Impact2010

The Premier Conference for Business and IT Leaders

Fabio Riva

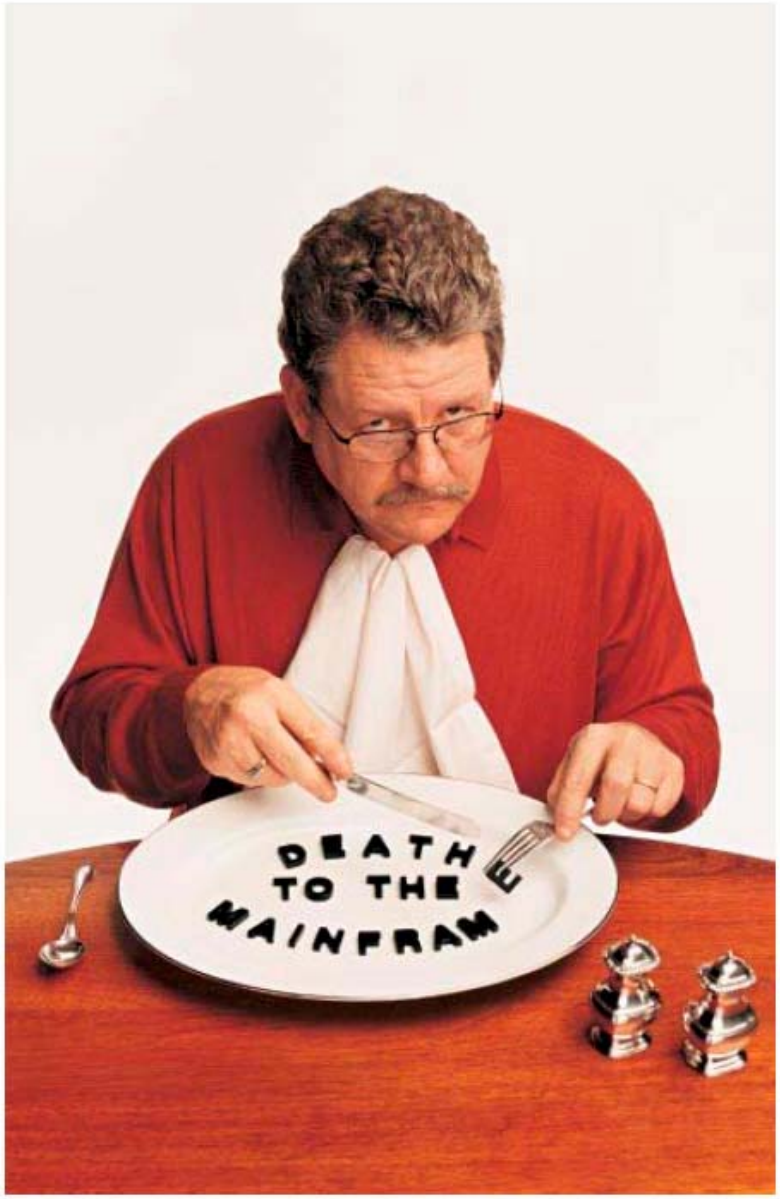
Mainframes e Web 2.0

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

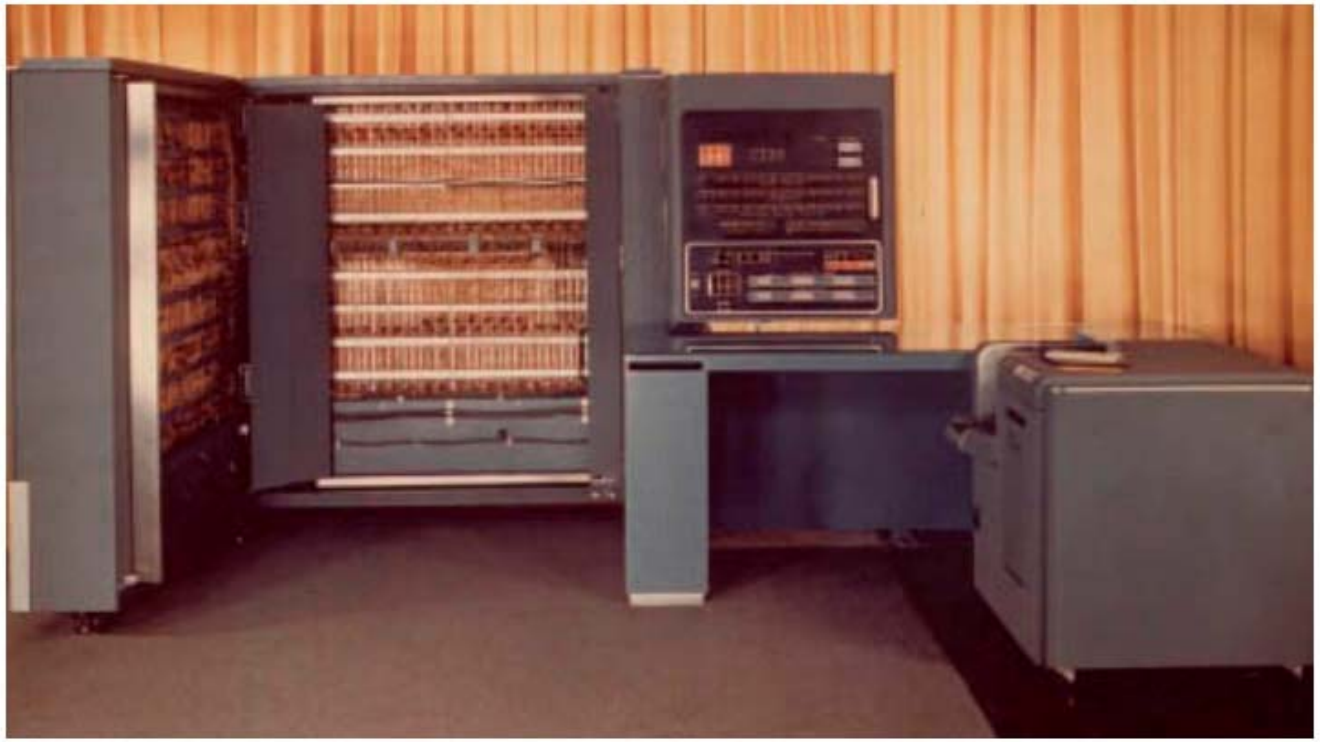
- **"It's clear that corporate customers still like to have centrally controlled, very predictable, reliable computing systems – exactly the kind of systems that IBM specializes in."**
  – Stewart Alsop, February 2002



IBM

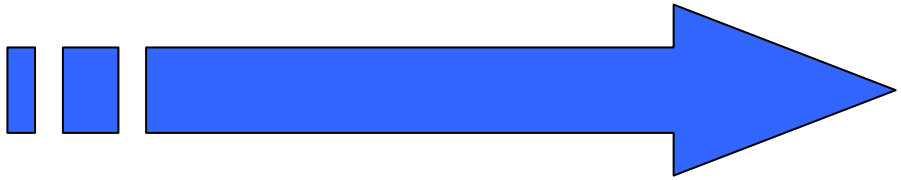
It all started with the IBM 701

- **Circa 1951:**



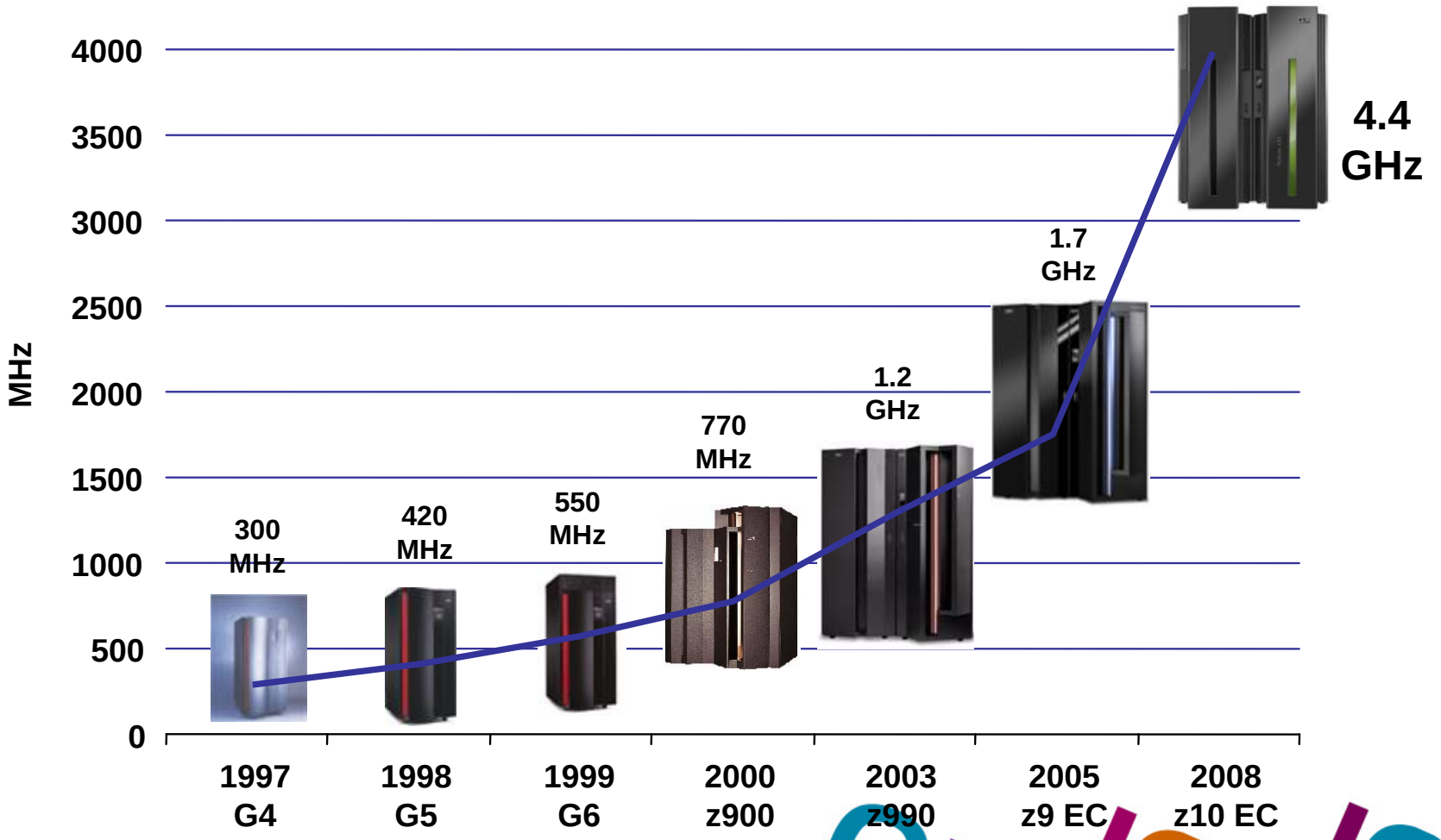
- **And proceeded through the 700, 1400, S/360 & 370, 303x, 308x, 4300, S/390 Parallel Server, to today's eServer zSeries**

Hardware – Continuous evolution

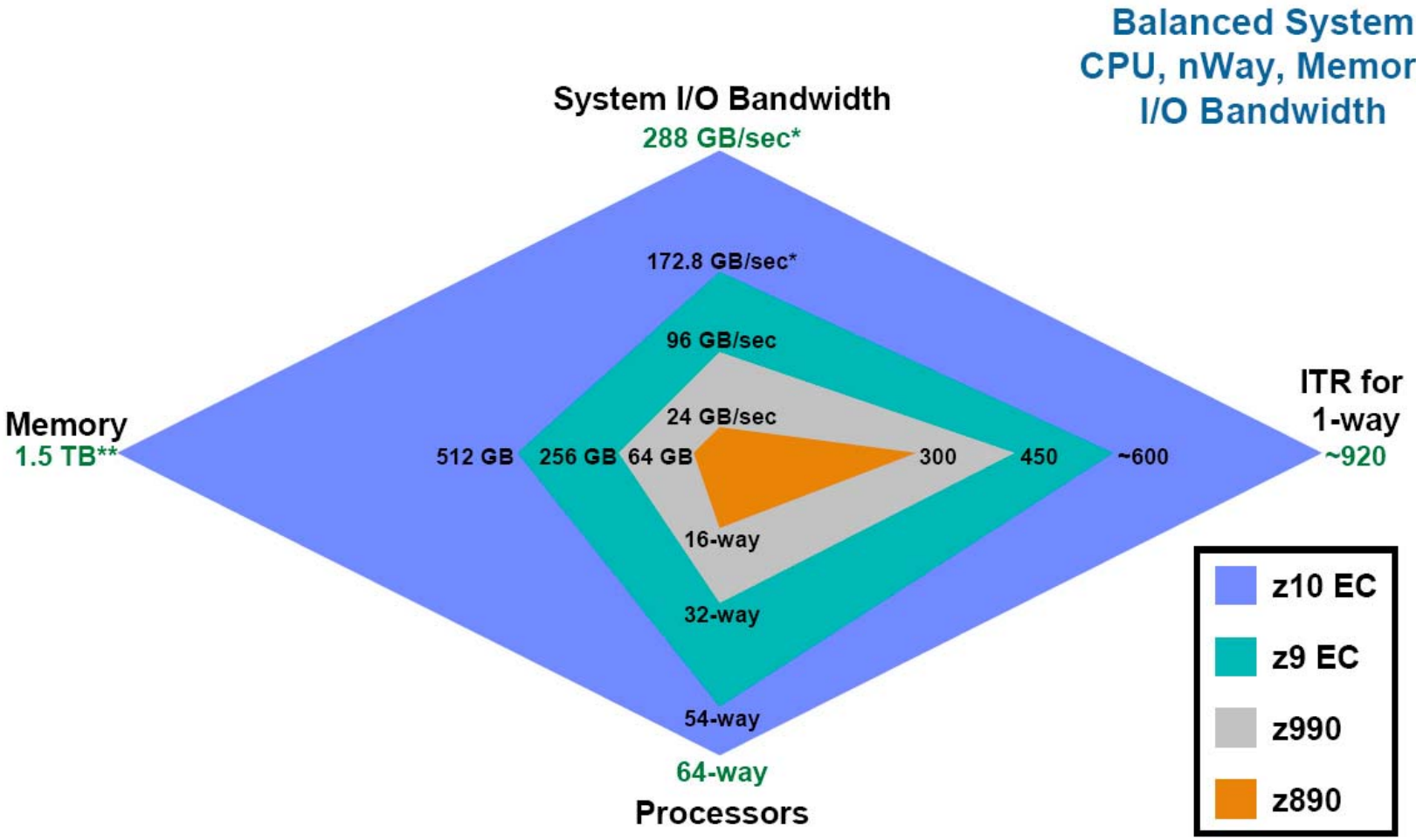


HW	S/360	S/370	S/370XA 31bits	ESA/390	CMOS Parallel Sysplex	z/Architecture 64bits
	1964	1970	1980	1990	2000	2004
O.S. SW	MVT, PCP, MFT, DOS	MVS, VTAM	MVS/XA	MVS/ESA OS/390	USS – TCP/IP	z/OS
	CICS / IMS	VM	DB2	WebSphere	Linux, z/VM	

