Sidechain Governance – Why Involve the Miners?

Paul Sztorc

May 2016

Motivation

com/r/Bitcoin/comments/3u3pv0/psztorc_reveals_drivechain_a_bitcoin_sidechains/

bitcoin

comments | other discussions (6) | Want to join

229

psztorc reveals 'Drivechain', a Bitcoin sidechains 2-way-peg proposal, with security analysis & FAQ -- ["With sidechains: altcoins are obsolete, Bitcoin smart contracts are possible, Bitcoin Core & XT can co-exist, and all hard forks can become soft forks. Cool upgrades to Bitcoin are on the way!"] (truthcoin.info)

submitted 4 months ago by eragmus

118 comments share

search

this post was subm
229 points (92%
shortlink: https://re

[–] [deleted] 23 points 4 months ago

paging u/adam3us u/petertodd u/nullc

permalink save report reply

Yes

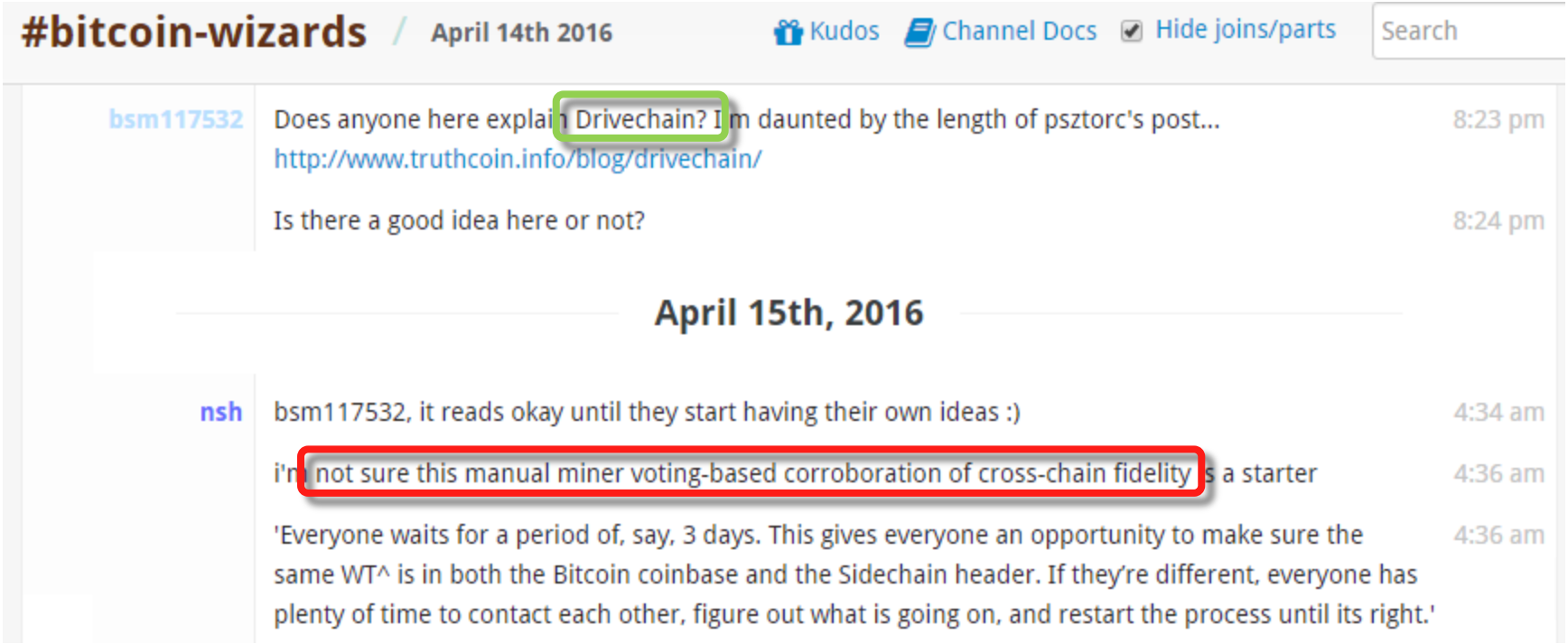
[–] nullc 15 points 4 months ago*

Whats to ping about? Interesting write-up and a fair bit to absorb.

My understanding is that psztorc is assuming a more tightly coupled model where (some) miners are *required* to verify the sidechain.

In the pegged sidechains whitepaper we specifically note (e.g. line 370) that this is possible for miners to also verify the sidechain and that it if it is done can can additive boost the security; but if it is done so completely that the miners would orphan Bitcoin blocks if they were making an invalid sidechain transfer that this would undermine the weak coupling we intended as a primary goal (see section 4.4 Risk of soft-fork). Tight coupling is particularly at odds with the concerns raised in section 4.3. I think Paul is suggesting binding somewhere in

Motivation

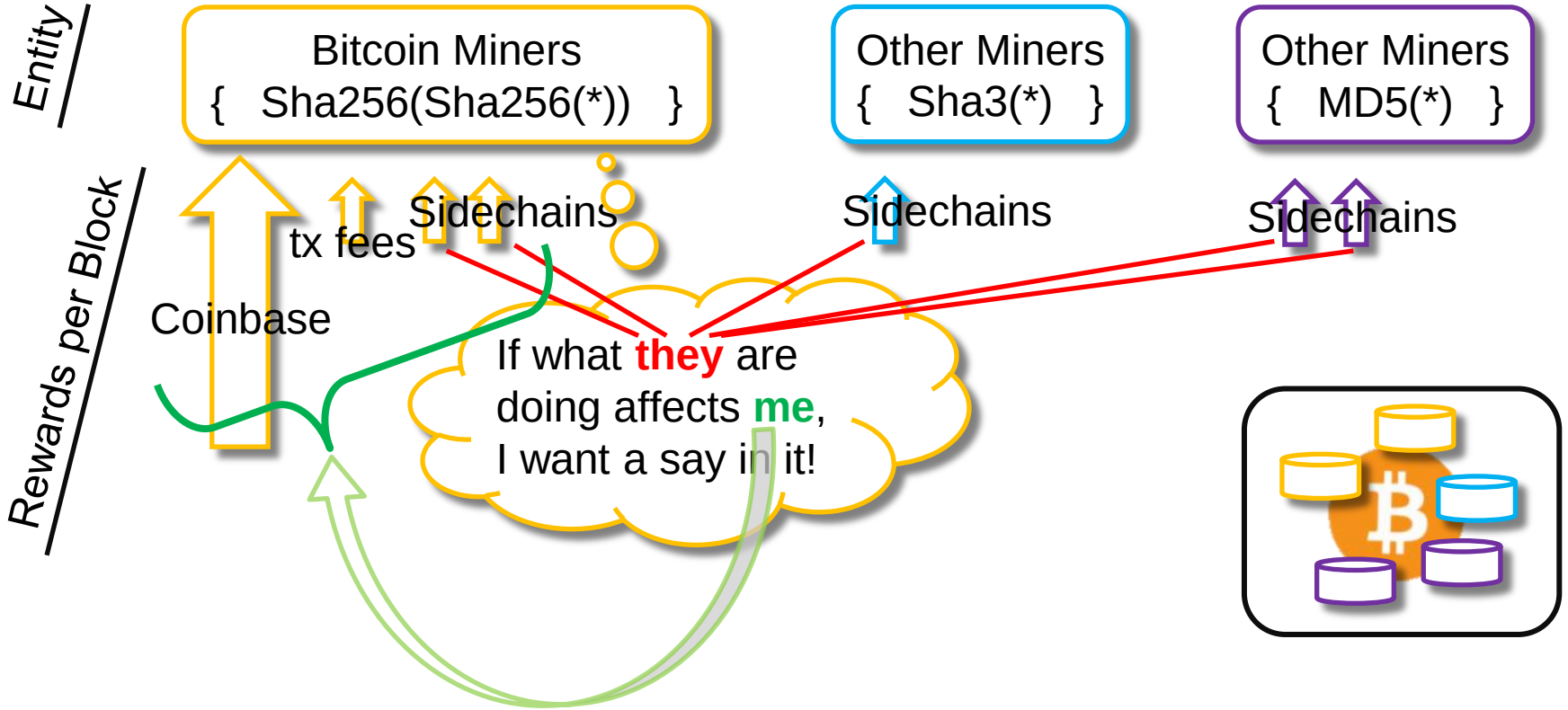

#bitcoin-wizards
April 14th 2016
Kudos
Channel Docs
Hide joins/parts
Search
bsm117532
Does anyone here explain Drivechain? I'm daunted by the length of psztorc's post...
http://www.truthcoin.info/blog/drivechain/
8:23 pm
Is there a good idea here or not?
8:24 pm
April 15th, 2016
nsh
bsm117532, it reads okay until they start having their own ideas :)
4:34 am
i'm not sure this manual miner voting-based corroboration of cross-chain fidelity is a starter
4:36 am
'Everyone waits for a period of, say, 3 days. This gives everyone an opportunity to make sure the same WT^ is in both the Bitcoin coinbase and the Sidechain header. If they're different, everyone has plenty of time to contact each other, figure out what is going on, and restart the process until its right.'
4:36 am

Motivation

People do not want
the miners to have control
over the sidechains...

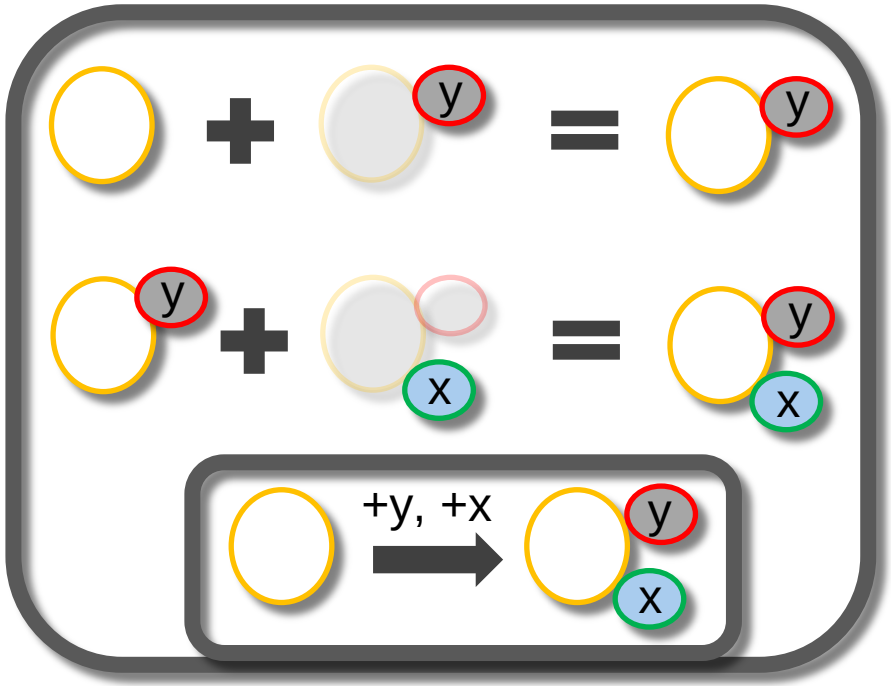
...but I do...

In One Slide – Contract Externalities



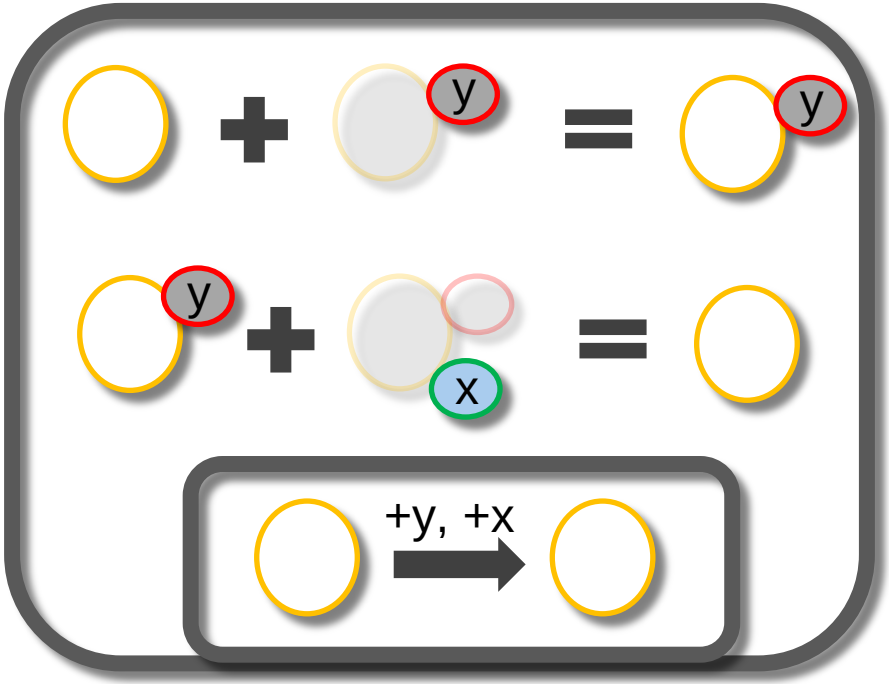
Problem

Expectation: Additive



Two new functionalities always *add* to each other.

