# Balancing On- & Off-Chain Architectures

Anna Carroll & Mitchell DeMarco


# Anna Carroll

@annascarroll

- B.S. Computer Science, Stanford ‘18
- Formerly led Stanford Blockchain Collective
- Engineer at Dharma
- Kleiner Perkins Product Fellow
- Generally trustworthy

# Mitchell DeMarco

@Neablist

- Engineer at Dharma
- Been at, like, a lot of startups
- Built products with 10m+ MAU
- Prefers cats over dogs *(don't tell my dog)*

# Dharma

@Dharma_HQ

- Technically invented the term #DeFi

  *(sorry, Maker)*

- Building the first Defi app that goes

  Mainstream

- Pretty colors

# We've learned *a lot* about building usable crypto apps at Dharma

# We want to help you build crypto products that work better for users

# To do that, you need an architecture that enables better UX

# Let’s dive into how you can do that with your next DeFi app

# Traditional Architectures

# Traditional Off-Chain Architectures

# Web2 Developers are very familiar with fully off-chain architecture

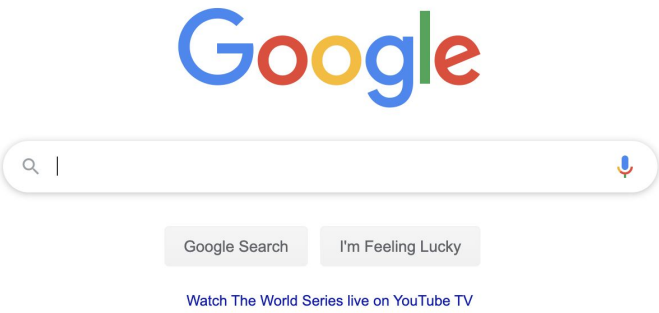
# Developing and Scaling has known solution

