Deep Science Publishing
https://doi.org/10.70593/978-81-989050-0-0_3

Chapter 3: Constructing scalable and interoperable digital service platforms for modern enterprises

3.1. Introduction

The recent acceleration of digital transformation for private and public organizations pushes them to rethink how to leverage digital technologies to improve productivity while providing new experiences to both their internal and external stakeholders. A central element of the digital transformation process is the development of digital product and service platforms that act both as enablers of business innovation and as instruments for the scaling of new products and services for consumers and businesses. Digital service platforms allow organizations to experiment with new ways of creating value by simplifying and accelerating processes that can heavily ease day-to-day activities in product design and development, supply chain and production, marketing and sales, internal and external communication, and privacy safeguarding (Baldwin & Woodard, 2009; Ghazawneh & Henfridsson, 2013; de Reuver et al., 2018).

If seen as federations of services, data, functionalities, and skills, organizations can use digital service platforms to provide their stakeholders – employees and management teams, consumers, investors, suppliers, partners, and institutions – with completely new services aimed at improving their daily lives. Employing a suitable mix of proprietary and third-party or externally sourced services, digital service platforms can shape completely personalized value propositions for the various actors in the organizational ecosystem. All these services, offered programmatically and reliably, improve productivity, provide intelligence for decision-making, and enrich consumer experiences.

The programmability characteristics of digital service platforms rely on the composability of APIs and microservices, i.e., the ease with which core functionalities can be combined and configured into new services to address changing requirements. Composability applies both to internally developed and externally sourced or third-party services. A typical feature of digital service platforms is therefore that they are best understood as a federation of internal and external components and services, according to a hybrid approach, allowing the easy response to the anticipated diverse internal needs and customer requirements that can emerge during various product and business life cycles (Tiwana et al., 2010; Hein et al., 2020).

3.1.1. Overview of Digital Transformation

The phrase "Digital Transformation" refers to the reallocation of tangible resources and the restructuring of concrete activities of organizations. Digital transformation exhibits far-reaching economic and social consequences, re-conceptualizing the nature of value in service transactions and the formation of collaboration networks. Digital transformation is closely related to a breadth of additional terms, including Digital Business Transformation, Digital Transformation, Digitalization, Digital Disruption, Digital Convergence, Business Model Innovation, etc. Generally, these different notions emphasize various perspectives of digitization development. With the increasing popularization of digital technology, the concept of Digital Transformation generally refers to the process by which enterprises send visible changes to be customer-oriented, business-capable, and ecosystem-based by utilizing digital technology to build digital operations. During Digital Transformation, enterprises build their digital behaviors and manifest changes through such stages as the Digital Capability Model Digitalization Business Model Innovation Digital Transformation of digital technology.

Firms cannot simply use digital technology as instruments to achieve the goals of their traditional business. The rapid development of digital technology has empowered the emergence of new consumption patterns, business models, and influencer networks. New competitors have entered the market. Some of them are startups, but others are old players who have completely revamped how they operate using digital technology - new medicine players, new online finance players, and new retail players. Furthermore, the current accelerating Digital Transformation of human society distinguishes any enterprises from various environments that have cross-regional, cross-product, and cross-institutional characteristics. Additionally, Data-centric Digital Transformation actively integrates internal and external data and encourages real-time and predictive decision-making capabilities, thus enhancing organizational resistance to risk.

3.2. Understanding Digital Service Platforms

Digital Service Platforms (DSP) aggregate, package, and expose both business and algorithmic capability as services, presenting a surface layer of enterprise-facing marketplaces, or a combination of both for developers and partners to consume with minimal to no friction. We distinguish between service platforms and service APIs that both consume external or internal services. The latter inherently lack a discovery or governance mechanism that would prevent developers and partners from having to guess which of the many endpoints that expose technical capability might be suitable for their use case. The mechanism would keep the enterprise-facing marketplace, with its semantic service model and zero-code interfaces, in sync. Traditional companies book a sizeable percentage of their revenue or savings on “manual” IT and cannot afford not to industrialize it. For example, financial and professional service companies allocate about 50% of their workforce to such tasks.

Infrastructural service platforms are critical to aggregate, industrialize, and expose these business capabilities, creating a service ecosystem around enterprise functions. The importance of digital service platforms knitting together the enterprise’s functions comes from their ability to manage complex business processes and rules. They effectively provide a messaging and orchestration function, bringing external and internal service mediators for business and algorithmic capability to an enterprise’s functions. The difference between a functional and a path-driven service layer is that the former would still have developers create and code every user journey, in addition to implementing each of its atomic steps via service composition. The latter allows users to compose and orchestrate unique user journeys without needing to code a single journey. This capability transfers much of the creative design work over to the business domain, with developers focusing on creating innovative and reusable component services.

Fig 3 . 1 : Digital Service Platforms DSPs

3.2.1. Definition and Scope

Almost two decades ago, an entire branch of enterprise systems design and architecture started moving from the computer-centric infrastructure used to design and run such systems, onto the new Internet web-centric infrastructure. Early pioneers in that move turned IT from a difficult and often frustrating support function into a strategic weapon for both customer engagement and immediate non-functional concerns like availability, reliability, and scalability. In the past few years, a plethora of alternative approaches in the form of cloud platforms of many types, marketplaces of Internet and mobile applications and services, integration and API platforms, automation frameworks, etc. have brought us to a cacophonic stage in that evolution. On one hand, enterprises and organizations have been provided with an enormous number of options to leverage that paradigm shift. On the other hand, they are struggling to make the right choices, integrate solutions coming from multiple vendors, and find unique elements of differentiation able to justify the often exorbitant costs involved. While choices vary by level of IT specialization, many small and medium enterprises are simply waiting to be presented with an integrated solution that saves them costs while boosting their growth potential in an increasingly complex marketplace.

