

Northeastern University

INFO 6255

Fall 2020

[m.Servattalab@northeastern.edu](mailto:m.Servattalab@northeastern.edu)



# What Is Cloud Computing

- **Cloud Computing** means storing and accessing data and programs over the Internet instead of your computer's hard drive. The cloud is just a metaphor for the Internet.
- It is the act of using an interconnected network of Internet-hosted remote servers to manage, store and process data.
- It is used for everything from banking to data sharing and, because it allows multiple computers on a local network to access data, it has become a popular form of technology among companies of all sizes.



# Uses of Cloud Computing

A Cloud allows an online service to send email, edit documents, watch movies or TV, listen to music, play games, or store pictures and other files.

- **Create cloud-native applications**

- Quickly build, deploy, and scale applications—web, mobile, and API. Take advantage of cloud-native technologies and approaches, such as containers, Kubernetes, microservices architecture, API-driven communication, and DevOps.

- **Store, back up, and recover data**

- Protect your data more cost-efficiently—and at massive scale—by transferring your data over the Internet to an offsite cloud storage system that's accessible from any location and any device.

- **Test and build applications**

- Reduce application development cost and time by using cloud infrastructures that can easily be scaled up or down.

- **Analyze data**

- Unify your data across teams, divisions, and locations in the cloud. Then use cloud services, such as machine learning and artificial intelligence, to uncover insights for more informed decisions.

# Cloud Deployment Models

A cloud deployment model represents a specific type of cloud environment, which are primarily distinguished by size, access and ownership.

There are 3 common cloud deployment models



Hybrid Cloud



Private Cloud



Public Cloud



# Public Cloud

• **Public Cloud:** are owned and operated by a third-party cloud service providers which deliver their computing resources, like servers and storage, over the internet.

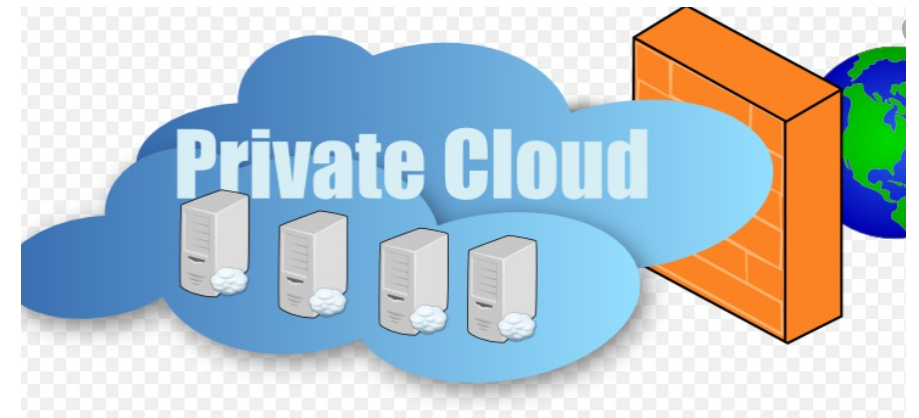
examples

- Microsoft Azure, Amazon Web Services, Salesforce are examples of public cloud.
- All hardware, software, and other supporting infrastructure is owned and managed by the Cloud provider.
- Anyone can access these services and manage your account using a web browser.

# Private Cloud

**Private Cloud:** Cloud computing resources are used exclusively by a single business or organization.

- It can be physically located on the company's on-site data center. Some companies also pay third-party service providers to host their private cloud.
- In **private cloud** the services and infrastructure are maintained on a private network.



# Hybrid Cloud

**Hybrid Cloud:** A hybrid cloud is a combination of private cloud and the public cloud services allowing data and applications to move between private and public clouds.

