

# Building Certificate Smart Contracts

---

FinTech  
Lesson 22.3



# Class Objectives

---

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



Define a smart contract based on the ERC-721 NFT standard to issue certificates.



Build a dApp using the Streamlit library.



Start working on your Capstone project.





What are some benefits of  
Solidity events?

# Recap

---

Some benefits of Solidity events:



They may have a lower cost for gas than functions do.



They allow you to keep an on-chain log of information.



They're significantly cheaper than contract storage.



They're the built-in way in Solidity to interact with something external, like a UI.



**What are some potential issues  
that IPFS seeks to solve?**



## IPFS seek to solve:

- File integrity issues, like not knowing whether files accessed over the web have changed.
- Security issues, like the fact that centralized servers provide a centralized attack vector.



**Having written many smart contracts and now deployed a decentralized application, or dApp, what are some contracts that you believe a dApp can use?**



A contract that tracks the unchanging  
locations of historical landmarks.



A contract that transfers tokens between two users  
in a decentralized product-swapping website.



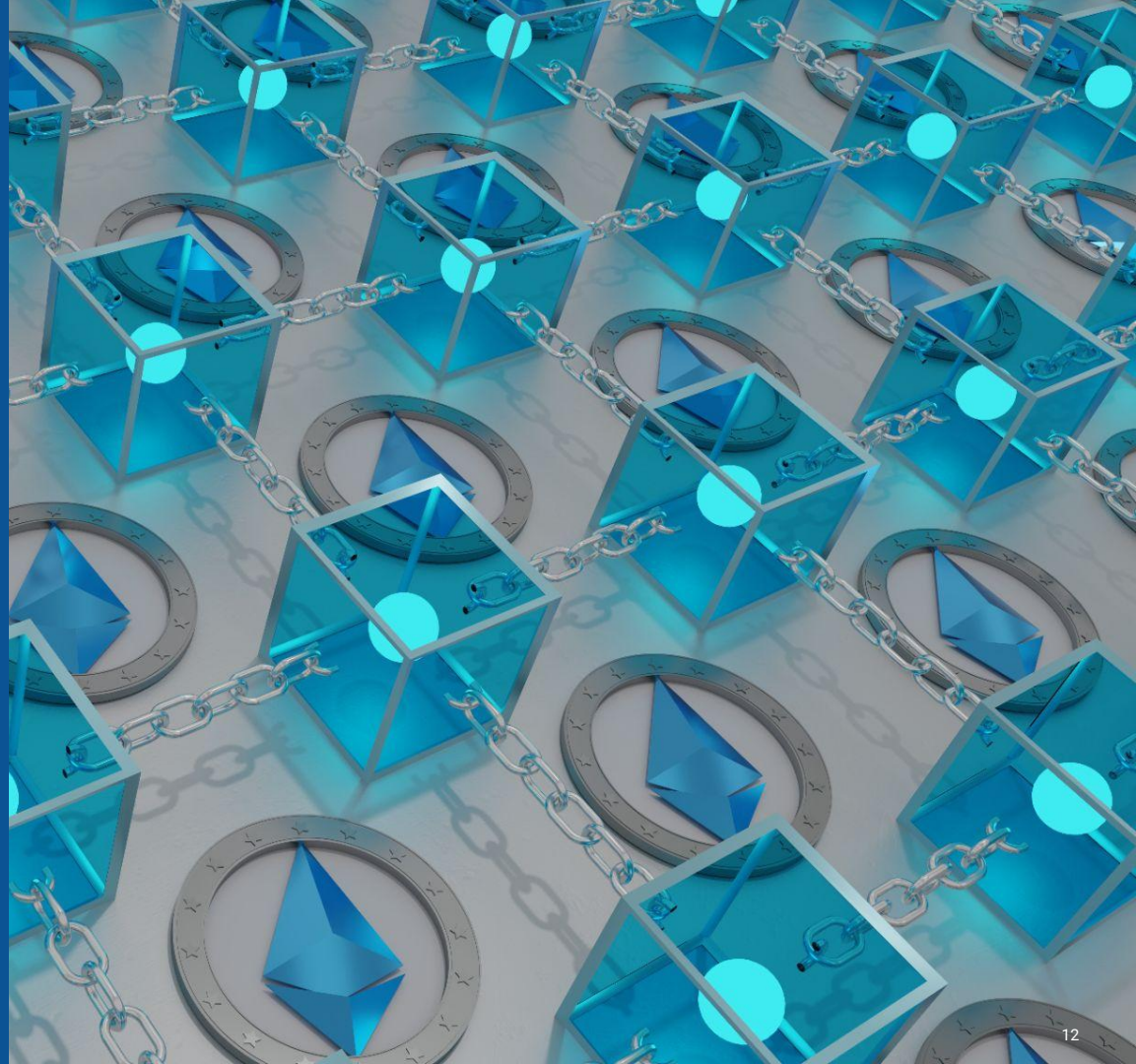
A contract that maintains  
an unchanging list of user  
achievements in an online  
course dApp.





