HOSTBRIDGE®
TECHNOLOGY

Mainframe Optimization

Connecting the Dots for Activity Correlation

By **S. Michael Benson**

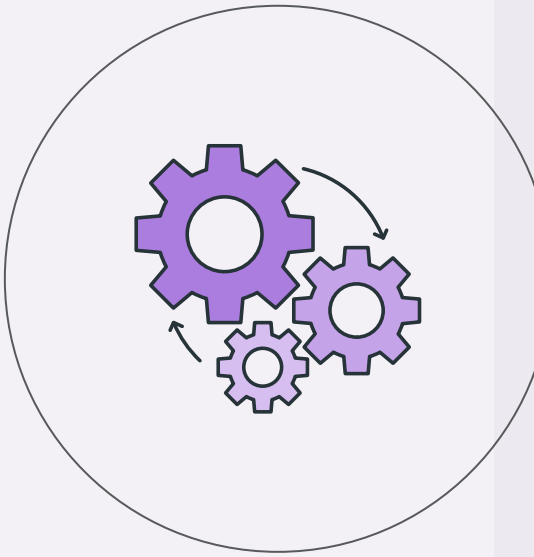
Executive Summary

Many enterprises that run core, legacy applications are busy automating mainframe business processes and integrating them with hybrid applications. As they do so, non-mainframe sources drive an increasing share of mainframe transaction volume. Effective mainframe optimization is critical to allowing the mainframe to fully participate in a hybrid IT strategy.

Optimization is complex, however, because the source of mainframe transaction volume from non-mainframe sources is difficult to know. Hybrid apps and automations often interact with the mainframe using suboptimal methods. Mainframe operations can easily monitor and measure workload, But it is difficult to manage and optimize because the source is difficult to determine. This lack of visibility has caused some enterprises to unnecessarily embark on legacy re-platforming initiatives, or guess at how to optimize.

Integration analytics allow the IT group to connect the dots by correlating non-mainframe requests to mainframe activity. With integration analytics it's possible to assess the operational impact of hybrid apps and automations on the mainframe. Using integration analytics data, dashboards identify suboptimal interactions and their sources, simplifying optimization. The results are better performing hybrid apps and automations that efficiently interact with the mainframe, keeping costs down.

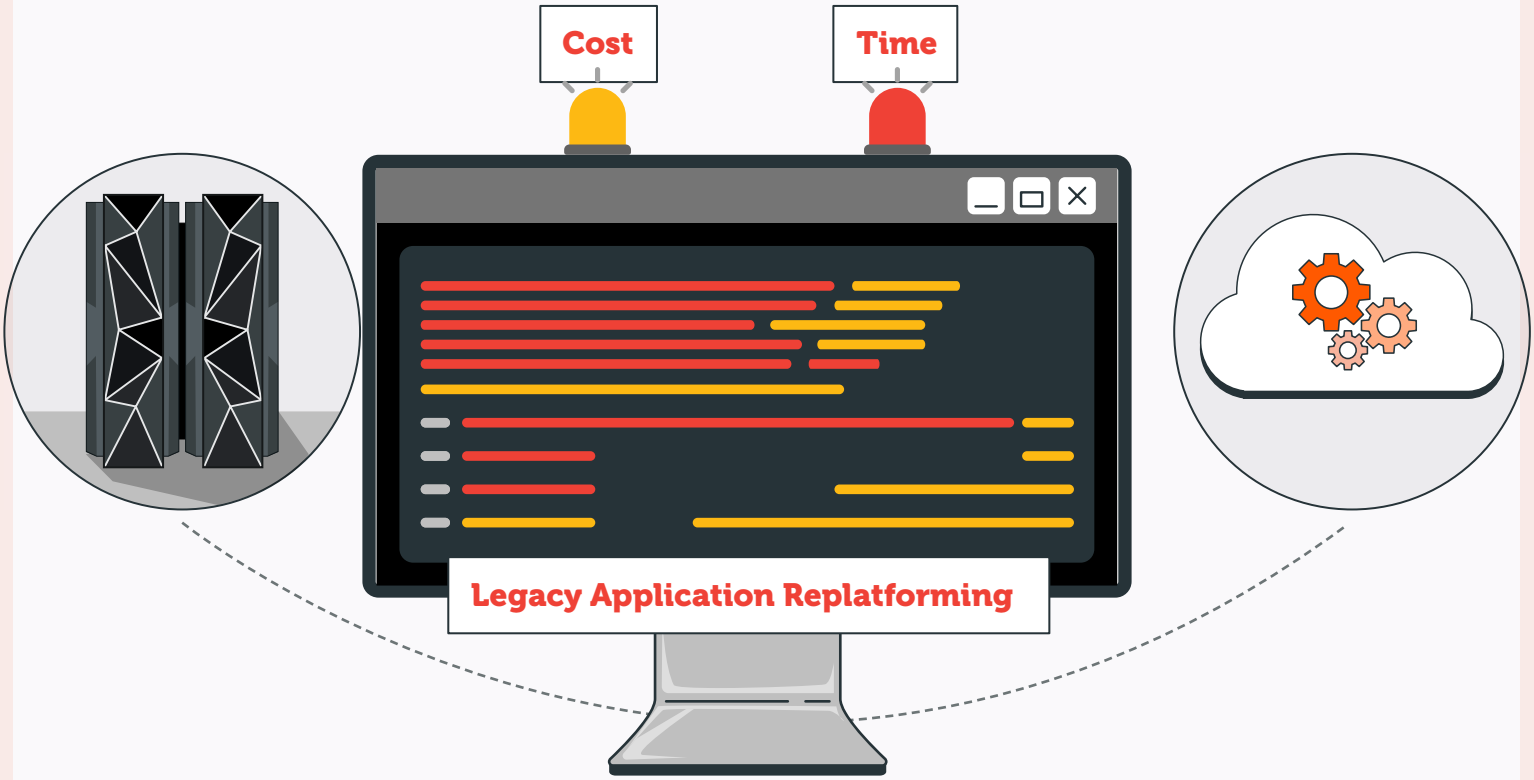
This eBook overviews mainframe optimization, describes trends and challenges in mainframe access, then proposes solutions for enabling integration analytics and remediating suboptimal interactions.



