Smart Contracts

Nicholas J. Szabo, The George Washington University Law School

http://szabo.best.vwh.net/

nszabo@law.gwu.edu

Overview of Talk

- Historical Importance of Transaction-Enabling Technology
  - Technology working alongside agreements and legal rules to facilitate transactions
  - Proto-money
  - Tamper-Evidence
  - Clocks and Work Schedules
- Smart Contracts
  - Increasingly, contractual logic itself will be embedded in technology
  - Security
    - Mapping between contractual terms and security protocols
    - Self-enforcing vs. evidence gathering
    - Privacy and integrity for multiparty protocols
  - Transaction Costs
    - Especially mental transaction costs
    - Value Measurement / Preferences (”what’s it worth to me?”)
    - Automated Negotiations, Contract Language

Style of Talk

- High level view of a lot of territory
- Goal is to give a taste of the importance and possibilities of smart contracts
- Mostly about the past and the future:
  - Historical, or
  - At the idea or development stage

Proto-Money

- Ostrich Eggshell Beads, 40,000 B.P.

- Mammoth Ivory Beads, 30,000 B.P.
  - c. 2 man*hours per bead

Proto-Money – Functions

- Store and Transfer Wealth
  - Transaction Costs
    - Avoid need for double-coincidence of provision of and desire for a specific good or service
    - Avoid large-scale favor-tracking
      - Credit / Reciprocal Altruism
  - Trade
    - Territory – Hunting and Gathering Rights ("Starvation Insurance")
    - Tools
    - Surplus Food
  - Marriage
  - Inheritance (kin altruism beyond the grave)
  - Tribute and Legal damages

Proto-Money – Security

- Secure Storage
  - On person
  - Burial
- Unforgeable Costliness
  - Mammoth ivory beads c. 2 man-hours per

- Eggshells < Seashells < Ivory < Silver < Gold
- Coins: reliably branded => lower assay costs
- Paper money – just rely on brand and currency markets

Proto-Money

- Shekels – coil and ring metal, c. 4,000 B.C.
  - To assay: (1) weigh, (2) cut at random points

- Coins (branded metal), c. 500 B.C.

Invention of Clocks

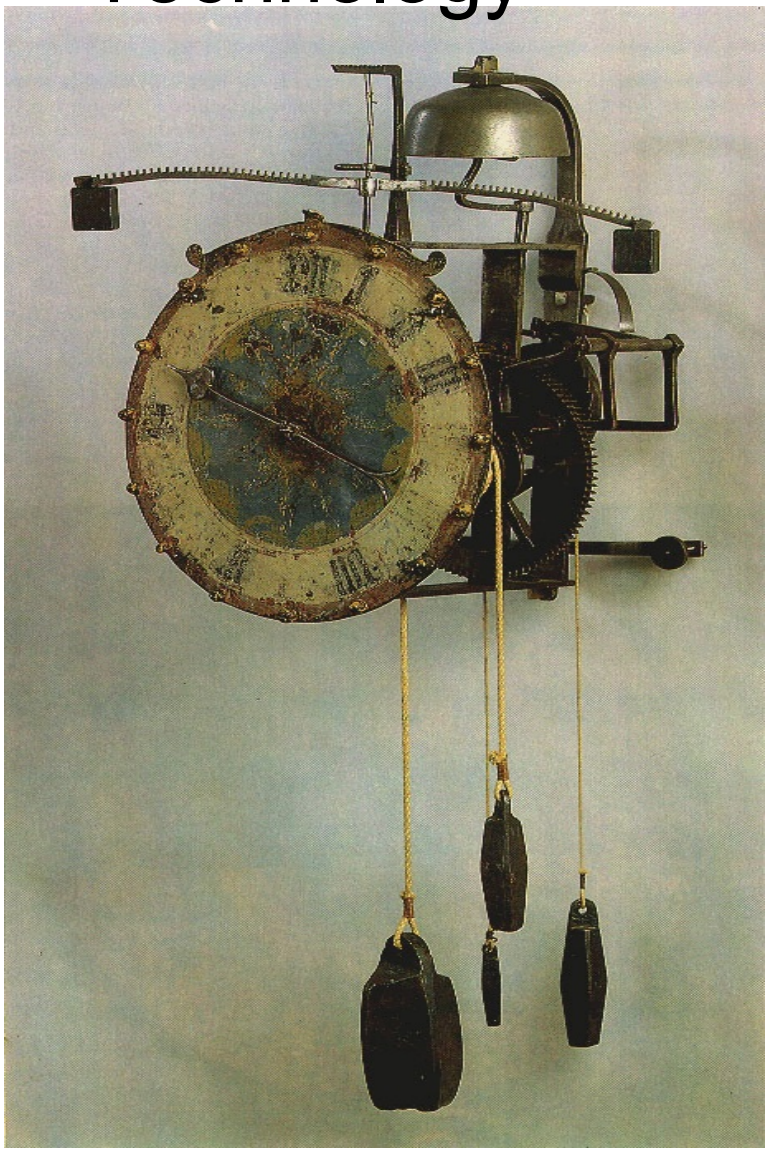
- Hourglass invented
  - sometime between 1275-1300, Northern Italy
- Mechanical clock invented
  - sometime between 1275- 1300, Northern Italy

Glass Technology

- The two technologies are very different

- So why did the inventions appear at the same time and place in history?

Mechanical Technology



- Already by 13th century, extensive use of tower bells in W. Europe to coordinate urban schedules
  - Warning signals
  - Church schedules
  - Work schedules

Bell tower, Ronciglione, southern Italy

Regular Hours

- Sundial – Unequal Hours

- matins--terce--------sext------nones----vespers

- Hourglass and Mechanical Clock – Equal Hours

- ---6---7---8---9--10--11--12---1---2---3---4---5

Bell tower and sundial in Vigenavo, Lombardy, Italy

Schedules – A Level of Indirection

Issue – Mental Transaction Costs

- How to schedule the town Baker, Butcher, Shoemaker, Candlemaker, etc.?
- Event bells
  - Guild rule = “CMs start work when CM bell rings”
  - Bell ringer(s) have detailed schedules
  - --Ba- Bu--Sh-CM--lunch----Ba--Sh,CM--Bu--
  - Complexity of schedule constrained by bell ringer knowledge and “bell space”
    - Another “bell space” – recognizing the sound of your cell phone
    - cf. name space

Schedules – A Level of Indirection

Issue – Mental Transaction Costs

- Event bells
  - Guild rule = “CMs start work when CM bell rings”
  - Bell ringer(s) have detailed schedules
  - --Ba- Bu--Sh-CM--lunch----Ba--Sh,CM--Bu--

- Clock bells
  - Guild rule = “CMs start work at X o’clock”
  - ---6---7---8---9--10--11--12---1--- 2---3---4---5
  - Complexity of schedule no longer constrained by ringer’s knowledge or “bell space”

