UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

THE UNITED STATES OF AMERICA,

Plaintiff,

-against-

ROMAN STORM, ET AL.,

Defendants.

Case No.: 23 Cr. 430 (KPF)

The Honorable Katherine Polk Failla

BRIEF OF THE DEFI EDUCATION FUND AS *AMICUS CURIAE* IN SUPPORT OF DEFENDANT ROMAN STORM'S MOTION TO DISMISS THE INDICTMENT

Amanda Tuminelli, Esq.
Jacob Chervinsky, Esq. (*pro hac vice motion pending*)
DEFI EDUCATION FUND
1155 F St. NW
Suite 300
Washington, DC 20004

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

Page(s)

Cases

Statutes and Regulations

Other Authorities

INTEREST OF *AMICUS CURIAE*[1]

The DeFi Education Fund ("DEF") is a nonpartisan research and advocacy group based in the United States. DEF's mission is to explain the benefits of decentralized finance ("DeFi"), help achieve regulatory clarity for DeFi technology, contribute to the realization of the transformative potential of DeFi, and protect the rights of all digital asset market participants, including users and developers.

DEF has a significant interest in this case, particularly regarding how the government's allegations and theories of liability affect the interests and rights of software developers. DEF also has an interest in educating courts about the nature of the digital asset industry, public blockchain technology generally, and DeFi technology specifically.

DEF takes no position on the guilt or innocence of Roman Storm. However, DEF strongly urges the Court to reject the government's novel and inconsistent theories of criminal liability for software developers set forth in the Indictment, which far exceed existing law and radically diverge from decades of its own criminal enforcement precedent.

[1] No person except *amicus curiae* authored this brief or contributed money to fund its preparation or submission.

INTRODUCTION

Since the Industrial Revolution, rapid advancements in technology have wrought extraordinary changes in the developed world. At first, those advancements largely came in the space of "atoms"—engineers designed and built transformative inventions like automobiles, skyscrapers, televisions, and many others that now seem commonplace. In recent years, however, technological advancement has shifted toward the space of "bits"—that is, software developers design and build computer programs that power nearly everything around us, from the smartphones and laptops that we use every day to the most critical infrastructure supporting health and human welfare. Software developers have become an unseen but enormous force for maintaining and shaping modern society.

The Indictment in this case presents a remarkable question to the Court: when should software developers be held criminally liable for the bad acts of third parties who misuse software that they created? In the space of atoms, questions like this are rare. In general, automobile manufacturers are not liable for drivers who use their vehicles as weapons; construction companies are not liable for businesses that use their office buildings to perpetrate fraud; and television manufacturers are not liable for newscasters who use their screens to make defamatory statements. But according to the Indictment, the same logic does not apply in the space of bits. Instead, the Indictment posits that software developers should have boundless liability for third-party misuse of their inventions, subject to no limiting principle whatsoever.

The Indictment sets forth three theories of criminal liability for software developers that are historically unprecedented and legally unsupported. *First*, the Indictment asserts that developers are liable for violating the International Emergency Economic Powers Act ("IEEPA"), 50 U.S.C. § 1701 et seq., when they publish open-source software that is later used by a sanctioned entity, even if there is no allegation that they engaged with that sanctioned entity directly. *Second*, the Indictment asserts that developers are liable for conspiracy to commit money laundering under 18 U.S.C. §§ 1956(a), (h), when they publish open-source software that is later used by a third party to conduct transactions concealing the proceeds of specified unlawful activity, even if the developers did not know about or participate in those transactions. *Third*, the Indictment asserts that developers are liable for conspiracy to operate an unregistered money transmitting business under 18 U.S.C.A. § 371 and §1960 when they publish open-source software that enables users to engage in peer-to-peer financial transactions, even if they have no ability to change the software and no control over user funds.[2]

These theories of liability—if validated by the Court—would grant the government unlimited power to prosecute any software developer who writes code that is later used by a third party for nefarious purposes, merely because the developer becomes aware of that later use. With no limiting principle in place, nearly all developers who create open-source software would be exposed to criminal liability for activity outside of their control years or decades later. The surface area for selective prosecution would be incalculable, as the government would be free to target software developers aligned with politically disfavored causes and industries, who would have little in the way of defense or recourse. Put simply, validating the Indictment's theories of liability would mean rejecting core principles of due process and the rule of law.

