

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

COINBASE, INC.,

Petitioner

v.

VERITASEUM CAPITAL LLC

Patent Owner.

IPR 2023-00751

U.S. Patent No. 11,196,566

**DECLARATION OF ANDREW MILLER IN SUPPORT OF PETITION
FOR INTER PARTES REVIEW OF
U.S. PATENT NO. 11,196,566**

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
U.S. Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

	Page
I. QUALIFICATIONS	4
II. UNDERSTANDING OF RELEVANT LEGAL PRINCIPLES	6
III. TECHNICAL BACKGROUND	9
A. Bitcoin	10
B. Transactions.....	13
C. Blockchain Technology.....	16
D. Cryptography and Bitcoin	17
IV. THE '566 PATENT	19
A. Summary of the '566 Patent.....	19
1. Exemplary Embodiments	19
a. Letter of Credit Embodiment.....	19
b. Swap Embodiment.....	21
2. Network Environment.....	23
B. The '566-Provisional does not support all the limitations of the independent claims.	24
C. Claim Construction	26
D. Level of Ordinary Skill in the Art	26
V. GROUND 1: COMBINATION OF HEARN AND ARMSTRONG RENDERS CLAIMS 1-3 OBVIOUS.....	28
A. Overview of Hearn	28
B. Overview of Armstrong	37
1. Armstrong	37
2. Armstrong-Prov1 and Armstrong-Prov2 Support at Least One Claim of Armstrong	42
C. Combination of Hearn and Armstrong.....	44
D. Independent Claim 1	49
1. Preamble [1P].....	51
a. “Memory” limitation [1A].	56
b. “Network interface” limitation [1B].	59
c. “Term” limitations [1B.1]-[1B.3].	62
(i) “Principal Data” [1B.1].....	63

TABLE OF CONTENTS (continued)

	Page
(ii) “Reference” to a “data source” [1B.2].....	68
(iii) “Expiration Timestamp” [1B.3]	71
d. “Computer Processor” limitation [1C].	74
e. “Computer Processor” action limitations [1C.1]-[1C.4].	75
(i) “Read the First Private Key” limitation [1C.1], “Compute a First Cryptographic Signature” limitation [1C.2]	76
(ii) “Create an Inchoate Data Record” [1C.3]	79
(iii) “Inchoate Data Record” content limitations [1C.3a]-[1C.3c]	80
(iv) “Publish the Inchoate Data Record” [1C.4]	85
f. “Distributed Ledger” limitation [1D]	86
g. “Create a Complete Data Record ...” [1E].....	88
h. “Sign[]” and “Save a Copy of the Inchoate Data Record” limitation [1F].....	90
i. “Verif[y] the Recording of the Complete Data Record in the Distributed Ledger” limitation [1G].....	91
E. Dependent Claim 2	93
F. Dependent claim 3	94
VI. GROUND 2: THE COMBINATION OF ARMSTRONG, HEARN, AND ZIEGLER RENDERS CLAIMS 1-3, 7 AND 8 OBVIOUS.	101
A. Overview of the Combination	101
1. Ziegler	101
2. Combination of Hearn, Armstrong, and Ziegler.....	103
B. Independent Claim 7	105
[7P] <i>A system for processing a transaction between a first client device and a second client device via a transfer mechanism, the system comprising a computing device, the first client device, the second client device, and the transfer mechanism;</i>	105
a. Preamble [7P]	108
b. “Computing Device” limitations [7A]	108
c. “First Client Device” limitations [7B]	110
(i) Components [7B.1]-[7B.3].....	110

TABLE OF CONTENTS

(continued)

	Page
(ii) “ <i>Second Computer Processor</i> ” action limitations.....	112
d. “ <i>Second Client Device</i> ” limitations [7C].....	117
e. “Sign[]” and “Save a Copy of the Inchoate Data Record” limitation [7D].....	118
f. “ <i>Distributed Ledger</i> ” limitation [7E].	119
g. “ <i>Transaction is Created by Broadcasting</i> ” limitation [7F].....	119
h. “ <i>Verif[y] the Recording of the Complete Data Record in the Distributed Ledger</i> ” limitation [7G].....	120
C. Dependent Claim 8	120
VII. CONCLUSION.....	121

I, Andrew Miller, declare as follows:

1. I have been engaged by Perkins Coie LLP on behalf of Coinbase (“Petitioner”), to provide this Declaration concerning technical subject matter relevant to the petition for Inter Partes Review (“Petition”) of U.S. Patent No. 11,196,566 to Middleton (“the ’566 patent”).

2. I am over 18 years of age. I have personal knowledge of the facts stated in this Declaration and could testify competently to them if asked to do so.

3. I have reviewed and am familiar with the specification and the claims of the ’566 patent. In general, I will cite to the specification of a United States patent using the following formats: (Patent No., Col:Line Number(s)) or (Patent No., Paragraph Number(s)). For example, the citation (’566 patent, 1:1-10) points to the ’566 patent specification at column 1, lines 1-10. Also, for convenience, I use italics to denote limitations from the challenged claims.

4. All of the opinions contained in this Declaration are based on the documents I reviewed and my knowledge and professional judgment. In forming the opinions expressed in this Declaration, I reviewed the following documents:

- U.S. Patent 11,196,566 to Middleton (“the ’566 patent”; CB-1001)
- File History of U.S. Patent 11,195,566 (“the ’566 FH”; CB-1002)
- U.S. Provisional Application 61/990,795 (“’566-Prov” CB-1005)

- U.S. Patent Publication 2015/0262168 to Armstrong, et al. (“Armstrong”; CB-1006)
- U.S. Provisional Application 61/954,434 (“Armstrong-Prov1”; CB-1007)
- U.S. Provisional Application 61/990,017 (“Armstrong-Prov2”; CB-1008)
- Hearn, “Contracts”, available at <https://web.archive.org/web/20140209124419/https://en.bitcoin.it/wiki/Contracts> (“Hearn”; CB-1009)
- U.S. Patent 7,387,240 to Ziegler (“Ziegler”; CB-1010)
- Antonopoulos, “Mastering Bitcoin”, chapters 1-4, published on GitHub May 3, 2014 (“Antonopoulos”; CB-1012)
- Schneier, “Applied Cryptography” (CB-1013)
- Menezes, “Handbook of Applied Cryptography” (CB-1014)
- U.S. Patent Publication 2015/0220892 to Allen (“Allen”; CB-1015)
- U.S. Patent 8,135,650 to Walker, et al (“Walker”; CB-1016)
- Webster’s Encyclopedic Unabridged Dictionary of the English Language, Random House (2001), page 966 (CB-1017)
- Hearn, “Contract”, available at <https://web.archive.org/web/2015110604270/https://en.bitcoin.it/wiki/Contracts> (CB-1018)
- Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System” (CB-1019)

I have also reviewed and am familiar with any other document referred to in this Declaration.

5. I have been asked to provide my technical opinions regarding how a person of ordinary skill in the art would have understood the claims of the '566 patent at the time of the alleged invention, which I have been asked to assume is the 2014 timeframe. For purposes of whether the teachings of the prior art render the claims of the '566 patent obvious, I have been asked to assume the date of May 9, 2014. I have also been asked to provide my technical opinions on how concepts in the '566 patent specification relate to claim limitations of the '566 patent. In reaching the opinions provided herein, I have considered the '566 patent, its prosecution history, and the references cited above and have drawn as appropriate on my own education, training, research, knowledge, and personal and professional experience.

I. QUALIFICATIONS

6. In formulating my opinions, I have relied on my knowledge, training, and experience in the relevant field, which I will briefly summarize. A more detailed summary of my background, education, experience, and publications is set forth in my curriculum vitae (CV), which is submitted as CB-1004.

7. I have personal knowledge of the facts and opinions set forth in this declaration and believe them to be true. If called upon to do so, I would testify competently thereto. I have been warned that willful false statements and the like are punishable by fine or imprisonment, or both.

8. I am being compensated for my time at my standard consulting rate. I am also being reimbursed for expenses that I incur during the course of this work. My compensation is not contingent upon the results of my study and analysis, the substance of my opinions, or the outcome of any proceeding involving the Challenged Claims. I have no financial interest in the outcome of this matter or in any litigation involving the '566 patent.

9. I am an Assistant Professor in the Electrical and Computer Engineering Department at University of Illinois at Urbana-Champaign. My research interests lie at the intersection of computer security, programming languages, and distributed systems. My current focus is in building secure decentralized systems, such as cryptocurrencies.

10. I received my Bachelor of Science degree in Computer Science (“CS”) in 2009 and my Master’s degree in CS in 2012, both, from University of Central Florida. I received my PhD in CS in 2016 from University of Maryland. My PhD thesis was on applied cryptography and programming language techniques for smart contracts and blockchain applications.

11. In 2016, I joined University of Illinois’ Electrical and Computer Engineering (“ECE”) department as a full-time Assistant Professor, a position I have held since that time. I have been a member of several Committees since the time I joined the University of Illinois. Some notable committees include Financial Cryptography Committee (2017-2022), where I was a member of the Steering Committee between 2018 and 2022. I was also a member of the IEEE Security and Privacy Program Committee (2017-2019), which is considered the most prestigious conference in computer security. In addition, in 2017 I was a Program Committee Chair for the 4th Bitcoin Research Workshop, which featured interdisciplinary blockchain and cryptocurrency related research.

12. I was an invited lecturer at Purdue University in April of 2020, where I lectured about the use of multiparty computation (“MPC”) in combination with smart contracts. I was also invited to talk at PLCrypto Workshop in May of 2022, which was related to programming language techniques for secure applications, including in blockchains. In addition, I was the keynote speaker at the 4th

Blockchain Technology Symposium in June of 2022, which focused on defining future research directions in blockchains, smart contracts, and privacy enhancing technology.

13. I have contributed, as an author, to over 20 publications (related to Security, Cryptocurrency, and more). Those publications cover a wide range of related technologies such as blockchain, cryptocurrencies (including Monero and Bitcoin), distributed key generation, and others. One main theme of my research has been to translate powerful tools from the applied cryptography field, such as zero knowledge proofs and multiparty computation, into programming libraries to make them available to smart contract developers.

14. I have received several awards including the NSF CAREER Award 1943499 (2021), which supports assistant professors that have the potential to serve as academic role models in research and education, the IEEE S&P Distinguished Reviewer Award (2018), which recognizes excellent work on a program committee, and the University of Maryland, Larry S. Davis Doctoral Dissertation Award, which annually recognizes the Computer Science department's strongest dissertation.

II. UNDERSTANDING OF RELEVANT LEGAL PRINCIPLES

15. I am not a lawyer, and I will not provide any legal opinions. Although I am not a lawyer, I have been advised certain legal standards are to be applied by

technical experts in forming opinions regarding the meaning and validity of patent claims.

16. I understand that a patent claim is invalid if it is anticipated or obvious in view of the prior art, and that a claim can be unpatentable even if all of the requirements of the claim cannot be found in a single prior-art reference. I further understand that invalidity of a claim requires that the claim be anticipated or obvious from the perspective of a person of ordinary skill in the art at the time the invention was made.

17. I have been informed that a patent claim is invalid if it would have been obvious to a person of ordinary skill in the art. In analyzing the obviousness of a claim, I understand the following factors may be taken into account: (1) the scope and content of the prior art; (2) the differences between the prior art and the claims; (3) the level of ordinary skill in the art; and (4) any so called “secondary considerations” of non-obviousness, if they are present. I am not aware of any evidence of secondary considerations of non-obviousness relevant to the ’566 patent. I reserve the right to supplement this Declaration if Patent Owner introduces evidence of secondary considerations of non-obviousness.

18. I understand that to prove that prior art or a combination of prior art renders a patent obvious, it is necessary to:

- (1) identify the particular references that, singly or in combination, make the patent obvious;
- (2) specifically identify which elements of the patent claim appear in each of the asserted references; and
- (3) explain why a person of ordinary skill in the art would have combined the references, and how they would have done so, to create the inventions claimed in the patent. I further understand that exemplary rationales that may support a conclusion of obviousness include:
 - combining prior art elements according to known methods to yield predictable results;
 - simple substitution of one known element for another to obtain predictable results;
 - use of known technique(s) to improve similar devices (methods or products) in the same way;
 - applying a known technique to a known device (method or product) ready for improvement to yield predictable results;
 - “obvious to try” – choosing from a finite number of identified, predictable solutions with a reasonable expectation of success;

- known work in one field of endeavor may prompt variations of the work for use in either the same field or a different field based on design incentives or other market forces if the variations are predictable to a person of ordinary skill in the art; and
- some teaching, suggestion, or motivation in the prior art that would have led a person of ordinary skill in the art to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

19. I have been informed that, in considering obviousness, hindsight reasoning derived from the patent-at-issue may not be used.

III. TECHNICAL BACKGROUND

20. The '566 patent “pertains to systems and methods enabling parties with little trust or no trust in each other to enter into and enforce agreements conditioned on input from or participation of a third party, over arbitrary distances, without special technical knowledge of the underlying transfer mechanism(s).” (CB-1001, 5:48-53.) The '566 patent includes a glossary of terms and in this glossary the patent provides a description of the term “transfer mechanism”: “A means (e.g., a decentralized digital currency) by which a transaction is created (e.g., by successful submission of a transaction record) and enforced.” (CB-1001, 38:8-11.) And this glossary also describes a “decentralized

digital currency” as a “transfer mechanism (110) comprising a distributed ledger of transactions (often referred to as a ‘block chain’, e.g., with the Bitcoin protocol and progeny) and typically one or more network participants, the network participants comprising one or more miners.” (CB-1001, 36:62-67.)

21. The ’566 patent provides details of one of these existing decentralized currencies, bitcoin, and provides details of the Bitcoin protocol in a lengthy section entitled “Background Art.” (*See* CB-1001, 1:30-5:44.) I agree that the ’566 patent’s admitted prior art section articulates details of the existing Bitcoin protocol which I discuss more fully in the next section.

A. Bitcoin

22. Bitcoin was first disclosed no later than 2009 in the famous Bitcoin White Paper by Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System.” (Nakamoto). I note that the Nakamoto white paper was cited as prior art in the ’566 patent. (CB-1001, 2.) In fact, Figure 24 in the ’566 patent, labeled “Prior Art”, is taken directly from the Nakamoto paper. (*See* CB-1019, 2.) I further note that the ’566 patent does not attribute this Figure to Nakamoto and only describes it as “a simplified chain of ownership in a decentralized digital currency.” (CB-1001, 7:48-49.)

23. Andreas Antonopoulos, a well-known expert on Bitcoin and other digital currencies, published a book entitled “Mastering Bitcoin” on GitHub in

2014. (See CB-1012, 1.) Four chapters of this book were committed to GitHub (published) by May 3, 2014. In my discussion of Bitcoin and blockchain technologies in this section, I rely on the May 3, 2014, version of “Mastering Bitcoin” which I also refer to as Antonopoulos and the admitted prior art discussion in the ’566 patent. I also note that a hardcopy of the “Mastering Bitcoin” book was published by O’Reilly in December 2014 and was published concurrently on GitHub. The December 2014 publication is prior to the filing date of the ’566 patent.

24. Bitcoin is a “collection of concepts and technologies that form the basis of a digital money ecosystem.” (CB-1012, 1.) Currency units “called bitcoins are used to store and transmit value among participants in the bitcoin network.” (CB-1012, 1.) “Bitcoin users communicate with each other using the bitcoin protocol primarily via the Internet” which “can be run on a wide range of computing devices.” (CB-1012, 1.) Bitcoins “are entirely virtual”—“coins are implied in transactions which transfer value from sender to recipient.” (CB-1012, 1.) Bitcoin users “own keys which allow them to prove ownership of transactions in the bitcoin network, unlocking the value to spend it and transfer it to a new recipient.” (CB-1012, 1.) These keys “are stored in a digital wallet on each user’s computer.” (CB-1012, 1.) Possession of the key unlocking a transaction “is the only prerequisite to spending bitcoins.” (CB-1012, 1.)

25. As noted in the '566 patent's admitted prior art, pointing to Figure 24 which is reproduced from Nakamoto's white paper, "[o]wnership of bitcoins implies that a user can spend bitcoins associated with a specific address." (CB-1001, 4:30-33.) To spend bitcoins, "a payer must digitally sign the transaction using the corresponding private key." (CB-1001, 4:33-34.) Without the private key, "the transaction cannot be signed and bitcoins cannot be spent." (CB-1001, 4:34-35.) I discuss Bitcoin transactions further in the next section.

26. "Bitcoin is a fully-distributed, peer-to-peer system." (CB-1012, 1.) The figure below illustrates an exemplary bitcoin system consisting "of users with wallets containing keys, transactions which are propagated across the network and miners who produce (through competitive computation) the consensus blockchain, the authoritative ledger of transactions." (CB-1012, 9.)

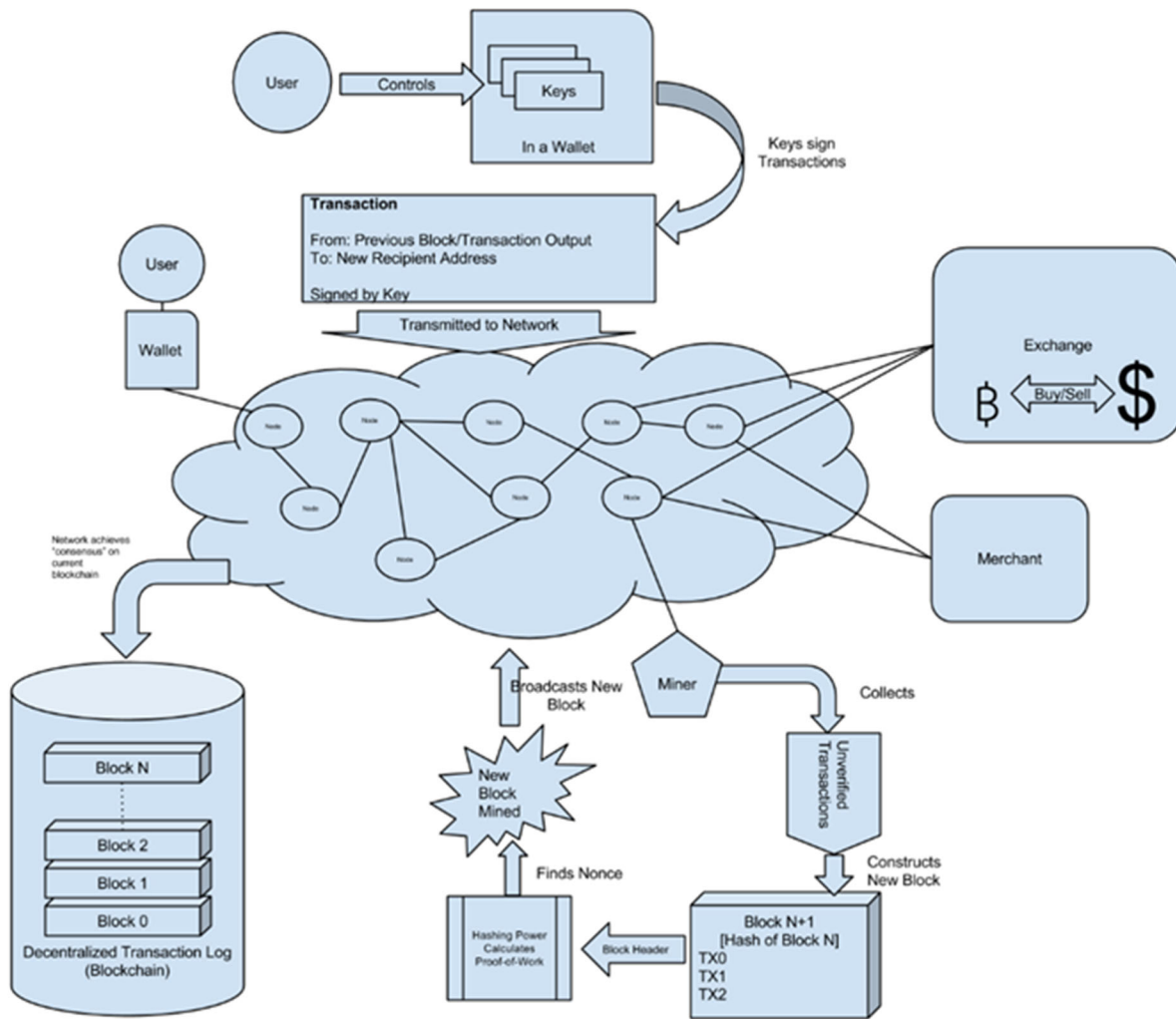


Figure 1

B. Transactions

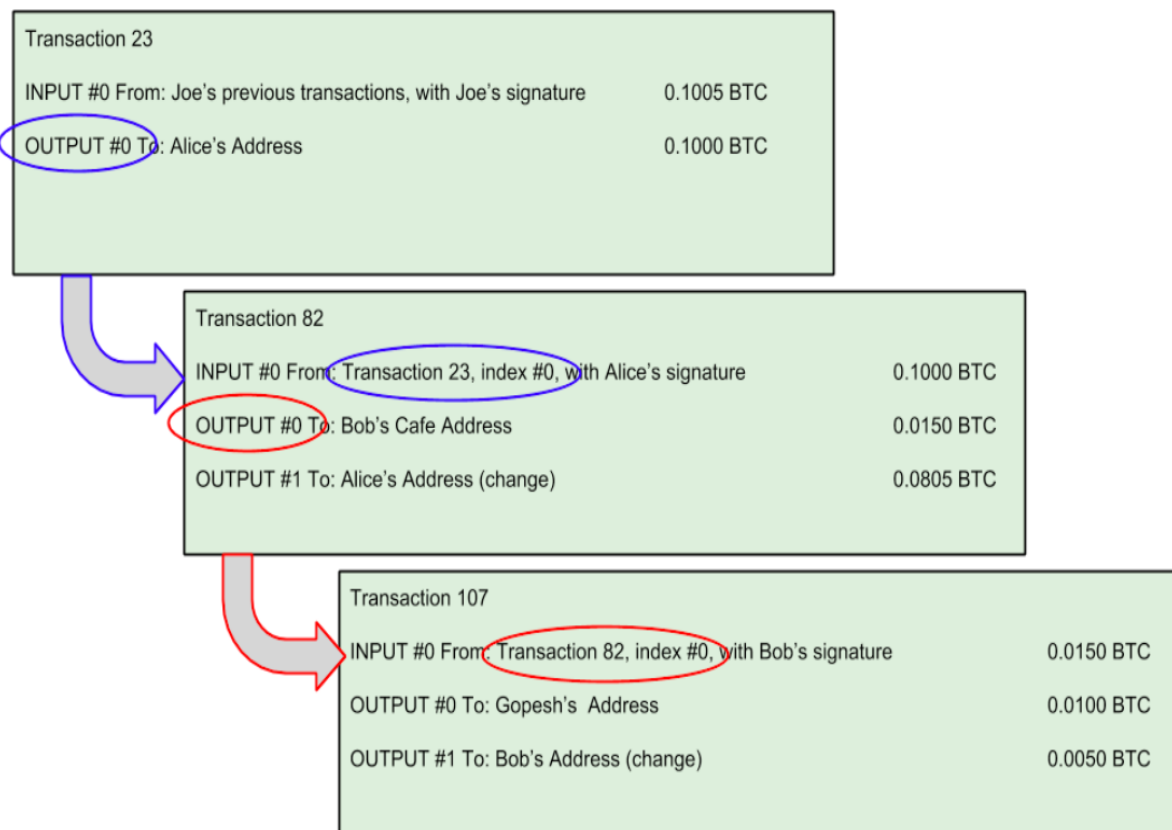
27. Antonopoulos explains that “a transaction tells the network that the owner of a number of bitcoins has authorized the transfer of some of those bitcoins to another owner” who “can now spend these bitcoins by creating another transaction that authorizes transfer to another owner, and so on, in a chain of ownership.” (CB-1012, 11.)

