Web Services

THE KISS OF LIFE FOR IDMS APPLICATIONS

Thomas Hebert, President, ObjEx, Inc.

ObjEx INC.

Agenda

Options for IDMS Integration

- Replication
- Direct real-time interoperation

Web Services Simply Explained

- Design time
- Run time
- Challenges faced by COBOL and ADSO

Options for IDMS Web Services

- Roll your own
- Use a helper
- Industry standard (WSDL)

Pros and Cons

Build one out

Options for IDMS Integration

The problem with IDMS to Relational Replication

- Made sense years ago
  - IDMS did not have many relational tools (better now)
  - The entire industry standardized on relational
    - Oracle, DB/2 and SQL Server
  - Replication made IDMS database content available to a broad set of development tools
- Not so much now
  - Requires two copies
  - Continuous monitoring to maintain sync
  - Coordinated schema changes
  - Careful management of diverging schemas
    - e.g. moving data from IDMS schema into and out of the schema for a COTS CRM system
  - Relational databases are fast becoming a “legacy” technology
    - “I know…let’s take our 40 year-old database and put it into our 30 year-old database!”

Options for IDMS Integration

Direct access to the data makes sense

Direct access to the investments in existing business rules makes even more sense

Real-time integration is very attractive and organization have tried many, many techniques

Custom techniques do nothing to enable access to todays' leading development tools and technologies

Industry standard Web Services make more sense than ever

- e.g. Salesforce, CRM, mapping systems, vendor integration

Web Services Explained

Applications need to service to each other:

- Built on different hardware, operating systems, languages, architectures
- Built by different people, at different times
- A contract is needed that describes what a client sends and what the server might send back

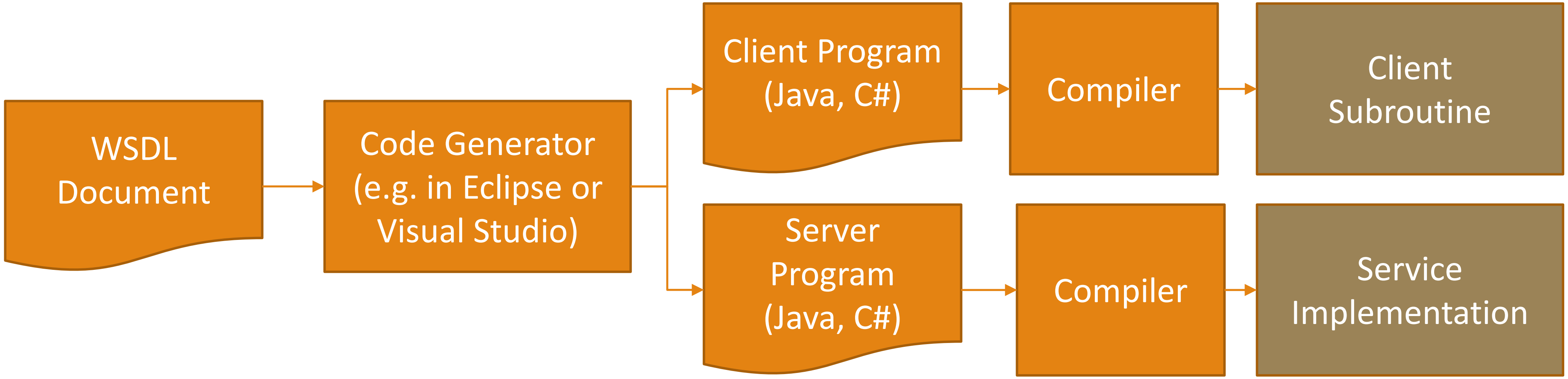
Enter Web Service Description Language (WSDL)

- Describes the methods that can be called
- Includes XML Schema(s) used to describe the request and response content
- Defines endpoints (where services can be accessed)

