



IBM Software Group

Enterprise Generation Language (EGL): la transition douce vers Java et le Web pour les développeurs Informix, DB2 et Cloudscape/Derby

Aomar Bariz
Technical Sales Informix & DB2 products
Aomar.Bariz@fr.ibm.com

Rational software



ON DEMAND BUSINESS™

Agenda



- La problématique liée au 4GL
- Les solutions IBM Rational pour concevoir des applications J2EE
- EGL : LE FUTUR D'INFORMIX 4GL
- Cloudscape/Derby et EGL
- Démonstration
- Bénéfices
- Questions / Réponses



Agenda



- ➔ ■ La problématique liée au 4GL
 - Les solutions IBM Rational pour concevoir des applications J2EE
 - EGL : LE FUTUR D'INFORMIX 4GL
 - Cloudscape/Derby et EGL
 - Démonstration
 - Bénéfices
 - Questions / Réponses



4GL : Problématique

- Langage simple, facile, robuste
- Langage non orienté objet
- Langage spécifique Unix dans sa version IBM Informix
- Langage vieillissant, aucune prise en compte des dernières technologies
- Langage pour l'exploitation uniquement des SGBDR IBM Informix
- Applications critiques développées en 4gl
- Applications batch ou TUI
- Faible possibilité de faire évoluer l'existant
- Peu ou pas de possibilité de déploiement

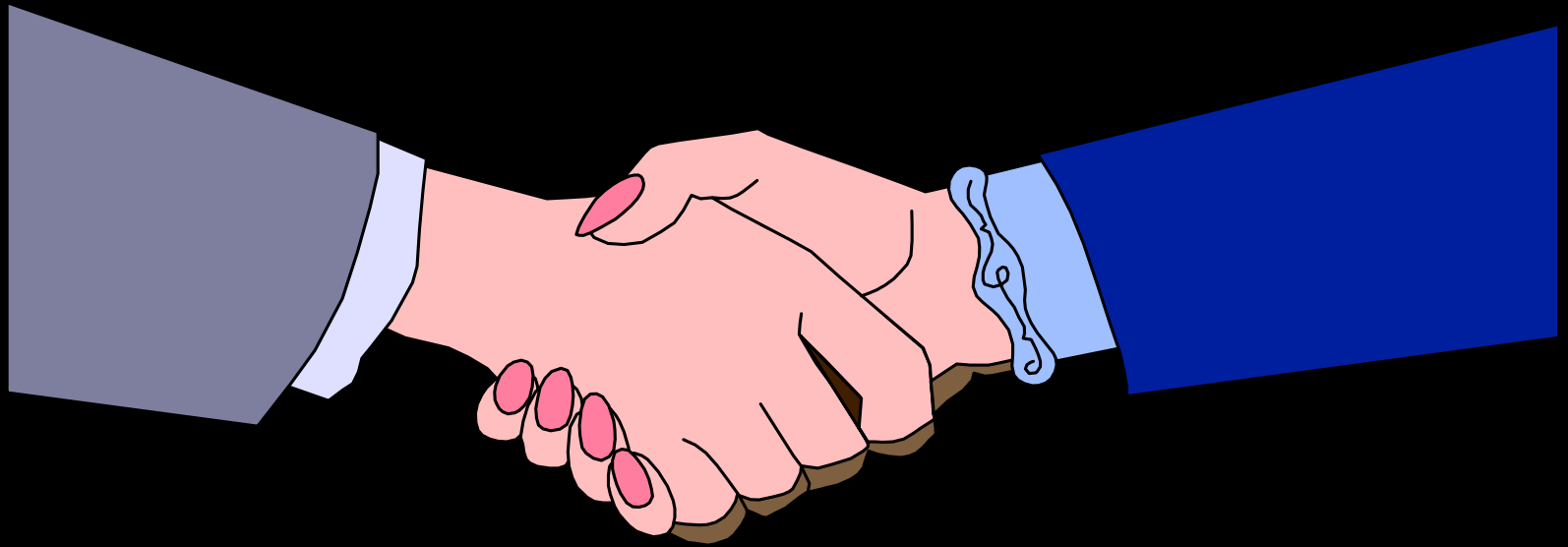


Plate-formes supportées

Vendor	Platform(s)	OS Version(s)
Compaq	Tru64 (64-bit)	5.1, 5.1A, 5.1B
HP	HP-UX (32-bit and 64-bit)	11, 11i, 11iv2*
Sun	Solaris (32-bit and 64-bit)	7, 8, 9
IBM	AIX (32-bit and 64-bit)	4.3.3, 5.1, 5.2
IBM	Linux (64-bit)	zSeries
Intel	SuSE Lin. Enterprise (64-bit)	8*
Intel	UnitedLinux (64-bit)	1*
Intel	RedHat Linux (32-bit and 64-bit)	9*
SCO	Caldera SCO (32-bit)	3.2.5.05
Intel	Linux (32-bit)	Kernel 2.4.2-5.0GLIBC 2.2x*



Intégration Informix 4GL / IBM Rational



Agenda



- La problématique liée au 4GL
- ➔ ■ Les solutions IBM Rational pour concevoir des applications J2EE
- EGL : LE FUTUR D'INFORMIX 4GL
- Cloudscape/Derby et EGL
- Démonstration
- Bénéfices
- Questions / Réponses



Construire une architecture flexible

Capacités



Analyste

Architecte

Développeur

Testeur

**Responsable du
déploiement**

Exigences et Analyse

Conception et Construction

Qualité logicielle

Gestion du déploiement

**Customer
Extensions**

ECLIPSE

**3rd Party
ISV Tools**



**Project
Manager**

Gestion de configuration logicielle



Executive

Gestion de projets et de processus



IBM Software Development Platform



Analyste

**WebSphere
Business
Integration
Modeler
& Monitor**

**Rational
Software
Modeler**



Architecte

**Rational
Software
Architect**



Développeur

**Rational
Application
Developer**

**Rational
Web
Developer**



Testeur

**Rational
Functional
& Manual
Tester**

**Rational
Performance
Tester**



**Responsable du
déploiement**

**Tivoli
Configuration
Manager**

**Tivoli
Monitoring**

*Customer
Extensions*

ECLIPSE

*3rd Party
ISV Tools*



Chef de projet

Rational Team Unifying Platform



Dirigeant

Rational Portfolio Manager



Produits d'analyse et de conception

- UML 2.0
- *Pattern/ Transform Authoring*
- *Reusable Asset Browser*

- Transformation modèles UML
- Contrôle et audit d'architecture
- Outils de développement C/C++

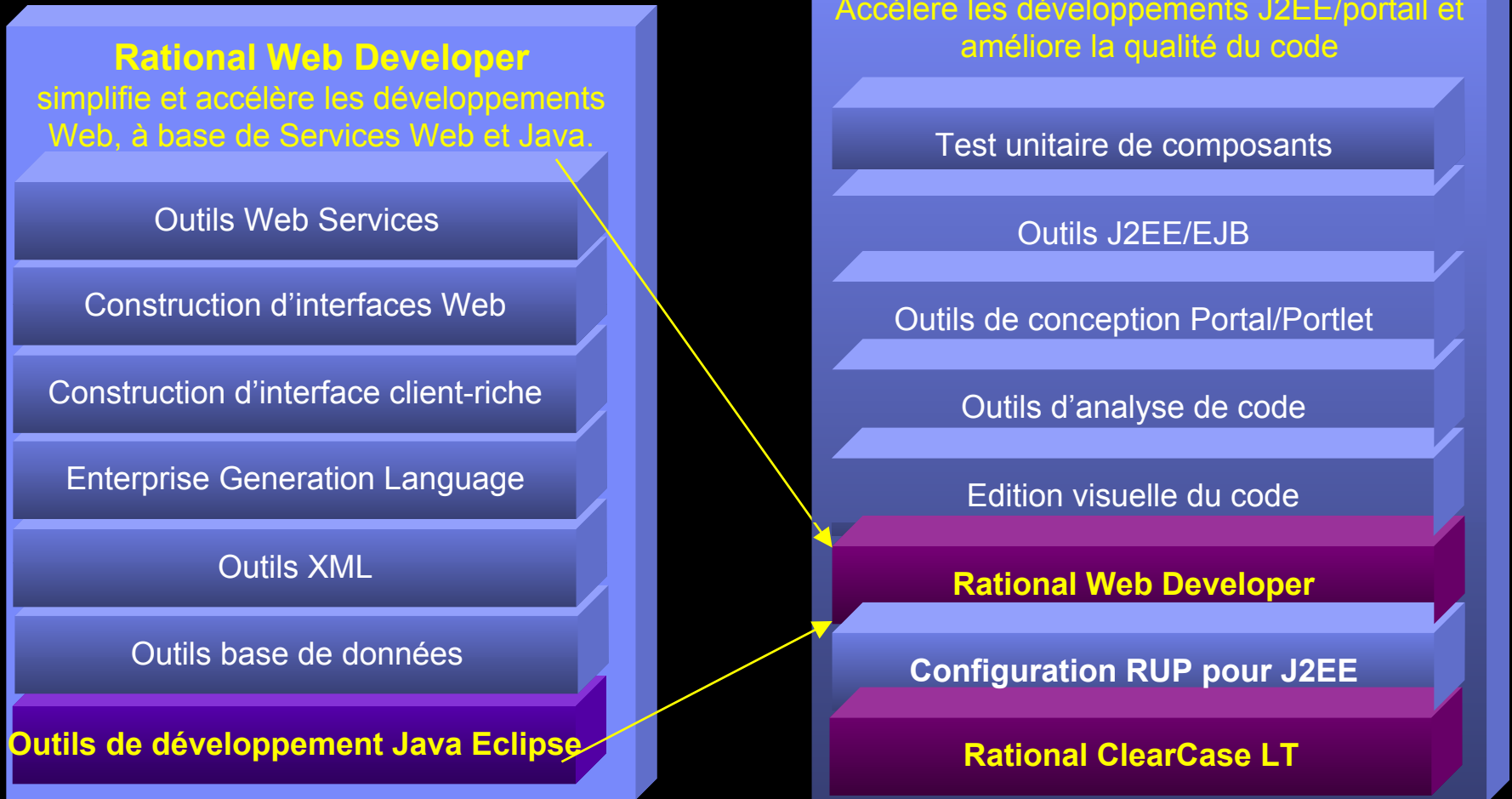


- Développement Web
- Développement Web Services
- Développement 'client riche'
- Outils XML et base de données
- Outils de génération Java
- Test unitaire

- Développement J2EE/EJB et portail
- Test de composant
- *Revue de code et analyse à l'exécution*
- Editeur visuel UML
- Gestion de configuration



Produits pour les développeurs



*Précédemment WebSphere Studio Site Developer

*Précédemment WebSphere Studio Application Developer



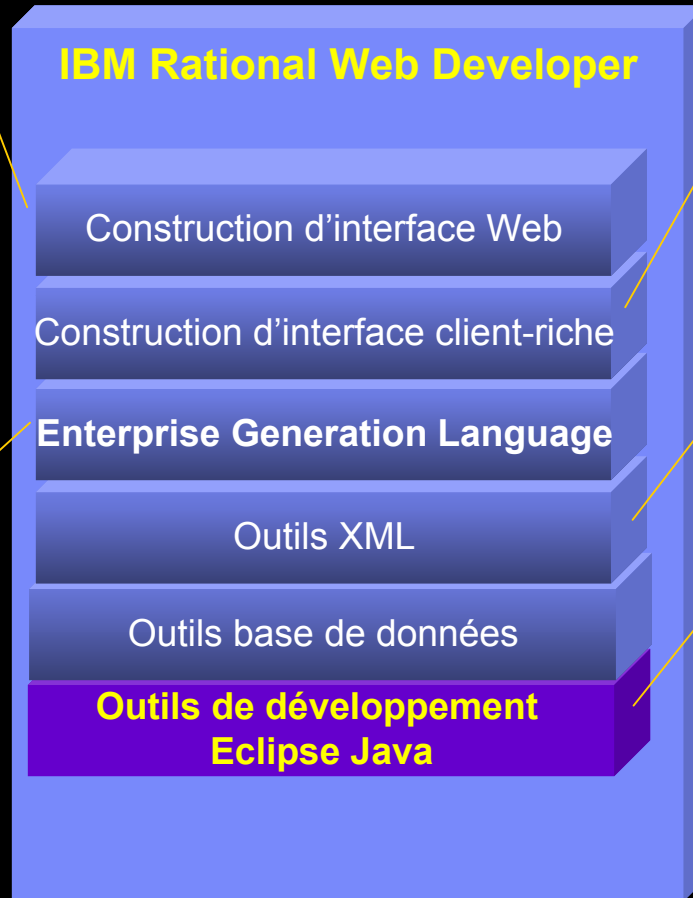
Contenu de IBM Rational Web Developer

“Construction d’interface Web”

