

Blockchains & Cryptocurrencies

Introduction



Instructor: Matthew Green
Johns Hopkins University - Spring 2026

My background

- Mainly work in applied cryptography (TLS, messaging systems, privacy-preserving protocols, cryptocurrencies)
- I co-founded a private cryptocurrency (**Zcash**) and boy was that weird
- Also helped start some companies:
 - **Aleo** (ZK smart contracts)
 - **Sealance** (regulatory compliance)
 - **Bolt** (custody)



What is a blockchain?

- A data structure
- A specific type of decentralized ledger (DLT) or “consensus network”
- Used for building decentralized cryptocurrencies such as Bitcoin, Ethereum
- Many other applications: distributed Domain Name system (DNS), supply chain, Public-Key Infrastructure (PKI), decentralized finance, etc.



Course objectives

- Understanding the mechanics of blockchains and consensus networks
- Understanding the necessary cryptographic background
- Exploring applications of blockchains to cryptocurrencies and smart contracts
- Learning to use some common DLT technologies
- Understanding limitations of current blockchains/DLT tech

Course objectives (contd)

- Introduction to recent exciting research
- **Main course goal:**
Extend this research
- Entrepreneurial or research projects by student teams
- These can be major projects, or even small applications

Disclaimer

This is not a finance course on cryptocurrencies. You should not expect to be taught how to invest in cryptocurrencies or how to become a millionaire overnight.

Disclaimer

This is not a finance course on cryptocurrencies. You should not expect to be taught how to invest in cryptocurrencies or how to become a millionaire overnight.

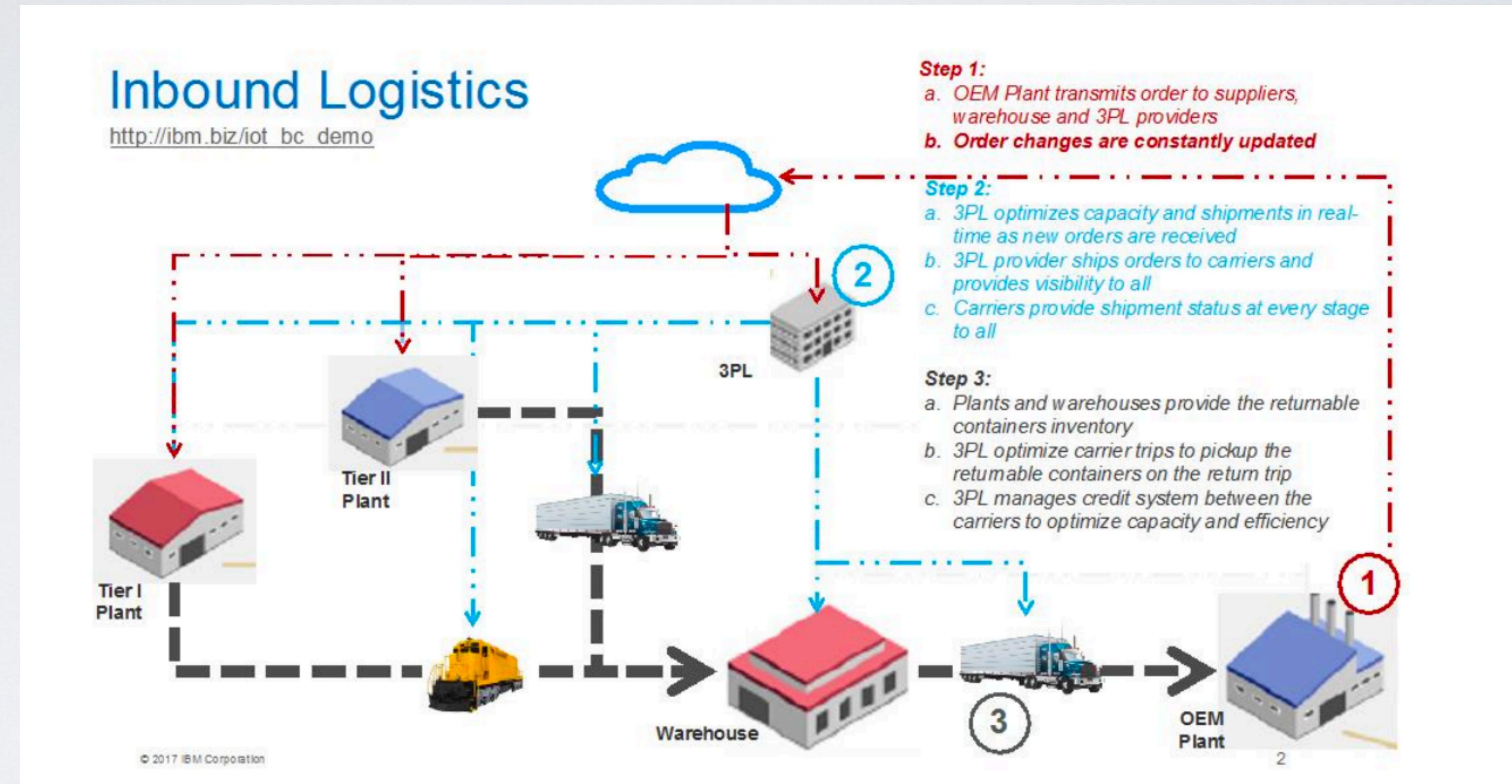
(Unless you're already a billionaire — then we can certainly help you become a millionaire.)



Pre-requisites

- No background in cryptography is necessary. However, the following are expected:
 - Basic mathematical maturity
 - Comfort with basic probability
 - Familiarity with asymptotic (Big-O) notation
 - Programming capability (in Python/Java, etc.), willingness to learn a new, weird language and also Git

Boring course logistics



Resources

- **Course website & syllabus:**

<https://github.com/matthewdgreen/blockchains2026>

(Or visit Piazza)

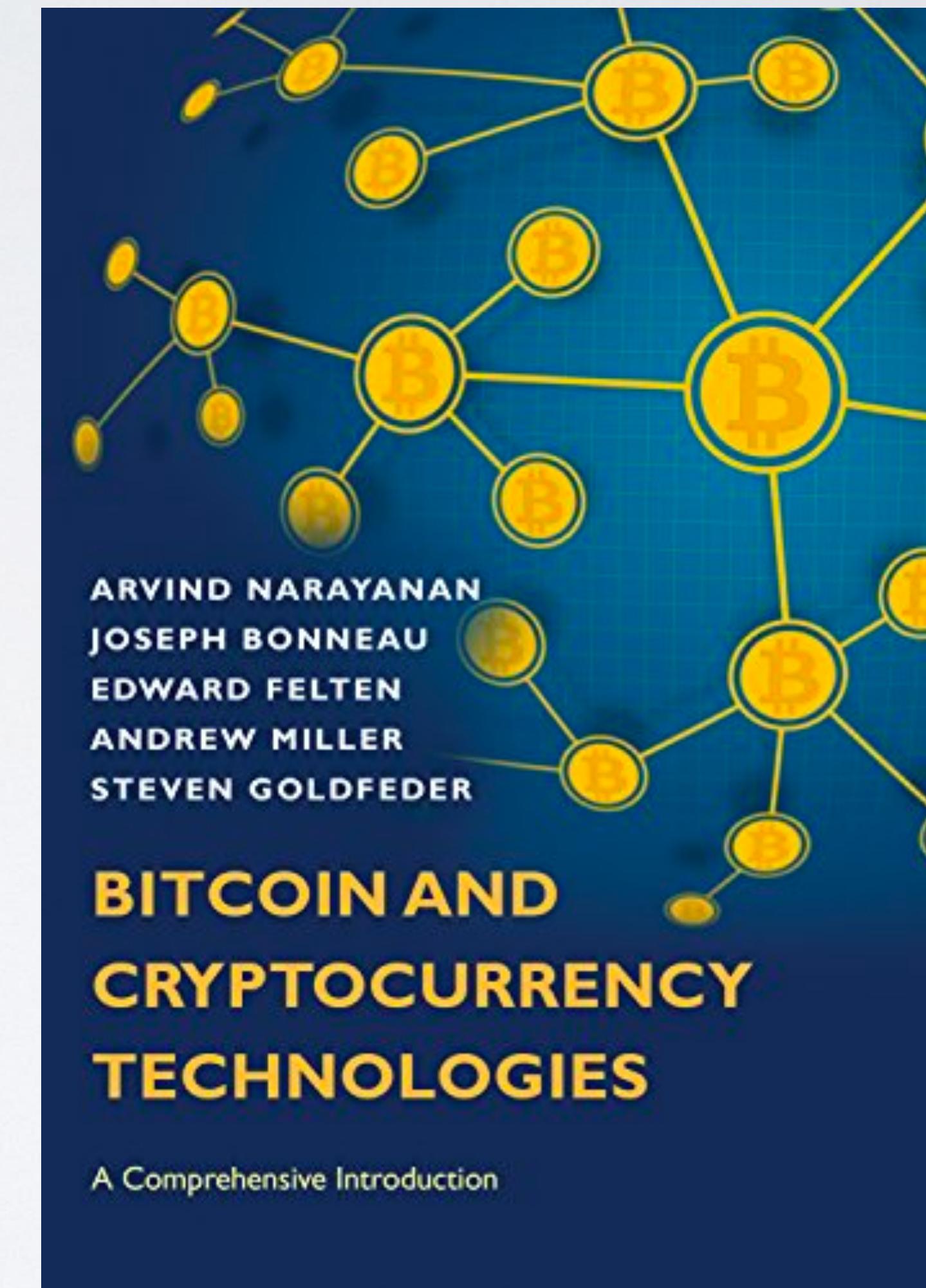
- **Piazza:** <https://piazza.com/class/mkne53lt5dm482>

Search for JHU 601.441/641 2026, or course title

- **Recordings:** I put them up eventually! Usually!