Although blockchain technology plays an important role in this case, the government's theories of liability have implications for every software developer in every industry. Writing computer code is both a technical skill and an act of creative expression, and software developers

[2] Although we do not address Count II, we concur with the analysis in the Memorandum of Law in Support of Roman Storm's Motion to Dismiss, *see* ECF No. 37-1, and the other arguments made by *amici curiae* Blockchain Association and Coin Center.

are as much artists as they are engineers. Code is their language—the means by which they invent, innovate, and instantiate their ideas into the world. A ruling that criminalized the development of software that is later used by third party-bad actors would expose nearly one hundred million developers around the world to liability. *See Octoverse 2022: 10 years of tracking open source*, Github Blog (Nov. 17, 2022), https://tinyurl.com/2p9rehx7. The chilling effect of that decision may spell doom for the open-source software movement.

There is no question that the subject matter of the Indictment is of the utmost importance to U.S. national security, and it is critical that our law enforcement agencies and officials have the resources and authority that they need to root out bad actors. But here, the bad actors are the North Korean cybercriminals engaged in recurring transnational theft. The responsibility for criminal use of open-source software should lie with those who intentionally use it for illicit purposes. We urge the Court not to compound that harm by subjecting software developers to unlimited liability for third-party use of their work.

TECHNICAL BACKGROUND

The Tornado Cash ecosystem includes a number of separate technical components that the Court must distinguish in order to accurately assess the Indictment's allegations. It is important to evaluate each component of a blockchain-based protocol or ecosystem precisely, especially when apportioning liability, just as this Court did in *Risley v. Universal Navigation Inc.*, No. 22 CIV. 2780 (KPF), 2023 WL 5609200, at *2-8 (S.D.N.Y. Aug. 29, 2023) (distinguishing and describing smart contracts, liquidity pools, token creators, user interface, governance, and token holders).

This exercise is particularly important in the context of open-source software, where it is common for different software developers to work on different components of the ecosystem at different times. For example, the software developer who begins to write code for a smart contract protocol may not be the same developer who finishes the code or reviews it for bugs years later. The protocol developers may have no touchpoint at all with the user interface operators, who pay for server space and likely have no ability to alter the protocol.

The Indictment does not appropriately differentiate the components of the Tornado Cash ecosystem.[3] Instead, it groups together a variety of separate actors and technology—such as the developers, the relayers, the user interface, the protocol, and the TORN governance participants—and calls them, collectively, the "Tornado Cash service." *See generally* Ind. ¶¶ 1-3, 9-10, 26-28.[4] The Indictment then asserts that the Tornado Cash founders "operated" the "Tornado Cash service." Ind. ¶ 1. Neither statement accurately reflects how Tornado Cash works or the founders' relationship to parts of the ecosystem.

We believe that Mr. Storm and *amici curiae* Blockchain Association and Coin Center have correctly explained the technology of Ethereum and smart contract protocols, *see*, *e.g.*, ECF No. 37-1 at 16-24, and thus we only highlight certain nuances of the technology below.

Off-Chain User Interfaces Are Separate From On-Chain Protocols

A user interface is a "computer application . . . accessible through any standard internet browser" that provides a gateway for users to interact with a smart contract protocol and is distinct from the protocol itself. Ind. ¶ 13. A protocol consists entirely of self-executing smart contracts that are "on-chain," meaning they are deployed to a public blockchain network, are

[3] Because we are limited to the facts alleged in the Indictment, DEF takes no position on the level of decentralization achieved by the various components of the Tornado Cash ecosystem.

[4] For an in-depth technical review of the Tornado Cash protocol and user interface, see §4.2.2 of *Secret Notes And Anonymous Coins: Examining FinCEN's 2019 Guidance On Money Transmitters In The Context Of The Tornado Cash Indictment* by Cravath lawyers Daniel Barabander, Benjamin Gruenstein, and Evan Norris, https://tinyurl.com/mr25xjyb ("Cravath Working paper").

stored on its ledger, and operate continuously as long as the blockchain is operational. User interfaces, on the other hand, consist of entirely separate code for websites, mobile applications, or other programs that are "off-chain," meaning they are typically owned and controlled by the developers who built them, are stored by a centralized cloud provider such as Amazon Web Services, and operate only so long as their developers provide for their maintenance and upkeep.

Because of the distinction between user interfaces and their underlying protocols, control over a user interface does not equate to control over the underlying protocol. Often, smart contract protocols are accessible through many different user interfaces that are created and maintained by developers who had no role in coding the underlying protocol and may not even know each other. Changes to a user interface do not have any effect on the protocol itself.

Operators of user interfaces can implement compliance measures such as user due diligence, geofencing, and sanctions screening, thereby blocking sanctioned entities from accessing the protocol *through the user interface*. *See generally* Ind. ¶¶ 34, 39, 42. However, regardless of any restrictions implemented into a user interface, the underlying smart contract protocol itself will remain accessible to anyone with an internet connection who knows how to use a public blockchain. *See generally id.* (alleging founders could have built compliance tools into user interface but not protocol itself).

