Mobilité, APIs, Système z

Régis David

zChampion Mobile, expert écoCICStèmes, expert SOA

regis_david@fr.ibm.com

Who are your developers?
Anyone
What is an application?
Anything
Who can access your information?
Everyone
Who is influencing your business?
Anyone
Where do transactions happen?
Everywhere

Agenda

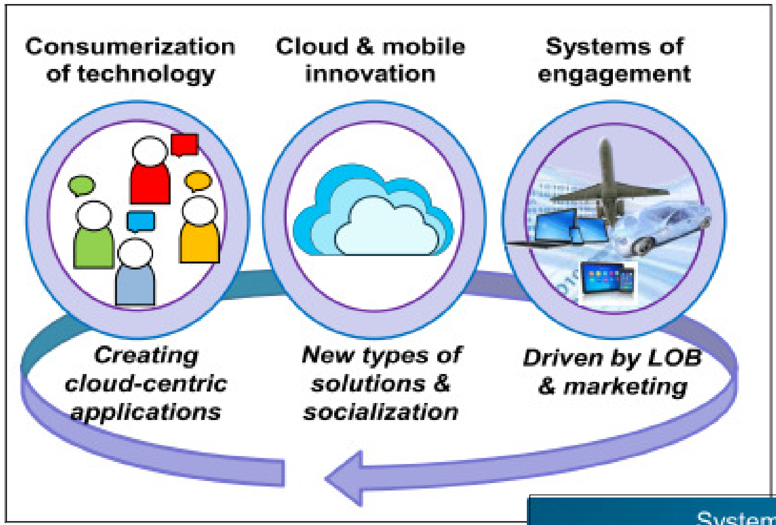
- Mobile, Cloud, you know what ? I'm API
- APIs et System z
- Ouf le System z est API/mobile compatible
- On s'en SoR très bien sur Z, et on s'engage selon vos SoE…

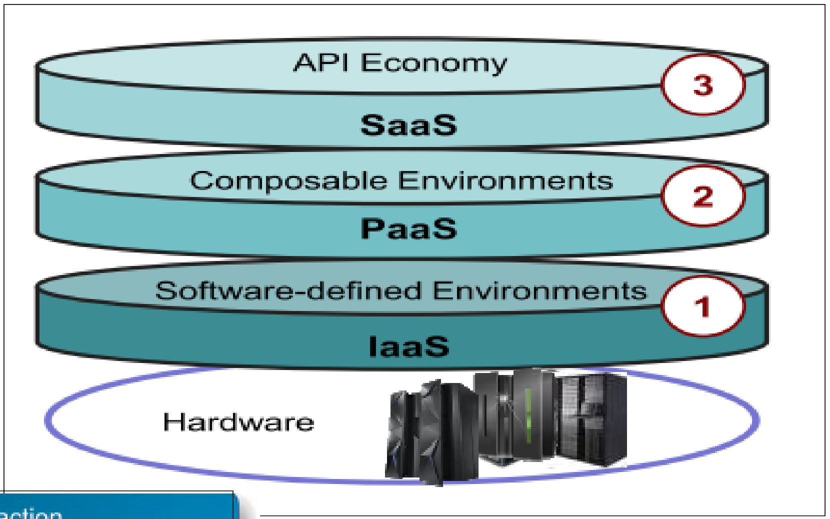
Ce qui est montré ici, votre serviteur l'a fait, le fait ou le fera

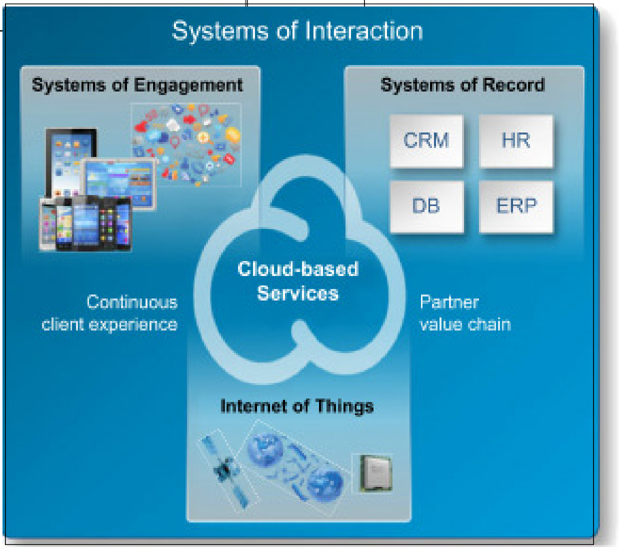
Mobile, Cloud

You know what ? I’m API

Mobile in context: new IT and business environment


Consumerization of technology
Cloud & mobile innovation
Systems of engagement
Creating cloud-centric applications
New types of solutions & socialization
Driven by LOB & marketing





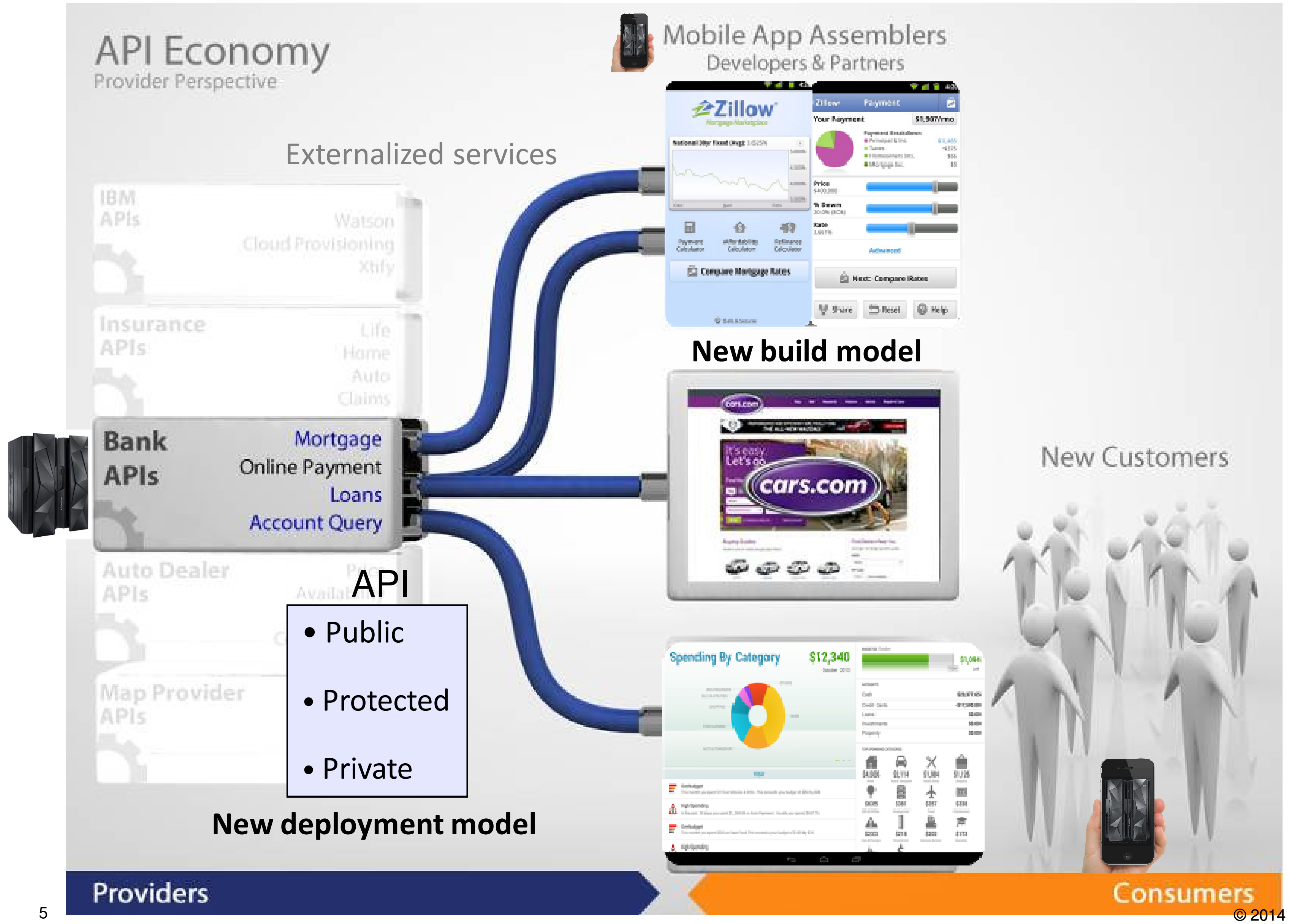
- *Engage* with *consumers*
  - ***Service* is not enough**
- Applications
  - *Born-on-the-cloud*
  - ***At a mobile speed***
  - ***Polyglot programming***
- *Ecosystem of services*

- *Service consumption economy*
- **Business services**
- Composite services composition
  - ***Simple web APIs***
  - Cloud, data, social, Mobile Apps
- Open standards & technologies
  - ***Emerging<>mature***

Ref: Redpaper 'Accelerate Development of New Enterprise for the cloud with Codename Bluemix
http://www.redbooks.ibm.com/abstracts/redp5011.html?Open

Not having an API today is like not having a website in the 1990s…

stores
(800) ###s
web sites
Web APIs
IBM



APIs just a new name for Smart SOA ?

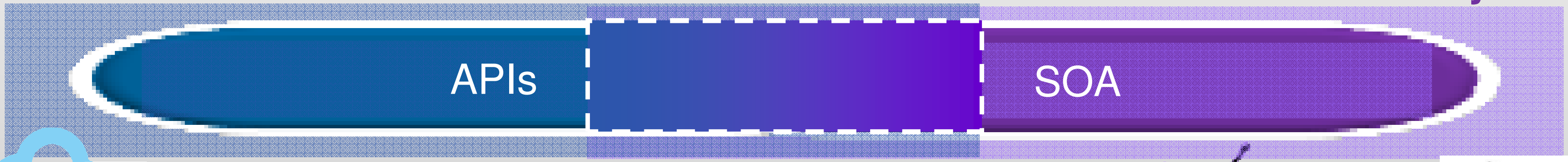
Many similarities … One important difference: the objective behind

Developer centric

“How can I increase the **pace of innovation**?”

IT service centric

“How can I increase the **interoperability** and **effectiveness of delivery**?”

