Using AnIML to Publish Data on the Internet

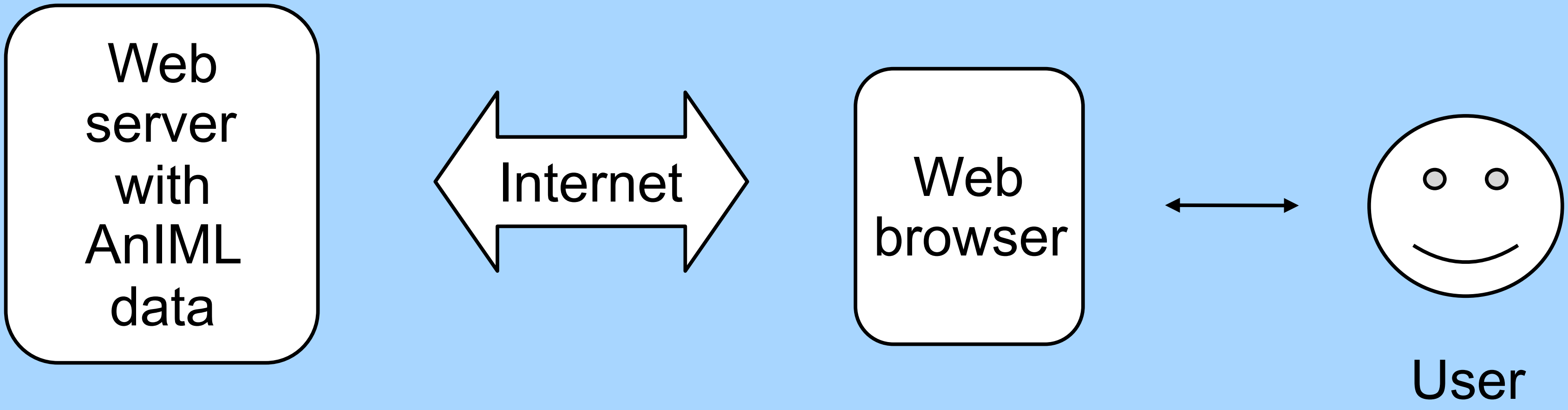
Peter J. Linstrom
Physical and Chemical Properties Division
National Institute of Standards and Technology

Summary

- Scope of this talk
- Potential applications
- AnIML and web servers
- Server side parsing (parsing AnIML files)
- Client side parsing
- AJAX
- Searching and other applications
- Guarding secrets

Scope of this talk

Will talk about this:



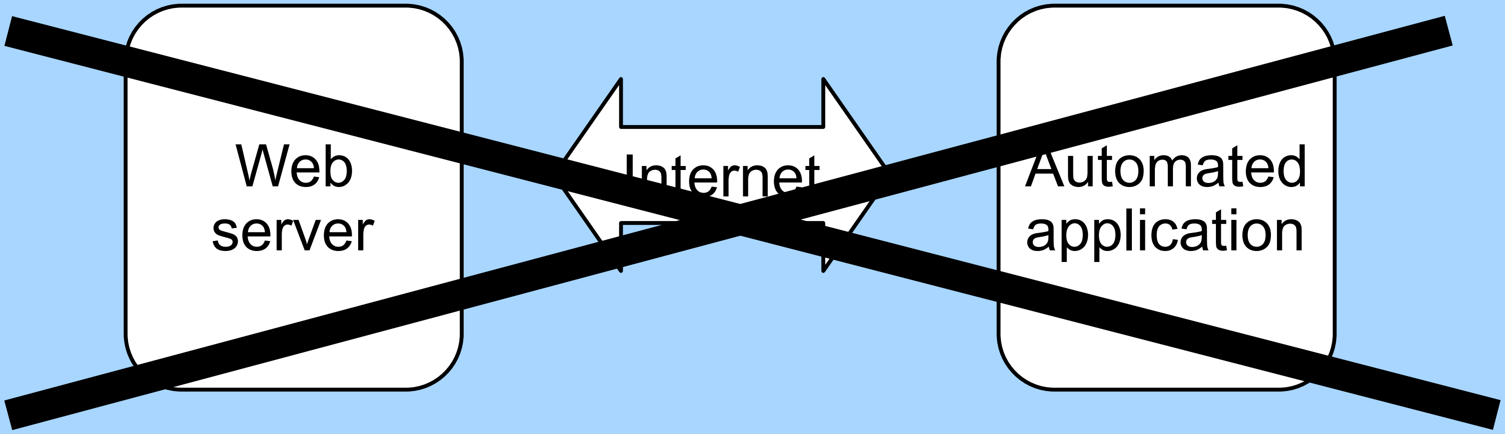
Web server with AnIML data

Internet

Web browser

User

Won't say much about this:



Web server

Internet

Automated application

Potential applications

- Academic applications (Internet).
  - Widespread dissemination of scientific knowledge
  - AnIML files include relevant meta-data and annotations

- Commercial publishers (Internet)
  - Security model(s) already in place for subscription sites

Potential applications

- Commercial (Intranet).
  - Uses existing infrastructure and security model
  - No or minimal client side software required.
  - Can tailor documents and indexes to applications specific to your organization (without breaking standard applications).

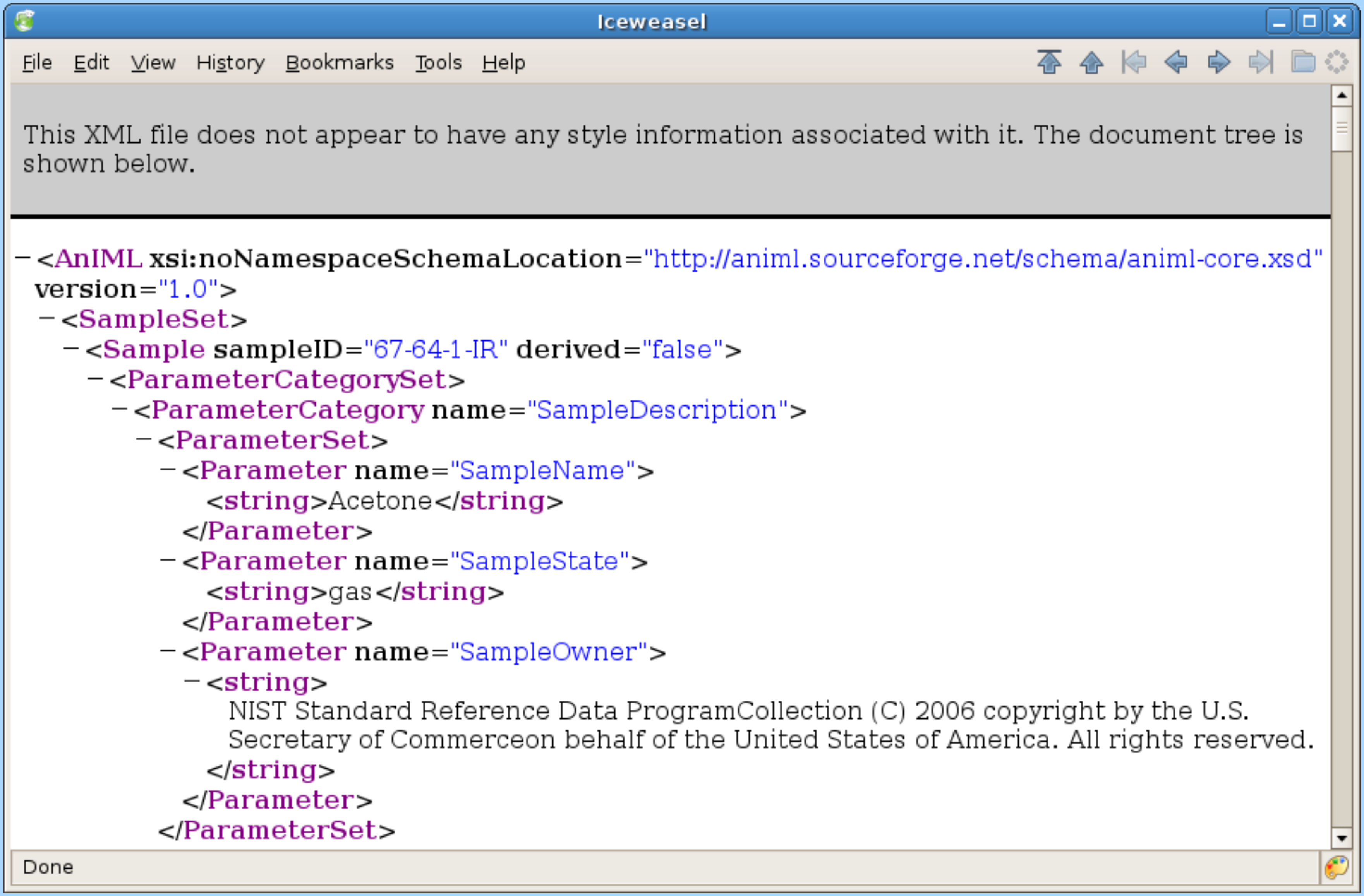
AnIML and web servers

- For this talk “web server” means HTTP server (e.g., Apache, IIS, JWS)
- Most large organizations have web servers for publishing information for internal or external consumption

Historical Note

- JCAMP-DX was designed to be “7-bit clean” for compatibility with early e-mail systems
- AnIML is XML based, so it reflects newer network technologies

