DMS : สัญญาอัจฉริยะสำหรับการให้บริการสตรีมเพลงบนเครือข่ายบล็อกเชนแบบ Ethereum

DMS : A Smart Contract for Decentralized Music Streaming Business on Ethereum-based Blockchain Network

ศุภกฤต แย้มวาจา

Suppakit Yamwaja

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science
Prince of Songkla University
2565

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

DMS : สัญญาอัจฉริยะสำหรับการให้บริการสตรีมเพลงบนเครือข่ายบล็อกเชนแบบ Ethereum

DMS : A Smart Contract for Decentralized Music Streaming Business on Ethereum-based Blockchain Network

ศุภกฤต แย้มวาจา

Suppakit Yamwaja

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science
Prince of Songkla University
2565

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

ชื่อวิทยานิพนธ์ DMS : สัญญาอัจฉริยะสำหรับการให้บริการสตรีมเพลงบนเครือข่ายบล็อกเชนแบบ Ethereum

ผู้เขียน นายศุภกฤต แย้มวาจา

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

.....

(ผู้ช่วยศาสตราจารย์ ดร.ชินพงศ์ อังสุโชติเมธี)

คณะกรรมการสอบ

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.พิศาล เศรษฐวงศ์)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.สุภาภรณ์ กานต์สมเกียรติ)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ภัทร อัยรักษ์)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ชินพงศ์ อังสุโชติเมธี)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิตสาขาวิชาวิทยาการคอมพิวเตอร์

.....

(ผู้ช่วยศาสตราจารย์ ดร. ดร.เถกิง วงศ์ศิริโชติ)

รักษาการแทนคณบดีบัณฑิตวิทยาลัย

ขอรับรองว่า ผลงานวิจัยนี้มาจากการศึกษาวิจัยของนักศึกษาเอง และได้แสดงความขอบคุณบุคคลที่มีส่วนช่วยเหลือแล้ว

ลงชื่อ .....
(ผู้ช่วยศาสตราจารย์ ดร.ชินพงศ์ อังสุโชติเมธี)
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

ลงชื่อ .....
(นายศุภกฤต แย้มวาจา)
นักศึกษา

ข้าพเจ้าขอรับรองว่า ผลงานวิจัยนี้ไม่เคยเป็นส่วนหนึ่งในการอนุมัติปริญญาในระดับใดมาก่อน และไม่ได้ถูกใช้ในการยื่นขออนุมัติปริญญาในขณะนี้

ลงชื่อ .....
(นายศุภกฤต แย้มวาจา)
นักศึกษา

ชื่อวิทยานิพนธ์ สัญญาอัจฉริยะสำหรับการให้บริการสตรีมเพลงบนเครือข่ายบล็อกเชนแบบ Ethereum

ผู้เขียน นายศุภกฤต แย้มวาจา

สาขาวิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2565

บทคัดย่อ

การจัดการค่าลิขสิทธิ์ในธุรกิจเพลงเป็นงานที่ทำโดยค่ายเพลง ผู้ค้าปลีกเพลง และบริษัทองค์กร performance-rights organization ที่จัดการค่าลิขสิทธิ์จากสื่อ-ออนไลน์ ค่าลิขสิทธิ์ส่วนใหญ่จะกระจายในลักษณะรวมศูนย์ แนวทางแบบรวมศูนย์อาจตรงไปตรงมาสำหรับการจัดการธุรกิจเพลง อย่างไรก็ตาม วิธีการแบบรวมศูนย์ทำให้ศิลปินไม่สามารถกำหนดรูปแบบรายได้สำหรับเพลงของตนและรับเงินได้ทันที เพื่อให้ศิลปินมีอิสระในการกำหนดรูปแบบรายได้ อาจจำเป็นต้องเสนอแนวทางแบบกระจายอำนาจ ในงานนี้ ผู้วิจัยได้เสนอให้ใช้ Blockchain และ Smart Contract ในธุรกิจเพลงเพื่อเสนอบริการสตรีมเพลงแบบกระจายศูนย์ ซึ่งช่วยให้ศิลปินสามารถออกแบบรูปแบบค่าลิขสิทธิ์ของตนเองได้อย่างอิสระและรับเงินทันที ข้อเสนอของเราอิงจากการขยายมาตรฐาน ERC20 และ ERC721 ในการเขียนสัญญาอัจฉริยะ ข้อเสนอของงานวิจัยนี้ถูกนำไปใช้และทดสอบในบล็อกเชนส่วนตัวบน Ethereum ในสภาพแวดล้อมที่ควบคุมโดยห้องปฏิบัติการเพื่อกำหนดความเป็นไปได้และค่าธรรมเนียม gas ที่จำเป็น ผลลัพธ์แสดงให้เห็นว่าตรรกะทางธุรกิจที่จำเป็นในสัญญาของเรานั้นไม่เสียค่าธรรมเนียม gas ในปริมาณที่สูง ดังนั้นจึงถือได้ว่าการออกแบบสัญญาของเราเหมาะสำหรับการปรับใช้ในบริการสตรีมเพลงแบบกระจายศูนย์จริง

Thesis Title DMS : A Smart Contract for Decentralized Music Streaming Business on Ethereum-based Blockchain Network
Author Mister. Suppakit Yamwaja
Major Program Computer Science
Academic Year 2022

ABSTRACT

Managing royalty fees in the music business is the task that is done by record labels, music retailers, and performance-right organization firms. Royalty fees are mostly distributed in a centralized manner. A centralized-based approach may be straightforward for managing a music business. However, a centralized-based approach prevents artists to be able to set their revenue model for their music and getting paid promptly. To allow an artist to have freedom in setting their revenue model, a decentralized-based approach may need to be proposed. In this work, we propose to apply Blockchain and Smart Contract in the music business to propose a decentralized-based music streaming service that allows artists can independently design their royalty fee model and get paid promptly. Our proposal is based on an extension of ERC20 and ERC721 standards on writing a smart contract. Our proposal is deployed and tested in a private Ethereum-based blockchain in a laboratory-controlled environment to determine the feasibility and gas fee required. The result suggests that the business logic required in our contract does not consume a high amount of gas fee. Hence, it can be deemed that our contract design is suitable for deploying in an actual decentralized-based music streaming service.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความช่วยเหลือและสนับสนุนจากบุคคลหลายฝ่ายผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูงมา ณ โอกาสนี้ คือ

ผู้ช่วยศาสตราจารย์ ดร.ชินพงศ์ อังสุโชติเมธี อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่กรุณาให้ความรู้ คำปรึกษาแนะนำ และช่วยเหลือในการแก้ปัญหาต่าง ๆ ให้แก่ผู้วิจัยเสมอมา รวมถึงถ่ายทอด หลักการ แนวคิดในการพัฒนาตัวเองและคุณธรรมสำหรับวิชาชีพแก่ผู้วิจัยมาโดยตลอด

ผู้ช่วยศาสตราจารย์ ดร.พิศาล เศรษฐวงศ์ ประธานกรรมการสอบวิทยานิพนธ์ ที่กรุณา ช่วยตรวจทานและแก้ไขวิทยานิพนธ์ให้มีความสมบูรณ์

ผู้ช่วยศาสตราจารย์ ดร.สุภาภรณ์ กานต์สมเกียรติ กรรมการสอบวิทยานิพนธ์ ที่ให้ ค้าปรึกษา และกรุณาช่วยตรวจทานและแก้ไขวิทยานิพนธ์ให้มีความสมบูรณ์

ผู้ช่วยศาสตราจารย์ ดร.ภัทร อัยรักษ์ กรรมการสอบวิทยานิพนธ์ ที่ให้ ค้าปรึกษา และกรุณาช่วยตรวจทานและแก้ไขวิทยานิพนธ์ให้มีความสมบูรณ์

อาจารย์ภาควิชาวิทยาการคอมพิวเตอร์คณะวิทยาศาสตร์มหาวิทยาลัยสงขลานครินทร์ทุกท่านที่ให้ความรู้ทางด้านวิชาการ ซึ่งสามารถนำความรู้นี้มาใช้ในการทำวิทยานิพนธ์

เจ้าหน้าที่ภาควิชาวิทยาการคอมพิวเตอร์ และเจ้าหน้าที่บัณฑิตวิทยาลัยทุกท่านที่ให้ ความช่วยเหลือ และอำนวยความสะดวกเกี่ยวกับเอกสารต่าง ๆ

เพื่อน ๆ พี่ ๆ และน้อง ๆ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ ที่ให้ คำปรึกษาและช่วยเหลือในการทำวิทยานิพนธ์

คุณแม่ธนภร แย้มวาจา คุณพ่อศุภสกุล แย้มวาจา และสมาชิกในครอบครัวทุกคน ที่ให้การสนับสนุนคอยเป็นห่วงสุขภาพ รวมทั้งให้การสนับสนุนในการทำวิทยานิพนธ์แก่ผู้วิจัยมาโดยตลอด ผู้วิจัยขอขอบพระคุณทุกท่านเป็นอย่างสูง ณ โอกาสนี้ด้วย

