

Impact2010

The Premier Conference for Business and IT Leaders



Fabio Riva

Mainframes e Web 2.0



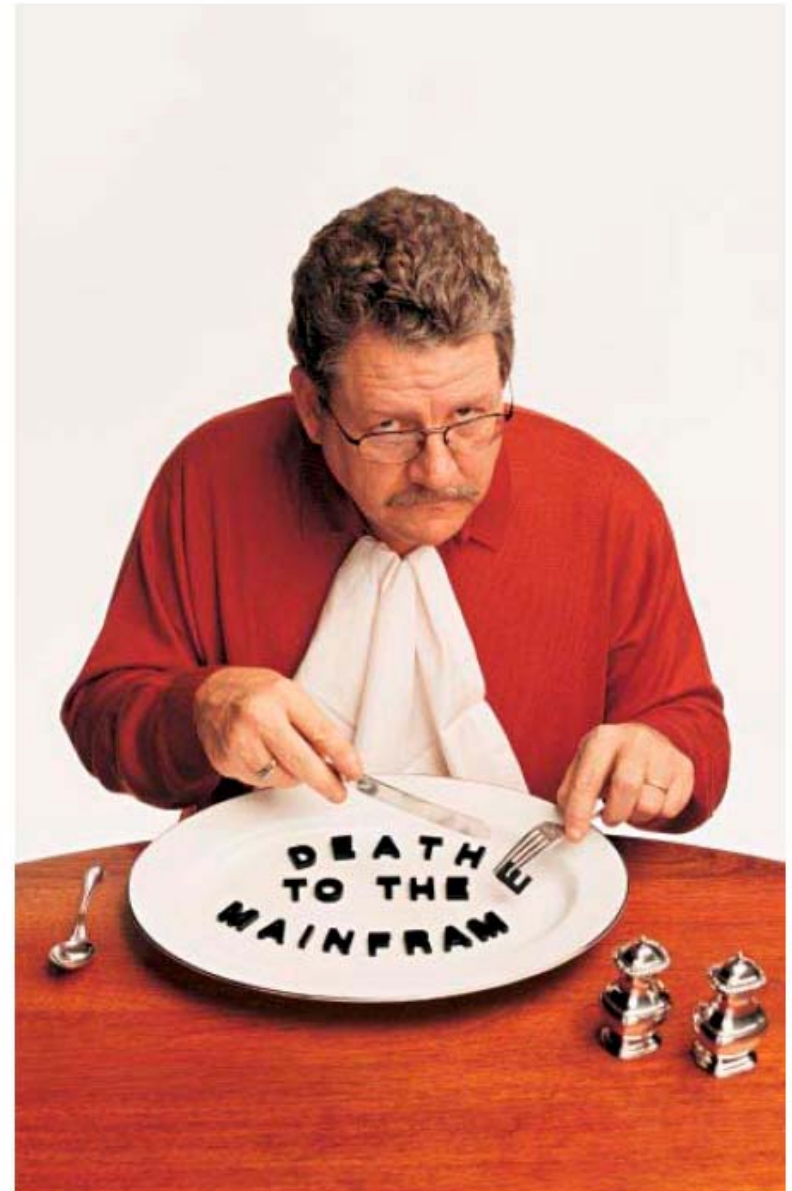
Mainframe evolution



Reports of the death of the mainframe were premature

- **“I predict that the last mainframe will be unplugged on March 15, 1996.”**
 - Stewart Alsop, March 1991

- **“It’s clear that corporate customers still like to have centrally controlled, very predictable, reliable computing systems – exactly the kind of systems that IBM specializes in.”**
 - Stewart Alsop, February 2002



Impact2010



It all started with the IBM 701

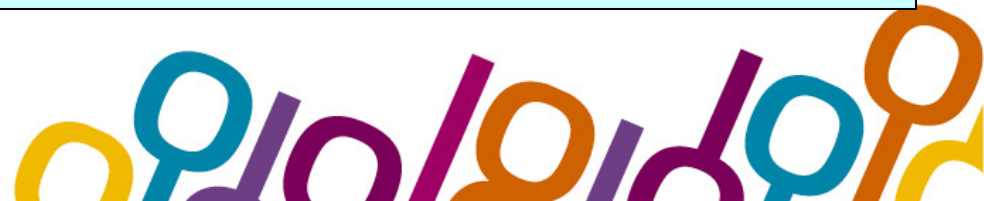
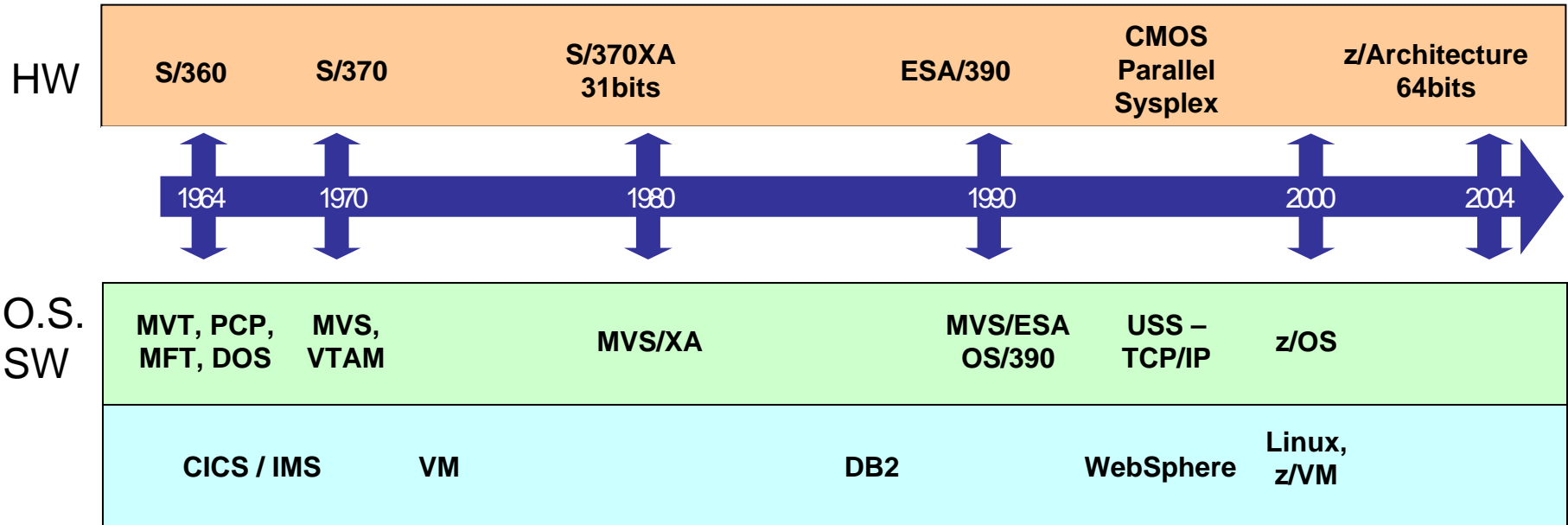
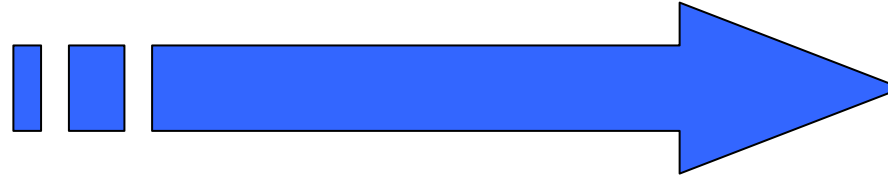
- Circa 1951:



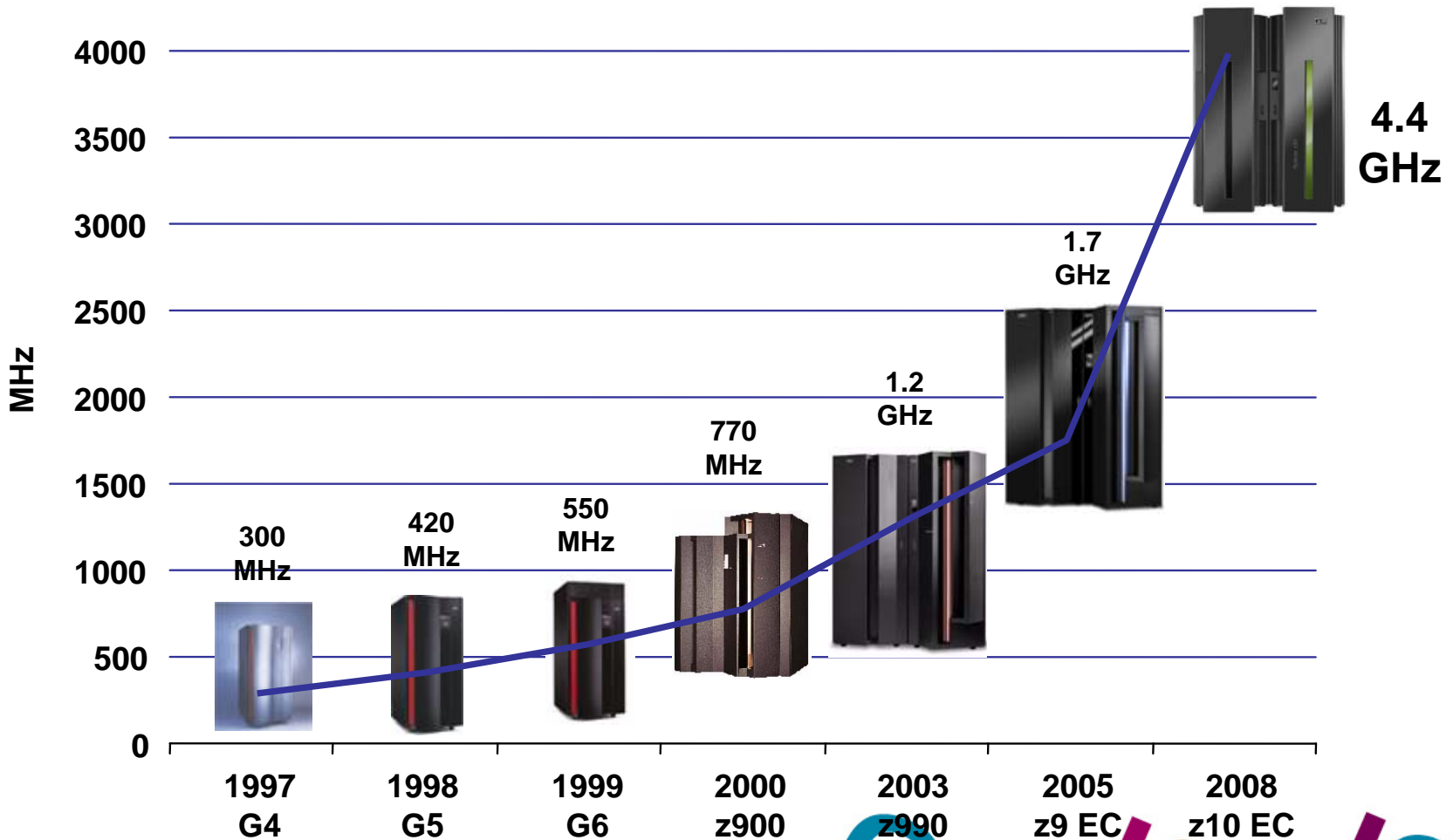
- And proceeded through the 700, 1400, S/360 & 370, 303x, 308x, 4300, S/390 Parallel Server, to today's eServer zSeries



