Business Agility in Action.

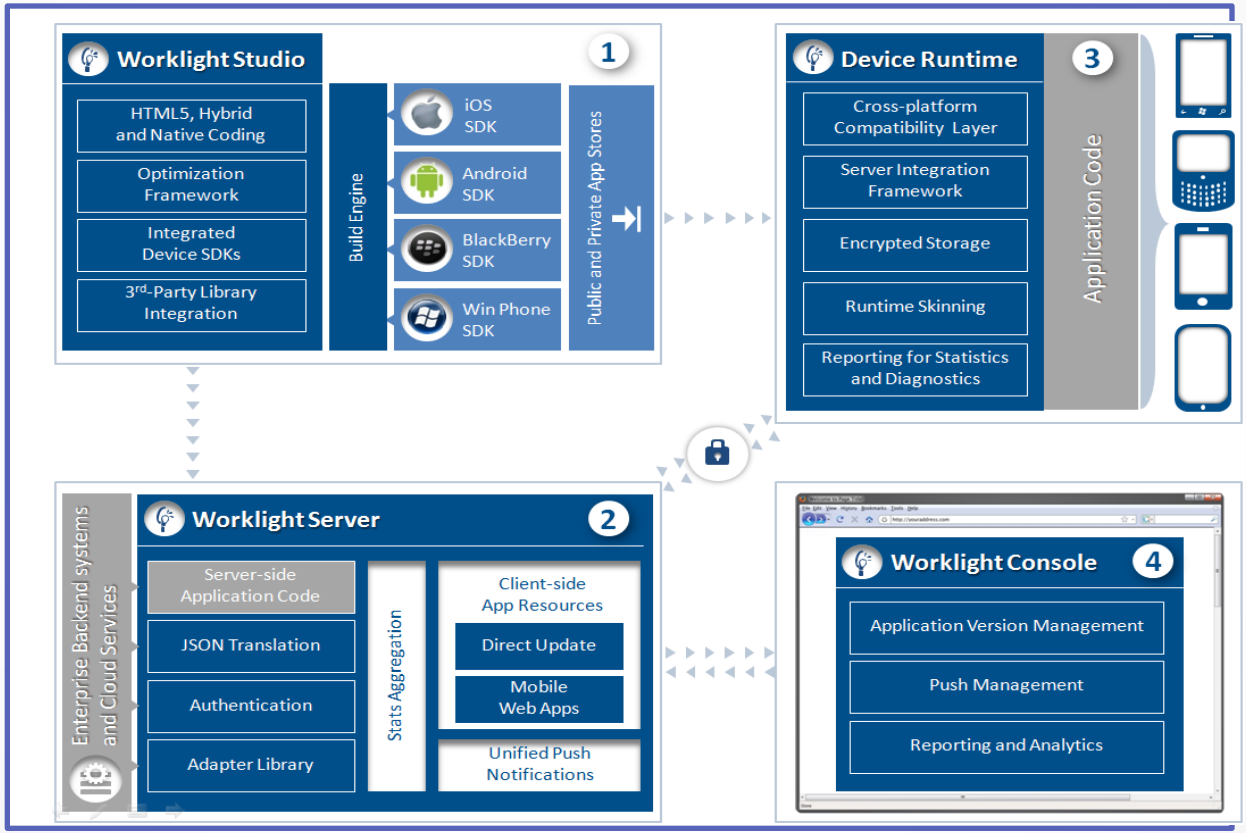
Innovate. Transform. Grow

IBM Worklight Technical Deep Dive

David Schlosser – IBM Technical Sales, Mobile/Worklight

schlossd@au1.ibm.com

Worklight Components



Worklight Studio

Eclipse/RAD Based IDE

- Native, hybrid and standard web development
- Open, familiar, standardized tooling

Environment-specific optimization

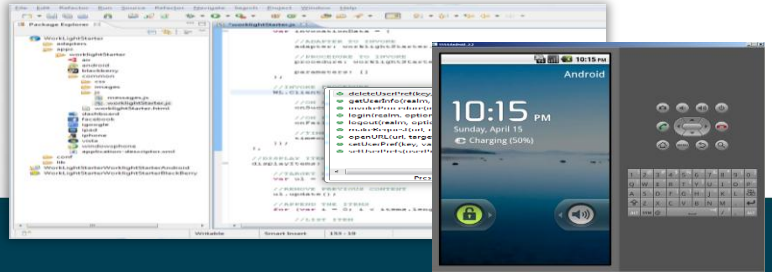
- Common code with overrides/extensions
- Native development options
- Runtime skins

