
Polkadot.

W3F

Web3 Tech Stack_

DApp

Protocol-extensible user-interface cradle (“browser”)

Protocol-extensible developer tools, APIs & languages

L2

Second layer protocols

State channels | Oracles | Encrypted storage | Storage incentivization | Heavy computation | Distributed secret management | Plasma protocols | Governance

L1

Zero and low-trust interaction protocols (blockchains, DAGs)

Zero and low-trust interaction platforms (shared security)

Data distribution protocols

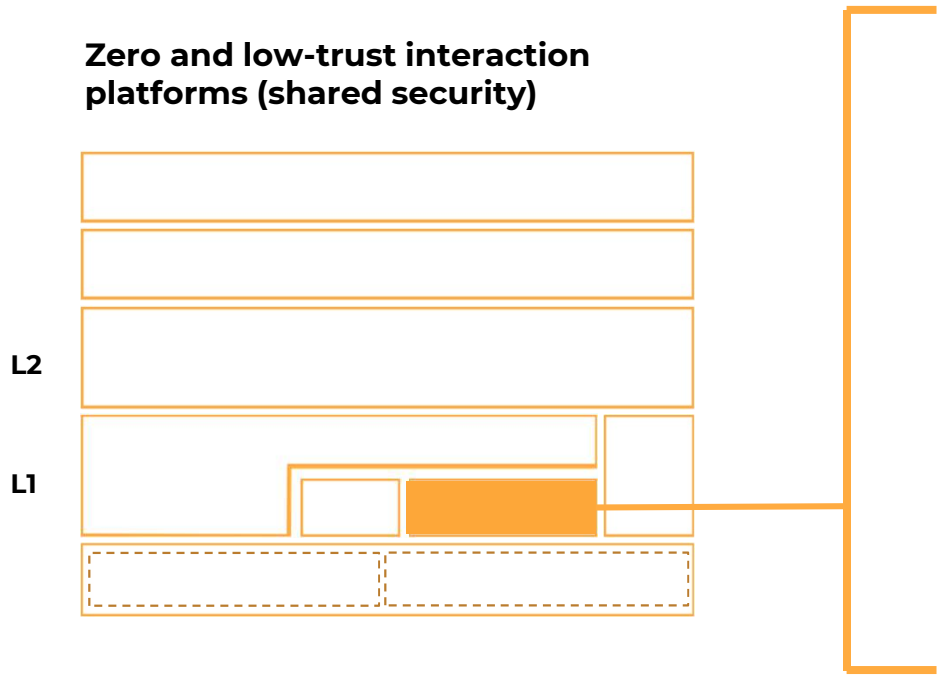
Transient data pub/sub messaging

Peer-to-peer (p2p) Internet overlay protocols

Platform-neutral computation description language

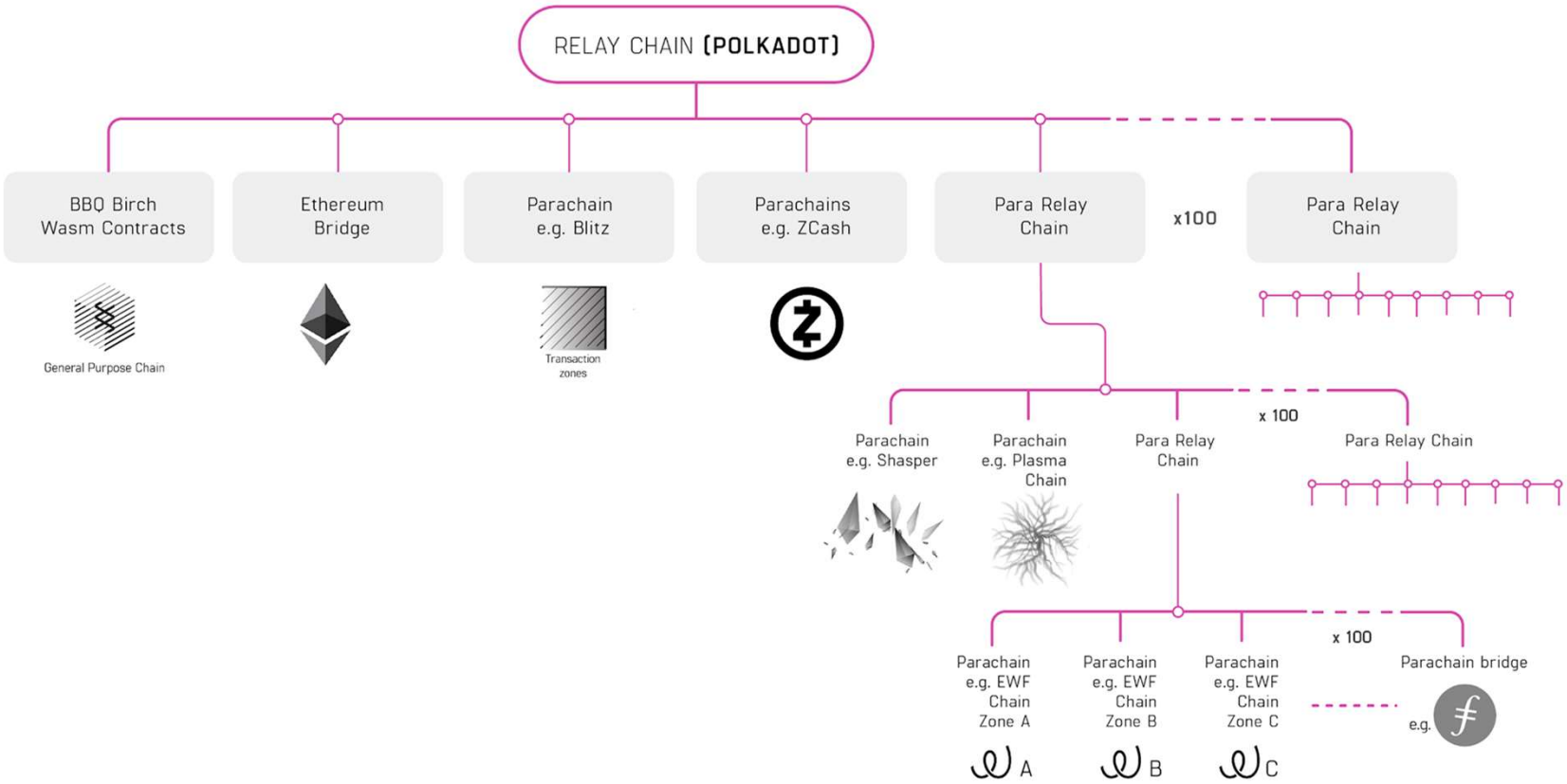
W3F

Introducing Polkadot



Polkadot.

Polkadot is designed to be composable!



STATUS QUO

Polkadot

ONE SIZE DOESN'T FIT ALL

Over the last years, we have seen an explosion in the number of different blockchain technologies.

Different projects need different designs, especially different state machine designs.

Polkadot.

ENTERPRISE NEEDS AREN'T MET

Public blockchains don't meet the permissioning and confidentiality requirements of businesses.

This results in businesses turning to isolated private chains.

Polkadot.

FACING A FRAGMENTED LANDSCAPE

Current generation blockchains cannot talk to each other without going through centralised services; which defeats the purpose of blockchains in the first place.

We end up with the similar data silos blockchain promised to break us out of.

Polkadot.

POLKADOT

Let’s Connect the DOTs

Polkadot

INTEROPERABILITY

- Connect chains with distinct state machines and consensus
- Support past, present, and future
- Public and private running **in the same network**

Polkadot.

THINKING BEYOND TOKENS

- Interoperability of tokens is clearly desirable
- Arbitrary message passing is a superset, and more valuable for innovation
- We want an interoperability framework that provides both

Polkadot.

