Santa Clara High Technology Law Journal

Volume 37 | Issue 1 Article 1

1-2-2021

MAKING MISTAKES WITH MACHINES

Dhanoa. Harsimar

MAKING MISTAKES WITH MACHINES

By Harsimar Dhanoa

As adoption of machine-executed smart contracts increases, so too does the risk that the machines underlying these agreements will deviate from the intentions of the contracting parties. While contract law allows a narrow opportunity for setting aside of a contract under the doctrine of mistake, the application of this doctrine is muddied when the machines entirely operate within the confines of their programming. This paper highlights two notable instances of such deviation, the flash crash of 2010 and the 2016 attack on the blockchain project, the DAO, before focusing on the first case to address the doctrine of mistake in the context of these smart contracts: the Singaporean case of B2C2 Ltd v. Quoine Pte Ltd. While the Singapore International Commercial Court and the Court of Appeal of Singapore, the country's highest court, reached the correct solution, this paper argues that the Singaporean courts incorrectly limited themselves to only considering the knowledge and intentions of the programmer behind the smart contracts. Further, the paper suggests that a threshold inquiry offered by the American Restatement (2d) on Contracts may help to establish when a party using a smart contract has assumed the risks associated with its execution.

CONTENTS

INTRODUCTION

Freedom of contract is fundamental to modern contract law. The principle balances on one hand, parties' rights to bind themselves legally, even where the agreement "may not seem desirable or pleasant to outside observers," whether they range from mowing a lawn for a paltry sum or enabling a multi-million-dollar merger.[1] On the other hand, the principle requires parties to accept a possible bad bargain without court interference.[2] However, to quote Alexander Pope, "to err is human."[3] While courts have not been so divine as to forgive such mistakes entirely, both English and American contract law allow for the rescission of some contracts under the doctrine of mistake. This doctrine traditionally has been narrow, reflecting both a concern that parties feeling disappointed in their deal may concoct mistakes after the fact and a recognition that the avoidance of mistakes is "precisely the quality which marks the successful and efficient businessmen."[4] Thus, for courts whose "primary goal in interpreting contracts is to determine and enforce the parties' intent," when should mistake be accounted for and on what grounds?[5]

While the task of remedying mistakes is complex among humans alone, the increasing role of machines in the formation and execution of contracts presents an additional wrinkle to resolve. Imagine the following scenario: Hunter, a computer programmer, turns to their spouse, Dakota, and says, "I'm heading to the store. Any requests?" Dakota responds "Pick up a loaf of bread. If they have eggs, pick up a dozen." With Dakota's instructions in tow, Hunter goes to the store and returns an hour later with thirteen loaves of bread in hand. Upon seeing the loaves, an exasperated Dakota asks, "Why'd you buy so much bread?" In response, Hunter simply says: "They had eggs."

While legal scholars may disagree on how Hunter should have interpreted Dakota's ambiguous instructions, in the eyes of a machine (or at the least, a computer programmer thinking like a machine), the instructions are incredibly clear. Computers are exacting textualists that dutifully obey their instructions, even where those instructions

[1] Guiliano v. Cleo, Inc., 995 S.W.2d 88, 100 (Tenn. 1999).

[2] *Id.*, *see also* CHARLES FRIED, CONTRACT AS PROMISE: A THEORY OF CONTRACTUAL OBLIGATION 113 (1981) ("If we take autonomy seriously as a principle for ordering human affairs . . . people must abide by the consequences of their choices . . .").

[3] ALEXANDER POPE, AN ESSAY ON CRITICISM 30 (1711).

[4] P.B.H. BIRKS, THE ROMAN LAW OF OBLIGATIONS 76–77 (2014).

[5] Old Kent Bank v. Sobczak, 243 Mich.App. 57, 63 (2000).

deviate from the programmer's intent.[6] As a result, Hunter's actions appear absurd in the context of a person interpreting Dakota's instructions and yet perfectly reasonable in the context of a computer attempting to do the same.

As "smart contracts" become increasingly more common, the significance of the disconnect between the ways that humans and machines understand language magnifies.[7] In turn, courts, which are well versed in the application of contract doctrine to traditional contracts, will need to apply this doctrine in the context of these smart contracts.[8] More specifically, courts will need to resolve the tension that exists when a computer's execution of an agreement is in line with the objective programming of the smart contract, but not in line with the subjective intents of one of the parties to the contract. For example, if Dakota sought to reverse the purchase of the additional dozen loaves, how should a court reconcile the competing objective and subjective views of contract in the context of her smart contract? And if the court decides that Dakota is bound to the purchase of the additional loaves, what measures might ensure that future parties relying on smart contracts are mindful of the risks?

This paper seeks to explore these questions in the context of *B2C2 Ltd v. Quoine Pte Ltd*, a decision of first impression in which the Singapore International Commercial Court (SICC) analyzed the doctrine of mistake in the context of an algorithmic trading contract on a cryptocurrency exchange, and its subsequent affirmation by Singapore's highest court.[9] These decisions provide the first opportunity to analyze how courts in practice assess the contours of the contract doctrine of mistake in light of smart contracts. In reviewing these decisions, this paper builds on previous work to explore the

[6] In fact, interpreting Dakota's instructions literally, Hunter would have never returned home, having never been told to do so. Instead, they would have stood in the grocery store for eternity after having physically picked up the bread.

[7] While some scholars use the term "smart contract" to refer to agreements that utilize "blockchain" or distributed ledger technology, for the purposes of this paper, the term refers to a broader category of agreements whose formation and/or execution is automated, often "through a computer running code that has translated the [human language] legal prose into an executable program." *See* MAX RASKIN, THE LAW AND LEGALITY OF SMART CONTRACTS, 1 GEO. L. TECH. REV. 305, 309 (2017).