- Conception visuelle du site
- Glisser/déposer des composants pour construction des clients Web supportant:
 - HTML, JSP, Servlet
 - Struts, JSF et SDO (connecteurs CRM)

“EGL”

- 4GL simplifié pour le développement d’application Web
- Text User Interface Programs Support (3270, Curses)
- VisualAge Generator -> Outil de migration EGL
- Java Runtime Targets



“Construction d’interface client riche”

- Editeur de code visuel Java pour la composition de client riche
- Support Swing, AWT, widgets SWT

“Outils XML”

- Support XML et XSD

“Outils de développement Eclipse Java”

- J2SE development tools
- Complétion du code, recherche, *refactoring*
- API extensible pour intégration CVS, ClearCase ...
- Plug-in de développement pour étendre l’environnement



Agenda



- La problématique liée au 4GL
- Les solutions IBM Rational pour concevoir des applications J2EE
- ➔ ■ **EGL : LE FUTUR D'INFORMIX 4GL**
- Cloudscape/Derby et EGL
- Démonstration
- Bénéfices
- Questions / Réponses



EGL Overview : C'est quoi EGL ?

- **EGL c'est ?**

- ▶ **Langage de programmation 4GL + Environment Developpement**

Langage de programmation 4GL

- ▶ Robuste, facilité d'apprentissage, mature
 - Basé sur la combinaison d' **Informix 4GL + VAGen**
 - 25 années R&D + production
 - Centaines de milliers d'utilisateurs
 - Milliers d'applications en productions

+ Environment Developpement

- ▶ Integration dans Rational Web Developer
 - Benefice du meilleur de la technologie IDE
- ▶ Integration d'une multitude d'outil d'aide au développement



EGL Overview

- **A qui s'adresse EGL?**
 - ▶ **Client/Server developers:**
 - Informix 4GL
 - VAGen
 - Oracle Forms
 - PowerBuilder
 - Visual Basic
 - ▶ **Legacy developers:**
 - RPG
 - COBOL

- **Aucune compétence Java/J2EE ... Le langage n'est pas *Orienté-Objet*.**

- **Le but est de se focaliser sur l'application métier plutôt que de perdre du temps dans l'apprentissage du langage.**



EGL Overview : A quoi cela sert ?

- **Que peut-on faire avec EGL?**
 - ▶ **Applications base de donnée**
 - ▶ Applications standalone TUI (Text User Interface)
 - **Linux, Unix, Windows, iSeries, CICS (zOS)**
 - ▶ Applications internet/intranet au format web
 - ▶ Applications standalone batch
 - ▶ EGL permet de générer du code Java (RWD & RAD) et COBOL
 - RWSED nécessaire pour générer du COBOL generation
 - ▶ Web Services
 - ▶ Applications orientée message (WebSphere MQ Support builtin)
 - ▶ Programme Appellable depuis des clients GUI Java riche
 - ▶ **GUI client riche (TBD 2005)**



EGL : Phase de Développement

Modelisation

- Import du modèle de donnée depuis le schema dans la base de donnée ou depuis le modèle UML vers EGL
- En option mais très utile

Developpement

- Définitions de la "business logic"
- Création des pages web ou des TUI
- Pas de prise en compte de la future plateforme
- Test interactif de la logique applicative
- Travail d'équipe dans les projets



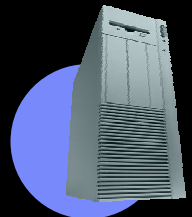
Generation

- Transformation EGL vers Java ou COBOL
- Creation de fichier JAR pour l'export et le déploiement
- Creation de script de build




Deployment et Execution

- Windows
- Linux, AIX (Solaris and HP in 6.0)
- iSeries
- zOS: CICS and Batch (IMS coming)



Phase de Development avec EGL

- 
- Définition du Modèle de donnée
 - Méthode d'accès au donnée
 - Création de la Business logique
 - Création des Interfaces Utilisateur
 - Debug Business Logique
 - Integration avec des applications existantes
 - Déploiement



EGL Model de donnée : Définitions Riches des données pour vos Business Applications

Utilisations du schéma des données et de la base existante:

Import du UML Class Model depuis Rational Rose/XDE

Import du model de donnée depuis le DB Schema

Définitions des Record avec SQL Retrieve

Dictionnaire Central de donnée:

Define a central data definition dictionary across multiple applications.

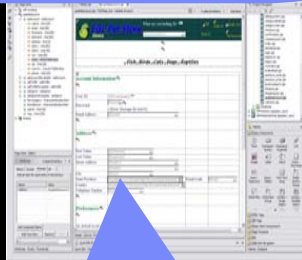
Definition du display, formatting, validation for data items

Definition de table dynamique pour le lookup, et la gestion des erreurs etc.

Réutilisation des définitions des Dataitem pour la création d'application Record

Access aux données depuis différentes sources :

Définitions des Record pour accéder aux données Relationnel depuis, MQ, CICS, IMS, XML* etc.*



```
// Customer SQL Record
Record Customer SQLRecord
{ tableNames=("DB2ADMIN.CUSTOMER"),
  keyItems=("customerId") }

  customerId CustomerId ←
  firstName FirstName;
  lastName LastName;
  ...
  state State; ←
end
```

```
Dataltem CustomerId
{ column=CUSTOMER_ID,
  range=(1, 1000),
  displayName="Customer Number",
  format="#####" }
end
```

```
Dataltem State char(2)
{ displayName="State Abbr.",
  format="AA",
  validatorTable=StateTable }
end
```

```
DataTable StateTable type matchValidTable
StateAbbreviation char(3);
{ contents = [{"NC"}, {"MN"}, {"TX"}, {"VA " ...}] }
end
```

```
// StoreLocation SQL Record
Record Store SQLRecord
...
state State; ←
end
```

Réutilisation des définitions de Record et des Data item à travers plusieurs pages:

Renforcement automatique de l'accès au donnée, display, formatting, et des règles de validation définies par le Record et dans les Dataitems

Focus sur les "business problems" plutôt que sur la mise au point



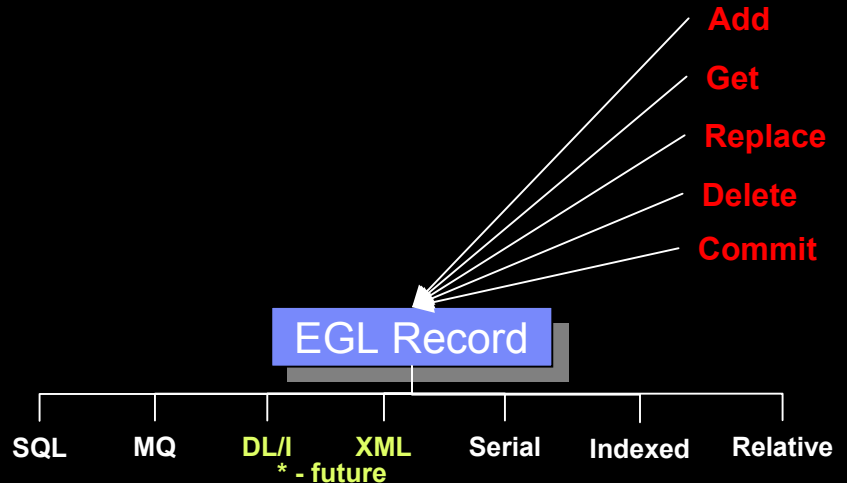
Phase de Development avec EGL

- Définition du Modèle de donnée
- ➔ ■ Méthode d'accès aux données
- Création de la Business logique
- Création des Interfaces Utilisateur
- Debug Business Logique
- Integration avec des applications existantes
- Déploiement



EGL Accès au Donnée : Modèle de programmation simplifié pour l'accès au donnée

- “Record” abstracts access to:
 - ▶ Donnée relationnelle
 - ▶ Messages MQ
 - ▶ Indexation, Relative, Serial data




- Syntaxe commune pour la manipulation de données
 - ▶ Add, Get, Replace, Delete, Commit
- Gestion des erreurs I/O unique en fonction des types de données
- Gestion automatique lors de la transformation du type de donnée

Focus sur les” business problemes” plutôt que sur la mise au point



Phase de Development avec EGL

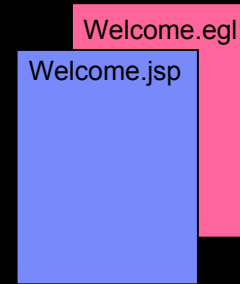
- Définition du Modèle de donnée
- Méthode d'accès aux données
-  ■ Création de la Business logique
- Création des Interface Utilisateur
- Debug Business Logique
- Integration avec des applications existantes
- Déploiement



EGL Développement de la business logique

- Page Handler

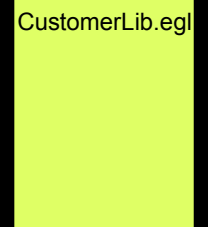
- ▶ Contient les fonctions et les données en relation avec le .jsp
- ▶ "On Page Load" fonction



- Librairie

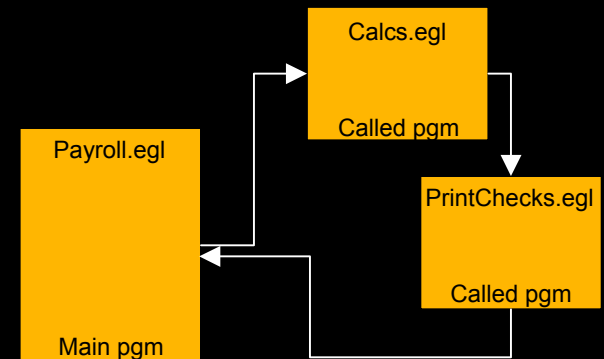
- ▶ Plusieurs point d'accès
- ▶ Définition de la "Business Logic" pour les apps web
- ▶ Container de fonctions et de déclaration de données
 - Example: "Data Access Library" pour la table Customer
 - Définition du SQL Record
 - Utilisation des CRUD fonctions
 - Générer par le "Data Access Library Wizard" depuis les schémas existants

