White Paper

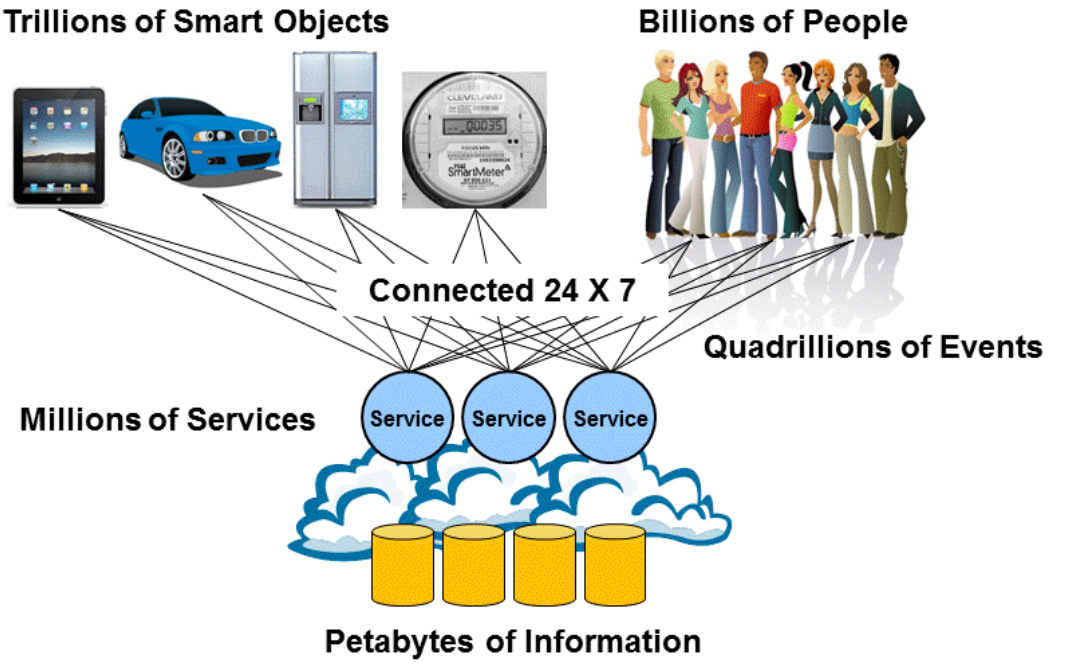
The Service Factory as a Service

A Strategic Approach to support Enterprise Transformation

Contents

January 2014
Everware-CBDI Inc.

The Changing Shape of Business



We all observe the waves of disruptive technology delivering high bandwidth, always-on connectivity, mobility, BYOD, social networks and big data, as well as next generation analytics, robotics and the Cloud. At this time there is little doubt that these technologies are enabling a new phase of business exploitation of information and communications technologies. In 2013, economist Paul Krugman, suggested[1] IT is at last becoming significant, enabling a technology revolution to rival previous technology revolutions. Specifically he cites driverless cars as an example of the move of technology into the physical world that has the potential to power growth.

This phenomenon is a good example of the digital business, the convergence of various disruptive technologies together with the Internet of Things, where consumers use technology to merge their business and personal roles to a very large extent. In this world of digital business, intermediaries become redundant as machines talk directly to machines with artificial intelligence increasingly governing responses to events.

These major trends will clearly have a profound impact on the very nature of the enterprise. The consumerization of IT and the real time nature of transactions will radically alter many business processes such as service desks and call centers. We must also expect complex ecosystems to form around concepts such as driverless cars, and new entrants emerge to challenge existing market leaders. In this environment the service interfaces between enterprises and their capabilities will become the focus of much attention, as ecosystems both internal and external to enterprises emerge and evolve.

The real time nature of the emerging digital business will compel enterprises to publish APIs to their core capability services, effectively turning the enterprise inside out in order to communicate and transact with customers and supply chains. In this new world, the successful organization will take the opportunity to rationalize internal application complexities into networks of highly independent business services that facilitate the inherently agile enterprise.

