Integrating Mobile apps with your Enterprise

Peter Niblett

Simon Dickerson

Trademark Statement

- IBM and the IBM logo are trademarks of International Business Machines Corporation, registered in many jurisdictions. Other marks may be trademarks or registered trademarks of their respective owners.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
- Other company, product and service names may be trademarks, registered marks or service marks of their respective owners.
- References in this publication to IBM products and services do not imply that IBM intends to make them available in all countries in which IBM operates.

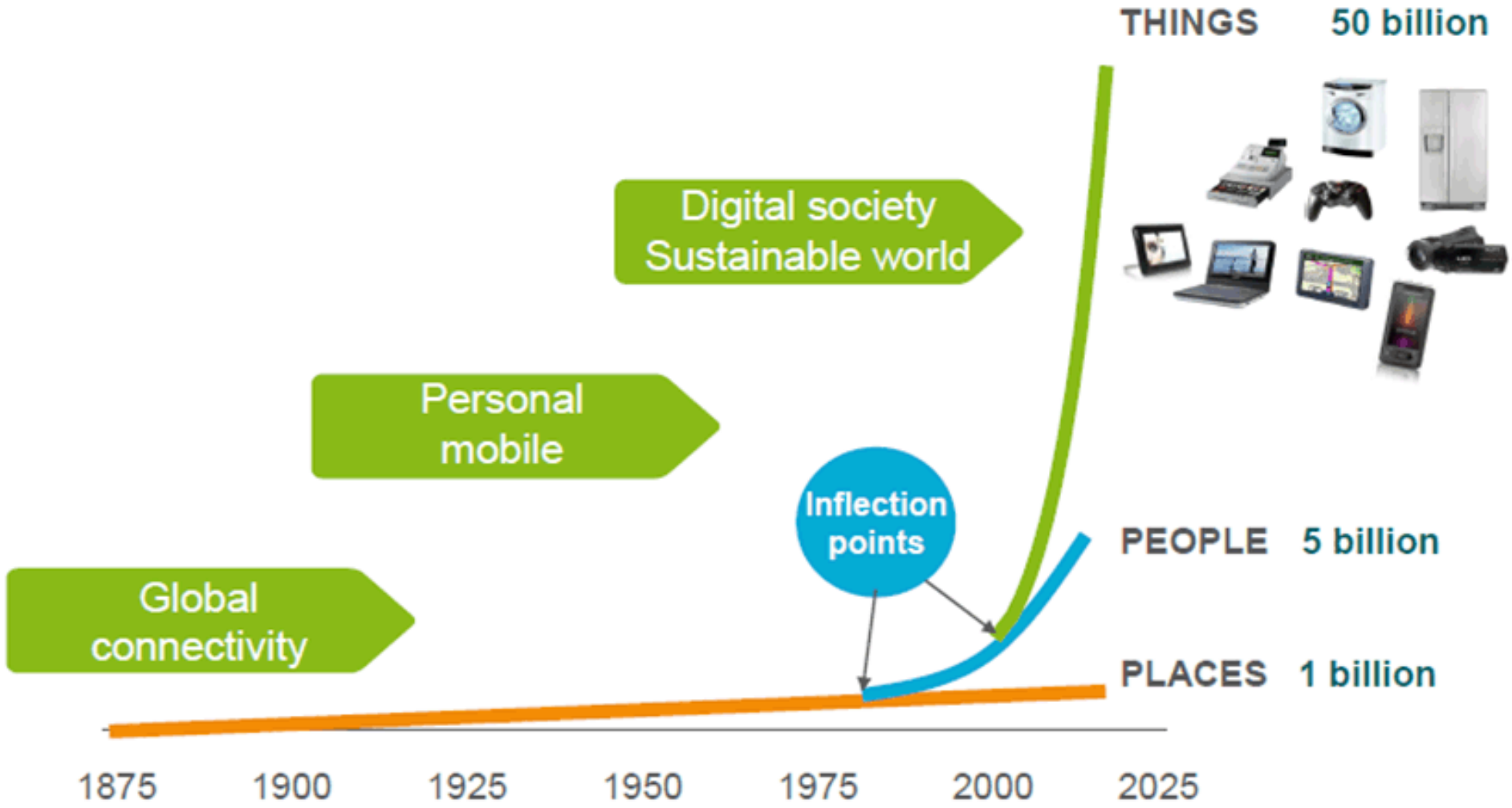
Agenda

- Mobile apps and the enterprise
- IBM Enterprise Mobile Platform
- Integrating with Enterprise Applications
- Mobile Messaging
- Summary

Mobile … a new frontier of business growth

- ***Mobile B2C***
  - *Increase customer satisfaction by enabling banking, insurance, and trading anywhere, anytime*
  - *Reach customers in new ways through mobile applications, SMS, email*
- ***Mobile B2E & B2B***
  - *Enable field employees for increased productivity*
  - *Greater efficiency and accuracy in supply chain operations*
  - *Exchange business information with partners securely*
- ***Mobile M2M***
  - *Enable the exchange of data and events between businesses and machines*
  - *Internet of Things - sensor events feeding information and driving a smarter planet*

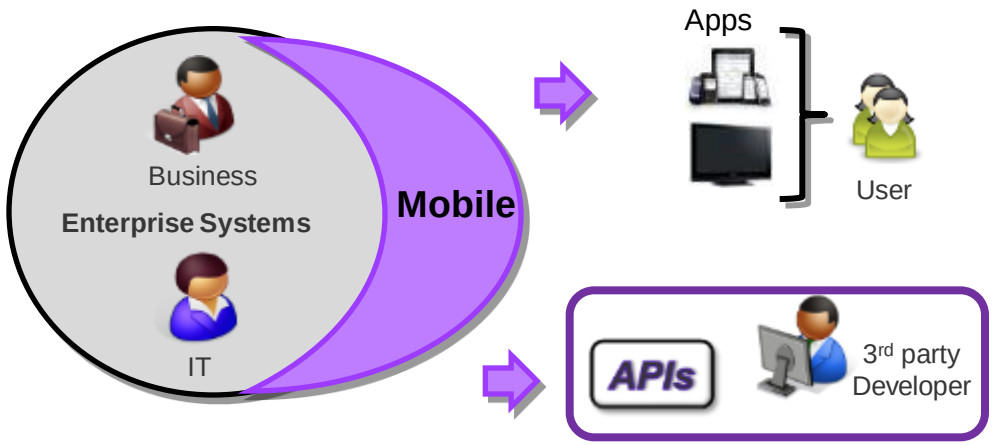
Mobile is a prelude to *The Internet of Things*



Source: Ericsson AB, "Infrastructure Innovation - Can the Challenge be met?," Sept 2010

Mobile Apps – your options

1. Develop and operate in-house
   - Typical choice for B2E apps
   - Some organizations are using this approach for B2C apps
2. Outsource development and/or operations
   - Use specialist web or mobile development company
   - Sometimes a spin-out from the main enterprise
3. Web APIs
   - Apps are developed, owned and operated by independent enterprises
   - Often branded by the third parties
   - Interface to systems of record is provided by public APIs



Mobile is different:

- Transformational business models
- Faster lifecycles
- More iterative

User requirements for mobile solutions and what they mean for you

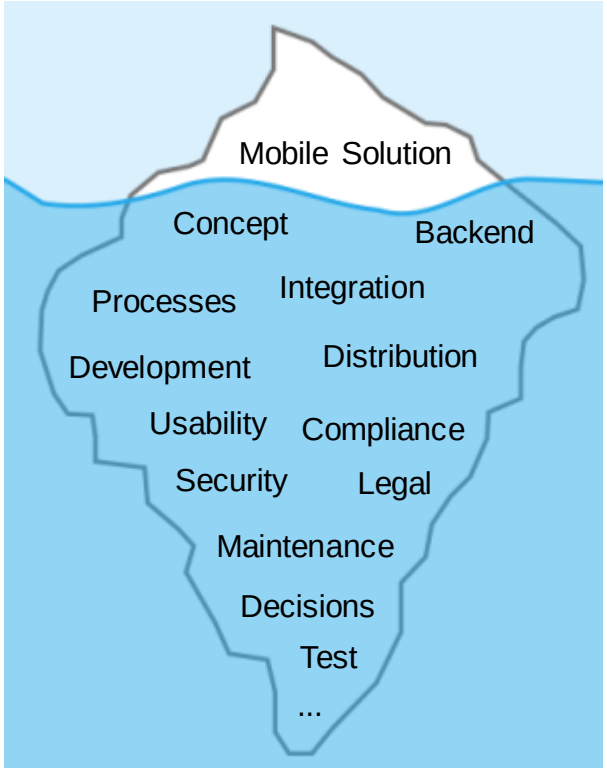
User expectations vs. Enterprise constraints

- High expectations of **usability, appearance and behavior**
- **Short and focused interactions**, interruptions are common

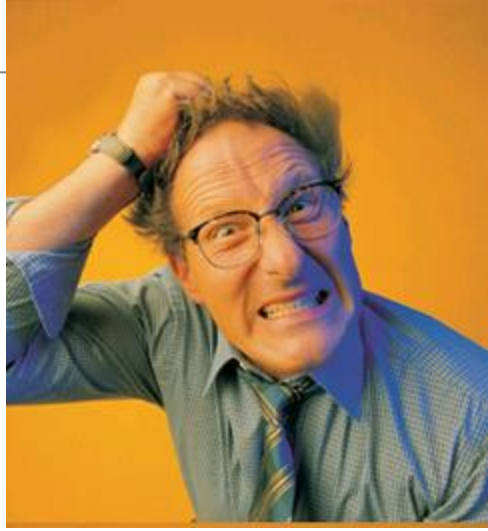
- Must be **usable when out of wireless coverage**
- **Context awareness** is the differentiator to provide the services the customer needs

- Offer **all features** provided on other channels, but adapted to the mobile needs
- **Mobile** does not necessarily mean on the go, it means **always with me**

- Highly **fragmented** set of devices and platforms
- Evolves fast, **frequent releases** and updates
- **Mobile is more than just apps**



Mobile is challenging



- Bring Your Own Device (BYOD)
  is forcing companies to support a range of devices.
- Mobile Development is more challenging than traditional Web App Development:
  - Which smartphone? Which tablet? Which form factor?
    - → iOS, Android, Blackberry, Windows Phone
    - → All of the above..
  - Skills?
    - → Web or native apps? Java or Objective C? Or other?
  - Maintenance?
    - → Separate software stacks for each major OS
    - → Separate applications for each major OS
    - → How do I keep software current?
  - Security?
    - → Encryption? Authentication?
    - → Response to stolen/lost devices?
  - Management?
    - → How do I provide support and service?
  - Enterprise Integration?
    - → How do I build cross-channel app?
- Meanwhile, IT budgets are shrinking.

Techniques for Creating Exceptional Mobile Experiences

Browser Based, Web Applications

- Accessible over the internet without need to download software
- Uses device browser to display content

Hybrid – Both Web and Native Components

- Native looking applications which utilize the browser interface to deliver content
- Provide the ability to use native device features without writing code for each device

Native

- Able to make use of phone's native features such as camera, GPS, accelerometer, calendar, etc..
- Supports the richest of user experiences (e.g., gaming applications)

What to consider: Native, Web, Hybrid

- Easy to develop and manage
- Only browser specifics

Web

- Compromise
- Depending on functionality of Development environment

Hybrid

- High Performance
- Best Usability
- Always limited to specific devices

Native

Characteristics

Web Application	Mobile Web Application	Hybrid Mobile Applic.	Native Mobile Applic.
Desktop and mobile using open web (HTML, JavaScript) client programming models Limited to no device-specific functionality	Mobile only using open web (HTML5, JavaScript) client programming models Off-line capabilities	Mobile only, app runs on device leveraging open web (HTML5, JS) via JavaScript bridge Native device capabilities (GPS, camera, contacts) Mimic native appearance	Mobile only, developed using native languages or transcode to native via MAP tools Native appearance and device capabilities, performance
Mobile Browser Execution		AppStore download and install	

Traditional Trade-offs (without MEAP/MAP)

Richness of Mobile Presentation / Services

Portability (cross-device reuse)

Maintenance Cost (TCO)

Agenda

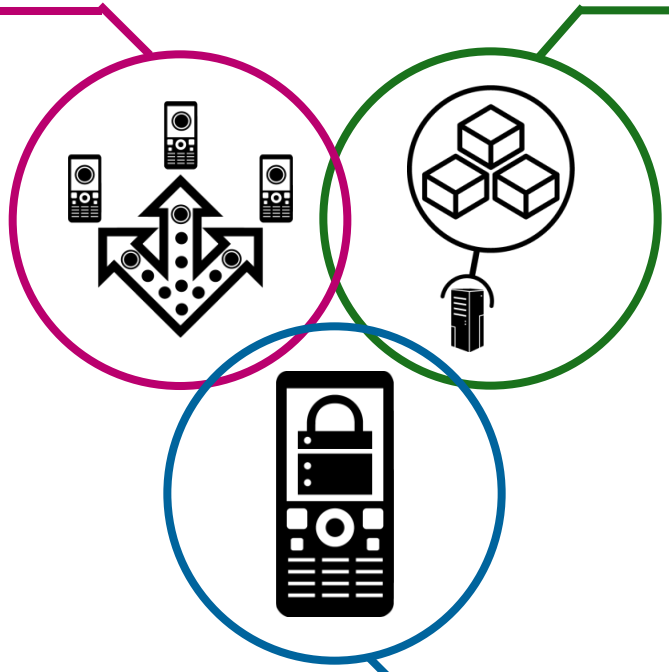
- Mobile apps and the enterprise
- **IBM Enterprise Mobile Platform**
- Integrating with Enterprise Applications
- Mobile Messaging
- Summary

