Multi-Cloud Service Management Business Guide

A business oriented accelerator guide to the end-end management in multi-cloud and machine-machine environments

TR195

Version 8.7

February, 2013

NOTICE

Direct inquiries to the TM Forum office:

240 Headquarters Plaza,
East Tower – 10th Floor,
Morristown, NJ 07960 USA
Tel No. +1 973 944 5100
Fax No. +1 973 944 5110
TM Forum Web Page: www.tmforum.org

CONTENTS

LIST OF FIGURES

1 FOREWORD

This introductory guide is part of the Forum's *Multi-cloud Management Accelerator Pack* aimed at helping readers address the challenges of delivering high quality services that are built from multiple cloud-based components. The issue to be solved is how each managed service component can expose sufficient operational and management information that the provider responsible for delivering the overall service can be sure that the service is meeting its design or contractual service levels.

Given the sheer variety of cloud based services that the market is generating, it is impossible to try to develop comprehensive management Application Programming Interfaces (API's) that cater for every type of cloud services, multiple industry verticals and machine-machine (M2M) services. The approach taken here is for a simple core set of basic management functions that can apply generically to many types of service and that can be easily extended to provide more service specific functions or map to the myriad of proprietary API's that already exist.

The Multi-cloud Management Accelerator Pack is based on the work of the Forum's Software Enabled Services Management team and the Multi-cloud catalyst project, the contribution of those member companies are gratefully acknowledged. In particular, Eric G Troup at Microsoft Corp was a significant contributor to this document and the lead editor of the Multi-cloud catalyst best practice team.

Throughout this and related documents the terms *cloud service, digital service and service provider* are used. These are generic terms and can relate to any kind of digitized service or provider – e.g. content or application 'software as a service' (SaaS); platform services (PaaS, such as authentication, security or monitoring) or infrastructure services (IaaS) such as cloud computing, storage or networking services. The main focus of the accelerator pack is on managing SaaS types of service.

With the rapid growth of cloud computing, the consumerization of IT, the trend towards services and content created increasingly at the edge of the network, the exponential growth of exposed Application Programming Interfaces (APIs), and increasingly mobile endpoints means that this work has never been more relevant to contemporary industry needs. Since the approach described here defines only the *management* aspects of the service, not the actual service functions themselves, this makes it equally useful in any industry vertical such as Healthcare, Retail, Manufacturing, Financial, Logistics, Public Sector and Defense.

The full accelerator pack consists of a number of documents, software tools, code and developer eco-system. More details on these are available here.

A companion document, the *Multi-cloud Management Accelerator Pack - Developer Guide*, provides a much deeper insight into how this can be implemented as well as being extended or mapped to other interfaces using a technique called design patterns. Developers and implementers will find this document and the related web-based developer tools and environment useful to accelerate their Service Oriented Architecture implementations leveraging cloud, network, and enterprise resources while delivering significantly better user experiences, better end-to-end operations management capabilities, and greatly improved developer efficiencies.

2 EXECUTIVE SUMMARY

The digital world is full of innovative ideas, great marketing and often good access to capital. But unless you can get those services to market rapidly; deliver a great customer experience; fix problems when they occur and collect or apportion revenue while keeping operating costs as low as possible, you won't survive for long. This need for an excellent 'operations experience' is especially true in the market for enterprise-class digital services where companies are increasingly dependent on managed digital services to run their business. They demand reliability, security and a quality of experience that meets their contractual expectations – if the service fails to do that it is not just inconvenient, it directly impacts their ability to operate and their bottom line.

The business of ensuring that service creation and delivery produces highly reliable, predictable and massively scalable services, while keeping operating costs in check doesn't happen by accident: it is a core competency of any successful service provider. This is a complex enough task where the provider has end-end control of all of the assets in the service delivery chain but it becomes orders of magnitude more complex where the overall service is comprised of multiple virtual services connected together. Each component service may be manageable on an individual basis, but the end user sees the net sum of all of the operational issues within the ecosystem and the final service provider – the one who contracts with the customer – inherits the problem of managing across the entire chain.

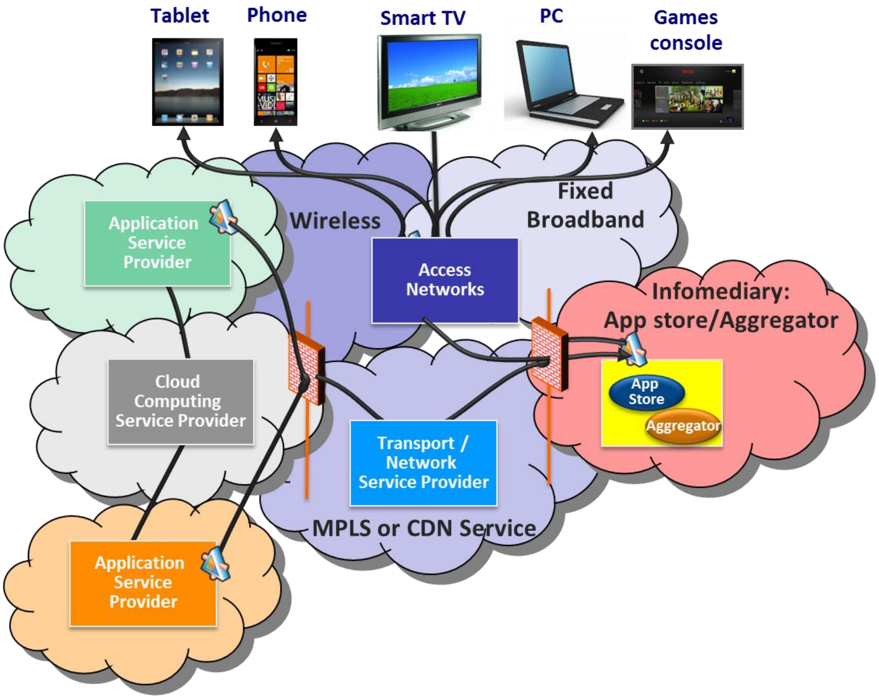

Tablet
Phone
Smart TV
PC
Games console
Application Service Provider
Wireless
Fixed Broadband
Access Networks
Infomediary: App store/Aggregator
App Store
Aggregator
Cloud Computing Service Provider
Transport / Network Service Provider
MPLS or CDN Service
Application Service Provider

Figure 1 – The Multi-Service Nature of Service Delivery

THE OPEN DIGITAL WORLD

Improving speed to market, eliminating manual intervention, cost and errors in routine operational business processes are key operational goals for digital service providers since they directly impact service profitability. But increasingly Content Owners, Communications Service Providers (CSPs), Multiple System Operators (MSOs), Cloud Providers, Businesses, Consumers and developers are all interdependent and this makes the processes of creating content / services, service delivery and of managing the overall experience end-to-end in such a multi-service provider / multi-cloud environment increasingly challenging.

Increasingly, discrete services are being aggregated into overall user solutions allows for wider uptake of those services into the marketplace. This adds to other major trends that are continually amplifying the challenges of providing digital services that work reliably such as:

- Multiple devices from a variety of manufacturers,
- Complex developer ecosystems,
- Exponential Growth of Service APIs
- Reality of Multi-Cloud Service Delivery.

But in this increasingly complex digital world, the need for highly automated and effective service operations has never been greater since they are essential to produce reasonable rates of return on investments. With countless billions of devices and a myriad of types of digital service in play, the idea of managing that involves any form of manual intervention is prohibitive from both a cost and practical standpoint. The problem is that automated, end-end management is simply not possible if the management information and service control is isolated in individual cloud 'islands - Figure 2 below shows a very simple example of this problem.

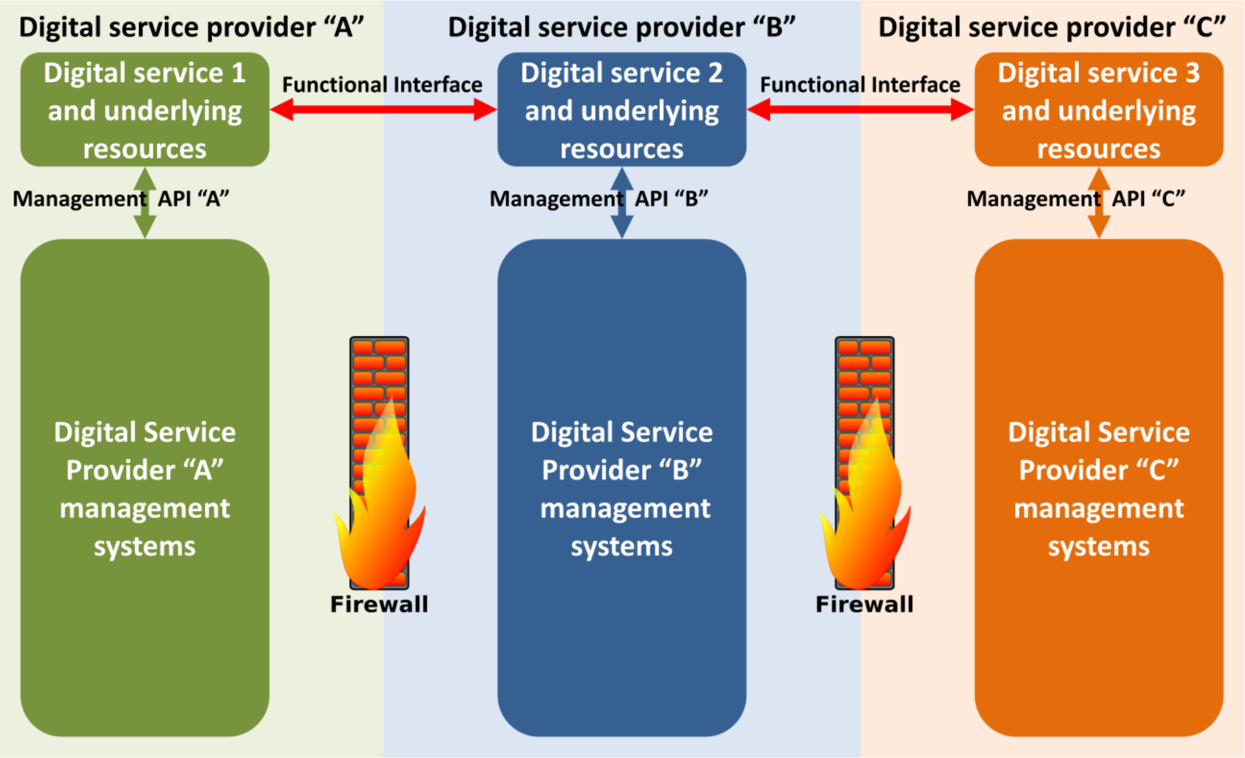

Digital service provider "A"
Digital service provider "B"
Digital service provider "C"
Digital service 1 and underlying resources
Functional Interface
Digital service 2 and underlying resources
Functional Interface
Digital service 3 and underlying resources
Management API "A"
Management API "B"
Management API "C"
Digital Service Provider "A" management systems
Firewall
Digital Service Provider "B" management systems
Firewall
Digital Service Provider "C" management systems

Figure 2 - Management locked within each component digital service

What is needed is exposure of a common set of management information at the 'edge' of the service (i.e. the point of interconnection) and this is the approach taken by the TM Forum. At the heart of the approach is a set of basis management functions called the Frameworx Simple Management API (or Frameworx SM-API). This is a suite of light weight, universal management interface's that is useful for services throughout the value chain, sample services are covered in the Developer pack today, but the value of the SM-API goes well beyond these initial examples. The expectation is that what is presented will evolve rapidly to meet the end to end management needs of a complex and evolving digital ecosystem. The developer package includes sample code, RESTful implementation and supporting documentation designed to help rapid deployment.

The sheer range of service complexity and a multitude of market needs means that it is very difficult to try to standardize every management function that will ever been needed. However, by standardizing a common core of generic management functions, allowing developers to easily extend that functionality and publish the results, the functionality of the core will extend over time. The approach allows companies to extend the functionality of those interfaces to suit particular business needs and easily map the common interface to proprietary interfaces.

While the need for competitive differentiation ensures that it is unlikely that providers will want to standardize the services themselves, exposing management information in common format is a positive business driver because increases the market applicability of the service by ensuring that it can be marketed through multiple channels and in combination with other services as part of an comprehensive customer solution. This is a fundamental plank of the drive for an open digital economy based on the ability of any digital service to 'trade' with any other digital service.

THE FORUM'S MULTI-CLOUD MANAGEMENT ACCELERATOR APPROACH

To address these issues, TM Forum members have developed the Frameworx Multi-Cloud Management Pack - a set of guides, software, tools and a developer ecosystem which allows companies to rapidly add a common Simple Management API to their cloud services. It also allows companies to extend the functionality of those interfaces to suit particular business needs and easily map the common interface to proprietary interfaces. The approach does *not* constrain the service provider to any particular functionality provided by the service i.e. via a *Functional Interface;* it only deals with the management issues via a *Management Interface, the Service Management API.*

The Forum's approach concentrates on two areas:

- The Frameworx Simple Management API (SM-API) which defines a design pattern for an API that reveals how to manage any given service from a Provisioning, Assurance and Usage/Charging perspective.
- The Frameworx Service Lifecycle Management (SLM) defines best practices and requirements for establishing a role based software/services factory and a Lifecycle Management Meta Data model. This is aligned to the rest of Frameworx and ITIL.

OVERCOMING END-TO-END CLOUD SERVICE MANAGEMENT ISSUES

There are two principal issues with digital services that make the problem of managing their resources difficult. First is the virtualization at the elastic compute and elastic network layers as well as the sheer scale of that virtualization. The second difference is that multiple clouds and multiple enterprise domains are increasingly involved in the delivery of digital services further complicating resource management.

