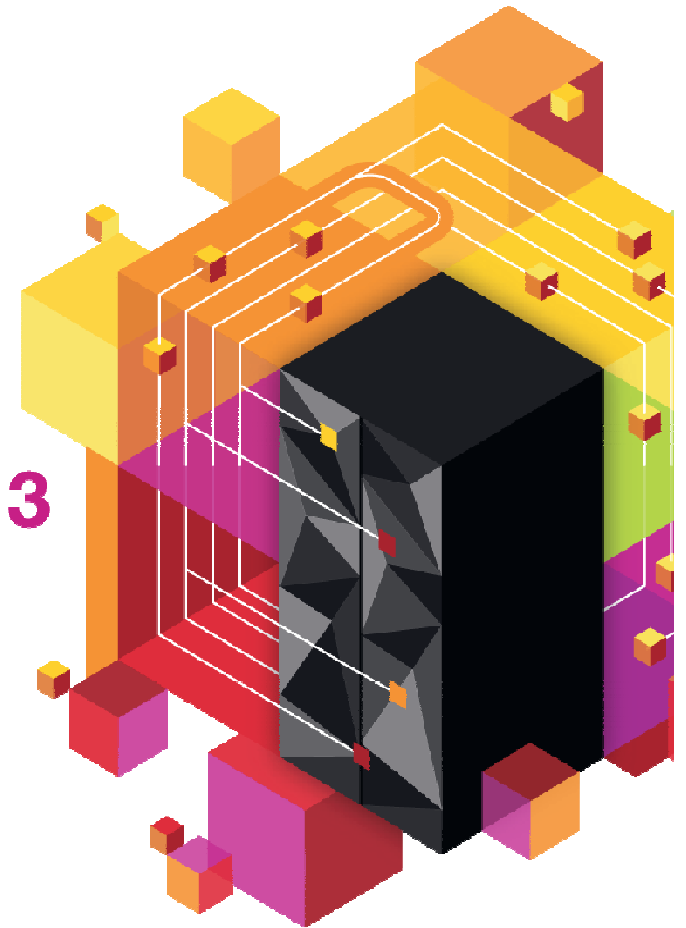




Université du Mainframe 2013

4-5 avril





Connectivité CICS, de la théorie à la pratique: en route vers le Cloud.



Régis DAVID
Expert CICS IBM
regis_david@fr.ibm.com

Université du Mainframe 2013

I ♥ CICS

4-5 avril



Infrastructure & Services

Introducing Application Interfaces



Agenda

- Interfaces
- Connectivity to the interfaces
- From programs to service operations
- The redblib showcase
- The redblib implementation
- CICS Cloud
- From operations to application entry points
- The redblib implementation

43 years of CICS evolution

Terminal Oriented

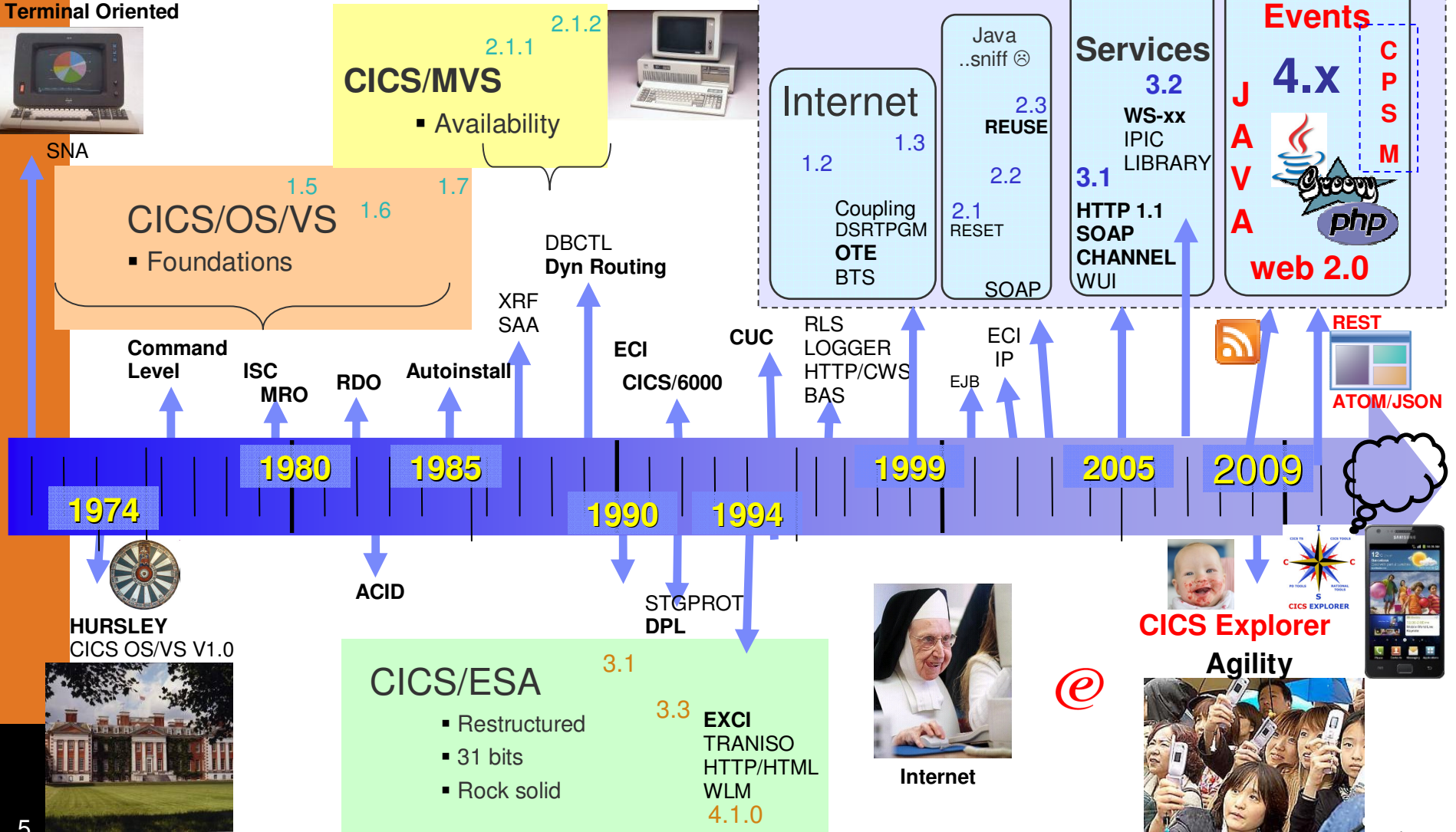


SNA

Client/Server

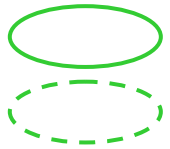
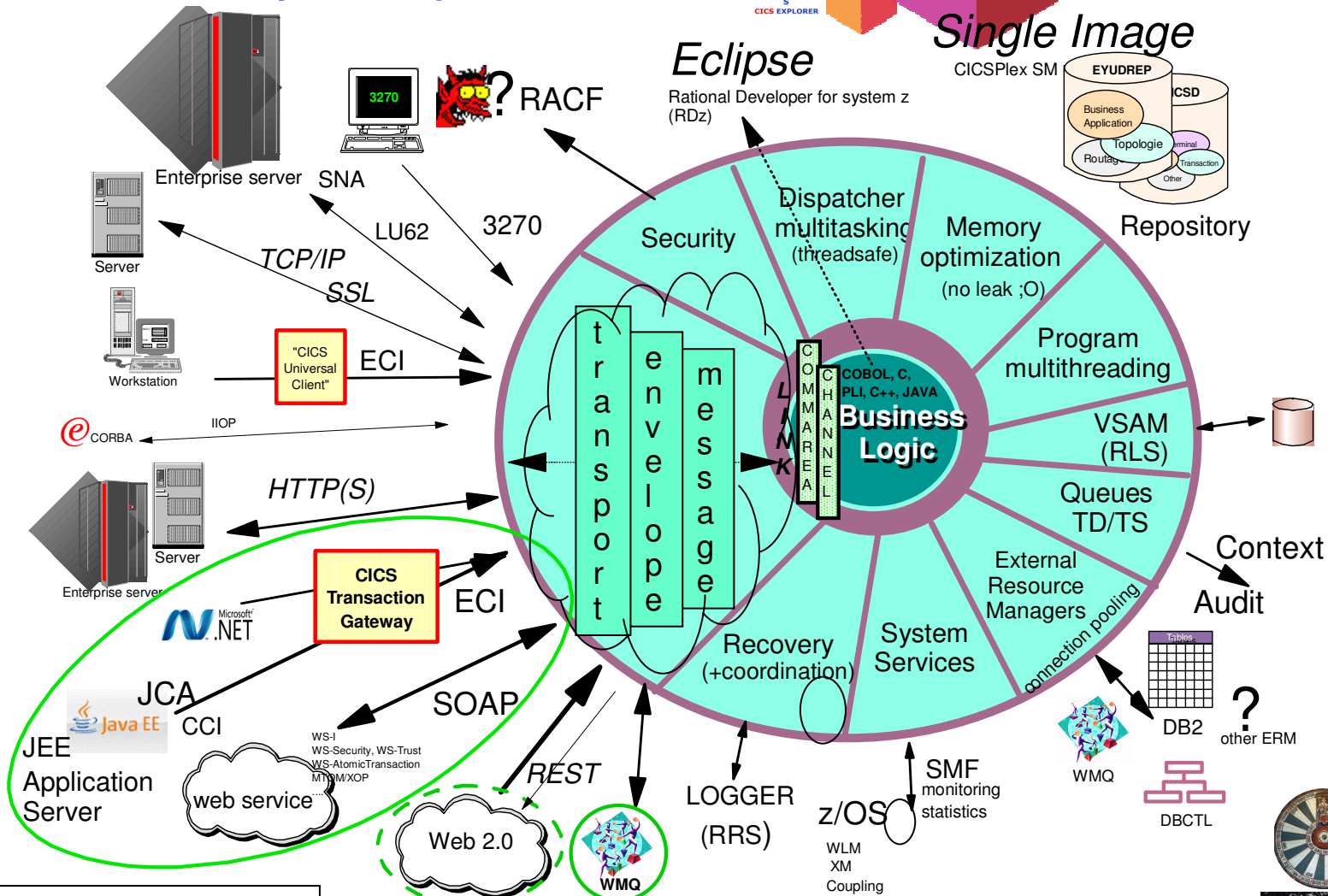


CICS TS

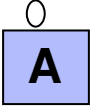




CICS is an open system...



