Solidity and Smart Contract Development

—

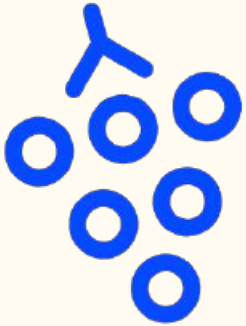
Overview & Syllabus

Syllabus

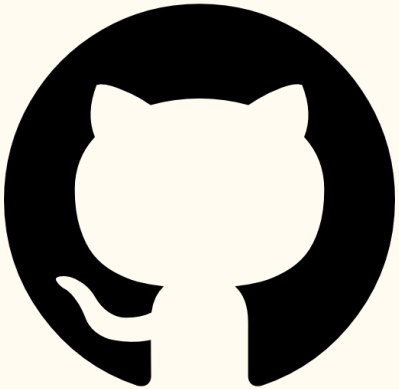
Lecture 1 - Blockchain Basics and Development
Lecture 2 - Web Development
Lecture 3 - Solidity Basics
Lecture 4 - Contracts and Functions
Lecture 5 - ERC20 Tokens
Lecture 6 - Decentralized Exchanges
Lecture 7 - Other DeFi Applications
Lecture 8 - NFTs and Auctions
Lecture 9 - ReFi and Social Good (Guest Lecture)
Lecture 10 - DAOs and Governance
Lecture 11 - Assembly and Gas Optimization
Lecture 12 - ZK and Rollups

Class Tooling

https://discord.gg/yRvr4AvhjJ



https://app.gather.town/app/bOFt4eJwDG85V9qk/Dauphine%20Solidity%20Course



https://github.com/Dauphine-Digital-Economics

Grading

- **Homework - 40%**
  - Weekly homework. Released on Sunday for the week. Submission deadline is next Sunday.
  - Submit on your own Github account.
- **Final project presentation - 30%**
  - Oral presentation (80%)
  - Written submission (20%)
- **In class presentation - 20 %**
  - Weekly on Monday
  - Graded by professor (50%) and a class DAO (50%)
- **Participation - 10%**
  - Github comments
  - Gather Town chats
  - Discord Activity

In Class Presentations

- 15 min presentation + 3 min questions
- Every team member must speak during presentation
- These are not topics covered in class. They are an extension to the previous week's discussion.
- Grading
  - 50% graded by professor
  - 50% voted on by your fellow classmates through a class DAO
    - To avoid problems of collusion and manipulation encountered by small DAOs, professor reserves the right to veto the class vote

Presentation Topics

- 6 February : Evaluate an open source community on Github
  - Intro of community, their Github stats (eg. stars/fork), interesting interactions (issues/comments/PRs). Finally, your assessment of this community.
  - No code. Week 1.
- 13 February: Advanced Solidity data structure: Trees
  - Introduction to trees and subtypes of trees (eg. binary, merkle)
  - Code: Show how to code a Tree
- 20 February: Stability mechanisms of stablecoins
  - Collateral, Seigniorage, etc.,
  - No code. Finance heavy.
- 27 February: decentralized NFT storage: IPFS
  - Introduction to IPFS
  - Demo walkthrough of IPFS
- 13 March: ReFi DAOs
  - DAOs for social good (Carbon, nature conservation, common good, etc)
  - Code optional. Free to decide on presentation style.

Presentation Groups

Group 1	Remy PIGNEL Yangjiawei XUE	Solidity Trees
Group 2	Aizhan ZHAKUPOVA Cedric LION	ReFi DAO
Group 3	Yanming ZHANG Laetitia ASSOR	NFT Storage: IPFS
Group 4	Margot MONGE Lea VIALA	Stability Mechanisms
Group 5	Yichen CHENG Valentin LOIRET	Evaluate a Community

Final Project

- Choose a Topic from next slide and begin working on it as soon as possible.
- Deliverables - due 27 March
  - Oral Presentation - 80%
  - Written Description (approx 500 words blog post) - 20%

Oral Presentation (20 - 25min + 5min questions)

- 5min startup pitch style
  - What is the value / problem addressed by your project?
- 10min project demo
- 10min Solidity code considerations
  - How did you structure your project and why?
  - What considerations did you make while coding (eg. gas optimization)

Written Blog Post

500 words

Description of your project and key features.

Mention interesting technical aspects.

Final Project - Topic List

All projects must be on the **Celo** Blockchain!!

1. Mobile NFT Marketplace
2. Mobile web3 game
3. IPFS storage dApp
4. Voting dApp
5. Celo payments through QR codes
6. Time-Lock Wallet
7. Crowdfunding dApp
8. Propose your own

Hint: check out celo-composer!

