Refining service blueprint for composable digital services

Osku Laurila
Master's degree Programme in Information Technology, Computer Engineering
Supervisor: Jerker Björkqvist
The Faculty of Science and Engineering
Åbo Akademi University
2023

Abstract

When the services have expanded to digital platforms and the number of concurrent actions has increased due to the use of the APIs and external integrations, the service blueprint framework has reached its limits and the ability to handle complex digital services is not enough any more to withstand digital service needs. This thesis focuses on the following questions: What is the current state of the service blueprint, are there viable options to satisfy customers' and service providers' needs, and can the service blueprint be used to efficiently visualise modern digital services? There are many other approaches and frameworks that are very good at explaining and visualising certain parts of digital services, such as Unified Modeling Language(UML) diagrams, system architecture diagrams, and user flow charts. None of these provide a good representation of the whole service and the interactions between the user and the technology. These diagrams and graphs are often too complex and too detailed for the stakeholders to understand. The service blueprint is a great alternative to these charts and diagrams especially for visualising traditional physical services, such as a mobile phone repair or a customers hotel visit, because it shows all the interactions between the user and the service. The aim of the thesis is to provide a good overview of how the services have changed their physical form during the digital transformation and whether the service blueprint keeps its position for digital services.

Keywords: Service blueprint, Composable, Packaged Business Capabilities, Application programming interface

List of Abbreviations and terms

API	Application programming interface
IHIP	Intangible, heterogeneous, inseparable, perishable
IaaS	Infrastructure as a Service
KPI	Key Performance Indicator
NFT	Non-Fungible Token
PaaS	Platform as a Service
PBC	Packaged Business Capabilities
ROI	Return on investment
SaaS	Software as a Service
UML	Unified Modeling Language

Contents

Introduction

If we look at how the world has changed over the last 50 years, the changes in the digital world are staggering. Digitalisation as a term has existed since 1971, when it first appeared in the *North American Review*. The most rapid changes in digital transformation took place between 1990 and 2000, when the digital channels reached customers and then between 2000 and 2015, when the customers were able to communicate with the companies in real time. This time was the period when social media and hand-held devices took over the industry [1].

The service blueprint is a good framework for visualising traditional physical services, such as a mobile phone repair or a customer's hotel visit, because it shows all the interactions between the user and the service. The aim of the thesis is to provide a good overview of how the services have changed their physical form during the digital transformation and whether the service blueprint keeps its position for modern digital services.

The structure of this thesis is divided into vague chapters where the first chapter helps the reader to understand what a service blueprint is and what are the differences between services and digital services. The paper explains the history and structure of a service blueprint and explores other alternative versions. The second chapter focuses on composability which is described in more detail. After introducing the framework and the technology used, the example case is described in detail to understand how a simple composable web service could be visualised using a service blueprint. This example case and its possible shortcomings will then be examined and improved. The output of this thesis will be a refined service blueprint and suggestions for future improvements and other possibilities.

Service blueprint

A service blueprint is a framework that is used to create a visual view of a service's components, interactions, and relationships between process steps that are essential for that particular service to function. Service blueprints act as an overview of a customer journey while showing underlying processes that are not visible for the customer and identifying the time spent between user actions [2].

Service blueprint was created to tackle poor service quality in real-world service situations, such as getting a haircut or having one's car repaired. A service blueprint provides a more detailed view of the processes and services that occur simultaneously.

It typically has five main components, physical evidence, customer actions, frontstage interactions, backstage interactions, and supporting processes. Each process step may have its own execution time from start to finish or between sequences of multiple actions. The example service blueprint in figure 3.2 shows a low-fidelity and simplified version of the digital service, where the customer opens the web page and performs a few simple actions in order to buy a bus ticket.

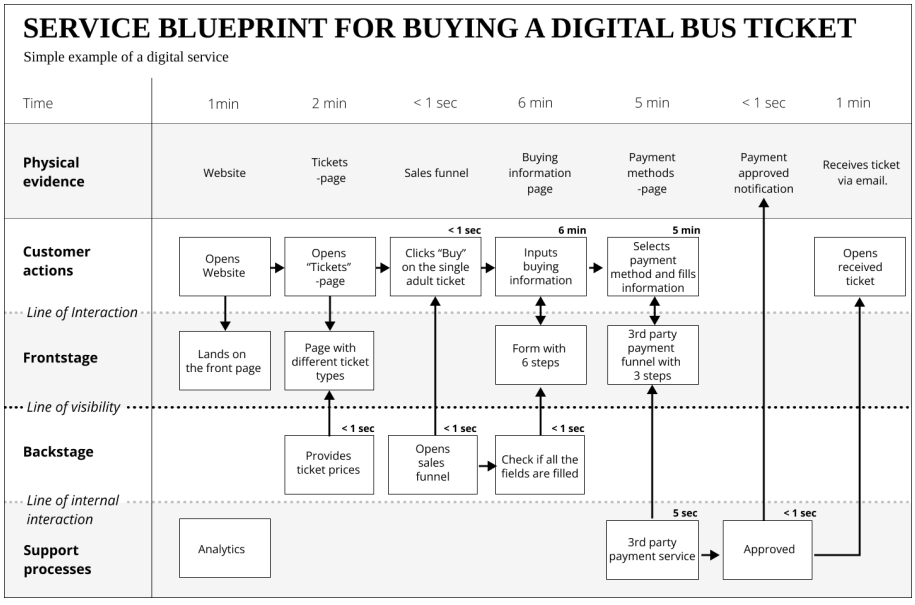

Figure 2.1: Simple service blueprint for buying a digital bus ticket

2.1 Products and services in digital environment - What is a service?

Before going into the details of the service blueprint, it is good to understand what a service actually is and how this paper defines the term, which has changed its meaning because of the digitalisation of the products and services.

This paper focuses mainly on the future of digital services and clarifies the boundary between a product and a service, which differs between digital and traditional services.

Ron Kaufman's (2020) [3] description of the service captures well how today's digital platforms perform. In his keynote in 2020 on "Uplifting Services", he described service as follows:

> *"Service is taking action to create value for someone else."*

In 2011, Parry, Newnes, and Huang [4] summarised this evolution of differences between services and products in their research:

> *"What we have seen is that the thinking has shifted from a pure service or pure product focus to a combination or product-service system (PSS). This recognises the offering of a combination of product and service for which greater revenue may be generated."* [4, p. 26]

Their research also describes that traditionally the goods and products are physical objects where their ownership rights can be established and basis consensus what makes a service as a service is by their features. This means that the services that are previously described as intangible, heterogeneous, inseparable and perishable are known as the IHIP model [4]. This IHIP model is often referred to as a legacy model and it is better suited to exploring different characteristics of a digital service than to describing it [5].

The service blueprint lays its foundations in the more traditional description of a service, where the IHIP model is strongly present, and the digitalisation of the services has drastically changed the state of service in these four decades since the first idea of the service blueprint was introduced in 1982 [6]. In the next section, we will take a brief look at the IHIP model and its different characteristics.

2.1.1 The IHIP model and digital services

Intangible

Intangible means something that has no physical dimensions, unlike tangible goods where the production of a tangible good consumes the material used to create the good itself Peter Hill (1999) [7]. For this reason, services are often described as intangible, because they can be a single action that someone or something does for the client or customer. For example, a customer might buy a massage as an intangible good known as a service, which gives value to the customer without a concrete physical form.

Currently, digital services have intangible characteristics, but digitalisation has shown us that digital services are very versatile and they are developing rapidly, so it is not an impossible idea to see fully tangible digital services in the future. Salminen (2014) [5] states that digital services such as websites share similarities with tangible products, so they are not completely intangible either.

In 1977, G. Shostack drew a figure (2.2) which showed how tangible dominant products shift gradually to intangible dominant where the fast-food outlets sit in the equilibrium between tangibility and intangibility.

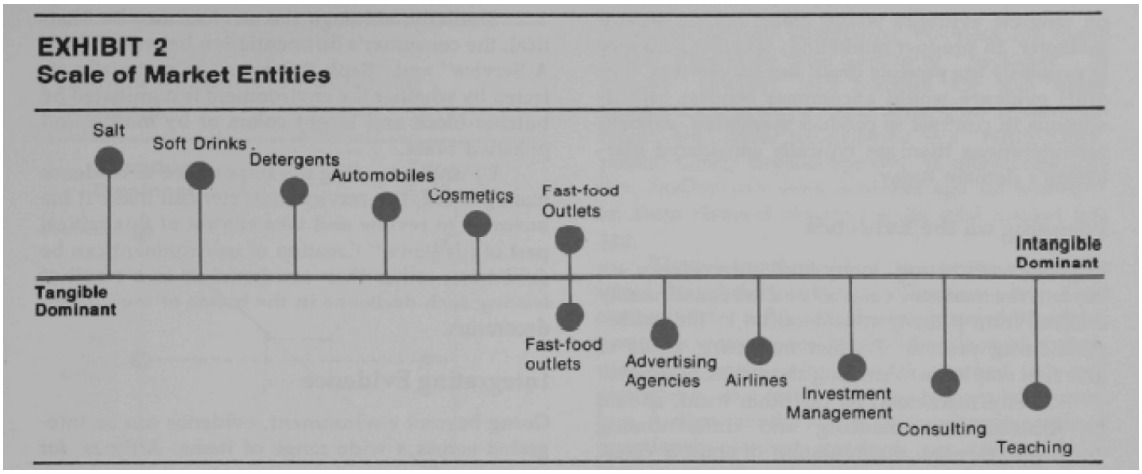

Figure 2.2: Tangible dominant to intangible dominant by G. Lynn Shostack (1977) [8]

Heterogeneous

When services are categorised as heterogeneous, they are based on the customer's or client's needs where the characteristics of the service are always different [4]. Heterogeneity applies to the services, but in digital services, the service provided can be homogeneous or heterogeneous. There may be a digital copy of a video game that is exactly the same for everyone who buys it.

Inseparable

Inseparability means that the goods are produced and consumed simultaneously, which gives the service its characteristic. Inseparability plays an important role in customer satisfaction, because the customer or a client already has some level of expectations of the service and these expectations have to be fulfilled during the production and consumption of the goods [9]. In digital services, consumption and production are usually separated from each other. The code for the digital service is usually produced before the customer pays for it, and consumption of the service can occur at any time after the purchase. Digital services can also be inseparable. A good example of an inseparable digital service is one that inherits a platform-based business model. This means that the digital service cannot be separated from the company itself, or the service will cease to exist. If you order a taxi through an application that gives you a multiple choice of different taxi companies and the provider of that service ceases to exist, then the layer between the customer and the actual service that is being purchased vanishes along with the company.

Perishable

When a service is described as perishable, it means the service perishes immediately during the delivery of the service. Parry, Newnes, and Huang (2011) [4] gave as an example of perishable service a seat on an airplane that perishes immediately when the delivery of the service ends. The customer has no ownership of the seat after the plane has landed and the customer has left the aircraft.

Since normally services are non-tradable and cannot be kept in stock for later consumption, the digital services can be non-perishable as well, since we already have a BlockChain technology that uses a cryptographic hash to verify ownership of the file with the generated timestamp. Having the technology to prove ownership creates the possibility of having completely non-perishable digital services and goods [10]. An example of a perishable digital service is the Non-Fungible Tokens, which uses BlockChain technology to create a completely unique token that cannot be divided or merged [11].

However, when we mix the intangibility of the digital service with the non-perishability characteristic of the NFTs, the real problems start to occur when the platform where your NFTs are stored is perishable [12].

A good way to describe this is to own any unique valuable physical item, like a painting that is stored in someone else's storage unit which can burn down at any time. This item in that unit cannot be sold without possessing the ownership of the verification code for that unique item, which makes it impossible to steal, but if the storage unit perishes, then your valuable vanishes with the platform and its value becomes worthless.

2.1.2 Services and digital services

After covering the formation of the service, we can dig deeper into digital services and how they have changed from traditional services to digital services. Table 2.1 shows that the only characteristic that is applied for the digital services and the service is intangibility. A similar table appeared in the *International Journal of Engineering and Management Research* (2017) [13] where Ali1 and Garg compared differences between physical goods and services. Salminen (2014) [5] did a digital service comparison to the IHIP model with a similar, but slightly shorter table.

Products, services and digital services compared			
Characteristics and features	Products	Services	Digital Services
Intangible	No	Yes	Yes
Heterogeneous	No	Yes	Can be both
Inseparable	No	Yes	Can be both
Perishable	No	Yes	Can be both
Can be kept in stock	Yes	No	Yes
Ownership can be transferred	Yes	No	Yes

Table 2.1: Table showing different characteristics between products, services and how the digital services compare against it.

2.2 History of a service blueprint

The history of a service blueprint goes back to 1977 when G. Lynn Shostack wrote an article about intangibility in products. In that article, she stated that there was a need for a framework that accommodates intangibility instead of denying it. She also explained how products and services are often separated, but both can contain tangible and intangible elements [8, p.74].

Five years later, in 1982, Shostack introduced the first form of a service blueprint. Figure 2.3 shows the first blueprint canvas that appeared in 1982.

A later appearance of this same blueprint was in 1984 in *Harvard Business Review* where Shostack introduced the service blueprint as a method to tackle problems of a poor service quality. She also described identifying the problems of a service often to be difficult, because when the customers are dissatisfied with the service, they tend to assume that they are having a non-expert treatment by the service provider. This leads to company management often blaming the individual employees of these service malfunctions without understanding the whole process of the service. Incompetence is not the only factor that affects the quality of the service, so the service blueprint was meant to correct the stiffness of the PERTT/GANTT charts which did not take into account customers' interaction with the service and the simultaneous actions during the process [14].

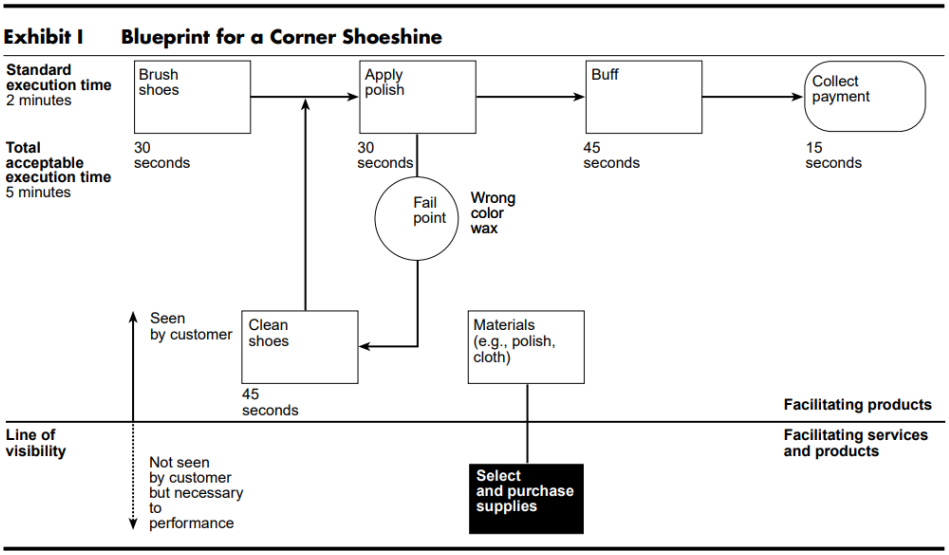

Figure 2.3: Example of one of the first service blueprints by G. Lynn Shostack from *European Journal of Marketing* - How to Design a Service [6].

2.3 Structure of a Service blueprint

There are multiple variations and existing service blueprints based on Shostack's original version. Shostack's version was developed further by Kingman-Brundage in 1988 and this version introduced a more detailed version with some adjustments, such as drawing a line of interaction between the customer and service provider which are shown as front stage and backstage actions.[15]

However, in this section, we focus on a very basic version of the service blueprint that includes all the key features from the original version to fully understand the structure and the level of detail introduced in 1982. This version is still very commonly used because of its simplicity and efficiency. Figure 2.4 includes one of the variations that share many similarities with Shostack's original version, which also works as a base in this section, is from Nielsen Norman Group posted by Gibbons [2].