Texts/Readings

- **Text:** NBFMG (right)
- However, many readings will be drawn from online papers and resources
- Readings on syllabus are assigned the day they are listed, will be discussed during the following lecture



Texts/Readings

Course Syllabus

matthewdgreen edited this page 1 hour ago · [2 revisions](#)

Dates and topics are subject to change. Reading is assigned on the day specified, and will be discussed in the following class.

1/21/2026: Introduction

- Reading: NBFMG Chapter 0

1/26/2026: Crypto Background

- Reading: NBFMG Chapter 1
- Reading: [Shafi+Mihir's Notes \(Chapter 8 and 10\)](#)

1/28/2026: Towards Bitcoin

- Reading: [Merkle-Damgard Transformation](#)

TA & Office Hours

- **TAs/CAs:**

- Dorian Liu
- *Office hours TBA*



Grading & Exams

- **Grading:**

60% assignments+exams,
40% project

- **Exams:**

Midterm exam: (sometime in March)
Final exam: as assigned by JHU

- **Assignments:**

Programming & written, submit
via Gradescope.

Course project

- This is a research-quality project, conducted by groups of 1-3 students. You must have instructor approval for your project topic.
- There will be a list of project ideas on the course website. Other ideas are also fine with approval.
- Deliverables: new software, high-quality written report, detailed presentation (choose 2)
- **I-page proposal due 9/30**

Honor Code

- Except where explicitly marked, assignments and exams are individual work. You're expected to do your own work on these. Don't give or receive exam-specific assistance on these.
- This particularly includes code you find on the Internet
- We disallow use of LLM type tools (e.g., ChatGPT)
for programming assignments unless specified
- See the JHU academic integrity code.

Honor Code ++, Cryptocurrency edition

- Many legal aspects are unsettled in the blockchain/cryptocurrency space
- E.g., what happens if you discover and exploit a vulnerability in an experimental blockchain project?
- In this course we practice responsible disclosure. If this comes up e.g., in your course project, **see the TA or instructors.**

News

- Read the news!!
- CoinDesk, CoinTelegraph, Hacker News, etc.
- X/Twitter (ugh): maybe @VitalikButerin, @pwuille, @IOHK_Charles, @ethereumJoseph, @starkness, @adam3us, etc.
- Lots of network-specific forum sites, Discords, etc.

Any other questions?

- Come and ask after class, or send a note via Piazza

Towards blockchains



Cryptocurrencies Aren't 'Crypto'

As the price of Bitcoin and Ethereum skyrocket, and more and more people who are unfamiliar with technology join in the craze, words start to lose their original and correct meaning.

SHARE



TWEET



Market Summary > Bitcoin

89,918.44 USD

+89,464.45 (19,706.26%) ↑ all time

Jan 21, 7:38 PM UTC · [Disclaimer](#)

1D

5D

1M

6M

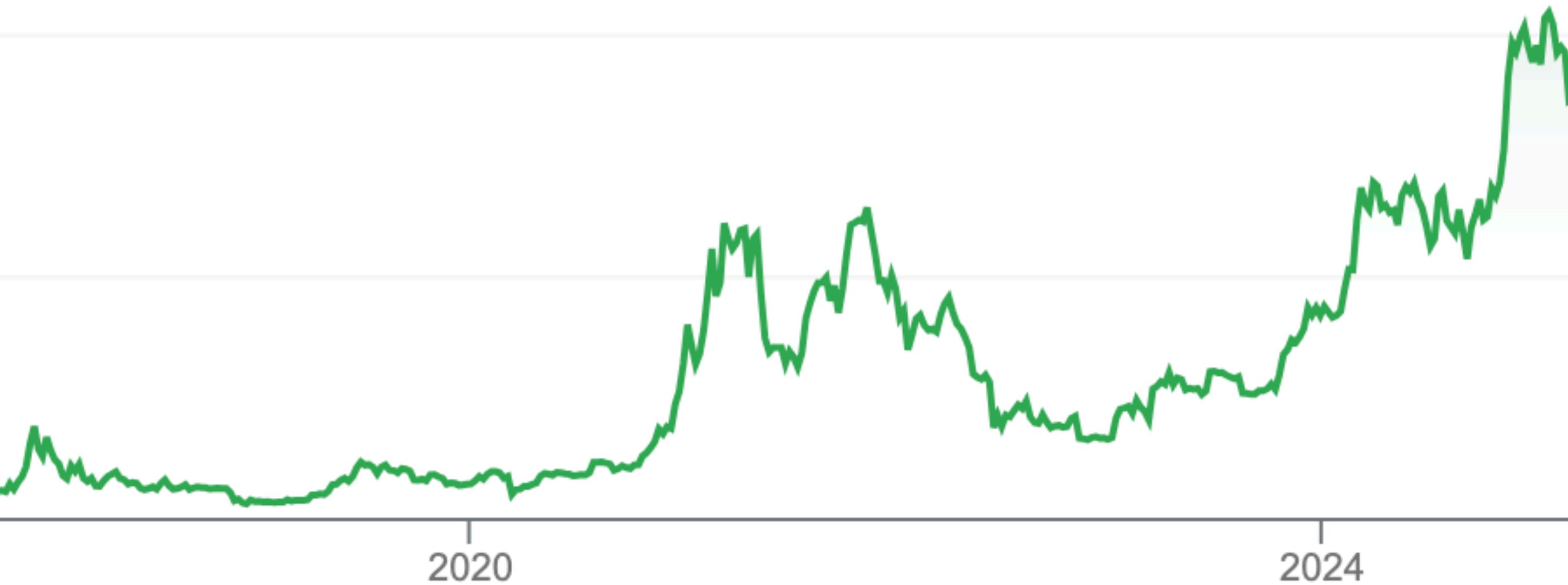
YTD

1Y

5Y

Max

150,000



- **Whether you love it or hate crypto...**

- Cryptocurrencies have exerted a massive influence on society and the field of CS
- Most people's first major exposure to cryptography and distributed systems
- That's both a good thing and a bad thing
 - The good: we get to deploy some really exciting new cryptography
 - The bad: if you stare into the abyss...

Digital payments before blockchains: 1980s-2007





1980s: Retail Payments

- **Goal: Digital payment system that**
 - Allows payments between customers and merchants (c2m)
 - Or between individual customers (c2c)
- **Strong cryptographic security**
- **Privacy**



1980s: Retail Payments

- **Some of the earliest ideas:**
 - Let's make digital cash!
 - Let's make digital checks!



“Digital cash”

- **Intuition:**

- Bills are just paper objects that have some data on them
- Why not convert them into digital files?
- I can “spend” a bill by sending the file to you and deleting my copy



The “double spending problem”

- Obviously this idea does not work:
- A malicious payer can simply retain a copy of the bill, and then spend it a second time with someone else
- We call this the “double spending problem” (DSP)
- How do we prevent double spending?



The “double spending problem”