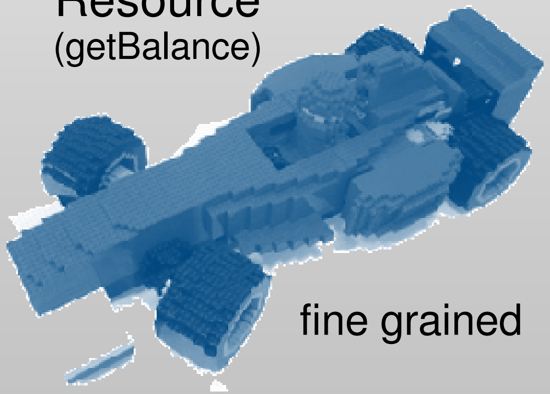


REST

JSON

‘human’

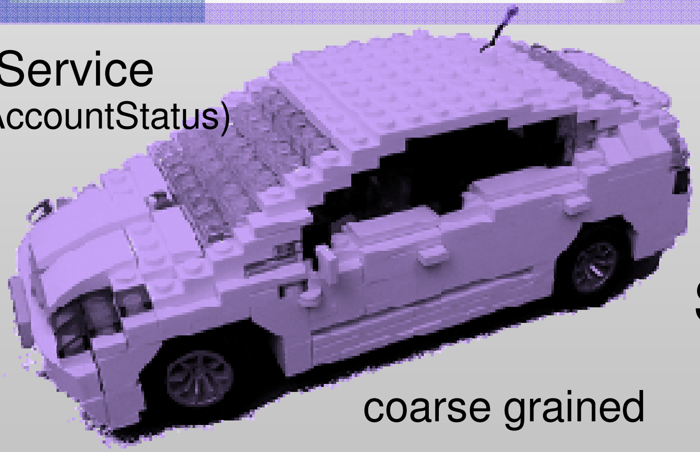
Resource
(getBalance)



fine grained

Speed to deliver
Expediency
Less to learn

Service
(getAccountStatus)



coarse grained

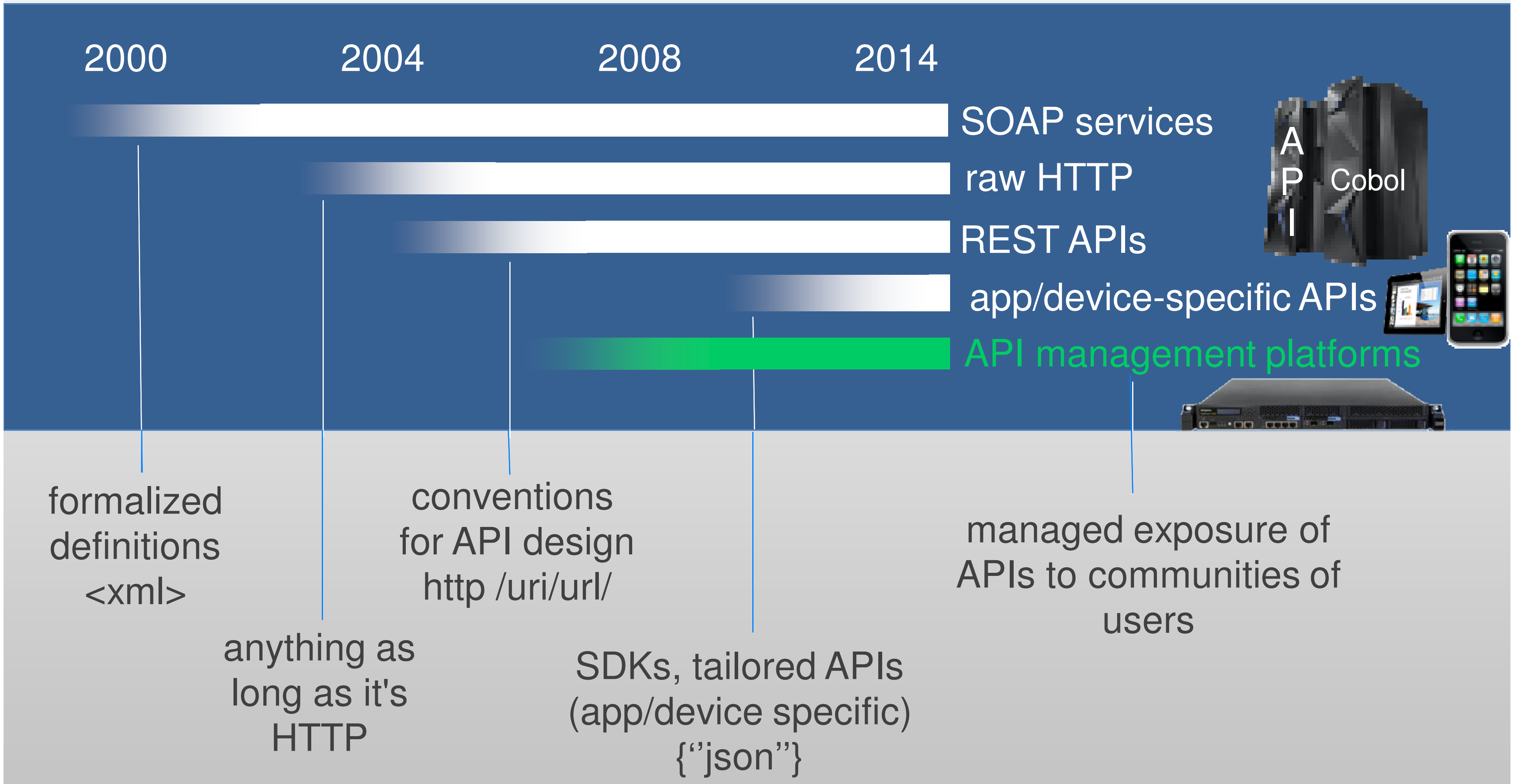
Effort to deliver
Effectiveness
Less to change

SOAP

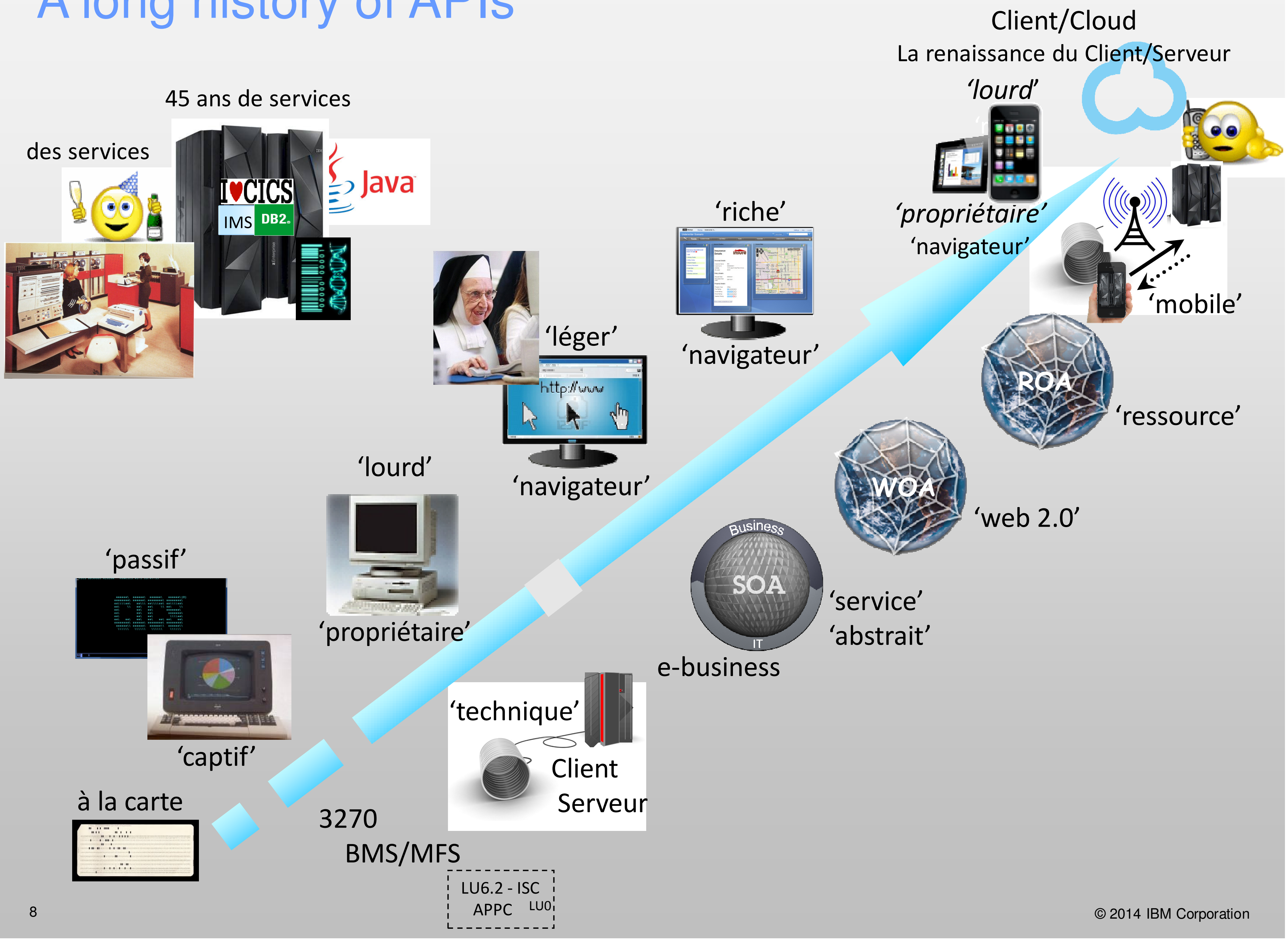
XML

WSDL

A brief history of APIs


2000
2004
2008
2014
SOAP services
raw HTTP
REST APIs
app/device-specific APIs
API management platforms
A
P
I
Cobol
formalized definitions <xml>
anything as long as it's HTTP
conventions for API design http /uri/url/
SDKs, tailored APIs (app/device specific) {"json"}
managed exposure of APIs to communities of users

