Optimizing Decentralized Applications

FinTech

Lesson 22.2

Class Objectives

By the end of the class, you will be able to:

Explain what Solidity events are and how they can be used to enhance a dApp.

Use filters in Web3.py to react to events from smart contracts.

Use the InterPlanetary File System (IPFS) to store immutable data off-chain in order to both save gas and ensure the decentralized nature of the dApp.


WELCOME

In the previous lesson…

- You learned more about decentralized applications, or dApps.
- dApps leverage the decentralized structure of the blockchain in order to build financial systems.
- dApps also provide the tools that directly enable these systems.

What are some differences between fungible and non-fungible tokens?

Review of Fungible and Non-fungible Tokens

Some differences:

Fungible tokens	Non-fungible tokens
Not unique	Unique
Interchangeable with one another	Not interchangeable with one another
Fungible tokens use the ERC-20 standard	Non-fungible tokens use ERC-721

What are some examples of non-fungible assets?

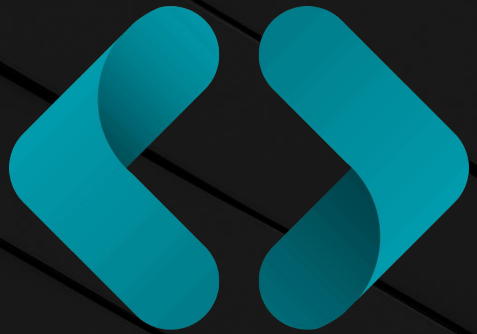
Examples of Fungible Assets

Art

Diamonds

Land ownership

Questions?



Create an Art Registry Contract

Time to <code>

Why is a `struct` a useful data structure?

A `struct` allows you to create new dynamic data structures—data structures that contain multiple types, and objects.

What is the purpose of Solidity events?

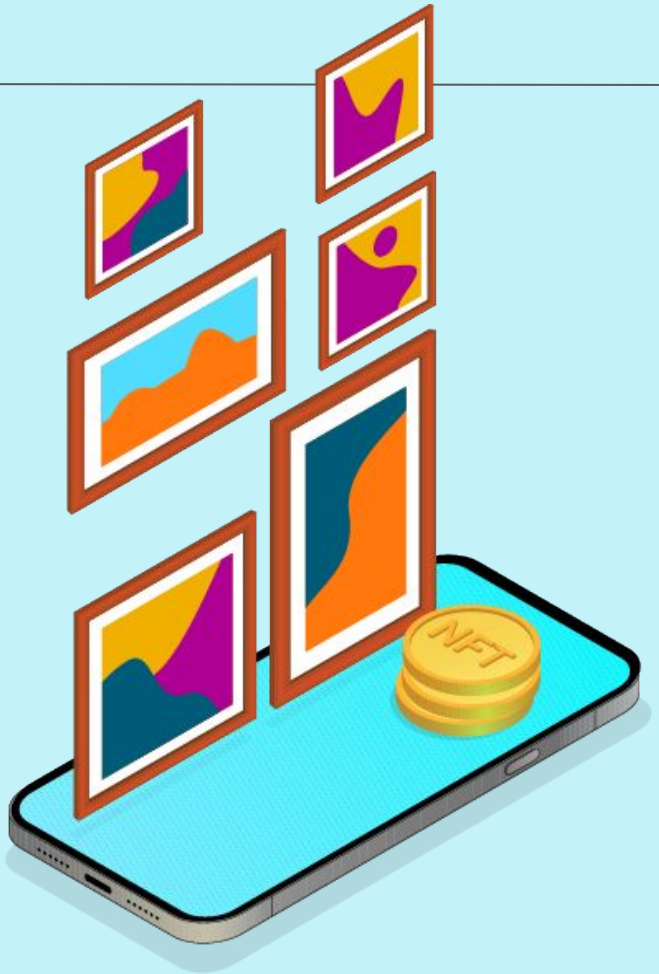
Events are an inexpensive way of logging information on the blockchain.

They allow dApps to update and monitor given values on the blockchain.

Congratulations

You created a smart contract that can both mint new artwork tokens and add new appraisals for each piece of art.

The contract also uses a cost-effective method of logging historical data and external appraisal reports via the Appraisal event that we created.



Next lesson

We'll demonstrate how the Web3.py filter functionality is used by the front end of the dApp to read the event log.

Questions?

Instructor Demonstration

Creating the dApp Filters

Real-World Token Examples

Because our contract can store data in the event log, we need to build a way to access that historical data.

The ERC-721 standard provides the tools needed for managing these events.

Solidity automatically inherits all the code for managing events from the ERC721Full contract.

To access the event data, a filter for the event needs to be created in the front end of a dApp.

Why a filter?

Real-World Token Examples

The reason is that more than one type of event might get logged for the smart contract.

We want to be able to choose the event type that we desire to access.

Web3.py offers the createFilter function that can be used to create the necessary filter.

Questions?

