STEP

1

Thinking in Layers and Aspects

Analyzing systems by separating them into layers and aspects

This step lays the foundation of our learning path through the blockchain by introducing a way to organize and standardize our communication about technology. This step explains how you can analyze a software system and why it is important to consider a software system as a composition of layers. Furthermore, this step illustrates what you can gain from considering different layers in a system and how this approach helps us to understand the blockchain. Finally, this step provides a short introduction to the concept of software integrity and highlights its importance.

why it is important to consider a software system as a composition of layers. Furthermore, this step illustrates what you can gain from considering different layers in a system and how this approach helps us to understand the blockchain. Finally, this step provides a short introduction to the concept of software integrity and highlights its importance.

The Metaphor

Do you have a mobile phone? I would guess yes, as most people now have at least one. How much do you know about the different wireless communication protocols that are used to send and receive data? How much do you know about electromagnetic waves that are the foundation of mobile communication? Well, most of us do not know very much about these details because it is not necessary to know them in order to use a mobile phone and most of us do not have the time to learn about them. We mentally separate the mobile phone into the parts we need to know and the parts that can be ignored or taken for granted.

This approach to technology is not restricted to mobile phones. We use it all the time when we learn how to use a new television set, a computer, a washing machine, and so forth. However, these mental partitions are highly individual since what is considered important and what is not depends on our individual preferences, the specific technology, and our goals and experiences. As a result, your mental partition of a mobile phone may differ from my mental partition of the same mobile phone. This typically leads to problems in communication in particular when I try to explain to you what you should know about a certain mobile phone. Hence, unifying the way of partitioning a system is the key point when teaching and discussing technology. This step explains how to partition or layer a system and hence sets the basis for our communication about the blockchain.

Layers of a Software System

The following two ways of partitioning a system are used throughout this book:

- Application vs. implementation
- Functional vs. nonfunctional aspects

Application vs. Implementation

Mentally separating the user's needs from the technical internals of a sys-

Layers of a Software System

The following two ways of partitioning a system are used throughout this book:

- Application vs. implementation
- Functional vs. nonfunctional aspects

Functional vs. Nonfunctional Aspects

Distinguishing between what a system does and how it does what it does leads to the separation of functional and nonfunctional aspects. Examples of functional aspects are sending data over a network, playing music, taking photos, and manipulating individual pixels of a picture. Examples of nonfunctional aspects are a beautiful graphical user interface, fast-running software, and an ability to keep user data private and save. Other important nonfunctional aspects of a system are security and integrity. *Integrity* means that a system behaves as intended, and it involves many aspects such as security and correctness.[1] There is a nice way to remember the difference between functional and nonfunctional aspects of a system by referring to grammar usage in the English language: verbs describe actions or what is done, while adverbs describe how an action is done. For example, a person can walk quickly or slowly. In both cases, the action of "walk" is identical but how the action is performed differs. As a rule of thumb, one can say that functional aspects are similar to verbs, while nonfunctional aspects are similar to adverbs.

Considering Two Layers at the Same Time

Identifying functional and nonfunctional aspects as well as separating application and implementation layer can be done at the same time, which leads to a two-dimensional table. Table 1-1 illustrates the result of mentally layering a mobile phone in this way.

Table 1-1. Example of Mentally Layering a Mobile Phone

Layer	Functional Aspects	Nonfunctional Aspects
Application	Taking photos Making phone calls Sending e-mails Browsing the Internet Sending chat messages	The graphical user interface looks beautiful Easy to use Messages are sent fast
Implementation	Saving user data internally Making a connection to the nearest	Store data efficiently Saving energy

Layer	Functional Aspects	Nonfunctional Aspects
Application	Taking photos Making phone calls Sending e-mails	The graphical user interface looks beautiful Easy to use Messages are sent fast

Table 1-1 may explain the visibility (or the lack of it) of specific elements of a system to its users. Functional aspects of the application layer are the most obvious elements of a system, because they serve obvious needs of the users. These elements are typically the ones users learn about. On the other hand, the nonfunctional aspects of the implementation layer are rarely seen as major elements of the system. They are typically taken for granted.

