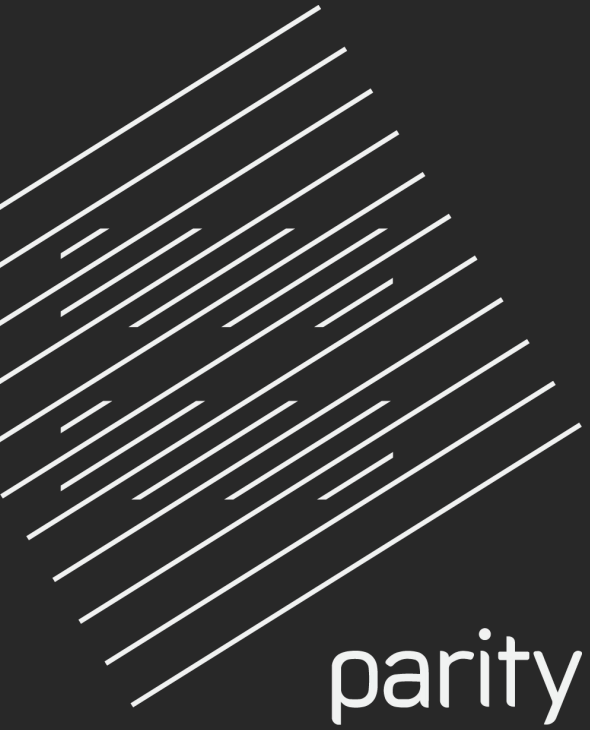


Enabling forkless blockchain upgrades with Substrate

Shawn Tabrizi
Software developer @ Parity Technologies Ltd.

shawn@parity.io | @shawntabrizi

Shawn Tabrizi

Software developer @ParityTech

Background:

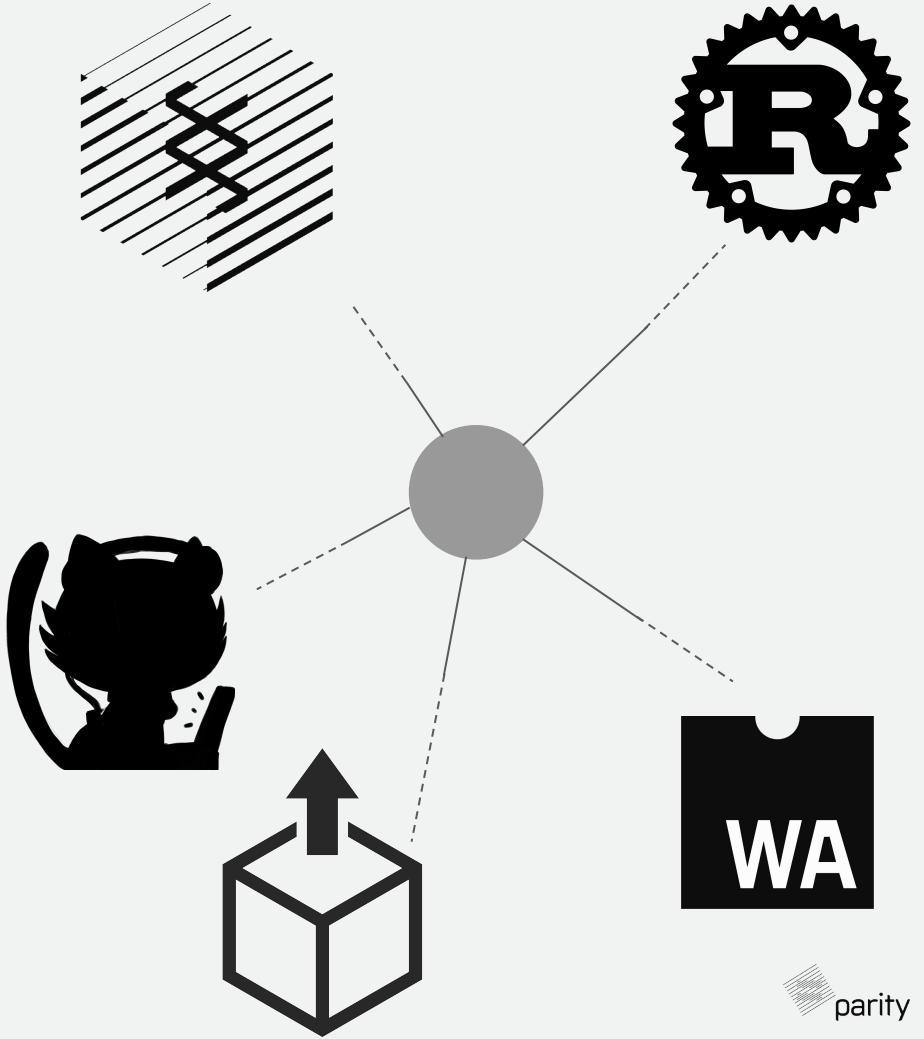
- Identity and Authentication
- Cloud Infrastructure
- Ethereum DApp Development
- Substrate Development