The successful organization will also embrace radical changes to the way services and solutions are delivered, which are more relevant to the fast emerging digital business world. This document introduces the concept of the Service Factory as a Service (SFaaS[1]), a genuinely ground breaking advance over conventional solution delivery practices which can deliver transformation to and realization of an agile business architecture without major up-front investment.

Software Services – the Foundation for Agile Business

We may anticipate that the emerging digital business environment will be highly competitive and fast moving in all industry sectors. In the past, the major drivers for business agility have included M&A activity and the need for business process improvement. Over the next decade we need to plan for the constant introduction and development of disruptive technologies and evolving and new business models. This, in turn, will trigger opportunities for new ecosystems and new capabilities that will be provided by existing and new market entrants. In many cases existing business processes will be subject to far-reaching change as real time, disintermediated relationships become commonplace.

Existing enterprises that have large application investments supporting today's business processes will be challenged by new entrants that do not have the legacy technology and cost infrastructure. To survive, enterprises will need to embrace an inherently agile business and IT architecture that exposes APIs to software services that are designed with agility in mind.

Of course it's not always easy to architect and specify an agile software service. In principle the service must be designed, within some limits of cost effectiveness, to be able to provide continued and uninterrupted support in a constantly changing and evolving business environment. For example, it must be capable of supporting new devices, appliances and machines that use the service's capability. The table below provides example policies for agility requirements. Experience tells us that many existing software services will not comply with these characteristics, and de facto software service delivery practices will fail to deliver compliance as well.

[1] SFaaS pronounced "ess faz"

The analysis in Table 1 below suggests that management of service specification and delivery must be elevated above short-term focused project development and treated as strategic infrastructure.

Agile Service Characteristics	Critical Architecture and Design Criteria
Able to support many different devices, appliances and machines.	**Strong reference architecture and layering.** Service behaviors, experiences and protocols are channel neutral and separate from channel specific business services.
Flexibility of supply; service capability can be provided by multiple providers concurrently or over time.	**Service Encapsulation** Contract hides all underlying capability behavior yet provides sufficient detail to support implementation independent consumption decisions.
Ease of change. Rapid response to change.	**Componentized Capability** Restricted horizon of change achieved by componentized service implementation.
No duplicate or replicated services.	**Strong Service Architecture** Shared Services provide common capabilities and eliminate redundancy.
Able to support changes in business process.	**Customizable by design** Policy based separation of behaviors. Atomic service scope. Assembly and composition. Rules based behavior specification.
Able to support variations in business (global, local, line of business, geography etc) from common core.	**Standardized Core, Customizable Variant Behaviors** Model based abstraction allows fine grained versioning of architecture and specification for multiple, concurrent variants of standardized core behaviors.

Table 1: Example Business Agility Policies

Challenges and Inhibitors

IT Executive Committees are constantly faced with difficult choices from dividing up a flat or declining budget to maximizing the response to demand for new and enhanced business capabilities. In the melee surrounding allocations to short-term projects, strategic requirements are frequently overlooked.

Recently, a number of well-known banks have unintentionally demonstrated, in a very public manner, the disastrous state of their core business systems as ATMs are shut down and millions of customers' accounts are corrupted, not by malicious activity but by process malfunctions caused by systems way past their sell-by dates that have ended

up as complex, unmanageable nightmares. Enterprises in other industry sectors should not be complacent; most of them have similar time bombs ticking away in the bowels of the enterprise, which may at some time explode causing huge disruption to business and customer confidence.

Because of overwhelming short term needs, few enterprises have rationalized their back end systems. For many enterprises this has been a simple matter of economics – until, as discussed, a failure to keep systems modernized has unintended consequences. Acting upon this issue becomes considerably more pressing because a complex legacy application portfolio will without question restrict future business growth and agility. While a portfolio of software services may be implemented as a façade on top of existing legacy applications, the transformation to a digital business will inevitably require changes to core business rules and data, which will still be subject to the extended delivery times and high costs associated with changes to legacy core systems.

