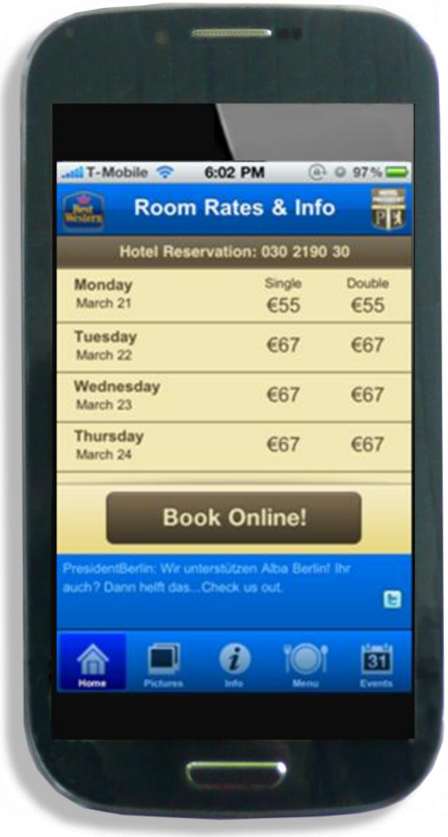
WebSphere Puts Business In Motion

Put People In Motion
With Mobile Apps

Use Mobile Apps to create new revenue opportunities

- A clothing store increases sales through personalized offers
  - Customers can scan merchandise tags while in the store to get special offers

- A hotel chain increases occupancy rate
  - Customers can easily book a room at the hotel that is nearest to their current location



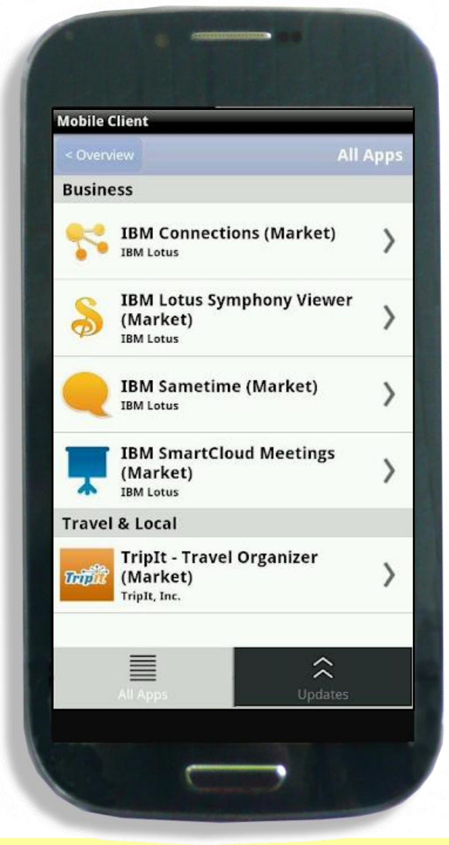
	Single	Double
Monday March 21	€55	€55
Tuesday March 22	€67	€67
Wednesday March 23	€67	€67
Thursday March 24	€67	€67

PresidentBerlin: Wir unterstützen Alba Berlin! Ihr auch? Dann helft das...Check us out.

Drive your business to a rapidly growing new channel

Use Mobile Apps to improve the effectiveness of your employees

- Heavy equipment manufacturer improves on-site product servicing
  - Technicians always have access to latest manuals and troubleshooting procedures, even at a job site
- Multinational company increases effectiveness of sales teams
  - Sales representatives have mobile dashboard with their opportunity list, next steps, pending sales, appointments, etc.



Mobile Client

< Overview

All Apps

Business

- IBM Connections (Market)
  IBM Lotus
- IBM Lotus Symphony Viewer (Market)
  IBM Lotus
- IBM Sametime (Market)
  IBM Lotus
- IBM SmartCloud Meetings (Market)
  IBM Lotus

Travel & Local

- TripIt - Travel Organizer (Market)
  TripIt, Inc.

All Apps | Updates

Improve Employee Productivity

Building and managing mobile apps pose considerable challenges

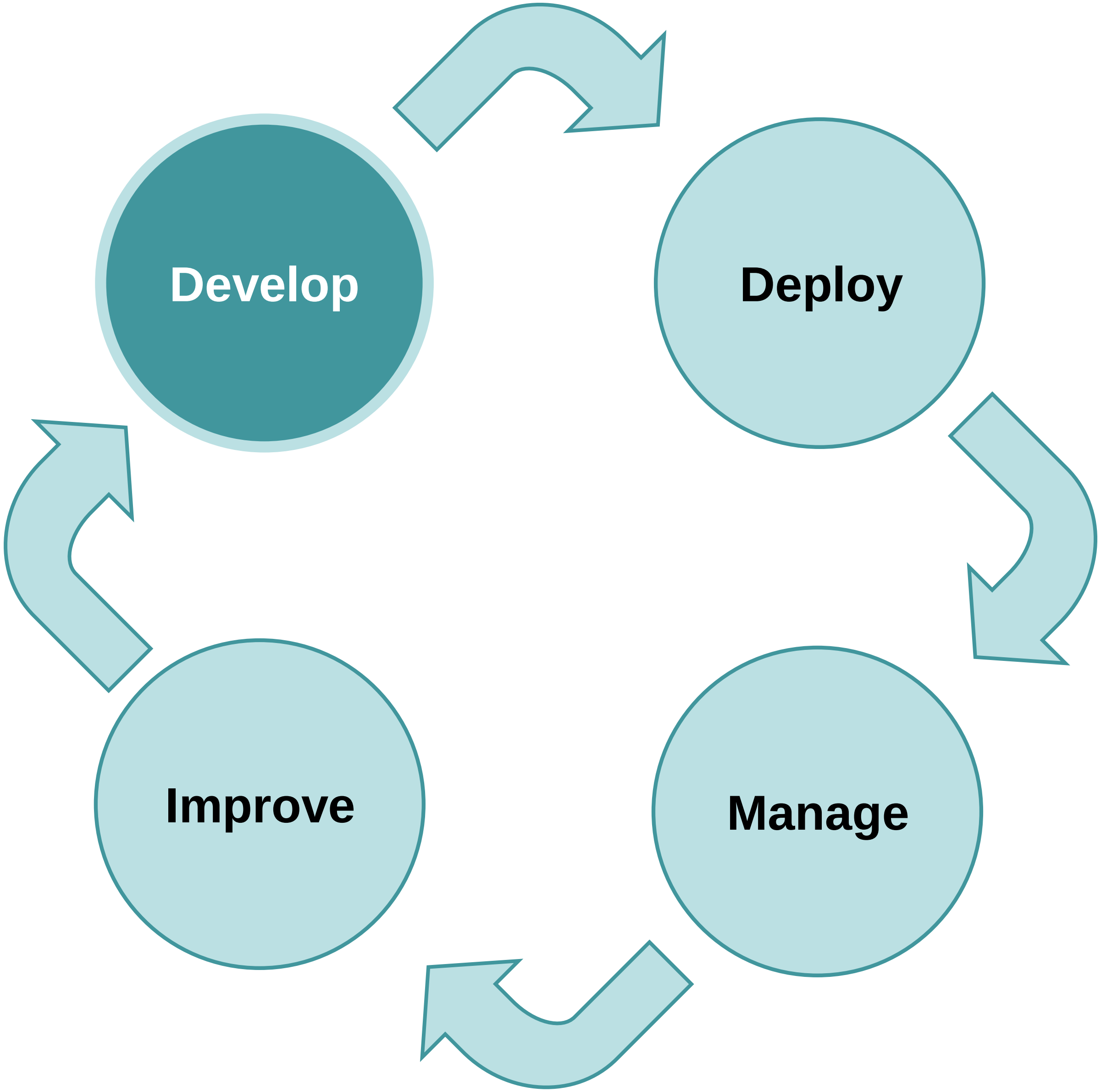
You want rich yet cost-effective mobile apps, but different mobile devices require different programming models, languages, and tools.

You want to reuse existing enterprise services, but you are concerned about giving enterprise access to non-enterprise assets.

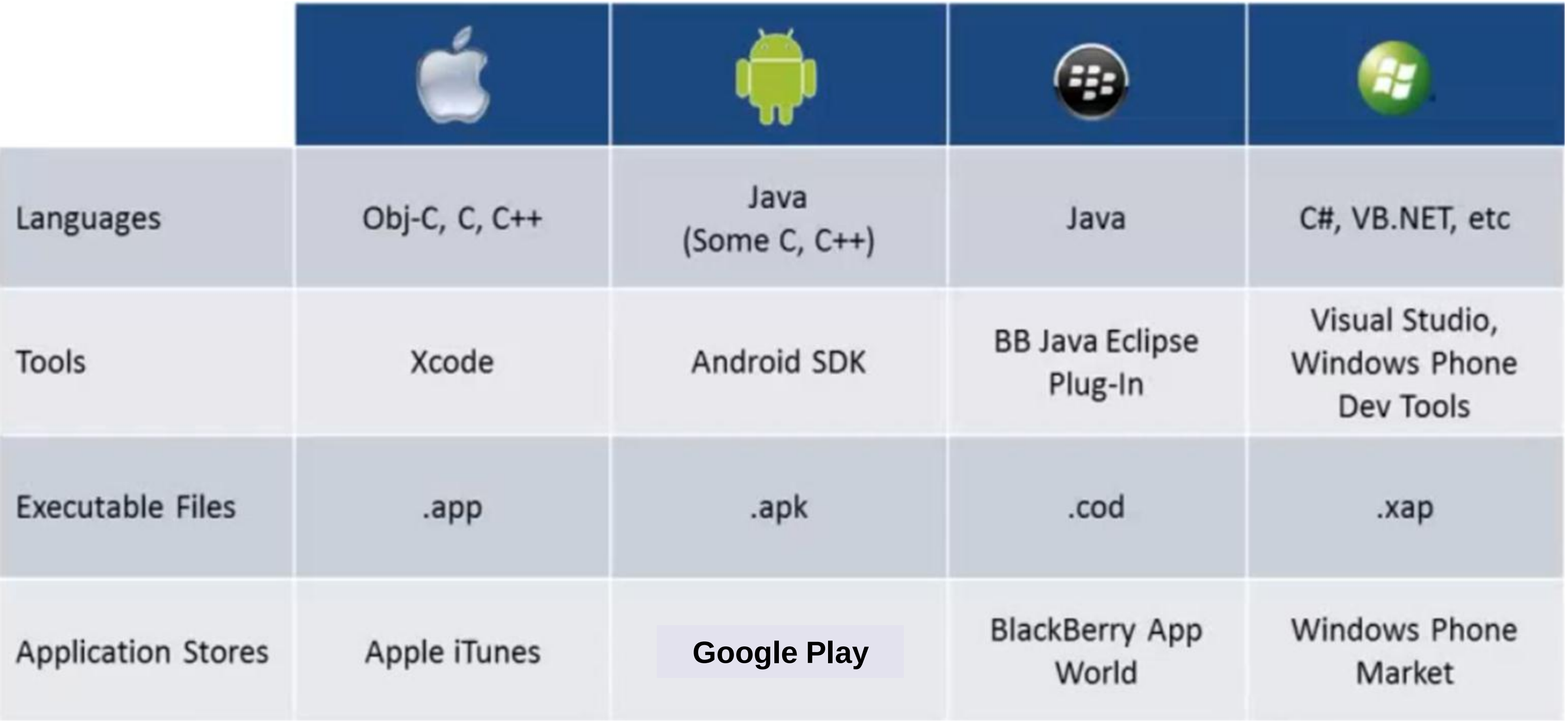
You need to manage your apps and data on personal and company devices, and manage versions of mobile apps on all devices.

You want to study mobile usage experiences in order to better design your mobile apps.

Develop a hybrid app to maximize common code used across platforms



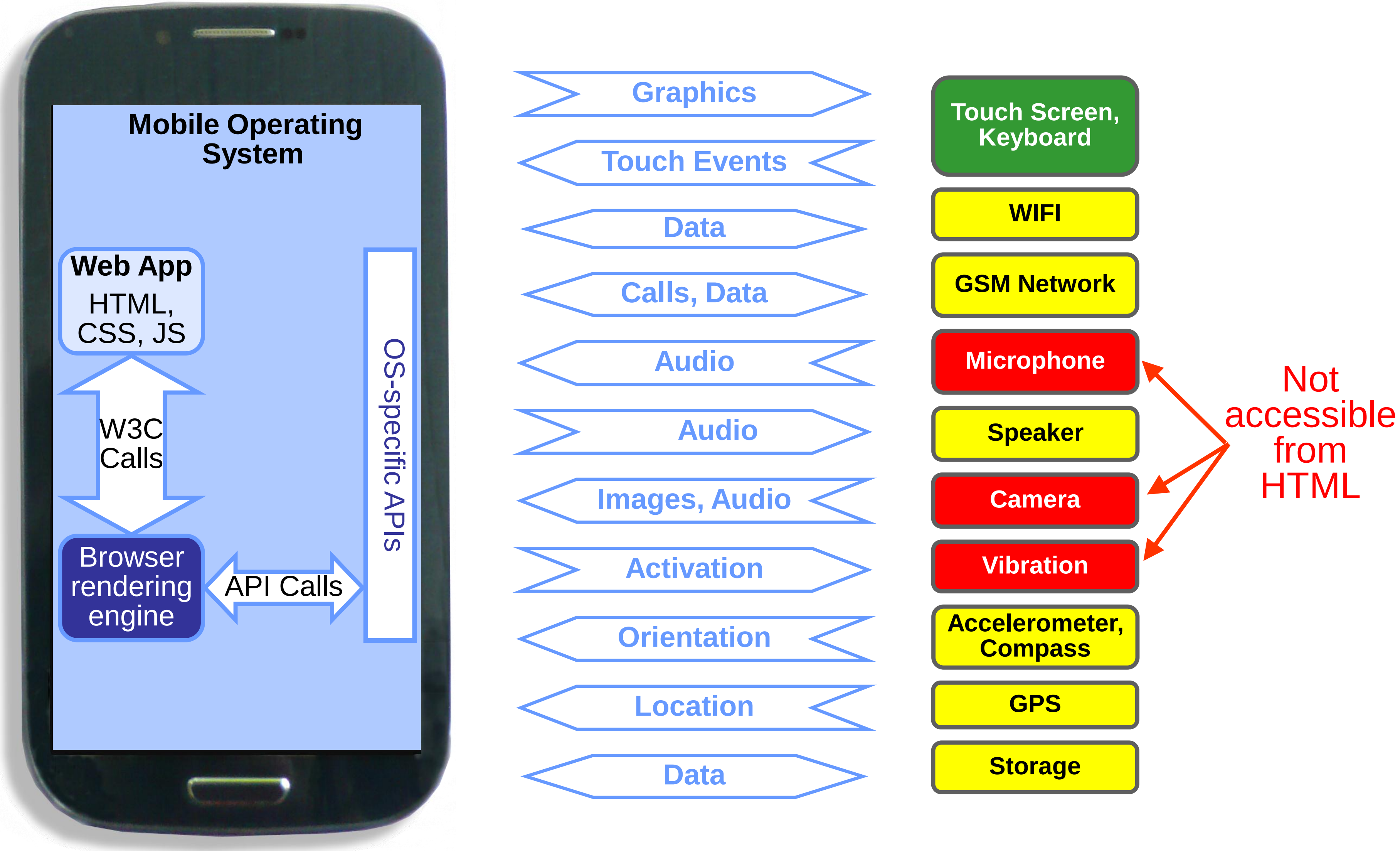
Native Mobile Apps require different tools, APIs, and executable formats



	iOS (Apple)	Android (Google)	Blackberry (RIM)	Windows Phone (Microsoft)
Languages	Obj-C, C, C++	Java (Some C, C++)	Java	C#, VB.NET, etc
Tools	Xcode	Android SDK	BB Java Eclipse Plug-In	Visual Studio, Windows Phone Dev Tools
Executable Files	.app	.apk	.cod	.xap
Application Stores	Apple iTunes	**Google Play**	BlackBerry App World	Windows Phone Market

A native app for one platform cannot run on another,
so you have to develop separate apps for each platform

Standards-based Mobile Apps have limited access to device features



Apps written in HTML run on all platforms, but can't use important device features and don't have device's native look and feel

Considering these trade-offs, what is the best way to create Mobile Apps?