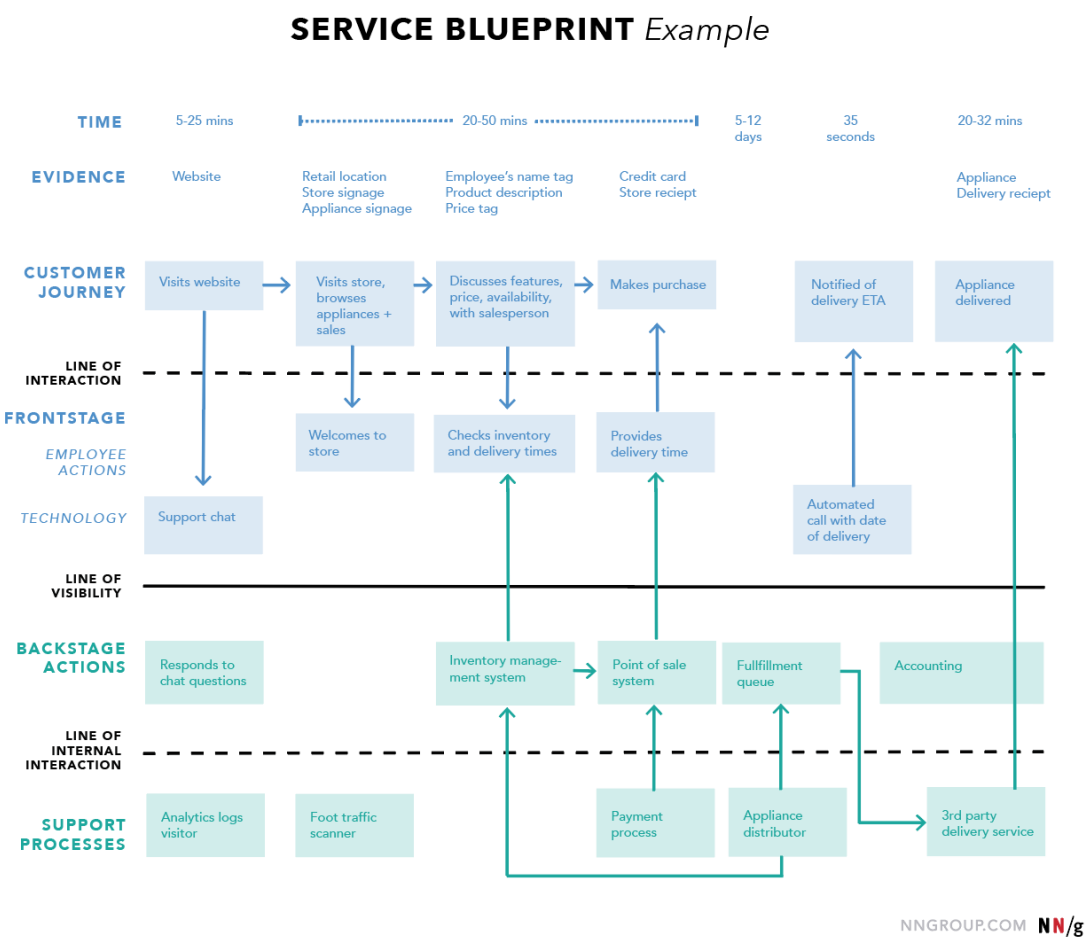

Figure 2.4: Service blueprint template by Nielsen Norman Group

2.3.1 Time

Time is one of the key features in the service blueprint because time also has a major role in overall user experience and it has a drastic effect on people's satisfaction with the services. Shorter waiting times in services tend to have positive effects on customer satisfaction. In the early 21st century, B. D. Weinberg (2000) [16] did an experiment where one-half of the sample users had to wait 5 seconds before the specific web page loaded and the other half of the sample had to wait 10 seconds. After that, the users had to give a rating based on the feeling of the web page's overall quality without them knowing the experiment was about the loading times. People who were randomly given the 10-second waiting time, rated the homepage quality to be lower than the other group.

Tracking the execution time of each individual action helps to calculate the total time of each simultaneously driven process between the steps in the service. The total time of the process may not be efficient enough to satisfy customer needs and it might have a negative effect on user experience, which leads to dissatisfied customers. In the highlighted figure 2.5, each column shows the average completion time before the user moves to the next step. This can reveal all possible underlying issues in the process. In the example, there might be an excessively long waiting time before the user can continue to the steps that create actual value for the customer.

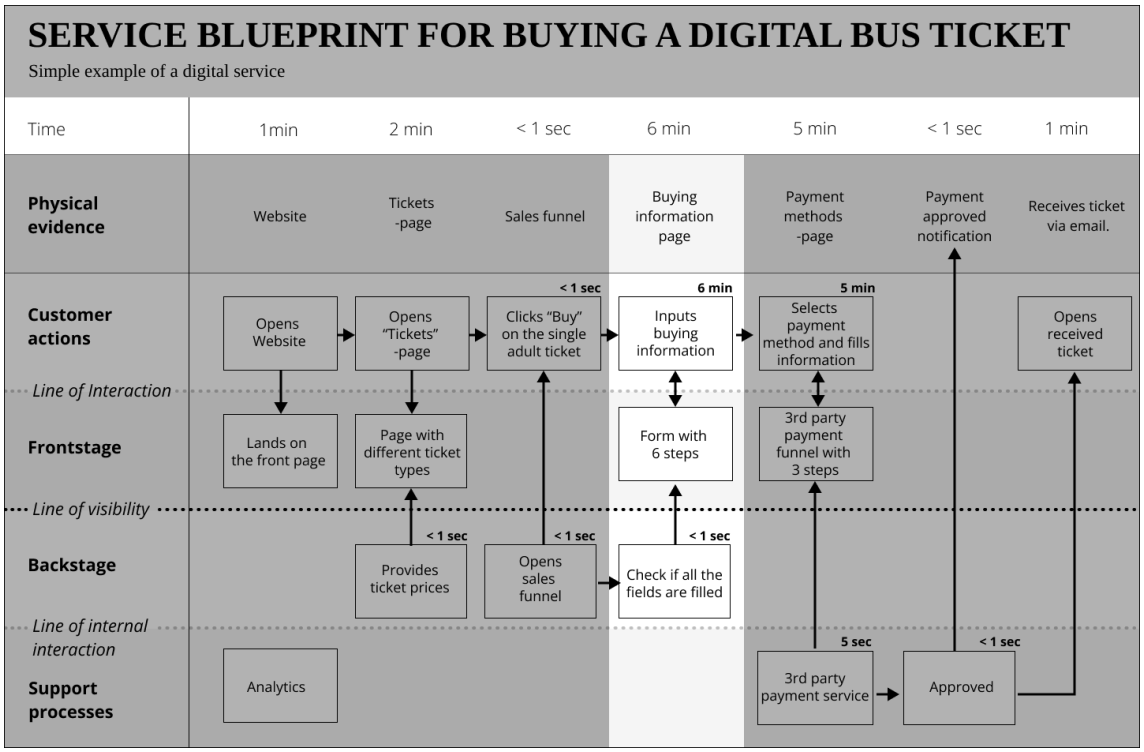

Figure 2.5: Time row and column of actions highlighted

2.3.2 Physical evidence

The physical evidence section is usually the second row of the service blueprint, but it is also the last line to be filled in, as it ties all the other actions to a physical or digital location. Physical evidence can also be something tangible or it can be a physical location where the specific actions in the current column take place. It can also be something that the user sees or receives in the digital environment, such as a confirmation email welcoming them to the service after they have created their account. It might also be a click on a marketing campaign on another website that brings the user to the digital service or a website where the customer actions occur. In figure 2.6, the physical evidence row is divided into seven columns. The first column is the website itself and the last column is when the user receives the bus ticket. A similar case of the physical evidence could be the ticketing interface at the bus station, or it could be the mobile phone interface as a physical evidence in obscure locations that could have an impact on how and where the user interacts with the service.

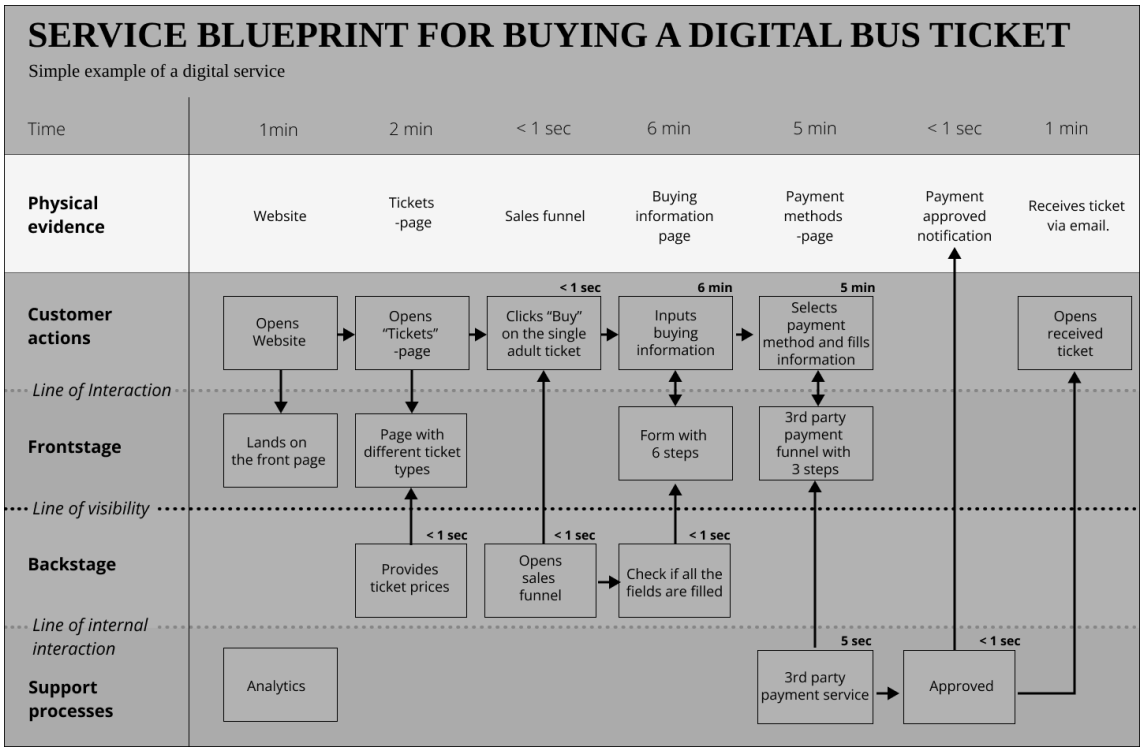

Figure 2.6: Highlighted Physical evidence row

2.3.3 Customers actions

The customer actions row contains all the interactions the customer needs to perform in order to achieve the goal. In this example, the user goal is to purchase a bus ticket. In this example, the user goal is to buy a bus ticket. In the highlighted figure 2.7, each rectangle is a customer action that takes a different amount of time to complete. The example case shows a user action where the user needs to fill in the form with the purchase information and, based on the data collected from the website analytics data, this gives an average time of 6 minutes to complete this specific action. Gibbons (2017) [2] describes that the customer actions include the steps that are commonly found in the customer journey map, which is a widely used method to understand how the customer interacts with the service. The service blueprint acts as an extension to the customer journey map, rather than as a replacement. [17]

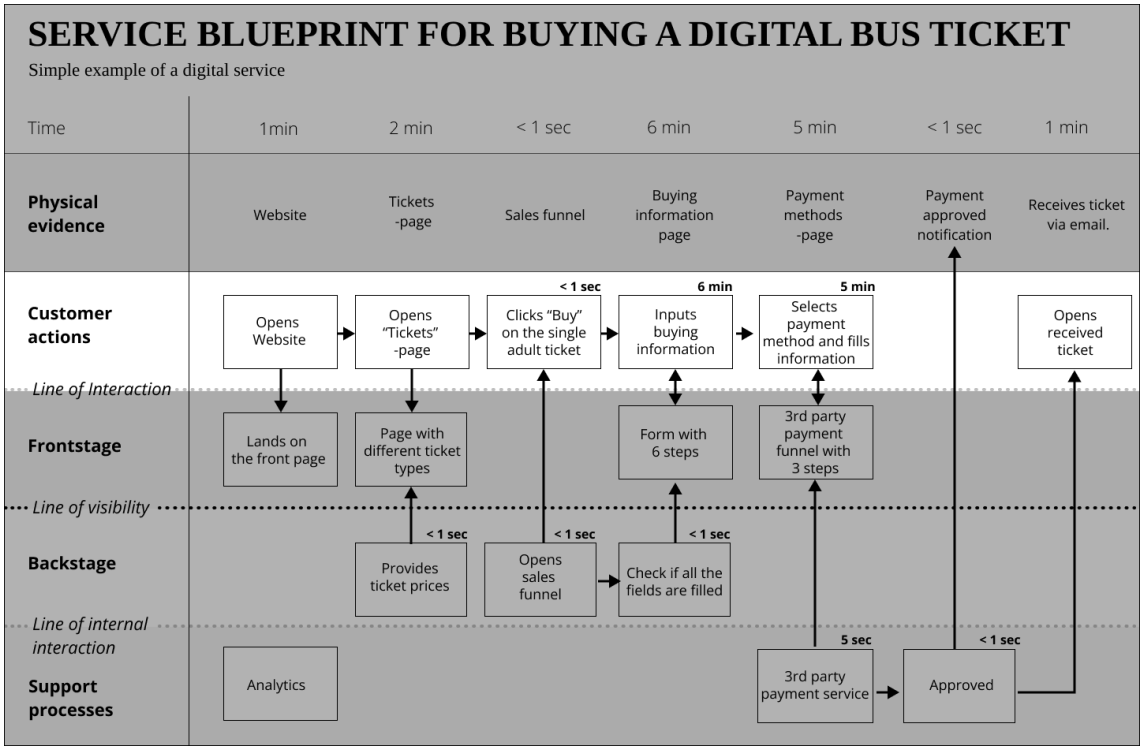

Figure 2.7: Highlighted customer actions row

2.3.3.1 Line of interaction and Line of visibility

Between the customer actions and the front stage, there is a line of interaction (2.8). This line represents the interaction between the customer and the service or technology.

In digital services, the line of interaction is crossed when the customer needs to use the user interface in order to complete certain actions and processes to move towards the goal. The line of interaction is also crossed when the customer interacts with the service personnel, for example, when checking into a hotel at the reception desk. Interaction between service personnel can also occur in digital services. A good example of a digital service that combines face-to-face interaction with digital service is the mandatory age check at the self-checkout in a grocery store, where the customer scans an age-restricted item that must be accepted by the clerk before the customer can continue interacting with the digital service itself.

