Reference Architecture for Mobile Infrastructure on System z

November 18, 2014

Steve Wehr, Nigel Williams,
Wilhelm Mild, Frank van der Wal

Abstract
This guide is meant to educate the IBM field force on standard architectures and configurations that can be used to create the infrastructure for mobile applications on System z.

Purpose
This guide is meant to be used by z IT Architects to prepare mobile architecture presentations for customers. This guide is meant to show what is possible, how to get started, and where the major components of a mobile solution would run on z. Therefore it will be mostly charts, with enough text to explain the decisions shown in the charts.

How to use the z System Mobile guides…

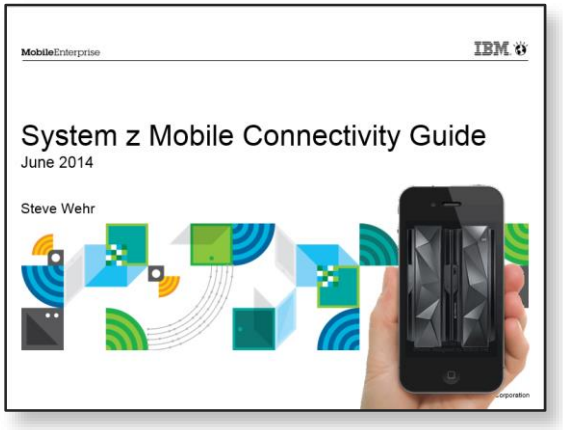
We recommend reading these in this order…

1

Contents

- Components of a mobile architecture.
- Mobile topology choices.
- MobileFirst Platform in production.
- MobileFirst Platform in dev/test
- Scalability and performance considerations.
- Conclusion

2



Contents

- Summary of z mobile connectivity options, including MobileFirst Platform Foundation
- Details about
  - Push Notification
  - IBM API Management
  - CICS
  - IMS
  - DB2
  - WMB

3

Contents

- Introduction to the MobileFirst security products – what they do and how they relate to System z.
- Building a Secure Enterprise Mobile environment using the MobileFirst Security products.
- Use Cases and Reference Architectures.

Contents

- Introduction, and major components of a mobile architecture.
- Mobile topology choices.
- Positioning for WebSphere Portal and MobileFirst Platform.
- Architecture for MobileFirst Platform Server in production.
- Architecture for Security
- Architecture for MobileFirst Platform server in dev/test
- Scalability and performance considerations.
- Conclusion

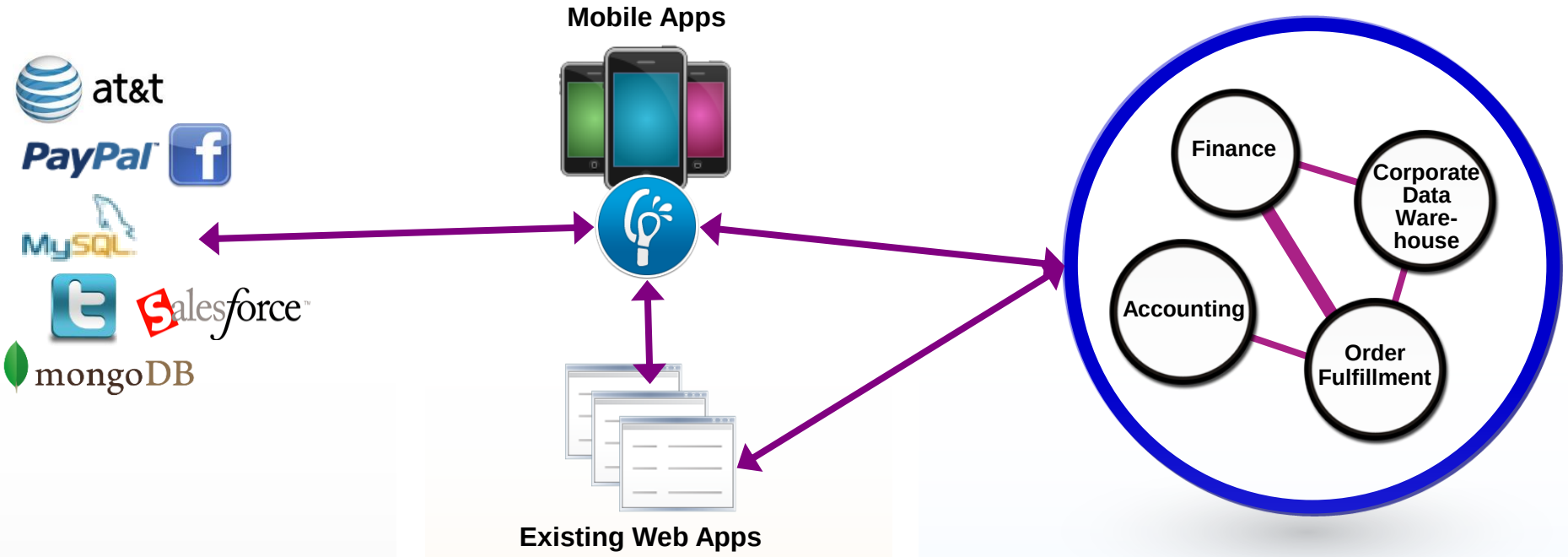
Reference Architecture Essentials / Definition

- A Reference Architecture captures the essence of the architecture of a collection of solutions. The purpose of a Reference Architecture is to provide guidance for the development of architectures for new versions of the solutions or extended solutions and product families.
- A Reference Architecture is created by capturing the essentials of existing architectures and by taking into account future opportunities, ranging from specific technologies, to patterns to business models and market segments.
- A Reference Architecture is a documented multi-tiered architecture
  - Architecture and implementation options
  - Recommended technologies
  - Considerations for functional and Non-functional Requirements
  - A standardized approach for description of mobile system architectures and high-level designs

System z bridges Systems of Record and Systems of Engagement

Systems of Engagement

Systems of Record



Cloud APIs

Systems of Engagement are cloud-based, decentralized, support rapid app development

Linux on z

z/OS

Systems of Record are well integrated, trusted repositories

Customer Requirements

Customers who choose to host mobile applications on System z are most interested in these infrastructure characteristics

- Access to services and data on z/OS
- Reliability of System z for a 24 X 7 operation
- Rapid, automatic scalability for mobile workloads
- End to end security
- Maximizing utilization of resources (extreme virtualization, CPU to 100%)
- Integrated network topology

The architectures in this guide will address these requirements

Introduction

The major components of a Mobile Architecture on System z

Owner: Steve Wehr

Tiered mobile environment

Mobile Devices

- iOS
- Android
- Blackberry
- Windows Phone

JSON

Systems of Engagement

Mobile Enterprise Application Platform (MEAP)

- Security
- Back-end connectivity
- Protocol conversion
- Caching Services

SOAP
XML
JSON
MQ
HTTP

Systems of Record

- CICS
- IMS
- WebSphere AS
- DB2
- Analytics

A Complete Mobile Environment Consists of

Mobile Device

Browser

Mobile Applications

Mobile Application Development

Security Layer 1

Caching Service (Optional)

Mobile Middle tier server
System of Engagement / MEAP

Mobile Device access Interfaces

Standard, flexible, Secure, protocols i.e. JSON, MQTT

Mobile Application Mgmt
App Store, handling Apps for different platform OS, versioning, updates

Mobile Device Mgmt
Management of device level security, access, policies.

Mobile Services Mgmt
Security services, push enforcement, integration services.

Operational support
Monitoring, workload mgmt, App analysis

Integration Interfaces

Functional Interface Support (standard, Universal, Transactional, with Protocols, i.e.HTTPs, JSON, MQ)

Resource Access Support (CICS, Databases,back-end systems)

Caching Service (Optional)

Security Layer 2

System of Record.
Common Data Services / Transaction Processing Platform

SoR Services and Applications

SoR Data
(i.e.DB2, Oracle)

Infrastructure

Server

Storage

Network

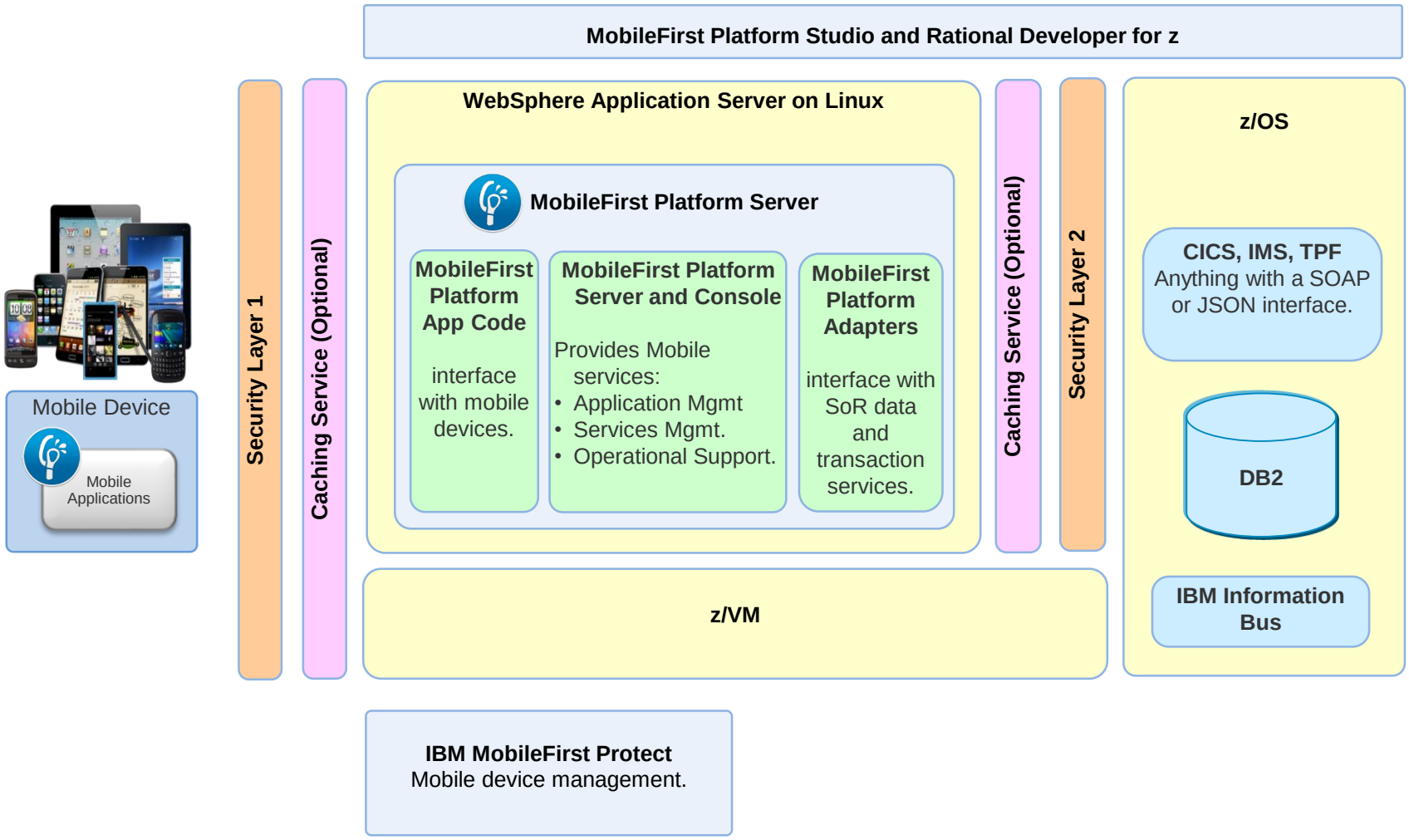
The Mobile Middle Tier adds these components, that are not present in typical web applications.

- **Mobile Device Access Interfaces**
  - Mobile devices can interact with the Mobile Middle Tier (Runtime Servers) using open source protocol standards for mobile devices like JSON or MQTT. The interfaces supported by the Middle tier server qualify it for universality and flexibility.
- **Mobile Application Management. (MAM)**
  - The ability to manage –multiple– applications with respect to versions, device specifics and OSes
- **Mobile Device Management (MDM)**
  - Management of device level security, access, policies.
  - The mobile device management is responsible to support multiple mobile devices and deliver an ease of use management for new devices and the process to keep existing ones current with the PUSH notification
- **Mobile Services Management (MSM)**
  - A variety of mechanisms to help control and manage mobile apps regardless of their type and OS, for example
    - Application versions (to block faulty or out-of-date version and seamlessly direct people to the (enterprise) app store
    - Authentication and access control
    - Push Services Management
    - Usage reports and analytics
- **Mobile Operational Support**
  - Mobile applications behave different than traditional applications, and have a much shorter life cycle management and change behavior, are more dynamic and have to respond very fast to customer requirements.
  - The behavior of the Mobile Server has to be monitored and (automatic) actions have to be taken to avoid unplanned outages.
- **Integration Interfaces**
  - The interfaces to access and interact with data services and transactional services, enable an integration of back-end systems such as transactional environments with CICS and Data Services from different Databases and platforms.

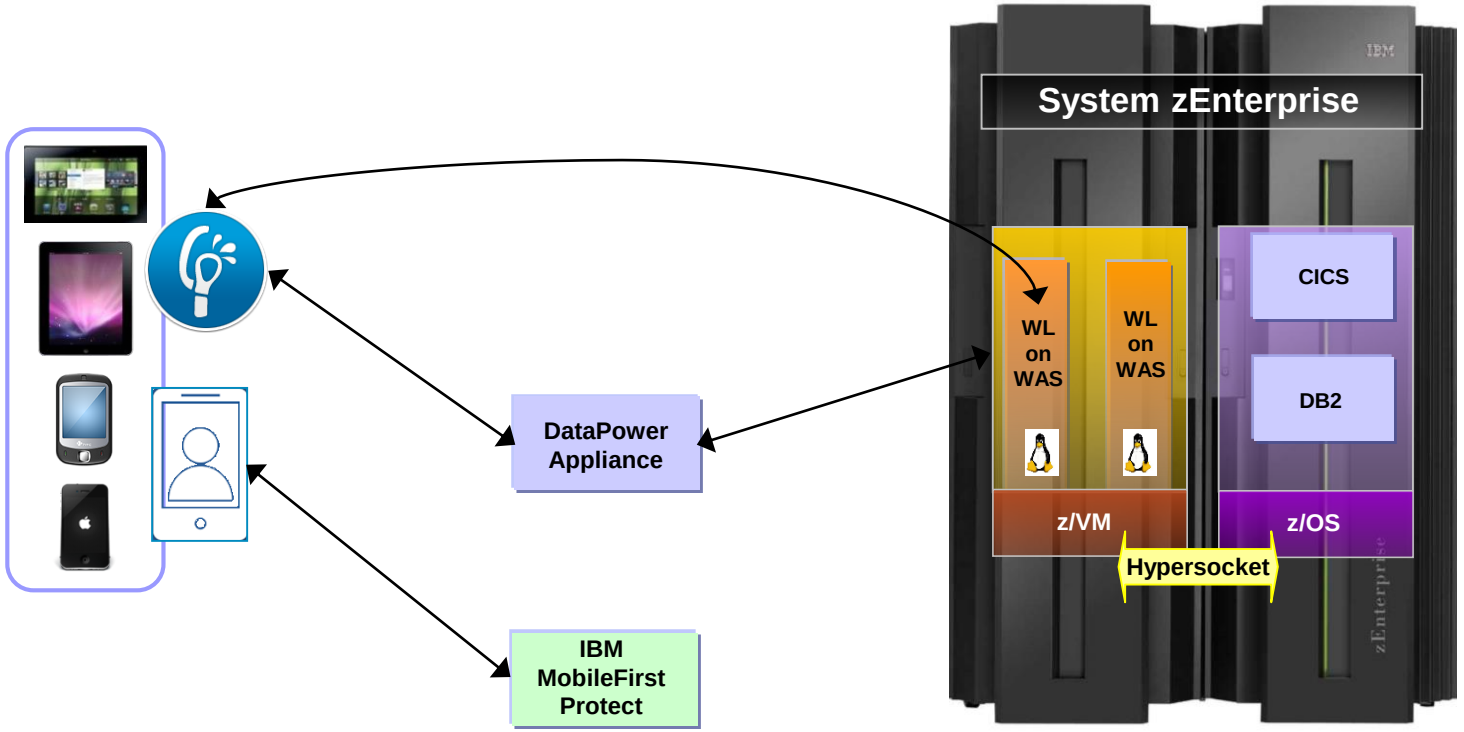
Other Mobile Terminology

- **Mobile Enterprise Application Platform (MEAP)**
  - MEAP is a comprehensive suite of products and services that enable development of mobile applications for Enterprises
  - MEAPs address the difficulties of developing mobile software by managing the diversity of devices, networks and user groups at the time of deployment and throughout the mobile solution's lifecycle. Unlike standalone apps, a MEAP provides a comprehensive, long-term approach to deploying mobility. Cross-platform considerations are one big driver behind using MEAPs
  - IBM MobileFirst Platform is an example of a MEAP.

On System z, this looks like



High level mobile architecture on System z



System z Mobile Enterprise with IBM MobileFirst Platform Server

Linux on z

z/OS

Device

PhoneGap

Worklight

- Security and Authentication
- Back-end Data Integration
- Caching and local data

Application Code

Linux on z

MobileFirst Platform on WAS

Server-side App Code	**Application Center Enterprise App Store**
Worklight Console	JSON Translation
Push Notifications	Authentication
Analytics	Adapter Library

SAP

SQL

WMB

SOAP

HTTP/REST

Cast Iron

z/OS

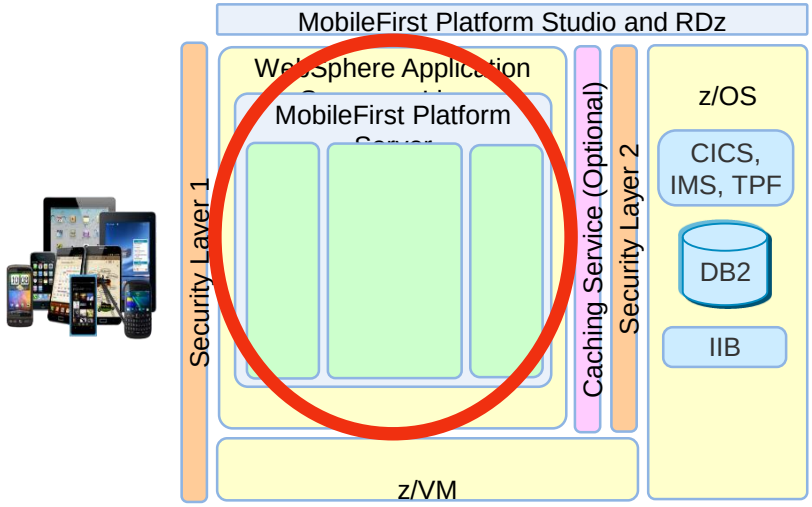
CICS

Worklight Video: http://www.youtube.com/watch?feature=player_embedded&v=zHnFw70XXXo

Mobile Topology Choices

What are the various types of mobile applications, and what are the various ways mobile applications can access System z data and transactions.

Owner: Steve Wehr, Nigel Williams, Wilhelm Mild.



What is a Mobile App?

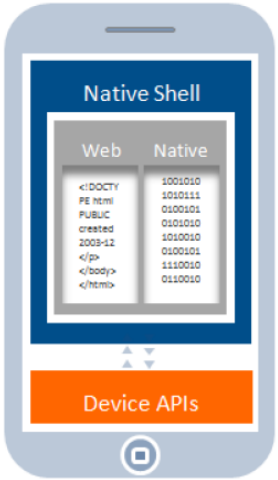
Browser Access

Written in HTML5 JavaScript and CSS3. Quick and cheap to develop, but less powerful than native.

Browser Access

Hybrid Apps - Mixed

User augments web code with native language for unique needs and maximized user experience.