While the majority of enterprises have embraced service oriented architecture (SOA), the reality for many is that short-term project priorities have sub-optimized the service architecture. In the digital business, the service architecture must deliver inherent business agility and not be compromised by legacy application or project specific constraints.

Service Factory as a Service

Figure 1: The Factory as a Black Box

The service factory concept has been in vogue for some time. As long ago as July 1989 the Harvard Business Review published a seminal article by Chase and Garvin titled The Service Factory[2]. They argued that "The factory of the future is not a place where computers, robots, and flexible machines do the drudge work . . . the next generation, then, will compete by bundling services with products, anticipating and responding to a truly comprehensive range of customer needs."

In this paper we explore the concept of the Service Factory as a Service (SFaaS) which provisions software services delivered on-demand using the SaaS (Software as a Service) model. As Chase and Garvin envisaged, the Service Factory as a Service is a combination of technology products that deliver very high quality and productivity,

combined with professional services that integrate the factory inputs and outputs into the digital business solution delivery process.

The service factory is a specialization of a generic software factory concept described by Greenfield and Short et al[3] in 2004. Both software and service factory concepts are based on the principle that the inputs and deliverables of the design and development tasks can be predetermined and defined as a schema – a detailed representation of the information that is created and managed in the development process. Further that the execution of each task may be automated to a greater or lesser extent by applying configurable patterns that facilitate transformations, such as models to code, with minimum human intervention. As might be expected, the internals of a factory are therefore specialized. They use technologies and techniques that the average solution developer would not need to be familiar with. In fact the principle of the factory, as illustrated in Figure 1, is that it is operated as a black box and for a given specification input, deployable services are output.

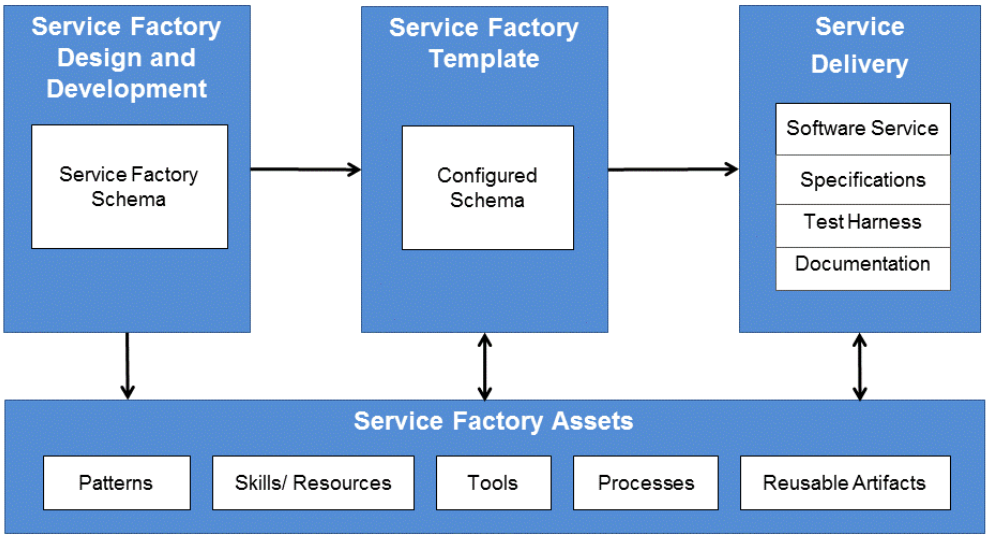


Figure 2: The Service Factory Concept

Chase and Garvin indicated that the factory of the future would anticipate and respond to a comprehensive range of customer needs, and Figure 2 illustrates this. The base factory schema provides the development framework that is designed to meet generic types of demand. In the case of the service factory this may be types of software services, such as channel, process, core business and underlying services for publication using popular service protocols such as REST and SOAP, and deployment to specific target platforms such as Java. The factory is effectively a collection of framework assets including patterns, tools, processes, skills and resources, as well as reusable artifacts that enable the quality and productivity outcomes. The factory schema may then be configured for specific customers to deliver variant types of service, pattern, protocol or target platform. Service delivery then uses the configured schema to deliver software services together with specifications, test harnesses and documentation.

