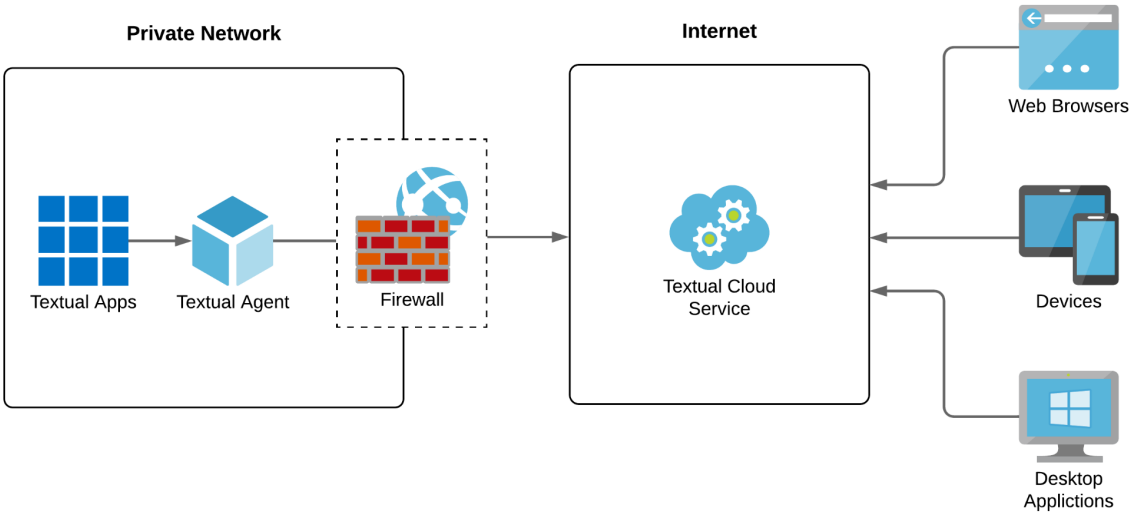
Textual Cloud Service

Textual apps running within an organisation can be made instantly web facing applications via a “Textual Cloud Service” and “Textual Agent”.

Architecture

The agent software maintains an outgoing TCP/IP connection to the cloud service, which is generally permitted by firewall rules. The cloud service manages the remote applications and serves the front-end to devices on the internet.



Textual Agent

In order to move textual applications to the web, clients can install the “Textual Agent” software which will be built in Python and distributed as a standard Pypi module.

The agent can be configured to serve a single or multiple textual apps. It maintains a TCP/IP connection to the cloud service running the Websocket protocol. There are a number of benefits to this approach:

- **Works over proxies** - The websocket protocol is designed to cooperate with network infrastructure such as proxies.
- **Bypasses firewalls** - Firewalls are generally configured to allow outgoing TCP/IP connections.
- **Encrypted** - Connections are encrypted with industry standards.
- **Compressed** - Data is compressed on the fly.

The Textual cloud service is responsible for communicating with the agent software to launch Textual instances on demand. The agent sends updates from the Textual app while receiving mouse and keyboard events from the cloud service.

The system is analogous to a remote desktop service such as VNC, but with considerably less overhead due to the low-graphics nature of terminal apps.

Front End

In addition to managing the websocket connections with the agent software, the cloud service is also a web server which reconstructs the terminal based applications within the web-browser.

The user experience will be broadly similar as if they were operating the terminal on the local device. It is expected that the network will introduce a latency in the order of around 100ms, well below the threshold that the user would notice it.

The web front-end can also add authentication for managed users, at a minimum requiring a username and password login to run the application, and likely also Two Factor Authentication.