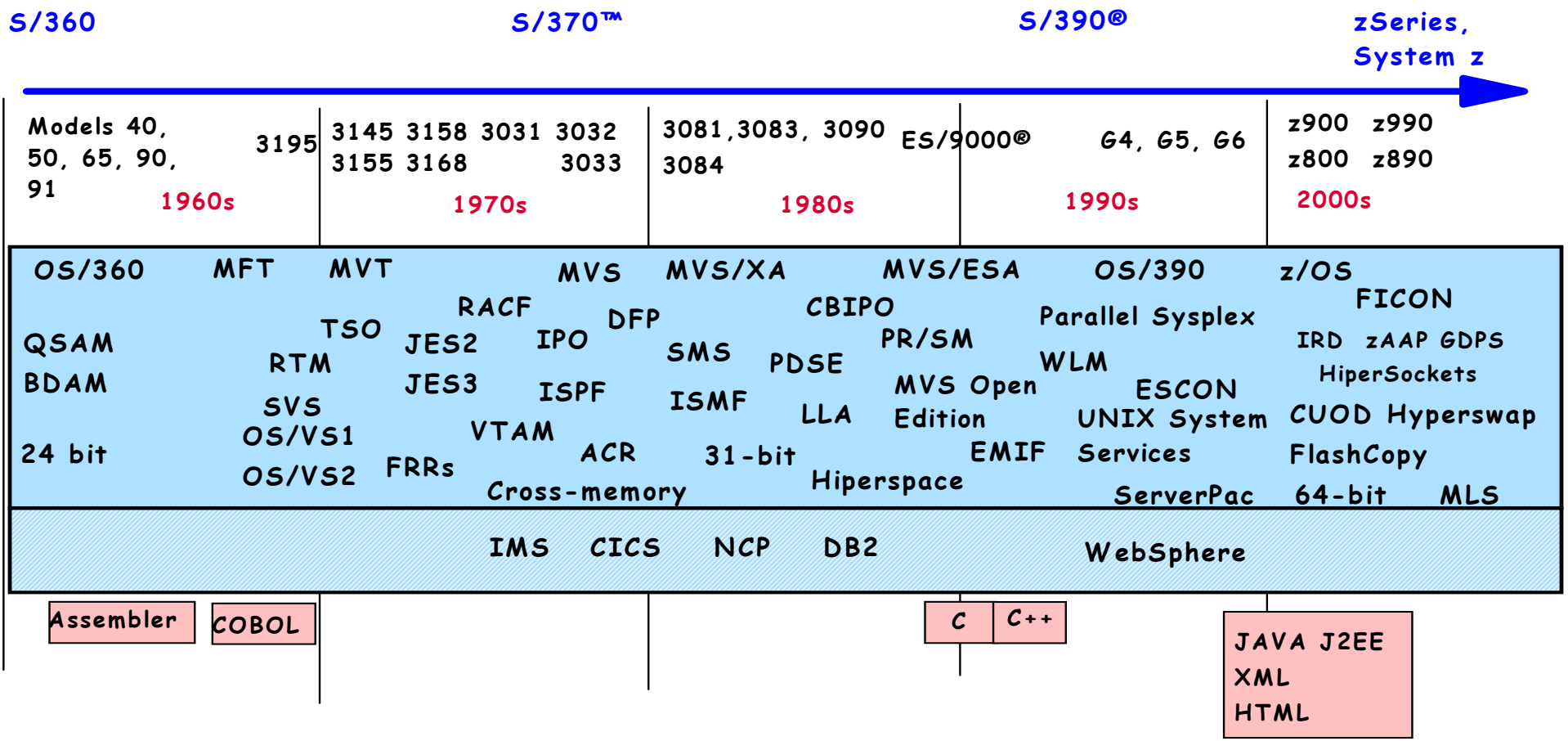
IBM z/Series machines continues the HW evolution



IBM System z: Balanced Systems Design

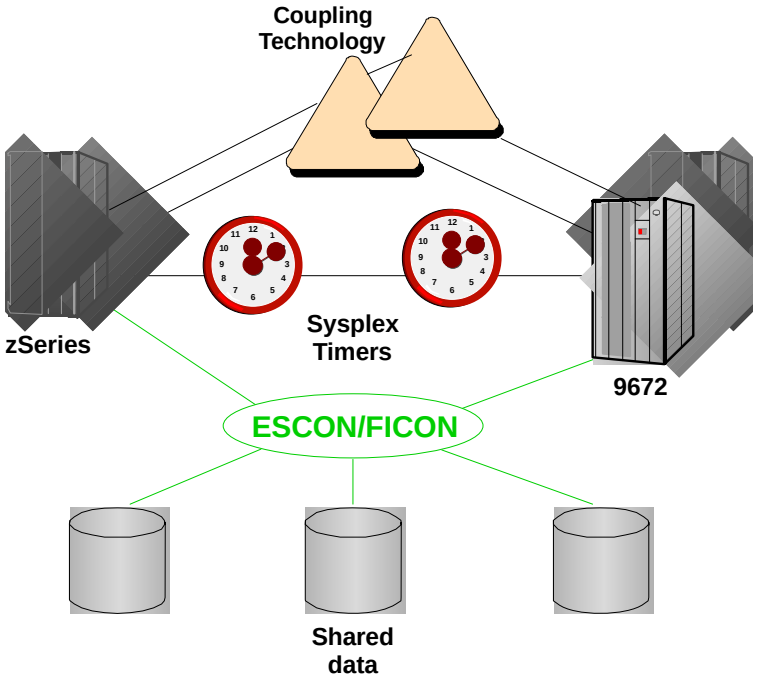


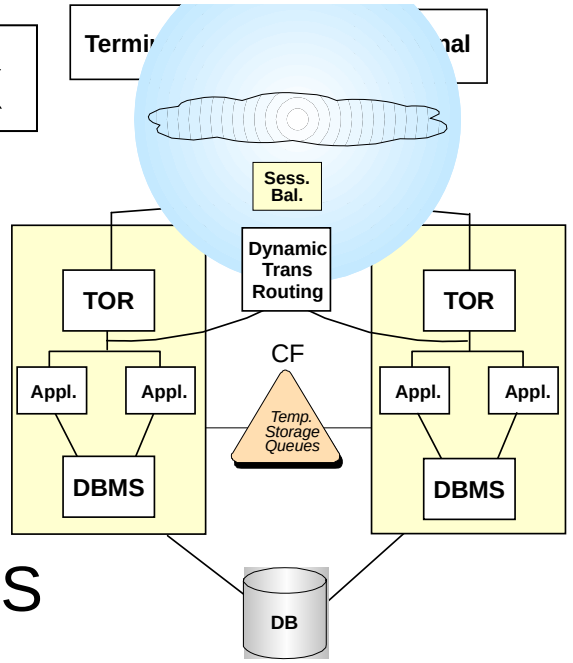
Software – Continuous innovation



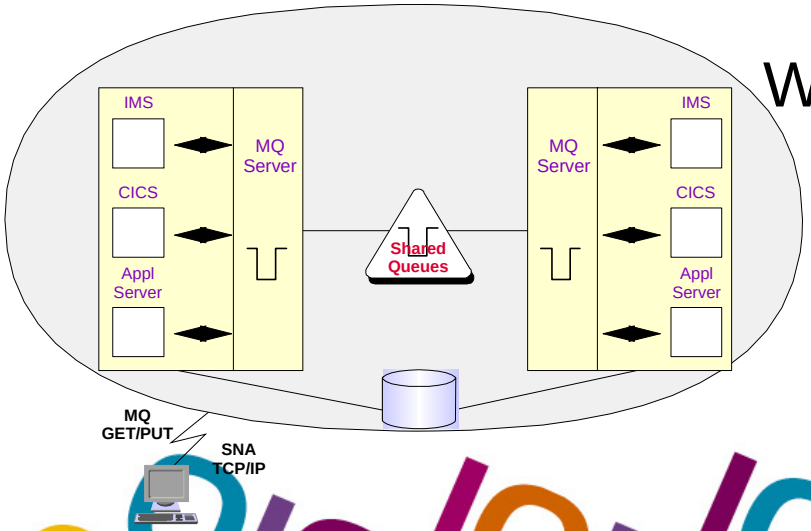
Application investment protection

Mainframe functions: Parallel Sysplex





CICS



WMQ

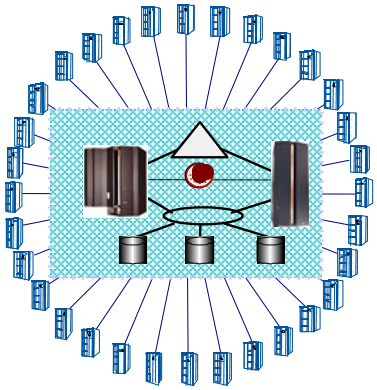
- Continuous Availability
- Flexible Growth
- Scalability
- Reduced Cost
- Leverage S/390 Investment

Mainframe functions: z/OS Continuous Availability

Single System

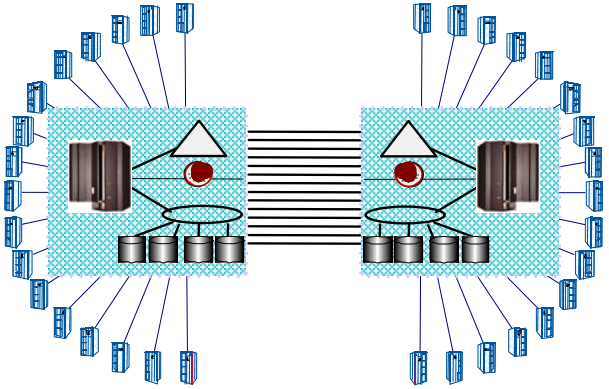
- **Built In Redundancy**
- **Capacity Upgrade on Demand**
- **Capacity Backup**
- **Hot Pluggable I/O**

Parallel Sysplex



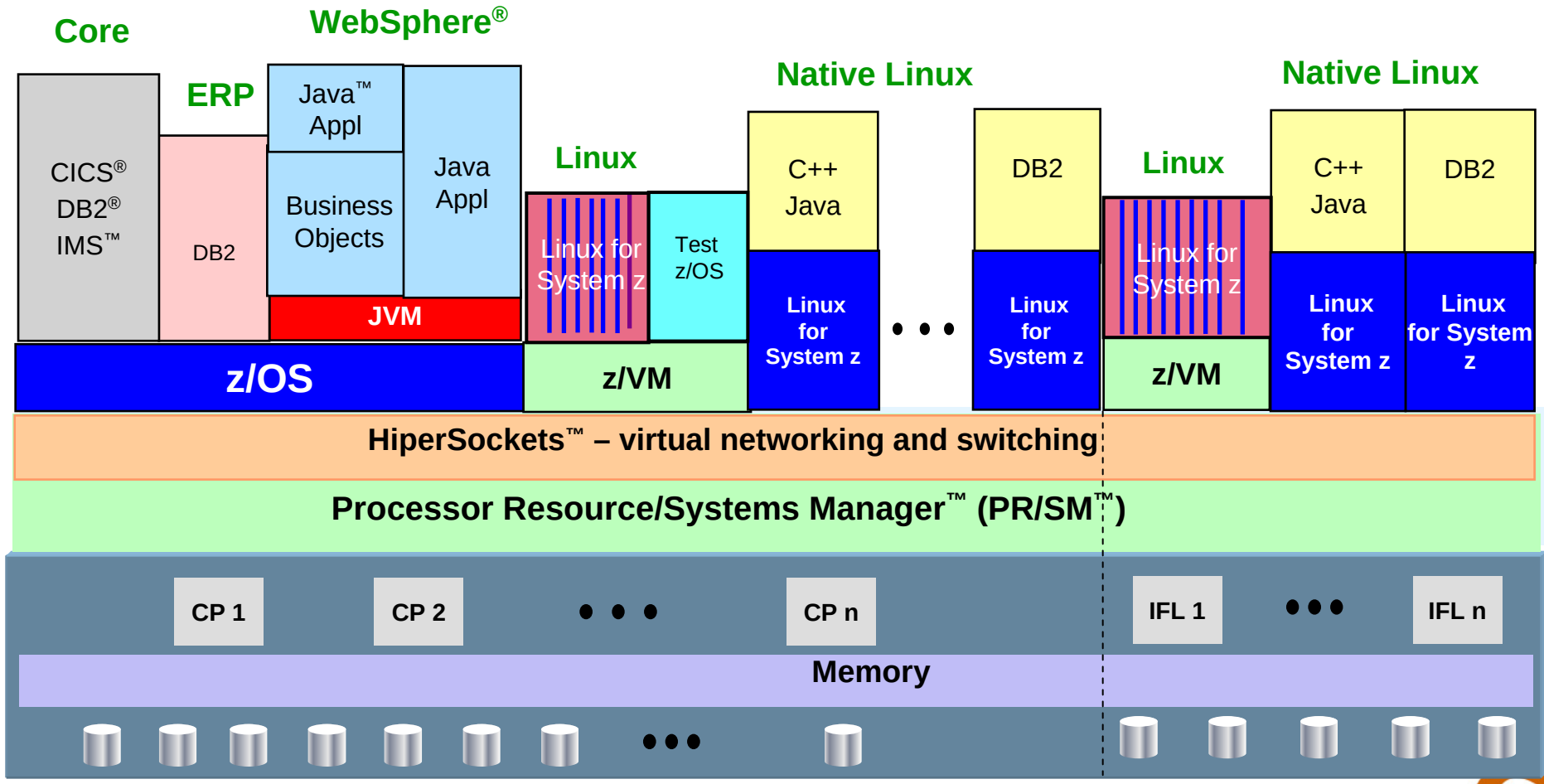
- **Addresses Planned/Unplanned HW/SW Outages**
- **Flexible, Nondisruptive Growth**
  - Capacity beyond largest CEC
  - Scales better than SMPs
- **Dynamic Workload/Resource Management**

GDPS



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

Mainframe functions: Virtualization



IBM

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 (tipically 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.

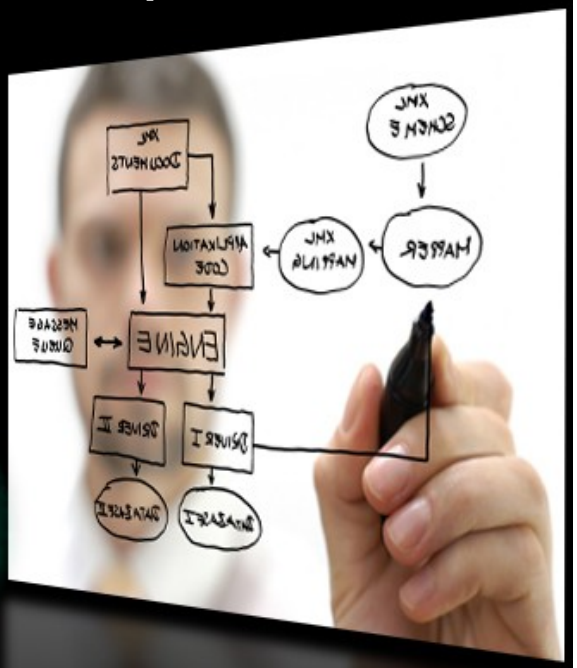
Easy access to the mainframe: use of Web 2.0

Simple to use

AJAX
- Highly Interactive
- Browser invoked services

JSON / XML / ATOM
- Information exchange
- JavaScript Friendly

Simple to access



REST
- Easily invoked
- HTTP-Centric Patterns

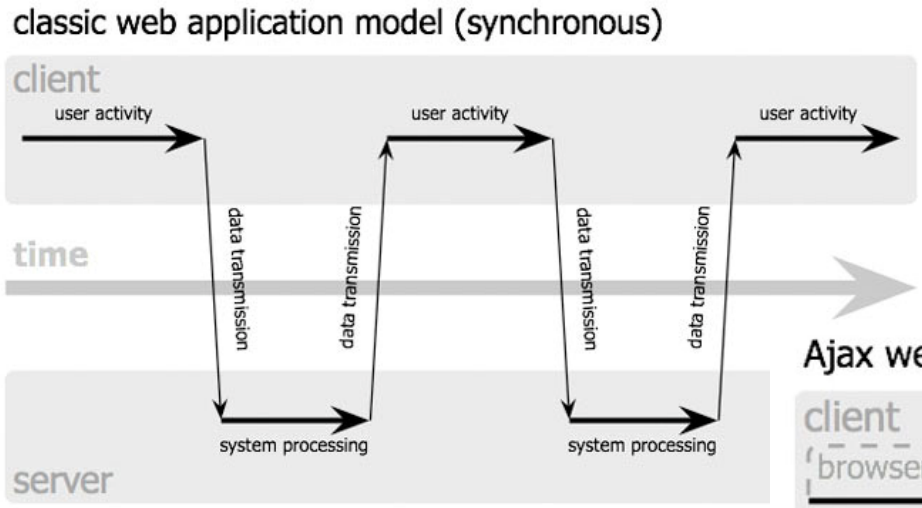
These are some technolgies used in Web 2.0 appls