Introduction: Mainframe Optimization

Mainframe optimization is a priority as businesses seek to streamline hybrid application performance and control costs even as usage continues to grow. Hybrid applications and RPA activity are driving higher mainframe transaction volumes and these interactions with the mainframe are often based on inefficient screen-scraping methods.

When hybrid applications and Robotic Process Automation (RPA) mainframe interactions occur at scale, costs increase, performance slows, and reliability suffers. The result is often a push to “optimize” mainframe applications. But how? Many businesses think re-platforming mainframe workload is the only (or best) way to optimize the hybrid environment.



The perception exists that moving workload off the mainframe reduces costs. However, this strategy takes far longer than most businesses anticipate, and simply moves costs from one platform to another without addressing the underlying issues. In many cases, the re-hosting solution is more expensive than leaving the workload on the mainframe and optimizing it in place.

A far simpler solution is optimizing mainframe interactions by:

Identifying the biggest drivers of mainframe transaction volumes.

Optimizing those interactions using better integration technology.

Mainframe optimization tends to focus on cost, but there are often significant performance gains that come from these efforts. Several technical and non-technical approaches exist for making mainframe processing more efficient, minimizing the need for capacity upgrades.

Mainframe Optimization Techniques

A common technique for mainframe optimization costs is segregating expensive workloads through logical partitions (LPARs). If your software is installed on all of your systems, you might be paying an unnecessarily high software bill for capacity that is not directly used by the software. Efficient mainframe businesses will isolate expensive software to only a few LPARs so that you are only paying for the capacity needed to run that specific workload.

The method of optimization that this e-book describes involves making mainframe interactions with external platforms, such as distributed systems and mobile applications, as efficient as possible. Efficiency gains come in two forms: using more efficient integration techniques, and reducing the number of interactions.

It may sound counter intuitive to claim that reducing mainframe interaction is a good optimization technique while moving workload off the mainframe is not.

The difference is in how you optimize interactions, not where you host your critical business applications. This e-book describes a technique to identify inefficient external accesses to mainframe CICS applications and presents a technique to address the inefficiencies.

IBM CICS Transaction Server

Many legacy applications continue to run under the IBM Customer Information Control System (CICS®), a highly scalable and reliable transaction processing platform.

PRESIDENT/CDS at DRAGON SLAYER CONSULTING

According to Marc Staimer, "CICS handles more than 1.1 million transactions per second worldwide. That's more than 95 billion transactions per day."[1]

With so many enterprises running critical business applications in CICS transactions every day, **identifying ways to optimize CICS is a high priority for the IT group**. The CICS transaction server itself has been the target of activities to reduce overhead for many years. The focus is now shifting to make external interactions with CICS transactions more efficient as well.

Why is that? Interactions with CICS from outside of the mainframe are increasing. The drivers of the increase are often hybrid applications and automation. These mainframe interactions are inefficient because they are often implemented by using antiquated interfaces to mimic human end users.