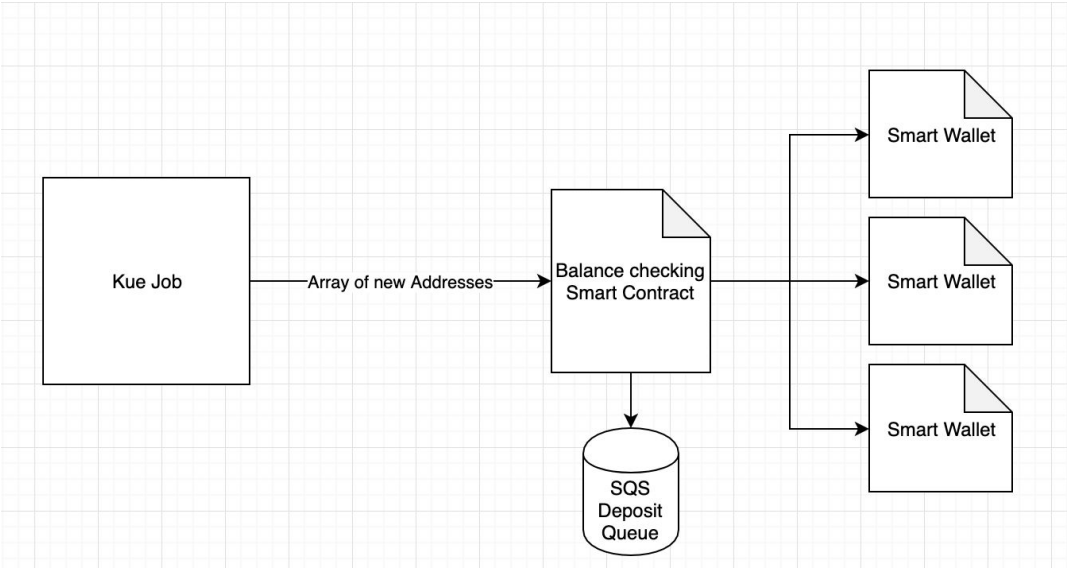


# The tools in the ecosystem are well developed


# Usually containing three major parts

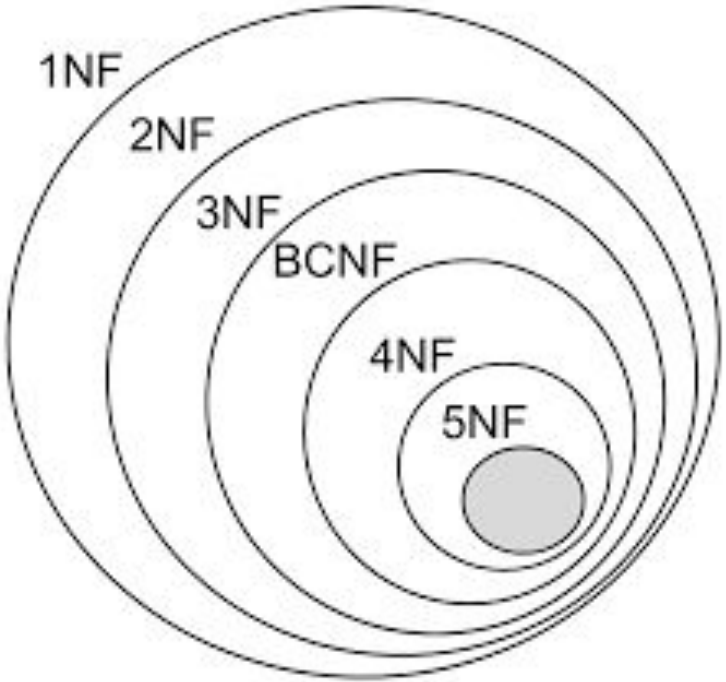
# Frontend


Google
Google Search
I'm Feeling Lucky
Watch The World Series live on YouTube TV




# Frontend

# Backend

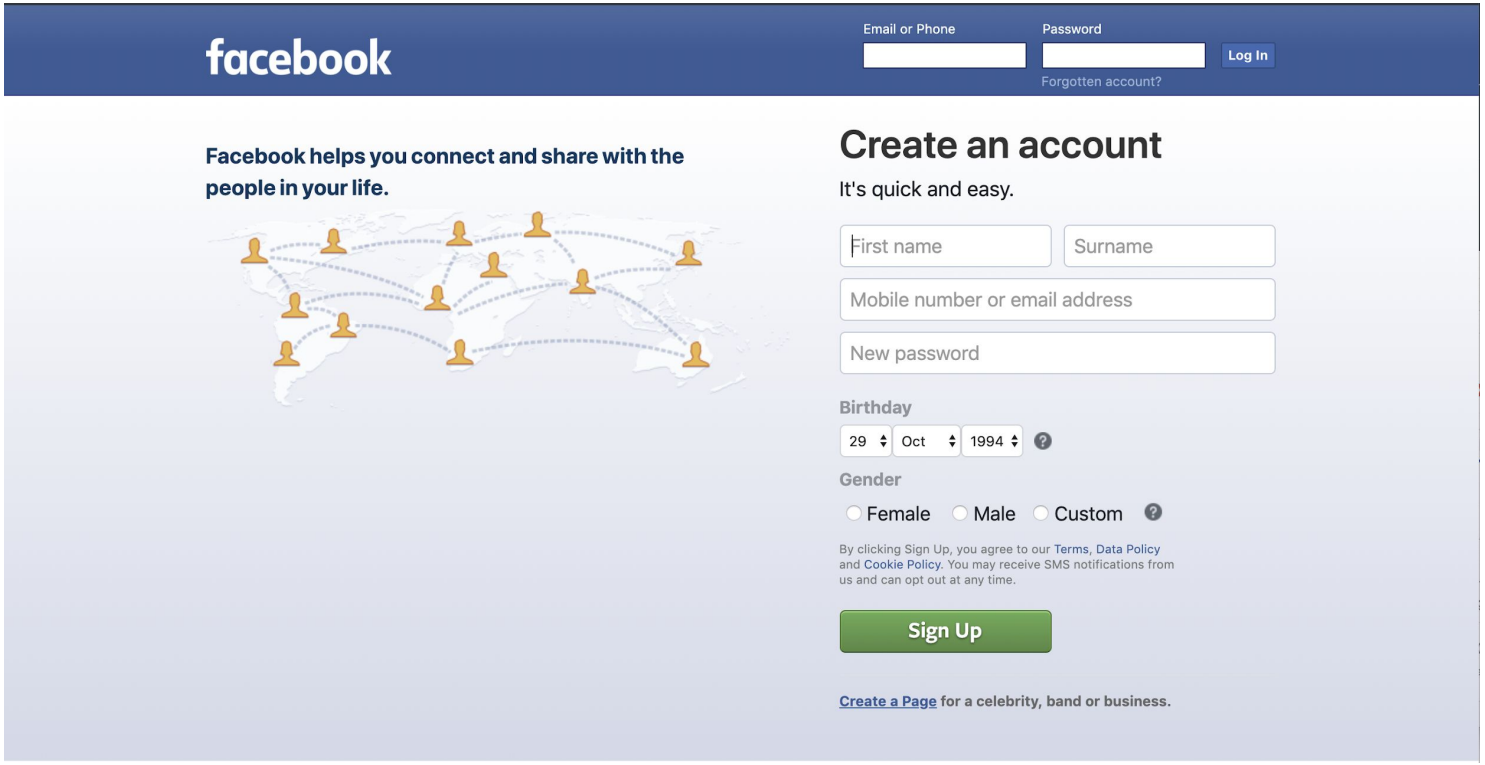



**Frontend**

**Backend**

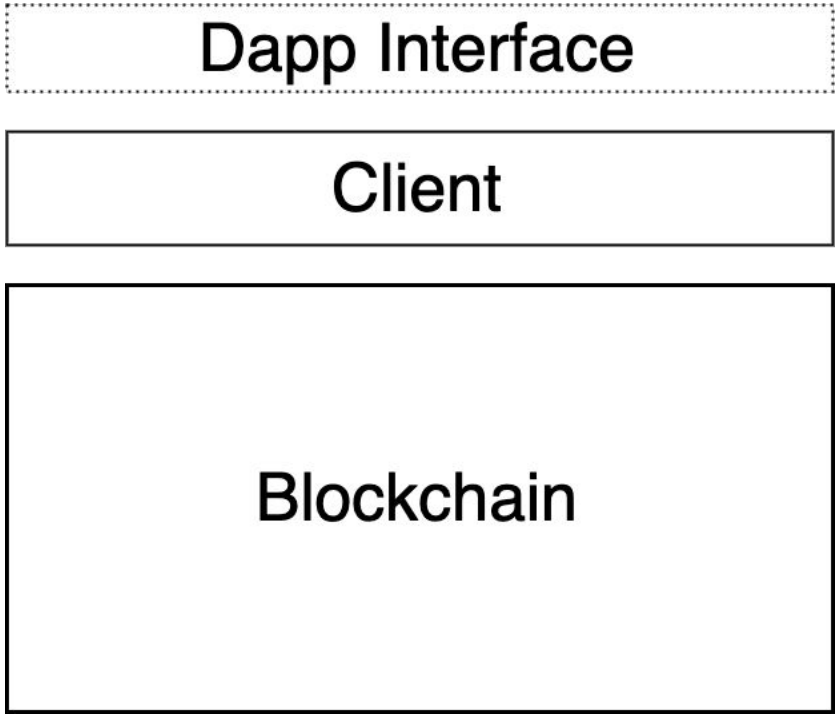
**Database**

# So Why Does Full Off-Chain Suck?


facebook
Email or Phone
Password
Log In
Forgotten account?
Facebook helps you connect and share with the people in your life.
Create an account
It's quick and easy.
First name
Surname
Mobile number or email address
New password
Birthday
29
Oct
1994
Gender
Female
Male
Custom
By clicking Sign Up, you agree to our Terms, Data Policy and Cookie Policy. You may receive SMS notifications from us and can opt out at any time.
Sign Up
Create a Page for a celebrity, band or business.

# Traditional On-Chain Architectures

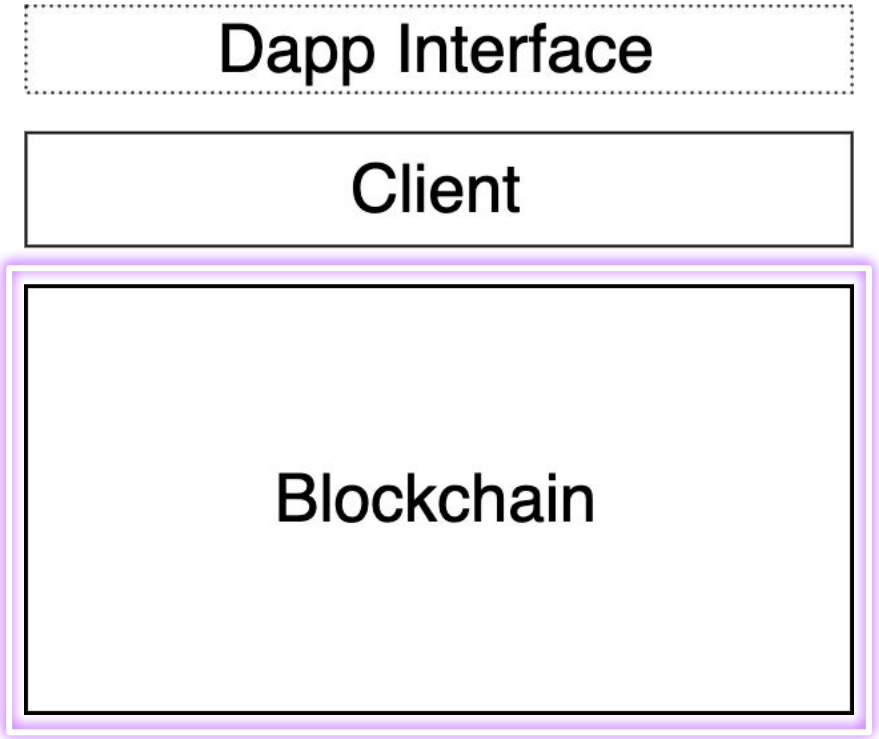
# Web3 Developers are very familiar with fully on-chain architecture



