RfIP Framework Overview

Ver. 2.1

1/08/2007

LRNI
Lost Recovery Network, Inc.

1 Summary

The Radio Frequency Integration Platform (RfIP) framework is a distributed system designed to harvest and report Radio Frequency Identification (RFID) data from across an enterprise using patented RfIP hardware devices. The framework is both scalable and extensible because of its pluggable architecture. The RFIP framework API also allows 3rd party developers to customize and extend the framework to meet their solution's needs.

RFIP is built using Microsoft's .NET framework. Developer's can take advantage of the Common Language Runtime (CLR) and Common Type System (CTS) to develop plug-ins for the framework using any supported .NET language (i.e. VB.NET, C#, J#,etc.)

2 High Level Installation Architecture

The RFIP framework is intended to be deployed in a distributed configuration in order to manage RFID load and service an entire enterprise. The framework can be configured as a global framework when an enterprise has multiple buildings and campuses that may exist in different cities or on opposite sides of the globe.

The most basic view of the framework's installation architecture is comprised of five distinct parts.

- LRNI Antennas
- RFIP Hardware
- Standalone RFID Readers
- Data Processing Server(s)
- RFIP Database

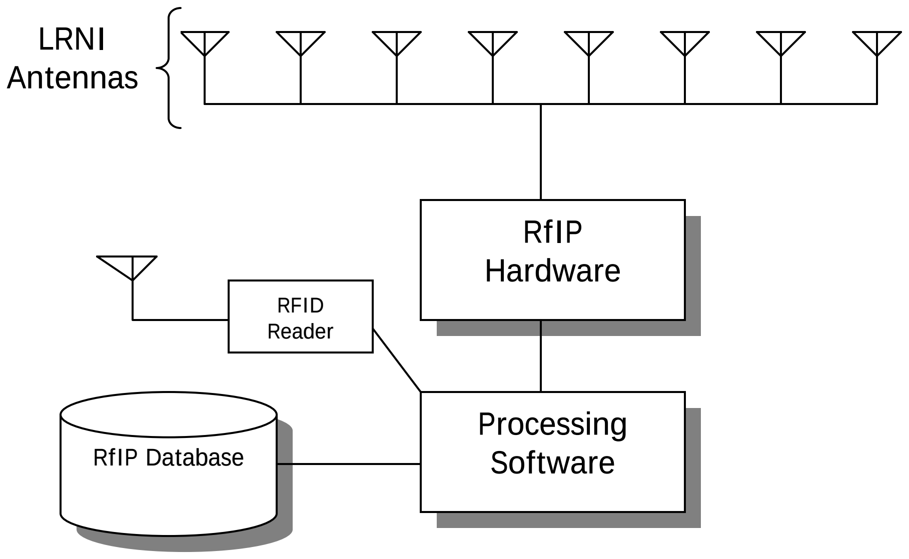

Figure 1: Basic Installation

2.1 RFIP Hardware

Each RFIP Hardware device can marshal data from several LRNI antennas to the backend processing software. As the size of the enterprise RFID footprint increases, the number of deployed RFIP devices will increase. RFIP devices increase scalability and reduce cost.

2.2 Processing Software

The processing software provides the pluggable software architecture that drives the RFIP framework. The processing software can be either integrated into a standalone server or into the RFIP hardware itself.

2.3 RFIP Database

The configuration database stores many of the configuration options for the RFIP framework as well as provides a tracing and error logging repository. 3rd party solutions that may want to extend the capabilities of the framework can access the RFIP database either through direct SQL connections or through the RFIP Data Manager API.

2.4 Enterprise Installations

As the RFID coverage area expands, multiple RFIP devices may be installed. RFIP devices can be chained in a hierarchy type configuration.

