

Beyond Smart Contracts

with Substrate

Dotsama

Relay chain provides security, Validators securing the network

Appchains (parachains), secured by Collators, provide specific business functionality:

- common good parachains, eg. Statemint (reduced fees), bridges
- appchain examples:
 - defi - Acala, Parallel Finance
 - identity - Litentry
 - IOT - Nodle
- secured by auctions - **pricey entry point**

Runtime

- state transition function with chain specific logic
- WASM or native
- gossip via intrinsics
- collection for pallets
- versioned

Pallets

- runtime building blocks
- comprise:
 - storage (simple, map, doublemap)
 - events
 - errors
 - extrinsics (aka transactions)
 - hooks (on_initialize(), on_finalize(), on_upgrade())
 - offchain worker (eg. chainlink)
- eg. sudo (ops), balances (currencies), contracts, democracy (gov), evm
- testable within mock runtime

Extrinsics

Aka external transactions

- origins:
 - root (need `sudo`) - potential for centralization
 - signed - any user
 - unsigned - utilized in OCW, but prone to DDOS, alleviated by Substrate hooks
- weighted - pre-calculates gas cost, inbuilt into pallet code
 - templates with benchmarking out of the box (for simple cases)
 - for more complex eg. loops - business logic interwoven with weight calculation logic - **code smell**

Offchain worker

- triggered by “block seal”, eg. periodic or when transaction is queued
- can submit transactions to onchain
- can write to shared offchain DB
- opted in by nodes
- `non_std` libs such as collections, serialization, HTTP client

XCMP

Interop between relay chains, parachains/threads

XCM - message format

XCMP - machinery to push messages

- executor
- pallet

Forkless deployment

- code deployed as a prioritized single block transaction
- updates runtime & it's pallets
- caters for data migrations

Test/Canary networks

Kusama - canary network, gets quicker updates, less stable

test nets:

- Rococo
- Westend

Snags

- Rust types:

- `type BalanceOf<T> = <<T as Config>::Currency as Currency<<T as frame_system::Config>::AccountId>>::Balance;`

- substrate macro magic

- poor modularity of code

- lives in github, not published to crates.io

Demo

```
git clone git@github.com:konrads/substrate-node-playground.git
```

```
git diff forkless-demo-t0..forkless-demo-t1
```

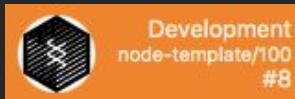
```
git checkout forkless-demo-t0 && cargo clean && cargo build && mv target/debug/node-template  
target/debug/node-template-t0 && git checkout forkless-demo-t1 && cargo build
```

```
target/debug/node-template-t0 --dev --tmp
```

<https://polkadot.js.org/apps/#/explorer>

As Alice: Developer > Extrinsic > sudo > sudoUncheckedWeight() > system > setCode() >
"target/debug/wbuild/node-template-runtime/node_template_runtime.compact.compressed.wasm"

See version change



-> 200, observe new postUpdateAck() extrinsic in playground pallet