A long history of APIs



Technology: REST(Ful) – REST/RPC – Lo/REST, Hi/REST…

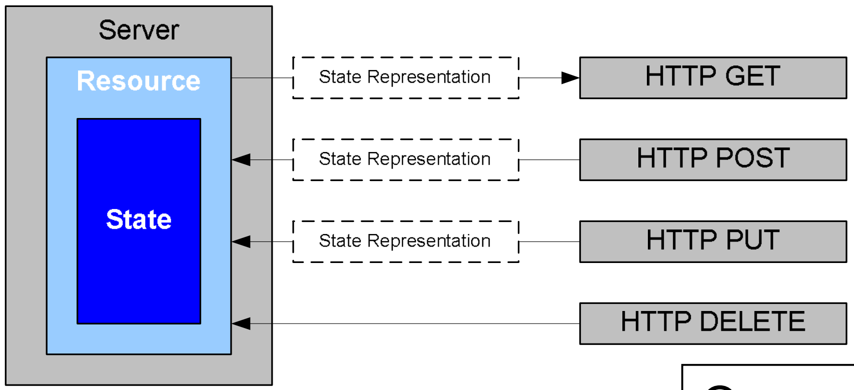
HTTP centric

Method = action

URL = resource abstraction

HTTP header = metadata

HTTP status



The request Body is (often) empty

URI conventions

…/Collection

…/Collection/Member

Query string

…/…?parm=value&parm=value…

Conventions		LCRUD 'operations'
GET	…/Collection	LIST
POST	…/Collection(/Member)	UPDATE/CREATE
GET	…/Collection/Member	READ
PUT	…/Collection/Member	CREATE/UPDATE
DELETE	…/Collection/Member	DELETE

HEAD	get resource metadata, no body
(OPTIONS)	supported methods
PATCH	emerging 'partial update'…

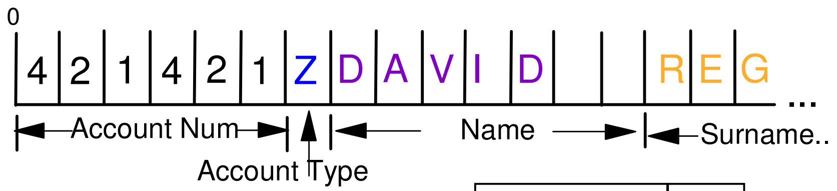
WADL…one day…

Reprentational … always ? i.e: list response…

Technology : message representation

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

- Bytes
  Tight



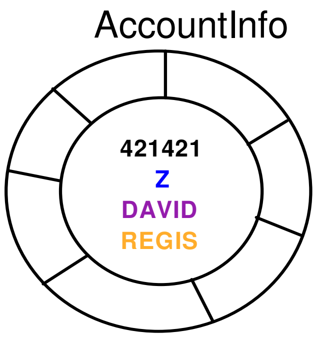
exclusive

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

COBOL	*IT*

- Object
  Permissive

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



getAccountInfo(…)
getAccountType(…)
getCustomerName(…)
etc..

« flexible »

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

bytes[];

Java	*Model*

- XML
  Defined

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

schema	*wsdl*

Services	*Interoperability*

adaptive

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

- JSON
  Emerging

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

schema	*wadl?*

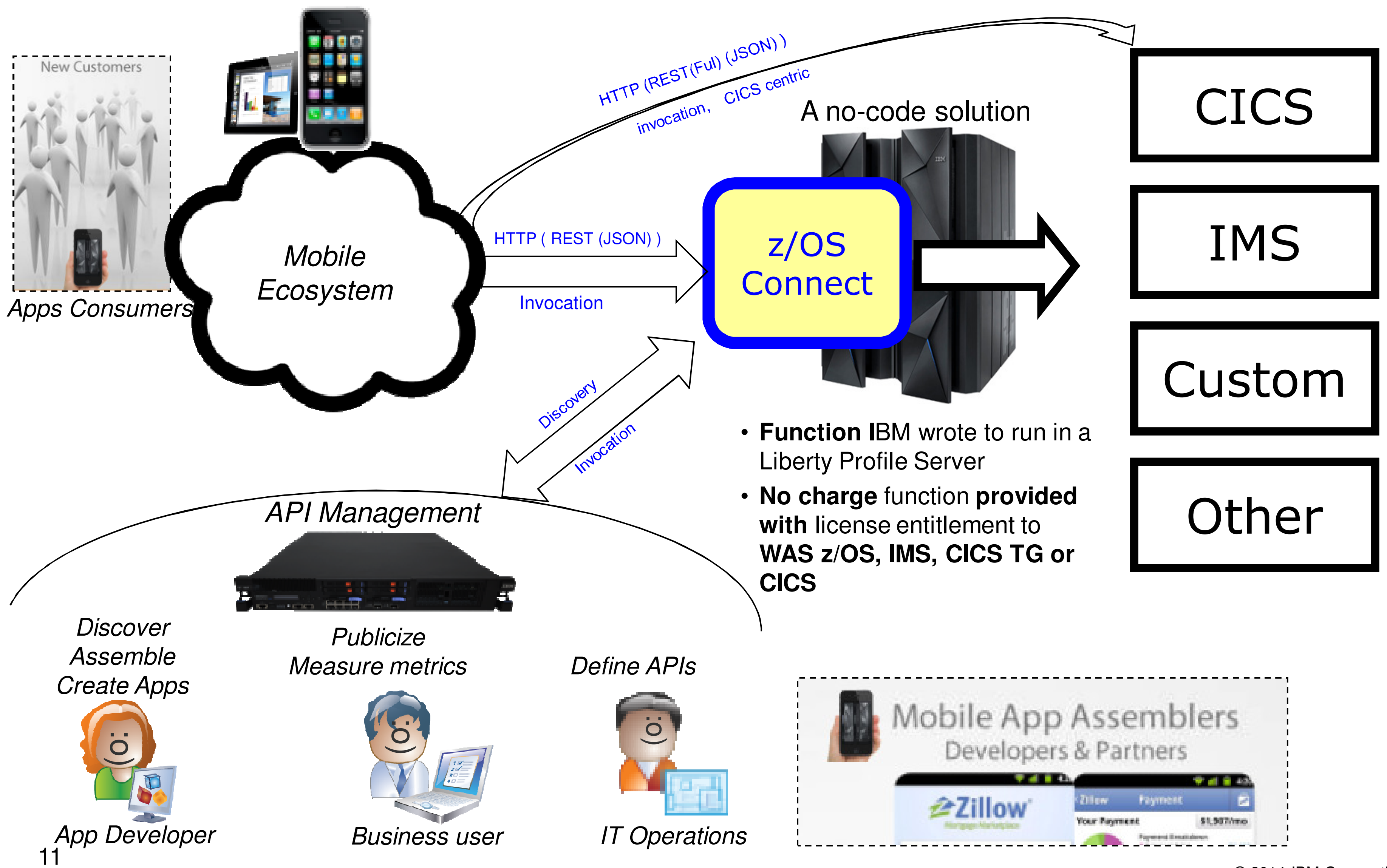
Web 2.0	*Simple WS*

Javascript

```
{“$schema":"http:\/\/json-schema.org\/draft-04\/schema#",
  {  "type":"object",
      "properties“ : {
        “ggx082_kztpac_cn“ : {
          "type“ : "string",  "maxLength“ : 9 }
                }
  }, …
```

z/OS Connect

A unified REST/JSON portal to your z/OS



- **Function** IBM wrote to run in a Liberty Profile Server
- **No charge** function **provided with** license entitlement to **WAS z/OS, IMS, CICS TG or CICS**

System z strategy for mobile…and web APIs…

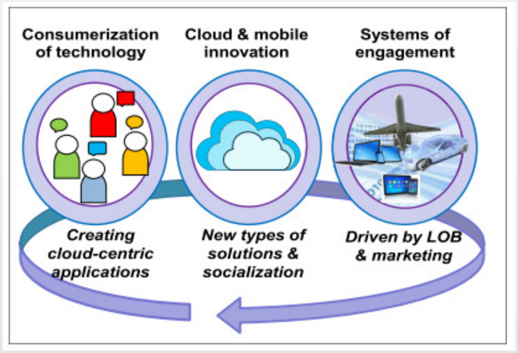
APIs: REST for the exchange protocol, JSON for the message representation

zMiddleware share common code for both runtime and tooling

- Strategic **JSON** <> Binary representation (i.e. Json <> Cobol)
  - Currently CICS 'wsbind' transformations technology
    - CICS TS V4.2 and above
    - CICS TG V9.1
    - z/OS Connect WebSphere Liberty repository feature
  - Common tools and RDz support generate the 'same' wsbind binary
- Strategic **REST** handler listener (i.e. HTTP GET on URL /Collection/Member is a read)
  - z/OS Connect WebSphere Liberty repository feature
    - Four distributions: WebSphere AS, IMS, CICS and CICS TG
  - CICS centric alternative: CICS native, JAX-RS in CICS, or CICS TG
  - WebSphere AS on z/OS centric alternative: JAX-RS
- Strategic **Web API** service **discovery** listener
  - z/OS Connect WebSphere Liberty repository feature

A long history of APIs, revisited

- **Business services**
- Composite services composition
  - ***Simple web APIs***
  - Cloud, data, social, Mobile Apps
- Open standards & technologies
  - ***Emerging<>mature***


Consumerization of technology
Cloud & mobile innovation
Systems of engagement
Creating cloud-centric applications
New types of solutions & socialization
Driven by LOB & marketing

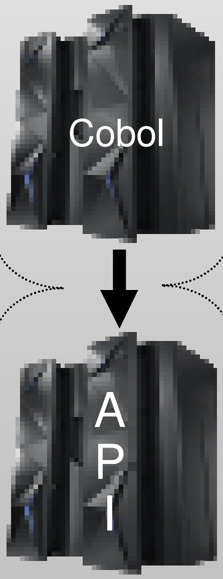
- Applications
  - *Born-on-the-cloud*
  - ***At a mobile speed***
  - ***Polyglot programming***

Des technologies stabilisées

✓ SOAP/XML
✓ REST/JSON
✓ HTTP
✓ Java

Un mainframe banalisé

✓ Java et JEE
✓ HTTP
✓ <xml>
✓ REST/JSON
✓ *SOAP/XML*
✓ Linux
✓ Eclipse

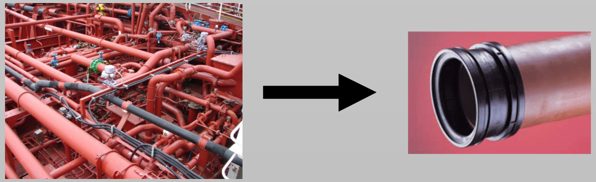


Le propriétaire expose ses services

✓ Sans intermédiaire
✓ A un coût maitrisé ⟶ Java ou pas
✓ Banalement
✓ A la granularité requise
✓ A la Qualité de Service requise (SLA)
✓ Avec l'exploitabilité en prime

