

IBM Mainframe Makeover: What Powers your Mobile Applications and Smart Phones?

Pramod Chandran

M.E Student

Kathir College of Engineering, Coimbatore

Abstract

The principal objective of this paper is to demonstrate the capabilities of legacy mainframes to support and use in smart phones and mobile applications devices. Legacy mainframes are trusted backbone platforms for executing critical business transactions. They are robust, reliable, scalable, and most importantly, resilient. Mainframe ecosystems have evolved to stay relevant, and despite their bad press, they have contributed to major technology milestones in the last decade. 96 of the world's top 100 banks, 23 of the top 25 US retailers, and 9 out of 10 of the world's largest insurance company's still use mainframes. Mainframes process roughly 30 billion business transactions per day, including most major credit card transactions and stock trades, money transfers, manufacturing processes, and ERP systems. Across the globe, more people now use mobile devices as their primary means of obtaining information and requesting services over the Internet. This shift is due to the convenience of being able to carry devices wherever they go, along with their user-friendly and intuitive nature. There have been many discussions on how to integrate the different capabilities and features of mainframes with the features that are required on a mobile application. The introduction of web services capabilities in CICS Transaction Server V3 provided the fundamental building blocks of service connectivity, enabling the adoption of service-oriented architecture (SOA), and underpinning today's mobile solutions. CICS has continued to add new capabilities to the run time. From a mobile perspective, the introduction of the CICS TS Feature Pack for Mobile Extensions provides JSON and Representational State Transfer (REST)-conforming (RESTful) web service support, further enhancing the options for enterprise applications to mobile devices. Customers around the world use CICS TS to host hundreds of millions, and in some cases billions, of transactions per day. As the number of mobile devices worldwide continues to grow, so does the variety and volume of workload that they drive. CICS has the capacity to scale up in support of this increasing mobile workload, providing an exceptional platform for hosting mobile workloads.

Keywords: CICSJSON and RESTIBM Worklight

I. INTRODUCTION

In late 2007, researchers reported that worldwide mobile device penetration had reached 3.3 billion, and that number has been steadily climbing since. That amounts to billions of handheld devices capable of communications and, increasingly, sophisticated computing.

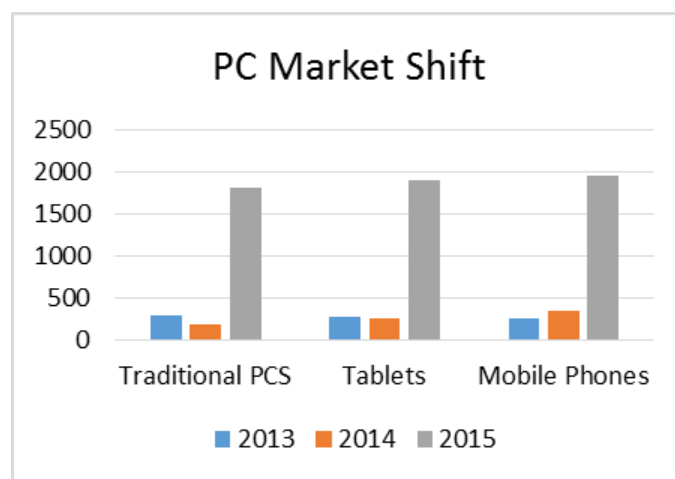


Fig. 1: PC market shift

Smartphones and tablets are in no way different from PCs, in that they're access devices calling on backend systems that run on mainframes. But because they're so handy and accessible, these new devices are generating a tsunami of requests for information. People now check account balances 12 times a month where they might have done so twice a month on a PC.

The CICS (Customer Information Control System) application server, which runs on the IBM mainframe, processes 1.1m transactions per second, significantly more than the number of Google searches. These features can be used to run mobile applications on mainframes where the amount of data that has to be processed is enormous.

IBM has designed the latest version of the mainframe, with the smartphone in mind. The new mainframe, the z13, has been engineered to cope with the huge volume of data and transactions generated by people using smartphones and tablets.

