

# Substrate

An open source framework for  
building blockchains

---

**Shawn Tabrizi**

Software developer @ Parity Technologies Ltd.

shawn@parity.io | @shawntabrizi



# Shawn Tabrizi

Software developer @ParityTech

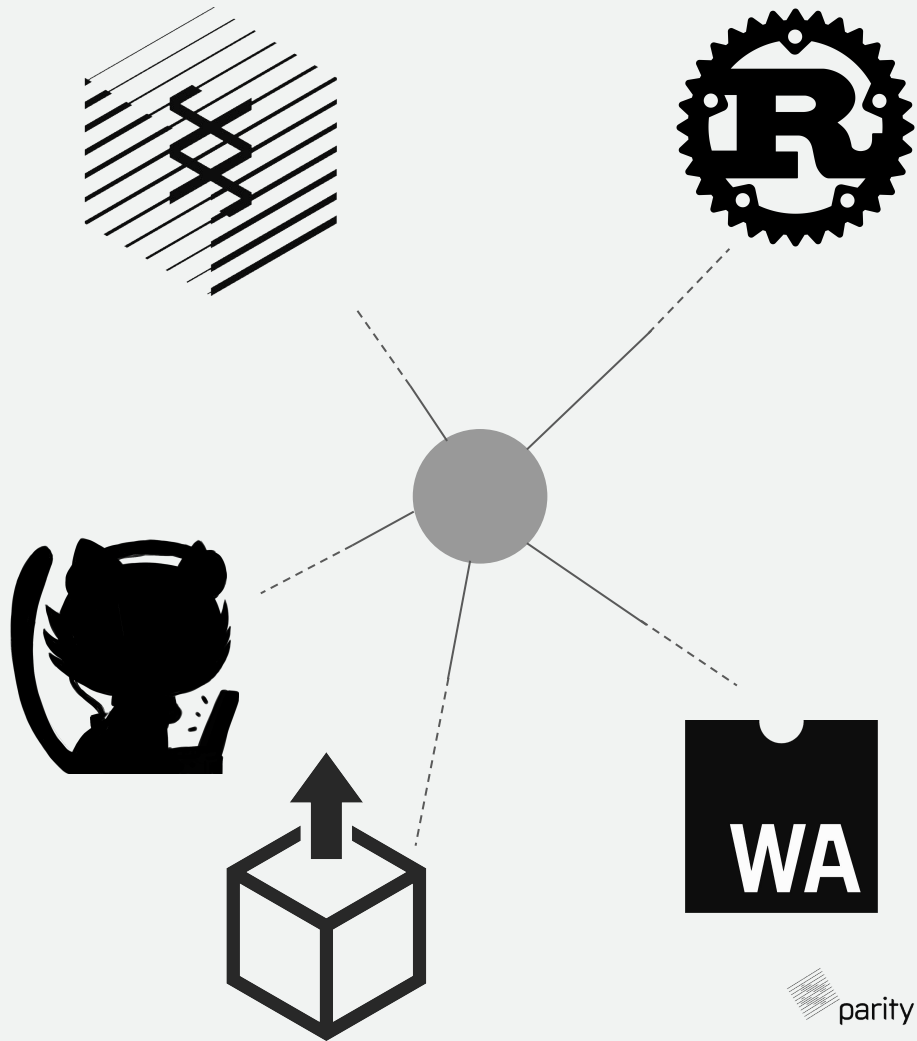
Background:

- Identity and Authentication
- Cloud Infrastructure
- Ethereum DApp Development
- Substrate Development