Reality: Ecological

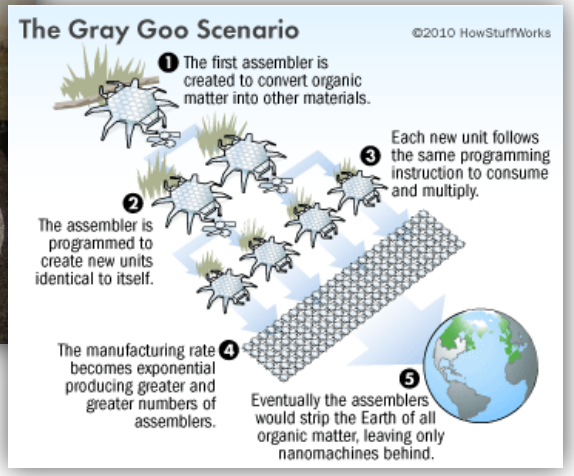


Two new functionalities potentially *subtract* from each other.

Metaphors for the Problem

Invasive Species

Grey Goo



Spam

“Censorship is Expression”
-- 1984 esque, but correct (b/c finite shared resources)

Restated – What we want = SCs

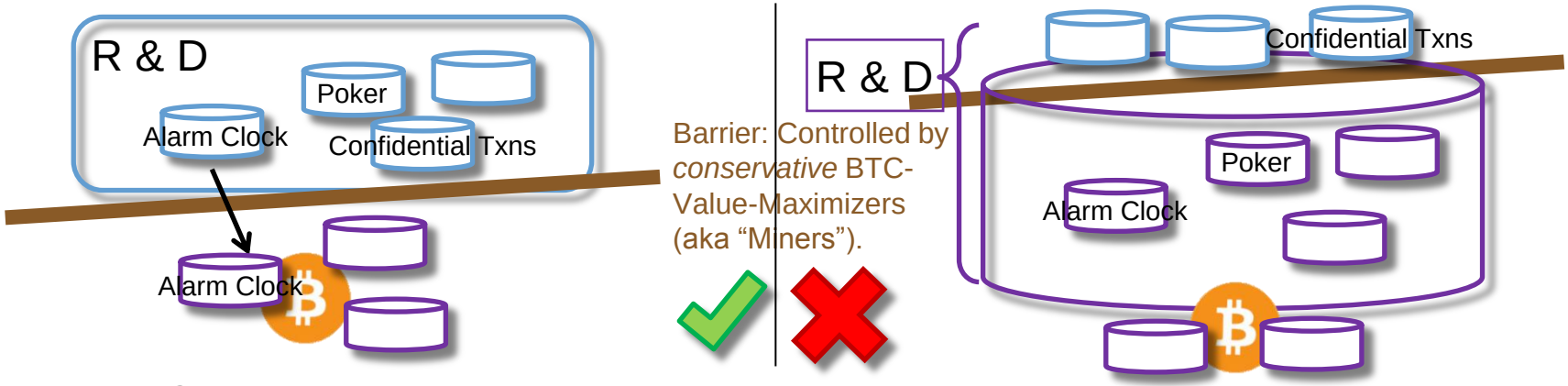
- **Obvious:** A smart contract enforces itself ... It does not require a 3rd party's permission.
- **Not Obvious:** This "permission" can be *negative* as well as *positive.*
  - *Positive* – "that someone approve".
  - *Negative* – "that no one disapprove".
  - (Smart Contracts attacking each other).

Turing Complete

Restated Again

- "Non-trivial smart contracts can never be Permissionless."
  - Permissionless ***Innovation*** ✓
  - Permissionless ***Implementation*** ✗

Sidechains, alt token systems, any new BTC-payment-mapping, or a system which implements those mappings ...



- Turing-Completeness can't be allowed (enables permissionless implementation).

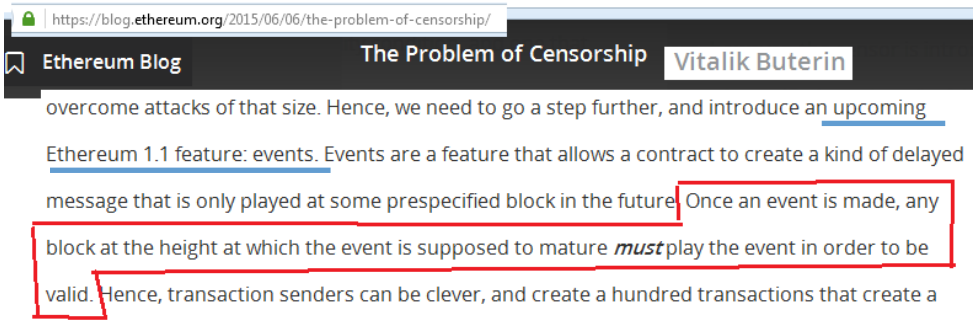
Why am I worried?

bloq

1. Two Examples of “Cannibalism” (SCs *Harming and Obviating* each other)
    1. PI Disables the (much much cooler) “Oracle” Contracts.
    2. Use PI (TC) to *steal* Bitcoin, while disabling TC!
2. Theory -- Why Blockchain “Permissionless Implementation” isn’t good, anyway.
    1. Costs and Benefits of General SC.
    2. Ethereum Misunderstands the Trust Problem (Solved by Brands / Blockchains) – TC without Ethereum.
3. Bitcoin = Game-Theory, not CS (and why that matters for permissionless-ness).

P. Impl. Harm - Assumptions

1. Any SC can get in, at least at first -- (the reverse = this talk's thesis).
    1. If miners attempt to censor, they face: obfuscation / multiple attempts / assembly-by-parts.
    2. Otherwise...not really censorship-resistant? (...not really TC?)


https://blog.ethereum.org/2015/06/06/the-problem-of-censorship/

Ethereum Blog

The Problem of Censorship

Vitalik Buterin

overcome attacks of that size. Hence, we need to go a step further, and introduce an upcoming Ethereum 1.1 feature: events. Events are a feature that allows a contract to create a kind of delayed message that is only played at some prespecified block in the future. Once an event is made, any block at the height at which the event is supposed to mature *must* play the event in order to be valid. Hence, transaction senders can be clever, and create a hundred transactions that create a

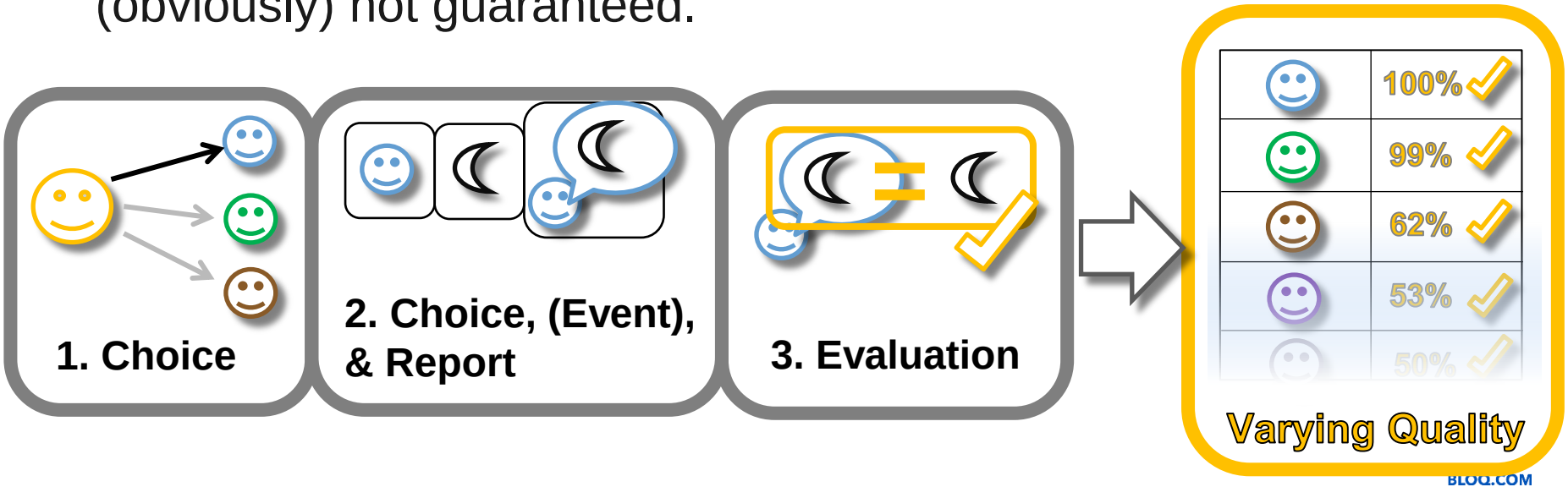
2. SC's allowed to be at-or-near the complexity of Bitcoin.

Ex 1 – Unsustainable Oracles

- P.I. Exposes a blockchain system to a ***"Reputation Free-Rider Problem"***
- Trivial Case: if Oracle is not going to control anything valuable, then no compulsion to lie, no need for trust, no need for blockchain.
- Important Case: otherwise, the Oracle is going to incur an *opportunity cost of theft* – "trust" is required.

Ex 1 – Oracle Basics

- Ultimately, oracles ***need*** to vary in quality (because we must choose them pre-report, and evaluate them post-report).
- We necessarily ‘trust’ them, mid-event. Performance is (obviously) not guaranteed.



Ex 1 – Reputation Free-Rider Problem

bloq

Varying Quality

	100% ✓
	99% ✓
	62% ✓
	53% ✓

Quality Premium

Labor

Setup

Oracle Fee (Paid Upfront)

Recall, honesty is *costly* to Oracle...Oracle is forgoing theft-opportunities.

I will copy, when he reports.

I'm always cheaper...

Quality Premium

Labor

Setup

Oracle Fee

Info on blockchain, now a public, non-excludable resource.

f( )

...and I'm always *exactly* as reliable.

+ Result: "crypto-reputation" is impossible (all always 50% ✓) . No different from trusting website.
+ **Other *impossible* things: all DACs, identity, fidelity bonds, financial markets.**
+ In contrast, a single 'mega-contract' can (with entrants excluded) "coordinate" payment-events and oracle-quality events. It can *force* a mapping from quality to $.

Quality varies, payments don't co-vary!

Can't buy quality!

OUT OF BUSINESS

gavintech.blogspot.com/2014/06/bit-thereum.html

The answer is "yes," if we're willing to replace "verified by the entire network" with "verified by a set of semi-trusted 'oracles'."

That's cheating, though, isn't it? We're not entirely decentralized if we are trusting eleven contract-verifying-services not to collude (or all get hacked) to violate conditions encoded in some contract(s).

It **is** cheating a bit... b[illegible] all [illegible]f the really interesting complex contracts I can think of require data from outside the blockchain. Like the BTC/[illegible] exchange rate on some future date (for blockchain-enforced futures contracts).

ethereum doesn't have some magic solution to the "data outside the blockchain" issue: as their whitepaper says, "a trusted source is still needed to provide the price ticker." And there is already at least one startup working on a

Ex 2 – Stealing BTC Without the Key

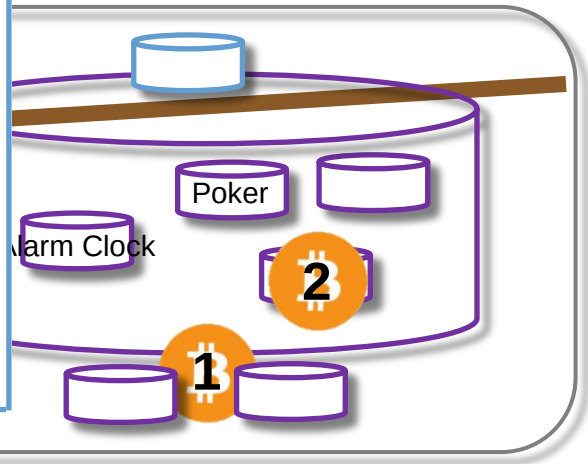
Ex 1: Basic, Inevitable
Ex 2: Contrived, Unlikely

Claim: Steal BTC + Disable TC

- Execution? Force miners to steal 1% of the outstanding Bitcoins (ie, 210,000...some individuals will lose *all* their BTC).
- Strategy? Create a “near copy” of Bitcoin, which frees up 1% of the BTC. This 1% can be claimed by miners, if they disable the original Bitcoin (and everything attached to it).

Tools

1. **“Observation”**
   - It is possible to watch Bitcoin-1 *from* Bitcoin-2.
   - Events in B2 can be made to depend on events in B1.
   - Possible to ~instantly move BTC from B1 to B2.



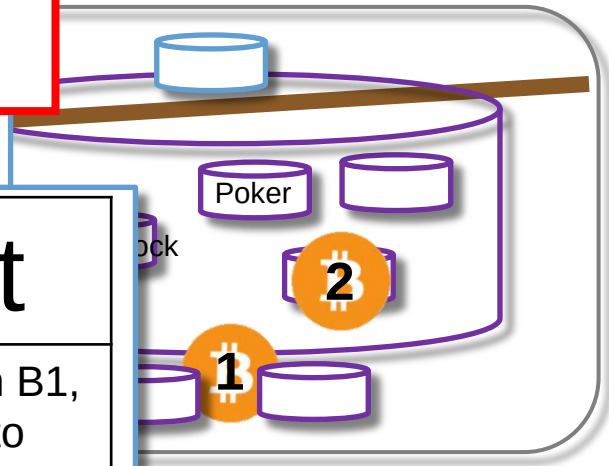
2. **“Half-Surrender”** (Voluntary / Recyclable 2wp)
   - The Rules: every 2 months, there’s one special block (in B2) where individuals can use their B1-keys to ‘mint’ B2-BTC. These minted coins can move freely throughout B2, as long as their parent coins have not moved *twice*.
   - After 99% of the B1-BTC have been H-surrendered, this stops working.

Tools

Dominant Strategy: “Half-Surrender” all BTC you own, at every opportunity.

