

# Service Delivery Framework – An Architectural Strategy for Next-Generation Service Delivery in Business Network

Alistair Barros, Uwe Kylau  
 SAP Research (Brisbane)  
 SAP AG  
 Brisbane, Australia  
 e-mail: {alistair.barros, uwe.kylau}@sap.com

**Abstract**—The next-generation of service-oriented architecture (SOA) needs to scale for flexible service consumption, beyond organizational and application boundaries, into communities, ecosystems and business networks. In wider and, ultimately, global settings, new capabilities are needed so that business partners can efficiently and reliably enable, adapt and expose services. Those services can then be discovered, ordered, consumed, metered and paid for, through new applications and opportunities, driven by third-parties in the global “village”. This trend is already underway, in different ways, through different early adopter market segments. This paper proposes an architectural strategy for the provisioning and delivery of services in communities, ecosystems and business networks – a Service Delivery Framework (SDF). The SDF is intended to support multiple industries and deployments where a SOA platform is needed for collaborating partners and diverse consumers. Specifically, it is envisaged that the SDF allows providers to publish their services into network directories so that they can be repurposed, traded and consumed, and leveraging network utilities like B2B gateways and cloud hosting. To support these different facets of service delivery, the SDF extends the conventional service provider, service broker and service consumer of the Web Services Architecture to include service gateway, service hoster, service aggregator and service channel maker.

*SOA, service delivery platform, business networks, software-as-a-service, BPO*

## I. INTRODUCTION

The growth in on-demand applications has surged to the point where Gartner [1] and IDC [2] forecast it to exceed growth of on-premise applications between 2008 and 2012. Through software-as-service (SaaS) and cloud-based deployments accessed over the Web, on-demand offerings now extend to desktop (e.g. Microsoft Office Live<sup>1</sup>), enterprise (e.g. SAP CRM<sup>2</sup>, Salesforce<sup>3</sup>), supply chain management/logistics (e.g. LeanLogistics<sup>4</sup>), utilities (e.g. EnergyCAP<sup>5</sup>) and many other segments. With a critical mass of offerings proliferating from diverse sources, *stage two* of

the on-demand revolution is underway. Inspired by examples like Salesforce AppExchange<sup>6</sup> or the Microsoft partnered Intuit Marketplace<sup>7</sup>, a new wave of *Appstore* initiatives has emerged as smaller SOA style platforms for SaaS [3], to procure service-based software from central points and communities. They use e-commerce marketplace concepts for: consolidated exposure and discovery of services; partner ecosystems; rapid development tools to diversify the supply of applications and drive up value-added composites; and service delivery featuring hosting, metering and pay-per-use billing.

With these developments, attention is turning to how large enterprises can similarly leverage SaaS, marketplaces and hubs. Business applications in most enterprises contain highly valuable data and functionality that could be unlocked, commoditized and turned into new and unforeseen service opportunities. In turn, these applications, themselves, need to integrate with external services in order to enhance the decision making quality and other aspects of business processes, since line-of-business users often have to resort to phone calls, Web site look-ups, and spreadsheets in their day-2-day business. Having access to a reserve of external services can reduce overheads and increase efficiencies in line-of-business activities, driving up a new class of SaaS marketplace consumers - from the corporate sector.

In broader strategy terms, the exploitation of services for large enterprises falls under business network transformation (BNT) [4], i.e. the transformation of business operations through global partnerships, under the competitive pressure of deregulation and improved communications technologies. At the heart of BNT are services that need to be out-sourced/in-sourced from the global “village”, enabling companies to focus on differentiation and leaving it to partners for the rest. Thus, the patterns of creating and accessing on-demand services and making use of marketplaces and hubs for business networks resonates with SaaS developments. There are, however, striking differences.

Services in large enterprises relate to a myriad of applications in the form of multi-vendor suite solutions, home-grown applications and third-party, business process outsourcing (BPO) solutions [5] – operating within complex IT landscapes and on-premise hosts. They are typically

<sup>1</sup> Microsoft Office Live: <http://www.officelive.com/>

<sup>2</sup> SAP CRM: <http://www.sap.com/solutions/business-suite/crm/>

<sup>3</sup> Salesforce: <http://www.salesforce.com/>

<sup>4</sup> LeanLogistics: <http://www.leanlogistics.com/>

<sup>5</sup> EnergyCAP: <http://www.energycap.com/>

<sup>6</sup> Salesforce AppExchange: <http://sites.force.com/appexchange/home>

<sup>7</sup> Intuit Marketplace: <http://marketplace.intuit.com/>

transactional, long-running and have complex data dependencies across different applications. This makes it difficult and sometimes impossible to create decoupled services that are redeployed onto external Clouds. Unlike today's SaaS offerings, opening up transactional services for wider, commoditized use through marketplaces/hubs requires that services continue to execute from their secure, hosted environments. Nonetheless, leveraging transactional applications through marketplaces/hubs is gaining traction.

As an important trend indicator, companies in the corporate sector use BPO solutions for coverage of coarse-grained business processes which are no longer sourced inside their organizations. Examples include payroll processing<sup>8</sup>, compliance<sup>9</sup> and logistics<sup>10</sup>. In the area of logistics a variety of BPO forms have emerged including B2B gateways, e.g. Crossgate and Sterling Commerce, which manage adaptation of services for interoperability with trading partners. This reduces the integration efforts and costs of companies whose applications have typically been built in isolation without foresights of flexible integration with other applications in business networks. In current BPO solutions, companies either have direct or indirect (offline) interactions, although on-demand, consumption-based pricing is used in BPO business models and a next-step to fully automated interactions through APIs (like a type of business platform-as-a-service) seems like natural one for BPO, as it has been for SaaS.

These various developments present major uncertainties for companies in their strategic planning for IT and the provisioning of their applications in particular. The question is how can applications be provisioned so that they can keep abreast of change triggers of business networks requiring improved service orchestration among network partners and more seamless access to cost-efficient on-demand applications and hubs through IT portfolios? Correspondingly, software vendors are faced with the challenge of creating a comprehensive platform strategy for not only allowing their applications across different product segments and technology stacks to be accessed - as services - but also to support service access and delivery needs of partners in a business network.

This paper describes a comprehensive platform strategy for supporting global service provisioning and delivery for business networks, namely a Service Delivery Framework. Unlike current service platforms used to enable software marketplaces, the SDF aims at publishing and brokering more complex business services, and allowing these to be repurposed, traded and consumed in network settings. In short, the SDF support diversified partnerships for provisioning of services to best-of-breed utility services and innovation resources on the network. The SDF sets the stage for the next service-oriented revolution, referred variously as

service ecosystems, future business value networks, and other forms of hubs and communities [6], underpinned by an Internet-scale infrastructure [7], to create a level playing field for supply and demand of services on the Web.