GitHub / Twitter: @shawntabrizi

# We will talk about:

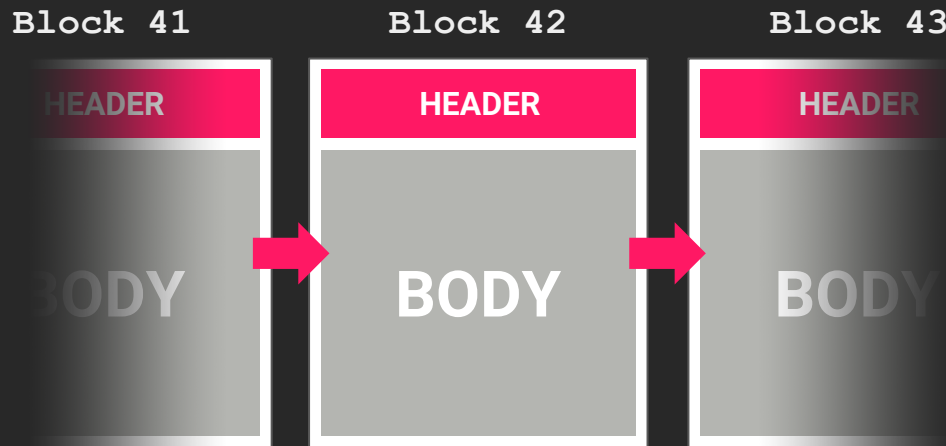
- Substrate
- Rust
- Wasm
- Blockchain Upgrades
- Building on Substrate



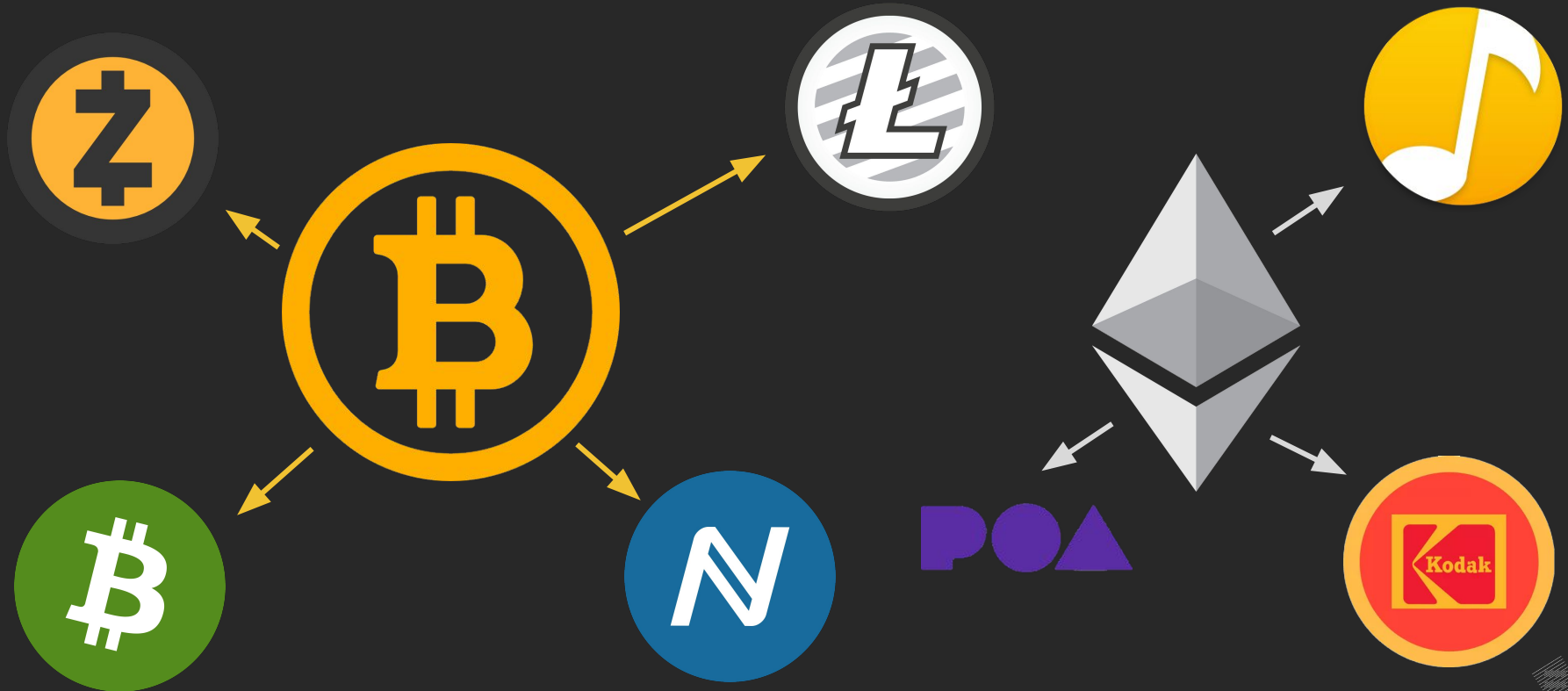
# A quick review of blockchains

## Blockchain nodes need:

- Database
- P2P Network
- Consensus Algorithm
- Transaction Handling
- State Transition Function (Runtime)

