IBM Software Group

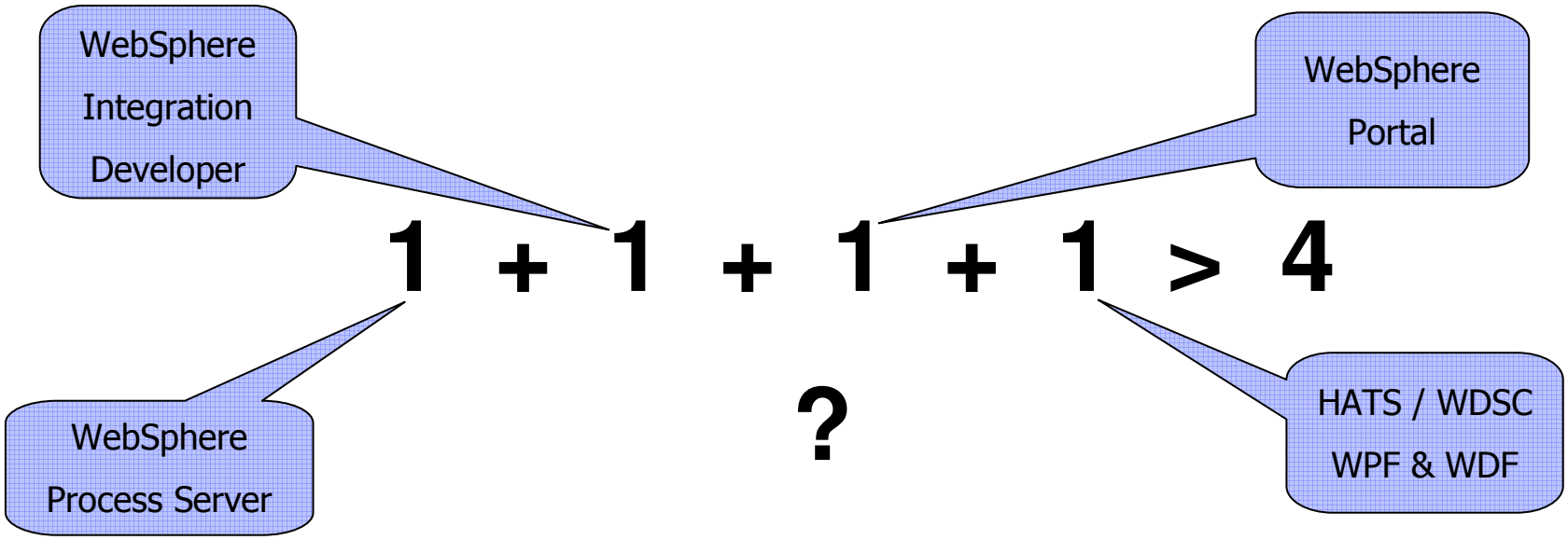
Order Management Process

Lab: Introduction

An IBM Proof of Technology

Introduction

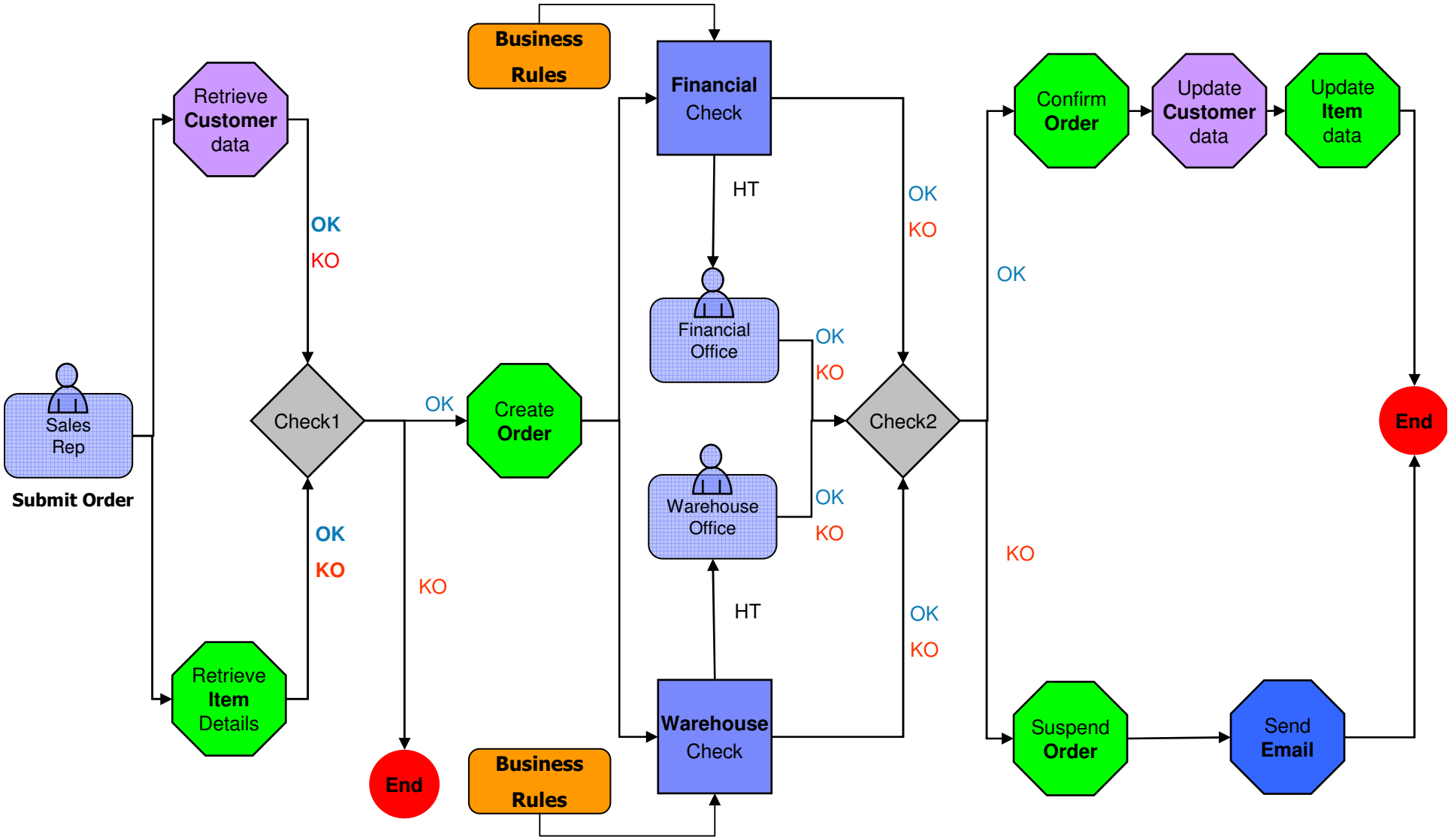
- The objective of this demo is to demonstrate the capabilities of a set of IBM Software product to build a **Process Portal** solution
- Specific product features will be exploited to show how **Composite Applications** can be designed and deployed over IBM middleware
- The final result will be the following . . .
- While you can discover and agree (we hope) with the value of each product, the most from these products come from their integration to build a solution !