IBM Strategy Addresses IBM Client Mobile Initiatives

Extend & Transform

Extend *existing business capabilities to mobile devices*

Transform *the business by creating new opportunities*



Build & Connect

Build *mobile apps*

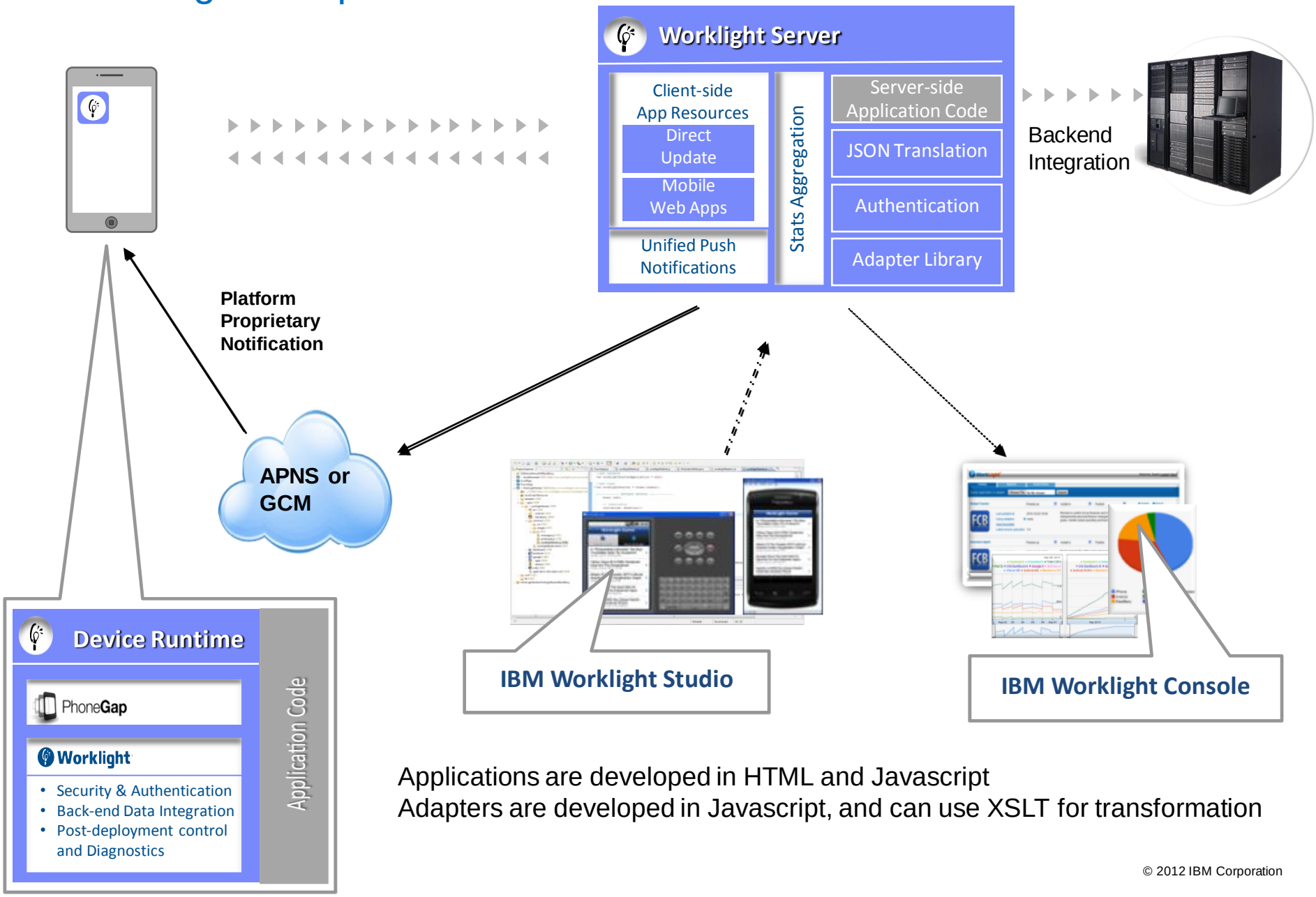
Connect *to, and* ***run*** *backend systems in support of mobile*

Manage & Secure

Manage *mobile devices and apps*

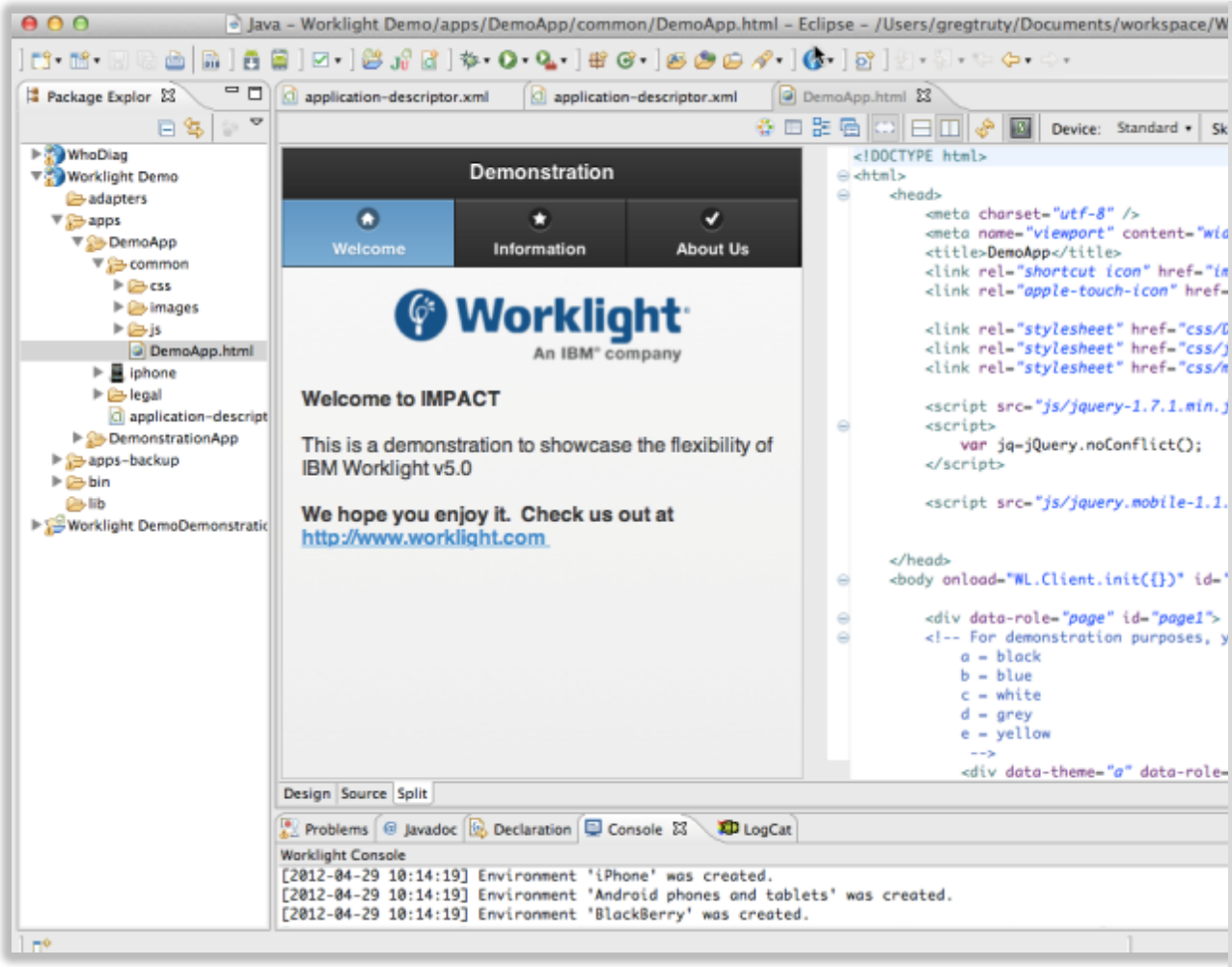
Secure *my mobile business*

IBM Worklight components



Applications are developed in HTML and Javascript
Adapters are developed in Javascript, and can use XSLT for transformation

IBM Worklight Studio



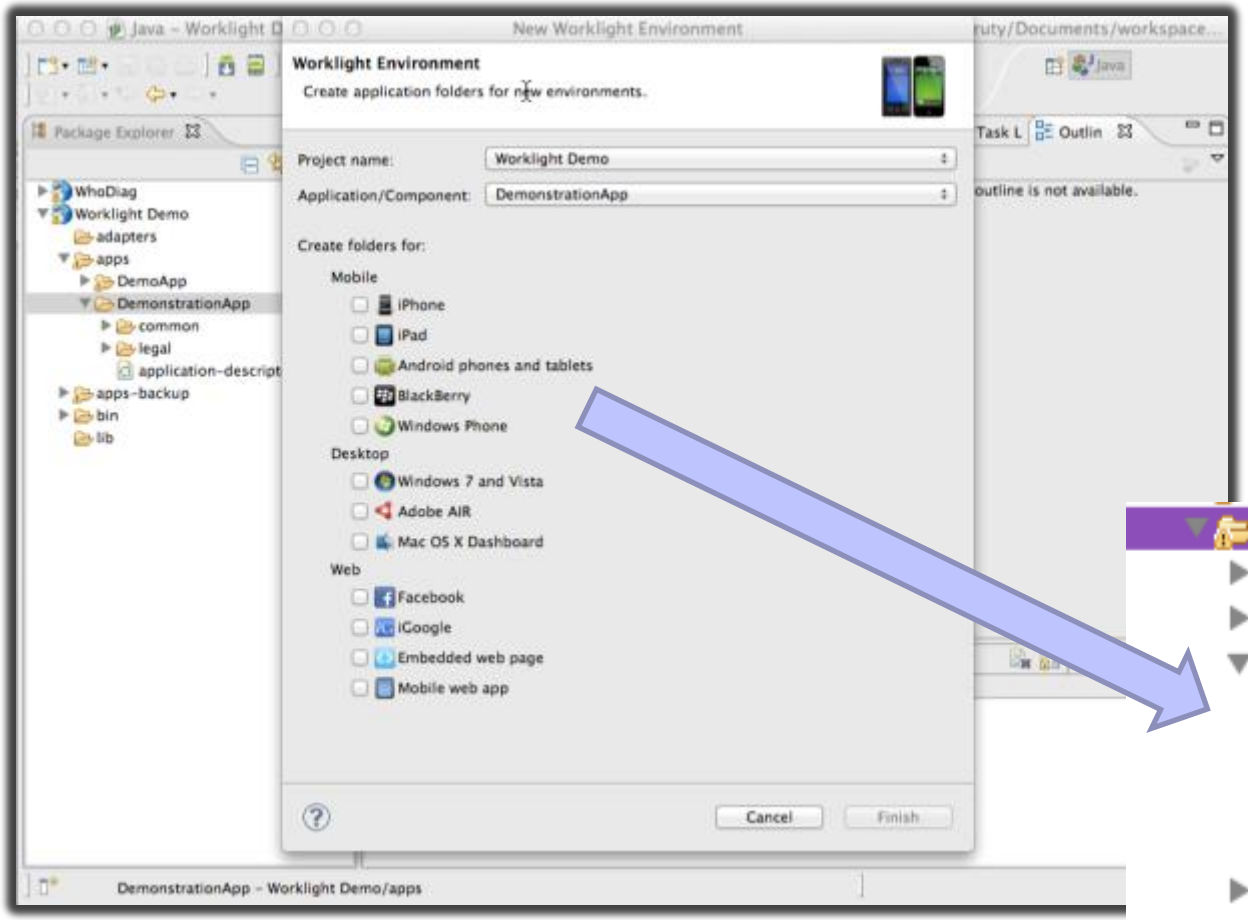
Integrated Development Environment (Eclipse Plug-in)

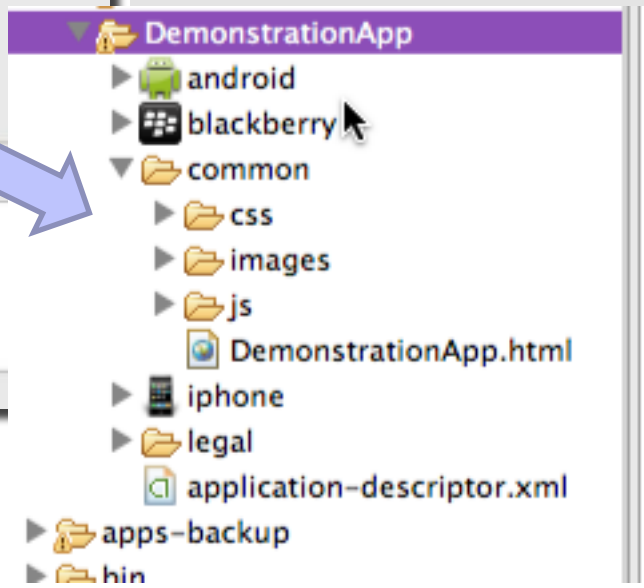
Application development using native and/or familiar web technologies:

- HTML5
- CSS3
- JavaScript

Integrated device SDKs allow direct access from within the IDE to emulators and code debugging utilities

Single Shared Codebase

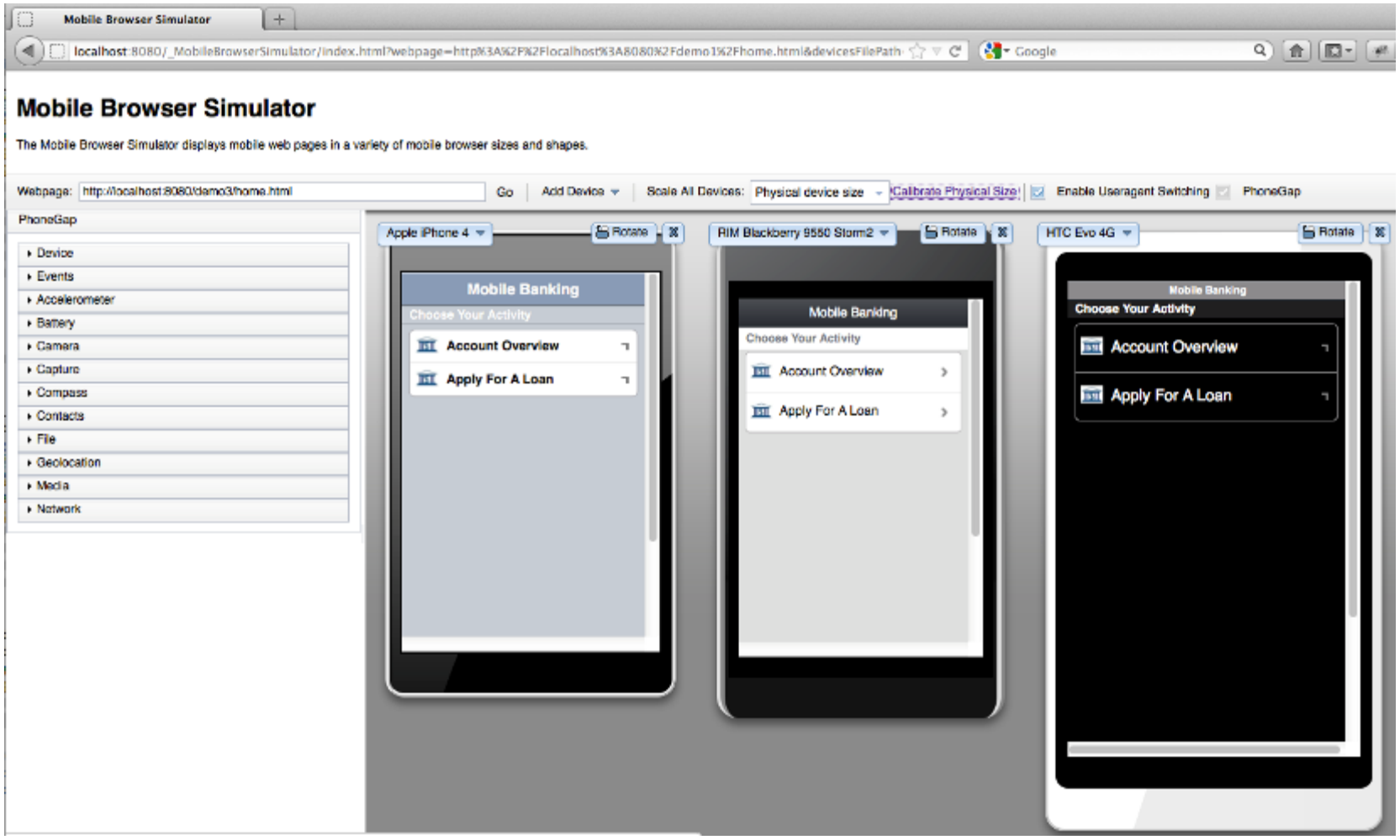




Common code placed in primary file

Environment optimization code is maintained separately

Preview in browser

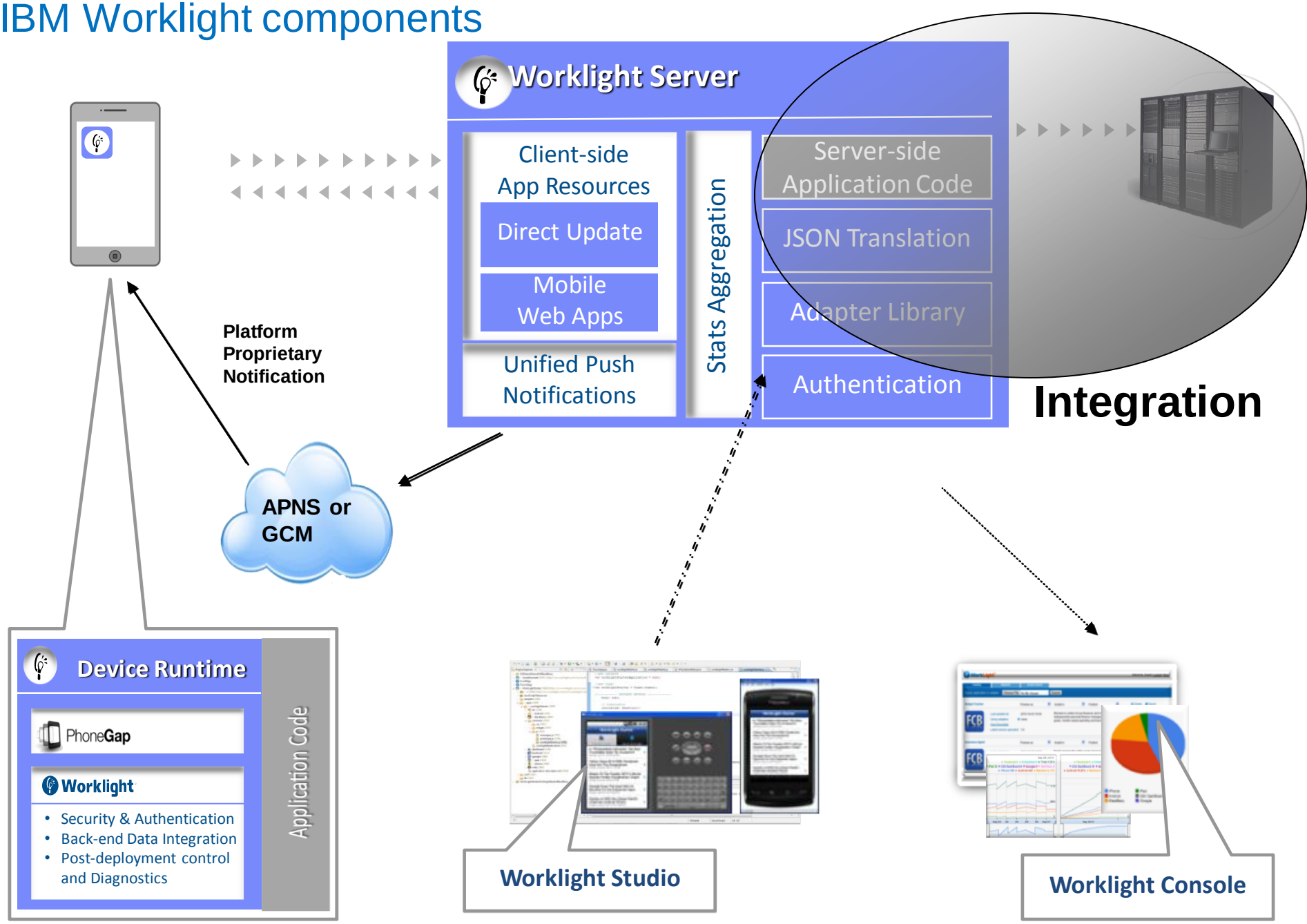


Perform device specific tests in the Mobile Browser Simulator: supports Cordova and IBM Worklight client API

Agenda

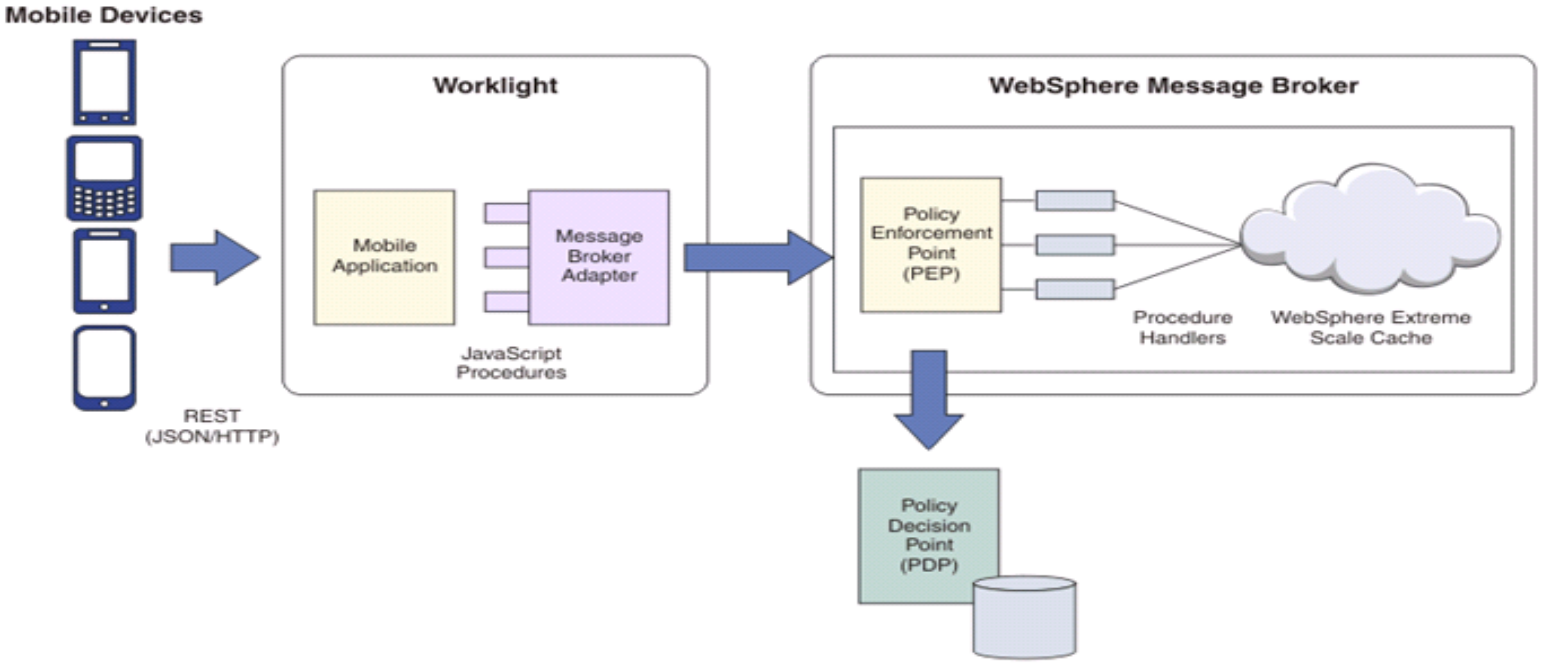
- Mobile apps and the enterprise
- IBM Enterprise Mobile Platform
- **Integrating with Enterprise Applications**
- Mobile Messaging
- Summary

IBM Worklight components



Worklight Adapters

- Adapters provide the glue between Worklight and back-end applications
  - Provides the extensibility mechanism for Worklight to call out to back-end systems
- Adapters are invoked from mobile applications using HTTP/JSON
  - This makes Worklight adapters easy to test using web browsers
- Worklight has built-in interfaces that adapters can use (HTTP, SQL and Cast Iron)
  - Worklight has client-side JavaScript APIs so that applications can invoke services
  - Likewise, server-side JavaScript APIs are available to implement procedures (adapters)



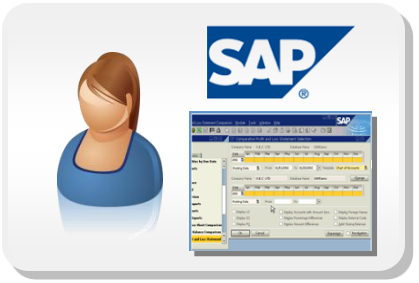
Adapter Benefits

- **Universality**
  Supports multiple integration technologies and back-end information systems
- **Read-only as well and Transactional Capabilities**
  Adapters support read-only and transactional access modes to back-end systems
- **Fast Development**
  Use simple XML syntax and easily configured 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
- **Scalability**
  Adapters can reduce the number of transactions on back-end systems by using cache to store retrieved back-end data
- **Transparency**
  Data retrieved from back-end applications is exposed in a uniform manner regardless of the adapter type