ศุภกฤต แย้มวาจา

สารบัญ

หน้าที่

สารบัญ **8**

รายการตาราง **10**

รายการรูปภาพ **11**

บทนำ **1**

1.1. ที่มาและความสำคัญของงานวิจัย **1**

1.2. วัตถุประสงค์ของการวิจัย **2**

1.3. ขอบเขตของการดำเนินการวิจัย **2**

1.4. ขั้นตอนและระยะเวลาการดำเนินการ **3**

1.4.1. ขั้นตอนการดำเนินการ 3

1.5. แผนการดำเนินการ **3**

เทคโนโลยีและงานวิจัยที่เกี่ยวข้อง **4**

2.1 BLOCKCHAIN 2.0 : ETHEREUM **4**

2.2 Interplanetary File System (IPFS) 6

2.3 งานวิจัยที่เกี่ยวข้อง **6**

2.3.1 BMCProtector: A Blockchain and Smart Contract Based Application for Music Copyright Protection (2018) 7

2.3.2 Computing exact worst-case gas consumption for smart contracts (2018) 7

2.3.3 Detecting Ponzi schemes on Ethereum: Towards healthier Blockchain technology (2018) 8

2.3.4 Music Streaming Application using Blockchain (2019) 8

2.3.5 A Study on Blockchain-based Music Distribution Framework: Focusing on Copyright Protection (2020) 9

2.3.6 Monetizing new music ventures through blockchain: Four possible futures? (2019) 9

2.3.7 The Impact of Blockchain Technology on the Music Industry (2019) 9

2.3.8 Using Blockchain for Online Multimedia Management: Characteristics of Existing Platforms (2020) 10
2.3.9 Piano Automatic Computer Composition by Deep Learning and Blockchain Technology (2020) 10
2.3.10 Securing music sharing platforms: A Blockchain-Based Approach (2022) 10
2.3.11 A DRM Solution for Online Content Using Blockchain - A Music Perspective (2018) 11
2.3.12 Blockchain-mediated Licensing: Legal Engineering for Artist Empowerment (2020) 11
2.3.13 Research on Decentralized Music Sharing Model Based on Consortium Blockchain (2019) 12
2.3.14 Digital music copyright management system based on blockchain (2020) 12
2.4 สรุปสิ่งที่ขาดไปในงานก่อนหน้านี้ **13**
การวิเคราะห์และการออกแบบสถาปัตยกรรมระบบการสตรีมเพลงแบบกระจาย **14**
การพัฒนา DMS CONTRACT และการทดลอง **21**
4.1. การพัฒนา DMS CONTRACT **21**
4.2. ผลลัพธ์ที่ได้จากเครือข่ายทดสอบ **24**
เครือข่ายทดสอบ **25**
สรุปผลงานวิจัย ปัญหา อุปสรรค และ แผนการทดลองต่อยอด **26**
5.1. สรุปผลการวิจัย **26**
5.2. ปัญหาและอุปสรรคในการวิจัย **27**
5.3. ข้อเสนอแนะ และ ทิศทางการดำเนินงานต่อในอนาคต **27**
บรรณานุกรม **28**

รายการตาราง

หน้าที่

ตารางที่ 1 ระยะเวลาในการดำเนินงานวิจัย..... 3
ตารางที่ 2 ผลวิเคราะห์มาตรฐาน ERC721 และ ERC2981 ต่อ REQUIREMENT การพัฒนา DMS 16
ตารางที่ 3 ผลการทดลอง: ค่าธรรมเนียมแก๊สที่จำเป็นสำหรับการดำเนินการ DMS 25

รายการรูปภาพ

หน้าที่

รูปที่ 1 กลไกและสัญญาการจ่ายค่าตอบแทนในธุรกิจดนตรีของระบบงานเดิม 1
รูปที่ 2 สถาปัตยกรรม DECENTRALIZED MUSIC STREAMING (DMS) 14
รูปที่ 3 DMS: METADATA JSON FORMAT 19
รูปที่ 4 DMS: ERC IMPORT 21
รูปที่ 5 FUNCTION LISTENINGMUSIC 22

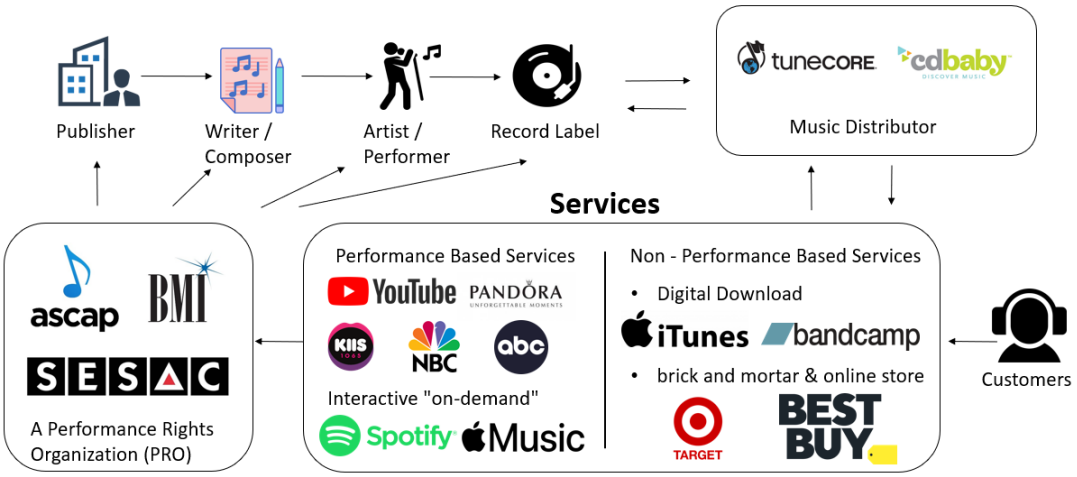
บทที่ 1

บทนำ

1.1. ที่มาและความสำคัญของงานวิจัย

ปัจจุบันการสตรีมเพลงในเทคโนโลยีดิจิทัลได้รับความนิยมอย่างแพร่หลาย ผู้คนนิยมบริการออนไลน์แบบสมัครสมาชิกเพื่อซื้อสำเนาอัลบั้มเพลงผ่านบริการสตรีมมิ่งออนไลน์เช่น Spotify , Apple iTunes, Google Music เป็นต้น ผู้ใช้งานสตรีมเพลงมีการเติบโตเพิ่มขึ้นอย่างรวดเร็ว ในปี 2021 มีผู้สมัครบริการแพลตฟอร์มดังกล่าวอย่างน้อยหนึ่งแห่งเป็นจำนวนถึง 523.9 ล้านคน เพิ่มขึ้นจากช่วงเวลาเดียวกันของปี 2020 ถึง 26.4% [1]

แม้ว่าระบบการสตรีมเพลงจะทำให้ธุรกิจดนตรีเติบโตขึ้นอย่างมาก แต่ประเด็นการผูกขาดของแต่ละแพลตฟอร์มยังทำให้ศิลปินและผู้ที่เกี่ยวข้องจำเป็นต้องยอมรับเงื่อนไขการจ่ายค่าตอบแทนของแต่ละแพลตฟอร์มซึ่งอาจมีความล่าช้าหรือมีสัดส่วนค่าตอบแทนที่ศิลปินไม่สามารถต่อรองได้ กลไกและสัญญาการจ่ายค่าตอบแทนแก่ผู้เกี่ยวข้องในธุรกิจดนตรีนั้นแสดงดังในรูปที่ 1



รูปที่ 1 กลไกและสัญญาการจ่ายค่าตอบแทนในธุรกิจดนตรีของระบบงานเดิม

จากรูปที่ 1 จะสังเกตได้ว่าลูกค้าจะทำการเข้าถึงบริการและเนื้อหาด้านดนตรีผ่านผู้ให้บริการเท่านั้น โดยผู้ให้บริการจะทำหน้าที่เก็บข้อมูลเนื้อหาและประสานงานกับหน่วยงาน Performance Rights Organization (PRO) กลุ่มผู้ค้าปลีกดนตรี และ ค่ายเพลงในการกระจายรายได้แก่ศิลปินและผู้ที่เกี่ยวข้อง สถานการณ์ดังกล่าวแม้ว่าจะเป็นปกติภายในธุรกิจดนตรี แต่ระบบดังกล่าวที่มีจุดด้อยคือ

ความรวดเร็วในการจ่ายค่าตอบแทนแก่ศิลปิน และความสามารถในการต่อรองสัดส่วนค่าตอบแทนของศิลปินโดยตรงเนื่องจากการบริหารงานในลักษณะรวมศูนย์โดยแพลตฟอร์มกลางหรือหน่วยงานกลางนี้

เพื่อแก้ปัญหาในด้านการผูกขาดของแพลตฟอร์มที่ใช้สถาปัตยกรรมแบบรวมศูนย์ในลักษณะงานดังที่กล่าวมาข้างต้น เทคโนโลยีที่ออกแบบมาเพื่อแก้ปัญหาเหล่านี้ได้คือเทคโนโลยีบล็อกเชน (Blockchain) ซึ่งมีคุณสมบัติที่สำคัญ 3 คุณสมบัติได้แก่ (1) ข้อมูลที่เขียนขึ้นมาแล้วนั้นไม่สามารถเปลี่ยนแปลงได้ (2) ทุกฝ่ายในระบบเป็นเจ้าของระบบร่วมกัน ไม่มีหน่วยงานกลางที่ดูแลระบบ และ (3) มีระบบ Smart Contract ที่ทำผู้ใช้งานระบบสามารถเขียนโปรแกรมใดๆก็ได้เพื่อให้ผู้ใช้งานระบบเรียกใช้งานร่วมกันได้ แต่ ณ ปัจจุบัน การนำ Blockchain มาใช้งานในธุรกิจดนตรีนั้นยังไม่มีการใช้งานในวงกว้าง การออกแบบโครงสร้างการให้บริการสตรีมเพลงบนเครือข่ายแบบกระจายนั้นจึงไม่มีการออกแบบที่แน่ชัดหรือมักจะเป็นความลับทางธุรกิจอยู่เสมอ