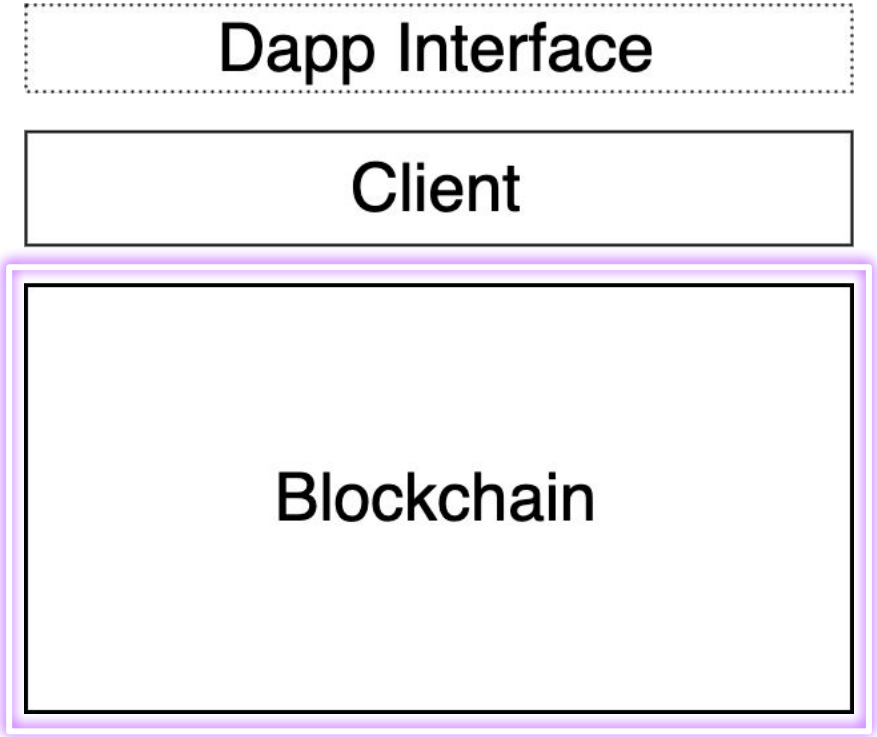

**dApp Interface**

**Client**

**Blockchain**

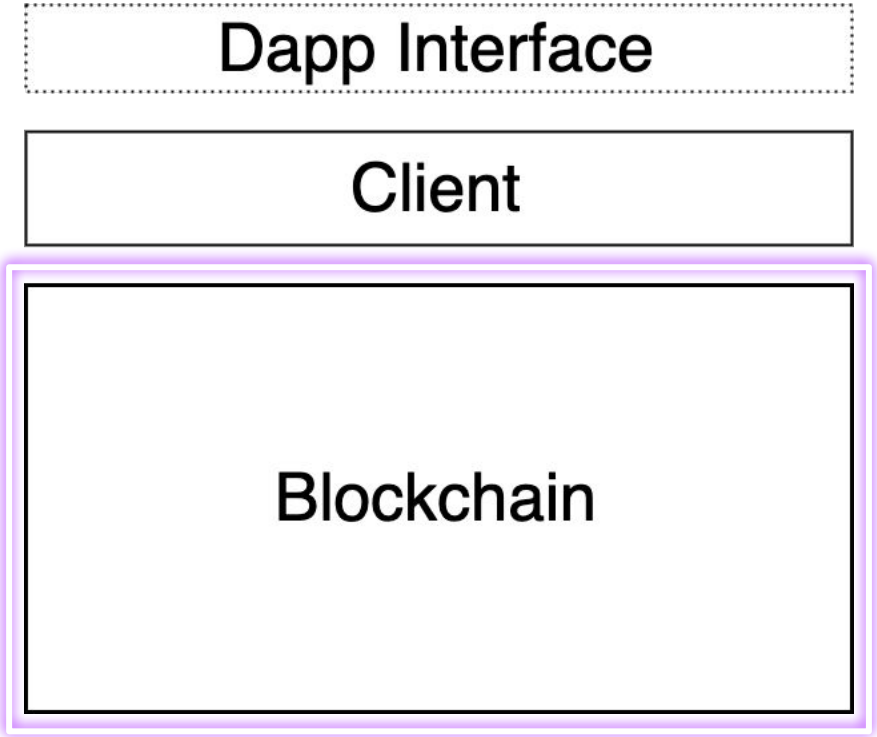



**Blockchain**



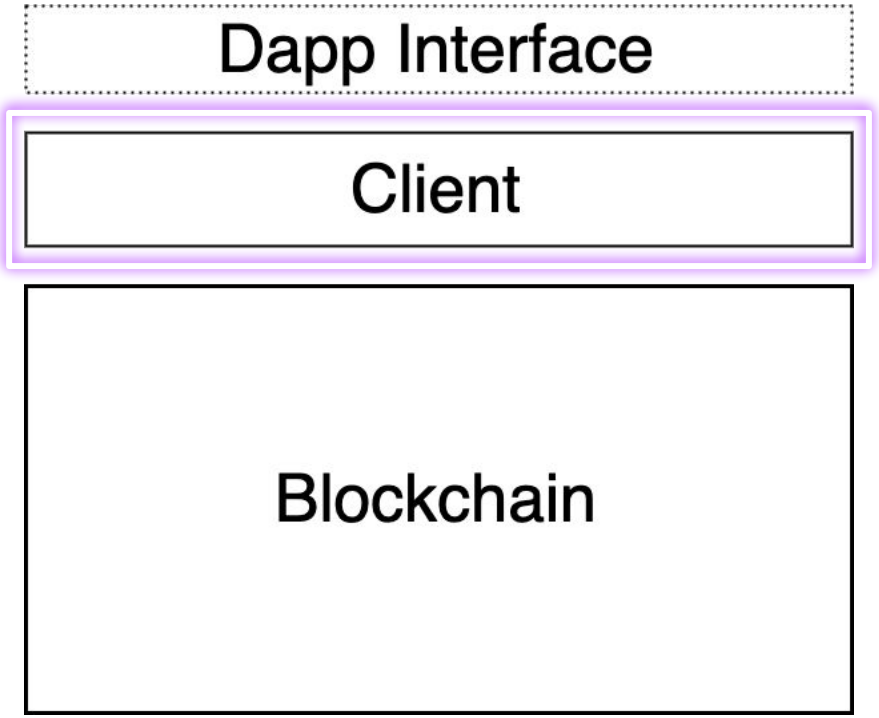

## Blockchain

- Handles Logic at protocol level

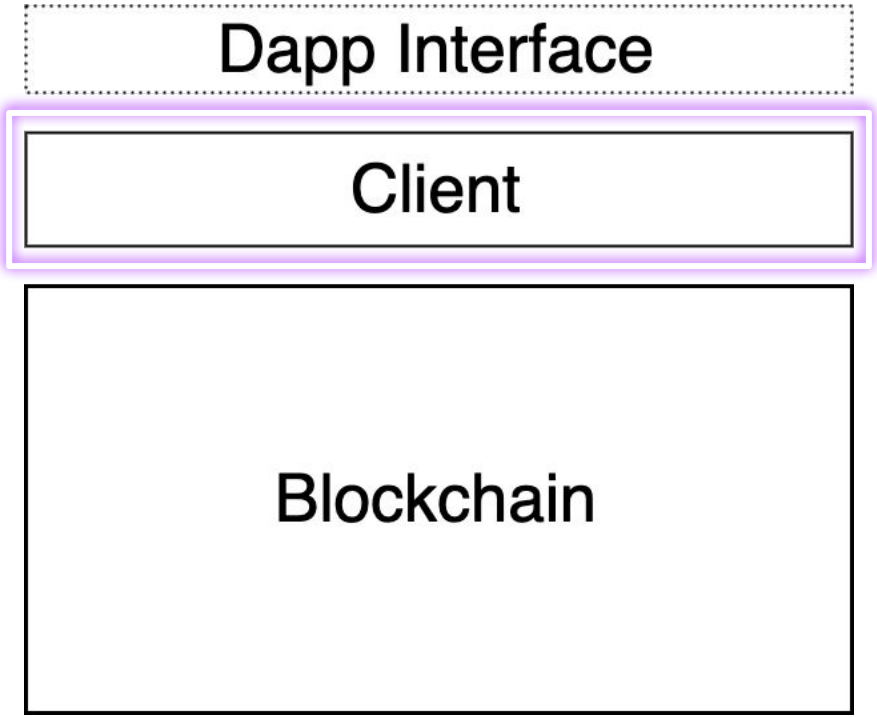



## Blockchain

- Handles Logic at protocol level
- Stores data on-chain

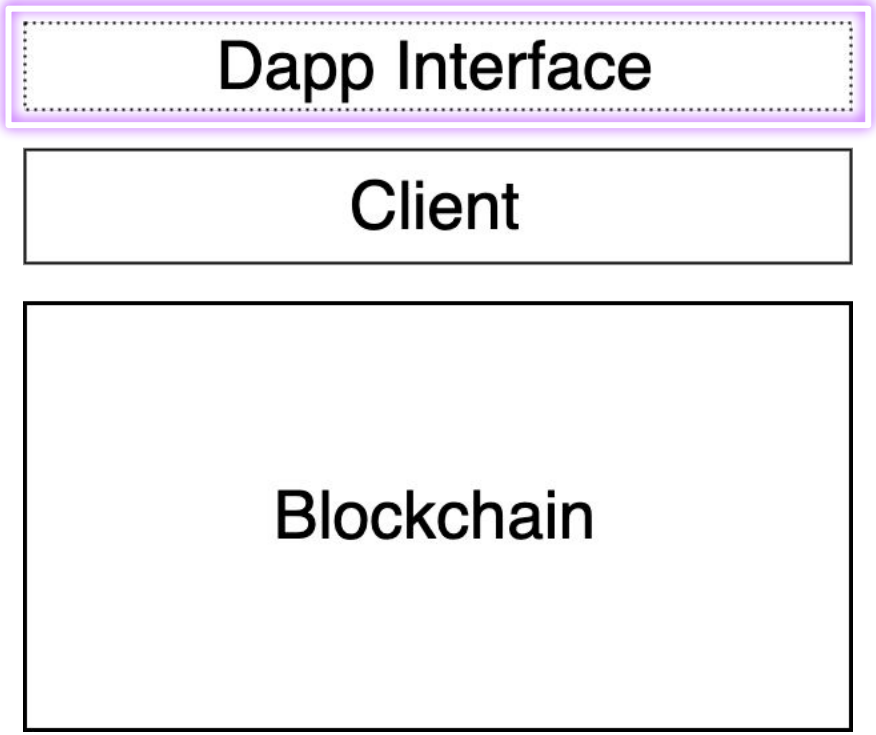



