



Fall 2021

Creating Cryptolaw for the Uniform Commercial Code

Carla L. Reyes

Southern Methodist University Dedman School of Law, carla.reyes@smu.edu

Follow this and additional works at: <https://scholarlycommons.law.wlu.edu/wlulr>



Part of the [Commercial Law Commons](#), [Contracts Commons](#), and the [Secured Transactions Commons](#)

Recommended Citation

Carla L. Reyes, *Creating Cryptolaw for the Uniform Commercial Code*, 78 Wash. & Lee L. Rev. 1521 (2021).

Available at: <https://scholarlycommons.law.wlu.edu/wlulr/vol78/iss4/7>

This Article is brought to you for free and open access by the Washington and Lee Law Review at Washington and Lee University School of Law Scholarly Commons. It has been accepted for inclusion in Washington and Lee Law Review by an authorized editor of Washington and Lee University School of Law Scholarly Commons. For more information, please contact christensena@wlu.edu.

Creating Cryptolaw for the Uniform Commercial Code

Carla L. Reyes*

Abstract

A contract generally only binds its parties. Security agreements, which create a security interest in specific personal property, stand out as a glaring exception to this rule. Under certain conditions, security interests not only bind the creditor and debtor, but also third-party creditors seeking to lend against the same collateral. To receive this extraordinary benefit, creditors must put the world on notice, usually by filing a financing statement with the state in which the debtor is located. Unfortunately, the Uniform Commercial Code (U.C.C.) Article 9 filing system fails to provide actual notice to interested parties and introduces risk of heavy financial losses.

To solve this problem, this Article introduces a smart-contract-based U.C.C.-1 form built using Lexon, an innovative new programming language that enables the

* Assistant Professor of Law, Southern Methodist University Dedman School of Law; Affiliated Faculty, Indiana University Ostrom Workshop on Cybersecurity and Internet Governance; Research Associate, University College London Blockchain Research Centre. I am deeply grateful to Henning Diedrich, founder of, and creative genius behind, Lexon, for his comments and input in the development of the prototype. I am also extremely grateful to Paul Hodnefield and Darrell W. Pierce for their excellent and detailed feedback on the prototype and its instructions over several iterations as part of the American College of Commercial Finance Lawyers mentor program. Additional thanks to the organizers and participants of the following conferences for extremely helpful feedback: the 2020 Association of American Law Schools Annual Meeting Commercial Law Section, the 2021 Stanford University CodeX Roundtable, ComplianceNet 2021, and the Chicago Kent Law's 2019 Block (Legal) Tech Conference. I am also extremely grateful to Sara Steves for invaluable research assistance.

development of smart contracts in English. The proposed “Lexon U.C.C. Financing Statement” does much more than merely replicate the financing statement in digital form; it also performs several U.C.C. rules so that, for the first time, the filing system works as intended. In demonstrating that such a system remains compatible with existing law, the Lexon U.C.C. Financing Statement also reveals important lessons about the interaction of technology and commercial law.

This Article brings cryptolaw to the U.C.C. in three parts. Part I examines the failure of the U.C.C. Article 9 filing system to achieve actual notice and argues that blockchain technology and smart contracts can help the system function as intended. Part II introduces the Lexon U.C.C. Financing Statement, demonstrating how the computer code implements U.C.C. provisions. Part II also examines the goals that influenced the design of the Lexon U.C.C. Financing Statement, discusses the new programming language used to build it, and argues that the prototype could be used now, under existing law. Part III proposes five innovations for the Article 9 filing system enabled by the Lexon U.C.C. Financing Statement. Part III then considers the broader implications of the project for commercial law, legal research around smart contracts, and the interplay between technology-neutral law and a lawyer’s increasingly important duty of technological competence. Ultimately, by providing the computer code needed to build the Lexon U.C.C. Financing Statement, this Article demonstrates not only that crypto-legal structures are possible, but that they can simplify the law and make it more accessible.

Table of Contents

INTRODUCTION	1524
I. A SMART U.C.C.-1 FINANCING STATEMENT CAN HELP THE ARTICLE 9 FILING SYSTEM PROVIDE ACTUAL NOTICE.....	1530
A. <i>The Article 9 Filing System is a Notice System that Fails to Actually Give Notice</i>	1532
B. <i>Blockchain Technology Gives Notice to Large, Dispersed Groups of People</i>	1537

C.	<i>Smart Contracts Can Perform Many U.C.C. Rules Related to Financing Statements</i>	1541
II.	A PROTOTYPE SMART-CONTRACT-BASED FINANCING STATEMENT EVERYONE UNDERSTANDS.....	1544
A.	<i>The Prototype Uses Lexon to Enable Transparency for a Public System</i>	1545
B.	<i>The Prototype Creates a Smart U.C.C.-1 Form, but also Fills the Gaps Between How the Filing System Was Intended to Work and How It Actually Works</i>	1549
1.	Traditional U.C.C.-1 Form Data Fields ...	1551
2.	Recitals and Clauses to Formalize the Logic of the Article 9 Filing System	1555
3.	Lapse, Continuation, and Clearing	1559
C.	<i>The Prototype Could Be Used Under Existing Law</i>	1564
III.	KEY LESSONS FROM ENGINEERING A U.C.C. FINANCING STATEMENT IN LEXON.....	1566
A.	<i>The U.C.C. Could Be Improved Beyond Creating New Rules to Accommodate Digital Assets</i>	1566
1.	New Search Capability Without New Indexing Obligations	1567
2.	Identify Specific Collateral Without Introducing Ambiguity.....	1568
3.	Generating New Fees.....	1570
4.	Potential to Escrow Digital Assets Within the Filing System.....	1572
5.	Curbing Filing Abuses by Offering Debtors the Opportunity to Electronically Authorize Financing Statements	1574
B.	<i>The Lexon U.C.C. Financing Statement as a Development in Computational Law and a Call for Lawyers to Expand Their View of the Duty of Technological Competence</i>	1576
	CONCLUSION.....	1582
	APPENDIX A: LEXON U.C.C. FINANCING STATEMENT USING LEX 0.2	1585
	APPENDIX B: SOLIDITY COMPILED FROM LEXON U.C.C. FINANCING STATEMENT	1587

APPENDIX C: JAVASCRIPT COMPILED FROM LEXON U.C.C. FINANCING STATEMENT	1592
APPENDIX D: LEXON U.C.C. FINANCING STATEMENT USING LEXON 0.3.....	1607

INTRODUCTION

Consider the following hypothetical, illustrating how two well-intentioned lawyers can lose a client \$1.5 billion. Imagine you are a very junior associate at BigLaw Firm, LLP (BigLaw). One of your supervising partners pulls you into ongoing maintenance matters on a deal BigLaw concluded for SuperBank, N.A. (SuperBank) about seven years ago. The partner explains that SuperBank, together with a syndicate of other lenders, lent \$300 million dollars to Big Four Automaker (Automaker).¹ To ensure repayment of the outstanding balance, SuperBank and its co-lenders took twelve pieces of Automaker's property as collateral. Automaker informs SuperBank that it intends to complete repayment of the loan. Your supervising partner asks you to prepare the documents necessary for SuperBank and the lenders to be repaid and to release the interests the lenders held in Automaker's property. You are familiar with the required closing checklist and draft documents for repayment but ask the supervising partner for additional detail regarding what might be needed to release the security interest in the collateral. The supervising partner tells you to prepare something called a "termination statement" for each security interest held by the lenders. "But," you ask, "how do I know what security interests need to be terminated?" "By conducting a U.C.C.-1 search," instructs the partner.

You conduct some initial research and determine that a termination statement is a form filed in the Uniform Commercial Code (U.C.C.) Article 9 filing system maintained by the state where the debtor is located.² You still remain unsure how to actually conduct a search in the U.C.C. filing system, but

1. This introductory story is based off of *In re Motors Liquidation Co.*, 777 F.3d 100 (2d Cir. 2015).

2. See U.C.C. §§ 9-501(a), 9-301, 9-521 U.C.C.-3 form (AM. L. INST. & UNIF. L. COMM'N 2010) (outlining the process for filing and the perfection of a security interest).

you understand from your research that it can be quite complicated.³ As a result, you do what very smart junior associates always do when they need assistance from colleagues with more practice experience—you ask a paralegal with experience in U.C.C.-1 filings for help in conducting the search. Unfortunately, although the paralegal enjoys a great deal of experience with U.C.C.-1 filings in general, the paralegal has no familiarity with the SuperBank and Automaker transaction. Thus, when the paralegal conducts the U.C.C.-1 search in Delaware's filing system and finds three financing statements documenting security interests in Automaker's property, you prepare termination statements for all three. Neither you nor the paralegal helping you realize that only two of the three financing statements relate to the \$300 million loan that is about to be fully repaid. The third financing statement perfects a security interest in collateral securing a separate \$1.5 billion loan to Automaker made by a different syndicate of lenders.

As good associates do, you present your draft documents to the supervising partner for correction and input. The supervising partner does not notice your error. As good partners in transactional deals do, the partner sends the prepared documents to Automaker's outside counsel for review, comment, and negotiation if needed. Automaker's outside counsel also fails to catch the error. The parties execute all of the documents and file all three termination statements. A year later, when Automaker files for bankruptcy, the bankruptcy court, and later the Second Circuit, determines that the erroneously filed termination statement released the collateral from the security interest, making the syndicated lending group and SuperBank unsecured creditors to the tune of \$1.5 billion. The result? The inherent difficulty you encountered in using the U.C.C. Article 9

3. See Lynn M. LoPucki, *Computerization of the Article 9 Filing System: Thoughts on Building the Electronic Highway*, 55 L. & CONTEMP. PROBS. 5, 6 (1992) [hereinafter LoPucki, *Computerization of the Article 9 Filing System*]

As the article 9 filing system is currently conceived and implemented, (1) it is impractical for a secured creditor to do everything necessary to make and maintain an effective filing, (2) many kinds of filings are effective even though they are, as a practical matter, impossible for searchers to discover, and (3) the processes for both filing and searching are unreasonably complex and error-prone.

filing system and the related error you made cost your client around one billion dollars.

This story, familiar to any student of secured transactions using a traditional casebook containing the actual case,⁴ *In re Motors Liquidation Co.*,⁵ represents a cautionary tale in the significant financial consequences that may stem from misunderstanding the U.C.C. Article 9 filing system. The problem, however, is that the Article 9 filing system does not work the way it was intended, making it easy to fall victim to one of its many traps, including the one that plagued the associate in the SuperBank and Automaker deal. The goal of the U.C.C. Article 9 filing system is to give prospective creditors knowledge of preexisting liens.⁶ Unfortunately, the current system routinely fails to provide actual notice.⁷ Although several scholars suggest reforming the filing system using technology,⁸ no one has reduced these theoretical applications to practice, until now.

This Article argues that blockchain technology and smart contracts can form the basis of an Article 9 filing system that remedies many of the current system's failures and proves it by unveiling the code for a prototype. Why turn to blockchain technology and smart contracts? Blockchain technology is

4. See, e.g., CANDACE M. ZIERDT ET AL., *ESSENTIAL UCC CONCEPTS: A SURVEY OF COMMERCIAL TRANSACTIONS* 642–47 (2018); STEVEN L. HARRIS & CHARLES W. MOONEY JR., *SECURITY INTERESTS IN PERSONAL PROPERTY: CASES, PROBLEMS, AND MATERIALS* 190–96 (6th ed. 2019); LYNN M. LOPUCKI ET AL., *SECURED TRANSACTIONS: A SYSTEMS APPROACH* 379–82 (8th ed. 2016).

5. 777 F.3d 100 (2d Cir. 2015).

6. See LoPucki, *Computerization of the Article 9 Filing System*, *supra* note 3, at 5 (“An important purpose of the system is to communicate the existence of filed financing statements to those who search the records. If a filing was made, a searcher should be able to find it.”).

7. See *id.* at 11–15 (detailing the failure of the Article 9 filing system that makes actual notice difficult or impossible).

8. See, e.g., *id.* at 15–19 (proposing the use of “computer technology” and distributed networks to enable nationwide searches); Christopher G. Bradley, *Disrupting Secured Transactions*, 56 HOUS. L. REV. 965, 966 (2019) (proposing a “regime [that] would allow creditors to stake their claims directly—by means of online ‘smart’ maps or by electronic tags identifying interests in particular items of collateral—and would eliminate numerous arcane, inefficient, and inequitable features of the current [Article 9 filing system] regime”); Carla L. Reyes, *Conceptualizing Cryptolaw*, 96 NEB. L. REV. 384, 402–03, 417–20 (2017) [hereinafter Reyes, *Conceptualizing Cryptolaw*] (proposing the smart-contract-based Article 9 filing system actually built in this Article).

particularly effective at enabling parties that do not necessarily know or trust each other to agree that they have access to the same record of evolving facts.⁹ This is precisely what the Article 9 filing system struggles to achieve.¹⁰ Further, to account for the inherent problems of the existing system, U.C.C. Article 9 contains a variety of rules to, whenever possible, mitigate errors such as the one at issue in the SuperBank and Automaker transaction.¹¹ Smart contracts can help implement or eliminate the need for many of those rules, reducing the complexity of the U.C.C. and reducing the risk of errors.¹²

Most scholarly discussion at the intersection of secured transactions, blockchain technology, and smart contracts focuses on how to treat digital assets like bitcoin when they are taken as collateral for secured transactions governed by U.C.C.

9. Richard Gendal Brown, *Introducing R3 Corda: A Distributed Ledger Designed for Financial Services*, THOUGHTS ON THE FUTURE OF FIN. (Apr. 5, 2016), <https://perma.cc/Y5ME-KV5M> (describing the key attribute of blockchain technology as “the emergence of platforms, shared across the Internet between mutually distrusting actors, that allow them to reach consensus about the existence and evolution of facts shared between them”); RICHARD GENDAL BROWN ET AL., *CORDA: AN INTRODUCTION* 4 (2016), <https://perma.cc/5E73-NE3T> (PDF) (explaining that blockchain technology enables “progression from a world where parties to shared facts record and manage their own records” to one “where they collaborate to maintain a shared record, assured to be consistent between them”).

10. See Reyes, *Conceptualizing Cryptolaw*, *supra* note 8, at 403 (“[P]arties using the [Article 9] system can never be certain that the facts that they are looking at are the same facts that the other parties in the system see. In other words, the system regularly fails to reach a consensus about the existence and evolution of shared facts.”); Bradley, *supra* note 8, at 979 (“[T]he system is riddled with loopholes, gaps, and exceptions. As a result, creditors remain unsure of how secure their interest in collateral really is.”); LoPucki, *Computerization of the Article 9 Filing System*, *supra* note 3, at 15 (“[T]he probative value of a search depends ultimately on off-record information about the collateral and the debtor, and that in many instances the debtor is the primary source of that information.”).

11. See LoPucki, *Computerization of the Article 9 Filing System*, *supra* note 3, at 15 (“[T]he system performs a quite different function than that envisioned in theory. Its rules are not instructions as to what one must do to obtain and maintain priority. So read, they are completely impractical. Instead, they have become rules of loss allocation.”).

12. See Reyes, *Conceptualizing Cryptolaw*, *supra* note 8, at 418–20 (providing examples of and explaining the ways smart contracts could theoretically fix the problems posed by the current Article 9 system).

Article 9.¹³ Indeed, the Uniform Law Commission and American Law Institute took up these questions and are currently working to create rules for digital asset collateral that better track the purpose, design, and policy goals of Article 9.¹⁴ This Article generally does not engage those discussions; rather, this Article picks up another strand of scholarly inquiry—the potential for and proper design of a technology-based reform of the U.C.C. Article 9 filing system¹⁵—and ties it to blockchain technology and smart contracts by reducing theory to practice.

13. See, e.g., Kevin V. Tu, *Perfecting Bitcoin*, 52 GA. L. REV. 505, 517 (2018) [hereinafter Tu, *Perfecting Bitcoin*] (arguing that virtual currency such as bitcoin needs to be defined as a separate collateral type under Article 9); Xavier Focroulle Menard, *Cryptocurrency: Collateral for Secured Transactions?*, 34 BANKING & FIN. L. REV. 347, 382–85 (2020) (proposing a legislative framework and definitions to govern security interests in cryptocurrency); James P. Nehf, Security Interests in Virtual Currencies 9 (Mar. 2, 2020) (unpublished manuscript), <https://perma.cc/G4GN-UAW9> (summarizing key definitions under Article 9 and their application to virtual currency); Kristin Johnson et al., *(Im)Perfect Regulation: Virtual Currency and Other Digital Assets as Collateral*, 21 SMU SCI. & TECH. L. REV. 115, 116 (2018) (addressing the “tensions that arise when new asset classes [such as bitcoin] challenge static regulatory approaches”); Ronald J. Mann, *Reliable Perfection of Security Interests in Crypto-Currency*, 21 SMU SCI. & TECH. L. REV. 159, 160–65 (2018) (explaining bitcoin’s categorization and how to perfect it under Article 8 and Article 9); Xuan-Thao Nguyen, *Lessons from Case Study of Secured Transactions with Bitcoin*, 21 SMU SCI. & TECH. L. REV. 181, 183 (2018) (arguing that, while imperfect, “the Article 9 system is adequate to accommodate cryptocurrencies” and thus “we should study the lending market with cryptocurrencies as collateral and observe how law and technology have been utilized in fostering the development in the market”); Kevin V. Tu, *Crypto-Collateral*, 21 SMU SCI. & TECH. L. REV. 205, 205 (2018) (examining whether Article 9 should be modified “to better accommodate crypto-collateral”); Jeanne L. Schroeder, *Bitcoin and the Uniform Commercial Code*, 24 U. MIAMI BUS. L. REV. 1, 9–10 (2016) (discussing the characterization of bitcoin under the U.C.C.); Andrew Balthazor, *The Bona Fide Acquisition Rule Applied to Cryptocurrency*, 3 GEO. L. TECH. REV. 402, 423 (2019) (arguing that “if bitcoin is ever to be widely adopted . . . as collateral for debts . . . there must be some mechanism for purchasers to perform a title inquiry”).

14. See *Uniform Commercial Code and Emerging Technologies Committee*, UNIF. L. COMM’N, <https://perma.cc/5DMF-JRZ5> (collecting materials from the Committee’s recent meetings). The Author is an observer on the Committee, and part of the Digital Asset Working Group that met weekly for months during 2020 and 2021 to discuss and propose adjustments to commercial law for digital assets.

15. See, e.g., John J. Eikenburg, Jr., Comment, *Filing Provisions of Revised Article 9*, 53 SMU L. REV. 1627, 1637–40 (2000) (suggesting the use of electronic data interchange); Steven L. Harris & Charles W. Mooney, *Negotiability, Electronic Commercial Practices, and a New Structure for the*

Specifically, this Article introduces a smart-contract-based U.C.C.-1 financing statement prototype (the “Lexon U.C.C. Financing Statement”) built using an innovative new programming language—Lexon. Lexon is a programming language that enables the development of smart contracts in English. This Article provides the program code needed to execute the Lexon U.C.C. Financing Statement in Lexon,¹⁶ Solidity, and JavaScript, the latter two being produced from the first by the Lexon compiler.¹⁷ The Lexon U.C.C. Financing Statement accomplishes more than just replicating the financing statement in digital form; it also performs several U.C.C. provisions so that, for the first time, the filing system works as intended. In demonstrating that such a system remains compatible with existing law, the Lexon U.C.C. Financing Statement also reveals important lessons about the interaction of technology and commercial law.

This Article brings the previously theoretical possibility of crypto-legal structures¹⁸ to life in three parts. Part I examines

UCC Article 9 Filing System: Tapping the Private Market for Information Technology, 31 IDAHO L. REV. 835, 848 (1995) (proposing that the Article 9 filing system should be revised to “take advantage of competition and private market forces in the market for information technology”); LoPucki, *Computerization of the Article 9 Filing System*, *supra* note 3; Bradley, *supra* note 8; Reyes, *Conceptualizing Cryptolaw*, *supra* note 8; Bryan G. Bosta, Comment, *Bringing Article 9 Up to Speed: The Need for a National Filing System*, 31 U. DAYTON L. REV. 25 (2005).

16. “Lexon is a computer language that anyone can read. It’s the first of a new generation of languages. It was made for blockchain smart contracts and can be used to write normal contracts that are blockchain smart contracts.” HENNING DIEDRICH, LEXON BIBLE: HITCHHIKER’S GUIDE TO DIGITAL CONTRACTS vi (2020) [hereinafter, DIEDRICH, LEXON BIBLE].

17. Solidity is the most frequently used programming language for smart contracts on the Ethereum blockchain protocol. Specifically, Solidity is “[a] procedural (imperative) programming language with a syntax similar to JavaScript, C++, or Java.” ANDREAS M. ANTONOPOULOS & GAVIN WOOD, MASTERING ETHEREUM: BUILDING SMART CONTRACTS AND DAPPS 131 (2018). The Lexon and Solidity versions of the Lexon U.C.C. Financing Statement would operate on the Ethereum blockchain, while the JavaScript version would not require a public blockchain to operate.