Order Management Process

- The Demo will show how to build and execute an automated **Order Management Process**
- The process will start from three simple data elements:
  - Customer ID
  - Item ID
  - Item Qty
- Financial and Warehouse checks are performed before accepting the order
- According to specific process situations, process may require the involvement of users to complete authorization tasks:
  - Customer financial situation
  - Item shortage at warehouses

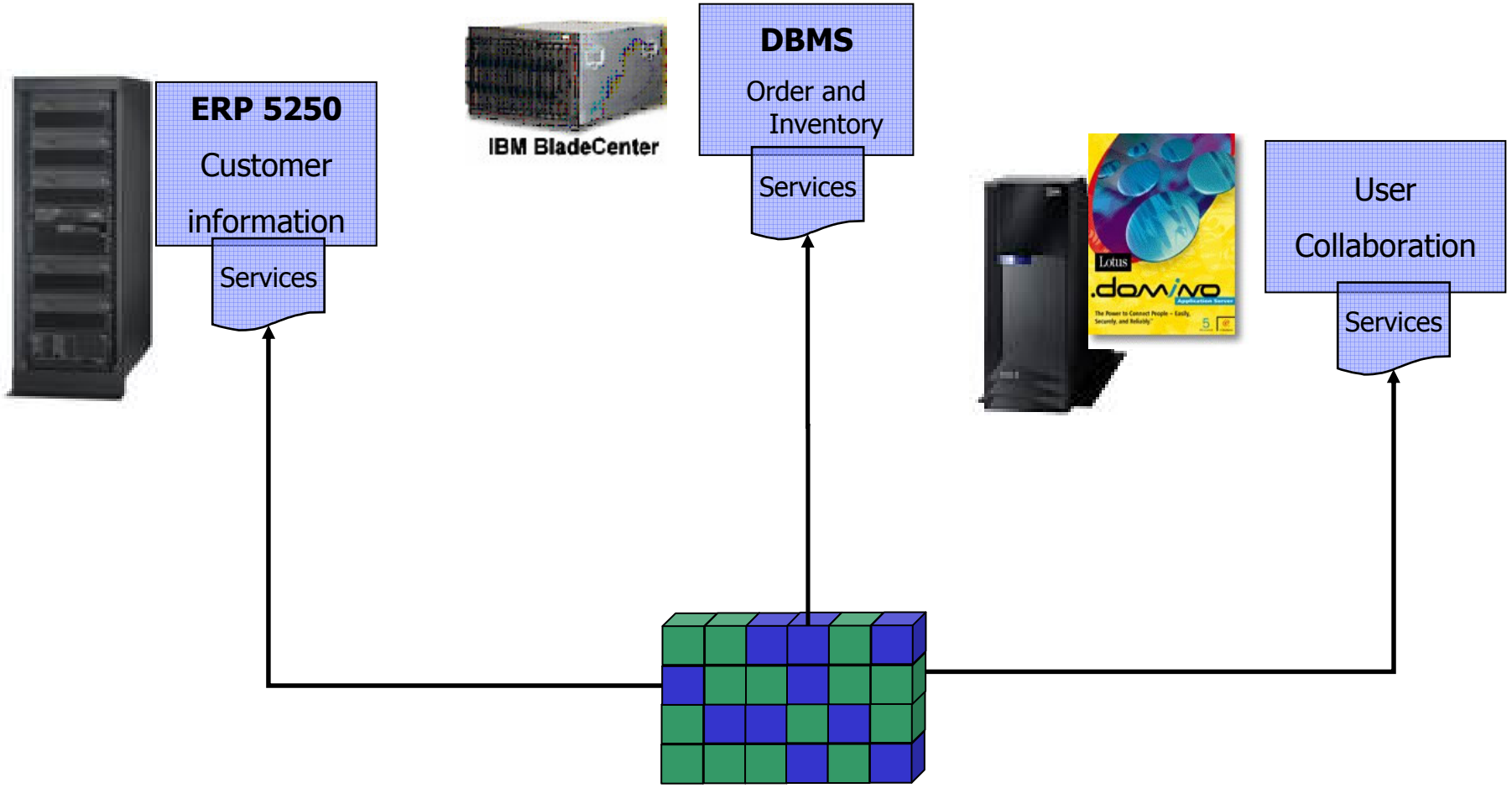
Order Management Process: High-level overview