Privacy-Preserving Technology Is Uniquely Important For Blockchain-Based Transactions

At the center of this case is a privacy-preserving technology that allows users to engage in financial transactions on a public blockchain without the rest of the world peering over their shoulders. There is nothing illicit about the desire for financial privacy—it is a fundamental right deeply rooted in the history of our nation and codified in the First and Fourth Amendments to the U.S. Constitution, among many other places in federal law.

Privacy preservation is particularly important in DeFi because of the transparent nature of public blockchains. Traditional off-chain transactions provide substantially more privacy than on-chain transactions: cash is virtually untraceable, and transactions involving financial institutions like banks, credit card networks, and payment processors expose sensitive information only to those institutions, not the public at large. But blockchain-based transactions are posted to a public ledger that anyone can see; all of a user's transactions can be viewed for all time by anyone with access to the internet and who knows the user's wallet address.

The unique transparency of public blockchains creates major privacy concerns that can expose individuals to exploitation, invite retaliation for politically-sensitive contributions, and leave users' private and sensitive affairs exposed. For example, thieves can identify cryptocurrency users with large holdings and threaten them unless they give up their assets. Popper, *Bitcoin Thieves Threaten Real Violence for Virtual Currencies*, N.Y. Times (Feb. 18, 2018), perma.cc/3KCU-3ELC. And dangerous groups, such as Russians who target donations to Ukraine for cyber attacks, can use public cryptocurrency transactions as a basis for retaliation. *See Cybersecurity Advisory: Russian State-Sponsored and Criminal Cyber Threats to Critical Infrastructure*, CISA (May 9, 2022), perma.cc/C5TN-QL62. Software that solves these problems by preserving user privacy in public transactions should be viewed as neutral tools.

ARGUMENT

I. *THE GOVERNMENT'S NOVEL IEEPA THEORY IS UNSUPPORTED BY LAW*

The Indictment alleges an unprecedented theory of criminal liability under IEEPA. Pursuant to the theory, software developers who create general use software—with no intent that it be used by sanctioned entities and with no direct or proactive engagement with a sanctioned entity—would be criminally liable for a sanctioned entity's later use of that software. This novel theory is unsupported by the law, would create vast new precedent that goes far beyond this case, and would have a chilling effect on software developers in every industry.

A. *IEEPA Has Never Been Used To Criminalize The Creation Of Software That Is Unintentionally And Spontaneously Used By A Sanctioned Entity*

IEEPA has never been used to penalize an engineer who created technology with no particular end user in mind and that technology is later used by a sanctioned entity. IEEPA punishes U.S. persons who willfully export blocked property in their "possession or control," or provide funds, goods, or services to Specially Designated Nationals and sanctioned entities ("SDNs"). *See* 50 U.S.C.A. §1705(c); Executive Orders 13466 and 13722; 31 C.F.R. § 510.201 (collectively, "North Korea sanctions"); *see also* Ind. ¶¶ 51-54. But, as explained below, a software developer who creates open-source software with no end user in mind has not conspired to willfully violate IEEPA merely because an SDN uses that software years later.

In reviewing the most recent 78 IEEPA criminal prosecutions against individual defendants, the government consistently charges individuals who it claims have willfully, and with knowledge of their counterparty or the destination, traded goods with SDNs, directly transacted with SDNs, or otherwise proactively engaged with an SDN. *See* Exhibit A. In each of the cases in Appendix A, the defendant is accused of directly dealing with an SDN or taking some proactive step knowing that their conduct would reach an SDN. *Id.*; *see, e.g.*, *United States v. Atilla*, 966 F.3d 118, 122 (2d Cir. 2020) (evidence at trial established Atilla conspired to evade Iran sanctions by laundering billions of dollars' worth of Iranian oil proceeds and lied to Treasury Dept. to hide scheme); *United States v. Banki,* 685 F.3d 99, 103-104 (2d Cir. 2012), as amended (Feb. 22, 2012) (defendant knowingly received $3.4M as part of Iranian hawala system); *United States v. Sarvestani,* 297 F.R.D. 228, 230 (S.D.N.Y. 2014) (defendant intentionally concealed destination of satellite technology exported to Iran). Our extensive review did not reveal a single case where the government charged a defendant with violating IEEPA or conspiring to do so *without* an allegation that the defendant knew their counterparty or knew that the counterparty was several degrees away from an SDN. Moreover, in each case, the defendant *had a counterparty*. By contrast, a software developer publishing general use, open-source software does not have an identifiable counterparty at all.