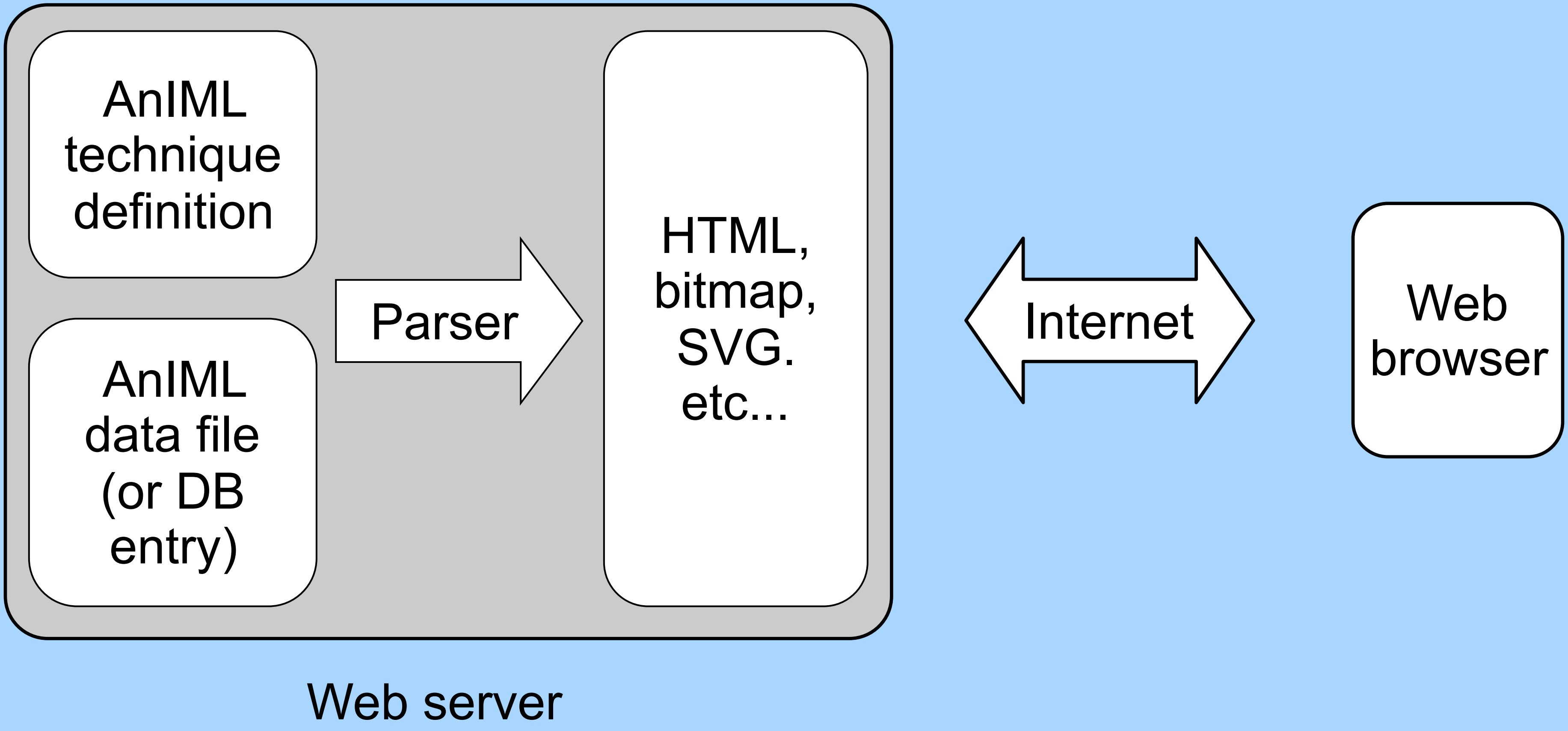
Raw AnIML data file



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
– <AnIML xsi:noNamespaceSchemaLocation="http://animl.sourceforge.net/schema/animl-core.xsd"
  version="1.0">
  – <SampleSet>
    – <Sample sampleID="67-64-1-IR" derived="false">
      – <ParameterCategorySet>
        – <ParameterCategory name="SampleDescription">
          – <ParameterSet>
            – <Parameter name="SampleName">
                <string>Acetone</string>
              </Parameter>
            – <Parameter name="SampleState">
                <string>gas</string>
              </Parameter>
            – <Parameter name="SampleOwner">
              – <string>
                  NIST Standard Reference Data ProgramCollection (C) 2006 copyright by the U.S.
                  Secretary of Commerceon behalf of the United States of America. All rights reserved.
                </string>
              </Parameter>