Order Management – application involved

- Process will reuse existing applications and data:
  - **Customer information**: interactive 5250 application based on System i5
  - **Order and Inventory data:** hosted on DB2 UDB
  - **User collaboration environment:** based on Lotus Domino

Order Management as a Composite Application



System i5 environment

- The System i5 demo environment consists of the **SWGDEMO** library.
- This library contains data, programs and other artifacts created to provide all that you need to manage a simple **Customer** archive.
- Different kind of programs (RPG and CL) were created to manage this simple archive in order to provide a sample of use of the following i5/OS integration patterns:
  - **Reface**
    New presentation logic tied to the original one and artifacts to automate the interaction with the 5250 application
    - Macros will be created using HATS and then published as Web Services
  - **Restructure**
    Separate presentation from the business logic
    - Java Bean will be created using WDSc and then published as Web Services
- The **SWGDEMO** library represents a demo environment to demonstrate how you can (or better, have to) transform your i5/OS monolithic application in order to "**open**" it to the integration with other external environment and let it to participate into a **Composite Application** scenario.

System i5 environment: data

- The data are stored into a PF-DTA → **CUSTM00F** and two different Logical Files (LF) where built to access the customers:
  - **CUSTM01L** (→KEY = CUSTID)
  - **CUSTM02L** (→KEY = STATEC)

CUSTM00F

```
Session A - [24 x 80]
File  Edit  View  Communication  Actions  Window  Help
Columns . . . :    1  71                                  SWGDEMO/QDDSSRC
SEU==> _____________________________________________________ CUSTM00F
FMT PF .....A..........T.Name++++++RLen++TDpB......Functions++++++++++++++++++
       *************** Beginning of data ************************************
0001.00      A          R CUSTM
0002.00      A            CUSTID         6S 0
0003.00      A            DESCPT        30A
0004.00      A            CUSTTP         1A
0005.00      A            FISCCD        16A
0006.00      A            ADDRSS        30A
0007.00      A            ZIPCDE         5  0
0008.00      A            CITYCS        30A
0009.00      A            STATEC         2A
0010.00      A            SOLDTD         9  0
0011.00      A            BUDGET         9  0
       ****************** End of data ***************************************
```