18. A crypto-legal structure is when “the law of any subject matter [is] implemented and delivered through smart-contracting, semi-autonomous cryptographic computer code.” Reyes, *Conceptualizing Cryptolaw*, *supra* note 8, at 387. I previously argued that the prototype system I develop in this Article was theoretically possible. See *generally id.* Professor Matthew Bruckner asked via Twitter whether I could actually build it. I am not a

the failure of the U.C.C. Article 9 filing system to achieve actual notice and argues that blockchain technology and smart contracts can help the system function as intended. Part II introduces the Lexon U.C.C. Financing Statement, considers the goals that influenced its design, discusses the new programming language used to build it, and argues that the prototype could be used now, under existing law. Part III proposes five innovations to the Article 9 system enabled by the Lexon U.C.C. Financing Statement. Part III then considers the broader implications of the project for commercial law, legal research around smart contracts, and the interplay between technology-neutral law and a lawyer's increasingly important duty of technological competence. Ultimately, this Article demonstrates not only that crypto-legal structures are possible, but they can simplify the law and make it more accessible.

I. A SMART U.C.C.-1 FINANCING STATEMENT CAN HELP THE ARTICLE 9 FILING SYSTEM PROVIDE ACTUAL NOTICE

SuperBank and Automaker do not stand alone as a cautionary tale of Article 9 filing system complexity. If anything could make the junior associate and paralegal at the center of that tale feel better, it would be the story of the bankruptcy trustee who failed to discover financing statements filed against

trained computer programmer, so I could not immediately build the system that I proposed. However, around the time that I was writing *Conceptualizing Cryptolaw*, Henning Diedrich was writing the first Lexon white paper. Having built Lexon into a useable programming language, Henning Diedrich enabled me, a non-programmer, to write the computer program that proves out my earlier theory. Beyond waiting for Lexon to develop, creating the Lexon U.C.C. Financing Statement also required decisions about how to achieve the many policy goals I wanted to bake into the system. Nevertheless, this Article presents the prototype Professor Bruckner requested, proving out not only the possibility of such a system, but also many of the theoretical ripple effects that I predicted in 2017. *Id.* at 414–37. As discussed *infra* in Parts II.A. and III.C., this Article goes beyond my predictions in 2017 by connecting cryptolaw to the broader discipline of computational law. As demonstrated in Appendix C, the Lexon code can compile to JavaScript, such that it can run on regular servers in addition to blockchain technology. If the Lexon U.C.C. Financing Statement was used in an off-chain environment, it would still rely on smart contracts in the sense intended by Nick Szabo, *see infra* note 90 and accompanying text, but would not need Ethereum to do so.

the debtor.¹⁹ Apparently, the secured creditor filed the financing statement under the debtor's trade name, sparking a dispute as to the effectiveness of the filing that wound its way to the Fifth Circuit Court of Appeals and continues to be a source of confusion in the filing system.²⁰ In all honesty, the junior associate and paralegal from the SuperBank and Automaker debacle probably do not feel better; but the point remains that the Article 9 filing system presents many traps for the unwary, even bankruptcy trustees and other repeat sophisticated players in the secured transactions marketplace.²¹ This Part details the failings of the Article 9 filing system that so often trap the unsuspecting. In particular, this Part shines a light on the many ways the Article 9 filing system fails to actually give notice to the dispersed and often unknown parties operating in the secured lending system.²² This Part then introduces a

19. See Jeffrey W. Morris, *The Fruits of Mischievous Seeds: Notice Filing Under Article 9 and the Continuing Problem of Trade Names*, 11 U. DAYTON L. REV. 241, 241–42 (1986) (detailing the facts of *Glasco, Inc. v. Citizens Bank*, 642 F.2d 793, 795 (5th Cir. 1981)).

20. See *id.* (explaining that the Fifth Circuit reversed the bankruptcy court's and district court's findings in *Glasco, Inc.*).

21. See, e.g., Eikenburg, *supra* note 15, at 1631–33 (identifying some of those problems as filing search delays, risk of human error, and expenses).

22. Scholars have voiced other concerns about the Article 9 filing system; however, blockchain technology cannot cure all ills. My goal in designing the Lexon U.C.C. Financing Statement is to resolve the many issues related to ensuring the Article 9 system provides actual notice—for which blockchain technology is actually quite useful. For a discussion of some of the other concerns with the Article 9 filing system, see Eric M. Sherman, Note, *Chasing Perfection: Collateral Indications and Ambiguous Debtor Names on Financing Statements Under Article 9*, 61 B.C. L. REV. 2229, 2258–60 (2020) (considering difficulties arising from requiring parties to conduct due diligence beyond the four corners of the financing statement); Lynn M. LoPucki, *The Spearing Tool Filing System Disaster*, 68 OHIO ST. L.J. 281, 283–84 (2007) [hereinafter LoPucki, *Spearing Tool Disaster*] (discussing problems with debtor names in filings); Meghan M. Sercombe, Note, *Good Technology and Bad Law: How Computerization Threatens Notice Filing Under Revised Article 9*, 84 TEX. L. REV. 1065, 1066–67 (2006) (arguing that computer search logic exacerbates the name problems in Article 9 filings); David R. Beran, *Financing Statements, Descriptions, Collateral and Confusion: Arkansas Courts Tackle the New Article 9*, 57 ARK. L. REV. 951, 952–53 (2005) (discussing confusion in Arkansas law regarding the difference between collateral descriptions in financing statements and security agreements); Juliet M. Moringiello, *Revised Article 9, Liens from the Fringe, and Why Sometimes Signatures Don't Matter*, 10 WIDENER J. PUB. L. 135, 135–36 (2001) (discussing the practice of filing financing statements to harass public officials).

technology designed to provide notice to large dispersed groups of people through a transparent and difficult-to-change record of transactions: blockchain technology.²³ Finally, this Part argues that using a smart-contract-based filing system would cause ripple effects through a variety of provisions in U.C.C. Article 9.

A. *The Article 9 Filing System is a Notice System that Fails to Actually Give Notice*

Article 9 of the U.C.C. governs security interests in personal property.²⁴ When a lender contracts (via a security agreement²⁵) with a debtor²⁶ for a property interest in the debtor's personal property contingent upon the nonpayment of a debt (a security interest),²⁷ the lender becomes a secured party.²⁸ The personal

23. PRIMAVERA DE FILIPPI & AARON WRIGHT, BLOCKCHAIN AND THE LAW: THE RULE OF CODE 2 (2018) (“[Blockchains] blend together several existing technologies, including peer-to-peer networks, public-private key cryptography, and consensus mechanisms, to create what can be thought of as a highly resilient and tamper-resistant database where people can store data in a transparent and nonrepudiable manner.”); ARVIND NARAYANAN ET AL., BITCOIN AND CRYPTOCURRENCY TECHNOLOGIES: A COMPREHENSIVE INTRODUCTION 51 (2016) (“[T]he Bitcoin consensus mechanism gives us an append-only ledger, a data structure that we can only write to. Once data is written to it, it’s there forever.”); ANTONOPOULOS & WOOD, *supra* note 17, at 1 (“From a computer science perspective, Ethereum is a deterministic but practically unbounded state machine, consisting of a globally accessible singleton state and a virtual machine that applies changes to that state. It uses a blockchain to synchronize and store the system’s state changes.”).

24. U.C.C. § 9-109(a)(1) (AM. L. INST. & UNIF. L. COMM’N 2010) (“Except as otherwise provided in subsections (c) and (d), this article applies to: (1) a transaction, regardless of its form, that creates a security interest in personal property”); *see* Bradley, *supra* note 8, at 965. All fifty states, Puerto Rico, and the District of Columbia have adopted a version of the U.C.C. Bradley, *supra* note 8, at 967, 967 n.4.

25. *See* U.C.C. § 9-102(a)(73)(A) (“‘Secured party’ means: (A) a person in whose favor a security interest is created or provided for under a security agreement, whether or not any obligation to be secured is outstanding. . . .”).

26. *See id.* § 9-102(a)(28) (“‘Debtor’ means: (A) a person having an interest, other than a security interest or other lien, in the collateral, whether or not the person is an obligor; (B) a seller of accounts, chattel paper, payment intangibles, or promissory notes; or (C) a consignee.”).

27. *See id.* § 1-201(b)(35) (“‘Security interest’ means an interest in personal property or fixtures which secures payment or performance of an obligation.”).

28. *See supra* note 25.

property attached²⁹ to the security interest (the collateral)³⁰ helps the secured lender mitigate risk of default³¹ on the loan by providing some incentive for repayment and supplying known value from which to recuperate (at least some of) the outstanding balance of the loan.³² If the debtor fails to pay, the creditor may take possession of the collateral,³³ and liquidate it³⁴ to satisfy the outstanding balance of the loan.³⁵ Indeed, in most circumstances, the secured creditor can engage in this process of repossession and sale without turning to the judicial system for enforcement.³⁶

But a debtor can offer the same personal property as collateral to multiple lenders.³⁷ The ability of a secured creditor to protect its interest in any given collateral against other creditors³⁸ depends on the ability of secured creditors to

29. Note that to say a security interest has attached to particular personal property means that the debtor and creditor have met certain requirements that make the security interest valid and enforceable as between the debtor and creditor. U.C.C. § 9-203(a)–(b) (AM. L. INST. & UNIF. L. COMM'N 2010) (requirements of attachment).

30. *See id.* § 9-102(a)(12) (“‘Collateral’ means the property subject to a security interest or agricultural lien.”).

31. *See* ZIERDT ET AL., *supra* note 4, at 475 (“[A]ny time a borrower gives a lender an interest in a specific asset (collateral) to provide additional assurance (security) that the borrower will pay the loan, the borrower and lender have created a secured transaction.”).

32. *See* U.C.C. § 9-615 (sale proceeds applied to outstanding balance of debt).

33. *See id.* § 9-609 (creditor can repossess collateral so long as there is no breach of the peace).

34. *See id.* § 9-610 (creditor right to sell collateral upon satisfying certain basic requirements).

35. *See id.* § 9-615 (sale proceeds applied to outstanding balance of debt).

36. *See* ZIERDT ET AL., *supra* note 4, at 580–81 (“Article 9 is unique in American law in that it permits self-help. Consider what this means: a secured party can, after default, seize (or more likely, hire someone to seize) the collateral without going to court first.”).

37. *See* Eikenburg, *supra* note 15, at 1628 (“The debtor wishes to gain as much credit as possible while risking as little collateral as necessary. The creditor wants to ensure that the loan will be repaid. Additionally, the creditor wants to be certain that he has priority over other creditors on any debt against the debtor’s collateral.” (citations omitted)).

38. *See* ROBERT E. SCOTT & GEORGE G. TRIANTIS, FOUNDATIONS OF COMMERCIAL LAW 268 (2006) (“Assuming a security interest is enforceable against the debtor, a secured creditor enjoys priority over the collateral if it

communicate the existence of their security interests to each other.³⁹ This is the role of what is known in secured transactions as perfection.⁴⁰ If a secured creditor takes the steps required by Article 9 to perfect its security interest, the security interest will not only be valid and enforceable against the debtor, but it will also be sufficient to fix the secured creditor's place in line for access to collateral over other creditors should the debtor default.⁴¹ Although U.C.C. Article 9 offers a variety of methods for perfection depending on the classification of the collateral, each of them seek to replicate the functional equivalent of putting the world on notice that a security interest in that collateral may exist.⁴² Being able to enforce a contractual right—here a security interest—against third parties with no prior knowledge of the contract is an extraordinary right, and to justify it, Article 9 requires secured creditors to publicly declare the existence of the security interest.⁴³ Publicly declaring the existence of security interests in personal property also enables later-in-time lenders to conduct necessary due diligence in

perfects the security interest by taking one of the prescribed steps to publicize it.”).

39. See *id.* (“The priority of a secured creditor hinges on the time at which she publicizes her lien.”); LoPucki, *Spearing Tool Disaster*, *supra* note 22, at 283 (“The function of the Article 9 filing system is to provide notice of prior security interests to those who consider taking subsequent ones (hereafter ‘searchers’).”).

40. See U.C.C. § 9-308(a) (AM. L. INST. & UNIF. L. COMM’N 2010) (detailing the requirements for perfection).

41. Fixing a creditor’s place in line over others is referred to as “priority.” LOPUCKI ET AL., *supra* note 4, at 439 (“To say that one creditor has priority over another is to say that if the value of the collateral is sufficient to pay only one of them, the law requires that value be used to pay the one who has priority.”).

42. See, e.g., *id.* at 324 (“Both Article 9 and real estate recording statutes recognize possession of some kinds of collateral as a substitute for public notice filing.”); *id.* at 329 (“Article 9 recognizes ‘control’ of some kinds of collateral as a substitute for filing.”).

43. See *id.* at 280 (“To ensure that the prospective lender can discover a lien that will have priority over its own, the laws under which liens are created almost invariably condition the priority on the holder taking steps to make the lien public and easily discoverable.”). Indeed, the policy prohibiting secret non-possessory liens in the United States can be traced back to at least 1819. SCOTT & TRIANTIS, *supra* note 38, at 306–07.

advance of a contemplated deal.⁴⁴ The primary way secured creditors communicate the existence of their security interest to anyone else considering becoming a creditor to the same debtor is through participation in the U.C.C. Article 9 filing system.⁴⁵

The goal of the U.C.C. Article 9 filing system is to give prospective creditors notice that preexisting liens on a debtor's assets may exist.⁴⁶ Although the formal explanation of the Article 9 filing system focuses on the provision of constructive notice, "it is intended to do more than that. In theory, at least, it is supposed to give actual notice to the later creditor."⁴⁷ Unfortunately, the current system routinely fails to provide actual notice.⁴⁸ As designed, the Article 9 filing system remains imprecise, difficult, and expensive to use because of at least the five following factors:⁴⁹ (1) there is not just one filing system, but many;⁵⁰ (2) "search methods differ widely from one filing system

44. See SCOTT & TRIANTIS, *supra* note 38, at 318 ("The searcher must obtain further information from the debtor and the secured party.").

45. See LOPUCKI ET AL., *supra* note 4, at 281 (describing the Article 9 filing system as a mechanism for "communicating the existence of a lien from the holder to a person who is considering becoming a creditor of the same debtor").

46. See SCOTT & TRIANTIS, *supra* note 38, at 318 ("A distinctive feature of the Article 9 registry is that a financing statement puts a searcher on notice that the debtor may have granted a security interest in some or all of the identified assets to the identified creditor."); Robert I. Donnellan, *Notice and Filing Under Article 9*, 29 MO. L. REV. 517, 517 (1964) ("The main purpose of any filing system for security transactions is to give notice to others."); Eikenburg, *supra* note 15, at 1630 ("The most common way to perfect is by filing a financing statement in a public office.").

47. LOPUCKI ET AL., *supra* note 4, at 281.

48. See Eikenburg, *supra* note 15, at 1631 ("When first drafted, the filing system may have been adequate, but currently many problems exist with the system."); Bosta, *supra* note 15, at 25 ("Companies in the business of lending must be able to find existing transactions between a potential debtor and its creditors efficiently, accurately, and as cost-effectively as possible to protect their legal rights. The current filing system under Revised Article 9 of the [U.C.C.] does not facilitate these goals.").

49. I recognize that this is only an introductory list of the failings of the Article 9 filing system. A full list is beyond the scope of this Article and has been considered extensively by the literature and cases on Article 9. See, e.g., LoPucki, *Computerization of the Article 9 Filing System*, *supra* note 3, at 6–15.

50. See LOPUCKI ET AL., *supra* note 4, at 283 (explaining that most U.S. counties and states maintain separate filing systems).

to another;⁵¹ (3) searches can only be conducted on the index of the filing, which is limited to an index of the debtor's name;⁵² (4) changes to an individual debtor's name, restructuring of a corporate debtor, or use of a corporate trade name can cause complexity for a filing system based on name searches;⁵³ and (5) complying with the intricate rules of lapse, continuation, and termination to ensure maintenance of a security interest until full repayment of the debt remains complicated.⁵⁴ In other words, when a prospective lender searches the Article 9 filing system, they can never be sure that the results obtained reflect the actual state of the filing records.⁵⁵ The actors in the secured transactions marketplace cannot be certain that they have the same record of existing and evolving facts as all of the other parties in the ecosystem.⁵⁶ Fortunately, an emerging technology specifically designed to help diffuse actors reach consensus about the existence and evolution of shared facts—namely, blockchain technology⁵⁷—offers an opportunity to improve the

51. *Id.* at 294; see Sercombe, *supra* note 22, at 1068–69 (stating that financing statements can be accessed by a variety of search programs, and that states vary, at least some, in computer programs and search logic used).

52. See LOPUCKI ET AL., *supra* note 4, at 298–99 (calling “[r]eliance on debtor’s names” the “Achilles’ heel of the Article 9 filing system”); Eikenburg, *supra* note 15, at 1631–32 (explaining that the potential for incorrect indexing is one of the problems with the Article 9 filing system).

53. See LoPucki, *Computerization of the Article 9 Filing System*, *supra* note 3, at 23 (explaining how simple errors in filing and searching can create challenges in finding corporate debtors); Eikenburg, *supra* note 15, at 1632 (explaining the propensity for human error in lender filing and searching).

54. See LoPucki, *Computerization of the Article 9 Filing System*, *supra* note 3, at 23–24.

55. See *id.* at 15 (“[T]he probative value of a search depends ultimately on off-record information about the collateral and the debtor, and that in many instances, the debtor is the primary source of that information.”); Bradley, *supra* note 8, at 979 (“[T]he promise of certainty is not fulfilled because Article 9’s current system of debtor-based identification is cumbersome and ineffective. It includes various ornate provisions for maintaining existing interests even where they aren’t really identified and can’t put another lender on notice.”).

56. Reyes, *Conceptualizing Cryptolaw*, *supra* note 8, at 403.

57. See GARRICK HILEMAN & MICHEL RAUCHS, GLOBAL BLOCKCHAIN BENCHMARKING STUDY 13 (2017), <https://perma.cc/R4K4-REKN> (PDF) (“[P]articipants in a blockchain network reach consensus about changes to the state of the shared database (i.e., transactions amongst participants) without needing to trust the integrity of any of the network participants or administrators.”).

efficiency, accuracy, and usefulness of the Article 9 filing system.

B. *Blockchain Technology Gives Notice to Large, Dispersed Groups of People*

Blockchain technology is often described as one type of distributed database known broadly as distributed ledger technology (DLT).⁵⁸ Researchers describe a distributed ledger as a “type of distributed database that assumes the possible presence of malicious users (nodes).”⁵⁹ Although commonly used interchangeably with DLT, the term blockchain more precisely refers to a sub-set of DLT protocols that structure their data in a literal “chain of blocks” by linking blocks of validated transactions together using one-way cryptographic hashes.⁶⁰ The combination and implementation of specific features, such

58. *See id.* at 11 (defining blockchain as a “type of distributed ledger”).

59. *Id.* As I have explained before, I am aware of the ongoing debate as to appropriate terminology, and, in particular, the discussion around the terms blockchain technology versus distributed ledger technology. Reyes, *Conceptualizing Cryptolaw*, *supra* note 8, at 389–90. Without intending to weigh in on the substance of that debate, I use the term “distributed ledger technology” as the broader, umbrella term to encompass both permissioned and permissionless blockchains, as well as protocols such as R3’s Corda that do not strictly fit the definition of a “chain of blocks.” HILEMAN & RAUCHS, *supra* note 57, at 11. Meanwhile, I use the term “blockchain technology” to refer specifically to those distributed ledgers that use data structures composed of a cryptographically linked chain of blocked data. *Id.* Adopting these terms is not a statement about the technical accuracy of this or any other terminology. I use these terms, consistently with other researchers such as Hileman and Rauchs, as a legal academic, grounded in the premise that all of these protocols exist and are in use, and that any legal and policy discussion of such systems should account for the full range of implementations, or explain why the analysis only matters for a specific implementation. For further insight into my position, see *id.* *See also* Tim Swanson, *A Brief History of R3—the Distributed Ledger Group*, GREAT WALL OF NOS. (Feb. 27, 2017) <https://perma.cc/B548-EM25> (differentiating between blockchains and R3); WILLIAM MOUGAYAR, *THE BUSINESS BLOCKCHAIN: PROMISE, PRACTICE, AND APPLICATIONS OF THE NEXT INTERNET TECHNOLOGY* 7 (2016) (“Since the Internet is comprised of a public version and several private variations, blockchains will also follow that path. Therefore, we will have public and private blockchains.”).

60. *See* HILEMAN & RAUCHS, *supra* note 57, at 11 (defining blockchain as a “type of distributed ledger that is composed of a chain of cryptographically linked ‘blocks’ containing batched transactions; generally broadcasts all data to all participants in the network”).

as the type of consensus mechanism used to verify transactions, vary by implementation among various DLT and blockchain protocols.⁶¹ Generally speaking, however, blockchain technology is a protocol technology.⁶² A protocol is “a set of instructions for the compilation and interaction of objects.”⁶³ Generally, a “network protocol” simply sets the rules that allow networked computers (nodes) to communicate with each other.⁶⁴ A blockchain protocol, for its part, sets the rules that enable networked computers to track transitions in the global state of recorded data without a centralized third party intermediary.

The blockchain protocol that receives the most attention is the Bitcoin blockchain, in part because it is the first of its kind.⁶⁵

61. There are, for example, any number of different ways to achieve consensus. Ethereum currently uses proof-of-work, but Ethereum is moving to proof-of-stake consensus. Alyssa Hertig, *Ethereum's Big Switch: The New Roadmap to Proof-of-Stake*, COINDESK (May 5, 2017, 8:00 AM), <https://perma.cc/8PHJ-TEJD> (last updated May 16, 2017, 11:27 AM). Ripple and Stellar use “a *unique node list* of at least one hundred nodes they can trust in voting on the state of affairs.” DON TAPSCOTT & ALEX TAPSCOTT, BLOCKCHAIN REVOLUTION 32 (2016). There are other mechanisms as well, including proof of activity, proof of capacity, and proof of storage. *Id.* DLT protocols may also vary in what activity must be cryptographically signed. DTCC, SECURITY OF DLT NETWORKS: A DISTRIBUTED LEDGER TECHNOLOGY SECURITY FRAMEWORK FOR THE FINANCIAL SERVICES INDUSTRY 4 (2020), <https://perma.cc/MR3D-98CY> (PDF). As alluded to above, the Bitcoin and Ethereum blockchains require transactions to be cryptographically signed, while in the Ethereum 2 protocol, computations, and results are also cryptographically signed. Other variations abound.