Call "getCustomer" function



- Programme


- ▶ Point Unique
- ▶ Programme TUI, Batch, GUI



Focus sur les "business problems" plutôt que sur la mise au point

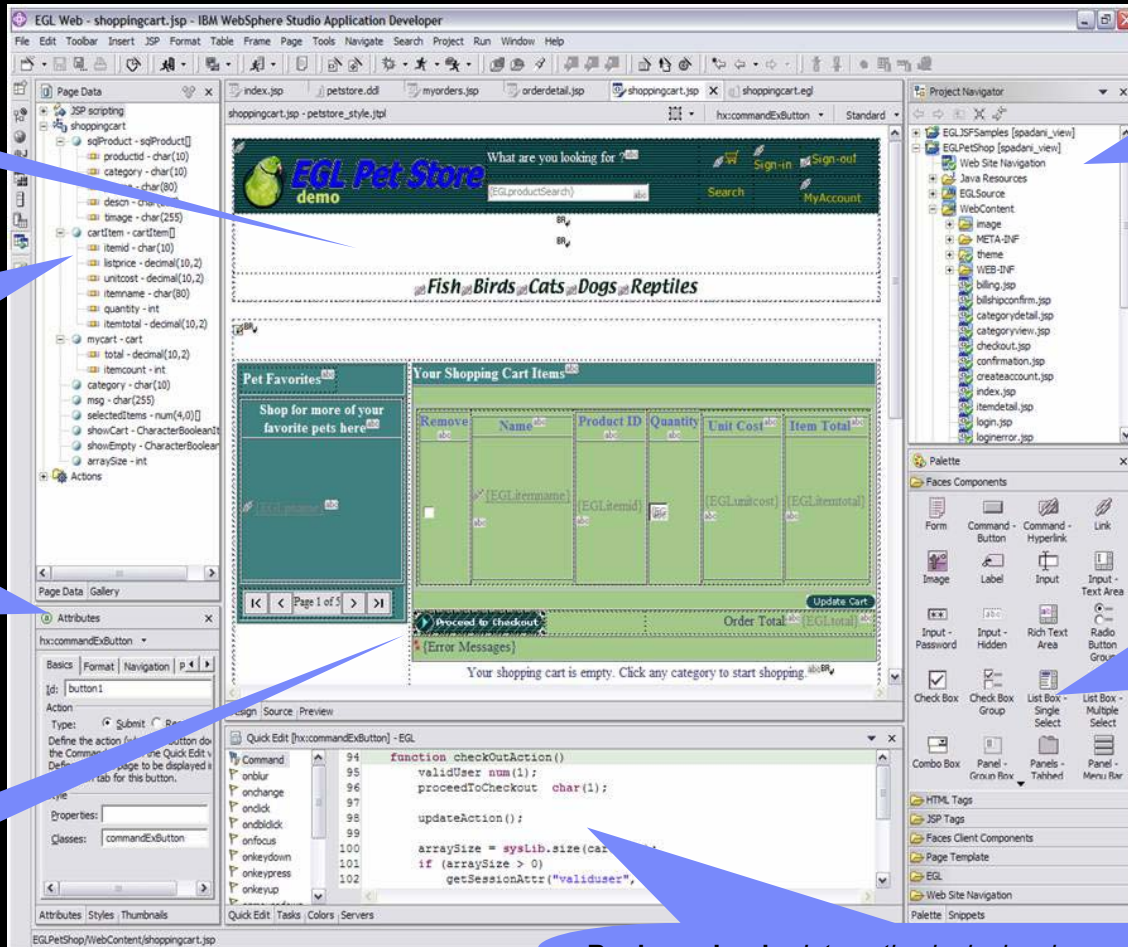


Phase de Développement avec EGL

- Définition du Model de donnée
- Méthode d'accès au donnée
- Création de la Business logique
-  ■ Création des Interfaces Utilisateur
- Debug Business Logique
- Integration avec des applications existantes
- Déploiement



EGL Design de vos interfaces et applications



Page Designer:
JavaServer Faces based
GUI Page Designer for Web

Page Data: Drag and Drop
EGL Data Model Records
and Data Items to build
dynamic web pages using
Page Designer

Control Attributes:
Customize visual, formatting,
validation, paging, navigation
properties for GUI controls

Command Event: Trigger
Server side EGL business
logic from visual controls

Project Navigator:
Application Artifacts
include Records,
Data Items, Page
Handlers, Libraries,
Pages, Styles,
Templates, etc...

Control Palette:
Faces
Components,
Faces Client
Components,
HTML Tags, JSP
Tags, EGL Data
Controls, etc...

Business Logic: Interactive logic development and debugging
in EGL (For developers experienced in COBOL, RPG, PL/SQL,
PowerBuilder, Informix, Visual Basic and other 4GL
programming languages.

Focus sur les "business problemes" plutôt que sur la mise au point



EGL – Applications Web JSF





Project Explorer

- Connector Projects
- EJB Projects
- Dynamic Web Projects
 - StoresWebDemo
 - EGLSource
 - (default package)
 - data
 - StoresLib.egl
 - pagehandlers
 - ShowAllCustom
 - StoresWebDemo.e
 - Web Site Navigation
 - Deployment Descriptor
 - Java Resources
 - WebContent
 - META-INF
 - theme
 - WEB-INF
 - ShowAllCustomers
- Other Projects
- Web Services
- Databases

*Navigation - StoresWebDemo *ShowAllCustomers.jsp

ShowAllCustomers.jsp - ShowAllCustomers.jsp *

tpl:put Standard

Informix
An IBM DB2 Database

ShowAllCustomers

All Customers from the Stores Demo database...

Customer_num ^{abc}	Fname ^{abc}	Lname ^{abc}	Company ^{abc}
{EGLcustomer_num} ^{abc}	{EGLfname} ^{abc}	{EGLlname} ^{abc}	{EGLcompany} ^{abc}

Page 1 of 5

Write all customers to file

HTML Tags

JSP Tags

Crystal Repor...

Faces Com...

Data Table

Panel - Group Box

Panels - Tabbed

Panel - Menu Bar

Command - Button

Command - Hyperlink

Link

Input

Input - Text Area

Rich Text Area

Input - Password

Check Box

Check Box Group

Radio Button Group

Combo Box

List Box - Single Select

List B...

Page Template

Web Site Navi...

EGL

Page Data

Scripting Variables

ShowAllCustomers

- customers - customer[]
 - address1 - char(20)
 - address2 - char(20)
 - city - char(15)
 - company - char(20)
 - customer_num - int
 - fname - char(15)
 - lname - char(15)
 - phone - char(18)

Design Source Preview

Properties Quick Edit Servers Console Problems EGL Generation Results

Server	Host name	Status	State
WebSphere Application Server v6.0	localhost	Started	Synchronized



Project Explorer

- Connector Projects
- EJB Projects
- Dynamic Web Projects
 - StoresWebDemo
 - EGLSource
 - (default package)
 - data
 - StoresLib.evl
 - pagehandlers
 - ShowAllCustom
 - StoresWebDemo.e
 - Web Site Navigation
 - Deployment Descriptor
 - Java Resources
 - WebContent
 - META-INF
 - theme
 - WEB-INF
 - ShowAllCustomers
- Other Projects
- Web Services
- Databases

```

package pagehandlers;

import data.*;

PageHandler ShowAllCustomers {view="ShowAllCustomers.jsp", onPageLoad=onPageLoad}

    Customers customer[];

    Function onPageLoad()
        get Customers;
    End

    function button1Action()
        i int;
        max int;
        Customer_file extfile;

        max = syslib.size(Customers);
        i = 1;

        while (i <= max)
            move Customers[i] to Customer_file byname;
            add Customer_file;
            i = i + 1;
        end
    end
End
  
```

A palette is not available.

Page Data >>4
No data components found.

Properties Quick Edit Servers Console Problems EGL Generation Results


Generation Results

IWN.VAL.9994.i 1/1 The part ShowAllCustomers was generated using build descriptor StoresWebDemoWebBuildOptions from file StoresWebDemo/E

IWN.VAL.9996.i 1/1 Generation completed for ShowAllCustomers with no errors.

ShowAllCustomers (StoresWebDemoWebBuildOptions)

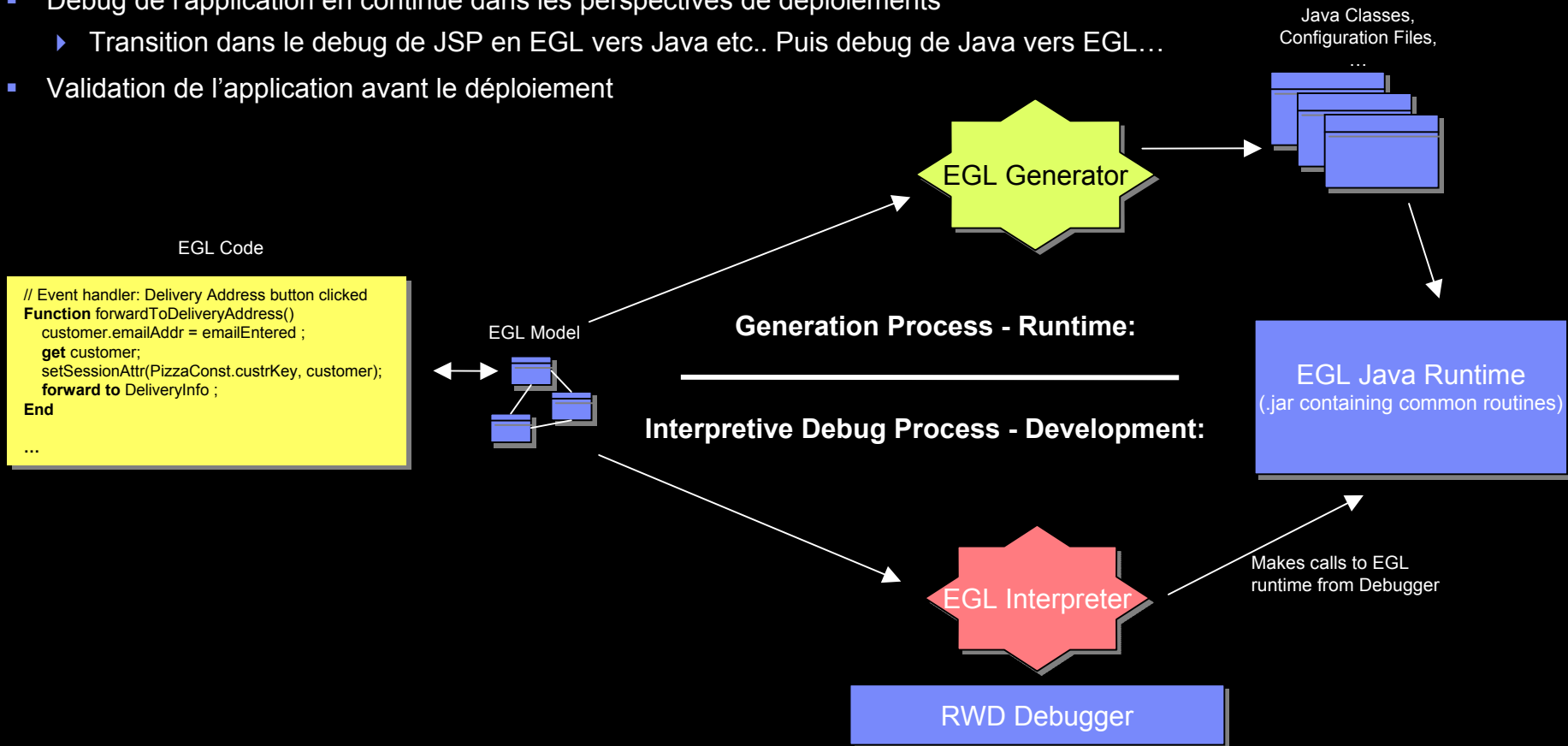
Phase de Developpement avec EGL

- Définition du Modèle de donnée
- Méthodes d'accès aux données
- Création de la Business logique
- Création des Interfaces Utilisateur
-  ■ Debug Business Logique
- Intégration avec des applications existantes
- Déploiement



EGL - Debugging

- EGL debugger
 - Breakpoints, visualisation des variables, change values, ...
 - Extension du debugger Eclipse de base
- Debug de l'application en continue dans les perspectives de déploiements
 - Transition dans le debug de JSP en EGL vers Java etc.. Puis debug de Java vers EGL...
- Validation de l'application avant le déploiement



Focus sur les "business problems" plutôt que sur la mise au point



Développement de Programme EGL - Debugging

Debug - CS50A.egl - IBM WebSphere Studio Application Developer

File Edit Navigate Search Project Profile Run Window Help

Debug

main line: 34 [CS50A]

- Thread [Command dispatcher] (Running)
- Thread [SunToolkit.PostEventQueue-0] (Running)
- Thread [AWT-EventQueue-0] (Running)
- Thread [AWT-Windows] (Running)
- Thread [AWT-InputMethodManager] (Running)
- Thread [Screen Updater] (Running)

Variables

- VAGen_EZE_WAIT_TIME = 0
- VAGen_EZE_ITEMLEN = 0
- CS10W
 - VAGen_ACTION =
 - EMPNO =
 - FIRSTNME =
 - MIDNIT =
- (char (1))

CS60A.egl CS50A.egl

```

31
32 function main()
33 // VAGen Info - initialization needed for migration
34 VAGen EZESYS = sysLib.getVAGSysType();
35 CS50F01-MAIN: CS50F01-MAIN();
36 end // end main
37 end // end CS50A
38
39
  
