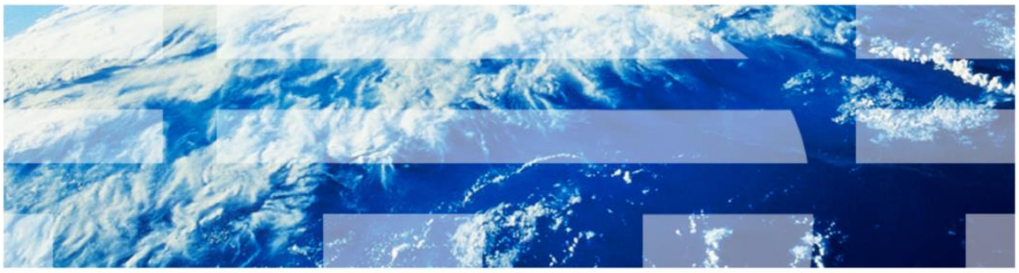
IBM Worklight

Backend connectivity



This presentation describes an overview for IBM Worklight's integration capabilities for backend system and cloud service connectivity. This presentation has been organized into three sections: background, connectivity to enterprise technologies, and common topologies and patterns.

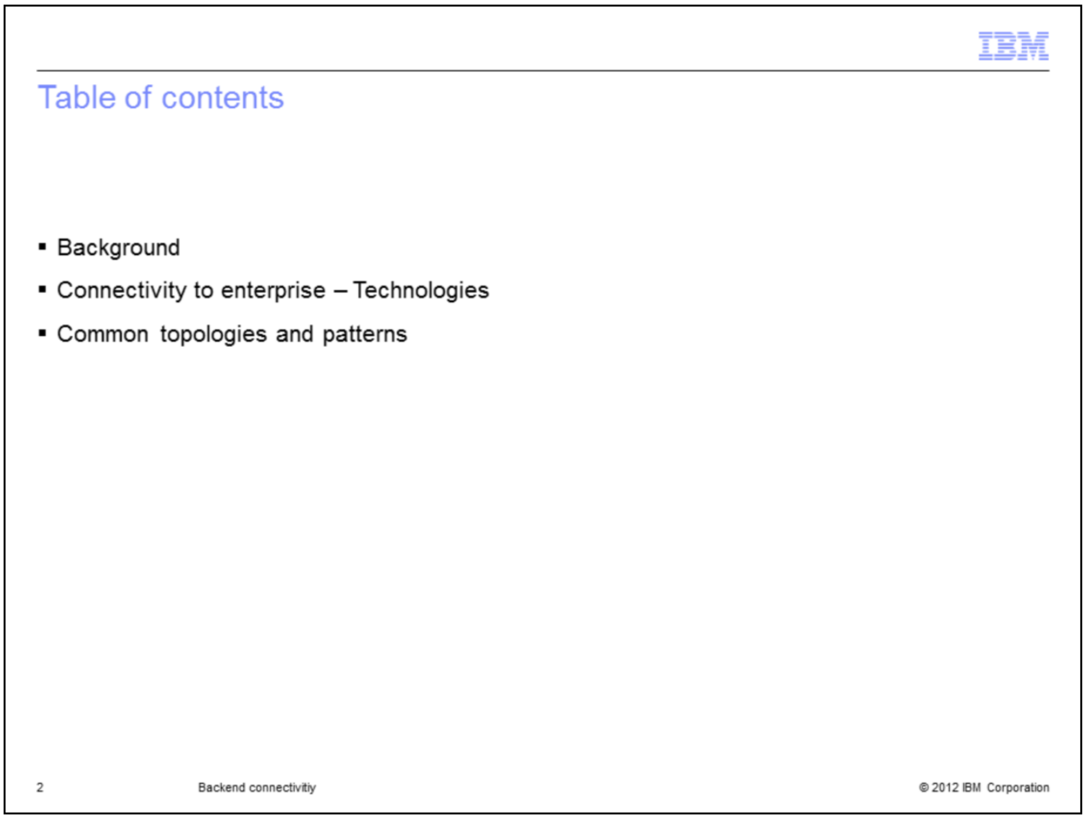
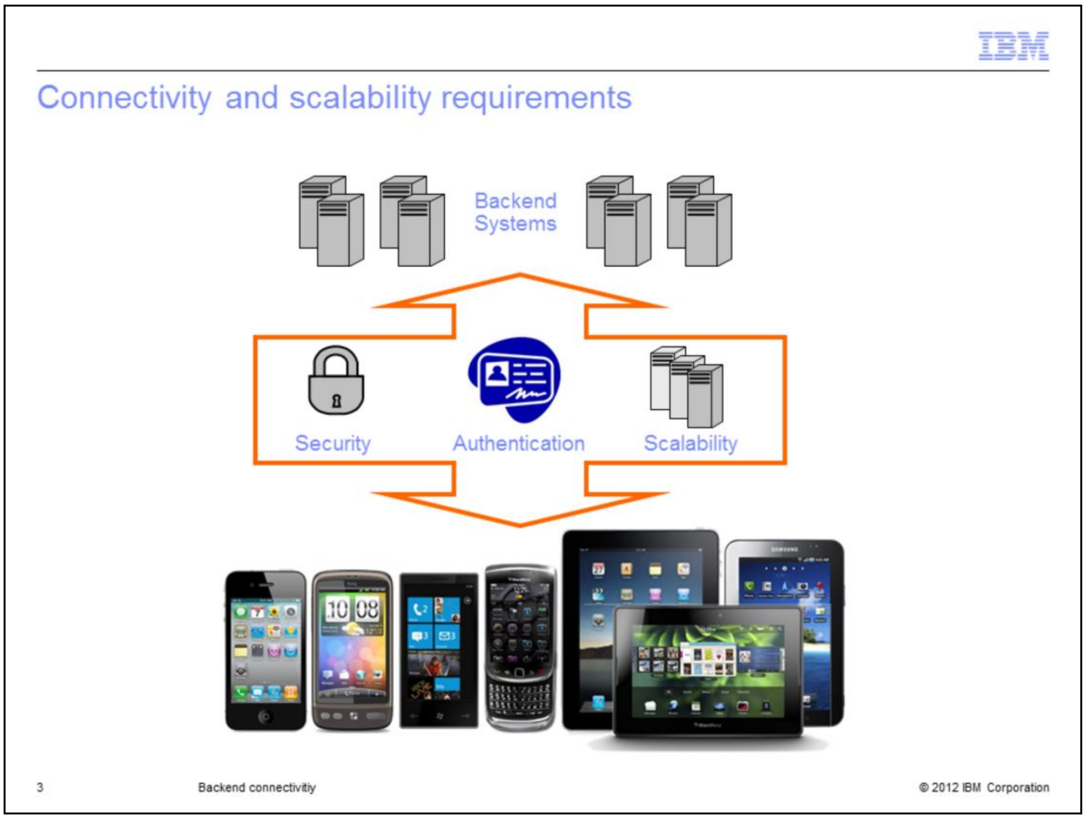
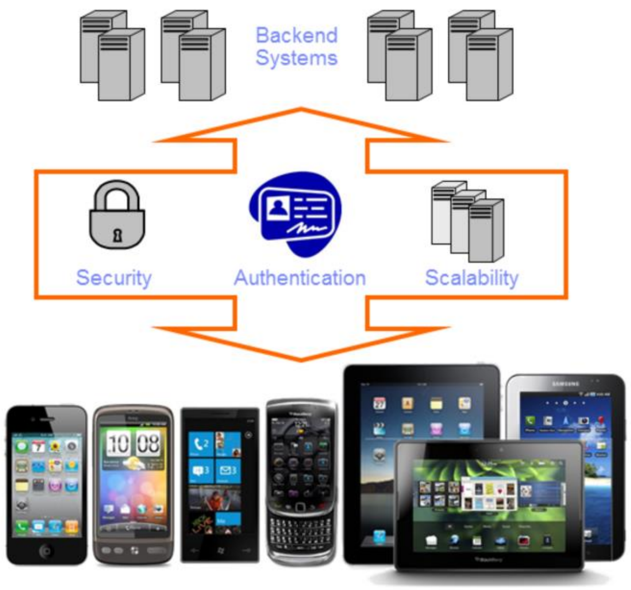


Table of contents

First, the background of IBM Worklight backend connectivity is discussed.

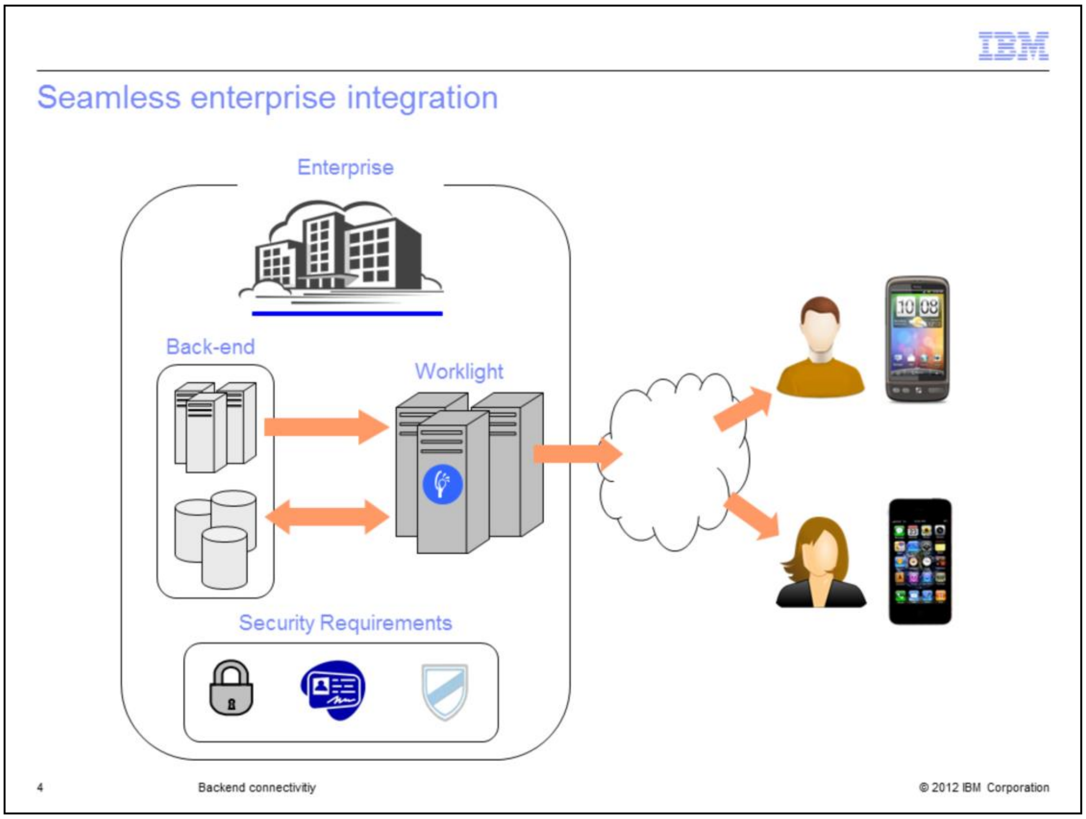


Connectivity and scalability requirements



3 Backend connectivitiy © 2012 IBM Corporation

Connecting the applications to backend systems presents multiple hurdles around security, authentication and scalability. As organizations mobilize more lines of business, controlling multiple applications in the wild and managing their ongoing performance can bring any IT department to their knees.