Mobile Web apps

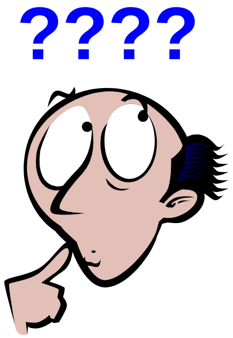
Written for portability using HTML, CSS, and JavaScript

- ✓ One app can be run on a wide range of platforms
- ✗ Some device capabilities aren’t accessible, preventing creation of a rich mobile experience
- ✗ Apps may not have native look and feel that users expect

Native apps

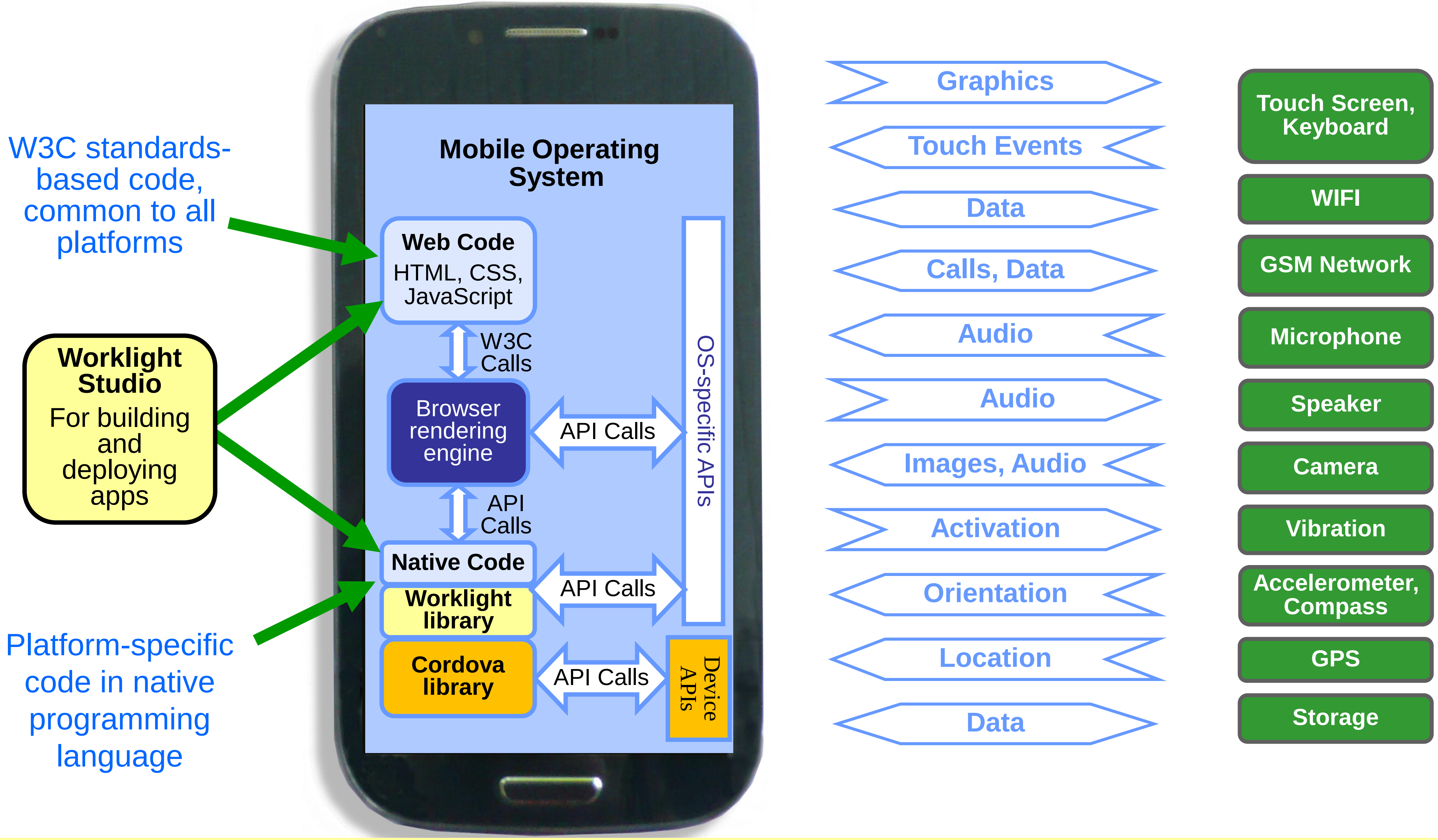
Written for specific device type, using the device’s architecture and language

- ✓ Best possible user experience for device
- ✗ App must be rewritten for each platform
- ✗ No code sharing between apps complicates consistent behavior



Build a ***Hybrid App*** with IBM Worklight!

Worklight Makes it easier to build rich hybrid app across platforms by integrating Apache Cordova



W3C standards-based code, common to all platforms

Worklight Studio

For building and deploying apps

Platform-specific code in native programming language

Mobile Operating System

Web Code

HTML, CSS, JavaScript

W3C Calls

Browser rendering engine

API Calls

API Calls

Native Code

API Calls

Worklight library

Cordova library

API Calls

OS-specific APIs

Device APIs

Graphics

Touch Events

Data

Calls, Data

Audio

Audio

Images, Audio

Activation

Orientation

Location

Data

Touch Screen, Keyboard

WIFI

GSM Network

Microphone

Speaker

Camera

Vibration

Accelerometer, Compass

GPS

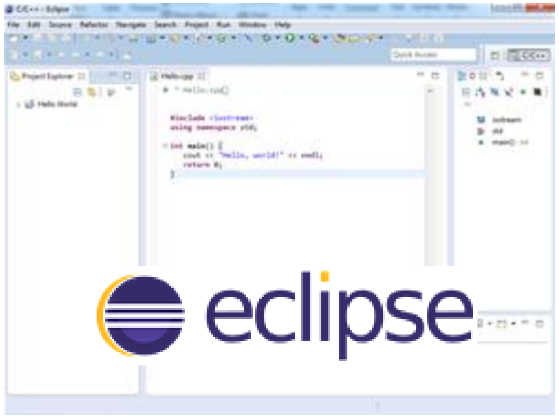
Storage

All device features can be accessed

Share common code between platforms

DIY (Do-It-Yourself) / Open Source offers useful tools for developing hybrid mobile apps

- Eclipse IDE for JavaScript Web Developers provides basic support for HTML, CSS, and Javascript
- Cordova/PhoneGap provides common JavaScript APIs to create a hybrid mobile app to maximize re-use of code among platforms
  - Access to device capabilities
  - Access to native UI functions
- …but you need more than Eclipse and Cordova to make it work.



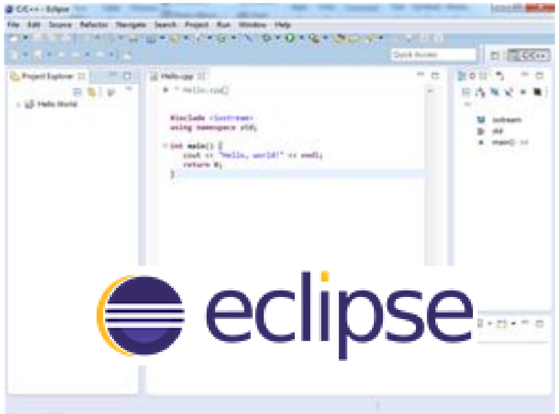
```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, world!" << endl;
    return 0;
}
```

APACHE CORDOVA™

The DIY / Open Source mobile development platform needs more than Eclipse and Cordova

- A list of required components is not provided
  - Have to study errors, do online research to figure out what else you need to install and configure
- Several supporting components need to be downloaded, installed, configured, integrated, and maintained, or it won't work
  - Node.js
  - Ant
  - Git
  - jQuery
  - SDKs for Android and iOS
- No formal support available



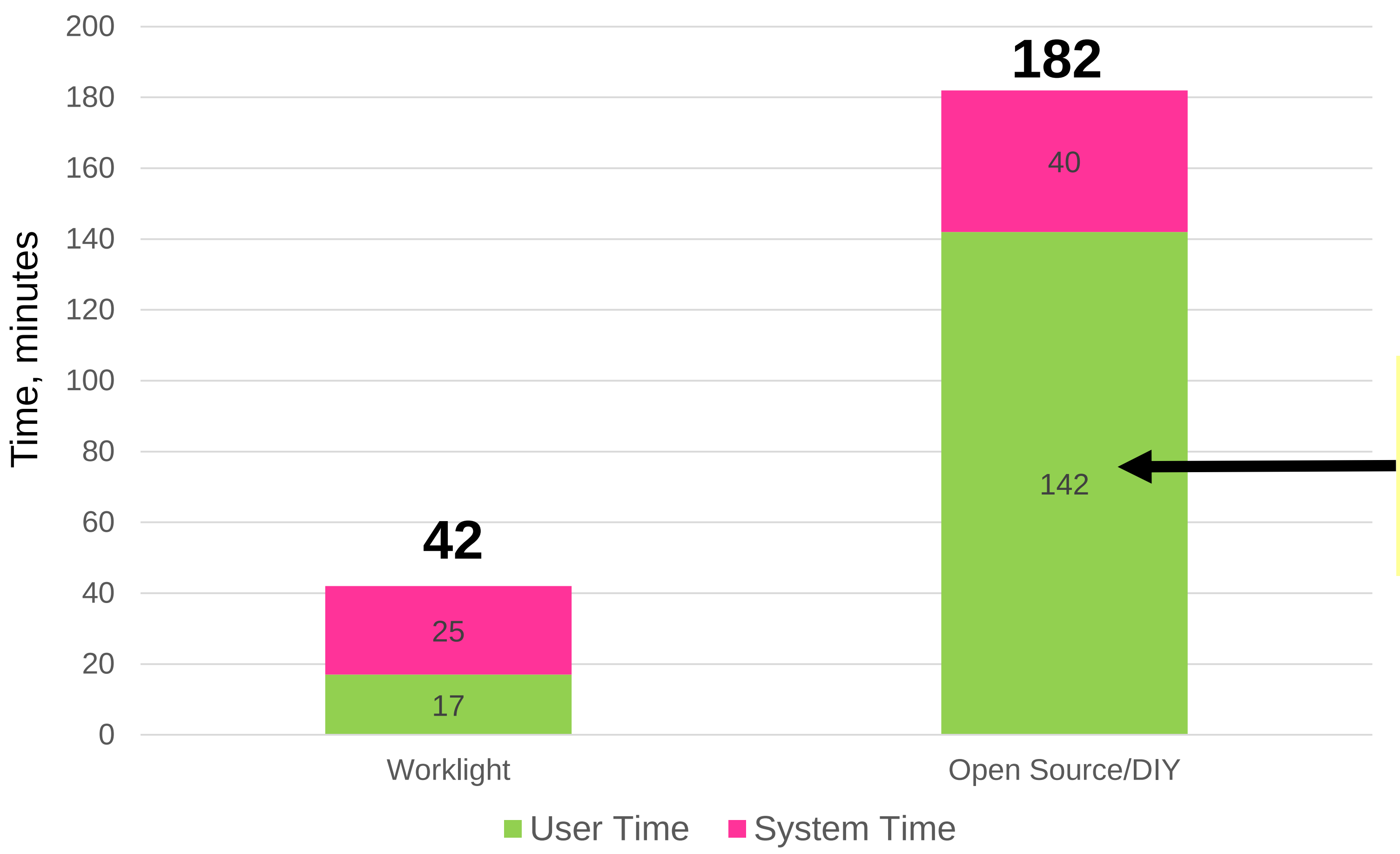
```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, world!" << endl;
    return 0;
}
```

APACHE CORDOVA™

GitHub | jQuery write less, do more. | APACHE ANT™ | node JS | Android SDK | Apple xCode and iOS SDK

Setting up a Worklight development environment is 4.3x faster than DIY



Let's have a look at why the User Time was so high for Open Source / DIY

Note: For open source/DIY scenario, time shown does not account for many hours and days reading documents and debugging integration problems.