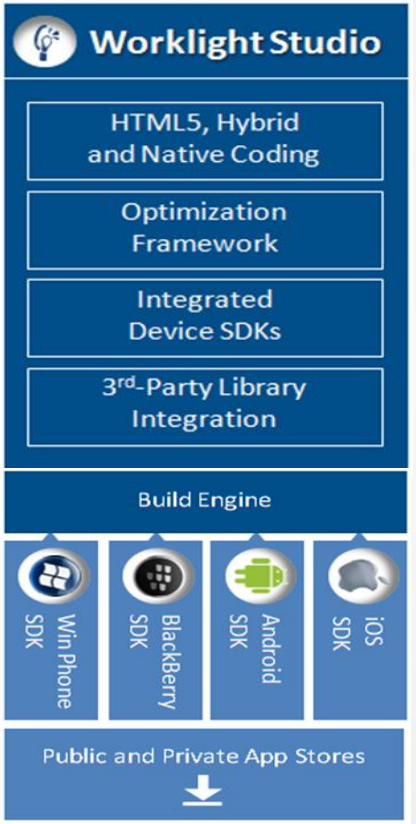
3rd-party library integration

Device SDK integration

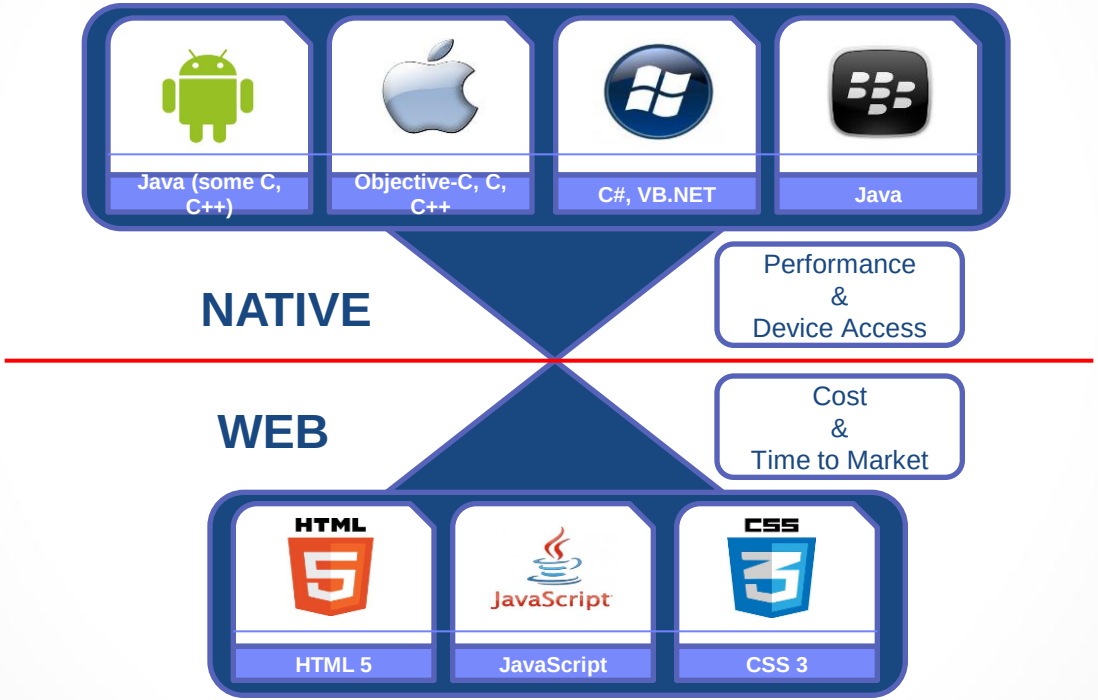
- Access to emulators and debugging tools

Supports auto-complete and validation





Traditional Mobile Development Models

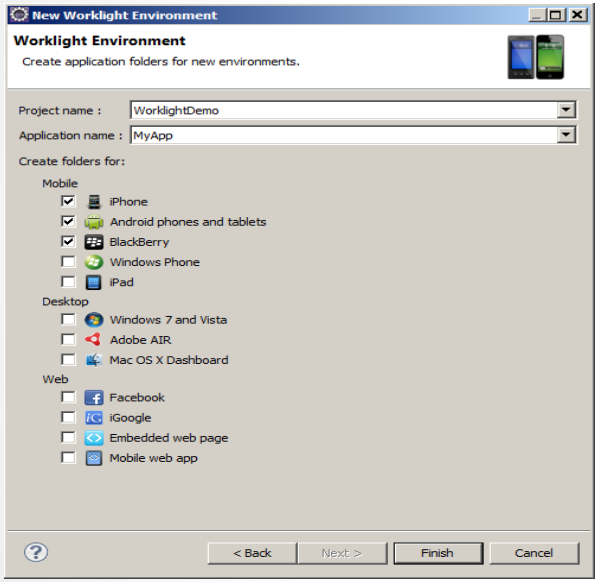


Worklight Studio
Development Models

Browser Access	Hybrid Apps - Web	Hybrid Apps - Mixed	Native Apps
Written in HTML5 JavaScript and CSS3. Quick and cheap to develop, but less powerful than native.	HTML5 code and Worklight runtime libraries packaged within the app and executed in a native shell.	User augments web code with native language for unique needs and maximized user experience.	Platform-specific. Requires unique expertise, pricy and long to develop. Can deliver higher user experience.
Browser Access	Downloadable	Downloadable	Downloadable

Worklight Studio

Common Code Structure and Environment Optimisation



common resources serve as the basis for the optimized environments added to the project

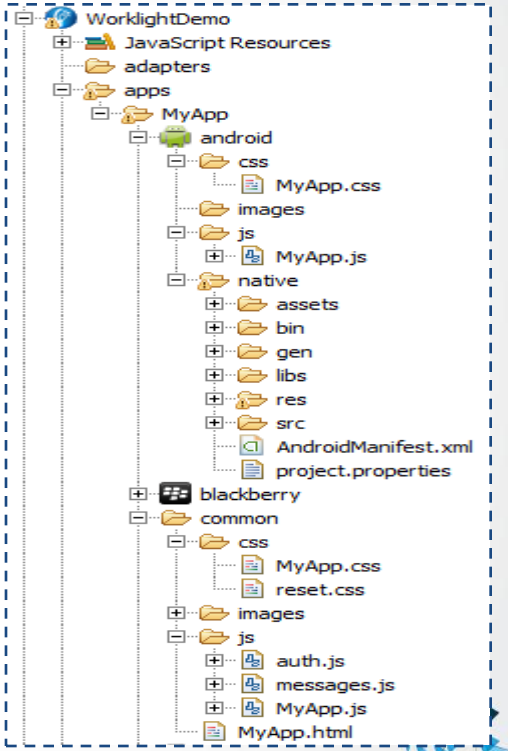
common holds resources that are shared by all environments.

JavaScript from an environment folder is appended to the same file in the common folder

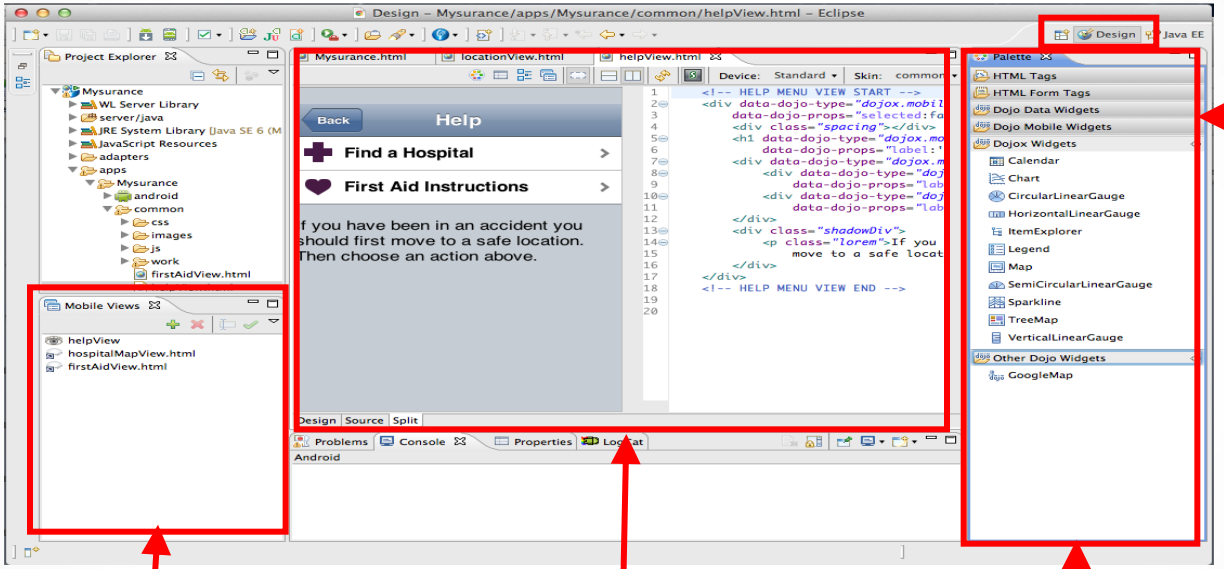
CSS files from an environment folder will be appended to the CSS files in the common folder

