Executive UPDATE
CUTTER CONSORTIUM
Access to the Experts
BUSINESS & ENTERPRISE ARCHITECTURE

The Role of EA in the IoT

by Charles W. Butler and Stephen C. Hayne

Enterprise architecture (EA) emerged as a strategy for specifying a logical blueprint that defines the structure and operation of an organization. The aim of an enterprise architecture is to determine how an organization can most effectively achieve its current and future objectives. EA aligns technology architecture capabilities with business architecture processes and services. Historically, this alignment embraced monolithic structures, including operational systems and data warehouses supported by relational databases and flat files. With today's Internet of Things (IoT) and all the connected devices it involves, solution design offers new opportunities, even though designs can be more challenging. The emerging solution environment has the following characteristics:

- New services and processes for business commerce, lifestyle, social interaction, culture diversity, and environment management
- A broad range of user platforms such as wearables, smartphones, and a multitude of appliances
- An expanding use of sensors that replicate human sight, hearing, smell, taste, and touch, and implement real-time concepts
- Better timing dependency among and between services and processes using large data volumes and new data analytics to drive process decision making

This *Executive Update* explores the impact of the IoT on traditional business and technology architectures and the role of EA as an effective methodology for developing and implementing IoT strategies. We examine business architecture and how it integrates IoT-driven processes with traditional processes.

The IoT is characterized by "Things" — many of them small in physical size, which can connect to other Things, generating a large network of collaborative IoT and non-IoT Things. In this paradigm, business process management (BPM) extends to accommodate innovative workflows that are more functionally robust with "Thing-to-Thing" linkages.

To manage a more diverse business architecture, we outline several EA techniques to ensure appropriate decision making and proper accountability during the execution of IoT strategy. Since business services and processes are driven by information fed from the infrastructure and data architecture, we also assess compute node (Spark) and storage node (NoSQL) technologies as platforms for managing non-IoT and IoT data. This examination identifies selected design principles that guide solution mapping of cloud compute and storage processes and services. We also answer questions such as:

- How does EA map to compute and storage solutions?
- Which non-IoT and IoT business services and processes should drive compute versus storage technologies?
- How will cloud resiliency and failure properties impact EA and IoT services and processes?

To illustrate IoT EA concepts, we utilize a Thing example based on smart appliances; in our case, a smart refrigerator. Our illustrations use a theoretical refrigerator as a Thing within the smart appliance sector. Smart appliances are similar to many other new "smart" products introduced in the market.

The IoT

The IoT is an increasingly important force in business operations and social lifestyle. According to a *Forbes* article by Bernard Marr, the market for wearable devices grew 223% in 2015, there are about 5 billion connected Things, and the profile will rapidly expand by 2020. Marr predicts the near future will show:

- Connected Things will reach 50 billion.
- 250 million vehicles will connect to the Internet.
- 10.2 million units of smart clothing will ship.
- The RFID tag market, used for identifying and tracking objects, will be US $21.9 billion.
- Many household devices, from smart thermostats to smart refrigerators, will be connected.

The IoT is a disruptive technology forcing businesses to modify their existing EA to accommodate the coarse granularity, diversity, volume, and speed of data of the IoT. The fact that the IoT increases the complexity of business data organization heightens the importance of EA. Thus, businesses should plan, design, and implement business solutions guided by their EA strategy.

Emerging EA

Enterprise architecture is a blueprint for the deployment of IT resources that support business services and processes. It is a high-level view of six fundamental domains: service, process, information, application, data, and infrastructure. As shown in Figure 1, EA aligns new technology capabilities with innovative business architecture processes and services. In the figure, new architecture elements associated with the IoT are accented in red. In the business architecture with IoT, commodity services with standard processes and operational data are augmented with custom services delivered through configurable processes. These processes provide on-demand information tailored to user preferences or customized processes delivering unique services associated with multiple, diverse connected devices. Due to larger consumer markets, more configurable processes will be required to meet broader customer preferences.

In the emerging EA in which the IoT is a force, the architecture is more diverse and complicated. New data platforms, including Hadoop, NoSQL, and cloud data, will be used to house huge volumes of data. The IoT will enable people in their daily lives to access custom services implemented using configurable processes. For example, a smart refrigerator user can set food expiration policies and rely upon automated leading signals before actual expiration dates are realized. A configurable process minimizes food loss and reduces consumer grocery cost. In another example, a food manufacturer can signal consumers if a product has been found to pose public health risks (e.g., the possible presence of e. coli) to minimize consumer exposure to this risk. With real-time design, a food producer might search purchased smart refrigerator data, identify the location of tainted food, and send an urgent message to affected consumers, which can be broadcast among a cluster of product users, increasing the speed and reach of the message within the targeted customer sector. These innovative workflows are enabled through smart refrigerator integration with applications and data located across selected infrastructure platforms.