62. Carla L. Reyes, *(Un)Corporate Crypto-Governance*, 88 FORDHAM L. REV. 1875, 1895 (2020) [hereinafter Reyes, *(Un)Corporate Crypto-Governance*].

63. ALEXANDER R. GALLOWAY, PROTOCOL: HOW CONTROL EXISTS AFTER DECENTRALIZATION 75 (2004).

64. Will Warren, *The Difference Between App Coins and Protocol Tokens*, MEDIUM: OX BLOG (Feb. 2, 2017), <https://perma.cc/GQC7-FS3E>. For example, the Internet Protocol is a network protocol that defines the digital message formats and rules for communication among connected computers. *Internet Protocol (IP)*, TECHOPEDIA, <https://perma.cc/VDD3-TLSY> (last updated Aug. 20, 2019). Email is also built on a protocol that allows users to communicate with one another; “[i]t’s just a way for two computers to talk to one another.” Ryan Shea, *When to Use Protocol Tokens*, MEDIUM (Nov. 13, 2017), <https://perma.cc/MXE7-W7Q6>.

65. See NARAYANAN ET AL., *supra* note 23, at XXII (“Bitcoin combines the idea of using computational puzzles to regulate the creation of new currency units with the idea of secure timestamping to record a ledger of transactions and prevent double spending. There were earlier, less sophisticated, proposals that combined these two ideas.”).

The value of the Bitcoin blockchain's native cryptocurrency,⁶⁶ bitcoin,⁶⁷ regularly skyrockets.⁶⁸ However, the core function of the Bitcoin blockchain remains tracking transitions in state.⁶⁹ Specifically, using transactions as the core building block of the protocol,⁷⁰ the Bitcoin blockchain tracks the use of unspent transaction outputs, or UTXOs.⁷¹ The UTXOs are locked by a script, a small computer program, that says "this can be redeemed by a public key that hashes to X, along with a signature from the owner of that public key."⁷² More complex programs, however, can be built on top of the Bitcoin blockchain only by going off-chain, by implementing them separately from the chain, interacting with it rather than being part of it.⁷³

As a result, new protocols have emerged. The Ethereum protocol, for example, enables the execution of complex computer programs via a blockchain protocol that is essentially

66. See Warren, *supra* note 64 (explaining that native cryptocurrency, or protocol tokens, "provide the financial incentives needed to drive a cryptoeconomic protocol which may or may not be implemented within an Ethereum smart contract"); HENNING DIEDRICH, *ETHEREUM* 43 (2016) [hereinafter DIEDRICH, *ETHEREUM*] ("Ether is the native token of Ethereum, its 'bitcoins' This is the official definition of Ether: it's the currency in which to pay the fee to be allowed to run your calculations, make your transactions, and store your data on the [Ethereum] blockchain." (emphasis omitted)).

67. Generally speaking, the Bitcoin software, protocol and network are referenced using the uppercase "Bitcoin" while the lowercase "bitcoin" refers to individual units of cryptocurrency. Angela Walch, *The Bitcoin Blockchain as Financial Market Infrastructure: A Consideration of Operational Risk*, 18 N.Y.U. J. LEGIS. & PUB. POL'Y 837, 846 n.41 (2015).

68. See, e.g., Daniel Cawrey, *Market Wrap: Bitcoin Hits Record \$48.2K as CME Ether Futures at \$33M Volume on First Day*, COINDESK (Feb. 9, 2021, 4:52 PM), <https://perma.cc/Z3GT-98UA> (describing the increase in Bitcoin value several days in a row).

69. See NARAYANAN ET AL., *supra* note 23, at 52 (describing how Bitcoin tracks transactions).

70. See *id.* at 51 ("Let's start with transactions, Bitcoin's fundamental building block.").

71. See *id.* at 52

Bitcoin doesn't use an account-based model. Instead, Bitcoin uses a ledger that just keeps track of transactions Transactions specify a number of inputs and a number of outputs You can think of the inputs as coins being consumed (created in a previous transaction) and the outputs as coins being created.

72. *Id.* at 55.

73. See *id.* at 58–59 (describing how certain scripts function).

a global decentralized computer.⁷⁴ To accomplish this, the Ethereum protocol offers a general purpose blockchain protocol, upon which developers can build a variety of applications.⁷⁵ Developers most commonly build programs to run on the Ethereum protocol in the programming language Solidity.⁷⁶ Another example of an increasingly popular blockchain protocol is Aeternity.⁷⁷ Aeternity also offers a platform for building complex smart contracts, but seeks to improve upon Ethereum by addressing several perceived scalability and usability factors.⁷⁸ Specifically, Aeternity seeks to offer increased transaction throughput, “to assign human readable, persistent names to objects on chain,” and to enable easier use of oracles for smart contract execution.⁷⁹ Developers build programs to run on Aeternity in the programming language Sophia.⁸⁰ The common theme among these protocols, despite their technical differences, is that, at base, blockchain protocols, and most DLT protocols, share a core attribute: they track transitions in state in order to allow participants in the network to reach agreement about the existence and evolution of shared facts.⁸¹

74. See ANTONOPOULOS & WOOD, *supra* note 17, at 2 (“Unlike Bitcoin, which has a very limited scripting language, Ethereum is designed to be a general-purpose programmable blockchain that runs a virtual machine capable of executing code of arbitrary and unbounded complexity.”); DIEDRICH, ETHEREUM, *supra* note 66, at 28 (“Where Bitcoin is intentionally dumbed down in its capabilities, the scripts run by Ethereum are stateful and Turing-complete. This simply means that Ethereum ‘contracts’ have memory that they will remember the next time they are called and that they can have loops.” (emphases omitted)).

75. See ANTONOPOULOS & WOOD, *supra* note 17, at 4 (“Ethereum is designed to be a general-purpose programmable blockchain”); DIEDRICH, ETHEREUM, *supra* note 66, at 30 (“Ethereum is a platform for decentralized applications, smart contracts and decentralized, autonomous organizations.”).

76. See ANTONOPOULOS & WOOD, *supra* note 17, at xxxiv (describing Solidity as “[t]he most popular and most frequently used language for Ethereum smart contracts”).

77. See generally THOMAS ARTS ET AL., AETERNITY: OPEN SOURCE BLOCKCHAIN FOR SCALABLE AND SECURE SMART CONTRACTS (2020), <https://perma.cc/9FDP-N9X8> (PDF).

78. See *id.* at 1.

79. *Id.* at 3.

80. See *id.* at 16 (“Sophia is a functional programming language. The main unit of code in Sophia is the contract.”).

81. See Peter Van Valkenburgh, *What’s a Blockchain, Anyway?*, COIN CTR. (Apr. 25, 2017), <https://perma.cc/P576-2HJP> (describing how bitcoin

C. *Smart Contracts Can Perform Many U.C.C. Rules Related to Financing Statements*

Recall that, as a protocol technology, computer programs can be built on top of, or incorporated into, blockchain technology.⁸² A smart contract is one type of computer program frequently used in connection with blockchain technology.⁸³ Like the variance among implementations of DLTs and blockchain protocols, the precise implementation of a smart contract can vary significantly. At base, however, a smart contract is very similar to a “persistent script”—a standing computer program—that says “if event x happens, then execute result y.”⁸⁴ Smart contracts are, however, quite passive.⁸⁵ Smart contracts cannot reach out to find data evidencing an event, “x,” has

works); Brown, *supra* note 9 (stating that DLT are “platforms, shared across the Internet between mutually distrusting actors, that allow them to reach consensus about the existence and evolution of facts shared between them”).

82. Kevin Werbach & Nicolas Cornell, *Contracts Ex Machina*, 67 DUKE L.J. 313, 333 (2017).

83. See, e.g., DIEDRICH, ETHEREUM, *supra* note 66, at 176 (“In Ethereum, ‘smart contract’ often just means ‘a Solidity script.’”); ANTONOPOULOUS & WOOD, *supra* note 17, at 127 (“In the context of Ethereum, the term . . . ‘smart contracts’ . . . refer[s] to immutable computer programs that run deterministically in the context of an Ethereum Virtual Machine as part of the Ethereum network protocol—i.e., on the decentralized Ethereum world computer.”).

84. See Carla L. Reyes, *A Unified Theory of Code Connected Contracts*, 46 J. CORP. L. 981, 987 (2021) [hereinafter Reyes, *A Unified Theory*] (“[A] smart contract is computer software that causes something to happen upon the fulfillment of pre-determined conditions.”). Vitalik Buterin defines smart contracts as “systems which automatically move digital assets according to arbitrary pre-specified rules.” VITALIK BUTERIN, ETHEREUM WHITE PAPER: A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM 1 (2013), <https://perma.cc/2PMK-JAT4> (PDF). Others define smart contract as “a computerized transaction protocol to execute contract terms.” ALEX NORTA, SETUP OF CROSS-ORGANIZATIONAL COLLABORATIONS FOR DECENTRALIZED AUTONOMOUS ORGANIZATIONS 1 (2015), <https://perma.cc/CBV9-9LTV> (PDF); see Richard Gendal Brown, *A Simple Model for Smart Contracts*, THOUGHTS ON THE FUTURE OF FIN. (Feb. 10, 2015), <https://perma.cc/VP5C-ZCEK> (“A smart contract is an event-driven program, with state, which runs on a replicated, shared ledger and which can take custody over assets on that ledger.”).

85. ANTONOPOULOUS & WOOD, *supra* note 18, at 128–29 (“All smart contracts in Ethereum are executed, ultimately, because a transaction initiated from an EOA. A contract can call another contract that can call another contracts and so on, but the first contract in such a chain of execution will always have been called by a transaction for an EOA.”).

occurred.⁸⁶ Rather, the smart contract must be triggered, i.e., sent a signal that an event, “x,” has occurred.⁸⁷ The signal that triggers execution of the smart contract, certifying that “x” has occurred, can be internal to the blockchain (i.e., coming from other smart contracts), or the smart contract can receive the signal, and the data specific to it, from an outside source.⁸⁸ This concept of smart contracts expands on the vision of smart contracts introduced in 1994 by Nick Szabo.⁸⁹

For Szabo, goals for using smart contracts include “to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries.”⁹⁰ Szabo’s idea was more advanced than the technologies of 1994 could accommodate.⁹¹ But when blockchain technology emerged, so did platforms for implementing smart contracts.⁹² What becomes clear from even this brief study of the nature of smart contracts is that the word contract is not used in the legal sense of legally enforceable contract. Rather, smart contracts encompass a far greater range of computer programs running on blockchain technology.⁹³

86. *Id.* at 129 (“Contracts never run ‘on their own’ or ‘in the background.’”).

87. *Id.* (“Contracts effectively lie dormant until a transaction triggers execution, either directly or indirectly as part of a chain or contract calls.”); see HILEMAN & RAUCHS, *supra* note 57, at 60 (“[S]mart contracts need to be triggered by specific events in order to execute.”).

88. DIEDRICH, ETHEREUM, *supra* note 66, at 167–73. When smart contracts receive data from outside sources, those outside sources are often referred to as “oracles.” See MOUGAYAR, *supra* note 59, at 43 (“Oracles are data sources that send actionable information to smart contracts.”).

89. See S. ASHARAF & S. ADARSH, DECENTRALIZED COMPUTING USING BLOCKCHAIN TECHNOLOGIES AND SMART CONTRACTS: EMERGING RESEARCH AND OPPORTUNITIES 45 (2017) (“The concept of smart contracts was first formally coined by Nick Szabo in 1994.”).

90. Nick Szabo, *Smart Contracts* (1994), <https://perma.cc/M93B-9UWM>.

91. See TAPSCOTT & TAPSCOTT, *supra* note 61, at 102 (“Back then, smart contracts were an idea all dressed up with nowhere to go, as no available technology could deploy them as Szabo described.”).

92. See MOUGAYAR, *supra* note 59, at 41.

93. See *id.* at 48; DIEDRICH, LEXON BIBLE, *supra* note 16, at 20; DIEDRICH, ETHEREUM, *supra* note 66, at 168–74.

Although many scholars debate whether smart contracts can be legally enforceable contracts,⁹⁴ or investigate whether all smart contracts might represent one kind of transaction or another,⁹⁵ smart contracts most often operate as tools used to perform obligations.⁹⁶ The obligations may not even stem from contracts. Indeed, many projects seek to use smart contracts to perform legal obligations imposed by statute and regulation.⁹⁷ These regulatory technology projects, or “RegTech,” projects leverage the power of a variety of technologies, including blockchain and smart contracts, to improve efficiencies while also decreasing compliance risk.⁹⁸ But what if an application combined the efficiencies of regulatory technology with the

94. See, e.g., Werbach & Cornell, *supra* note 82, at 339 (“In a very real way, smart contracts are not intended to be legally enforceable.”); Jeremy M. Sklaroff, Comment, *Smart Contracts and the Cost of Inflexibility*, 166 U. PA. L. REV. 263, 286–91 (2017) (discussing the consequences of smart contracts on legal transactions); Karen E.C. Levy, *Book-Smart, Not Street-Smart: Blockchain-Based Smart Contracts and the Social Workings of Law*, 3 ENGAGING SCI. TECH. & SOC’Y 1, 4 (2017) (“Smart contract technology . . . depends on a thin conceptualization of what law does, and how it does it.”); Jeffrey M. Lipshaw, *The Persistence of “Dumb” Contracts*, 2 STAN. J. BLOCKCHAIN L. & POL’Y 1, 27–29 (2019) (arguing that “dumb contracts will persist” because humans are unlikely to reach the amount of consensus necessary to exclusively make smart contracts); Lauren Henry Scholz, *Algorithmic Contracts*, 20 STAN. TECH. L. REV. 128, 166 (2017) (“The enforceability of algorithmic contracts must turn on general private law rules.”).

95. See, e.g., Heather Hughes, *Blockchain and the Future of Secured Transactions Law*, 3 STAN. J. BLOCKCHAIN L. & POL’Y 21, 26 (2020) (“Blockchain-based smart contracts can conflate property and contract functions, making distinctions among private law categories—property, contract, entity law—potentially obsolete.”).

96. See Reyes, *A Unified Theory*, *supra* note 84, at 983.

97. See, e.g., *Securitize*, SECURITIZE INC. (2021), <https://perma.cc/5RDH-3B78> (using smart contracts to automate compliance with securities regulations); Jonathan Keane, *Fresh Off Funding, R3 Moves Corda Toward Critical Pilot*, COINDESK (May 30, 2017, 5:00 AM), <https://perma.cc/8FZW-WPFY> (last updated May 31, 2017) (explaining how blockchain finance solutions “will reduce fraud, increase transparency and enhance the shift away from paper records”); *R3 Unlocks Regulatory Reporting on Corda with Financial Conduct Authority and Two Global Banks*, R3 (Sept. 12, 2017), <https://perma.cc/PUW7-N7SZ> (describing blockchain networks that “can enable continuous regulatory reporting for financial institutions”).

98. See *Financial Conduct Authority, REGTECH*, <https://perma.cc/PQH2-BHUL> (last updated Jan. 12, 2021).

potential power of computational law? A smart-contract-based U.C.C.-1 financing statement can do both.

II. A PROTOTYPE SMART-CONTRACT-BASED FINANCING STATEMENT EVERYONE UNDERSTANDS

This Part unveils a prototype of the Lexon U.C.C. Financing Statement—a smart-contract-based financing statement—and thereby proves that cryptolaw for the U.C.C. filing system is possible.⁹⁹ Several core goals guided the design and build of the Lexon U.C.C. Financing Statement. The first key goal centers on ensuring that the same transparency provided by the current form—a one-page PDF¹⁰⁰—would be retained in the smart contract version. This Part begins by explaining how this project sought to achieve that goal by using Lexon, a new, sixth generation programming language, to code the smart-financing statement in English.¹⁰¹ In using Lexon to build the Lexon U.C.C. Financing Statement, this Part squarely roots the approach in the broader field of computational law.¹⁰² The second core goal of the project involves demonstrating that the

99. In other words, a core component of the scholarly contribution offered in this Article is a software program. “Software is scholarship to the extent that software functionality is derived from scholarly research, software is used as a means to develop scholarship, or software is used as a medium to communicate scholarly ideas.” Houman B. Shadab, *Software is Scholarship*, MIT COMPUTATIONAL L. REP., Nov. 20, 2020, at 2. Here, the software program—the Lexon U.C.C. Financing Statement—both proves out prior scholarly research and advances scholarly inquiry into potential reforms for the Article 9 filing system at a time when states are actively considering adoption of the proposed reform. See, e.g., Wyoming Legislature, *Select Committee on Blockchain, Financial Tech. & Digital Innovation Tech. Meeting*, Nov. 2, 2020, YOUTUBE (Nov. 2, 2020), <https://perma.cc/NEZ3-TAK6>. It also makes a contribution to the field of computational law by unveiling an approach, thanks to Lexon, that is more accessible to a broader range of lawyers, judges, legal academics, and lawmakers, than more traditional computational legal inquiries. The dual nature of the function of the software in this Article reflects the dual nature of the Lexon code, as both strictly logic program and plain English description of itself.

100. See UCC-1 Financing Statement (Form UCC1), <https://perma.cc/8UPV-KKBR> (PDF).

101. DIEDRICH, LEXON BIBLE, *supra* note 16, at 15.

102. “Computational Law is that branch of legal informatics concerned with the codification of regulations in precise, computable form.” MICHAEL GENESERETH, COMPUTATIONAL LAW: THE COP IN THE BACKSEAT 2 (2015), <https://perma.cc/39HT-CGFE> (PDF).

proposed form achieves more than merely creating a complicated electronic version of the existing form. Rather, the Lexon U.C.C. Financing Statement seeks to automate as many of the existing Article 9 rules related to U.C.C.-1 initial financing statement and U.C.C.-3 continuation and termination statements as possible. To that end, this Part next uncovers the various provisions of Article 9 automated by the Lexon U.C.C. Financing Statement. This Part concludes by arguing that states can seamlessly adopt the Lexon U.C.C. Financing Statement without adopting non-uniform amendments to the U.C.C. in order to prove out the final core goal of the project: creating a system that can generally be used now, under existing law.

A. *The Prototype Uses Lexon to Enable Transparency for a Public System*

The Lexon U.C.C. Financing Statement applies computational law principles to commercial law. Computational law, generally speaking, seeks to represent law in formal logic in order to apply computational techniques in predicting the consequences of law applied to specific facts.¹⁰³ Interest in computational law stems from interest in better tailoring legal rules, decreasing monitoring and enforcement costs, and reducing risks.¹⁰⁴ Although new areas of computational legal study continue to emerge,¹⁰⁵ computational contracting remains

103. See *id.* (defining computational law); JEFFERY ATIK & VALENTIN JEUTNER, QUANTUM COMPUTING AND ALGORITHMIC LAW 10 (2019), <https://perma.cc/B3Z8-2NDE> (PDF) (“It involves forming legal algorithms that proceed through logical processes . . . to create legal conclusions.”).

104. See Throsten Kaeseberg, *The Code-ification of Law and Its Potential Effects*, 2 STAN. J. BLOCKCHAIN L. & POL’Y 232, 233 (2019) (acknowledging that digital technologies reduce costs); James Grimmelman, Note, *Regulation by Software*, 114 YALE L.J. 1719, 1721 (2005) (“Computer software . . . can constrain behavior as effectively as law can.”); Harry Surden, *Machine Learning and Law*, 89 WASH. L. REV. 87, 101 (2014) (highlighting the potential for legal efficiencies as a result of machine learning); John O. McGinnis & Steven Wasick, *Law’s Algorithm*, 66 FLA. L. REV. 991, 993 (2014) (noting that “the computational revolution reduces . . . information costs” and thereby enables the law to better uphold its values and goals).

105. See, e.g., Thibault Schrepel, *Computational Antitrust: An Introduction and Research Agenda*, 1 STAN. COMPUTATIONAL ANTITRUST 1, 2–4 (2021) (exploring the new field of computational antitrust).

among the longest standing branches of computational law¹⁰⁶ and continues to capture legal academic interest.¹⁰⁷ Computational law traditionally views adjudication and language as two of the most significant barriers to advances in computational contracting.¹⁰⁸ Lexon seeks to break through both barriers¹⁰⁹ by being a computer programming language that anyone, including a judicial arbiter, can read—“the first of a new generation of languages.”¹¹⁰

Lexon turns traditional wisdom about the genesis of a programming language on its head; rather than designing for what a computer can read, Lexon starts from natural language and limits it down to a regular ruleset that is just unambiguous enough that a computer can process it.¹¹¹ In doing so, Lexon

“shortcuts” the process of natural language processing . . . and implements . . . a *pass-through principle*] instead of trying to achieve intelligent “understanding” of its input Accordingly, no representation of thought beyond the language input itself is attempted by the Lexon compiler, i.e. no translation of

106. Inquiries into computational contracting can be identified in the academic literature as early as the 1950s. See Megan Ma, *Writing in Sign: Code as the Next Contract Language?*, MIT COMPUTATIONAL L. REP., Aug. 14, 2020, at 5.