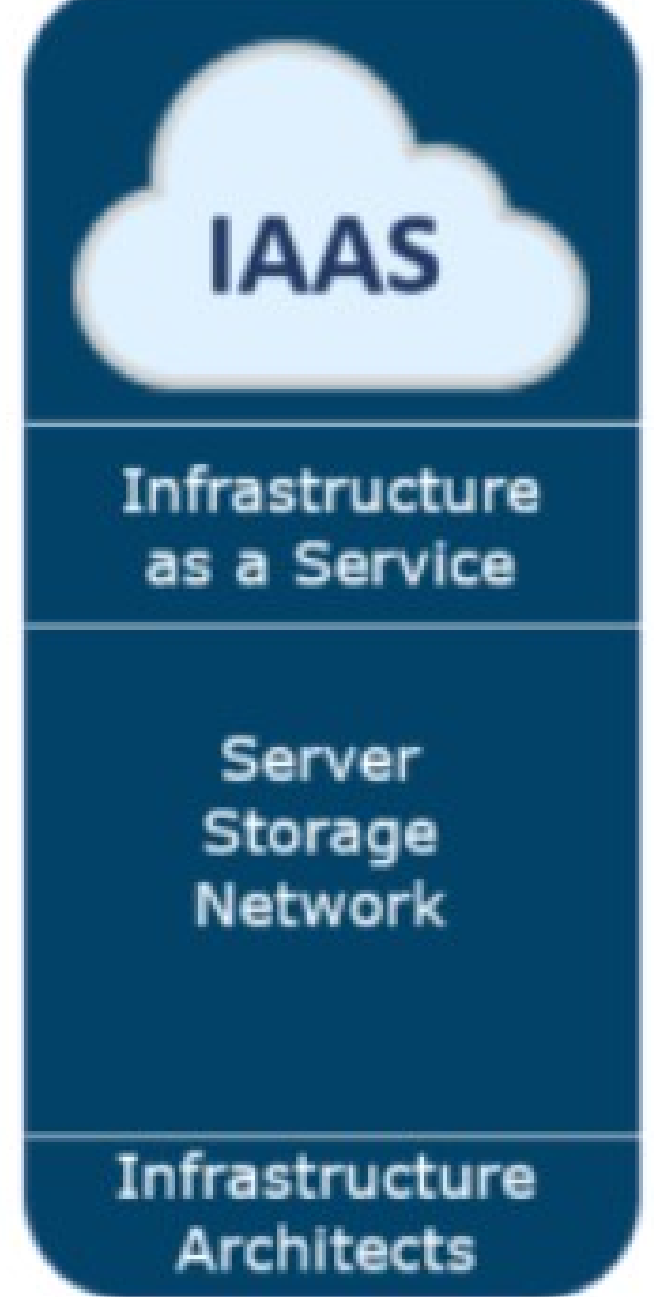
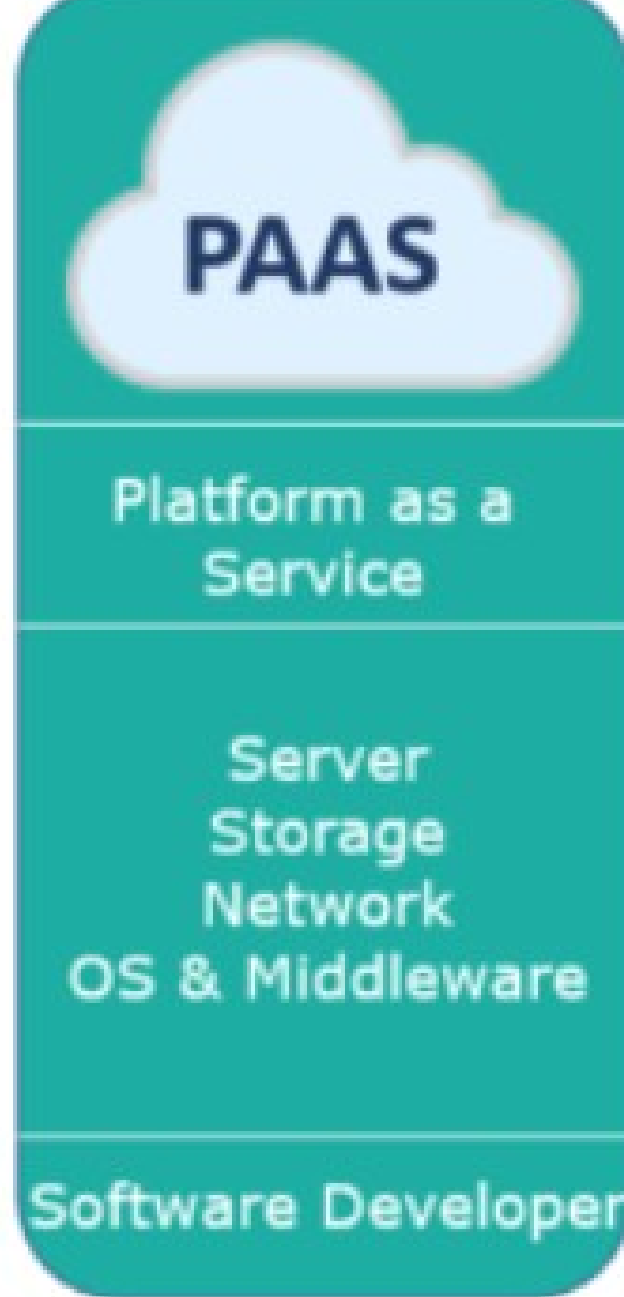
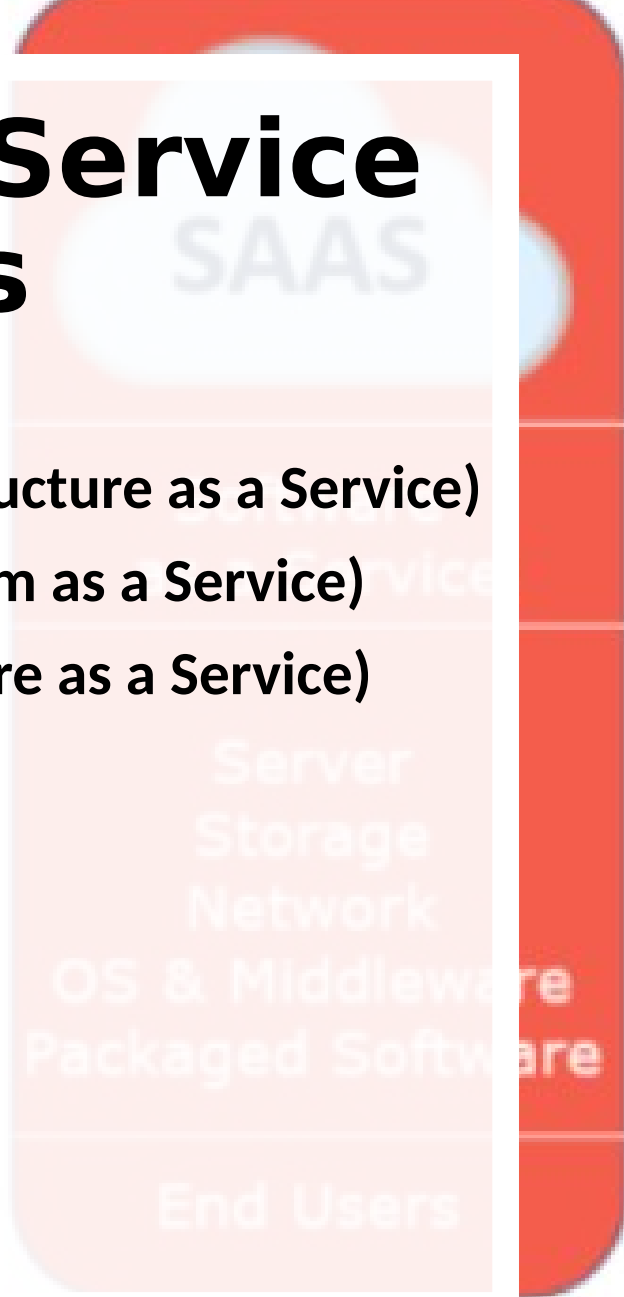
优点

