An Optimized Service Oriented Middleware Design for RFID Enabled Solution

Prasenjit Bhadra, IBM India Pvt. Limited, India; E-mail: prbhadra@in.ibm.com

ABSTRACT

Introduction of RFID at the Enterprise level has accelerated the automation of data interchange at various data points across the entire supply chain. The proliferation of RFID tags and readers will require dedicated middleware solutions that manage readers and process the vast amount of captured data. On the other hand, Service oriented architecture (SOA) represents a style of information systems architecture that enables the creation of applications, built by combining loosely coupled and interoperable services [1]. SOA Evolution promises a smooth integration and consolidation of large complex RFID implementation within large enterprise system architecture. The paper aims at the value propositions to the recent practices on such RFID middleware development using the SOA centric design approach and proposes model architecture along with the framework components for such optimization.

I. INTRODUCTION

The recent drift of innovation in an enterprise space is approaching towards two distinct spectrum of technological trends: turning massive amount of data into an operational intelligence by integrating Sensory /classical mobile devices at the edge of the enterprise network and on the other hand, building more adaptive and extensible application environment by addressing need based services with in an enterprise-wide heterogeneous Infrastructure. The essence of the "paradigm shifting" practices reveals that the collaboration of such emerging technology and a handshaking of conceptual abstracts would plot the road map for future solution offerings in an enterprise space. The widespread adoption of RFID requires not only low cost tags and readers, but also the appropriate networking infrastructure [2]. The constant effort of delivering solutions to such emerging areas has brought more focus into developing middleware and frameworks – that promises an application-agnostic offering for a diverse set of requirements. Functionally it manages readers, filters and aggregate captured RFID data and delivers these to the appropriate consumers. The platform also facilitates domain specific/independent integration, adaptation of a chosen set of RFID peripherals, combines the RFID data with application logic, and generates application-level events.

The main involvement of this paper is a middleware design approach delivering open standards along with the solution architecture of such middleware addressing new challenges of Automatic data collection using RFID and seamless integration over SOA platform. The governing idea of such system is primarily built on recent strategies on RFID enablement and extending service delivery approach demanding operational efficiencies within enterprise-wide "On Demand" space.

This paper is organized as follows: Section II focuses on the motivation that drives such design considerations. Section III depicts the principle of a scalable RFID middleware design and the value propositions. Section IV describes the proposed architecture of subsystems within the solution domain and integration methodology. Section V is a brief description of the reference implementation and Section VI illustrates how the service oriented approach addresses the recent challenges on solution design and integration. The paper concludes with a summary of the overall approach and discusses future directions in Section VII.

II. MOTIVATION

The recent pilots and rollouts of RFID implementations reveal that as adoption grows, the realization of benefits and scope of operational areas also grow. In industrial automation and supply chain management domains, RFID technology holds the promise to eliminate many existing business problems by bridging the economically costly gap between the virtual world of IT systems and the real world of products and logistical units [3] .The following points illustrate the objective of an extensible RFID middleware design suitable for wide range of implementations:

Leveraging opportunities provided by RFID to both private and public sectors involve understanding of the potential applications and the different business cases for the technology and its applications, along with their limitations and current challenges, to develop forward-looking policies [4]. Therefore the successful outcome of RFID enablement potentially depends on a set of technology advancement and smooth integration with the effective process change in an enterprise environment. The RFID implementation on top of the existing infrastructure would not highlight much on discrete business process issues and high impact on existing enterprise application architecture.

Item level tracking and asset management using RFID generate enormous amount of data which needs to processed, stored and transferred to an appropriate business system. Studies have been made to baseline the standard practices on data management and application component design to overcome technological challenges from passive RFID based solution. The current practices of solution development should also extend a mind-share on how RFID data can be interpreted in a given business context and turned into the corresponding application events for an optimized service offerings within entire enterprise application domain.

The solution providers have been working on standard set of applications /middleware design to cater most of the requirements evolved from the recent business practices. Such solutions could effectively accomplish the objective of integrating RFID with the operational environment but fails to anticipate the future business problems and collaborating on the solutions for a complex rollout.