1. “C

- It is possible to watch Bitcoin-1 *from* Bitcoin-2.
- Event
  in B1.
- Possi



B2 Won	B2 Lost
Burn the coins on B1, by sending them to a provably-unspendable address. Now, other people will accept your B2 coins.	Reclaim the coins on B1, by sending them to yourself twice. (Or, doing nothing.)

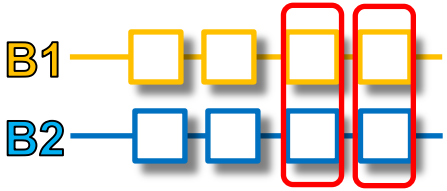
2. **“Half-Su**

- The R (in B2) where
  indivi ese minted coins
  can m t coins have not
  moved *twice*.
- After 99% of the B1-BTC have been H-surrendered, this stops working.

Tools (targeting miners)

3. Forced Dilemma

- After a certain network time is reached, B2 needs 1 of 2:
  - B2 must be empty (ie, B2 is choosing never to update).
  - Nearest B1 block is complying with ‘arbitrary soft fork S’.
- Thus, B2 can “ask” B1 to perform any soft fork.

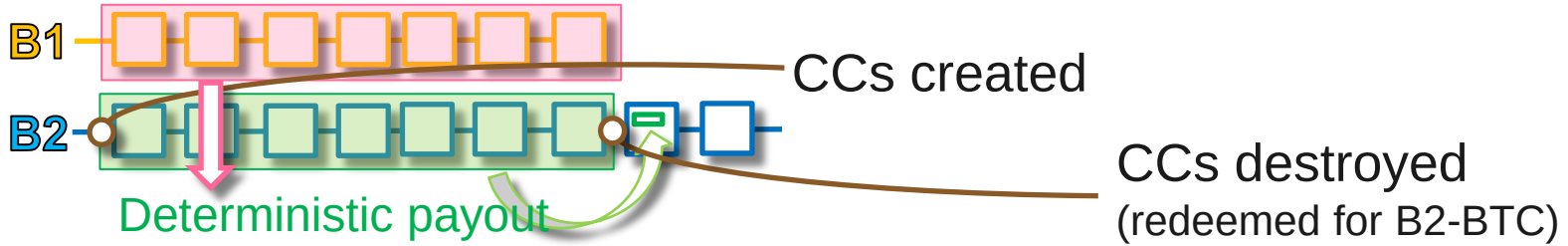


4. Endgame Payout

- Pays X coins (on B2) to Y recipients, conditional on some future block being reached.
- Choosing X and Y?

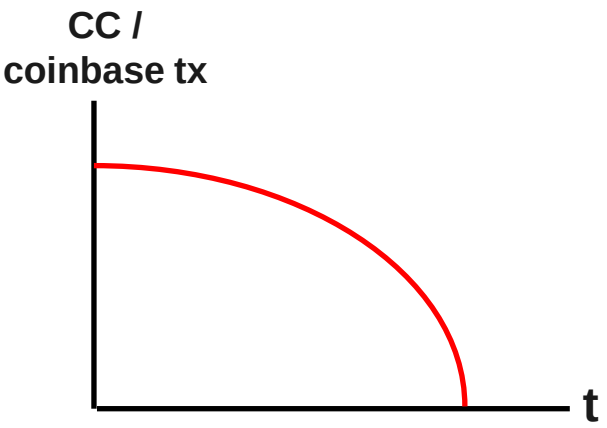
X&Y to Entice Miners

- X (Coin Payout) = Easy
  - Large enough to be enticing, but small enough to make victims ignorable.
  - *...1% of the currently outstanding BTC*
- Y (Recipients) = More Complex
  - Who do we still need to bribe? The miners.
  - I propose a way to recruit miners which [1] **rewards early rule-compliance** [2] is **ambiguous (contains plausible deniability)**.
  - Create temporary 2nd coin type: “compliance credits”.



More Detail re: Two Factors

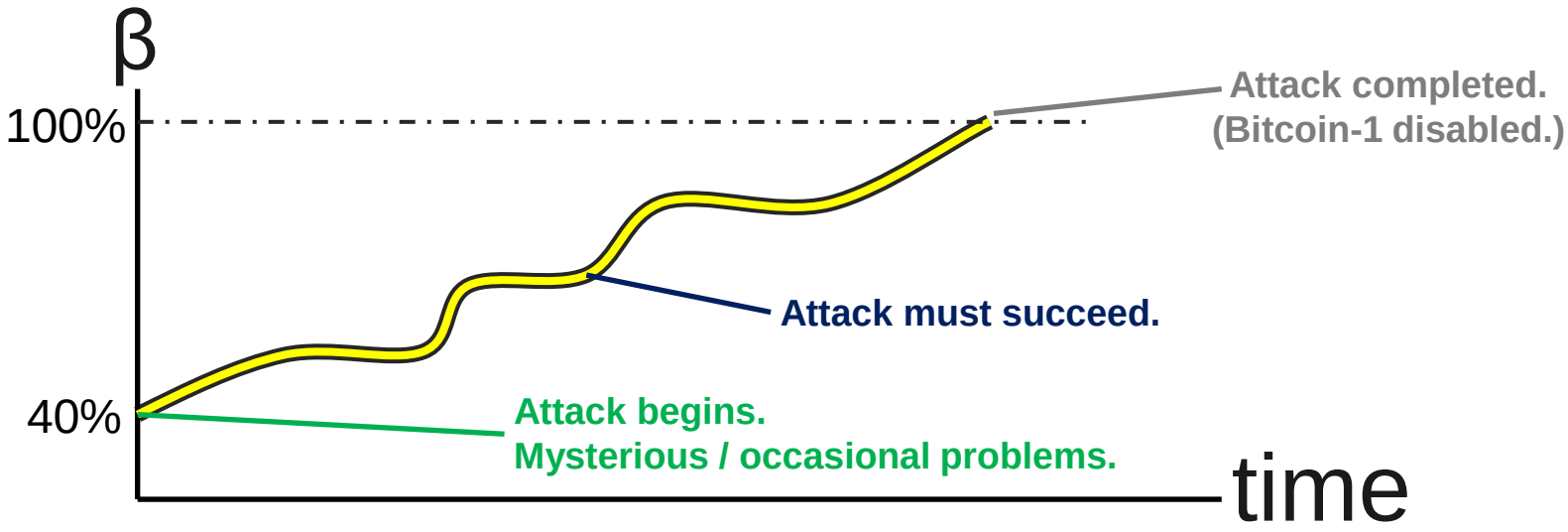
- CCs (on B2) are awarded to B1 miners (identified by coinbase transaction).
- Issuance schedule favors “early adoption”.



- To achieve ambiguity:
  - For each B1 block, use (Attack Seed +) PrevBlock hash to (deterministically / pseudo-randomly) “sort” the B1-UTXOs.
  - The “top” β% are designated “frozen”. If anything is spent from them, the B2 chain does *not* give miners their Compliance Credits!
  - Miners have plausible deniability: “did not get tx”, “insufficient fee”.

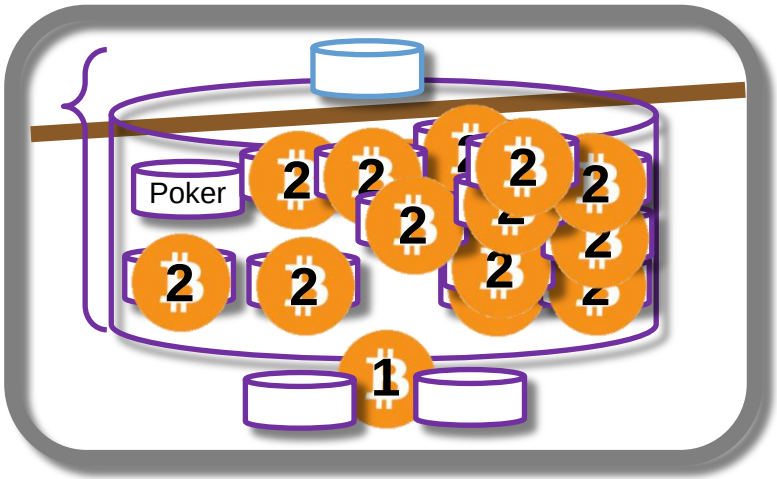
Compliance Credits (CCs)

- Ideally, our signal would be tunably ambiguous:
  - At **first**, the signal is very ambiguous. **Later**, the signal is allowed to "lose" its ambiguity.
  - This is because: any *identifiable* miners who are ***purposefully*** ***malicious*** are likely to suffer retribution.



Dominant Strategy for Miners

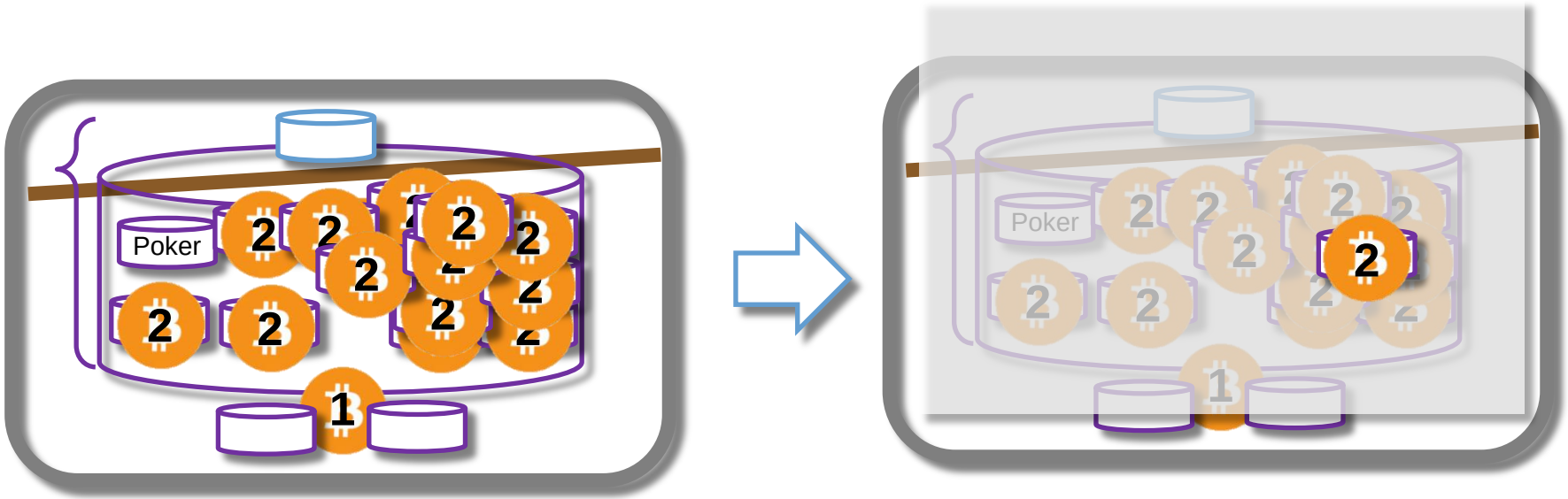
- Create *many* “B2”s (and seeds).
- Initially: accrue CC’s passively.



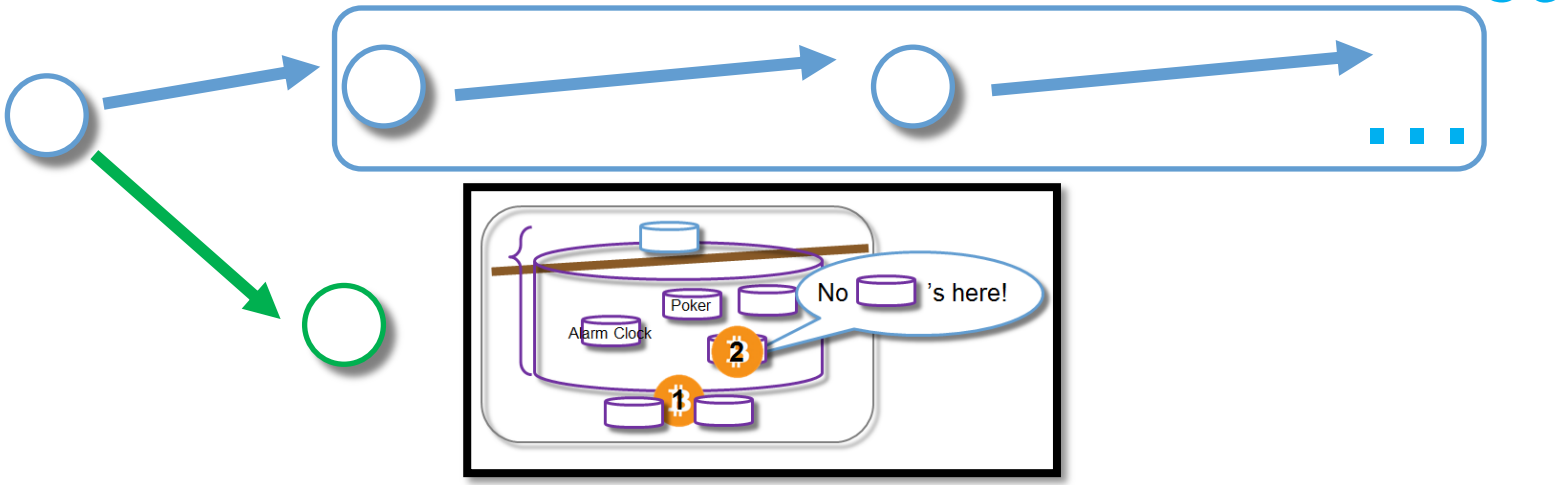
- BTC txns provide entropy.
- New gravitational centers will emerge and attract miners.
  - These miners now have a vested interest in the attack.
  - If slow to join, the deck might shuffle against them.
  - Miners may recruit a 51% group with side-payments.