28. The admitted prior art of the '566 patent describes that a “transaction comprises at least one input and at least one output.” (CB-1001, 2:51-52.) In simple terms, the inputs “are debits against a bitcoin account” and outputs are “credits added to a bitcoin account.” (CB-1012, 11.) The inputs and outputs “do not necessarily add up to the same amount.” (CB-1012, 11.) Instead, outputs add up to slightly less than the inputs with the difference representing a small fee payment collected by a miner that includes the transaction in the ledger. (CB-1012, 11.) I discuss the “ledger” and mining generally in the next section.

29. Outputs can be claimed either by single signature or by multiple signatures. If an output can be claimed by providing a single signature, it is referred to as a “singlesig” output. In contrast, a multiple signature or “multisig” output requires a set of n public keys and requires m -of- n (with $m \leq n$) valid signatures from different public keys in the set to be valid. For example, in a 2-of-2 output, two parties (e.g., Party A and Party B) must sign transactions spending the output.

30. In Bitcoin, “spending” is “signing the value of a previous transaction for which you have the keys over to a new owner identified by a bitcoin address.” (CB-1012, 12.) A transaction output is considered “spent” if it is associated with a valid child transaction and “unspent” if it is not associated with a valid child transaction. (CB-1001, 2:61-66.) For a transaction to be valid, “every input must be an unspent output of a previous transaction.” (CB-1001, 4:45-46.) The following

Figure from Antonopoulos illustrates a transaction chain. Transaction 23 has an input of 0.1005 BTC from one of Joe's "previous transactions" and outputs 0.1000 BTC to Alice's address. In Transaction 82, the input is the output of transaction 23 with Alice's signature. In this transaction, the output is split with a portion going to the address associated with Bob's Café and the remainder (the "change") going back to Alice's address. The output of Transaction 82 is subsequently spent by Bob in an input including Bob's signature.



Antonopoulos, Figure 4

31. A transaction can also have a “lock time.” (CB-1001, 4:57.) The lock time “prevents the transaction from being accepted as valid and allows the transaction to be pending and replaceable until an agreed-upon future time.” (CB-1001, 4:58-60.) A transaction “will not be accepted for inclusion in the block chain until the transaction’s lock time has been reached.” (CB-1001, 4:63-65.)

C. Blockchain Technology

32. After a transaction is created, “the transaction must be transmitted to the bitcoin network where it will become part of the distributed ledger, the blockchain.” (CB-1012, 16.) The blockchain “is a public ledger that records bitcoin transactions.” (CB-1001, 3:63-64.) A transaction “does not become part of the shared ledger (the *blockchain*) until it is verified and included in a block by a process called *mining*.” (CB-1012, 17 (emphasis in original).) Each “block” is a “group of transactions” that is “chained” to the previous block using cryptographic techniques. (CB-1001, 5:1-6; *see also*, CB-1012, 17 (“Transactions are bundled into *blocks* which require an enormous amount of computation to prove, but only a small amount of computation to verify as proven”).) “Every ten minutes, miners generate a new block, which contains all the transactions since the last block.” (CB-1012, 17.) Antonopoulos explains that mining can be thought of “like a giant competitive game of sudoku that resets every time someone finds a solution and

whose difficulty automatically adjusts so that it takes approximately 10 minutes to find a solution.” (*Id.*)

33. Blocks must contain a “proof-of-work” from a bitcoin miner. (CB-1001, 5:9-10.) The “proof-of-work” contains a “difficulty target” and a “nonce” value, and bitcoin miners solve mathematical problems to calculate a hash value that is smaller than the number provided in the difficulty target. (CB-1001, 5:9-14.) When a “miner has discovered a solution to the Proof-of-Work algorithm for this block”, the transactions “within the block will be accepted as ‘confirmed’ by the network and can be spent.” (CB-1012, 7.)

D. Cryptography and Bitcoin

34. Cryptographic tools such as digital signatures are core components of digital currencies. In general, an encryption algorithm is a mathematical function used for encryption and decryption. There are two general types of cryptographic algorithms—symmetric and public-key algorithms. (CB-1013, 4-5) Symmetric algorithms use the same key to encrypt and decrypt data. (CB-1013, 4.) Asymmetric or public-key algorithms use two keys that are mathematically related. (CB-1013, 464.) One key is kept private (the private key), and the other key is distributed publicly (the public key). (CB-1013, 4-5.) With asymmetric cryptography, the sender of a message uses the recipient’s public key to encrypt the message and the recipient uses its private key to decrypt the message. (*Id.*)

35. In Bitcoin, a “bitcoin wallet contains a collection of key pairs, each consisting of a private key and a public key.” (CB-1012, 40.) The private key is “used to create a corresponding public key.” (CB-1012, 40.) “The public key can then be converted into a bitcoin address, which is shared with anyone who we want to send us bitcoin.” (CB-1012, 40.) As I mentioned above, “[o]wnership and control over the private key is the root of user control over all funds associated with the corresponding bitcoin address.” (CB-1012, 40.)

36. Antonopoulos explains that in Bitcoin, “[p]ublic-key cryptography is like a digital padlock which can only be opened by the owner of a secret, also known as a private key.” (CB-1012, 46.) The owner of the key “can hand out as many copies of the padlocks as they want, and others can use it to ‘lock’ bitcoins inside transactions recorded on the blockchain.” (CB-1012, 46.) Only the private key’s owner “can then create a signature to unlock and ‘redeem’ these transactions, as only they can open the digital padlock.” (CB-1012, 46.)

37. Digital signatures are generally used to protect the integrity of a message or data within a message. (CB-1013, 40.) To create a digital signature, a message originator computes the message digest for the message (or a portion of the message) and encrypts the message digest with the originator’s private key. (CB-1013, 38.) A technique used to create a message digest is a one-way hash algorithm. (*Id.*) The recipient of a digital signature verifies the digital signature by

decrypting the signature using the public key of the message originator to recover the message digest. (*Id.*) The recipient separately computes the digest of the received message and compares it to the decrypted version. (*Id.*) If the message has not been altered, the calculated digest and recovered digest will be the same. (*Id.*)

IV. THE '566 PATENT

A. Summary of the '566 Patent

38. The '566 Patent describes “systems and methods enabling parties with little trust or no trust in each other to enter into and enforce agreements conditioned on input from or participation of a third party.” (CB-1001, 5:48-55.) To illustrate these agreements the '566 patent “explores example embodiments enabling two forms of value transfer: *arbitrary swaps and L/Cs*,” with L/C representing a letter of credit. (CB-1001, 5:60-51.)

1. Exemplary Embodiments

a. Letter of Credit Embodiment

39. The '566 patent states that a “letter of credit (L/C) is well known in the art, but is fundamentally an agreement where a third party transfers assets to a second party on behalf of a first party upon some agreed [upon] condition being demonstrated as met before an agreed upon time (an expiration).” (CB-1001, 25:24-28.)

40. In one example, the '566 patent “depicts aspects of one embodiment pertaining to a L/C comprising a commit transaction and an expiration transaction, the expiration transaction being analogous to the refund transaction”. The commit transaction is analogous to (T1) in Hearn that is setup in steps 1 and 2, and broadcast in step 6. (CB-1009, Example 1.) “However, where a refund transaction is meant exclusively for recovery of funds in the event of an exception (e.g., the facilitator becomes unavailable to create or sign a disbursement transaction record), use of an expiration transaction, in addition to recovery, is contemplated by the offer (e.g., if the conditions set forth are not satisfied before the expiration timestamp, despite the facilitator being operational)”. (CB-1001, 26:7-18.) This transaction is analogous to Hearn’s (T2) that enables a refund described in steps 3 and 4 of example 7. (CB-1009, Example 7.) Furthermore, the commit transaction “comprises a first input for accepting a first amount from a first source transaction (i.e., from a first party), and one or more outputs for directing a portion of the first amount *to one or more other transactions (not depicted)*. The one or more transactions that are not depicted correspond to Hearn’s (T3) illustrated in step 9 as an initial transaction and updated transactions for incrementing payment amounts. (CB-1009, Example 7.)

41. FIG. 9 of the '566 patent describes “aspects of one embodiment pertaining to a L/C comprising a commit transaction and an expiration transaction,

the expiration transaction being analogous to the refund transaction in previously described embodiments. Although FIG. 9 contemplates a different use of the expiration transaction than the refund transaction, the '566 patent states that the “two function almost identically within the invention.” (CB-1001, 26:6-20.)

42. In another example, the '566 patent describes an embodiment where “the transacting parties agree on a third party to act as a mediator in a dispute. For example, if the facilitator becomes unavailable, rather than electing to invoke a refund, one party triggers a dispute whereby a mediator stands in place of the unavailable facilitator.” (CB-1001, 33:50-55.) This type of transaction is also illustrated by Hearn, where a client and a merchant agree on a mediator to mediate a dispute. (CB-1009, Example 2.)

b. Swap Embodiment

43. In yet another example, the '566 patent describes a swap transaction. In particular, the '566 patent describes the “major phases effecting a value transfer in the form of a swap.” (CB-1001, 35:27-30.) This type of transfer uses “a transfer mechanism comprising a decentralized digital currency comprising a block chain.” (CB-1001, 35:29-30.)

44. In a **first phase**, the client “validates a first order comprising terms (e.g., base instrument, quote instrument, principal, collateral, disbursement function, expiration timestamp, etc.).” (CB-1001, 35:30-34.) The client then

“submits (broadcasts) a first principal transaction record conforming to the terms to the transfer mechanism to create a first principal transaction.” (CB-1001, 35:34-37.)

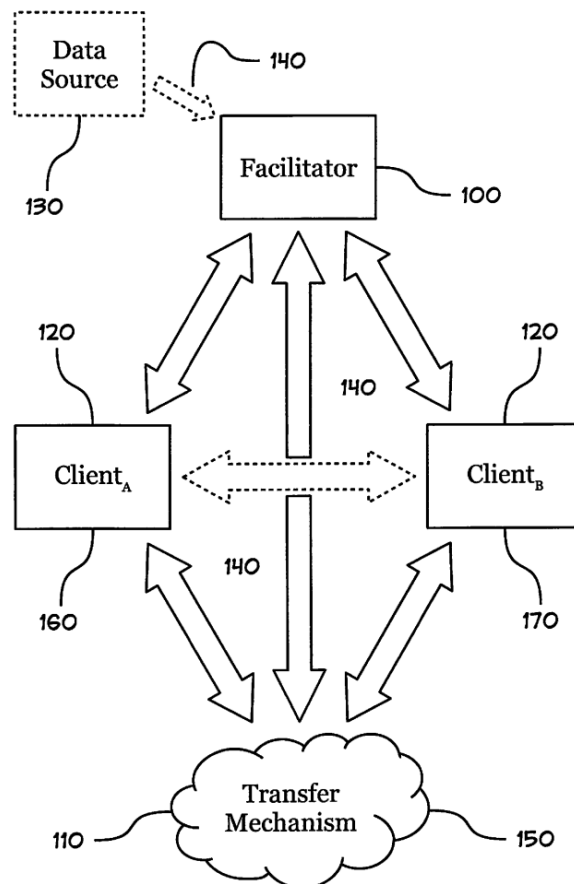
45. In the **second** phase, the outputs from the first principal transaction are committed “by creating and submitting (broadcasting) a commit transaction record to the transfer mechanism to create a commit transaction.” (CB-1001, 35:39-45.) Optionally, “a refund or ‘rollback’ transaction record” is created and made available for the client “that spends the outputs from the commit transaction, but cannot be used until well after the expiration timestamp.” (CB-1001, 35:45-48.) In this way, “[i]f the facilitator fails catastrophically,” the client can “sign and submit the refund transaction record to place” the client back in its original position. (CB-1001, 35:48-51.)

46. In the **third** phase, a value is received from a data source and the valuation is monitored “by applying a disbursement function to the value[]”, the principal, and any collateral to check whether the principal and any collateral of any one party is exhausted.” (CB-1001, 35:52-56.) In a final phase, “after the expiration timestamp (or if the principal and any collateral of any party is exhausted, whichever is sooner),” an inchoate disbursement transaction “that spends the commit transaction’s output(s)” is created and signed. (CB-1001, 35:59-66.) When the inchoate disbursement transaction is completed, the “client submits

(broadcasts) the complete disbursement transaction record to the transfer mechanism to create the disbursement transaction.” (CB-1001, 35:66-36:4.)

2. Network Environment

47. Furthermore, the '566 patent depicts an exemplary network environment for implementing its agreements in Figure 1 (below). The network includes a transfer mechanism (110) such as a decentralized digital currency (150), clients (120, 160, 170), facilitator (100), and data source (130) connected by a computer network (140). (CB-1001, 7:9-14, 8:49-49.)



'566 Patent, Figure 1

B. The '566-Provisional does not support all the limitations of the independent claims

48. The '566 Patent issued on December 7, 2021, based on a patent application filed May 5, 2015, and claims priority to a provisional patent application filed May 9, 2014, which I refer to herein as the "'566 provisional."

49. I have been informed by counsel that a claim in a U.S. application is entitled to the benefit of the filing date of an earlier U.S. application if the subject matter of the claim is disclosed in a manner that is sufficient to meet the written description requirement provided by 35 U.S.C. §112(a) in the earlier filed application.

50. I have been informed by counsel that in order to satisfy the "written description" requirement of 35 U.S.C. §112(a), the disclosure of the prior application must convey with reasonable clarity to those skilled in the art that, as of the filing date sought, the inventor was in possession of the invention.

51. It is my opinion that the '566-Prov does not provide written description support for at least the limitation that "*at least one of the computing device, the first client device, or the second client device verifies the recording of*

*the complete data record in the distributed ledger by observing an external state*¹” ([1G]/[7G]).

52. The ’566-Prov does not explicitly disclose verification by any network device of “*the recording of the complete data record in the distributed ledger by observing **an external state***”² or “***verify[ing]** the recording of the complete data record in the distributed ledger.*” The provisional mentions “external state” twice:

“In one embodiment comprising a swap, a first client and a second client participate in a series of transactions where assets from a first party and assets from a second party are committed until a combination of two of the first party, the second party, and an intermediary release them in accordance with a calculation by the intermediary based on observation of external state, such as the relative value of certain assets at a specific time.

In another embodiment comprising an L/C, a first client and a second client participate in a series of transactions where assets from a first party are committed until either the first client or an intermediary releases them based on observation of external state, such as verification of delivery to a shipper or an address.”

(CB-1005, ¶¶17-18.)

¹ Limitations of the challenged claims are indicated by italics throughout.

² Unless otherwise indicated all emphasis added.

These two mentions of “external state” are completely unrelated to verification of the recording of a data record in a distributed ledger. Both examples describe transactions in which assets committed to a transaction are **released** based on observation of an external state such as “the relative value of certain assets at a certain time” or “verification of delivery to a shipper or an address.” (CB-1005, ¶¶17-18.)

C. Claim Construction

53. I understand that the Patent Office applies the claim construction standard used in district courts which seeks to give claim terms their plain and ordinary meaning to a person of ordinary skill in the art at the time of the claimed invention. I also understand that if the patent applicant gave a term a special meaning in the specification or during prosecution, then the special meaning should be used. My analysis uses the plain and ordinary meaning to each of the claim terms.

D. Level of Ordinary Skill in the Art

54. I understand that certain issues relating to the validity of the '566 patent must be judged from the perspective of a person of ordinary skill in the relevant art, as I discuss below. I have been asked to define the level of a “person of ordinary skill in the art” or “POSITA” that someone would have had at the time the alleged invention as claimed was made. In §IV.A, I provided an overview of

the '566 patent. In deciding the level of ordinary skill, I have considered the following factors, which I have been informed are relevant to the determination of the level of ordinary skill in the art:

- the levels of education and experience of persons working in the field;
- the types of problems encountered in the art;
- the prior art solutions to these problems;
- the rapidity with which innovations are made; and
- the sophistication of the technology.

55. In my opinion, a person having ordinary skill in the art ("POSITA") would possess either (i) a Bachelor of Science degree in computer science, computer engineering, electrical engineering, or mathematics, or equivalent and 2-3 years of experience in implementation, programming, or design of cryptocurrencies or blockchain technologies; or (ii) a doctoral degree in computer science, computer engineering, electrical engineering, or mathematics, or equivalent with experience in cryptography including the study, design, or implementation thereof for use in computer systems.

**V. GROUND 1: COMBINATION OF HEARN AND ARMSTRONG
RENDERS CLAIMS 1-3 OBVIOUS**

A. Overview of Hearn

56. The paper “Contracts” by Mike Hearn is an extremely well-known paper. Hearn describes a set of “distributed contract[s]” which Hearn defines as “method[s] of using Bitcoin to form agreements with people via the block chain.” (CB-1009, 1.) Hearn notes that “[C]ontracts don't make anything possible that was previously impossible, but rather, they allow you to solve common problems in a way that minimizes trust.” (CB-1009, 1.) That is, Hearn teaches that a distributed contract allows an agreement to be encoded using Bitcoin transactions which guarantee an action agreed upon by the parties to the contract is performed.

57. Hearn provides 8 example distributed contracts that can be implemented using the Bitcoin blockchain. I focus my analysis in this declaration on “Example 7” which Hearn titles “Rapidly adjusted (micro)payments to a pre-determined party.” (CB-1009, 5.)

58. A micropayment is a small payment between parties. Micropayments were not (and remain) impractical to implement with traditional credit or debit card payment systems. In these systems, each transaction incurs a transaction fee. Therefore, the benefit of using a credit/debit card decreases when the transaction fee becomes a significant percentage of the payment. Additionally, these systems

further incur costs to mitigate fraud in transactions where there is no or limited trust (e.g., between individuals). Because of these transaction fees and costs associated with mitigating fraud in trustless transaction, it has not been cost-effective to use existing credit/debit cards for such small payments. Hearn's micropayment protocol of Example 7 overcomes these concerns. The micropayment protocol allows a party (e.g., a server) to accept metered payments from another party over the duration of the service while minimizing transaction costs. In Example 7, Hearn uses an exemplary application of the purchase of WiFi service from an untrusted café. Specifically, Hearn articulates that a Party A (client) wishes "to pay 0.001 BTC per 10 kilobytes of usage" of an untrusted café's WiFi hotspot "without opening an account with the café." (CB-1009, 5.) Hearn's micropayment protocol allows Party A (client) to periodically pay the untrusted server (Party B) as it uses the WiFi service. A POSITA would understand that micropayments are applicable to many other service offerings.

59. Hearn's micropayment protocol works in two stages. In the first stage, a micropayment channel is established between the parties to the contract which Hearn refers to as the client and server. In my discussion I refer to the client as Party A and the server as Party B as well. During channel establishment, Party A (client) and Party B (server) place in escrow a maximum expected amount to be paid for the micropayment contract. That amount is then "locked" for the duration

of the contract (CB-1009, 5 (steps 1-2 and 6).) In the second stage, Party A (client) makes small usage-based payments from the escrow which increment a total to be paid to Party B (server) for the service. (CB-1009, 5 (steps 7-9).) Despite receiving a series of small payments, Party B (server) only broadcasts the final payment transaction and only when the contract has expired. Through this technique only two transactions (the “escrow” and final payment transactions) are broadcast, minimizing the total Bitcoin transactions fees. I describe these two stages in further detail below.

STAGE 1: ESTABLISHING THE CHANNEL

60. Hearn’s protocol starts with the creation and/or gathering of the necessary cryptographic keys needed for the Bitcoin transaction. Specifically, in step 1, Party A (client) creates a public key (K1) and requests a public key (K2) from the server. (CB-1009, 5 (step 1)). As I discussed in the Background section, each public key (e.g., K1 and K2) has a corresponding private key that is used in the digital signature operations required by later steps in the protocol.

61. After creation of a public key for Party A (client) and receipt of a public key for Party B (step 1), Party A (client) creates and signs a “transaction (T1) that sets up a payment of (for example) 10 BTC to an output requiring both the server’s public key and one of [the client’s own keys] to be used.” (CB-1009,

5 (step 2).) A POSITA would have understood that the digital signature is created using Party A's (client's) private key corresponding to its public key, K1. Hearn discloses that a "good way to do this is use the OP_CHECKMULTISIG" capability, which is an opcode that enables verifying multiple signatures for a transaction. (CB-1009, 2, 5.)