CUSTM01L

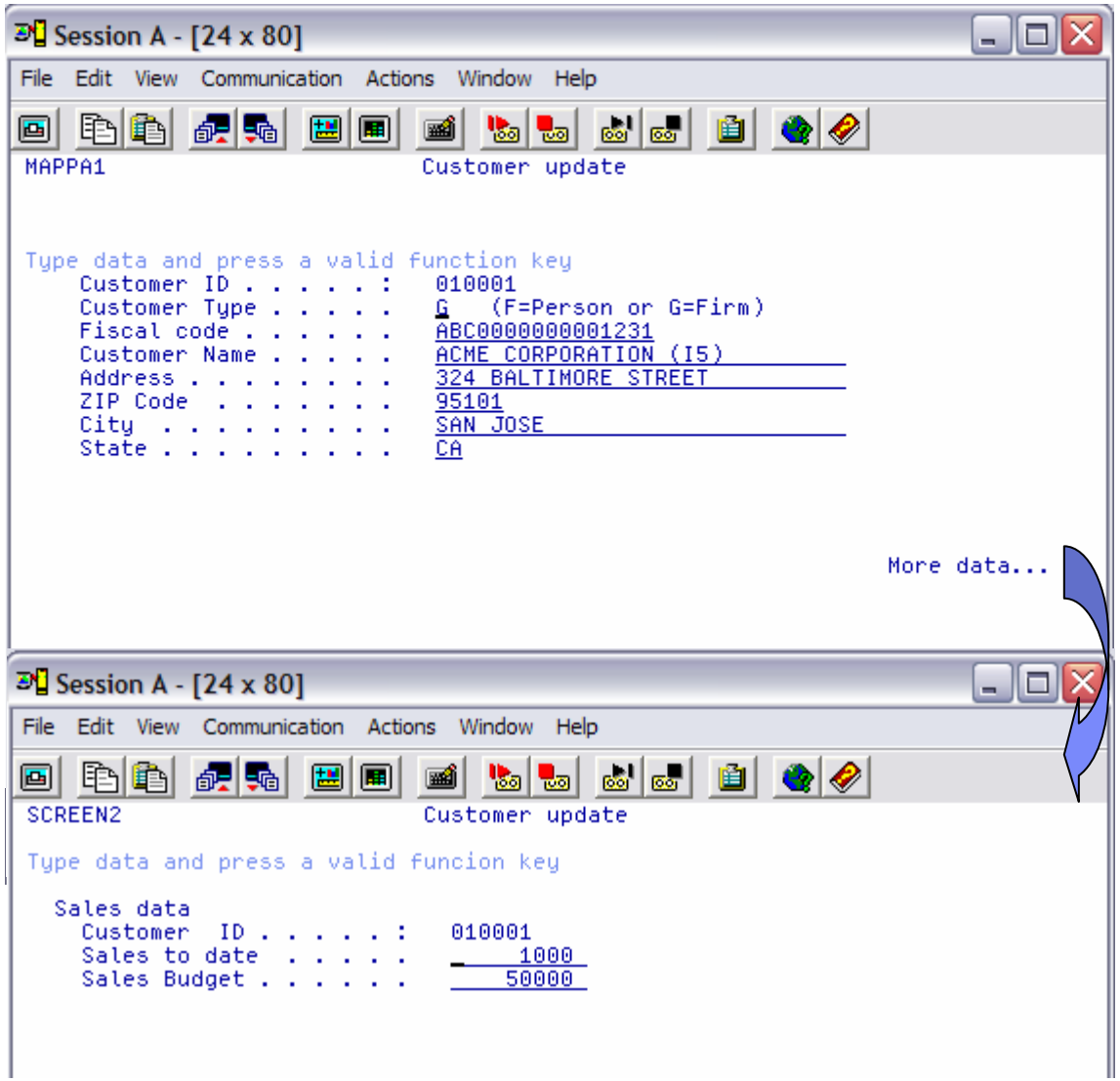
```
Session A - [24 x 80]
File  Edit  View  Communication  Actions  Window  Help
Columns . . . :    1  71                                  SWGDEMO/QDDSSRC
SEU==> _____________________________________________________ CUSTM01L
FMT LF .....A..........T.Name++++++.Len++TDpB......Functions++++++++++++++++++
       *************** Beginning of data ************************************
0001.00      A                                  UNIQUE
0002.00      A          R CUSTM                 PFILE(CUSTM00F)
0003.00       *
0004.00      A          K CUSTID
       ****************** End of data ***************************************
```

CUSTM02L

```
Session A - [24 x 80]
File  Edit  View  Communication  Actions  Window  Help
Columns . . . :    1  71                                  SWGDEMO/QDDSSRC
SEU==> _____________________________________________________ CUSTM02L
FMT LF .....A..........T.Name++++++.Len++TDpB......Functions++++++++++++++++++
       *************** Beginning of data ************************************
0001.00      A          R CUSTM                 PFILE(CUSTM00F)
0002.00       *
0003.00      A          K STATEC
       ****************** End of data ***************************************
```

System i5 environment: original application functionalities

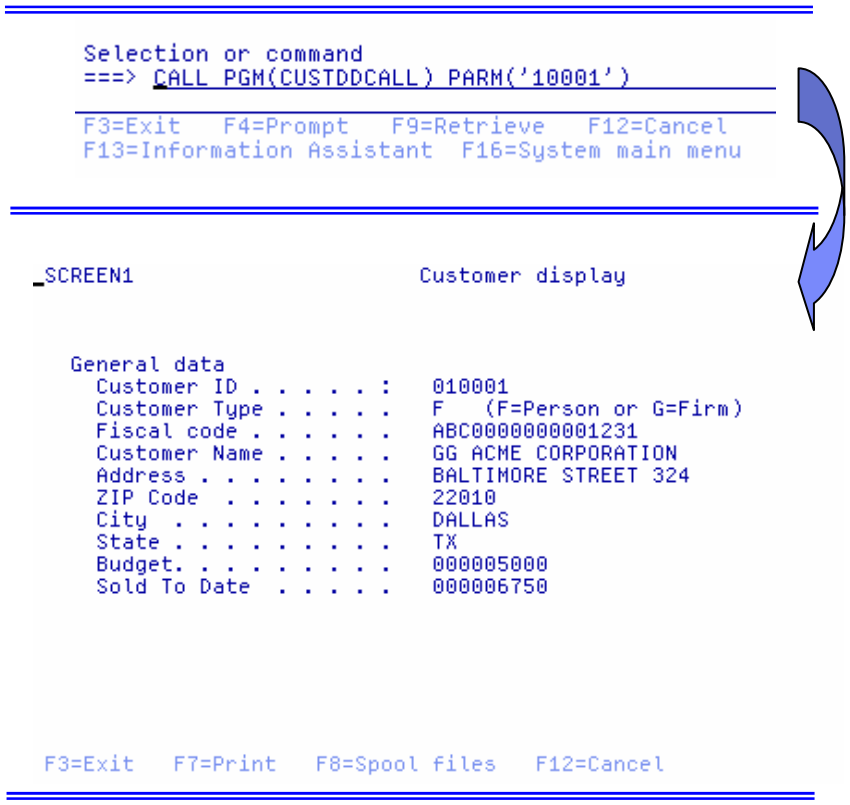
- Once logged to the system, the customer application **Starting Menu** will appear
- Choose the available option "**1**" to access the customer list
- Use option "**5**" to → **Display** Customer data
- Use option "**2**" to → **Update** Customer data



