



Mainframe and Mobile

Aligning mainframe development with mobile

Brian Colbert, IBM Rational on z Technical Lead for GMU
colbertb@sg.ibm.com





Comparisons...

- We are always making comparisons
 - New / old, fast / slow, big / small
 - z vs distributed
 - COBOL vs Java
 - Enhance vs rewrite
- Do you replace one with the other just because of some advantages?
- It is not about which is better
 - but what is right for the situation?
 - can we use the best parts of each technology?
- IBM continues to make significant investments and advances in the z platform



Mainframe COBOL is still here, Z is not going away

70% world's data processed with COBOL
95% of Financial and Insurance



30 Billion COBOL transactions ran every day
83% of transactions worldwide



300 Billion Lines of COBOL running in production
5 Billion added each year



```
PROCEDURE DIVISION USING PATIENT-MASTER-REC, RETURN-CD.
PERFORM 000-SETUP-RTN THRU 000-EXIT.
IF NOT PATINS-OPEN
    MOVE -2 TO RETURN-CD
    GO TO 1000-ERROR-RTN.
PERFORM 100-GET-PLAN-DATA THRU 100-EXIT.
IF NOT PLAN-FOUND
    MOVE -1 TO RETURN-CD
    GO TO 1000-ERROR-RTN.
PERFORM 200-GET-PRIMARY-PROVIDER THRU 200-EXIT.
IF PROVIDER-FOUND
    IF NETWORK-FLAG = "Y"
```





Mobile is a significant component in the evolution of computing





Mobile is a mandatory transformation

10 Billion devices by 2020

61% of CIOs put mobile as priority

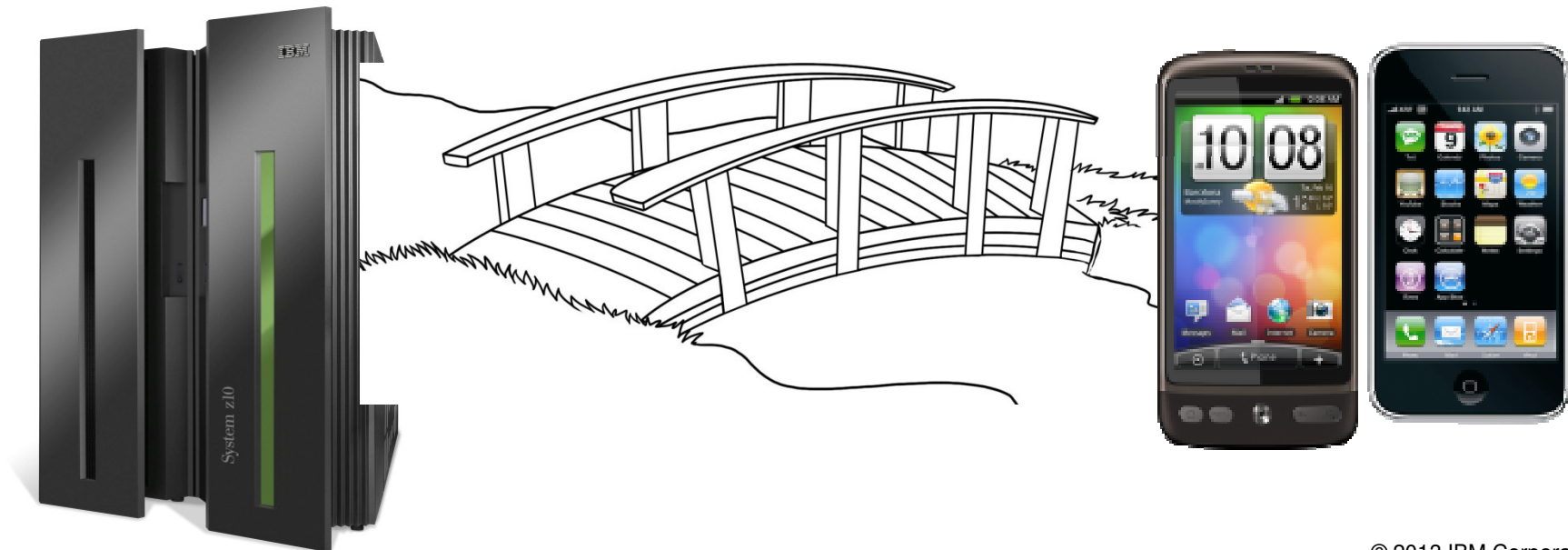
45% increased productivity with mobile apps





Why do we need to bridge the gap between mainframe and mobile?

- All the statistics just mentioned
- The mainframe is central to big business
- Mobile devices are part of every day life
- Data and transactions on the host need to be more accessible
 - Non intrusive manner
 - Dependable
 - Easy to create
 - Flexible

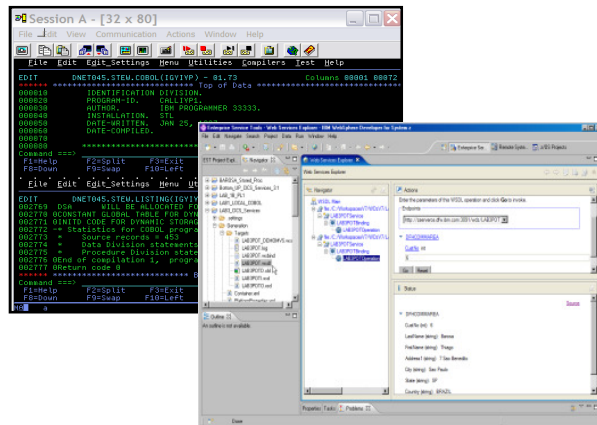




Mainframe Hardware has improved greatly, what about development software?

In September 1956, IBM launched the 305 RAMAC, the first computer with a hard disk drive. The drive weighed over a ton and stored **5MB** of data.”

