



Technical environment

This document describes the technological aspects of GIRO's applications.

This application is comprised of many components, and these components are developed in a manner that ensures a seamless integration into today's Enterprise IT Architecture.

The applications are built to meet the requirements of the enterprise in the following areas:

- Ergonomic user interface.
- Scalable architecture.
- Centralized installation and administration.
- Support for lightweight client workstations, both local and at remote locations.

- Easy integration with Service-Oriented Architectures (SOA) and Enterprise Application Integration (EAI).
- Role-based user authorization and integrated authentication for security control.
- Open Data model.
- Web access to personal information for the "subjects" of the application (drivers, operators, clients, letter carriers, etc.).

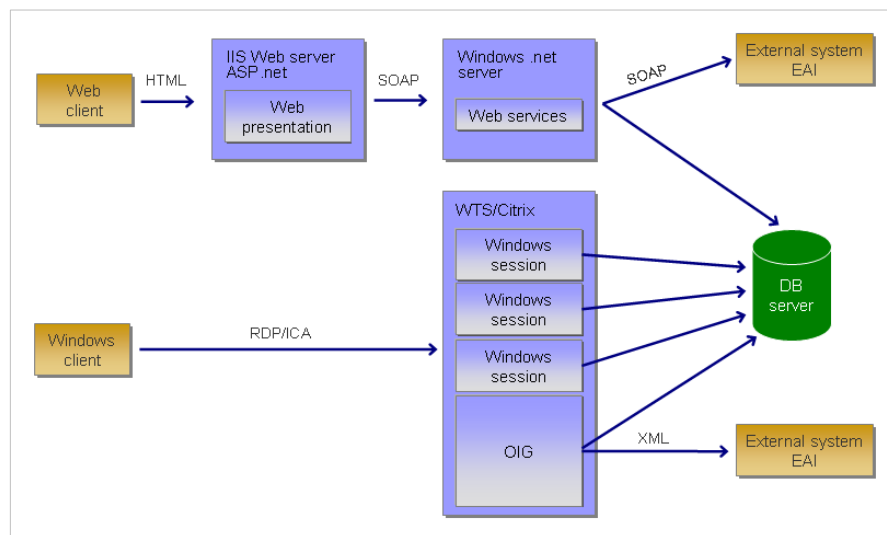


Figure 1 – Architecture schema.

Ergonomic user interface

Based on the Windows® Gui, the user interface for most of the application's functions takes advantage of all the rich tools of a Windows environment. Graphics, toolbars, colors, and fonts are leveraged to maximize the user experience and ease of use.

The user interface is very configurable. All the object lists are configurable. An accessible model of object relations allows building lists of data for any operational need. Object attributes to display can be chosen by the user at will, and formula expressions can be entered interactively to modify their calculations.

Working sets, coloring sets, filters, etc., all work together to customize and streamline the user work environment, allowing him to perform his tasks in the most efficient and comfortable manner.

Scaleable architecture

Windows-based thin clients

Windows Terminal Services® allows to install the application in a 3-tier model, for a flexible and scaleable hardware setup. The user workstation is based on a simple Windows Terminal PC and only requires Internet Explorer® as installed software. Through the distributed display technology of Windows Terminal Services (WTS) or Citrix Metaframe, each user has access to the full Windows Gui of the application.

On the server side, the installation is straightforward. We provide formulas to estimate the hardware sizing based on typical user profiles and activities. This will indicate the number and power of servers to be installed initially. As needs grow, additional demand can be met by simply adding extra CPU and central memory to the servers.

Web-based clients

Our Web-based functions follow a 3-tier architecture.

- The client browser must be Internet Explorer.
- The presentation tier is hosted on Windows IIS, with ASP.net. This server can be scaled according to the demand, either by using a powerful server ("scaling up"), or by using more than one server ("scaling out").

- The business logic tier is a Web Service that runs on a Windows Server back-end server. This tier can also be run on more than one instance. Therefore, a large user demand can be satisfied with the proper number of server instances running on the appropriate hardware. A smart dispatch process will load balance the demands according to availability of the resources and the nature of the requests.
- The Data tier is the central relational database server for the whole installation.

Database

The database is hosted on a relational database tier – Oracle® and Microsoft SQL Server® are supported. These database engines are very powerful and can support a large user population. Again, growing needs can be served by simply increasing the hardware resources to the database server.

Lowering the cost of ownership

Central installation and administration

For any of the installation models chosen, either client-server, 3-tier with Windows Terminal Services (WTS), or Web-Based, all the installation and administration tasks can be handled from a central site, which helps in maintaining a low Total Cost of Ownership (TCO).

For a client-server installation, no component is deployed to the user desktop. All the necessary application components are centrally installed on a file server. The user's desktop only needs a desktop shortcut to start the application.

For a WTS installation, the execution of the software is also done centrally. The user desktop workstation can be installed with the client portion of WTS (or Citrix ICA client), but it can also start from Internet Explorer to link to an enterprise Portal, and from there, initiate the application. This means that there is no need to do a physical installation on the individual desktop.

Support for lightweight workstations

Our application architecture allows an organization to install the products in a configuration where 100% of the user desktops are “light”. With WTS, the user desktop may be a very thin desktop PC – a simple Pentium is powerful enough. With Citrix Presentation Server/XenApp, the ICA protocol allows support for other platforms, such as Linux or UNIX desktops.

This means that any hardware upgrade that may be needed as the use of the applications expands, or as future versions may require supporting new features, can be completely done at the central servers, while maintaining a stable user computing platform.