            </ParameterSet>
```

Server side parsing


AnIML technique definition
AnIML data file (or DB entry)
Parser
HTML, bitmap, SVG. etc...
Internet
Web browser
Web server

Server side parsing

- All the work is done on the server.
- AnIML files can be converted to graphics and reports (text) “on the fly”.
- Should work with almost any browser
  - No “plug-ins” required
  - Unfortunately there is no widely implemented vector graphics standard
- Interactivity is limited unless something like AJAX is used.

Parsing AnIML files – Good news

- There are many tools for parsing XML
  - Commercial and free software
  - Multiple language support – some languages have XML parsers in their standard libraries
  - Standard interfaces (SAX and DOM)
  - Tokenizing parsers can help handle large files

- XSL (eXtensible Stylesheet Language Family) can be used to extract and process meta-data

Parsing AnIML files – XSL tools

- XSLT (eXtensible Stylesheet Language Transformations)
  - Maps XML from one dialect to another (or to plain text)
- XSL-FO (eXtensible Stylesheet Language – Formatting Objects)
  - Formats text documents based on XML input
- XPath
  - Accesses or refers to part of an XML file

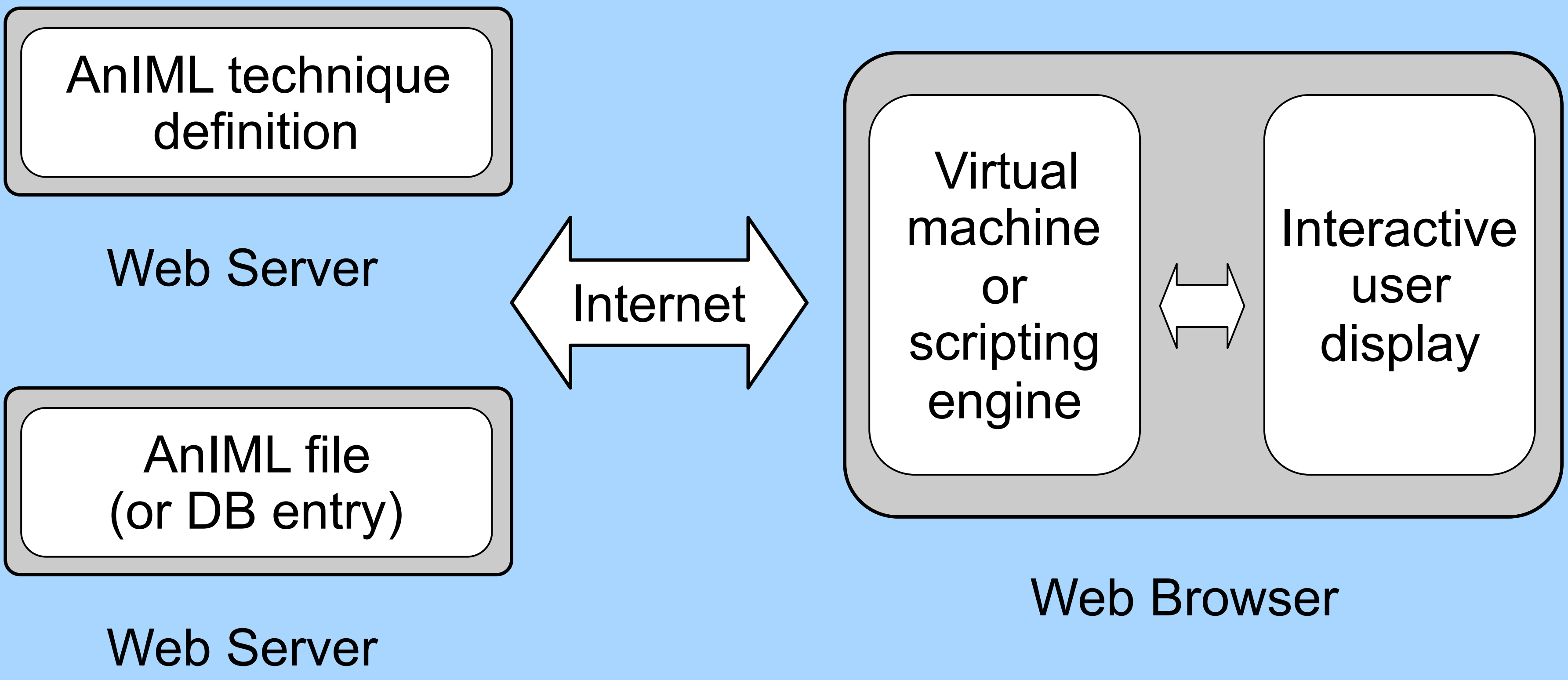
Parsing AnIML files – Work to do

- Using XML provides many advantages, but producing useful reports from AnIML files will still require significant AnIML specific code.

- XSLT is inappropriate for processing measurement (graphical) data from AnIML files

- Interpreting technique definition files requires more work but produces code which is more versatile (lower life cycle cost)

Client side parsing



AnIML technique definition

Web Server

AnIML file (or DB entry)

Web Server

Internet

Virtual machine or scripting engine

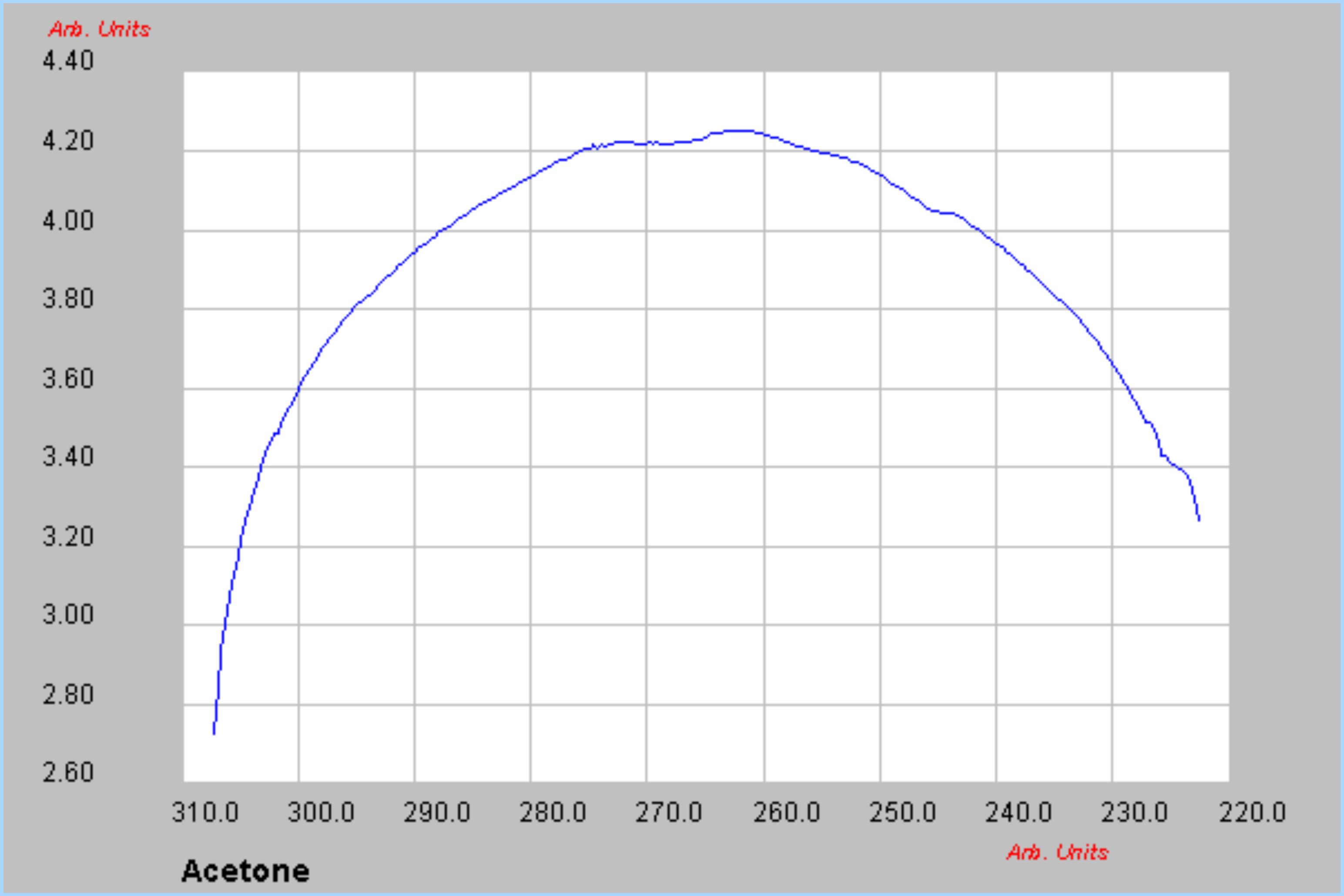
Interactive user display

Web Browser

Client side parsing

- Offloads work from server