The structure of the paper is as follows. Section II provides a background, through prominent examples, of business networks and how services are central to their transformation. With this insight, Section III motivates the need for an integration platform for service provisioning and delivery in business networks, and accordingly proposes the SDF, focused on its architectural strategy without elaboration of implementation details. Section IV discusses 2 examples of how the SDF supports key transformations in business networks - creation of industry-specific service marketplaces and accelerating partner connectivity. Related work is presented in Section V and Section VI concludes the paper.

## II. EXAMPLES OF SERVICE-FOCUSED BUSINESS NETWORKS

### A. eGovernment Citizen and Constituent Services

Governments worldwide have, over several years, and at multiple levels of jurisdiction, invested in strategies aimed at improving service delivery between government agencies, between government and business, and government and citizens. eGovernment projects have been central to this goal, embarking on operational frameworks [8] that include cross-agency organizational structures, standard business processes and data, and technical integration, to allow otherwise stove-piped services running in heterogeneous back-ends to be integrated and consumed outside the channels and delivery operations of individual agencies. In addressing the challenges of enabling services to be accessed through standard business and technical interoperability mechanisms, eGovernment frameworks open up the prospect for the public sector to make services available through multiple channels, across local, state and federal levels, and for services to be delivered through private sector partnerships.

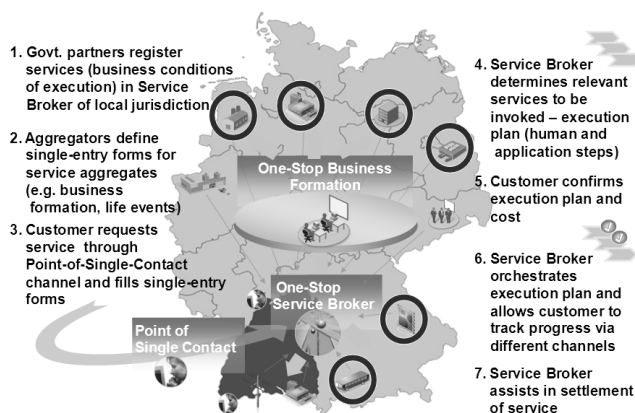


Figure 1. eGovernment one-stop citizens and constituent services access

Figure 1 illustrates the setting of eGovernment initiatives concerning one-stop service delivery for citizens and constituents.

<sup>8</sup> Accenture payroll BPO: [http://www.accenture.com/Global/Consulting/Talent\\_and\\_Organization/Human\\_Resources\\_Mgmt/Services/Payroll.htm](http://www.accenture.com/Global/Consulting/Talent_and_Organization/Human_Resources_Mgmt/Services/Payroll.htm)

<sup>9</sup> Technidata compliance BPO: <http://www.technidata.com/>

<sup>10</sup> GT Nexus: <http://www.gtnexus.com/>

For service delivery to be opened up in this way, agencies need to be able to publish their services into central directories in governments, operated by lead agencies acting as brokers (1. in Figure 1). Services available through service brokers would cut across different agencies and not be confined to services available to a single agency only. For citizens, this would mean that services as diverse as passports and visas, change of address and life events, permits/licenses such building permits, parking permits and business licenses, would be conveniently discovered and interacted with in a one-stop fashion. For this, services would need to be sufficiently described and their execution components like business processes, operations, global data types and UIs, would have to be sufficiently externalized so that brokers could allow consumers to interact with them (requests) and mediate the required operations with corresponding backend applications of services running in hosted environments controlled/outsourced by service providers (for responses).

Services such as life events, business formation and property conveyance involve orchestration of services from several agencies. To facilitate integration of cross-agency services, agencies need to agree on data types and formats, processes and UI presentation, for their services. The effort to drive cross-agency services requires that agencies form coalitions or work through lead agencies, as aggregators, who manage the standardization of related services, service repurposing and aggregation - publishing new value-added services through central directories and brokers (2.).

Citizens and constituent services are exposed through central directories and brokers and their exposed interfaces permit central access with backend applications. In this way citizens and constituents discover, order, interact with, track and pay for services through standard, cross-government channels e.g. service desks, web sites and operator call centers (3.-7.).

In practice, eGovernment projects have mostly produced central portals containing information and documents concerning services and how they can be accessed [9]. Actual integration with agencies has mostly proven allusive due to political, operational and technical obstacles. Overcoming this with high-level political mandates, like for example the European Union Services Directive [10], has yet to prove actual effectiveness. Nevertheless, where governments have invested in integration platforms, even if they are *home-grown* and limited in scalability, service access has significantly improved and agencies have demonstrated a willingness to open themselves up as partners in business networks. Take the example of Directgov UK's one-stop citizen services initiative and one of the largest of its kind in the world, which receives more than five million visits a month. Services are linked to public sector agency applications through the DirectGov gateway, which provides secure access and some mediation where required. Given its success, the British Government decided to reduce some 550 Web sites, dedicating DirectGov as one of two primary channels to public sector services.

Pro-active governments are also recognizing that the private sector can add value to consolidation and innovation

in service delivery. Singapore TradeXchange<sup>11</sup> aims at streamlining information exchange and activities between commercial entities and regulation agencies for efficient flow of goods within, into, and out of, Singapore. TradeXchange went live in October 2007 and has some 10,000 subscribers from the trading community.

### B. Global Logistics and Supply Chain Management

Supply chain management is undergoing systemic restructures under globalization and deregulation, with companies seeking partners from around the globe for offshore manufacturing and market penetration in multiple regions. For manufacturers, suppliers, retailers and buyers alike, logistics, a core function of supply chain management concerning the flow of goods, resources and information, has been outsourced progressively to logistic service providers (LSPs), allowing companies to reduce costs, improve delivery and focus on core competencies. As supply chain chains have expanded to global settings, the range of services provided by LSPs has inevitably widened to simplify the outsourcing arrangements for shippers (companies needing goods transported). Figure 2 illustrates this variety.

