Fraser Bohm – STSM CICS TS Product Architect –
fbohm @ uk.ibm.com

CICS just keeps on delivering, CICS Feature Packs to enhance your CICS Environment

Please Note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Agenda

- Feature Packs – What are they?
- CICS Mobile Extensions
- CICS Modern Batch
- CICS Security Extensions
- Summary and Questions

CICS Feature Packs – What are they?

No additional charge Features

Order on Shop Z

Available for current releases (4.2 and 5.1)

Fully Supported via PMR and APAR process

CICS Mobile Extensions

CICS Modern Batch

CICS Security Token Extensions

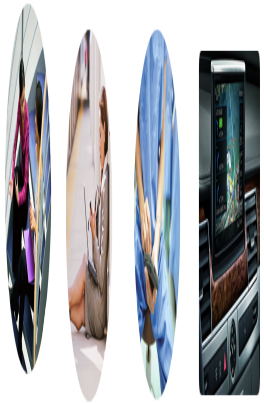
CICS Mobile Extensions

Mobile is a mandatory transformation

10 Billion devices by 2020

61% of CIOs put mobile as priority

45% increased productivity with mobile apps



Bridging the gap between mobile and CICS

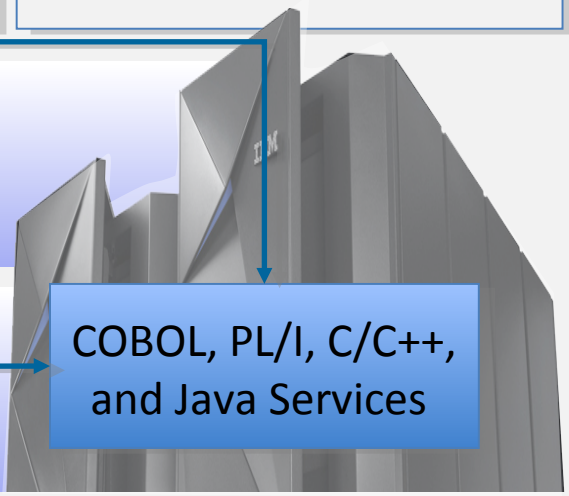
Mobile extensions – simplified integration with mobile devices

CICS Transaction Server Feature Pack for Mobile Extensions

Agile integration	*Service enablement*	*Scalable platform*
JSON data and RESTful interface makes integration with mobile devices simple and efficient	*COBOL, C/C++, PL/I and Java programs can be RESTful service providers*	*CICS TS V5.1's advanced scalability makes it the ideal platform for managing mobile workloads*

Via Worklight Server

Worklight® Server



Direct to CICS

COBOL, PL/I, C/C++, and Java Services

What is REST?

The basics of RESTful services

- REST is an architectural style, unlike SOAP which is a protocol
- Typically use JSON to describe the data
- Uses the HTTP verbs (below) to indicate the nature of an operation
- Uses the URI to determine the resource(s) to operate upon

Create	Read	Update	Delete
HTTP **POST**	HTTP **GET**	HTTP **PUT**	HTTP **DELETE**

Example URIs

A collection

```
http://my.cics.example/resources/

http://my.cics.example/resources/specific_resource
```


REST Simple Sample – via HTTP

- Request

```
GET /mortgage/231677 HTTP/1.1
  Host: www.example.com
  Accept-Language: en
  Charset: UTF-8
```

- Response

or

```
HTTP/1.1 200 OK
  Language: en_us
  Charset: UTF-8
  Content-Type: application/json
{“principal”:”238000”,”rate”:”3.5”, “type”:”5/1 ARM”}
```

```
HTTP/1.1 200 OK
  Language: en_us
  Charset: UTF-8
  Content-Type: text/xml
<mortgage><principal>238000</principal><rate>3.5</rate><type>5/1 ARM</type></mortgage>
```

JSON is the mobile platform of choice?

JSON is the mobile format of choice

- The growth in mobile helped boost the popularity of JSON
- The lightweight data format is ideally suited to mobile data transfer
- As a result numerous tools and frameworks now support JSON…