```

Outline

- VAGen_EZEREPLY : nul
- VAGen_EZE_WAIT_TIN
- VAGen_EZE_ITEMLEN :
- use -> CS50G
- main()
- CS50F01-MAIN()
- CS50F02-CONVERSE()
- CS50F03-RESET()
- CS50F04-ENTER()

Console [D:\wsed51-0317\ eclipse\jre\bin\javaw.exe (3/26/04 9:37 AM)]


Console Tasks EGL Debug Validation Errors

Writable Insert 34 : 43

Focus sur les "business problems" plutôt que sur la mise au point

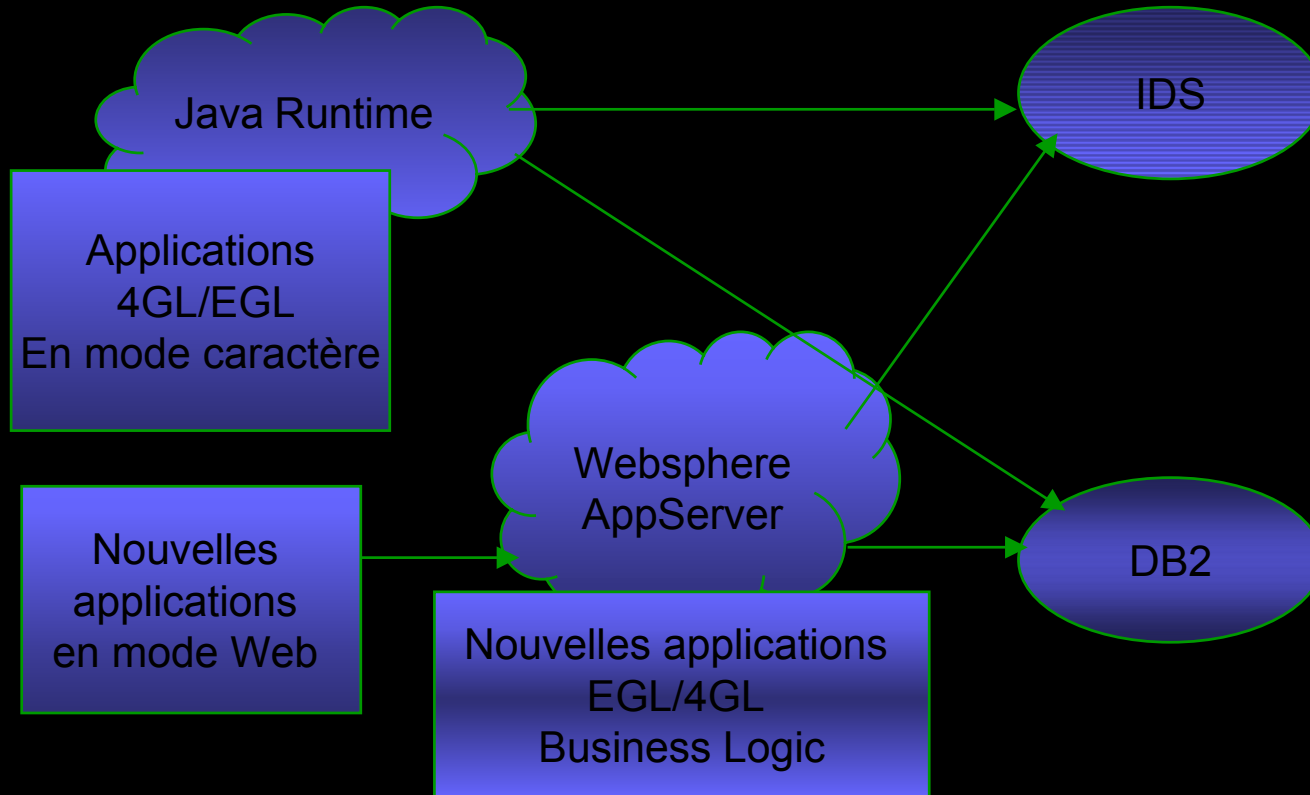


Phase de Developpement avec EGL

- Définition du Modèle de donnée
- Méthodes d'accès aux données
- Création de la Business logique
- Création des Interfaces Utilisateur
- Debug Business Logique
-  ■ Intégration avec des applications existantes
- Déploiement

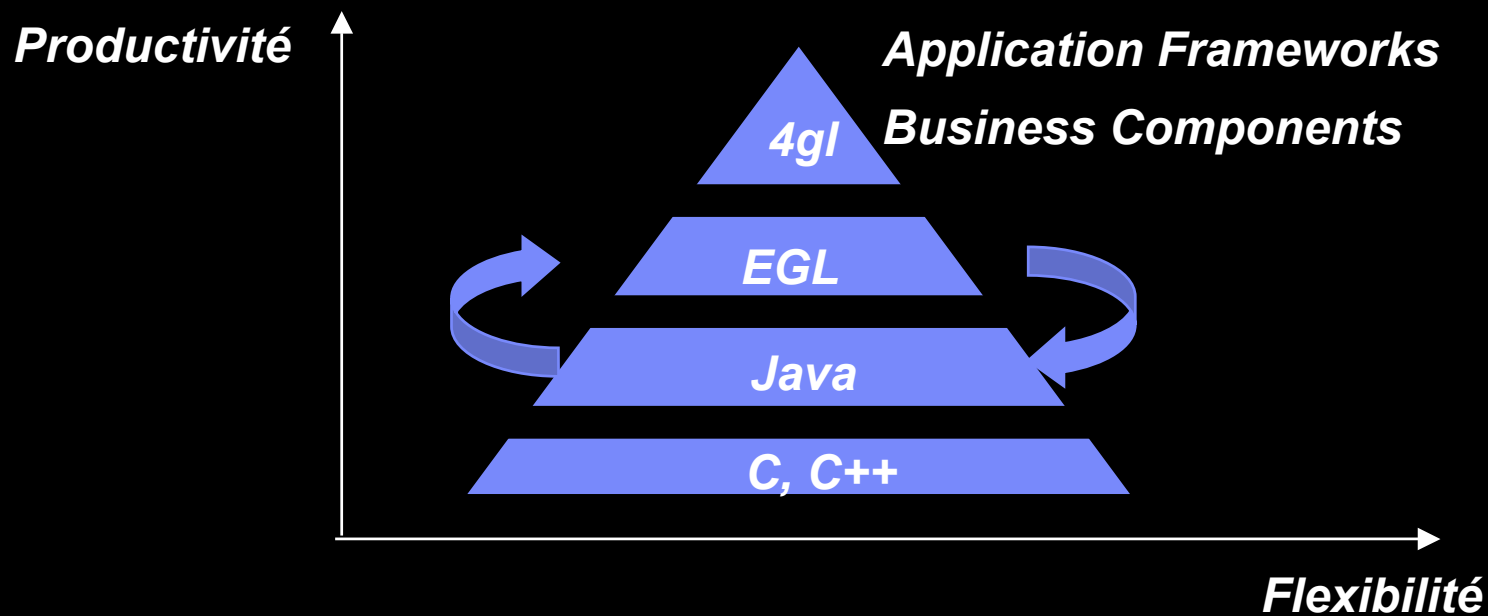


EGL Intégration avec des 4gl existants



EGL intégration et interopérabilité : garantie un maximum de flexibilité


- Intégration puissante pour EGL et Java
 - ✓ Maximum de productivité, maximum de flexibilité
 - ✓ Interopérabilité bidirectionnelle
 - ✓ Outil Intégré (workbench, Editeur, Debuggers)



Focus sur les "business problems" plutôt que sur la mise au point



Phase de Development avec EGL

- Définition du Modèle de donnée
- Méthode d'accès aux données
- Création de la Business logique
- Création des Interfaces Utilisateur
- Debug Business Logique
- Intégration avec des applications existantes
-  ■ Déploiement



EGL Déploiement


- Définition de la cible
- Installation du package JRE compatible et définie pour la cible
- Déploiement et exécution des binaires EGL
 - Windows
 - Linux, AIX (Solaris and HP in 6.0)
 - iSeries
 - zOS: CICS and Batch (IMS coming)



Focus sur les" business problemes" plutôt que sur la mise au point



EGL & 4GL

-  ■ Integration du 4GL
- Technical Overview
 - ▶ Différence entre 4GL et EGL
 - ▶ Syntaxe EGL



4GL Intégration avec Rational Web Développeur

WebSphere Studio is...

- an industry leading IDE
- well integrated with WebSphere Application Server (WAS), the leading web app server in the world

4GL customers will get . . .

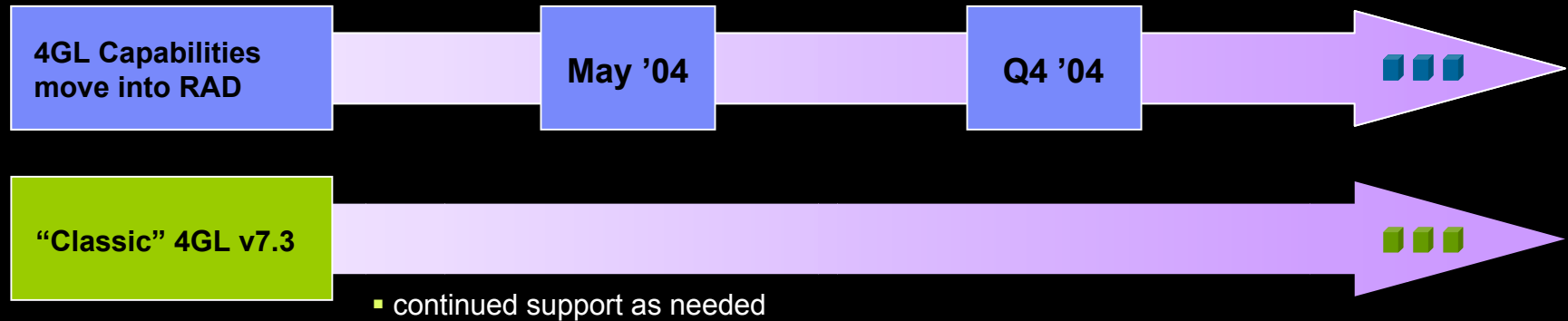
- web applications
- Windows platform
- open database connectivity
- context-sensitive EGL editor and debugger
- enhanced reporting
- continued text-based UI support

WebSphere Studio V5.1.2

- A well-defined and easy to use IDE
- New web application development
- 4GL-syntax incorporated in EGL
- Connectivity to DB2 and Informix
- Develop new batch database programs