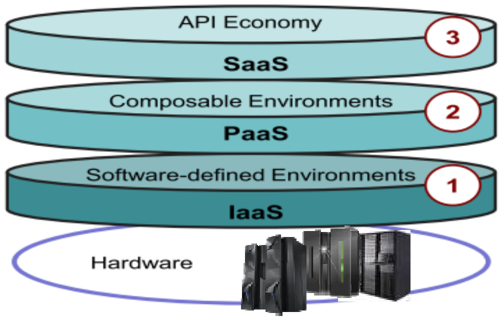
Simplification de l'architecture

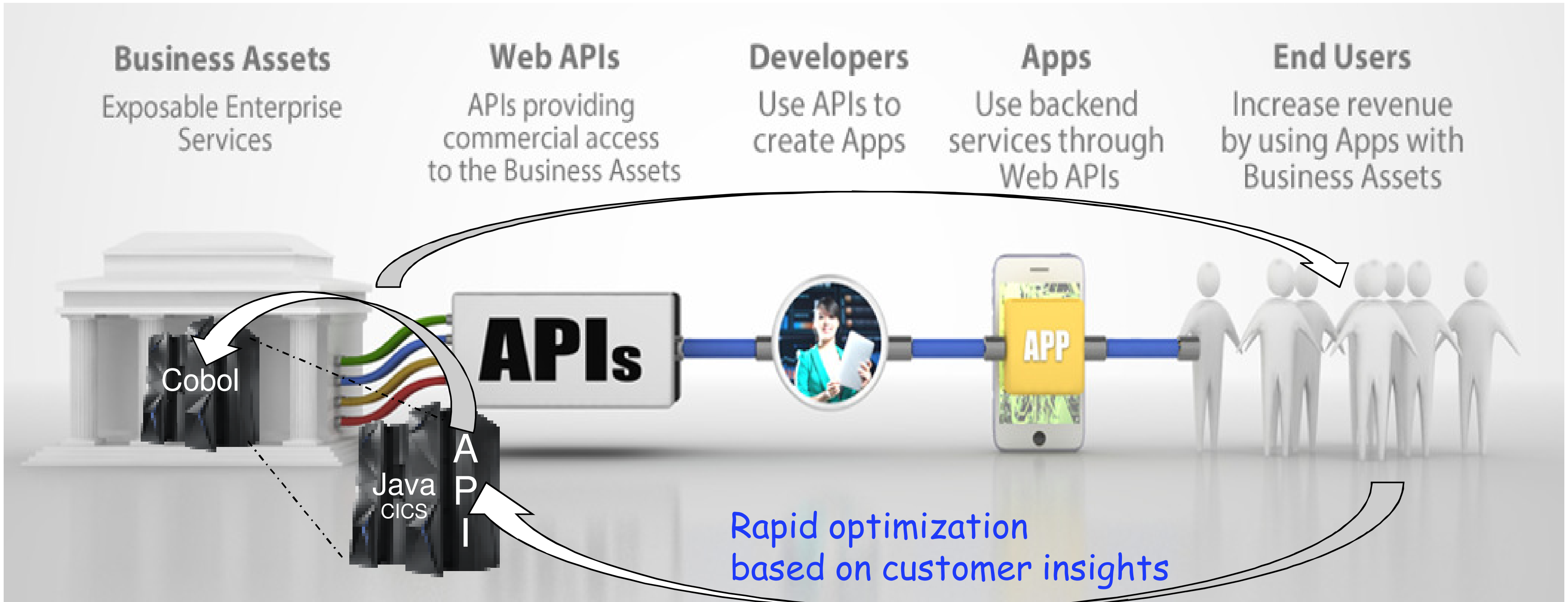
✓ Agilité « at a mobile speed »



✓ Pas d'intermédiaire 'inutile'
✓« One team, one silo, one budget »

API Economy supply chain




Business Assets
Exposable Enterprise Services
Web APIs
APIs providing commercial access to the Business Assets
Developers
Use APIs to create Apps
Apps
Use backend services through Web APIs
End Users
Increase revenue by using Apps with Business Assets
Cobol
APIs
APP
Java
CICS
API
Rapid optimization based on customer insights

The API Economy

Where companies [providers] expose their (internal) digital business assets or services in the form of (Web) APIs to third parties [consumers] with the goal of unlocking additional business value through the creation of new assets

Complexity and complication

- ✓ **Le problème peut paraître complexe**

- ✓ **La solution doit répondre aux critères du mobile et de l’API economy**
  - ✓ **Evolutions, optimisations à la vitesse requise**
  - ✓ **Time to market , coûts maîtrisés**

- ✓ **Elle doit donc être simple**
  - ✓ **Facile à comprendre, à exécuter**

- ✓ **Donc éviter toute complication**
  - ✓ **Elément qui entrave la simplicité d’un déroulement**

- ✓ **Nos organisations sont compliquées**
  - ✓ **Nos architectures sont à leur image**

Complexity and complication…

- ✓ **La complication induit une perte d'autonomie, de pouvoir**
  - ✓ **Compensés par des processus et contrôles renforcés**
  - ✓ **Elle favorise une volonté d'indépendance, d'autonomie**
    - ✓ **En résultent des silos, une collaboration par obligation**
      - ✓ **De la redondance et des délais, donc des coûts inutiles**
    - ✓ **Alors que notre problème requiert de la coopération (but commun)**

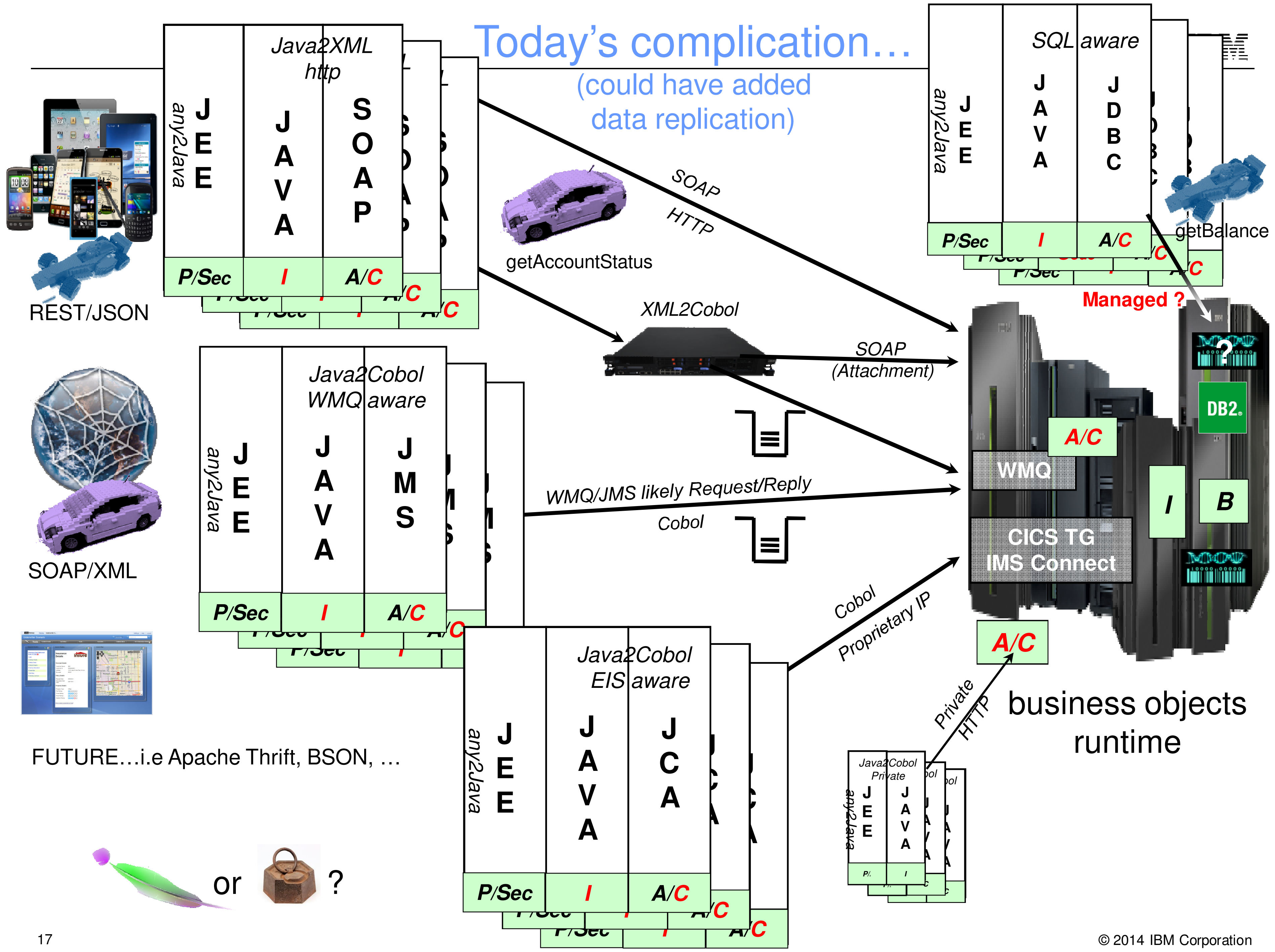
- ✓ **La complication induit une perte de vision globale**
  - ✓ **Donc de connaissance de la réalité**
  - ✓ **Et paradoxalement une perte de contrôle**

- ✓ **La solution simple au problème complexe requiert une simplification**
  - ✓ **Réforme qui induit une résistance des écosystèmes qui vivent de la complication**

- ✓ **Et…la simplification ne doit pas produire de la complication**
  - ✓ **Que pourraient induire des processus de contrôle de ses effets ;O)**

Today's complication…

(could have added data replication)



Simplification

- ✓ **La simplification part d’une vision objective de la réalité dans sa globalité**

- ✓ **La solution idéale apparait quand on ne peut plus rien retirer**
  - ✓ **Consommateur > Fournisseur**
  - ✓ **Retirer est anxiogène**

- ✓ **Et non quand on ne peut plus rien ajouter**
  - ✓ **Consommateur > Tiers/Silo 1 > … > Tiers/Silo n … > Fournisseur/Silo z**
  - ✓ **Ajouter est valorisant**

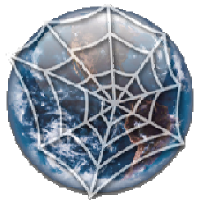
- ✓ **Le problème complexe à adresser:**
  - ✓ **Consommer des API z à partir de mobiles**