SOA has emerged as one of the best alternative to turn the existing and new applications and data sources into an agile IT infrastructure that can help organizations meet changing business needs. This also facilitates reusability of components and services over time. I have tried to propose an architecture of the RFID middleware which extends the bandwidth of solution offerings over SOA centric design. Such optimization promises a wide spectrum of service delivery in a cross platform and cross domain automation where an extensible design not only targets to minimize the concerns as mentioned above but also try to eliminate technology specific constraints.

III. DESIGN RATIONALE AND OPTIMIZATION

Pilots carried out in various industry verticals have potentially extended the expected business values out off such automation. From the Architectural standpoint, an RFID middleware should not target to accomplish a specific goal or project requirements rather facilitate the scope of delivering anything beyond what is featured at the POC/ Pilot phase .

The recent practices and implementations have envisaged an ideal design approach of RFID middleware. The lessons are:

- Realization of early benefits would grow complexity in the incremental RFID enablement as the roll out would embrace more challenges beyond the traditional integration .Therefore a top-Down approach to design will not be well suited in the long run.

- The scalable design of such framework should be holistic to the entire enterprise and embed its capabilities in the overall system architecture.
- A device agonistic abstraction creates a common standard of data exchange protocol for the device service layers.
- Seamless integration of filter events, monitoring/logging and event routing should extend a common standard of event processing capabilities for processing streams of RFID and/or sensory inputs (Data/signals) into business events.
- A generic and scalable enterprise application framework in a multilayered architecture creating distinct service end points would serve data /information interchange and rule checking over the enterprise service bus (ESB).
- The solution framework would opt for a modular approach and well-defined separation of concern for the information interchange and smooth integration with the existing enterprise architecture.

The ongoing development of real-time asset tracking solution has widened the scope of distributed data aggregation and processing over an SOA layer to address the mentioned optimization sentinels.

IV. SOLUTION ARCHITECTURE-APPLICATION COMPONENTS AT VARIOUS LAYERS

The proposed architecture for the middleware design points to different challenges and the corresponding research approach in developing a RFID middleware to provide a seamless environment with in the enterprise network; moving data from the point of transaction to the enterprise systems. An Extensible Event broker model and robust messaging infrastructure constitute the backbone of the solution design. It encapsulates intelligent data processing and routing, interoperable solution components on top of business integration layer and standard communication protocol for data interchange. The multi layered architecture can be customized for specific operational need by configuring services and subsystems as per requirement. The basic design is intend to achieve a reasonable combination of core infrastructure and packaged application features, including device management, integration, data management, and generic business logic .The application layers are broadly structured as follows:

A. The Device Service Provider Domain

The sensory devices are controlled through the processing unit connected at the edge of the enterprise network. The service components primarily provide a device abstraction and logical filtering/aggregation services, supporting the analysis of volumes of data provided by the readers, applying local decision making and intelligence. The group of applications termed as Edgeware or Edgecontroller, are built on a standard framework which can be extended for specific service agent development. The contract between client or consumer and service is interface driven. Discovering and binding for the corresponding implementations are done through the framework layer. Agents are softly coupled and primarily rely on standard messaging protocol over an asynchronous public-subscribe model for information interchange. Thus the runtime components embrace an observer-observable pattern with in the processing environment. There is an external configuration /XML file which contains the declaration of service dependencies with in the runtime environment.

The functional features of the edgeware application environment are as follows:

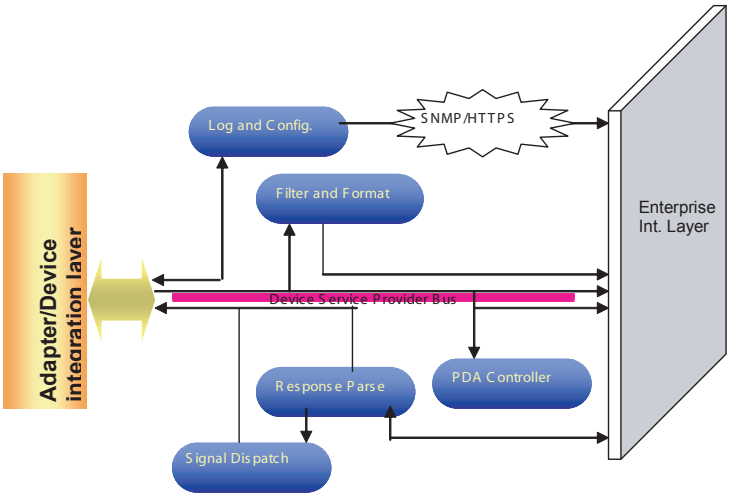
- The Device Integration Layer creates a façade for the transport and device protocol. This layer interprets the incoming signals/ data lists into a common XML format. The application components in this layer are also capable of translating standard commands into device specific instructions. A single interface contains a list of functions to transfer and execute standard read/write, I/O operations and device management command list. As the data/command formats are not yet standardized the implementation of the same adheres the dedicated supports from device/Vendor libraries and APIs in the heterogeneous reader landscape. This is a loosely coupled service agent which can be customized for various readers.[5].
- Depending upon the nature of implementation and volume of data, the notification latency should be configured on top of the data broadcasting layer. For a large scale implementation(i.e. Dock door receiving ,Real-time asset tracking using active tags ,Smart Shelves) a filtering and smoothing service is configured for minimizing the redundancy of the captured raw tag-reads and funneling the unique reads in a predefined format(Text/XML).The service agent additionally performs the checking of the Data format of the incoming raw tags considering either standard specifications(e.g. EPC SGTIN -96 , ISO 15693 , ISO/IEC18000-3) or proprietary data structure as per need .Such optional filter service at the local execution environment mitigates the risk of unwanted injection of bytes from the physical world. Similar filter service would be extended with respect to specific portal/Reader identifiers. Since a single Edge Controller is operational for multiple readers at a particular location , the service can be configured for a specific set of readers or logical grouping which would be considered for a specific application context[6]
- Each edge unit comprises of a Tag-Read- Handler Router -an agent broadcasts the formatted tag list to the centrally located Data aggregation service layers at the enterprise space. The performance of this service layer is optimized through a set of Latency parameters viz. 'paramPublish_Interval', 'param-Purging _Interval' at the enterprise space. On the other hand, relatively simpler application environments or solutions handling low volume of data but need to respond immediately to local interaction with the physical objects (i.e. Data collection using hand-held reader and PDA, WIP tracking, Security monitoring, vehicle tracking and fleet management etc) reduce the overhead of the filter service by eliminating the same within the runtime environment or consider a short notification latency that is comparable to the observation latency and receive immediate notification from the device layer[6].
- The edgeware establishes a common secure message subscription channels with the enterprise layer for data/signal event exchange asynchronously. The router agent publishes the XML messages contain the Event source /Destination ID, Timestamp along with Tag-data elements which are subscribed by the consumers at the enterprise layer to accomplish specific task.
- There is an SNMP [7] agent running on each Edge controller application environment. This service agent is exposed through a secure Web service for the device /infrastructure management. The centrally located Management Information Base with in the enterprise application layer queries for current performance statistics periodically in a systematic fashion using SNMP traps (Synchronous calls). Each Device Agent extends a worker thread which delegates the device/peripheral status to the SNMP agent The Agent persist the details in an XML format, which is forwarded upon request from the management station.
- The response parsing agent carry outs optional service management tasks locally when a notification is received from the other device agents and response/instruction received from the enterprise management base. The operations like Audio visual alarms for any failure, Sensing tagged physical objects in proximity and switch on the reader, maintaining a temporary persistence (using in-memory DB or XML) for the Tag reads if the network is down etc for immediate recovery and faster response at the local site.
- The mobile/handheld devices operational with in the edge domain use an additional GUI based controller (developed on top of standard GUI framework) for data collection trigger, formatting, Manipulation of the data elements and persisting/routing the same with in the application environment.

B. The Enterprise Integration Domain

The enterprise integration is the intermediary between enterprise applications and the Edge Domain. It compiles business sense of RFID read information and enables automatic decision making. The application infrastructure is built upon Service delivery framework defining so called plug-in to the real world. The application tier offers the following features:

- The Enterprise layer persist the configuration details and a logical mapping of the readers /sensory hubs within the business environment data points. The device infrastructure operational need to be registered with the system with respect to such predefined mapping. For an offline mode of operation the same details need to be validated prior to the data aggregation at the enterprise space.
- The Data interchange layer at the top of the enterprise space provides a common gateway for standard publish/subscribe mechanism (centrally Located) handing dataflow across the sensor hubs /handheld terminals and enterprise layer. The unidirectional communication channels are used for subscribing incoming tag lists and other operational inputs. There are standard set of XSDs to validate the data content and format of the incoming Tag-lists. For any non compliance, this service layer generates alert messages as a notification on

Figure 1. Functional components: Device service provider layer



the subscription feedback. The local application can subscribe to the same using the response handler agent to take immediate action (send message via alert console or switching on the Audio-visual alarms).

