

Imagining, using and building an XSLT virtual machine

Mark Howe

Tony Graham

www.xcruciate.co.uk







Church of Fools
shipoffools.com



specialmoves



SEARCH



The Web



CNN.com

- Home Page
- Asia
- Europe**
- U.S.
- World
- World Business
- Technology
- Science & Space
- Entertainment
- World Sport
- Travel
- Weather
- Special Reports

- ON TV**
- What's on
- Business Traveller
- Design 360
- Global Office
- Principal Voices
- Spark

WORLD

Devils hit cyber church

By CNN's Graham Jones

Wednesday, May 19, 2004 Posted: 1442 GMT (2242 HKT)

LONDON, England (CNN) -- The world's first online church has been forced into an urgent rethink after computer hackers logged in as "Satan" and disrupted services with four-letter expletives and racist remarks.

The Church of Fools said Wednesday it had shut to outsiders its pulpit, lectern and space round the altar to stop those who





Pick up thy coffee and chat

Skip the intro and [view hot topics](#).



Welcome to the discussion area of St Pixels. Find a table, grab a coffee and join in.

Our current discussion areas are:

- [Church life](#) and what it means to be christian community.
- [Everyday life](#) table where we talk about work, play and everything in between.
- [Current events](#) and the news of the world.
- [Culture](#) for discussing movies, music, cultural icons and the worldview surrounding us.
- [Opinion](#) The place for longer articles, thoughtful pieces or expositions.
- [Talk theology](#) for those curlier questions that we love to argue over
- [That book](#) for discussions about the bible.
- [View hot topics](#) lets you cheat and find a list of the latest discussions.

Anyone who wants to can suggest any topic for discussion [here](#).

The team

These people are your hosts.

- [Steven](#) (Coordinator)
- [Jame](#)
- [Jeff](#)
- [Jools](#)
- [Karl](#)
- [Matthew Jones](#)
- [mymoss](#)

We are here to help in a question or concern y

(image from Flickr

Our Mighty A

Are we as good r

Karl says:

Today, after sav
a \$55,00000r

Two geese.

You can br

Does thi

[Join th](#)



```

<rss version="2.0" xmlns:blogchannel="http://backend.userland.com/blogChannelModule">
  <channel>
    <title>St Pixels blog entries</title>
    <link>http://www.stpixels.com/view_bloglatest.cgi</link>
    <description>Blog entry feed from St Pixels</description>
    <language>en</language>
    <item>
      <title>and still it goes on ...</title>
      <link>http://www.stpixels.com/view_blogentry.cgi?nugget=381530</link>
      <description>and still it goes on ...</description>
      <pubDate>Mon, 16 Mar 2009 9:01:10 EDT</pubDate>
    </item>
    <item>
      <title>It's wrong ...</title>
      <link>http://www.stpixels.com/view_blogentry.cgi?nugget=381524</link>
      <description>It's wrong ...</description>
      <pubDate>Mon, 16 Mar 2009 8:41:08 EDT</pubDate>
    </item>
    <item>
      <title>Hymn verse for Today Tues 17 March</title>
      <link>http://www.stpixels.com/view_blogentry.cgi?nugget=381523</link>
      <description>Hymn verse for Today Tues 17 March</description>
      <pubDate>Tue, 17 Mar 2009 3:49:05 EDT</pubDate>
    </item>
    <item>
      <title>Pidge: Guff</title>
      <link>http://www.stpixels.com/view_blogentry.cgi?nugget=381522</link>
      <description>Guff</description>
      <pubDate>Mon, 16 Mar 2009 16:44:17 EDT</pubDate>
    </item>
    <item>
      <title>Dom: woohoo</title>
      <link>http://www.stpixels.com/view_blogentry.cgi?nugget=381521</link>
    </item>
  </channel>
</rss>