62. The CHECKMULTISIG opcode instructs the blockchain to verify multiple signatures within a verification script of a transaction. For example, the "opcode provides n-of-m checking: you provide multiple public keys, and specify the number of valid signatures that must be present." (CB-1009, 1-2.) For example, when the CHECKMULTISIG opcode is used for 2-of-3 checking, the blockchain is instructed to verify 2 out of 3 signatures are provided.

63. Transaction (T1) acts as an "escrow" transaction transferring the maximum anticipated total value to be paid for the contract (e.g., 10 BTC) **from** the output funds of a previous transaction owned by A **to** an "escrow" account owned by both Party A (client) and Party B (server). I have illustrated Transaction 1 in Figure A below. The "escrow" transaction essentially funds the channel. Any transactions spending the bitcoin of this "escrow" account must be signed by both Party A and Party B.

Transaction ID (T1)

INPUTS From:

previous transaction received by Party A 10 BTC
with Party A's DIGITAL SIGNATURE

OUTPUTS To:

2 K1 K2 2 CHECKMULTISIG 10 BTC

Figure A

64. Next, in step 3, Party A (client) creates a refund transaction (T2) “that is connected to the output of T1 which sends all the money back” to Party A (client). (CB-1009, 5 (step 3).) When Hearn states that T2 “is connected to the output of T1”, Hearn means that the input of refund transaction T2 is the output of the “escrow” transaction (T1). “Send[ing] all the money back” to Party A means that T2 refunds the total amount of the contract set forth in T1 to Party A (client). I illustrate transaction T2 in Figure B below. As shown in this figure, the refund transaction (T2) spends the multisig output of T1 and transfers the total value (e.g., 10 BTC) to an address associated with Party A (client). Hearn teaches that Transaction (T2) also “has a time lock set for some time in the future, for instance a few hours.” (CB-1009, 5 (steps 3-5).) Thus, (T2) is time locked to ensure that the transaction cannot be accepted into the blockchain until a specified time window

has expired. (CB-1009, 5.) In other words, Party A (client) cannot redeem the refund until after the lock time has expired. Ultimately, refund transaction (T2) allows Party A (client) to reclaim its committed bitcoin if Party B becomes unresponsive or unavailable and the bitcoin is stuck “in limbo”. (CB-1009, 5 (“The refund transaction is needed to handle the case where the server disappears or halts at any point, leaving the allocated value in limbo”).)

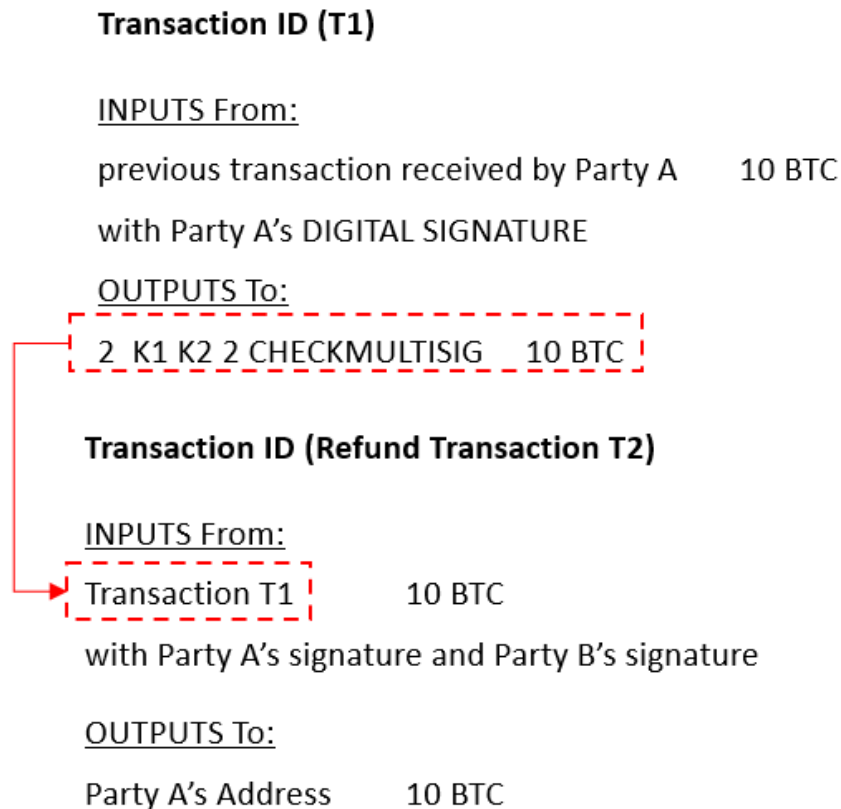


Figure B

65. Party A (client) does not sign the refund transaction. Instead, Party A (client) sends the refund transaction (T2) to Party B (server) which signs the refund

transaction (step 4) and returns the signature to Party A (client). (CB-1009, 5 (steps 3-4).) I note that the February 9, 2014 version of Hearn (CB-1009) includes a typographical error in step 4. Specifically, Hearn states that the “server signs T2 using its public key K2.” (CB-1009, 5.) A POSITA would have understood that the server is signing using its private key as was well-known and understood in Bitcoin transactions as I discussed in the technology background section. I note that later versions of Hearn have corrected this error. (CB-1018, 5.)

66. Party A (client) verifies the signature using the Party B’s public key (step 5). In the technology background section, I discussed the technique used to determine whether a received signature is valid. If the signature is valid, Party A (client) signs the “escrow” transaction (T1) and “passes the signature to” Party B (server). (CB-1009, 5 (step 6).) The server then “broadcasts the transaction” (T1). (CB-1009, 5 (step 6).) As discussed in §III.B, “broadcasting” transaction (T1) commits (T1) to the blockchain which “locks in the money.” (CB-1009, 5.) The broadcasting of (T1) also indicates that Party A (client) that Party A can start making incremental payments to Party B (server) for the service. That is, the payment channel has been established between Party A and Party B.

STAGE 2: MAKING PAYMENTS

67. As Party B (server) provides the service to Party A (client), Party A issues a series of transactions transferring more and more of the amount in the “escrow” account to Party B and less and less to itself. (*See Hearn, 8.*) Specifically, Party A creates an initial payment transaction (T3) “which connects to T1 like the refund transaction and has two outputs” – one goes to Party A and the other goes to Party B. (CB-1009, 5 (step 7).) I note that Hearn sometimes refers to the parties by their respective public keys. For example, K1 refers to Party A (client) and K2 refers to Party B (server). In this initial payment transaction, all value of (T1) is “allocated to the first output (K1), i.e., it does the same thing as the refund transaction but is not time locked.” (CB-1009, 5 (step 7).) Party A (client) “signs T3 and provides the signature” to Party B (server). (CB-1009, 5 (step 7).)

68. Party A (client) then provides metered payments to Party B (server). Specifically, Party A (client) adjusts its copy of T3 to allocate more value to Party B’s output (K2) and less to its own (K1). (CB-1009, 5 (step 9).) Party A (client) then sends each signed transaction to Party B (server). For example, Party A (client) may send an updated transaction after a pre-determined amount of usage (e.g., 100 kilobytes) or after a pre-determined amount of time. Party A (client) “re-signs the new T3 and sends the signature to the server.” (CB-1009, 5.) Only the signature

and the amount to increment (T3) is required to be sent to the server. (CB-1009, 5.) The server then “adjusts its copy of T3 to match the new amounts, verifies the signature and continues.” (CB-1009, 5 (step 9).) These transactions are not broadcast. Therefore, the most that can be lost by Party A (client) if the server stops providing service is one incremental payment. However, if Party B (server) breaks down or otherwise disappears, Party A can broadcast the refund transaction (T2) and get all the bitcoin back after the lock time has expired.

69. Because all the updated payment transactions (T3) spend the same output, only one version can be committed to the blockchain. Party B (server) therefore broadcasts only one of the T3 transactions—the final payment update of T3 which allocates the maximum amount of the “escrow” transaction to Party B, effectively closing the payment channel between the parties. The payment channel may be considered closed when the session ends (e.g., both parties agree to close the channel) or a pre-determined period is close to expiring. (CB-1009, 5 (“This continues until the session ends, or the 1-day period is getting close to expiry. The AP then signs and broadcasts the last transaction it saw, allocating the final amount to itself”).)

* * *

70. In Hearn’s micropayment protocol, only two transactions are broadcast to the Bitcoin network – the “escrow” transaction (T1) and the final

payment transaction (T3). This protocol minimizes transaction fees because fees are only paid for two transactions rather than for every micropayment between Party A (client) and Party B (server).

B. Overview of Armstrong

71. Armstrong relates to a “system and method for transacting bitcoin” such as the Bitcoin transactions used in Hearn’s Example 7. (CB-1006, ¶3; CB-1007, ¶1; CB-1008, ¶1.) Armstrong claims priority to a series of provisional applications including Application 61/954,434 which I refer to as “Armstrong-Prov1” (CB-1007) and Application 61/990,017 which I refer to as “Armstrong-Prov2” (CB-1008). Armstrong-Prov1 and Armstrong-Prov2 have overlapping disclosures with the materials that I rely on from Armstrong in my analysis of Ground 1. Therefore, these provisionals fully support the Armstrong citations in my analysis. Throughout this declaration, I cite to Armstrong and provide the supporting cites from Armstrong-Prov1 and Armstrong-Prov2 to demonstrate support for the subject matter of the cites. Additionally, in §IV.B.2, I show that at least one claim of Armstrong is supported by both provisionals.

1. Armstrong

72. Armstrong’s Figure 1A illustrates an exemplary network environment 10 for processing Bitcoin transactions. Network environment 10 includes Bitcoin network 12, first host computer system 14, second host computer system 16, first

user device 18, and second user device 20 connected over the Internet 22 to first host computer system 14. (CB-1006, ¶81; CB-1007, Figure 1A; CB-1008, ¶¶47-48, Figure 1A.)

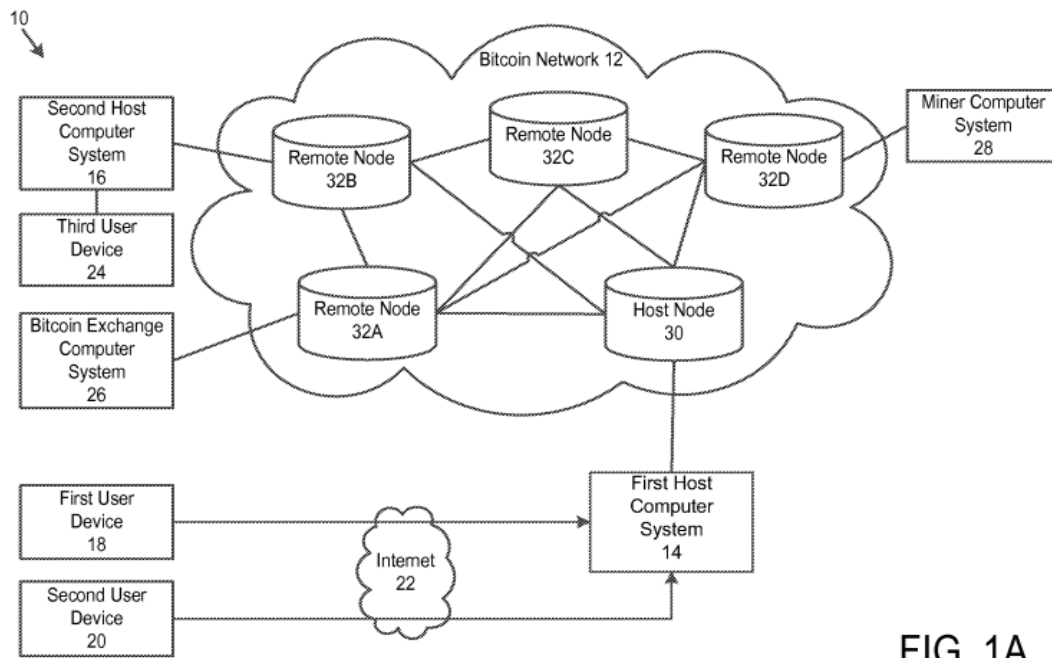


FIG. 1A

Armstrong, Figure 1A

73. First host computer system 14 carries out “a method for transacting bitcoin” on a Bitcoin network. (CB-1006, ¶3; CB-1007, ¶1; CB-1008, ¶1.) A host computer, like first host computer 14 and second host computer 16, “host[s] accounts or ‘wallets’ that allow users to make and accept payments using bitcoin.” (CB-1006, ¶8; CB-1007, ¶5; CB-1008, ¶5.) That is, a single host computer is configured to host a number of wallets for the same or different clients.

74. Different types of transfers are envisioned by Armstrong. One type of transfer is between two wallets hosted on the same host computer. These transfers are not recorded on the blockchain and “do not need block chain verification outside of the first host computer system 14.” (CB-1006, ¶102; CB-1007, ¶53; CB-1008, ¶68.)

75. Another type of transaction is between wallets on different host computers. For example, a transaction (Transfer 7) between a wallet held by a first host computer system 14 and a wallet held by second host computer system 16 (e.g., Wallet C) is illustrated in Figure 45 below. In this example, a user enters a bitcoin address (Bitcoin address 6) “located in the third wallet (Wallet C) and an amount of bitcoin to be transferred to the third wallet (Wallet C).” (CB-1006, ¶103; CB-1007, ¶54; CB-1008, ¶69.) The transfer (shaded red) “is recorded within the second wallet (Wallet B).” (CB-1006, ¶103; CB-1007, ¶54; CB-1008, ¶69.) In this case, the control of the bitcoin changes from first host computer 14 to second host computer 16. In addition, the transaction is broadcast to the bitcoin network where “a miner’s fee is charged for the transfer and has to be paid by the first host computer 14 to the miner computer system 28.” (CB-1006, ¶103; CB-1007, ¶54; CB-1008, ¶69.) That is, first host computer 14 and second host computer 16 perform functions of Bitcoin network nodes such that transactions between a wallet

on one host computer and a wallet on the other host computer are broadcast to the Bitcoin network.

	First Host Computer System 14			Second Host Computer System 16	
	Wallet A		Wallet B	Wallet C	
	Email Address A; Login Details		Email Address B; Login Details		
Transfer 1	Bitcoin address 1				
		Amount in bitcoin			
	Bitcoin address 2				
Transfer 2		From: Network address			
		Amount in bitcoin			
		Host Fee: 0 bitcoin			
Transfer 3	Bitcoin address 2		Bitcoin address 3		
		To: Email address B	From: Wallet A		
		Amount in bitcoin	Amount in bitcoin		
		Miner's Fee = 0 bitcoin			
		Host Fee: 0 bitcoin			
Transfer 4			Bitcoin address 4		
			From: First Host		
			Amount in bitcoin		
			Host Fee: x%		
Transfer 5	Bitcoin address 5		Bitcoin address 4		
		From: Wallet B	To: Email address A		
		Amount in bitcoin	Amount in bitcoin		
			Miner's Fee = 0 bitcoin		
			Host Fee: 0 bitcoin		
Transfer 6	Bitcoin address 5		Bitcoin address 4		
		From: Wallet B	To: Email address A		
		Amount in bitcoin	Amount in bitcoin		
			Miner's Fee = 0 bitcoin		
			Host Fee: 0 bitcoin		
Transfer 7			Bitcoin address 4	Bitcoin address 6	
			To: Bitcoin Address 6		From: Wallet B (First Host)
			Amount in bitcoin		Amount in bitcoin
			Miner's Fee = 001 Bitcoin		
			Host Fee: 0 bitcoin		

Armstrong, Annotated Figure 45

76. Another type of transaction discussed by Armstrong is a transaction where “bitcoin is sent to one of the nodes, e.g., node 32C,” which “could be a fourth user device which is owned by the recipient of the bitcoin transfer having its own bitcoin address.” (CB-1006, ¶106; CB-1007, ¶57; CB-1008, ¶72.) That is, first host

computer 14 may be used as a node on the Bitcoin network to send bitcoin to any other node on the Bitcoin network, including second host computer 16 or node 32C.

77. To initiate Bitcoin transactions like the transaction in Figure 45 above, an application on first user device 18, illustrated in Figure 1B, “transmits a user interface request over the Internet 22 to the website 34.” (CB-1006, ¶85; CB-1007, ¶36; CB-1008, ¶51.) Website 34 “responds to the user interface request by transmitting the user interface 36 over the Internet 22 to the first user device 18.” (CB-1006, ¶85; CB-1007, ¶36; CB-1008, ¶51.) User interface 36 provides fields to permit a user to enter information regarding the bitcoin transaction (e.g., email address of party, amount of bitcoin, etc.). (CB-1006, ¶86; CB-1007, ¶37; CB-1008, ¶52.) After receiving the transaction information, the first host computer (e.g., host computer system 14) then processes the Bitcoin transaction on behalf of the user.

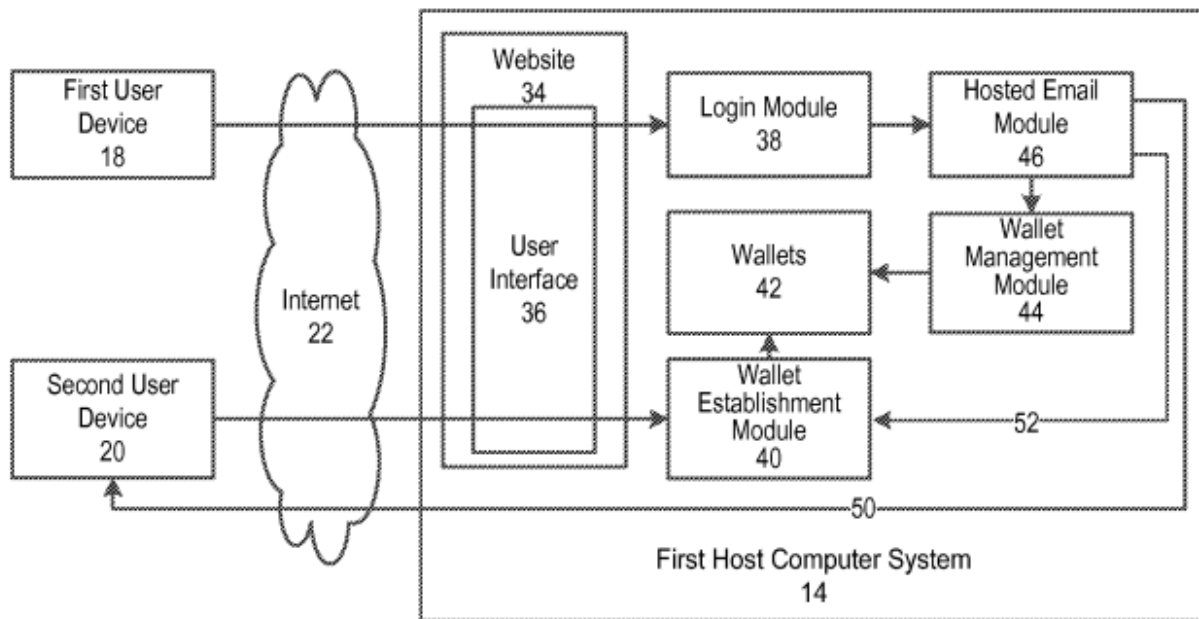


FIG. 1B

Armstrong, Figure 1B

2. Armstrong-Prov1 and Armstrong-Prov2 Support at Least One Claim of Armstrong

78. The inventor was in possession of the subject matter claimed in at least claim 1 of the '566 patent as of the filing date of Armstrong-Prov1 and Armstrong-Prov2 as I set forth in the following table.