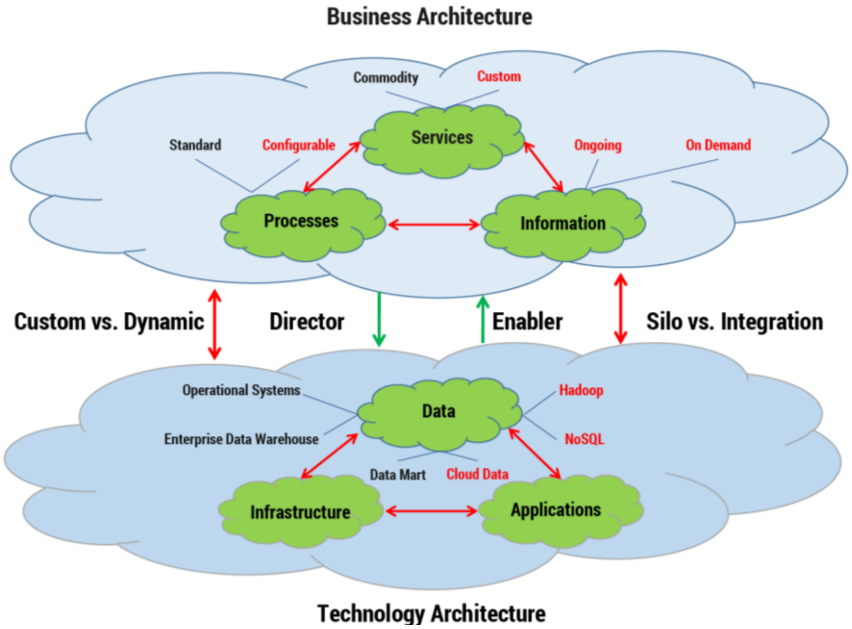


Figure 1 — IoT-enabled enterprise architecture.

High-Level EA and Techniques

Figure 2 illustrates a high-level view of building an enterprise architecture. The techniques and methods for non-IoT and IoT Things are essentially the same, even though it is probably more difficult to architect IoT's small granularity, diversity, and volume metrics. Business strategy is driven by external and internal forces; enterprises must consider competition and market conditions when formulating business strategy. Furthermore, regulatory agencies force a company to operate under certain conditions that must be assimilated into established business strategy. In this business model, six EA techniques and methods are utilized to implement and sustain EA:

1. **Reference models** — taxonomies that provide standardized categorization for strategic, business, and technology models and information.

2. **Governance** — processes for connecting the business vision and project execution with the current and future enterprise reality.

3. **Standards** — best practices that ensure services align with company strategy.

4. **Principles** — general rules or guidelines for designing a system.

5. **Key process indicators** — measures of outcomes demonstrating EA benefits.

6. **Security** — access and compliance management configurations and practices based on industry recommendations for secure and interoperable identity management and access control; these configurations and practices ensure security capabilities and plan a roadmap to meet the security needs of business.