- Can provide better user interactivity (graphics)
- Javascript
  - Scripting language native to browser
  - Limited interactivity without using browser extensions or some server side processing
- Java
  - Virtual machine run via a “plug-in”
  - Capabilities only limited by security model

Java Applet – Dr. Robert Lancashire

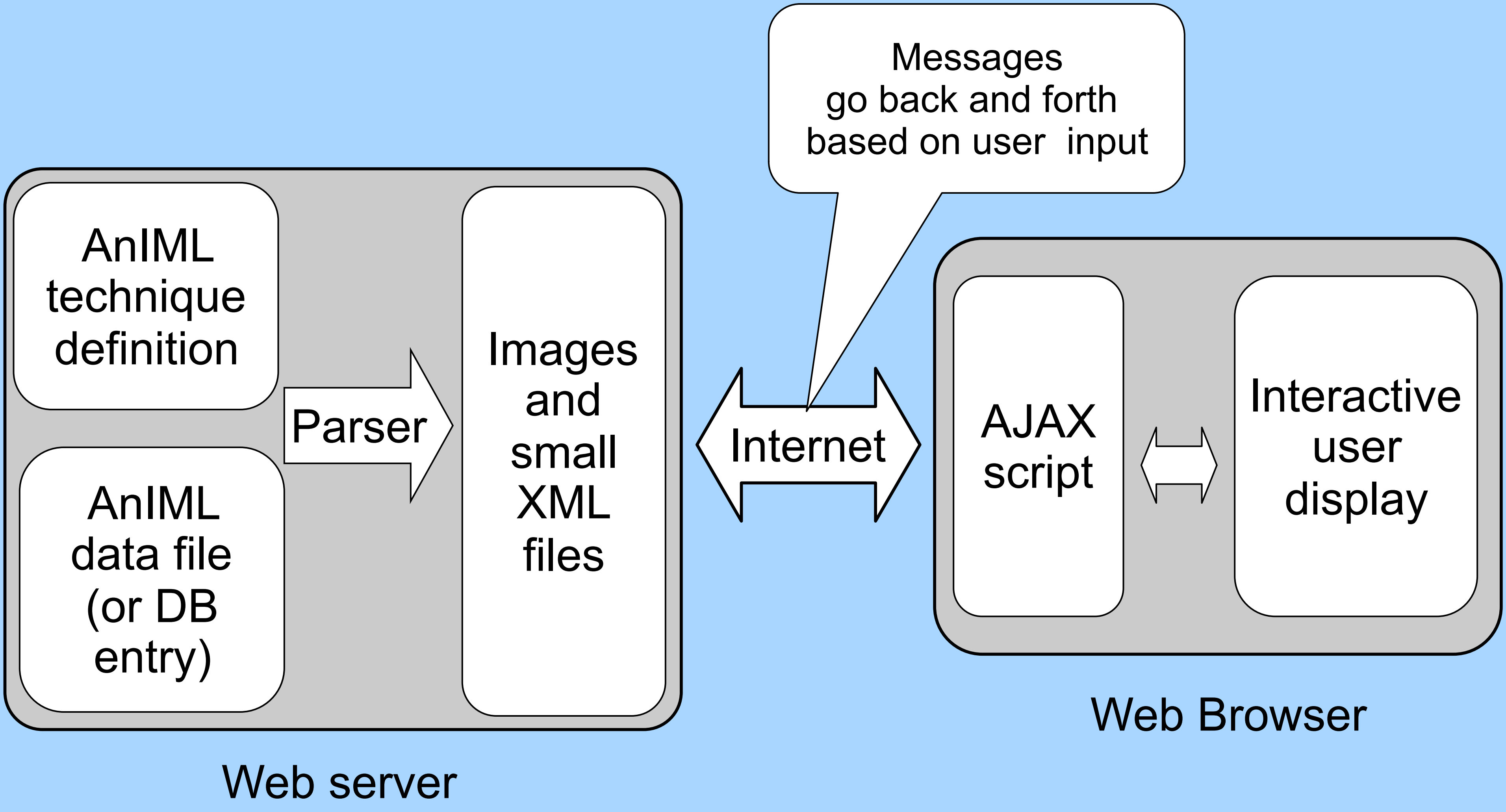


http://jspecview.sourceforge.net/

AJAX

- Asynchronous Javascript and XML
- A group of technologies which create interactive applications via the exchange of messages between the client and the server.
- Examples include the PubChem chemical structure editor and Google Maps.
- Usually a combination of server side and client side parsing
  - client would probably not be parsing AnIML files

AJAX



Messages go back and forth based on user input

AnIML technique definition

AnIML data file (or DB entry)

Parser

Images and small XML files

Internet

AJAX script

Interactive user display

Web server