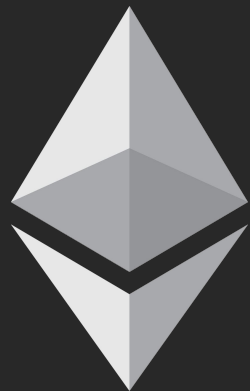


# How does blockchain development happen today?



# Parity has a lot of blockchain building experience...

---



 [github.com/paritytech/](https://github.com/paritytech/)  
**parity-ethereum**



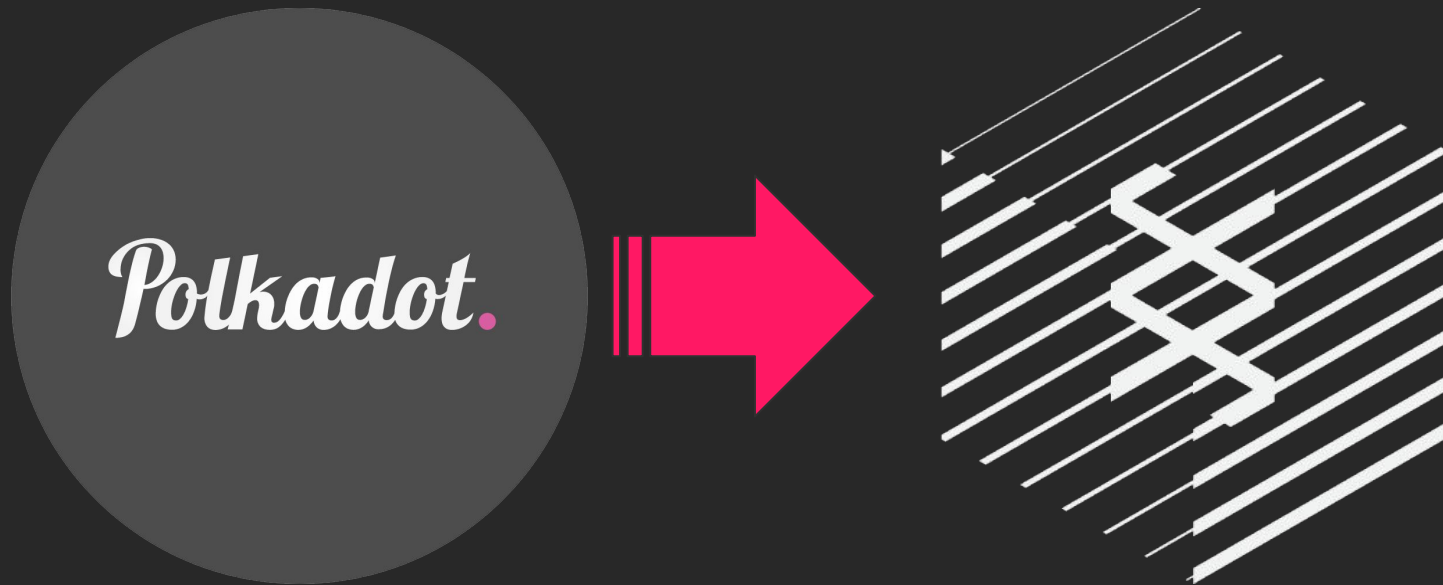
 [github.com/paritytech/](https://github.com/paritytech/)  
**parity-bitcoin**



 [github.com/paritytech/](https://github.com/paritytech/)  
**polkadot**

# From Polkadot, came Substrate.

---



# What is Substrate?

---

Substrate is an **open source, modular, and extensible** framework for building blockchains.