- The valid Data/Tag-lists subscribed over the enterprise gateway are processed by the data aggregation subsystem analyzing the source/content, are delegated to the respective event generation layer to process the same. The standard XML list contains sensor/RFID event specific inputs is processed as a generic message format and delivered in enterprise application infrastructure. The subscribers consume the message of its interest (RFID event specific to Dock door receiving, choke points, audit details or read counts etc.).Such subscription is structured through a management console considering the reader/sensory hub configuration/mapping and decentralized operations for a specific set of Data elements. A dynamic delegation pattern is incorporated in this layer to bind the message selectors with in the runtime environment. This service is a construct of the Application event emitter framework, which process the incoming messages and check the rules applicable for source of sensory events and specific problem domain.
- The implementation of a rule engine on top of this emitter layer extends an adaptive execution environment for the available service components. The predefined rules act on a set of subscribed data elements to generate Application level events. Such rules are derived from the custom process maps applicable for specific business context. The administrative consoles are available for add/modify the rules applicable at various service layer of the predefined event emitters. Hence the integration layer can be leveraged for numerous distinct process execution and service offerings and provide information directly to "line –of business" applications. The set of rules would be applicable for correlation, precedence checking and complex mapping of the data items from within the event repository. The service layer also facilitates a real-time state transition updation and deletion on the base events.
  The Application Level Events(ALE) generated by such event emitters are persisted in ESB for the consumers interested in specific business operation .The consumers can be any business service, pluggable subsystem which can access these data for raising notification /alert , update backend system and/or constitute an information hub for the business events in an organized report format.
- Reader devices which are part of a secured network are registered and go through a process of device authentication on an ongoing basis to prevent "reader spoofing" (e.g. unplugging a reader and plugging in a (e.g. unplugging a reader and plugging in a laptop which creates a false data stream which is meant to look like it comes from a reader) using a centrally located Administrative and monitoring service[8].The solution architecture is designed such a way that it can protect both stored data as well as data that is in transit. The system should also extend comprehensive management of the readers and other peripherals constitutes RFID enabled infrastructure. .The application component act as an SNMP management station analyzing SNMP traps send form the distributed device infrastructure.
- The centrally located management application service layer includes the product details lookup , rule management, Device infrastructure management , Log details and views ,analytics and other reports/enterprise content solutions