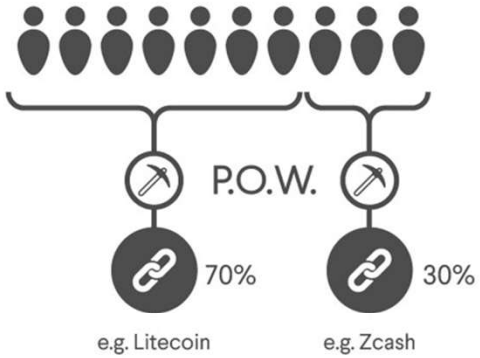
Polkadot.

Co-securing the Community

- Connecting blockchains is hard. Securing them is even harder.
- Blockchains naturally compete with each other over security resources.
- Polkadot lets chains pool security resources. Competition turns into rule based cooperation.

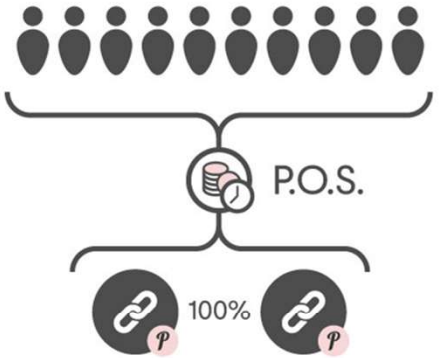
Polkadot.

Traditional isolated security



V.S.

Polkadot shared security



Polkadot.

Polkadot.

Scaling Blockchain Horizontally

- Multiple chains running in parallel
- Increasing throughput by parallelizing transactions

Polkadot.

POLKADOT

The Breakdown

Polkadot

Building Blocks: Technology

- **The Rust Programming Language**: safe, secure, efficient
- **WebAssembly**: (for parachains) portable, fast, and well-supported low-level target.
- **Libp2p**: cross-platform and flexible p2p protocol from Protocol Labs

Polkadot.

Design Principles

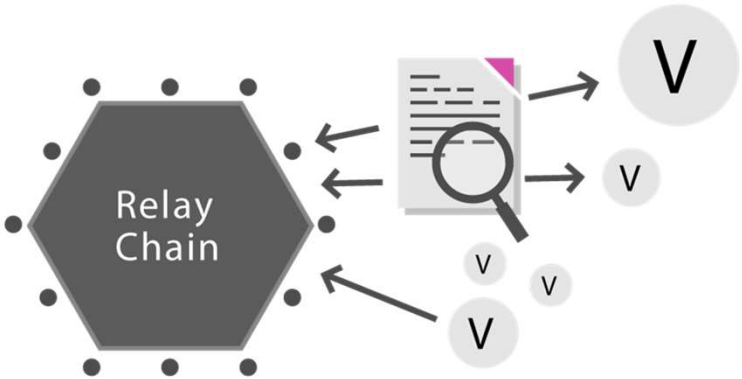
- **Heterogeneous:** support an ecosystem of diverse and complementary utilities
- **Scalable**: scalable on-chain and off-chain with an ultra-efficient root layer
- **Secure:** define rigorous and formal models of security of the system, protected with *economic games*

Polkadot.

Explaining Parachains

- **Validity Function:** WebAssembly stored on-chain in the parachain registry
- **Collator Node:** Creates “candidate” blocks that satisfy the validity function.
- **Message Queues:** candidates must also process incoming and produce outgoing messages

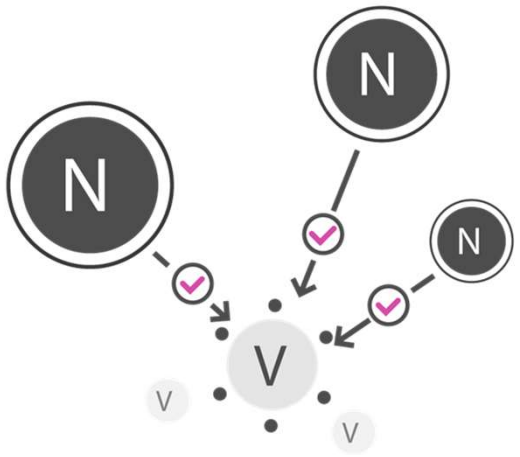
Polkadot.