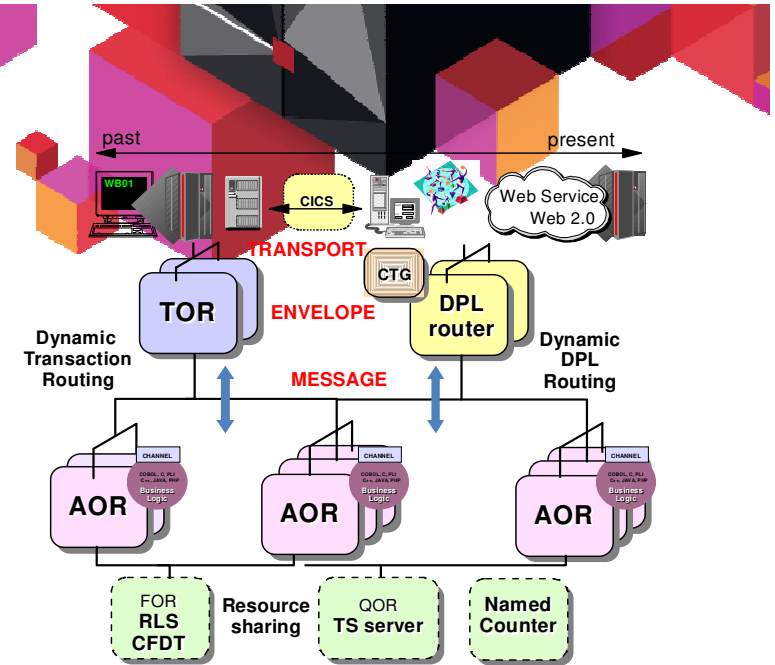
Strategic
Emerging



Infrastructure and services

■ Infrastructure

- Inbound
 - Transport, envelope, message, routing
- Outbound
 - Message, routing, envelope, transport
- Multitasking, memory, sharing, data, security
 - XML: <infra>?</infra> JSON: { "infra": "?" }
- Flexibility



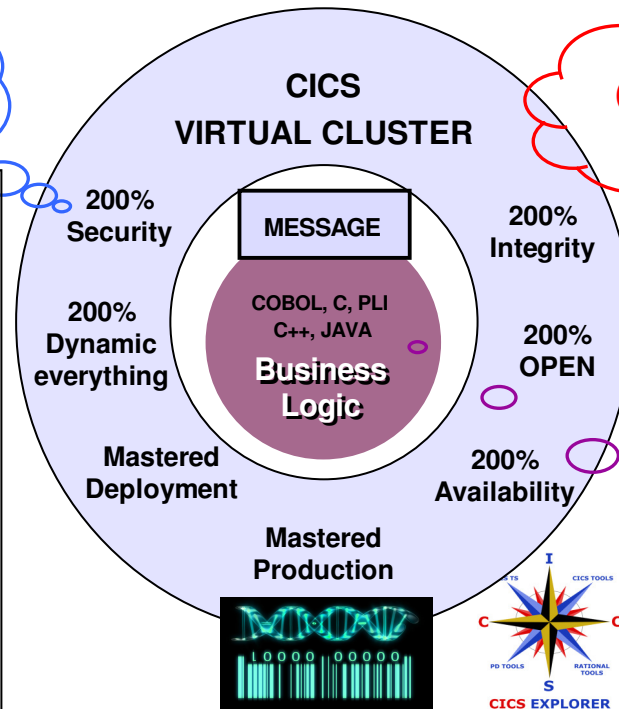
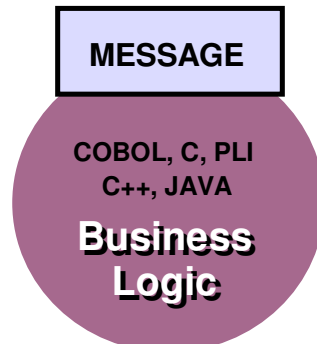
PaaS

CICS TS V5.1

SaaS

■ Business « Service »

- Business message, business logic
- Granularity
- Productivity, agility





Application Interface

- Wikipedia is my friend

- An interface is a « *point of operation between components* »
- These components « *communicate using input/output systems and transport protocol* »

- SG24-5446-06 is my friend

- The point of *operation* is a combination of:
 - An operation, function, method, etc, which identifies the point of interaction
i.e: CICS program, SOAP operation, object method

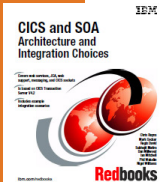
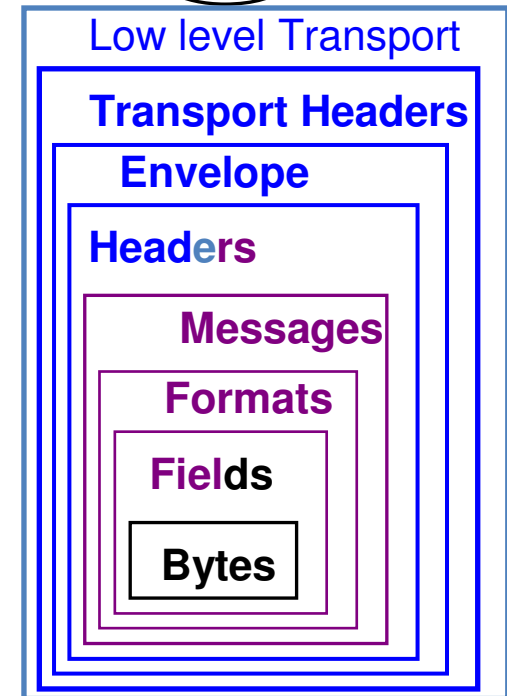
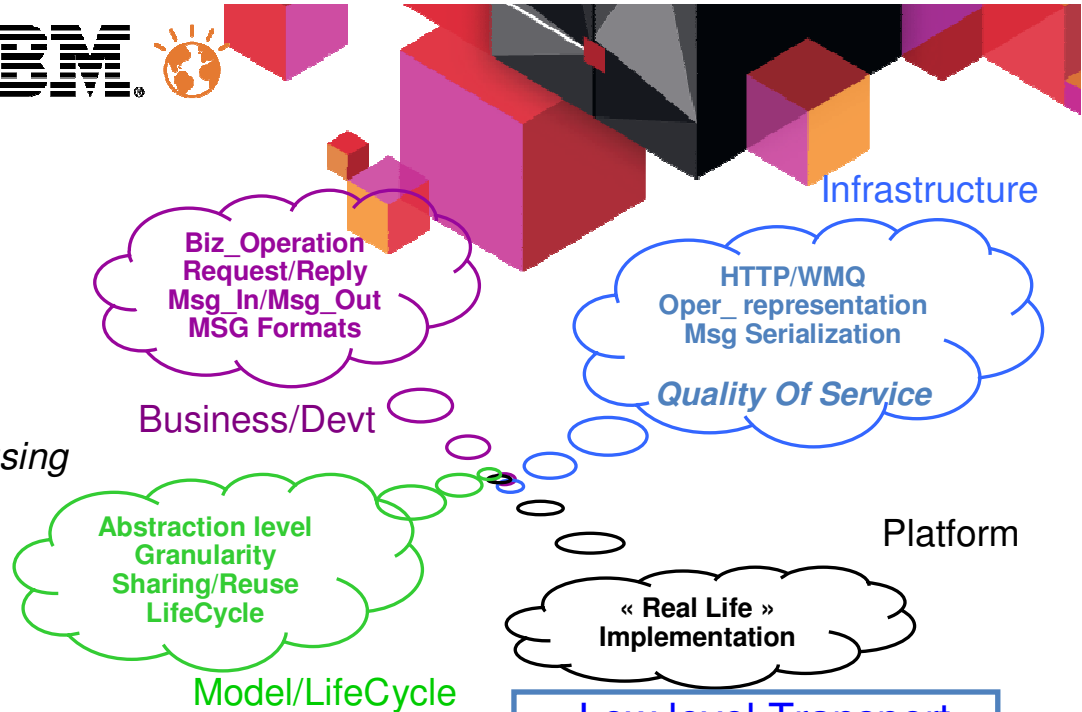
Input and/or output *messages* exchanged when using the operation
i.e: CICS COMMAREA, SOAP envelope, object IIOF

Message *formats* which define the message structure
i.e: COBOL copy, SOAP message, object method signature

Message *exchange* patterns which identify the type of interaction
i.e: synchronous request/reply, notification

A *transport* protocol
i.e: CICS IRP, http, WebSphere MQSeries

Tactical/Strategic
Share/Reuse
Granularity/Lifecycle



The interfaces can be implicitly or explicitly defined

i.e: inline code comments, WSDL



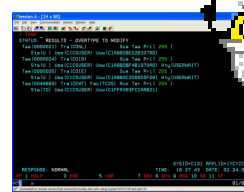
Challenges

- Transport handling
 - CICS magic !
- Operation identification
 - Real CICS program name, URI, ...
- Message extraction from its envelope
 - CICS magic ! Or from APIs
- Message deserialization/serialization
 - Client or server side ? CICS magic ! ...
- Message exchange pattern identification
 - Implicit or explicit, known or documented



*May be, but we are now OPEN and using STANDARDS
we are click and go !*

CICS DOS/VS V1.3.0



*Hum hum,
we do that for years !*

*Good news, will be easy,
CICS is OPEN to STANDARDS*

- Transport handling :
 - SNA
- Operation identification
 - First 4 characters of the stream
 - 3270 AID
- Message extraction from its envelope:
 - CICS terminal control
- Message deserialization/serialization
 - CICS BMS APIs or 3270 datstream
- Message exchange pattern identification
 - Request/Reply or 3270 AID

CICS implementation **transparency** to the world, standards **transparency** to the CICS world

- **Adapters**

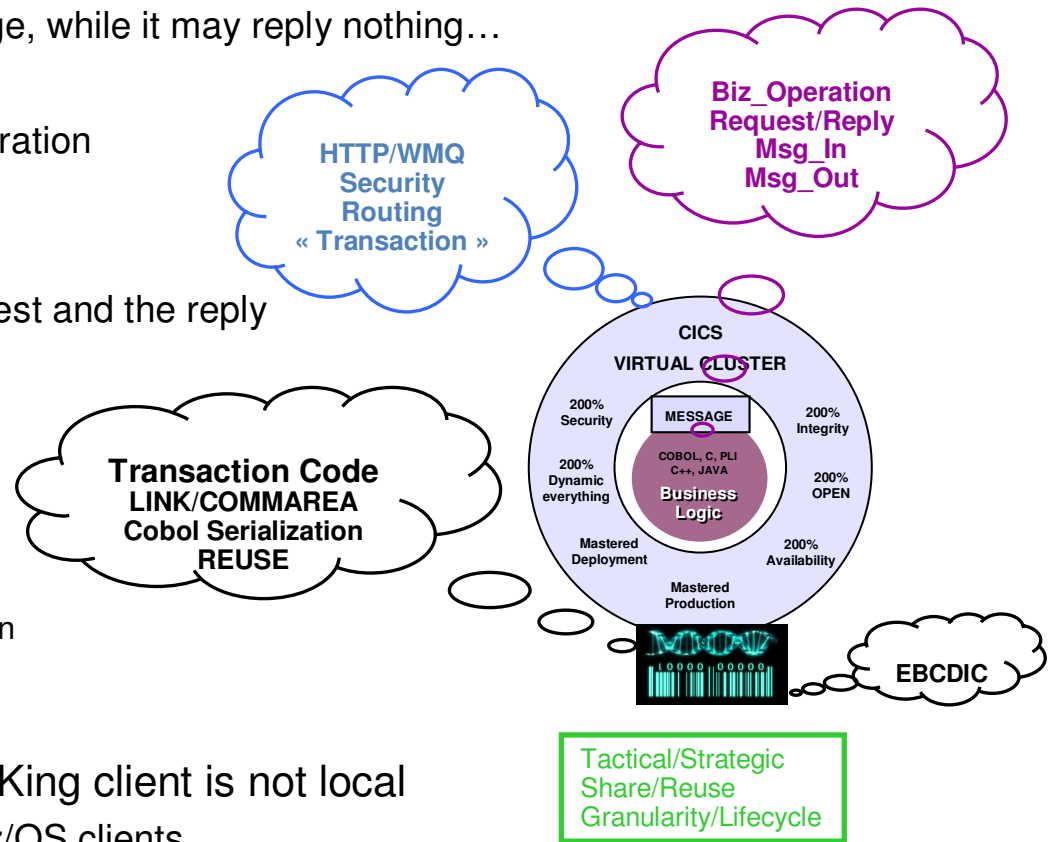


The CICS Application Interface



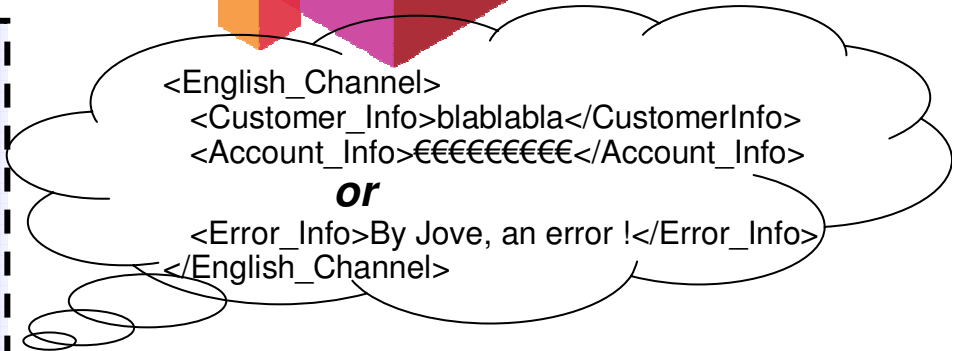
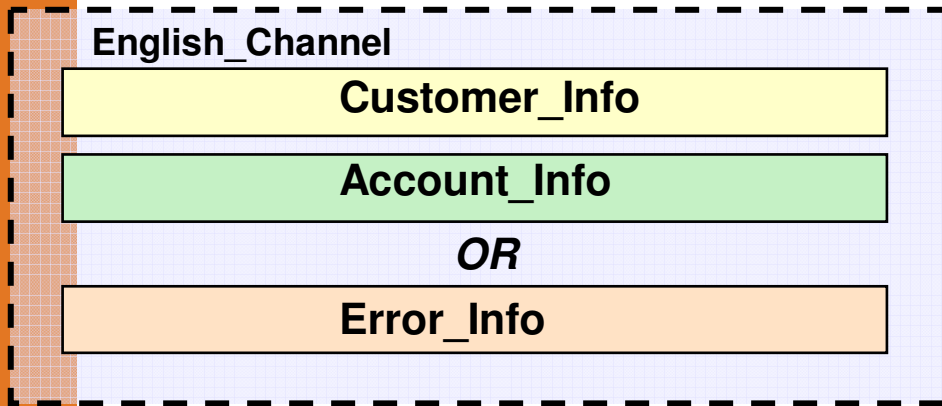
The CICS LINK interface