Study subject was an experienced software engineer with open source experience

DIY / Open Source installation has a steep learning curve

There is **NO** comprehensive guide to installing and configuring Cordova

- We installed the base requirements (Android SDK, Node.js, Cordova 3.3)
- Attempting to build resulted in errors that had to be resolved. For example:

```
cordova create MyFirstApp (ANT missing error)
        [Error: An error occured during creation of android sub-project.
        C:\Users\.cordova\lib\android\cordova\3.3.0\bin\create: line 104: ant: command not
        found
        An unexpected error occurred: ant jar > /dev/null exited with 127
        Deleting project...
Error: spawn ENOENT
           at errnoException (child_process.js:980:11)
           at Process.ChildProcess._handle.onexit (child_process.js:771:34)
```

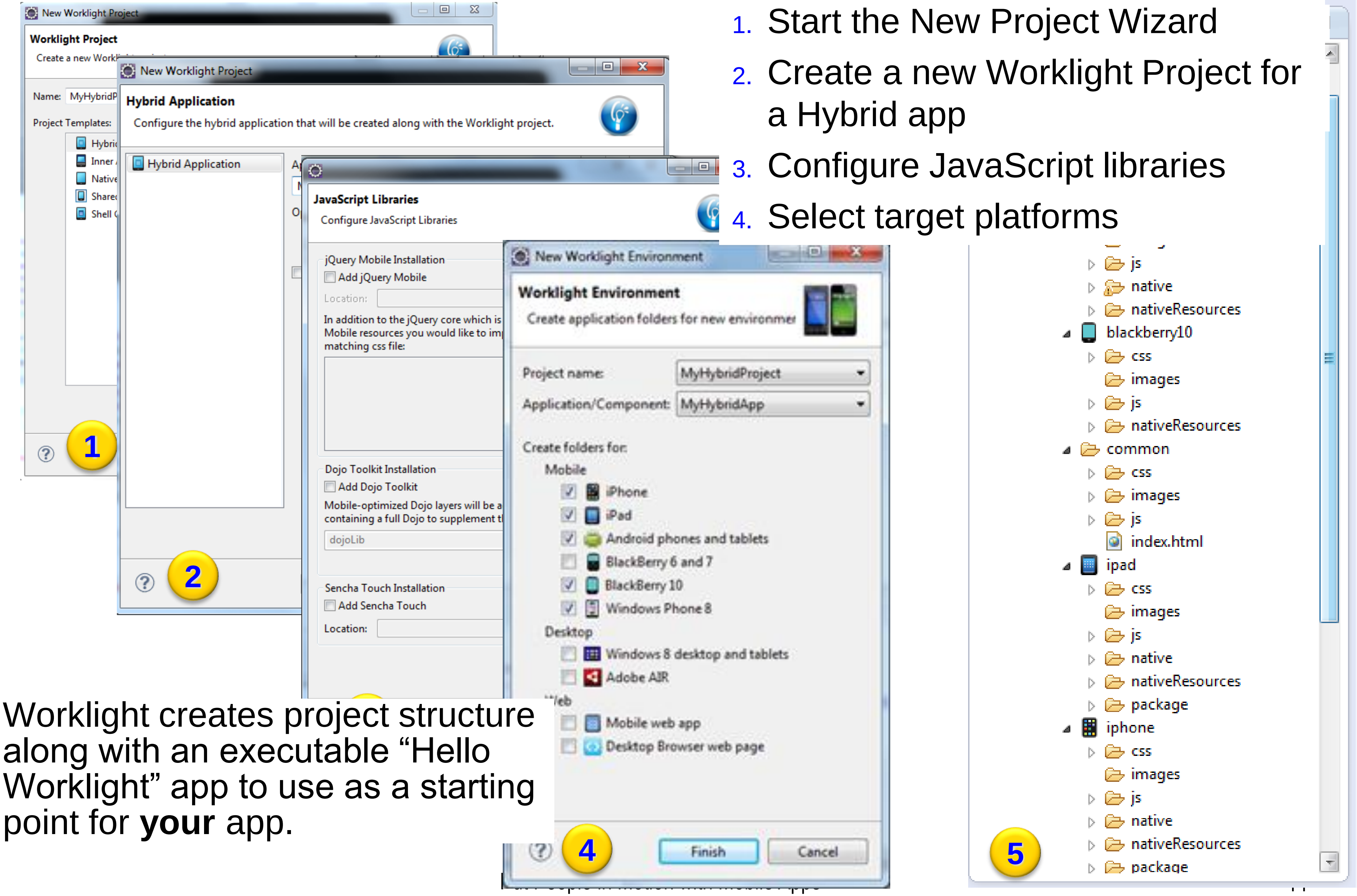
These errors were caused by a component that was not listed as required

- ANT – for build (had to download and install)
- Android Target – for Android emulator (use Eclipse to create target)
- GitHub – for installing Cordova device plug-in (had to download and install)

After installation, we still got the error message “navigator.camera.undefined”

- Several hours of Web searching did not resolve this problem
- This extra time was not included in our timing results
- Eventually the developer uninstalled Cordova 3.3, installed PhoneGap 3.3, and **then** it worked, but these two releases should be the same

Worklight New Project Wizard quickly creates a new sample application as a starting point



New Worklight Project

Worklight Project

Create a new Workl

Name: MyHybridP

Project Templates:

- Hybri
- Inner
- Native
- Share
- Shell

New Worklight Project

Hybrid Application

Configure the hybrid application that will be created along with the Worklight project.

Hybrid Application

JavaScript Libraries

Configure JavaScript Libraries

jQuery Mobile Installation

□ Add jQuery Mobile

Location:

In addition to the jQuery core which is
Mobile resources you would like to im
matching css file:

Dojo Toolkit Installation

□ Add Dojo Toolkit

Mobile-optimized Dojo layers will be a
containing a full Dojo to supplement t

dojoLib

Sencha Touch Installation

□ Add Sencha Touch

Location:

New Worklight Environment

Worklight Environment

Create application folders for new environmer

Project name: MyHybridProject

Application/Component: MyHybridApp

Create folders for:

Mobile
- ☑ iPhone
- ☑ iPad
- ☑ Android phones and tablets
- □ BlackBerry 6 and 7
- ☑ BlackBerry 10
- ☑ Windows Phone 8

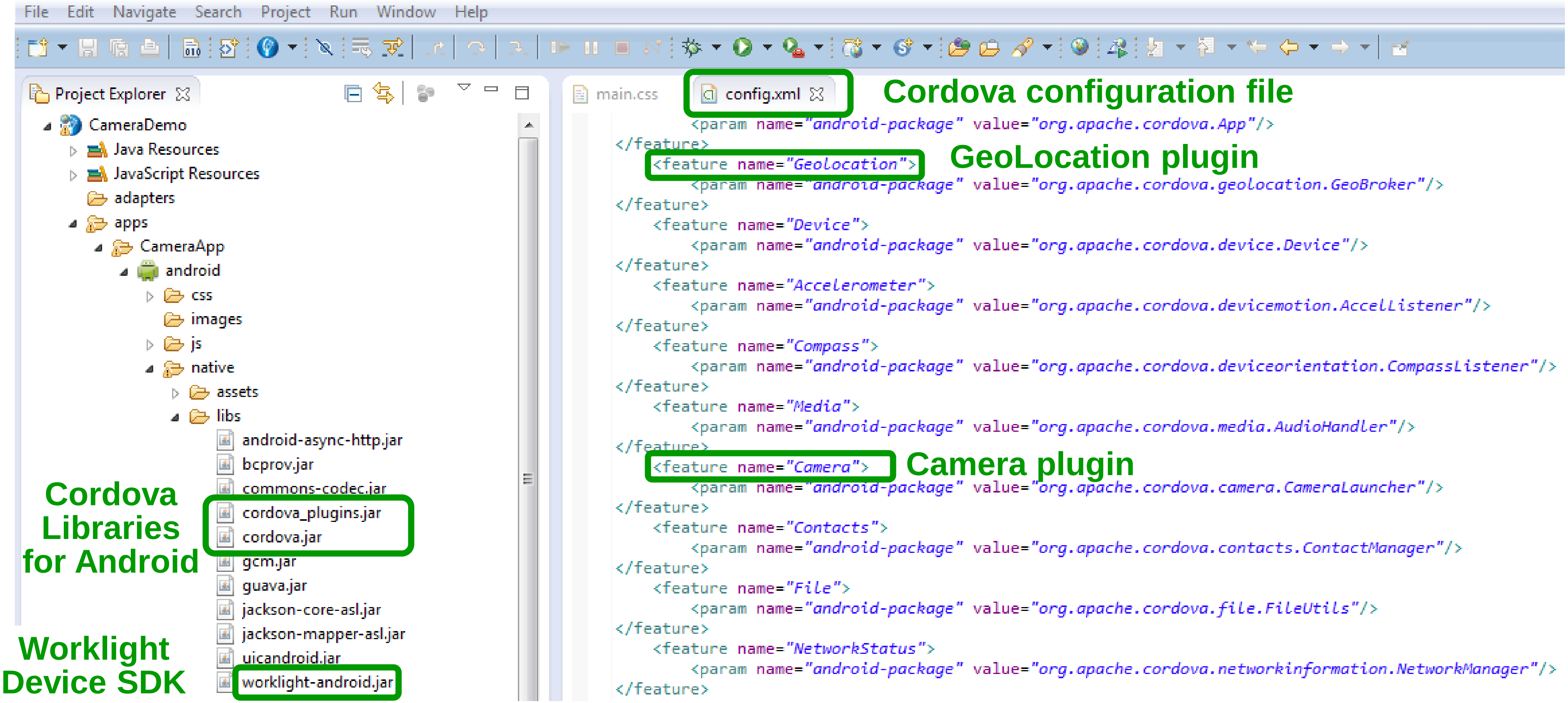
Desktop
- □ Windows 8 desktop and tablets
- □ Adobe AIR

Web
- □ Mobile web app
- □ Desktop Browser web page

```
    ▷ js
    ▷ native
    ▷ nativeResources
◢ blackberry10
    ▷ css
      images
    ▷ js
    ▷ nativeResources
◢ common
    ▷ css
    ▷ images
    ▷ js
      index.html
◢ ipad
    ▷ css
      images
    ▷ js
    ▷ native
    ▷ nativeResources
    ▷ package
◢ iphone
    ▷ css
      images
    ▷ js
    ▷ native
    ▷ nativeResources
    ▷ package
```

1. Start the New Project Wizard
2. Create a new Worklight Project for a Hybrid app
3. Configure JavaScript libraries
4. Select target platforms
5. Worklight creates project structure along with an executable “Hello Worklight” app to use as a starting point for **your** app.

Worklight integrates Android-specific Cordova and Worklight SDKs directly into project



```
CameraDemo
  Java Resources
  JavaScript Resources
  adapters
  apps
    CameraApp
      android
        css
        images
        js
        native
          assets
          libs
            android-async-http.jar
            bcprov.jar
            commons-codec.jar
            cordova_plugins.jar
            cordova.jar
            gcm.jar
            guava.jar
            jackson-core-asl.jar
            jackson-mapper-asl.jar
            uicandroid.jar
            worklight-android.jar
```

```
        <param name="android-package" value="org.apache.cordova.App"/>
</feature>
    <feature name="Geolocation">
        <param name="android-package" value="org.apache.cordova.geolocation.GeoBroker"/>
</feature>
    <feature name="Device">
        <param name="android-package" value="org.apache.cordova.device.Device"/>
</feature>
    <feature name="Accelerometer">
        <param name="android-package" value="org.apache.cordova.devicemotion.AccelListener"/>
</feature>
    <feature name="Compass">
        <param name="android-package" value="org.apache.cordova.deviceorientation.CompassListener"/>
</feature>
    <feature name="Media">
        <param name="android-package" value="org.apache.cordova.media.AudioHandler"/>
</feature>
    <feature name="Camera">
        <param name="android-package" value="org.apache.cordova.camera.CameraLauncher"/>
</feature>
    <feature name="Contacts">
        <param name="android-package" value="org.apache.cordova.contacts.ContactManager"/>
</feature>
    <feature name="File">
        <param name="android-package" value="org.apache.cordova.file.FileUtils"/>
</feature>
    <feature name="NetworkStatus">
        <param name="android-package" value="org.apache.cordova.networkinformation.NetworkManager"/>
</feature>
```

