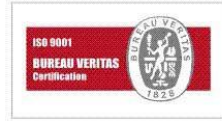




RIS d.o.o.
Address: Pilepic 10, 51215 Kastav, Croatia
Phone: +385 51 687 500, Fax: +385 51 687 501
e-mail: info@ris.hr, web: http://www.ris.hr



Last Update: August 2015



Thin@ System Architecture V3.2

Introduction

Welcome to Thin@ System Architecture manual!

Modern business applications are available to end users as an internet service, deployed as Software as Service (SaaS) applications.

The acronym RIA (from Rich Internet Application) emphasizes that the application is not only run over the internet, but that it is also rich in contents and graphic elements. Such applications differ greatly from the usual HTML-based web technologies, also in the way of how users interact with the application. Their similarities with 'desktop' applications make them more user-friendly.

The most famous brands in the software industry such as Microsoft, Adobe, and Sun (now Oracle) have tackled this question and each made their own software development platform and RIA solution – Adobe AIR, Java FX and Microsoft Silverlight.

Thin@ also makes it possible to run Rich Internet Applications in and out of browser. However, contrary to other RIA solutions, it relies on Clarion – a 4GL programming language and ARAD (Advanced Rapid Application Development) environment specialized in fast development of business applications.

Due to this specialization a Clarion/Thin@ application development platform can offer many advantages compared to using general-purpose solutions.

Moreover, Clarion/Thin@ is one of the few software development platforms which promise to deliver the 'write once – run anywhere' paradigm, meaning that *the same* Thin@ Clarion application can run on any platform (Windows, Linux, Mac), and it will be able to run on mobile devices as well (Android, iPhone).

These are some of the reasons which place Clarion/Thin@ as a respectable and competitive solution for deploying SaaS Business Internet Applications.

On the following pages you'll learn what Thin@ is, how it works and what are its system components, as well as its distinctive features.

Contents

- 1. What is Thin@? 3
- 2. Thin@ system overview 4
- 3. Thin@ Application Server Grid 5
 - 3.1. Thin@ Application Server components 6
 - 3.2. Functions performed by Thin@ Server-side components 7
- 4. Thin@ Client-Server Communication..... 8
- 5. Thin@ Clients 9
 - 5.1. Thin@ WinApi Client 9
 - 5.2. Thin@ Java Client (beta)..... 9
- 6. Thin@ Key Features 10
 - 6.1. Thin@ Additional Features..... 11

1. What is Thin@?

Thin@ is a Rich Internet Application (RIA) platform and Smart Client solution for Clarion.

With Thin@ it is possible to deploy Clarion client-server solutions over the Internet/intranet to end-users. This solution avoids standard client-server-like side effects (database client requirements, big amount of application code on client side, slow version refresh, database security issues, etc).

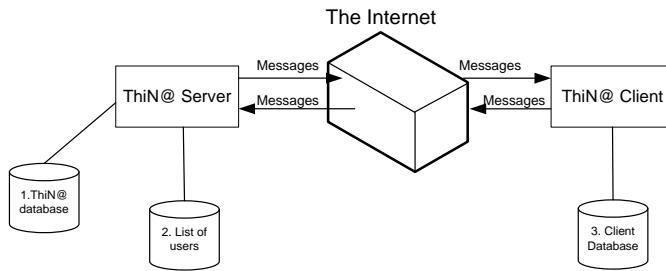
It means that with a RIA your client-server Clarion application does not have to suffer from any transition penalty from the client-server environment to the web environment. Client-server Windows API environment was always more suitable (user-friendly), faster and more reliable for end-users. You don't have to lose those advantages if you decide to run your application over the Internet in a RIA solution.

Basically, you install the Thin@ Developer Environment (Clarion template, binaries, source etc.), add the Thin@ template to your application and compile. Now your application is Thin@-ready.

To deploy your application over the Internet a small client program is used (cca. 3MB in size) called Thin@ Client. End users will have a complete client-server feel, as if they are using your application in a LAN environment with classic Windows API application look & feel.

There is also the Thin@ Java Client which can run a Clarion Thin@ application inside a web browser and on other platforms (Linux and Mac).

2. Thin@ system overview



What is a Thin@ Application Server?

A Thin@ Application Server is a computer with installed Thin@ server-side components. It represents the middle tier(logic tier) in a **multi-tier architecture**, as it sits between the Thin@ Client (presentation tier) and a Database (data tier).