Final Project - Groups

Group 1	Group 2	Group 3
Laetitia ASSOR	Yichen CHENG	Valentin LOIRET
Yangjiawei XUE	Margot MONGE	Lea VIALA
Cedric LION	Remy PIGNEL	Yanming ZHANG
	Aizhan ZHAKUPOVA	

Final Project - fast track your web3 career!

All groups will have their written submission edited and published on the Celo Medium Blog to boost their web3 CV.

For the winning team - marketing support and exposure to the Celo Ecosystem.

Have a great semester and good luck!

Lecture 1

Blockchain Basics and Development

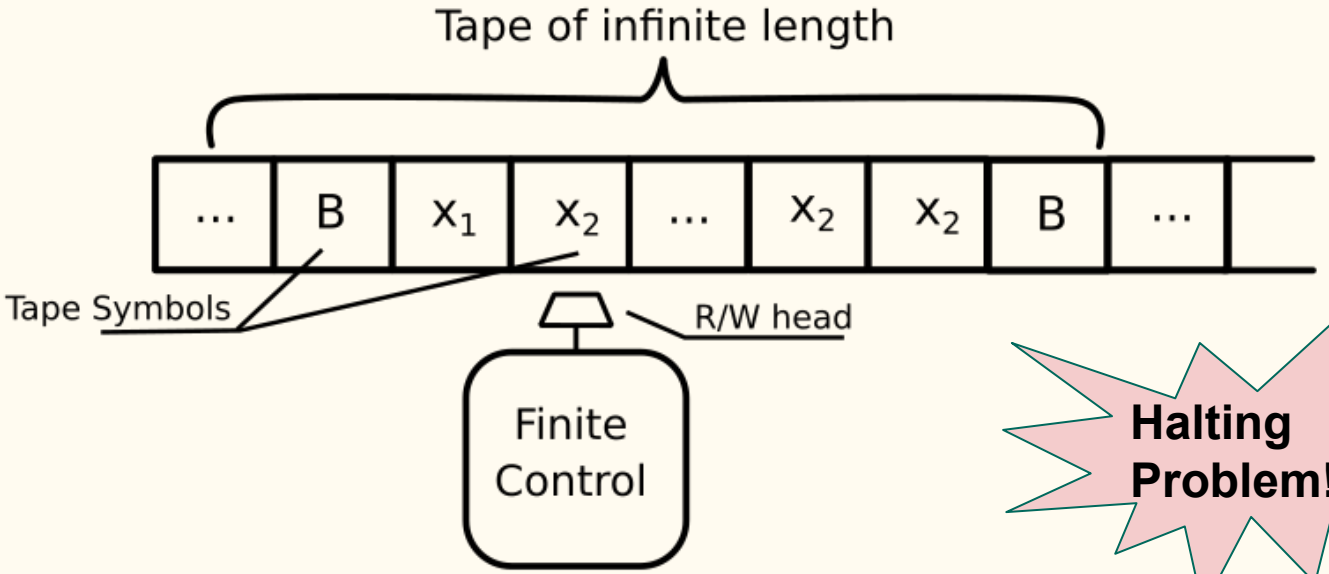
Birth of Crypto
Rise of Ethereum

Cryptocurrencies - Secure, Anonymous, Independent

- Many attempts to create a digital currency and break free from traditional banking
  - eCash by David Chaum in 1983
  - HashCash, eGold, BitCash