```

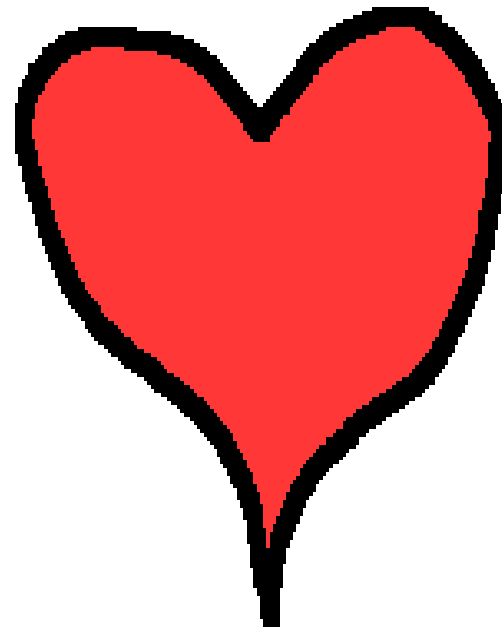
Building

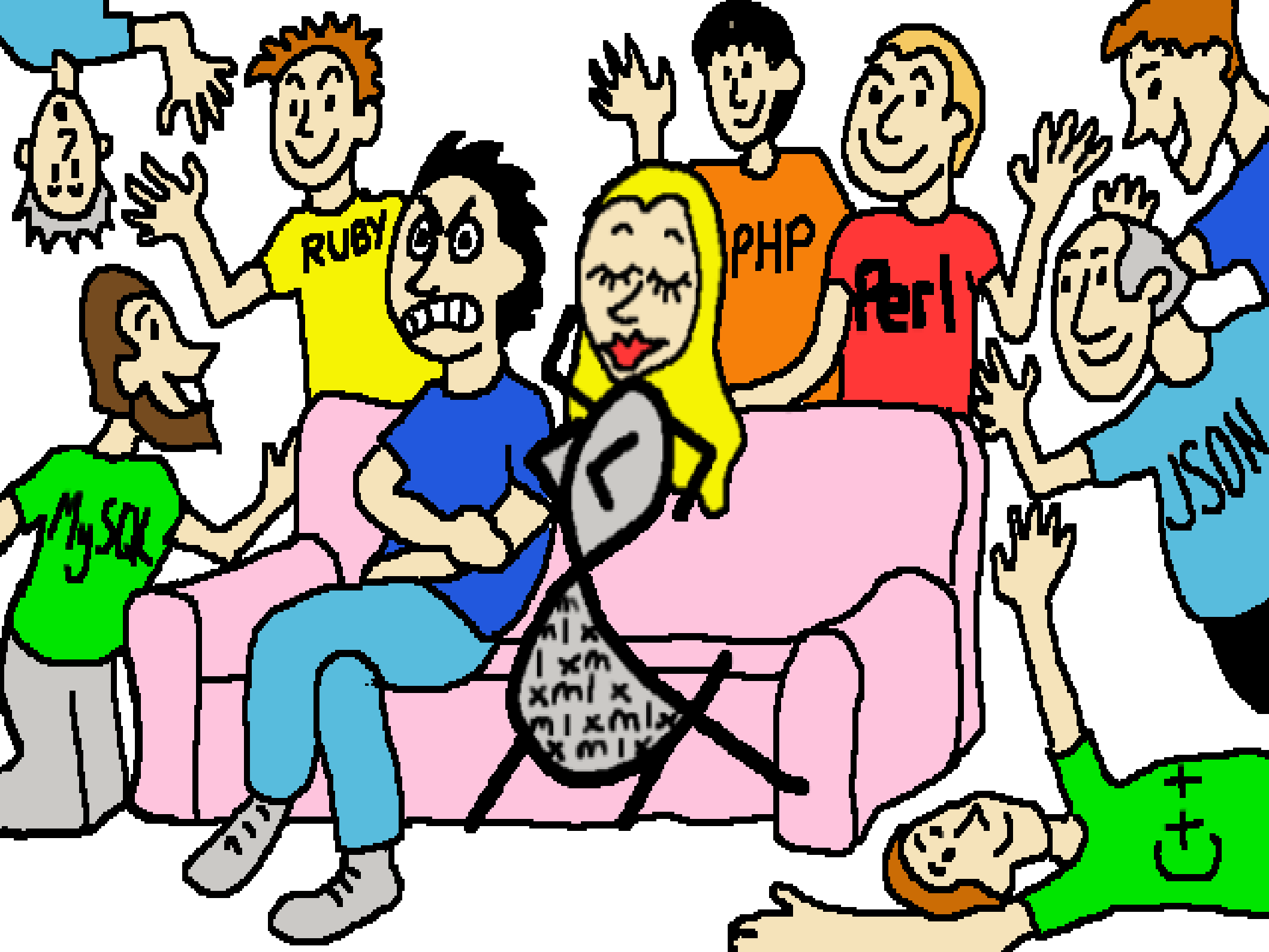


mark