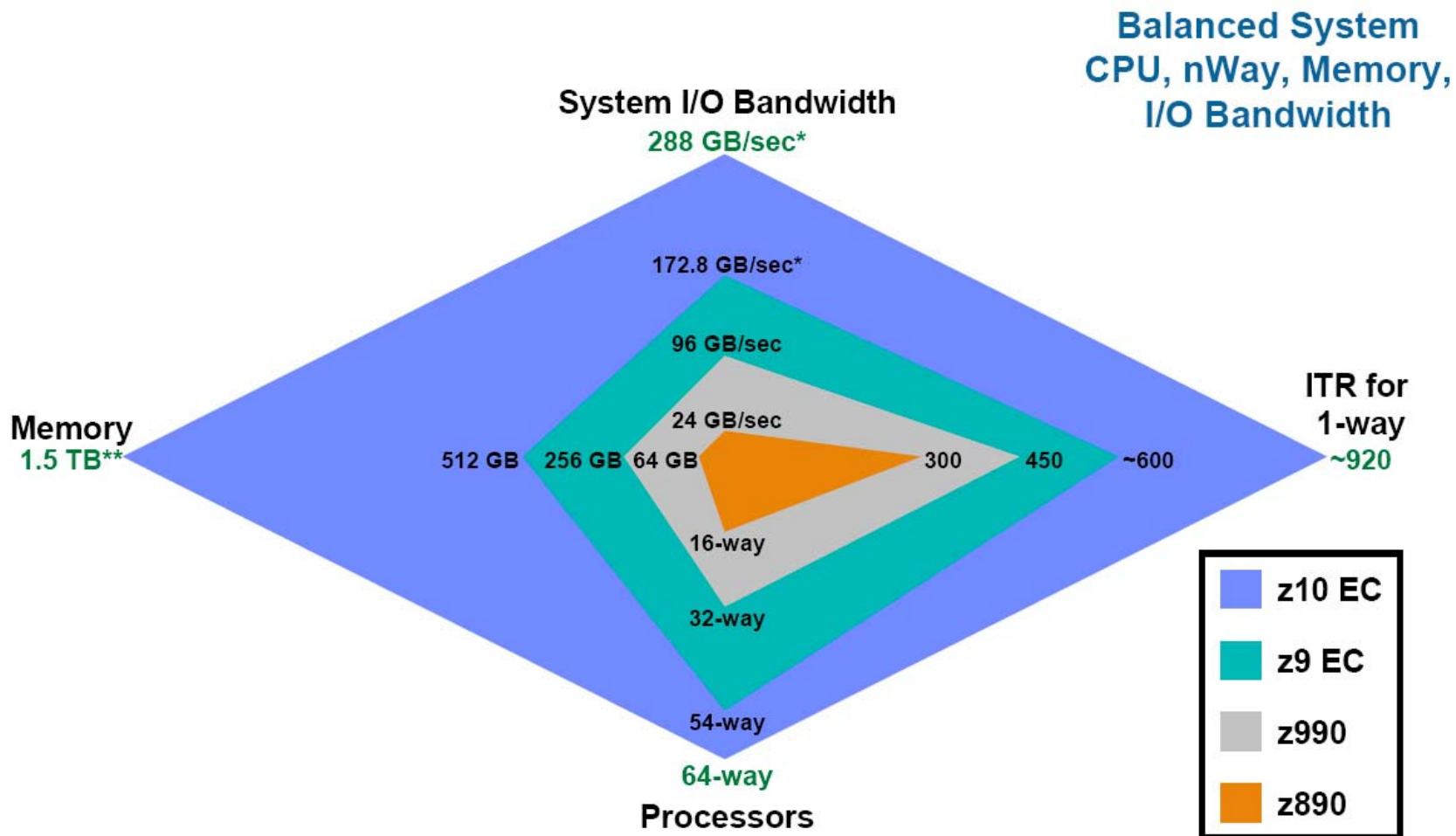
Hardware – Continuous evolution



IBM z/Series machines continues the HW evolution



IBM System z: Balanced Systems Design



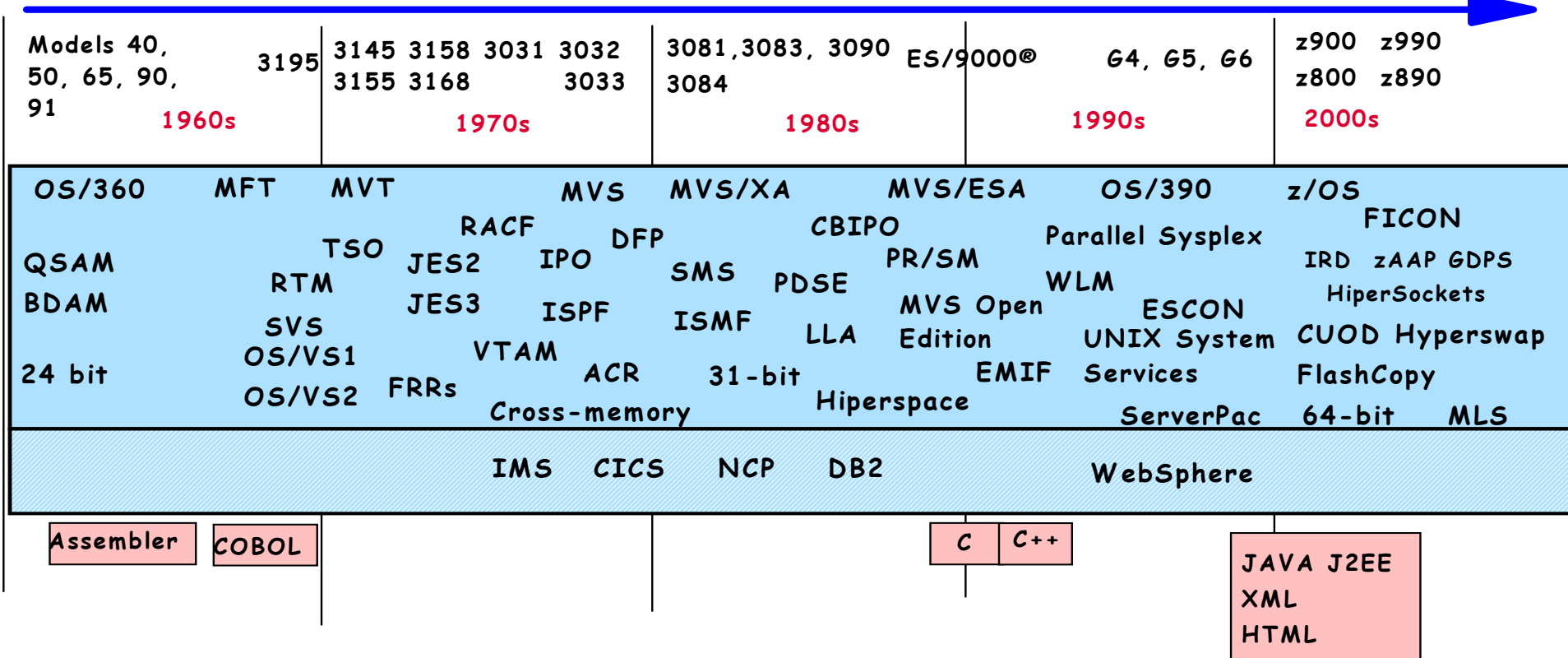
Software – Continuous innovation

S/360

S/370™

S/390®

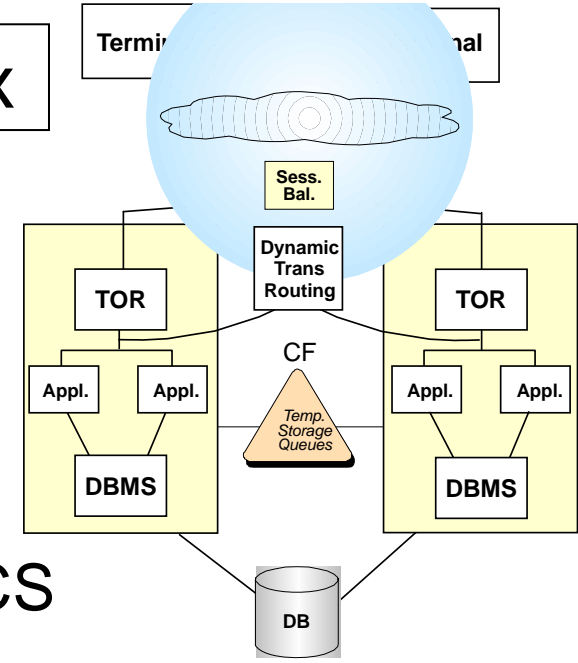
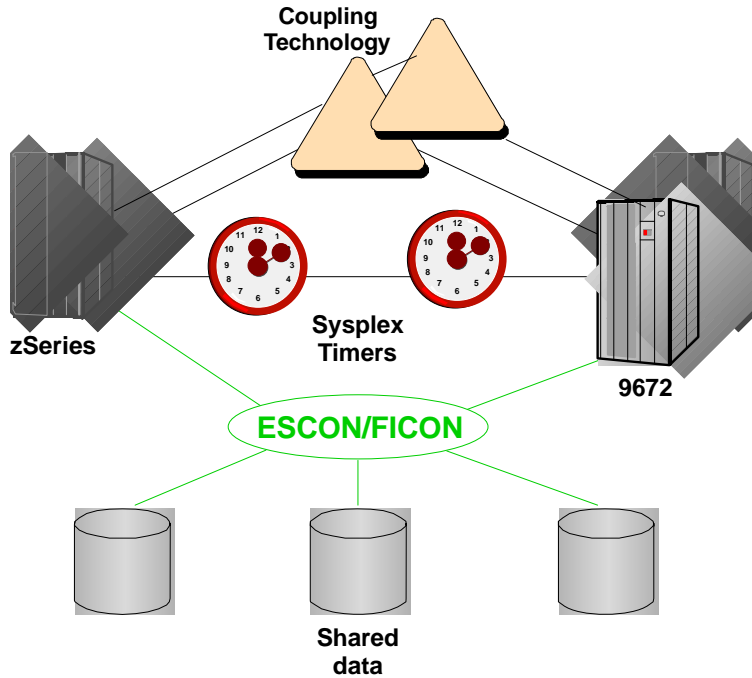
zSeries,
System z



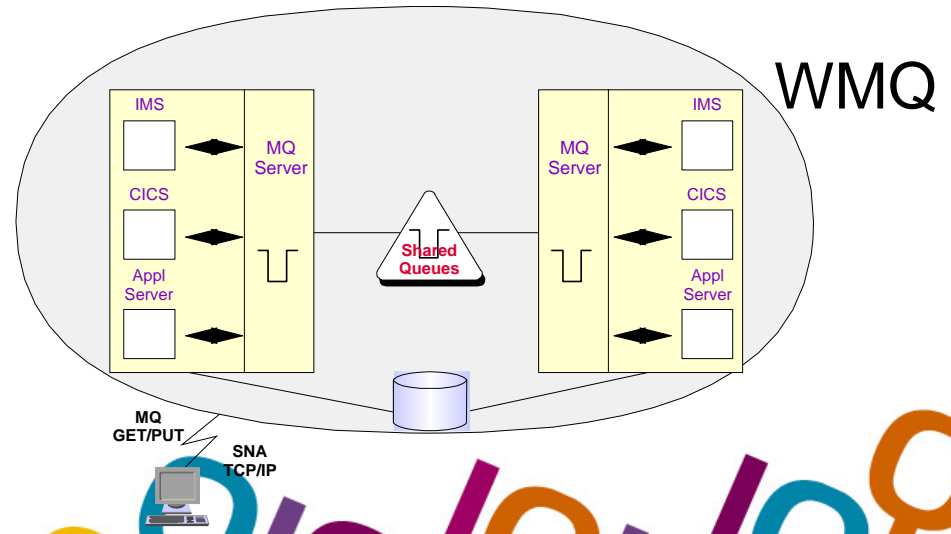
Application investment protection



Mainframe functions: Parallel Sysplex



CICS



WMQ

- Continuous Availability
- Flexible Growth
- Scalability
- Reduced Cost
- Leverage S/390 Investment



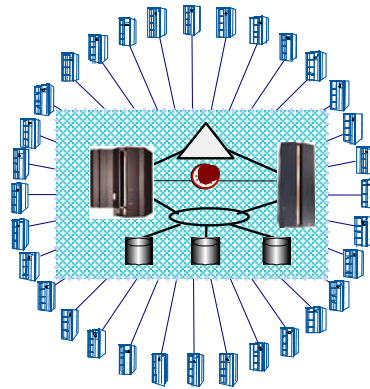
Mainframe functions: z/OS Continuous Availability

Single System



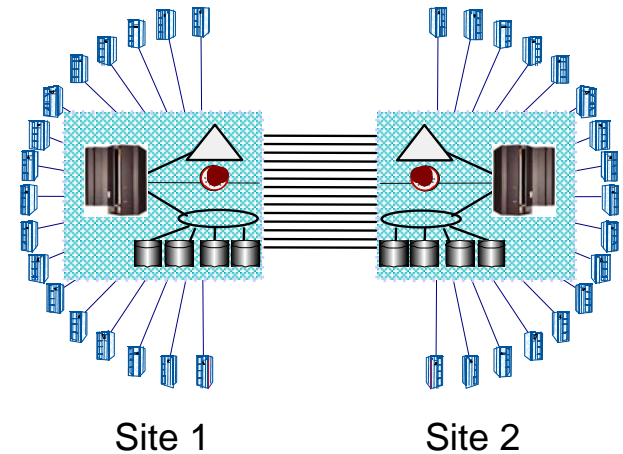
- Built In Redundancy
- Capacity Upgrade on Demand
- Capacity Backup
- Hot Pluggable I/O

Parallel Sysplex



- Addresses Planned/Unplanned HW/SW Outages
- Flexible, Nondisruptive Growth
 - ▶ Capacity beyond largest CEC
 - ▶ Scales better than SMPs
- Dynamic Workload/Resource Management