- Lacked awareness, suffered from attacks, used for dark web activities

Turing machine / Turing Completeness



Source: https://iq.opengenus.org/general-introduction-to-turing-machine/

Ethereum - A Turing Complete State Machine

Ethereum World State
Merkle Patricia Trie

EOA
Smart contracts
Address

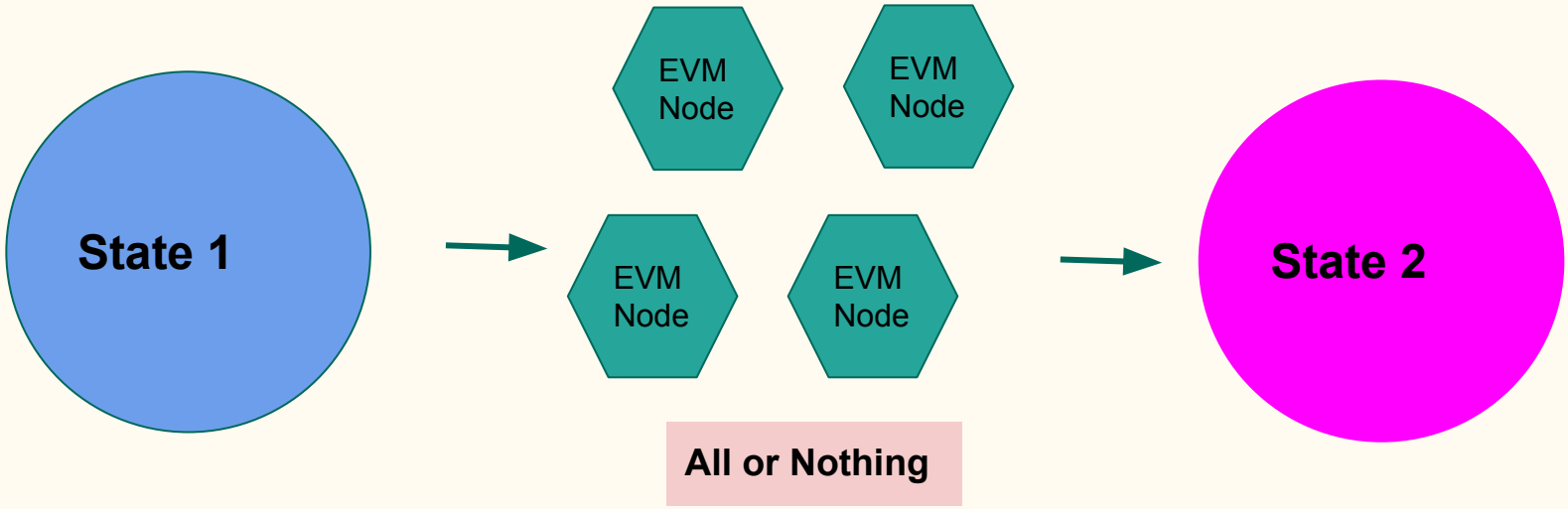
To

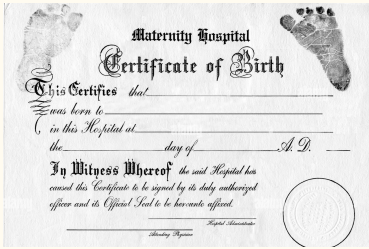
Balances, Nonce, Variables, address pointers, gas limits
……

Externally Owned Accounts
- No Code, no data
- Public/private keys
- Hardware, software

Smart Contracts
- Hash of code and data storage
- Needs a sender address

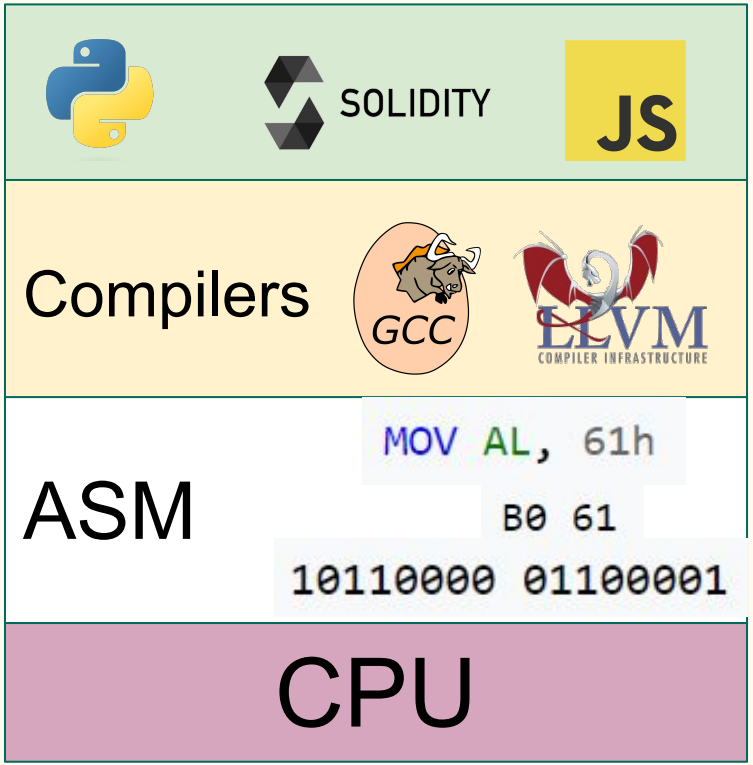
Ethereum - A Turing Complete State Machine