107. See generally Harry Surden, *Computable Contracts*, 46 U.C. DAVIS L. REV. 629 (2012); Werbach & Cornell, *supra* note 82; Scholz, *supra* note 94; Sudhir Agarwal et al., *Toward Machine-Understandable Contracts*, ARTIFICIAL INTEL. FOR JUST. 1, 1–8 (2016), <https://perma.cc/QT2F-GXYT> (PDF).

108. See Ma, *supra* note 106, at 4 (discussing the present state of computational contracting).

109. See DIEDRICH, LEXON BIBLE, *supra* note 16, at 13 (“Lexon solves some of the major challenges of Computational Law.”).

110. *Id.* at vi.

111. See *id.* at 16 (“[T]he models underlying the tools used to create programming languages come from linguistics and were originally designed to reason about natural languages. But it’s not how they are used today.”); Ma, *supra* note 106, at 24

Lexon uses a subset of natural language grammar as the programming language of the legal contract. This approach is known as “controlled natural language.” Rather than processing *all* of natural language, a machine needs only to process an assigned vocabulary and grammar. The assigned set becomes the operatives of the language game. Additionally, Lexon embraces the legacy of Chomskyan formal semantics, whereby the syntactic structure is both a projection and a vessel of its function.

meaning into bits and bytes. Instead, the basic devices of compiler building are turned on processing natural language *itself*, as if it were a program.¹¹²

Lexon's design enables anyone to read and understand code, such that users no longer need to trust smart contract developers' description of the code.¹¹³ Essentially, Lexon—the computer code—reads like English to such an extent that no deep prior study is required to understand the program.¹¹⁴

Because Lexon enables the embedding of natural language code directly into other non-code written language,¹¹⁵ each Lexon program starts with an indicator—LEX—and a short title for the code that follows.¹¹⁶ In the case of the Lexon U.C.C. Financing Statement, the head and title are “LEX UCC Financing Statement.”¹¹⁷ Then, just as many contracts contain definition sections, a Lexon smart contract follows the head with definitions.¹¹⁸ These correspond to a programmer's type declarations.¹¹⁹ The definitions are followed by Lexon recitals, which provide the foundation for the clauses that will follow.¹²⁰ In drafting legally enforceable contracts, lawyers use recitals to “describe the background and purpose of a contract.”¹²¹ Lexon recitals fulfill a similar function—they build the foundation of the program that will enable the rest of the program to be performed.¹²² The recitals are then followed by Lexon clauses,¹²³

112. HENNING DIEDRICH, LEXON: DIGITAL CONTRACTS 10 (2020).

113. DIEDRICH, LEXON BIBLE, *supra*, note 16, at 5.

114. *Id.* at 23 (“Lexon code is optimized for reading, because programs, as well as contracts, are usually more often read than written, and usually are read by more people than there are writers.”).

115. *Id.* at 7.

116. *Id.*

117. *See infra* APPENDIX A.

118. *See* DIEDRICH, LEXON BIBLE, *supra* note 16, at 7 (outlining the format of a Lexon digital contract).

119. *See id.* at 7 (“[Definitions] are similar to how a lawyer is used to finding them in a legal contract—and which any programmer knows from *type declarations*.”).

120. *See id.* at 8 (explaining that recitals spell out what is to happen at the beginning of the digital contract, before any clause can be executed).

121. TINA L. STARK, DRAFTING CONTRACTS: HOW AND WHY LAWYERS DO WHAT THEY DO 80 (2d ed. 2014).

122. *See* DIEDRICH, LEXON BIBLE, *supra* note 16, at 8.

123. *See id.* at 27.

which themselves possess a structure that mirrors the overall structure of the Lexon program: head, definitions, permissions, conditions, and statements.¹²⁴ To create any of these pieces of a program in Lexon, a drafter would write sentences with subjects and predicates, the same as when writing sentences in natural language.¹²⁵ Ultimately, a program in Lexon is expressed in natural language while simultaneously being readable by machines,¹²⁶ with the vocabulary, the grammar, and the document structure all comprising part of the program.¹²⁷ This dual readability makes Lexon the programming language of choice for the Lexon U.C.C. Financing Statement. The U.C.C. Article 9 filing system is operated and maintained by a public agency, and serves a public function.¹²⁸ How to maintain transparency of computational techniques when applied to public systems is a core issue explored in computational legal literature.¹²⁹ The Lexon U.C.C. Financing Statement, cognizant of that issue, responds to the concern by maintaining the same level of transparency currently enjoyed by users of the filing system—the ability to read and understand the financing statement in English.

124. *See id.* at 29.

125. *See Ma, supra* note 106, at 14 (“Lexon uses a subset of natural language grammar as the programming language of the legal contract.”).

126. *See id.* at 15

Lexon uses Context Free Grammars (CFG). First theorized by Chomsky, CFG does not depend on context; its rules operate independently of the objects in question. Chomsky had originally developed CFG in an effort to formalize natural language rule structures. While this was largely unsuccessful in linguistics, it has since been popularized in computer science. Consequently, Lexon applies the model to create a programming language that is both expressible in natural language and readable by machines.

127. DIEDRICH, LEXON BIBLE, *supra* note 16, at 15.

128. *See Schroeder, supra* note 13, at 47 (noting that a state’s U.C.C. filing system is usually maintained by the state’s Secretary of State).

129. *See, e.g.,* Deven R. Desai & Joshua A. Kroll, *Trust but Verify: A Guide to Algorithms and the Law*, 31 HARV. J.L. & TECH. 1, 9 (2017) (arguing “that the demands for transparency must confront the realities of computer science considering software”).

B. *The Prototype Creates a Smart U.C.C.-1 Form, but also Fills the Gaps Between How the Filing System Was Intended to Work and How It Actually Works*

At present, when a secured party files a financing statement, two common filing methods predominate: filing the U.C.C. Section 9-521 provided U.C.C.-1 form, often by filling out and submitting a PDF, or by electronic filings.¹³⁰ Although the § 9-521 form seems outdated in terms of available technology, use of the form persists because § 9-521 insists that “a filing office that accepts written records may not refuse to accept a written initial financing statement” filed using the U.C.C.-1 form, except for very specific and limited reasons detailed in § 9-516(b).¹³¹ Further, practices related to the acceptance of electronic filings vary widely.¹³² Some filing offices limit the acceptance of electronic filings to the electronic submission of the PDF form.¹³³ Others provide direct electronic filings through

130. Although most states offer filers the option of paper or electronic filing, all states allow for some form of electronically filing the U.C.C.-1 form. See CORP. SERV. CO., UCC ELECTRONIC FILING METHODS AND CAPABILITIES BY STATE 3 (2016), <https://perma.cc/FA5D-AF3Q> (PDF) (presenting how all states file U.C.C. forms). All states except Arizona and Louisiana allow for electronic filing of the U.C.C.-3 termination statement. See *Uniform Commercial Code*, ST. OF ARIZ., <https://perma.cc/89U7-CQ5E> (listing options to file U.C.C.-3 termination statements in person or by mail, but not electronically); *FILE UCC ONLINE*, LA. SEC’Y OF ST. (2021), <https://perma.cc/3MZ2-XK7B> (providing the ability to electronically file U.C.C.-1 and U.C.C.-1F forms, but not U.C.C.-3 forms).

131. U.C.C. § 9-521(a) (AM. L. INST. & UNIF. L. COMM’N 2010); *id.* § 9-516(b).

132. See LOPUCKI ET AL., *supra* note 4, at 295

The statewide UCC systems have switched from micro-media to digital storage of financing statements. The filer may send an image of a financing statement—as it appears in UCC § 9-521, but with the blanks filled in—or merely transmit the data necessary to fill the blanks in. In the later event, the filing office creates the financing statement automatically. The financing statements are stored as digital images. Searchers can locate them through the index, view them on the screen, and download them. Still, for both technological and political reasons it remains impossible to search the full text of the financing statements online. Online searches are limited to the index.

133. See CORP. SERV. CO., *supra* note 130, at 6–7 (explaining that Arizona and Georgia, among other states, require uploading a PDF, and that Washington, D.C. requires that filers upload “an image of the UCC record with accompanying index data”).

the use of an interactive user interface.¹³⁴ While most states that allow electronic filings still allow the option to submit the paper U.C.C.-1 form, several states require all U.C.C. filings to be submitted electronically.¹³⁵ Notably, for the states that allow interactive web filings, adopting the Lexon U.C.C. Financing Statement proposed here would not be a very heavy lift.¹³⁶ It merely requires a shift in the software on the backend that supports the user interface. Indeed, as demonstrated in Appendix C, Lexon code can compile to JavaScript, such that an interested Secretary of State's Office could very easily adapt the Lexon U.C.C. Financing Statement to run on its existing internal servers without converting to blockchain technology-based rails.¹³⁷

With that in mind, what follows are portions of the Lexon U.C.C. Financing Statement together with explanations of the related U.C.C. provisions.¹³⁸ Both the Lexon code and the

134. See, e.g., *Online UCC Filing Home Page*, COMMONWEALTH OF KY. OFF. OF SEC'Y OF ST., <https://perma.cc/F7SS-Z774>; *UCC Search & Filing*, ME. BUREAU OF CORPS., ELECTIONS, & COMM'NS (2014), <https://perma.cc/QX46-ZCM2>; *UCC Electronic Filing, Search & Retrieval*, MD. DEP'T OF ASSESSMENTS & TAX'N, <https://perma.cc/447A-GC33>; *Corporations Division*, COMMONWEALTH OF MASS. (2020), <https://perma.cc/LC4T-HS3K>. Even in states where interactive web filings are available, a filer can often opt to file a paper U.C.C.-1 form. See, e.g., *UCC Frequently Asked Questions*, MD. DEP'T OF ASSESSMENTS & TAX'N, <https://perma.cc/FVQ4-PH22> (stating that paper filings are accepted, but take longer to process than online filings); MICH. ADMIN. CODE r. 440.103 (2021) (providing the ways in which financing statements may be filed); MICH. ADMIN. CODE r. 440.107 (2021) (permitting the filing officer to adopt electronic filing procedures in addition to a paper-based system).

135. See, e.g., *Uniform Commercial Code*, DEL. DIV. OF CORPS., <https://perma.cc/TLR8-NURY> (requiring all U.C.C. filings submitted after December 1, 2015 to be filed electronically); *Frequently Asked Questions*, MONT. SEC'Y OF ST., <https://perma.cc/KB2U-BSR2> (stating that paper forms are no longer accepted); *Central Indexing System*, N.D. SEC'Y OF ST. (2015), <https://perma.cc/A44A-BF99> (stating that state law requires online filing for all filings completed after March 1, 2016); CORP. SERV. CO., *supra* note 130, at 3–4, 7 (naming Colorado, Delaware, and New Jersey as states that only accept electronic filing of U.C.C.-1 forms).

136. See Schroeder, *supra* note 13, at 47 (“[T]here is no reason under the language of the U.C.C. that a state’s filing office, usually the Secretary of State, could not establish a blockchain recording system [so] that recording on the blockchain would constitute filing with that office . . .”).

137. See *infra* APPENDIX C.

138. For a full, uninterrupted copy of the Lexon code, see *infra* APPENDIX A.

Solidity code to which it compiles are included side-by-side for comparison. Lexon can create various outputs. The Lexon U.C.C. Financing Statement Solidity output is a computer program that acts as a backend template for a web-based interactive electronic U.C.C. Article 9 filing system. The filer would not see the code; it is deployed and runs on a blockchain in the background. Rather, the filer would experience a user interface on a website that simply requires the filer to input the same information required by the existing U.C.C.-1 form. In other words, the Lexon U.C.C. Financing Statement is itself both the expression of various U.C.C. Article 9 provisions and the U.C.C.-1 form. It is a template that can be completed and filed by parties over and over again.¹³⁹ Indeed, the Lexon U.C.C. Financing Statement begins with the core fields that each filer must complete for a financing statement to be accepted by a filing office.¹⁴⁰

1. Traditional U.C.C.-1 Form Data Fields

In order to provide notice of preexisting liens to those searching the Article 9 filing system, Article 9 requires that a financing statement contain the name of the debtor, the name of the secured creditor, and an indication of the collateral.¹⁴¹ These three pieces of information are required by U.C.C. Section 9-502(a) in order for a financing statement to be effective. As a result, they are referred to by some “as ‘the three holies’ of the financing statement. Each must be given, and given correctly for the financing statement to be sufficient.”¹⁴² Article 9 also requires an initial financing statement to contain the mailing address of the debtor and secured creditor and an indication of whether the debtor is an individual or an organization.¹⁴³ Although these pieces of information are not required for the financing statement to be effective, a filing officer is directed to reject a financing statement unless it

139. See DIEDRICH, LEXON BIBLE, *supra* note 16, at 8.

140. See U.C.C. §§ 9-502(a), 9-516(b)(3) (AM. L. INST. & UNIF. L. COMM’N 2010).

141. *Id.* § 9-502(a).

142. JAMES BROOK, PROBLEMS AND CASES ON SECURED TRANSACTIONS 145 (3d ed. 2016).

143. U.C.C. § 9-516(b)(4)–(5).

contains the names of the debtor and secured creditor, and these three additional pieces of information.¹⁴⁴ Upon receipt of a sufficient financing statement and the filing fee,¹⁴⁵ the filing office must add the financing statement to the searchable public records.¹⁴⁶ The Lexon U.C.C. Financing Statement takes these U.C.C. provisions as its starting point for the computer program. The Lexon code that implements these rules, and the Solidity code that it automatically compiles to,¹⁴⁷ is provided below.¹⁴⁸

144. *See id.* § 9-520(a) (“A filing office shall refuse to accept a record for filing for a reason set forth in Section 9-516(b) and may refuse to accept a record for filing only for a reason set forth in Section 9-516(b).”).

145. *See id.* § 9-516(a) (“Except as otherwise provided in subsection (b), communication of a record to a filing office and tender of the filing fee or acceptance of the record by the filing office constitutes filing.”).

146. *See id.* § 9-519(a)(3) (“For each record in a filing office, the filing office shall . . . maintain the filed record for public inspection. . .”). Not all of the information in the financing statement is searchable, however, as the filing office need only index and make searchable the debtor’s name. *See id.* § 9-519(c) (listing the general indexing requirements). Thus, when a prospective creditor searches the filing statement for a pertinent financing statement, the creditor may currently only search an index, not the full text of the document, and the index only contains the debtor’s name as it is contained in the financing statement on file.

147. For a full, uninterrupted copy of the Solidity code output of the Lexon U.C.C. Financing Statement, see *infra* APPENDIX B.

148. I note that Lexon also compiles to Javascript. A full uninterrupted copy of the JavaScript code compiled from this Lexon code is provided in Appendix C. For brevity, I do not include it in the text of the Article. Further, Lexon compiles to Sophia, the programming language for the Aeternity blockchain. DIEDRICH, LEXON BIBLE, *supra* note 16, at 12. For brevity, I have not included that code in either the text or appendices of this Article.

<u>Lexon</u>	<u>Solidity</u>
LEX UCC Financing Statement.	<code>pragma solidity ^0.5.0;</code>
LEXON: 0.2.12	<code>contract UCC_Financing_Statement{</code>
"Financing Statement" is this contract.	<code>bytes32 file_number;</code>
"File Number" is data.	<code>uint initial_statement_date;</code>
"Initial Statement Date" is a time.	<code>address payable filer;</code>
"Filer" is a person.	<code>address payable debtor;</code>
"Debtor" is a person.	<code>address payable secured_party;</code>
"Secured Party" is a person.	<code>address payable filing_office;</code>
"Filing Office" is a person.	<code>bytes32 collateral;</code>
"Collateral" is data.	

It is worth pausing here and observing that the Lexon text is perfectly readable. Indeed, as depicted above, the Lexon code strongly resembles the definition section of a contract. Note, however, that these definitions are building a template.¹⁴⁹ The terms “data,” “time,” and “person” are type declarations that alert the computer to the type of data to expect and the kind of function that can be performed on it.¹⁵⁰ At the level of the user interface, however, the user on the webpage would see fillable text boxes in which to enter the file number, the name, address, and public key address of the debtor, secured party, and the filer, and to include a description of the collateral. To illustrate the intricacies of the code, even when it looks so simple, consider each element here in comparison to its corresponding U.C.C. definition.

The Lexon U.C.C. Financing Statement includes the three key parties mentioned in the current U.C.C.-1 form: the debtor, the secured party, and the filer.¹⁵¹ By designating each of these as a person, the resulting Solidity code relates the name and address to a public-private key pair,¹⁵² such that each of the

149. See DIEDRICH, LEXON BIBLE, *supra* note 16, at 8.

150. See *id.* at 7.

151. See U.C.C. § 9-521 (AM. L. INST. & UNIF. L. COMM'N 2010).

152. BRUCE SCHNEIER, APPLIED CRYPTOGRAPHY: PROTOCOLS, ALGORITHMS, AND SOURCE CODE IN C 4–5 (2d ed. 1996)

Public key algorithms (also called asymmetric algorithms) are designed so that the key used for encryption is different from the key used for decryption. Furthermore, the decryption key cannot (at

parties can be known by a public key address,¹⁵³ and can each authorize certain activity within the system. Essentially, the type declaration corresponding to “person” creates an address payable type contract attribute in Solidity code.¹⁵⁴ As a result, each “person” identified in the Lexon U.C.C. Financing Statement must operate in the system via a public-private key pair. The filing office can simply assign the address to the parties, similar to the way it assigns a Secretary of State account ID to users of its electronic filing system.¹⁵⁵ Clearly then, some of the administrative mechanics, in terms of backend software that makes a state filing system work, will change if a Lexon U.C.C. Financing Statement is adopted. Meanwhile, the actual legal rules remain constant.

For example, the U.C.C. definitions of debtor (“a person having an interest, other than a security interest or other lien, in the collateral, whether or not the person is an obligor”)¹⁵⁶ and secured party (“a person in whose favor a security interest is created or provided for under a security agreement”)¹⁵⁷ do not change. In order to continue to capture the required information of the U.C.C.-1 form—name, address, and entity or individual debtor status—the filer would complete the fields as normal, and then a middleware layer¹⁵⁸ of software, or a service like The

least in any reasonable amount of time) be calculated from the encryption key. The algorithms are called “public key” because the encryption key can be made public: A complete stranger can use the encryption key to encrypt a message, but only a specific person with the corresponding decryption key can decrypt the message. In these systems, the encryption key is called the public key and the decryption key is often called the private key.

153. See NARAYANAN ET AL., *supra* note 23, at 18–19 (“The idea is to take a public key, one of those public verification keys from a digital signature scheme, and equate it to an identity of a person or an actor in a system.”).

154. See DIEDRICH, LEXON BIBLE, *supra* note 16, at 223 (explaining that the term ‘person’ denotes a blockchain account or address).

155. See, e.g., *Texas Secretary of State SOSDirect System*, TEX. SEC’Y OF ST., <https://perma.cc/3D3L-4S9W>.

156. U.C.C. § 9-102(a)(28)(A) (AM. L. INST. & UNIF. L. COMM’N 2010).

157. *Id.* § 9-102(a)(73)(A).

158. See *Middleware*, PCMAG ENCYCLOPEDIA, <https://perma.cc/Y39H-7HUN> (“Software that functions as a conversion or translation layer. Middleware is also a consolidator and integrator.”).

Graph,¹⁵⁹ would associate each of those entries with the public key address created for each. The debtor's name could be used to look up the public key, or the public key address could be used to look up the debtor's name. As a result, this approach retains compliance with the rule that the filing office index financing statements by the debtor's name,¹⁶⁰ but also allows for additional search mechanisms.¹⁶¹ In terms of the collateral description, the Lexon U.C.C. Financing Statement intends for the rules related to collateral descriptions to remain unchanged, although that is difficult to capture in this short excerpt of the program code, which simply represents this fact by defining collateral with the type name "data," which is implemented differently for different underlying blockchain protocols.¹⁶² With the definitions set, the Lexon U.C.C. Financing Statement next begins setting up the architecture upon which the U.C.C. Article 9 filing system runs, spelling out precisely and concisely what filers and participants in the secured lending marketplace often take for granted.

2. Recitals and Clauses to Formalize the Logic of the Article 9 Filing System

Implementing legal requirements through code requires taking a step back from the text of the law itself, considering the broader elements of the system that enable the law to function and identifying which of those elements need to be formalized in the logic of the code.¹⁶³ For example, when a person or entity becomes a secured party, and often even when just

159. The Graph is a blockchain service that indexes, for example, Ethereum smart contracts, to make them searchable by arbitrary criteria. *APIs for a Vibrant Decentralized Future*, GRAPH, <https://perma.cc/E5EH-AH9P>.

160. See U.C.C. § 9-519 (a)(3)–(4) (setting forth rules for maintenance and indexing of files).

161. See *infra* Part III.A.1.

162. See DIEDRICH, LEXON BIBLE, *supra* note 16, at 169 (explaining that in Lexon, the type name data "is used to state that a name stands for the unique identifier for a data set, e.g. its cryptographic hash").

163. The founder of Lexon, Henning Diedrich refers to this as "the puzzling triviality of Lexon code," by which he means that Lexon requires one to express everything, including information previously considered among that which is absolutely taken-for-granted. Letter from Henning Diedrich, Founder of Lexon (Feb. 12, 2021) (on file with author).

contemplating a deal,¹⁶⁴ the person or entity, having obtained debtor authorization, will give instructions that a financing statement be filed.¹⁶⁵ The person actually filing the financing statement, the filer, may or may not be an employee of the secured party.¹⁶⁶ The secured party may, for example, ask outside counsel to file the financing statement as part of preparing the many deal documents that underlie the ultimate transaction.¹⁶⁷ The fact that someone needs to actually fill out the fields of the financing statement and provide it to the filing office is something most secured lending participants do not actively think about. And yet, without this very mundane reality, no filing would ever occur. In order to create the Lexon U.C.C. Financing Statement, the code must capture and formalize this background requirement. The Lexon U.C.C. Financing Statement does so via a simple recital sentence, provided, together with the Solidity compiled from it, below.