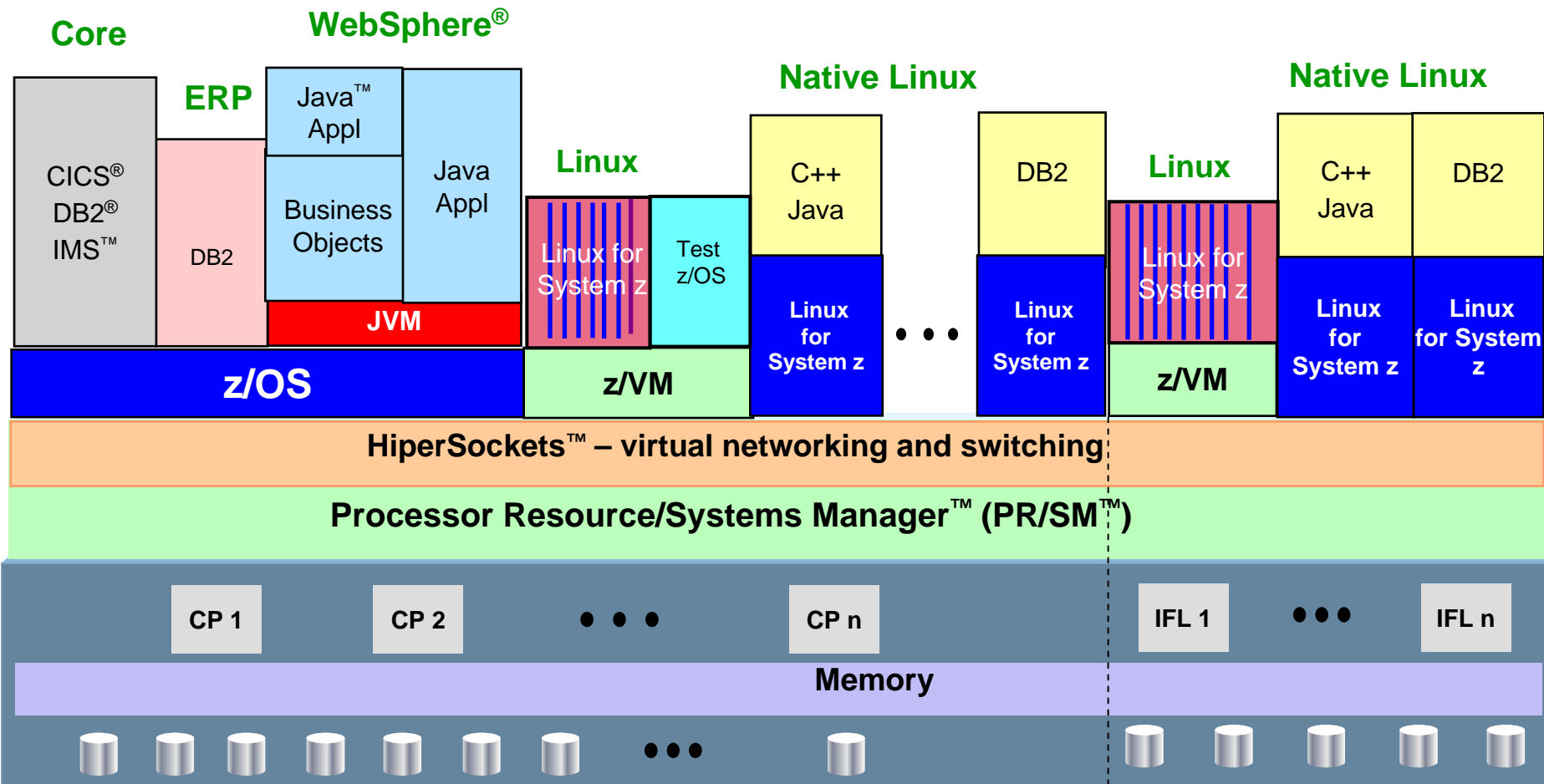
GDPS



- Addresses Site Failure/Maintenance
- Sync/Async Data Mirroring
 - ▶ Eliminates Tape/Disk SPOF
 - ▶ No/Some Data Loss
- Application Independent



Mainframe functions: Virtualization



Mainframes ...and Web 2.0



Easy access to mainframe services



Easy access to services

- **Web 2.0 applications should be:**
- *Simple to access* because the services are accessed via a simple **Web API interface (typically using REST with AJAX capable browsers)**
- *Simple to use* as the same services are exposed on the enterprise with the use of **URLs and feeds (generally ATOM feeds)**
- Web 2.0 is nothing like a rocket-science. It is simply a very intuitive, fast and customizable user interface, joined to high performance (yet easy to use) software products and tools.



Easy access to the mainframe: use of Web 2.0

Simple to use



AJAX

- Highly Interactive
- Browser invoked services

Technology

Community

Economic

JSON / XML / ATOM

- Information exchange
- JavaScript Friendly

Simple to access



REST

- Easily invoked
- HTTP-Centric Patterns

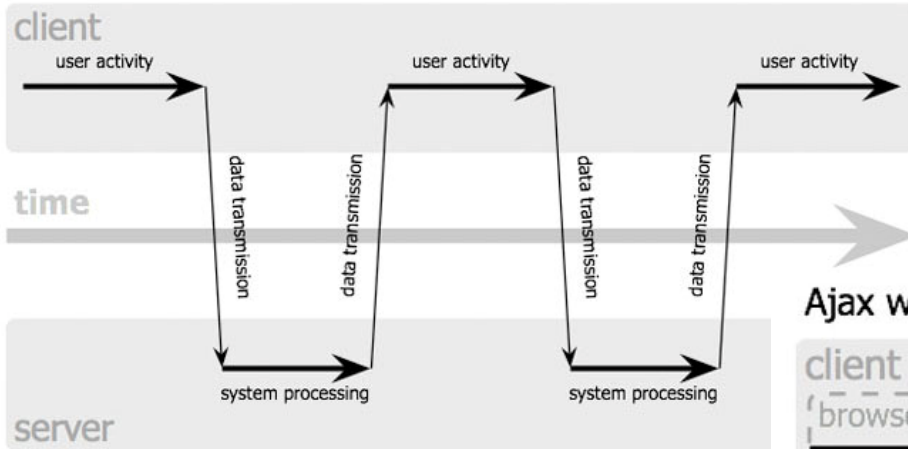
These are some technologies used in Web 2.0 apps

- **AJAX:** This technology allows to access data in asynchronous way reducing user wait times. AJAX is not really new, but it's a combination of existing technologies: XMLHttpRequest, Javascript, CSS and XML. AJAX is asynchronous. It allows updating only parts of a web page so you no longer need to refresh the whole page when you want the user to interact with the server.
- **REST** makes easier to access data on a server, as it uses a set of well-known, standard operations (like GET, POST, PUT, DELETE). Using them, it is possible to interact with a piece of data, called resource. An URI (Uniform Resource Identifier is a string of characters used to identify a name or a resource on the Internet) identifies the resource, and you can get its value with GET operation, update it with POST, etc.
- **ATOM:** The Atom Syndication Format is an XML language used for web feeds. It's an evolution of RSS web feed format, and it's more structured and it was developed to solve some RSS limitations

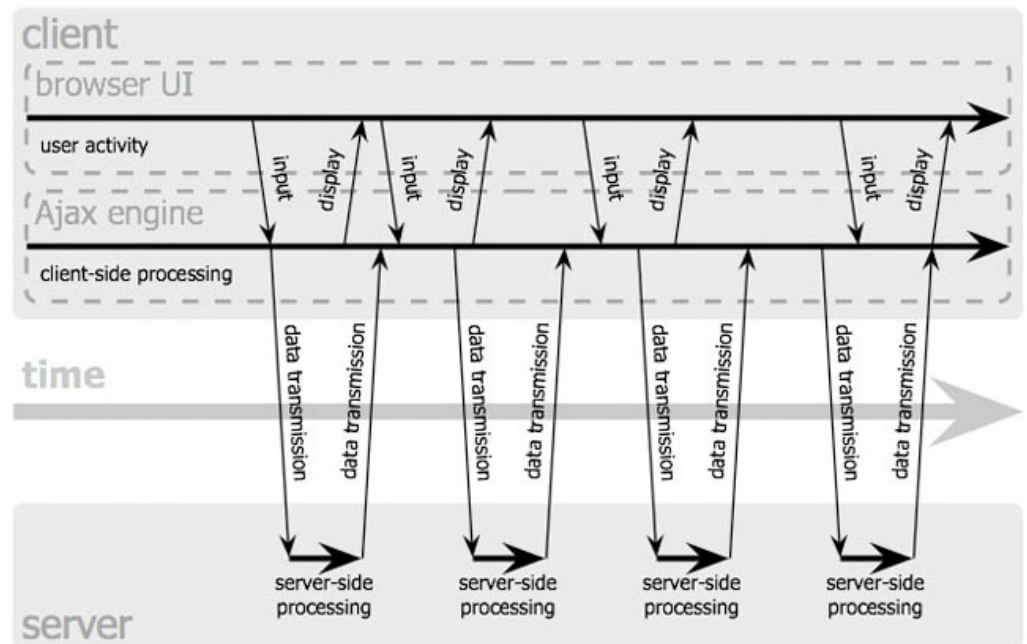


Asynchronous Javascript And Xml (AJAX)

classic web application model (synchronous)



Ajax web application model (asynchronous)



REpresentational State Transfer (REST)

A RESTful Web service is formed like a sentence – it simplifies how developers access services

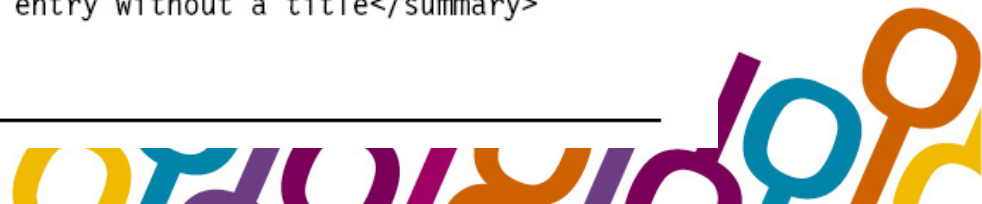
- **Verb** = HTTP Action (GET, POST, PUT, DELETE)
- **Noun** = the URI of the Service (the document)
- **Adjective** = MIME type of the resulting document



ATOM feed example

Example 7-21 Atom feed example

```
<feed xmlns="http://www.w3.org/2005/Atom"
      xml:lang="en"
      xml:base="http://www.example.org">
  <id>http://www.example.org/myfeed</id>
  <title>A Sample Feed</title>
  <updated>2008-06-16T18:34:42Z</updated>
  <link href="/blog" />
  <link rel="self" href="/myfeed" />
  <entry>
    <id>http://www.example.org/entries/1</id>
    <title>A sample blog entry</title>
    <link href="/blog/2008/06/1" />
    <updated>2008-06-16T18:34:42Z</updated>
    <summary>This is a sample blog entry</summary>
  </entry>
  <entry>
    <id>http://www.example.org/entries/2</id>
    <title />
    <link href="/blog/2008/06/2" />
    <updated>2008-06-16T18:34:42Z</updated>
    <summary>This is sample blog entry without a title</summary>
  </entry>
</feed>
```