Seamless enterprise integration

Enterprise
Back-end
Worklight
Security Requirements

4 Backend connectivitiy © 2012 IBM Corporation

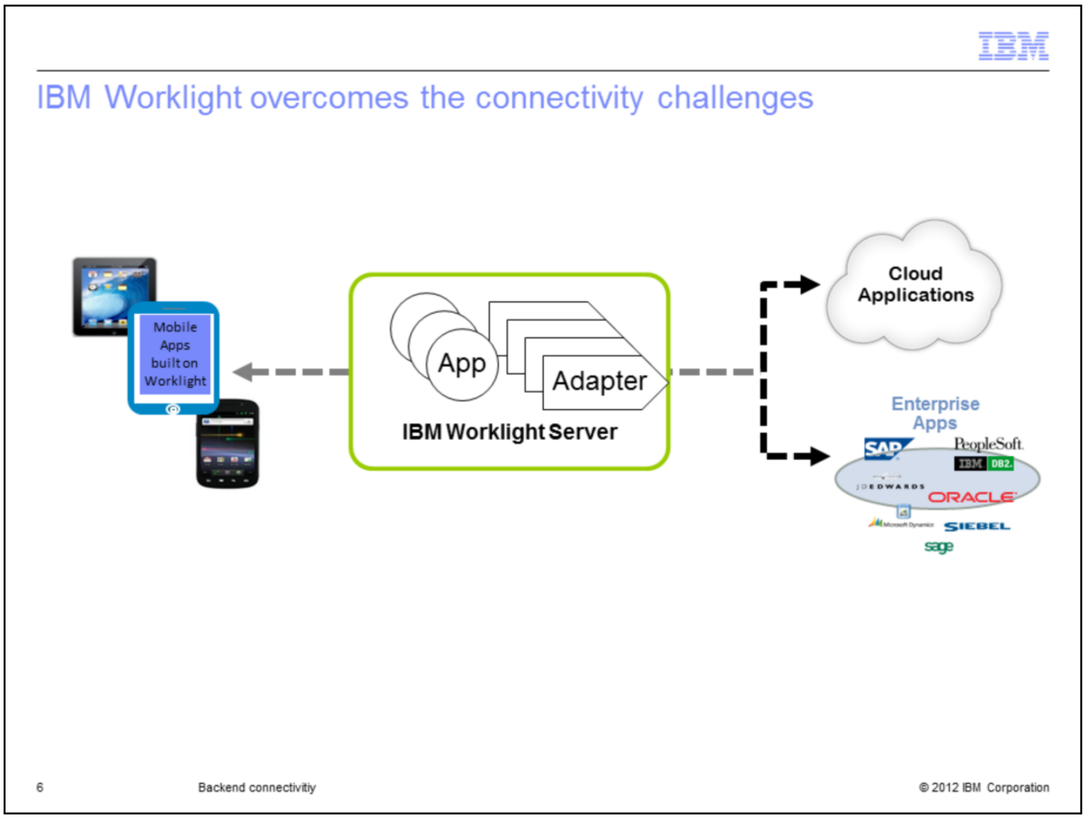
The IBM Worklight platform is designed to seamlessly integrate into the enterprise environment, leveraging its existing resources and adhere to the strictest security requirements of the organization. The platform channels back-end systems to the end-user, retrieves and updates data from multiple sources, and supports transactional capabilities and the invocation of different services and applications.

Section

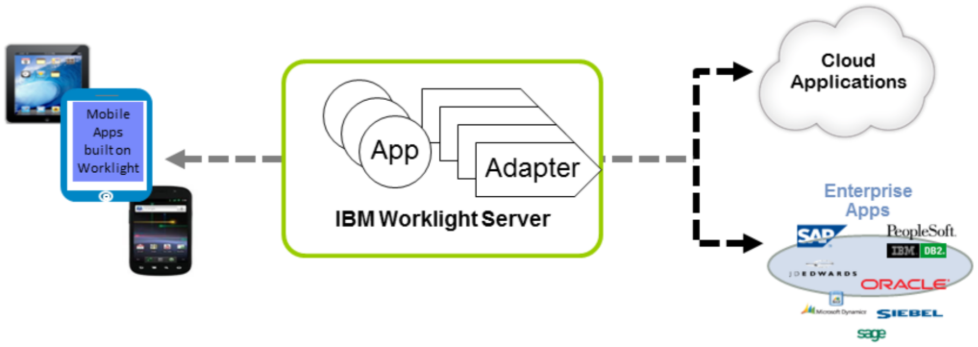
Connectivity to enterprise technologies

5 Backend connectivitiy © 2012 IBM Corporation

Next, connectivity to enterprise technologies is discussed.

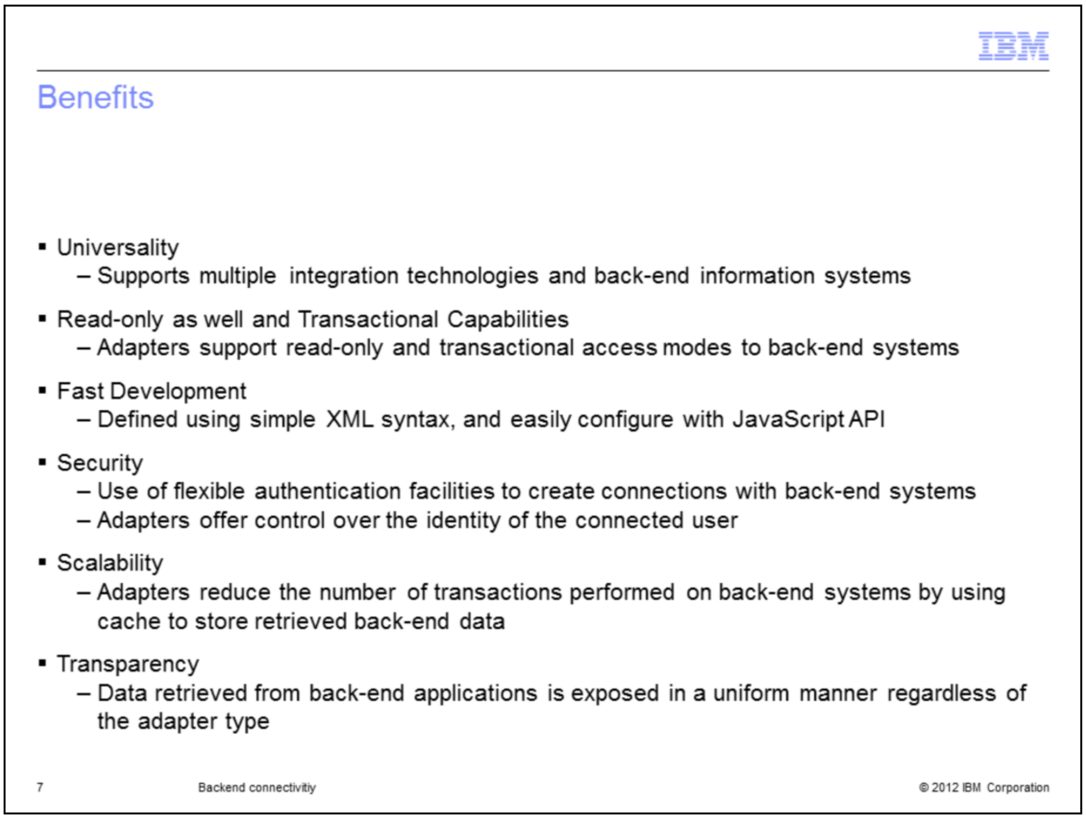


IBM Worklight overcomes the connectivity challenges



6 Backend connectivitiy © 2012 IBM Corporation

Worklight adapters address the needs of cloud and on-premise enterprise connectivity. These adapters are an integral part of the Worklight project structure as you will see in this presentation.

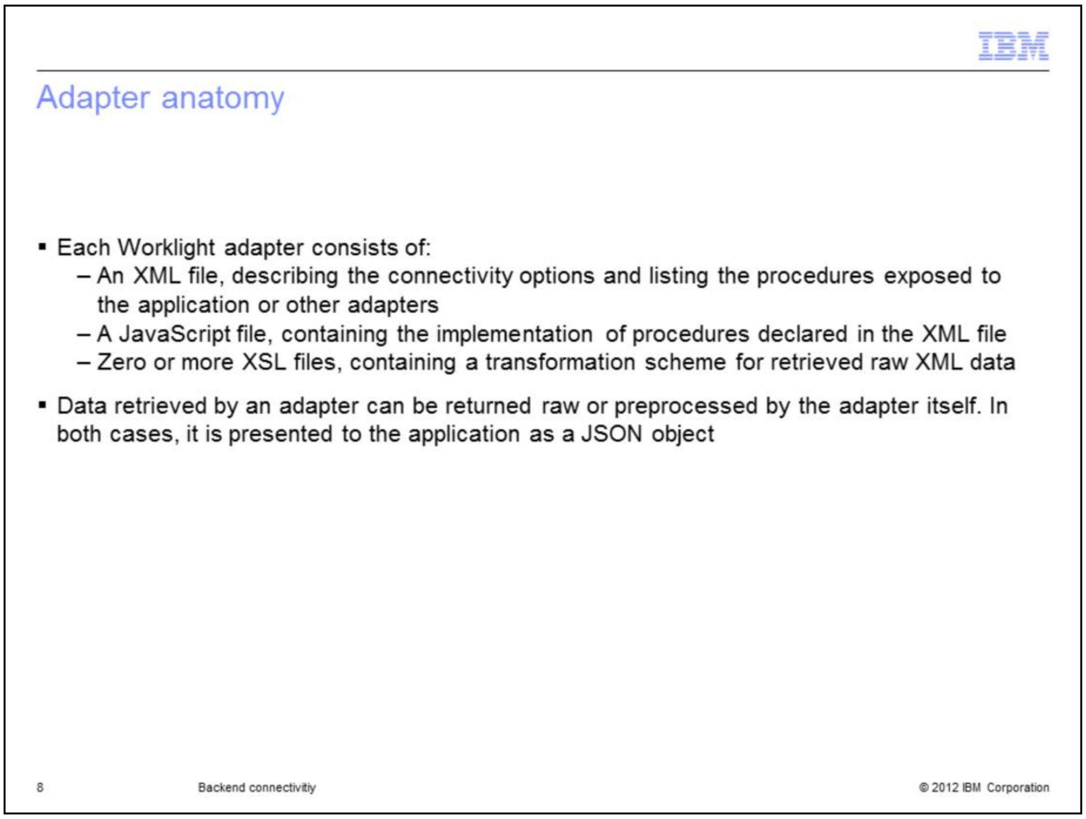


Benefits

- 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
  - Defined using simple XML syntax, and easily configure 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 reduce the number of transactions performed 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

