Initial Administration Training

UI and Architecture

12 February 2018

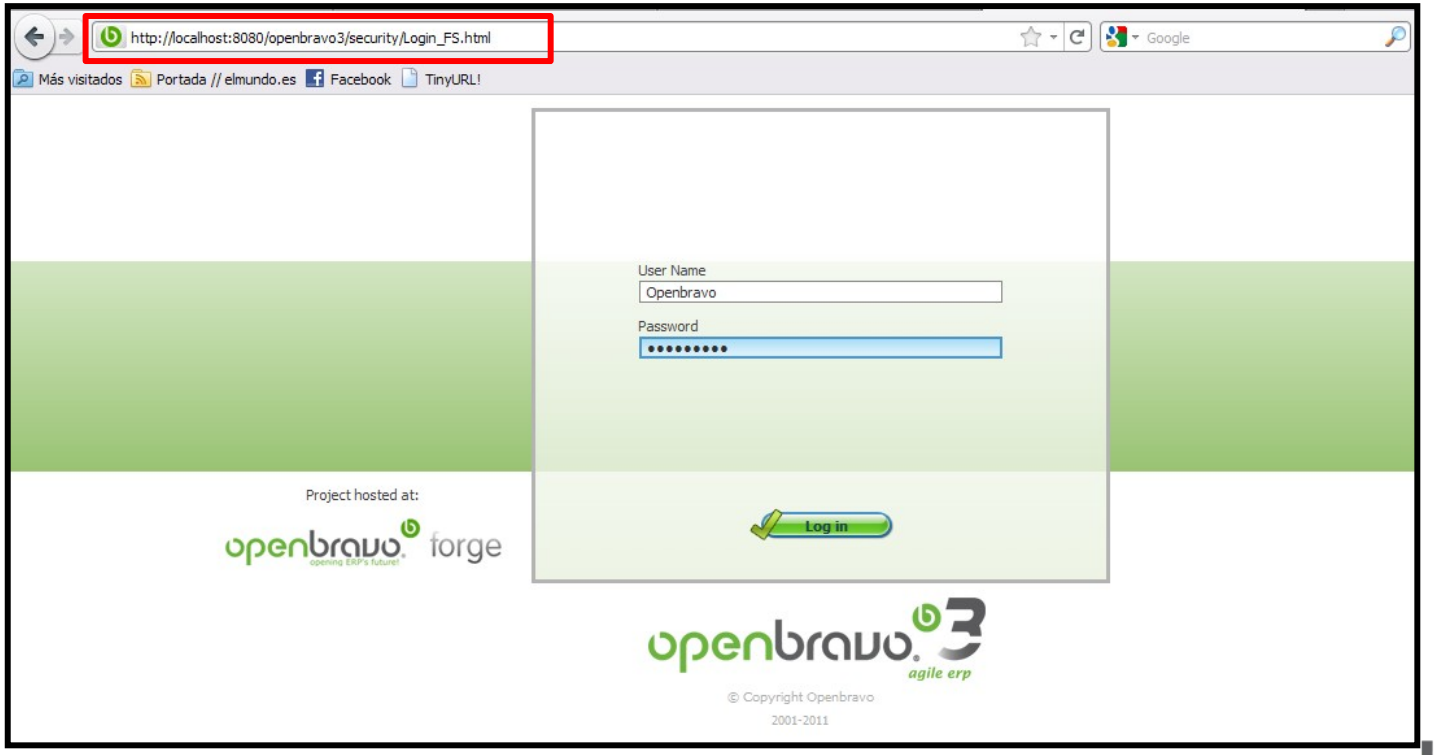
User interface

Architecture

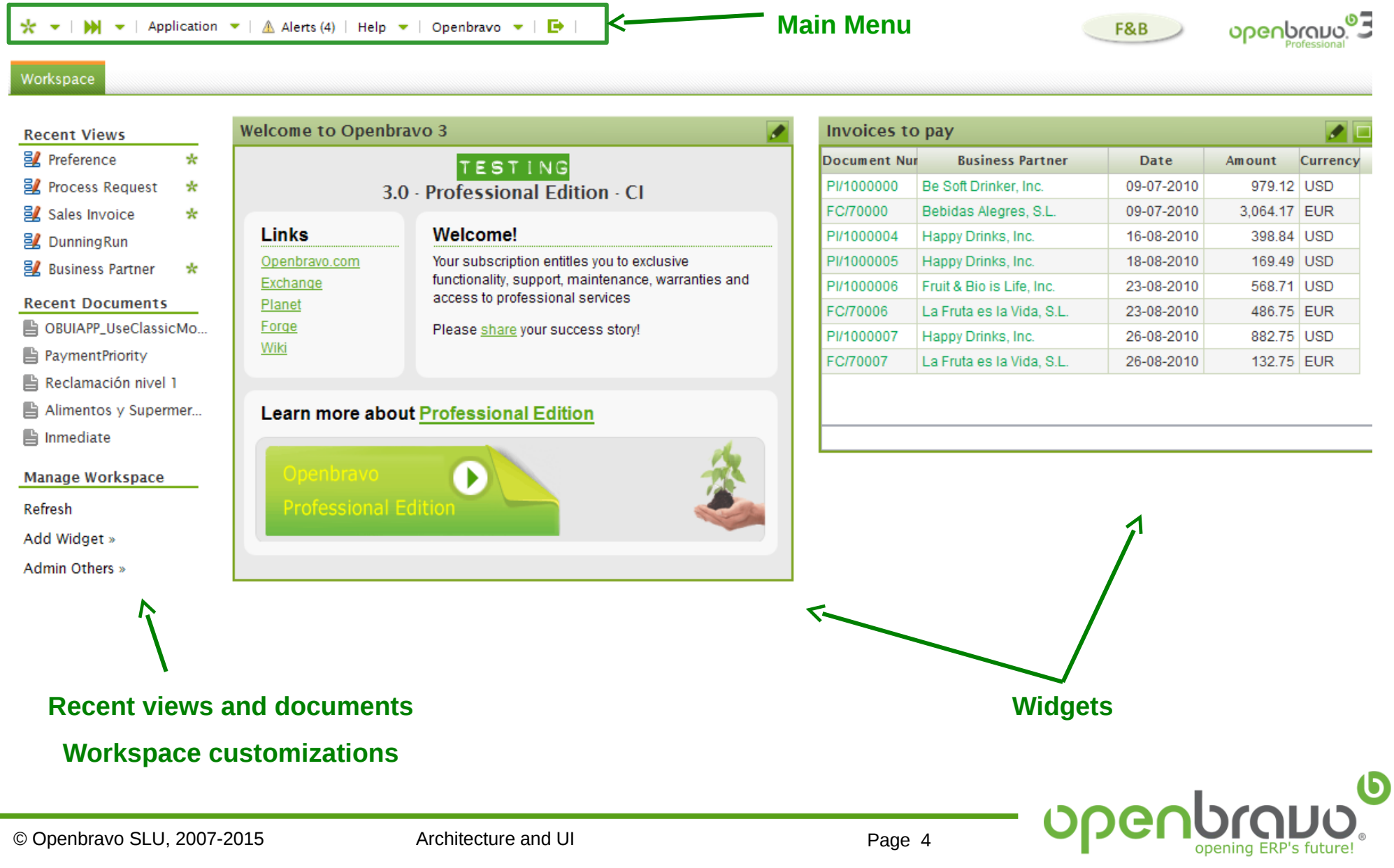
User interface
Web-based application

Openbravo has a native **web-based architecture**, which enables a universal access to the application from each device connected to the network. OS-independent browser enables full-function use to support modern, flexible and secure (SSL) usage scenarios like work-from-home, remote contractors, customer access to information, etc.

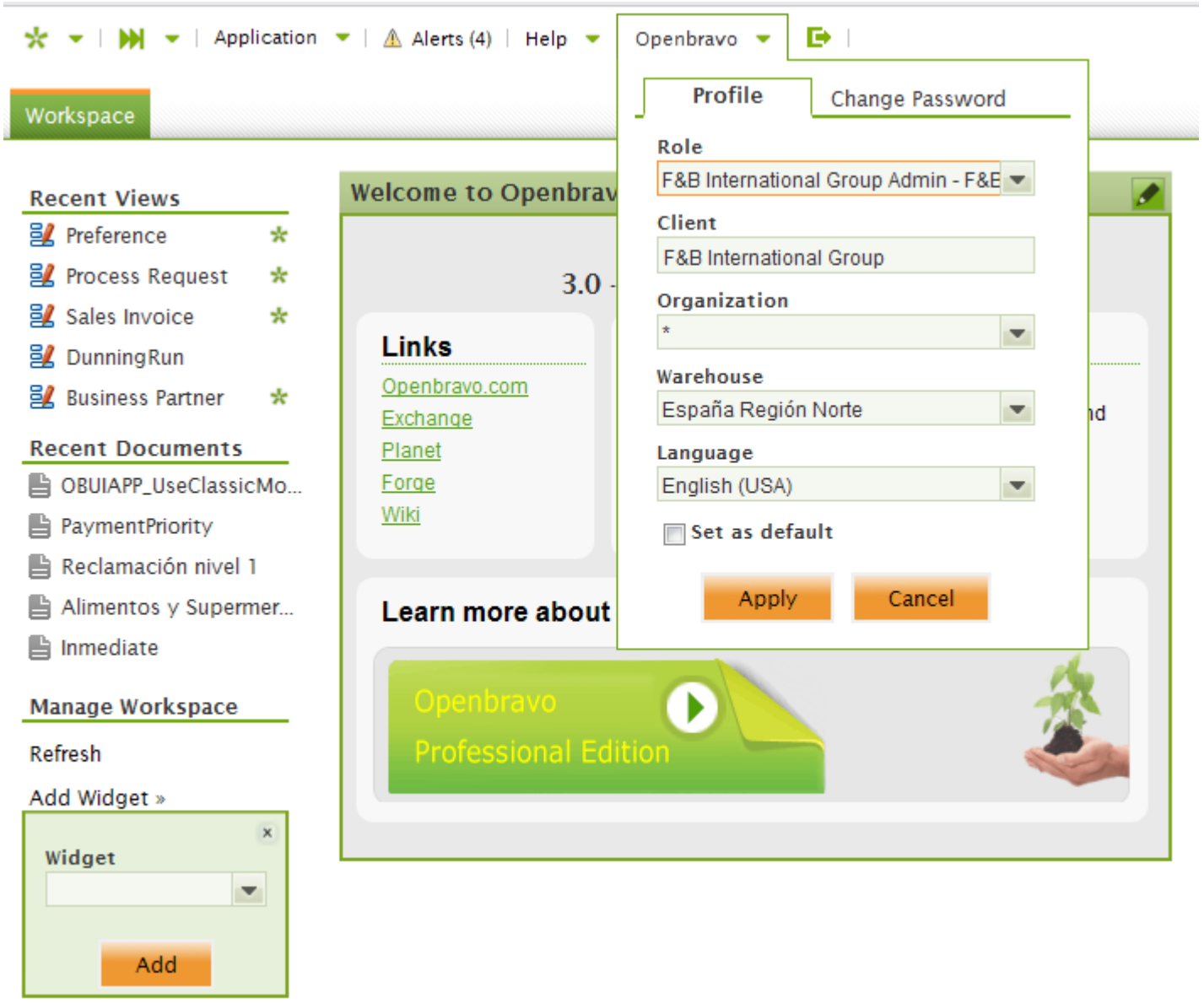
Recommended browsers are: ***Chrome 9, Firefox 4 and Internet Explorer 9.0***