จากปัญหาที่กล่าวมาแล้วทั้งหมด ในงานวิจัยนี้ขอเสนอสถาปัตยกรรมการให้บริการการสตรีมเพลงโดยอาศัยเทคโนโลยีบล็อกเชน โครงสร้างของสถาปัตยกรรมและ Smart Contract ที่นำเสนอขึ้นใหม่ในงานนี้มีชื่อว่า “DMS” ที่มีชื่อที่ย่อมาจาก “Decentralized Music Streaming” สิ่งที่งานวิจัยนี้เสนอได้แก่ (1) ข้อเสนอการออกแบบสถาปัตยกรรมระบบ และ (2) โครงสร้างของ Smart Contract ที่เหมาะสมในการในการให้บริการสตรีมเพลงบน Blockchain ประโยชน์ที่ได้จากงานวิจัยนี้สามารถนำไปต่อยอดสร้างเป็นระบบ Open Platform สำหรับให้บริการสตรีมเพลงได้จริงในเชิงพาณิชย์

1.2. วัตถุประสงค์ของการวิจัย

เพื่อนำเสนอสถาปัตยกรรมแบบกระจายที่อาศัย Blockchain และ Smart Contract เพื่อให้บริการสตรีมเพลงที่ศิลปินสามารถกำหนดค่าตอบแทนได้ด้วยตนเองโดยไม่ผ่านผู้ให้บริการกลาง

1.3. ขอบเขตของการดำเนินการวิจัย

- นำเสนอปัญหาของระบบงานแบบเก่า ที่สอดคล้องกับลิขสิทธิ์เพลง ตั้งแต่กระบวนการจ่ายเงินจนกระทั่งถึงช่วงการกระจายส่วนแบ่งลิขสิทธิ์ไปยังวงของศิลปิน
- ออกแบบวิธีการแก้ไขปรับปรุงปัญหาการกระจายส่วนแบ่งลิขสิทธิ์ไปยังวงของศิลปิน
- พัฒนาสัญญาอัจฉริยะในการกระจายส่วนแบ่งลิขสิทธิ์ไปยังวงของศิลปิน

1.4. ขั้นตอนและระยะเวลาการดำเนินการ

1.4.1. ขั้นตอนการดำเนินการ

1.4.1.1. ศึกษางานวิจัยและเอกสารที่เกี่ยวข้อง

1.4.1.2. ศึกษาเทคโนโลยีและเครื่องมือสำหรับงานวิจัย

1.4.1.3. กำหนดขอบเขตของปัญหาในการวิจัย

1.4.1.4. วิเคราะห์และออกแบบกระบวนการประเมินปัญหาความสามารถในการใช้งาน

1.4.1.5. พัฒนาและทดสอบโดยรวมตามวิธีการที่ได้ออกแบบไว้

1.4.1.6. เขียนบทความวิจัยและเผยแพร่

1.4.1.7. จัดทำเอกสารวิทยานิพนธ์

1.5. แผนการดำเนินการ

ระยะเวลาดำเนินงานวิจัยแสดงในตารางที่ 1

ตารางที่ 1 ระยะเวลาในการดำเนินงานวิจัย

ขั้นตอน	2564							2565						
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.
1	→	→	→	→	→	→								
2		→	→	→	→	→								
3			→	→	→	→	→							
4				→	→	→	→	→	→					
5					→	→	→	→	→					
6								→	→	→	→	→	→	
7									→	→	→	→	→	

→ ระยะเวลาที่กำหนด

→ ระยะเวลาที่ได้ดำเนินการ

บทที่ 2

เทคโนโลยีและงานวิจัยที่เกี่ยวข้อง

เนื้อหาในบทนี้แบ่งเป็นสามส่วนได้แก่ (1) เทคโนโลยีที่เกี่ยวข้องด้าน Blockchain Ethereum (2) เทคโนโลยี Interplanetary File System สำหรับบันทึกข้อมูลลงเครือข่าย และ (3) งานวิจัยที่เกี่ยวข้องด้านการประยุกต์ใช้ Blockchain กับบริการต่างๆ

2.1 Blockchain 2.0 : Ethereum

อีเธอเรียม (Ethereum) [2] เป็นเทคโนโลยีบล็อกเชนที่มีแนวคิดว่าต้องการต่อยอดจากบล็อกเชนแบบเดิมที่ใช้ใน Bitcoin ให้สามารถมีการเขียน Business Logic ใดๆลงไปในเครือข่ายเพื่อเรียกใช้งานได้ ความสามารถในการเพิ่ม Business Logic ลงไปในเครือข่ายได้นี้ทำให้อีเทอเรียมได้รับการยอมรับในฐานะว่าเป็น “Blockchain 2.0” โดยที่ Bitcoin ได้ถูกการเรียกว่าเป็น “Blockchain 1.0” แทน

ปัจจุบันอีเธอเรียม บล็อกเชนแบบโอเพ่นซอร์สแบบกระจายอำนาจที่อนุญาตให้นักพัฒนาสร้างแอปพลิเคชันกระจายอำนาจ (Decentralized Application) ที่สามารถดำเนินการผ่านสัญญาอัจฉริยะ (Smart contract) ที่เขียนขึ้นโดยใช้ภาษาคอมพิวเตอร์ชื่อ Solidity โดยมี Ether เป็นสกุลเงินดิจิทัลดั้งเดิมของแพลตฟอร์ม เกิดขึ้นในปี 2013 และเริ่มใช้ในปี 2015 โดยมี Vitalik Buterin เป็นผู้ก่อตั้งแพลตฟอร์มนี้

เนื่องจากภาษา Solidity ในอีเธอเรียม มีความสามารถในระดับ Turing Complete กล่าวคือผู้พัฒนาสามารถใช้ภาษา Solidity ในการพัฒนา Business Logic ใดๆก็ได้อย่างไร้ข้อจำกัด ทำให้มี Smart Contract ในรูปแบบต่างๆเกิดขึ้นมากมายตามแต่นักพัฒนา ณ ปัจจุบันกลุ่มผู้พัฒนาอีเธอเรียมได้มีการรวบรวมและตั้งมาตรฐานรูปแบบของ Smart Contract กลางเพื่อให้เกิดมาตรฐานในการเขียนโปรแกรม Smart Contract ชุดรูปแบบมาตรฐานดังกล่าวนั้นถูกเรียกว่า Ethereum Request for Comments (ERC) [3] โดย ERC เป็นชุดรูปแบบมาตรฐานที่ ERC แต่ละแบบจะสื่อถึงมาตรฐานและโค้ดที่ใช้ในการพัฒนา Contract นั้นๆที่แตกต่างกัน โดยภาพรวมนั้นรูปแบบของ

Smart Contract ภายใน ERC นั้นแบ่งได้เป็นสามรูปแบบได้แก่ (1) Fungible Token (2) Non Fungible Token และ (3) Semi-Fungible Token รายละเอียดของแต่ละรูปแบบนั้นมีดังนี้

- **Fungible Token**

Fungible Token หมายถึง โทเคนที่ไม่ได้มีลักษณะเฉพาะตัว อันไหนก็สามารถแทนกันได้ [4] จะมี ERC20 เป็นมาตรฐาน มีการนำมาใช้เป็นเหรียญ SAND ที่สร้างมาเพื่อเป็นสกุลเงินที่ใช้ซื้อขายแลกเปลี่ยนสินค้าและบริการต่าง ๆ บนแพลตฟอร์ม

- **Non-Fungible Token**

Non-Fungible Token หมายถึง โทเคนที่นำเทคโนโลยีบล็อกเชนมาใช้ ทำให้สามารถแสดงความเป็นเจ้าของของสินทรัพย์นั้นๆ ได้ โดยแต่ละโทเคนจะแตกต่างกัน ไม่ต่างอะไรจากของสะสมหรือของมีมูลค่า ที่แต่ละชิ้นก็จะมีลักษณะเฉพาะตัว [5] โดยจะมี ERC721 เป็นมาตรฐาน มีการนำมาสร้างที่ดินและสินทรัพย์ต่าง ๆ ภายในเกม ซึ่งอาจมีลักษณะที่จำกัดจำนวนในการสร้างไว้ในตอนแรก ตัวอย่างเช่น ที่ดินบน The Sandbox นั้นมีเพียง 166,464 LANDs เท่านั้น

- ***Semi-Fungible Token***

Semi-fungible Token หมายถึงตัวแทนกลุ่มของสินทรัพย์ แทนที่จะใช้จับคู่กับสินทรัพย์เดี่ยว ๆ ที่แตกต่างกันเหมือน ERC721 โดยจะมี ERC1155 เป็นมาตรฐานใหม่ [6] ในขณะที่ ERC1155 สามารถใช้คำสั่ง transfer From ในการเคลื่อนย้ายเพลงหลาย ๆ เพลงพร้อมกันได้ แต่ก็ต้องแลกมาด้วยข้อเสียที่ทำให้เราไม่สามารถตรวจสอบประวัติของเพลงย้อนหลังของเพลงในอัลบั้มได้ เพราะเพลงที่ถูกเขียนด้วย ERC1155 นั้นมีคุณสมบัติเหมือนกันหมด ไม่เหมือน ERC721 ที่แต่ละสินทรัพย์จะมีความพิเศษไม่ซ้ำกัน ทำให้การตรวจสอบประวัติย้อนหลังนั้นทำได้ง่าย