To address this problem, this guide describes how the management systems that manage and control a service can coordinate the SM-API aspects of an application or service with the associated state, health and welfare of underlying cloud and network resources.

The Frameworx SM-API provides a light weight and consistent method for services to interact with service management applications in single or multiple provider environments. Leveraging this capability enables complex service ordering and provisioning as well as customer dashboards to accurately display the status of a service including underlying service component not under the direct control of the local service provider or customer.

As illustrated below, these management systems can maintain the relationship between an application instance and the specific virtualized resources supporting that instance. This enables relevant telemetry from that service and associated underlying compute and network resource layers to be relayed to a and used to update a Service Model dashboard.

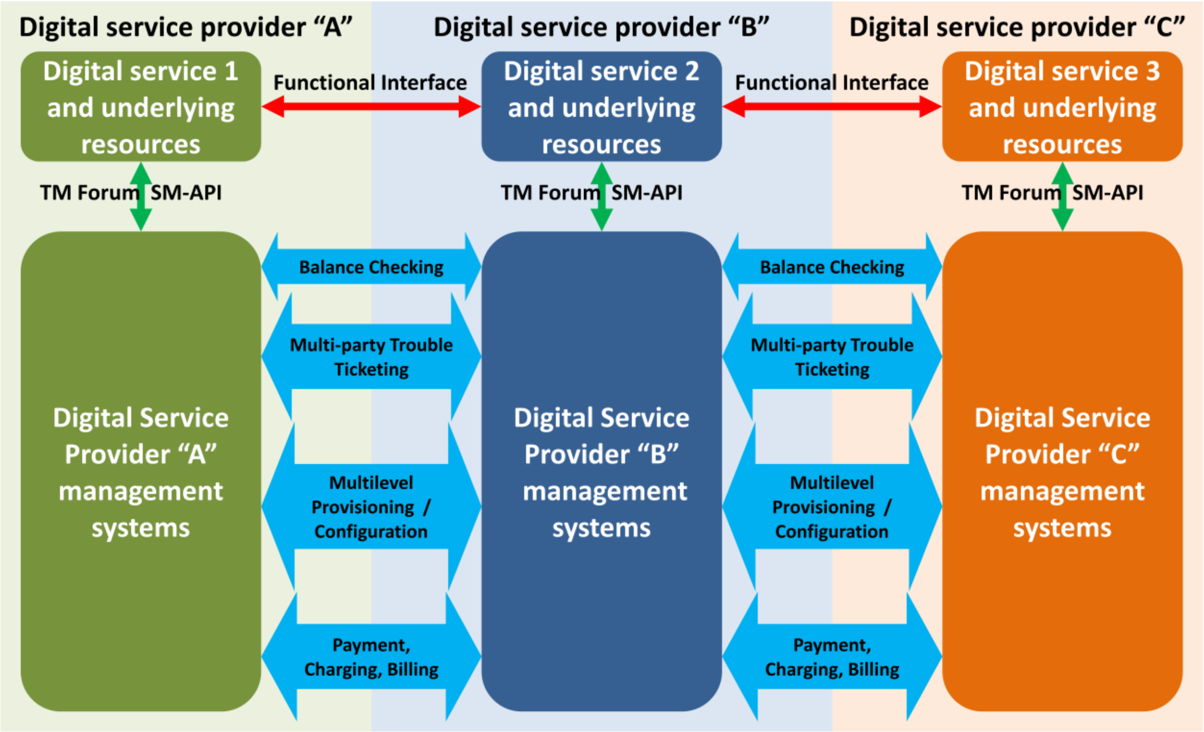

Digital service provider "A"
Digital service provider "B"
Digital service provider "C"
Digital service 1 and underlying resources
Functional Interface
Digital service 2 and underlying resources
Functional Interface
Digital service 3 and underlying resources
TM Forum SM-API
TM Forum SM-API
TM Forum SM-API
Balance Checking
Multi-party Trouble Ticketing
Multilevel Provisioning / Configuration
Payment, Charging, Billing
Balance Checking
Multi-party Trouble Ticketing
Multilevel Provisioning / Configuration
Payment, Charging, Billing
Digital Service Provider "A" management systems
Digital Service Provider "B" management systems
Digital Service Provider "C" management systems

Figure 3 - Managing Services in a Virtualized Multi-Cloud Environment

Where different component services in the service delivery chain are all using the SM-API to expose management information, end-end management is made very much simpler and cheaper to perform. In cases where some component services expose management information in non-standard or proprietary formats, some form of mediation platform or Service Delivery Broker (described in more detail in section 5) can be useful to help manage service creation and delivery in this environment.

Using a more standardized approach to multi-cloud service management delivers three key value propositions:

- A High Quality User Experience – Users are able to access the business application or see the content in the manner they expect.
- A High Quality Developer Experience – Developers are able to more quickly create applications in a consistent manner that can be easily incorporated into SOA Service Compositions that are readily manageable from a QoS and SLA viewpoint.

- A High Quality Operations Experience – Service Providers are able to provide a great user experience because they have the information necessary to measure what is going on, quickly assess root causes and impacts, and react to problems in a proactive manner.

IMPLEMENTING THE ACCELERATOR

Producing industry standards and more importantly, getting traction in the marketplace, requires a very different approach in the cloud world than the telecommunications market. Historically, communications were closely governed by a relatively small number of often government controlled bodies. Standards developed by the International Telecommunications union and other bodies could be adopted by common agreement and even when the market liberalized, new entrants needed to adopt common standards for say, international roaming, otherwise they could not do business.

The digital world has no such central governance and will not simply adopt standards just because they exist - the market is too fast moving and volatile for that. Standards have to emerge because they provide economic benefits to implementers such as providing a faster, cheaper, better way of achieving the business goals of the provider.

Thus the Forum's approach is aimed at developers rather than a traditional procurement-led standards conformance approach i.e. it aims to get in to the development bloodstream by providing developers with easy to use tools and development sandbox environments that helps them move faster by proving pre-built, management API's in a variety of technologies and formats, plus all of the developer eco-system to support this. Crucially, this also contains a library of pre-built mappings form stands API's to a variety of popular proprietary interfaces and the ability for users to place their own extensions and mappings for others to use (on a free or paid for basis).

The Forum therefore supports a range of developer tools and aids for implementing approach defined in this guide and much more detail can be found in the *Developer Pack*.

3 THE CHALLENGES OF A CLOUD BASED WORLD

GOVERNANCE BECOMES MORE COMPLEX

The nature of service delivery is changing. Historically the majority of online services were tightly coupled to specific communications networks to provide a variety of fixed and mobile communications services, and content distribution via cable TV networks. Communications service providers and cable operators usually owned and controlled all of the physical assets comprising these networks and were able, through a variety of approaches, to manage service creation and delivery end-to-end. At points of interconnection between networks, the information required to manage roaming or charging was defined by standards from the International Telecommunications Union and any a new provider who wished interconnect with others had to follow suit.

The operations and maintenance requirements of these network resources and telecom services were historically based on a mixture of manual processes and supporting management support systems and the trend in recent years has been to organize and automate these processes and systems around the Forum's Frameworx reference enterprise architecture. This has allowed the deployment of commercial-off-the shelf systems that can be more easily integrated together to provide end-end manageability. In parallel, the underlying IT software and compute resource management processes has become organized in accordance with the Information Technology Infrastructure Library (ITIL). In recent years, ITIL and Frameworx have been progressively aligned and now complement each other.

At the same time, the concept of Service Oriented Architectures (SOA) has grown in popularity as the basis of how to design and integrate applications. Frameworx is built on a services oriented design and uses standard, reusable, generic blocks that can be assembled in unique ways to gain the advantages of standardization while still allowing customization and enabling differentiation and competition at the service level.

However, implementing solutions based upon these frameworks requires strong coordinating governance and extensive cooperation between various divisions within an organization. This can be challenging to achieve even within a single organization but in the digital world, where large numbers of constantly changing heterogeneous services are operated by many different organizations, it is very difficult to dictate such standards – a more flexible approach and organic approach, based on delivering rapid benefits to the service providers operating the services is required and is the basis of the Forum's approach.

DIGITAL SERVICES ARE DISRUPTIVE

This is an umbrella term for any kind of virtualized service such as cloud computing, storage, applications, content or networking. These are disrupting the status quo and forcing rapid convergence between previously discrete sectors of networking, web and IT. Cloud based approaches unlock a number of disrupters accelerating the convergence of IT and communications. Some of these key disrupters include:

- Maturity of web services standards
- The adoption of IP and SIP in telecom and cable networks
- Growth of mobile devices routinely connected to 3G/4G/LTE or Wi-Fi networks
- Increasingly ubiquitous and higher speed broadband
- Proliferation of cloud platforms

While SOA and virtualization have contributed to the transformation of monolithic "applications" into "services" hosted on virtualized compute and network infrastructures, cloud approaches create the reality that the majority of services available for composition and consumption are not all contained within the boundaries of any one company or enterprise. First, applications began to be built following standards, and then those applications began to be exposed as "coarse-grained" services. Later services began to be further broken down into "fine grained" service components. With costs reducing and more new entrants appearing, the industry is moving towards the commoditization of services.

MARKET COMPLEXITY GROWING

The growth of the digital world is driving increasing complexity for the providers that have to deliver services that work reliably, deliver a great customer experience yet can be provided at an economic rate that delivers value for customers as well as their shareholders. They are dealing with dealing with four major trends:

- **Trend 1**: *Multiple devices from a variety of manufacturers*: Service providers are faced with the reality of having to support an array of mobile devices from different manufacturers, using several different operating systems, having several different form factors, catering to the needs of businesses and consumers. There are feature phones, smart phones, PCs/Tablets/ iPads, game consoles, and TVs with some are connected via dedicated facilities such as IPTV or DOCSIS. Increasingly devices are embedded in cars, heath monitors, security systems, home automation systems etc. using Machine-machine (M2M) approaches. All of these devices are increasingly connected via Wi-Fi or cellular broadband services such as 3G/4G or LTE.

- **Trend 2:** *Complex developer ecosystems*: Applications are core to the generation of revenue for entire value chains. Today, each mobile device platform such as Apple or Google, comes with unique application development support requirements. The back-end platforms for hosted services also have unique application development and runtime support requirements. Enterprise IT Professional developers have certain requirements related to conformance with best practices and standards for technology use, identity management, security, and privacy. The growing community of 3rd party developers, empowered by the widespread availability of cloud based platforms has different needs, particularly requiring support for more lightweight standards (e.g., REST, OAuth etc.).

- **Trend 3:** *Exponential Growth of Service APIs:* Cloud based digital services have contributed to expediential growth in the number of published APIs and Service End Points. Efficient application development requires effective mechanisms to create, catalog and publish, maintain, and consume these APIs. The dependencies that are created within applications that rely on the incremental bits of functionality must be understood.

- **Trend 4:** *Reality of Multi-Cloud Service Delivery:* Virtually every service has other services upon which it depends or creates dependencies has soon as it is consumed. It is very rare today to find 100% of the resources living in a "walled garden". This collection of services typically resides in multiple different service domains and a service owner may not, in fact, be able to directly control prerequisite services. Service delivery today requires multiple clouds and multiple service domains to work together in harmony throughout the entire lifecycle of that service.

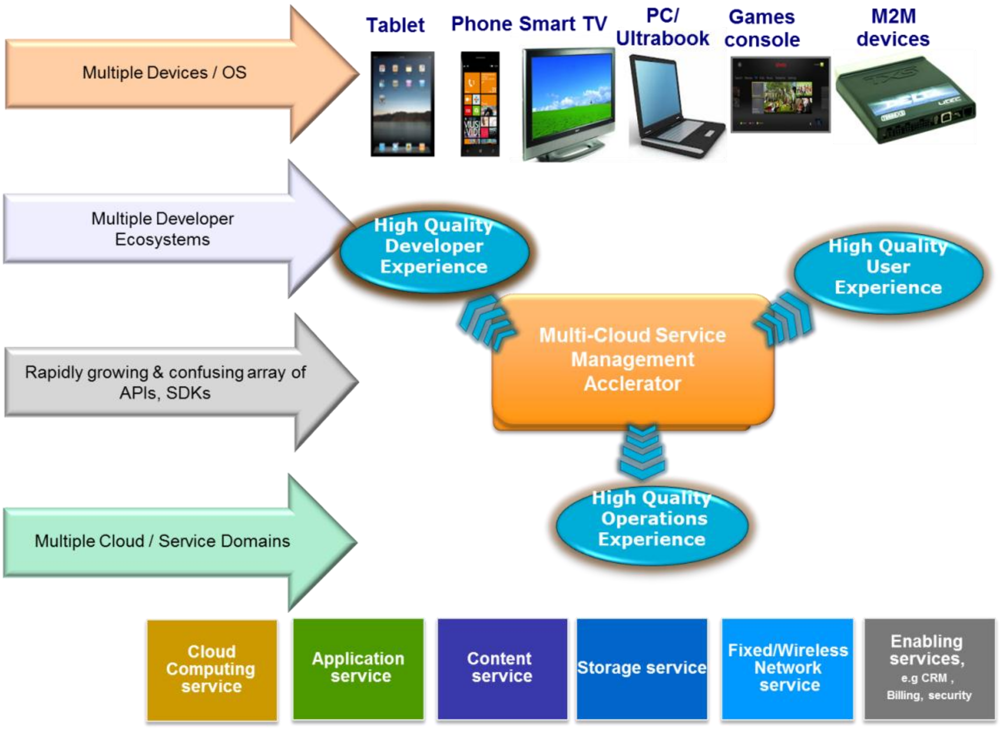

Multiple Devices / OS
Tablet
Phone
Smart TV
PC/ Ultrabook
Games console
M2M devices
Multiple Developer Ecosystems
Rapidly growing & confusing array of APIs, SDKs
Multiple Cloud / Service Domains
High Quality Developer Experience
High Quality User Experience
Multi-Cloud Service Management Acclerator
High Quality Operations Experience
Cloud Computing service
Application service
Content service
Storage service
Fixed/Wireless Network service
Enabling services, e.g CRM , Billing, security

Figure 4 – Market complexity and the need for high quality experiences

These trends continually stress the provider's objectives to deliver a high quality operational, user and developer experience and these are the major objectives behind the Forum's Multi-cloud accelerator program.