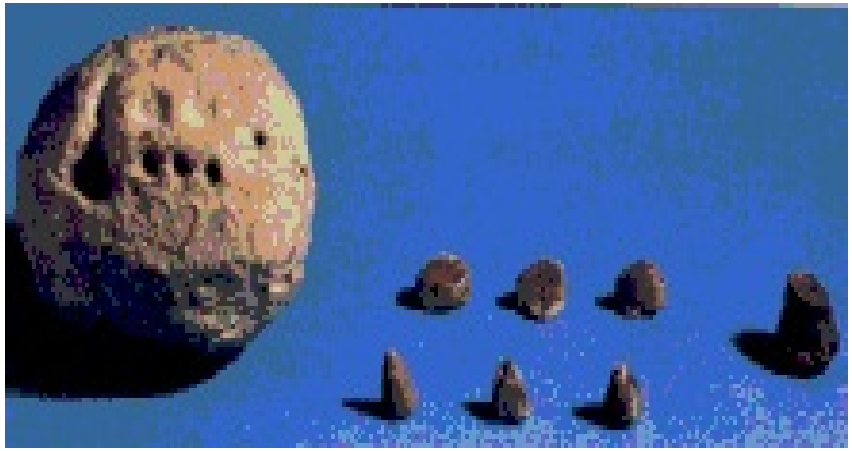
Honest Time

Security vs. Schedule Spoofing

- Level of indirection also means: many schedules impacted, increases chances of getting caught
- Tower bells – broadcast
  - Everybody gets the same time
    - Not secure on modern computer networks!
- Often 2 people in clock tower to keep each other honest
- Incorruptible heavens
  - Clocks can be checked most nights against the stars
- Hourglass
  - Possibility of catching even small discrepencies

Tamper Evidence

- Seals and Sealings
  - Tamper-evidence for stored or transported goods
- Clay tokens
  - Warehouse receipts
  - Bills of lading
  - Bearer instruments

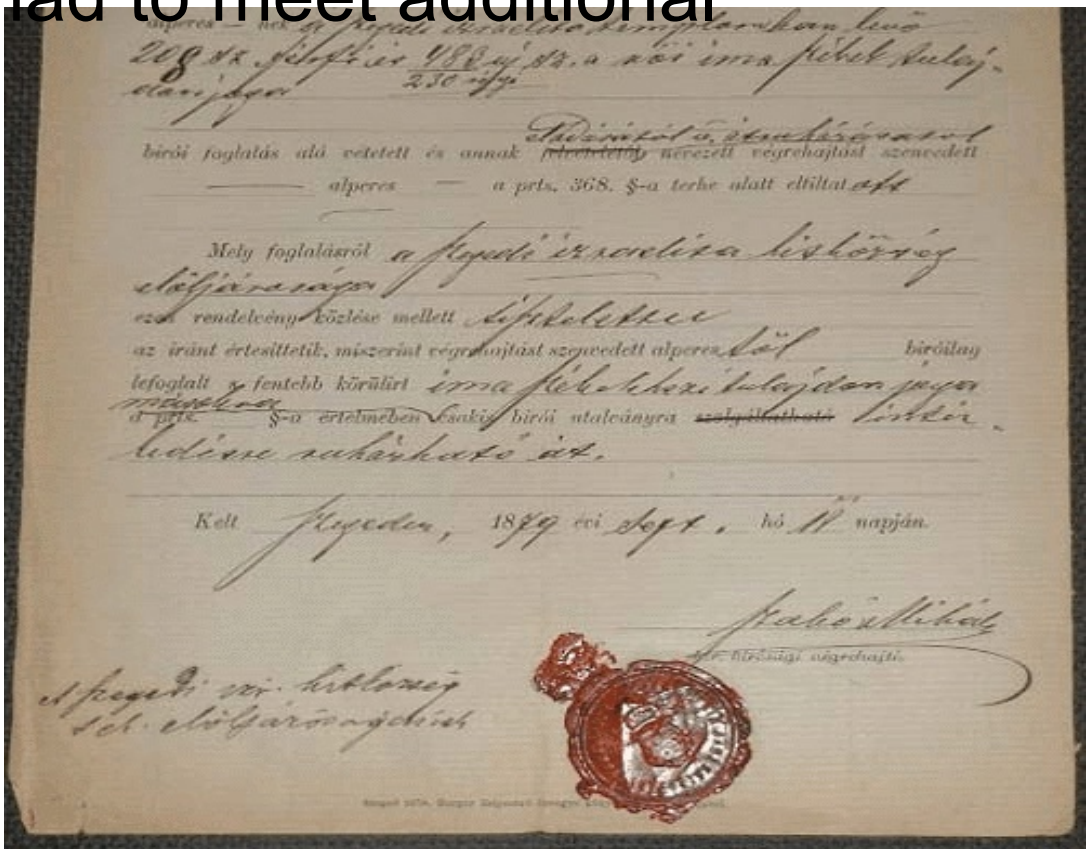


Tamper Evidence

- Modern examples
  - Evidence bags
    - secure evidence trails
    - Police
    - Banks & other fiduciaries
  - TSA bag closures
  - Door seals
  - Serial number has been recorded. To defeat system, must both access and forge the serial number log and replace with a new device matching the new serial number
  - cf. data – “Post-unforgeable” – sealed in amber

Tamper Evidence & Caution

- Traditionally, English contract law took seals very seriously:
  - Promises in writing and sealed were always enforced
  - Promises not sealed had to meet additional criteria
- Seal here has two functions
  - Security
    - Tamper-evidence
  - Mental transaction cost
    - Cautionary



Caution

- Lon Fuller (cautionary principle)
  - Don't enforce unless party has considered the contract carefully before committing to it
- Donald Norman – affordances
  - Don’t make it easy for user to commit to an important contract
- “Smart fine print” problem
  - White Box approach – fine print is bad
  - Black Box approach – fine print is OK; customer goes by reputation of the counterparty
    - Judge Easterbrook, *ProCD, Inc. v. Zeidenberg* (1996) – shrink wrap software
      - By opening software package and not returning it to the store, you agree to the terms written enclosed *inside* the package.
      - users don’t inspect computer code, so why should they inspect legal code (contract terms)?

Legal Code vs. Software Code

	Legal Code	Software Code
Logic	grounds on subjective * Iwaihara, Jiang, Kambayashi	grounds on bits
Security	Contempt/imprisonment	In code or hardware (can also be legal or reputational)
Predictability	Flexible	Rigid, Fragile
Maturity	Highly evolved / many cases	Novel / few cases
Economics	Lawsuits expensive	Cheap, once R&D amortized

4 Kinds of Contractual Relationships

- Caveat Emptor – buyer can measure value of goods to him ex-ante (before purchasing)
- Long-Term Relationship – buyer can measure value of goods to him ex-post (after using it) and cost of repeat purchases is low

- State-enforced Contract – warrantee – buyer and third party (court) can measure value of goods ex-post but cost of repeating the purchase is too high
- Firm Integration / Employment – none of the above hold, and transaction becomes work of an employee under supervision

Caveat Emptor or Long Term Relationship

Caveat Emptor or Long-Term Relationship

Firm/Employment

- Control employee access to customer cash

Smart Contract

- A protocol that helps execute the terms of a contract
- Challenges of smart contracts:
  - Security
    - Self-enforcement
    - Evidence
      - Observation by parties in privity
      - Verification by adjudicator
  - Mental Transaction Costs
    - Measurement of Value
    - Ex Ante
      - Negotiations – should I agree to this smart contract?
    - Ex Post
      - Determination of damages by adjudicator

Smart Contracts

- Problems w/state enforceability
  - Economic – Lawsuits cost big $$$
  - Moral – Monetary damages are backed up by threat of imprisonment
    - "Contempt of court"
    - Security that avoids threats of imprisonment is morally superior

Smart Contracts

- "Candy Rights Management"
  - Integrated transaction: takes in coins, distributes goods
  - Actions defined by a state machine
  - Security
    - Mostly self-enforcing
    - Economics: breach cost > amount in till
  - Mental transaction costs
    - Is it worth $0.75 to buy a Snickers?
    - Which buttons to push & coins to put in to buy Snickers?