เมื่อมีสกุลเงินที่ใช้ซื้อขายแลกเปลี่ยนแล้วก็จะต้องมีกระเป๋าเงินดิจิทัล (e-wallet) [7] รองรับที่ช่วยให้ผู้ใช้จัดการการเงินของตนได้ภายใต้แพลตฟอร์มเดียวเมื่อเติมเงินในบัญชีของตนผ่านวิธีการชำระเงินของผู้ให้บริการรายใดรายหนึ่งจึงสามารถใช้ e-wallet เพื่อเก็บเงินโอนเงินให้ผู้ใช้รายอื่นหรือชำระค่าสินค้าที่ร้านค้า โดยแต่ละร้านค้าที่ต้องการทำธุรกรรมผ่าน e-wallet ดังกล่าวนี้จำเป็นต้องเขียน Smart Contract ของร้านค้าตนเองเพื่อให้สอดรับกับมาตรฐาน ERC เพื่อที่จะได้รับการยอมรับว่ามีมาตรฐานและสามารถเชื่อมต่อระบบเพื่อแลกเปลี่ยนเงินตราหรือทรัพย์สินได้สะดวก

จุดสำคัญที่ควรระวังในการเขียนโปรแกรม Smart Contract และการเรียกใช้งาน คือทุกๆ Transaction ใดๆที่เกิดขึ้นบนอีเธอเรียมจำเป็นต้องมีการจ่ายค่าธรรมเนียมในการตรวจสอบยืนยัน Transaction แก่ Validator Node ภายในเครือข่ายอีเทอเรียม ค่าธรรมเนียมดังกล่าวเรียกว่า “Gas” [8] โดยปริมาณค่า Gas ที่ใช้ต่อ Transaction นั้นขึ้นอยู่กับความซับซ้อนและปริมาณข้อมูลที่จำเป็นต้องใช้ในการบันทึกบล็อกใหม่เพื่อเก็บ Transaction ดังกล่าวลงในเครือข่าย

2.2 Interplanetary File System (IPFS)

ระบบ Interplanetary File System (IPFS) ทีเป็นระบบสำหรับบันทึกไฟล์ใดๆแบบ peer-to-peer โดยไฟล์ที่บันทึกนั้นจะถูกกระจายไฟล์ไปเก็บไว้ยังเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Node ทั่วโลก แทนที่จะเก็บไฟล์ไว้ใน Server หรือคอมพิวเตอร์เพียงเครื่องเดียวทำให้ไฟล์ยังคงอยู่ตลอดไปตามใดที่ยังมีคนสนใจไฟล์นั้น และเก็บไฟล์นั้นไว้ [9] การทำงานของ IPFS จะต่างจาก HTTP แบบเดิมที่เมื่อเปิดเว็บใด ๆขึ้นมา Browser จะวิ่งไปเอาข้อมูลจาก Server ของเว็บนั้น ๆ แต่สำหรับ IPFS แทนที่จะส่งคำขอถึงเครือข่าย และรับเนื้อหาที่เก็บไว้เปลี่ยนเป็นการรับเนื้อหาโดยค้นหาจากค่า Cryptographic Hash ของไฟล์ โดยไฟล์ที่มีอาจถูกเก็บในหลายๆ Node หากว่ามี Node หนึ่งที่มีปัญหามันก็จะมองหาและใช้งานใน Node ต่อ ๆไป

เนื่องจากข้อจำกัดเรื่องค่า Gas ที่อาจมีปริมาณที่สูงหากต้องการบันทึกข้อมูลในปริมาณที่มาก และการขาดพื้นที่ในเก็บข้อมูลกลุ่มสื่อมัลติมีเดียหรือไฟล์ Binary ต่างๆ ระบบได้ถูกวางเป็นเทคโนโลยีที่นำมาใช้ร่วมกับระบบ Smart Contract เพื่อเก็บข้อมูลที่มีขนาดใหญ่เกินกว่าที่จะบันทึกลงไปใน Contract ได้ เช่นรูปภาพ หรือไฟล์เพลง เป็นต้น การทำงานร่วมกันของระบบ IPFS และ Smart Contract ทำให้เกิด Application ใหม่ๆบนบล็อกเชนหลายบริการ เช่น บริการตลาดค้ารูปแบบ NFT [10], [11] บริการฝากไฟล์ที่ใช้สถาปัตยกรรมแบบกระจาย [12], [13]เป็นต้น

2.3 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวกับเทคโนโลยีบล็อกเชนแพลตฟอร์ม Ethereum งานวิจัยที่เกี่ยวข้องของงานนี้อยู่ในการศึกษาผลกระทบของสัญญาอัจฉริยะใน Ethereum โดยการวิจัยในสาขานี้สามารถจำแนกได้เป็น 2 ประเภท คือ การวิเคราะห์เชิงวิเคราะห์และจากการทดลองโดยเรียงตามปีที่วิจัยได้เผยแพร่ดังนี้

2.3.1 BMCProtector: A Blockchain and Smart Contract Based Application for Music Copyright Protection (2018)

งานวิจัยนี้ได้เสนอให้นักดนตรีสามารถได้การคุ้มครองลิขสิทธิ์เพลงบนบล็อกเชนผ่านการทดลองโดยใช้ Ropsten Test Network งานวิจัยนี้ศึกษารูปแบบการคุ้มครองลิขสิทธิ์เพลงแบบใหม่ซึ่งไม่เพียงแต่รับประกันกระแสรายได้ของผู้ถือสิทธิ์เท่านั้น แต่ยังพิจารณาการคุ้มครองและติดตามลิขสิทธิ์ด้วยหลังจากซื้อเพลงแล้วงานนี้ไม่ว่าสิทธิของศิลปินจะถูกทำซ้ำ และจำหน่ายงานดนตรีที่อื่นด้วยการฝังลายน้ำดิจิทัลไว้ติดตามและเปิดเผยเจ้าของลิขสิทธิ์ที่แท้จริงและผู้ซื้อได้ เมื่อมีการเผยแพร่ไฟล์เสียงนี้อย่างผิดกฎหมาย เจ้าของลิขสิทธิ์สามารถค้นพบผู้ซื้อเดิมได้ [14]

อย่างไรก็ตาม ยังมีปัญหาบางอย่างที่น่ากังวลในอุตสาหกรรมเพลงดิจิทัล รวมถึงการจัดการสิทธิ์และการอนุญาต การเรียกเก็บค่าธรรมเนียมจากตัวกลาง และปัญหาการละเมิดลิขสิทธิ์นี่เป็นสิ่งสำคัญอย่างยิ่ง เนื่องจากหลายประเทศทั่วโลกได้เผยแพร่กฎหมายและกฎเกณฑ์ที่เกี่ยวข้องเพื่อช่วยปกป้องสิทธิประเภทนี้ ยังมีปัญหาทางธุรกิจและการเมืองบางอย่างที่อาจขัดขวางไม่ให้ผู้ใช้มีส่วนร่วม ตัวอย่างเช่น บางบริษัทไม่ต้องการเสียการควบคุมกระบวนการ ศิลปินบางคนไม่ต้องการเปิดเผยรายได้ต่อสาธารณะ แต่อย่างไรก็ตามงานนี้ทำให้สามารถที่จะทำให้ติดตามผู้ซื้อแม้จะมีการเผยแพร่ไฟล์เสียงนี้อย่างผิดกฎหมายได้

2.3.2 Computing exact worst-case gas consumption for smart contracts (2018)

เป็นงานวิจัยเชิงวิเคราะห์ที่มีจุดมุ่งหมายเพื่อค้นหาโมเดลการคำนวณที่สามารถอธิบายสัญญาอัจฉริยะใน Ethereum ได้ โดยแบบจำลองการวิเคราะห์ในงานนี้เรียกว่า Model-Checking Techniques [15] เป็นการคำนวณการใช้ค่าธรรมเนียม(Gas)ในกรณีที่เลวร้ายที่สุดสำหรับสัญญา

อัจฉริยะ มีแนวทางสองวิธีในการกำหนดปริมาณการใช้ Gas วิธีแรกใช้การแจงนับเส้นทางการดำเนินการเชิงสัญลักษณ์ และอีกวิธีหนึ่งใช้เส้นทางการคำนวณแบบแยกส่วนตามโครงสร้างโปรแกรม

อย่างไรก็ตามความไม่แน่นอนในการใช้ Gas อาจส่งผลให้เกิดความไร้ประสิทธิภาพ สูญเสียเงิน ซึ่งทำให้ยากต่อการสร้างสัญญาอัจฉริยะที่ซับซ้อนมากขึ้น เช่น สถาปัตยกรรมการสตรีมเพลง งานวิจัยนี้จึงทำให้สามารถที่จะตรวจสอบการใช้ค่าธรรมเนียมในกรณีที่เลวร้ายที่สุดสำหรับสัญญาอัจฉริยะได้

2.3.3 Detecting Ponzi schemes on Ethereum: Towards healthier Blockchain technology (2018)

งานวิจัยนี้เป็นงานเชิงวิเคราะห์ที่มีจุดมุ่งหมายเพื่อตรวจหาปัญหา Ponzi schemes [16] ซึ่งเป็นการดึงดูดนักลงทุนและจ่ายผลกำไรให้กับนักลงทุนรุ่นก่อน ๆ ด้วยเงินทุนจากนักลงทุนรายใหม่ ๆ โดยทำให้ผู้ที่ตกเป็นเหยื่อเชื่อว่าผลกำไรมาจากกิจกรรมทางธุรกิจที่ถูกกฎหมายที่เกิดขึ้นบนบล็อกเชน โดยใช้การทำเหมืองข้อมูล (data mining) และ machine learning ซึ่งแตกต่างจากการตรวจสอบแบบจำลองก่อนหน้านี้ งานวิจัยชิ้นนี้ประสบความสำเร็จในการสร้างแบบจำลองการจำแนกประเภทเพื่อตรวจจับรูปแบบการใช้ค่าธรรมเนียม