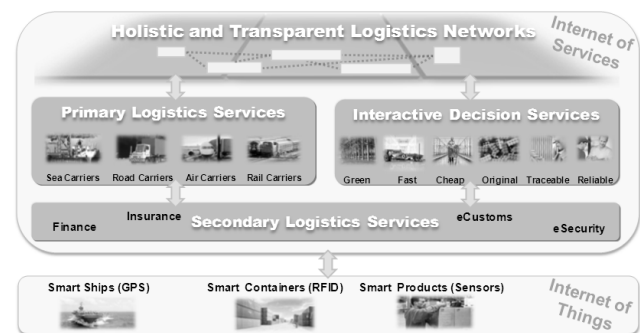


Figure 2. Vision of a future transport and logistics network

A major part of logistics, accounting for 50% of its costs, is transportation management, traditionally delivered through asset-based LSPs in freight forwarders/carriers (air, ocean, rail, less-than-truckload, full-truckload, parcel, and private fleet) and storage nodes/warehouses. Third-party logistics providers (3PLs) have emerged to provide more comprehensive coverage - considering that in-/outbound orders need to be recorded, freight forwarders and carriers need to be assigned for the different segments of the transportation route, shipments need to be cross-docked and consolidated at intermediate nodes, the necessary documentation needs to be put in place for import/export clearance through customs and quarantine, and goods need to be delivered to consignment nodes and distribution centers of buyers and retailers. Along the way, orders may be cancelled and either need to be returned or be procured through new orders in nearby destinations (a process known as reverse logistics). As a result of delivery, fulfillment of an order needs to be assessed with respect to quantity, timeliness and damage for quick turn-around on payment and re-imburement. Banks

<sup>11</sup> <https://www.tradexchange.gov.sg/tradexchange/default.portal>



circumvent un-trusted payments for lengthy shipments through letters of credit.

Trends towards outsourcing of end-to-end logistics are significant with a recent survey [11] showing that on average 47% (in North America) to 66% (in Europe) of shippers report that their total logistics expenditures go to outsourcing, and all geographies anticipate increases in these percentages in the next five years. Some asset-based LSPs have evolved into 3PLs. However, 3PLs have increasingly become orchestrators of external asset-based services for market coverage across different regions, with IT services (e.g. order management, carrier selection and bookings, RFID, event management and track-and-trace, letter of credit) as their differentiators.

Thus, global business networks are becoming crucial as shippers seek end-to-end logistics management from 3PLs who, in turn, leverage diversified partners and services for comprehensive provisioning. Business networks allow scalability of service provisioning in multiple regions for international customers and shipments, and the IT services of 3PLs play a major part in coordinating logistics business networks. With only 20% of companies involved in using IT solutions, a growing number of on-demand or Web-based solutions have emerged in the form of software-as-a-service (e.g. LeanLogistics), B2B gateways (e.g. Crossgate, GXS<sup>12</sup>) and BPO (e.g. GTNexus, Intra<sup>13</sup>, e2open<sup>14</sup>). They support different segments of transport and logistics and have created networks around them. They, too, need to be harnessed in SCM software portfolios, along with on-premise IT solutions, for SCM customers.

A further networking trend is shippers looking beyond transportation services for outsourced logistics to sharing responsibilities and therefore risks in supply chain planning, product merchandising and demand planning. For example, retailers have traditionally controlled stock replenishments however suppliers are increasingly engaged to carry out that duty by being involved in forecasting and having access to retailers' point-of-sale data, stock inventory and forecasts by line-item. In turn, retailers are afforded governance in product merchandising, otherwise an activity of suppliers, to safeguard and expand their markets. Fourth-party logistics providers (4PLs) are emerging to provide services that can cover these concerns, and for this, partners in a business network are required to further open up their systems (e.g. inventories and point-of-sale data) for outside monitoring and influence.

### III. TOWARDS A SERVICE DELIVERY FRAMEWORK

To support the way in which services are exposed and delivered in business networks, as discussed in the previous section, an *integration platform* that can be utilized by service providers, intermediaries and consumers on the level of the network, is necessary.

#### A. Requirements for Extension of Service Platforms

The following needs for extension are proposed for service provisioning and delivery in business networks and thus for integration platforms supporting them.

##### 1) Diversity of Services

Services in business networks are more than Web services or other forms of application interfaces. Some services concern human activity only, e.g. manual audits, consultancy and therapeutic services. At the other extreme, purely technical services providing on-demand platform and infrastructure functionality, e.g. Amazon EC2, also exist. In between are business applications whose services involve a mix of human and automated parts. In order to support different service industries, an integration platform should support the cataloguing, access and consumption of a wide range of services: purely human services would simply be catalogued, business applications could be accessed through central mediation, and purely technical services would be directly accessed once discovered.

##### 2) Diversity of provisioning partners

Services are provisioned by providers for consumers. In business networks, services can be opened up to partners who can extend their capabilities through different forms of third-party provisioning. For instance, traditional aggregators as well as SaaS and BPO players are driving new business out of existing services combining them for value-added functionality. Intermediaries, providing platform and infrastructure resources over the Web, extend provisioning, thus lowering the total-cost-of-ownership for providers to run services. New channels and marketplaces are also emerging, as intermediaries, to expose services to new audiences and market segments. Hence, an integration platform for services should foster partner diversification for service provisioning, allowing a "network effect" of services, once they have been opened into wider settings.

##### 3) Diversity of consumption modes

In-house applications are accessed by different business users through different environments inside an organization, and through external interactions with business partners. On the other hand, services exposed in today's marketplaces and SaaS/BPO hubs are accessed through the e-commerce marketplace consumption model, like that available for eBay or Amazon users who search, order, access, pay for, and track their orders. While this is an important development for expanding the content that can be procured and traded for Web consumers, the same consumption experience should be available for business users. In other words, business users should have the same flexibility to discover and access external services to enhance decision points, reduce media breaks etc., in their line-of-business activities. Additionally, business users shouldn't have to assume two different modes of access - through portals or work centers for internal applications and the e-commerce marketplace access for external services. Rather, they should get the best of both worlds. Services should be available in business processes within organizations, regardless of whether they are internal applications or external services. In general, an integration

<sup>12</sup> GXS: <http://www.gxs.com/>

<sup>13</sup> Intra: <http://www.intra.com/home/home.aspx>

<sup>14</sup> e2Open: <http://www.e2open.com/>

platform should support seamless consumption of services on the Web, on devices, and in business processes.

4) *Diversity of service delivery functionality*

In addition to the middleware required for run-time service access, an integration platform also should provide access to business applications supporting service delivery. This can be seen in present-day intermediaries that require, for example, customer-relationship management applications to hold records of their consumers, consumption patterns and rewards, and payment engines for billing, invoicing,

imbursement and disbursement of services fees. Master-data management for data consistency and synchronization used needed for delivery of services from diverse sources would also be beneficial as would helpdesk management, (service) contract management, and even warehouse management/logistics as part of the physical fulfillment of services (e.g. delivery of goods). These are just some of the many functions that come into view when considering the commercial realities of service delivery.

