



IBM SOA ARCHITECT SUMMIT
LE 22 MAI 2008



Philippe BOURNHONESQUE

Directeur stratégie IBM Software France & NWA

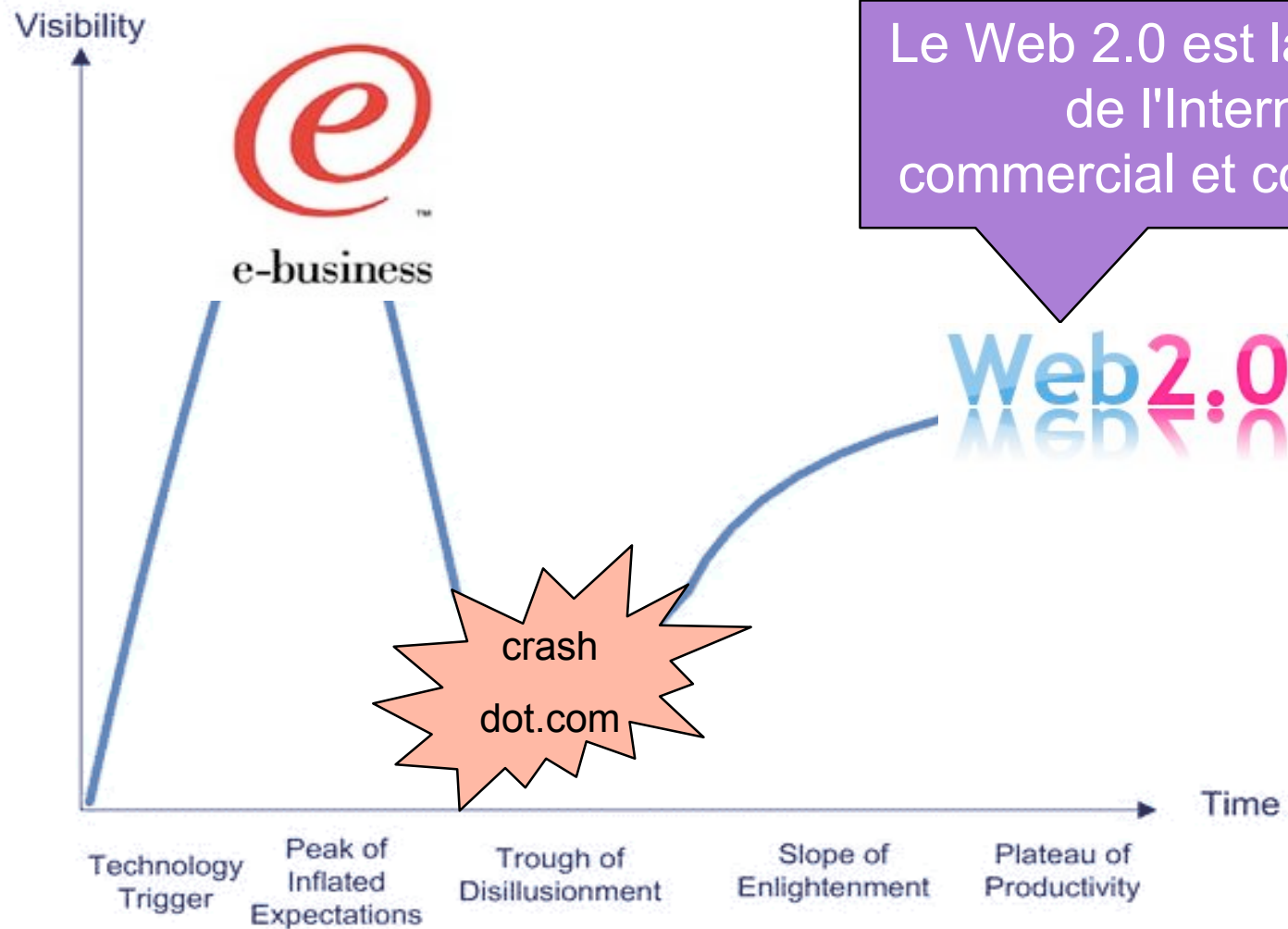


 **IBM SOA ARCHITECT SUMMIT**
LE 22 MAI 2008

The logo for the IBM SOA Architect Summit is a blue circle containing a white arrow that points to the right. The arrow is stylized with a thick shaft and a curved tip.