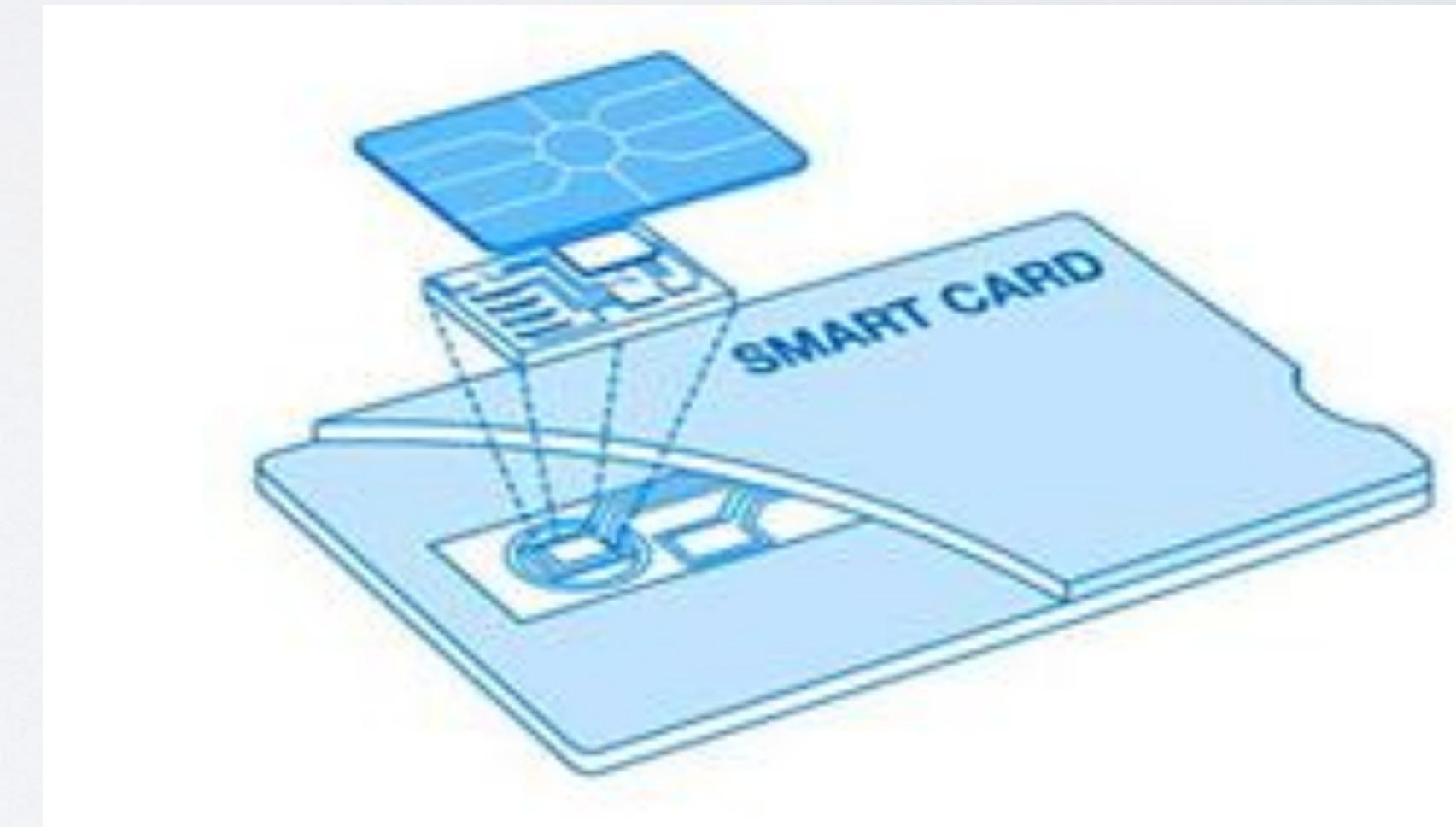
- Two common solutions to the DSP
 - **Online solution:** have a centralized party that records transactions
 - Can be serial numbers of bills, and who owns them now
 - Can be a full ledger of account/balances (more like “checks”)
 - **Offline solution:** build computers that never misbehave (i.e., they always delete the original copy of the bill...)

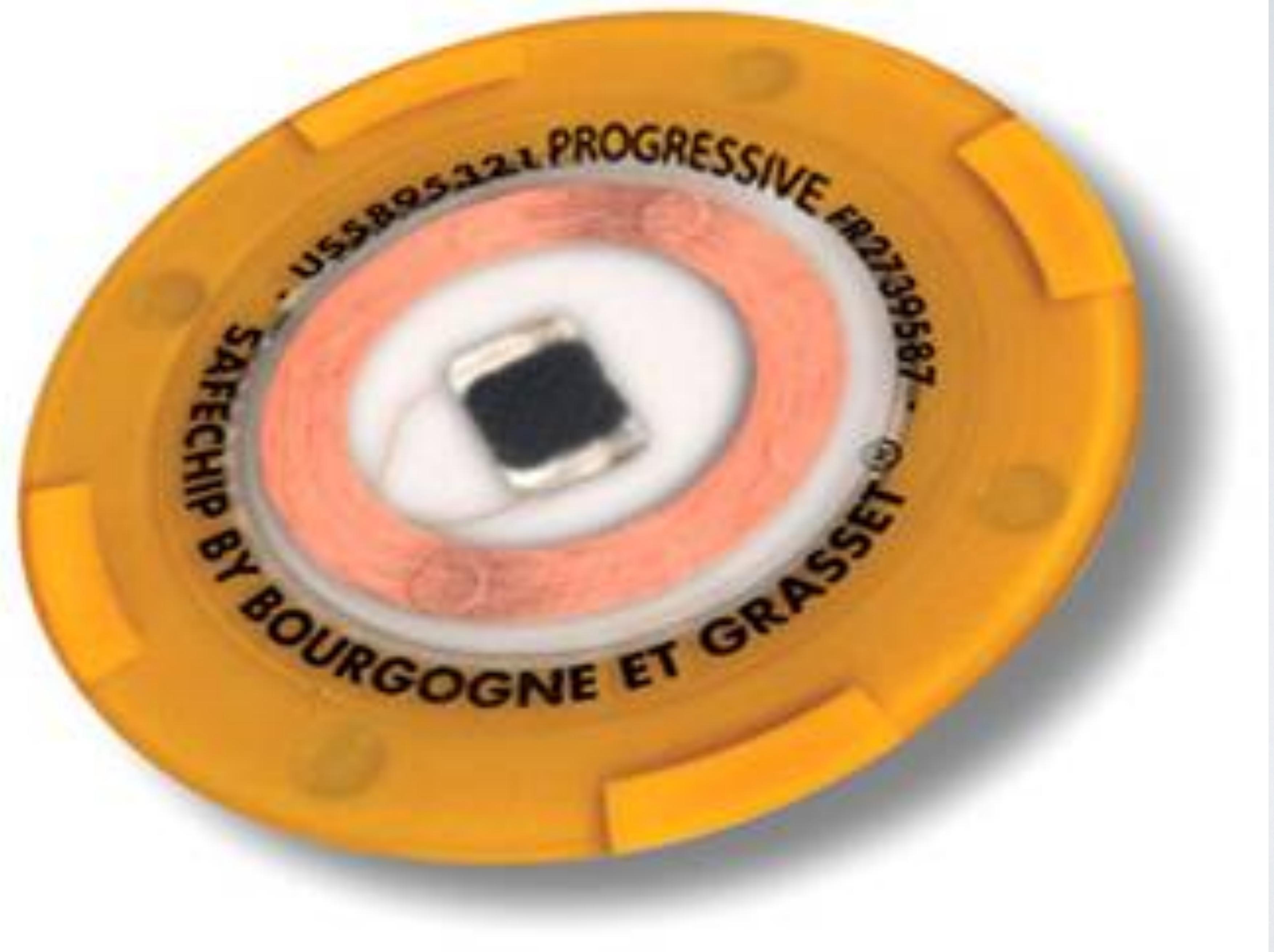


Offline cash

- **Some of the earliest ideas:**

- BankAmericard ATMs, 1980s: recorded debit balance on magstripe
- A few more recent systems use offline “smartcards” to store balances
- These almost always get broken if the values are high enough





\$1.5M Robbery of Bellagio Casino Foiled Thanks to RFID Chips

By **Aaron Saenz** - Feb 12, 2011

51,689

If you're thinking of robbing a Las Vegas casino, and you're not George Clooney, I have a word of advice: give up now. As Anthony Carleo recently found out, even if you leave the casino in one piece, the chips you stole are going to be worthless long before you make your get away. The 29 year old suspect is accused of robbing the Bellagio on December 14th of 2010, stealing chips whose face value totaled around \$1.5 million dollars. Their real value, however, was zero. Thanks to RFID tags embedded inside them, the chips with denominations of \$100 to \$25,000 could be immediately deactivated rendering them unredeemable for cash value. Watch CCTV footage from the December 14th robbery in the video clip below, followed by the recent press conference from the Las Vegas Police concerning Carleo's arrest. Stealing worthless chips and then getting caught trying to sell them to undercover officers? Danny Ocean this guy is not.

The many problems of online cash

- **Double spending**

- To capture double spending you usually need a (trustworthy?) online party

- **Integration with “real money”**

- If you're using fiat currency, you need to settle to that currency
- If it's some new form of cash, who mints it?

- **Identity & authorization**

- How do I know you have the right to spend?