- LINK
 - Synchronous Request/Reply exchange, while it may reply nothing...
- PROGRAM(Operation)
 - The CICS program is the point of operation
- COMMAREA(CICS_Message)
 - 32kb
 - Common envelope between the request and the reply
 - messy redefines
 - Low cost: this is a pointer reference
- CHANNEL(Message)
 - « No limit »
 - Dynamic content
 - structured, message abstraction, clean
 - CICS cost: must use the CICS API
- A LINK becomes a DPL when the LINKing client is not local
 - CICS Function Shipping or EXCI for z/OS clients
 - ECI for CICS TG connector clients
 - DPL specifics
 - TRANSID(CICS workload identifier), a technology dependant default can be assigned
 - TIMEOUT, a specific value, a global value or ignored depending on the architecture
 - SYNCONRETURN to decide on the « transaction » type: local or global

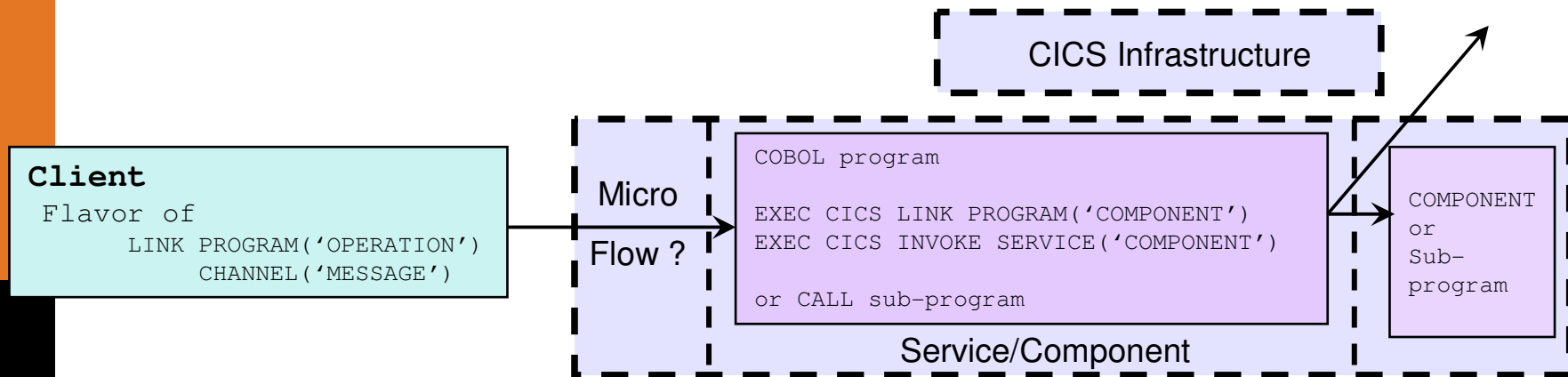




The CICS message envelope

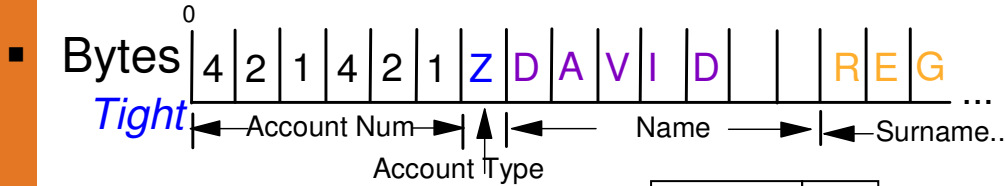


- Dynamic interface through an LCRUD interface
- Fully controlled by CICS: garbage collection, monitoring/statistics, data conversion, DPL optimisation
- More than a unlimited COMMAREA: reuse, maintenance, structure, CICS neutrality, abstraction





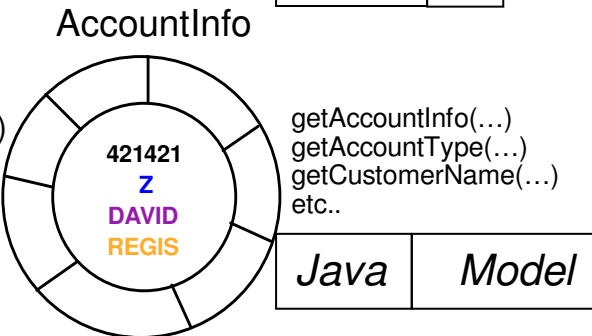
The message



01 GGX082-KZMFO-QQ. *static*
 05 GGX082-KZFON PIC 9(6).
 05 GGX082-KZTPAC-T PIC X.
 05 GGX082-KZTPAC-CN PIC X(9).
 etc...

Object
Permissive

setAccountNumber(...)
 setAccountType(...)
 setCustomerName(...)
 etc..



Int account_Number;
 String account_Number;
 String account_Type;
 etc..

« flexible »
 bytes[];

XML
Defined

```
<accountInfo>
  <ggx082_kzfon>421421</ggx082_kzfon>
  <accountType>Z</accountType>
  <ggx082_kztpac_n>DAVID...
```

```
<complexType name="ggx082_kzmfo_qq">      adaptive
  <sequence>
    <element form="unqualified" name="ggx082_kztpac_cn">
      <simpleType><restriction base="string">
        <maxLength value="9"/>
        <whiteSpace value="collapse"/>
      </restriction>...
```

Web Services *Interoperability*

JSON
Emerging

```
{ "accountInfo" : {
  "ggx082_kzmfon" : "421421",
  "accountType" : "Z",
  "ggx082_kztpac_n" : " DAVID ", ...}}
```



SF 1D SBA 11 AID Enter key IC 13 ...
 7D1D114040421421131D114DF0Z1D115CF0DAVID

Web 2.0 *Simple WS*

ASCII <> EBCDIC <> UNICODE

loves character messages, always an issue © 2013 IBM Corporation



CICS Integration options



CICS strategic integration

1. **SOAP Web Services over HTTP (or WebSphere MQ)**
2. **JEE JCA : CICS TG or WOLA (WebSphere AS for z/OS)**
3. **Asynchronous messaging using WebSphere MQSeries**
4. ***HTTP: Atom and REST***

A nice whitepaper: Delivering quick access to CICS systems using strategic integration options: <http://www.ibm.com/cics/tserver/v32/library/#wpapers>

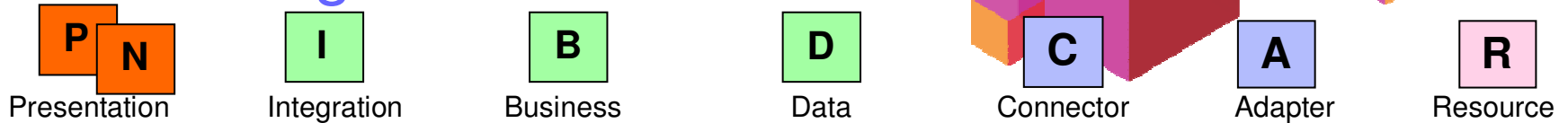
A nice redbook:



SG24-5466-06 on www.redbooks.ibm.com



CICS Integration



Best practice = no mix...

Web Services, provider & requester

High QoS, abstractions, XML, standards, tooling, repository, loose coupling

JCA - CICS TG/WOLA

High performance & QoS, JEE standards, tooling, CICS coupling

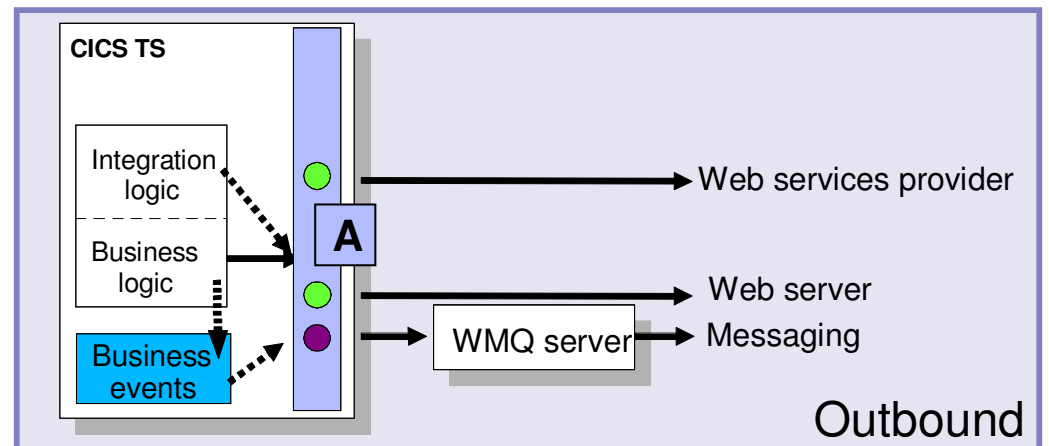
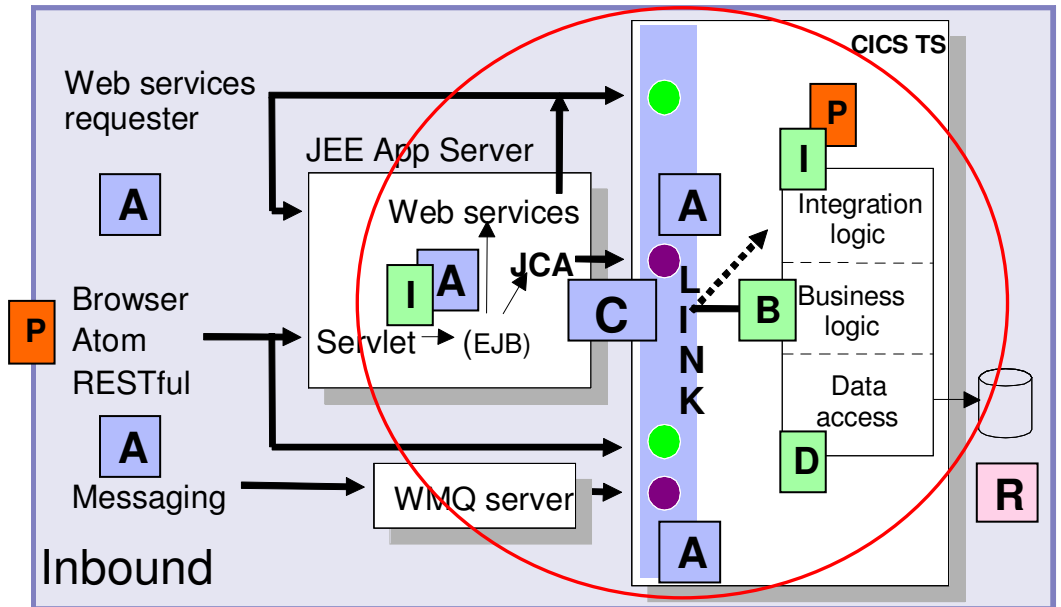
HTTP : Atom and "RESTful" interfaces

Simplification, HTTP conventions, emerging Web 2.0

Messaging - WebSphere MQ

Asynchronous, high QoS, WMQ performance, loose coupling, transport layer

Adapters <> transparency





Adapters

- Whatever is the implementation, adapters have always been required (proxy, skeleton, ...)