- ✓ **La solution simple**
  - ✓ **Consommateur > Briques industrielles dédiées à la QoS > Fournisseur z**

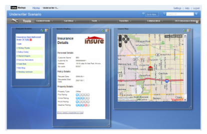
- ✓ **Une trajectoire simple**
  - ✓ **Vision objective de la valeur d’un tiers à la lumière des capacités d’aujourd’hui**
  - ✓ **Identification des facteurs de complication**
  - ✓ **Suppression de ceux-ci ou concentration sur un seul tiers**

Make it simpler, at low cost = CentraliZ

REST/JSON



SOAP/XML



FUTURE…

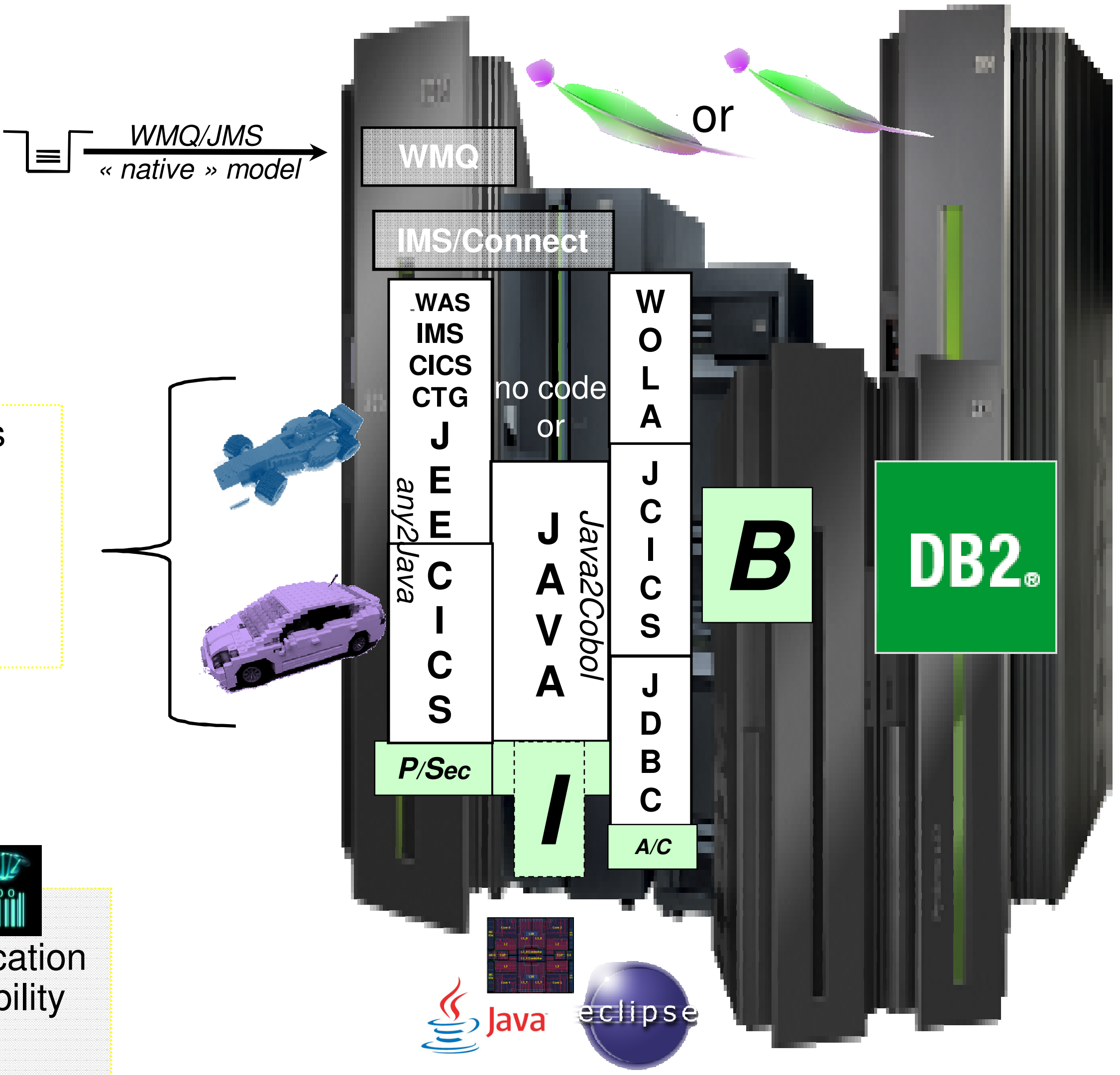
Room for QoS tiers

ESB
BPM
Datapower
…

zAble…
Secured by colocation
Workload traceability
Accountable
SLA compliant

Yes, even for JDBC

zAAP/zIIP/VUE/MWP

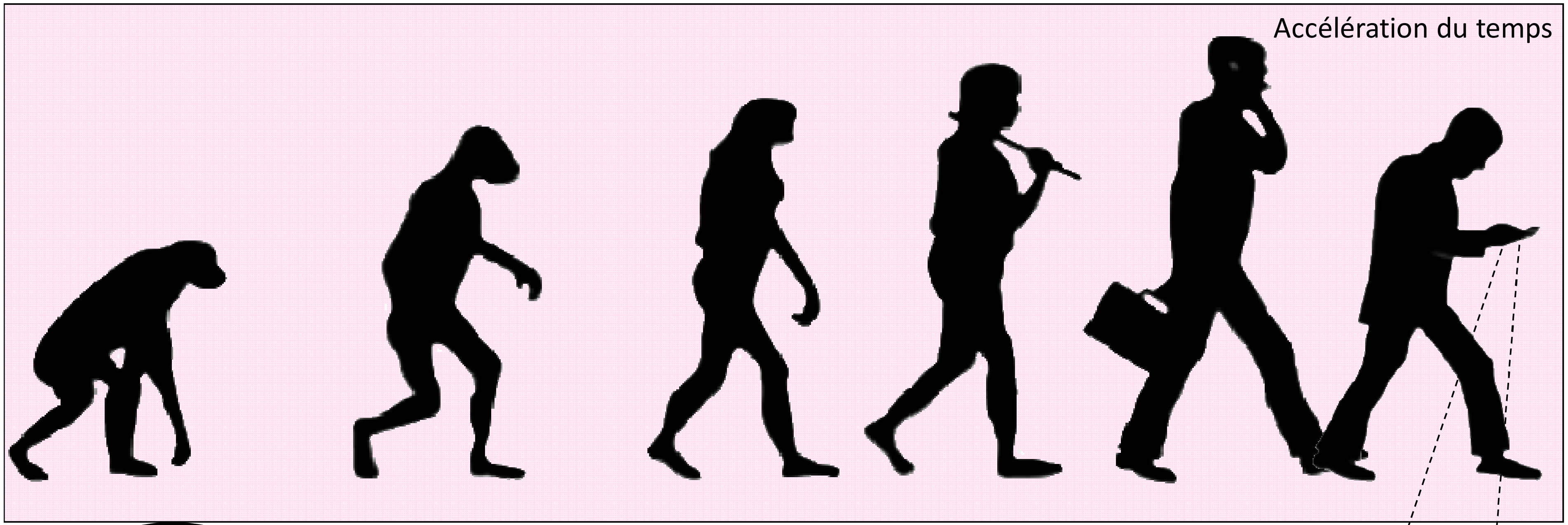


business objects runtime

One silo, one team, one budget

Le Z est mobile

R-evolution mobile…Darwinienne ?



Géolocalisé
Photographe
Traçable

Externe à l'entreprise
ou Interne

Connecté

Partageur

Insouciant

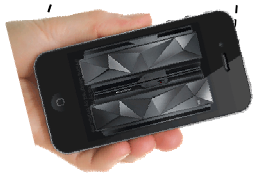
Exigeant
Impatient

Interruptible

'Technologique'