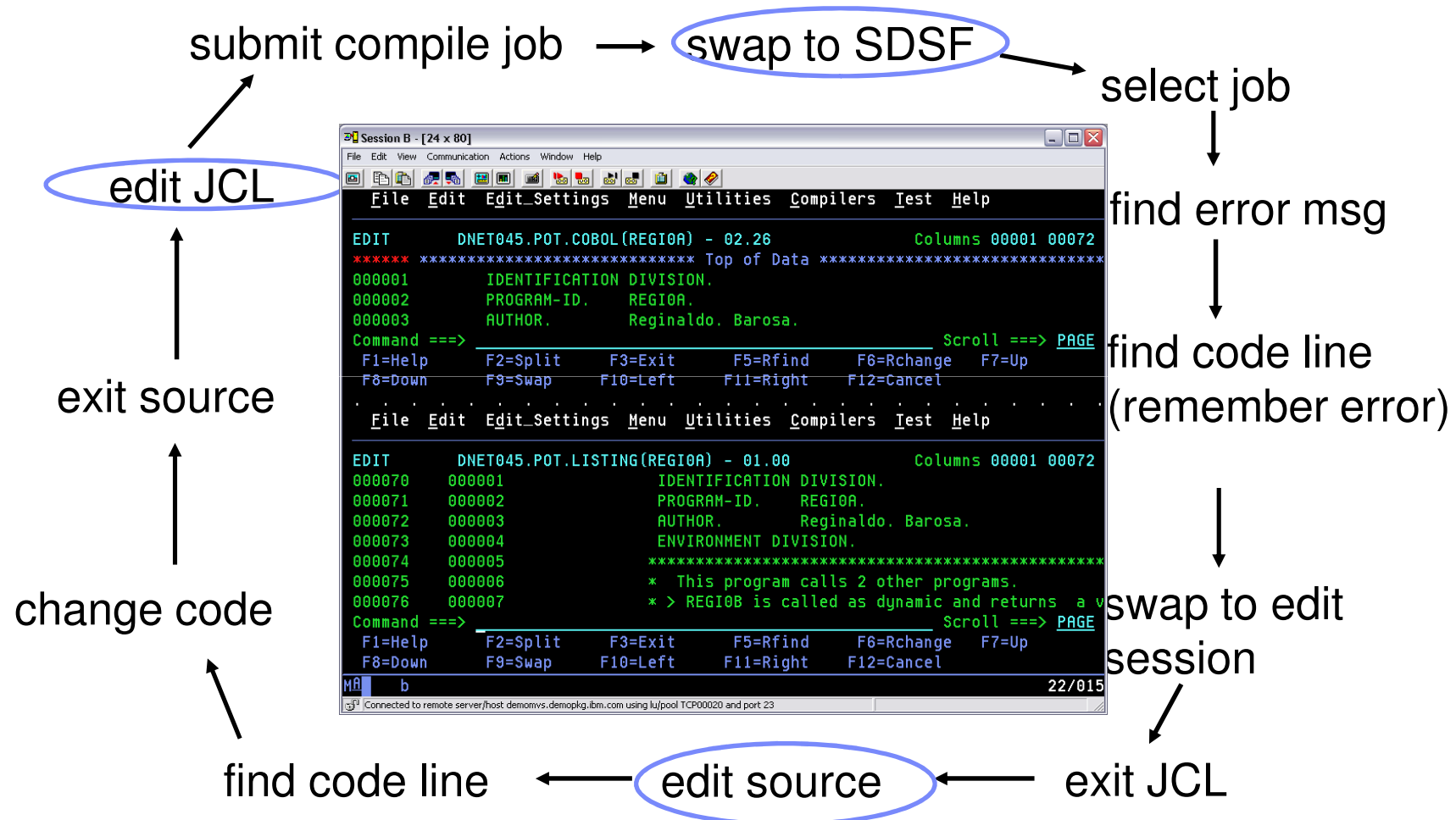
Makes you appreciate your thumb drive with 4 - 128 GB of storage.



➔ Software also needs Improvements....



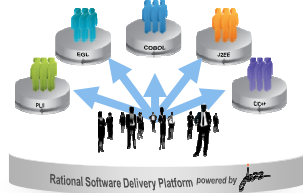
ISPF Based Development





RDz provides a complete set of System z Development and Test capabilities

Integration with Team Concert for Lifecycle and Source Management

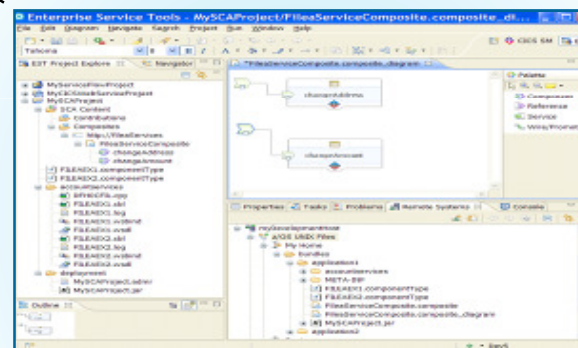


A modern IDE for productive development of cross-platform applications written in COBOL, PL/I, ASM, Java, EGL or C/C++ in System z CICS, IMS, DB2, Batch applications

Access to typical System z sub-system functionality in z/OS, CICS, IMS, DB2, WAS



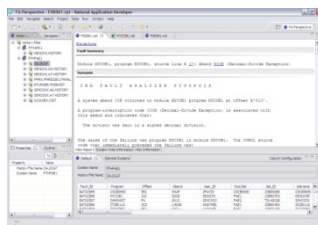
Integration with RD&T for flexible access to System z environment



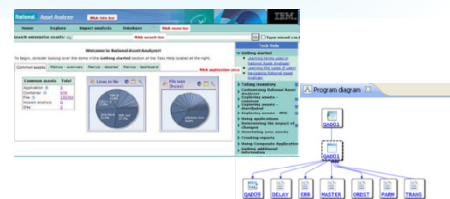
Robust Mobile Development in conjunction with Worklight



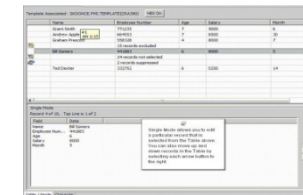
Integration with Fault Analyzer for Dump Analysis



Integration with Asset Analyzer for Application Understanding and Impact Analysis



Integration with File Manager and Fault Analyzer for file and test data handling and Dump Analysis





Architectures: CICS

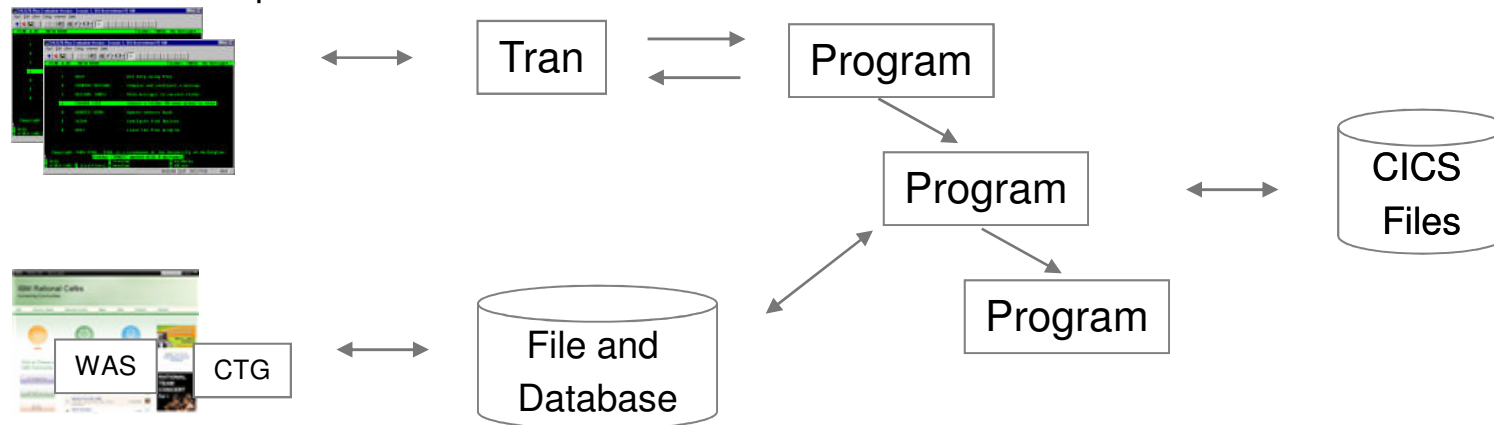
Modernization Requirements

- Make business logic more accessible as services
- Make Data more accessible
- Integrate with new Java processes
- Link with process and rules engines
- Don't impact my current transactional throughput