- A HIGH QUALITY USER EXPERIENCE: delivering a high quality user experience is essential to gain and maintain customers. Whatever the service or content the user is consuming; that experience must meet the expectations of the customer and is true for both consumers and business users but it is particularly acute in enterprise situations where the dependency on the service is high. The Multi-Cloud Management Accelerator describes provides tools to implement and measure performance against established standards and methodology to iterate towards better user experiences.

- A HIGH QUALITY DEVELOPER EXPERIENCE: The developer community must be provided the tools and guidance needed to build the types of applications needed to deliver great experiences. In many cases, the experience cannot just be a "best effort". Each service must be buildable in an efficient manner that facilitates combining into more complex service compositions. Major integration efforts must become progressively less necessary at this level. Therefore, developers need governance, documentation, tooling, and wizards to guide them in the development of services that are much easier to manage individually and combine into service compositions / mash-ups that are also manageable and end-to-end,

- A HIGH QUALITY OPERATIONS EXPERIENCE: All of the stakeholders in the service delivery process need to be able to manage their services efficiently and effectively even though they rely on components hosted by different services providers in different clouds. The ability to readily manage services that span multiple clouds and resource domains is critical to achieving revenue objectives. Service providers and their partners need transparency and visibility across value chains in order to have the confidence to leverage efficient multi-cloud ecosystems to deliver core value added services to businesses and consumers.

END-TO-END SERVICE MANAGEMENT IN A COMPLEX ECOSYSTEM

As the digital world advances, the number and types of service and service provider are mushrooming and in order for them to work together in an open ecosystem, each needs to work with the other and that means having standards. The basic connectivity and interoperability standards such as IP and HTML5 are key building and are widely adopted. However, the area of management and operational standards is much less mature and there are significant barriers to managing a service on an end-end basis across multiple component services which impacts operating costs, agility and the customer experience. These barriers not only impose operational difficulties, they limit the extent to which the digital economy can grow by allowing open 'trading' between any digital service provider.

End-end service operational issues have been largely overlooked in the early stages of the digital economy which has been built largely on 'best efforts', consumer services. However, as the use of cloud based services for critical business needs rises; importance of service integrity, security, reliability and a high quality customer experience is also rising. The issue of the costs of managing services is also becoming critical, particularly to reduce costs of resolving issues between providers.

This impacts the automation of key operational issues such as:

- managing contractually-binding service level agreements and problem resolution when multiple players are involved
- managing charging and settlements so that everyone in the ecosystem gets what they are owed, with proper auditing and ensuring that billing is at the right rate
- managing the product lifecycle from new product concept to eventual withdrawal across an ecosystem
- managing risk management and fraud
- managing order management, authentication, security etc.

Any type of digital services needs agile, low-cost operations but this is especially required where services are aggregated or 'mashed' together into overall solutions. Increasingly, no service provider 'owns' all the services that make up total value being delivered to a customer and this trend is likely to continue with more and more service specialization. There are several reasons for this: one important driver is associated with core competencies and the economics of delivering services at scale. Different types of service providers are specializing to focus on areas where they have unique capabilities they can deliver at such scale. On the other hand, customers increasingly want sophisticated solutions that can be achieved most economically by combining the best services exposed by several different providers into a new value added service chain.

MINIMIZING CUSTOMER INTERACTION COSTS

Every customer initiated contact that involves the service provider responding with a human customer service agent is a costly event and it is an economic necessity to minimize events that cause customer contacts in the first place. They do this by designing their services and operations to be largely based on customer self-service, to be highly reliable and minimize failures sufficiently to meet customer expectations – the primary reason why certain components are built to 99.995% reliability standards or better.

However, when things do go wrong, it is important that the provider knows about the problem before the customer and can set about resolution with the least user impact. Minimizing the need for user contact, especially when the customer calls to complain, is a key element of operational design and is a major driver for resolving the problems of managing across service boundaries.

Another reason for needing management capability that can span the entire service delivery chain is the need for the provider to have all of the appropriate information immediately accessible and actionable either on a customer self-service portal or where appropriate, via a customer service agent. Functions might include Service Order History, Trouble / Performance / SLA History, and Billing information. The goal is to be able to handle during the initial contact any questions concerning billing, service configuration, faults, or performance. This includes being able to see via service dashboards and event lists what has transpired and to drill down to obtain greater details on any significant item. It should also be possible to initiate an order for new or changed service configurations as a possible outcome to the customer initiated contact.

In the digital world, customer self-service is vital and needs to provide sufficient information and capabilities to preclude the generation of a customer initiated call into a work centre. This is also very important to meet the needs of the service developer building, deploying, or maintaining a service leveraging cloud computing resources; a self-service portal should successfully guide the developer through all business and technical interactions with the cloud. This includes the process of setting up an account, consuming service APIs, building an application that will work reliably, deploying the application to the cloud, configuring cloud resources efficiently, and being able to visualize usage and performance via dashboards.

There are two primary measures that determine the success of service delivery businesses in the digital services arena which are critically dependent on getting access to all of the relevant management information in play, regardless of which component service actually holds the information:

> CUSTOMER SATISFACTION – Will the customer be satisfied with service and if there is a problem, will they be satisfied with level of customer service they receive to rectify the issue? If the self-service portal or call agent delivers credible knowledge about a service problem and is capable of decisive and effective action, customer satisfaction will remain high and customer churn will be suppressed. To achieve this, the proper information and tools must be available, accurate and effective.
>
> OPERATIONAL EXPENSE – How many steps are necessary to address typical issues? If approximately 80% of problems can be handled effectively at the first point of contact, whether a self-service portal or an agent, then the cost incurred in handling problems is minimized and the profitability of the service is protected. However, if the agent does not have useful tools in front of them and can only create a trouble ticket and pass that off to another work centre for action, then operational expense can skyrocket making the service in question unprofitable. This is multiplied by orders of magnitude if a problem has to be manually 'chased' across multiple service providers each of whom measure service quality in different ways and have different problems resolution approaches. If the error is not capture in real-time by recording the event along with actionable fault and performance data permitting root cause analysis, the expenses of handling trouble reports and acting on them after the fact may exceed revenues.

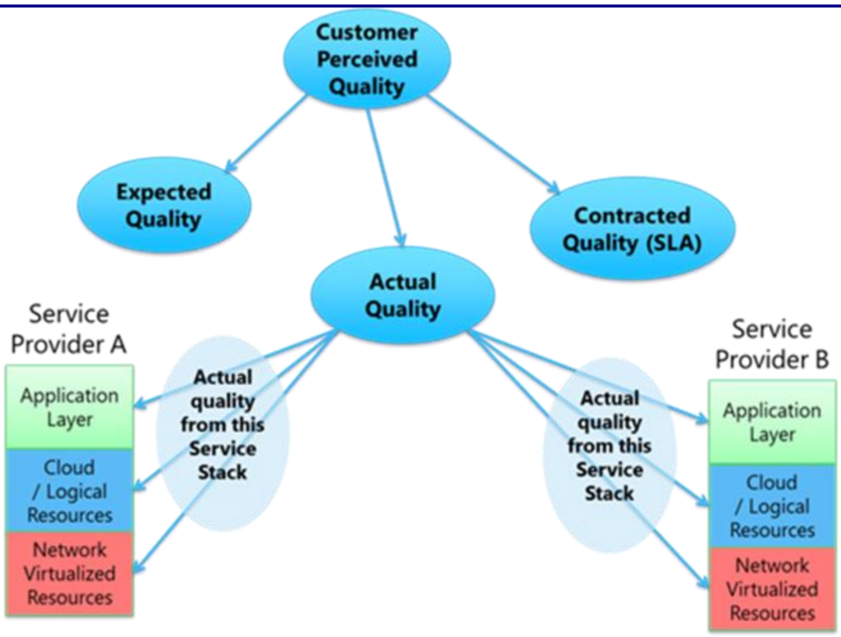

Customer Perceived Quality
Expected Quality
Contracted Quality (SLA)
Actual Quality
Service Provider A
Application Layer
Cloud / Logical Resources
Network Virtualized Resources
Actual quality from this Service Stack
Actual quality from this Service Stack
Service Provider B
Application Layer
Cloud / Logical Resources
Network Virtualized Resources

Figure 5 - Components of Service Quality

SERVICE COMPLEXITY IS GROWING

The market is increasingly fragmenting into discrete services as service providers introduce next generation network services combined with cloud hosted. Increasingly, communication providers are exposing network capabilities via a Service Exposure Layer. 3rd party applications can include a service logic component, possibly hosted on a cloud infrastructure, accessed via a client application marketed via the appropriate Application Store or via web browser user interface.

In many cases these applications invoke services hosted on other environments. The services may, in turn, implement one or more calls to other services supported by underlying wholesale business relationships between several service providers. It is important to understand that the challenges of end-to-end service management exist regardless of the set of business partnerships and the problems multiply exponentially as the number of partners in the service delivery chain increases.

Take the case of an enterprise IT department, who sources a managed service from a service provider to handle the needs of their mobile sales people, and includes functions such as mobile phone, messaging, appointments, sales order processing as well as basic functions such as e-mail. This type of managed service would be composed of a number of component services sourced through different specialist providers. Such a service would normally be covered by service level guarantees as disruptions to the service as they would impact revenue if the service wasn't delivering to its specification.

The difficulty in this example is that the service provider holding responsibility for the end-end service level guarantee will almost certainly know less about the quality of service being delivered than the customer because they will have limited or no visibility of the service quality being delivered by each component service. This means that the service provider cannot proactively manage the service to ensure a consistently high quality customer experience. Further, reported breaches of the service level agreement will probably have to be handled manually with expensive and time consuming consultation with each component service provider. The problem is further compounded if each service provider in the chain uses different ways of measuring quality of service, different escalation processes and incident reporting approaches. The result will be the inverse of what a profitable provider wants: dissatisfied customers and high operating costs.

THE DIGITAL SERVICES LIFECYCLE AND VALUE CHAIN

At this point it may be useful to review the lifecycle within which digital services are developed, and the value chain within which they are deployed. Figure 6 depicts the digital services lifecycle, which consists of three phases, described in detail below.

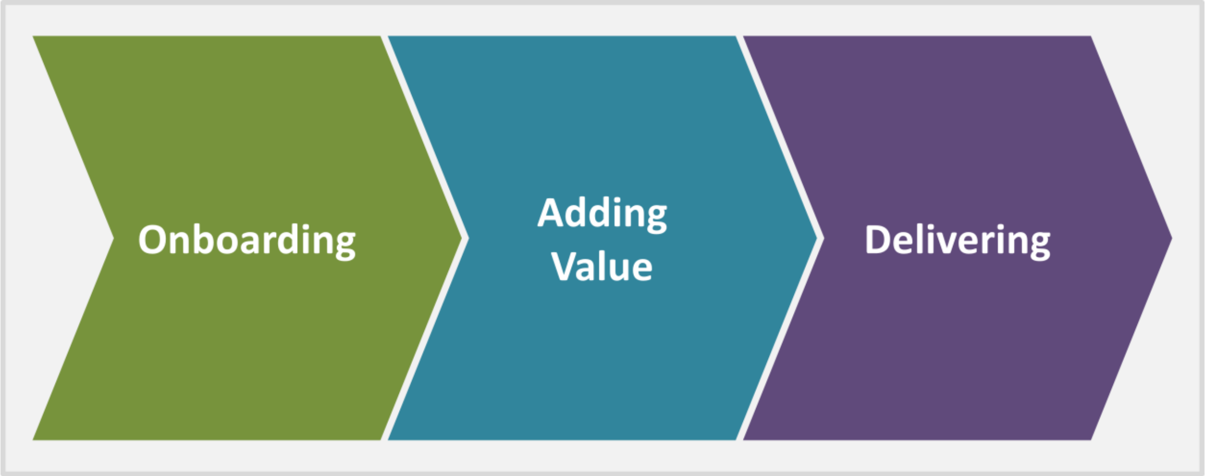


Figure 6 - The digital services lifecycle

The onboarding phase, shown in Figure 7, is the phase in which digital services are developed and/or imported by a DSP. The diagram shows four possible sources for digital services:

1. External: obtained from an ISV and brought permanently in-house
2. Federated: obtained from a SaaS ISV
3. Syndicated: obtained from another service provider
4. Internal: developed in-house by the service provider

Federated services are exposed by a DSP but actually run in the cloud of a SaaS ISV. Similarly, syndicated services are exposed by a DSP but actually run in the cloud of another DSP or a SaaS ISV, depending on their origin. Although they often provide compelling cost advantages, both federated and syndicated services require special attention because they introduce significant management issues not found with externally purchased or internally developed services.