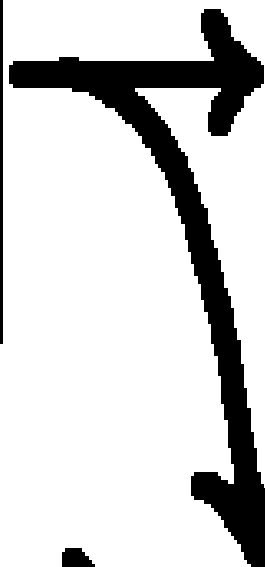
Chat Profile Ignore

Talk Prefs Website Sound Status: Normal





XML

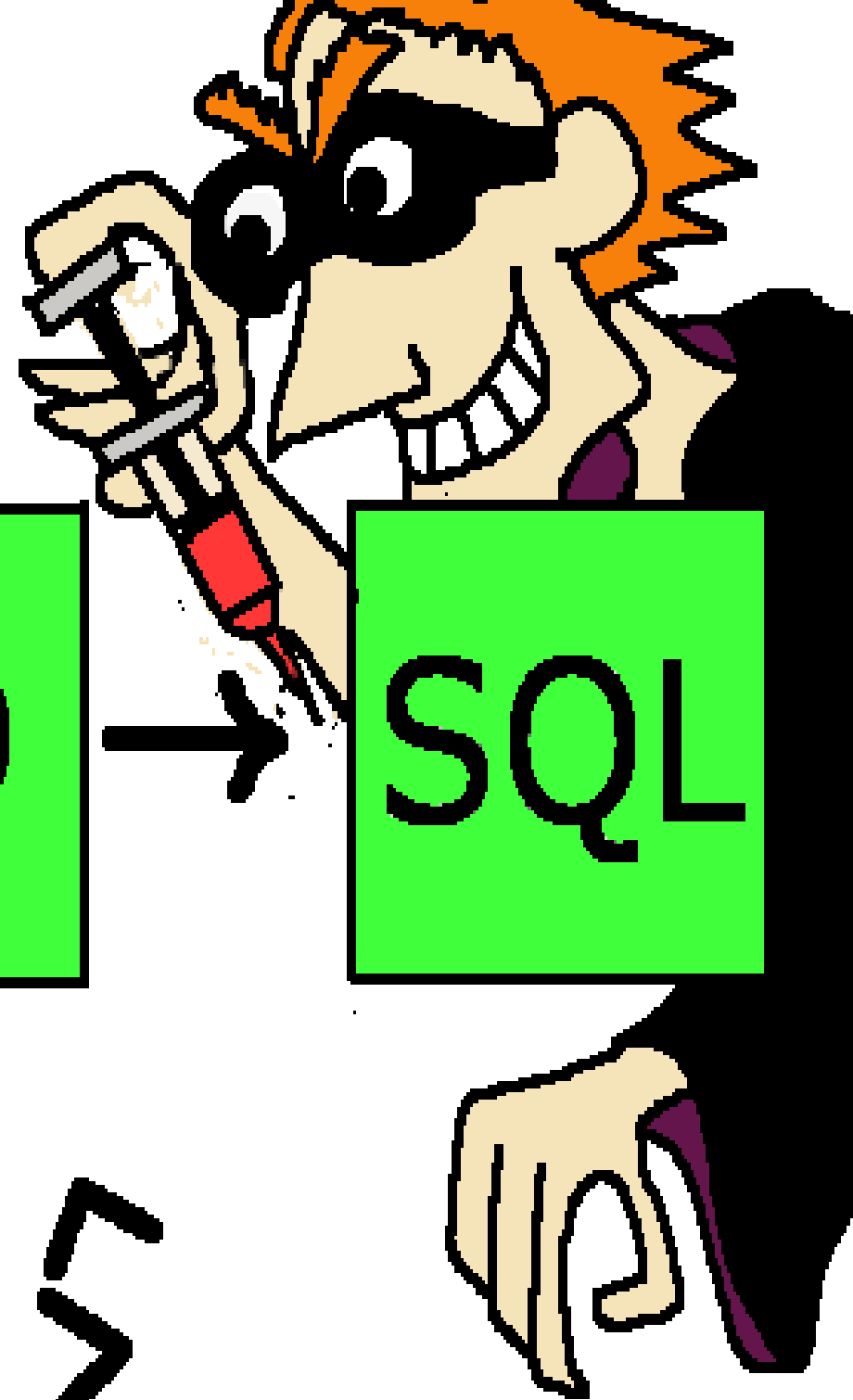


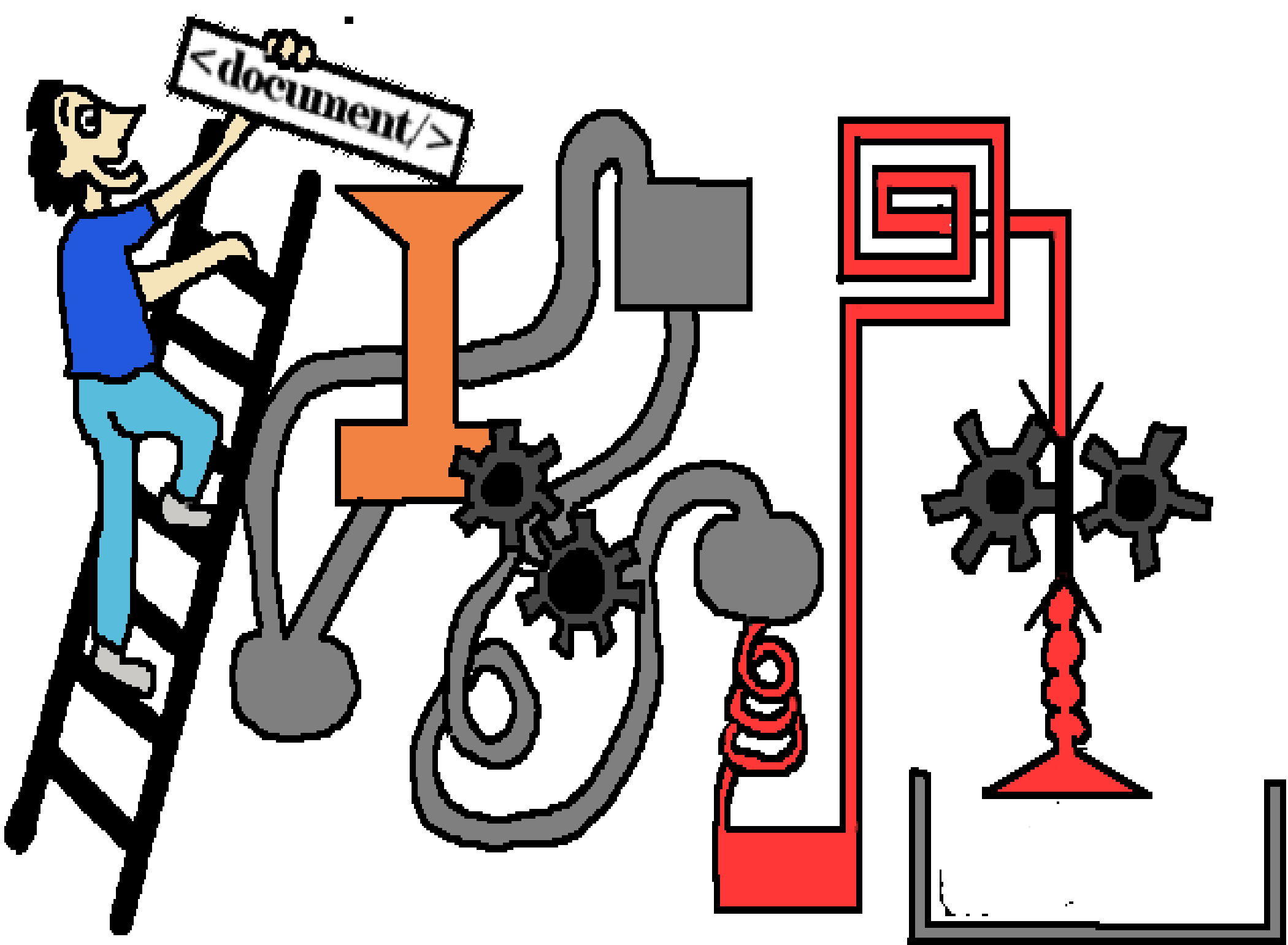
OO

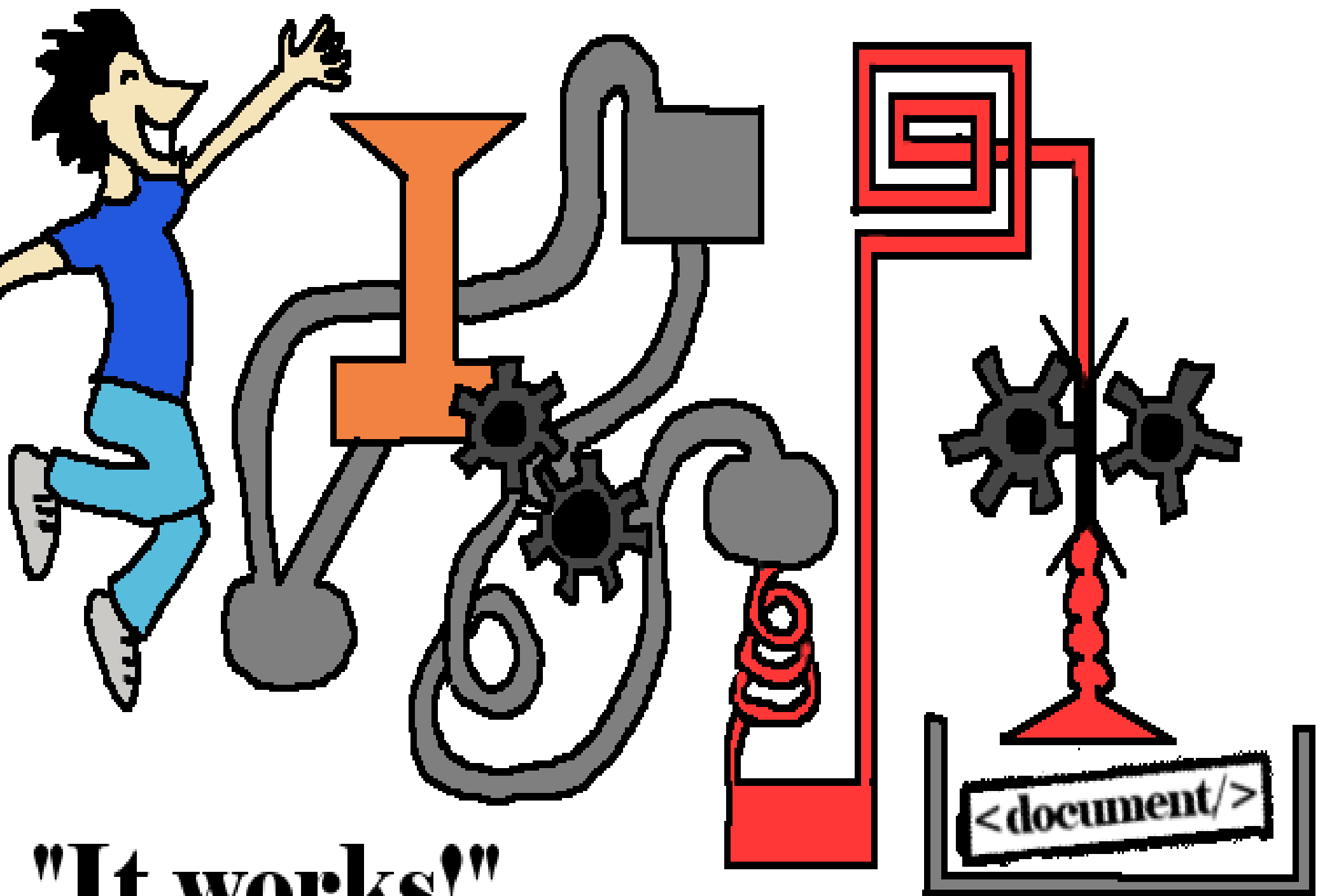


SQL