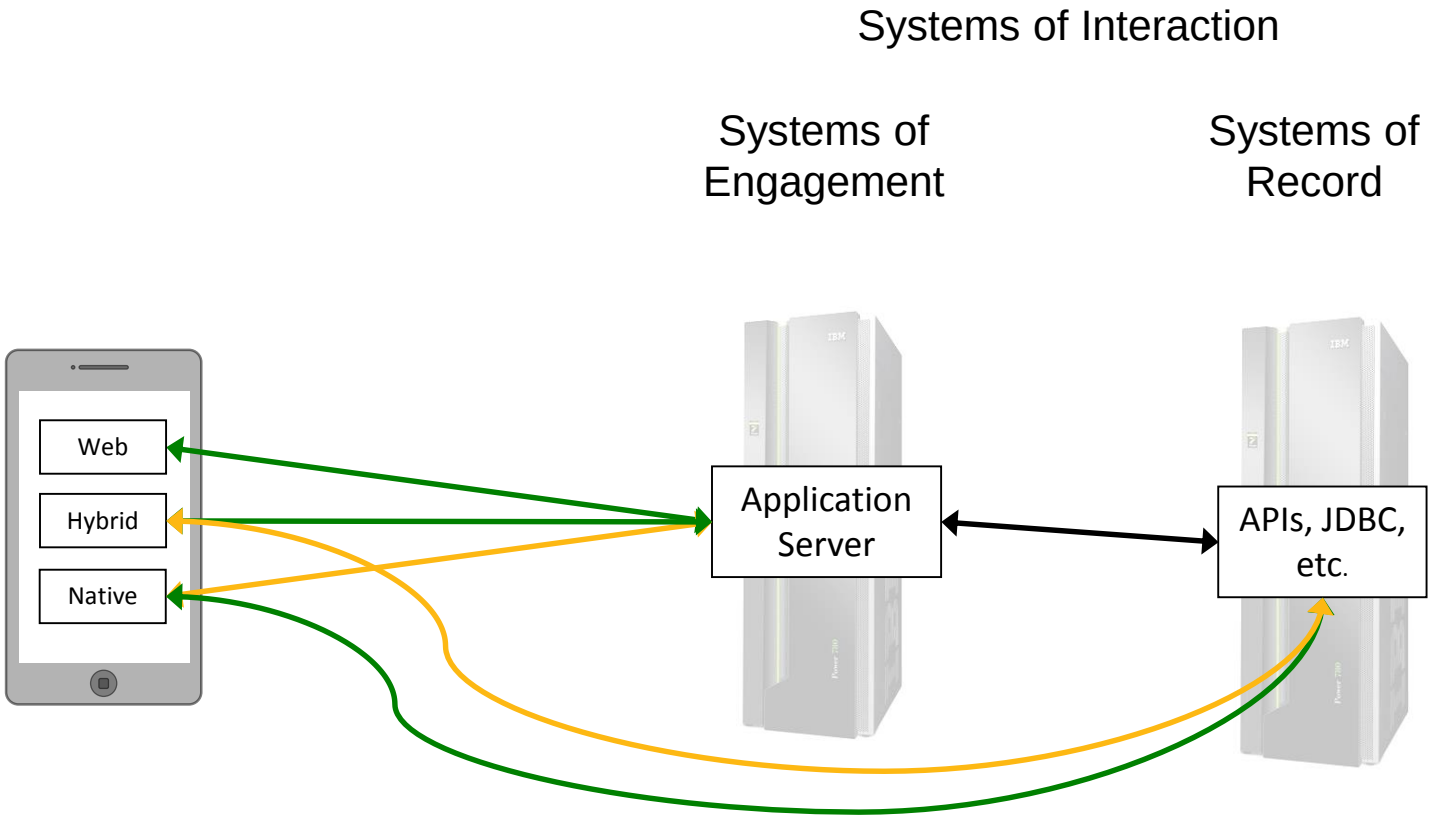
Downloadable

Native Apps

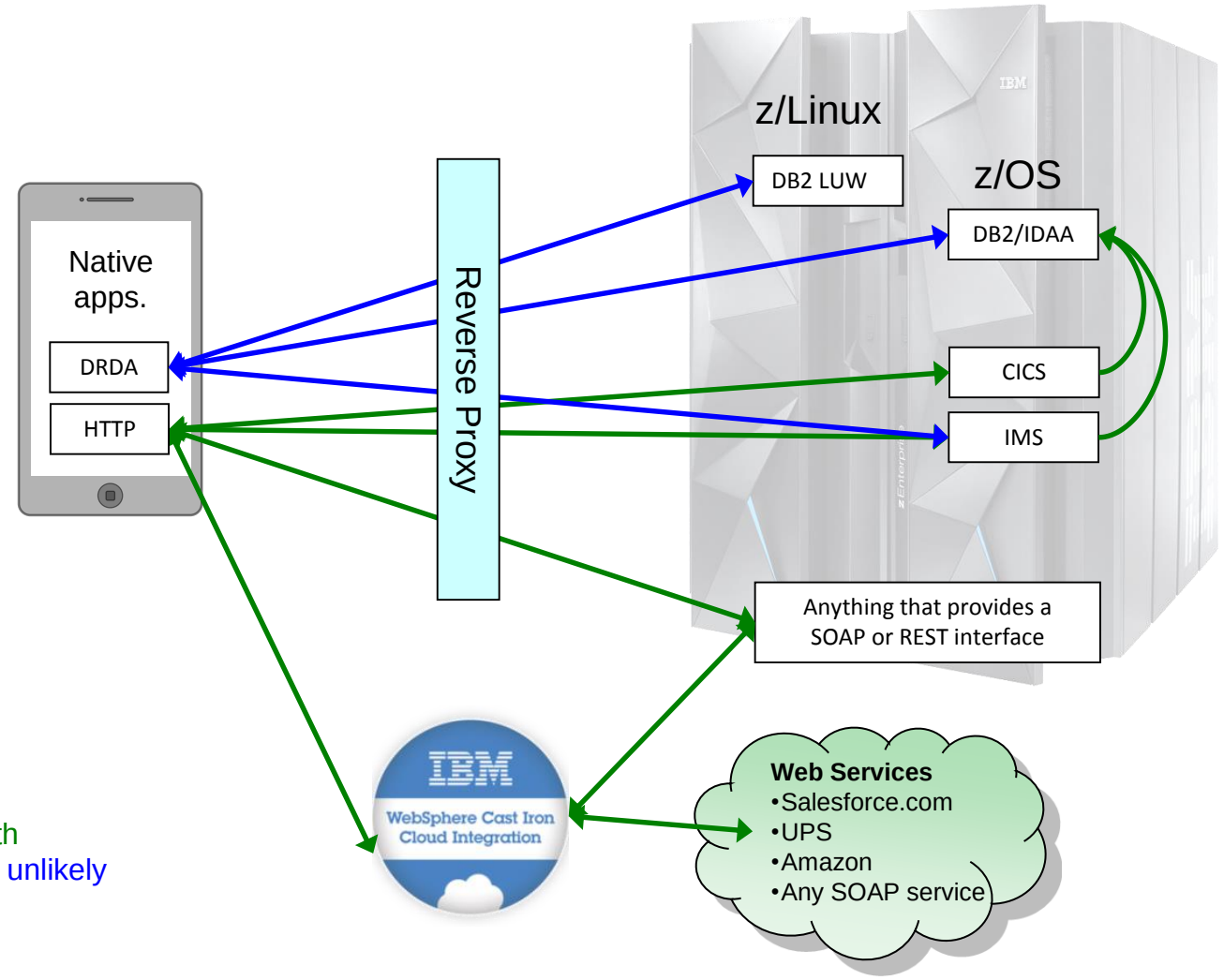
Platform-specific. Requires unique expertise, pricy and long to develop. Can deliver higher user experience.

Downloadable

What is a Mobile App?

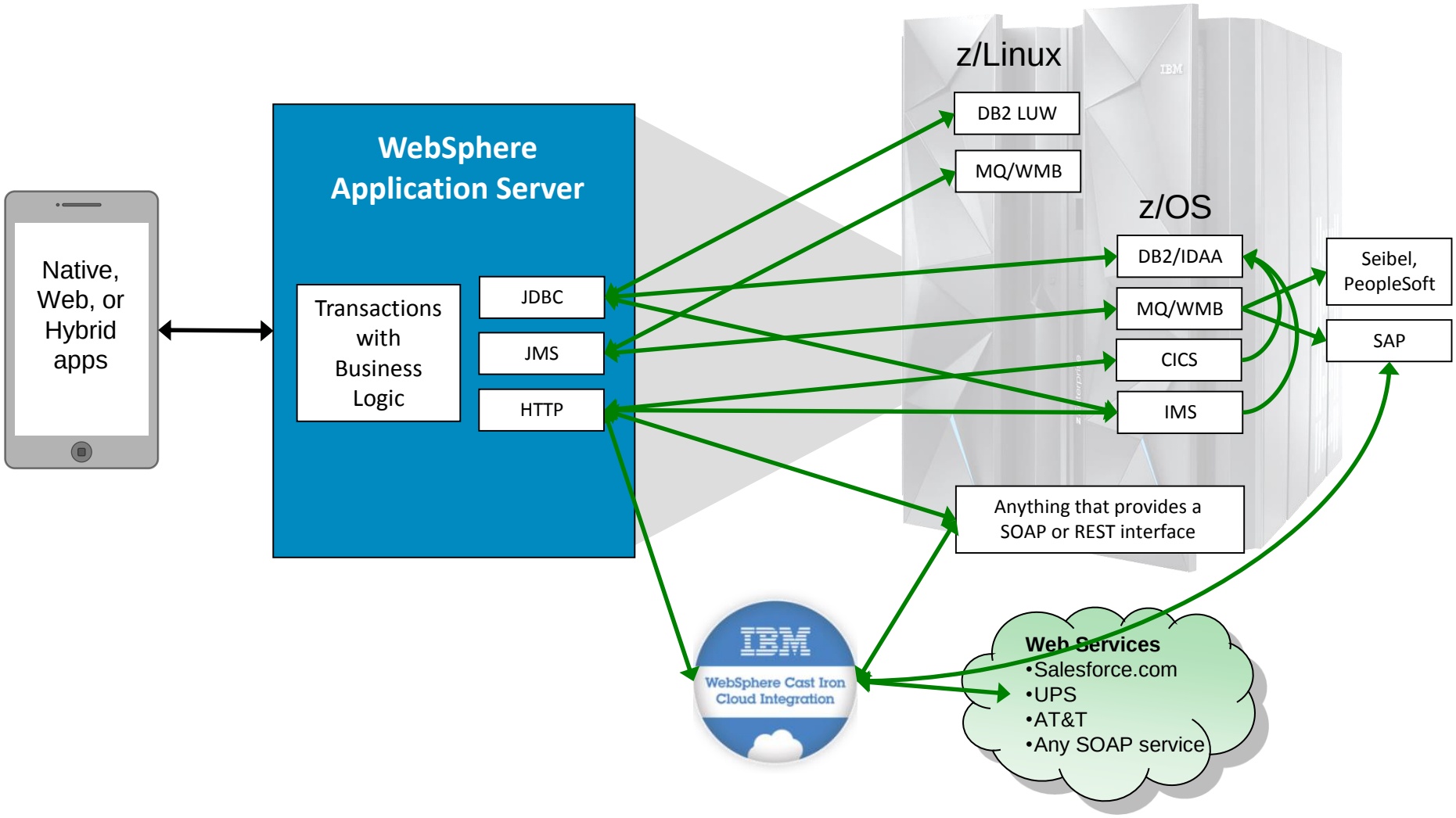


Mobile Native App Connectivity to System z – without an app server

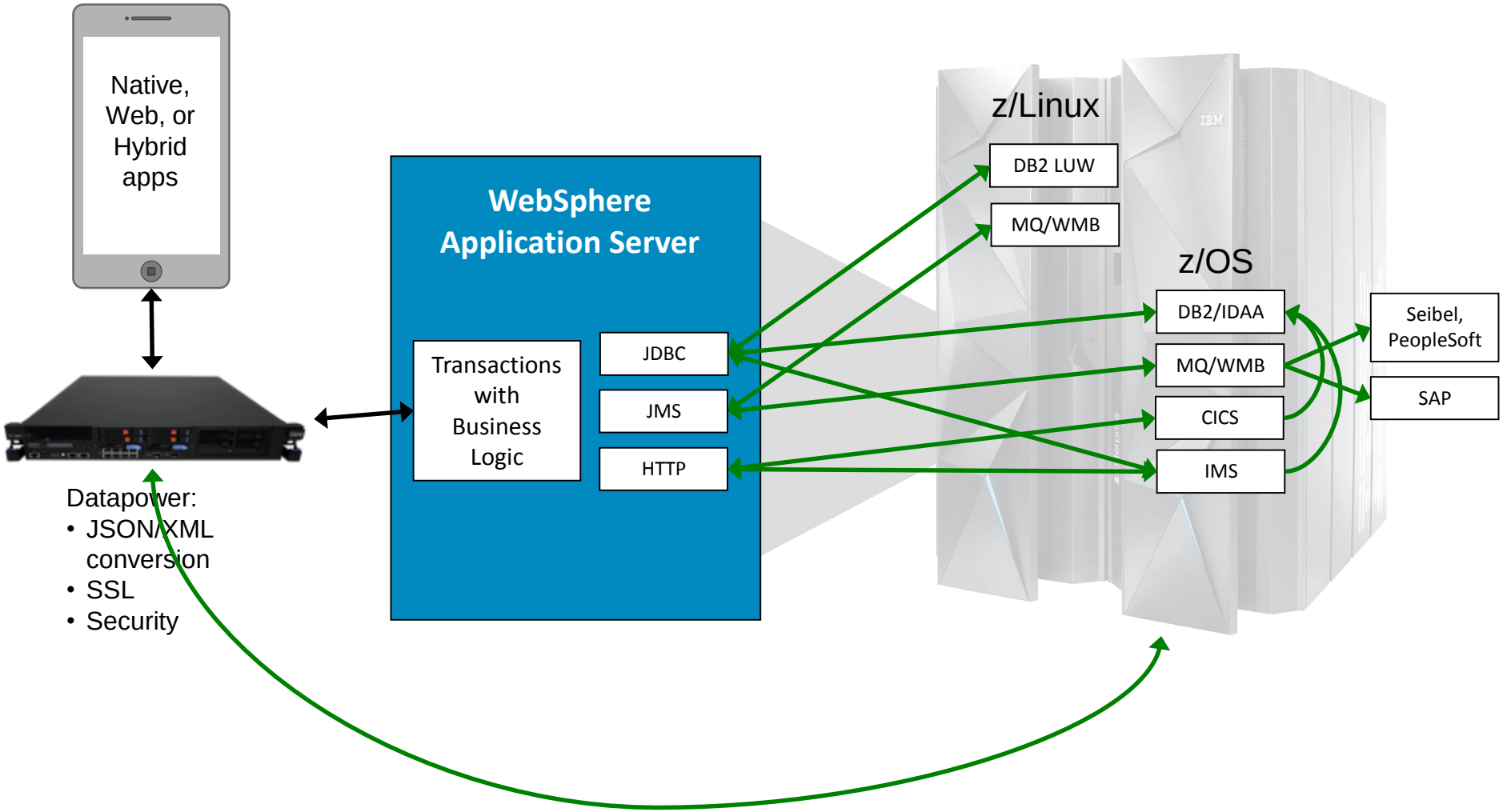


Available Path
Available but unlikely

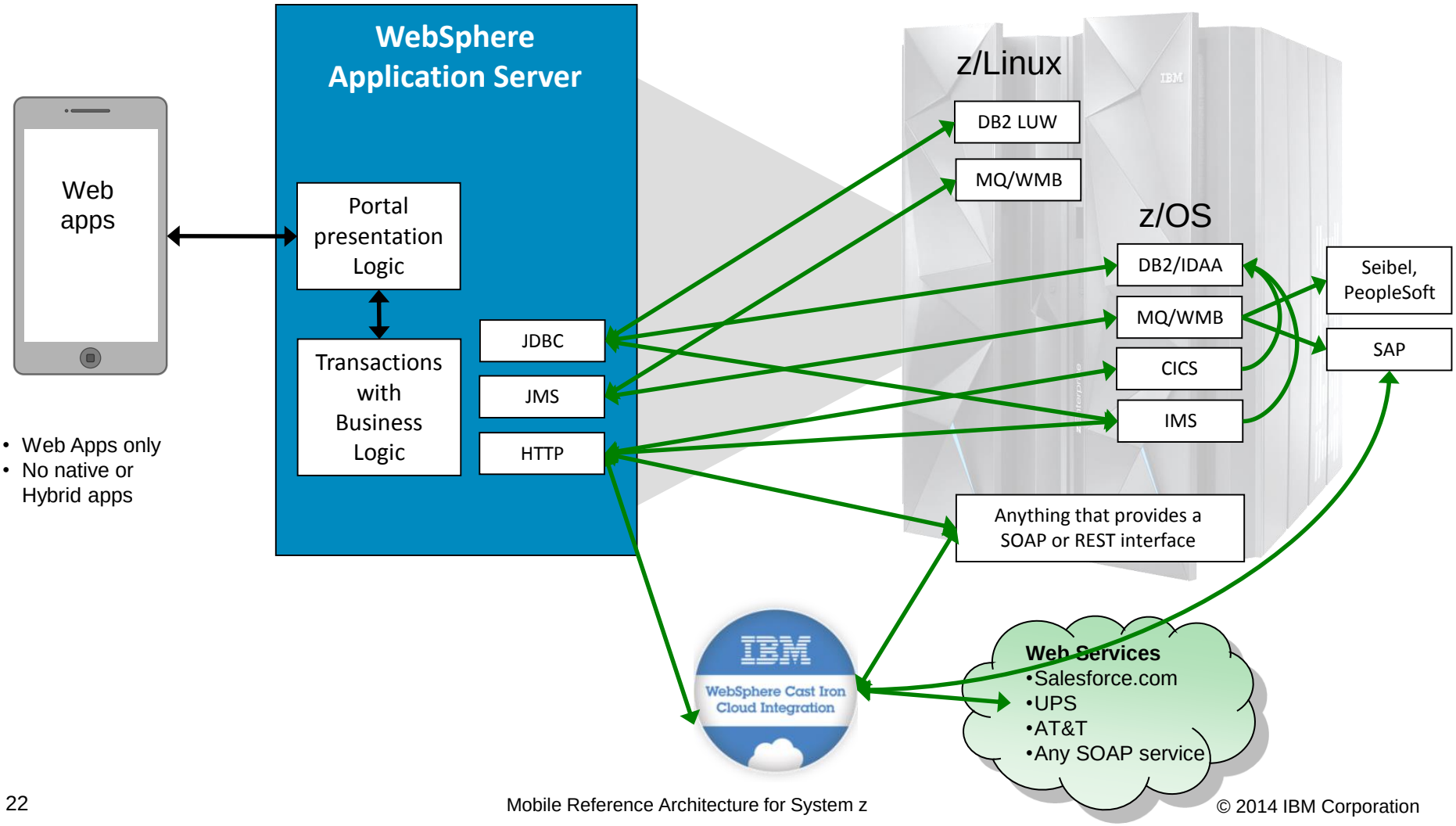
Mobile App Connectivity to System z -- without MobileFirst Platform



Mobile App Connectivity to System z – via Datapower (DMZ)