Time to Code

Appraising the Situation

Suggested Time:

15 minutes

Congratulations

You have just built another NFT that is compliant with the ERC-721 standard, complete with on-chain, custom members and several linked token URIs.

You used Solidity events and URIs that allow you to connect both data and a front end to your Solidity smart contract from outside the blockchain.

You created a sophisticated dApp for this contract, but it's not quite finished.

Next lesson

We'll explore how to use decentralized file storage with smart contracts in order to reduce costs while ensuring the integrity of the digital asset.

Instructor Demonstration

Introduction to Decentralized Storage

IPFS stands for
Inter**P**lanetary **F**ile **S**ystem**.**

(ipfs.io)

IPFS: The InterPlanetary File System

01 A protocol,

02 a network,

03 and a filesystem.

But what exactly does this mean, and how does it all fit together?

IPFS: The InterPlanetary File System

For two users to exchange data with one another across the internet, they need a common set of rules for how the information is sent between them; this is a communication protocol.

IPFS: The InterPlanetary File System

Communication protocols are usually built within a stack known as a protocol suite.

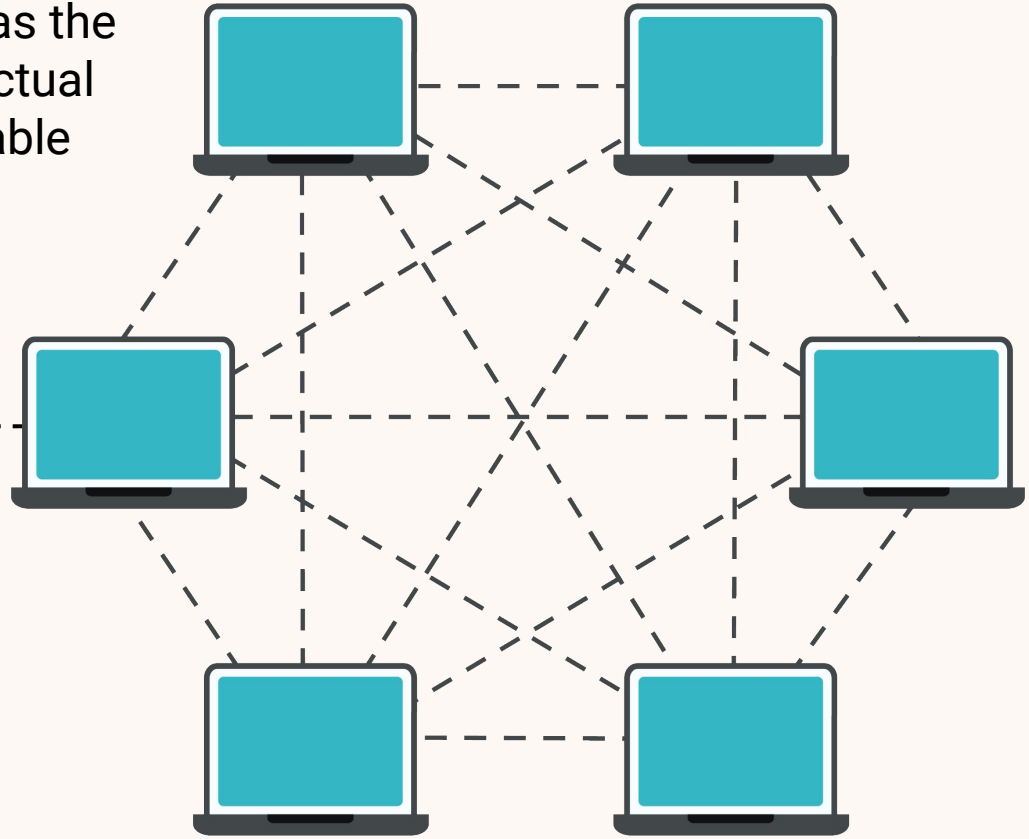
For example…

the internet protocol suite is widely used today, and of the protocols that make up the suite HyperText Transfer Protocol or HTTP is the foundation for communication.

IPFS: The InterPlanetary File System

Another important piece is known as the system's **architecture** or how the actual computers within the network are able to communicate with one another.

