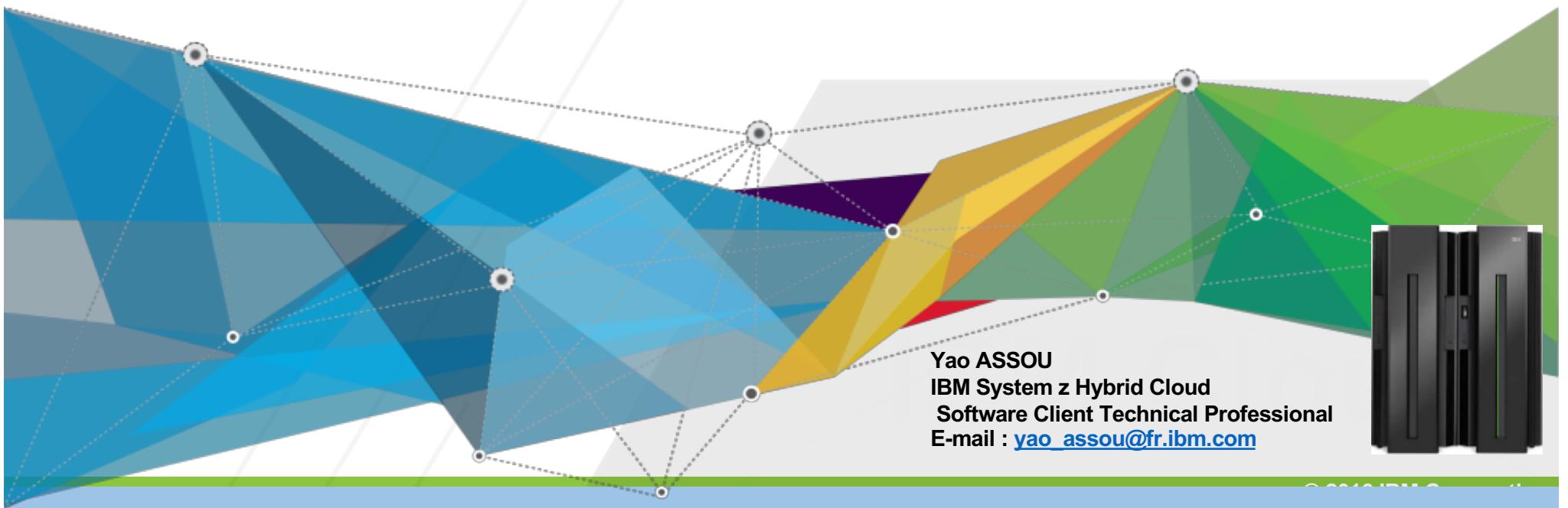


# z Systems Platform

## Integration in Digital Economy



# AGENDA

- 1
- 2
- 3
- 4
- 5

❑ What is “Transactional System”

❑ What is “CICS” ?

❑ What is WebSphere Application Server → (WAS) ?

➤ Focus on WAS z/OS

❑ What is Message Management ?

➤ Focus on IBM MQ Series

❑ Open z Systems « assets » for New Business

➤ A big picture

# AGENDA

1

□ What is “Transactional System”

# Besoins Métiers : Exemple de scénario type

Une société commerciale célèbre sur le marché du multimédia souhaite ouvrir ses collections à une clientèle privée en avant-première .

Les clients auront les possibilités suivantes :

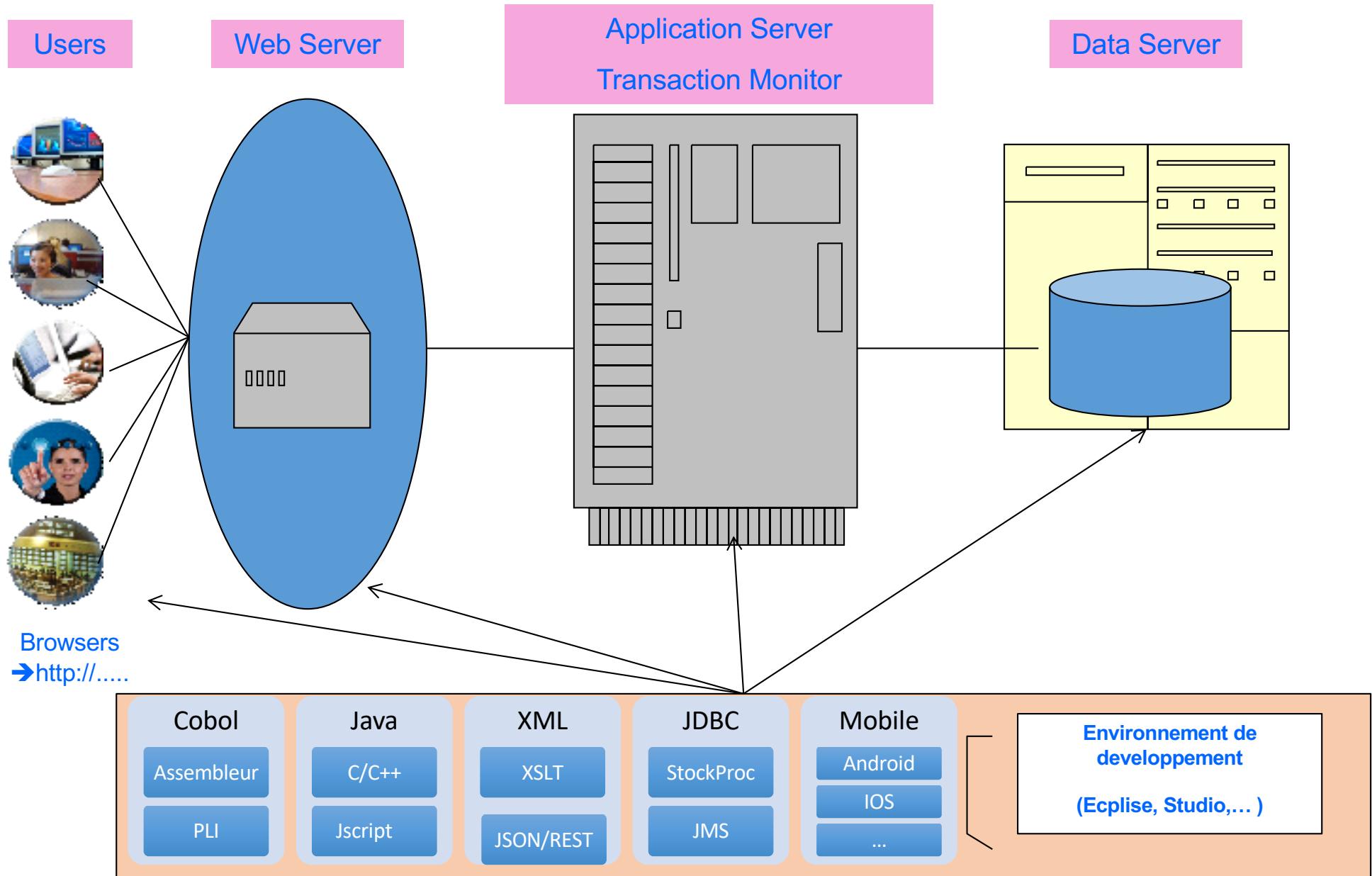
- un **accès privilégié** sur le site en fonction de leur profil
  - La possibilité de négocier en **direct** avec le service clientèle pour demande de réduction, réclamation
  - La possibilité de **donner un avis** sur chaque article proposé
  - accès via tout type de support
    - **Web**
    - **Tablet**
    - **Smartphone**
    - **Call center**
    - ..
  - Les **transactions** seront protégées
  - Les paiements **sécurisés**
  - La **cohérence** de chaque transaction sera garantie ➔ gestion des **pannes sans affectation** d'une transaction en cours.
  - les transactions seront **sauvegardées** en vue d'une analyse pour un ciblage personnalisé à chaque connexion
  - La **haute disponibilité** du site
- ....

## Il est demandé :

- De proposer un **inventaire des ressources** matérielles et humaines indispensables pour la construction de ce site
- Les **compétences** nécessaires pour sa réalisation
- Les **outils indispensables**
- De construire une vue générale de **l'architecture du site**.

# Architecture standard et basique

IBM



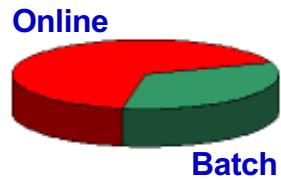
# Batch processing is evolving



Windows of time which used to be dedicated to batch processing are shrinking, while batch workloads continue to grow as online transactions increase



In the past ...



Today ...



**24 x 7 x 365 Access**

**Users of your online systems expect availability at all hours.**  
**Users from other parts of the world means availability is expected around the clock.**



**Mobile Users**

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

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

**The need to perform batch concurrent with online has emerged.**

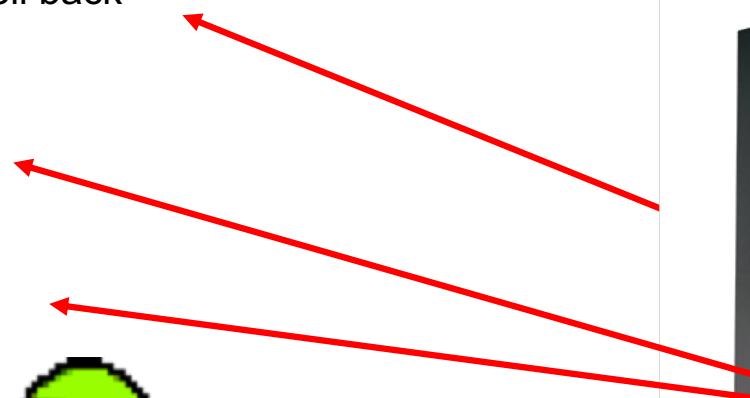
# Batch and Online Transactions

	Batch Processing	Online Transaction Processing
Initiation	Scheduled for specific time Job Control Language	Initiated by end user when services required
Input	Program Input is batch of transactions	Input supplied by end user for single request
Output	Delayed output	Immediate output
Resource access	Usually exclusive file control	Shared file, exclusive record control

# Transactional systems: terminology

IBM

- Commit and roll back

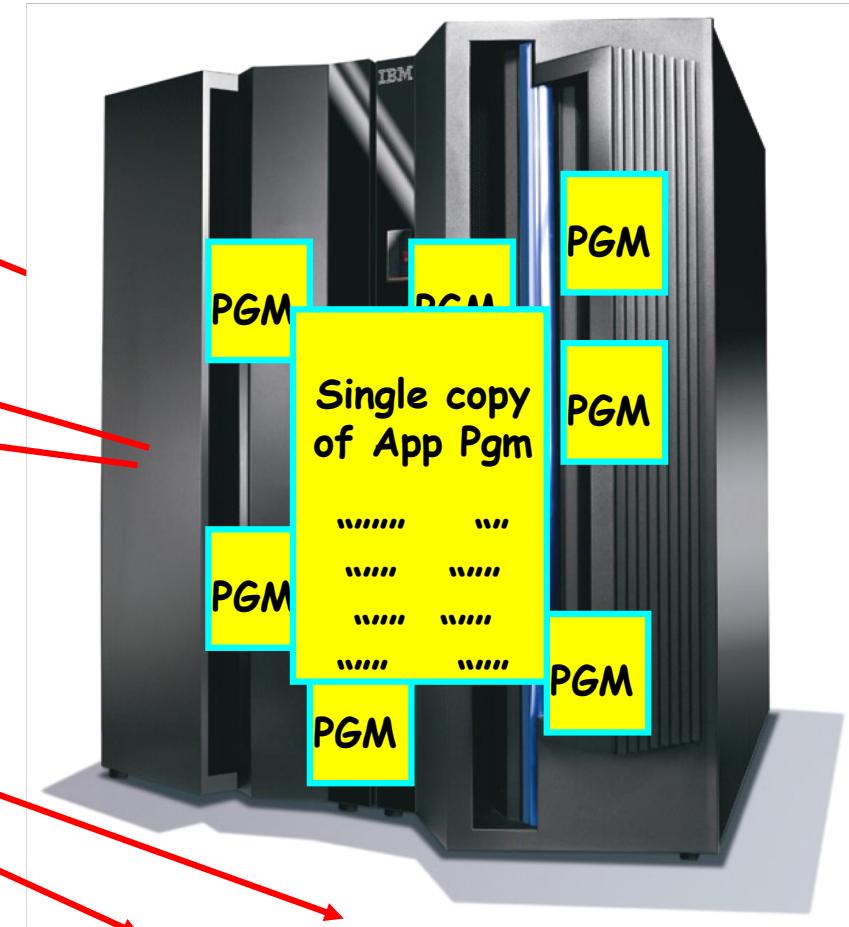


- Multitasking

- Multithreading

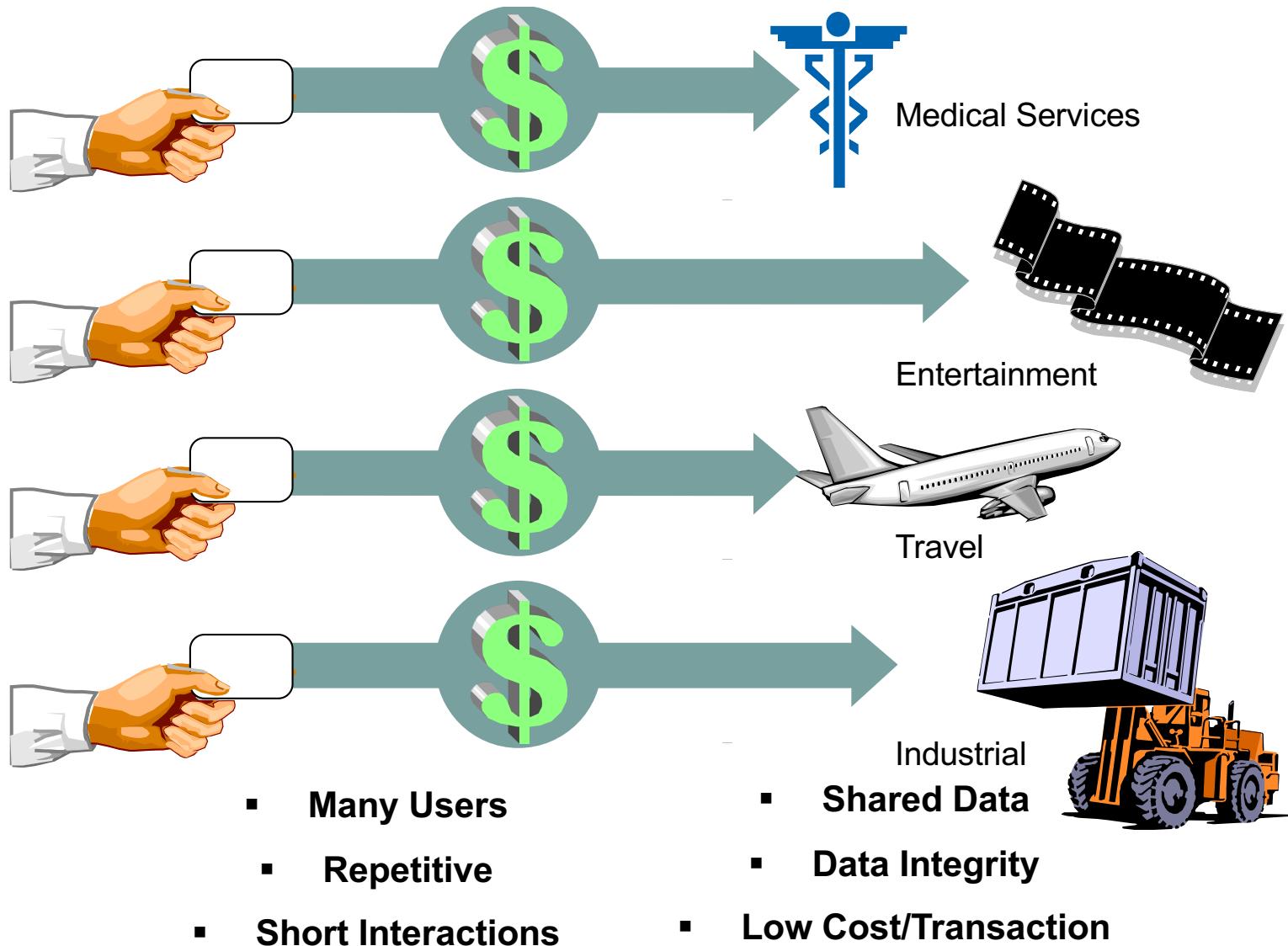
- Thread

- Reentrancy

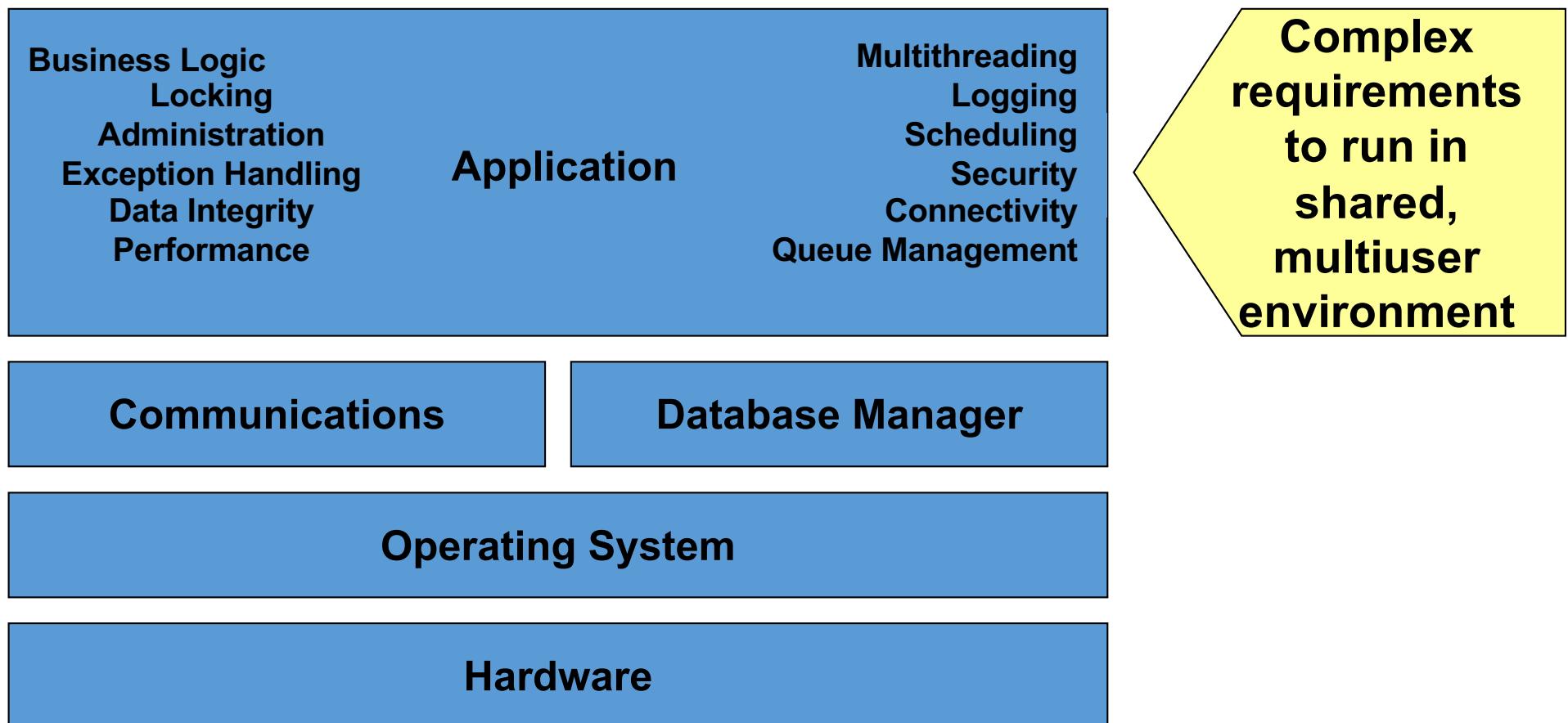


# Business Transactions

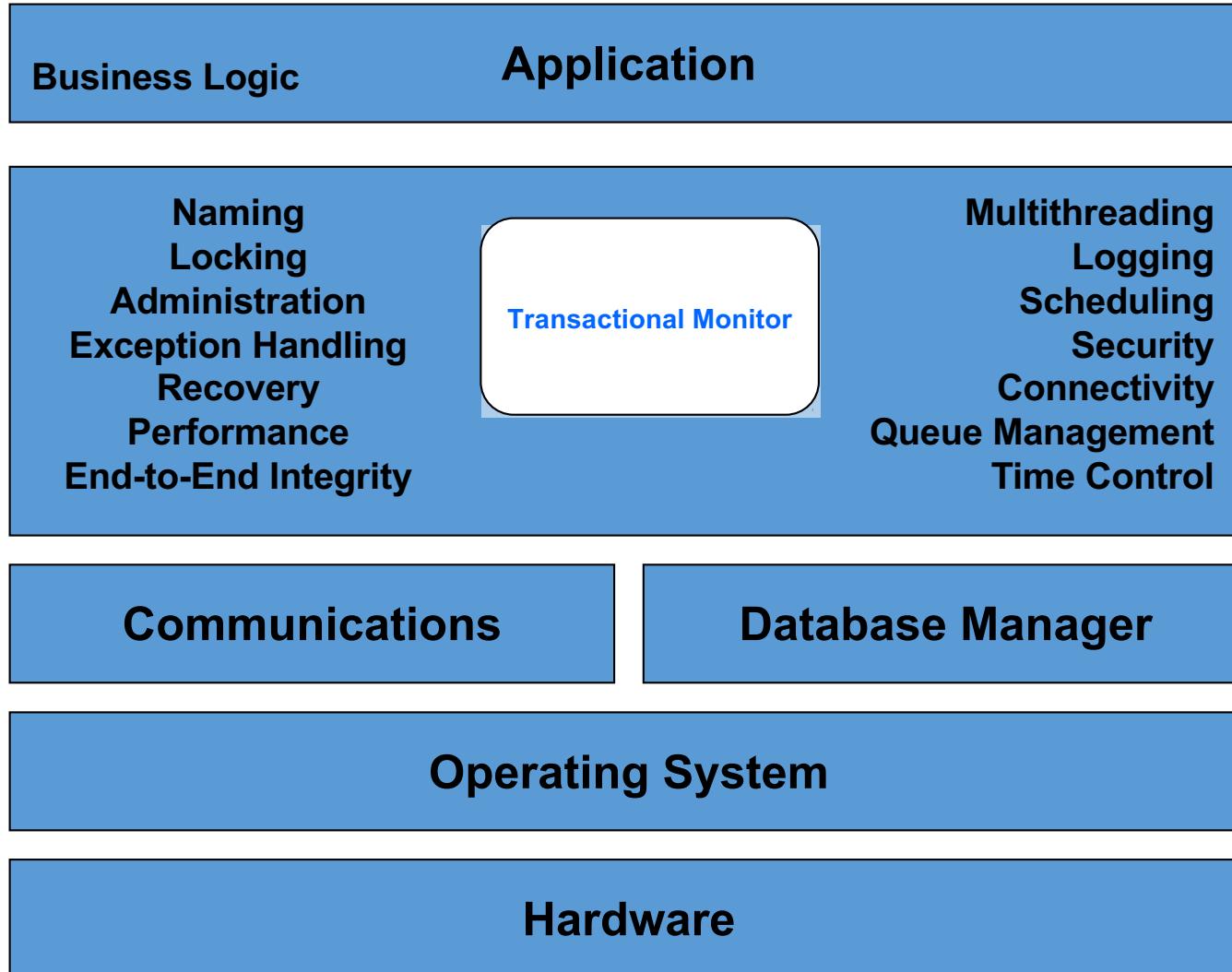
IBM.