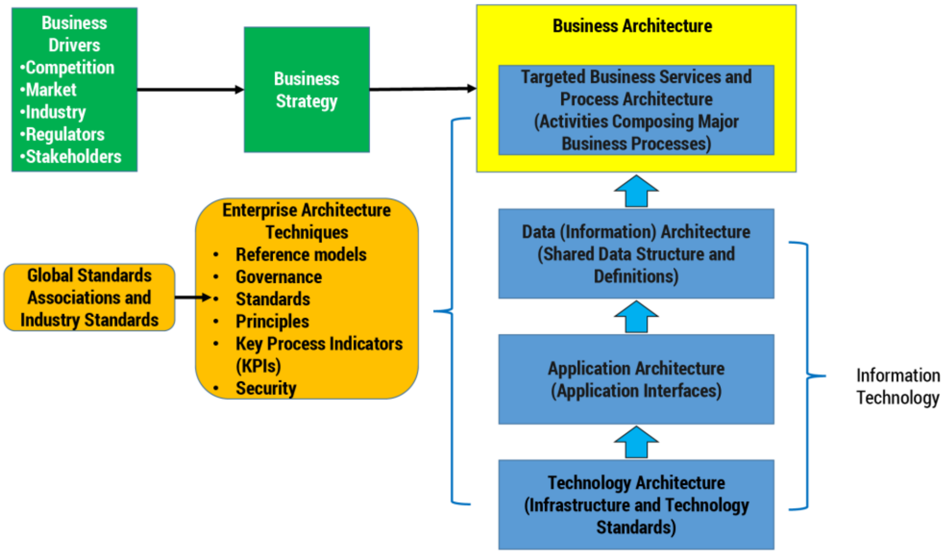

Business Drivers
•Competition
•Market
•Industry
•Regulators
•Stakeholders
Business Strategy
Business Architecture
Targeted Business Services and Process Architecture (Activities Composing Major Business Processes)
Global Standards Associations and Industry Standards
Enterprise Architecture Techniques
• Reference models
• Governance
• Standards
• Principles
• Key Process Indicators (KPIs)
• Security
Data (Information) Architecture (Shared Data Structure and Definitions)
Application Architecture (Application Interfaces)
Technology Architecture (Infrastructure and Technology Standards)
Information Technology

Figure 2 — Building an IoT enterprise architecture.

The outcome in using these techniques and methods is a concrete set of artifacts that guide a company and its stakeholders. It is not enough to see the artifacts, however; they should be used as a blueprint for how to build Things, focusing on the relationship of IT (director and enabler) and business domain components.

As the IoT becomes a greater part of strategy, an enterprise may require an expanded EA. IoT complexity could result in changes to services, processes, and information. These changes will require architectural adjustments to enable IT elements, including applications, data, and infrastructure. The outcome from this synthesis is a new EA, complementary standards, tuned governance processes, and EA-measuring processes. The business and technology architecture may or may not be a layered model as illustrated in Figure 2.

There are numerous architecture frameworks sponsored by recognized groups, including TOGAF and the US Department of Defense Technical Reference Model (DoD TRM). Figure 2 illustrates a layered framework to reinforce the layered architecture in Figure 1. The technology architecture in Figure 1 is layered in Figure 2 to demonstrate that the three IT domains are the underlying components that enable the business architecture (with its three domains) and respond to directives originating in the business architecture. The following sections describe several of the EA techniques in an IoT context.

IoT Business Service Reference Model

A reference model is the most important EA tool. It is a schematic of the structure of selected architecture components. A reference model can be constructed for business components as well as technology components. There are no hard and fast rules defining how many reference models are needed to define enterprise architecture or what the format of reference models should be. The US federal government's EA framework identifies six reference models:

1. **Performance reference model (PRM)** — a standard method for performance measurement providing a common approach to performance and outcome measurements

2. **Business reference model (BRM)** — a map of an organization's lines of business and business activities to services within and between federal government organizations

3. **Data reference model (DRM)** — a flexible common framework for effective sharing of government information across organizational boundaries that increases integration and reuse opportunities and supports semantic interoperability while respecting security, privacy, and appropriate use of that information

4. **Application reference model (ARM)** — the system-/application-related standards and technologies that support and enable the delivery of service components and capabilities

5. **Infrastructure reference model (IRM)** — the network-/cloud-related standards and technologies to support and enable the delivery of voice, data, video, and mobile service components and capabilities

6. **Security reference model (SRM)** — a taxonomy for the establishment of security controls in an architecture as well as a scalable, repeatable, and risk-based methodology for addressing information security and privacy requirements within and across systems and agencies

In using reference models — such as those in the federal government's EA approach — users, managers, architects, and developers can share a common framework for services, processes, information, applications, data, and infrastructure. It is important to know that a reference model stimulates integration among services and technology by refocusing vision away from operational silos. The reference model also serves as a common dialogue and language source for business participants. A reference model facilitates service-level agreements as business practices embrace outsourcing and cloud services. The completion of a reference model is challenging because there is a need to find consensus among many potentially contentious parties.

Figure 3 illustrates an enterprise service architecture for our smart appliance company example. This model is the foundation for both its business and technology services. Our smart appliance service reference model has a business service scope containing Manage Owners, Operate Product, Manage Integration, and Data Analytics services. Each service contains subservices that compose the aggregate service. Consider the Manage Owners service: nine subservices are specified. For example, when managing owners on our smart refrigerator, the owner might utilize the Customization Service to personalize the main menu for each family member. The business service spine contains four subservices that span Manage Owners, Operate Product, Manage Integration, and Data Analytics services. For example, Customer Help Line is a subservice integrated with Manage Owners, Operate Product, Manage Integration, and Data Analytics services.