Cordova configuration file

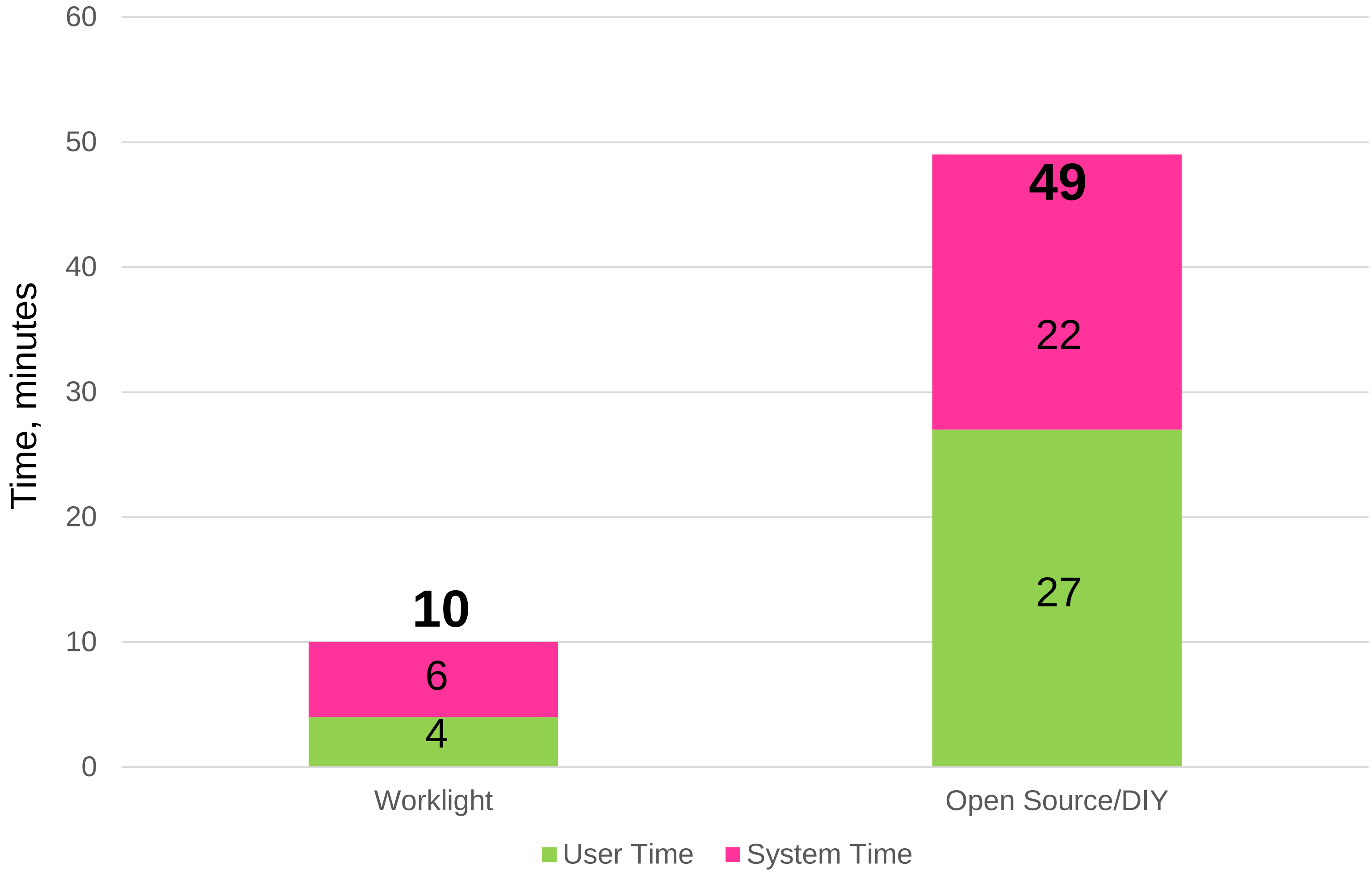
GeoLocation plugin

Camera plugin

Cordova Libraries for Android

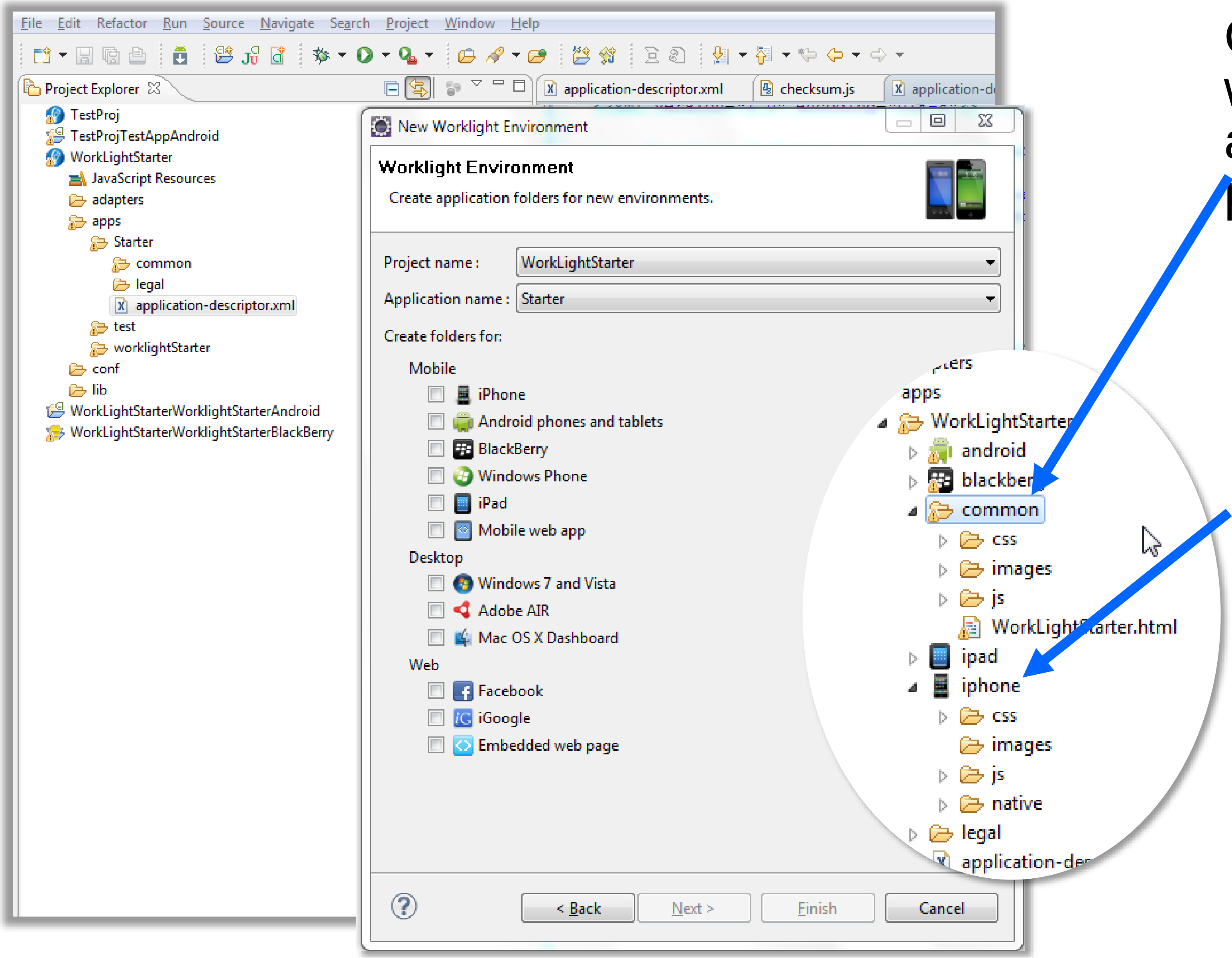
Worklight Device SDK for Android

Creating an executable Hello World project is 4.9x faster with Worklight than DIY



Study subject was the same experienced software engineer with open source experience

Worklight Studio helps you write once but deploy to many devices using a common project



File Edit Refactor Run Source Navigate Search Project Window Help
Project Explorer
TestProj
TestProjTestAppAndroid
WorkLightStarter
JavaScript Resources
adapters
apps
Starter
common
legal
application-descriptor.xml
test
worklightStarter
conf
lib
WorkLightStarterWorklightStarterAndroid
WorkLightStarterWorklightStarterBlackBerry
application-descriptor.xml
checksum.js
application-d
New Worklight Environment

Worklight Environment

Create application folders for new environments.

Project name : WorkLightStarter

Application name : Starter

Create folders for:

Mobile

- ☐ iPhone
- ☐ Android phones and tablets
- ☐ BlackBerry
- ☐ Windows Phone
- ☐ iPad
- ☐ Mobile web app

Desktop

- ☐ Windows 7 and Vista
- ☐ Adobe AIR
- ☐ Mac OS X Dashboard

Web

- ☐ Facebook
- ☐ iGoogle
- ☐ Embedded web page

Common code will be used for apps on all platforms

Platform-specific code is maintained separately

Worklight provides comprehensive mobile security in the app and on the server

Secure Applications

- Worklight Server is based on IBM WebSphere, providing proven security
- Supports Server Identity Verification with SSL, preventing man-in-the-middle attacks
  - WebSphere never had the Heartbleed Bug
- Encrypt application code and resources to prevent tampering
- Server authentication can use existing custom code, database, or enterprise directory

Device Data Protection

- Can use an optional offline data cache, encrypted with PKCS and AES standards
- When not connected, optional offline authentication protects the app and its data
- App and device authenticity ensures that apps are used only on authorized devices

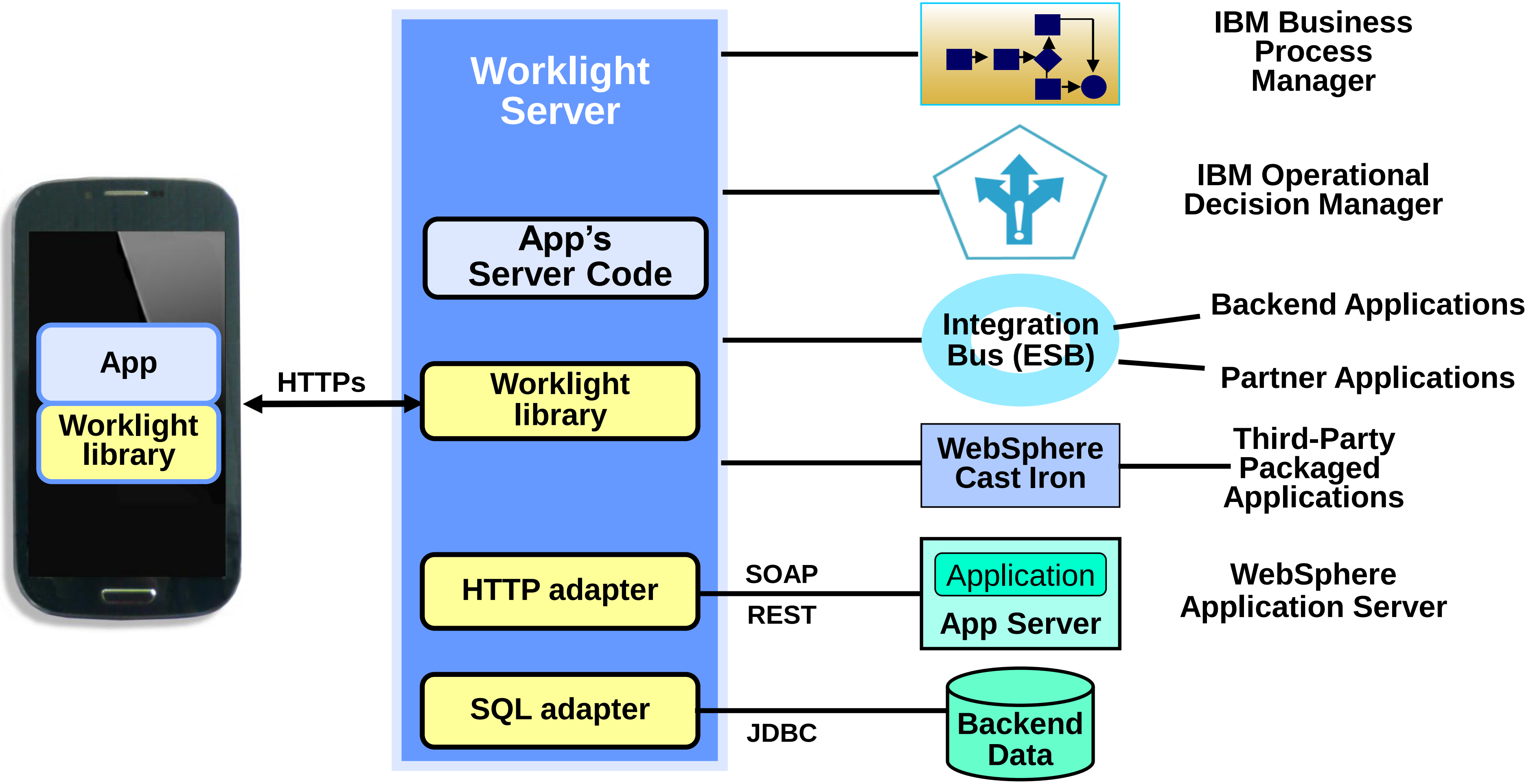
Robust Authentication and Authorization

- Server-side authentication is based on JAAS and client-side framework
- Resources are protected by authentication realms
  - When a user accesses a resource, Worklight checks user’s authentication, and triggers re-authentication if required
- Unique IDs identify devices to Worklight Server

Enforce Security Updates

- Administrators can remotely disable old versions of an app
- Can force an update to an app next time the app is started

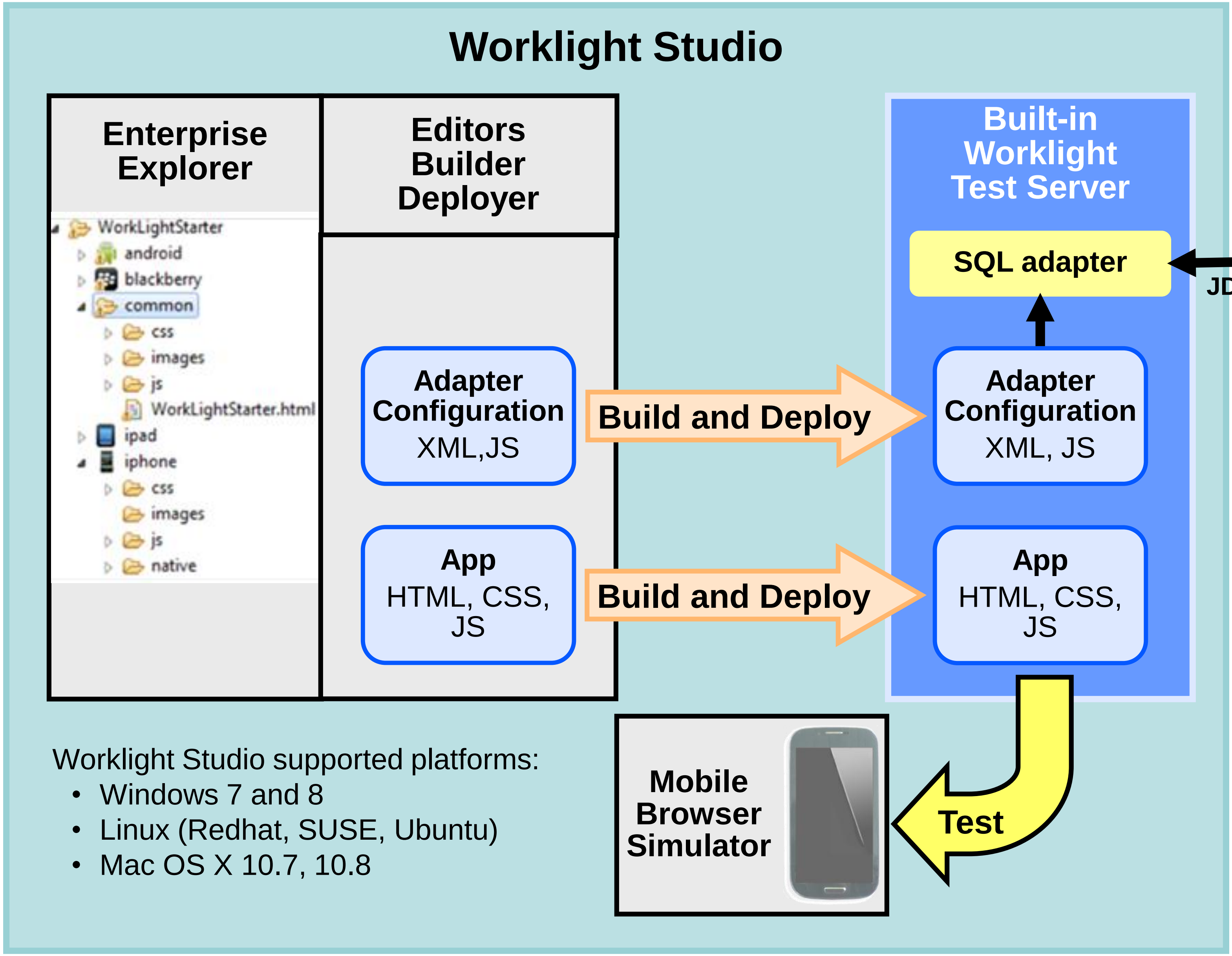
Worklight Adapter Framework provides secure backend access