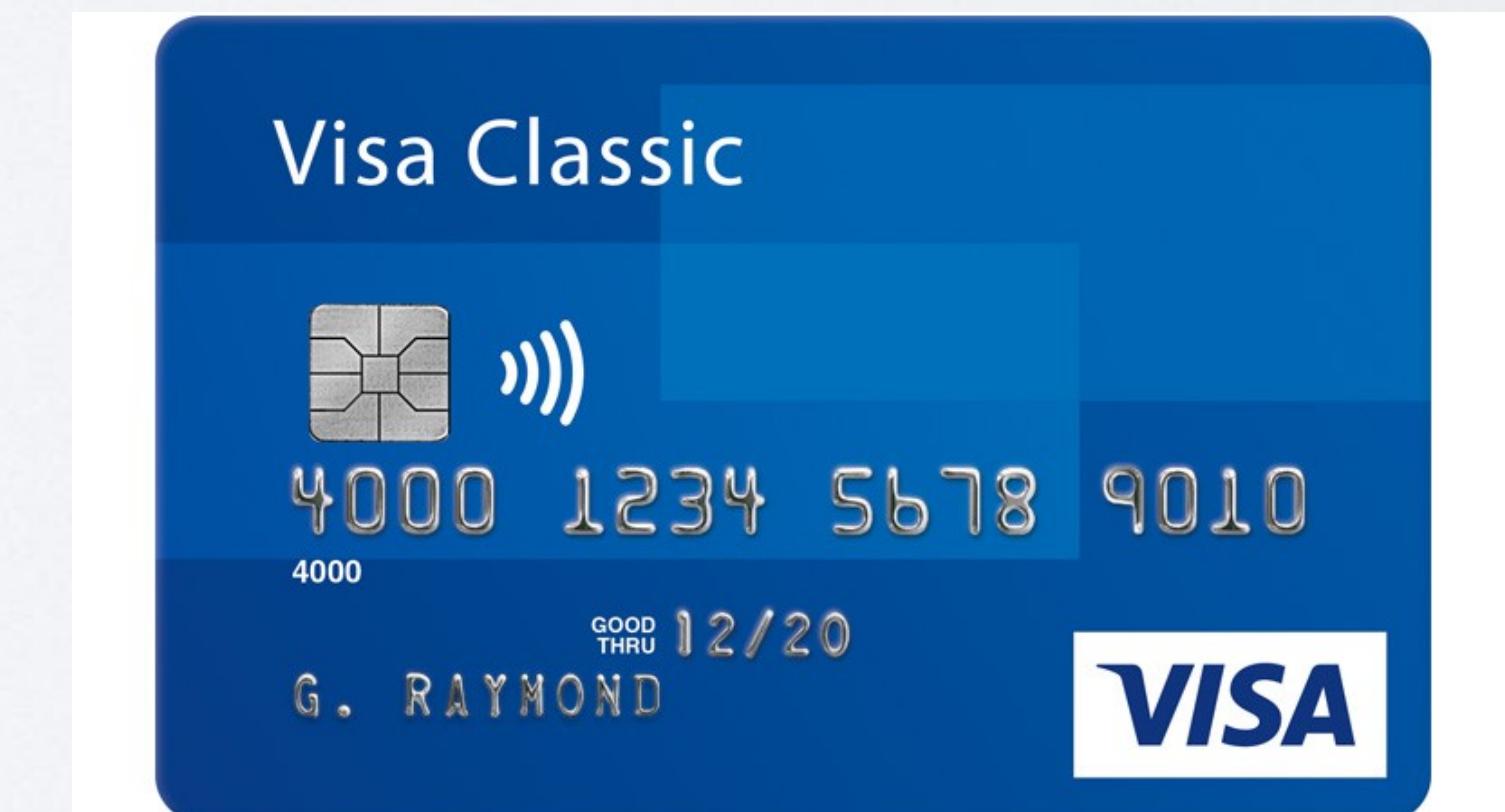
- **Privacy**

- The ledger sees all transactions



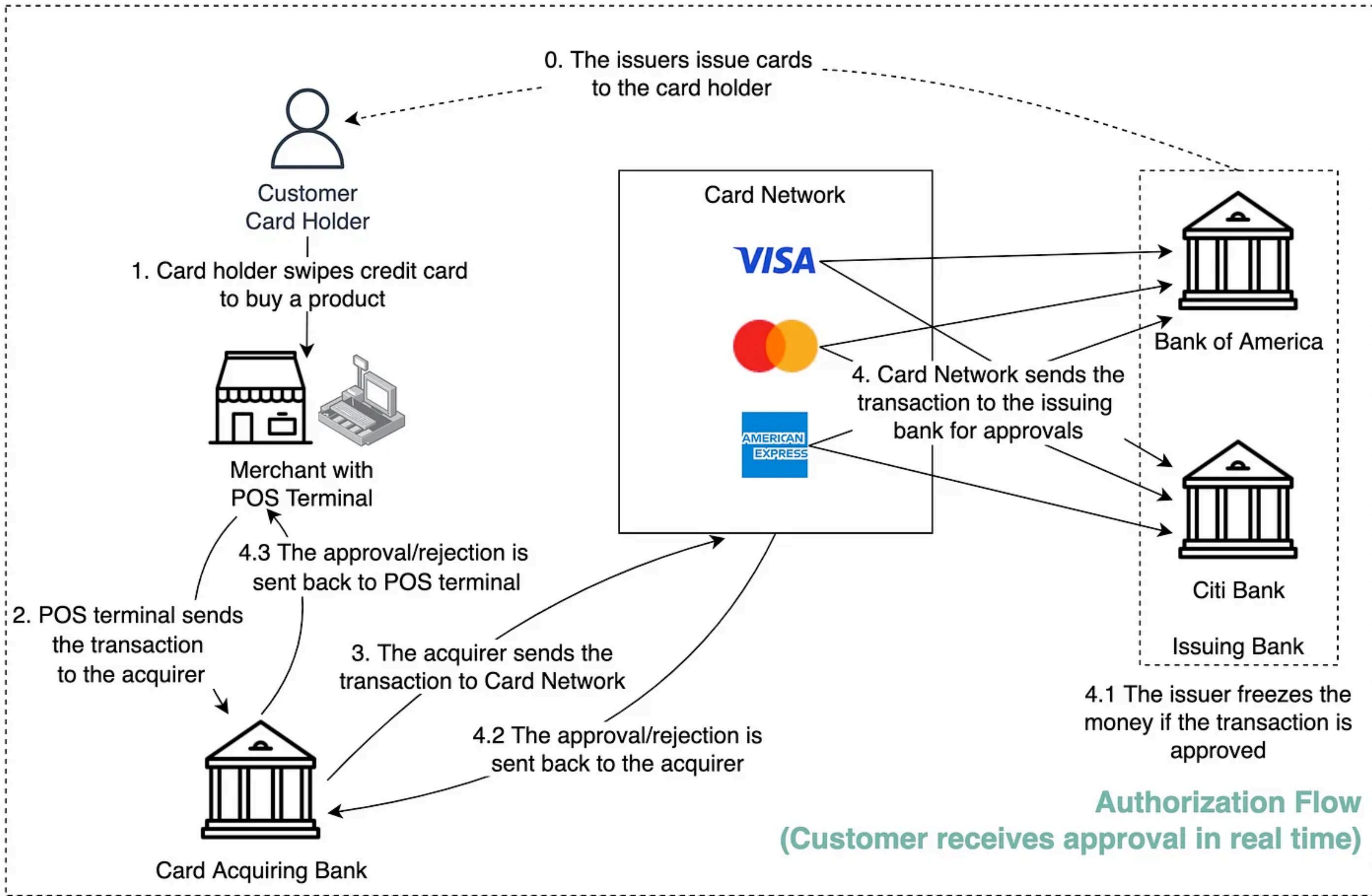
Centralized electronic \$

- Use a centralized bank database (“ledger”) to record account balances
- Require merchants/ATMs to contact the bank for approval
- Ledger can be “account-based” or “transaction-based”
- Typically it's both, and the two are reconciled

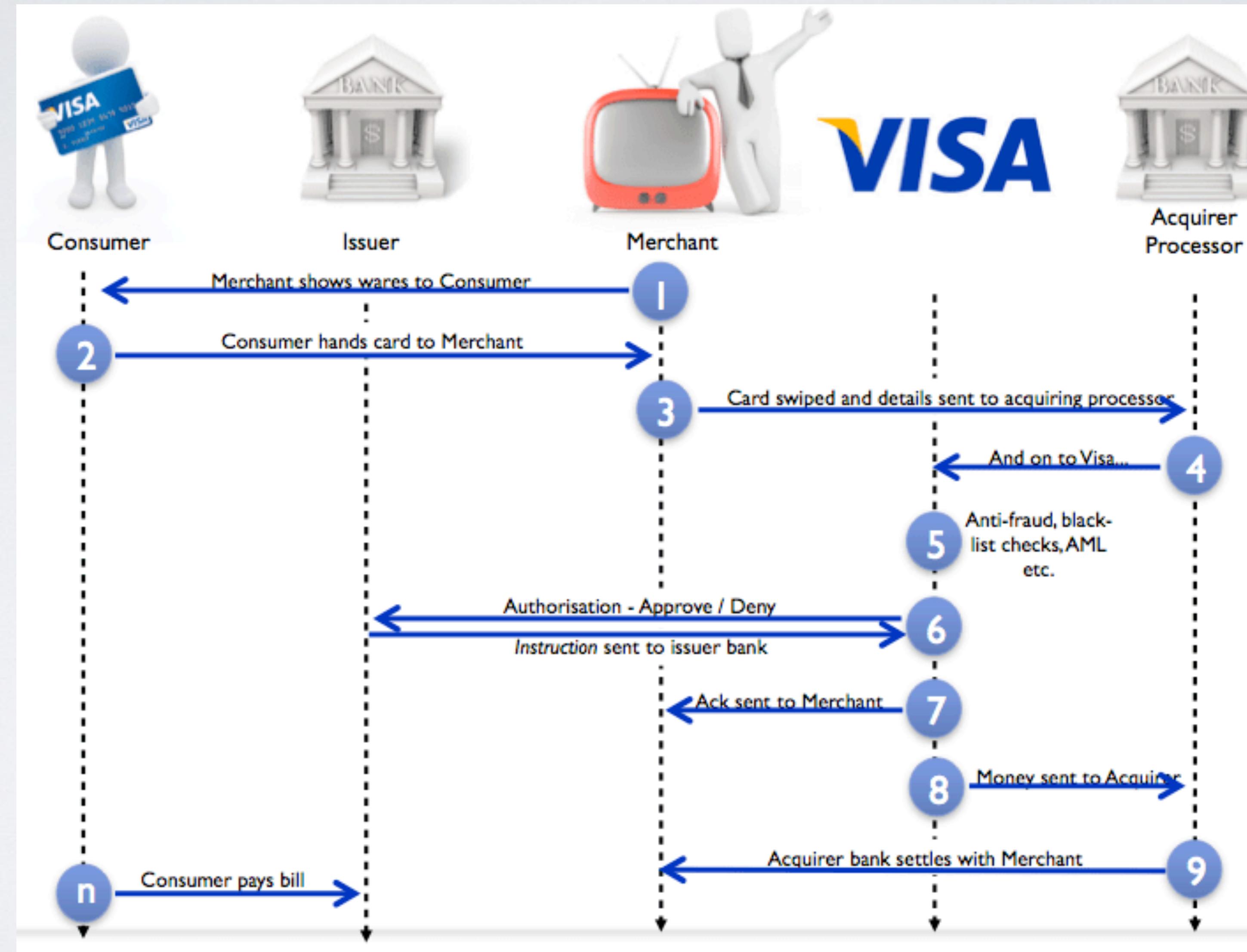


How does VISA Work?