Worklight

jQuery mobile dojo

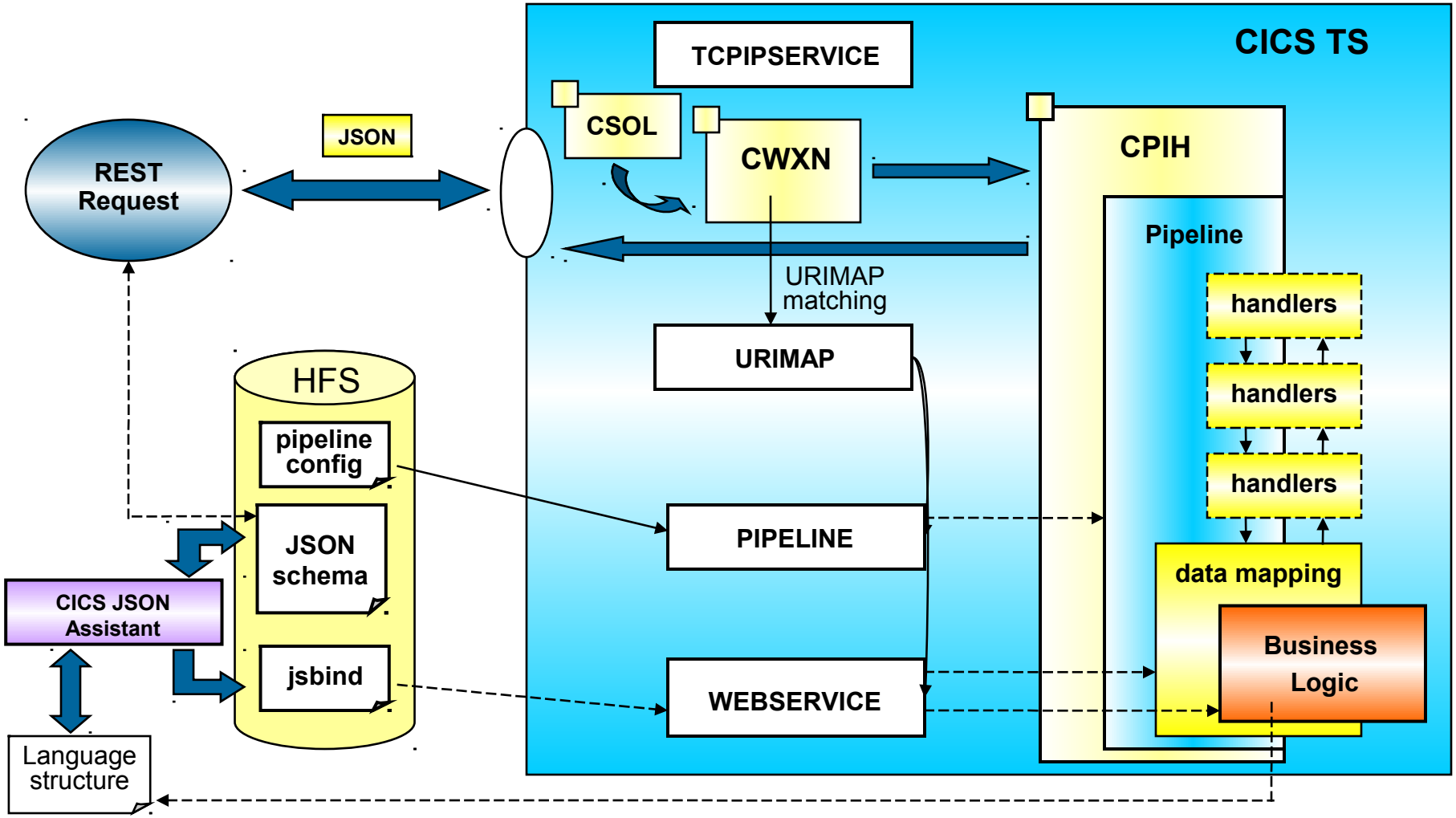
IBM Worklight uses JSON:
- For communication between a mobile application and the Worklight Server
- Provides a JSON Store for offline storage of data
- Automatically converts Webservice SOAP replies into JSON