Digital Service Platforms are flexible and scalable collections of integrated digital services, hosted on a cloud computing infrastructure, and serving as a core element of enterprise Systems of Engagement. Digital Service Platforms allow enterprises to build, develop, and host Digital Services: focused applications used to engage customers, suppliers, and partners with digital experiences that are attractive, usable, and integrated around the concept of a journey. Digital Service Platforms can integrate Digital Services from across third-party Internet Service Providers and use those digital services together with internally developed ones to design and run enterprise Systems of Engagement addressing specific enterprise functions or Operations-level Processes like Sales, Marketing, and Customer Service Management.

3.2.2. Importance in Modern Enterprises

Digital service platforms (DSPs) are characterized by two major roles: the managers of the platform with control over the basic resources and the third-party developers that deliver complementary services on top of the platform resources. In addition to this, the following eight characteristics can be considered as defining the scope of a digital platform: Co-Creation: The provider collaborates with users, customers, or other partners to create an experience for the user or customer. Modular Architecture: The component systems and the third-party solutions and applications must be able to integrate easily and leverage the capabilities of the Foundation Platform. Plug-and-Play Operations: The component systems and the third-party solutions and applications must be able to start operations and plug into the Foundation Platform. Marketplace Model: These solutions and applications need to operate in a marketplace scenario with physical or virtual trading. Accessible Sub-Bus: There must be an expandable and open sub-bus that will enable both the third-party developers and the owner to exchange information and validate that the third-party solutions and applications are authentic and are allowed to interface with the Foundation Platform. Configuration Requirement: The Foundation Platform will need to allow configuration and personalization to enable third-party developers to offer its solutions and applications for specific needs or preferences of segments or individual developers. Interaction Support: The Foundation Platform must provide the infrastructure and services for the third-party developers to provide an interactive environment for the users of the solutions and applications developed by third-party providers. Solution Multiplicity and Variability: The Foundation Platform must be able to host a variety of similar solutions or applications so that different usage profiles can be met. This enables the configuration or personalization requirement of the Foundation Platform and the solution multiplicity and variability characteristics to be offered efficiently.

3.3. Key Characteristics of Scalable Platforms

Digital technology and the unlimited scalability of host infrastructure is a primary motivator in the shift to platform-centric enterprise strategies. Companies have proven that the right platform strategy enables enormous market capitalizations and company growth. Small and medium enterprises look to adapt successful digital strategies to their more modest operations. In doing so, they build a localized digital platform around local enterprise value propositions, supported by a base of customers, suppliers, and service partners. For both large and small companies, the ability to quickly increase market share and infrastructure resources is of paramount importance. Building correctly for scalability opens up market opportunities, whereas incorrect design strategies hinder business growth. At best, and unless significant investment is made, growth is constrained, and market opportunity is lost for the foreseeable future.

For small to medium companies, the potential for constraint and missed opportunity is significant if platform technology choices and implementation strategies are misaligned with business goals. In the worst case, the chosen technology becomes a liability as these companies increase market reach and customer services. Correcting the situation is fraught with cost and risk and may even lead to business demise during periods of intense corporate growth. Thus, small and medium enterprise use of these digital service platforms requires a focus not just on the criteria for choice, but also on the multiple technical and architectural dimensions that may influence the future resource utilization and performance of the technologies in a production business environment.

3.3.1. Scalability Defined

In enterprises, the term scale is often associated with workforce size. Also, the scale of an enterprise is understood mostly in a quantitative manner: the amounts of its assets, accounts, employees, locations, production equipment, clients, customers, services, and products. However, all these assets and the operations related to them represent a specific type of enterprise scale. Moreover, not all enterprise scales are equally important. Indeed, different types of enterprise-scale are stressed by different enterprise architects and enterprise architecture experts in the context of specific business goals, taxpayers' needs, specific instruments and models, and other specific architectural decisions related to enterprise construction. Predictably, scalable platforms for enterprise digital ecosystems should be capable of supporting different types of enterprise-scale recognized by enterprise architects and experts in the context of their specific enterprise ecosystem models, business goals, and architectural decisions.

That makes enterprise deployment of scalable platforms for digital service ecosystems a challenging task. Currently, three types of enterprise-scale are usually emphasized: growth of the amounts of assets, the scope of enterprise operations, and diversity of enterprise services, products, and resources. As a consequence of the IT-centric nature of the digital service ecosystems of modern enterprises, almost all definitions of scalability adopted in the context of these ecosystems are expressed in terms of enterprise IT capacity. Indeed, the smart city, as a realization of the digital twin for a modern city asset management enterprise, is usually considered an interconnected system of distributed sensors for the real-time collection of heterogeneous data describing urban life processes. These emphasize how important diversity is for scalable platforms deployed in the context of conceptually similar yet unique smart city environments. Over the last decade, the scalability of cloud computing capabilities has been substantially enhanced. Since modern digital service platforms for enterprise digital ecosystems are commonly hosted in the cloud, these enhancements can be used for the platforms in question.

3.3.2. Vertical vs. Horizontal Scaling

The architecture of a system can be depicted both as its physical layout and as its logical layout. The logical architecture describes the composition of the overall system, the definition and responsibilities of the different logical components, and their interaction protocols. The physical architecture specifies the mapping of the logical components on the computers of a system network. In both cases, the architecture can further depict the internals of the computers involved, including their processors, memories, and I/O peripherals. For general-purpose computers, it is common that each processor is part of a multiprocessor, its memory is shared with the other processors of the multiprocessor, and the I/O peripherals are also shared by the different multiprocessors of a system. For special-purpose embedded devices, each processor can have its private memory and I/O components.