Smart Contract

Candy Vending – Specification of State Machine

```
sellCandy(candyPrice = $0.90) =
   variable moneyAmount = $0.00
   then
      # coins also fall into a temporary till tempTill
   when choiceOf(Counterparty, nickel)
      to TempTill nickel
      then to Counterparty add(moneyAmount, $0.05)
      then to Counterparty display(moneyAmount)
   when choiceOf(Counterparty, dime)
      to TempTill dime
      then to Counterparty add(moneyAmount, $0.10)
      then to Counterparty display(moneyAmount)
   when choiceOf(Counterparty, quarter)
      to TempTill quarter
      then to Counterparty add(moneyAmount, $0.25)
      then to Counterparty display(moneyAmount)
   when choiceOf(Counterparty, moneyReturn)
      to Counterparty dropCoins(tempTill, returnTill)
      with moneyAmount = $0.00
      then to Counterparty display(moneyAmount)
   when threshold(moneyAmount, candyPrice)
      to Holder (nickel | dime | quarter)
         to CounterParty redirectNewCoinsTo(returnTill)
      also display("ready to dispense --please select candy")
      then when (candySelection)
         to Counterparty dropCandy(candyRacks, candySelection)
         with to PermanentTill dropCoins(TempTill)
         with moneyAmount = $0.00
   continue
```

Accounting Controls

Paper practices carried over to IT

- "Data flow" – flow of forms
- "Controls" – checks and procedures
  - "Checks and balances" – separation of duties and cross-checking
- These traditional controls serve many of the same functions as cryptography – integrity, authorization, confidentiality, and so on

Smart Contracts

The Auto-Repo Auto

- If we could design the perfect auto security system, what would it be?
  - First cut – a perfect lock to let in the owner and exclude third parties
    - Problem: excludes a contractually interested party (creditor)
  - A special key to let in the creditor (repo man)
    - Problem: if perfect, then creditor should not be able to get in if the payments have been made
  - Tie in to payment system – creditor key switched on only if in arrears on payment
    - The final payment permanently switches off creditor key
- We've successively refined the security specification until it is isomorphic with the contractual terms

Smart Contracts

The Auto-Repo Auto

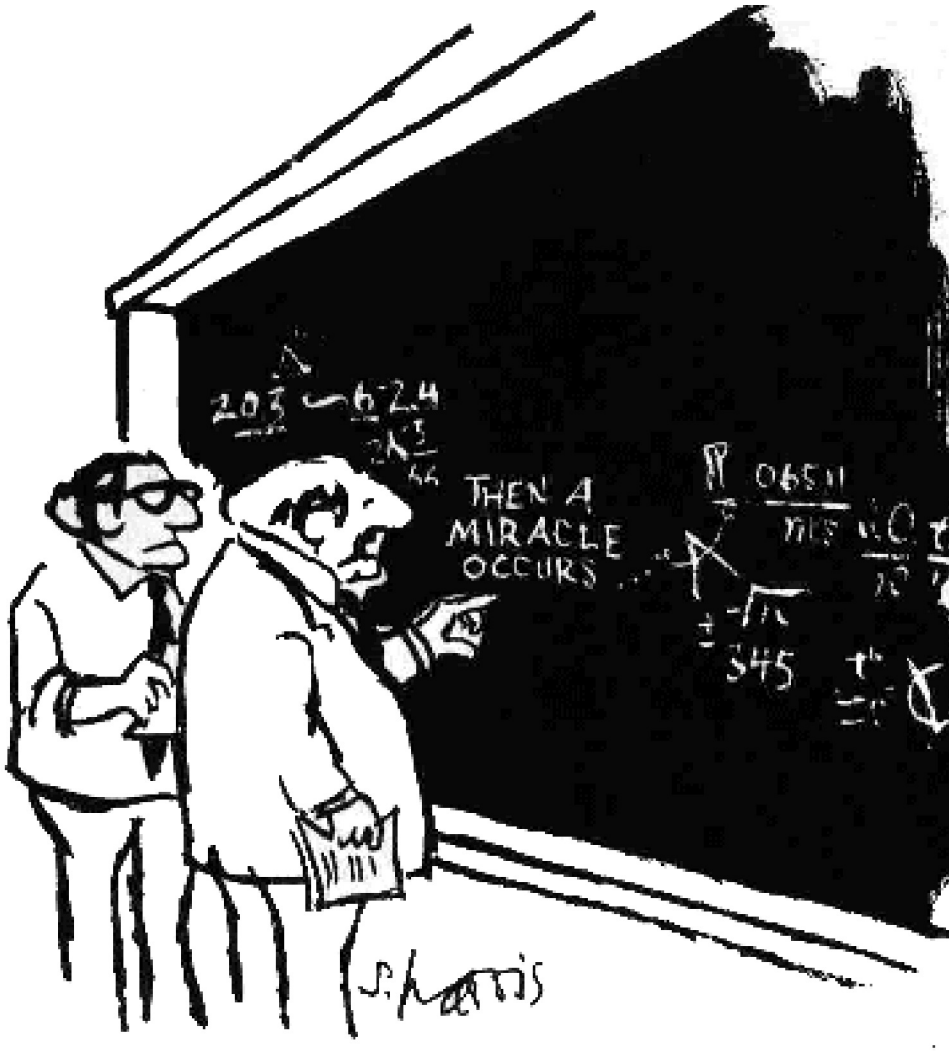
- Smart contracts as security paradigm – the security protocol should be isomorphic with the contractual terms
  - Security protocols enforce or gather evidence of contract performance
  - Policy and enforcement can be bundled
- In real world, reliability and user error are also important
  - E.g. OnStar, creditor can let you into car if you've been locked out, but can always get in
  - Depends on legal system rather than security system to enforce the contractual properties

Multiparty Protocols

- Multiparty Protocols – some applications
  - Auctions
  - Exchanges
  - Contract Intermediary
    - Mediator
    - Adjudicator (ex post)
      - Judge
      - Person at credit card company who decides whether chargeback is legitimate
  - Property titles service

Multiparty Protocols

- Trusted Third Parties (TTPs) are security holes
  - **Takes very costly traditional controls to actually create trustworthy fiduciary organization:**
    - Segregation of duties
    - Secure buildings and rooms
    - Surveillance
    - Tamper-evident devices
    - Etc.
  - **Tempting in security protocol design**
    - "And then a miracle occurs"
  - **Minimize trust assumptions**