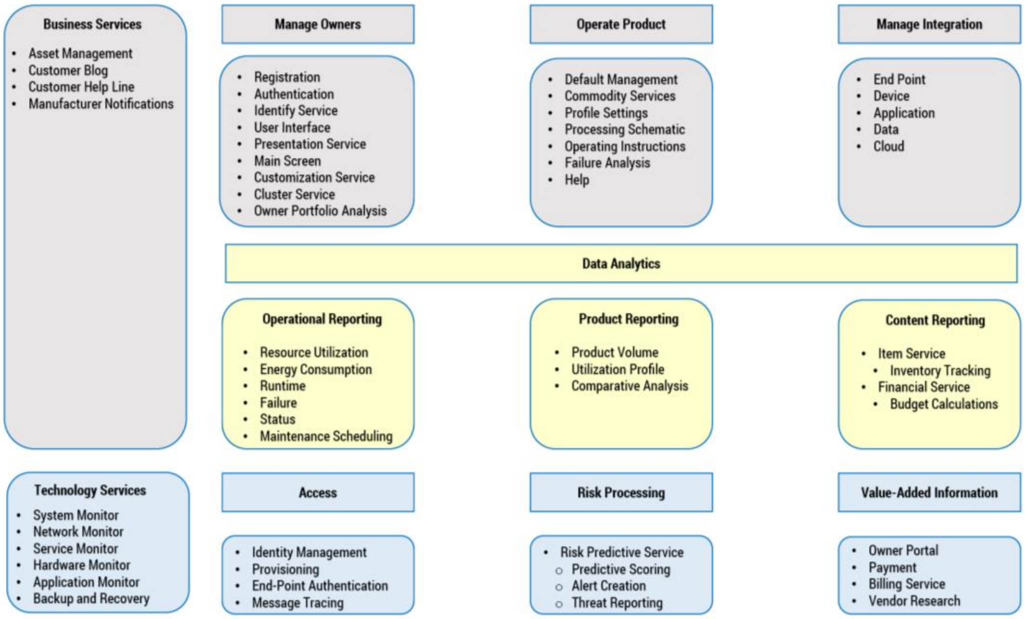


Figure 3 — Example of a "Thing" business service reference model.

As new or modified IoT devices become components of products or services, the Thing service reference model supports architectural analysis in the business services view of the overall EA. For example, if the RFID tag technology penetrates the packaged food sector, new IoT services are viable. In reviewing the Operate Product service, there is no subservice for inventory management. RFID-enabled technology could support a new subservice, called Inventory Control, which tabulates when food products are placed in the smart refrigerator, calculates the frequency and duration of food product usage, establishes reorder points for food product replenishment, and issues reorder instructions based on usage analysis.

IoT “Thing” Reference Model

The DRM is a framework whose purpose is to build consistency and to enable information sharing and reuse across business operations. The model is the standard by which data may be described and classified. Traditionally, business data organization has been dominated by files and relational databases. Today, Thing data generation is more unstructured than structured. Things are the sources of voluminous data, and Thing structure is an emerging concept. Figure 4 shows our proposed logical structure of a Thing, which could populate an enterprise DRM; the structure is a schema that describes and defines properties for data content and sharing. The Thing nucleus provides a standard identity so that it is known and shared within the enterprise. Characteristics can be both structured and unstructured supporting the big data property of Things, allowing for RDMS and NoSQL physical implementation. In this reference model segment, the interface property addresses Thing integration with the outside world. A Thing might interface with another Thing (T2T), a service or process (T2S/P), an application (T2A), or data repository (T2D). A logical Thing model is the basis for service and application design.