When a system uses an architecture of multiple computers, which may or may not be multiprocessor computers, the logical architecture can be changed at any moment by adding or removing components. Such a change may have a low impact on user programs, especially if the interaction protocols between the platforms remain unchanged. For example, consider a platform with three replicable components defined by its logical architecture. The vertical scaling consists of replicating on an additional computer the component that has the highest workload. The horizontal scaling consists of removing one of the replicable components and the added component performs the module's functionality with the most reduced workload. The horizontal scaling can be done widely by having the same type of additional computer for the platform, and only changing the mapping of the modules in the logical architecture.

3.3.3. Performance Metrics

Imagine a shopping mall. The more visitors that enter, the more sales owners make and the more property taxes the local government can collect. Up to a point, there’s a near-perfect relationship between visitors, sales, and taxes. But what happens when the number of people exceeds the mall's design limits? Too many people trying to enter the mall through too few doors cause queues to build up. Shops become crammed, and sales eventually dwindle because the displacement costs for leaving are too great. Would-be customers complain, and some even leave without purchasing, causing a loss of revenue. Consumers eventually avoid the mall altogether, reducing overall traffic and even negatively impacting local business owners and local government. Now consider all this at a cloud scale, due to the hyperscale nature of cloud infrastructures, servicing millions and billions of users and transactions every day.

Two key performance metrics are primarily utilized to characterize the performance of scalable cloud services; the capacity to handle an increasing number of requests and the execution time per transaction when under load. It is typical to use response times under various load levels as the basis for performance evaluation. In nearly all cases, it has been observed that executing a transaction under load will take longer than the execution time of an idle system. Empirical studies have shown that response times deviate more and more from the idle time as the load level approaches capacity. Above capacity, however, there is an abrupt transition behavior that is the most striking characteristic of overload states. There is often a local maximum response time at capacity, after which response times drastically change. About that behavior, we can distinguish between two types of platform services. A platform is considered overloaded if all response times are becoming prohibitively long or if users avoid using the service because of a worsening experience.

3.4. Interoperability in Digital Services

Increasingly, enterprises recognize the need for connected services that provide synergy beyond the boundaries of the enterprise. This demand creates a pressing need for service interoperability. Consequently, there exists a plethora of definitions, strategies, standards, initiatives, libraries, frameworks, implementations, and products for service interoperability, together with some not-insignificant segments of enterprise loss and frustration. Addressing the demands for meaningful characteristics of service interoperability requires a clear approach to defining interoperability. The purpose of this chapter is to clarify some aspects of service interoperability definitions and associated work.

What do we mean by interoperability? Interoperability is what allows digital service networks to work together, and without it, there is no connected economy. Whereas the importance of network interoperability for telecommunications networks is widely recognized and has been the source of significant effort, discussion standards, and implementation work in the telecommunications domain, for other kinds of service networks this is more overlooked. In the Information Systems domain, interoperability has a strong connection to work on integration within and between enterprises. The wider challenges for enterprise information integration drive the need for interoperability in information services. From the perspective of parties to transactions, the interoperability question asks how many different kinds of digital public service connections are required to connect to all others. Consider Banks and other Financial Services providers. Existing knowledge – Gaps, Bridges, and Islands of Information. The problem might be ignored because Service Providers know how to work together without digging deeper, they are not eliminated. Enterprise Collaboration costs can be reduced by removing some barriers, Examples exist workaround, without formally making tweaks to "Public" Pharr, "Private" Pharr, and "Public" Directories.

3.4.1. Defining Interoperability

Digital technology enables organizations to service their clients more securely and more efficiently through yet at times less expensive solutions. Such costs involve risks that require careful planning, testing, deployment, and management of digital service systems and their connections. These services are typically developed using a mix of cloud or on-premise service silos that can be standalone or mapped onto other services controlling touchpoints through which end clients interact with the enterprise. Business needs and innovation assume different forms at different points in time over the life cycle of the enterprise and yet the enterprise is composed of digital services that have a natural tendency to grow increasingly interdependent and entangled. As organizations advance their dependency on collaborations with clients, suppliers, industry partners, and other organizations or potentially competitors, enterprise strategies require consideration of the evolving status of the international domain name system, the transit routing system, certification authorities with their registries, industry standards or the absence thereof, digital identity and reputation systems, and their correct implementation through digital services deployed by service partnering organizations.

Consequently, interoperability should become ingrained in the digital service corporate strategy, and more specifically in a code of practice, that surveys these six foundational stakeholders at any point in time, defines their expected practices, and states their roles in terms of mutual expectations and dependencies over time as transformative technological innovation offers organizations increasing access to new forms of collaboration. This code of practice should focus on such different dependencies through their interoperability with regard to their intended purposes and relation to specific predefined sets of standards. Supported by shared language, process, and models, this code of practice allows the enterprise partners to define and use their sets of semantic and syntactic interoperability models throughout their desired levels of practical automation and digital service transparency. It also enables them to share, negotiate, and track any level of potential non-conformance, whether it be proactive or reactive from an enterprise risk perspective.

3.4.2. Challenges to Interoperability

Interoperability can be difficult to achieve. Significant barriers may arise in its people, technology, and business aspects. In the content domain, the primary issue facing digitalization is to convince politicians and civil servants (and their partners in the private and third sectors) to publish data held on behalf of the public in standardized, interoperable ways. For instance, making the relevant format available for creating invoices, signing, and verifying digital signatures could promote stability and content interoperability in the economic domain. The semantic aspect of digitalization is even more complex. Creating an infrastructure on which the multitude of applications in the public sector, such as citizen registration systems, tax collection systems, and company registration systems, as well as those in the private sector—such as the Chamber of Commerce, company audit, and commercial partner systems—can share and use data autonomously and sustainably is an incredibly difficult task requiring strong cooperation.

Technical interoperability is challenged in the transition from premises-based infrastructure to cloud-based services. A first step toward this new model for the delivery of digital services is to move applications to cloud service providers. Interoperability between private partners and government agencies on the capabilities provided by such cloud infrastructure services—such as digital signatures and verified data exchange—is demonstrated through the capability proven by various projects. However, simply porting legacy applications to the cloud does not guarantee interoperability within this new ecosystem. Constraining data formatting and formatting, service calls, and the security aspects of the services provided on assurable application services are performed through government-provided management subsystems.

