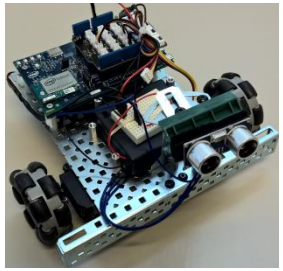
JSON Object Defined for IoT Communication
Text Chapter 9 Example

```
ROBOT OUTPUT

name: string (touch, distance, sound, light, color, motorEncoder)
id: int
value: For touch sensor, value will be an int (0 = not pressed and 1 = pressed).
       For other sensors, value will be a double

{"sensors": [{"name":"touch", "id":0, "value":0},{"name":"distance", "id":1, "value":12.8}]}
```

ASU VIPLE Visual IoT Programming



IoT / Robot

An object pair, with the second element an array of objects

```
ROBOT INPUT

servoId: int

servoSpeed: double between -1 and 1
 - negative values represent a backwards motion

{"servos": [{"servoId":3, "servoSpeed":0.5},{"servoId":5, "servoSpeed":-0.5}]}
```

Object

Array

Object

Object

Where are RSS, Atom, and JSON used?

They are widely used in

- Web data integration – Web page feed: Another Web page can conveniently read your Web data with a structure, instead of in a string.
- They are widely used as the RESTful service outputs.

Why do we need RSS, Atom, JSON, instead of using general XML? Choose one!

A. General XML is too simple to represent feed data.

B. General XML cannot represent a collection of data.

C. General XML is a meta language, and it can be used to define languages only.

D. General XML file requires a schema definition, while RSS, Atom, and JSON have fixed schemas and do not need to attach a schema file. A simpler and fixed structure is good enough for most feed applications.


Ira A. Fulton Schools of
Engineering
Arizona State University