HTML files from an environment folder will override files in the common folder

images from an environment folder will override files in the common folder



Worklight Studio
Visual UI Construction



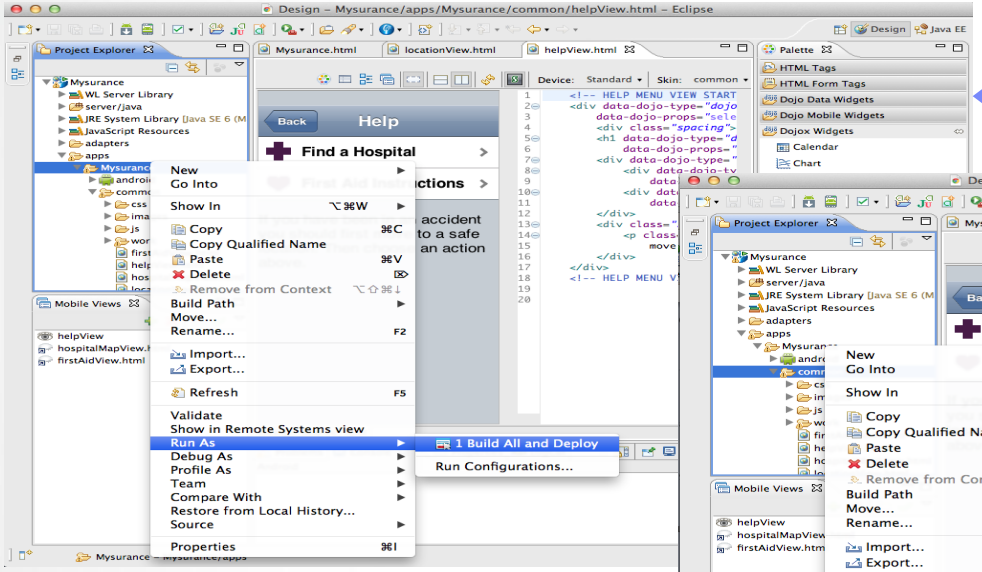
Worklight Design Perspective

Mobile Views
- Select which view displays in design tab
- Modify view properties
- Create and delete mobile views

Rich Page Editor
- Source, Design and Split views
- UI rendering using browser engine, synched with source pane

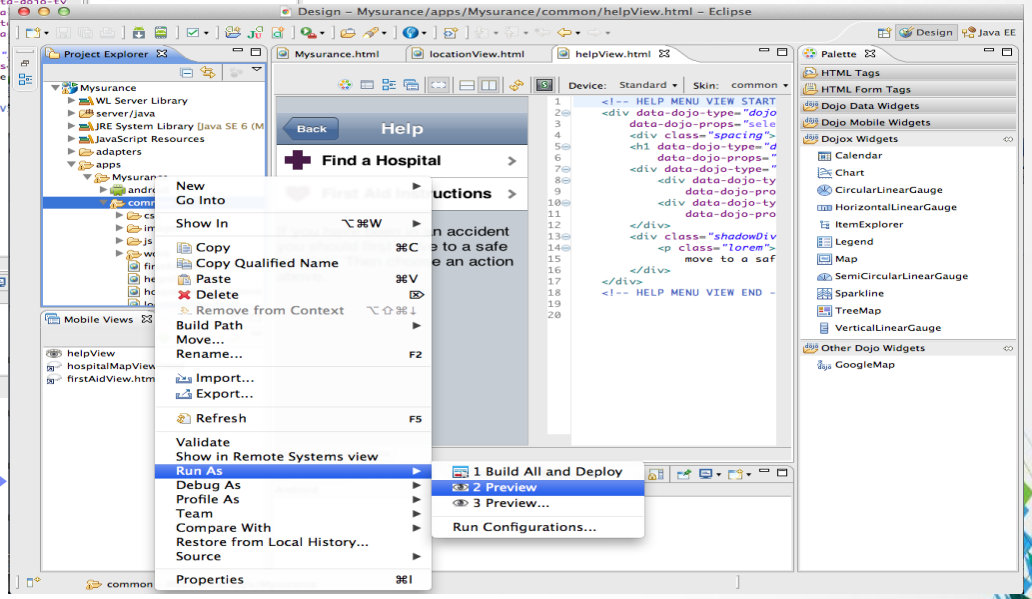
Palette
- Drag and drop visual elements into the Editor pane.
- Offers HTML constructs and Dojo widgets

Worklight Studio
Build, Package, Deploy and Test Applications

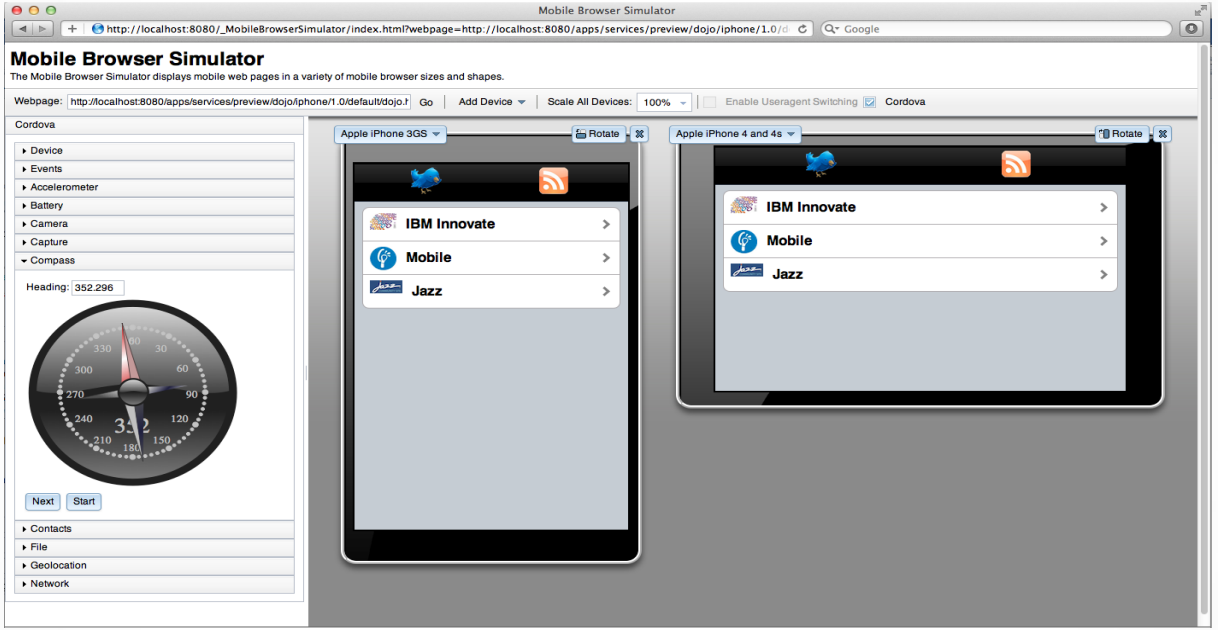


Build and Deploy the Worklight Application to Studio's **embedded web container**