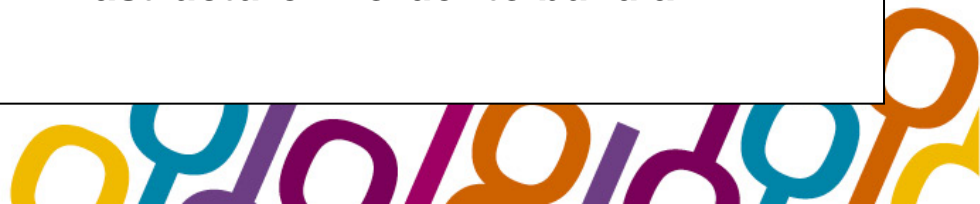


How to use it

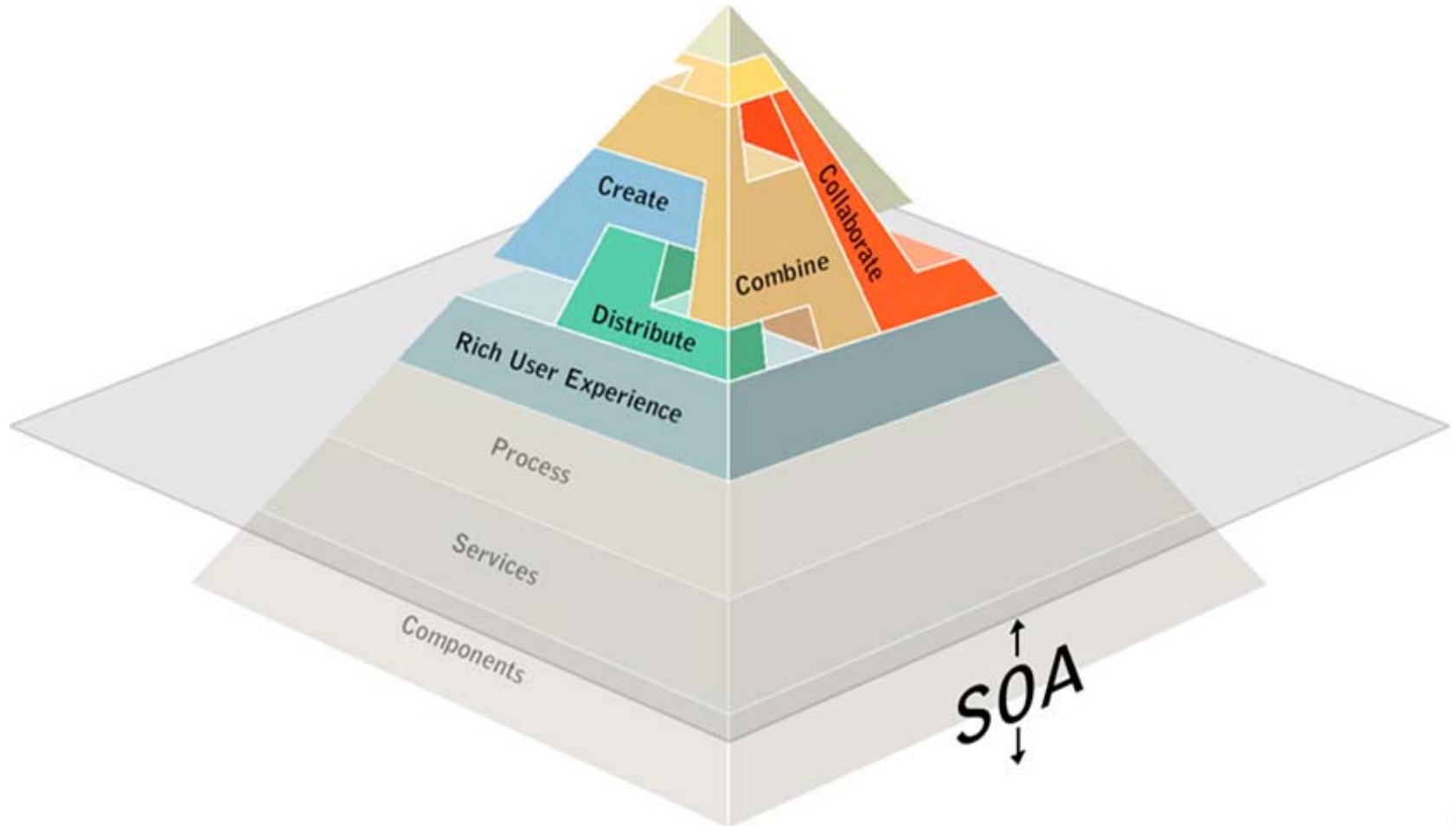


Reuse of existing services: RESTful SOA

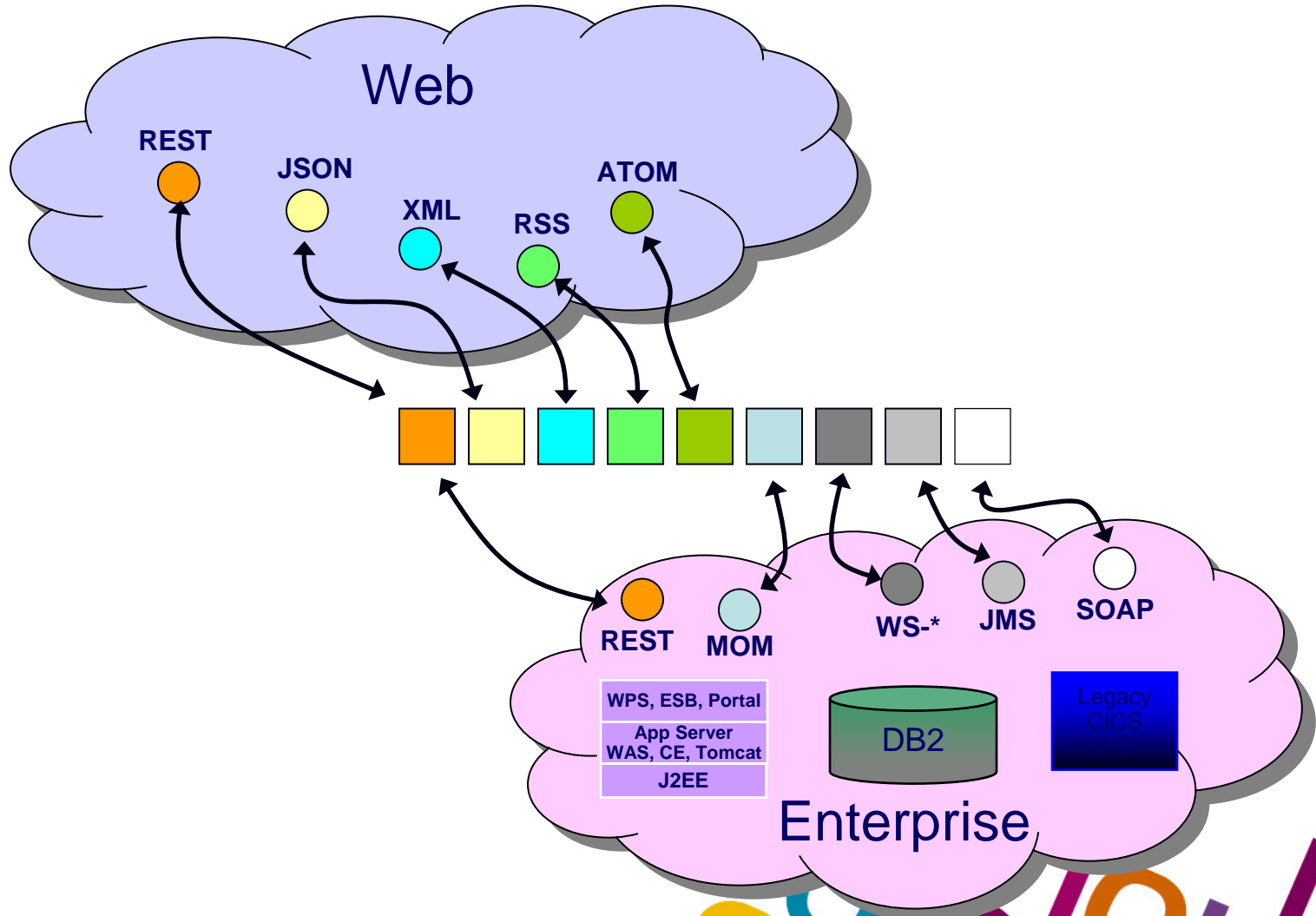
- **Using the Web 2.0 technology described before, it's possible to expose existing services at enterprise level. More and more IBM products allowed user to access data via REST protocol. REST has become perhaps the single most widely deployed form of service orientation because of its simplicity.**
- **The join between lightweight Web 2.0 technologies with robust SOA infrastructure is called RESTful SOA.**
- **Key aspects of building an effective RESTful SOA are:**
 - Take advantage of your existing infrastructure wherever possible
 - Use well-established, ubiquitous technologies for scalability, performance and security
 - Build rich UI's that run in any commodity browser
 - Make content simple and human readable
- **In the following picture this concept can be better explained. Web 2.0 is nice but should be based on a robust SOA infrastructure in order to build a powerful IT solution.**



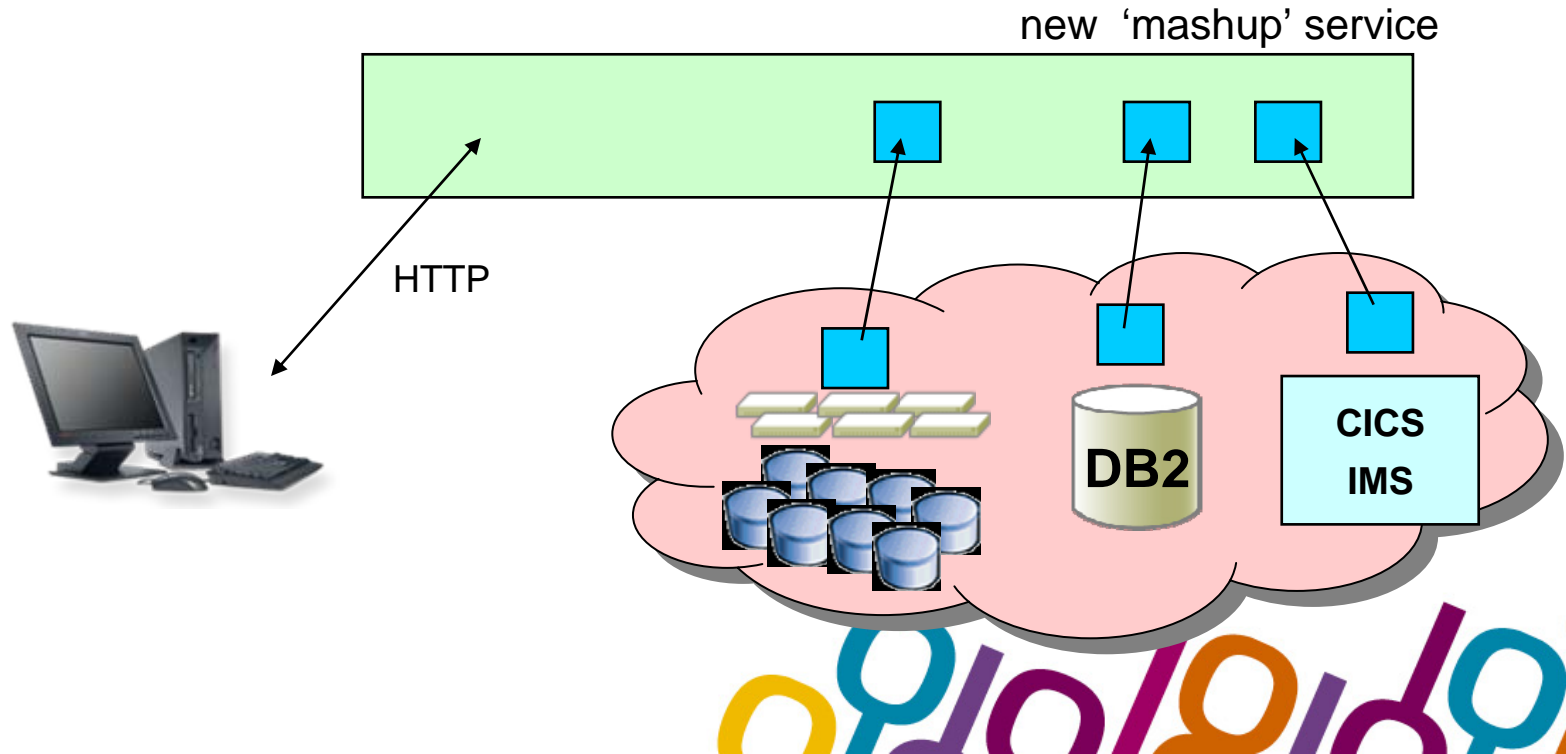
RESTful SOA: SOA and Web 2.0



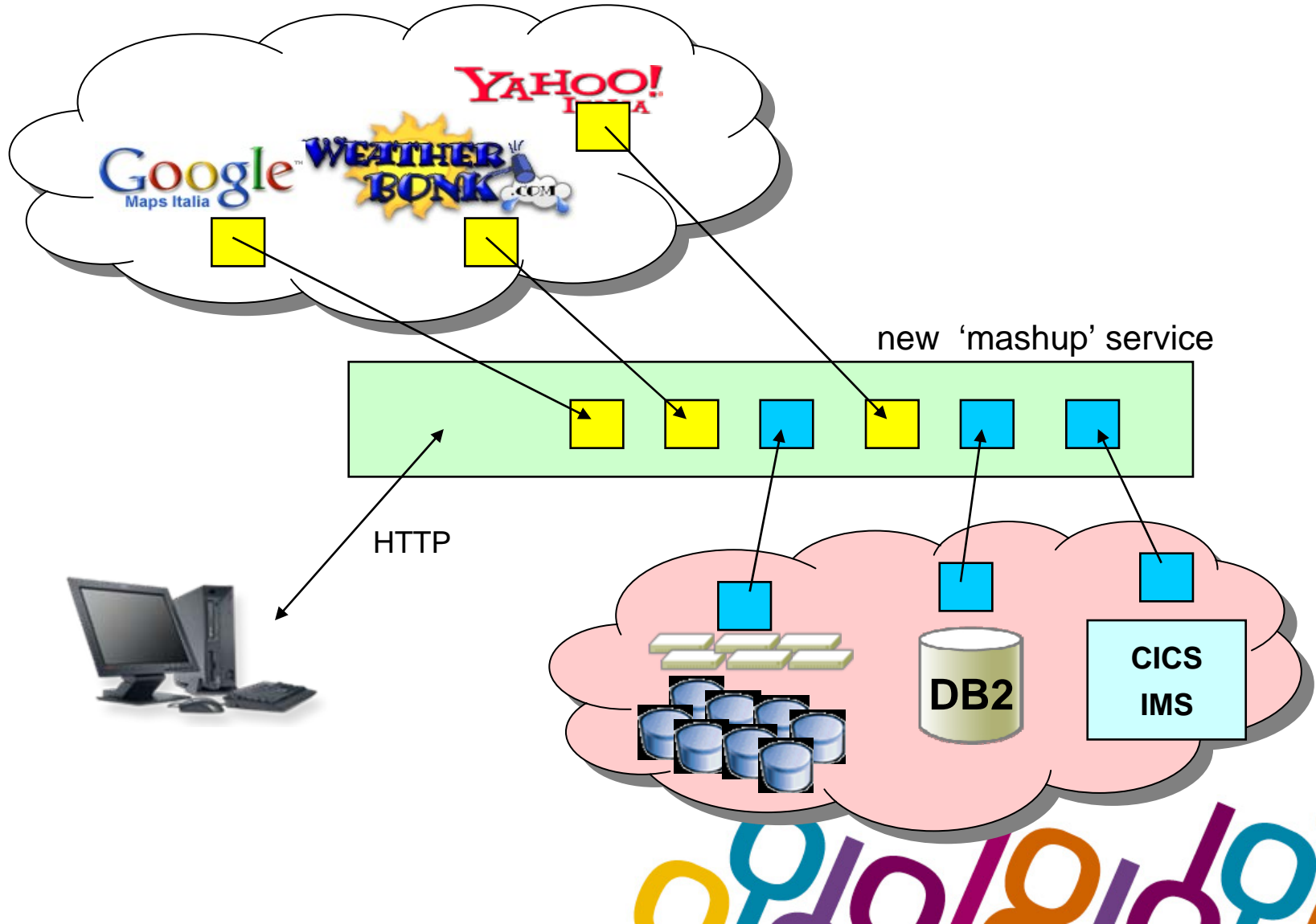
Bridging “RESTful SOA” and “Enterprise SOA”