Distribution en **AppStore**

Appendice corporel
…intelligent…



Toujours **context**ualisé

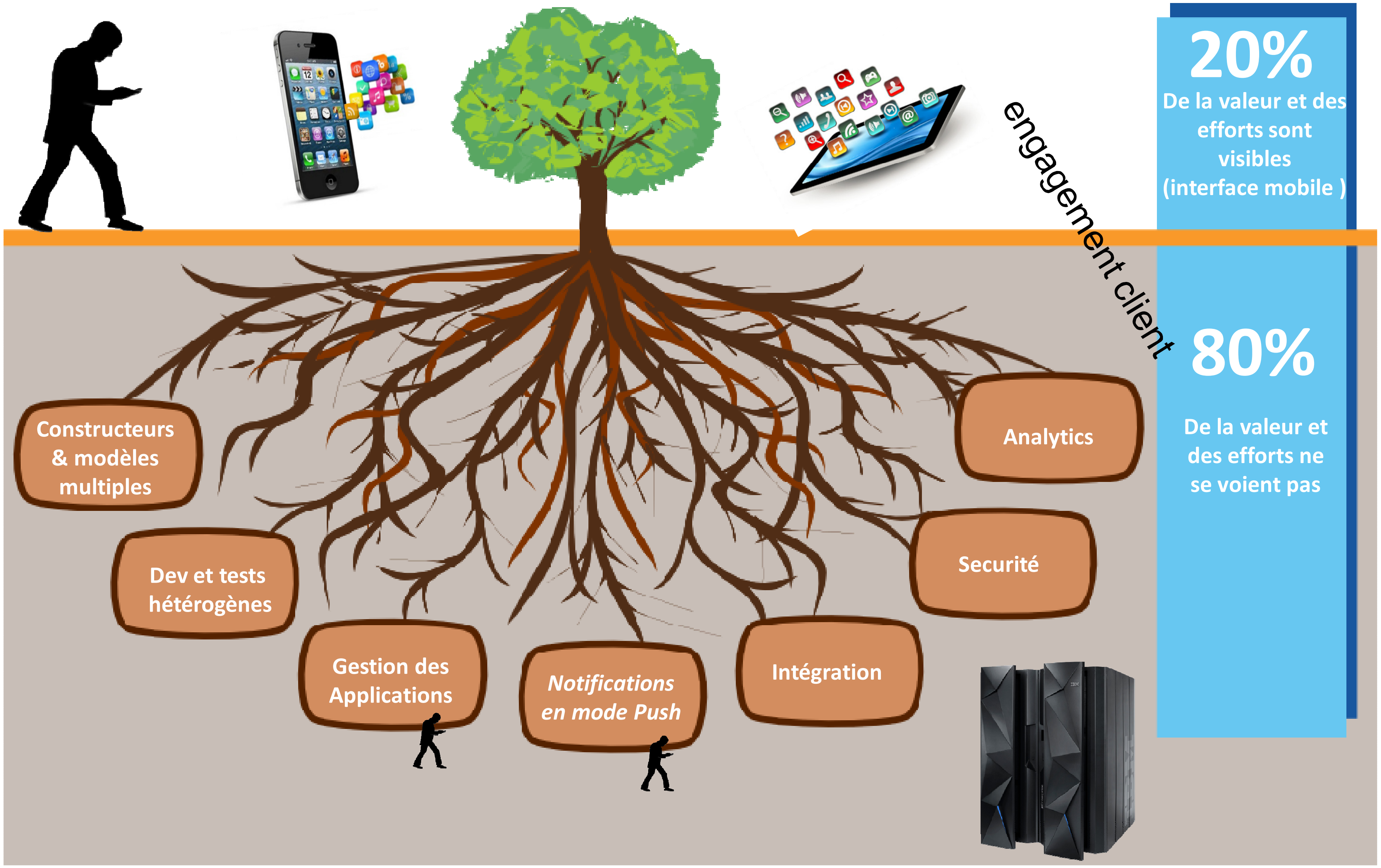
Nombreux
Imprévisibles

Permanent

problème à résoudre
opportunité

Une loi toujours vérifiée…

Une application mobile……ce n'est pas juste une interface

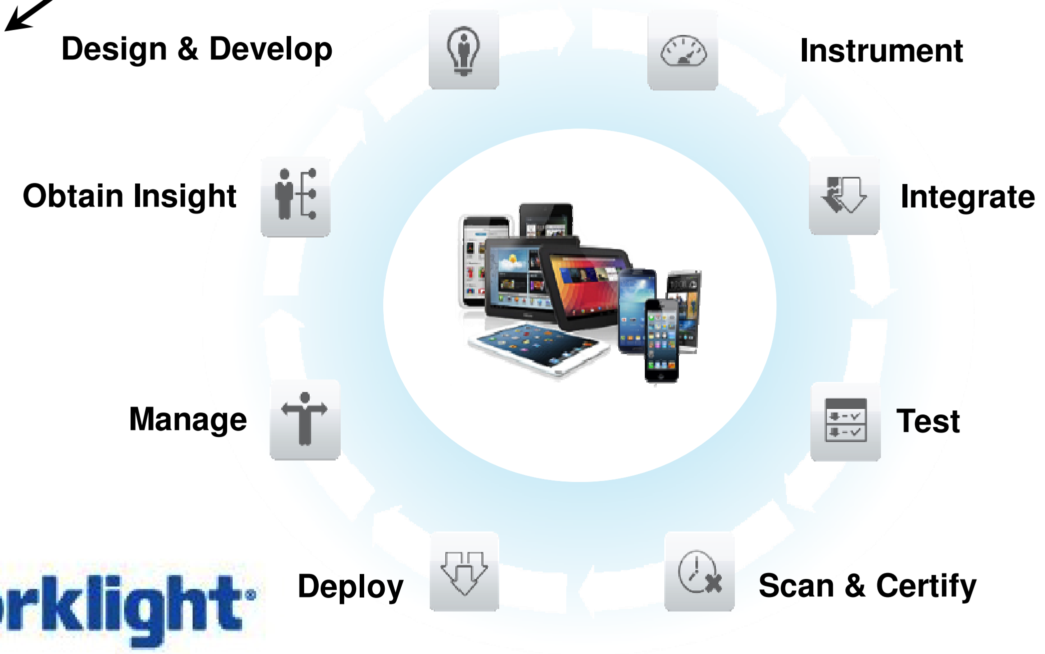

engagement client
20%
De la valeur et des efforts sont visibles (interface mobile)
80%
De la valeur et des efforts ne se voient pas
Constructeurs & modèles multiples
Dev et tests hétérogènes
Gestion des Applications
Notifications en mode Push
Intégration
Securité
Analytics

Mobile Enterprise Application Platform

- Du C/S Mobile
- Client 'lourd' et propriétaire
- Distribution en AppStore
- Standards 'légers'
- Réseau instable…et insécure
- Client 'mobile' en permanence
- Client 'pushable'

- Exigeant, zAAPeur
- Impatient, speed
- Partageur, insouciant
- Connecté, Nombreux
- Sensible à l'offre
- 'Pushable'

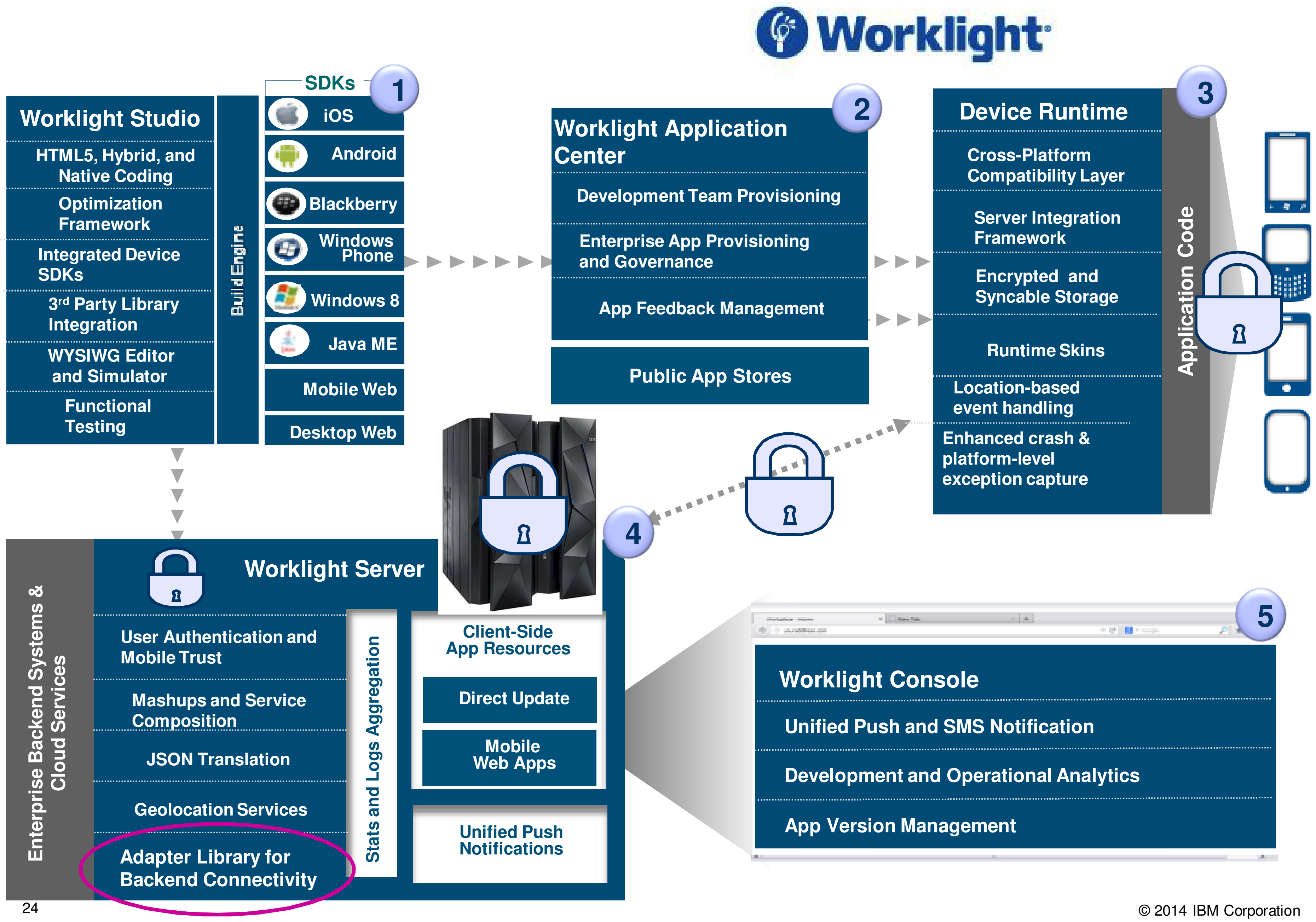
MEAP



- 'Write mobileApp once, run everywhere'
- AppStore as required
- **Integrate and secure on system Z**
  – secure everything, everywhere
- **Manage**, analytics, as required

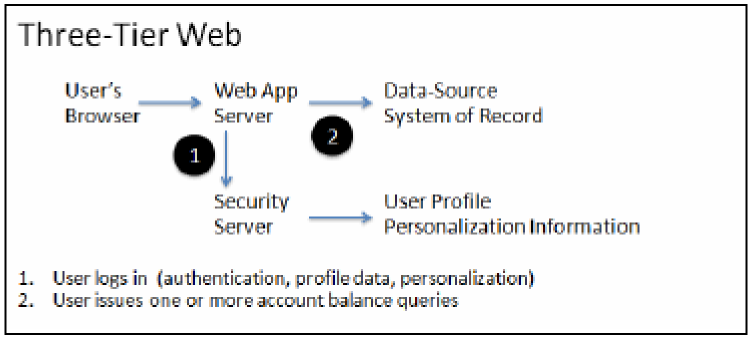
Oh ma MEAP blue Worklight

Oh, ma MEAP blue :


Worklight
1
SDKs
iOS
Android
Blackberry
Windows Phone
Windows 8
Java ME
Mobile Web
Desktop Web
Worklight Studio
HTML5, Hybrid, and Native Coding
Optimization Framework
Integrated Device SDKs
3rd Party Library Integration
WYSIWG Editor and Simulator
Functional Testing
Build Engine
2
Worklight Application Center
Development Team Provisioning
Enterprise App Provisioning and Governance
App Feedback Management
Public App Stores
3
Device Runtime
Cross-Platform Compatibility Layer
Server Integration Framework
Encrypted and Syncable Storage
Runtime Skins
Location-based event handling
Enhanced crash & platform-level exception capture
Application Code
4
Enterprise Backend Systems & Cloud Services
Worklight Server
User Authentication and Mobile Trust
Mashups and Service Composition
JSON Translation
Geolocation Services
Adapter Library for Backend Connectivity
Stats and Logs Aggregation
Client-Side App Resources
Direct Update
Mobile Web Apps
Unified Push Notifications
5
Worklight Console
Unified Push and SMS Notification
Development and Operational Analytics
App Version Management