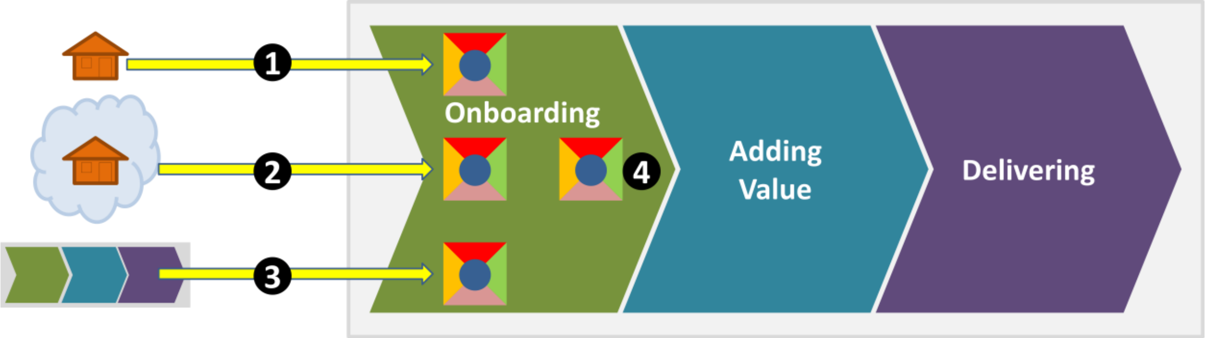


Figure 7 - The on-boarding phase

The second phase of the digital services lifecycle is adding value. The purpose of this phase is to add value beyond simply aggregating other services - creating something that is more than the sum of its parts. A DSP can add value to a service in several ways:

- Configuration: setting characteristics that differentiate the service
- Branding/Co-branding: applying a trusted or popular brand to increase appeal
- Packaging/Bundling: combining a service with others to improve value provided to customer
- Mash-up: discussed in previous pages
- Commercialization: providing access to a market that is otherwise unavailable to a supplier

The third phase of the digital services lifecycle is shown in Figure 8. As shown in the diagram, digital service providers can deliver digital services in two ways:

1. Direct delivery: the service is delivered directly to a customer of the digital service provider
2. Syndication: the service is delivered indirectly to a customer through one or more other digital service providers

It is important to note that these two delivery methods are not mutually exclusive. In fact, the ability to deliver a service through multiple channels, possibly under multiple brands and with different pricing and packaging, can be key to monetization and cost recovery.

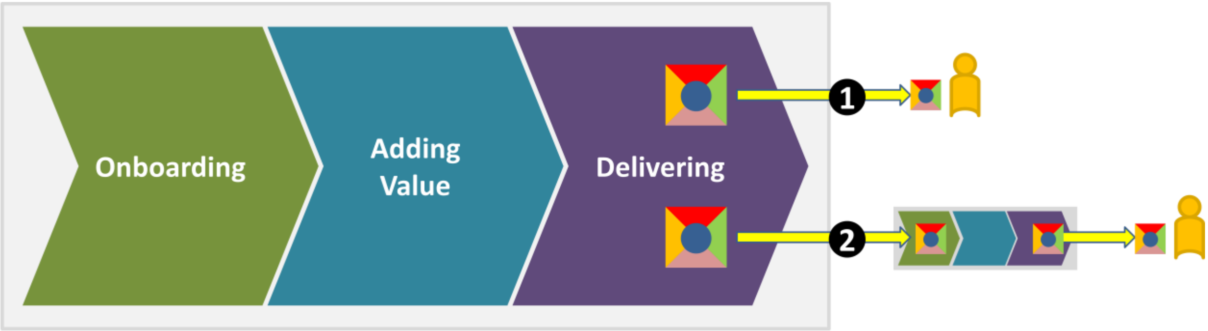


Figure 8 - The delivery phase

The digital services lifecycle model can be used to construct an example that highlights the issues and opportunities faced by DSPs.

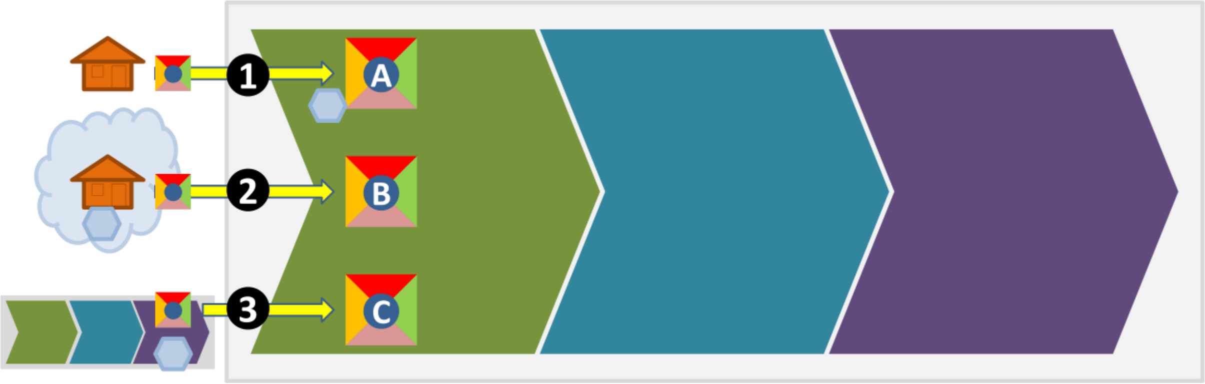


Figure 9 - Onboarding example

Figure 9 shows the onboarding phase of a DSP's creation of a new digital service, during which the DSP onboards three digital services:

1. Service A is purchased from an ISV
2. Service B is federated from a SaaS ISV
3. Service C is syndicated from another DSP

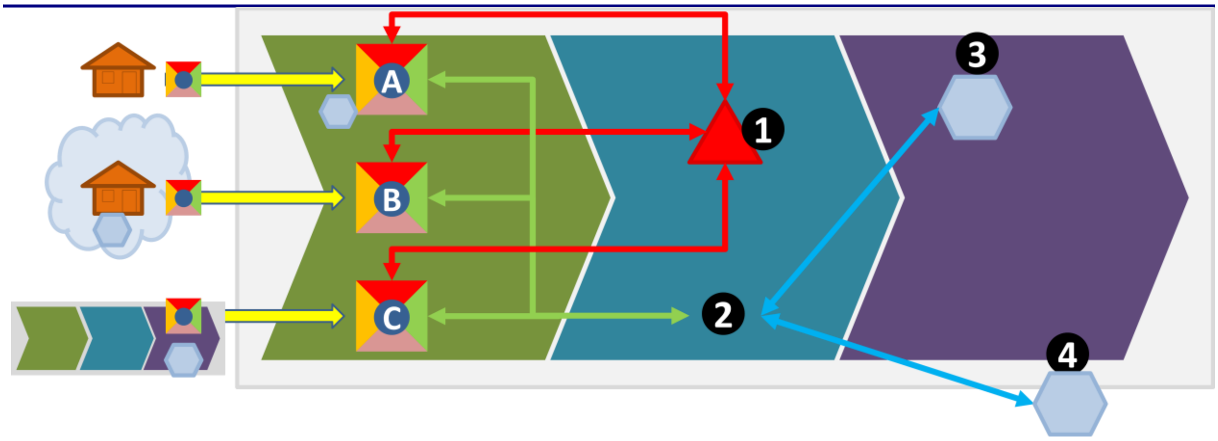


Figure 10 - Adding value example

The example is continued in Figure 10, showing the DSP aggregating the functional interfaces of the three services to create a new service (1), assembling the management interfaces of the services (2) and linking them to internally supported (3) and externally supported (4) management systems.

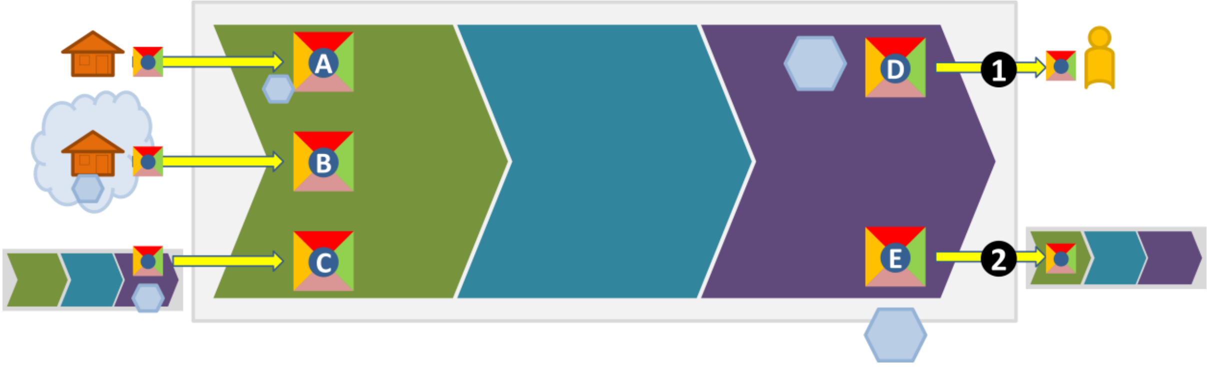


Figure 11 - Delivery example

The delivery phase of the example is shown in figure 11, as the DSP delivers the managed cloud mashup as a new service in two different configurations and pricings:

1. A branded service D sold directly to customers (1) and managed with internally provided business support services.
2. A co-branded service E syndicated to other DSPs (2) and managed with externally provided business support services.

Rather than develop the entire service in-house, and likely reinventing the wheel in the process, the DSP has leveraged less-expensive third-party services, focused its own resources on adding value, and further leveraged the resulting new service by taking advantage of syndication along with direct delivery. While these actions all make sense from a competitive and budgetary standpoint, the DSP also faces new management challenges, as shown in Figure 12:

- Management systems for service D should be linked to the management systems that support services A (1), B (2) and C (3).
- Management systems for service E should also be linked to the management systems that support services A (1), B (2) and C (3).

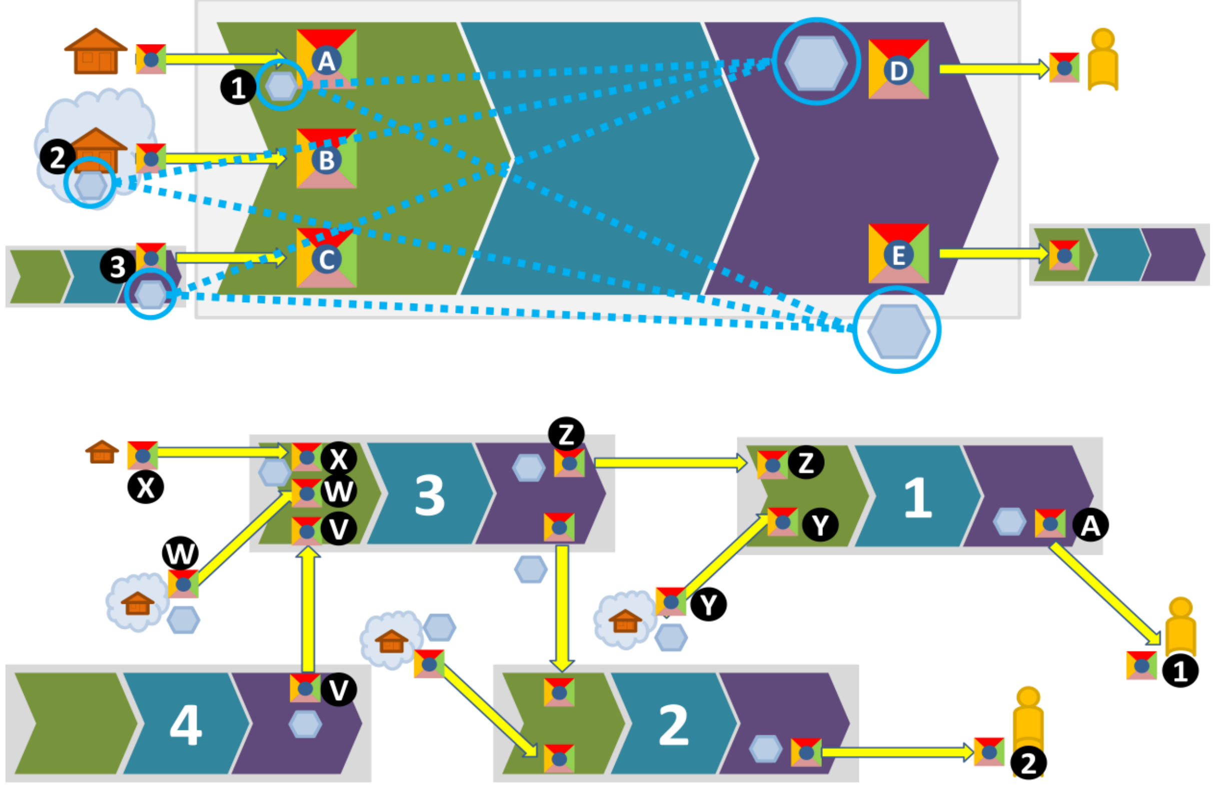


Figure 12– Complexity of managing multiple component services

In

Figure 12 above, communications service provider 1 is acting as the service aggregator and point of sale to customer 1, and is sourcing in multiple application services from a value chain of other service providers and ISVs, each of whom may use underlying cloud computing, storage and communications providers as part of their own service delivery. The ultimate quality of experience for customer 1 is the net sum of all of these services. There may be limited visibility and control over the management and operational aspects even within a single service domain, but certainly between such domains unless all have agreed to expose a common Simple Management API (SM-API) and service assurance metrics.

The Forum's approach provides a way to expose and share management information between the service providers so that all parties in the service delivery chain can understand the quality of service being delivered and that the customer can get also get a composite online dashboard of service performance and interact with the service. Ideally, no manual intervention is required except in exceptional circumstances.