User interface
My Openbravo



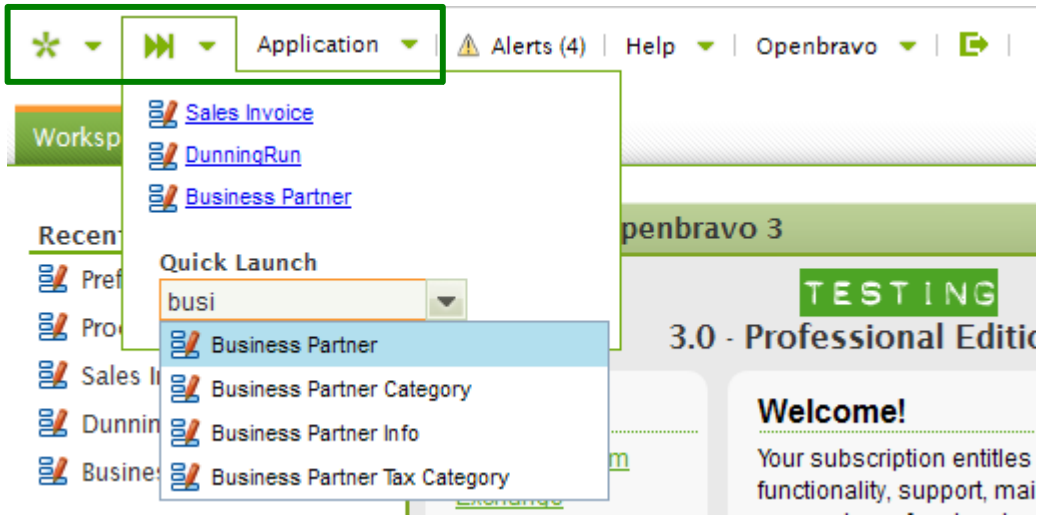
User interface
Users and roles

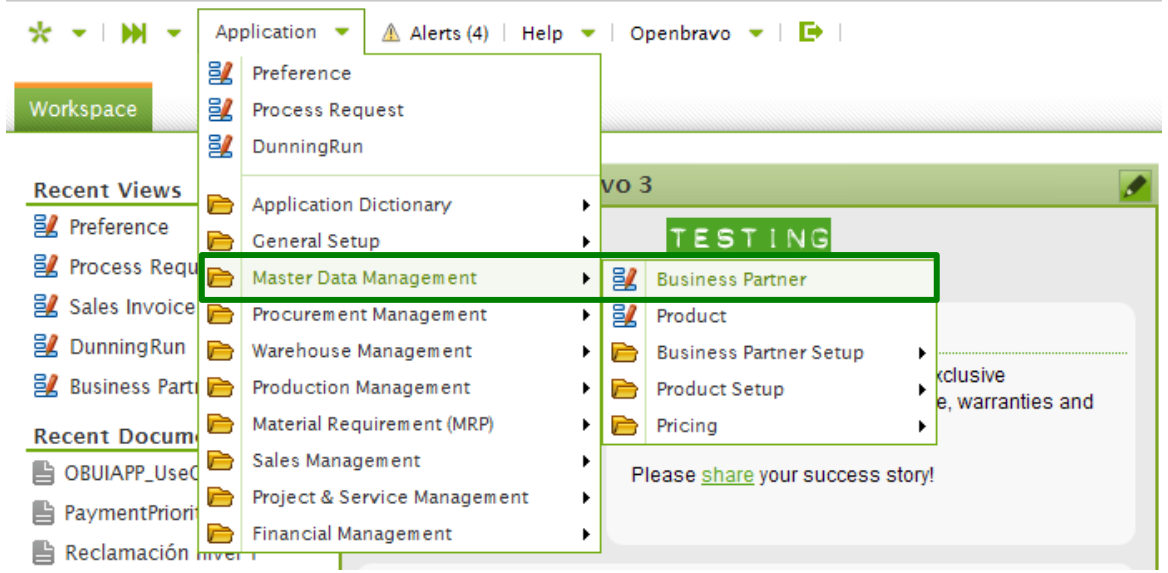


Menu and access privileges depend on the role with we are logged-in in the application.

In Openbravo ERP, each user can have more than an associated role.

User interface
Navigation – application menu

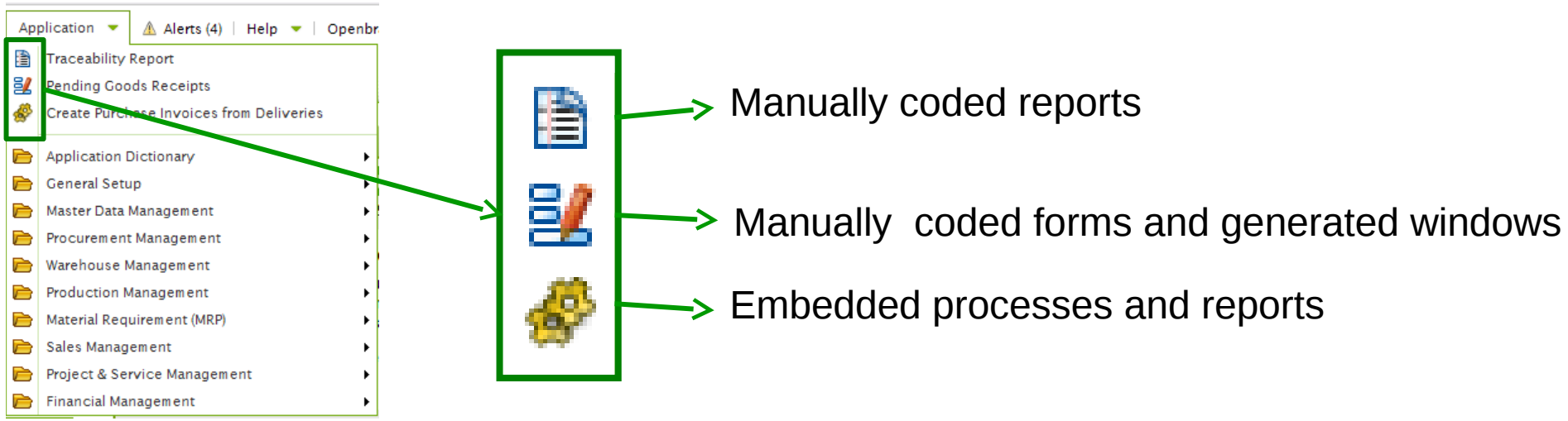




Access to processes, data and reports of the ERP can be done by using

- Create New
- Quick Launch
- Application menu

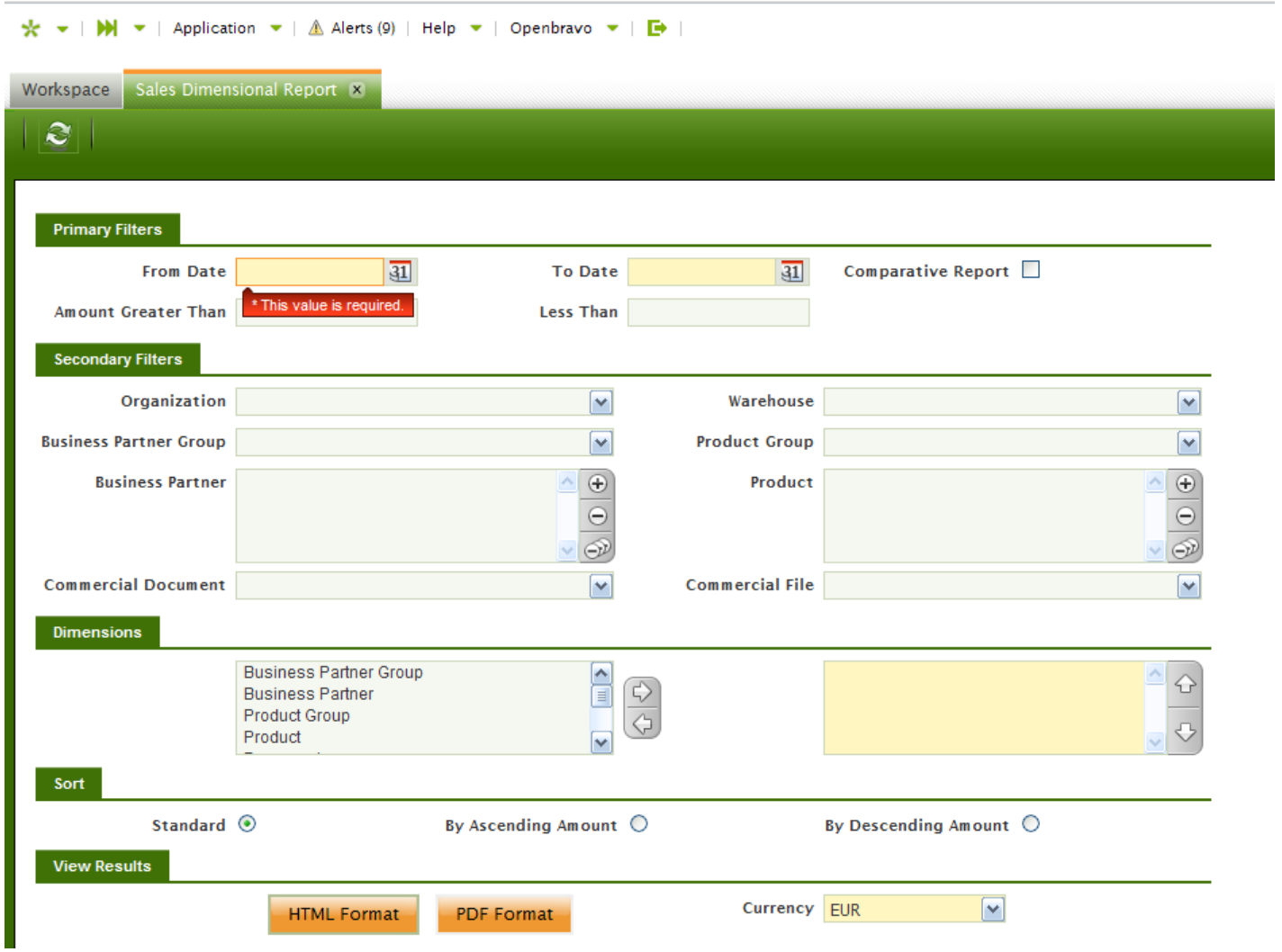
User interface
Type of menu items



3.0

1. **Generated windows**: automatically generated windows from metadata
2. **Embedded processes**: execution of processes with view and controller automatically generated
3. **Embedded reports**: execution of reports with filters automatically generated by the system engine

1. Manually coded reports

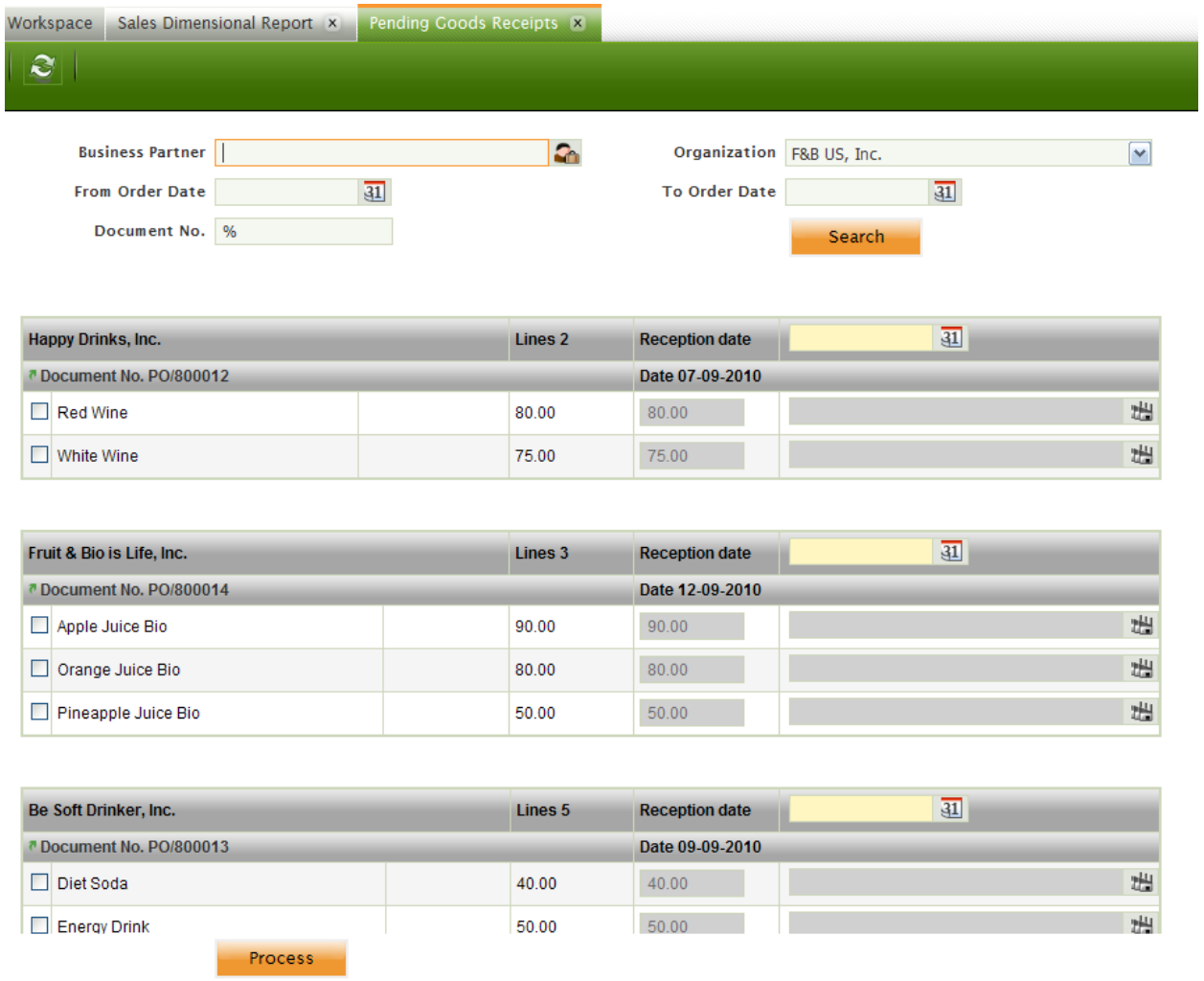


Most elements under folders ***Analysis Tools***.