Numerous other frameworks depend upon JSON data…

Eg. jQuery & dojo use JSON:
- JSON Store
- Ajax calls
- All data interchange

Producing JSON from CICS: Scenario 1…

- Overview – top down

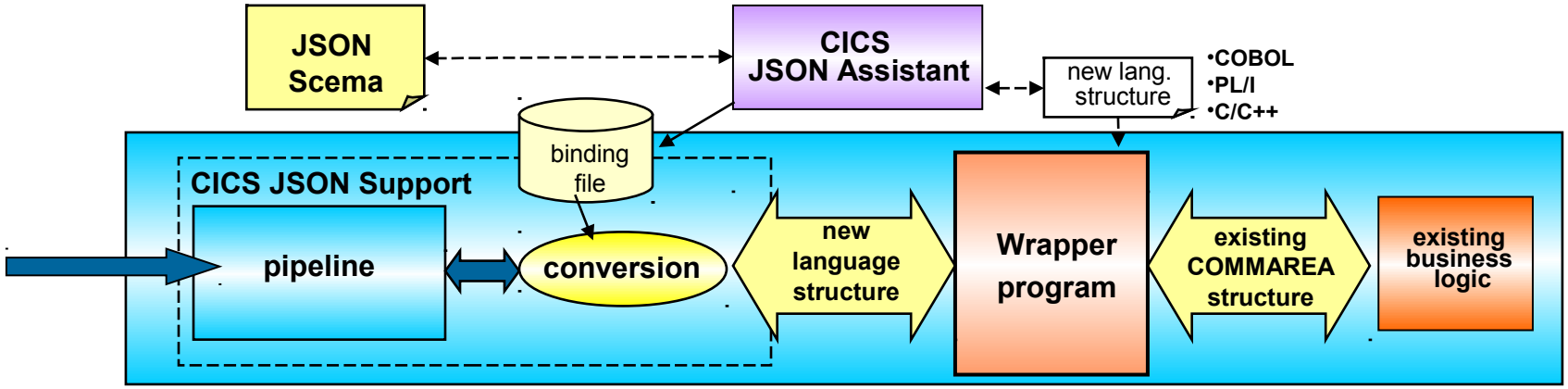


Producing JSON from CICS: Scenario 1…

- Expose an existing COMMAREA or channel LINKable program
  - Define an AXIS2 JVMServer to do the COMMAREA or channel conversion (one time)
  - Run the DFHLS2JS utility
  - Define a PIPELINE, URIMAP, and WEBSERVICE to CICS
  - Test
  - Advantage:
    - Only had to run the utility to expose CICS program, plus resource definitions (15 minutes)
  - Disadvantage
    - Likely does not fit the REST paradigm where a GET reads, and POST adds, etc

Producing JSON from CICS: Meet-in-Middle, Scenario 2…

- Write a wrapper program that understands paths and methods, and returns HTTP status codes
- It invokes the existing CICS program
- Wrapper program looks at the HTTP “method” to determine the action to take, the path to see what to take the action on, and returns an appropriate status code



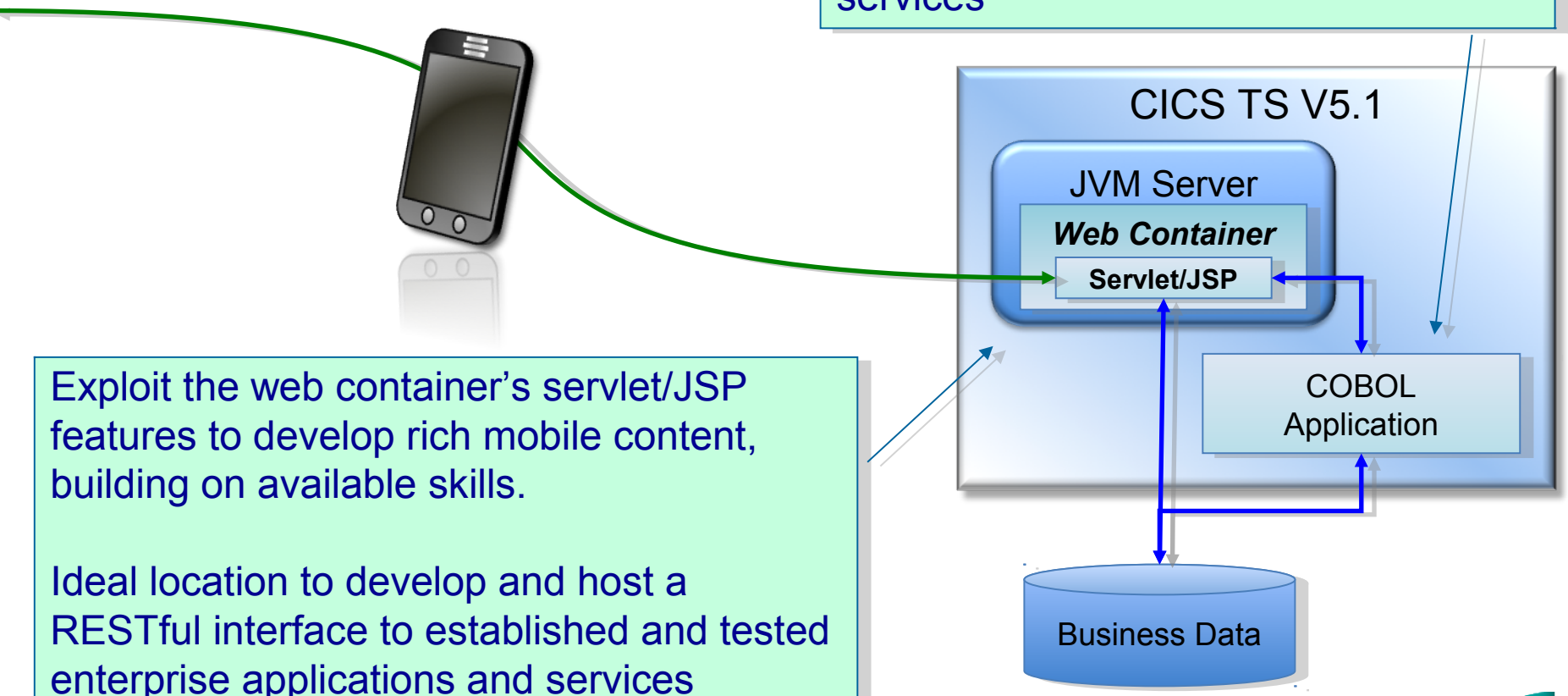
Producing JSON from CICS: Meet-in-Middle, Scenario 2…

- The wrapper program:
  - Can look at the ‘path’ in the DFHWS-URI container
  - Can test the ‘method’ in the DFHHTTPMETHOD container
  - Can prepare a COMMAREA or channel and containers to invoke the target business logic program
  - Can return selected fields (just what the invoking program wants)
  - Can return a 404 Not Found if the data is not found, etc

Doing it all Java style – Liberty Profile

RESTful services can now be hosted within the CICS Web Container, with new support for the JAX-RS API

Link to existing C/C++, COBOL, PL/I, and Java applications to exploit existing enterprise applications and services



Exploit the web container's servlet/JSP features to develop rich mobile content, building on available skills.