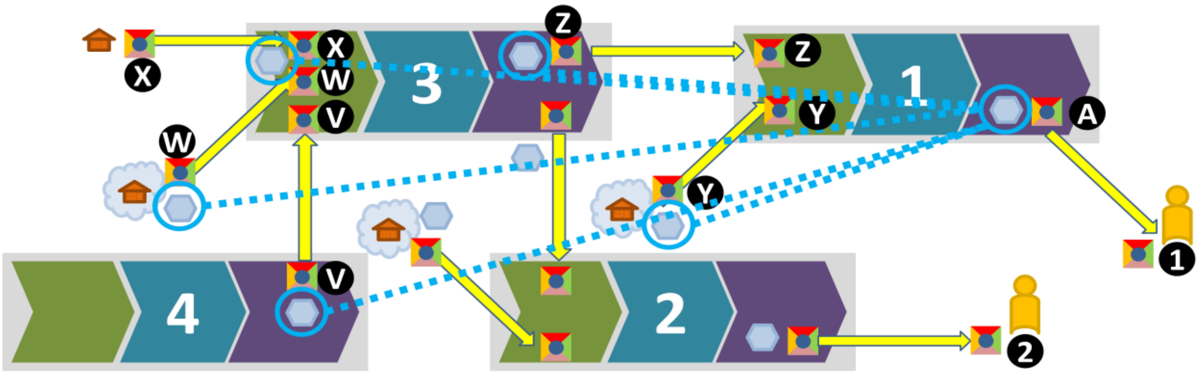


Figure 13– Complexity of managing syndicated services

The challenge and the focus of the TM Forum is to enable all of the required management system interactions represented by the dotted lines, by linking each digital service to its management system with the SM-API, and by specifying the inter-management-system interfaces to exchange management information.

4 THE MULTI- CLOUD MANAGEMENT ACCELERATOR PACK

The core of the Forum's Multi-Cloud Management Accelerator is the Simple Management API (SM-API) and the concept of Service Lifecycle Management (SLM). These were developed by the Forum's Software Enabled Services team and draws on other key Frameworx best practices and are defined in detail in a series of documents listed in the References section.

The Multi-Cloud Management Accelerator has 3 key elements:

- The Frameworx SM-API is an API that reveals how to manage any given service. It defines for developers a key design pattern for including management capabilities in a service as they design and build it. It enables service providers to manage each service or composition of services in an efficient manner.
- The Frameworx SLM defines best practices and requirements for establishing a role based software/services factory and a Lifecycle Management Meta Data model. Aligned to other aspects of Frameworx and ITILv3 2011 Service Lifecycle Management Governance, this architectural component aids in the management of APIs through their lifecycle impacting both service creation and runtime processes.
- A developer ecosystem providing software, tools, reference implementations, a developer 'sandbox' and other capabilities to support rapid deployment of the solutions outlined in this guide.

THE SIMPLE MANAGEMENT API

The Frameworx SM-API is an API that provides a set of capabilities exposed by a digital service through which it can be managed and may be implemented as a part of the service itself. It may be supported by an underlying Management Support Service (MSS[1]) that provides a management capability on its behalf or via a Service Delivery Broker (SDB) which provide a mediation function between different management systems. It is also possible for an SDB to emulate a Service's Management Interface; to expose a virtualized SM-API making it available through a mediation component such as a message broker or gateway. Service Delivery Brokers are described in more detail in Section 5.

Through mediation, the SM-API will then appear to its consumers as if it was originally designed as part of the service enabler. This can be necessary because it is not always possible to change the underlying service enablers or simply because an SM-API capability needs to be composed by assembling a composition of different services.

SERVICE LIFECYCLE MANAGEMENT

In addition to the concept of a consistent design pattern for an SM-API, the Multi-Cloud Management Accelerator Pack provides a consistent approach to service lifecycle management. This includes representative definitions for the phases a service passes through from concept to retirement as well as a Lifecycle Management Metadata (LMM) model which can hold all the data about a service throughout its lifecycle.

The Service Lifecycle Management definition consists of three parts:

- Management Phase of the service – ITILv3 2011 aligned lifecycle phases. This is the most well-defined and covers the following phases:

[1] A Management Support Service is a service embedded in any Business or Operations support system (BSS/OSS) that consumes the SM-API function

  - Concept
  - Design
  - Deploy
  - Operate
  - Retire
- Management Dependencies of the service – resources that are prerequisites for the service to function.
- Additional information about the SM-API

SUPPORTING EFFECTIVE CUSTOMER EXPERIENCE MANAGEMENT

Customer Experience Management (CEM) has become a critical area of focus for digital service providers that are concerned with customer loyalty and retention, and with maximizing both customer satisfaction and ARPU at a sustainable cost. Although CEM is not directly implemented by multi-cloud service management, it is impossible to implement a successful CEM program in a multi-cloud environment without addressing the management requirements of digital services and linking their management systems together effectively to enable the multiple stakeholders in a value chain to interact. The key to successful CEM is the ability to collect metrics and failures in a timely, comprehensive, and economical manner. Metrics and failures reported by a digital service may refer to other services, allowing the service provider to assemble a comprehensive view of service performance, usage, and operation. The SM-API supports the collection of metrics and failures in an interface that is lightweight yet contains all of the essential data elements.

This section describes a scenario that may be useful in illustrating how the SM-API supports CEM. A DSP is building up its catalog of enterprise services. The DSP wants to leverage third party services to reduce costs, but also to add its own distinct value to each service, leveraging its unique advantages (e.g. trust, identity, account management, billing, QoS, SLA). One of the enterprise services under development coordinates a set of third-party-provided services to make reservations for enterprise employees in a way provides value (e.g. cost savings, expense control) to an enterprise. The DSP decides to try out the concept on a limited basis, using a subset of possible capabilities, as a restaurant reservation service, with the intention to expand to more capabilities and other kinds of reservations if the trial is successful.

The DSP's initial implementation of the new enterprise service is shown in Figure 9. The reservation manager, developed in-house by the DSP (DSP A in the diagram), coordinates the operations of two third-party services that are federated or syndicated by DSP A: a restaurant finder service provided by DSP B, and a reservation maker service provided by DSP C. The reservation manager supports a mobile app intended for enterprise employees, allowing employees to select a location, date, time and purpose (e.g. business entertainment versus employee meal). The reservation manager obtains a list of restaurants from the restaurant finder service, applies business rules maintained by enterprise managers through a second mobile app, and presents a filtered list of permitted restaurants to the employee, and attempts to reserve the selected restaurant using the reservation maker service.

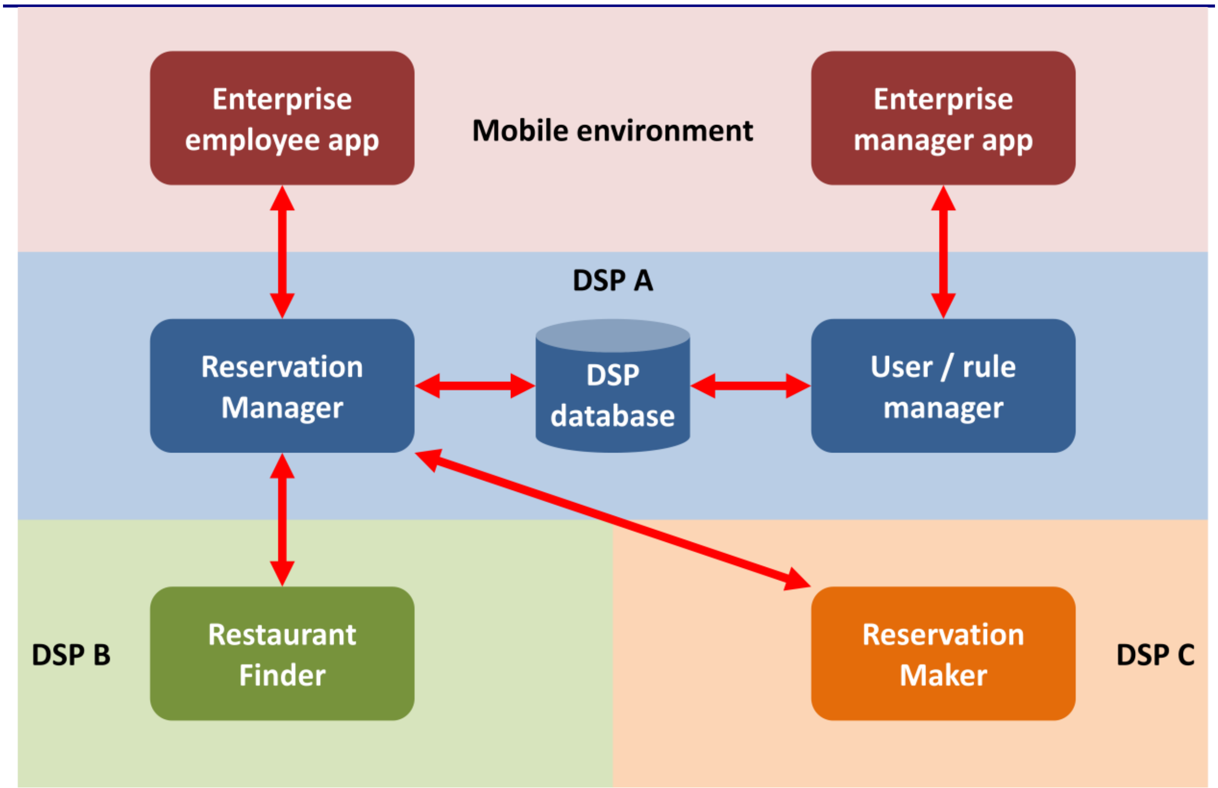


Figure 14 - An enterprise service composed from third party services

In order to meet its SLA guarantees and to maximize customer experience, DSP A will obviously monitor the performance, usage and reliability of its in-house reservation manager service. But it also needs visibility into the same aspects of the federated / syndicated third-party services. This presents a problem, as those services are not running in its cloud and are not under its control. Metrics and failures collected via SM-API can provide the required visibility, as shown in the next two diagrams.

Figure 16 shows one of several ways that the SM-API can be used to obtain essential information on metrics and failures for management systems. In this example, each DSP uses the SM-API to monitor its own service. Because the SM-API provides identifiers that tie metrics and failures to extended definitions in repositories, each DSP is able to make full use of the information, even though the data passed through the SM-API is extremely lightweight. The information obtained by each DSP through the SM-API enables intelligent communication between their respective management systems to manage SLA guarantees, conduct multi-party settlement, etc. In the case of DSP A, this information is also critical to fulfilling the goals of its CEM initiative.

Another possible configuration for the collection and use of information on metrics and failures is shown in Figure 17. Here the customer-facing DSP is monitoring not only its own service, but also those of its partners, with SM-API. Using this configuration, only the customer-facing DSP needs to maintain a repository of metrics and failures definitions. However, the need for communication between its management systems and those of its partners remains the same.

There are many other possible permutations of metrics and failures collection and utilization, and the possibilities become more numerous and more complex as the value chain discussed in section 3 grows in length and complexity. As this happens, the role of the SM-API in supporting inter-management system interaction and in supporting successful CEM becomes increasingly important.