1. **Whitepaper: Why IBM z System Mainframe Cloud Storage Is No Longer An Oxymoron** *Marc Staimer, President & CDS of Dragon Slayer Consulting, 2016*

Trends in Mainframe Access

For many enterprises, mainframe-based **systems of record** are still the heart of critical business logic and data. IT organizations have increased investment in modern **systems of engagement** to interact with stakeholders and provide the front-end presentation access to existing systems of record.

Many systems of record, especially those running in CICS, were originally developed for end user access through 3270 terminals. As modern front-end presentation technology has evolved over the years, most of the interfaces to back-end systems of record have remained untouched.

To allow legacy systems of record to fully participate in modern hybrid systems of engagement requires integrating them. Terminal emulators that scrape data from screens or other application integration tools such as the HostBridge JavaScript Engine (HB.js) enable the necessary integration.

CICS has evolved to support newer interface technologies that enable integration without requiring changes to systems of record. But inefficient, antiquated approaches like screen scraping persist. Opting for sub-optimal integration technologies, particularly when transaction volumes are high, is the root of many failed integration efforts.

When organizations don't have the budget or time to reface existing applications, sub-optimal integrations are seen as the lesser of two evils, especially when integration technologies exist that enable them to remain on the existing interfaces.

Trends in Mainframe Access

Hybrid IT Computing

Many business processes have components that reside in both mainframe and distributed server environments. These cross-platform applications are known as **Hybrid IT** because they include heterogeneous server environments for specific components of the business process.

Hybrid IT applications often include presentation and possibly some business logic residing in distributed environments. They use modern interfaces such as REST or SOAP APIs to support web services calls to external services. They use data formats such as XML or JSON to exchange data.

Mainframe applications provide the data access and integrity component of the composite applications, traditionally referred to as transactional support. Transactional support is critical to business operations integrity. Making changes to the applications that provide this support is considered risky.

Trends in Mainframe Access

Process Automation

Another type of specialized IT processing growing in popularity is **Robotic Process Automation** (RPA). RPA is actually a form of process automation where manual process steps are "observed" and "learned" by software and become repeatable. Automations built with software from third party vendors mimic and replace human interaction within the business process.

In the general sense, process automation allows you to automate steps that normally require manual activity. For instance, you develop automation for terminal oriented requests through a terminal emulator to CICS transactions without requiring any human interaction. Many organizations are automating processes using various scripts and macros in addition to or instead of using RPA software.

Mainframe Access Challenges

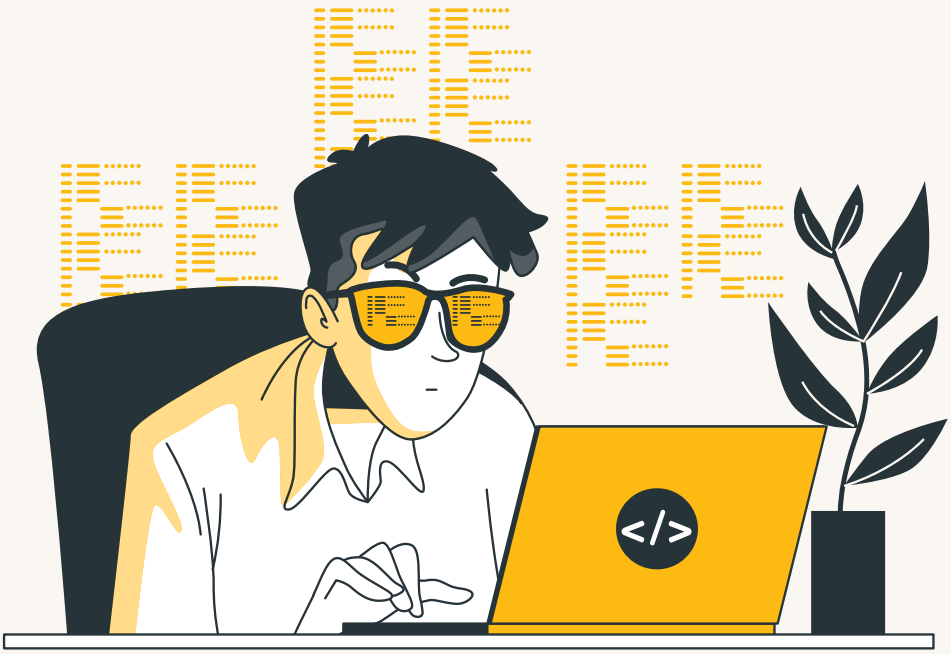
Accessing mainframe transactions and data from hybrid applications or process automation using antiquated approaches such as screen scraping presents significant challenges. Even though they can facilitate mainframe access without changes, the approach can result in unintended negative consequences.