System i5 environment: programs for Reface Pattern

CUSTDD & CUSTDDCALL

- These programs provide direct interactive access to the **delete** and **display** actions
- **CUSTDD** → is the interactive RPG program that manages the Display File and the delete/display functions.
  - The program receives some parameter. The most important are:
    - **FUNZ** → define the **type of function**.
    - **CUSTID** → the **customer ID** you want to work with
    - **RETCD** → the **Return Code** from the program.
- **CUSTDDCALL** → is a CL program used to launch the RPG one, customized to perform only a display function (→ **FUNZ** = "**DS**")
  - The program receives only one parameter: → **CUSTID** (that represent the customer you want to work with)

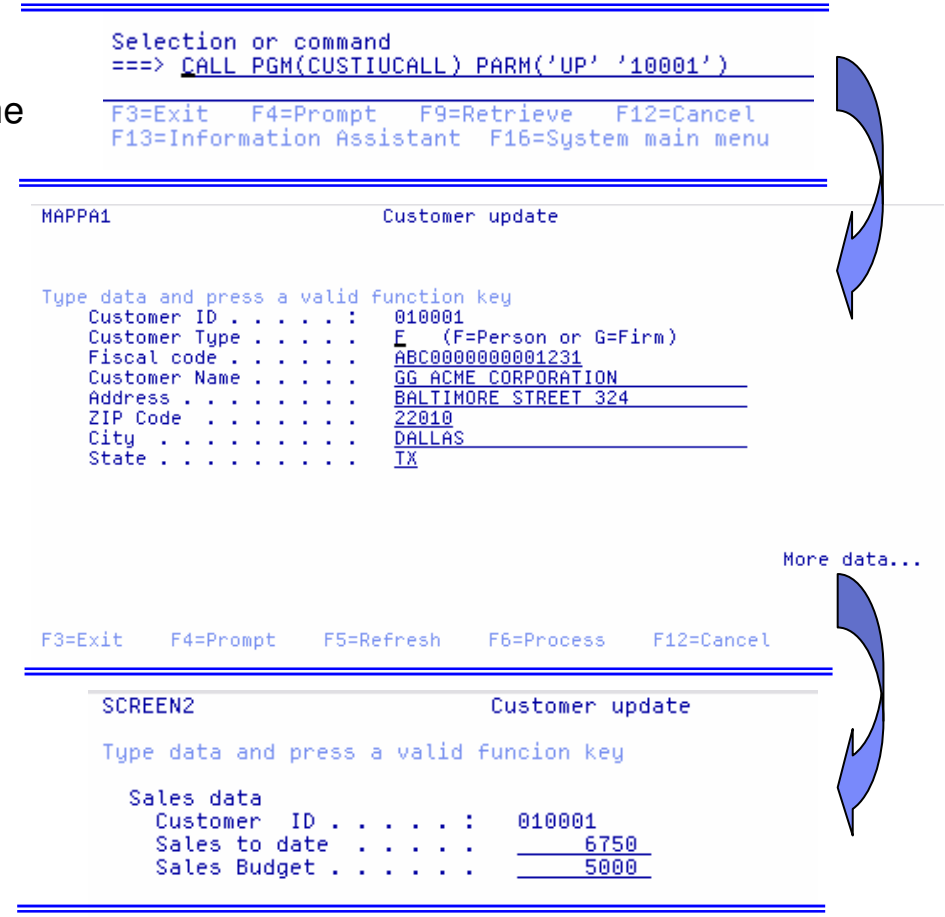


- You can use **HATS** to easily create a macro to interact with the programs and then wrap this macro into a Web Service. This Web Service will be used to **retrieve** Customer data