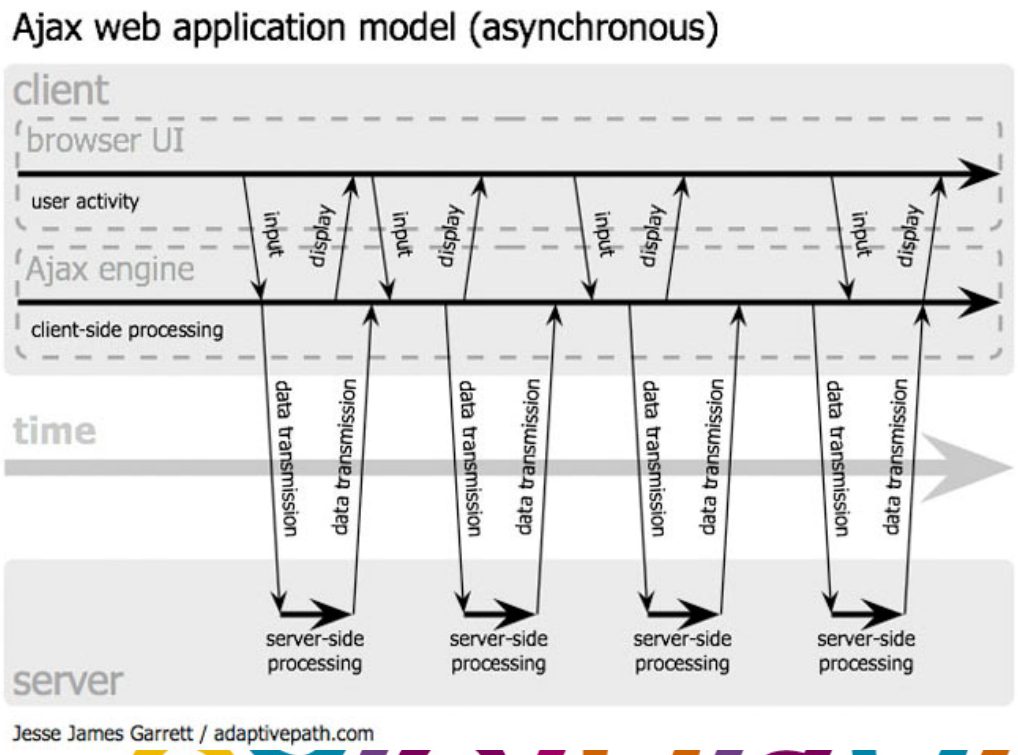
- **AJAX: This technology allows to access data in asyncronous way reducing user wait times. AJAX is not really new, but it's a combination of existing technologies: XMLHttpRequest, Javascript, CSS and XML. AJAX is asyncronous. it allows updating only parts of a web page so you no longer need to refresh 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**

Asyncronous Javascript And Xml (AJAX)