Rational Studio V6.0 / 6.0.0.1

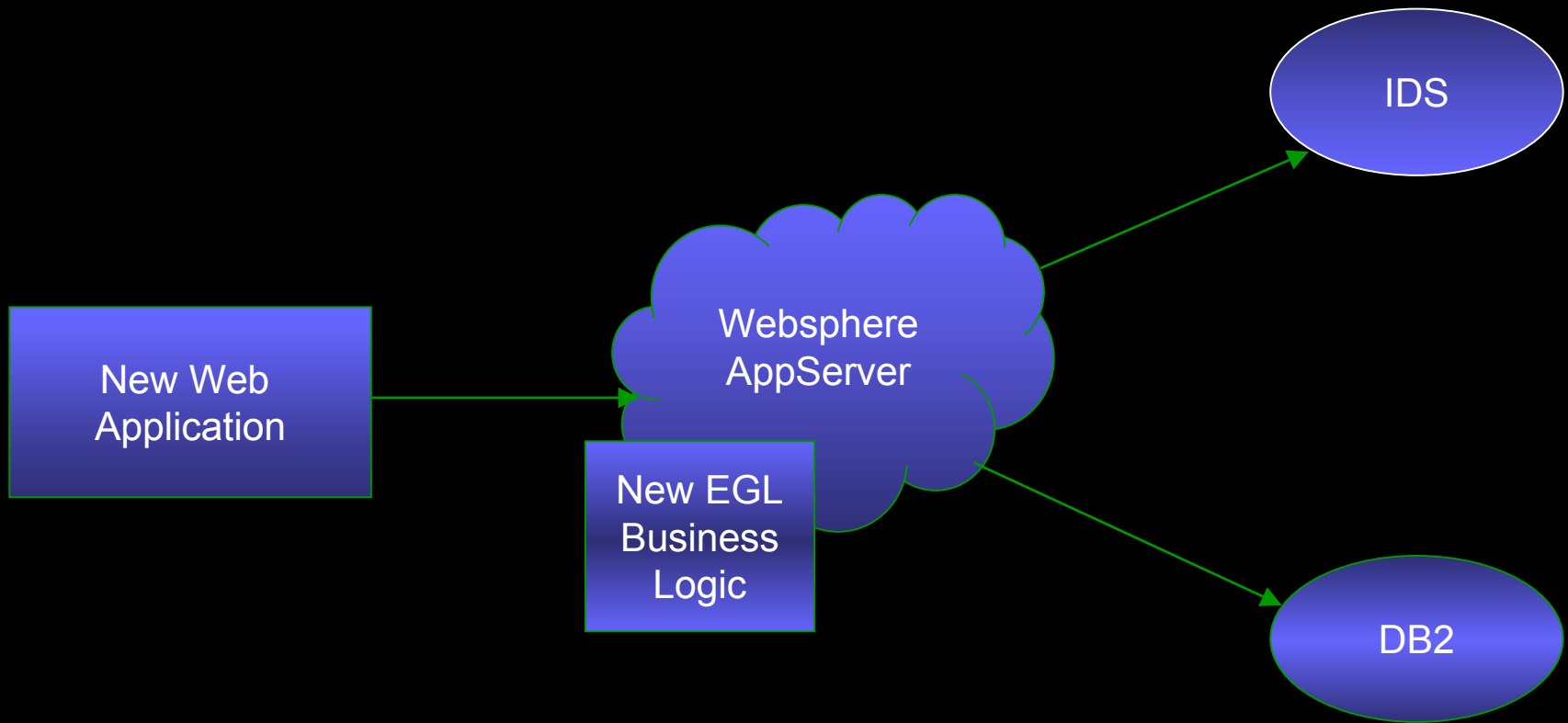
- Provide DB2 connectivity to migrated applications
- Develop new web applications using current I4GL business logic
- Develop new TUI applications or maintain existing ones
- Deploy on new platforms, including Windows
- Use enhanced reporting capabilities



2004



Websphere 5.1.2 – Phase 1 de l'offre EGL



- Développement de nouvelles applications WEB
- Syntaxe 4GL intégrée dans EGL
- Connectivité vers DB2 et Informix



EGL Release 5.1.2

▪ Que contient-elle ?

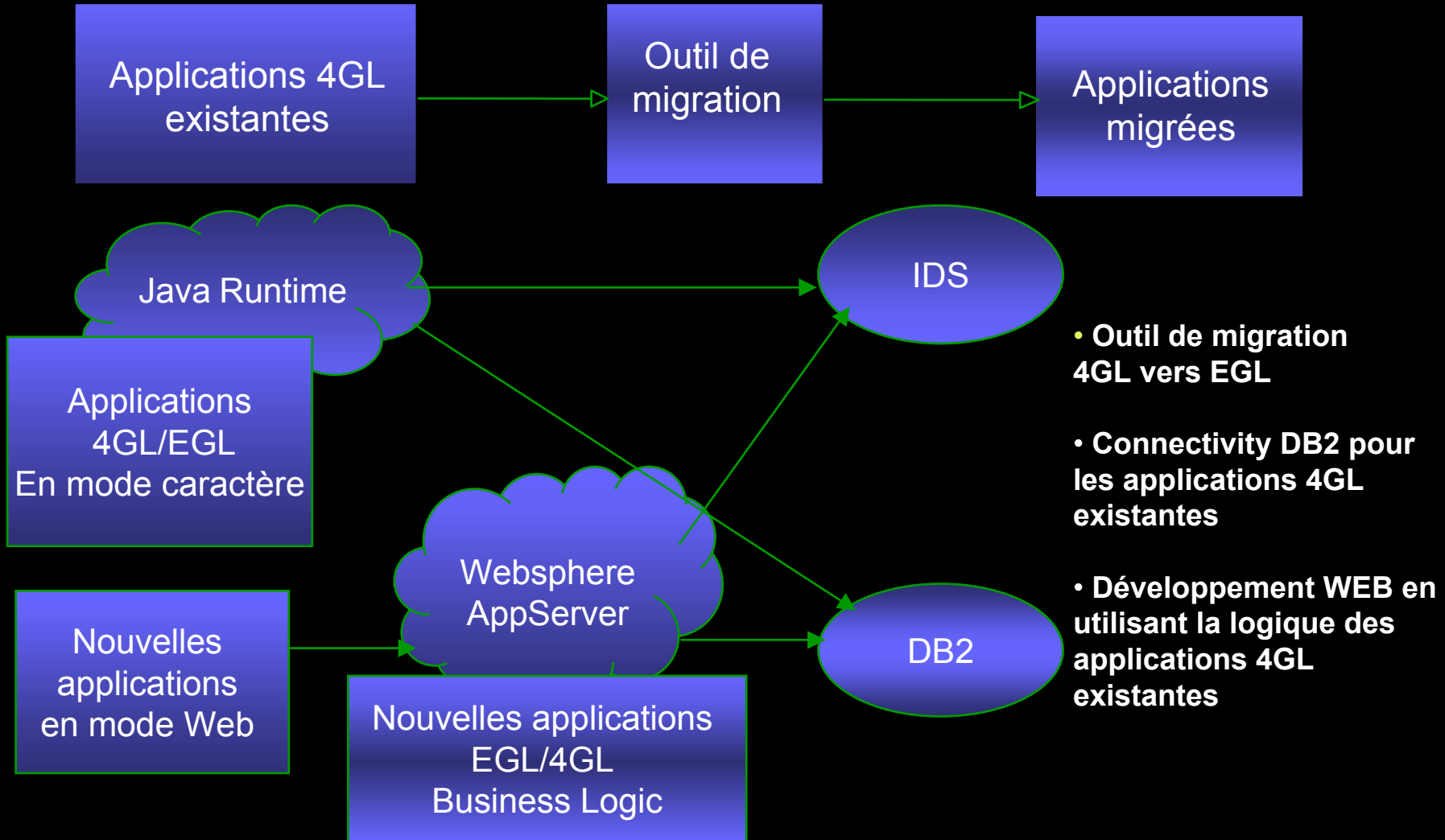
- ▶ Un outil de développement intégré facile à utiliser (IDE)
- ▶ La plupart des constructions I4GL's (syntaxe, types de données)
- ▶ Support des technologies JSF/JSP dans les programmes EGL.
- ▶ Support du déploiement sur zSeries et iSeries en mode caractère

▪ Que peut-on faire avec ?

- ▶ Se familiariser avec le langage EGL et WebSphere Studio
- ▶ Développer des applications 4GL batch
- ▶ Développer de nouvelles applications en mode Web
- ▶ Bénéficier d'un outil IDE utilisable dès aujourd'hui



Websphere 6.0 – Phase 2 de l'offre EGL



EGL 6.0.1 Outil de migration 4GL - EGL

▪ Que contient-elle ?

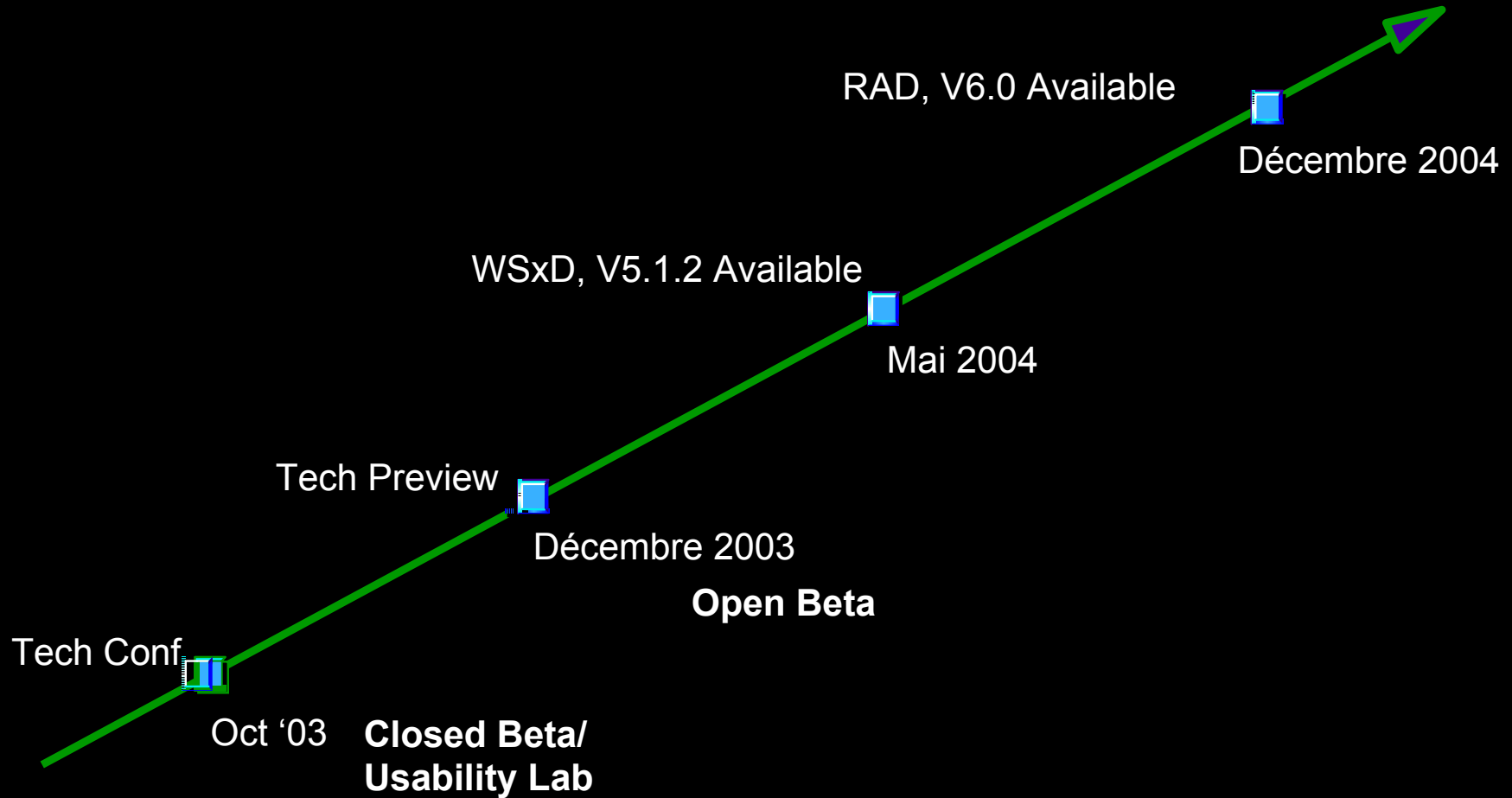
- ▶ Des outils de migration des applications 4GL vers EGL
- ▶ Outils de reporting évolués
- ▶ Support de l'interface en mode caractère
- ▶ Support de toutes les constructions, syntaxes, types de données I4GL encore absentes

▪ Que peut-on faire avec ?

- ▶ Développer de nouvelles applications en utilisant la logique applicative d'applications 4GL existantes (réutilisation)
- ▶ Développer de nouvelles applications en mode caractère ou maintenir des applications existantes
- ▶ Déployer sur de nouvelles plate-formes (Windows, Macintosh ...)
- ▶ Utiliser des nouvelles fonctionnalités de reporting



Quelques jalons



Rational Configurations

Rational Enterprise Developer

J2EE and zOS Modernization solutions

- COBOL and PL/I Development
- **EGL generation for zOS and iSeries**
- Web Services for zOS

Rational Application Dev.