- Adapters run on Worklight Server, not the mobile device, for better security
- Information about the backend systems (URL or DB name, credentials, etc.) is only stored on the Worklight server, not the mobile device

More control, better security, fewer server connections, less app code

DEMO: Worklight Studio creates a single, integrated project for the mobile programming model

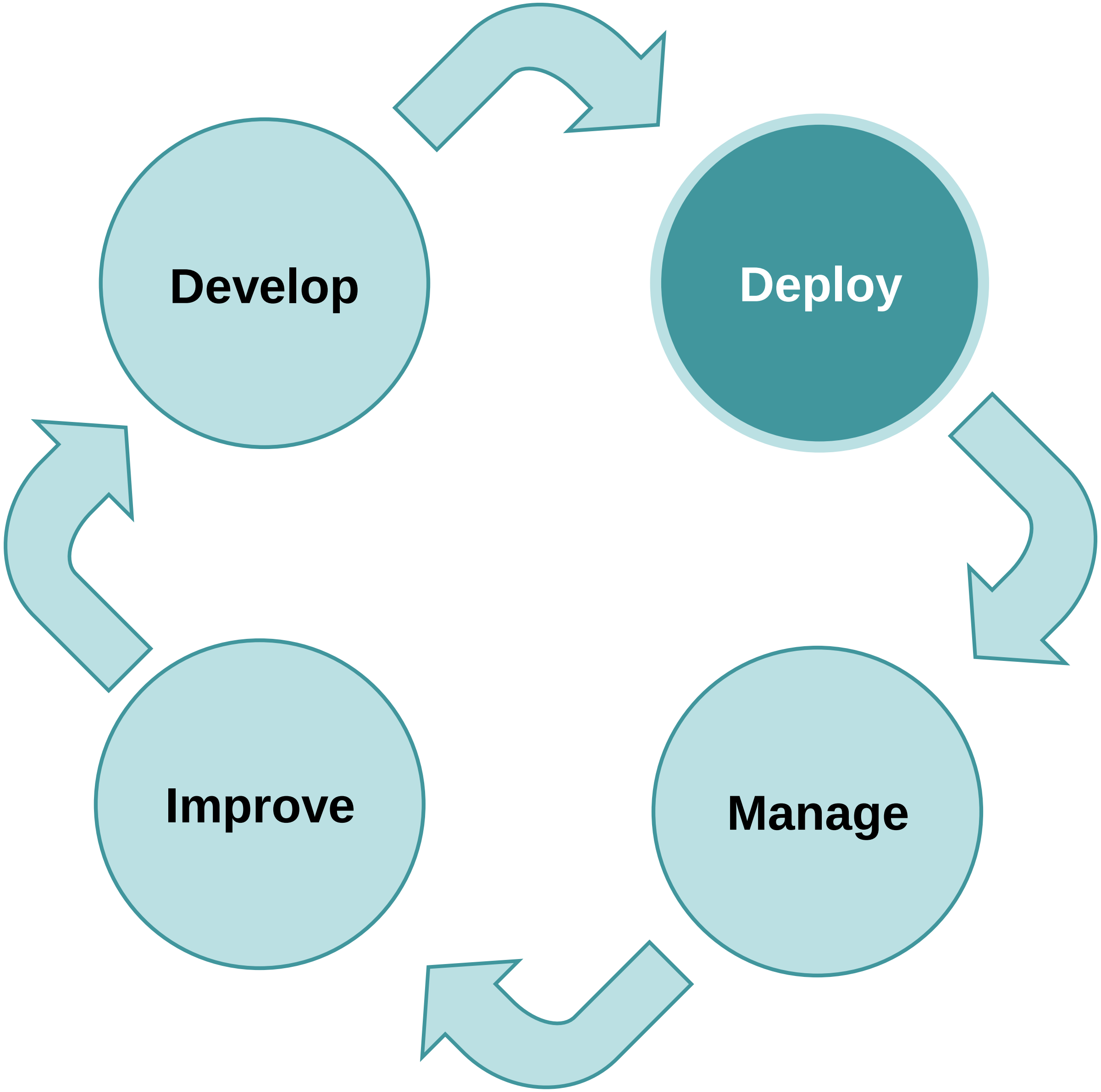


Worklight Studio supported platforms:

- Windows 7 and 8
- Linux (Redhat, SUSE, Ubuntu)
- Mac OS X 10.7, 10.8

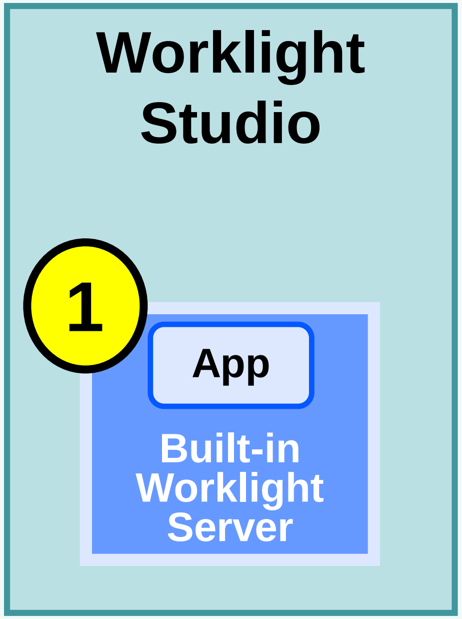
- Edit and Manage source code
- Integrate app with database
- Build and test

Deploy and test using a built-in server, a standalone server, or a cloud-based server



Worklight has multiple deployment options

Development



Worklight Studio

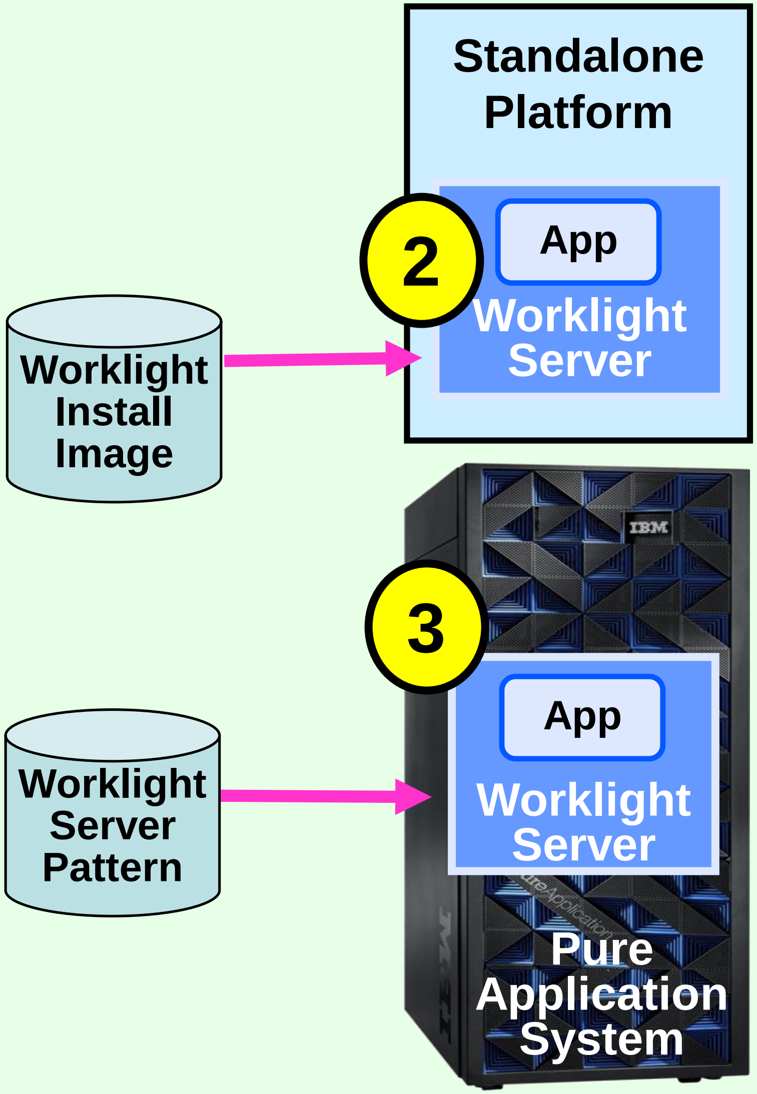
1

App

Built-in Worklight Server

- Worklight Studio deploys app to built-in Worklight Server

Production



Standalone Platform

2

App

Worklight Server

Worklight Install Image

3

App

Worklight Server

Pure Application System

Worklight Server Pattern

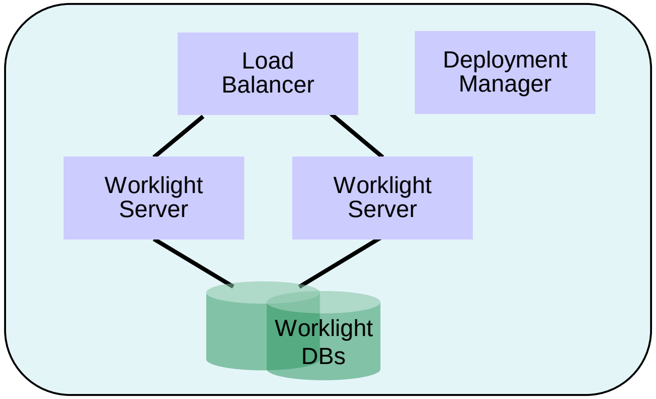
- Worklight Server is installed from an install image
- The app is deployed by Worklight Studio

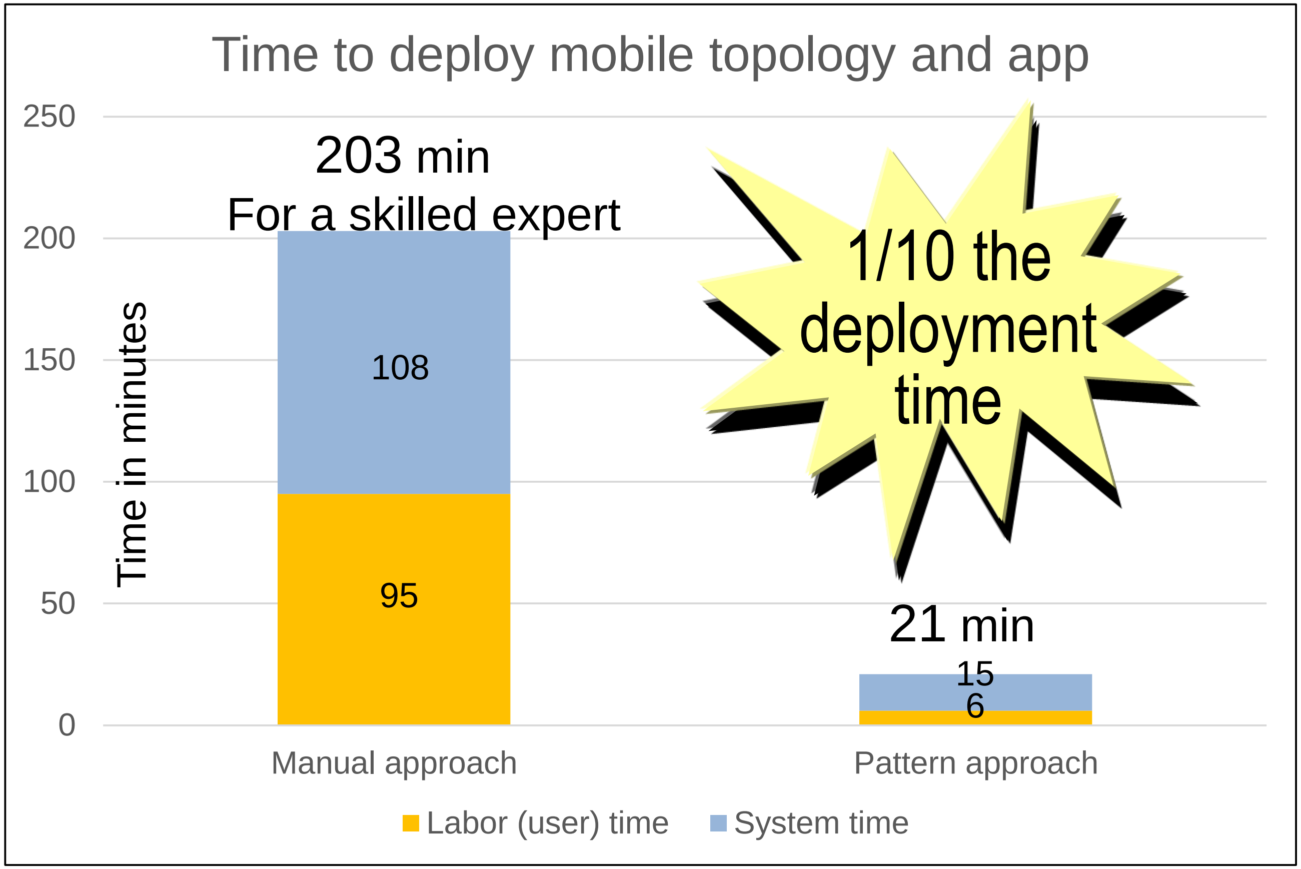
- Worklight Mobile Pattern and all app components are created and installed easily using Worklight Studio or base Mobile Pattern

Fastest, and least skill required.