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

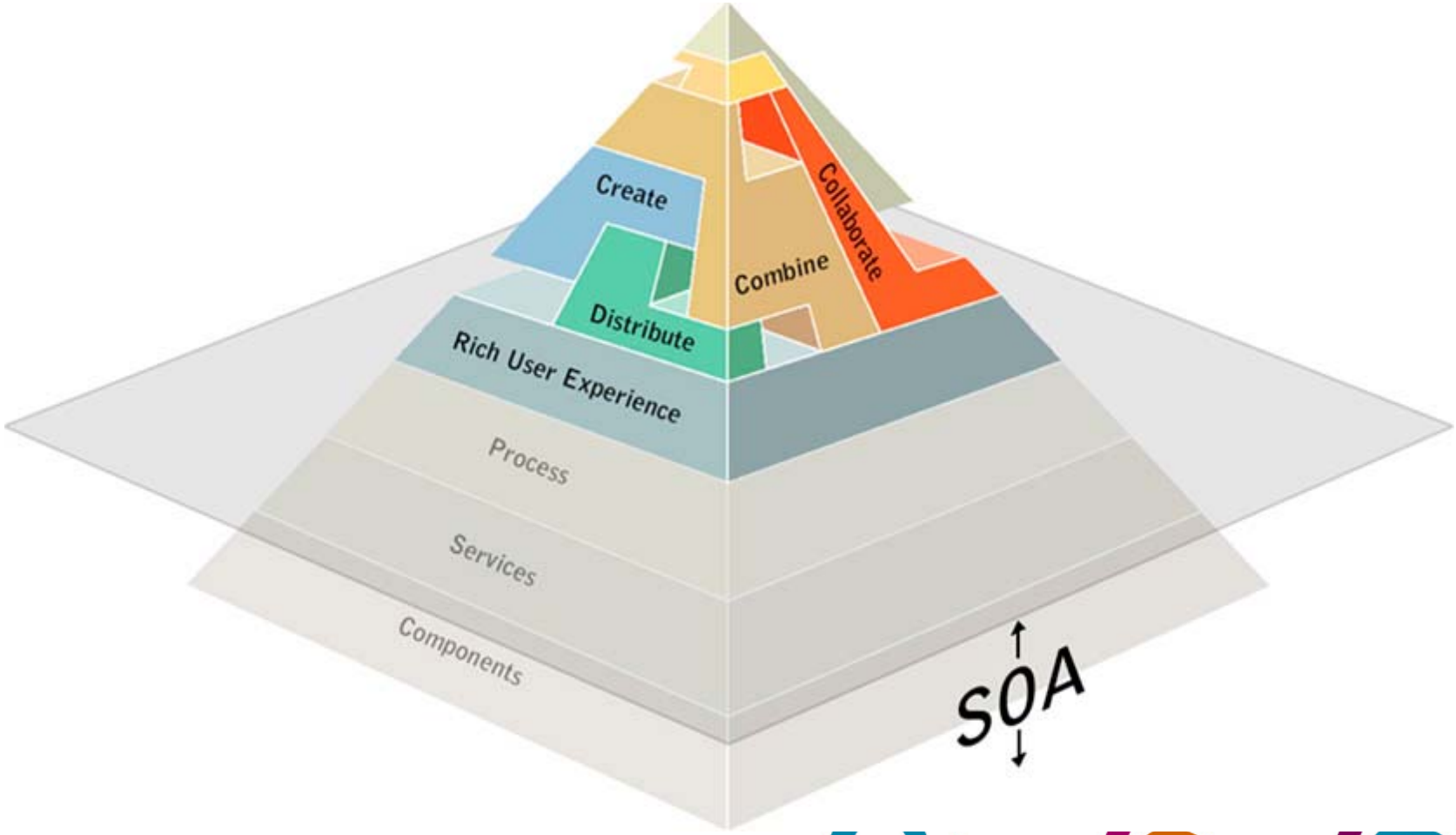
```
<feed xmlns="http://www.w3.org/2005/Atom"
      xml:lang="en"
      xml:base="http://www.example.org">
  <id>http://www.example.org/myfeed</id>
  <title>A Sample Feed</title>
  <updated>2008-06-16T18:34:42Z</updated>
  <link href="/blog" />
  <link rel="self" href="/myfeed" />
  <entry>
    <id>http://www.example.org/entries/1</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

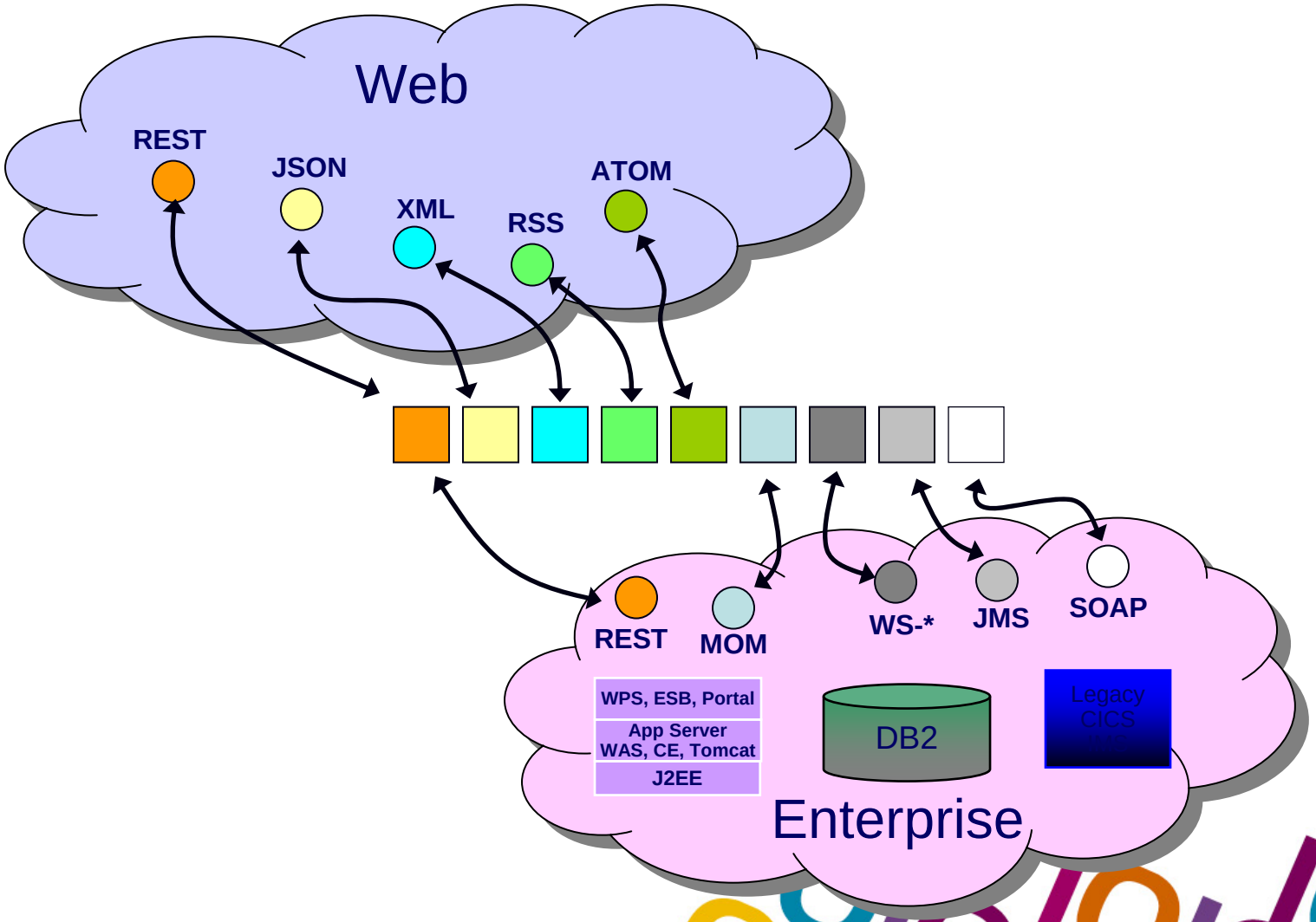
Reuse of existing services: RESTful SOA

- **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 bocome perhaps the single most widely deployed form of service orientation because of its semplicity.**
- **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.**

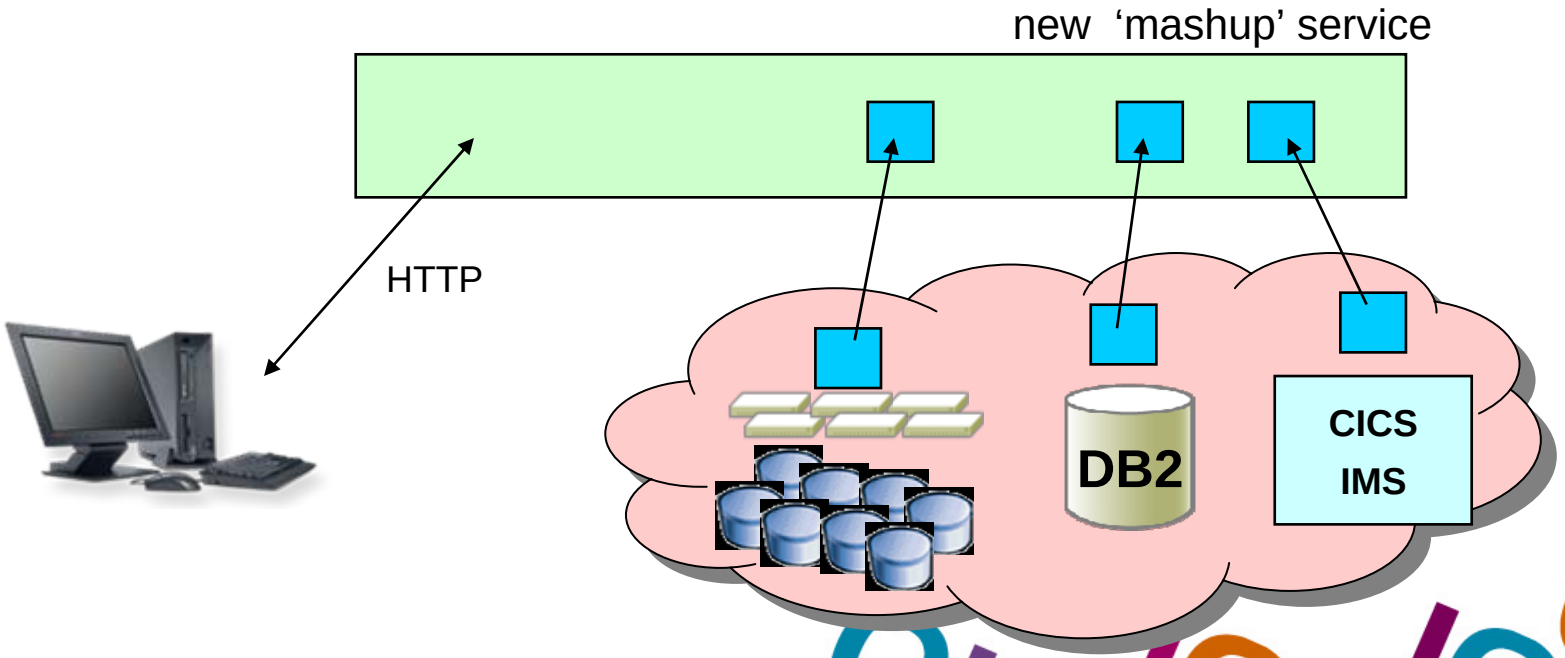
RESTful SOA: SOA and Web 2.0