Integrity

Integrity is an important nonfunctional aspect of any software system. It has three major components[2]:

- *Data integrity*: The data used and maintained by the system are complete, correct, and free of contradictions.
- *Behavioral integrity*: The system behaves as intended and it is free of logical errors.
- *Security*: The system is able to restrict access to its data and functionality to authorized users only.

Most of us may take integrity of software systems for granted because most of the time we luckily interact with systems that keep their integrity. This is due to the fact that programmers and software engineers have invested a lot of time and effort into the development of systems to achieve and maintain integrity. As a result, we may be a bit spoiled when it comes to appreciating the work done by software engineers to create systems that maintain a high level of integrity. But our feelings may change as soon as we interact with a system that fails to do so. These are the occasions when you face a loss of data, illogical software behavior, or realize that strangers were able to access your private data. These are the occasions when your mobile phone, your computer, your e-mail software, your word processor, or your spreadsheet calculator make you angry and forget your good manners! On these occasions, we begin to realize that software integrity is a highly valuable commodity. Hence, it should not come as a surprise that software professionals spend a lot of their time working on this seemingly tiny nonfunctional aspect of the implementation layer.

system that fails to do so. These are the occasions when you face a loss of data, illogical software behavior, or realize that strangers were able to access your private data. These are the occasions when your mobile phone, your computer, your e-mail software, your word processor, or your spreadsheet calculator make you angry and forget your good manners! On these occasions, we begin to realize that software integrity is a highly valuable commodity.

Outlook

This step provided an introduction to some general principles of software engineering. In particular, the concepts of integrity and functional vs. nonfunctional aspects as well as application vs. implementation of a software system were illustrated. Understanding these concepts will help you appreciate the wider scope in which the blockchain exists. The next step will present the bigger picture by using the concepts introduced in this step.

Summary

- Systems can be analyzed by separating them into:
  - Application and implementation layer
  - Functional and nonfunctional aspects
- The application layer focuses on the user's needs, while the implementation layer focuses on making things happen.
- Functional aspects focus on what is done, while nonfunctional aspects focus on how things are done.
- Most users are concerned with the functional aspects of the application layer of a system, while nonfunctional aspects of a system, in particular those of the implementation layer, are less visible to users.
- Integrity is an important nonfunctional aspect of any software system and it has three major elements:
  - Data integrity
  - Behavioral integrity
  - Security
- Most software failures, such as losses of data, illogical

implementation layer, are less visible to users.

- Integrity is an important nonfunctional aspect of any software system and it has three major elements:
  - Data integrity
  - Behavioral integrity

STEP

2

Seeing the Big Picture

Software architecture and its relation to the blockchain

This step not only provides the big picture in which the blockchain is located, but it also highlights its location within the big picture. In order to allow you to see the big picture, this step introduces the concept of software architecture and explains its relation to the concept of separating a system into layers and aspects. In order to help you recognize the location of the blockchain within the big picture, this step highlights the relationship between the blockchain and software architecture. Finally, this step points out the core purpose of the blockchain in just one sentence. Appreciating its purpose is a cornerstone in understanding the blockchain and understanding the course of the succeeding steps.

to see the big picture, this step introduces the concept of software architecture and explains its relation to the concept of separating a system into layers and aspects. In order to help you recognize the location of the blockchain within the big picture, this step highlights the relationship between the blockchain and software architecture. Finally, this step points out the core purpose of the blockchain in just one sentence. Appreciating its purpose is a cornerstone in understanding the blockchain and understanding the course

of modularization, which is the result of applying the idea of layering to cars. Having the choice among different engines when buying a car can result in amazing differences in the vehicle. Two cars that look identical from the outside can differ dramatically with respect to the power of their engines and hence have very different driving performance. Additionally, your choice of the engine will have an impact on other characteristics of the car, like its price, its operational costs, the type of fuel consumed, the exhaust system, and the dimensions of the brakes. With this picture in mind, understanding the role of the blockchain within the big picture will be much easier.

A Payment System

Let's apply the concept of layering to a payment system. Table 2-1 shows some of the user's needs as well as some of the nonfunctional aspects of both the application and the implementation layers.