7 Backend connectivitiy © 2012 IBM Corporation

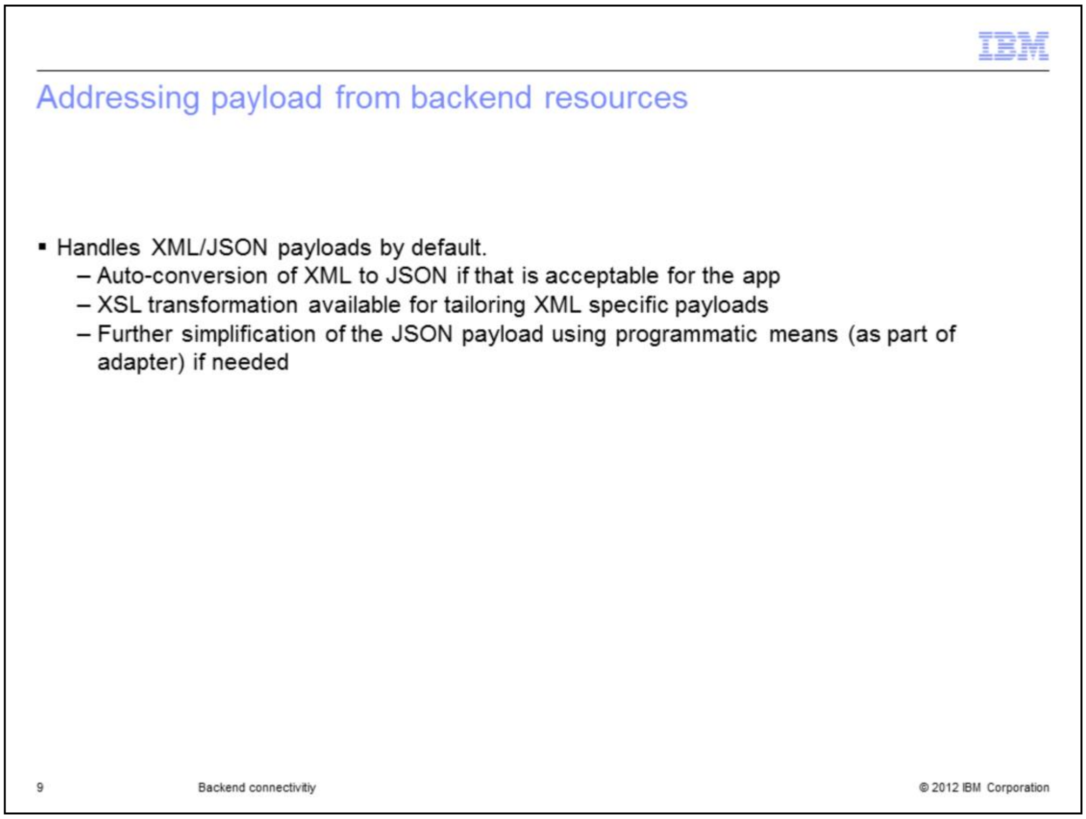
The adapters provide a fast development life cycle and is amenable to the iterative agile application development life cycle. Worklight adapters provide a simple declarative specification for defining backend connectivity with security and programmatic support for backend resource consumption. The development tools use the Worklight Studio to create adapters and using Javascript to write any custom server-side logic. There is also support for simplified testing of the adapters to ensure the behavior is consistent when used from within Worklight application.


Adapter anatomy

- Each Worklight adapter consists of:
  - An XML file, describing the connectivity options and listing the procedures exposed to the application or other adapters
  - A JavaScript file, containing the implementation of procedures declared in the XML file
  - Zero or more XSL files, containing a transformation scheme for retrieved raw XML data
- Data retrieved by an adapter can be returned raw or preprocessed by the adapter itself. In both cases, it is presented to the application as a JSON object

8 Backend connectivitiy © 2012 IBM Corporation

Each adapter is defined by an XML file with custom application logic in Javascript and additional ability to process XML payload using custom stylesheets. The primary means of communication between the adapter (on server) and the application (on client) is over HTTP's and using JSON.

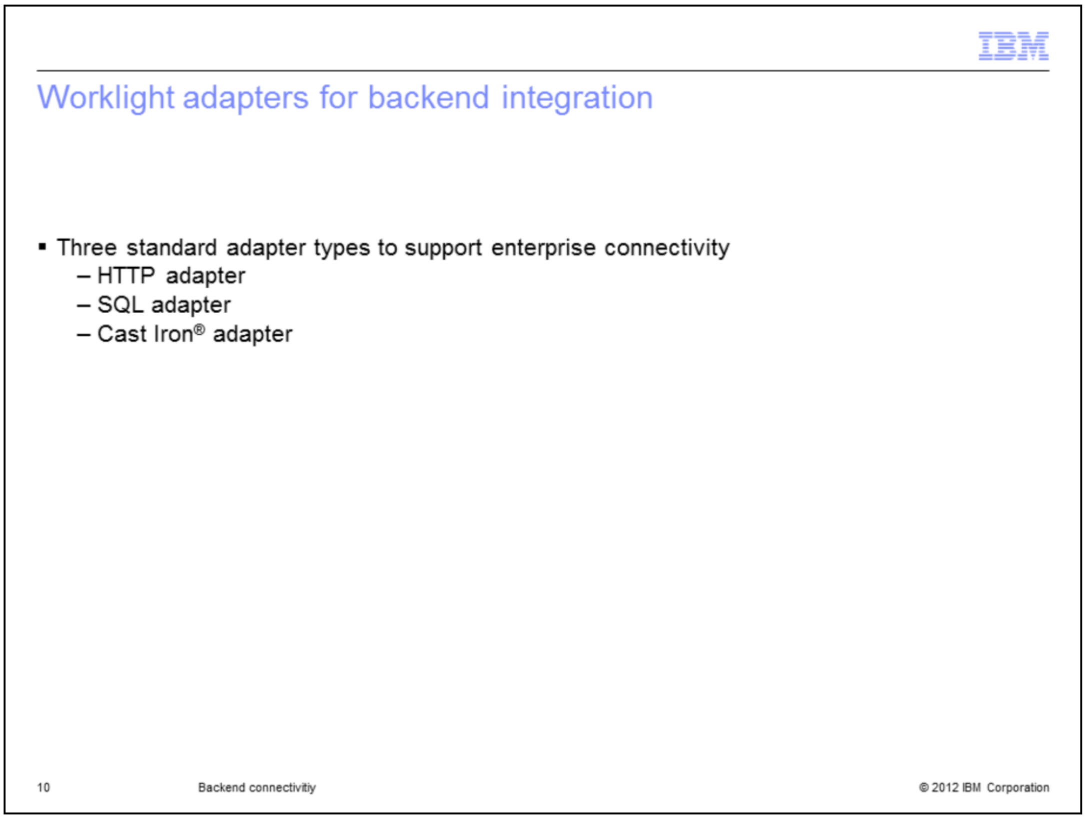


Addressing payload from backend resources

- Handles XML/JSON payloads by default.
  - Auto-conversion of XML to JSON if that is acceptable for the app
  - XSL transformation available for tailoring XML specific payloads
  - Further simplification of the JSON payload using programmatic means (as part of adapter) if needed

9 Backend connectivitiy © 2012 IBM Corporation

The adapter converts all XML responses to JSON by default, but can be further fine tuned using one of two ways – either programmatically using Javascript or using a custom stylesheet and the XSL processing.

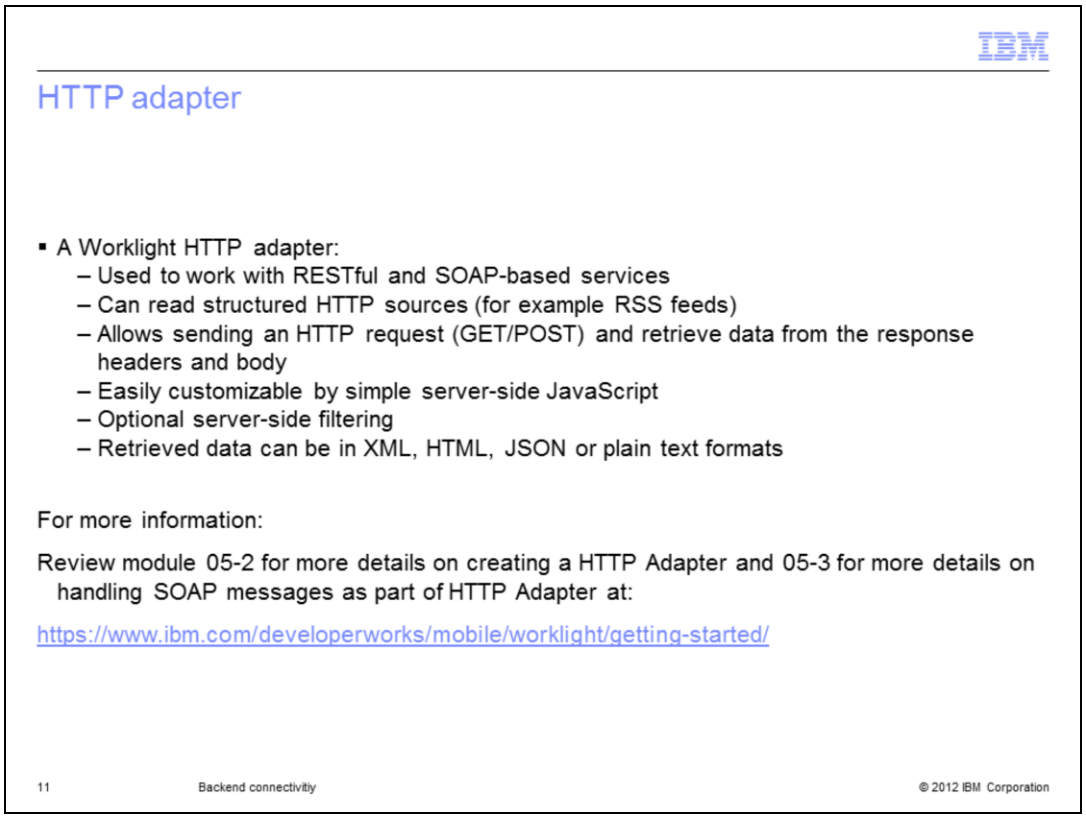


Worklight adapters for backend integration

- Three standard adapter types to support enterprise connectivity
  - HTTP adapter
  - SQL adapter
  - Cast Iron® adapter

10 Backend connectivitiy © 2012 IBM Corporation

There are three types of adapter types available: HTTP Adapter – for any HTTP based endpoint – RESTful or WS* based web services. SQL Adapter – for any database that supports JDBC. Cast Iron Adapter – for Cast Iron integration either in the cloud (Cast Iron Live) or on-premise (Cast Iron Hypervisor or Cast Iron appliance).