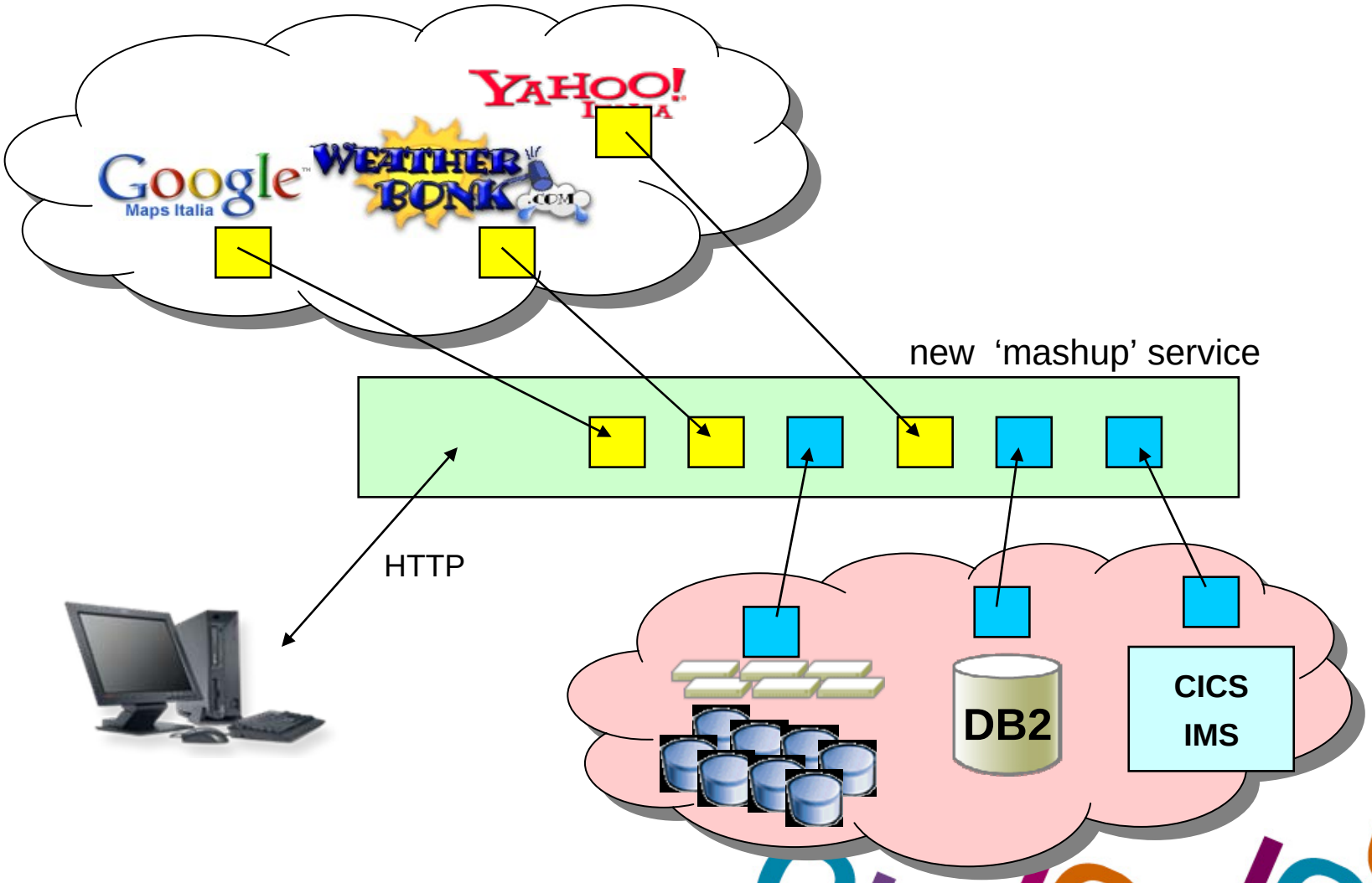
Bridging “RESTful SOA” and “Enterprise SOA”



Mashup of internal services

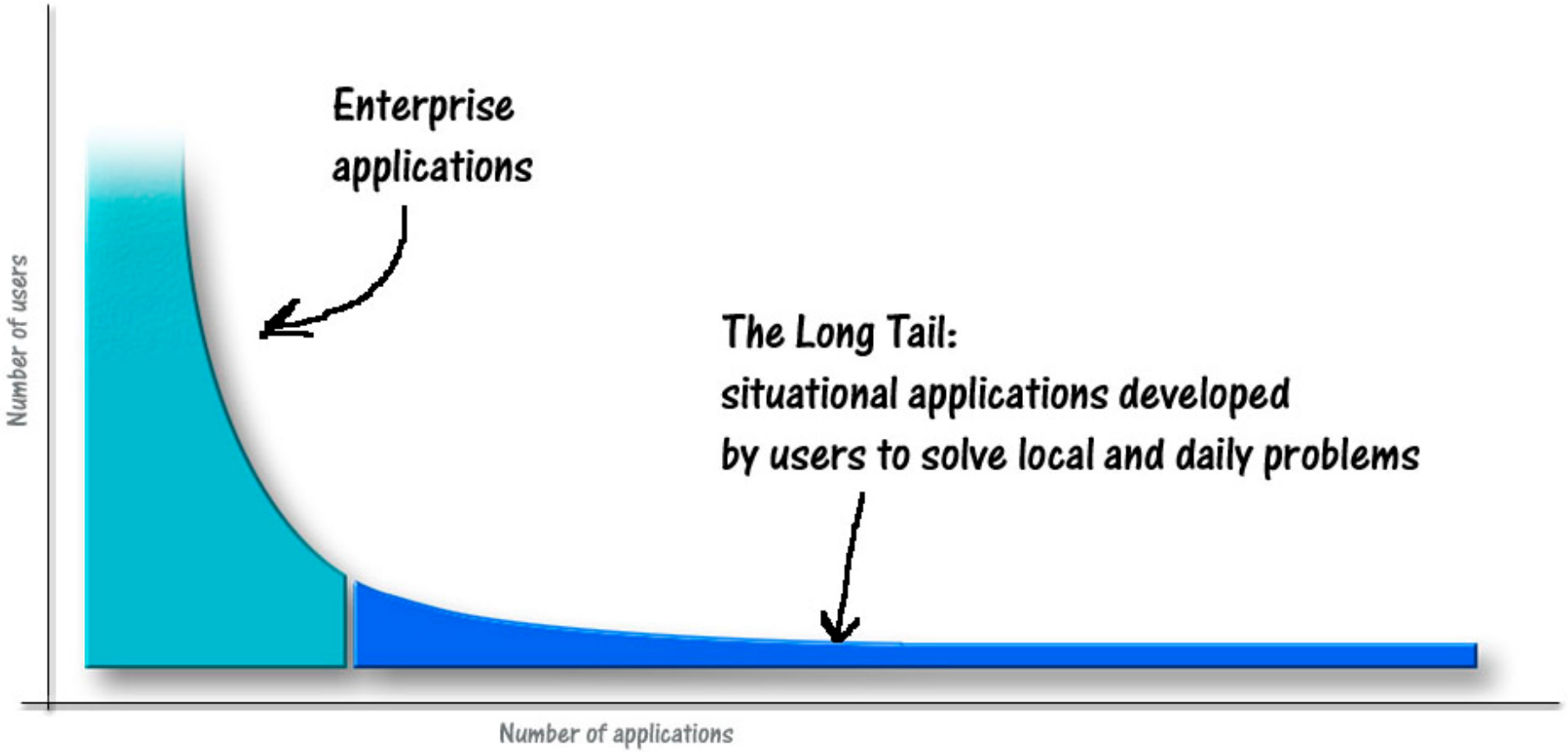


Mashup of internal services and external services

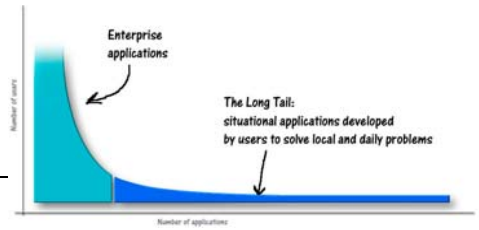


Long tail and situational appls

Long tail applications...



Long tail and situational appls

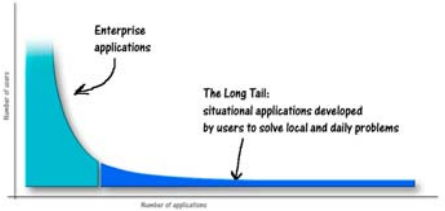


•'Long Tail' **was first coined by Chris Anderson in an October 2004 Wired magazine article to describe certain business and economic models such as Amazon.com or Netflix. Chris Anderson said that businesses with distribution power can sell a greater variety of hard to find items at small volumes than of few popular items at large volumes.**

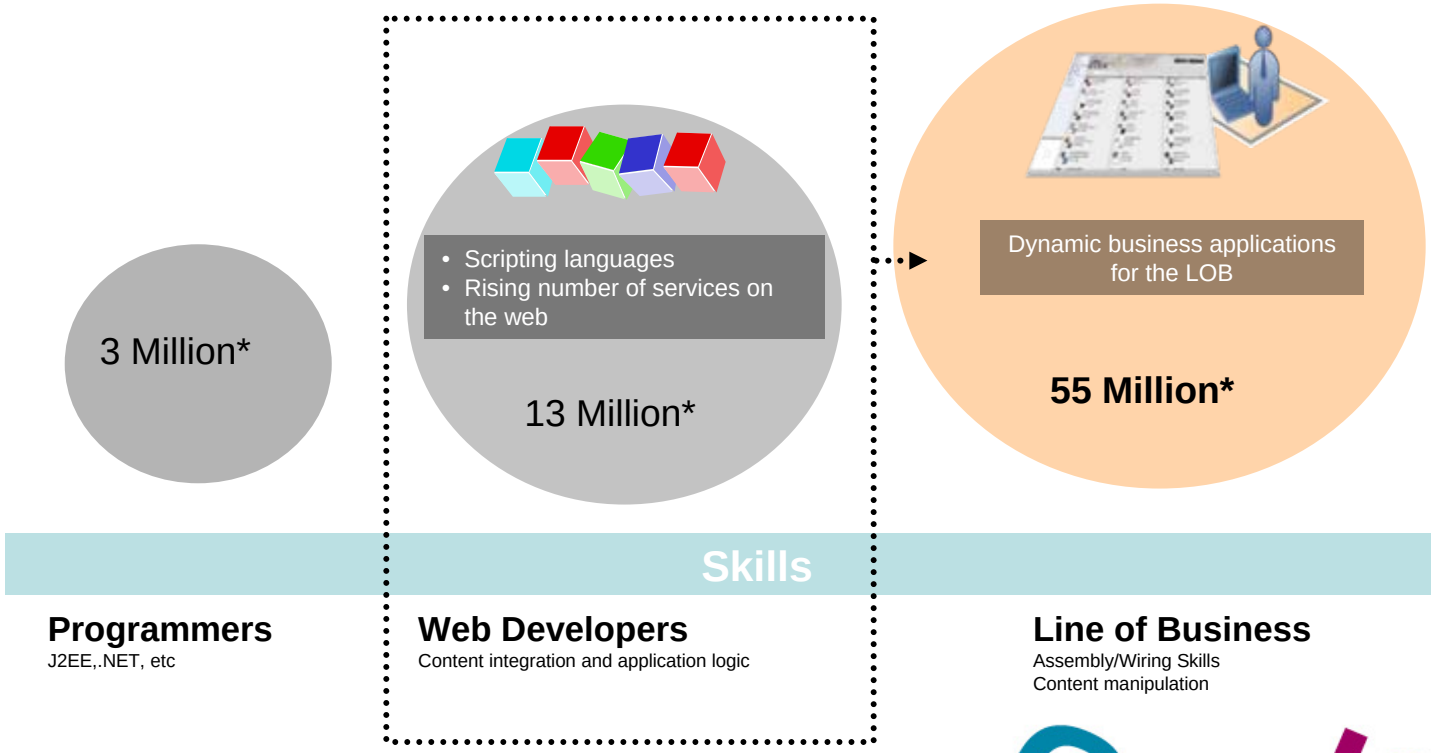
