First Principles of Blockchains

Mike Neuder & Matt Weinberg, Princeton University

Session VI, Metrics in Motion, World Bank

Thursday, November 20th, 2025

Theme for this talk

- **Blockchains are confusing!**
  - ✓ *Goal: clearly present some key ideas.*
- **Blockchains combine several technologies.**
  - ✓ *Goal: understand some technologies, how they synergize, which you need.*
- **Blockchains "have" several properties.**
  - ✓ *Goal: understand what some of these words really mean.*
- **Blockchains are mathy and technically complex.**
  - ✓ *Goal: no math – focus on core principles. But we will definitely make you think!*
- **Please interrupt as needed with any questions!!**

Outline

- **Blockchains combine several technologies. Will overview some.**
  - Ledgers.
  - Digital Signatures.
  - Automation.
  - Zero-Knowledge Proofs.
- **Blockchains “have” several properties. Will overview some.**
  - Permissionless vs. permissioned.
  - Centralized vs. “decentralized”.
- **Again, please interrupt as needed with any questions!!**

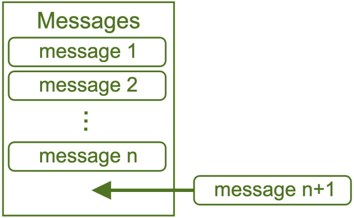
Some Examples of Technologies in Blockchains

Example 1: Ledger

A **ledger**:

- stores **valid** **messages**,
- in a single consistent **order**,
- and never deletes.

That's it!



Example 1: Ledger

A **ledger**: stores **valid** **messages**, in a single consistent **order**, and never deletes.

Things we will come back to in fifteen minutes:

- Who decides what is **valid**?
- Who decides what the **messages** mean?
- Who decides the **order**?

Example 1: Ledger

A **ledger**: stores **valid** **messages**, in a single consistent **order**, and never deletes.

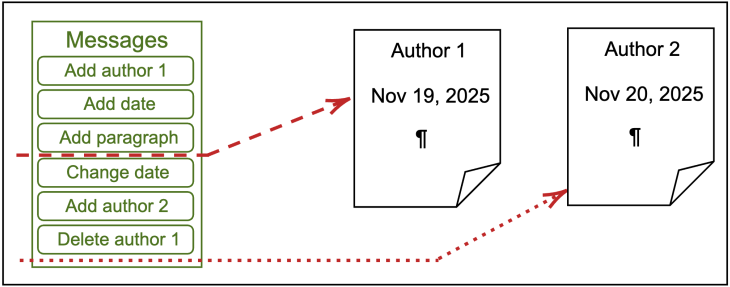
What can you do with *just* a ledger? **Example:** Google Docs.

- **Valid**: from a user with access, performs a valid text operation (add, erase, etc.).
- **Messages**: mean “please update the doc according to the operation.”
- **Order**: in whatever order Google servers receive them.

Example 1: Ledger

A **ledger**: stores **valid** **messages**, in a single consistent **order**, and never deletes.

What can you do with *just* a ledger? **Example:** Google Docs.



Example 2: Digital Signatures

- Imagine a magic pen with a **truly personalized color**.
- Only this one magic pen can write in this **truly personalized color**.
    - Called the "private key."
- Anyone can recognize the ink's **color** and which magic pen it is from.
    - Called the "public key."
- Then, you could write messages with this pen, and everyone would know that you must have written it (or that someone stole your pen).
    - Called "verification," using the public key.
- Cryptography: Digital Signatures use fancy math to accomplish this digitally.

Example 2: Digital Signatures

Digital Signatures: write messages, everyone knows **you** wrote it.

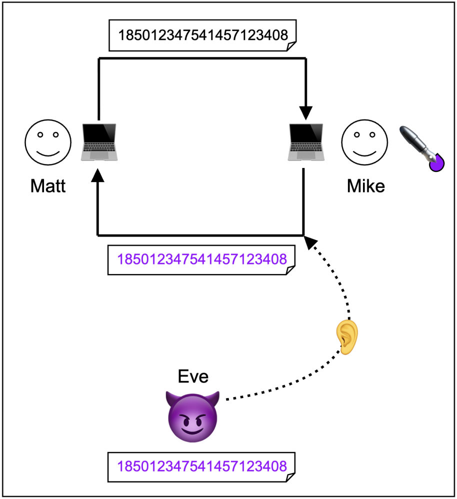
What can you do with just Digital Signatures? **Example**: authentication/HTTPS.