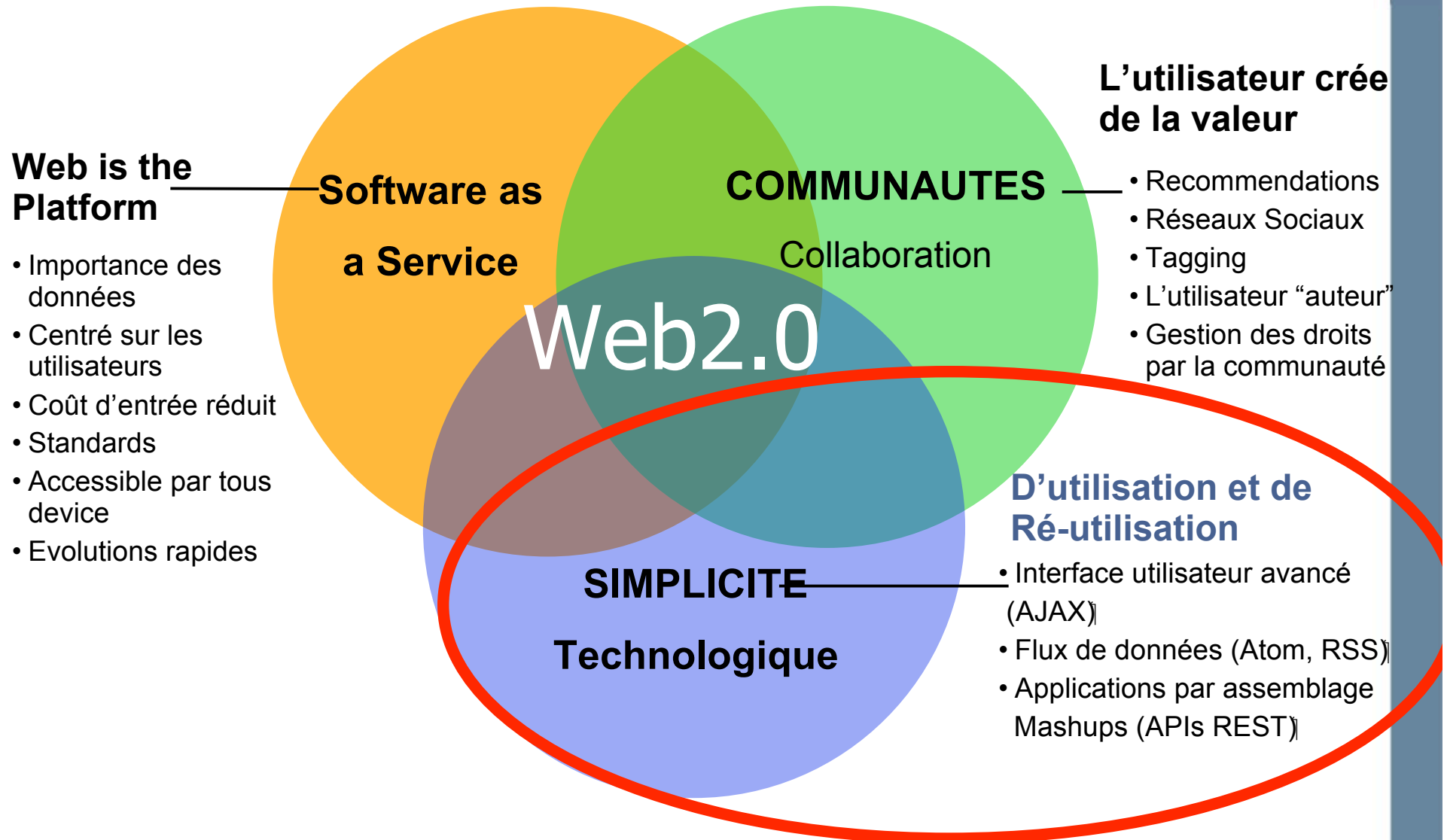
Nouvelles solutions SOA et Web 2.0

La révolution industrielle du Web – Phase 2



Le Web 2.0 est la maturité de l'Internet commercial et collaboratif

Web 2.0 : les concepts principaux



La technologie au service de la simplicité

- Langages de script
- REST (Resource Oriented Architecture)
- Feeds (Atom, RSS)
- Ajax
- Widgets
- Mashup
- Microformats (Tags sémantiques)