GitHub / Twitter: @shawntabrizi

parity

We will talk about:

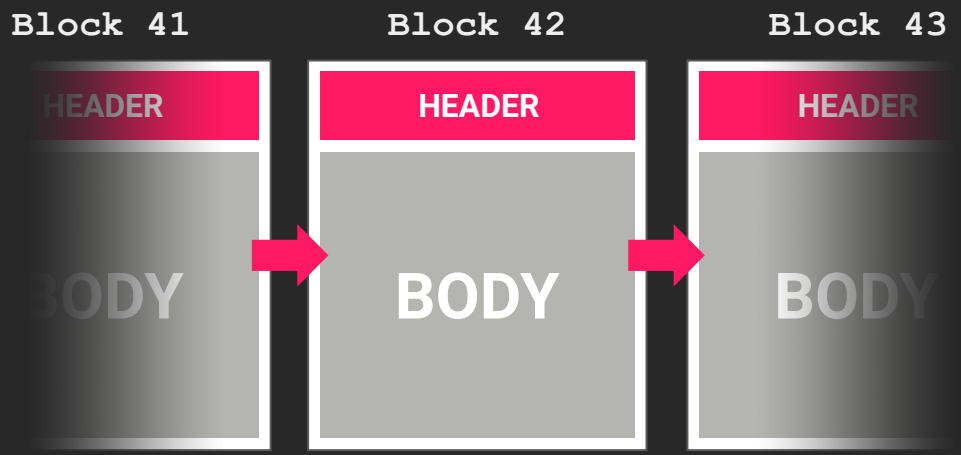
- Substrate
- Rust
- Wasm
- Blockchain Upgrades
- Building on Substrate