Modernization Strategies

- Modularize (and service enable) code
- Migrate legacy data stores to relational DB2
- Modify data access / add data layer to access relational DB2
- Rewrite and/or migrate code to rules and/or WAS

3270 - BMS Maps





Architectures: IMS

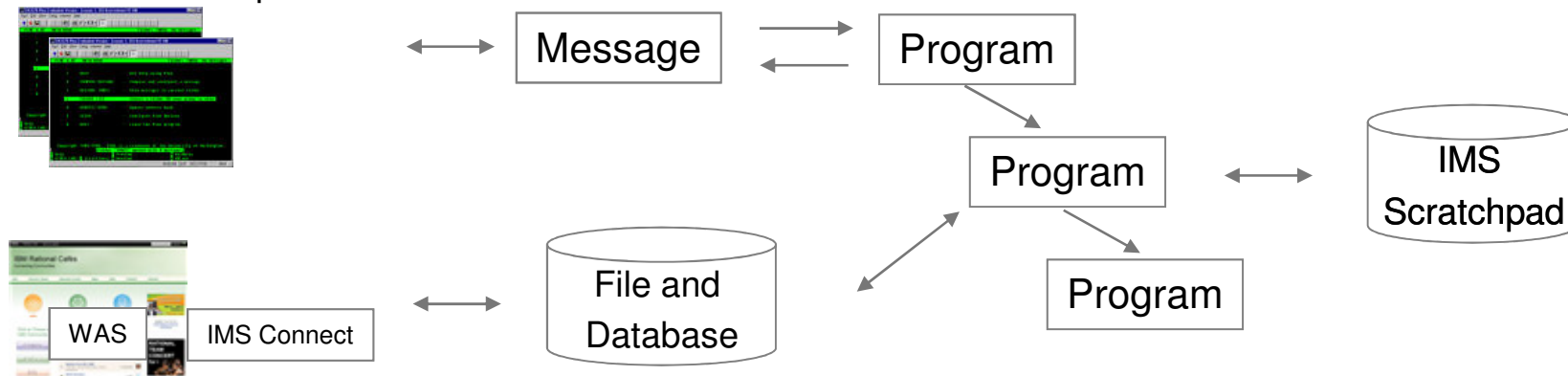
Modernization Requirements

- Make business logic more accessible as services
- Make Data more accessible
- Integrate with new Java processes
- Link with process and rules engines
- Don't impact my current transactional throughput

Modernization Strategies

- Modularize (and service enable) code
- Migrate legacy data stores to relational DB2
- Modify data access / add data layer to access relational DB2
- Rewrite and/or migrate code to rules and/or WAS

3270 – MFS Maps





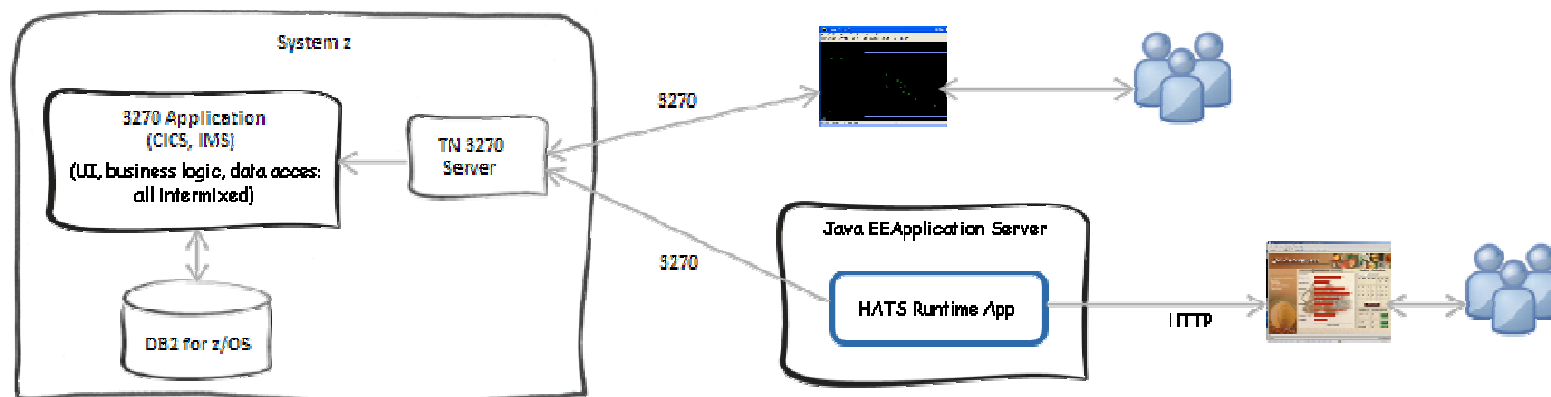
Strategies: Re-facing

Requirements

- Provide web/mobile access to existing apps fast
- Reduce IT operational costs associated with maintaining multiple emulators/versions across desktops
- Applications cannot be modified (lack of source code, lack of skills, too risky, etc)
- Continue providing access via 3270 for power or back-office users

Solution Overview

- Eclipse-based tool for developing and testing
- WebSphere/JEE or Eclipse/Notes runtime
- Quickly transforms green screen applications to Web, Mobile, Portal and rich client interfaces
- Retain 3270 look/feel/navigation if needed
- Iteratively improve application look and usability
- No changes required to existing applications