Multi-choice selectors and radio buttons are available in this type of items.

100% manually coded with previous 3.0 platform

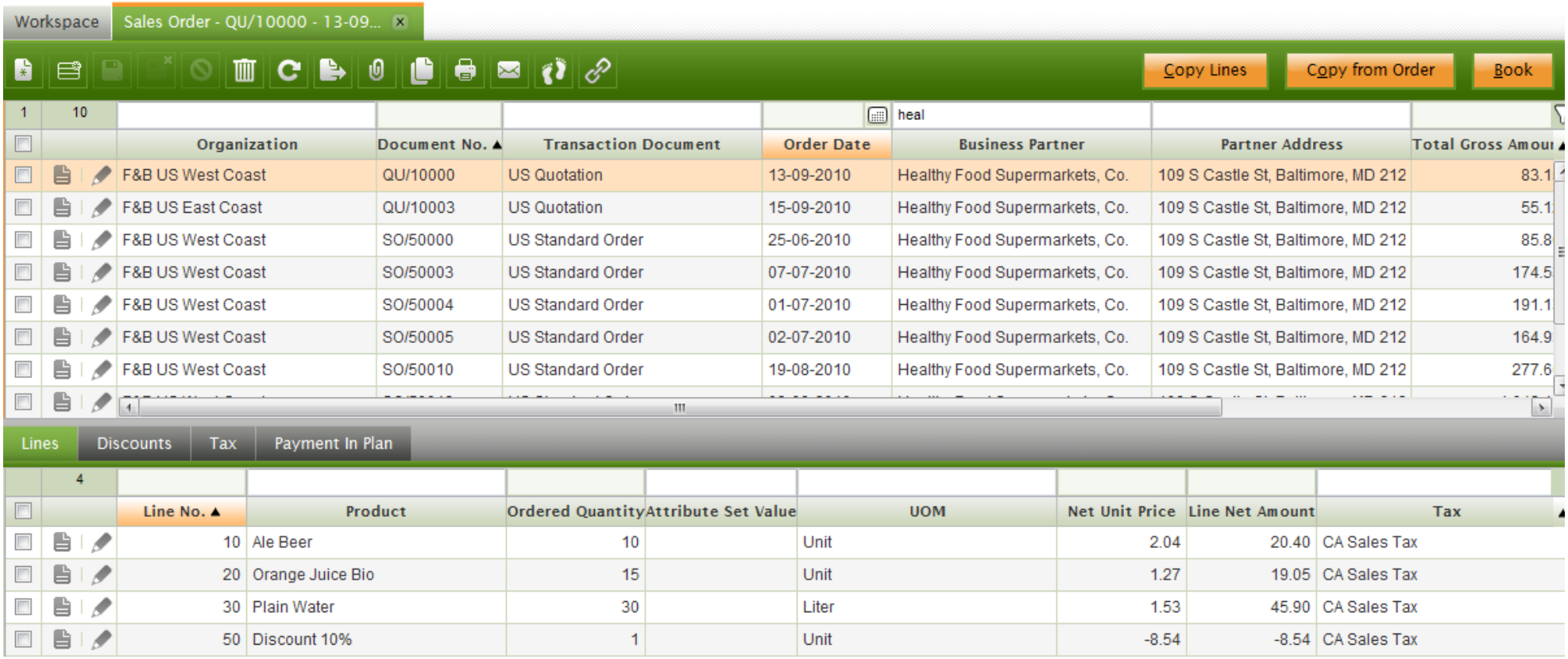
2. Manually coded forms



Rapid entry forms to process batches of data.

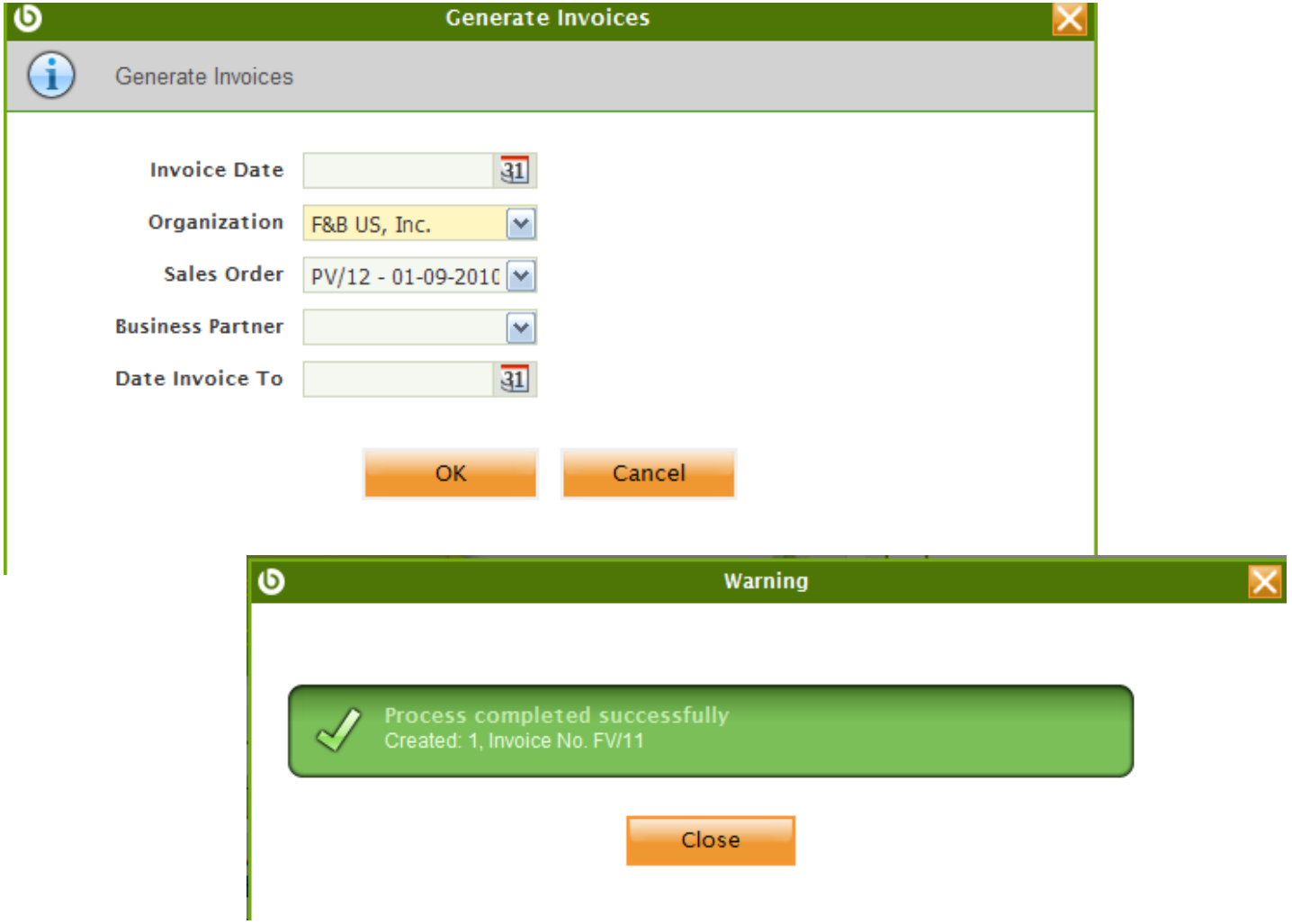
100% manually coded with previous 3.0 platform

3. Generated windows



Generated by the system based on metadata. Most of the windows of the application follows this pattern. A table or a view is always behind this type of elements. Little manual coding is required.

3. Embedded processes

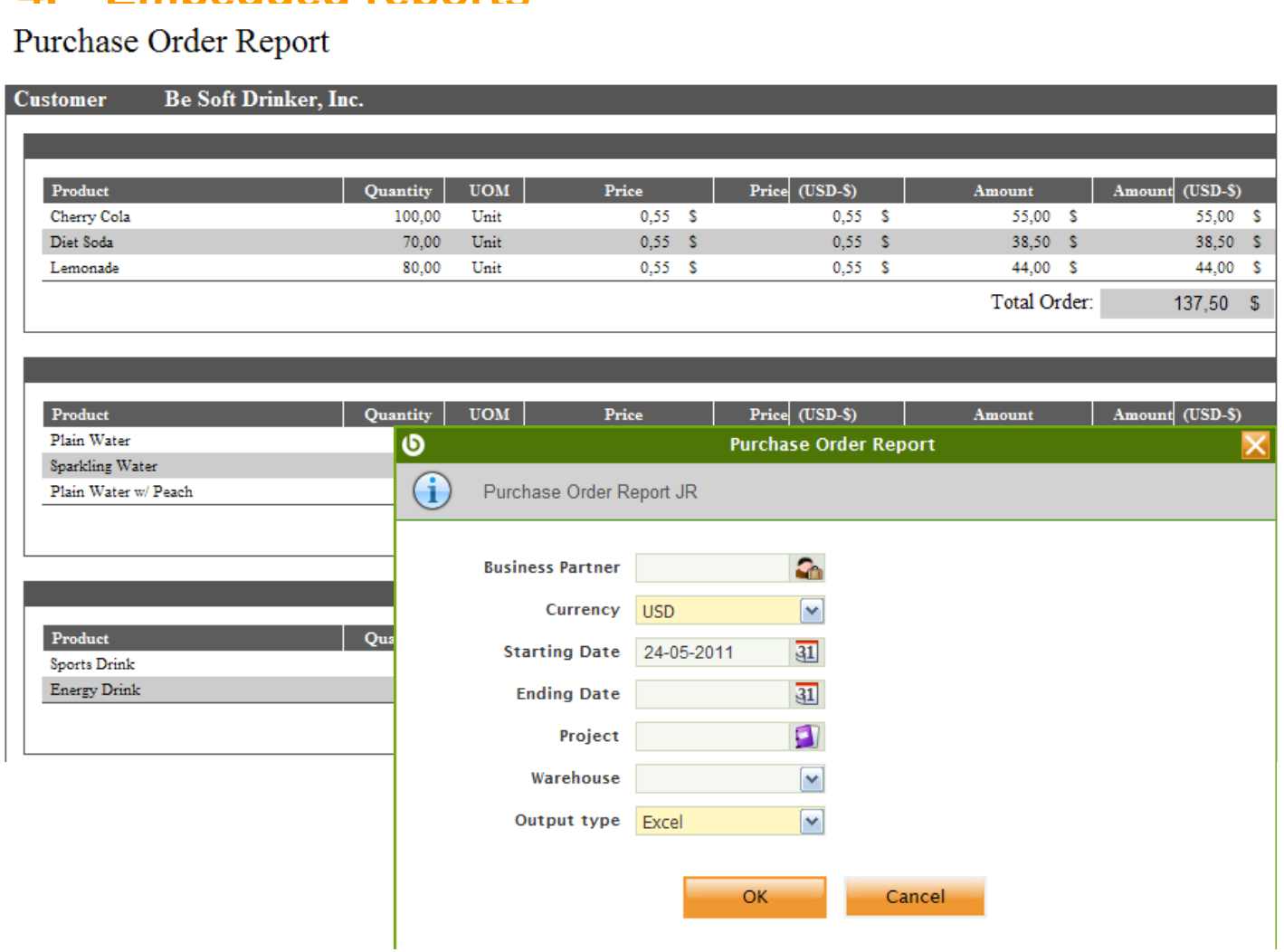


Execution of java processes or database store procedures.

