SupplyChainDigest™
The Supply Chain Technologist – January 20, 2005

Can an RFID System Play Well with the Rest of Your Network? (Part II)

By Mark Fralick
SupplyChainDigest Contributing Editor

Last time in this column series we discussed how readers exist on the network and my illustration of the RFID domain model. To review: Using the model, below, we can start to see where the network pressure will manifest itself. The largest issue currently has to do with the raw, unpersisted data moving between the Edge and the Execution domains. As you plan your RFID implementation, some of the key decisions facing you are as follows:

1) Where should physical and logical filtering occur?
2) How and where RFID events are triggered (correctly implementing context decisions)?
3) How to coordinate processing between the Edge and Execution domains?

Having good answers to these 3 questions will yield a very scalable RFID implementation that fits into your business processes.

Tag Object Domain	**Antenna and Reader Domain**	**Edge Domain**	**Execution Domain**	**Business Process Integration Domain**	**Enterprise and Collab-oration Domain**
The **Tagged Object Domain** contains the tagged products in a supply chain; or any other assets or locations will be tracked via tags.	The **Antenna and Reader Domain** is the interface between the world of Radio Frequency and the world of the corporate network.	The **Edge Domain** is responsible for filtering and the aggregation of the volumes and volumes of data from the readers.	The **Execution Domain** can be thought of as the application domain and is the intermediary between the Edge and the rest of the Enterprise.	The **Business Process Domain** provides the connective tissue for the RFID information to fit into the context of business processes.	The **Enterprise and Collab-oration Domain** provides the abilities for the enterprise to operate within the Supply Chain community.
	The **System Management Domain** exists horizontally across the other domains and provides remote management, monitoring, software/firmware upgrades, etc for hardware (readers, antennas, concentrators, servers, etc.)				

RFID Domain Model Data Volumes

Raw, Unpersisted Data	Lots	Some	None	
Clean, Persisted Data	None	Some	More	Lots

Push Processing Down as Far as Feasible

In the domain model diagram, the lines separating the domains are nice and neat. The readers do this job, the edge does that job, and the execution system deals with the application world. It is a convenient way to think about the problem. In the real world, as things tend to be, it is just not so nice and neat. Some of the lines are pretty blurry. Given that fact, it is logical to ask what the value is of this conceptual approach. It is valuable, however, and for two reasons. First, the work of each domain still needs to be performed. While the venue of that work may be a bit flexible, the fact that the work needs to occur is not. Second, the lines between the domains are useful if considered as potential network hops. If you have the opportunity to push some of the logical work of one of the domains down a level it could have a significant impact on the data moving between the domains. In other words, as you consider things like filtering, aggregation and context processing, you want to push processing as far down the domain ladder as possible.

Consider these examples:

Simple Smoothing and Filtering

The simplest situation there probably would be a situation where some event causes you to want to get a unique list of tags back from *one* reader. Additionally, maybe you'd also like just SGTINs (Serialized GTIN values) for any pallet level tags that are seen. This is currently one of the jobs of the Edge – however it isn't a very big value-add as far as edge processing goes. As I indicated last time, look for this type of function to be pushed down to the reader domain as 'embeddable' edge-ware becomes available. One of the biggest wins for getting this done will be for RF devices where good application context automatically exists (i.e. we asked the user to scan a pallet ID – I know I only need look for pallet tags). This capability actually makes the use of on-board RFID readers feasible within the context of an RF application. As embeddable edge-ware makes its way into the mainstream there would be an opportunity to push simple smoothing and filtering down into the Reader Domain. Remember from my previous column, anything done at the reader level doesn't impact the network until the reader starts publishing tags to whatever is listening. The possible impact to the network of even simple filtering and smoothing at this level could be significant.

Aggregation and Logical Filtering

In the real world, this example of simple smoothing and filtering, while valid, is a very small set of what actually occurs – maybe limited to RF devices, in-line devices (on conveyors or other MHE), or other unique implementations. In many cases simply sampling one reader, even if it is smoothed, is not enough. Here are two typical examples: Multiple readers covering a large area and one physical reader split into multiple "logical" readers.