- Transport and protocol adapters

- Handled by CICS, i.e. Web Service provider
- Use CICS Apis, i.e. http server
- Address CICS program and CICS transaction issues

IT orientation

- Message serialization adapter

- Handles the message representation conversions
 - i.e. Java ascii or XML Unicode to/from COBOL ebcdic
- Handled by CICS, i.e. Web Services
- Handled by client code
 - Generated from tooling i.e. RDz J2C wizards
 - Can benefit from CICS ascii/ebcdic DFHCNV conversion services

Business Service orientation

- CICS LINK or LINK COMMAREA transparency adapter

- i.e. Abstract service/operation<>program name, Abstract message<>COMMAREA

- Service orientation adapter

- Transforms a COBOL or IT interface into a Business interface
 - Field selection, sensible names, fine or coarse granularity, error representation



CICS application interfaces integration

CICS TS

3270 datastream

Client : i.e. **JCA** or EClv2

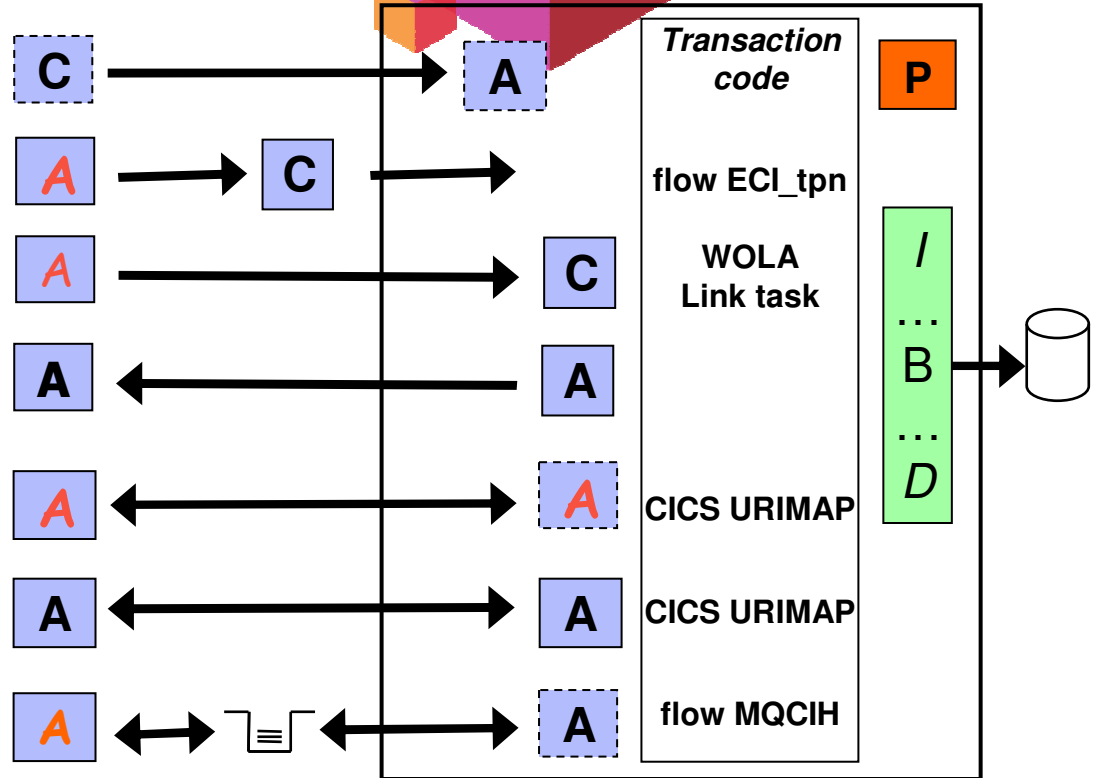
Client : i.e. JCA over WOLA

Server : i.e. WOLA CICS to EJB

Client or server: **Web Service**

Client or server : i.e. HTTP ...Web 2.0

Client or server: WebSphere **MQ**



- Adapter **A** is supported by tooling
- Adapter **C** can be CICS facilities
- **Blue** is not "CICS aware"
- Other is "CICS aware"
- The connector is HATS, CICS TG or WOLA
- Connector **C** is optional
- **Green** may or may not be "CICS aware"
- Adapter **A** can implement integration logic (i.e. grain)



**Connectivité CICS,
de la théorie à la pratique:
en route vers le Cloud.
???**



The journey from my CICS perspective

1. SOA is an evolution of the on-demand initiative for IT Applications
 - Integration/interoperability from Web Services technologies
 - SOAP over HTTP
 - WSDL
 - Abstract Services exposed from programs
 - Business services
 - Operations <> messages
 - Bindings
 - Real world implementation
 - Components
 - SCA

2. Cloud is an evolution of the on-demand initiative for IT Infrastructures
 - On-demand Provisioning (self-service)
 - Applications ...a collection of Services
 - Rapid elasticity
 - We have the event technology...
 - CICS is quite good at « Broad Network Access », « ressource Pooling »
 - and Measured Service
 - While focussed on transaction codes...

OSGi like Versioning



Simple showcase



The redbl原因 project story

- I required a **simple** showcase for a CICS Dynamic Scripting demo
- Started reusing an old library sample I found on the web
 - JavaScript/PHP frontend to a Program/Commarea backend
 - REST patterns + JSON message representation
- CICS DS was a good opportunity to introduce Java support within CICS
 - Created a JCICS REST frontend from a std Helios then Indigo Eclipse IDE
 - Used the open source J2J SourceForge project for JSON support
 - J2J provides an intuitive way to convert Java to JSON and JSON to Java.
 - Annotate any Java class then use JsonWriter to emit JSON from java or JsonReader to convert JSON to java objects.
- Enabled a tactical Web Service support
 - Used the SOAP UI open source solution
- Abstracted the Program/Commarea into a Service interface
- Reused this Service interface from the CICS TG
- Leveraged on the CICS TS V5.1 cloud for deployment awareness



The existing interface

- LINK PROGRAM('DAVREDBB') COMMAREA(.....)

```

EDIT      DAVIDR.JSONSAMP (DAVREDBB) - 01.10      Columns 00007 00072
Command ==> _____ Scroll ==> BSR
004900  01 DFHCOMMAREA.
005000  *   DAVJNLIB COMMAREA structure
005100    03 LIB-REQUEST-TYPE      PIC X(6).
005200    03 LIB-RETURN-CODE      PIC 9(2).
005300    03 LIB-ITEM-COUNT       PIC 9(4).
005400    03 LIB-BOOK-ITEM OCCURS 24 TIMES.
005500      05 BOOK-ITEM-REF      PIC 9(04).
005600      05 BOOK-REDB-REF      PIC X(12).
005700      05 BOOK-TITLE         PIC X(50).
005800      05 BOOK-AUTHOR        PIC X(20).
005900      05 BOOK-PUBYEAR      PIC X(04).
006000      05 BOOK-RATING       PIC 9(02).
006100      05 BOOK-LOAN-STATUS  PIC 9(02).
006200      88 BOOK-ONLOAN VALUE 1.
006300      88 BOOK-UNLENT VALUE 0.
006400      05 BOOK-BORROWER    PIC X(20).
006500      05 FILLER            PIC X(18).
F1=Help   F2=Split   F3=Exit   F5=Rfind
F8=Down   F9=Swap   F10=Left  F11=Right  F
MA  A
001800  01 DAVJNLIB-CONSTANTS.
001900    03 DAVJNLIB-RESPCODES.
002000      05 DAVJNLIB-OK      PIC 9(2) VALUE 0.
002100      05 DAVJNLIB-NOTFOUND PIC 9(2) VALUE 1.
002200      05 DAVJNLIB-ONLOAN  PIC 9(2) VALUE 2.
002300      05 DAVJNLIB-DUPLICATE PIC 9(2) VALUE 3.
002400      05 DAVJNLIB-INVREQ  PIC 9(2) VALUE 4.
002500      05 DAVJNLIB-FULL    PIC 9(2) VALUE 5.
002600      05 DAVJNLIB-ERROR   PIC 9(2) VALUE 99.
002700    03 DAVJNLIB-FUNCTIONS.
002800      05 LIB-FUNC-LIST     PIC X(6) VALUE 'LIST'.
002900      05 LIB-FUNC-QUERY    PIC X(6) VALUE 'QUERY'.
003000      05 LIB-FUNC-ADD     PIC X(6) VALUE 'ADD'.
003100      05 LIB-FUNC-DELETE  PIC X(6) VALUE 'DELETE'.
003200      05 LIB-FUNC-BORROW  PIC X(6) VALUE 'BORROW'.
003300      05 LIB-FUNC-RETURN  PIC X(6) VALUE 'RETURN'.
003400      05 LIB-FUNC-RATE    PIC X(6) VALUE 'RATE'.

```

- Type of request
- Return code
- Item number (max 24)
- Redbook details

- COMMAREA is INPUT/OUTPUT = 3180 bytes
- 'Operations' are LIB-REQUEST-TYPEs

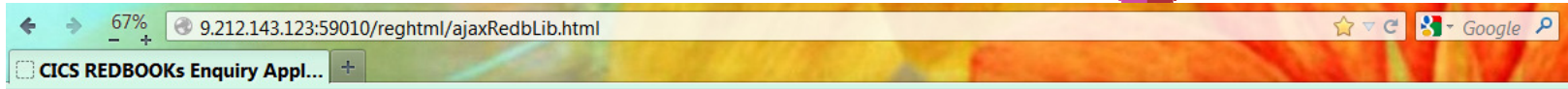


The message exchanges

- **LIST operation**
 - Supplies ...the LIST request...
 - Returns LIB-ITEM-COUNT * LIB-BOOK-ITEMs details
- **QUERY operation**
 - Supplies the LIB-BOOK-REF(1) to query about
 - Returns the LIB-BOOK-ITEM(1) details of BOOK-ITEM-REF(1)
- **ADD operation**
 - Supplies the LIB-BOOK-ITEM(1) details to add to the redbook library
 - Returns the BOOK-ITEM-REF(1) index of the added LIB-BOOK-ITEM(1)
- **DELETE operation**
 - Supplies the LIB-BOOK-REF(1) to delete from the redbook library
 - Deletes the BOOK-ITEM-REF(1) entry
- **BORROW operation**
 - Mark BOOK-ITEM-REF(1) as Borrowed & update the BOOK-BORROWER info
 - No validity check (relies on PL)
- **RETURN operation**
 - Mark BOOK-ITEM-REF(1) as not Borrowed & delete BOOK-BORROWER info
 - No validity check (relies on PL)
- **RATE operation**
 - Update the BOOK-ITEM-REF(1) BOOK-RATING info
 - Check on BOOK-ONLOAN to enable the rating



The JavaScript user experience



CICS REDBOOKS Library Application

■ Demo

To sort a column, just click on a column header
To enter a Borrower rating, use the B.Rating column
The first 13 entries are protected, the remaining ones are yours :O)

Book No	Book Ref	Title	Author	Year	Rating	On Loan?	Borrower	B.Rating
10	SG24-6351-04	Threadsafe Considerations for CICS	G.Bogner, J.Tilling	2012	90	false	<input type="text"/> Borrow!	
20	REDP-4850-00	CICS Performance Series: CICS TS V4.2 Java Perf.	Graham Rawson	2012	75	false	<input type="text"/> Borrow!	
30	SG24-5466-06	CICS and SOA: Architecture and Integration Choices	N.Williams, R.David	2012	99	true	YouShould ReadIt....	<input type="text"/> Return! <input type="text"/> Rate it!
40	REDP-4810-00	Gaining Insight into IBM CICS Systems with Events	Catherine Moxey	2011	80	true	Real Interest	<input type="text"/> Return! <input type="text"/> Rate it!
50	SG24-7850-00	z/OS Identity Propagation	P.Wakelin, A.Roessle	2011	70	false	<input type="text"/> Borrow!	
60	SG24-5275-03	Java Application Development for CICS	C.Rayns, G.Burgess	2009	90	true	Steve Wall	<input type="text"/> Return! <input type="text"/> Rate it!
70	SG24-7658-00	Securing CICS Web Services	O'Grady & Williams	2008	68	false	<input type="text"/> Borrow!	
80	SG24-7815-00	Smarter Banking with CICS Transaction Server	F.Jarassat, V.Eibel	2010	79	true	Open YourMind	<input type="text"/> Return! <input type="text"/> Rate it!
90	SG24-7819-00	Extend the CICS Explorer: A Better CICS Management	S.Wall, J.Taylor	2010	60	false	<input type="text"/> Borrow!	
100	REDP-4809-00	CICS Event Processing: new features in V4.2	C.Moxey, J.He	2011	79	false	<input type="text"/> Borrow!	
110	SG24-7924-00	Introduction to Dynamic Scripting	R.David, J.O'Grady	2011	70	true	Project Zero	<input type="text"/> Return! <input type="text"/> Rate it!
120	SG24-7126-01	Application Development for CICS Web Services	P.Cooper, P.Klein	2010	88	true	Soa Journey	<input type="text"/> Return! <input type="text"/> Rate it!
130	SG24-7657-00	Implementing CICS Web Services	M.Pocock, C.Rayns	2008	90	true	ILove Soa	<input type="text"/> Return! <input type="text"/> Rate it!
140	<input checked="" type="checkbox"/> SG24-7952-00	CICS Transaction Server from Start to Finish	E.Woerner, C.Carlin	2011	87	false	<input type="text"/> Borrow!	
150	<input checked="" type="checkbox"/> REDP-4824-00	Impl of Popular Business Solutions with CICS Tools	P.Siddell, E.Higgins	2012	67	false	<input type="text"/> Borrow!	