Table 2-1. Aspects and Layers of a Payment System

Layer	Functional Aspects	Nonfunctional Aspects
Application	Deposit money	The graphical user interface looks beautiful
	Withdraw money	Easy to use
	Transfer money	Transfer of money is done fast
	Monitor account balance	System has many participants
Implementation	?	Available 24 hours a day
		Fraud resistant
		Maintaining integrity
		Ensure user privacy

Have you spotted the question mark in that part of the table were you normally see information about the technology used to make the system work? This space was left blank on purpose. It is the place where you decide which "engine" should be used to run your system. The next section will tell you a bit more about the engine equivalent in software systems

Fraud resistant

Maintaining integrity

Ensure user privacy

Have you spotted the question mark in that part of the table were you nor-

The two major architectural approaches for software systems are centralized and distributed.[1]

In centralized software systems, the components are located around and connected with one central component. In contrast, the components of distributed systems form a network of connected components without having any central element of coordination or control.

Figure 2-1 depicts these two contrary architectures. The circles in the figure represent system components, also called nodes, and the lines represent connections between them. At this point, it is not important to know the details of what these components do and what information is exchanged between the nodes. The important point is the existence of these two different ways of organizing software systems. On the left-hand side of Figure 2-1, a distributed architecture is illustrated where components are connected with one another without having a central element. It is important to see that none of the components is directly connected with all other components. However, all components are connected with one another at least indirectly. The right-hand side of Figure 2-1 illustrates a centralized architecture where each component is connected to one central component. The components are not connected with one another directly. They only have one direct connection to the central component.

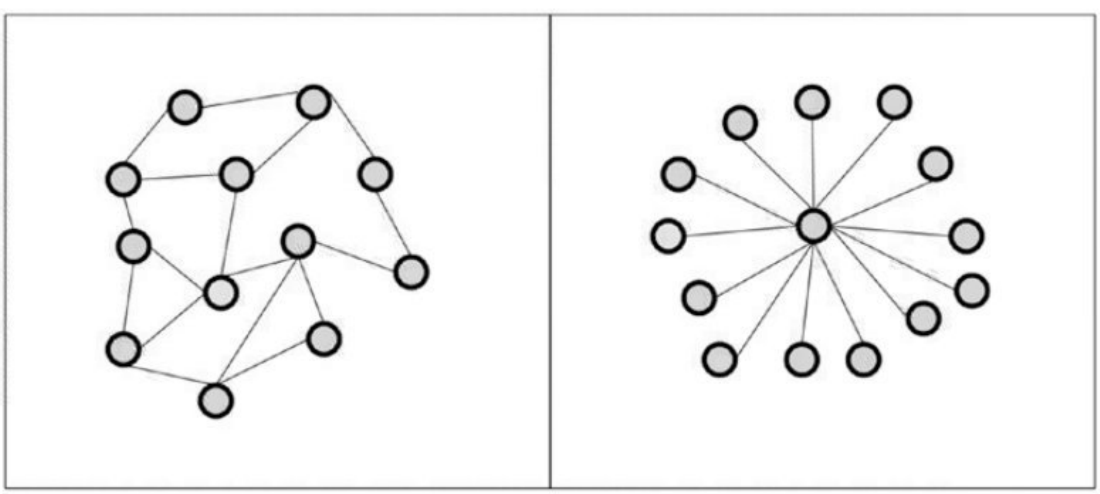
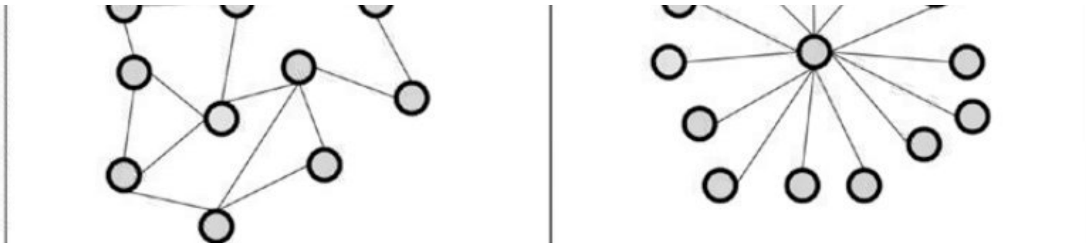


Figure 2-1. Distributed (left) vs. centralized (right) system architecture



The Advantages of Distributed Systems

The major advantages of a distributed system over single computers are[2]:

- Higher computing power
- Cost reduction
- Higher reliability
- Ability to grow naturally

Higher Computing Power

The computing power of a distributed system is the result of combining the computing power of all connected computers. Hence, distributed systems typically have more computing power than each individual computer. This has been proven true even when comparing distributed systems comprised of computers of relatively low computing power with isolated super computers.