Robust J2EE transactional Applications

Integrated business solutions

- EJB development
- Stored procedures
- Application profiling
- UML Class Diagrams
- Team Support Clearcase LT

- JCA Application Adapters & Tools (CICS, IMS, HOD, SAP, Custom)
- SOA Workflows with human interaction & compensation

Rational Web Developer

Web Applications

- Page & site visual composition
- JSP, Servlet, XML, & Web services tools
- Rich Java Client tools
- Rapid, simplified Development
 - JSF/WDO, Wizards, Struts Visual Builder
 - **Enterprise Generation Language (EGL)**
- Integrated Test Environment

Workbench:

A comprehensive Java IDE with unprecedented flexibility & extensibility

- Configurations & "perspectives"
- Plug-in Architecture
- Local & Team resource management
- Common IDE across Windows & Linux
- Superior Java programming facilities
- <http://www.eclipse.org>



EGL & 4GL

- Integration du 4GL
- Technical Overview
-  Différences entre 4GL et EGL
 - ▶ Syntaxe EGL
- Migration de 4GL vers EGL



EGL : Différences entre 4GL et EGL

- ▶ EGL business logic est utilisable par des applications J2EE
- ▶ EGL génère du Java; I4GL génère du C
- ▶ EGL supporte le mode caractère aussi bien que le mode Web
- ▶ EGL supporte les environnements Windows & Macintosh !
- ▶ EGL est beaucoup plus riche que 4GL (appel de code Java, Entrées/sorties fichiers ...)
- ▶ EGL dispose de connectivité JDBC
- ▶ EGL est intégré dans un outil de développement moderne et complètement intégré
- ▶ EGL peut être déployé vers les environnements zSeries et iSeries
- ▶ EGL permet de bénéficier des fonctionnalités de "Message Queuing" entre applications hétérogènes (MQ series)
- ▶ Permet de développer des "Web Services"



EGL & 4GL

- Integration du 4GL
- Technical Overview
 - ▶ Différences entre 4GL et EGL
 -  ▶ Syntaxe EGL
- Migration de 4GL vers EGL





Project Explorer showing a tree view of project components:

- EGL Projects
 - BatchTe
 - Enterprise A
 - Application C
 - Connector P
 - EJB Projects
 - Dynamic We
 - Other Project
 - Web Service
 - Databases
 - Database Se

```

-- Database used by this program
DATABASE prototype --EGL cannot execute CRE

-- Globals will defines variables used across
GLOBALS

DEFINE counter INTEGER
DEFINE iterations INTEGER
DEFINE myrecord INTEGER

DEFINE my_customer RECORD LIKE customer.*
DEFINE my_orders RECORD LIKE orders.*
DEFINE my_items RECORD LIKE items.*
DEFINE my_stock RECORD LIKE stock.*

END GLOBALS

--Main 4GL function calls sub funtions
MAIN

LET iterations=5      -- Number of loops
LET myrecord=0       -- Initilize Number

CALL DataLoad()      -- Loads data into da
CALL DataManupulation() -- UPDATE, ALTER s
CALL SelectLoop()    -- SELECT statement

END MAIN
  
```

```

//Put EGL File Contents Here

Program Test1

Use dataLibrary;

counter int;           // simple
iterations int;       // define
myrecord int;         // Variab

myCustomer Customer;
myOrders Orders;
myItems Items;
myStock Stock;

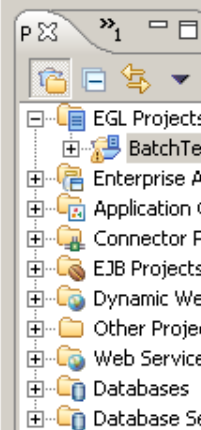
//Main 4GL function calls sub funtions
function main()
  //setCurrentDatabase("prototype");

  iterations = 5;    // Number of loops SELECT
  myrecord = 0;     // Initilize Number of

  dataLoad();       // Loads data into da
  commit();
  dataManipulation(); // execute sql UPDATE
  commit();
  selectLoop();     // SELECT statement i
  commit();
  
```

A palette is not available

No data components found.



```

FUNCTION DataLoad()
    WHENEVER ERROR STOP      -- STOPS THE PROG
    FOR counter = 1 TO 100    -- Repeat select r
        -- customer
        INSERT INTO customer VALUES (101,"Ludwig
        INSERT INTO customer VALUES (102,"Carole
        INSERT INTO customer VALUES (103, "Phili
        INSERT INTO customer VALUES (104, "Antho
        INSERT INTO customer VALUES (105, "Raymc
    -- orders
        INSERT INTO orders VALUES (1001,104, "up
        INSERT INTO orders VALUES (1002,105, "dk
        INSERT INTO orders VALUES (1003,106, "an
        INSERT INTO orders VALUES (1004,107, "be
        INSERT INTO orders VALUES (1005,108, "ik
    -- items
        INSERT INTO items VALUES (1,1001,1,"HRO"
        INSERT INTO items VALUES (2,2001,2,"REW"
        INSERT INTO items VALUES (3,3001,3,"UIL"
        INSERT INTO items VALUES (4,4001,4,"GHG"
        INSERT INTO items VALUES (5,5001,5,"OIU"
    --stock
        INSERT INTO stock VALUES (1,"HRO","basek
        INSERT INTO stock VALUES (2,"REW","footh
        INSERT INTO stock VALUES (3,"UIL","foosk
        INSERT INTO stock VALUES (4,"GHG","crick
        INSERT INTO stock VALUES (5,"OIU","kaba
    END FOR
END FUNCTION -- DataLoad ends
  
```

```

function dataLoad()
    handleHardIOErrors = 1;
    counter = 1;

    while ( counter <= 100 )
        execute #sql( INSERT INTO customer VALUES
        execute #sql( INSERT INTO customer VALUES
        execute #sql( INSERT INTO customer VALUES
        execute #sql( INSERT INTO customer VALUES
        execute #sql( INSERT INTO customer VALUES
        execute #sql( INSERT INTO orders VALUES (
        execute #sql( INSERT INTO orders VALUES (
        execute #sql( INSERT INTO orders VALUES (
        execute #sql( INSERT INTO orders VALUES (
        execute #sql( INSERT INTO orders VALUES (
        execute #sql( INSERT INTO items VALUES (1
        execute #sql( INSERT INTO items VALUES (2
        execute #sql( INSERT INTO items VALUES (3
        execute #sql( INSERT INTO items VALUES (4
        execute #sql( INSERT INTO items VALUES (5
        execute #sql( INSERT INTO stock VALUES (1
        execute #sql( INSERT INTO stock VALUES (2
        execute #sql( INSERT INTO stock VALUES (3
        execute #sql( INSERT INTO stock VALUES (4
        execute #sql( INSERT INTO stock VALUES (5
        counter = counter + 1;
    end
end
  
```

A palette is not available

No data components found.



Project Explorer showing a tree view of EGL Projects:

- EGL Projects
 - BatchTe
 - Enterprise A
 - Application C
 - Connector P
 - EJB Projects
 - Dynamic We
 - Other Project
 - Web Service
 - Databases
 - Database Se

Properties view showing: No data components found.

```

FUNCTION SelectLoop()
  WHENEVER ERROR STOP

  FOR counter = 1 TO iterations  -- Repeat s

    -- Select from customer table
    DECLARE a_curs CURSOR FOR
    SELECT * FROM customer
    FOREACH a_curs into my_customer.*
      LET myrecord=myrecord + 1
    -- DISPLAY my_customer.*
    END FOREACH

    -- Select from orders table
    DECLARE b_curs CURSOR FOR
    SELECT * FROM orders
    FOREACH b_curs into my_orders.*
      LET myrecord=myrecord + 1
    -- DISPLAY my_orders.*
    END FOREACH

    -- Select from items table
    DECLARE c_curs CURSOR FOR
    SELECT * FROM items
    FOREACH c_curs into my_items.*
      LET myrecord=myrecord + 1
    -- DISPLAY my_items.*
    END FOREACH

    -- Select from stock table
    DECLARE d_curs CURSOR FOR
    SELECT * FROM stock
  
```

```

function selectLoop()
  counter = 1;
  while ( counter <= iterations )

    open aCurs
    with
    #sql{
    select * from customer
    }
    for myCustomer;

    try
      while(sqlcode = 0)
        // get next is analogous to fetch

        get next from aCurs;
        if(sqlcode = 100)
          exit while;
        end
        myrecord = myrecord + 1;
      end
    onException
    end

    open bCurs
    with
    #sql{
    select * from orders
    }
    for myOrders;
    // Select from orders table
    try
      while(sqlcode = 0)
  
```

A palette is not available

EGL & 4GL

- Integration du 4GL
- Technical Overview
 - ▶ Différences entre 4GL et EGL
 - ▶ Syntaxe EGL
- ▶ ■ Migration de 4GL vers EGL



Comment la migration va-t-elle se passer ?

- **Principe: traduire automatiquement les programmes 4GL en EGL**
 - Puis maintenir, ou améliorer la logique applicative.

- **Outils ligne de commande:**
 - Utilitaire qui se basera sur un fichier de description de migration:
 - Description des répertoires
 - Des noms de fichiers et types de fichiers
 - Pour obtenir des fichiers EGL (abc.4gl → abc.egl)

- **RAD plug-in de migration**
 - ▶ Assistant graphique (wizard)
 - ▶ Génèrera automatiquement des fichiers EGL ainsi qu'un fichier "projet de migration"

- **L'outil prévoit de récupérer automatiquement du code 4GL**
 - ▶ Certains programmes 4GL nécessiteront une intervention manuelle
 - ▶ Certains autres ne migreront pas tout de suite (ou pas du tout)



Que devient la syntaxe I4GL et les reports?

- Presque toutes les directives 4GL seront supportées !

- Certaines sont encore sujettes à discussion:
 - VALIDATE LIKE
 - SCROLL
 - UPSCOL

- Les rapports seront repris intégralement sans problème.



QUID des types de données ?

- Tous les types de données Informix 4GL auront une équivalence en EGL !

- Certains seront remplacés par le type “SQL standard” correspondant
 - À cause de JDBC
 - Exemple: DATETIME YEAR TO FRACTION → TIMESTAMP(3)
 - Informix INTERVAL, MONEY, DATE, TIME, etc. sont repris tels quels

- Nouveau support pour les CLOBS et BLOBS



Qu'advient-il de mes applications caractère ?

- Le support de “Text User Interface (TUI)” est une priorité de EGL
 - Aussi bien pour 4GL que pour Visual Age Generator

- Le rendu avec EGL pourra être légèrement différent
 - Notamment dans le support des couleurs
 - Le support clavier sera compatible “touche pour touche”

- EGL supportera le mode caractère (TUI) pour:
 - Unix “curses”
 - 3270 style terminals



Mes programmes 4GL appellent des fonctions “C” !