REST

Appliquer les principes d'Internet à l'accès aux services

- **Protocole HTTP**

- **Interface uniforme**

- 4 « méthodes » (get/put /post /delete) plutôt que API métier

- **Adressabilité : Tout est URL**

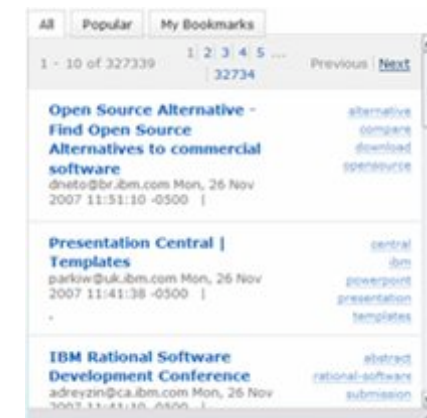
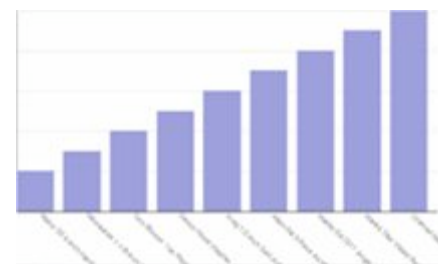
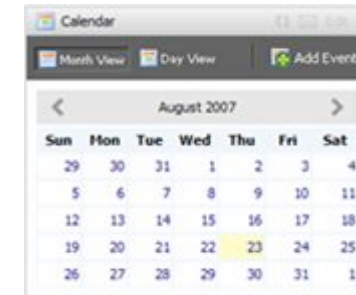
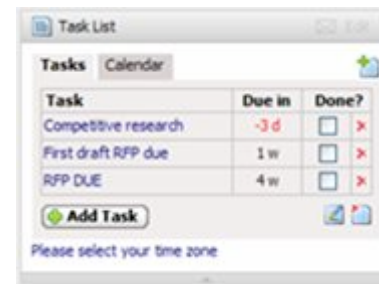
- <http://macompagnie.com/clients/>
- `http://macompagnie.com/clients/abc/affaires/123`
- <http://macompagnie.com/reservations/monvoyage>
- `http://macompagnie.com/commandes/232`

- **Sans Etat**

- Le logiciel client gère l'état
- Le serveur fournit une proposition d'état (liste d'URLs)

WIDGET - WIDGET, vous avez dit WIDGET ?

- Un **widget** est une application ou un élément de contenu dynamique qui s'insère facilement dans une page Web.
- Les Widgets existent sous des noms / technologies diverses : gadgets, blocks, flakes...
- Les Widgets peuvent être développés dans n'importe quel langage (Java™, .NET, PHP, etc.) et peuvent même être de simples fragments HTML.
- Les Widgets peuvent être non visuels.
- Les Widgets encapsulent fréquemment un Appel vers un service
- “Les widgets “Mashables” gèrent des “events”, et peuvent ainsi être reliés les uns aux autres pour créer de véritables applications.



Mashups – Pourquoi faire ?

Domaine	Self-service Agrégation d'information	Applications personnalisées centrées client	Développement applicatif 'Quick & Dirty'	Interface Web20 au Legacy	Syndication rapide de contenu
Objectifs	<p>Création d'applications situationnelles par les lignes métier :</p> <ul style="list-style-type: none"> - Rapidité d'analyse - Accélération décisions - Collaboration plus efficace - Visibilité accrue sur les informations métier 	<ul style="list-style-type: none"> - Amélioration de la satisfaction et de la loyauté client - Rajeunir la base clientèle par des features 'fun' - Efficacité grâce à un environnement personnalisé 	<ul style="list-style-type: none"> - Applications 'Good enough' - Développement RAD - Gouvernance - Iterations rapides 	<ul style="list-style-type: none"> - Exploitation des données de l'entreprise - Création de 'feeds' mashables - Accès direct évitant la duplication 	<ul style="list-style-type: none"> - Packager les données sous forme de feeds+Widgets - Réduction des coûts d'intégration - Support de nouveaux modèles
Exemples	<ul style="list-style-type: none"> - Analyse des risques - Systèmes d'alertes - Etude de marché - Analyse compétition - Customer intelligence - Reporting 	<ul style="list-style-type: none"> - Expérience personnalisée de banque en ligne - Application personnalisée d'agence immobilière - Site de voyage personnalisé 	<ul style="list-style-type: none"> - Prototypes - Démonstrations - Applications spécifiques pour équipes réduites (développées typiquement par des profils métier) 	<ul style="list-style-type: none"> - Exposing LOB siloed systems, including spreadsheets and access databases, as consumable feeds 	<ul style="list-style-type: none"> - Fournisseur de contenus riches personnalisés : services: météo, financiers, d'entreprise....
Solutions alternatives	<ul style="list-style-type: none"> - Tableurs - Assemblage manuel 	<ul style="list-style-type: none"> - Développement spécifique web 2.0. - Portails (peut être utilisé conjointement avec des outils d'assemblage de mashups) 	<p>Applications développées from scratch (pas réellement agiles...)</p> <ul style="list-style-type: none"> - VB, .Net - HTML, Photoshop 	<ul style="list-style-type: none"> - Développement spécifique - Upgrades ou remplacements disruptifs 	<ul style="list-style-type: none"> - Gadgets Google - Approche manuelle

