

# Flynet Vitality Brief: The Key to Accessing your Mainframe Application When it has Moved to the Cloud



As more organizations consider exporting their mainframe workload from either their own data center or third party hosted solution, to one of the growing number of Z series cloud workload vendors (IBM Cloud, Kyndryl, Dual Run -GCP), there are a few things you need to consider.

In the last few weeks there have been more articles about mainframes and legacy workloads in the cloud than in the past five years. There are a number of considerations at play, the most underappreciated of which is often access.

## Public Infrastructure

Terminal connections are very sensitive to any break in connection. Any interruption to the connection will result in it being dropped, essentially the terminal session is lost. Typical desktop terminal emulators have no contingency for this type of problem. Quite simply they were never designed to exist in such a stateless universe as the internet- resulting in disgruntled users and orphaned mainframe sessions.

Traversing public infrastructure is inherently problematic for mainframe application access. To solve this, you need a server-based web-delivered terminal emulation solution that conducts a zero-touch conversion from your traditional 3270 terminal to a stateless pure HTML front end application.

By co-locating your web-based terminal emulation server in the same cloud setting as your mainframe workload you will be, in effect, converting your mainframe application to a web application on the fly, enabling the application with all of the stateless benefits of a true web citizen.

What's more, it will look the same and work the same as your old terminal emulator. Using web-based terminal emulation will also protect you from the fragility of public infrastructure traversal and be available securely in any web browser of your corporate choosing.

## Performance

One of the key attributes of mainframe applications is their performance, moving the workload to the cloud can add latency in many ways:

1. Non-native workload latency. With some cloud vendors the computer power is not Z series based (commodity hardware is used in conjunction with software mainframes like TMAX soft, LZ Labs, Dual Run or containerization), so compromises have to be made which can often reduce the performance of the original application.
2. Split workload latency, some cloud vendors offer the ability to offload just part of your mainframe application, typically the database. In these scenarios you cannot avoid the latency of distance introduced between the front-end application and the newly located cloud based back-end data access.
3. Bottom-up distance latency, purely down to distance and speed, your new workload location will not benefit from any co-location that your original architecture permitted between workforce and mainframe.
4. Top Down 'Remote' Working Latency, this has become an issue since the pandemic, with a large part of the workforce now working at least some of the time from non-mainframe co-located places, with varying degrees of connectivity, both in terms of bandwidth and robustness.

From a bottom-up perspective some of these latency issues are simply the cost of the technology strategy choice and at the application level there may seem little you can do to mitigate it. However, a top-down approach might be to choose a web based terminal emulation solution that proactively minimizes the amount of data sent between the host application and client.

Also, reducing the data weight of terminal emulation access improves the user experience for remote users and for those users that may wish to use tablet or mobile devices on less robust/persistent connections like Wi-Fi and cellular.

**An important aside**, any applications that are delivered via a web browser inherits the performance benefits of the browser wars, technology battles fought by Microsoft, Google and Apple. Whom have seemingly bottomless budgets to spend on trying to make their respective web browser faster than that of their rivals. The result being that web browsers have become the fastest applications on the planet.

## Cyber Security

Some organizations are unfortunately using plain text telnet to connect to their mainframe applications, believing that their organizational corporate LAN is safe.

Most terminal emulators are rarely updated or patched by their vendors, so they tend to be the least secure application on the corporate desktop. Most recently the very popular open-source desktop emulator PuTTY was discovered to have been regularly used as a breach point for advanced and persistent threat hacker groups, such as the North Korean State backed Lazarus.

Over the last decade there have been countless actors that have taken advantage of the rarely updated application that has the key to the front door of most organizational critical business systems. It stands to reason that when moving the workload to the cloud you would want to improve that access security. This is an opportunity to embrace Multi Factor authentication (SAML, OAUTH, Open Id) and similar federated identity management methodologies. Most classic terminal emulators aren't designed for the internet, so don't have that kind of modern access security.

This is another great reason to examine the advantages of a modern web based terminal emulation solution. Most support multi-factor and federated identity solutions out of the box, or commonly via security API's, immediately reducing the threat surface by employing the latest multi gated authentication and authorization solutions.

Most modern web-based terminal emulation solutions support the latest levels of encryption HTTPS, TLS 1.3 and SSH. Moreover, they are designed to work natively in web hardened environments where firewalling, DMZ and micro segmentation is in place. The architecture of most web based terminal emulation solutions also allow you to lock down your mainframe even further with whitelisting- reducing access to your mainframe to purely the web based emulation server, which greatly reduces the threat surface on your mainframe

## Web 2.0

As more and more organizations are having to support users working in any setting, be it the office, on the move or at home, the relevance of the 2.0 desktop is becoming a real point of advantage.

A true web 2.0 desktop employs applications that work securely and natively in HTML over the web. Applications like MS Office 365, Outlook and Salesforce are all good examples of this, enabling a user to sit down anywhere, including low bandwidth environments, and get to work. Removing the need for 1000's of feeds of super high bandwidth bi-directional video cluttering up the internet, typically seen with



applications like Citrix and similar hyper visor tools. They were great on corporate LAN's but are an unnecessary burden on public infrastructure.

Your web-based terminal emulation solution should be a true well-behaved web citizen, providing the same functional experience as a desktop solution, but delivered securely in pure HTML in a true Web 2.0 setting.

### **Maintenance and Support**

The terminal emulation market is notoriously poor for support. New threats emerge all the time, LogJ, the Solar Winds hack etc. You never really get a sense of whether your emulator is safe. Decades will pass without an update, let alone a security patch.

Modern web-based terminal emulation solutions have to be agile in this area, but they are also commonly advantaged from two other benefits, firstly they run inside web servers, the most common enterprise web servers are some of the first to be patched against emergent threats or failings. The second benefit is the web browser, commonly the first applications to receive updates for both performance and security.

All of this comes together to ensure that your web based terminal emulation solution is the most performant and secure solution for accessing your mainframe workload in the cloud

### **In Conclusion**

Don't imagine that getting the workload to the cloud is the only problem to solve, if you rely on the old methods of access your help desk will be the hottest place in your organization. You wouldn't expect a Formula 1 car to do well in the off-road setting of the Paris to Dakar Rally, so don't expect a traditional desktop terminal emulator to do well in the equivalent internet setting.

A modern, web-based terminal emulation solution will not only make putting the mainframe application in the cloud workable, it will also dramatically improve the security, increase the performance and allow your key business systems to co-exist in your web 2.0 desktop. Best of all, the time to value and enterprise readiness is commonly less than an hour.

[Learn More](#)

[Contact](#)