HTTP adapter

- A Worklight HTTP adapter:
  - Used to work with RESTful and SOAP-based services
  - Can read structured HTTP sources (for example RSS feeds)
  - Allows sending an HTTP request (GET/POST) and retrieve data from the response headers and body
  - Easily customizable by simple server-side JavaScript
  - Optional server-side filtering
  - Retrieved data can be in XML, HTML, JSON or plain text formats

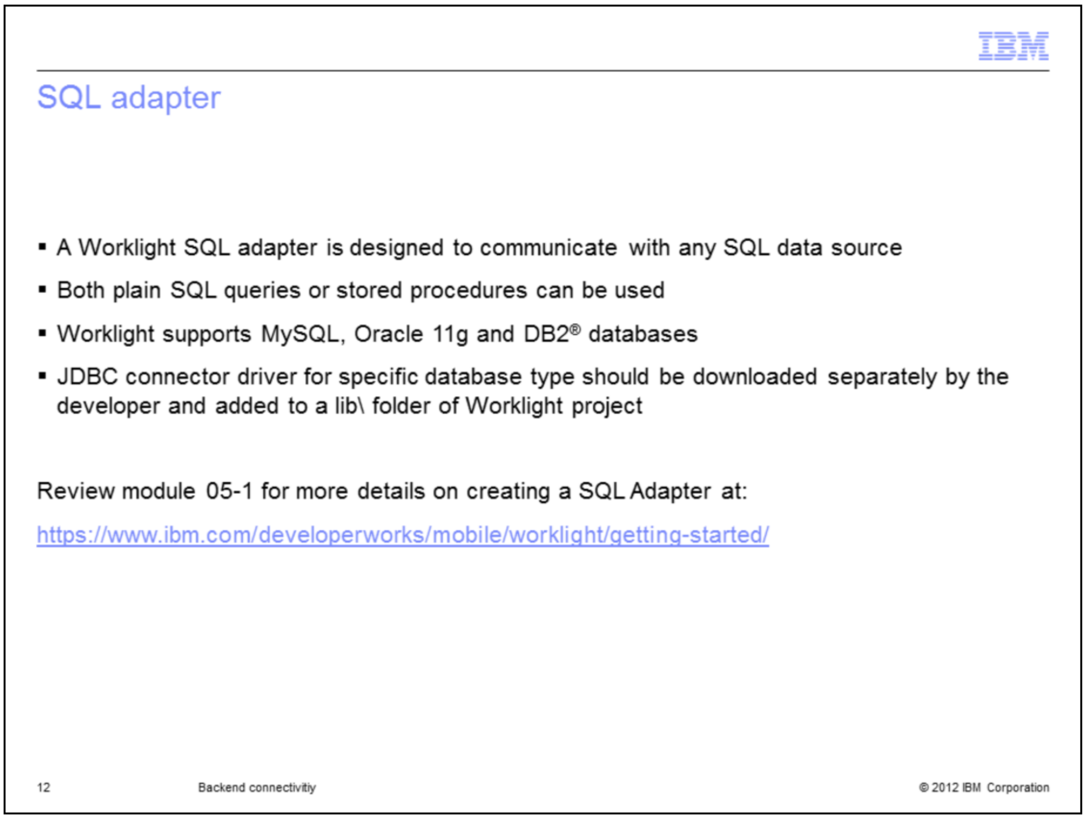
For more information:

Review module 05-2 for more details on creating a HTTP Adapter and 05-3 for more details on handling SOAP messages as part of HTTP Adapter at:

https://www.ibm.com/developerworks/mobile/worklight/getting-started/

11 Backend connectivitiy © 2012 IBM Corporation

As the name states, the Worklight HTTP adapter is used to connect and consume HTTP endpoints whether they are REST based or SOAP based. Some of the examples are RSS feeds, enterprise WSDL based web services and custom REST services supporting JSON. Review module 05-2 for more details on creating a HTTP Adapter and 05-3 for more details on handling SOAP messages as part of HTTP Adapter.



SQL adapter

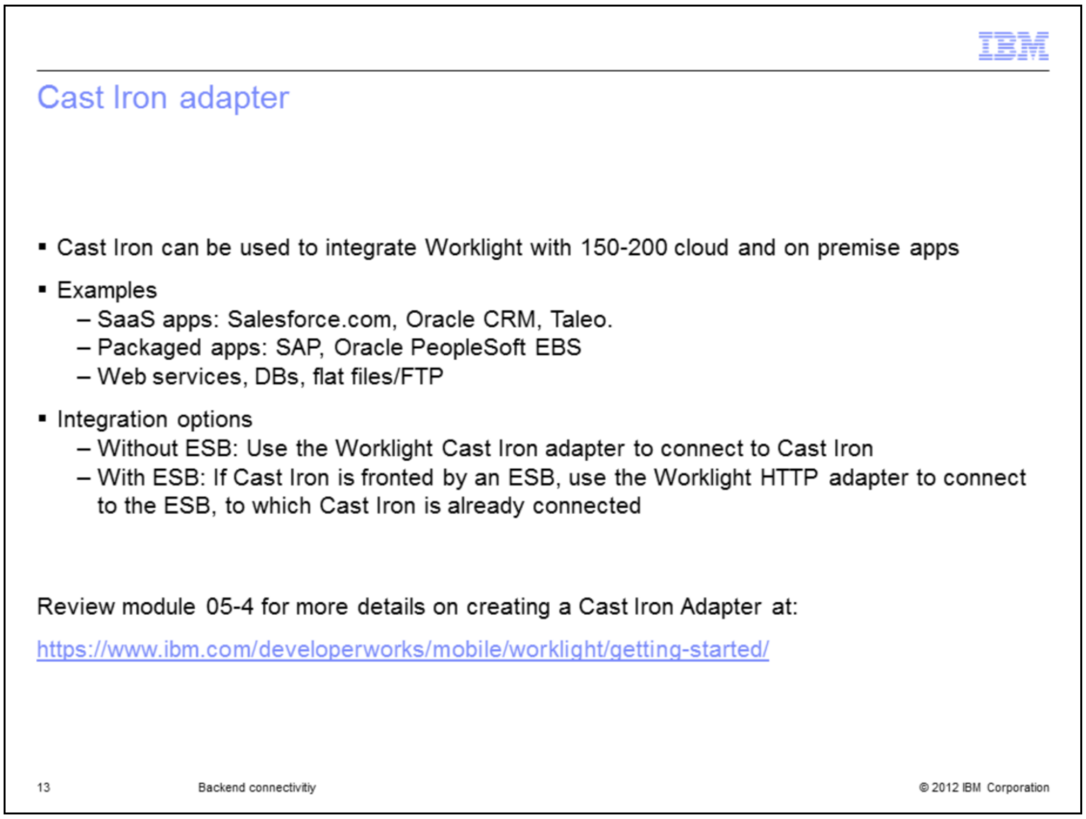
- A Worklight SQL adapter is designed to communicate with any SQL data source
- Both plain SQL queries or stored procedures can be used
- Worklight supports MySQL, Oracle 11g and DB2® databases
- JDBC connector driver for specific database type should be downloaded separately by the developer and added to a lib\ folder of Worklight project

Review module 05-1 for more details on creating a SQL Adapter at:

https://www.ibm.com/developerworks/mobile/worklight/getting-started/

12 Backend connectivitiy © 2012 IBM Corporation

As the name suggests Worklight SQL adapter is designed to communicate with any SQL data source using straight forward SQL queries or stored procedures. Both plain SQL queries or stored procedures can be used. The SQL adapter supports MySQL, Oracle and DB2. Review module 05-1 for more details on creating a SQL Adapter.