These same principles hold true across all IEEPA cases, including those related to software developers. Among dozens of recent cases, just two appear to involve criminal IEEPA conspiracy allegations against a software developer. *See United States v. Amirnazmi*, 648 F. Supp. 2d 718, 721 (E.D. Pa. 2009), aff'd, 645 F.3d 564 (3d Cir. 2011) (Amirnazmi conspired to market and sell proprietary software to Iranian companies); *United States v. Galgoul*, No. 2:07-cr-00211, ECF No. 13 (E.D. La. Jun. 7, 2007) (Galgoul conspired to market and sell oilfield industry software to Iranian companies). In both cases, the defendant was accused of marketing and providing software directly to an SDN and providing expert advice or training on such software. *Amirnazmi*, 648 F. Supp. 2d at 721; *Galgoul*, ECF No. 13 ¶ 3. In other words, each case included allegations that the defendant manifested a knowing, *active* engagement with an SDN as his counterparty. General use software created without an SDN in mind has never been the subject of an IEEPA charge.[5]

The dearth of cases with theories analogous to the Indictment in this case is not surprising: a software developer who creates a program without an SDN in mind, without any evidence of communications with an SDN, and without the goal of future use by an SDN, has not acted "willfully." *See* 50 U.S.C.A. § 1705(c).

[5] Even in the recent case against Roman Sterlingov, who was convicted at trial for conduct related to running a mixer, the government did not accuse Mr. Sterlingov of violating IEEPA. *See United States v. Sterlingov*, 1:21-cr-00399 (RDM), ECF No. 8 (D.D.C. June 14, 2021).

To establish that a defendant "willfully" violated IEEPA, the government has to prove that he "acted with knowledge that his conduct was unlawful." *See Bryan v. United States*, 524 U.S. 184, 191–92 (1998); *United States v. Homa Int'l Trading Corp*., 387 F.3d 144, 147 (2d Cir. 2004) (applying the *Bryan* willfulness standard to IEEPA). Courts in this Circuit have consistently required that a defendant engage in conduct *knowing* that his counterparty, or the recipient or destination of his goods or services, was an SDN. *See, e.g., United States v. Kuyumcu,* 803 F. App'x 513, 514-515 (2d Cir. 2020) (plea allocution for IEEPA conspiracy was sufficient where defendant said he was "fully aware" he was shipping goods to "a company in Iran"); *United States v. Griffith*, 515 F. Supp. 3d 106, 120 (S.D.N.Y. 2021) ("It will be part of the government's burden to prove that . . . Griffith knowingly and willfully joined a conspiracy with knowledge of its unlawful object, *i.e.* the providing services to the DPRK"). Because IEEPA requires a specific intent, it is not enough for the government to demonstrate "proof of knowledge of past unlawful activity alone," but instead, "[t]he government must show some additional intent to further future unlawful activity to support criminal liability." *United States v. Al-Arian*, 308 F. Supp. 2d 1322, 1340 (M.D. Fla. 2004) (interpreting "willfully" under IEEPA: "the government must prove a defendant: (a) knew [] that an organization was a [SDN] . . . and (b) had a specific intent that the contribution be used to further the unlawful activities of the [SDN]"). And in order to allege an IEEPA conspiracy, the government must allege the defendant acted willfully "*at the time they joined* in a plan to engage in the unlawful acts." *United States v. Quinn*, 403 F. Supp. 2d 57, 60 (D.D.C. 2005) (emphasis added).

The lack of similar cases belies the novelty of the government's theory in the Indictment: that software developers become liable for violating IEEPA when they become aware that an SDN has used their software. The Indictment does not allege that the founders created privacyenhancing software for use by SDNs in particular, or for the purpose of sanctions evasion. *See* Ind. ¶¶ 9-10. Relevant here, the Indictment alleges that: Tornado Cash technology was published in August 2019 (¶ 9); OFAC designated a North Korean hacking group as an SDN and identified its ETH wallet as blocked property on September 13, 2019, and April 14, 2022, respectively (¶ 55); the sanctioned hacking group used Tornado Cash from April 4, 2022 to May 19, 2022 (¶¶ 58, 60); and the Tornado Cash founders knew about the group's use of the protocol as it happened in 2022 (¶¶ 61-63).

The Indictment then asserts that the Tornado Cash founders "continued to operate the Tornado Cash service and facilitate the Lazarus Group's money laundering and sanctions evasion" without alleging that the founders did a single thing differently than they had done for the prior two years. *See* Ind. ¶ 68. The Indictment seems to imply that the Tornado Cash protocol's continued existence after use by an SDN, and the founders conducting business as usual, constitutes an IEEPA violation. *See generally* Ind. ¶¶ 56-68.