Preview the application in the **Mobile Browser Simulator** for device-specific rendering and testing

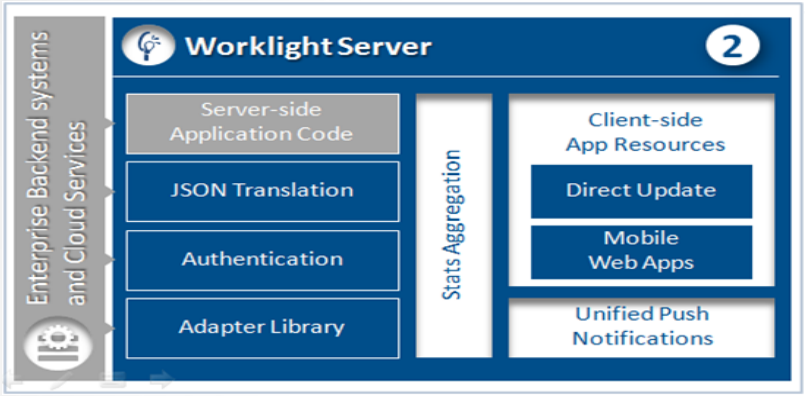


Worklight Studio
Preview in Mobile Browser Simulator



- Device specific visualisation and testing on the developer desktop
- Change device types, orientation, device settings (GPS loc, compass, etc…)
- Supports Cordova and Worklight client API
- Live debug with Web Inspector/Firefox

Worklight Server



Controls **Application Deployment** and **Versioning**

Server and device **Security** control

Adapters with support for SOAP, REST, SQL, LDAP, SAP and more

Performs **Data Transformation** to streamline back-end data for mobile consumption

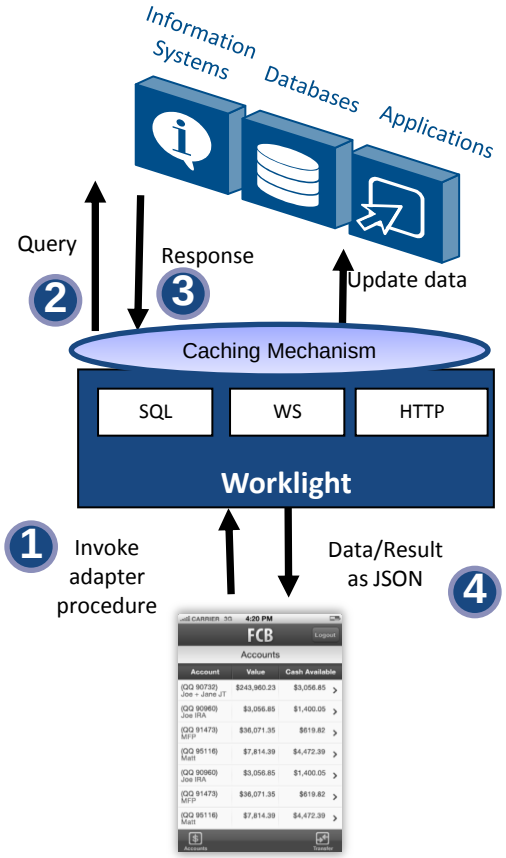
Push Notification administration

Analytics including user adoption and usage data

Runs on **WebSphere Application Server** versions 7, 8, 8.5 including the **Liberty Profile**

Supports **Physical Clustering** for high availability

Worklight Server Adapters



Universality
Supports multiple integration technologies and back-end information systems.
Out of the box support for SQL, HTTP (REST and SOAP) and Cast Iron Adapters

Read-only as well as Transactional Capabilities
Adapters support read-only and transactional access modes to back-end systems

Fast Development
Defined using simple XML syntax, and easily invoked by clients with JavaScript API

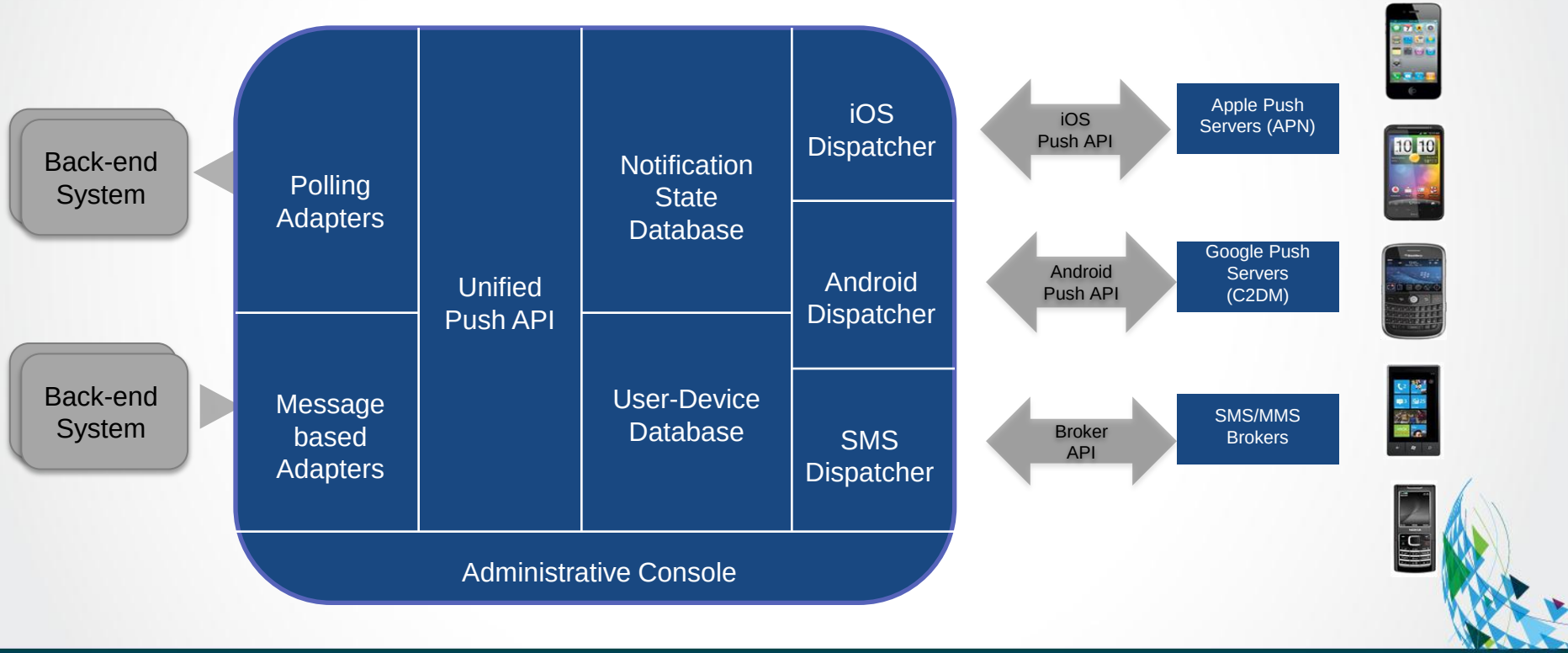
Security
Use of flexible authentication facilities to create connections with back-end systems
Adapters offer control over the identity of the connected user