Mashup of internal services



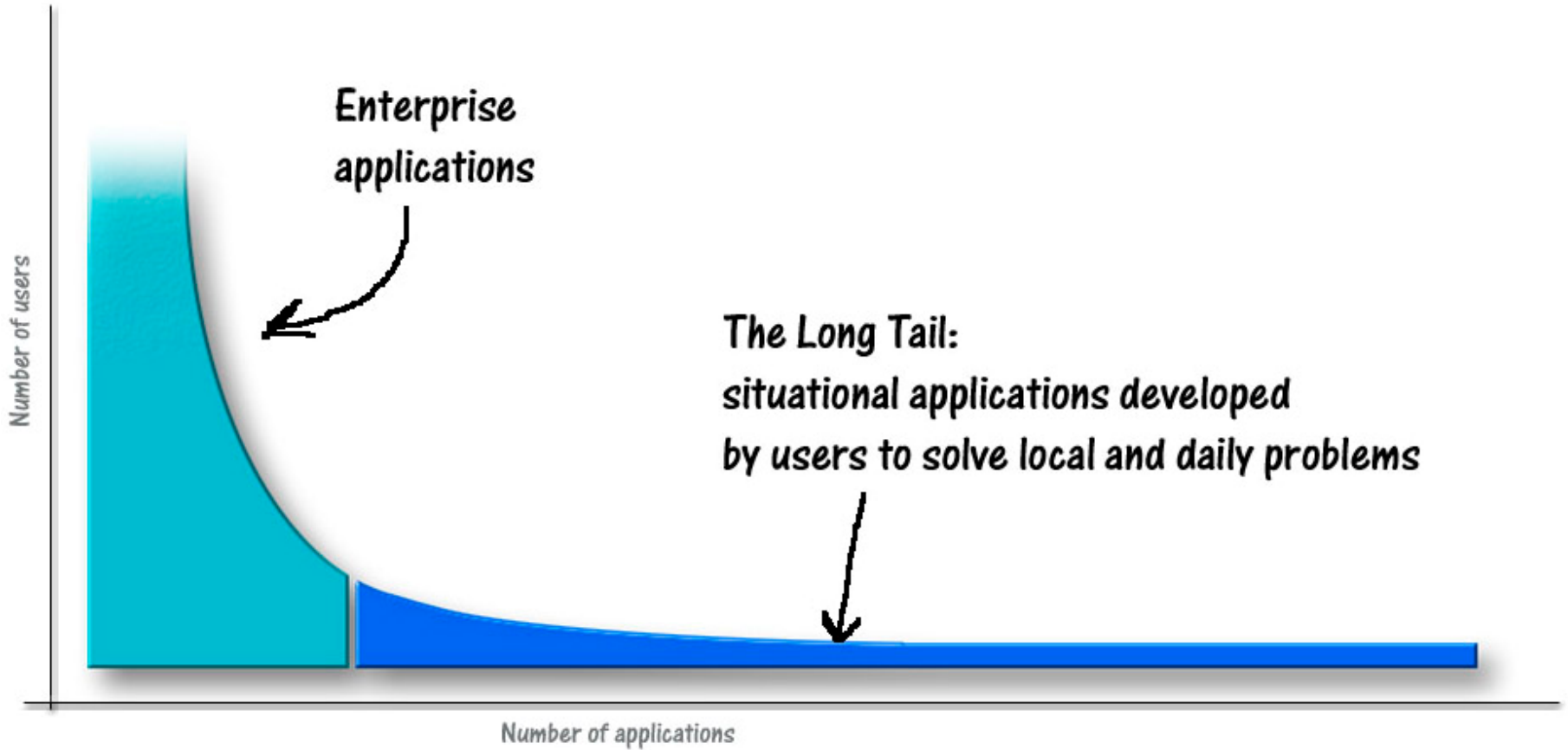
Mashup of internal services and external services



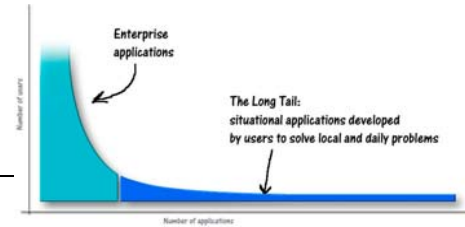
Long tail and situational apps



Long tail applications...



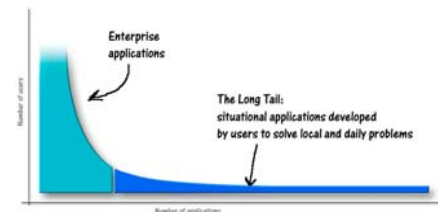
Long tail and situational appls



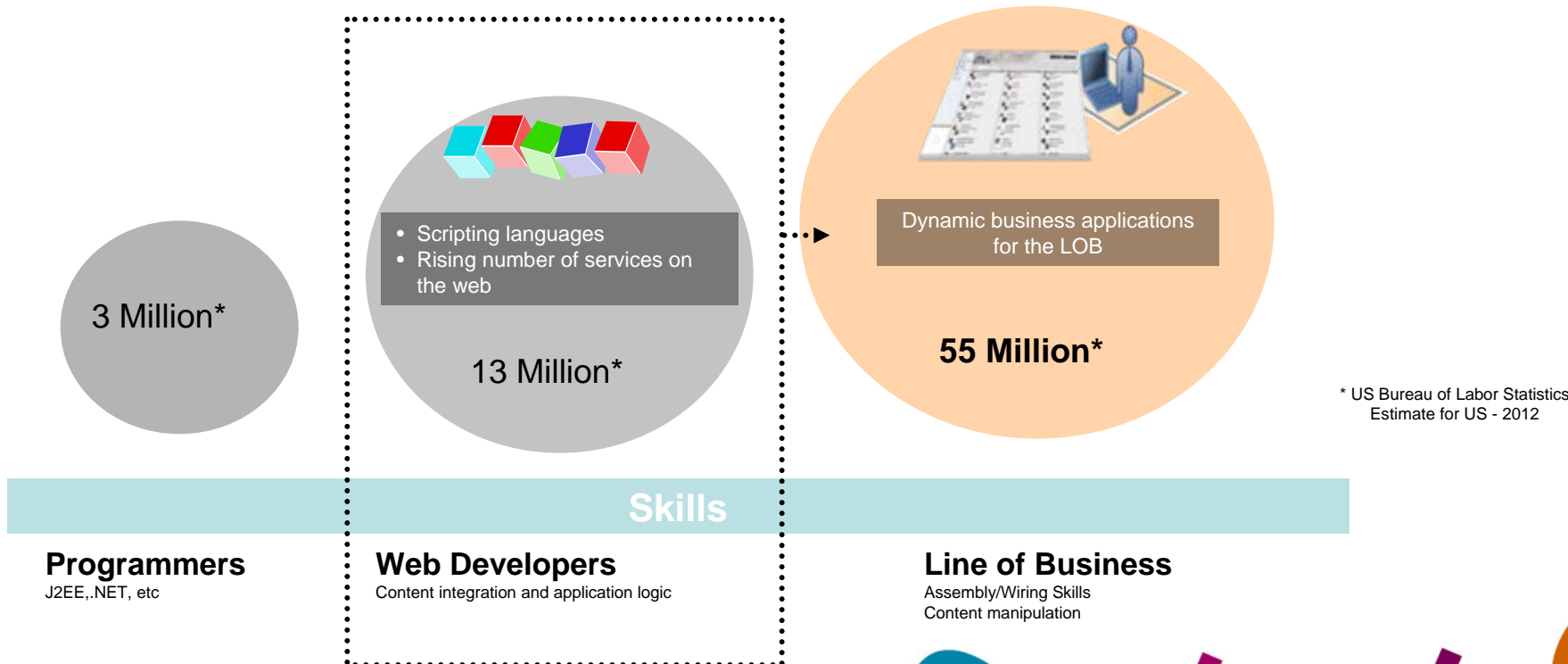
- ‘Long Tail’ was first coined by Chris Anderson in an October 2004 Wired magazine article to describe certain business and economic models such as Amazon.com or Netflix. Chris Anderson said that businesses with distribution power can sell a greater variety of hard to find items at small volumes than of few popular items at large volumes.
- Applied to IT world, this concept can be translated in the following: in addition to mass-market and enterprise applications, there are also a big number of situational and nich applications developed directly by end users. **We should take them into consideration, due to their number.**
- Some of these situational applications are now critical for the company. **They were developed by users with low technical skill but with high knowledge in company business. Generally these applications are mash-up of existing services, combining data from several sources.**



Long tail and situational applications



dal 2012, negli USA si prevede di avere oltre 55 milioni di utenti business che richiederanno applicazioni web-based per risolvere necessità "stuzionali"



Long tail and consolidation of situational apps

- **The strategic direction in this area should be to provide a common platform to develop these kind of applications. This platform should be open and flexible. In order to be sure it will be used by users, this platform should be free and easy to implement.**
- **If we use a common platform to develop and execute situational applications, we'll be able in future to consolidate all of them or convert them on another similar platform, but managed by IT structure for all enterprise. The solution is already in place and it's available free of charge in internet: it's called WebSphere sMash, also known as Project Zero.**
- **Consolidation of sMash applications can be realized either migrating applications to WebSphere Application Server platform (some conversion activities are involved) or directly consolidating sMash apps on z/Series platform, using zLinux (starting from WebSphere sMash V1.1.1).**



WebSphere sMash application page

The screenshot shows a Mozilla Firefox browser window displaying the WebSphere sMash application page. The browser title is "My Applications - App Builder - Mozilla Firefox" and the address bar shows "http://localhost:8070/". The page has a purple header with "My Applications" and a search filter. A left sidebar contains an "Actions" menu with options: "Create new application", "Create from repository", "Open existing application", and "Import application". The main content area lists several applications with their details:

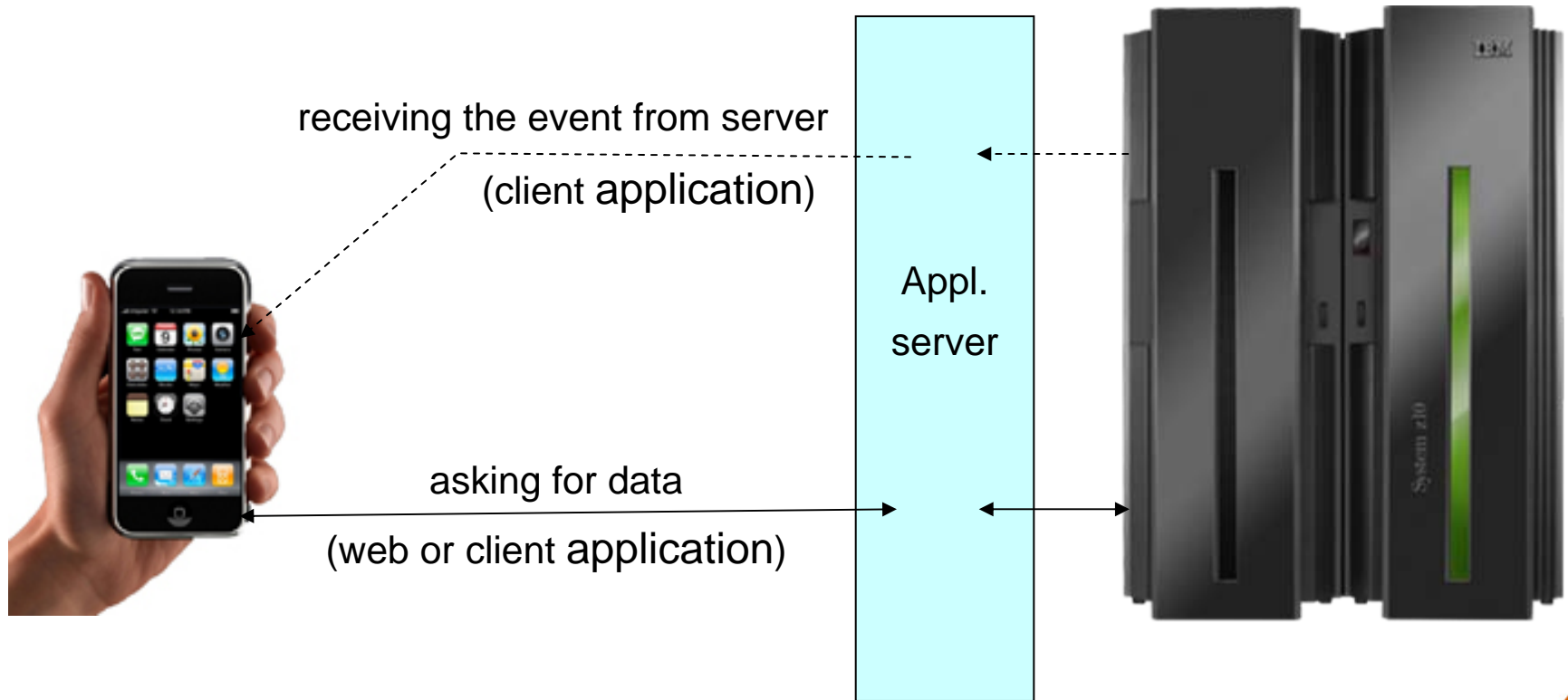
Application Name	URL	Module Group	Tags
CLEAR Amsterdam	http://localhost:8080/ C:\zero\RIVA\	stable	ZeroProject
RIVA demo.iPhone2.demo	http://localhost:8080/ C:\zero\RIVA\	stable	ZeroProject
zero.connection.demo	http://localhost:8080/ C:\zero\RIVA\	stable	ZeroProject
zero.hellodojo.demo	http://localhost:8080/ C:\zero\RIVA\	stable	ZeroProject

At the bottom of the page, there are navigation links for "Project Zero" (Home, Samples), "Help" (Documentation), "Project Zero Forums", and "About" (About, Project Zero). The status bar at the bottom left shows "Completato".