อย่างไรก็ตาม ผลอัลกอริทึมที่เสนอมาแสดงให้เห็นว่าแนวทางที่เสนอสามารถบรรลุความแม่นยำสูงสำหรับการใช้งานจริง แต่ต้องแยกคุณลักษณะจากบัญชีผู้ใช้ (accounts) และรหัสการดำเนินการของสัญญาอัจฉริยะ ซึ่งทำให้อัลกอริทึมใช้งานได้จริงน้อยลง แต่ก็ยังสามารถที่จะตรวจสอบการใช้ค่าธรรมเนียมและแบ่งแยกเป็นกลุ่มได้ว่าสัญญาอัจฉริยะนั้นสมควรที่นำมาใช้หรือไม่ด้วยกลุ่มของความคุ้มค่าของค่าธรรมเนียม

2.3.4 Music Streaming Application using Blockchain (2019)

งานวิจัยนี้เสนอการใช้ IPFS ในการจัดเก็บไฟล์เพลงแบบ peer to peer และใช้รูปแบบ Pay-Per-Play [17] ที่จะทำให้การสตรีมเพลงแต่ละครั้งจะต้องมีราคาคงที่หรือตามที่ศิลปินกำหนดไว้ในสัญญาอัจฉริยะเพื่อเล่นเพลงที่กำหนด ผู้ฟังสามารถให้ทิปศิลปินเพื่อสนับสนุนเพิ่มเติมได้จากการฟัง จ่ายเป็น ERC-20 โดยงานวิจัยเน้นไปที่การทำสถาปัตยกรรมระบบให้เข้ากับ IPFS และการกระจาย

ค่าลิขสิทธิ์ไปยังวงดนตรีของศิลปิน โดยจากงานวิจัยนี้ทำให้กระจายค่าลิขสิทธิ์ไปยังวงดนตรีของศิลปินเมื่อมีการเข้ามาฟังเพลงนั้น ๆ และเพิ่มรายได้จากการที่ผู้ฟังให้ทิปเพิ่มได้

2.3.5 A Study on Blockchain-based Music Distribution Framework: Focusing on Copyright Protection (2020)

งานวิจัยนี้เสนอการจัดจำหน่ายเพลงเพื่อปกป้องลิขสิทธิ์เพลงและสิทธิ์ของผู้ถือสิทธิ์ตามเทคโนโลยีบล็อกเชนได้โดยอัตโนมัติและทันทีโดยใช้เหรียญ (Token) ที่ทำหน้าที่เป็นเหมือนสิ่งของนั้นถูกสร้างขึ้นมาจากสัญญาอัจฉริยะงานวิจัยนี้ก็จะสร้าง Hash tree [18] ซึ่งเป็นที่อยู่ของบัญชีผู้ถือลิขสิทธิ์ขึ้นมาเพื่อตรวจสอบผู้มีส่วนร่วมในลิขสิทธิ์เพลงนั้น ๆผ่าน Hash tree ได้ก่อนที่จะเกิดบล็อกขึ้นมาในบล็อกเชนนั้นเอง ทำให้งานนี้สามารถแก้ปัญหาว่าผู้ที่มีส่วนร่วมในลิขสิทธิ์เพลงนั้นยังมีตัวตนอยู่หรือไม่บัญชีมีปัญหารึเปล่าก่อนที่จะทำการส่งค่าลิขสิทธิ์ได้

2.3.6 Monetizing new music ventures through blockchain: Four possible futures? (2019)

งานวิจัยนี้กล่าวถึงบทบาทที่เกิดขึ้นใหม่จากการที่เทคโนโลยีบล็อกเชนจะลบตัวกลางออกไปซึ่งอำนวยความสะดวกในความสัมพันธ์โดยตรงระหว่างศิลปินและแฟน ๆ การค้นพบเบื้องต้นก็คือตัวกลางในรูปแบบใดรูปแบบหนึ่งจะยังคงอยู่ซึ่งบทความนี้จะเรียกว่า infomediaries เป็นผู้ให้ข้อมูลโดยสรุปความเป็นไปได้ในการนำเทคโนโลยีบล็อกเชนมาใช้ในอุตสาหกรรมเพลง ตั้งแต่กลุ่มศิลปินอิสรเสรีไปจนถึงค่ายเพลง [19]

2.3.7 The Impact of Blockchain Technology on the Music Industry (2019)

เป็นงานวิจัยที่ได้ตรวจสอบผลกระทบที่อาจเกิดขึ้นของเทคโนโลยีบล็อกเชนต่ออุตสาหกรรมเพลงโดยการวิเคราะห์มุมมองของนักวิชาการและผู้เชี่ยวชาญในอุตสาหกรรม โดยใช้ข้อมูลการเติบโตและขนาดของอุตสาหกรรมเพลง รวมถึงส่วนแบ่งที่ได้จากค่าลิขสิทธ์ผ่านทางดิจิตอล ของปี 2017 ผลสรุปออกมาว่าผลกระทบเชิงบวกต่อวงการเพลงอาจเกิดขึ้นได้ก็ต่อเมื่อมีลูกค้าที่เข้าใจในเทคโนโลยีใหม่นี้มากพอ [20]

2.3.8 Using Blockchain for Online Multimedia Management: Characteristics of Existing Platforms (2020)

เป็นงานวิจัยที่ศึกษาเกี่ยวกับการใช้บล็อกเชนในอุตสาหกรรมมัลติมีเดียออนไลน์ผ่านการตรวจสอบโดยงานวิจัยอื่นๆจำนวน 30 ฉบับ ที่เผยแพร่ระหว่างปี 2016 ถึง 2018 ซึ่งรายงานการใช้บล็อกเชนสำหรับการจัดการมัลติมีเดียที่หลากหลายในอุตสาหกรรมเพลงและโฆษณา การดูแลสุขภาพ โซเชียลมีเดีย อีเธอเรียม ถูกพบว่าเป็นบล็อกเชนที่ได้รับความนิยมมากที่สุดและเป็นเครื่องพิสูจน์การทำงานของกลไก Consensus ที่จะอธิบายในบทที่ 3 แพลตฟอร์มมากกว่าครึ่งให้รางวัลแก่ผู้ใช้ที่เข้ามาเป็นผู้ตรวจสอบธุรกรรม แพลตฟอร์มส่วนใหญ่ใช้โทเค็นและสัญญาอัจฉริยะเพื่อทำให้การกระจายรายได้เป็นไปโดยอัตโนมัติหรือเพื่อให้สามารถเข้าถึงข้อมูลได้ แสดงให้เห็นว่าแพลตฟอร์มบล็อกเชนมัลติมีเดียส่วนใหญ่ได้ใช้ความสามารถในการสร้างรายได้แล้ว [22]

2.3.9 Piano Automatic Computer Composition by Deep Learning and Blockchain Technology (2020)

เป็นงานวิจัยที่สำรวจการคุ้มครองลิขสิทธิ์และการจัดการเพลงดิจิทัลโดยประยุกต์ใช้เทคโนโลยี deep learning และบล็อกเชนในการสร้างผลงานเพลงดิจิทัล ใช้เพลงบรรเลงเปียโนมาเป็นตัวอย่าง ผ่านการอธิบายอย่างละเอียดของเครือข่ายประสาทเทียมตามวิธีการดังนี้ Recurrent Neural Network (RNN) , Long-Short-Term Memory (LSTM) และสุดท้าย Gated Recurrent Unit (GRU) ผลลัพธ์แสดงให้เห็นว่าแบบจำลอง GRU-RNN แสดงผลที่น่าพอใจในการประเมินการวิเคราะห์ด้วยตนเองและวิเคราะห์ด้วยข้อความ ผลลัพธ์มีผลในเชิงบวกต่อการส่งเสริมการพัฒนาและการประยุกต์ใช้วิธีการเรียนรู้เชิงลึกและเทคโนโลยีบล็อกเชนในเพลงดิจิทัล [23]

2.3.10 Securing music sharing platforms: A Blockchain-Based Approach (2022)

งานวิจัยนี้ได้ศึกษาผลกระทบจากเทคโนโลยีดิจิทัลระบบจัดเก็บข้อมูลเพลงบนเครือข่ายบล็อกเชนโดยใช้สัญญาอัจฉริยะเพื่อให้ศิลปินได้รับค่าลิขสิทธิ์ที่รวดเร็วและมีประสิทธิภาพ โดยงานนี้จะใช้

Node.js เพื่อทำการทดลองระบบ และได้ทดสอบ Remote Procedure Calls (RPC) ด้วยบัญชีที่มีอยู่สำหรับการพัฒนาสัญญา และใช้ตัวสำรวจบล็อกเพื่อติดตามข้อมูลเพลงบนบล็อกเชน สถานะการป้องกันการโจมตีของเครือข่ายแสดงให้เห็นแล้วว่าบันทึกที่เก็บไว้ในไฟล์เพลงที่ดาวน์โหลดนั้นสอดคล้องกับรายได้ที่งานวิจัยนี้คาดหวังไว้ และมีรายได้อย่างมีประสิทธิภาพเกิดขึ้นได้ด้วยระบบที่งานวิจัยนี้นำเสนอ [24]