Cost Reduction

The price of mainstream computers, memory, disk space, and networking equipment has fallen dramatically during the past 20 years. Since distributed systems consist of many computers, the initial costs of distributed systems are higher than the initial costs of individual computers. However, the costs of creating, maintaining, and operating a super computer are still much higher than the costs of creating, maintaining, and operating a distributed system. This is particularly true since replacing individual computers of a distributed system can be done with no significant overall system impact.

Higher Reliability

The increased reliability of a distributed system is based on the fact that the whole network of computers can continue operating even when individual machines crash. A distributed system does not have a single point of failure. If one element fails, the remaining elements can take over. Hence, a single super particularly true since replacing individual computers of a distributed system can be done with no significant overall system impact.

Higher Reliability

The increased reliability of a distributed system is based on the fact that the

Ability to Grow Naturally

The computing power of a distributed system is the result of the aggregated computing power of its constituents. One can increase the computing power of the whole system by connecting additional computers with the system. As a result, the computing power of the whole system can be increased incrementally on a fine-grained scale. This supports the way in which the demand for computing power increases in many organizations. The incremental growth of distributed systems is in contrast to the growth of the computing power of individual computers. Individual computers provide identical power until they are replaced by a more powerful computer. This results in a discontinuous growth of computing power, which is only rarely appreciated by the consumers of computing services.

The Disadvantages of Distributed Systems

The disadvantages of distributed systems compared to single computers are:

- Coordination overhead
- Communication overhead
- Dependency on networks
- Higher program complexity
- Security issues

Coordination Overhead

Distributed systems do not have central entities that coordinate their members. Hence, the coordination must be done by the members of the system themselves. Coordinating work among coworkers in a distributed system is challenging and costs effort and computing power that cannot be spent on the genuine computing task, hence, the term coordination overhead.

Communication Overhead

Coordination Overhead

Distributed systems do not have central entities that coordinate their members. Hence, the coordination must be done by the members of the system themselves. Coordinating work among coworkers in a distributed system is challenging and costs effort and computing power that cannot be spent on the

processing of messages, which in turn costs effort and computing power that cannot be spend on the genuine computing task, hence, the term communication overhead.

Dependencies on Networks

Any kind of communication requires a medium. The medium is responsible for transferring information between the entities communicating with one another. Computers in distributed systems communicate by means of messages passed through a network. Networks have their own challenges and adversities, which in turn impact the communication and coordination among computers that form a distributed system. However, without any network, there will be no distributed system, no communication, and therefore no coordination among the nodes, thus the dependency on networks.

Higher Program Complexity

Solving a computation problem involves writing programs and software. Due to the disadvantages mentioned previously, any software in a distributed system has to solve additional problems such as coordination, communication, and utilizing of networks. This increases the complexity of the software.

Security Issues

Communication over a network means sending and sharing data that are critical for the genuine computing task. However, sending information through a network implies security concerns as untrustworthy entities may misuse the network in order to access and exploit information. Hence, any distributed system has to address security concerns. The less restricted the access to the network over which the distributed nodes communicate is, the higher the security concerns are for the distributed system.

Distributed Peer-to-Peer Systems

Communication over a network means sending and sharing data that are critical for the genuine computing task. However, sending information through a network implies security concerns as untrustworthy entities may misuse the network in order to access and exploit information. Hence, any distributed system has to address security concerns. The less restricted the access to the network over which the distributed nodes communicate is, the higher the