# Business Application Requirements



# Business Application Requirements



# AGENDA

2

□ What is “CICS” ?

# CICS – what does it stand for?

IBM



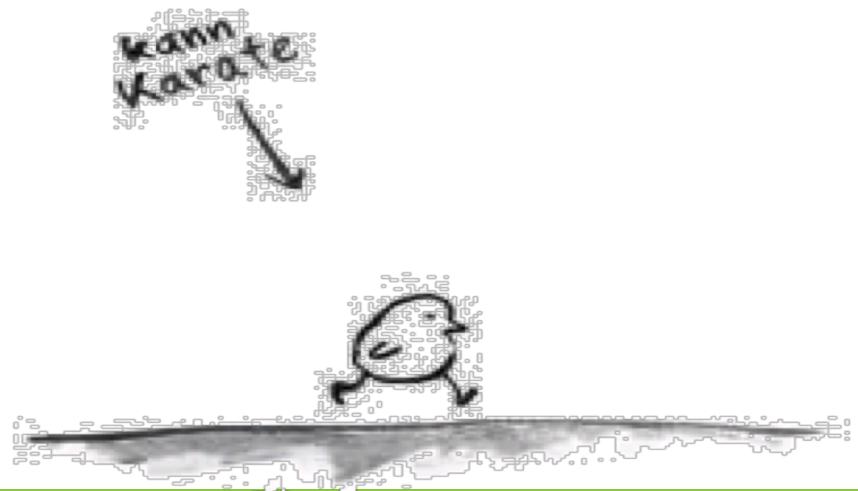
- Canadian Institute for Climate Studies
- Cypriot Internet Chess Server
- Crotoxin Inhibitor from Crotalus Serum
- Conference Internationale Catholique du Scoutisme
- Center for Information and Crisis Services
- Constantly IMS Comes Second ☺
- Customer Information and Control System

# How it is pronounced ?

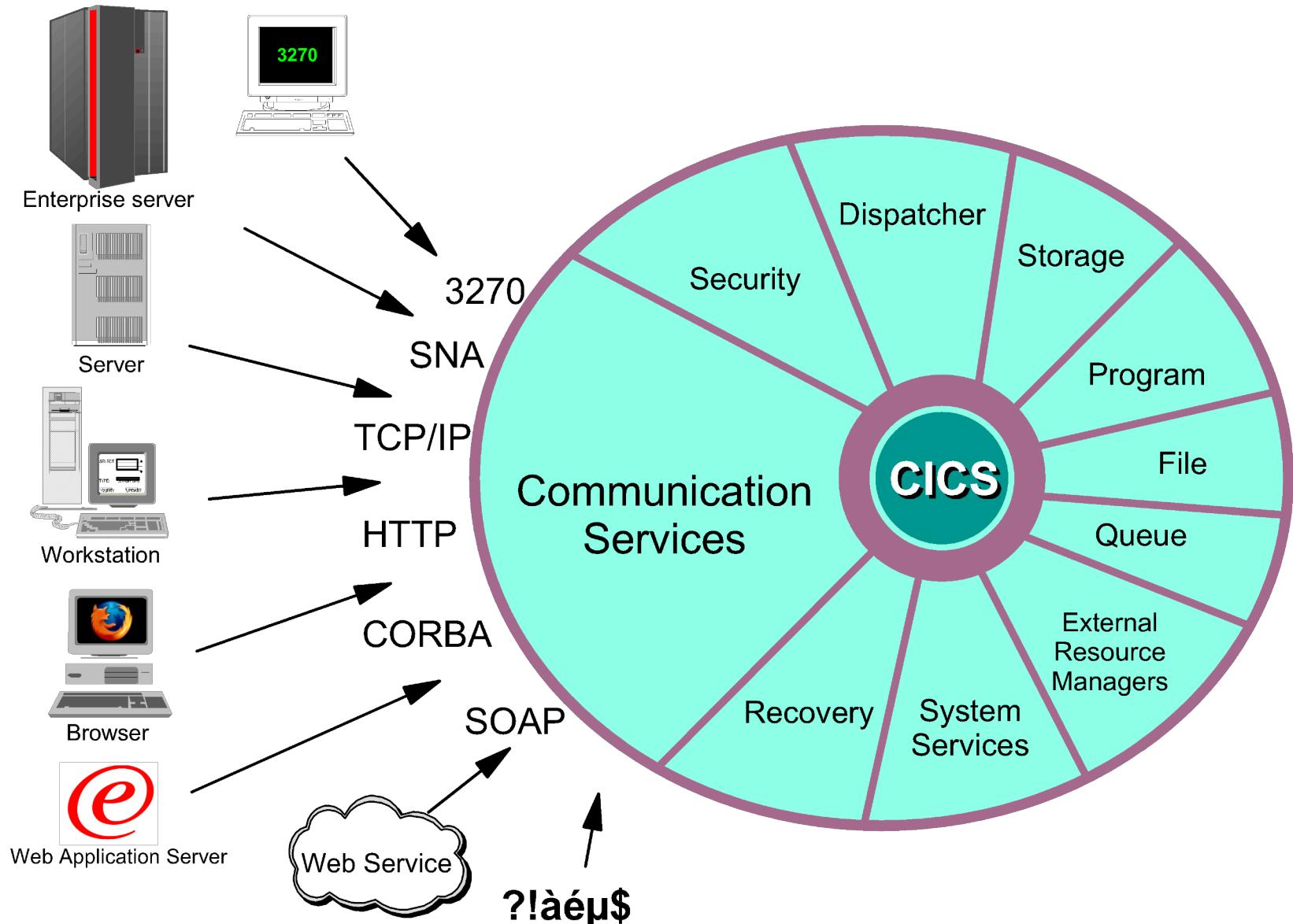
IBM.



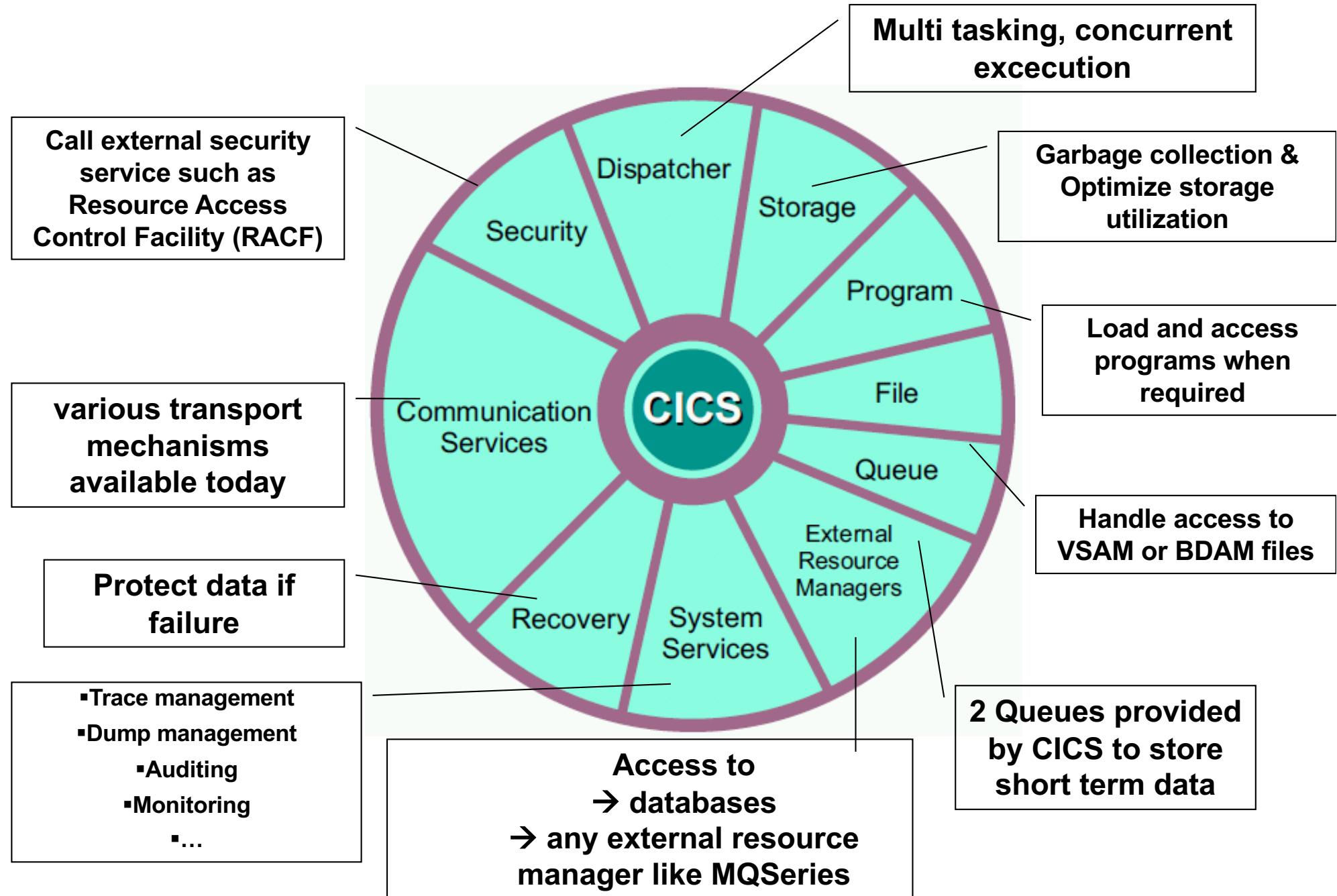
- In Britain, Canada, Australia, France, Belgium and some other countries, CICS is pronounced the same as the word *kicks*.
- US - usually pronounced by reciting each letter (C-I-C-S)
- Germany - pronounced *zicks* and, less often, *kicks*
- Brazil, Peru and Mexico - pronounced *sicks*
- Spain - pronounced *thicks*
- Italy - pronounced *chicks*



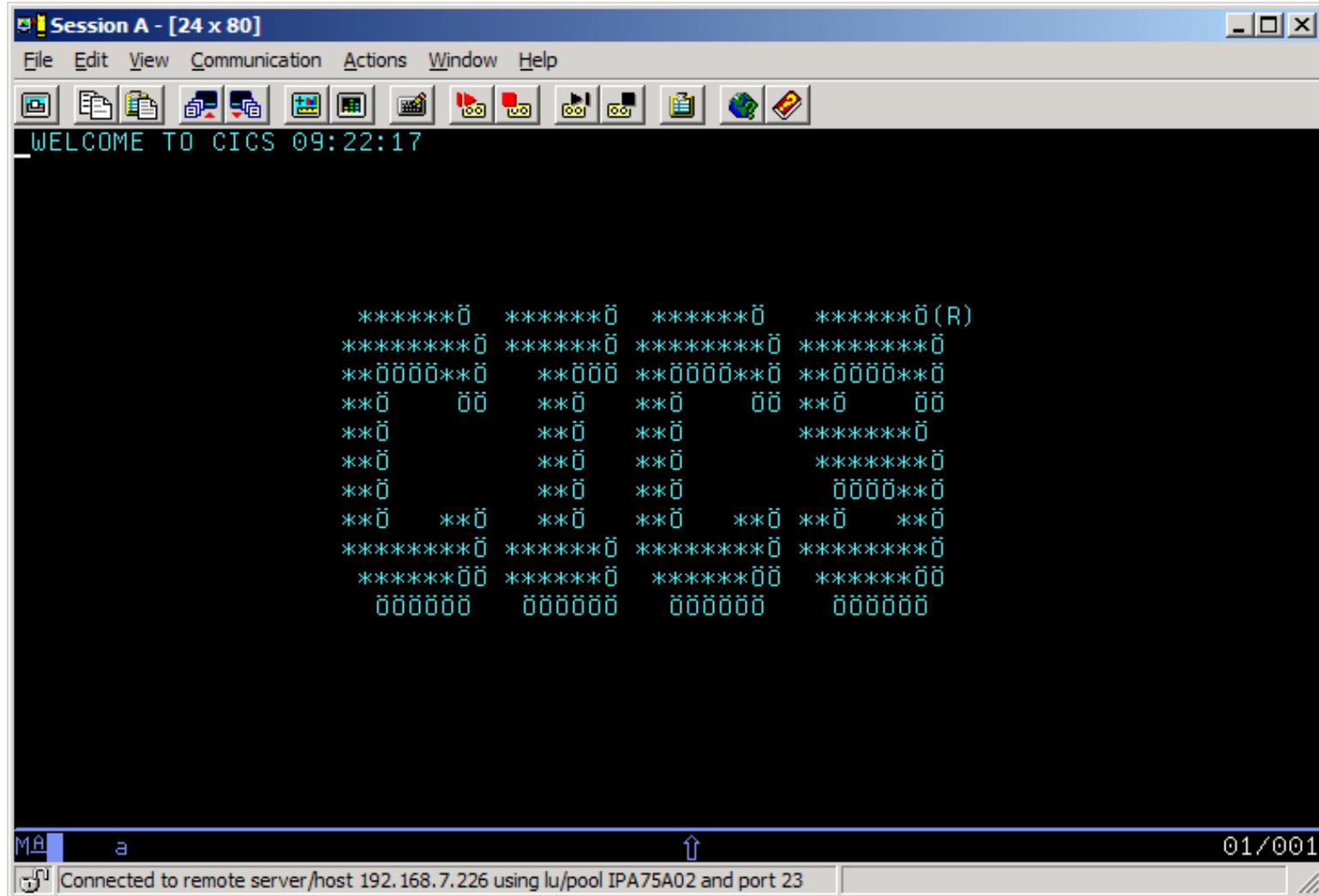
# CICS Management Services



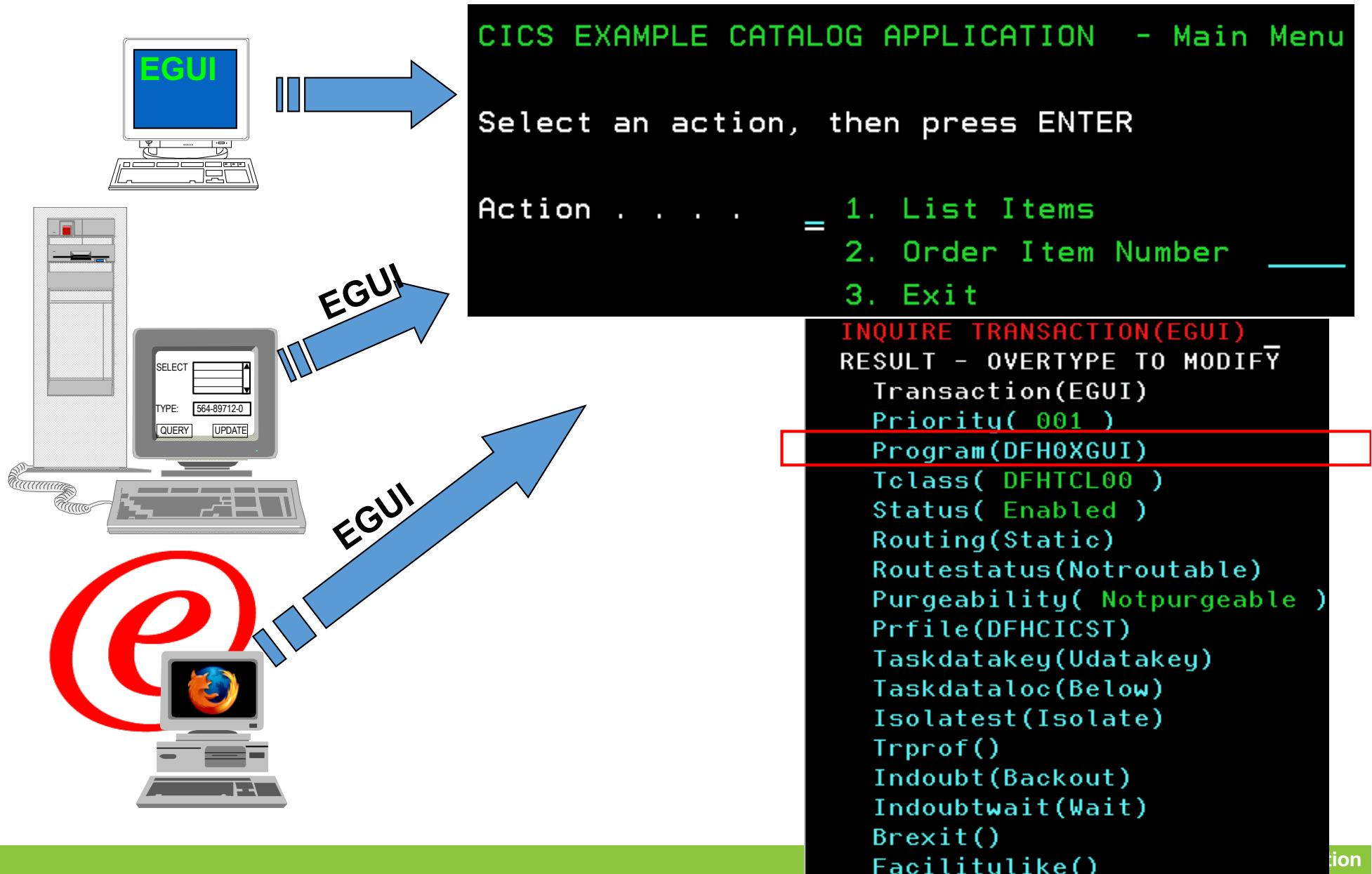
# CICS Management Services



# How CICS usually looked



# Initiating CICS Transactions



# CICS Application Server today

**More than 40 years  
(since '69) invested  
in Applications**

**5000 packages from  
2000 ISVs**

**30 million users**

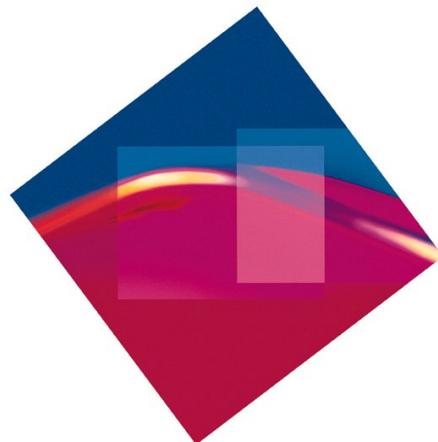
**90% of the Fortune  
500 use CICS**



\$1Trillion  
applications  
....IDC

**50,000  
CICS/390  
licenses**

**30 Billion transactions/day)  
\$1Bn/day (~ Nasdaq)**



**20,000 customers  
worldwide**

**950,000 programmers  
earn their living  
from CICS**

**Used by 490 + of IBM's  
top 500 customers**

**Over 950,000 concurrent  
users/system**

# AGENDA

3

- What is WebSphere Application Server → (WAS) ?
  - *Focus on WAS z/OS*

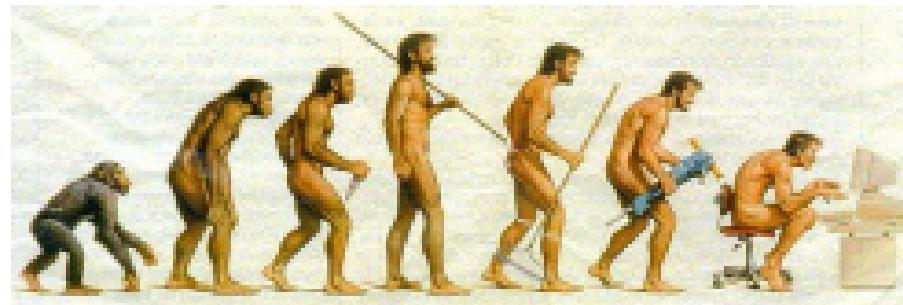
# Agenda

- **Introduction WebSphere Application Server for z/OS**
- **WAS z/OS: what's different from “Distributed” WAS?**
- **Co-location benefits**
- **WebSphere Application Server v7**

