

Wasting time in the bit economy

Description

default watermark

One way to demonstrate your wealth is to show how much free time you have. Freed from the drudgery required to keep fed, secure and comfortable, the wealthy have time on their hands to sit around in coffee shops, take long holidays, indulge in unprofitable hobbies, and acquire esoteric skills and affectations that are surplus to the requirements to survive.

According to a seminal 19th century book by sociologist Thorstein Veblen, these are customs of the so-called â??leisure class.â?• Note that leisure pursuits are here signs of wealth, and may operate independently of oneâ??s actual financial status. So the middle classes emulate such practices in order to appear to participate in this wealth-leisure economy.

Nor do we need to think about the pleasure, sociability or therapeutic value of our various pastimes; nor how they fill time and keep us from being bored. According to Veblenâ??s provocative theory, they are initially at least, signs of our participation in temporal surplus.

Leisure games

Solving artificial puzzles, like the Rubikâ??s Cube fits as a means of soaking up surplus time, and demonstrating as much. Never mind the mind-training and the joy of the challenge. Sociable and competitive puzzle solving (e.g. Cluedo) provides a similar function as a show of temporal surplus. (Blogging, and/or trying to understand blockchain tech, provide similar time-soaking challenges for some.)

By a different reading, surplus time is never useless down time, especially if it involves sociability. After all, sitting around the campfire at the end of a heavy day of prehistoric hunting, or playing Cluedo with company and a glass of wine, helps reinforce community ties and trust.

Digital money

The *Theory of the Leisure Class* provides a perverse introduction to one intriguing aspect of digital money (see post). Computer CPUs lie idle much of the time. As well as transferring inputs to screen

displays and databases, and pushing data around networks, they can also do useless things such as work out the solution to a cryptographic puzzle. Why do that?

Dumb computer processing can be used to check and confirm that a financial transaction is genuine, hasnâ??t appeared before in the ledger, and hasnâ??t been corrupted in some way. How?



Error checking

default watermark

A mobile phone text message â??helloâ?• consists of ascii characters 104, 101, 108, 108 and 111. If you treat this as a row of 5 numbers and add them up then you get the number 532. If the message gets messed up on the way it may arrive at the receiver as â??helLo.â?• That gives up a sum of 500 when you add up the asci numbers. So the message transmitter software should not only transmit the message, â??hello,â?• but also append the sum 532.

The receiver software would then calculate the sum for the message it received and compare that with the number 532 submitted with the original message. 532 and 500 are not the same, and so thereâ??s a discrepancy. The receiver software would then send a signal to the message sending software to re-send the message. What I have described is a crude form of error checking.

The number representing a reduced string of characters in this way is called a â??checksum.â?• It can be done in a much more efficient way than Iâ??ve described here, and with elaboration. The *checksum function* will be more elaborate than simply adding up ascii numbers. In the method I described above, â??helloâ?• would have the same checksum as â??helol,â?• so the checksum function needs to take account of the position of the characters, and cases where errors cancel each other out. Note that checksum doesnâ??t signal any corrections. It just tells the sender that the message was corrupted.

Checksum functions can also be applied to whole files, such as word processed documents, and provide a means of checking that documents have not been corrupted or altered. Note that it is not possible to reconstruct the message from the checksum. You canâ??t directly deduce the message â??helloâ?• from the number 532.

Cryptographic challenge

However, knowing that the checksum was derived by adding ascii numbers, a computer program would have little difficulty iterating all combinations of ascii characters that add up to 532. It could then compare these with dictionary entries to deduce that the word is â??hello,â?• by brute force. So thatâ??s an example of a cryptographic puzzle. For longer messages, and with much more

sophisticated checksum functions this decoding would be near impossible.

(Some computational techniques generate what is known as a â??hash.â?• Itâ??s a standardised way of representing a string of characters, as in the case of a book title or a personâ??s full name, as a single number, similar to how I have described a checksum. Itâ??s a surrogate for the longer title. Programs that index, order and search databases work best with hash surrogates rather than moving longer character strings around.)

Adding value to a bit string

One way to make something as ephemeral as a character string in a database scarce and hence of value is to inject time and effort into its production, use, transmission or verification. It takes more time, effort and energy to create a fake new Â£1 coin than it does to buy one â?? for a pound. One way to deter forgers of digital strings, files, or ledger entries is to ensure that it would be prohibitively expense to do so.

default watermark

One method is to ensure that the only way an entry can appear on a shared ledger is for the computer of a co-owner of the ledger to undertake a challenge to vouch for the transactionâ??s authenticity. That challenge involves solving a cryptographic puzzle based on a hash of the current state of the ledger and the latest group of transactions. Once solved, the solution is distributed to other co-owners of the ledger i.e. it is attached to the transaction in the shared ledger.

All of that involves work, soaking up computer â??leisure time.â?• Itâ??s not possible to go back and alter transactions in the ledger as transactions (or blocks of transactions) already include authenticated hash codes. Any alterations would disrupt the integrity of the shared ledger and be detected. Similar methods keep bogus transactions at bay.

Thereâ??s more to be said, and better metaphors can be deployed, though many online explanations of bitcoin, â??proof of work,â?• â??blockchainâ?• and related theories seem to assume a reasonable level of proficiency in computer theory. Wikipedia includes a reference to how Pacific Islanders added value to common, ordinary sea shells by expending labour on them â?? as a proto-bit currency. Iâ??ll keep looking.

While investigating for further clarification and for some design implications of bit coin methods and technologies see Chris Speedâ??s post on Practicing the Blockchain.

Reference

- Veblen, Thorstein. 1998. *The Theory of the Leisure Class*. Amherst, New York: Promethius. First published in 1899.

Hereâ??s people sitting around a campfire: The discovery of fire, illustration by Cesare Cesariano (1475-1543) to Vitruviusâ?? *Ten Books of Architecture*. [Various online sources.]



Insert watermark

Category

1. Economics

Tags

1. bitcoin
2. blockchain
3. ciphercity
4. digital money

Date Created
July 1, 2017
Author
rcoyne99