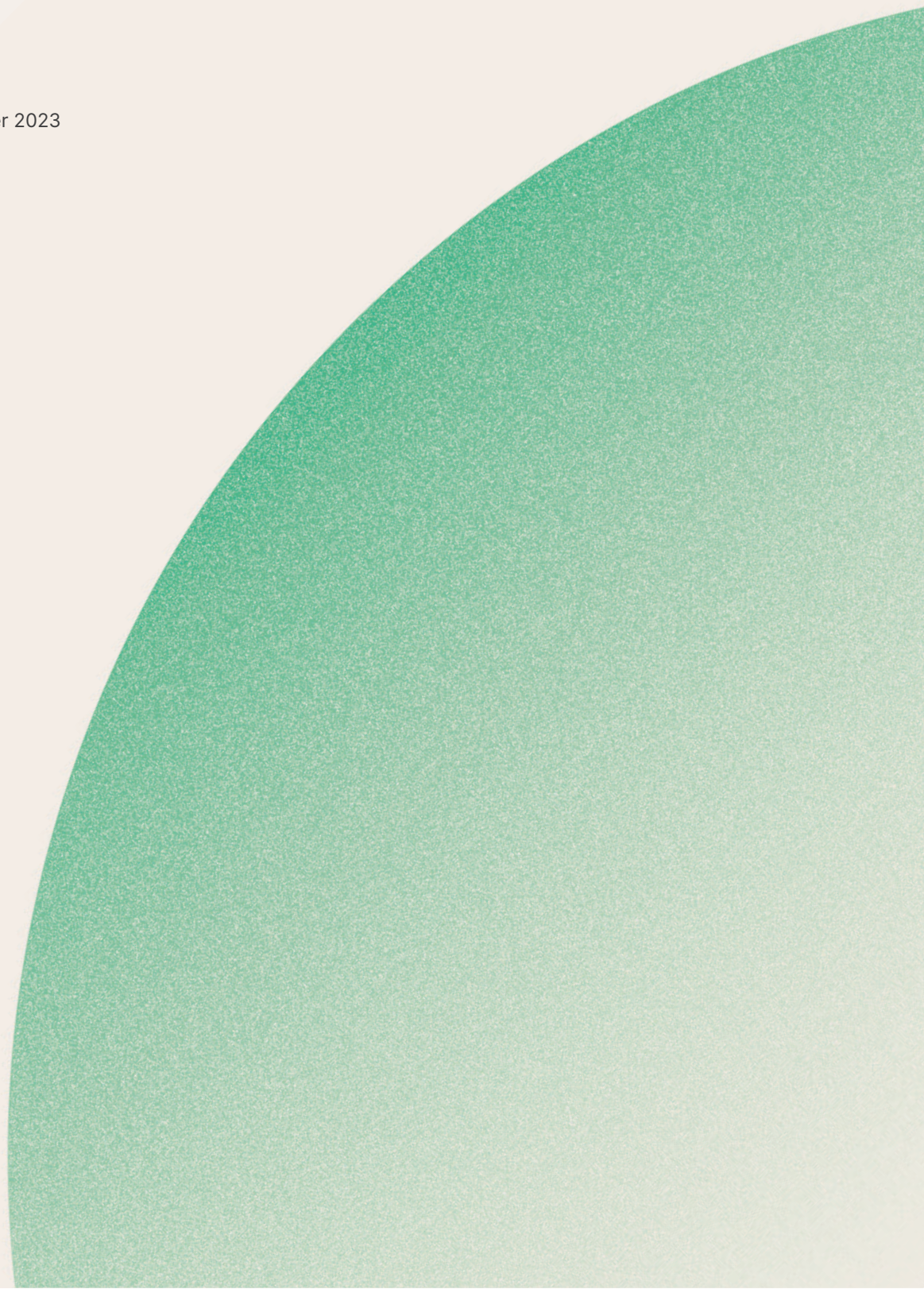
Island

Browser Self Protection and Attack Surface Reduction

White paper | September 2023

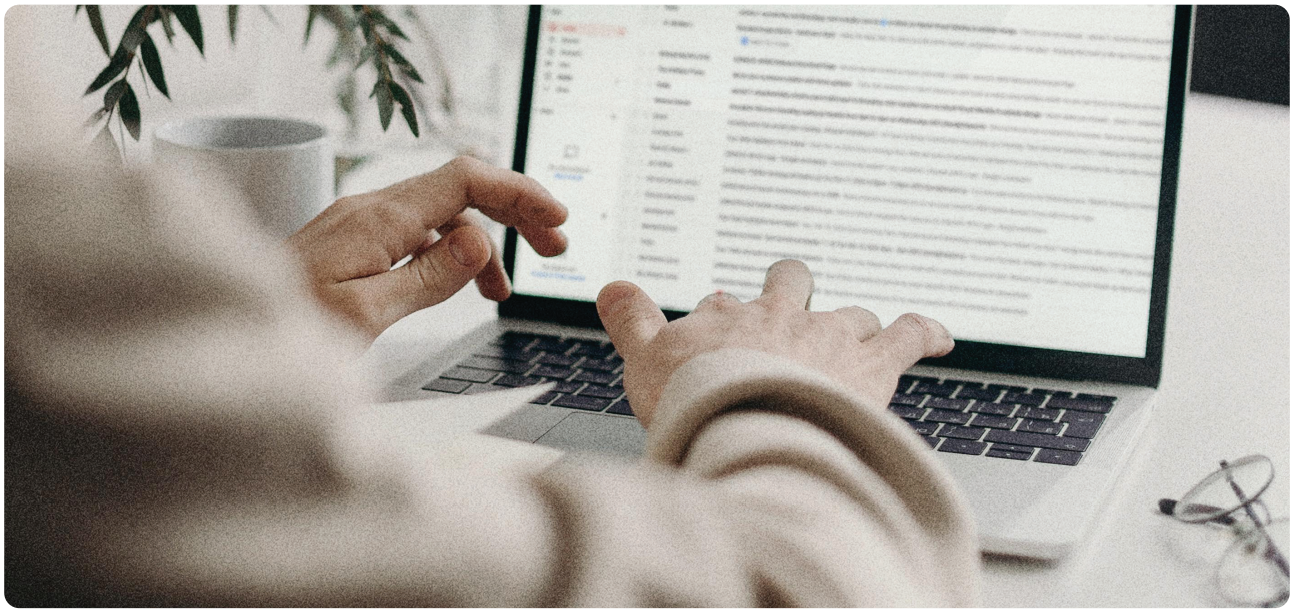


Overview

By their very nature, web browsers are built to run third-party code directly on the endpoint. The majority of these application engagements come without verification, creating fertile ground for attackers. While most enterprises perform continuous efforts focused on educating users to minimize risky behaviors, not even the best education can prevent all of today's sophisticated threats. Phishing, malware, ransomware, and many other threats often begin with a web-based engagement. The consumer browser is an unwilling participant in these engagements, so enterprises need to layer control after control around these web browsers to insulate them from danger. The traditional answer? Stand up a stack of controls in front of these consumer browsers to protect the browsing experience.

Protecting web usage typically starts with web gateway (proxy) infrastructure for many organizations. While these were practical approaches in years past, the growth of encrypted traffic (SSL) and sophisticated threats such as browser code injection leave existing proxies and SASE solutions unable to protect end-users adequately. These attacks and defenses leave the user's consumer browser (which cannot defend itself from such techniques) subject to exploitation.

In an attempt to combat these attacks, many organizations explored using Remote Browser Isolation (RBI) technologies to augment existing proxy resources. The concept behind RBI is to force uncategorized or untrusted web traffic into a virtualized cloud environment for remote execution. As the user engages web content in this way, the site is rendered over a video stream (often HTML5) back to the user's consumer browser. In principle, the user is protected from any harmful content.