Dominant Strategy for Miners

- Create *many* “B2”s (and seeds).
- Initially: accrue CC’s passively.
- BTC txnx provide entropy.



TC / PI is Automatically Removed



- By **leaving the attack open to repeat**, agents will have an incentive to disable the “repeat-enabler”.

- Consider the *removal* of Turing-Completeness – it [1] has benefits (stability, “no more attack contracts”), and [2] can only be done once (can’t remove something which doesn’t exist).

Part II – Cost/Benefit

What are we throwing away if we lose Permissionless Implementation?

PI – Costs and Benefits

- Costs
  - Bad Smart Contracts “Anarchy” (Unreliable Environment)
  - Uncertainty / Open-Endedness / Instability
- Benefits
  - Immune to censorship *from miners*.
  - If **many applications** need to be created/added **quickly**, or on an ongoing basis, then we benefit from faster onboarding.

SC Applications

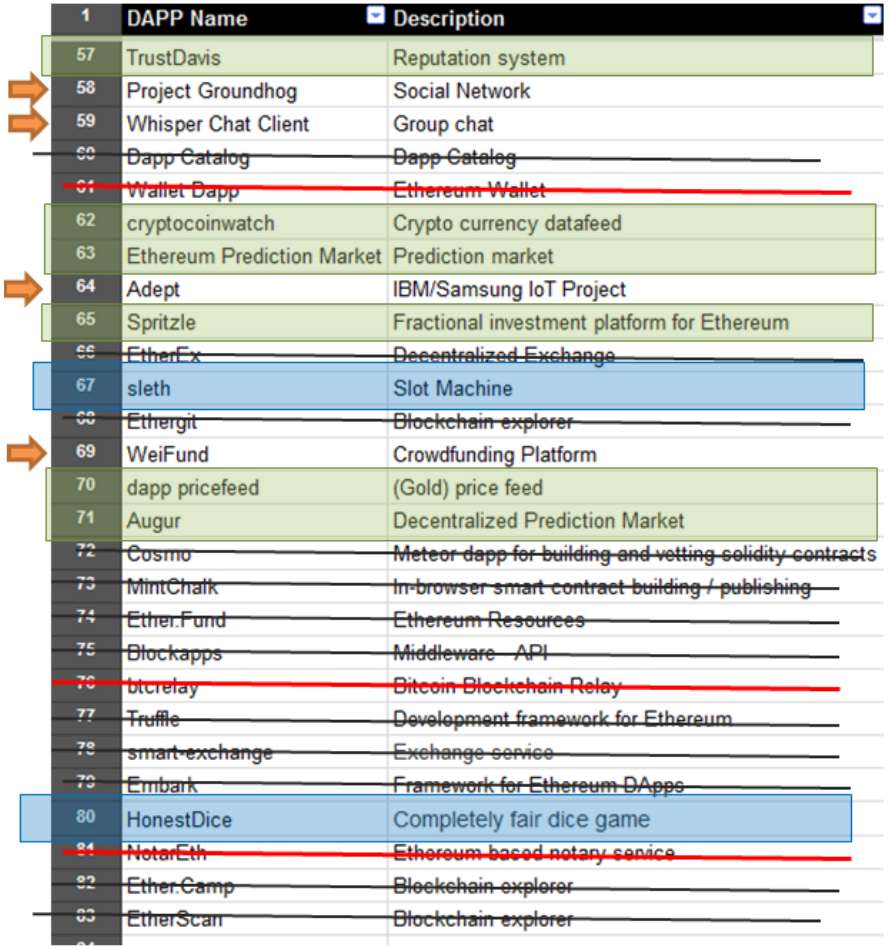
- Aug 2015
- At “Demo” level, or higher.
- Provided by Ethereum Team.

~~Intermediate~~

~~In Bitcoin Already~~

Oracle (flawed)

Casino



1	DAPP Name	Description
57	TrustDavis	Reputation system
58	Project Groundhog	Social Network
59	Whisper Chat Client	Group chat
60	~~Dapp Catalog~~	~~Dapp Catalog~~
61	~~Wallet Dapp~~	~~Ethereum Wallet~~
62	cryptocoinwatch	Crypto currency datafeed
63	Ethereum Prediction Market	Prediction market
64	Adept	IBM/Samsung IoT Project
65	Spritzle	Fractional investment platform for Ethereum
66	~~EtherEx~~	~~Decentralized Exchange~~
67	sleth	Slot Machine
68	~~Ethergit~~	~~Blockchain explorer~~
69	WeiFund	Crowdfunding Platform
70	dapp pricefeed	(Gold) price feed
71	Augur	Decentralized Prediction Market
72	~~Cosmo~~	~~Meteor dapp for building and vetting solidity contracts~~
73	~~MintChalk~~	~~In-browser smart contract building / publishing~~
74	~~Ether.Fund~~	~~Ethereum Resources~~
75	~~Blockapps~~	~~Middleware - API~~
76	~~btcrelay~~	~~Bitcoin Blockchain Relay~~
77	~~Truffle~~	~~Development framework for Ethereum~~
78	~~smart-exchange~~	~~Exchange service~~
79	~~Embark~~	~~Framework for Ethereum DApps~~
80	HonestDice	Completely fair dice game
81	~~NotarEth~~	~~Ethereum based notary service~~
82	~~Ether.Camp~~	~~Blockchain explorer~~
83	~~EtherScan~~	~~Blockchain explorer~~

dapps.ethercasts.com

Multi-Sig Wallet with 2FA and mist support

Working Prototype 2016-03-24

ThanksCoin

ThanksCoin

Reputation Ranking and monetary reward of internet users

MIT

Working Prototype 2016-04-02

HonestDice

etherapps.info

Completely fair dice game

Live 2015-08-12

NotarEth

Maran Hidskes

Ethereum based notary service

MIT

Live 2015-08-13

Ether.Camp

Roman Mandeleil

Blockchain explorer

Live 2015-08-13

EtherScan

Matt Tan

Blockchain explorer

Live 2015-08-14

Embark

Iuri Matias

Framework for Ethereum DApps

MIT

Live 2015-08-15

EtherListen

Kobi Gurkan

Realtime Ethereum transaction visualizer.

MIT

Live 2015-08-24

Universal ĐApp

d11e9

A Universal Interface for contracts on the Ethereum blockchain

MIT

Live 2015-09-03

CryptoRPS

CryptoRPS

Rock-Paper-Scissor game with a twist

Live 2015-09-24

Ethereum Alarm Clock

Piper Merriam

Schedule contract calls

MIT

Live 2015-09-24

Grove

Piper Merriam

Fast, efficient, queryable storage for ethereum contracts

MIT

Live 2015-10-07

Etheria

fivedogit

The first-ever decentralized virtual world

GPL

Live 2015-11-07

Oraclize

Thomas Bertani

Provable honest oracle service

Live 2015-11-10

Ethereum Pyramid

ethererik

Ethereum Pyramid Contract

Live 2015-11-13

EtherDice

vnovak

Provably fair and escrowed gambling

Live 2015-11-15

Browser-Solidity

chriseth & d11e9

Browser based solidity contract compiler & runtime

MIT

Live 2015-11-20

EtherID

Alexandre Navarniouk

Ethereum Name Registrar

Apache

Live 2015-11-30

Dapp Store

Tim Coulter

Marketplace for Dapps

Live 2015-12-11

Dapple

Nexus Dev

smart contract package manager and build tool

MIT

Live 2016-01-02

EthHypeDns

slothbag

Resolve Hyperboria/CJDNS ipv6 addresses via etherid.org contact

proprietary

Live 2016-01-05

Icebox

Christian Lundkvist

A cold storage solution for Ether

MIT

Live 2016-02-11

EtherDoubler

Satoshi :)

The first doubler with verified contract

Live 2016-02-17

EtherWall

Aleš Katona

GUI desktop wallet for Ethereum

GPLv3

Live 2016-02-18

EthereumWall

LPMitchell

Decentralized unmoderated public message board

Live 2016-02-25

Etheroll

James Britt

Ether dice game casino / gamble ether

proprietary

Live 2016-03-02

GroupGnosis

ConsenSys / Martin Köppelmann & Stefan George

Prediction market

Live 2016-03-10

GovernMental

g0vernmental

Educational Ponzi Scheme

proprietary

Live 2016-03-11

Blockapps

Consensys / Kieren James-Lubin

Middleware - API

Live 2016-03-16

Ethereum x 1.8

Diana

Multiplier ethereum 1.8x Payouts

Live 2016-03-16

BlockApps

Consensys / Kieren James-Lubin

Scalable Enterprise Blockchain Platform

Live 2016-03-16

Ether Wheel

doppio

A simple Ethereum lottery with a user-friendly interface

MIT

Live 2016-03-17

Protect The Castle

MilkyWayz

Ponzi Game: Protect The Castle - x2 + Jackpot

Live 2016-03-21

Etheropt

Etherboost

Decentralized Ethereum options exchange

Live 2016-03-23

Proof of Physical Address

ConsenSys

Smart oracle that serves as a primitive form of Know-Your-Customer

Live 2016-03-25

Dynamic Payout Pyramid

Daedalus

A pyramid with payouts and fees that alter for prompt payout

Live 2016-03-29

THE GREED PIT

Katatsuki

Fast pyramid-inspired game with random and strategic elements

GPL

Live 2016-03-29

Rubix

Rubix by Deloitte / Jinius Tu

Enterprise Blockchain Platform

Live 2016-03-29

LittleCactus

Tdecha

Faster Pyramid (lower investment, lower payout)

Live 2016-04-02

Ethereum Jackpot

ETH Jackpot

If 1m people play with just $1 we can make one a millionaire

Live 2016-04-08

ESports Bets

Masaca

Decentralized ESports betting platform

Live 2016-04-10

EthStick

Katatsuki

A satirical ponzi game w/ Ranking, designed to minimize risk

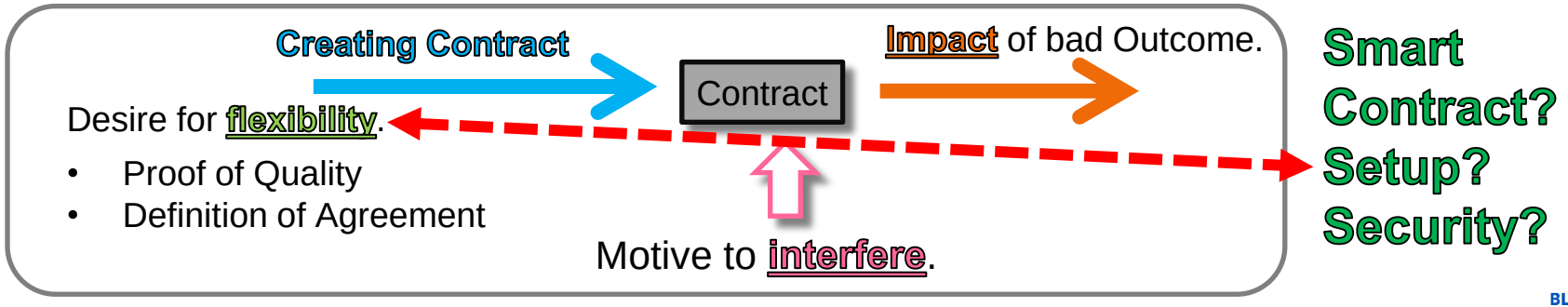
Live 2016-04-13


dapps.ethercasts.com
ThanksCoin
ThanksCoin
Reputation Ranking and monetary reward of internet users
Working Prototype
2016-04-02
HonestDice
Live
2015-08-12
NotarEth
Maran Hidskes
Live
2015-08-13
EtherLamp
Live
2015-08-13
EtherScan
Matt Tan
Live
2015-08-14
Live
2015-08-15
Live
2015-08-24
Live
2015-09-03
Live
2015-09-24
Ethereum Alarm Clock
Piper Merriam
Live
2015-09-24
Live
2015-10-07
Etheria
fivedogit
Live
2015-11-07
Oraclize
Thomas Bertani
Live
2015-11-10
Ethereum Pyramid
Live
2015-11-13
Etherdice
vnovak
Live
2015-11-15
Browser-Solidity
chriseth & d11e9
Live
2015-11-20
EtherID
Alexandre Naverniouk
Ethereum Name Registrar
Apache
Live
2015-11-30
Live
2015-12-11
Live
2016-01-02
EthHypeDns
slothbag
Resolve HyperboniaIQDNS ipv6 addresses via etherid.org contract
proprietary
Live
2016-01-05
Live
2016-02-11
EtherDouble
Satoshi :)
Live
2016-02-17
Live
2016-02-18
EthereumWall
LPMitchell
Live
2016-02-25
Live
2016-03-02
Live
2016-03-10
Live
2016-03-11
Live
2016-03-16
Live
2016-03-16
Live
2016-03-16
Live
2016-03-17
Live
2016-03-21
Live
2016-03-23
Live
2016-03-25
Live
2016-03-29
Live
2016-03-29
Live
2016-04-02
Live
2016-04-08
Live
2016-04-10
EthStick
Katatsuki
Live
2016-04-13

Misunderstanding the “Trust Problem”

bloq

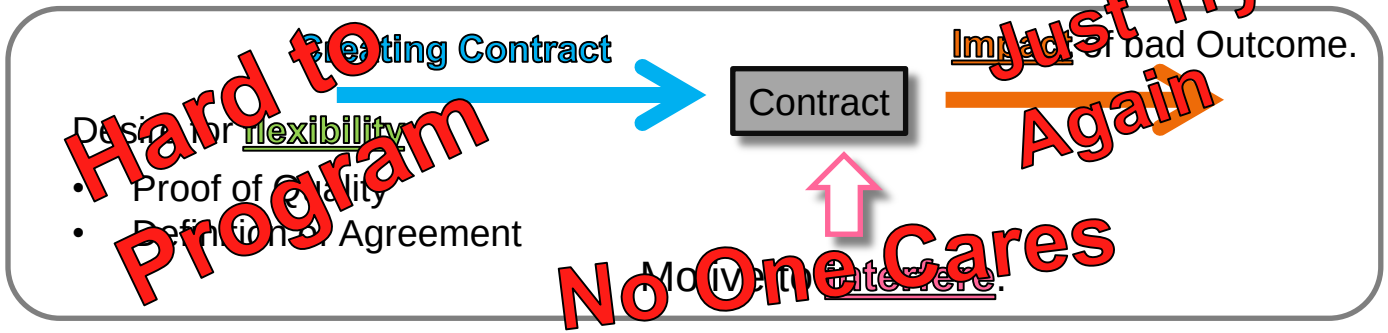
Institutional Value-Usage	Accepts Value	Stores Value
Examples	Restaurant, retail store, gas station, hotel, Netflix, iPhone Games, Uber.	Bank, brokerage firm, lawyer, government, bearer assets.
Qualities of Demand Met	Today’s needs: known, specific, **flexible** / **capricious**.	Tomorrow’s needs: not yet specified, (storage task is **stable** / **well-defined**).
Failures	Small, expected / **investigative**.	Large, unexpected / **catastrophic**.
Fail-**Motive**	**Low**: “Cash on hand.”	**High**: Total stored assets.



