Lecture 10

—

Decentralised Autonomous Organisations

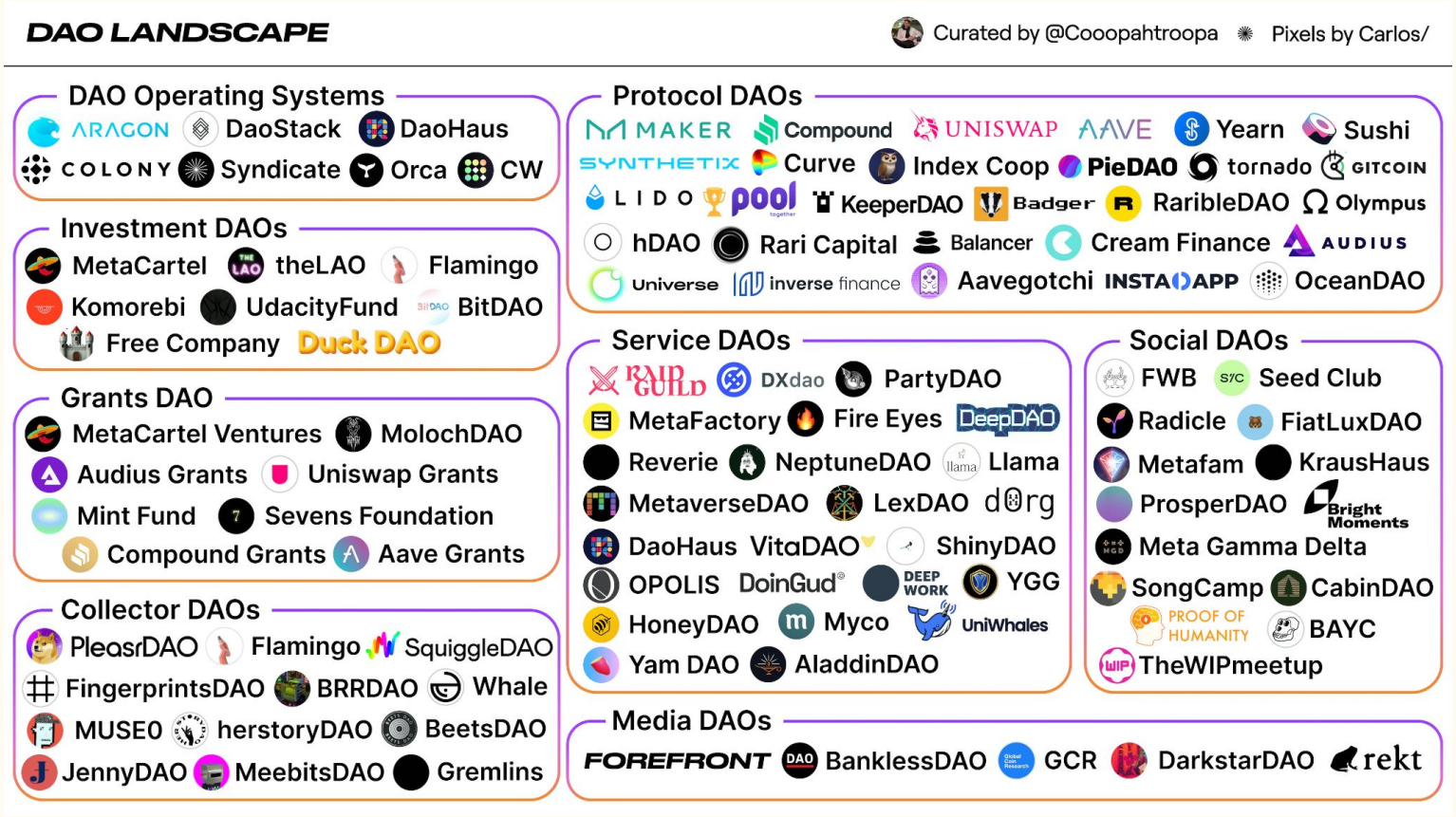
DAO - A definition

Decentralized - Everybody gets to voice their opinion. No hierarchy or authority.

Autonomous - Able to take action and achieve goals without a stamp of approval

Organisation - A community working towards a common goal instead of paid labour

DAO landscape



The two pillars of DAOs

Governance Model

The process of voting and proposal lifecycle management. (Lecture 1 EIP process)