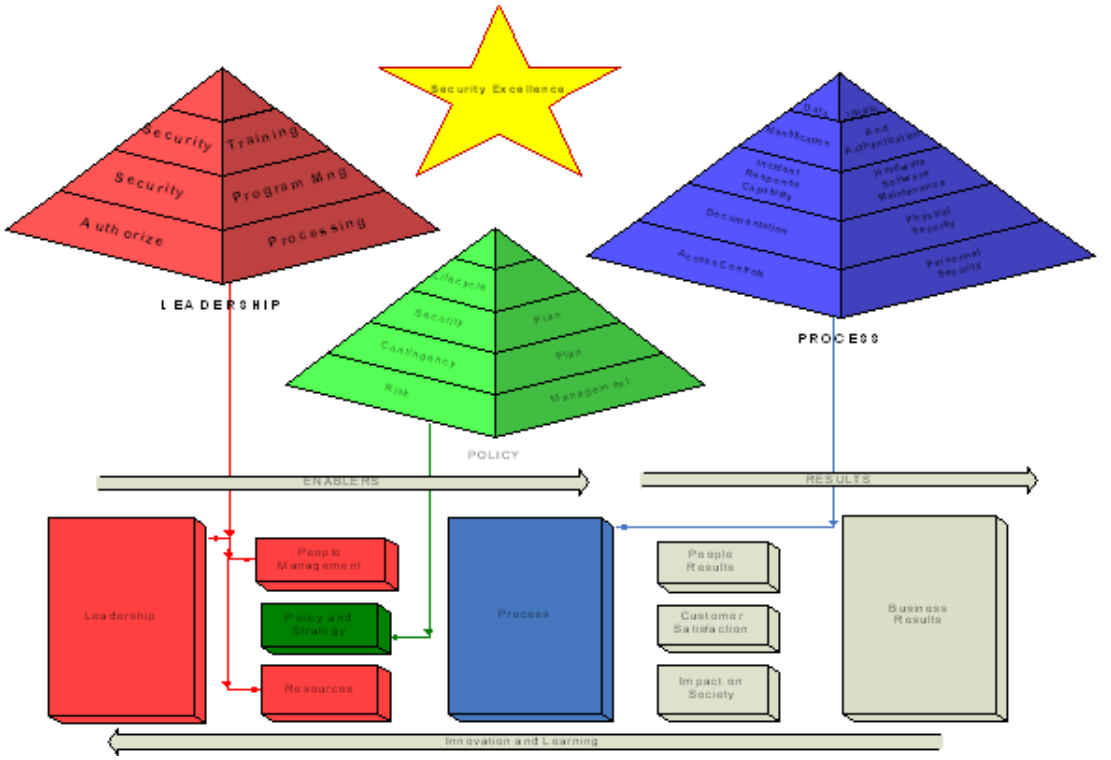
The Object Directory Domain associated with management application services running in the server and other handheld terminals is responsible for user authentication, authorization, and data encryption practices in accordance with a security and privacy management policy.

V. THE REFERANCE IMPLEMENTATION

The proposed design has been realized through an asset development activity in our lab. We have chosen OSGi application framework (SMF on top of IBM J9 environment) to build loosely coupled bundles of the solution .Each bundle exchange signal data over MQ Everyplace messaging infrastructure .The controller uses Message Queue Bridge for store/ forward signal/Data transfer over the network. In offline mode, DB2e is used for local persistence of the tag-lists. The SNMP client bundle updates the network status to the controller to route/persist tag reads. The basic SOA centric design promises a dynamic modularity for the edgeware application environment and accomplish following points of interest:

- The evolving standards/hardware and business requirements demands new generation of devices and sophisticated process automation .with in the enterprise space. The service components thus built on such framework can afford such enhancements following the evolutionary trend and helps to optimize the runtime environment by activating /deactivating services and configuring their service delivery behavior.
- The runtime environment should be integrated with large number of devices viz.; PC, Microcontrollers, OS embedded in the stationary readers, and micro

Figure 2. Functional components: Enterprise integration layer



environment of the handheld devices/PDAs. Therefore the framework components are built on J2ME runtime that supports diverse set of environments.

The server side components are built on WebSphere platform with DB2 as a backend. A service component is built on this platform to receive inbound messages as an MQe Gateway. The application module extends the Jeline event infrastructure and uses pub-sub broker on top of MQ series which exposes a Standard Event Infrastructure and management.

The application services builds on the J2EE layer .They are used for storing devices and other agent configuration along with configuring rules associated with the data aggregation, visualization and correlation. The asynchronous reads and other sensory inputs are parsed by the dedicated event emitter at the event processing layer based on such preconfigured rules and the same is persisted in the ESB layer as ALE.

A configurable supervisor service monitors the ALEs to update, Logs transitions and consumptions of the events in the back end database and sending notification to other systems using Response handler service components.

Some generic application service layers are built to subscribe the events and query the data objects and application level filter to look up the ESB contents for the specific needs.

There are other standard application services built for the health status check using secure SNMP trap, device /user authentication/registration and device agent configuration download over LDAP.