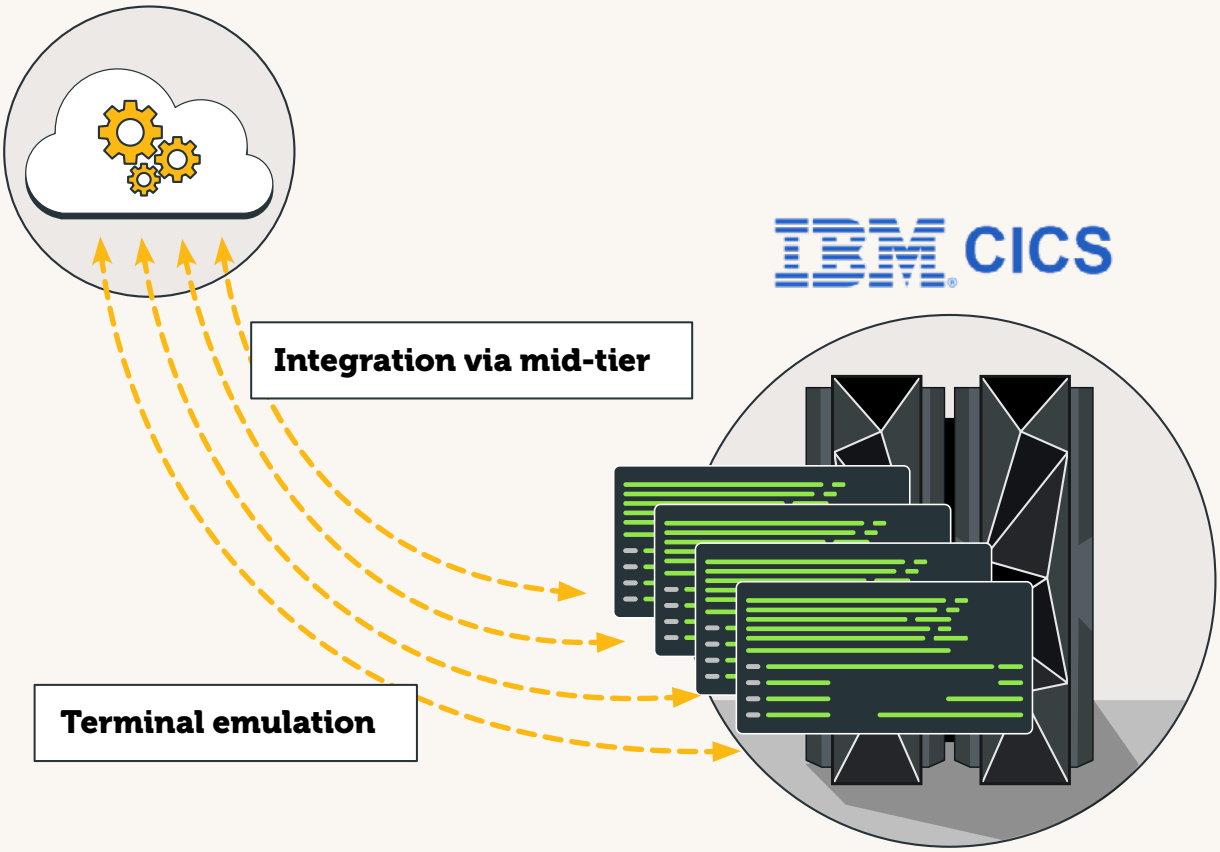
Screen scraping through terminal emulators is widely recognized as an inefficient and brittle integration choice. Requiring that middle tier applications understand the format of every 3270 screens is solving the access challenge in a sub-optimal way.

Middle tier applications become tightly coupled to the screen layouts when they use screen scraping. Both sides of the hybrid application have to change in lock step for it to work. Making any application changes is viewed as risky and avoided.

Hybrid application performance suffers when using sub-optimal integration techniques like screen scraping. Interactions between components of hybrid IT applications cross network boundaries which adds latency and overhead due to protocol packing and unpacking and transferring across the wire. At scale, this latency can have an observable negative effect on performance and CPU usage.

In addition, automations often require many 3270 screen interactions across the network to do a single step in the process which multiplies the latency and CPU usage for that step. Stepping through multiple screens using a terminal emulator is very inefficient.





Those responsible for the middle tier applications and process automations are usually unaware of the negative impact of using terminal emulators built on screen scraping. Often, the increased CPU usage catches the attention of mainframe operations but they are not sure where the requests originate or how to mitigate the problem.

