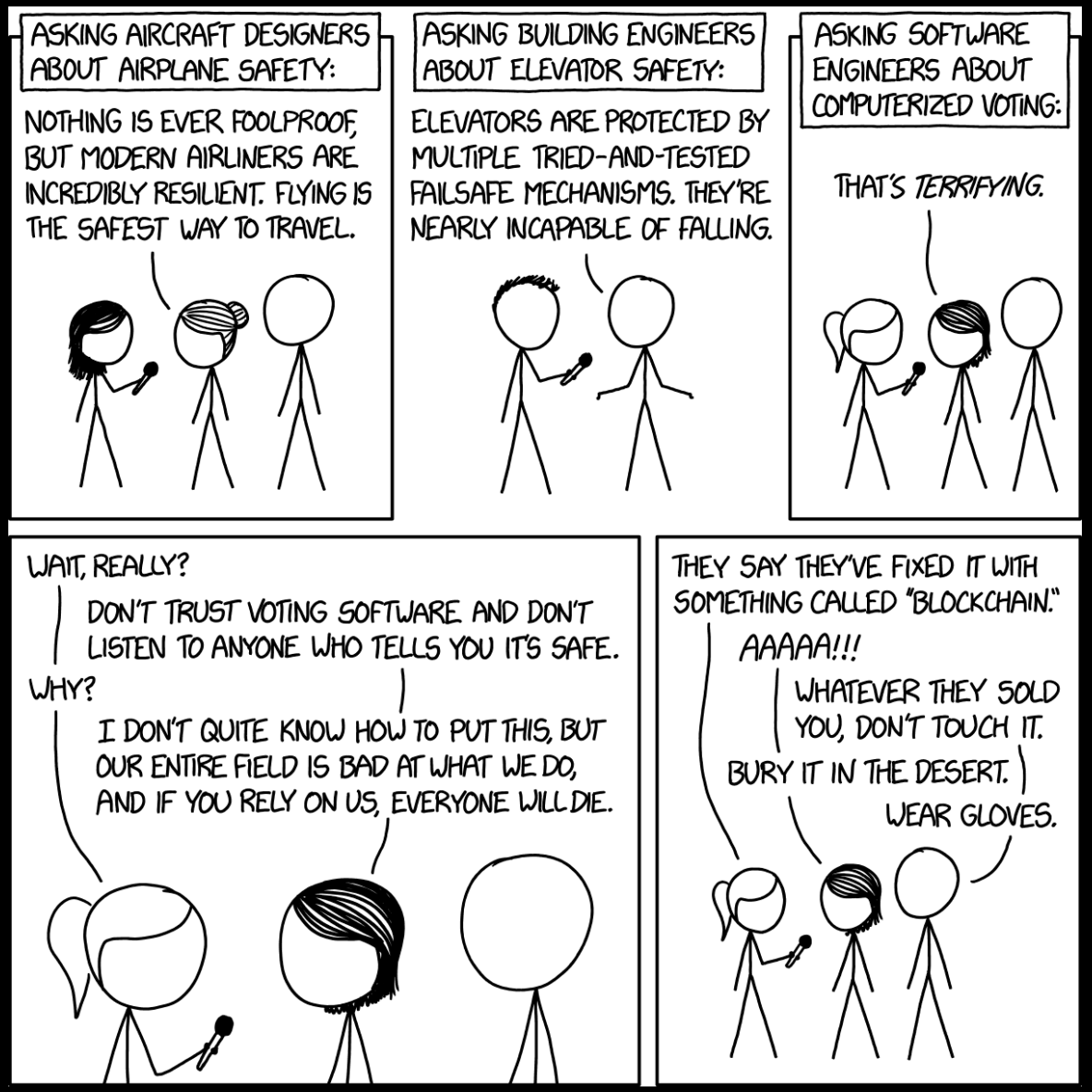
"Voting Software"

xkcd.com/2030/



CS 168: Blockchain and Cryptocurrencies

Challenges of Decentralized Systems

Prof. Tom Austin
San José State University

HW1: DigiCash Lite (DCL)

So why did DigiCash fail?