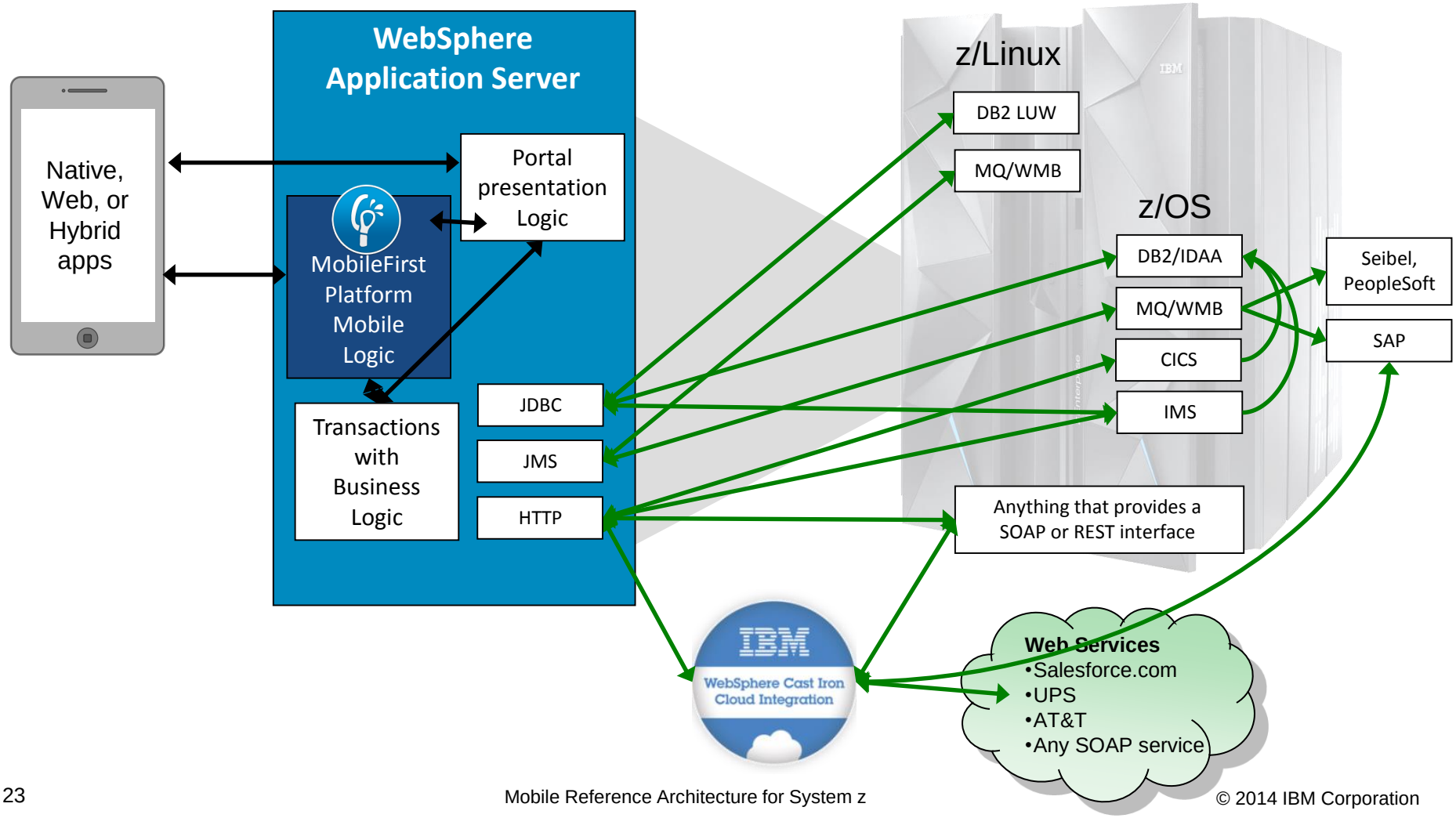


Mobile Web App Middleware options to System z – with Portal

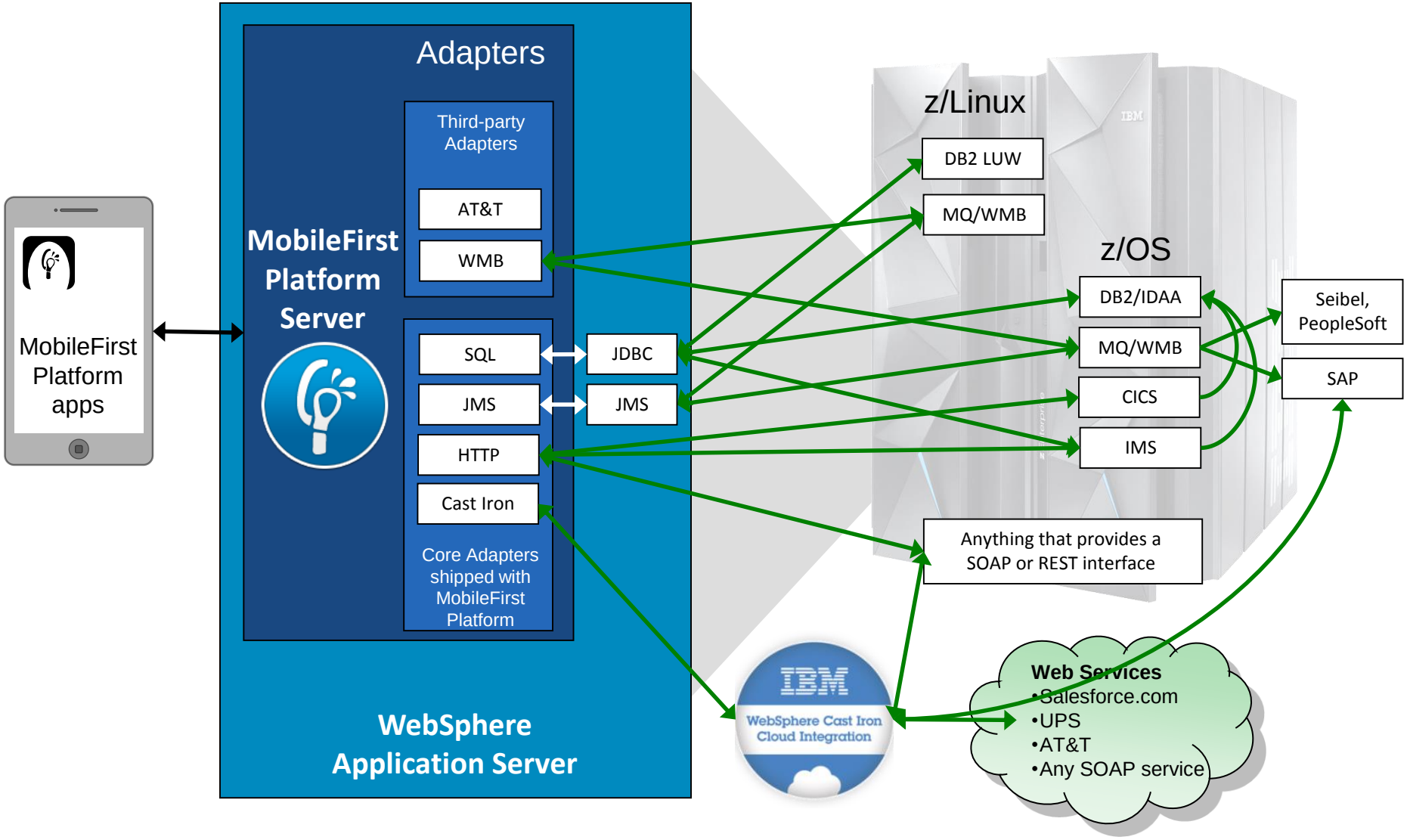


- Web Apps only
- No native or Hybrid apps

Mobile App Middleware options to System z – MobileFirst Platform and Portal



Mobile App Connectivity to System z -- via MobileFirst Platform



MobileFirst Platform Benefits for System z Customers

MobileFirst Platform Server

- **Adapters** ease communication with z/OS services like DB2, CICS, IMS, and MQ, with third-party solutions like SAP, and with could-based services like AT&T.
- **Unified Push API** for push services to multiple mobile devices.
- **Performs protocol transforms** from SOAP to JSON.
- **Enterprise App Store** delivers distribution and management of mobile applications within a company
  - Supports all mobile platforms centrally – iOS, Android, Windows Phone, Blackberry
  - Provides versioning and updates. Can push app updates to all users.
  - Enforces security, renewal, and expiration.
  - Centralizes rating and feedback information.
  - Controls who can modify or install an application.
- **Captures analytics** to give insight on mobile apps usage.
- **Runs in System z Linux**, providing scalability in a highly virtualized z/VM environment, and low-latency connections to z/OS data and transactions using System z internal network technoligies.

MobileFirst Platform Studio

- Eclipse based mobile **Integrated Development Environment (IDE)** that integrates with RDz.
- **Run-time components** allow you to create one version of your mobile app that runs on all types of mobile devices.
- Integrated **Tealeaf** helps diagnose usability problems.
- **Mobile device simulator** simplifies unit testing.

Architectural decisions

Architectural Decision	Rationale and decision points
When to connect mobiles directly to z/OS subsystems like CICS	It is possible that in certain limited implementations mobiles will connect directly to System z services. However, for security reasons, most customers will not want to do this. A reverse proxy (in a separate security domain) at least will be an intermediary between the mobile device and the SoR. Database connectivity is difficult since the DRDA driver in the mobile device must directly communicate with the database.
When to use Web apps connected to WAS z/OS	Web apps running in WAS can present web pages formatted for mobile devices. Mobile web apps are easy to code and require no additional tooling, using existing HTML5 skills. Since all the app logic runs in the WAS server, no app is required on the mobile device, no local data to protect, etc. However, if the mobile app requires access to native OS functions like the camera or GPS, these are not available to web apps. Also WAS does not have all the mobile capabilities offered by MobileFirst Platform.
When to use WebSphere Portal Server	WebSphere Portal can automatically reformat web applications to fit mobile screens.
When to use DataPower	DataPower can be used for threat protection, AAA (Authentication, Authorization and Audit) policies and data transformation (e.g mapping JSON to XML). DataPower can be used as a mobile gateway with or without MobileFirst Platform.
When to use MobileFirst Platform	IBM MobileFirst Platform supports native, web, and hybrid applications, and the MobileFirst Platform suite provides many advantages for these mobile apps. (See the list on the previous chart).

Why MobileFirst Platform?

We have shown here an introduction to other methods of creating mobile apps that talk to System z. But we will use MobileFirst Platform as the foundation for all the coming sections because MobileFirst Platform best satisfies the customer requirements (from page 7) that we set out to solve.

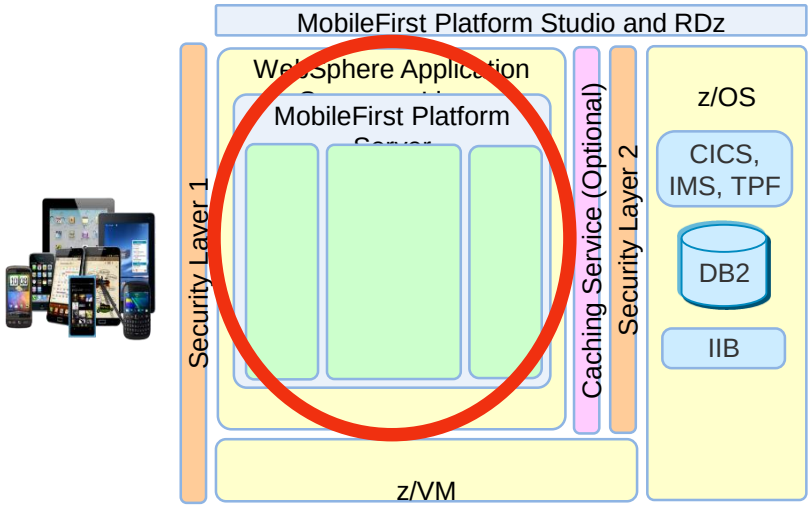
For more detailed information on how to connect mobile applications to z/OS subsystems, refer to the "System z Mobile Connectivity Guide".

For more information about IBM MobileFirst Platform, refer to:

- MobileFirst Platform Sales Kit.
- MobileFirst Platform InfoCenter.

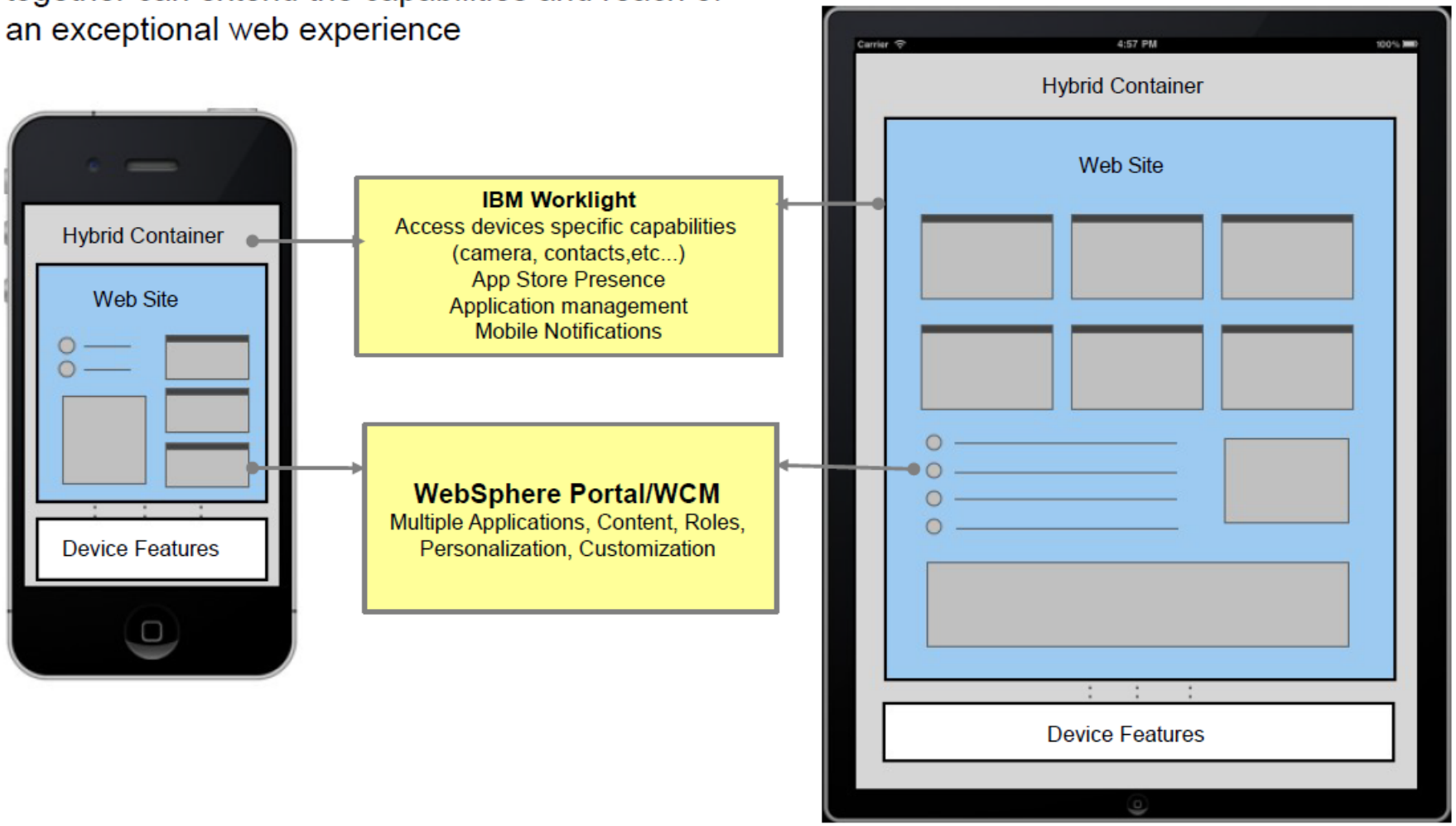
Positioning for WebSphere Portal and MobileFirst Platform

Owner: Wilhelm Mild



Hybrid – MobileFirst Platform and WebSphere Portal together

WebSphere Portal/WCM and IBM Worklight used together can extend the capabilities and reach of an exceptional web experience



WCM = Web Content Manager

IBM solutions address both mobile needs

Multichannel Sites

Provide a consistent integrated web experience across multiple channels (desktop browser, smart phones, tablets, etc..)

IBM WebSphere Portal Solutions

Mobile Applications

Provide an experience that takes full advantage of the device and its ecosystem

IBM MobileFirst Platform Solutions

WebSphere Portal and MobileFirst Platform

- A **website** aggregates:
  - web content
  - multiple web applications into a single user interface
  - works across multiple channels, including desktop browser, smartphones and tablets

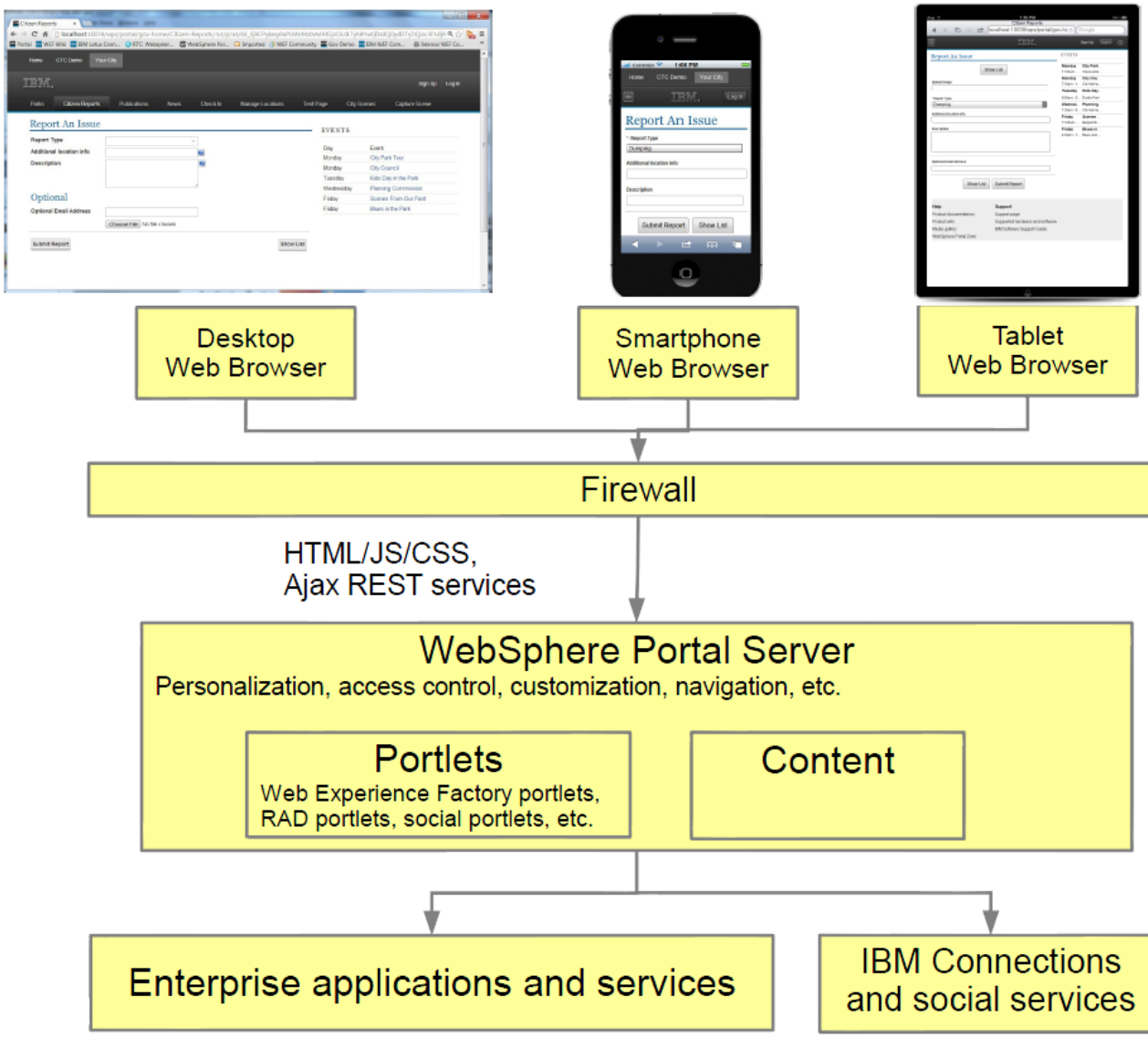
  A simple example might be an airline's website

  They probably have a mobile website, too, for smartphones and tablets.
- WebSphere Portal is the right platform for serving a personalized multi application website.

- A **web application** is custom-built and often targets specific tasks.
  - For example, your favorite airline app from an app store that lets you:
    - book a flight or
    - reserve a seat
  - It usually contains a subset of the website's features, targeted to what you can practically do on the device.
- IBM MobileFirst Platform provides the ability to create both native and hybrid applications

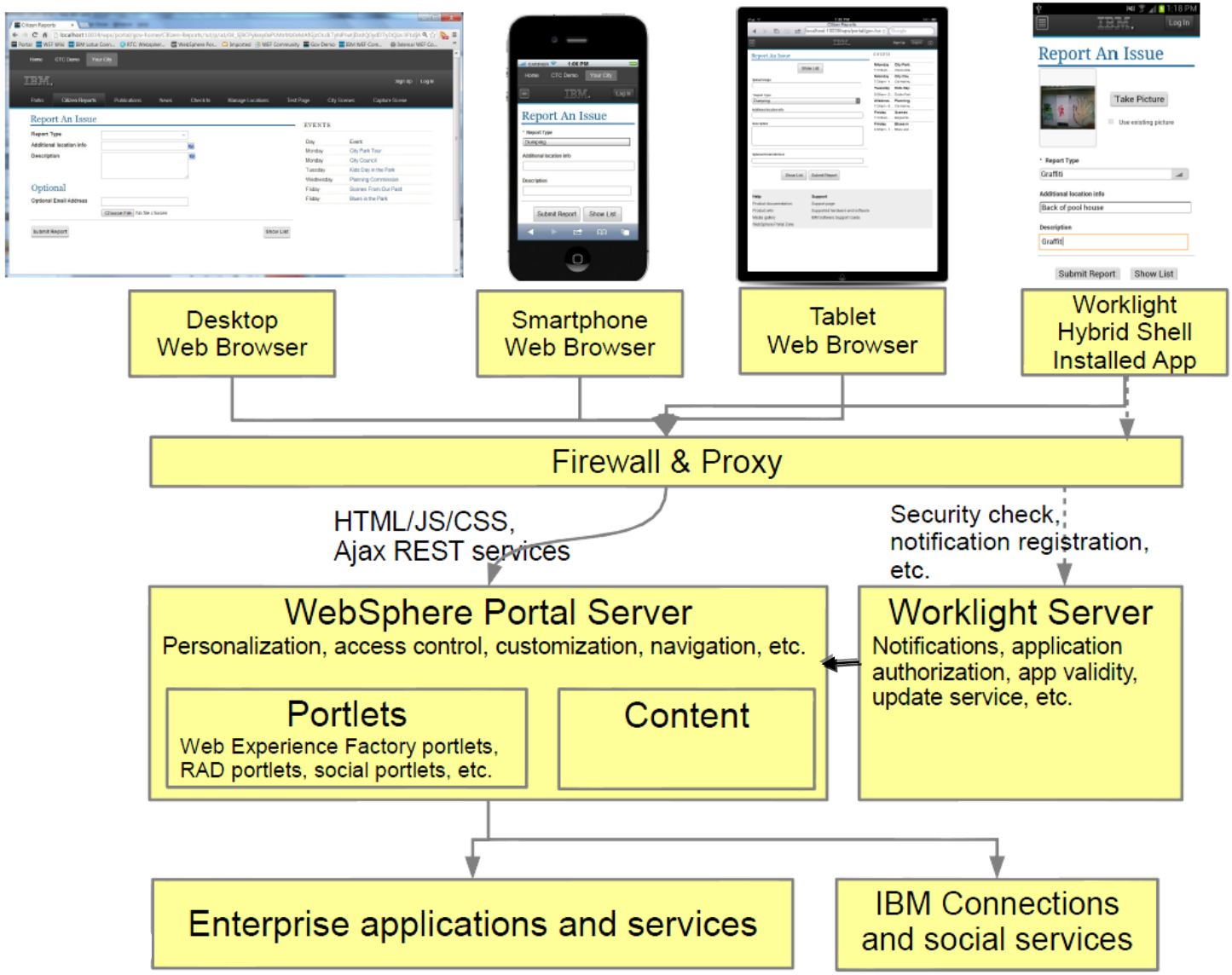
You can use either WebSphere Application Server with MobileFirst Platform Server as the back end (if you are creating hybrid apps), or WebSphere Portal as the backend (if you are creating hybrid websites).