This model gives businesses greater flexibility and more deployment options.



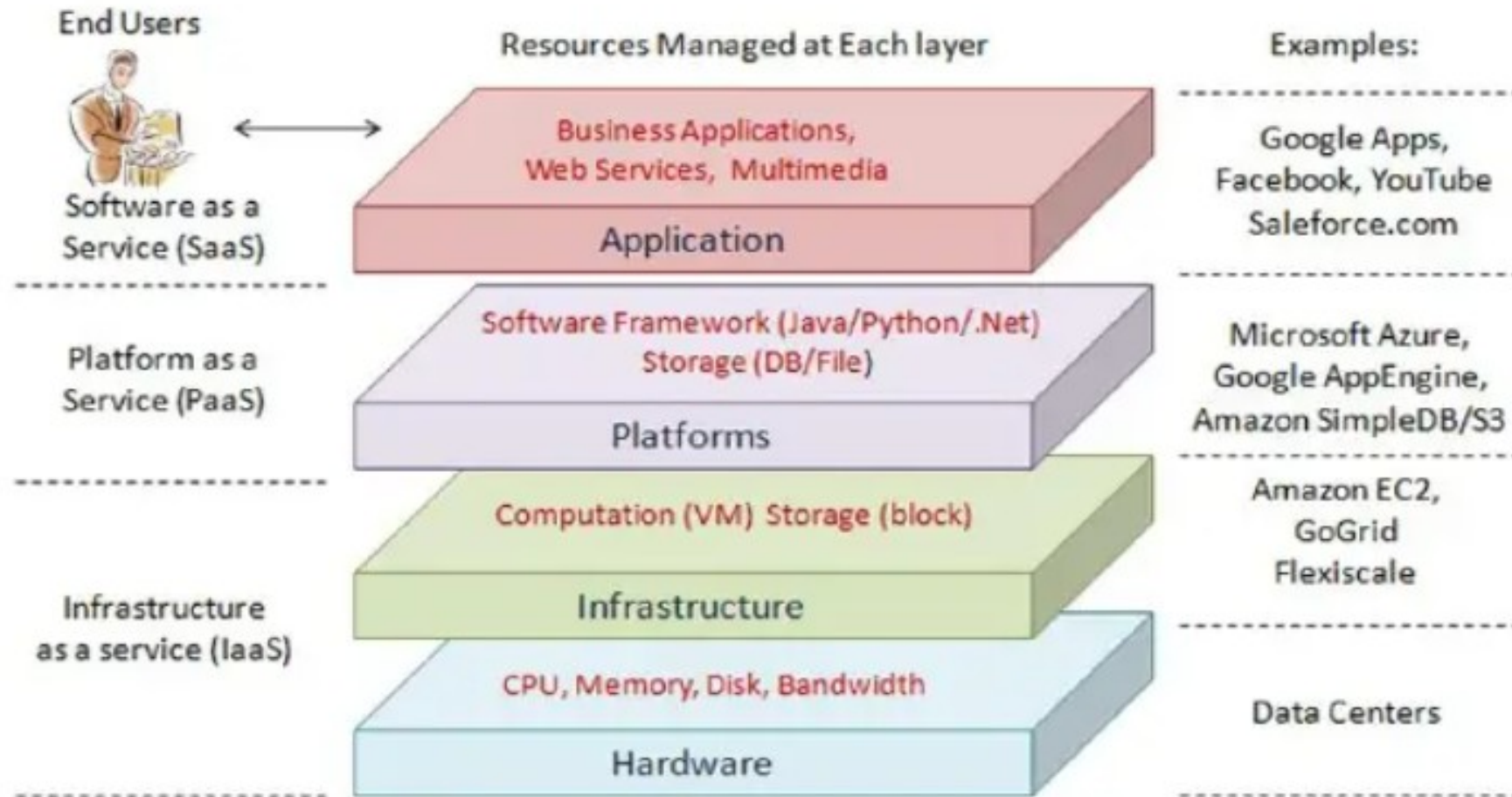
# Cloud Service Models

- IaaS (Infrastructure as a Service)
- PaaS (Platform as a Service)
- SaaS (Software as a Service)





# Cloud Layer Architecture



# Cloud Service Models

**IaaS (Infrastructure as a Service):** Vendor provides users access to computing resources such as servers, storage and networking. Organizations use their own platforms and applications within a service provider's infrastructure.