Mobile Application Pattern has 10% of the deployment time, while reducing risk

Deploy a mobile app to an IBM Worklight high availability production environment





If you don’t have an expert, expect as much as 100x more time needed for manual approach

- Learning time adds significant labor time to the manual work effort
- Coordination between multiple people with deep application server, database, mobile, OS skills
- Testing efforts to ensure each step was done properly
- Risk of errors due to manual labor

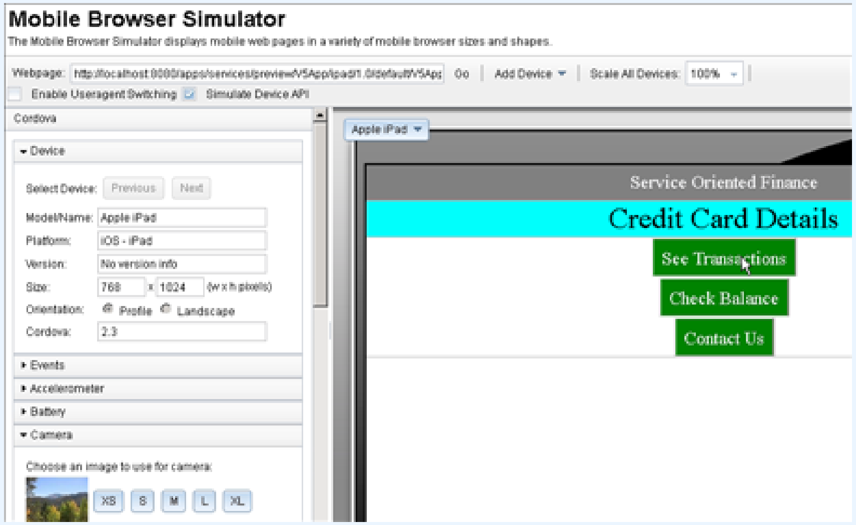
Worklight makes testing easy using integrated browser simulator or manufacturer’s emulators

Worklight provides a **Mobile Browser Simulator** for many kinds of devices

- No need to download emulators
- Preview and test Web content in the app, even iPhone and iPad on a non-Apple computer
  - Android, iPhone, iPad, BlackBerry 6 and 7, Windows Phone 7 and 8, and mobile Web app environments
- Simulate device-specific capabilities
  - Camera, Accelerometer, Geolocation, Events, Battery status, Capture, Compass, Contacts, File, Network

Worklight supports the manufacturer’s **emulator** from their SDK

- Preview platform fidelity using manufacturer’s emulator
  - Android, Blackberry, Windows 8 (Metro), Windows Phone, iOS (on Mac only)
  - Truest view of device without using an actual device
- Debug native parts of the app (capabilities varies with emulators)



Mobile Browser Simulator

The Mobile Browser Simulator displays mobile web pages in a variety of mobile browser sizes and shapes.

Webpage: http://localhost:8080/apps/services/preview/V5App/ipad/1.0/default/V5Apr

☐ Enable Useragent Switching ☑ Simulate Device API

Select Device: Previous Next

Model/Name: Apple iPad

Platform: iOS - iPad

Version: No version info

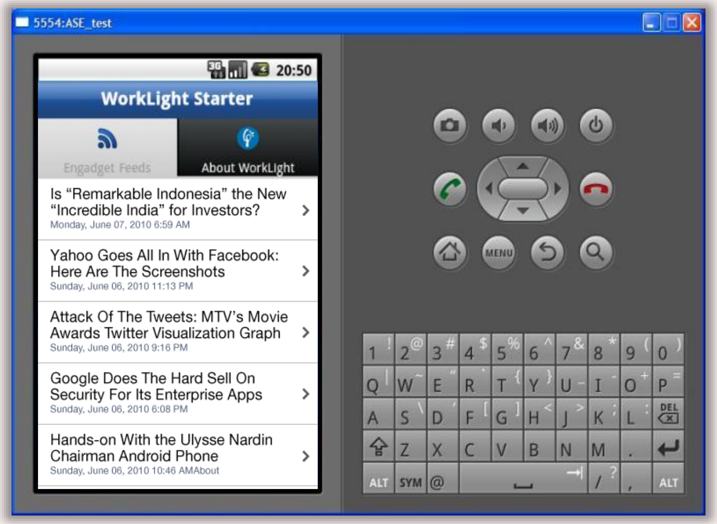
Size: 768 x 1024 (w x h pixels)

Orientation: ● Profile ○ Landscape

Cordova: 2.3

Choose an image to use for camera:

Worklight Mobile Browser Simulator
(shown here simulating an iPad)



5554:ASE_test

WorkLight Starter

Is “Remarkable Indonesia” the New “Incredible India” for Investors?
Monday, June 07, 2010 6:59 AM

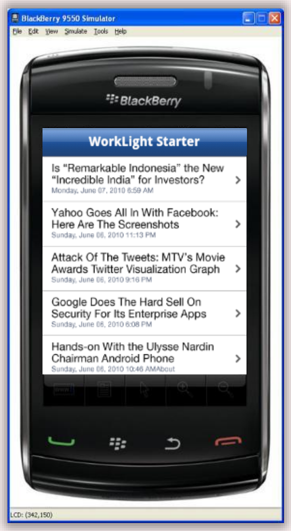
Yahoo Goes All In With Facebook: Here Are The Screenshots
Sunday, June 06, 2010 11:13 PM

Attack Of The Tweets: MTV’s Movie Awards Twitter Visualization Graph
Sunday, June 06, 2010 9:16 PM

Google Does The Hard Sell On Security For Its Enterprise Apps
Sunday, June 06, 2010 6:08 PM

Hands-on With the Ulysse Nardin Chairman Android Phone
Sunday, June 06, 2010 10:46 AMAbout

Google Android Emulator



BlackBerry 9550 Simulator

File Edit View Simulate Tools Help

WorkLight Starter

Is “Remarkable Indonesia” the New “Incredible India” for Investors?
Monday, June 07, 2010 6:59 AM

Yahoo Goes All In With Facebook: Here Are The Screenshots
Sunday, June 06, 2010 11:13 PM

Attack Of The Tweets: MTV’s Movie Awards Twitter Visualization Graph
Sunday, June 06, 2010 9:16 PM

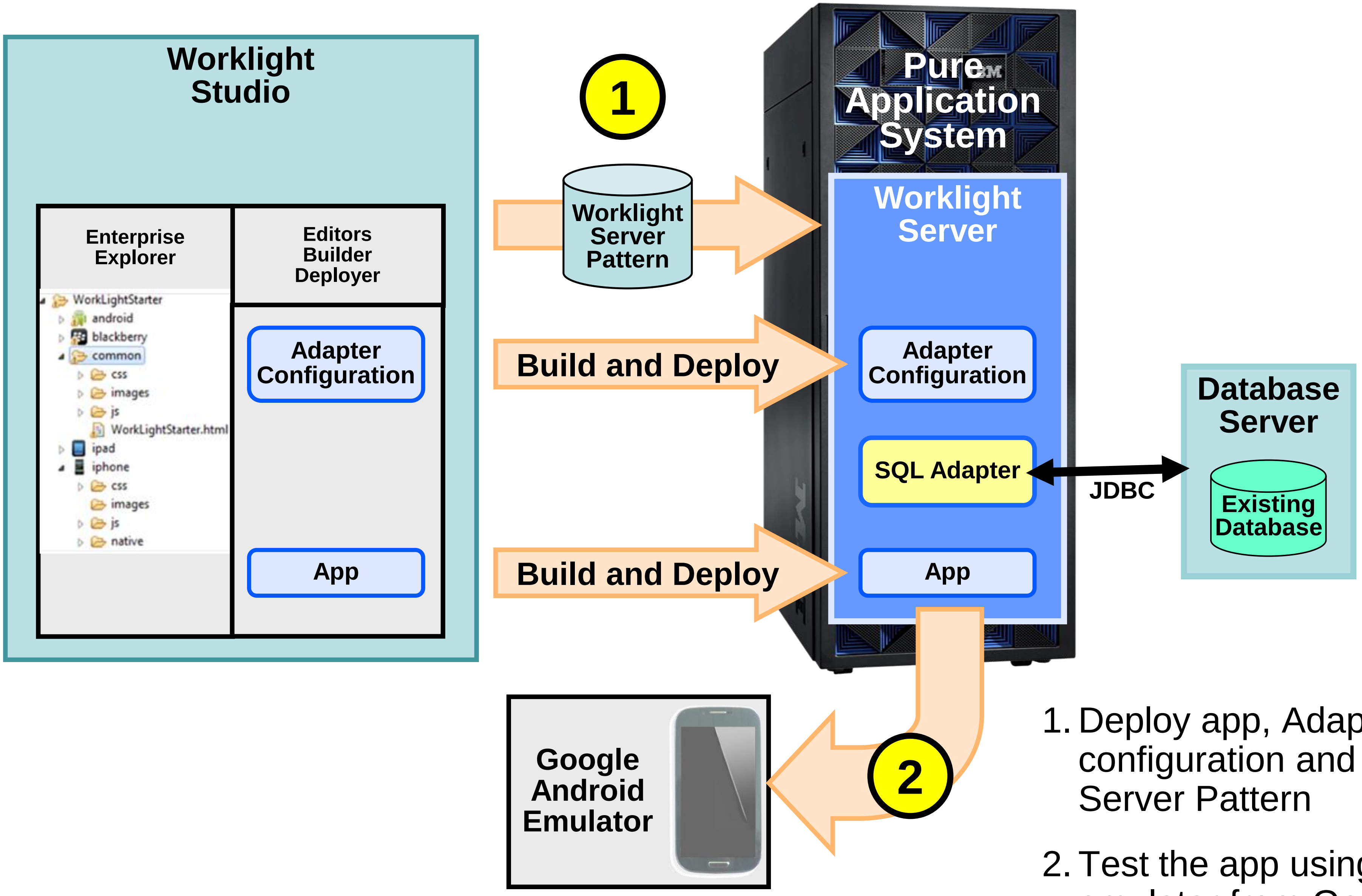
Google Does The Hard Sell On Security For Its Enterprise Apps
Sunday, June 06, 2010 6:08 PM

Hands-on With the Ulysse Nardin Chairman Android Phone
Sunday, June 06, 2010 10:46 AMAbout

LCD: (342,150)

RIM Blackberry Emulator

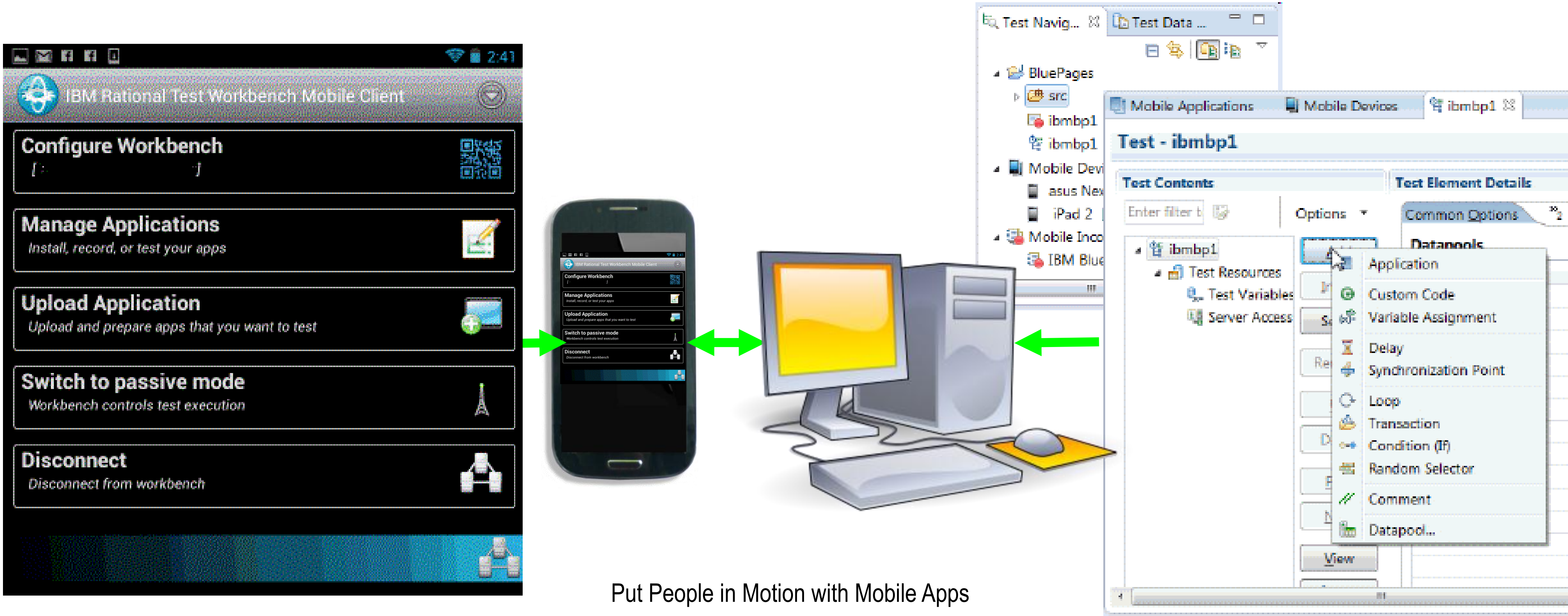
DEMO: Worklight Studio makes it easy to deploy a Worklight Server to the cloud



1. Deploy app, Adapter configuration and Worklight Server Pattern
2. Test the app using Android emulator from Google SDK

Mobile Test Workbench for Worklight automates testing of Worklight apps for Android and iOS

- Automate the creation, execution, and analysis of functional tests using scripts
- Based on Rational Test Workbench and Eclipse platform
- The Mobile Test Workbench client is used to upload apps to the Test Workbench, to record and run test scripts, and to view reports
- Can be used to test both native and hybrid apps created by Worklight Studio for Android, iOS 6.x, and Windows Phone 8