Armstrong - Claim 1	Support
1. A method of effecting payment comprising:	CB-1007, ¶¶9-11 CB-1008, ¶¶9-11

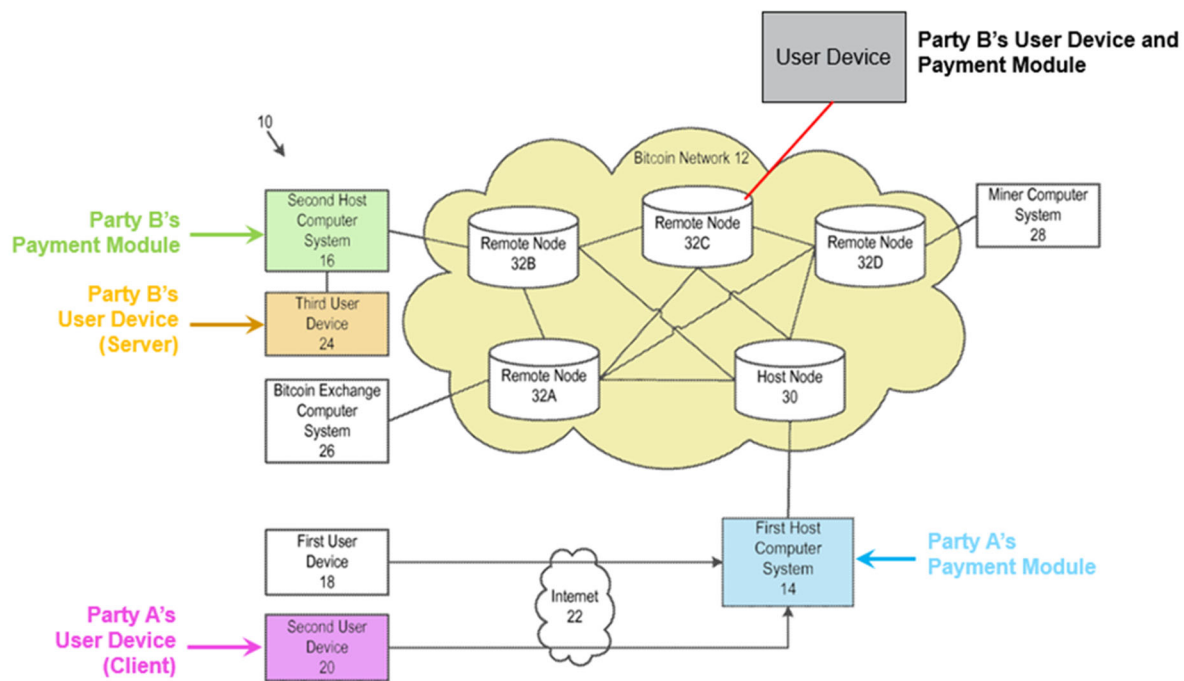
receiving, by a host computer system, a request for payment from a merchant computer system, including an amount in a currency;	CB-1008, ¶¶9-11, Figure 4 CB-1007, ¶41; Figure 7 CB-1007, ¶¶34-36, 38 CB-1008, ¶¶9-11, Figure 4 CB-1008, ¶56; Figure 7 CB-1008, ¶¶49-50, 52
determining, by the host computer system, a first exchange rate, wherein the first exchange rate fluctuates and the first exchange rate is determined at a first moment in time;	CB-1007, ¶37 CB-1007, ¶52, Figure 42 CB-1008, ¶52 CB-1008, 67; Figure 42
converting, by the host computer system, the amount in the currency to an amount in bitcoin using the first exchange rate at the first moment in time;	CB-1007, ¶37 CB-1007, 52; Figure 42 CB-1008, ¶52 CB-1008, 67; Figure 42
receiving, by the host computer system, a send instruction from the customer computer system, wherein the send instruction is at a second moment in time later than the first moment in time and the exchange rate at the second moment in time is a second exchange rate that is different than the first exchange rate at the first moment in time;	CB-1007, ¶37, 39, 41; Figure 4 CB-1008, ¶52, 54, 56; Figure 4
receiving, by the host computer system, payment in bitcoin from the customer in an amount that is based on the amount in bitcoin; and	CB-1007, ¶38, ¶41, ¶¶52-53; Figure 3, Figure 42

	CB-1008, ¶¶53, ¶56, ¶¶67-68; Figure 3, Figure 42
transmitting, by the host computer system, in response to receiving the send instruction from the customer computer system, a payment instruction to pay currency to the merchant,	CB-1007, ¶¶41-43; Figures 34-38 CB-1008, ¶¶56-58; Figures 16-19, 34-38
wherein the currency paid to the merchant is for an amount that is at least in part based on the amount in the currency that is converted to bitcoin at the first moment in time even though the exchange rate is different at the second moment in time.	CB-1007, ¶¶41-43; Figures 34-38 CB-1008, ¶¶56-58; Figures 16-19, 34-38

C. Combination of Hearn and Armstrong

79. A POSITA would have been motivated to implement Hearn's distributed contracts, including Example 7, in Armstrong's Bitcoin network environment. Specifically, when implementing Hearn's Example 7, a host computer in Armstrong's network environment that acts as a payment module for Party A (client) (i.e., holds the client's wallet), such as first host computer 14, includes the logic (software) necessary to handle Hearn's micropayment protocol including logic for processing the transactions specified in the protocol. Armstrong contemplates various options for a component acting as Party B's payment module (holding the server's wallet). In one option, second host computer 16 acts as Party B's (server's) payment module. (See CB-1006, ¶103 (describing example

transaction between a wallet held by first host computer and a wallet held by second host computer); CB-1007, ¶¶54; CB-1008, ¶¶69.) In another option, Party B's user device (the server) acts as Party B's payment module. That is, the server both provides the service and holds Party B's wallet. This alternative is contemplated by Hearn and supported by Armstrong as I discussed above. I have modified Figure 1A to include this second alternative with the user device shaded gray.



Armstrong, Annotated and Modified Figure 1A

80. In a micropayment contract initiated by Party A (client) with Party B (server), the first host computer (which hosts Party A's wallet) handles the client-side portion of the protocol and processes transactions on behalf of Party A. The

component hosting Party B's wallet, and therefore acting as its payment module, handles the server-side portion of the protocol and processes transactions on behalf of Party B. Hearn teaches that Party B's user device acts as its payment module. Armstrong teaches an embodiment where a host computer, coupled to a party's user device, acts as payment module for that party. (CB-1006, ¶103 (describing example transaction between a wallet held by first host computer and a wallet held by second host computer); CB-1007, ¶54; CB-1008, ¶69.) A POSITA would have understood Hearn's micropayment contract and Armstrong's architecture permits both options for a component acting as Party B's payment module. (CB-1003, ¶¶79-80, 90.) Figure 1A above has been modified to include a user device (shaded gray) acting as Party B's payment module.

81. A POSITA would have been motivated to combine Hearn with Armstrong for a number of reasons. First, Hearn explicitly suggests the combination. Hearn's distributed contracts are "method[s] of using **Bitcoin** to form agreements with people via the block chain." (CB-1009, 1.) However, Hearn provides few details on the Bitcoin infrastructure for performing its distributed contracts. Hearn also does not indicate that there are any unique requirements for an architecture beyond supporting Bitcoin needed to implement its contracts. A POSITA would have therefore been motivated to search for references describing Bitcoin architectures and would have been led to Armstrong which like Hearn, is

directed to Bitcoin and includes all the necessary components for implementing Hearn's contracts.

82. Both Armstrong and Hearn are in the same field as the '566 patent—systems and methods for use with decentralized digital currencies such as Bitcoin. Armstrong describes a “system and method for transaction [sic] bitcoin is described.” (CB-1006, Abstract; CB-1007, Abstract; CB-1007, Abstract.) Hearn describes “distributed contracts” which use “Bitcoin to form agreements with people via the block chain.” (CB-1009, 1.) The asserted claims of the '566 patent are directed to “processing a transaction ... via a transfer mechanism, the transfer mechanism comprising a decentralized digital currency.” (CB-1001, claim 1, *see also*, claim 7.)

83. Second, a POSITA would have been motivated to implement Hearn's distributed contracts, including Example 7's micropayments, using Armstrong's system to enhance the capabilities of the Armstrong's Bitcoin environment. In particular, Armstrong describes a system enabling users to transfer bitcoin from one wallet to another. A POSITA would recognize that adding the capability to perform contracts to Armstrong environment would provide more options for users which in turn would drive more users interested in the new functionality to Armstrong's network. Hearn describes a motivation to implement a “micropayments” contract protocol in a network. Specifically, as described by

Hearn, cases exist where parties “want to rapidly and cheaply adjust the amount of money sent to a particular recipient without incurring the cost of a broadcast transaction.” (CB-1009, 5.) A POSITA would have also recognized that transaction fees make implementation of micropayment options impractical in existing payment systems such as credit/debit card payments systems. Because of the need to provide an effective on-line method to pay for small dollar amount transactions as articulated by Hearn, a POSITA would have been motivated to implement Hearn’s micropayment technique within Armstrong’s Bitcoin environment to provide a cost-effective option to users wishing to make small payments.

84. Third, a POSITA would have been motivated to use Armstrong’s web-based user interfaces and host computers which facilitate Bitcoin transactions to provide a simple mechanism for a user to enter into complex distributed Bitcoin contracts. As I discussed above in the overview of Armstrong, Armstrong provides a web interface with pre-defined fields allowing a user to simply enter values for the transaction. Additionally, the use of a facilitator to handle the processing involved in a Bitcoin transaction on behalf of an end-user allows for a more simplified application to be deployed and maintained on an end-user device. That is, with Armstrong, the Hearn contract logic is deployed on the host computer allowing a more simplified, thinner client to be deployed on the client device.

Updates to Hearn's logic need only to be made to the host computer, simplifying maintenance of the system.

85. Finally, the combination is nothing more than the application of a known technique (Hearn's distributed contracts) to a known device (Armstrong's Bitcoin environment and devices) that is ready for improvement. The results of the combination would have been predictable to a POSITA because the combination merely involves the implementation of a protocol within the framework of the existing and known Bitcoin technology. A POSITA would have had a reasonable expectation of success because, as noted by Hearn, a version of a micropayment protocol was implemented "in bitcoinj"³ and published prior to the earliest possible priority date of the '566 patent. (*See* CB-1009, 5.)

D. Independent Claim 1

86. The combination of Hearn and Armstrong teaches every limitation of claim 1 which I discuss in further detail in this section. For ease of discussion, I have provided labels for the preamble and limitations of claim 1.

[1P] *A computing device for processing a transaction between a first client device, and a second client device via a transfer mechanism, the transfer mechanism comprising a decentralized digital currency, the computing device comprising:*

³ "bitcoinj" is a software library for working with the Bitcoin protocol.

[1A] *a memory for storing a first asymmetric key pair, the first asymmetric key pair comprising a first private key and a first public key;*

[1B] *a network interface for receiving terms, the terms comprising:*

[1B.1] *at least one of a first principal data or a second principal data;*

[1B.2] *a reference to at least one of a first data source or a second data source; and*

[1B.3] *an expiration timestamp;*

[1C] *a computer processor coupled to the memory and the network interface, the computer processor configured to:*

[1C.1] *read the first private key from the memory;*

[1C.2] *compute a first cryptographic signature from the first private key;*

[1C.3] *create an inchoate data record comprising:*

[1C.3a] *a commit input for receiving a commit data from a commit transaction;*

[1C.3b] *one or more output data obtained from at least one of the first principal data or the second principal data, and a value data from at least one of the first data source or the second data source; and*

[1C.3c] *the first cryptographic signature; and*

[1C.4] *publish the inchoate data record to at least one of the first client device or the second client device,*

[1D] *wherein the decentralized digital currency comprises a distributed ledger that enables processing the transaction between the first*

client device and the second client device without the need for a trusted central authority,

[1E] *wherein the inchoate data record is used by at least one of the first client device or the second client device to create a complete data record and to create the transaction by broadcasting the complete data record for transmitting and receiving among network participants in the computer network for recording in the distributed ledger,*

[1F] *wherein at least one of the first client device or the second client device signs the inchoate data record and saves a copy of the inchoate data record on at least one of the first client device or the second client device; and*

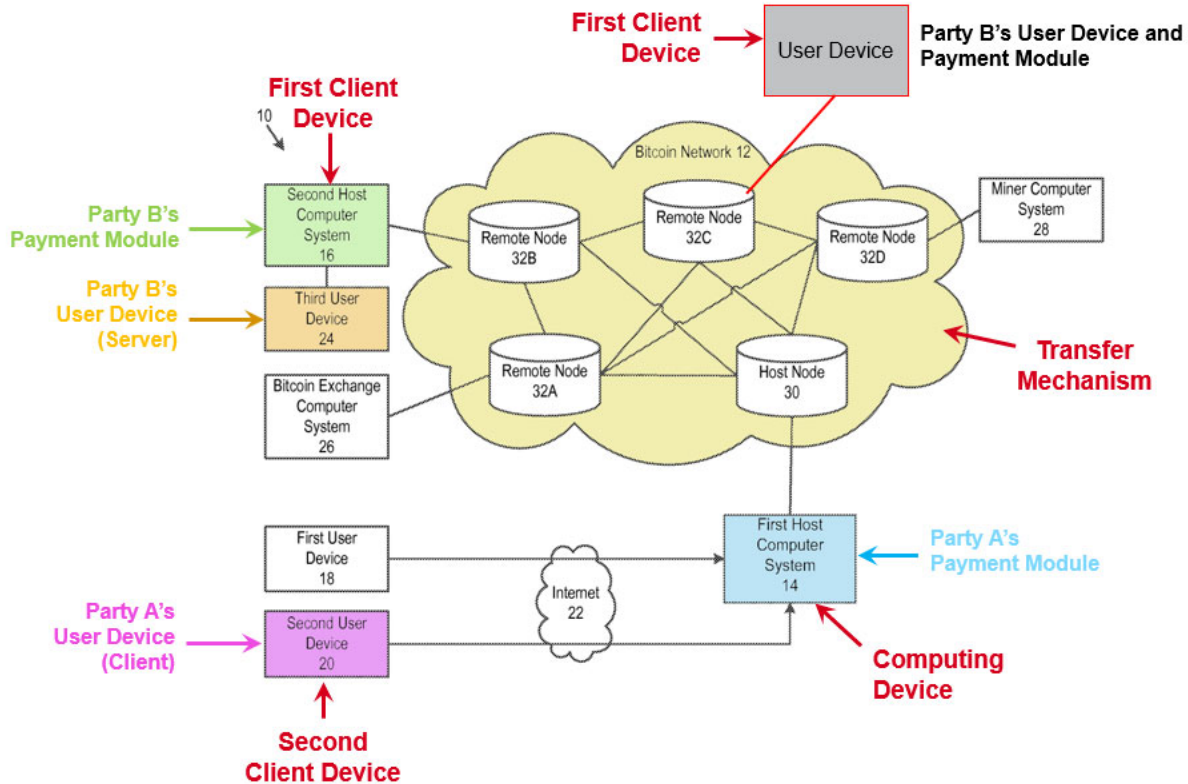
[1G] *wherein the at least one of the computing device, the first client device, or the second client device verifies the recording of the complete data record in the distributed ledger by observing an external state.*

1. Preamble [1P]

87. The combination of Armstrong and Hearn discloses a “*computing device for processing a transaction between a first client device, and a second client device via a transfer mechanism, the transfer mechanism comprising a decentralized digital currency*” [1P].

88. Hearn’s Example 7, which I discussed in detail in §V.A, discloses a distributed contract for “*rapidly-adjusted (micro)payments*” using Bitcoin. Hearn’s Example 7 encodes a micropayment contract between two parties which uses three Bitcoin transactions—an “escrow” (commit) transaction (T1), a refund transaction

(T2), and a payment transaction (T3). (*See* CB-1009, 5.) The application Hearn uses to describe Example 7 is the purchase of Internet access by a client (Party A) from a server (Party B) representing an entity offering the Wi-Fi access point (AP). (*See* CB-1009, 5.) Example 7 allows the client (Party A) to pay as she goes for the service. In other words, Party A (client) makes a series of micropayments throughout the duration of the contract as she uses the service. In the combined system of Hearn and Armstrong, illustrated below, Armstrong's first host computer 14 (shaded blue) acts as the payment module for Party A (client), hosting Party A's (client's) wallet. (*See* CB-1006, ¶8 (a host computer system "host[s] accounts or 'wallets' that allow users to make and accept payments using bitcoin"); CB-1007, ¶5; CB-1008, ¶5.) As I discussed in the Armstrong's overview, Armstrong's host computer "is used primarily for transacting bitcoin." (CB-1006, ¶83; CB-1007, ¶34; CB-1008, ¶49.) Armstrong's first host computer 14 in the combined system of Hearn and Armstrong is therefore the "*computing device for processing a transaction.*"



Armstrong, Annotated and Modified Figure 1A

89. In Armstrong, an application on a user device “transmits a user interface request over the Internet 22 to the website 34” on the host computer. (CB-1006, ¶85; CB-1007, ¶36; CB-1008, ¶51.) The second user device of Armstrong (shaded pink) is therefore the “*second client device*” because it is used by the user to enter transaction and contract terms when the Party A (client) is initiating a contract.

90. In Hearn, Party A (client) enters a micropayment contract with Party B (server). That is, Party A (client) initiates the contract and Party B (server) is the

recipient. Second host computer 16 hosts the wallet for Party B and acts as its payment module. Therefore, the second host computer is the “*first client device*” because it performs all the actions required during the contract on behalf of Party B without requiring any direct input from Party B. Should an argument be made that a host computer is not a “*client device*”, the combination further teaches that Party B’s user device acts as its payment module, as illustrated by the gray box added to Figure 1A. In this alternative, Party B’s user device is the “*first client device*.” I note that Party B’s payment module operates in the same manner when Party B is the recipient of Hearn’s micropayment contract, regardless of where it is physically located.

91. In stage 2 of Hearn’s micropayment protocol, first host computer 14, acting as the payment module for Party A (client), creates, signs, and sends an initial payment transaction T3 to the component acting as Party B’s (server’s) payment module. (CB-1009, 5 (steps 7-9).) During the contract period between Party A (client) and Party B (server), when “the client wishes to pay the server, it adjusts its copy of T3 to allocate more value to the server’s output and less to its ow [sic].” (CB-1009, 5 (step 9).) The signed adjustments to T3 are provided by host computer 14 (Party A’s (client’s) payment module) to Party B’s (server’s) payment module (host computer 16 or Party B’s user device) which then adjusts its copy of T3 “to match the new amounts.” (*Id.*) The initial, incomplete version of T3 and the

T3 updates are sent to Party B's payment module (second host computer 16 or Party B's device) over a communication medium such as the Internet.

92. In Hearn's micropayment protocol (Example 7), "[w]hen the internet session ends or the predetermined time-period (e.g., 1-day) [for the contract] is close to expiring," the server "signs and **broadcasts** the last transaction it saw, allocating the final amount to itself." (CB-1009, 5.) Armstrong explains that a "transaction is broadcast to the Bitcoin network for block chain verification." (CB-1006, ¶10; CB-1007, ¶7; CB-1008, ¶7.) Armstrong's Bitcoin network 12 is therefore a "*transfer mechanism*." (CB-1001, 38:8-11.) Bitcoin was a well-known "*decentralized digital currency*" prior to the '566 patent. The admitted prior art of the '566 patent notes that "Decentralized digital currencies (or so-called 'cryptocurrencies') ... are relatively new creatures. Bitcoin protocol and progeny (Ethereum, Litecoin, etc.) are one class of technologies that have recently enjoyed **meteoric rises in popularity** (and valuation)." (CB-1001, 2:33-43.) Antonopoulos also describes Bitcoin as a decentralized digital currency. (*See, e.g.*, CB-1012, 1-3.)

93. The combination of Hearn and Armstrong therefore discloses a "*computing device* [first host computer 14 acting as payment module for Party A] *for processing a transaction* [e.g., payment transaction T3] *between a first client device* [payment module for Party B provided by second host computer 16 or Party

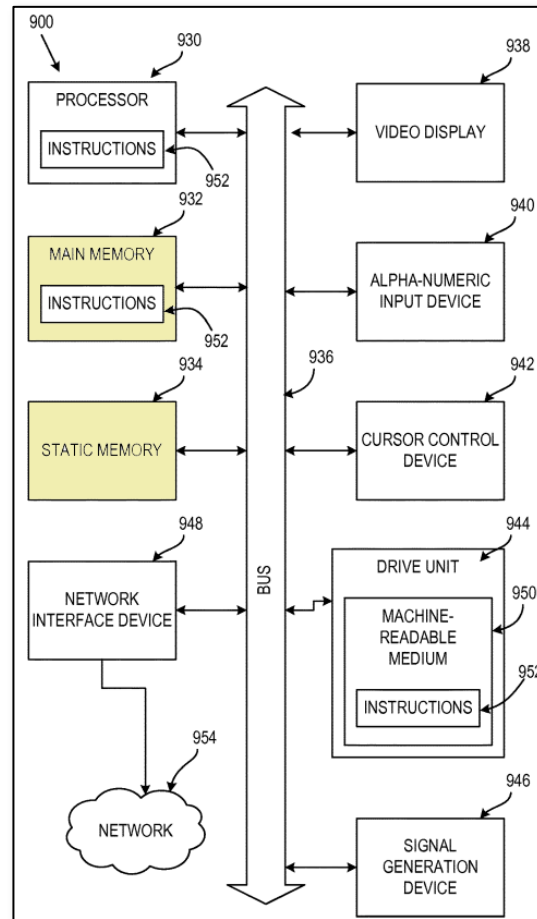
B's user device], *and a second client device* [second user device] *via a transfer mechanism* [Bitcoin network], *the transfer mechanism comprising a decentralized digital currency* [bitcoin].”

a. “Memory” limitation [1A]

94. The combination of Armstrong and Hearn discloses “*a memory for storing a first asymmetric key pair, the first asymmetric key pair comprising a first private key and a first public key*” [1A].

95. The use of memory for storing information in a computing device has been known for many decades prior to the '566 patent. Armstrong discloses “a machine in the exemplary form of a computer system 900” illustrated in Figure 74 below. (CB-1006, ¶225; CB-1007, ¶58, Figure 54; CB-1008, ¶93, Figure 54.) The computer system includes “main memory 932 (e.g., read-only memory (ROM), flash memory, dynamic random access memory (DRAM) such as synchronous DRAM (SDRAM) or Rambus DRAM (RDRAM), etc.), and a static memory 934 (e.g., flash memory, static random access memory (SRAM, etc.).” (CB-1006, ¶226, Figure 74; CB-1007, ¶59, Figure 54; CB-1008, ¶94, Figure 54.) Armstrong teaches that its “machine may operate in the capacity of ... a peer machine in a peer-to-peer (or distributed) network environment.” (CB-1006, ¶225; CB-1007, ¶58; CB-1008, ¶93.) That is, each device in Armstrong, including the host computer devices, utilizes this “computer system” architecture. (CB-1006, ¶225; CB-1007, ¶58; CB-

1008, ¶¶93.) Thus, the “*computing device*” [first host computer] includes “*a memory*” [1A].



Armstrong, Annotated Figure 74

96. The “*memory*” in the first host computer “*stor[es] a first asymmetric key pair.*” In Armstrong, the wallet establishment module of the host computer “establishes a first wallet in the data store.” (See CB-1006, ¶¶16, 111 (“local register 60 includes a plurality of wallets 42”); CB-1007, ¶8; CB-1008, ¶¶9, 77.) The wallet further “stores the **public key of the Bitcoin address and its associated**

private key.” (CB-1006, ¶8; CB-1007, ¶5; CB-1008, ¶5.) A POSITA would have understood the public/private key pair stored by Armstrong for a user’s wallet is an “*asymmetric key pair*.” Asymmetric (or public-key) encryption algorithms were known for nearly 40 years before the ’566 patent. (*See* CB-1013, 4-5, 461.) In asymmetric encryption, keys “come in pairs”—a public key and corresponding private key, referred to as an asymmetric key pair. (CB-1013, 461, CB-1014, 4-5, 283.) I provide further details regarding asymmetric cryptographic algorithms in the technology section of this declaration. I also note that the admitted prior art of the ’566 patent confirms that it was known to store keys persistently and securely in memory of the device. (CB-1001, 4:30-41.)