 blog.bytebytego.com

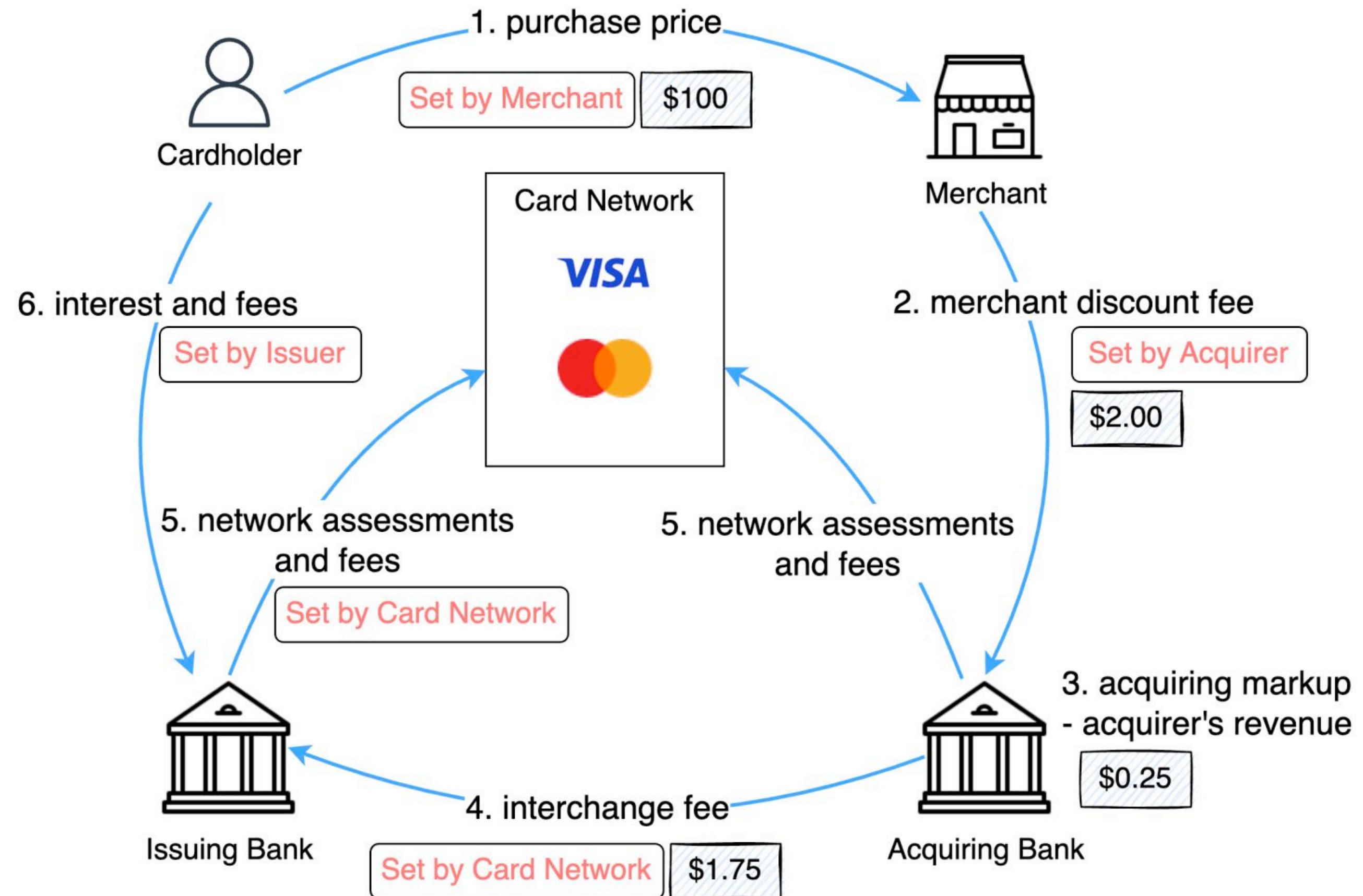


Credit card networks (2)



How does VISA Make Money?

 blog.bytebytego.com



merchant discount fee = interchange fee + acquiring markup

\$2.00

\$1.75

\$0.25

The merchant needs to compensate issuer and acquirer

Trust relationships

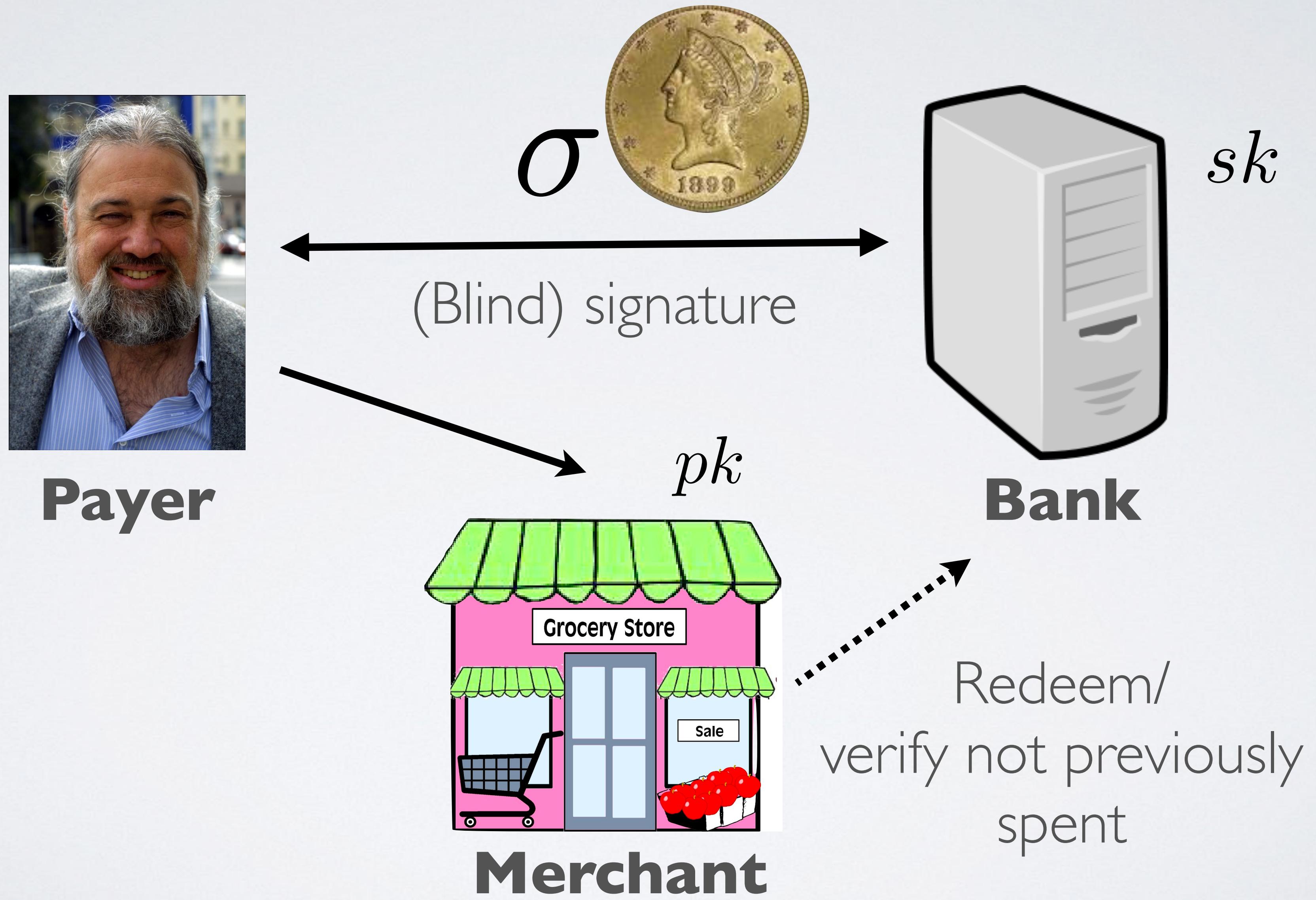
- In any system, parties must trust other parties to:
 - Correctly record transactions
 - Settle those transactions into actual “money”
 - This requires some degree of trust and creditworthiness between parties
 - As the number of parties increases, it becomes infeasible: hence the use of “hub and spoke” trust architectures

e-Cash

- Devised by Chaum, Chaum/Fiat/Naor, Brands, etc.
- Move to a “cash” model, with added privacy
- Individuals would carry redeemable tokens
- Reduces the problem to detecting double spending and user privacy



Chaum (CRYPTO '83)