2.3.11 A DRM Solution for Online Content Using Blockchain - A Music Perspective (2018)

เป็นงานวิจัยที่เสนอระบบ DRM เพื่อสร้างรายได้ ติดตาม และควบคุมเนื้อหาออนไลน์ข้ามแพลตฟอร์ม งานวิจัยนี้นำเสนอสกุลเงินดิจิทัล Asset Assertion (AA) บนบล็อกเชนที่ได้รับอนุญาต เพื่อให้สามารถเผยแพร่และติดตามเนื้อหาดิจิทัลได้อย่างปลอดภัย ด้วยวิธีที่งานวิจัยนี้เสนอนี้ช่วยให้เจ้าของข้อมูลสามารถควบคุมการไหลของข้อมูลเพลงได้แม้หลังจากที่เผยแพร่ โดยการสร้างตัวอ่านข้ามแพลตฟอร์มที่ปลอดภัยและติดตั้งเองซึ่งอยู่ในส่วนหัวของไฟล์เนื้อหาดิจิทัลที่แทรกเข้าไปก่อนที่จะนำไฟล์เพลงเข้าสู่ระบบกระจายอำนาจ [25]

2.3.12 Blockchain-mediated Licensing: Legal Engineering for Artist Empowerment (2020)

งานวิจัยนี้ตรวจสอบว่าเทคโนโลยีและรูปแบบการออกแบบใหม่ๆ จาก Web3 เช่น บล็อกเชน ข้อมูลที่เชื่อมโยง และสัญญา Ricardian สามารถทำให้กระบวนการทำงานโดยอัตโนมัติเพื่อลดความซับซ้อน ความเร็ว เพิ่มการชำระเงิน ปรับปรุงการติดตาม และให้ประโยชน์อื่นๆ ในอุตสาหกรรมเพลง ในงานวิจัยนี้พยายามเพื่อลดความซับซ้อนและค่าใช้จ่ายในการทำธุรกรรมในกระบวนการออกใบอนุญาตโดยการพัฒนาใบอนุญาตเพลงอัตโนมัติ ในการทำ ขั้นแรกได้ทำการตรวจสอบวิจัยที่เกี่ยวกับความซับซ้อนของดนตรีและเทคโนโลยี Web3 เพื่อให้พื้นหลังและบริบทในการออกใบอนุญาตเพลงโดยอัตโนมัติ จากนั้นจึงพัฒนาวิธี Practice Tokenized Drafting (PTD) ซึ่งเป็นหลักการและแนวทางปฏิบัติหลักสำหรับการร่างสัญญา Ricardian ที่โต้ตอบกับเทคโนโลยี Web3 และ Tokenized Music License (TML) มาตรฐาน RC-Web3 แบบฟอร์มขอใบอนุญาตเพลงบนแพลตฟอร์ม OpenLaw ทั้ง PTD และ TML สามารถปรับให้เข้ากับความต้องการของผู้มีส่วน

ได้ส่วนเสียในวงการเพลงและให้คำแนะนำแก่ผู้ปฏิบัติงานด้านกฎหมายในการร่าง RC-Web3 ขึ้นมาอีกด้วย [26]

2.3.13 Research on Decentralized Music Sharing Model Based on Consortium Blockchain (2019)

งานวิจัยนี้ได้สร้างเครือข่ายทดสอบ อีเธอเรียม เพื่อทดสอบข้อตกลงร่วมอย่าง PoW และ PoA ว่าอัลกอริธึมใดที่เหมาะสมกับแอปพลิเคชันของลิขสิทธิ์เพลงดิจิทัลบนบล็อกเชนแพลตฟอร์มอีเธอเรียม ผลการวิจัยพบว่าเมื่อเวลาตรวจสอบธุรกรรมเท่ากับ 100 วินาที ปริมาณงานของระบบจะสูงถึง 324 บล็อกที่รอดำเนินการต่อวินาที ซึ่งสูงกว่าอัลกอริทึม PoW อย่างมาก และมีประสิทธิภาพการทำงานสูง แต่ไม่มีความเป็นสถาปัตยกรรมแบบกระจาย [27]

2.3.14 Digital music copyright management system based on blockchain (2020)

งานวิจัยนี้จัดการลิขสิทธิ์เพลงดิจิทัลได้รับการออกแบบและสร้างโดยใช้แพลตฟอร์มบล็อกเชน VNT Chain ในระบบเทคโนโลยีบล็อกเชนถูกใช้เพื่อให้หลักฐานของลิขสิทธิ์เพลงอัลกอริทึมของ Shazam ถูกใช้เพื่อพิสูจน์การสร้างสรรค์ผลงานเพลงลิขสิทธิ์ และใช้สัญญาอัจฉริยะเพื่อรับประกันความปลอดภัยและความน่าเชื่อถือของ การทำธุรกรรม ระบบนี้ประกอบด้วยการทำงาน 6 อย่างได้แก่ การจัดการผู้ใช้ , การจดทะเบียนลิขสิทธิ์ , การซื้อขายลิขสิทธิ์,การตรวจสอบการละเมิด,การพิสูจน์หลักฐาน และระบบนิเวศทางดนตรี ครอบคลุมส่วนหลักของการจัดการลิขสิทธิ์ ตามความต้องการที่แตกต่างกันของข้อมูลทางธุรกิจบล็อกเชน ใช้ IPFS และ MySQL ในการจัดเก็บข้อมูลตามลำดับสำหรับระบบ ผลการทดลองแสดงให้เห็นว่าเวลาในการลงทะเบียนลิขสิทธิ์ของเพลงแต่ละเพลงเพิ่มขึ้นประมาณ 1.9 วินาที และการจัดเก็บข้อมูลลายนิ้วมือของเพลงหนึ่งเพลงบน IPFS นั้นมีขนาดประมาณ 8 MB ซึ่งตรงตามข้อกำหนดด้านประสิทธิภาพของระบบที่คาดไว้ [28]

2.4 สรุปสิ่งที่ขาดไปในงานก่อนหน้านี้

จากงานวิจัยที่ผ่านมาสรุปได้ว่าสิ่งที่ขาดไปนั้น แม้ว่าจะมีการคิดค่าลิขสิทธิ์จากการที่มีผู้ฟังเข้ามาเล่นเพลงในระบบแต่ไม่ได้รวมไปถึงการแลกเปลี่ยนเหรียญที่เกิดจากการสร้างด้วยสัญญาอัจฉริยะ ทำให้เมื่อเปลี่ยนความเป็นเจ้าของเพลงยังไม่มีการคิดค่าลิขสิทธิ์ที่เกิดจากมูลค่าในการแลกเปลี่ยนครั้งนั้น และงานวิจัยเชิงทดลองก่อนหน้านี้ยังไม่มีการคำนวนค่าธรรมเนียมของการทำสัญญาอัจฉริยะอย่างที่งานวิจัยเชิงวิเคราะห์ได้เสนอมาก่อนหน้านี้ และสุดท้ายยังไม่มีงานใดเสนอในส่วนของความหน่วงในการส่งค่าลิขสิทธิ์ไปยังสมาชิกวงดนตรีแต่ละคนว่าสามารถได้รับล่าช้าหรือรวดเร็วขนาดไหน

งานนี้จึงนำเสนองานวิจัยที่ลงรายละเอียดไปที่โครงสร้างของสัญญาอัจฉริยะว่ามีรายละเอียดอย่างไรให้สามารถส่งค่าลิขสิทธิ์ไปยังผู้มีส่วนเกี่ยวข้องกับเพลงได้พร้อมๆกันพร้อมกับจัดการสิทธิ์ต่างๆที่เกี่ยวข้องกับเพลงหรืออัลบัมนั้นๆได้เพื่อให้เหมาะสมในการในการให้บริการสตรีมเพลงบน Blockchain โดยจะทำการวิเคราะห์และการออกแบบในบทถัดไป

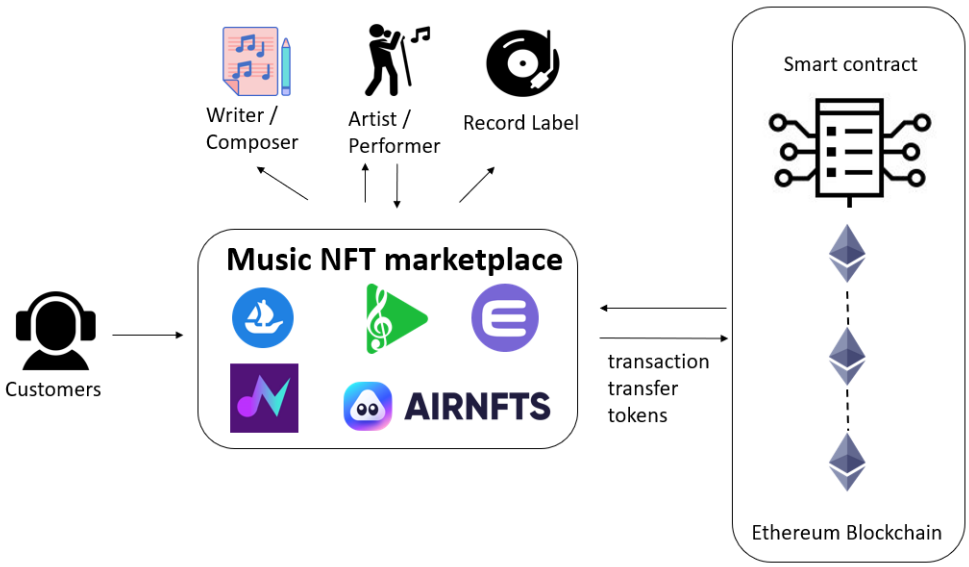
บทที่ 3

การวิเคราะห์และการออกแบบสถาปัตยกรรมระบบการสตรีมเพลงแบบกระจาย

เนื้อหาในบทนี้จะกล่าวถึงรายละเอียดในการวิเคราะห์ และการออกแบบสถาปัตยกรรมการสตรีมเพลงแบบกระจายบนเครือข่ายบล็อกเชน เพื่อการดังกล่าว สิ่งที่จำเป็นต้องทำการกำหนดวิเคราะห์ได้แก่ (1) ภาพรวมของสถาปัตยกรรมโครงสร้างของระบบ และ (2) บทวิเคราะห์การออกแบบว่าจากสถาปัตยกรรมที่กำหนดและความต้องการจากผู้ใช้งานนั้น Smart Contract ที่เหมาะสมควรมีโครงสร้างแบบใด รายละเอียดของการวิเคราะห์และข้อเสนอของทั้งสองส่วนนั้นมีดังต่อไปนี้