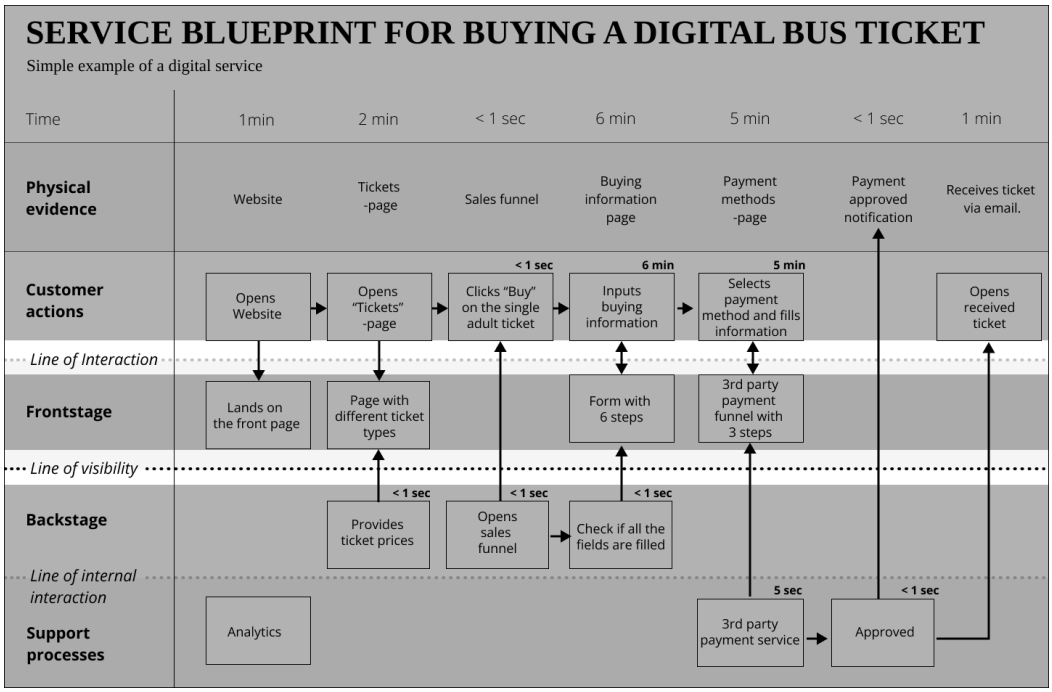

Figure 2.8: Line of interaction

All the actions that are above the line of visibility are visible for the customer. These elements are anything the customer can see, touch, use or interact with. It could be an encounter with the service provider's staff, a digital user interface or physical buttons. These have a direct impact on how the customer feels about the service. Below the line of visibility is everything that is hidden from the customer. Anything that does not work properly here can have a negative impact on the customer. Actions below the line of visibility can be very difficult to identify, because the customer or the service provider may not know what is really causing the problem. A key function of Service Blueprints is

to provide this information and help identify the underlying issues that are the root cause of the problem. A good example of a hard-to-identify problem is a 3rd party service that is maintained by a third-party company. Users may be redirected to another platform to complete certain actions, such as the payment funnel. Users experience the entire service as a whole and because of the line of visibility, they do not know the problems which are caused by the third-party company and that these problems can lead to negative customer feedback.

2.3.4 Front stage interactions

Front stage interactions, or sometimes referred to as a front office interactions (2.9), are the immediate actions that occur directly with the customer.These front stage interactions can occur between customers and service agents, or between customers and digital interfaces such as web applications or self-service checkouts that involve customised physical interfaces with digital applications. Even if there is no human-to-human interaction, the front stage is still a very important step, because it is above the line of visibility. Everything that happens in this front stage has a huge impact on how the customer experiences the service. A bad user experience with a digital user interface can be compared to experience of a bad service experience with a service agent.

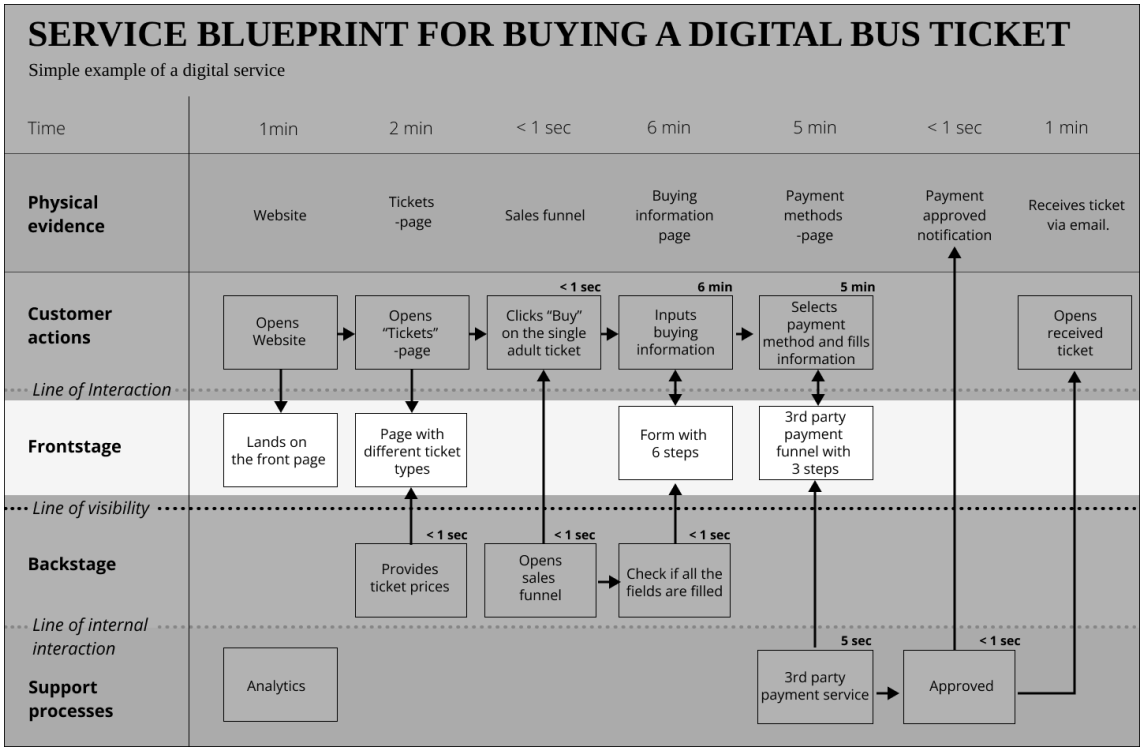

Figure 2.9: Front stage interactions

2.3.5 Backstage interactions

Below the line of visibility (2.12) are the backstage actions. Everything below the line of visibility happens out of sight of the customer. These actions can have a significant impact on the quality of the service [18]. Backstage actions can also include physical actions that need to be performed by the service agent. A good example of backstage actions involving physical actions that may have led to a poor experience with the service is food delivery services and platforms that rely on human interaction. If the order has to be accepted before the user can receive the delivery notification, the delivery person forgets to mark the order as delivered. This can lead to dissatisfaction with the service, followed by leaving negative feedback for the restaurant, if food gets cold during delivery, or an employee in a grocery store may fail to put goods on the shelves properly, leading to customer dissatisfaction with the store.

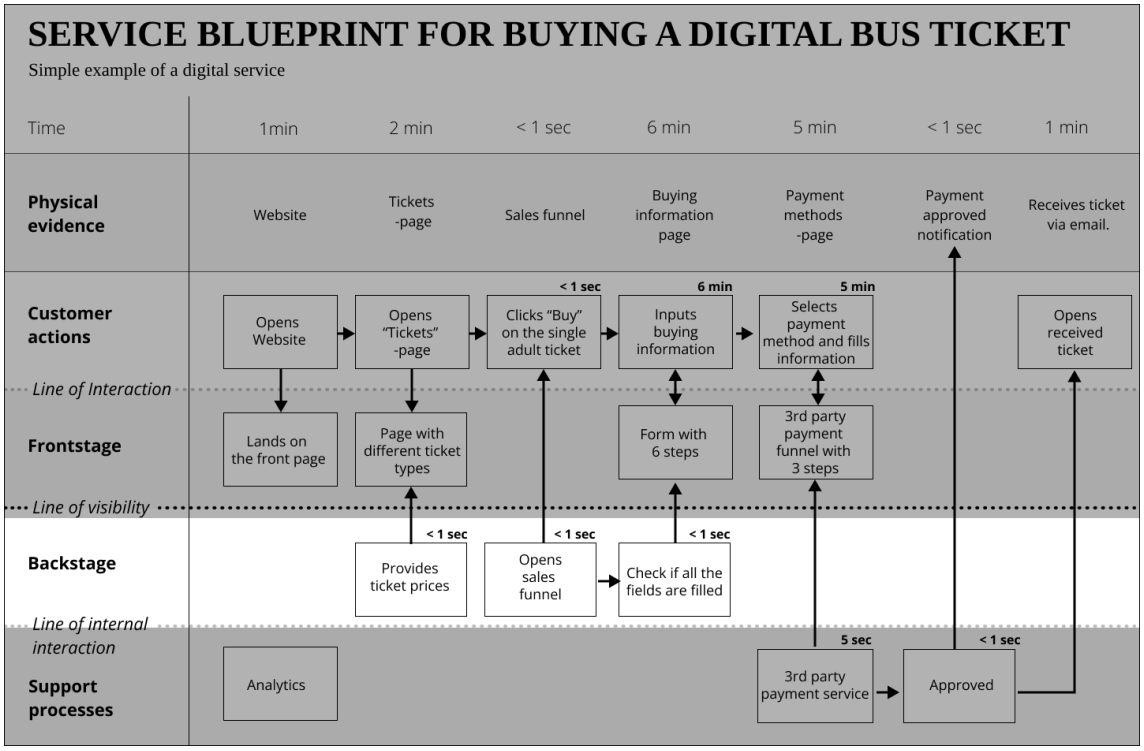

Figure 2.10: Example of a backstage interactions

2.3.6 Line of internal interaction and Support processes

The internal interaction line separates the backstage actions from the support processes. Anything below this line may contain various functionalities provided by the 3rd party service. These support processes are usually crucial for the whole system to work. This could be a payment funnel allowing the user to use any modern payment method. In traditional physical services, these support processes may include the delivery of supplies from the manufacturer/factory or stocking the shelves with physical goods before the store opens.

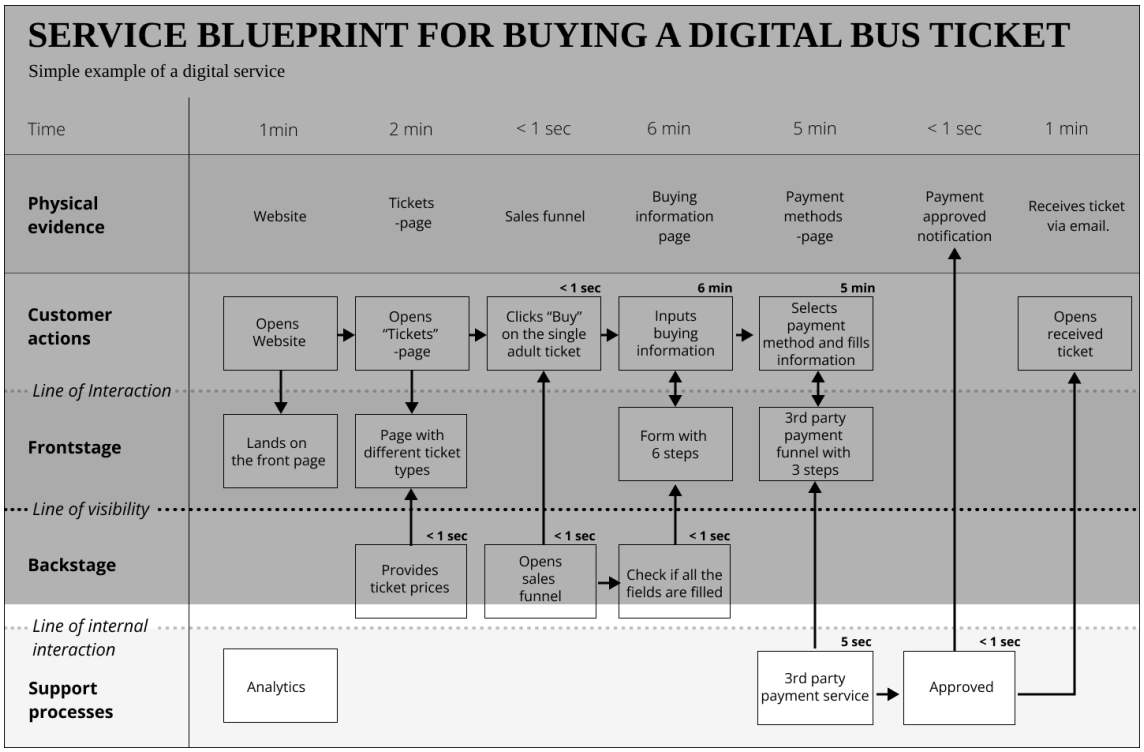

Figure 2.11: Example of a Line of internal interactions and support processes

2.4 Other Service blueprints

In 2013, Haugen M. [15] wrote a paper about the future potential of the service blueprints and how they have developed. This paper included a figure 2.12 of a model of how the service blueprint has evolved from 1977 to 2013. This informative presentation includes several variations of the service blueprint variations over this period. The paper included the more detailed version of a service blueprint by Polaine et al. However, the timeline is still rather incomplete as it is almost 10 years since the paper was published.

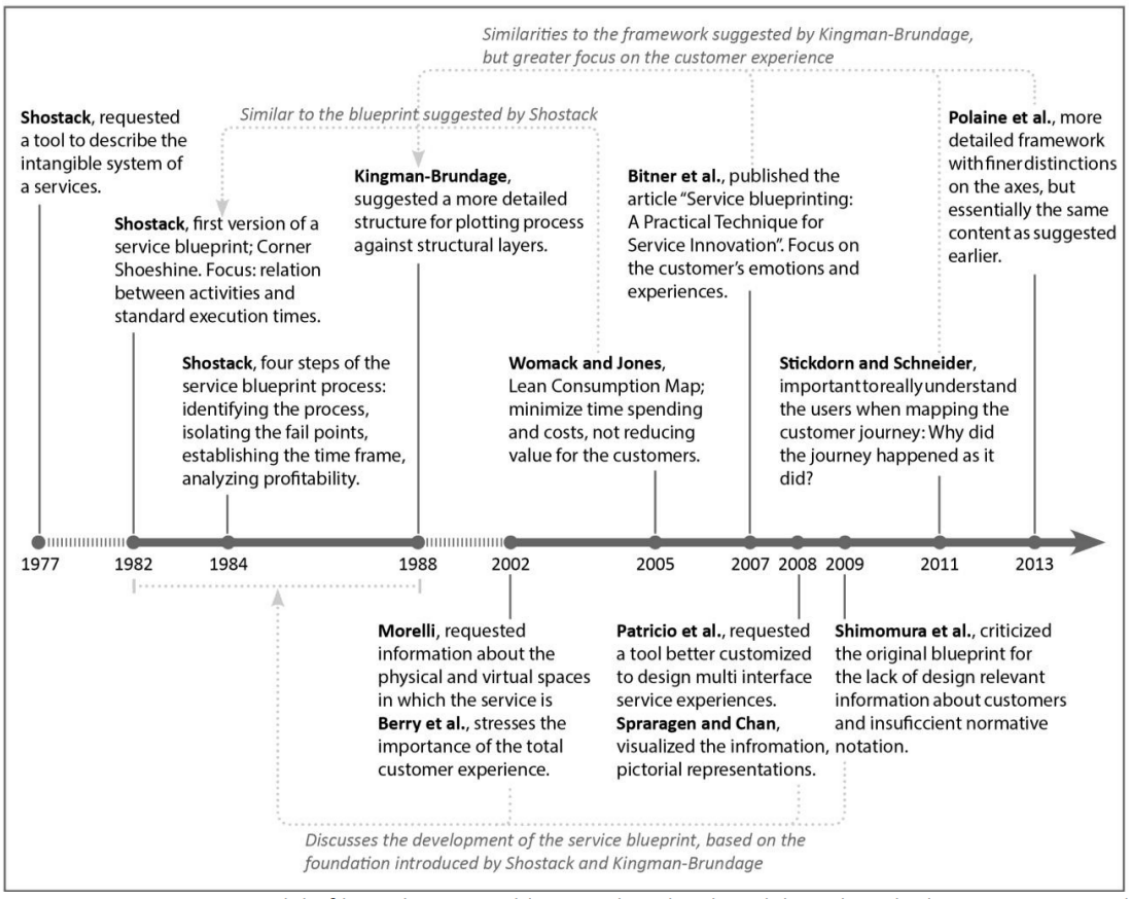

Figure 2.12: Service blueprint development between 1977 and 2013. Haugen (2013) [15].