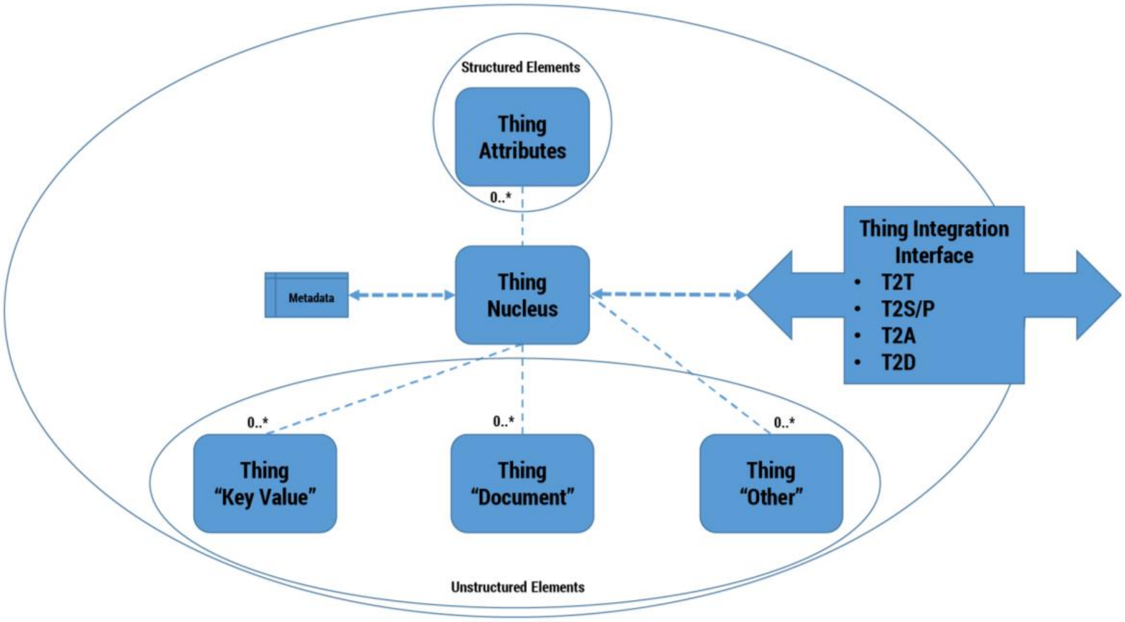


Figure 4 — A logical Thing data reference model.

When implementing designs using the DRM, consider the allocation of services and processes to available processors. For instance, our smart refrigerator is a single instance of the appliance Thing. If our smart refrigerator is connected to the Internet via a service provider, design decisions must balance processing on the refrigerator with processing on infrastructure provided by a service provider. The service provider might support intermediate fog infrastructure as well as cloud services for deployment of applications and data storage. We explore these options in more detail later in this *Update*.

IoT Principles

Principles are critical guidelines for enterprise architecture. They set expectations and reinforce that EA decisions are not arbitrary. In order to illustrate an EA principle, consider first a democratic government principle. "Freedom of religion," for example, is a principle that supports the freedom of an individual or community, in public or private, to manifest a religion or belief in teaching, practice, worship, and observance. It also includes the freedom to change one's religion or belief. The freedom of religion principle has endured over 200 years in the US and has guided government, public, business, social, cultural, and individual decision making.

Similarly, EA principles are simple, promote consistency, support flexibility, and help enterprises realize their strategies. EA principles can be defined for six fundamental domains: service, process, information, application, data, and infrastructure. The diversity and volume of Things are natural targets for EA principles in order to establish guiding views on what and how Things can be used. EA principles are stated in the following standard format:

- **Name** — establishes the essence of the rule as well as consistent identification
- **Statement** — unambiguously communicates the fundamental rule
- **Rationale** — describes the business benefits of adhering to the principle, using business terminology
- **Implication** — highlights the requirements, both for the business and IT, for carrying out the principle in terms of resources, costs, and activities/tasks

A simple IoT principle example for the smart refrigerator is:

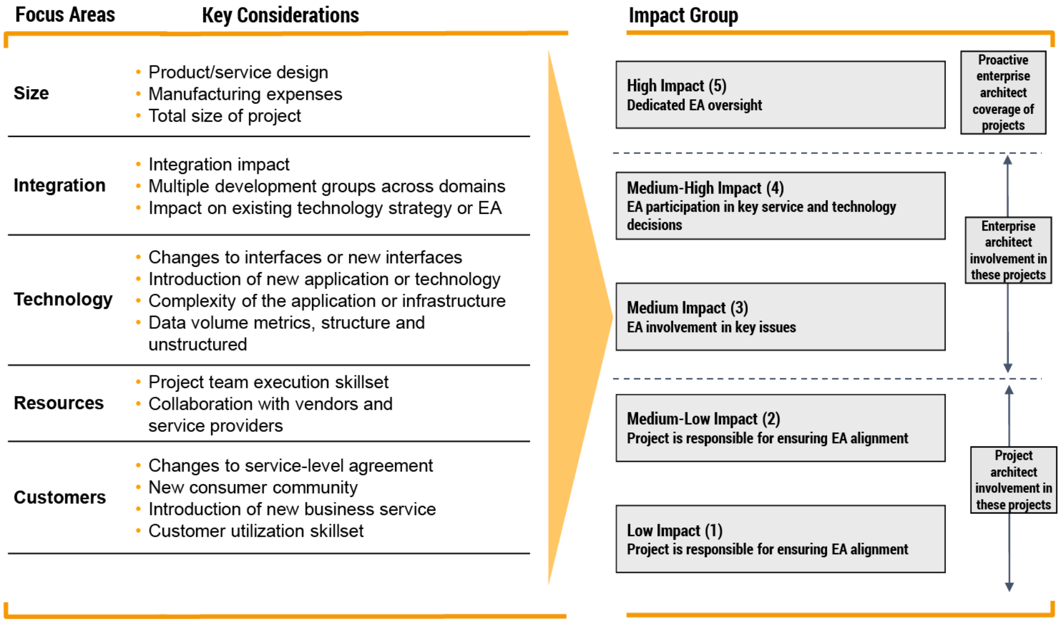
> *If a smart appliance enterprise implements the IoT Processor Design principle, its products will have a simpler internal processor (firmware) and robust integration interfaces. Application processing and data storage will be conducted on platforms external to the Thing.*