Turing Machines
A history and A comparison

Web 2 vs Web 3 Architecture



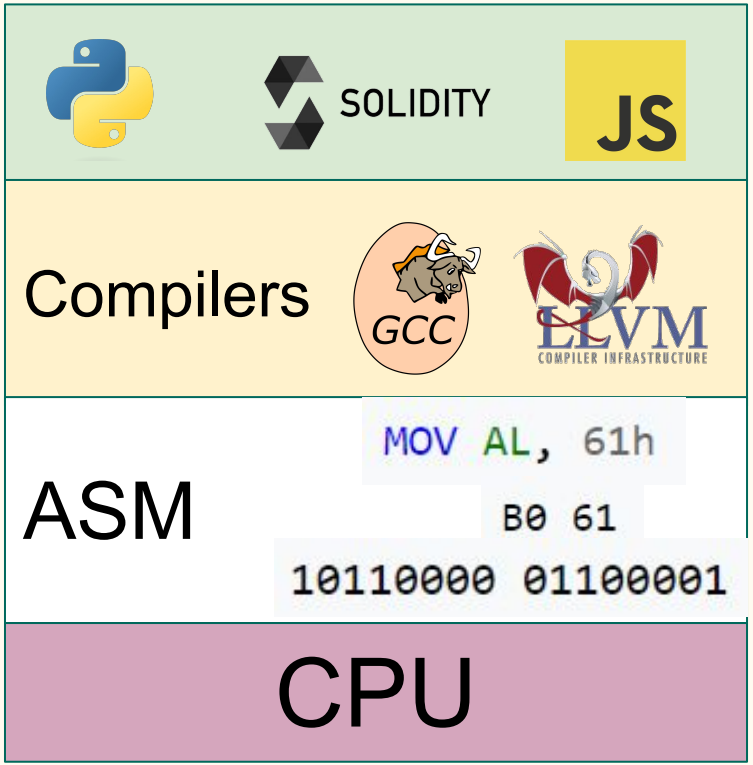
Human readable -
"High Level"

Translation program:
Bytecode -VM
Machine Code - Binary
ASM - Instructions

Machine Language -
"Low Level"