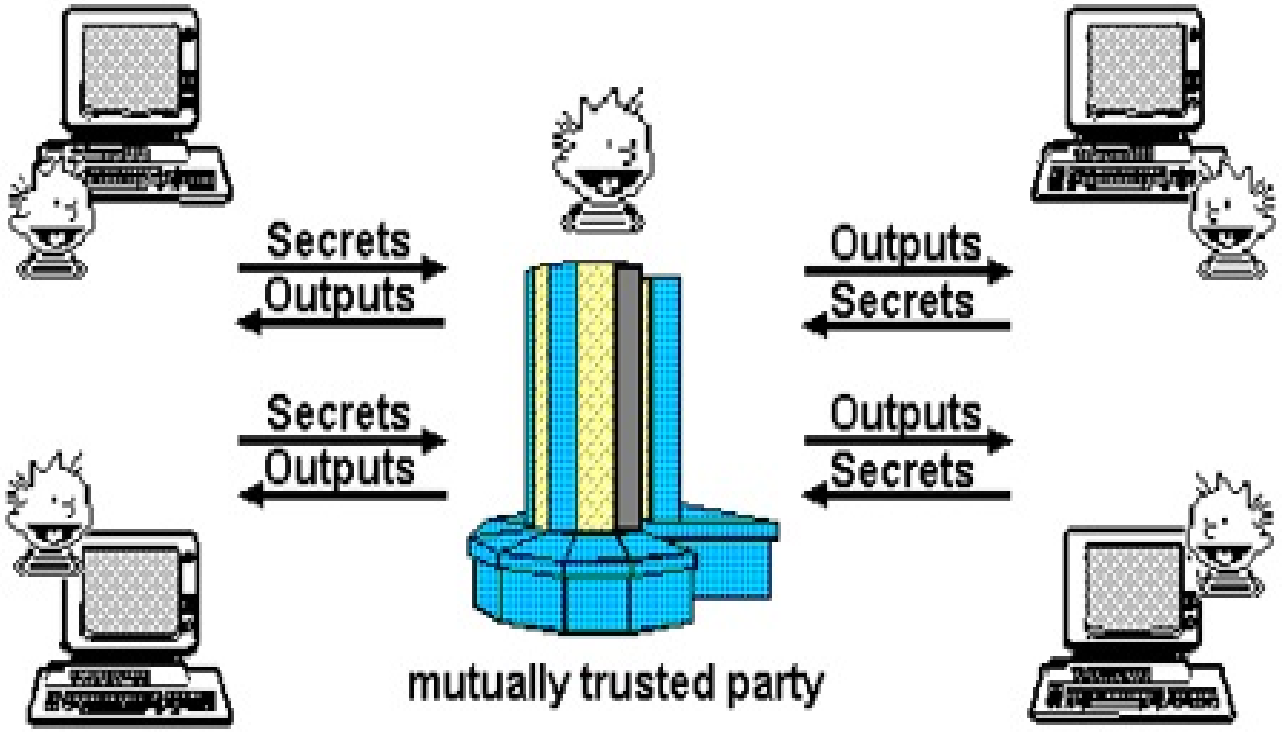


"I THINK YOU SHOULD BE MORE EXPLICIT HERE IN STEP TWO."

Multiparty Protocols

Privacy – Multiparty Secure Computation

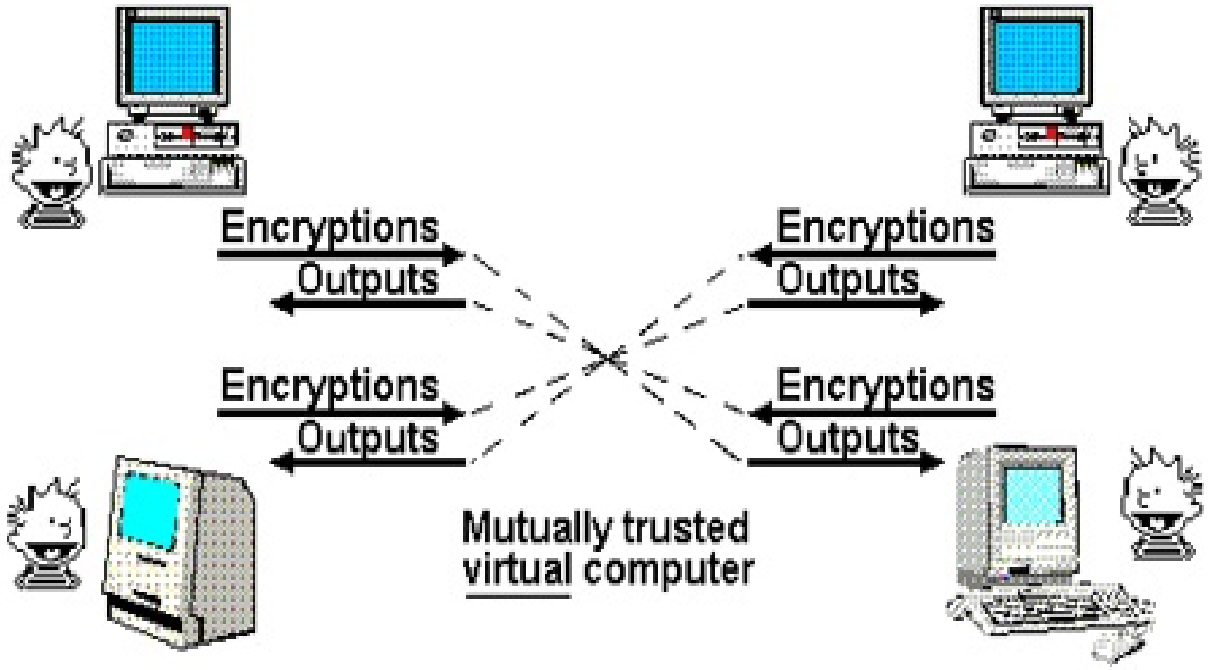
- Problem -- privacy loss is not verifiably reversible