## Key Features:

- Infrastructure is scalable depending on processing and storage needed
- Teams can quickly set up and dismantle test and development environments, bringing new applications to market faster.
- It is a quick and economical way to scale up dev-test environments up and down.
- Saves businesses the cost of purchasing and maintaining their own hardware. Users pay for IaaS on demand.
- Organizations avoid the capital outlay for storage and complexity of storage management, which typically requires a skilled staff to manage data and meet legal and compliance requirements.



# Cloud Service Models...

**PaaS (Platform as a Service):** Provides users a cloud environment in which they can develop, manage and deliver applications.

It is an IaaS Service offering, plus middleware, development tools, business intelligence (BI) services and database management systems.

## Key Features:

- PaaS is designed to support the complete web application lifecycle: building, testing, deploying, managing, and updating.
- Providers manage security, operating systems, server software and backups.
- Facilitates collaborative work even if the teams are working remotely.



# Cloud Service Models...

**SaaS (Software as a Service):** Provides users with access to a vendor's cloud-based software. Users do not install applications on their local devices, instead cloud providers host and manage the software application and underlying infrastructure, and handle maintenance like upgrades and system patching.

## Key Features:

- SaaS makes sophisticated enterprise applications, such as ERP and CRM, affordable for organizations that lack the resources to buy, deploy, and manage the required infrastructure and software themselves.
- You also save money because the SaaS service automatically scales up and down according to the level of usage.
- Data is secure in the cloud; equipment failure does not result in loss of data.



# Cloud Based Testing

- **Cloud-based testing** can be applied for testing cloud, web, and installed applications.
- Providers of cloud testing services and tools offer test environments that can be configured according to application's requirements.
- Cloud testing has given rise to **Testing as a Service (TaaS)**, which allows organizations to outsource their testing efforts.
- **TaaS** can be used for overall software testing as well as for conducting specialized types of testing such as performance, security, or functional testing.
- Cloud testing focuses on the **core components** like:
  - **Application:** It covers testing of functions, end-to-end business workflows, data security, browser compatibility, etc.
  - **Network:** It includes testing various **network bandwidths, protocols** and successful transfer of data through networks.
  - **Infrastructure:** It covers **disaster recovery test, backups, secure connection, and storage policies.** The infrastructure needs to be validated for regulatory compliances

# Benefits of Cloud-Based testing

In contrast to traditional software testing, cloud-based testing has several unique advantages:

1. **Scalability:** Cloud computing allows testers to increase or decrease computing resources according to their needs.
  - This is very useful in cases when the client frequently **changes their business requirements**.
2. **Cost-Cutting:** In cloud computing, you pay only for those resources that you use.
  - This means that there's no need to invest in **expensive equipment and spend money** maintaining and upgrading it.
  - There can be all the **software and hardware you might need at your disposal** while only paying for it when you actually use it.
3. **Easily Customizable:** By using cloud-based tools and services, testers can easily emulate an end-user-centric environment with minimum cost and time.
  - The test team can perform various types of testing in any combination of device environments.



# Benefits of Cloud-Based Testing

4. **Ensure Comprehensive Testing:** For comprehensive testing, the test team needs to run an application on all possible devices that support different platforms, OS, and browsers.
  - Cloud-based testing provides you with all these devices and configurations, eliminating the need to purchase all of them.
5. **Faster Testing:** Cloud-based testing tools ensure automated testing, which greatly reduces the time to market for software.
  - This is achieved by the ease of building testing infrastructure, increased collaboration within the test team.
6. **Constant Availability:** Software testing in the cloud is available to testers at any time and anywhere.
  - So testers can speed up software deployment and testing.

# Types of Testing in the Cloud

The Cloud Testing is segmented into four main categories:

1. **Testing of the whole cloud:** The cloud is viewed as a whole entity and based on its features testing is carried out.
  - Cloud and SaaS vendors, as well as end users, are interested in carrying out this type of testing
2. **Testing within a cloud:** By checking each of its internal features, testing is carried out.
  - Only cloud vendors can perform this type of testing
3. **Testing across cloud:** Testing is carried out on different types of cloud-like private, public and hybrid clouds.
4. **SaaS testing in cloud:** Functional and non-functional testing is carried out on the basis of application requirements.