[8] *See, e.g.,* Mark Giancaspro, *Is a 'Smart Contract' Really a Smart Idea? Insights from a Legal Perspective*, 33 COMPUT. L. & SEC. REV. 825, 835 (2017) ("It is not yet entirely clear whether smart contracts are a smart idea, but there is little doubt the question will soon be tested in the courts.").

[9] *See generally* B2C2 Ltd v. Quoine Pte Ltd [2019] SGHC(I) 03, *aff'd* Quoine Pte Ltd v. B2C2 Ltd [2020] SGCA(I) 02.

disagreements between the SICC and Singapore's highest court, and to suggest that a threshold inquiry imported from American contract law to identify when the doctrine of mistake should be available to parties using smart contracts and when they instead have undertaken the risk of that outcome.

Part I provides a background to the rise of machine-executed smart contracts, using J.G. Allen's model of the contract stack to provide a comparison to traditional contracts. Part I additionally highlights four applications of these agreements and subsequent instances where the machine's execution diverged from the parties' intentions. Part II begins with the factual background of the dispute between B2C2 Ltd. and Quoine and discusses the Singapore International Commercial Court's application of the mistake doctrine as well as the subsequent decision to deny rescission, its affirmation by Singapore's highest court and compares the application of American contract common law. Part III discusses the implications of the B2C2 decision and whether the doctrine of mistake should treat machine-executed contracts differently.

I. BACKGROUND

A. The Rise of Machine-Executed Smart Contracts

The interaction between the natural language used by humans and the formal language used by machines is not new. Rather, the use of machines to form and execute contracts goes back nearly half a century.[10] However, as technology enables greater machine autonomy, computers increasingly operate not only as the mediums of communication for humans seeking to contract. Instead, these machines also act as active instruments in allowing parties to contract. As the technology develops further and becomes more sophisticated, computers may be able to act autonomously in forming contracts, requiring little to no human input as they negotiate complex terms beyond just price.[11] Because of their analytical sophistication, smart

[10] Electronic Data Interchange (EDI) systems—automated digital communications between or within firms—have been around since the 1970s. While the scope of these systems goes beyond contracting electronic communication, they facilitate particular types of contracts, such as purchase orders. *See generally* JANE K. WINN & BENJAMIN WRIGHT, LAW OF ELECTRONIC COMMERCE § 5-09 (4th ed. 2001).

[11] *See generally* Samir Chopra & Laurence White, *Artificial Agents and the Contracting Problem: A Solution Via an Agency Analysis*, 2009 U. ILL. J.L. TECH. & POL'Y 363 (2009). While some scholars raise questions as to application of contract law to artificially intelligent machines that extend this

contracts can process a vast amount of data thanks to advances in data collection and storage.[12] Moreover, advances in processing speed allow machines to achieve a level of sophistication that would otherwise be impractical for humans to perform given the time and effort required.[13]

Scholar J.G. Allen's model of the contract stack illustrates the parallels between traditional contracts and their "smart" equivalents.[14] The model utilizes the idea of a "software stack"—a set of software and hardware subsystems that are needed to implement a fully functional computing solution.[15] In computing, a simplified solution stack consists of the application as well as the programming environment, data management layer, operating system, and hardware platform.[16] With a "paper" contract, the contract is similarly comprised of multiple layers: "(i) the spoken words through which the contractual terms were negotiated and against which the text was drafted, (ii) the written text, and (iii) legal rules implying terms and governing construction."[17] Depending on the nature of the agreement, this second layer is often highly complex as the layer of legal rules generally excludes the consideration of "off-contract" materials, such as pre-contract negotiations, with limited exception.[18] In a smart contract, the written text codifying the parties' intentions is complemented (or in some instances fully replaced) by machine-executable code.[19] Depending on the design and nature of the agreement, this code layer may cover some to all of the contract as a whole.[20] This model highlights that modern smart contracts remain "wrapped" in a traditional contractual framework given the relationship between the contract's written instruments and its digital ones.[21]

active role to be able to independently contract absent any humans, such issues are beyond the scope of this paper.

[12] *See* Michal S. Gal & Niva Elkin-Koren, *Algorithmic Consumers*, 30 HARV. J. LAW & TECH. 309, 318–19 (2017).

[13] *Id.* at 318.

[14] *See generally* J.G. Allen, *Wrapped and Stacked: "Smart Contracts" and the Interaction of Natural and Formal Language,* 14 EUR. REV. CONT. L. 307 (2018).

[15] Bettina Hein, *0+0=1: The Appliance Model of Selling Software Bundled with Hardware* 17–18 (May 11, 2007) (unpublished M.S. thesis, Massachusetts Institute of Technology) (on file with author).

[16] *Id.*

[17] Allen, *supra* note 14, at 331.

[18] *Id.*

[19] *Id.*

[20] *Id.*

[21] *Id.*

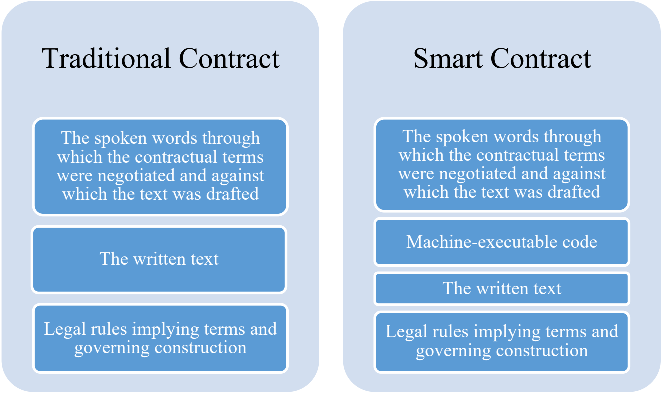
Traditional Contract
The spoken words through which the contractual terms were negotiated and against which the text was drafted
The written text
Legal rules implying terms and governing construction
Smart Contract
The spoken words through which the contractual terms were negotiated and against which the text was drafted
Machine-executable code
The written text
Legal rules implying terms and governing construction

Figure 1. Comparison of the elements of a traditional contract and those of a smart contract under J.G. Allen's contract stack theory.