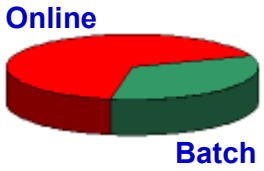
Ideal location to develop and host a RESTful interface to established and tested enterprise applications and services

CICS Modern Batch

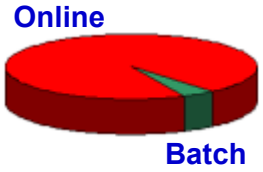
Concept of "Dedicated Batch" Window Going Away

Windows of time which used to be dedicated to batch processing are shrinking.
The demands of online processing require more and more ...

In the past ...



Today ...



24 x 7 x 365 Access

Users of your online systems expect availability at all hours

Users from other parts of the world means availability is expected around the clock

Mobile Users

Users are no longer tied to a desk and a computer. Today users have access to mobile computing devices that are with the user wherever they may be. Day or night, home or office.

The need to process batch work has *not* gone away.

The need to perform the work concurrent with OLTP has emerged.

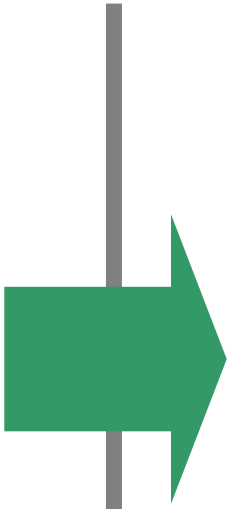
The Value of Shared Services

It's not *just* that the window is shrinking ... it's also the cost pressures on maintaining the batch and OLTP environments:

Batch Support Staff

OLTP Support Staff

- Batch Infrastructure
- OLTP Infrastructure
- Batch Applications
- OLTP Applications
- Batch Development Tools
- OLTP Development Tools
- Homegrown Middleware Infrastructure



Common Support Staff

- Batch + OLTP Common Infrastructure
- Batch Applications
- Shared Java Assets
- OLTP Applications
- Batch + OLTP Common Tooling
- Homegrown Middleware Infrastructure

Efficiencies through consolidation around common assets

Java for Batch Processing?

Yes ... for many very good reasons:

Availability of Skills

Java is a programming language with wide adoption in the industry. Skills for Java programming are common and affordable.

Tooling Support

Development tooling for Java has advanced to the point where some tools (IBM Rational Application Developer) are very powerful and sophisticated.

This also provides an opportunity to consolidate to a common tooling environment for both OLTP and batch development.

z/OS Specialty Engines

Pressures on cost containment often dictate greater use of z/OS specialty engines. Java offloads to zAAP. Java batch does as well.

Processing in OLTP Runtime

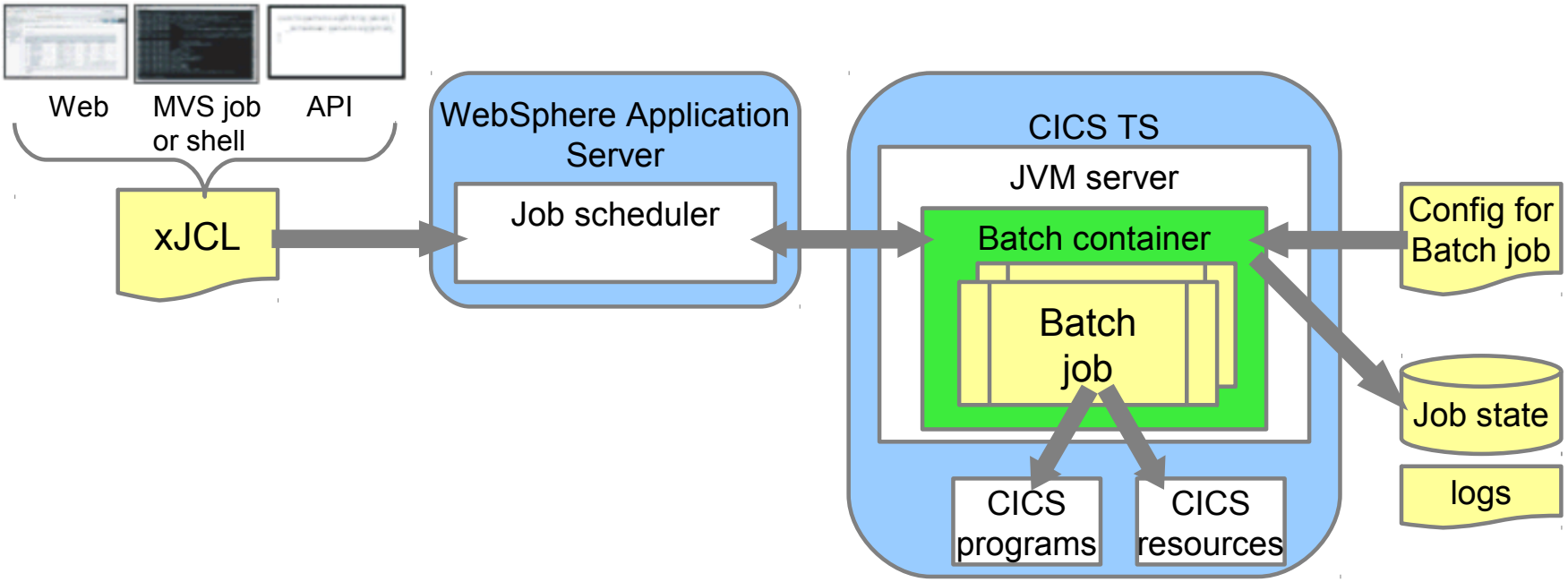
Running Java batch in the same execution runtime as Java OLTP provides an opportunity to mix and manage the two processing types together under the same management model.