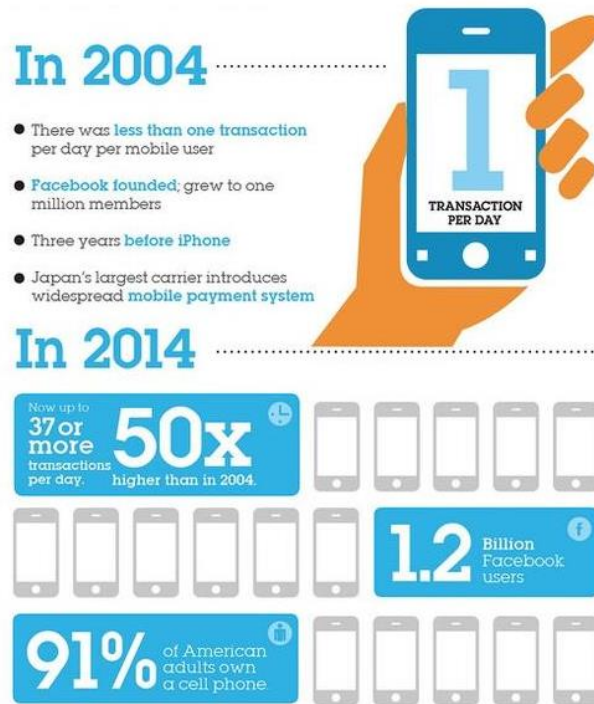


Fig. 2:

II. OVERVIEW

For many years CICS Transaction Server has been capable of hosting mobile enterprise services. The introduction of web services capabilities in CICS Transaction Server V3 provided the fundamental building blocks of service connectivity, enabling the adoption of service-oriented architecture (SOA), and underpinning today's mobile solutions.

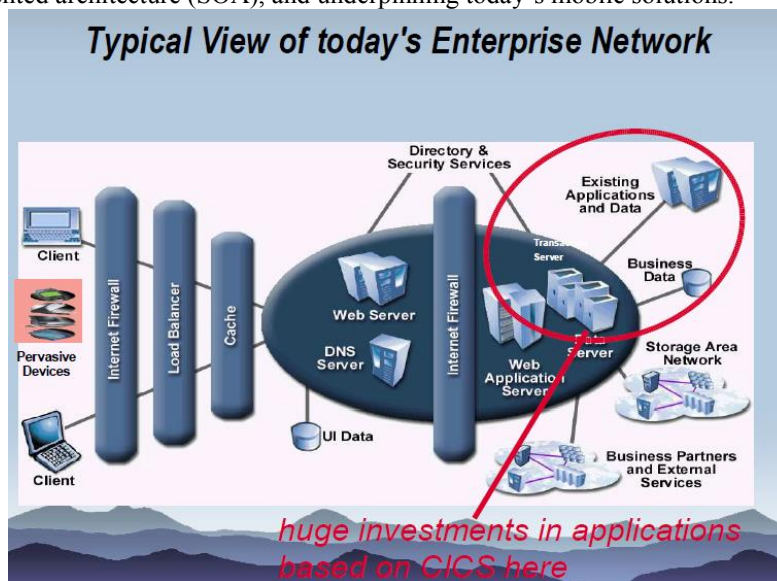


Fig. 3:

CICS has continued to add new capabilities to the run time. From a mobile perspective, the introduction of the CICS TS Feature Pack for Mobile Extensions provides JSON and Representational State Transfer (REST)-conforming (RESTful) web service support, further enhancing the options for enterprise applications to mobile devices. Customers around the world use

CICS TS to host hundreds of millions, and in some cases billions, of transactions per day. As the number of mobile devices worldwide continues to grow, so does the variety and volume of workload that they drive. CICS has the capacity to scale up in support of this increasing mobile workload, providing an exceptional platform for hosting mobile workloads.