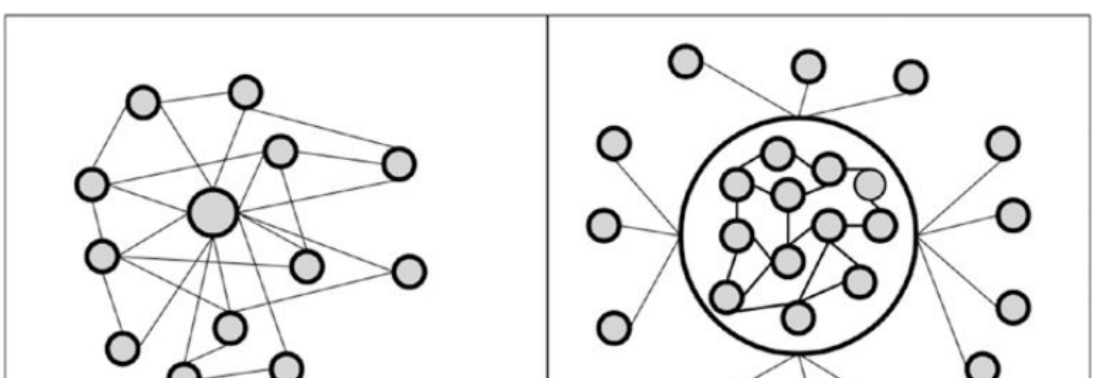
any central point of coordination. The nodes in the network are equal concerning their rights and roles in the system. Furthermore, all of them are both suppliers and consumers of resources.

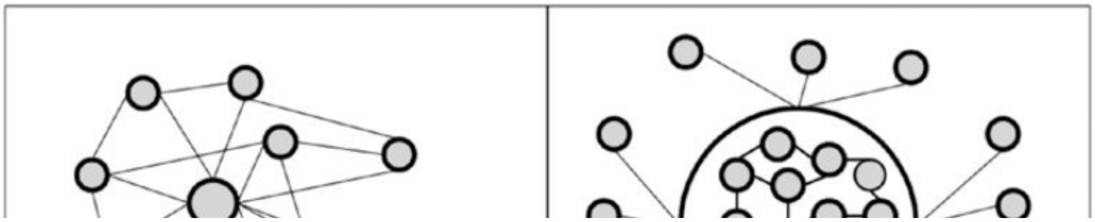
Peer-to-peer systems have interesting applications such as file sharing, content distribution, and privacy protection. Most of these applications utilize a simple but powerful idea: turning the computers of the users into nodes that make up the whole distributed system. As a result, the more users or customers use the software, the larger and more powerful the system becomes. This idea, its consequences, and it challenges are discussed in the following steps.

Mixing Centralized and Distributed Systems

Centralized and distributed systems are architectural antipodes. Technical antipodes have always inspired engineers to create hybrid systems that inherit the strength of their parents. Centralized and distributed systems are no exception to this. There are two archetypical ways of combining these antipodes, and they need to be understood since they will become important when learning about blockchain applications in the real world. They are centrality within a distributed system and the distributed system inside the center.

The graphic on the left-hand side of Figure 2-2 illustrates an architecture that establishes a central component within a distributed system. On first glance, the components seem to form a distributed system. However, all of the circles are connected with the larger circle located in the middle. Hence, such a system only appears to be distributed on a superficial view, but it is a centralized system in reality.





The graph on the right-hand side of Figure 2-2 illustrates the opposite approach. Such a system appears to be a centralized system on first glance, because all the circles in the periphery only have one direct connection to a large central component. However, the central component contains a distributed system inside. The components in the periphery may not even be aware of the distributed system that lives within the central component.

What these two approaches have in common is that it is hard to determine their true nature. Are they distributed or centralized? It may not be necessary to give these architectures unique names. However, it is important to point out their dual nature. This is particularly important because it may not be easy to spot the centrality or the distributed nature within them. I will come back to this point later when I discuss the way the blockchain is commercialized.

Identifying Distributed Systems

The emergence of hybrid architectures makes it hard to identify distributed systems clearly. Formulating a generally accepted definition of distributed systems is beyond the scope of this book. However, for the course of this book it is important to have an idea of what a distributed system is and how it differs from other software systems. If you are in doubt whether or not a system is distributed, look for a single component (e.g., a database, a name or user registry, a login or logoff component, or an emergency switch-off button) that could terminate the whole system. If you find such a component, the system under consideration is not distributed.

Note If one single component exists, e.g., a single switch-off button that can bring down the whole system, then the system is not distributed.

The Purpose of the Blockchain

When designing a software system, one can choose which architectural style will be used, similar to choosing an engine for a car. The architectural decision can be done independently from the functional aspects of the application

Note If one single component exists, e.g., a single switch-off button that can bring down the whole system, then the system is not distributed.