Technological Challenges of Remote Browsers

On the surface, a remote vehicle to execute potentially dangerous web content for the user seems like a viable protection strategy. However, this approach is fraught with its own set of challenges. To begin with, it is not palatable to force all users' traffic through RBI. Why? Because the user experience and performance simply are not acceptable for everyday use. Rendering the content remotely and streaming it back to the user adds noticeable lag and visual imperfections. Thus, because the experience is generally poor, RBI technologies are usually invoked only in specific situations. For example, RBI is often used where content must be isolated for potentially malicious web content on untrusted sites. This means that only a tiny subset of traffic (usually 1-2%) is passed through RBI technologies in the first place. By reducing the scope of where RBI is engaged, the organization can attempt to minimize the concern over end-user friction. Of course, this leaves a significant gap for sites categorized as collaboration, file sharing, social media, and others. In these cases, the web traffic is never passed through an RBI solution, yet risky content still exists. Further, Single Page Applications (SPA) and HTML5 canvas rendering are meant to be executed locally and would not be candidates for passing through RBI solutions. Put simply; the attack surface is much larger than the exploitation footprint protected by RBI. These limitations call into question the value of the investment.

RBI technologies are also quite limited where other types of common browser-related attack techniques might be employed, such as:

- Sophisticated Phishing Attacks
- Exfiltration Of Data
- Man-in-the-Middle Attacks
- Malicious Extension Exploitation
- Embedded Malicious Document Content
- Localized Browser Tampering
- Man-in-the-Browser Attacks

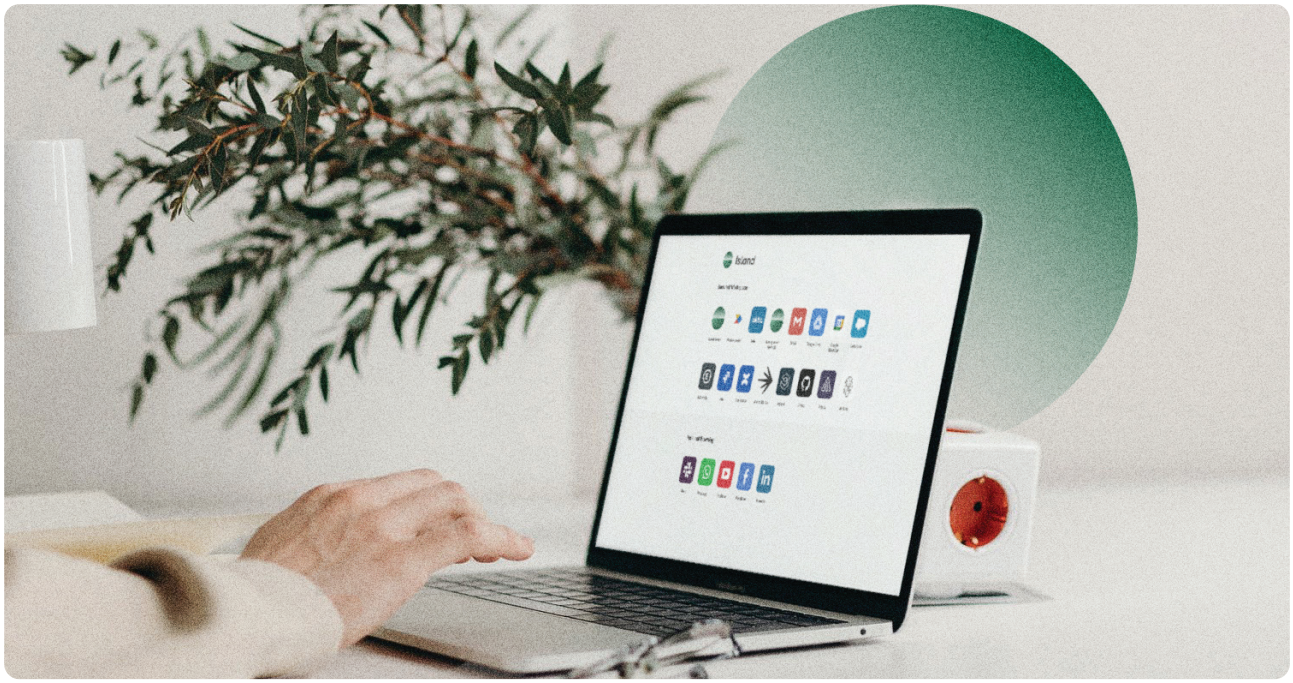
In each of these cases above, RBI either has no role in protecting against the attack or cannot offer full protection because it's only used for a fraction of the web traffic.