•**Applied to IT world, this concept can be traslated in the following: in addition to mass-market and enterprise applications, there are also** a big number of situational and nich applications developed directly by end users**. We should take them into consideration, due to their number.**

•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 “stuazionali”



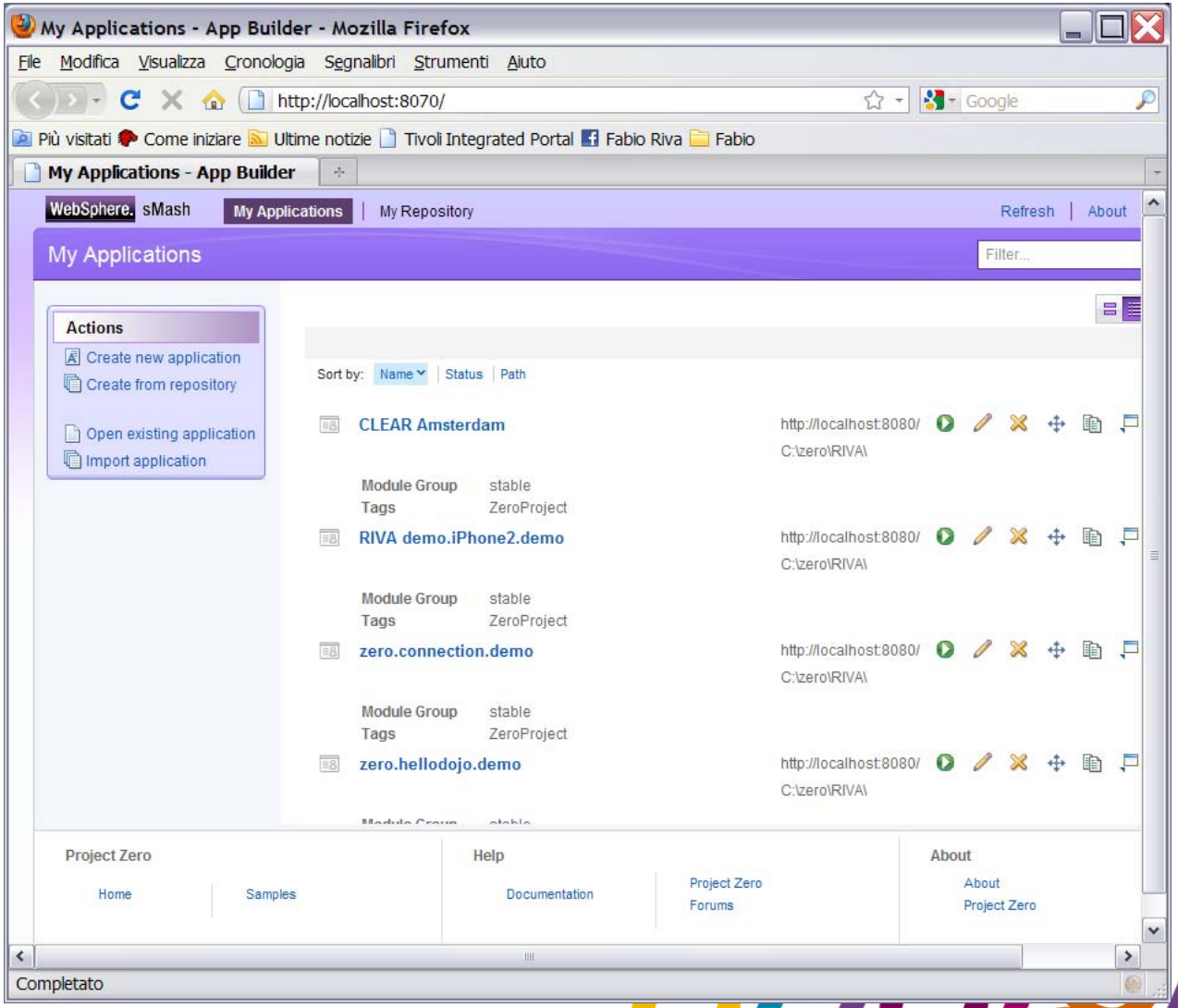
Long tail and consolidation of situational appls

- **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.**

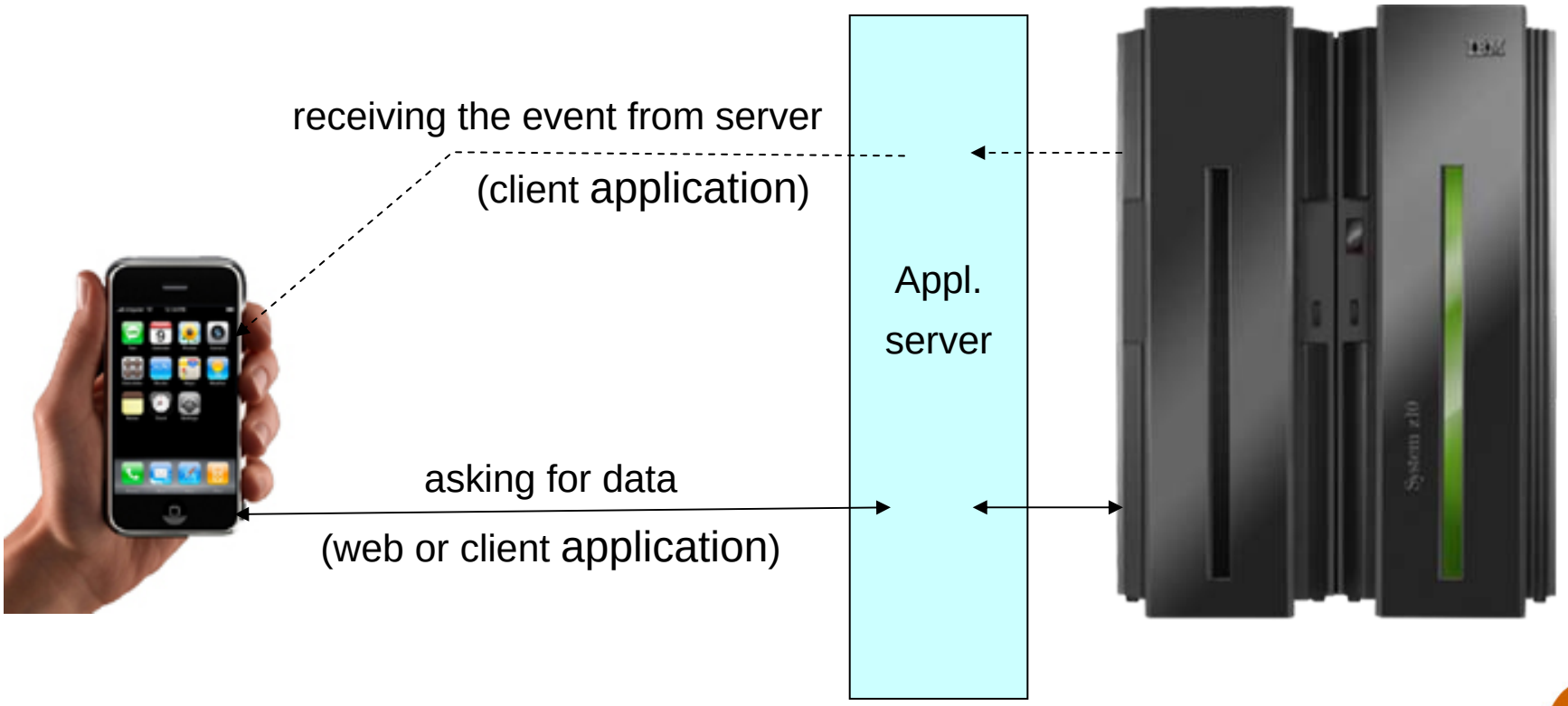
- Consolidation of sMash applications can be realized **either migrating applications to WebSphere Application Server platform (some conversion activities are involved) or directly consolidating sMash appls on z/Series platform, using zLinux (starting from WebSphere sMash V1.1.1).**

WebSphere sMash application page



Extention to smartphone devices

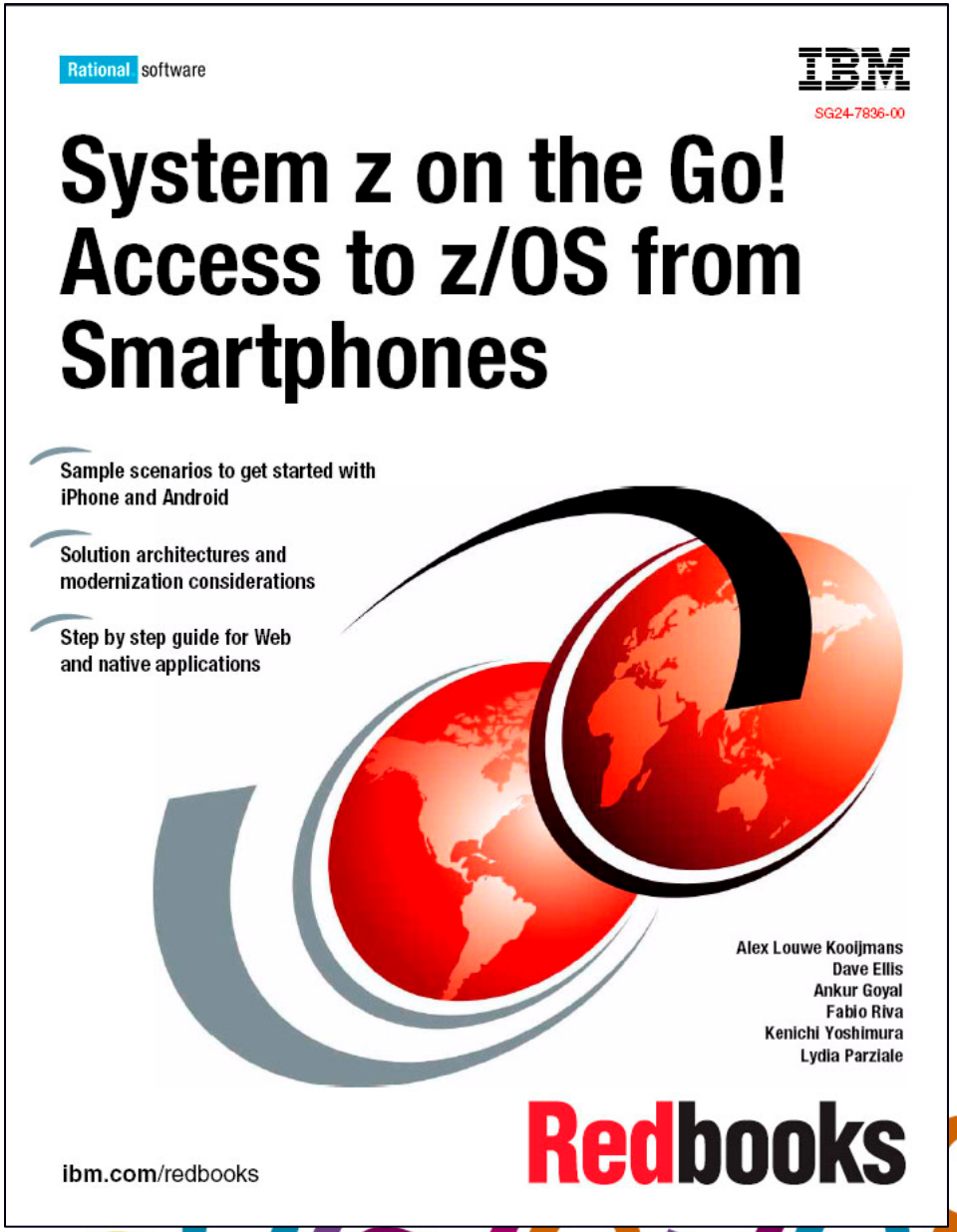