Moreover, the Indictment does not allege that there was anything the founders could do to prevent an SDN from accessing the Tornado Cash protocol or that they failed to take action that would have been effective.[6] There is no allegation that at any point in time, the founders directly or indirectly engaged with SDNs, communicated with SDNs, solicited SDNs to use the Tornado Cash protocol, or did anything proactive to assist an SDN. Instead, the theory of liability in the Indictment involves neither proactive nor direct engagement: the software

[6] Notably, the Indictment alleges that the founders *did* attempt to block SDNs from accessing the user interface, but learned that effort was ineffective because the hacking group could evade the sanctions screen on the user interface. Ind. ¶¶ 65-66. However, the standard for criminal violations is not that it must be impossible for sophisticated sanctioned actors to evade controls, but rather that efforts to create controls were made. Otherwise, every OFAC civil sanctions enforcement case could be a criminal case.

developers are responsible for creating software that is later used by an SDN and merely becoming aware of it.[7]

But as discussed above, IEEPA has never been used to impose criminal liability on software developers for an SDN's spontaneous use of software, nor would such conduct meet IEEPA's "willful" mens rea requirement. This case appears to be one of first impression in which the government is using IEEPA to target individuals who merely became aware that an SDN was misusing a tool they created years prior and were not able to stop it. However, this theory of IEEPA liability would be a massive and unjustified expansion of the law and have far-reaching consequences for software developers and creators of new technology alike.[8]

B. *Developers of Non-Custodial Smart Contract Protocols Do Not Have "Possession or Control" of Any Person's "Property or Interests in Property"*

A software developer of a non-custodial smart contract protocol used to send and receive transaction messages does not have "possession or control" over anyone's "property or interests in property" as contemplated by North Korean sanctions. In each of the relevant provisions of 31 C.F.R. § 510.201(a), the law blocks "[a]ll property and interests in property that are in the United States, that come within the United States, or that are or come within the possession or control of any U.S. person of" North Korea-related SDNs.

[7] There was—and is—nothing anyone can do to stop someone from using the smart contracts. If the founders shut down the user interface and publicly abandoned the ecosystem, SDNs could still use the Tornado Cash smart contracts. Only a few weeks ago, someone used the protocol to send funds to a wallet associated with Blackrock. *See* Young, *Wallet Associated With BlackRock's Tokenized Fund Spammed With Unsolicited ETH From Tornado Cash*, Unchained (Mar. 21, 2024), https://tinyurl.com/2jxz52au. This is the nature of immutable smart contracts: users can be excluded from the user interface but not the protocol itself.

[8] For example, under this excessive theory by the government, every software developer who contributed to virtual private network (VPN) technology that secures internet traffic could be criminally (not just civilly) liable for the many sanctioned actors who have evaded geo-blocking and other sanctions controls across web2, fintech and banking generally, using VPNs.

"Possession or control" is not defined in IEEPA or the executive orders relevant here. However, a basic understanding of "possession or control" exemplifies why it does not apply to the developers of a smart contract protocol. Black's Law Dictionary defines "possession" as "[t]he fact of having or holding property in one's power; the exercise of dominion over property," and "[t]he right under which one may exercise control over something to the exclusion of all others." POSSESSION, Black's Law Dictionary (11th ed. 2019). "Control" is defined as "the power or authority to manage, direct, or oversee." CONTROL, Black's Law Dictionary (11th ed. 2019). In the context of an immutable smart contract protocol, a developer who initially created the protocol but has no ability to make changes to it cannot be said to have "possession or control" over any user funds that pass through it.

Likewise, a "property interest" is a formal term of art. It includes "rights of possession and control, held by an owner" and a "legitimate claim of entitlement to some legal or contractual benefit that cannot be taken away without due process." INTEREST, Black's Law Dictionary (11th ed. 2019) (definition of "property interest"). The term is synonymous with "ownership interest." *See id.* ("property interest . . . also termed ... ownership interest"). When someone has a property interest in something, he typically has the "rights of possession and control," *id.*, the "right to exclude," *Cedar Point Nursery v. Hassid*, 141 S. Ct. 2063, 2072 (2021), and the right of disposition, *Loretto v. Teleprompter Manhattan CATV Corp*., 458 U.S. 419, 435-36 (1982).