Input parameters can be automatically configured y using metadata

50% of coding required

4. Embedded reports

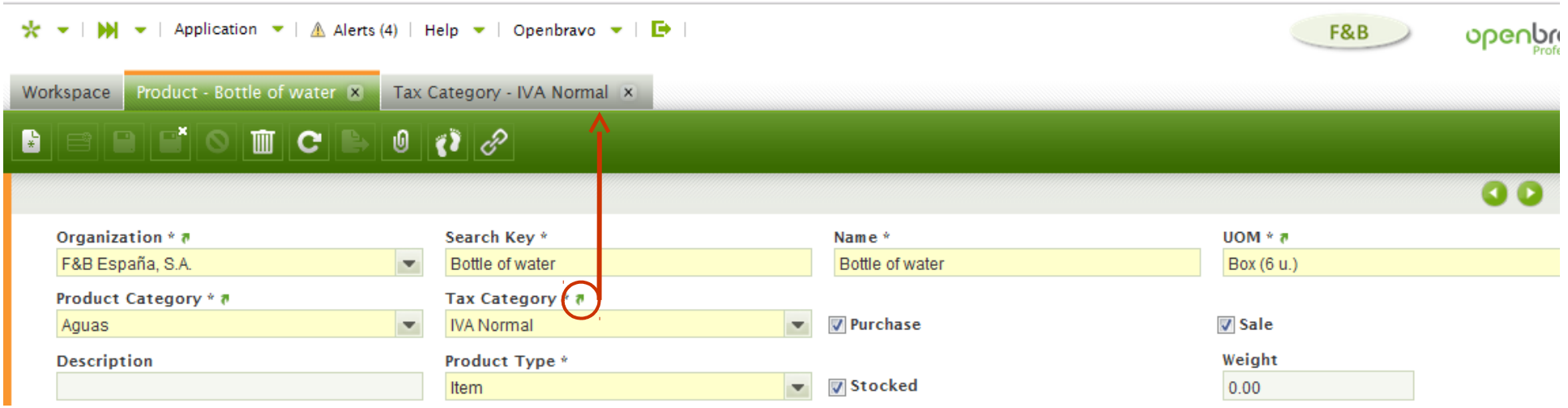


Rendering of jasper reports.

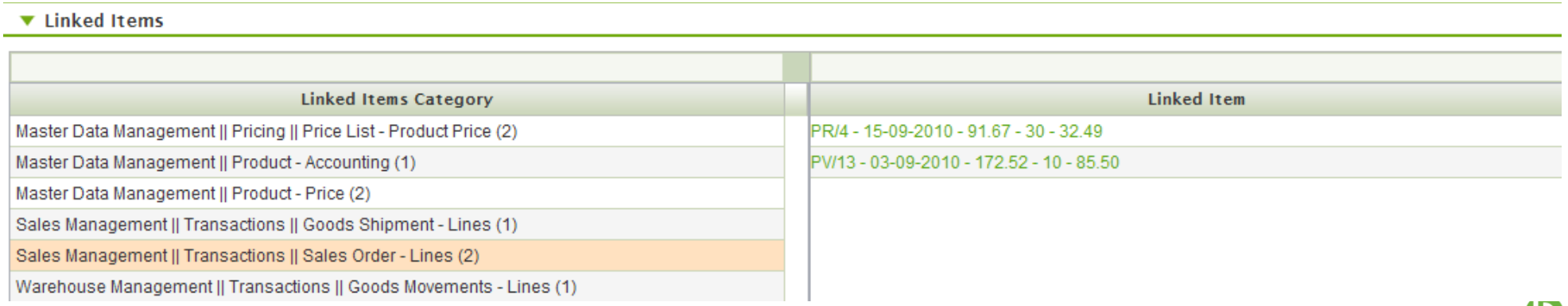
Input parameters and output format can be automatically configured y using metadata

50% of coding required

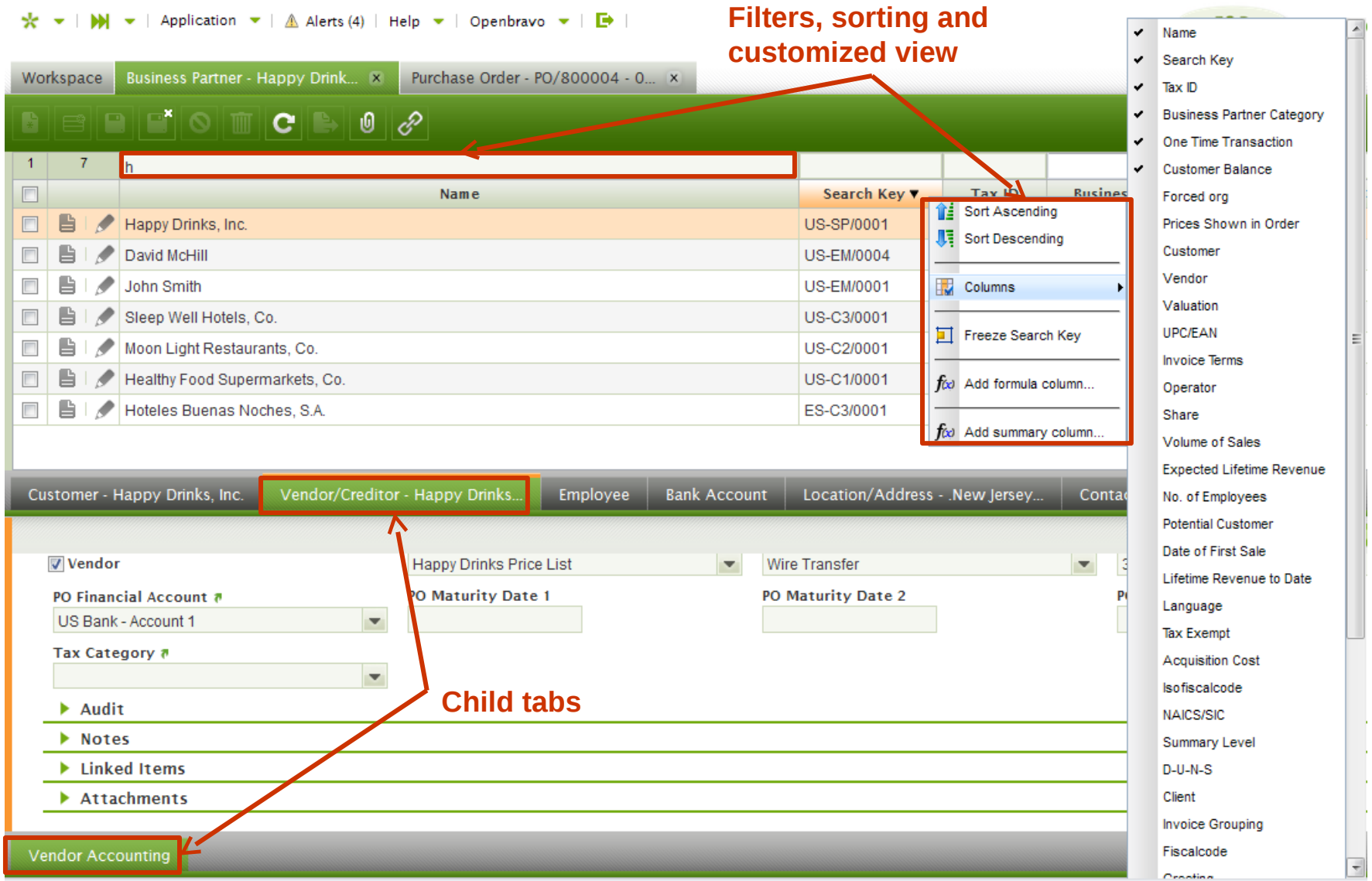
Related records and multi-tab



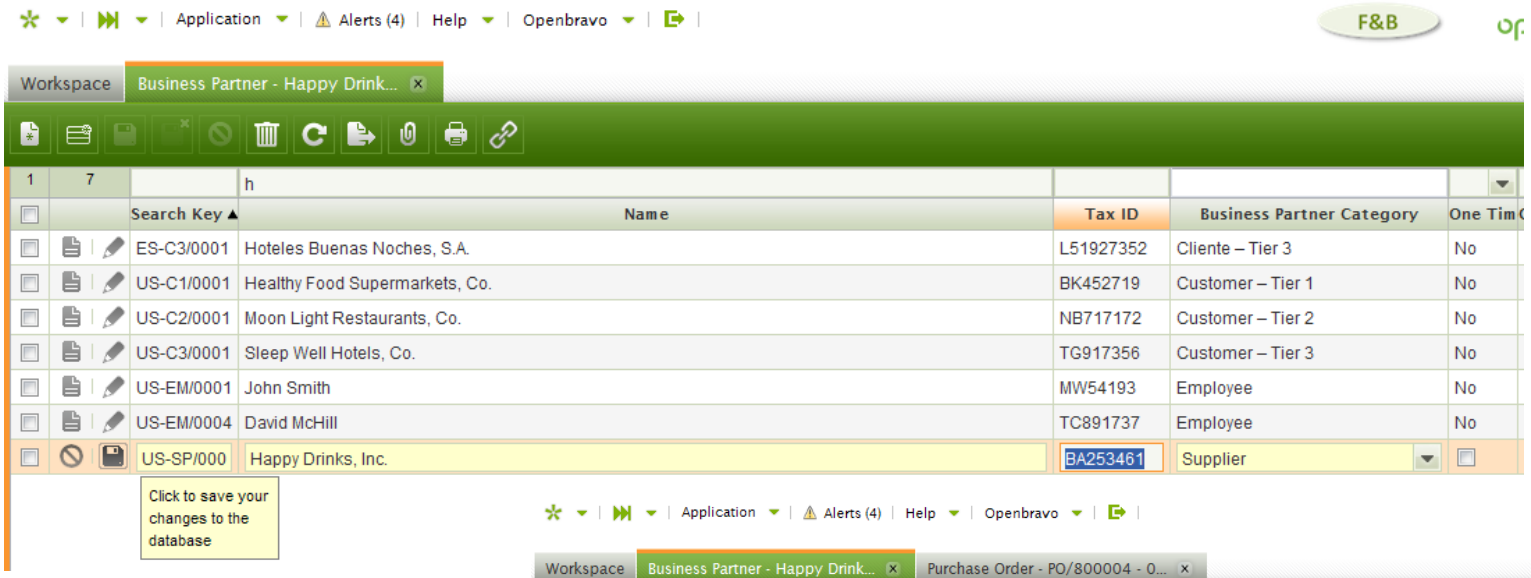
Linked items



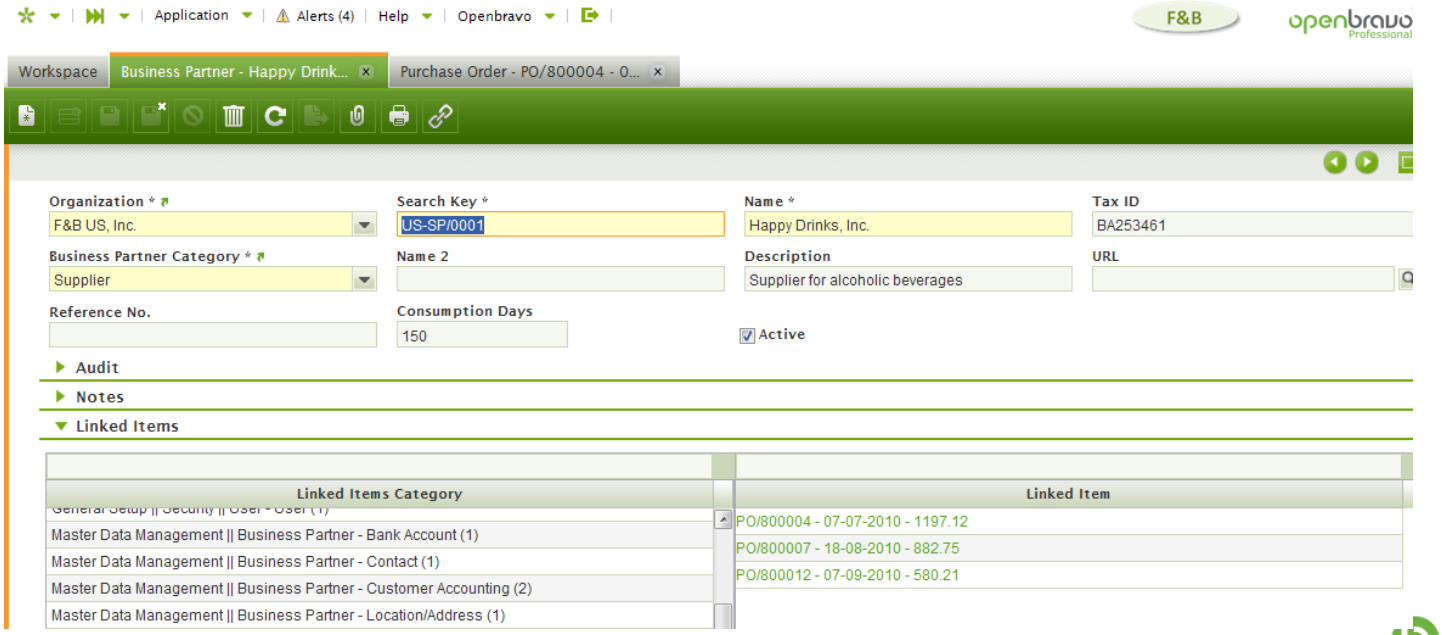
User interface
Window layout – header and tabs