Caching
Caching utilized to store retrieved back-end data

Transparency
Data retrieved from back-end applications is exposed in a uniform manner regardless of the adapter type

Flexibility
Open interface for custom adapter development

Worklight Server
Unified Push Notifications Architecture



Worklight Runtime
Shell Approach

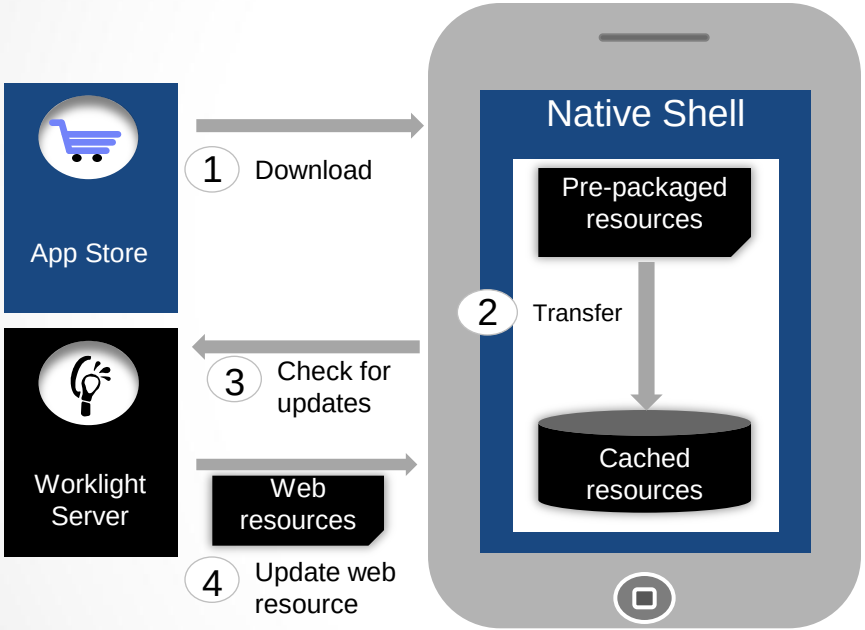
Inner Application:

- Common web code
- Utilizes External Shell API's
- Required to comply with shell parameters

External Shell:

- Install as a native App (IPA/APK file)
- Customizable container
- Provides JS access to native functionality
- Branding, Security, Authentication

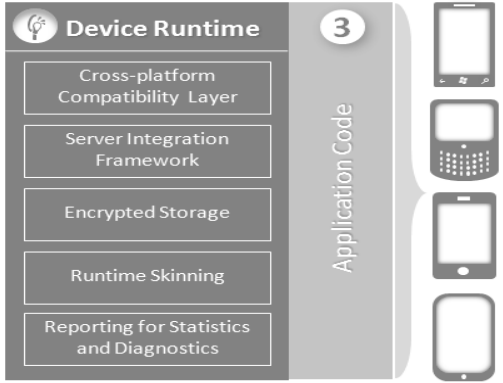
Worklight Runtime
Direct Update – On-device Logic

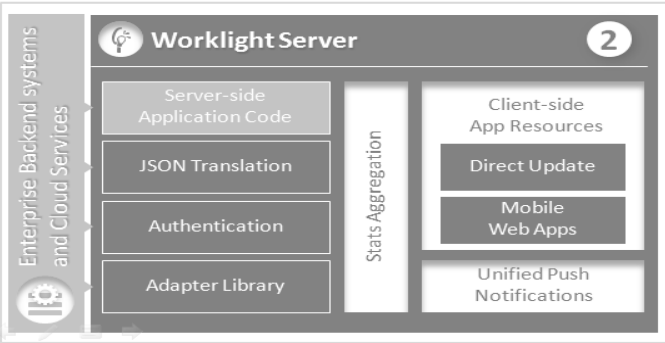


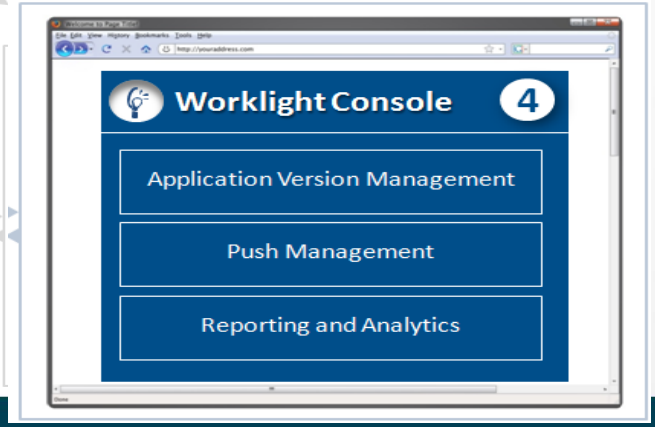
1. Web resources packaged with app to ensure initial offline availability
2. Web resources transferred to app's cache storage
3. App checks for updates on startup and foreground events
4. Updated web resources downloaded when necessary