Smartphone devices with mainframes



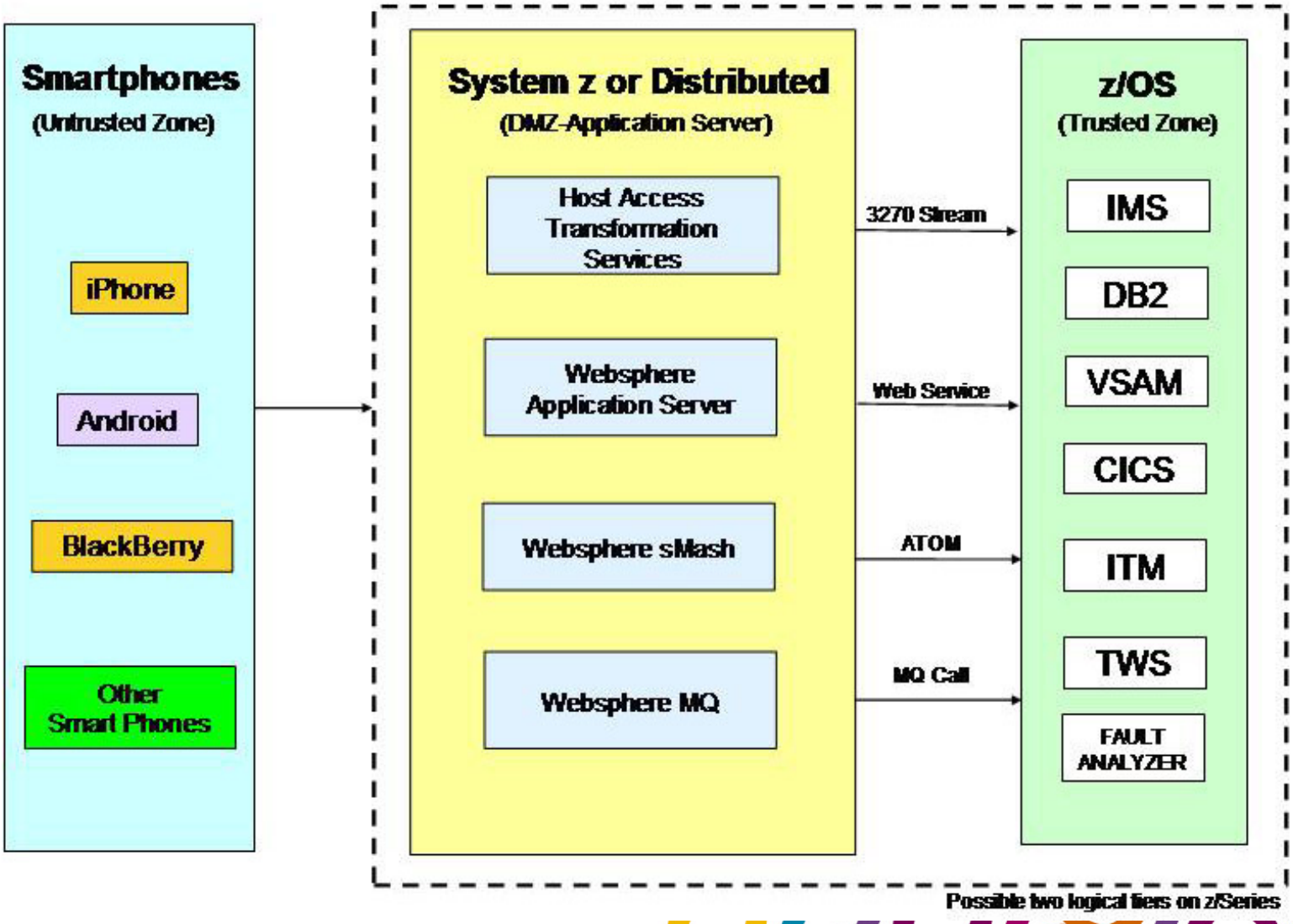
New redbook

IBM

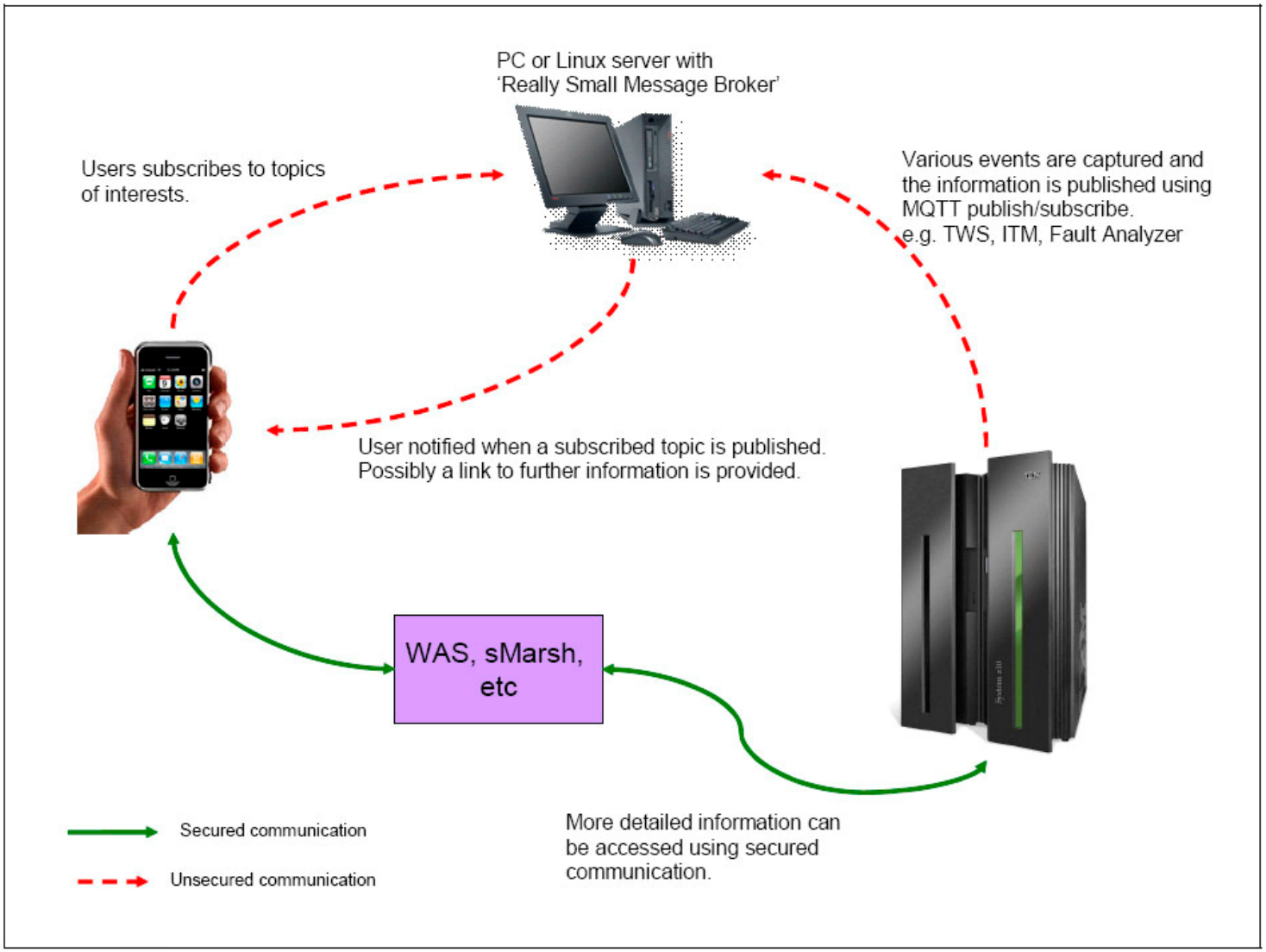
New redbook



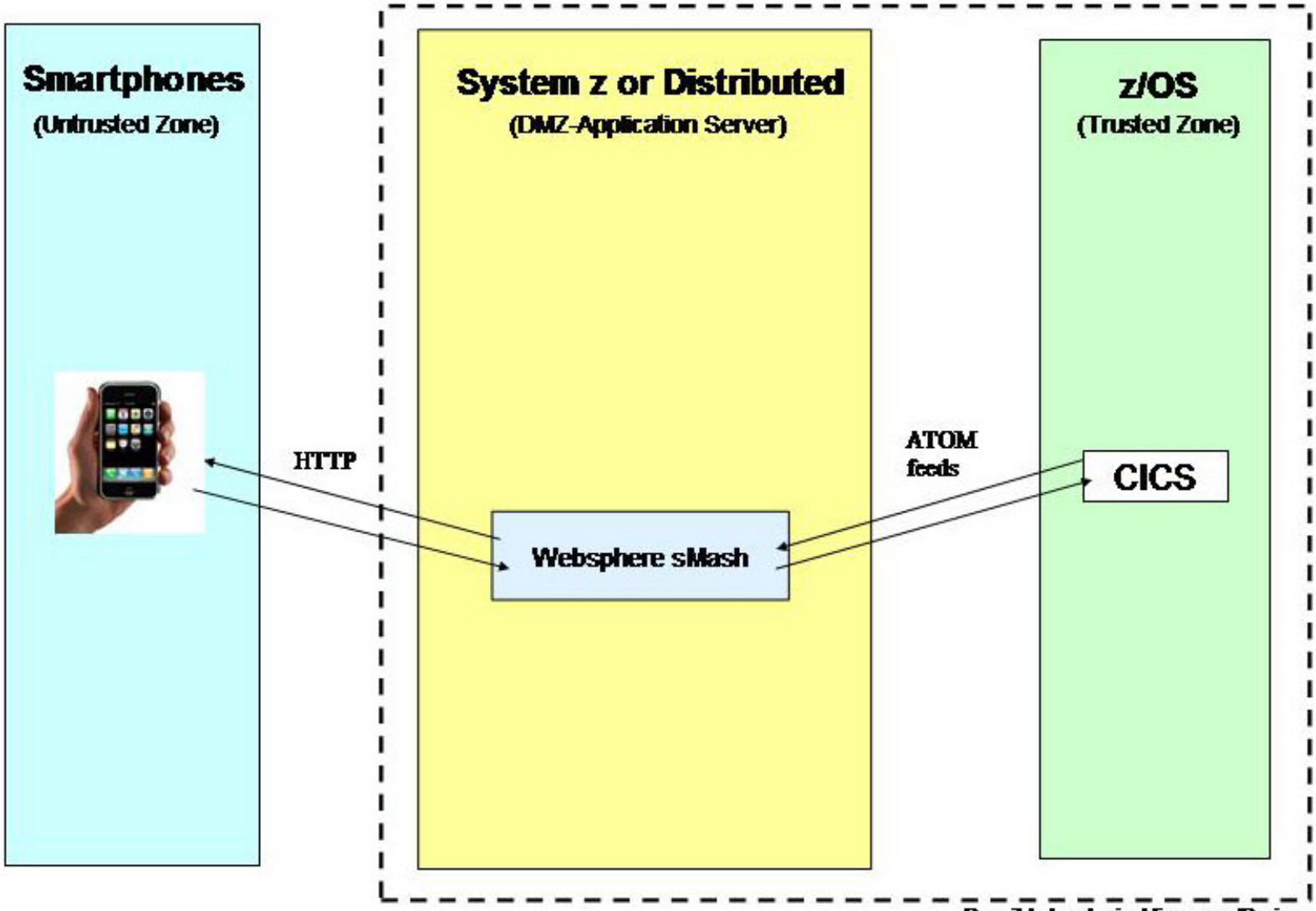
Architecture overview



Using events with smartphone devices

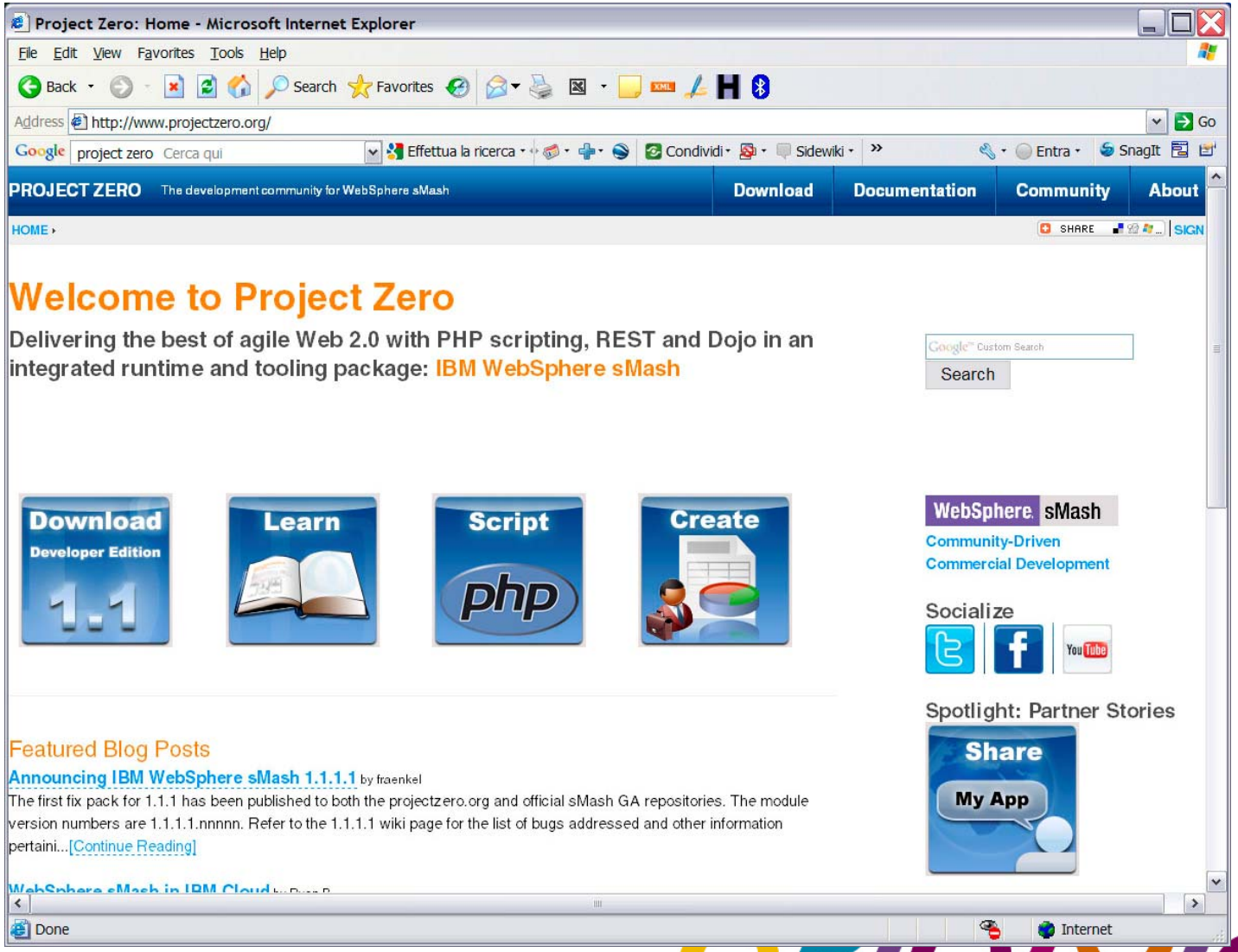


Accessing CICS ATOM feeds with WebSphere sMash

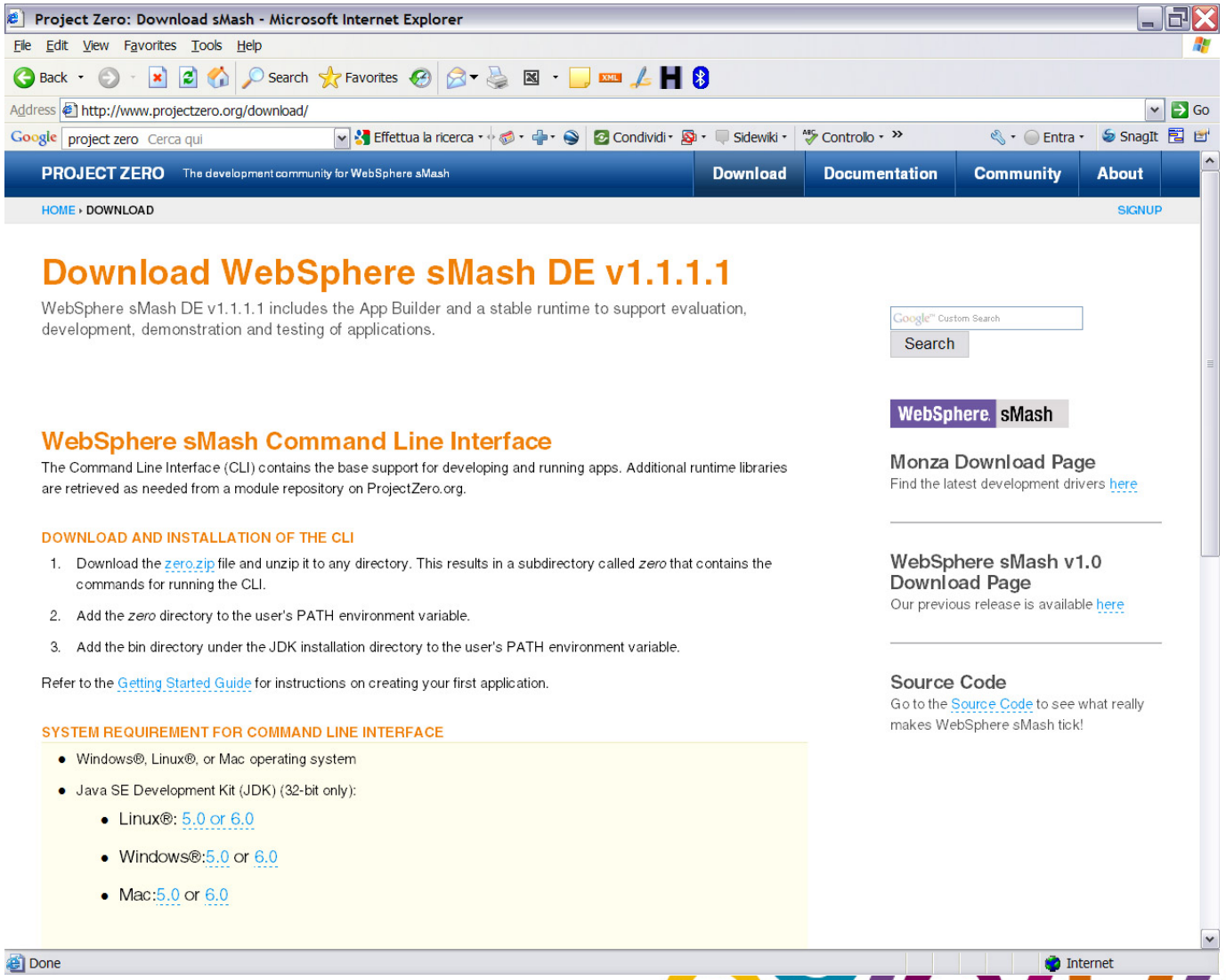


Demo for WebSphere sMash

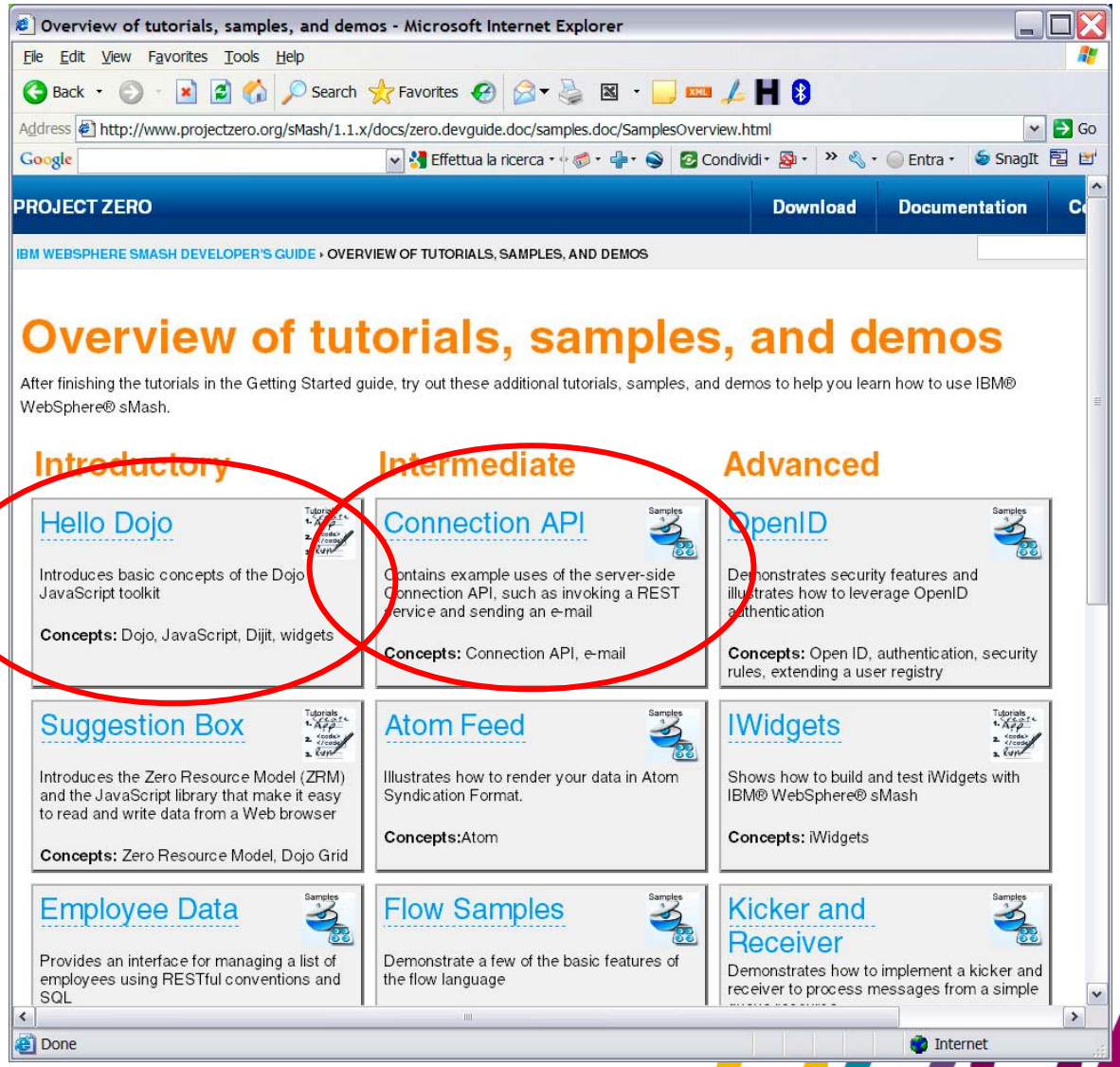
Project zero main web page



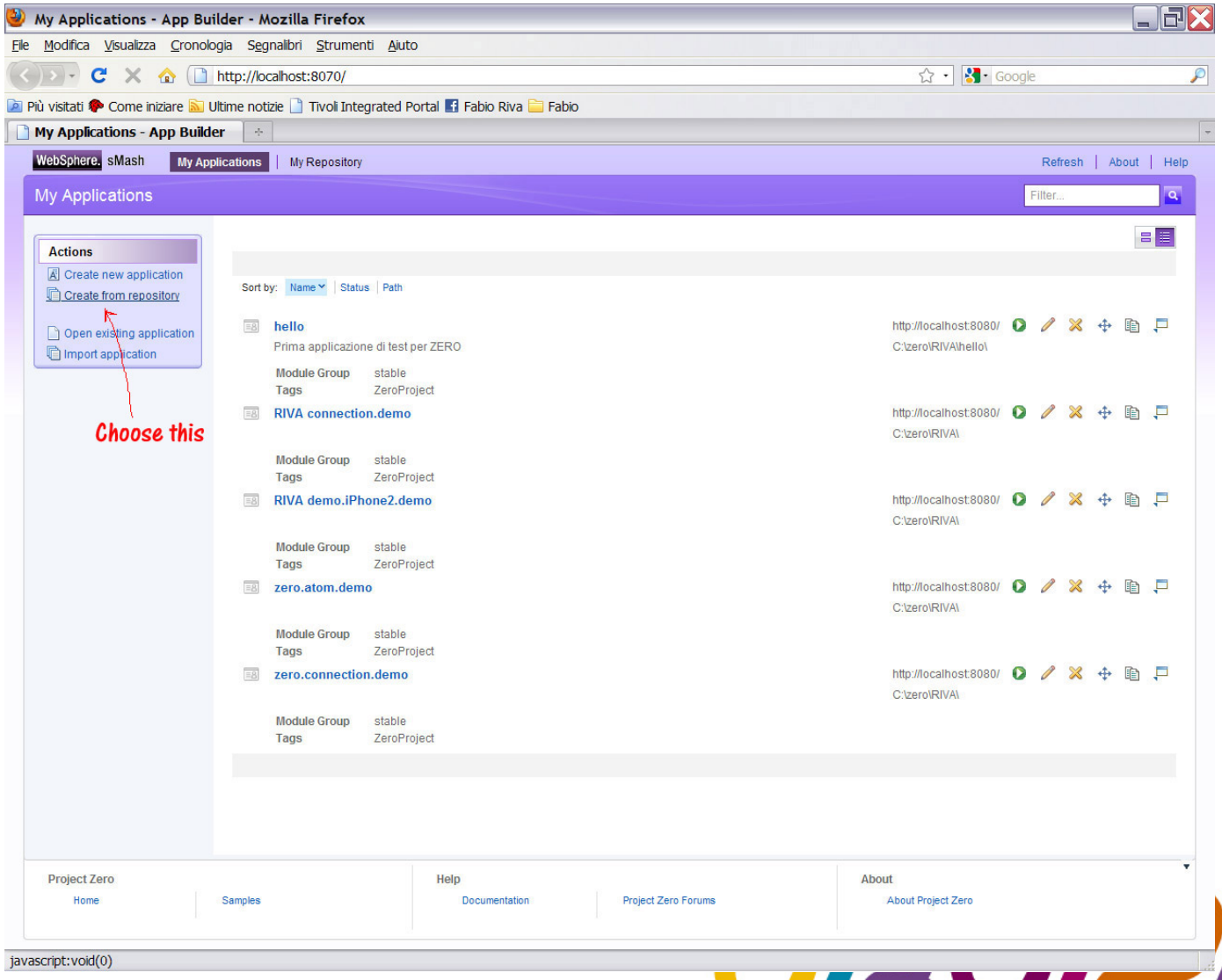
Downloading websphere smash



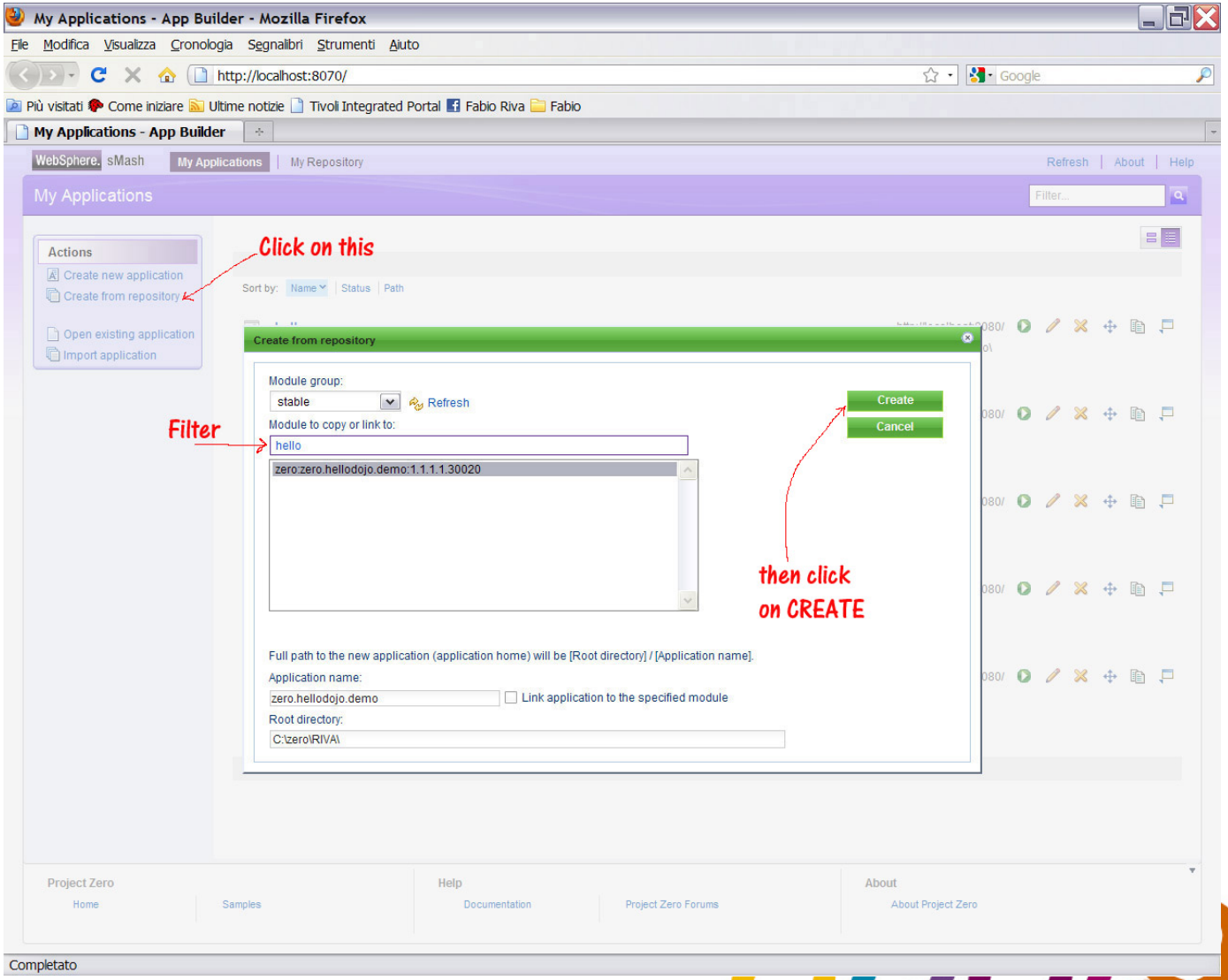
Overview of samples



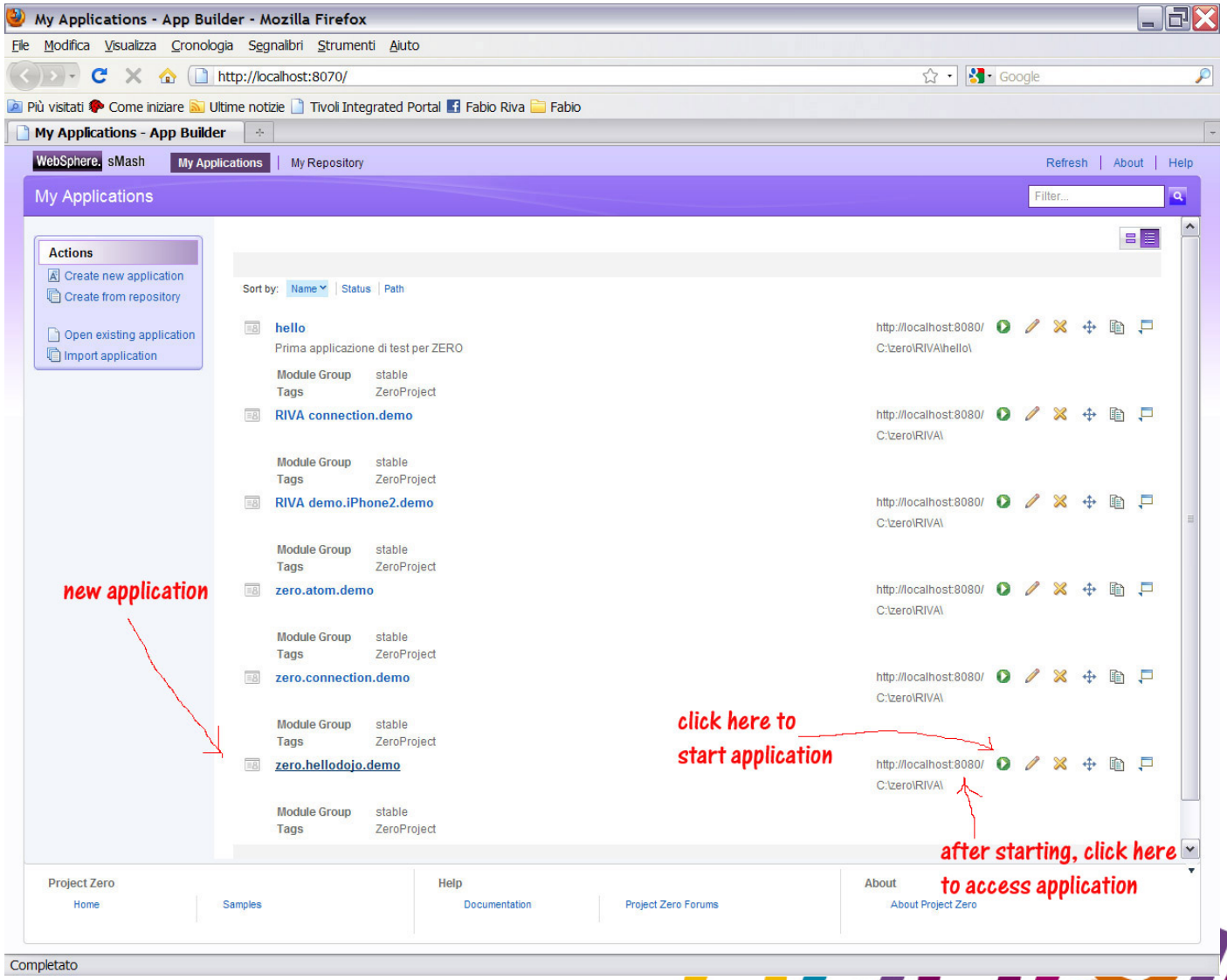
Downloading sample



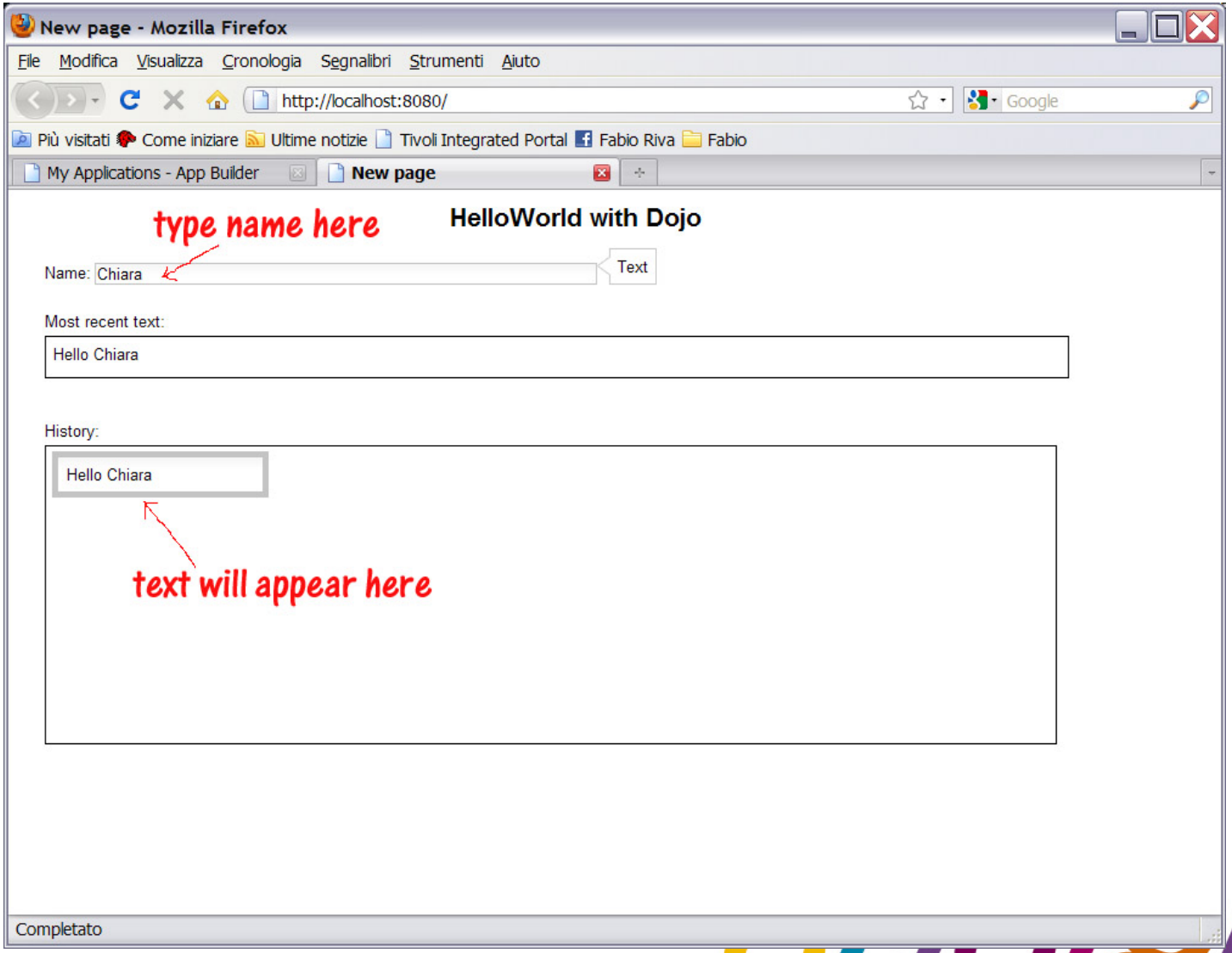
Adding sample



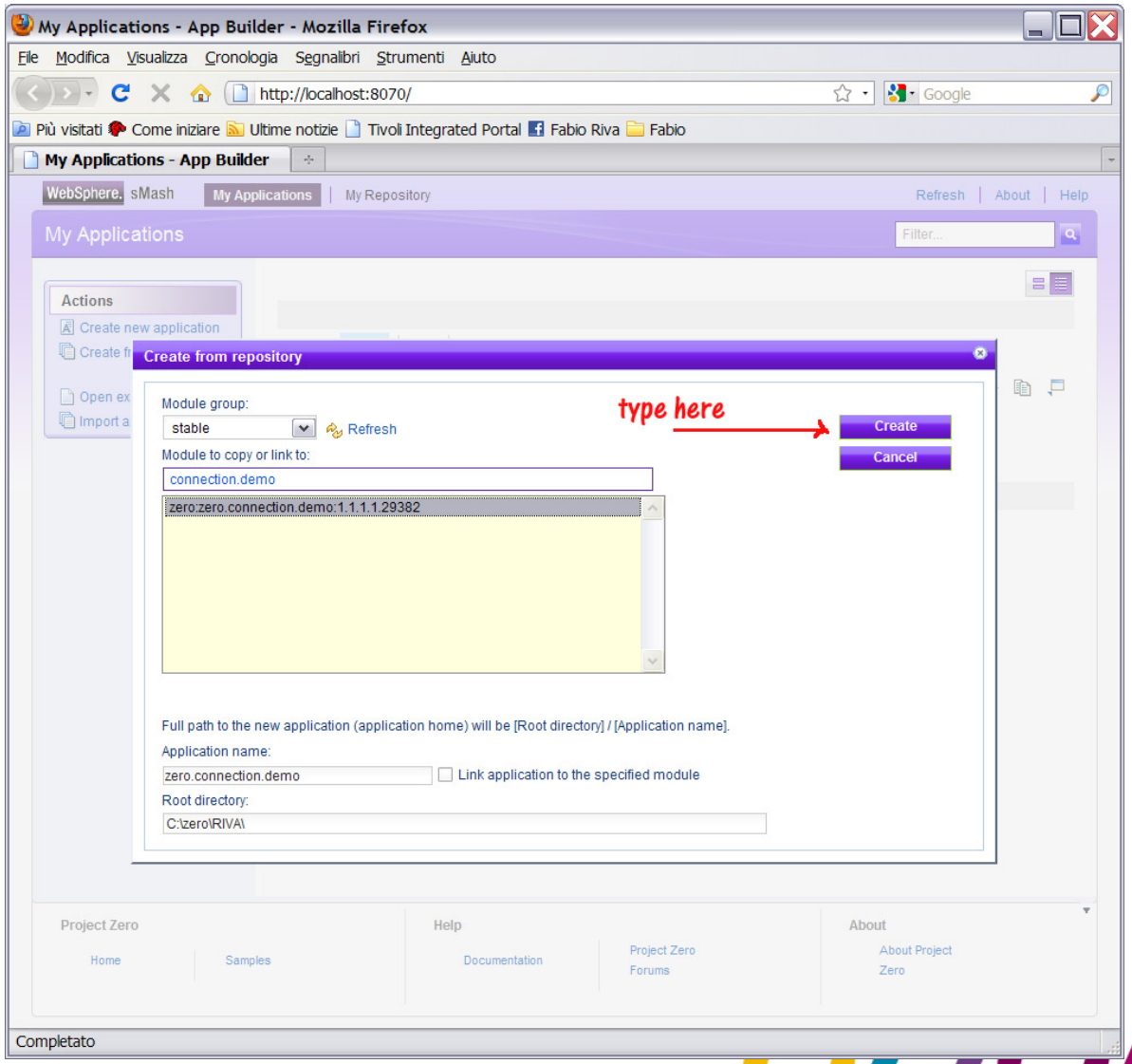
Websphere sMash application page



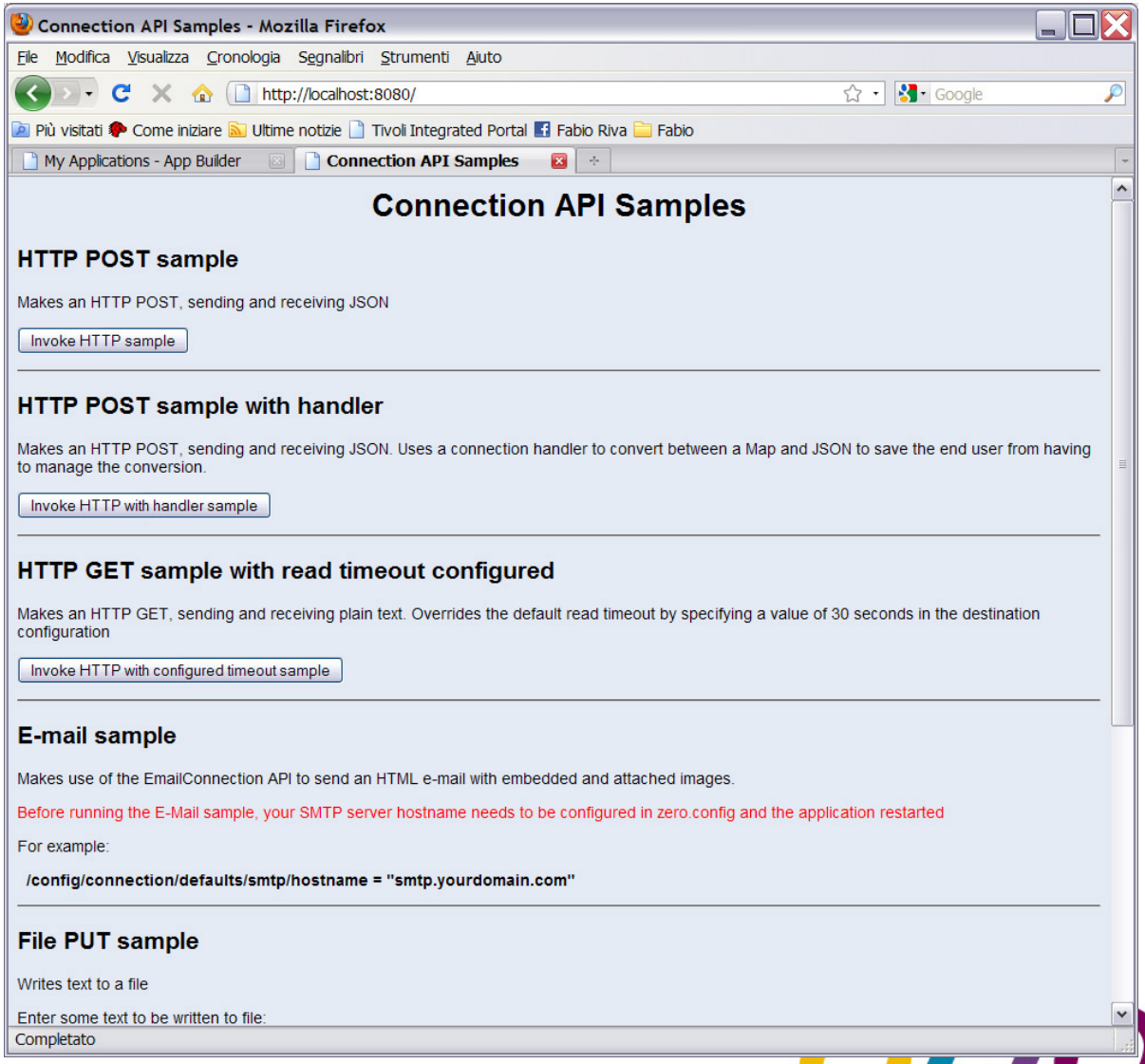
Testing sample application