# Client

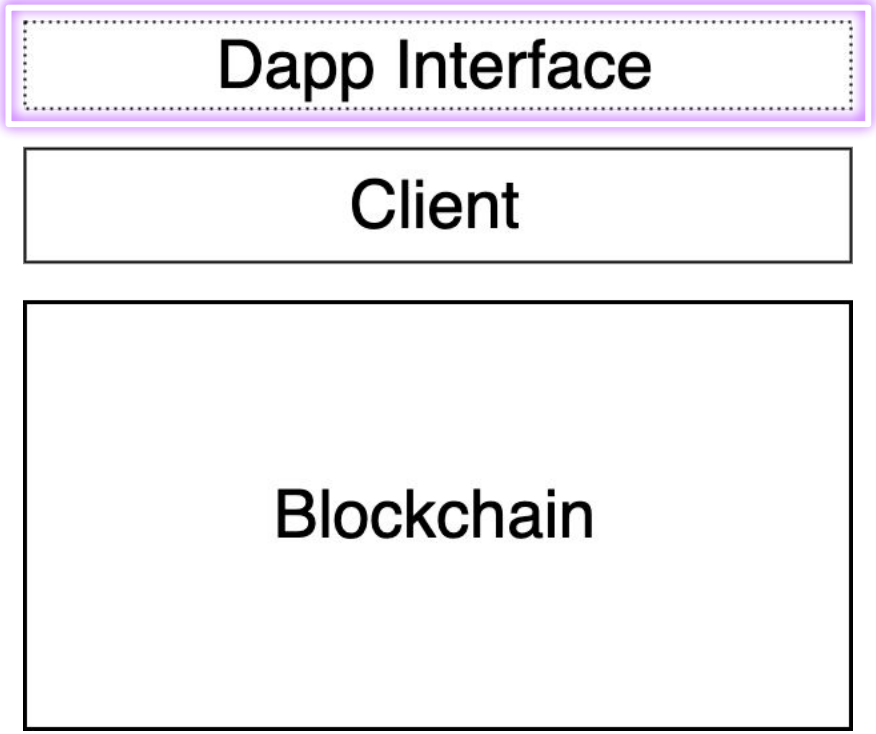



# Client

- **Facilitates interactions with the blockchain**

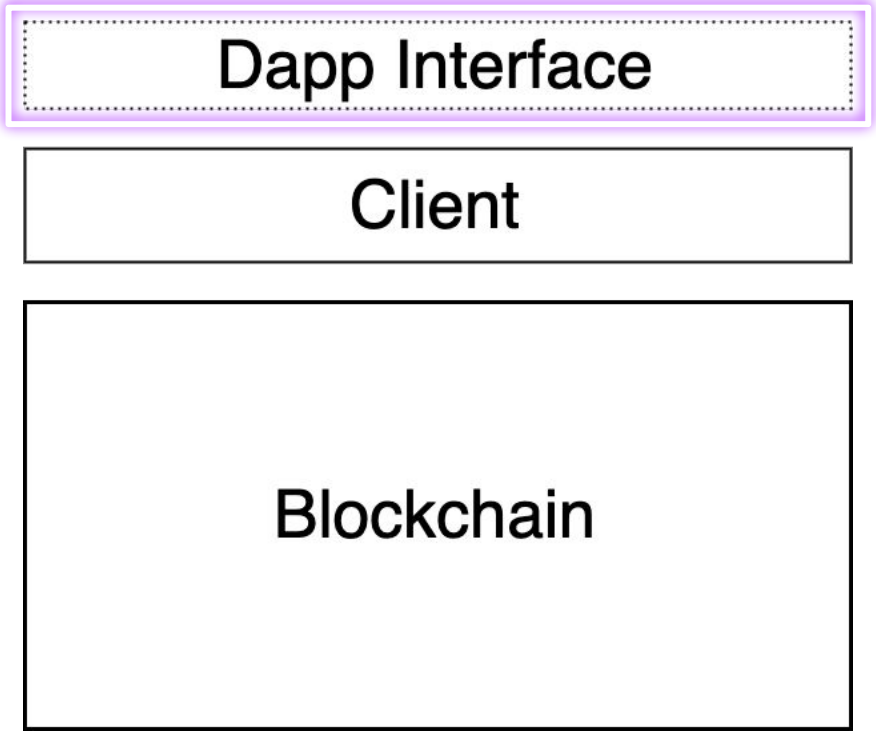



# Dapp Interface




# Dapp Interface

- Relies on client to interact with chain




# Dapp Interface

- Relies on client to interact with chain
- Like a visual skin for a protocol

# In A Nutshell

The **Dharma Settlement Contracts** are...