Cast Iron adapter

- Cast Iron can be used to integrate Worklight with 150-200 cloud and on premise apps
- Examples
  - SaaS apps: Salesforce.com, Oracle CRM, Taleo.
  - Packaged apps: SAP, Oracle PeopleSoft EBS
  - Web services, DBs, flat files/FTP
- Integration options
  - Without ESB: Use the Worklight Cast Iron adapter to connect to Cast Iron
  - With ESB: If Cast Iron is fronted by an ESB, use the Worklight HTTP adapter to connect to the ESB, to which Cast Iron is already connected

Review module 05-4 for more details on creating a Cast Iron Adapter at:

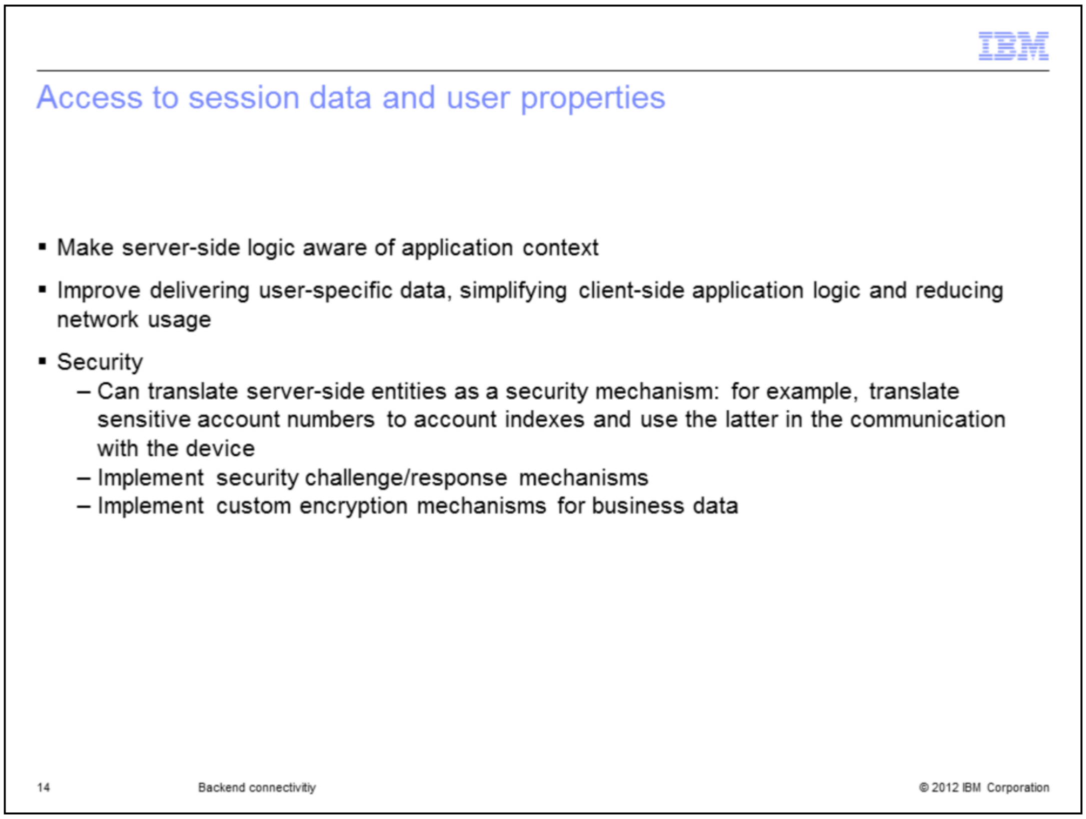
https://www.ibm.com/developerworks/mobile/worklight/getting-started/

13 Backend connectivitiy © 2012 IBM Corporation

IBM WebSphere® Cast Iron provides enterprise and Cloud connectivity to over 200 types of applications by using a simple diagrammatic programming model. In addition, Cast Iron has on-premise based software or hardware form-factor and a cloud instance to be accessed over the web. Some of the connectivity provided by Cast Iron include cloud services like Salesforce.com and existing enterprise apps like SAP, and Oracle.

IBM Worklight provides a Cast Iron adapter that helps connect to Cast Iron which then can be used to connect to any one of the backend resources supported by Cast Iron. Hence Worklight apps can deliver business critical applications to the mobile device leveraging this adapter type. All editions of IBM Worklight ship with this adapter type. But if you want to use the full power of the adapter, your enterprise needs to have access to a Cast Iron install (or on the cloud) or can be procured through the IBM Mobile Foundation enterprise and consumer edition which bundles Cast Iron with Worklight. A few deployment patterns that is covered in the next section of this presentation.

Review module 05-4 for more details on creating a Cast Iron Adapter



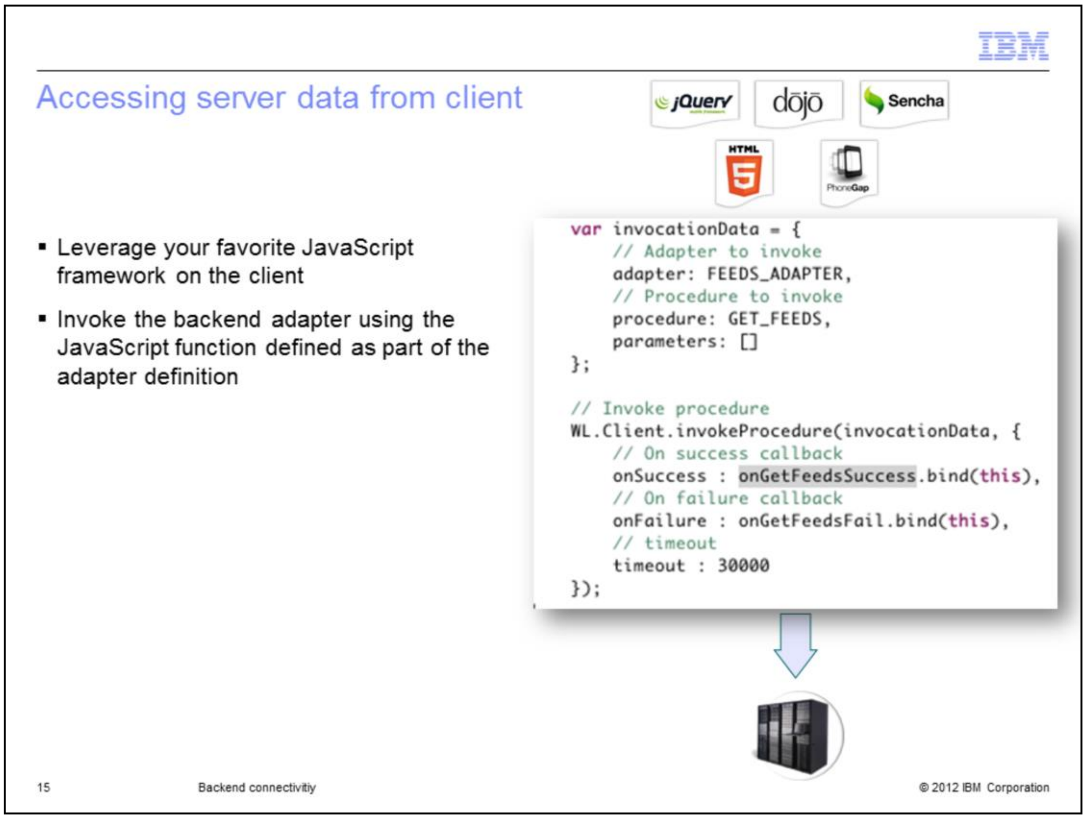
Access to session data and user properties

- Make server-side logic aware of application context
- Improve delivering user-specific data, simplifying client-side application logic and reducing network usage
- Security
  - Can translate server-side entities as a security mechanism: for example, translate sensitive account numbers to account indexes and use the latter in the communication with the device
  - Implement security challenge/response mechanisms
  - Implement custom encryption mechanisms for business data

14 Backend connectivitiy © 2012 IBM Corporation

Adapters have access to HTTP session data and user properties to bring application context to the server-side logic. This can be used to deliver personalized and targeted user experience while keeping the client-side logic simple.

The same session context can be used to facilitate other server-side security mechanisms such as specialized encryption, end to end single sign on and to protect privacy of the user.



Accessing server data from client

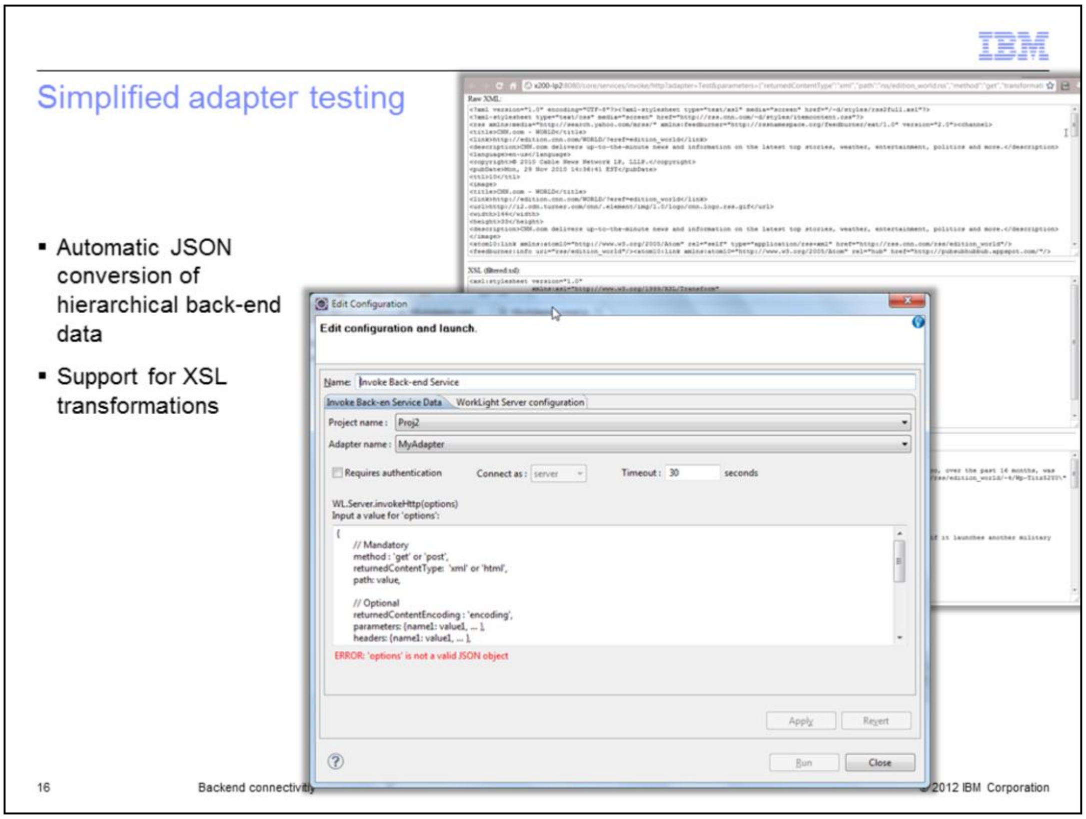
- Leverage your favorite JavaScript framework on the client
- Invoke the backend adapter using the JavaScript function defined as part of the adapter definition

```
var invocationData = {
    // Adapter to invoke
    adapter: FEEDS_ADAPTER,
    // Procedure to invoke
    procedure: GET_FEEDS,
    parameters: []
};

// Invoke procedure
WL.Client.invokeProcedure(invocationData, {
    // On success callback
    onSuccess : onGetFeedsSuccess.bind(this),
    // On failure callback
    onFailure : onGetFeedsFail.bind(this),
    // timeout
    timeout : 30000
});
```

15 Backend connectivitiy © 2012 IBM Corporation

The Javascript procedure is used to invoke application logic as part of the adapter. You can use your favorite Javascript framework on the client to invoke the backend adapter using the Javscript function defined as part of the adapter definition.