As a platform, the primary benefits offered by CICS in support of mobile devices are noted in the following list:

- Provide reuse of existing enterprise services.
- Using the established web service technology within CICS, it is relatively simple to build a set of enterprise services that can be used by a mobile device.
- Provide simplified consumption of enterprise data using JSON-formatted data. A common misconception is that enterprise data in CICS can be hard to use. The CICSTS Feature Pack for Mobile Extensions provides support for JSON data, which is rapidly becoming the standard format for data interchange on mobile devices.
- CICS already operates at the heart of the enterprise. Hosting mobile applications within CICS brings them closer to the enterprise data that they are accessing, minimizing application path lengths and keeping response times down.
- Adopt a RESTful architectural style for service delivery. A RESTful architectural style is one where the target resource, and the operation to be performed against it, are defined by a combination of a well-structured Uniform Resource Identifier (URI) and one of the four Hypertext Transfer Protocol (HTTP) methods (GET, POST, PUT, and DELETE).
- Provide capacity to manage mobile workload. It was earlier that customers around the world use CICS TS to host hundreds of millions, and in some cases billions, of transactions per day. CICS Workload Management provides a robust and scalable platform suitable for supporting the heaviest of mobile workloads.

III. SOLUTION OVERVIEW

The different approaches that I describe in detail are summarized in the following list:

A. The top-down approach

Figure 1-1 shows the suggested method of building new enterprise services for a mobile application in CICS. This approach lends itself to the RESTful architectural style. This approach enables you to create a set of services with a concise interface.

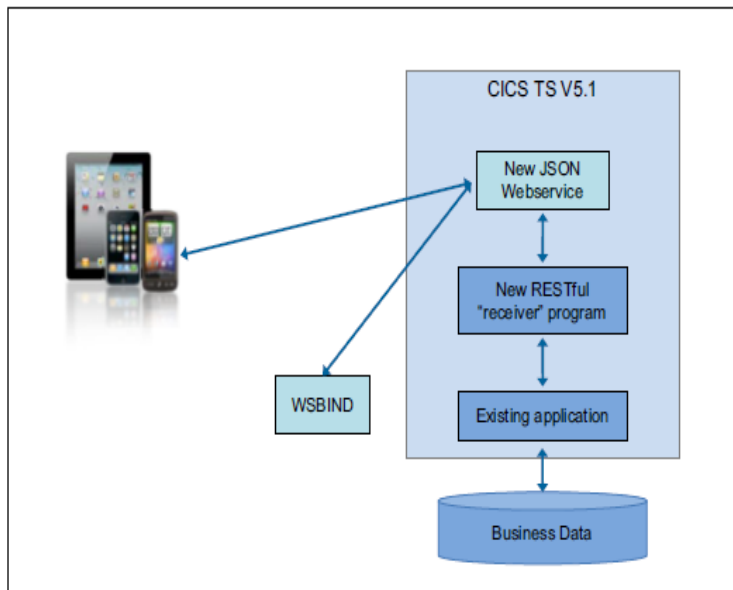


Fig. 4: A possible way to implement a JSON web service from the JSON schema

B. The bottom-up approach

Figure 1-2 on page 6 shows perhaps the fastest approach for delivering enterprise services to mobile devices. Building on an existing SOA, the bottom-up approach enables you to define a JSON or SOAP interface to an existing Common Business Oriented Language (COBOL), C/C++, PL/I, or Java application. This approach maximizes the reuse of existing assets, and minimizes the creation of new components.

