EXECUTIVE BRIEF | SEPTEMBER 2014

ca technologies

Service Virtualization in Mainframe Environments:

Removing Constraints to Speed Development with Higher Quality and Lower Cost

Executive summary

Mainframe systems form the IT backbone for many organizations, providing back-end transaction processing, databases and reporting functionality. Although they are often thought of as legacy systems, this misleading characterization discounts the fact that these systems often hold critical business logic and still receive heavy investment.

Organizations that develop and test applications in mainframe environments have struggled to keep up with users' increasing demands for innovative, new functionality. For one, mainframe development can require decades' worth of legacy expertise, which is in danger of disappearing as a generation of workers approach retirement. In addition, modern, client-facing applications driven by the mainframe are being accessed by growing numbers of users via smartphones, tablets and other devices – stressing mainframes in unforeseen ways. Finally, there is the significant cost of sourcing the additional MIPS needed to support new development and test requirements within the mainframe – not to mention the time and resources required to install these complex systems.

Because of these factors, new challenges have arisen within mainframe development and test operations, such as unavailable systems, inability to accurately model performance and complex test data requirements. However, by employing service virtualization to model the core business logic in mainframe platforms and their integrations with other in-house and third-party systems, teams can free themselves of these constraints, leading to faster build and test cycles, higher quality applications and lower costs.

By employing service virtualization to model the core business logic in mainframe platforms and their integrations with other in-house and third-party systems, teams can free themselves of constraints, leading to faster build and test cycles, higher quality applications and lower costs.

Complex Systems and Constrained Development Stifle Innovation

The proliferation of web-based technology (e.g., HTTP, SOAP, REST, etc.) and messaging technology (e.g., Websphere MQ, etc.) has opened the wealth of information residing in the mainframe to distributed applications. These distributed applications rely on the availability of mainframe test and QA systems for development, testing, debugging, quality assurance and performance testing. As reliable as mainframes can be, however, their applications and associated systems may be unavailable at various times due to system maintenance, resource constraints and mainframe development activities that temporarily cause a transaction to fail. It may also be difficult for distributed systems developers to return mainframe databases to a steady state, resulting in additional challenges for systems development and QA staff.

These distributed system developers are under pressure to deliver innovative new products and services to market at a higher quality and lower cost than their competitors. Yet despite how modern these new offerings and products can be, they often rely on business logic residing in legacy mainframe environments. What's more, development teams have adopted Agile methodologies that do not always match up well with the constraints imposed by mainframe systems – which have become especially acute as organizations embraced composite application deployment and development techniques, including service-oriented architecture (SOA), business process management (BPM), cloud and software-as-a-service (SaaS) applications. As a result, application environment costs and risks have grown due to increasing complexity and frequent unavailability of needed systems.

With the "consumerization of IT" in full effect and the demands of today's technology users ever increasing, however, these constraints in the software deployment life cycle (SDLC) can no longer be an excuse. IT organizations need to find ways to overcome them, so they can improve communication, integration and collaboration between development and operations (a.k.a., DevOps) and accelerate the delivery of high-quality, new services to the marketplace. How? By removing these constraints with service virtualization.

With the "consumerization of IT" in full effect and the demands of today's technology users ever increasing, constraints in the software deployment life cycle can no longer be an excuse for service-oriented products that are delivered late, over budget and with questionable quality.

Service Virtualization in Mainframe Environments

Most new enterprise applications are built in modern, distributed environments and supported by multiple service teams and delivery partners. Existing mainframe systems provide dependent system data for these heterogeneous applications – an approach that leads to multiple constraints within application development, including unavailable dependencies, poor performance, conflicting delivery schedules and test data constraints. On top of these constraints, it is often too cost-prohibitive for most organizations to purchase the additional MIPS it would take to resolve these conflicts within their mainframe environments.

By capturing and simulating the behavior, data and performance characteristics of dependent systems and deploying a virtual service that represents those systems, service virtualization removes the development constraints of unavailable systems, conflicting delivery schedules and data volatility. This allows software to be developed faster, with lower costs and higher reliability.