A quick review of blockchains

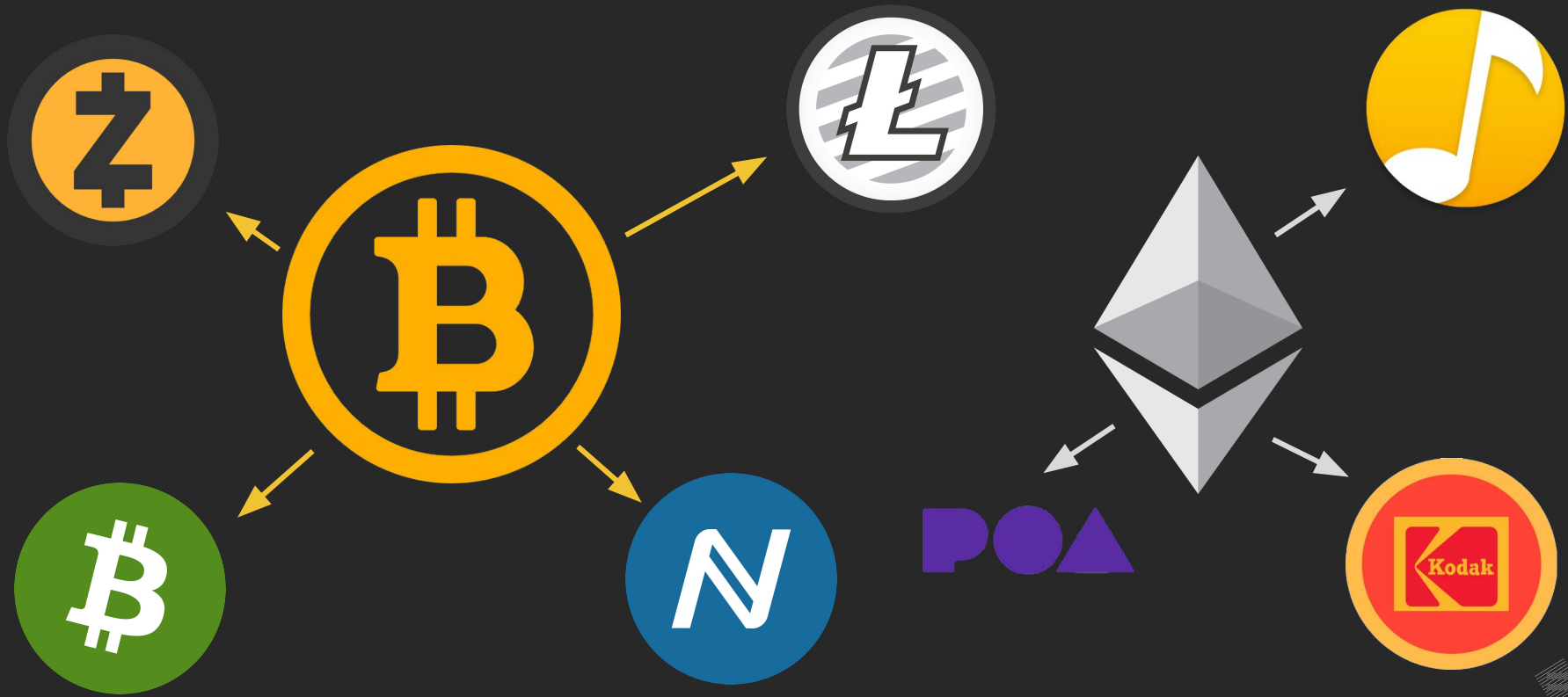
Blockchain nodes need:

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

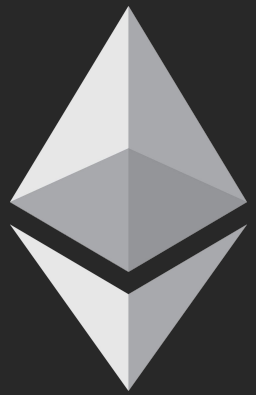


parity

How does blockchain development happen today?



Parity has a lot of blockchain building experience...