Validators

- Manage relay-chain block authorship
- Parachain Candidate agreement
- Steward availability of external data

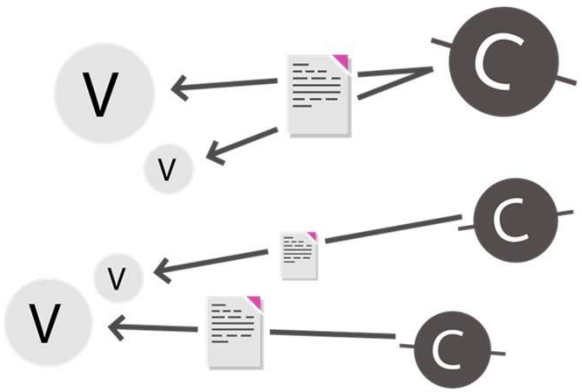
Polkadot.



Nominators

- Stake on behalf of good validators
- Economic security without additional consensus overhead
- Heuristic-based assignment

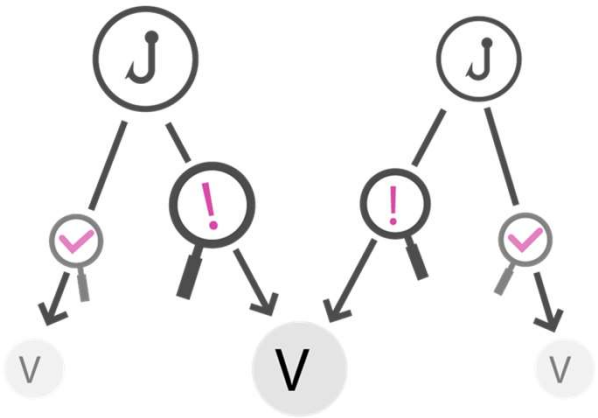
Polkadot.



Collators

- Create parachain candidates to give to validators
- Work on single parachain only
- Monitor parachain sub-net for misbehavior

Polkadot.



Fishermen

- Final line of defense: watch for misbehavior of validators
- Anyone can be a fisherman
- Can trigger the “validity/availability” game and slash bad validators

Polkadot.

Where are we now?

- **PoC-1:** Governance, staking, basic UI, and forkless upgrades
- **PoC-2:** “co-finalization” of non-communicating parachains and basic light client
- **PoC-3:** Implementation of hybrid consensus described in these slides.

Polkadot.

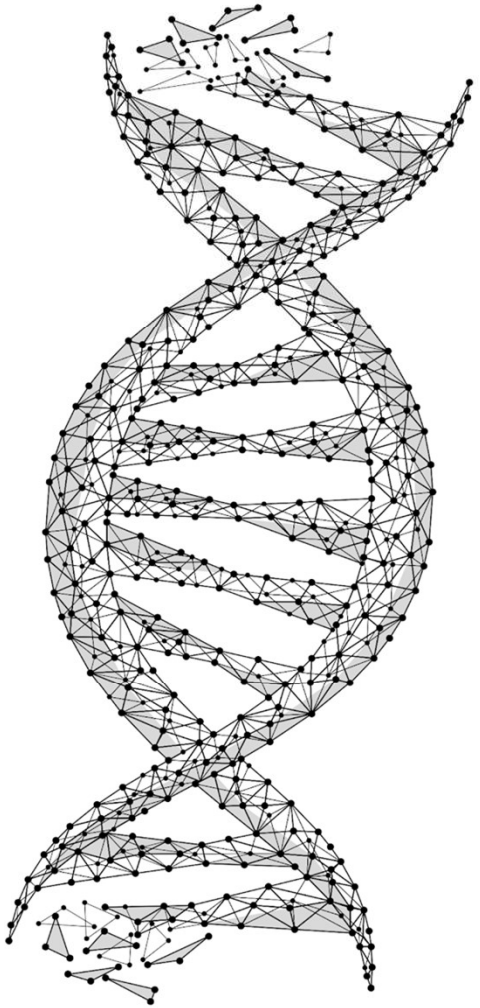
What's Next?