Testing Type	Description
Performance Testing	While cloud solutions should <u>be scalable on demand</u> , this type of testing ensures that the application performs correctly with various numbers of users.
Security Testing	<p>This type of testing is necessary for ensuring that data is stored and transmitted safely. Security mechanisms of applications are tested according to three criteria: <b>effectiveness, accuracy, and performance</b>.</p> <p>The most popular tools for testing security in the cloud are <b>Nessus, Wireshark and Nmap</b>.</p>
Functional Testing	<p>Functional software testing checks <u>all the features and functions</u> of software and its interaction with hardware.</p> <p>For conducting functional testing, testers can use such tools as <b>TimeShiftX, Rapise and Sauce Labs</b></p>
Load & Stress Testing	<p>During load testing, testers measure software response time while the system is subjected to increasing load. It's also necessary to check how the application will work under <b>excessive stress</b>.</p> <p>If you want to measure the application response delay after deploying it in the cloud, then you can <u>conduct latency testing</u>.</p>
Interoperability & Compatibility Testing	<p><b>It is carried out to validate</b> the compatibility requirements of the application under test system. <u>It checks browser compatibility in a cloud environment</u>. It identifies the <u>Defects that might arise while connecting to a cloud</u>.</p> <p>It is carried out to verify if application works across a different platform of cloud.</p>

# Top 10 Cloud-based Software Testing tools

Testing in the cloud brings with it benefits of easy availability, high scalability and low cost.

It allows for web and mobile testing in different environments and multiple machines without building your own infrastructure.

Here are some of the hugely popular cloud-based software testing tools:

1. SOASTA CloudTest
2. LoadStorm
3. BlazeMeter
4. Nessus
5. App Thwack
6. Jenkins Dev@Cloud
7. Xamarin Test Cloud
8. TestLink
9. Test collab
10. Test Sigma



# Cloud Testing vs. Conventional Testing

Testing Parameters	Conventional Testing	Cloud Testing
Primary Testing Objective	Check interoperability, compatibility, usability. <ul style="list-style-type: none"><li>Verifies the quality of system function and performance based on the given specification.</li></ul>	Verifies the quality of performance and functions of SaaS, Clouds, and applications by leveraging a cloud environment.
Testing Costs	Costing <u>remains high</u> due to hardware and software requirements.	Only have to pay for operational charges <ul style="list-style-type: none"><li><u>Pay only for what is being used.</u></li></ul>
Functional Testing	Validating functions (unit and system) as well as its features.	Testing end-to-end application function on SaaS or Cloud.
Testing Environment	A pre-fixed and configured test environment in a test lab.	An open public test environment with diverse computing resources.
Integration Testing	Component, architecture, and function-based testing.	SaaS-based Integration Testing.
Security Testing	Testing security features based on process, server and privacy.	Testing security features based on cloud, SaaS and real time tests in vendor's cloud.
Performance and Scalability Testing	Performed in a fixed test environment.	Apply both real time and virtual online test data.

# References

- <https://www.apriorit.com/dev-blog/548-cloud-based-testing>
- <https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/#cloud-computing-models>
- <https://www.investopedia.com/terms/c/cloud-computing.asp>

# Testing Blockchain Transactions & Cryptocurrency

Northeastern University

INFO 6255

Fall 2020

[m.Servattalab@northeastern.edu](mailto:m.Servattalab@northeastern.edu)



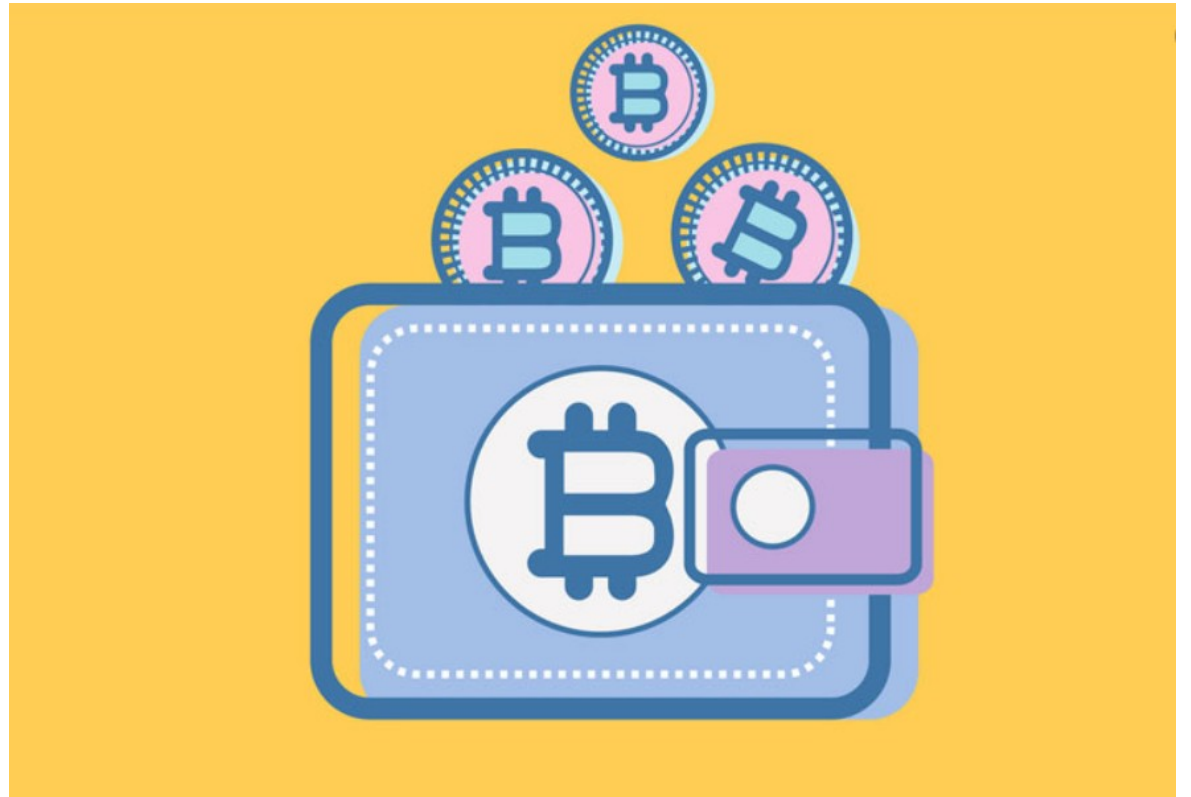
# What is Blockchain?