Use the following form to add a new Redbook entry.

Add a Book					
Book Ref:	Title:	Author:	Year:	Rating:	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Add book!"/> <input type="button" value="Clear Fields"/>

```

Raw response data or Error information :
Response from CICS: {"statusMessage": "Successfully updated status of book 160 with new Borrower info"}
Response from CICS: {"statusMessage": "Successfully updated status of book 160with new Rating"}
Response from CICS: {"statusMessage": "Successfully deleted book 160 from the Library"}
Response from CICS: {"statusMessage": "Successfully updated status of book 10 with new Borrower info"}

```



Tactical versus Service Technical versus Abstract

- 3180 bytes COMMAREA exchanges
 - CICS TG Headers + Body
 - WMQSeries Headers + opt MQCIH + Body

- 11770 > < 8199 bytes XML COMMAREA like exchanges
 - Web Services « submit & go », COMMAREA SOAP body in a SOAP envelope

- RedbLib Service Operations, In-Out exchange patterns, Optimized Messages

<ul style="list-style-type: none"> – LIST – CREATE – REPLACE – UPDATE <li style="margin-left: 20px;">• Borrow <li style="margin-left: 20px;">• Rate <li style="margin-left: 20px;">• Return – DELETE 	<p>small In (?)</p> <p>In as required</p> <p>In as required</p> <p>small In</p>	<p>Out as required</p> <p>small Out</p> <p>small Out</p> <p>small Out</p>	<p>Meaningfull</p> <p>Meaningfull</p> <p>Meaningfull</p> <p>Meaningfull</p>
	<ul style="list-style-type: none"> • REST by nature • Web Service by maturity 	<ul style="list-style-type: none"> • CICS TG by evolution • WMQSeries by the time 	
	small In	small Out	Meaningfull



Web Services today = SOAP over HTTP + WSDL

- Transport and protocol
 - Transparent support supplied by the CICS infrastructure
 - Low level http or WMQ
 - SOAP
- Operation Identification
 - Transparent support supplied by the CICS infrastructure: URIMAP magic
 - DFHLS2WS (OPERATION-NAME) or DFHWS2LS (WSDL) (best practice ?) utility
- Message serialization
 - Transparent support supplied by the CICS infrastructure...when (likely) supported
 - COBOL redefines lack of support should not be a problem
 - Reply redefines the Request, or multi-operation redefine
 - DFHLS2WS
 - COBOL filler are not exposed
 - Trailing whitespaces can be suppressed
 - XML namespaces best practices supported
 - Rational Developer for system z (RDz) support for complex interfaces



Service oriented web services

- Simple DFHLS2WS is a straight CICS program interface to XML adapter
 - Bottom-up approach
 - Focus is on IT interoperability

- Business service orientation may require business oriented optimization
 - Simple no-code
 - Field names can be changed to have a business meaning (good for silos interoperability)
 - Fields can be hidden from the interface exposed: filler
 - A complex multipurpose sedimentary interface can be cleaned and split

 - Common issue: list optimization
 - Lists exposed tend to be fixed, variable minOccurs/maxOccurs may be required
 - DFHLS2WS + WSDL modification + DFHWS2LS, OR Top-down from DFHLS2SC
 - Requires a « simple wrapper » code for this meet-in-the-middle approach
 - DFHWS2LS INLINE-MAXOCCURS-LIMIT to trigger the behavior

 - May be an issue
 - XML technologies consume IT resources: coarse or fine grain ?
 - CICS integration logic to increase the grain, pseudo-conversational minding may be counterproductive

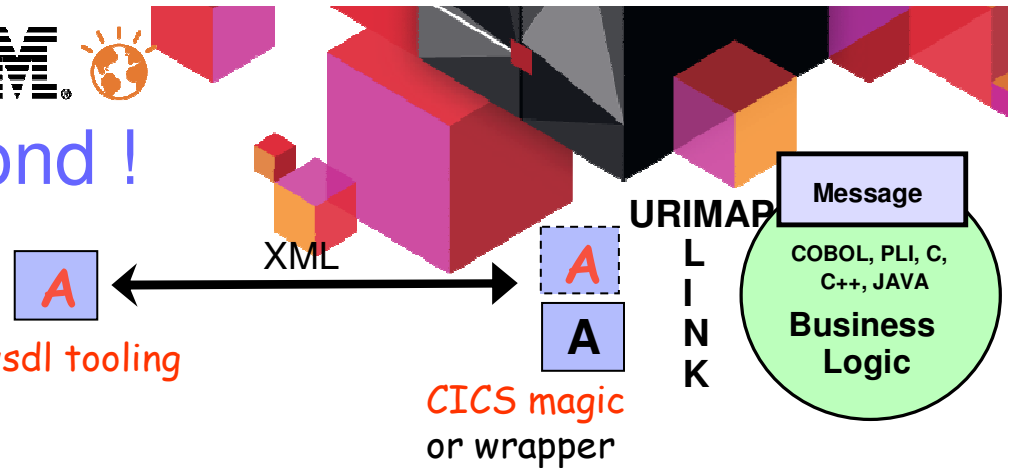
 - New code or extension
 - Use a top-down approach, think business service ! WSDL + DFHWS2LS



Web Services in 5 second !

Client or server: **Web Service**

Non CICS aware = *wSDL2myWay* and *myWay2wSDL* tooling



Message Exchange Patterns

- Explicit WSDL 2.0
 - In-Only where no response is required, i.e. pushed notification
 - In-Out which is an RPC equivalent, i.e. request/response
 - In-Optional-Out for a possible combination of the previous patterns
 - Robust-In-Only response required only in case of an error (SOAP Fault)
- Implicit WSDL 1.0
 - In-Out when the WSDL defines a request and a response messages
 - In-Only when the WSDL does not define a response message

LOCALCCSID for specific EBCDIC<>UNICODE conversions, i.e. French 1147

Coupling

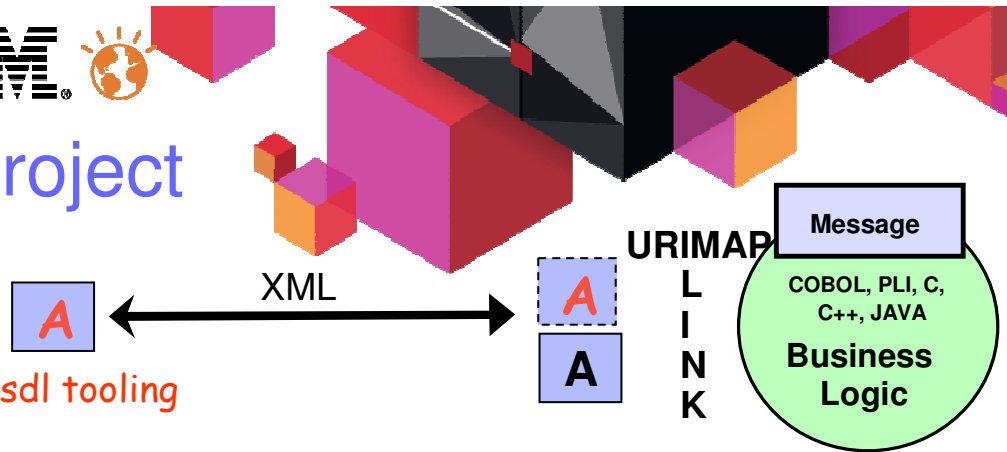
- IT perspective: Loosely coupled client/server technology
- Application interface perspective: « what happens if I modify a message ? »
 - non-object bottom-up approach tends to be tight, answer is « I must supply a new WSDL »
 - object bottom-up approach tends to ignore *real life concerns*, answer is « help yourself »
- To be balanced: *IT* value versus *Business* value versus *Lifecycle* value



Web Services redlib project

Client or server: **Web Service**

Non CICS aware = *wSDL2myWay* and *myWay2wSDL* tooling



▪ Straight COMMAREA exposure

- LINK COMMAREA technical adapter from DFHLS2WS
- Single operation service (WSDL packaging ?)
- `<.../redlib/commarea >` `< single operation >` `< CICS COMMAREA messages >`

▪ Demo

▪ Semi-Optimized « no change » exposure

- Filler + Business words COBOL copybook transformation from DFHLS2WS
- Quick win and Single operation service (WSDL packaging ?)
- `<.../redlib/commareaOpt >` `< single operation >` `< CICS semi-optimized messages >`

▪ Demo

▪ Optimized Service exposure

- Multi-operation service with dedicated business messages
- Top down approach from DFHLS2SC > xsd > wSDL modeling > DFHWS2LS
- Meet in the middle wrapper with simple CHANNEL/CONTAINER exposure
 - Multiple Container interface study in the to-do list !?
- `<..../redlib/optimized >` `< multiple business operations >` `< optimized business messages >`

▪ Demo

▪ Demo

- Added complex BIGLIST, dateTime, variable length, nillable, optional, unbound and Fault processing for demonstration, good for a Requester implementation.
 - BIGLIST = 0-10 * 1-100 * 0-200 * 0-1000 * 100bytes2000000000bytes potential ;O)



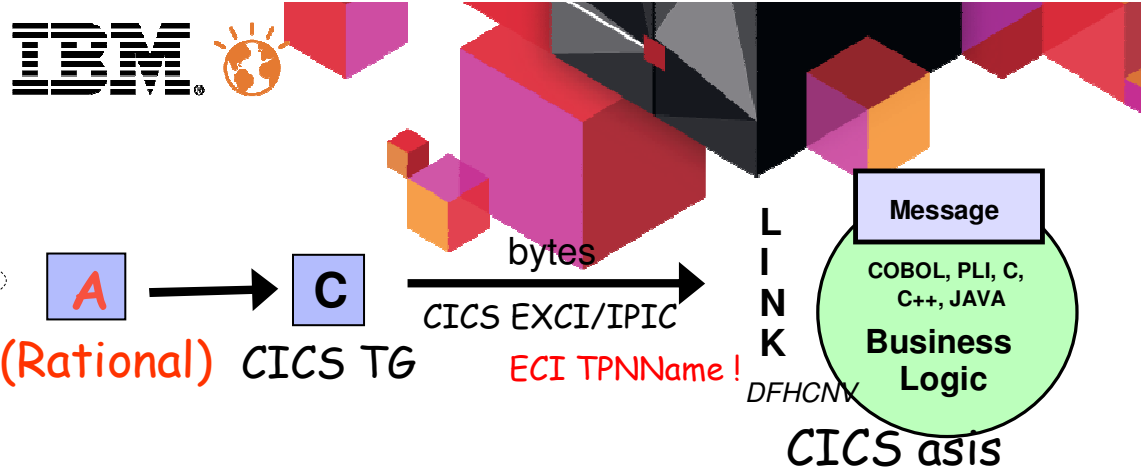
Optimized Service

- Top-Down (meet in the middle) WSDL, business operations & messages
- CICS interface
 - Containers DFHWS-OPERATION, RedbLib_MsgIntf
 - * Get the REQUEST MESSAGE from the current CHANNEL
 - EXEC CICS GET CONTAINER('RedbLib_MsgIntf')
 - SET(RedbLib-Msg-PTR)
 - FLENGTH(RedbLib-Generic-Message-Length)
 - END-EXEC.
 - Set Address of RedbLib-Generic-Message to RedbLib-Msg-PTR.
- COBOL interface
 - One optimized COPYBOOK per message from DFHWS2LS
- Issue with the semi-optimized option...Occurs...
 - DFHPI1010 03/25/2013 14:30:31 CICSFRAT 10951 XML generation failed. A conversion error (INVALID_ZONED_DEC) occurred when converting field itemRef for WEBSERVICE RedbLibCommaOpt.