Cast Iron demo integration topology

Mobile App

Worklight
An IBM company



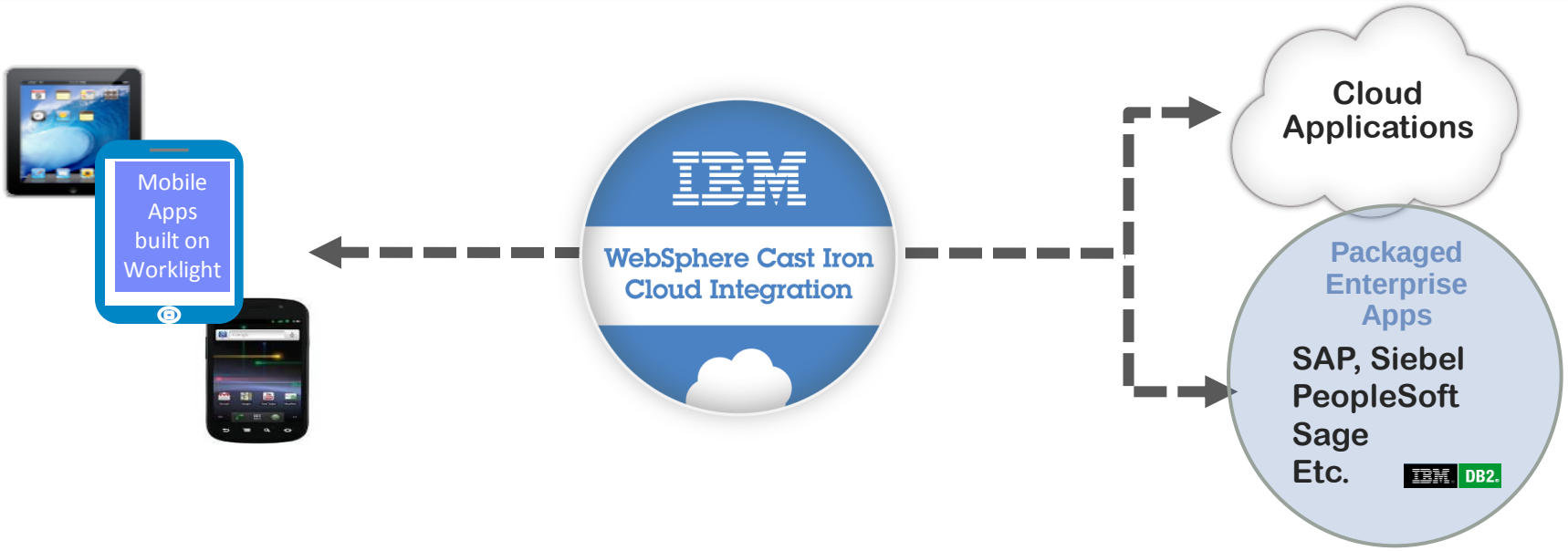
IBM
Cast Iron Cloud
Cast Iron

WebSphere

Message Broker

IBM WebSphere Cast Iron and IBM Mobile Foundation

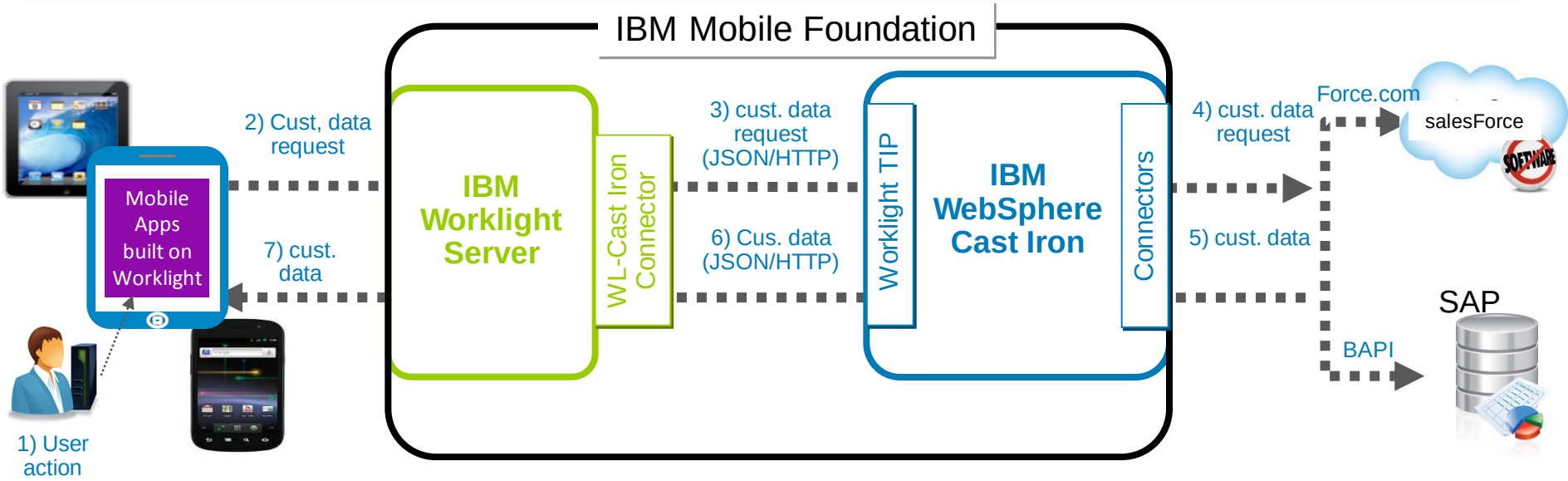
Connects Worklight Apps with Cloud & On Premise Applications in Days



- **Rapid** connectivity to hundreds of Cloud and On premise applications
  - Real-time data feeds from hundreds of enterprise applications to Mobile apps built on Worklight
- **Simple** *configuration, not coding* approach
  - Eliminates need to custom code Mobile integration scenarios
- **Optimized** for Worklight integration scenarios
  - Worklight connector to seamlessly communicate with Cast Iron

Using IBM WebSphere Cast Iron to access ERP applications

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) Mobile application on client device communicates with Worklight Server

  3) IBM Worklight Server invokes its WL Cast Iron connector to send customer data request to IBM WebSphere Cast Iron – JSON/HTTP (Rest)

  4) IBM WebSphere Cast Iron TIP receives request (JSON) & invokes connectivity with salesforce.com and SAP to extract customer data

- **Response from IBM WebSphere Cast Iron**

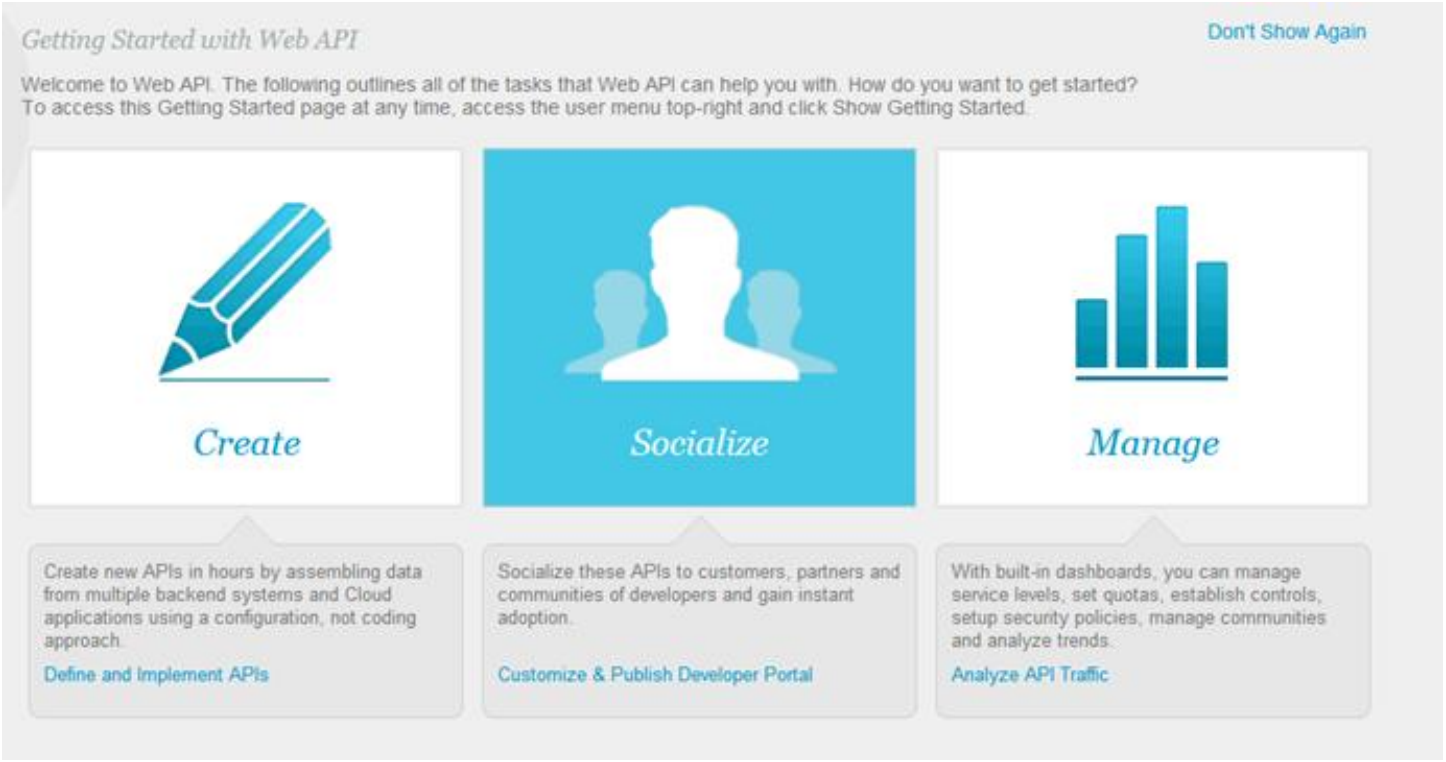
  5) TIP transforms the data based on requested fields

  6) TIP converts XML data into JSON format and sends to IBM Worklight Server via HTTP (Rest)

  7) IBM Worklight Server delivers data to Mobile application on the device

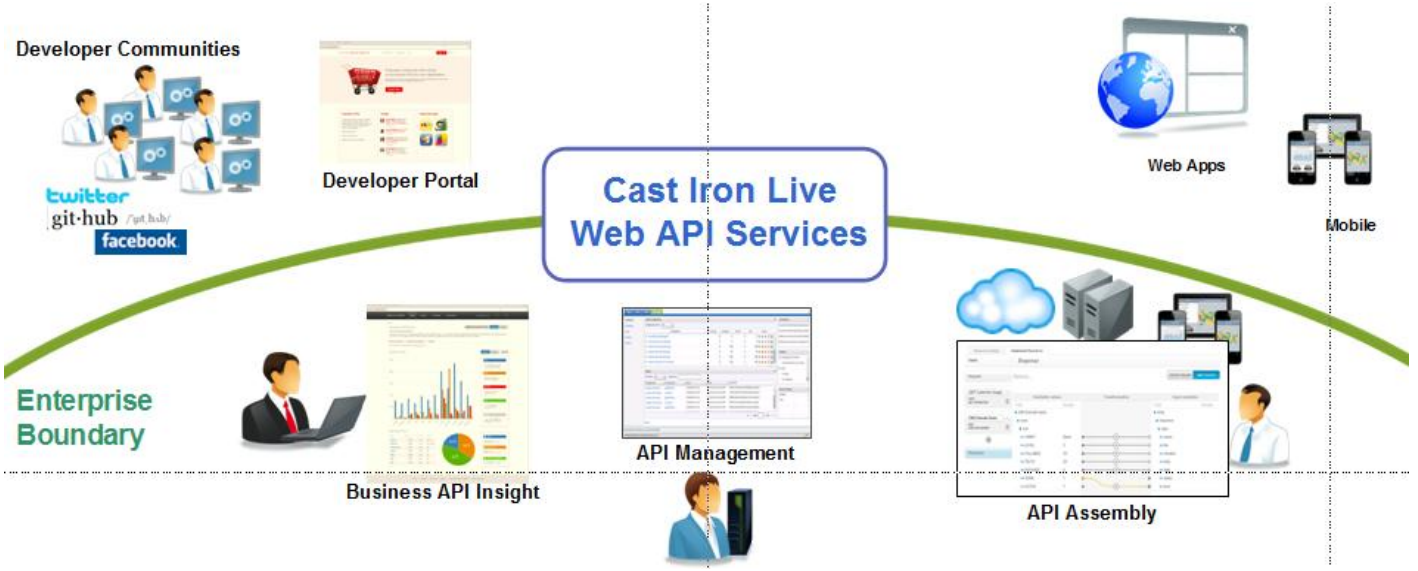
WebAPI

IBM Cast Iron Web API is a ‘software as a service’ offering that enables companies to rapidly create new web APIs, socialize the web APIs in various communities, and manage web APIs.