# What is WebSphere Application Server?



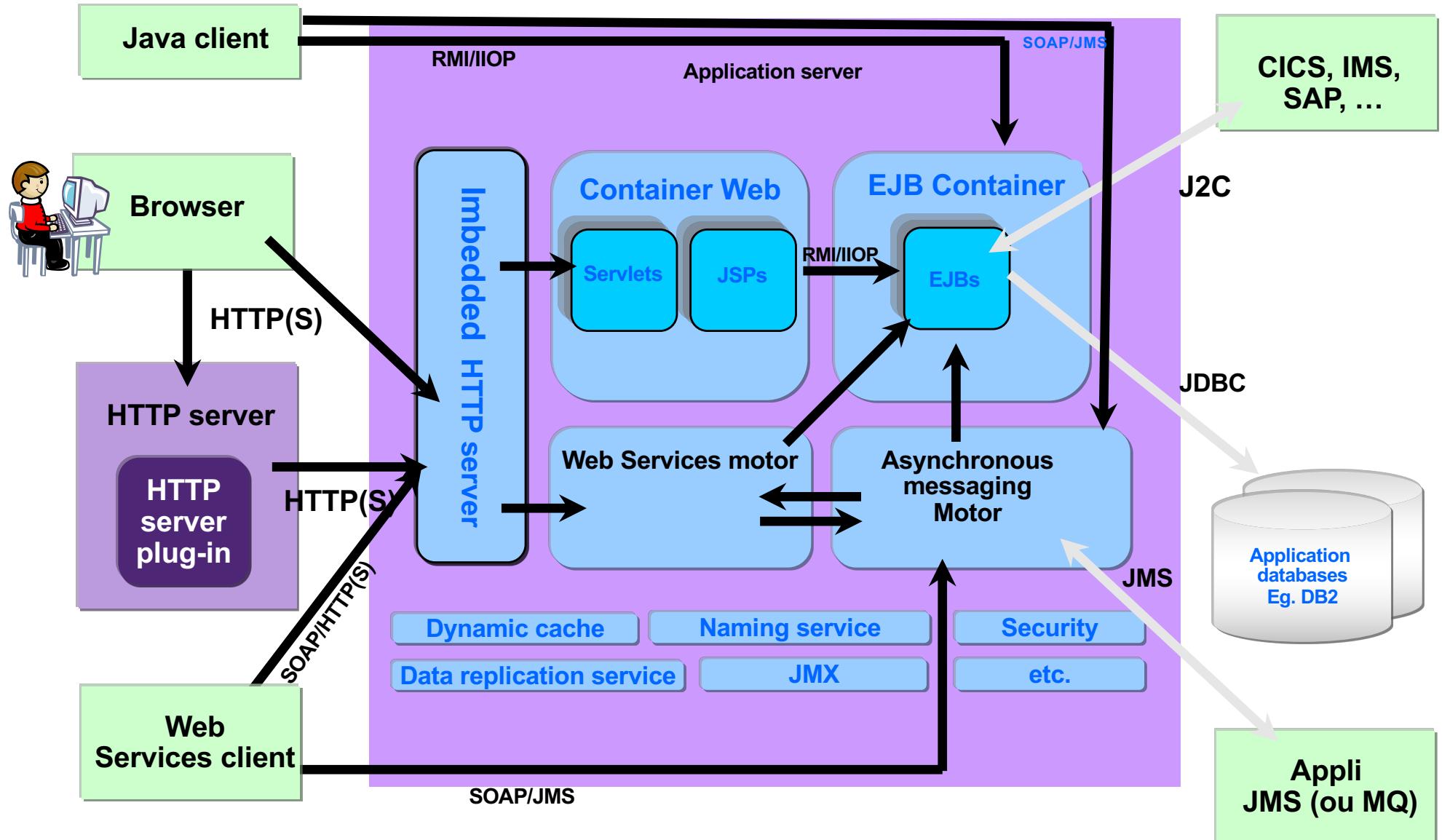
- **WebSphere Application Server (WAS) is a *runtime execution environment* for applications that provides numerous standard functions...**
  - Communication interface
  - Security interface
  - Transactional interface
  - Standard services for logging, alerts, administration, etc.
- **With WAS, you're getting a runtime environment**
  - Like IBM has been providing for 30+ years with CICS & IMS...
  - But exclusively in **Java**, and **standardised** (JEE)



22

# Logical architecture

IBM

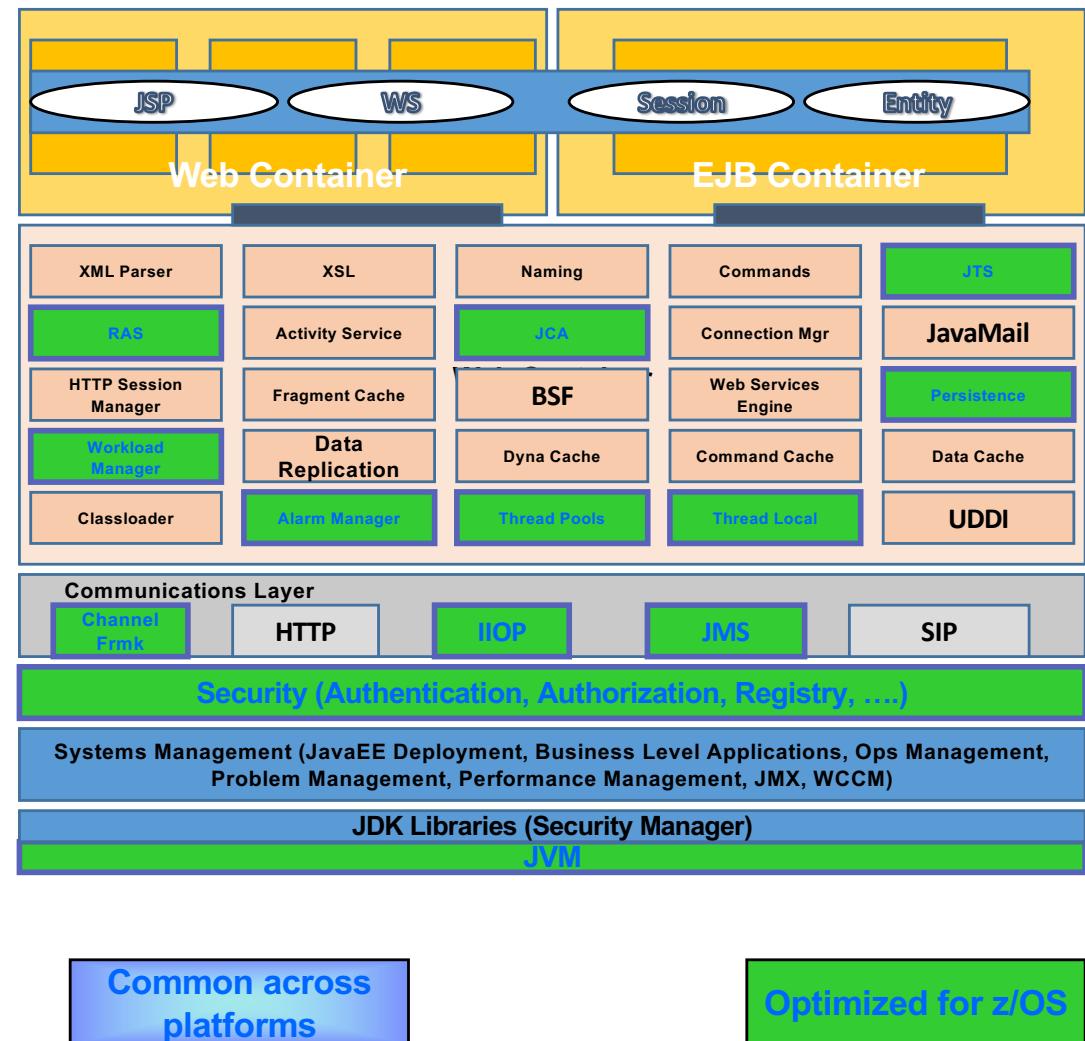


# WAS for z/OS: Active benefits



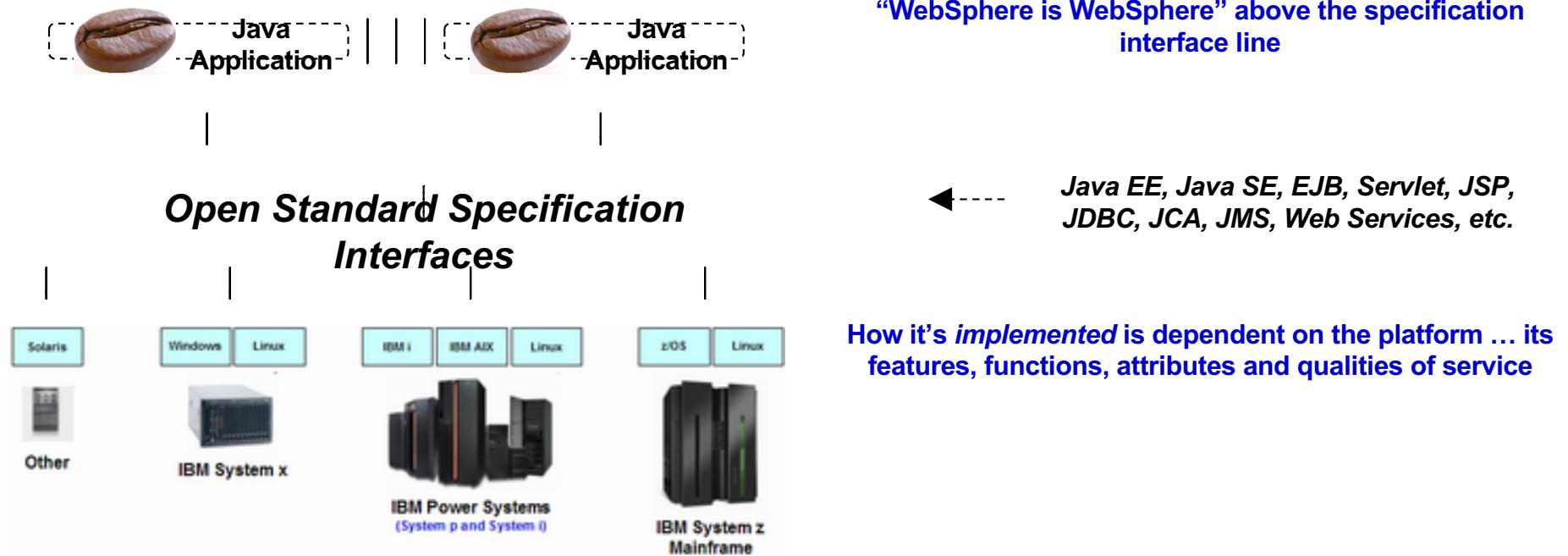
*Integration with z/OS that maintains application transparency*

- Server Architecture
  - Control/Servant Region Split
- Workload Management
  - Leverages Workload Manager
- Security
  - Use of the Security Authorization Facility
- Transaction Management
  - Leverages Resource Recovery Services
- Connectors
  - Leverages available local (Type 2) connectors
- Thread Management
  - OS level threads for monitoring and control
- Scalability
  - Multiple Servant Region
- Communications layer
  - True Asynchio model
- Recovery
  - Leverages Automatic Restart Manager
- Reporting
  - System Management Facility



# Very Important Starting Concept

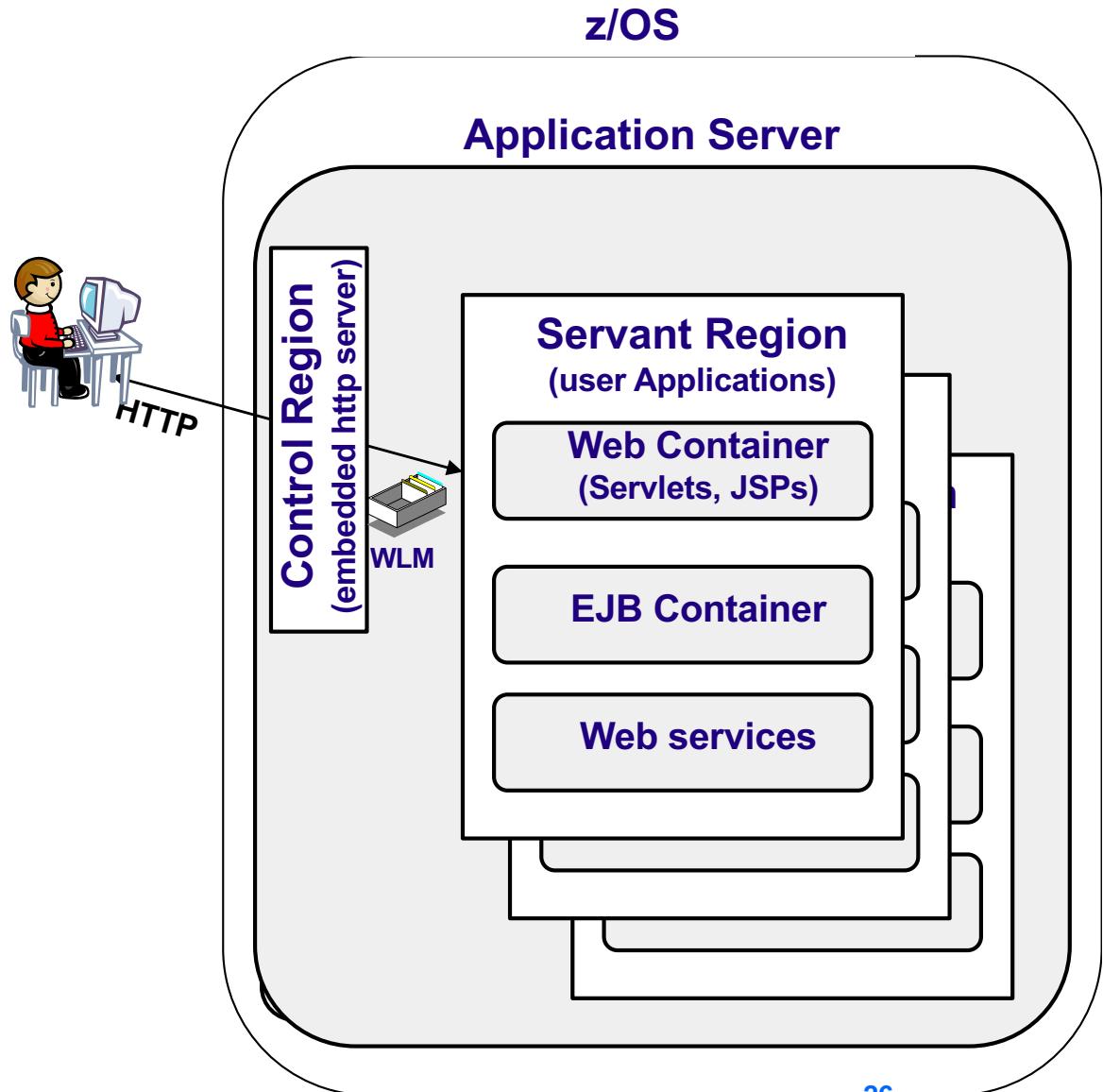
This point can't be stressed enough -- the differentiation is *not* in the open standard specification support offered. *That is common across platforms!*



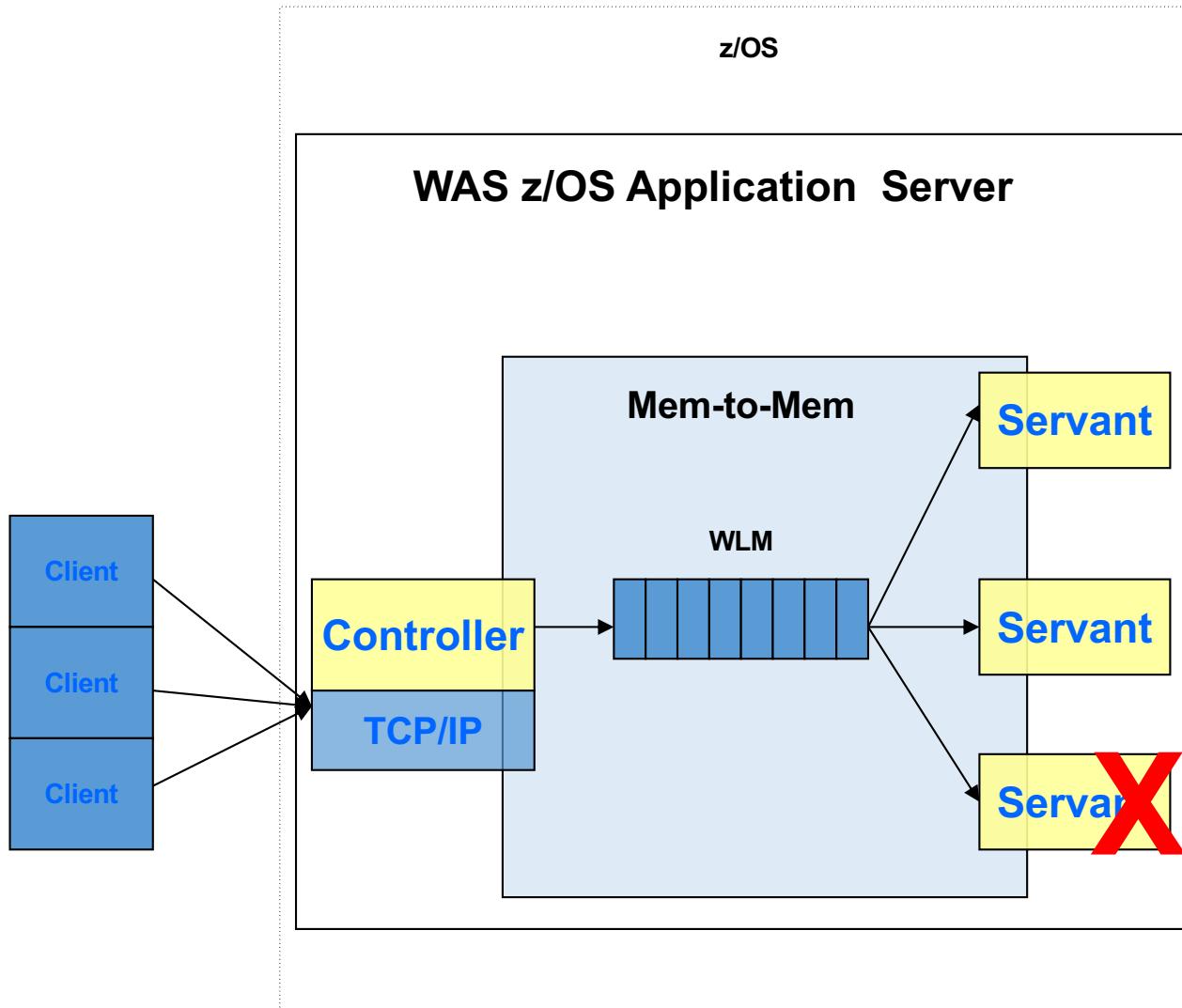
In older versions there were differences ... starting with V6.0 they're aligned

# WAS z/OS internal architecture – very different!

- z/OS implementation uses a “master-slave” hierarchy
  - ▶ This architecture found in found in CICS and IMS also
- A WAS “Control Region” distributes the application work to its “Servant Regions”
- This is key to understanding the value of WAS on z/OS
- This is a big part of what brings the business value...
  - ▶ Reliability
  - ▶ Reactivity

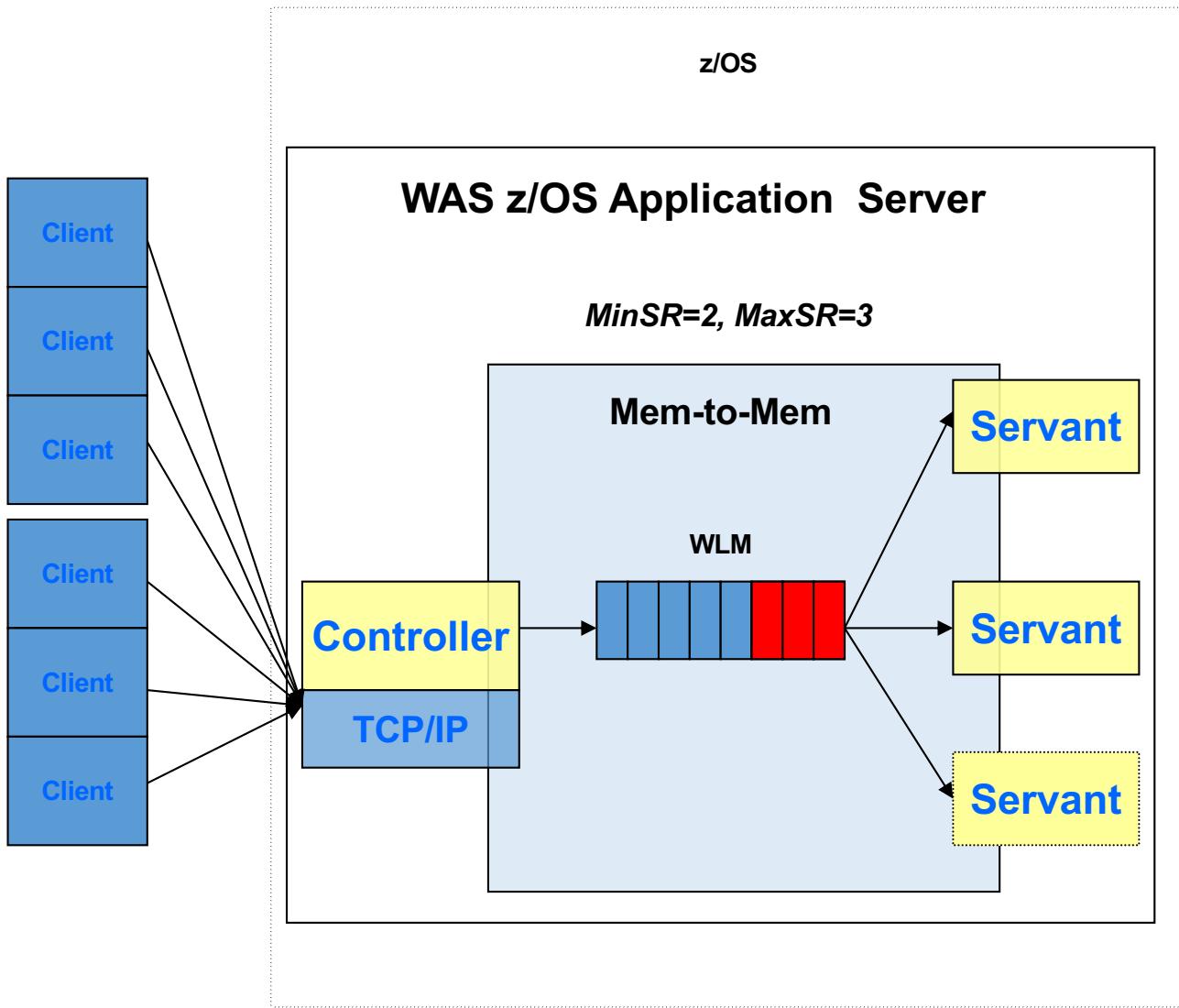


# WAS z/OS reliability



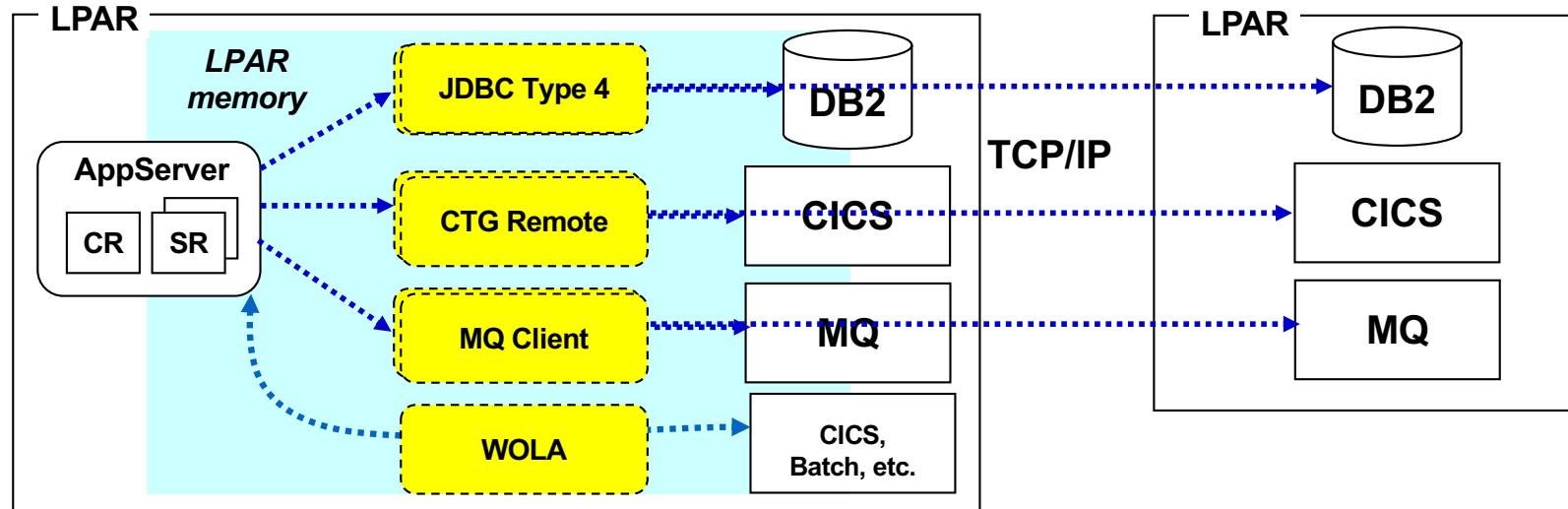
- Isolation
  - System code runs in the WAS Controller
  - Application code runs in the Servant(s)
  - WLM automagically starts Servants (under parameter control)
- In case of a Servant failure, the Controller detects and routes requests to the other Servant(s)
  - App session context maintained by Controller
- The Controller can start (and stop) additional Servant instances