Mashup – Exemple mashup « Compétition »

The screenshot shows a web application titled "Competitive" with several widgets and callouts:

- My Note:** A text area containing "Nicole Carrier's competitive competitors."
- Competitor list:** A table with columns for competitor names and a "Relier graphiquement le nom d'un compétiteur avec d'autres Widgets de la page" callout pointing to it.
- Experts on mashup competition:** A list of names including Boezman, John; Carrier, Nicole; Li, Tong; Orm, Mike; and a callout "Cliquez pour obtenir Le profil d'un expert (Connections)" pointing to the list.
- Index Search:** A search bar and a list of documents.
- Document Quickr:** A callout pointing to the document list.
- Universal Blog Search:** A search bar with "popfly" entered and a "Recherche dans Blogs" callout.
- Twitter Updates:** A widget showing recent tweets.
- google news:** A callout pointing to a news widget.
- Bookmarks (Digg, Dogear, etc.):** A callout pointing to a Digg.com widget.
- YouTube Tag - Popfly:** A widget showing YouTube videos with a "Videos youtube" callout.



Challenges

- **Besoin d'un standard de Widgets largement repris par l'industrie**

- IBM a créé un modèle de widgets légers bien aligné sur les modèles existants (tels que Google et NetVibes)

- L'objectif d'IBM est de définir un standard pour les widgets.

IBM dirige un groupe de travail de spécification de widgets au sein d'OpenAJAX.

Tirer parti de notre expérience de l'intégration des Gadgets Google ,
du modèle de widgets IBM, et des standards JSR 168/286 + WSRP
afin de créer la spécification widget OpenAJAX .

- **Sécurité**

- Mashup intégrant des données internes et externes (Non trusted API)

Possibilité de code malintentionné / malcodé (rogue widgets)