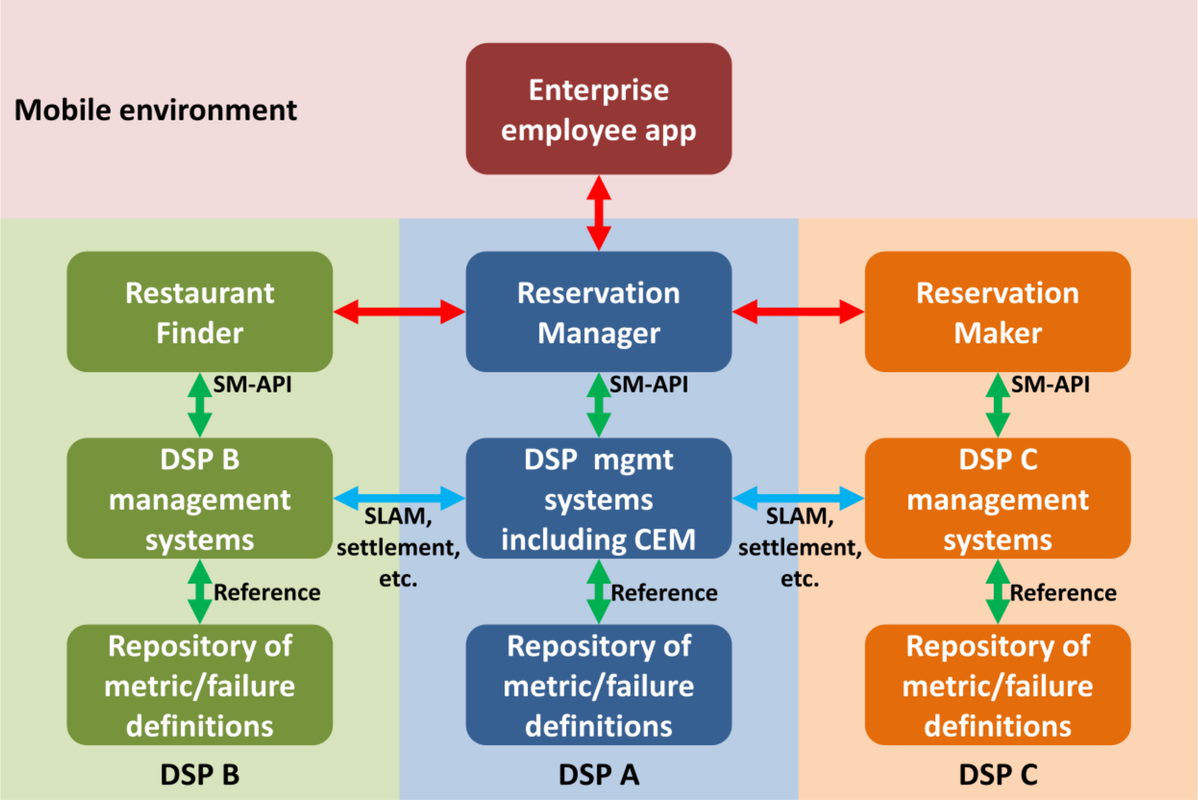


Figure 15 - Independent collection of metrics and failures

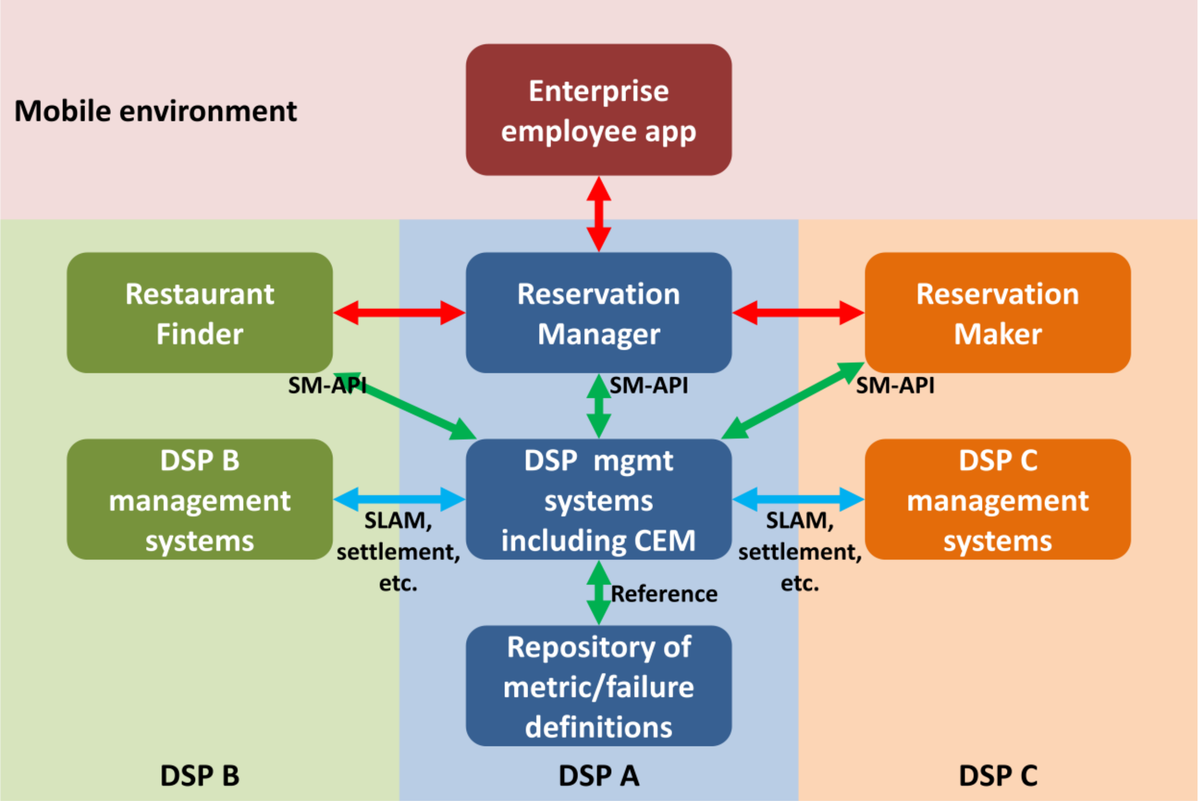


Figure 16 - Centralized collection of metrics and failures

5 SERVICE DELIVERY PLATFORMS AND SERVICE DELIVERY BROKERS

This section is not intended to be a comprehensive overview of Service Brokers or Service Delivery Platforms but to explaining key aspects particularly relevant to Multi-Cloud Service Management. The terms *Cloud Service Broker* or *Service Delivery Broker* are used when focussing on exposing a management layer that spans multiple platforms or clouds.

SERVICE DELIVERY PLATFORMS

The concepts behind a Service Delivery Platform (SDP) are fairly straightforward provided the services are all contained within one well-defined boundary; for example, an SDP for one set of functions in a mobile operator’s network. SDP’s are generally proprietary in design and the Forum has worked to define a common Service Delivery Framework that can enable a community of service providers each managing their own domain of services, to collaborate and deliver manageable services controlled by local SDPs and associated BSS/OSS for management functions. The core focus of this work has been the ability to manage the resulting services end-to-end and this has been very valuable in extending some of these concepts to the management of multiple cloud services.

The SDP approach is well suited to a world where a provider owns and controls all of the assets that make up the service (the so-called “walled garden”) and requires services to be built to very specific standards especially on how they interact with other services and the network before being allowed to operate. This particular type of certifying was sometimes referred to as “on boarding” a service onto the SDP.

However, the digital world is evolving significantly different business models and once service providers begin to meld together services from multiple domains, problems abound. Setting aside service interoperability for a moment, the other major issue has to do with lifecycle management and the details of operations and maintenance of the services and their components.

New issues begin to crop up with a service bundle consisting of discrete services from multiple domains and get more challenging when the services from different domains will be actually integrated together in a loosely coupled service mash-up.

CONVERGED SERVICE DELIVERY BROKERS

Service Delivery Platforms first evolved in the telecom industry and some communications players have attempted to extend those SDPs to also govern the creation, deployment and operation of web services. This was greatly accelerated with the original vision of IMS and its notion of an application layer hosting services governed by underlying IMS control functions for policy, charging and rules. There have been several attempts at expanding telecom/network services centric SDP environments to assume overall lifecycle management and runtime control over a broader array of services including those being abstracted by enterprise IT and web/cloud developers.

This has led to the concept of a converged service delivery platform or broker to cover a broad spectrum of service types. In addition to Communications Service Provider and Cable Operators, new types of service providers offering internet web hosting, cloud services, and the social network platforms have expanded the requirements for service lifecycle management and runtime from a telecom network centric topic to much broader digital landscape. Furthermore, the evolution of services orientation approach to architecture contributed additional needs to accelerate and manage more effectively at the Integration Framework layer.

While the original requirements of an SDP still exist, solutions now need to appeal to these broader needs of the digital world as a whole. This converged Service Delivery Broker can provide a set of reusable core capabilities (services) to both speed development processes and to support runtime operations. Some of the reusable services such an SDB can provide include:

- common transports,
- bindings and protocol mediation,
- support for all needed message patterns,
- common tasks such as security & access control,
- event processing engine,
- routings,
- performance / traffic monitoring,
- mechanisms for real time visibility into performance and usage including Dashboards.

INTEGRATING MULTIPLE SILOS

Although the technical details, platforms and tools tend to be different for every major service 'silo' shown in Figure 17 below, each loosely adheres to the following architectural concepts:

- Fine grained service creation and management using tools often unique to that silo.
- Coarse grained service abstractions typically via SOAP and Web Services interfaces.

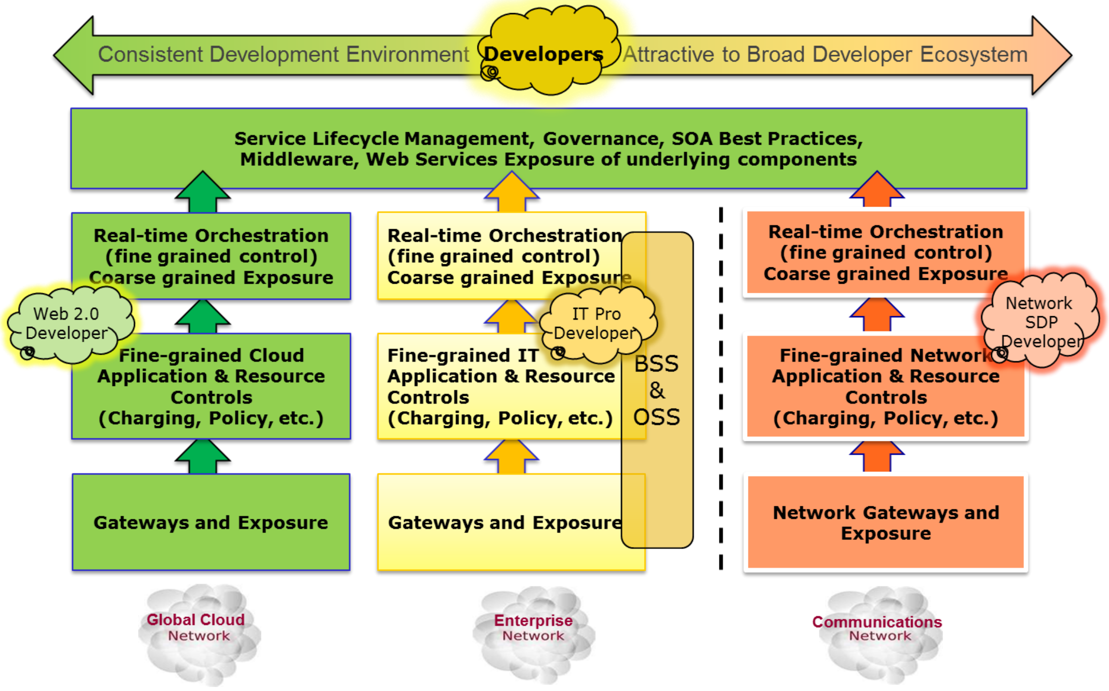

Consistent Development Environment
Developers
Attractive to Broad Developer Ecosystem
Service Lifecycle Management, Governance, SOA Best Practices, Middleware, Web Services Exposure of underlying components
Real-time Orchestration (fine grained control) Coarse grained Exposure
Web 2.0 Developer
Fine-grained Cloud Application & Resource Controls (Charging, Policy, etc.)
Gateways and Exposure
Real-time Orchestration (fine grained control) Coarse grained Exposure
IT Pro Developer
Fine-grained IT Application & Resource Controls (Charging, Policy, etc.)
Gateways and Exposure
BSS & OSS
Real-time Orchestration (fine grained control) Coarse grained Exposure
Network SDP Developer
Fine-grained Network Application & Resource Controls (Charging, Policy, etc.)
Network Gateways and Exposure
Global Cloud Network
Enterprise Network
Communications Network

Figure 17– Multi-Cloud Service Management - one Framework for CSP, IT/Enterprise & Cloud

On the right-hand side of the diagram is the communications environment and where Service Delivery Platforms excel at creating and implementing services that require real-time event processing as well as policy, charging, and rules functions in the course of setting up connections and delivering services.

In the middle column is the enterprise IT technology stack and in this domain, IT professionals design and implement mission critical applications appropriate for their industry. Each industry sector may have their own interpretations of an Enterprise IT Reference Architecture that in turn typically leverages best practices (such as from ITIL and TOGAF) for management and governance. Many older legacy mainframe, client/server, enterprise service bus environments live in this space as well as newer service oriented implementations.

Finally on the left side is the cloud service developer and while an IT Professional implementing enterprise service oriented architecture solutions could also be represented by this section of the drawing, the intent is to emphasize newer Web 2.0 renditions of services and service compositions. The bulk of 3rd party developers being recruited into various developer ecosystems live in this space. The trend here is towards lighter weight interfaces such as REST using JSON.

It is important to note that a digital service can use services in all three of the above silos.

API MANAGEMENT

APIs are becoming the critical common currency of service creation and delivery. Developers have been creating interfaces to applications and services for years. However in the absence of the structure that a service delivery broker can provide, these APIs provide only a limited capability for application integration, assume or favor a specific programing language, contain programing techniques that are not best practice from an industry level (example non W3C compliant code generation) and often require significant system integration efforts to implement.

When an organization truly adopts a service orientation there is a very significant material impact on how APIs are built and how they are used. APIs developed without a true services orientation have a tendency to be coarse grained providing a limited exposure to the underlying fine grained features. Often much of the actual work flow in applications happens outside of these coarse grained interfaces. Therefore, there tends to be a fewer number of APIs that are only used for a limited subset of use cases. For instance, a function might be exposed externally via a JAVA API but internal users of that service use different, more feature rich interfaces.

When a true services orientation is adopted, the same APIs become used by both internal and external users. Policy and rules evaluation processes become reusable supporting services invoked in conjunction with the use of a service creating policy-based use governance enabling secure reusability. As the number of internal only APIs is reduced, the number of published reusable service APIs can expand greatly. This leads to a new requirement of being able to manage efficiently a growing catalog of services throughout their lifecycle.

ADDRESSING SCALE

Cloud services enable many different entities, from large businesses to individual developers, to host and publish and ever increasing number of APIs. The proliferation of APIs has been very rapid and is accelerating creating confusion and requirements for API inventory management control. A systematic approach is required to guide the creation of APIs according to a set of common guidelines.

A management system is required to perform common functions that can be consistently applied across all APIs which include:

- Service Versioning
- Service Policy
- Service Abstraction
- Service Routing and Transport
- Service Management

Difficulties arise when the number of services becomes very large. Custom point-to-point integration can get the job done as long as there are not frequent changes but when the APIs number in the thousands; there are frequent updates and version control becomes an issue, or when the number of service provider domains expands into many to many relationships, a much more systematic approach is needed for API Service Lifecycle Management, Integration, and Runtime operations. The growth of M2M based services and devices drives home the point that a much more efficient and reliable means of service lifecycle and operations management is required.

A broker is one mechanism that addresses these operational problems. A service delivery broker or cloud service broker does this by providing some of the core functions defined in the Integration Framework of Frameworx as reusable services. Furthermore, by facilitating the process of creating and integrating management functions exposed by each Simple Management API, a broker can reduce the need for custom integration that would otherwise be required to effectively manage these new service compositions.

An SDB can provide the following API management functions:

- Service API Catalog Functions
- ITILv3 2011 and Frameworx Aligned Service Lifecycle Management and Metadata Model Management
- Tools to Implement standard Simple Management APIs
- Wizards to support a true services orientation
- Contract first development methodology and Governance
- Runtime Management Operations

Depending upon the service in question, the broker could be involved in helping to manage the lifecycle management of services APIs, the runtime management of the Service APIs, or both.