This is where a new offering from HostBridge can help: The HostBridge Transaction Analytics Connector (HTAC).

Closing the Visibility Gap with HTAC

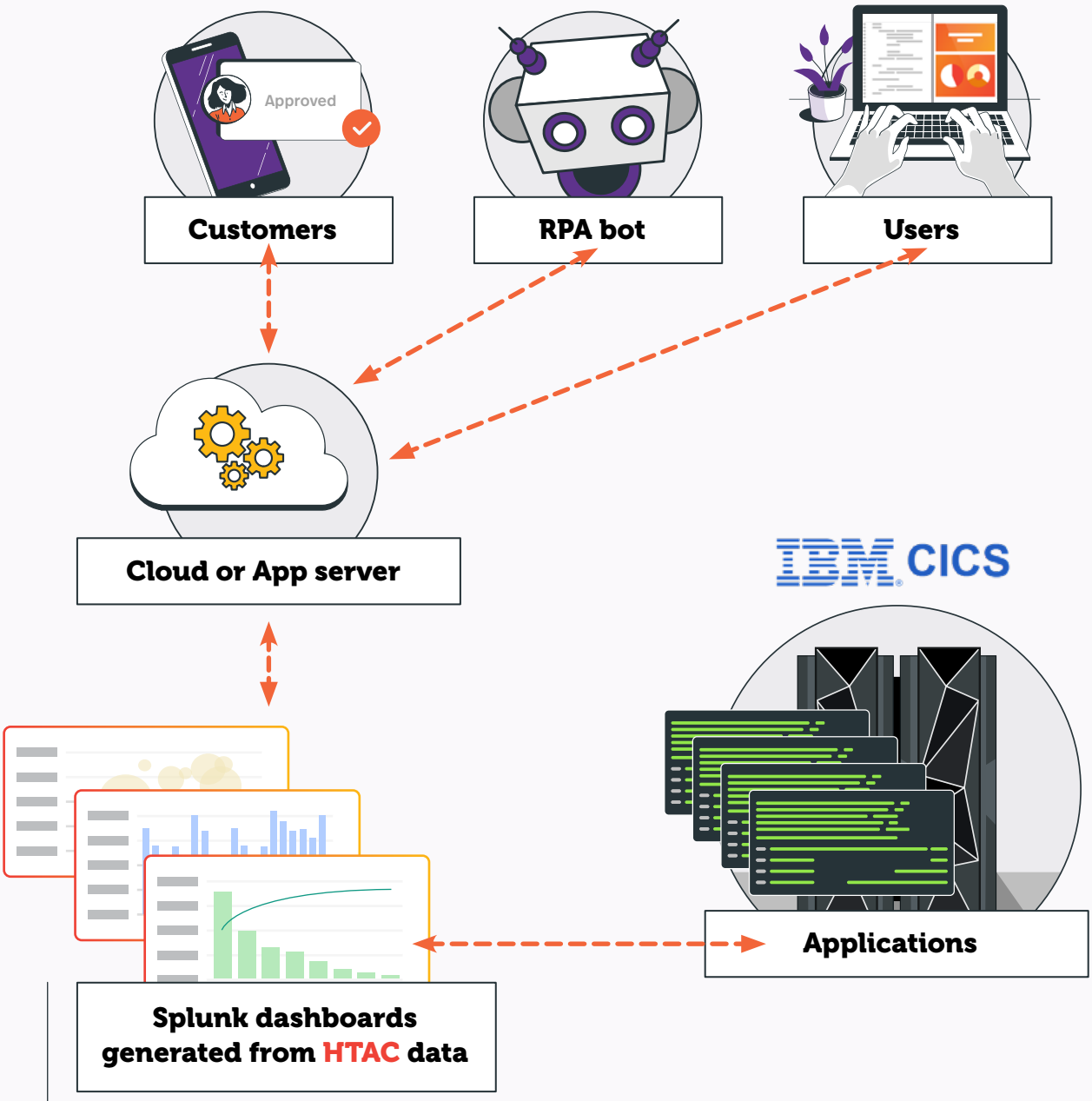
The HostBridge Transaction Analytic Connector (HTAC) is a software product running under CICS that helps businesses identify who is sending requests to CICS from external platforms. HTAC takes information from external requests and includes it in the CICS monitoring data for later analysis.

HTAC enriches the IBM System Management Facility (SMF) record type 110 that is generated by CICS with correlation information that identifies the requester. This data is stored in the transaction tracking Origin Data section of the record and can identify the caller for almost any type of inbound CICS request, including hybrid applications and process automation scripts and macros.