- Poor business decisions?
- Financial institutions not ready for cryptocurrencies?
- Governments worried about money laundering?

Alternate approach: everyone tracks all transactions.

Bitcoin (BTC)

- Protocol designed by Satoshi Nakamoto in 2008 https://bitcoin.org/bitcoin.pdf
- First Bitcoin client launched in 2009
- Peer-to-peer – no centralized control
  - Every client keeps track of the history of all bitcoins

Bitcoin Terminology

- **Bitcoin** (w/ capital B) – the protocol
- **bitcoin** (w/ lowercase b) – the coins
- **Miners** – validate transactions for bitcoin rewards
- **Blockchain** – distributed ledger
  - Organized in blocks of transactions
  - Blocks "chained" together w/ cryptographic hashes
- **Genesis block** – 1st block of BTC transactions

Building a Cryptocurrency

What is a cryptocoin worth?

Some cryptocurrencies tie there value to another currency.

Other cryptocurrencies (such as Bitcoin) are not tied to any other currency. We'll follow this model.

Digital Currency – Ledger

Alice: 20
Bob: 11
Charlie: 5
David: 34

Alice

"I am giving 10 cryptocoins to Bob"

Alice

Bob

Alice: 20
Bob: 11
Charlie: 5
David: 34

David

Alice: 20
Bob: 11
Charlie: 5
David: 34

Charlie

Alice: 20
Bob: 11
Charlie: 5
David: 34

Digital Currency – Ledger

Alice

"I am giving 15 cryptocoins to Charlie"

Alice

Bob

David

Charlie

Digital Currency – Ledger

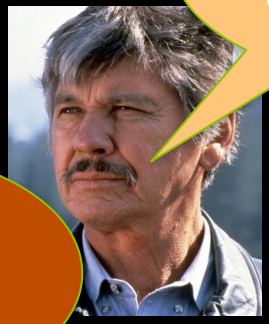
Alice

"I am giving 8 cryptocoins to David"

Alice

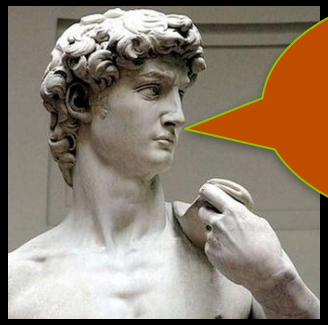
Bob

Invalid transaction!



Charlie

Invalid transaction!



David

Invalid transaction!

Lab, part 1:
Implement a Decentralized Ledger

Details in Canvas and on course website.

Lab, part 1: Implement distributed ledger

Each user keeps a local record of all balances.

A **`punishCheater`** method is called if a user attempts to spend money they do not have.

Messages are sent in JSON format. The fields in the JSON are up to you.

Decentralized Protocol Problem

Centralized bank:
Bank's view is "truth"

With decentralized protocols, how do we ensure everyone agrees?

Goals of a Distributed Protocol

- **Consistency**
  - Every read receives most recent write (or an error).
- **Availability**
  - Every request receives a (possibly stale) response.
- **Partition tolerance**
  - System continues to operate despite messages being dropped/delayed.

Unfortunately, we can't have all three.

(At least, not all of the time).

CAP theorem

- Also known as *Brewer's theorem*.
- Proves we can't guarantee consistency, availability, and partition tolerance.
  - We can get all 3 *most of the time*.
- When there is an error, which do we choose?

Which do protocols forfeit?

All have their place.

- Availability + consistency
  - Single-site databases
  - (Not an option for distributed systems)
- Partition tolerance + consistency
  - Distributed databases
  - Majority protocols
- Partition tolerance + availability
  - DNS

Bitcoin

- Partition tolerance
  - Yes – pretty much essential for dist. protocols
- Availability
  - Yes
  - Extremely resistant to censorship
- Consistency
  - "Eventually consistent"
    - Fancy term for "not consistent"
  - Transactions may be dropped
  - But… pretty good after a while

Lab, part 2: Break consistency of distributed ledger

Eject a user by framing them as a cheater.

Send different messages to different clients to confuse them.

How could we defend against this attack?