- Beaucoup de programmes 4GL font appel à des fonctions C
- EGL dispose d’une “stack” au format 4GL
 - Les mêmes modules C peuvent être appelés depuis EGL
- EGL dispose d’une nouvelle interface JNI respectant le format de la pile I4GL
 - Permet au code java généré de faire appel aux fonctions C
- Les fonctions C doivent simplement être sous forme de “libraries partagées”
- L’outil de migration permettra de construire les “librairies partagées”
 - Création d’un makefile
 - Et relink avec la librairie JNI
- Tout code C qui ne “triche” pas passera sans problème
 - Notamment pas d’appels à des internes 4GL non documentés

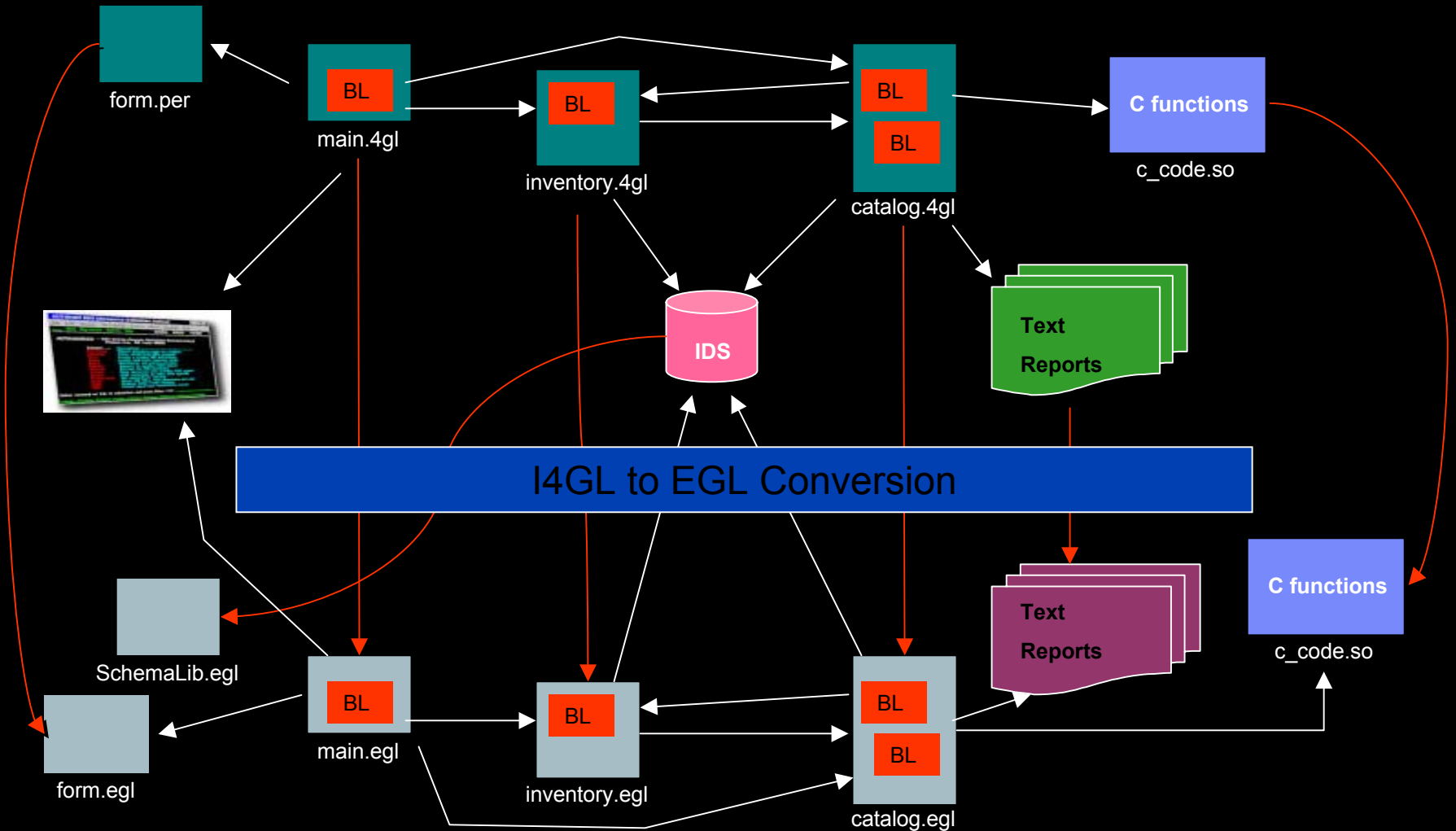


Et mon produit 4gl Forms ?

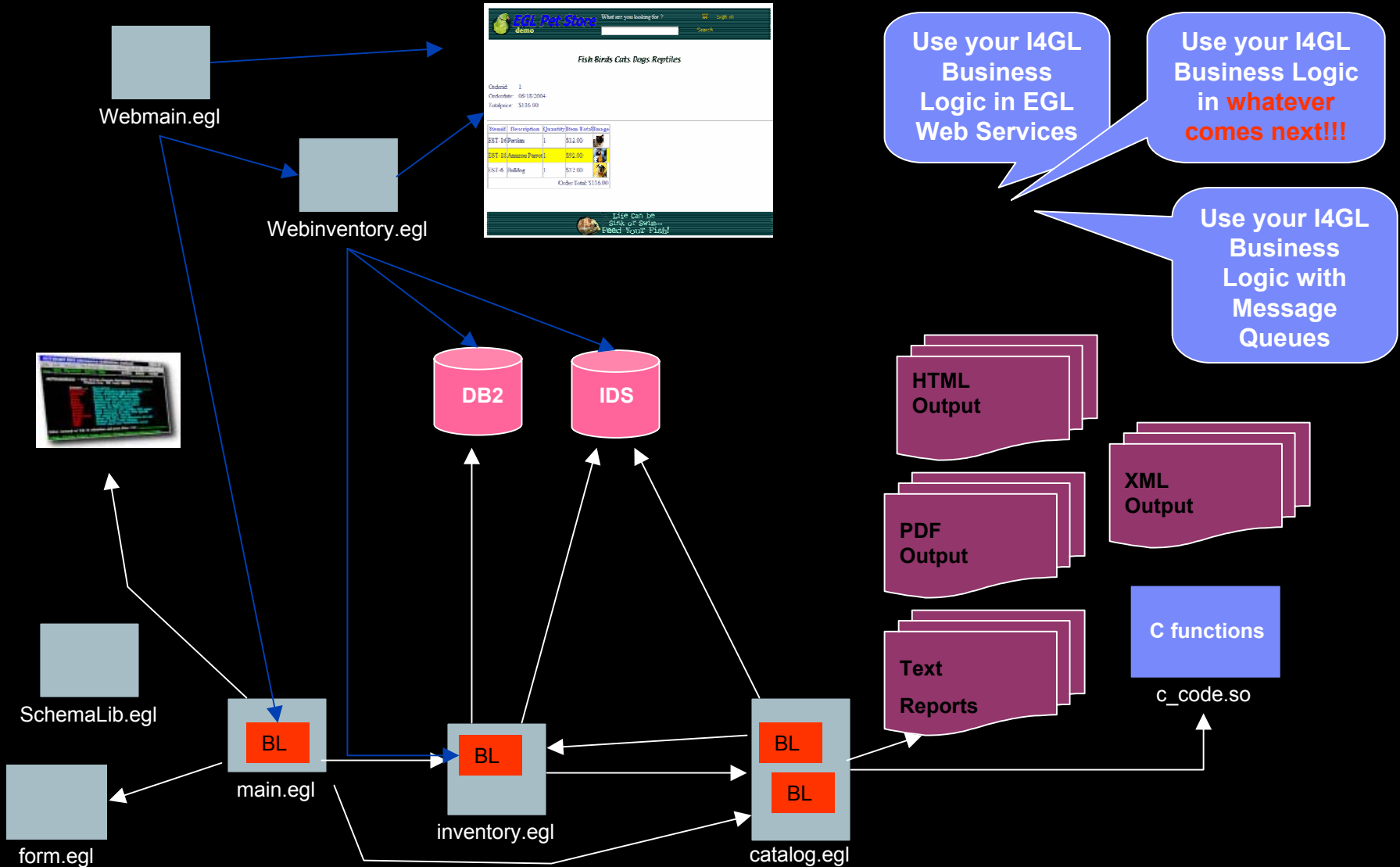
- Tous vos développements en 4gl Forms seront repris !



Migration : I4GL Programme ... Devient EGL Programme



Migration : Avec EGL vous pourrez



Agenda



- La problématique lié au 4GL
- Les solutions IBM Rational pour concevoir des applications J2EE
- EGL : LE FUTUR D'INFORMIX 4GL
- ➔ ■ Cloudscape/Derby et EGL
- Démonstration
- Bénéfices
- Questions / Réponses



Cloudscape

Cloudscape est une base de de donnée relationnelle 100 % java ultra légère. Cette base de donnée facilite le développement d'applications embarquées. – Aucune Administration!

*Fully Embeddable
or Server-based*



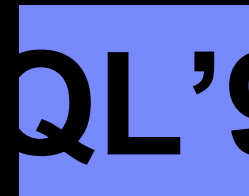
**Small
Foot Print**




**Java
Based**



**No
DBAs**



**Upwardly
Compatible with **



IBM Cloudscape: *Feature Rich; DBA Free...*

- **Cloudscape est une base de donnée pur java, embarquable**
 - ▶ Java → Un développement unique puis déploiement partout !
 - ▶ SQL'92 → Standard Ouvert → Portable
 - ▶ Transactionnel → Assure l'intégrité des données

- **Aucune Administration (Pas de DBA!)**
 - ▶ Toute la gestion peut être faite au niveau de l'application

- **Footprint: 2 MB fichier**
 - ▶ Petite Partie de l'application
 - ▶ La base de donnée est dans un fichier JAR

- **Cloudscape peut aussi fonctionner en tant que Serveur**

- **Un Plug-in Eclipse en cours de développement**
 - ▶ Facilitera dans le futur de le développement d'applications

- *100 % Java*
- *Zéro Administration*
- *Embarqué ou Server*



Distribution Strategies

- **Open Source**
 - ▶ Availability via Apache → 'Derby'
 - ▶ Linux Distributions
 - Planned to be available soon on popular Linux distributions

- **IBM Versions: Based on Derby**
 - ▶ No-charge download of IBM Cloudscape binary from IBM.COM
 - Can be included in developerWorks subscription
 - Application developer support via IBM developerWorks
 - Manuals, How-To Articles, Forums, ...
 - ▶ Fully Supported Product Version
 - Extended capability: e.g. Standard IBM Multiplatform Installer, ODBC Driver...
 - Support Options
 - Full IBM 24x7 IBM Support Available



EGL et Cloudscape

- RAD et RWD 6.0 supporte Cloudscape pour les applications EGL
- La combinaison Cloudscape et EGL permet le portage d'application de 4GL vers un framework Java
- Facilité de déploiement à travers les projets EGL
- EGL/Cloudscape: la solution optimale pour une solution sans administration, et avec une base donnée ultra légère
- Alternative pour DB2 UDB ou Informix IDS

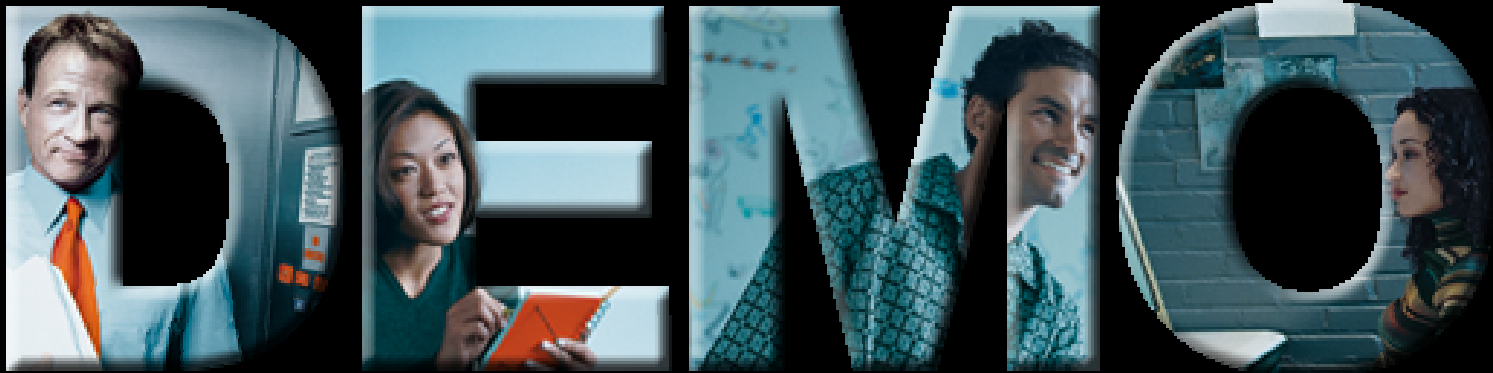


Agenda



- La problématique lié au 4GL
- Les solutions IBM Rational pour concevoir des applications J2EE
- EGL : LE FUTUR D'INFORMIX 4GL
- Cloudscape/Derby et EGL
- ➔ ■ Démonstration
- Bénéfices
- Questions / Réponses





Évaluer EGL ?