- **PoC-4:** Interchain message passing, Substrate chains can become parachains
- **PoC-5-7:** Implementation of Validity/Availability game
- **In parallel:** developer tools for parachains

Polkadot.

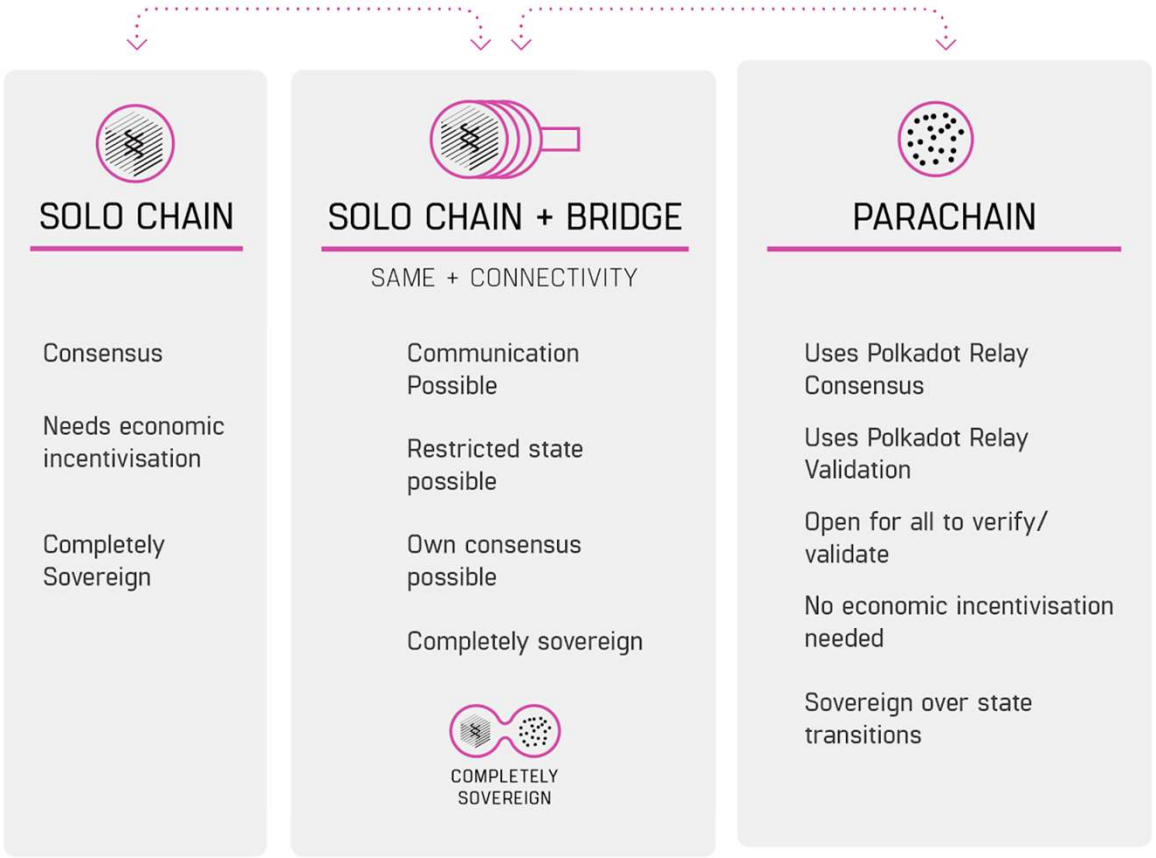
What do I get with Substrate?

- Hot-swappable, pluggable consensus
- Hot-upgradeable, pluggable STF
- Light client
- Chain synchronisation
- Pub/Sub WebSocket JSON-RPC
- Transaction queue
- Pervasive, secure networking
- JS implementation
- Modular SRML if you want
- Interchain connectivity via Polkadot



THE STATE OF CHAINS

Substrate



Polkadot.

Growing Polkadot Ecosystem

Teams building Parachains on Polkadot