3.4.3. Standards and Protocols

Among the different computerized solutions that attempt to practically implement digital services in environments perceived as being interoperable, the concept of standardized language presents itself as the core foundation. Such a standardized language serves as a protocol for conversation among the participating entities; this particular protocol functions as an intermediary between all involved partners of the existing interface; it ensures that all involved parties synchronize their actions accordingly; and it communicates information released from one partner to the other pertinent ones.

A standardized language can consist of a specific graphical representation of objects and services and of the way the individuals structure and describe objects or services using a linguistic-type description. The former is primarily concerned with how the different entities are expressed and queried, while the latter relates to the policies associated with the interpretation and exchanges made on the data layer. Moreover, standardized languages can exist under different technical levels, such as in the technical level enabling data structure standardization via a defined meta-data description, in the message couple level, in the ontological level, or at the API level.

However, an existing problem at the technical level is that different enterprise partners often use distinct layouts of the same language; thereby preventing exchangeability. Due to this, a technical layer based only on the definitions of description languages does not necessarily provide interoperability. Thus, API standards become crucial for enabling real business interoperability.

3.5. Architectural Considerations

Enterprises looking to adopt an agile digital service-centric approach need to consider new architectural and technological solutions to support such a vision. With digital services becoming the primary means of interaction for modern enterprises, the DCP levels the ground so that the enterprise services can support interactions at a worldwide scale and support trustworthy interoperability with third-party services. The DCP becomes a partnership decision - where partners are then incentivized to build their services on the DCP as the service fabric that offers interoperability capabilities, private data sharing controls, along user-led service design oversight. The DCP solutions have to be operationally very efficient since the value derived from attractive partnerships would depend on the capabilities of the platforms supporting the partnerships; going to the next level and building a mesh of interconnected enterprise DCPs.

1. Microservices Architecture There are numerous technology choices to build Digital Service Platform solutions. Digital services have a micro-front end and micro-back end structure - as the user interface and back-end services exposed through APIs, are independently managed and designed for maximum re-economy for rapid and agile design, interaction, and use. Above such backend services, API management becomes an essential facet of a DCP solution. The most essential requirement for API management is usability for non-developers enabling users to find APIs and services for their requirements, using the API, along with test and documentation capabilities. Supporting interaction and service use tracking is also considered important.

2. API Management User interface design-as-a-service based capabilities in the form of low code or no-code platforms that can interact with programmable API service and can provide data, interaction design services are necessary. Tools have started providing such capabilities that allow enterprises to develop lightly governed but fast DCP projects. These tools use templates, patterns, and AI, along with adaptive and programmable connectors, which require some stabilization for popular third-party services but have capabilities to connect to any service to facilitate the building of integration solutions quickly.

3.5.1. Microservices Architecture

Architecture plays an important role in the overall success of a Digital Service Platform, and therefore, there are some architectural considerations that need to be addressed in order to implement processes and form data pipelines. The first point is about Microservices Architecture. The emergence of cloud computing and its enabling technologies, including virtualization, containerization, and orchestration, give rise to a shift in the design of enterprise systems and applications from a monolithic architecture towards a distributed microservices architecture. In the monolithic architecture, all different functions that are managed under the organization are packed together into a single unit, thus delivering a single bundle of functionality. Although the monolithic architecture enables fast code execution, it lacks scalability and extensibility. Besides, due to shared processes and data, it is difficult to maintain and deploy the code independently, blocking the development of new features in time. On the other hand, microservices architecture contains microservices that are loosely coupled, lightweight, and independently deployable, providing a highly decoupled architecture for modern enterprises. These microservices focus on specific business capabilities or data pipeline functionalities, like login authorization, search and recommendation, tracking user journeys, and storing user behavior data. With a microservices architecture, organizations can continuously integrate and deploy their applications, thus enabling rapid application development, and some other technology companies using this approach can release new features, updates, or fixes into production multiple times a day.

However, building a microservices architecture does not come without cost and challenges. There is an upfront investment to decompose the monolithic codebase into microservices, especially if the code is large, providing a complicated migration journey. Combining multiple databases is very helpful when moving toward a microservices architecture. Each microservice may have its database to hold its state, thus being truly decoupled and independently deployable. So building multiple data pipelines is a good practice here. Other related technology challenges include network latency, complexity with distributed systems, cross-cutting concerns, more difficult debugging, and storage overhead. Due to these benefits and challenges, an organization needs to assess and analyze the requirements of its applications before deciding to migrate to the microservice model.

3.5.2. API Management

An API is the software component that enables highly productive interactions and collaborations between applications, data, and service capabilities running on increasingly complex enterprise IT infrastructure. Typically, APIs expose some software service or capability to enable internal and external users, partners, customers, and third-party application developers to easily integrate and interact with to create value and drive revenue for the business. Today, the API ecosystem is expanding to enable enterprises to rapidly position new businesses, explore new markets, and harness new opportunities. Increased interconnectivity and growing demand for composite services that rely on enterprise services and third-party services, data, and capabilities for the delivery of business value are driving the development of enterprise APIs. But post-2008 in particular, the emergence of Web 2.0, increasingly popular Web and mobile applications, social networks, and other third-party services and applications are driving enterprises and service providers to develop supplier APIs. By coupling a defined API strategy with emerging market forces, enterprise business leaders can influence how the

API ecosystem evolves in terms of its underlying design principles and governing structures and mechanisms as a means of competing for commodity products and service capabilities and collaborating for differentiation. APIs are a key enabling technology for enterprises to create a digital mirror of their original business. By implementing a business-first approach to enterprise API development, enterprises drive APIs to create business value.