Figure 1 illustrates the fact that you may need to have more than one physical reader covering an area where you may be placing antennas both along the sides of the region and also over it. In this type of situation, it would not be uncommon to require at least two physical readers. Each of the physical readers can only report what they see, and each may only see a portion of the reads. What is required is the idea of a logical reader made up of these two physical readers. This is an area of significant value-add for the Edge. In fact, no matter how much intelligence is built into the readers or what embedded logic sits in them, only at the edge can we aggregate these readers to a level that makes *logical* sense.

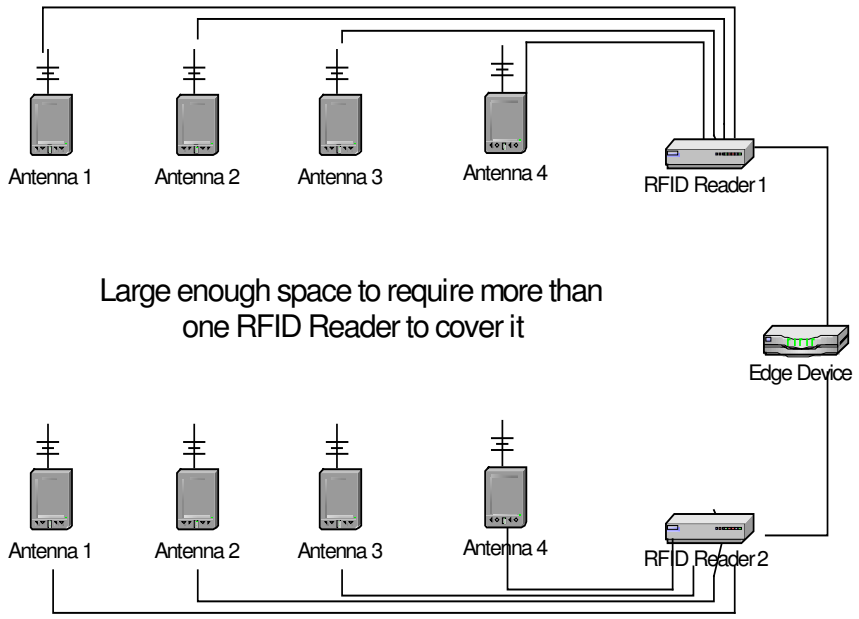
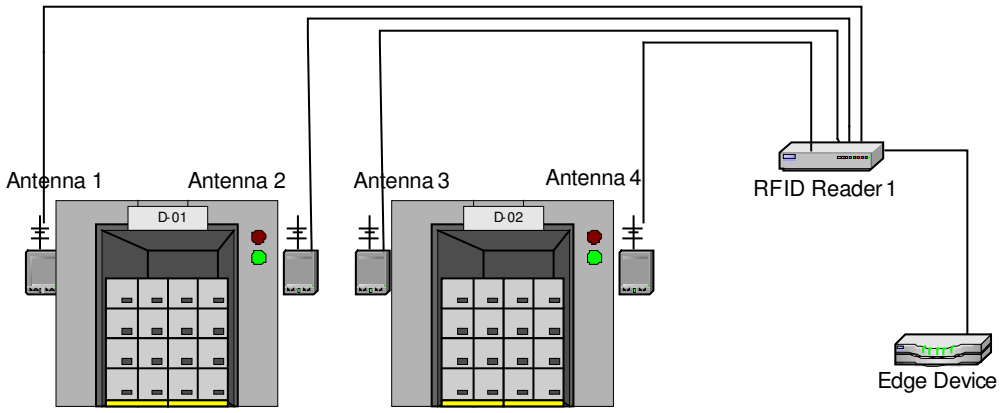


Figure 1

Figure 2 is somewhat the opposite problem. We need to do some logical filtering to really understand what is going on. In the figure, we are attempting to cover multiple outbound dock doors with one reader. The reader antennas are split – two covering one door, two covering another. You'd like to be able to treat Reader 1, Antenna 1 and 2 as one *logical* reader and reader 1, antenna 3 and 4 as another. Again, until embeddable edge-ware is readily available, this would be an area of value add for the Edge. Pushing this type of processing down to the edge domain limits the amount of inconsequential data that moves up into the execution domain.



Concept of *logical* readers

Figure 2

Context Processing

Context processing is one the real keys to RFID system success. It is critical that you understand your vendor(s) approach to dealing with context. Ultimately, understanding how the Edge and Execution domains have to contribute to and sort through context will be your roadmap to a scalable RFID system.