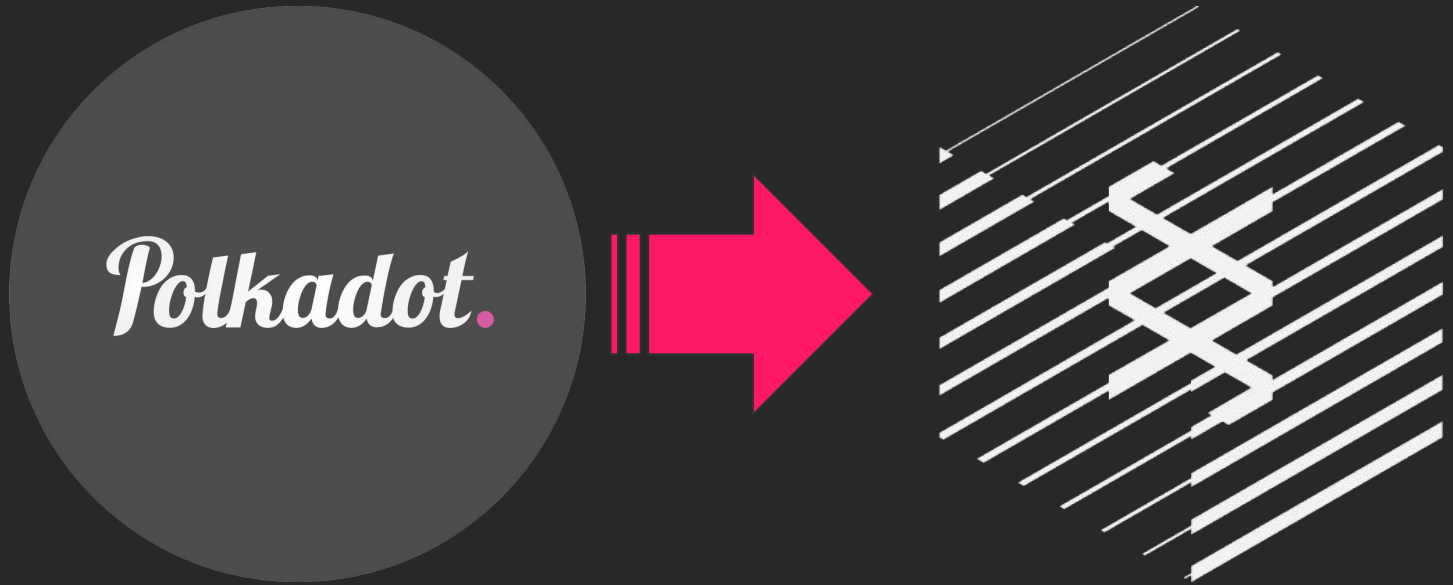


github.com/paritytech/
parity-ethereum

github.com/paritytech/
parity-bitcoin

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 all the core components of a Blockchain:

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

Each of which can be customized and extended.

parity

What is a Runtime?

The runtime is the **block execution logic** of the blockchain, a.k.a. the State Transition Function.

It is composed of **Runtime Modules**.

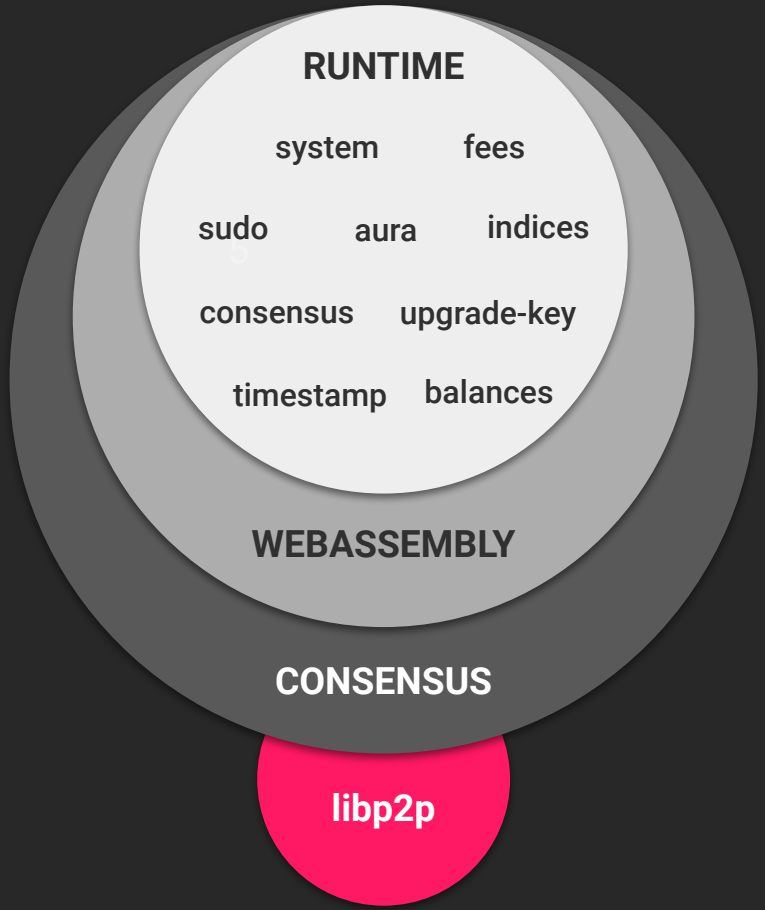
Substrate Runtime Module Library (SRML)			
assets	aura	balances	consensus
contract	council	democracy	executive
fees	grandpa	indices	metadata
session	staking	sudo	system
timestamp	treasury	upgrade-key	and more...