JEE JCA today = CICS TG for z/OS

- Transport and protocol
 - Transparent support supplied by the CICS TG for z/OS connector product
 - CICS private transport and protocol
 - Supports JCA JEE optimizations, i.e. connection pooling (also true with WAS for Web Services)
- Operation Identification
 - Real CICS program name from a « function » property on the Client side
- Message serialization
 - Real CICS COMMAREA
 - Java Record representation (bytes) with null stripping to solve Commarea single envelope issues
 - Real CICS CHANNEL and CONTAINERS
 - Rational tooling (RAD or RDz) support from J2C tooling
 - JZOS record tooling support when the client code is deployed in WAS z/OS
- Supports the JCA Common Client Interface or CCI
 - ECICConnectionSpec and ECIIInteractionSpec
 - Interaction verb such as SYNC_SEND_RECEIVE
 - Function name such as the CICS program to LINK to
 - CICS COMMAREA or CHANNEL message
 - CICS transaction code used to identify the CICS workload (setTPNName method)
 - Rational tooling (RAD or RDz) support from J2C tooling



JEE JCA in a day !

Client : i.e. **JCA** or ECIv2

CICS aware = CICS coupling

Message Exchange Patterns

– CCI support

- SYNC_SEND_RECEIVE
- SYNC_SEND
+ SYNC-RECEIVE

which is a synchronous RPC equivalent request/wait/response

which is an asynchronous RPC equivalent,

note that it is not connectionless like WMQ

– ECIv2 support

- ECI_SYNC

ECI variation of the CCI SYNS_SEND_RECEIVE

DFHCVN macro table support for ASCII<>EBCDIC conversions

- Must be in sync with clients, conversion can be done on the Client side

Coupling

– IT perspective

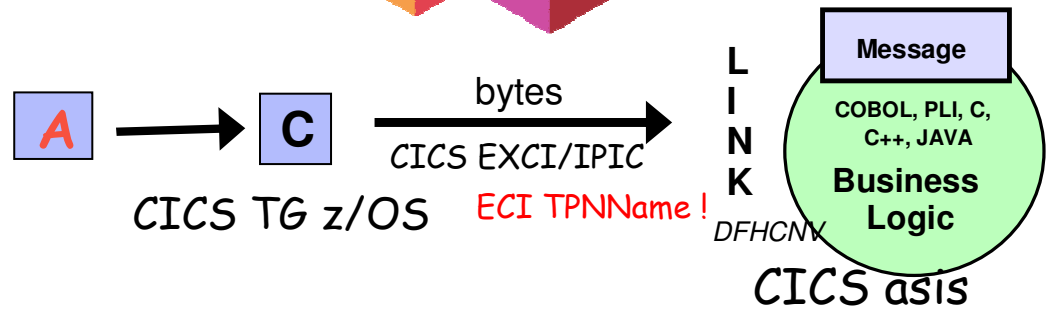
- Tight coupling through the CICS TG connector
- Tight coupling through the DFHCVN (Thanks to character data messages)

– Application interface perspective

- Tight coupling with the CICS bytes message (Thank's to the CHANNEL message)



JCA redlib project



- Java ECIRequest
 - LINK COMMAREA technical adapter
 - Reuse CH50 CTGPing supportpac, with minor changes
 - Program name
 - Commarea content
 - Messages exchange logging

▪ Demo

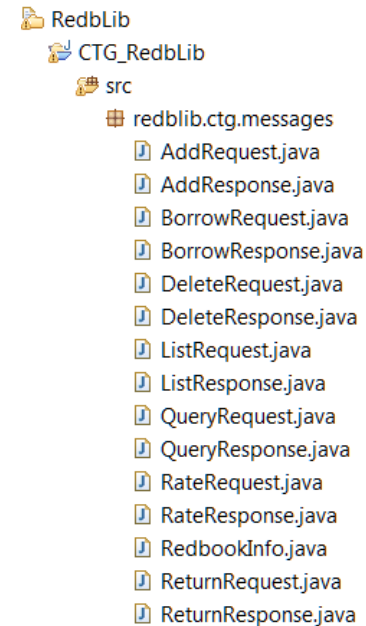
- Java CCI application
 - Business layer adapter
 - LINK COMMAREA to CICS
 - Structured code, message beans
 - Name=value pairs for input message tests
 - Web Service wrapper reuse in the to-do list. Single and multiple Container interfaces.
 - DFHWS-OPERATION + RedbLib_MsgIntf ?

▪ Demo



The redlib to-do list

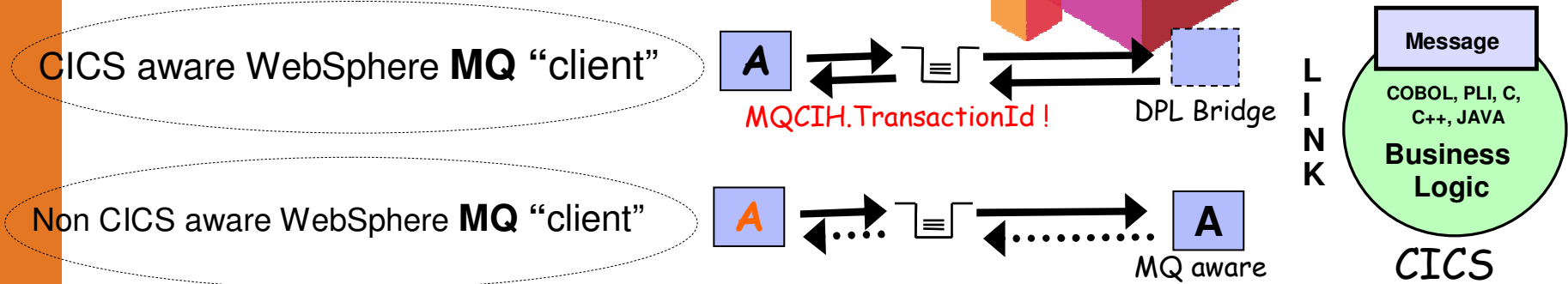
- Web Services
 - Use a multi container interface ?
- JCA
 - Reuse the optimized Web Service wrapper ?
 - DFHWS-OPERATION + RedbLib_MsgIntf
 - Use a multi container interface ?
- Channel interface
 - Request_Type container versus meaningful name ?
 - Meaningful request/response container names



REDBLIBPort			
* RedbLibLIST			
input	ListRequestPart	LISTRequest	
output	ListResponsePart	LISTResponse	
* RedbLibQUERY			
input	QueryRequestPart	QUERYRequest	
output	QueryResponsePart	QUERYResponse	
* RedbLibADD			
input	AddRequestPart	ADDRequest	
output	AddRequestResponsePart	ADDResponse	
* RedbLibDELETE			
input	DeleteRequestPart	DELETERequest	
output	DeleteResponsePart	DELETEResponse	
* RedbLibUPDATE			
input	UpdateRequestPart	UPDATERequest	
output	UpdateResponsePart	UPDATEResponse	



WebSphere MQSeries in 2 days



- Message Exchange Patterns
 - Standard WMQ exchanges message types, request/reply with the DPL bridge
- ASCII<>EBCDIC conversions
 - Standard WMQ facilities
- Coupling
 - IT perspective
 - Some coupling through the WebSphere MQSeries product,
 - MQFMT_STRING best practice lowers the encoding coupling
 - Application interface perspective
 - Loose coupling from the QUEUE object
 - Tight coupling with the CICS bytes message (CICS XMLTRANSFORM APIs ?)



WebSphere MQSeries...in our context

- Transport and protocol
 - CICS supplies transparent support for the QUEUE MANAGER connection
 - MQ aware applications access the QUEUE object, MQ DPL bridge otherwise ?
- Operation Identification
 - May be, (can, should ?), the QUEUE name for MQ aware applications
 - Requires private conventions otherwise, i.e. a header
 - MQ DPL Bridge (CICS TG ECI like)
 - Real CICS program name from the first bytes of the MQ message
- Message serialization
 - WMQ as usual for MQ aware application. Thank's to the String format
 - MQ DPL Bridge (CICS TG ECI like)
 - Real CICS COMMAREA, null stripping from MQCIH_REPLY_WITHOUT_NULLS
 - CICS CHANNEL support with CICS TS V5.1...
 - Rational J2C tooling for JEE JMS
- The CICS MQ DPL Bridge interface (CTG ECI like)
 - MQCIH header, MQMD MQFMT_CICS, MQCIH.Format for data conversion
 - MQCIH.TransactionId

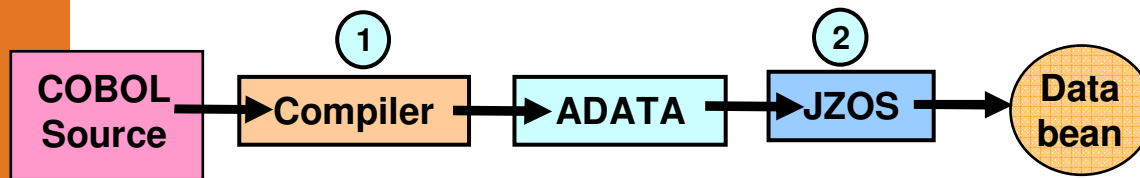
Web 2.0: Dynamic Scripting

Tactical solution !

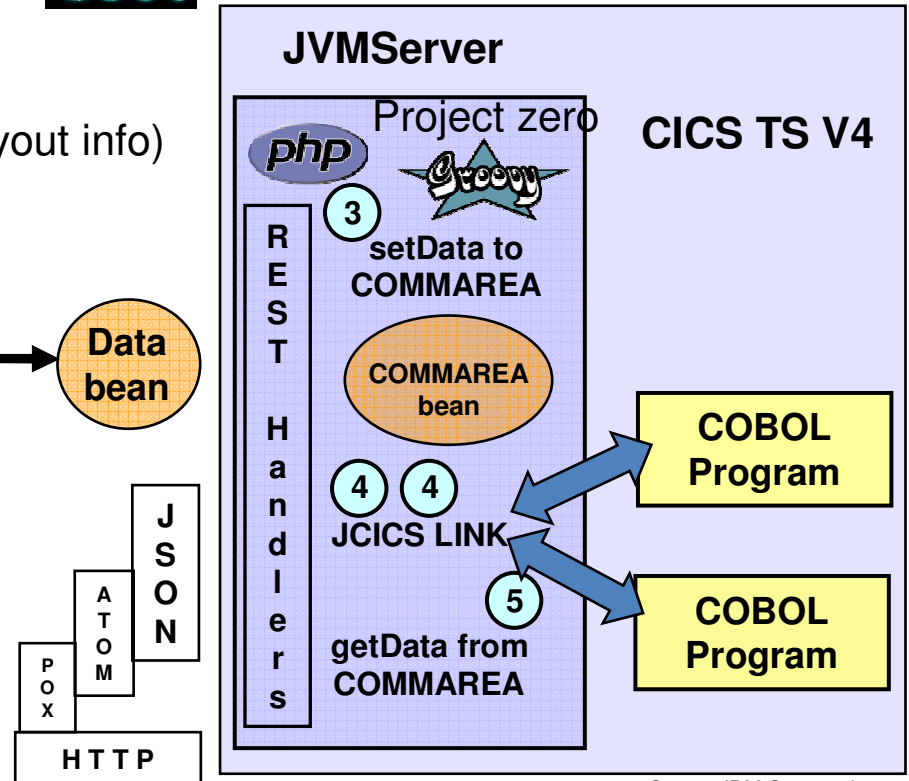
- Simplicity, speed, zero complexity, zero obstacles
- Dynamic Scripting and CICS
 - Build a Java object representation of your COBOL copybook with JZOS
 - Serialize, access the CICS resource using JCICS, de-serialize
 - CICS execution container DNA benefits