Depending upon the context of the service, and the design decisions of the service provider or enterprise implementing enterprise service oriented architecture, the broker could accommodate the allocation of specific functions to different resources. For instance, in the case of a set of cloud services that are largely hosted web services, the broker could provide certain necessary policy, charging, and rules based functions directly. Alternatively, some functions could instead be handled by a dedicated billing BSS/OSS application. Alternatively, in the case of certain services leveraging telecom service exposure layer, it may well be optimal to leverage the IMS Policy, Charging, and Rules Function (PCRF) in the network.

This approach enables the reference architecture to provide three key value propositions mentioned earlier:

- A High Quality User Experience
- A High Quality Developer Experience
- A High Quality Operations Experience

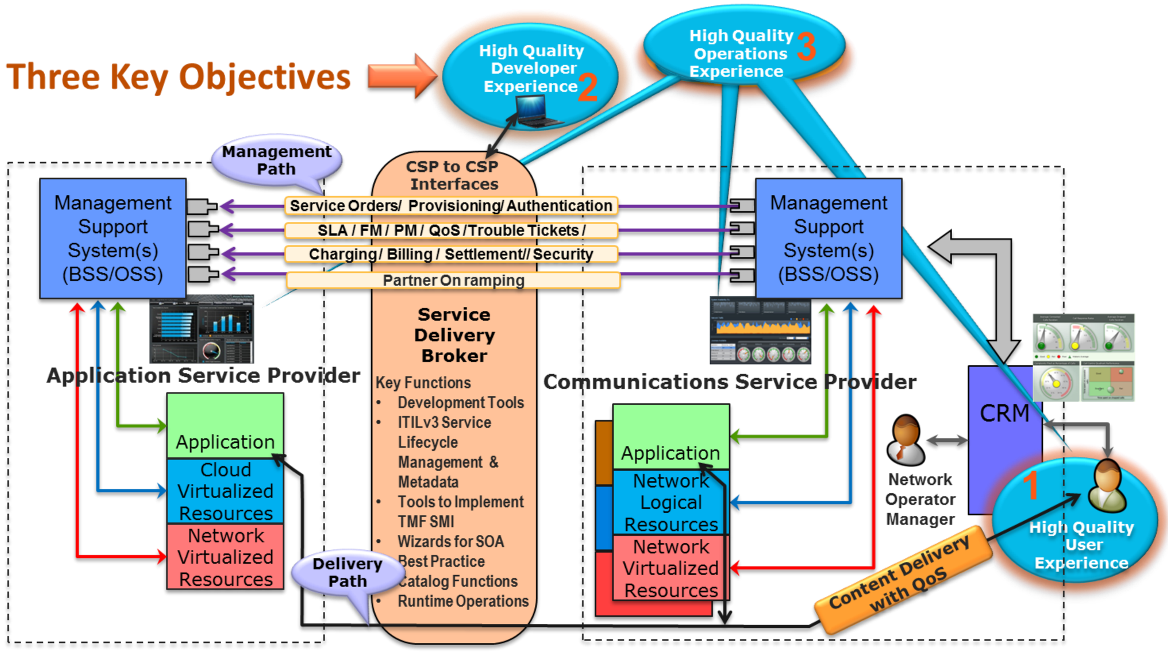

Three Key Objectives
High Quality Developer Experience 2
High Quality Operations Experience 3
Management Path
CSP to CSP Interfaces
Management Support System(s) (BSS/OSS)
Service Orders/ Provisioning/ Authentication
SLA / FM / PM / QoS / Trouble Tickets /
Charging/ Billing / Settlement// Security
Partner On ramping
Management Support System(s) (BSS/OSS)
Service Delivery Broker
Key Functions
Development Tools
ITILv3 Service Lifecycle Management & Metadata
Tools to Implement TMF SMI
Wizards for SOA Best Practice
Catalog Functions
Runtime Operations
Application Service Provider
Communications Service Provider
Application
Cloud Virtualized Resources
Network Virtualized Resources
Application
Network Logical Resources
Network Virtualized Resources
Delivery Path
CRM
Network Operator Manager
1 High Quality User Experience
Content Delivery with QoS

Figure 18– The SDB in a Services Orientated developer governance role

In Figure 18, there are two cloud stacks depicted on the Communications Service provider side. One depicts a cloud service infrastructure and the other stack is the telecom network itself. The telecom network (voice/data, core transport, backhaul, access) is just another cloud running over a virtualized stack of resources exposing services for consumption by developers in service compositions.

Several important points need to be clarified at this point:

- *Service APIs can be created independent of a Service Broker*: Most service APIs being exposed today are in fact created independently of the governance provided by a Service Delivery Broker. They may be created out of fine grained services by a lower level Service Delivery Platform and exposed as coarse grained services. However, the wide inconsistency today in the application of standards as these services are exposed results in a significant amount of custom integration work when combining these service APIs into more complex offerings.
- *Many Service Delivery Brokers*: Figure 18 is not meant to imply there should be one “all-controlling” master Service Delivery Broker. In reality, there will be many SDBs in operation. An SDB enables one view into a universe of Service APIs. It is likely that most service providers will want to have a Cloud / Service Delivery Broker to support their localized developer ecosystem and to support their version of a service marketplace. Enterprises are beginning to discover they too can use this type of Service Delivery Broker to guide a true service orientation across their enterprise architecture.
- *SDB as a Service*: Some stakeholders may be interested in leveraging a “Service Delivery Broker as a Service” offering. A service provider could gain the benefits of managing their own view into a set of service APIs without incurring the cost and expense of implementing their own Broker. An enterprise can use a cloud hosted SDB as a core part of their services orientation governance.
- *SDB as a mediator between other Service Domains*: A particular service may become exposed from two or more service brokers. The broker a developer or enterprise chooses to use will depend upon the value that particular broker brings to the development, runtime management, and service monetization process versus a different broker offering the similar services. A well-

executed SDB can deliver up to a 50% reduction in service creation and integration costs for an enterprise or service provider / operator. SDB as a Service with global reach could help create consistent service creation and runtime environments very efficiently.

This has two impacts:

- There is a trend towards lightweight APIs that work consistently across multiple cloud and service domains, for instance REST Web Services using JSON.
- How well any given SDB is executed and helps operators achieve significant operational efficiencies and drive service monetization will likely determine which SDB implementations will become most prevalent in the future.

6 MULTI-CLOUD MANAGEMENT IN ACTION

As we said earlier, managing end-end services that comprise component services from multiple partners requires a practical approach to integration of the partners so the effective operate as a single extended enterprise.

Traditionally such problems have been tackled by service providers setting out detailed requirements for forming and maintaining value chains, and developing specifications through a consensus process. In a rapidly changing business environment, the need to experiment with different business models and different types of value chain it is difficult to see how even capturing a stable of requirements for integrating partners can be achieved.

The Multi-cloud Management Accelerator provides a more flexible and heuristic approach to integration across a cloud eco-system and can be characterized as a Bazaar[2] approach to API development and includes the creation of groups of like-minded developers sharing best practice developing APIs and code snippets and a ruthless focus on 'show me how this works' with sample code and examples.

Given the sheer variety of cloud based services that the market is generating, it is impossible to try to develop comprehensive management API's that cater for every type of cloud service, and so the approach taken here is for a simple core set of basic management functions that can apply generically to many types of service and that can be easily extended to provide more service specific functions. It uses a holistic, iterative and implementation focused approach that copes with changing business requirements and can quick exploit and assimilate new integration approaches and technologies. This development led, API centric integration approach is underpinned by the following supporting capabilities:

- Ability for developers can find and share and try out API solutions;
- Support for developer to test and try APIs using a sandbox
- Documentation and presentation of API functionality
- Common and familiar licensing regimes
- Support for developer queries

The Multi-cloud Management Pack is supported by:

- A Developer Pack comprising web pages, sample code implementations, guidance on implementation, to show developers how to apply these interfaces to practical use cases.
- API development support e.g. test and sandbox, on the TM Forum web site and replication on the WEB through API specialist sites, will be launched in the coming weeks.

Materials from SM-API demonstrations by Microsoft and PT-SAPO at Management World Americas 2012 can be found at Using Simple Management API with Windows Azure Media Services (Microsoft) and Using the Simple Management API for IPTV (PT-SAPO).

[2] http://en.wikipedia.org/wiki/The_Cathedral_and_the_Bazaar

PRACTICAL EXAMPLES

The companion Developer Pack gives some examples of practical application of the approach outlined in the guide, taken from proof of concept 'Catalyst[3]' projects demonstrated the Forum's *Management World* around the world.

The scenarios reference commercial products of the participants in the most recent project:

Multi-Cloud Developer Experience Catalyst

Showcased a developer ecosystem where applications can be created with service components from various cloud providers, at the same time with improved developer and application consumer experience. This ecosystem utilizes a service delivery broker with standardized service management mechanism for developers to assemble, test, launch, and manage, using the Cloud Management Interface (SM-API), e2e the Service Quality of the resulting application.

at&t

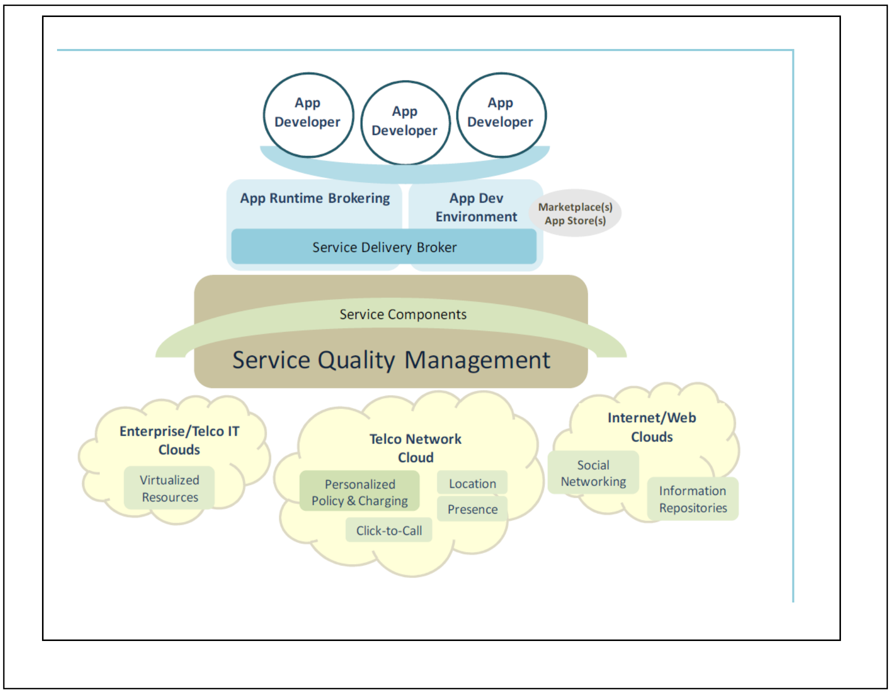
PT sapo

IBM

Microsoft

Telcordia.

Base^n



[3] Catalysts provide a rapid prototyping environment where service providers, suppliers and systems integrators work together to create solutions for critical industry operational and systems challenges by drawing on TM Forum standards and best practice, The benefits of these projects are that they accelerate development and validate standards and best practices and 'show and tell' at the Bi annual TM Forum Management World events. The findings of each Catalyst project are contributed to TM Forum's Collaboration Community as extensions to existing best practices and standards, or as the groundwork to launch new Collaboration projects. The results are often translated into commercially valuable products. A white paper was published at the conclusion of this Catalyst titled: "Best Practices and Standards Recommendations derived from the Multi-Cloud Developer Experience Catalyst",

6 FRAMEWORX OVERVIEW

The Multi-Cloud Management Accelerator is part of the wider suite of Frameworx management best practices. The suite consists of four major components (shown in Figure 19) plus a large number of supporting services to aid understanding and implementation. The four major component frameworks are:

- Business Process Framework (formerly known as eTOM) – a comprehensive standardized process architecture for both business and functional processes
- Information Framework (formerly named the SID) – a common reference model for enterprise information that digital service providers, software providers, and integrators use to describe management information
- Application Framework (formerly known as TAM) – which provides a common language between digital service providers and their suppliers to describe systems and their functions, as well as a common way of grouping them
- Integration Framework – which delivers a service oriented integration approach with standardized interfaces and support tools

Although the four frameworks can be used independently, the real power comes when they are used together. This is very important in achieving high levels of automation of business processes while maximizing agility and supporting changes:

- A clear 'map' of the business processes being automated (the Business Process Framework);
- A common way of describing information (a model) implemented on all systems so that information can be utilized seamlessly between the systems (that is, the information Framework)
- A well understood set of the functions provided by the business systems that handle those transactions (the Application Framework)
- An integration approach that allows for simple and flexible interconnection (through the Integration Framework)

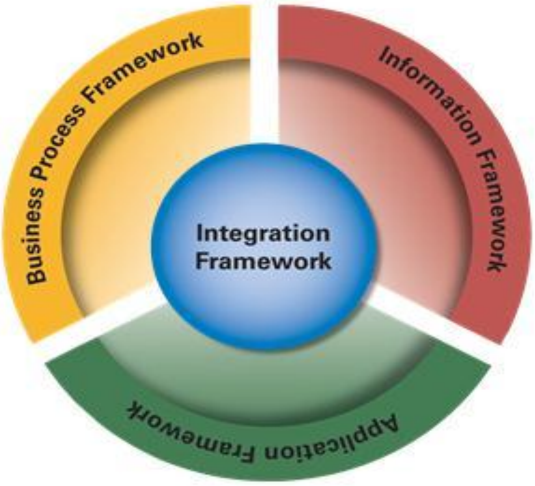


Figure 19– Frameworx

THE BUSINESS PROCESS FRAMEWORK

The Business Process Framework is the heart of Frameworx because without a clear and comprehensive view of the end-end business processes running either inside a service provider or between them in an ecosystem, it's simply not possible to drive towards maximizing business effectiveness – in other words achieving high levels of business agility and a great customer experience with the least possible operating cost. A consistent view across an entire process chain – which increasingly spans multiple companies, is critical to prospering in the digital services market. Unfortunately, many large organizations often only have a high level, limited view of this and cannot drill down to the detail that really matters.