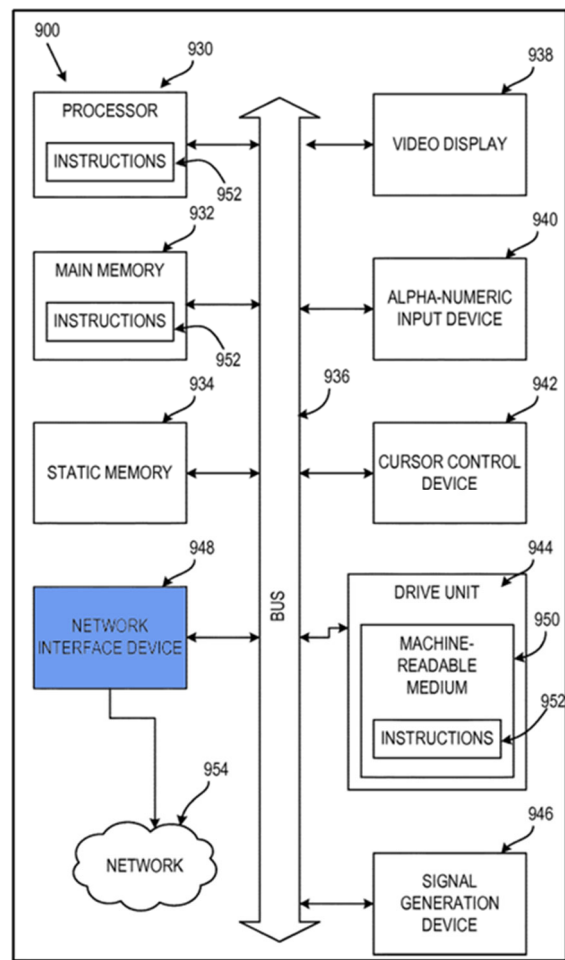
97. Hearn discloses that its micropayment protocol uses public and private key pairs. (CB-1009, 5.) For example, in step 1, Party A “create(s) a public key (K1).” (CB-1009, 5.) The process of creating the public key also creates an associated private key, as I discussed in the technical background section. (CB-1013, 461, CB-1014, 4-5, 283.) Hearn further teaches Party A’s payment module signs transaction T3 with its private key and Party B’s payment module verifies the signature using Party A’s public key. (*See* CB-1009, 5 (steps 7-8).) In the combination, the component acting as Party A’s payment module (the host computer) accesses Party A’s private key to sign T3. Thus, in the combination of Hearn and Armstrong, the first host computer acting as Party A’s payment module

includes “*memory for storing a first asymmetric key pair, the asymmetric key pair comprising a first private key and a first public key*” associated with Hearn’s Party A.

b. “Network interface” limitation [1B]

98. The combination of Armstrong and Hearn discloses “*a network interface for receiving terms*” [1B].

99. Like computer memory, network interfaces have been known as long as computers have been connected to networks. Armstrong’s first host computer system is connected over the Internet to the first and second user devices 18 and 20. (CB-1006, ¶81; CB-1007, ¶32; CB-1008, ¶47; *see also*, CB-1006, ¶83; CB-1007, ¶34; CB-1008, ¶49.) The computer system of Armstrong, used to implement the first host computer, includes a “network interface device 948” [shaded blue] as shown in Figure 74 below. (CB-1006, ¶227; CB-1007, ¶60; CB-1008, ¶95.) The network interface is “connected to the processor ...” as shown. (CB-1006, ¶16; CB-1007, ¶8; CB-1008, ¶9.)



Armstrong, Figure 74

100. In Armstrong, an application on the user device “transmits a user interface request” to website 34 on the host computer (CB-1006, ¶85; CB-1007, ¶36; CB-1008, ¶51.) Website 34 responds “by transmitting the user interface 36” to the user device. (CB-1006, ¶85; CB-1007, ¶36; CB-1008, ¶51.) User interface 36 provides fields for entering information regarding terms of a bitcoin transaction (e.g., amount of bitcoin, etc.). (CB-1006, ¶86; CB-1007, ¶37; CB-1008, ¶52.) For

example, Armstrong's Figure 42 shows "a view displayed to the user of the second user device 20 ... to make a transfer from their wallet (Wallet B)" to another wallet, including entry of either an amount of Bitcoin or an amount in USD to transfer to another party. (CB-1006, ¶101, Figure 42; CB-1007, ¶52, Figure 42; CB-1008, ¶67, Figure 42.) Thus, the combination of Hearn and Armstrong discloses "*a network interface for receiving*" data [1B].

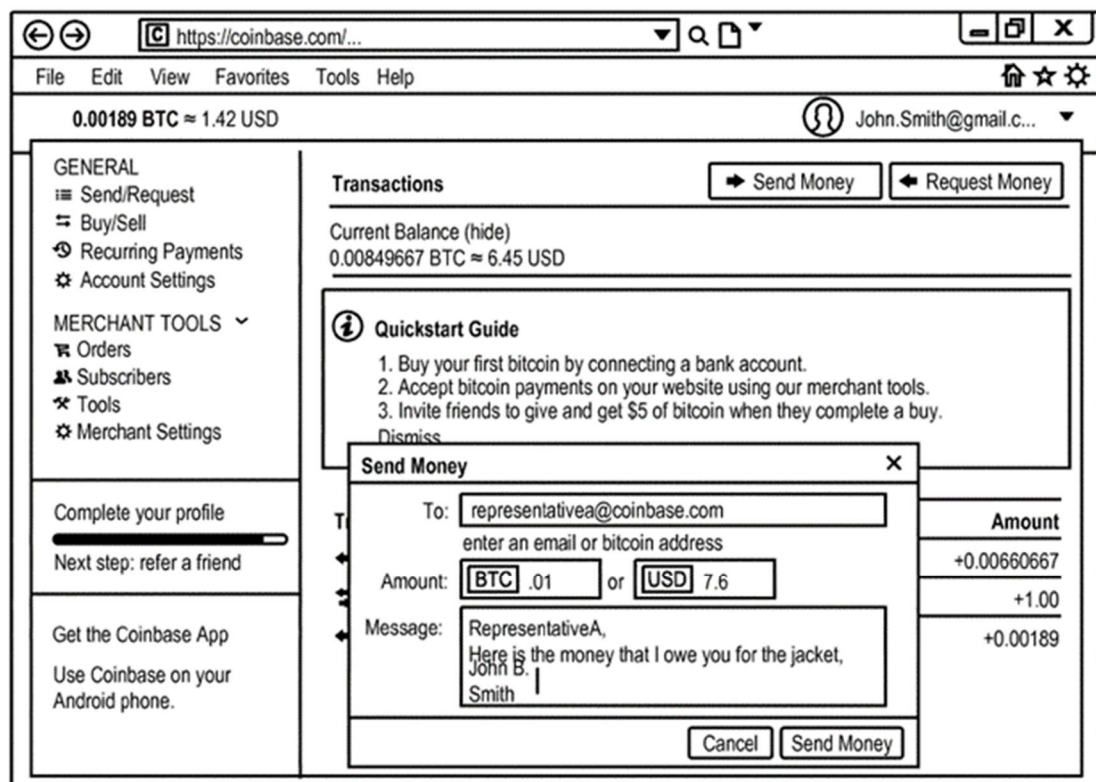


FIG. 42

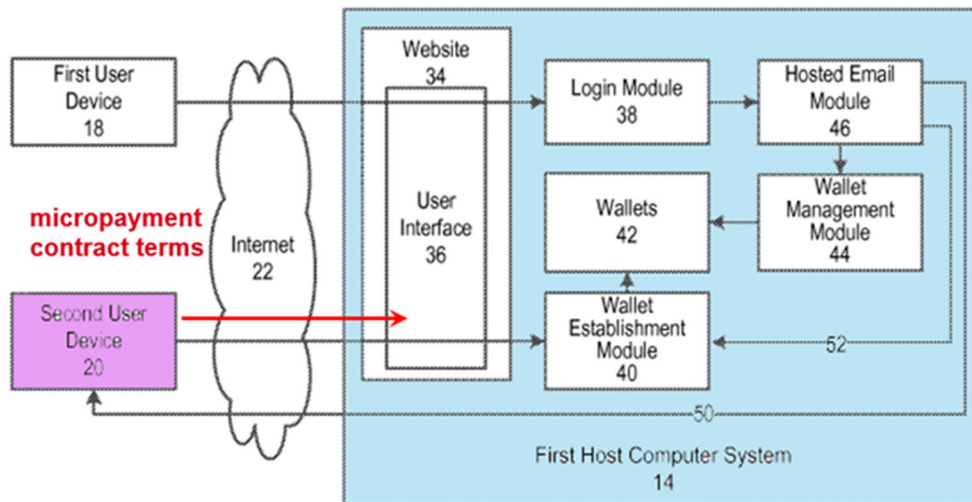
Armstrong, Figure 42

c. “Term” limitations [1B.1]-[1B.3]

101. Claim 1 recites a specific set of terms received over the “*network interface*”: “*at least one of a first principal data or a second principal data*” [1B.1]; “*a reference to at least one of a first data source or a second data source*” [1B.2] and “*an expiration timestamp*” [1B.3]. The combination of Hearn and Armstrong discloses each of these terms and discloses that the terms are received over the network interface.

102. Hearn’s Example 7 details a protocol for implementing a contract between Party A (client) and Party B (server) for a service offered by Party B (e.g., Wi-Fi access) using incremental, metered micropayments from Party A (client). The contract is initiated by the individual wishing to purchase the service. Therefore, Party A (client) must take steps to initiate the contract for purchase of Wi-Fi using micropayments with Party B (server). To set up the micropayment contract, Party A (client) provides the terms governing the agreement such as the recipient, total maximum value for the contract, and length of the contract. As taught by Armstrong, the party initiating the contract (Party A (client) in this example) enters transaction terms using the web interface provided by the host. In the combination, the contractual terms are received from the “*second user device*” (Party A’s user device) over the “*network interface*” of the first host computer 14

(“*computing device*”) in the combination of Hearn and Armstrong, as illustrated in annotated Figure 1B below.



Armstrong, Annotated Figure 1B

(i) “Principal Data” [1B.1]

103. In the combination of Hearn and Armstrong, the “*computing device*” (first host computer) receives “*at least one of a first principal data or a second principal data*” from Party A’s user device (second user device). The ’566 patent provides an example providing contractual terms which I reproduce below with principal data highlighted:

1. A first client transmits an offer to a facilitator, the offer comprising terms, the terms comprising:
 - a. a reference to a data source comprising at least one of:
a base instrument and a quote instrument;
 - b. a principal amount;
 - c. an expiration timestamp;
 - d. optionally a reference to a denominating asset;
 - e. optionally, a collateral amount; and
 - f. optionally, a disbursement function;

Example Terms:

Base: USD

Quote: AUD

Denominating: BTC

Principal: 0.5 (BTC)

Collateral: 2×principal

$$res_{base}(b_o, q_o, b_f, q_f): \text{principal} \times \frac{b_f - b_o}{q_f - q_o}$$

Expiration: 2014-06-01T12:34:56Z

...

(CB-1001, 11:53-12:11.)

104. Hearn's micropayment contract requires the specification of a principal data amount for the contract. In Hearn's Wi-Fi access example, Party A (client) indicates that it would "like to pay 0.001 BTC per 10 kilobytes of usage" with a maximum total payment of 10 BTC (100 Mbytes of usage) for the entire contractual period. (CB-1009, 5.) That is, Party A (client) sets the maximum amount it will pay for the service over the duration of the contract and sets an agreed upon price for unit of the service. So, Party A (client) makes micropayments to Party B (server) as it uses the service at 0.001 BTC per 10 kilobytes per usage

but the contract/service will end when Party A (client) has consumed 10 BTC worth of service. The setting of the maximum value for the payment channel between the parties is set using Hearn's "escrow" transaction. Specifically, during establishment of the payment channel between Party A (client) and Party B (host), the payment module for Party A creates an "escrow" transaction (T1) specifying the **maximum anticipated value to be paid** by Party A (client) to Party B (server) for the duration of the micropayment contract. (CB-1009, 5 (step 2).) This "maximum anticipated value" is specified by Party A (client) who initiates the contract and is the "*first principal data*."

105. Armstrong discloses the party initiating a transaction identifies the amount to be transferred from its wallet to the recipient of the transaction. In the discussion of a transaction between two parties whose wallets are held by different payment modules, Armstrong discloses that the "user of the second user device 20 enters a bitcoin address (Bitcoin address 6) that is located in the third wallet (Wallet C) and an amount of bitcoin to be transferred to the third wallet." (CB-1006, ¶103; CB-1007, ¶54; CB-1008, ¶69.) That is, the user enters the recipient for the transaction and the total amount of the transaction. Armstrong stresses throughout that the amount of the transaction can be entered in either bitcoin or in the local currency over its user interface: "The user interface 36 also includes a field for entering an **amount in bitcoin** (or an **amount in local currency** that is converted

to bitcoin using an exchange rate) that is being transferred from the wallet ... (CB-1006, ¶86; CB-1007, ¶37; CB-1008, ¶52; *see also*, CB-1006, ¶101; CB-1007, ¶52; CB-1008, ¶67 (“The view includes a field for the user to enter the email address (email address A) and fields for entering either an **amount in bitcoin (BTC) or an amount in a local currency (USD-United States Dollar).**”).)

106. Although Hearn only mentions its contractual terms in the context of a total amount in bitcoin, a POSITA would have been motivated to integrate a capability to allow a party to the micropayment contract to enter the contract amount in either bitcoin or in a local currency as taught by Armstrong. For example, the café may offer its Wi-Fi access on a per-usage basis for 0.01 USD for 10 kilobytes and allow payment in bitcoin. Business and market realities would have motivated a POSITA to provide this capability, particularly in the time period before the '566 patent. Prior to the '566 patent, use of digital currencies was still very limited. Therefore, a service provider (e.g., a café offering Wi-Fi service) would have been motivated to offer its service in terms of the local currency but also provide the ability for a user to pay with bitcoin. This provides an additional payment option for customers but providers pricing in terms of the most likely form of payment. Additionally, the exchange rate between bitcoin and local currency fluctuates. Therefore, a POSITA would have recognized that it would be easier and financially safer for the service provider to set a price in local currency, allow a

user to pay with bitcoin, and rely on the Bitcoin network provider to determine the equivalent price of the service at the instant it is purchased.

107. The combination of Hearn and Armstrong also teaches that the “*principal data*” is received via the “*network interface*.” As I discussed above, in the combination of Hearn and Armstrong, the first host computer 14 acts as the payment module for Party A (client), hosting Party A’s wallet and processing transactions on behalf of Party A (client). Therefore, to handle the micropayment contract for Party A (client) with Party B (server), the contractual terms must be received from the user including the total amount for the contract. As I have discussed above, Armstrong discloses that a user enters an amount for a transaction through the user interface field provided by the host computer. (*See, e.g.*, CB-1006, ¶86(user interface 36 includes “field for entering an **amount in bitcoin** (or an **amount in local currency** that is converted to bitcoin using an exchange rate) that is being transferred from the wallet”), ¶103(User “enters a bitcoin address ... and **an amount of bitcoin to be transferred** to the third wallet”); CB-1007, ¶¶52, 54; CB-1008, ¶¶67-69.) The transactional amount specified in Armstrong’s example is equivalent to the total contractual amount entered by the user for Hearn’s micropayment contract. Therefore, in the combination of Hearn and Armstrong, the “*principal data*” (e.g., maximum contract value in either bitcoin or local

currency) is entered by Party A (user) in Party A's user device and received by the host computer via the "*network interface*."

(ii) "*Reference*" to a "*data source*" [1B.2]

108. In the combination of Hearn and Armstrong, the "*computing device*" (first host computer) receives "*a reference to at least one of a first data source or a second data source*" over the network interface. (CB-1001, 39:59-60.) The '566 patent provides an example providing contractual terms which I reproduce below with reference data highlighted. As highlighted, a reference to a data source comprises at least one of a base instrument (e.g., USD) and a quote instrument (e.g., AUD).

1. A first client transmits an offer to a facilitator, the offer comprising terms, the terms comprising:
 - a. a reference to a data source comprising at least one of:
a base instrument and a quote instrument;
 - b. a principal amount;
 - c. an expiration timestamp;
 - d. optionally a reference to a denominating asset;
 - e. optionally, a collateral amount; and
 - f. optionally, a disbursement function;

Example Terms:

Base: USD

Quote: AUD

Denominating: BTC

Principal: 0.5 (BTC)

Collateral: 2×principal

$$res_{base}(b_o, q_o, b_f, q_f): \text{principal} \times \frac{b_f - b_o}{q_f - q_o}$$

Expiration: 2014-06-01T12:34:56Z

...

(CB-1001, 11:53-12:11.)

109. As I discussed in the previous section, a POSITA would have been motivated in the combination to provide a field to specify the contractual value in the local currency. The local currency is therefore at least the “quote instrument” because the price is quoted using that currency. Armstrong’s user interface further presents to the user the “exchange rate between bitcoin and the local currency” as shown in Figure 42 below. (See CB-1006, Figure 42; CB-1007, Figure 42; CB-1008, Figure 42.) An exchange rate is necessary to allow Armstrong’s user to

specify a transaction in terms of the local currency. Therefore, in the combination of Hearn and Armstrong, to support use of bitcoin when a seller provides pricing in a local currency, a POSITA would have been motivated to use an exchange rate to automatically convert an entered local currency amount to a bitcoin amount so that the user does not have to manually convert between a local currency and bitcoin and the parties can be confident the most recent exchange rate is being used and the conversion amount is accurate. A POSITA would understand that the exchange rate is provided from a “*data source*” such as an external feed from an exchange rate provider. Many external sources for exchange rates existed prior to the ’566 patent including bitcoincharts.com, bitcoinaverage.com, and ZeroBlock. (See CB-1012, 6-7.) This is consistent with the ’566 patent which states that the “data source comprises an external data feed, internal database, another data source, etc.” (CB-1001, 15:38-40.)

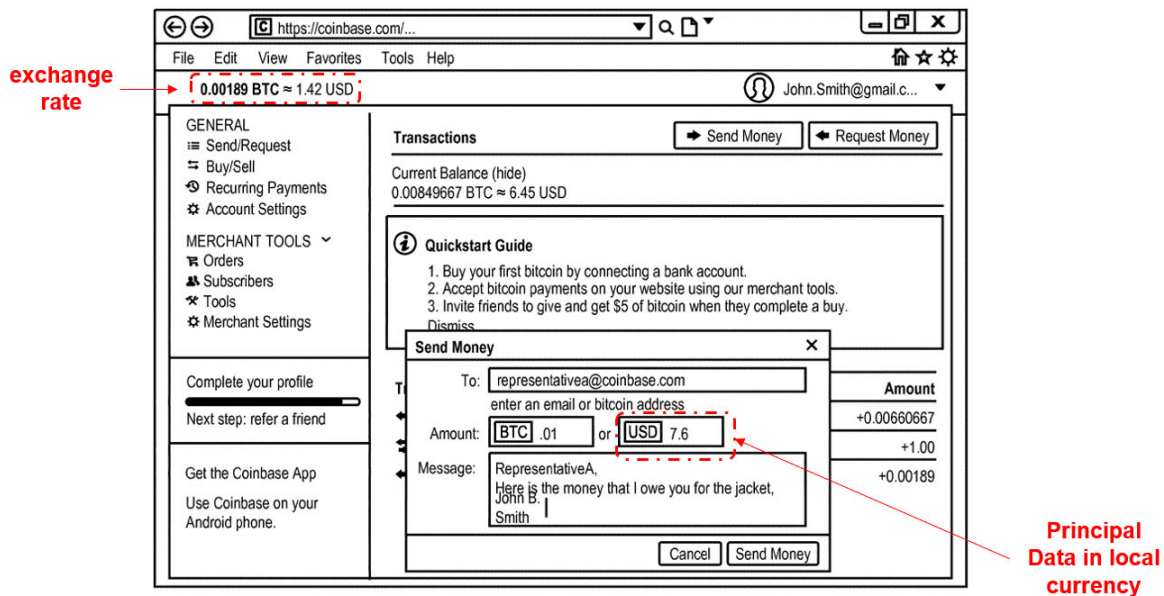


FIG. 42

Armstrong, Annotated Figure 42

110. Thus, because the user enters the amount in a specified local currency and the exchange rate is received from an external source, the combination of Hearn and Armstrong teaches or suggests that “a reference to at least one of a first data source or a second data source” is received over the “network interface” of the first host computer (recited “computing device”).

(iii) “Expiration Timestamp” [1B.3]

111. The combination of Hearn and Armstrong teaches or suggests the “computing device” receives an “expiration timestamp” from Party A’s user device over the “network interface.”

112. Hearn’s Example 7 discloses that its micropayment channel between the Party A (client) and Party B (server) expires after a pre-determined time. Hearn

specifically teaches that incremental micropayments from Party A (client) to Party B (server) “continue[] until the session ends, or the 1-day period is getting close to expiry.” (CB-1009, 5.) Thus, when this pre-determined time expires, Party A (client) makes no further micropayments to Party B (server) and conversely Party B (server) accepts no further micropayments from Party A (client). That is, this pre-determined time sets an expiration time for the payment channel (and the contract). Therefore, this pre-determined time is an “*expiration timestamp*.”

113. As I described in §V.A, refund transaction T2 includes a lock time which specifies a time period which must expire before the refund transaction T2 is broadcast to the Bitcoin network. A POSITA would have understood that the expiration time for the channel occurs prior to when the refund transaction becomes spendable by Party A (client). Both the refund transaction and final payment transaction T3 spend the value of T1. Therefore, the first of T2 or T3 to make it to the network claims the output of T1 and makes the second transaction invalid. Therefore, the expiration time for the channel must occur a sufficient time before the refund is “unlocked” to permit Party B (client) sufficient time to broadcast the final payment transaction T3 to the Bitcoin network. That is, the lock time for the refund transaction is offset by a set amount from the expiration time of the payment channel to foreclose malicious action by Party A (client) to use Party B’s service and get the entire contract amount refunded.