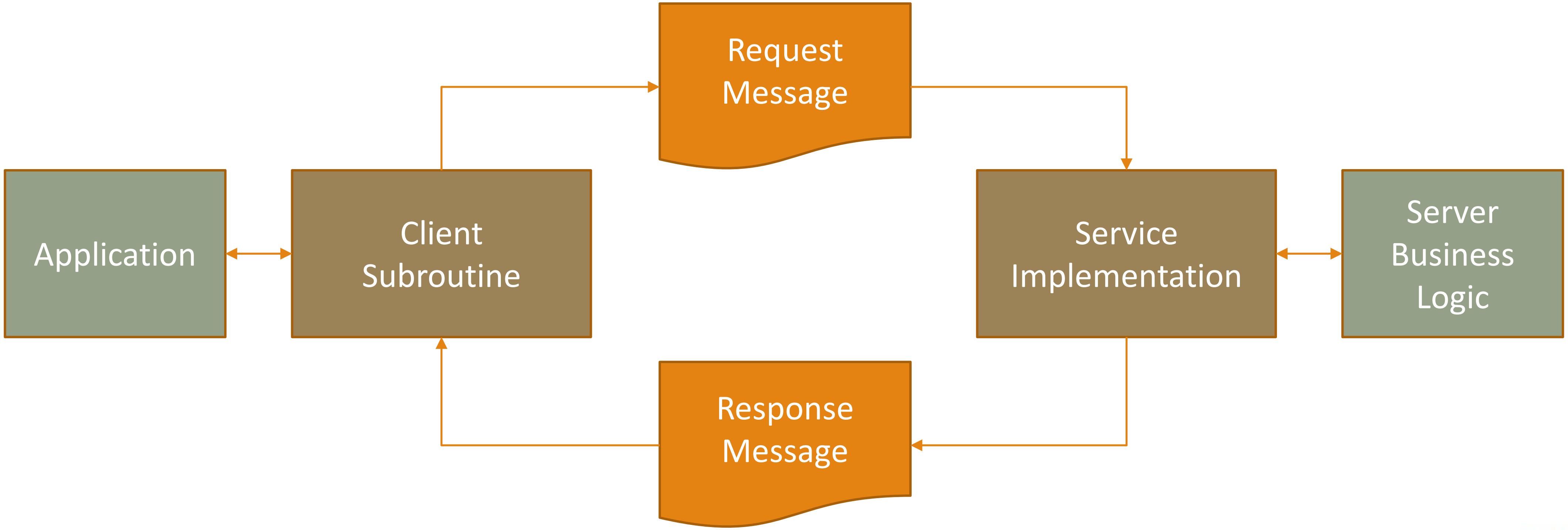
Standards defined by the World Wide Web Consortium (W3C)

- http://www.w3.org/standards/webofservices/

Web Services Design Time


WSDL Document
Code Generator (e.g. in Eclipse or Visual Studio)
Client Program (Java, C#)
Compiler
Client Subroutine
Server Program (Java, C#)
Compiler
Service Implementation

Web Services Run Time



Options for IDMS Web Services

Roll your own

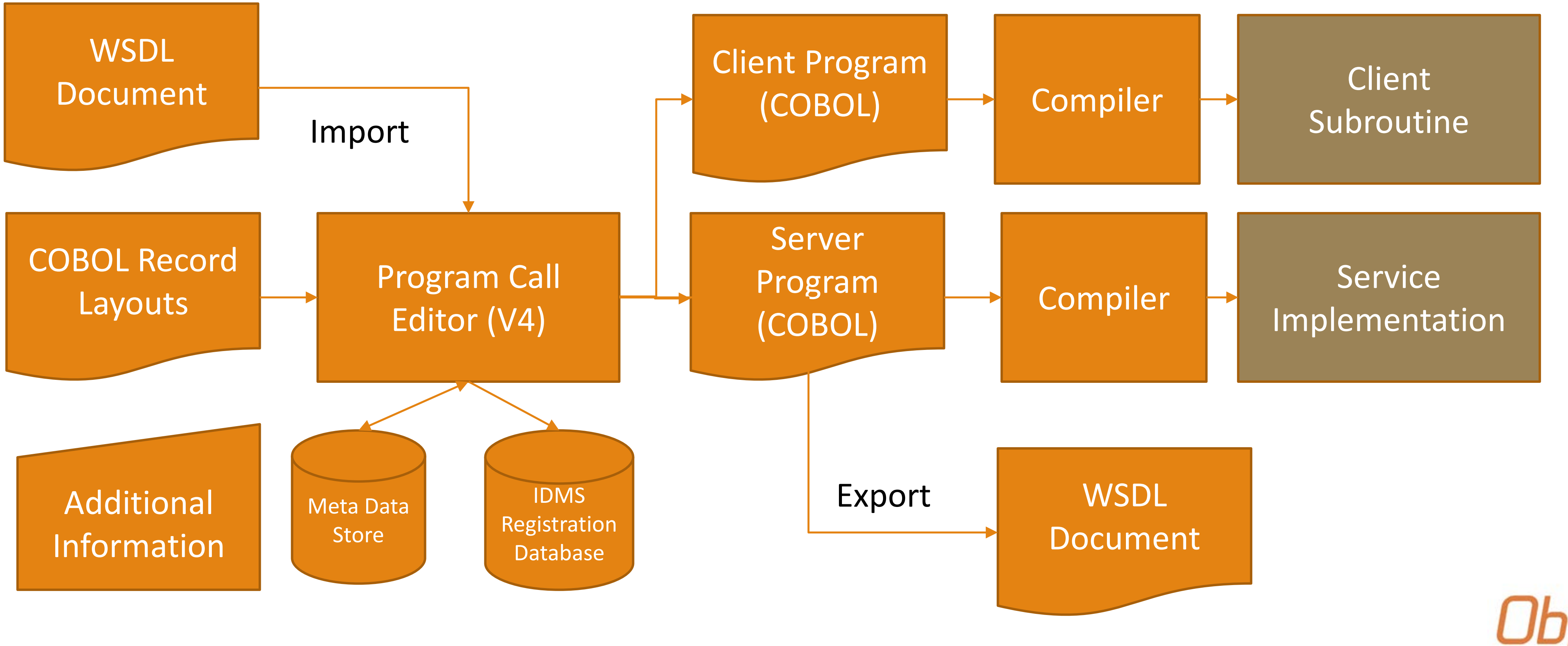
- Read the WSDL, or write one up by hand
- Build your request document in your application program
  - Create XML from your data
  - Wrap with SOAP Headers, Body and schema namespace declaration
  - Create and manage a low-level socket
  - Write the request
  - Read the reply
  - Parse the reply
  - Handle errors
  - Handle SOAP Exceptions
  - Copy the XML content to your COBOL or ADSO record elements
  - Handle conversions from string to numeric
  - Handle ASCII to EBCDIC conversions to and from