Written in Rust

Built into both native & Wasm

Written in Rust

Built into native



Building components of Substrate

parity

Substrate is built using Rust and Wasm.

Why WebAssembly?

Wasm is a platform independent executable format

- Wasm is Sandboxed
- Wasm is Fast
- Wasm is Compact
- Wasm is Well Supported

Why Rust?

Parity is a Rust company as much as it is a blockchain company.

- Rust builds directly to Wasm
- Native and Wasm binaries use the same runtime codebase
- No garbage collector overhead
- Optimization removes dead code
- Resulting in small .wasm files

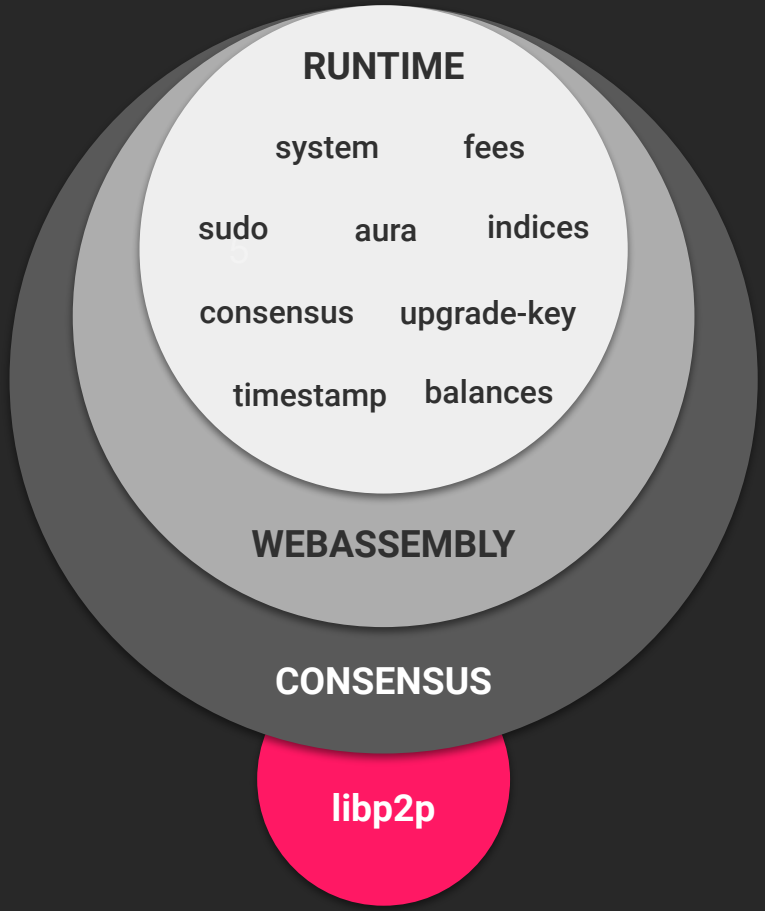
Written in Rust

Built into both native & Wasm

Wasm stored on-chain

Written in Rust

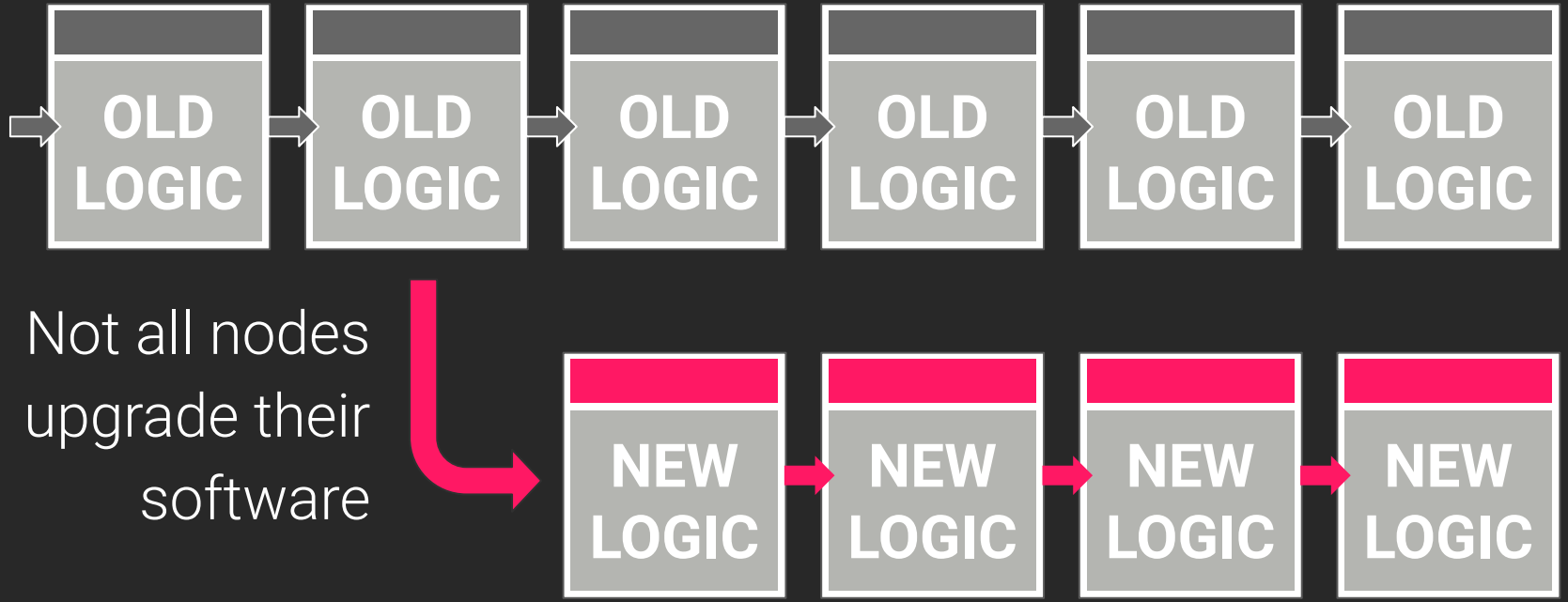