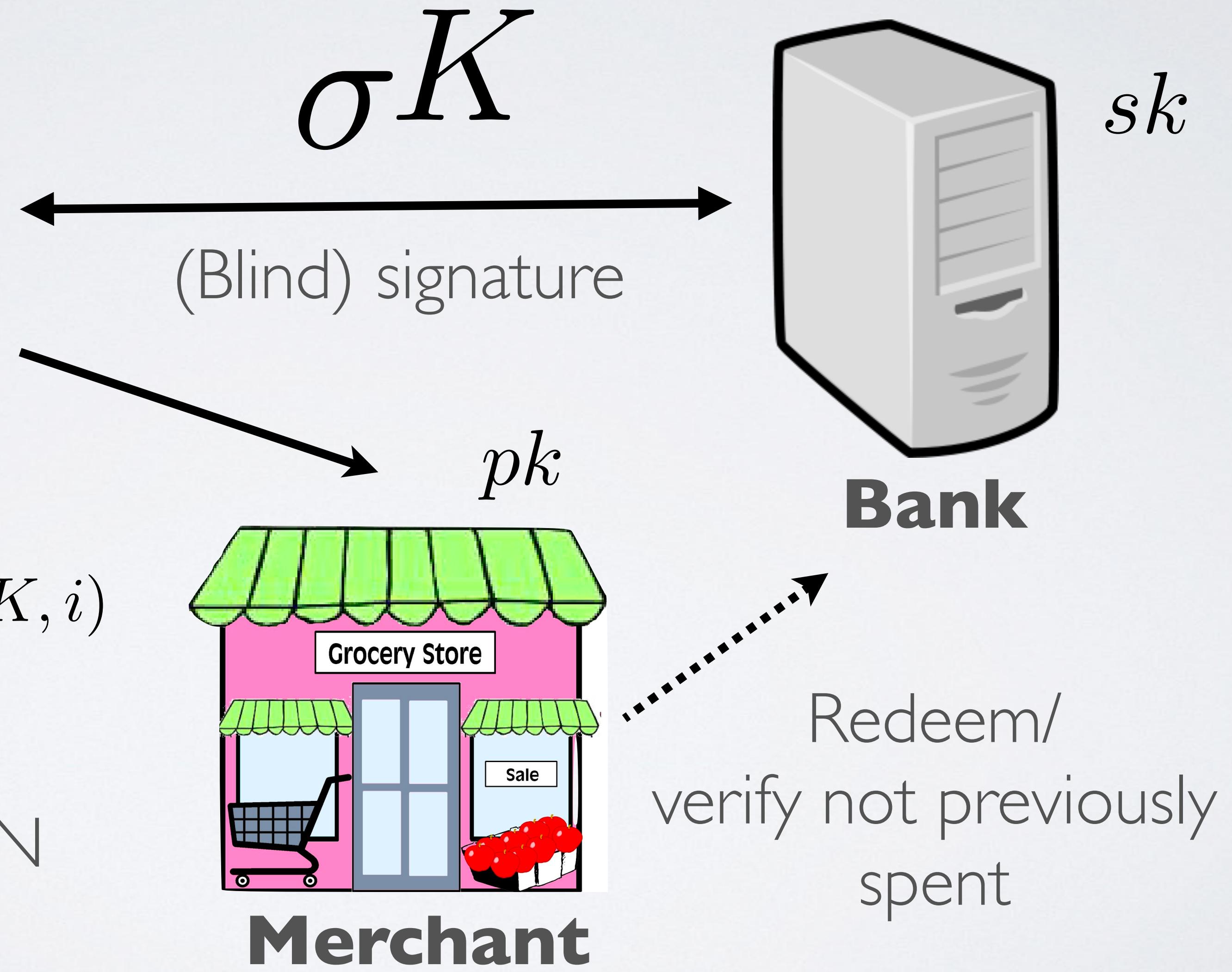


CHL (Eurocrypt '05)



Payer
 $SN = PRF(K, i)$
NIZK Π

For $I = 1$ to N



e-Cash

- Huge number of academic works / practical improvements
- Online schemes / offline schemes
- (Offline required using tamper-resistant storage)
- Main research problem continued to be privacy

≡ Google Scholar

"electronic cash"



Articles

About 35,600 results (0.09 sec)

Why did centralized e-Cash fail?

- Deploying e-Cash systems required a centralized bank
- Required a trusted server with money issuing powers
- In 1994, EU regulations made this more challenging
- 9/11 and beyond saw closures of *non-anonymous currencies* (e-Gold and Liberty Reserve)



Why did e-Cash fail? (2)

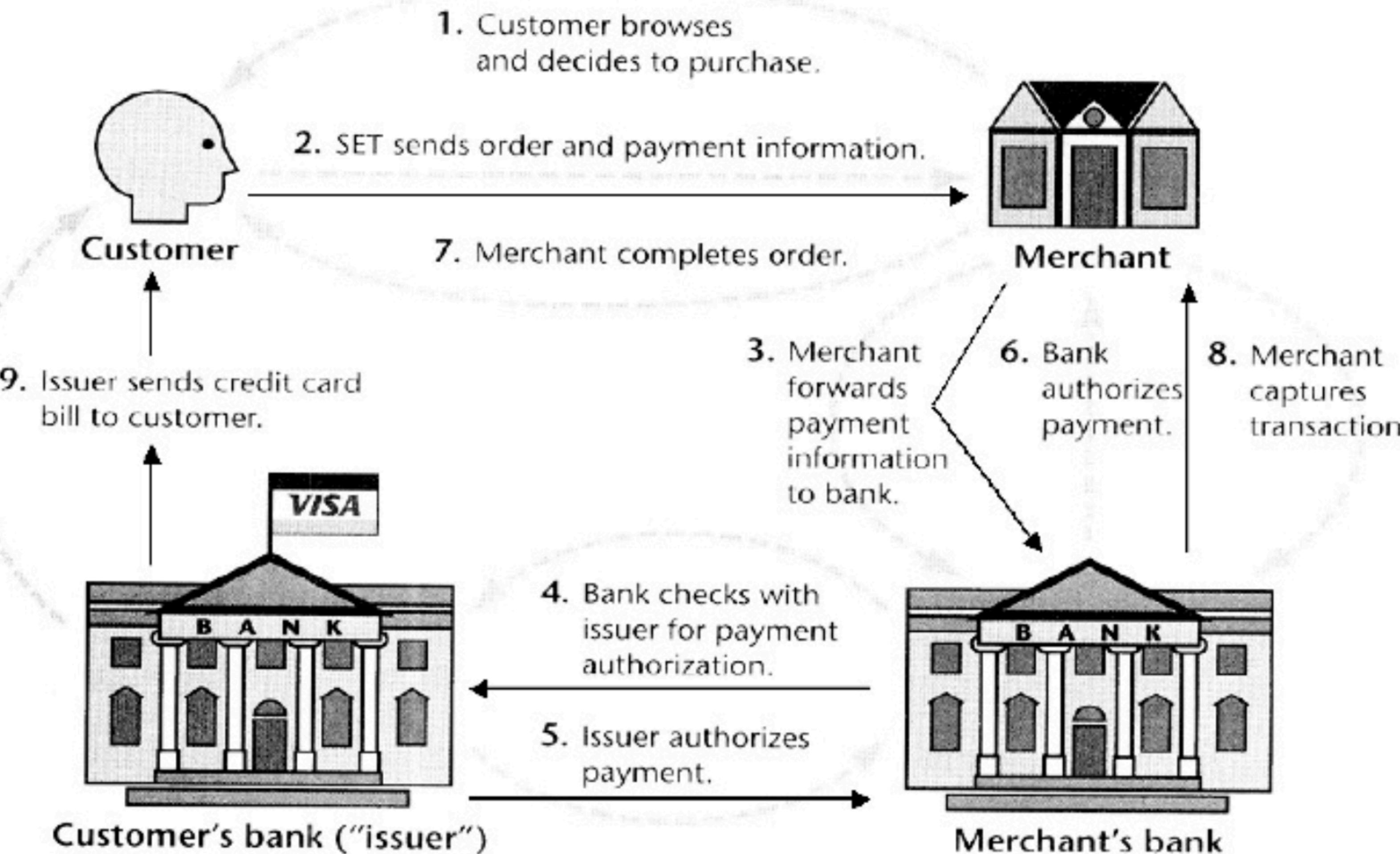
- Were these technical or policy failures? Maybe both.
- The e-Cash model was centralized and relied on a vulnerable interface with the banking system
- Privacy was (eventually) off the table for regulators
- Any solution would have to work around those (manufactured) technical problems