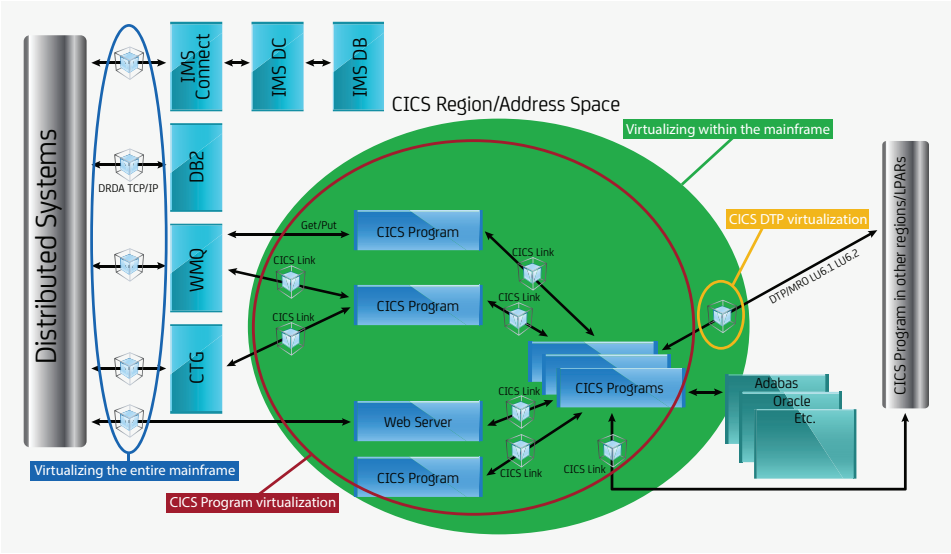
A good analogy for how service virtualization works is the development of an airplane. Engineers don't wait until a plane is fully assembled to test it. They model each individual component in a computer simulation and test its viability in a controlled environment, so by the time the first physicalprototype is assembled, its component parts have been validated to operate as expected. Swap the plane parts for stages in the SDLC, and it becomes easy to see how service virtualization can drastically improve the software development process.

Additional benefits of service virtualization include:

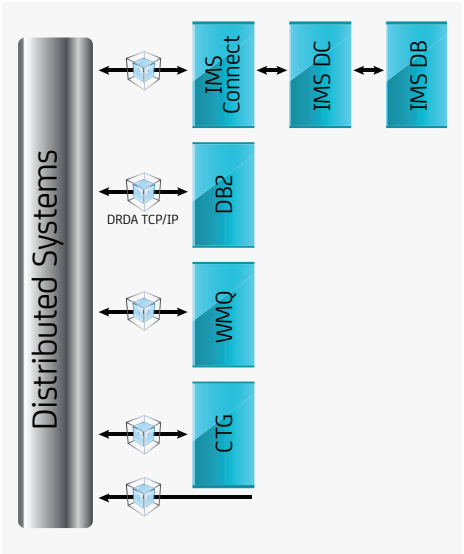
- **"Shift left."** Move software development into parallel and test and validate sooner in the software lifecycle, where it is less expensive and issues are easier, less disruptive and less expensive to resolve.
- **Infrastructure requirement reduction.** Eliminate much of the concurrent demand for environments and hardware Agile methodologies create.
- **Performance readiness.** Solve the challenging problems of properly evaluating the scalability of applications by load testing at the component level instead of waiting until the application is complete – in conjunction with incorporating production network conditions in the test lab.
- **Data and scenario management.** Reduce or eliminate the need for complex test data management, system setup and other complexities by virtualizing the system behavior to account for edge conditions, negative test scenarios and error conditions.

Organizations can move development and test tasks earlier in the software lifecycle, resulting in reduced time to market, lower infrastructure costs, reduced contention for labs and better overall application quality.

Types of Virtualization in Mainframe Environments,



Virtualizing the Entire Mainframe



Description

- Recording desired mainframe responses to distributed system requests when the mainframe and all back-end systems are available.
- Virtual model is utilized by the distributed system under development or test, and these activities may proceed independent of the mainframe.
- Referred to as “on the wire” virtualization. The requests and responses are recorded as they flow out of and into the distributed system. During virtualization, the flow to the mainframe is replaced with a flow to the virtual services environment, so no actual connection to the mainframe is required.