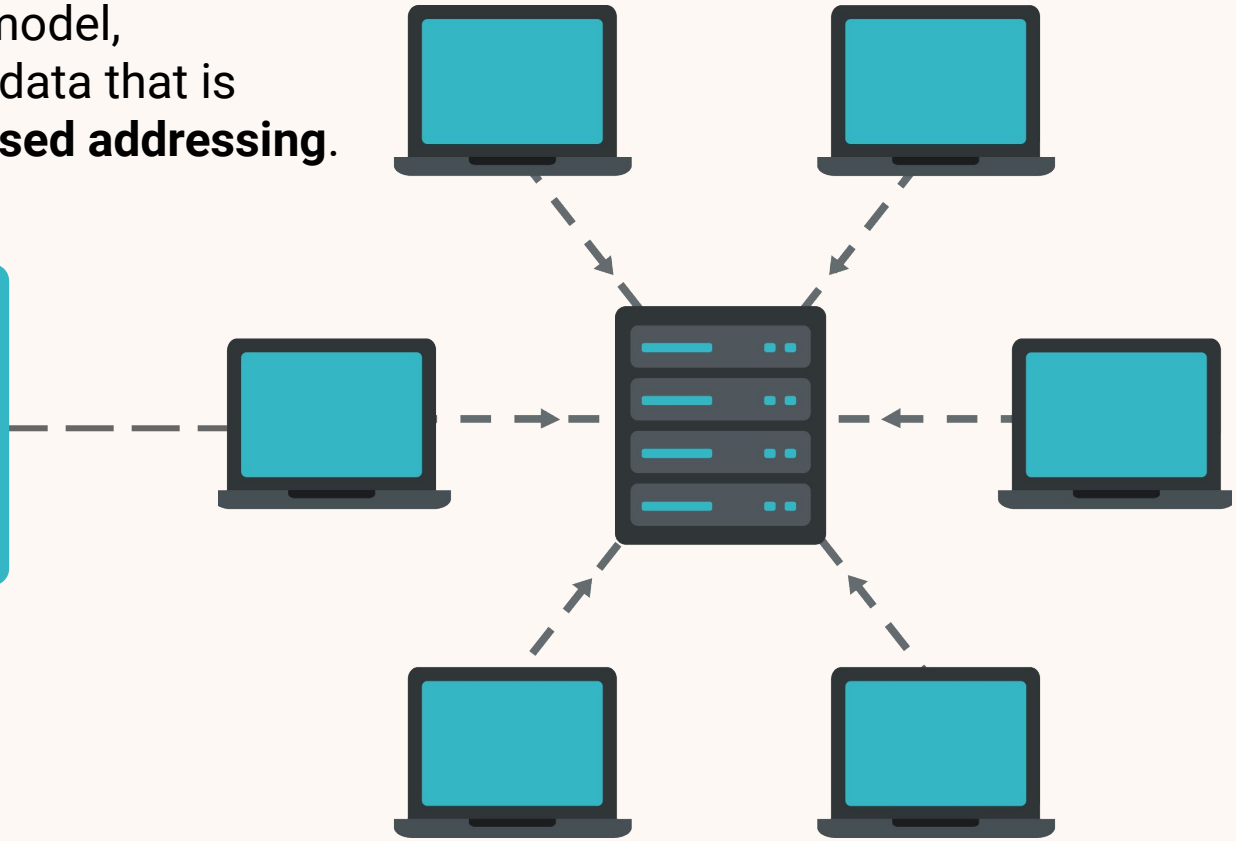
Traditionally this is done in a **client-server model**, however, IPFS leverages a **peer-to-peer network model** of connection.



IPFS: The InterPlanetary File System

In a typical client-server model, centralized servers store data that is accessed via **location-based addressing**.

This provides an easy way to secure and manage data in a scalable manner, though it doesn't come without its drawbacks.



IPFS: The InterPlanetary File System

Because data is stored on centralized servers, anyone with access to those servers; whether an authorized admin or a hacker with malicious intent, can alter and remove data.

This poses problems in both the realms of privacy and security because in this model control of the server is equal to control of the data.