B. Breakdowns in the Contract Stack

Two applications of smart contracts include high frequency trading, and distributed organizations.

1. High Frequency Trading

One widespread application of machine-executed smart contracts is their use in high frequency trading. While no one universal or legal definition of high frequency trading exists, the term generally refers to the trading of financial instruments, such as securities and derivatives through the use of supercomputers with the capability to execute trades within microseconds or milliseconds.[22] High frequency trading thus works as a smart contract whereby the algorithmic code is used for "decision making, order initiation, generation, routing, or execution, for each individual transaction without human direction."[23] By trading hundreds or thousands of times per day, traders that use high frequency trading are able to profit by aggregating small amounts of profit per trade, often in the magnitude of fractions of a cent.[24] Over the past decade, the use of high frequency trading has grown significantly. Within the U.S., it accounts for approximately 55% of the equity market trading volume and about 40% in European equity markets.[25]

[22] Rena S. Miller & Gary Short, Cong. Research. Serv, R44443, High Frequency Trading: Overview of Recent Developments 1 (2016), https://fas.org/sgp/crs/misc/R44443.pdf.

[23] *Id.*

[24] *Id.* at 2, 4 n.14.

[25] Austin Gerig, *High-Frequency Trading Synchronizes Prices in Financial Markets* 1 (Jan. 21, 2015) (unpublished manuscript) (on file with U.S. Securities and Exchange Commission, Division of Economic and Risk

Algorithmic trading programs generally influence the trading decisions of as many as seventy percent of the securities transactions executed in the United States.[26]

However, high frequency trading is not without its risks, as it can unintentionally contribute to market volatility in times of stress. The most notable example of this is the flash crash of 2010, during which the Dow Jones Industrial Average dropped by nine percent and millions of dollars were lost in the span of minutes.[27] While the official report issued by the joint advisory committee, comprised of staff from both the Commodity Futures Trading Commission (CFTC) and the Securities and Exchange Commission (SEC), investigating the crash did not explicitly attribute the crash to high frequency traders, it did find that the algorithm of a non-high frequency traded mutual fund sold nearly a tenth of its previous volume as a hedge against unusually high volatility in the market.[28] The execution of the algorithmic sell contract resulted in the largest net change in daily position of any trader in the E-Mini S&P 500 futures contracts market since the beginning of the year.[29] While the high frequency traders initially absorbed the downward pressure caused by this sale, they eventually aggressively sold their contracts, increasing the rate at which the mutual fund sold its position.[30] According to a report on the flash crash, "[a]s time passed, the aggressiveness only increased, with these violent selling events occurring more often, until finally the e-Mini circuit breaker kicked in and paused trading for 5 seconds, ending the market slide."[31]

While the flash crash occurred in a regulated securities market, not all high frequency trading occurs in such venues. Dark pools are alternative trading systems that permit the trading of securities listed on national securities exchanges as well as "off-exchange," or unlisted securities without any disclosure of the trading information required on

Analysis), https://www.sec.gov/dera/staff-papers/working-papers/dera-wp-hft-synchronizes.html.

[26] *See* Michael Mackenzie, *High-Frequency Trading Under Scrutiny*, FIN. TIMES (July 28, 2009), http://www.ft.com/intl/cms/s/0/d5fa0660-7b95-11de-9772-00144feabdc0.html#axzz3kPGSbBkJ.

[27] CTFC & SEC, FINDINGS REGARDING THE MARKET EVENTS OF MAY 6, 2010, 1–4 (Sept. 30, 2010), https://www.sec.gov/news/studies/2010/marketevents-report.pdf.

[28] *Id.* at 2.

[29] *Id.*

[30] *May 6'th 2010 Flash Crash Analysis*, NANEX (Oct. 14, 2010), http://www.nanex.net/FlashCrashFinal/FlashCrashAnalysis_Theory.html.

[31] *Id.*

nationally regulated exchanges.[32] As of April 2020, there are over fifty dark pools in operation.[33] While the SEC's Regulation ATS imposed some rules on these dark pools, the regulation lowers the standard of applicable regulation compared to national securities exchange if the alternative trading system complies with the Regulations reporting requirements.[34]

Like with the initial hypothetical with Hunter and Dakota, the performance of the high frequency trading programs highlights how machines, while dutifully following their coding, can lead to disastrous results. While such issues may have never arisen if only one algorithm was in place, the features that make smart contracts used for high frequency trading—their speed and their analytical sophistication—meant that their sensitivity to market forces created economic resonance that affected the market in ways that the traders behind the high frequency trading never intended.

2. Decentralized Organizations

Another application of smart contracts is the use of distributed ledger technology as a centralized platform to create decentralized organizations.[35] One of the most prominent examples of such a decentralized organization is the DAO, a decentralized autonomous organization programmed by the German startup Slock.it.[36] Having raised more than $160 million from more than 10,000 people globally, the DAO operated on the same technology that drives Bitcoin.[37] However, rather than operate as a cryptocurrency, the DAO's purpose was to effectively be a venture capital firm that raises funds for projects run on the Ethereum blockchain—the leading blockchain-based platform for smart contracts—and then to disperse the funds based on

[32] Kristin N. Johnson, *Regulating Innovation: High Frequency Trading in Dark Pools*, 42 J. CORP. L. 833, 864 (2017).