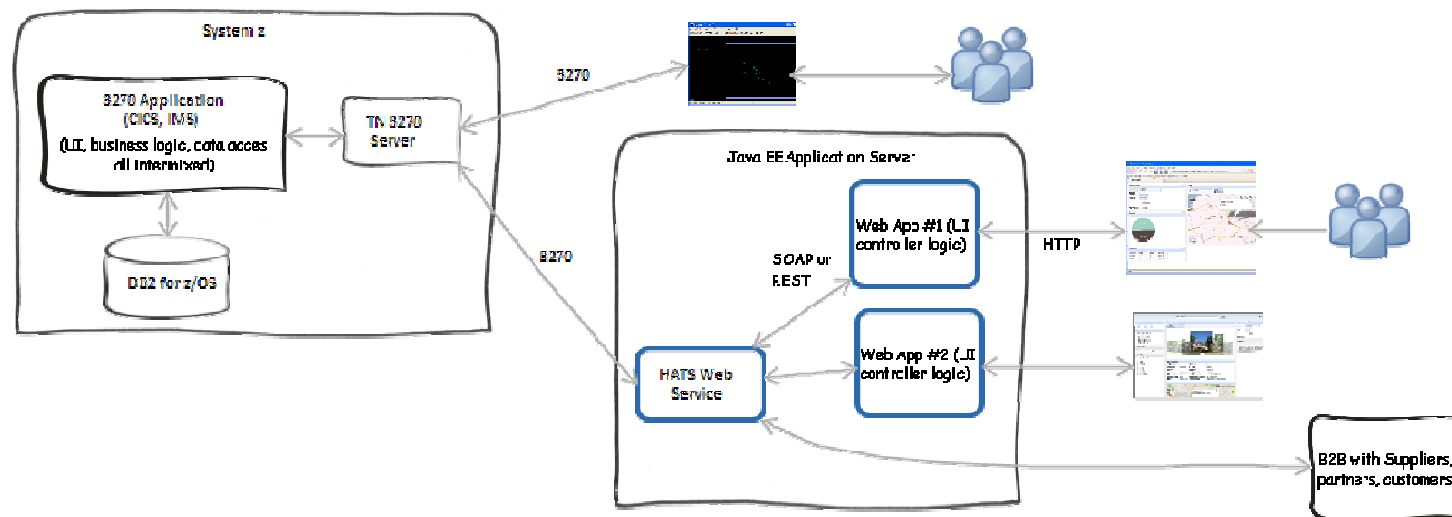
Strategies: Service Wrapping

Requirements

- Quickly expose existing business logic for integration with other systems
- Web UI needs to drive application flow and support integration with other services (mashup scenario)
- Applications cannot be modified (lack of source code, lack of skills, too risky, etc)
- Keep access via 3270 intact

Solution Overview

- Web service-enable flows through 3270 applications without changing the existing applications
- Enables integration of host business processes and data with other Web, Portal, Mobile, and rich client applications





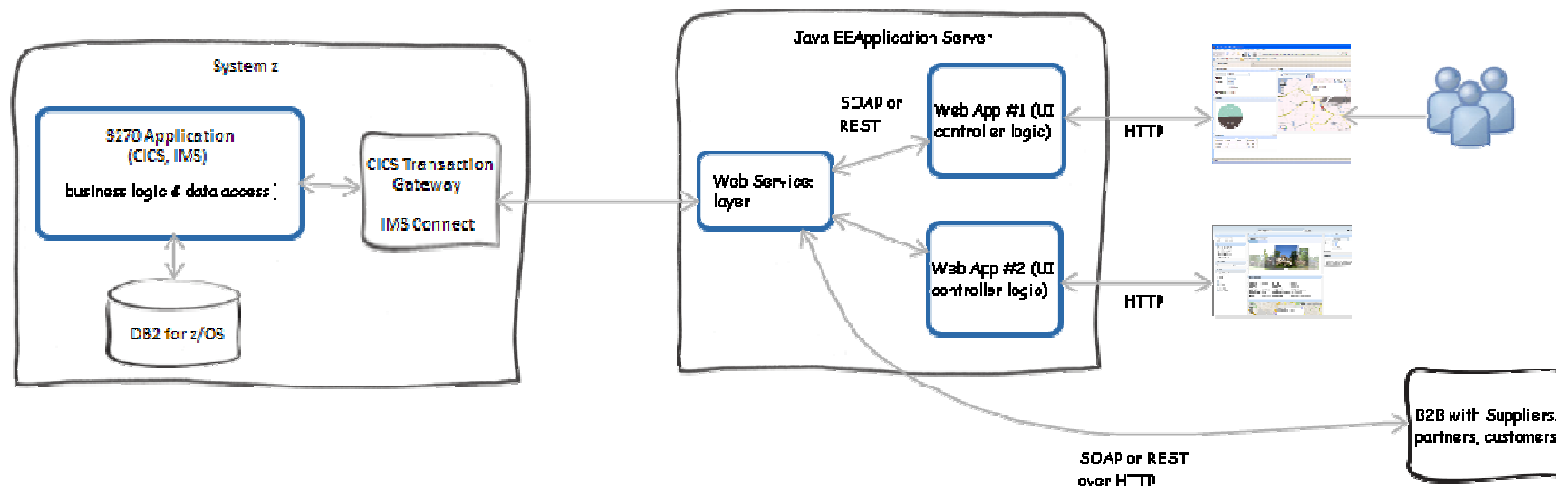
Strategies: Application Refactoring

Requirements

- Meet new business needs that overcome the constraints of the old system
- More flexibility and improved ability to respond quickly to business requirements for a competitive advantage
- Enable business process automation by integrating systems (internal, suppliers, partners, clients)