Neither smart contracts nor their developers have "possession or control" of user funds. Smart contracts are merely on-chain software programs that codify the preset terms of a transaction messages between users. The nature of a non-custodial protocol is that no person has custody of user funds other than the user herself. Essentially, a smart contract is simply a set of instructions for how a user's assets are to be handled and has no ability to alter those instructions after being deployed on the blockchain. Furthermore, the developer of an immutable smart contract protocol who can no longer change that protocol certainly does not have "possession or control" of user funds merely because they coded the protocol in the past.[9]

When using a non-custodial smart contract protocol like the Tornado Cash protocol, a user never gives up their property interest rights of possession and control. While common parlance talks about a user "sending" funds somewhere, which evokes an image of funds leaving the user's possession, that is not what happens in reality. An analogy is best here. In a non-custodial smart contract protocol that involves pooling assets, it is as if a user has asked a safe manufacturer build a safe around their assets, with the combination known only to the user. The safe holds the assets in place until the user decides to withdraw them by putting in the combination. While that safe is there, the safe manufacturer cannot access the assets, move them, or even prevent the user from opening the safe to withdraw them, because the combination to the safe is known only to the user. Throughout the encounter, the user maintains all of their property interests in the assets and never gives up their rights of possession or control over them, and the safe manufacturer never gains any property interests in user assets.

The software developer, like the safe manufacturer, never has "possession or control" over user property or property interests. Accordingly, it would be contrary to the plain meaning

[9] For an in-depth discussion of the degree of control the founders had over user funds and in assessing whether the Tornado Cash protocol is a money services business according to FinCEN guidance, see the Cravath Working Paper, §4. Notably, after "review of the public smart contract code" and the Indictment, this paper concludes, "The allegations make clear that the founders **controlled the UI and the relayers** at various times, but not how they, or Tornado Cash, **controlled the value itself,**" and "The founders had at most necessary control . . . but not sufficient control - meaning the founders could not have transferred value independently from the customer." *Id.* §4.1, 4.2.3 (emphasis in original). Necessary control over the user interface and/or the relayers, without sufficient control over the ability to transfer funds, does not equate to "possession or control" over property or property interests of users.

of IEEPA and North Korean sanctions to impose liability on a software developer of a non-custodial smart contract protocol.

C. The Novel Theory Of Developer Liability For Conspiracy To Violate IEEPA Contravenes The Rule Of Lenity And Due Process Principles

The government alleges its unprecedented theory of IEEPA liability in the absence of a single factual assertion that: (1) at the time the technology was created, the developers had an SDN in mind or designed their technology to be used by an SDN; (2) the developers had the power to exclude an SDN from this technology; or (3) the developers took any proactive steps to engage with an SDN at any point in time. *See* Ind. ¶¶ 84-88. The theory results in developers of any new technology being criminally responsible for an SDN's later use of that technology, even where there is no direct or active engagement with an SDN. As discussed above, our extensive review uncovered no previous IEEPA case that rested on such a theory, and because it is also not supported by the statutory text of IEEPA or North Korean sanctions, the government's attempt to use it here violates the principles of due process and the rule of lenity.

The "familiar principle that ambiguity concerning the ambit of criminal statutes should be resolved in favor of lenity" counsels against expanding IEEPA and North Korean sanctions to criminalize the creation of technology without any active engagement with a sanctioned entity. *See Skilling v. United States*, 561 U.S. 358, 410 (2010) (citations omitted); *United States v. Lanier*, 520 U.S. 259, 266 (1997) ("[T]he canon of strict construction of criminal statutes, or rule of lenity, ensures fair warning by so resolving ambiguity in a criminal statute as to apply it only to conduct clearly covered."); *see also Banki*, 685 F.3d at 109 (vacating convictions for violating Iran sanctions: "[t]he rule of lenity requires ambiguous criminal laws to be interpreted in favor of the defendants subjected to them") (quoting *United States v. Santos*, 553 U.S. 507, 514 (2008)).

The rule of lenity "vindicates the fundamental principle that no citizen should be held accountable for a violation of a statute whose commands are uncertain." *Id.*

While the Indictment does not specify which statutory provision the government relies on for Count III, there is no provision that clearly and unambiguously encompasses the facts here: imposition of liability for creating software that is later used by an SDN, without any direct or proactive engagement between the software developer and the SDN.[10]

Prosecuting a software developer based on a novel interpretation of the sanctions laws also violates the "fundamental principle" that "laws which regulate persons or entities must give fair notice of conduct that is forbidden or required," which is "essential to the protections provided by the Due Process Clause of the Fifth Amendment." *See F.C.C. v. Fox Television Stations, Inc.*, 567 U.S. 239 (2012) (due process prohibited agency from penalizing conduct occurring before change in interpretation); *Lanier*, 520 U.S. at 266 ("[D]ue process bars courts from applying a novel construction of a criminal statute to conduct that neither the statute nor any prior judicial decision has fairly disclosed to be within its scope[.]").