IBM WebSphere Portal Server – for Mobile solutions



Apps on Mobile devices:
- with WPS Web Sites
- aware of Mobile device disply charcteristics

Multi-channel site – with WebSphere Portal and MobileFirst Platform



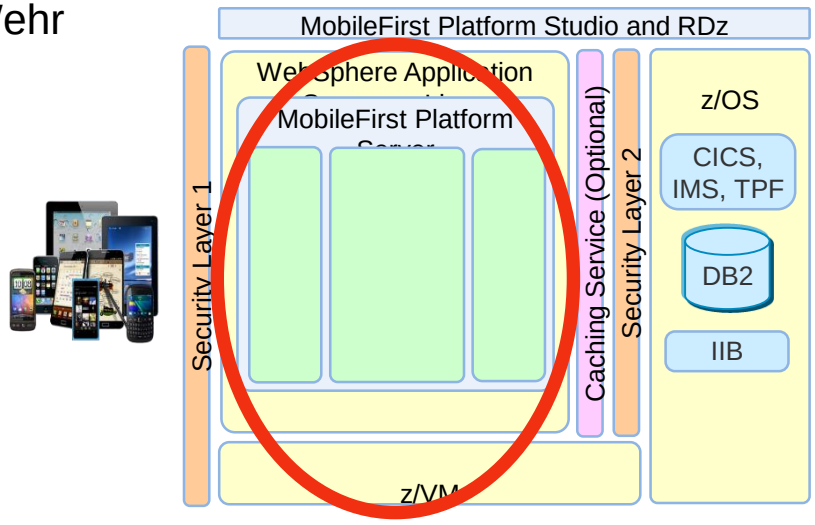
Apps on Mobile devices:
- with WPS Web Sites
- with apps from WL
- with hybrid content from WPS through WL

Architectural decisions (WebSphere Portal and MobileFirst Platform)

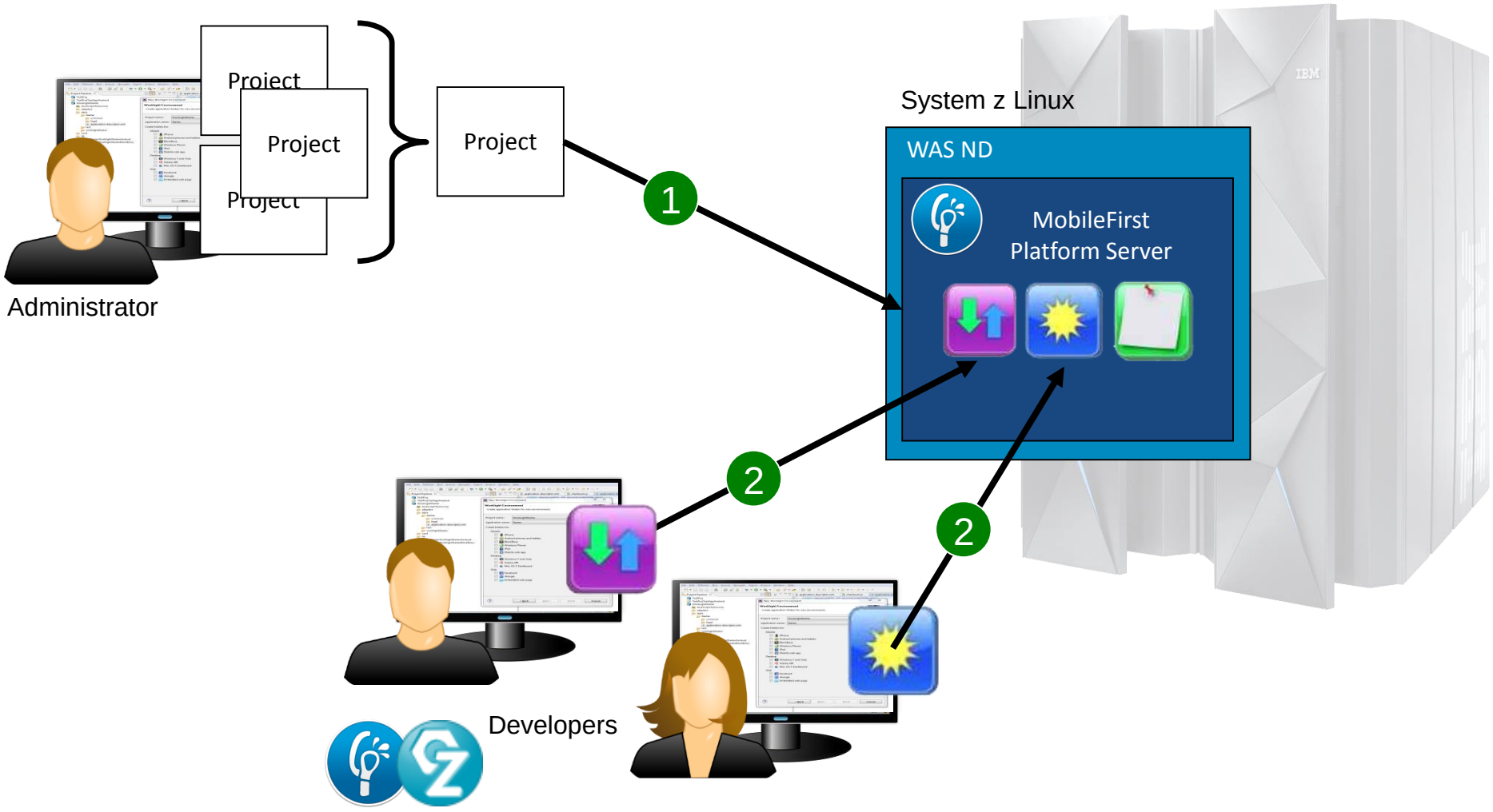
Architectural Decision	Rationale and decision points for WAS Portal or MobileFirst Platform
What to use for Web pages and Portlets?	WAS Portal or MobileFirst Platform. WAS Portal is suited to handle Web pages based on screen size of mobile device. MobileFirst Platform is an alternative with the same capability enabling the app for device specific fucntions.
What to use if the mobile App is accessing a transactional System in a web page ?	WAS Portal or MobileFirst Platform. WAS Portal can handle end-to-end transactional actions with results in a web prowser on a mobile device. MobileFirst Platform is an alternative that has even more capabilities.
How is a mobile app working compared with web pages?	A hybrid App for Mobile device which includes device functions such as Camera or GPS and touch control or voice control, requires a MobileFirst Platform environment. Web pages from WAS Portal can be used and enriched via MobileFirst Platform with Mobile device functionality.
How is transaction security implemented?	Portal and MobileFirst Platform have transactional capabilities with end-to-end security.
How can communication between mobile devices and System z be realized with mobile devices.	MobileFirst Platform and Portal can communicate via JSON MQTT and HTTP(s) with a mobile device.
How can the distribution of your mobile applications be controlled?	MobileFirst Platform has build-in, PUSH Notification‘ functions for mobile Apps and can host the APp Store (Application Center) for an enterprise.
How can the mobile app be disabled to run against a mobile middle tier?	MobileFirst Platform has capabilities to force a renewal of an app and disable the old one and keep track of different Versions and different Mobile Platforms and device charcteristics.

Architecture for MobileFirst Platform Server in Production

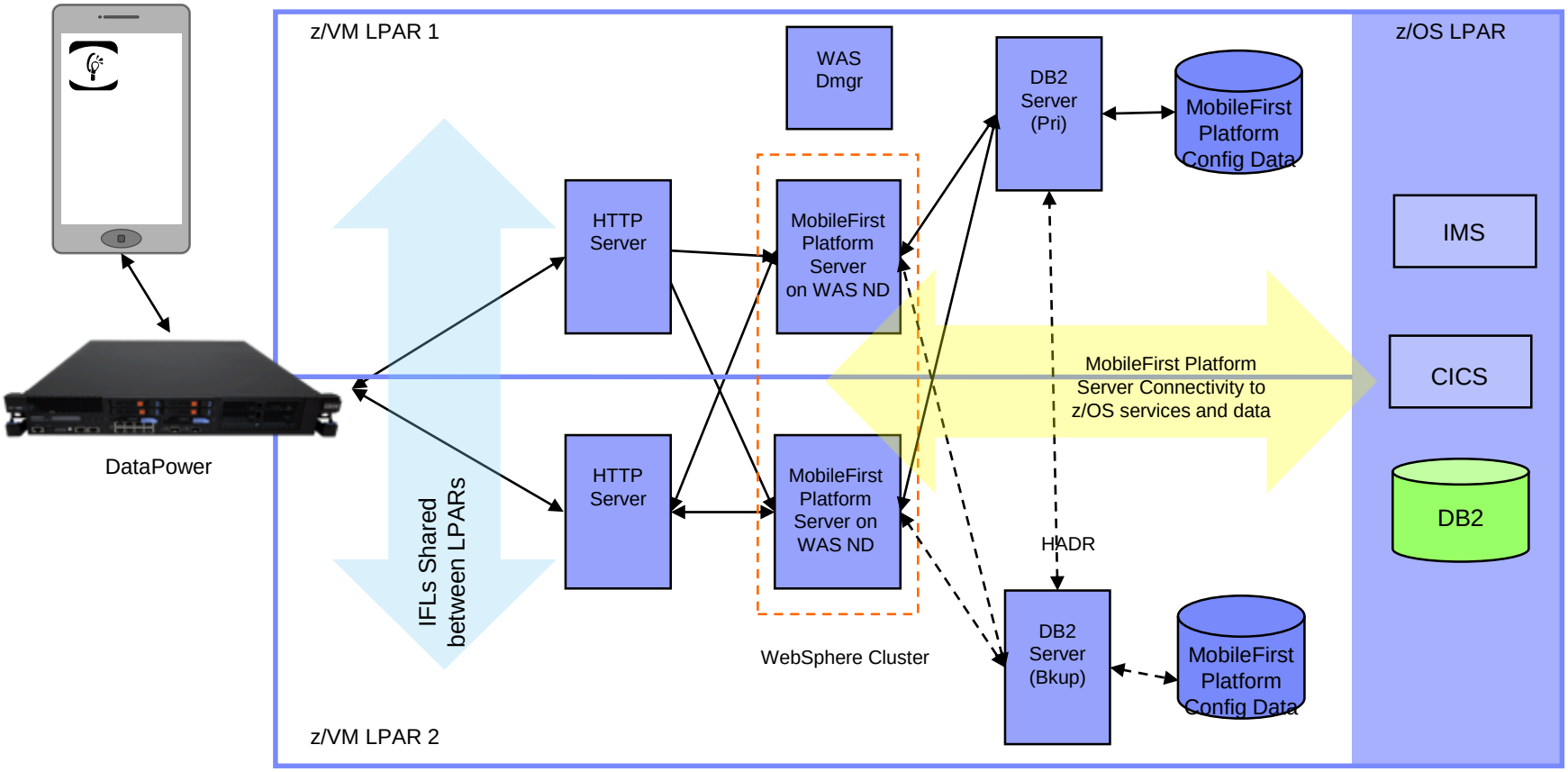
Owner: Steve Wehr



MobileFirst Platform Server Topology for Production



MobileFirst Platform Server on System z – Production High Availability



Solid Lines denote **primary** data path, dashed lines denote **backup** data path.

Flow

1. **Communications within an LPAR.** All communications within each z/VM LPAR are done via a z/VM Virtual Switch (vswitch). One vswitch with two vswitch controllers (VM userids) is used in each LPAR. Each vswitch uses two sets of OSA ports, preferably on two separate OSA features. Should one vswitch controller or OSA feature fail, communications fail over to the other.
2. **Load Balancer.** This is typically a DataPower or ISAM acting as a reverse proxy. DataPower XI50z is preferred since it can terminate SSL transactions.
3. **HTTP Server.** The Load Balancer sprays requests between the two HTTP servers. Should one of the HTTP servers fail, the Load Balancer detects this and will not route requests to it. The HTTP server serves static pages. It also routes WebSphere requests via the WebSphere plugin to the two WebSphere servers in the cluster. Should one of the WebSphere servers fail, the plugin will detect this and not route requests to it.
4. **WebSphere.** A single MobileFirst Platform server is typically created in each WebSphere Application Server cluster consisting of two nodes. Multiple MobileFirst Platform apps are deployed into the MobileFirst Platform servers on each WebSphere node of the cluster.

Flow

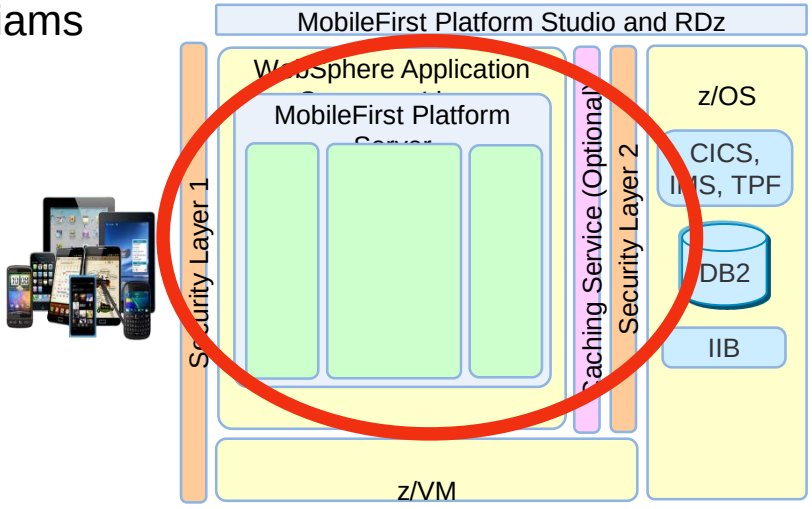
5. **DB2 Client (JDBC).** WebSphere runs the application and sends DB2 data requests to the Primary DB2 Server. Optionally, with DB2 9.7, use the RoS (Read on Standby) feature to direct read requests from reporting applications to the backup (standby) DB2 server.
6. **The DB2 HADR** (High Availability Disaster Recovery) feature is used to provide high availability for the MobileFirst Platform metadata. HADR uses two DB2 servers and two databases to mirror the data from the primary database to the standby.
    1. HADR also communicates to the DB2 clients (the JDBC driver in our case) to inform them of the address of the standby server.
    2. IBM Tivoli System Automation running on both DB2 Servers automatically detects a failure of the primary and issues commands on the standby for its DB2 to become the primary.
    3. When any communication to the primary DB2 server fails, the clients automatically route requests to the standby server (in-flight transactions are rolled back and the application can then continue from where it left off).

Architectural decisions

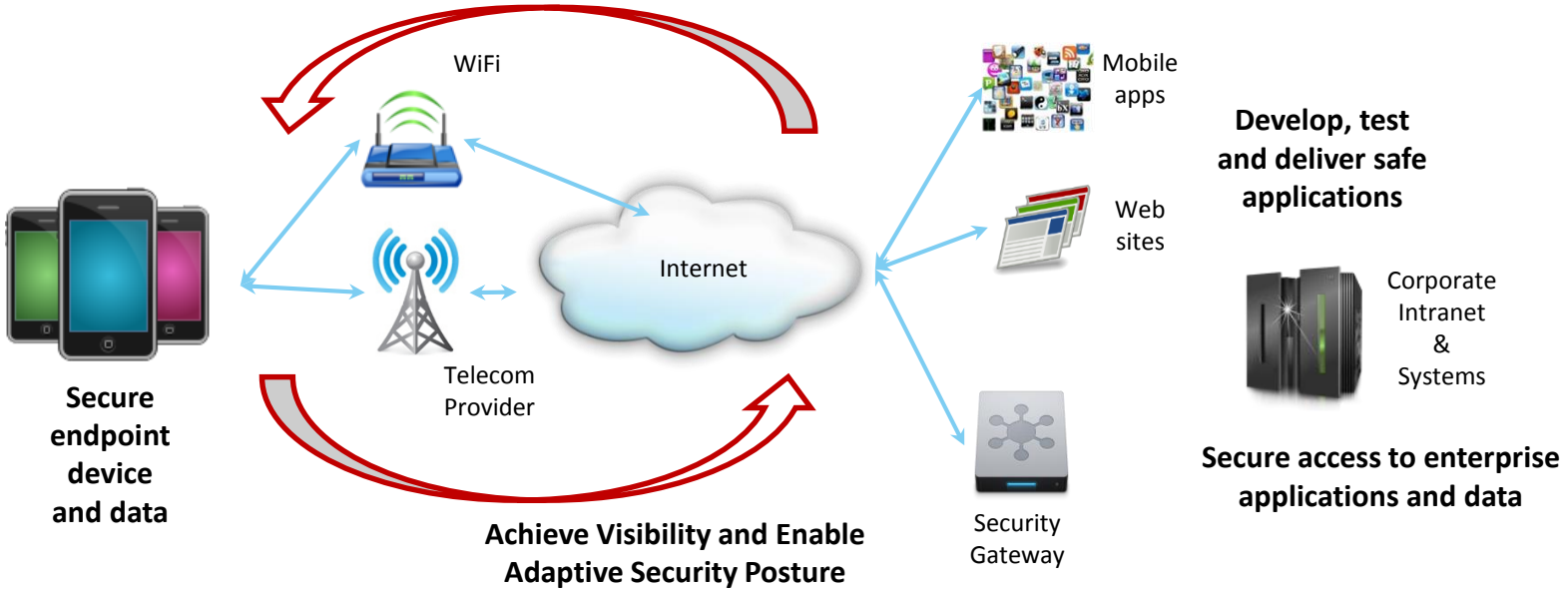
Architectural Decision	Rationale and decision points
Why run MobileFirst Platform Server in z/Linux?	The same rationale that we have been using for a decade to place web apps on z/Linux also applies to MobileFirst Platform on z/Linux: ▪Co-location with data on z/OS. Hipersockets provides the lowest-latency communication between WL and z/OS SoR. ▪Security and crypto integration The co-location argument is still the strongest. So we recommend that MobileFirst Platform Server should be run in z/Linux **only for data-rich applications that will heavily leverage data and transactions from z/OS**.
How many MobileFirst Platform servers are needed in production?	Workight Server is a light-weight application, and all the tests we have seen so far show that it scales very well, with a small CPU and memory footprint. But for high availability and scalability, one server is not sufficient. Two MobileFirst Platform servers are a good starting point for all but the largest mobile applications.
Why run the two MobileFirst Platform servers in separate LPARs?	By splitting the MobileFirst Platform topology across two separate LPARs, we remove z/VM and LPAR as single points of failure. This allows z/VM updates and LPAR hardware upgrades to be done without affecting the availability of the mobile application. When the two LPARs are on the same z machine, IFLs can then be shared between the two LPARs.
Why use WAS ND?	This is a tough one. Since the Liberty profile v8.5.5 supports clustering, it is now considered production ready for HA configurations. So either WAS ND or Liberty could be used for MobileFirst Platform production topologies.

Architecture for Security

Owner: Nigel Williams



Mobile security challenges



- Adapting to the Bring Your Own Device (BYOD) to Work Trend
  - Device Management & Security
  - Application management
- Achieving Data Separation
  - Privacy
  - Corporate Data protection

- Providing secure access to enterprise applications & data
  - Secure connectivity
  - Identity, Access & Authorization
- Developing Secure Mobile Apps
  - Vulnerability testing
- Designing an Adaptive Security Posture
  - Policy Management
  - Security Intelligence

Major considerations for System z

- As a result of the increased mobile access and decreased control, security must be incorporated into the mobile application itself, the mobile application infrastructure, and the traditional network and server security infrastructure (including System z infrastructure, Comms server, RACF, Crypto, Subsystems …)

- System z will play an important role in meeting **some** of the mobile security challenges shown on the previous chart (but not all)

- Initial focus of our customers will be on securing access to System z applications and data
  - How to protect mainframe applications from unathorized mobile users, threats and malicious attacks
  - How to authenticate mobile users and mobile devices
  - How to protect against access from unauthorized mobile applications and devices
  - How to control access to application based upon the mobile user context e.g geo-location
  - How to audit the mobile user access?