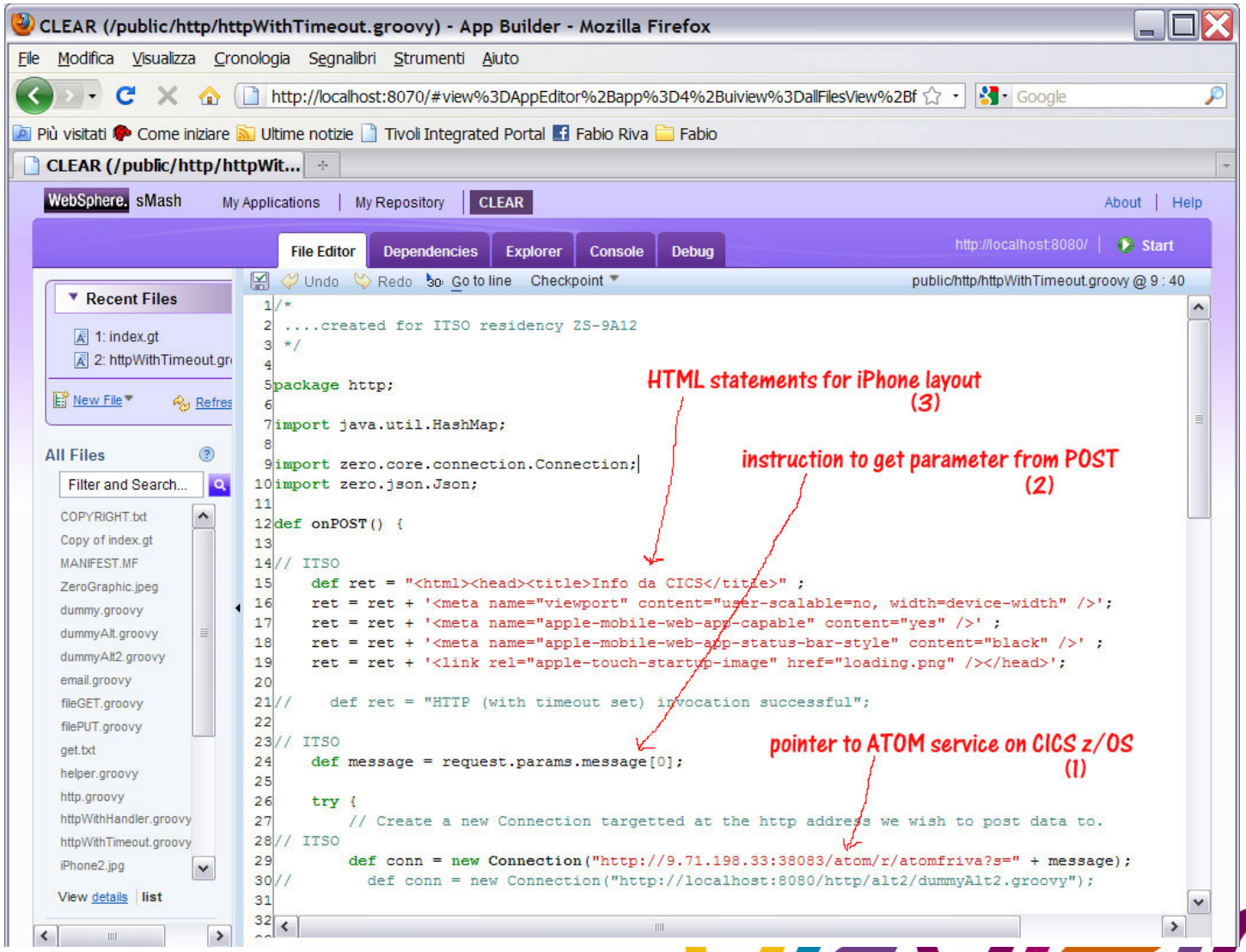
Adding connection sample, in order to modify it



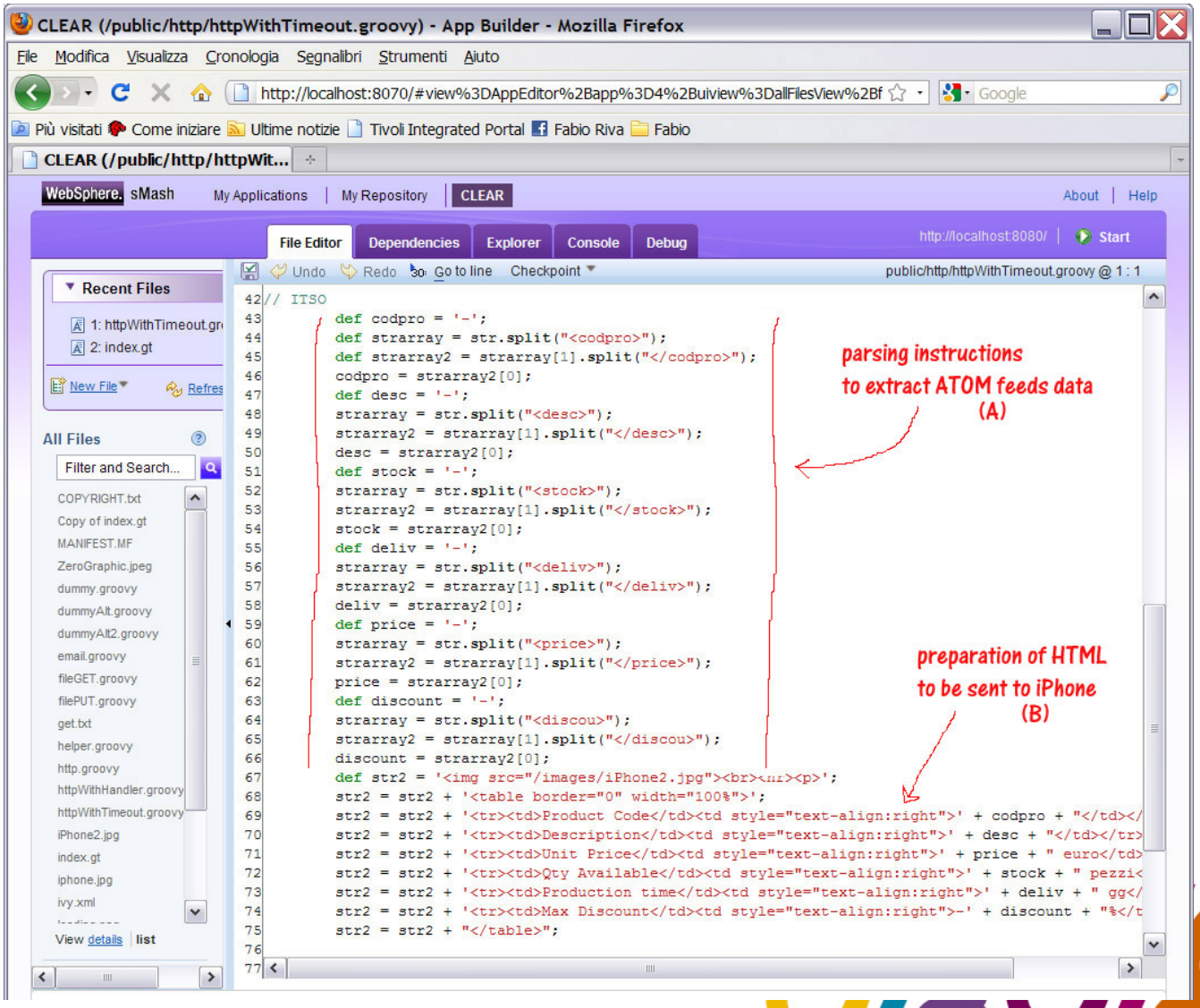
Connection sample



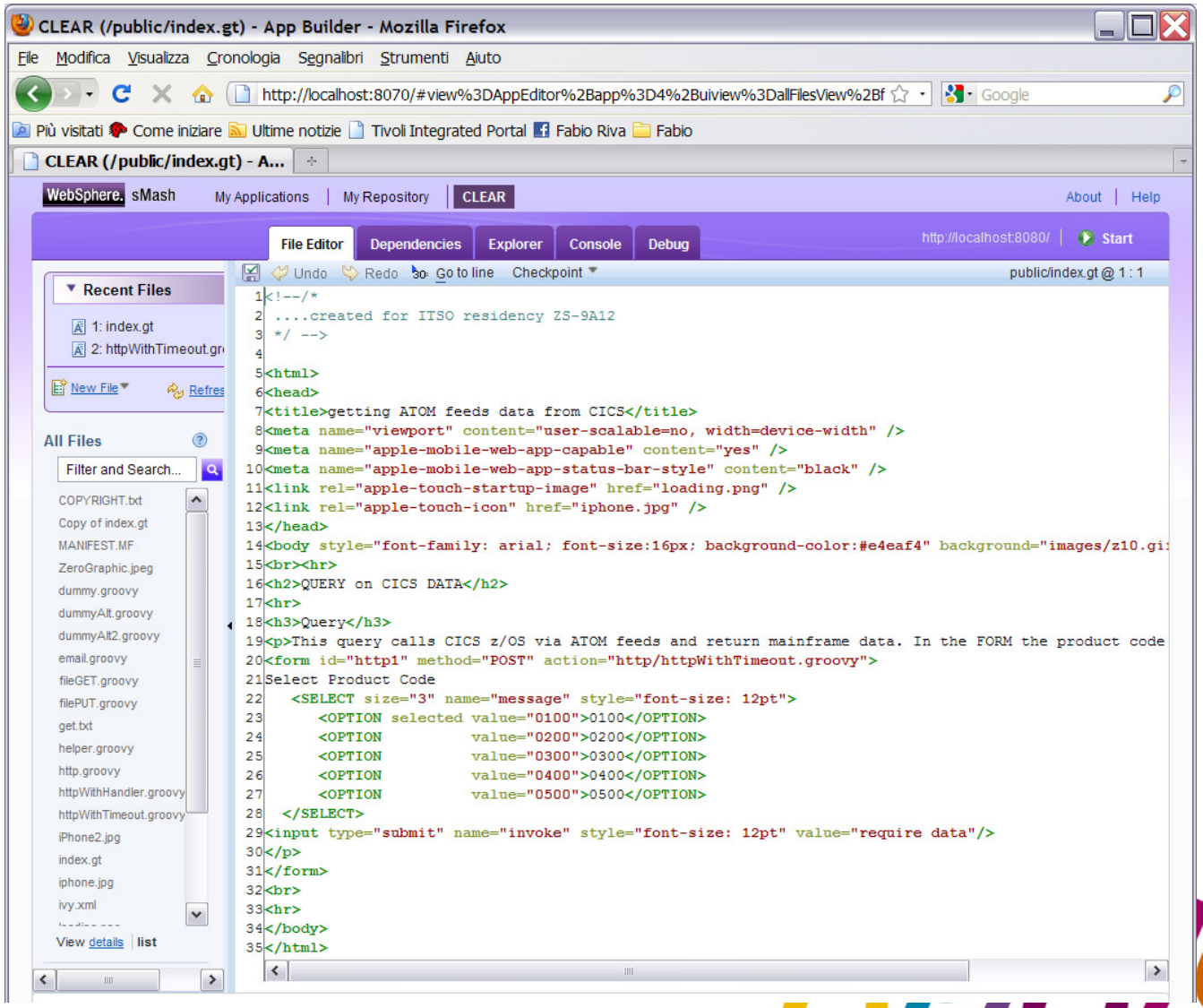
Modifying it...



Modifying it...



Inserting FORM



My demo situational appl

APPLICATION ICON

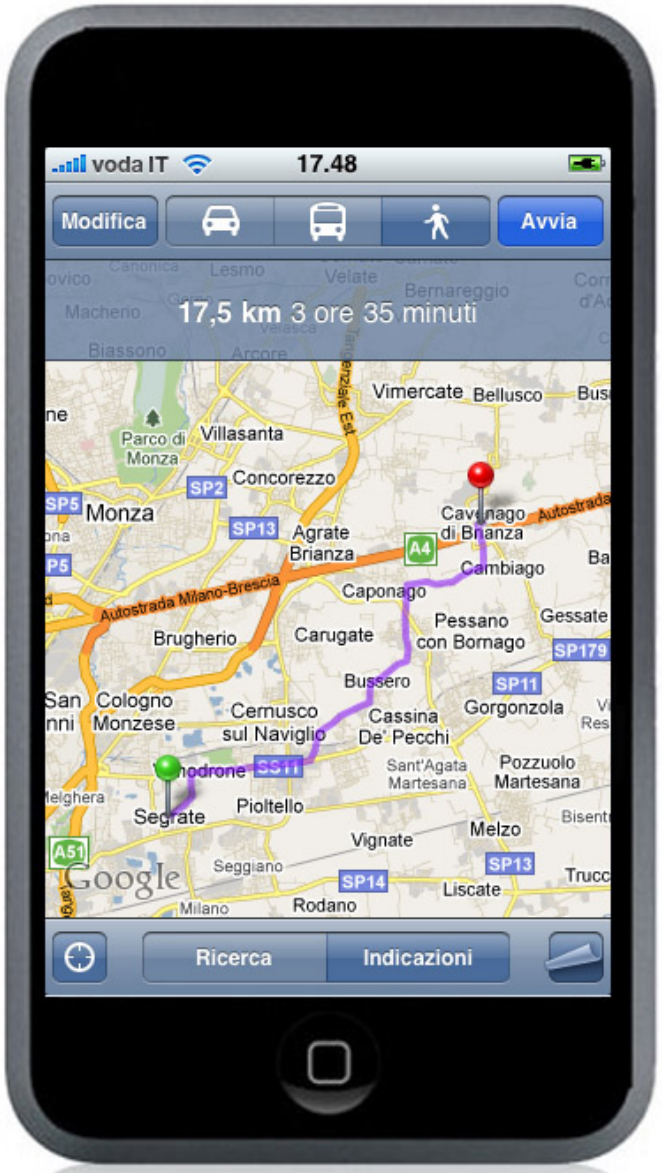
....CICS query


voda IT
21.52
richiesta viene fatta direttamente al CICS di produzione per verificare in tempo reale lo stato giacenze e i valori aggiornati dei dati commerciali.
Seleziona codice prodotto
0200
richiedi disponibilita'
Prec.
Succ.
Riemp. autom.
Fine
0100
✓ 0200
0300
0400


voda IT
21.52
CICS
Codice prodotto 0200
Descrizione Bicicletta Uomo 28
Prezzo unitario 449.00 euro
Disp. a magazzino 550 pezzi
Tempo di produzione 15 gg
Sconto possibile - 10%

....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.**