HostBridge currently uses Splunk[2] to perform data analytics on the SMF records that are enriched by HTAC. Splunk is an industry leader in the Security Information and Event Management (SIEM) market. It works with network and machine generated data to provide views and insight into operational security and performance. Prior to HTAC, many businesses that used Splunk viewed the mainframe as a black box in the end-to-end analysis.

Prior to sending the enriched SMF 110 records to Splunk, HTAC scrubs and reformats them into JSON documents that are a common form of Splunk input. With HTAC, HostBridge provides a set of Splunk dashboards to analyze the enriched SMF 110 data and present it in a consumable format. With these dashboards, users get true end-to-end visibility into how hybrid application are driving CICS transactions.

2. Plans to leverage the ELK Stack as an alternative to Splunk are currently under consideration.



These custom dashboards reveal precisely which interactions occur most frequently and consume the greatest amount of mainframe resources. They provide the insights necessary to focus optimization efforts in the places that provide the greatest benefit.

Optimizing Mainframe Interactions with the HostBridge JavaScript Engine

Closing the visibility gap in hybrid applications and process automations can point out specific cases where inefficient access patterns are being used. Often, these access patterns can be optimized without the risk of touching the mainframe CICS application by simply changing the integration approach.

Instead of using terminal emulation to connect to CICS applications, you can use an integration tool called the HostBridge JavaScript Engine (HB.js). As the name implies, this product runs a JavaScript engine inside of CICS and provides a landing place for requests using several modern interface styles.

HB.js optimizes mainframe CICS interactions in two ways: replacing 3270 terminal emulation and providing transaction orchestration. Both of these techniques can reduce latency and CPU usage associated with external CICS calls.

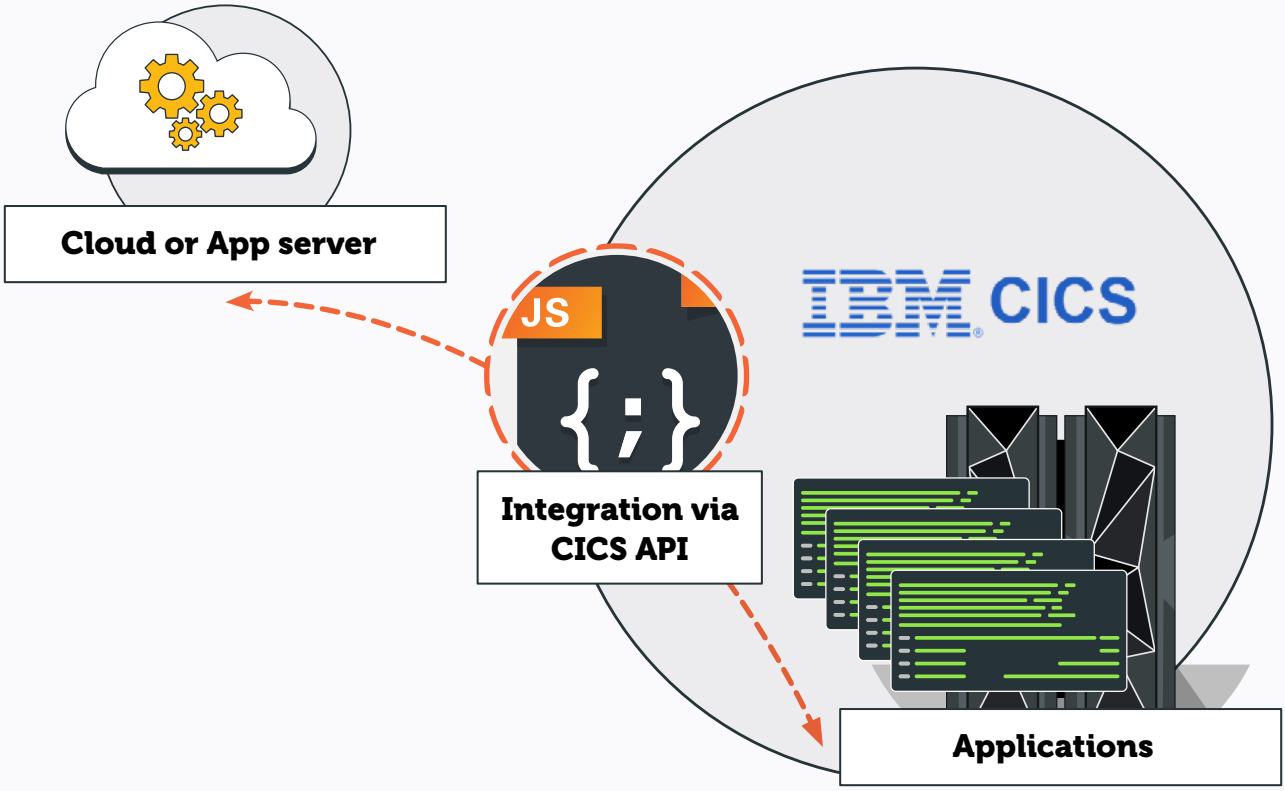
HB.js uses the CICS 3270 bridge to access the screen data without referencing the actual screen layout. By using the Application Data Structure (ADS) that defines the field names for each field on the screen, there is no need for row / column layout knowledge. This loose coupling allows the applications to change independently from each other.