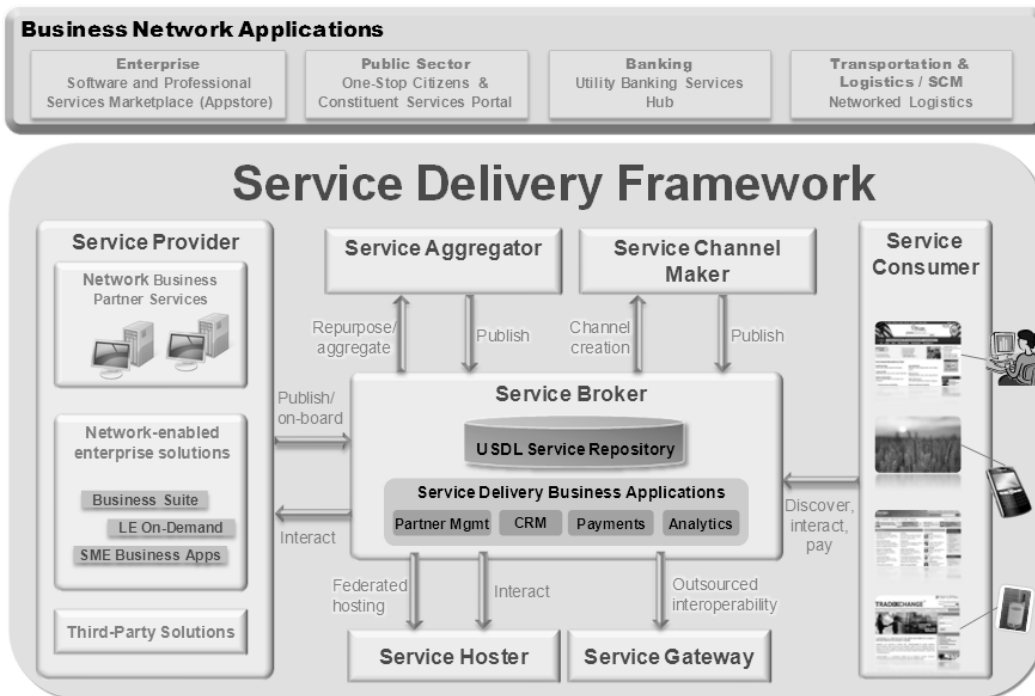


Figure 3. High-level architecture of the Service Delivery Framework

B. *SDF High-Level Architecture*

In order to support service provisioning and delivery in business networks of different sorts, and applicability in different industries, a Service Delivery Framework is proposed, as depicted in Figure 3.

Unlike integration platforms for services such as service delivery platforms [12], which centralize service delivery around a dominant partner (like a telecommunications provider), the SDF aims at supporting a *diversity of provisioning partners*, as seen through different architectural roles like Service Provider, Service Broker, and Service Aggregator. Different provisioning partners can use particular platform functionality dedicated for their service provisioning niches. This functionality offered through the SDF roles combines technical capabilities (e.g. service composition tools, service repository, adaptation and run-time environment, network convergence) with business functionality (e.g. CRM, payments) and aims at supporting *diversity of service delivery*. It is drawn from a foundation layer of

components accessed through a SOA enabled Business Process Platform. As shown in the top part of Figure 3, the architectural strategy for SDF recognizes that specific applications of SDF are needed for business networks operating in different industries, e.g. eGovernment platforms citizens and constituent services. These applications provide UIs and logic that is specific for particular domains, whilst coded against the generic SDF layer. They could be deployed individually as on-premise installations or could be run on-demand for several customers. Since a variety of domains are targeted, the requirements of *diversity of services* and *diversity of consumption modes* need to be supported by SDF.

1) *Service Provider*

The architectural role Service Provider supports partners that govern services and hold operational responsibility through their business operations, business processes, and organizational units, as well as systems and other implementation artifacts. The Service Provider provides the dedicated tooling to engineer, expose and interface services, whether from “scratch” or from existing

applications. In terms of the latter, it supports companies service-enabling their applications by visualizing functional dependencies of applications, analyzing them for coupling and cohesion, and assessing the costs and feasibility of a service interface layer, e.g. SOAP-style Web Services or REST-style interfaces.

The Service Provider role allows services to be exposed into a business network without compromising the mechanisms providers put in place to deliver services. For example, a provider should be able to publish a service to a third-party broker while still requiring that it runs through its current hosting environment and be interacted with through a secure gateway. Some providers may free up constraints of access by allowing services to be re-hosted onto a third-party cloud or be downloaded and run in a consumer's environment. Given that the service is accessed through a third-party, the provider needs to ensure that it is comprehensively described, especially for the benefit of consumers, for example: service ownership, functionality, dependencies on other services, pricing, as well as consumer rights, penalties and obligations in terms of service performance, exceptions, information disclosure and other legal aspects. For services situated in wider business contexts, domain-specific ontologies, vertical industry standards and other service-related information (like policies, legislation and best practices) are also useful to enhance discovery and comprehension of services.

As depicted in Figure 3, the Service Provider interacts with the Service Broker to publish/on-board service details (descriptions, execution artifacts, etc.) for service access through the SDF's central point - the Service Broker. In addition, the Service Provider is embedded into provider environments so that services they host can be integrated with the Service Broker for required inbound and outbound run-time interactions. It allows deployment of standard provider interfaces in provider environments so that, on the one hand, interactions from the Service Broker are guarded against security risks and convoluted logic of hosted service applications, and on the other hand, service providers have well-defined and allowable operations exposed to them for interacting with the Service Broker.

## 2) Service Broker

The architectural role Service Broker supports brokerage agencies that specialize in exposing services from diverse providers into new markets, matching consumer requests with capabilities of services advertised through it. These agencies manage the "front-desk" delivery of services to consumers, without assuming "back-office" responsibility. Although services are accessed through a broker, the execution of their core parts resides elsewhere in a hosted environment. The benefits for service providers lie in reduced costs from outsourced delivery as well as new markets created by brokers. The Web has opened up the prospects of carrying out brokerage practice through online partnerships, reducing the threshold for establishing trust and performance. Moreover, brokers can use Web "tricks of trade" to enhance consumer pull through cost-offsets of advertisers,

market analysts and other entities that benefit from making their businesses visible through other services.

As shown in Figure 3, the Service Broker is central to the SDF and is used to expose services from providers, to be on-boarded and delivered through the broker's service delivery functionality (e.g. run-time service access environment, billing/invoicing, metering). Services consumed through end users or applications, and services offered in a business network through intermediaries like cloud providers and B2B gateways, can be published, alike, in the Service Broker. Third-parties can extend and repurpose services through the Service Broker, e.g. a service aggregator using the directory facilities of the Service Broker to discover and make use of services through design-time tooling (see below for further details). Ultimately service delivery is managed through the Service Broker when services are accessed at run-time by consumers (end users, applications, business processes). In short, the Broker provides a directory and delivery resource for service access.

To support different needs, the Service Broker allows providers to specify different delivery models that constrain the way a service is accessed at run-time. Based on the delivery needs, the Service Broker generates the necessary delivery logic, alleviating providers with the burden of developing and maintaining this. The different delivery models are as follows:

- Ordering professional/human services without any run-time execution (*offline delivery*);
- Downloading services to run in a client environment (*download delivery*);
- Direct access to service endpoints in their backend environments (*reference delivery*);
- Managed access for individual service steps through the intermediary (*mediated delivery*).