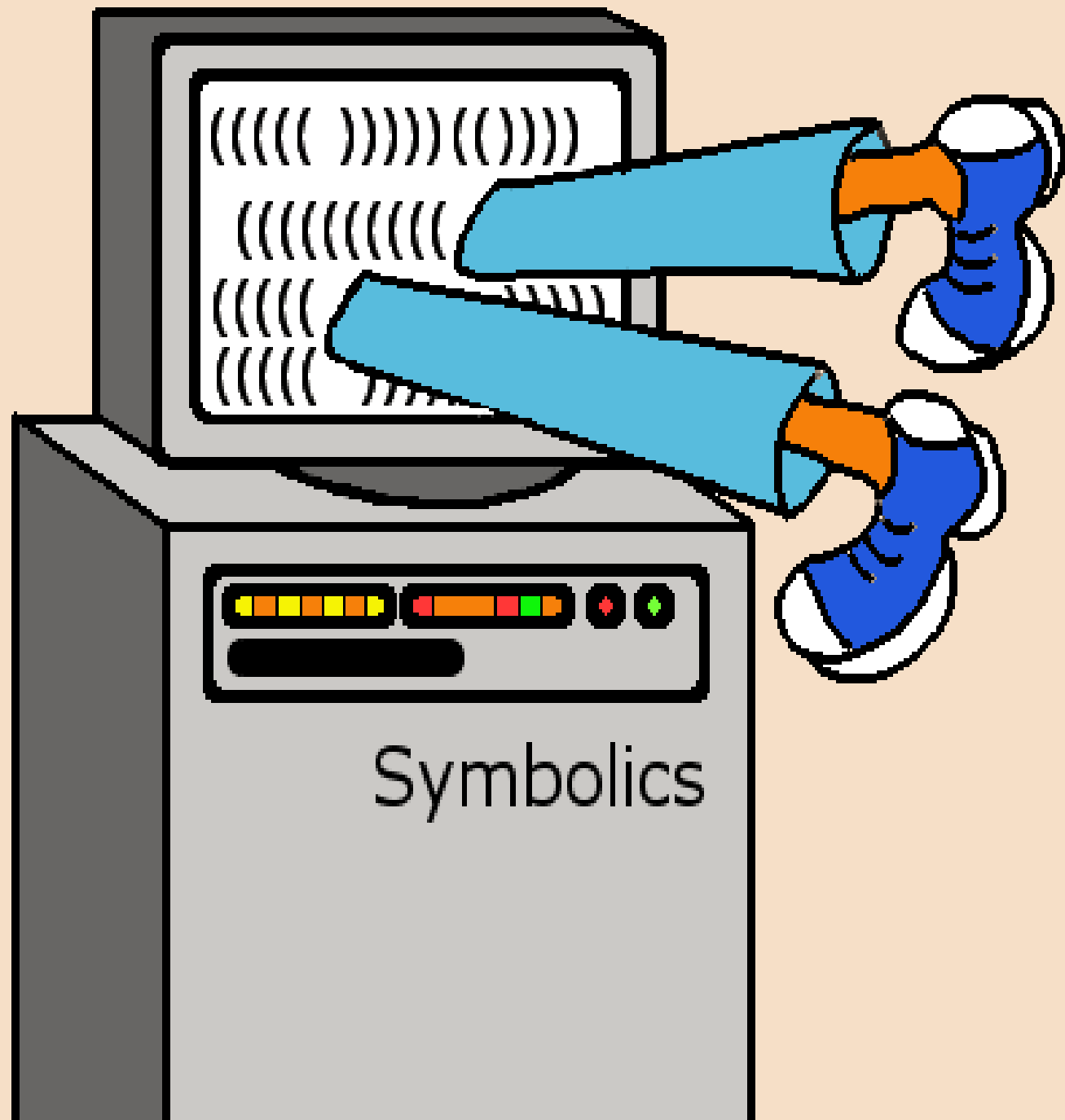
∞
∟ ∟
∟ ∟
∟ ∟
∟ ∟







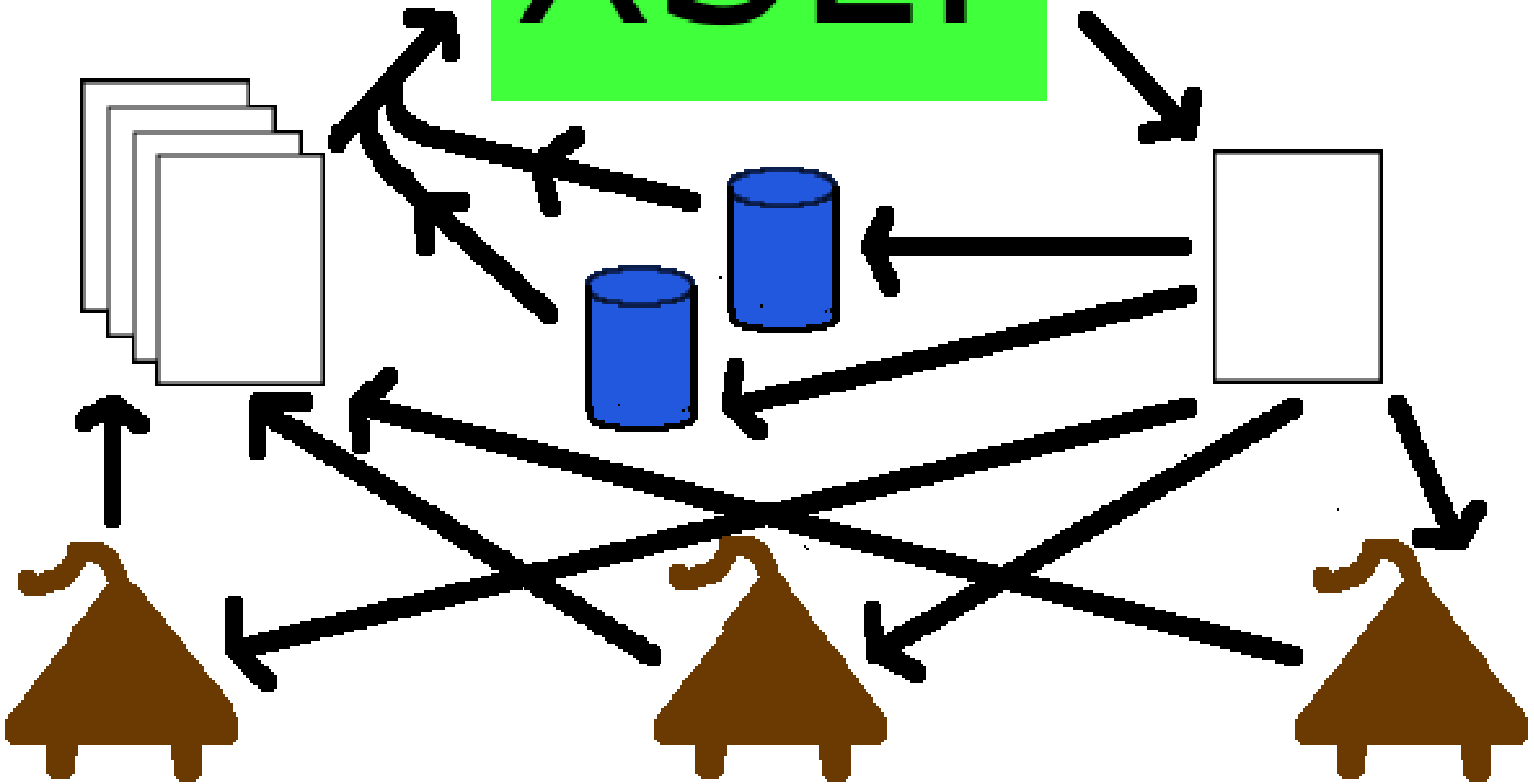
"It works!"