HB.js uses scripts to control the processing of requests that arrive. These scripts are created with Eclipse-based tooling that is augmented with HostBridge support so that the JavaScript engine can control the orchestration of CICS transactions. A single HB.js request can start any number of CICS transactions, can link to any number of programs, and assess any CICS data source -- and return a single result set.

By orchestrating complex requests from a single API call, HB.js reduces the latency of using multiple network calls. The orchestration is done locally on the mainframe so **no additional network calls are needed**. CPU usage is reduced because there is far less network stack overhead. And, since HB.js is zIIP-enabled, there is no downside to performing the orchestration on the mainframe.

Here is an example. Suppose you need some mainframe data in your hybrid application. In order to get the data, you need to navigate through several CICS screens that contains pieces of the data. When you use terminal emulation to get the data, each CICS screen interaction results in a network call to the mainframe. You might need dozens of calls to get the needed data.

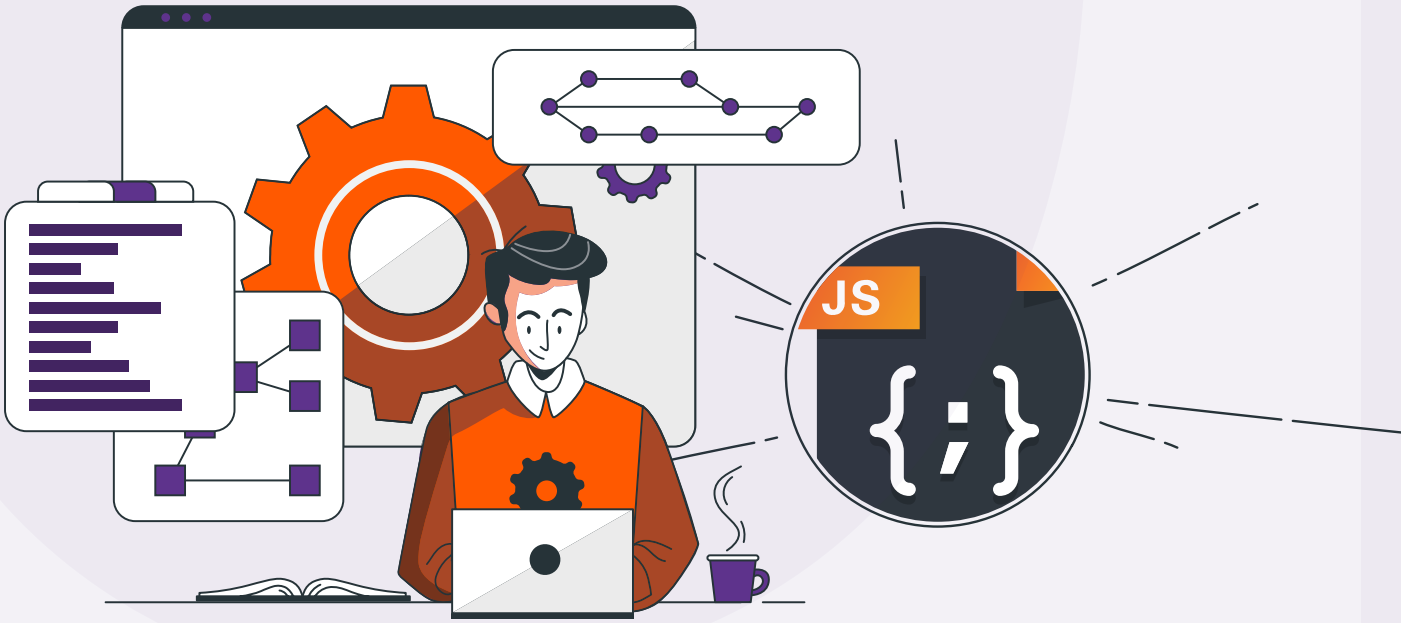
By using HB.js, the distributed app sends a single request to the mainframe using a modern, HTTP-based API. An HB.js script will “fulfill” the intent of the API by decomposing the request into individual transaction calls. The HB.js script will aggregate the transaction output and return the single result to the distributed application. It’s as simple as that. The end result is eliminating terminal emulation and screen scraping from the integration picture.



