Why I Won't Use the Mainnet

For Business

Ten reasons why a person responsible for the information security of a company wouldn't use the **Ethereum Mainnet**

And how to make it so they will

The Job of the EEA Enterprise Mainnet Working Group:

Show serious IT professionals how the Ethereum public Mainnet can be used in a safe and effective way.	Give the Ethereum core developer community specifics on what industry needs to adopt mainnet.

The Serious IT Professional

I'm in charge of our company's information security. We have 10,000 employees and maintain records on 5 million active customer accounts. We engage 250 vendors and partners.

We run hundreds of applications and deal with nearly as many vendors (plus a half-dozen VARs and solutions firms implementing or maintaining them). That's just the stuff I know about.

People approach me all the time, sometimes the CEO herself, and they ask me to consider using some new system, some new pattern. I usually smile and wave. They usually don't come back.

It's not that I'm a Luddite. I'm just busy.

I see what happens to the careers of people like me when the scandal hits, when PII is exposed.

I'm fine with the Cloud for some things. The CIA opting for AWS turned my head. And I have to admit, their security environment is better than I could ever replicate with our resources. So, more and more of our infrastructure is going there, though even then, we prefer full bare-metal...no multitenant systems for anything involving mission critical operations or sensitive data.

The Ethereum Mainnet

An always-on public utility, a state machine that resists tampering and censorship but sacrifices speed, scalability and fast finality in order to allow anyone to help maintain the integrity of the database.

Because the Mainnet is a permanent public record, encrypted information placed there can be observed by anyone at any time, forever. This includes parties with the means and know-how to perform advanced analytics and ascertain patterns that may reveal strategic intelligence even without having to defeat the encryption of the data itself.

Used correctly, the Mainnet can be employed by business to solve long-standing problems:

1. Automate B2B agreements without creating new silos (like private blockchains do)
2. Integrate new relationships with existing ones flexibly without losing system integrity or adding integrations on top of integrations
3. Enforce consistency between different parties' records without having to move the data or business logic from legacy systems
4. Enforce continuity in a multiparty workflow (e.g., an invoice always agrees with the purchase order) while compartmentalizing which parties know the details in each step

The Enterprise Ethereum Alliance Mainnet Working Group is compiling a list of the questions and concerns that IT professionals have regarding the Ethereum Mainnet. We will use these to communicate requirements, specifications and suggested patterns-of-use to the community.

Ten Issues With The Idea of Using the Mainnet In Business

The Data Locality Problem
GDPR requires that I can account for where PII data is stored, even when it is encrypted. And I need to be able to delete that data permanently upon request. (That's just the tip of the iceberg.) If the data is sitting permanently on any number of nodes not controlled by me everywhere...yeah.

The Strategy Leak Problem
Transaction metadata can be used to game the system or collect / analyze for strategic counter-intelligence or corp. espionage. In the age of AI, any trace activity done on a permanent, public ledger can be used to figure out who is doing what, even if it's just little changes to merkel tries.

The Finality Problem
Ethereum is an "eventual consistency". If that's changing with Eth2.0, I don't understand it...something about a magical fast finality something. I dunno. What I do know is that all my systems are ones where a change to data is final the second it's written.

The Speed and Latency Problems
Our CRM and ERP systems don't need the kind of Transaction Per Second speeds of a Visa or Mastercard (and even they get those TPS rates through parallelization...can't fool me). But long wait times for round-trip + consensus makes things I might do with mainnet a bad user experience.

The Cost Problem
It's not just about how much it costs per transaction or that our company isn't yet set up to hold $Ether for "gas." It's also about the unpredicatability of the cost. And I don't want to have to go back for my degree in finance to understand stablecoins just to manage IT.

The Private Data Problem
Eighty percent of our data is considered sensitive, internal or personally identifiable client, customer or user data. Encryption isn't enough. Any data can be deanonymized and decrypted given time. And anyone with a full node has forever to crunch the bits on the ledger.

The Scaling Problem
The often touted TPS problem is one thing. But the bigger problem for us is what happens when there are many enterprises hitting the mainnet simultaneously. Business isn't Cryptokitties. Need to enable massive numbers of concurrent sessions. Reads and writes by many parties.

The Confidential Code Problem
You can't just hide the data with something like zkSnarks and think that everything is ok from a corporate perspective. Many business agreements are embodied in code...business logic. If a machine can execute a smart contract, it can decompile and look at the logic, and that can leak sensitive info.

The Responsible Party Problem
My legal structure requires that there be a responsible party handling all aspects of my data and business logic. If I put data on the Mainnet, I lose a key responsible party.

The Anonymous Party Problem
Everything I do involves access control, down to the user. I can't work with a system that doesn't allow me to control who has access to a function or a piece of data..not just who can decrypt it but who has physical access to the bits.