- Blockchain is a decentralized data source.
- Blockchains rely on data being verified by a number of different sources controlled by different entities, instead of a central repository.
- Information is confirmed by multiple disparate sources, and each piece of information builds on the previous one.



# What is Blockchain?

- One example: **Bitcoin**
  - The blockchain is a ledger of transactions detailing how currency has moved from one address to another, with multiple versions being updated simultaneously and continuously by multiple parties.





# Testing Digital Currency Constraints

- When testing **digital currency** transactions on a blockchain, there are some important constraints:
  - You cannot manipulate or change data to create certain outcomes
  - There are limited options for testing without real assets
  - It's difficult to control the timing of your transactions





# Testing Digital currency Constraints...



- Unable to manipulate data for testing to create different situations or create different errors.
- When you're testing on a public blockchain, you have no control over where crucial transaction data is sourced or stored

# Testing Digital Currency Constraints...

- There are two separate bitcoin blockchains to consider for testing:
  - The **mainnet chain**: It is the real bitcoin chain. Transactions here utilize the genuine digital currency
  - The **testnet chain**: It's transactions function in the same way, except these are test currency and are never supposed to have any value.
    - **Testnet** can be a great way to test transactions.
  - You can send as much as you want, and there's no real cost to your organization.
  - You still need to get the funds. One of the easiest ways to acquire **testnet** funds is to use a Faucet.
  - **Faucets** are websites set up by other Bitcoin developers with funds that can be shared



# Testing Digital Currency Constraints...

- When you're setting up tests for blockchain transactions, you may need to **be prepared to wait**.
  - Block confirmations take time.
- One way to try to get your transactions **picked up** by the network more quickly is to send higher amounts of money
  - You'll be charged a **higher fee** for sending a larger amount
- Another situation where timing can be a problem is when your wallet syncs directly with the **public blockchain**
  - Tests that involve a full sync require advanced planning.



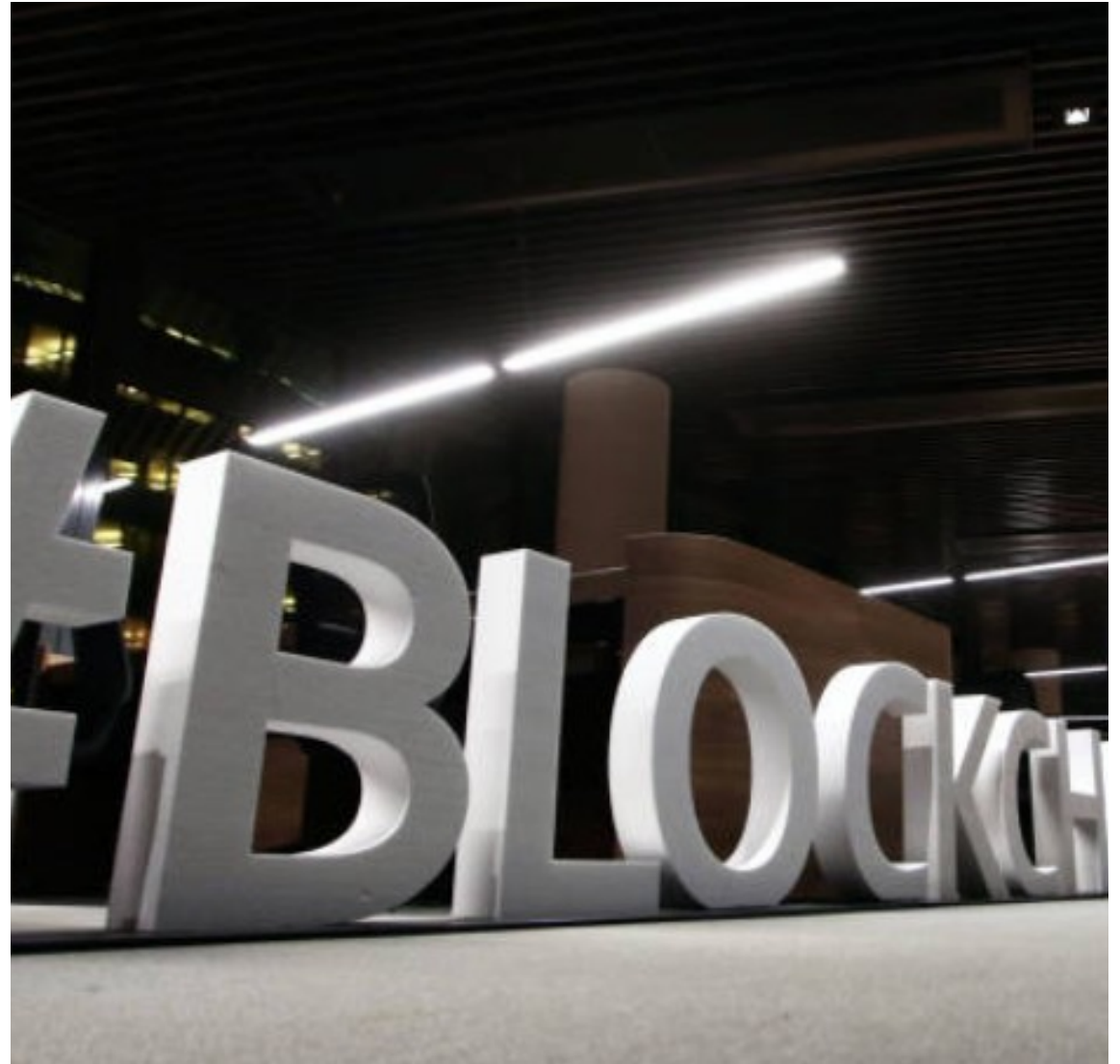
# Crypto Wallet Transaction Verification