Extention to smartphone devices



Smartphone devices with mainframes



New redbook



New redbook

Rational software

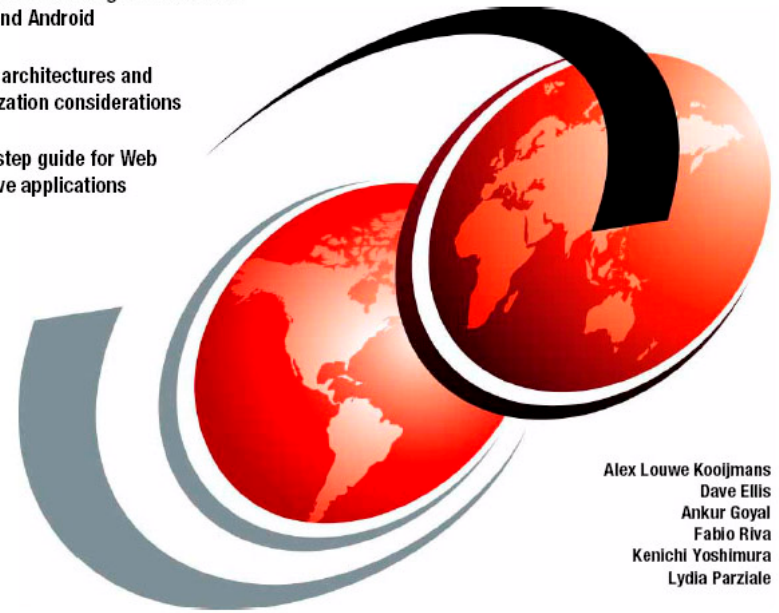
SG24-7836-00

System z on the Go! Access to z/OS from Smartphones

Sample scenarios to get started with
iPhone and Android

Solution architectures and
modernization considerations

Step by step guide for Web
and native applications



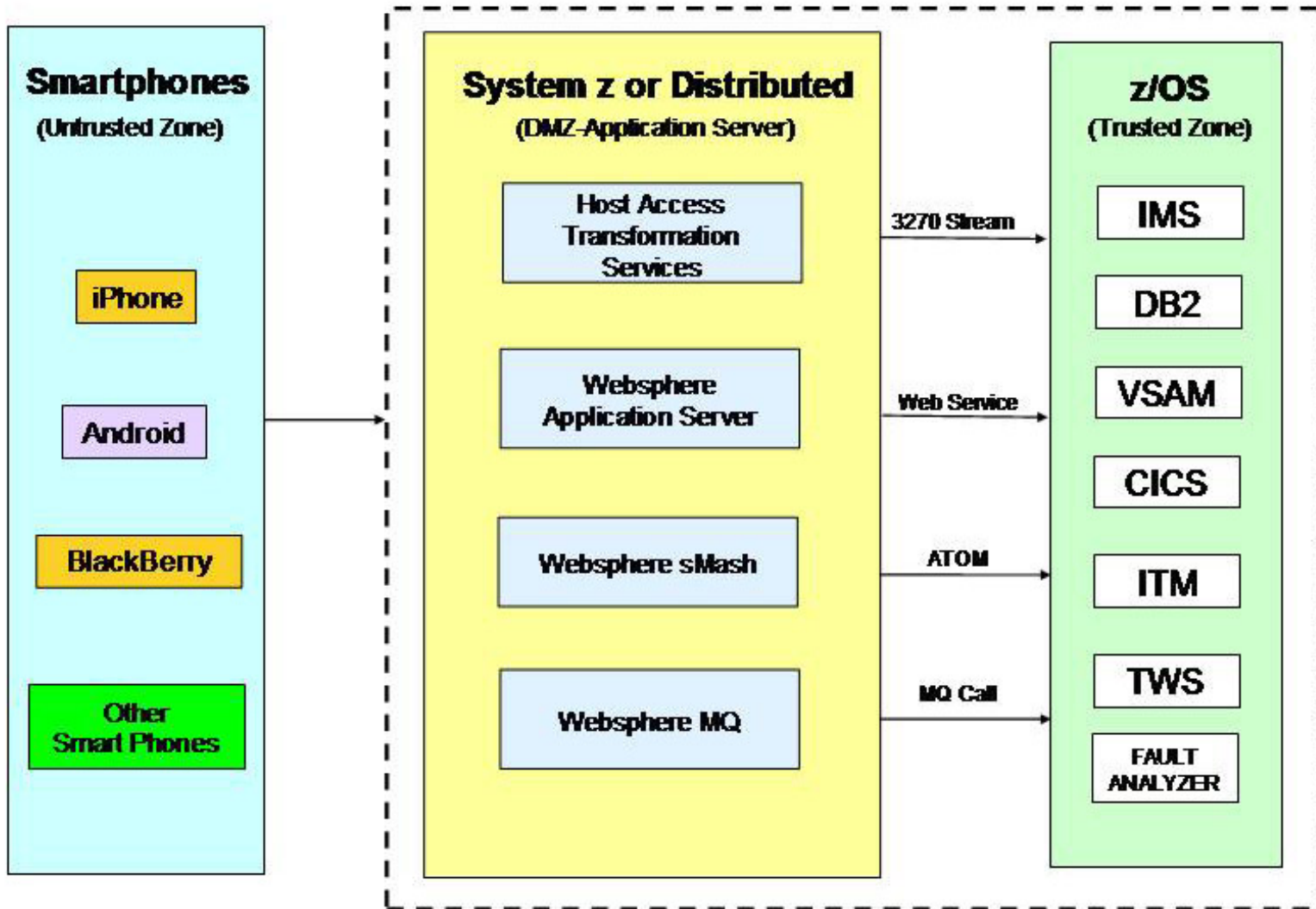
Alex Louwe Koolijmans
Dave Ellis
Ankur Goyal
Fabio Riva
Kenichi Yoshimura
Lydia Parziale