Misunderstanding the “Trust Problem”

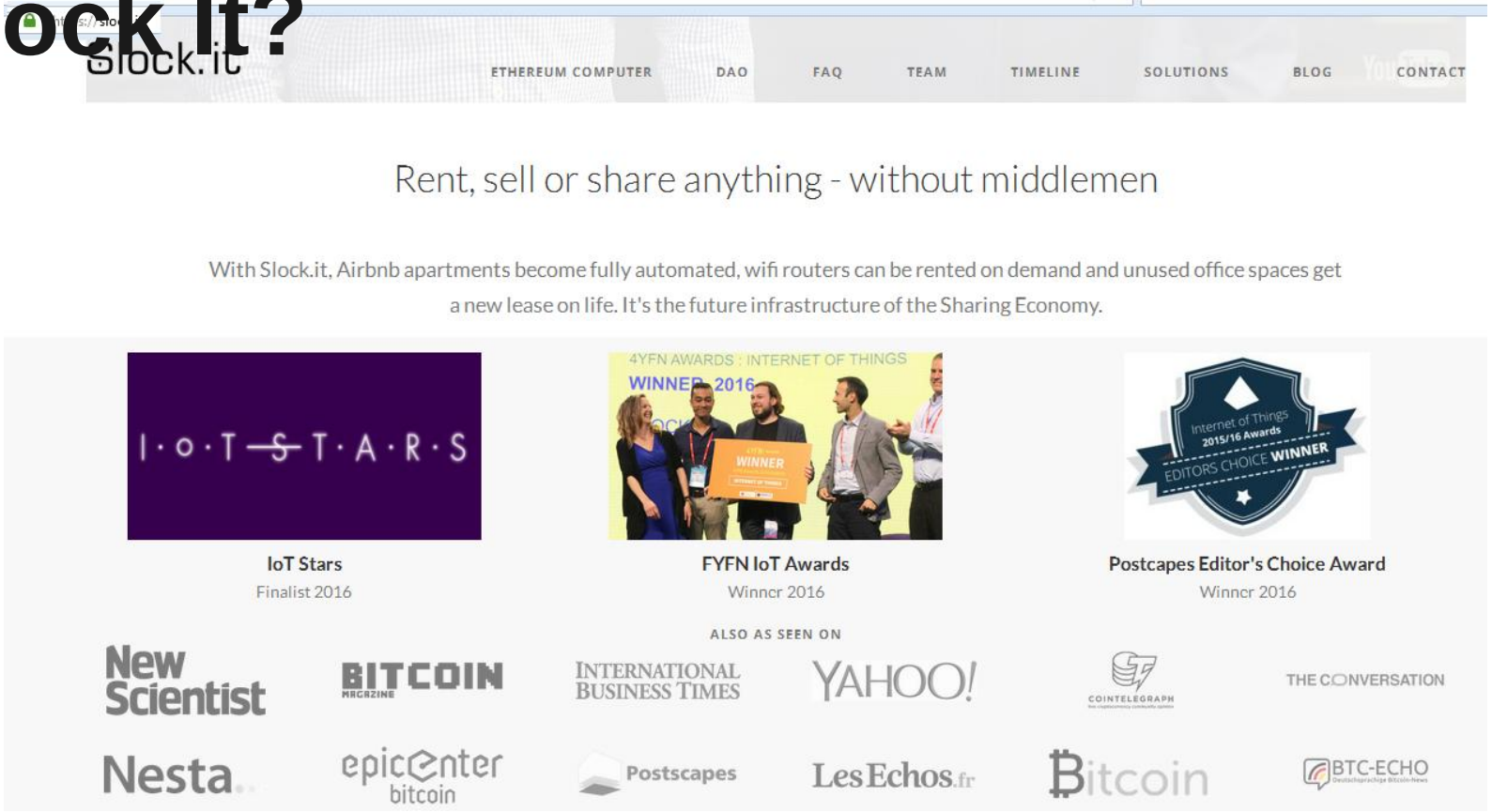
bloq

Institutional Value-Usage	Accepts Value	Stores Value
Examples	Restaurant, retail store, gas station, hotel, Netflix, iPhone Games, Uber.	Bank, brokerage firm, lawyer, government, bearer assets.
Qualities of Demand Met	Today’s needs: known, specific, **flexible** / **capricious**.	Tomorrow’s needs: not yet specified, (storage task is **stable** / **well-defined**).
Failures	Small, expected / **investigative**.	Large, unexpected / **catastrophic**.
Fail-**Motive**	**Low**: “Cash on hand.”	**High**: Total stored assets.



vs. Personal Relationships?
vs. Brands?

Slock It?


Slock.it
ETHEREUM COMPUTER
DAO
FAQ
TEAM
TIMELINE
SOLUTIONS
BLOG
CONTACT
Rent, sell or share anything - without middlemen
With Slock.it, Airbnb apartments become fully automated, wifi routers can be rented on demand and unused office spaces get a new lease on life. It's the future infrastructure of the Sharing Economy.
I·o·T S T·A·R·S
IoT Stars
Finalist 2016
4YFN AWARDS : INTERNET OF THINGS
WINNER 2016
WINNER
FYFN IoT Awards
Winner 2016
Internet of Things
2015/16 Awards
EDITORS CHOICE WINNER
Postcapes Editor's Choice Award
Winner 2016
ALSO AS SEEN ON
New Scientist
BITCOIN MAGAZINE
INTERNATIONAL BUSINESS TIMES
YAHOO!
COINTELEGRAPH
THE CONVERSATION
Nesta
epicenter bitcoin
Postscapes
Les Echos.fr
Bitcoin
BTC-ECHO

Sl

bloq

Russ Harben
@RussHarben

Follow

Wait...you're saying I don't need a smart contract and a blockchain to use a smart loc

reddit.com/r/Python/comme …

[-] cathalgarvey = __import__('py3k') 9 points 23 hours ago

Implemented a door lock for our local hackerspace that uses TOTP keys, so members can use their phones to generate time-sensitive codes to unlock the door using a cheap usb keypad.

Rewriting it in Go though, and want to use different hardware because the RaspPi isn't Free/Open hardware.

permalink save report give gold reply pocket

[+] andrewpolidori 2 points 18 hours ago (3 children)

[+] lherr 1 point 19 hours ago (0 children)

[-] flapflip 3 points 22 hours ago

That's cool as fuck

permalink save parent report give gold reply pocket

[-] cathalgarvey = __import__('py3k') 2 points 12 hours ago

Thank you! It was a necessity for us, because managing physical keys was already becoming a problem and we had an electronic door latch in the building.

It means we can be more welcoming to new members and still have the power to revoke access on lapsed memberships and/or trouble members.

RETWEETS 9 LIKES 18

1:43 PM - 11 Apr 2016

Riccardo Spagni @fluffyponyza · Apr 12

@RussHarben that's crazy talk, it'll never work. That's why I'm so focused on my Pacemaker-on-the-Blockchain altcoin / investment scheme.

(He's joking.)

Creating Contract → Contract → Impact of bad Outcome.

Desire for flexibility.

- Proof of Quality

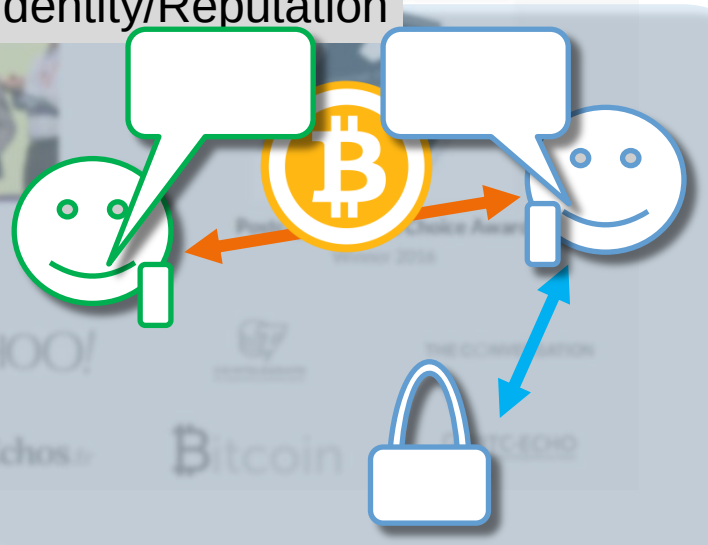
ive to interfere.

Smart Contract? Setup? Security?

Location Revealed

Theft?

Identity/Reputation



If few, why interest? What do they know?

bloq

1. Perhaps nothing? Retail transactions, mining, marketcap, developer mindshare. Usual suspects: fad / bubble (“dot-com”, housing market, Beanie Babies), groupthink / tribalism, money / fame.

2. Bitcoin’s Affinity for Illicit Transactions

3. “Construal Level Theory” (Near/Far Modes)

Network Effects

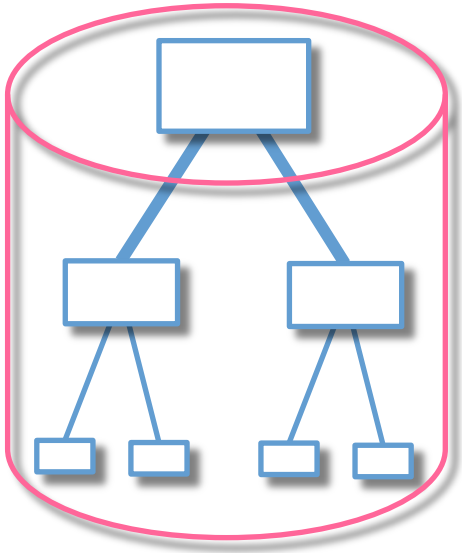
	Bitcoin	Legacy E-Payments	Cash
Dark Market	✓	✗	✓
Light (e)Market	✓	✓	✗

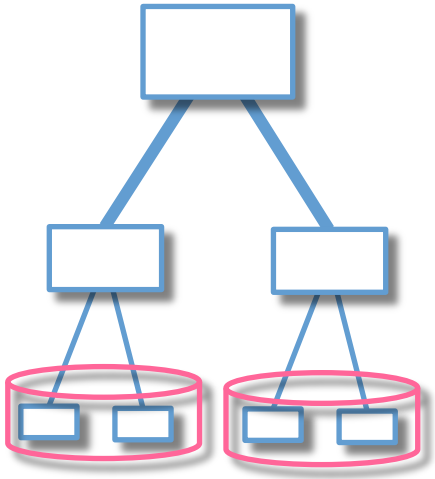
    1. Humans love to “profess” abstraction, to seem impressive. Reality is more specific, sensory, practical. Leads to grandiose planning errors, and instinctual pretentiousness (“social immune system” / “optical illusion”).

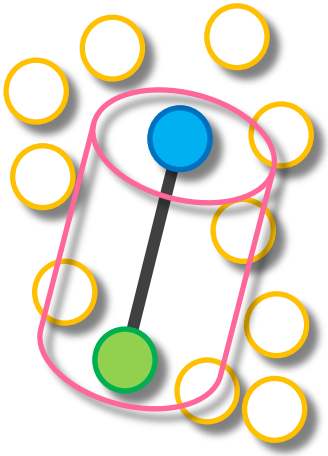
    2. “One day I’ll write a book” vs. “The first sentence will be ‘…’ ”.

    3. “One day we’ll have *smart contracts*” vs. “The first smart contract will be..”