Use Case Limitations

As previously mentioned, RBI is most often invoked for web traffic destined for suspicious sites that might cause remote browser code injection or attempts to phish users leveraging fake sites. However, this limited usage of RBI means that it cannot fulfill more valuable browser-based use cases that may be important to the organization, such as:

- SaaS and Internal Web Application Protection
- Contractor and Third-Party Provisioning/Protection
- Call-Center Worker Governance
- Bring-Your-Own-Device Policies
- Privileged User Protection

A web browsing experience is often central to the needs of such use-cases. Yet it is essential to note that RBI offers little for these scenarios. To begin with, the necessary traffic for these needs usually isn't routed to RBI. Further, RBI just isn't built to solve these challenges and lacks the mechanics required to add value to these core browsing use-cases.



Rethinking Browser Isolation Outcomes

The proliferation of threats leveraging the web has piqued interest in RBI technologies. However, RBI provides limited solutions solving only the symptoms like browser exploits and remote code injection. This pattern is all too frequent in cybersecurity, where vendors build solutions to address a handful of symptoms without addressing the core problem. Here, the core problem is that consumer browsers were never developed to accommodate the needs of the enterprise.

What if the browser was built for the enterprise? This is precisely what Island considered as we created the industry's first Enterprise Browser. As users engage with all corners of the web, Island's innovations deliver a true self-protecting browser to ensure that all engagements are safe. These capabilities deliver far more effective outcomes than clunky RBI solutions while doing so in a native browsing experience. This ensures that users have complete protection without the negative impacts on their experience. Let's dive in more.

As the inventor of Remote Browser Isolation, Island co-founder and CTO Dan Amiga has extensive experience with browser technologies and a deep understanding of the pitfalls. From the beginning, Island put significant expertise and effort into delivering the advantages of browser isolation without the need for the "remote" part.

At its core, Island is built on the Chromium project. This open-source project is the basis of modern browsers like Google Chrome, Microsoft Edge, and many others. Using Chromium ensures a web browsing experience that end-users are familiar with and the snappy performance they expect. The ubiquity of Chromium also makes it a favored target for attackers and malware developers. Security researchers and the Chromium project team go to great lengths to patch known vulnerabilities, but there will always be zero-day exploits.

One example is the Chromium Just-In-Time (JIT) compiler. This mechanism improves web application performance, but it has been at the core of many recent zero-day vulnerabilities across all browsers which leverage Chromium. Rather than attempting unnatural techniques such as Remote Browser Isolation, with all its shortcomings discussed above, Island took a different approach by going straight to the source of the problem. While the JIT was originally designed to improve performance, those performance improvements are quite modest on modern hardware. The Enterprise Browser disables the JIT as a default configuration to eliminate a source of vulnerability with undetectable performance impact. Disabling the JIT also disables WebAssembly, further reducing the attack surface. Island offers policy-driven capabilities to selectively enable the JIT and WebAssembly in the rare situations that require it.

A similar approach is used with other browser components that are vulnerable to exploitation. Island will detect potentially malicious javascript from untrusted web destinations and dynamically block execution across over a dozen APIs and modules, including WebRTC, WebGL, and others. Again, these configurations are policy-driven and offer complete flexibility to ensure that any trusted, enterprise apps work as expected while reducing the browser attack surface for malicious attackers. Island also leverages several additional protective capabilities by enabling Arbitrary Code Guard, Control Flow Enforcement, and Control Flow Guard. Each of these capabilities ensures that arbitrary code cannot be injected directly in an attempt to manipulate the memory or execution flow of the Enterprise Browser.