Magic pen analogy:

- Matt has never met Mike, but Matt knows Mike's magic pen **color**.
- Matt and Mike meet up, Matt wants Mike to prove it is him.
- Matt tells Mike: "I will speak to you only if you bring a paper with 185012347541457123408 (a random number) in magic pen."
- Matt feels safe trusting – how could a phony get the random number in Mike's magic **color**?
- Mike feels safe giving this to Matt – can only impersonate Mike on this one particular message.

Example 2: Digital Signatures

Digital Signatures: write messages, everyone knows **you** wrote it.



Example 2: Digital Signatures

Digital Signatures: write messages, everyone knows **you** wrote it.

What can you do with *just* Digital Signatures? **Example**: authentication/HTTPS.

- Matt knows **World Bank's public key**.
- Matt is talking to someone on the Internet, wants it to be **World Bank**.
- Matt tells the contact: "I will speak to you only if you **digitally sign** 185012347541457123408 (a random number) with your **private key**."
- Matt feels safe trusting – how could a phony get access to **World Bank's private key**?
- World Bank feels safe giving this to Matt – can only impersonate **World Bank** on this one particular message.

Example 1+2: Ledger + Digital Signatures = "payments"

- Everyone initially gets 100 "tokens", attached to a public/private key.
- A message is **valid** according to the ledger if it says "A pays *n* tokens to B", and:
  - it is digitally signed by A, and
  - considering prior transactions **in the ledger**, A received *n* tokens more than paid (including the initial 100).

Mike offers Matt: "I will sign a message 'Mike pays 3 tokens to Matt' if you give coffee."

- Matt might think: **sure!** Now I can buy that sandwich from JP I have always wanted, by offering "I will sign a message 'Matt pays 3 tokens to JP' for a sandwich."
- (Matt might also think: **no!** All my friends think tokens are dumb and they will not give me sandwiches in exchange for signed messages – can do as he pleases.)

Example 1+2: Ledger w/o DigSigns = “payments?”

- Everyone initially gets 100 “tokens”, ~~attached to a public/private key~~.
- A message is **valid** according to the ledger if it says “A pays *n* tokens to B”, and:
    - ~~it is digitally signed by A, and~~
    - considering prior transactions **in the ledger**, A received *n* tokens more than paid (including the initial 100).

Mike offers Matt: “I will ~~sign a~~ message ‘Mike pays 3 tokens to Matt’ if you give coffee.”

- Matt would think: lol, why give you coffee, I can write the message myself.
- **Digital Signatures ⇒ Matt cannot write the message himself.**

Example 1+2: DigSigs without Ledger = "payments?"

- Everyone initially gets 100 "tokens", attached to a public/private key.
- A message is valid to *you* if it says "A pays n tokens to B", and:
  - it is digitally signed by A, and
  - considering prior transactions **you know**, A received n tokens more than paid (including the initial 100).

Mike offers Matt: "I will sign a message 'Mike pays 3 tokens to Matt' if you give coffee."

- Matt would think: lol, I do not need your tokens, I will just spend my 100 repeatedly.
- Sign to JP: "Matt pays 100 tokens to JP", get 33 sandwiches.
- Sign to Andrés: "Matt pays 100 tokens to Andrés", get 33 coffees.
- **Ledger ⇒ single consistent definition of "valid".**

Example 1+2: Ledger + Digital Signatures = "payments"

Really cool! "Payments" form the basis of cryptocurrency.

But, still important to remember:

- It is all just messages! There is no "real" token being moved around.
- If someone steals your private key, they control your tokens.
- "One BTC is worth \$93k" means someone will give you \$93k USD to digitally sign a message 'you pay one BTC to [someone]'.

Example 3: Automation

Automation: write new messages *automatically*, **when some conditions are met**.