Better: “Ethereum without ETH”



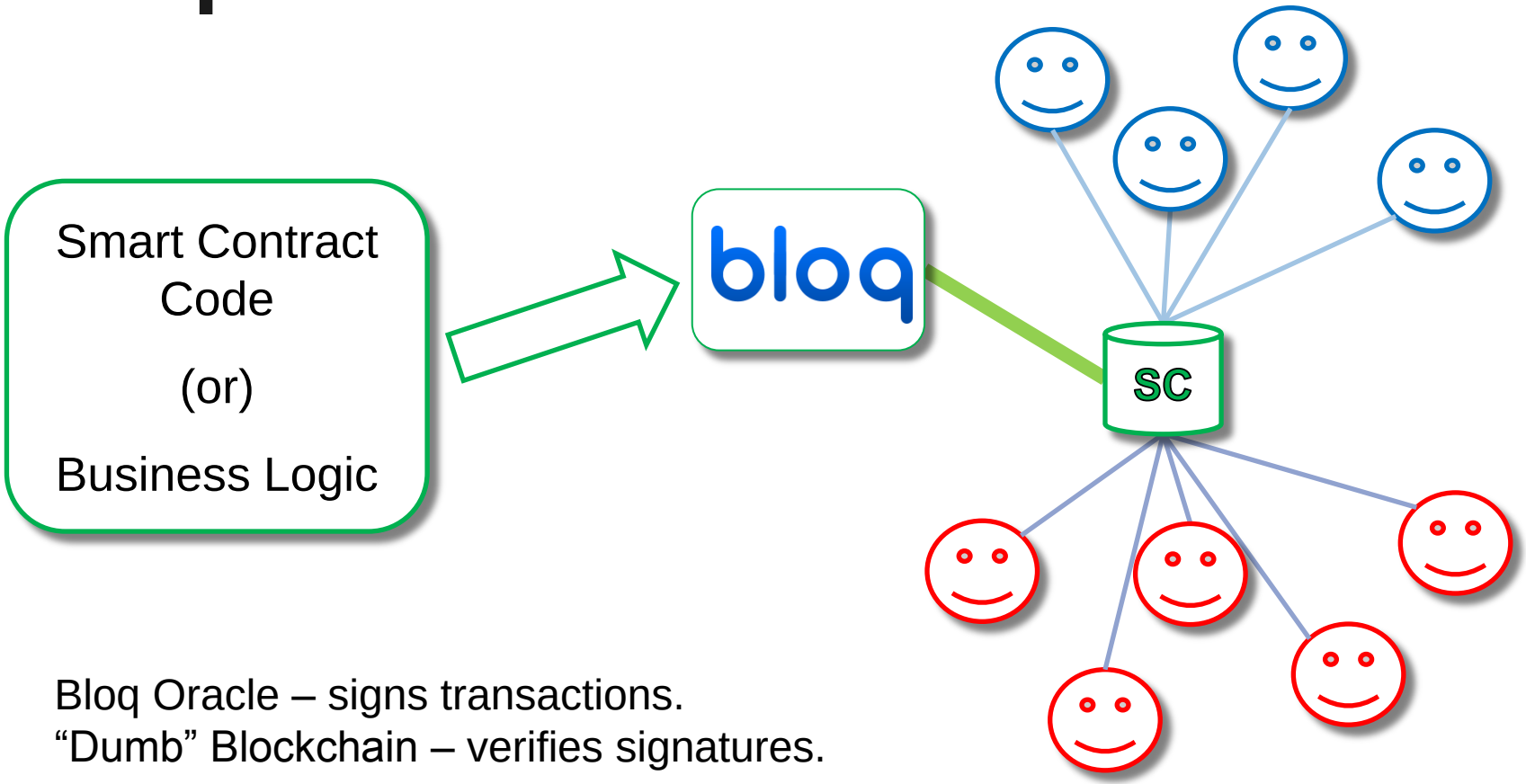




Shard	New Instance
• Access to mining. • (Protects Value of global ETH Token)	• Speed (Realtime, no need for BFT) • Security (Independence, blame allocation) • Modular (Use of BTC/PGP for value/ID)

Mining?

Bloq Ora



Bloq Oracle – signs transactions.
“Dumb” Blockchain – verifies signatures.

Part II - Theory

What are we throwing away if we lose Permissionless Implementation?

Vs. “Oracles” (awesome) ?
Vs. “Brands” (already have) ?
Vs. Bitcoin Soft-forks ?

Multi-Sig
Hivemind

Local Bitcoins
Purse.io

P2SH
Lightning Network

Part III - Theory

Do “Software Developers” and “Smart Contract Designers” have ***fundamentally opposite*** goals?

Contracts: Not Your Typical Software

bloq

- Deceptive: “If you can use **X** to *do Bitcoin*, as well as *do other things*, then **X** must be better!” (ie, solving *the general case*).
- Typically with software, built for one entity -- who wants maximal control/feature-set. **More flexibility = (new = always good). No externalities.**
- Can simply set create_litecoin = FALSE
- ***Additive*** View vs. ***Ecological*** View

Mechanism Design (“Reverse Game Theory”)

bloq

- Bitcoin is what mathematicians would call a “mechanism”.
  - With game theory, task = you start with a **game**, and then describe the **equilibria** under different solution-concepts.
  - With a mechanism, task = you start with a desired **equilibria**, and then try to build a **game** which takes you there.
  - With, software, *more* is never *bad* …however...

MD: Less is More

Main page
Contents
Featured content

Price of anarchy

From Wikipedia, the free encyclopedia

The **Price of Anarchy** (**PoA**) [1] is a concept in economics and game theory that measures how the efficiency of a system degrades due to selfish behavior of its agents. It is a general notion that can be extended to diverse systems and notions of efficiency. For example,

Mathematical definition [edit]

Consider a game $G = (N, S, u)$, defined by a set of players N, strategy sets S_i for each player and utilities $u_i : S \to \mathbb{R}$ (where $S = S_1 \times ... \times S_n$ also called set of outcomes). We can define a measure of efficiency of each outcome which we call welfare function $W : S \to \mathbb{R}$. Natural candidates include the sum of players utilities (utilitarian objective) $W(s) = \sum_{i \in N} u_i(s)$, minimum utility (fairness or egalitarian objective) $W(s) = \min_{i \in N} u_i(s)$, ..., or any function that is meaningful for the particular game being analyzed and is desirable to be maximized.

We can define a subset $E \subseteq S$ to be the set of strategies in equilibrium (for example, the set of Nash equilibria). The Price of Anarchy is then defined as the ratio between the optimal 'centralized' solution and the 'worst equilibrium':

$$PoA = \frac{\max_{s \in S} W(s)}{\min_{s \in E} W(s)}$$

Price of anarchy

From Wikipedia, the free encyclopedia

Prisoner's dilemma [edit]

Consider the 2x2 game called prisoner's dilemma, given by the following cost matrix:

	Cooperate	Defect
Cooperate	1, 1	7, 0
Defect	0, 7	5, 5

and let the cost function be $C(s_1, s_2) = u_1(s_1, s_2) + u_2(s_1, s_2)$. Now, the minimum cost would be when both players cooperate and the resulting cost is $1 + 1 = 2$. However, the only Nash equilibrium occurs when both defect, in which case the cost is $5 + 5 = 10$. Thus the Price of Anarchy of this game will be $10/2 = 5$.

Contracts Tame Anarchy...via *Subtraction*

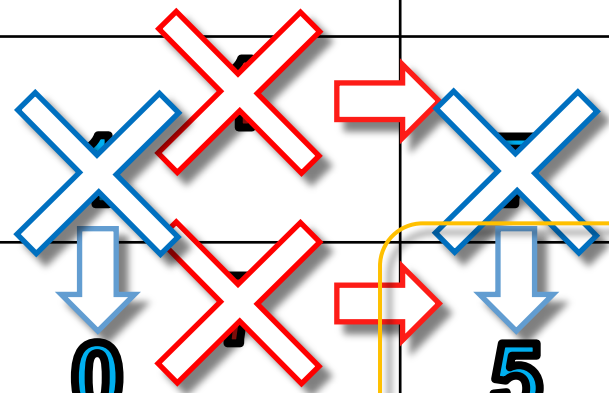
bloq

	Cooperate	Defect
Cooperate	1, 1	0, 7
Defect	7, 0	5, 5

Usual Prisoner's Dilemma (sans Contracts)

: (

	Cooperate	Defect
Cooperate		0
Defect	0	5



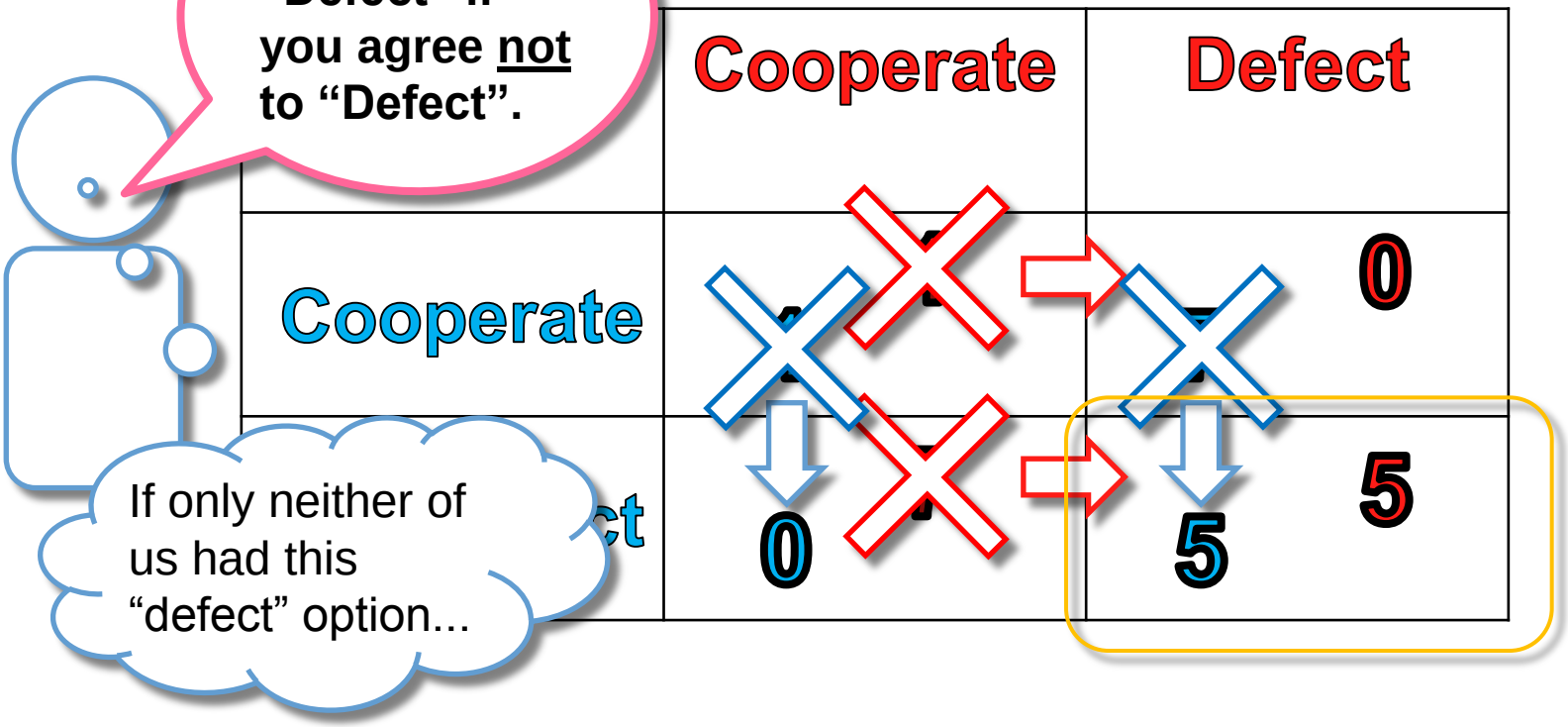
Contracts Tame Anarchy...via *Subtraction*

bloq

I will **NOT** "Defect" if you agree **not** to "Defect".

If only neither of us had this "defect" option...

The "contract" is born...



Contracts Tame Anarchy...via *Subtraction*

	Cooperate	Defect
Cooperate	1 1	7 0
Defect	0 7	5 5

4 fewer years in prison, each...in a world where the players can NOT defect.

Each player would work up to 4 years to *prevent* such an option from existing!

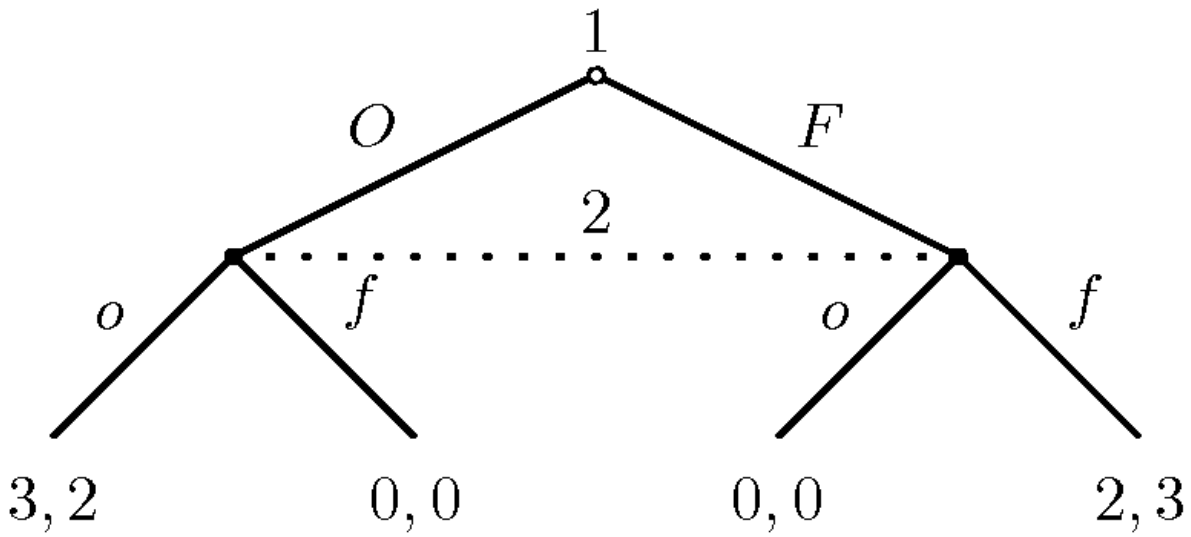
The introduction of the “Defect” option effectively robbed the players of 8 total years of freedom.