The Purpose of the Blockchain

nonfunctional aspects of a system. In particular, both architectural concepts have very different approaches to ensure integrity. And this is the point where the blockchain enters the picture. The blockchain is a tool for achieving integrity in distributed software systems. Hence, it can be seen as a tool to achieve a nonfunctional aspect of the implementation layer.

Note The purpose of the blockchain is to achieve and maintain integrity in distributed systems.

Outlook

Achieving integrity in a distributed system is very technical and it may sound a bit boring. However, the question that makes this achievement exciting for many people depends on what the distributed system will do and what kind of centralized system it replaces. The next step explains how a peer-to-peer system has changed our world and why the blockchain as a tool for achieving integrity in distributed software systems has the potential to change the world too.

Summary

- The architecture of a software system determines how its components are organized and related to one another.
- Centralized and distributed software architectures can be seen as antipodes.
- A distributed system consists of a number of independent computers that cooperate with one another by using a communication medium in order to achieve a specific objective without having any centralized element of control or coordination.
- As a rule of thumb, one can state that as soon as a system has a single component that could bring down the whole system it is not distributed, regardless of how complex its architecture looks.

be seen as antipodes.

- A distributed system consists of a number of independent computers that cooperate with one another by using a communication medium in order to achieve a specific objective without having any centralized element of control or coordination.

STEP

3

Recognizing the Potential

How peer-to-peer systems may change the world

This step deepens our understanding of the purpose of the blockchain by considering a specific kind of distributed system: the peer-to-peer system. As a result, this step will help you understanding why there is so much excitement about the blockchain among technologists and business professionals alike. This step also points out the major area of application in which the blockchain is expected to provide the most value. Additionally, this step discusses some consequences of peer-to-peer systems in the real world.

The Metaphor

Can you remember the last time you bought a CD for yourself in a music store or in a department store? Most people have not bought actual CDs for result, this step will help you understanding why there is so much excitement about the blockchain among technologists and business professionals alike. This step also points out the major area of application in which the blockchain is expected to provide the most value. Additionally, this step discusses some consequences of peer-to-peer systems in the real world.

that allowed people to share their music files with one another. But what was so special about that software? This is what one of its inventors had to say about this:

> *This system, what's most interesting about it is, you're interacting with peers, you're exchanging information with a person down the street.*
>
> —Shawn Fanning, cofounder of Napster

What Fanning and his coworkers invented was a peer-to-peer system for sharing music. Back in the late 1990s, this software ushered in a new era for the established business model of the music industry. This step explains what the emergence of Napster, the decline of CD sales, and the dramatic changes of the music industry have to do with the blockchain.

How a Peer-to-Peer System Changed a Whole Industry

The music industry has worked for a long time in the following way: musicians made contracts with studios, which recorded the songs, produced and marketed the music records on a variety of media (e.g., vinyl, tape, or CD), which in turn were sold to the customers via a variety of distribution channels, including department stores and specialized shops. The studios actually worked as intermediaries between musicians and people who enjoy listening to music. Music studios could maintain their role as intermediaries due to their exclusive knowledge and skills in producing, marketing, and distributing records. However, in the first decade of the 2000s, the environment in which the music studios operated changed dramatically.

The digitalization of music, the availability of recording equipment at affordable prices, the growing spread of privately used PCs, and the emergence of the Internet made music studios dispensable. The three functions of music studios—producing, marketing, and distributing records—could be done by the artists and the consumers themselves. Napster played a major role in the replacement of the music studios as intermediaries. With Napster, people no longer relied on the music studios to get the latest hits. It was possible to their exclusive knowledge and skills in producing, marketing, and distributing records. However, in the first decade of the 2000s, the environment in which the music studios operated changed dramatically.

The digitalization of music, the availability of recording equipment at affordable prices, the growing spread of privately used PCs, and the emergence of the Internet made music studios dispensable. The three functions of music

a digital sharing bazaar for mp3 files, gave consumers access to a wider range of music than ever before, making the music studios partly dispensable and causing them significant losses.[1]

The Potential of Peer-to-Peer Systems

The Napster case taught us that peer-to-peer systems have the potential to reshape whole industries based on a simple idea: replacing the middleman with peer-to-peer interactions. In the case of the music industry, the traditional studios and their marketing and distribution channels that acted as the middlemen between artists and consumers have been replaced by peer-to-peer file sharing systems. The major characteristics that made the music industry so vulnerable to being replaced by peer-to-peer systems are the immaterial nature of music and the low costs of copying and transferring data.