The Service Broker is also equipped with CRM, personalization, payments, authentication/authorization, metering/analytics among other components, for support of delivery on a business level.

## 3) Service Hoster

The architectural role Service Hoster allows the various infrastructure services in cloud environments to be leveraged as part of provisioning an application in a business network. For example, there are certain types of service in ERP suites that could be re-hosted from their on-premise environments to cloud-based, on-demand environments to attract new users at much lower cost, without the overheads of requiring access to large application back-ends. Order2Cash, for instance, is required for sales support in many domains from supply chains to financial industries. Indeed, a business network could be used to expand sales lines by allowing SMEs to access a low-cost, cloud-based order-to-cash service integrated with on-premise large-scale ERP systems.

A service can be automatically deployed onto a specific cloud using the Service Hoster's interface. By providing a generic interface for infrastructure services, the Service Hoster opens up the possibility to access cloud

providers, like Amazon EC2 or Rackspace, through a uniform interface. This is important for business networks because cloud services are highly commoditized with slim margins and are subject to business volatility. The Service Hoster offers business networks partners the option to shift cloud providers efficiently, avoiding cloud “lock-in”. As indicated in Figure 3, cloud services exposed to a business network would need to “plug-in” to the Service Hoster and be advertised through the Service Broker. When a service needs to be hosted, the Service Broker can match its hosting needs (e.g. platform services, operating systems) with cloud services that have the required virtual machine templates. As alluded to in Figure 3, the Service Broker performs the matching and lists candidate cloud services for a user (a provider requiring hosting as a part of exposing a service or a consumer negotiating hosting options/costs when a service is ordered). To deploy a service into the selected cloud service of choice, the Service Hoster’s virtualization management interface allows for interaction with specific cloud services. Once hosted, the cloud service may be monitored for performance and reliability through the Service Hoster interface. Monitoring information can be streamed through this interface to the Service Provider.

#### 4) *Service Gateway*

The architectural role Service Gateway supports agencies in selecting a choice of solutions that provide economies-of-scale B2B interoperability – as a service. Different B2B gateways and integrators such as Crossgate, Sterling Commerce and EasyLink have emerged for providing a rich variety of vertical B2B industry standards (e.g. PIDX, VICS) for which they provide design-time business message mapping, as well as run-time message store-forward and message translation. This is beneficial when services need to be exposed on a heterogeneous business network, as it can be used by providers to create different service versions that have interfaces enabling interactions with the different message standards of the partners that are allowed to interact with the service.

Distinct as well as common message standards are supported by different B2B gateway providers. As indicated in Figure 3, the Service Gateway allows a number of them to be registered with it so that their services can be accessed through a generic interface. The B2B gateway services are advertised through the Service Broker, allowing providers to discover candidates for interface adaptation to particular message standards. Key differentiators are pricing models, B2B communities engaging in their hubs, and qualities of service. Some gateways are evolving into suite solutions, providing common master-data and reference business processes in particular industries, e.g. e2open in high-tech manufacturing and GXS for transportation management. They offer greater value since interoperability occurs through a reference business process context. For design-time mapping, the Service Gateway provides model-based tooling for adapting services, regardless of which B2B gateway provider is being used. The tooling supports automated schema matching, user-specified service-to-

service associations on message type and data type levels and automatic adapter generation.

For run-time adaptation, B2B gateways address robustness of interactions through different mechanisms, e.g. by supporting integrity checking of service invocations (e.g. whether an action is allowed on a service given its current state), message correlation, fan-out/fan-in invocations (e.g. an external invocation translate to a number of services calls in different orders) and contingency handling for when service invocations fail. Like the interactions between a Service Broker and Service Hoster, abstract interfaces are offered for engaging, interoperating and monitoring these sorts of functions across the B2B gateway services registered.

#### 5) *Service Aggregator*

The architectural role Service Aggregator supports domain specialists and third-parties in aggregating services for new and unforeseen opportunities and needs. Services may be integrated into corporate solutions, creating greater value for users not otherwise available in applications, or they could be aggregated as value-added, reusable services made available for wider use by business network users. In either case, the Service Aggregator provides the dedicated tooling for aggregating services at different levels - UI, service operation, business process or business object levels. Best-of-breed service aggregation tooling and techniques are expected to be exploited through the Service Aggregator.

Aggregators are similar to service providers; however, there is an important difference. Aggregators do not own all services that they aggregate. Rather, they obtain custodial rights from providers to repurpose services, subject to usage context, cost, trust and other factors. Despite new aggregated services being created, constituent services execute within their existing environments and arrangements: they continue to be accessed through the Service Broker based on a delivery model (as introduced earlier). The SDF includes service-level agreement support through its different roles so that service providers and aggregators can understand the constraints and risks when provisioning services in different situations, including aggregation of services.

The Service Aggregator provides pure design-time tooling functionality. As shown in Figure 3, it interacts with the Service Broker for discovery of services to be aggregated and for publishing aggregations of services for network partners to use. It also interacts with Service Consumer role when services are consumed into existing applications (as discussed below).

#### 6) *Service Channel Maker*

The architectural role Service Channel Maker supports agencies in creating outlets through which services are consumed. Channels, in a broad sense, are resources, such as Web sites/portals, mobile channels and work centers, through which services are accessed by end users. Examples are Internet Banking, Mobile Banking and Retail Banking, respectively. The mode of access is governed by technical channels like the Web, mobile, and voice response.

The notion of *channelling* has obvious resonance with service brokerage. Virtually all the prominent examples of Web-based brokers like iTunes, eBay and Amazon expose services directly for consumption and can also be seen as channels. In SDF terms they would be *single channel brokers*. SDF's separate designations of the service channel and service broker addresses a further level of flexibility for market penetration of services: service channels are purely points of service consumption while brokers are points of accessing services situated between channels and hosting environments where the core parts of service reside. This separation allows different service UIs and channels to be created, outside the capacity of those who broker services. The SDF, in other words, allows for consolidation of service discovery and access through the Service Broker and independent consumption points through Service Channel Maker. This is especially useful for mainstream verticals like the public sector dedicating whole-of-government resources for service discovery and access, but requiring a variety of channels for its different and emerging audiences.

The creation of a channel involves selection of to-be-consumed services, constraining which service operations are to be used, what business constraints apply (e.g. availability constraints), and how the output of an operation should be displayed (forms template). Taking Web page development as an example, service operations together with text blocks, widgets, images, tables etc., would be selected into different parts of the page. Different service forms or outputs presented on the same page may require data field correlations (like widget mash-ups). Considering this among other features, the Service Channel Maker clearly has similarities with Web authoring tools – indeed, Web authoring applied specifically for services (e.g. requiring that service output be equivalently represented as text) should be supported through the Service Channel Maker. Multi-modal interactions are also envisaged, featuring integrated conventional, spatial and voice interactions.