What can you do with just Automation? **Example**: tax filing.

- I manage all my investments in a single place.
- On Tax Day, I have programmed my account to automatically calculate the amount of capital gains/losses for the year and produce a 1099.
  - Plus a message to my bank to please transfer the funds to IRS, if desired.

Example 1+2+3: “Smart Contracts”

Automation: write new messages *automatically*, **when some conditions are met**.

Cryptocurrency + Automation = “Smart Contracts”.
Automated tax form preparation.

- I manage all my investments on a **ledger**.
- On Tax Day, I have programmed my account to automatically calculate the amount of capital gains/losses for the year and **write a message on the ledger**: “Matt pays [appropriate capital gains/losses tax] tokens to IRS.”

Takeaway:

- Automation automatically writes messages.
- **When cool stuff is just messages**, Automation automatically **does** cool stuff (i.e., makes “payments”).

Example 4 (fancy!): Zero-Knowledge Proofs (ZKPs)

Zero-Knowledge Proofs: share *exactly* the information you want to share.

- India wants to inform the World Bank: "I promise that the following are true":
  - I digitally signed 1000 messages of the form "A pays 100 tokens to B",
  - but I do not want to tell you which specific participants I paid.
- Imagine a trusted auditor that is 100% honest with access to all data.
  - India shows the **auditor** the claimed digitally signed messages, asks the auditor to tell the World Bank.
  - **Auditor** tells WB: "yes, what India said is true."
  - Good news: WB learns exactly what India wants to share.
  - Bad news: WB trusts the **auditor** to be honest and India trusts the **auditor** with private info.

Example 4 (fancy!): Zero-Knowledge Proofs (ZKPs)

Zero-Knowledge Proofs: share *exactly* the information you want to share.

- India wants to inform the World Bank: “I promise that the following are true”:
    - I digitally signed 1000 messages of the form “A pays 100 tokens to B”,
    - but I do not want to tell you which specific participants I paid.
- **ZKP**: this is possible without the trusted auditor!
    - India shows WB a **ZKP**.
    - WB reads the **ZKP**, learns exactly what India wants to share.
    - With only the **ZKP**, WB cannot learn India’s private information!
    - This should feel like magic! It is really fancy math, and truly remarkable (but it works!).

Example 1+2+4: Ledger + Digital Signatures + ZKPs = private "payments"

Cryptocurrency + ZKPs for "payments":

- Everyone initially gets 100 "tokens", attached to a public/private key.
- A message is **valid** according to the ledger if it has the following components:
  - an encryption of a message, and
  - a **ZKP** that the message is of the form "A pays *n* tokens to B", plus a digital signature of A, such that:
    - considering prior transactions in the ledger, A received *n* tokens more than paid.
    - And no specifics of A, B, or *n* are revealed.

Has all the properties (and caveats) of cryptocurrency, and:

- **ZKP** guarantees included messages are **valid** according to the ledger.
- **ZKP** guarantees **no one learns who pays whom** how much!
- (Not offering an opinion on whether this is "good for society" – but it is cool!)

Key takeaways: technologies in blockchains

Blockchains combine several technologies. We saw:

- Ledgers: store **valid** **messages**, in a single consistent **order**, and never delete.
- Digital Signatures: write messages, everyone knows you must have **written** it.
- Automation: write new messages automatically, when conditions are met.
- **Zero-Knowledge Proofs**: share exactly the information you want to share.
- Ledgers + Digital Signatures = cryptocurrency.
- Cryptocurrency + Automation = smart contracts.
- Cryptocurrency + ZKPs = private cryptocurrencies.
- ...and many more! (Many more individual technologies, and many more cool combos.)

Key takeaway: good to think hard about which **properties you want**, and how to **combine them**.

Some Properties of Blockchains

Rest of this talk: key properties

Warning: there is a lot of nuance here!

The rest of this talk will focus on **ledgers**, without worrying about:

- Whether the messages are digitally signed.
- Whether the messages are payments or code or something else.
- Whether the messages are private behind ZKPs.

Note: The key properties **absolutely make sense to talk about outside of ledgers**.

- See Andrés' talk for an example!