A Turing Complete,
Finite State Machine

Web 2 vs Web 3 Architecture

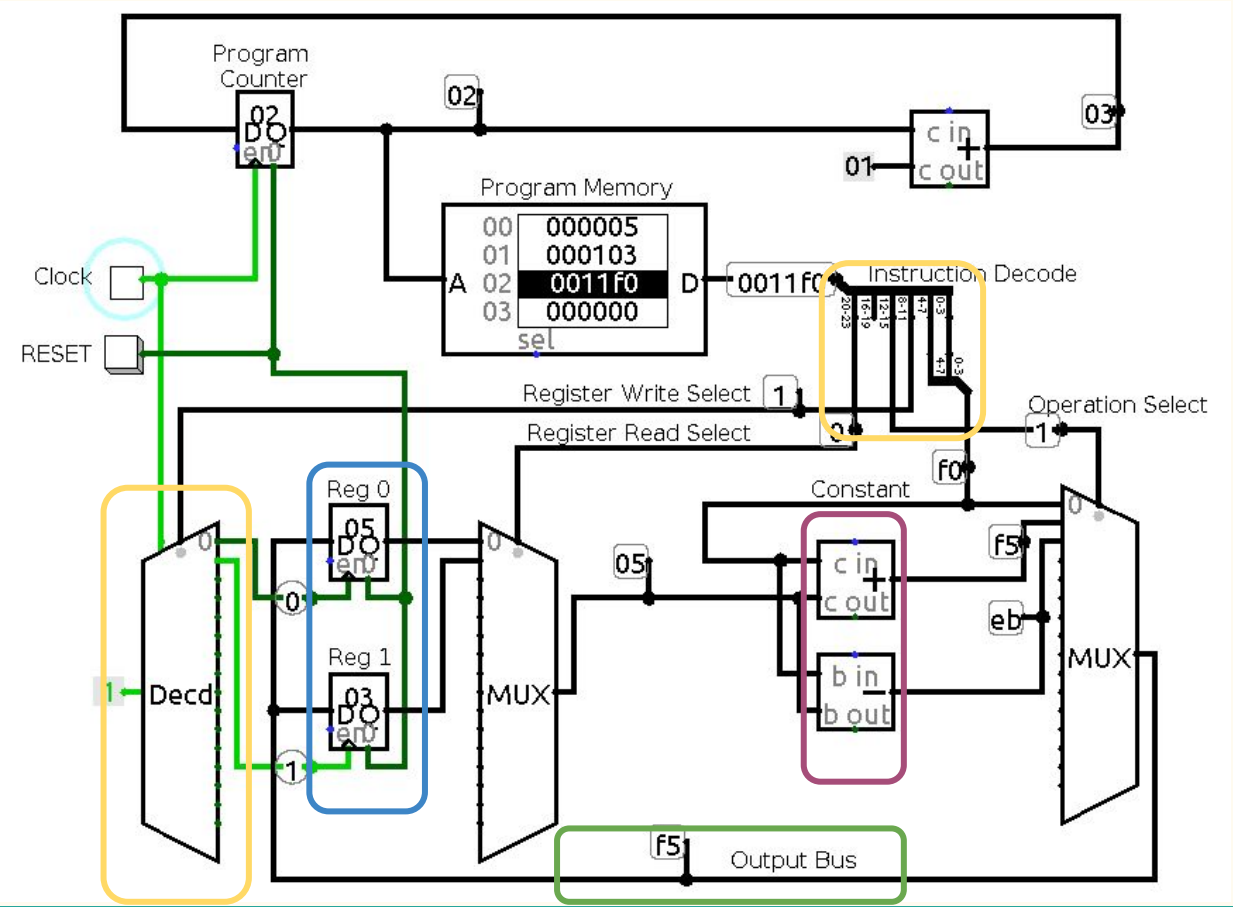


var a;
a = 1+1

malloc 256;
add 1 1
write a 2

01101 256
00100 0001 0001
10001 0x456 0010

Web 2 vs Web 3 Architecture



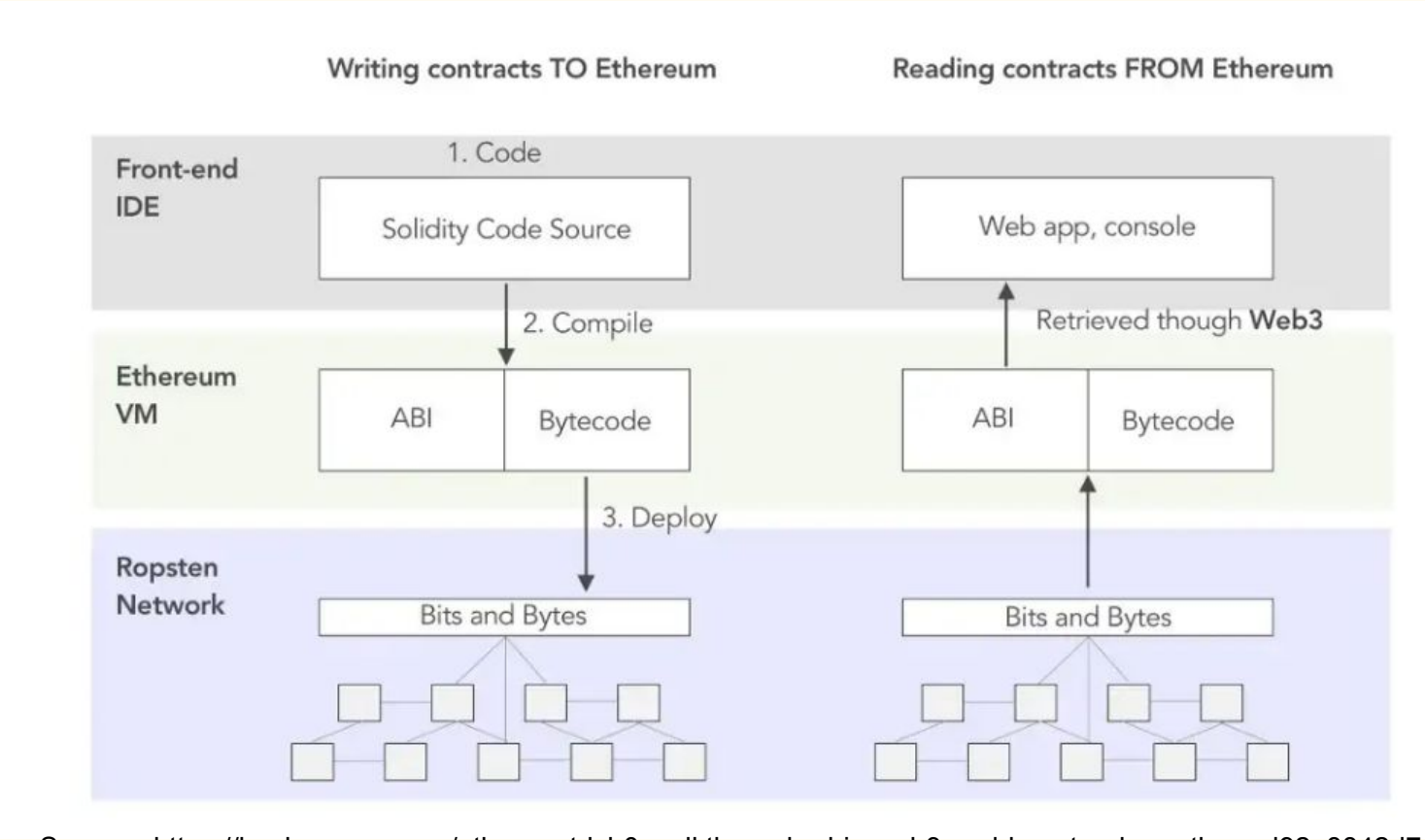
Decode Instructions into opcode and data