# WAS z/OS scalability



- The WAS administrator specifies the minimum & maximum number of Servants that can run
- If WLM detects the need for additional resources, extra Servants are started
- And if the load decreases, WLM can stop unnecessary Servants to economise your System z resources

# Co-Location - Cross-Memory Communications



- ✓ Cross memory speed
- ✓ Avoids encryption overhead
- ✓ Security ID propagation
- ✓ Exploitation of z/OS transaction management (RRS)
- ✓ Avoid serialization of parameters
- ✓ Single thread of execution

*Benefits: Save mips, increase robustness, augment security*

For details, see: [WAS z/OS – the value of Co-Location](http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101476), <http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101476>

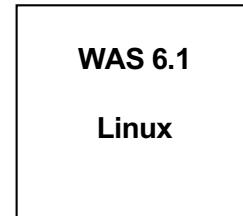
# JDBC DB2 Local vs. Remote



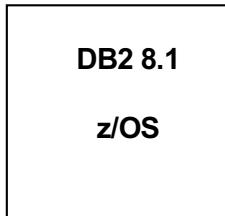
①

## Separate Machines

4 CPUs (32% busy)



4 CPUs (98%)

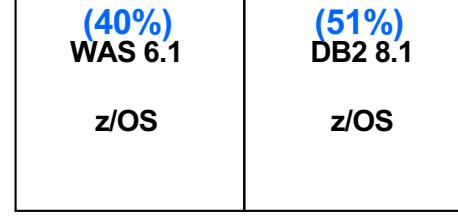


Power System

150

## Separate LPARs

8 CPUs in shared pool (91%)



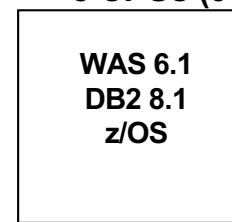
System z

160 tps

## Same LPAR

8 CPUs (91%)

WAS 6.1  
DB2 8.1  
z/OS



System z

243 tps

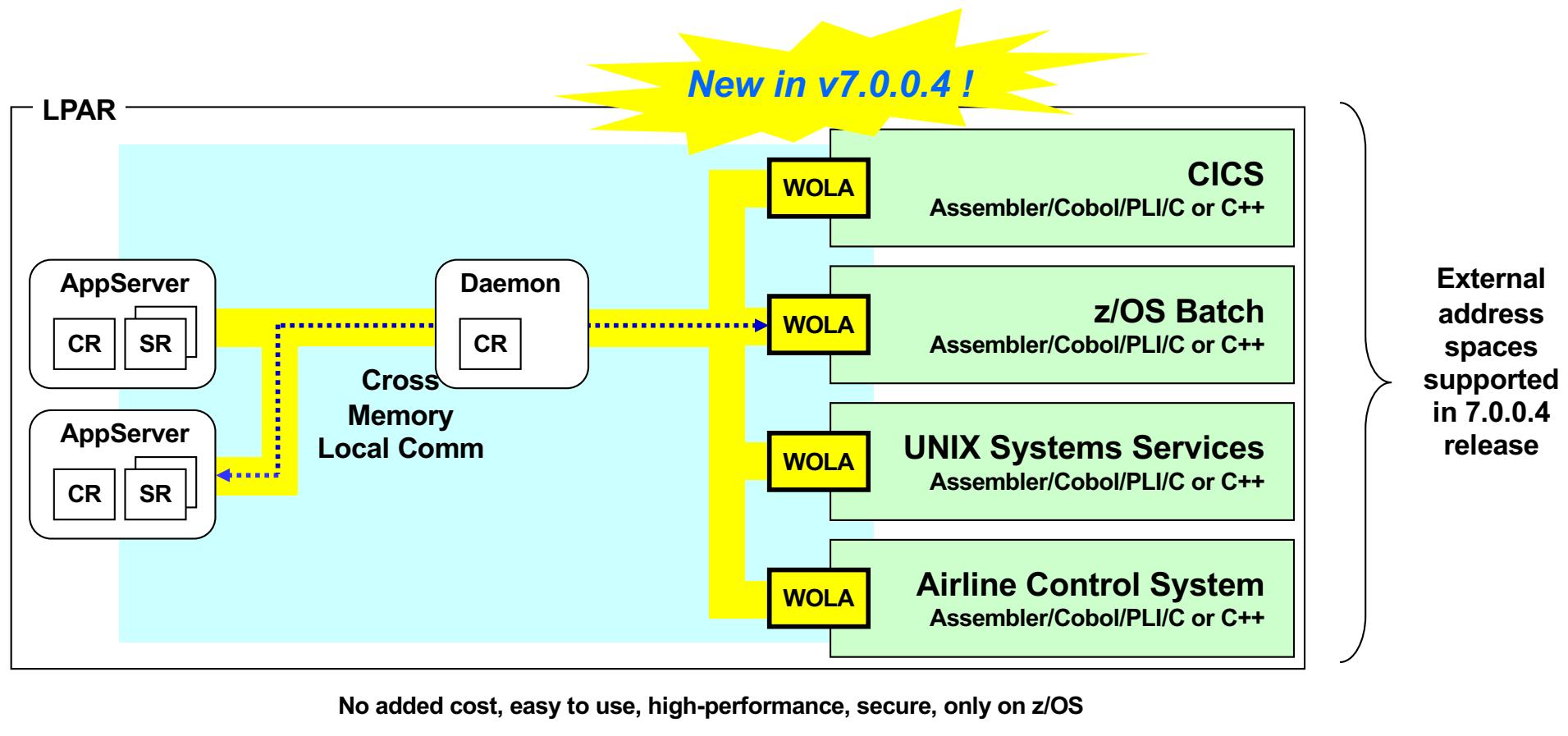
**52% more throughput with colocated workload**

For details, see

[https://www.ibm.com/events/wwe/impact/impact09cms.nsf/download/k880ee712da22da2120f796e00f/\\$FILE/TCO%20IMPACT%20%232056%20Shedletsky%20for%20Distribution.pdf](https://www.ibm.com/events/wwe/impact/impact09cms.nsf/download/k880ee712da22da2120f796e00f/$FILE/TCO%20IMPACT%20%232056%20Shedletsky%20for%20Distribution.pdf)

Additional (excellent!) report at <http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101476>

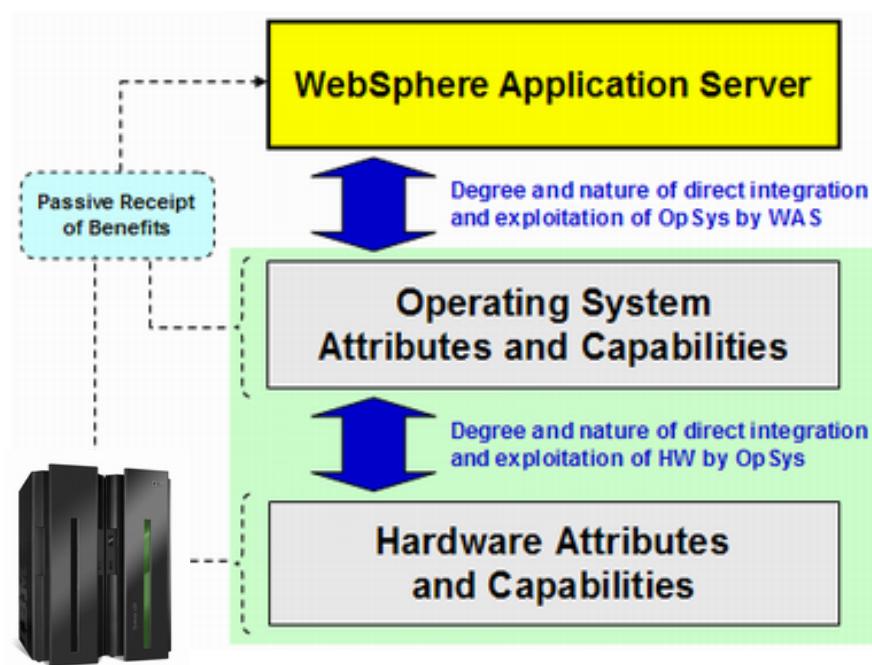
# Cross-Memory: WebSphere Optimized Local Adapters (WOLA)



## Customer benefits:

- Significant reduction in mips for WAS interfacing to other z/OS non-WAS applications
- The ability for customers to fully exploit their assets, eg.
  - WAS can call “legacy” assets as a peer – very high speed, very low cost
  - Batch programs, CICS, etc. can make use of corporate Java-based assets

Programs that run on System z and z/OS receive passive benefits in several different areas:



## Hardware

- Inherent maturity and stability of design
- Redundancy and flexible updates
- Balanced design offers very high throughput
- Mature and proven virtualization through LPAR

## Operating System

- Tight integration with server hardware design
- Extremely mature architecture
- Tremendous scalability
- Parallel Sysplex architecture
- Workload Manager (WLM)
- Intelligent Resource Director (IRD)
- Local TCP optimization
- Mature systems management tools (SMP/E, SMF..)

# WAS on z/OS value: the net net



- **WebSphere Application Server on z/OS compared to WAS “Distributed”**
  - Share a common code base and JEE applications can be placed on the platform that makes the most sense for the customer
  - WAS z/OS does have a unique underlying architecture to leverage the QOS that customers expect of the z/OS platform
- **WebSphere for z/OS takes full advantage of the platform’s qualities**
  - Typically of greatest interest when WebSphere is connecting to the z/OS applications and data
  - Provides the very best in scalability and full resource optimisation
  - Exploits the intrinsic z/OS QOS in transactionality, security and high-availability
- **WebSphere Application Server brings significant new value**
- **All of the WAS “stack” products profit from WAS enhancements**

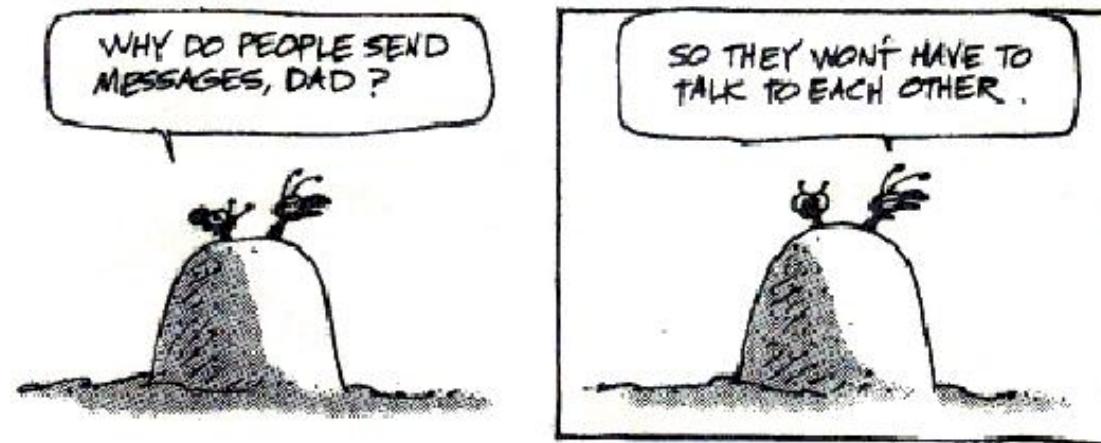
# AGENDA

4

## ❑ What is Message Management ?

- Focus on IBM MQ Series

# Messaging



# Synchronous vs. Asynchronous

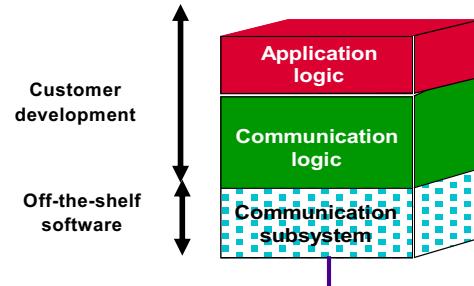


- 1 to 1
- “tightly coupled”
- Blocking

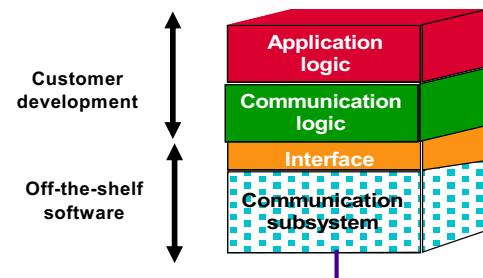
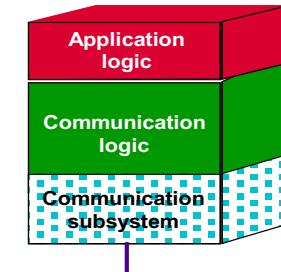


- Via an intermediary
- “loosely coupled”
- End-points liberated quickly
- Particularly interesting when the “sending” party doesn’t necessarily need an acknowledgment

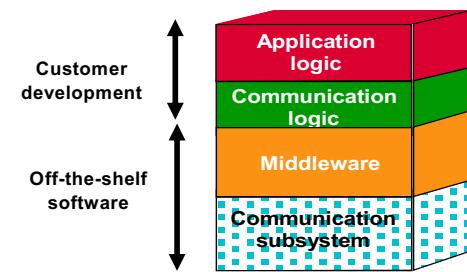
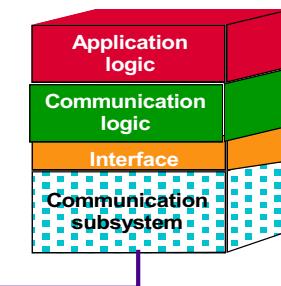
# Communications modes



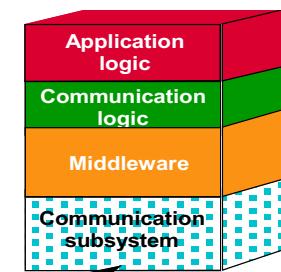
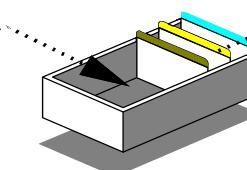
- APPCC, CPI-C (LU6.2), Sockets, NetBIOS, ...
- Fastidious, long, complex coding
  - Depends upon an active communication session



- RPC, HTTP, SOAP, ...
- Simplified programming
  - Still depends upon an active communication session

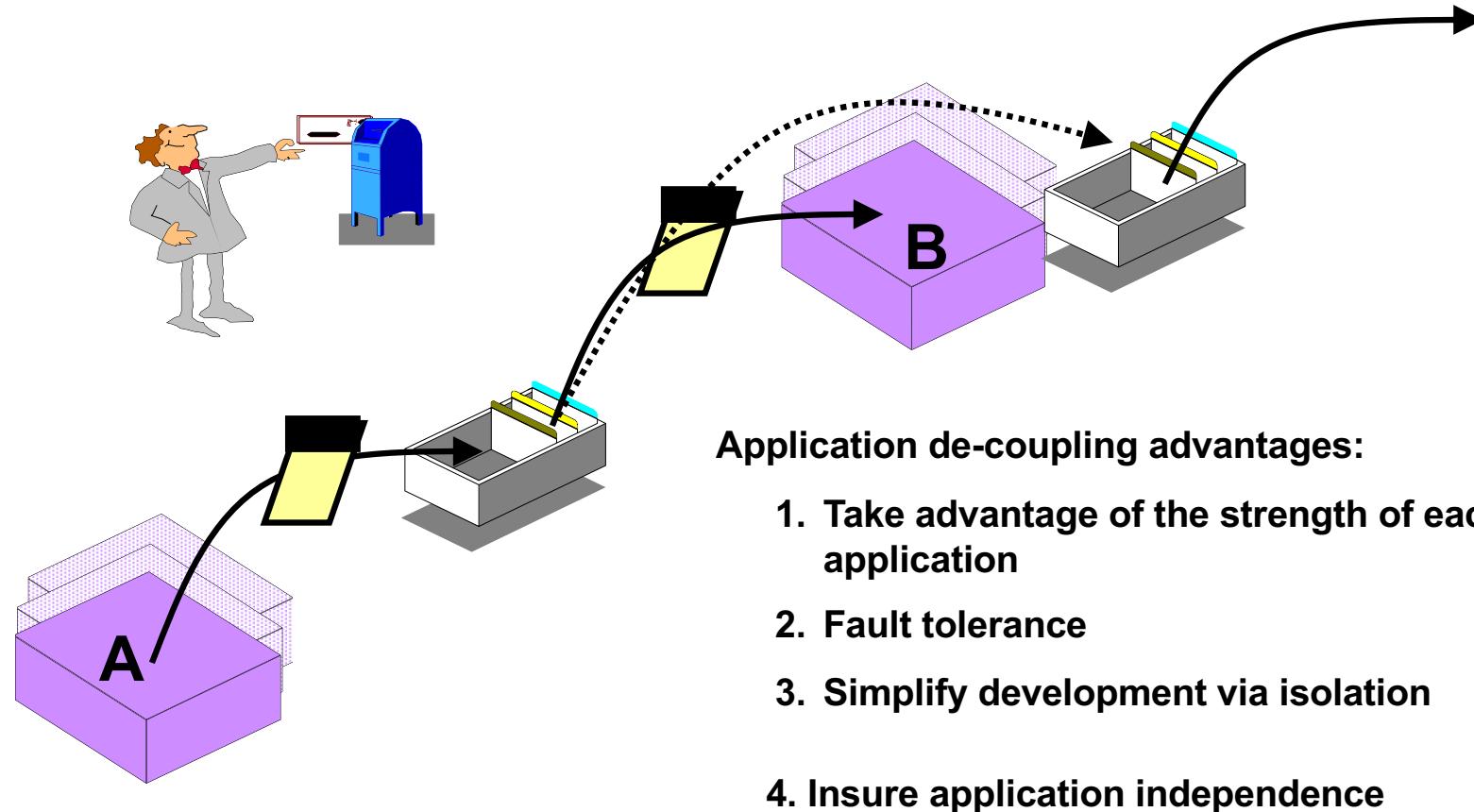


- MOM**
- Simplified programming
  - No requirement for permanent connection



# What's a MOM?

Messaging software (“Message Oriented Middleware”, or MOM) allows two applications to exchange data asynchronously via “queues”.



# IBM MQ (MQSeries)



*...transport mechanism between platforms  
and applications*

**M = Message**

An object containing a string of data  
Created and read by applications

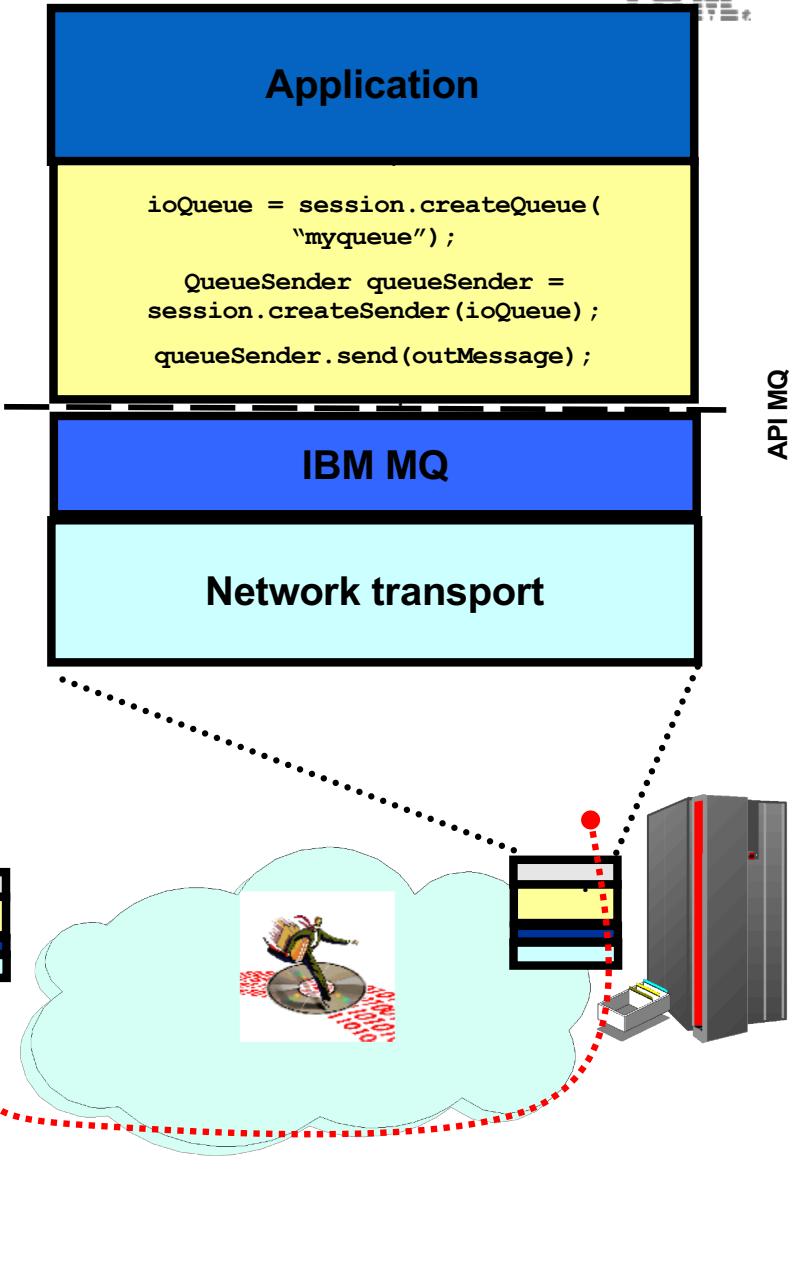
**Q = Queuing**

A container for messages

Message-oriented middleware

# What's IBM MQ?

- A simple, efficient API for sending/receiving data messages
- Enables fast, reliable asynchronous messaging from application to application
- An industry standard for Message-Oriented-Middlewares (MOM) with 65-80% of the market.
- Can be used on over 45 different platforms (Windows, Linux, AIX, Solaris, HP-UX, iSeries-AS/400, etc, etc., and of course..... System Z !)
- Can be used from all major programming languages (C, C++, COBOL, Fortran, BAL, PL/I, Java/JMS, VB, RPG, etc.)
- Includes support for the MQ Client – a remote API accessing the queue manager resources.
- Other unique features of MQ...
  - Provides a 2PC syncpoint manager for distributed MQ and full 2PC transactional participation on z (CICS, IMS, RRS)
  - Support for high-availability and load-balancing via MQ Clusters as well as Shared queues with Sysplex on Z
  - Support for point to point and pub/sub messaging
  - Support for message grouping and segmentation
  - Support for SSL authentication and encryption
  - Support for message compression
  - Exploits each platform, and in particular z/OS, taking full advantage of RACF, Sysplex, SMF, CICS & IMS Bridges, etc.