IoT Governance

EA governance is the set of mechanisms through which architecture is endorsed in the enterprise. Governance can be active or passive depending on how assertively architectural principles are enforced. If EA is to be successful, management must provide leadership by funding EA techniques and methods; sponsoring the construction of the architecture policies, principles, and processes; and supporting the EA framework once completed. Usually, responsibility for EA falls under the purview of the CTO. In practice, a team of architects is assigned responsibility for guiding business units and projects in order to adopt the enterprise's architecture when designing products and services.

Figure 5 shows an IoT architecture governance model as a foundation for the EA team's responsibility. To ensure adherence to the enterprise architecture, an architect might be assigned to a project. One of the responsibilities of an architect participating in smart refrigerator design is to determine the project's design impact (i.e., the key considerations in Figure 5). If the design decisions are considered to have a low impact on the reference model, the project executes without formal resolution of EA inconsistencies. On the contrary, if the project's design decisions are assessed to have a major impact on the reference model, the project requires architectural guidance to ensure adherence to the EA framework, standards, and principles.

In the example in Figure 5, medium-low and low-impact projects rely on project architecture expertise to guide design decisions. For medium-, medium-high, and high-impact projects, a member of the EA team provides the expertise to guide design decisions, since team members are subject matter experts in business and technology. Given Thing newness and complexity, EA team participation is a probable resource assignment when formulating a Thing project team.



Focus Areas	Key Considerations
Size	• Product/service design • Manufacturing expenses • Total size of project
Integration	• Integration impact • Multiple development groups across domains • Impact on existing technology strategy or EA
Technology	• Changes to interfaces or new interfaces • Introduction of new application or technology • Complexity of the application or infrastructure • Data volume metrics, structure and unstructured
Resources	• Project team execution skillset • Collaboration with vendors and service providers
Customers	• Changes to service-level agreement • New consumer community • Introduction of new business service • Customer utilization skillset

Impact Group

High Impact (5)
Dedicated EA oversight

Proactive enterprise architect coverage of projects

Medium-High Impact (4)
EA participation in key service and technology decisions

Medium Impact (3)
EA involvement in key issues

Enterprise architect involvement in these projects

Medium-Low Impact (2)
Project is responsible for ensuring EA alignment

Low Impact (1)
Project is responsible for ensuring EA alignment

Project architect involvement in these projects

Figure 5 — Impact assessment of the governance of a Thing EA.

Thing Solution Design Concepts

Recall that the purpose of EA is to create a blueprint of business services and processes and a map of IT assets to support the business services. When possible, business alignment, standardization, and reuse of IT assets is sought. The IoT opens up a new world of flexible design alternatives. Looking back at Figure 1, note the numerous options for data architecture and the alternatives for deployment: operational systems, data warehouses, data marts, cloud data, Hadoop, and NoSQL. Review Figure 4 and consider how the various DRM elements can be partitioned onto numerous platforms from an endpoint to a cloud. Let's explore a few of these design possibilities in more depth.

Data Design

Not all data fits into the row and column structure of a RDBMS. IoT, mobile, social, and cloud computing have created a massive flood of data. Some estimate that 90% of the world's data was created in the last few years, with 80% of all enterprise data being unstructured. Furthermore, unstructured data is growing at twice the rate of structured data. With the rise of IoT and the millions of devices coming online, this pattern will accelerate. In smart refrigerator design, for example, data design includes volumes of unstructured data generated by system monitor, network monitor, service monitor, and hardware monitor services. The data will be continuously generated 24/7, 365 days a year. This data volume will require new unstructured data design concepts.