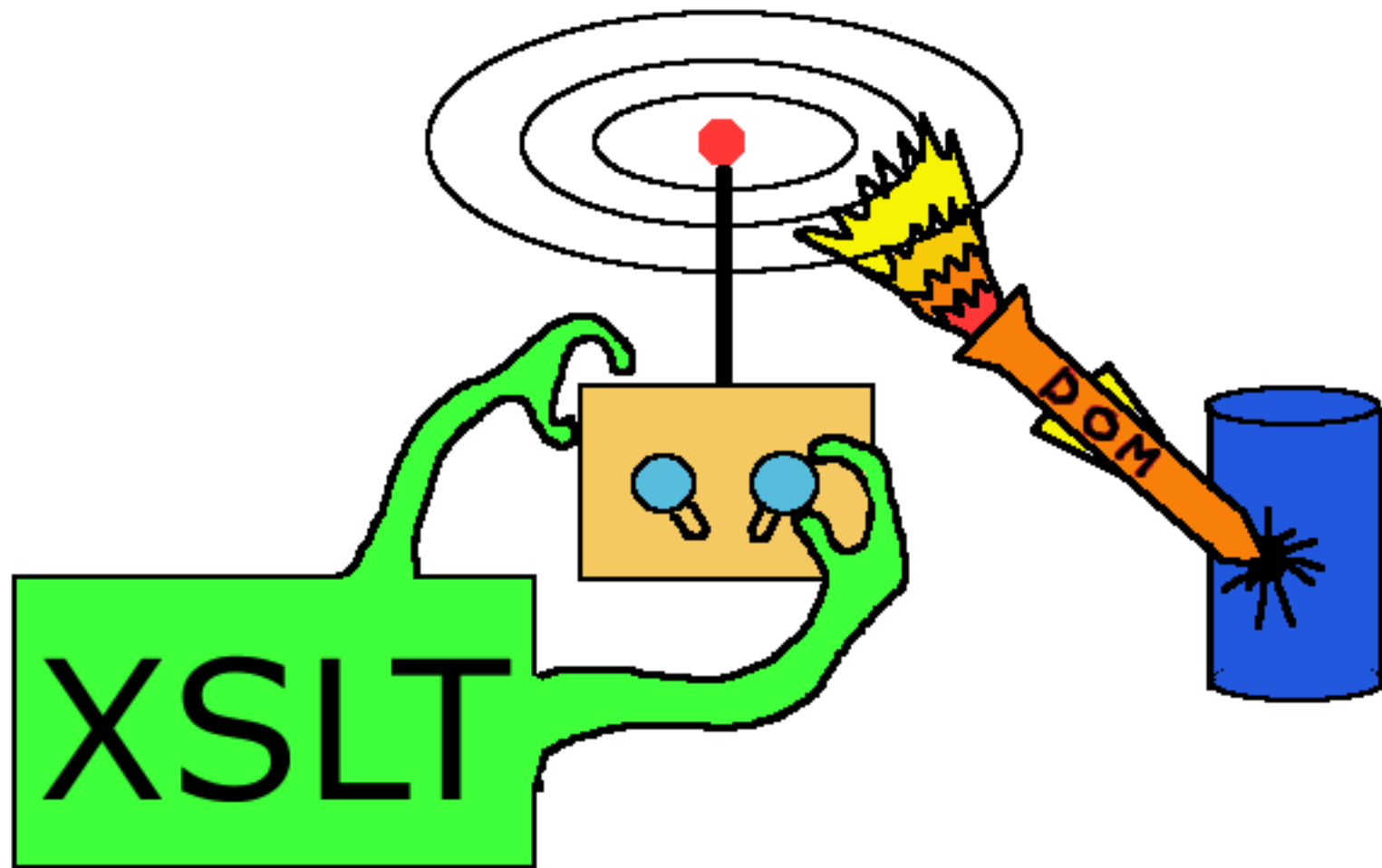


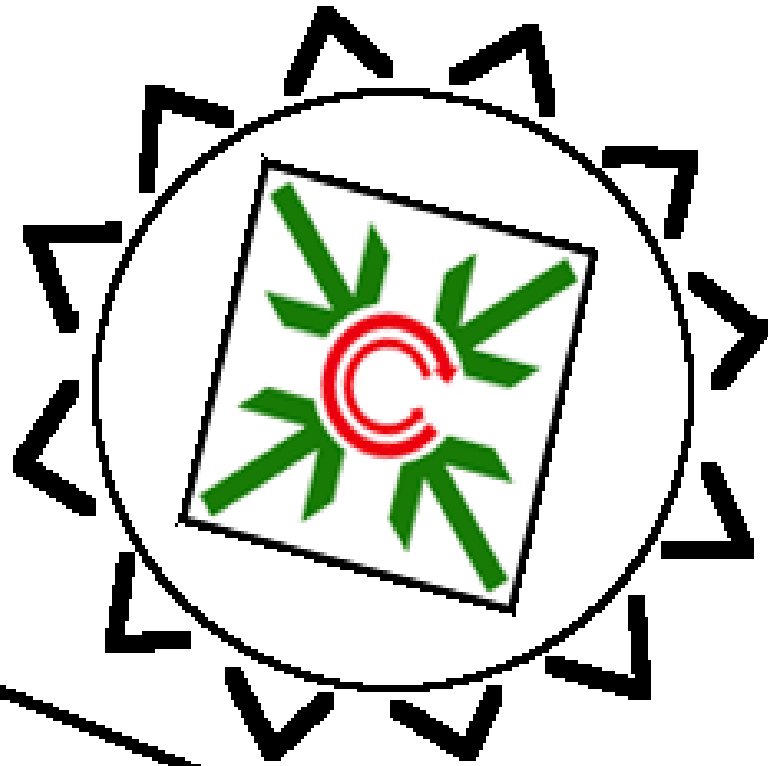
Lost in the brackets.....