Registers to hold **program** essentials: Data, loops state, pointers

Algorithmic Logic Unit

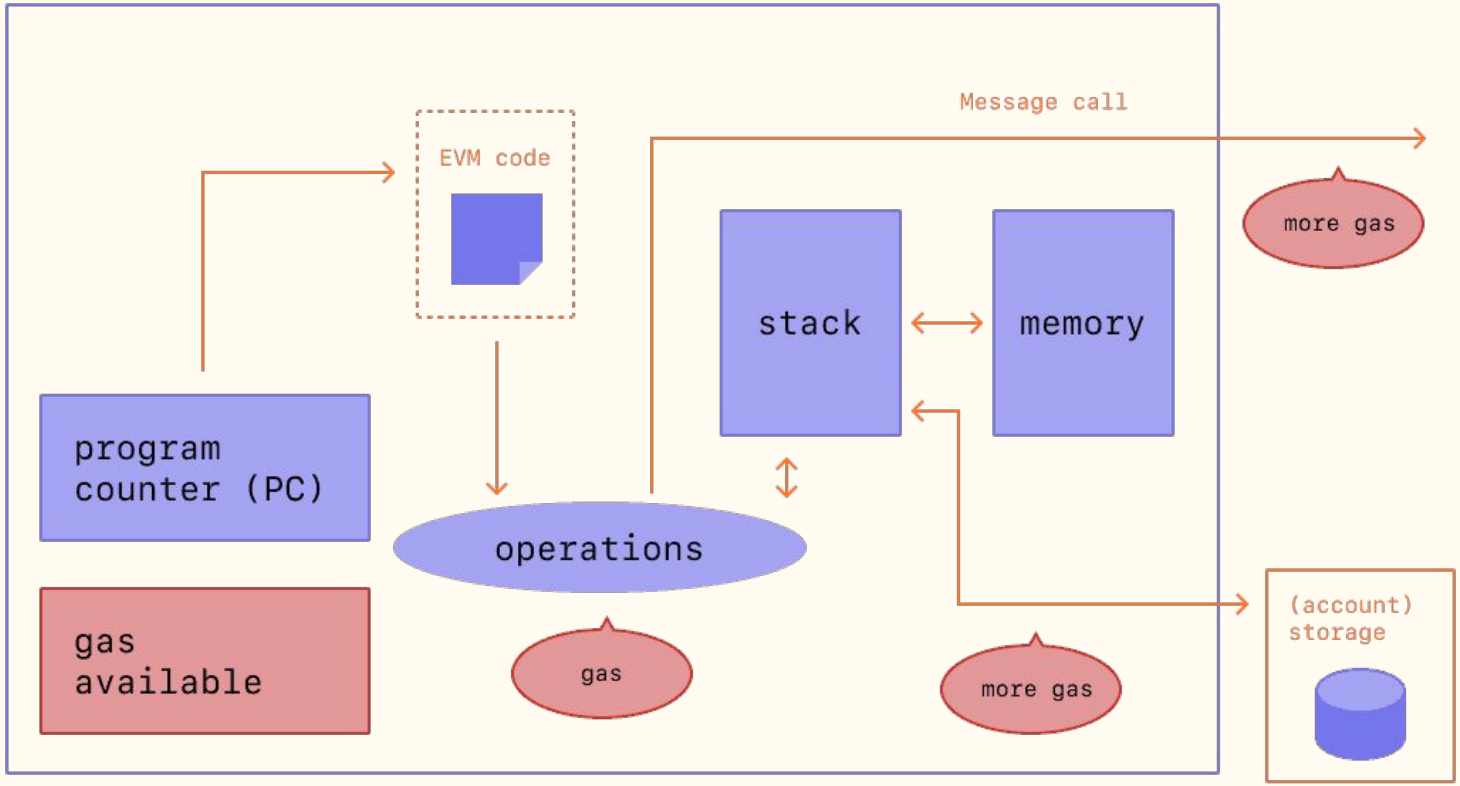
Ram read write, towards more permanent storage. Indexed by **Addresses**

Web 2 vs Web 3 Architecture



Source: https://hackernoon.com/ethernaut-lvl-0-walkthrough-abis-web3-and-how-to-abuse-them-d92a8842d71b