**A smart contract framework for tokenized debt agreements**

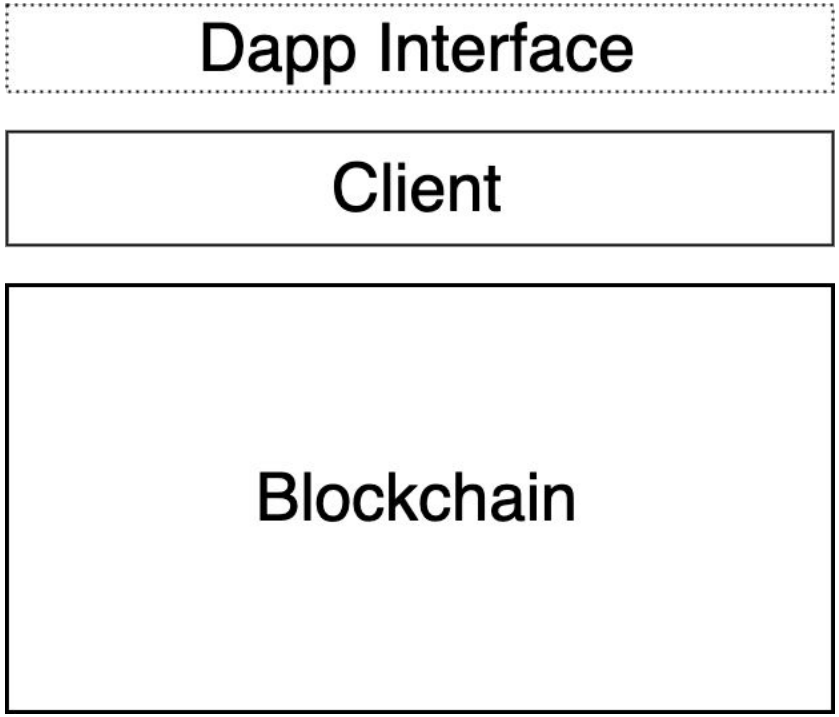
- Administer the **entire life-cycle of a loan** through smart contracts
- Collateral is **held in escrow in smart contracts** and released to creditors upon a borrower's default
- Creditor's stake in a loan is "tokenized" – it can be **traded, repackaged, and programmed** like any other token

**An open, permissionless credit market**

- A marketplace of **Relayers** who earn fees for hosting "order books" that connect borrowers and lenders
- A marketplace of **Underwriters** who earn fees for pricing borrower default risk
- A standardized message schema for connoting intent to borrow / lend, referred to as a **Debt Order**, enables increased liquidity through programmatic lending

**A generic and modular system**

- Virtually any type of debt agreement can be defined with a **Terms Contract**, be it a consumer margin loan or a corporate bond
- Developers can extend Dharma Protocol by programming new **Terms Contracts** for radically different types of debt agreements
- Terms Contracts have a standard interface that makes it easy for developers to build **credit derivatives, structured financial products, insurance contracts, and more**




**dApp Interface**

**Client**

**Blockchain**

# This is where the industry is largely at today

# Examples:
# Compound, Uniswap

# Let’s explore what this looks like for users

Compound Protocol

app.compound.finance

Apps | Want to Read | furniture | Shopping | work | Metabase | Reveal Bullshit | Coveralls | Dharma Discord | Eng Meeting Pres... | Dharma's Next Ph... | Compound | Devel... | User Feedback | Product Communi...

Compound

USD | Connect Wallet

Supply Balance

...

owing Power

Available

| Market | | | Borrow APR |
|---|---|---|---|
| Basic Attention Tok | | | 3.38% |
| Dai | | | 11.57% |
| Ether | | | 2.52% |
| Augur | | | 2.31% |
| USD Coin | | | 9.99% |
| Wrapped BTC | $1.24M | 1.03% | 6.94% |
| 0x | $2.41M | 0.19% | 3.71% |