Integration into the enterprise IT architecture

We have built our applications with a set of tools that allow them to be seamlessly integrated into today’s IT architectures. Many of our clients have implemented ERP systems, and advanced EAI and SOA solutions. We have incorporated many facilities into our software to be able to work in such environments.

- Open Data Model.
- Built-in configurable interface generation tool.
- Web Services for real-time systems integration, directly or through EAI infrastructure.

This document gives some details about each of these aspects.

Open data model

Our data model is completely open. Each application is delivered with a documented database model showing entities, data fields, and relationships. This model describes the contents of the database tables.

Also, each application has a completely documented Object Model. The Object Model represents the dynamic application objects manipulated through the application business logic.

In addition to the static data contained in the database, the Object model gives access to dynamically calculated values (“attributes”) and relationships.

Object attributes are evaluated either from static data fields in the database or dynamically through some computation. In addition, attribute evaluation rules may be re-defined with “attribute formulas”, to closely model each client’s business rules, union work agreements, optimization rules, and preferences. A formula is expressed with simple and standard logic and arithmetic keywords, and may be defined or adapted by authorized client personnel.

Here is an example of a formula that computes the distribution distance in a route, depending if there is some travel time included or not:

```
if (rou_trvl_len = NULL) then
  rou_distance
else
  rou_distance - rou_trvl_len
```

This formula determines if a type of service is “costly”, depending on its position in a route and the preceding service type

```
(dem_serv_type = "Pickup") AND
(dem_serv_type[PREV()] <> "Drop") AND
(rsg_pos_in_route <> 1)
```

The object model is used in the interactive application, but it is also leveraged by the OIG tool, as described in the following section.

OIG interface tool

A built-in tool, OIG (Object Interface Generator) is provided to build interface file definitions. The tool contains a scripting language that allows defining any form of data extraction from the application's data objects, using any of the object's attributes.

The tool offers powerful data manipulation and formatting capabilities and can generate interface files either in traditional fixed-field or CSV formats, or as XML documents (with related XSD schema definitions).

Figure 2 is an example of a simple interface definition.

The scripting language allows any properly trained user that understands the business entities manipulated by the application, to define his own data extraction, for example to feed into Office applications such as Excel or Word, or Desktop publishing systems.

The use of XML data extraction, along with of our Object Model ensures easy integration with third party tools, such as Crystal Reports for ad-hoc report definitions.

This interface creation capability is used in many cases to exchange data with other enterprise systems. These interface files are the simplest method to integrate into applications where the data does not need to be updated in real-time.

The scripts can be run on an ad-hoc basis, or can be scheduled to run on a regular calendar, using any standard job scheduling tool.

Real-time integration

Depending on the needs, we can offer functions built as Web services, or we can access Web services offered by enterprise applications or an EAI infrastructure.

For more real-time access to data within our data model, or to allow an external process to trigger transactional actions within our business logic, we offer a set of Web Services (API) that give access to application data and functions. We have a basic set of functions, which vary according to the application.

These Web Services can be extended to support any specific needs of an enterprise. The Web Services support all the major standards and protocols, such as SOAP, HTTP, XML, and WSDL. The Web Services are built using the Microsoft .NET infrastructure.

Web services also allow building Web pages or sites giving external users, clients, or employees, remote access to some set of functionalities.

```
file service_route
{
  foreach service_route
  {
    order_by rou_identifier
    line service_route
    {
      item text {value '"Route : "'}
      item rou_identifier
    }
  }

  foreach service_unit
  {
    order_by dem_identifier
    line service_unit
    {
      item text {value '"Demand : "'}
      item dem_identifier
      item rsg_arrival
      item rsg_qty_1_cum
    }
  }
}
```

Figure 2 – Example of a simple interface definition.

Security

Application access is role-based. User authentication is based on the Windows user logon process. For authorization, the application contains tools to tailor any number of profiles where a profile has access to all or parts of the application. The accesses can be specified at the sub-system, module, screen, or function level, and for each part of the system, access level can be specified as read-only, update, delete, or none. The parts without access are removed from the user's menus.

These profiles are exposed through Microsoft Active Directory, which allows an easy administration task, where any help-desk agent may assign users to specific roles within each application.

Thus, end-to-end security is maintained, through single sign-on, the fact that all of the application is run on central servers, without any data traveling to the end-user workstation.

For Web-based modules, SSL-based communications ensure secure sessions.

Web access

In most of our applications, there is a need for the "subjects" of the applications to access data of interest to them. This can concern public transit vehicle drivers and operators, transport clients, Postal Office Branch Managers, and letter carriers.

For this, we have developed a Web-based access module. The Web-based access functions are implemented in ASP.NET. They are implemented as a presentation-logic tier that takes advantage of the SOAP Web Services of each application, giving access to business logic that can serve the user's needs.

This type of Web access can also be used to deploy functions of the applications to occasional users of the applications, for who it may not be cost-efficient to invest in a full training session for the whole product.

Technical requirements

Database:

- Oracle® or Microsoft SQL Server®

Application:

- Windows 2003/2008 Terminal Services® (optionally with Citrix Presentation Server/XenApp)
- Microsoft® .net framework

Web Server:

- Microsoft IIS

Web Browser:

- Microsoft IE 7 or later

Protocols and standards:

- SOAP, HTTP, TCP-IP, SQL, XML, XSL, and XSD

Outputs:

- HP-compatible or Postscript-compatible

Custom Reporting:

- Crystal Reports®