Web 2 vs Web 3 Architecture



Open Source

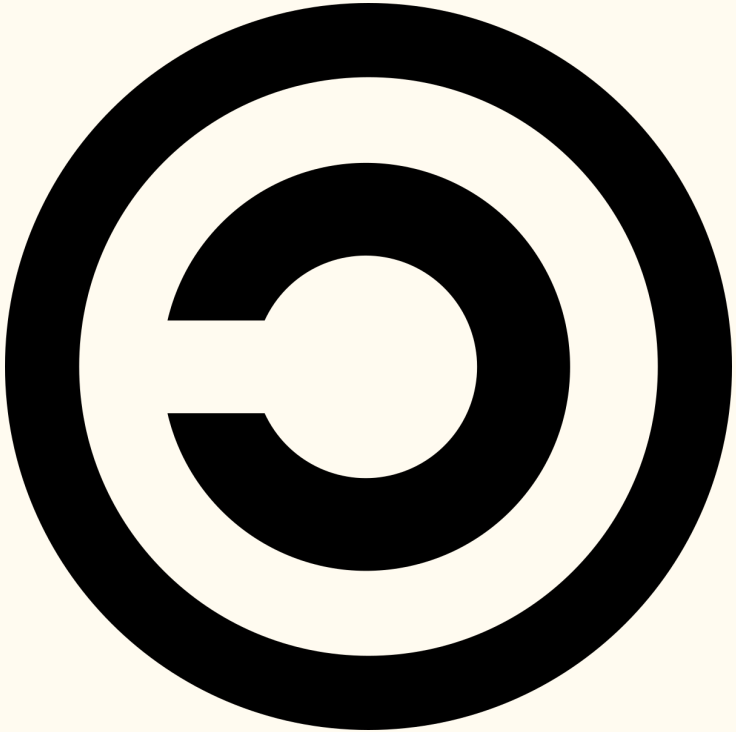
The principle that inspired Decentralization

Free Software Movement

4 Fundamental Freedoms

Purpose Freedom to run the program as you wish, for any purpose.	**Knowledge** Freedom to study and modify the program.
Solidarity Freedom to distribute your creations to help others	**Community** Freedom to redistribute changes and improvements for the benefit of the community

Battle of the Copies - Left vs Right



Open Source Licensing - The corporate strikes back

Permissive

- Do what you want
- You can copyright your version
- Don't sue me
- No marketing? (BSD3)
- Can withdraw (Apache2)

Copy Left

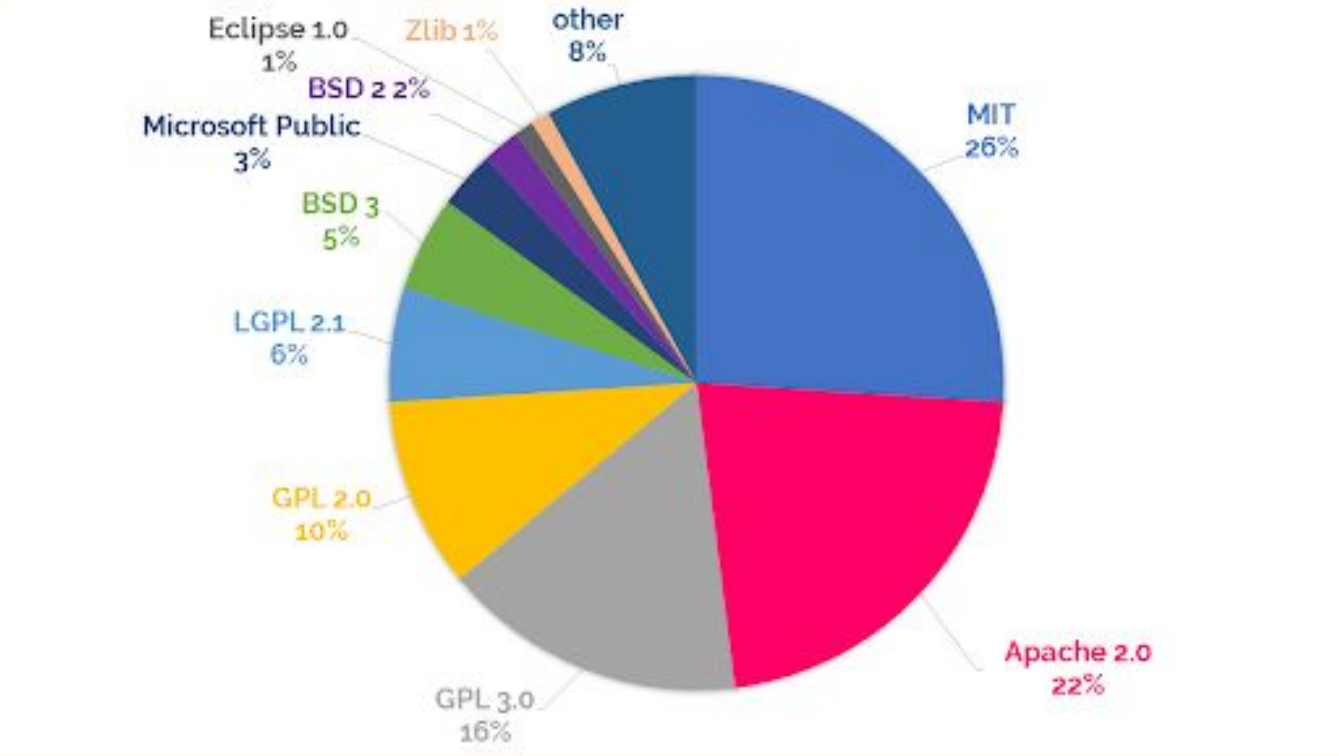
Weak

- Must show source code of the original or direct modification
- your own code can be proprietary (use original as library)