There are several examples of software factory frameworks available in the marketplace. These are intended as accelerators for end user or service provider enterprises to create their own software factory. The real breakthrough, however, is to establish a type of software factory which is purpose-designed for software services, and to make this available on-demand as a comprehensive service factory as envisaged by Chase and Garvin.

Figure 3 below illustrates the "as a Service" aspect of the Everware-CBDI Service Factory, providing on demand provisioning of software services to conventional, in-house and outsourced solution delivery processes. The service provisioning value chain of specification, automation and completion interacts with the solution delivery streams using purpose designed portals and APIs that coordinate the exchange of requirements and deliverables. Professional services support the set-up and smooth running of the factory interface. Set-up activities will usually establish the customer relationships and process interactions with the service factory provider or, for larger programs or enterprises, the establishment of an in-house service factory. The architecture, service specification and design and integration support professional services are provided to guide the client in achieving high quality outcomes from the process as well as providing skills transfer, facilitation and mentoring as appropriate.

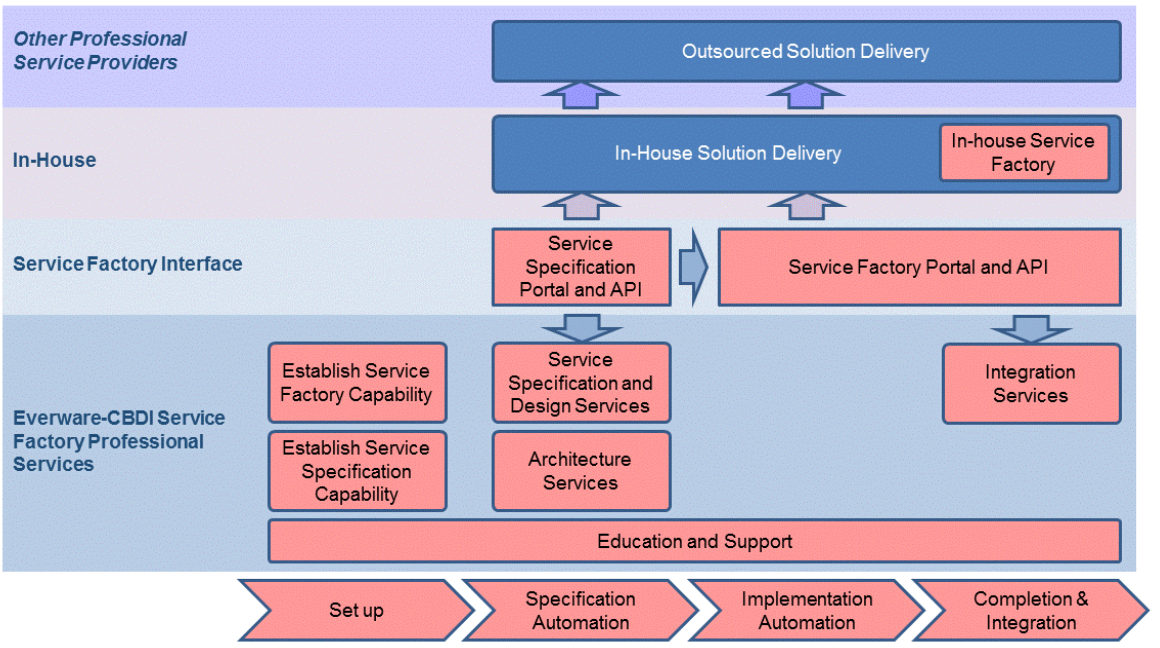


Figure 3: Everware-CBDI Service Factory as a Service - Delivery Models

Service Specification as a Service

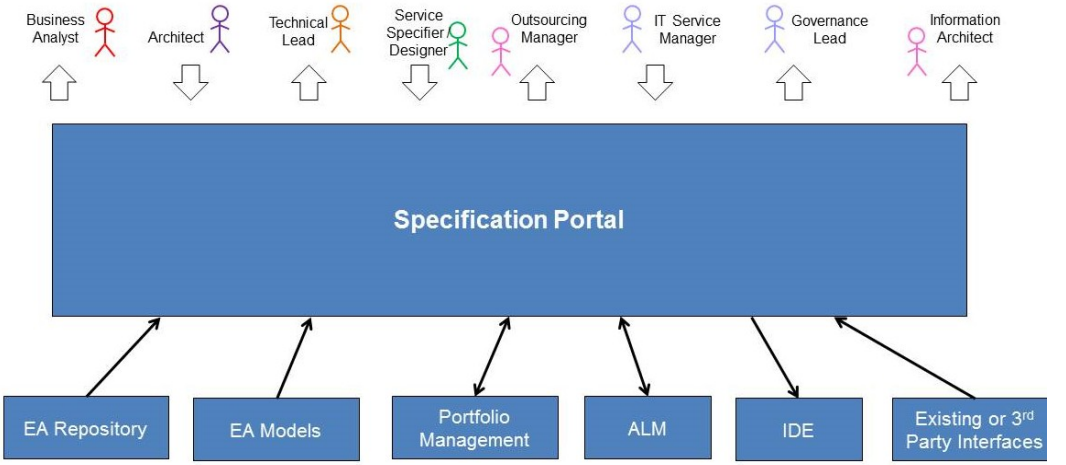


Figure 4: The Service Specification Portal

Realizing the architectural vision is one of the most difficult outcomes to achieve in enterprise solution delivery. It is all too easy to specify services to meet the requirements of known projects. However, as discussed, it is essential that the business needs for agility are translated into terminology that is equally understood by business subject matter experts as by software developers.

The service specification is therefore the critical deliverable in the overall delivery process. The service specification interprets the architectural and business intents for key agility criteria such as standardization or localization, specificity or generalization, customizability and capability independence, and translates these into unequivocal implementation independent details.