Besoin d'isolation entre Widgets

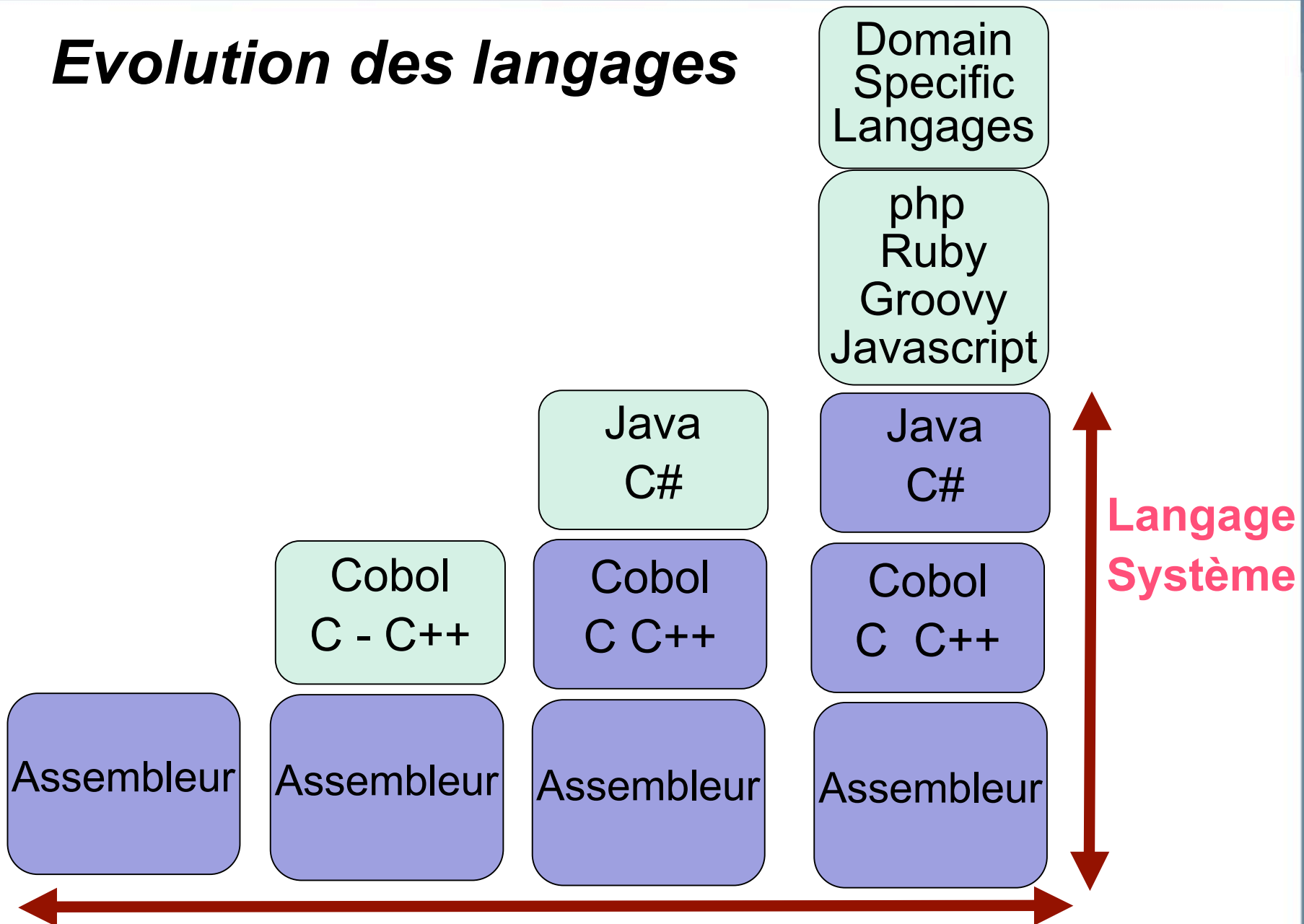
IBM contributeur à un standard de communication cross domaine (OpenAjax Alliance Hub 1.1)