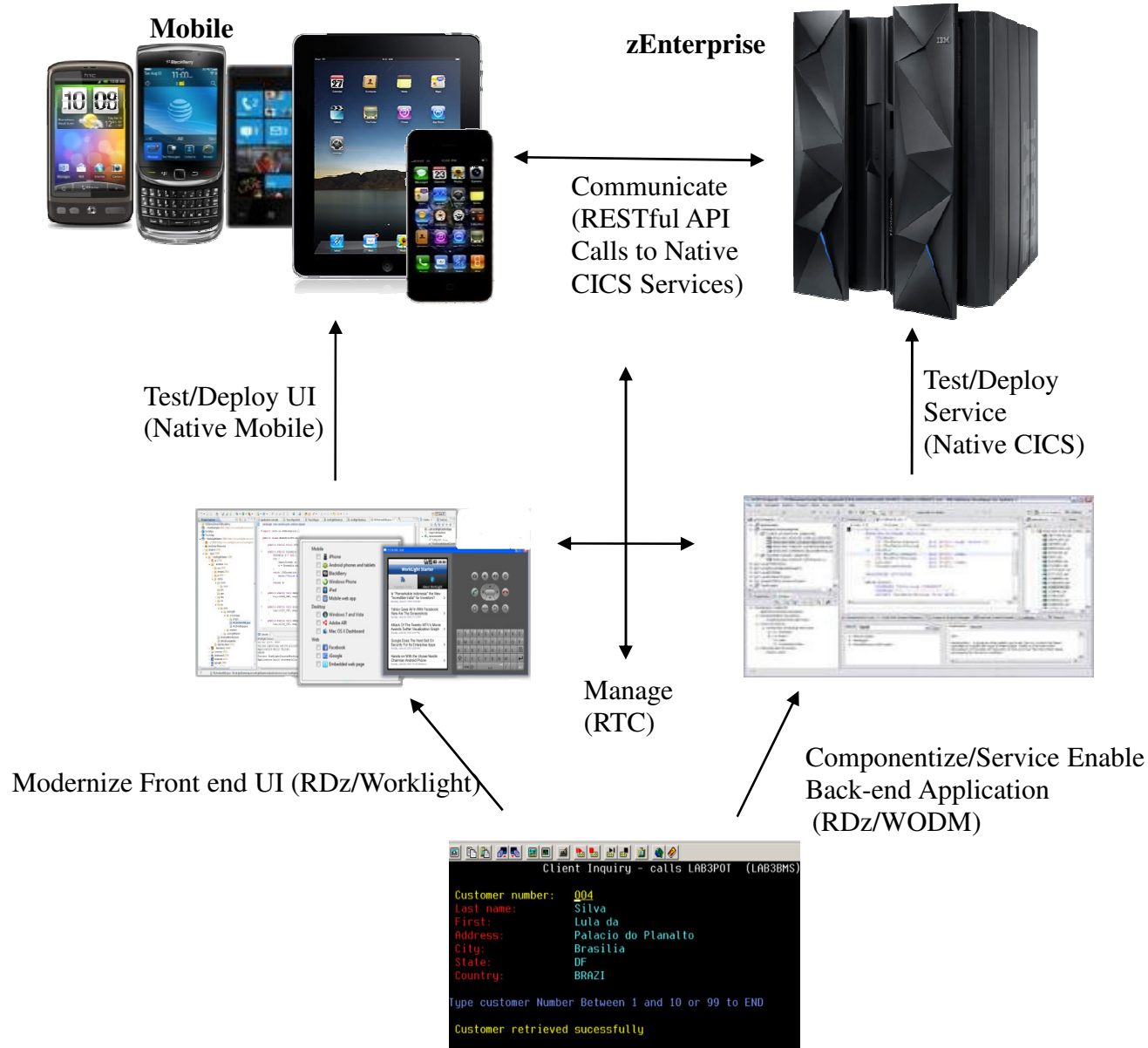
Solution Overview

- “De-tangle” UI from COBOL business logic and create programs & service programs that have well-defined interfaces
- Build a services layer that exposes these reusable programs as web services, either with Java or EGL wrappers
 - ▶ Driven by new web/mobile applications
- Complement RD for z tools with other Rational tools for application understanding and analysis





Typical Mainframe to Mobile Scenario With Rational Developer for System z





The Broadest Portfolio of Mobile Solutions

IBM MobileFirst Platform

IBM MobileFirst Management

IBM MobileFirst Security

IBM MobileFirst Analytics



Key mobile development and delivery challenges

Fragmentation and developing for multiple mobile platforms

- Highly fragmented set of devices, platforms, languages, and tools complicates development, test, and operations



Delivering high quality apps

- Consumers demand a high quality user experience where quality is influenced as much by design as it is by function

Customer Ratings



Accelerated time to market requirements

- Higher frequency of new releases puts added pressure on teams to deliver on time and with high quality



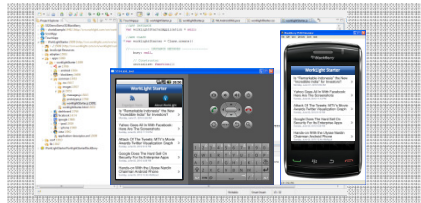
Connecting apps with existing enterprise systems

- Apps typically need to leverage existing enterprise services, which must be made mobile-consumable
- Enterprise wireless networks are running out of bandwidth to accommodate employee devices



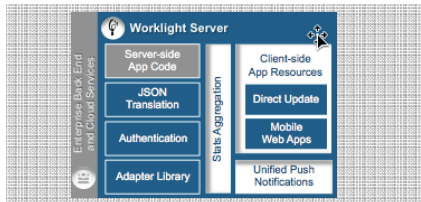


Worklight Overview



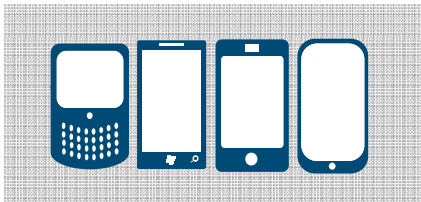
Worklight Studio

The most complete, extensible environment with maximum code reuse and per-device optimization



Worklight Server

Unified notifications, runtime skinning, version management, security, integration and delivery



Worklight Runtime Components

Extensive libraries and client APIs that expose and interface with native device functionality



Worklight Console

A web-based console for real-time analytics and control of your mobile apps and infrastructure

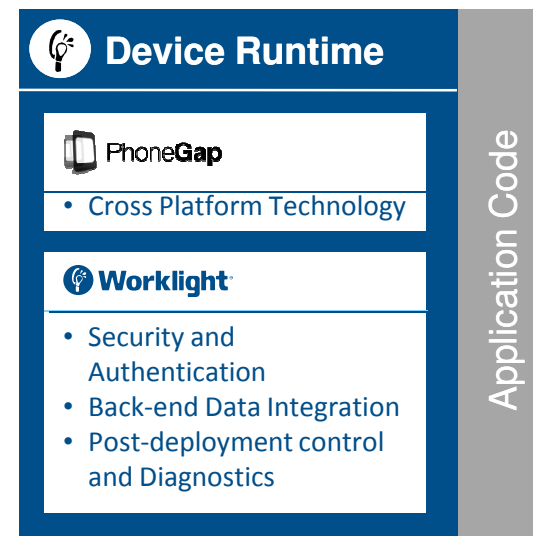
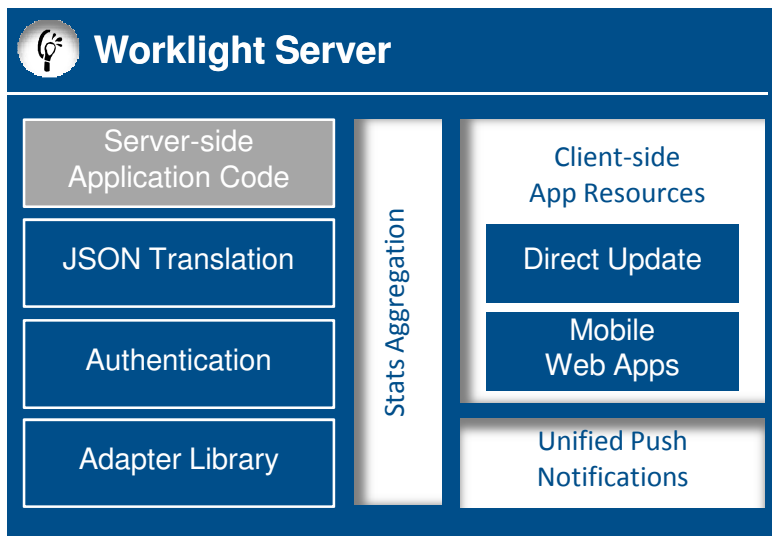
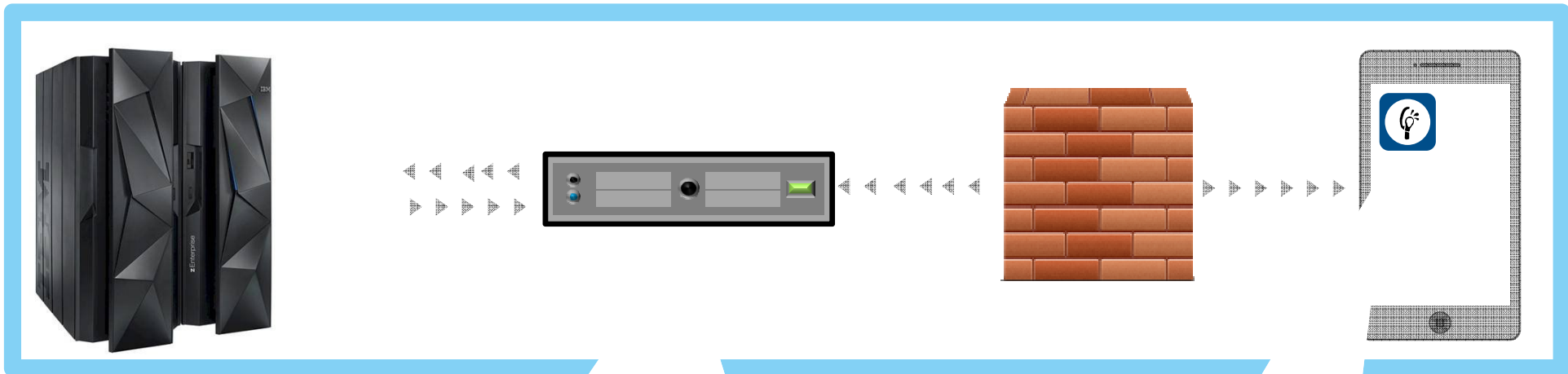