**Block Explorer:** is a website that allows you to search for transactions on the blockchain.

- You can search and quickly find a number of reliable options; many are set up for a variety of blockchains.

## **CryptoCurrenty Addresses:**

- Crypto investors have the ability to transact with anonymity.
- Your crypto address is created by the wallet and tied to a private key that is stored on your device.
- Your address is available publicly but not tied to any of your personal identifying information.

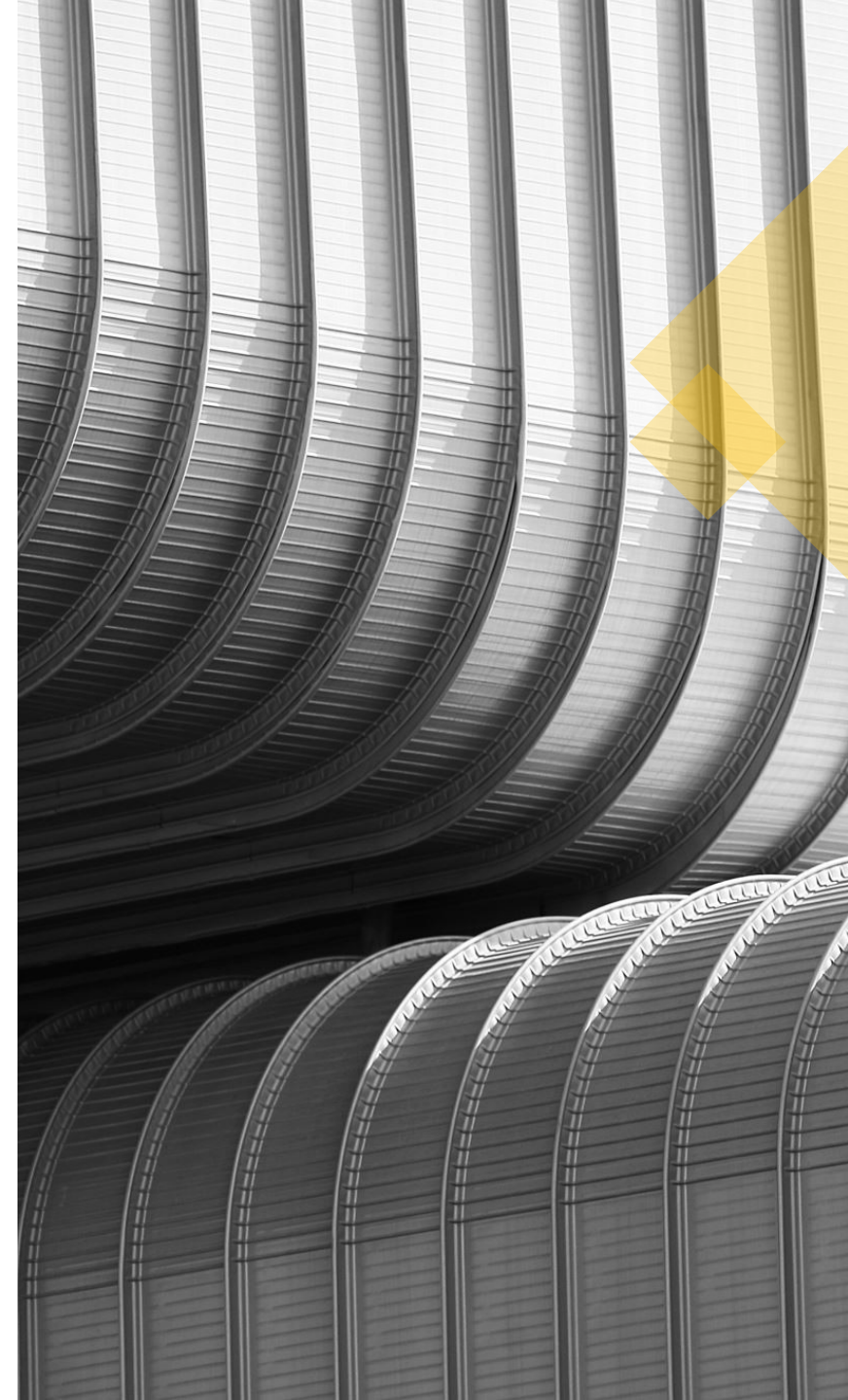




# Crypto wallet transaction Verification...

To ensure that the transaction has been registered properly on the blockchain with the correct value:

- **Transaction Accuracy:**
  - Verify transactions in three locations:
    - The sending wallet
    - The receiving wallet
    - On the blockchain
- **Block confirmations**
  - Blockchain transactions need to be confirmed before they can be finalized.
  - Some wallets may have rules about when to display a transaction as pending or completed, based on the number of block confirmations.



