SAMP: Application Messaging for Desktop and Web Applications

ADASS XXI, Paris, 10 November 2011

Mark Taylor (Bristol University)
Thomas Boch (CDS)
Jonathan Fay (Microsoft Research)
Mike Fitzpatrick (NOAO)
Luigi Paioro (INAF)

& IVOA Applications Working Group
History:

- PLASTIC v1 (Platform for Astronomical InterConnection), Euro-VO protocol 2006
- SAMP v1.11, IVOA Recommendation 2009
- Useful client-side technology for VO work patterns
- . . . but not specific to VO applications

Status:

- Quite widely used in desktop tools
  - Java, Python, Perl, C, C#, Tcl, IDL, . . .
- Visible at ADASS
  - ADASS XIX: 5 subject index entries for “SAMP”
  - ADASS XX: 16 subject index entries for “SAMP” (5th after Java, Python, VO & XML)
Row List
Principles to maximise interoperability:

- Simple to use and learn for client developers and users
  - Platform independent
  - Lightweight to implement
- Message semantics are typically vague
  - "Here’s a table!" not “Plot entries from this catalogue over the current image”
  - but also extensible
  - Arise from usage, not decreed by committee

Consequences:

- Loosely coupled suites of interoperating tools
  - ... selected by the user
    - ... without conscious effort
  - ... from a pool of tools contributed by developers
    - ... who do not need close collaboration
- It works!
Hub-based operation

- Hub is daemon process, freestanding or within one client
- Clients *register* with Hub to send/receive (publish/subscribe to) messages
- Hub brokers messages and provides directory services
- All communication is Client ↔ Hub
  (but messaging conceptually Client↔Client)
3-Layer Architecture

Abstract API
- Data types
- Message structure
- Control functions

Profile
- Transport protocol
- Data encoding
- Hub discovery

MTypes
- Message semantics
- Arguments
- Return values

Standard Profile
- XML-RPC (= HTTP + XML)
- Hub URL in ~/.samp file

Examples:
- image.load.fits
- table.highlight.row
- coords.sky.pointAt
- ...
Web Applications

SAMP works well for *desktop clients*

Would like it to work for *web clients* (code running in a browser)

- In-browser platforms:
  - JavaScript (a.k.a. JScript, ECMAScript)
  - Adobe Flash
  - MS Silverlight
  - Java applet *(when signed, works already)*

- Example scenarios:
  - Button to send a table/image/spectrum from web to a suitable desktop viewer
  - Web page receives information from desktop clients, e.g. highlight catalogue rows
  - Web page communicates with other pages loaded in the same browser
  - Rich application that happens to live in a browser

- Many persuasive use cases!
Web Applications

SAMP works well for *desktop clients*

Would like it to work for *web clients* (code running in a browser)

- In-browser platforms:
  - JavaScript (a.k.a. JScript, ECMAScript)
  - Adobe Flash
  - MS Silverlight
  - Java applet *(when signed, works already)*

- Example scenarios:
  - Button to send a table/image/spectrum from web to a suitable desktop viewer
  - Web page receives information from desktop clients, e.g. highlight catalogue rows
  - Web page communicates with other pages loaded in the same browser
  - Rich application that happens to live in a browser

- Many persuasive use cases!
Web SAMP Technical Barriers

Browsers impose security restrictions ("sandbox") on web clients:

- can’t read local files
- can’t access URLs on localhost or external hosts (*cross-domain restrictions*)
- can’t run an HTTP server to receive callbacks

⇒ Untrusted web clients can’t exercise user privileges

- to damage the user’s system
- to send/receive SAMP messages using the Standard Profile

Options to circumvent sandbox?

- Signed Java applet — requires Java (*WebSampConnector, VO Paris Data Centre*)
- Browser plugin — browser-specific
- Alternative Profile
3-Layer Architecture

Abstract API
- Data types
- Message structure
- Control functions

Profile
- Transport protocol
- Data encoding
- Hub discovery

MTypes
- Message semantics
- Arguments
- Return values

Standard Profile
- XML-RPC (= HTTP + XML)
- Hub URL in ~/samp file

Example MTypes:
- image.load.fits
- table.highlight.row
- coords.sky.pointAt
- ...
3-Layer Architecture

Abstract API
- Data types
- Message structure
- Control functions

Profile
- Transport protocol
- Data encoding
- Hub discovery

MTypes
- Message semantics
- Arguments
- Return values

Standard Profile
- XML-RPC (= HTTP + XML)
- Hub URL in ~/.samp file

Web Profile
- XML-RPC
- Hub URL on well-known port
- Web-friendly HTTP server

Examples:
- image.load.fits
- table.highlight.row
- coords.sky.pointAt
- ...
Web Profile

Like Standard Profile (uses XML-RPC) but:

- Hub server resides on well-known port
- Hub HTTP server configured so browser allows cross-domain access
  - W3C Cross Origin Resource Sharing — JavaScript, newer browsers
  - /crossdomain.xml — Flash, Java, Silverlight, nearly all browsers
  - /clientaccesspolicy.xml — Silverlight, IE
- Callbacks use reverse HTTP / “Long Poll”
- Hub provides proxy service for external URLs