System i5 environment: programs for Reface Pattern

CUSTIU & CUSTIUCALL

- These programs provide direct interactive access to the **insert** and **update** actions
- **CUSTIU** → is the interactive RPG program that manages the Display File and the insert/update functions.
  - The program receives some parameter. The most important are:
    - **FUNZ** → define the **type of function**.
    - **CUSTID** → the **customer ID** you want to work with
    - **RETCD** → the **Return Code** from the program.
    .
- **CUSTIUCALL** → is a CL program used to launch the RPG one
  - The program receives two parameters:
    - **FUNZ** → Must be set to "**UP**" or "**IN**"
    - **CUSTID** (that represent the customer you want to work with



- You can use **HATS** to easily create a macro to interact with the programs and then wrap this macro into a Web Service. This Web Service will be used to **update** Customer data

System i5 environment: programs for Restructure Pattern

CUSTDATA

- This program represents a sample of “batch” access to display customer data. It doesn’t perform any data display, but simply works with a PLIST that summarize input and output data.

```
*ENTRY      PLIST
            PARM                Custid          6 0
            PARM                Name           30
            PARM                Type            1
            PARM                FiscalCode     16
            PARM                Address        30
            PARM                ZipCode         5 0
            PARM                City           30
            PARM                State           2
            PARM                SoldToDate      9 0
            PARM                Budget          9 0
            PARM                RetCod          2
```

- You can use **WDSc** to easily create a Java Bean to wrap the RPG program call and then wrap this macro into a Web Service. This Web Service will be used to **retrieve** Customer data

System i5 environment: programs for Restructure Pattern

CUSTDATAUP

- This program represents a sample of “batch” access to **update** some customer data. It doesn’t perform any data display, but simply works with a PLIST that summarize input and output data.
  - In the demo scenario, we need to update only the **SOLDTD** attribute of the customer whenever an order is entered in a → **CONFIRMED** status. This is why in this case the PLIST of the program is restricted to only three parameters.

```
input/output program parameters
   *ENTRY        PLIST
                 PARM                    CUSTID
                 PARM                    AMOUNT           9 0
                 PARM                    RETCD
```

- You can use **WDSc** to easily create a Java Bean to wrap the RPG program call and then wrap this macro into a Web Service. This Web Service will be used to **update** Customer data

DB2 UDB environment

- The DB2 UDB demo environment consists of the **SWGWPSWP** database.
- This database contains the tables to manage the following entities:
  - **Items**
  - **Warehouses**
  - **Orders**
  - **Customer Financial data**
    - This table is simply used to simulate an historical (last three years) financial situation for each customer

DB2 UDB environment

- The following pictures describes the structure of the different tables:

Table - SWGITEMS

Schema : DB2ADMIN
Creator : DB2ADMIN
Columns : 3

Columns

Key	Name	Data type	Length	Nullable
1	ITEMID	VARCHAR	10	No
	ITEMNAME	VARCHAR	30	Yes
	PRICE	INTEGER	4	No

Table - SWGWAREHOUSE

Schema : DB2ADMIN
Creator : DB2ADMIN
Columns : 3

Columns

Key	Name	Data type	Length	Nullable
1	WHSID	VARCHAR	10	No
	WHSDESC	VARCHAR	30	No
	WHSLOCATION	VARCHAR	30	No

Table - SWGITEMWHS

Schema : DB2ADMIN
Creator : DB2ADMIN
Columns : 5

Columns

Key	Name	Data type	Length	Nullable
1	ITEMID	VARCHAR	10	No
2	WHSID	VARCHAR	10	No
	INDELIVERY	INTEGER	4	No
	ORDERED	INTEGER	4	No
	STOCK	INTEGER	4	No

Table - SWGORDERS

Schema : DB2ADMIN
Creator : DB2ADMIN
Columns : 9

Columns

Key	Name	Data type	Length	Nullable
1	ORDID	INTEGER	4	No
	CUSTID	VARCHAR	20	Yes
	AMOUNT	INTEGER	4	Yes
	SUBMITTERID	VARCHAR	30	Yes
	STATE	VARCHAR	10	Yes
	CREATIONDATE	TIMESTAMP	10	Yes
	COMPLETIOND...	TIMESTAMP	10	Yes
	ITEMID	VARCHAR	10	No
	ITEMQTY	INTEGER	4	No

Table - CUSTOMERFINANCIAL

Schema : DB2ADMIN
Creator : DB2ADMIN
Columns : 4

Columns

Key	Name	Data type	Length	Nullable
	CUSTID	INTEGER	4	No
	"YEAR"	VARCHAR	4	No
	FORECAST	INTEGER	4	No
	YEARSOLD	INTEGER	4	No

Order Management: application architecture guidelines 1/3

- Demo has been implemented by applying principles and guidelines of SOA and composite applications
  - Applications to be integrated are **services** described through standard interfaces (WSDL)
  - Data exchanged between applications and integration broker (WebSphere Process Server) are represented as **XML documents**
  - Business Process Logic is described through **Business Process Execution Language** (BPEL)
  - Process is decoupled from applications to be integrated. Enterprise Service Bus **mediations** will manage:
    - Interaction from the process to applications
    - Data transformation from XML to application specific types