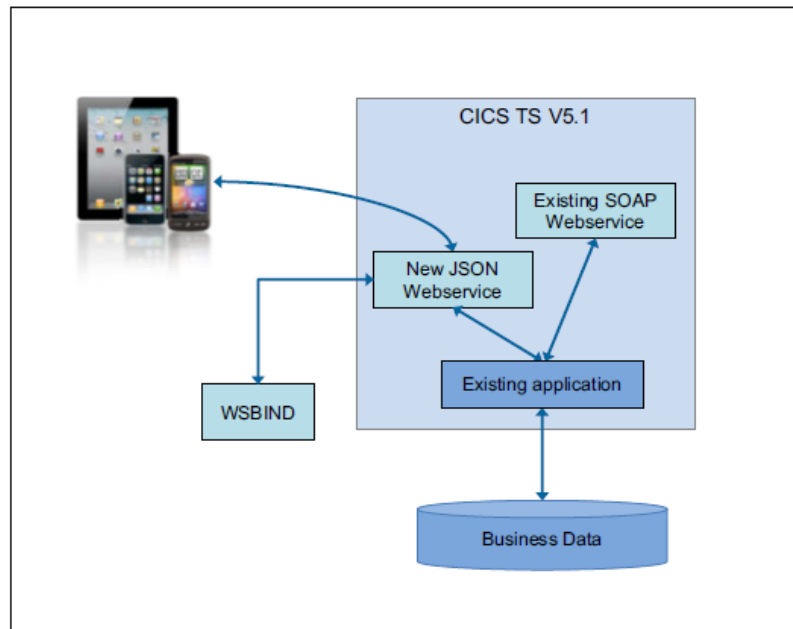


Fig. 5: architecture for a JSON web service to be driven by high-level language data structure

C. The requester mode approach

Figure 1-3 shows how this approach enables CICS to participate in JSON-based interactions, and to make requests against external service providers that offer a JSON-based interface. Though not strictly a pure mobile scenario, the capabilities offered by the CICS TS Feature Pack for Mobile Extensions provide CICS with additional options for connecting to the wider enterprise using JSON-formatted data.

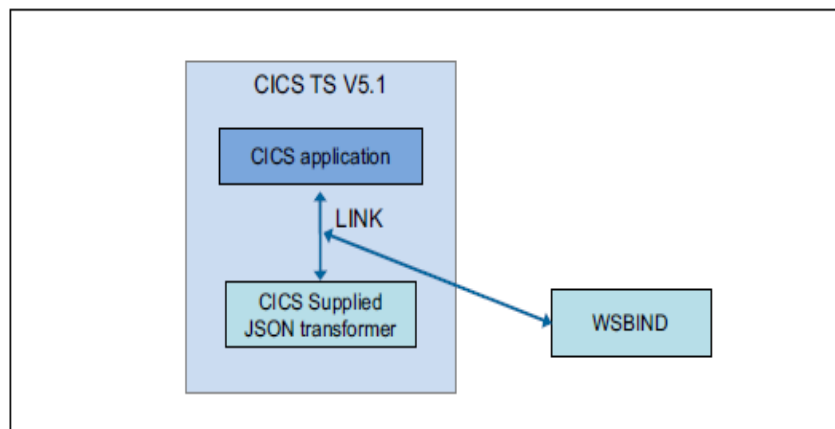


Fig. 6: Linkable CICS program transforms data between high-level data structures JSON

Java API for RESTful Web Services (JAX-RS) is a programming interface that provides support in creating web services according to the REST architectural pattern, as shown in Figure 1-4.

REST is an architecture style for designing networked applications without the need for complex mechanisms, such as Common Object Request Broker architecture (CORBA) or SOAP. The pattern involves client/server communications where the state of an application is held by the client, which reduces processing required on the server. Using the IBM WebSphere Liberty profile provided in CICS TS V5.1, you are able to write your business applications using JAX-RS.

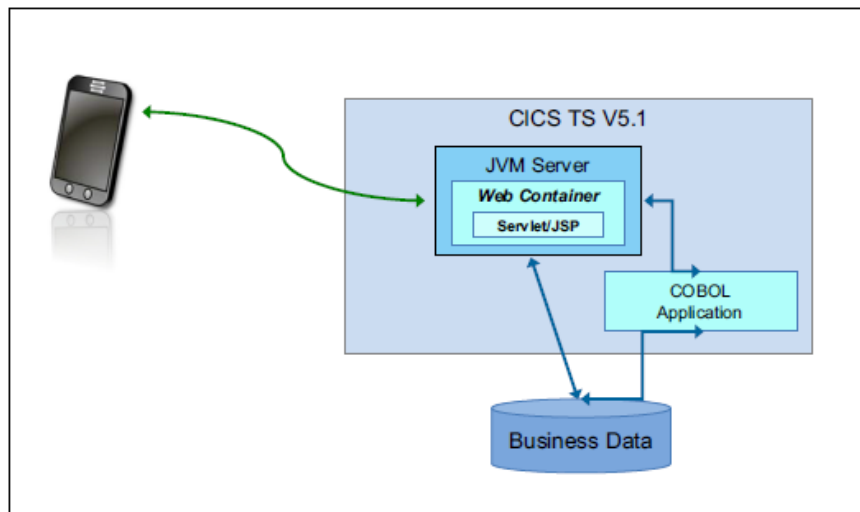


Fig. 7: JAX-RS provides a rapid and easy development of RESTful-enabled java applications