The Business Process Framework is an industry-agreed, comprehensive and multi-layered view of all of the key business processes that a digital service provider needs to run their business. It provides a guide to best practice and as a target process template for use either within or between service providers in the ecosystem. It's supported by a range of off-the-shelf tools and comes in machine readable as well as document-based formats.

In addition, as digital services become ever more software centric, the core service creation and delivery processes need to be well integrated with IT management processes, so thanks to close collaboration between TM Forum and another organization, the itSMF, the Business Process Framework and the major IT process standard called ITIL can be used In close combination. The Business Process Framework simplifies and assures consistent, enterprise-wide ITIL implementation by providing an out-of-the-box business set of processes and flows.

Frameworx is also aligned with the architectural standards of yet another body called The Open Group's systems, whose TOGAF architectural framework is widely adopted. This means it is able to deliver business agility through component re-use, essential in today's fast changing market.

THE INFORMATION FRAMEWORK

Just as it's difficult to develop highly effective processes without a clear 'map' of how they work and interact, it's also impossible to automate business processes and get the right information to the right people or system at the right time without consistent use of data right across the enterprise. Without a common information approach, information is locked up into isolated islands of data making it very difficult to get a comprehensive view of what is happening. This hampers customer self-service, holds up marketing and product management activities, and damages the customer's experience.

It also makes the organization rigid because any change means major re-engineering of information flows. With a common data approach, the organization can be much more agile and able to cope with rapid change; information is available to the right people at the right time and in the right format. In comparison, piecemeal integrations between dissimilar systems usually results in the lowest common denominator level of information.

The Information Framework provides a solution to this – a definition for information that flows throughout the enterprise and between business partners. Supported by off-the-shelf tools and published in machine readable formats, the Information Framework provides a common information model that reduces complexity and allows the definition of standardized integration points. There is a strong link between this Framework and its sister Business Process Framework. It describes the types of information present in a service provider's business and is particularly useful for bridging between players in the ecosystem.

Information is usually transferred between various systems and applications using a technology called an enterprise data bus. Although it tempting to think that a common data bus is all that is needed to share information between systems, this is a bit like saying all you need for the whole world to communicate is a telephone system. What happens when one person is speaking in Mandarin and the other in Swahili? You need some form of translation.

Many data buses do this well but even if the language can be translated, the meaning of what is being said may not work. There's an old saying that even if a lion could speak you would still not understand what he was saying because his view of the world would be so different to yours. Imagine a quantum physicist and a musician trying to have a conversation about their relative expertise where they both use the same words but mean entirely different things by them.

So to share information between applications along an automated process (which while pretty smart are still a lot more stupid than physicists and musicians), you need a telephone system (the communication bus); a translator (protocol mediation) and common ground for understanding of information to be meaningful. This latter function is the key value that the Information Framework provides.

So the Information Framework provides the next important piece of the puzzle to get high levels of automation of business processes through tight but flexible integration between systems.

THE APPLICATION FRAMEWORK

Application Framework provides a standard classification structure for operational applications and it has been widely adopted as the lingua franca of the operations systems world. A common approach and definition means that systems procurement is much more straightforward as both buyers and sellers can actually communicate using the same classification approach and terminology.

The Application Framework provides a way of grouping process functions and associated information into recognizable applications. It provides a common language and identification system between buyer and supplier for all application areas.

THE INTEGRATION FRAMEWORK

The Integration Framework defines how the processes and information behind these systems can be automated by defining standardized SOA-based interfaces called Business Services. These are specifications for pieces of functionality useful in the design and implementation of complex software. Software developers implement Business Services and make those services available through any popular software technology (examples include Corba, Java, XML and Web Services) and increasingly via cloud services.

The Integration Framework aims to include:

- A library of Business Services and interfaces
- Service Oriented Enterprise development guidance and guidelines for the development of additional Business Services
- Software tools that speed up the production of standard interfaces along with reference implementations and conformance testing capability

All of the main Frameworks have direct links to each other and the Integration Framework uses these to produce SOA Business Services which are the mainstay of it. A Business Service is a standard package of reusable capability that carries out some part of a business process using a software application. In effect it 'hides' the underlying complexity of implementation aspects of a system by exposing the information that another system may need to use in a common format. A Business Service may represent a capability either at a high level (for example a business transaction between two corporations), at a low level (the functionality supported by a particular aspect of a software program) or anywhere in between.

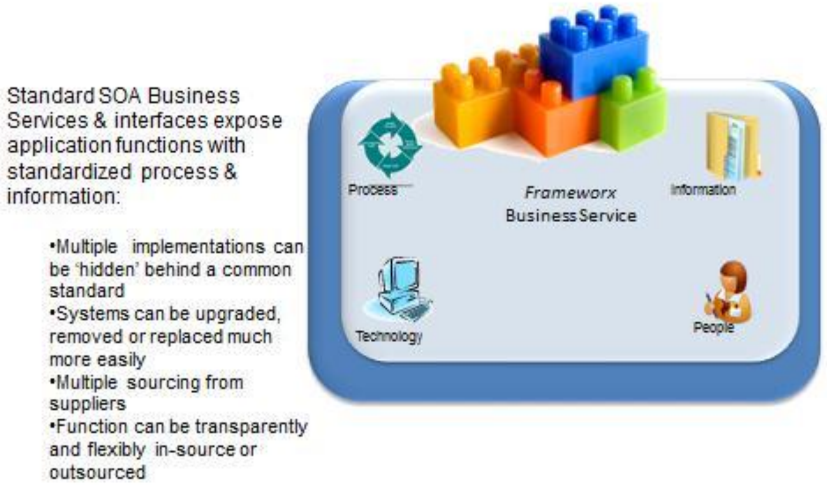

Standard SOA Business Services & interfaces expose application functions with standardized process & information:
•Multiple implementations can be 'hidden' behind a common standard
•Systems can be upgraded, removed or replaced much more easily
•Multiple sourcing from suppliers
•Function can be transparently and flexibly in-source or outsourced
Process
Frameworx Business Service
Information
Technology
People

Figure 20– Frameworx Business Services

The Integration Framework provides a template against which new Business Services can be created. The Business Service specification is a detailed definition of what the Business Service provides and which can be implemented as a real piece of software, created using a standard toolset, programming language, etc.

In the past, interfaces were developed in isolation and tended to incorporate fixed ideas about the information model and business process it was serving. Like many other standards organizations, the Forum has developed a number of these types of interface. They can be very valuable, but have the disadvantage of often being restrictive when it comes to incorporating features and functions outside the scope of the standard. There may be many reasons why these extensions are needed – specific legislation or work practices within the service provider or exposure of special features from a supplier's technology or system.

The Integration Framework aims to overcome this lack of extensibility by linking the other Frameworks; it uses software tooling that can directly generate interface software code. This is known as a model driven approach and such tools can radically reduce the time taken to develop new interfaces and improve their quality as they generate documentation and test kits at the same time as producing the interface specifications.

FRAMEWORX IMPLEMENTATION SUPPORT

As you can tell from the descriptions above, Frameworx is a comprehensive and rich set of standards and guidelines, so inevitably they are complex. This is why the Forum provides a lot of support services to help companies implement them, shown in Figure 21 and followed by a brief description of each of the support services.

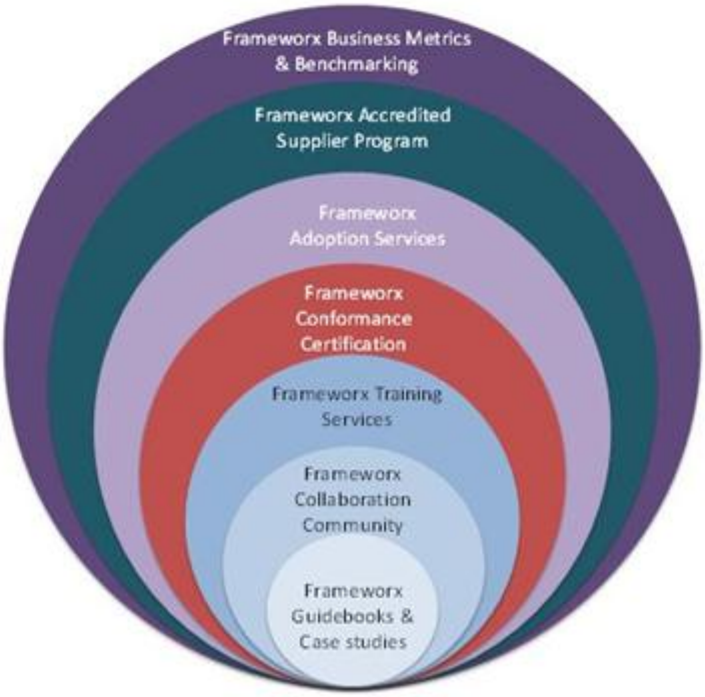

Frameworx Business Metrics & Benchmarking
Frameworx Accredited Supplier Program
Frameworx Adoption Services
Frameworx Conformance Certification
Frameworx Training Services
Frameworx Collaboration Community
Frameworx Guidebooks & Case studies

Figure 21– Frameworx implementation support services

7 REFERENCES

Microsoft System Center. (2011, July). Guide for System Center Monitoring Pack for Windows Azure Applications. Retrieved from http://www.microsoft.com/en-us/download/details.aspx?id=11324

Portugal Telecom SAPO. (2012). Service Delivery Broker. Retrieved from http://sdb.sapo.pt/en/index.html

Thomas Mendel, P. a. (2006). Converged Service Delivery: The Missing Link In Achieving Business Flexibility. Cambridge, MA: Forester Research Inc.

TM Forum. (2008, June). TR141 - SDF - Industry Groups Positioning Document. Retrieved from Software Enabled Services Management: http://www.tmforum.org/TechnicalReports/TR141ServiceDelivery/35371/article.html

TM Forum. (2009, April). TMF061 - Service Delivery Framework Reference Architecture. Retrieved from TM Forum Software Enables Services Management: http://www.tmforum.org/TechnicalSpecifications/TMF061ServiceDelivery/39341/article.html

TM Forum. (2009, Jul). TR139 - Service Delivery Framework Overview. Retrieved from TM Forum Software Enabled Services Management: http://www.tmforum.org/TechnicalReports/TR139ServiceDelivery/34303/article.html

TM Forum. (2010). TMF618 - Software Enabled Services Lifecycle Management Metadata Interface Information Agreement. Retrieved from TM Forum Software Enabled Services Management: http://www.tmforum.org/InformationAgreements/TMF618SoftwareEnabled/46847/article.html

TM Forum. (2012, February). TMF617 - Software Enabled Services Management Interface Information Agreement. Retrieved from Software Enabled Services Management: http://www.tmforum.org/DocumentCenter/11777/home.html#TRCDocuments/artf2533

TM Forum. (2012, February). TR168 - Software Enabled Services (SES) Management Solution and Frameworx Relationships. Retrieved from Software Enabled Services Management: http://www.tmforum.org/TechnicalReports/TR168SoftwareEnabled/46857/article.html

Troup, E., & Huang, J. (2011, Dec). Best Practices and Standards Recommendations derived from the Multi-Cloud Developer Experience Catalyst. Retrieved from http://collab.tmforum.org/sf/docman/do/downloadDocument/projects.service_delivery_framework/docman.root.phase_vi.contributions_discussion_materia/doc20976

Willetts, K. (2012). Unzipping the Digital World. Morristown, NJ: TM Forum.

8 ADMINISTRATIVE APPENDIX

This Appendix provides additional background material about this document.

VERSION HISTORY

Version	Date Modified	Modified by:	Description of changes
1	14 July 2012	Keith Willetts	First draft
2	14 Aug 2012	Keith Willetts	Restructure and additional material
3	5th Sep 2012	Dave Milham	Restructure and re-organised fixed template issues
4	6th Sept 2012	Marco G	Added SES materials
5	18th Sept 2012	Dave Milham	Review add missing materials and improve flow (
6	22 Oct 2012	Keith Willetts	Final edits and naming
7	20 Nov 2012	Joann O'Brien	Final edits based on member feedback
8.1	21 Nov 2012	Joann O'Brien	Editorial Changes
8.2	21 Nov 2012	TM Forum Staff	Format edits, notice statement update, additional corrections.
8.3	26 Nov 2012	Joann O'Brien	Link validation and correction across the document
8.4	26 Nov 2012	TM Forum Staff	Removed R1.0 line from cover page per current process
8.5	22 Feb 2013	John Wilmes	Added metrics for CEM, fixed typos/comments, reduced repetition
8.6	25 Feb 2013	TM Forum Staff	Cosmetic updates.
8.7	27 Feb 2013	TM Forum Staff	Cosmetic updates prior to web posting

RELEASE HISTORY

Release Number	Date Modified	Modified by:	Description of changes
1.0	20/Nov/12	Keith Willetts	First issue of document
1.5	22/Feb/13	John Wilmes	Second issue of document

ACKNOWLEDGMENTS

This document was prepared by the following people:

- Authors: Eric Troup, *Microsoft*; Keith Willetts, David Milham, Marco Giaccaglini, *TM Forum*
- Editor: Keith Willetts, Joann O'Brien, John Wilmes *TM Forum*
- Reviewers and contributors:
  - Jenny Huang, AT&T

- Steve Fratini, Ericsson
- Johan Vandenberghe, Alcatel-Lucent