# Crypto wallet transaction Verification...

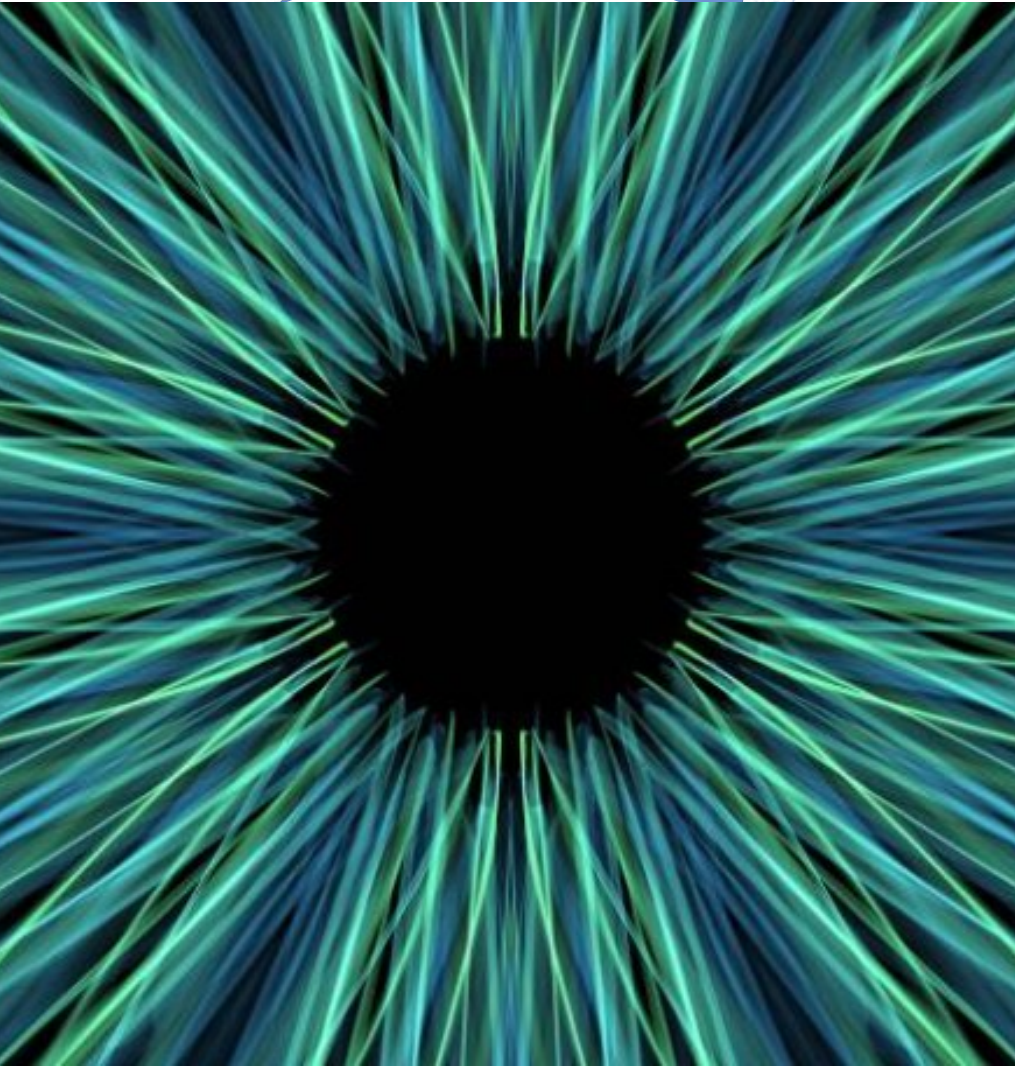
- **Memos or tags**

- **Memos or tags**, are important for certain tokens.
- **For example**, Ripple's digital currency, XRP, uses a destination tag with transactions.
- This identifier helps ensure that transactions on the XRP blockchain are routed correctly.

- **Fees or gas**

- Each digital currency has its own rules and fee structure.
- Fees on the **Ethereum network**. The Ethereum blockchain has a number of currencies that can be exchanged, called ERC20 tokens.
- Unlike Bitcoin, whose fees are paid in bitcoin, any fees on the Ethereum blockchain are paid in **Ethereum**, instead of the token you are sending.





# Summary

- **Blockchain** presents a seemingly never-ending place to learn and explore.
- On the surface, some aspects of this technology are easy to understand, but things get complex very quickly.
- You can learn as much as you can about all the rules of any given chain.
- It can be overwhelming at first, but there are a wealth of online sources

# Reference

- [https://blog.gurock.com/testing-blockchain-cryptocurrency/?utm\\_source=newsletter&utm\\_medium=email&utm\\_campaign=blog-nl-al-em-2020-09-30](https://blog.gurock.com/testing-blockchain-cryptocurrency/?utm_source=newsletter&utm_medium=email&utm_campaign=blog-nl-al-em-2020-09-30)