**Connect Wallet**

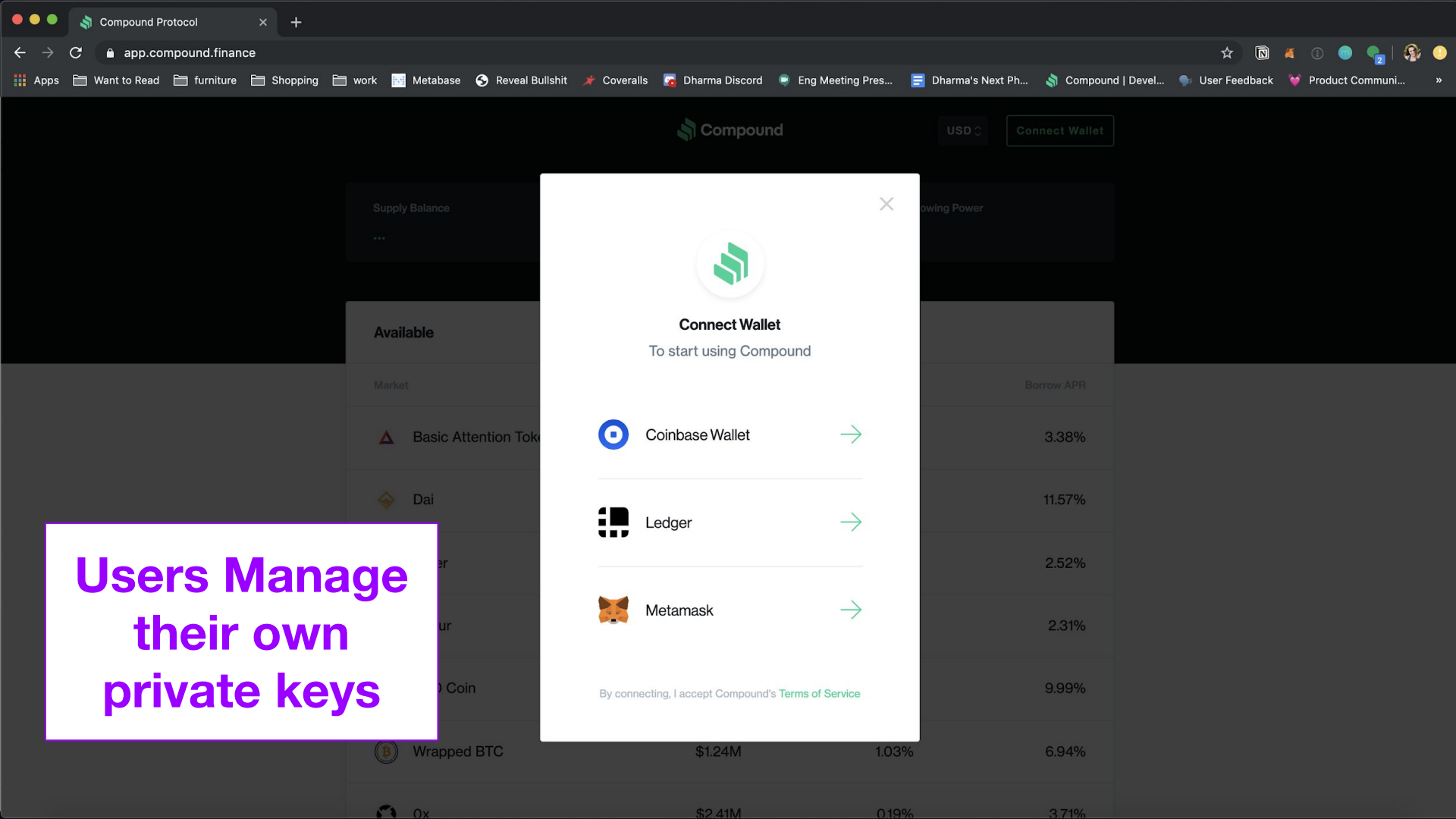
To start using Compound

Coinbase Wallet

Ledger

Metamask

By connecting, I accept Compound's Terms of Service




**Users Manage their own private keys**

MetaMask Notification

2 of 2
requests waiting to be acknowledged

Main Ethereum Network

Anna Workin... → 0x3d98...Cd3B

CONTRACT INTERACTION

0

$0.00

DETAILS DATA

EDIT

GAS FEE 0.003558
$0.66

AMOUNT + GAS FEE

TOTAL 0.003558
$0.66

Reject Confirm

REJECT 2 TRANSACTIONS

work Metabase Reveal Bullshit Coveralls Dharma Discord Eng Meeting Pres... Dharma's Next Ph... Compound | Devel... User Feedback Product Communi...

Compound USD 0X7A...5E7E

Supply Balance
$0.00

Compound allows you to borrow any supported asset.

ENABLE BORROWING

## Available

| Market | Market Size | Supply APR | Borrow APR |
|---|---|---|---|
| Basic Attention Token | $1.72M | 0.14% | 3.38% |
| Dai | $39.36M | 6.01% | 11.57% |
| Ether | $79.82M | 0.04% | 2.52% |
| Augur | $6.48M | 0.02% | 2.31% |
| USD Coin | $32.87M | 4.50% | 9.99% |
| Wrapped BTC | $1.24M | 1.03% | 6.94% |
| 0x | $2.41M | 0.19% | 3.71% |

MetaMask Notification

2 of 2
requests waiting to be acknowledged

Main Ethereum Network

Anna Workin... → 0x3d98...Cd3B

CONTRACT INTERACTION

0

$0.00

DETAILS DATA

EDIT

GAS FEE 0.003558
$0.66

AMOUNT + GAS FEE

TOTAL 0.003558
$0.66

Reject Confirm

REJECT 2 TRANSACTIONS

work Metabase Reveal Bullshit Coveralls Dharma Discord Eng Meeting Pres... Dharma's Next Ph... Compound | Devel... User Feedback Product Communi...

Compound USD 0X7A...5E7E

Supply Balance
$0.00

Compound allows you to borrow any supported asset.

ENABLE BORROWING

Available

| Market | Market Size | Supply APR | Borrow APR |
|---|---|---|---|
| Basic Attention Token | $1.72M | 0.14% | 3.38% |
| Dai | $39.36M | 6.01% | 11.57% |
| er | $79.82M | 0.04% | 2.52% |
| ur | $6.48M | 0.02% | 2.31% |
| Coin | $32.87M | 4.50% | 9.99% |
| Wrapped BTC | $1.24M | 1.03% | 6.94% |
| 0x | $2.41M | 0.19% | 3.71% |

**Users pay their own gas**

# Users deal with all the complexity of a distributed system

Queue (9)

| | | | |
|---|---|---|---|
| Sent Ether | #2 - 11/23/2018 at 16:30 | PENDING | -$8.68 USD<br>-0.081659 ETH |
| Cancel Attempt | #2 - 11/25/2018 at 00:39 | PENDING | -$0.00 USD<br>-0 ETH |
| Cancel Attempt | #2 - 11/25/2018 at 00:40 | PENDING | -$0.00 USD<br>-0 ETH |
| Cancel Attempt | #2 - 11/25/2018 at 00:40 | PENDING | -$0.00 USD<br>-0 ETH |
| Sent Ether | #3 - 11/25/2018 at 00:42 | PENDING | -$1.86 USD<br>-0.017487 ETH |
| Cancel Attempt | #3 - 11/25/2018 at 10:48 | PENDING | -$0.00 USD<br>-0 ETH |
| Cancel Attempt | #3 - 11/25/2018 at 10:53 | PENDING | -$0.00 USD<br>-0 ETH |
| Cancel Attempt | #2 - 11/25/2018 at 10:53 | PENDING | -$0.00 USD<br>-0 ETH |
| Cancel Attempt | #3 - 11/25/2018 at 10:53 | PENDING | -$0.00 USD<br>-0 ETH |

History

# My dad is never going to do this

# My dad is never going to do this

# We love to talk about wen mass adoption

# Yet look at what we expect of our users.

# Users need to...

- ➔ Manage private keys
- ➔ Possess a baseline technical understanding of the blockchain
- ➔ Pay gas
- ➔ Grant token allowances
- ➔ Handle dropped transactions

# Many of us have lost touch of how bad this is

# THIS SUCKS FOR USERS

# You should never expect your users to manage everything going on under the hood

***Especially*** **when**
**“under the hood”**
**=**
**the complexity of the blockchain**

# People just want to use the product

**How can we give users the experience they expect**

***WHILE***

**still giving them the benefits of the blockchain?**

# Balancing
# On- & Off-Chain

# Let’s look at the pieces of the System

# Frontend

# Its still the UI that users interact with

# Minimising pain

# Minimising complexity

# Minimising latency

# Backend

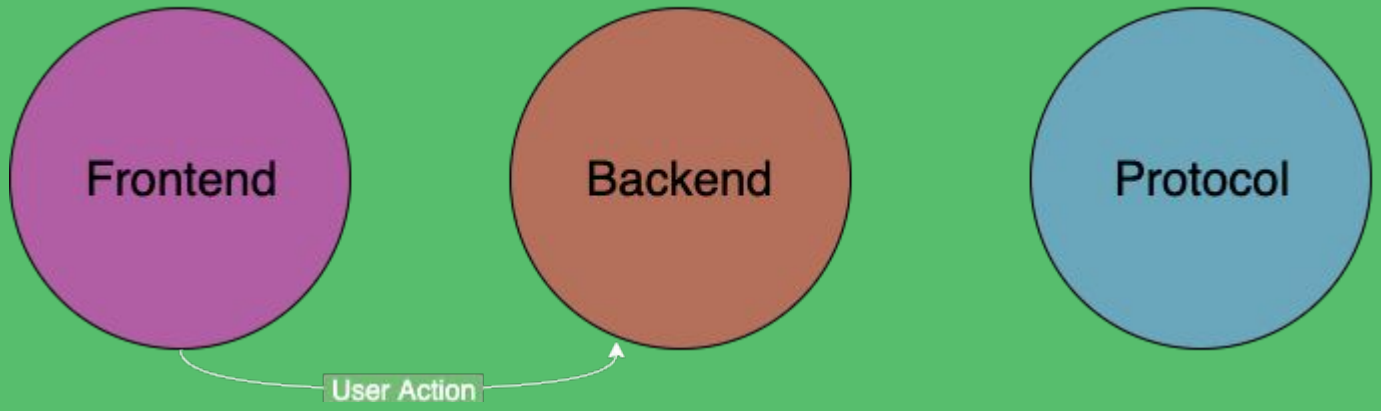
# The backend is a two-way facilitator user ↔ blockchain