Service specifications must also reflect the concerns of multiple stakeholders including functional subject matter experts, architects, technology and deployment specialists, procurement and outsourcing specialists, information and data specialists, service management and the governance role. A fundamental principle of service specification therefore is that the deliverable must be the result of cross functional collaboration which achieves compliance with policies and an optimal achievement of the architectural goals. It is also important to recognize that a well formed service specification should be easily transformed into a guide for consumers of the service.

Across the IT industry there are many and varied approaches to service specification. Some organizations detail the service using models or documents but rarely have tight linkage between the specification and the delivery or consuming processes. Other organizations outline the service and leave the detailing to a solution delivery project, which may place too much weight on the specific project requirements as opposed to creating a widely reusable shared service. This variability of approach reflects the immature status of tool support to the specification task. A recent survey[4] reported that

64% of respondents are clearly dissatisfied with tools for communicating service specifications, including requirements capture, reviewing, reporting and management.

The Service Factory as a Service process is of course critically dependent upon the quality of the service specification, and therefore provides a mature capability to address these issues. A specification portal, illustrated in Figure 4, provides a multi-stakeholder specification capture capability that integrates stakeholder perspectives and ensures comprehensive, consistent specifications.

The specification portal also integrates with the various modeling environments, repositories and tool platforms that allow the service specifiers to make use of and align with upstream information sources including enterprise architecture, model and portfolio management repositories, and to feed downstream delivery platforms including development environments and application life cycle management tools. In addition, flexible reporting capabilities allow custom reports to be prepared for each stakeholder.