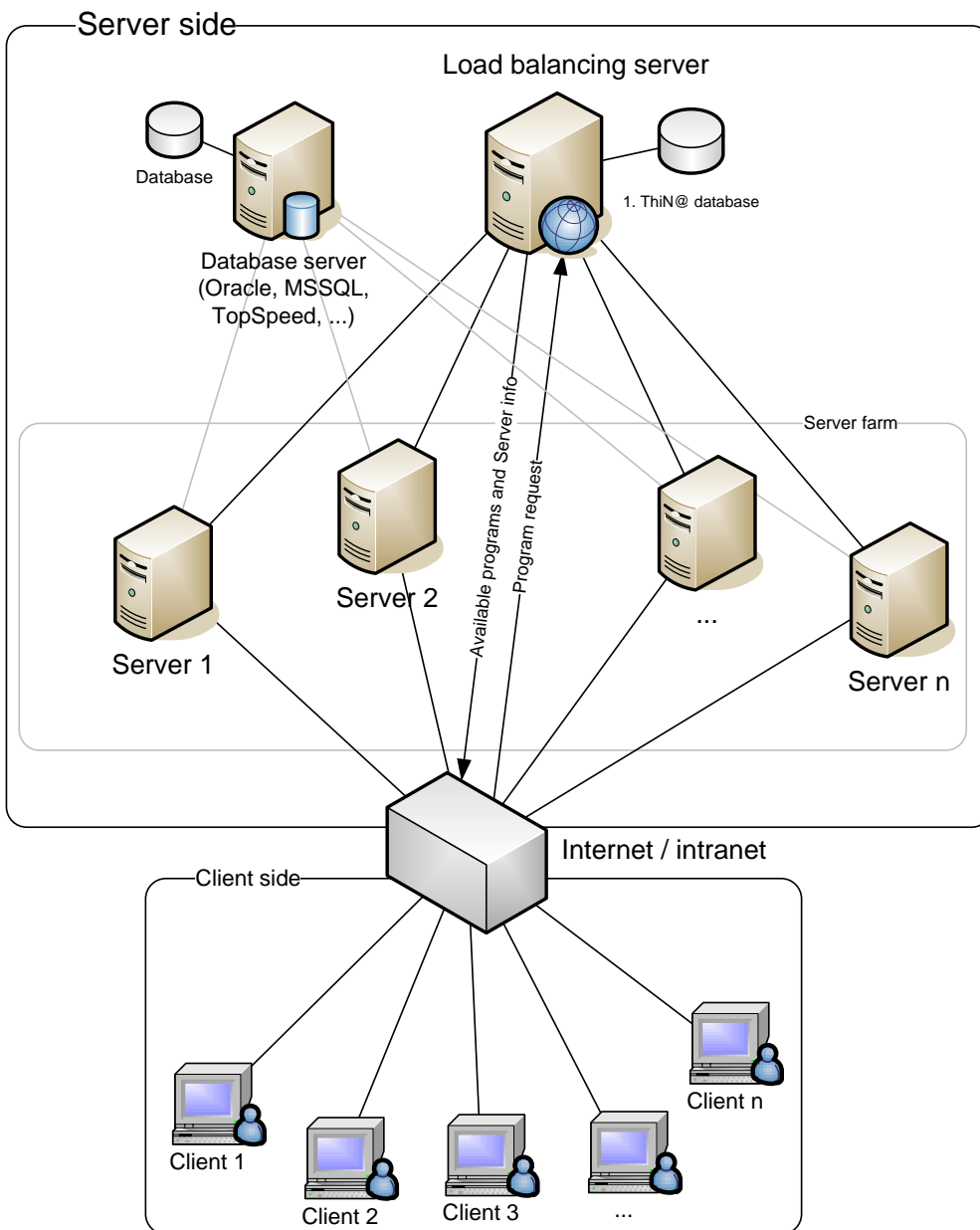
What is a Thin@ Client?

A Thin@ Client is a **Smart Client** application which communicates with a Thin@ Server over the internet or intranet and renders the Graphical User Interface.

Currently there are two types of Thin@ Clients:

- the Thin@ WinApi Client (which requires to download the client application and works outside of a browser)
- the new **Thin@ Java Client (beta)** which does not require a download and works on any OS with the Java Runtime Environment (JRE) installed. It works either inside or outside the browser.