Order Management: application architecture guidelines 2/3

- “Ready to use” adapters are exploited to simply the integration of external applications
  - **WebSphere Adapter JDBC** to integrate relational data
  - **WebSphere Adapter for eMail** to exchange information via email
  - **Host Access Transformation Services** to publish 5250 interactive applications as Web Services
  - **WDSC Program Call Bean** to call RPG programs and wrapped as Web Service as well

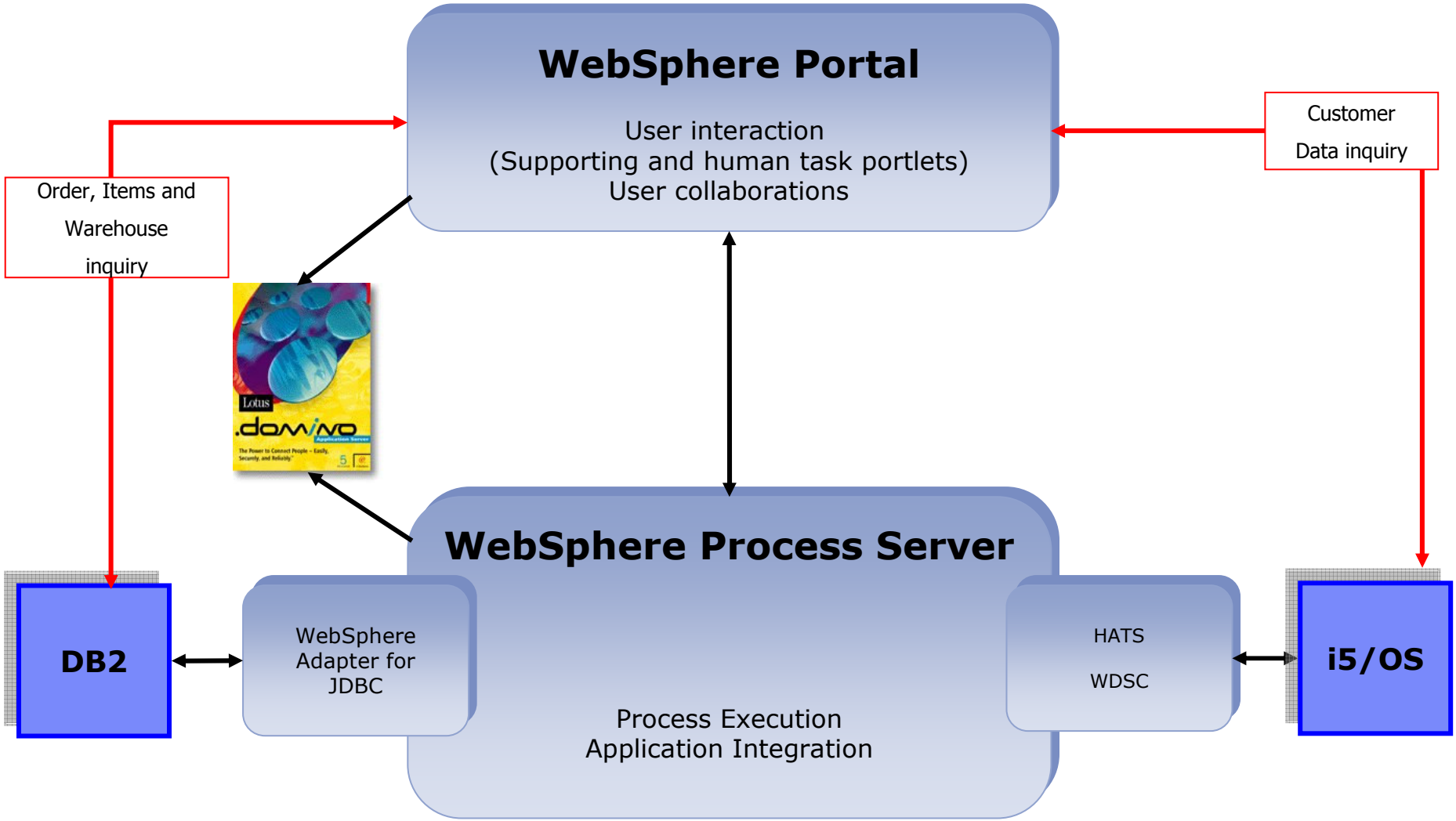
Order Management: application architecture guidelines 3/3

- A portal based user interface (WebSphere Portal) is used to access to process human tasks
  - Portal aggregates human task specific portlets to other supporting portlets
  - Supporting portlets provide users with specific information to complete the human task
  - The aggregation of these portlets is a “composite applications”
  - Resulting composed page is the “task portal”
- Supporting portlets have been developed using
  - **WebSphere Portlet Factory & Dashboard Framework**
  - **Web Content Management**
  - **Portal Document Manager**
  - **Personalization**
- Other portlets are directly provided by portal
  - Domino Web Access
    - to access user mail-box