Multiparty Protocols

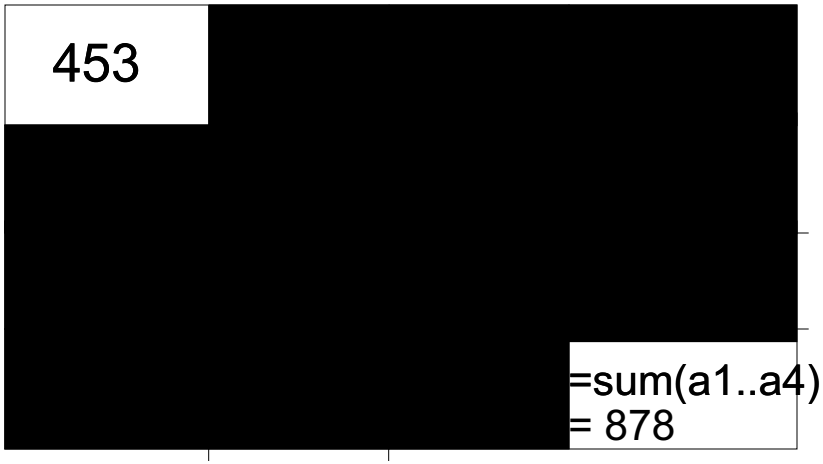
Privacy – Multiparty Secure Computation

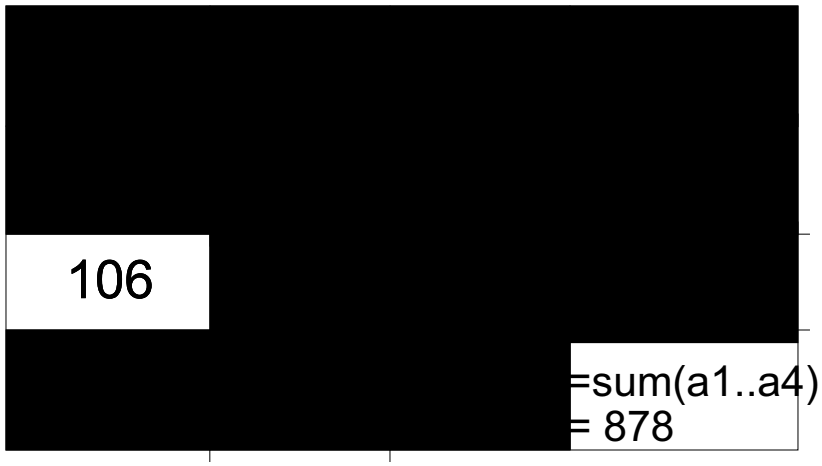
- Secures privacy

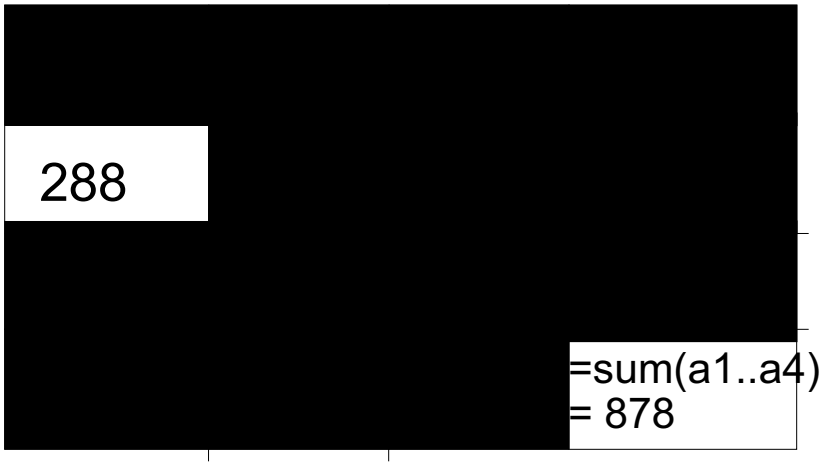


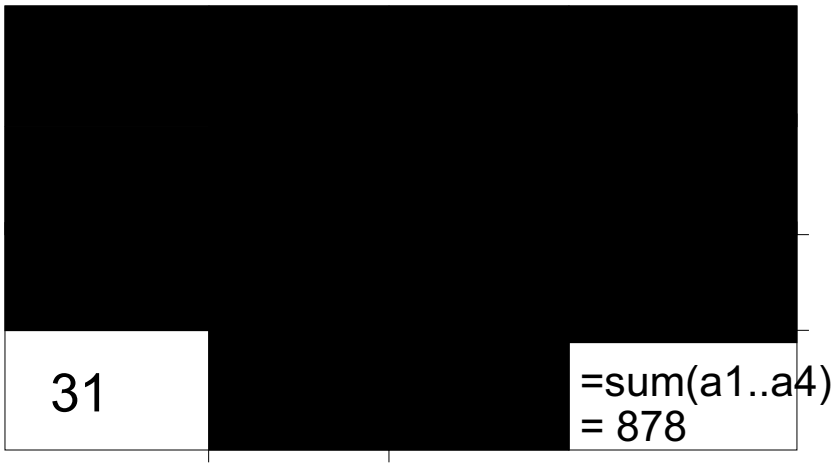
Multiparty Protocols

Privacy – Multiparty Secure Computation









Multiparty Protocols

Multiparty Secure Computation

- Often slow (many network messages)
- So we often use special purpose protocols:
  - Multiparty signing (takes N out of M to sign a document using a distributed key)
  - Digital cash mint
  - Secure auctions
- Financial Cryptography conferences

Unforgeable Transaction Logs

- "Unforgeable" is goal
- Post-Unforgeability – Prevent forging of log ex post (after the transaction)
  -
  - Digital Signatures **Digital Timestamping** – Publication of chained hash functions
- Pre-Unforgeability – Prevent forging of inputs ex ante (before or during the transaction)
  - Computation replicated:
    - Byzantine replication
    - Each party performs an identical function
  - Computation separated into functions
    - Segregation of duties / check integrity rules
    - Each party performs a different function
    - Each party checks to make sure others follow integrity rules

Multiparty Protocols

Integrity and Reliability

- Replicated computation
  - Theoretical model – Byzantine generals
  - Reliability – can complete with e.g. X/Y working
  - Integrity – can complete with e.g. 1-X/Y working

Multiparty Protocols

Integrity and Reliability

- Computation divided into separate functions
  - Reliability – requires all nodes to be working
  - Integrity – prevents or detects certain breaches if any node is working
  - Examples:
    - Purchasing and Sales cycles: accounting controls
      - Separation of duties: sales, warehouse, accounting
    - Life cycle of a law:
      - Legislative (draft / approve)
      - Executive (endorse / enforce)
      - Judicial (final say / interpretation)
    - Chaumian mixes (remailers)

Terminology

- Don’t take terminology too seriously

Called	Really More Like
“Digital Signature”	Seal
“Digital Cash”	Bearer Certificate

Bearer Certificates & Scarce Objects

- Digital cash (Chaum, 1983)
- Generalization – Bearer Certificates
  - A.k.a. tickets, tokens, etc.
  - Represent a standard service
  - Fungible
- Access Control
  - ACLs, capabilities, etc
- vs. Usage Control
  - Scarce objects
  - Agorics (online micromarkets)
    - (Miller et. al.1980s)