https://www.webapi.castiron.com

WebAPI



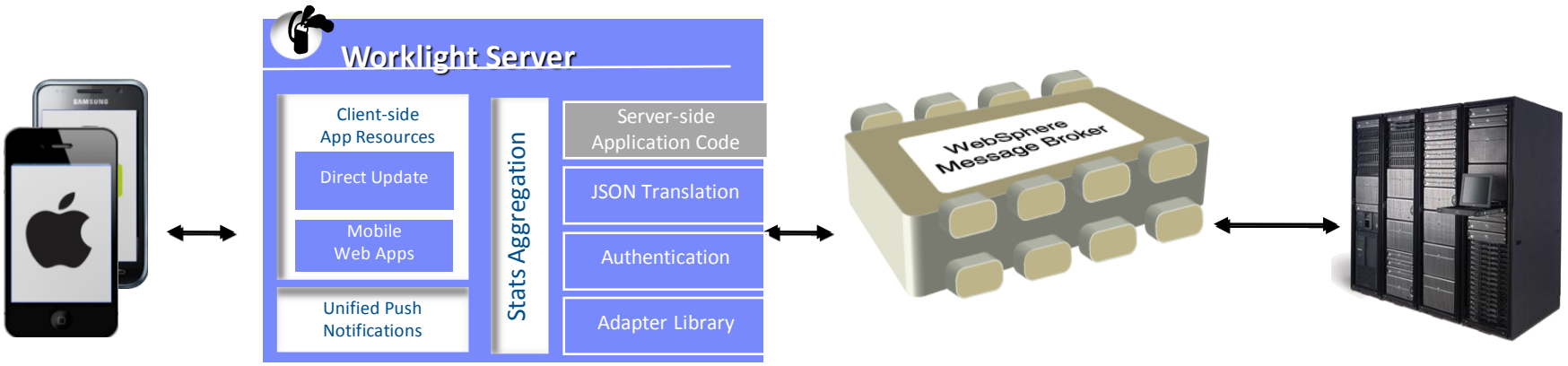
- Up and running in minutes
- Proxy to existing services
- Rapidly assemble new APIs
- Documentation of APIs
- Full analytics
- Rate limiting
- Developer portal
- Caching and flood control

There is a trend to use API Management for exposing data internal to the enterprise,

It is not just about making your data public

2

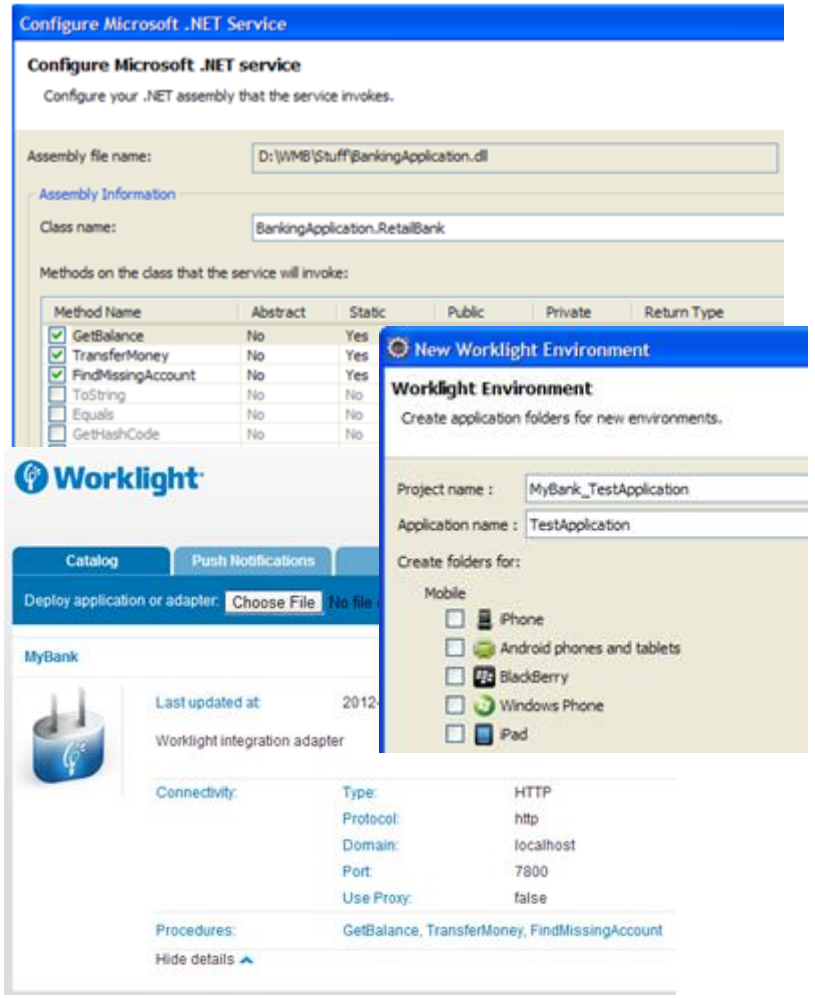
IBM WebSphere Message Broker and IBM Worklight



- Four new patterns shipped in IBM WebSphere Message Broker 8.0.0.1
  - Expose Microsoft .Net application as a mobile service
  - Mobile enable any enterprise service
  - Controlled REST style access to a Resource (with caching option)
  - Push Notification (IBM WebSphere MQ to IBM Worklight)

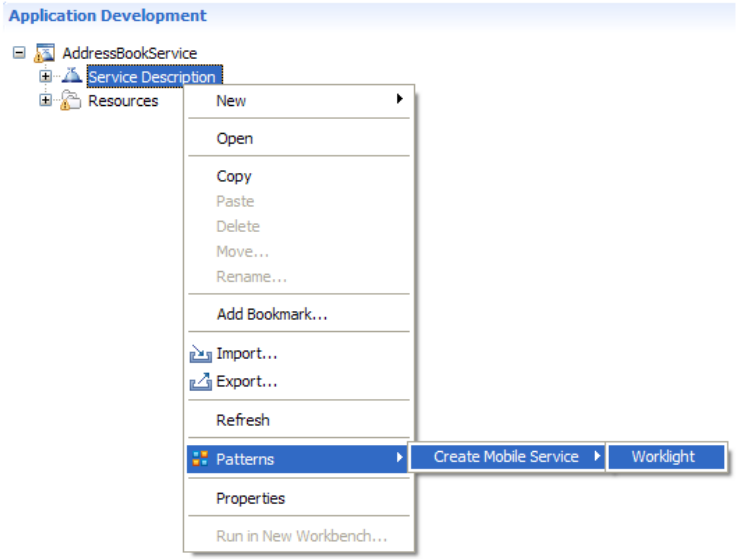
Pattern 1 - Expose a Microsoft .NET application as a mobile service

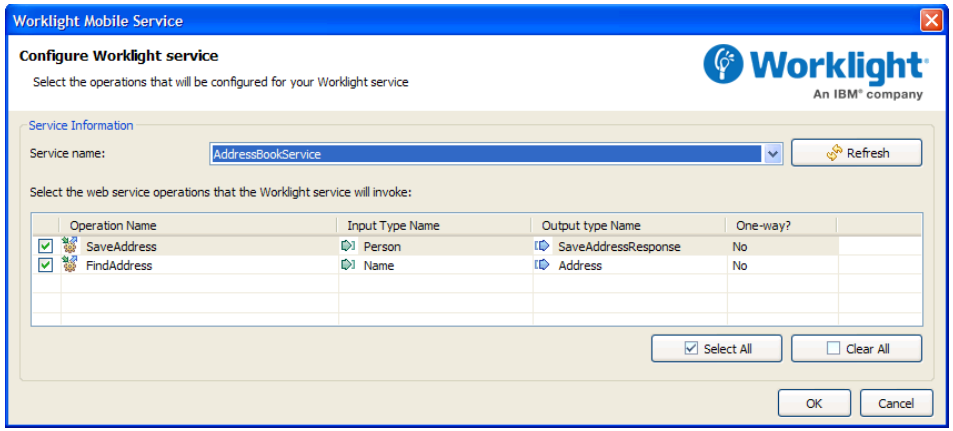
- **Simple to configure** – Drag and drop .NET assembly and enter Worklight adapter details
- **Super quick** – Pattern does all the hard work in less than a minute, generating…
  - A Web service implementation exposing desired operations
  - An adapter **ready for deployment to IBM Worklight Server**
  - A sample mobile application for **easy testing**
    - Optimised for small screen mobile devices; easy to add extra environments for iOS, Android and many more!

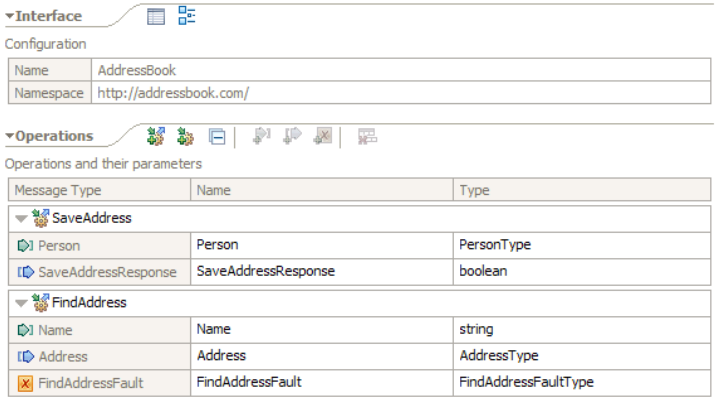


Pattern 2 - Mobile enable ANY enterprise service

- **As few as 2 Clicks!**
  - Right-click on any enterprise service (MQ, Database, Web service, CICS, IMS, etc…)
  - Left-click to create mobile service with default options
- Pattern supports **extra options**…
  - Choose operations to be available to mobile applications
  - Enable auditing of service requests

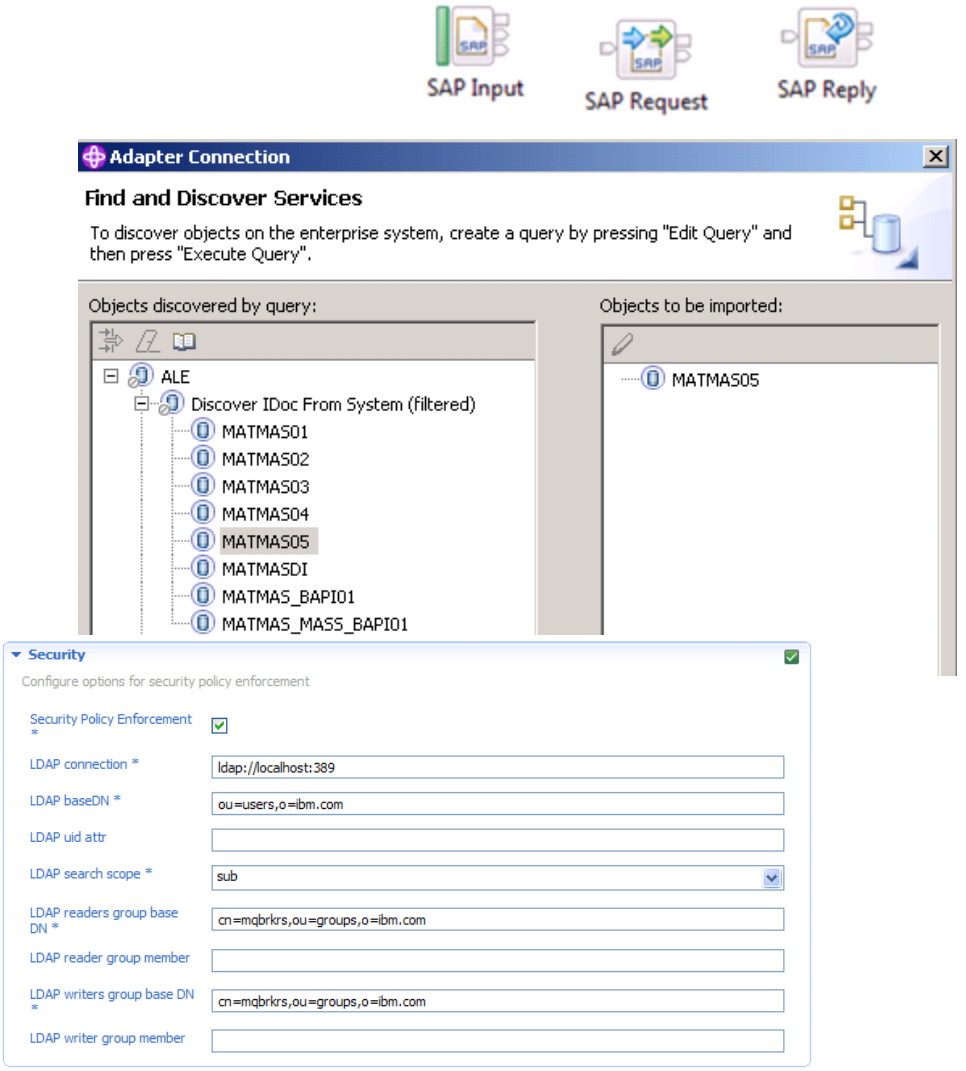






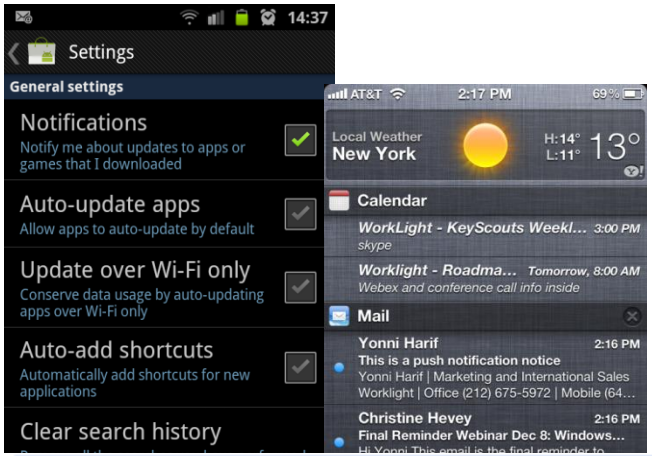
Pattern 3 - Allow mobile apps controlled access to enterprise data