Built into native



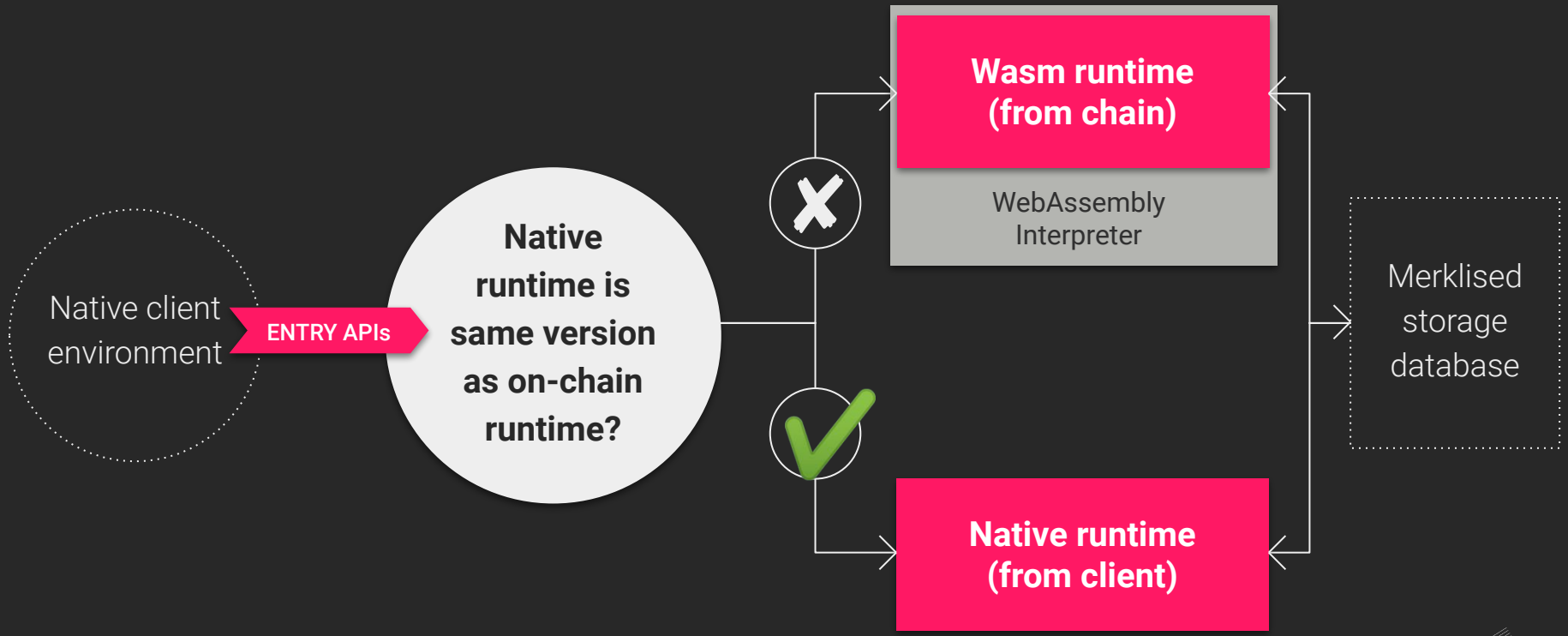
Building components of Substrate

parity

Hard Fork Upgrades



Forkless Runtime Upgrades



parity

A Need for Upgrades

- Fix important security vulnerabilities
- Change core rules in the protocol
- Add new functionality
- Repair chain state

➢ Hard fork upgrades require a lot of coordination

➢ Unclear governance/signaling around upgrades

Governing Runtime Upgrades

- Runtime code is accessible through on-chain governance
- Sudo Module
- Democracy Module
- Your own module and logic
- **Runtime Upgrades are Optional**

parity

Extending and Upgrading Your Substrate Runtime

- We created a demonstration runtime based on the popular CryptoKitties DApp
- Runtime upgrades allow you to extend and improve your DApp Chains