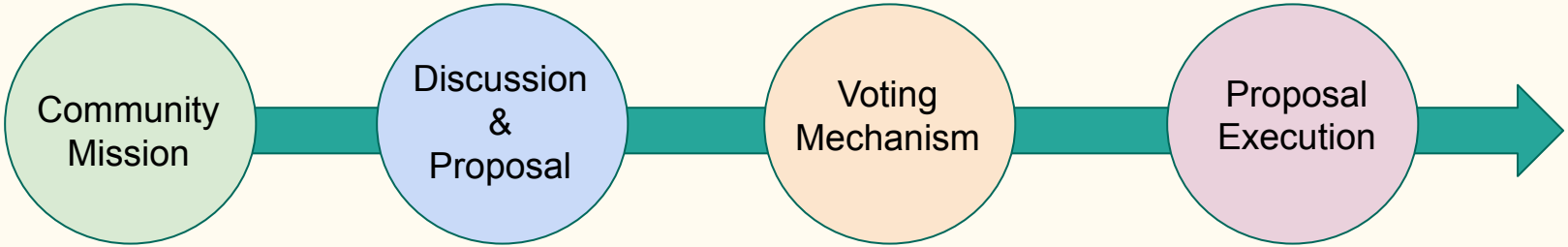
- Tools used to discuss and post proposals.
- Voting mechanisms to ensure fair representation and avoid token manipulation
- Timelocks and execution processes of passed proposals

Treasury Management

Managing funds and tokenomics of the DAO

- Manage asset portfolio and value volatility of the treasury
- Secure smart contract for treasuries
- Native token tokenomics

DAO lifecycle and tooling



Define the purpose of your DAO and the target audience it serves.
.
Define the success metrics and ROI.

Token distribution and delegation

Have a place where people can freely exchange. A sense of community builds loyalty
.
Forum; Orbit; Community as a Service?

Onchain - history, transparency but gas fees
.
Offchain - fast, no gas, needs a separate onchain execution call

Eligibility, Token Power, Duration

Timelocks

Rage Quit (trade with treasury)

Who has power to execute?

Result feedback?

DAO treasuries

Valuation and Asset Portfolio of DAO treasuries

- Holding one token type - ok if multiple treasuries?
- Governance token separate from native/utility token?
- Perform yield earning activities with treasury?

Secure Storage and Execution

- Who owns the purse strings? A single person, an entity or a smart contract?
- Multisig Wallets
  - A wallet which requires a certain number of addresses to sign before execution.
  - Can be a specific number of signatures or a percentage out of the whole

Legal status and Regulation

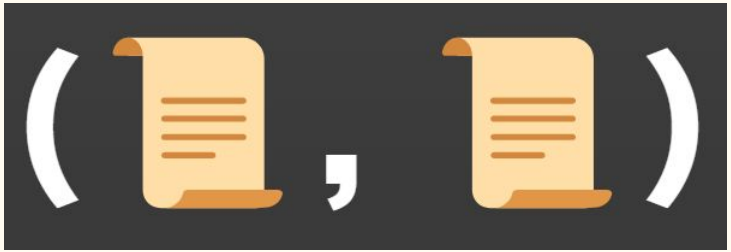
The US has made an attempt to recognise DAOs

- Wyoming became the first state to recognise DAOs on 1 July 2021. Legal definition here.
- Similar in status to LLC but with a certain % of governance done by Smart Contracts, membership based on token ownership and enforced dissolution if DAo does not pass any proposals in 1 year.
- A number of other US states have created similar legislation.

The EU is still undecided

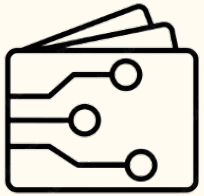
- Non official status in the EU. Most registered as an "association". Provides basic legal protection for private individuals.
- EU MiCA bill - Markets in Crypto Assets
  - Different types of token classification
  - Duties of Token issuers
  - Registration of Crypto Issuers

ConstitutionDAO - An example



A DAO that lived for 7 days, existing fo the sole purpose of purchasing an original copy of the US Constitution

Community was solely managed through Discord. The mission was simply: Putting the Constitution in People's hands



Created a Multisig Wallet. 7/13 Signatures required. Community distrust due to anonymity. Signers volunteered identity.

Created the $PEOPLE token. Exchange ETH for $PEOPLE. Voting rights to what to do after Constitution was purchased.

ConstitutionDAO2

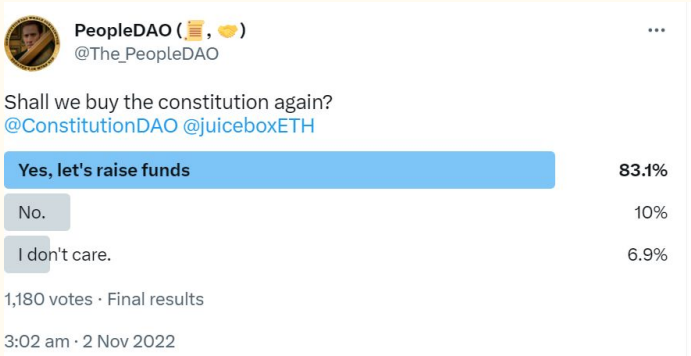
Sotheby's

Raised $47Mil in 7 days. Gained confidence from Sotheby to participate.
Result: Failed to win auction

Full refunds were given on Juicebox at a hard pegged rate of 1ETH : 1Mil PEOPLE.

Varying strategies - get ETH back, hodl PEOPLE tokens for round 2.

In the end, no voting was ever done on the DAO since they failed to purchase the Constitution. However in December 2022… Round 2 starting on Twitter!