Strong

- Must show source code!
- If you use this code, your code must also be show
- Known as 'viral'

Open Source Licensing - current distribution

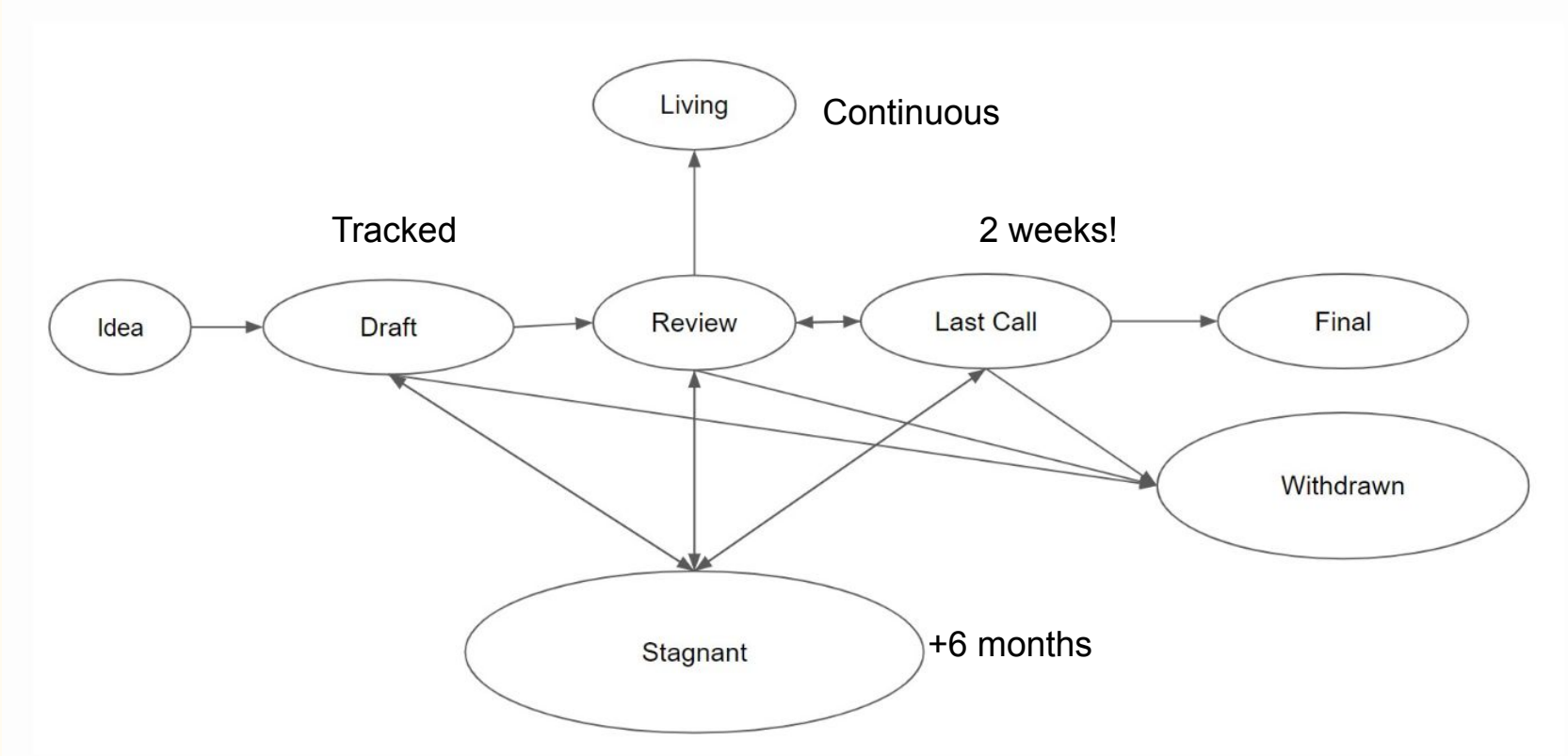


Source:
https://www.activestate.com/blog/the-developers-guide-open-source-software-license-comparison/

Check out our class Github!

https://github.com/Dauphine-Digital-Economics

How to be an open source Contributor - Ethereum Improvement Proposals



How to be an open source Contributor - Ethereum Improvement Proposals

01 **Standards Track**

- Core: Improvements on the protocol
- Network: How Ethereum talks to low level structures like databases
- Interface: How ethereum talks to clients(dApps)
- ERC: Application layer

02 **Meta**

- Discussions on the EIP process itself

03 **Informational**

- Gives Information or recommendation or opinion about certain aspects of the Ethereum Blockchain. Can ignore.