- System z may also have other roles in the overall security architecture e.g security policy management, certificate and key management

- Mobile security may be improved by deploying part of the mobile infrastructure on System z

Steps to consider when securing the mobile enterprise

IBM Security Framework domains

- Security Intelligence, Analytics & GRC
- People
- Data
- Applications
- Infrastructure

Mobile Security Strategy and Lifecycle Management

At the Device

Enroll
Register owner and services

Configure
Set appropriate security policies

Monitor
Ensure device compliance

Reconfigure
Add new policies over-the-air

De-provision
Remove services and wipe

Over the Network & Enterprise

Authenticate
Properly identify mobile users

Encrypt
Secure network connectivity

Monitor
Log network access and events

Control
Allow or deny access to apps

Block
Identify and stop mobile threats

For the Mobile App

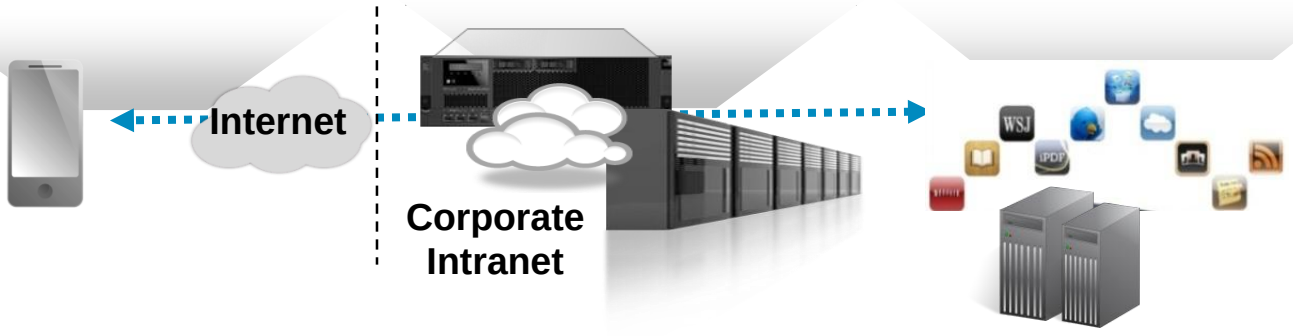
Develop
Utilize secure coding practices

Test
Identify application vulnerabilities

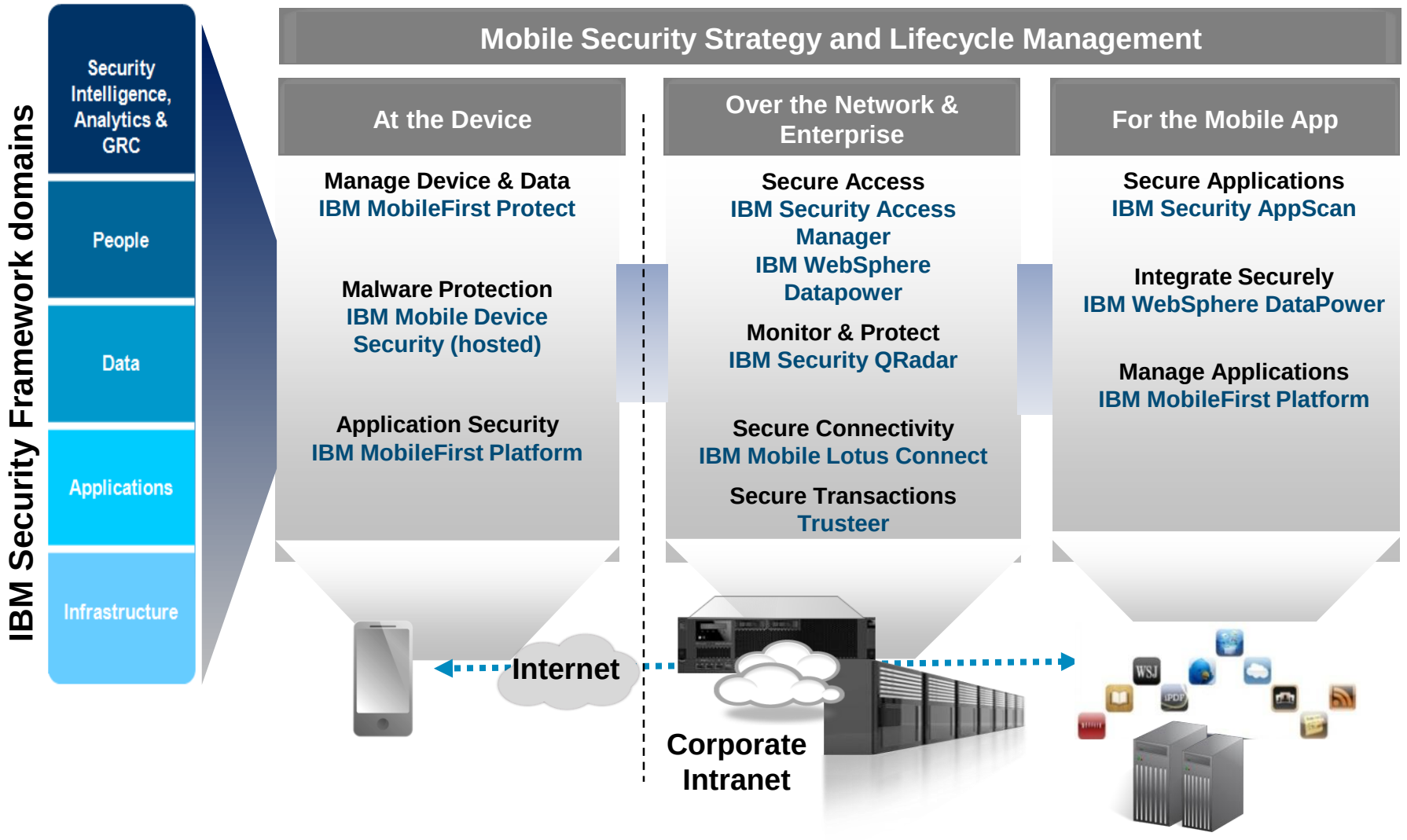
Monitor
Correlate unauthorized activity

Protect
Defend against application attacks

Update
Patch old or vulnerable apps



IBM MobileFirst offerings to secure the enterprise



Some observations and assumptions

- It is possible that in certain limited implementations mobiles will connect directly to System z services
  - Security in this case will be based on transport security i.e SSL/TLS and basic authentication

- However in most cases a Mobile Enterprise Application Platform (MEAP) such as MobileFirst Platform will be present so the security features of the MEAP can be used

- And for high volume or internet-based mobile applications a **Mobile Security Gateway** is recommended

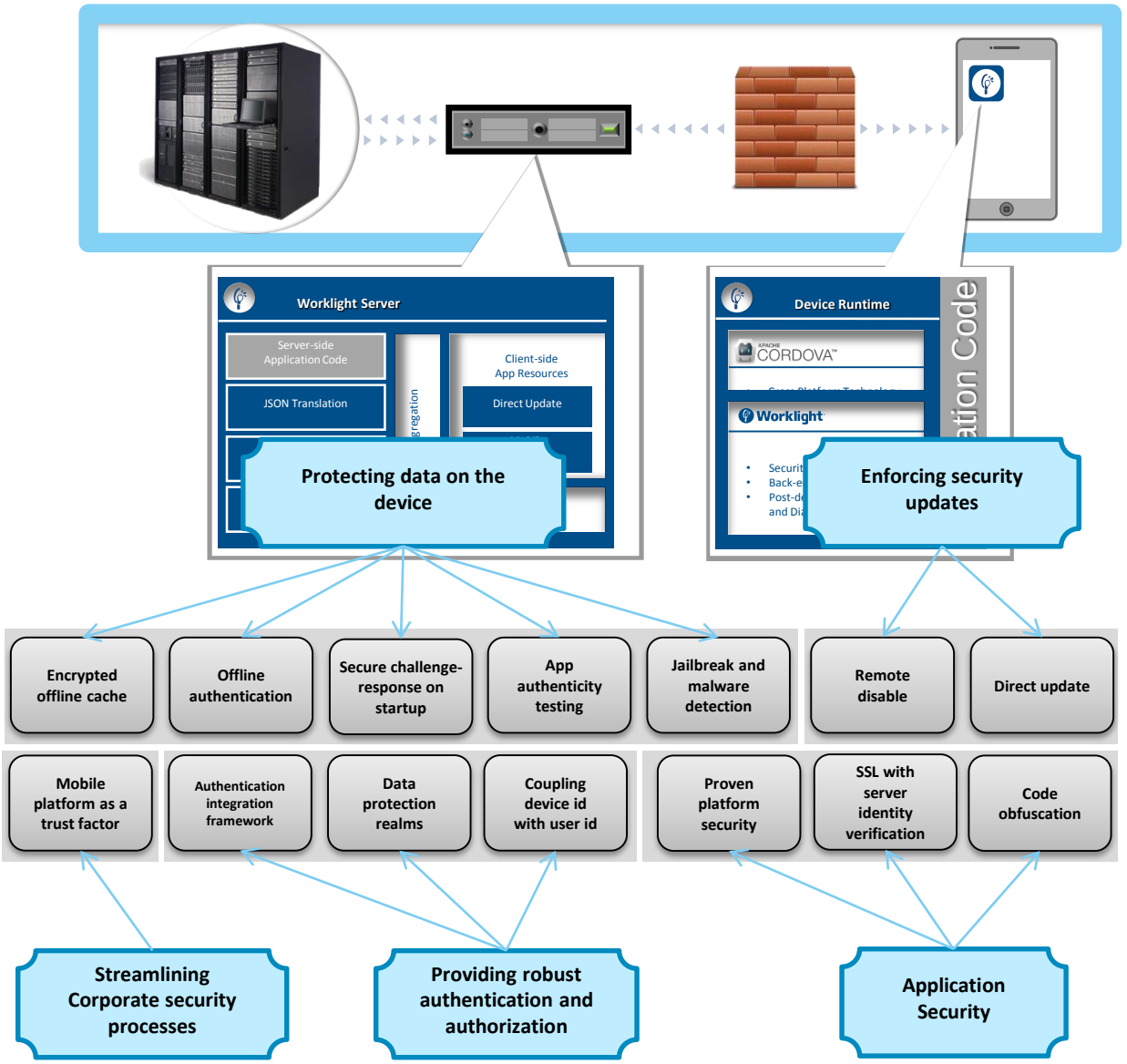
- The key products to consider for protecting mobile access to System z are
  - **MobileFirst Platform server** for MEAP security features
  - **DataPower** as a mobile security gateway
  - **IBM Security Access Manager** for risk-based access
  - **QRadar** for visibility of mobile security events

- Device management with IBM MobileFirst Protect is unlikely to be deployed on System z

- Trusteer is an important recent acquisition in the area of mobile banking fraud prevention

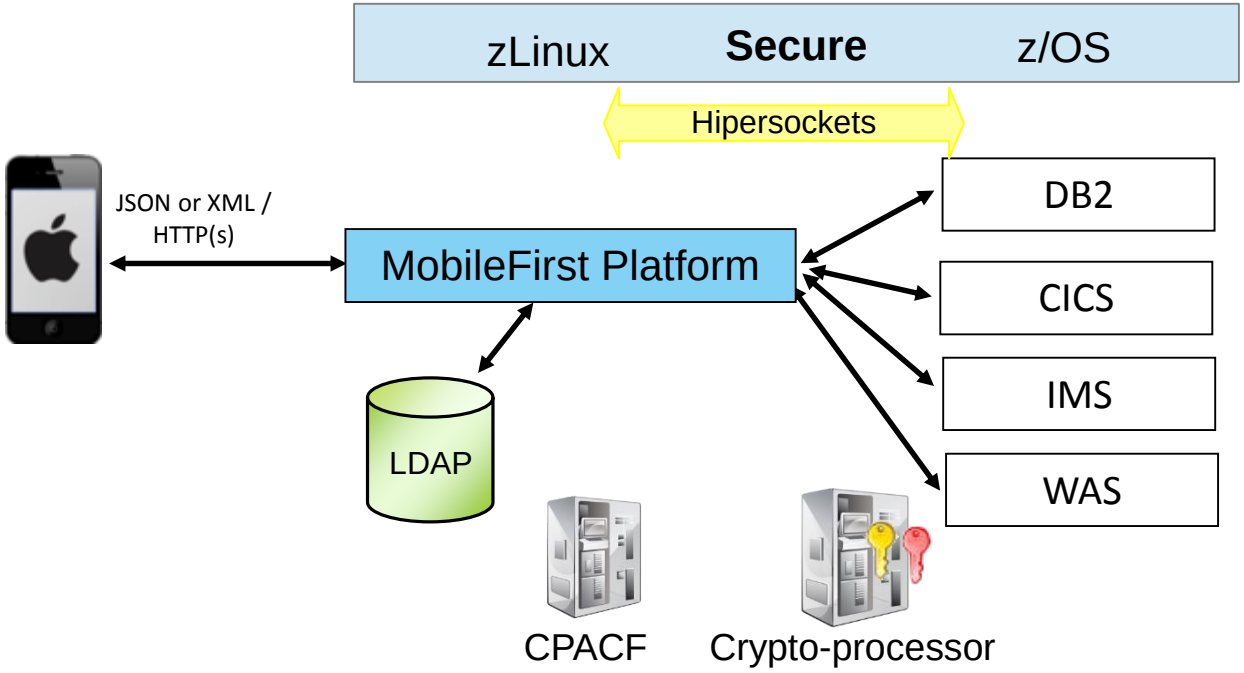
MobileFirst Platform Security Features

- Ensure that only specific applications on specific devices can connect to enterprise systems
- Extensible framework for authentication of mobile application users
- Encrypt data on the device
- Enforce security updates
- Propagate identity to enterprise systems



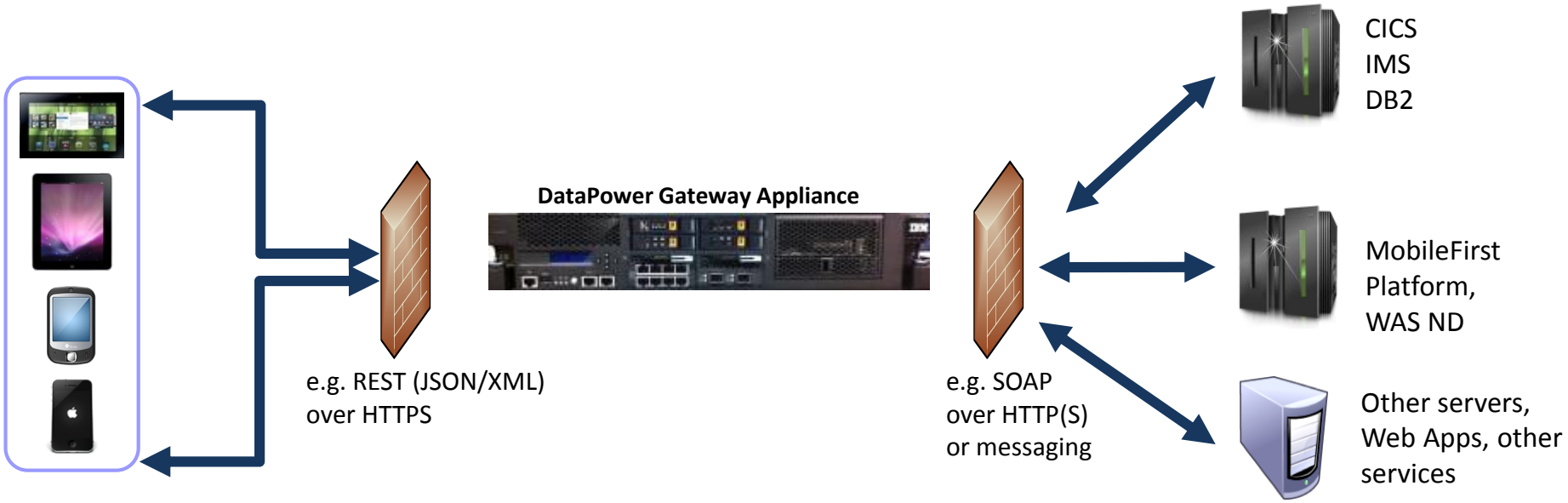
Topology 1 – MobileFirst Platform security

Capabilities	Deployment scenarios	System z benefits
• Authentication: HTTP Basic, form-based,Custom • Device authentication • Offline authentication • Application updates and authenticity • Authorization: Policy • Interoperate: LDAP, WebSphere	• Small enterprise, with minimal scalability needs • MobileFirst Platform is only MEAP • Non-DMZ • Traditional web user authentication mechanisms are sufficient • Minimal interoperability required with enterprise-wide security solutions	• Benefits of platform security and certification EAL 4+ for MobileFirst Platform • Reduce cost and improve performance by using HW crypto cards and CPACF • Security advantages of Hipersockets • Opportunity to eliminate encryption between MobileFirst Platform server and backend



DataPower Mobile Security Features

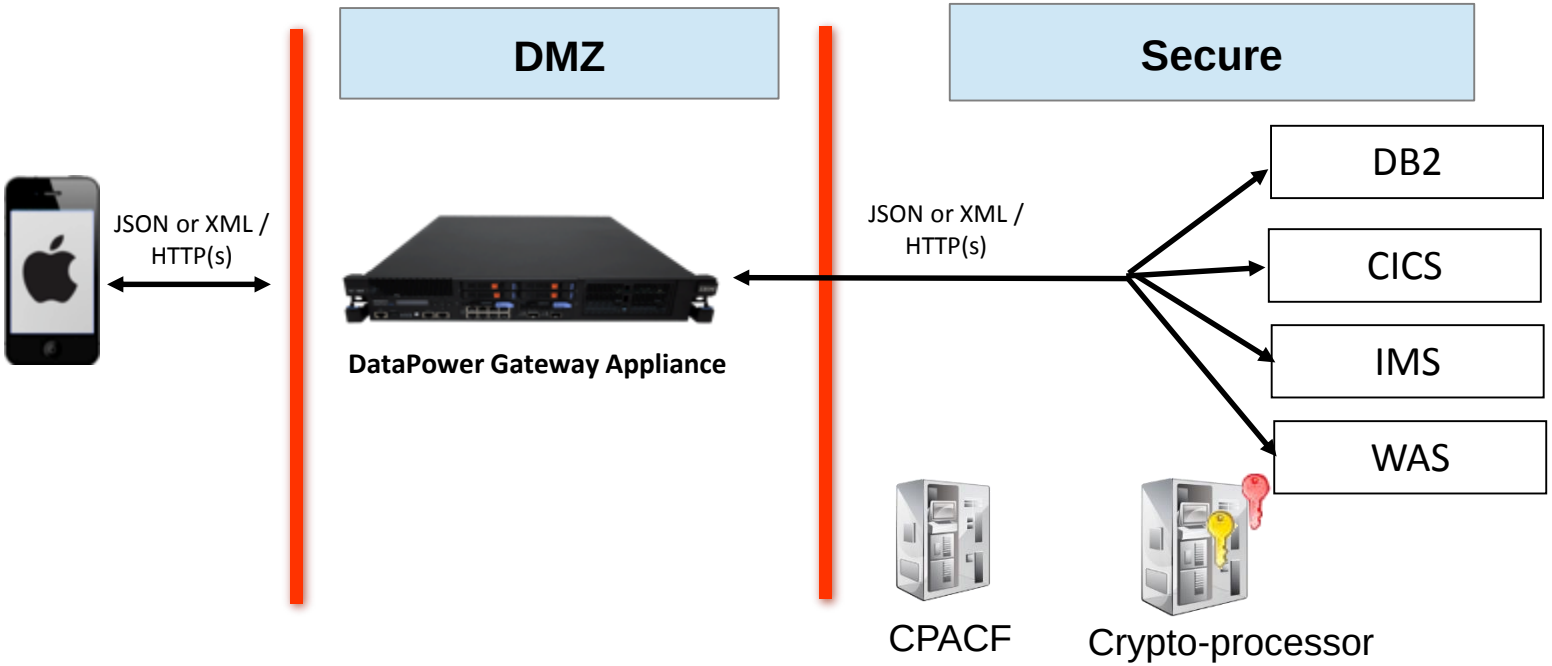
Available as a physical or virtual appliance



- Security, Control, Integration & Optimization of mobile workload
- Enforcement point for centralized security policies
- Authentication, Authorization, SAML, OAuth 2.0, Audit
- Threat protection for XML and JSON
- Message validation and filtering
- Centralized management and monitoring point
- Traffic control / Rate limiting
- Integration with MobileFirst Platform

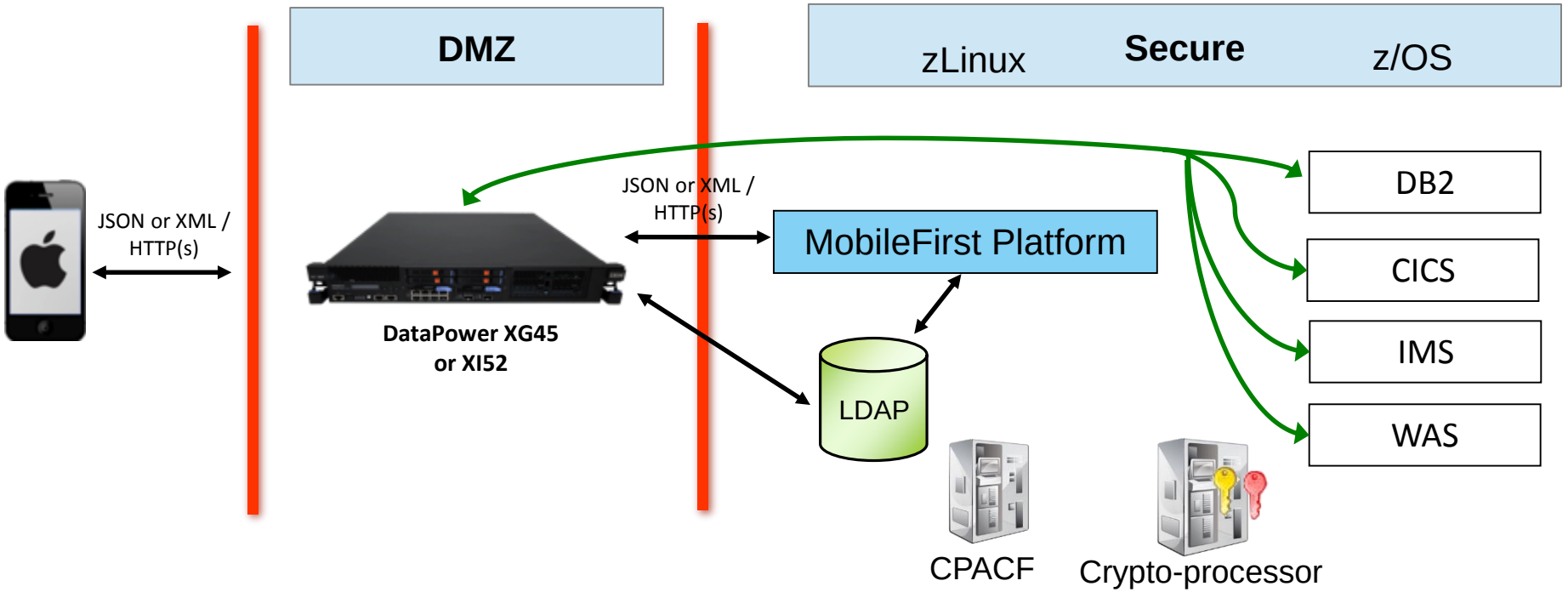
Topology 2 – DataPower security

Capabilities	Deployment scenarios	System z benefits
• Authentication: HTTP Basic, form-based, WS-*, SSL, Kerberos, SAML, LTPA, OAuth • Authorization: LDAP, ISAM, SiteMinder, SAML, XACML, OAuth, System z (RACF) • Interoperate: LDAP, SiteMinder, ISAM, TFIM, WebSphere	• When mobile apps are heavily focused on REST/API/web service based interactions • High volume or internet mobile access • DMZ or non-DMZ • Support for Web APIs	• Additional benefits of DataPower as a mobile security gateway for System z • DataPower can securely access backend services directly • Supports a wide range of authentication and authorization models • Good integrations with System z (RACF, z/OS identity propagation)