Mental Transaction Costs

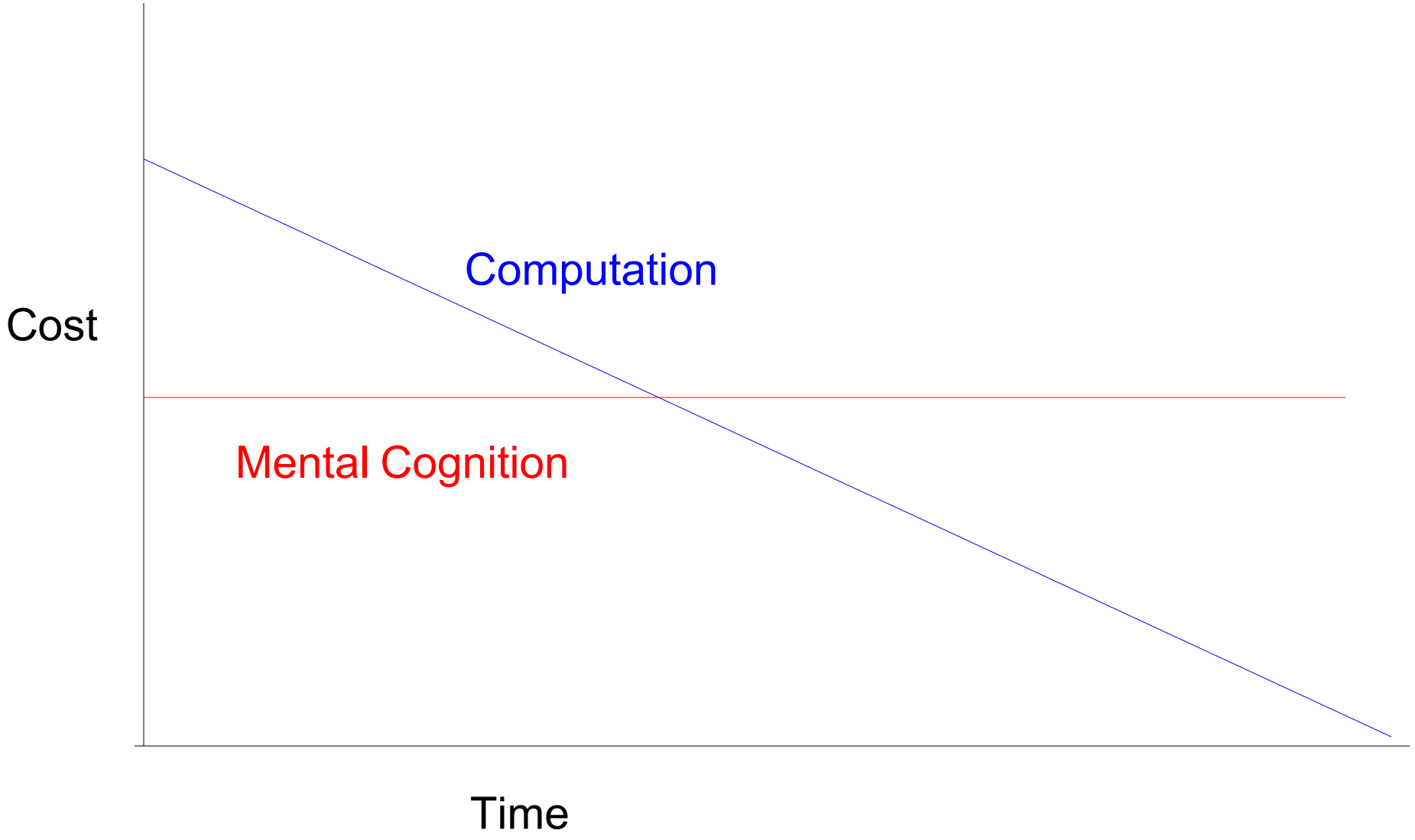
- Engineers have traditionally focused on computational transaction costs
  - E.g. speed of a digital signature
- Mental transaction costs are more important
  - Measuring value – evaluating whether something is worth it to buy
  - Sets a floor on granularity of payments practical
    - Szabo, “Micropayments and Mental Transaction Costs”, Berlin Internet Economics Workshop 1999
- Also a problem for automated markets
  - Valuation comes from preferences
  - How do customer preferences get into the computer?
  -

4 Kinds of Contractual Relationships

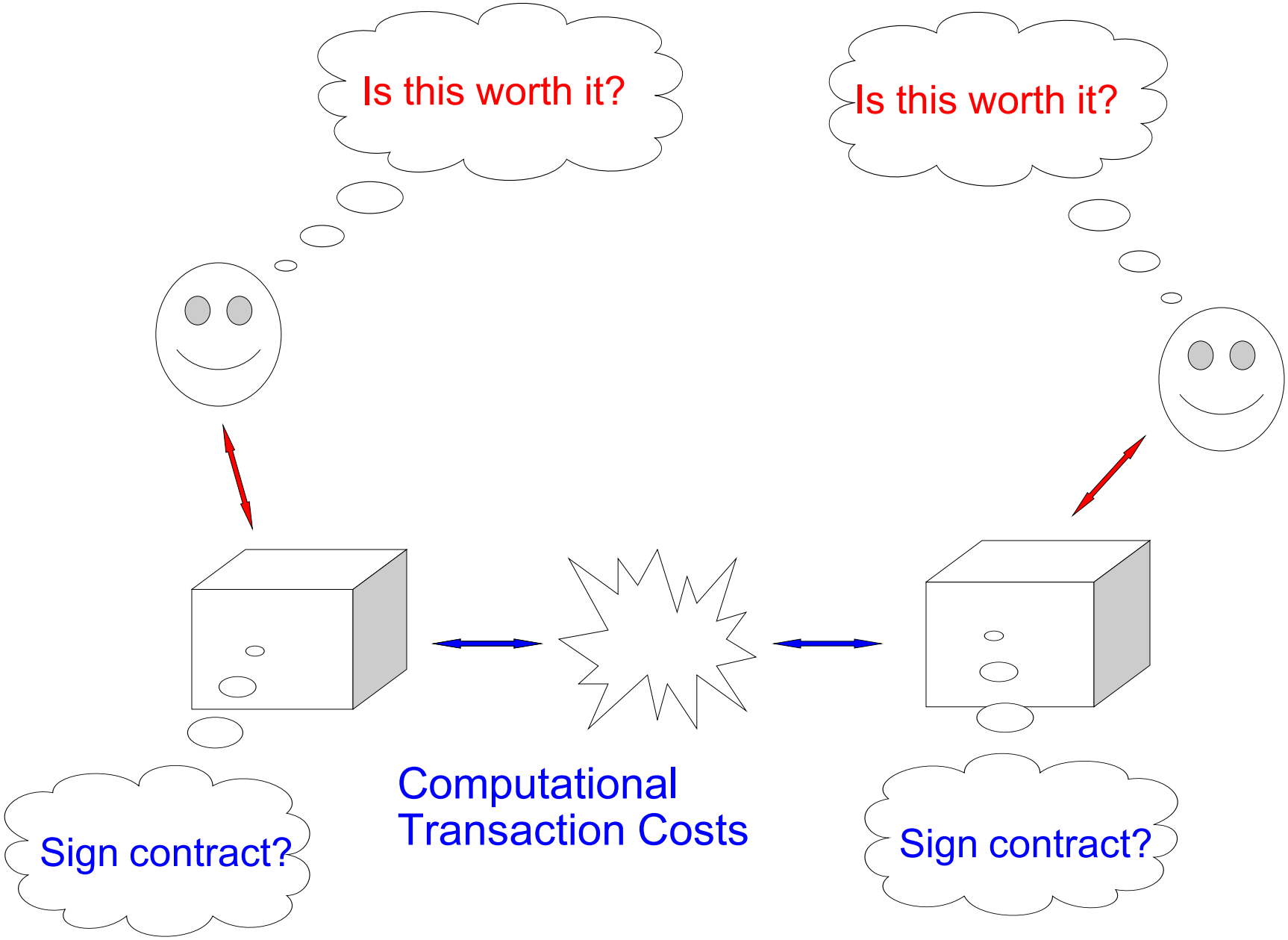
- Caveat Emptor – buyer can measure value of goods to him ex-ante (before purchasing)
- Long-Term Relationship – buyer can measure value of goods to him ex-post (after using it) and cost of repeat purchases is low