[33] SEC, ALTERNATIVE TRADING SYSTEMS WITH FORM ATS ON FILE WITH THE SEC AS OF APRIL 30, 2020 (Apr. 30, 2020), https://www.sec.gov/files/node/add/data_distribution/atslist043020.pdf.

[34] *See* Regulation of Exchanges and Alternative Trading System, Exchange Act Release No. 34-40760, 63 Fed. Reg. 245 (Dec. 11, 1998) (providing alternative trading systems with a choice in regulatory treatment as an exchange or as a broker-dealer).

[35] *See generally, e.g.*, Laila Metjahic, *Deconstructing the DAO: The Need for Legal Recognition and the Application of Securities Laws to Decentralized Organizations*, 39 CARDOZO L. REV. 1533 (2018).

[36] *See id.* at 1534; David Siegel, *Understanding the DAO Attack*, COINDESK (June 25, 2016), https://www.coindesk.com/understanding-dao-hack-journalists.

[37] Metjahic, *supra* note 35, at 1534.

its members' votes.[38] Ultimately, it contained roughly fifteen percent of all ether on the Ethereum network.[39] Like with corporations that rely on a set of bylaws that guide how shareholders, employees, and the board of the directors interact with one another, the DAO relied on a set of smart contracts that encoded the bylaws of the organization.[40] However, despite its name, the DAO was not a true decentralized autonomous organization as it still depended on human involvement.[41]

Unfortunately, the smart contracts that powered the DAO proved to be its undoing. Shortly after the DAO's funding period closed, one of the DAO's creators announced a vulnerability that was common to all Ethereum smart contracts, as well as a fix that was meant to prevent the vulnerability from being used to drain funds from the DAO.[42] At the time, more than fifty project proposals were waiting to be voted on, but the blog post reiterated that "no DAO funds [were] at risk."[43] However, less than a week later, an attacker used the supposedly-fixed vulnerability in combination with the inherent functionality of the DAO's smart contracts.[44] First, the vulnerability allowed the attacker to send ether to the DAO's smart contract, which triggered the smart contract to add the ether to its balance using an addToBalance function.[45] Next, the hacker withdrew the same amount of currency, which first triggered the smart contract to disperse the

[38] *See* Nathaniel Popper, *A Hacking of More Than $50 Million Dashes Hopes in the World of Virtual Currency*, N.Y. TIMES (June 17, 2016), https://www.nytimes.com/2016/06/18/business/dealbook/hacker-may-have-removed-more-than-50-million-from-experimental-cybercurrency-project.html.

[39] Siegel, *supra* note 36.

[40] *See* Obrea Poindexter & Temidayo O. Odusolu, *Code-Based Fund - the Future of Startup Funding*, LAW360 (Aug. 10, 2016, 12:55 PM), https://www.law360.com/privateequity/articles/826986/code-based-fund-the-future-of-startup-funding.

[41] *See* Metjahic, *supra* note 35, at 1544.

[42] Stephan Tual, *No DAO Funds at Risk Following the Ethereum Smart Contract 'Recursive Call' Bug Discovery*, SLOCK.IT BLOG (June 12, 2016), https://blog.slock.it/no-dao-funds-at-risk-following-the-ethereum-smart-contract-recursive-call-bug-discovery-29f482d348b#.mbfqikiyo.

[43] *See id.*; Siegel, *supra* note 36.

[44] *See* Maria P. Gomez Gelvez, *Explaining the DAO Exploit for Beginners in Solidity*, MEDIUM (Oct. 16, 2016), https://medium.com/@MyPaoG/explaining-the-dao-exploit-for-beginners-in-solidity-80ee84f0d470. For a more detailed discussion of the multi-stage attack, *see generally* Phil Daian, *Analysis of the DAO Exploit*, HACKING DISTRIBUTED (June 18, 2016), https://hackingdistributed.com/2016/06/18/analysis-of-the-dao-exploit/.

[45] Gelvez, *supra* note 44.

ether using the withdrawBalance function before updating the smart contract's balance.[46] This order of operations, combined with the previously-discovered vulnerability, allowed the attacker to repeatedly request to withdraw the ether before the smart contract ever updated its balances.[47]

In response, nearly ninety percent of individuals with voting rights voted to implement a hard fork in the DAO's protocol. This hard fork split the Etherium network into two chains.[48] On one chain, which kept the original Ethereum name, a new smart contract was created that would allow the initial investors to withdraw their initial investments.[49] On the other unforked version, entitled Ethereum Class, the attacker received the funds.[50] The decision to implement a hard fork led to controversy as it required the violation of the immutability principle of the distributed ledger technology underlying the DAO.[51]

Interestingly, the attacker also opined on whether a hard fork was appropriate after the attack, noting that they had "carefully examined the code of The DAO and decided to participate after finding the feature where splitting is rewarded with additional ether. I have made use of this feature and have rightfully claimed 3,641,694 ether, and would like to thank the DAO for this reward."[52] The attacker opposed the term "theft," instead highlighting that their actions made use of explicitly coded features within the smart contract—code which was described as "control[ling] and sett[ing] forth all terms of the DAO Creation."[53] Thus, the attacker claimed that a soft or hard fork would amount to an unlawful seizure of the attacker's "legitimate and rightful ether, claimed legally through the terms of a smart contract" and they reserved "all rights to take any and all legal action against any accomplices of illegitimate theft, freezing, or seizure of my legitimate ether."[54] However, nothing came of the attacker's threats of legal action.