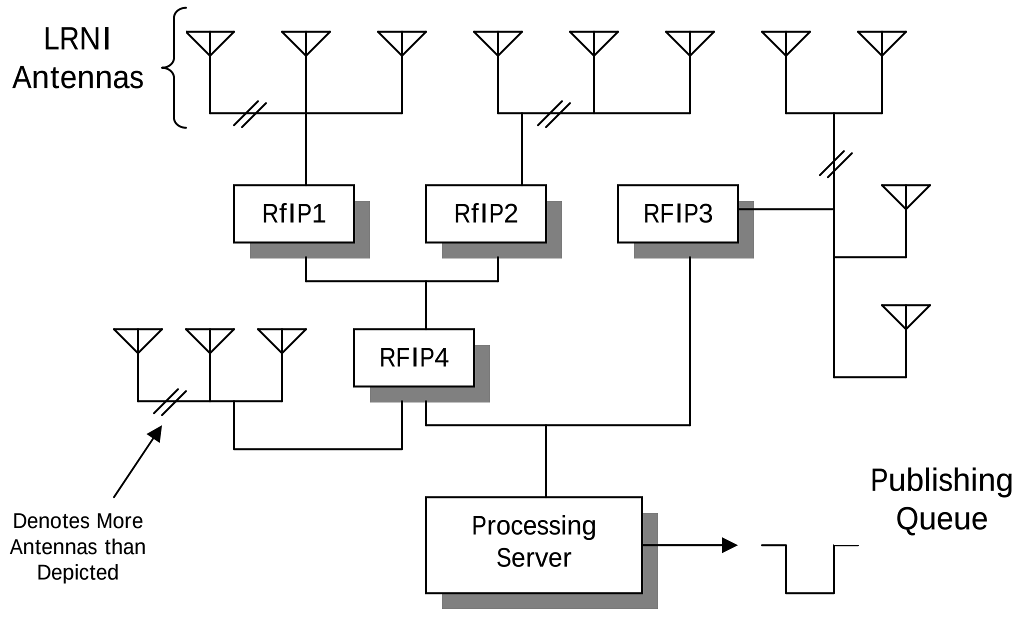

Figure 2: More Advanced Installation

The installation depicted in figure 2 shows a more advanced installation that uses multiple RFIP hardware devices. It is important to note that each RFIP could be in completely different locations (i.e. different cities) and the transport mechanism between the RFIP devices can be batch or real time interfaces using any type of protocol necessary. Out of the box, the framework provides a raw TCP/IP transport mechanism, but queues or web services could also be leveraged.

3 High Level Software Architecture

The RFIP software framework is architected as a pluggable, componentized framework that allows easy customization and extendibility. The framework provides for three configurable interfaces. These interfaces are data readers, data processors, and data publishers.

All pluggable components are created by implementing the appropriate .NET interface for the desired plug-in. Plug-ins are installed into the system by an XML based configuration file which is inspected when the system is initialized.

3.1 Data Reader

A data reader is the component that reads data from an RFID device. Out of the box, the RFIP framework supports an RFCode^{TM} based data reader, but a new data reader could easily be developed to support any other RFID hardware manufacturer, both active and passive based.

The data reader does not necessary have to read data from an RFID reader device. It may be configured to read data from another RFIP hardware device in a hierarchal, distributed install.

3.2 Data Processor

A data processor allows the framework to process data as it is read from the RFID hardware. The framework allows for any number of data processors to be chained together to transform, filter, or manipulate the data in any way necessary. Data processors may also provide trigger capabilities when certain events occur. Trigger can perform such tasks as sending emails or starting other processes.

3.3 Data Publisher

The data publisher is the final pluggable component. It is responsible for publishing the processed data to some data sink such as a queue, file, or database. A data publisher may also publish data to another RFIP device in a hierarchal, distributed install. Data publishers along with data processors can smooth and filter RFID data so that data presented to the application layer is clean. Publishers can either send all RFID tag beacons or specific tag events such as inter-zone or intra-zone moves or tag property changes.

3.4 3rd Party Integration

The RfIP framework provides a web service API through which applications can interface with the framework. The API is completely web service enabled and allows for such actions as:

1. Managing Assets
2. Managing RFID Tags
3. Managing Asset Categories
4. Locating Assets
5. Starting/Stopping the system

In addition to the web service API, the system can publish application level events to other applications using simple XML formatted messages. Some of these events include:

1. Asset Appear
2. Asset Disappear
3. Asset Intra-zone Move
4. Asset Inter-zone Move
5. Asset State Change
6. System error

The way in which events are published is configurable. The plug-in mechanism discussed earlier allows for publishing to queues, web services, databases, or any other service.