Web Browser

Searching and other applications

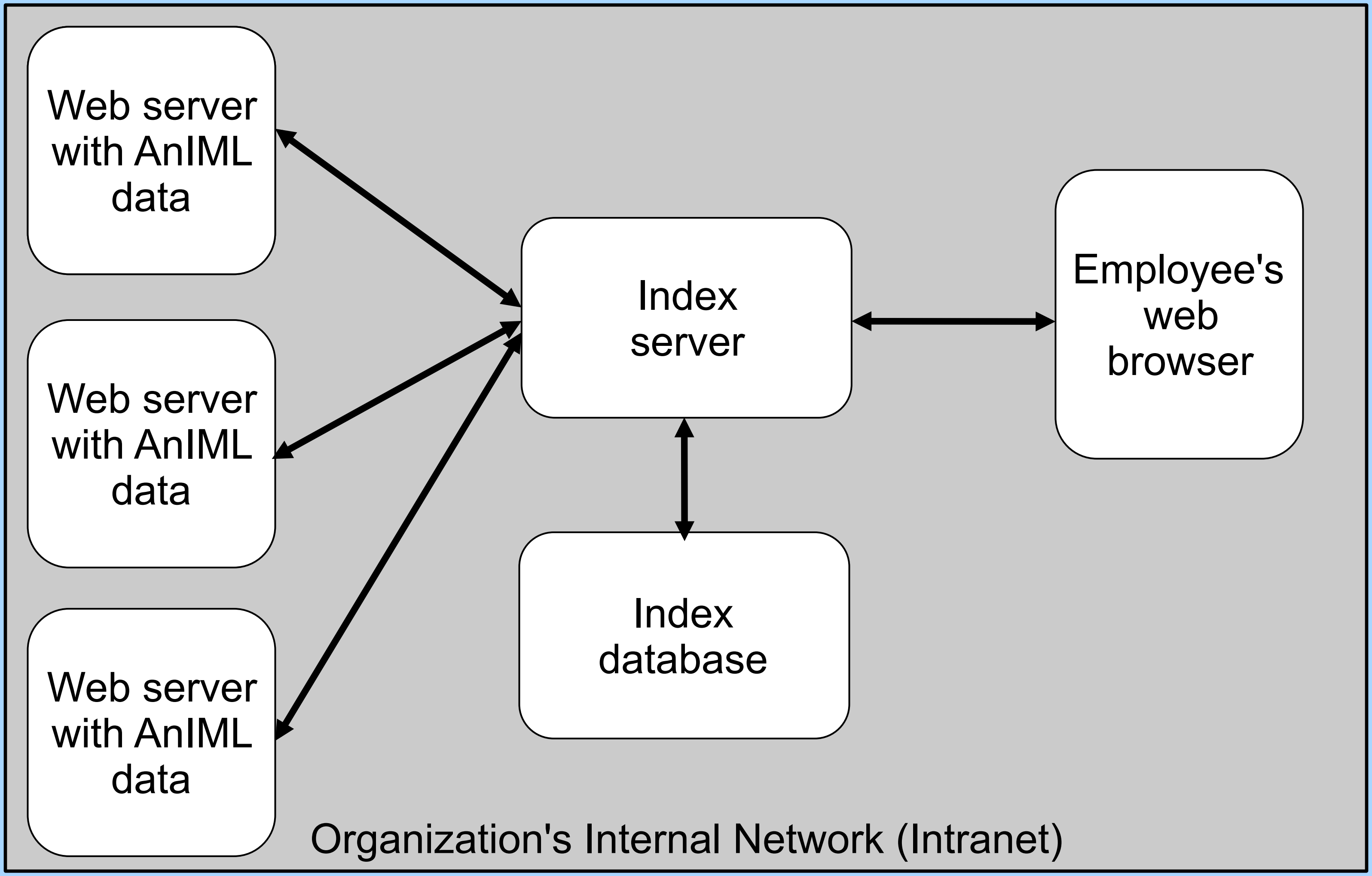
- AnIML files on server can be indexed by search engines (no new technology required)
- Organizations can use XML tools to build specialized applications for their needs
  - Build indexes of important information on the companies servers
  - Develop interfaces for applications to remotely process data stored on servers
  - Can scale throughout an organization
  - This is the stuff with an “X” on it on slide 3

Making an index

- Extract all the parameters called "SampleName" from an AnIML file:

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

<xsl:output method="text" />

<xsl:template match="/">
  <xsl:apply-templates select="AnIML//Parameter[@name='SampleName']" />
</xsl:template>

<xsl:template match="/AnIML//Parameter[@name='SampleName']">
<xsl:value-of select="normalize-space(.)" /><xsl:text>
</xsl:text>
</xsl:template>

</xsl:stylesheet>
```