MD: Less is More (continued)

- *How* did that work? They agreed to do “fewer” valid things.
- ***Contracts aren’t magic!***
  - They “create nothing”.
  - They only operate on the *space of human action…*by **shrinking** it.
- **Less trust** was required, under contract, because untrustworthy actions were *removed.* “Freedom” was *destroyed*.

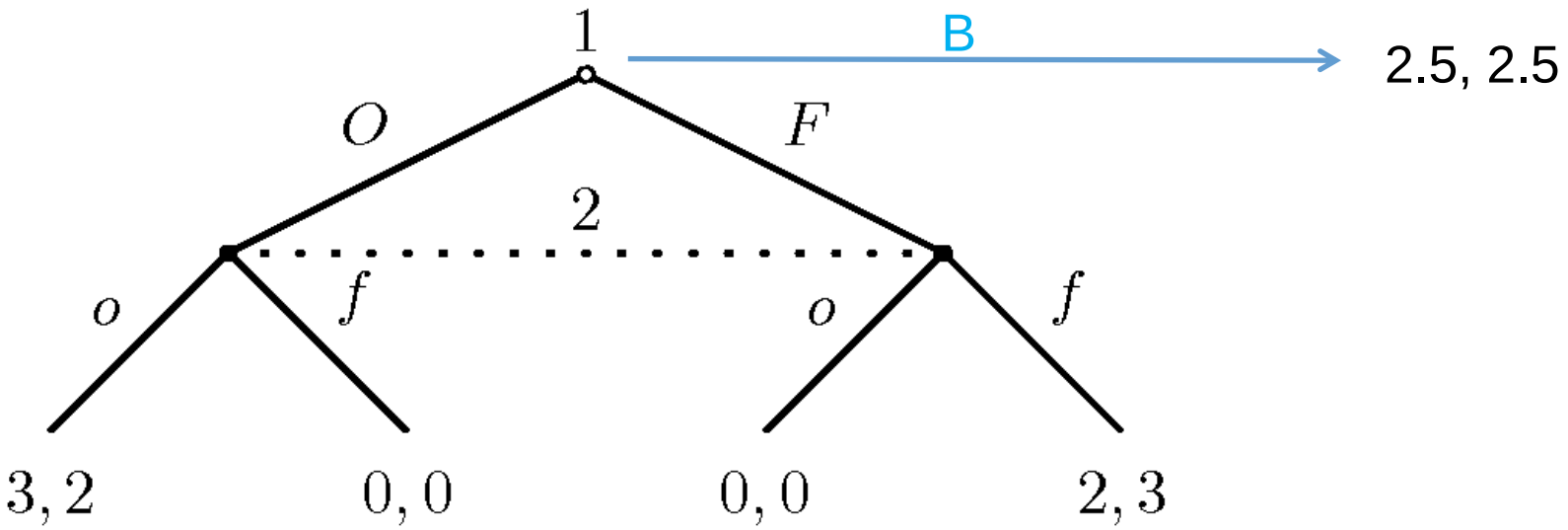
A Converse Example

"Battle of the Sexes"

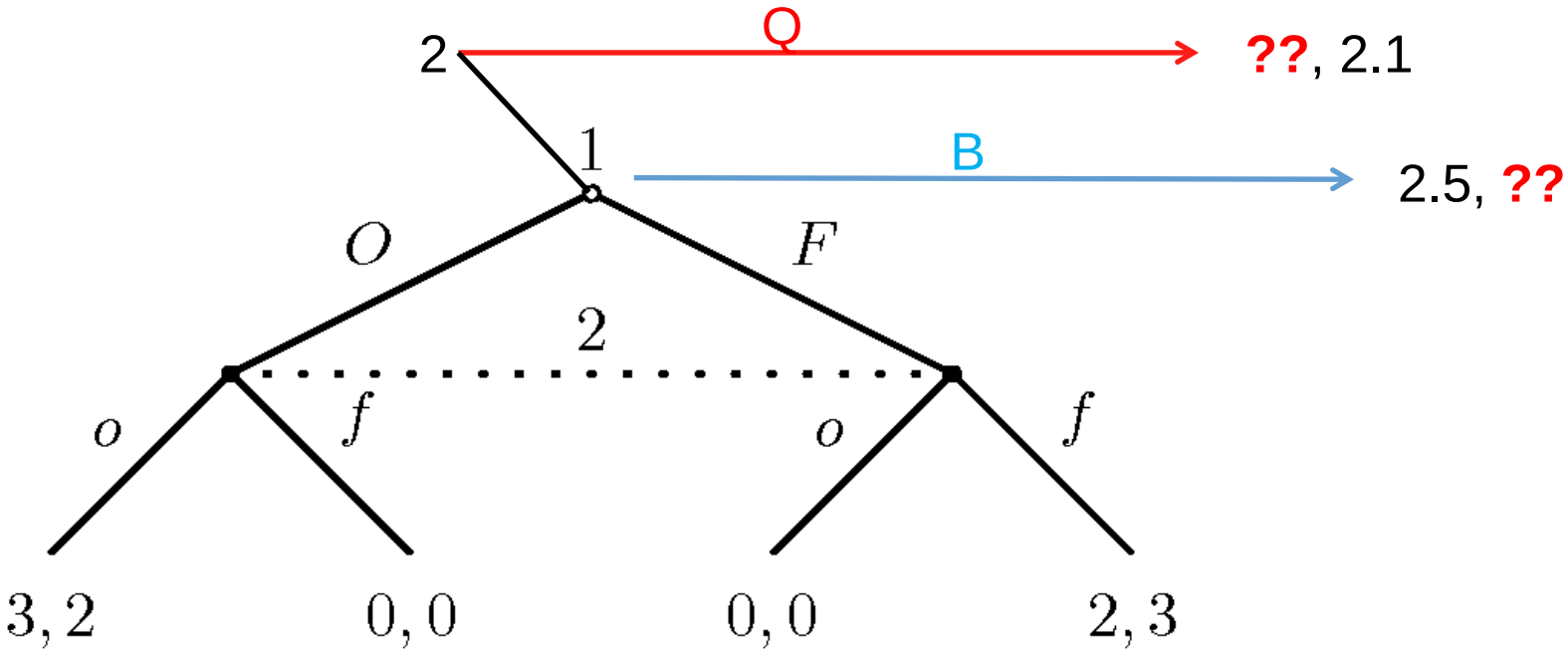


	_o	_f
O	3 2	0, 0
F	0, 0	2 3

“Battle of the Sexes + Bar”



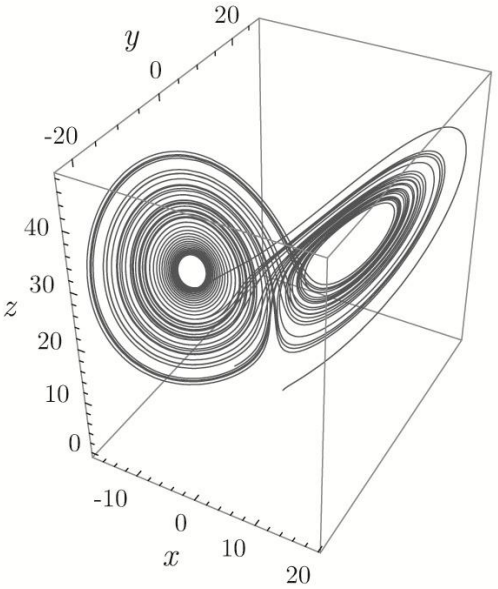
“ “Battle of the Sexes + Bar” + Bar ”

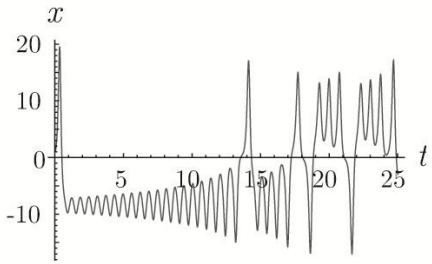


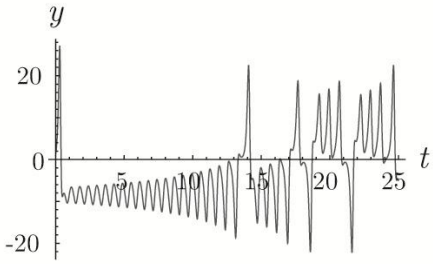
Escalating Interaction

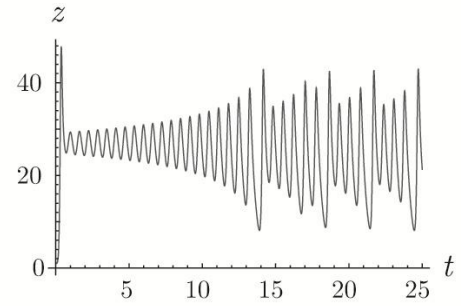
(Lorenz System)

$$dx/dt = 10(y - x),$$
$$dy/dt = 28\ x - y - x\ z,$$
$$dz/dt = -(8/3)\ z + x\ y,$$
$$x(0) = y(0) = z(0) = 1.$$









Curse of dimensionality.

2 ---> 3
3 ---> 4
8 ---> 9

Often, Controls are Good,
(they help with teamwork).

The Bitcoin Contract

bloq

1. Blocks are ***prohibited*** from including:
   * transactions with bad signatures
   * double-spends

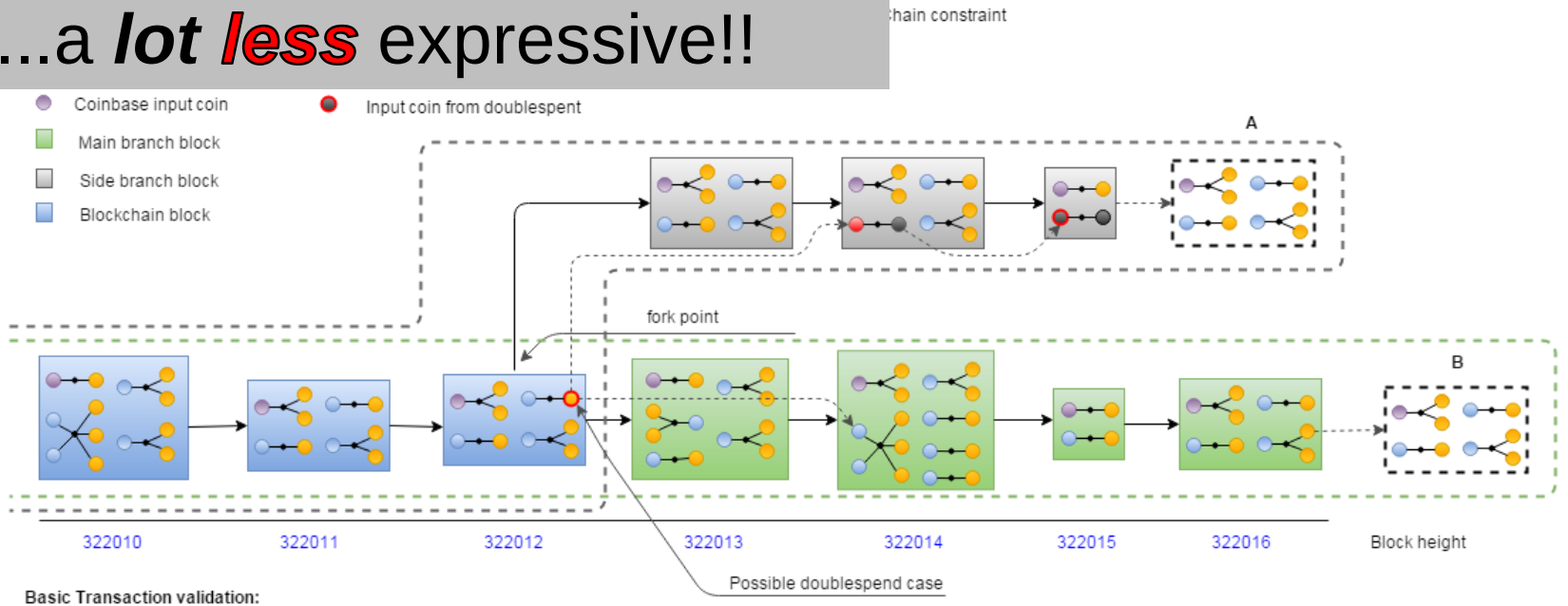
2. Bitcoin's *main* revolutionary feature: ***banned double-spends.*** No need to ***trust*** a server to protect you from double-spends.

3. Bitcoin is ***less*** functional / expressive than LevelDB...

Compare to:
"Permissionless"
Transacting

Blockchain transaction validation rules

...a ***lot less*** expressive!!