<u>Lexon</u>	<u>Solidity</u>
<p>The Filer fixes the Filing Office, fixes the Debtor, fixes the Secured Party, and fixes the Collateral.</p>	<pre> constructor(address payable _filing_office, address payable _debtor, address payable _secured_party, bytes32 _collateral) public payable { filer=msg.sender; filing_office=_filing_office; debtor=_debtor; secured_party=_secured_party; collateral=_collateral; } </pre>

The role of the filer is not the only background function that must be formalized in the logic of the Lexon U.C.C. Financing Statement. Each state adopting the U.C.C. designates a “filing office” in U.C.C. Section 9-501¹⁶⁸ as the place that secured

164. HARRIS & MOONEY, *supra* note 4, at 251 (“[T]he current rule affords secured parties the opportunity to fix their place in line by filing a financing statement even before the details of a secured loan have been finalized.”).

165. See U.C.C. § 9-509 (AM. L. INST. & UNIF. L. COMM’N 2010) (outlining the authorization requirement and certain narrow exceptions).

166. See *id.* § 9-509 cmt. 2 (“As long as the appropriate person authorizes the filing, or, in the case of a termination statement, the debtor is entitled to the termination, it is insignificant whether the secured party or another person files any given record.”).

167. See *id.* (“Under these sections, the identity of the person who effects a filing is immaterial.”).

168. See *id.* § 9-501 (demonstrating the designation requirement); see also ZIERDT ET AL., *supra* note 4, at 606 n.1 (noting that most states designate the filing office as the Secretary of State’s office, with some exceptions).

parties file financing statements.¹⁶⁹ The U.C.C. requires the filing office to “assign [the file number and] create a record that bears the number assigned to the filed record and the date and time of filing.”¹⁷⁰ Even this very routine aspect of filing financing statements must be captured in the code. To that end, the first two clauses of the Lexon U.C.C.-1 Financing Statement program focus on the Filing Office fulfilling these duties. Note the use of the word “may” in the clauses below. In Lexon, the term “may” denotes the actor “allowed to initiate the performance of a given clause.”¹⁷¹ Meanwhile, the term “certify” indicates that upon the filing office’s receipt of the financing statement, the filing office will assign a file number to that financing statement.¹⁷² The filing office could opt to assign a hash of the data in the financing statement as the file number.¹⁷³ Further, when the filing office “fixes” the Initial Statement Date as the current time, it sets the initial financing statement filing date as the point in time that the filing office receives the data via the Lexon U.C.C. Financing Statement.¹⁷⁴ Thus, together, these two clauses fulfill the filing office obligations to assign a unique number to, and record the date of, each record filed.¹⁷⁵ Particularly where the filing office uses a hash as the file number, the Lexon U.C.C. Financing Statement fulfills the filing office duty to use a file number that includes “a digit that (1) is mathematically derived from or related to the other digits of the file number; and (2) aids the

169. See U.C.C. § 9-102(37) (“Filing Office’ means an office designated in Section 9-501 as the place to file a financing statement.”).

170. *Id.* § 9-519(a); accord *id.* § 9-102(36) (“File Number’ means the number assigned to an initial filing statement pursuant to Section 9-519(a).”).

171. See DIEDRICH, LEXON BIBLE, *supra* note 16, at 204. “Only the person named before ‘may’ can initiate the performance of what is described in a clause. No one can initiate the performance of a clause that is not lead in with a ‘may,’ except by reference from another clause.” *Id.*

172. See *id.* at 160.

173. Indeed, in Lexon 0.3, the Lexon U.C.C. Financing Statement automatically assigns a hash. The best version of this cryptolaw for U.C.C. envisions the use of a hash for a file number. See *infra* APPENDIX C.

174. See DIEDRICH, LEXON BIBLE, *supra* note 16, at 183 (stating that the term “fix” allows the actor to assign meaning to a type name); *id.* at 168 (stating that the term “current time” is a value for the point in time at which the actor of the clause acts).

175. See U.C.C. § 9-519(a) (AM. L. INST. & UNIF. L. COMM’N 2010) (mandating that the filing office assign unique numbers to each filed record and make the records available for search by the public).

filing office in determining whether a number communicated as the file number includes a single digit or transpositional error.”¹⁷⁶ That is what hashes do.¹⁷⁷ Finally, with the help of some middleware, or a service like The Graph,¹⁷⁸ the financing statement remains searchable by the public.

It is also worth noting that the Lexon U.C.C. Financing Statement uses the defined term Initial Statement Date to reflect current U.C.C. Article 9 practice, in which the Initial Financing Statement refers to the original U.C.C.-1 form perfecting a security interest, while the term Financing Statement refers to the Initial Financing Statement, together with all subsequent amendments.¹⁷⁹ The Lexon clauses and corresponding Solidity functions that perform the basic obligations of the filing office are provided below.

<u>Lexon</u>	<u>Solidity</u>
<p>Clause: Certify. The Filing Office may certify the File Number. Clause: Set File Date. The Filing Office may fix the Initial Statement Date as the current time.</p>	<pre>function Certify(bytes32 _file_number) external { if (msg.sender == _filing_office){ file_number=_file_number; }else{ require(false); } } function Set_File_Date() external { if (msg.sender == _filing_office){ initial_statement_date=now; }else{ require(false); } }</pre>

Together, the recitals and the first two clauses already reduce the gap between the law in the books and the law in action in ways that limit risk to secured parties. By performing the filing office obligations to index and make searchable filed financing statements, the Lexon U.C.C. Financing Statement reduces the room for error by the filing office. Typically, risk of loss resulting from any indexing or other errors on the part of the filing office

176. *Id.* § 9-519(b).

177. See SCHNEIER, *supra* note 152, at 30 (“A hash function is a function, mathematical or otherwise, that takes a variable-length input string (called a pre-image) and converts it to a fixed-length (generally smaller) output string (called a hash value).”).

178. See *APIs for a Vibrant Decentralized Future*, *supra* note 159.

179. See U.C.C. § 9-102(39) (“‘Financing Statement’ means a record or records composed of an initial financing statement and any filed record relating to the initial financing statement.”).

are borne by either the filing party,¹⁸⁰ or a future searcher unable to find records as a result of the error.¹⁸¹ Thus, by reducing the opportunity for error in the way the filing office fulfills the requirements of Article 9, the Lexon U.C.C. Financing Statement improves system operation and reduces risk.

3. Lapse, Continuation, and Clearing

“A filing system consists not only of the filed records, but also of subsystems for (1) adding new records, (2) searching among the records, and (3) removing obsolete records.”¹⁸² In the code discussed so far, the Lexon U.C.C. Financing Statement addressed the process for creating new records¹⁸³ and how to search for them,¹⁸⁴ and now turns to the process for removing obsolete records. A financing statement will lapse after five years unless a continuation statement is filed during the six-month window directly prior to the lapse date.¹⁸⁵ If a secured party or its filer files a continuation statement within that continuation window, the lapse date will be reset an additional five years after what would have been the lapse date.¹⁸⁶ A secured party or its filer may also file a termination statement to signal the extinguishment of a security interest.¹⁸⁷ One year after either lapse or termination, the filing office may clear the financing statement from the filing system.¹⁸⁸

Although these rules may seem relatively simple, filers often unintentionally forget to file a continuation statement during the six-month continuation window,¹⁸⁹ and many filing

180. See LOPUCKI ET AL., *supra* note 4, at 311–13 (explaining when incorrect information will still be effective against certain types of parties).

181. See U.C.C. § 9-517 (AM. L. INST. & UNIF. L. COMM’N 2010) (“The failure of the filing office to index a record correctly does not affect the effectiveness of the filed record.”).

182. LOPUCKI ET AL., *supra* note 4, at 294.

183. Functionality native to the Lexon U.C.C. Financing Statement.

184. Functionality enabled by middleware or a blockchain service like The Graph.

185. U.C.C. § 9-515(a), (c)–(d).

186. *Id.* § 9-515(e).

187. *Id.* § 9-513.

188. *Id.* § 9-522.

189. LOPUCKI ET AL., *supra* note 4, at 388.

offices excel at accepting new filings, but struggle to clear out obsolete records so that “[t]he store of records simply grows each year.”¹⁹⁰ The Lexon U.C.C. Financing Statement includes the rules related to lapse, continuation, termination, and post-lapse clearance of records in order to clarify the timing responsibilities of the secured party of record and reduce the number of records against which a search must be conducted.¹⁹¹ Doing so improves efficiencies in the filing system and reduces secured party risk. Doing so also clarifies the Lexon U.C.C. Financing Statement contribution to computational law. In the clauses related to lapse, continuation, and clearing, more clearly than elsewhere in the Lexon U.C.C. Financing Statement, the Lexon code simultaneously restates law and performs it. As a result, these clauses clearly demonstrate the way in which Lexon can act as its own specification. The law in the books *literally* becomes the law in action.

190. *Id.* at 294.

191. To emphasize for clarity, the search capacity will be enabled by middleware or a blockchain service like The Graph.

<u>Lexon</u>	<u>Solidity</u>
<p>"Continuation Window Start" is a time. "Continuation Statement Date" is a time. "Continuation Statement Filing Number" is data. "Lapse Date" is a time. "Continuation Statement" is a binary. "Termination Statement" is a binary. "Termination Statement Time" is a time. Clause: Set Lapse. The Filing Office may fix the Lapse Date. Clause: Set Continuation Start. The Filing Office may fix the Continuation Window Start. Clause: File Continuation. The Secured Party may file the Continuation Statement.</p>	<pre> uint continuation_window_start; uint continuation_statement_date; bytes32 continuation_statement_filing_number; uint lapse_date; bool continuation_statement; bool termination_statement; uint termination_statement_time; function Set_Lapse(uint _lapse_date) external { if (msg.sender == _filing_office){ lapse_date=_lapse_date; }else{ require(false); } } function Set_Continuation_Start(uint _continuation_window_start) external { if (msg.sender == _filing_office){ continuation_window_start=_continuation_window_start; }else{ require(false); } } function File_Continuation() external { if (msg.sender == secured_party){ continuation_statement=true; }else{ require(false); } } </pre>

In the portion of the Lexon U.C.C. Financing Statement code reproduced above, the terms “Continuation Window Start,” “Continuation Statement Date,” and “Lapse Date” are all designated as a “time.” The type name “time” is used in Lexon “to state that a name means a timestamp, consisting of a date and a time with precision of one second.”¹⁹² The terms “Continuation Statement” and “Termination Statement” are designated as a “binary,” which Lexon uses to refer to a type name that is either yes or no, or true or false.¹⁹³ The filing office can set the dates via automatic calculations when it certifies the filing of the initial financing statement. The secured party or its filer can simply check a box to indicate continuation or termination, and then also supply any further or changed information that would normally be supplied in a U.C.C.-3 form.¹⁹⁴ Note how closely this replicates the current process: in a U.C.C.-3 form the secured party or its filer checks a box to signal whether the filing is a continuation or termination statement.¹⁹⁵

192. DIEDRICH, LEXON BIBLE, *supra* note 16, at 257.

193. *Id.* at 157.

194. U.C.C. § 9-521(b) (AM. L. INST. & UNIF. L. COMM’N 2010).

195. *Id.*

In the portion of the code reproduced below, the Lexon U.C.C. Financing Statement automates resetting the lapse date once a secured party has filed a continuation statement and automates the clearance of old records once a secured party has terminated a financing statement.

Lexon	Solidity
<p>Clause: Set Continuation Lapse. The Filing Office may, if the Continuation Statement is filed, fix the Continuation Statement Date.</p> <p>Clause: File Termination. The Secured Party may file a Termination Statement, and certify the Termination Statement Time as the then current time.</p> <p>Clause: Termination Period. "Termination Period" is defined as 365 days after the Termination Statement Time.</p> <p>Clause: Terminate and Clear. The Filing Office may, if the Termination Period has passed, terminate this contract.</p>	<pre> function Set_Continuation_Lapse(uint _continuation_statement_date) external { if (msg.sender == _filing_office){ if(continuation_statement){ continuation_statement_date=_continuation_statement_date; } }else{ require(false); } } function File_Termination() external { if (msg.sender == secured_party){ termination_statement=true; termination_statement_time=now; }else{ require(false); } } function Termination_Period() public view returns(uint) { return termination_statement_time + 365 * 24 * 60 * 60; } function Terminate_and_Clear() external { if (msg.sender == _filing_office){ if(Termination_Period() <= now){ selfdestruct(debtor); } }else{ require(false); } } </pre>

At this point, the code of the Lexon U.C.C. Financing Statement implements most of the Article 9 provisions relating to financing statements and reduces the likelihood that others expose either the filing party or searchers using the system to certain risks of economic loss common under the current regime. Indeed, the Lexon U.C.C. Financing Statement implements U.C.C. Sections 9-502(a) (sufficiency of financing statement), 9-510 (effectiveness of filed record), 9-511 (secured party of record), 9-515 (duration of financing statement), 9-516 (what constitutes filing and refusal grounds), 9-519 (numbering, maintaining, indexing and communicating information in records), 9-520 (acceptance and refusal to accept record), 9-521 (safe harbor U.C.C.-1 and U.C.C.-3 forms), and 9-522 (maintenance and destruction of records); reduces risk of loss from Sections 9-503 (name of debtor and secured party), 9-504 (indication of collateral), 9-506 (errors and omissions), 9-509 (debtor authorization), 9-517 (effect of indexing errors), 9-524 (filing office delay); and otherwise, does not interfere with the parties' ability to fulfill the requirements of Sections 9-512 (amending financing statement), 9-513 (termination statement), 9-518 (claim related to inaccurate or wrongful record), 9-525 (fees), 9-526 (filing office rules), and 9-527 (duty to report).

Notably, the Lexon U.C.C. Financing Statement code presented up to this point was written using Lexon version 0.2. Lexon, as a bleeding edge programming language, remains under active development. As its developers receive input from attorneys and potential use cases they create, Lexon becomes more elegant and looks more like what lawyers expect in terms of legal written expression. For example, next generation Lexon code (version 0.3) for the Lexon U.C.C. Financing Statement would read, as provided partially below, where the brackets are fillable fields, and certain elements of the definitions are more clearly optional, such as assigning a hash to specific collateral. Further, the next iteration of Lexon will enable making more plain the connection between a filer, debtor, or secured party's name and their public key in so far as providing greater transparency into how the form operates on the backend.¹⁹⁶

LEX UCC Financing Statement

TERMS per UCC Financing Statement:

"File Number" is a hash.

"Filer" means a person by the name [name], with phone number of [phone number] and email of [email].

"Debtor" means a person by the name [name], identified by [Public Key], located at [mailing address, city, state, postal code, country].

"Secured Party" means a person by the name [name], identified by [Public Key], located at [mailing address, city, state, postal code, country].

"Filing Office" means a person by the name [name], identified by [Public Key].

"Collateral" means a [UCC Category of Collateral], identified by the number: [hash].

"UCC Category of Collateral" means one of "..", "..".

The rest of this incrementally more readable version of the Lexon U.C.C. Financing Statement is available in Appendix D. Importantly, the Lexon 0.3 version of the Lexon U.C.C. Financing Statement retains all of the same functionality in terms of implementing provisions of Article 9 and reducing risk for those participating in the filing system as the Lexon 0.2 version.

196. Recall, of course, that the filer and searcher would not necessarily need to know the public key associated with the name—just the debtor's name.

C. *The Prototype Could Be Used Under Existing Law*

Each state determines where secured creditors file financing statements.¹⁹⁷ For example, the Texas U.C.C. names the Texas Secretary of State as the filing office for the purpose of U.C.C.-1 filings.¹⁹⁸ Indeed, most states name the Secretary of State's Office as the filing office, but not all.¹⁹⁹ In any event, the designated filing office maintains its own filing office rules.²⁰⁰ Under those rules, states may elect to allow electronic U.C.C.-1 filings, including filings by electronic data entry.²⁰¹ For states that already allow electronic U.C.C.-1 filings by electronic data entry, adopting a smart-contract-based U.C.C.-1 financing statement only represents an incremental step in improving the existing system. Namely, a smart contract system like the one proposed here could seamlessly be adopted on the backend with only minimal required changes to the public-facing user interface. Further, states that allow electronic filings often offer it as an option, retaining the option to use the traditional U.C.C.-1 safe harbor qualifying form.²⁰² The smart-contract-based U.C.C.-1 form proposed here could also be implemented as an additional optional method of filing.²⁰³ The

197. U.C.C. § 9-501(a)(2).

198. See TEX. BUS. & COM. CODE ANN. § 9.501(a)(2) (West 2021).

199. See CORP. SERV. CO., *supra* note 130, at 4 (stating that, for example, a private company operates Florida's filing system and in Georgia, the Georgia Superior Court Clerks Cooperative Authority is the central filing authority).

200. See U.C.C. § 9-526 (AM. L. INST. & UNIF. L. COMM'N 2010).

201. See, e.g., 1 TEX. ADMIN. CODE § 95.101(6) (2021).

202. See *supra* note 130 and accompanying text. However, other states require all filings to be made electronically. See, e.g., *Uniform Commercial Code*, MISS. SEC'Y OF ST., <https://perma.cc/YYT5-TURN> (Mississippi); *FAQ - Frequently Asked Questions*, MONT. SEC'Y OF ST., <https://perma.cc/R8UA-DPSZ> (Montana); *Central Indexing (UCC)*, N.D. SEC'Y OF ST., <https://perma.cc/C94F-LAEK> (North Dakota); CORP. SERV. CO., *supra* note 130, at 4 (Colorado, Delaware, and New Jersey).

203. Because the Lexon U.C.C. Financing Statement follows the requirements of the U.C.C.-1 form in § 9-521, use of the Lexon U.C.C. Financing Statement should receive the same safe harbor as offered to those using the U.C.C.-1 form. See U.C.C. § 9-521 cmt. 2 (“[T]his section provides sample written forms that must be accepted in every filing office in the country, as long as the filing office’s rules permit it to accept written communications.”).

key to adoption, even on an optional basis, lies with the filing office.²⁰⁴

As a result, there exist two potential routes for a state to adopt a smart-contract-based U.C.C.-1 filing system. First, if a state's filing office (usually the Secretary of State) is not interested in adopting a smart-contract-based Article 9 filing system, the legislature could change U.C.C. Section 9-501(a)(2) to appoint a new filing office—namely, either a government agency that is amenable to improving the existing system with smart contracts, or a private entity (the legislative notes to § 9-501 note that the filing office need not be a government entity).²⁰⁵ The second route to adoption assumes that the existing filing office would be willing to adopt the smart-contract-based filing system with legislative authority to do so. To provide the filing office the necessary cover, the legislature could enact legislation specifically authorizing the filing office to amend its existing online data entry system for U.C.C.-1 financing statements to use a blockchain-based smart-contract enabled backend.²⁰⁶

In either case, the legislature should amend U.C.C. Section 9-521 to confirm that an electronic version of the existing U.C.C.-1 form, including electronic financing statements based on smart contracts, qualifies for the safe harbor currently provided to the PDF version of the U.C.C.-1 form.²⁰⁷ Otherwise, the Lexon U.C.C. Financing Statement is entirely compatible with the existing U.C.C. Article 9 provisions, and would remain entirely compatible with the

204. My thinking on this point has significantly matured since I first wrote *Conceptualizing Cryptolaw* in 2017. At that time, I envisioned this smart-contract-based filing system as an opportunity to eliminate the filing office altogether. While the proposed prototype certainly reduces the risk that error by the filing office will impact the rights and duties of commercially transacting parties, its adoption and maintenance rely upon the continued relevance of the filing office. Many thanks to Professor Christopher Bradley for engaging me on this point and opening my eyes to the political realities that may doom a proposal to fully eliminate a filing office.

205. U.C.C. § 9-501(b) (AM. L. INST. & UNIF. L. COMM'N 2010); see CORP. SERV. CO., *supra* note 130, at 4 (noting that a private company operates the Florida U.C.C. filing system).

206. For an example, consider a bill introduced by Wyoming, which would enable the Secretary of State's Office to adopt a system like the Lexon U.C.C. H.B. 0142, 2021 Reg. Sess., (Wyo. 2021), <https://perma.cc/TCH6-UQAX> (PDF).

207. U.C.C. § 9-521 cmt. 2; see *supra* note 203.

forthcoming revisions from the Uniform Law Commission and American Law Institute Joint Study Committee on the U.C.C. and Emerging Technologies, so that any state to adopt a smart-contract-based Article 9 filing system may remain nationally uniform in the law related to commercial transactions.

III. KEY LESSONS FROM ENGINEERING A U.C.C. FINANCING STATEMENT IN LEXON

The Lexon U.C.C. Financing Statement demonstrates potential innovations to the Article 9 filing system enabled by blockchain and smart contracts. Some such innovations are endogenously produced by moving the filing system to a smart-contract-based tool. Others are intentional changes suggested as optional opportunities to capitalize on the nature of blockchain technology to improve the Article 9 filing system. Specifically, this Part advances five potential innovations that state filing offices could introduce to harness the unique aspects of blockchain technology to improve the filing system. In doing so, this Article demonstrates the possibility of advancing the U.C.C. beyond simply creating new rules to accommodate its application to emerging technologies. Rather, the relationship between law and technology is multi-directional. Even as technology challenges law to address new activity, technology also reflects back to law areas for improvement in old paradigms. In evidencing this reality, the Lexon U.C.C. Financing Statement and the process of creating it also offer lessons about the intersection of the principle of technology neutrality and the increasing importance of lawyers' technological competence.