# What is Substrate?

---

## Substrate Provides:

- Database Layer
- Networking Layer
- Consensus Engine
- Transaction Queue
- Library of Runtime Modules



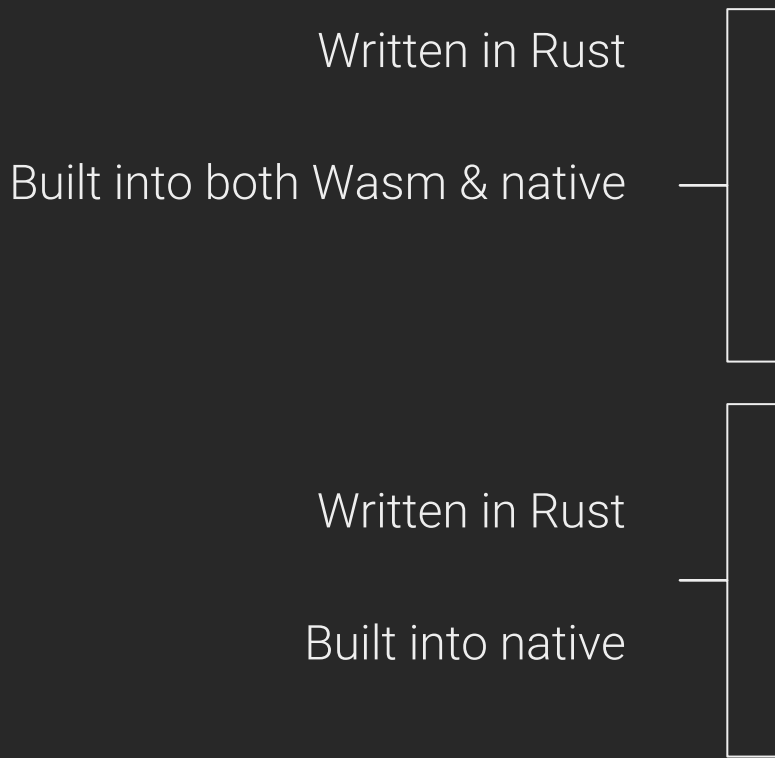
**Each of which can be customized and extended.**

# What else do you get with Substrate?

---

- Interchain connectivity via Polkadot
- Hot-swappable, pluggable consensus
- Light client
- Chain synchronisation
- Pub/Sub WebSocket JSON-RPC
- JS Libraries/Tooling
- Telemetry
- Forkless runtime upgrades





Building components of Substrate

# Substrate is written in Rust.

---

# Why Rust?

---

Parity is a **Rust team** as much as it is a blockchain team

- Rust is Safe
- Rust is Fast
- Rust is Lightweight
- Rust is Idiomatic
- Rust is Fun



<https://parity.io/why-rust/>

# Why WebAssembly?

---

Wasm is a platform independent executable format

- Wasm is Compact
- Wasm is Sandboxable
- Wasm is Deterministic
- Wasm is Near Native Speed
- Wasm is Well Supported



# Rust for Wasm

---

...and these two technologies work really well together!

- Rust compiles directly to Wasm
- Native and Wasm binaries use the same runtime codebase
- Parity has built a Wasm interpreter in Rust
- Small .wasm files



