Evaluation Factors for BrowserGap

1. No local agent or client required.
2. We use the latest industry-standard Chrome engine to render HTML and keep up to date with the latest changes in HTML5 and web platform features.
3. We don't currently support PDF or Flash, but have a roadmap for displaying PDF securely remotely.
4. SaaS applications such as Office 365 and G Suite are Fully Supported. They behave exactly the same as if you were using your regular consumer insecure browser, with no additional latency.
5. We don't currently provide the option to download any files from the public internet. We have a roadmap to explore secure solutions to this in future.
6. We provide full cut and paste abilities. You can cut text from the page via a special popup window, and you can also paste text into the page. If desired you can also completely disable these features as well.
7. We are OS independent. Most of our deployments use Linux (Such as Centos or RedHat) and we can also be configured to run atop Windows or even Mac OS hardware.
8. We use full VMs for browser isolation. A single tenant per VM, and typically a customer will want to occupy multiple VMs to provide for their organization. We take care of provisioning and management of these VMs.
9. You can reset the browser session to a known good state by clearing the cache, history and cookies manually, or you can set this to happen every time the browser starts up. You can also choose to keep the cookies (to keep you logged in to regular sites) while clearing the history and cache. You can also open incognito mode tabs that have no history, cache or cookies associated with them.
10. Web content such as YouTube is first converted to pixels before being sent to your network and device. We do not stream the remote media (such as video and audio) directly, we render it remotely first, then convert it to pixels and compress it before sending to you. You can watch YouTube normally albeit at somewhat reduce resolution, and we do not

currently support audio. We have a proof-of-concept of recording audio from a web page, and a roadmap for streaming this recorded audio in the future. A workaround for now is turning on subtitles on Youtube videos that support it.

11. The bandwidth used is typically less than what you normally experience using an Consumer-gradie Insecure Browser. This is because you don't download any of the page's assets (such as JavaScript files, CSS stylesheets, images and other objects). Instead we download them for you remotely, render the application, convert it to pixels and send it to you. Many web sites these days contain megabytes of assets (images, styles, 3rd-party JavaScript for tracking, etc) and by saving you from needing to download any of that, the bandwidth used often ends up less. Depending on the compression achieved for the stream, and how small the remote web page is, the bandwidth can sometimes be more, but often it is 10 to 40x less than normal. Especially on mobile, where the screen size is smaller, the bandwidth is significantly reduced. Because we convert to pixels and compress, the image quality is not as high as if you were viewing an uncompressed image. We have a roadmap to allow you to set the compression level of the image stream in future. In terms of protocol, we use a proprietary SSL encrypted protocol to communicate between your device and our servers.
12. Web conferencing applications such as WebEx are not currently supported. We are exploring a roadmap for the future for securely enabling access to the microphone and camera but for now this introduces too many complexities and risks. Alternately, a workaround that is somewhat risky is to whitelist such services and run them natively on local browsers.
13. Our service is cloud-based. However we are also cloud-agnostic and have deployments on AWS, GCP and IBM/BLUEMIX cloud IaaS platforms. If necessary we can also deploy straight to your data center, or onto custom secure hardware and bare metal servers. Typically we will locate your dedicated VMs in a geographic region that is convenient for you, and provision dedicated bandwidth (typically 1Gbps). We serve multiple customers, but each customer gets their own servers, meaning each deployment is single-tenant. We can offer flexible and hybrid

deployments as a mixture of on-premise and cloud depending on your users and locations.

14. Mobile users are fully supported (and even encouraged). Using on mobile is a delight because it often saves bandwidth. There are a few issues with typing on mobile (because of the virtual keyboard, especially when composing text in languages other than English), but these have mostly been worked out to provide a very familiar typing experience as if you were using a regular Consumer-grade insecure browser (CIB) to render the page. We have a roadmap to resolve the remaining typing bugs in the medium term. Mobile users get mobile versions of the pages just like they would using their regular CIB.
15. The isolation model is the strongest, the most complete and wholly total. We do not transmit any DOM content, only pixels. We have explored a DOM mirroring approach (involving filtering markup), however we assessed that the small gains to bandwidth (which are already often less than regular CIB browsing), are far outweighed by the many, and extensive usability issues that occur when any given web site or app is "filtered". What typically happens is that style and appearance often become unfamiliar, and behaviour often goes completely out the window, making the formerly-familiar web site or app completely unusable without significant adjustment to the "altered interaction dynamics" that filtering the markup has created. In addition to that the systems needed to filter, transmit and reconstruct web pages like this are orders of magnitude more complicated than those that simply convert the remotely rendered page to pixels. Because of the additional complexity of such an approach it is inevitable that bugs and attack vectors will creep in. As well as that, no filtering scheme can be assumed to be perfect, so it is possible that "filtering" and mirroring the DOM will fail to detect and thus transmit some exploits and zero-days to the user, breaking the intended security perimeter of the system. This is highly undesirable and for these reasons, we doubled down on our initial approach of using Interactive Image™ technology, rather than investing further into research of the DOM filtering approach we tested later, as described above. Our motivation for testing DOM filtering was simply to see "how far can we go" in reducing bandwidth, because there was a line of thinking that some customer in developing countries (or in remote

sites where bandwidth is expensive over satellite) would want the convenience of using the internet without paying dollars on the megabyte for the privilege. In the end, though, it seems our strongest market, at least for now, is security-conscious (instead of bandwidth-conscious) organizations and personnel, so we have doubled down on our singular original mission to provide secure remote browsers, not just explore all the possibilities that such technology may be applied to, interesting as they are.

16. We do not have any SWG partnership. If you would like to contact us about that, please go ahead. As stated earlier, we have a roadmap for enabling secure access to files from the public internet. Our main idea is rendering these files remotely and transmitting the pixels. While unlikely we would adopt any approach that results in actual (and exploitable) file content being sent to our customers, are are happy to research and explore solutions from SWG vendors, in case this approach becomes viably secure. For SWG consumers, we are happy to attempt to replace your existing SWG capabilities with regards to web content, respecting the limitations we already described as to not currently supporting PDF, and having a roadmap to do so in the future.
17. While we have no specific roadmap item for remotely rendered email integration, we do fully support all web mail providers (such as G Suite, Outlook.com, Gmail.com, etc). And we are happy to explore a possible solution for your organization regarding remotely rendered email integration. It should at least be possible in most cases, the trick in this area of remotely rendered email integration will be in getting the integration secure and convenient and working together well with your current systems.
18. We do not support the reverse-direction isolation of enterprise mobile apps, nor Windows apps. However, we can be configured to serve your internal web applications in a secure fashion. This is quite an extensive project however.