# Why IBM MQ?



Performance

Rich, simple API

Ease of installation, maintenance

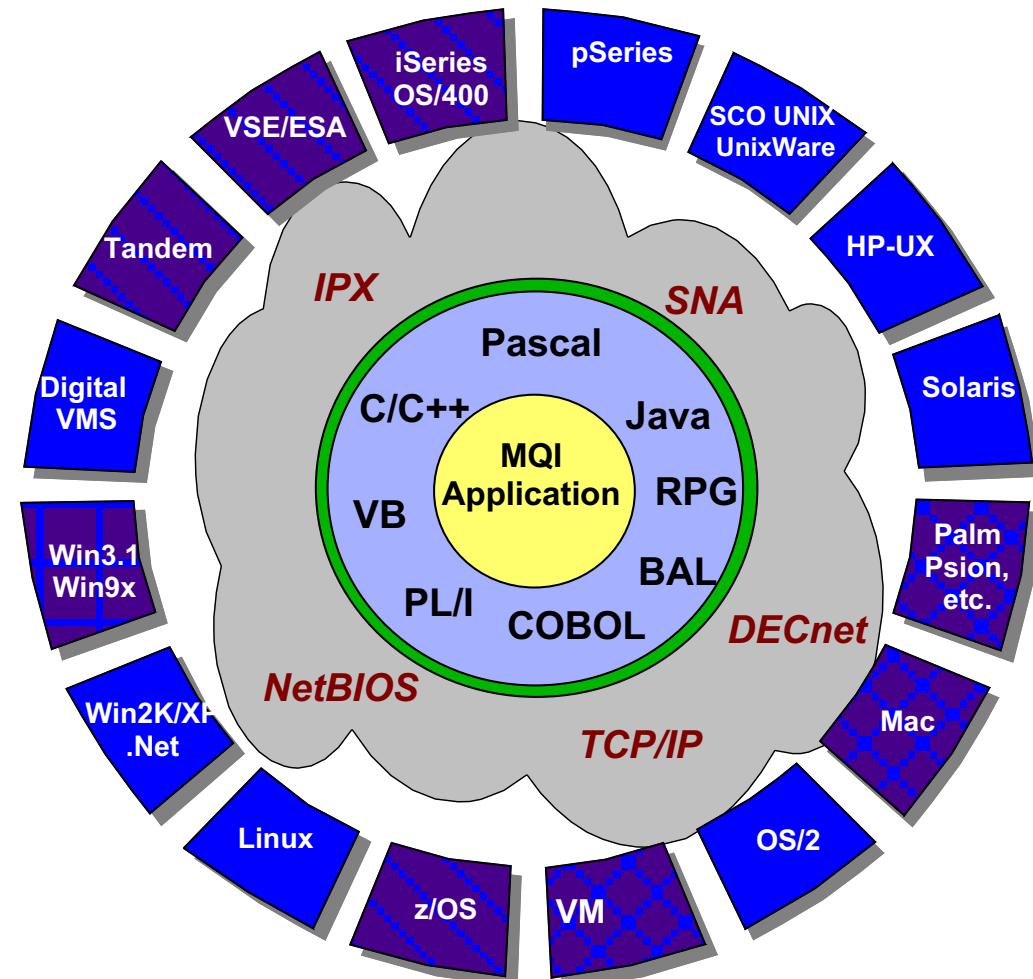
Breadth of platforms

Robust

Wealth of partner applications

Many extra features

- MQ Clusters
- Shared Queues ("MQPlex")
- Distribution lists
- Transaction management (2PC)
- SSL (encryption, certificates)
- HTTP enveloping and bridging
- Data compression
- Reply and report options
- Pub/Sub
- Message grouping/segmentation
- Message expirations
- etc.



*Above all, IBM MQ is the market standard for guaranteed message delivery between applications.*

Similar to accessing a database or file system:

**MQCONN** Connect to a queue manager

**MQOPEN** Open a queue (either for input or output)

**MQPUT** Put a message to a queue

**MQGET** Get a message from a queue

**MQCLOSE** Close a queue

**MQDISC** Disconnect from a queue manager

**MQPUT1** Perform an open, put and close to a queue

**MQCMIT** Commit a set of messages

**MQBACK** Rollback a set of messages

**MQBEGIN** Begin a unit of work as part of a commit

**MQINQ** Inquire on attributes of objects

**MQSET** Change attributes of objects

# Business Success

## ◆ Usage

- >7000 customer sites
- >350 of IBM's top 500 customers
- >66% of NA and European banks
- In heavy production usage, e.g. State Street Bank peaked at 375M messages per day, average 144M per day
- Business real-time usage, e.g. Chicago Mercantile Exchange, Sydney Olympics, Wimbledon, Delta Airlines

## ◆ Achievement

- > 75+% WW market share
- > 4000+ certified specialists (1000 companies)
- > 500+ ISVs offer services & products
- > 60+ significant formal IBM announcements since Jan 96

## ◆ Recognition

- 7 major industry awards
- Strong analyst interest in loosely-coupled environments
- Seen as de-facto standard by many analysts

All Industries with any of key data characteristics

- High volume
- High performance
- High value
- Integrated value chains

*Finance, Manufacturing,  
Retail, Telco, Travel,  
Insurance, Healthcare,  
Utilities, Govt, Sports*

7 industry  
awards

# AGENDA

5

- Open z Systems « assets » for New Business
  - A big picture

## Think of your Current z/OS Applications are like a Lego Castle



**Everything is complete.  
Unfortunately today the castle is all one piece – It's all or  
nothing  
You have to take the castle as it is**

20

## What if you took apart that castle into really usable pieces



Many of the pieces have value on their own.

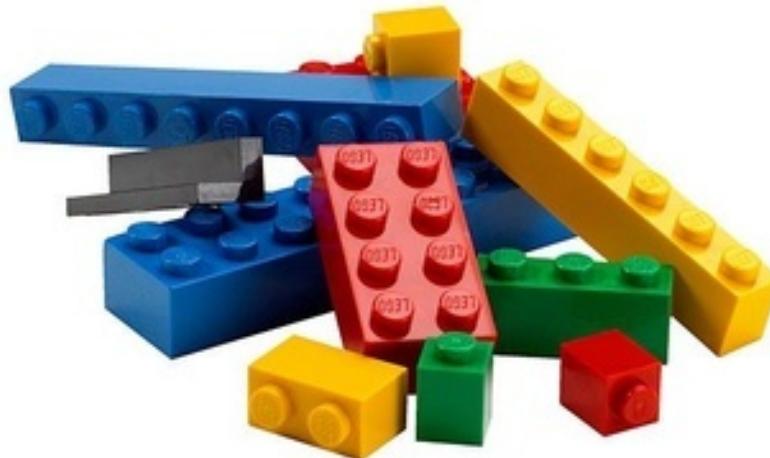
## Those z/OS pieces could be connected to other cloud pieces



To create something completely new (really fast)

22

**Today we are working with clients to break down their castles into pieces**



We call these pieces Services

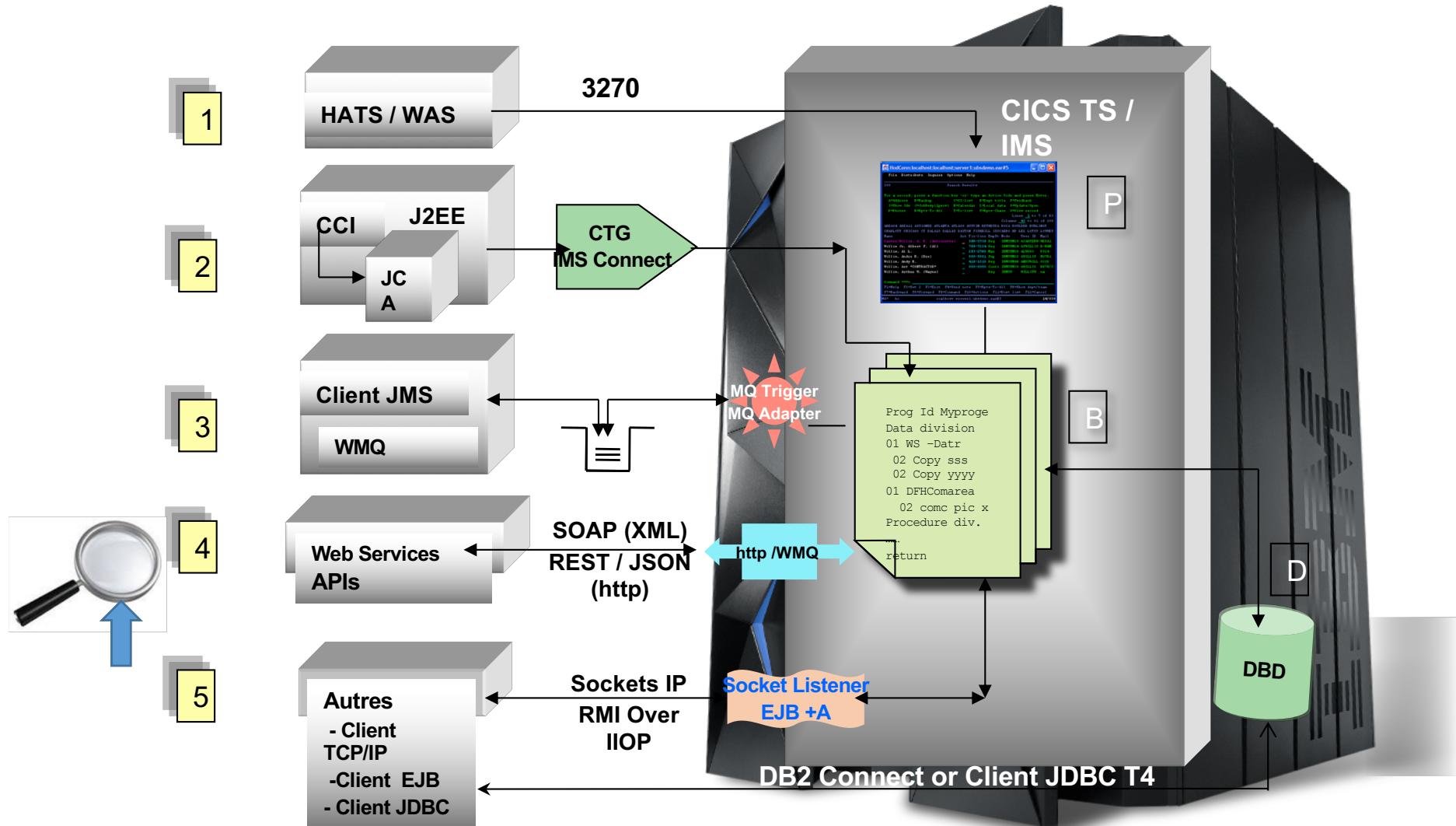
## An Important Retail Customer in US created a Mainframe Cloud Service because the existing Cloud Service wouldn't scale!

- The Customer mainframe service became a favorite of the Cloud developers.
- Now it is at the center of their Web, Smartphone and Store product information
- It's been running for 4 years – Over 16 Billion calls without an outage
- Can handle over 300,000,000 calls in a day
  - Last Black Friday it handled over 1,400 calls a second with 1 milisecond response time



# Ouverture de z Systems :

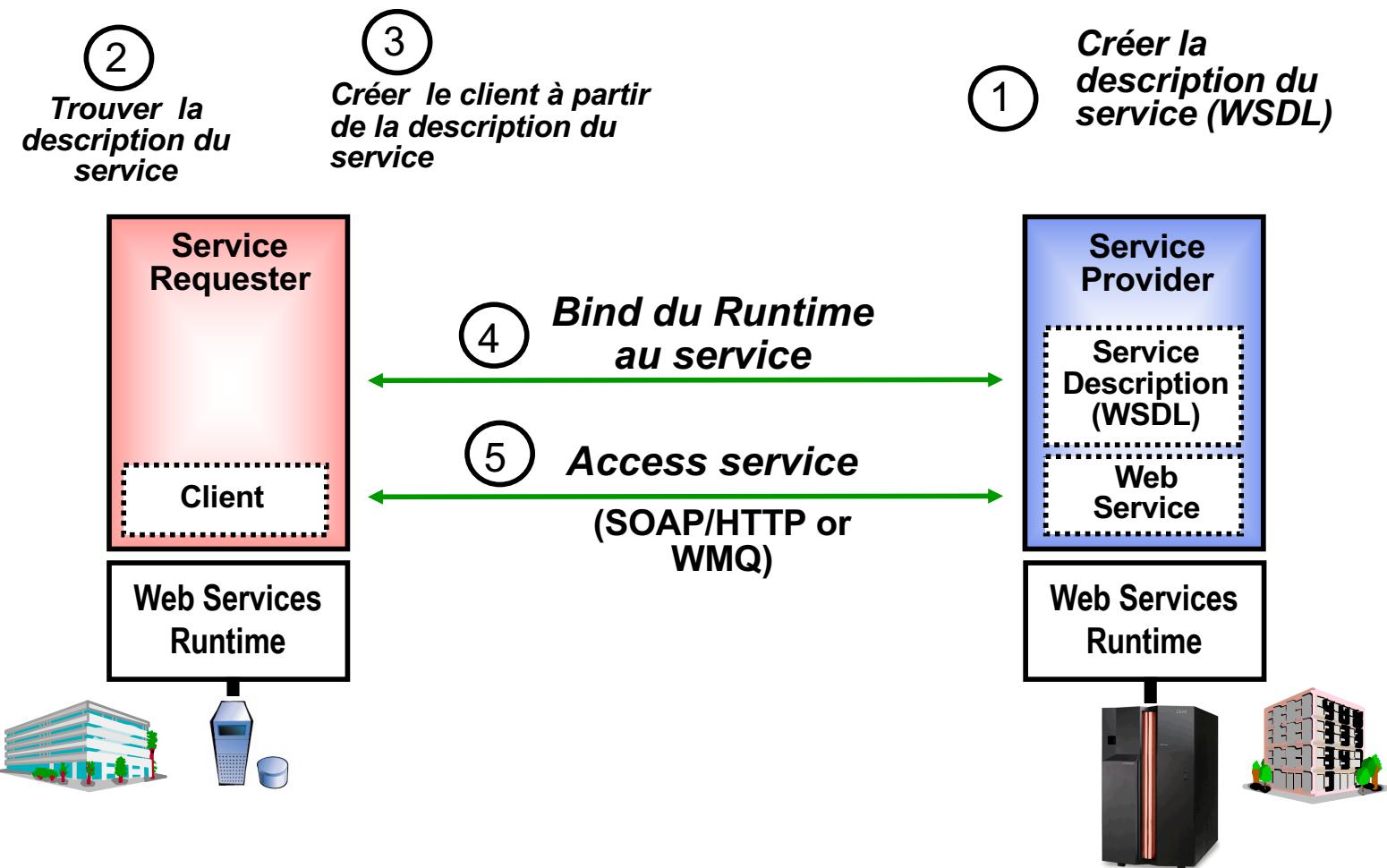
## Un choix très riche d'ouverture vers les standards de connexion du marché



# Zoom sur l'ouverture du Mainframe vers SOA

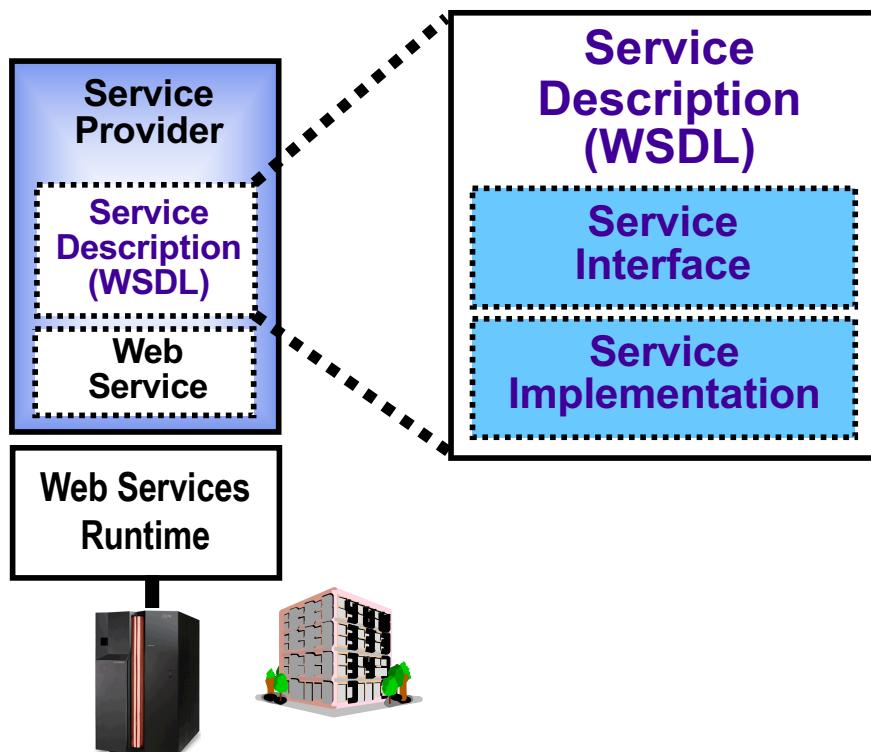
## Services Web via XML /SOAP

# Web Services : Quelques rappels



# Description du Service par WSDL

IBM



- Web Services Description Language (WSDL) écrit en XML
  - Abstraction sur la définition des opérations
  - Paramètres Input / output et types de données
  - Implementation en language neutre
- Description Complete “Network services or component”
  - fonctionnalités
    - Service Interface
  - Comment ils communiquent
  - Ou ils résident
    - Service Implementation
- Outils de Developpement
  - Les “Provider” l'utilisent pour documenter les services
  - Les “Consumer” l'utilisent pour générer le code du client accédant au service