Options for IDMS Web Services

Use a helper (CA’s current enhancements)

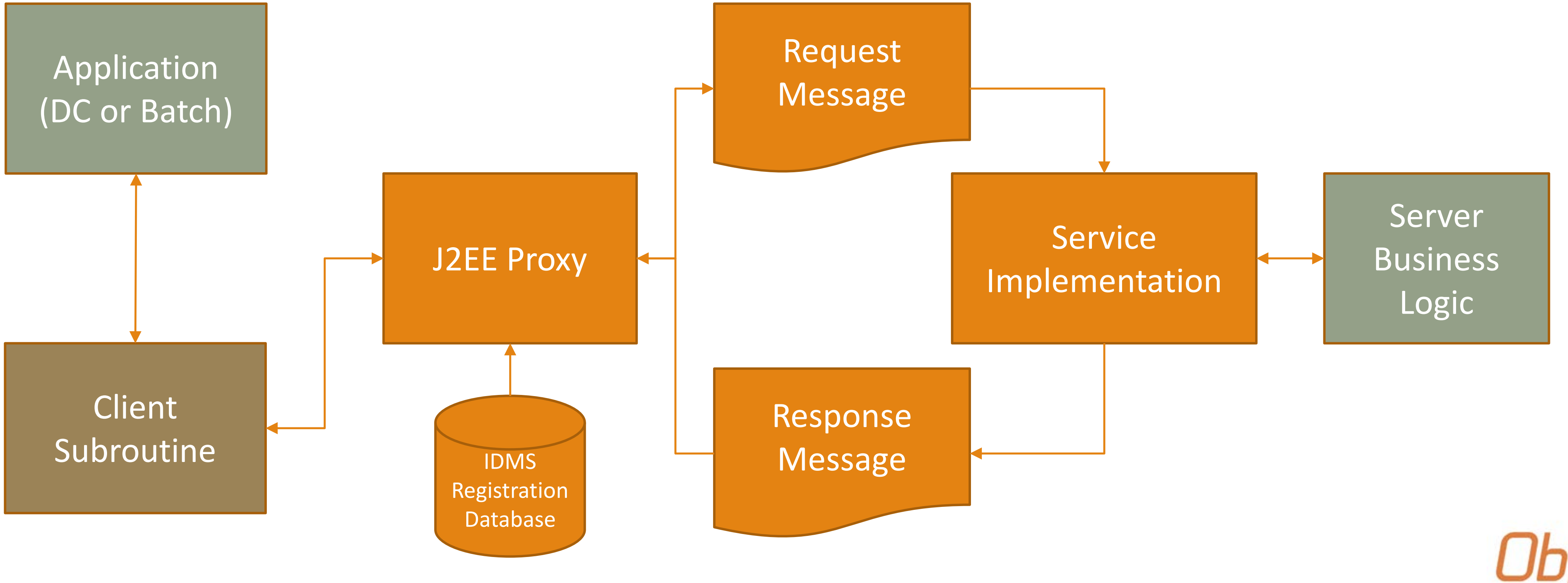
- No WSDL generation or consumption
- CA helper routine to call the web service
- CA helper routine to check for SOAP fault
- CA helper routine to extract the SOAP message and then manually move to work record
- XML to COBOL element marshalling is still up to you
- Manual cleanup and data checks, including freeing storage allocated on your behalf