2:41

IBM Rational Test Workbench Mobile Client

Configure Workbench

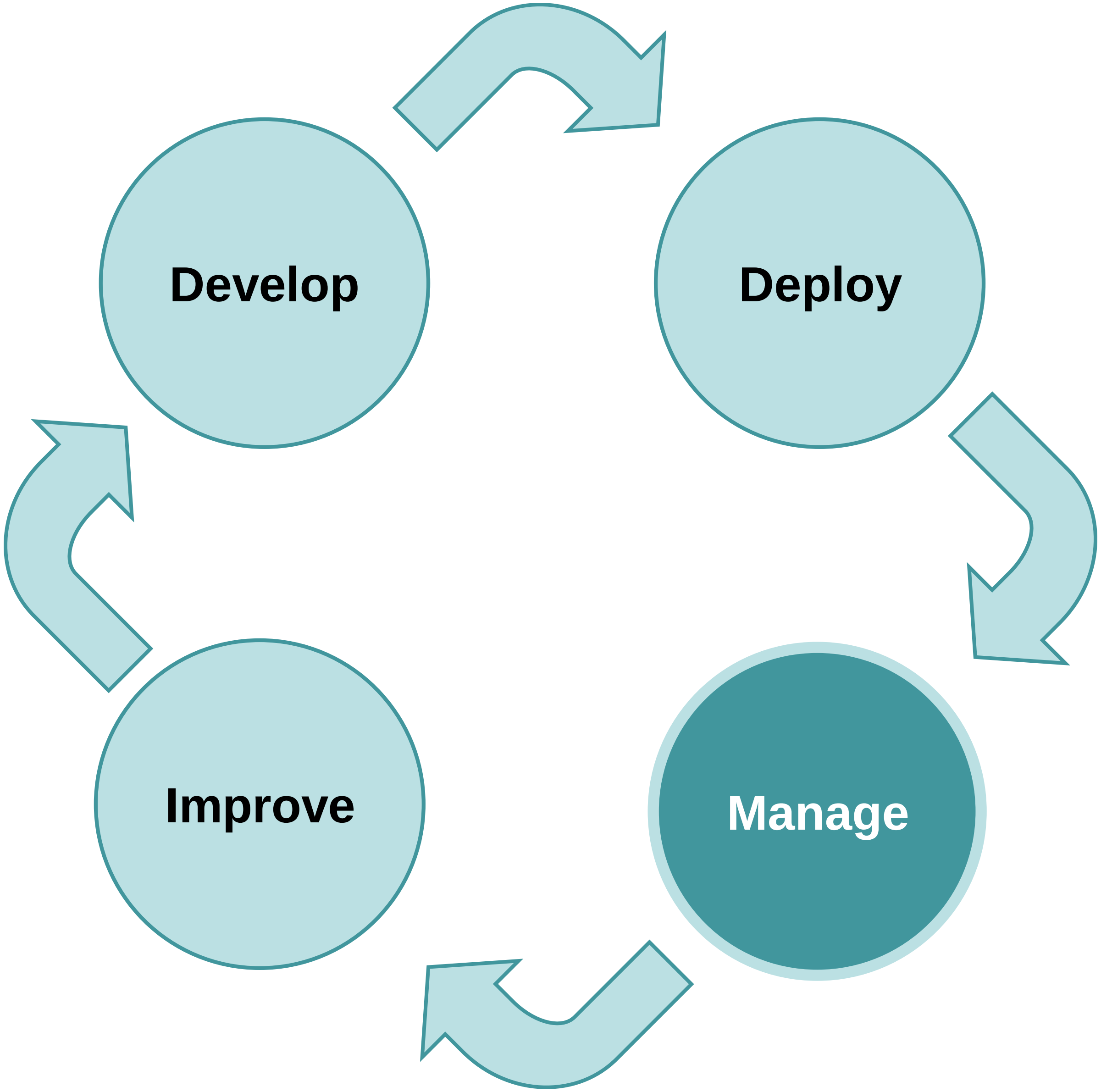
Manage Applications
Install, record, or test your apps

Upload Application
Upload and prepare apps that you want to test

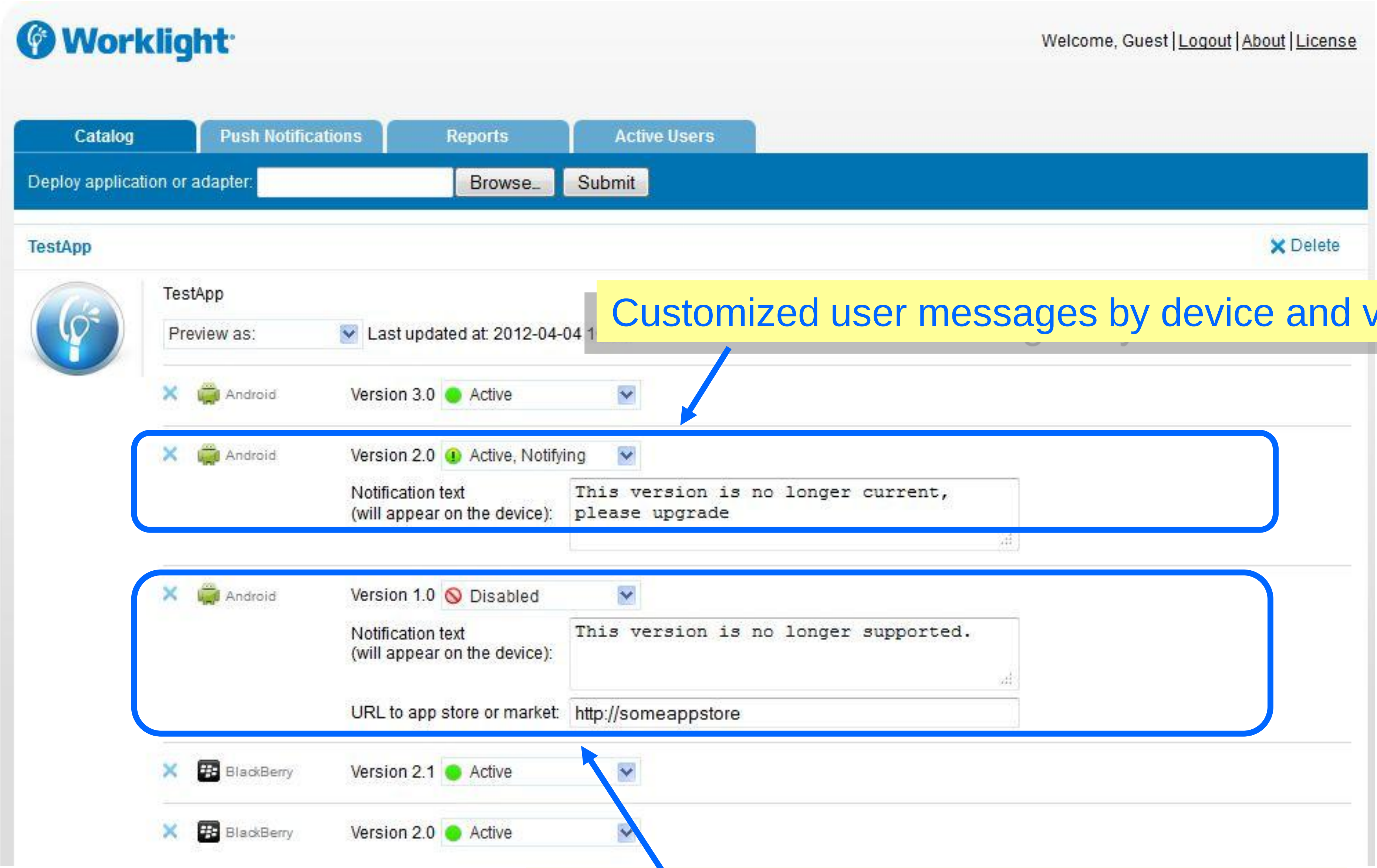
Switch to passive mode
Workbench controls test execution

Disconnect
Disconnect from workbench

Manage assets on corporate devices, make apps available on company store, and control app versions



IBM Worklight Console provides version management for deployed mobile apps



Worklight

Welcome, Guest | Logout | About | License

Deploy application or adapter: Browse... Submit

TestApp ✕ Delete

TestApp

Preview as: Last updated at: 2012-04-04 1

✕ Android Version 3.0 ● Active

✕ Android Version 2.0 ● Active, Notifying

Notification text (will appear on the device): This version is no longer current, please upgrade

✕ Android Version 1.0 ⊘ Disabled

Notification text (will appear on the device): This version is no longer supported.

URL to app store or market: http://someappstore

✕ BlackBerry Version 2.1 ● Active

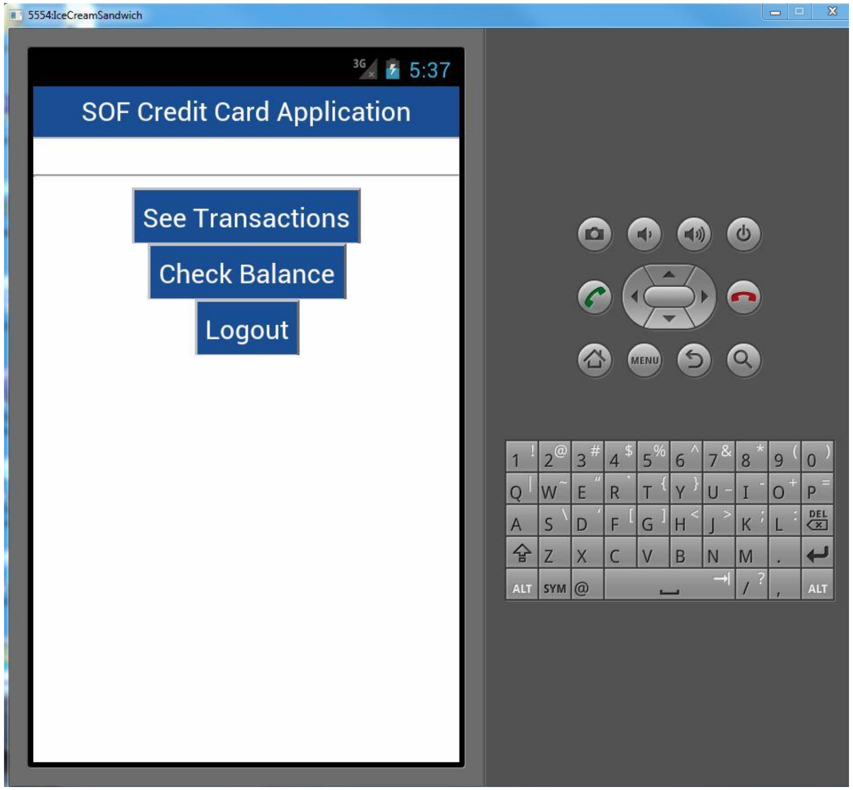
✕ BlackBerry Version 2.0 ● Active

Customized user messages by device and version

Remotely disable apps by device and version

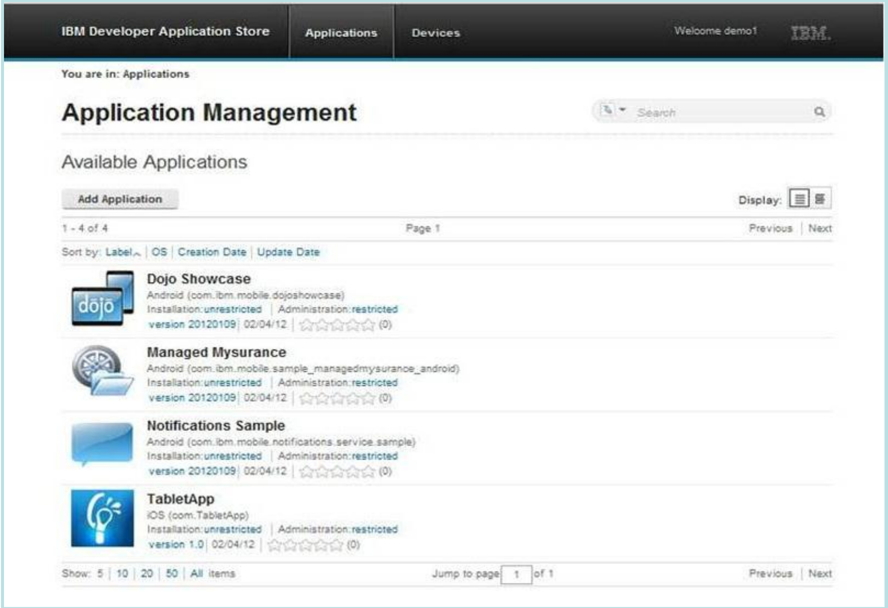
DEMO: Managing app changes

- Use Worklight Studio to make a change
- Use Worklight Console to upload the new binary
- Use Android Emulator to show the result



Worklight Application Center is a private enterprise app store

- Install, configure, and administer mobile apps for use by individuals and groups within your enterprise
- Similar to Google Play or iTunes Store
- Supports app management
  - Access control lists specify who can access and upload specific apps by user or group names
  - Collect user feedback
  - Learn which devices your apps have been installed on
- Provides a mobile client app to allow users to browse the catalog of apps
- Includes the Application Center Console to administer deployed apps



You are in: Applications

Application Management

Available Applications

1 - 4 of 4 | Page 1 | Previous | Next

Sort by: Label | OS | Creation Date | Update Date

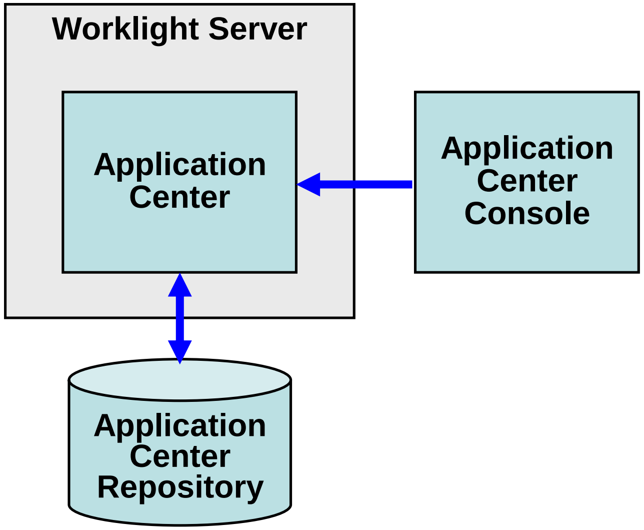
Dojo Showcase
Android (com.ibm.mobile.dojoshowcase)
Installation:unrestricted | Administration:restricted
version 20120109 | 02/04/12 | ☆☆☆☆☆ (0)

Managed Mysurance
Android (com.ibm.mobile.sample_managedmysurance_android)
Installation:unrestricted | Administration:restricted
version 20120109 | 02/04/12 | ☆☆☆☆☆ (0)

Notifications Sample
Android (com.ibm.mobile.notifications.service.sample)
Installation:unrestricted | Administration:restricted
version 20120109 | 02/04/12 | ☆☆☆☆☆ (0)