Fig 3 . 2 : APIs Power Modern Business

3.5.3. Data Architecture

Digital Service Platforms (DSPs) and the diverse set of Digital Services (DSs) that utilize them are designed to support large numbers of service requests from many clients, potentially including both internal users and external consumers, and must therefore allocate data among the various DSP Component building blocks and DS implementation code layers in such a manner as to optimize throughput, minimize response time and maximize availability while minimizing costs. The data architecture of a DSP must also provide the DSP with the capability to scale in a mass-market, multi-customer environment in such a way that it can cost-effectively support the creation and ongoing maintenance of potentially hundreds of thousands of DSs. The data architecture of a DSP can be viewed as a collection of logical entity types, called information objects or data entities, along with a distribution model that determines where each of the entity instances of each entity type will be located, usually a mix of independently defined local resources along with pools of shared resources. The DSP data entities provide the DSP with a set of identifiable data objects that contain descriptive information needed for processing service requests and a cognitive model by which the DSP can interface with the business world. The DSP data entities should share an underlying logic model and related set of description standards in a manner that avoids unnecessary duplication of identical or very similar pieces of information or relationships between pieces of information. Without such facilities, the DSP will be unable to implement the data architectures needed to meet the scalability and cost objectives described and will resort instead to a more traditional monolithic architecture that performs the same or similar functions for many different Services, Business Activities, and Clients.

3.6. Technology Stack for Digital Platforms

The technology stack is a major decision point in developing a digital service platform. The chosen technology stack has major implications for how scalable, reliable, and flexible the resulting platform is. In this paper, we will discuss a few key technology choices that the digital platform development teams should consider. This technology stack is proposed based on our experiences of developing two major platforms: a vulnerability discovery and management platform and an identity and fraud mitigation technology platform. A critical list of technologies chosen for these two platforms is summarized in the next section.

A major choice of technology stack for cloud-based digital service platforms is the choice of cloud computing solution. Many companies across the world, including digital natives, have taken the plunge into the cloud. Public cloud computing services allow on-demand, self-service, flexible resource pooling, as well as payment based on actual usage. Companies need not worry about building physical infrastructure like computer servers but focus on writing business-critical services. Public cloud computing services provide the proverbial "run and forget" functionality where providers take care of deploying services, scaling performance during demand spikes, and performing automated healing as well as upgrading. The deployment of on-demand capabilities at the edges of the digital service platform infrastructure would allow them to scale elastically as and when required.

While the convenience of cloud functions is extremely attractive for building the digital service platform, many companies also recognize that they could incur large costs if the cloud functions are utilized heavily. Management of cloud computing budgets can become cumbersome and complex. So, while companies should explore the suitability of serverless computing models provided by cloud infrastructure vendors, they must also evaluate the economics of using virtual machines and containers compared to on-demand functions for core business service logic.

3.6.1. Cloud Computing Solutions

Many of the challenges of modern enterprise cannot be practically solved using the traditional on-premise approach to serviced infrastructure provisioning and management. Whether in the form of a full-fledged service center or bundles of servers in a data center room under the stairs, on-premise services are complex, costly, and troublesome to deploy, grow, and maintain. The main disadvantages are that initial purchases and setup are time-consuming and complex, with tiered capital investments to access the required capacity. Up-sizing or down-sizing existing deployments is painfully complex, requiring physical provisioning and setup. Maintenance overhead has a proportionate high cost and often does not deliver sufficient – if any – long-term cost advantage compared to outsourcing. Lastly, practically, enterprises are seldom in control of provisioning and management, either relying on an understaffed internal service team or external contractors. Neither option delivers the necessary flexibility, expertise, or attention.

As a result, choosing a Cloud-based solution is often a more suitable and cost-effective approach to building the technology foundation for Modern Digital Platforms. Cloud-based services are infrastructure built using the multi-tenancy concept, where Cloud Providers invest in and take over making the necessary tiered-capital investments, including cost-efficient provisioning, maintenance up-sizing, and down-sizing physical and virtual resources. Enterprises then use Cloud to provision what they need, in a pay-for-what-you-use model. Cloud Providers make huge investments in their services to deliver high-availability and reliability Service Level Agreements that would cost enterprises orders of magnitude more if built in-house. This means that enterprises can concentrate on their core business and capability, rather than on expensive complements that – while necessary – do not differentiate their core offering.

3.6.2. Containerization and Orchestration

Due to requests for diverse resources, digital service platforms need to rely on pooled computing resources, deployed to address specific demands at a specific time. We already covered how cloud computing enables pooled provisioning of IT resources, but how can the concepts of provisioning of logical resource templates be taken into the level of the service logic workloads, those that run the operation's actual business logic? This can be achieved in physical clouds using containers and orchestration engines to deploy them. Containerization refers to the techniques and tools that allow to package workloads in lightweight runtime environments called containers, which behave as services that can be started and stopped on demand. Rationally, containers are the discretizing element of a service platform when policies and workload granularity are defined in a way that service execution occurs inside a container.

Having small yet powerful runtime environments is not useful if they cannot be allocated in required amounts whenever needed, motivated by the need to reduce allocated resources to the bare minimum to reduce costs, as well as actively balancing the existing workload across containers in the infrastructures where they are allocated. Allocation and load balancing are the responsibility of orchestration platforms, whose workloads are in constant contact with the IaaS interface and/or agents running in IaaS nodes. Orchestrators are capable of elastic scaling of both containers and infrastructures accredited to run those containers, as well as automating processes of container deployment and ongoing management, therefore allowing DevOps practices. Even different orchestrators can work together, as larger platforms can be composed in this manner.