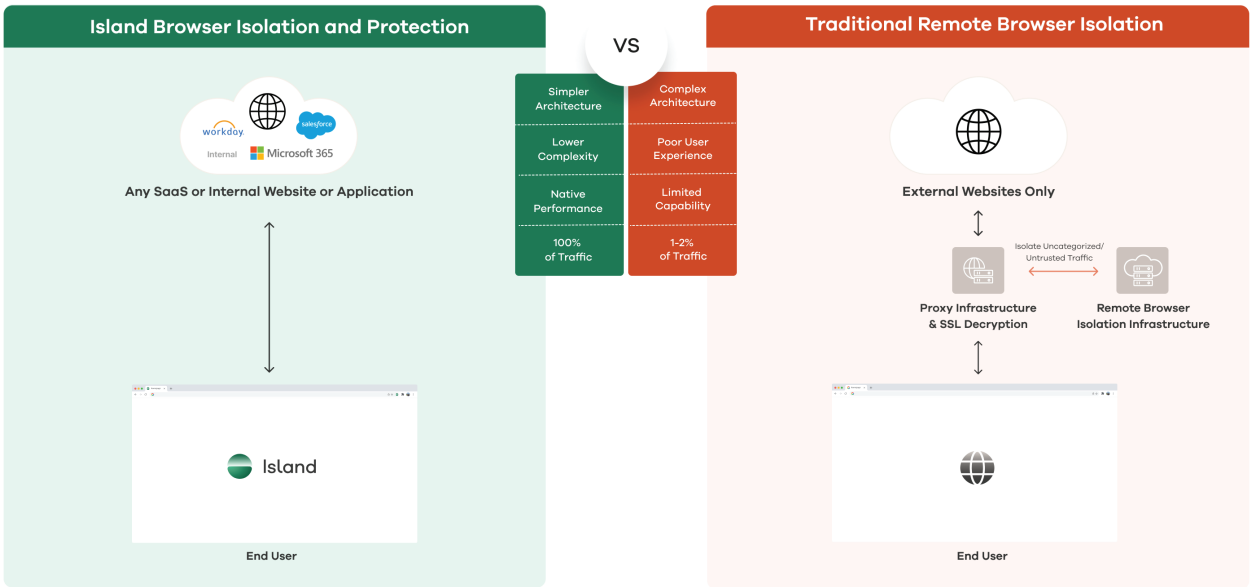
By delivering Browser Isolation directly into the Enterprise Browser, Island removed the most significant areas of browser vulnerability and added capabilities to protect against exploits. As previously mentioned, this solves the core problem of advanced web threats rather than the symptoms. These alone negate the need for Remote Browser Isolation solutions by preventing malicious code execution directly within the browser.

Creating a Self Protecting Browser

While Island has embedded browser isolation capabilities directly into the browser, delivering a safe browsing experience must go deeper. Thus, Island pioneered the self-protecting browser that goes beyond browser isolation. This unique approach protects the browser, applications, and data both from external and internal (or local) threats. Below are a few additional capabilities that Island delivers within the Enterprise Browser:

- **Device Posture Assessment -** Island offers deep inspection of the device it's running on to evaluate access policies. For example, an organization may restrict access to critical applications if a device is running an outdated OS version or lacks full disk encryption.
- **Man-in-the-Middle Protection -** Island is the first browser to provide policy-driven capabilities to recognize when an untrusted man-in-the-middle technique is employed. This allows the organization to completely prevent data theft by a man-in-the-middle attack.
- **Document Isolation -** With built-in secure storage and document viewer, Island provides a facility to allow interaction with a document without the risk of malicious embedded code being executed on the desktop. Organizations can also redirect document downloads to their preferred secure cloud storage location, by policy.
- **Malicious Extension Protection -** By controlling the entire browsing experience, Island also includes oversight of extension usage. This gives the organization the power to control which extensions are allowed and which are not, by policy. In addition, Island's Extension Guard can ensure that critical applications and data are protected from extensions where required.
- **Local Tamper Prevention -** Any attempt to modify the Island executable, its memory footprint, or the libraries it calls on will alert administrators and completely disable the browser. This is essential for protecting against advanced malware attacks and sophisticated insider threats.