- Télécharger et installer la version d'évaluation Websphere Studio 5.1.2 pour Windows ou Linux
 - ▶ <http://www-106.ibm.com/developerworks/websphere/downloads/WSADsupport.html>
 - ▶ Télécharger tous les fichiers requis, plus WAS 5.1 Test Environment (WAS51.bin) et EGL support files (EGL.bin)

- Télécharger et installer la dernière version du driver Informix JDBC 2.21.JC5
 - ▶ <http://www-306.ibm.com/software/data/informix/downloads.html>



Évaluer EGL ? (Partie 2)

- Télécharger EGL Quick Start Tutorial, donnant un bon départ sur le développement Web avec EGL
 - ▶ <http://www-306.ibm.com/software/data/informix/4gl-wss/>
 - ▶ Le quick start tutorial est fourni avec le SGBD embarqué Cloudscape.

- Télécharger EGL 5.1.2 Reference Guide depuis IBM developerWorks...
 - ▶ <http://www-106.ibm.com/developerworks/websphere/zones/studio/egldocs.html>



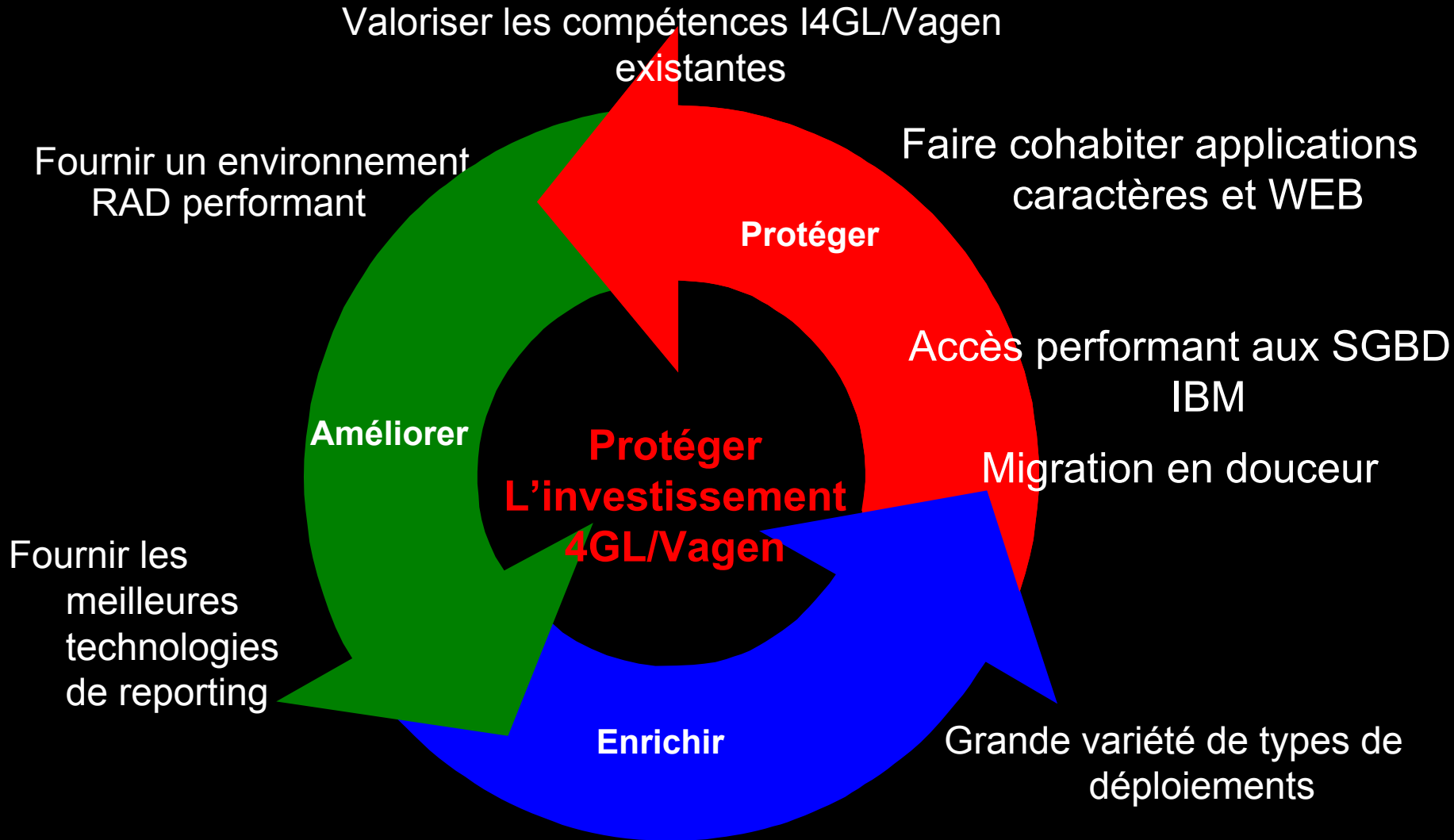
Agenda



- La problématique lié au 4GL
- Les solutions IBM Rational pour concevoir des applications J2EE
- EGL : LE FUTUR D'INFORMIX 4GL
- Cloudscape/Derby et EGL
- Démonstration
- ➔ ■ Bénéfices
- Questions / Réponses



Bénéfices



Outil de développement sophistiqué pour les applications WEB



Bénéfices : Protéger l'existant

- **Valoriser les compétences 4GL existantes**
 - ▶ EGL est un langage simple, robuste et de haut niveau.
 - ▶ Il comprend une majorité de constructions 4GL (toutes à terme)
 - ▶ Directives 4GL sont incorporés dans WebSphere EGL

- **Support des interfaces caractères et Web**
 - ▶ Mode 3270, 5250, Unix, Windows
 - ▶ Mais également interface Web (JSP's)

- **Amélioration de l'accès aux SGBD IBM**
 - ▶ Travail important réalisé au niveau des drivers JDBC vis à vis des bases Informix et DB2
 - ▶ Support de n'importe quelle database fournissant un driver JDBC

- **Support des 4GL Forms**
 - ▶ Les I4GL forms seront transformées en EGL forms

- **Outils de migration automatique 4GL -> EGL**
 - ▶ Prochainement en version 6.0.1



Bénéfices : Capitaliser sur les investissements Informix 4GL

- **Outil RAD très performant**
 - ▶ Fournit des templates, exemples qui font gagner beaucoup de temps
 - ▶ Couche d'abstraction, permet de travailler de façon similaire sur:
 - Tables Relationnelles
 - Fichiers standards
 - MQ Series data queues

- **Codage de l'application dans un seul langage**
 - ▶ Pas ou peu d'apprentissage
 - ▶ Evite la complexité de Java

- **Programmation identique quelle que soit l'interface utilisateur**
 - ▶ Mode Caractère et/ou mode Web
 - ▶ Développement simple d'applications Web
 - ▶ Facilite le passage du mode caractère vers le Web



Bénéfices : Enrichir l'existant

- **Outil sophistiqué de développement Web**
 - ▶ Complètement intégré à Webspere Studio
 - ▶ Permet le développement en "Equipe"
 - ▶ fonctionnalités de debugging
 - ▶ Outil de conception graphique des formes d'écrans et des reports
 - ▶ Outil de reporting sophistiqué

- **Nombreuses options de déploiement**
 - ▶ En Java vers Windows, Unix, iSeries, macintosh ...
 - ▶ J2EE (Servlets, JSP, EJB, Web Services)
 - ▶ En COBOL vers iSeries and zSeries



Bénéfices

- EGL offre toutes les caractéristiques que les développeurs I4GL demandent depuis des années:
 - IDE
 - Capacité à générer des applications WEB
 - Support des environnements Windows
 - Connectivité vers d'autres databases
 - Debugger graphique
 - Outils de reporting sophistiqués
 - Migration de votre ancienne application 4GL
 - Aucune compétence Java nécessaire
 - Faible effort d'apprentissage, formation très courte
 - Prise en main rapide
 - Développement rapide
 - Langage simple
 - Et plus encore ...

INFORMIX 4GL un avenir chez IBM !



Agenda



- La problématique lié au 4GL
- Les solutions IBM Rational pour concevoir des applications J2EE
- EGL : LE FUTUR D'INFORMIX 4GL
- Cloudscape/Derby et EGL
- Démonstration
- Bénéfices
- ➔ ■ Questions / Réponses

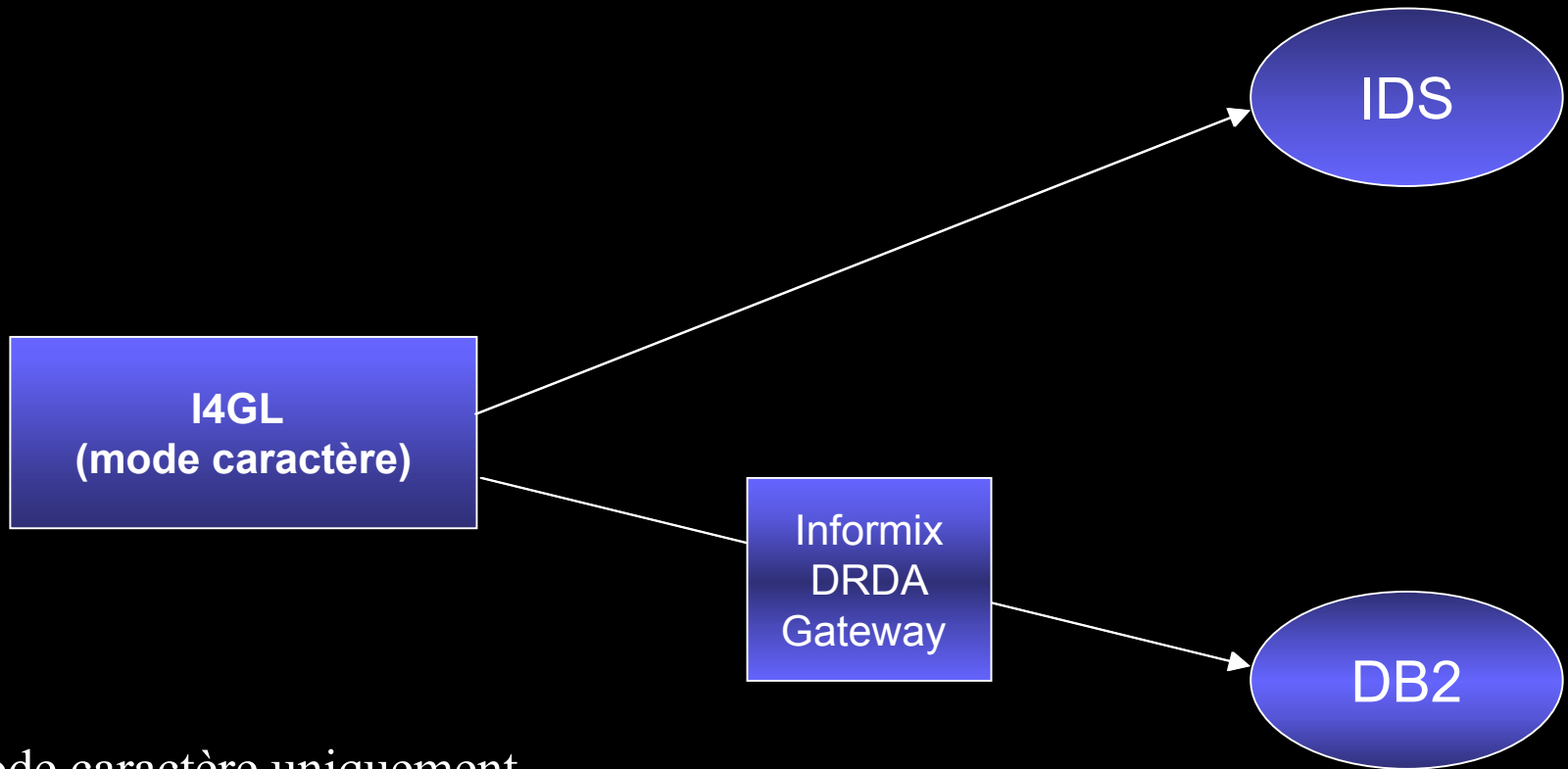




Aomar.Bariz@fr.ibm.com



Aujourd'hui I4GL, V7.32.xC2



Mode caractère uniquement

Bases de données Informix uniquement en natif