# The Importance of XML in Web Services



You will see that XML is the common mechanism to exchange information in a web services environment. What is XML, and why is it valuable?

```
<SOAP-ENV:Envelope>
  <SOAP-ENV:Body>
    <q0:DFHCOMMAREA>
      <CustNo>3</CustNo>
    </q0:DFHCOMMAREA>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

A series of “tags” that mark the beginning and end of blocks of XML

It holds both the data, as well as description of the data

<CustNo> provides an indicator of what the data is; “3” is the actual data.

It is both machine readable and human readable, which makes things relatively easy to understand

Contrast with bit-format protocols, where bits within bytes meant certain things.  
Machine readable yes; human readable less so.

Characters use “Unicode” encoding, which means it’s universally understood

As opposed to the old EBCDIC vs. ASCII debates



Example of XML SOAP envelope we'll use in one of the labs

## XML is “Self Describing”

Something called a “Schema Definition” (XSD) is used to tell a program what XML tags to expect.  
The WSDL file (more in moment) has XSD information

## XML can be “parsed”

If a program knows what tags to expect (the WSDL supplies this), then the program can “parse” (extract) information from the XML.

# The WSDL File



WSDL contains information about the service -- where it's located, what parameters it takes as input, what it gives back as output, what XML tags to use, etc. It is sometimes known as a "bindings file".

It can be long and complicated ... what follows is a boiled-down snippet to show essence

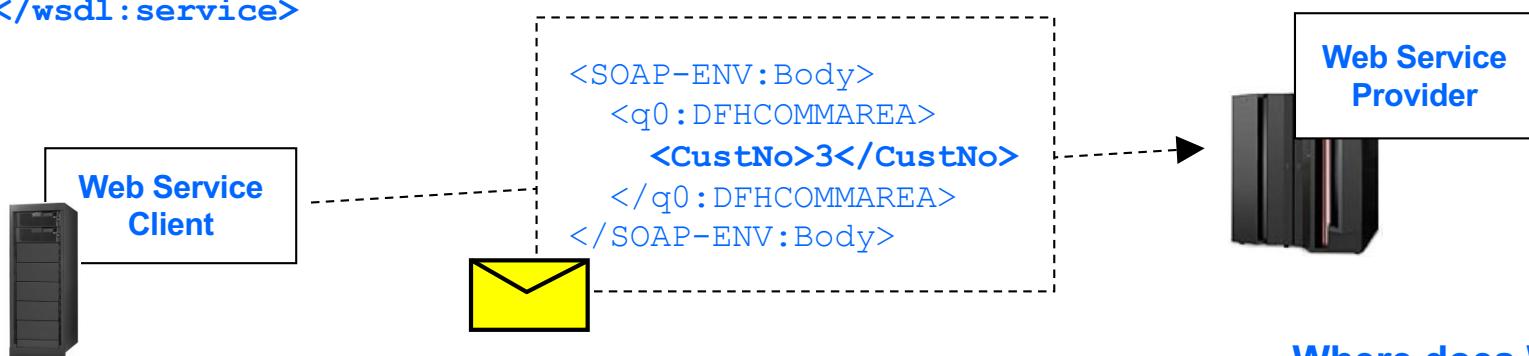
```
<complexType name="DFHCOMMAREA">
  <sequence>
    <element name="CustNo">
      <simpleType>
        <restriction base="int"/>
      </simpleType>
    </element>
  </sequence>
</complexType>
```

Client knows input XML and data requirements based on this

- What service will return was here ... removed to save space

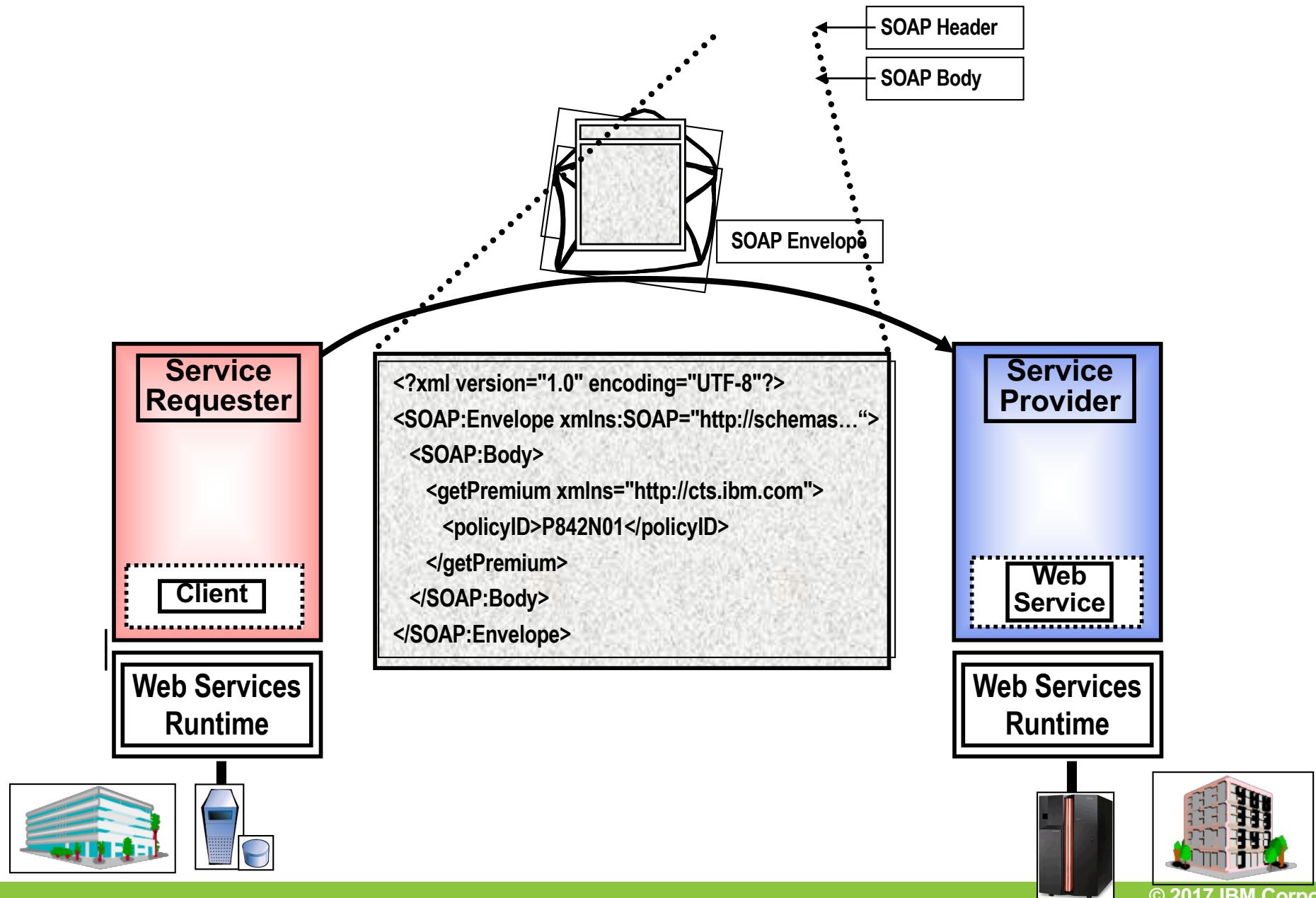
```
<wsdl:service name="WBCSCUSService">
  <wsdl:port binding="tns:WBCSCUSBinding" name="WBCSCUSPort">
    <soap:address location="http://mig.null.washington.ibm.com:12301/WBCSCUST"/>
  </wsdl:port>
</wsdl:service>
```

Client knows where service is located based on this



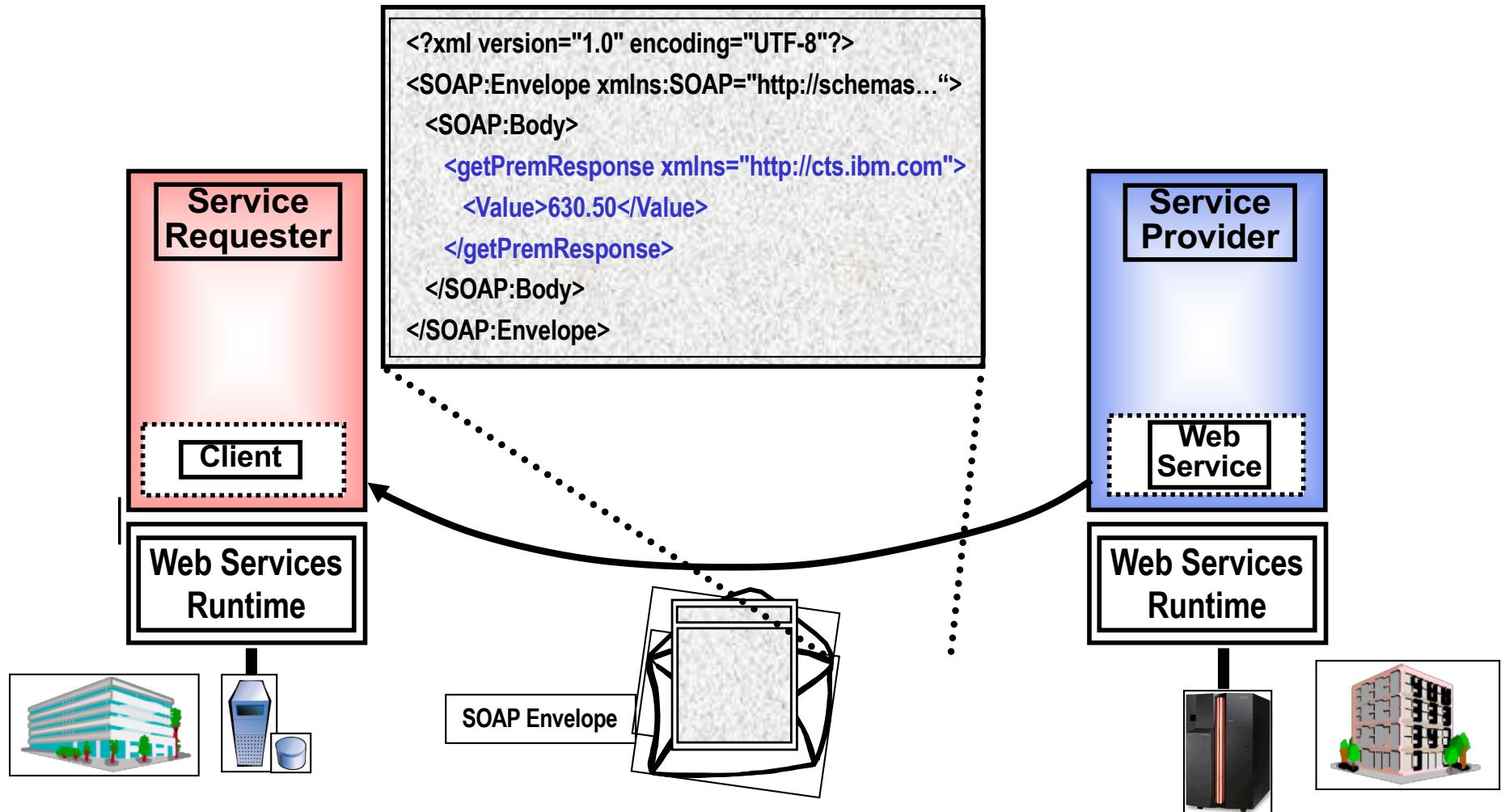
Where does WSDL come from? ...

# Fonctionnement d'un Web Service à l'exécution



# Fonctionnement d'un Web Service à l'exécution

IBM.

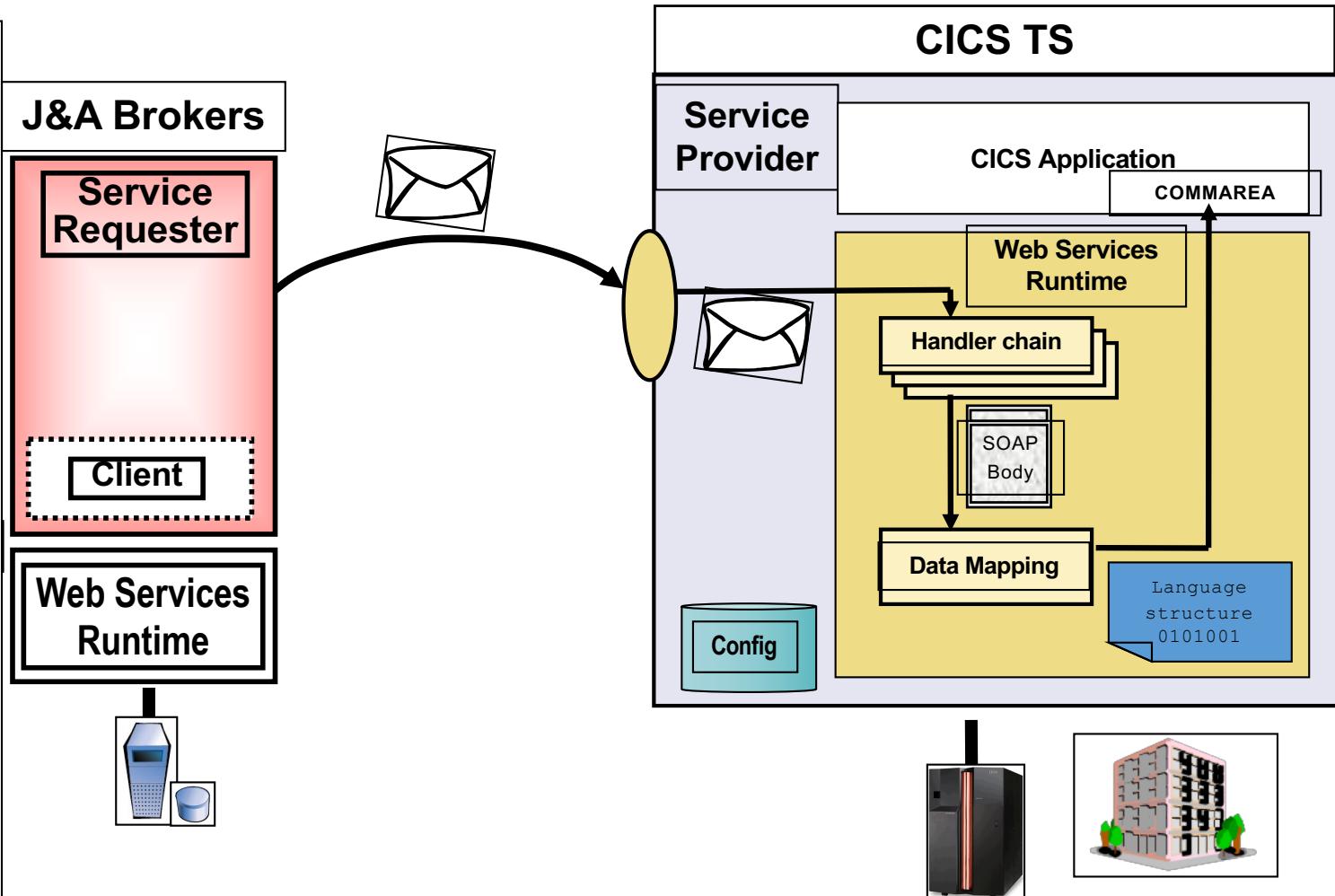


*Chaque partie en jeu comprend le message car SOAP utilise XML → indépendance / plate forme.*

# Fonctionnement des Web Services CICS



1. Réception de la requête SOAP
2. CICS Web Services runtime récupère le message
3. Handler chain traite le message SOAP
4. Data Mapping transforme l'XML en bytes, appel l'application serveur



# **Zoom sur l'ouverture du Mainframe vers le Monde de l'API Economy**

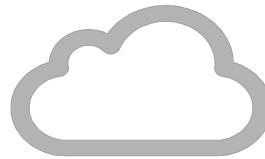
## **La Transformation Digitale**

# Transformation digitale en marche

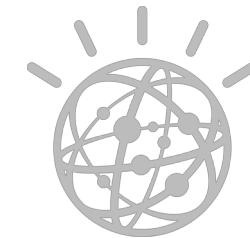
IBM.



**La manière dont  
les entreprises  
interagissent avec  
leurs clients  
change**



**L'endroit où  
'tournent' nos  
applications  
importe peu**



**Nos données sont  
interrogées aussi  
bien par de  
simples requêtes  
que par des  
processus  
cognitifs**

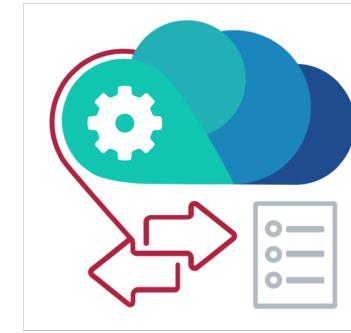
# Transformation digitale en marche



**Application de commande et de paiement en ligne représentant \$1.6M et traitant 21% des transactions de l'entreprise**



**Concept du véhicule connecté pour améliorer l'expérience conducteur et passager, et aussi vendre des données du véhicules à certains partenaires**



**Secteur public et l'Open data pour plus de transparence, éliminer la corruption, favoriser l'innovation, améliorer la réactivité en cas de catastrophe...**

API

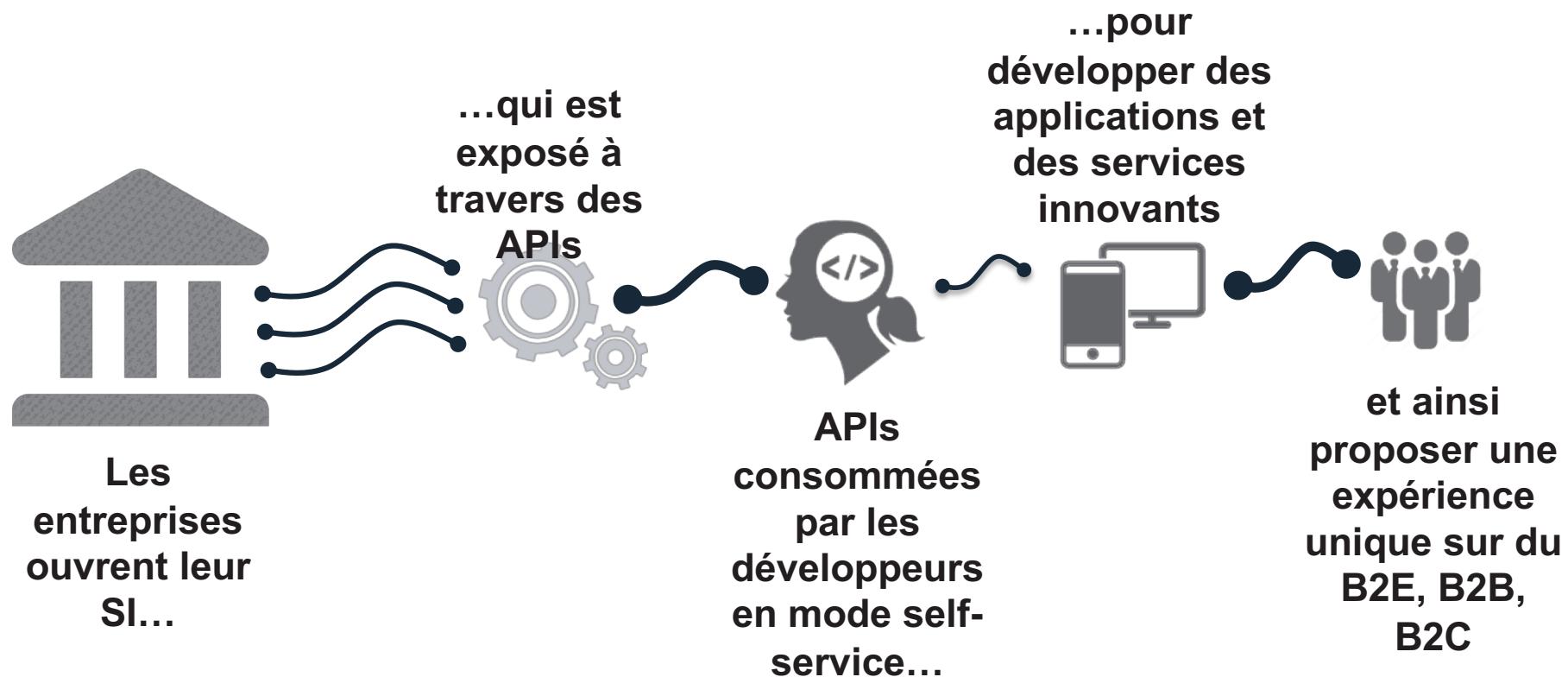


DIGITAL



# Utilisez les assets de votre SI pour en tirer un avantage compétitif

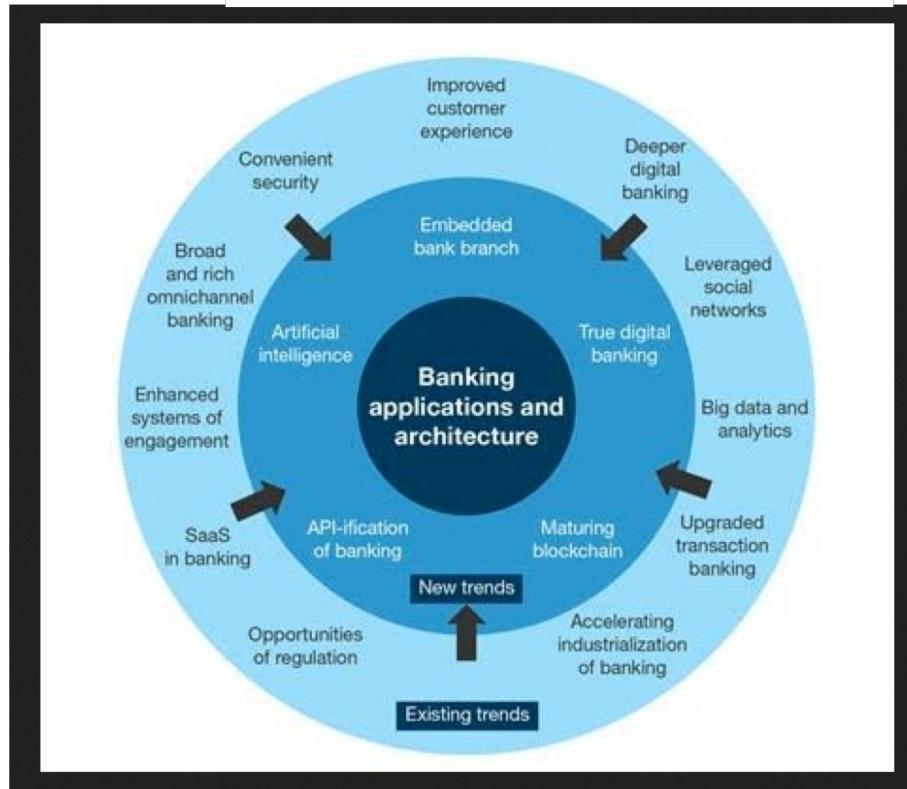
IBM



# Les APIs sont essentiels pour la transformation numérique

IBM.

## Banque digitale : 5 tendances à surveiller en 2017



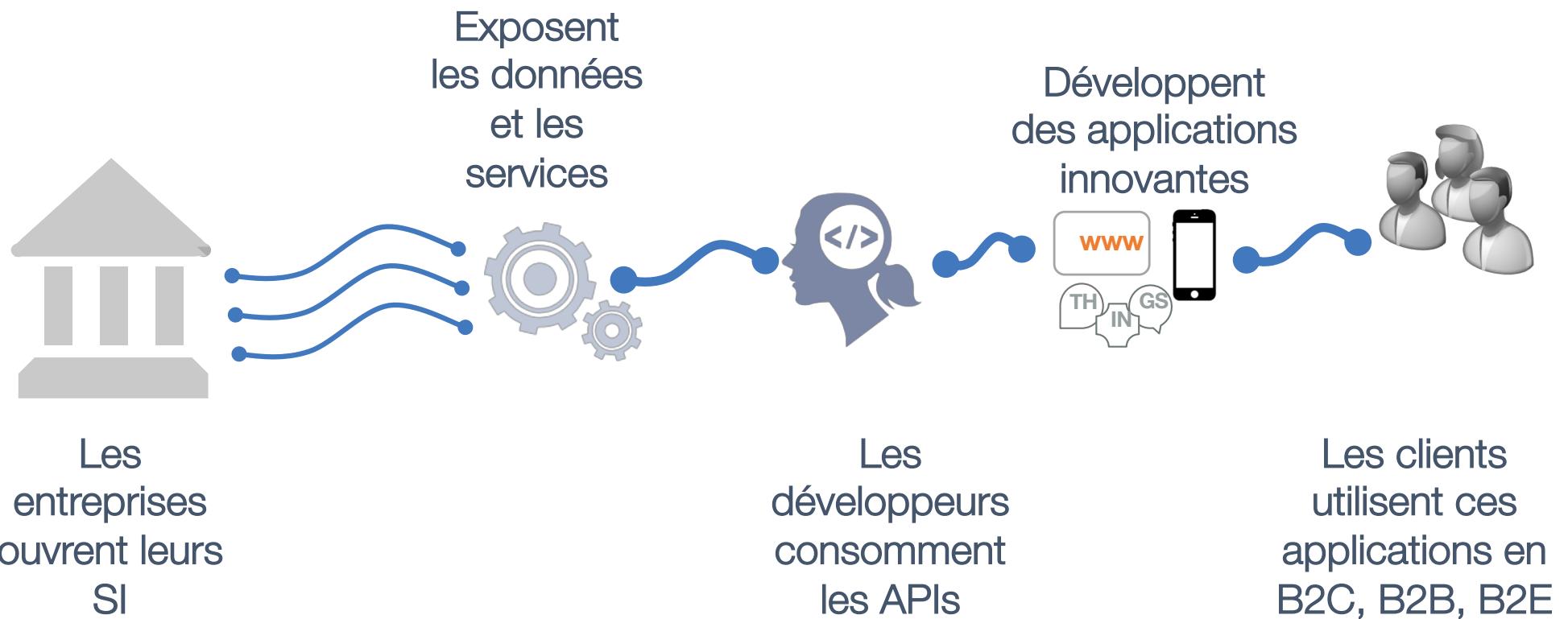
Le Monde Informatique

Le 01 Septembre 2016

Le cabinet Forrester a publié une étude consacrée aux tendances en matière d'applications bancaires à considérer en 2017. Blockchain, intelligence artificielle et adaptation des API aux apps et services tiers en font partie.

<http://www.lemondeinformatique.fr/actualites/lire-banque-digitale-5-tendances-a-surveiller-en-2017-65787.html>

## La chaîne de valeur des APIs



# Un bénéfice pour les fournisseurs et les consommateurs d'API



Fournisseurs

Consommateurs

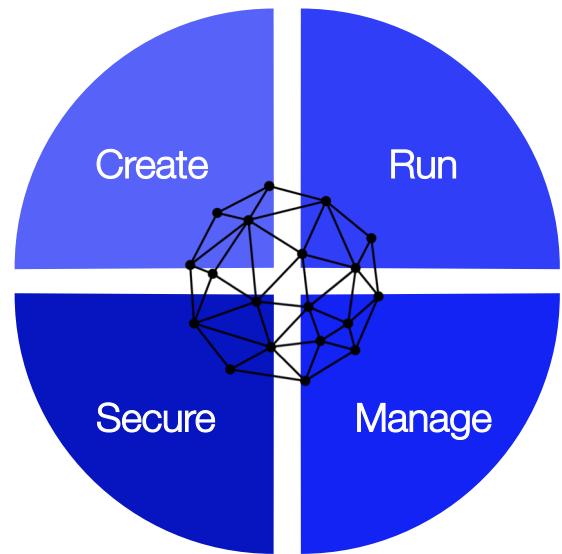
# Un bénéfice pour les fournisseurs et les consommateurs d'API



Fournisseurs