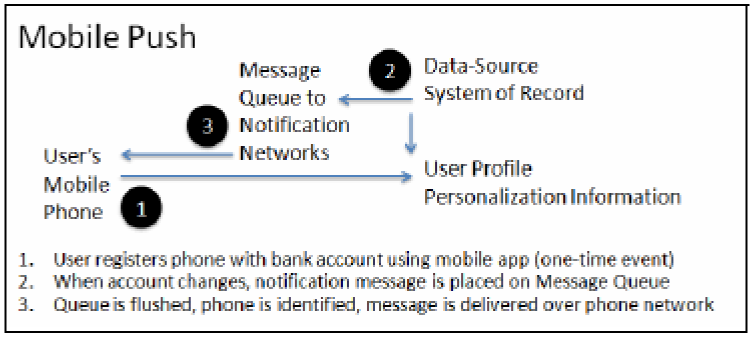
Petit aparté architectural: Push, Don't pull

- A push model may be more effective for low value transactions like balance inquiries
- Traditional three-tier web '**pull**' model


Three-Tier Web
User's Browser
Web App Server
Data-Source System of Record
Security Server
User Profile Personalization Information
1
2

1. User logs in (authentication, profile data, personalization)
2. User issues one or more account balance queries

- '**Push**' model


Mobile Push
Message Queue to Notification Networks
Data-Source System of Record
User's Mobile Phone
User Profile Personalization Information
1
2
3

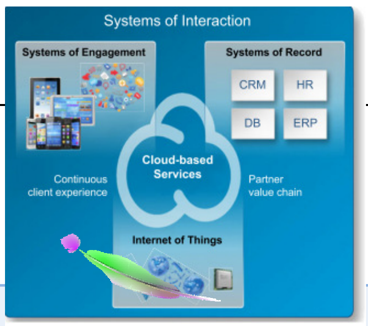
1. User registers phone with bank account using mobile app (one-time event)
2. When account changes, notification message is placed on Message Queue
3. Queue is flushed, phone is identified, message is delivered over phone network

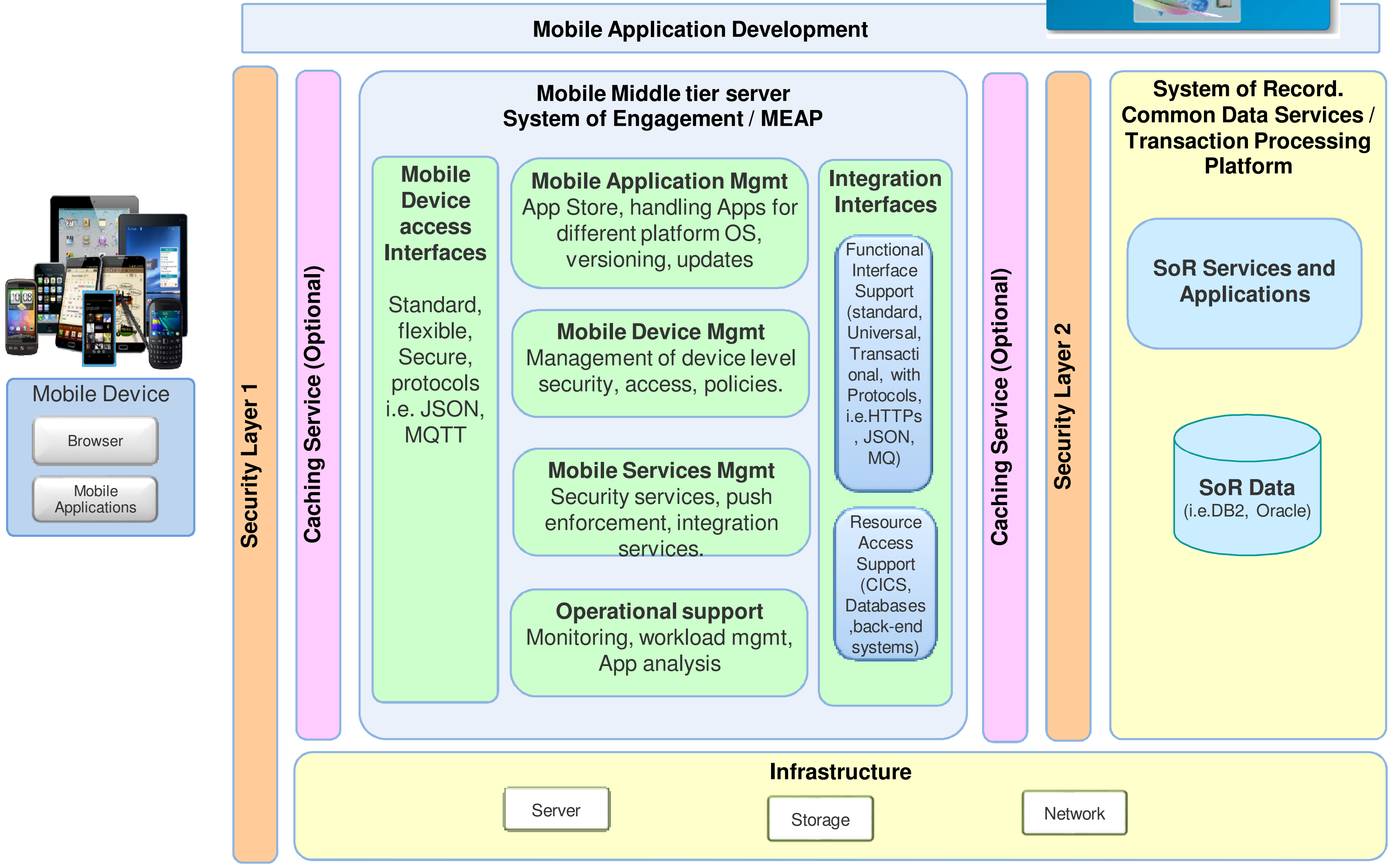
- Push model results in less transactions and transactions are spread out more evenly

See '*Mobile Design Patterns: Push, Don't Pull*', RED-5072
http://www.redbooks.ibm.com/abstracts/redp5072.html?Open

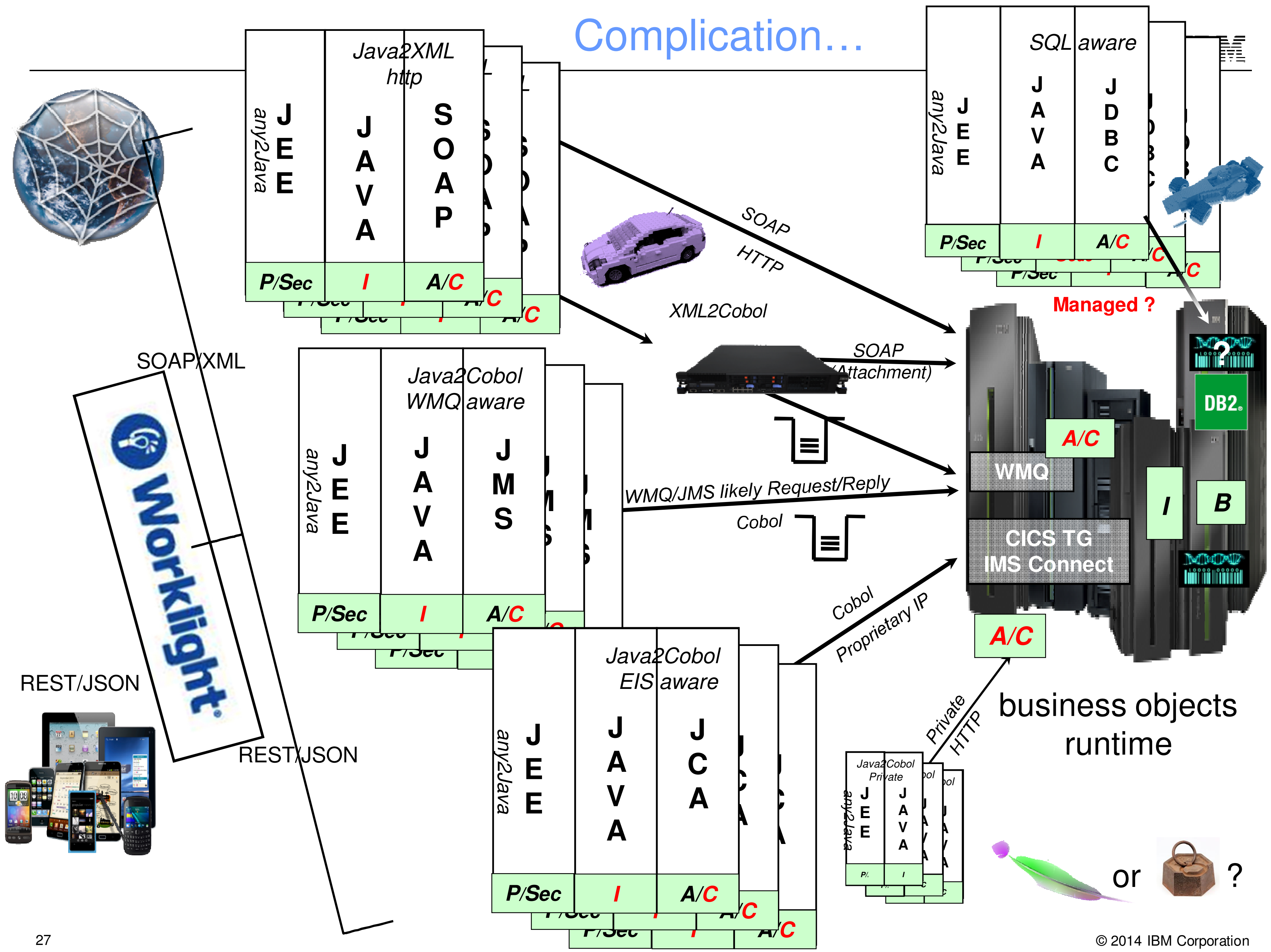
- Leverages system z middleware, i.e. *CICS events technology*
- *ALSO : API = HTTP = HTTP Caching*

