



## → Introduction

Open mobile IS is an open source project that aims to provide all the necessary tools, API, and documents enabling powerful nomad applications development. Heart of the project, the java framework is divided into components providing all the needed functionalities.

Mobile applications are any type of applications used by mobile workers outside the company area. Mostly that kind of applications concern sales forces, technicians, drivers and deliverymen... Open mobile IS allows you to build complex and performant applications like SFA (sales force automation), CRM (client relationship management) applications, running on any kind of devices, even the lowest CPU ones.

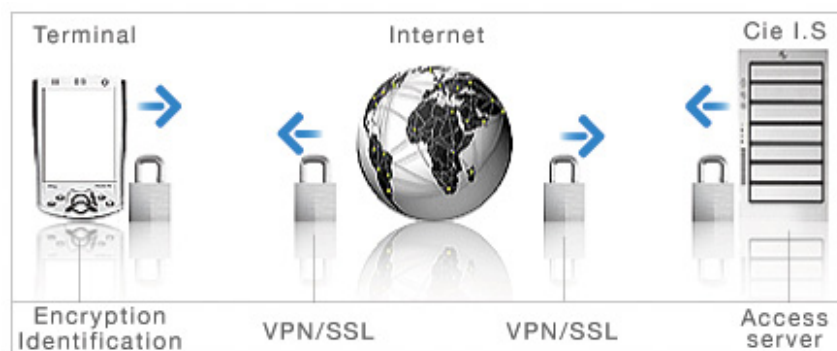
The key point is that nomad users need access to the company data anywhere, at anytime. To do so, they use a terminal (PDA, smartphone, laptop, Tablet PC...) offering an access to the company applications. Terminals are connected to the company from time to time to update there data. One of the main frameworks' added value is to provide an efficient and secured synchronization mechanism between the terminal and the company information system.

## → Main concepts

This framework is based on our 5 years experience in the development of mobile applications for our customers. The 4 main principles that have driven our technical and architecture choices in the making of this framework are:

**security:** the framework provides an answer to all security constraints generated by mobile applications. The 3 main security fields are :

- Data protection inside the terminal.
- Data protection during transfer between the terminal and the information system.
- The information system access protection.



**Accessibility:** A mobile application has to be user friendly. Final users are not computer aware. In order to maximize the chance to see nomad application accepted, these applications must have an ergonomic adapted to users' needs.

**Availability:** Applications, and therefore the data, must be available everywhere, at anytime. If the data are available only from time to time, users won't adopt the application.

**Evolution capabilities:** terminals, software technologies evolve rapidly. Applications must be able to follow this evolution.

## → The Frame-work

### Introduction

We answered these constrains by developing a java framework providing the following components:

- **An embedded object database** optimized for low CPU or memory terminal called FODB.
- **An embedded application server** providing the servlet API, or a specific services API.
- **A synchronization API** dedicated to data and applications synchronization.
- **A high level module** providing basic functionalities needed by all nomad applications.

### -Access integration

In order to make easier application usability, the framework provides some ergonomic solutions to classic data management problems. These solutions have been tested on real applications since years and have been fully approved by nomad users. As a base of all these solutions the framework provides a web interface for all application. Internet applications are now used by billions of people and there GUI (graphic User Interface) paradigms have proven to be the most accessible. Beside this web integration, the framework provides some functionalities making easier the development of web application for mobile devices.

## -Availability integration

Availability is provided as follows :

- **data synchronization**, so users can access their data anywhere at anytime.
- **high stability of the framework**. The framework is in production since several years with many different type of applications :
  - CRM (client relationship management)
  - SFA (Sales force automation)
  - Maintenance apps....

## -Application evolution

Application evolutions are facilitated by :

- **The framework is developed in Java and is compatible with Java JDK 1.1.**  
Native functionalities like RAS use or PIM integration are mapped with a Java API and we provide a version for all supported terminals and OS (windows, Pocket PC and Linux).  
The framework provides an abstraction layer from the system that minimizes application development when changing the terminal. Some of our customers have developed applications which have been deployed on heterogeneous OS and terminals environment : Pocket PC 2000, 2002, 2003, 2003 SE... with no change in the application code. Only the version of the framework changed.
- **Web interfaces dissociating the GUI from the business code.**  
When the terminal changes (and so the screen), only the presentation layer has to be modified. Some of our customers' applications have been deployed on both windows PC and Pocket PC for PDA, with an identical business logic code.

## → Embedded database

### Introduction

**Open Mobile IS Embedded database** is a simple object database optimized for low CPU and Low memory terminal. It provides a simple API storing and retrieving objects from the database. The Open Mobile IS embedded database or **Fast Object DB** or **FODB** is perfectly suited for mobile and embedded applications. The object storage capability facilitates its use inside Java (avoiding mapping tool use and so on).

Simple query API (see from SODA API for more details) provides an easy way to manage objects inside the application.

### Description

**FODB** is constructed using collections. Each collection is a set of objects of the same type. To retrieve objects, indexes are added to the collection. Index can index objects, methods or fields. The main FODB functionalities are:

- Objects method or fields indexation.
- Simple query API derived from SODA API.
- Transaction management with auto commit.
- Plug-ins synchronization facilities.

## → Web Server and Servlet API

### Embedded Web Server

All OpenMobile IS developments are based on the Internet paradigms. Applications are developed on a web server and all user interaction are made through a browser. In the case of embedded application, both server and browser lean on the same terminal.

**Open Mobile IS embedded server** provides an implementation of the javax.servlet API. Thus, OpenMobile IS development can be deployed on any HTTP server supporting servlet API (Tomcat for example).

In order to facilitate embedded applications development, the project provides an embedded java WebServer (based on Acme.Serve.Serve single-class WebServer : <http://www.acme.com/java/software/Acme.Serve.Serve.html>) optimised for low cpu/memory devices.

### Open Mobile IS : a Service Manager

with an aim of making easier OpenMobile IS development portability across different architectures, we added to the servlet API a new abstract layer named **Service API**. Service API offers a better integration to all functionality provided by the OpenMobile IS platform (template service, module and profile management, ...).

Service API is very close from the servlet API, using HttpServletRequest and HttpServletResponse (from the javax.servlet API), for the treatment of the request and the response.

As most of the services are meant to send a graphical response (HTML pages), our API also provides a service called TemplateService. This service uses a template engine facilitating HTML generation.

This technic also provides an easy way to manage devices' display changes.

The default template engine used is freemarker (<http://freemarker.sourceforge.net/>).