VI. SOA ENABLEMENT- THE VALUE PROPOSITIONS

In order to comply with the typical SOA centric middleware design the proposed framework focuses on three fold approaches:

- The sensory devices constitute the edges in enterprise infrastructure, should promise unique services deliver behavior that transcend the underlying technology for data collection using a common standard event model (XML based) over the network and visualized by enterprise application layer. Such event model is also adaptive to the connectivity of the sensory devices; logging performance is in an assorted device infrastructure and notification through audio/visual alarm and signals.
- The enterprise space extends a managed/scalable environment built on service oriented container/content model. Services comprise of standard event bus, complex calculations and analysis on composite business events using correlation technology and rules engine, which assure potential means of semantically rich communication between systems targeting various business operations. Apart from the data acquisition and translation of the business events, the enterprise application creates a centralized device management service and reduce the on site administration hassles.
- The integration layer relies on synchronization services with the legacy system .The dynamic event generation and logical data aggregation at the enterprise space is accessed through a Hub and spoke model on top of the Synchronization layer. The wrappers built for such synchronization services thus eliminates the hard coded point-to-point interfaces between the middleware layer and multi-platform heterogeneous enterprise IT infrastructure.

The architecture of such focuses beyond a point solution approach, reducing the overhead of replicating information between the Middleware infrastructure and other multitude enterprise systems .smoothen the transition of the events generated at the physical layer to the structured business events. It promises a readily available, reliable scope for real-time asynchronous Event propagation and state-based decision making.The mentioned design approach uses standards-based technologies on an extensible platform to ensure that it's well positioned for the future.

VII. CONCLUSION

Integration of RFID based data collection and automation is a cross functional exercise. Organizations need the freedom to implement RFID in increments and

the flexibility to re-engineer processes that are RFID-dependent. This should also minimize the cost and time to frame a new architecture. The proposed approach targets primarily on Service-oriented architecture (SOA) and features both services and events which could deliver the desired scalability and performance.

The proposed solution architecture is harvested from encouraging experience in designing and implementing RFID solutions for various industries. The optimized middleware design would promise an RFID- technology agnostic and holistic distributed platform for an enterprise wide integration and process automation. It targets to those requirements by exploiting the Service oriented application components, easy-to-use Event Infrastructure for Business process integration seamlessly with wide range of enterprise solutions.

The proposed solution demands continuous supports for integrating new breed of readers as the vendors don't adopt a standard protocol for device –to-application integration. However the recent researches in hardware and firmware development has evolved mature device specification which would have standard implementation for the communication protocol along with some of the built-in services of the device integration .With the advent of such maturity the Edgeware application would be upgraded to more slandered adapter with the evolution of mature specification in future and would be functional as proxy to the enterprise integration layer focusing more on local decision-making services.

At the enterprise space the custom service layers are developed to emit application level events with respect to the applicable rules for various business needs .The current research is targeted on using BPEL for service orchestration or process choreography to design much more flexible service layers and seamless integration.

Therefore considering the potential areas of improvement in the enterprise scale RFID implementation the loosely couple design can be optimized for applications of any size and can be leveraged with the realization of the multi-dimensional impact of the growing need for such automation.

VIII. REFERENCES

[1] Service oriented Architecture at: : http://en.wikipedia.org/wiki/Service-iented_architecture

[2] S. Sarma, D. L. Brock, and K. Ashton. The networked physical world – proposals for engineering the next generation of computing, commerce & automatic identification. Technical Report MITAUTOID- WH-001, MIT Auto-ID Center, 2000. Available at: http://www.autoidcenter.org/research/MIT-AUTOID-WH-001.pdf.

[3] E. Fleisch and M. Dierkes. Ubiquitous computing:Why auto-id is the logical next step in enterprise automation. Technical Report STG-AUTOIDWH-004, Auto-ID Center, 2003. Available at : http://www.autoidcenter.org/research/STG-AUTOID-WH-004.pdf.

[4] Organisation for Economic Co-operation and Development : Research abstract RADIO-FREQUENCY IDENTIFICATION (RFID): DRIVERS, CHALLENGES AND PUBLIC POLICY CONSIDERATIONS . Available at www.oecd.org

[5] P .Bhadra . RFREAD :The generic framework for RFID reader Adapter Publication at International conference EAIT 2006 , CSI .Page 357-360.

[6] RFID middleware design - addressing application requirements and RFID constraints Christian Floerkemeier and Matthias Lampe . Available at : www.vs.inf.ethz.ch/publ/papers/floerkem-rfidmi-2005.pdf

[7] Simple Network Management Protocol . Available at : http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.pdf

[8] An Architecture Framework for RFID by Steffen Schaefer, IBM. Available at : http://www.lac2004.nl/docs/fvbg2hdsb83/Track9/S.%20Schaefer.pdf