Worklight Console

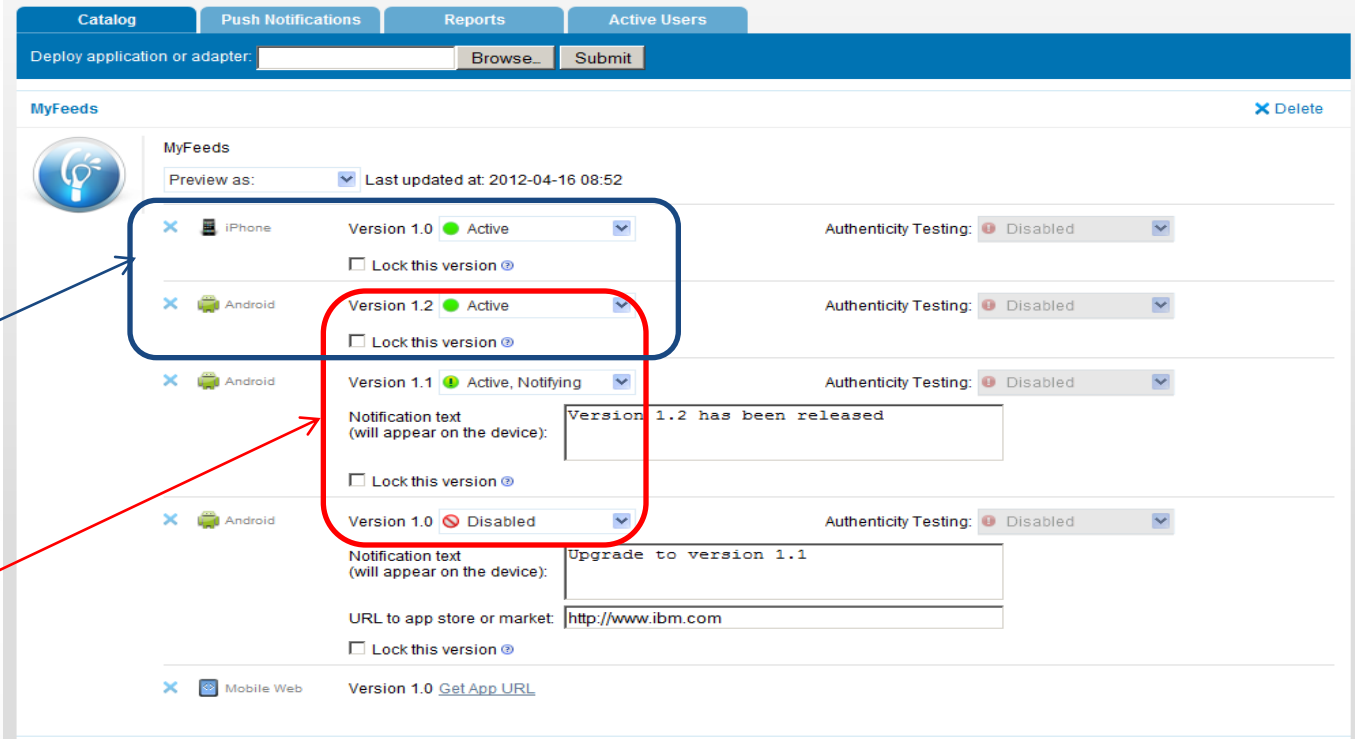
- Application Version Management
- Push management
- Usage reports and analytics
- Configurable audit log
- Administrative dashboards for:
  - Deployed applications
  - Installed adapters
- Push notifications
- Data export to BI enterprise systems







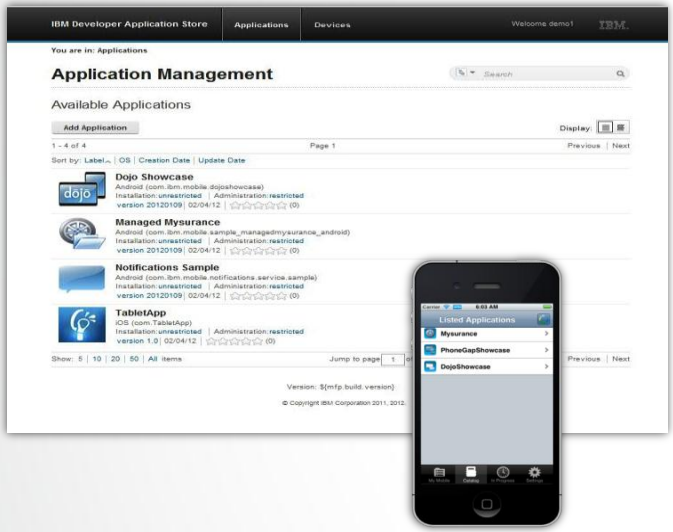
Worklight Console
Application Versioning



Device specific versions are uncoupled

Supports multiple versions on the same platform

Worklight Application Center
Private App Store



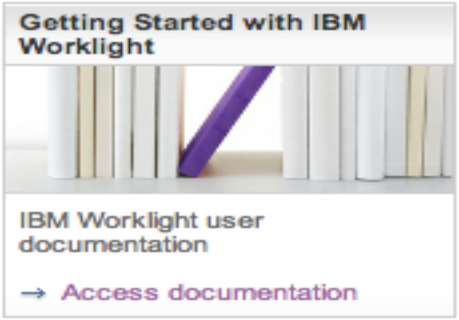
A cross-platform, private mobile application store similar to public app stores but focused on the needs of an organization or a team

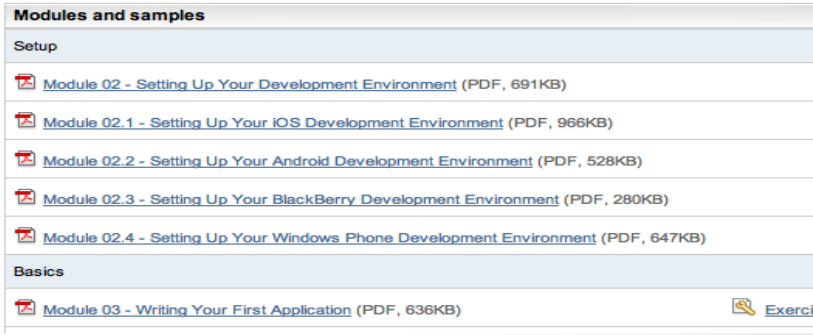
Key capabilities:

- delivers distribution and management of mobile applications within a company / teams
- easy distribution of iOS and Android apps within a team
- provides versioning and updates
- centralizes rating and feedback information
- controls who can modify or install an application
- easy to install and simple to run