**Any smart contract that  
one can think of.**

The Ethereum blockchain is a  
globally distributed data store  
and supports the development  
of any type of software  
application that uses the  
Solidity programming language.



# Certificate Contract

# Certificate Contract

---

Suppose that the administrators of this course decided to issue your fintech certificate of completion on the Ethereum blockchain.



Would you consider this certificate a fungible or a non-fungible token?



What kinds of information might the certificate include?



Would you include this information inside or outside the token?



# Certificate Contract

---

Because certificates are unique to each individual, they're non-fungible assets.

Each certificate will likely include the program title, the date of completion, and the individual's name.

Although this information can be stored directly in the token, storing on-chain data has an associated cost.

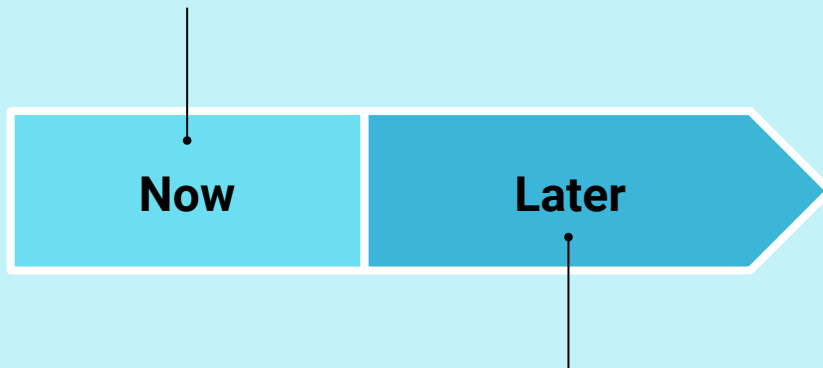
This is because Ethereum [charges fees for smart contracts](#).



# Certificate Contract

---

For the short term, we'll ignore those extra storage costs and focus on building a simple certificate token.



Later, with time permitting, we can expand the contract to use more-efficient methods of data storage.



# Certificate Contract

---

We'll build our certificate token smart contract in sections:

01

Add the Pragma and Import Statements

02

Define the Contract

03

Define the Constructor

04

Define a Function to Award Certificate Tokens

05

Complete the Function



# Time to Code



## Certificate Contract

Suggested Time:

---

30 minutes

# Questions?



# Building Decentralized dApps

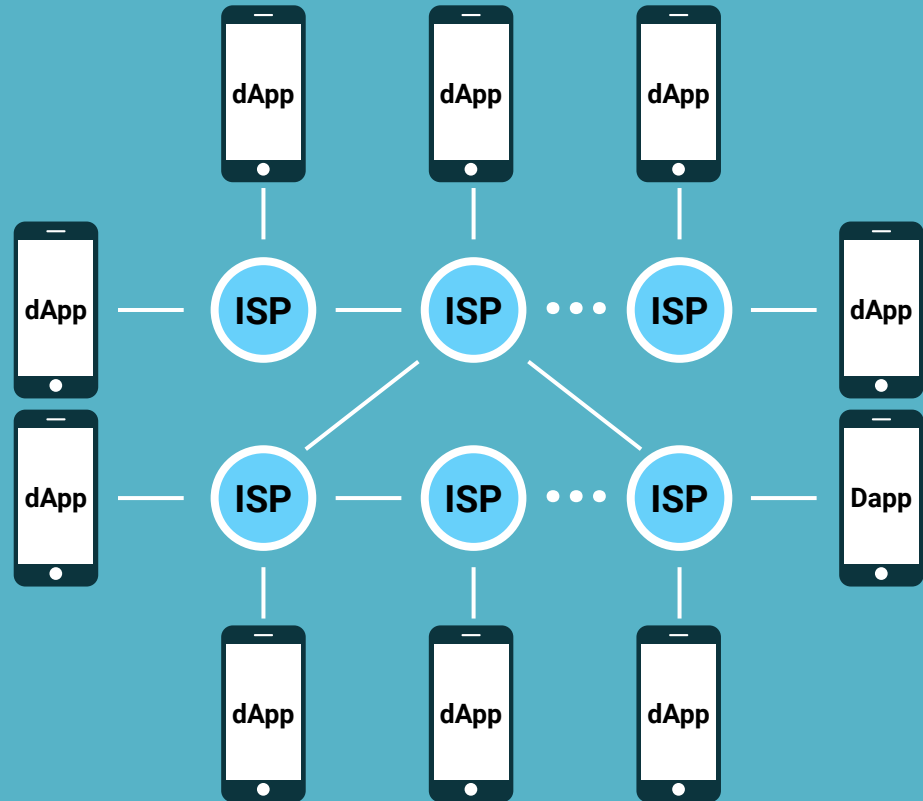


**Can you explain what a dApp is and how it differs from a normal application?**

One of the main features that's related to the success of Ethereum is dApps.