The DAO highlights again the complicated relationship between the terms of the smart contract as implemented in the code

[46] *Id.*
[47] *Id.*
[48] Ben Kaufman, *The DAO Hack — Stolen $50M & The Hard Fork*, MEDIUM: CYPTOCURRENCY HUB (Apr. 21, 2018), https://cryptocurrencyhub.io/the-dao-hack-stolen-50m-the-hard-fork-8719fb5f28be.
[49] *Id.*
[50] *Id.*
[51] *Id.*
[52] *See generally An Open Letter*, PASTEBIN (June 18, 2016), https://pastebin.com/CcGUBgDG.
[53] *Id.*
[54] *See id.*

itself and the expectations of the parties who contributed funds to the project. On one hand, the attack did not require rewriting or changing the terms of the initial DAO smart contract, but merely interacting with the existing code such that it produced the response that the attacker wanted. Conversely, however, the response the attacker wanted—repeatedly withdrawing funds from the DAO's coffers—was not within the realm of possibilities that they had intended to produce. If the attacker had followed through on his threat of legal action, one can only begin to imagine how a court would struggle to detangle these threads especially where, unlike in other smart contracts, the code of the DAO fully replaced any natural language agreements that would offer additional interpretative guidance.

II. THE DISPUTE BETWEEN B2C2 AND QUOINE

The threads of high frequency trading and the DAO converge in the dispute between B2C2 and Quoine. Quoine, a Singapore-incorporated company, operated a currency exchange platform which allowed third parties to trade Bitcoin and Ethereum "for other virtual currencies or for fiat currencies such as Singapore or US dollars."[55] All traders entered into an agreement with the platform owner, Quoine, in order to trade on the platform.[56] Quoine offered three types of trading on its platform: spot trading, where trades are settled instantly; margin trading, where traders can trade using borrowed funds; and futures trading, where the traders agree to sell at a future date at a given price.[57] For traders engaged in margin trading, the trader could source the borrowed funds from Quoine itself or from other users on the platform who offered their funds for peer-to-peer loans, with the assets in the margin trader's accounts serving as collateral.[58] If the collateral in the account fell below a pre-determined percentage of the loan, the platform would automatically make a margin call and would close out all or part of the margin trader's position in an attempt to prevent further loss and ultimately a default.[59] Additionally, Quoine operated as a market maker on its platform, using its "quoter program" to source prices from other exchanges, akin in some ways to high frequency trading.[60] Similarly, B2C2 was an electronic market maker incorporated in England.[61] While B2C2's software normally evaluates

[55] B2C2 Ltd v. Quoine Pte Ltd [2019] SGHC(I) 03 at 10 (Sing.).
[56] *Id.* at 10.
[57] *Id.* at 7.
[58] *Id.*
[59] *Id.* at 8.
[60] *Id.* at 5–6.
[61] B2C2 Ltd [2019] SGHC(I) 03 at 1.

the first twenty market prices and calculates an appropriate price to trade at, the software maintains a fail-safe price of 10 bitcoin to one ether when there is insufficient market data.[62]

A. The April 19th Reversal

On April 13, 2017, Quoine made changes to several login passwords made earlier in the week for security reasons, but did not make necessary changes to its quoter program.[63] As a result, the quoter program was unable to access data from other exchanges and stopped creating new bitcoin/ether orders on the platform.[64] Over the next few days, existing orders on the bitcoin/ether order book were matched with customer orders, leading to the eventual depletion of the platform's order book.[65]

On April 19th, two margin traders were trading in the Ethereum/Bitcoin market using ether borrowed from Quoine.[66] Despite having insufficient bitcoin to maintain their position, the depletion of the bitcoin/ether order book led the platform, which had no separate program to check if a trader had sufficient available assets, to calculate that the margin traders' positions were in a "margin sell-out position," which served to trigger margin calls and the placement of orders to buy ether at the best available price on the platform.[67] Within a half hour, the only remaining orders belonged to B2C2, priced at its fail-safe price of 10 bitcoin to 1 ether, even though this was 250 times the going rate of the previous existing orders.[68]

The next day, Quoine became aware of the trades and "considered the exchange rate to be such a highly abnormal deviation from the previous going rate that the trades should be reversed."[69] As a result, Quoine cancelled B2C2's trades and reversed the debit and credit transactions.[70] Subsequently, B2C2 filed suit against Quoine, asserting that the unilateral cancellation of the trades constituted breach of contract of the agreement required of all traders seeking to trade on the platform.[71] Quoine argued that it was entitled to reverse the trades

[62] *Id.* at 33–34.
[63] *Id.*
[64] *Id.* at 27.
[65] *Id.* at 28.
[66] *Id.* at 12.
[67] B2C2 Ltd [2019] SGHC(I) 03 at 28–29.
[68] *See id.* at 2, 32.
[69] *Id.*
[70] *Id.*
[71] *Id.* at 53.

on the basis that its contracts with B2C2 were void under the doctrine of unilateral mistake at common law.[72]

B. Applying Unilateral Mistake

Under Singaporean law, to render a contract void under the common law doctrine of unilateral mistake, a party must show that it acted while operating under a mistake of a fundamental term of the contract and that the non-mistaken party had actual knowledge of the mistaken party's error.[73] Additionally, the doctrine of unilateral mistake in equity substitutes the non-mistaken party's actual knowledge with constructive knowledge and also requires the mistaken party to show that the non-mistaken party engaged in unconscionable conduct in relation to the mistake in order to establish that the contract is voidable.[74] Because of the lack of any human intervention in the formation and execution of the contracts, the primary issue became whose knowledge was to be assessed as to the mistake, raising the following questions: "What mistakes have been made and to what extent are they fundamental? How does the Court assess knowledge or intention when the operation is carried out by computers acting as programmed? Whose knowledge is relevant? At what date is knowledge to be assessed?"[75]