User interface
Window layout – view modes



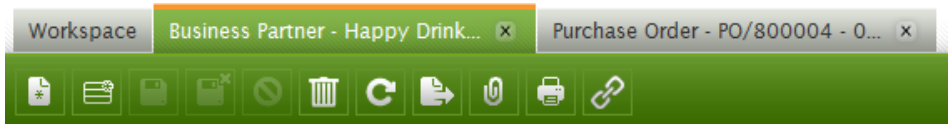
Grid view



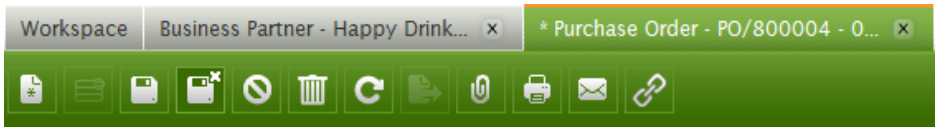
Form view

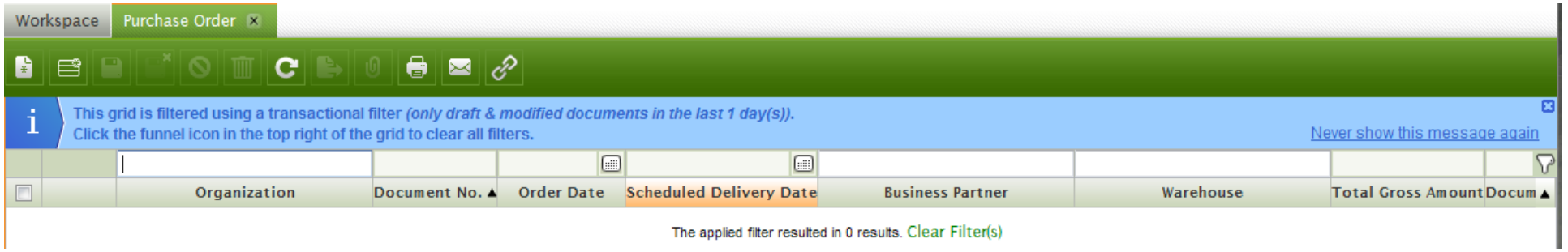
Depending on the view mode, icons are enabled/disabled in the toolbar

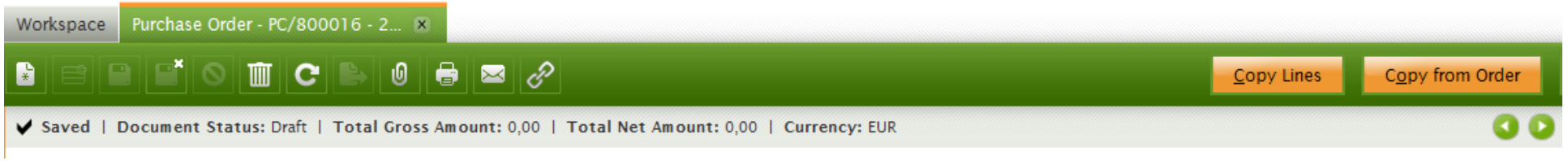
Grid view

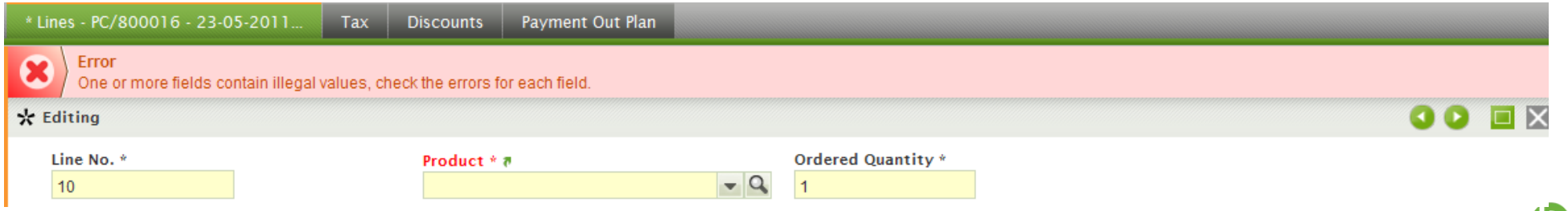


Form view



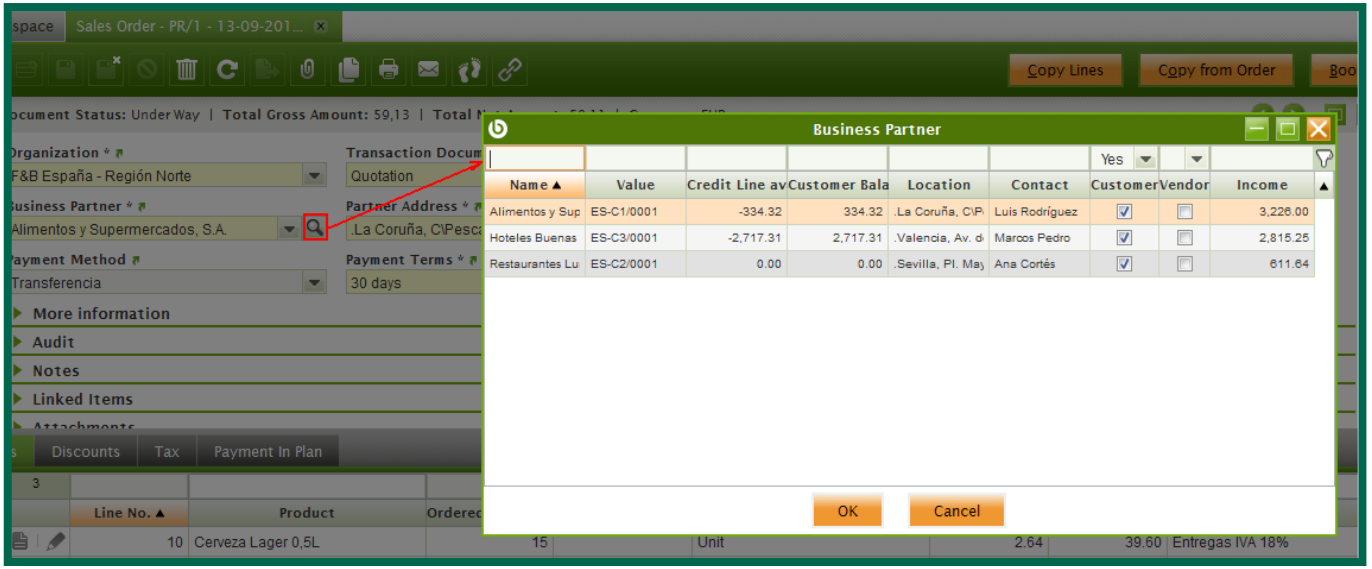






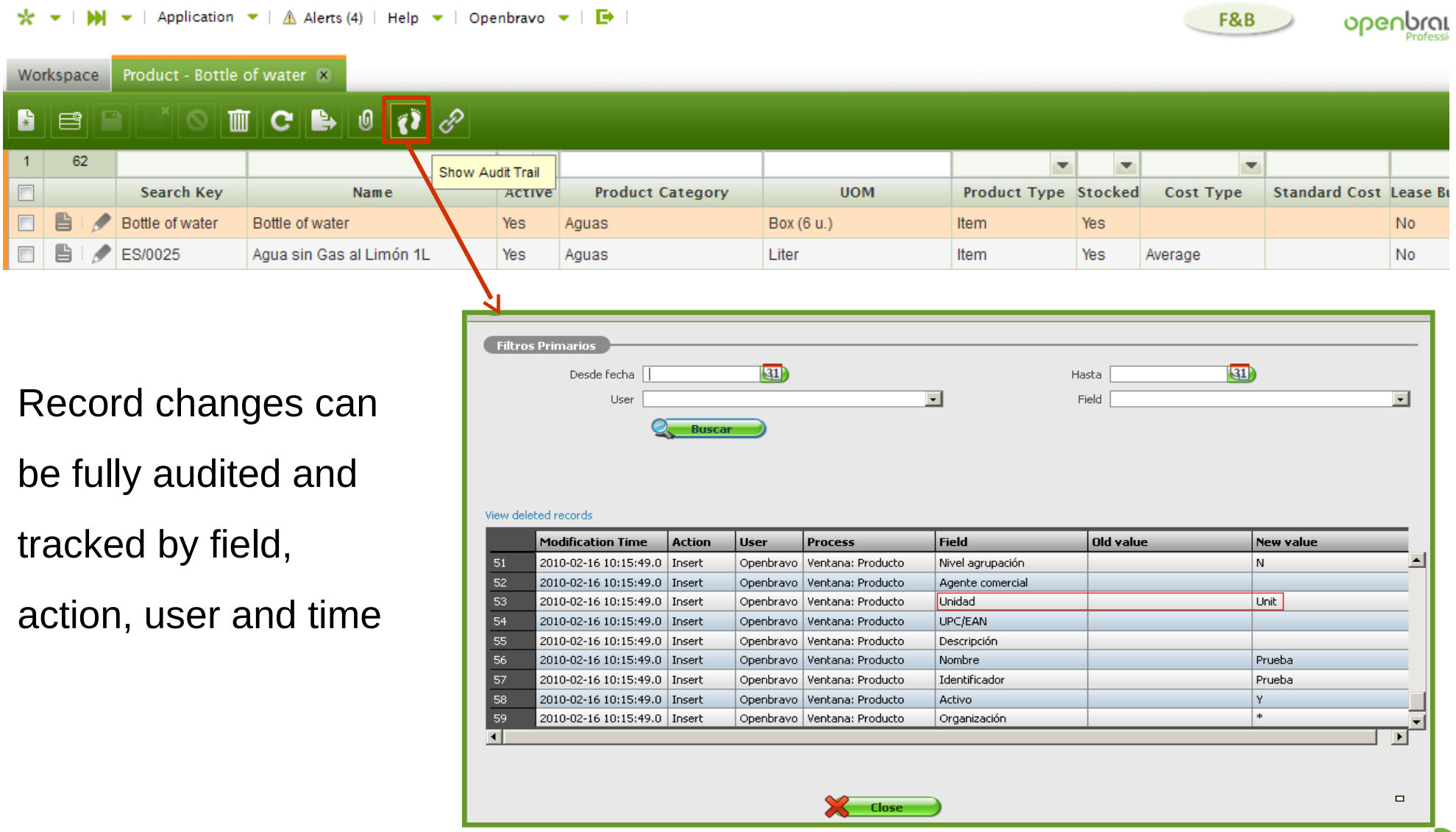
Selectors combine a suggestion box with a popup window with filter and sorting capabilities.

- **Definition** (columns, search criteria, where clause) **can be changed** at runtime without re-compiling or re-starting the system.
- Easy to show **linked information** of a selected entity or search, sort and filter on linked information.
- Suggestion box and popup grid support paging with look ahead, makes them also suitable for **large datasets**



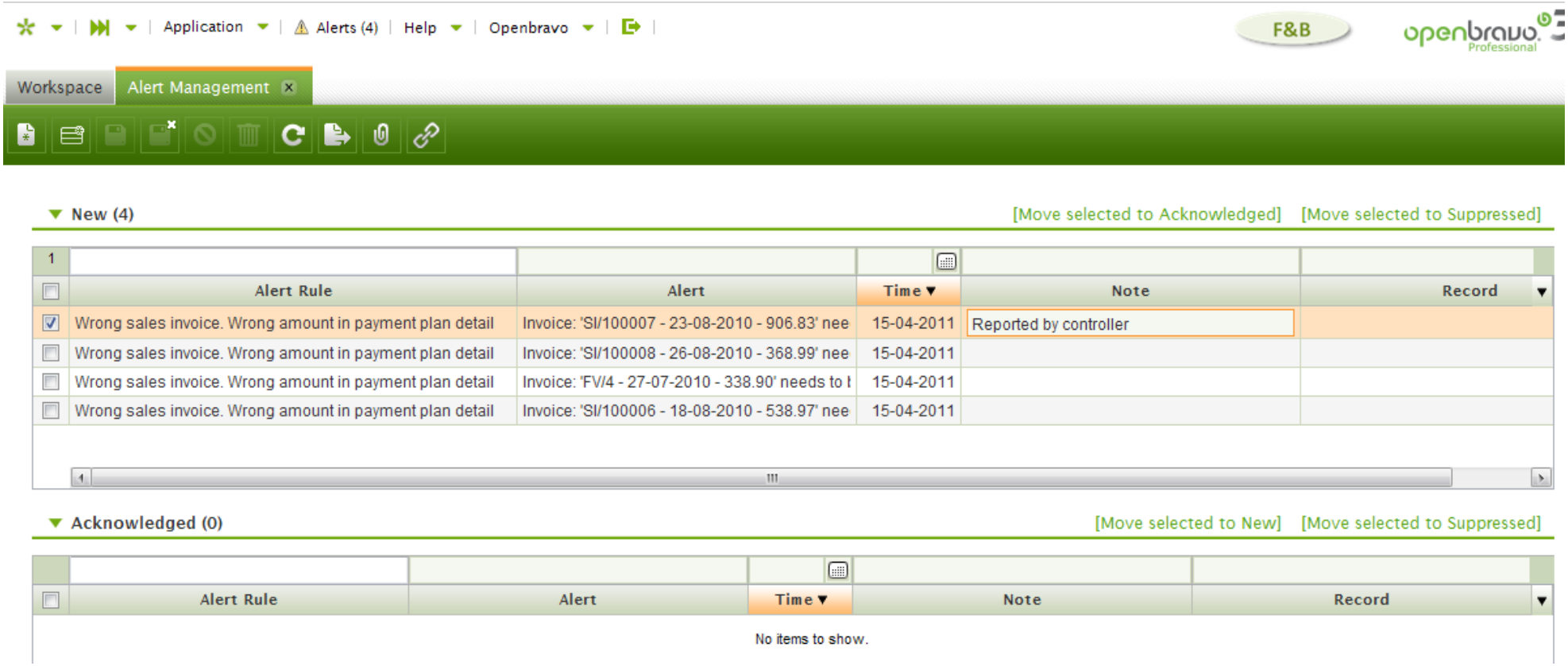
http://wiki.openbravo.com/wiki/Projects/Selector/Developers_Manual