# The backend is a two-way facilitator

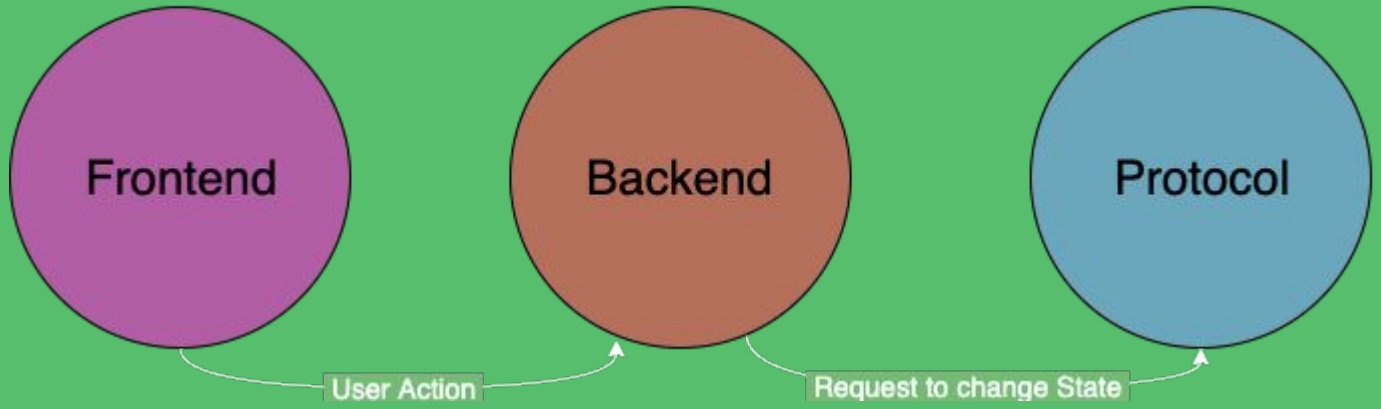
**user ↔ blockchain**

# User → Backend → Blockchain

# User → Backend → Blockchain

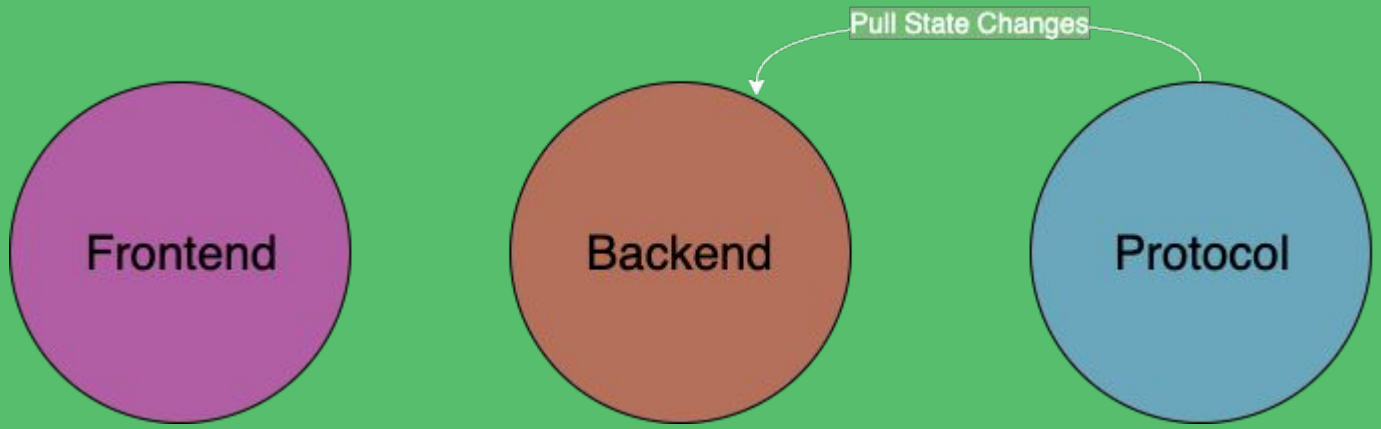


# User → Backend → Blockchain

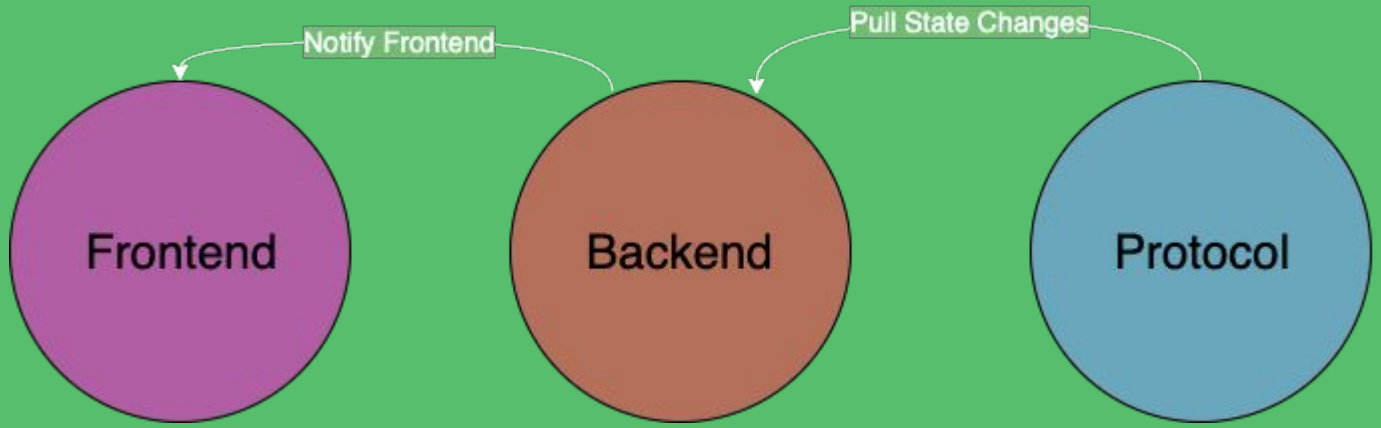


# Blockchain → Backend → User

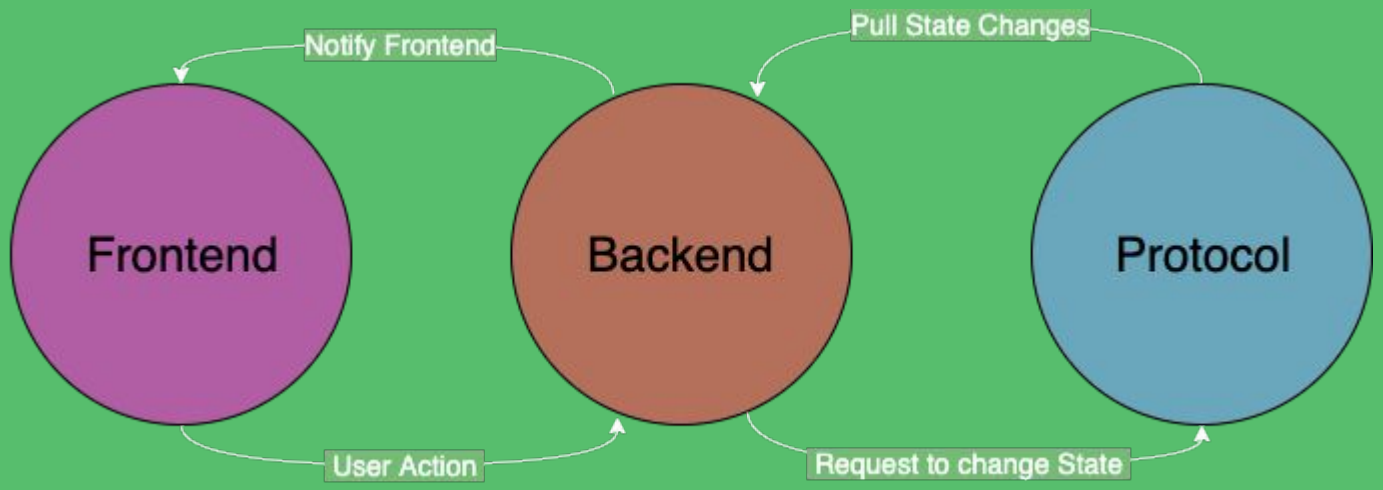
# Blockchain → Backend → User



# Blockchain → Backend → User



# Two-Way Flow


Notify Frontend
Pull State Changes
Frontend
Backend
Protocol
User Action
Request to change State

# Database

# Database VS Blockchain

# Two duplicated stores of your data exists

# In traditional architecture, DB is a source of truth

# Your database & the blockchain always need to agree, or you've got a problem!

# Whenever there's an argument between the two...

# THE

# BLOCKCHAIN

# ALWAYS

# Wins

# Database
# =
# low-latency cache of blockchain

# Parsed & transformed into a format that's easier for the frontend to consume

# Blockchain

# Blockchain is the Settlement Layer of the stack

# It stores user funds

# It executes state changes on those funds

# It acts as the Source of Truth for “what happened” to those funds

# Enables user to custody their funds

# Excuses company from custodying their funds

# Properties of their Interactions

# Immutable

# Blockchain is source of truth & the database is a reflection of it

# When the blockchain is immutable

# The DB has to be also!

# So we have a few simple rules

# *Rule 1*

**Database models should never be overwritten or modified**

# Instead of

Users Tables

| ID | Name | Deleted |
|---|---|---|
| 1 | Fred | True |
| 2 | Sarah | Null |

# Instead of

Users Tables

| ID | Name | Deleted |
|---|---|---|
| 1 | Fred | True |
| 2 | Sarah | Null |

