Direct first fetches the main HTML page directly from the web server, parses it and loads the other objects required to render the page using multiple (parallel) HTTP connections. The page is then rendered locally on the device. On the other hand, CB connects to the cloud proxy and after a Wait Time receives a compact version of the page from the proxy in a proprietary format (which we call Cloud Browser Markup Language (CBML)). During the Wait Time, the cloud proxy employs data compaction techniques on the page like reformatting, re-sizing images, compression for small-screen rendering. The client extracts the CBML and renders the page.

CB offloads JS processing to the cloud proxy. It supports two modes of operation for pages with JS. In older versions CB supports a mode where all JS in a page are run for a timeout of 5s, after which they are stopped and a CBML is sent to the client. There are many web pages that change their content either automatically (e.g., timer-based) or through user-interaction (e.g., user-clicks) using JS. The older version could not accurately render many pages that change content through long-running JS (> 5s) and does not support rich interactivity. In newer versions CB supports a second mode where the client opens a secure, persistent connection with the cloud proxy and the proxy continuously runs the JS, pushes different bursts of objects at different times whenever the page changes. This approach helps CB accurately render most pages. Even with the second approach, in our experiments we have seen CB failing to render some pages that continuously download objects through JS that run forever. We use the second mode of CB in our experiments since it renders most pages accurately.