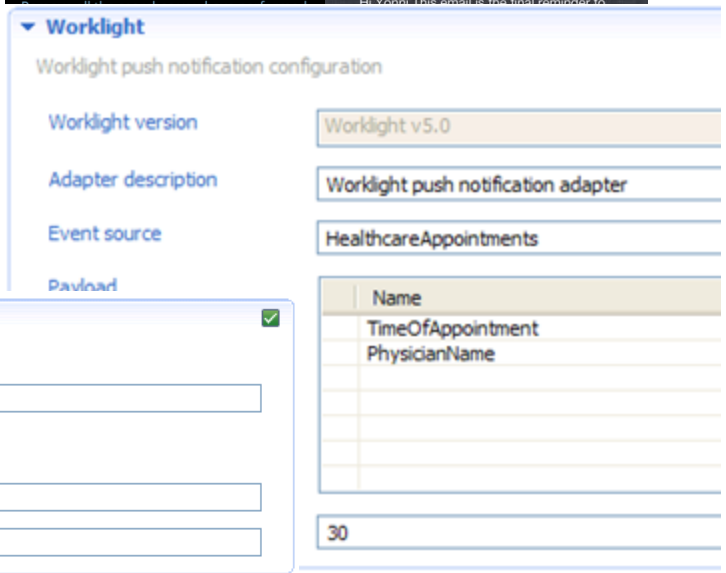
- Simple to enable mobile applications to **Create, Read, Update and Delete** enterprise data
  - Message Broker has excellent support for a wide range of enterprise applications (SAP, Siebel, JDEdwards, PeopleSoft etc…)
  - Pattern generates Worklight adapter and stubs for implementing CRUD operations
- **Quickly configure security policy** to authorize and authenticate access via external LDAP provider
- **One click to cache read resources** in IBM WebSphere Extreme Scale
  - High performing data access crucial for large volumes of mobile devices



Pattern 4 - Push notification to mobile apps from within the enterprise

- IBM Worklight supports asynchronous push notifications to mobile applications
  - e.g. to deliver out of band messages such as special offers
- Pattern to **rapidly enable enterprise services to send notifications**
  - e.g. MQ, SAP, Database, Medical system, etc…
  - Generates Web service and Worklight adapter to deliver notifications to mobile applications



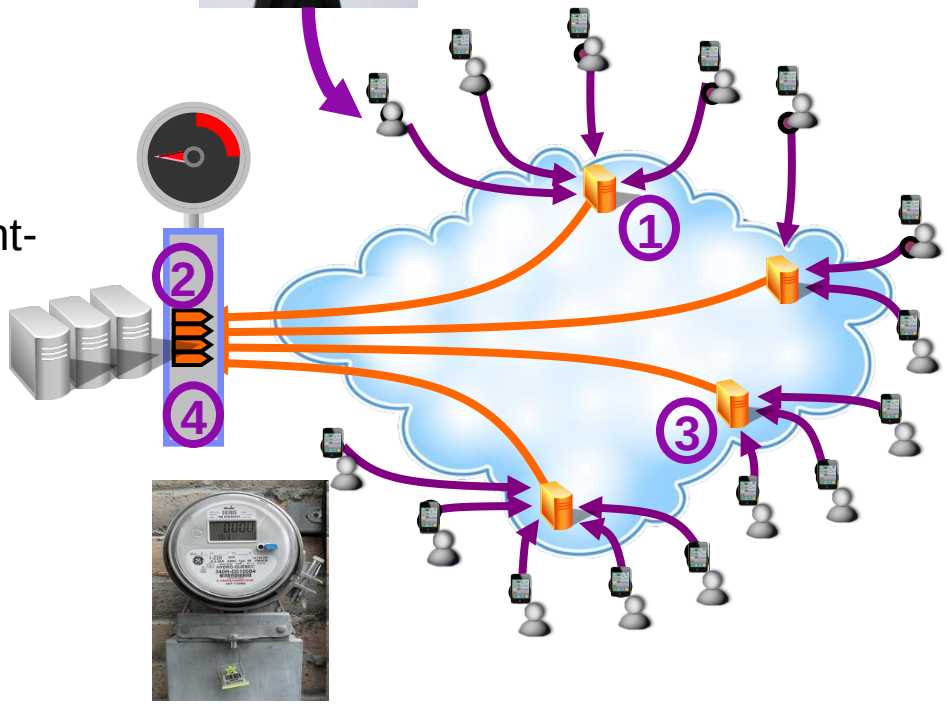


Agenda

- Mobile apps and the enterprise
- IBM Enterprise Mobile Platform
- Integrating with Enterprise Applications
- **Mobile Messaging**
- Summary

Connecting using more than just HTTP?

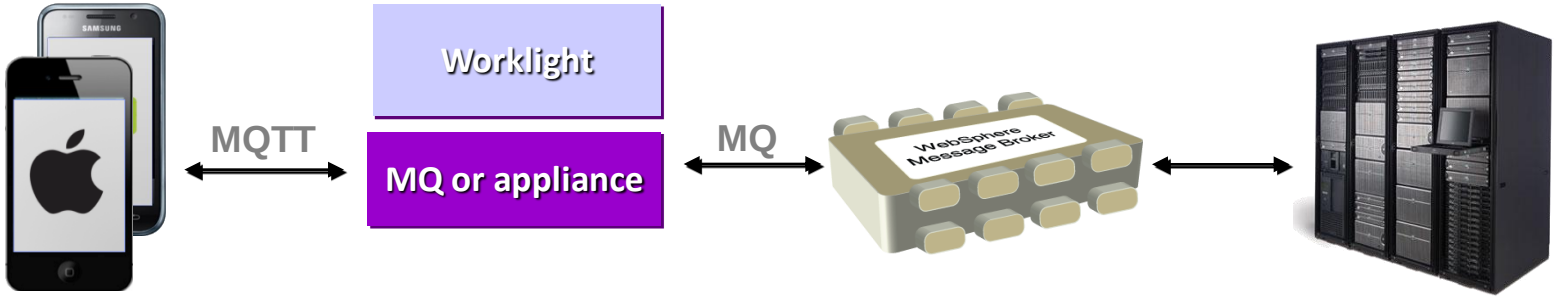
- The HTTP standard revolutionized how *people* consume data
  - A single simple model: Send a request, read the response
  - Available via any tablet, laptop, phone, PC etc. ideal for requesting data from a known source
    - Mobile user sending or requesting info
- BUT when a mobile user submits a *transaction* or *distribution needs to be optimized across MANY users….* an event-oriented paradigm can be a game changer:
  - Publishing information *one to many*
  - Listening for events *whenever they happen*
  - Distributing minimal packets of data in *huge volumes*
  - *Pushing* information over *unreliable networks*



Key challenges for Mobile apps

- Volume (cost) of data being transmitted (especially in M2M with limited data plans)
- Power consumption (battery powered devices)
- Responsiveness (near-real time delivery of information)
- Reliable delivery over fragile connections
- Security and privacy
- Scalability

Mobile message exchange patterns – beyond simple request/response

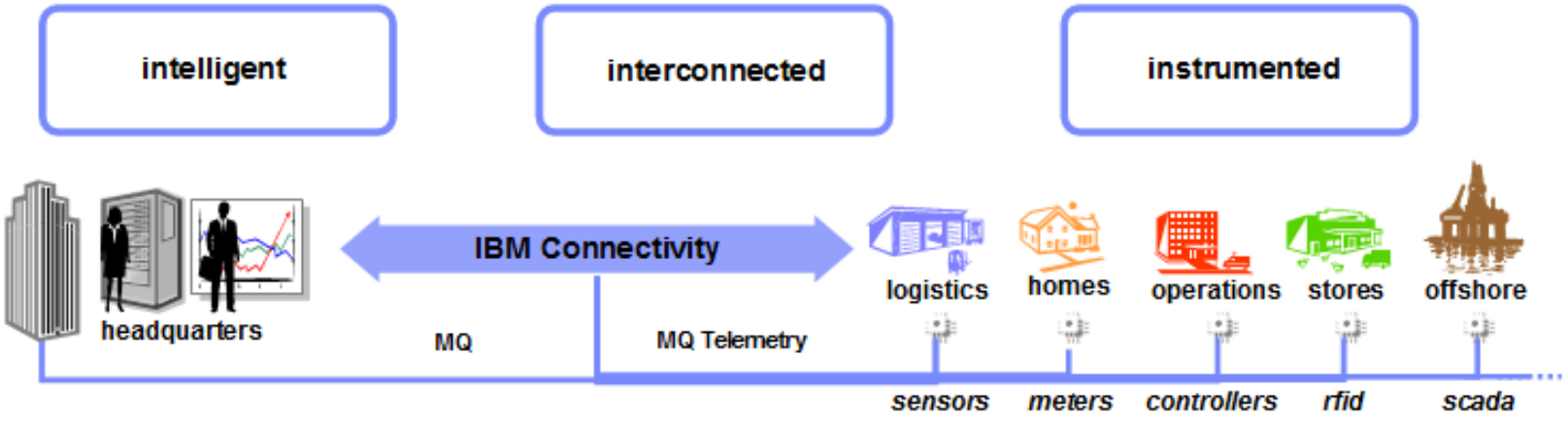


Reliable asynchronous transactions	User submits a transaction. One or more responses may come back over time.	MQTT provides **reliability** and **store/forward** of requests and responses if needed – reducing the amount of application code
Continuous update of realtime information	Server-side data is "streamed" to the device and used to update the UI. In most cases this is only required when the app is in the foreground	Small MQTT header size reduces **battery consumption** and **network traffic. One->many** publish/subscribe reduces load on application
Notification	Sending alert or other informational message to the device. The app may or may not be running at the time.	Avoidance of polling reduces **battery consumption** and **network traffic**. **Store/forward** of important notifications if app/device is not contactable
Server-initiated request	Server-side application initiates a dialog with the user (e.g. to query suspicious credit card transaction)	Avoidance of polling reduces **battery consumption** and **network traffic**. **Store/forward** of request if app/device is not contactable
Collection of data from device	Data sent to the server coming either from User Interface, of from onboard sensors or from devices attached to the phone	Small MQTT header size reduces **battery consumption** and **network traffic. Store/forward** of messages. **One->many** publish/subscribe

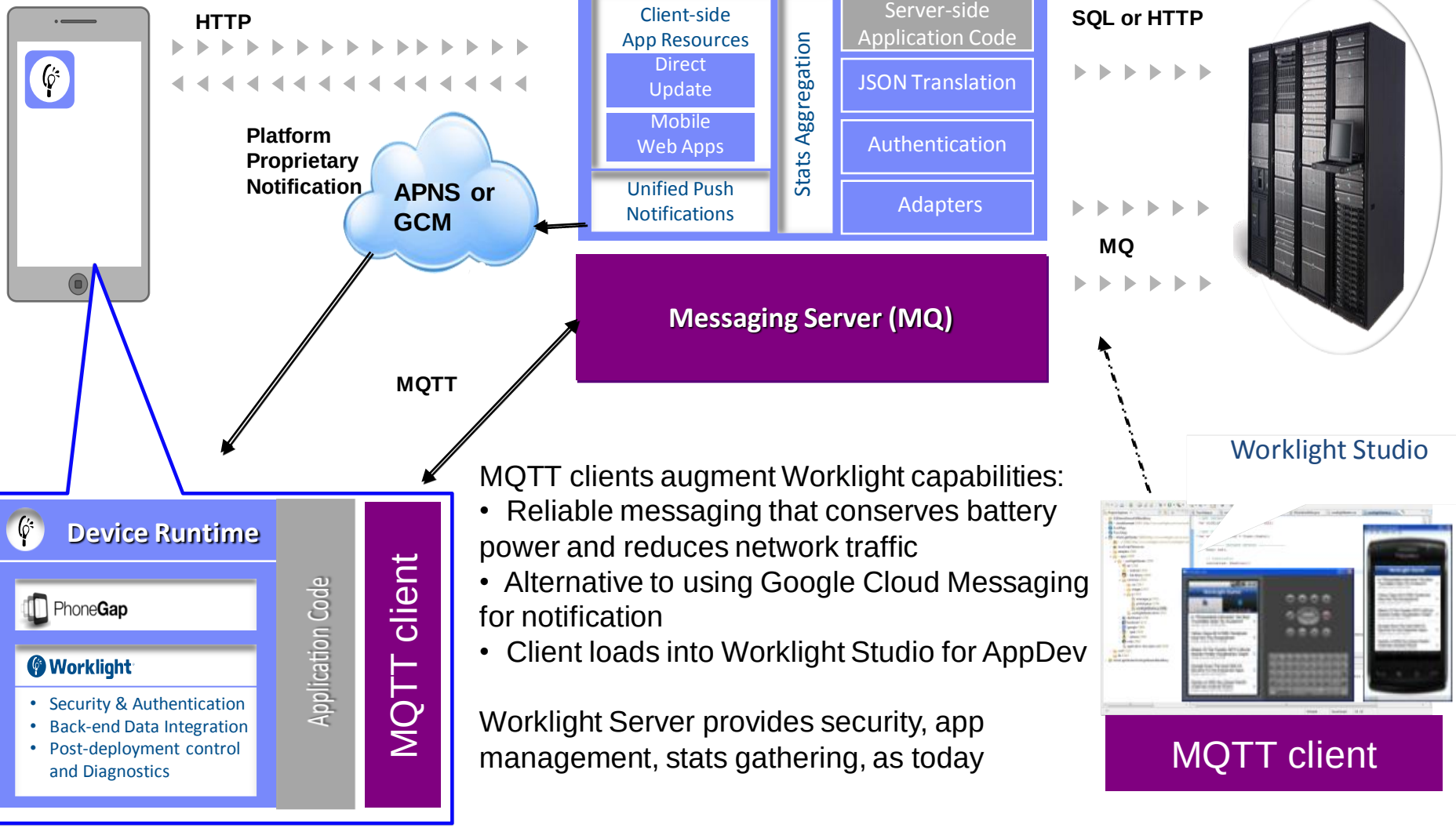
How do we attach Mobile devices? The Smarter Planet Protocol