3.1 สถาปัตยกรรมของระบบ Decentralized Music Streaming (DMS)

เพื่อให้บริการระบบการสตรีมเพลงด้วยสถาปัตยกรรมแบบกระจาย ในงานวิจัยนี้ขอเสนอว่าควรออกแบบให้เพลง หรืออัลบั้มเพลงนั้นเป็น Token ประเภท Non-Fungible กล่าวคือ ผู้ใช้งานหรือศิลปินนั้นสามารถขายหรือเปิดเพลงได้โดยตรงผ่าน NFT ดังสถาปัตยกรรมที่แสดงดังรูปที่ 2 ดังต่อไปนี้



รูปที่ 2 สถาปัตยกรรม Decentralized Music Streaming (DMS)

จากรูปที่ 2 จะสังเกตได้ว่าลูกค้าจะทำการเข้าถึงบริการและเนื้อหาด้านดนตรีผ่านแพลตฟอร์มบล็อกเชนได้โดยตรง โดยแพลตฟอร์มทำหน้าที่ในการเป็นตลาดกลางโดยมีสัญญาอัจฉริยะในการกระจายรายได้แก่ศิลปินและผู้ที่เกี่ยวข้องได้โดยตรงไม่ต้องผ่านตัวกลางอย่างหน่วยงาน Performance Right Organization (PRO) กลุ่มผู้ค้าปลีกดนตรี อีกต่อไป สถานการณ์ดังกล่าวทำให้ธุรกิจดนตรี ได้สัดส่วนค่าตอบแทนของศิลปินโดยตรงเนื่องจากการบริหารงานในลักษณะกระจายศูนย์โดยแพลตฟอร์มบล็อกเชนนี้

แนวคิดหลักของ DMS จะสังเกตเห็นได้ว่าจะเน้นการใช้งานโครงข่าย Blockchain 2.0 หรืออีเธอเรียม ที่สามารถให้ผู้ใช้งานและนักพัฒนาสามารถทำการติดตั้ง Business Logic ของตนเองลงไปในเน็ทเวิร์คได้ ในกรณีของ DMS นั้นมีข้อเสนอว่าให้เพลงแต่ละเพลงหรือแต่ละอัลบั้มนั้นเป็น NFT Token ของตนเอง 1 เพลง หรือ 1 อัลบั้มต่อ 1 Token แต่อย่างไรก็ตาม ความสามารถแต่เดิมของ NFT นั้นทำได้เพียงการระบุว่าผู้ใดเป็นเจ้าของ Digital Asset ใดเพียงเท่านั้น ไม่สามารถนำมาใช้งานครอบคลุมได้ถึง Business Logic ด้านการให้บริการสตรีมเพลง เพื่อการดังกล่าว ใน DMS ผู้วิจัยได้กำหนดข้อกำหนดขั้นต่ำว่า Smart Contract ที่เหมาะสมในการใช้งานใน DMS นั้นจำเป็นต้องมีความสามารถดังรายการขั้นต่ำ 6 ข้อต่อไปนี้

1) ตั้งชื่อเพลง รายละเอียดต่างๆของศิลปิน ผู้แต่งเพลง ค่ายเพลงได้
2) คิดค่าบริการผู้ฟังเพลงได้
3) กำหนดส่วนแบ่งได้ทั้งในรูปแบบ Utility Coin หรือ ERC 20 Token
4) ตั้งมูลค่าสำหรับกรณีการโอนลิขสิทธิ์ถาวรได้
5) กำหนดให้ในแต่ละ Token มีได้อย่างน้อย 1 เพลง (กรณีเป็น Single) หรือมากกว่า 1 เพลง (กรณีเป็น Album)
6) จัดสรรรายได้ส่วนแบ่งได้แก่ศิลปิน นักดนตรี หรือค่ายเพลงได้โดยตรงโดยไม่ต้องรอการ Clearing โดย Performance Right Organization

เพื่อลดความจำเป็นในการออกแบบโครงสร้างของ Contract ใหม่ และเพื่อให้งานสอดรับกับมาตรฐาน ERC เดิมได้ด้วย งานวิจัยนี้จึงได้เลือกที่จะพัฒนา DMS โดยใช้วิธีการต่อยอดจากโครงสร้าง ERC แบบเดิมและเพิ่มส่วนต่อขยายให้ ERC มามีความสามารถดังที่ต้องการใช้ใน DMS ได้บทวิเคราะห์ดังกล่าวนั้นจะกล่าวในหัวข้อย่อยต่อไปของบทนี้

3.2 บทวิเคราะห์มาตรฐาน ERC ต่อการพัฒนา DMS

มาตรฐานที่เกี่ยวข้องกับการพัฒนา DMS ที่งานนี้ได้นำมาวิเคราะห์นั้นเป็นมาตรฐานที่เกี่ยวข้องกับ NFT ได้แก่ ERC721 และ ERC2981 [29] ผลวิเคราะห์ความสามารถของ ERC721 และ ERC2981 เดิมได้อธิบายไว้ในตารางที่ 2

ตารางที่ 2 ผลวิเคราะห์มาตรฐาน ERC721 และ ERC2981 ต่อ Requirement การพัฒนา DMS

Requirement	ส่วนของความสามารการใช้งานจาก ERC พื้นฐานต่างๆ	
	ERC721	ERC2981
ตั้งชื่อเพลง รายละเอียดต่างๆของศิลปิน ผู้แต่งเพลง ค่ายเพลง	Partial	Not covered
คิดค่าบริการผู้ฟังเพลงได้	Partial	Not covered
กำหนดส่วนแบ่งได้ทั้งรูปแบบ Utility Coin หรือ ERC20	Partial	Not covered
ตั้งมูลค่าสำหรับกรณีการโอนลิขสิทธิ์ถาวรได้	Partial	Not covered
กำหนดให้ในแต่ละ Token มีได้อย่างน้อย 1 เพลง (กรณีเป็น Single) หรือมากกว่า 1 เพลง (กรณีเป็น Album)	Partial	Not covered
ตั้งค่าการจัดสรรเปอร์เซ็นต์ที่ได้จากค่าบริการและการขายเพลงจ่ายเป็นค่าลิขสิทธิ์เพลงให้ผู้มีส่วนร่วมในผลงานเพลง	Not covered	Partial

จากตารางที่ 2 จะสังเกตเห็นได้ว่าทั้ง ERC721 และ ERC2981 ที่เป็นมาตรฐานเดิมในการจัดการ NFT นั้นยังไม่สามารถใช้ในการจัดการ Business Logic ของการให้บริการเพลงได้อย่างครบถ้วน โดยรายละเอียดในแต่ละประเด็นนั้นมีดังต่อไปนี้

- ERC721 สามารถตั้งชื่อเจ้าของ Digital Assets ได้แต่ยังไม่สามารถระบุข้อมูลได้ละเอียดถึงว่าเพลงแต่ละเพลงนั้นมีใครเป็นผู้แต่ง เป็นเพลงของค่ายใด ผู้เกี่ยวข้องกับเพลงนั้นๆมีผู้ใดบ้าง ERC 2981 ไม่ครอบคลุมประเด็นนี้