ibm.com/redbooks

Redbooks



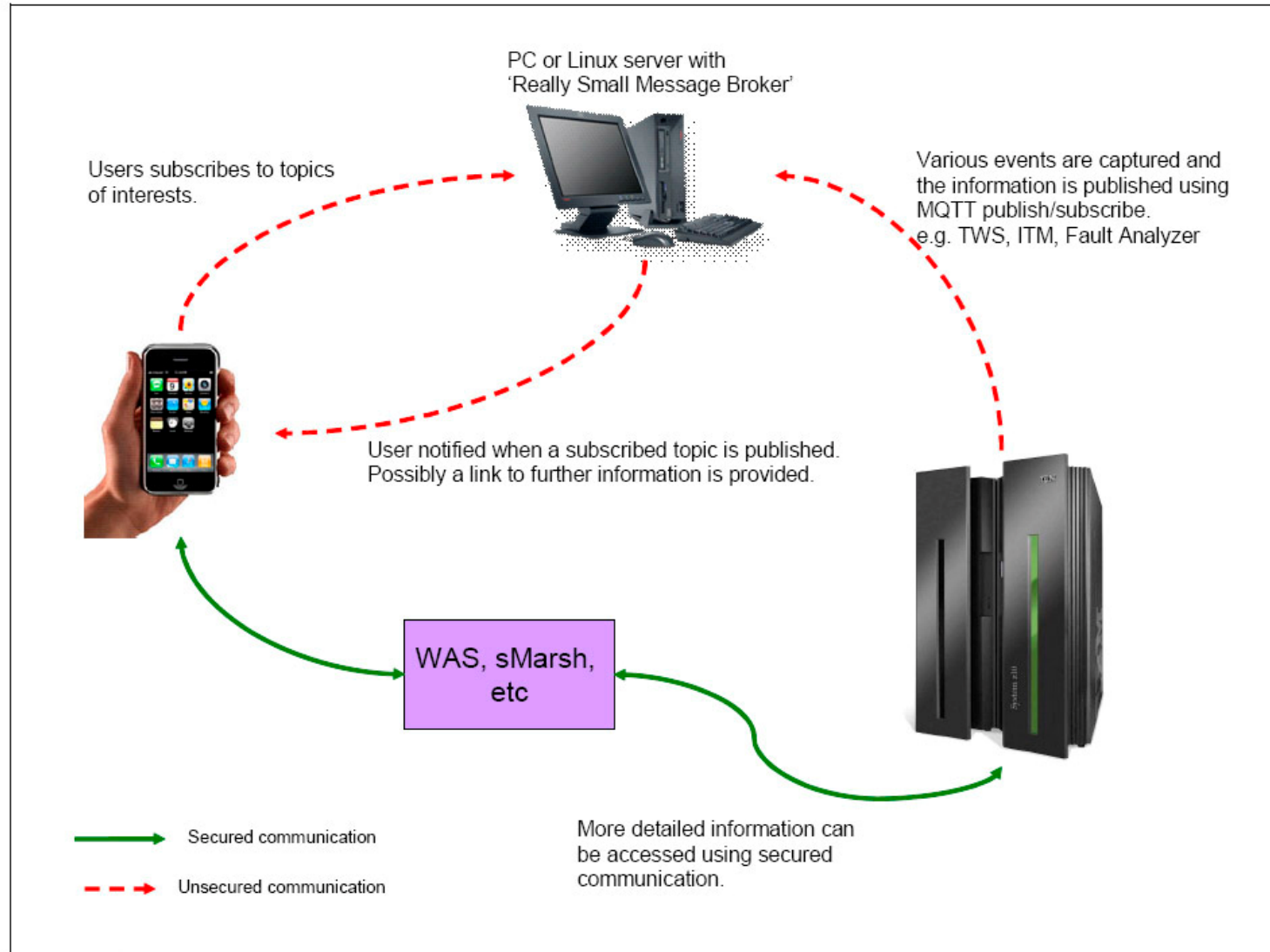
Architecture overview



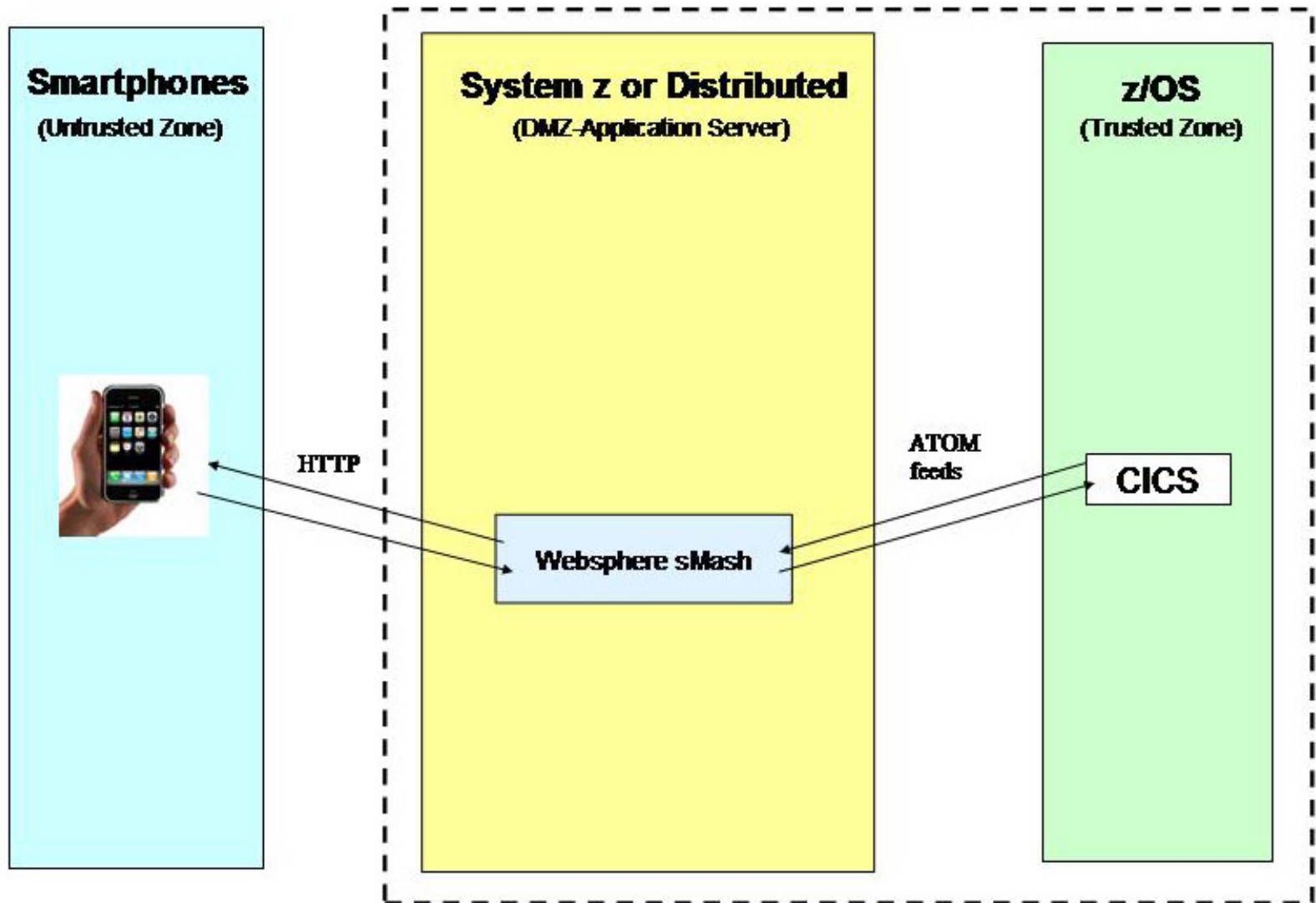
Possible two logical tiers on zSeries



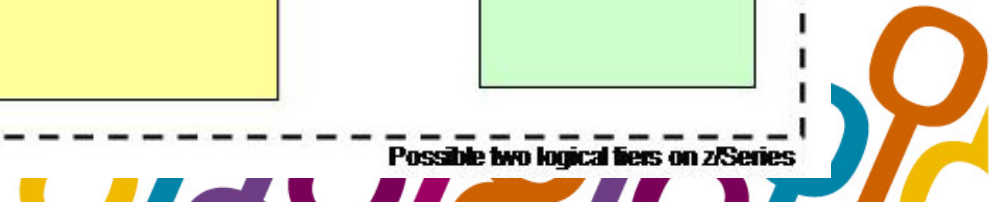
Using events with smartphone devices



Accessing CICS ATOM feeds with WebSphere sMash



Possible two logical tiers on zSeries



Demo for WebSphere sMash



Project zero main web page

The screenshot shows the Project Zero website in a Microsoft Internet Explorer browser window. The address bar displays <http://www.projectzero.org/>. The page features a blue navigation bar with links for **Download**, **Documentation**, **Community**, and **About**. The main content area includes a large orange heading "Welcome to Project Zero" followed by the text: "Delivering the best of agile Web 2.0 with PHP scripting, REST and Dojo in an integrated runtime and tooling package: **IBM WebSphere sMash**". Below this, there are four blue buttons: "Download Developer Edition 1.1", "Learn" (with an open book icon), "Script" (with the PHP logo), and "Create" (with a person and a pie chart icon). To the right, there is a "Google Custom Search" box and a "WebSphere sMash" section with the text "Community-Driven Commercial Development". Below that are social media icons for Twitter, Facebook, and YouTube. At the bottom, there is a "Featured Blog Posts" section with a link to "Announcing IBM WebSphere sMash 1.1.1.1 by fraenkel" and a "Spotlight: Partner Stories" section with a "Share My App" button. The browser's status bar at the bottom shows "Done" and "Internet".

Downloading websphere smash

PROJECT ZERO The development community for WebSphere sMash

[Download](#) [Documentation](#) [Community](#) [About](#)

[HOME](#) [DOWNLOAD](#) [SIGNUP](#)

Download WebSphere sMash DE v1.1.1.1

WebSphere sMash DE v1.1.1.1 includes the App Builder and a stable runtime to support evaluation, development, demonstration and testing of applications.

WebSphere sMash Command Line Interface

The Command Line Interface (CLI) contains the base support for developing and running apps. Additional runtime libraries are retrieved as needed from a module repository on ProjectZero.org.

DOWNLOAD AND INSTALLATION OF THE CLI

1. Download the [zero.zip](#) file and unzip it to any directory. This results in a subdirectory called zero that contains the commands for running the CLI.
2. Add the zero directory to the user's PATH environment variable.
3. Add the bin directory under the JDK installation directory to the user's PATH environment variable.

Refer to the [Getting Started Guide](#) for instructions on creating your first application.

SYSTEM REQUIREMENT FOR COMMAND LINE INTERFACE

- Windows®, Linux®, or Mac operating system
- Java SE Development Kit (JDK) (32-bit only):
 - Linux®: [5.0](#) or [6.0](#)
 - Windows®: [5.0](#) or [6.0](#)
 - Mac: [5.0](#) or [6.0](#)

Google Custom Search

Search

WebSphere sMash

Monza Download Page

Find the latest development drivers [here](#)

WebSphere sMash v1.0 Download Page

Our previous release is available [here](#)

Source Code

Go to the [Source Code](#) to see what really makes WebSphere sMash tick!

Done Internet

Overview of samples

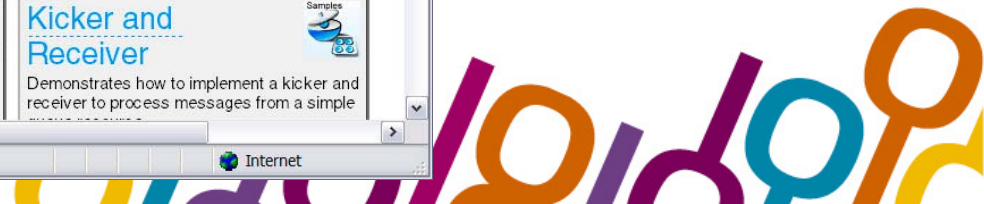
PROJECT ZERO Download Documentation

IBM WEBSHERE SMASH DEVELOPER'S GUIDE · OVERVIEW OF TUTORIALS, SAMPLES, AND DEMOS

Overview of tutorials, samples, and demos

After finishing the tutorials in the Getting Started guide, try out these additional tutorials, samples, and demos to help you learn how to use IBM® WebSphere® sMash.

Introductory	Intermediate	Advanced
<h4>Hello Dojo</h4> <p>Introduces basic concepts of the Dojo JavaScript toolkit</p> <p>Concepts: Dojo, JavaScript, Dijit, widgets</p>	<h4>Connection API</h4> <p>Contains example uses of the server-side Connection API, such as invoking a REST service and sending an e-mail</p> <p>Concepts: Connection API, e-mail</p>	<h4>OpenID</h4> <p>Demonstrates security features and illustrates how to leverage OpenID authentication</p> <p>Concepts: Open ID, authentication, security rules, extending a user registry</p>
<h4>Suggestion Box</h4> <p>Introduces the Zero Resource Model (ZRM) and the JavaScript library that make it easy to read and write data from a Web browser</p> <p>Concepts: Zero Resource Model, Dojo Grid</p>	<h4>Atom Feed</h4> <p>Illustrates how to render your data in Atom Syndication Format.</p> <p>Concepts: Atom</p>	<h4>iWidgets</h4> <p>Shows how to build and test iWidgets with IBM® WebSphere® sMash</p> <p>Concepts: iWidgets</p>
<h4>Employee Data</h4> <p>Provides an interface for managing a list of employees using RESTful conventions and SQL</p>	<h4>Flow Samples</h4> <p>Demonstrate a few of the basic features of the flow language</p>	<h4>Kicker and Receiver</h4> <p>Demonstrates how to implement a kicker and receiver to process messages from a simple</p>



Downloading sample

The screenshot shows the 'My Applications - App Builder' interface in Mozilla Firefox. The browser address bar shows 'http://localhost:8070/'. The application list is sorted by Name and includes the following entries:

Name	URL	Path
hello	http://localhost:8080/	C:\zero\IRIVA\hello\
RIVA connection.demo	http://localhost:8080/	C:\zero\IRIVA\
RIVA demo.iPhone2.demo	http://localhost:8080/	C:\zero\IRIVA\
zero.atom.demo	http://localhost:8080/	C:\zero\IRIVA\
zero.connection.demo	http://localhost:8080/	C:\zero\IRIVA\

A red arrow points to the 'Create from repository' option in the 'Actions' menu, with the text 'Choose this' written below it.



Adding sample

The screenshot shows the IBM App Builder web interface in a Mozilla Firefox browser window. The browser address bar shows `http://localhost:8070/`. The page title is "My Applications - App Builder".

The main content area is titled "My Applications" and includes a search bar and a "Filter..." input. On the left, an "Actions" sidebar contains the following options:

- Create new application
- Create from repository
- Open existing application
- Import application

A red arrow points to "Create from repository" with the text "Click on this".

The "Create from repository" dialog box is open, showing the following fields and options:

- Module group: `stable` (with a "Refresh" button)
- Module to copy or link to: `hello` (with a "Filter" label and arrow pointing to the input)
- Search results list: `zero:zero.hello:demo:1.1.1.1.30020`
- Buttons: "Create" and "Cancel"

A red arrow points to the "Create" button with the text "then click on CREATE".

Below the dialog box, the following text is displayed:

Full path to the new application (application home) will be [Root directory] / [Application name].
Application name: `zero.hello:demo` Link application to the specified module
Root directory: `C:\zero\IRIVA\`

The footer of the page contains navigation links: Project Zero Home, Samples, Help Documentation, Project Zero Forums, and About About Project Zero.

At the bottom left, the text "Completato" is visible.

WebSphere sMash application page

new application

click here to start application

after starting, click here to access application

Application Name	URL	Path
hello	http://localhost:8080/	C:\zero\RIVA\hello!
RIVA connection.demo	http://localhost:8080/	C:\zero\RIVA\
RIVA demo.iPhone2.demo	http://localhost:8080/	C:\zero\RIVA\
zero.atom.demo	http://localhost:8080/	C:\zero\RIVA\
zero.connection.demo	http://localhost:8080/	C:\zero\RIVA\
zero.hellodojo.demo	http://localhost:8080/	C:\zero\RIVA\



Testing sample application

New page - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti Aiuto

http://localhost:8080/ Google

Più visitati Come iniziare Ultime notizie Tivoli Integrated Portal Fabio Riva Fabio

My Applications - App Builder New page

type name here HelloWorld with Dojo

Name: Chiara Text

Most recent text:

Hello Chiara

History:

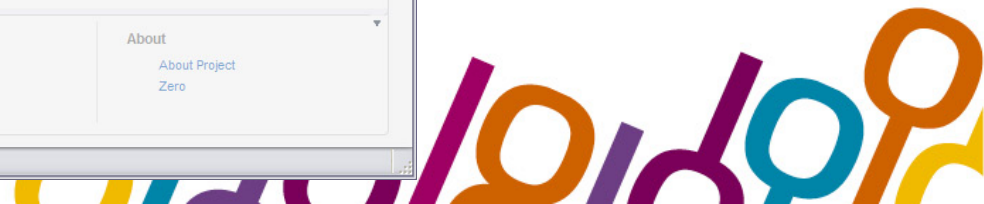
Hello Chiara

text will appear here

Completato

Adding connection sample, in order to modify it

The screenshot shows the 'My Applications - App Builder' interface in Mozilla Firefox. The browser address bar shows 'http://localhost:8070/'. The main content area displays 'My Applications' with a search filter. A 'Create from repository' dialog box is open, showing a list of modules. A red arrow points to the 'Create' button with the text 'type here' above it. The 'Module group' is set to 'stable' and the 'Module to copy or link to' is 'connection.demo'. The 'Application name' is 'zero.connection.demo' and the 'Root directory' is 'C:\zero\RIVA\'. The 'Link application to the specified module' checkbox is unchecked.



Connection sample

Connection API Samples - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti Aiuto

http://localhost:8080/

Più visitati Come iniziare Ultime notizie Tivoli Integrated Portal Fabio Riva Fabio

My Applications - App Builder Connection API Samples

Connection API Samples

HTTP POST sample

Makes an HTTP POST, sending and receiving JSON

Invoke HTTP sample

HTTP POST sample with handler

Makes an HTTP POST, sending and receiving JSON. Uses a connection handler to convert between a Map and JSON to save the end user from having to manage the conversion.

Invoke HTTP with handler sample

HTTP GET sample with read timeout configured

Makes an HTTP GET, sending and receiving plain text. Overrides the default read timeout by specifying a value of 30 seconds in the destination configuration

Invoke HTTP with configured timeout sample

E-mail sample

Makes use of the EmailConnection API to send an HTML e-mail with embedded and attached images.

Before running the E-Mail sample, your SMTP server hostname needs to be configured in zero.config and the application restarted

For example:

```
/config/connection/defaults/smtp/hostname = "smtp.yourdomain.com"
```

File PUT sample

Writes text to a file

Enter some text to be written to file:

Completato

Modifying it...

The screenshot shows a Mozilla Firefox browser window displaying the CLEAR App Builder interface. The browser address bar shows `http://localhost:8070/#view%3DAppEditor%2Bapp%3D4%2Buiview%3DallFilesView%2Bf`. The CLEAR interface includes a menu bar (File, Modifica, Visualizza, Cronologia, Segnalibri, Strumenti, Aiuto) and a toolbar with buttons for File Editor, Dependencies, Explorer, Console, and Debug. The main content area shows a code editor with Groovy code for an `onPOST()` method. The code includes HTML meta tags for viewport, mobile app capabilities, and status bar style, and a Groovy call to a connection service. Three red annotations with arrows point to specific parts of the code:

- HTML statements for iPhone layout (3)**: Points to lines 15-19, which define the HTML head and meta tags.
- instruction to get parameter from POST (2)**: Points to line 24, `def message = request.params.message[0];`.
- pointer to ATOM service on CICS z/OS (1)**: Points to line 29, `def conn = new Connection("http://9.71.198.33:38083/atom/z/atomfriva?s=" + message);`.

```
1/*
2....created for ITSO residency ZS-9A12
3*/
4
5package http;
6
7import java.util.HashMap;
8
9import zero.core.connection.Connection;
10import zero.json.Json;
11
12def onPOST() {
13
14// ITSO
15  def ret = "<html><head><title>Info da CICS</title>" ;
16  ret = ret + '<meta name="viewport" content="user-scalable=no, width=device-width" />';
17  ret = ret + '<meta name="apple-mobile-web-app-capable" content="yes" />';
18  ret = ret + '<meta name="apple-mobile-web-app-status-bar-style" content="black" />';
19  ret = ret + '<link rel="apple-touch-startup-image" href="loading.png" /></head>';
20
21//  def ret = "HTTP (with timeout set) invocation successful";
22
23// ITSO
24  def message = request.params.message[0];
25
26  try {
27    // Create a new Connection targetted at the http address we wish to post data to.
28// ITSO
29    def conn = new Connection("http://9.71.198.33:38083/atom/z/atomfriva?s=" + message);
30//    def conn = new Connection("http://localhost:8080/http/alt2/dummyAlt2.groovy");
31
32
```



Modifying it...