⇒ Browser-based clients can talk to desktop Hub
Cross-domain restrictions lifted for Hub HTTP server

- Untrusted web applications can access the Hub. What can they do?
  - Request to register with Hub
  - *If refused:*
    - nothing
  - *If granted:*
    - send/receive SAMP messages
    - read cross-domain URLs

- Address security concerns:
  1. Require explicit user approval for registration
     - Popup dialogue at registration time
  2. Ensure actions when registered are fairly harmless
     - Restrict message types to list with known harmless semantics
     - Restrict local URLs to ones referenced in previous SAMP messages

⇒ Hostile web apps can only take action if the user is explicitly careless. Even in this case, the worst attacks are fairly harmless.
Register...
Register...
### VizieR Result Page (ready for Bookmark)

**Simple Target**

<table>
<thead>
<tr>
<th>Target Name (resolved by Simbad)</th>
<th>Target dimension</th>
<th>Submit Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear 300.0 - 29.0</td>
<td>136 arcmin</td>
<td></td>
</tr>
</tbody>
</table>

- Disconnect from SAMP
- Broadcast results table

---

#### VIII/81A/sumss2_1

- Sydney University Molonglo Sky Survey (SUMSS V21)
- \[\text{Maschke}, 2008\]

The SUMSS Catalog, Version 2.1 (2008-03-11) (211063 rows)

To get all details for a row, just click on the row number in the leftmost "Full" column.

<table>
<thead>
<tr>
<th>Full</th>
<th>RA2000</th>
<th>Dec2000</th>
<th>e_ra</th>
<th>e_dec</th>
<th>Sp_maj</th>
<th>e_spmaj</th>
<th>MajAxis</th>
<th>e_majaxis</th>
<th>MinAxis</th>
<th>e_minaxis</th>
<th>PA_deg</th>
<th>Mosaic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01 04 33.48</td>
<td>-88 04 57.60</td>
<td>2.3</td>
<td>10.6</td>
<td>51.7</td>
<td>45.0</td>
<td>86.6</td>
<td>Broadcast image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>01 07 58.46</td>
<td>-88 04 38.70</td>
<td>1.8</td>
<td>38.1</td>
<td>50.7</td>
<td>45.0</td>
<td>97.0</td>
<td>Broadcast image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>01 09 47.02</td>
<td>-87 16 34.50</td>
<td>1.9</td>
<td>22.3</td>
<td>52.6</td>
<td>45.4</td>
<td>117.6</td>
<td>Broadcast image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>01 12 02.76</td>
<td>-87 21 07.60</td>
<td>4.5</td>
<td>7.7</td>
<td>83.3</td>
<td>66.8</td>
<td>24.4</td>
<td>Broadcast image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>01 19 47.80</td>
<td>-86 55 20.30</td>
<td>1.0</td>
<td>36.2</td>
<td>45.0</td>
<td>45.0</td>
<td>25.7</td>
<td>Broadcast image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>01 21 31.76</td>
<td>-87 10 41.40</td>
<td>1.7</td>
<td>80.6</td>
<td>61.6</td>
<td>52.3</td>
<td>25.0</td>
<td>Broadcast image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>01 21 45.74</td>
<td>-87 39 32.10</td>
<td>2.1</td>
<td>15.9</td>
<td>52.8</td>
<td>45.1</td>
<td>117.9</td>
<td>Broadcast image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>01 24 11.45</td>
<td>-87 15 24.60</td>
<td>3.1</td>
<td>3.3</td>
<td>50.0</td>
<td>45.0</td>
<td>84.7</td>
<td>Broadcast image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>01 26 17.46</td>
<td>-87 08 01.20</td>
<td>1.7</td>
<td>56.2</td>
<td>46.8</td>
<td>45.0</td>
<td>81.8</td>
<td>Broadcast image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>01 26 30.35</td>
<td>-86 39 32.50</td>
<td>1.9</td>
<td>18.9</td>
<td>40.3</td>
<td>45.0</td>
<td>63.1</td>
<td>Broadcast image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>01 30 03.34</td>
<td>-86 38 25.30</td>
<td>1.7</td>
<td>101.9</td>
<td>47.3</td>
<td>45.8</td>
<td>44.9</td>
<td>Broadcast image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>01 30 55.78</td>
<td>-86 39 14.70</td>
<td>1.7</td>
<td>59.3</td>
<td>47.8</td>
<td>45.5</td>
<td>112.7</td>
<td>Broadcast image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Done
Web Profile Status

SAMP Web Profile **Standard**:  
- IVOA Proposed Recommendation — Recommendation expected early 2012

SAMP Web Profile **clients**:
- Experiments (JavaScript) at CDS, HEASARC, MAST, ESAC, ESO, . . .

SAMP Web Profile **Hub** now (last few weeks) deployed by default in:
- TOPCAT v3.8 *(embedded hub)*
- Aladin Beta 7.068 *(embedded hub)*
- JSAMP 1.3-1 *(standalone hub)*
- . . . more soon?

⇒ Science users will be able to make use of SAMP web applications now or soon
Summary

- SAMP is successful on the desktop
- It’s set to move into the browser

http://www.ivoa.net/samp/