As shown in Figure 3, the Service Channel Maker interacts with the Service Broker for service discovery during the process of creating/updating specifications of channels, as well as for storing these specifications and channeled service constraints back in the Broker. Channels can then be discovered through the Service Broker and allocated through the Service Consumer for usage through designated users/groups in the business network.

#### 7) Service Consumer

The architectural role Service Consumer completes the service supply chain, effectively fostered by the SDF. Through the Service Consumer, parties can manage the “last mile” integration where services are consumed through different environments. This involves the allocation of resources to consumer environments and the interfacing of consumer environments so that services can be accessed at run-time in a secure and reliable way.

As discussed above, the resources that consume services are either explicit channels, or applications that have services integrated (“hard-wired” into code) or

accessible (through a discover/access interface). The Service Channel Maker and Service Aggregator are used for supporting the processes of design-time integration of channels and applications, respectively. Since the allocation of applications in organizations is a specialized consideration controlled through administration systems, the Service Consumer is used mostly for allocating channels in consumer environments (inside organizations or wider availability on the Web).

As depicted in Figure 3, the Service Consumer supports consumer environments with accessing the services they require from the Service Broker through inbound and outbound run-time interactions, and in the context of different delivery models. Interfaces are required on remote consumer environments so that applications running in these environments have well-defined operations for the Service Broker to invoke, and, what important for the consumer, these invocations are free from security risks and application-specific logic.

#### IV. HOW THE SDF SUPPORTS BUSINESS NETWORKS

Having discussed the broad architectural ingredients of the SDF, attention now turns to use cases that illustrate how it can support business networks. In addition to conceptual argumentation, practical insights based on how an SDF prototype, developed by SAP Research, supports the use cases, is discussed.

##### A. Creation of Service Marketplaces

The SDF provides an industrialized platform for creating service marketplaces for different business networks, in different industries. Figure 4 shows the conceptual architecture of an example to illustrate how this is possible. In the example a *One-Stop Citizens and Constituent Services Broker* operates an eGovernment service marketplace that is supported through the SDF Service Broker. The marketplace allows public sector and private sector partners, supported by the SDF Service Provider, to register at the broker and publish services. It provides full partner management so that users can administer their business relationships with consumers and intermediaries on the platform. A broker operator governs the marketplace. Agencies would be granted permissions for publishing services in a central directory of the SDF Service Broker, making these available to the marketplace. Services execute in hosted backend environments controlled by agencies and are interfaced through the Broker so that they can be interacted with (*mediated delivery model*). A Provider-Broker interface available in the SDF Service Provider allows secure and reliable interactions between the agencies and the broker, and it supports internal gateways of agencies to *plug-in* so that these gateways are not compromised as part of this. Setup of the Provider-Broker interface takes place before services can be deployed, published and on-boarded.

This complex process of provisioning is supported through a *service provisioning manager* (SPM) distributed across the Service Provider and Service Broker, which coordinates the necessary provisioning steps. Everything



usually starts with the assembly of the to-be-exposed service out of a variety of internal components, including service descriptions, UIs, processes, service interfaces and other implementation artifacts. The assembled service package is deployed on the broker where it is subjected to a range of checks and validations. It is expected that deployment of complex business services will not happen in a single transaction. Rather a series of provisioning activities takes place between the provider and broker to develop a version of the service that can be accessed through the marketplace. Once the technical and business requirements are satisfied, which includes specifying constraints for service access and repurposing, the service is properly registered on the marketplace. The subsequent on-boarding part of provisioning is concerned with the integration of various business delivery components (e.g. identity management, payment/billing, or advertising) provided by the broker to simplify delivery, as well as enrich the service experience towards end-users.

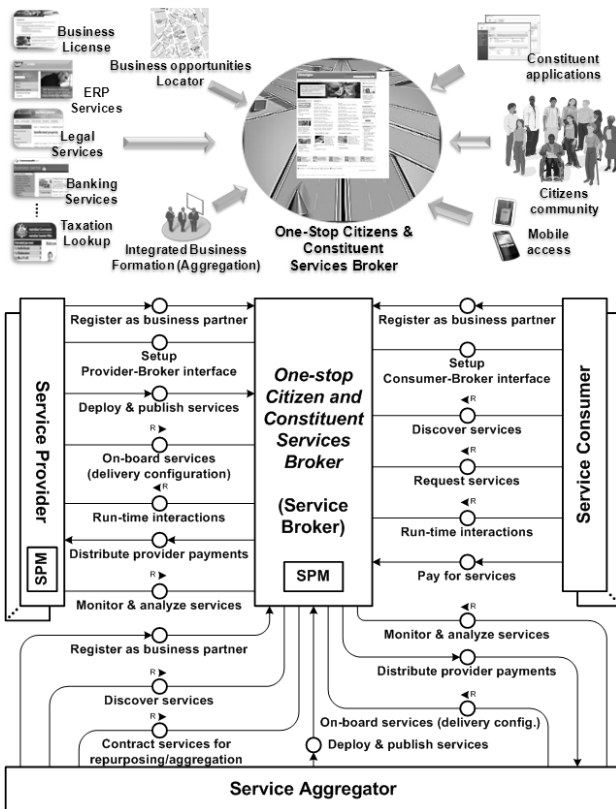


Figure 4. SDF support for service marketplaces (architectural layer modelled as FMC block diagram)

A variety of consumers can access the broker agency by being *plugged-in* through the SDF Service Consumer (involves registration and setup of Consumer-Broker interface). This part of the SDF integrates and executes in consumer environments, and it supports interfaces to the broker for service discovery, ordering, access, metering, payment etc. A potentially open-ended set of consumer points can register to access the broker. The pattern of

interaction for service access is a request-response for service steps: forms/UIs are displayed and filled in by consumers, they are sent to the SDF Service Broker which orchestrates execution of the service and used delivery components. It passes the request to the relevant service providers through the SDF Service Provider. Responses from backend service applications are returned to the SDF Service Broker and passed to the SDF Service Consumer for display through the consumer's device. The SDF Service Consumer can also be used to support just-in-time forms transformation for the channel-specific display (e.g. on an iPhone).

Some partners specialize in creating aggregated services. They would employ the SDF Service Aggregator to utilize different techniques for aggregation of services discovered on the Broker and creating new products. Before services can be contracted for aggregation, the Service Broker checks, if business constraints set by the provider allow the aggregator to use the service for the intended purpose. Permission, among other things, often depends on whether the aggregator built the aggregate for use in a single consumption context, or whether s/he intends to deploy and republish the service for access through the Service Broker. The latter would happen in a fashion similar to providers (using the Service Provider).