Worklight Application Center

A cross-platform private mobile application store focused on the needs of a development organization or a team



Worklight Application Topology





IBM MobileFirst Offering Portfolio

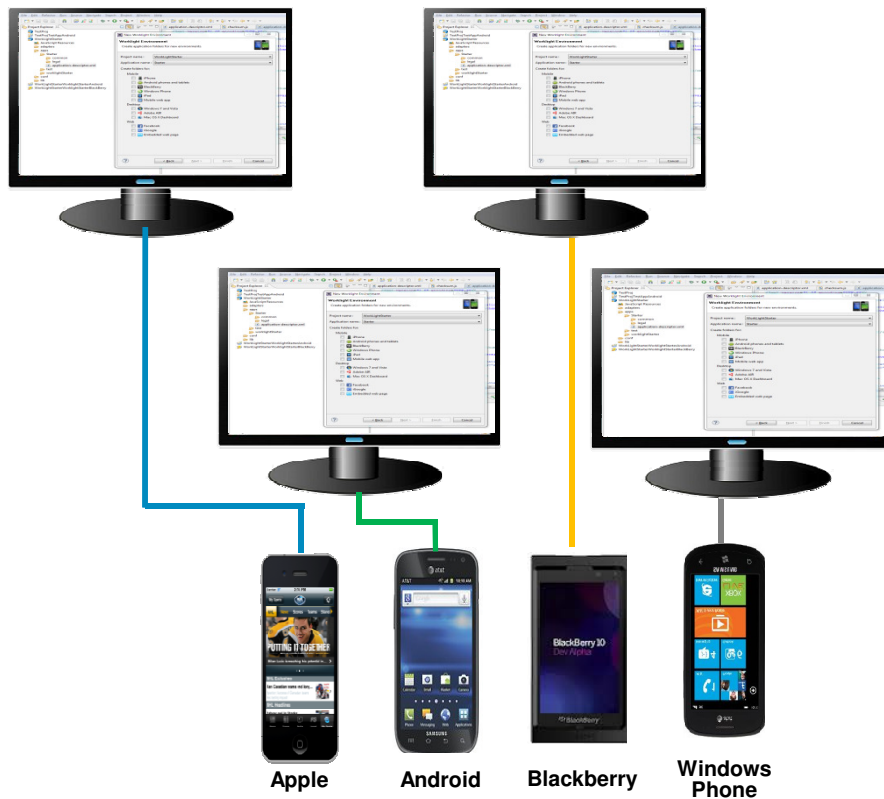




Rapid multi-platform development using a single shared codebase

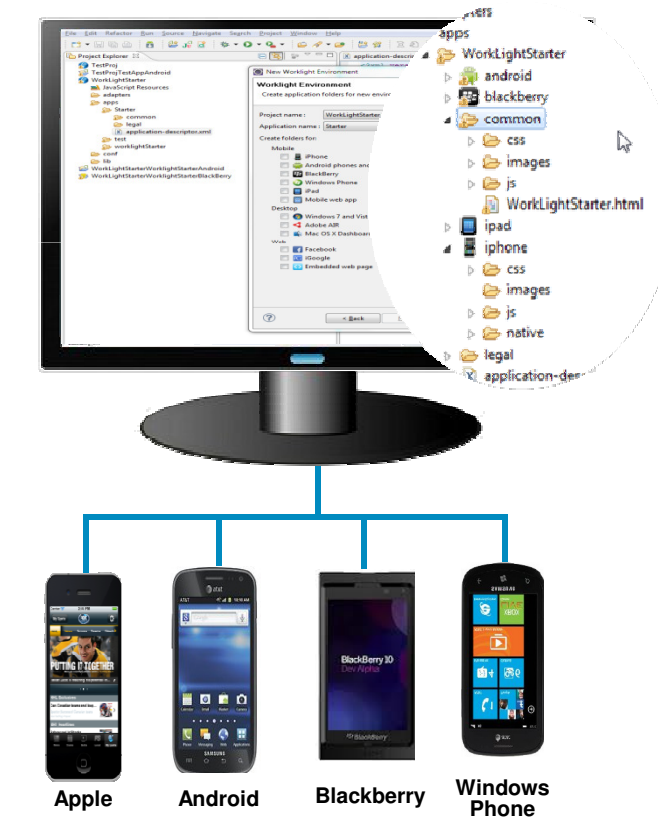
From the complexity of many...