IPFS: The InterPlanetary File System

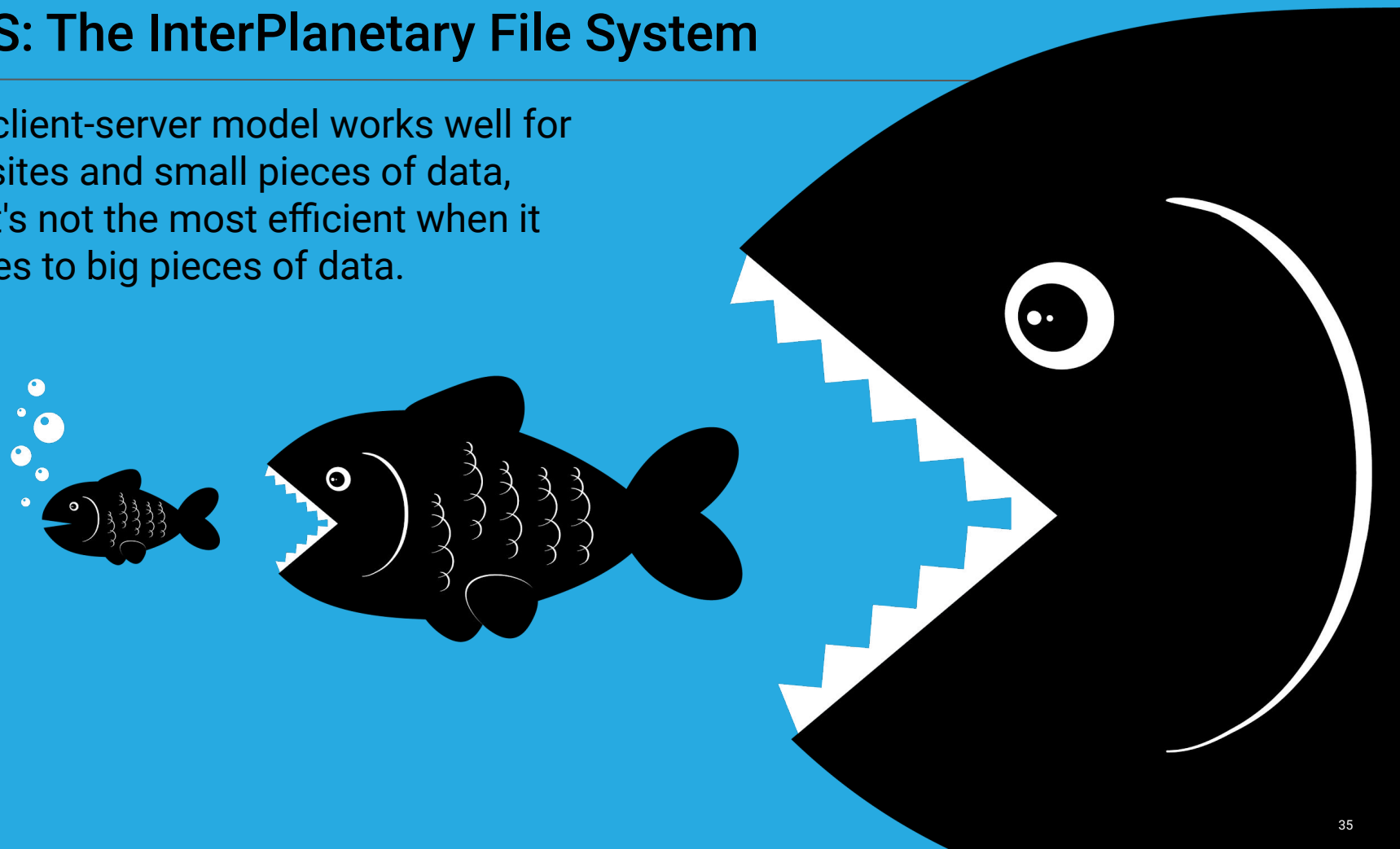
In location-based addressing, a piece of data is recognized by location as opposed to its content. This means that to access a piece of data you must go to its specific location even if the same data is available from a closer source.

IPFS: The InterPlanetary File System

This also means a client has no de facto way to tell if the data it has received has been altered; the client isn't concerned with what the data is but rather where it is.