As a threshold matter, the SICC rejected Quoine's argument that the court should approach these questions as if the parties had met on the "floor of the exchange" when executing the trades at issue.[76] Instead, the SICC noted that the parties chose intentionally to use computers as their means of trading and that in doing so, were aware that no human element was involved.[77] The SICC concluded that the mistake must be a mistake by the person on whose behalf the computer placed the order, and that the mistake must be in existence on or prior to the time of the trades.[78]

Quoine argued that B2C2 had actual or at least constructive knowledge of the margin traders' mistaken belief that they were buying ether for bitcoin at prices which accurately represented or at least did not deviate significantly from the true market value because B2C2's use of the fail-safe price of 10 bitcoin for 1 ether was set to allow it to unconscionably profit from potential errors of the other market

[72] *Id.* at 54.
[73] B2C2 Ltd [2019] SGHC(I) 03 at 77.
[74] *Id.* at 82, 83.
[75] *Id.* at 86.
[76] *Id.*
[77] *Id.* at 86–87.
[78] *Id.*

participants.[79] While the SICC held that price was a fundamental term of the agreement, on appeal, Singapore's highest court disagreed, instead characterizing the mistaken belief as a "mistaken *assumption* on the part of the Counterparties as to how the Platform would operate" because the prices of the disputed trades were determined by the parties' respective algorithms.[80] Thus, while the margin traders may have been mistaken in their assumptions about the circumstances under which the trades would be completed, the fundamental term at issue—price—was not mistaken.[81]

While the SICC and Singapore's highest court disagreed as to whether the mistaken belief went to a fundamental term of the trades, they both agreed that B2C2 did not have actual or constructive knowledge of the margin traders' mistaken belief.[82] In attempting to address whose knowledge was central to the inquiry, the SICC adopted Quoine's argument that it consider what the programmer of the software in question would have known and intended when writing the software.[83] The court, attempting to limit its decision to the facts of the case, noted that while it did not intend to express any views on the legal relationship between computers and those who control or program them, the programs in the present case were entirely deterministic in that they "do and only do what they have been programmed to do," akin to "a kitchen blender relieving a cook of the manual act of mixing ingredients."[84] Thus, the SICC held that when determining the intention or knowledge underlying a machine's mode of operation, the court ought to turn to the operator or controller of the machine.[85] In doing so, it inquired as to whether the programmer had actual knowledge that other traders believed that "in no circumstances would a trade be transacted on the Platform at prices which deviated substantially from the actual market prices . . . ."[86] Finding that the programmer foresaw a number of factors which might cause the program to lack adequate pricing information and thus rely on the fail-safe price, the court determined that his intention of including the fail-safe price was because of his knowledge of the possibility of trades at

[79] B2C2 Ltd [2019] SGHC(I) 03 at 91.

[80] *Cf. id.* at 84 (holding that price was a fundamental term of the agreement), *with* Quoine Pte Ltd. v. B2C2 Ltd., [2020] SGCA(I) 02 at 39 (Sing.) (holding that the mistaken belief at issue was a mistaken assumption as to how the platform would operate).

[81] Quoine Pte Ltd. v. B2C2 Ltd., [2020] SGCA(I) 02 at 56–57 (Sing.).

[82] *Id.* at 60–62.

[83] B2C2 Ltd. v. Quoine Pte Ltd., [2019] SGHC(I) 03 at 89 (Sing.).

[84] *Id.* at 88–89.

[85] *Id.*

[86] *Id.* at 99.

that price being executed.[87] Because the programmer did not have this actual knowledge, the SICC extrapolated and held that he could not have had actual knowledge that other traders on the platform had this belief.[88] Accordingly, the SICC rejected Quoine's unilateral mistake at common law defense.[89]

With regard to its unilateral mistake in equity defense, the SICC held that the programmer did not "turn a blind eye to that which would have been obvious to everyone else in his position," especially given his rationale and motivation for inserting the fail-safe price.[90] Further, the SICC noted that there was no unconscionability in the programmer's actions, and that while they may have been an "opportunistic" business position to ensure that what would otherwise be an unlikely event was not necessarily an unlikely loss, it was not sinister.[91] Thus, the SICC similarly rejected Quoine's unilateral mistake in equity defense.[92]

C. Evaluating the B2C2 Decision

While it is not surprising that what appears to be the first case of the doctrine of mistake in the context of a smart contracts would emerge from Singapore, a jurisdiction with a reputation for embracing new technologies,[93] the question remains: did Singapore reach the right decision? While the court reached the correct result, it incorrectly limited its inquiry to the knowledge and intentions of the programmer, the overall outcome is consistent with American contract law.

The SICC's tailoring of its inquiry to solely focus on the knowledge and motivations of the programmer behind B2C2's price algorithm is concerning in its exclusion of the knowledge and motivation of B2C2 itself.[94] In reaching the decision to inquire into the programmer's state of mind, the SICC concedes that "[t]he knowledge or intention cannot be that of the person who turns it on, it must be that of the person who was responsible for causing it to work in the way it

[87] *Id.*

[88] *Id.* at 99.

[89] Quoine Pte Ltd., [2019] SGHC(I) 03 at 101.

[90] *Id.* at 100.

[91] *Id.* at 101.

[92] *Id.*

[93] *See* Lee U-Wen, *Singapore a 'Global Leader' in Embracing Technology: Deloitte*, SGSME (May 17, 2017, 5:50 AM), https://www.sgsme.sg/news/singapore-global-leader-embracing-technology-deloitte.