### B. Fostering B2B Collaboration Hubs

A major barrier for business networks to grow beyond static partnerships and inefficient process orchestration (basic fax, email and paper-based communications) is connectivity. Networks thrive when partners use a common application platform since a standard set of reference processes, master data and message exchange applies for the different partners. However, when partners want to interoperate services from heterogeneous applications and platforms, especially evident in wide spanning, global networks, the heterogeneity of service interfaces and message types used by them, will be a stumbling block. In this case, the services need to be adapted so that the message types will be compatible for service interoperability. For each pair of services interoperating, adaptors need to be created, leading to an explosion of adaptors as the number of services increases. Alternatively, a common standard needs to be established through which the individual services are adapted. This, however, assumes that a common standard can be established for all services.

The SDF can accelerate connectivity between partners because it allows different B2B gateways to be exposed and utilized through a broker for service adaptation. As discussed previously, B2B gateways are particular third-party BPO services that support outsourced interoperability based on the B2B standards that they support. If partners need to interact with a service using a certain standard (e.g. the VICS standard used by Walmart suppliers), then it can be selected so that an interface over that service can be created and used for interactions with the partners. B2B gateways allow for design-time interface creation (mapping of message types from the service and

the target standard), and run-time message translation through store-forward messaging. Figure 5 illustrates how the SDF supports partner connectivity via third-party B2B gateways. It shows the conceptual architecture of a supply chain working through a particular orchestrator or commercial network (e.g. the DHL network), indicated by existing line-of-business applications (upper-right hand side) operating within the partner companies. The partners have some pre-defined interoperability in place, like Web portals, email and faxes for communication. Some of the applications requiring business critical interaction would have dedicated integration mechanisms deployed (through software consultancy efforts). To extend supply chain partnerships, one-off integration efforts can be costly and inefficient. The SDF can improve efficiency for existing and future partnership growth. As shown in the figure, business services can be published through the SDF Service Broker where they can be adapted using third-party B2B gateways to improve the range of B2B standards coverage and reduce the effort involved in adaptation.

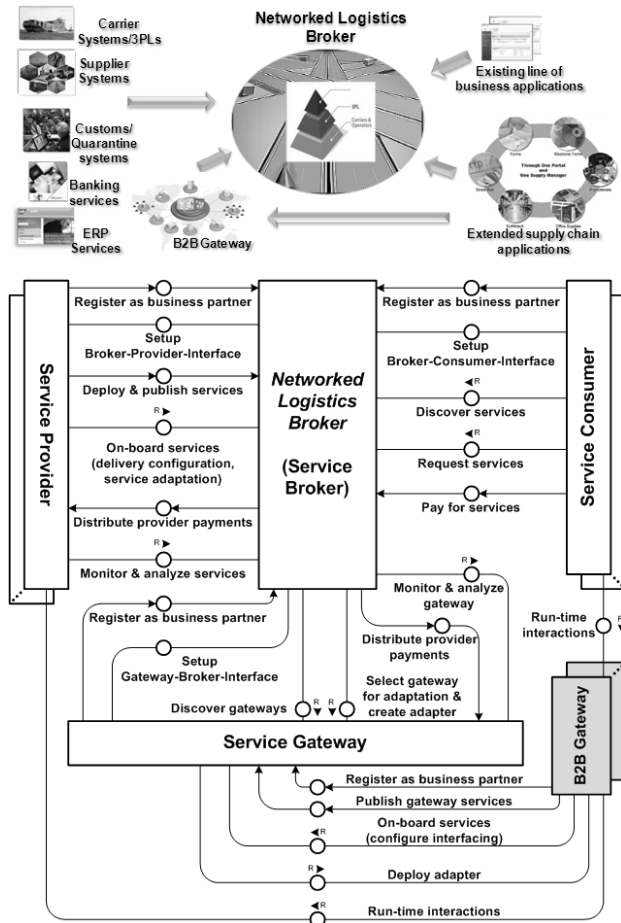


Figure 5. SDF support for B2B partner connectivity (architectural layer modelled as FMC block diagram)

B2B gateways such as Crossgate can be accessed through the SDF after being *plugged-into* the SDF Service

Gateway, a process that involves partner registration and publication of individual gateway services in the directory of the Service Broker. Actual integration of these individual B2B gateways is achieved through connection and configuration of interfaces in the Service Gateway, which are used to interact with the gateways. Towards providers the Service Gateway provides generic tooling for design-time adaptation of services into the various standards available through the different gateways. Towards consumers it provides a run-time façade interface which delegates calls to the B2B gateways that are *plugged-in*. When a service needs to be adapted (usually part of on-boarding), suitable B2B gateway services are discovered on the broker, a particular one (e.g. Crossgate) is selected, and the run-time adapter for the actual service is designed or generated. The adaptor is then deployed in the selected B2B gateway and the adapted service version gets published on the broker. When network partners need to interact with a service interface in a particular B2B standard, they can discover the version of the service that has the required interface and interact with it (either using the *reference delivery model*, as indicated in Figure 5, or using the *mediated delivery model*).

## V. RELATED WORK

Although SDF shares the interoperability concerns that classical SOA addresses, within and beyond companies, SDF's focus on interoperability is much more focussed. The SDF works at a higher level above SOA and the middleware layer, and focuses on the way services can be directly provisioned and delivered by different players in a business network. SDF is in fact geared to provide the platform blue-print for an integration and delivery platform for services and is not directly comparable to SOA.

The role-based structuring of the SDF can be compared with the Web Services Architecture [13], and its conception of service interoperability through service provider, service broker (based on UDDI) and service consumer roles. This architecture has been the central reference in the SOA and service sciences fields for understanding the way services are interfaced, discovered, accessed and consumed. We have significantly added to these conventional roles, under the development of business networks and other forms of service-based ecosystems, hubs and intermediaries. Our claim has been relevance, not necessarily completeness, of the refined set of roles. A further difference with the Web Services Architecture is the focus of the roles on provisioning of services – each role service to provision and potentially extend a service in a new settings including the service consumer role (where as we've seen, services can be programmatically interfaced to end consumer applications where they can be consumed at run-time).

Considering the different types of business networks, ecosystems etc, the SDF can be compared to the platforms that are used in these. Using the requirements we have identified for service integration platforms in business networks (discussed in section III), SDF has a greater

*diversity of services* compared with current generation *Appstores* since these concern business applications for the SME segment only. SDF, on the other hand, is designed to support a wide range of services including professional (i.e. human), business application, digital media, platform, and infrastructure services. SDF has a greater *diversity of provisioning partners* by comparison, since *Appstores* are restricted to providers, aggregators (developers) and a central broker. While their services are hosted, a single cloud is involved, and there is no scalability for managing interoperability through different solutions as conceived through the SDF service gateway. Service consumption is limited to order/access of service through a basic marketplace, hence SDF has greater *diversity of consumption* through its multi-channel posit and has a greater *diversity of service delivery* functionality since it supports mediated interactions for long-running services.