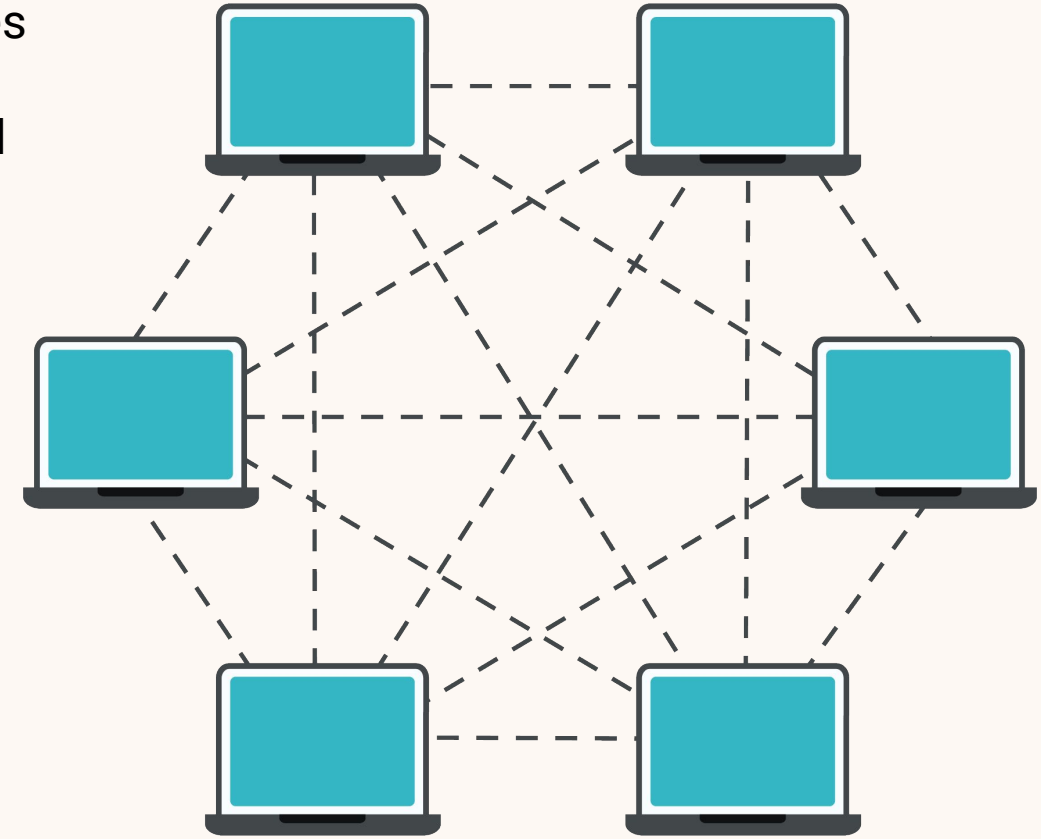
IPFS: The InterPlanetary File System

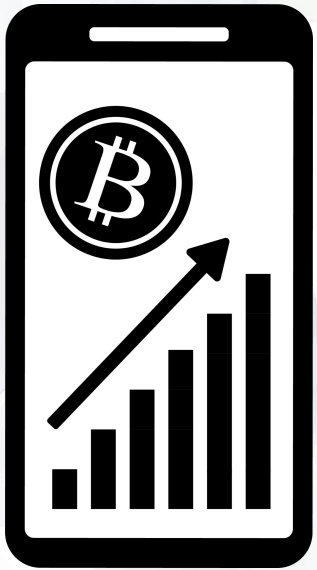
The client-server model works well for websites and small pieces of data, but it's not the most efficient when it comes to big pieces of data.



IPFS: The InterPlanetary File System

IPFS hopes to address these issues with the traditional client-server model with the use of a distributed peer-to-peer file-sharing system.





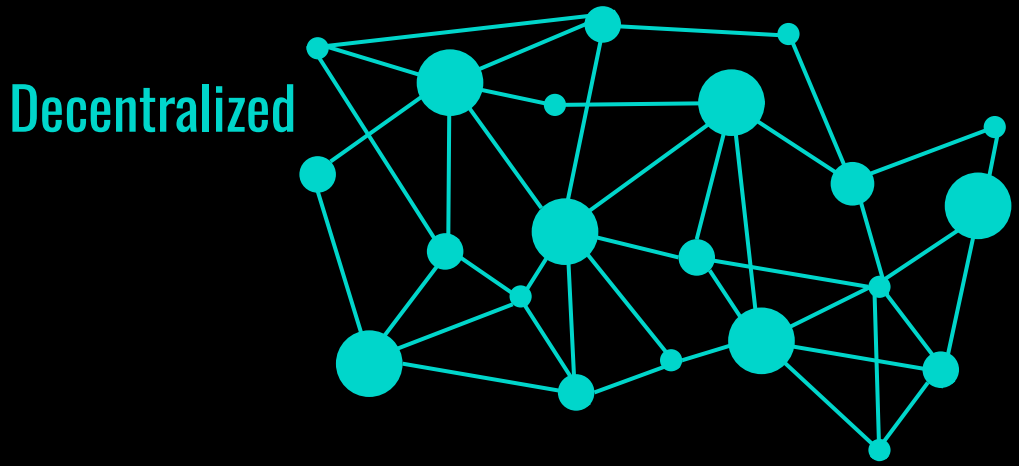
Blockchain offers a powerful way to maintain the integrity of data, but storing data on a chain is expensive.

When combined with blockchain, IPFS can store large datasets with the same degree of integrity as on-chain data.

IPFS: The InterPlanetary File System

IPFS stores and retrieves files from a decentralized system.

- Instead of using centralized storage, like a database, IPFS distributes each file across multiple nodes in its own network.
- It breaks down the file into pieces of data and then distributes the pieces across multiple nodes.
- It does this by using a custom data structure and rules for storing and retrieving the data pieces.