[94] The issue here is slightly muddled as the programmer is also a B2C2 director.

did."[95] However, in applying this principle, the SICC incorrectly limits itself only to the programmer. Read narrowly as the Singapore courts did, this limitation does not necessarily comply with the reality of the relationship between the programmer and the contracting party and where it does, it near entirely forecloses the possibility of successfully applying the doctrine of unilateral mistake.

First, this narrow reading assumes the relationship between the programmer and the contracting party is entirely separate, when the development of the algorithm is a byproduct of communication between both parties. An issue with the formal language instrument within a smart contract "stack" is the challenge in ensuring consistency between the formal language used by a programmer within the instrument and the natural language understandings of what that formal language is meant to accomplish.[96] In the context of programmers acting on behalf of a separate party, these natural language understandings are not those of the programmers themselves, but rather the separate party on whose behalf they are creating the program.[97] This relationship mirrors the relationship between conventional transaction lawyers and the parties on whose behalf they draft traditional human language contract instruments. Similarly limiting the inquiry to the knowledge and intentions of the lawyer who drafted instruments rather than inquiring into the intentions of the contracting client would produce absurd results. In both circumstances, the approach appears to frustrate the court's pursuit of determining and enforcing the parties' intent by excluding a party from the inquiry. Thus, while a deterministic program may be limited to "only do what the programmer has programmed it to do,"[98] the relevant inquiry must also include what the programmer was told by the contracting party to do.

Further, even if no relationship exists between the contracting party and the programming, the sole consideration of the programmer's knowledge and motivations would foreclose near any possibility of successfully applying the doctrine of unilateral mistake. Such a relationship may arise in instances where an "off the rack" smart

[95] Quoine Pte Ltd., [2019] SGHC(I) 03 at 89.

[96] *See* Allen, *supra* note 14, at 336 ("Essentially, the challenge is to ensure that the terms written as down in the formal language by a programmer are isomorphic with the natural language understandings on which it is based. This is difficult because of the way that natural and formal languages work. It is not possible at the present time, in my view, and may never be fully possible. Even should we design systems which generate a natural and a formal language version simultaneously, it is impossible to say that their semantic content will be identical in all possible future states of the contract.").

[97] *See id.* at 329.

[98] Quoine Pte Ltd., [2019] SGHC(I) 03 at 88.

contract solution is developed from consumer use. In such instances, the programmer may never have any sense of another party's mistaken belief, let alone actual knowledge of them. Mistaken parties, however, may be able to find some relief in the form of unilateral mistake in equity. Under that approach, a court could theoretically find that a remote programmer had constructive knowledge that their program would be used to capitalize on a generalized party's mistaken belief, and if the non-mistaken party's conduct was unconscionable, the contract is voidable in equity. However, such an approach is narrower than the traditional alternative under common law and may not produce consistent results as it might under law.[99] Further, it does not address the effective elimination of the common law defense in such situations. Instead, even where no prior relationship exists between the programmer and the non-mistaken party, the courts should inquire into the contracting party's state of knowledge. Such an approach would allow the doctrine of unilateral mistake to attach in situations where a program operated as it was programmed to but at the hands of a contracting party who sought to use it to take advantage of the other party's mistaken belief.

While Singapore's highest court noted that the contracting parties at issue and Quoine did not know that the specific trading contracts would be entered into or what specific terms they would contain,[100] the court takes for granted that contracting parties manifested their willingness to contract when they entered into agreements with Quoine to trade on the platform. More specifically, the court ignores the parties' involvement in the choice of using the specific algorithm. In the present case, the proprietary nature of the algorithms collapses the inquiry because the programmer's knowledge mirrored that of B2C2 generally. However, had B2C2 used a generally available algorithm rather than a bespoke one, a court limiting itself only to the consideration of the programmer's knowledge would omit any consideration of why B2C2 chose that program over another, even if that choice was to use it to take advantage of another party's mistaken belief.

[99] While unilateral mistake in common law requires actual knowledge, unilateral mistake in equity requires both constructive knowledge and "an additional element of impropriety" based on "the presence of other facts which could invoke the conscience of the court." *Id.* at 83. Thus, while equity's "flexibility to achieve the ends of justice" may serve as an advantage, the use of these more flexible standards could affect the consistency and predictability of the applicability of this doctrine depending on the judges hearing the case. *See id* at 82.

[100] Quoine Pte Ltd. v. B2C2 Ltd., [2020] SGCA(I) 02 at 46 (Sing.).

III. Potential Improvements Through a Threshold Inquiry

The inclusion of a threshold inquiry mirroring § 154 of the Restatement (Second) on Contracts may improve the application of the doctrine of unilateral mistake. Section 153 of the Restatement lays out the requirements for the American doctrine of unilateral mistake:

> Where a mistake of one party at the time a contract was made as to a basic assumption on which he made the contract has a material effect on the agreed exchange of performances that is adverse to him, the contract is voidable by him if he does not bear the risk of the mistake under the rule stated in § 154, and
>
> (a) the effect of the mistake is such that enforcement of the contract would be unconscionable, or
>
> (b) the other party had reason to know of the mistake or his fault caused the mistake.[101]

This doctrine mostly mirrors the Singaporean approach (which in turn mirrors the approach under English contract law) in its requirement that the mistaken party's belief is a fundamental component of the contract.[102] Notably, however, the American approach limits the doctrine to situations where the mistaken party has not assumed the risk of the mistake. Section 154 clarifies when these situations exist:

> A party bears the risk of a mistake when
>
> (a) the risk is allocated to him by agreement of the parties, or
>
> (b) he is aware, at the time the contract is made, that he has only limited knowledge with respect to the facts to which the mistake relates but

[101] Restatement (Second) of Contracts § 153 (Am. Law Inst. 1981).

[102] *See* B2C2 Ltd v. Quoine Pte Ltd [2019] SGHC(I) 03 at 83 (Sing.).