Worklight
Getting Started Resources

http://ibm.com/worklight



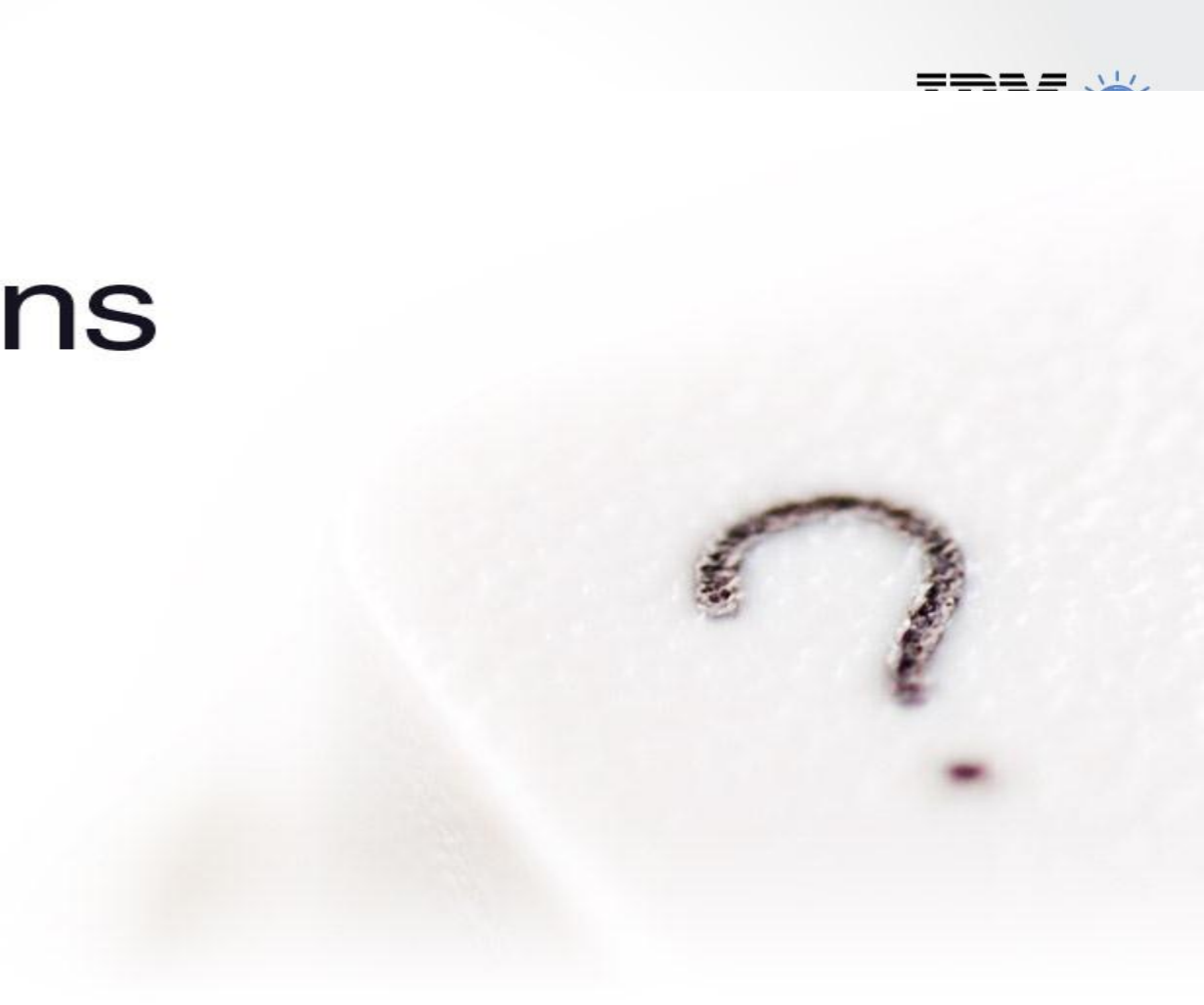


Local IBM Team

An IBM Proof of Technology

Discovering the value of the IBM Mobile Foundation – Worklight Core

Questions



IBM Mobile Foundation
Comprehensive Mobile Strategy

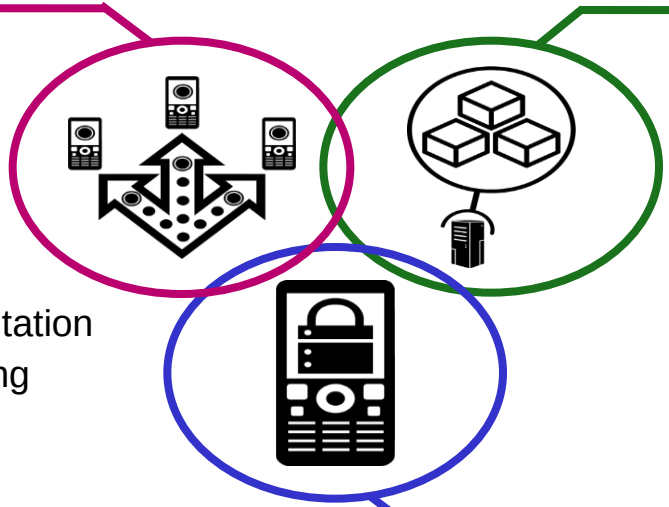
Extend & Transform

Extend *existing business capabilities to mobile devices*

Transform *the business by creating new opportunities*

Key Capabilities

- Strategy, planning and implementation
- Mobile-enabled solutions including analytics, commerce, and social business
- Mobile as a service



Build & Connect

Build *mobile applications*

Connect *to, and* ***run*** *backend systems in support of mobile*

Key Capabilities

- Multiplatform mobile web, hybrid and native app development
- Enterprise data, service, and application integration
- Iterative teamwork

Manage & Secure

Manage *mobile devices, services and applications*

Secure *my mobile business*

Key Capabilities

- Mobile Governance
- Device analytics and control
- Secure network communications & management

IBM Mobile Foundation
IBM Endpoint Manager for Mobile Devices

Traditional Endpoint Management

- OS provisioning
- Patching
- Power Mgmt

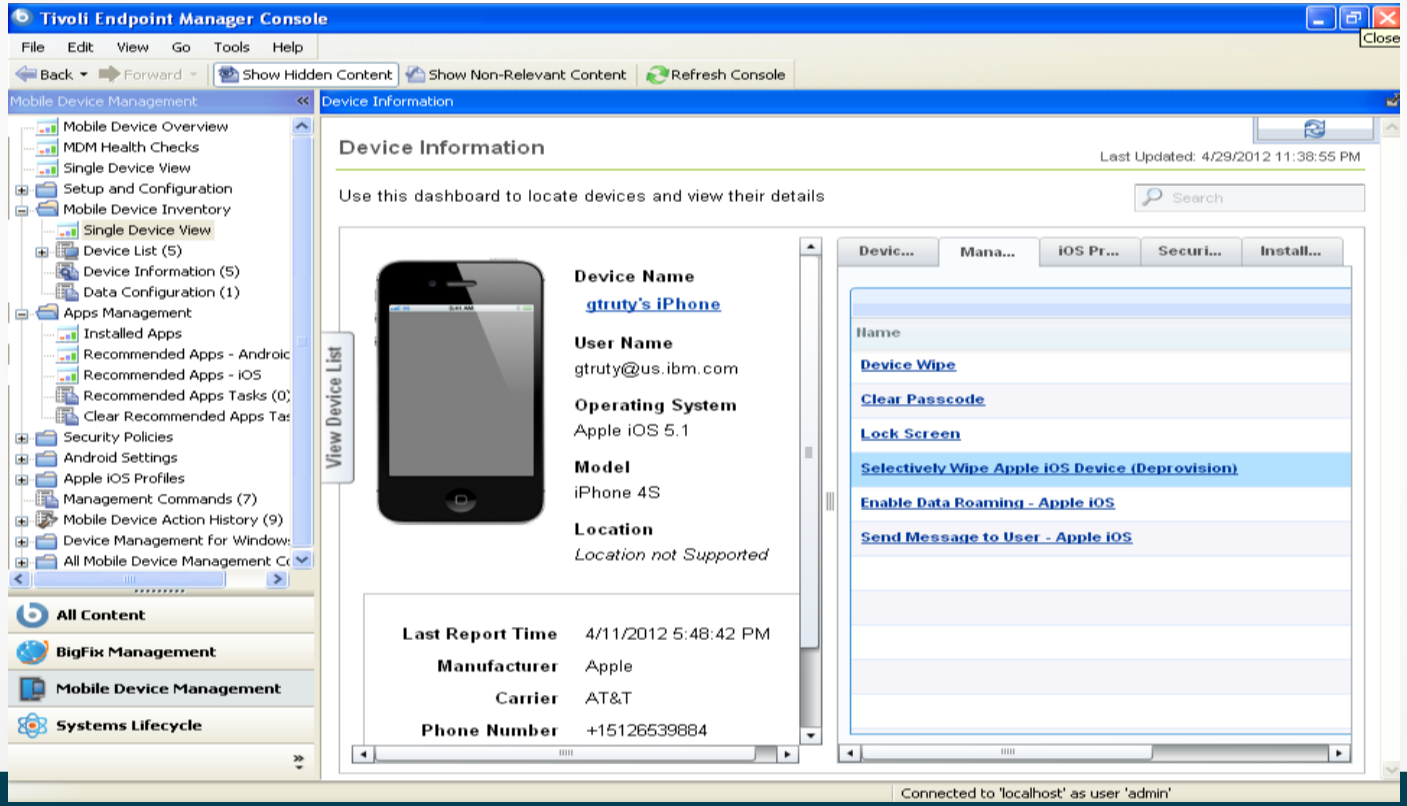
- Device inventory
- Security policy mgmt
- Application mgmt
- Device config (VPN/Email/Wifi)
- Encryption mgmt
- Roaming device support
- Integration with internal systems
- Scalable/Secure solution
- Easy-to-deploy
- Multiple OS support
- Consolidated infrastructure