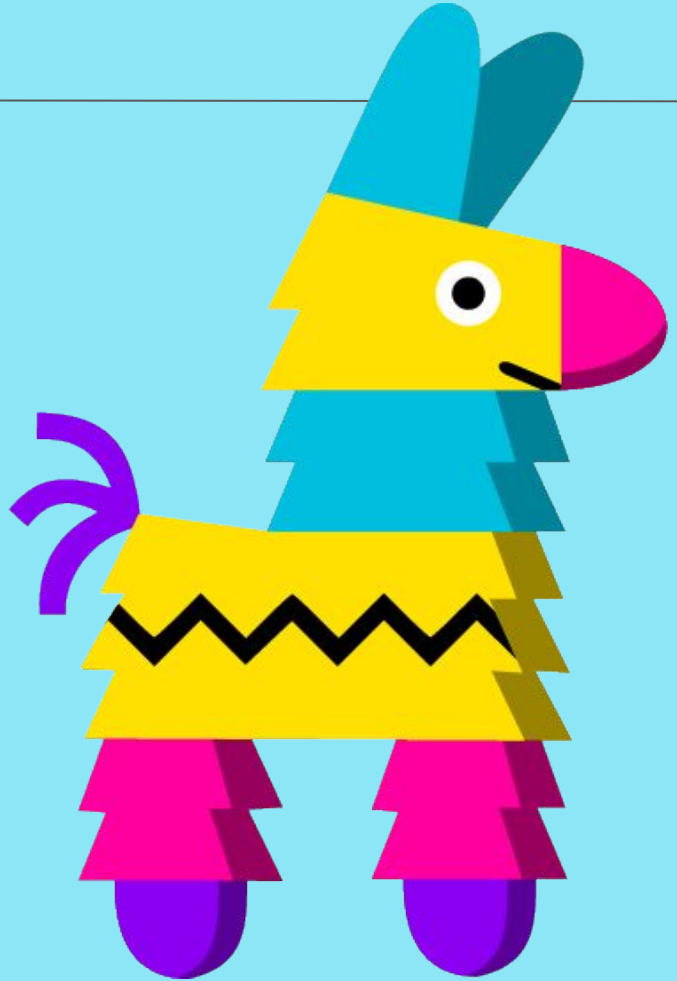
IPFS: The InterPlanetary File System

Pinata IPFS Service

One of the most popular ways for dApps to access IPFS services is through the Pinata IPFS cloud service.

Pinata is a file pinning service for IPFS.

We'll learn how to use Pinata to pin files in the next section.



REMINDER...

You were asked to create an account in Pinata and access the required API keys as part of the setup for this unit.

Questions?


Break

Instructor Demonstration

Pinning Files with IPFS and Pinata

Storing Decentralized Data with IPFS

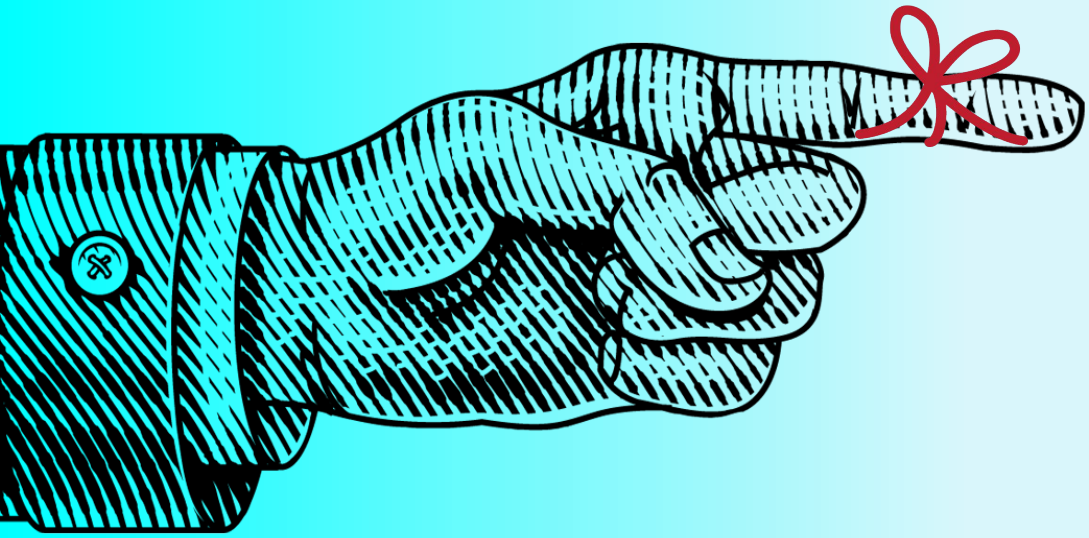
For the integrity of our dApps, we need the data that they reference to always be available. One way to achieve this is to use decentralized storage technologies that are designed to work with dApps.

Centralization issues include:

Having a single point of failure

Being susceptible to attack

Encountering access problems



Note:

- The techniques for using decentralized storage aren't limited to blockchain.
- Decentralized storage is applicable to any financial application, machine learning dataset, or blockchain application that can benefit from the robust storage that IPFS provides.
- You might consider using IPFS for your final project.