Simplified adapter testing

- Automatic JSON conversion of hierarchical back-end data
- Support for XSL transformations

x200-lp2:8080/core/services/invoke/http?adapter=Test&parameters=["returnedContentType":"xml","path":"rss/edition_world.rss","method":"get","transformati

Raw XML:

```
<?xml version="1.0" encoding="UTF-8"?><?xml-stylesheet type="text/xsl" media="screen" href="/~d/styles/rss2full.xsl"?>
<?xml-stylesheet type="text/css" media="screen" href="http://rss.cnn.com/~d/styles/itemcontent.css"?>
<rss xmlns:media="http://search.yahoo.com/mrss/" xmlns:feedburner="http://rssnamespace.org/feedburner/ext/1.0" version="2.0"><channel>
<title>CNN.com - WORLD</title>
<link>http://edition.cnn.com/WORLD/?eref=edition_world</link>
<description>CNN.com delivers up-to-the-minute news and information on the latest top stories, weather, entertainment, politics and more.</description>
<language>en-us</language>
<copyright>© 2010 Cable News Network LP, LLLP.</copyright>
<pubDate>Mon, 29 Nov 2010 14:36:41 EST</pubDate>
<ttl>10</ttl>
<image>
<title>CNN.com - WORLD</title>
<link>http://edition.cnn.com/WORLD/?eref=edition_world</link>
<url>http://i2.cdn.turner.com/cnn/.element/img/1.0/logo/cnn.logo.rss.gif</url>
<width>144</width>
<height>33</height>
<description>CNN.com delivers up-to-the-minute news and information on the latest top stories, weather, entertainment, politics and more.</description>
</image>
<atom10:link xmlns:atom10="http://www.w3.org/2005/Atom" rel="self" type="application/rss+xml" href="http://rss.cnn.com/rss/edition_world"/>
<feedburner:info uri="rss/edition_world"/><atom10:link xmlns:atom10="http://www.w3.org/2005/Atom" rel="hub" href="http://pubsubhubbub.appspot.com/"/>
```

XSL (filtered.xsl):

```
<xsl:stylesheet version="1.0"
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
```

Edit Configuration

Edit configuration and launch.

Name: Invoke Back-end Service

Invoke Back-en Service Data | WorkLight Server configuration

Project name : Proj2

Adapter name : MyAdapter

Requires authentication   Connect as : server   Timeout : 30 seconds

WL.Server.invokeHttp(options)
Input a value for 'options':

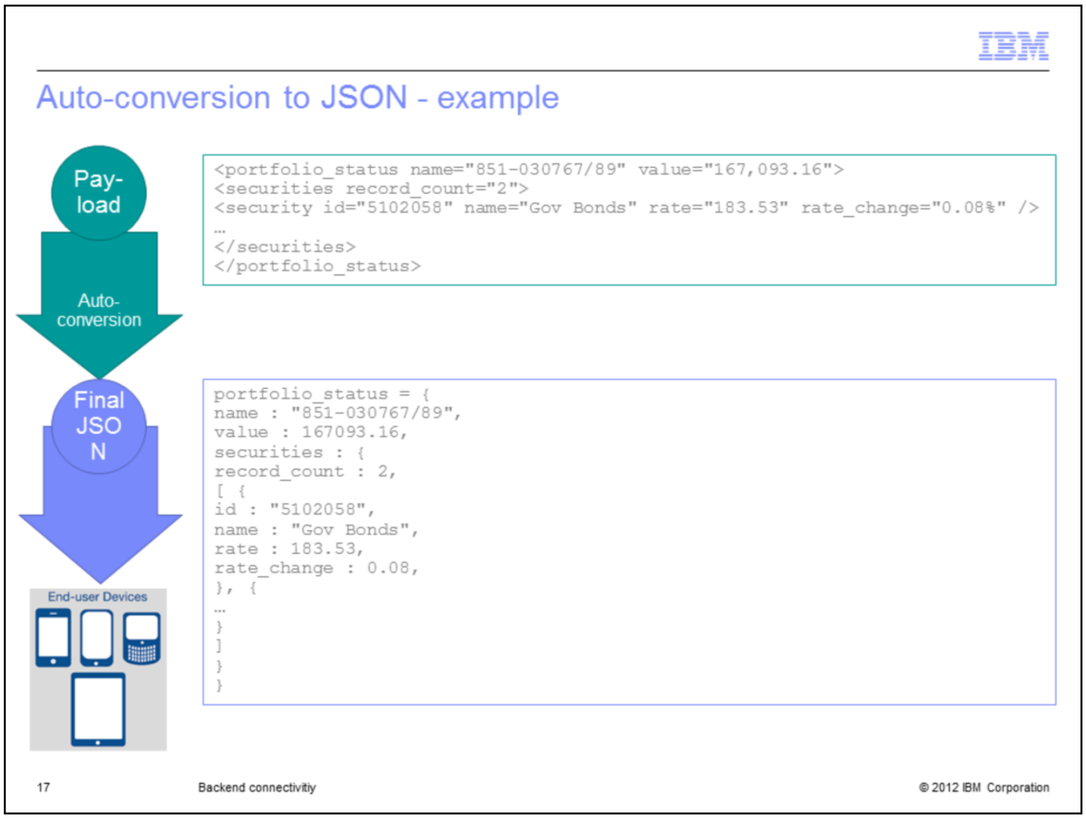
```
{
    // Mandatory
    method : 'get' or 'post',
    returnedContentType: 'xml' or 'html',
    path: value,

    // Optional
    returnedContentEncoding : 'encoding',
    parameters: {name1: value1, ... },
    headers: {name1: value1, ... },
```

ERROR: 'options' is not a valid JSON object

16   Backend connectivity   © 2012 IBM Corporation

The adapter developed in Worklight Studio (the developer tools for Worklight, an eclipse plug-in) can be used to test the request and response to the backend from within Worklight Studio. Both the transformed and plain JSON payloads are supported in the unit testing of the adapter.



Auto-conversion to JSON - example

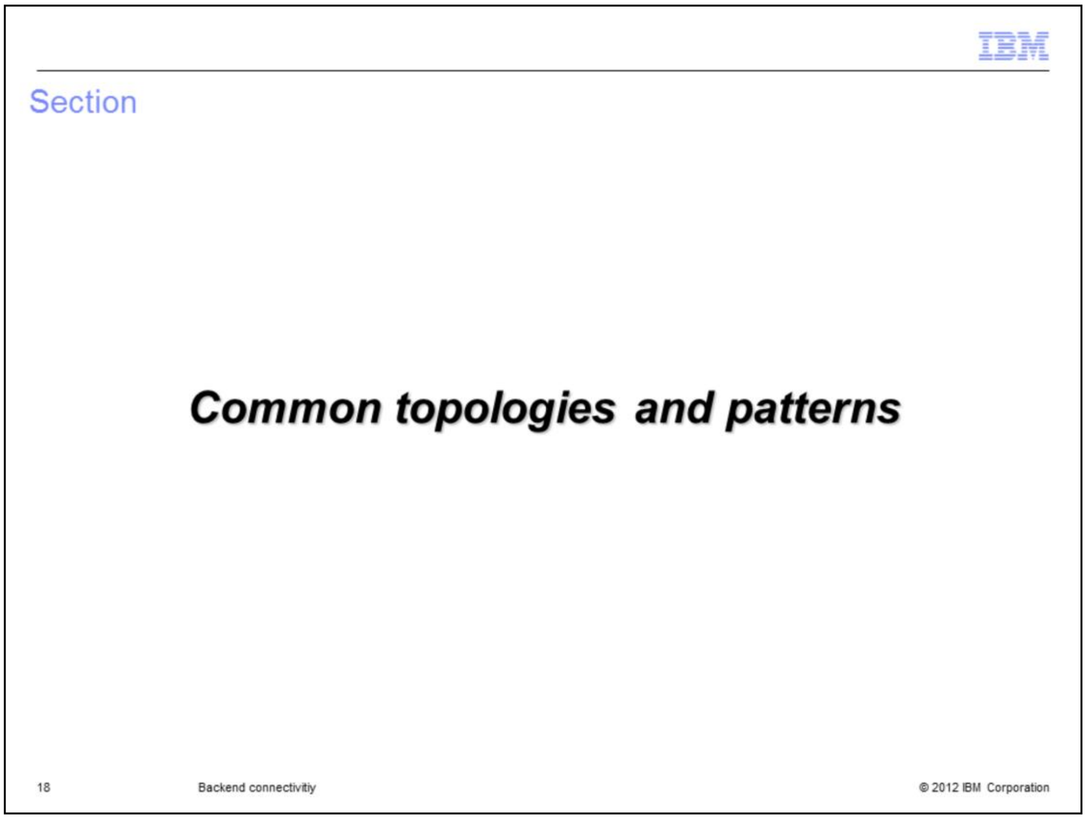
Pay-load
Auto-conversion
Final JSON
End-user Devices

```
<portfolio_status name="851-030767/89" value="167,093.16">
<securities record_count="2">
<security id="5102058" name="Gov Bonds" rate="183.53" rate_change="0.08%" />
…
</securities>
</portfolio_status>
```

```
portfolio_status = {
name : "851-030767/89",
value : 167093.16,
securities : {
record_count : 2,
[ {
id : "5102058",
name : "Gov Bonds",
rate : 183.53,
rate_change : 0.08,
}, {
…
}
]
}
}
```

17 Backend connectivity © 2012 IBM Corporation

This slide shows an example of auto-conversion of JSON from XML.