2.4.1 Project Wall

Wipro Digital has created an extension to the service blueprint template which is meant to be used through the collaboration platform called Miro. Project Wall was created to tackle missing collaboration when creating traditional service blueprints and it includes many innovative and great approaches to rethink the classic service blueprint. Project Wall contains elements such as customer journeys, business metrics, processes, DevOps roadmap, timelines, architecture, and interfaces. Figure 2.13 shows the Project Wall and its high level of details [19]. A major advantage of the project wall is a good hierarchy with the swimlane structure. It includes stages and actions at the top, which gives a clear view of the user actions, and everything below that is tied to those top actions.

Project Wall is a technical approach rather than an easy-to-understand overview for the stakeholders. It is a tool designed to be used by everyone in the development team. There are complex data relationships, layers of functionality and very detailed data that works more like a backlog with subtasks for the developers.

Project Wall is not intended to replace the service blueprint, but is more like a backlog or Kanban board that has taken best practices from the service blueprint to provide a detailed view of the web service for the teams to use. Project Wall also includes API and Service Layer to work better with digital services.

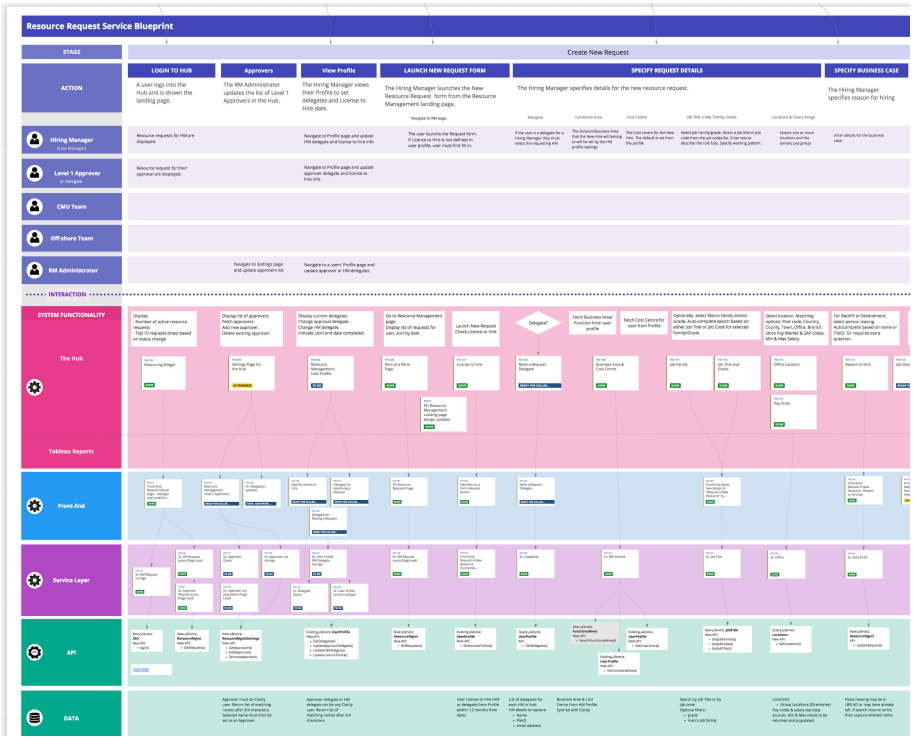

Figure 2.13: Example of a Project Wall [15].

Composability

Composability is mentioned in different contexts, such as composable business, composable architecture, composable infrastructure, composable commerce, composable enterprise, or composable web services. It is sometimes referred to as modular or adaptive. While there are different applications where the term composability can be used, it usually means the same thing. Composability is a method or way of building applications, websites, e-commerce, or other web services piece by piece. This generally results in faster development, lower costs, and a more robust code base. These different composable pieces, or more commonly referred to as building blocks or packaged business capabilities (PBCs), are like Lego pieces that allow users to pick and choose the best technology that fits their business and provides the most value for their money. These building blocks are designed to be interchangeable, reusable, and modular. [20]).

3.1 Benefits of building composable digital services

The number of professional developers in 2019 was 23.9 million, which means that the current percentage of the total population of the Earth who were professional developers at that time was only 0.3 percent [**evansdata**]. This creates a strong dependency between the users who do not know how to code or develop web services and the professional developers. As the IT industry continues to grow and the demand for digital services may eventually exceed the number of professional developers to meet the market demand, this means that there is a great need for low-code or no-code solutions to fill this imbalance in demand.

3.1.1 What is low-code and no-code?

When building web services, using these composable building blocks transforms the traditional slow and expensive development process into low-code and no-code solutions. Developers do not need to focus their skills and time on re-creating the same solutions and basic components over and over again [22]. This frees up development resources from these repetitive tasks to focus more on the core business of the service.

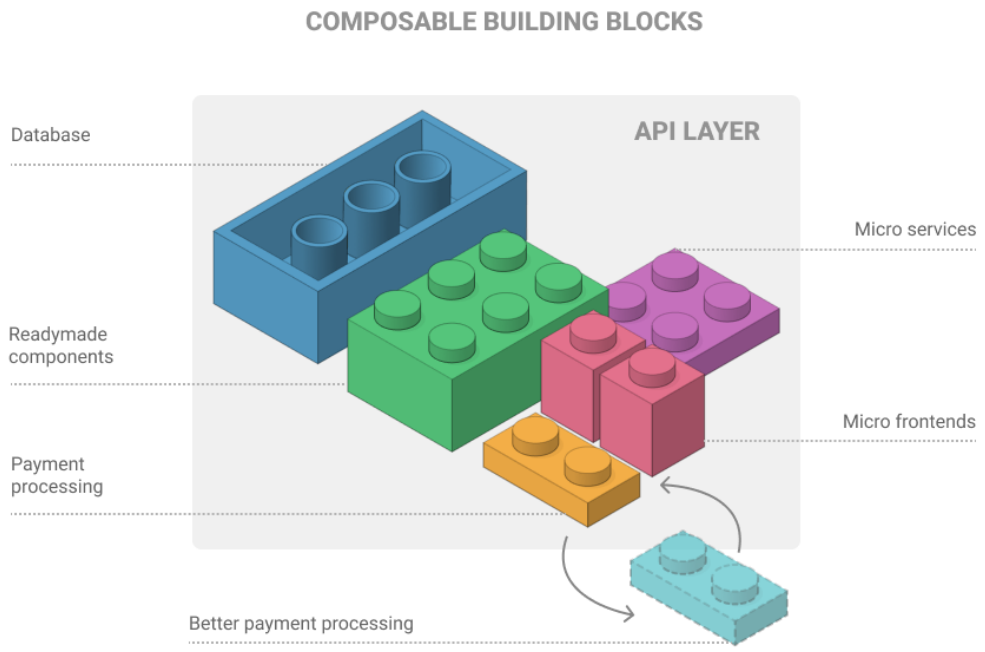

Figure 3.1: Example visualisation of composable building blocks. Original image San, A. [21]

Word composability is receiving more and more attention and composability is predicted to be the future of the business [23]. According to Malinen (2022)[22], we are currently living in the era of Adaptable Business.

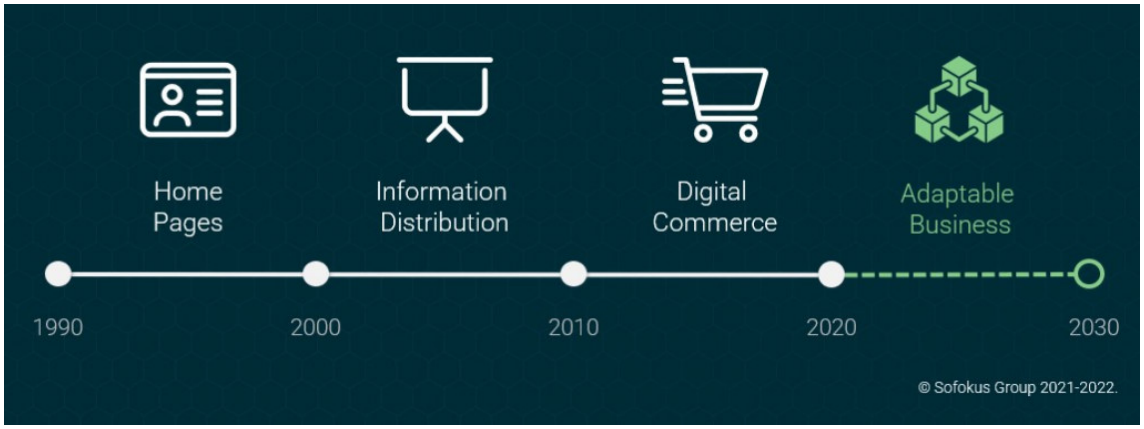

Figure 3.2: Visualisation of a different eras of the internet.[24]

3.1.2 Packaged Business Capabilities (PBCs)

Packaged Business Capabilities are small applications. These applications may or may not have a user interface. Components within these PBCs may include various user interactions. Components without a user interface are called "headless" and they provide only a portion of the main functionality that is used by other components. This allows the user interface to be created in-house as a separate PBC. PBCs are one part of custom composable web services. These small software packages can be accessed through the PBCs API for integration [25]. PBCs are called packages because they can vary in size. These packages contain a collection of one or more microservices (Figure: 3.3). PBCs are packages that are used when creating composable web services that are ready for the market and end-user consumption. PBCs are considered too large if any of their parts are replaced by another third-party microservice. Then the original PBC should be split into multiple PBCs for user consumption. The value of PBCs comes from the single vendor point of contact. Without a PBC, the user would have to interact with each of the microservices included in the PBC separately, and they could all be from different service providers, as owners of the service typically want to build their composable web service with "best-of-the-breed" solutions [26].

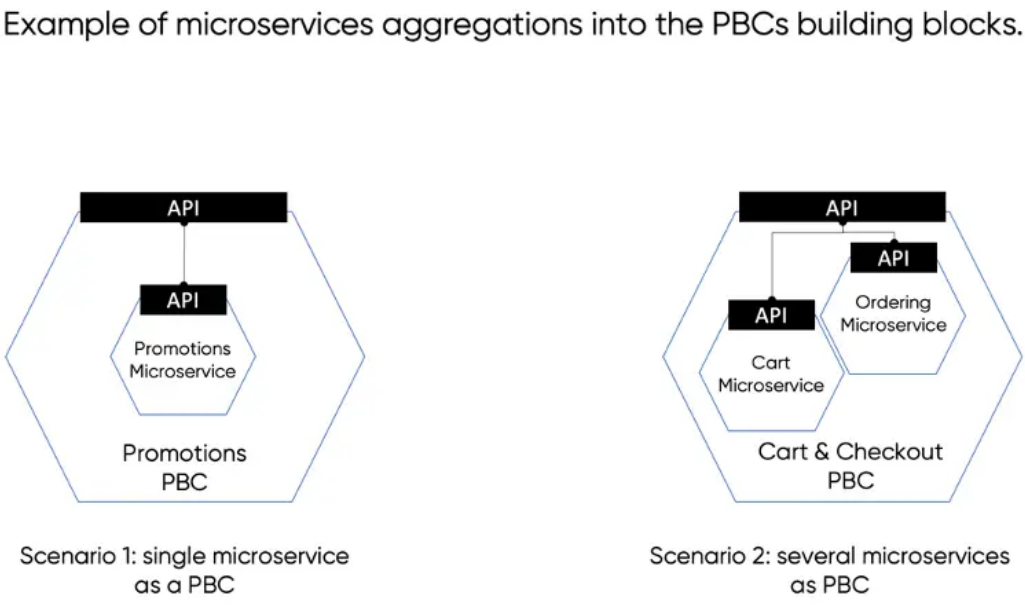

Figure 3.3: Example of single and collection of microservices as a pbc by Elasticpath (2021) [26]

.

3.1.3 Scaling and Composing experiences with PBCs

Having covered the basics of Packaged Building Capabilities (PBCs), it is good to dig a little bit deeper into how they can be used to build composable web services. In 2021, Mike Lowndes from Gartner [27] spoke in a webinar about the rise of composable commerce. PBCs can be built in three different ways. Either all the microservices included are third-party built by a vendor or a partner. This PBC is built entirely outside of the organisation and is referred to as a "third-party PBC". The second way is the fully in-house developed "Custom-built PBC" where all the services included in the PBC are built by the organisation's team with no third-party solutions involved. The third version is referred to as a "hybrid PBC" which is a mix of third-party solutions and custom-built services (Figure: 3.4).

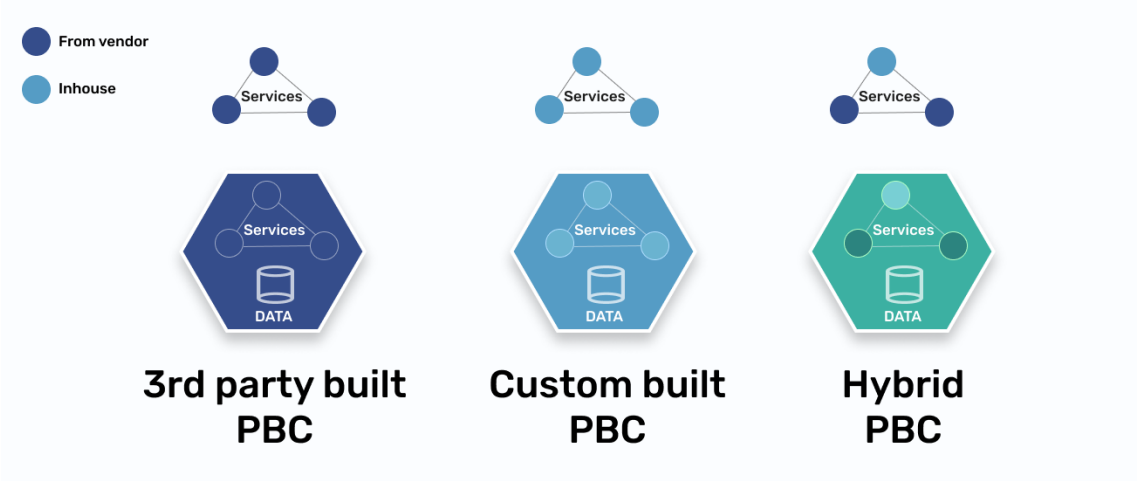

Figure 3.4: Different ways to build Packaged Business Capabilities (Original image Gartner (2021) [25]).

These PBCs are self-contained packages that can have their own user interface or provide event and API data to other composable user interfaces to create different composable experiences (Figure: 3.5). These PBCs can be combined with multiple other PBCs to create a "Composed Application Suite" with custom or third-party user interfaces just as with standalone PBCs but in this solution, all the PBCs and services communicate with each other to create a seamless user experience using multiple different PBCs (Figure: 3.5). These application suites can be packed even further to create an "Enterprise PBC/API Portfolio" (Figure: 3.5). These Enterprise PBCs can contain multiple composed application suites; if these composed application suites are made of multiple PBCs, then the API portfolio can contain hundreds of PBCs. For enterprise solutions, all these PBCs can be used to combine different user experiences for users, by selecting only the building blocks needed.

Currently, the problem is that there are rather few PBC management solutions that can assemble or compose seamless user interfaces and flows with little or no code. In figure 3.6, these API portfolios are used by these management/orchestration applications to compose functional composed web services. Users of these management/orchestration applications can be anyone with or without previous programming experience or it can be a customer who wants to create their application without writing any lines of code. It could also be a service designer or even a salesperson who creates working solutions for their client in a fraction of the time compared to custom-built solutions with developers.