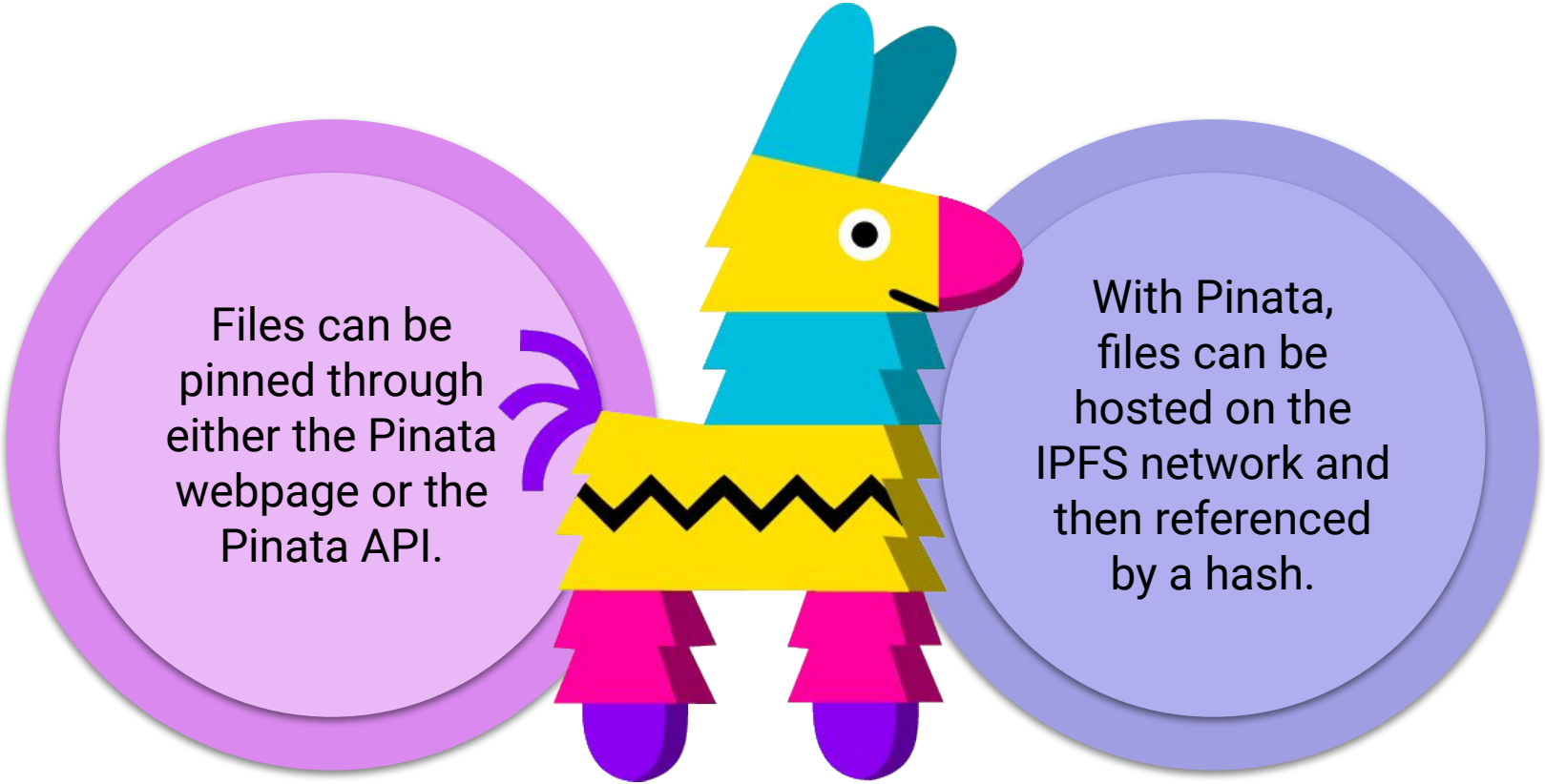
Pinning Files with IPFS and Pinata

Pinata is a file pinning service for IPFS.

- With IPFS, computers can communicate directly with other computers on the network to store and access files.
- This differs from the way that web browsers work.
- Pinata provides a web service, called a gateway, that allows access to files through a web browser without installing IPFS on a computer.
- This **gateway** acts as a bridge between any single computer and the IPFS network.

Pinning is the act of storing a file on the decentralized IPFS network.

Pinning Files with IPFS and Pinata




RECAP

What is IPFS?

IPFS stands for
Inter**P**lanetary **F**ile **S**ystem.

It's a protocol, a network, and a file system that allows users to store almost any type of data on the file system in a decentralized manner.

How does uploading data to a decentralized storage system aid dApps?

Using a decentralized storage system reduces the size of the information that is required to be kept on-chain.

This greatly reduces any gas fees associated with creating and maintaining the token.

How does Pinata work with IPFS?

Pinata is an application that allows a developer to easily upload files to the IPFS.

Users can upload folders or files, or provide a CID.

The result of uploading data to Pinata is an IPFS CID or URL that can be copied and used in an application.

Instructor Demonstration

Using IPFS and Pinata in a dApp

Using IPFS and Pinata in a dApp

Now update our dApp so that it uses the Pinata API to pin artwork files and appraisal reports.

Rather than using the Pinata webpage to upload files, we'll reconfigure our dApp so that it pins the artwork files and appraisal reports through the Pinata API.

Users will then be able to directly upload their files and reports through the dApp. They won't need to visit the Pinata webpage.

The API will also return the hashes for the pinned files. So, the dApp will be able to permanently attach them to the tokens and events in the blockchain.

To ease pinning files and reports from the dApp, we'll build a new Python file that has helper functions for interacting with the Pinata API.

A **helper function** is a regular function that codes some of the complexity that using a service, like the Pinata API, requires

Helper Functions

- A helper function is a regular function that codes some of the complexity that using a service, like the Pinata API, requires.
- All the code for formatting and sending the Pinata API request will be added to helper functions.
- These helper functions, with the required data, can be called anytime that we need to use this code.
- This helps the code to use the Don’t Repeat Yourself (DRY) principle.