IV. SOLUTION ARCHITECTURE

The architecture for an enterprise mobile solution based on CICS will vary, depending on business requirements and the business data that the applications require. This paper describes the following solution architectures:

- Direct to CICS
- A two-tier Worklight and CICS solution

The direct-to-CICS solution is one where the mobile devices communicate directly to CICS. In this architecture, other devices communicate with CICS through the existing web services provided by CICS. This scenario is ideal when the devices and networks involved are all trusted, the applications involved do not require frequent updates, and the applications do not run on multiple platforms.

An alternative solution is architecture with one or more layers between CICS and the user devices.

D. IBM Worklight

Worklight is a solution that provides governance and security for your mobile applications, along with a powerful software development kit (SDK) for rapid development of your enterprise applications on most major platforms. In this architecture, the mobile devices communicate with the Worklight Server, which ensures that the device has access rights to make requests to CICS. If approved, this request is then sent to CICS, and the CICS application is run.

IBM Worklight also manages the versioning of applications, enabling new versions of the application to be created without the need for multiple versions of the back-end business applications, each with logic to handle the different requests. On the different platforms, features, such as notifications, are also handled and standardized by Worklight.

Figure 1-5 shows a typical architecture of how Worklight and CICS TS can be used in conjunction to extend the reach of your CICS applications to a mobile platform.

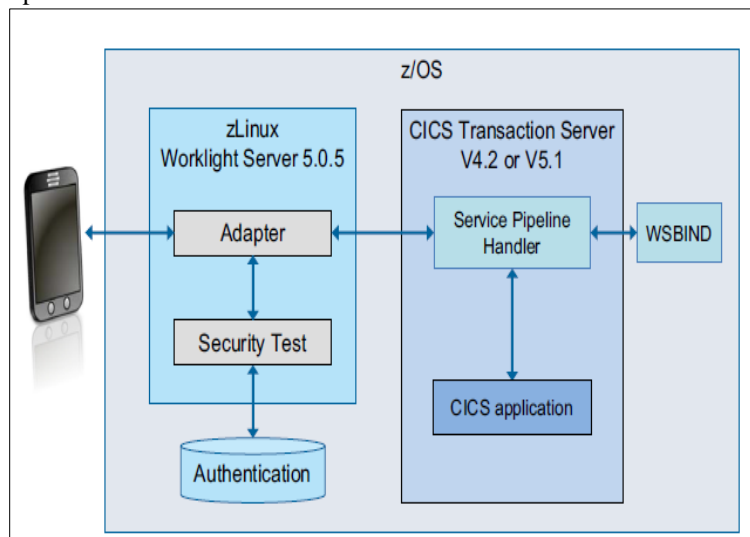


Fig. 8: mpbile devices access services hosted on CICS TS using worklight and CICS TS

E. REST and JSON

Representational State Transfer (REST) is a defined set of architectural principles by which you can design web services that focus on service resources. The REST architectural pattern takes advantage of the technologies and protocols of the World Wide Web to describe how data objects can be defined and modified.

In contrast to a request-response model such as SOAP, which focuses on procedures made available by the system, REST is modeled around the resources in the system. Each resource is globally identifiable through its Uniform Resource Identifier (URI). Because REST does not focus on the procedures and services provided by a system, a small number of actions are defined based on the existing Hypertext Transfer Protocol (HTTP) methods: GET, POST, PUT, DELETE, HEAD. The methods are used as shown in the following list:

- 1) GET Retrieve a resource representation.
- 2) PUT Modify a resource representation.
- 3) POST Create a new resource representation.
- 4) DELETE Delete a resource representation.
- 5) HEAD Retrieve a resource's metadata.