Consommateurs

## API MANAGEMENT



*Simplified &  
Comprehensive  
API foundation*

### Qu'est ce qu'API Connect?

Une solution complète et intégrée pour créer, sécuriser exposer et gérer des APIs.

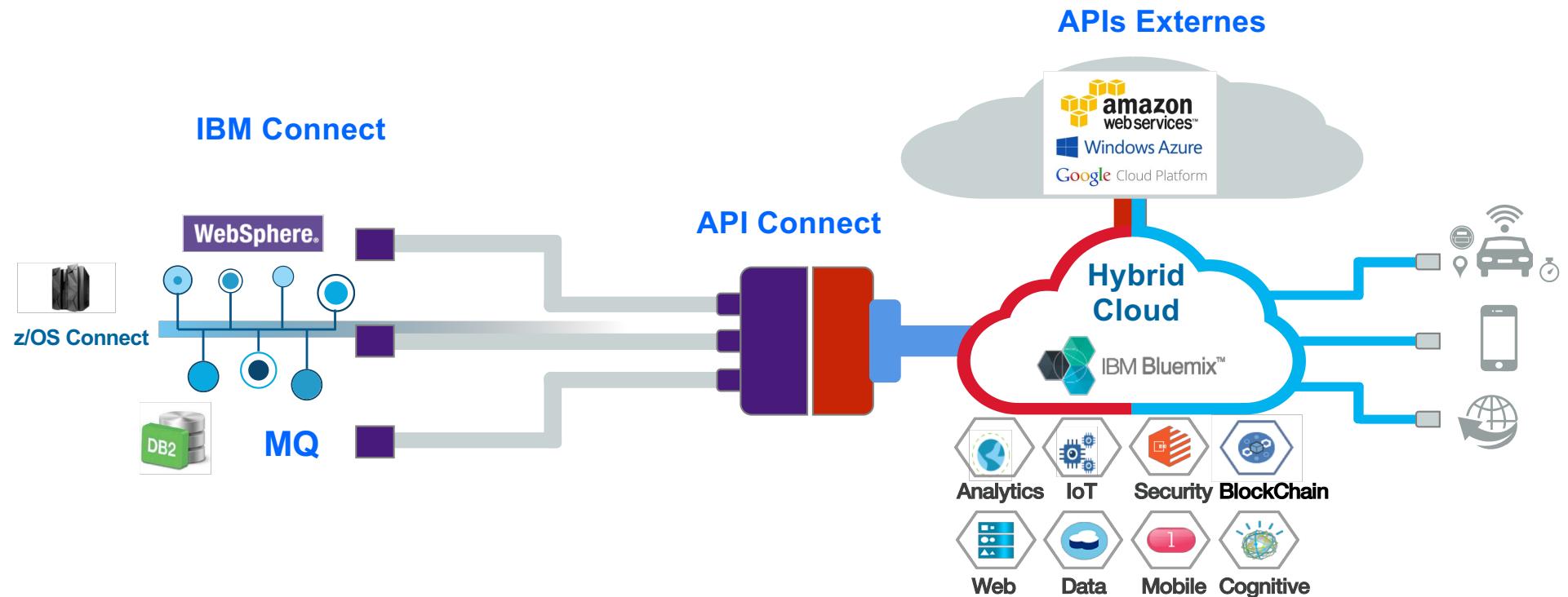
### Que propose API Connect ?

- Interface moderne pour créer, développer, assembler visuellement des APIs
  - Création et assemblage de Microservices en Node.js et Java
- Gestion des droits d'accès, d'organisation et et de publication des APIs
  - Analyses de l'usage des APIs et portail pour les développeurs
  - Modèle de sécurité et de contrôle des APIs

# API Connect est clé pour la transformation numérique

IBM.

IBM rend le Cloud hybride concret



# Grande distribution



## Business Challenge

- Connecter les magasins au site e-commerce : Click and Collect
- Digitalisation des magasins

## Solution

- API Connect
- Mise en œuvre : 4 semaines

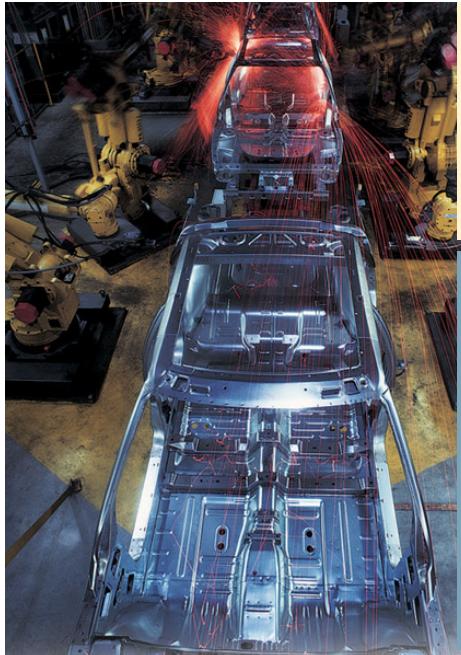
## Résultats

- Application tablette en boutique
- Omni Canalité

## Types d'APIs

- Catalogue produit
  - Liste de prix
  - Accès au stock
  - Omni canalité

# Automobile : PSA



## Business Challenge

- Ouvrir les APIs aux partenaires pour développer de nouveaux services
- Monétiser les données du véhicule

## Solution

- API Connect
- IOT for Auto
- Services GBS

## Result

- Nouvelles sources de revenus
- Image d'entreprise innovante

### Types d'APIs

- Location based services
- Sensor data to warranty systems
  - Alerts to dealers
- Service record access

# Telecom



## Business Challenge

- Se démarquer de la concurrence
- Fédérer un écosystème de partenaire
- Trouver de nouvelles sources de revenu

## Solution

- Bluemix et API Connect
  - Hackathon

## Result

- Projets innovants monétisables
  - simulation de présence
  - Maison connectée
  - Sécurité des particuliers

### Types d'APIs

- Payment options (credit/debit)
  - Partner Loyalty Programs
- Merchant specific promotions
- Access to payment account information

# APIs : Besoins du consommateur



Du point de vue du Consommateur : le développeur

Consommer des APIs **RESTful, autodécouvrables, agnostiques**  
de la technologie sous-jacente

Le verbe HTTP indique  
l'action/opération  
effectuée sur la ressource

L'URI indique une ressource  
(collection/member)

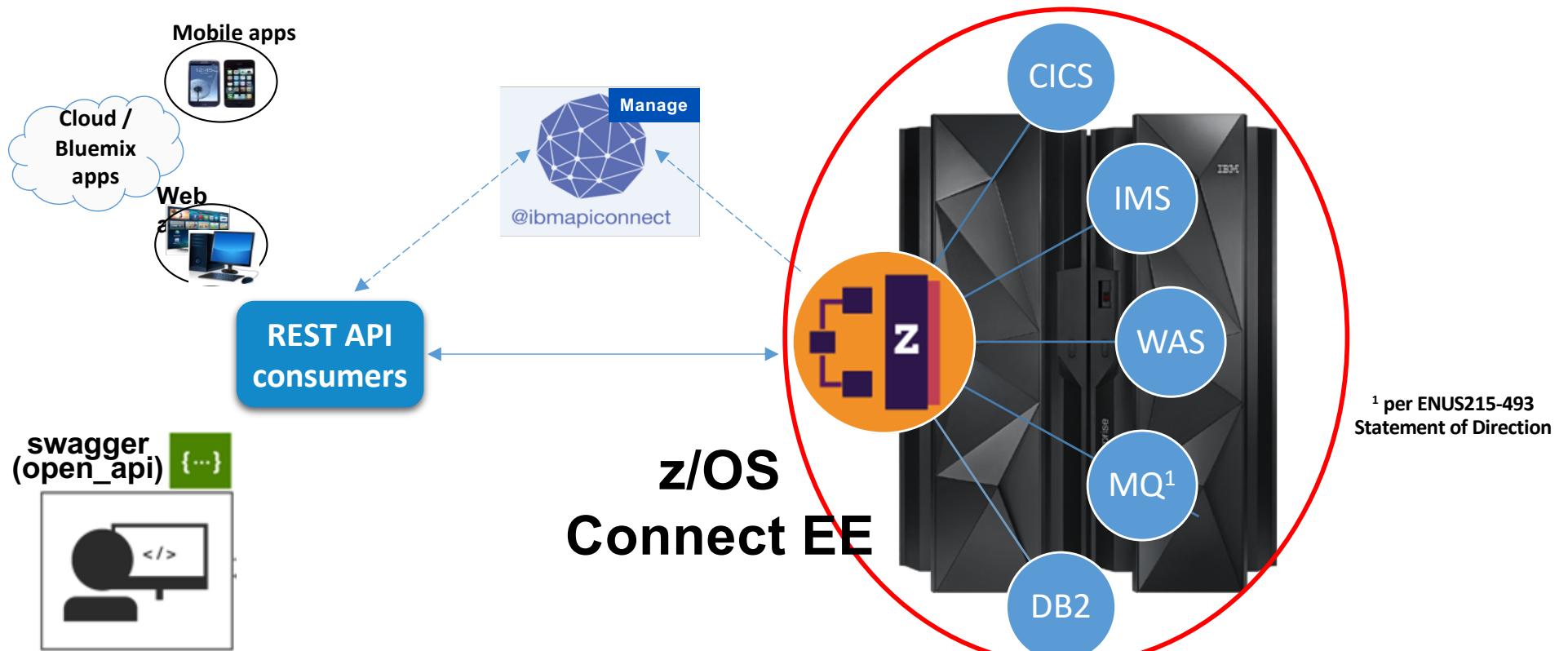
Paramètres de requête  
pouvant être fournis pour  
identifier la ressource

GET  
POST `http://<serveur>:<port>/path?name=value&name=value`  
PUT  
DELETE



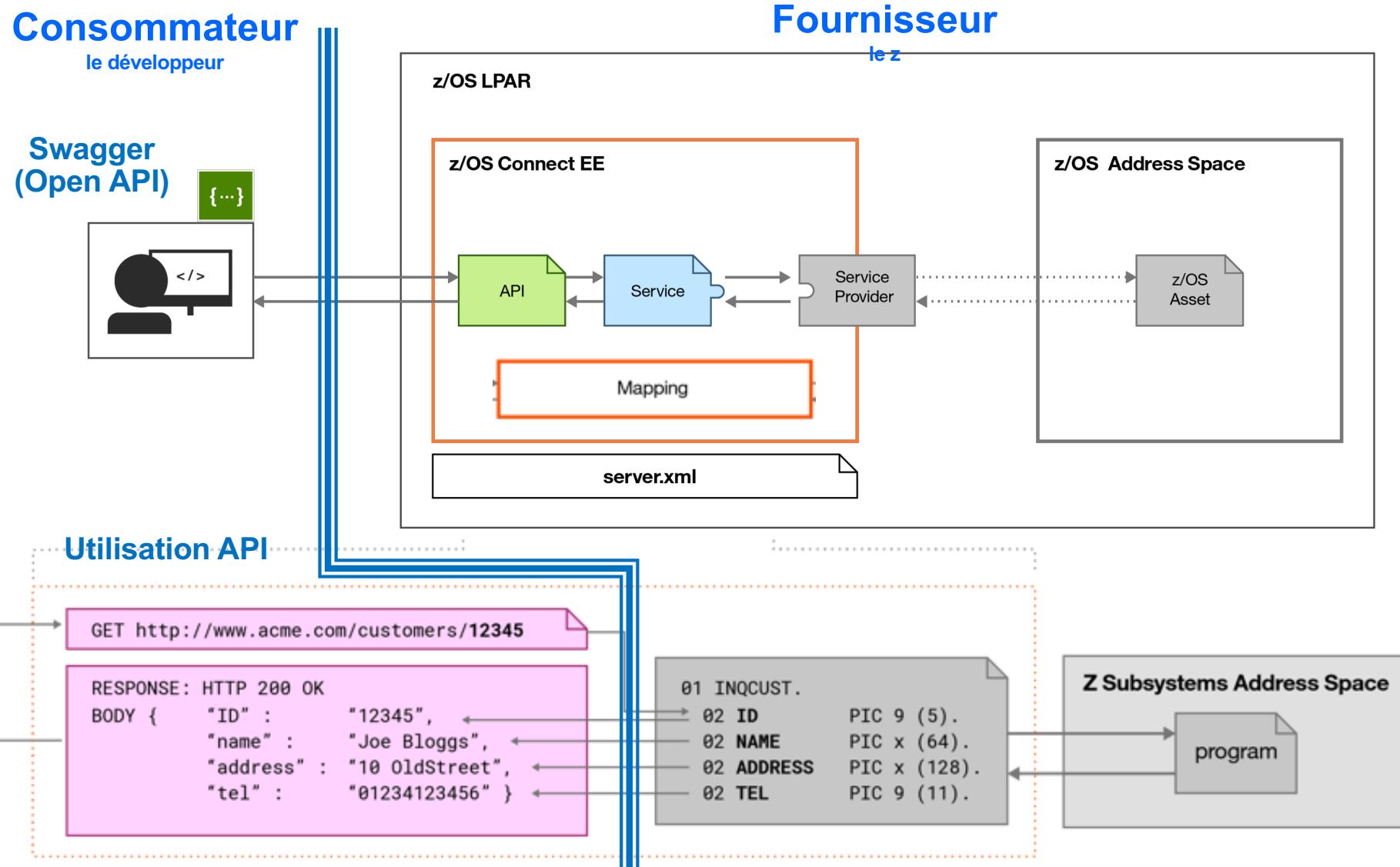
## Du point de vue du Fournisseur : le z

Exposer les assets z (CICS, IMS, DB2, ...) sous forme d'API  
de façon unifiée en intégrant la sécurité, l'audit et l'évolutivité  
Sans modifier les programmes existants

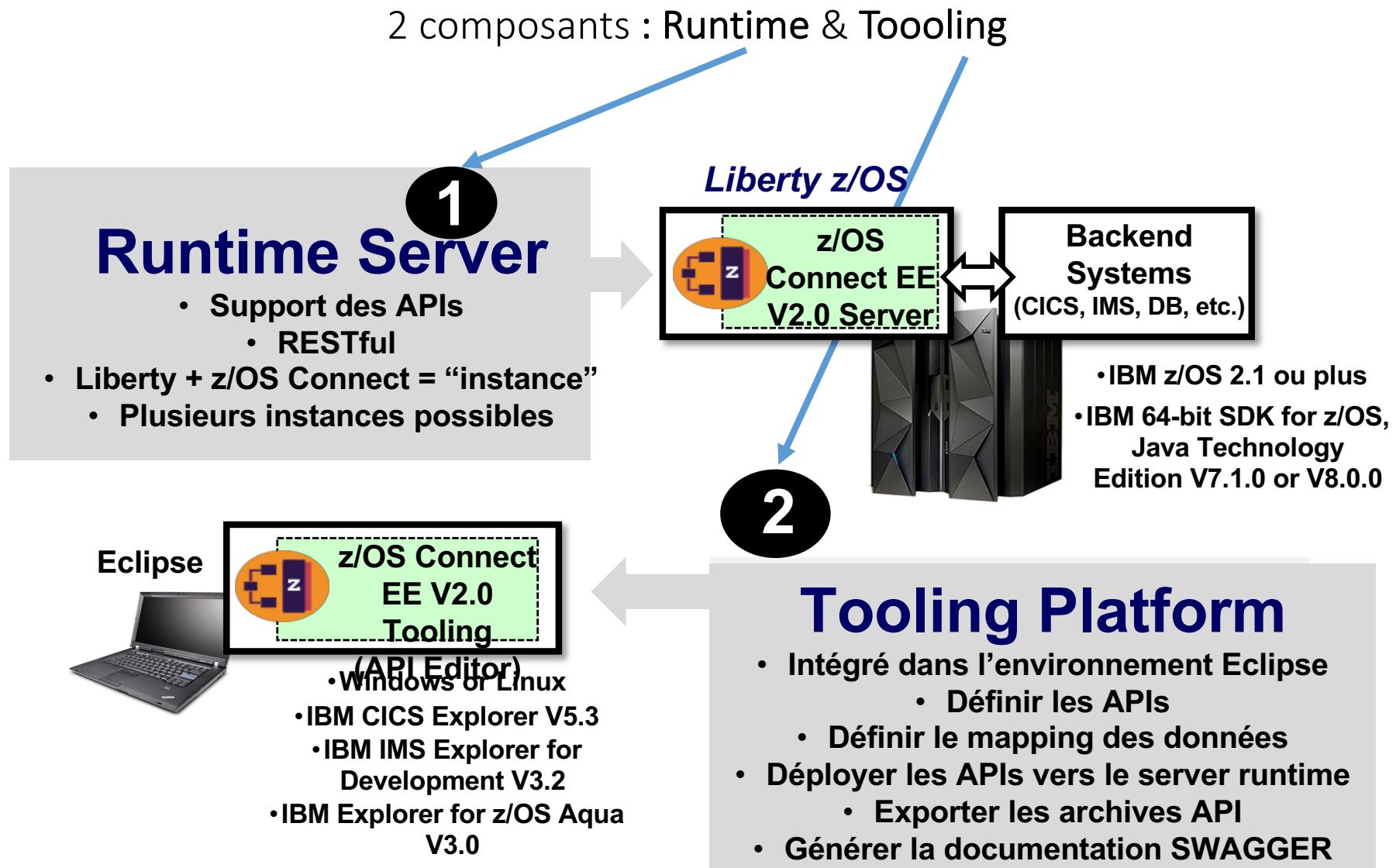


# Consommer/Exposer des APIs z

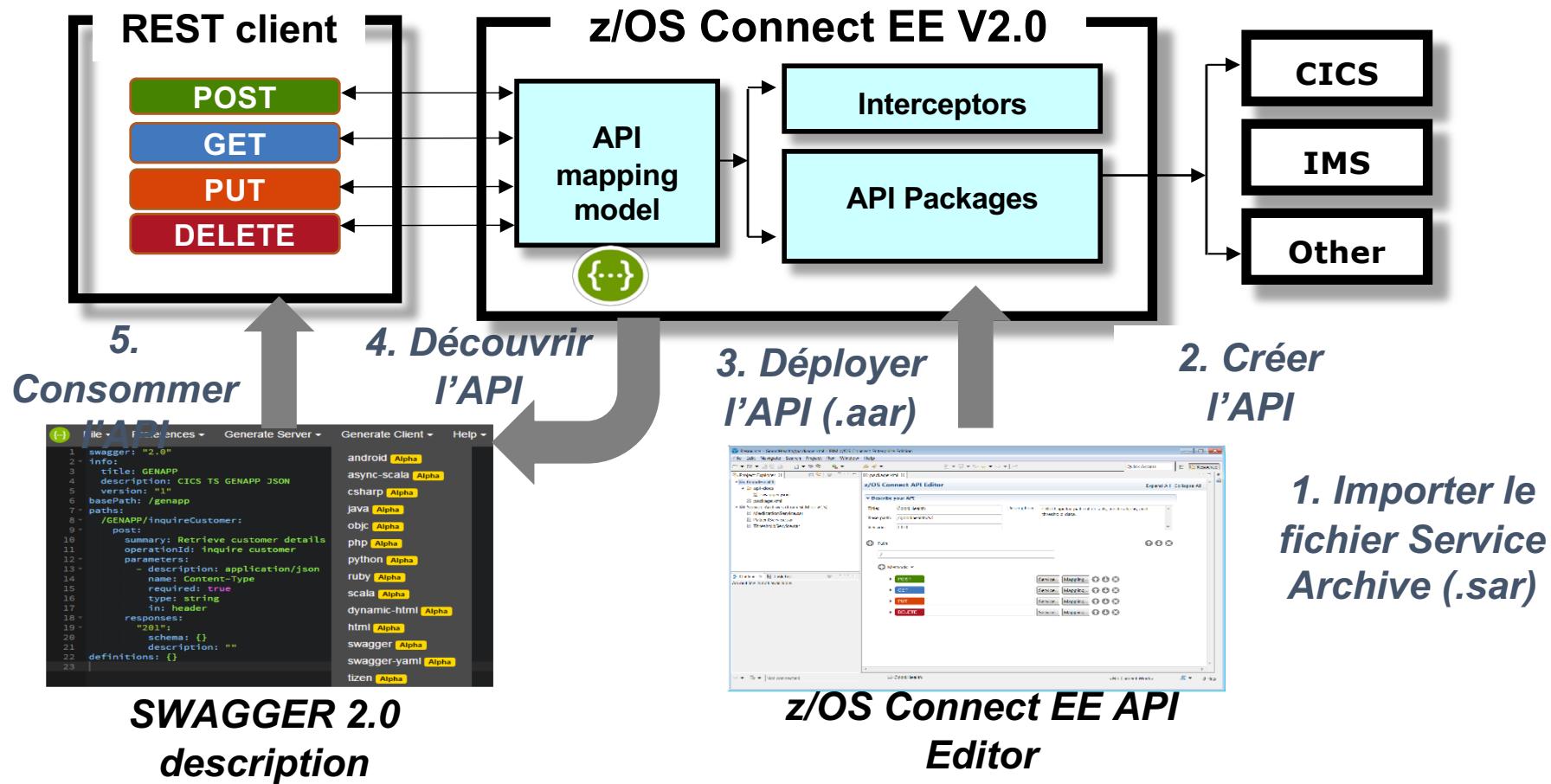
IBM



# z/OS Connect Enterprise Edition



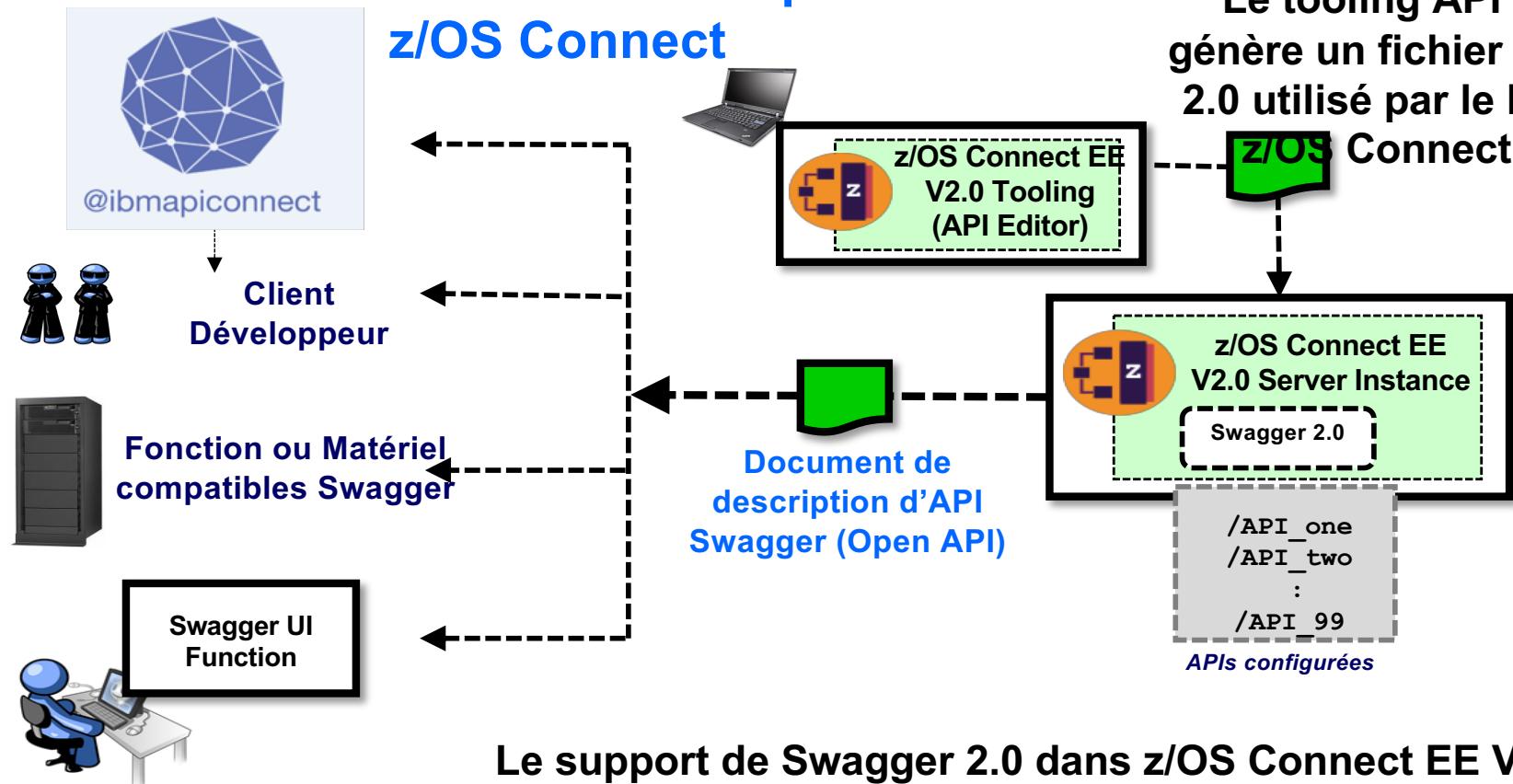
# API z – Vue générale



# Découverte des APIs



Les APIs peuvent être découvertes via les documents SWAGGER\* fournis par le serveur z/OS Connect



Le support de Swagger 2.0 dans z/OS Connect EE V2.0 permet de standardiser l'échange d'informations sur les APIs, ce qui assure une compatibilité avec un ensemble de fonctions et matériels

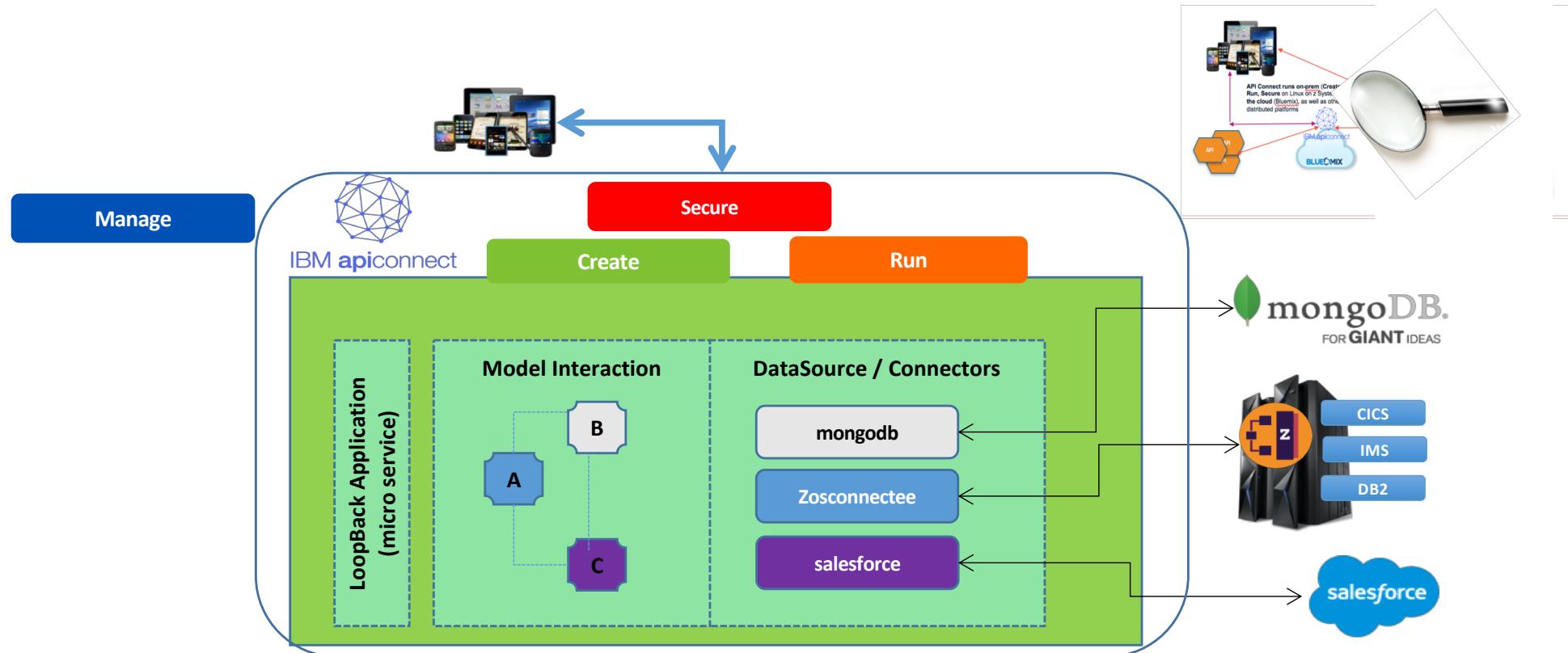
\* SWAGGER ou Open API



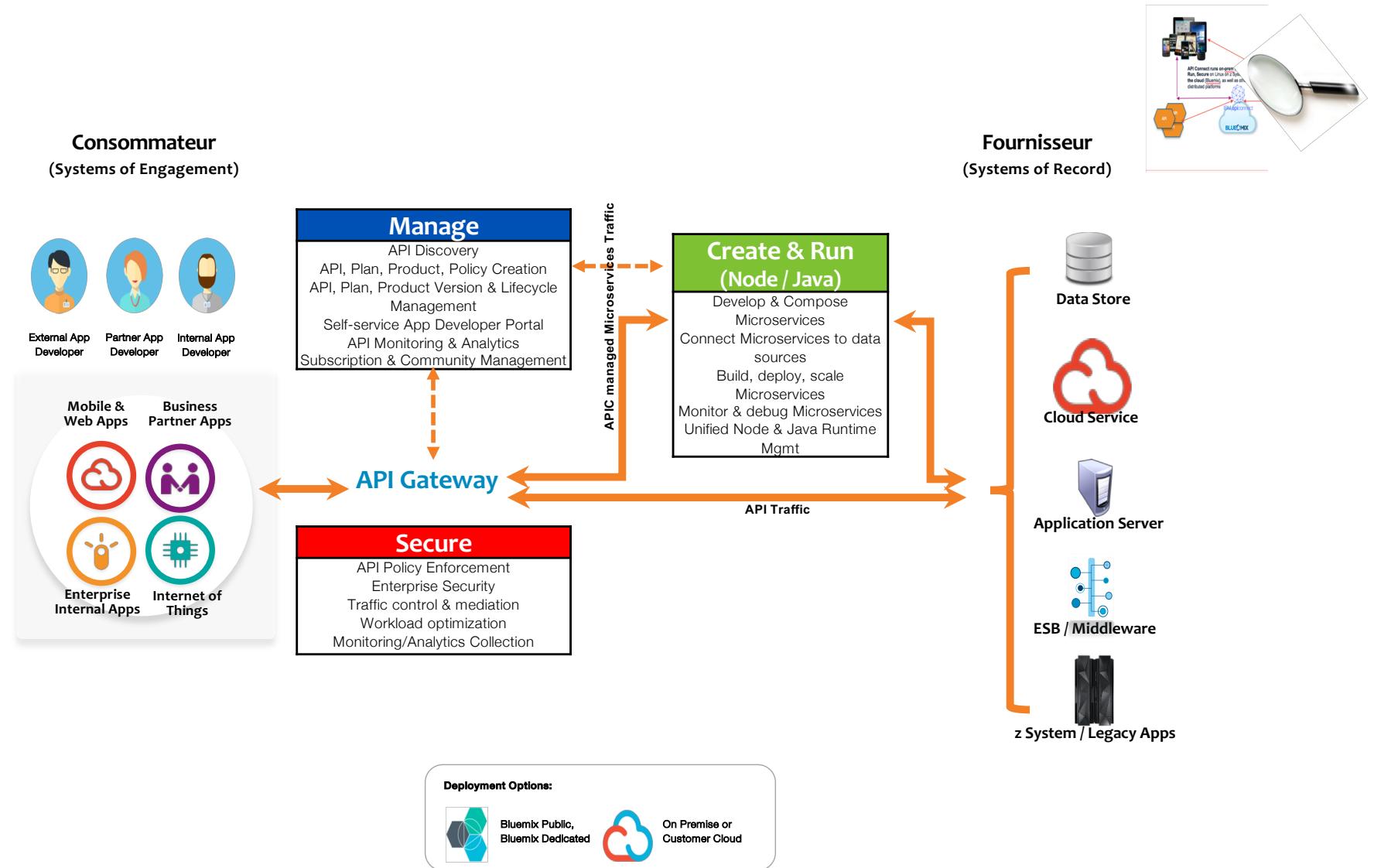
# IBM API Connect

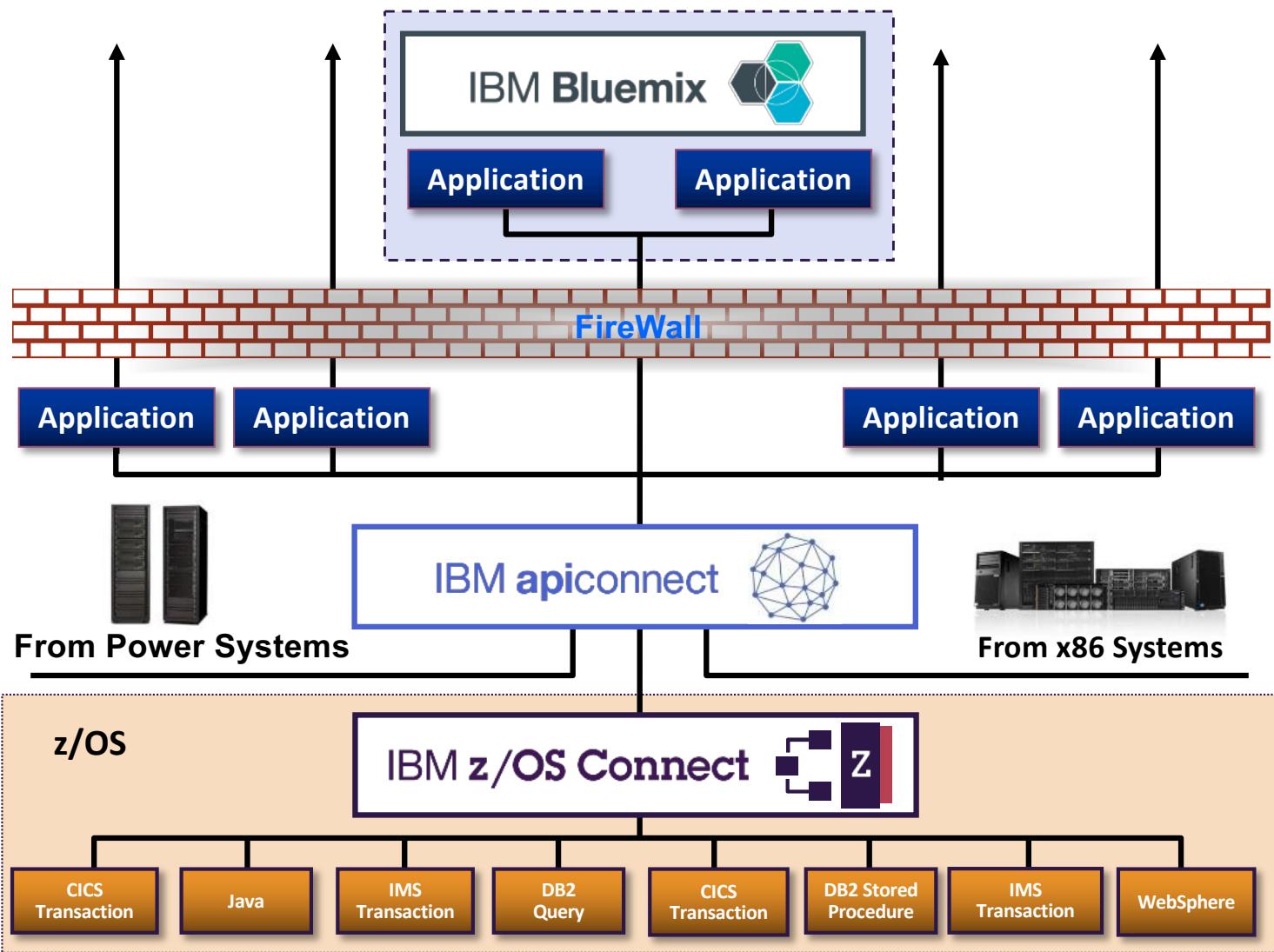
*La solution stratégique d'IBM pour gouverner, sécuriser, Interagir, exposer vos APIs*

# Créer des applications composites dans IBM API Connect



# Quels sont les composants de API Connect?





# En Résumé : Le MainFrame et l'API Economy



Le Mainframe fait partie intégrante de la nouvelle économie des APIs.

On peut exposer les programmes et les données basés sur z/OS afin de les mettre à disposition des applications « Mobiles » et « Cloud ».

IBM propose aujourd'hui plusieurs solutions :

## IBM z/OS Connect EE :

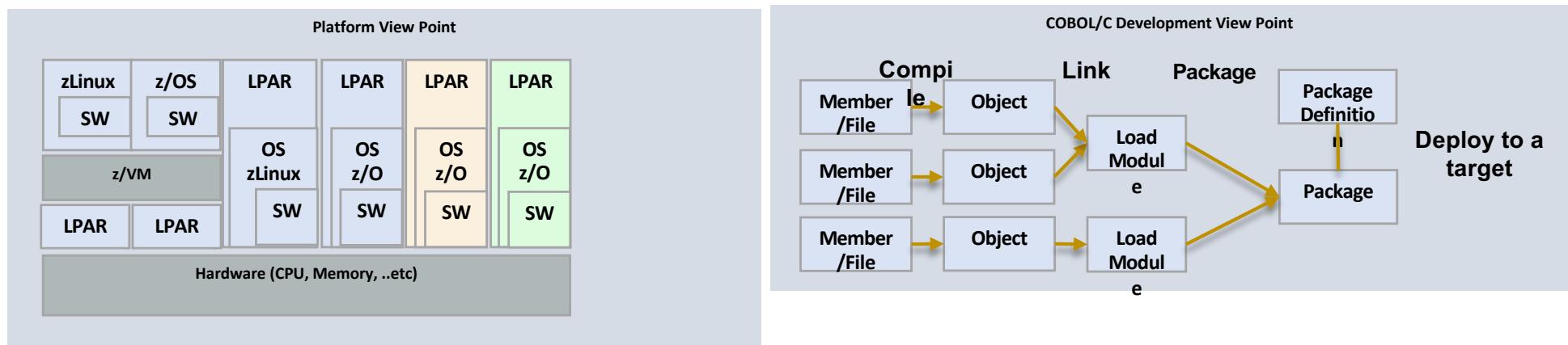
- *Solution, exclusivement z/OS, permet d'accéder aux sous-systèmes z/OS tels que CICS, IMS et Batch, en utilisant les API RESTful avec des messages au format JSON.*
- *Cette infrastructure offre un accès simultané(via une interface commune) à plusieurs sous-systèmes z/OS.*
- *Elle permet de faire bénéficier les entreprises de deux types d'avantages.*
  - ❖ Fourniture d'outils intuitifs basés sur le poste de travail du développeur, avec ou sans compétences z/OS, pour créer des API RESTful à partir d'actifs z/OS traditionnels.
  - ❖ Incorporation des données et transactions z/OS dans les applications mobiles et en cloud sans comprendre les méandres du sous-système z/OS.
  - ❖ Les ressources z/OS se présentent comme n'importe quelle autre API RESTful.