Conclusion

Mainframe optimization can take make forms, but understanding where you are using suboptimal integration techniques and then replacing them with modern interfaces that reduce latency and resource usage is the goal of the HostBridge HTAC and HB.js tools.

No matter where you are in your mainframe optimization journey, CICS integration analytics can help you get the greatest return on your efforts. **Contact the HostBridge team to set up a free mainframe optimization analysis using your CICS SMF data: https://www.hostbridge.com/mainframe-analysis.**



About HostBridge Technology, LLC

HostBridge Technology develops and delivers enterprise software to enable integration of CICS applications and data with mobile, web and any other mid-tier applications and technologies. With HostBridge, integrations are easy to develop, fast to deploy, durable and high-performing because they do not rely on screen-scraping technology. For more information about HostBridge, please visit **https://www.hostbridge.com**.

About EmpriZe IT Consulting, LLC

EmpriZe IT Consulting, LLC offers several different consulting and freelance services predominately for IBM mainframe clients. S. Michael (Mike) Benson, the founder and principal architect, has over 40 years of IT experience. Mike left IBM after 30 years of service as an Executive IT Architect in technical sales where he worked with numerous clients creating IT solution architectures. You can find out more by visiting **https://www.emprizeitconsulting.com**.

HOSTBRIDGE®
TECHNOLOGY

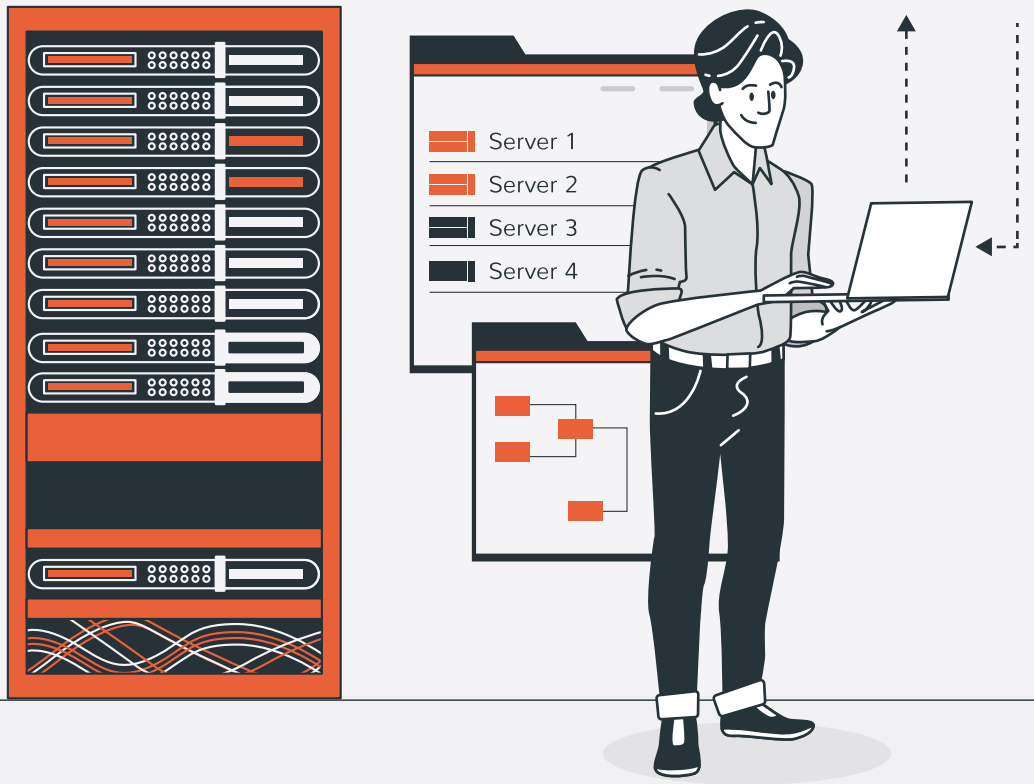
Toll Free **866.965.2427**
International **+1.405.533.2900**
info@hostbridge.com

Solutions

Company

Blog

Schedule a **free mainframe analysis**!



Illustrations **Freepik Storyset**