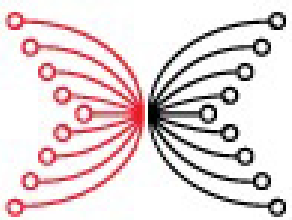
Insights into Cardano Development: Past, Present and Future

Duncan Coutts, Philipp Kant, Neil Davies, Darko Mijić

April 2018, London



INPUT | OUTPUT

The past

Brief history

Timeline

2016	September	first git commit
2017	April	last beta testnet
2017	August	release candidate testnet
2017	September	mainnet release
2018	March	start of regular rolling releases

Brief history

Exchanges

- started with one exchange
- added a second exchange a couple months later
- currently a dozen exchanges with significant ADA volume
- new wallet API in 1.2 is better designed for exchanges
- expect more exchanges to integrate our wallet

The present

Lessons

Lots of things work well!

- core system stability has been excellent
- system monitoring gives great insights

But...

Lessons

- performance requirements need to be clearly understood
- performance engineering needs to be done much earlier
- distributed concurrency and networking are hard

Corollary: hard problems need more formal approaches

Lessons

Example: the wallet backend

- proved not to be up to the task for use by exchanges
- rewriting the wallet backend's data layer from scratch
- now have a 30 page semi-formal specification

Robustness & Resilience in Cardano SL

- robust
- designed to survive
- risk/Hazards assessment + mitigation plans
- performant (benchmark testing)
- mean “time to chain” of approx 11s under load
- multiple outputs per transaction $\rightarrow$ high “user transaction throughput”
- reliable
- running 24/7 since launch
- 95% transactions in “next block”

Semi-formal software development

Example: the wallet backend

- a precise specification
- mathematical notation and style
- forces one to think clearly and simplify
- highlights tricky issues
- lemmas! but otherwise semi-formal
  - don't prove everything
  - test the implementation matches the specification

Leads to dramatically simpler and more robust implementations

The future

Smart contract platform(s)

Two approaches

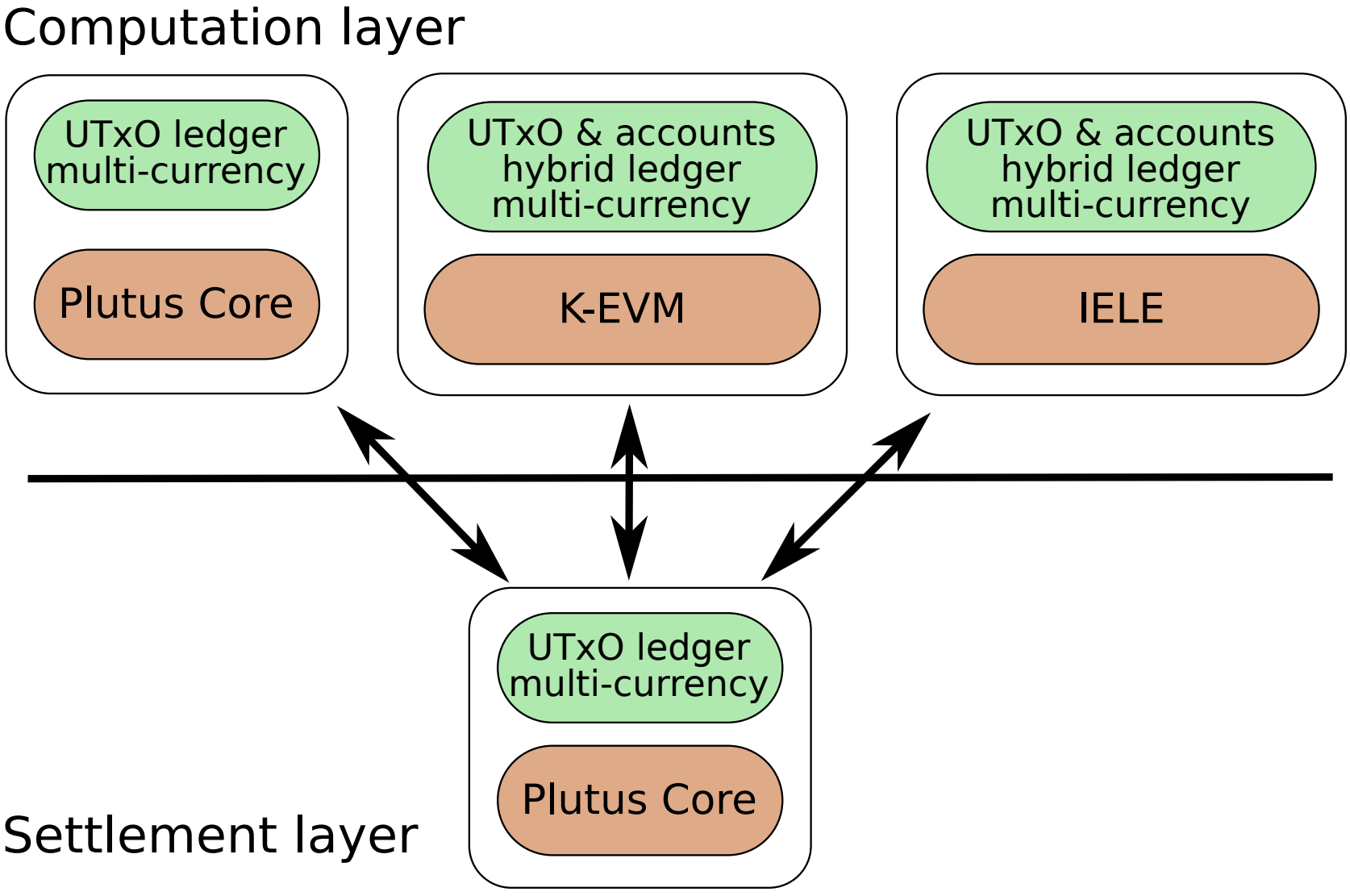
- covering legacy and traditional: K-EVM & IELE
- more radical: Plutus core, Plutus and Marlowe

Fits with the SL / CL split

- SL stays simple, reliable
- CLs host the more complex program execution environments

Also considering client side code.

Smart contract platform(s)



Things in the pipeline

Major features

- full decentralisation with stake pools
- hardware, light and mobile wallets
- SL / CL split with sidechains
- multi-currency ledger
- smart contracts with IELE, Plutus and Marlowe

Nice tech

- new higher-assurance Ouroboros Praos implementation
- comprehensive new networking layer based on RINA