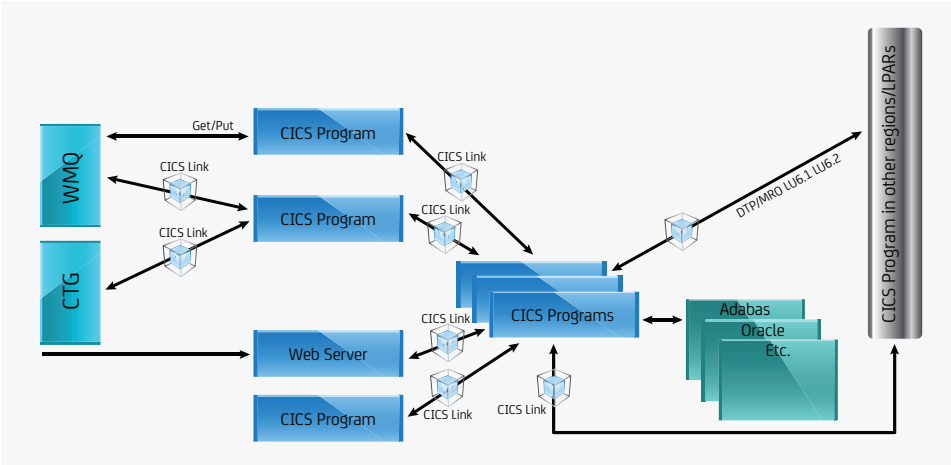
Benefits

- Uninterrupted, stable development and testing of the distributed application
- Parallel development of distributed applications and mainframe
- Reduced mainframe costs by eliminating the requirement for additional MIPS, disk space, software licenses, etc.

CA Support

- HTTP
- Websphere MQ
- IMS Connect Gateway
- CICS Transaction Gateway
- Distributed Relational Database Access (DRDA) to DB2
- APPC/CPIC Clients using IBM Enterprise Extender

Virtualizing Within the Mainframe



Description

- Targets specific mainframe components.
- "Agent based" virtualization with requests and responses recorded between mainframe components.
- During virtualization, the agent inserts itself between the mainframe components providing the "virtual" component.

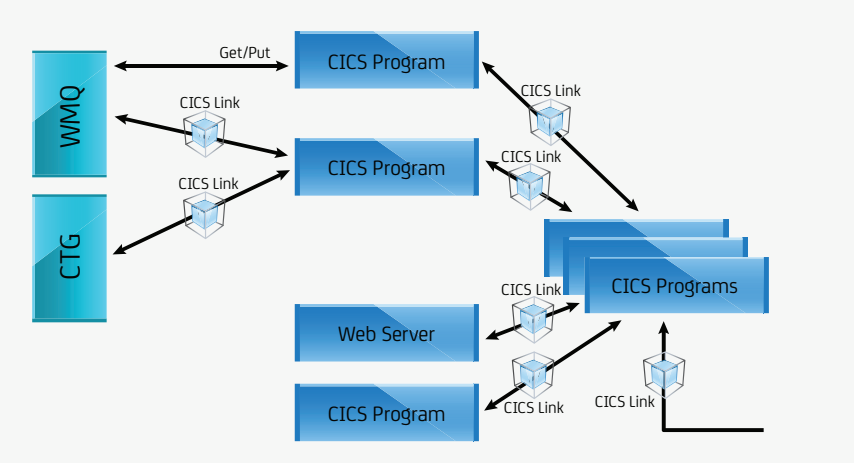
Benefits

- Elimination of constrained components that interrupt development, debugging, testing and QA of distributed system applications and mainframe applications
- Testing and QA of a path that includes the majority of the mainframe infrastructure while eliminating constraints
- More stable development, test and QA
- Parallel development of a mainframe application and a dependent mainframe application
- Significant mainframe cost reduction by eliminating particularly expensive mainframe resources

CA Support

- z/OS CICS program virtualization
- z/OS CICS Distributed Transaction Processing (DTP) virtualization

CICS Program Virtualization



Description

- Utilizes the primary mechanism within CICS for invoking a program in same region, the CICS LINK command, and in different regions, the Distributed Program Link (DPL).
- Reasons that a CICS program might be virtualized:
  - Does not yet exist or is being revised and is effectively unavailable
  - Performs thousands of database calls
  - Requires a database that is often offline
  - Invokes an external system that is often offline or costly on a per-transaction basis
  - Performs database updates, which require a database restore between testing cycles

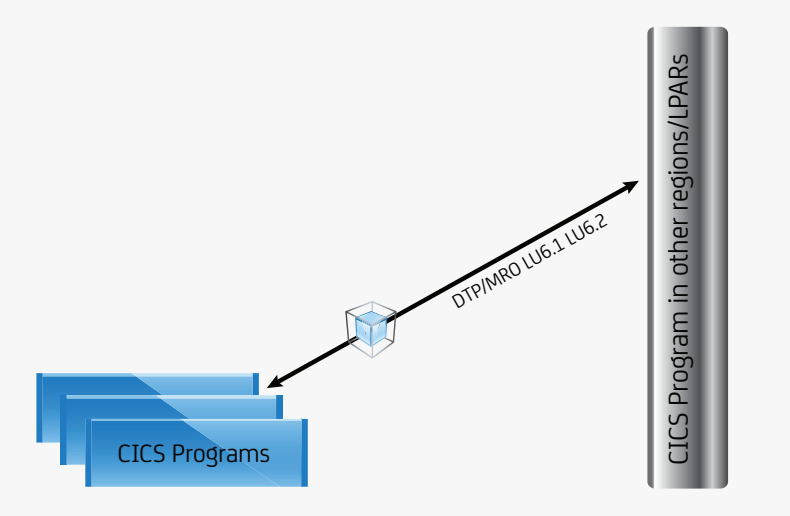
Benefits

- CICS program is virtualized with no modifications to the program issuing the LINK, the target program or the CICS configuration. When a CICS program is virtualized, all users of the program receive the virtual service, so the deployment of a virtual service to virtualize a CICS program is transparent.