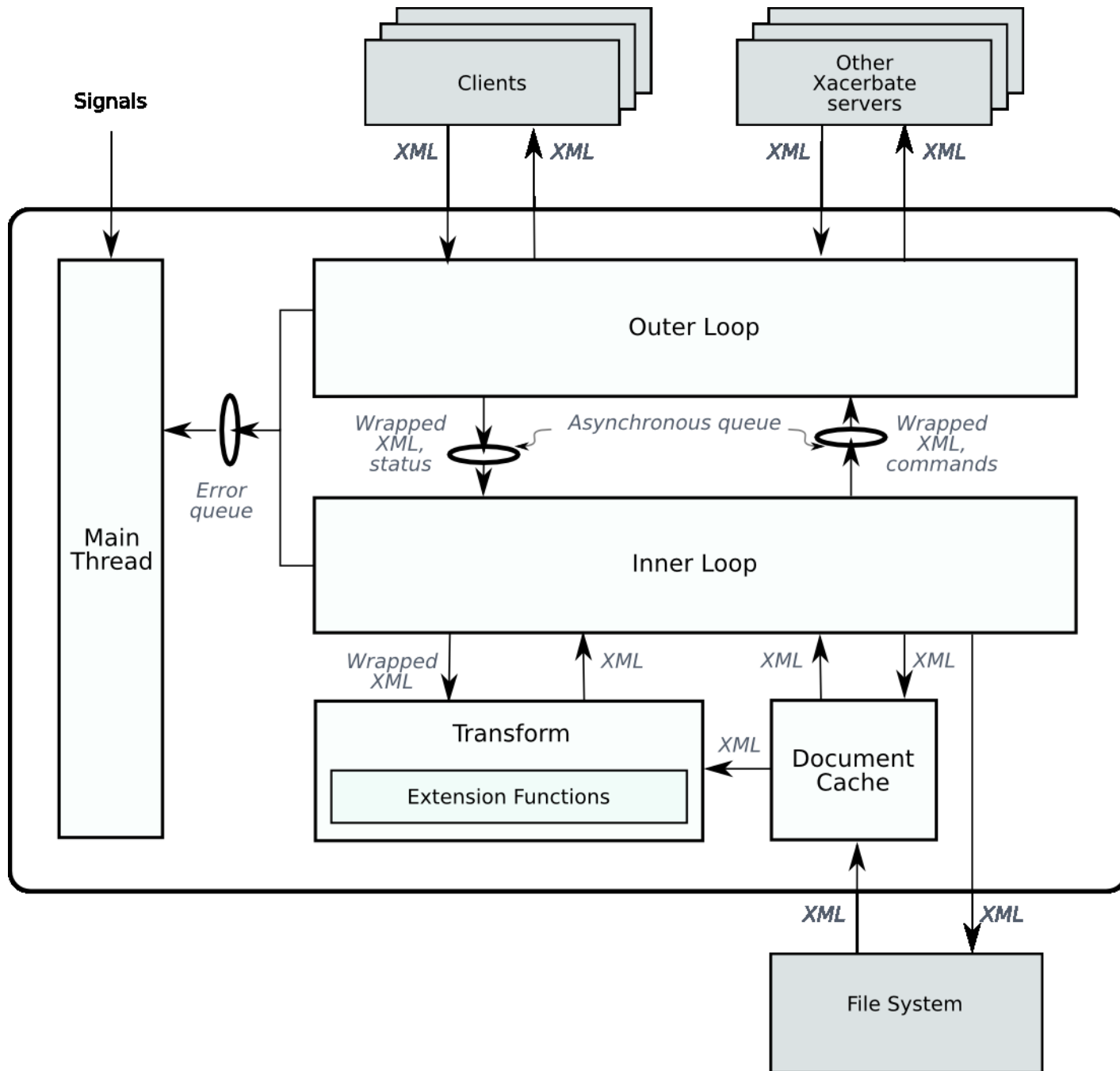


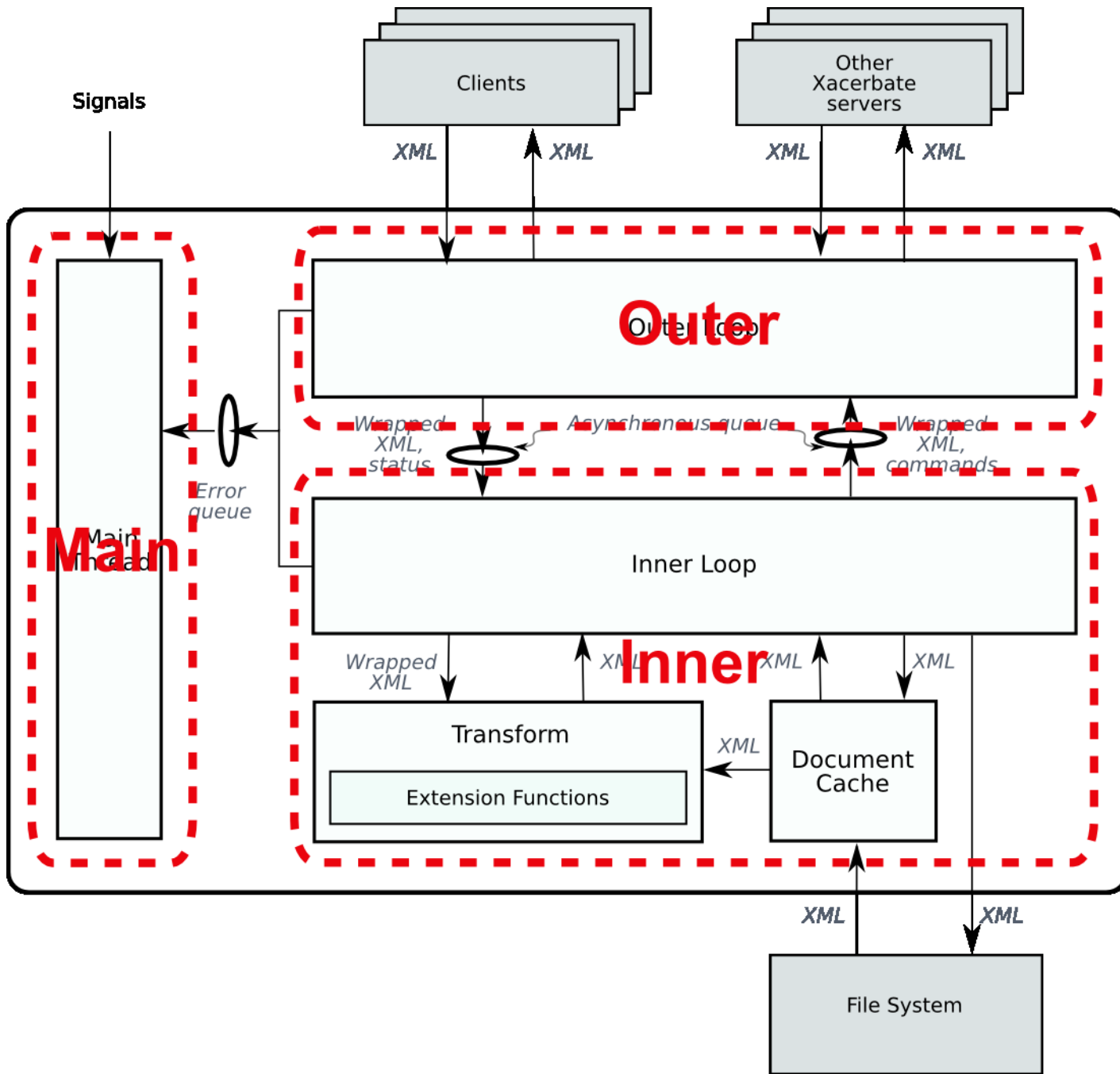
XSLT

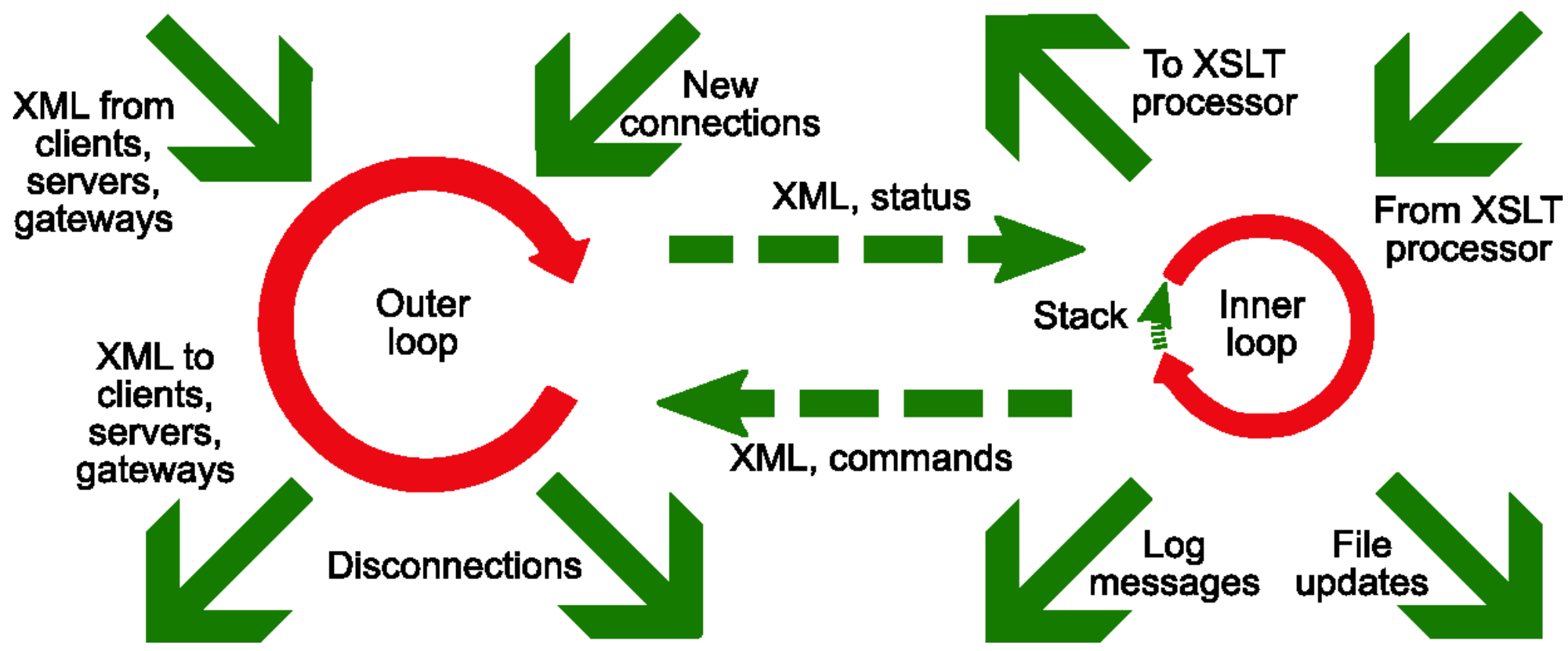












Extension Functions

- ID functions
 - `xacf:declare_id(id)`, etc.
 - IDs usable across multiple XSLT transformations
- Error functions
 - `xacf:set_error(name)`, etc.
 - Tracking error conditions within one transformation