- State-enforced Contract – warrantee – buyer and third party (court) can measure value of goods ex-post but cost of repeating the purchase is too high
- Firm Integration / Employment – none of the above hold, and transaction becomes work of an employee under supervision

Mental Transaction Cotss



Mental Transaction Costs



Mental Transaction Costs

Limits to Micropayments

- “The Mental Accounting Barrier to Micropayments”, 1996
- “Micropayments and Mental Transaction Costs”, 1999 (Berlin Internet Economics Conference)
- Practical lower limit to payment size is set by mental transaction costs unless shopping preferences can be represented in the computer
  - More generally, this is a limit to contract complexity

Mental Transaction Costs

Potential Solutions

- Gas gauges
  - Useful for fungible commodities
  - Can information services be bundled into fungible units?
- Market translator (MT)
  - Automatically obtain market prices
    - Primitive example – ATM machine that does automatic currency translation
  - Input partial preferences from user
  - Translate “source” into “target” contract using market prices and partial preferences
  - Enables automated negotiations

Automated Negotiations

The Market Translator

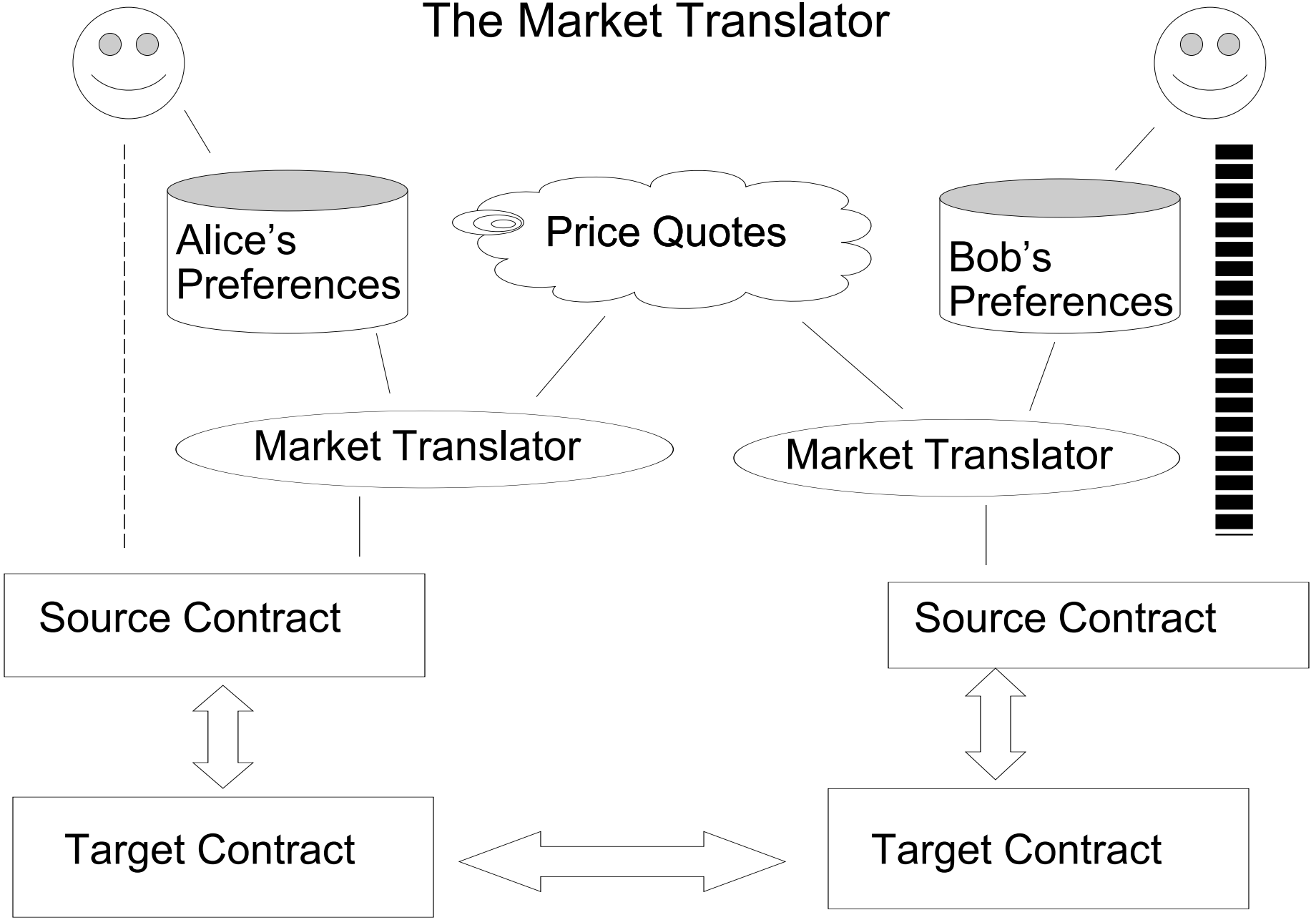
- Goal
  - Preferences
    - Simple, or
    - Automatically generated from other user input
  - Arbitrarily sophisticated contracts

Automated Negotiations

- Market translator – complex example
  - Alice and Bob
  - Two kinds of information needed:
    - Alice's preferences partially expressed in her budget
    - Market prices come from online listings
  - Source contract expressed in personal utility units (PAUs)
    - MT generates candidate source contracts for approval
  - MT translates into a target contract which is sent to Bob
  - Bob's MT translates back to a source contract expressed in Bob PAUs
  - Alice and Bob use MT to negotiate until they agree, at which point the target contract is executed as a smart contract

Automated Negotiations

The Market Translator



Formal Contract Languages

- Too many examples to list
  - Obligations, rights, schedules, composition, etc.
- My own entry – SEAL
  - Event-driven
  - State machine – active and inactive clauses
  - Based on “reverse engineering” actual contracts

Formal Language for Agreement Terms & Legal Rules

Futures contract

```
future(rightA="1 round lot pork bellies",
    rightB="$1,500.00",
    p = "for delivery in July 2002") =

    when withinPeriod(p)
        to Holder rightA   with   to Counterparty rightB
    then terminate
```

Formal Language for Agreement Terms and Legal Rules

Option Contract / Multi-threaded State Machine

```
callOptionAmerican (rightA="1 round lot XYZ Corp.",
   rightB="$2,000/lot",
   time="end of trading on last trading day of August") =

    when beforeTime(time)
          when choiceOf(Holder)
             to Holder rightA with to Counterparty rightB
    when afterTime(time)
          terminate
```

Formal Language for Agreement Terms and Legal Rules

Option Contract / Multi-Threaded State Machine

```
callOptionAmerican (rightA="1 round lot XYZ Corp.",
   rightB="$2,000/lot",
   time="end of trading on last trading day of August") =

    when beforeTime(time)
          when choiceOf(Holder)
              to Holder rightA with to Counterparty rightB
    when afterTime(time)
          terminate
```

Formal Language for Agreement Terms and Legal Rules

Bond contract / Schedule of payments

```
bond(coupon, principal, schedule) =

    for schedule
        when withinPeriod(schedule.next)
            to Holder coupon
    then
    when withinPeriod(schedule.next)
        to Holder principal
```