# Try

Users Tables

| ID | Name |
|---|---|
| 1 | Fred |
| 2 | Sarah |

Deleted Table

| ID | UserId |
|---|---|
| 1 | 1 |

# *Rule 2*

**It should ALWAYS be possible to recreate history via the DB**

# Deposits and Withdrawals

| ID | UserId | Amount | Type | Date |
|---|---|---|---|---|
| 1 | 1 | 15 | Deposit | May 1 |
| 2 | 1 | 10 | Withdrawal | May 2 |
| 3 | 1 | 5 | Deposit | May 3 |
| 4 | 1 | 2 | Withdrawal | May 4 |

# *Rule 3*

**Always leave a paper trail of actions on chain**

# Idempotent



# Remember how we talked about developer tooling in the Web 2.0 world?


INFURA

# In crypto, many of these tools are immature

WEB3

# It’s important to be able to handle outages (they *will* happen often)

> “Idempotence is the property of certain operations in mathematics and computer science whereby they can be applied multiple times without changing the result beyond the initial application.”

# Idempotency allows you retry operations with peace of mind

# Idempotency allows you to write code that handles the worst case scenarios

# While also functioning perfectly in the happy case

# This gives you stronger guarantees that your system will remain consistent with the blockchain

# And be able to handle failures gracefully

# Event-Driven

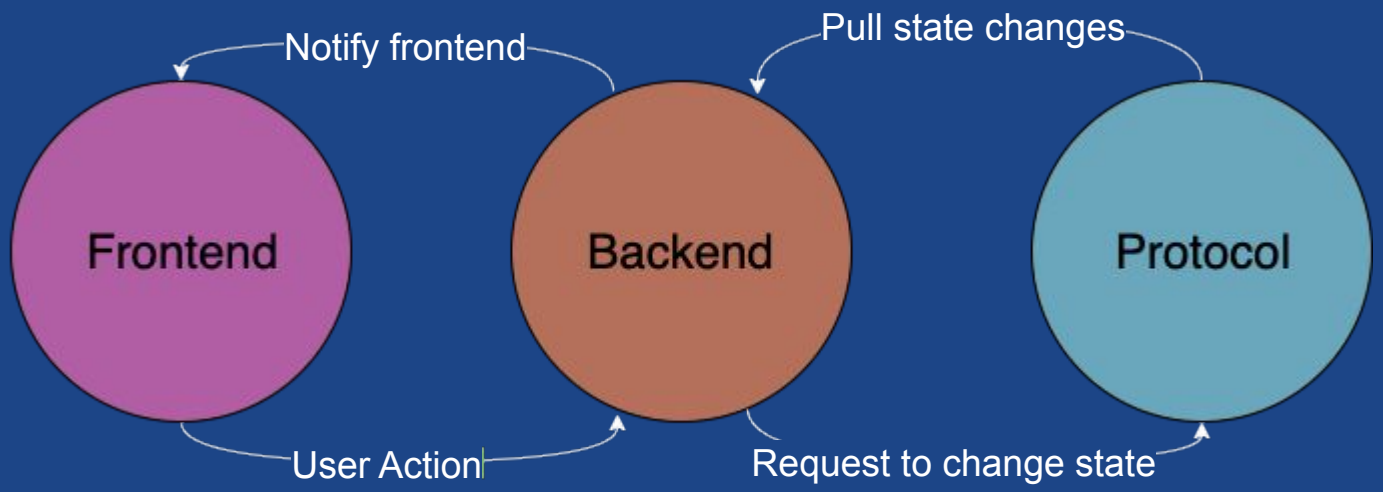
# Adding code to a system can be tricky as the code base gets large

# As you start to interact with many smart contracts things can get *tricky*

# So we need a simple mental model to write our code

# Adding code should be simple to slot into the event stream

# Event-Driven



# Conclusion

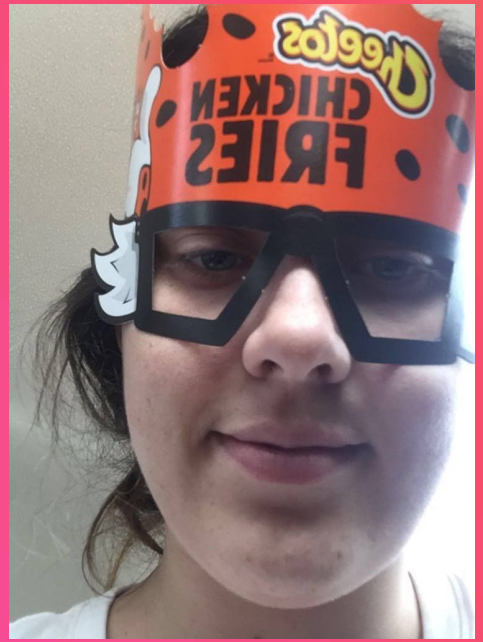
# A balanced architecture gives you….

# Awesome UX

$1,615,695.6467

# Blockchain Magic

# Questions?




@annascarroll

dharma.io

@neablist