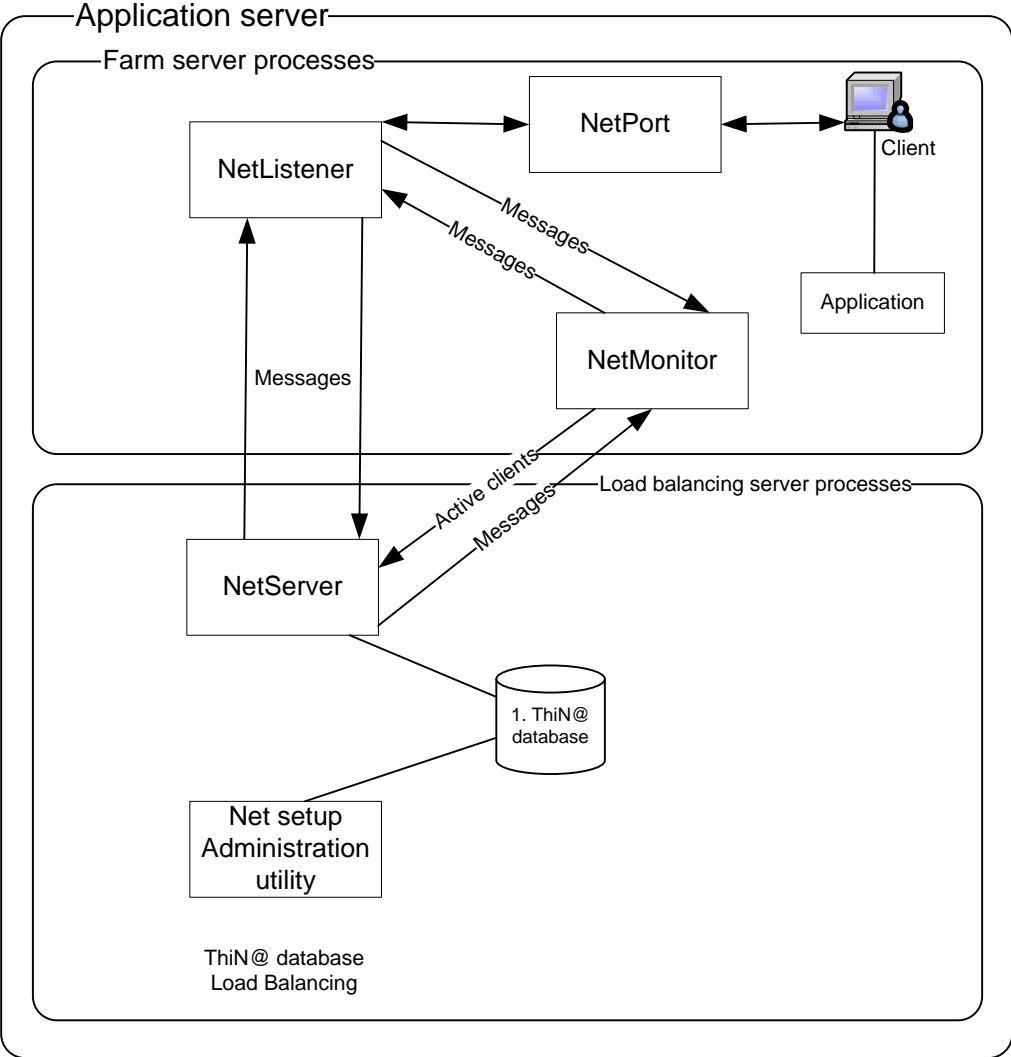
3. Thin@ Application Server Grid



Thin@ Application Server Grid

The main part of a Thin@ Server Grid is the Thin@ Load Balancing (Main) Server. A Thin@ Server Grid has one Main Thin@ Server and can have a practically unlimited number of Thin@ Application Servers. As you add more users and applications to the Thin@ environment, you will need more computing power, which can be extended by simply adding a new Thin@ Server.

3.1. Thin@ Application Server components



3.2. Functions performed by Thin@ Server-side components

I Listener functions (NetListen.exe):

- receives client request
- sends information to the client
- sends server address and port information
- starts the application
- retrieves the status from the main server

II Monitor functions (NetMon.exe):

- monitors all the applications and kills the inactive ones
- sends information to the main server

III Server functions (NetSrv.exe):

- resides only on the main server
- receives information from the monitor
- sends information to the listener
- sends request for application to be terminated (either from client or net setup administration tool)

IV Port functions (NetPort.exe):

- Improved security – clients connect through a single port
- Reduced administrative tasks (opening additional ports on firewall etc.)

V Net setup administration utility functions (NetSetup.exe):

- update and overview of the main server options
- management (including connection kill) of server connections

4. Thin@ Client-Server Communication

The Thin@ Application Server executes Clarion applications in hidden mode and sends the User Interface definition in an XML-like file to the Thin@ Client. The Thin@ Client library reads the XML and renders the UI.

When the user performs an action, for example clicks a button, the Thin@ Client will send information about that action back to the Thin@ Server, which will perform the exact same action in the Server side application. If there is a visual change in the application screen, the Server side application will now send a new XML-like file to the Client, so that the Client User Interface reflects that change.