Context is an issue to the Edge and (potentially) the Execution domains. Obviously, the more context decision points you can move down the Edge, the better. While philosophically correct and absolutely true from a network scalability viewpoint, things are not always that simple. The execution system (especially a WMS) may actually be better suited to make certain context decisions. The key lies in the processes that are taking place, and who initiates and controls them. Context can be made up of both physical and logical elements. These are not mutually exclusive.

Physical Context Elements

As the name indicates - Physical Context lives in the physical world. Things like photo-eyes, scales, barcode readers, other types of proximity devices, etc. are all devices that can be used to create physical context. A simple example from figure 2: If you placed a simple presence/absence photo-eye in front of D-01 and D-02 you could tell which logical reader needed to be sampled. Photo-eyes and other types of proximity devices are the simplest mechanisms to implement since they are, effectively, triggers. Additionally, you might have a combination of these to create context. I recently worked on a conveyor-based RFID system where we used a photo-eye to indicate presence. When presence was indicated, we read the GTIN and slash-code barcodes. If the slash-code indicated product to tag, we'd encode a previously applied tag (inlay) on the fly. In this case, all the processing was done at the edge level with no need to communicate upward to the execution system until much later (post palletizer).

Logical Context Elements

Logical context is created by having an understanding of what is going on logically within an execution or business process. Examples of logical context elements are:

1. A reader saw 20 cases and a pallet tag – you may decide this indicates we saw a 'full' pallet based on a couple factors. First, we'd have to have knowledge that the 20 cases and the pallet tag all belong together (have been previously associated). Secondly, we may have a threshold limit that indicates a percentage of the cases that need to be read before we believe we've seen a 'full' pallet.

2. You have knowledge about the product in a particular location or on a particular pallet. Inventory location and pallet information is typically persisted in the Execution Domain. However, there are ways to make the edge aware of what is going on (to minimize the data movement between domains). An example of this is the idea of having the idea of a 'location' or 'device' tag placed on an execution system device (say a WMS fork truck). If you then drive the device, along with the inventory being carried by the device, through an RFID portal, you could then make the association back to the WMS device via the location/device tag. At that point you could make the determination, based on the device, the other tags read, what the WMS thought the device was doing – to do certain processes.

Now that we understand how to plan out placement of Readers, Filtering, and Edge Devices - and have a good concept of physical and logical context, the next question has to do with placement of the actual processing. The Edgeware vendors like to insist that they can do everything related to RFID. The

Execution vendors, on the other hand, are scurrying to show that they support RFID integration (in many cases using that same Edgeware). Even with all the discussion of pushing the processing down the domain ladder behind us, the question of where to do the actual work (the processing) may not be obvious. However, it is an extremely important question for you to answer. It *is* so important because ultimately all of the RFID equipment, Edgeware, and Integration must yield scalable business processes. Understanding the processing tradeoffs between the Edge and the Execution domains will empower you to challenge your Edgeware and/or Execution vendors to deliver the best solution to meet the needs of your business.

This question of "*where to do the processing?"* drives us to what we call the Push or Pull decision. Do we *push* the data (and some of the context) up from the Edge, does the Execution System *pull* the information from Edge Devices, or is it a combination? We'll explore this in our next column.

Delivering ROI without the "blah, blah, blah"

About the Author

Mark Fralick is president of ROI Solutions and a SuppyChainDigest Contributing Editor. Having operational, implementation and software development experience, along with detailed understanding the workings of software and services companies, puts ROI Solutions in a unique position to stand strongly on the customer's side in the battle for ROI in system selection and implementation. Learn more at www.GetUsROI.com, or email Mark Fralick at Mark.Fralick@GetUsROI.com.

About SupplyChainDigest

SupplyChainDigest™ is the industry's premier interactive knowledge source, providing timely, relevant, in-context information. Reaching tens of thousands of supply chain and logistics decision-makers each week, our flagship publications - *SupplyChainDigest* and *SupplyChainDigest – Logistics Edition*, and web site (www.scdigest.com) deliver news, opinions and information to help end users improve supply chain processes and find technology solutions.

For more information, contact SupplyChainDigest at:

937-885-3253
www.scdigest.com
email: info@scdigest.com