  - Add new features and functions
  - Add new storage items
  - Repair or upgrade existing storage
- What would it look like to upgrade this runtime?

https://www.cryptokitties.co/

Let’s do an upgrade...

Live Demo

parity

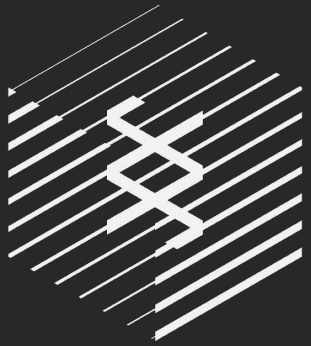
Kitty by David Revoy

Substrate Collectables Workshop

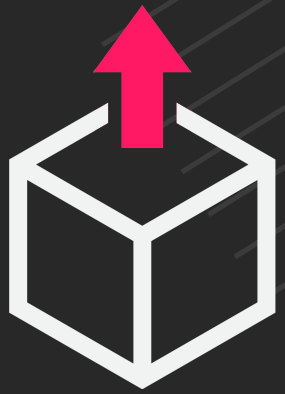
- ➔ Run a local Substrate node
- ➔ Learn about runtime development and best practices
- ➔ Build a working chain with UI
- ➔ Minimal Rust Experience

substrate.dev/substrate-collectables-workshop

parity




WA



We’re hiring!

parity.io/jobs

parity

Parity events and updates

parity.io/newsletter

parity

Questions?

shawn@parity.io
@shawntabrizi

parity