114. A POSITA would have been motivated to permit a user (e.g., Party A) to enter the duration of the micropayment channel (contract) as well as the refund lock time and/or time offset over the network interface with other contract terms (e.g., “*principal data*”) when the contract is being formed. A POSITA would have been motivated to provide this feature to enhance the offering to a client and to permit services with different timing needs to use the micropayment contract capability. For example, this feature provides parties with the flexibility to modify contract duration based on, for example, the type of service being provided or the level of trust between the parties. For example, one user may wish to purchase internet access over a 2-hour period (e.g., from a café) whereas another user may wish to purchase internet access over a 24-hour period (e.g., from a hotel where the user is staying). A user may further wish to limit duration of a contract with an unknown, untrusted entity but may feel more comfortable setting a longer duration with a trusted entity (e.g., a friend).

115. I note that the association of an expiration time/date for a contract between parties was well-known long before the ’566 patent and would be common knowledge even to a lay person. For example, a contract between a professional athletics team and a player typically specifies an end date for the contract. More specifically in the field of digital currencies, Allen teaches that in an option contract, the user has “the right to buy digital currency at a set price before or on

the expiration date of the contract.” (CB-1015, ¶54.) And Walker discloses the creation of a conditional purchase offer (CPO) for goods or services over the Internet specifying an “expiration date” for the CPO. (CB-1016, 16:28-29.)

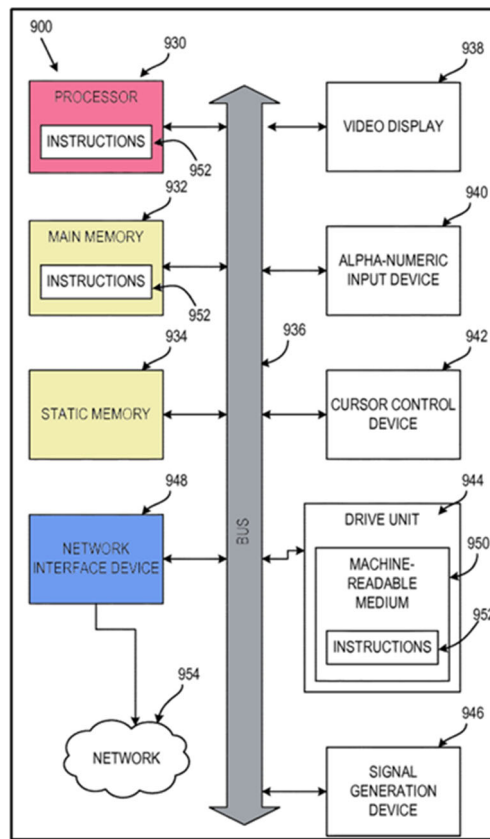
116. Thus, the combination of Hearn and Armstrong teaches or suggests to a POSITA the entry of an “*expiration timestamp*” by the party initiating the micropayment contract into the user interface and receipt of this term with the other contractual terms over the “*network interface*” of the first host computer (“*computing device*”).

d. “Computer Processor” limitation [1C]

117. The combination of Hearn and Armstrong discloses “*a computer processor coupled to the memory and the network interface.*” [1C].

118. Armstrong’s computer system illustrated below in Figure 74 includes **processor 930** coupled to memory 932, memory 934, and network interface device 948 via bus 936. (See CB-1006, ¶226-227, CB-1007, ¶59; CB-1008, ¶94.) As shown, and taught by Armstrong, the processor is coupled to both the memory and network interface. (See CB-1006, ¶16; CB-1007, ¶8; CB-1008, ¶9 (“host computer ... including a **processor**, a network interface device **connected to the processor**, a computer readable medium **connected to the processor**, a data store on the computer readable medium”), CB-1006, ¶225; CB-1007, ¶58; CB-1008, ¶93.) As I have discussed previously, Armstrong discloses that each device in its

network utilizes the computer system architecture depicted in Figure 74. (See CB-1006, ¶226; CB-1007, ¶58; CB-1008, ¶93.) Therefore, the “*computing device*” (first host computer) also includes “*a computer processor coupled to the memory and the network interface.*”



Armstrong, Annotated Figure 74

e. “Computer Processor” action limitations [1C.1]-[1C.4]

119. Claim 1 recites a set of four actions (labeled [1C.1]-[1C.4]) performed by the “*computer processor*” of the “*computing device*”:

[1C] *a computer processor coupled to the memory and the network interface, the computer processor configured to:*

[1C.1] *read the first private key from the memory;*

[1C.2] *compute a first cryptographic signature from the first private key;*

[1C.3] *create an inchoate data record comprising:*

[1C.3a] *a commit input for receiving a commit data from a commit transaction;*

[1C.3b] *one or more output data obtained from at least one of the first principal data or the second principal data, and a value data from at least one of the first data source or the second data source; and the first cryptographic signature; and*

[1C.3c] *publish the inchoate data record to at least one of the first client device or the second client device,*

[1C.4] *publish the inchoate data record to at least one of the first client device or the second client device*

The combination of Hearn and Armstrong teaches that the “processor” of the first host computer performs each of these actions as I detail in this section.

(i) **“Read the First Private Key” limitation [1C.1],
“Compute a First Cryptographic Signature”
limitation [1C.2]**

120. Limitations [1C.1] and [1C.2] are both actions related to the private key and therefore I address these limitations together. The combination of Hearn and Armstrong discloses the computer processor of the first host computer is

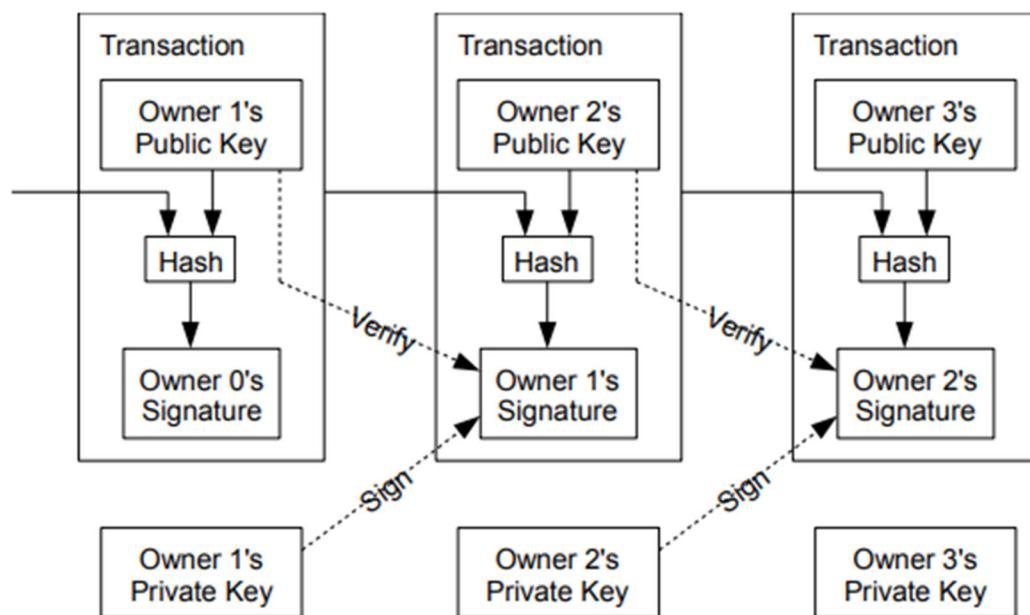
configured to “*read the first private key from the memory*” [1C.1] and to “*compute a first cryptographic signature from the first private key*” [1C.2].

121. In Hearn’s Example 7, the initial version of the transaction T3 includes a digital signature: “The client then creates a new transaction, T3” which the “client signs.” (CB-1009, 5 (step 7).) Party A (client) “provides the transaction and signature to the server” (Party B’s payment module). (CB-1009, 5 (step 7).) I discussed digital signatures in §III.B. Digital signatures were well-known for decades and were a standard element of cryptographic applications prior to the ’566 patent. As I explained, a digital signature is created from content by computing a digest of the content (e.g., using a hash algorithm) and then encrypting the digest using the private key of a party. (CB-1013, 38.) The encrypted content digest is a “*cryptographic signature*.” Accordingly, Hearn teaches use of a “*private key*” to “*compute a first cryptographic signature*” [1C.2] when creating the initial version of T3.

122. The ’566 patent admits that the use of digital signatures in Bitcoin transactions was known before its filing date in its admitted prior art section. (See CB-1001, 4:32-33 (“a payer must digitally sign the transaction using the corresponding private key”), 4:46-47 (“Every input must be digitally signed”).) The seminal paper on Bitcoin by Nakamoto, which I introduced in §III.A confirms use of cryptographic signatures in Bitcoin. Specifically, Nakamoto explains that

“[e]ach owner transfers the coin to the next by **digitally signing a hash** of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.”

(CB-1019, 2.) This process is illustrated in the following figure from Nakamoto which the '566 patent reproduces as its prior art Figure 24 without attribution to Nakamoto.



Nakamoto

123. Furthermore, as I discussed in §V.B, the “*memory*” of Armstrong’s first host computer (“*computing device*”) stores the “*private key*” and “*public key*” associated with Party A’s wallet. Thus, in the combination implementing Hearn’s Example 7, the “*memory*” stores the “*first private key*” for the agreement originator

(Party A). A POSITA would have understood that the “*first private key*” of Party A (client) must be “*read*” from “*the memory*” for the processor to compute the signature for transaction T3. That is, when initial T3 is created, the processor of the first host computer which implements the client-side logic for Hearn’s micropayment protocol retrieves the private key for Party A (client) and uses this private key to sign initial T3. This process is confirmed by both the ’566 patent’s discussion of the prior art and Nakamoto which teach the private key must be accessed to compute the signature for a transaction.

124. For these reasons, the combination of Armstrong and Hearn teaches “*read[ing] the first private key from the first private key*” [1C.1] and teaches “*comput[ing] a first cryptographic signature from the first private key.*” [1C.2]

(ii) “*Create an Inchoate Data Record*” [1C.3]

125. The combination of Hearn and Armstrong discloses a computer processor configured to “*create an inchoate data record.*” [1C.3].

126. In the payment stage of Hearn’s micropayment protocol occurring after the channel has been established, Party A (client) “creates a new transaction, T3” which allocates the entire maximum payment value to Party A (client) and nothing to Party B (server). (CB-1009, 5 (step 7).) That is, when T3 is first created, Party A (client) has not consumed any service and therefore retains the entire value of the contract. As Party A (client) uses the service, Party A (client) provides small

payments. Hearn implements this ability through updates to the initial version of T3. Therefore, during the micropayment contract between the parties (as set by the expiration timestamp), transaction T3 is adjusted by Party A (client) to assign more of the payment to Party B (server) and less to Party A (client) and “re-signed.” (CB-1009, 5.) The process of adjusting T3 “continues until the session ends, or the 1-day period is getting close to expiry.” (*Id.*) Party B (server) then “signs and broadcasts the last transaction it saw, allocating the final amount to itself.” (*Id.*) At this point, transaction T3 is complete.

127. The initial version of T3 created in step 7 is therefore an “*inchoate data record*” because it is unsigned by the receiving party (Party B) and is not fully formed.

**(iii) “*Inchoate Data Record*” content limitations
[1C.3a]-[1C.3c]**

128. Claim 1 further recites specific components of the “*inchoate data record*”: “a commit input for receiving a commit data from a commit transaction,” [1C.3a], “one or more output data obtained from at least one of the first principal data or the second principal data, and a value data from at least one of the first data source or the second data source,” [1C.3b] and “the first cryptographic signature” [1C.3c]. The initial version of T3 disclosed by Hearn includes each of these components as I discuss below.

(a) “Commit Input” [1C.3a]

129. Hearn’s initial T3 includes “*a commit input for receiving a commit data from a commit transaction*” [1C.3a]. As acknowledged in the admitted prior art of the ’566 patent, “a transaction must have one or more inputs.” (CB-1001, 4:42-43, 49.) For a bitcoin transaction to be valid, “every input must be an unspent output of a previous transaction.” (CB-1001, 4:45-46; *see also*, CB-1006, Figure 45 (illustrating that transaction uses previous transaction output as input); CB-1007, Figure 45; CB-1008, Figure 45.) I discussed these well-known and necessary aspects of Bitcoin in §III.A.

130. Hearn teaches that transaction, T3, “connects to T1.” (CB-1009, 5.) That is, as I discussed in §V.A, the unspent output of the escrow transaction, T1, is the “*input*” for the initial payment transaction T3 as shown in my Figure C below.

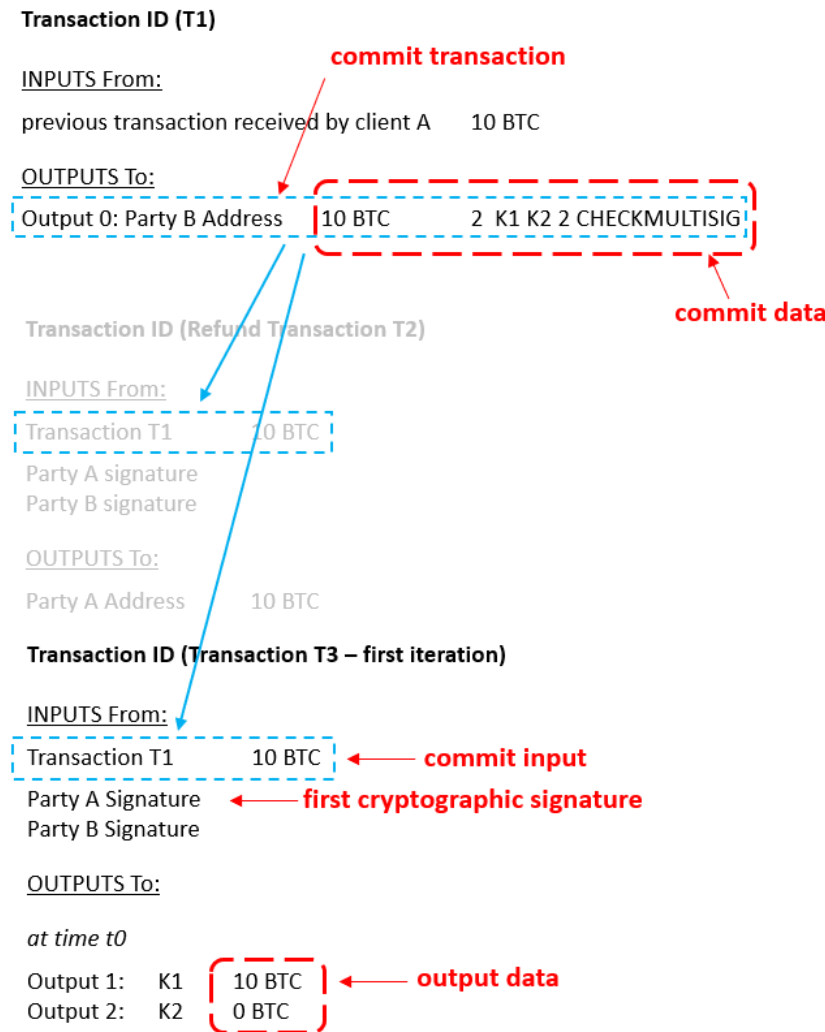


Figure C

131. The “escrow” transaction T1 “sets up a payment of (for example) 10 BTC to an output requiring both the server’s public key and one of your own to be used.” (CB-1009, 5 (step 2).) After both parties sign “escrow” transaction T1 (step 6), “escrow” transaction T1 is “broadcast[.]” to the Bitcoin network which commits T1 to the blockchain and “locks” in the bitcoin. (CB-1009, 5 (step 6).) Additional details on this aspect of Hearn are provided in §V.A. “Escrow” transaction T1 is a

“commit transaction” because it commits Party A (client) to transfer an amount of BTC (the *“commit data”*). The input of Transaction T3 (Transaction T1) is a *“commit input”* through which the *“commit data”* (total contract value) is received from Transaction T1 (*“commit transaction”*). Hearn’s initial payment transaction T3 therefore includes *“a commit input for receiving a commit data from a commit transaction”* [1C.3a].

(b) *“One or More Output Data”* [1C.3b]

132. The initial version of T3 includes *“one or more output data obtained from at least one of the first principal data or the second principal data, and a value data from at least one of the first data source or the second data source”* [1C.3b].

133. Hearn’s initial T3 has “two outputs”⁴ – “[o]ne goes to K1 and the other goes to K2.” (CB-1009, 5 (step 7).) As I discussed in §III, a POSITA would have also understood that the output would also include a “miner’s fee.” Both Armstrong and the admitted prior art in the ’566 patent support this understanding. (See CB-1001, 4:55-56; CB-1006, ¶103; CB-1007, ¶56; CB-1008, ¶71.) Transaction T3

⁴ A POSITA would have understood that the output of initial T3 also includes a “miner’s fee.” (CB-1001 (admitted prior art), 4:55-56; Armstrong, 15:62-16:10.)

“starts with all value allocated to the first output (K1)” (e.g., 10 BTC are assigned to Party A and 0 assigned to Party B which is reflected in my Figure C above). (CB-1009, 5.)

134. The “two outputs” (BTC amounts) of initial T3 are the recited “*one or more output data*” [1C.3b]. As discussed in §V.D.1.c.i, the “*principal data*” for Hearn’s Example 7 micropayment contract are received from Party A’s (client’s) user device at the host computer in local currency (e.g., U.S. Dollars) in the combination of Armstrong and Hearn. As taught by Armstrong, “an amount in local currency [] is converted to bitcoin using an exchange rate.” (CB-1006, ¶86; CB-1007, ¶37; CB-1008, ¶52.) The exchange rate is “*a value data from at least one of the first data source*” (exchange rate provider).

135. Thus, the “*one or more output data*” specified by an amount in bitcoin in Transaction T3 is “*obtained from at least one of the **first principal data** [agreement amount in local currency] or the second principal data and a value data [exchange rate] from at least one of the **first data source** [exchange rate provider] or the second data source*” [1C.3b]. That is, the “*output data*” in the combination is the “*principal data*” expressed in U.S. dollars multiplied by the “*value data*” (exchange rate) which was provided by a “*data source*.”

(c) ***“First Cryptographic Signature” [1C.3c]***

136. In Hearn’s Example 7 (step 7), the “client signs T3 and provides the transaction **and signature** to the server.” (CB-1009, 5.) A POSITA would therefore understand that initial T3 includes “*the first cryptographic signature*” [1C.3c] computed in [1C.2].

(iv) ***“Publish the Inchoate Data Record” [1C.4]***

137. Claim 1 recites “*publish[ing] the inchoate data record to at least one of the first client device or the second client device*” [1C.4]. In the “Glossary” provided in the ’566 patent, the ’566 patent describes “publish[ing] [a] transaction record to [a device]” as “[m]aking the transaction record available for reading or copying by the device, for example, by sending the transaction record to the device via a network interface.” (CB-1001, 37:23-35.) I note that the ’566 patent treats the concept of “*publishing*” broadly.

138. The combination Hearn and Armstrong teach “*publishing*” the initial version of transaction T3 (“*inchoate data record*”). In Hearn Example 7, after initial T3 (“*the inchoate data record*”) is created, the “client signs T3” using Party A’s private key and “**provides the transaction and signature to the server.**” (CB-1009, 5 (step 7).) In the combination of Hearn and Armstrong, initial transaction T3 is provided with Party A’s signature to Party B’s payment module (e.g., the second host computer or Party B’s user device; “*first client device*”). The sending

of the initial version of the payment transaction T3 to the component acting as Party B's payment module ("*first client device*") "*publish[es] the inchoate data record*" under the '566 patent's definition of "*publishing*" because it sends the record to the "*first client device via a network interface.*"

139. The combination of Armstrong and Hearn therefore discloses that the "*processor*" of the "*computing device*" [first host computer] "*publish[es] the inchoate data record to at least one of the first client device [second host computer] or the second client device*" [1C.4].

f. "*Distributed Ledger*" limitation [1D]

140. The combination of Hearn and Armstrong discloses that "*the decentralized digital currency comprises a distributed ledger that enables processing the transaction between the first client device and the second client device without the need for a trusted central authority*" [1D]

141. It was well-known long before the '566 patent that bitcoin is a decentralized peer-to-peer network that uses a public transaction ledger (the blockchain). (CB-1012, 1-3.) When Satoshi Nakamoto published his famous whitepaper on Bitcoin, he envisioned "[a] purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution." (CB-1019, Abstract.) That is,

Satoshi Nakamoto envisioned a decentralized (peer-to-peer) version of digital currency (electronic cash). (CB-1019, Abstract.)

142. Armstrong explains that in Bitcoin, “[w]hen a transaction is made between two wallets at the same or different host computer systems, the transaction is broadcast to the Bitcoin network for **block chain verification**.” (CB-1006, ¶10; CB-1007, ¶7; CB-1008, ¶7.) Specifically, “[p]articipants known as ‘miners’ at miner computer systems verify and timestamp transactions into a shared public database called a ‘block chain’.” (CB-1006, ¶7; CB-1007, ¶4; CB-1008, ¶4.)

143. The ’566 patent admits in its admitted prior art section that the “block chain” is a “public ledger” (a “distributed database”) that “records bitcoin transactions” without the need for “any trusted central authority.” (CB-1001, 3:63-64, 4:5.) This admitted prior art is consistent with the discussion of Antonopoulos describing the blockchain as “the distributed ledger of all transaction.” (CB-1012, 9.)