Using an index


Web server with AnIML data
Web server with AnIML data
Web server with AnIML data
Index server
Employee's web browser
Index database
Organization's Internal Network (Intranet)

Guarding secrets

- As noted earlier, HTTP servers are common in many organizations
- Access control tools are available
  - Built into the server: network address and simple password based authentication
  - Available as third party packages: subscription managers and interfaces to enterprise identity management systems
- Working with established technology → lower risk

Conclusion

- There are a variety of ways data from an AnIML file can be effectively published using Internet technologies.
- For more information about AnIML, visit the web site at:

http://animl.sourceforge.net/

WSDL (SJC)

```
<!--
 WSDL description of the Google Web APIs.
     The Google Web APIs are in beta release. All interfaces are subject to
     change as we refine and extend our APIs. Please see the terms of use
     for more information.
-->
<!-- Revision 2002-08-16 -->
<definitions name="GoogleSearch" targetNamespace="urn:GoogleSearch">
<types>
    <xsd:schema targetNamespace="urn:GoogleSearch">
    <xsd:complexType name="GoogleSearchResult">
    <xsd:all>
    <xsd:element name="documentFiltering" type="xsd:boolean"/>
    <xsd:element name="searchComments" type="xsd:string"/>
    <xsd:element name="estimatedTotalResultsCount" type="xsd:int"/>
    <xsd:element name="estimateIsExact" type="xsd:boolean"/>
    <xsd:element name="resultElements" type="typens:ResultElementArray"/>
    <xsd:element name="searchQuery" type="xsd:string"/>
    <xsd:element name="startIndex" type="xsd:int"/>
    <xsd:element name="endIndex" type="xsd:int"/>
    <xsd:element name="searchTips" type="xsd:string"/>
    <xsd:element name="directoryCategories" type="typens:DirectoryCategoryArray"/>
    <xsd:element name="searchTime" type="xsd:double"/>
    </xsd:all>
    </xsd:complexType>
```