High-level overview of the CICS Feature Pack

- What it is
  - A Java Batch Container for CICS 4.2 / 5.1 providing
    - Checkpointing, logging, recovery etc
  - Jobs scheduled and managed from WebSphere Application Server

- Delivered as a fully supported Feature Pack
  - See Announcement ENUS213-177
    - Order via Shop Z

Architecture



CICS Security Extensions

SAML – Security Assertion Markup Language

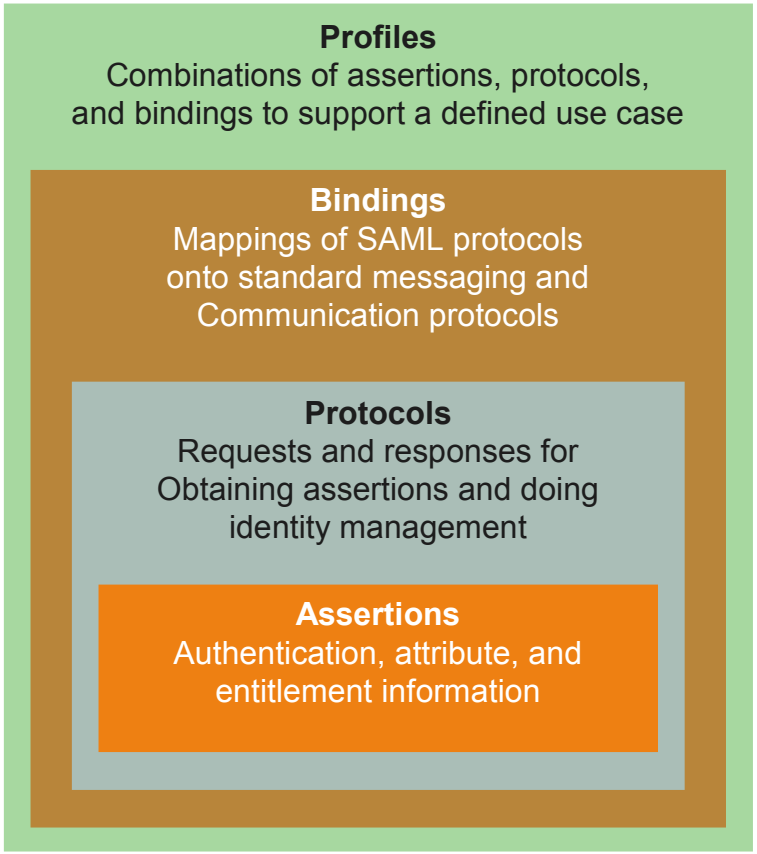
OASIS open standard

"XML based framework for describing and exchanging security information between on-line business partners."