Software Development Components

Packages, Testing, Automation

Hardhat Project structure

Project Root

- contracts — Smart Contracts
- test — Unit Tests
- scripts — Automation Scripts
- hardhat.config.js — Project Environment Configuration
- node-modules
- package.js
- package-lock.js

} Package Management

Package Management - Software Lifecycle

Managing software is hard:

Installing - target platform, directories, dependencies, download security

Upgrading - version management, possibly break other software dependent on older version

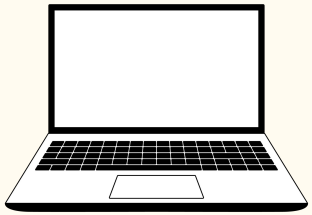
Uninstalling - guarantee a clean uninstall. Not just the core files and dependencies, all symlinks as well. Eg. What happens to your desktop icons when uninstalling?

Package Management - Source and Storage

OS

Framework

Local Package Stores



Remote Central Stores

Every package manager calls to a remote **Package Registry** on private or public cloud**.**

- Stores Packages
- Metadata
- Version tracking
- A nice API to query the packages and run automations

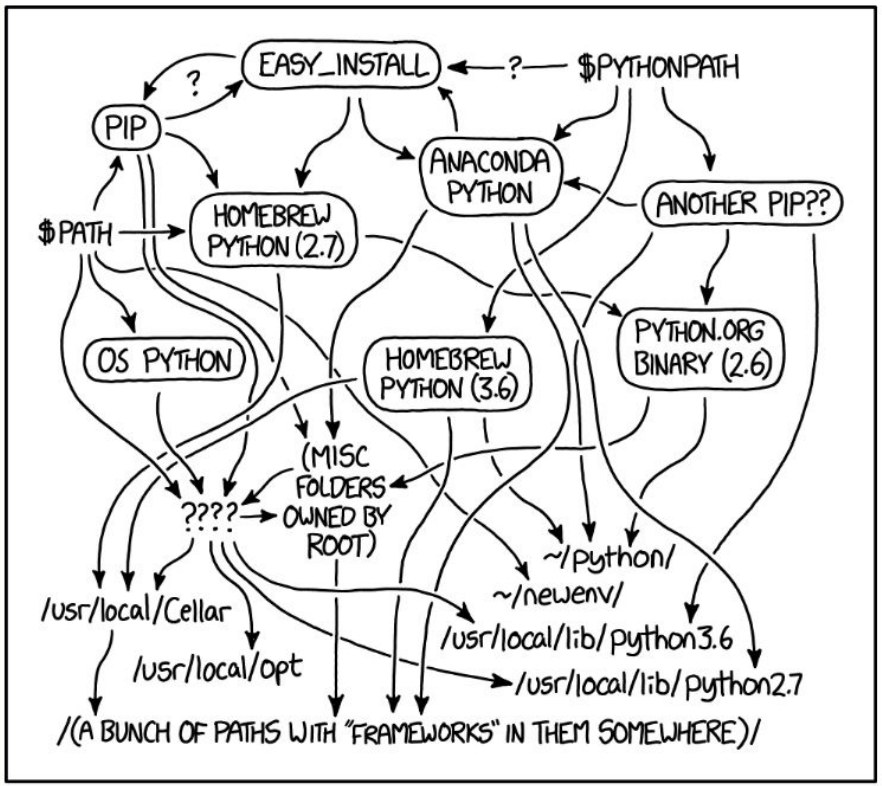
Package Management - Dependency Hell

Many or long chain dependencies

- A project can have many duplicate dependencies
- Project in different locations has a complete duplicate
- Chain - hardhat - npm- node -nvm

Circular or diamond dependencies

- A-B-C-A
- A- B, C - both rely on D.1 D.2
  - Conflicting dependencies



Package Management - .gitignore

.gitignore is a file that exists in the root of your project and prevents the uploading of certain local files onto the public Github platform.

Some file types to include in .gitignore:

- **Package management folders** like node-module: these folders may contain thousands of very large files
- **Logs**: like cache and artifact in Hardhat. They hold execution history which is irrelevant to other people's executions and can expose vulnerabilities
- **Environment Variables**: Never share your private key!! Other sensitive environment variables include API keys, passwords, etc.

Software Testing - Methodologies

Functional	Non - Functional
Unit testing	Performance testing
Integration testing	Security testing
System testing	User testing
Acceptance testing (QA)	Compatibility / Consistency testing

Manual vs Dynamic Tests

Software Testing - Mocha / Chai

Node Test Framework

describe();
it();

- Test Report generation
- Coverage - code and test
- Test execution and performance

Assertion Library

Assert - two things asserted should be equal. Stops execution.
Expect - expect an object with have a number of properties. Continues execution
Should - similar to expect. Obj.should

Automation - CI/CD

Continuous Integration

1. Compile, Lint (grammar check for code)
2. Automatic Testing of new modules (unit tests)
3. Merge into main project code

Continuous Delivery/Deployment

Automatic final testing and release to public (git push or prod deploy)

How does this look like in smart contracts (upgradeable proxy)