3.6.3. DevOps Practices

In addition to the cloud computing and containerization technologies discussed in the last two subsections, the way organizations build up services with fast iterations, how often they deploy services, and how long the outages last are also important indicators of a modern enterprise's technical capability. Modern enterprises promote sophisticated engineering and coding practices. Such practices have been best described with the term DevOps, which signifies the close collaboration between the development and operations teams, who were traditionally kept separate. The core idea of DevOps is to reduce cycle times by enabling fast, automated deployment and subsequent monitoring so that, if any aberration or failure is detected, the service can either be rolled back to the previous stable version, or appropriate remediating action can be taken to fix the fault.

Continuous deployment can release new features, fixes, and amendments at a rate dictated by business constraints rather than technical capabilities. Continuous monitoring enables enterprises to react quickly to failures. Automation is the key: the goal of DevOps practices is to automate as much of the software delivery process as possible, so that well-tested code is automatically assembled as binaries with a set of scripts, deployed as containers, and their execution load balanced across multiple instances, and monitored for behavioral anomalies throughout their life cycle. Continuous integration makes it possible for developers to merge their code changes onto a shared mainline several times a day. Automated as much as possible, verification checks can detect problems quickly and early. Frequent and small releases enable enterprises to reconfigure services to deal with ever-changing business circumstances.

3.7. Security and Compliance

Data security and regulatory compliance are critical to any digital service project and may have an impact on the architecture, usability, and operational efficiency of a service platform. Regulatory compliance ensures that a company adheres to rules and regulations in how it manages private data and what controls it puts in place to protect that data. Cybersecurity strategy defines the various data security measures that are implemented to provide a defense-in-depth approach to security.

1. Data Security Measures

While most companies feel confident that they have a good handle on data security, a surprising number discover that they don't even know what data they must protect or the nature of that data. To start, data must be classified according to its nature, sensitivity, regulatory requirements, and importance. This is as simple as sorting it into buckets, but it is critical to know all of the data that is used and its context. The companies then define the parameters for what constitutes sensitive data that requires special controls, and this is typically based on the volume of traffic.

2. Regulatory Compliance

Implementing an enterprise digital twin needs to take into consideration industry regulatory compliance requirements for how data is protected data. For example, if a company is in the Healthcare industry, it may be required to comply with the provisions and that means all platforms that touch patient data must follow the access control and encryption requirements as detailed in the regulations. Another industry regulation example is the GDPR, which governs what companies are allowed to do with Personally Identifying Information data. The GDPR outlines what companies must do with acquisition, storage, access, and data deletion.

3. Risk Management Strategies

Once a digital service platform has been developed and put into production, a company's risk management strategy takes over to ensure that adhering to the data security policies and regulatory compliance rules outlined in an earlier chapter is followed. Risks can come from not implementing the existing policies or incidents outside of controls put in place for business continuity.

3.7.1. Data Security Measures

Moving any type of corporate or private data to the Cloud must consider specific information security issues and concerns. Protecting your data once it's passed over to a cloud service, whether it be infrastructure, a platform, or some software, is significant.

To competitively sign off third parties to control an Enterprise's cash flow, operations, and/or business intelligence capacities, the Enterprise must ensure that its data is protected adequately by using as much as possible the most common and powerful data security measures that can help minimize breach risks, whether it be a technical or non-technical concern that a Company may have.

There are many strategies for preventing unauthorized access to business data and its illegal exploitation. Identity and access management guarantee that the right individuals have the appropriate access to the computing resources to which they are entitled. One way to do this is to perform fine-grained access control consistently while understanding that it comes with the cost of complexity that can lead to security weaknesses because not all companies or enterprises can handle efficiently the management of IAM devices and services. Integrate privilege management into the IAM strategy through business-related processes to issue access provisioning; evaluate privileges periodically; monitor the elevation of privileges; and regularly update the separation of duty and routing controls. Best practices recommend defaulting to the least privilege, regularly optimizing permissions, and maintaining clear accountability over privileged access and activities.

Businesses may enforce rules on individuals' accounts to require all access to use secure authentication such as multi-element authentication. Data encryption is a technique for coding into an irregular character or text, which protects it from being read by anyone without knowledge of the relevant private key. This regard can be done while data is "in flight" as it traverses the internet and "at rest" where it resides on a hardware or on-memory store or asset, or where it is generally exchanged between servers. Businesses can act for both purposes through well-known algorithms based on public-private keys, as well as shared secret keys.

3.7.2. Regulatory Compliance

Enterprise digital service platforms are called to tackle an increasingly large number of requirements regarding regulatory compliance, in many areas. Safety and reliability laws and policies require that service providers not only offer a high-performance service but can also guarantee that the service is always available and working according to its specifications: performance, latency, security, etc. Data protection laws impose bounds on the type of information that enterprise platforms can collect and process. What is worse is that these compliance requirements are a moving target, as new compliance regulations and policies are continuously being issued.

Enterprise service platforms are required to maintain the operational and technical structure that allows internal and external audits to guarantee the confidentiality of data that can potentially cause damage to any of the users of the service or external exploitations due to unintended data. They have to allow also auditing systems that check for security holes that hackers could use to assume unauthorized privileges, change processes, or influence the good operation of their execution. These platforms should provide an allowed operational region that minimizes the probability of facing a disaster that could excessively disturb the normal operation of regulated services – for example, a cooperative computing service involved in a judicial investigation and exposed without security measures.

3.7.3. Risk Management Strategies

Risk management is a fundamental part of smooth business operations. Enormous enterprises face big operational areas as well as big distribution of their business activities and hence they need appropriate and sometimes complex risk management strategies. Generally, risk management strategies can be classified as reactive or active. Reactive strategies implement corrective actions after the occurrence of risks. On the other hand, active or preventive strategies distinguish between risk avoidance and risk control. Management can decide to avoid the risk and therefore give up a business activity. However, the acceptance of certain risks is an inherent part of business processes. Consequently, risks cannot be fully avoided, but risk control strategies allow for limiting risks to an acceptable level. They reduce an organization's vulnerability by strengthening its resilience or capabilities and by implementing measures that enhance the organization's ability to recover from negative impacts and restore operational efficiency. Risk reduction controls lessen the probability of risk occurrence and lessen the severity of damage of those risks that will occur, like contingency measures.