## dApps

Run in a decentralized environment that the blockchain nodes provide.



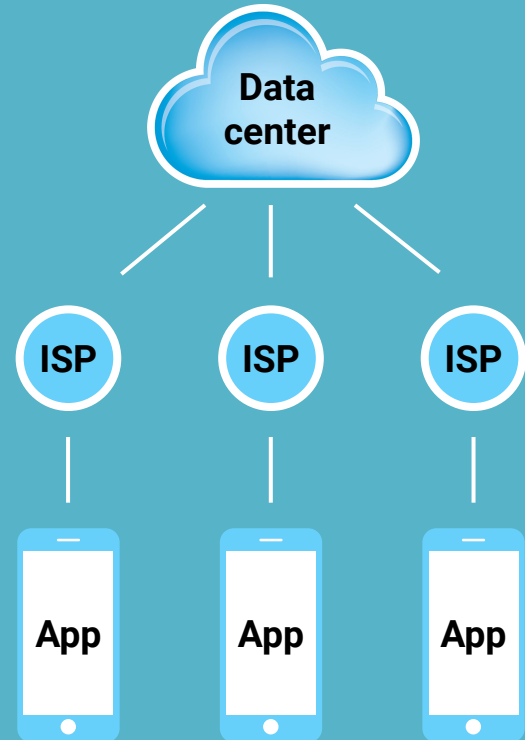
Most applications are centralized, with a central computer that runs the programming logic for the application.

Centralized applications suffer from many of the same limitations and issues as centralized financial systems.

These include having a single point of failure and ownership and governance issues.

## Apps

Run on a centralized server and use centralized storage.

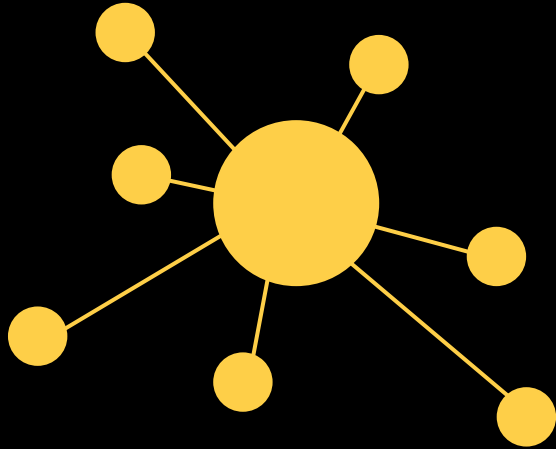


# Building Decentralized dApps

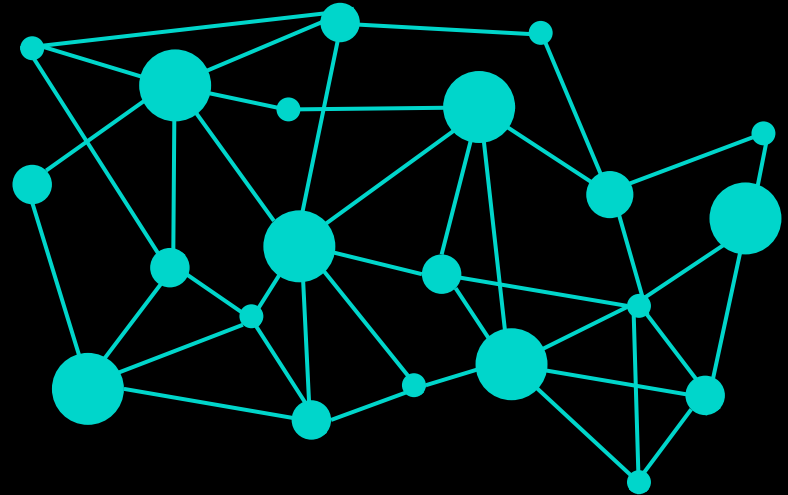
---

Even though dApps often behave like regular applications, they run their core logic on the blockchain in a decentralized manner.

In fact, web applications themselves can become decentralized by using blockchain web hosting services.



**Centralized**



**Decentralized**

---



# Building Decentralized dApps

---

dApps act as the bridge between the users and the contracts.





# Time to Code

## Building Decentralized dApps

Suggested Time:

---

20 minutes



# Time to Code



## Building a Certificate dApp

Suggested Time:

---

25 minutes



Time's Up! Let's Review.





# Time to Code

## Project Work Time

Suggested Time:

---

70 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*