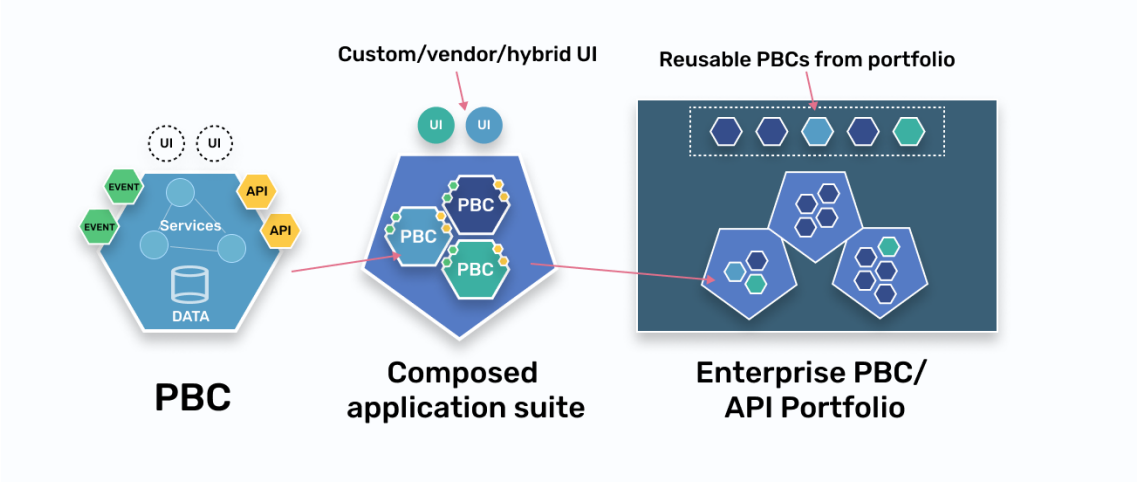

Figure 3.5: Example of how PBCs are combined from single PBC to API Portfolio. Original image Gartner 2021 [25]

After selecting several different PBCs to compose a web service, the user interface (UI) is created based on the selection and modifiability of the PBCs. For example, one of these PBCs may be a third-party payment service that has its own UI. When the end users add items to the shopping cart through the custom UI, they must click through the sales funnel until they reach the payment step. Users are then redirected to the payment funnel which can have a different UI than the previous funnel. After payment, the users are moved back to the custom UI or even to another third-party UI. These composed web services can be easily modified or changed by simply swapping or modifying the PBCs or even combined with another functionality from the API portfolio.

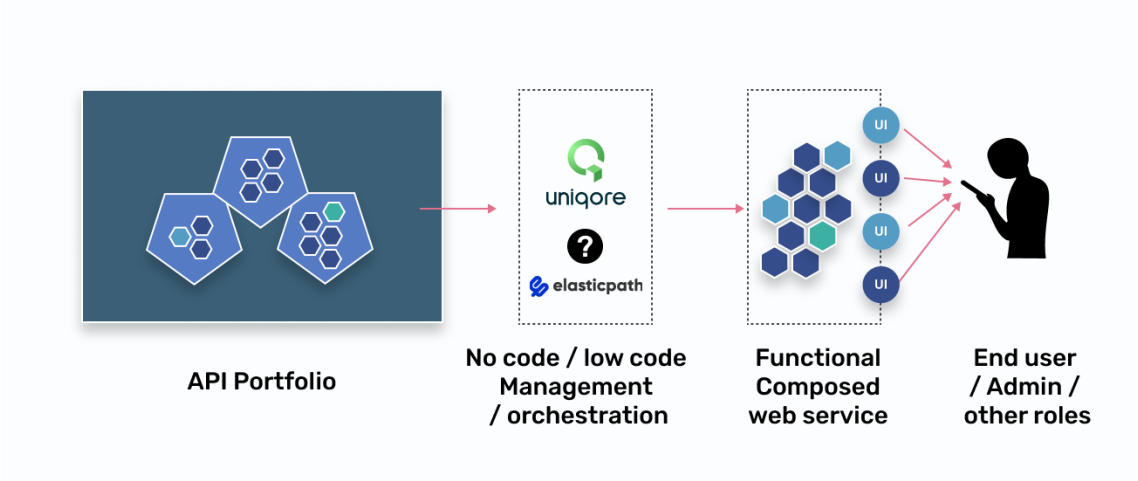

Figure 3.6: Figure to explain how you build composable web services

3.2 Application programming interface (API)

Application Programming Interface is a set of protocols, standards, and tools that allow different software applications to communicate with each other. The language between these two applications can be independent, allowing data from other applications to be implemented for various uses without compromising the user interface of the application [28]. As the use of APIs in individual applications continues to grow, they are becoming more complex and difficult to integrate. This has led companies to develop API management and integration technologies to make it easier to integrate their APIs by breaking their applications into smaller packages [25]. APIs in composable Web services means seamless integration between APIs using these PBCs, so that each building block can be easily swapped, removed, or extended by users who are not professional developers.

Service blueprint case study: Coffee bean subscription

To refine a service blueprint we need to have an example case to explore further. This chapter focuses on a more detailed service blueprint with an example case that represents a possible real-world composable solution. There is never a perfect or correct solution to what a final service blueprint should look like, and the final result will depend on the service being visualised and the amount of data and information being added to the service blueprint. For digital services in particular, the amount of concurrent data that needs to be presented in a single column can be significant. The example service blueprint is broken down into smaller parts to provide a clearer and more detailed view of different use cases that require refinement. The full-size service blueprint can be enormous, and even if the example is kept fairly simple, it still contains a large amount of repetitive data that is not worth looking at multiple times if it is covered in the previous section. The full-size service blueprint of the case (Figure: 5.1) is described in detail in the next chapter.

4.1 Case: Coffee bean subscription

An example case visualised with a service blueprint is a subscription for coffee beans. The reason for using the recurring subscription fee as an example is the simplicity of the process, and the amount of data and customer actions is minimal. There are a few key steps that take place between the service and the user. The visualisation of the service starts with the admin panel, where the owner or the administrator can create and manage all products, orders, and stock availability. This helps to understand the big picture and what other services need to be implemented outside of the customer journey. Once the items to be sold on the site have been created, the customer journey starts from the login page to the product selection and continues to the sales funnel and the stage where the successful payment has been processed. Once the subscription is placed, the items are collected, packed, and shipped to the customer. While the subscription is active, the customer is billed. The subscription will be fulfilled each month until the customer cancels

the subscription. When the subscription is cancelled, the remarketing campaign begins for the old customer to bring them back with discounts and other targeted emails.

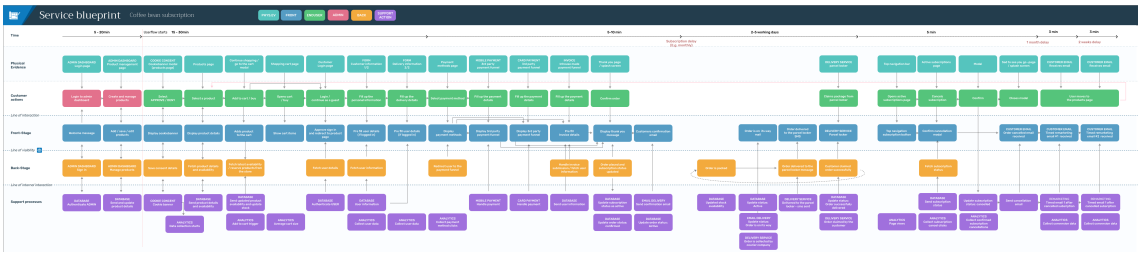

Figure 4.1: Service Blueprint of a coffee bean subscription

4.1.1 Service blueprint: Admin adding products to the backend

Here we look at the first two columns of the service blueprint which contain a small number of admin control panel actions (Figure: 4.2).

Normally, admin actions are not added to the same flow as the customer actions, but in this example, it is good to have some understanding of how many third-party integrations the service might need before the customer journey starts. The admin section is not very interesting to follow on a larger scale, as it does not have a direct impact on customers' feelings about the service and usually does not contain any information that would generate better conversion. External services may include monthly fees to keep the service running, which may be important information for the stakeholders.

The administration part is marked in red to distinguish it from the green customer actions. Before the customer can view products and buy a coffee subscription, the products need to be managed somewhere. In this case, the admin logs in to the third-party admin control panel such as JetAdmin, Retool, or something similar that is connected to the backend such as Firebase or Airtable. Even before any customer actions have taken place, the digital service has already connected to two external APIs that will handle the data for the actual digital service. In the control panel, the admin can control all the product details, create new products, and add images and write different descriptions for the products. The estimated time for the admin to create new or manage existing products is usually 5-20 minutes.

The physical evidence row contains the page that is displayed to the user, which is built using completely separate third-party software. The customer action row contains separate actions that the admin is supposed to do with the control panel.

Frontstage may contain a welcome message or some other information that is displayed to the admin after login. Backstage handles the customer actions and support

processes, including the entire backend, which contains all the data stored from the customer and admin actions. Managing the products with a third-party admin panel is not mandatory, but the user may need limited access to the data, in which case these third-party dashboards may come in handy.

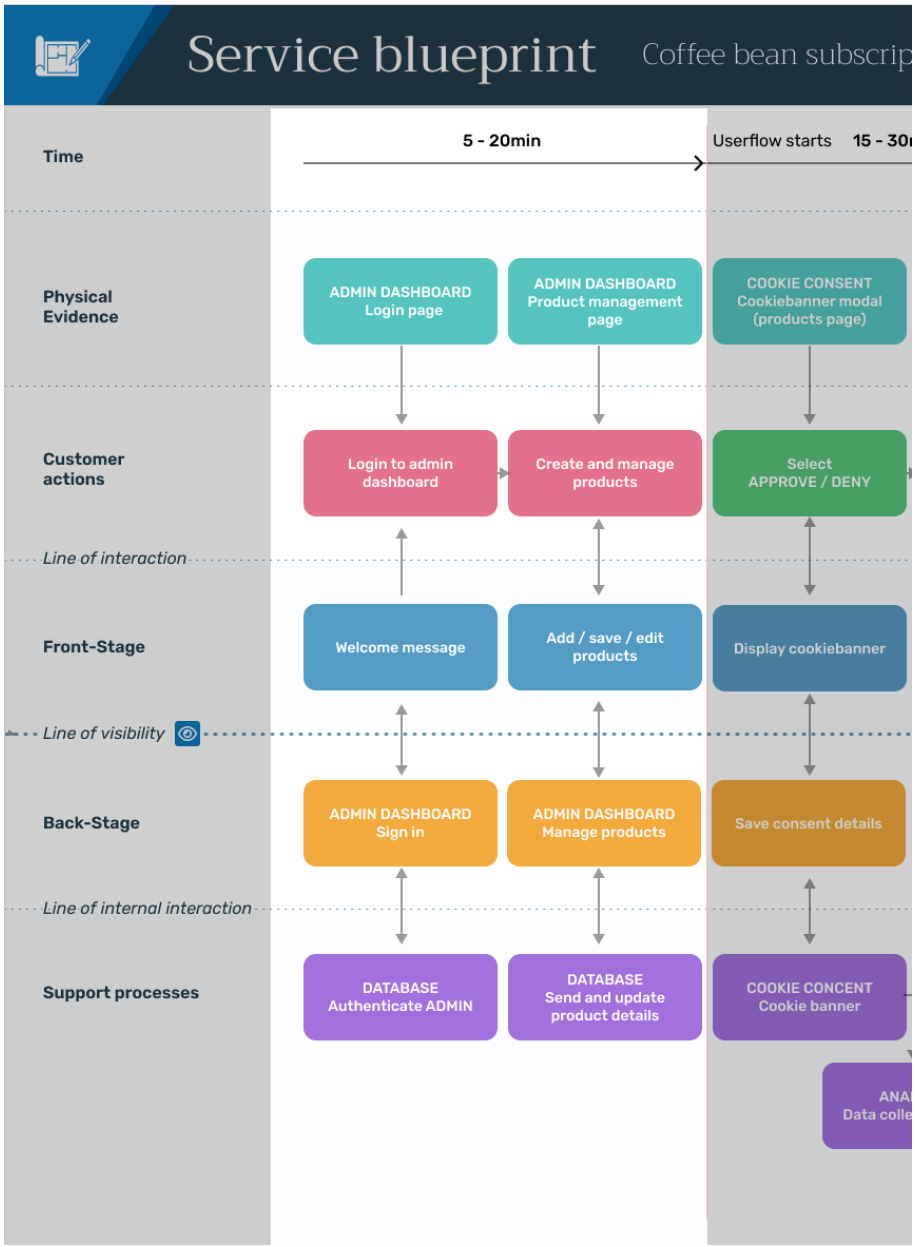

Figure 4.2: Highlighted view of an administrators actions

4.1.2 Service blueprint: Phase 1 - Customer selects and adds the product to the shopping cart

The first actual customer phase of the coffee bean subscription in this service blueprint is the customer actions before the payment funnel (Figure: 4.3). This phase contains all the actions that the user has to complete before moving on to the payment funnel. Usually, this is considered the most important phase, because the highest customer drop-out rate occurs after the customer has added the product to the cart [29]. This phase contains all the information from the user's first contact with the service, a decision on which product to buy, and all the steps involved in moving towards the payment funnel.

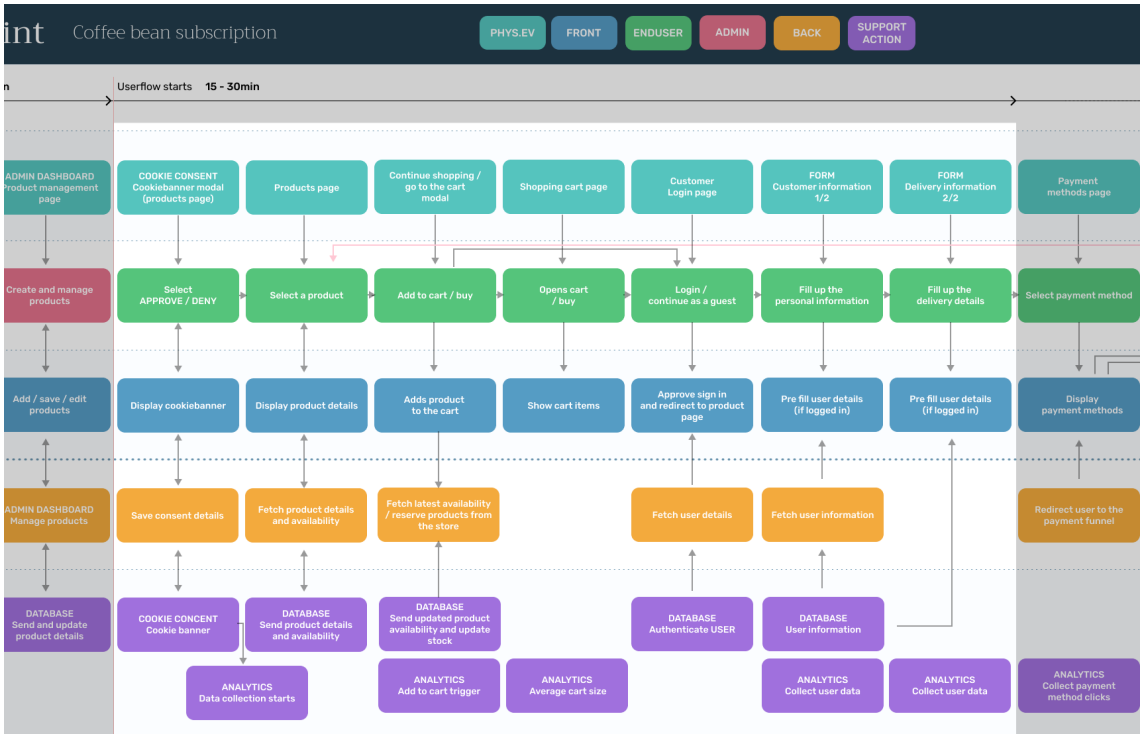

Figure 4.3: Highlighted view of a phase 1, where the customer selects and adds the product to the shopping cart

4.1.2.1 Phase 1a - Customer selecting product

Because the phase is so long, it is better to split it into two smaller parts. In figure 4.4, the first column is about the cookie consent banner that is shown to the user before they can proceed further. This is the user's first touch point when they land on the website and this is where data measurement such as Google Analytics starts. Due to GDPR [30], user consent must be obtained before any data can be collected. The cookie banner can sometimes be a third-party software, which may include monthly fees, and it ensures that no other than necessary cookies are sent before consent is given. Once the user has opted in, the data collection can begin. Then the user can start looking at the products on the landing page, which in this case is the product page.