144. Hearn’s micropayment protocol broadcasts two transactions between Party A and Party B to the Bitcoin network for verification (complete T1 and final, complete T3). As taught by both Hearn and Armstrong, the Bitcoin network and its distributed ledger (blockchain) enable processing of the two transactions by the distributed network of mining computers and does not need a trusted central authority. The combination of Hearn and Armstrong therefore discloses “*the*

decentralized digital currency [bitcoin] comprises a distributed ledger [blockchain] that enables processing the transaction [payment transaction T3] between the first client device and the second client device without the need for a trusted central authority” [1D].

g. “Create a Complete Data Record ...” [1E]

145. The combination of Hearn and Armstrong discloses that “*the inchoate data record is used by at least one of the first client device or the second client device to create a complete data record and to create the transaction by broadcasting the complete data record for transmitting and receiving among network participants in the computer network for recording in the distributed ledger*” [1E].

146. In the payment stage of Hearn’s micropayment protocol as implemented in Armstrong, the first host computer, acting as Party A’s payment module, sends the initial version of payment transaction T3 (“*inchoate data record*”) to the component acting as Party B’s payment module which is the second host computer or Party B’s user device. (CB-1009, 5 (step 7).) The first host computer device then “adjusts its copy of T3 to allocate more value to the server’s output and less to its ow [sic]” as Party A (client) consumes the service offered by Party B (server). (*Id.* (step 9).) The first host computer “re-signs the new T3” using Party A’s private key and “sends the signature to the server.” (*Id.*) The component

acting as Party B's payment module then "adjusts its copy of T3 to match the new amounts, verifies the signature and continues." (*Id.*) Party A continues to make payments during the contractual period. Therefore, this incremental payment process which adjusts T3 "continues until the session ends" or the period set by the expiration timestamp "is getting close to expiry." (*See* CB-1009, 5.) At the end of the session, Party B's (server's) payment module "**signs and broadcasts** the last transaction it saw, allocating the final amount to itself." (*Id.*) The final version of T3 signed by both parties and broadcast to the Bitcoin network for verification is a "*complete data record*." In this way, the component acting as Party B's payment module (second host computer or Party B's user device) "*create[s] a complete data record*" from the initial version of T3 ("*inchoate data record*").

147. Armstrong explains that "[w]hen a transaction is made between two wallets at the same or different host computer systems, the transaction is **broadcast to the Bitcoin network** for block chain verification." (CB-1006, ¶10; CB-1007, ¶7; CB-1008, ¶7.) As part of block chain verification, "'miners' at miner computer systems verify and timestamp transactions into a shared public database called a 'block chain'." (CB-1006, ¶7; CB-1007, ¶4; CB-1008, ¶4.) After the transactions are added to the ledger, the ledger additions are communicated to other nodes. (*See*, CB-1006, ¶102; CB-1007, ¶53; CB-1008, ¶68; CB-1001, 3:63-4:5 (applicant admitted prior art related to block chain).) Thus, the combination of Armstrong and

Hearn discloses “*creat[ing] the transaction by broadcasting the complete data record for transmitting and receiving among network participants in the computer network for recording in the distributed ledger*” [1E].

h. “Sign[]” and “Save a Copy of the Inchoate Data Record” limitation [1F]

148. The combination of Hearn and Armstrong discloses that “*at least one of the first client device or the second client device signs the inchoate data record and saves a copy of the inchoate data record on at least one of the first client device or the second client device*” [1F].

149. In Hearn’s Example 7, after Party B (server) receives the initial version of payment transaction T3, the first host computer (“*computing device*”) adjusts its copy of T3 and sends “the signature and the amount to increment” to Party B (server) as Party A (client) uses the service. (CB-1009, 5 (step 9).) This continues periodically throughout the contract as I have discussed previously. Party B (server) then “adjusts **its copy of T3** to match the new amounts.” (*Id.*) That is, the component acting as Party B’s payment module (second host computer or Party B’s user device) stores a copy of the initial version of T3 (“*inchoate data record*”) so that it can be updated when incremental payments are received from Party A (client). Thus, in the combination of Hearn and Armstrong, the component acting as Party B’s payment module (second host computer or Party B’s user device; “*first*

client device”) “*saves a copy of the inchoate data record on at least one of the first client device ...*” [1F].

150. When the session ends (contractual period expires), Party B’s payment module “**signs and broadcasts** the last transaction” (i.e., the final version of T3) to the Bitcoin network. (CB-1009, 5.) The final version of T3 is the initial version of T3 adjusted based on the incremental updates received. Thus, in the combination of Hearn and Armstrong, the component acting as Party B’s payment module (second host computer or Party B’s user device) “*signs the inchoate data record*” [1F].

i. “*Verify*] the Recording of the Complete Data Record in the Distributed Ledger” limitation [1G]

151. The combination of Hearn and Armstrong discloses that “*the at least one of the computing device, the first client device, or the second client device verifies the recording of the complete data record in the distributed ledger by observing an external state*” [1G].

152. As part of blockchain verification, “‘miners’ at miner computer systems verify and timestamp transactions into a shared public database called a ‘block chain’.” (CB-1006, ¶7; CB-1007, ¶4; CB-1008, ¶4.) In Armstrong, a “block chain verification notice” is broadcast by miner computer system 28. (CB-1006, ¶102; CB-1007, ¶53; CB-1008, ¶68.) The components of Armstrong’s

network—including Party A’s payment module, first host computer, (the “*computing device*”) and Party B’s payment module (“*first client device*”)—monitor for these notices. (CB-1006, ¶102; CB-1007, 53; CB-1008, ¶68.) When a notice is detected, the components determine whether to change a “marker” for a transaction, such as payment transaction T3, from “‘PENDING’ to ‘COMPLETE’” in the system. (CB-1006, ¶102; CB-1007, 53; CB-1008, ¶68.) This is shown in Figure 43 below illustrating a set of transactions as “complete” and one as “pending.” The notice, originating for a source external to the host, provides “*an external state*” indicating whether the “*complete data record*” (final T3) has been recorded in the blockchain. By monitoring for these notices and reviewing their contents, the first host computer system and the component acting as Party B’s payment module “*observe*” the “*external state*” reflective of whether the transaction was recorded in the blockchain—i.e., by “*observing*” the “*external state*” it “*verifies the recording of the complete data record in the distributed ledger.*”.

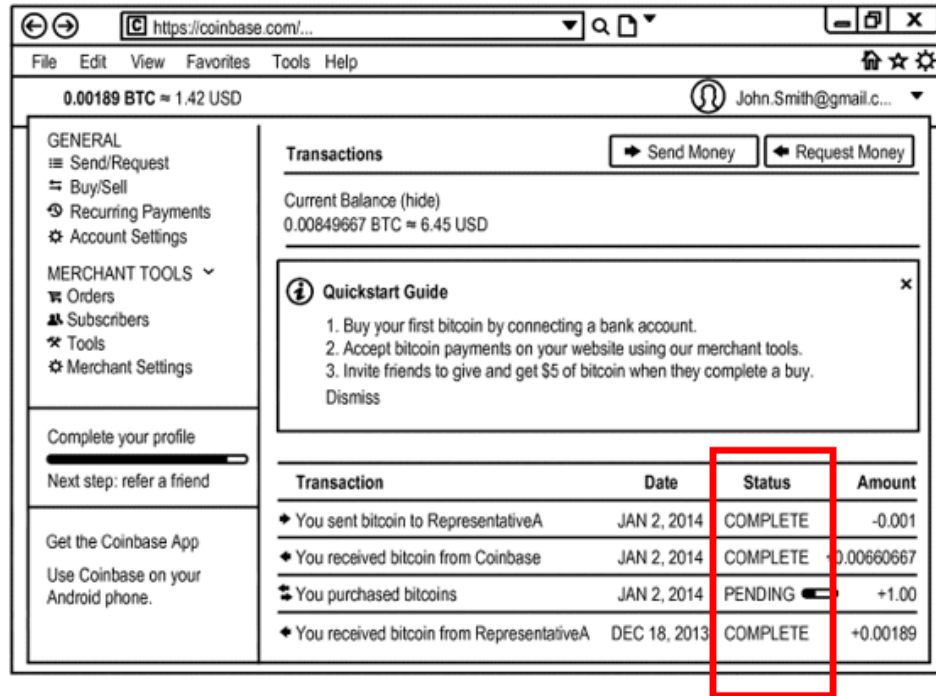


FIG. 43

Armstrong, Annotated Figure 43

153. Thus, in the combination of Armstrong and Hearn, both the first host computer (*“the computing device”*) and the second host computer (*“the first client device”*) *“verif[y] the recording of the complete data record in the distributed ledger by observing an external state”* [1G].

E. Dependent Claim 2

154. Claim 2 depends from claim 1. For ease of discussion, I have reproduced claim 2 below. Claim 2 narrows limitation [1C.3b] of claim 1 by specifying *“the one or more output data”* is obtained from *“the first principal data”* rather than from either the *“first principal data or the second principal data”* and

from “*the value data from the first data source*” rather than “*the value data from the first data source or the second data source.*” As I have discussed in §V.D.1.e.iii.b, the “*output data*” is obtained from the “*first principal data*” and “*value data from the first data source.*” Therefore, for the reasons set forth in §V.D, the combination of Hearn and Armstrong also renders claim 2 obvious.

The device of claim 1, where:

[2A] *the computer processor is configured to obtain the one or more output data based on:*

[2A.1] *the first principal data; and*

[2A.2] *the value data from the first data source.*

F. Dependent Claim 3

155. Claim 3 depends from claim 1. For ease of discussion, I have reproduced claim 3 with labels below. Claim 3 is directed to a refund transaction. I start my discussion with limitation [3B] and address limitation [3A] with limitation [3B.3] reciting “*the second cryptographic signature.*”

The device of claim 1, where the computer processor is further configured to:

[3A] *compute a second cryptographic signature from the first private key;*

[3B] *create an another inchoate data record comprising:*

[3B.1] *a commit input for receiving the commit data from the commit transaction;*

[3B.2] *a refund output comprising a refund data;*
[3B.3] *the second cryptographic signature; and*
[3B.4] *a lock time; and*
[3C] *publish the another inchoate data record to at least one of the first client device or the second client device.*

156. The combination of Hearn and Armstrong discloses the “*computer processor*” of the first host computer “*create[s]an another inchoate data record*” [3B]. As I discussed in §V.A, parties to a micropayment contract collaborate to create a signed refund transaction, T2, that spends the maximum payment value for the contract back to Party A. Specifically, after creation of “escrow” transaction T1, the first host computer (“*computing device*”) creates a refund transaction (T2) “which sends all the money back” to Party A. (CB-1009, 5 (step 3).) Refund transaction T2 is an “*inchoate data record*” until all the conditions are satisfied to make it final—it is signed by both parties, the time lock for the transaction expires, and the transaction is broadcast to the Bitcoin network for verification.

157. Hearn’s transaction T2 “is connected to T1.” (CB-1009, 5 (step 3).) That is, the unspent output of “escrow” transaction T1 is the “*input*” for the refund transaction, as illustrated in my Figure D below. As discussed in §V.D.1.e.iii.a, the “escrow” transaction T1 is a “*commit transaction*” because it commits Party A to transfer an amount of BTC (the recited “*commit data*”). Refund transaction T2

therefore also includes “a commit input for receiving a commit data from the commit transaction” [3B.1].

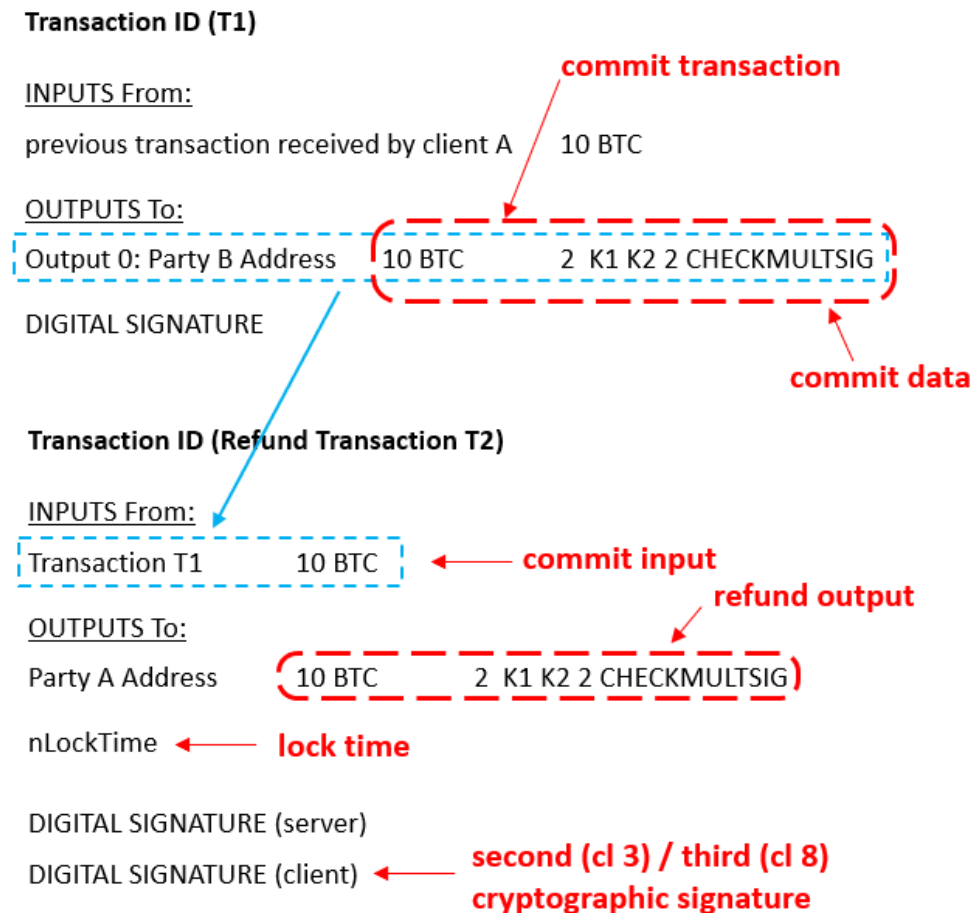


Figure D

158. Because refund transaction T2 “sends all the money” from the “escrow” transaction “back” to Party A (client), the value of the maximum total payment for the contract specified in T1 is also the output value in the refund

transaction. (See CB-1009, 5 (step 3).) Thus, refund transaction T2 includes “*a refund output comprising a refund data*” [3B.2].

159. Hearn’s refund transaction also includes a “time lock set for some time in the future, for instance a few hours.” (CB-1009, 5.) I note that the ’566 patent admits in its admitted prior art section that use of lock times was known before its priority date. Specifically, in the prior art section, the ’566 patent states:

Every transaction record can have a “lock time” associated with it. This prevents the transaction from being accepted as valid and allows the transaction to be pending and replaceable until an agreed-upon future time. In the case of the Bitcoin and similar protocols, this can be specified either as a block index or as a timestamp. The transaction record will not be accepted for inclusion in the block chain until the transaction's lock time has been reached.

(CB-1001, 4:57-65.) Thus, refund transaction T3 in the combination of Hearn and Armstrong includes “*a lock time*” [3B.4].

160. The input of the refund transaction is tied to the output of T1. Therefore, refund transaction T2 requires “**two signatures to be spent**”—the signature of Party A (client) and signature of Party B (server). (CB-1009, 5.) The first host computer obtains Party B’s signature by sending T2 to Party B (server) which “signs T2” and “returns the signature to the client” (step 4). At this point in the protocol, Party A (client) has the refund transaction T2 with one signature

(Party B's signature). Hearn further teaches that when the lock time condition is met Party A (client) "can broadcast the refund transaction and get back all the money." (CB-1009, 5.) A POSITA would understand Party A (client) must sign the refund transaction before the refund transaction is broadcast or the transaction would be deemed invalid by the Bitcoin network because T2 requires both signatures to be spent. For these reasons, refund transaction T2 includes a "*second cryptographic signature*" [3B.3] and the "*computer processor*" of the first host computer ("*the computing device*") must "*compute a second cryptographic signature from the first private key*" [3A] of Party A.

161. In the combination of Hearn and Armstrong, the first host computer holds signed refund transaction T2 until the necessary conditions are met, including the expiration of the time lock. (CB-1009, 5.) T2 remains incomplete until the conditions are met. For example, Hearn teaches the refund transaction is only finalized by broadcasting T2 to the Bitcoin network in cases "where the server disappears or halts at any point, leaving the allocated value in limbo." (CB-1009, 5.) Thus, in the combination of Hearn and Armstrong, the first host computer ("*computing device*") must identify the condition when the server becomes either unavailable or uncooperative during the micropayment contract.

162. A POSITA would have been motivated to inform Party A (client) when this condition occurs by posting the refund transaction on the user interface

(e.g., the interface shown in Figure 43) with a status of Pending. Alternatively, a POSITA would have been motivated to simply post all the transactions associated with a micropayment contract when the contract is initiated. The user interface would then show the escrow transaction T1, the refund transaction T3, and the current version of the payment transaction T3. The status of a refund transaction will be listed as pending until expiration of the contract and expiration of the lock time. A pending refund would simply be cancelled if the contract completes successfully, meaning that the refund transaction is not and cannot successfully be validated by the Bitcoin network.

163. Armstrong supports such notification of the status of transaction, teaching the host computer provides a webpage displaying to a user “transactions that have been initiated or completed” and illustrating this in Figure 42. (CB-1006, ¶102; CB-1007, ¶53; CB-1008, ¶68.) A POSITA would have been motivated to display the refund transaction T2 on Armstrong’s page with a status of pending when the client reports non-performance or during every micropayment contract so the user can track the status of all the transactions associated with its micropayment contract. The refund transaction remains posted as “pending” during the contractual time period and before the lock time expires. When the lock time expires and the refund transaction T2 broadcast, the status changes to “complete.”

164. The display of a pending refund transaction includes details of T2 and therefore is based on the contents of the “*inchoate data record*” associated with T2. In the “Glossary” provided in the ’566 patent, the ’566 patent describes “publish[ing] [a] transaction record to [a device]” as “[m]aking the transaction record available for reading or copying by the device, for example, by sending the transaction record to the device via a network interface.” (CB-1001, 37:23-35.) As I noted previously the ’566 patent treats the concept of “*publishing*” broadly. The action of displaying T2 “*publish[es] the another inchoate data record*” because the inchoate data record is stored in a way that permits the record to be read by the client device associated with Party A (“*the second client device*”). This display meets the ’566 patent’s definition of publishing—“making the transaction record available for reading or copying by the device, for example, by sending the transaction record to the device via a network interface (190), or writing the transaction record to a transaction sector in a memory in such a way that the transaction record can be read or copied by the device, optionally implementing a permissions scheme allowing the device to read or copy, but not create, update, or destroy the transaction record.” (CB-1001, 37:23-31.) The combination of Hearn and Armstrong teaches or suggests to a POSITA “*publish[ing] the another inchoate data record to at least one of the first client device or the second client device*” [3C].

VI. GROUND 2: THE COMBINATION OF ARMSTRONG, HEARN, AND ZIEGLER RENDERS CLAIMS 7 AND 8 OBVIOUS.

165. As I discussed in §V, the combination of Hearn and Armstrong teaches or suggests every limitation of claim 1. The combination does not explicitly disclose that the “*second client device*” includes a memory for “*storing a third asymmetric key pair*” [7C.1] and a processor “*configured to read the third private key from the third memory*” [7C.3]. However, the storage and use of private keys in a networked device, such as the “*second client device*”, were well-known and taught by Ziegler.

A. Overview of the Combination

1. Ziegler

166. Ziegler relates to “secure information transfer.” (CB-1010, 1:12-13.) Ziegler’s information transfer system, illustrated in Figure 2, includes computer 102 coupled to Web server 120 which provides an interface to application server 122 “connected to a host cryptographic module 124” (“Host”). (CB-1010, 3:33-37.)

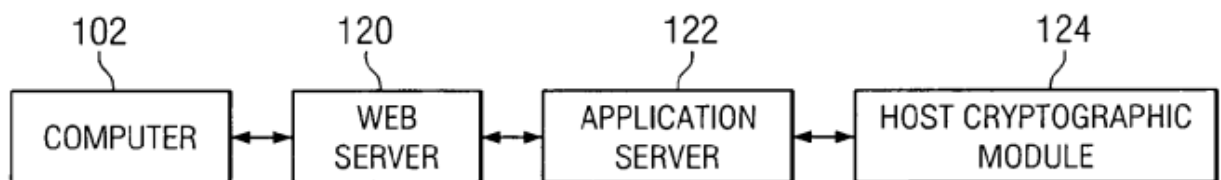


FIG. 2

Ziegler, Figure 2

167. After a user accesses a website, an “SSL secured session is established between the Web server and the client PC” and the system’s software is downloaded “and installed on the client.” (CB-1010, 4:31-40.) Next, an “RSA Public/Private Key pair (1024) is generated” at the client. (CB-1010, 4:47-48.) An RSA Public/Private key pair is an asymmetric key pair. The Public Key is sent to the Web Server and shared with the Application server and Host. (CB-1010, 4:48-50.) The Public/Private Key pair for the client is subsequently used to establish session (symmetric) keys for communication between the client and server. (CB-10101, 4:46-47.)

168. Specifically, the Host generates symmetric keys used to secure communications and used to secure data. (CB-1010, 4:50-55.) To securely share these keys with client, the Host creates two cryptograms—(1) the communications key and a hash of the communication key, encrypted (wrapped) with the client’s RSA public key and (2) the data key and a hash of the data key, encrypted (wrapped) with the client’s RSA public key and encrypted with the communications key. (CB-1010, 4:56-67.) The Host then combines these two cryptograms, “generates a cryptographic hash”, encrypts the combination and its hash with the client’s public key, and sends this resulting cryptogram (referred to as the “key package”) to the client. (CB-1010, 5:1-9.)

169. After the client receives the “key package”, the client “decrypts the key package with its Private Key” to obtain the two cryptograms. (CB-1010, 5:10-11.) The client then decrypts the first cryptogram using its private key and verifies its integrity. (CB-1010, 5:17-21.) The client then decrypts the second cryptogram with the communications key and with the client’s private key and verifies its integrity. (CB-1010, 5:21-26.) The client now has the communications key and data key “for further communication with the Application Server” during a session. (CB-1010, 5:26-28.)