Extension Functions

- Authentication functions
 - `xacf:crypt(key, salt)`, etc.
 - Authentication-related tasks made quicker and easier
- General functions
 - `xacf:epoch_time()`, `xacf:parse(string)`, etc.
 - Utility functions

```
<!-- Say hello on connect -->
<xsl:template match="/xacw:input[@source='self'][@event_type='socket']">
  <xsl:if test="xacw:event_input[@event='client_connected']">
    <xacw:outputs>
      <xsl:call-template name="to_socket">
        <xsl:with-param name="socket" select="xacw:event_input/@socket_id"/>
        <xsl:with-param name="content">
          <hello you_are="world"/>
        </xsl:with-param>
      </xsl:call-template>
    </xacw:outputs>
  </xsl:if>
</xsl:template>
```

```
<!-- The xigf:to_socket function can provide this functionality too. -->
<xsl:template name="to_socket">
  <xsl:param name="socket"/>
  <xsl:param name="content"/>
  <xacw:output
    destination="connection"
    socket_id="{ $socket }">
    <xsl:copy-of select="$content"/>
  </xacw:output>
</xsl:template>
```

```

<!-- Say hello via http or xiguous-type connection -->
<xsl:template match="/xacw:input[@source='connection']">
  <xacw:outputs>
    <xsl:call-template name="to_socket">
      <xsl:with-param name="socket" select="@socket_id"/>
      <xsl:with-param name="content">
        <xsl:choose>
          <xsl:when test="http_request">
            <http_outputs id="{http_request/http_info[@name='xteriorize_request_id']/.}">
              <http_output type="response" response_code="200" response_code_text="OK" http_version="{http_request/http_version/.}">
                <http_document document_type="text/html" uuencoded="no" doctype="xhtml1-strict">
                  <html xmlns="http://www.w3.org/1999/xhtml"> <!-- XHTML namespace for html expected by Xteriorize -->
                    <head>
                      <title>Minimal response to input</title>
                    </head>
                    <body>
                      <h1>Thanks for your HTTP request</h1>
                      <h2>You asked for <b><xsl:value-of select="http_request/http_url/."/></b></h2>
                    </body>
                  </html>
                </http_output>
              </http_outputs>
            </xsl:when>
            <xsl:otherwise>
              <message>Thanks for your zero-terminated input element called <xsl:value-of select="local-name(*[1])"/></message>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:with-param>
      </xsl:call-template>
    </xacw:outputs>
  </xsl:template>

```

```
<func:function name="app:html_body">
  <xsl:param name="request"/>
  <xsl:param name="content"/>
  <func:result>
    <body id="top">
      <div id="wrapper">
        <div id="header">
          <a href="/">
            
          </a>
          
        </div>
        <div id="nav">
          <xsl:copy-of select="xcaf:build_ul_menu()"/>
        </div>
        <xsl:copy-of select="$content"/>
      </div>
      <xsl:copy-of select="xcaf:footer(xtef:i18n('footer'))"/>
    </body>
  </func:result>
</func:function>
```

```
<!-- Function for hello world flash script -->

<func:function name="app:hello_flash">
  <xsl:param name="name"/>
  <func:result>
    <mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
      <mx:Button label="{concat('Hello ', $name)}"></mx:Button>
    </mx:Application>
  </func:result>
</func:function>

<!-- Function for hello world pdf script -->

<func:function name="app:hello_pdf">
  <xsl:param name="name"/>
  <func:result>
    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
      <fo:layout-master-set>
        <fo:simple-page-master
          font-size="inherit"
          margin-right="1in" margin-left="1in" margin-bottom="1in" margin-top="1in"
          page-width="21cm" page-height="29.7cm"
          master-name="first">
          <fo:region-body region-name="xsl-region-body"/>
        </fo:simple-page-master>
      </fo:layout-master-set>
      <fo:page-sequence
        master-reference="first">
        <fo:flow flow-name="xsl-region-body">
          <fo:block>
            <xsl:value-of select="concat('Hello, ', $name)"/>
          </fo:block>
        </fo:flow>
      </fo:page-sequence>
    </fo:root>
  </func:result>
</func:function>
```

```
</algorithm>
</script>
<script id="makeaflash2" secure="0">
  <method>get</method>
  <post_max>4096</post_max>
  <parameter name="screenname" presence_optional="0" value_optional="0">
    <valid_type>screenname</valid_type>
    <invalid_error>invalid_screen_name</invalid_error>
    <missing_error>you_must_supply_a_screenname</missing_error>
  </parameter>
  <algorithm>
    <variable name="fileid"><value-of>xacf:generate_id()</value-of></variable>
    <copy-of>xtef:write_file('flash',
                          concat('hello',$fileid,'.swf'),
                          'mxmlc',
                          app:hello_flash(xtef:param($REQUEST,'screenname')))</copy-of>
    <copy-of>xcaf:response_html($REQUEST,xtef:i18n('makeaflash_return_page'),
                               xcaf:link(concat('/flash/hello', $fileid, '.swf'),'Click here'))</copy-of>
  </algorithm>
</script>
<script id="makeapdf" secure="0">
  <method>get</method>
```

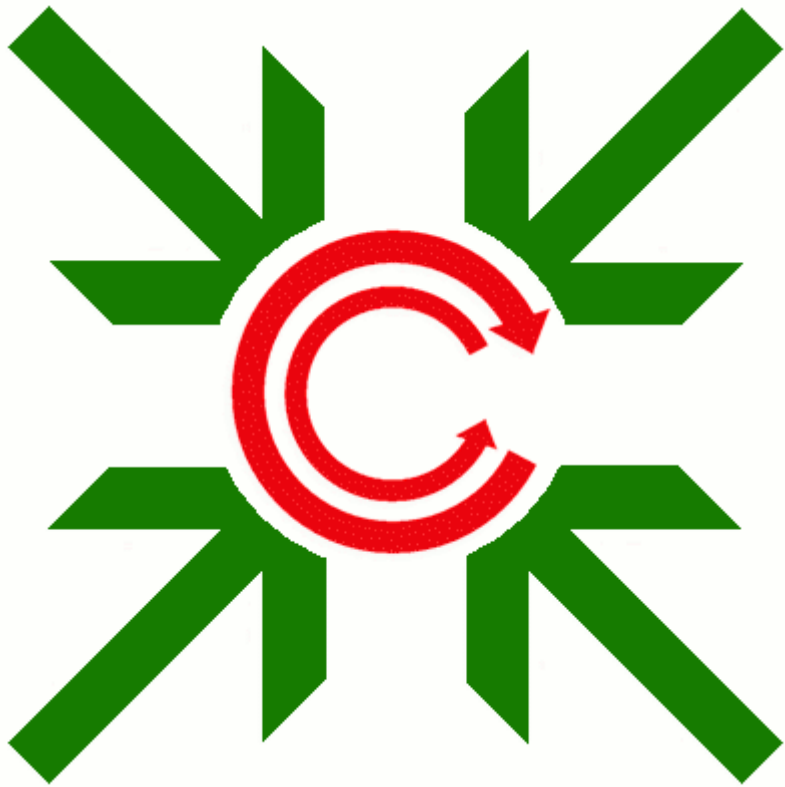
```
<xsl:choose>
  <xsl:when test="not(xacf:check_errors())">
    <xsl:copy-of select="dyn:evaluate(concat('anon:', $script/@id, '($request, $script)'))"/>
  </xsl:when>
  <xsl:otherwise>
```



Our server's gone polymorphic!



....and keeps calling me Dave!



XCRUCIATE

an all-XML server

www.xcruciate.co.uk