The screenshot shows the CLEAR application editor in Mozilla Firefox. The browser address bar shows `http://localhost:8070/#view%3DAppEditor%2Bapp%3D4%2Buiview%3DallFilesView%2Bf`. The application interface includes a menu bar (File, Modifica, Visualizza, Cronologia, Segnalibri, Strumenti, Aiuto), a toolbar (Undo, Redo, Go to line, Checkpoint), and a sidebar with Recent Files and All Files. The main editor displays Groovy code for parsing instructions and preparing HTML for an iPhone. Red annotations highlight the parsing and HTML preparation steps.

```
42 // IISO
43 def codpro = '-';
44 def strarray = str.split("<codpro>");
45 def strarray2 = strarray[1].split("</codpro>");
46 codpro = strarray2[0];
47 def desc = '-';
48 strarray = str.split("<desc>");
49 strarray2 = strarray[1].split("</desc>");
50 desc = strarray2[0];
51 def stock = '-';
52 strarray = str.split("<stock>");
53 strarray2 = strarray[1].split("</stock>");
54 stock = strarray2[0];
55 def deliv = '-';
56 strarray = str.split("<deliv>");
57 strarray2 = strarray[1].split("</deliv>");
58 deliv = strarray2[0];
59 def price = '-';
60 strarray = str.split("<price>");
61 strarray2 = strarray[1].split("</price>");
62 price = strarray2[0];
63 def discount = '-';
64 strarray = str.split("<discou>");
65 strarray2 = strarray[1].split("</discou>");
66 discount = strarray2[0];
67 def str2 = '<br><hr><p>';
68 str2 = str2 + '<table border="0" width="100%">';
69 str2 = str2 + '<tr><td>Product Code</td><td style="text-align:right">' + codpro + '</td></tr>';
70 str2 = str2 + '<tr><td>Description</td><td style="text-align:right">' + desc + '</td></tr>';
71 str2 = str2 + '<tr><td>Unit Price</td><td style="text-align:right">' + price + " euro</td>";
72 str2 = str2 + '<tr><td>Qty Available</td><td style="text-align:right">' + stock + " pezzi<";
73 str2 = str2 + '<tr><td>Production time</td><td style="text-align:right">' + deliv + " gg<";
74 str2 = str2 + '<tr><td>Max Discount</td><td style="text-align:right">' + discount + "%<t";
75 str2 = str2 + "</table>";
76
77
```

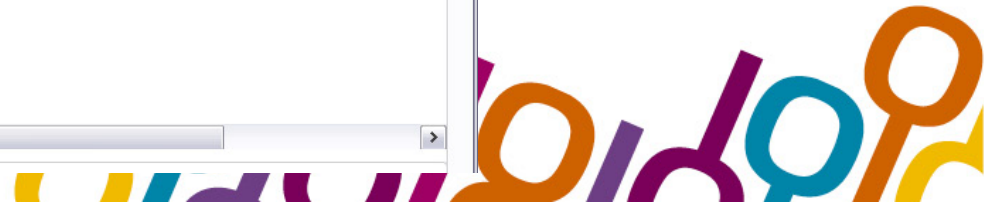
parsing instructions to extract ATOM feeds data (A)

preparation of HTML to be sent to iPhone (B)

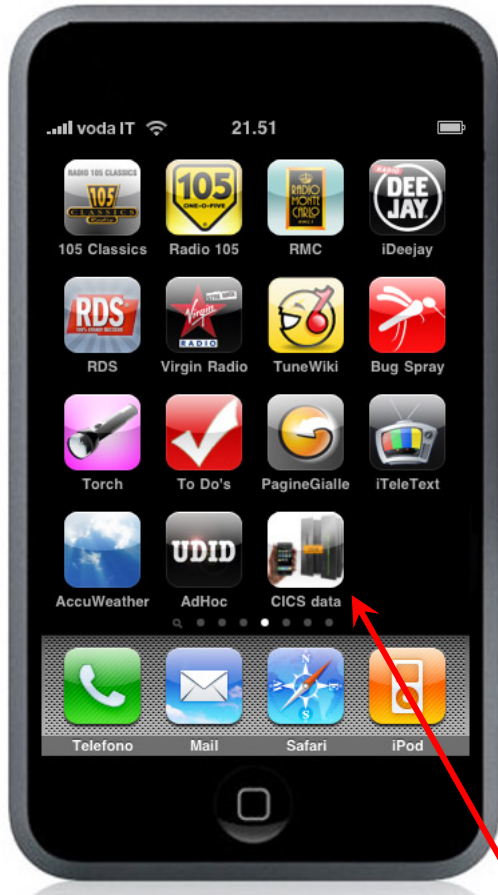
Inserting FORM

The screenshot shows the CLEAR App Builder interface in Mozilla Firefox. The browser address bar shows the URL: `http://localhost:8070/#view%3DAppEditor%2Bapp%3D4%2Buiview%3DallFilesView%2Bf`. The interface includes a navigation bar with "WebSphere", "sMash", "My Applications", "My Repository", and "CLEAR". Below this is a toolbar with "File Editor", "Dependencies", "Explorer", "Console", and "Debug". The main area is a code editor for `public/index.gt @ 1:1`. The code is as follows:

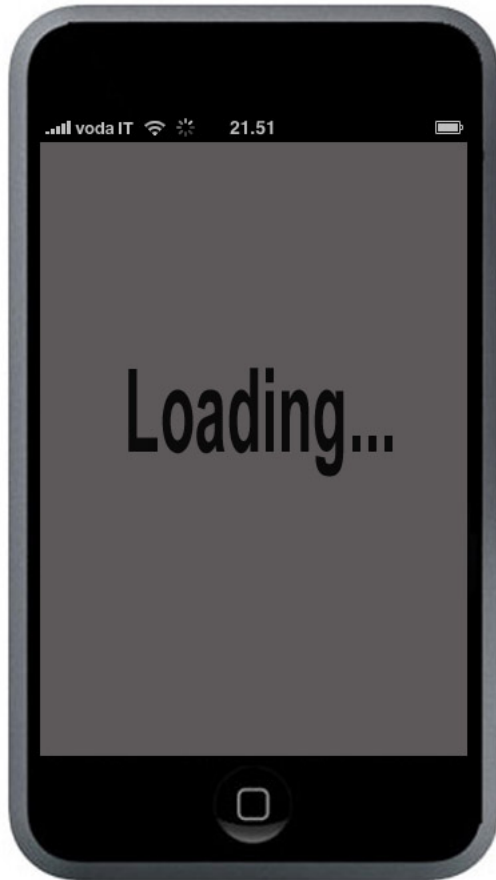
```
1<!--/*
2...created for IISO residency ZS-9A12
3*/-->
4
5<html>
6<head>
7<title>getting ATOM feeds data from CICS</title>
8<meta name="viewport" content="user-scalable=no, width=device-width" />
9<meta name="apple-mobile-web-app-capable" content="yes" />
10<meta name="apple-mobile-web-app-status-bar-style" content="black" />
11<link rel="apple-touch-startup-image" href="loading.png" />
12<link rel="apple-touch-icon" href="iphone.jpg" />
13</head>
14<body style="font-family: arial; font-size:16px; background-color:#e4eaf4" background="images/z10.gi:
15<br><hr>
16<h2>QUERY ON CICS DATA</h2>
17<hr>
18<h3>Query</h3>
19<p>This query calls CICS z/OS via ATOM feeds and return mainframe data. In the FORM the product code
20<form id="http1" method="POST" action="http/httpWithTimeout.groovy">
21Select Product Code
22  <SELECT size="3" name="message" style="font-size: 12pt">
23    <OPTION selected value="0100">0100</OPTION>
24    <OPTION value="0200">0200</OPTION>
25    <OPTION value="0300">0300</OPTION>
26    <OPTION value="0400">0400</OPTION>
27    <OPTION value="0500">0500</OPTION>
28  </SELECT>
29<input type="submit" name="invoke" style="font-size: 12pt" value="require data"/>
30</p>
31</form>
32<br>
33<hr>
34</body>
35</html>
```



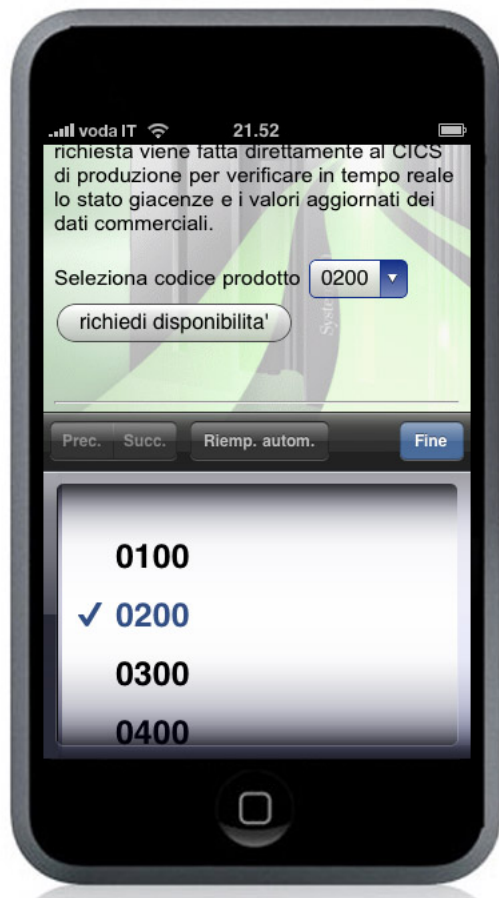
My demo situational appl



APPLICATION ICON



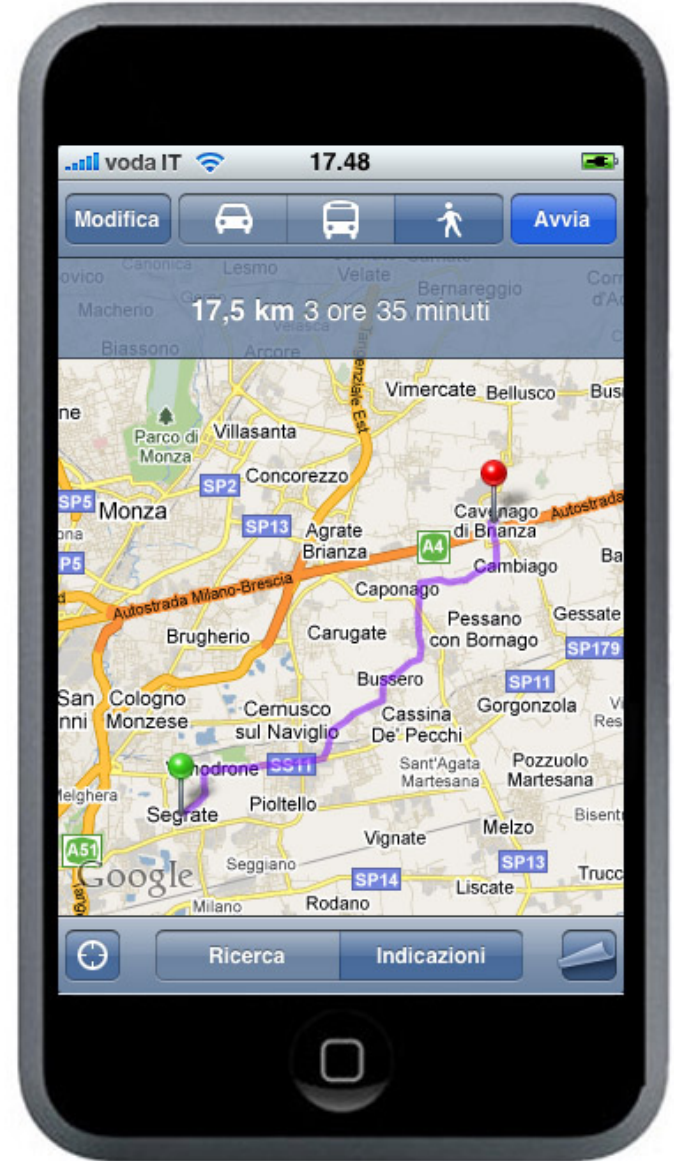
....CICS query



....call to external service



LINK TO EXTERNAL SERVICE



Video of the demo

[Video for iPhone demo](#)



Summary



Summary

- **Mainframe evolution was impressive both in hardware and in software. A lot of new functions were introduced during the years and mainframe is still the leader platform for business, but there's still lack of user interface accessing the platform.**
- **Web 2.0 offers desktop-like functionality in accessing mainframe. Joining lightweight Web 2.0 technologies with robust SOA infrastructure is a very powerful solution for IT industry.**
- **With mash-up services it's possible to join existing mainframe services with external ones, creating new opportunities. This 'mix' can be done with situational applications developed by users, and it's possible to consolidate them on z/Linux or WebSphere.**
- **Smart-phones devices offer new possibility to mainframe applications. Joining openness provided by smart-phones devices with openness provided by web 2.0 technologies will offer mainframe the possibility to be fully integrated in new IT technologies.**