A. *The U.C.C. Could Be Improved Beyond Creating New Rules to Accommodate Digital Assets*

Although the Lexon U.C.C. Financing Statement aims for compatibility with existing law, the software program offers optional features that illustrate obvious potential innovations to the existing system. Other innovations emerged as endogenous features of the Lexon U.C.C. Financing Statement. In crafting the code, five such potential innovations to the Article 9 filing system emerged: (1) enabling new search capability without

imposing new indexing burdens on the filing office; (2) identifying specific collateral by hash of a serial number or other unique identifier; (3) generating new fees for the filing office by paying for reminders to file a continuation statement within the continuation window; (4) offering a way to escrow digital asset collateral seamlessly within the U.C.C.-1 form; and (5) offering debtors an optional opportunity to authorize filing statements via an electronic signature (and thereby preventing the use of unauthorized filings for harassment). The Lexon U.C.C. Financing Statement contemplates each of these features as optional features that filing offices could adopt, or not, as they see appropriate to their own circumstances. Failure to use these features does not render the Lexon U.C.C. Financing Statement inoperable.

1. New Search Capability Without New Indexing Obligations

Currently, the U.C.C. only requires a state filing office to index the debtor's name.²⁰⁸ As a result, the debtor's name, as written, becomes the only way to search for a financing statement.²⁰⁹ Although scholars suggest a variety of changes to the searchability of financing statements,²¹⁰ actual changes to the U.C.C. over the years focused on making name-related errors less likely to occur.²¹¹ This may represent a desire to improve search capabilities without imposing new indexing

208. U.C.C. § 9-519(e)(1).

209. Edward S. Adams & Steve H. Nickles, *Amending the Article Nine Filing System to Meet Current Deficiencies*, 59 MO. L. REV. 833, 840 (1994).

210. See, e.g., Kenneth C. Kettering, *Standard Search Logic Under Article 9 and the Florida Debacle*, 66 U. MIAMI L. REV. 907, 921 (2012) (proposing an amendment to the Article 9 filing system by tolerating errors in the debtor's name in a filing); Margit Livingston, *A Rose by Any Other Name Would Smell as Sweet (or Would It?): Filing and Searching in Article 9's Public Records*, 2007 BYU L. REV. 111, 124–25 (2007) (arguing that Secretary of State offices should use new computer search logic for Article 9 records that would help searchers find financing statements that contain minor errors in the debtor's name); Jonathan C. Lipson, *Secrets and Liens: The End of Notice in Commercial Finance Law*, 21 EMORY BANKR. DEVS. J. 421, 424–25 (2005) (pointing out how our laws have grown to accept and tolerate secret liens and cautioning about the collateral consequences of further undermining the U.C.C.-1 financing statement system).

211. LoPucki, *Spearing Tool Disaster*, *supra* note 22, at 285, 292–93.

obligations on the filing office.²¹² The desire to spare the filing office additional responsibility stems not only from a cost-efficiency concern, but also from the view that the indexing process often causes filing errors,²¹³ and, therefore, introducing new indexing responsibility introduces new error risks.

The Lexon U.C.C. Financing Statement contemplates enabling search by the file number, or by a public key address.²¹⁴ This is true of the Lexon U.C.C. Financing Statement because of the potential to combine inherent features of its architecture with emerging block explorer solutions, like The Graph. The result is that, without introducing any new filing office indexing obligations or any new risk related to such obligations, the Lexon U.C.C. Financing Statement, together with a middleware layer and a web-based user interface, enables search by file number, debtor public key address, and secured party public key address. Redundant search capability may enable searchers to find financing statements that they would not otherwise discover using debtor name searches alone.

2. Identify Specific Collateral Without Introducing Ambiguity

The 2001 revisions to U.C.C. Article 9 loosened the collateral description requirement for financing statements so that a super-generic description suffices to make a financing statement effective.²¹⁵ The use of super-generic collateral descriptions can cause confusion among searchers and introduce ambiguity into the filing system.²¹⁶ Drafting a more narrow collateral description in the financing statement, however, poses

212. Eikenburg, *supra* note 15, at 1632.

213. *Id.*

214. See Andrew M. Hinkes, *Throw Away the Key, or the Key Holder? Coercive Contempt for Lost or Forgotten Cryptocurrency Private Keys, or Obstinate Holders*, 16 NW. J. TECH. & INTELL. PROP. 225, 230 (2019) (describing the data endogenously made public as part of a blockchain protocol).

215. See U.C.C. §§ 9-504 cmt. 2, 9-101 cmt. 4(h) (AM. L. INST. & UNIF. L. COMM'N 2010). Notably, the description of the collateral in the security agreement may not be super-generic. *Id.* § 9-203(b)(3)(A).

216. See Helen G. Xiang, Note, *Generic Financing Statements Under Revised Article 9: A Proposed Reform*, 97 TEX. L. REV. 1061, 1065–66 (2019) (“Despite the language of Revised Article 9 clearly providing for the mere indication of the collateral to be sufficient for perfection, numerous cases have arisen out of confusion about these descriptions.”).

some risk to the secured party because it “acts as an exclusionary document, where collateral that was not described in the financing statement is unperfected.”²¹⁷ Part of the problem in drafting more narrow descriptions stems from the ambiguity often introduced by language used to describe specific collateral.²¹⁸ For example, if the financing statement lists a type of collateral located at a specific address, and the debtor later moves the collateral to a different address, does an unchanged financing statement still cover the collateral?²¹⁹

The Lexon U.C.C. Financing Statement offers an opportunity to be very specific about the collateral covered by the financing statement without introducing this type of ambiguity. Specifically, whenever collateral can be identified by a unique serial number of some kind, the Lexon U.C.C. Financing Statement could create a unique hash for that serial number, such that the hash can only refer to that specific collateral. Like the other innovations proposed in this Article, this proposal to use hashes to increase collateral description specificity need not be adopted by a filing office in order for the Lexon U.C.C. Financing Statement to function as intended. Indeed, even if adopted by a filing office, not every secured party need use it. In fact, many times collateral may not be amenable to using this option. Nevertheless, the potential to enable more specific collateral descriptions without introducing ambiguity and related risk is worth exploring. In particular, because this innovation arises endogenously from the mere act of transplanting the Article 9 provisions into code, it is an example of the type of unexpected ripple effect that should be expected at the intersection of law and technology; it is an example of cryptolaw.²²⁰

217. *Id.* at 1063 (citing *In re The Holladay House*, 387 B.R. 689, 695 (Bankr. E.D. Va. 2008)).

218. *See id.* at 1063–64 (describing the challenges of describing collateral in security agreements).

219. *See, e.g., In re Sterling United, Inc.*, 519 B.R. 586 (Bankr. W.D.N.Y. 2014); *Cnty. Tr. Bank v. First Nat’l Bank*, 924 So. 2d. 498, 502 (La. Ct. App. 2006).

220. *See Reyes, Conceptualizing Cryptolaw, supra* note 8, at 439–44 (stating that the use of crypto-legal structures may lead to unexpected results, such as the emergence of new regulatory actors, and that cryptolaw provides a framework for considering the impact of those structures on legal culture).

3. Generating New Fees

Blockchain technology has been referred to as the Internet of Value.²²¹ Regardless of the dubious technical accuracy of that reference, the point remains that blockchain technology enables unknown parties to securely conduct payment transactions electronically.²²² As a result, it seems quite natural to consider using a blockchain-based Article 9 filing system to generate new fees for the filing office. Indeed, one of the most common payment-related applications of blockchain technology is decentralized escrow transactions.²²³ Meanwhile, a common source of error for parties in the filing system stems from forgetting to file a timely continuation statement or miscalculating the continuation window and lapse date.²²⁴ To capitalize on these innate features of blockchain technology and smart contracts while solving a known problem in the Article 9 filing system, the Lexon U.C.C. Financing Statement proposes an escrow transaction between the secured party and the filing office. The secured party would offer a fee to the filing office for the service of reminding the secured party that the continuation window has begun. To construct this arrangement, the secured party would put the reminder fee into escrow, and if the filing office sent a reminder (the notification statement in the code reproduced below), then the filing office would be entitled to receive payment of the fee from escrow. The filing office is under no obligation to send the reminder, and if it elects not to do so, the reminder fee would be returned to the secured party upon filing of a termination statement. Notably, the optional nature of the filing office's responsibility vis-à-vis the escrow transaction is a feature of all smart contracts, not a bug. Smart contracts cannot force action by a party, rather, they incentivize performance.

221. See MARIA KARAJOVIC ET AL., BLOCKCHAIN: A MANAGER'S GUIDE § 3.4 (2016), <https://perma.cc/HXA9-ZPWF> (PDF).

222. See NARAYANAN ET AL., *supra* note 23, at 51–55 (detailing the mechanics of bitcoin transactions).

223. *Id.* at 60–61.

224. LOPUCKI ET AL., *supra* note 4, at 388 (“Failure to file a necessary continuation statement timely is both a common error and a common source of legal malpractice claims.”).

Lexon

"Reminder Fee" is an amount.
 "Notification Statement" is a text.
 Clause: Pay Fee.
 The Secured Party may pay a Reminder Fee into escrow.
 Clause: Notice.
 The Filing Office fixes the Notification Statement.
 Clause: Notify.
 The Filing Office may, if Default is declared, pay the Digital Asset Collateral to the Secured Party.
 Clause: Release Reminder Fee.
 The Filing Office may, if the Termination Statement is filed, return the Reminder Fee to the Secured Party.

Solidity

```

uint reminder_fee;
event Notification(address receiver, string notification_statement);
function Pay_Fee() external payable {
    if (msg.sender == secured_party){
        reminder_fee=msg.value;
    }else{
        require(false);
    }
}

function Notice(string memory _notification_statement) private {
    notification_statement=_notification_statement;
}

function Notify() external {
    if (msg.sender == _filing_office){
        if(continuation_window_start<=now){
            emit Notification(secured_party, notification_statement);
        }
    }else{
        require(false);
    }
}

function Release_Reminder_Fee() external {
    if (msg.sender == _filing_office){
        if(termination_statement){
        }
        secured_party.transfer(reminder_fee);
    }else{
        require(false);
    }
}

```

Again, adopting this element of the Lexon U.C.C. Financing Statement is completely optional and failure to adopt it does not prevent the Lexon U.C.C. Financing Statement from working. By including the suggestion, however, this Article demonstrates that the natural capacity of blockchain technology to incentivize efficient transactions can be used to improve the Article 9 filing system. It is also worth noting that these nine lines of Lexon code represent all that would be required to introduce this potentially powerful improvement. This simplicity stands as a testament to Lexon's contribution to computational law. Many computational law approaches use very complex computational techniques that remain inaccessible to their potential user base.²²⁵ Lexon, on the other hand, can be read and written in plain language, with no prior training. As a result, Lexon offers an opportunity to democratize computational law by enabling a wider audience to contribute to its formation and assess its applications. Further, Lexon can provide the simplicity and transparency imperative for public computational law (as opposed to private law contracts).

225. See GENESERETH, *supra* note 102, at 2.

4. Potential to Escrow Digital Assets Within the Filing System

Another potential use of the natural capacity of blockchain technology and smart contracts to enable decentralized escrow transactions centers on addressing certain difficulties with the current treatment of cryptocurrency when it acts as collateral for a secured loan. Currently, cryptocurrency used as collateral for a secured loan falls under the Article 9 collateral classification of general intangibles.²²⁶ Perfection in general intangibles requires filing a financing statement in the Article 9 filing system.²²⁷ However, general intangibles do not enjoy the benefit of strong take free rules.²²⁸ The resulting difficulty for encumbered cryptocurrency has been documented extensively in the literature²²⁹—namely, there is no good way for subsequent transferees to know whether their cryptocurrency is encumbered by a security interest under the current Article 9 regime.²³⁰ The Uniform Law Commission and American Law Institute Joint Study Committee on the Uniform Commercial Code and Emerging Technologies are currently working to address this issue by reforming these and other U.C.C.

226. See Tu, *Perfecting Bitcoin*, *supra* note 13, at 546 (“Although there is no reference to virtual currency in Article 9, the ‘general intangible’ collateral type is broad enough to encompass it.”).

227. See *id.* at 552 (“As applied to virtual currency, there is only one viable method of perfection: the filing of a financing statement that appropriately describes the collateral. Because virtual currency constitutes a general intangible, the special rules allowing for perfection by possession or control are inapplicable.”).

228. See Schroeder, *supra* note 13, at 30

[The categorization as general intangibles] has the potential of negatively affecting the marketability of bitcoin. This is because Article 9 has no negotiation rule for the buyers of general intangibles that are subject to a perfected security interest. That is, once a security interest in a general intangible is perfected, it survives even after multiple transfers to third parties.

229. See, e.g., *id.*; George K. Fogg, *The UCC and Bitcoins: Solution to Existing Fatal Flaw*, BLOOMBERG BNA (Apr. 1, 2015), <https://perma.cc/E4MT-PB5F> (PDF); Bob Lawless, *Is UCC Article 8 Bitcoin’s Savior (for Commercial Law)?*, CREDIT SLIPS (Mar. 28, 2014, 3:47 PM), <https://perma.cc/WAX6-QYJY>.

230. See Schroeder, *supra* note 13, at 30 (highlighting the issue that once a security interest is perfected it attaches to the cryptocurrency in subsequent transactions); Fogg, *supra* note 229; Lawless, *supra* note 229.

provisions to better fit the unique contexts of digital assets.²³¹ In the meantime, the Lexon U.C.C. Financing Statement offers an elegant, if radical, alternative solution to this problem.

The Lexon U.C.C. Financing Statement, *itself* built out of smart contracts and running on blockchain, could escrow digital asset collateral until such time as the obligation has been repaid and the debtor is entitled to a termination statement. Indeed, smart contracts, such as those executed on Ethereum, can hold funds, and, in this sense, have escrow built in. Typically, of course, a debtor retains the right to use encumbered assets during the life of the secured transaction.²³² However, common practice in the market for secured lending against cryptocurrency is for the secured party to take control of the cryptocurrency collateral until the debtor repays the obligation.²³³ Thus, using the Lexon U.C.C. Financing Statement to escrow the collateral does not adversely affect the debtor's position vis-à-vis existing market practice, but rather, it reduces risk related to a secured party siphoning off the collateral or refusing to return it upon repayment of the underlying obligation. The secured party, for its part, would retain its right under existing law to declare default and repossess the collateral.²³⁴ To that end, the Lexon U.C.C. Financing Statement creates a definition for "digital asset collateral," creates a binary option for signaling that a default has occurred, allows the debtor to pay the digital asset collateral into escrow, and upon the secured party's signal that default has occurred, gives the filing office the right to release the escrow to the secured party. Upon termination of the financing statement, the filing office would release the escrow to the debtor.

231. See *Uniform Commercial Code and Emerging Technology Committee Description*, UNIF. L. COMM'N (2021), <https://perma.cc/9P9N-9EJJ> (describing the Committee's approach to addressing the applicability of U.C.C. provisions to emerging technology).

232. See LOPUCKI ET AL., *supra* note 4, at 43 ("Security agreements typically require that the debtor surrender possession upon default, and some debtors actually do just that.")

233. Nguyen, *supra* note 13, at 192–93.

234. Indeed, as described here, the Digital Asset Collateral escrow mechanisms look a lot like a control agreement that a secured party may enter into with a bank related to deposit accounts to establish control under existing law. In other words, the proposal is not unprecedented in either the Article 9 system or the cryptocurrency lending economy.

Lexon

"Digital Asset Collateral" is an amount.
 "Default" is a binary.
 Clause: Pay Escrow In.
 The Debtor may pay the Digital Asset Collateral into escrow.
 Clause: Fail to Pay.
 The Secured Party may declare Default.
 Clause: Take Possession.
 The Filing Office may, if Default is declared, pay the Digital Asset Collateral to the Secured Party.
 Clause: Release Escrow.
 The Filing Office may, if the Termination Statement is filed, return the Digital Asset Collateral to the Debtor.

Solidity

```

uint digital_asset_collateral;
bool _default;
function Pay_Escrow_In() external payable {
    if (msg.sender == debtor){
        digital_asset_collateral=msg.value;
    }else{
        require(false);
    }
}

function Fail_to_Pay() external {
    if (msg.sender == secured_party){
        _default=true;
    }else{
        require(false);
    }
}

function Take_Possession() external {
    if (msg.sender == _filing_office){
        if(_default){
            secured_party.transfer(digital_asset_collateral);
        }
    }else{
        require(false);
    }
}

function Release_Escrow() external {
    if (msg.sender == _filing_office){
        if(termination_statement){
            debtor.transfer(digital_asset_collateral);
        }else{
            require(false);
        }
    }
}

```

As with the reminder fee escrow, adopting this element of the Lexon U.C.C. Financing Statement is completely optional, and failure to adopt the digital asset collateral escrow feature will not impinge upon the functionality of the core ability of the Lexon U.C.C. Financing Statement to perform existing law. By demonstrating the innovation, this Article proves out the ability of the technology itself to help law address new issues that the technology raises. In other words, the digital asset collateral escrow option is an example of regulatory technology applied to solve a new legal issue posed by the technology.²³⁵

5. Curbing Filing Abuses by Offering Debtors the Opportunity to Electronically Authorize Financing Statements

Along with the changes to the collateral description requirement for a sufficient financing statement, the 2001 revisions to U.C.C. Article 9 sought to make the filing process more amenable to electronic filing procedures by eliminating the

235. See Carla L. Reyes, *Autonomous Business Reality*, 21 NEV. L.J. 437, 488–89 (2021).

requirement that the debtor sign a financing statement.²³⁶ Instead, a financing statement need only be authorized by the debtor.²³⁷ Fraudulent filings occur regularly.²³⁸ For example, “[i]n Connecticut, a convicted drug dealer filed \$12 billion worth of liens against two federal judges and the lawyer who prosecuted him.”²³⁹ The filing office has no authority to reject such bogus filings because their role in the filing system is limited to confirming that the requisite fields of the financing statement have been completed.²⁴⁰ The filing office has no power to assess the veracity of the information the financing statement contains.²⁴¹ While it may not make sense to require a debtor’s signature for every financing statement,²⁴² the Lexon U.C.C. Financing Statement would enable rules that specifically targeted the practice of filing bogus financing statements related to public officials and other high-profile individuals.

Namely, each debtor in the Lexon U.C.C. Financing Statement is assigned a public key address. The filing office could maintain a politically important person list (a “PIP List”) that flagged the public key addresses of public officials and other high-profile individuals. When a filer attempts to file a financing statement related to one of those public key addresses, the filing would trigger on the PIP List and be held up for manual review. The filing office could request that the owner of the public key address review the attempted filing and confirm that it was

236. See U.C.C. §§ 9-101 cmt. 4(h), 9-502 cmt. 3 (AM. L. INST. & UNIF. L. COMM’N 2010); see also Moringiello, *supra* note 22, at 144 (describing the U.C.C. revisions to accommodate electronic filings).

237. U.C.C. § 9-509.

238. See Moringiello, *supra* note 22, at 138–41 (recounting instances of fraudulent filings).

239. *Id.* at 139.

240. See NAT’L ASS’N OF SECY’S OF STATE, STATE STRATEGIES TO SUBVERT FRAUDULENT UNIFORM COMMERCIAL CODE (UCC) FILINGS: A REPORT FOR STATE BUSINESS FILING AGENCIES 6 (2019), <https://perma.cc/GST8-4JVJ> (PDF) (highlighting that the filing office may only reject documents for failure to pay a fee, incompleteness, or ineligible writing).

241. See *id.* (“The [Secretary of State’s] office does not have the authority to verify the accuracy or the validity of documents.”).

242. See Moringiello, *supra* note 22, at 152–57 (arguing that a financing statement is akin to a notice, whereas a signature on the security agreement authorizes the debtor to file the financing statements).

authorized by using their private key to sign the filing and release it from manual review.

Beyond curbing filing abuses, the ability to easily enable debtors to electronically sign financing statements may also address other policy concerns. At least one writer proposed requiring a debtor's signature when a financing statement used a super-generic description of the collateral.²⁴³ The policy argument for doing so revolves around the fact that even though the description of the collateral in the financing statement does not magically extend the reach of the secured party's security interest to collateral not covered by the security agreement, it can curtail a debtor's ability to access capital in the future by confusing future filers and introducing unnecessary ambiguity.²⁴⁴ Requiring a debtor's signature would at least ensure that the debtor knew that a super-generic description would be used and offer the debtor an opportunity to inquire as to its potential impact.²⁴⁵

Along with a super-generic description manual review rule much like the PIP List proposed above, the Lexon U.C.C. Financing Statement enables a simple mechanism to implement such protective measures for the debtor. Like enabling new search mechanisms and enabling more specific identification of collateral, using a debtor's cryptographic signature to curb fraudulent filings represents a ripple effect resulting from transplanting the Article 9 provisions into code. Together, those three endogenous innovations, which exist because of the inherent nature of the Lexon U.C.C. Financing Statement, bring cryptolaw to the U.C.C. Article 9 filing system.