User interface
Fully audited

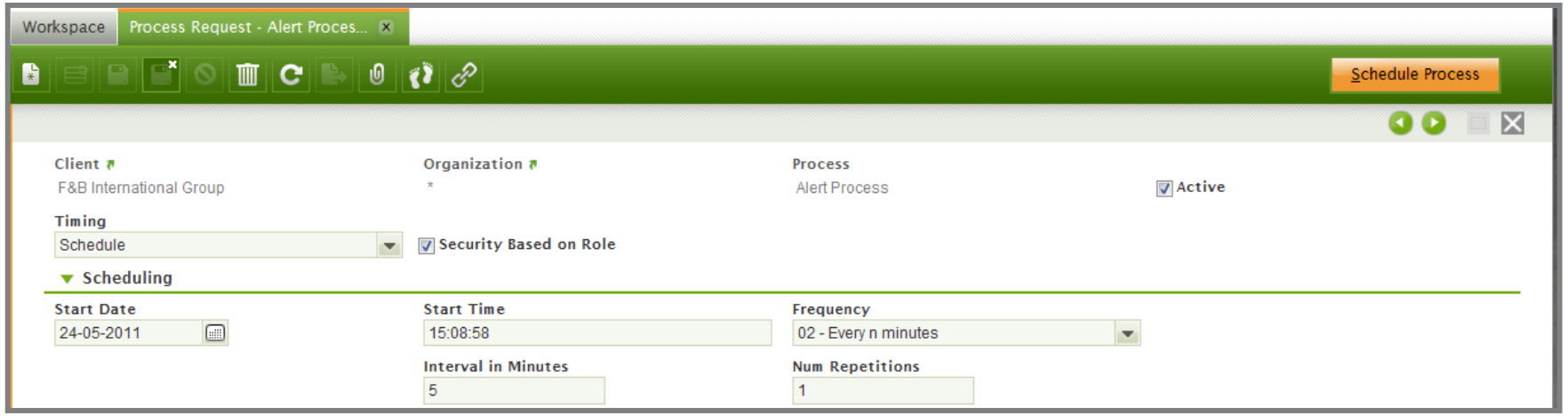


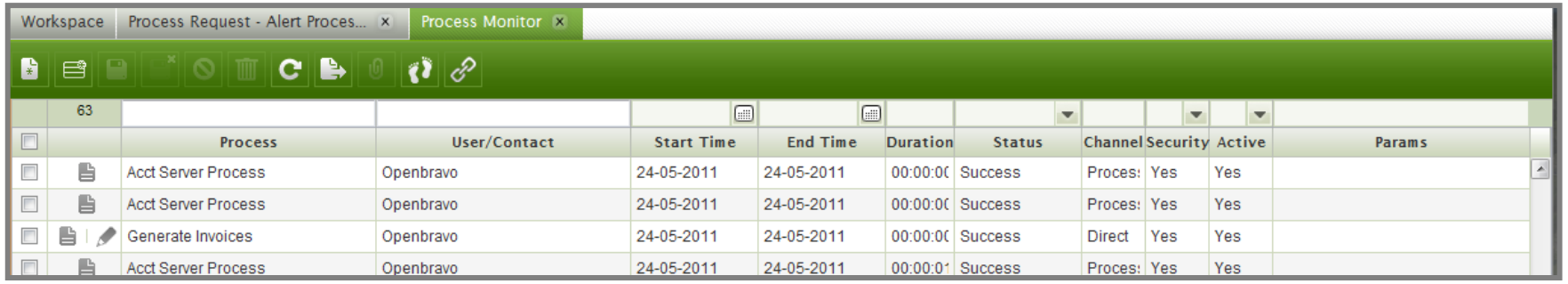
Record changes can be fully audited and tracked by field, action, user and time

Alert platform enables configuration and creation of alerts by role and user

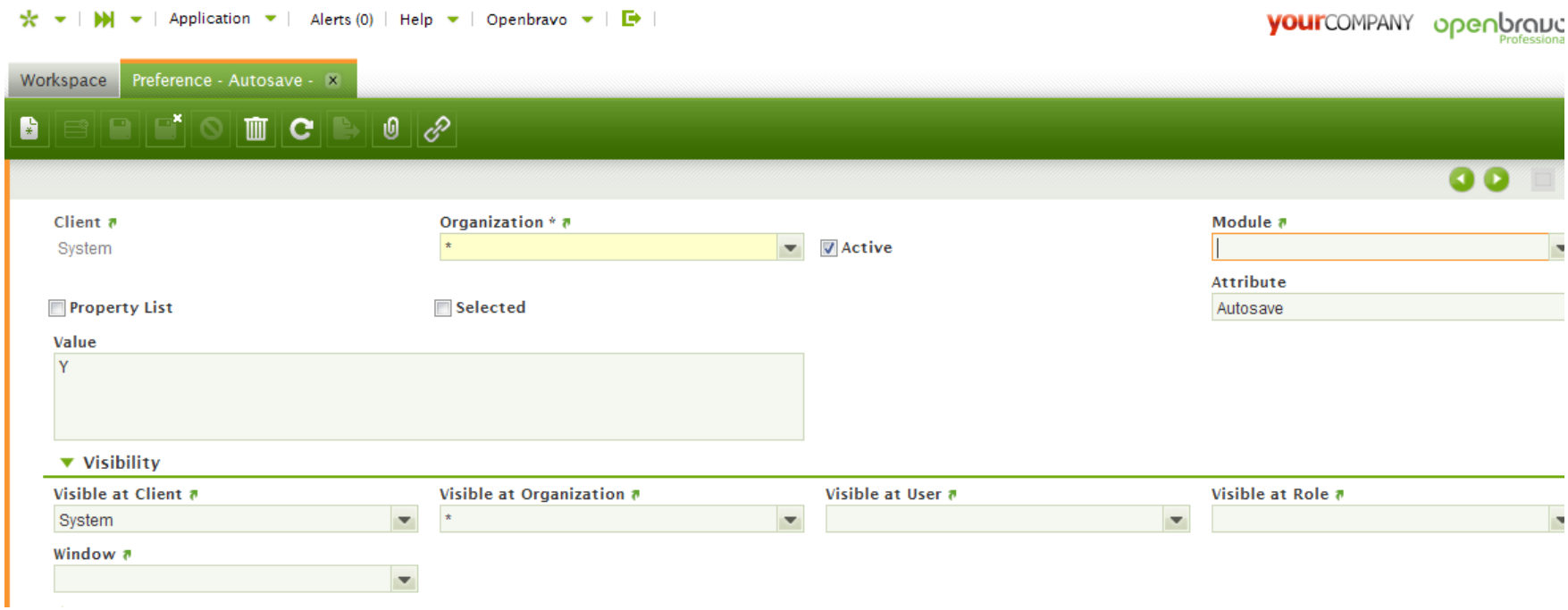


Openbravo ERP enables the scheduling of background processes. Existing processes are ***Alert management, Accounting*** , ***Hearbeat and Payment Monitor.*** Execution of these processes can be monitored.

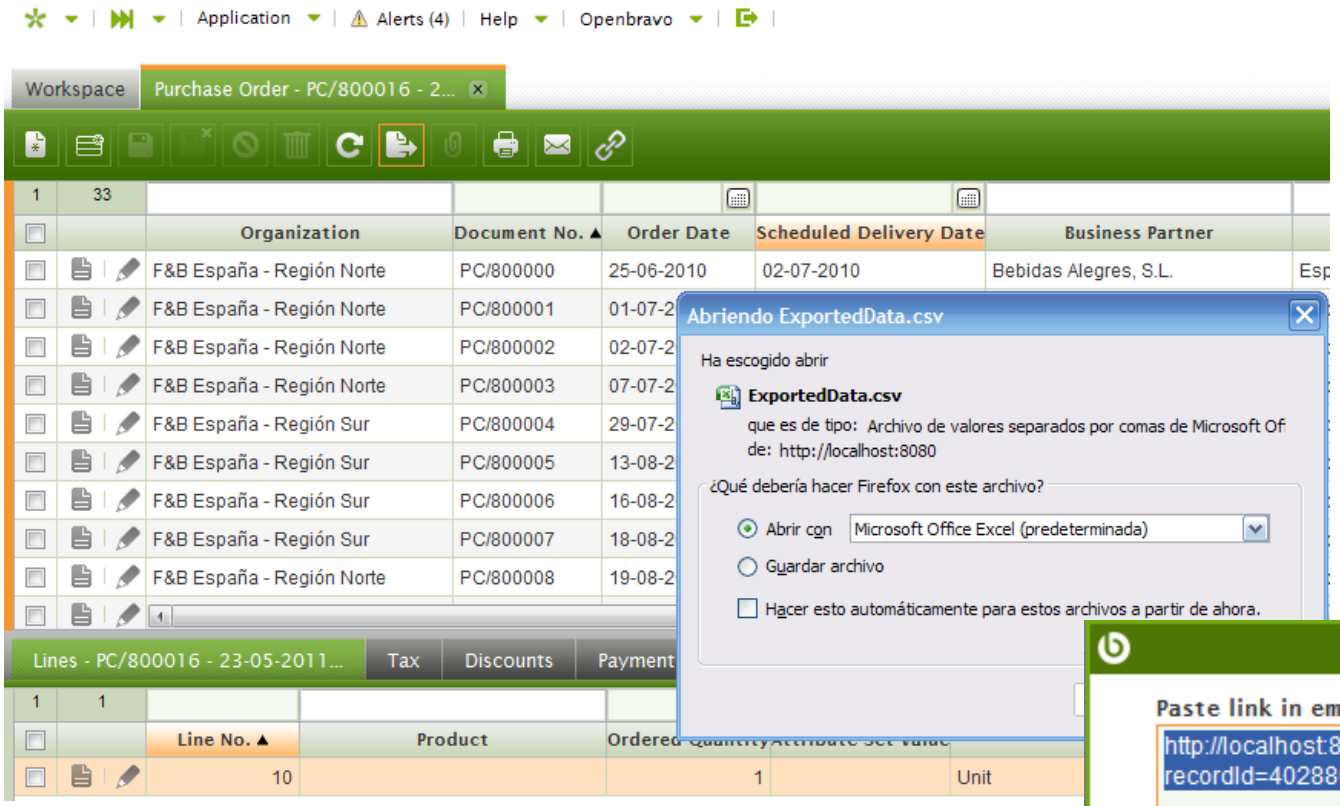


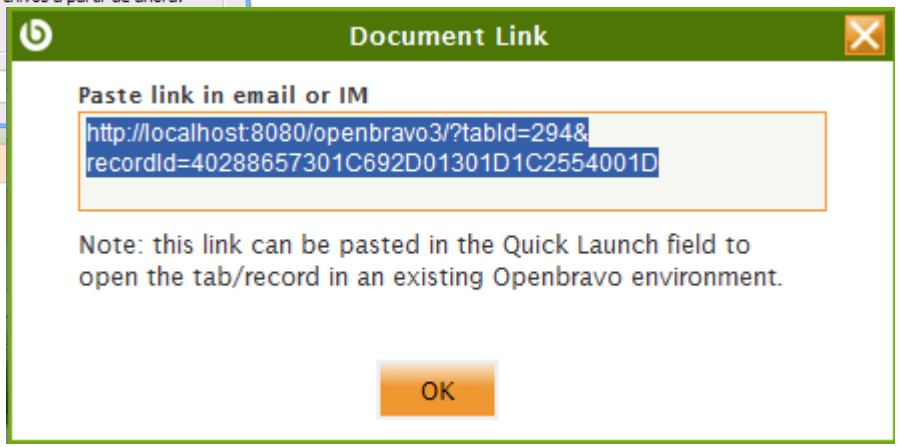


For each application user, log-in preferences as language and organization can be set as preferences, as well as default values for different processes, windows... etc.