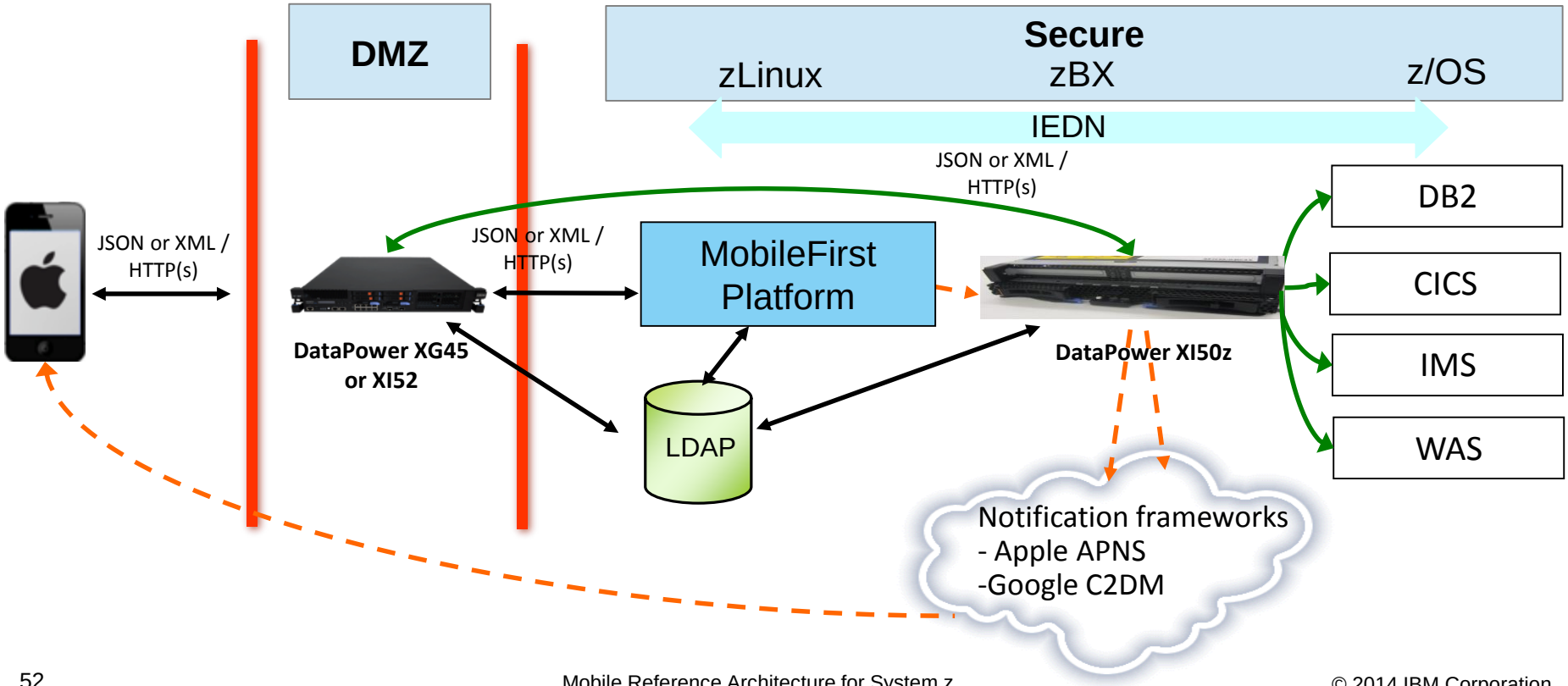
Topology 3 – DataPower as a reverse proxy for MobileFirst Platform server

Capabilities	Deployment scenarios	System z benefits
• Combined capabilities of MobileFirst Platform and DataPower	• When hybrid mobile apps use a combination of web and Restful interactions • High volume or internet mobile access	• Additional benefits of DataPower as a mobile security gateway for MobileFirst Platform on zLinux • LDAP user registry shared between DataPower and MobileFirst Platform

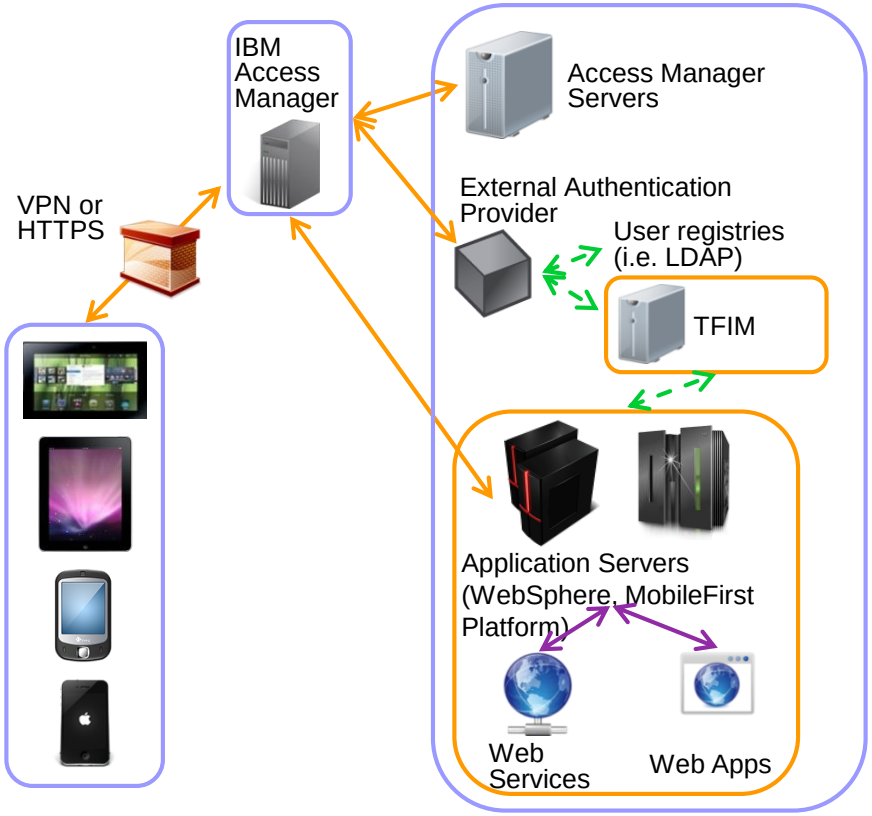


Topology 4 – DataPower XI50z as a 2nd security layer

Capabilities	Deployment scenarios	System z benefits
• DataPower XI50z (zBX blade) contains the same functionality as a stand-alone device or virtual appliance … but benefits from co-location with System z services • Defence in depth	• For offload of security processing (e.g SSL) and to perform identity mapping • Secure proxy for push notifications from MobileFirst Platform server to the mobile device	• DataPower XI50z acts as an additional security layer for backend services • IEDN provides a secure private network for communication between zLinux, zBX and z/OS



ISAM Mobile Security features

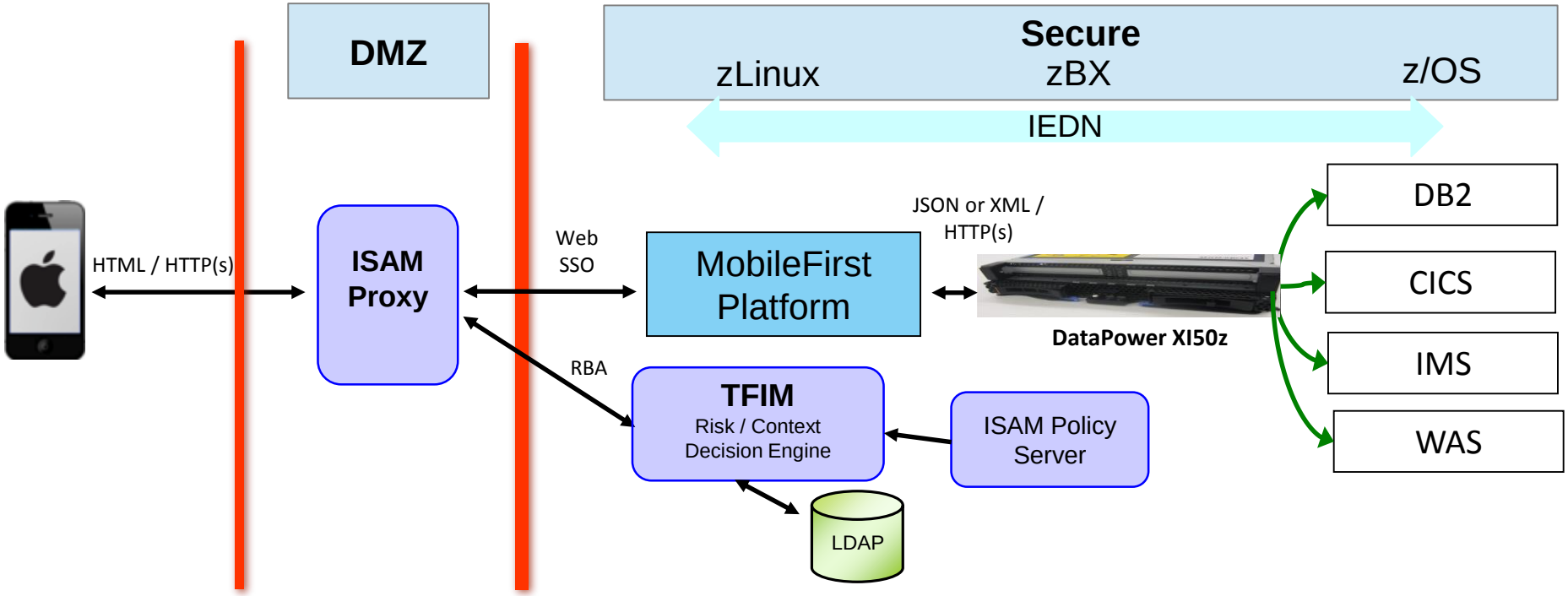


- Dynamically assess the security risk of an access request
- Quickly enforce Risk-Based Access
- Flexibility and strength in authentication: user id/password, biometrics, certificate, or custom
- Protect applications from known security threats by analyzing HTTP traffic
- Integration with MobileFirst Platform and DataPower

Tivoli Federated Identity Manager
Tivoli Security Policy Manager

Topology 5 – IBM Security Access Manager

Capabilities	Deployment scenarios	System z benefits
• Authentication: HTTP Basic, form based, SSL, Kerberos, SAML, LTPA, NTLM, OAuth, multi-factor, step-up, Risk based • Device authentication • Authorization: LDAP, ISAM, SiteMinder, SAML, XACML, OAuth, System z (RACF) • Interoperate: LDAP, SiteMinder, TFIM, .NET, WebSphere, QRadar	• Mobile apps are heavily focused on mobile web/browser interactions • DMZ or non-DMZ • Strong authentication (2FA,MFA) or risk based authentication (RBA) is required • Comprehensive SSO and session management is required	• Benefits of platform security and certification EAL 4+ for TFIM • Mobile authorization rules policies consolidated in TFIM on zLinux **Note**: DataPower and ISAM can also be used together: ISAM for web requests and DataPower for Restful service requests.



IBM Security QRadar

Deliver mobile security intelligence by monitoring data collected from other mobile security solutions – visibility, reporting and threat detection

- Integrated intelligent actionable platform for:
  - Searching
  - Filtering
  - Rule writing
  - Reporting functions
- A single user interface for:
  - Log management
  - Risk modeling
  - Vulnerability prioritization
  - Incident detection
  - Impact analysis tasks

Architectural decisions

Architectural Decision	Rationale and decision points
How to securely manage BYOD within an enterprise?	It is best practice for enterprises wishing to implement highly secure mobile environments to deploy an MDM solution (such as MobileFirst protect). Most relevant for B2E scenarios.
How to authenticate the device?	Use MobileFirst Platform or ISAM. Most relevant for B2E scenarios.
How to authenticate the mobile user?	This is the area with the widest range of choices: 1. Traditional web user authentication and authorization mechanisms (user ID/password, single sign on (SSO), secure token exchange, and SSL mutual authentication) 2. Two-factor authentication combining a password authentication with a second factor, which may be token or certificate-based authentication, or a one-time password (OTP). 3. Risk based authentication (RBA) is often used in banking and highly secure and sensitive applications MobileFirst Platform, DataPower or ISAM can authenicate mobile users.
How to authenticate the mobile application?	Use MobileFirst Platform to verify the authenticity of the application.

For a full set of mobile security architectural decisions refer to the ISSW Mobile Reference architecture. The questions covered in this presentation are those which are most closely related to System z and mobile security. Link to the ISSW Mobile Reference Architecture

Architectural decisions

Architectural Decision	Rationale and decision points
How to authorise mobile requests to System z applications and resources?	End to end security solutions may require that the mobile user's identity (and potentially other security context) flows with the request message as it passes through the different layers of the application architecture and until it arrives in the backend System z server. This is very difficult to achieve and .
How to manage mobile single sign-on for hybrid mobile apps	The OAuth (Open authorization) allows a resource owner to grant permission for access to their resources without the sharing of credentials, and to provide limited access to resources hosted by web-based services accessed over HTTP. It is more often used in social media rather than in OLTP although it is being considered by some banks.
Can OAuth be used with System z?	OAuth is not widely supported on System z. However DataPower can authenticate/authorize using OAuth and then map to another token type understood by the backend system z services.
How to transport identity and mobile security context in request messages?	SAML (Security Assertion Markup Language) is too heavy to be used on the mobile device, however it is a standard for transporting identity and other user attributes between the different layers of the application architecture.
Can SAML be used with System z?	SAML is implemented in WAS z/OS and in also in CICS with the new Feature Pack for Security Extensions. DataPower has wide support for SAML.

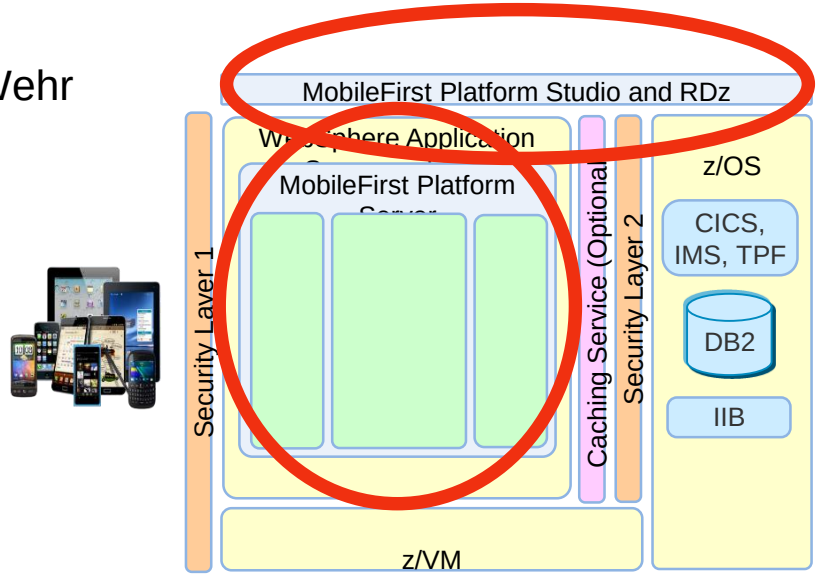
Architectural decisions

Architectural Decision	Rationale and decision points
How to secure sensitive data sent by mobiles?	Use SSL and VPN technologies. SSL is more granular in securing specific message exchanges, where VPN secures all communication to enterprise services from the Mobile application. VPN technology is typically used to access internal services from public networks. SSL communication is commonly used to access services through an enterprise DMZ.
How to optimize performance of SSL/TLS on System z?	Use hardware crypto. Follow the specific best practice for each subsystem (MobileFirst Platform, WAS, CICS etc). Consider the use of DataPower XI50z to offload SSL processing.
How to manage certificates used in SSL/TLS communication?	Consider using z/OS PKI services for creation and management of X.509 certificates.
How to protect System z from unauthorized mobile access and malicious attacks?	Use DataPower as a secure mobile gateway. Consider also the use of intrusion detection systems (IDS) and other traditional network security e.g firewalls.
How to monitor mobile secure access to System z?	Use QRadar to provide a comprehensive solution to detect malicious behavior from mobile applications.

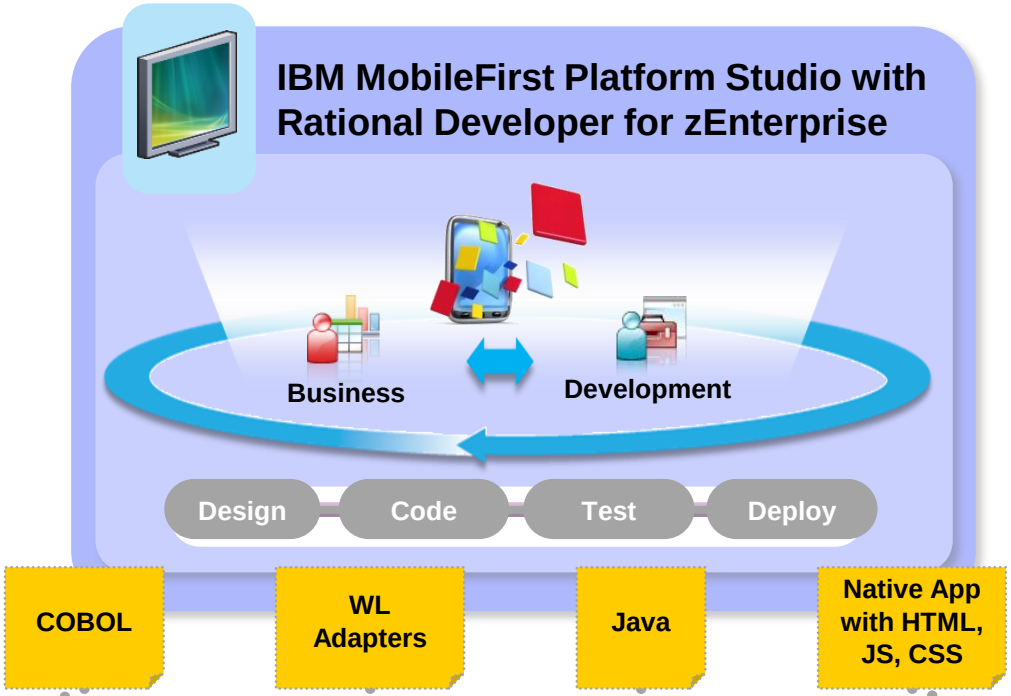
More information and use cases will be covered in the ‘System z Mobile Security Guide’ which is planned for completion in December 2013.