F. JSON

JavaScript Object Notation (JSON) is an open standard format for data interchange. Although originally used in the JavaScript scripting language JSON is now language-independent, with many parsers available in many languages. Compared to Extensible Markup Language (XML), JSON has many advantages. Most predominantly, JSON is more suited to data interchange. XML is an extremely verbose language: Every element in the tree has a name, and the element must be enclosed in a matching pair of tags.

Alternatively, JSON expresses trees in a nested array format similar to JavaScript. This enables the same data to be transferred in a far smaller data package with JSON than with XML. This lightweight data package lends itself to better performance when parsing.

JSON supports two structures: Objects and arrays. Objects are an unordered collection of name-value pairs, where arrays are ordered sequences of values. JSON also supports four simple types: Strings, numbers, Boolean expressions, and null values. This enables JSON to describe any resource. JSON can be seen as both human and machine-readable. JSON is an easy language for humans to read, and for machines to parse.

G. JSON with feature pack

Support has now been introduced, through the feature pack, to enable CICS to accept and receive HTTP web service requests using the JSON data format when calling your CICS applications using CICS web services. CICS can be configured so that your existing CICS applications can take advantage of this without any need for the application to be updated.

This configuration can be driven from either a JSON schema, where CICS will generate your high-level data structure format, or from the structure to a JSON schema.

Using this feature, CICS will process an HTTP payload in JSON data format, and convert the data into the high-level language structure of a target CICS application, whether that application is in Common Business Oriented Language (COBOL), PL/I, C, or C++. The data will then be passed to the CICS application in either Channels and Containers or the communication area (COMMAREA). When the application exits and control is returned, CICS will convert the output of the CICS application (in its high-level language structure) back to the JSON data type. This response is then sent back to the service requester through an HTTP payload. The JSON to high-level language structure conversion service, and the high-level language structure to JSON conversion service, are also available through a LINKable program. CICS applications can use this LINKable program to call any external service expecting data in the JSON format. This enables your business applications to focus on business logic, and enables CICS to handle the burden of the data transformation.

When configured from a JSON schema, CICS can also be used to call your CICS applications in a RESTful architectural style. You are able to configure your JSON web services to call a number of different CICS programs based on the HTTP method used to make the call. The CICS programs then also receive further information, such as the query string that was used to make the RESTful request. This information can then be used in the logic of your CICS applications

H. IBM Worklight

Worklight provides an open and extensible mobile application platform. This platform enables organizations of all sizes to develop, run, and manage Hypertext Markup Language (HTML5), hybrid, and native mobile applications.

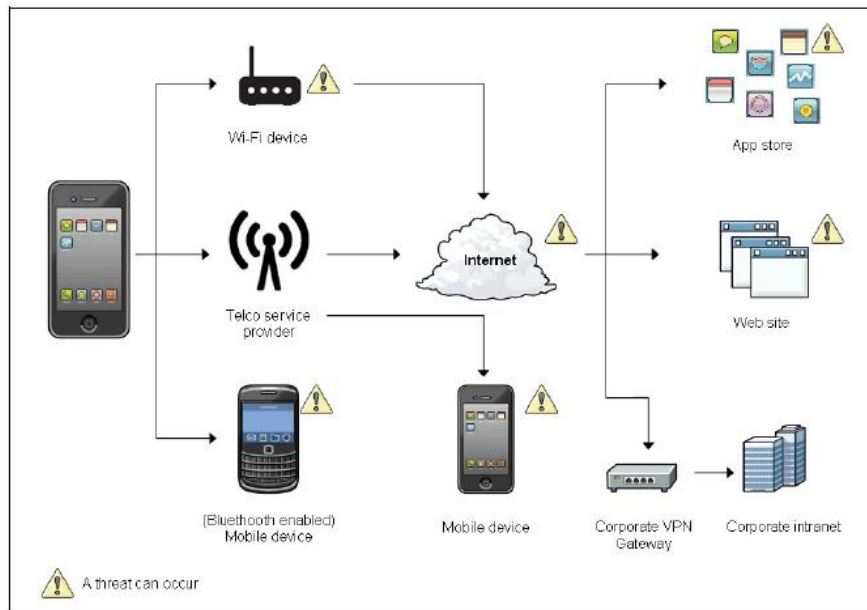


Fig. 9: IBM Worklight

The Worklight platform consists of five main components: Worklight Studio, Worklight Server, Worklight device runtime components, Worklight Application Center, and Worklight Console.