A reference architecture

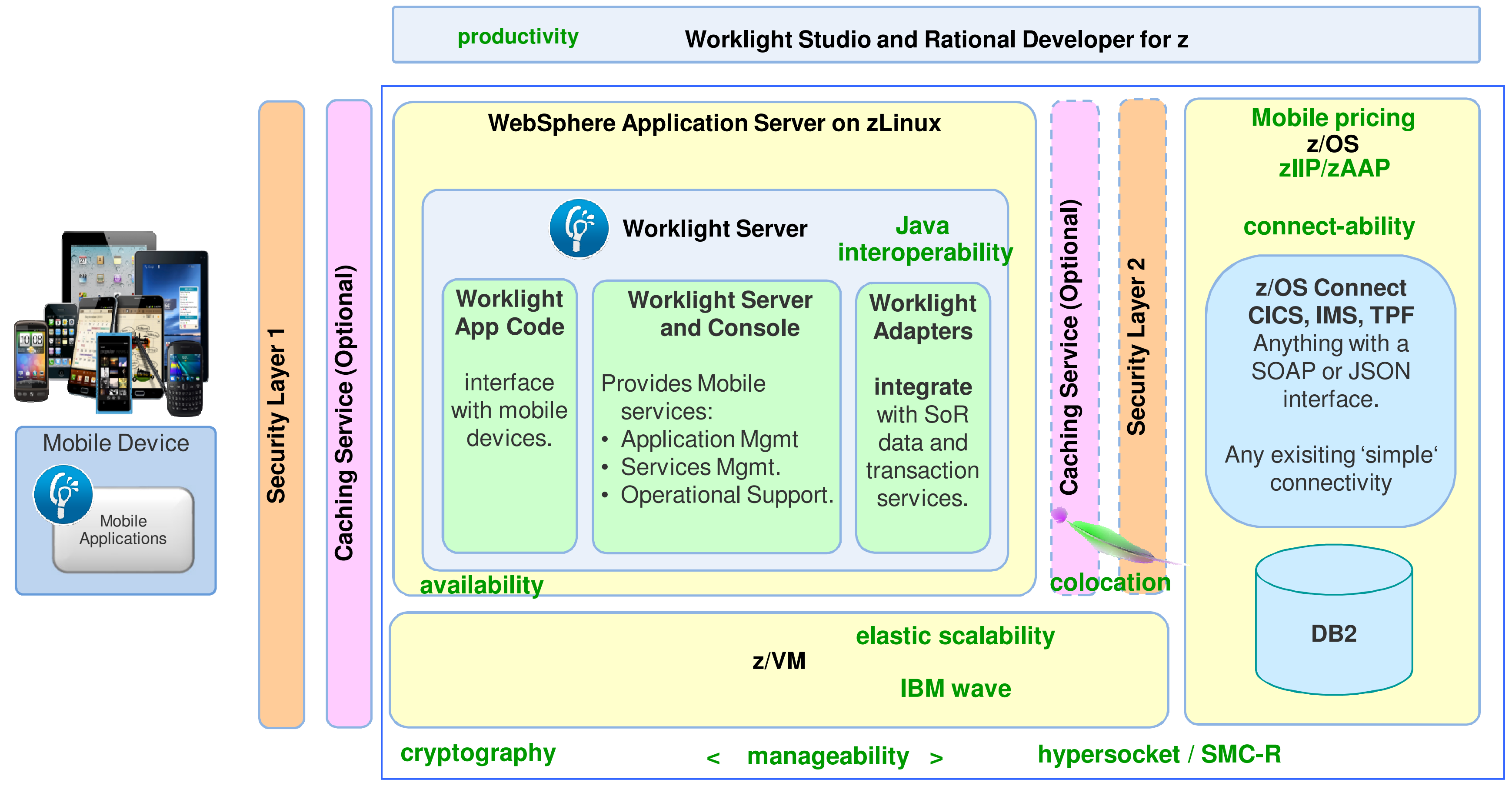



Mobile Application Development
Mobile Device
Browser
Mobile Applications
Security Layer 1
Caching Service (Optional)
Mobile Middle tier server
System of Engagement / MEAP
Mobile Device access Interfaces
Standard, flexible, Secure, protocols i.e. JSON, MQTT
Mobile Application Mgmt
App Store, handling Apps for different platform OS, versioning, updates
Mobile Device Mgmt
Management of device level security, access, policies.
Mobile Services Mgmt
Security services, push enforcement, integration services.
Operational support
Monitoring, workload mgmt, App analysis
Integration Interfaces
Functional Interface Support (standard, Universal, Transactional, with Protocols, i.e.HTTPs, JSON, MQ)
Resource Access Support (CICS, Databases ,back-end systems)
Caching Service (Optional)
Security Layer 2
System of Record. Common Data Services / Transaction Processing Platform
SoR Services and Applications
SoR Data
(i.e.DB2, Oracle)
Infrastructure
Server
Storage
Network

Complication…

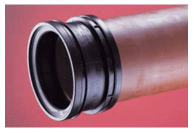


Reference architecture on System z



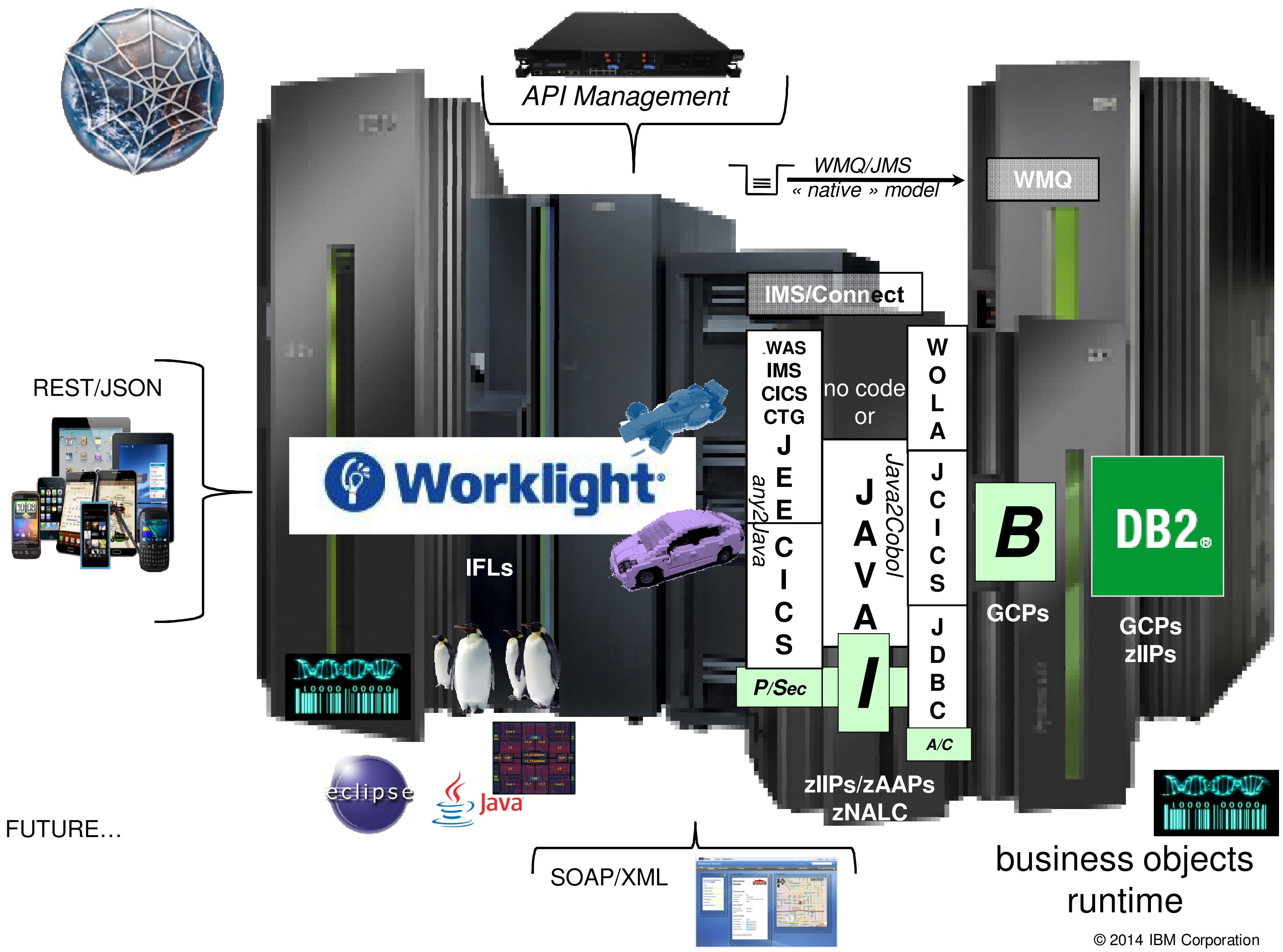
IBM Endpoint Manager
Mobile device management.

Simplicity helps Agility >>

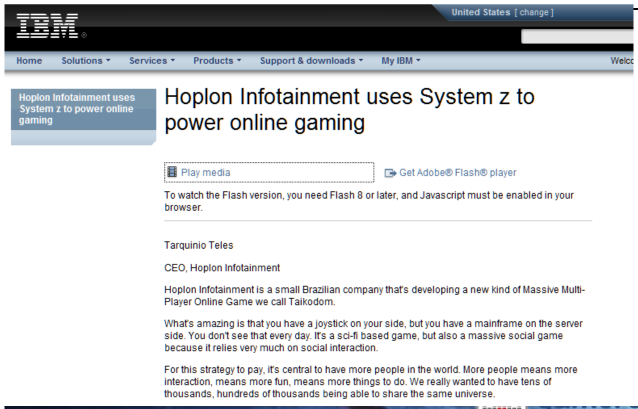


architecture simplicify

Make it simpler, at low cost



System z security



Hoplon Infotainment uses System z to power online gaming

Play media | Get Adobe® Flash® player

To watch the Flash version, you need Flash 8 or later, and Javascript must be enabled in your browser.

Tarquinio Teles

CEO, Hoplon Infotainment

Hoplon Infotainment is a small Brazilian company that's developing a new kind of Massive Multi-Player Online Game we call Taikodom.

What's amazing is that you have a joystick on your side, but you have a mainframe on the server side. You don't see that every day. It's a sci-fi based game, but also a massive social game because it relies very much on social interaction.

For this strategy to pay, it's central to have more people in the world. More people means more interaction, means more fun, means more things to do. We really wanted to have tens of thousands, hundreds of thousands being able to share the same universe.

- zLinux for hacking protection
- Java for the 'real world'
- Scalability as number of gamers increases
  – 20mn to provision a new server
- 24/24 7/7 reliability
- Mainframe shared with IBM Brazil retirement funds system