Rest of this talk: key properties

Warning: there is a lot of nuance here!

For a **ledger**, it is vital to know:

- Who decides what is **valid**?
- Who decides what the **messages** mean for the real world?
- Who is allowed to write/read **messages**?
- Who decides the **order**?

Different ledgers make different decisions.

- When discussing non-ledgers, the questions are different, but similar themes appear.

Property 1: permissioned vs. permissionless

Warning: there is a lot of nuance here!

- Permissioned: **some authority** dictates who is allowed to perform [activity].
- Permissionless: **anyone** can perform [activity].
- Nuanced: anyone technically *can* perform [activity], but there is a high barrier to entry.

Property 2: centralized vs. decentralized

Warning: there is a lot of nuance here! (These particularly lack rigorous definitions.)

- Centralized: a **single party** (or few parties) do [activity].
- Decentralized: an **open protocol** with open participation determines [activity].
- Nuanced: participation is technically open, but in practice dominated by few parties.

Example 1: Bitcoin / Ethereum

Warning: there is a lot of nuance here!

- Who is allowed to write/read **messages**? Anyone!
  - Truly permissionless.
- Who decides the **order**? Miners/stakers participate in a consensus protocol.
  - Decentralized: anyone can buy hardware (BTC) or lock up capital (ETH).
  - **This is the key cool feature –** no single entity decides the **ordering**.
  - Nuance: it is expensive – in practice, mining/staking is dominated by few big players.

Example 2: Stablecoins (USDC, USDT)

Warning: there is a lot of nuance here!

- Who is allowed to write/read **messages**? Anyone!
  - Still truly permissionless.
- Who decides what is **valid**? Who decides what the **messages** mean?
  - Circle / Tether.
  - Permissioned / centralized – they have a lot of power here!

Example 3: Tempo (Stripe)

Warning: there is a lot of nuance here!

- Who is allowed to write/read **messages**? Anyone! (for now.)
  - Permissionless (for now).
- Who decides what is **valid**? Who decides the **order**?
  - Stripe and corporate partners.
  - Permissioned and centralized.

Example 4: Permissioned blockchain (between banks)

Warning: there is a lot of nuance here!

- Who decides what is **valid**? Who decides what the **messages** mean? Who is allowed to write/read **messages**? Who decides the **order**?
  - The banks.
  - All forms of participation are fully permissioned.
  - Generally, this is intended!
  - **Very** different than permissionless blockchains – **very** different applications.

Key takeaways: permissionless vs. permissioned, centralized vs. decentralized

Warning: there is a lot of nuance here!

- These are key discussion points when evaluating blockchains.
- Permissionless/decentralized blockchains are *wildly different* than permissioned/centralized ones.
- Make sure you know what you want!
- There is nothing "right" or "wrong" about one vs. the other.
- These questions also make sense to discuss for similar technologies.
  - See Andrés' talk next!
  - Precise questions will differ.

Case study: property rights

When you “own” USDT on Coinbase, what do you “own”?

- Tether owns lots of Treasury bills and dollars; “issues” USDT on Ethereum.
- Account owners can permissionlessly transfer USDT on Ethereum (no KYC).
- Coinbase controls an account that controls USDT.
- You have a user agreement with Coinbase, who serves as a custodian. The US legal system enforces it.
- Trusting Coinbase/US law, anyone can transfer USDT.
- If you want to actually exchange your USDT for USD:
    - Need to investigate your user agreement with Coinbase (permissioned/centralized).
    - Also need to investigate Tether’s policy on redemptions (permissioned/centralized).

Wrapping up

- **Blockchains are confusing!**
  - Hope you learned *some* basics. There is a lot more basics out there!
- **Blockchains combine several technologies.**
  - Hope you learned *some* examples, how to think about what you need, and how to creatively combine them.
- **Blockchains “have” several properties.**
  - Hope you learned that *nuance* matters, and how to think about permissionless vs. permissioned, decentralized vs. centralized, and what your application needs.
- **Blockchains are mathy and technically complex.**
  - Hope you learned that it is possible to think deeply and understand what is *really* going on, even if you do not get into the mathy details.

Thanks!