I. Worklight Studio

Worklight Studio is an Eclipse-based integrated development environment (IDE), which can be used by your organization's mobile application developers for coding rich, cross-platform applications with a single, shared code base. This is achieved using standards-based technologies, and does not require the use of code translators or proprietary interpreters.

Worklight Studio can be used to build applications for most current mobile operating environments, including iOS, Android, BlackBerry, Microsoft Windows Phone, and Windows 8, as well as mobile web browsers. In addition, you can create applications for feature phones and embedded systems with Java Platform, Micro Edition (Java ME).

Developers can make use of third-party libraries and frameworks, such as Apache Cordova, Dojo Mobile, and jQuery Mobile. Furthermore, native code or JavaScript can be used to access mobile device application programming interfaces (APIs). Application testing can be completed using the emulators provided with Worklight Studio. In addition, the ability to record, edit, and play back codeless test scripts on physical or emulated devices, reduces the time to value.

J. Worklight Server

The Worklight Server is mobile-optimized middleware that provides a secure gateway between mobile applications, enterprise systems, and cloud-based services.

Multiple security mechanisms are supported, including integration with existing authentication and security methods. These features support the safeguarding of the device, application, and network layer.

The Worklight Server adapters add value to your mobile solution by providing server-side application code that connects to back-end systems and delivers data to and from mobile applications. Necessary server-side processing can be performed, reducing the need for processing on the mobile devices, or the modification of existing back-end systems and applications.

K. Worklight device runtime components

Mobile applications deployed using IBM Worklight include client-side runtime APIs that embed server functionality in the applications. These APIs support the authentication between applications and Worklight Server, provide on-device encryption, and provide for the remote disablement of applications.

The APIs also provide a bridge between standard web technologies, such as HTML5, Cascading Style Sheets (CSS3), and JavaScript, with the native functions of the various mobile platforms.

The embedded code also assists with the push notification framework, and supports usage and event-based reports.

L. IBM Worklight Application Center

Employees within your organization might be familiar with the application stores of the various mobile platforms through personal use. The Worklight Application Center enables your company to set up an internal one-stop shop enterprise application store for the distribution of pre-release and production-ready applications.

Existing frameworks, such as Lightweight Directory Access Protocol (LDAP) and access control list (ACL), can be used to control distribution by department, job role, function, geographical area, or other topology. Your enterprise application store can be used to obtain feedback, feature requests, and enforce upgrades.

M. Worklight Console

Worklight Server is administered through a graphical web-based interface, the Worklight Console.

The Worklight Console supports the management of the server, adapters, and push services.

Worklight Server can act as a gateway between many mobile devices and CICS. Taking this approach, you protect your CICS systems by preventing direct access from mobile devices.

You also benefit from the many other features of the Worklight platform concerning application development, deployment, and management.

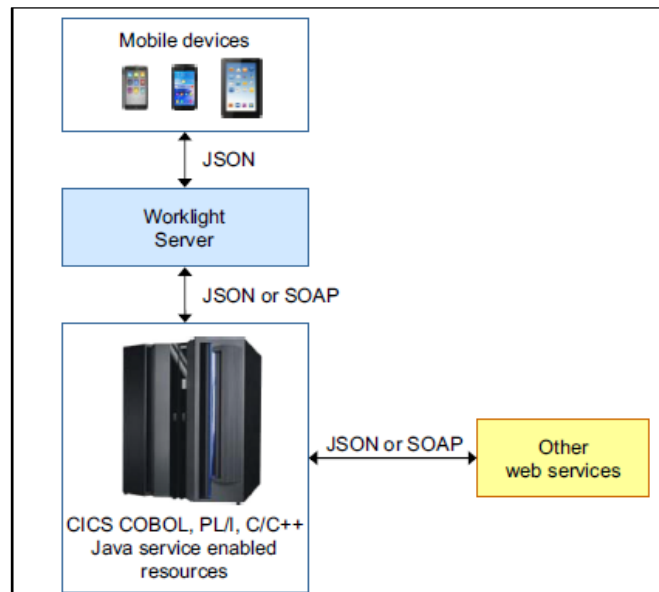


Fig. 10:

V. CONFIGURATION

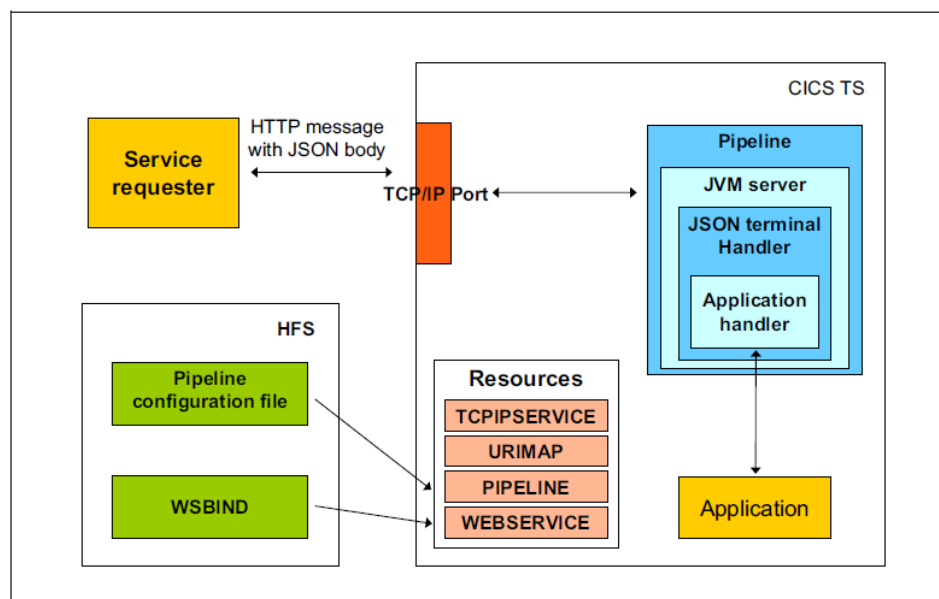


Fig. 11:

- 1) A request is made by the Hypertext Transfer Protocol (HTTP) protocol on a port opened by an installed Transmission Control Protocol/Internet Protocol (TCP/IP) TCPIP SERVICE definition. The port is monitored by the CICS sockets listener (CSOL) transaction.
CSOL attaches the transaction specified in the TRANSACTION attribute of the TCPIP SERVICE definition. Usually, this is the CICS web attach transaction CWXN.
- 2) CWXN matches the incoming request with a Uniform Resource Identifier (URI) URIMAP by scanning all URIMAP definitions for one that has its USAGE attribute set to PIPELINE and its PATH attribute set to the URI found in the request. The URIMAP definition indicates the PIPELINE and WEBSERVICE definitions to be used, and the TRANSACTION that should be attached to process the PIPELINE. The transaction is usually CPIH.
- 3) As with SOAP web services, the WEBSERVICE definition points to a WSBIND file. This will be used later for data transformation between JSON and application data. The WSBIND file will have been created using the JSON assistant.
- 4) Pipeline processing then takes place, passing the request through any defined handlers.
- 5) The JSON terminal handler and CICS application handler are called. The JSON data is converted into application data, using the language structure description in the WSBIND file.
The application handler links to the application program. Note, the application program might be run in a different CICS region.
- 6) Finally, the PROGRAM output is captured and sent back to the requester as an HTTP response with JSON data.

VI. CONCLUSION

Still in the industry there are people who believe that mainframe is an outdated technology and its out of the scene. The power and capabilities of mainframes in supporting mobile applications and smartphones will definitely shows that the future of this industry is safe on mainframes. This will definitely attract more people to mainframes in coming days.

REFERENCES

- [1] IBM redbook on CICS Mobile implementation
- [2] CICS TS 5.1 installation and operations guide – IBM
- [3] www.share.ibm.co.in
- [4] Mainframes- Bridging legacy Systems, building digital futures – Brian Ekwall
- [5] CICS Transaction Server 5.1- Operations and utilities Guide- IBM