> treats his limited knowledge as sufficient, or
>
> (c) the risk is allocated to him by the court on the ground that it is reasonable in the circumstances to do so.[103]

Comment A reiterates the general commitment to freedom of contract underlying contract law and notes that the limitation serves to maintain contracting parties' risk allocation, even when a change in circumstance "upset basic assumptions and unexpectedly affect the agreed exchange of performances."[104]

This threshold inquiry, however, requires some tailoring for the context of machine-executed smart contracts. First, when assessing allocation by agreement, courts should either not rely solely on the digital instrument within the contract "stack" as the explicit assertion of the agreement of the parties or should not rely on this prong altogether. If they were to do otherwise, a non-mistaken party would be able to submit the digital instrument's execution as evidence of a non-ambiguous agreement that allocates the risk to the mistaken party, as was the case with the individual who was behind the 2016 DAO attack.[105] In the case of Hunter and Dakota, out of fear that the formal language used to encode their instructions may be used against them, such an approach may encourage parties like Dakota to spend their digital instruments from the specter of any undesirable outcome, however improbable. In the worst case, parties may avoid using smart contracts altogether unless they can accomplish the nigh-impossible task of accounting for every possible outcome. Thus, while this prong may be easiest to administer, its adoption may be an overcorrection.

Instead, the threshold inquiry should focus on the second and third prongs. As to the second prong, focusing on scenarios in which the mistaken party has consciously ignored their limited knowledge allows the court to separate cases where the deviation between a party's intentions and a smart contract's execution warrants consideration under the doctrine of unilateral mistake from those that do not. In the case of B2C2 and Quoine, such a prong would clearly highlight that Quoine assumed the risk of such trades.[106]

The Singapore International Commercial Court's ruling identified, among others, five "errors or omissions" that would have

[103] Restatement (Second) of Contracts § 154 (Am. Law Inst. 1981).
[104] *Id.* cmt. a.
[105] *Cf. An Open Letter, supra* note 52.
[106] *See* Quoine Pte Ltd., [2019] SGHC(I) 03.

limited Quoine's liability and would have demonstrated that it did not intend to allocate the risk to itself.[107] First, it failed "to incorporate an exception message in its quoter program to alert Quoine to the fact that it was not working."[108] Second, it failed "to incorporate a circuit breaker in the Platform's software to prevent trades when the order book was empty."[109] Third, it failed "to incorporate a circuit breaker to prevent orders at an abnormal price from being placed on the order book."[110] Fourth, it failed "to ensure that, in the case of a force-closure, the forced sales were only within a given price range."[111] And last, it failed "to ensure that, in the case of a force-closure, only assets which were actually held by a counterparty in its account at the time were the subject of a market order."[112] The cumulative omission of these various mechanisms that would otherwise have limited Quoine's liability could thus support the claim that in consciously failing to include them, Quoine assumed the risk that the algorithm underlying its smart contract would behave outside of its expectations.

For similar reasons as with the first prong, courts should not implement this prong so as to require, explicitly or implicitly, that parties wishing to use smart contracts to implement mechanisms to address every possible outcome, lest they be accused of consciously ignoring it. Instead, however, it could be used to import a reasonable standard of precaution based on prevailing risk-limitation practices, such as redundant sources of information and circuit breakers. Such an interpretation would highlight for parties that the choice to forego any mechanisms to cabin in the behavior of their smart contracts are liable for their contracts' execution, even when they deviate from the parties' expectations. In other words, parties would be on notice that if they plan to use smart contracts to "move fast," they're responsible for the things they "break" along the way. While this interpretation would not resolve all issues of risk allocation, it would however encourage parties to focus on ensuring that the risk mitigation mechanisms they impose are appropriately tailored.

Last, the third prong would provide courts with a mechanism by which to assign the risk to a party when the other factors are not illustrative. Comment D to § 154 provides that when using this prong, the court "will consider the purposes of the parties and will have recourse to its own general knowledge of human behavior in bargain

[107] *See id.*

[108] *Id.*

[109] *Id.*

[110] *Id.*

[111] *Id.* at 90.

[112] Quoine Pte Ltd., [2019] SGHC(I) 03 at 90.

transactions, as it will in the analogous situation in which it is asked to supply a term . . . ."[113] In this way, the third prong allows courts to allocate the risk to a party where it is reasonable to do so for reasons outside of the first two prongs.[114] For example, where a smart contract's terms are silent as to which party bears the risk, and neither party consciously ignored the risks within the agreement, the court can then apply its collective experience with traditional contract to the general circumstances of the case. While this approach does not delineate specific factors the court should look at in applying this prong, applying this prong only as a last resort limits the risk of uncertain application while affording courts flexibility where they need it.

CONCLUSION

Despite the centrality of freedom of contract to modern contract law, courts have grappled with the task of when to curb this principle and instead allow a party to claim that a mistake occurred. As smart contracts proliferate, however, so too does the risk that discrepancies will arise between what a party intends and the result produced by the machine-executable code underlying the contract. In turn, as these parties claim the doctrine of mistake, courts must assess where these discrepancies fit within traditional contract law. In the first decision of its kind, the Singapore International Commercial Court and the Court of Appeal of Singapore addressed this issue directly, and while they reached the correct result, their self-imposed limitation of considering only the knowledge and intentions of the programmer behind the smart contracts goes beyond what is necessary. Further, a threshold inquiry, offered by American contract law, may help to more easily disambiguate circumstances in which a party using a smart contract has assumed the risks associated with its execution, and when a party has truly made mistakes with machines.

[113] Restatement (Second) of Contracts § 154 cmt. d (Am. Law Inst. 1981).
[114] *Id.*