Basic Transaction validation:

- Size in bytes from 64 to [MAX_BLOCK_SIZE (1Mb) - BLOCK_HEADER_SIZE(80)]
- Inputs count >=1, Outputs count >= 1
- Sum of input values >= sum of output values
- No duplicate inputs (doublespend inside transaction)
- No coinbase input coin
- All input coins are exist
- In case input coin is from Coinbase transaction, this Coinbase transaction must have at least COINBASE_MATURITY (100) confirmations
- Script is valid (execute input signature script then execute input coin pubkey script in case P2SH execute reedom script)
- Outputs valuee in legal money range (0 - 21 000 000 BTC)
- Locktime must be in the past (or less than or equal to the current blockchain height), or all of inputs sequence numbers must be 0xffffffff.

Transaction validation inside new block:

- Chain constraints: All input coins must be exist and unspent in Blockchain or in new block or in extending branch in case of fork.
- All transaction must pass Basic validation.
- Coinbase transaction is exception and must be first and single transaction in new block. And have 1 Input coin (coinbase input coin [hash=0, n=-1])
- Coinbase transaction Signature script length in bytes 2-100
- Coinbase transaction input coin value <= (50 * 100000000) >> (block height / 210000) + sum of all block transaction fees
- Limit of transaction sig opcounts MAX_BLOCK_SIGOPS = MAX_BLOCK_SIZE/50

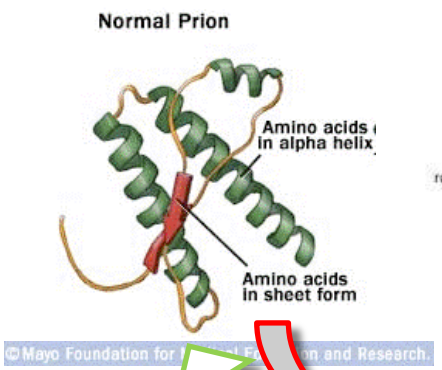
Not My Work -- http://i.stack.imgur.com/QvgMr.png

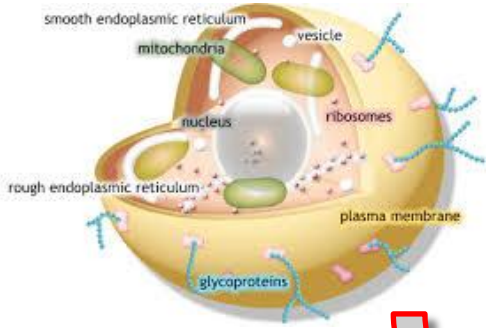
How is Bitcoin Upgraded?

- Notice that **100%** of Bitcoin's upgrades have been rolled out via "soft fork".
- Each soft fork is a *reduction* in total permission!
- Forwards compatibility = no breach of contract.

Autonomy and Coordination

bloq





Constrains

"Freedom"

Prion Disease

- Lysosomes
- Structure (Cell Wall)
- Composition (Water)

Constrains

"Freedom"

Cancer

- DNA
- G - hormone
- RNase H2
- gene p53
- T-cells

Constrains

"Freedom"

Rioting / Theft

- Firms / Employers
- Property rights
- Contract Adjudication
- Police / Defense
- Norms / Traditions

Less is More – Biology

- Life – Eukaryotic Cell – Multicellular Life – Social Animals – Domestication of Plants/Animals
- Mitochondrial disease, cancer (individual cells start pursuing their own self-interest, they reject all laws as ‘unjust coercion’, but they don’t think it through, kill host, kill themselves), prey gets away, chickens kill farmer! Would we tolerate one desire to kill everyone, zebra cant be tamed…
- Mutations are good *across* organisms, but bad within-organisms. Every improvement is a change, but random changes to our stuff is 99.99% catastrophic.
- Local enslavement is global autonomy. Local autonomy is global chaos. Free market “budget constraint”! No free market has ever existed in a society without reliable capital preservation / theft-prevention. Limited Government. Soviet empires.
- As animals, what would be best for us would be to watch something else evolve (or force it to evolve), and then bring in anything we like. For blockchains, R&D to take place outside the system, and then be consciously brought into the system.

Code Obfuscation

That is a valid computer program. ----→

www.ioccc.org/1988/phillipps.c

5th International Obfuscated C Code Contest (1988)

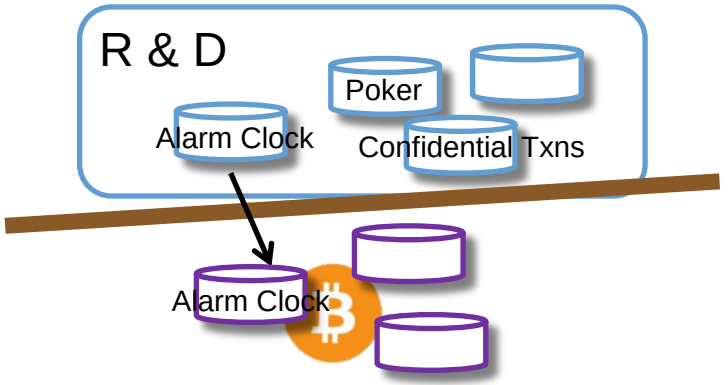
```
main(t,_,a )
char
*
a;
{
                                return!

0<t?
t<3?

main(-79,-13,a+
main(-87,1-_,
main(-86, 0, a+1 )


+a)):

1,
t<_?
main( t+1, _, a )
:3,

main ( -94, -27+t, a )
&&t == 2 ?_
<13 ?

main ( 2, _+1, "%s %d %d\n" )

:9:16:
t<0?
t<-72?
main( _, t,
"@n'+,#'/*{}w+/w#cdnr/+,{}r/*de}+,/*{*+,/w{%+,/w#q#n+,/#{l,+,/n{n+,/+#n+,/#;#q#n+,/+k#;*+,/'r :'d*'3,}{w+K w'K:'+}e#';dq#'l q#'+d'K#!/+l
nc{nl]'/#{l,+'K {rw' iK{;[{nl]'/w#q#n'wk nw' iwk{KK{nl]!/w{%'l##w#' i; :{nl]'/*{q#'ld;r'}{nlwb!/*de}'c ;;{nl'-{}rw]'/+,}##'*}#nc,',#nw]
:
t<-50?
_==*a ?
putchar(31[a]):

main(-65,_,a+1)
:
main((*a == '/') + t, _, a + 1 )
:

0<t?

main ( 2, 2 , "%s")
:*a=='/'||

main(0,

main(-61,*a, "!ek;dc i@bK'(q)-[w]*%n+r3#l,{}:\nuwloca-O;m .vpbks,fxntdCeghiry")

,a+1);}
```

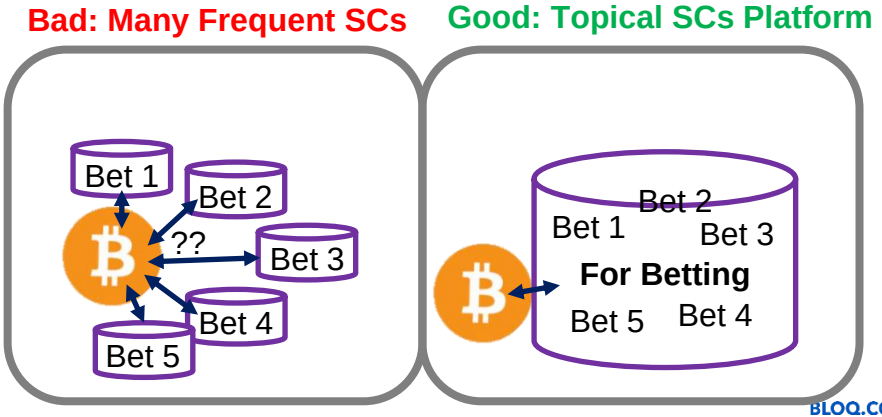
Restatement

Treat sidechains with the care/respect of a soft fork:

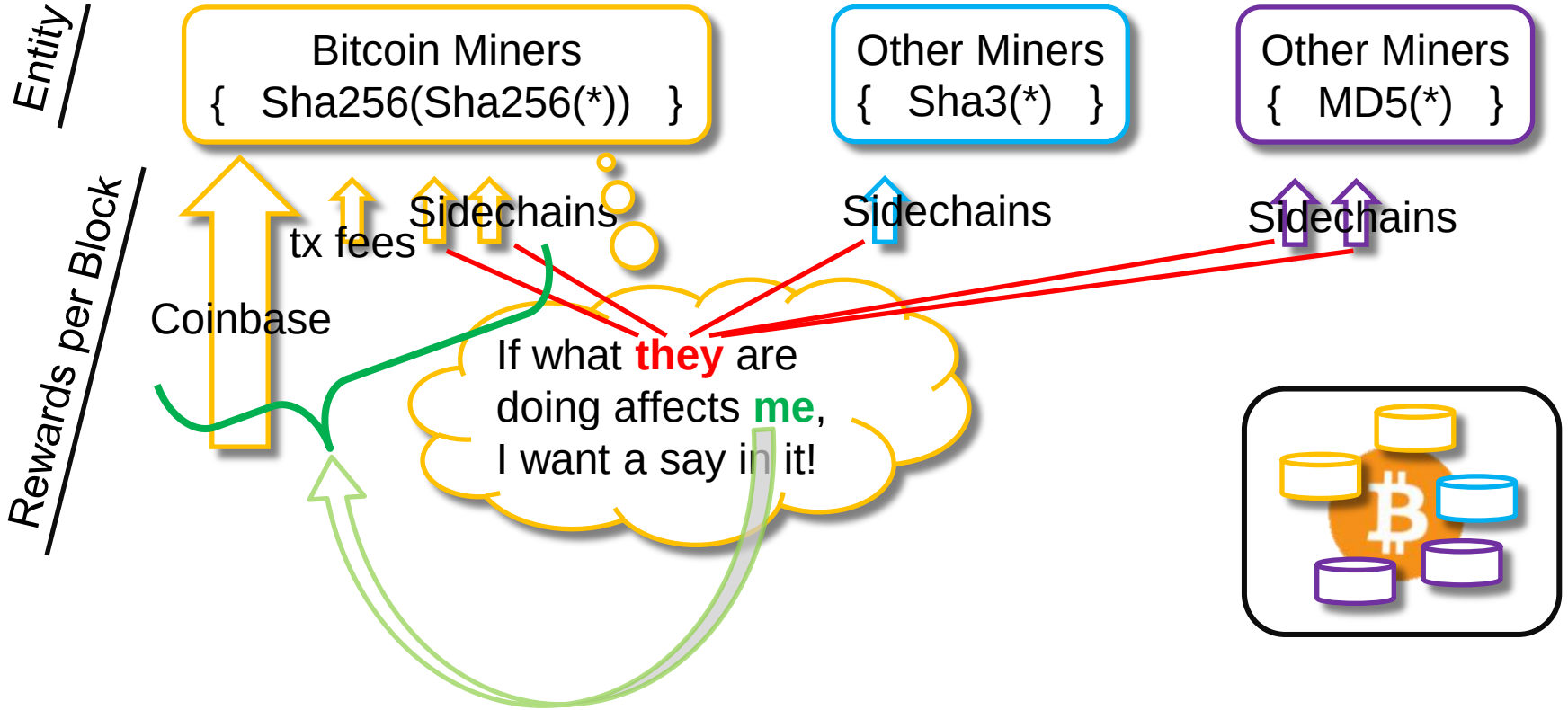
- Slow, Rare
- Documented, Discussed
- Willfully Activated

Miners need to *understand* the Sidechains' purpose.



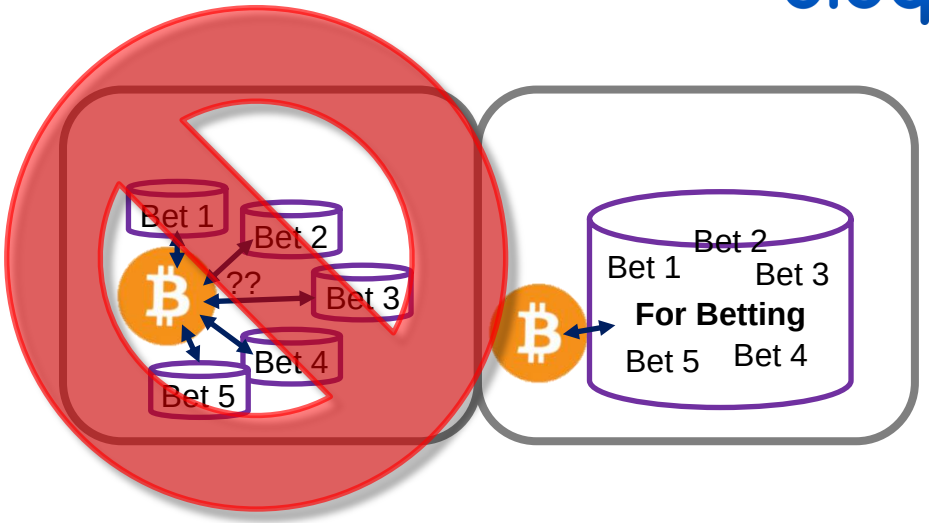


Restatement – Internalize the Externalities



Conclusion

- Avoid the Grey Goo
  - P. Innovation = Good.
  - P. Implementation = Bad.
- Mechanism Design / “contracts”
  - ....where the *emphasis* is on ***what can’t be done*** .
  - ...allowing miners to ***ban*** things, is appropriate. It’s just a “bigger” version of what a normal contract does.
- Script upgrades, MAST, OP_VirtualBox – don’t overdo it!



bloq

Thank You
@truthcoin

paul.sztorc@bloq.com