B. *The Lexon U.C.C. Financing Statement as a Development
in Computational Law and a Call for Lawyers to Expand Their
View of the Duty of Technological Competence*

Two parallel discussions exist among legal academics and legal practitioners: one debates the extent to which the future

243. See Xiang, *supra* note 216, at 1072 (“[W]hen the financing statement’s collateral description is generic or supergeneric, it would be better policy to require the debtor to authenticate the document before it is filed and becomes enforceable.”).

244. *Id.*

245. See *id.* at 1072–73.

requires an army of legal engineers—lawyers who can code—to help build and interact with computational law,²⁴⁶ and the other considers the scope and implications of the emerging professional duty of technological competence.²⁴⁷ The Lexon U.C.C. Financing Statement hints at the potential relationship between these two arenas. Computational law traditionally requires a high level of technological expertise to build, discuss, and understand.²⁴⁸ This relegation of computational law to dual experts—experts in law and code—leads to predictions of a new professional class of legal engineers, and the death of traditional law.²⁴⁹ The prediction of the rise of a new class of legal engineers rests in part on the intuition that coding law by translating law

246. See generally RICHARD SUSSKIND, *THE END OF LAWYERS?: RETHINKING THE NATURE OF LEGAL SERVICES* (2010); Philip Segal, *Legal Jobs in the Age of Artificial Intelligence: Moving From Today's Limited Universe of Data Toward the Great Beyond*, 5 SAVANNAH L. REV. 211 (2018); J.B. Ruhl & Daniel Martin Katz, *Measuring, Monitoring, and Managing Legal Complexity*, 101 IOWA L. REV. 191, 243 (2015); Benjamin Alarie et al., *How Artificial Intelligence Will Affect the Practice of Law*, 68 U. TORONTO L.J. 106 (2018).

247. See generally Brittany Stringfellow Otey, *Millennials, Technology, and Professional Responsibility: Training a New Generation in Technological Professionalism*, 37 J. LEGAL PRO. 199 (2013); Caitlin Moon, *Delta Model Lawyer: Lawyer Competencies for the Computational Age*, MIT COMPUTATIONAL L. REP., Dec. 6, 2019; Catherin J. Lanctot, *Becoming a Competent 21st Century Legal Ethics Professor: Everything You Always Wanted to Know About Technology (But Were Afraid to Ask)*, 2015 J. PRO. LAW. 75 (2015); Agnieszka McPeak, *Disruptive Technology and the Ethical Lawyer*, 50 TOL. L. REV. 457 (2019).

248. See Ruhl & Katz, *supra* note 246, at 232–37 (providing an example of the technological demands to build a multi-layered computational legal system that would react to the evolution of the legal system).

249. See, e.g., Kevin P. Lee, *The Citizen Lawyer in the Coming Era: Technology Is Changing the Practice of Law, but Legal Education Must Remain Committed to Humanistic Learning*, 40 OHIO N.U. L. REV. 1, 3–4 (2013) (examining whether legal education can keep up with technology); RICHARD SUSSKIND, *TOMORROW'S LAWYERS: AN INTRODUCTION TO YOUR FUTURE* 132 (2013) (addressing that successful lawyers in the future will need to be legal technologists and have a strong understanding of technology to provide innovative services); Daniel Martin Katz, *Quantitative Legal Prediction—Or—How I Learned to Stop Worrying and Start Preparing for the Data Driven Future of the Legal Services Industry*, 62 EMORY L.J. 909, 910–14 (2013) (observing that the advances in machine learning and automation will change the legal industry); PAUL CAMPOS, *DON'T GO TO LAW SCHOOL (UNLESS): A LAW PROFESSOR'S INSIDE GUIDE TO MAXIMIZING OPPORTUNITY AND MINIMIZING RISK 1* (2012) (addressing that the demand for new lawyers has decreased over time in part due to technological advances).

as understood by a legal expert into code designed by a software engineer may imbed unintended errors and side effects, including giving inadvertent power over legal processes to software developers with no legal training.²⁵⁰ The Lexon U.C.C. Financing Statement simultaneously challenges the narrative that lawyers must learn how to code while confirming the intuition that crafting computational law works best when undertaken directly by lawyers.

The Lexon U.C.C. Financing Statement, written in plain English that is also machine-readable code, challenges the narrative that computational law belongs only to legal engineers by democratizing the process of building software. Interestingly, the ability for a non-programmer to write Lexon was not a Lexon development goal.²⁵¹ Lexon is optimized for reading, so that anyone can audit the code and understand what it says.²⁵² That non-programmers also can succeed at writing programs with Lexon appears to be a happy side effect of optimizing the language for readability.²⁵³ At the same time, the process of building the Lexon U.C.C. Financing Statement demonstrated the need for a shift in the mental model of its lawyer and law professor designer. Many of the elements necessary for making the Lexon U.C.C. Financing Statement achieve its goals only became apparent through the drafting process.²⁵⁴ This confirms

250. Reyes, *Conceptualizing Cryptolaw*, *supra* note 8, at 436–37.

251. DIEDRICH, LEXON BIBLE, *supra* note 16, at 23.

252. *Id.* at 24.

253. *See id.*

254. Indeed, I first tried to build a prototype of this system by explaining what I thought was needed to a group of computer science students who had agreed to investigate how to build it as a research project. The project was lost in translation. I simply did not know how to explain the requirements and specifications of the system because although I had thought about it from a legal perspective, I did not fully appreciate the translation between that legal perspective and formal logic for the blockchain. It was not until writing iterations of the Lexon code that I slowly learned where I was going wrong. This, of course, is only anecdotal evidence that may be peculiar to me, or to this project, however, my experience fits neatly within pedagogical lessons about how people learn—that learning by doing often produces the best results. Despite the anecdotal nature of this evidence, the experience was transformative enough that I plan to launch a pedagogical tool to help law students and lawyers unpack emerging technologies by using emerging technologies. *See* Michael Jeffrey, *What Would an Integrated Development Environment for Law Look Like?*, MIT COMPUTATIONAL L. REP., Apr. 1. 2020

the intuition that translations between law and code by a lawyer-programmer development team may risk introducing error or misunderstanding. The lawyer knows the law, but in a professional sense of how to use it to obtain a good outcome for a client. The programmer understands the limits, architecture, and intricacies of code. Understanding how to bridge the two, at least in the case of the Lexon U.C.C. Financing Statement, required the lawyer trying her hand at programming, which illuminated many of the elements of the legal process related to U.C.C. Article 9 that lawyers may take for granted in practice. In this way, the Lexon U.C.C. Financing Statement, and the experience of creating it, contributes to the development of computational law by calling on the field of computational law to critically assess its own narratives against advances in the technology that make computational law available to a broader group of lawyers and legal academics.²⁵⁵ This potential for a broader range of players creating computational legal tools also raises issues related to a lawyer's duty of technological competence.²⁵⁶

Specifically, the Lexon U.C.C. Financing Statement highlights a potential collision between two important principles in the law: technology neutrality and a lawyer's duty of technological competence. Technology neutrality represents a central principle of lawmaking, particularly with regard to private law.²⁵⁷ When lawmaking takes a functional approach,

(highlighting coding opportunities and innovations in contract drafting and transactional legal practice).

255. For a review of a variety of such technologies, particularly in the area of computable contracts, see generally Ma, *supra* note 106.

256. See Harry Surden, *Ethics of AI in Law: Basic Questions*, in OXFORD HANDBOOK OF ETHICS OF AI 719, 735 (Markus D. Dubber et al. eds., 2020) (“[I]f the best and most accurate predictions are made by lawyers using AI systems, will the ethical or professional standards shift such that lawyers are obligated to use such systems . . . ?”).

257. See, e.g., Chris Reed, *Taking Sides on Technology Neutrality*, 4 SCRIPT-ED 263, 264 (2007) (“Technology neutrality has long been held up as a guiding principle for the proper regulation of technology, particularly the information and communications technologies.”); UNCITRAL MODEL LAW ON ELECTRONIC COMMERCE WITH GUIDE TO ENACTMENT 17 (United Nations 1996), <https://perma.cc/N4EZ-F289> (PDF)

The objectives of the Model law, which include enabling or facilitating the use of electronic commerce and providing equal treatment to users of paper-based documentation and to users of

the law can apply equally to an activity, regardless of the medium through which it is conducted.²⁵⁸ As a result of this emphasis on a functional approach to law, the legal profession tends to silo the areas for which lawyers must understand how technology works from areas where they do not.²⁵⁹ If a lawyer is not a patent prosecutor or product liability litigation specialist, the legal profession holds the lawyer only to an ethical obligation of technological competence in a very metered sense.²⁶⁰ Generally speaking, the duty of technological competence relates to understanding technology-based legal practice tools sufficiently to zealously represent a client's interests and protect client data and confidentiality.²⁶¹ But understanding the technology at a level deep enough to build law that can accommodate it, or to build legal structures with

computer-based information, are essential for fostering economy and efficiency in international trade. By incorporating the procedures prescribed in the Model Law in its national legislation for those situations where parties opt to use electronic means of communication, an enacting State would create a media-neutral environment.

Id. at 23–24 (“It was felt during the preparation of the Model Law that exclusion of any form or medium by way of a limitation in the scope of the Model Law might result in practical difficulties and would run counter to the purpose of providing truly ‘media-neutral’ rules.”); Bert-Jaap Koops, *Should ICT Regulation Be Technology-Neutral?*, in *STARTING POINTS FOR ICT REGULATION: DECONSTRUCTING PREVALENT POLICY ONE-LINERS* 77, 77 (Bert-Jaap Koops et al. eds., 2006).

258. For the classic defense of this approach, see Frank Easterbrook, *Cyberspace and the Law of the Horse*, 1996 U. CHI. LEGAL FORUM 207, 208–13 (1996).

259. The patent prosecution bar, for example, is expected to have a higher level of technical and scientific substantive expertise than is required for product lawyers. See John R. Allison & Mark A. Lemley, *The Growing Complexity of the United States Patent System*, 82 B.U. L. REV. 77 (2002); Jason J. Du Mont & Mark D. Janis, *Virtual Designs*, 17 STAN. TECH. L. REV. 107 (2014).

260. See Jamie J. Baker, *Beyond the Information Age: The Duty of Technology Competence in the Algorithmic Society*, 69 S.C. L. REV. 557, 557–58 (2018). In 2012, the ABA Model Rules of Professional Conduct were amended “to state that a lawyer’s duty of competence now also requires keeping ‘abreast of changes in the law and its practice, including the benefits and risks associated with relevant technology.’” *Id.* at 560.

261. See *id.* at 561 (“Given that this falls under the Duty of Competence, however, the foundation of technology competence means, in part, that lawyers are now ‘required to take reasonable steps to protect their clients from ill-conceived uses of technology.’”).

technology? That is thought to be the realm of legal engineers²⁶² and computational law,²⁶³ both of which are thought to be separate from the more traditional arena of law-making and legal practice. By taking a very traditional commercial law system, the U.C.C. Article 9 filing system, and demonstrating the potential of a regular lawyer and law professor (not a legal engineer) to build technology for it, this Article evidences the need to eradicate the silos between legal engineering inquiries and the broader duty of technological due competence.

Lawyers may not need to know how to code.²⁶⁴ Indeed, Lexon may obviate the need for formal training in software development by allowing the creation of code in plain English. However, lawyers should understand how technology works, right down to the very important but highly technical details. Otherwise, the legal profession and lawmakers will miss key opportunities to improve systems within the law that would benefit from the application of technology. Perhaps more worrisome, a lack of technology competence beyond the protection of client data and confidentiality may lead to the adoption of bad law.²⁶⁵ The Lexon U.C.C. Financing Statement makes plain one area in which misunderstanding of technology leads to poor policy priorities and sub-optimal law making: smart contracts. The Lexon U.C.C. Financing Statement is built

262. See Daniel W. Linna Jr., *The Future of Law and Computational Technologies: Two Sides of the Same Coin*, MIT COMPUTATIONAL L. REP., Dec. 06, 2019.

263. See Alex Pentland, *A Perspective on Legal Algorithms*, MIT COMPUTATIONAL L. REP., Dec. 06, 2019.

264. See Mark Fenwick et al., *Legal Education in the Blockchain Revolution*, 20 VAND. J. ENT. & TECH. L. 351, 355 (2017) (advocating that legal education should evolve to include classes on technological disruption and addressing how lawyers can team with technologists to solve legal and business problems); MARK FENWICK ET AL., LEGAL EDUCATION IN A DIGITAL AGE: WHY 'CODING FOR LAWYERS' MATTERS (2018), <https://perma.cc/4TNJ-VDVZ> (PDF) (promoting legal education that teaches coding because understanding coding concepts will be help lawyers be more successful). *But see* Basel Farag, *Please Don't Learn to Code*, TECHCRUNCH (May 10, 2016), <https://perma.cc/9DTD-SKEY> (arguing that most people should not become coders).

265. See, e.g., Angela Walch, *The Path of Blockchain Lexicon (and the Law)*, 36 REV. BANKING & FIN. L. 713, 731–33 (2017) (detailing how regulators' lack of knowledge about blockchain could result in "undesirable regulatory outcomes").

out of smart contracts. Yet, the Lexon U.C.C. Financing Statement is not, itself, a contract, nor would anyone say that a financing statement is intended as a legally enforceable contract. And yet, law has been eminently preoccupied with the application of contract law to smart contracts.²⁶⁶ As a result, states have adopted laws to “accommodate” smart contracts.²⁶⁷ In misunderstanding smart contracts and their primary function, however, most such state laws actually carve smart contracts out from the beneficial statutory coverage they previously enjoyed.²⁶⁸ The Lexon U.C.C. Financing Statement and the process of building it stands as a call for greater exploration at the intersection of law, technology, language, and advocacy. Indeed, it stands as a call to expand the notion of a lawyer’s technological competence beyond mere competent use of legal technology tools for improving efficiency while protecting client data.

CONCLUSION

The junior associate and paralegal from the SuperBank and Automaker deal could have avoided a catastrophic loss if the Delaware filing office employed the Lexon U.C.C. Financing Statement at the time. When the search by Automaker’s name revealed three financing statements, the junior associate could have run separate searches by the collateral, secured party name, or file number and compared the results to reveal that only two of the three financing statements related to the repaid debt. Such redundant search capability would also have helped the bankruptcy trustee that could not find filings because of a

266. See *supra* note 94 and accompanying text; Giesela Rühl, *Smart (Legal) Contracts, or: Which (Contract) Law for Smart Contracts?*, in BLOCKCHAIN, LAW AND GOVERNANCE 159, 160 (Benedetta Cappiello & Gherardo Carullo eds., 2018) (“[S]mart contracts need contract law . . .”); Max Raskin, *The Law and Legality of Smart Contracts*, 1 GEO. L. TECH. REV. 305, 306 (2017) (“[T]here is little difficulty situating smart contracts within existing contract law.”).

267. See Reyes, *A Unified Theory*, *supra* note 84, at 999 n.88 (surveying the downsides of trying to accommodate smart contracts within state contract laws).

268. See *id.* (documenting how state laws that attempt to incorporate smart contracts into the Uniform Electronic Transactions Act (UETA) actually end up “excluding smart contracts from UETA’s coverage rather than ‘accommodating’ smart contracts”).

misspelled debtor name. Ultimately, then, in proving out the previously theoretical possibility of crypto-legal structures by creating cryptolaw for the U.C.C., this Article demonstrated the power of fully transforming the law on the books into the law in action literally and simultaneously.

Although counterintuitive at first, the Lexon U.C.C. Financing Statement bears out the promise of computation law: applying complex computational techniques to law does not necessarily make law more complex, but rather can simplify law and make it more accessible. But the Lexon U.C.C. Financing Statement goes beyond that by applying the same maxim to computational law itself. The Lexon U.C.C. Financing Statement brings simplicity and accessibility to computational law—code anyone can read, and even a law professor can write.²⁶⁹ It is this simplicity and accessibility that makes the Lexon U.C.C. Financing Statement so powerful. The Lexon U.C.C. Financing Statement simultaneously summarizes Article 9 provisions in English and performs them in machine-readable and executable code.²⁷⁰ The law on the books literally becomes the law in action. Further, the Lexon U.C.C. Financing Statement makes this leap not in the context of private law such as contract, but in the form of a publicly administered system important to the proper functioning of the secured lending market.²⁷¹ In doing so, the Lexon U.C.C. Financing Statement offers a first glimpse into a transparent approach to computational law that can alleviate some of the concerns related to computational opacity and explainability that otherwise permeate discussions at the intersection of computational and public law.²⁷²

In particular, the Lexon U.C.C. Financing Statement addresses concerns that the computational complexity required to formalize legal logic in code will shift the locus of legal culture from lawyers to software developers.²⁷³ A lawyer and law

269. *See supra* Part II.A.

270. *See supra* Part II.A.

271. *See supra* Part II.B.

272. For discussions of these concerns, see generally Joshua A. Kroll et al., *Accountable Algorithms*, 165 U. PA. L. REV. 633 (2017); Hannah Bloch-Wehba, *Access to Algorithms*, 88 FORDHAM L. REV. 1265 (2020); Andrew D. Selbst, *The Intuitive Appeal of Explainable Machines*, 87 FORDHAM L. REV. 1085 (2018).

273. *See supra* Part III.B.

professor, with zero formal programming training, built the Lexon U.C.C. Financing Statement by writing code in plain English. The drafting process offered opportunities to solve known practical difficulties that plague the system, hold parties accountable to the provisions and intent of the U.C.C., and maintain the necessary balance between creditor and debtor rights. While these steps to bring cryptolaw to the U.C.C. are themselves powerful, the Lexon U.C.C. Financing Statement also stands as a call for further experimentation and research. What other areas of the law are ripe for the use of a transparent, accessible, and productive form of computational law? How can lawyers leverage their duty of technological competence beyond individual client representation to help the law achieve its intended function? These and related questions signal that, while powerful, the Lexon U.C.C. Financing Statement represents just the beginning of a potential new age of computation and the law.

APPENDIX A: LEXON U.C.C. FINANCING STATEMENT USING LEX
0.2²⁷⁴

LEX UCC Financing Statement.

LEXON: 0.2.12

"Financing Statement" is this contract.

"File Number" is data.

"Initial Statement Date" is a time.

"Filer" is a person.

"Debtor" is a person.

"Secured Party" is a person.

"Filing Office" is a person.

"Collateral" is data.

"Digital Asset Collateral" is an amount.

"Reminder Fee" is an amount.

"Continuation Window Start" is a time.

"Continuation Statement Date" is a time.

"Continuation Statement Filing Number" is data.

"Lapse Date" is a time.

"Default" is a binary.

"Continuation Statement" is a binary.

"Termination Statement" is a binary.

"Termination Statement Time" is a time.

"Notification Statement" is a text.

The Filer fixes the Filing Office, fixes the Debtor, fixes the Secured Party, and fixes the Collateral.

Clause: Certify.

The Filing Office may certify the File Number.

Clause: Set File Date.

The Filing Office may fix the Initial Statement Date as the current time.

Clause: Set Lapse.

274. A note on the code. This code is not perfect as it does not yet handle fringe cases. For example, it does not explicitly formalize processes for assignment or information statements. In the interest of brevity, this fine tuning has been relegated. As Lexon develops and the Lexon U.C.C. Financing Statement code continues to be refined, updates will be posted to <http://lexon.digital/reyes.html>, along with additional technical details and instructions for those who may wish to implement the code.

The Filing Office may fix the Lapse Date.

Clause: Set Continuation Start.

The Filing Office may fix the Continuation Window Start.

Clause: Pay Fee.

The Secured Party may pay a Reminder Fee into escrow.

Clause: Notice.

The Filing Office fixes the Notification Statement.

Clause: Notify.

The Filing Office may, if the Continuation Window Start has passed, send the Notification Statement to the Secured Party.

Clause: Pay Escrow In.

The Debtor may pay the Digital Asset Collateral into escrow.

Clause: Fail to Pay.

The Secured Party may declare Default.

Clause: Take Possession.

The Filing Office may, if Default is declared, pay the Digital Asset Collateral to the Secured Party.

Clause: File Continuation.

The Secured Party may file the Continuation Statement.

Clause: Set Continuation Lapse.

The Filing Office may, if the Continuation Statement is filed, fix the Continuation Statement Date.

Clause: File Termination.

The Secured Party may file a Termination Statement, and certify the Termination Statement Time as the then current time.

Clause: Release Escrow.

The Filing Office may, if the Termination Statement is filed, return the Digital Asset Collateral to the Debtor.

Clause: Release Reminder Fee.

The Filing Office may, if the Termination Statement is filed, return the Reminder Fee to the Secured Party.

Clause: Termination Period.

"Termination Period" is defined as 365 days after the Termination Statement Time.

Clause: Terminate and Clear.

The Filing Office may, if the Termination Period has passed, terminate this contract.