In this way higher quality specifications are produced that are a prerequisite for higher quality delivered software services and associated documentation including the published API specifications. The integrated specification approach ensures a higher level of real governance that addresses compliance with architectural goals and traceability from business requirements to executable services. The same integrated, automated and detailed approach also facilitates greater governance over the use and creation of services in outsourced solution delivery.

Service Delivery

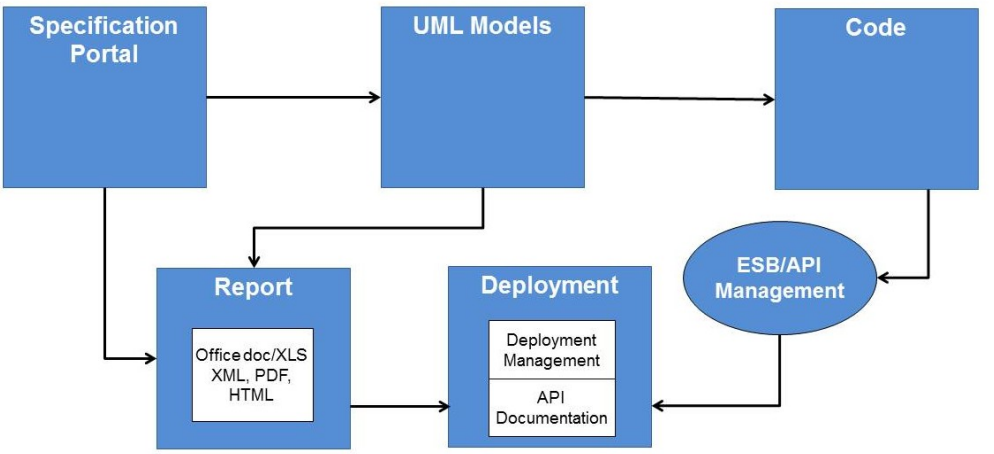


Figure 5: Consuming Organization's View of the Service Factory

From the consuming organization's perspective the service factory must be a black box. This allows the factory to constantly evolve and optimize the internal factory operations. Figure 5 illustrates the external perspective. The specification portal is the primary interface for expressing detailed service requirements and can import existing

knowledge about legacy systems, services and data structures from models and other sources.

Specification details may be exported to a variety of formats including Office, XML, PDF and HTML to permit reporting. The comprehensive data recorded in the portal permits customized reporting for each of the various stakeholders. The details recorded in the specification portal are exported to UML models which are the primary interface to the factory code generation process, but also represent an important documentation, communication and reporting mechanism for provisioners and architects.

Typically the SFaaS process automatically generates approximately 80% of the ultimately deployable code. The remaining code must be manually added as part of the factory process guided by business rules and behaviors defined in the service specification. However, some customers may elect to use the factory to deliver the 80% and to incorporate completion into their own process. The generated code is referred to as Architecture Owned Code (AOC) and represents an important form of governance in terms of automating the implementation of IT policies. It is recommended therefore that the ownership of the factory-produced codebase remains within the factory, as there is considerable benefit in regenerating the AOC in an iteration or upgrade cycle in terms of both productivity and integrity. However it is emphasized that all deliverables are based on open standards and are not restricted to the factory delivery environment.

This approach, generally referred to as Model Driven Architecture and Development (MDA/MDD)[5], allows management of code at the model level of abstraction. Initial specifications and changes are applied at the portal and model levels from which code is generated. In practice this allows automated management of functional dependency and impact of change at a fine grained level which facilitates coordination of collections of services within a program or an entire portfolio. This has proven in practice to allow functional changes to be applied with a very low level of effort and high level of managed consistency. This can be highly effective in support of an Agile delivery process that has an inherently iterative approach to specification and development. Equally the managed consistency is very effective in the ongoing evolution of service functionality, both delivering rapid response to change at low cost and, even more important, retaining the architectural integrity of the individual service and its relationship with the broader portfolio.