- 1 Generate ADATA file from compiler (data layout info)
- 2 Generate Java Data bean using JZOS



- 3 setData() in COMMAREA object
- 4 JCICS LINK to business logic
- 5 getData() from COMMAREA object





JZOS recgen tooling

- The ADATA compile option

```
...
//COB EXEC PGM=IGYCRCTL,REGION=&REG,
//PARM='NODYNAM,LIB,OBJECT,RENT,APOST,ADATA,LIST,TRUNC(BIN)'
...
//COB.SYSADATA DD DISP=SHR,DSN=DAVIDR.JZOS.ADATA(DAVREDBB)
...
```

- The jzos.recgen.jar magic

```
java com.ibm.jzos.recordgen.cobol.RecordClassGenerator adataFile="//DAVIDR.JZOS.ADATA(DAVREDBB)"
symbol=WS-COMMAREA outputDir=. package=redbllib class=RedbLib_Commarea genCache=false
```

...a second for DAVJNLIB-CONSTANTS and class RedbLib_Constants

- ...generates two nice java beans

```
- RedbLib_Commarea.java and RedbLib_Constants.java
package redbllib;
import com.ibm.jzos.fields.*;
// Generated by com.ibm.jzos.recordgen.cobol.RecordClassGenerator
public class RedbLib_Commarea { ....
public static final int WS_COMMAREA_len = 3180;

/** <pre>
* DAVJNLIB COMMAREA structure
 03 LIB-REQUEST-TYPE PIC X(6). </pre> */
protected static StringField LIB_REQUEST_TYPE = factory.getStringField(6);
...

```



Web 2.0 today = JSON over HTTP

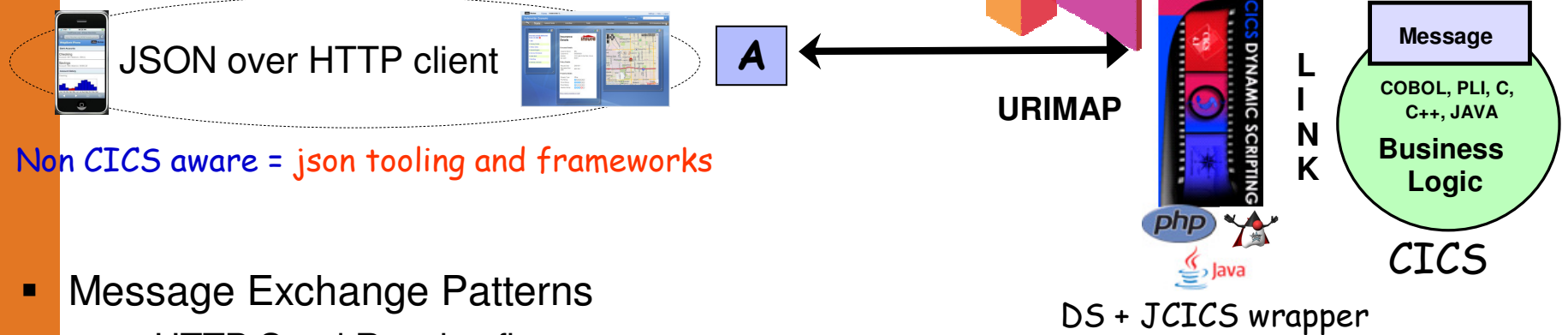
- Transport and protocol
 - Transparent support supplied by the CICS infrastructure
 - Dynamic scripting configuration
- Operation Identification
 - Transparent support supplied by the CICS infrastructure: URIMAP magic
 - Dynamic scripting configuration
- Message serialization
 - JSON <> scripting language supplied by dynamic scripting facilities, i.e.
 - php

```
json_decode($HTTP_RAW_POST_DATA);
json_encode($message)
```
 - Projectzero

```
zput("/request/view", "JSON");
zput("/request/json/output", $obj);
render_view();
```
- CICS LINK interaction from JCICS
- REST conventions from event handlers, i.e.
 - php

```
$event = zget('/event/_name');
switch($event) {
case 'list': // GET on a Collection
case 'retrieve': // GET on a Member of a Collection
case 'create': // POST on a new Member of a Collection
```


JSON over HTTP in 30 seconds

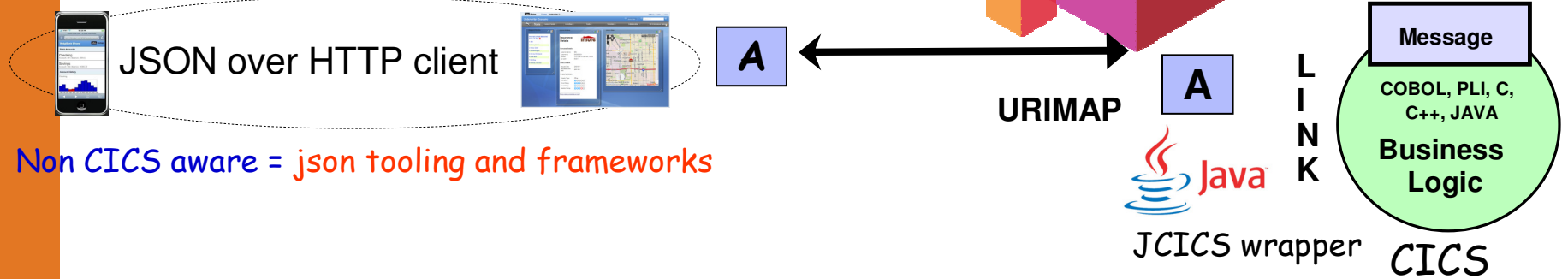


Non CICS aware = json tooling and frameworks

- Message Exchange Patterns
 - HTTP Send Receive flows
- Data conversion
 - Java file encoding
 - Native Java
- Coupling
 - IT perspective
 - Loose HTTP coupling
 - Tight adapter coupling
 - Application interface perspective
 - Similar to the Web Services support, JSON can be seen as a lightweight XML
 - No WSDL implies client/server tighter coupling



REST/JSON redlib project



Non CICS aware = json tooling and frameworks

- REST http protocol
 - Simple java HttpWrapper class
 - Uses JCICS
- JSON messages
 - Simple J2J open source mapper
- Decided to enable CICS architecture choices
 - REST handler <> Business logic wrapper separation (CICS LINK)
 - Java object byte[] serialization in a CICS Channel
 - RMI over IIOP made simple ! Local JEE people like it !
- To-do: Java enable the Web Service wrapper !
 - Statement Of Direction (SOD)
 - AXIS2 pipeline ?!

▪ Patterns

▪ Demo



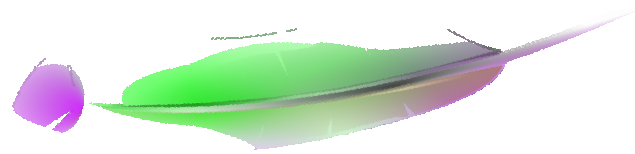
Statements of direction

- ENUS213-093
 - IBM intends to deliver enhanced support for mobile applications interacting with IBM CICS Transaction Server for z/OS (CICS TS) services, using the lightweight data-interchange format JavaScript Object Notation (JSON).

 - In addition, IBM intends to introduce support for deploying qualified new CICS TS workloads on IBM System z New Application License Charges (zNALC) Logical Partitions (LPARs). Qualified new CICS TS applications, including approved mobile and service-enabled applications running in the CICS TS Java Virtual Machine (JVM) Server, will be eligible for CICS TS one-time-charge (OTC) pricing when deployed to a zNALC LPAR.



CICS TS V5.1
Liberty is...



A LIGHTWEIGHT



COMPOSABLE

**servlet/jsp
Web container**



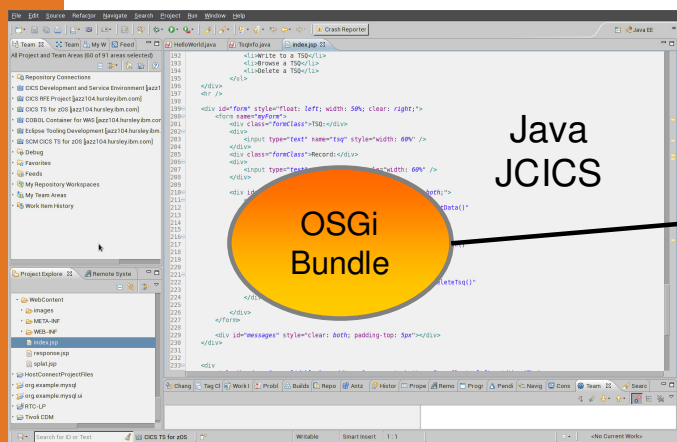
FA

... 'Profile'
of WebSphere Application Server

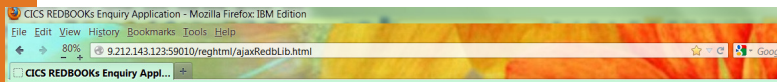
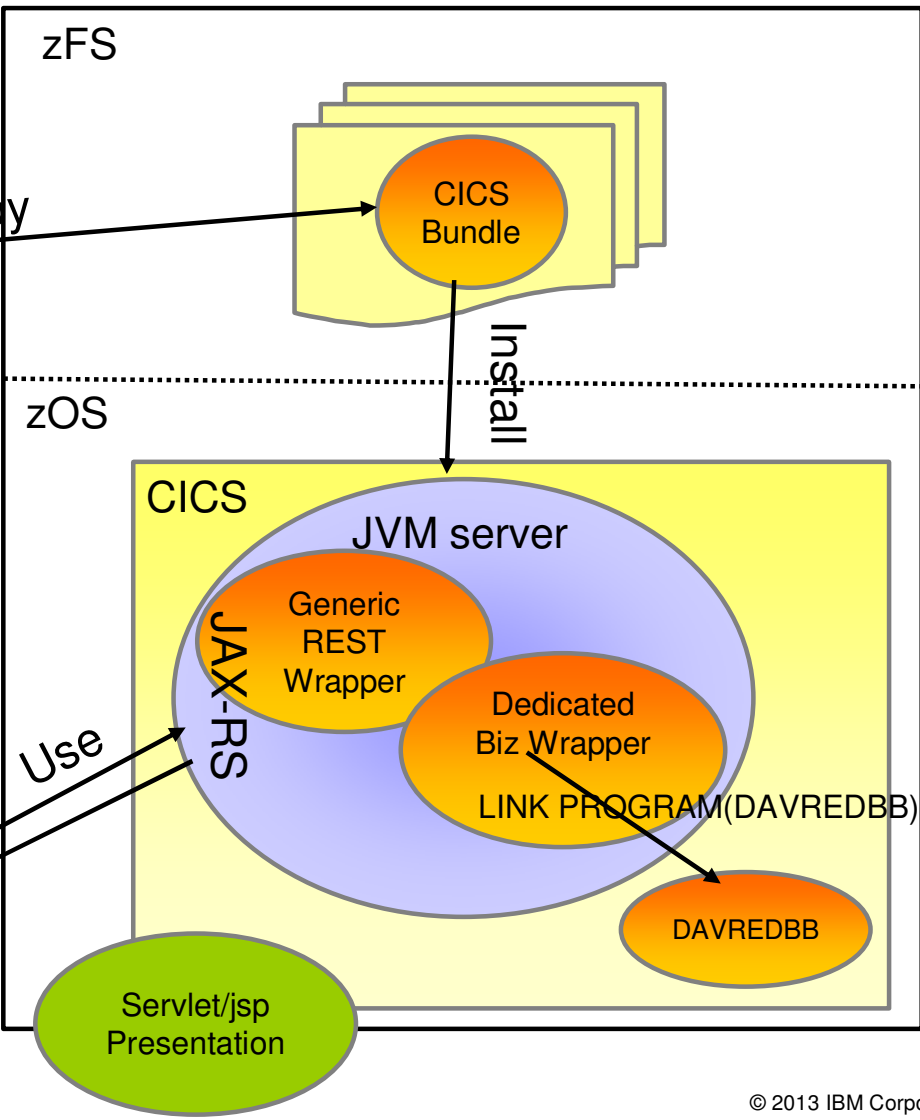


The redblif vNext project...