Formal Language for Agreement Terms and Legal Rules

Insurance contract (Genoa, late 14th century)

```
insureGoods(goodsPremium, principal, penalty, t1, t2, goodsInsured) =
    counterpartySecurity =    pledge(allGoods(Counterparty))
    with to Counterparty getTitle(goodsPremium)
    insurancePayment(goodsInsured, principal, t1, t2)
    with when breachedPerformance(insurancePayment)
        to Holder foreclose(counterpartySecurity, penalty)

insurancePayment(goodsInsured, principal, t1, t2) =
    when safeArrival(goodsInsured) terminate
    when withinPeriod(t1,t2)
        when choiceOf(Holder)
            to Holder principal
```

Formal Language for Agreement Terms and Legal Rules

Property Deed

feeSimpleSubjectToExecutoryLimitation
(Property, Grantee, Condition, Remainderman) =

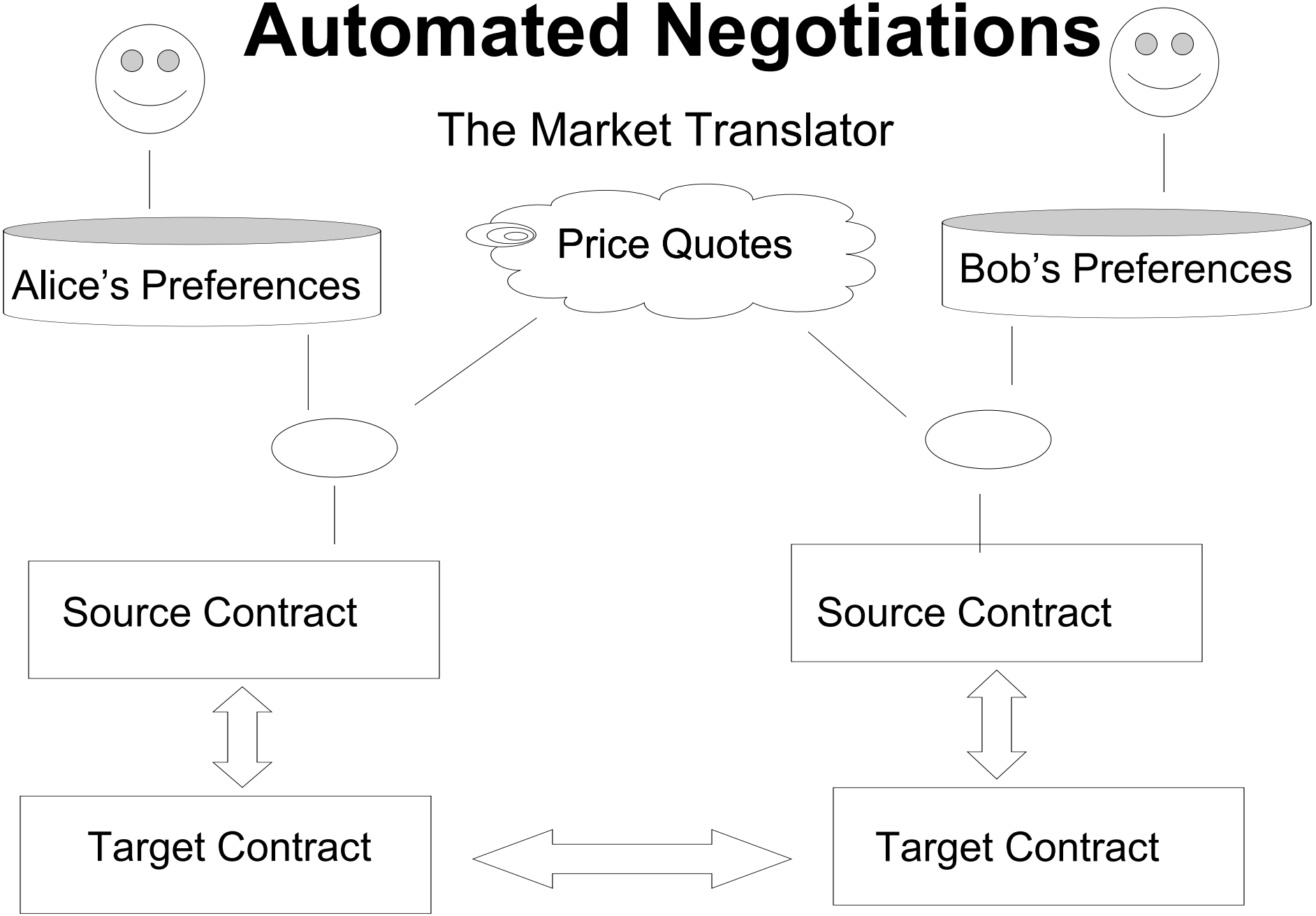
to Grantee Property
then when Condtion(Property)
to Remainderman Property

Formal Language for Agreement Terms and Legal Rules

Vending Machine

```
sellCandy(candyPrice = $0.90) =
   variable moneyAmount = $0.00
   then
      # coins also fall into a temporary till tempTill
   when choiceOf(Counterparty, nickel)
      to TempTill nickel
      then to Counterparty add(moneyAmount, $0.05)
      then to Counterparty display(moneyAmount)
   when choiceOf(Counterparty, dime)
      to TempTill dime
      then to Counterparty add(moneyAmount, $0.10)
      then to Counterparty display(moneyAmount)
   when choiceOf(Counterparty, quarter)
      to TempTill quarter
      then to Counterparty add(moneyAmount, $0.25)
      then to Counterparty display(moneyAmount)
   when choiceOf(Counterparty, moneyReturn)
      to Counterparty dropCoins(tempTill, returnTill)
      with moneyAmount = $0.00
      then to Counterparty display(moneyAmount)
   when threshold(moneyAmount, candyPrice)
      to Holder (nickel | dime | quarter)
         to CounterParty redirectNewCoinsTo(returnTill)
      also display("ready to dispense --please select candy")
      then when (candySelection)
         to Counterparty dropCandy(candyRacks, candySelection)
         with to PermanentTill dropCoins(TempTill)
         with moneyAmount = $0.00
   continue
```

Automated Negotiations



Scarce Objects

- Security between parties in privity
  - Access control and Usage Control
- Access control
  - ACLs vs. Capabilities
  - E language
    – Capabilities as object references => automatic POLA
- Usage control
  - Scarce objects
  - Wrap objects up in a layer that requires payment of a ticket, and restricts holder of that ticket to a specific number of invocations
  - Need a special-purpose market translator to obtain and spend tickets for performing computations w/minimal input from user
  - Don’t need full-fledged contract language

Overview of Talk

- Historical Importance of Transaction-Enabling Technology
  - Technology working alongside agreements and legal rules to facilitate transactions
  - Proto-money
  - Tamper-Evidence
  - Clocks and Work Schedules
- Smart Contracts
  - Increasingly, contractual logic itself will be embedded in technology
  - Security
    - Mapping between contractual terms and security protocol
    - Self-enforcing vs. evidence gathering
    - Privacy and integrity of multiparty protocols
  - Mental Transaction Costs
    - Value Measurement / Preferences (”what’s it worth to me?”)
    - Automated Negotiations, Contract Language

Smart Contracts

Nicholas J. Szabo, The George Washington University Law School

http://szabo.best.vwh.net/

nszabo@law.gwu.edu