To manage risks, traditional enterprise risk management emphasizes an integrated and holistic view of process-related enterprise risks. External from the organization's point of view, the portfolio of business activities creates revenues. Internal from the organization's point of view; the operational structure contains business processes, which convert physical products and services. The appropriateness of risk management strategies is confirmed by the results of a multi-criteria decision analysis. It evaluates the effect of decision criteria diversity, criteria independence and criteria interdependencies, risk, and benefit scalability on the basic characteristics of the risk management strategy. In more formal terms, the independence of risks is guaranteed, and security offers the highest reliability concerning false negative errors, while risk and benefit scalability allows reliable risk and benefit estimations. Business process risk management is able to define the quantitative key performance indicators.

3.8. User Experience and Interface Design

Modern digital services, which share many similarities with software applications, are also expected to have an engaging user experience. Service users, in addition to being offered a user-friendly interface to interact with, are also expected to find the overall service satisfactory. This means that, in addition to being visually appealing, the interface design needs to consider how it meets the capabilities of users and how it connects with the organization behind the service. Digital service platforms enable the design of various services through specific combinations of reusable components and business logic. However, it's important to ensure that every service component meets the same user experience criteria. Service users need to find all service components consistent and seamless while using the service. How each component behaves and the design style should be similar, if not identical, to be perceived as coherent and transparent. Service development teams should then work together to ensure that, even if they are responsible for different service components, they follow the same design guidelines and patterns. While service configuration and digital service platform components aim for high usability and ease of use, component designers should put their emphasis on the interface design and overall user experience. Seamless user interaction and response can help with making service interfaces visually appealing, thus converting them into successful digital services. Since service interfaces and their incorporation into digital service platforms can seem trivial, it is equally important to support user interface design for all available capabilities and focus on reaching the widest possible range of service users.

3.8.1. Design Principles

In addition to technological research, design principles provide distilled, translatable design insights founded on real-world results that can be applied to DSP UX/UI development. In the 1980s, eight presentation rules were codified from decades of human-computer interaction work. These rules became heuristics that designers and evaluators could apply in various interface development and assessment situations. Strategic design principles were published that address the critical aspect of the digital experience's business value, favoring exceptional user experiences over minimal-cost development approaches.

This direct relation between the measures can be visually modeled using a matrix approach. However, it is cautioned that the relationships cannot be directly quantified, nor that the measures are the only ones relevant. The target zone usually should favor one of the two extremes, but the design process will eventually lead to points in different quadrants. The principle of least effort is one of the essential design and evaluation principles introduced within the usability-centered design movement. It emphasizes experience ease of use through design characteristics such as quick task completion, minimized errors with correct solutions easily recoverable, learnability, and low cognitive load. It also emphasizes business goals and the evolving state of the audience and audience relationships, not only the interface features.

The Digital Business principles and the least effort principle reflect the essential character design requirements: real-time operation like what is expected and natural, fitting behavioral expectations for short-term and longer-term goals; visibility and appeal, easily capturing and keeping the audience's attention; and task focus, helping instead of getting in the way.

3.8.2. User-Centered Design

User-Centered Design, commonly referred to as UCD is an established approach for designing end-user interactive systems and processes. It has been actively used in research and standardized in practice for many years. The arsenal of UCD methods is quite broad including, but not limited to, participatory design, design thinking, design-based research, ethnographic studies, cultural probes, storyboarding, personas, pluralistic testing, paper prototyping, and usability testing, among many others. These methods offer activities at every stage of UCD from concept generation to evaluation of prototypes including deployable products. The methodologies have their own goals and guidelines but generally adhere to the principle of actively involving end-users as participants throughout the design process.

UCD has become such an important part of interactive systems and service design that it is difficult to imagine designing such systems without applying these principles. UCD is also heavily emphasized in the designs of digital service platforms; users from different affiliated parties can come to the platform with different needs, backgrounds, and devices. Overly rigid or flawed designs can severely impact the experience of a large group of users, leading to contract disputes and non-utilization of the service platform. Examples include instability or slowness of a website or app for end-users, overly confusing tools for developers, and unintuitive User Experience for non-technical internal users.

For all these reasons we outline and implement a roadmap for UCD in the design of a digital service platform for a specific target group: challenges to the Internal Audit Governance process of Enterprises. We adopt a mix of methods from the UCD repertoire, utilizing channels such as interviews with both potential users and experts in the domain, heuristic analysis of competing platforms, and identifying personas in order to delineate the target audience. From there, we elaborate User Stories and Storyboards, impacting both our service platform's functionalities as well as its accessibility.

3.8.3. Accessibility Considerations

The internet is not just for people who can see and hear perfectly. Many people get to the web using methods that are quite different from the norm. For example, someone with a limited amount of movement may get to a website through a joystick that lets them select options on the screen. Someone who is blind may be using a speech synthesizer that announces each item on the page as they navigate through it. Someone who has limited vision may be using a magnifier that enlarges the text and images on the screen. As web designers, it is our job to ensure that the people using the site with these tools have the same experience as everyone else. This means that the site has to be accessible, in the broadest sense.

First of all, though, we should establish what accessibility means in the context of websites. Accessible means "capable of being reached" and "capable of being used." The first definition indicates that people can get to the site by type of browsing method, and the second says they can use the site in the same way the original designers intended. So, if someone is using a text-only browser, they have as good of an experience of the site as someone using a graphic browser. Many times, it seems like accessibility means more about what the web developers want to include, but it depends on the people and the methods of web browsing that they are using. After establishing this basis for accessibility, we can discuss some of the more technical areas of accessible design.