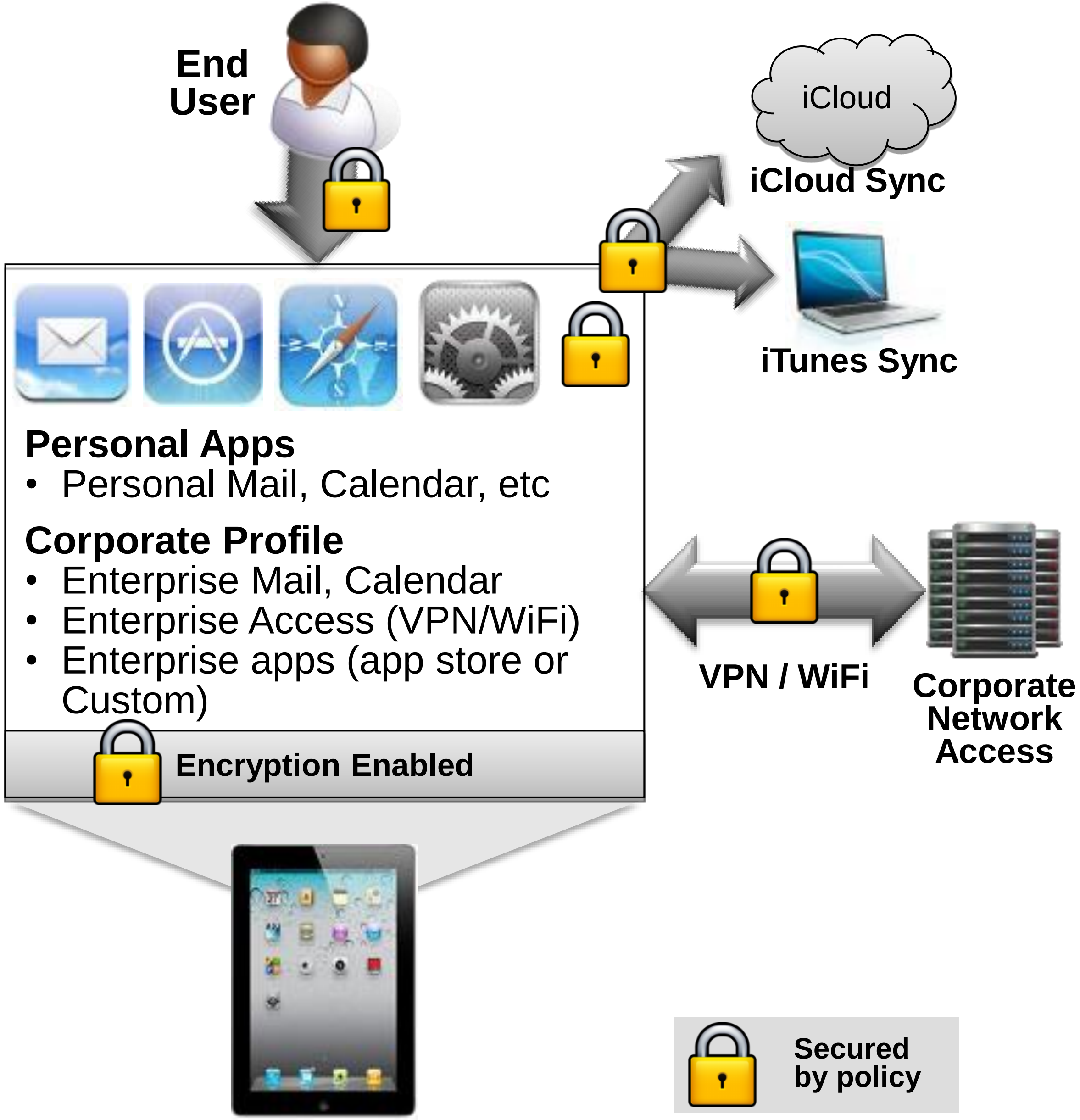
TabletApp
iOS (com.TabletApp)
Installation:unrestricted | Administration:restricted
version 1.0 | 02/04/12 | ☆☆☆☆☆ (0)

Show: 5 | 10 | 20 | 50 | All items | Jump to page 1 of 1 | Previous | Next



IBM Endpoint Manager for mobile devices provides policy-based control

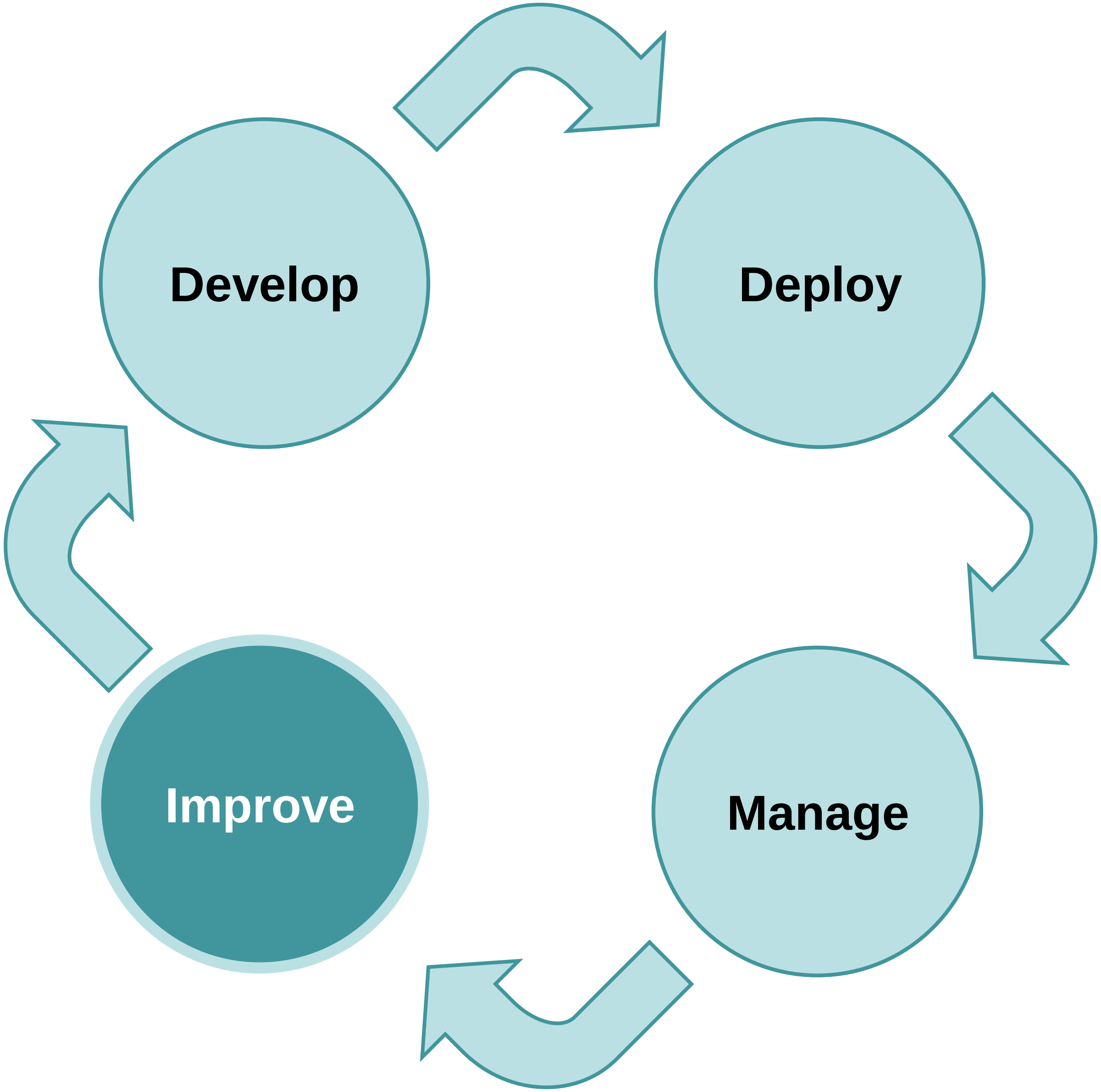
- Enforce policy-based management of devices
  - Enable password policies
  - Enable device encryption
  - Force encrypted backup
  - Disable iCloud sync
  - Disable access to corporate email, apps, VPN, WiFi if device is not compliant with policies
- Protect corporate data
  - Selectively wipe corporate data if employee leaves company
  - Fully wipe device if lost or stolen



Personal Apps
- Personal Mail, Calendar, etc

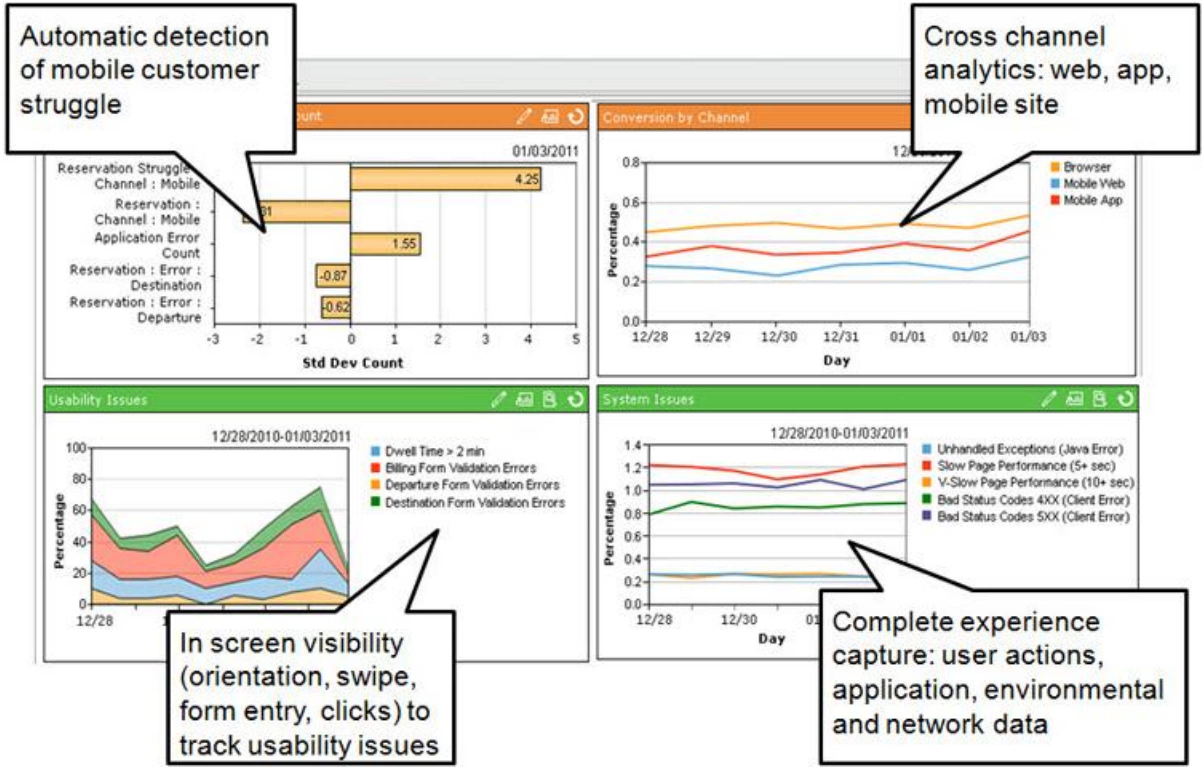
Corporate Profile
- Enterprise Mail, Calendar
- Enterprise Access (VPN/WiFi)
- Enterprise apps (app store or Custom)

Improve your app by studying actual usage patterns and eliminating user problems



Improve the mobile user experience by analyzing usage patterns and problems

- **Gain visibility** by capturing user interactions
  - Touch-screen gestures such as swiping, zooming, scrolling and device orientation
  - Replay the results to gain understanding
- **Automatically detect** app failures and usability issues
  - Find groups of users that have the same problem
  - Learn what they have in common


ount
01/03/2011
Reservation Struggle Channel : Mobile
Reservation : Channel : Mobile
Application Error Count
Reservation : Error : Destination
Reservation : Error : Departure
4.25
31
1.55
-0.87
-0.62
-3
-2
-1
0
1
2
3
4
5
Std Dev Count
Conversion by Channel
12/
0.8
0.6
0.4
0.2
0.0
Percentage
12/28
12/29
12/30
12/31
01/01
01/02
01/03
Day
Browser
Mobile Web
Mobile App
Usability Issues
12/28/2010-01/03/2011
100
80
60
40
20
0
Percentage
12/28
Dwell Time > 2 min
Billing Form Validation Errors
Departure Form Validation Errors
Destination Form Validation Errors
System Issues
12/28/2010-01/03/2011
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Percentage
12/28
12/30
01
Day
Unhandled Exceptions (Java Error)
Slow Page Performance (5+ sec)
V-Slow Page Performance (10+ sec)
Bad Status Codes 4XX (Client Error)
Bad Status Codes 5XX (Client Error)

Automatic detection of mobile customer struggle

Cross channel analytics: web, app, mobile site

In screen visibility (orientation, swipe, form entry, clicks) to track usability issues

Complete experience capture: user actions, application, environmental and network data

- **Discover what to fix** to avoid failed transactions, abandonment, poor app store rankings and negative feedback
- **Quantify revenue impact and segmentation** by analyzing specific mobile user behaviors or device attributes
  - Segment and slice the data by device
  - View cross-channel analytics to compare Web, app, mobile sites

tealeaf®
CX Mobile

Example: Tealeaf CX Mobile helps determine the source of mobile app usage problems

- You set up Tealeaf to capture session data and send alerts for app failures and abandonment
- You receive a Tealeaf alert indicating an app problem: user receives an unexpected error message
- Using Tealeaf to examine statistics, you learn that it works reliably for 97% of users, but fails for 3%
- Choosing a failed session, you replay user interactions, see the screen as the user saw it
- You see that other users had the same problem
- Analyzing captured data, you discover that all users with the problem were using the same credit card
- You report the problem to developers
- Developers can reproduce the problem, significantly cutting resolution time

Solve the challenges of building and managing mobile apps with IBM's mobile solutions

Reduce development and maintenance costs for creating and managing mobile apps on a variety of device platforms with IBM Worklight Studio

Simplify security and quickly integrate apps to back-end systems using IBM Worklight Server

Choose immediate update or update when convenient for new versions of apps with Worklight Console

Gain ability to selectively or totally wipe lost devices to protect corporate assets with IBM Endpoint Manager

Analyze actual customer usage patterns to find problem areas for continuous improvement with IBM Tealeaf