Neither IEEPA nor North Korean sanctions even facially appear to cover the government's novel theory that a software developer may be held liable for conspiring to violate sanctions laws by creating a general-purpose tool that is later used by an SDN without any proactive engagement with that SDN. The government's theory also leaves developers guessing as to what they must do when they cannot stop sanctioned entities from using their software, or when their efforts to do so prove ineffective. To avoid criminal liability, must the developers

[10] In Count III, the Indictment broadly refers to 50 U.S.C.A. § 1705, Executive Order 13722, and 31 C.F.R. § 510.201, but does not allege a violation of specific provision of Executive Order 13722 or 31 C.F.R. § 510.201, each of which are lengthy and have multiple provisions. *See* Ind. ¶¶ 84-88.

shutter operations within their control? And what must they do for software outside of their control? The government cannot answer these questions based on the law.

Under the government's theory, the following actions could result in criminal IEEPA charges for software developers, who would be shocked to learn that they violated sanctions law:

- A video game developer creates an online game where users can interact live and barter in-game goods, and pays for hosting services. An SDN uses a VPN to get around IP blocking and makes an account to play the game as intended. The developer continues to pay for website hosting services after finding out that this occurred.
- An email client such as Gmail continues to provide their software and pay for server space after finding out that an SDN used their email client to buy and sell goods.
- An iPhone app developer creates a general-purpose app available in the App Store. Despite Apple's efforts to comply with U.S. sanctions, an individual from a sanctioned country downloads the app, and the developer continues to fix bugs and put out updates for the app.

Taking the government's theory to its logical end, the law would require developers of any tool to, at the time of creation, anticipate the myriad ways that bad actors might someday use their tool and wall off all possible entry points. If it turns out that their efforts to exclude bad actors were insufficient, the developers must cease operations the day that an SDN uses their tool, lest they be found "liable for a third party's use of the [tool] to commit a traffic violation or to rob a bank." *Risley*, 2023 WL 5609200 at *14. In the blockchain context, requiring a software developer to do something in real time to stop an SDN from using an immutable smart contract protocol that they created is no more realistic or legally supportable than telling a self-driving car developer to make sure that a getaway car stops working in transit from the bank heist. *See id.* (directing liability in civil context: "In those circumstances, one would not sue the car company for facilitating the wrongdoing; they would sue the individual who committed the wrong.").[11]

[11] By comparison and as discussed in Coin Center's brief, the SWIFT system, central to funds transfers in the traditional financial system, has far *more* control over participants than developers of immutable smart contract protocols over users. Yet after a February 2016 hack by the DPRK "through unauthorized transactions" on SWIFT, no SWIFT developers or operators were charged and instead, the

But most importantly, these requirements are not actually found in the law itself. For that reason, the Court should reject the government's newfound theory of IEEPA liability for software developers as contrary to the rule of lenity and fundamental principles of due process.

II. *THE GOVERNMENT'S MONEY LAUNDERING THEORY MISUNDERSTANDS HOW IMMUTABLE SMART CONTRACT PROTOCOLS WORK*

Similar to its theory of IEEPA liability, the government's money laundering theory also seeks to hold software developers liable for conduct outside of their control. Count I is premised on the government's belief that software developers "operat[e]" the smart contracts that they publish and "conduct" transactions using those smart contracts. Ind. ¶¶ 1, 78. That belief misunderstands the basic relationship between smart contract protocols and their developers.

Smart contracts "are self-executing, self-enforcing programs that write the terms of the agreement between [users] directly into the program's code[.]" *Risley,* 2023 WL 5609200 at *3. When a person uses a smart contract protocol, they ordinarily do not engage with the protocol's developers in any way— "that is, when a given event occurs, the [transaction] auto-executes, without the need for third-party intervention from banks, lawyers, accountants, or the like." *Id.* The developers' work is done when they finish coding and launch the smart contract protocol, and then the protocol continues to operate on its own for as long as the underlying blockchain continues to function. That fact holds true regardless of the type of transaction that the protocol enables, whether a decentralized exchange as in *Risley* or privacy preservation as in this case.

To prove a conspiracy to commit money laundering, the government must establish that the defendant: (1) agreed to conduct, or to attempt to conduct, a financial transaction; (2) knew

hackers responsible for the heist were charged. *See* DOJ, *North Korean Regime-Backed Programmer Charged With Conspiracy to Conduct Multiple Cyber Attacks and Intrusions* (Sep. 6, 2018), https://tinyurl.com/5n8m6ptv.

"that the property involved [in the transaction] represents the proceeds of unlawful activity"; and (3) "proof of intent to conceal" the proceeds. *See* 18 U.S.C.A. § 1956(a)(1)(B)(1), (h); *United States v. Odiase*, 788 F. App'x 760, 762 (2d Cir. 2019); *United States v. Huezo*, 546 F.3d 174, 179-80 (2d Cir. 2008). These elements are absent in the case of software developers who create and deploy immutable smart contract protocols.