The power of peer-to-peer systems is not restricted to the music industry. Each industry that mainly acts as a middleman between producers and customers of immaterial or digital goods and services is vulnerable to being replaced by a peer-to-peer system. This statement may sound a bit abstract, but you may discover many middlemen for immaterial and digital goods and services around you once you recognize the largest of them all: the financial industry.

What is it that you have in your bank account or on your credit or debit card? Is it really money? The money you own has been turned into immaterial bits and bytes long ago. Only a small amount of actual money exists as physical banknotes and coins. The vast majority of the world's money and assets exists as immaterial bits and bytes in the centralized information technology systems of the financial industry. Banks and many other players of the financial industry are just middlemen between producers and consumers of bits and bytes that make up our money and our wealth. The act of borrowing, lending, or transferring money from one account to another is just the transfer of an immaterial good operated by middlemen, also called intermediaries. It is amazing how many middlemen are involved in seemingly simple transactions (e.g., transferring money from one bank account to another one in a different country involves up to five middlemen, which all need their processing time and impose their own fees). As a result, something as simple as transferring an amount of money from one bank account to another in a different country involves a long of the financial industry. Banks and many other players of the financial industry are just middlemen between producers and consumers of bits and bytes that make up our money and our wealth. The act of borrowing, lending, or transferring money from one account to another is just the transfer of an immaterial good operated by middlemen, also called intermediaries. It is amazing how many middlemen are involved in seemingly simple transactions (e.g., transfer-

processing time and incurs high transactions costs. In a peer-to-peer system, the same transfer would be much simpler and it would take less time and costs since it could be processed as what it is: a transfer of bits and bytes between two peers or nodes, respectively.

The advantage of peer-to-peer systems over centralized systems is that direct interactions occur between contractual partners instead of indirect interactions through a middleman, hence, there is less processing time and lower costs.

The advantages of peer-to-peer systems are not restricted to money transfer. Every industry that mainly acts as a middleman between producers and customers of immaterial or digital goods and services is vulnerable to being replaced by a peer-to-peer system. As digitalization continues, more and more items of everyday life and an increasing amount of goods and services will become immaterial and will benefit from the efficiencies of peer-to-peer systems. Advocates of peer-to-peer systems argue that almost all aspects of our life will be affected by the emergence of digitalization and peer-to-peer networks such as payments, money saving, loans, insurance, as well as issuance and validation of birth certificates, driving licenses, passports, identity cards, educational certificates, and patents and labor contracts. Most of them already exist in digital form in centralized systems run by institutions that are nothing other than a middleman between natural suppliers and customers.

Note Replacing the middleman is also called disintermediation. It is considered a serious threat to many business and companies that mainly act as intermediaries between different groups of people, such as buyers and seller, borrowers and lenders, or producers and consumers.

Terminology and the Link to the Blockchain

Now that you have learned about the potential of peer-to-peer systems, it is necessary to clarify the terminology of the problem domain and to explain its relation to the blockchain. In particular, the following points need to be discussed:

of people, such as buyers and seller, borrowers and lenders, or producers and consumers.

Terminology and the Link to the Blockchain

Now that you have learned about the potential of peer-to-peer systems, it is

The Definition of a Peer-to-Peer System

Peer-to-peer systems are distributed software systems that consist of nodes (individual computers), which make their computational resources (e.g., processing power, storage capacity, or information distribution) directly available to another. When joining a peer-to-peer system, users turn their computers into nodes of the system that are equal concerning their rights and roles. Although users may differ with respect to the resources they contribute, all the nodes in the system have the same functional capability and responsibility. Hence, the computers of all users are both suppliers and consumers of resources.[2]

For example, in a peer-to-peer file sharing system, the individual files are stored on the users' machines. When someone wants to download a file in such a system, he or she is downloading it from another person's machine, which could be the next door neighbor or someone located halfway around the world.