- Multiple sets of tools & frameworks
- Four codebases to develop and maintain



To the simplicity of one

- One development environment
- One codebase to develop and maintain



IBM MobileFirst Platform



Runtime Skinning – Use Cases

Different
Screen Sizes



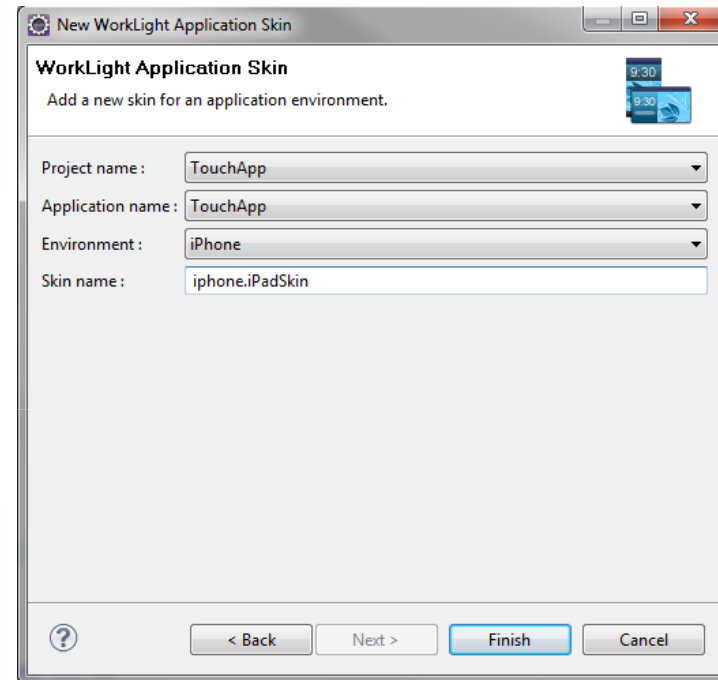
Different
Screen Densities



Different
Input Method

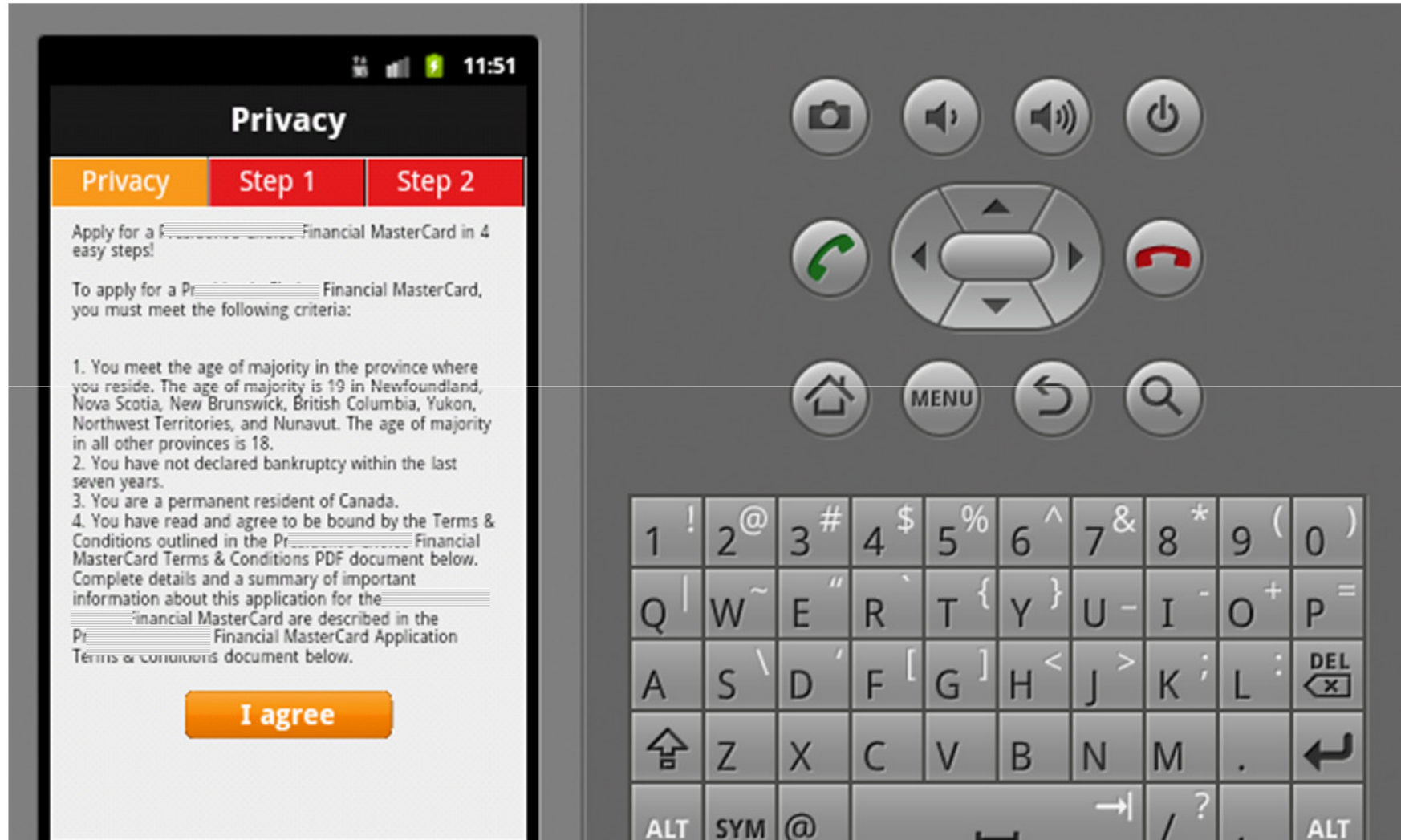


Support
for HTML5



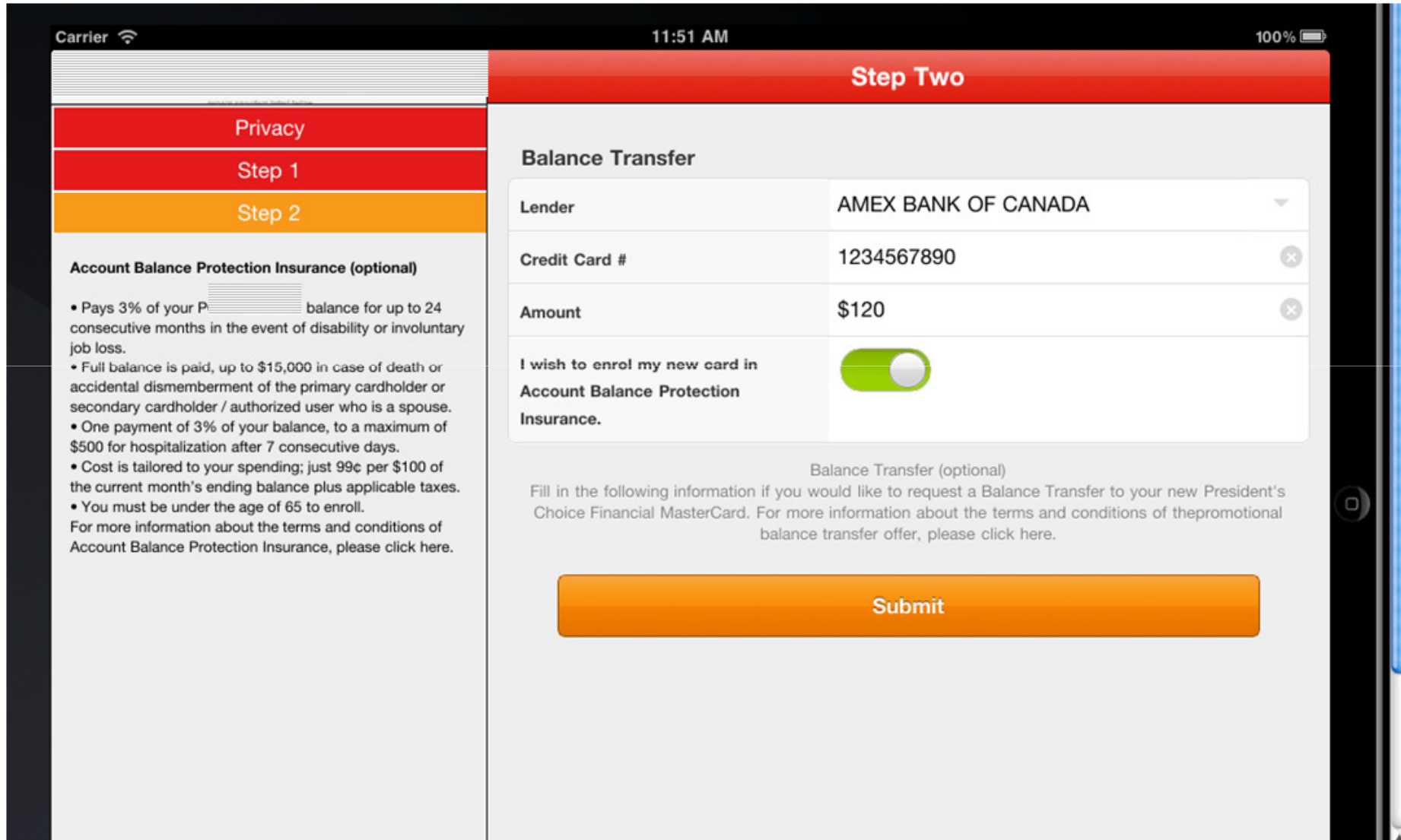


Example Mobile Skin on Android





Example Mobile Skin on iPad





Mobile user Interface Automation

High quality mobile apps built and tested for a rapidly evolving mobile infrastructure