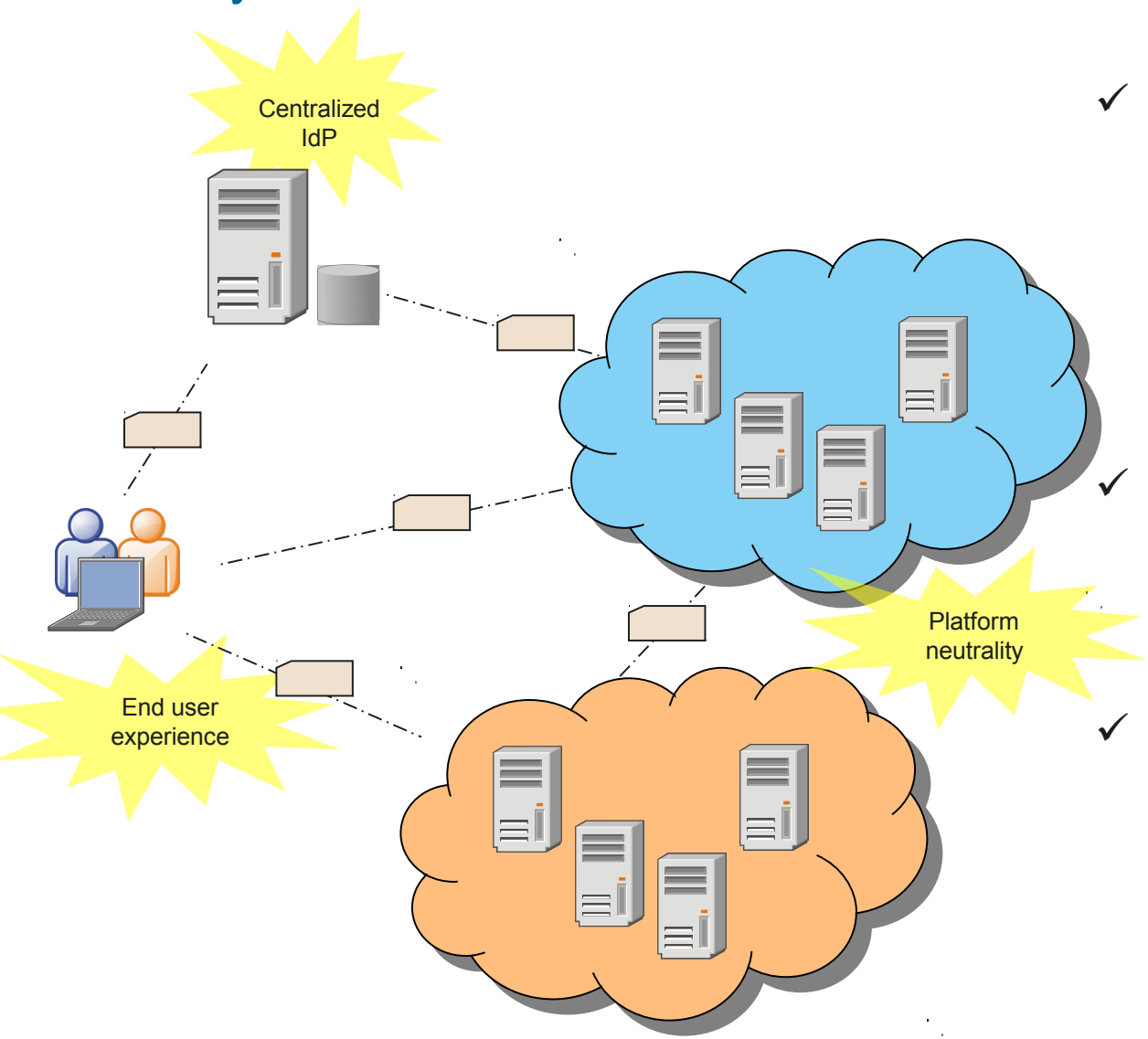
SAML dates from 2001; most recent update from 2005

Used for:

- Web SSO
- Attribute-based authorization
- Web service security

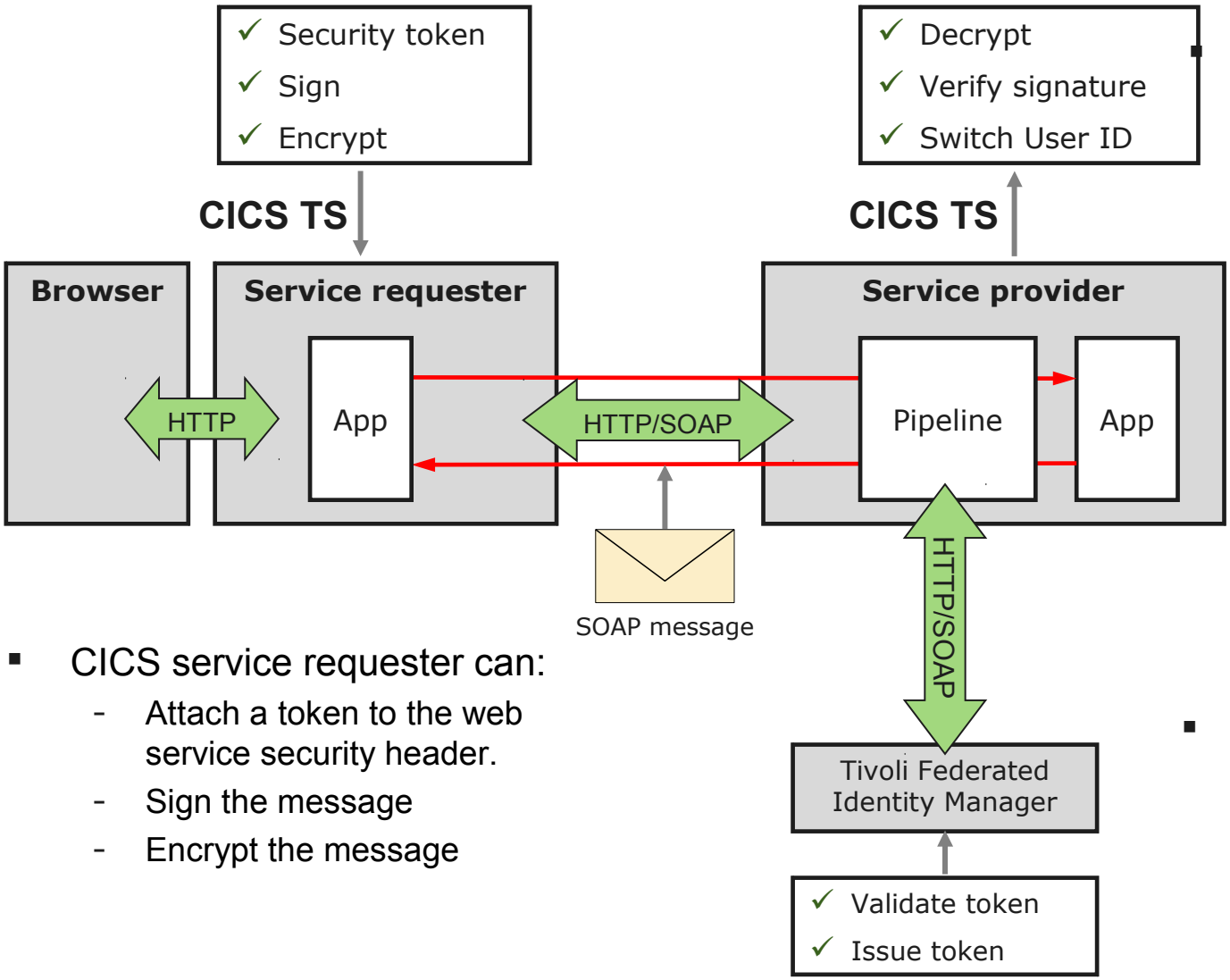


Why SAML?



- ✓ Centralized IdP
  - **Reduce service providers administration costs**
  - Loose coupling of directories
  - Responsibility of identity management can be pushed to the IdP
- ✓ Platform neutrality
  - Security framework independent from architecture and implementation
- ✓ Improved end user experience
  - No need to authenticate at each service provider

Web Service Security in CICS

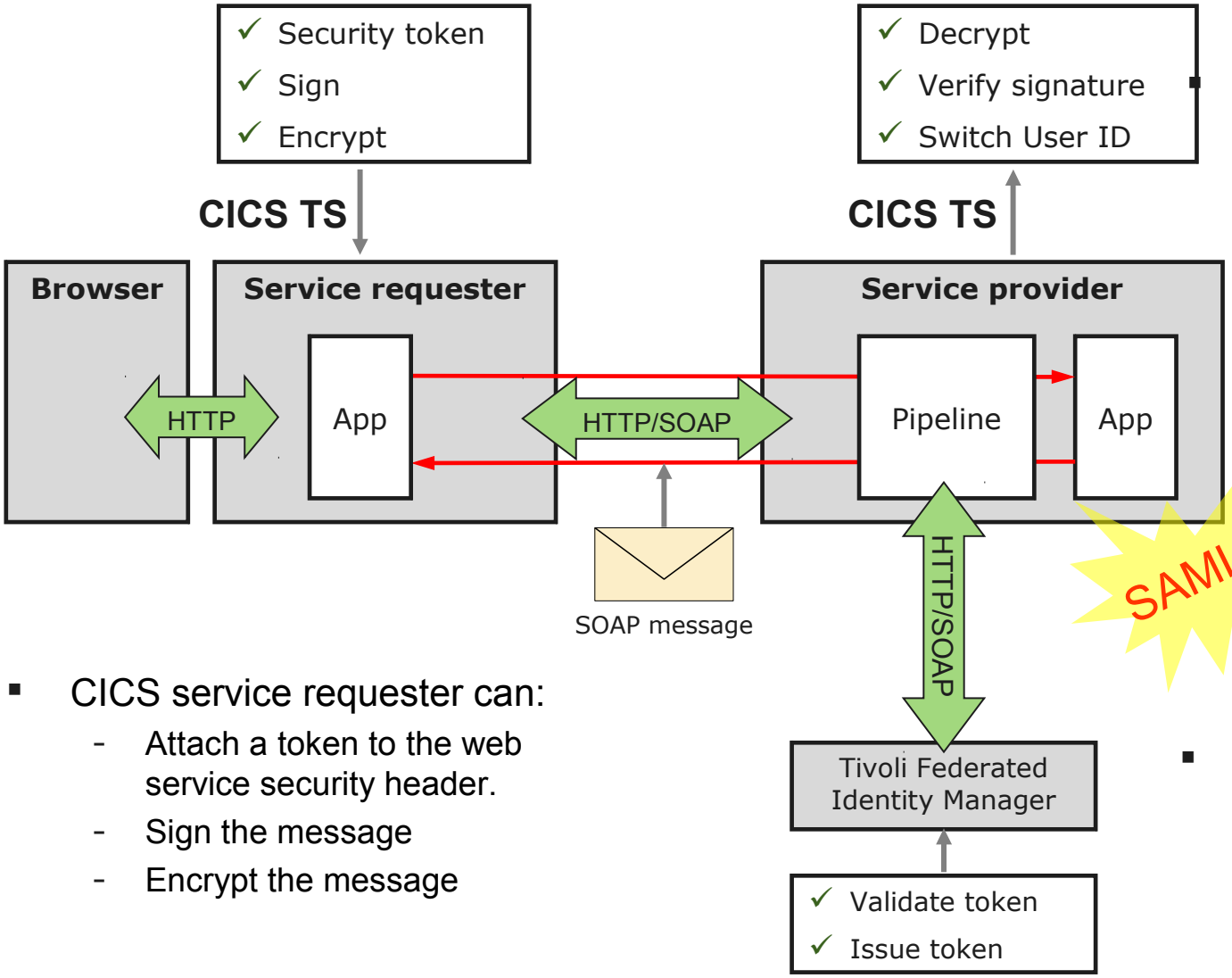


- CICS service provider can:
  - Decrypt the message
  - Verify the signature
  - Map the identity in a token to a RACF user ID and switch to it.