## IBM API Connect.

- Offre d'entreprise destinée à la gestion, à l'exécution et à la sécurisation des APIs (toutes plateformes confondues). L'une des grandes fonctions d'API Connect est également la création d'APIs d'interaction, permettant l'agrégation de différentes APIs réparties dans l'entreprise.
- Technologies basées sur l'open sources telles que Node.js, et pouvant donc s'exécuter dans un environnement "Linux on z Systems", garantissant une interaction forte et efficace avec z/OS Connect.

IBM Cloud Technical Academy

# System Z Concepts – What you need to know 1/4



# System Z Concepts – What you need to know 2/4

- Operating Systems running on System Z
    - z/OS, z/VSE, z/VM, z/TPF, z/Linux
  - COBOL
    - Development Language used for legacy mainframe application
    - Many z/OS applications are/were developed with COBOL
  - CICS
    - Middleware family of mixed language application servers designed to support rapid, high-volume online transaction processing
    - CSD → CICS System Definition
    - CICS Transaction Server is a family of scalable general-purpose transaction processing solutions for z/OS®
    - CICS Configuration Manager is a tool for administering and maintaining CICS resource definitions
- 
- **TSO (Time Sharing Option):**
    - Multi-User support on z/OS that allows more than one user to use the system in the same time
  - **ISPF (Interactive System Productivity Facility):**
    - A collection of menus and panels to interact with z/OS
  - **LPAR (Logical Partition)**
    - A slice of hardware resources where you can install a mainframe OS on it and runs independently.
  - **DATA SET**
    - Can be expressed as a file in a hierarchical file system, it resides on z/OS side.
  - **PDS/PDSE (Partitioned Data Set/Extended)**
    - Is like a folder and it contains data set members.

# System Z Concepts – What you need to know 3/4

- JCL (Job Control Language)
  - Scripting language used on System z to create run batch jobs
  - Batch jobs can be used to automate tasks or even start a subsystem
- REXX (Restructured Extend Executor)
  - Structured interpreted programming language
  - Often used for processing data and text and generating reports.
- USS (Unix System Services)
  - **A component of z/OS operating system**
  - **Officially certified UNIX system that allows z/OS users to work on System Z in a fully functional UNIX environment including using hierarchical file system and running UNIX process on Z**
  - **z/Linux (Linux on z Systems) is the collective term for the Linux operating system compiled to run on SystemZ.**
    - It can run natively or in an LPAR or under z/VM

# System Z Concepts – What you need to know 4/4

- **SYSPLEX (Systems Complex)**
  - Something like OS cluster with z/OS
  - Allows authorized components in up to 8 LPARs to communicate and cooperate with each other using XCF Protocol
- **Significant Deployment Type on System Z**
  - **Incremental**
    - Typically for DB2 on z/OS and CICS clients where thousands of applications are running.
    - Only changed programs need to be deployed
    - Rollback is required for almost all clients
    - Some programs have not been recompiled for years
  - **Batch Processing**
    - Typically for programs, DBRM (Database Request Modules), JCLs ...etc
    - There is deployment process like BIND for Database Request Modules
  - **Deploy Applications on USS or z/Linux**
    - Same semantics like on any other UNIX/Linux system

- RESTful APIs – A common, industry standard, platform independent way of calling a business service (REST)
- Bluemix – IBM's Cloud-based Application Development and hosting environment
- API Management – A simple way to manage all those 1000s of internal and external RESTful APIs that you have in the enterprise, and making them visible in Bluemix !
- API Connect – A product that does API Management
- z/OS Connect Enterprise Edition – A product that allows you to connect from outside the z environment INTO the z environment, working as part of API Management
- CICS - Customer Interactive Control System  
One of those applications on z that you want to provide access to – it's a runtime environment
- IMS - Information Management System  
Another one of those applications you want to get access to – it's a runtime environment and database environment
- DB2 – one of the premier relational databases that works with CICS and IMS
- MQ - Message Queueing  
Similar capability to alternatives like RabbitMQ, ActiveMQ. This is their grandpa.

IBM

धन्यवाद

Hindi

多謝

Traditional Chinese

Спасибо

Russian

شُكْرًا

Arabic

Grazie

Italian

நன்றி

Tamil

*Thank You*

English

Merci

Français

ありがとうございます

Japanese

ขอบคุณ

Thai

Gracias

Spanish

Obrigado

Brazilian Portuguese

Danke  
German

多谢

Simplified Chinese

감사합니다

Korean