▪ Mobile Test Automation

- Support for native and hybrid applications running on Android and iOS
- Natural language test and visual test editing
- Simplified IDE and mobile device clients for test authoring, execution, and reporting

▪ Eclipse environment

- Mobile Test Automation
- Support for Selenium web GUI test automation
- Multi-channel test scenarios
- Integration with Rational Quality Manager and the IBM Worklight mobile development platform

▪ End-to-end mobile testing

- Virtualization of middle-tier and back-end systems
- Dynamic instrumentation of mobile applications
- Capture and high-fidelity replay of multi-touch events

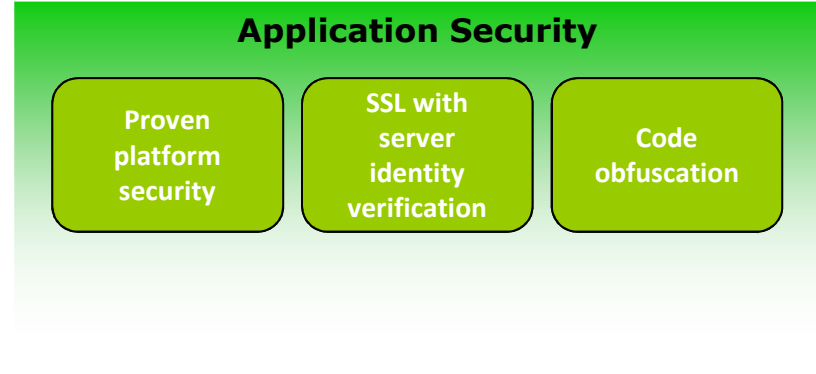
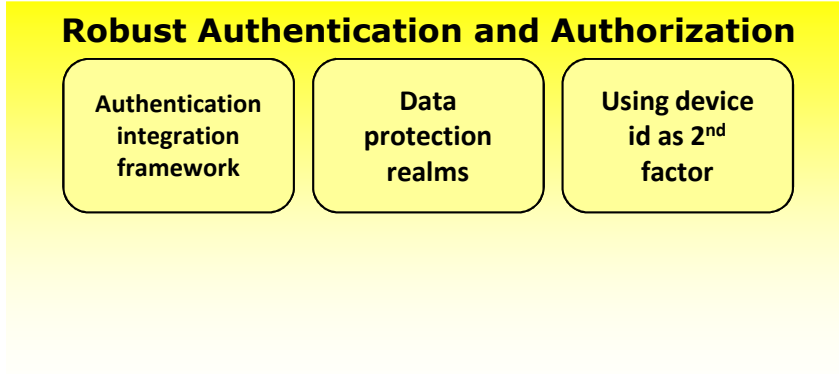
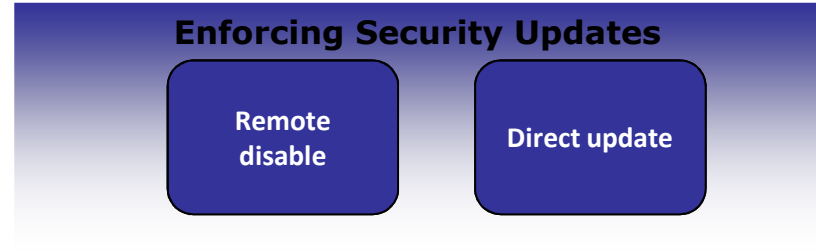
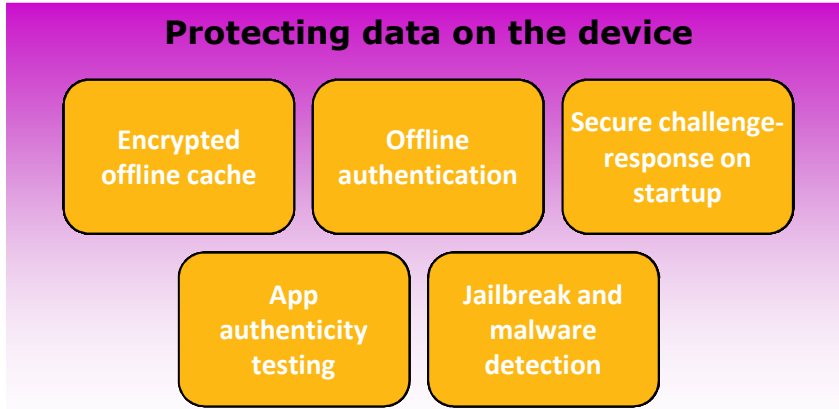


IBM Rational Test Workbench

BETA now
available !



Securing Mobile Apps

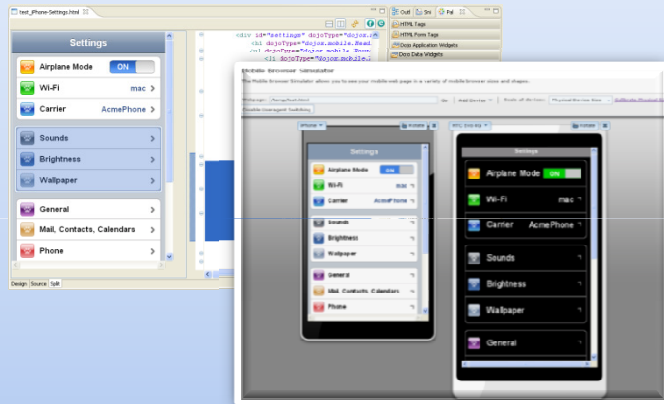




Rational Developer for zEnterprise with IBM Worklight

Design, code, build, test, and deploy mobile apps that run on a wide variety of mobile platforms; extend existing back-end services and data to mobile apps

Integrated multi-platform development environments



Construct, debug, and test mobile UIs



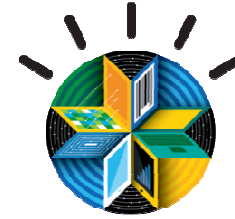
Refactor and extend existing logic on enterprise platforms (System z, Power) as mobile-consumable services

IBM Worklight will be included in the following IDEs (for development purposes only):

- Rational Developer for zEnterprise
- Rational Developer for Power Systems
- Rational Application Developer
- Rational Software Architect



IBM MobileFirst



Thank you!