Companies are looking to capitalize on the advantage of the unstructured big data storage driven through the IoT. In the world of unstructured big data, "not only" SQL (NoSQL) systems have emerged as major players. NoSQL is an alternative to building operational applications that drive business through flexible systems. High-capacity analytical systems, such as Spark or Hadoop, can be deployed to analyze IoT data delivering powerful insights. Two main categories that will support the IoT very well are "key value" systems like Apache's Cassandra and document-based systems like MongoDB. Both of these vendors currently hold the largest market share of this space. MongoDB allows developers to "more easily and expressively model the Things at the heart of our applications … having the same basic data model in our code and in the database is the superior method for most use cases, as it dramatically simplifies the task of application development, and eliminates the layers of complex mapping code that are otherwise required." While MongoDB can scale to the very large and supports horizontal sharding, replication appears to be supported better by Cassandra.

Sharding is essentially the concept of horizontal partitioning of a "data table" (from RDBMS) where each partition forms a shard, which may in turn be located on a separate database server in the same or separate physical location. This segmentation reduces the number of "rows" in a physical "data table," which reduces index size and therefore improves search performance. For the IoT, sharding should probably be based on the physical location of the device/data (e.g., Asia vs. America). This design will make it possible to find the right shard easily and automatically and to query only the relevant shard. Cassandra particularly shines in scale-out deployments of new shards (including replication copies for increased performance and backup). Multiple data centers are easily supported because to add capacity, "You simply boot up a new 'machine,' tell Cassandra where the other nodes are, and it takes care of the rest." (This ease of scaling, exceptional write performance, and predictable query performance make systems like Cassandra the next-level choice for supporting IoT huge data volumes.)

Reference Model Partitioning

When reviewing the logical Thing DRM, we can fully appreciate the power of design flexibility. With a Thing, device firmware capability can vary from low to high. Firmware with low processing capability can implement limited logic at the Thing end point; firmware with high processing capability can implement more complex logic. Partitioning, in which processes are allocated to layered technology platforms, is a developer's dream. Consider a Thing connected to a cloud service. Cloud services may be used in different ways depending on the IoT compute capability. Low-capacity devices are expected to be more transaction-oriented with fewer service changes (due to requiring firmware updates). These "thin" devices are best-suited for driving data directly into key value type storage, following an event-driven model. The shards could be designed for latency and volume (geographically perhaps) and replicated for redundancy. Dedicated virtual machines (VMs) in the cloud for command and control would respond to an individual Thing with required services. Other dedicated analytic clusters would work over the aggregated data, providing other services and discovering patterns to lead to new services. Higher-capacity devices may have a more complex transaction and service mix, as well as being able to dynamically change (due to patches/upgrades or local/remote hosting changes). These "thick" devices would use both key value (for event data as above) and document-based storage systems for more complex objects. Fewer command-and-control VMs would be required, while the analytic clusters would remain.

Current cloud architecture will likely evolve to allow for the hosting of data and VMs out to the "edges" of the network infrastructure. VMs would migrate from the deep cloud to the edge (fog) depending on size, capability, demand, or latency. Clearly, limited amounts of data could be cached in the fog before being flushed to larger and more capable cloud storage systems. For example, instead of Amazon Cloud hosting the VMs, Comcast, or even the Cisco router in the home/business, could host.

Consider two versions of our smart refrigerator. Version A is thick and has high compute capacity. It can capture many events (e.g., durations between door openings and closings, temperature swings, RFID of items removed and replaced). These and other sensor data could be processed locally to provide services to users, such as fridge contents, alerts on dates, shopping recommendations, or even collaborating with the "smart" grid to draw less power during peak times. This version might need fewer cloud services due to local processing, and push data to the cloud only for backup, diagnostics, or third-party/ancillary analytics. Data will be both structured and unstructured. At the other end, Version B of our smart refrigerator is thin and has low compute capacity. All events from Version A are still captured, but immediately sent to repositories in the cloud for processing. A VM is listening and responding with the same services as above, which are sent back for local display (or on an app on another device). Clearly, more cloud services are required.

In either version, decisions about where to host services need to be made, either deep in the cloud or on the edges in the fog. Collaborating with other services will be conducted "in-cloud" or "cloud-to-cloud." As an example of IoT collaboration, think about Amazon's Echo. One could ask, "How much milk do we have?" The Echo cloud service would send that query to the refrigerator (either the cloud service if thin or directly to the device if thick), receive a quantity, and reply, "You have less than half a gallon. Would you like me to put milk on the shopping list?" — assuming authorization is enabled between them.

Designing for security must be done at all places in the architecture. Not only are IoT devices another entry point for hackers, but the raw sensor data contains many widely known privacy concerns (e.g., knowing that the refrigerator has not been opened in several days compromises premise physical security). Analytic data has similar privacy issues (e.g., knowing the daily/weekly rhythms of the refrigerator). Security is not just encrypting data at both rest and in motion; it is defining access control over data and processes.