Architecture of Peer-to-Peer Systems

Peer-to-peer systems are distributed computer systems by construction since they are made of individual nodes that share their computational resources among others. However, there are also peer-to-peer systems that still utilize elements of centralization. Centralized peer-to-peer systems maintain central nodes to facilitate the interaction between peers, to maintain directories that describe the services offered by the peer nodes, or to perform look-ups and identification of the nodes.[3] Centralized peer-to-peer systems typically utilize a hybrid architecture, such as the one that was illustrated on the left-hand side of Figure 2-2. Such architecture allows combining the advantages of centralized and distributed computing. On the other hand, purely distributed peer-to-peer systems do not have any element of central control or coordination. Hence, all nodes in those systems perform the same tasks, acting both as providers and consumers of resources and services.

An example of a centralized peer-to-peer system is Napster, which maintained a central database of all nodes connected with the system and the songs available on these nodes.

identification of the nodes. Centralized peer to peer systems typically utilize a hybrid architecture, such as the one that was illustrated on the left-hand side of Figure 2-2. Such architecture allows combining the advantages of centralized and distributed computing. On the other hand, purely distributed peer-to-peer systems do not have any element of central control or coordination. Hence, all nodes in those systems perform the same tasks, acting both as providers and consumers of resources and services.

The Link Between Peer-to-Peer Systems and the Blockchain

As discussed in Step 2, the blockchain can be considered a tool for achieving and maintaining integrity in distributed systems. Purely distributed peer-to-peer systems may use the blockchain in order to achieve and to maintain system integrity. Hence, the link between purely distributed peer-to-peer systems and the blockchain is its usage for achieving and maintaining integrity in purely distributed systems.

The Potential of the Blockchain

The relation between purely distributed peer-to-peer systems to the blockchain is that the former uses the latter as a tool to achieve and maintain integrity. Hence, the argument that explains the excitement about and the potential of the blockchain is: Purely distributed peer-to-peer systems have a huge commercial potential as they can replace centralized systems and change whole industries due to disintermediation. Since purely distributed peer-to-peer systems may use the blockchain for achieving and maintaining integrity, the blockchain becomes important as well. However, the major fact that excites people is the disintermediation. The blockchain is only a means to an end that helps to achieve that.

Note The excitement about the blockchain is based on its ability to serve as a tool for achieving and maintaining integrity in purely distributed peer-to-peer systems that have the potential to change whole industries due to disintermediation.

Outlook

This step explained what peer-to-peer systems are and highlighted their potential to change whole industries due to disintermediation. Additionally, this step pointed out that the excitement about the blockchain is due to its ability to serve purely distributed peer-to-peer systems to fulfill their tasks and maintaining integrity in purely distributed peer-to-peer systems that have the potential to change whole industries due to disintermediation.

Outlook

Summary

- Peer-to-peer systems consist of computers, which make their computational resources directly available to another.
- The advantage of peer-to-peer systems is their ability to allow users to interact directly with one another instead of interacting indirectly through middlemen.
- Replacing middlemen with peer-to-peer systems increases processing speed and reduces costs.
- Peer-to-peer systems can be centralized or purely distributed.
- Purely distributed peer-to-peer systems form a network of equal members that interact directly with one another without having any central coordination.
- Napster demonstrated the power of peer-to-peer systems as its file sharing system ushered in a new era for the business model of the traditional music industry, which mainly acted as a middleman between artists and consumers.
- Every industry that mainly acts as a middleman between producers and customers of immaterial or digital goods and services is vulnerable to being replaced by peer-to-peer systems.
- A huge part of our financial system is simple intermediation between suppliers and consumers of money, which mainly exists as digital or immaterial good. Hence, digitalization and peer-to-peer systems may reshape the financial industry in a similar fashion as Napster reshaped the music industry.
- As digitalization continues, more aspects of our everyday lives and an increasing amount of goods and services will

peer systems.

- A huge part of our financial system is simple intermediation between suppliers and consumers of money, which mainly exists as digital or immaterial good. Hence, digitalization and peer-to-peer systems may reshape the financial industry in a similar fashion as Napster reshaped the music industry.
- As digitalization continues, more aspects of our everyday lives and an increasing amount of goods and services will become immaterial and will benefit from the advantages of peer-to-peer systems.
- The excitement about the blockchain is based on its ability to serve as a tool for achieving and maintaining integrity in purely distributed peer-to-peer systems that have the potential to change whole industries due to disintermediation.