- Lotus Mashups supportera les mashups sécurisés

Authentification, credentials, autorisation

Isolation des widgets par le hub Openajax

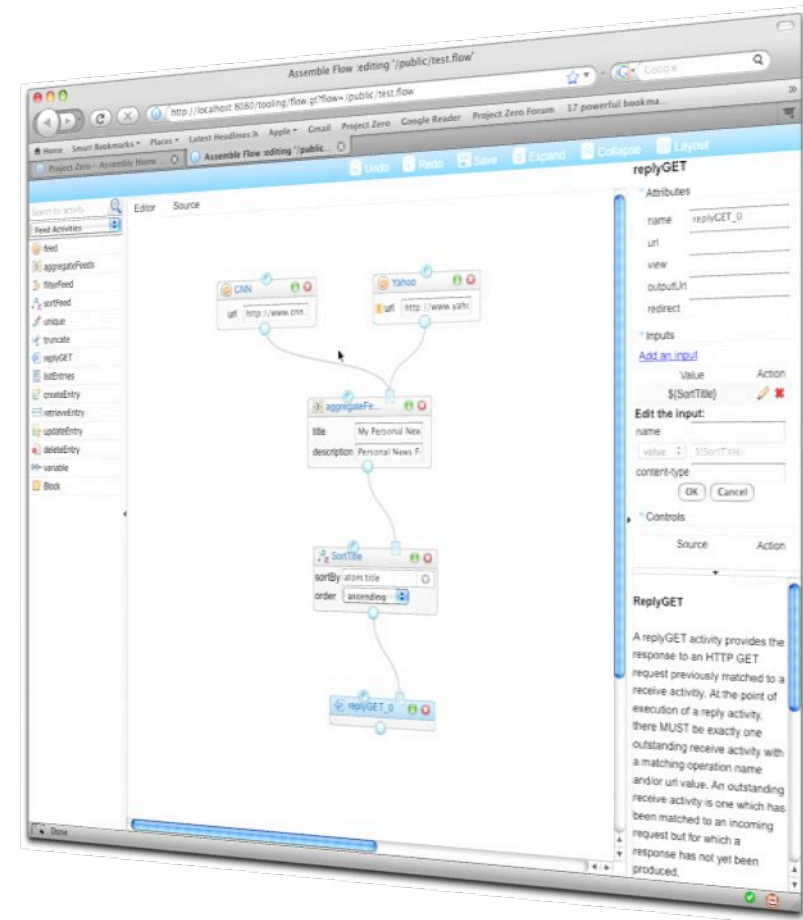
Evolution des langages



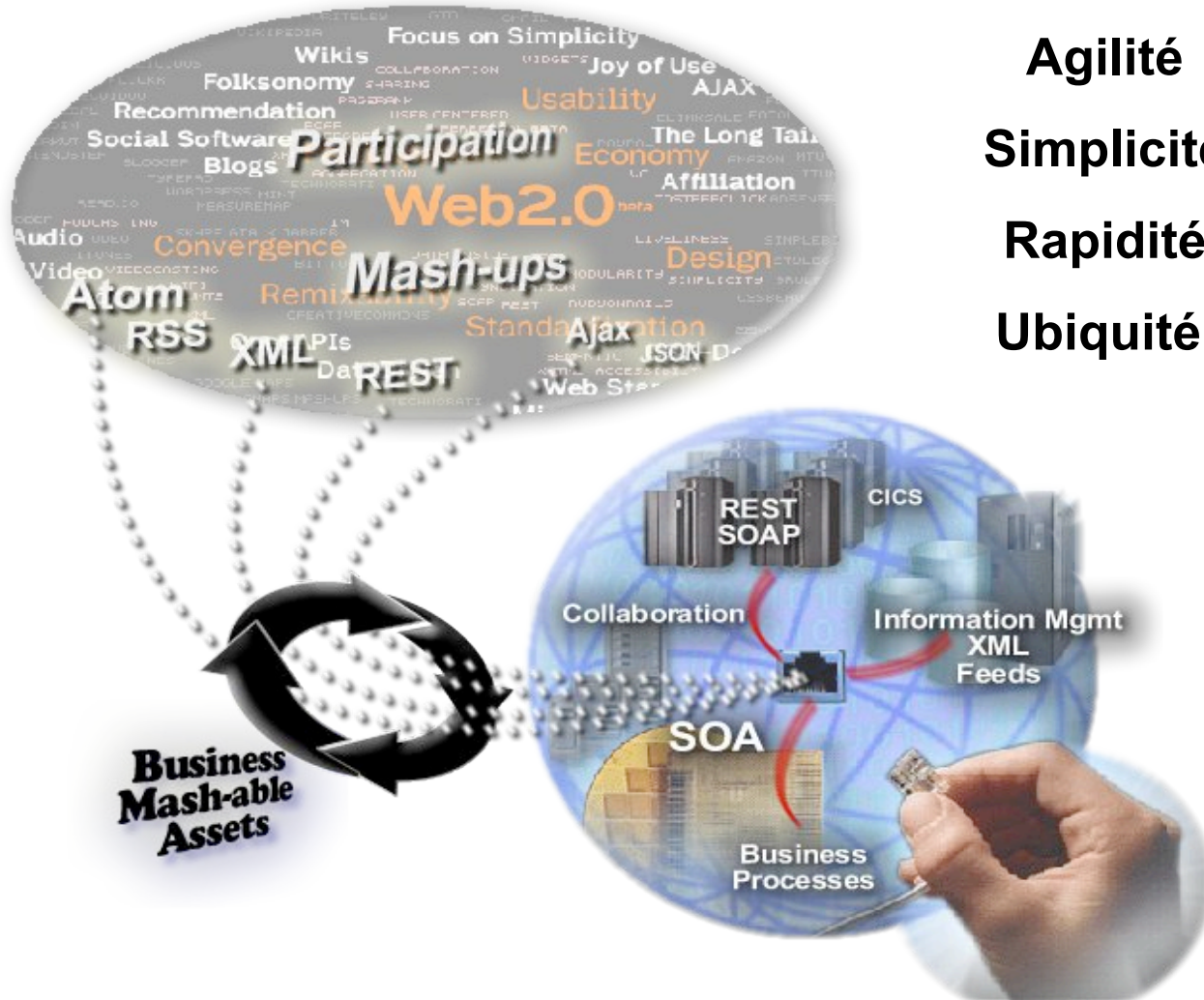
WEBSPHERE SMASH

La simplicité au service des développeurs

- Langages de script : Groovy et Php
- Modèle événementiel REST
- Framework ' Convention plutôt que Configuration'
- Composer des applications en “cablant” visuellement des services REST
- “Application is the server” : zip, copy & Run
- Développement open commercial : <http://www.projectzero.org>



Conclusion



Agilité
Simplicité
Rapidité
Ubiquité

Urbanisation
Sécurité
Résilience
Scalabilité