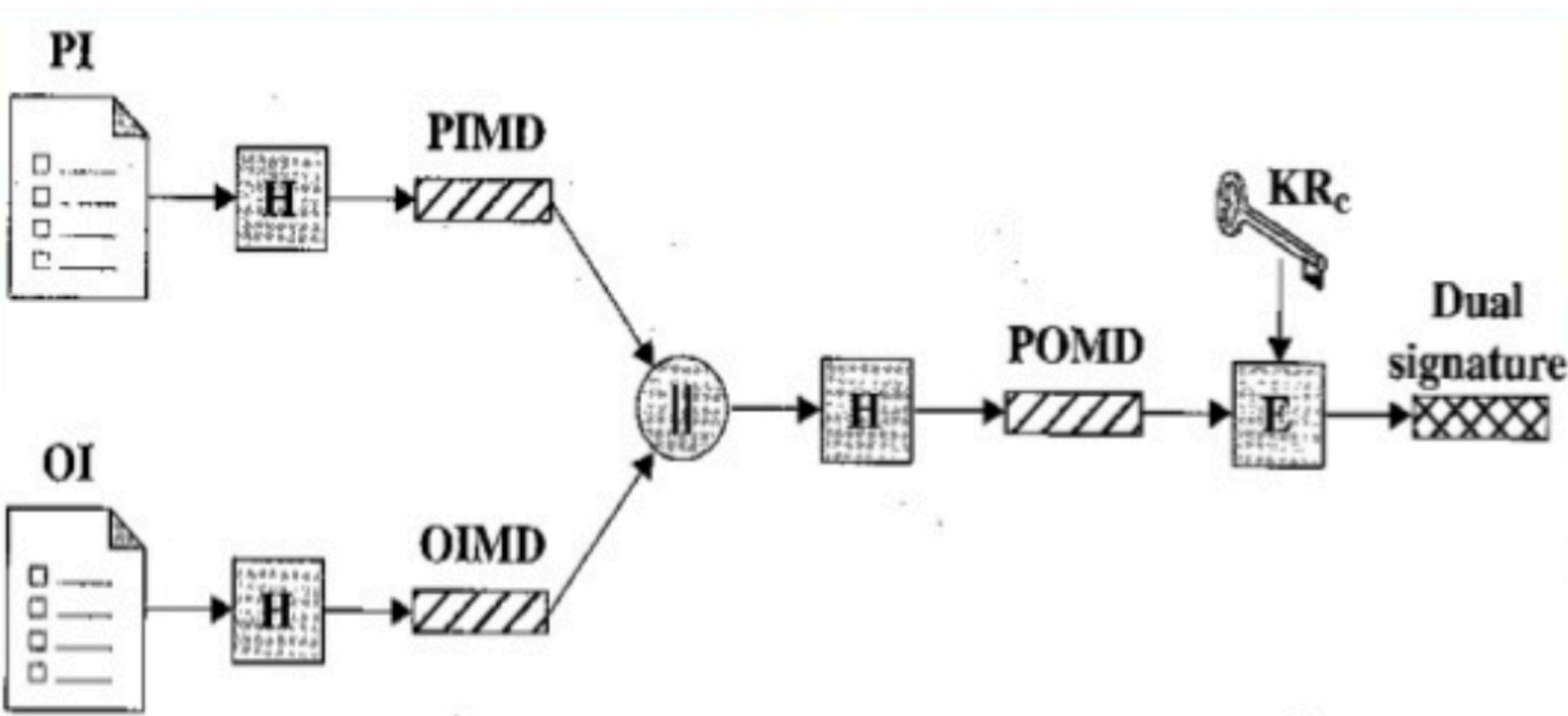


1996: SET

- Developed by Visa and MasterCard
 - Cryptographic architecture based on certificates
 - Assurance, authenticity and confidentiality







PI = Payment information

OI = Order information

H = Hash function (SHA-1)

|| = Concatenation

PIMD = PI message digest

OIMD = OI message digest

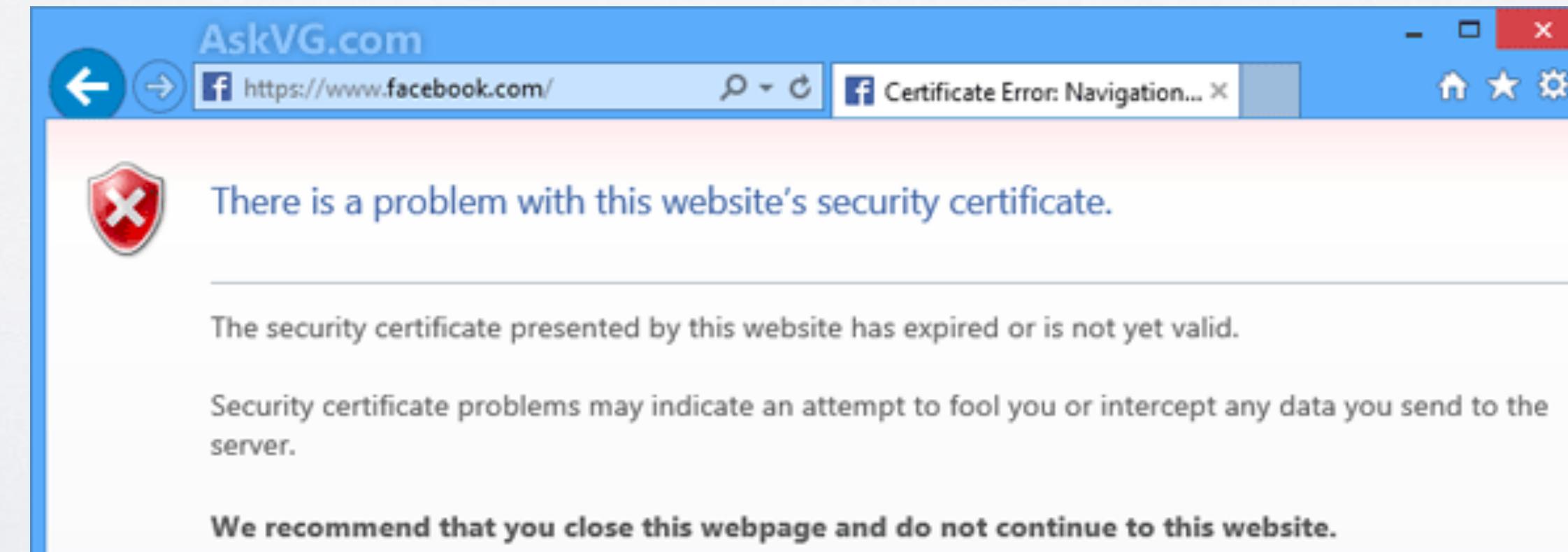
POMD = Payment order message digest

E = Encryption (RSA)

KR_c = Customer's private signature key

Why SET failed

- Required end-user certificates
- All the problems of key management PLUS all of the problems of identity verification
- Binding keys to user identities seems to trouble users



Liberty Reserve / e-Gold

- A few (centralized) systems tried to build “cash like” payments
 - Liberty Reserve: used a centralized database to issue a “stablecoin” (currency tied 1:1 to the US dollar)
 - e-Gold: similar idea, but tied to gold
 - Both became very popular in the early 2000s

Liberty Reserve / e-Gold



THE UNITED STATES
DEPARTMENT *of* JUSTICE

ABOUT

OUR AGENCY

OUR WORK

NEWS

RESOURCES

CAREERS

[Home](#) » [Office of Public Affairs](#) » [News](#)

JUSTICE NEWS

Department of Justice

Office of Public Affairs

FOR IMMEDIATE RELEASE

Friday, January 29, 2016

Founder of Liberty Reserve Pleads Guilty to Laundering More Than \$250 Million through His Digital Currency Business

The founder of Liberty Reserve, a virtual currency once used by cybercriminals around the world to launder the proceeds of their illegal activity, pleaded guilty today to running a massive money laundering enterprise, announced Assistant Attorney General Leslie R. Caldwell of the Justice Department's Criminal Division and U.S. Attorney Preet Bharara of the Southern

Conclusions (1980s-2007)

- Most cryptographic solutions too complex, or had “undesirable” features (privacy)
- Commercial solutions (existing credit cards, SET) failed to support the case of person->person transfers
- Web browsers didn't support fancy crypto anyway.
- **We got PayPal**





- Mc
- “ur
- Co
- sup
- We
- an)
- W

You can no longer use PayPal

At PayPal, we value a safer community in which our customers can do business. Some of your recent transactions violated our [User Agreement](#) and [Acceptable Use Policy](#).

Any bank account or card linked to your PayPal account cannot be removed or used to create a new account. You can still log in and see your account information but you can't send or receive payments. Any money in your balance will be held for 180 days, at which point we'll email you instructions about withdrawing your money.

Reference # PP-005-921-770-133

[Continue](#)

Conclusions (1980s-2007)

- Most systems were too complex, or had undefined security
- Current systems (including credit cards, SET) failed to support person-to-person transfers
- We don't support fancy crypto
- We don't support anything



Conclusions (1980s-2007)

- Most
under
standing
- Con
sider
support
- We
anyw
here
- We
are
not
the
best
in
the
world



The decentralized era

2008-2019



Nakamoto, 2008

- **Start with the basic intuition of e-Gold**
 - I.e., create a currency ledger on the Internet
 - Then remove the centralized components:
 - No centralized database that can be shut down
 - No backing currency/settlement



Nakamoto, 2008

- Replace the server with a **decentralized** ledger
- Use a new **distributed consensus** algorithm to allow many volunteer computers to maintain a consistent ledger



Decentralizing a ledger

- What problems arise here?