- **WebSphere MQ Telemetry Transport (MQTT)**
  - Messaging optimized for mobile, smart sensors and telemetry devices
  - Enables intelligent decision-making based on remote real-world events
  - Remote resource management of static or moving assets, people, locations

- **An open standard with Industry leadership & mindshare**
  - MQTT Protocol and client code contributed to open source effort
  - see MQTT.org and Eclipse Paho
  - allows development communities to provide further client code & device support



MQ and Worklight

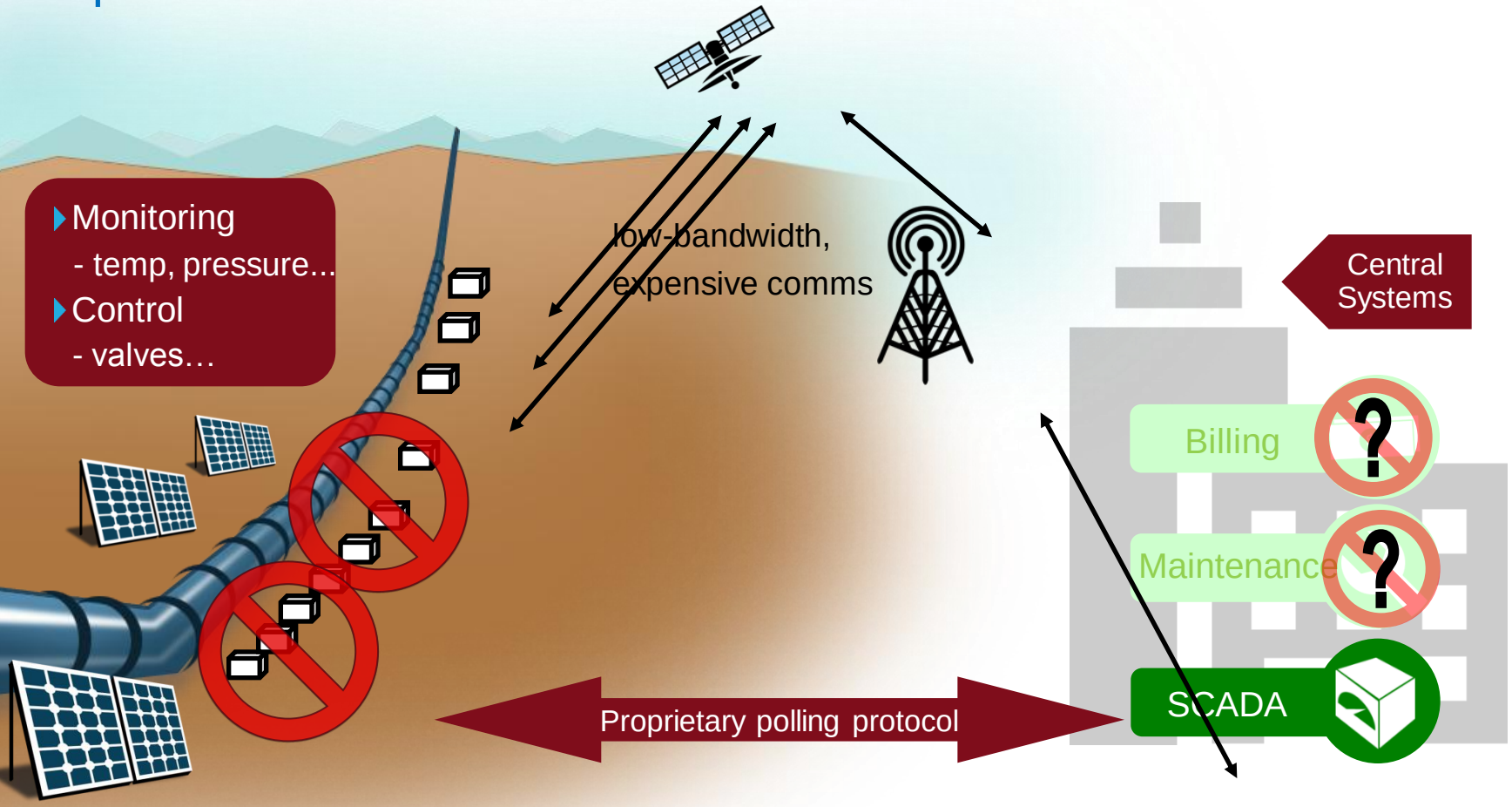


MQTT clients augment Worklight capabilities:

- Reliable messaging that conserves battery power and reduces network traffic
- Alternative to using Google Cloud Messaging for notification
- Client loads into Worklight Studio for AppDev

Worklight Server provides security, app management, stats gathering, as today

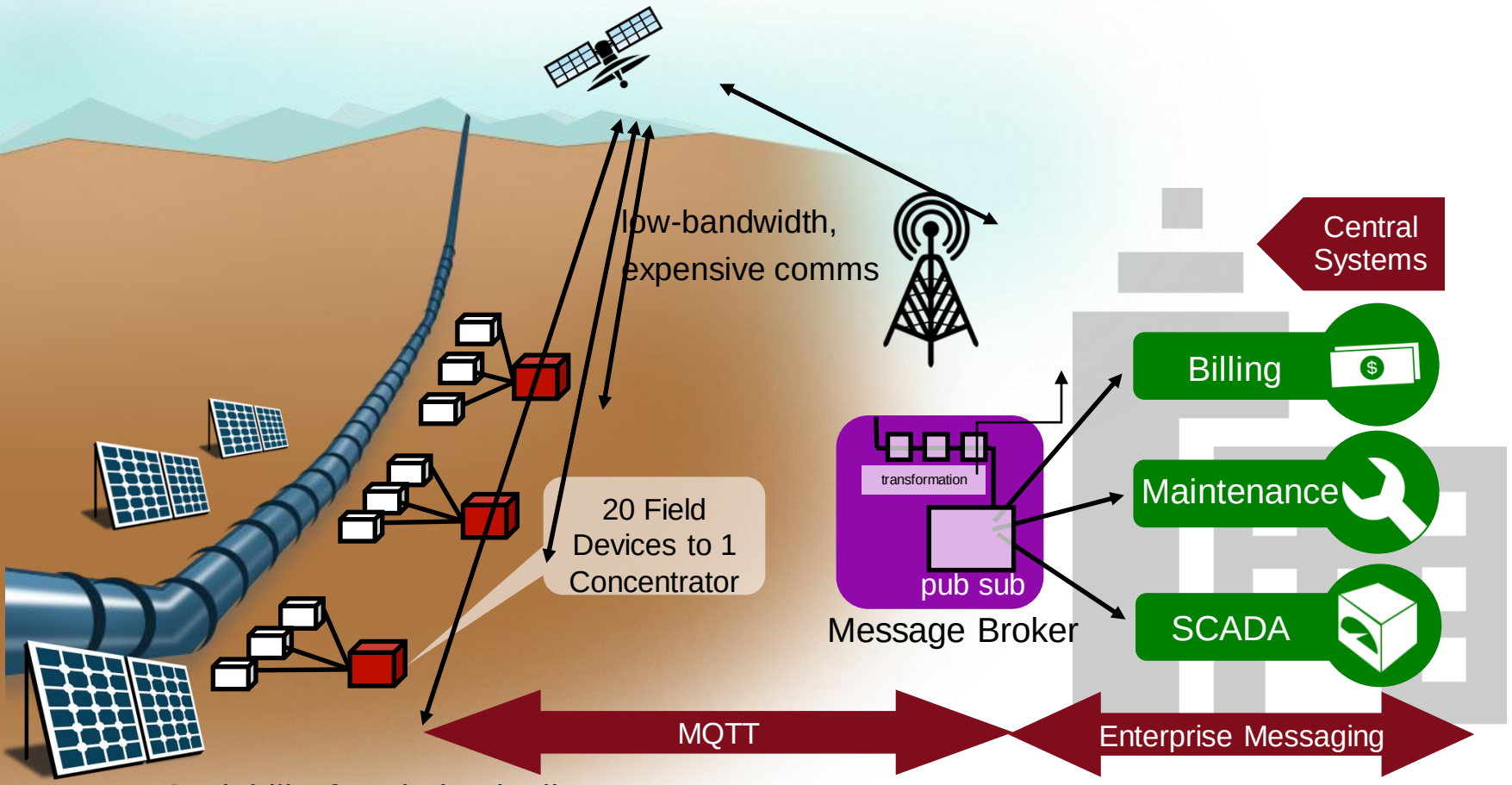
Pipeline – the need for scalable communications



4000 devices integrated, need to add 8000 more BUT:

- Satellite network saturated due to polling of device
- VALMET system CPU at 100%
- Other applications needed access to data ("SCADA prison")

Enter MQTT



Scalability for whole pipeline!
Network traffic much lower - events pushed to/from devices and report by exception
Network cost reduced
Lower CPU utilization
Broken out of the SCADA prison – data accessible to other applications

MQTT: Key Features

Open

- Open published spec designed for the world of “devices”
  - MQTT client code (C and Java) donated to the Eclipse "Paho" M2M project

Lean

- Minimized on-the-wire format
  - Smallest possible packet size is 2 bytes
  - No application message headers
- Reduced complexity/footprint
  - Clients: C=50Kb; Java=100Kb

Reliable

- Three qualities of service:
  - 0 – at most once delivery
  - 1 – assured delivery but may be duplicated
  - 2 – once and once only delivery
- In-built constructs to support loss of contact between client and server.
  - “Last will and testament” to publish a message if the client goes offline.
- Stateful “roll-forward” semantics and “durable” subscriptions.

Simple

- Simple / minimal pub/sub messaging semantics
  - Asynchronous (“push”) delivery
  - Simple set of verbs -- connect, publish, subscribe and disconnect.

Included in IBM WebSphere MQ v7.1 and 7.5

MQTT – the WebSphere MQ implementation

Massive Scale

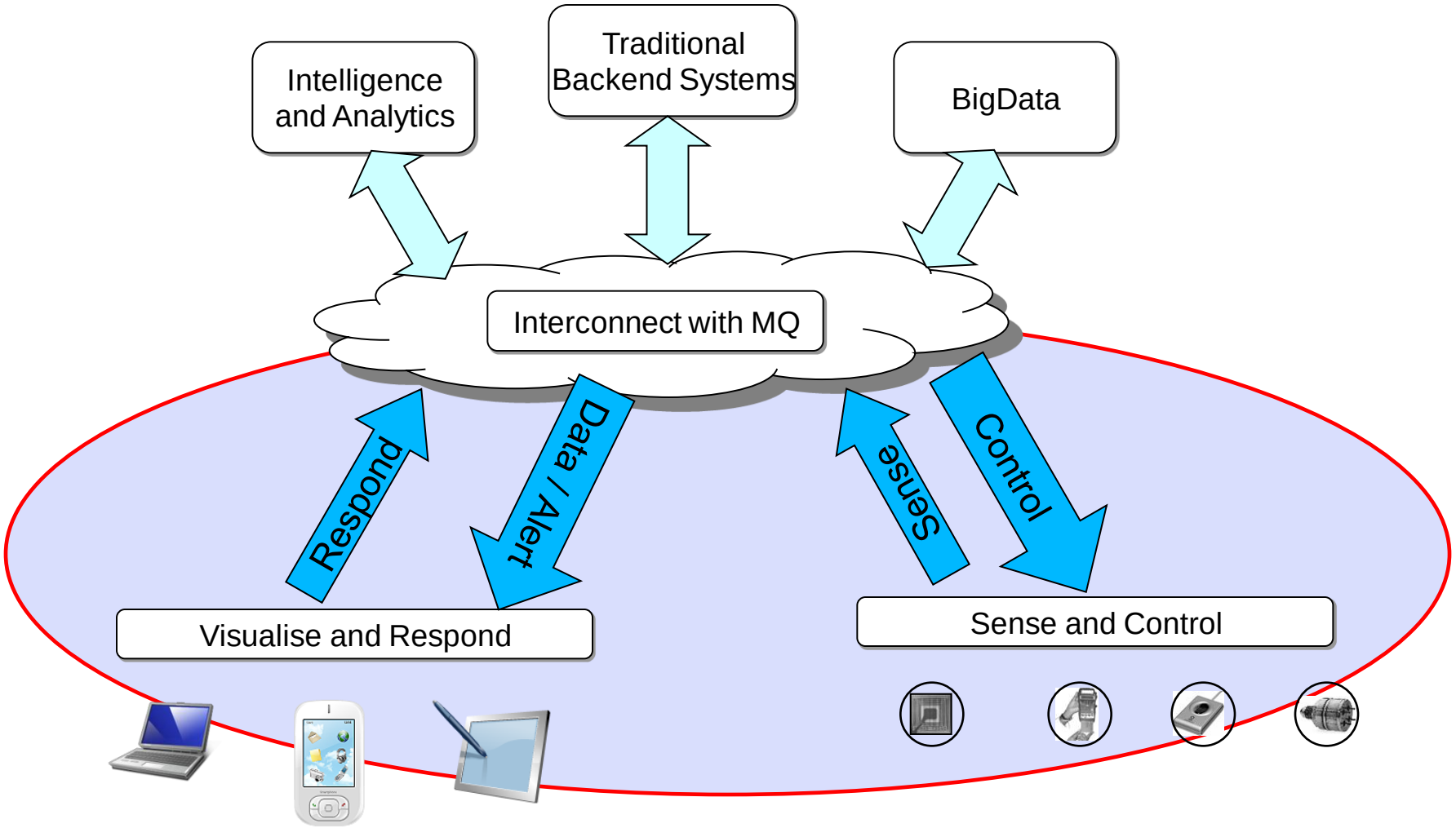
Included in IBM WebSphere MQ v7.1 and 7.5