- **Encrypted Browser Data -** Island encrypts all local data stores to prevent exfiltration of cookies, cache, or stored passwords. This extends to documents as well, with local secure storage for documents that are only viewable by using the Enterprise Browser.
  Enterprise Browser.
- **Man-in-the-Browser Protection -** By leveraging many of the core technological capabilities within Island's browser isolation, the Enterprise Browser delivers native man-in-the-browser protection. This ensures that attempts to insert code impersonating a legitimate site are stopped before malicious code is rendered.
- **Anti-Phishing -** Island built a unique facility directly within the browser to protect users' credentials against phishing attempts. With a combination of web classification, risk scoring, and enterprise domain awareness, users are warned and stopped from entering their credentials if they navigate to a phony site.
- **Password Manager -** Island further protects credentials through its integrated enterprise password manager. This makes it easy for users to follow the best-practice of creating unique, complex passwords for each site or application.
- **Keylogger Protection -** Island protects users against malicious keyloggers by continuously filling the keystroke buffer with random characters. Even if an attacker is successful at installing a keylogger, the data they intercept is meaningless.
- **Web Categorization & Risk Scores -** Categorizing web content for safe browsing has been a hallmark of web proxy technologies for years. Island simply embeds web categorization and reputation scoring directly into the browser for safe browsing and compliance needs.
- **Malware Inspection -** Island Enterprise Browser has built-in malware inspection to ensure that all files uploaded or downloaded from any web destination can be inspected as policy requires. Island also offers integration with third-party EPP platforms (e.g., CrowdStrike, OPSWAT, others) for malware inspection.



The Takeaway

Protecting users and the organization's most valuable application resources is vitally important. While Remote Browser Isolation technologies were an interesting concept many years ago, their adoption never made it to the mainstream. Real-world implementations of RBI are complex, limited in outcomes, and generally deliver a poor end-user experience. They are designed to address a few symptoms of browser vulnerabilities yet do not address the core problems.

> **Island has taken the outcomes promised by RBI technologies and brought them natively into the Enterprise Browser. Instead of solving symptoms, Island went to the core of the problem by building a browser specifically for the enterprise.**

Yet Island didn't stop with browser isolation alone; the Enterprise Browser delivers a full spectrum of capabilities to protect against the widest range of threats. It boils down to a simple question, why offer a subpar remote browsing experience if you can deliver an easier, more complete, and natural experience locally?

	Island Enterprise Browser	Remote Browser Isolation
Performance	Native Browser Performance	Poor Performance
Impact on UX	Natural User Experience	Unpleasant User Experience
Traffic Coverage	All Traffic	1-2% of Traffic
Anti Exploitation	Proactive Built-In Exploit Prevention	Remote Execution of Content
Phishing Protection	Domain Misuse Prevention	Render Site Remotely as Read-Only for Uncategorized Traffic
Password Manager	Integrated Enterprise Password Manager	None (requires third-party service & extension)
Man-in-the-Middle Protection	Complete Man-in-the-Middle Protection	None
Man-in-the-Browser Protection	Complete Man-in-the-Browser Protection	None
Malware & Ransomware Protection	File scanning for upload and downloads to block malicious payloads	Limited
Extension Protection	Full Extension Control and Protection	None
Device Posture Support	Full Device Posture Assessment for Policy Driven Decisions	None
Document Isolation	Full Localized Document Isolation with Complete File Engagement	Rendering of Content in Cloud with No Engagement
Secure Storage	Built-in Secure Storage For Full File Engagement	No Secure Storage
Last Mile Controls	Full-Last Mile Control for Natural Application Protection and Interaction	No Last Mile Controls
Industry Trend	The Future	The Past