

# WebSocket Requirements

(draft-loreto-hybi-requirements-01)

HyBi WG, IETF-77

March 24, 2010

*Salvatore Loreto*

# Overview

---

The goal of HyBi wg is to provide an efficient / clean two-way communication channel between client and server

The communication channel will:

- Allow each side to, independently from the other, send data when is willing and ready to do it
- Rely on a single TCP connection for traffic on both the directions
- Reduce the high overhead produced by HTTP headers in each request/response

# Req.: Internationalization

---

Do we need a requirement on intern.?

- Req. 4: Textual data MUST be encoded as UTF-8

The specifications must include internationalization consideration even if it is not part of the req.

# Req.: Transport Protocol

---

- Req. 1: The WebSocket Protocols **MUST** run directly on top of a transport protocol (e.g. TCP, UDP, SCTP or DCCP)

# Req.: Data Management

---

*Message*: a block of related data with identified boundaries

- Req. 2: The WebSocket Protocols **MUST** be able to handle (send and receive) message on top of a TCP data stream.
- Req. 3: It **MUST** be possible to send a message when the total size is either unknown or exceeds a fixed buffer size.

# Req.: Handshake

---

- Req. 9: The WebSocket Client MUST be able to set up a communication channel sending to a WebSocket Server a well defined handshake.
- Req. 14: The WebSocket Server that accept to set up, with a WebSocket Client, a communication channel MUST send back to the WebSocket Client a well defined handshake.

# Req.: HTTP – WebSocket

---

- Req. 3: The WebSocket protocol **MUST** allow HTTP and WebSocket connections to be served from the same port. Consideration **MUST** be given:
  - ➔ to provide WebSocket services via modules that plug in to existing web infrastructure.
  - ➔ to making it possible and practical to implement standalone implementations of the protocol without requiring a fully conforming HTTP implementation.

# Req.: HTTP – WebSocket

---

- Req. 7: When sharing host and "well known" port with HTTP, the WebSocket protocol **MUST** be HTTP compatible until both ends have established the WebSocket protocol.
- Req. 8: The protocol **SHOULD** make it possible and practical to reuse existing HTTP components where appropriate.



# Req.: WebSocket close

---

- Req. 10: WebSocket Protocol MUST provide for Graceful close of an active WebSocket connection on request from the user Application.
- Open Issue: do we need also a requirement for Ungraceful close ?

# WebSocket extensions: where?

---

- Some really should remain in Javascript.
- Some are \*better\* if done by the browser, but it is still good to have the ability to fall back to Javascript on an older browser.  
(e.g. Multiplexing connections from a single tab/window)
- Some cannot be done from Javascript at all.  
(e.g. Opportunistically multiplexing connections from all tabs/windows to the same destinations - or proxy if there is one)

# WebSocket extensions: how?

---

# Req.: WebSocket sub-protocol

---

- Req. 11: The WebSocket Client **MUST** be able to request the server, during the handshake, to use a specific WebSocket sub-protocol.

# Req.: Security

---

- Req.17 The WebSocket Protocol MUST use the Origin-based security model commonly used by Web browsers to restrict which Web pages can contact a WebSocket sever when the WebSocket protocol is used from a Web page.
- Req.18 When used directly (not from a Web page), the WebSocket Protocol MUST use an equivalent security model.

# Req.: Security

---

- Req.19 WebSocket should be designed to be robust against cross-protocol attacks.  
The protocol design should consider and mitigate the risk presented by WebSocket clients to existing servers (including HTTP servers).  
It should also consider and mitigate the risk to WebSocket servers presented by clients for other protocols (including HTTP).

# References

---

- ◆ “Best Practices for the Use of Long Polling and Streaming in Bidirectional HTTP”  
<http://www.ietf.org/id/draft-loreto-http-bidirectional-02.txt>
- ◆ “The WebSocket protocol”  
<http://tools.ietf.org/html/draft-hixie-thewebsocketprotocol>
- ◆ “The WebSocket API”  
<http://dev.w3.org/html5/websockets/>