Mobile Device Management

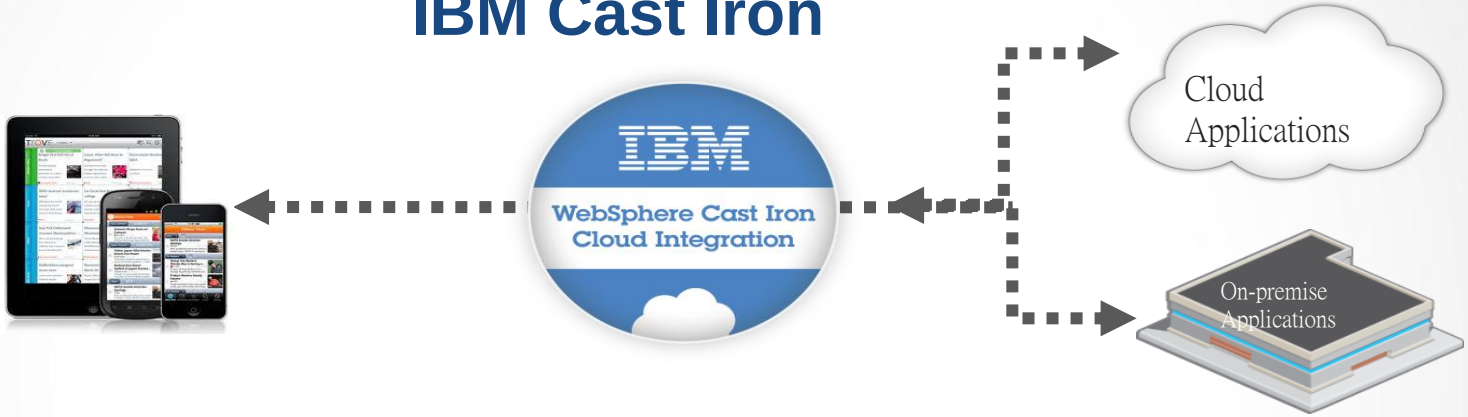
- Device Wipe
- Location info
- Jailbreak/Root detection
- Enterprise App store
- Self-service portal

IBM Mobile Foundation

IBM Endpoint Manager for Mobile Devices Device View



IBM Mobile Foundation
IBM Cast Iron



Flexible

Webspere Cast Iron Live

Hypervisor Edition

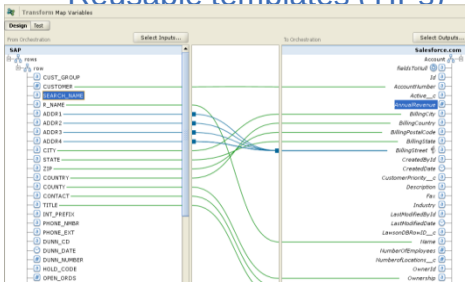
Physical appliances (IBM)

Comprehensive

Connectivity to hundreds of apps

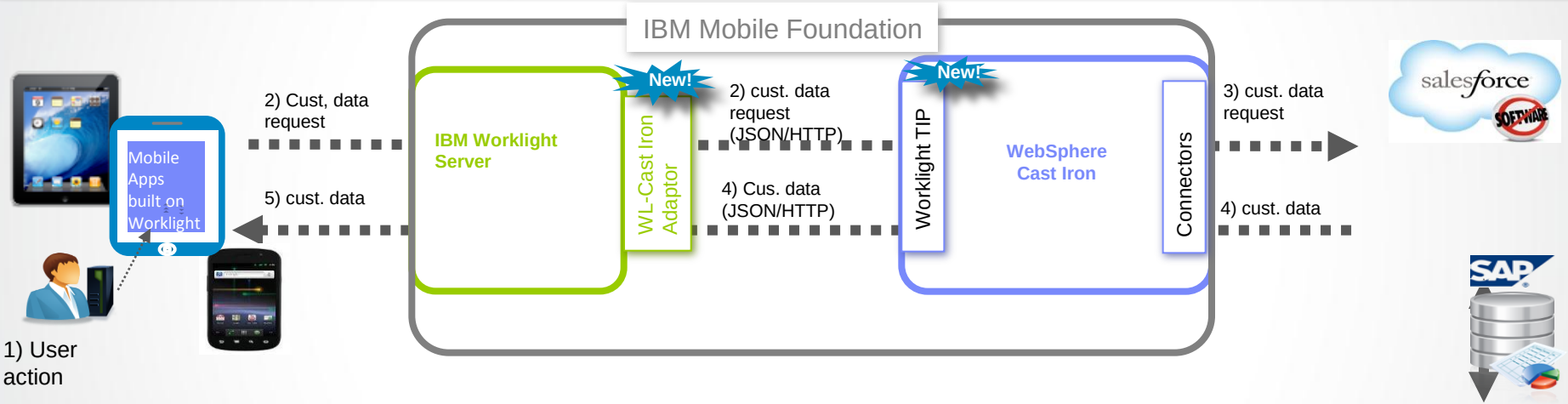
Rapid

Reusable templates (TIPs)



IBM Mobile Foundation
Integrating Worklight and IBM Cast Iron

Scenario: Worklight Application makes request for enterprise data



Request from Mobile app for customer data from SAP & salesforce

1) User invokes action on mobile application to request data
2) Worklight Server invokes it's WL Cast Iron Adapter to send customer data request to Cast Iron
3) Cast Iron receives request & invokes connectivity with salesforce.com and SAP to extract customer data
4) Cast Iron sends customer to Worklight Server
5) Worklight Server delivers data to mobile application on the device