User interface
Other interesting features



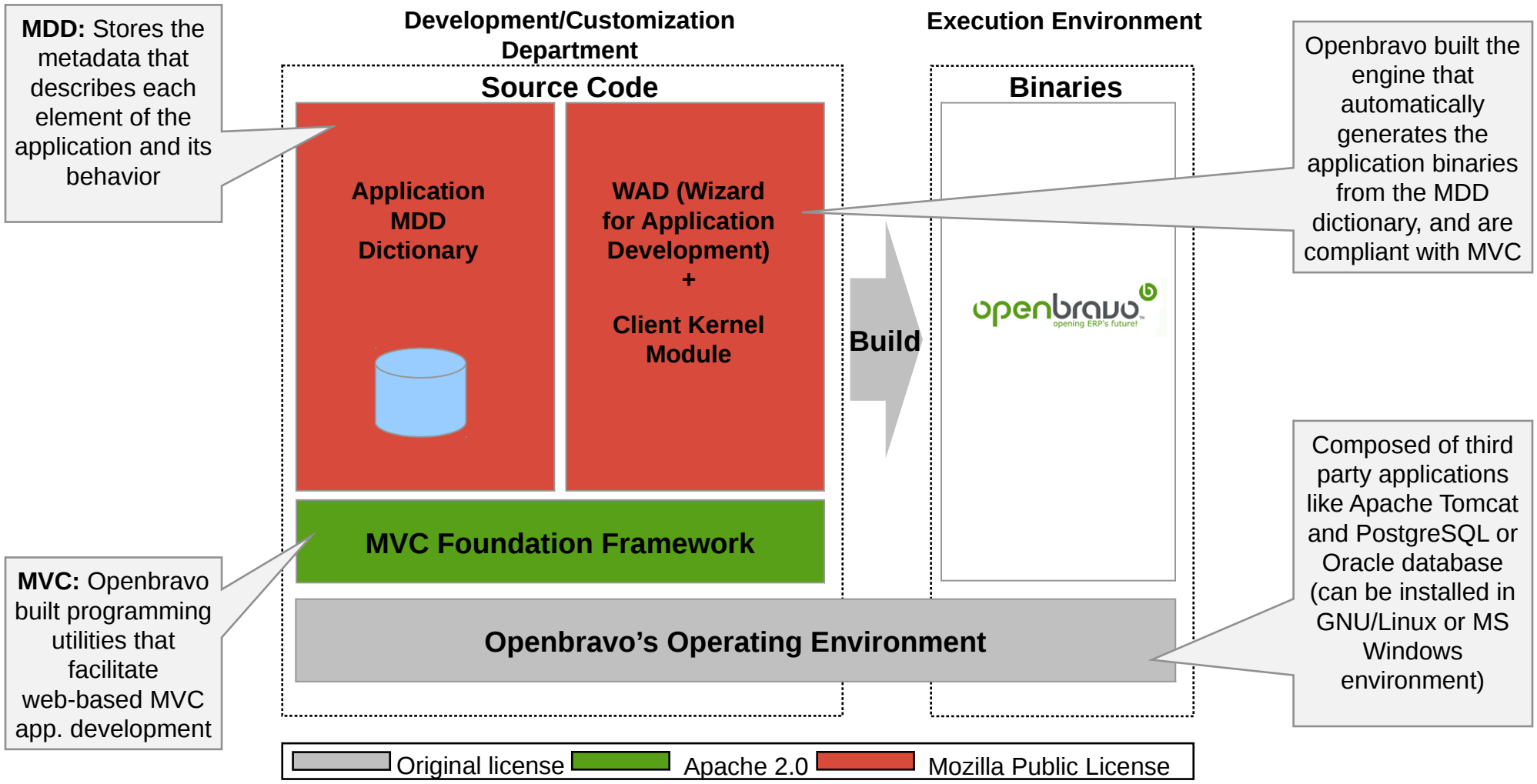


- Export data to CSV file
- Send by e-mail
- Direct link to a particular record

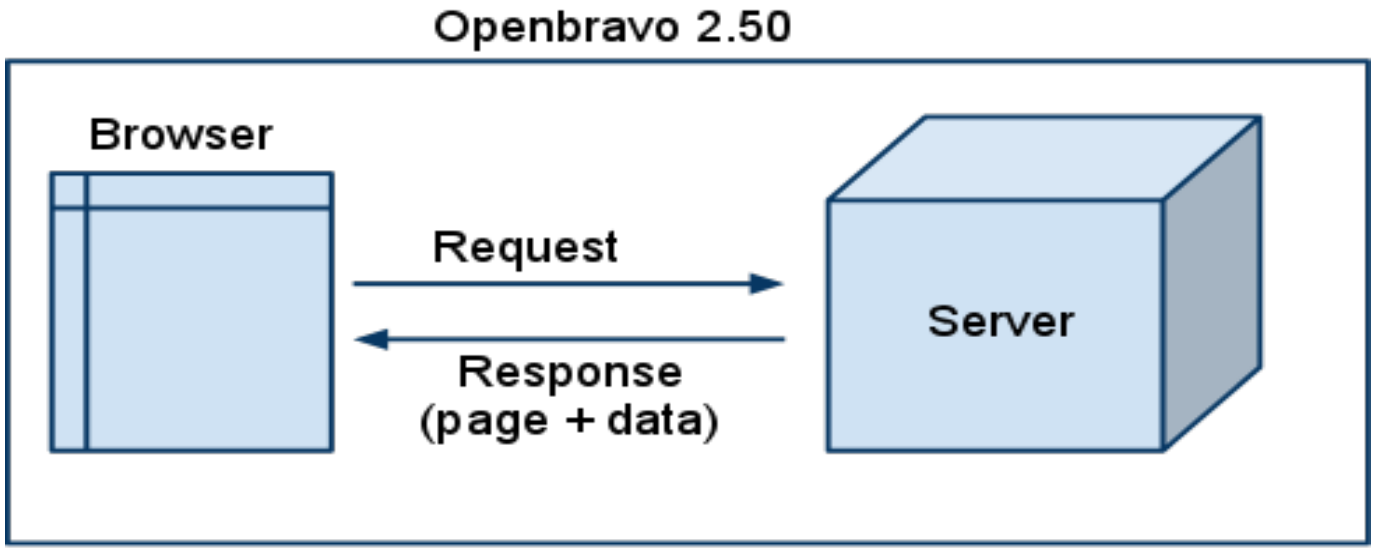
User interface

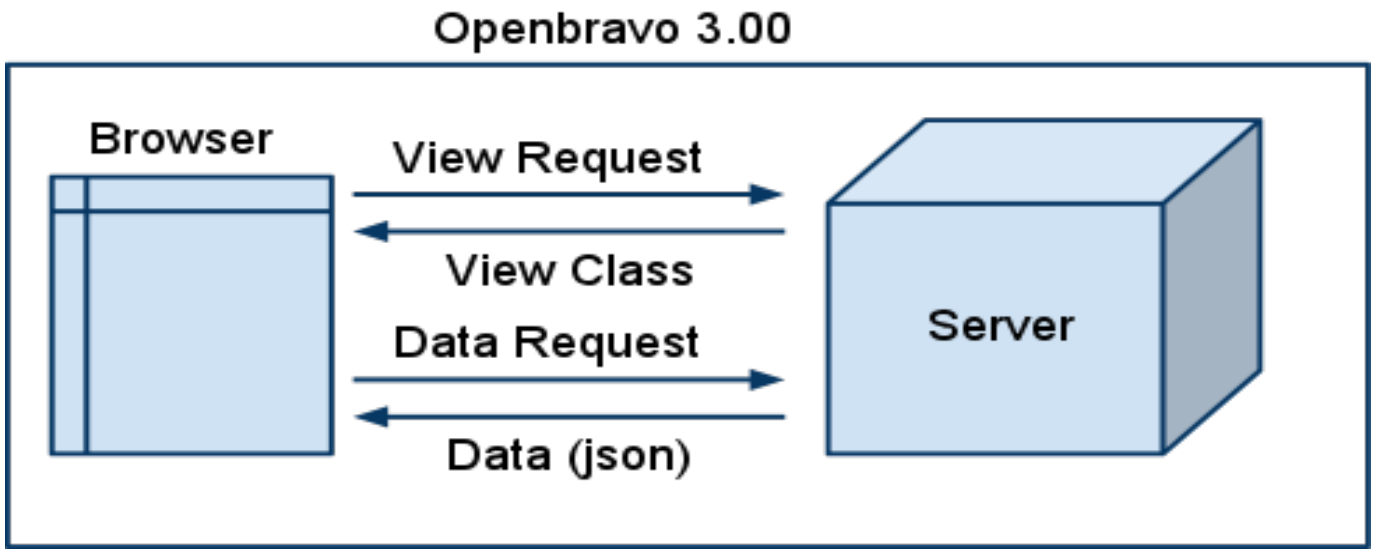
Architecture

Architecture
How Does It Work?



Arquitecture
2.50 vs 3.0





- JavaScript code is generated at runtime
- Increases execution on the client (browser) side
- Data requests are separated from view requests
- Generated JavaScript code extends JS classes statically defined

© Openbravo SLU, 2007-2015

openbravo
opening ERP's future!

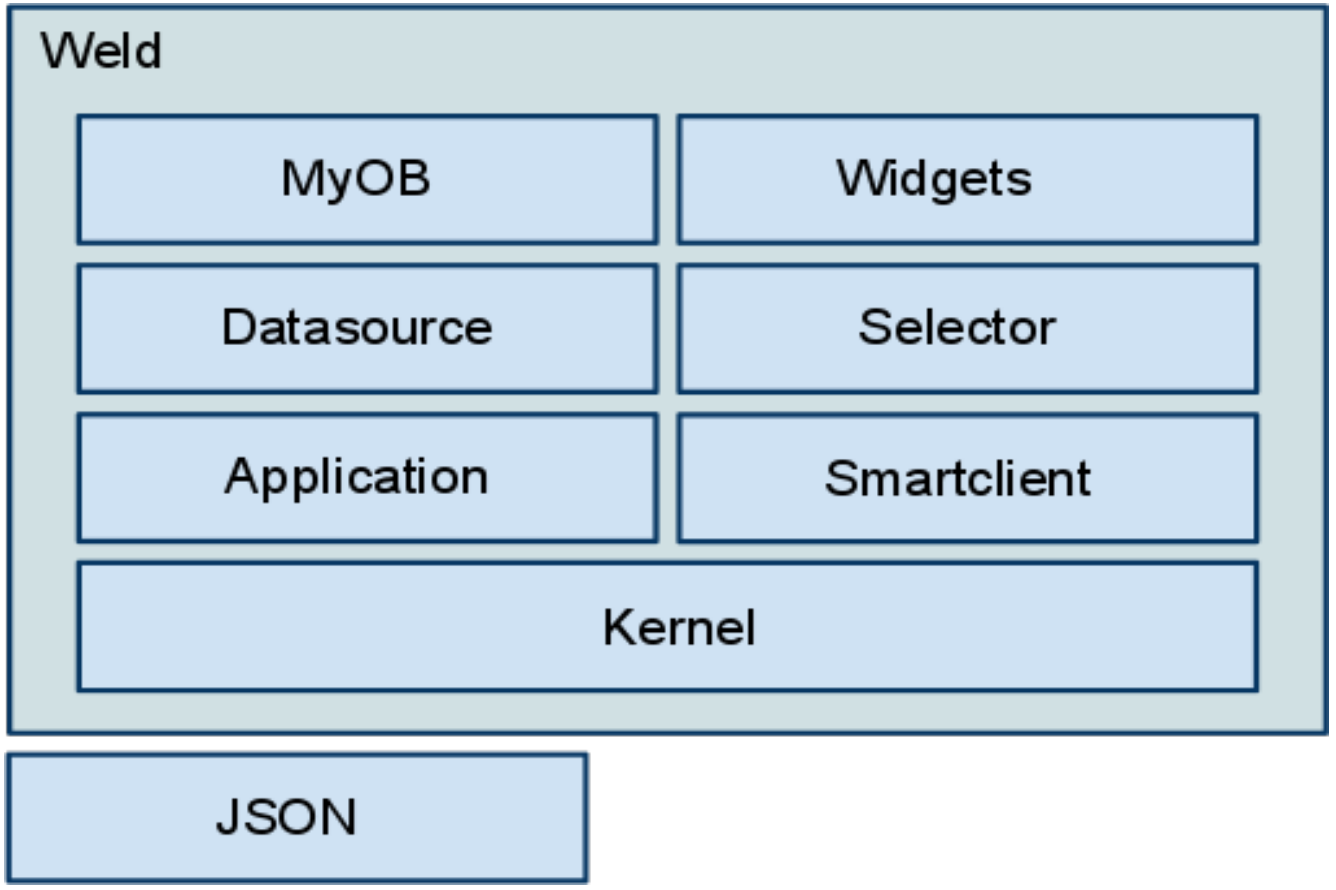
- Oracle 11g or PostgreSQL 8.3
- Sun JDK 1.6
- Apache Tomcat 6.0
- Apache Ant 1.7

