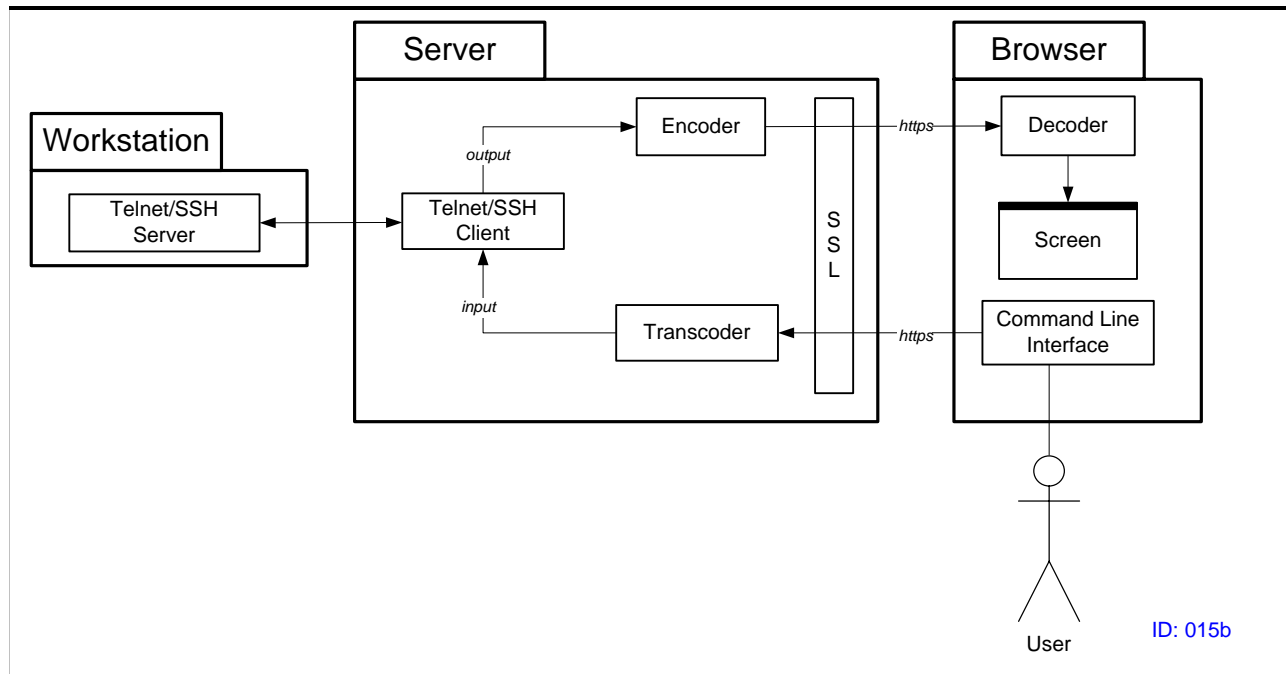


# Prototype B1

## Web Interface to Remote Telnet/SSH Client Prototype

This is a proof-of-concept project to answer questions on how the std-output of an SSH/Telnet client running on a server may be efficiently and securely relayed to a web browser. We wish to hire a consultant to analyze current approaches and design and build a test system that we can use to measure performance.

### 1. Usage Scenario



A Workstation has a standard Telnet or SSH server.

A Server is running a special-purpose Telnet/SSH Client (or similar application). This special-purpose Telnet/SSH Client delivers its output to an Encoder which relays the output in a secure manner to a Browser. The Server receives commands from the Browser and transcodes these commands into a format appropriate for the special-purpose Telnet/SSH Client using a Transcoder.

The Browser un-encrypts the Client's output and Decodes it in order to display the Telnet/SSH client's std-output in a window, called the Screen. The Browser has a Command Line Interface in which a User may issue standard operating system commands. Upon hitting "Enter", the command is sent to the Server.

An Administrator uses an Internet Explorer 6 or higher web Browser to connect to the Server. The User initiates a session by opening a web page to the Server's IP address (no authentication will be required). On the Browser, the User specifies the Workstation's IP address, Port number and connection method (SSH or Telnet). The special-purpose Telnet/SSH Client connects to the Workstation. The User may enter commands using the Command Line Interface and see the Workstation's response on the Screen.

The objective is to allow a Browser to have a Screen that is as responsive as a regular Telnet or SSH client's. In this proof-of-concept, there is no mouse-control and the Command Line Interface will be limited to alphanumeric commands; no support for arrow-keys, escape sequences, etc., is required.

## **2. System Design**

### **2.1. Encoder – Decoder Subsystem**

The Encoder-Decoder subsystem must:

- Allow for a wide variety of character sets, including color codes
- Minimize latency between Workstation output and Screen refresh

The Encoder must minimize bandwidth between Server and Browser by

- Sending only incremental changes
- Possibly compressing the changes

The Decoder must be a compact application that can run in IE 6+ browsers.

### **2.2. SSL Encryption**

All communication between the Browser and the Server must be encrypted.

### **2.3. Telnet/SSH Client**

The special-purpose Telnet/SSH Client should re-use an existing, open-source application. This application does not need to be an actual Telnet/SSH Client, but anything that provides similar functionality.

### **2.4. Transcoder**

It may be necessary to transcode UTF-8 to iso-8859-1 format if XmlHTTP Request method is used to relay the Command Line Interface text to the Telnet/SSH Client.

### **2.5. Screen**

The Screen must be a rich client, such as a Java application where changes occur quickly. The Screen must look like a standard Telnet/SSH window and be a compact application that can run in IE 6+ browsers. The Screen must display colored text as it would appear on a regular Telnet/SSH client.

## **3. Similar Applications**

The following applications are similar to the service described here. However, they each provide Telnet/SSH access to a host computer. The service described above establishes connections to remote computers (and not the host).

Application	Website	Note
AnyTerm	<a href="http://www.anyterm.org">www.anyterm.org</a>	Web-based SSH/Telnet to a server running AnyTerm application
AjaxTerm	<a href="http://antony.lesuisse.org/qweb/trac/wiki/AjaxTerm">http://antony.lesuisse.org/qweb/trac/wiki/AjaxTerm</a>	Same as AnyTerm, but using AJAX technology for client and Python 2.3 for server
Abinition	<a href="http://www.koikids.com/dashboard.htm">http://www.koikids.com/dashboard.htm</a>	Secure AJAX Telnet to server

## 4. Deliverables

The Consultant will deliver the Server and Browser components required to provide the functionality described in the Usage Scenario and System Design. The Consultant should re-use existing open-source, GPL software as much as possible.

### 4.1. Important Deliverables

The consultant must fully specify the Encoder-Decoder and Screen subsystems. We are mostly interested in these components and their design must be fully documented, including the decision criteria used in selecting the technology or approach. The Encoder-Decoder and Screen subsystems must be properly documented with the following minimal UML diagrams:

#### 4.1.A Domain Model

All components must be properly defined and described using a Domain Model and a glossary. The relationships between components must be specified.

#### 4.1.B Activity/Robustness Diagrams

A flowchart of each component's activity must be summarized in an Activity (or Robustness) diagram.

#### 4.1.C Class Diagrams

All methods and attributes for all Classes must be documented and described. Putting the documentation in the code will not be sufficient.

#### 4.1.D Sequence Diagrams

The behavior between components must be described through sequence diagrams