Written in Rust

Built into both Wasm & native

Wasm stored on-chain

Written in Rust

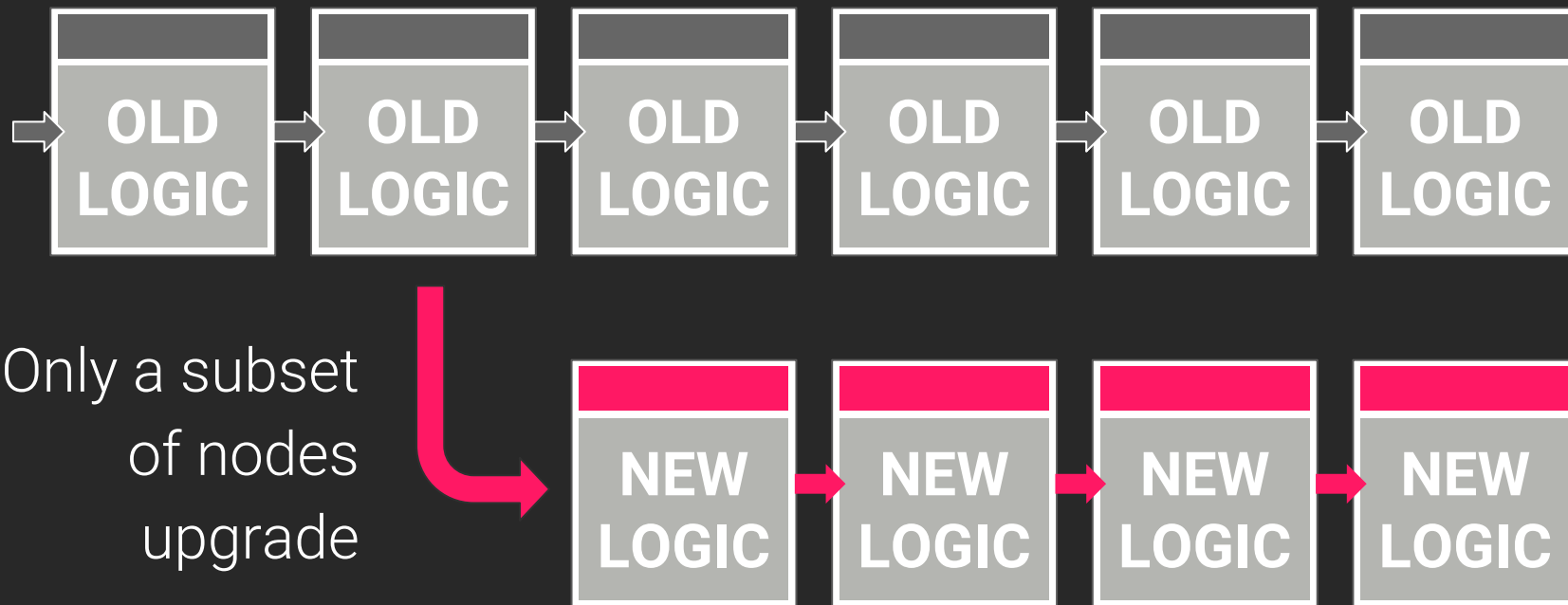
Built into native



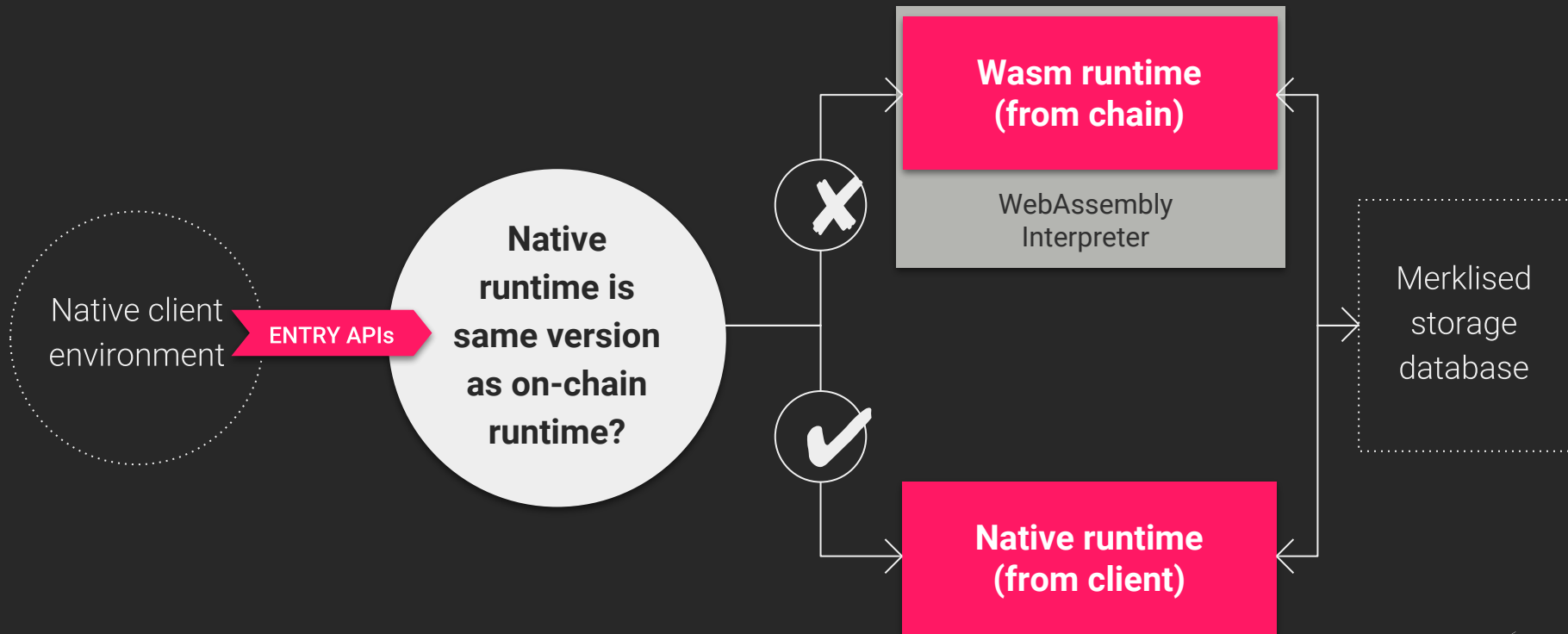
Building components of Substrate



# Forked Blockchain Upgrades



# Forkless Runtime Upgrades

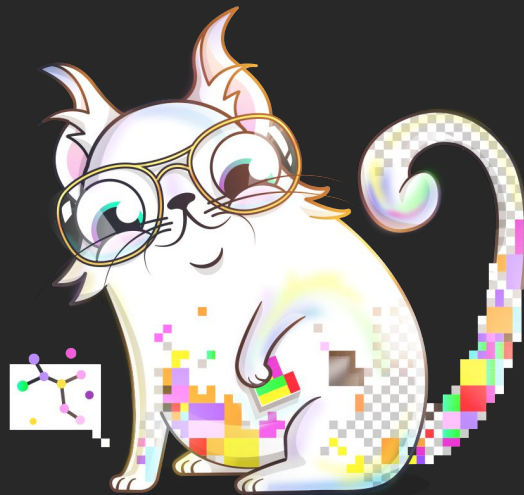


# Context: The CryptoKitties Craze

---

Cats almost broke the internet... again.

- Non-fungible tokens are unique
- CryptoKitties represents these tokens as digital pets
- Peak popularity in December 2017
- Almost “broke Ethereum”
- What if Substrate was available...?



<https://www.cryptokitties.co/>