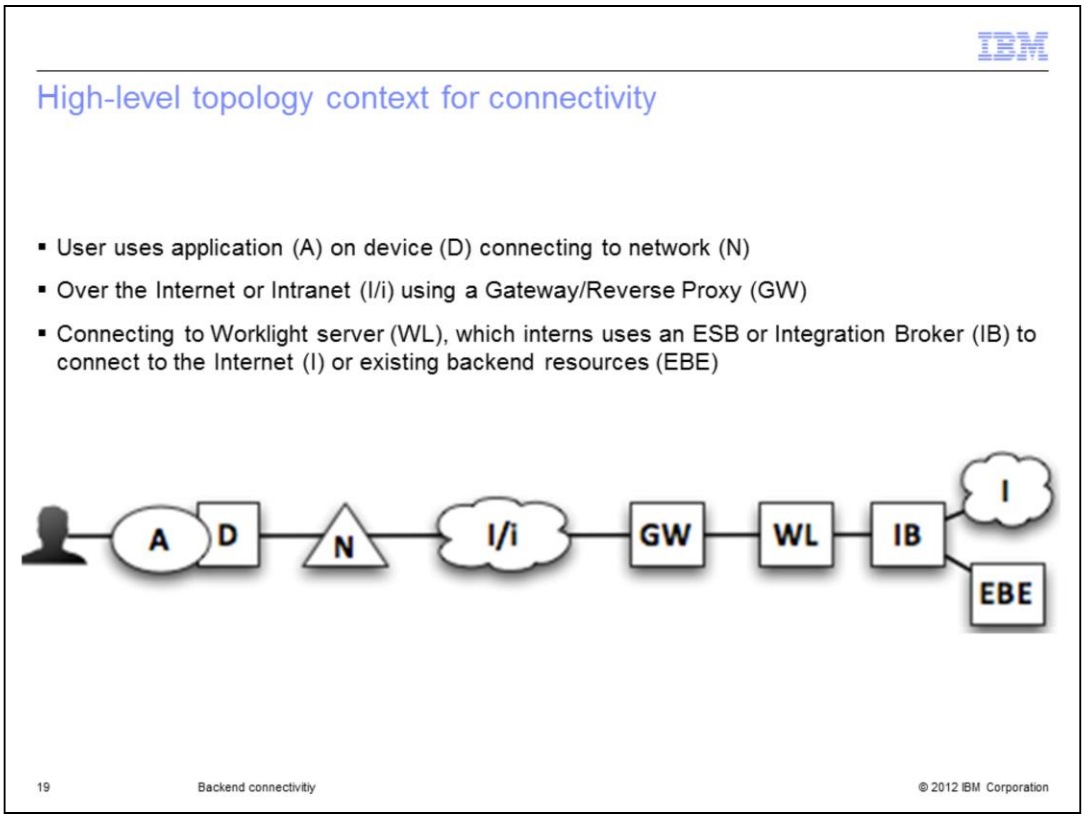


Section

Common topologies and patterns

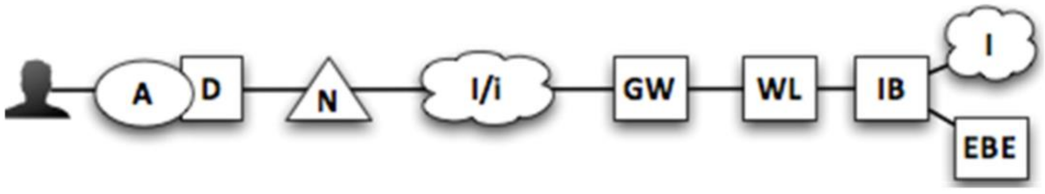
18 Backend connectivitiy © 2012 IBM Corporation

Finally, common topologies and patterns that follow is discussed.



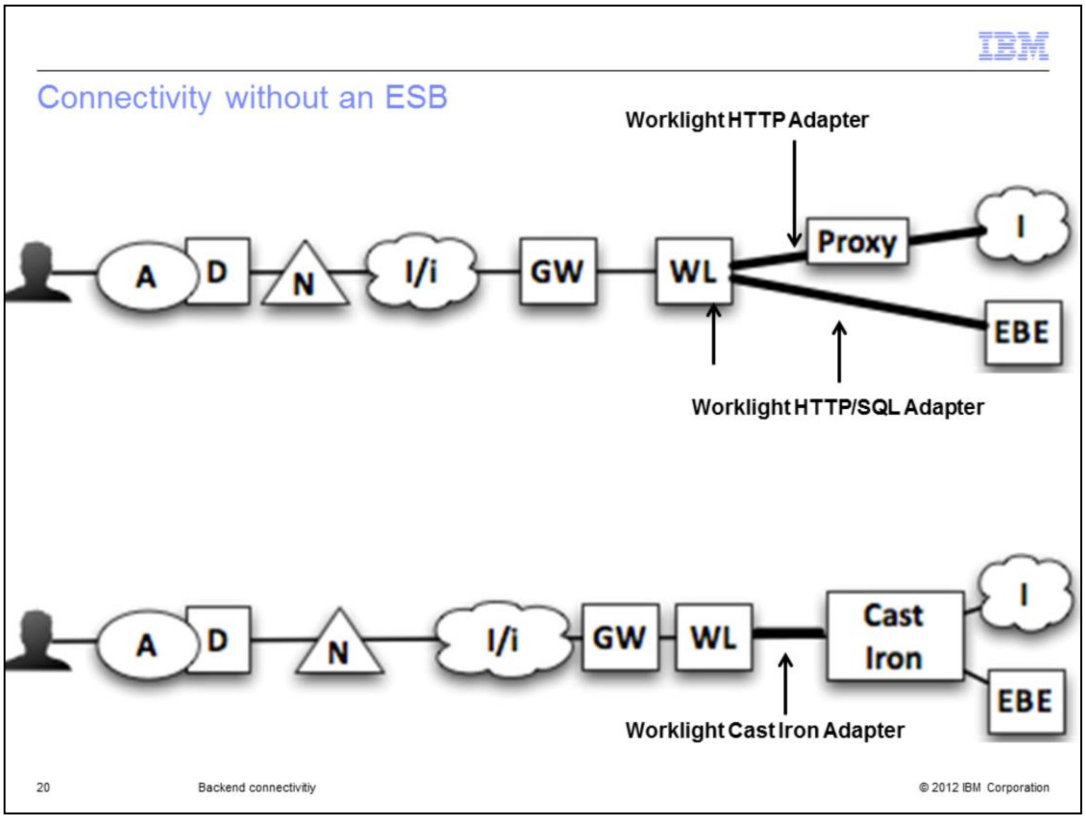
High-level topology context for connectivity

- User uses application (A) on device (D) connecting to network (N)
- Over the Internet or Intranet (I/i) using a Gateway/Reverse Proxy (GW)
- Connecting to Worklight server (WL), which interns uses an ESB or Integration Broker (IB) to connect to the Internet (I) or existing backend resources (EBE)

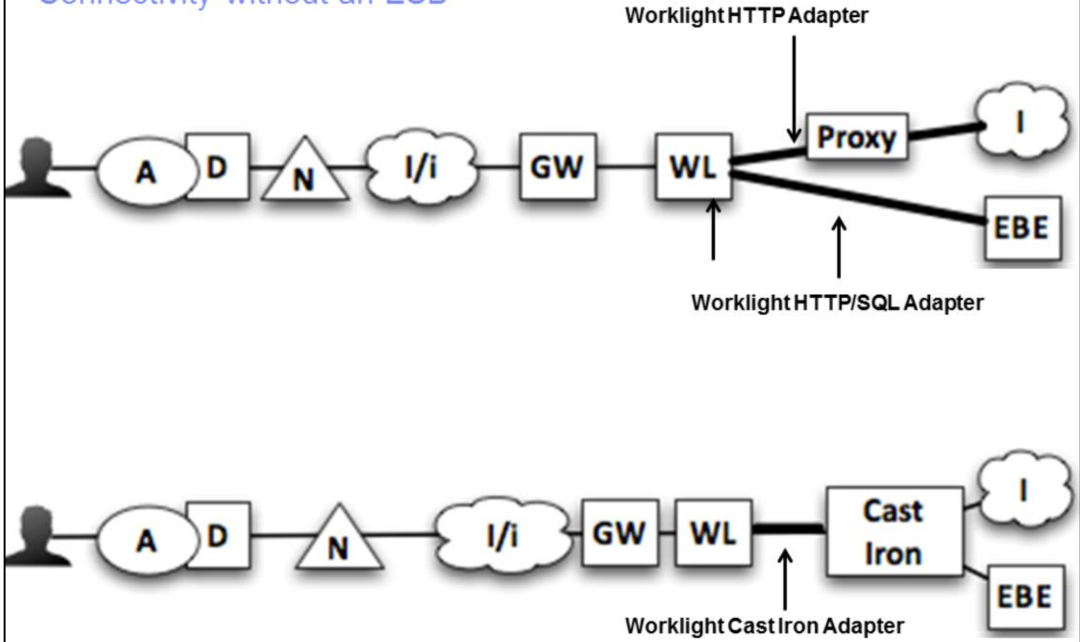


19 Backend connectivitiy © 2012 IBM Corporation

This is the end to end topology view starting with User uses application (A) on device D connecting to network (N). Over the Internet or Intranet (I/i) using a Gateway/Reverse Proxy (GW), connecting to Worklight server (WL), which in turn uses an ESB or Integration Broker (IB) to connect to the Internet (I) or existing backend resources (EBE). Variations on this pattern for specific use cases is discussed.

