What's *not* in VWRAP
A dissection of the Linden Lab "Legacy" Protocol

Joshua Bell
Linden Research, Inc.

March 23, 2010 - IETF 77 Anaheim
Assumptions...

1. Take Linden Lab's "legacy protocol" (LLLP)
2. Convert UDP message system to LLSD over HTTP
3. Publish RFC

*No! No! A Thousand Times No!!*
LLLP Overview

Currently protocol:
- Binary messages over UDP (client<->server)
- LLSD messages POSTed over HTTP(S) (client->server)
- LLSD messages over EventQueue/HTTP(S) (server->client)
- LLSD over RESTful APIs (via capabilities)

Issues:
- Bulk of traffic over UDP; non-standard reliability layer
- Non-RESTful traffic follows UDP message structure
  - Usually migrated only for TLS
- Mix of "protocol" types (REST, stream, CRUD, ...)
- Mix of conceptual types (rez, manage, chat, view, ...)
LLLP Analysis

- 475 defined message types
- 377 sent/received by viewer (others are sim-sim, etc)
- 263 map to REST semantics
  - ... with wildly varying granularity

Caveat: This is ignoring post-UDP parts of the protocol: RESTful APIs and messages never sent over UDP (~30 messages), including group chat and voice setup.
Radical Claim:

Only 23% to 33% of LLLP should be in VWRAP

(Fraction by message type; probably >90% of the message traffic would be standardized.)
But is he claiming...

... $\frac{3}{4}$ of the client/server communication should be proprietary data formats?

...Maybe!

But here's why you probably want that too.
### Message Categories/Counts in LLLP

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Properties</td>
<td>8</td>
</tr>
<tr>
<td>Administration</td>
<td>5</td>
</tr>
<tr>
<td>Agent Control</td>
<td>25</td>
</tr>
<tr>
<td>Agent Profile</td>
<td>9</td>
</tr>
<tr>
<td>Agent Properties</td>
<td>6</td>
</tr>
<tr>
<td>Asset Transport</td>
<td>11</td>
</tr>
<tr>
<td>Avatar Appearance</td>
<td>14</td>
</tr>
<tr>
<td>Build Tools</td>
<td>37</td>
</tr>
<tr>
<td>Build Tools (Land)</td>
<td>2</td>
</tr>
<tr>
<td>Communication</td>
<td>7</td>
</tr>
<tr>
<td>Connection Management</td>
<td>20</td>
</tr>
<tr>
<td>Event Profiles</td>
<td>7</td>
</tr>
<tr>
<td>Friends</td>
<td>11</td>
</tr>
<tr>
<td>Grid Status</td>
<td>2</td>
</tr>
<tr>
<td>Groups</td>
<td>40</td>
</tr>
<tr>
<td>Inventory</td>
<td>26</td>
</tr>
<tr>
<td>Parcels</td>
<td>28</td>
</tr>
<tr>
<td>&quot;Pick&quot; Profiles</td>
<td>4</td>
</tr>
<tr>
<td>Postcards</td>
<td>1</td>
</tr>
<tr>
<td>Region Administration</td>
<td>14</td>
</tr>
<tr>
<td>Region Properties</td>
<td>2</td>
</tr>
<tr>
<td>Rez/Derez</td>
<td>6</td>
</tr>
<tr>
<td>Scene Graph</td>
<td>23</td>
</tr>
<tr>
<td>Script UI</td>
<td>9</td>
</tr>
<tr>
<td>Search</td>
<td>29</td>
</tr>
<tr>
<td>Teleport</td>
<td>11</td>
</tr>
<tr>
<td>Textures</td>
<td>6</td>
</tr>
<tr>
<td>Virtual Currency</td>
<td>8</td>
</tr>
<tr>
<td>World Map</td>
<td>7</td>
</tr>
</tbody>
</table>
What *should* be detailed in VWRAP?

At the very least...
- Avatars, Objects
- Communication
- Connection Management
- Rez/Derez
- Scene Graph
- Teleport

Three broad protocol requirements:
- RESTful (avatars, objects)
- Reliable, mid-latency streams (control, chat, build)
- Unreliable, high-latency stream (scene graph)
Grid-Wide Services

- Search
- Events, Classifieds, "Top Picks"
- Agents, Places
- World Map

Claim: VWRAP should define common set of Grid services, but only describe each as a URL to Web page, not a service-specific sub-protocol.

URL is for a Web page not service - human readable and usable (given contemporary Web browser).
Aside: Risks

- *By not enforcing a data standard, this fractures the uni/multi/meta/omni-verse of virtual worlds!*  
- Requires that VWRAP viewers are supersets of Web browsers

Yes, *but*...

- Standardizing on Web technology (URL to page) has worked rather well on the Internet  
- Gives service providers maximum flexibility for implementation and deployment  
- Expose world services to non-VWRAP viewers
In-World Property Pages

- Agent Properties
- Object Properties
- Region Administration
- Land Administration

Claim: *VWRAP should standardize mechanism to interrogate agent/object/land/region for (optional) property page URL.*

Again: URL is for a Web page not service.
Claim: *VWRAP should not standardize agent inventory, just the Viewer/AD/RD interaction for rez/de-rez.*

Agent Domain would provide Agent Inventory
- Web page? Custom REST API? Outside of scope!
- Source of URLs for rez/sink for derez (HTML5 drag/drop?)

Allows for arbitrary hosted inventories:
- Current Region library (costumes, meeting tools)
- "Creative Commons" library (cross-grid freebies)
- Group libraries
- Store inventories (pay-on-rez?)
Viewer / Agent Domain

Do these need a standard, or just conventions?

- Postcards
- Mute Lists, Gestures
- Landmarks, Calling Cards

Or even these?

- Friends
- Groups
How about...

- Object Edit/Build Tools
- Land/Terraform Tools

But... unlike inventory, friends, etc. these require scene-graph interaction, possibly AD/RD permission checks.

So... Maybe?
Remove (with a vengeance...)

- Asset Upload/Download - *just use HTTP*
- Textures - *ditto*

*(May be hidden behind per-item capabilities for access control, etc.)*

- Others?
These hurt my brain...

Parcels
- is region subdivision first class?
- easy: "gimme a web page to inspect/edit land at X/Y"
- but: visualizing boundaries

Script UI
- Do we mandate a set of common actions any server-side logic can trigger (JavaScript alert/confirm/prompt)
- Wait for client + server-side scripting standards?

Virtual Currency
- Tourist transactions
Calls to Action

- Does it make sense to decouple this much from VWRAP?
- Populate three buckets:
  - What's in, what's out, what's deferred
- Explore:
  - Presence across virtual worlds
  - Non-RESTful protocols (XMPP? RTSP?)
  - Parcels, script UI, ...

Thanks to David W. Levine (IBM Research), for feedback on this presentation.