# Let's do an upgrade...

---

Live Demo

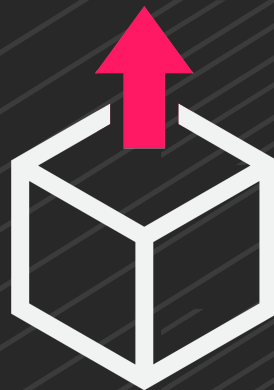
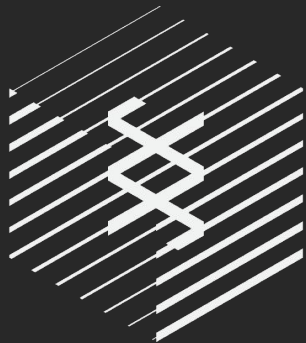
# Substrate Collectables Workshop

- Run a local Substrate node
- Learn about runtime development and best practices
- Build a working chain with UI
- Minimal Rust Experience



Kitty by David Revoy

[tiny.cc/paritySubstrateWorkshop](https://tiny.cc/paritySubstrateWorkshop)



# We're hiring!

---

[parity.io/jobs](https://parity.io/jobs)

# Parity events and updates

---

[parity.io/newsletter](https://parity.io/newsletter)



# Questions?

---

shawn@parity.io  
@shawntabrizi