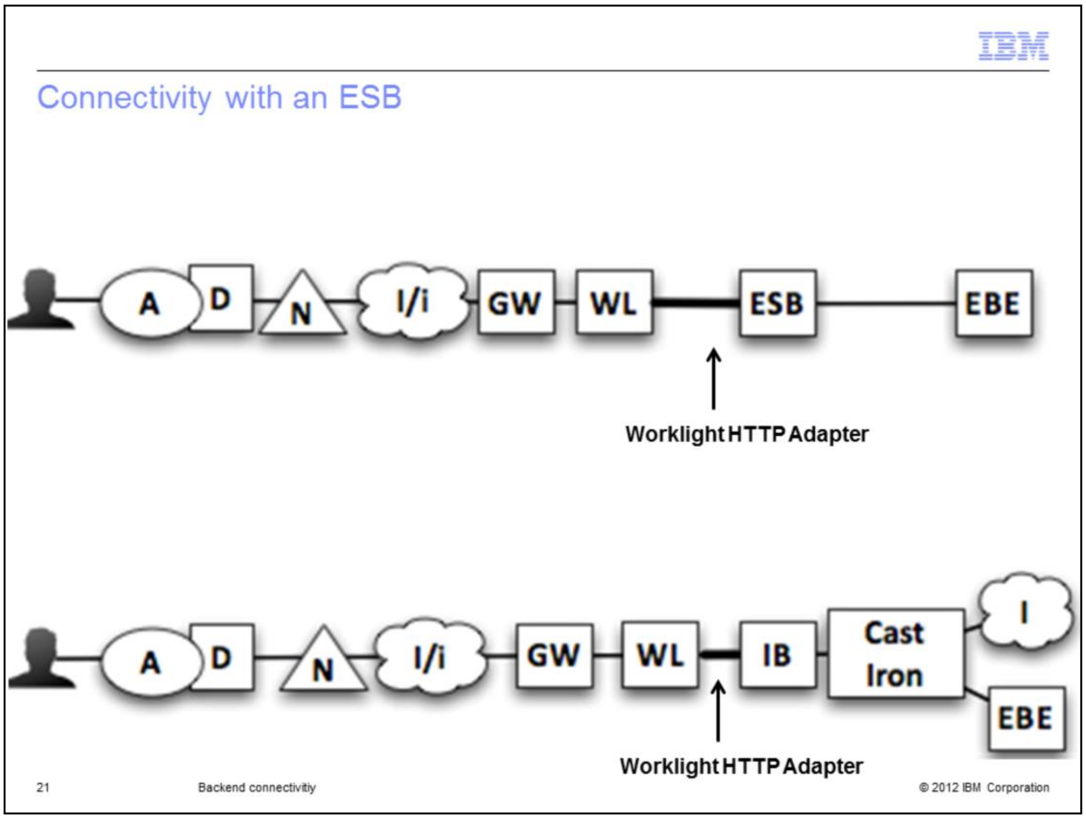


Connectivity without an ESB

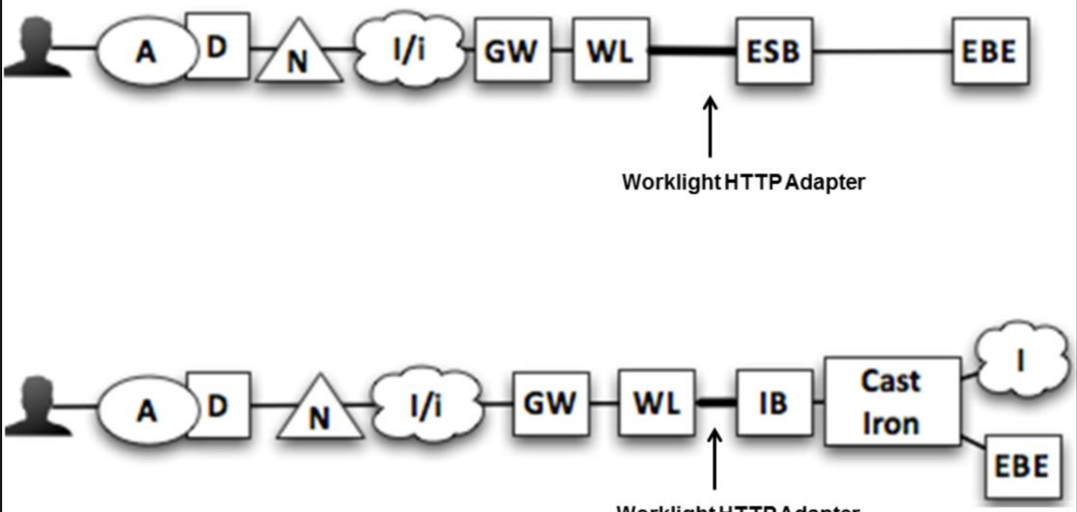


20 Backend connectivitiy © 2012 IBM Corporation

If your enterprise does not have an ESB or the application requirements support going directly to backend or cloud services or if you have access to a Cast Iron install, then Worklight can connect using HTTP adapter or the Cast Iron adapter as indicated in the topology diagrams. The top diagram illustrates Worklight application using the adapter technology to access the backend and an Internet service. Typically using a Proxy to connect to Internet services. The bottom diagram illustrates how using Cast Iron adapter and Cast Iron simplifies the connectivity to both backend resources and Internet services.

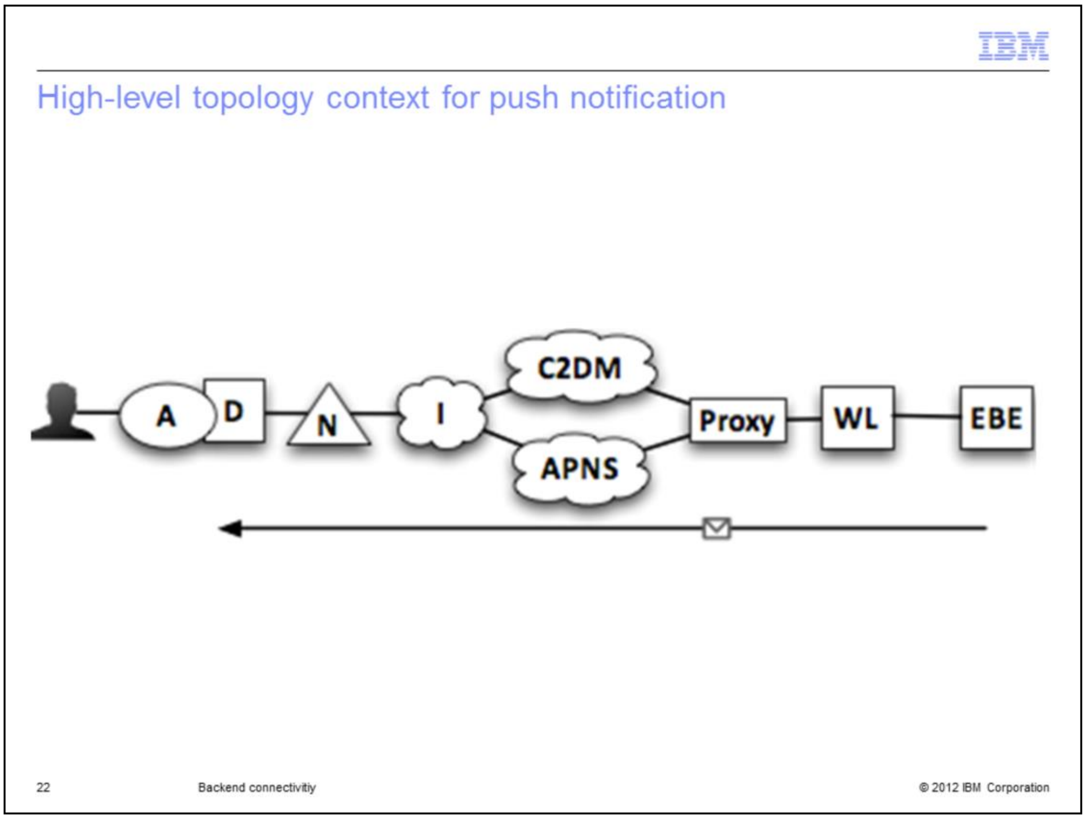


Connectivity with an ESB

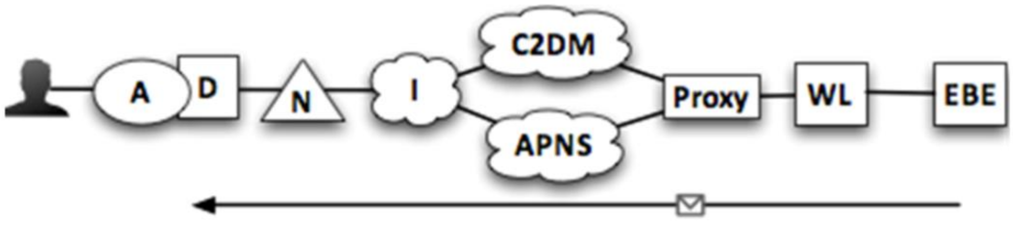


21 Backend connectivitiy © 2012 IBM Corporation

If your enterprise does have an ESB, then Worklight can connect using HTTP adapter to the ESB, where the backend resources are mediated by the ESB. In instances where you have Cast Iron, the backend connectivity is addressed by Cast Iron, but Worklight still connects to the ESB using HTTP adapter. Example of ESBs in the enterprise include Datapower Appliance, WebSphere Message Broker, and WebSpher ESB.

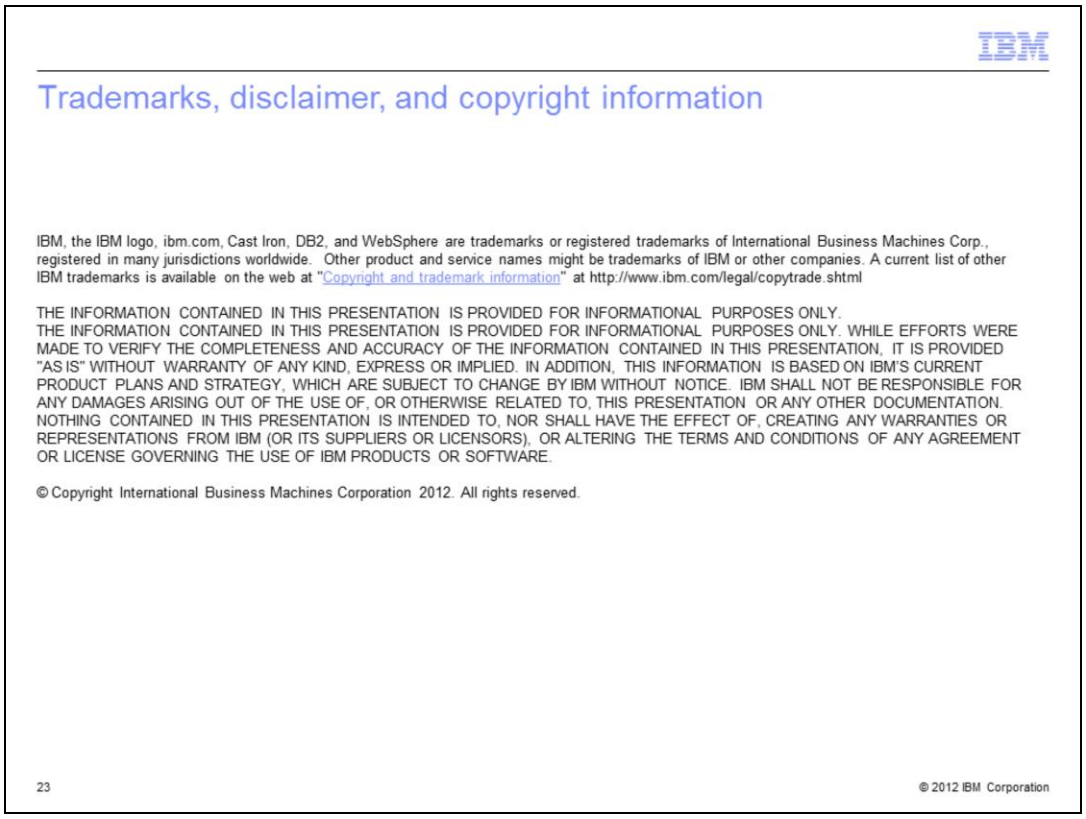


High-level topology context for push notification



22 Backend connectivitiy © 2012 IBM Corporation

The last case is for backend triggering notification to devices. An enterprise backend system (EBE) triggers an event that Worklight server (WL) picks up and sends push notification to Apple (APNS) and Android (C2DM) devices by connecting to the respective services using a Proxy. The user is notified on the device (D) for the given application (A).



Trademarks, disclaimer, and copyright information