In software engineering, **Don’t Repeat Yourself (DRY)** is a principle of software development aimed at reducing repetition of software patterns, replacing it with abstractions or using data normalization to avoid redundancy.

Questions?

Time to Code

Art Registry with IPFS

Suggested Time:

15 minutes

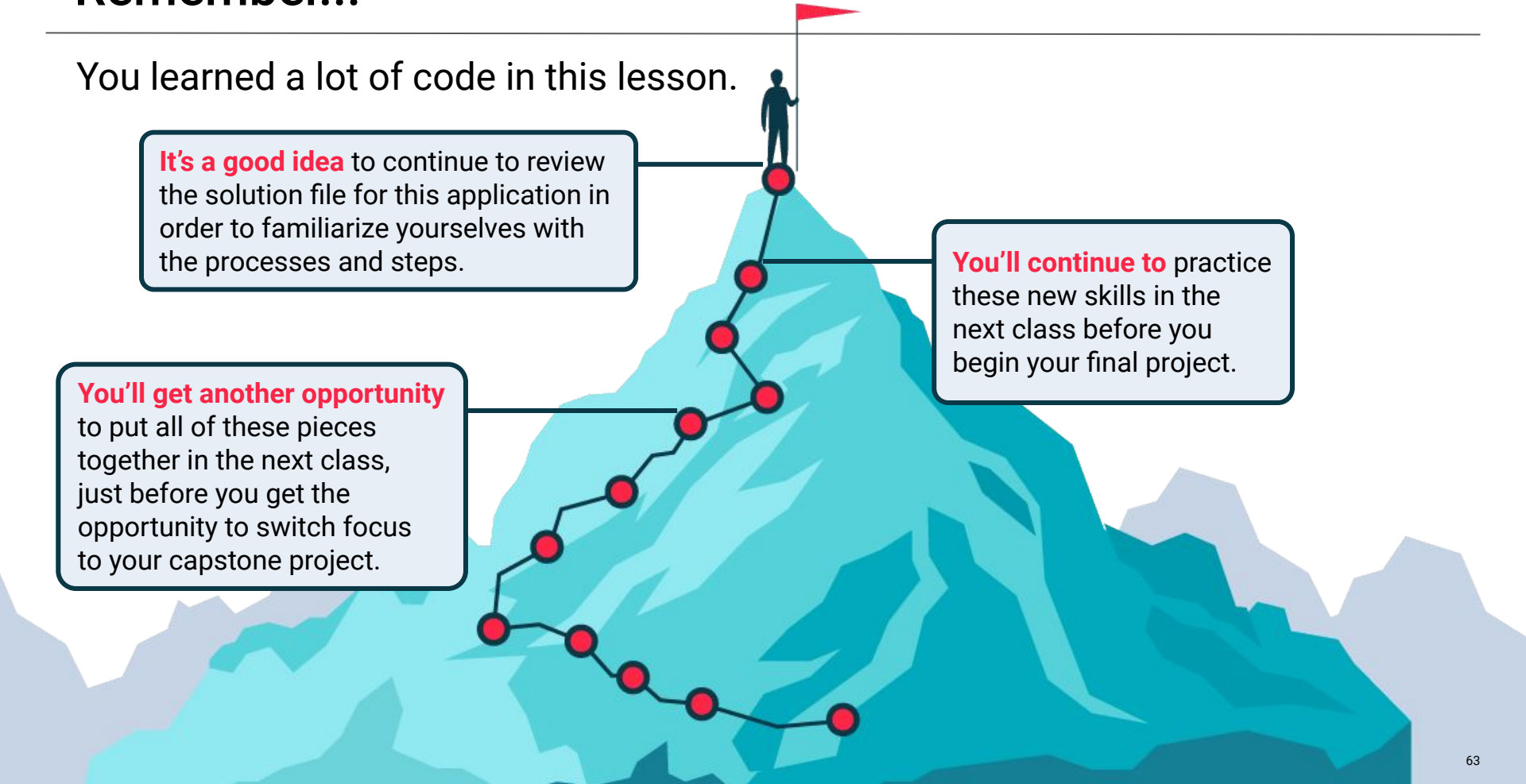
Remember...

You learned a lot of code in this lesson.

It's a good idea to continue to review the solution file for this application in order to familiarize yourselves with the processes and steps.

You'll continue to practice these new skills in the next class before you begin your final project.

You'll get another opportunity to put all of these pieces together in the next class, just before you get the opportunity to switch focus to your capstone project.



Time to Code

Project Time

Suggested Time:

60 minutes

Project Time

Implement best practices that you learned from your previous two projects:

01

Use office hours to ask questions about your project.

02

Communication is key when it comes to successfully completing a group project. Be sure to meet with your group regularly.

03

Plan to work with your group outside of class.


The End