A central value of the SFaaS concept is that the very detailed specifications intrinsic to the process enable comprehensive testing of all the behaviors that have been specified. Accordingly the delivered services are accompanied by a test harness reflecting the specification, and the services are certified as compliant with the specification. The deliverables from the factory process will be deployable and include the detailed Service Description for publication in the Service API for use by service consumers.

Experience with a fully configured service factory has demonstrated achievement of 5 to 10X improvements in development productivity, with dramatic reduction in time to

market and significantly reduced costs. The service factory would usually provide the service at fixed costs based on an outline service specification.

Service Factory Roadmap

The "as a Service" model inherent in the Everware-CBDI service factory is key to reducing barriers to adoption. Like all SaaS services capital investment will be dramatically reduced or eliminated. While there will of course be set-up costs to establish the processes and skills, these can be phased in, in-line with the roll-out plan. Consequently the impact on solution budgets should be minimal or positive, insofar as services will provide core functionality to solutions that should be delivered at lower cost and be reusable across multiple projects and programs.

Areas that will require attention are in raising the capability maturity of architecture, specification, and governance as well as a multi-vendor provisioning process. While initial or pilot service factory projects will require minimum architecture effort, there will be considerable value realized from the development of a comprehensive service and API architecture and portfolio. This effort must be complemented with effective coordination and governance that exerts control over demand management, project and program chartering activities and establishes service deliveries and upgrades in support of solution projects.

Roadmap Items	Business Value
Service factory pilot deliveries	Establish viability of factory process including technical fit, economics and business value.
Business case	Clarity of how business value will be derived
Reference architecture for services and APIs and process for continual improvement	Consistent approach will optimize business agility in delivered services
Service architecture and service portfolio transition plan (Key domains and or ecosystems)	Identified shared services and strategy for integrating into solution deliveries
Twin track (service and solution) process	Separation of service and solution tracks enables specialization and optimization of procurement, skills and practices.
Socialization of architecture in the solution delivery processes.	Ensures common understanding and enlists proactive support by all impacted roles
Architecture involvement in the demand management process	Optimal delivery approach is selected and chartered
Architecture governance in the solution delivery life cycle	Compliance with reference and instance architectures ensures realization of agility based architecture

Table 2: Key Roadmap Priorities

Conclusions

Realization of architecture is frequently more challenging than might be expected. There are countless reasons for this including short term pressures, ineffective governance and cultural mismatch between architects and delivery teams. The business value accruing from SFaaS is therefore potentially much greater than simple improvements in productivity, operational quality, delivery speed and cost. A primary opportunity is to leverage the separation of concerns to govern the delivery of quality services and APIs that facilitate a genuinely agile business today and tomorrow.

Introducing change in established working practices is however always difficult and IT organizations are no different in this regard to other disciplines and stakeholders. An important attribute of the SFaaS is therefore the "as a Service" delivery method that enables progressive adoption, and all the benefits to be realized even in initial service deliveries, without significant up-front capital investment or adverse impact on solution project budgets.

Accreditation

Everware-CBDI is a specialist firm with extensive experience of guiding larger commercial and government enterprises in structured approaches to service-oriented application modernization. The Everware-CBDI Service Factory is a productization of concepts, tools and technologies that have been successfully used in numerous projects and professional services engagements and is, we believe, a major advance towards realizable architecture. For more information see: http://everware-cbdi.com

References

[1] Paul Krugman, http://truth-out.org/opinion/item/14347-a-technological-drive-toward-driverless-cars

[2] The Service Factory, Chase, Garvin, Harvard Business Review, July 1989

[3] Software Factories, Assembling Applications with Patterns, Models, Frameworks and Tools, by Jack Greenfield and Keith Short with Steve Cook and Stuart Kent, Wiley Publishing Inc. 2004

[4] Everware-CBDI Service Specification Survey 2013 http://cbdiforum.com/sssr

[5] MDA/MDD Model Driven Architecture and Model Driven Development http://www.omg.org/mda/