The user selects a product to add to the shopping cart, with the option to continue shopping and just add the product to the shopping cart, or the user is taken directly to the purchase funnel. If the user decides to add a product to the shopping cart and continue shopping, then they can open the shopping cart page later.

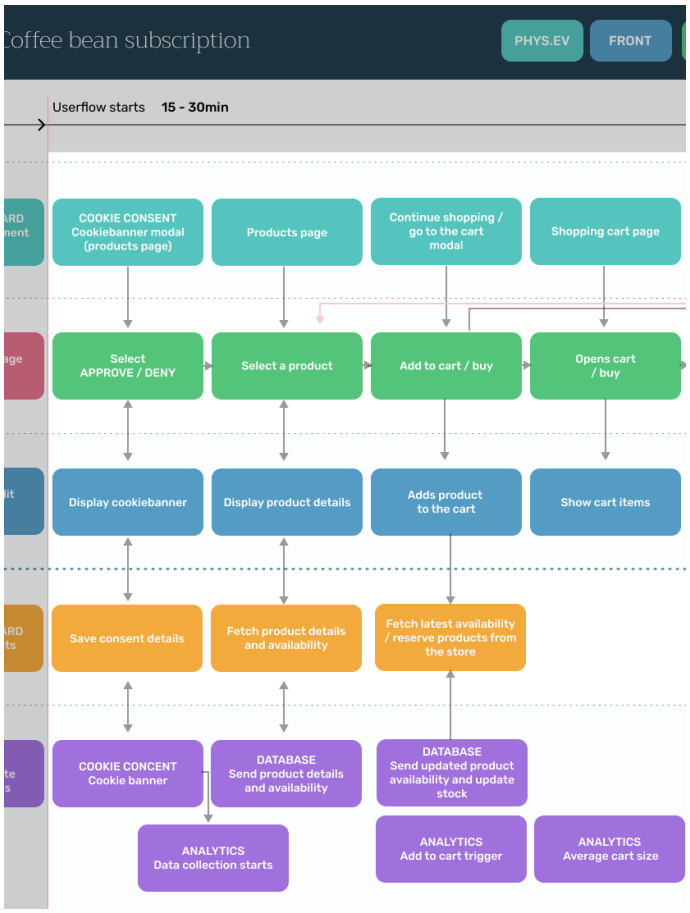

Figure 4.4: Highlighted first part of a phase 1

4.1.2.2 Phase 1b - Login and fill in the forms

After the user either clicked the "buy" button, after adding the product to the cart or opened the shopping cart with items in it separately, the user is moved to the buy section of the phase 1 where the user has to either login as an existing user or continue as a guest. In this example, we assume the user logged in with a previously created account, so we can fetch already saved user information from the back end. When the first page of the form is shown to the customer and we have identified the correct user after login, we can fetch the user information and fill up the details as much as possible. Now the user has to fill up a bare minimum of the customer information details before moving to the next step of the form, which includes filling in the delivery information. This is also pre-filled based on the users data from the back end. After filling in all the information, the user needs to choose a payment method. The user is redirected to the correct payment funnel after selection.

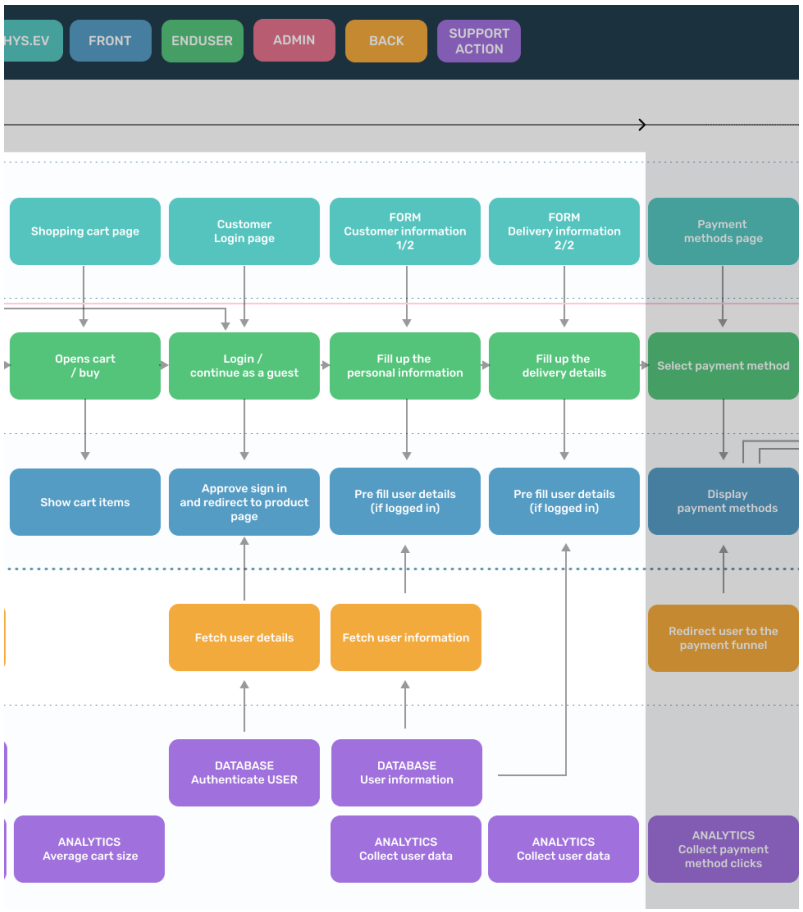

Figure 4.5: Highlighted second part of a phase 1

4.1.3 Phase 2 - Payment funnel

The next important phase in the customers' actions is the payment funnel. Depending on the payment method, the user may need to use a different third-party payment funnel. In this example case, the user has a standard card payment solution provided by Stripe or other similar third-party software, which includes all the modern credit and debit card payment methods. If the user wants to use MobilePay, then it is a different platform, or if the user chooses to pay by invoice, it is an in-house-built payment funnel that does not redirect the user outside of the website. After completing the payment funnel, the user receives a thank you message and software, such as Mailgun, then sends a confirmation email with a summary of the purchase. This phase depends mainly on the third-party payment funnel and how well it is designed and how difficult it is for the user to make a purchase. This phase might approximately take 5 to 10 minutes.

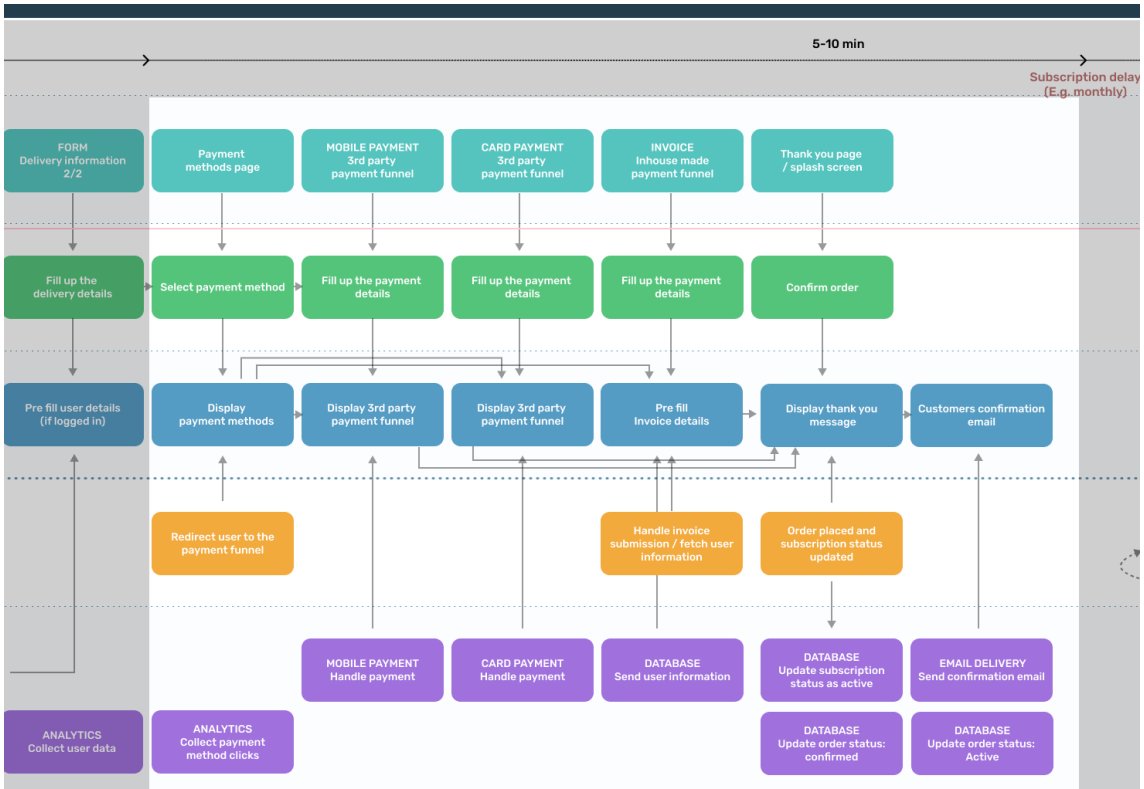

Figure 4.6: Highlighted view of a phase 2

4.1.4 Phase 3 - Subscription fulfilment phase

In the subscription fulfilment phase, most of the actions are performed by real people under the line of visibility. In this phase, the order is packed and the operator updates stock availability. In the order phase, the products are only marked as reserved, but now they are actually marked as fulfilled. Now the courier company arrives and the user and the courier receive the order information that the order is packed and ready for collection. The user also receives a notification on the website's front end that the order is packed and waiting to be delivered to the parcel locker. The user will receive a notification when the parcel is delivered to the parcel locker. When the user claims the package from the locker, the order status is updated as delivered. This phase is repeated depending on the customer's order. For example, the user has ordered a monthly subscription that includes two bags of coffee beans once a month. This phase is then repeated every month until the customer cancels the subscription or it is automatically cancelled, if the payment fails.

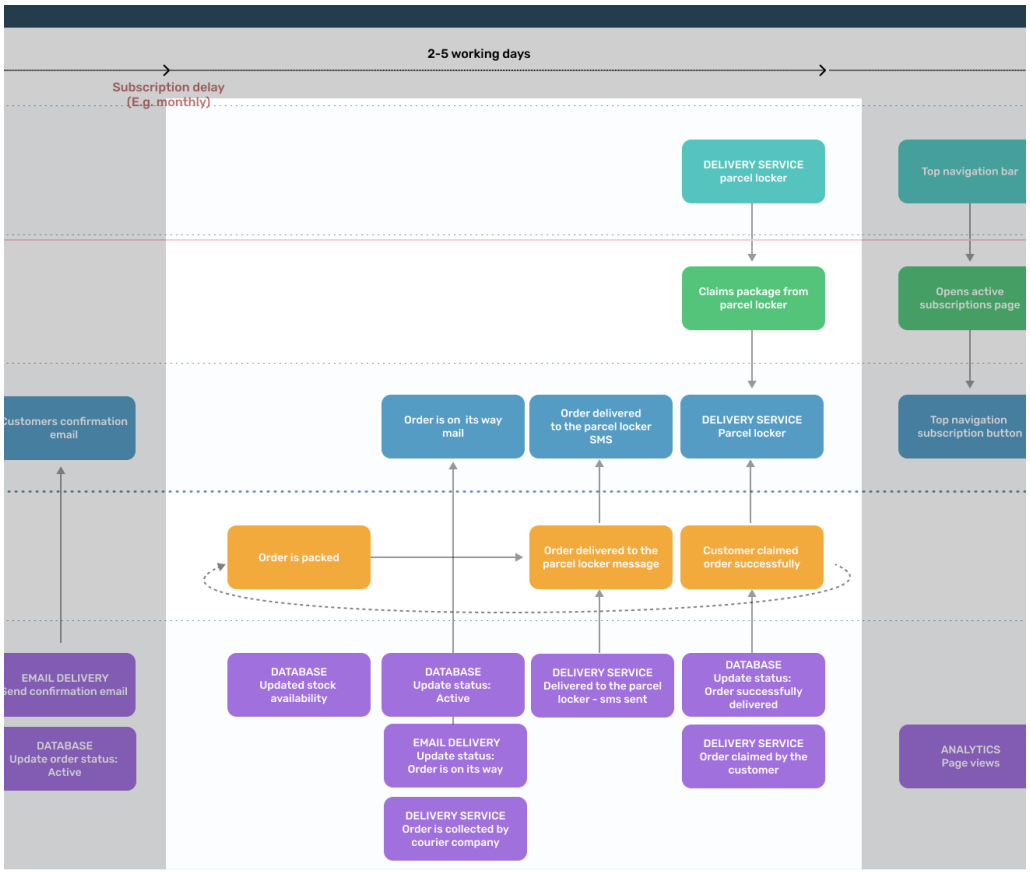

Figure 4.7: Highlighted view of a phase 3

4.1.5 Phase 4 - Customer cancelling their subscription

If the customer decides to cancel their subscription, they will need to go through the top navigation bar to the active subscriptions page where the subscription can be cancelled. After cancelling the subscription, a confirmation modal will be displayed and the cancellation will be confirmed. The user will be notified with the cancellation email and the subscription status will be updated in the back-end. Analytics will collect all the data about the cancellations. This phase should take no more than a few minutes to finish.

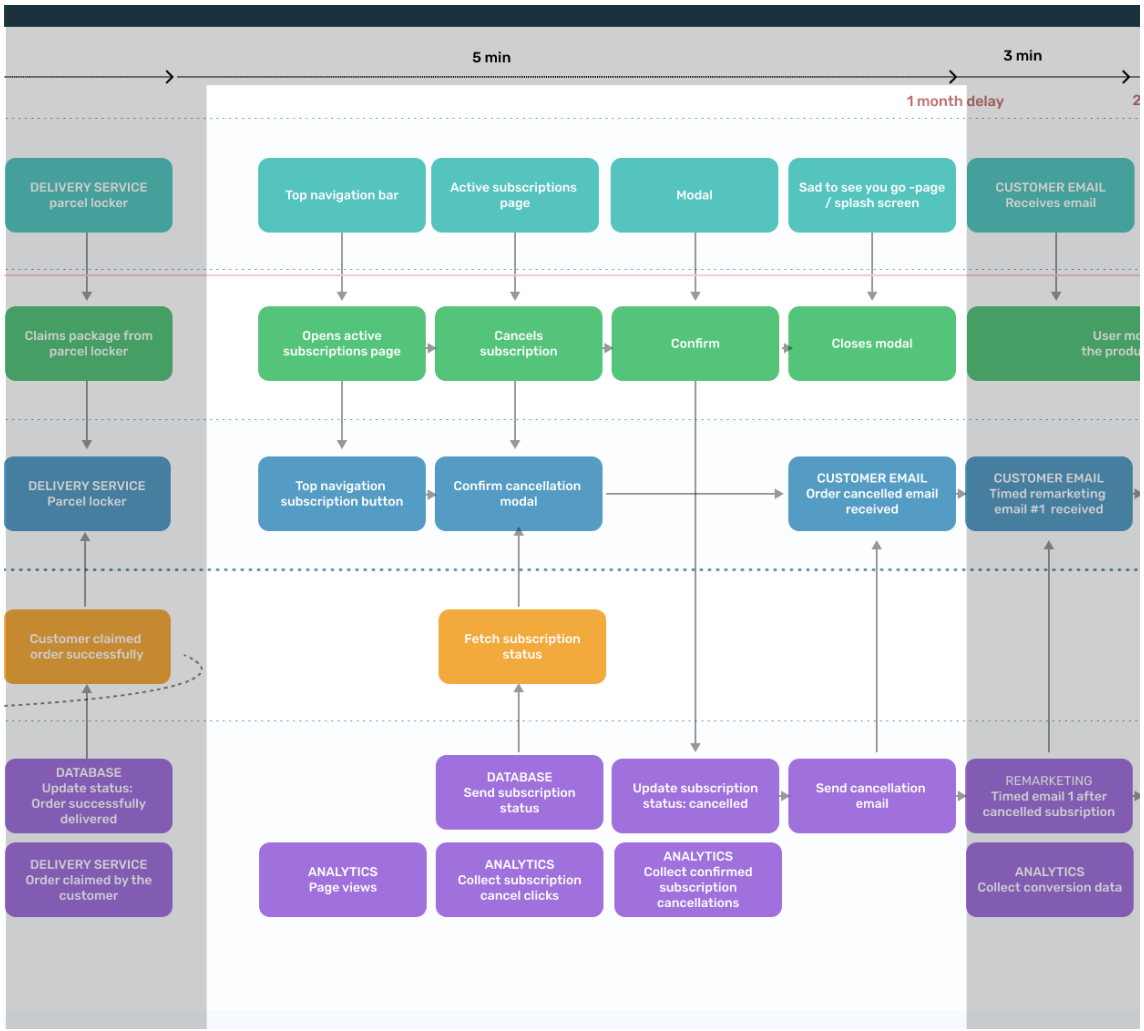

Figure 4.8: Highlighted view of a phase 4

4.1.6 Phase 5 - Email marketing

After the user has cancelled their subscription, the "returning user" marketing emails can be activated. After a month or two, when the user has been without their coffee subscription for a short period of time, they can be offered some very tempting discounts to make them re-activate their subscription. Then the second round of scheduled emails can be sent. These emails can be handled by third-party software, such as Mailchimp, and analytics will collect all the conversion data about the users. Conversion in this campaign could be what percentage of all people who received the email reactivated their subscription or placed another order with an active subscription. From the email campaign, former customers are sent back to the product page and the whole process starts again.