Another category of platforms is Service delivery Platforms (SDP) developed in the telecommunications sector to enable a core provider, namely communications service providers, to attract services from an ecosystem/hub, in this case digital content, mash-ups and entertainment services, and to allow these to be bundled and delivered with conventional telephony services. Diversifying service offers through an ecosystem of providers has been critical to the success in mass market growth of mobile and multimedia device users (OSS Observer, 2009). Prior to SDPs, communications service providers offered voice services through in-house, network-enabled multiplier developments. SDPs integrate class-carrier platforms, conventional middleware, service composition tooling and enterprise services for supporting exposure, aggregation, access and technical network delivery of services. The SDF can be seen as an extrapolation of SDPs. It provides a greater *diversity of services*, not just telephony and value-added content services. It supports a greater *diversity of provisioning partners* rather than a central (telco) aggregator/broker role orchestrating providers to multi-channel consumers. It has a greater *diversity of consumption* since B2B is out of scope for current SDPs and a greater *diversity of delivery functionality*, mixing business and technical delivery.

A third category evident in business networks is B2B connectivity platforms of which we have previously discussed different forms in the BPO segment. The focus of these platforms is to allow collaborating partners to interoperate across heterogeneous applications and application messaging standards. Some platforms like B2B gateway, e.g. Crossgate, focus purely on message translation. Others, e.g. INTRRA, Descartes<sup>15</sup>, e2Open are vertical trading platforms while others such as GT Nexus and GXS are a horizontal application platform by comparison. They are comparable to SDF in the diversity of services that they offer since they concern business transactions, however in terms of provisioning partners their sole focus is to aggregate and interoperate with services from their partners. SDF not only targets this

environment but also e-commerce marketplaces, hence it has greater diversity of provisioning partners, greater diversity of consumption and greater diversity of delivery functionality.

Table I at the end of the paper provides a summary of the comparison between SDF and these other platform categories.

## VI. CONCLUSION

This paper provided an architectural strategy for a more comprehensive and business network scalable services platform in the SDF. Using insights from industry specific corporate business networks, unfolding from B2B value-chains into global service hubs, it required four tactical characteristics for the SDF: support of diverse services - ranging from those that involve purely human activity, to business applications and processes, and out to platform and infrastructure functionality; diversity of provisioning partners; diversity of consumption modes; and diversity of service delivery beyond middleware to business applications, e.g. payments, CRM.

The result was a distributed platform functionally partitioned into key composites targeting key service provisioning and delivery roles that network partners play. The Service Broker with a business directory for services as the core allows services through the Service Provider to be published and on-boarded for delivery. It also allows technical services from Cloud providers and B2B gateways to be used, to allow services to be re-hosted and interoperated, through the Service Host and Service Gateway respectively. In addition to extending provisioning concerns to these forms of IT utilities, the SDF also supports innovative third-parties to repurpose and aggregation services and to drive new channels through the Service Aggregator and Service Channel Maker respectively. Through the Service Consumer role, an open-ended and remote set of consuming environments can discover and access services subject to delivery conditions notably making payments.

- [1] S. A. Mertz, C. Eschinger, T. Eid, H. H. Swinehart, and C. Pang, "Forecast analysis: Software as a Service, worldwide, 2009-2014," Gartner, Gartner Report G00201597, <http://www.gartner.com/>, July 2010.
- [2] R. P. Mahowald, "Worldwide Software as a Service 2010-2014 forecast: software will never be the same," IDC, IDC Report 223628, <http://www.idc.com>, June 2010.
- [3] V. Singla, "The overlapping worlds of SaaS and SOA," SYS-CON Media, <http://soa.sys-con.com/node/1047073>, July 2009.
- [4] J. Word (Ed.), Business Network Transformation: strategies to reconfigure your business relationships for competitive advantage, 1st ed., J. Word, Eds. San Francisco: Jossey-Bass, 2009, pp. xi-xv.
- [5] R. H. Brown, L. Scardino, G. Tramacere, M. Goldman, and F. Karamouzis, "Hype cycle for Business Process Outsourcing," Gartner, Gartner Report G00157425, <http://www.gartner.com/>, July 2008.
- [6] A. P. Barros, and M. Dumas, "The rise of Web service ecosystems," IT Professional, vol. 8, no. 5, Sep./Oct. 2006, pp. 31-37, doi: 10.1109/MITP.2006.123.

<sup>15</sup> Descartes: <http://www.descartes.com/>

- [7] P. Rodriguez, "Web infrastructure for the 21st century," Proc. 18th International World Wide Web Conference (WWW2009), April 2009.
- [8] S. K. Sharma, "An E-Government services framework," in Encyclopedia of Commerce, E-Government and Mobile Commerce, vol. 1, M. Khosrow-Pour, Eds. Hershey: IGI Global, 2006, pp. 373-378.
- [9] C. S. J. Palvia, and S. S. Sharma, "E-Government and E-Governance: definitions/domain framework and status around the world," in Foundations of E-government, A. Agarwal and V.V. Ramana, Computer Society of India, 2006.
- [10] European Union, "Directive 2006/123/EC of the European parliament and the council on services in the internal market," Official Journal of the European Union, L376, 2006, pp. 36-68.
- [11] C. J. Langley Jr., "The state of of logisitcs outsourcing: 2009 third-party logistics, results and findings of the 14th annual survey," <http://www.3plstudy.com/>, 2009.
- [12] P. Mottishaw, "Service delivery platform market share report 2009", Analysys Mason, <http://www.analysismason.com/>, May 2010.
- [13] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, "Web Services: Concepts, Architectures and Applications", Springer-Verlag, 2004.

TABLE I. COMPARISON OF SDF WITH OTHER SERVICE ARCHITECTURES AND PLATFORMS

Platform category	Diversity of services	Diversity of provisioning partners	Diversity of consumption	Diversity of delivery functionality
Appstore platform	Software applications for SME segment	Service providers (developers) and central broker	End-consumer model – to B2B support	Basic service broker functionality
Service Delivery Platform	Mostly content and telephony services – value-added bundling by telecos	Service providers (content service providers) and service aggregator/broker controlled by single provider (teleco)	End-consumer model – no B2B support	Combined technical (OSS) and business delivery layer (BSS)
B2B Connectivity Platform	Transactional services operating in companies in mainstream corporate industries (financials, logistics etc)	Service gateway with some service broker support	B2B consumption – some end consumer model	Technical adaptativity/connectivity but some business broker functionality (payments)
SDF	Professional/human, business applications, digital content, platform and infrastructure services available through central broker – USDL directory of services	Diversified roles for different delivery roles in service supply chain: service provider, service broker, service hoster, service gateway, service aggregator, service channel maker, service consumer	B2B and end-consumer models supported	Technical and business delivery functionality for different roles