- 240,000 concurrent clients tested with <5% CPU on a single IBM WebSphere MQ queue manager
  - Maxed out number of load testing client machines
- Interoperable with Enterprise Applications:
  - Seamless interchange with JMS or MQI applications

High Security

- Direct connection between your enterprise and devices
- Network: TLS/SSL
- Authentication: JAAS
- Authorization: OAM

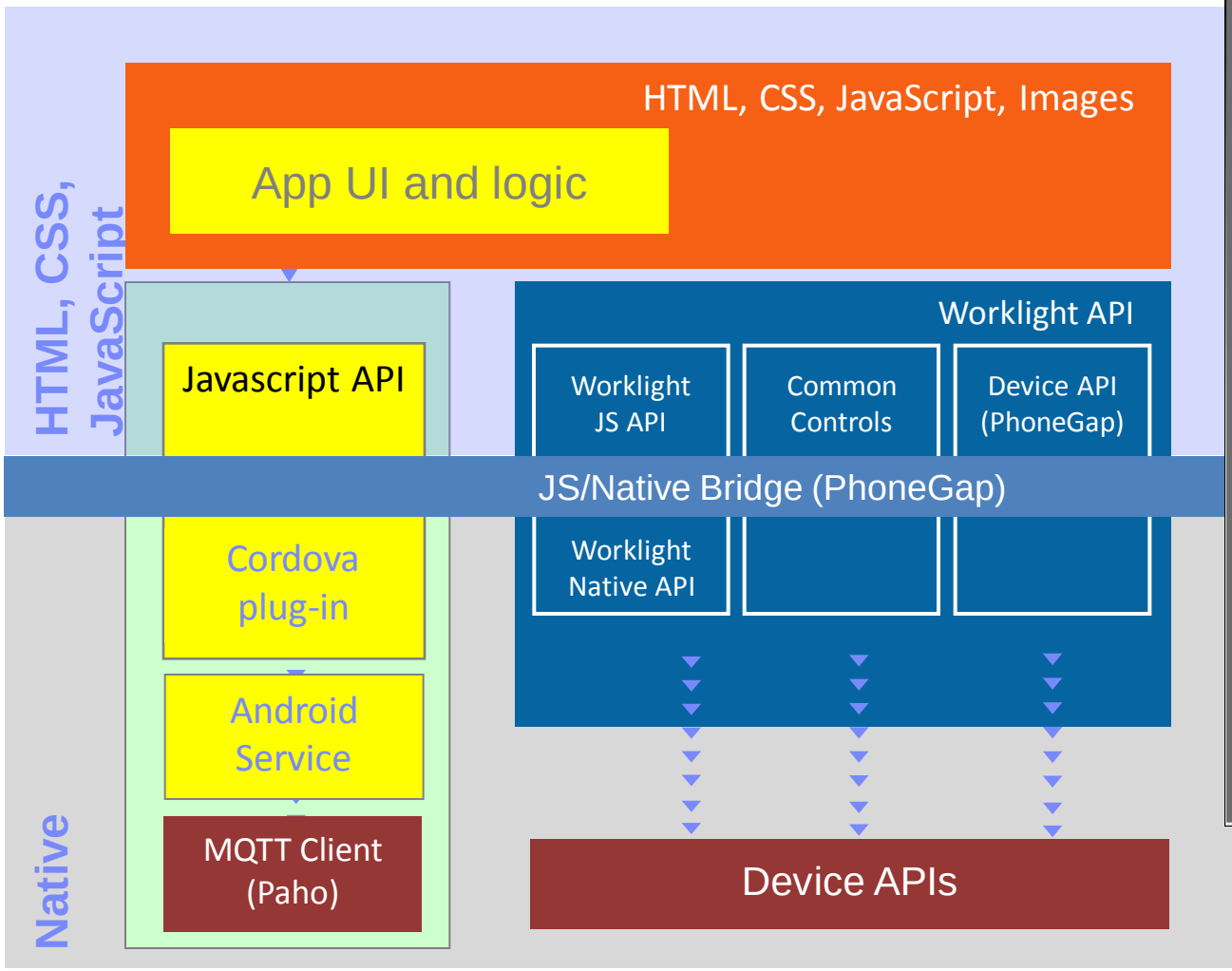
The Realm of MQTT



Providing Reliable Mobile device access to the Enterprise
with the new MQ Mobile client pack

- **Reliably extend your business to mobile devices**, building on the value of your messaging infrastructure
- With our **new Mobile client pack** available from the MQdev community on developerWorks
- Introduces a new **Javascript messaging API**
- Helps you design & deploy a **hybrid mobile app** for the Android platform using IBM Worklight Studio
- Underpinned by **IBM WebSphere MQ Telemetry Transport (MQTT)** providing
  - Ease of integration with enterprise applications
  - Reliable delivery over fragile connections
  - Access to information provided by smart devices
  - Reduced Power consumption
  - Security and privacy
  - Scalability

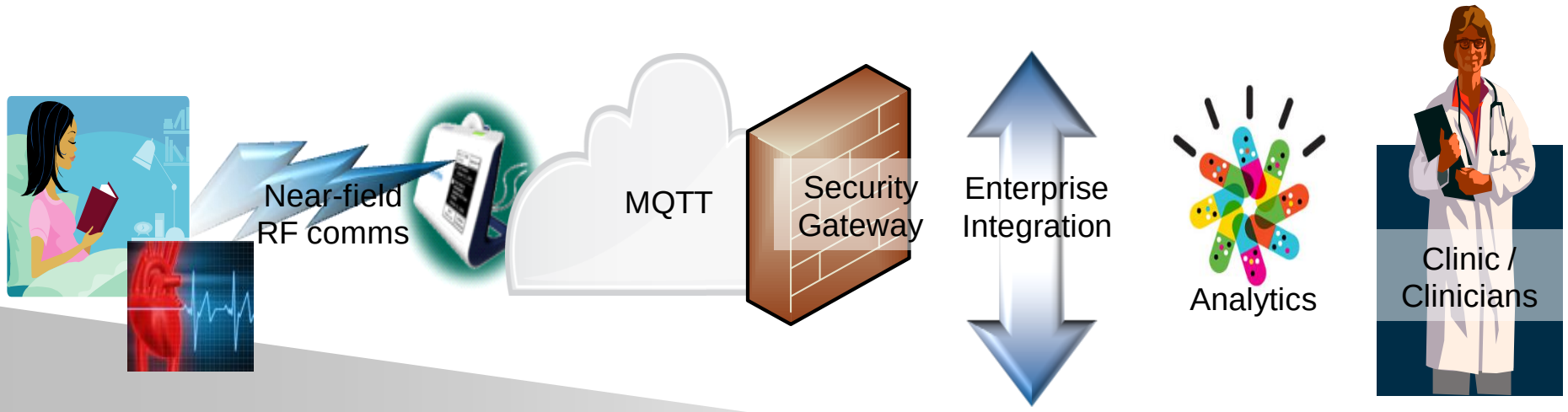
Client Pack Article & Sample Worklight App



http://www-01.ibm.com/support/docview.wss?uid=swg24033580

A real world Healthcare example

- Smart, connected, pacemakers eliminate the need for regular clinic visits
- Problems are detected early, preventing potentially life threatening incidents



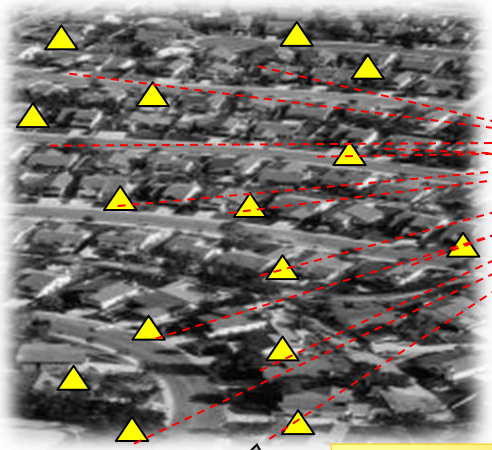
Smart pacemaker records data

Bedside appliance extracts information

Analytics build patient knowledge and extract insight

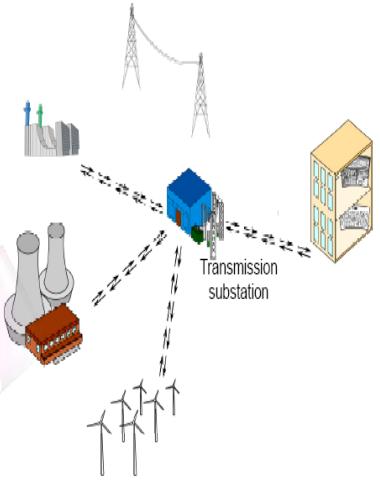
Professionals perform the action

A real world Energy example



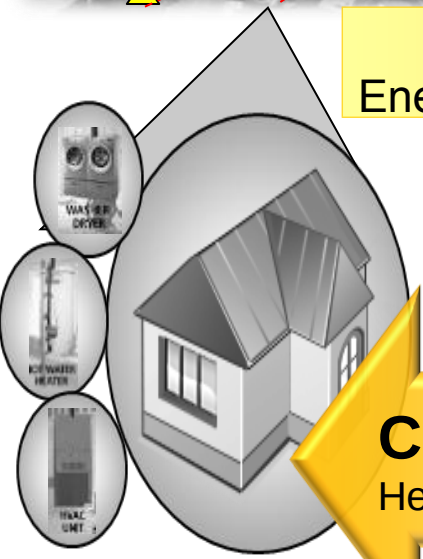
Demand Trends

Capacity



Usage
Energy, Temp…

MQTT

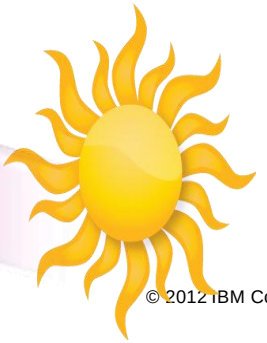


Virtual Power Plant

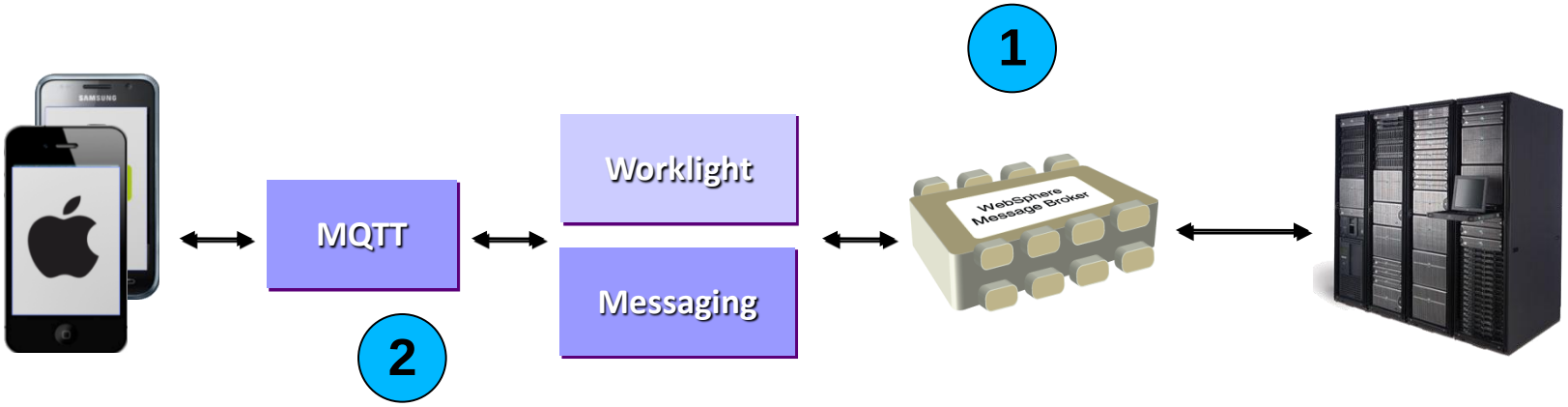
Futures Market

Control
Heating, Appliances

Weather



Summary



1. IBM Worklight and Integration of Worklight apps with cloud and enterprise applications
   - IBM WebSphere Cast Iron for cloud integration (included in IBM Mobile Foundation 2Q12)
   - IBM WebSphere Message Broker Patterns using HTTP (shipped in 8.0.0.1)
2. Reliable, asynch messaging with the mobile devices
   - Sample MQTT support for Worklight Hybrid Android apps (October 2012)
   - Messaging appliance (planned)
3. Collateral material
   - Tutorial material on developerWorks (available Oct 2012)
   - Community development via Eclipse Paho (ongoing)