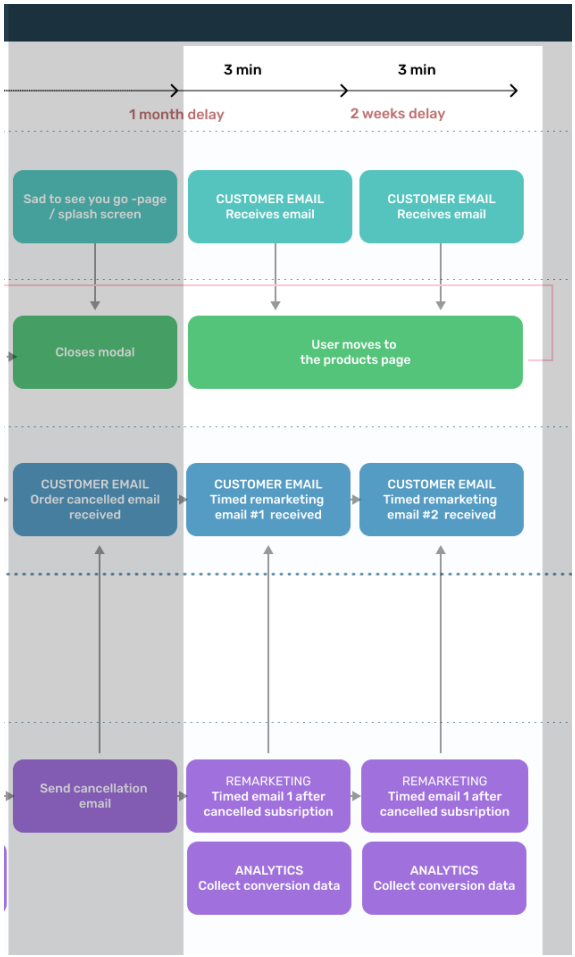

Figure 4.9: Highlighted view of a phase 5

Refining service blueprint

5.1 Main flaws of the service blueprint

When we think about the service blueprint and its whole evolution from the traditional services to the digital services, the concept has always been exactly the same and the differences in the whole design are very minor when the digital services has been constantly evolving and with an incredible fast phase. This makes service blueprint appear very outdated, but yet it is still very capable way of representing modern digital services. Different lanes are difficult to understand and all human interaction disappears in the flow without any clear differences. In the following few sections, the main flaws are viewed in detail and what could be the possible solutions to implement in the refined version of the service blueprint (Figure: 5.1)

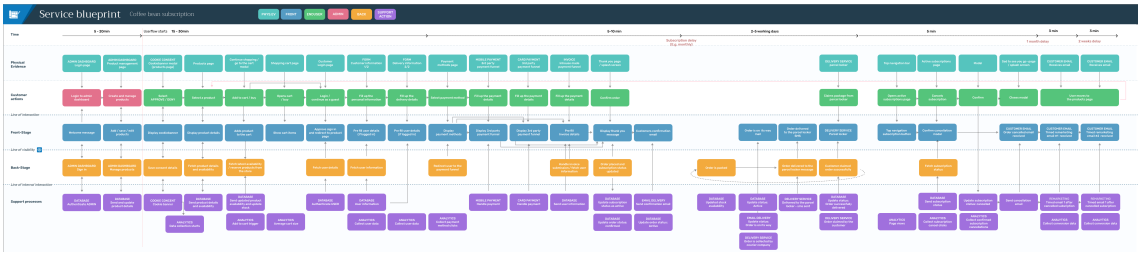

Figure 5.1: Service Blueprint of a coffee bean subscription

5.1.1 Lack of phases and efficient timeline

Digital services can easily be divided into different phases that represent one user flow to another. This flow or phase consists of several steps that together represent a user goal. An example of a goal might be for the user to create an account for the service. The timeline should include a clear way of presenting the phases so that it is easy to understand which section is linked or grouped. In digital services, there may be several individual steps that happen in a matter of seconds, and grouping these steps helps to understand where the actions begin and end within a group.

One of the major shortcomings of the service blueprint is the lack of a clear representation of phases. There is currently no consistent way of presenting phases and the time taken to complete them. For the example project (Figure:5.2), I have added arrows to indicate a point where one phase begins and the other ends. This is not a very informative way, as there are no titles or labels. There is no consistent position or clear place to show the time taken to complete this phase.

Figure 5.2: Current timeline in the example service blueprint

To solve this problem, the timeline is divided into several phases, with the start and end of each phase marked so that the reader can see how long it will take the user to complete the phase. Each phase has a title to help understand the goal and the purpose of the phase. Adding these makes the service structure easier to follow and read. Each phase on the timeline can be separated into individual columns with their completion times, and the phase timeline shows the total time it will take the user to reach the goal. Now it is easy to identify all the time-consuming phases and each individual action within them (Figure:5.3).

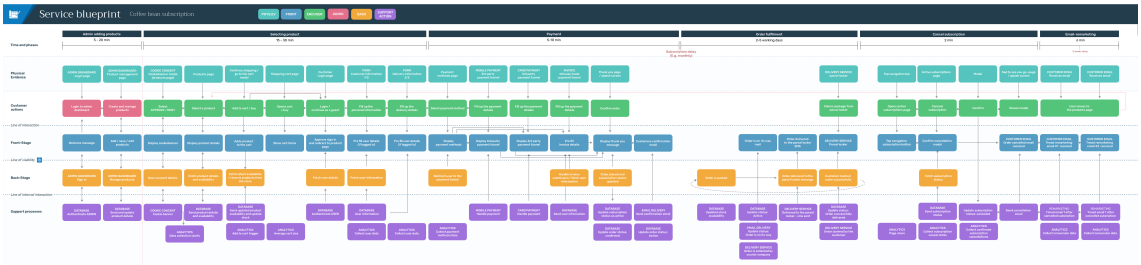

Figure 5.3: Service blueprint with updated timeline including phases and phase completion times

5.1.2 Composable building blocks or PBCs are not clearly visualised in the service blueprint

When we think about service blueprints and how they are a simple way to show the big picture of digital services, there is still a major flaw in creating a service blueprint that would serve all the stakeholders or the owners of the digital service. Section 3.1.3

describes three different types of PBCs and these are very difficult to identify through the service blueprint. The value of this information can be significant if the stakeholder could immediately see which blocks are produced in-house for the service, rather than being purchased separately from external third-party companies. This information helps to show more clearly what needs to be considered, if the data collected from the end users show any shortcomings in the service flow. For example, if there is a flaw in the user flow based on negative user feedback, then it is easy to see whether that step in the process was developed in-house or purchased separately. There may be a case where all the external third-party components are performing extremely well, while all or most of the in-house developed PBCs are not performing as well as they should, implying a lack of funding, time, or skills and competence on the part of the team that developed those PBCs.

Figure 5.4: Composable building blocks for service blueprint

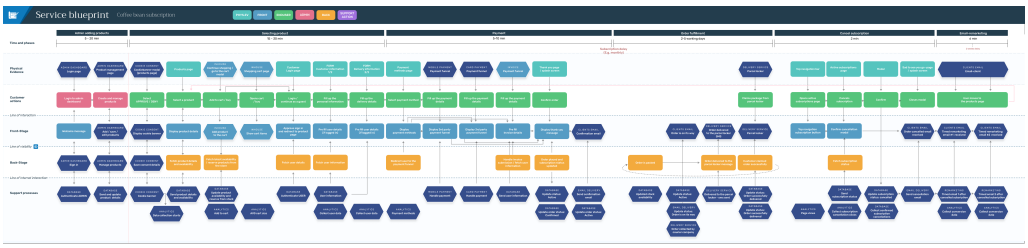

Figure 5.5: Service blueprint with building blocks added

5.1.3 Orchestration layer

The next addition to the service blueprint is the orchestration layer for composable building blocks. This adds an extra layer after the line of internal interaction and it contains all the external, internal, and hybrid building blocks used in the service. This layer contains APIs for the main service to use properly and orchestrates external building blocks. All PBCs that are used above the line of internal interaction have their user interfaces and the end user must interact with external user interfaces at these points. This provides a clearer view of all the touch points where the user experience can be impacted, potentially resulting in negative user feedback. The user experience can be negatively impacted if the user has to interact with multiple different interfaces from multiple different vendors within the service. The orchestration layer does not try to be an API layer or a service layer, but instead uses the APIs and manages them so that the end user can use the web service seamlessly.

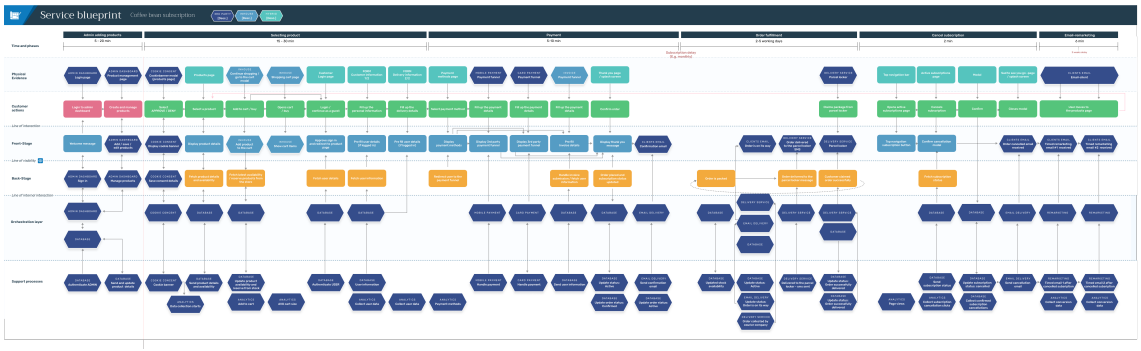

Figure 5.6: Service blueprint with orchestration layer added

5.1.4 improvement of the service blueprint structure

5.1.4.1 Visual cues and colours

If we take a look at the big picture of the service blueprint with the orchestration layer added (Figure: 5.8), we can see that it is fairly easy to follow using the classic swim lane structure. In the first refined version, each lane is highlighted in its own colour to make it slightly easier to follow each lane. This approach gives no further information about the step itself, it is just stating the obvious as they are already separated into their own lanes. To improve this design, some key parts are either missing, difficult to understand or there is no clear way to display different data that is being executed simultaneously in the same column.

Looking at the current service blueprint after adding the orchestration layer, there are multiple colours present to indicate different lanes and even a red colour for different user types. After adding the orchestration layer and the composable building blocks with their respective colours to the service blueprint, the different colours in the swim lanes make the design more difficult to read and more complex to understand without providing any real value to the user. The row and different shapes indicate the type without colour coding the different lanes. Only the composable building blocks with hexagonal shapes are coloured differently depending on the type, whether it is a third-party, in-house made, or hybrid solution, which is explained in section 3.1.3 and in Figure: 5.4. The rest of the blocks are coloured with only one specific colour which in this case is green. This helps the reader to quickly separate all the composable building blocks from other user interactions (Figure: 5.7).

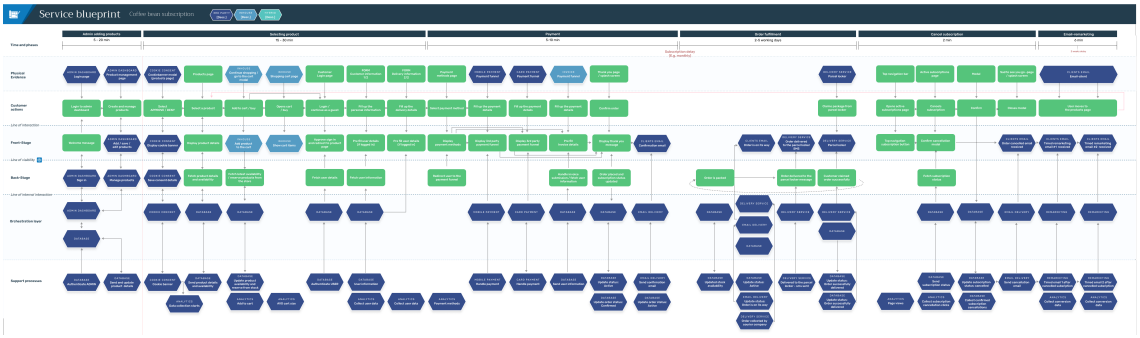

Figure 5.7: Service blueprint with simplified colours

5.1.4.2 Order of the actions

If we look at the current state of the service blueprint in the previous figure (Figure: 5.7), we can see that the customer actions are all the same shape and type, as they are all about the customer actions. The physical evidence row is before the customer actions, which makes the order very odd as we are talking about the web services and not the physical services. The Customer Actions and Physical Evidence layers should be swapped, so the Customer Actions would be on top of the service blueprint and the Physical Evidence row should be renamed as a User Interface.

When we consider web services and how users interact with them, it is always through some form of a user interface. We could use the Physical Evidence layer on top as the evidence of where the customer heard first time about the actual service, such evidence as a marketing campaign or an advertisement. This layer is useless because most web services have one or more linear flows where the user has to move from start to finish, so there is too little evidence that is not a user interface to have its own row. For those services that have heavy advertising campaigns, or the most of their traffic comes from external sources,it might be useful to add the Physical Evidence layer on top of the User Interface row and rename the Physical Evidence as an External Evidence.