Notice that the Thin@ Client and Thin@ Server DO NOT exchange graphic information on the screen, as would a **Remote Desktop Connection** solutions such as **Terminal Services**. Therefore, Thin@ provides the user with a fast and interactive user interface over the internet which was impossible before the evolution of **RIA/BIA**.

5. *Thin@ Clients*

5.1. *Thin@ WinApi Client*

The Thin@ WinApi Client is written in Clarion and is using the Windows API to draw the user interface. The Thin@ WinApi client runs only on the MS Windows platform.

The WinApi client is open-source, meaning that any Clarion programmer can modify and recompile the client to suit their own needs, e.g. make interfaces to external devices (scanners, plotters etc.), add custom support for 3rd party products etc.

5.2. *Thin@ Java Client*

The Thin@ Java Client is a new Thin@ Client written in [Java SE](#).

It makes a Clarion applications available inside(and outside) a web browser and on various platforms, including Windows, Linux and Mac.

It uses the [Java SE Swing library](#) to render the application User Interface. It can run on any hardware that supports the [Java Virtual Machine \(JVM\)](#). The Thin@ Java Client can be run in three distinct ways: 1) as a standalone application on your desktop 2) as a Java Applet in a Web Browser 3) as a standalone application that is accessible through a Web browser using [JavaWebStart](#).

6. *Thin@ Key Features*

- **easy internet deployment** with Windows API look style
- **very stable** working environment
- **run as fast as desktop applications**
- **runs on Windows, Linux, Mac**
- **identical user interface** as in original application including pictures, icons, menus and most Clarion control types
- **Support for complex ActiveX components**
- Possibility to create a **Rich User Interface** with the use of Codejock Wrapper products (<http://www.noyantis.com/clarion/codejock.html>)
- Support for a number of **Clarion 3rd party products**
- **multithreaded application support** (allows multi-window environment inside applications)
- **auto-reconnect option** included (if you lose your internet connection or something happens with your network, the client will automatically try to reconnect and as soon as the network is back you can continue your work without any additional action)
- **balanced application load** on multiple servers
- **SaaS (Software as a Service) maturity level 4 support:** usage-based billing calculation algorithms integrated allowing the host company to charge by usage time

6.1. *Thin@ Additional Features*

- **Powerful and easy server administration**
 - **centralized server management tool** (the farm database is maintained centrally)
 - **wizards** for all administrative options
 - capable of **reviewing and terminating all user connections**
 - With integrated **log of user activity** (log of user login / logoff actions can be viewed through the NetSetup administration utility. It's possible to view all session data including session time and active session time.)

- **Unique features of the Thin@ Client**
 - **client user connection review** (user connections can be reviewed in the thin@ client main window)
 - integrated **automatic client upgrade solution** (new versions of the client with additional files, for example new functionalities, can easily be distributed among users)
 - possible to modify the appearance and behavior of the Thin@ Client (open source)
 - integrated client proxy support with authentication

- **User licensing and authentication**
 - **username / password user authentication** with licensing included (you can set the expire date per user license)
 - **licensing per application** included (it is possible to declare which application a user can activate)
 - **client recognition by username or machine unique key** (allowing clients to connect through available license slots. For example, if a client has only one license slot and he tries to connect again, he will be asked to terminate the existing connection. Also, this is useful in the case of an unpredicted application failure where such a failure would generate "garbage" as "ghost" application process on the server side that would wait inactivity timeout)
 - **external authentication solution** (for example you can authenticate thin@ users with database authentication or with windows authentication, such as Active Directory)
 - **company / user classification supported**

- **Other features**
 - **does not require any other products to load** (such as Clarion app broker) except the Clarion client tool, the server program and the Clarion application
 - integrated and unique **internet communication error checking** (in case of instable networks)
 - **file download and file upload** included (file size limit can be tweaked)
 - allows **running command prompt commands, scripts and programs on the client machine**

- allows running a **web test environment** on the local machine without any additional server program
 - **integrated time check on download** of icons, pictures and other window file controls (icons and pictures download only if there is a file time difference)
 - instead of using only one timeout per application, thin@ uses the **'long timeout'** feature as well, which is triggered in case of long running operations or reports
 - **dynamic creation of client server directory** (allowing the client's uploaded files to be distinguished on the file system from other clients' files)
- **Almost all Clarion features supported, including:**
 - **Multithreading**
 - **ActiveX**
 - **Edit-in-Place**
 - **Legacy support**
 - **Listbox column header sorting**
 - **Listbox format manager**
 - **Query By Example**
 - **Calendar class**
 - **RTF controls**
 - **ASCII Viewer**
 - **SVGGraph**
 - **integrated report preview** of classic WMF files with **PDF export support**
 - **PROP:Timer** events
 - POSTing events to other threads
 - **etc.**