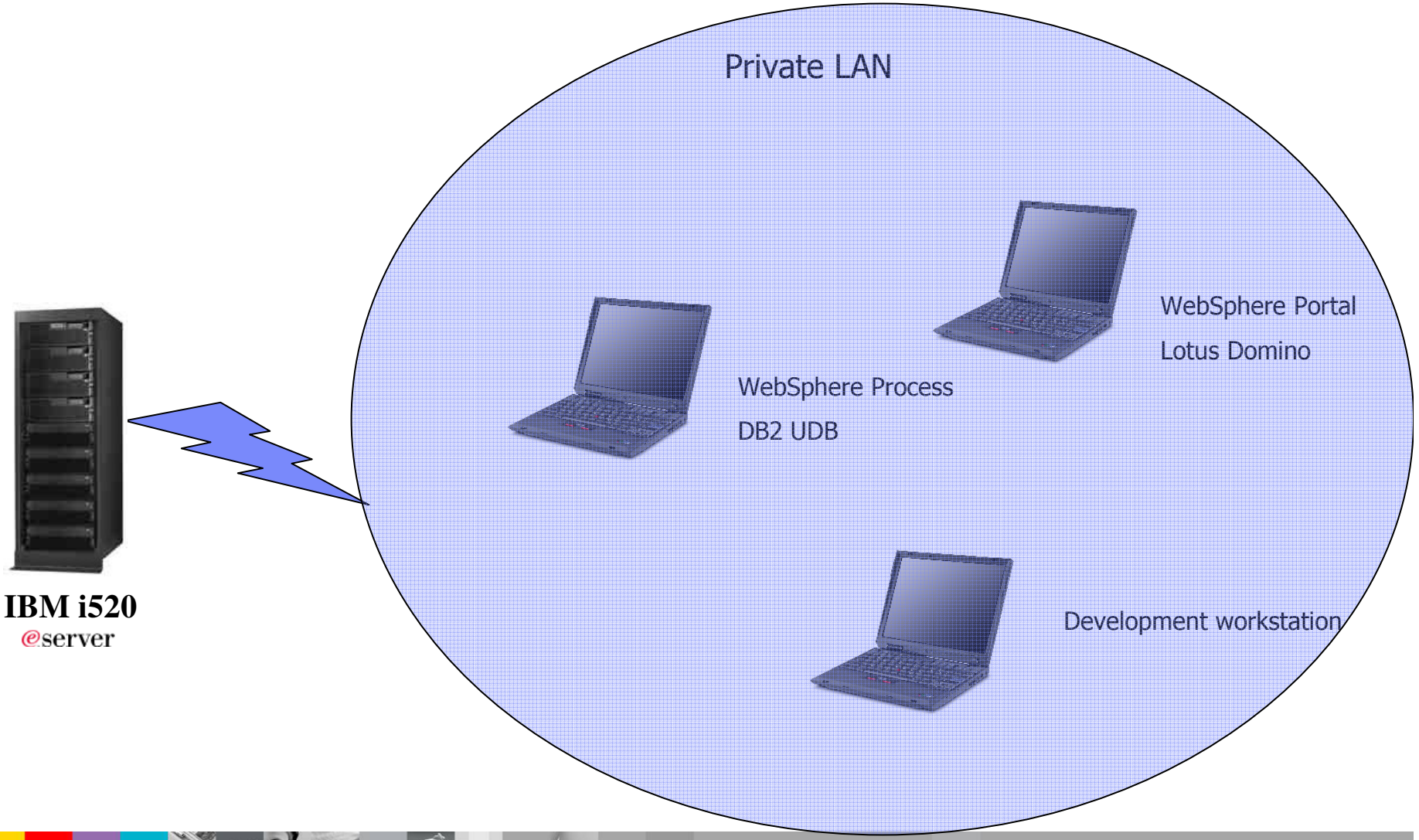
Order Management: Development environment

- To support the development of this composite application, a single development workstation has been equipped with the following IBM products
  - **Rational Application Developer**
    - Java, J2EE, Portlet and Web Services
  - **WebSphere Integration Developer**
    - Business Process, Mediation, Adapters, data transformation
  - **Host Access Transformation Services**
    - iSeries 5250 integration components
  - **WebSphere Development Client Studio for iSeries**
    - RPG programs integration components
  - **WebSphere Portlet Factory and Dashboard Framework**
    - Portlets and dashboards
- All products are based on **Rational Development Platform** (Eclipse) to share the same user interface (shell sharing)

Order Management: technological components



Demo: Operating environment



Demo: Process users

User	Role
Roberto Boccadoro	Sales representative
Federico Senese	Financial Officer
Giancarlo Giannini	Warehouse Officer

Demo: Test cases

Case	Parameters	Financial Check	Warehouse Check	Result
#1	• Customer ID: 10001 • Item ID: IT_001 • Quantity: 15	OK	OK	Automatic completion
#2	• Customer ID: 10002 • Item ID : IT_002 • Quantity : 10	Not OK	Not OK	Automatic refusal email sent to the submitter
#3	• Customer ID: 10003 • Item ID : IT_003 • Quantity : 20	Human decision	Human decision	? email sent to the submitter in case of refusal

Questions