Architecture for MobileFirst Platform Server in Development and Test

Owner: Steve Wehr

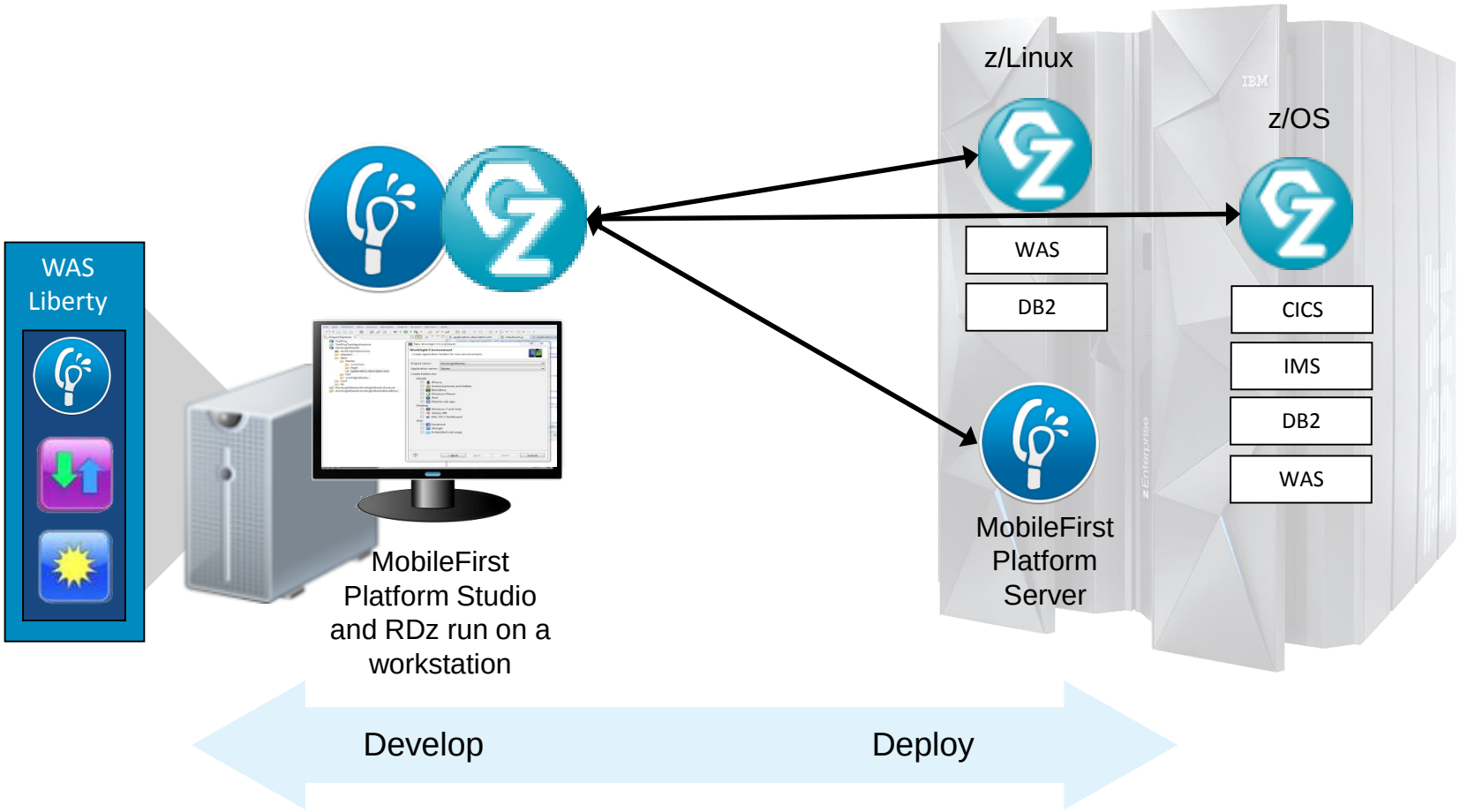


Development for Mobile Devices for IBM MobileFirst Platform on System z

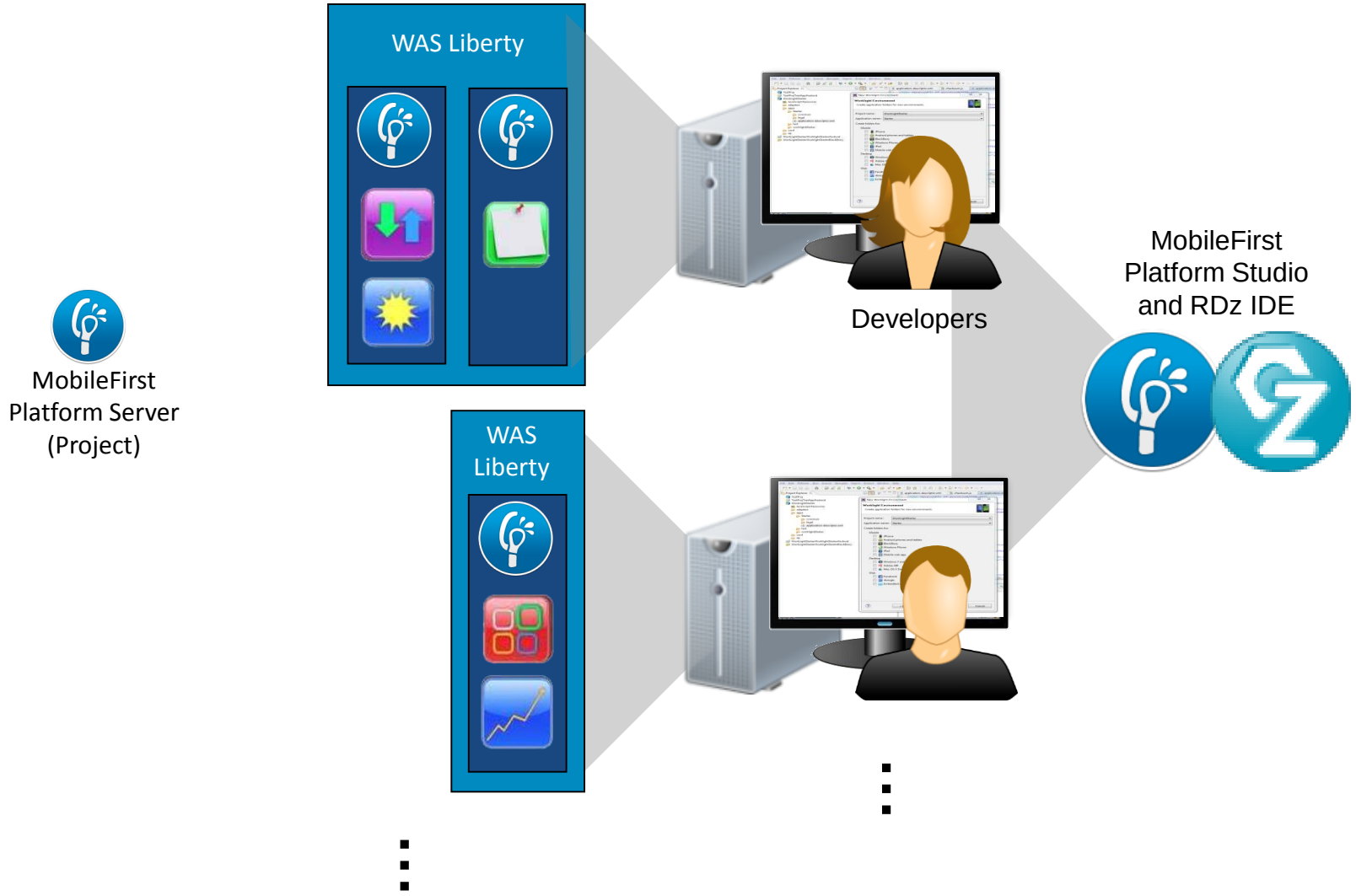


- Built on Eclipse
- Common code base across all mobile platforms (with ability to override at platform level)
- Build, preview, and deploy within the IDE
- Mobile simulator (for unit test)
- End-to-end debug
- Integrate with third-party SDKs (e.g. Android Development Tools)

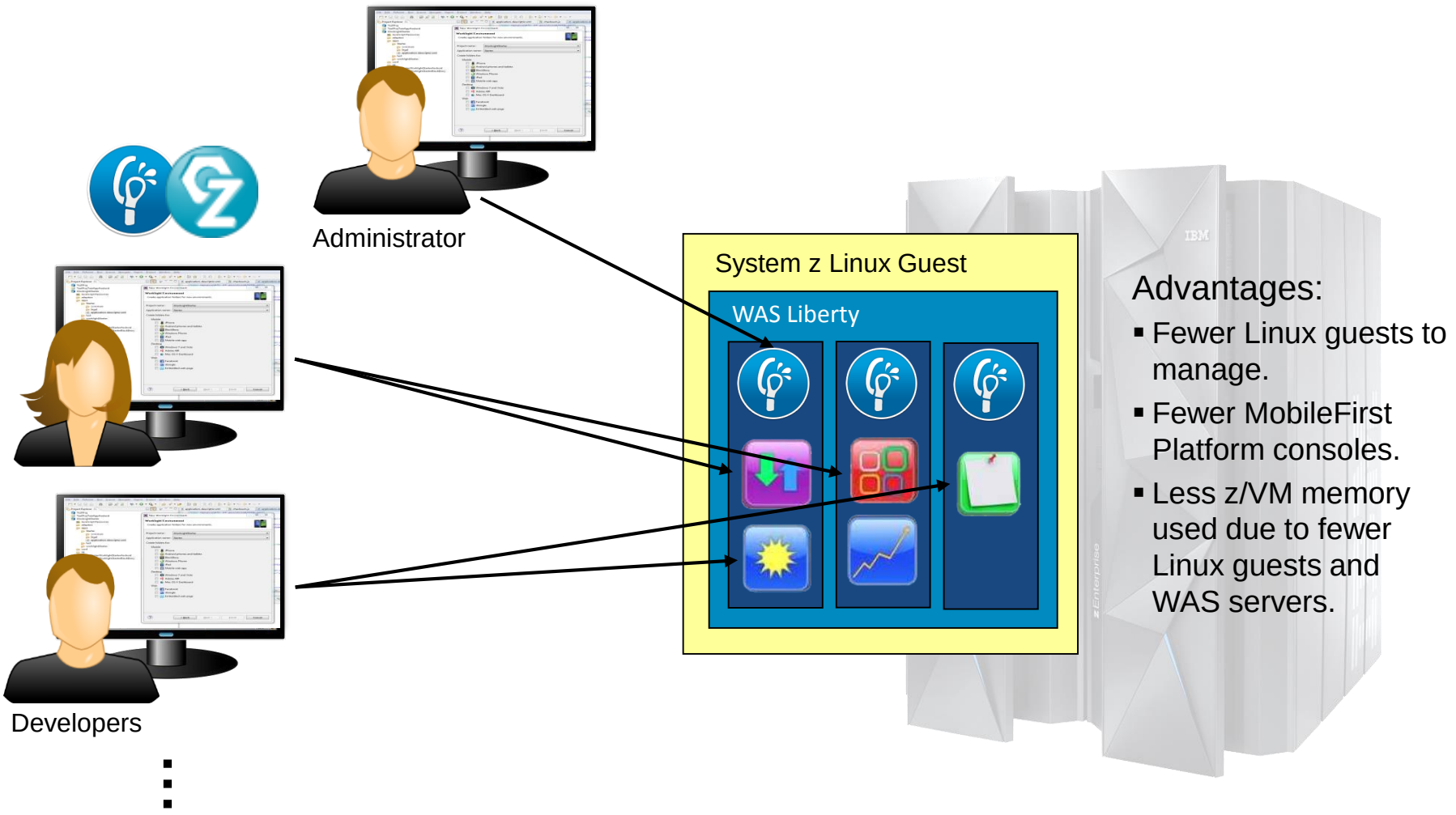
MobileFirst Platform Studio integrates with Rational Developer for z



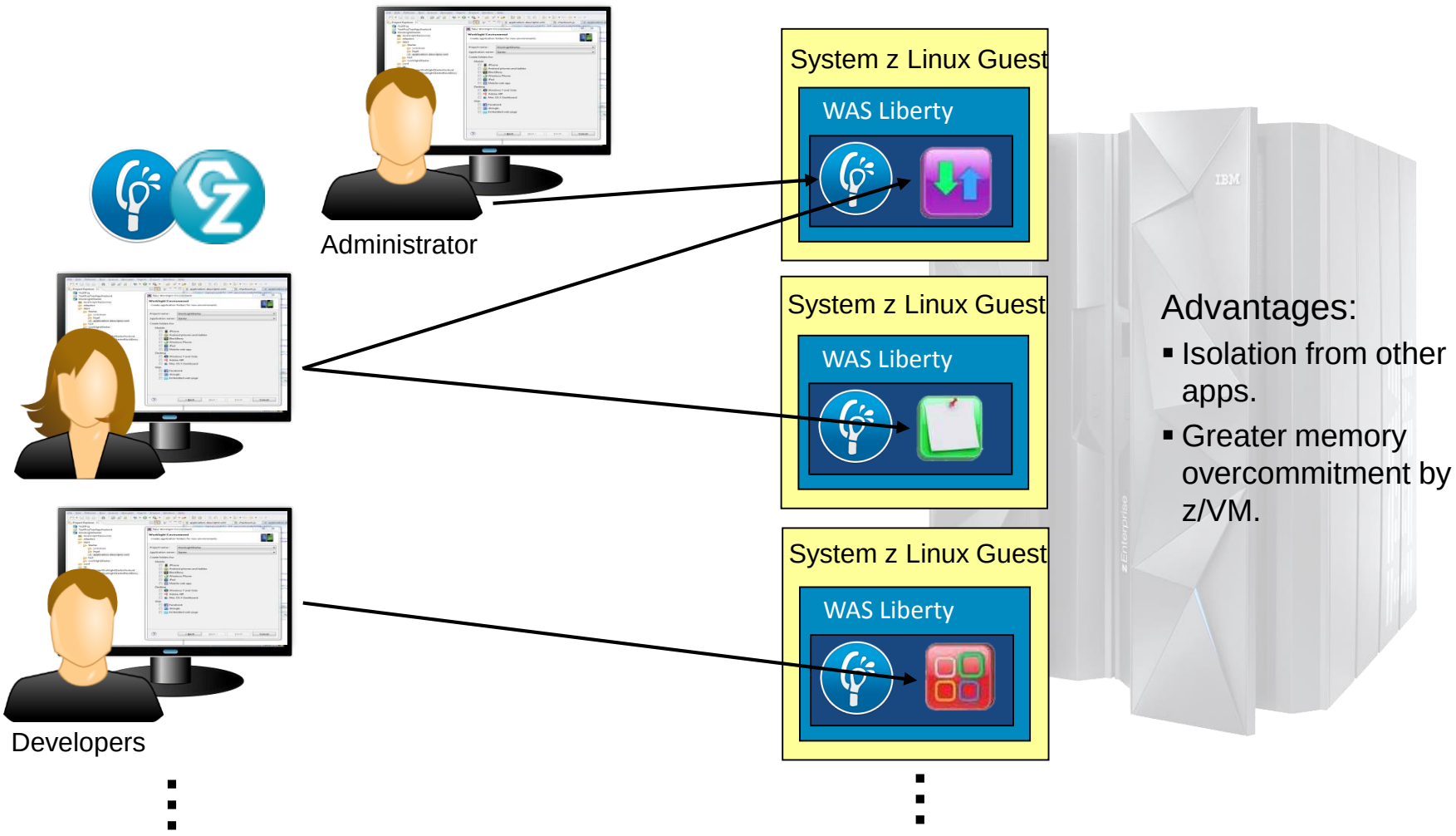
MobileFirst Platform Server Topology for Development



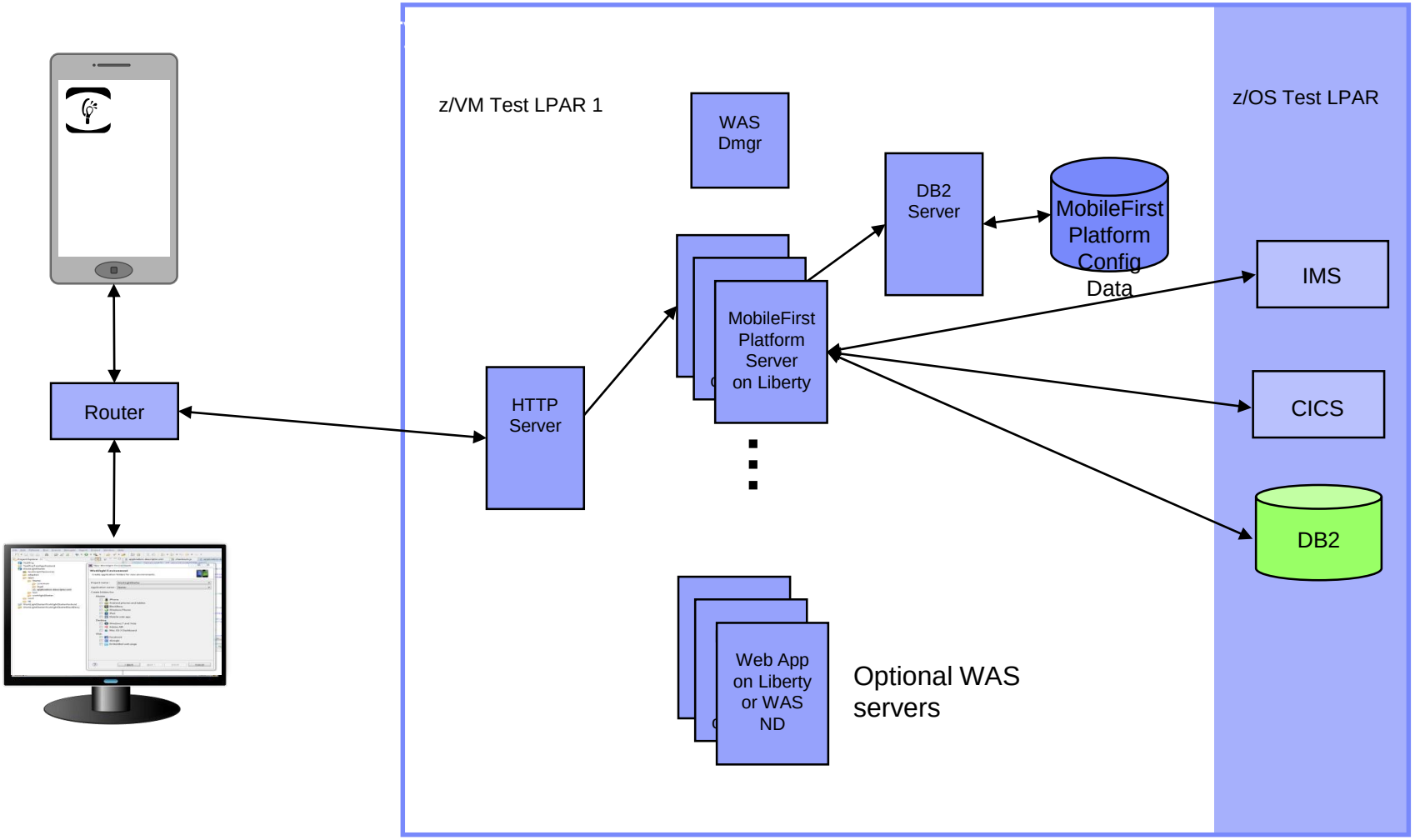
MobileFirst Platform Server Topology for Testing
Topology 1 – Dense Deployment



MobileFirst Platform Server Topology for Testing
Topology 2 – Spread Deployment



MobileFirst Platform Server – Guest Topology for Testing



Components

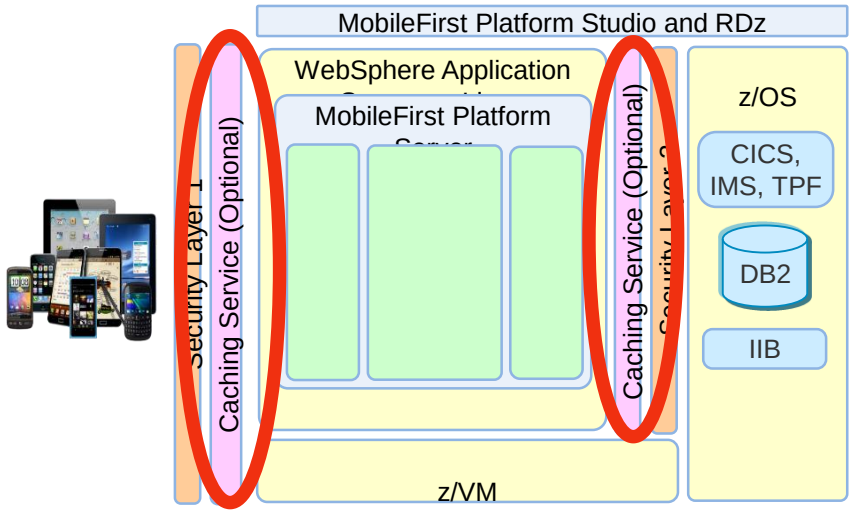
1. **MobileFirst Platform Server.** Each WAS profile can host multiple MobileFirst Platform projects, which can each host multiple apps. Chose either the dense or spread deployment depending on how you prefer to manage applications and guests. We recommend running MobileFirst Platform on the WAS Liberty profile. You get this free with MobileFirst Platform, and it can be configured to use only those services needed by MobileFirst Platform, thus saving memory, CPU and startup time.
2. **WebSphere Application Server (optional).** Separate WAS servers may be needed if the MobileFirst Platform apps use WebSphere java applications that the customer wants to host in separate JVMs. This may be because
   - The WebSphere apps already exist and the new mobile app is using an API they publish.
   - The customer decides to split the business logic of the mobile application into some components outside of MobileFirst Platform.

Architectural decisions

Architectural Decision	Rationale and decision points
How many LPARs for Linux on z?	In Dev/Test, all the MobileFirst Platform servers can be run in a single LPAR because there is no need to replicate the HA environment used for production.
Why use WAS Liberty?	WAS Liberty servers are recommended to host the MobileFirst Platform server because they can be configured to have a smaller memory footprint that WAS ND, start faster, and use less CPU when idle. All of these are key performance features for a development server that will be started and stopped often. The WAS ND clustering features are not needed in development.
How many LPARs for z/OS?	The MobileFirst Platform applications should be using test versions of z/OS services and so should only be interacting with the test LPAR(s) for those services.