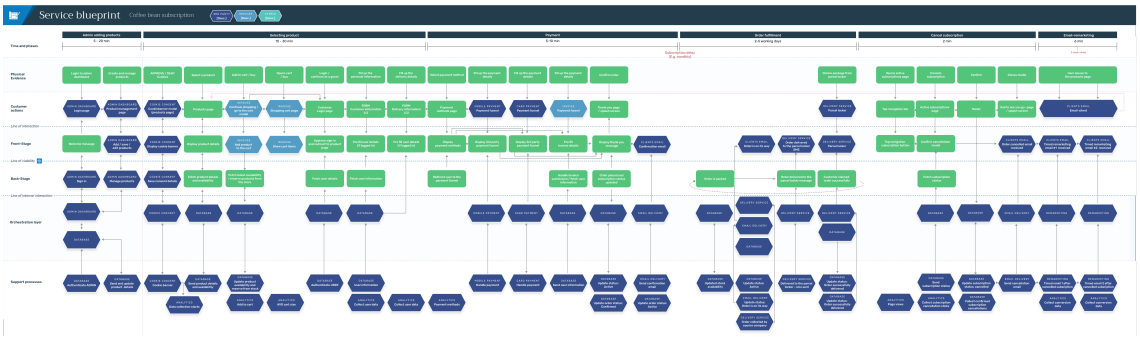

Figure 5.8: Service blueprint with KPI touchpoints (yellow rectangles)

5.1.5 Analytics and KPI touch points

Key Performance Indicators, or KPIs, are goals that a web service owner wants to achieve. These KPIs are important for measuring the core functionality of the service, as these KPIs are the reason why the service exists in the first place. Without KPIs, the service may work perfectly for the customer, but it may be built solely from the customer's perspective and the service model may not be sustainable to cover the cost of maintaining the service. If the web service is not profitable and the service blueprint does not support KPIs, this could lead to a shorter lifespan of the service. To tackle the bloated support processes, it is better to separate all the analytics and KPIs from the support processes into their

own row. This row briefly shows what and where the analytics data is collected and where the possible KPI touch points could be. In this example, the customer KPI could be to either improve or maintain a conversion rate of at least five percent. This means that for every 100 customers that land on the products page, 5 of them successfully buys a subscription. This row helps to understand what data is really necessary to collect and what the owner of the web service expects from the service. Final image includes few example key performance indicator, which are marked as yellow under the analytics. These help to understand the most important steps which needs to be focused. As a KPI's in this example from left to right, there are raising percentage of engaged sessions. Engaged session represents an user that performs some user actions on the site, such as scrolling or viewing multiple pages. Engaged user does not include users that doesn't do anything on the site. Then there are raising the average cart value and conversion. Conversion includes all the successful orders where the transaction has been completed. After conversion there are also lowering subscription cancellation rate and raising amount of returning customers.

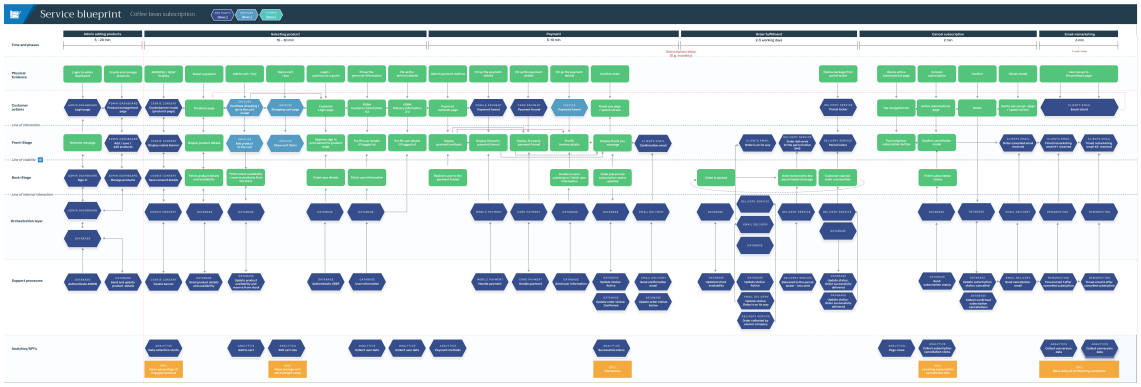

Figure 5.9: Final version of the refined service blueprint

5.2 Refined service blueprint

The final version of the service blueprint now contains all the improvements which all are made to satisfy the needs of a composable service. Now the stakeholders have a clear view of the whole service structure without going too deeply into the technical details. All the phases and sections that the user needs to accomplish before reaching the goal of the service are clearly visible with the service key performance indicators as well. Current colour coding helps the user to immediately pin point all the sections where the user is interacting with in-house made components or components that are controlled by third party services. API layer helps the non-technical and technical viewers to understand why and where the data is stored and fetched from third party services. This acts as a guide to understanding what they are paying for, if there are multiple SaaS (Software as a Service) subscriptions needed which may stack up huge monthly fees.

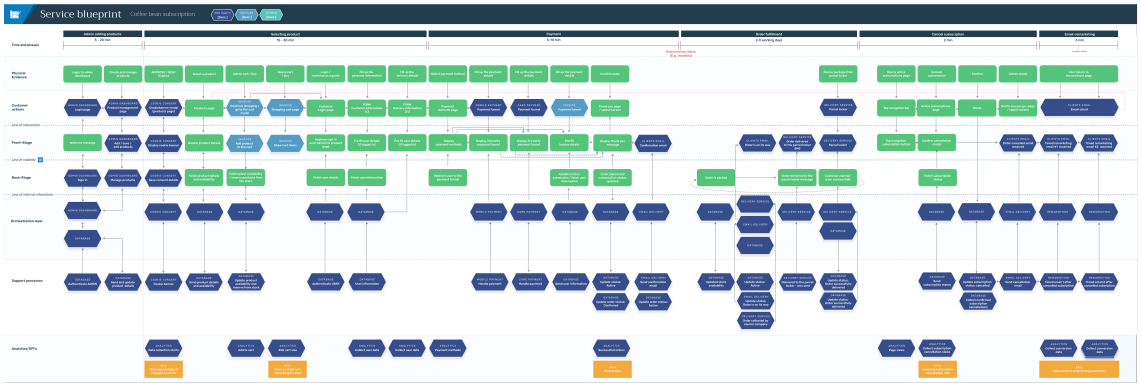

Figure 5.10: Implemented building blocks

Conclusions

The service blueprint acts as an easy-to-understand overview of the services while revealing the underlying problems within the user path from start to finish; it is quite obvious that some improvements are to be expected as the services grow and the real human interaction between the user and the service disappears. This paper has focused on the main shortcomings of the original version of the service blueprint when scaled to a more complex and fast-paced digital world. The refined version is more accessible in this way, as the customer actions start from the top row, where the reader can easily follow either the horizontal or vertical paths of the blueprint.

Keeping the physical evidence as the top row, to represent the first touch point of the service would have been really inconvenient since all the channels and ways in which people the first time hear about the digital service environment need their own section, which ends up being more on the marketing side than the technical side. There can be hundreds of different marketing campaigns, physical and digital evidence, so adding these would clutter the whole process without adding any information to the efficiency of the service. The service blueprint should focus on the digital service as a whole. In the refined version, the physical evidence is placed after the user's actions and now represents the physical interface with which the user interacts.

The addition of analytical metrics data to the service blueprint may depend on the service itself. The analytics layer is not mandatory, but it helps to understand where all the key steps are and where the data needs to be measured. Tracking the efficiency of digital services is impossible without data of the users behaviour, because the whole service can contain hundreds of different steps that the user has to take in a very short period of time.

Without any collected data, inefficient parts of the service blueprint may be impossible to find. For example, if there is data on the percentage of each step about how many users move on to the next step until the final goal is reached, then the steps with the most issues can be easily fixed. A service blueprint is an excellent tool for designing and mapping out the entire user flow, which can be transferred to the development board. These development boards, such as Kanban boards, can contain an entire service blueprint

broken down into smaller pieces for the developers. Now the service blueprint acts as a good tool for the stakeholders, while the developers can focus on the more detailed and technical parts of the service in the development board and in technical documentation.

6.0.1 Future development of the service blueprint

The next section suggests ways in which the Service Blueprint could be further developed to provide more flexibility in larger and more complex projects. Because the current example project about the coffee bean subscription is so simple and involves very short user journeys, the next suggestions do not apply to this project and include suggestions for how the service blueprint can be extended into the different and more complex projects.

6.0.1.1 Dividing phases to multiple service blueprints

For more complex digital services, all the phases included in the service blueprint could be broken down into their own individual service blueprints. As the example service is very simple and does not have many steps, the whole service could be presented as one service blueprint without it being excessively long. The most convenient way to structure the service blueprint hierarchy for the more complex and longer user journeys would be to have a separate service blueprint for each of the phases (Figure: 6.1), which can either be displayed as separate service blueprints or opened as one big service blueprint that shows the whole project at once without jumping between phases.

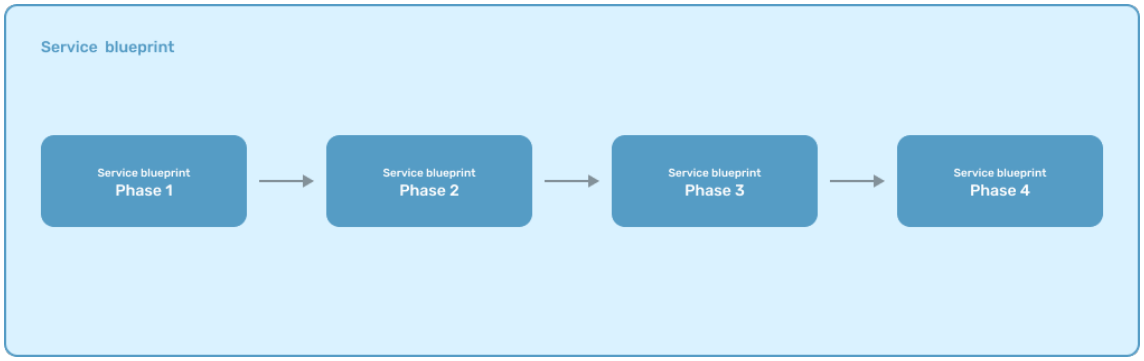

Figure 6.1: Example of multiple phases and each of them would lead to different service blueprints

6.0.1.2 Cost of third-party services

An important part of the ever-increasing use of third-party services, such as Saas (Software as a Service), PaaS (Platform as a Service) and IaaS (Infrastructure as a Service), has a relatively big impact on the monthly costs that the owner of the digital service may

face when building their service with third-party composable building blocks. Reviewing all the parts that add to the monthly cost, gives a good idea of how much money they are spending on all these services. At some point, it may be necessary to discard the inefficient parts and replace them with cheaper or in-house-made options. Thus including the visible costs of the third-party building blocks could provide very good information about efficiency and whether the investment is worth the money. These third-party elements are key parts of any composable digital service, as developing them in-house can be expensive and the end product may not deliver a good return on the time and money invested. For example, the payment funnel may be relatively easy to create and implement, but why bother when you can have a payment funnel that has taken thousands of hours of development and dozens of iterations to create a smooth and highly converting funnel for a fraction of the price. However, the return on investment (ROI) could be inefficient if the third party service is a bottleneck for the whole service.

List of Figures

References

[1] D. Schallmo, C. Williams, and D. Schallmo, "History of digital transformation," in Jan. 2018, pp. 3–8, ISBN: 978-3-319-72843-8.

[2] S. Gibbons. "Service blueprints: Definition." (2017), [Online]. Available: `https://www.nngroup.com/articles/service-blueprints-definition/` (visited on 01/07/2022).

[3] R. Kaufman, *"what is service?" ron kaufman's eye-opening insights on the definition of service*, 2020. [Online]. Available: `https://www.youtube.com/watch?v=EKYpBaiZsc8` (visited on 06/02/2022).

[4] G. Parry, L. Newnes, and X. Huang, "Goods, products and services," in *Service Design and Delivery*, M. Macintyre, G. Parry, and Angelis, Eds., Springer Science+Business Media, LLC 201, 2011, pp. 19–22,26.

[5] *Digital Services – How Are They Different*, (Taipei, Taiwan), Proceedings of International Conference on Business, Information, and Cultural Creative Industry (ICBIC14), Aug. 2014, pp. 2–3.

[6] G. L. Shostack, "How to design a service," vol. 16, no. 1, p. 52, 1982.

[7] P. Hill, "Tangibles, intangibles and services: A new taxonomy for the classification of output," *Canadian Journal of Economics*, vol. 32, no. 2, p. 439, 1999.

[8] G. L. Shostack, "Breaking free from product marketing," *Journal of Marketing*, vol. 42, no. 2, pp. 74–77, 1977.

[9] J. J. Sierra and S. McQuitty, "Service providers and customers," in *Journal of Services Marketing*, 6, R. Russell-Bennett and M. Rosenbaum, Eds., vol. 19, Emerald Publishing Limited, 2005, p. 393.

[10] C. Michael, Nachiappan, P. Pradhan, V. Sanjeev, and K. Vignesh, "Blockchain technology," Sutardja Center for Entrepreneurship and Technology, 1111 Franklin St. Oakland, CA 94607, Tech. Rep., 2015.

[11] F. Regner, A. Schweizer, and N. Urbach, "Nfts in practice – non-fungible tokens as core component of a blockchain-based event ticketing application," Dec. 2019.

[12] C. Smyth. “Nft tokens issued on most platforms are perishable.” (2021), [Online]. Available: https://hackernoon.com/nft-tokens-issued-on-most-platforms-are-perishable-cliff-smyth-fk1r330v (visited on 06/02/2022).

[13] I. Ali and R. K. Garg, “Marketing of services: Challenges and opportunities in context of the globalization of business,” *International Journal of Engineering and Management Research*, vol. 7, no. 3, p. 524, 2017.

[14] G. L. Shostack, “Designing services that deliver,” in *Harvard Business Review*, 1, vol. 62, Harvard Business Publishing, 1984, pp. 133–139. [Online]. Available: https://hbr.org/1984/01/designing-services-that-deliver/ (visited on 01/08/2022).

[15] M. Haugen, “Service blueprints persistent qualities and future potential,” 2013.

[16] B. D. Weinberg, “Don’t keep your internet customers waiting too long at the (virtual) front door,” *Journal of Interactive Marketing*, vol. 14, no. 1, pp. 30–39, 2000, ISSN: 1094-9968. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1094996800701976.

[17] Z. Gecis. “The many ways of mapping the customer journey.” (2020), [Online]. Available: https://uxdesign.cc/customer-journey-map-service-blueprinting-f06750312150.

[18] F. N. M. Mary Jo Bitner Amy L. Ostrom, “Service blueprinting: A practical technique for service innovation,” vol. 50, no. 3, p. 95, 1999.

[19] P. Harrisson. “Rethinking service blueprints for agile delivery.” (2019), [Online]. Available: https://www.wipro.com/digital/rethinking-service-blueprints-for-agile-delivery/ (visited on 04/03/2023).

[20] Tonkean. “What is composability? the road to the composable enterprise.” (2022), [Online]. Available: https://tonkean.com/blog/it/it2/what-is-composability-road-composable-enterprise/.

[21] S. Andreas. (2021), [Online]. Available: https://commons.wikimedia.org/wiki/File:Construction_Bricks.svg.

[22] T. Malinen. “What is low-code and no-code?” (2022), [Online]. Available: https://www.sofokus.com/what-is-low-code-and-no-code/.

[23] K. Panetta. “Gartner keynote: The future of business is composable.” (2020), [Online]. Available: https://www.gartner.com/smarterwithgartner/gartner-keynote-the-future-of-business-is-composable.

[24] T. Malinen. “Adaptable business.” (2022), [Online]. Available: https://www.sofokus.com/adaptable-business/.

[25] D. Free. “Gartner strategic roadmap 2021.” (2021), [Online]. Available: https://www.gartner.com/en/doc/433984-2021-strategic-roadmap-for-the-composable-future-of-applications.

[26] H. Jarret. “What are packaged business capabilities?” (2021), [Online]. Available: https://www.elasticpath.com/blog/what-are-packaged-business-capablities.

[27] M. Lowndes. (2021), [Online]. Available: https://www.youtube.com/watch?v=mP6m9Yq0rS8&t=1s.

[28] G. Vimont, “Application programming interface,” 2015.

[29] R. Mangiaracina. (2009), [Online]. Available: https://www.icommercecentral.com/open-access/the-ecommerce-customer-journey-a-model-to-assess-and-compare-the-user-experience-of-the-ecommerce-websites.php?aid=38368&view=mobile (visited on 07/03/2023).

[30] B. Wolford. (2023), [Online]. Available: https://gdpr.eu/gdpr-consent-requirements/ (visited on 07/03/2023).

[31] N. Remis, *A guide to service blueprinting*, 2016. [Online]. Available: https://medium.com/capitalonedesign/download-our-guide-to-service-blueprinting-d70bb2717ddf.

[32] V. Lieby. “Worldwide professional developer population of 24 million projected to grow amid shifting geographical concentrations.” (2019), [Online]. Available: https://evansdata.com/press/viewRelease.php?pressID=278.