Nakamoto, 2008

- Use a new consensus technique to construct the ledger
- Uses a new data structure (“blockchain”) to store and cryptographically authenticate transaction history between nodes
- Use a new technique to determine which parties get to add new transactions, with network influence weighted by computational power (“proof of work”)



Nakamoto, 2008

- Eliminate the need for real-world currency interface, by creating a new currency entirely (Bitcoin)
- Eliminate the need for explicit identities and authorizations, by using public-key cryptography to authenticate and control currency ownership



Nakamoto, 2008

- Eliminate the need for real-world currency interface, by creating a new currency entirely (Bitcoin)
- Eliminate the need for explicit identities and authorizations, by using public-key cryptography to authenticate and control currency ownership
- **Everything else is straightforward crypto and excellent engineering**



Lessons of Bitcoin

- Getting the consensus algorithm right makes all the difference



Lessons of Bitcoin

- Getting the consensus algorithm right makes all the difference

[B]lockchain-style consensus indeed achieves certain robustness property in the presence of sporadic participation and node churn that none of the classical style protocols can attain.

- Pass, Shi 2018 (also '16, '17, Daian, Pass, Shi '16)



Lessons of Bitcoin

- Using the right consensus algorithm really makes a difference
- **Eliminating the need for key/identity management significantly simplifies the currency problem**



Lessons of Bitcoin

- Using the right consensus algorithm really makes a difference
- Eliminating the need for key/identity management significantly simplifies the currency problem
- **Human beings are weird**



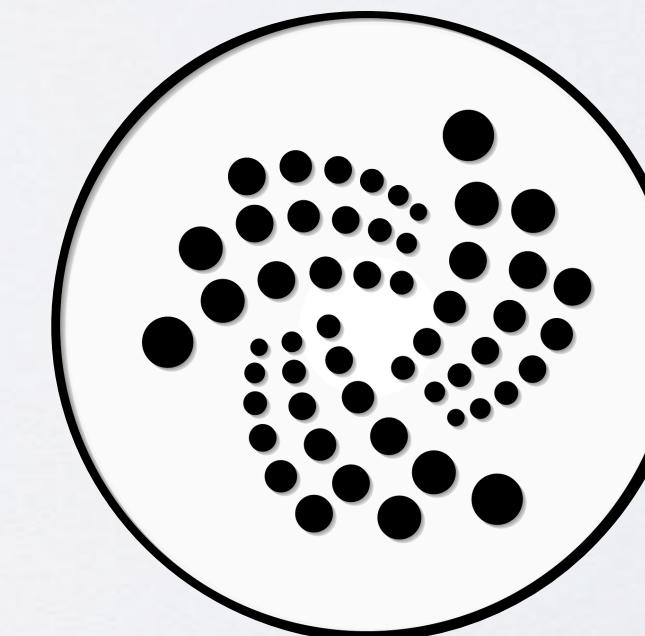
Lessons of Bitcoin

- **Useless tokens** have value.
- **Everyone** has a **useless token**.
- **Human nature** is **greedy**.
- **People** will assign significant value to **meaningless electronic tokens** — *if you* convince them that the tokens are **secure** and have a **predictable supply**.

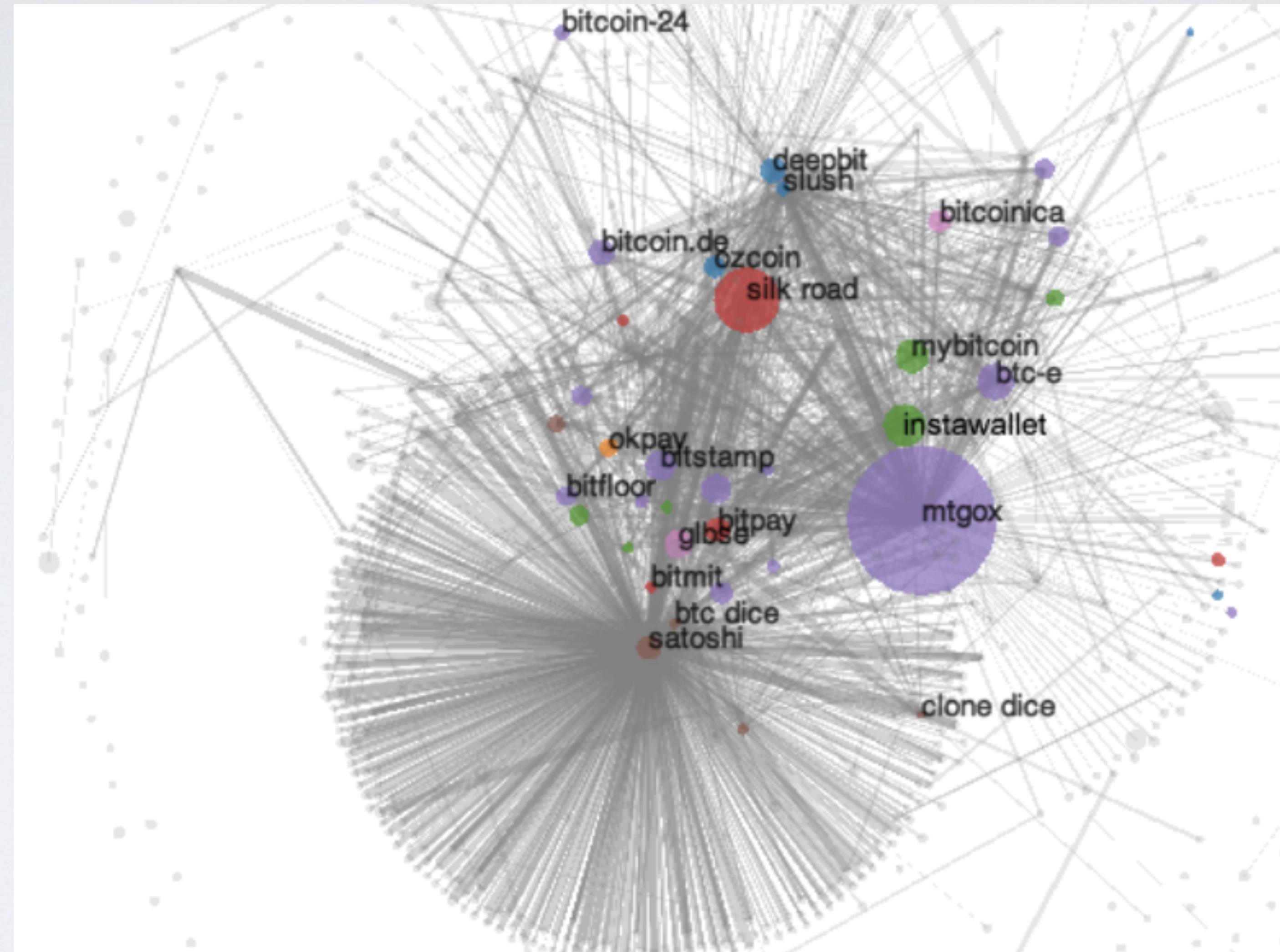


Limitations of Bitcoin

- Privacy limitations
- Functionality limitations
- Scalability & Sustainability limitations



Bitcoin & Privacy



Source: MPJLMVS13

Zerocoin/Zcash

WARNING

THIS IS DEVELOPMENT SOFTWARE. WE DON'T CERTIFY IT FOR PRODUCTION USE. WE ARE RELEASING THIS DEV VERSION FOR THE COMMUNITY TO EXAMINE, TEST AND (PROBABLY) BREAK. IF YOU SEE SOMETHING, **SAY SOMETHING!** IN THE COMING WEEKS WE WILL LIKELY MAKE CHANGES TO THE WIRE PROTOCOL THAT COULD BREAK CLIENT COMPATIBILITY. SEE [HOW TO CONTRIBUTE](#) FOR A LIST OF WAYS YOU CAN HELP US.

WARNING WARNING

NO, SERIOUSLY. THE ABOVE WARNING IS NOT JUST BOILERPLATE. THIS REALLY IS DEVELOPMENT CODE AND WE'RE STILL ACTIVELY LOOKING FOR THE THINGS WE'VE INEVITABLY DONE WRONG. PLEASE DON'T BE SURPRISED IF YOU FIND OUT WE MISSED SOMETHING FUNDAMENTAL. WE WILL BE TESTING AND IMPROVING IT OVER THE COMING WEEKS.

WARNING WARNING WARNING

WE'RE NOT JOKING. DON'T MAKE US PULL AN ADAM LANGLEY AND [TAKE AWAY THE MAKEFILE](#).

From payments to state

- Of course once you have a ledger...
- Each Bitcoin transaction can be considered a function $f()$ consuming some previous state and producing a state update
- Obviously this generalizes nicely to more complex programs and stored data



From payments to state

```
1 contract MetaCoin {
2     mapping (address => uint) balances;
3
4     function MetaCoin() {
5         balances[tx.origin] = 10000;
6     }
7
8     function sendCoin(address receiver, uint amount) returns(bool sufficient) {
9         if (balances[msg.sender] < amount) return false;
10        balances[msg.sender] -= amount;
11        balances[receiver] += amount;
12        return true;
13    }
14
15    function getBalance(address addr) returns(uint) {
16        return balances[addr];
17    }
18 }
19 |
```





What we'll talk about in this class

- Scaling blockchains to large numbers of transactions
 - Rollups, sidechains, channels, bridges
- Newer types of consensus network (e.g., Avalanche, BFT)
- Privacy
- Other advanced applications (not related to currency)
- Decentralized Finance,
- Identity...



Scaling

- Current Bitcoin/Ethereum transaction rate is ~7TX/s
- Compare with Visa at 10,000-40,000+ TX.s globally
- This gets worse as transaction complexity increases
- Problems are storage/throughput/validation bandwidth