All cloud storage and analytical technologies leverage off the concept of "designing for failure." Our challenge is how to apply this concept to process management. Returning to Figure 3, we see that the Operate Product service contains a Failure Analysis subservice. If the processes for this service are executed in the cloud and the processor fails, a soft failure can be realized through redundancy. Service control can be transferred to another VM with a redundant Failure Analysis subservice. Generally, when application processing and data storage are pushed out from the endpoint, designing for failure can be more robust.

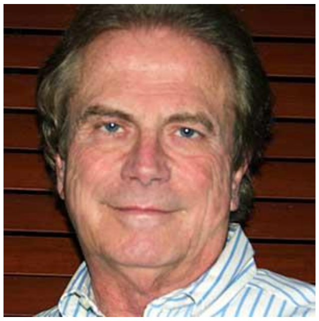
Conclusion

Enterprise architecture is an important methodology for improving and cultivating the business value of IT. The emerging capabilities of the IoT are valuable processing, data, and analytical contributions to new enterprise services and processes. Consequently, the business value of the IoT can be realized and validated when an EA framework is applied to IoT strategy and projects.

When an EA framework for IoT is constructed, it is not isolated from non-IoT components. In fact, this point is a major reason that EA is more important than in the past. Architectures should integrate non-IoT and IoT into a unified blueprint. This framework is composed of reference models, standards, and principles that encompass the expanded functionality of new non-IoT and IoT products and services. One specific IoT element that deserves focused attention is Thing data, since both structured and unstructured data are IoT elements. Over the past 40 years, IT has successfully implemented RDBMS as the platform for structured data. The IoT offers a challenge for unstructured data. Today, new data technologies and flexible platform configurations can be implemented to manage this data. From a market perspective, user-enabled products are a major growth sector targeted by many companies. Popular user-enabled products, including smartphones and wearables, are in demand in broad consumer markets. User-enabled products are increasingly powered by the IoT.

When designing a Thing, a wide range of processing and storing options are available. An architecture framework provides the guidance that directs wide variability into practical implementation, consistency, and reuse. Enterprises can use today's endpoint, fog, and cloud processing layers to allocate IoT processing requirements onto selected processors. Replication, redundancy, and failure in this environment can be designed and managed in line with business strategy. The huge data volumes realized with many IoT products and services can be organized and managed using cloud computing and NoSQL databases.

About the Authors



Charles W. Butler is a Professor in the Department of Computer Information Systems at Colorado State University. He teaches and conducts research in IT and collaborates with IT managers in developing improved IT management strategies and processes, software development methodologies, and metrics and QA for traditional and object software. For 15 years, Dr. Butler served as Chief Software Scientist for McCabe & Associates and completed consulting engagements in more than 50 Fortune 500 *companies. At McCabe, he served as an expert witness in software litigation trials. Dr. Butler has also worked with Texaco, Inc., and Star Enterprise for over 15 years on internal process improvement projects. He is the author of "IT for Management," a Harvard Business Publishing online course. Dr. Butler holds a bachelor of arts degree and a masters of science degree from the University of South Florida, a PhD from Texas A&M University, and is certified as a Project Management Professional (PMP).*

Stephen C. Hayne is a Professor of the Department of Computer Information Systems at Colorado State University. He has received more than US $6 million in grants from NSF, ONR, and DHS as well as an IBM Faculty Fellowship. Dr. Hayne has published more than 70 papers in journals such as Journal of Management Information Systems, Decision Sciences, Database, Journal of Information and Management, Journal of Computer Supported Collaborative Work, IBM Systems Journal, Electronic Markets, Journal of Organizational Computing & Electronic Commerce*, and* International Journal of Human Computer Studies*, and has presented at major conferences. His research is based on the desire to use innovative technologies to solve real business problems. Dr. Hayne has developed theories and implemented systems to help groups with communication and decision making, using concepts such as shared cognition, collaborative drawing, group brainstorming, concurrent issue surfacing/consolidation, consensus building, choice, pattern recognition, and team bidding in auctions. Other research interests lie with social networks theory/analytics, distributed systems, big data, auctions, and ecommerce. His most recent research award involved detecting and mitigating large-scale distributed denial-of-service attacks. Dr. Hayne's teaching in the subjects of network security, database, strategy, ecommerce, and software engineering have led to nominations as Best Teacher at Colorado State University. He has been General Chair and Treasurer for the many of the* GROUP *and* CSCW *conferences and was cochair of the* 2007 America's Conference on Information Systems*. Dr. Haynes received his PhD from the University of Arizona.*