In the context of a traditional software company that offers financial transactions as a proprietary service, such as a centralized trading platform or payment processor, it may be reasonable to treat the company as "conducting" all of the transactions that it facilitates for its customers. That type of company typically has a direct legal relationship with its customers; has total control over who can access the service, how, when, and at what cost; has knowledge of all prospective transactions before they take place; and plays an essential role in accepting and executing its customers' transaction requests. Without the company's ongoing engagement and understanding, its customers' transactions could not and would not take place.

The technology at issue here is fundamentally different. The developers of an immutable smart contract protocol typically have no relationship with the protocol's users; have no control over the protocol's operation; have no knowledge of prospective transactions until after they take place; and play no role in executing transactions. Not only is there no "need for third-party intervention" in the case of an immutable smart contract protocol, *Risley*, 2023 WL 5609200 at *3, the protocol is *intervention-proof* by its very nature. Once a protocol is immutable, the developers have the same relationship with the protocol as everyone else in the world—the only way for them to "conduct" transactions related to the protocol is to become users themselves. In this way, smart contract developers are unlike financial institutions offering a proprietary service.

Instead, they are akin to construction workers building public works projects that anyone can use without their involvement after the build is complete.

Here, the Indictment acknowledges that the Tornado Cash protocol was immutable as of May 2020. *See* Ind. ¶ 26. The relevant time period for Count I did not begin until months later: "in or about September 2020 up to and including on or about August 8, 2022[.]" Ind. ¶ 77. During that period, the Indictment does not allege—and provides no basis to conclude—that the Tornado Cash founders agreed to conduct, or to attempt to conduct, transactions involving the protocol. Nor does the Indictment allege that they became users of the protocol themselves.

The Indictment only appears to allege the founders conducted two categories of "transactions" during the relevant time period that fit the definition in 18 U.S.C.A. § 1956(c)(4). First, Mr. Storm allegedly made regular payments to a service provider related to the Tornado Cash UI in 2021 and 2022. *See* Ind. ¶ 23. But the Indictment does not allege that those payments "in fact involve[d] the proceeds of specified unlawful activity," 18 U.S.C. § 1956(a), or that Mr. Storm made those payments "knowing" that they were the proceeds of unlawful activity and with intent to conceal such proceeds. *See Huezo*, 546 F.3d at 179 (holding that "a conviction for transaction money laundering . . . requires proof that the purpose or intended aim of the transaction was to conceal or disguise a specified attribute of the funds").

Second, the founders allegedly conducted various transactions related to the TORN token, including the initial distribution of the token in December 2020 and sales of tokens using a Binance account in 2022. *See* Ind. ¶¶ 27, 69, 73, 75. But again, the Indictment does not allege that any of these transactions involved the proceeds of specified unlawful activity or that the founders knew that they did, as required by 18 U.S.C.A. § 1956(a)(1). And the Indictment does not allege that the founders' TORN tokens represented the proceeds of wire fraud or a violation of the Computer Fraud and Abuse Act. *See* Ind. ¶ 78; *Odiase*, 788 F. App'x at 762 (requiring the government to show "knowledge that the property involved represents the proceeds of unlawful activity"). The allegation that the founders attempted to conceal their TORN sales, Ind. ¶ 73, does not save Count I—it is not a crime to conceal personal transactions involving lawful funds.

Instead of alleging specific transactions that satisfy Count I, the Indictment relies on conclusory and unsupported allegations about the Tornado Cash founders' relationship with the software that they created. *See generally* Ind. ¶¶ 45-50. For example, the Indictment states that the founders "were facilitating and participating in [money laundering transactions] through their operation of the Tornado Cash service." Ind. ¶ 50. But these sweeping statements cannot overcome the technical reality of an immutable smart contract protocol to substantiate an allegation that the Tornado Cash founders entered into a conspiracy to commit money laundering. None are alleged to have "conducted" user transactions, let alone to have done so with the required mens rea for a money laundering conspiracy. Since the Indictment does not allege a single transaction that satisfies 18 U.S.C.A. § 1956(a), Count I fails.

CONCLUSION

For the foregoing reasons, DEF urges the Court to reject the government's novel and inconsistent theories of criminal liability for software developers set forth in the Indictment and to consider the impact that its decision will have on all software developers in all industries.

April 5, 2024

Respectfully submitted,

DEFI EDUCATION FUND

Amanda Tuminelli
tuminelli@defieducationfund.org

Jacob Chervinsky
jake@defieducationfund.org