Further information about stack components tuning

http://wiki.openbravo.com/wiki/ERP/2.50/Openbravo_environment_installation

Build Ant tasks

http://wiki.openbravo.com/wiki/ERP/2.50/Developers_Guide/Concepts/Development_Build_Tasks


Weld
MyOB
Widgets
Datasource
Selector
Application
Smartclient
Kernel
JSON

- **DBSourceManager**: Java Library based on Apache DDLUtils to manage database structure with XML files
- **MyOB:** The MyOB module provides a customizable home page for a user in the new user interface. MyOb is visualized in a main tab and exists of widgets which can be provided by modules
  http://wiki.openbravo.com/wiki/Projects/MyOB/Developers_Manual
- **Widgets**
  - *URL Widget*: Shows the content of the specified URL.
  - *HTML Widget*: Embeds the specified HTML code in the widget.
  - *Query/List Widget*: Shows in a grid the result of the specified HQL.
- **Selector**

  Provides a new selector field implementation supporting both a suggestion box as well as a popup window with a grid. Supports all Openbravo tables out-of-the-box, including new tables added by modules.

  http://wiki.openbravo.com/wiki/Projects/Selector/Developers_Manual
- **Datasource**

  The Datasource Service module provides a client side javascript implementation of a datasource and a serverside implementation which can handle JSON retrieval and update requests

  http://wiki.openbravo.com/wiki/Projects/DatasourceService/Developers_Manual

- **Application**

  The Client Application module is responsible for building the main layout of the new Openbravo user interface. It makes it possible for other modules to add new navigation bar components and view implementations to Openbravo

- **Smartclient**

  This module integrates Openbravo and Smartclient.

  http://wiki.openbravo.com/wiki/Projects/Smartclient/Developers_Manual

- **Kernel**

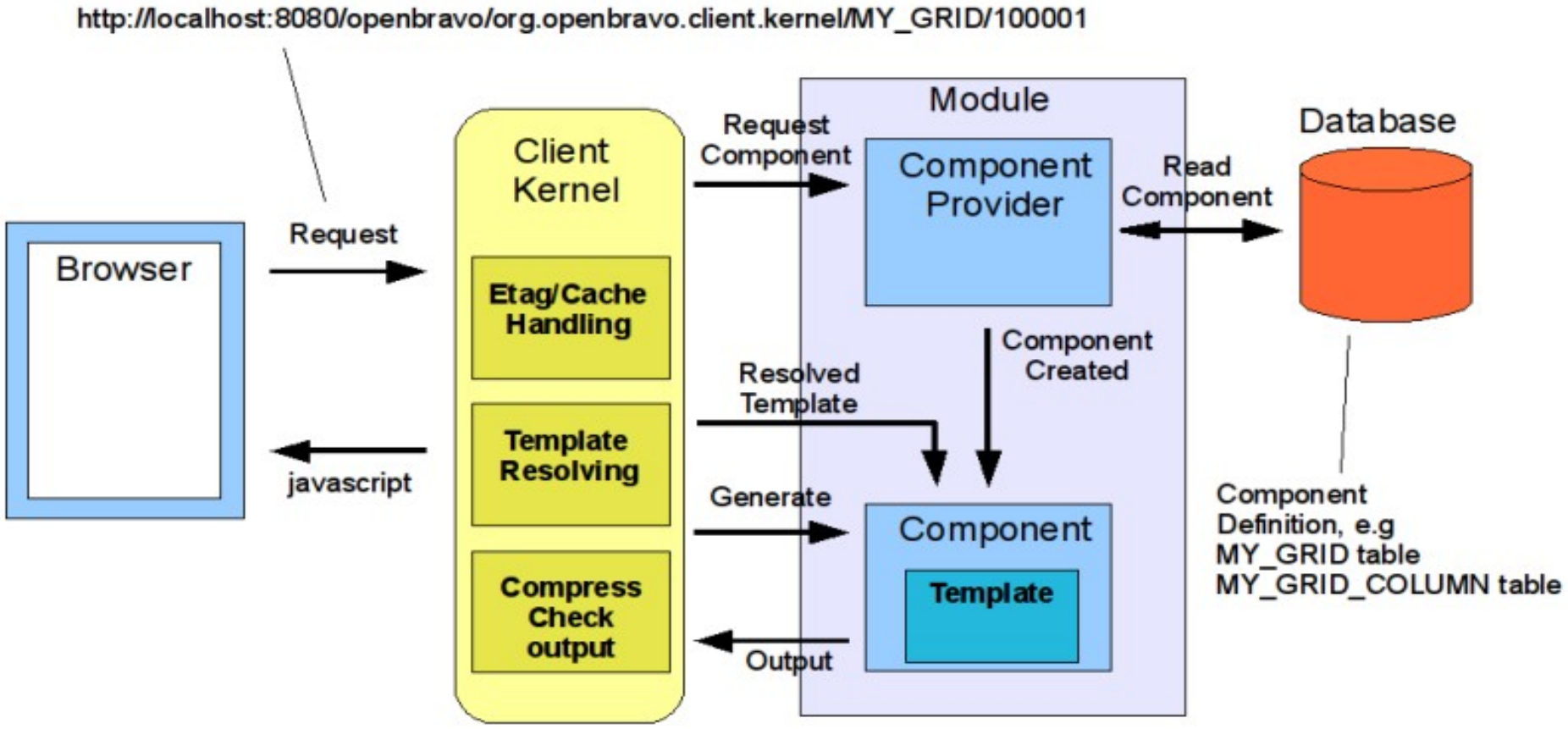
  The Kernel module is the core of Openbravo's new UI The Client Kernel module is the core of the new Openbravo User Interface. The responsibilities of this module:

  - serve (javascript) components to the client (browser)
  - caching/compression/compiling of components
  - generation of components using templates
  - offering an extensible templating mechanism

  Modules can extend and use the Client Kernel module by:

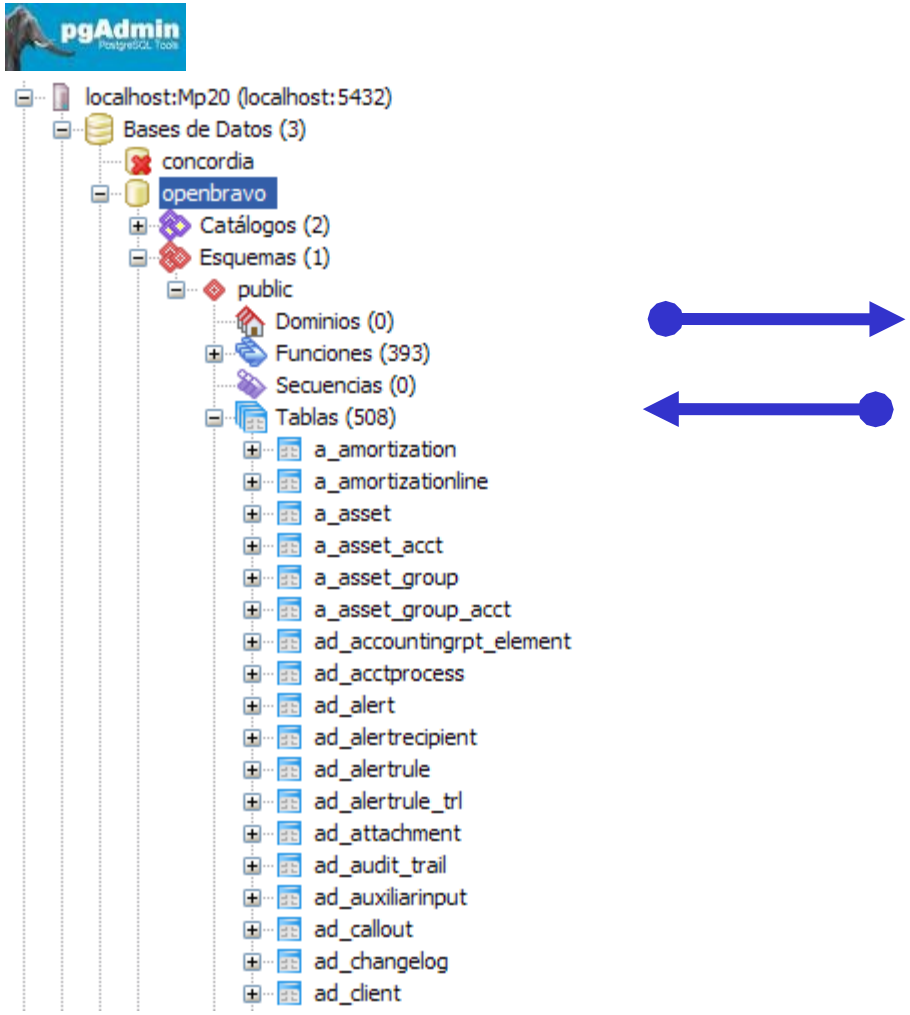
  - implementing new user interface components
  - adding new templates for existing components
  - add new templating languages to the system.

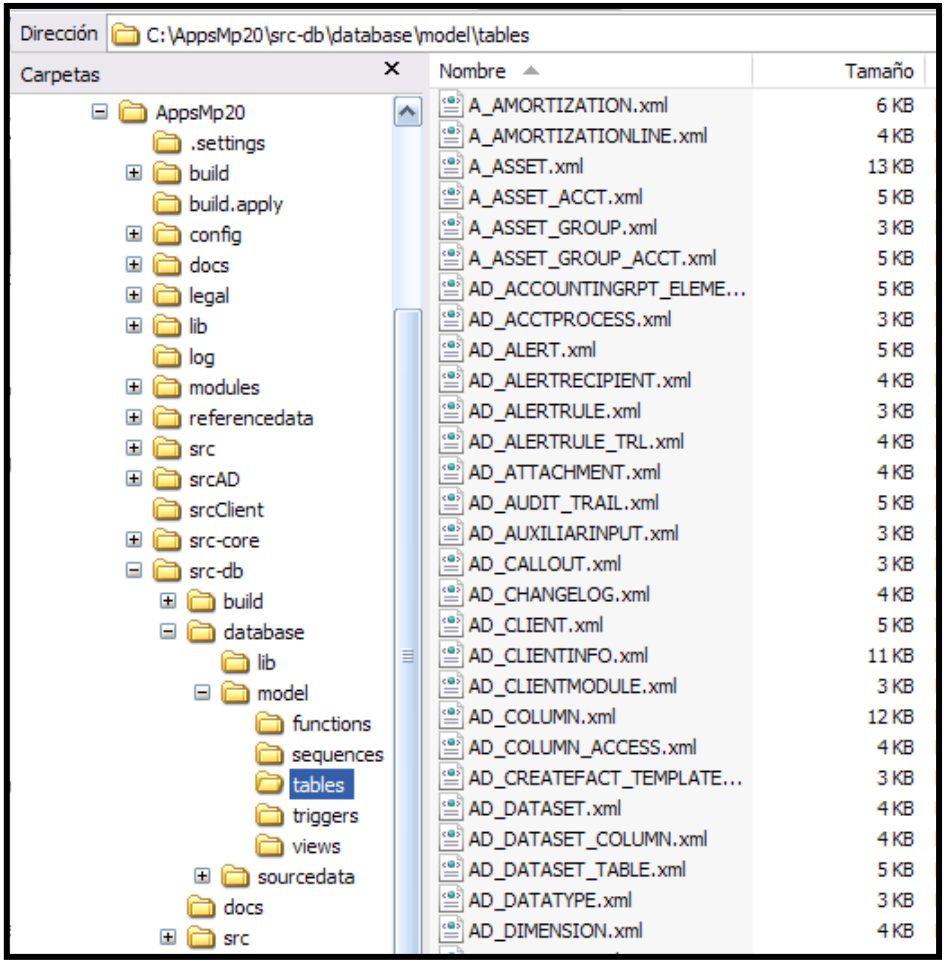
- **JSON** The JSON REST module adds a JSON webservice to Openbravo



http://localhost:8080/openbravo/org.openbravo.client.kernel/OBUIAPP_MainLayout/View?viewId=_102

Architecture
DBSourceManager Library





www.openbravo.com