CA Support

- Local LINK
- Distributed Program LINK

CICS Distributed Transaction Processing (DTP) Virtualization



Description

A CICS program establishes communication with another CICS program (or IMS program, or any other peer program that supports LU 6.1 or LU 6.2) using CICS DTP commands. Once the connection is established, the programs exchange data. A typical DTP client program connects to the server program, sends a request and receives a response.

Examples of DTP clients that could be virtualized are:

- A CICS program that uses DTP LU 6.2 commands to invoke a service on a CICS region at a remote site that is unavailable or has high per-transaction costs.
  - A CICS program that uses DTP LU 6.1 commands to invoke a service on a remote IMS region that is unavailable in the middle of the day due to time-zone differences.
  - A CICS program that uses DTP Multi-Region Option (MRO) to invoke a service on a CICS region within the z/OS Sysplex that is unavailable while a long batch job runs.

Benefits

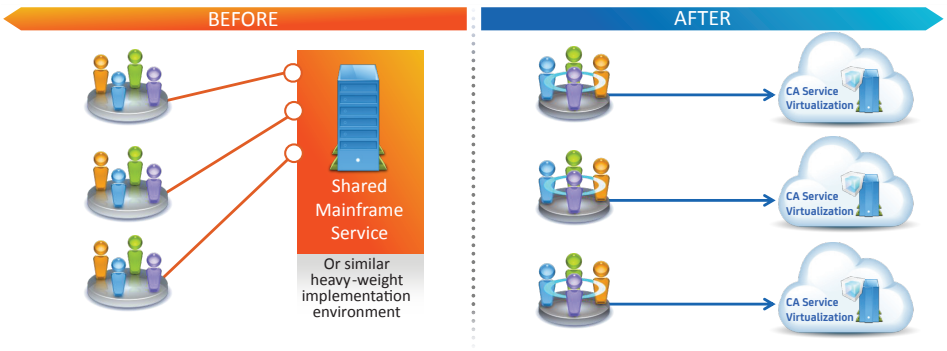
- CICS DTP connection is virtualized with no modifications to the program issuing the DTP command, the target program or the CICS configuration. When a DTP program is virtualized, all users of the program receive the virtual service, so the deployment of a virtual service to virtualize a DTP connection is transparent.

CA Support

- Virtualization of a DTP connection/server by virtualizing the DTP client commands

CA Service Virtualization for Mainframe Environment Development from CA Technologies2

CA Service Virtualization was built from the ground up to virtualize and validate complex, multi-tier applications. As illustrated below, CA Service Virtualization is highly suited to helping IT teams overcome the development challenges they often face in mainframe environments:



Before

- Development teams are dependent on shared mainframe services that are subject to system unavailability and conflicting development schedules.

After

- CA Service Virtualization simulates production-only systems like mainframes that are not normally available for load testing, creating an always-ready, highly scalable virtual back-end that is immune to traditional load-testing constraints and concerns.
- Development teams become more productive as they integrate with the virtual service, and the costs to replicate expensive back-end systems are eliminated.

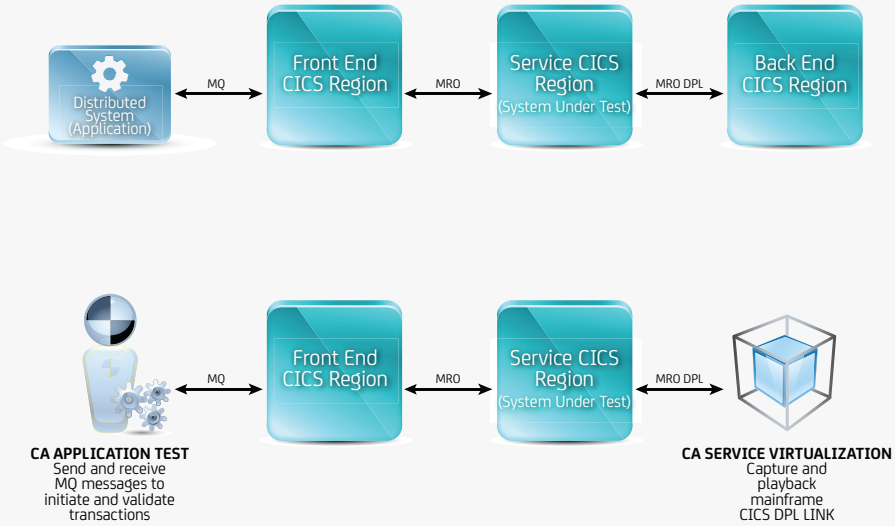
CA Service Virtualization was built from the ground up to virtualize and validate complex, multi-tier applications. As illustrated below, CA Service Virtualization is highly suited to helping IT teams overcome the development challenges they often face in mainframe environments:

- Added support for CICS commands, including CICS START, CICS RETRIEVE and CICS XCTL
- A new ABEND message type for identifying abnormal message termination of a CICS transaction
- Ability to use CICS LINK and DTP MRO data as input for building a virtual service
- Support for Function Management Headers (FMH) within DTP MRO and LU 6.1
- New features that support CICS Transaction Gateway (CTG) and IMS Connect commands, enabling virtualization of transactions through mainframe CICS

Service Virtualization for Mainframe Success Stories

CA Service Virtualization has been implemented at over 200 enterprises, including some of the most challenging mainframe environments.

Case study: *Large bank faces difficulties with testing due to dependent, unavailable CICS components.*



Challenge

- Multiple interdependent CICS components are required to provide services
- Dependencies drive overall system unavailability if even one component is down
- Testing schedule is extended and difficult to meet

Solution

- Implemented to facilitate testing of Personal Banking application and eliminate constraints from dependent systems
- Developed virtual services to support testing without the dependency on backend systems

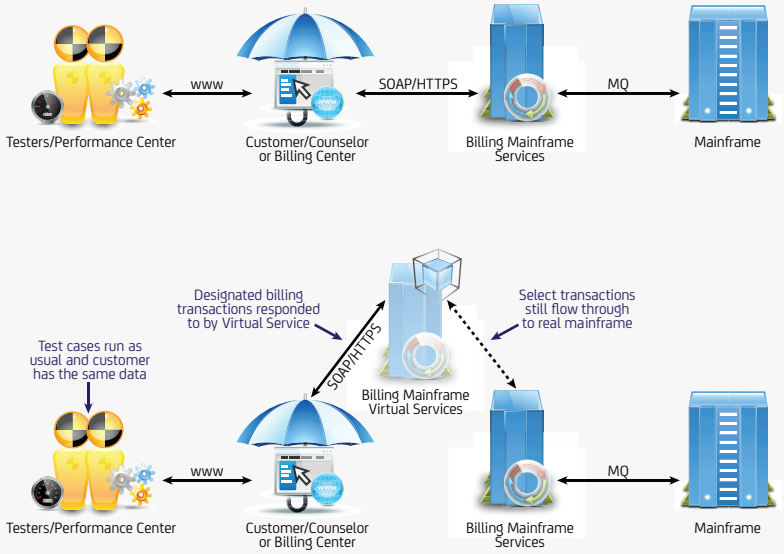
Benefits

- Implemented to facilitate testing of Personal Banking application and eliminate constraints from dependent systems
- Developed virtual services to support testing without the dependency on backend systems

Service Virtualization for Mainframe Success Stories

CA Service Virtualization has been implemented at over 200 enterprises, including some of the most challenging mainframe environments.

Case study: Large insurer faces high data setup costs and persistent system unavailability.



Challenge

- Multiple interdependent CICS components are required to provide services
- Dependencies drive overall system unavailability if even one component is down
- Testing schedule is extended and difficult to meet

Solution

- Implemented to facilitate testing of Personal Banking application and eliminate constraints from dependent systems
- Developed virtual services to support testing without the dependency on backend systems

Benefits

- Implemented to facilitate testing of Personal Banking application and eliminate constraints from dependent systems
- Developed virtual services to support testing without the dependency on backend systems

Learn more about CA Service Virtualization.

Many CA customers report seeing 25% to 50% reductions in cycle times, dramatically reduced lab infrastructure costs and better applications where 60% to 90% more defects are identified at least one cycle earlier in the software deployment life cycle.

Connect with CA Technologies at ca.com

CA Technologies (NASDAQ: CA) creates software that fuels transformation for companies and enables them to seize the opportunities of the application economy. Software is at the heart of every business, in every industry. From planning to development to management and security, CA is working with companies worldwide to change the way we live, transact and communicate – across mobile, private and public cloud, distributed and mainframe environments. Learn more at **ca.com**.

CS200-87070_0914