ObjEx Program Call Design Time



ObjEx Program Call Run Time

Outbound from mainframe:



Application (DC or Batch)

Client Subroutine

J2EE Proxy

IDMS Registration Database

Request Message

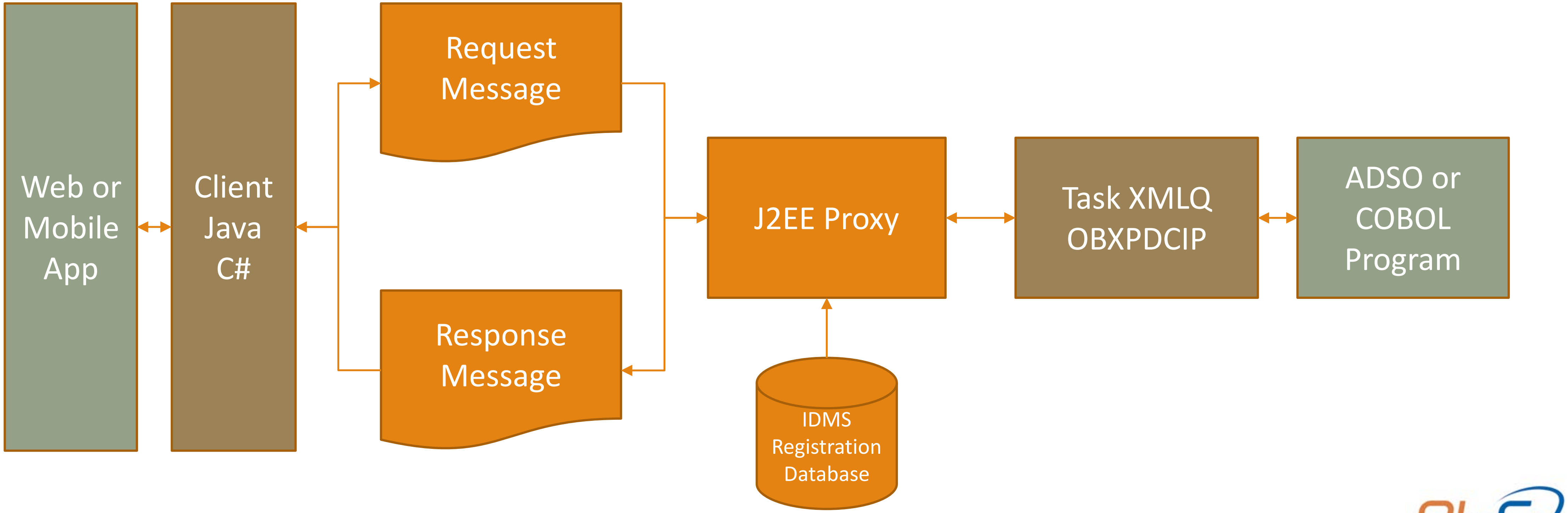
Response Message

Service Implementation

Server Business Logic

ObjEx Program Call Run Time

Inbound to mainframe:



Build One Right Now

Create a service

Import some copybooks

Register the program call with IDMS

Review the service

Review the code

Make the call

Why do this?

Time

- Time-to-market
  - Faster time to market with vendor and customer offerings
  - Easier integration with cloud solutions and virtually all up-to-date applications
  - It just seems like a bad idea to wait until we replace IDMS to get the features and functions we need in today's technology environment
- Less time coordinating layouts
- Consistent reliable content mapping
  - XML Element to COBOL Element
  - COBOL Element to XML Element
- Standards

Why do this?

Risk

- Roll your own and risk being left behind
- You will never keep up or you will incur great expense

Money

- Using WebSphere MQ from IDMS
  - Mainframe can be as high as $1 MM US
  - Run it on a Windows Server it’s more like $7-$30k
  - Call a web service and store the queues off mainframe
  - Better yet:
    - use Azure Service Bus, or Java

What’s next?

Web Services are the way forward for IDMS users

- Now – with W3C web services
- Next for us
  - JSON Proxy
  - More code generation options

For IDMS applications

- We have the data
- We have the rules
- But they aren’t accessible
- 30 years ago IDMS took a hit with the advent of relational databases. Now we can finally hit back with web services

It’s time to move forward

A connected application is a happy application!