Caching Considerations

Owner: Frank van der Wal



Considerations on caching from middle tier to back-end

- Mobile devices have become a disruptor to the technology industry. In 2013, the number of mobile devices accessing the internet are expected to exceed the number of desktop machines.
- Data is being accessed at a rapidly increasing rate to service mobile applications and the workload is becoming a burden to back-end servers. System z customers, in particular, are seeing their costs increase in z/OS as mobile transactions drive up query workloads.
- The ability to quickly react and adapt to this increased workload for application servers has increased the importance of caching at various levels in the network topology.
- Platforms for hosting mobile applications most have the ability to be agile to handle this increased workload
- Caching can realize increased response times and support larger numbers of concurrent client devices.
- Application logic needs to be adapted (depending on caching mechanisms used)

Caching in Mobile Enterprise scenarios

As caching is a common practice, Mobile workload can put a different angle on caching. Modern enterprise can benefit from caching into its IT systems in five important areas:

- Cost saving
- Scalability
  - Growth in Mobile transactions and data volumes
  - Processor load and memory consumption on back-end systems
  - Constant need for scale in and scale out front-end systems
- Availability
  - Composite applications (aggregation from various sources)
  - Session persistence
- Failover
- Flexibility

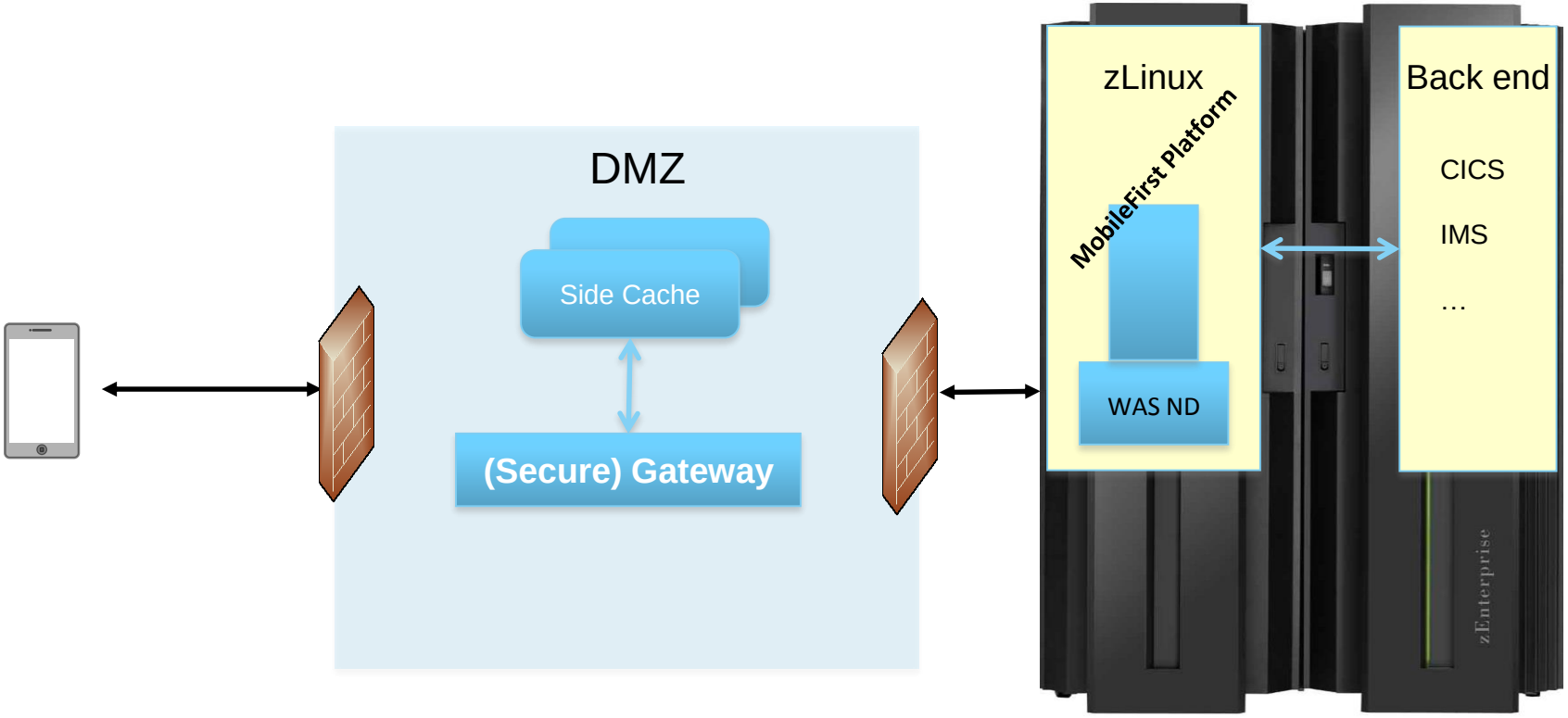
Intro to Caching

Caching pattern	characteristics
Read-only	Static data. Improve performance and scalability. Data owned by database
Read/write	Volatile data. Applications can read, add, modify or delete data. Cache plays master role
Read-through	As read/write but back-end system plays master role
Write-through	Changes in cache are simultaneous written in back-end. Slow but data consistency is ensured
Write-behind	Changes in cache are propagated asynchronously to back-end.

Caching scenario	characteristics
Side cache for applications. This is the cache that we recommend in this section.	An application first attempts to get the data from the side cache. If it finds it there, it uses that data. If the data does not exist in the cache, the application reads the data from back-end and stores a copy of it in the cache.
Side cache for Enterprise Service Bus (ESB)	Cache resides on the ESB where multiple SOA equipped applications can make use of like the side bus scenario
Cache as integration point	Cache between (multiple) applications

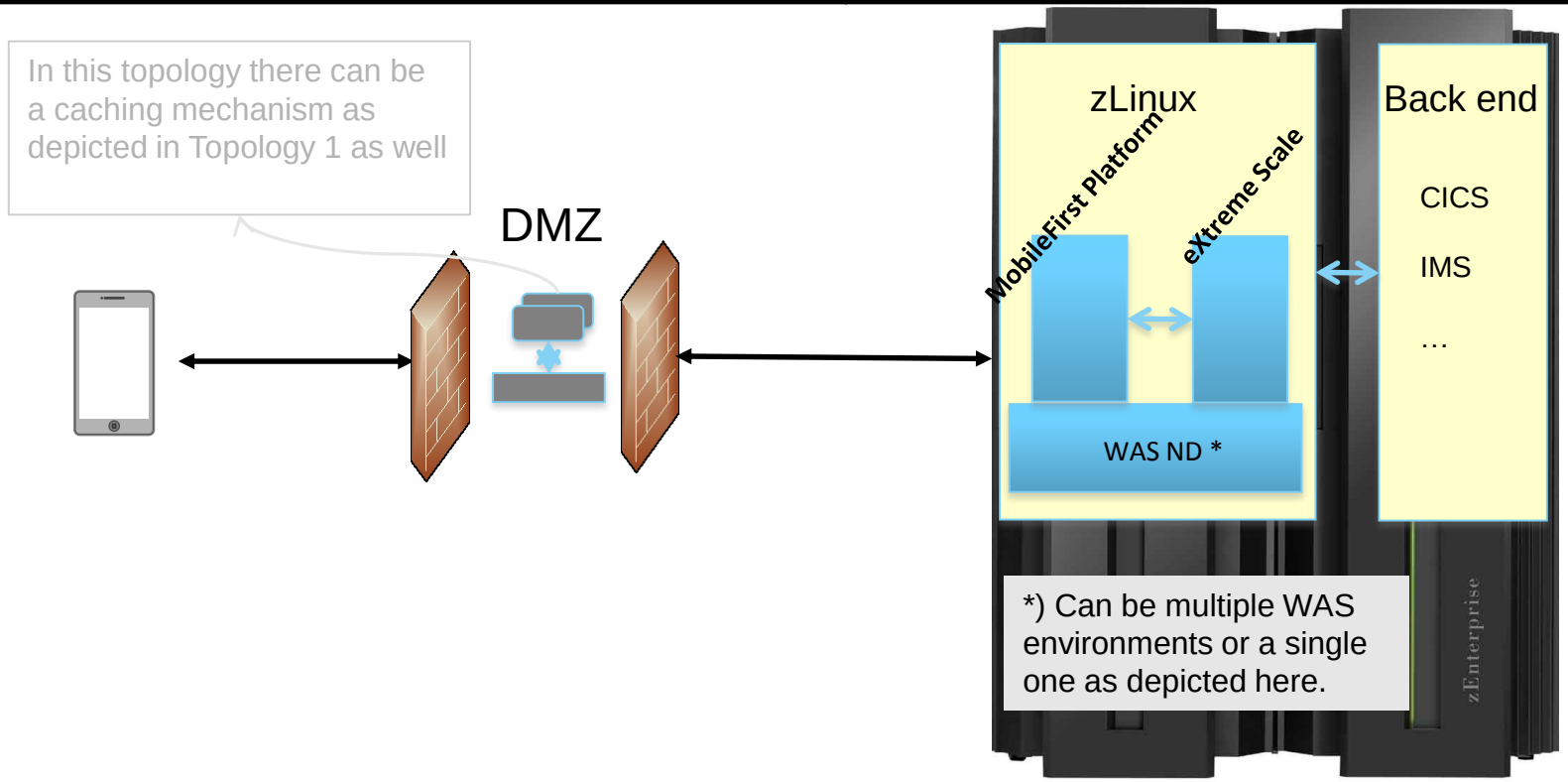
Topology 1 – Caching front-end requests

Capabilities	Deployment scenarios
• Keep data as close as possible to the mobile device • Caching of static data • Cache hit ratio is high on a large dataset for all requests can be cached here	• Cache static information like HTML, Images, User profiles etc • E-commerce scenarios, retail with much static information • Response time sensitive user scenarios (on-line shopping)



Topology 2 – Caching back-end requests

Capabilities	Deployment scenarios
• Cache data that otherwise has to be pulled from back end system driving up CPU load • Compared to Topology 1, limited data will be cached • Enterprise data will be cached • When using WebSphere eXtreme scale, mobile app can be enriched on code level. App code has to be altered.	• Cache dynamic data from back end (System z, CICS, IMS) • In the situation where queries and not updates are performed



Two IBM caching products

IBM WebSphere DataPower XC Caching Appliance vs IBM WebSphere eXtreme Scale

DataPower XC10 Appliance

- Drop-in cache solution optimized and hardened for data-oriented scenarios
- High density, low footprint improves datacenter efficiency

Data-Oriented

Session management
Elastic DynaCache
Web-side cache
Petabyte analytics
Data buffer
Event Processing
Worldwide cache
In-memory OLTP
In-memory SOA

Application-Oriented

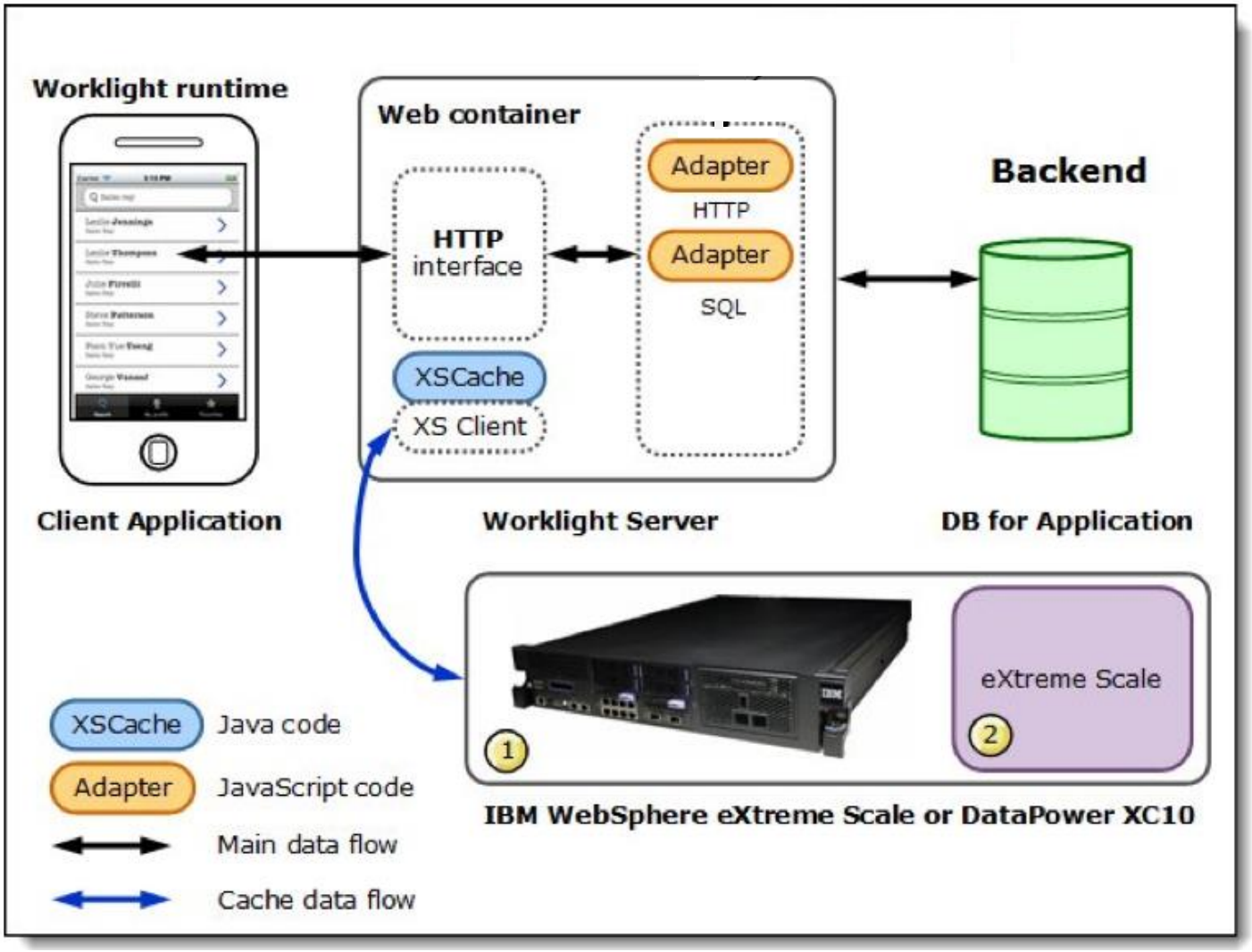
eXtreme Scale

- Ultimate flexibility across a broad range of caching scenarios
- In-memory capabilities for application-oriented scenarios

Elastic caching for linear scalability
High availability data replication
Simplified management, monitoring, and administration

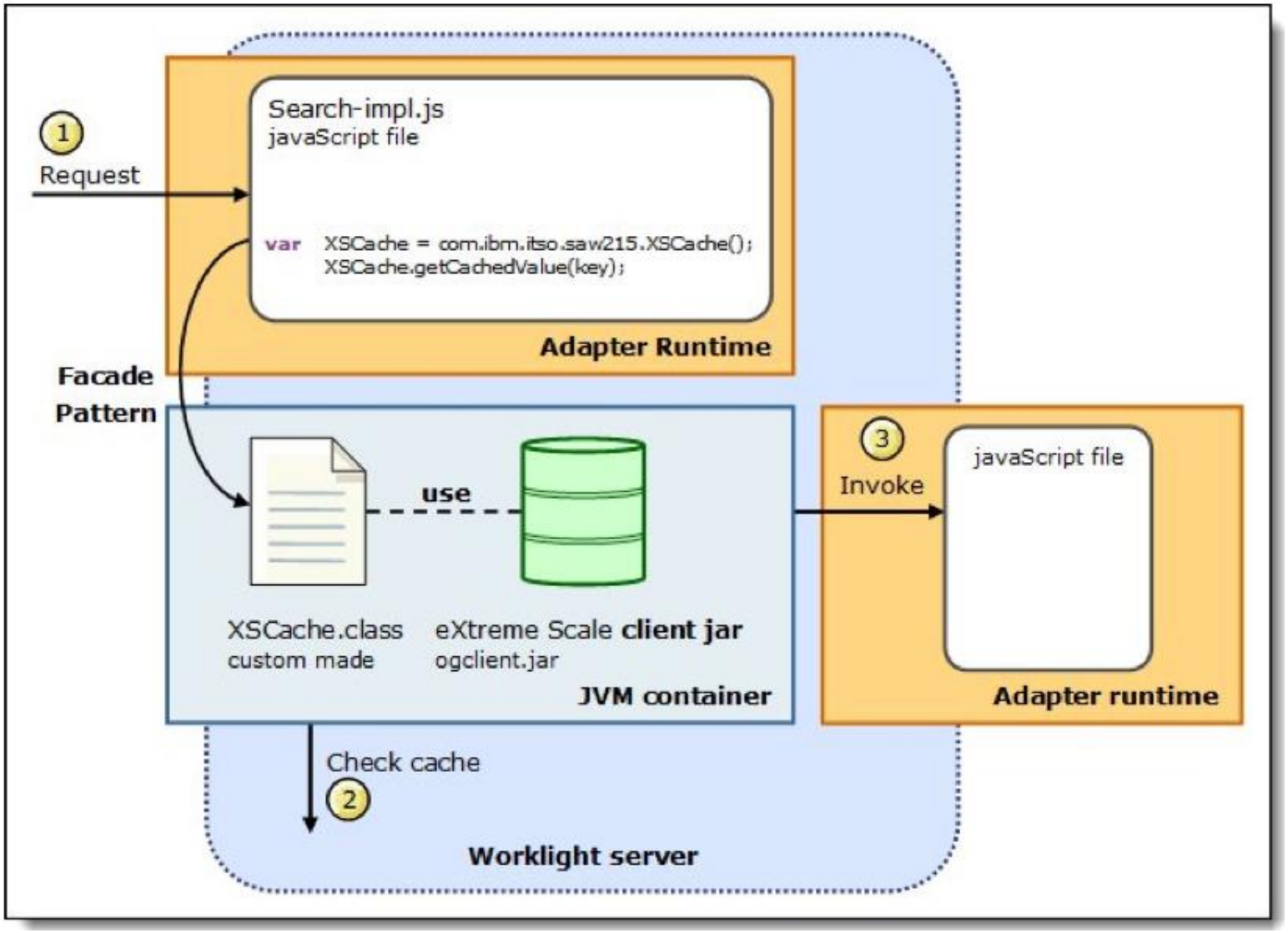
For more information http://www.redbooks.ibm.com/redpapers/pdfs/redp4851.pdf

MobileFirst Platform integration with IBM caching products



Solution Architecture from: http://www.redbooks.ibm.com/abstracts/tips0953.html#contents

MobileFirst Platform Adapter integration with WebSphere Extreme Scale



Architectural decisions

Architectural Decision	Rationale and decision points
When use WebSphere Extreme Scale (WXS)?	WXS is a general-purpose scalable cache. It can be added to any java application running in the mid-tier without requiring changes to any transactions running in the back-end. JavaScript code has to be implemented in the mobile application source to take full benefit of WXS.
When use DataPower XC10 appliance?	Out-of-the box caching appliance that can deliver benefits without adaption of (mobile) application needed. Just configure the network topology to point to the XC10 Typically placed in DMZ to cache static data.
Why use front end caching?	In cases where static data like images, user profiles, product description and HTML are to be cached. Front end caching makes it possible to cache a large set of data, for all requests for (back end) services are processed here. Performance improvement tends to be more of an entry point.
Why use back end caching?	Typically to off-load back end queries in cases where inquiries are made but no transactions are performed.

Conclusion

System z Unique Characteristics to support Mobile Applications

- Easy-to-consume APIs from CICS, DB2, IMS allow you to leverage your investment in z/OS transactions to quickly add a mobile channel.
- z/OS enables massive and simple scalability in a single footprint, to handle the workload of millions of devices and sensors
- MobileFirst Platform security integrates with z/OS security providing end-to-end security and data privacy for mobile apps.
- z/OS Workload Management ensures your crucial applications remain responsive during sharp spikes in demand.
- Low-latency I/O. Mobile usage patterns favor short, read-only data requests (Users check account balances) So fast access to operational data, with low latency, is key. The mainframe offers exceptional I/O with dedicated hardware I/O processors. This reduces latency, which increases mobile app response times.
- Business Resiliency for critical mobile apps

Infrastructure matters for mobile applications. The System z platform's scalability, security, and resilience can enhance critical mobile applications.

Why run MobileFirst Platform Server on System z Linux?

For the same reasons you run web apps there for over a decade:

- Co-location of the MobileFirst Platform server application with data and transactions on z/OS **reduces the latency of access to z/OS data**. Hipersockets provides the lowest latency communication between MobileFirst Platform apps and z/OS SOR. Hipersockets eliminates the need to encrypt traffic between MobileFirst Platform and z/OS.
- Availability and scalability of z/Linux as an environment for both MobileFirst Platform dev/test and production.
- Hardware encryption speeds SSL applications
- All the traditional advantages of consolidating multiple distributed servers onto z/Linux -- Reduce data center footprint, WAS software license savings, simpler management, energy savings.

We recommend running MobileFirst Platform Server in System z Linux for data-rich applications that will heavily leverage data and transactions from z/OS.

See this wiki for more rationale for WL on z.

Links to More Helpful Documents

- System z Mobile Connectivity Guide. This document shows all the ways mobile applications can connect to System z data and transactions, using subsystems like DB2, CICS, IMS, and MQ.
- System z Mobile Security Guide. Coming Soon. A complete description of the security options when using MobileFirst Platform on z and the integration points where z and MobileFirst Platform security features can compliment each other. It shows end-to-end security implementation.
- IBM Mobile Reference Architecture. This is an ISSW (IBM Software Services for WebSphere) architecture that covers the choices and best practices for building mobile applications. It cover the IBM and open source software that can be used. It does not discuss platform considerations, and so compliments well our System z Mobile architecture.

Contacts for more help

- System z Mobile (virtual) Center of Competence
  - Steve Wehr (POK)
  - Gary Puchkoff (POK)
  - Nigel Williams (MOP)
  - Frank van der Wal (MOP)
  - Wilhelm Mild (BOE)
  - Theresa Tai (POK)
- System z Lab Services Mobile practice
  - Richard Young (POK)

THE END