2. Combination of Hearn, Armstrong, and Ziegler

170. In §V.C, I explained the combination of Hearn and Armstrong and the motivations to combine the teachings of Hearn and Armstrong. A POSITA would have been motivated to use Ziegler’s secure information transfer technique for communication between Armstrong’s second user device (“*second client device*”) and the host computer in the combined system of Hearn and Armstrong.

171. Financial transactions have always been targets of malicious individuals attempting to defraud one or more of the parties. This is especially true with cryptocurrency and blockchain technologies transactions are often between parties that do not know or trust one another. A POSITA would have therefore been motivated to combine Ziegler with Hearn and Armstrong to provide secure communications for financial transactions between a user device and the host

computer to prevent fraud and/or disclosure of client confidential information. For example, a POSITA would have been motivated to prevent an attack where a party eavesdrops to obtain information from communications between Party A and the host which allows a malicious party to masquerade as Party A to set up a contract with Party B that transfers significant funds from Party A to Party B.

172. Ziegler also explicitly suggests the combination, noting “[a]ny system implementing an electronic transfer of funds, for example, requires multi-level security measures.” (CB-1010, 1:19-21.) Specifically, “transactions between strangers and more particularly, transactions conducted on an open-network like the Internet cannot be conducted within reasonable risk limitations without some kind of identity verification.” (CB-1010, 1:28-31.) Because Armstrong uses a web application that communicates sensitive financial information between a user device and the host over the Internet and Hearn’s contracts are designed for use between strangers (trustless transactions), a POSITA would have been motivated to incorporate additional communications security to remove risks.

173. The combination of Ziegler with Hearn and Armstrong is also the combination of a known technique (Ziegler’s secure communication technique) to improve similar devices (Hearn and Armstrong’s systems for transacting financial transactions) in the same way (use of secure communications). The results of the

combination would have been predictable because secure communication protocols were well known and in use long before the '566 patent. (*See* CB-1014, 4-5, 283.)

B. Independent Claim 7

174. The combination of Hearn, Armstrong, and Ziegler teaches every limitation of claim 7 which I discuss in further detail in this section. For ease of discussion, I have provided labels for the preamble and limitations of claim 7.

[7P] A system for processing a transaction between a first client device and a second client device via a transfer mechanism, the system comprising a computing device, the first client device, the second client device, and the transfer mechanism;

[7A] the computing device comprising:

[7A.1] a first memory comprising [sic] for storing a first asymmetric key pair, the first asymmetric key pair comprising a first private key and a first public key;

[7A.2] a first network interface for receiving terms, the terms comprising:

[7A.2a] at least one of a first principal data or a second principal data

[7A.2b] a reference to at least one of a first data source or a second data source; and

[7A.2c] an expiration timestamp; and

[7A.3] a first computer processor coupled to the first memory and the first network interface, the first computer processor configured to:

[7A.3a] read the first private key from the first memory;

[7A.3b] *compute a first cryptographic signature from the first private key;*

[7A.3c] *create an inchoate data record comprising:*

[7A.3c.i] *a commit input for receiving a commit data from a commit transaction;*

[7A.3c.ii] *one or more outputs obtained from at least one of the first principal data or the second principal data, and a value data from at least one of the first data source or the second data source; and*

[7A.3c.ii] *the first cryptographic signature; and*

[7A.3d] *publish the inchoate data record to at least one of the first client device or the second client device;*

[7B] *the first client device comprising:*

[7B.1] *a second memory for storing a second asymmetric key pair, the second asymmetric key pair comprising a second private key and a second public key;*

[7B.2] *a second network interface; and,*

[7B.3] *a second computer processor coupled to the second memory and the second network interface, the second computer processor configured to:*

[7B.3a] *read the second private key from the second memory;*

[7B.3b] *read the inchoate data record;*

[7B.3c] *compute a second cryptographic signature from the second private key;*

[7B.3d] *create a complete data record comprising:*

[7B.3d.i] *the commit input;*

[7B.3d.ii] *the output data;*

[7B.3d.iii] *the first cryptographic signature; and*

[7B.3d.iv] *the second cryptographic signature; and*

[7B.3e] *create a transaction by submitting the complete data record to the transfer mechanism;*

[7C] *the second client device comprising:*

[7C.1] *a third memory for storing a third asymmetric key pair, the third asymmetric key pair comprising a third private key and a third public key;*

[7C.2] *a third network interface; and,*

[7C.3] *a third computer processor coupled to the third memory and the third network interface, the third computer processor configured to read the third private key from the third memory; and*

[7D] *wherein the at least one of the first client device or the second client device signs the inchoate data record and saves a copy of the inchoate data record on at least one of the first client device or the second client device,*

[7E] *wherein the transfer mechanism comprising a decentralized digital currency that comprises a distributed ledger that enables processing the transaction between the first client device and the second client device without the need for a trusted central authority,*

[7F] *wherein the transaction is created by broadcasting the complete data record for transmitting and receiving among network participants in the computer network for recording in the distributed ledger, and*

[7G] *wherein at least one of the computer device, the first client device, or the second client device verifies the recording of the complete data record in the distributed ledger by observing an external state.*

a. Preamble [7P]

175. I discussed in §V.C that the combination of Hearn and Armstrong discloses a “*computing device for processing a transaction between a first client device, and a second client device via a transfer mechanism*” [1P]. The preamble of claim 7 recites overlapping subject matter using slightly different language: “[*a*] *system for processing a transaction between a first client device and a second client device via a transfer mechanism, the system comprising a computing device, the first client device, the second client device, and the transfer mechanism*” [7P]. Because claim 1 addresses a “*computing device*” in a system having a “*first*” and “*second*” client device and a “*transfer mechanism,*” the combination of Hearn, Armstrong, and Ziegler discloses the preamble for the same reason as set forth for preamble [1P].

b. “Computing Device” limitations [7A]

176. Claim 7 recites a “*computing device*” [7A] and a set of limitations directed to the “*computing device*” [7A.1]-[7A.3]. These limitations are identical to limitations of claim 1 as set forth in the following chart. For the reasons I have

set forth in the section cited in the table, the combination of Hearn, Armstrong, and Ziegler teach the “*computing device*” limitations of claim 7.

Claim 7	Claim 1	Cite
[7A.1] <i>a first memory comprising [sic] for storing a first asymmetric key pair, the first asymmetric key pair comprising a first private key and a first public key;</i>	[1A] <i>a memory for storing a first asymmetric key pair, the first asymmetric key pair comprising a first private key and a first public key</i>	<i>See §V.B.2</i>
[7A.2] <i>a first network interface for receiving terms, the terms comprising:</i>	[1B] <i>a network interface for receiving terms, the terms comprising:</i>	<i>See §V.B.3</i>
[7A.2a] <i>at least one of a first principal data or a second principal data</i>	[1B.1] <i>at least one of a first principal data or a second principal data;</i>	<i>See §V.B.4.a</i>
[7A.2b] <i>a reference to at least one of a first data source or a second data source; and</i>	[1B.2] <i>a reference to at least one of a first data source or a second data source; and</i>	<i>See §V.B.4.b</i>
[7A.2c] <i>expiration timestamp; and</i>	[1B.3] <i>an expiration timestamp;</i>	<i>See §V.B.4.c</i>
[7A.3] <i>a first computer processor coupled to the first memory and the first network interface, the first computer processor configured to:</i>	[1C] <i>a computer processor coupled to the memory and the network interface, the computer processor configured to:</i>	<i>See §V.B.5</i>
[7A.3a] <i>read the first private key from the first memory;</i>	[1C.1] <i>read the first private key from the memory;</i>	<i>See §V.B.6.a</i>
[7A.3b] <i>compute a first cryptographic signature from the first private key;</i>	[1C.2] <i>compute a first cryptographic signature from the first private key;</i>	<i>See §V.B.6.a</i>
[7A.3c] <i>create inchoate data record comprising;</i>	[1C.3] <i>create an inchoate data record comprising:</i>	<i>See §V.B.6.b</i>

[7A.3c.i] <i>a commit input for receiving a commit data from a commit transaction;</i>	[1C.3a] <i>a commit input for receiving a commit data from a commit transaction;</i>	<i>See §V.B.6.b.i</i>
[7A.3c.ii] <i>one or more outputs obtained from at least one of the first principal data or the second principal data, and a value data from at least one of the first data source or the second data source; and</i>	[1C.3b] <i>one or more output data obtained from at least one of the first principal data or the second principal data, and a value data from at least one of the first data source or the second data source;</i>	<i>See §V.B.6.b.ii</i>
[7A.3c.iii] <i>the first cryptographic signature; and</i>	[1C.3c] <i>and the first cryptographic signature; and</i>	<i>See §V.B.6.b.iii</i>
[7A.3d] <i>publish the inchoate data record to at least one of the first client device or the second client device;</i>	[1C.4] <i>publish the inchoate data record to at least one of the first client device or the second client device</i>	<i>See §V.B.6.c [1C.4]</i>

c. “First Client Device” limitations [7B]

(i) Components [7B.1]-[7B.3]

177. Claim 7 includes a set of limitations directed to the “*first client device*”:

[7B.1] *a second memory for storing a second asymmetric key pair, the second asymmetric key pair comprising a second private key and a second public key;*

[7B.2] *a second network interface; and*

[7B.3] *a second computer processor coupled to the second memory and the second network interface,*

The combination of Hearn, Armstrong, and Ziegler discloses these limitations.

178. As discussed in §V.D.1, Armstrong's second host computer is the recited "*first client device*." Should an argument be made that a host computer is not a "*client device*", the combination further teaches that Party B's user device acts as its payment module. In this alternative, Party B's user device is the "*first client device*." Party B's payment module operates in the same manner when Party B is the recipient of Hearn's micropayment contract, regardless of where it is physically located. Each device in Armstrong's network, including the second host computer and stand-alone user devices, utilize the "computer system" architecture illustrated in Figure 74. (CB-1006, ¶226 (machine "may operate in the capacity of ... a peer machine in a peer-to-peer (or distributed) network environment"), ¶226; CB-1007, ¶58; CB-1008, ¶93.) Armstrong's computer system includes "main memory 932 (e.g., read-only memory (ROM), flash memory, dynamic random-access memory (DRAM) such as synchronous DRAM (SDRAM) or Rambus DRAM (RDRAM), etc.), and a static memory 934 (e.g., flash memory, static random access memory (SRAM, etc.)", "network interface device 948", and processor 930 coupled to memory 932, memory 934, and network interface device 948 via bus 936. (See CB-1006, ¶226-227; CB-1007, ¶59-60; CB-1008, ¶94-95.)

179. Armstrong discloses that the “*first client device*” comprises “*a second memory*” [7B.1], “*a second network interface*” [7B.2], and “*a second computer processor coupled to the second memory and the second network interface*” [7B.3].

180. The “*memory*” in the component acting as Party B’s payment module “*stor[es] a second asymmetric key pair.*” As I discussed in §V.D.1.a, a wallet “stores the public key of the Bitcoin address and its associated private key.” (CB-1006, ¶8; CB-1007, ¶5; CB-1008, ¶5.) This public/private key pair is known as “*an asymmetric key pair.*” Hearn’s micropayment contract protocol (Example 7) uses the public/private key pair for Party B. Specifically, Party A “[r]equests a public key from the server.” (CB-1009, 5 (step 1).) Thus, Hearn discloses Party B has a “*public key.*” Because a public key has an associated private key, a POSITA would further understand the server also has a “*private key.*” (See §III.A and III.D.)

(ii) “*Second Computer Processor*” action limitations.

181. As I discussed in §V.A.3, in the combination of Hearn and Armstrong, the component acting as Party B’s payment module (“*first client device*”) is configured to include logic (software) to process Hearn’s micropayment contract. Therefore, this component implements Hearn’s micropayment protocol including logic for processing the required transactions (T1, T2, and T3). Accordingly, the combination of Hearn and Armstrong teaches or suggests the “*processor*” of the

component acting as Party B's payment module (second host computer or Party B's user device; "*first client device*") is configured to perform the functions recited in limitations [7B.3a]-[7B.3e] as detailed below.

(a) ***“Read the Second Private Key” limitation [7B.3a], “Read the Inchoate Data Record” limitation [7B.3b], “Compute a Second Cryptographic Signature” limitation [7B.3c]***

182. The combination of Hearn and Armstrong teaches the action limitations associated with the “*second computer processor*”: “*read[ing] the second private key from the second memory*” [7B.3a], “*read[ing] the inchoate data record*” [7B.3b], and “*comput[ing] a second cryptographic signature from the second private key*” [7B.3c].

183. In Hearn, Party A (client) provides the initial version of T3 (“*inchoate data record*”) to Party B (server). (CB-1009, 5 (step 7).) As I discussed in §V.B.9, the component acting as Party B's payment module stores a copy of the initial version of T3. Periodically, during the time period of the contract, Party A “adjusts its copy of T3 to allocate more value to the server's output and less to its ow [sic]” and sends the “signature and the amount to increment” T3 to Party B (server). (CB-1009, 5 (step 9).) Party B (server) then “adjusts its copy of T3 to match the new amounts.” (*Id.*) To be able to adjust its copy of T3, the component acting as Party

B's payment module (the second host computer or Party B's user device) must "*read[s] the inchoate data record*" (the initial version of T3) [7B.3b].

184. When the session ends (contract expires), Party B (server) "signs and broadcasts the last transaction" (i.e., the final version of T3). (CB-1009, 5.) The digital signature is created using Party B's private key as I discussed in Ground 1 analysis and in the Technical Background section. Accordingly, Hearn teaches use of a "*private key*" to "*compute a second cryptographic signature*" [7B.3c].

185. The "*memory*" of the "*first client device*" stores the "*private key*" and "*public key*" associated with Party B (server). A POSITA would have therefore understood the "*second private key*" of Party B (server) must be "*read*" [7B.3a] from "*the second memory*" to compute the second cryptographic signature for the final version of transaction T3.

(b) "*Create a Complete Data Record*" [7B.3d]

186. Claim 1 recites that "*the inchoate data record is used by ... the first client device ... to create a complete data record*" [1E]. For the reasons that I discussed in §V.D.1.g, the "*second computer processor*" of the "*the first client device*" "*create[s] a complete data record*" [7B.3d].

187. As I discussed in ground 1, the final version of T3 is the ("*complete data record*"). Transaction T3 "connects to T1." (CB-1009, 5.) That is, the output of T1 is the "*commit input*" [7B.3c.i] for all versions of transaction T3 including

the final version, as shown in my Figure E below. Final T3, like initial T3, has “two outputs”—one to Party A and one to Party B. (CB-1009, 5.) When the session ends or the expiration time for the contract “is close to expiry”, Party B (server) “signs and broadcasts the last transaction it saw, allocating the final amount to itself.” (CB-1009, 5.) That is, the maximum amount for the contract is allocated to Party B (server) in the final version of the T3 data record as shown below. The “two outputs” (BTC amounts) of final T3 are “*the output data*” [7B.3c.ii]. Because both parties sign the final version of T3, final T3 (“*complete data record*”) includes the “*first cryptographic signature*” [7B.3c.iii] and “*second cryptographic signature*” [7B.3c.iv].

Transaction ID (T1)

INPUTS From:

previous transaction received by client A 10 BTC

OUTPUTS To:

Output 0: Party B Address 10 BTC 2 K1 K2 2 CHECKMULTSIG

DIGITAL SIGNATURE

Transaction ID (Transaction T3 – first iteration)

INPUTS From:

Transaction T1 10 BTC ← **commit input**

OUTPUTS To:

at time t0

Output 1: K1 10 BTC

Output 2: K2 0 BTC

DIGITAL SIGNATURE (client)

at time t1

Output 1: K1 5 BTC

Output 2: K2 5 BTC

at time t2 (final)

Output 1: K1 0 BTC

Output2: K2 10 BTC

DIGITAL SIGNATURE (client – K1) ← **first cryptographic signature**

DIGITAL SIGNATURE (server – K2) ← **second cryptographic signature**

Figure E

**(c) “Create a Transaction” Limitation
[7B.3e]**

188. As discussed in §V.D.1.g, the component acting as Party B’s payment module (second host computer or Party B’s user device) “creat[es] the transaction

by broadcasting the complete data record ...for recording in the distributed ledger” [1E]. Party B “broadcasts” the final version of T3 to the Bitcoin network for verification. (CB-1009, 5; CB-1006, ¶10; CB-1007, ¶7; CB-1008, ¶7.) Armstrong’s Bitcoin network 12 is the “*transfer mechanism*” for the reasons I discussed in the preamble of claim 1. (See §§V.D.1, VI.B.a.)

189. For the same reasons that I discussed in §V.D.1.g, the combination of Hearn, Armstrong, and Ziegler discloses that the second host computer (“*first client device*”) “*create[s] a transaction by submitting the complete data record to the transfer mechanism*” [7B.3e].

d. “*Second Client Device*” limitations [7C]

190. Claim 7 recites a set of limitations directed to the “*second client device*” [7C]: “*a third memory*” [7C.1], “*a third network interface*” [7C.2], and “*a third computer processor*” [7C.3]. The combination of Hearn, Armstrong, and Ziegler discloses these limitations.

191. As I discussed in §V.D.1, Armstrong’s second user device is the “*second client device*.” Each device in Armstrong’s network utilizes the “computer system” architecture illustrated in Figure 74. (CB-1006, ¶225; CB-1007, ¶58; CB-1008, ¶93.) For the reasons I discussed in §V.D.1 and §§V.D.1.e.i-V.D.1.e.iii, Armstrong’s second user device (“*the second client device*”) includes “*a third*

memory” [7C.1], “*a third network interface*” [7C.2], and “*a third computer processor coupled to the third memory and the third network interface*” [7C.3].

192. In the combination of Hearn, Armstrong, and Ziegler, the “*second user device*” generates an “RSA Public/Private Key pair.” (CB-1010, 4:47-48.) This asymmetric key pair is stored in a memory at the client device. The private key is then read by the processor of the client device to perform decryption during communication and data key establishment. (CB-1010, 4:46-47; 5:10-28; *See* §VI.A.2.)

193. Accordingly, the combination of Hearn, Armstrong, and Ziegler teaches the “*second client device*” includes a memory for “*storing a third asymmetric key pair, the third asymmetric key pair comprising a third private key and a third public key*” [7C.1] and a processor “*configured to read the third private key from the third memory*” [7C.3].

e. “*Sign[]*” and “*Save a Copy of the Inchoate Data Record*” limitation [7D]

194. Limitation [7D] (“*wherein the at least one of the first client device or the second client device signs the inchoate data record and saves a copy of the inchoate data record on at least one of the first client device or the second client device*”) is identical to limitation [1F] (“*wherein at least one of the first client device or the second client device signs the inchoate data record and saves a copy*

of the inchoate data record on at least one of the first client device or the second client device.) For the reasons I discussed in §V.D.1.h, the combination of Hearn, Armstrong, and Ziegler discloses limitation [7D].

f. “*Distributed Ledger*” limitation [7E]

195. Limitation [7E] is substantially identical to limitation [1D]. The primary difference is that limitation [1D] recites that “*the decentralized digital currency comprises a distributed ledger*” whereas limitation [7E] recites “*the transfer mechanism comprising a decentralized digital currency that comprises a distributed ledger.*” As I discussed in §V.D.1 , “*the transfer mechanism compris[es] a decentralized digital currency.*” Accordingly, for the reasons I discussed in §§ V.D.1, V.D.1.f, the combination of Hearn, Armstrong, and Ziegler discloses limitation [7E].

g. “*Transaction is Created by Broadcasting*” limitation [7F]

196. Limitation [7F] is substantially identical to a portion of limitation [1E]:

[7F] *wherein the transaction is created by broadcasting the complete data record for transmitting and receiving among network participants in the computer network for recording in the distributed ledger*

[1E] *create the transaction by broadcasting the complete data record for transmitting and receiving among network participants in the computer network for recording in the distributed ledger*

The only minor difference is that claim 7 recites “*the transaction is created*” whereas claim 1 recites “*create the transaction.*” For the reasons discussed in §V.D.1.g, the combination of Hearn, Armstrong, and Ziegler also discloses limitation [7F].

h. “Verify] the Recording of the Complete Data Record in the Distributed Ledger” limitation [7G]

197. Limitation [7G] is identical to limitation [1G]. For at least the reasons I discussed in §V.D.1.i, the combination of Armstrong and Hearn discloses limitation [7G].

C. Dependent Claim 8

198. Claim 8 depends from claim 7. I have reproduced the language of claim 8 below. Claim 8 recites substantially the same subject matter as claim 3 with the sole difference being claim 8 recites a “*third cryptographic signature*” because independent claim 7 recites “*first*” and “*second*” cryptographic signatures. Accordingly, for the reasons I discussed in §V.F, the combination of Armstrong, Hearn, and Ziegler renders obvious claim 8.

[8P] *The system of claim 7, where the first computer processor is further configured to:*

[8A] *compute a third cryptographic signature from the first private key;*

[8B] *create another inchoate data record comprising:*

[8B.1] *a commit input for receiving the commit data from the commit transaction;*

[8B.2] *a refund output comprising a refund data; and*

[8B.3] *the third cryptographic signature; and*

[8C] *publish the another inchoate data record to at least one of the first client and the second client.*

VII. CONCLUSION

199. The findings and opinions set forth in this declaration are based on my work and analysis to date. I may continue my analysis. I may also receive additional documentation and other factual evidence over the course of this proceeding that will allow me to supplement and/or refine my opinions. I reserve the right to add to, alter, or delete my opinions and my declaration upon discovery of any additional information. I reserve the right to make such changes as may be deemed necessary.

200. In signing this declaration, I recognize that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-

examination within the United States during the time allotted for cross-examination.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Executed on this 30th day in March 2023 in Champaign, IL.

A handwritten signature in black ink, appearing to read "Andrew Miller". The signature is fluid and cursive, with the first name "Andrew" written in a larger, more prominent script than the last name "Miller".

Andrew Miller