APPENDIX B: SOLIDITY COMPILED FROM LEXON U.C.C.
FINANCING STATEMENT²⁷⁵

```
pragma solidity ^0.5.0;

contract UCC_Financing_Statement{
    bytes32 file_number;
    uint initial_statement_date;
    address payable filer;
    address payable debtor;
    address payable secured_party;
    address payable filing_office;
    bytes32 collateral;
    uint digital_asset_collateral;
    uint reminder_fee;
    uint continuation_window_start;
    uint continuation_statement_date;
    bytes32 continuation_statement_filing_number;
    uint lapse_date;
    bool _default;
    bool continuation_statement;
    bool termination_statement;
    uint termination_statement_time;
    string notification_statement;

    event Notification(address receiver, string
notification_statement);

    constructor(address payable _filing_office, address
payable _debtor, address payable _secured_party,
bytes32 _collateral) public payable {
        filer=msg.sender;
        filing_office=_filing_office;
        debtor=_debtor;
        secured_party=_secured_party;
        collateral=_collateral;
    }
}
```

275. Further notes on the code and imperfections. Lexon is progressing to version 0.3, which addresses imperfections of version 0.2. Among the improvements are automatisms to better catch human errors like double-sending of fees (Pay_Fee()) and the undefined state that would occur if the Filing Office failed to execute Certify(), Set_File_Date() and Set_Lapse() as the first action after the filing.

```
}

function Certify(bytes32 _file_number) external {
    if (msg.sender == _filing_office){
        file_number=_file_number;
    }else{
        require(false);
    }
}

function Set_File_Date() external {
    if (msg.sender == _filing_office){
        initial_statement_date=now;
    }else{
        require(false);
    }
}

function Set_Lapse(uint _lapse_date) external {
    if (msg.sender == _filing_office){
        lapse_date=_lapse_date;
    }else{
        require(false);
    }
}

function Set_Continuation_Start(uint
_continuation_window_start) external {
    if (msg.sender == _filing_office){
        continuation_window_start=_continuation_window_start;
    }else{
        require(false);
    }
}

function Pay_Fee() external payable {
    if (msg.sender == secured_party){
        reminder_fee=msg.value;
    }else{
        require(false);
    }
}
```

```
    }

    function Notice(string memory
_notification_statement) private {
        notification_statement=_notification_statement;
    }

    function Notify() external {
        if (msg.sender == _filing_office){
            if(continuation_window_start<=now){
                emit Notification(secured_party,
notification_statement);
            }
        }else{
            require(false);
        }
    }

    function Pay_Escrow_In() external payable {
        if (msg.sender == debtor){
            digital_asset_collateral=msg.value;
        }else{
            require(false);
        }
    }

    function Fail_to_Pay() external {
        if (msg.sender == secured_party){
            _default=true;
        }else{
            require(false);
        }
    }

    function Take_Possession() external {
        if (msg.sender == _filing_office){
            if(_default){
                secured_party.transfer(digital_asset_collateral);
            }
        }else{
```

```
        require(false);
    }
}

function File_Continuation() external {
    if (msg.sender == secured_party){
        continuation_statement=true;
    }else{
        require(false);
    }
}

function Set_Continuation_Lapse(uint
_continuation_statement_date) external {
    if (msg.sender == _filing_office){
        if(continuation_statement){
            continuation_statement_date=_continuation_statement_date;
        }
    }else{
        require(false);
    }
}

function File_Termination() external {
    if (msg.sender == secured_party){
        termination_statement=true;
        termination_statement_time=now;
    }else{
        require(false);
    }
}

function Release_Reminder_Fee() external {
    if (msg.sender == _filing_office){
        if(termination_statement){
        }
        secured_party.transfer(reminder_fee);
    }else{
        require(false);
    }
}
```

```
    }

    function Release_Escrow() external {
        if (msg.sender == _filing_office){
            if(termination_statement){
            }
            debtor.transfer(digital_asset_collateral);
        }else{
            require(false);
        }
    }

    function Termination_Period() public view
    returns(uint) {
        return termination_statement_time + 365 * 24 *
60 * 60;
    }

    function Terminate_and_Clear() external {
        if (msg.sender == _filing_office){
            if(Termination_Period() <= now){
                selfdestruct(debtor);
            }
        }else{
            require(false);
        }
    }
}
```


APPENDIX C: JAVASCRIPT COMPILED FROM LEXON U.C.C.
FINANCING STATEMENT²⁷⁶

```
/* Lexon-generated Javascript
```

```
code:      UCC Financing Statement
```

```
file:      statement.lex
```

```
code tagged:  0.2.12
```

```
compiler:    lexon 0.3 alpha 60
```

```
grammar:     0.2.18.59 / subset 0.3.7 alpha 59 - English /  
Reyes
```

```
backend:     javascript 0.3.60
```

```
target:      node 14.1+
```

```
parameters:  --javascript --all-auxiliaries
```

INSTRUCTIONS FOR USE:

Execute this program using node. Replace the <parameters> with literal values.

Running this program as-is requires beginners programmer knowledge. This phase is yet not covered by lexon's mission to make code readable and useful for non-coders.

In the future, an interface will be generated to complete this last mile. However, embedding this code into a self-explanatory user interface is a straight forward task for a full-stack programmer.

276. This code and related instructions are available at <http://lexon.digital/reyes.html>.

Note that the instructions below reflect your lexon code as well as the parameters used during compilation of the code: different functions and parameters will result from different input. Some functions are 'built-in' but only appear when needed as per compiled-in features – a list of which is available with `lexon -h`. The functions are not given in a specific order of execution but as listed in the lexon source.

These node modules have to be installed once:

```
$ npm install serialize-javascript
$ npm install tar
$ npm install nodemailer
$ npm install prompt-sync
```

Parameters below are marked with double angle brackets <<>> for the respective required caller. If the role is defined earlier, it can only be performed by this person. (But remember that this entire setup is trustful: anyone can manipulate anything about this contract. Though they cannot sign it or change the signed log.) If the role is not defined earlier, the call makes the role be assigned to the person named for the call. Some functions can be called without naming a caller. Some clauses of the original lexon source will not appear below. Namely, those that have no permission phrase, wherefore they are regarded as internal.

The main contract system is initialized by loading the module and instantiating:

```

$ node
> contract = require("./statement.jsx");
> statement = new contract(<<filer>>, <filing_office>,
<debtor>, <secured_party>, <collateral>);

```

Remember to reset node's module cache each time you edit and recompile your code:

```
> delete require.cache[require.resolve('./statement.jsx')];
```

These are the state progress functions that allow to interact with the contract:

```

> statement.certify(<<filing_office>>, <file_number>)
> statement.set_file_date(<<filing_office>>)
> statement.set_lapse(<<filing_office>>, <lapse_date>)
> statement.set_continuation_start(<<filing_office>>,
<continuation_window_start>)
> statement.pay_fee(<<secured_party>>, <reminder_fee>)
> statement.notice(<<filing_office>>,
<notification_statement>)
> statement.notify(<<filing_office>>)
> statement.pay_escrow_in(<<debtor>>,
<digital_asset_collateral>)
> statement.fail_to_pay(<<secured_party>>)
> statement.take_possession(<<filing_office>>)
> statement.file_continuation(<<secured_party>>,
<continuation_statement>)
> statement.set_continuation_lapse(<<filing_office>>,
<continuation_statement_date>)
> statement.file_termination(<<secured_party>>,
<termination_statement>)
> statement.release_escrow(<<filing_office>>)
> statement.release_reminder_fee(<<filing_office>>)
> statement.termination_period()
> statement.terminate_and_clear(<<filing_office>>)

```

state changes of the contract can be listed, e.g. actions performed by

a party to it, or agents who are assigned privileges. In case hash chains or signatures are used, they are visible in this log. The log is stored in the file 'log'.

```
> statement.history()
```

The complete contract state can be saved to disk and reloaded at a later point in time. This serves to continue work after stopping and restarting node; or to send the entire contract system and its current state - which can include hashes and signatures - to another party, who may perform the next steps.

```
> statement.persist()
> statement.load()
```

The contract code, state and log can be bundled into one file to exchange or archive it:

```
> statement.bundle()
> statement.unbundle()
```

The contract code, state and log can be sent to a counterparty. This requires configuring an email account in the file 'config'.

```
> statement.send()
```

Keys for signing log entries are expected on-file, by default named after the actor, with the extension .key. For demo purposes, key files can be created using this utility function:

```
> system.create_key(name, passphrase)
*/

var fs = require('fs');
var crypto = require('crypto');
var serialize = require('serialize-javascript');
var prompt = require('prompt-sync')();
var tar = require('tar');
var nodemailer = require('nodemailer');
var last_caller;
var last_passphrase;

/**
**
** Main UCC Financing Statement contract system
**
**/

module.exports = class UCCFinancingStatement {

  constructor(filer, filing_office, debtor, secured_party,
collateral) {

    /* object members: skip for restoring serialized object */
    if(typeof filer !== 'undefined') {
      this.financing_statement = null;
      this.file_number = null;
      this.initial_statement_date = null;
      this.filer = filer;
      this.debtor = debtor;
      this.secured_party = secured_party;
      this.filing_office = filing_office;
      this.collateral = collateral;
      this.digital_asset_collateral = null;
      this.reminder_fee = null;
      this.continuation_window_start = null;
      this.continuation_statement_date = null;
    }
  }
}
```

```

    this.continuation_statement_filing_number = null;
    this.lapse_date = null;
    this.default_ = null;
    this.continuation_statement = null;
    this.termination_statement = null;
    this.termination_statement_time = null;
    this.notification_statement = null;
    this.logname = 'log';

    /* start log - overwrites previous by same name */
    fs.writeFileSync(this.logname, "Lexon log " + (new
Date).toLocaleString('en-US') + "\n", 0=>{});
    this.log(filer, "✓ Filing Office fixed");
    this.log(filer, "✓ Debtor fixed");
    this.log(filer, "✓ Secured Party fixed");
    this.log(filer, "✓ Collateral fixed");
  }

  /* restore object from file (must be below class definition)
*/
  if(typeof filer === 'undefined') {
    console.log("> restore from file 'state'");
    var data = fs.readFileSync('state', 0=>{});
    var live = eval('(' + data + ')');
    Object.assign(this, live);
  }
}

/* Certify clause */
certify.caller, file_number) {
  if(caller == this.filing_office) {
    this.file_number = file_number;
    this.log(caller, "✓ File Number certified");
  } else {
    return 'not permitted.';
  }
}
return 'done.';

```

```
}

/* Set File Date clause */
set_file_date(caller) {
  if(caller == this.filing_office) {
    this.initial_statement_date = Date.now();
    this.log(caller, "✓ Initial Statement Date fixed");
  } else {
    return 'not permitted.';
  }
  return 'done.';
}

/* Set Lapse clause */
set_lapse(caller, lapse_date) {
  if(caller == this.filing_office) {
    this.lapse_date = lapse_date;
    this.log(caller, "✓ Lapse Date fixed");
  } else {
    return 'not permitted.';
  }
  return 'done.';
}

/* Set Continuation Start clause */
set_continuation_start(caller, continuation_window_start) {
  if(caller == this.filing_office) {
    this.continuation_window_start =
continuation_window_start;
    this.log(caller, "✓ Continuation Window Start fixed");
  } else {
    return 'not permitted.';
  }
  return 'done.';
}

/* Pay Fee clause */
pay_fee(caller, reminder_fee) {
  if(caller == this.secured_party) {
```

```
        this.reminder_fee = reminder_fee;
        this._pay.caller, this.secured_party, 'escrow',
reminder_fee);
    } else {
        return 'not permitted.';
    }
    return 'done.';
}

/* Notice clause */
notice.caller, notification_statement) {
    if.caller == this.filing_office) {
        this.notification_statement = notification_statement;
        this.log.caller, "✓ Notification Statement fixed");
    } else {
        return 'not permitted.';
    }
    return 'done.';
}

/* Notify clause */
notify.caller) {
    if.caller == this.filing_office) {
        if(this.continuation_window_start <= Date.now()) {
            this_send.caller, this.filing_office, this.secured_party,
this.notification_statement);
        }
    } else {
        return 'not permitted.';
    }
    return 'done.';
}

/* Pay Escrow In clause */
pay_escrow_in.caller, digital_asset_collateral) {
    if.caller == this.debtor) {
        this.digital_asset_collateral = digital_asset_collateral;
        this._pay.caller, this.debtor, 'escrow',
digital_asset_collateral);
    }
}
```



```
    } else {
      return 'not permitted.';
    }
    return 'done.';
  }

  /* Fail to Pay clause */
  fail_to_pay(caller) {
    if(caller == this.secured_party) {
      this.default_ = true;
      this.log(caller, "✓ Default declared");
    } else {
      return 'not permitted.';
    }
    return 'done.';
  }

  /* Take Possession clause */
  take_possession(caller) {
    if(caller == this.filing_office) {
      if((this.default_ != null)) {
        this._pay(caller, this.filing_office, this.secured_party,
this.digital_asset_collateral);
      }
    } else {
      return 'not permitted.';
    }
    return 'done.';
  }

  /* File Continuation clause */
  file_continuation(caller, continuation_statement) {
    if(caller == this.secured_party) {
      this.continuation_statement = continuation_statement;
      this.log(caller, "✓ Continuation Statement filed");
    } else {
      return 'not permitted.';
    }
    return 'done.';
  }
}
```

```
    }

    /* Set Continuation Lapse clause */
    set_continuation_lapse(caller, continuation_statement_date)
    {
        if(caller == this.filing_office) {
            if((this.continuation_statement != null)) {
                this.continuation_statement_date =
continuation_statement_date;
                this.log(caller, "✓ Continuation Statement Date
fixed");
            }
        } else {
            return 'not permitted.';
        }
        return 'done.';
    }

    /* File Termination clause */
    file_termination(caller, termination_statement) {
        if(caller == this.secured_party) {
            this.termination_statement = termination_statement;
            this.log(caller, "✓ Termination Statement filed");
            this.termination_statement_time = Date.now();
            this.log(caller, "✓ Termination Statement Time
certified");
        } else {
            return 'not permitted.';
        }
        return 'done.';
    }

    /* Release Escrow clause */
    release_escrow(caller) {
        if(caller == this.filing_office) {
            if((this.termination_statement != null)) {
                this._pay(caller, this.filing_office, this.debtor,
this.digital_asset_collateral);
            }
        }
    }
}
```

```
    } else {
      return 'not permitted.';
    }
    return 'done.';
  }

  /* Release Reminder Fee clause */
  release_reminder_fee(caller) {
    if(caller == this.filing_office) {
      if((this.termination_statement != null)) {
        this._pay(caller, this.filing_office, this.secured_party,
this.reminder_fee);
      }
    } else {
      return 'not permitted.';
    }
    return 'done.';
  }

  /* Termination Period clause */
  termination_period() {
    return (this.termination_statement_time + (365 *
86400000));
  }

  /* Terminate and Clear clause */
  terminate_and_clear(caller) {
    if(caller == this.filing_office) {
      if(this.termination_period() <= Date.now()) {
        this.termination(caller);
      }
    } else {
      return 'not permitted.';
    }
    return 'done.';
  }

  /* built-in convenience function to view state change log. */
  history() {
```

```
        fs.readFile(this.logname,
(e,d)=>{console.log(d.toString());
    }

    /* built-in serialization and storage of entire contract system
state. */
    persist() {
        console.log('> persisting');
        var data = serialize(this, {space: 4});
        fs.writeFileSync('state', data, ()=>{});
    }

    /* re-instate entire contract system from serialized file store
*/
    static load() {
        return new UCCFinancingStatement();
    }

    /* built-in tar-balling of code, log and state. */
    bundle() {
        console.log('> bundling into contract.tgz');
        tar.create({gzip:true, file:'contract.tgz'}, ['statement.lex',
'statement.jsx', 'state', 'log', 'INSTRUCTIONS.TXT']);
    }

    /* built-in untar-balling of code, log and state. */
    static unbundle() {
        console.log('> unbundling contract.tgz');
        tar.extract('contract.tgz');
    }

    /* built-in email sending of code, log and state. */
    send() {

        this.persist();
        this.bundle();

        console.log('> sending via email');
        var receiver = prompt('enter receiver address: ');
```

```
var config = fs.readFileSync('config', 0=>{});
var email = eval('(' + config + ')').email;
console.log(email);

var transporter = nodemailer.createTransport({
  service: email.service,
  auth: { user: email.user, pass: email.pass }});

var mailOptions = {
  from: email.from,
  to: receiver,
  subject: email.subject,
  text: email.text,
  attachments: { path: './contract.tgz', contentType:
'application/gzip' }};

  transporter.sendMail(mailOptions, function(error, info){
    if (error) {
      console.log(error);
    } else {
      console.log('> email sent: ' + info.response);
    }
  });

/* built-in termination of the entire contract system */
termination(caller) {
  this.terminated = true;
  this.log(caller, '■ contract system terminated');
}

already_terminated() {
  if(!this.terminated) return false;
  console.log('× contract system previously terminated');
  return true;
}

/* built-in logging of state changes. */
log(caller, msg) {
  console.log(msg);
}
```

```

    let stamp = (new Date()).toLocaleString('en-US');
    var entry = `⊕ ${stamp} ◆ ${caller} ${msg}`;
    var passphrase = this.sync_passphrase(caller);
    var pem = fs.readFileSync(caller + '.key');
    var key = pem.toString('ascii');
    var sign = crypto.createSign('RSA-SHA256');
    sign.update(entry);
    var sig = sign.sign({ key: key, passphrase: passphrase },
'hex');
    fs.appendFileSync(this.logname, `${entry} * ${sig}\n`);
    let pay = fs.readFileSync(this.logname);
    let hash = crypto.createHash('sha256').update(pay);
    fs.appendFileSync(this.logname, '☐ ' +
hash.digest('hex').substr(0, 12) + " ");
  }

  /* built-in password query for private key file, with cache. */
  sync_passphrase(caller) {
    if(!caller) process.exit('no caller information');
    if(caller == last_caller) return last_passphrase;
    last_caller = caller;
    return last_passphrase = prompt('enter pass phrase for ' +
caller + ': ', {echo: ""});
  }

  /* built-in convenience function to create keys for users. */
  static create_key(name, passphrase) {
    const { publicKey, privateKey } =
      crypto.generateKeyPairSync('rsa',
        { modulusLength: 2048,
          publicKeyEncoding: { type: 'spki', format: 'pem' },
          privateKeyEncoding: { type: 'pkcs8', format: 'pem',
cipher: 'aes-256-cbc', passphrase: passphrase }});

    fs.writeFileSync(name+'.key', privateKey);
    fs.writeFileSync(name+'.pub', publicKey);
    return true;
  }

```

```
/* built-in pay message */
_pay(caller, from, to, amount) {
  this.log(caller, `➡ system message: pay ${amount} from
${from} to ${to}.`);
}

/* built-in send message */
_send(caller, from, to, message) {
  this.log(caller, `➡ system message: send message
«${message}» from ${from} to ${to}.`);
}
}

/* end */
```

APPENDIX D: LEXON U.C.C. FINANCING STATEMENT USING
LEXON 0.3

LEX UCC Financing Statement.

TERMS per UCC Financing Statement:

"File Number" is a hash.

"Filer" means a person by the name [name], with phone number of [phone number] and email of [email].

"Debtor" means a person by the name [name], identified by [Public Key], located at [mailing address, city, state, postal code, country].

"Secured Party" means a person by the name [name], identified by [Public Key], located at [mailing address, city, state, postal code, country].

"The Filing Office" means a person by the name [name], identified by [Public Key].

"Collateral" means a [UCC Category of Collateral], identified by the number: [hash].

"UCC Category of Collateral" means one of "goods", "equipment", "farm products", "consumer goods", "inventory", "negotiable instruments", "investment property", "accounts", "deposit accounts", "general intangibles", "payment intangibles", "chattel paper", "electronic chattel paper", "fixtures", "proceeds".

"Digital Asset Collateral" means [an amount].

"Financing Statement Date" means the filing date of the Initial Financing Statement.

"Continuation Statement Date" is a date.

"Amended Statement Filing Number" is a number.

The "Financing Statement" means the Initial Financing Statement and the List of Amendments.

"Initial Financing Statement" means the Financing Statement known by the Filing Number.

"List of Amendments" means all UCC Financing Statements that have the File Number of this UCC

Financing Statement set as their Amended Statement Filing Number.

"This Statement Amends an Initial Financing Statement" means that the Amended Statement Filing Number is set.

"Amended Statement" means the Financing Statement known by the Amended Statement Filing Number.

"Default" is a binary.

RECITALS:

The Filer sets the Debtor, the Secured Party, the Collateral and the Digital Asset Collateral.

The Filer sets the Amendment Filing Number, or not.

If the Amendment Number is set, the Filing Number is the Amendment Number; else, the Filing Number is the hash of the file.

CLAUSE: Lapse Date.

The "Lapse Date" means five years after the Financing Statement Date or five years after the Continuation Statement Date, whichever is later.

CLAUSE: Continuation Window.

"Continuation Window" means from six months before the Lapse Date to the Lapse Date.

CLAUSE: Payment of Reminder Fee.

The Secured Party may pay a Reminder Fee into escrow.

CLAUSE: Notification.

The Filing Office may, at the first day of the Continuation Window, send a Notification Statement to the Secured Party and then pay the Reminder Fee to themselves.

CLAUSE: Notification Statement.

"Notification Statement" means the text "Your Continuation Statement for [the Filing Number of the Initial Financing Statement] is due on or before the [Lapse Date]."

CLAUSE: Continuation Statement.

The Secured Party may, during the Continuation

Window, certify the Continuation Statement Date to be the current date.

CLAUSE: Termination Statement.

The Secured Party may terminate this Financing Statement, and return the Digital Asset Collateral from escrow to the Debtor, and return the remainder of the escrow to themselves.

CLAUSE: Clear.

The Filing Office may terminate this Financing Statement one year after the Lapse Dated has passed.

CLAUSE: Pay Escrow In.

The Debtor pays Digital Asset Collateral into escrow.

CLAUSE: Declaration of Default.

The Secured Party may declare Default.

CLAUSE: Take Possession.

The Filing Office may, upon Default, pay the escrow to the Secured Party.

CLAUSE: Release Escrow.

The Filing Office may, upon termination of the Initial Financing Statement, return the Digital Asset Collateral from escrow to the Debtor, and return the remainder of the escrow to the Secured Party.