Eclipse + CICS SDK plugin



Deploy



Book No	Book Ref	Title	Author	Year	Rating	On Loan?	Borrower	B.Rating
10	SG24-6351-04	Threadsafe Considerations for CICS	G.Bogner, J.Tilling	2012	90	false		
20	REDP-4850-00	CICS Performance Series: CICS TS V4.2 Java Perf.	Graham Rawson	2012	75	false		
30	SG24-5466-06	CICS and SOA: Architecture and Integration Choices	N.Williams, R.David	2012	99	true	YouShould Readt...	Rate it!
40	REDP-4810-00	Gaining Insight into IBM CICS Systems with Events	Catherine Moxey	2011	80	true	Real Interest	Rate it!
50	SG24-7850-00	z/OS Identity Propagation	P.Wakelin, A.Roessele	2011	70	false		
60	SG24-5275-03	Java Application Development for CICS	C.Rayns, G.Burgess	2009	90	true	Steve Wall	Rate it!
70	SG24-7658-00	Securing CICS Web Services	O'Grady & Williams	2008	68	false		
80	SG24-7815-00	Smarter Banking with CICS Transaction Server	F.Jarassat, V.Elbel	2010	79	true	Open YourMind	Rate it!
90	SG24-7819-00	Extend the CICS Explorer: A Better CICS Management	S.Wall, J.Taylor	2010	60	false		
100	REDP-4809-00	CICS Event Processing: new features in V4.2	C.Moxey, J.He	2011	79	false		
110	SG24-7924-00	Introduction to Dynamic Scripting	R.David, J.O'Grady	2011	70	true	Project Zero	Rate it!
120	SG24-7126-01	Application Development for CICS Web Services	P.Cooper, P.Xlein	2010	88	true	Soa Journey	Rate it!
130	SG24-7657-00	Implementing CICS Web Services	M.Pocock, C.Rayns	2008	90	true	iLove Soa	Rate it!
140	SG24-7952-00	CICS Transaction Server from Start to Finish	E.Woerner, C.Carlin	2011	87	false		
150	REDP-4824-00	Impl of Popular Business Solutions with CICS Tools	P.Siddell, E.Higgins	2012	67	false		

Use the following form to add a new Redbook entry.

Add a Book

Book Ref: Title: Author: Year: Rating:



CICS Cloudification



Simple concepts

- Application
 - A collection of resources
 - A set of entry points/operations
 - A set of dependencies

- Platform
 - A collection of resources
 - A collection of regions, region types

- Policies
 - A collection of behavior rules for applications and/or platforms
 - Graduated actions

- Bindings
 - A set of application deployment specifications
 - Addresses platform characteristics transparency for applications



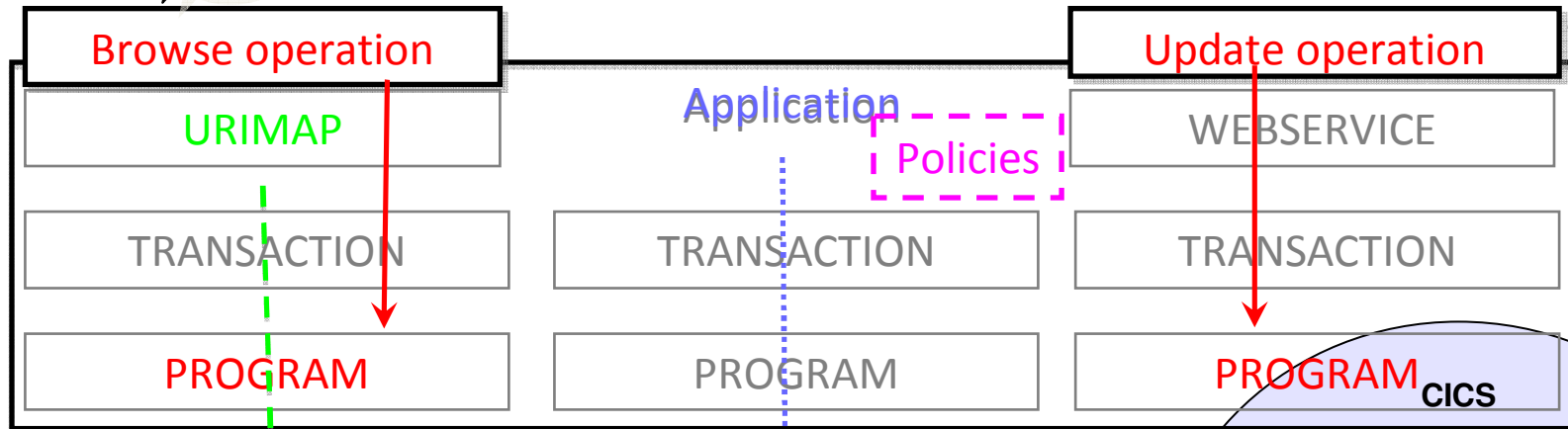
Application Lifecycle

INSTALL
ENABLE
DISABLE
DISCARD

Platform Lifecycle

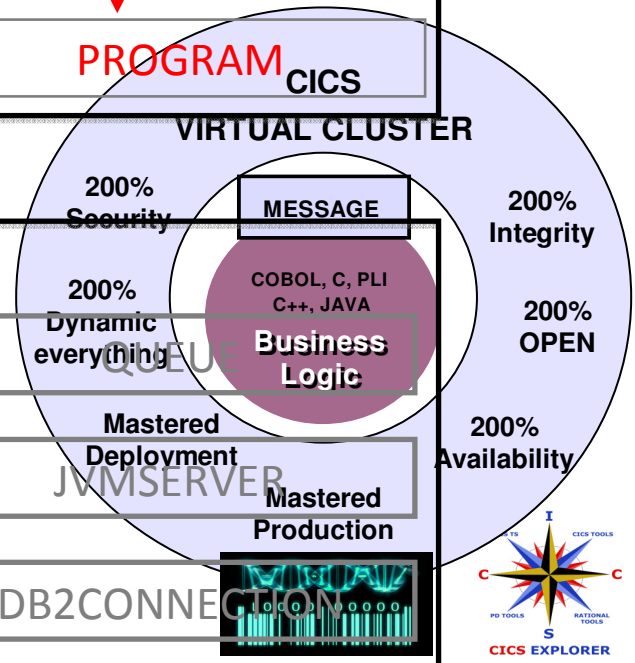
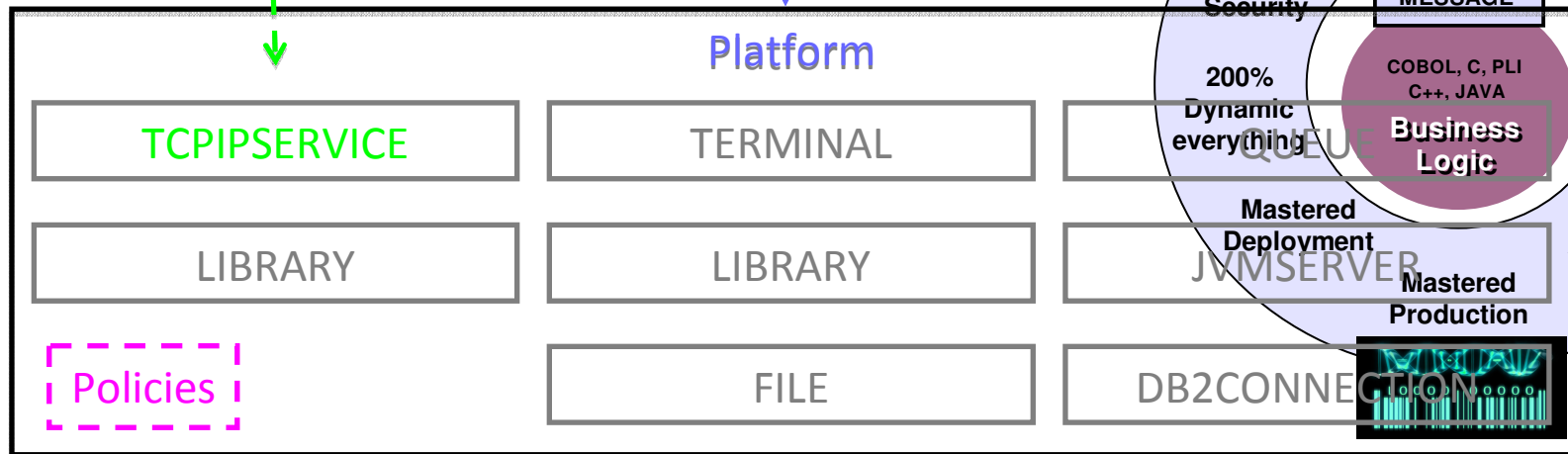
INSTALL
ACTIVATE/DEACTIVATE
ENABLE/DISABLE
DISCARD

Application & Platform



Dependancy

Binding





redlib Cloud...ification

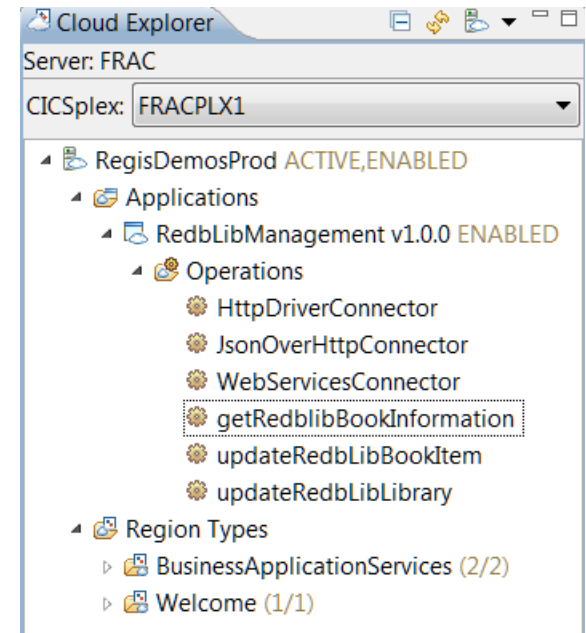
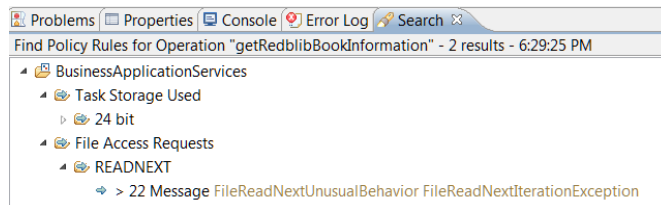


RedbLibManagement application exposes operation entry points:

- Connector type identification, classified as
 - JsonOverHttp
 - Web Services
 - http native
- RedbLib operation identification, classified as
 - Information only
 - Policy rule on unusual number of books in the library
 - Book item update
 - Library file update

RegisDemosProd platform made of :

- The Welcome set of 1 regions
- The BusinessApplicationServices set of 2 regions
 - Policy rule on 24bits usage detection



Platform/Application/Version context in SMF110 or real time query context

Tasks associated with task "0004118" in region "CICSFRAT" - 3 results - 12:40:15 PM

Tasks	Trans ID	Appl ID	Start Time	Run Status	Current Suspen...	Suspend Reason	Application Na...	Application Op...	Application Pla...
0004117	CWXN	CICSFRAT	2013-04-05T10:...						
0004118	CWBA	CICSFRAT	2013-04-05T10:...	SUSPENDED	0000:03:11	IRLINK	RedbLibManag...	JsonOverHttpCo...	RegisDemosProd
0000781	CSMI	CICSFRAX	2013-04-05T10:...	SUSPENDED	0000:03:08	IRLINK	RedbLibManag...	JsonOverHttpCo...	RegisDemosProd
0004119	CSMI	CICSFRAT	2013-04-05T10:...	SUSPENDED	0000:03:08	FCPSSUSP	RedbLibManag...	JsonOverHttpCo...	RegisDemosProd