- CICS service requester can:
  - Attach a token to the web service security header.
  - Sign the message
  - Encrypt the message

- WS-Trust support:
  - Validate token
  - Issue token
  - Send request to an external STS

Web Service Security in CICS



- CICS service requester can:
  - Attach a token to the web service security header.
  - Sign the message
  - Encrypt the message

- CICS service provider can:
  - Decrypt the message
  - Verify the signature
  - Map the identity in a token to a RACF user ID and switch to it.

- SAML support:
  - Validate token
  - Extract SAML information

- WS-Trust support:
  - Validate token
  - Issue token
  - Send request to an external STS

CICS TS Feature Pack for Security Token Extensions V1.0

Support provided for SAML core specification V1.1 and V2.0

Enables CICS applications and provider pipelines to validate SAML assertions and extract information.

Provides:

- an API for CICS applications, consisting of:
  - A linkable interface, DFHSAML
  - A channel and a set of containers to exchange SAML information
- pipeline configuration for SAML support in inbound web services
- Copybooks DFHSAMLx
- Group DFHSAML with program and default JVM server definition
- Group DFH$SAML for the IVP Sample

Core implementation is a Java component: workload can be offloaded to zAAP

CICS TS Feature Pack for Security Token Extensions V1.0

Supported on CICS TS V5.1 and CICS TS V4.2

Prerequisites:

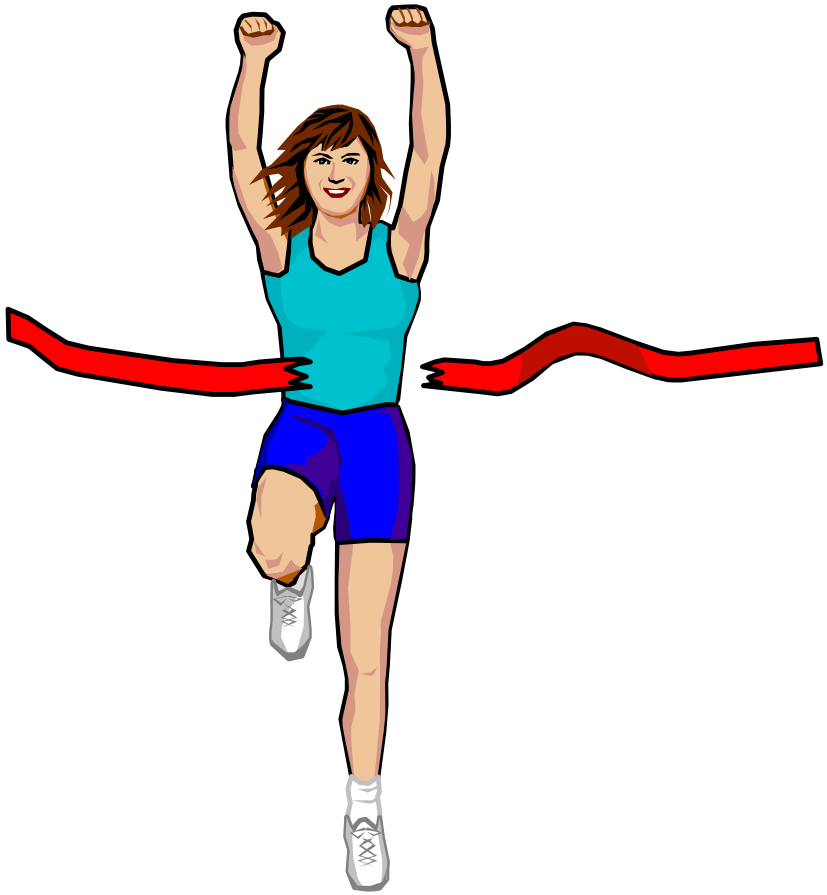
- PM82005 for CICS TS 5.1
- PM82012 for CICS TS 4.2

Available for download on Shopz

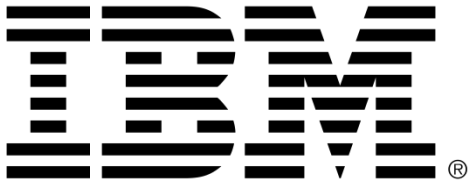
Summary

- Feature Packs are no charge, fully supported, product extension features

- Mobile Extensions
- Modern Batch
- Security Token Extensions
- Dynamic Scripting



Questions…



Legal Disclaimer