3.9. Conclusion

Drawing on numerous international case studies, this book delves into the design and deployment of digitized business transaction services which constitute a Digital Service Platform for modern, technologically advanced enterprises. Digital service platforms have evolved rapidly in the last two decades, driven primarily by new technologies and the demand for globally integrated business services. However, capitalizing on DSP capabilities is neither simple nor straightforward. Enabling digitized transactions via a DSP requires a clear understanding of business requirements, a deliberate design for integrated business services, and a sound technology implementation approach. Many businesses have thus far followed the traditional technology-driven enterprise applications approach that fails to realize promised benefits. This book presents a coherent and well-structured design framework for developing DSP capabilities for modern enterprises.

There are at least two key conclusions that can be drawn from the case studies presented in this book. First, the DSP framework delineated herein is able to capture the entire enterprise service component. This captures in a structured manner the underlying business requirements for enabling end-to-end digitally integrated business services, the transaction service design for integrated service delivery, the underlying enterprise services and their technology dimensions, and the technology landscape mandates required for the required DSP capabilities in specific industry sectors. Second, this comprehensive framework could be equally well used by both established business organizations as well as new organizations that seek to integrate their business services around a DSP. In both of these contexts, the consideration governing the selection of an appropriate framework design would be operationalizing a coherent utility service logic that delineates the nature of utility value to the customer.

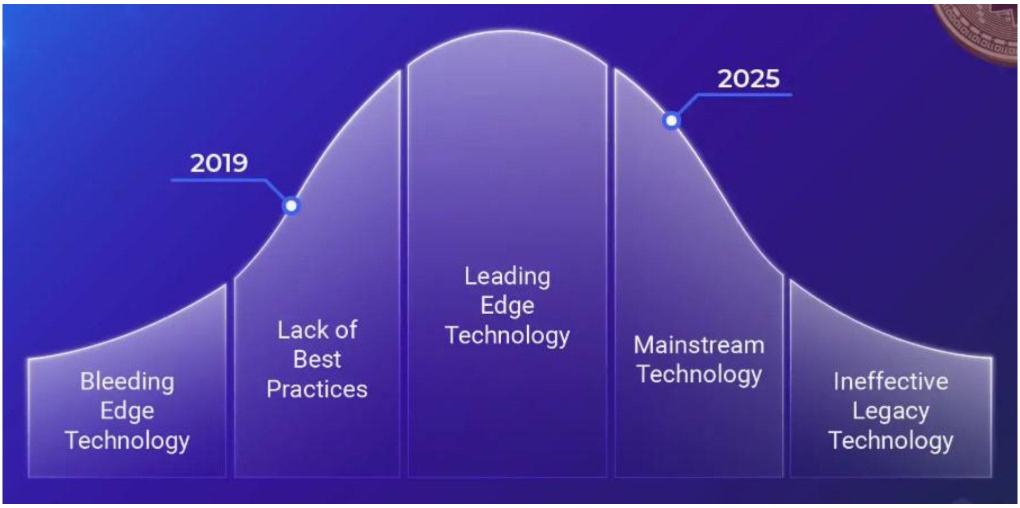

Fig 3 . 3 : Interoperability Impact Enterprise Adoption

3.9.1. Final Thoughts and Future Directions

For the last few decades, Enterprise Application Integration has been a challenging issue with mixed results in managing and creating value from enterprise systems. With the advent of granular and modular digital services and a new wave of cloud-based service platforms, the focus has shifted from EAI to Digital Service Orchestration: Digital Service ecosystems aim at integrating services from different providers for value co-creation via a loosely, yet coordinated approach. In this book, we proposed a Digital Service Platform as an architectural artifact that aims to facilitate the design of DSO systems. We then presented several building blocks, services, and APIs that can be customized and assembled to create a whole range of scalable and interoperable DSPs. We discussed orchestration patterns and design principles on how to reliably compose Services, easily assemble and promulgate DSPs from component Templates, and optimally exploit the Digital Services features of cacheability, idempotent state change, and asynchronous messaging. Moreover, we discussed monitoring tooling to check the health of a DSP service and its multiple versions.

We propose the following future directions. We started addressing low-latency long-running transactions with orchestrations combining APIs and Messaging. Furthermore, asynchronous messaging is critical in a multi-cloud environment where major clouds have no physical interconnection. Enterprise DSPs heavily rely on APIs, and the technology to design and monitor the lifecycle of our orchestration Templates needs to evolve to make customizable domain-specific orchestration design accessible to business domain experts. The DSP governs intensity in detecting changes in DSO orchestration via its event broker, as this “data pollution” can delay the actuation of the orchestration's regular processing. Finally, DSPs have the potential of also embedding DSP developer/interface services as extra components via DSP internal APIs.

References

Tiwana A., Konsynski B., Bush A.A. (2010). Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics. Information Systems Research, 21(4), 675–687. https://doi.org/10.1287/isre.1100.0323

de Reuver M., Sørensen C., Basole R.C. (2018). The Digital Platform: A Research Agenda. Journal of Information Technology, 33(2), 124–135. https://doi.org/10.1057/s41265-016-0033-3

Hein A., Schreieck M., Riasanow T., Setzke D.S., Wiesche M., Böhm M., Krcmar H. (2020). Digital Platform Ecosystems. Electronic Markets, 30, 87–98. https://doi.org/10.1007/s12525-019-00377-4

Baldwin C.Y., Woodard C.J. (2009). The Architecture of Platforms: A Unified View. In Gawer A. (Ed.), Platforms, Markets and Innovation, 19–44. https://doi.org/10.2139/ssrn.1283864

Ghazawneh A., Henfridsson O. (2013). Balancing Platform Control and External Contribution in Third-Party Development: The Boundary Resources Model. Information Systems Journal, 23(2), 173–192. https://doi.org/10.1111/j.1365-2575.2012.00406.x