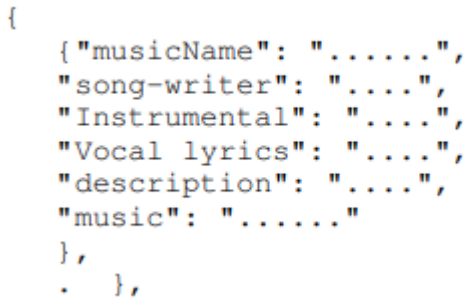
- ERC721 สามารถคิดค่าบริการการใช้งานผู้ขอใช้ NFT ได้แต่ทำได้เพียงในรูปแบบการขายสิทธิ์ขาด แต่ไม่สามารถจัดการดำเนินการคิดค่าบริการในลักษณะ PRO ที่ค่าตอบแทนจำเป็นต้องกระจายไปยังหลายฝ่ายได้ ERC 2981 ไม่ครอบคลุมประเด็นนี้
- ERC721 สามารถกำหนดได้เพียงเงื่อนไขในการโอนสิทธิ์ NFT แต่การกำหนดสกุลเงินในการโอนนั้นยังไม่มีการกำหนดได้ชัดว่าจะเป็น Utility Coin หรือ สกุลอื่นๆ ERC2981 ไม่ครอบคลุมประเด็นนี้
- ERC721 มีเงื่อนไขในการโอนสิทธิถาวรในลักษณะการ Transfer สิทธิ NFT แต่การเก็บค่าใช้จ่ายในการโอนสิทธินั้นรูปแบบที่เป็นมาตรฐานที่สามารถใช้ใน Music Business ได้นั้นยังไม่ได้มีการกำหนดตายตัว ERC2981 ไม่กล่าวถึงประเด็นนี้
- ERC721 นั้นมีการกำหนดว่าแต่ละ Token ต้องมี Metadata ของตนเอง แต่มาตรฐาน NFT Metadata ที่ตอบรับกับธุรกิจเพลงที่แต่ละเพลงมีเจ้าของและผู้เกี่ยวข้องที่ต่างกัน นอกจากนั้นแต่ละ Token ยังอาจมีมากกว่า 1 เพลงได้นั้นยังไม่มีการกำหนดมาตรฐาน ERC2981 ยังไม่ได้กล่าวถึงประเด็นนี้
- ERC721 ไม่สนับสนุนการจ่ายค่าตอบแทนในลักษณะ Royalty Fee แบบธุรกิจเพลง ERC2981 สามารถทำระบบ Royalty Fee ได้แต่รูปแบบการจ่ายนั้นไม่หลากหลายเพียงพอที่จะ Support การแบ่งส่วนแบ่งให้ผู้เกี่ยวข้องหลายฝ่ายได้

จากผลวิเคราะห์ แม้ว่า ERC721 และ ERC2981 จะเป็นตัวเลือกที่ใกล้เคียงในการนำมาใช้งานในการพัฒนาระบบ DMS แต่อย่างไรก็ตามนั้น ทั้ง ERC721 และ ERC2981 ยังไม่สามารถรองรับการทำธุรกิจ Music Streaming บนสถาปัตยกรรมแบบกระจายดังที่ DMS ต้องการได้ รายละเอียดการนำ ERC721 และ ERC2981 มาต่อยอดเพื่อพัฒนาเป็น DMS Contract นั้นแสดงในหัวข้อถัดไป

3.2 DMS Contract : A Smart Contract for Decentralized Music Streaming System

ในหัวข้อนี้จะกล่าวถึงการพัฒนาผ่านการเพิ่มเติมของ ERC721 และ ERC2981 เพื่อรวมกันเป็น DMS Contract. DMS Contract นั้นมีข้อเสนอว่าควรมีความสามารถของ NFT ดัง Inherit มาจาก ERC721 และ ERC2981 ทั้งหมดเพื่อให้ได้ความสามารถเดิมของระบบการทำ NFT และ Royalty Fee System เข้ามาเป็นพื้นฐาน หลังจากนั้นให้มีการเพิ่มเติมความสามารถฟังก์ชันทั้งหมดดังต่อไปนี้

- ความสามารถในการตั้งชื่อเพลง : DMS Contract จะต้องสามารถตั้งชื่อของเพลงหรืออัลบั้มเพลงได้ ผู้แต่งเพลงนั้นให้ทำการกำหนดโดยใช้ข้อมูลชนิด String หนึ่งชุด
- ความสามารถในการกำหนดค่าให้บริการ โดยค่าบริการนั้นต้องหนดได้ทั้งรูปแบบกรณีการเปิดไฟล์เป็นครั้งๆในรูปแบบ Royalty Fee หรือการโอนสิทธิถาวร
- ความสามารถในการเพิ่ม Contributor : สามารถเพิ่มรายนามผู้เกี่ยวข้องโดยแยกเป็นประเภท เช่น สามารถระบุได้ว่า เพลงในอัลบั้มนี้แต่งโดยใคร ใครเป็นผู้ประพันธ์ดนตรีเบื้องหลัง ใครเป็นผู้แต่งเนื้อร้อง ทำนอง เป็นต้น โดยใน DMS นั้นเสนอว่าโดยข้อมูล Contributor แต่ละคนนั้นควรประกอบด้วยข้อมูลดังต่อไปนี้
    - ชื่อ Contributor
    - ตำแหน่งที่เกี่ยวข้อง (เช่น ผู้แต่งคำร้อง นักร้อง นักดนตรี เป็นต้น)
    - เลขบัญชีในการโอนค่าตอบแทนในรูปแบบ Cryptographic Hash Wallet Address
    - สัดส่วนค่าตอบแทนที่ Contributor ควรได้ในแต่ละครั้งที่มีการจ่ายค่า Royalty
- กำหนดสกุลเงินที่ใช้ในการรับชำระค่าฟังเพลงได้ในรูปแบบว่าเป็น ERC20 แบบใด หรือเป็น Utility Token ได้
- มีการกำหนดมาตรฐาน Metadata ที่เหมาะสมที่ทำงานได้ทั้งแบบเพลงเดียวหรือแบบ Playlist กรณีเป็นอัลบั้ม โดยในงานนี้ได้เสนอว่าให้ Metadata นั้นมีลักษณะดังนี้



รูปที่ 3 DMS: Metadata JSON format

จากรูปที่ 3 เป็นรูปของการเก็บ Metadata ในกรณีที่เก็บไฟล์เป็นอัลบั้มจัดเก็บไฟล์เป็นรูปแบบ JSON โดยในรายละเอียดจะมีดังต่อไปนี้

- “musicName” เป็นการเก็บชื่อเพลง
- “songwriter” เป็นการเก็บผู้แต่งเพลง
- “Instrumental” เป็นการเก็บชื่อวงเพลง หรือ ชื่อผู้เล่นเครื่องดนตรี
- “Vocal lyrics” เป็นการเก็บเนื้อเพลง
- “description” เป็นการเก็บอธิบายรายละเอียดเพิ่มเติมของเพลง
- “music” เป็นการเก็บลิงค์ที่อยู่ของเพลง

รายการข้อมูลดังปรากฏในตัวอย่างนี้เป็นรายการต่อเพลง 1 เพลง กรณีเป็นอัลบั้มนั้นให้ผู้สร้าง Token สามารถเพิ่มข้อมูลชุด Data ดังกำหนดต่อหลัง “,” ในลักษณะของ JSON format ได้

- มีฟังก์ชัน “Listening to Music” สำหรับการฟังเพลงที่สามารถจ่ายค่า Royalty แก่ Contributor ตามสัดส่วนที่กำหนดได้ โดยการทำงานของ Listening to Music นั้นควรมีขั้นตอนดังต่อไปนี้
    - รับค่าเข้าเป็น Wallet ของผู้ที่ต้องการเปิดเพลง
    - ทำการจ่ายค่าตอบแทนแก่ Contributor ทุกคนตามราคาและสัดส่วนที่กำหนด
    - ส่งรายการไฟล์ที่สามารถเปิดได้ส่งคืนสู่ผู้ขอใช้ Contract เมื่อการชำระค่าตอบแทนเสร็จสิ้น

จากที่กล่าวมาแล้วทั้งหมด จะสังเกตได้ว่า DMS Contract นั้นในงานวิจัยนี้เสนอว่าควรทำการพัฒนาต่อยอดจาก ERC721 และ ERC2981 โดยเพิ่มฟังก์ชันทีครอบคลุมกับการใช้งาน Music Streaming Business ให้สมบูรณ์ยิ่งขึ้นเข้าไปใน DMS รายละเอียดการ Implement และ การทดลองใช้งาน DMS นั้นแสดงในบทถัดไป

บทที่ 4

การพัฒนา DMS Contract และการทดลอง

บทนี้จะกล่าวถึงการเขียนโค้ดเพื่อพัฒนา DMS Contract ในส่วนสำคัญ ถัดไปจะกล่าวถึงแผนการทดลอง ทดสอบและการประเมินประสิทธิภาพของ DMS

4.1. การพัฒนา DMS Contract

DMS Contract ในเวอร์ชันทดลองเพื่อการทดสอบและการวิจัยในงานวิจัยนี้นักพัฒนาโดยอาศัยเฟรมเวิร์คเพื่อการพัฒนา Smart Contract ที่ชื่อว่า OpenZeppelin เขียนโค้ดโดยอาศัยภาษา Solidity เวอร์ชัน 0.8.4 ส่วนสำคัญช่วงแรกของ DMS Contract เวอร์ชันทดลองนี้คือการออกแบบให้ DMS Contract นั้นทำการ Inherit คุณสมบัติดังต่อไปนี้จาก ERC แบบเดิม

- ERC721: เพื่อให้ได้ความสามารถของ ERC721
- ERC721URI Storage: เพื่อให้ได้ความสามารถการเก็บ Storage ของ ERC721 เป็น URI
- ERC721Burnable: เพื่อให้ Token สามารถมีการทำลายออกจากระบบได้หากต้องการ
- Ownable: เพื่อให้ Token ยังคงมีเจ้าของและเปลี่ยนเจ้าของได้
- โดยใช้เฟรมเวิร์คสืบทอดมาในลักษณะดังรูปต่อไปนี้

```
import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721Burnable.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/token/ERC20/IERC20.sol";

contract TestDMS is ERC721, ERC721URIStorage, ERC721Burnable, Ownable {
    constructor() ERC721("TestDMS", "TDMS") {}
```

*รูปที่ 4 **DMS: ERC import***

จากรูปที่ 4 สังเกตว่ายังไม่ได้มีการนำ ERC2981 มาใช้ใน DMS Contract เวอร์ชันทดลองเนื่องจากความสามารถในการจ่ายค่า Royalty Fee ใน ERC2981 นั้นจำเป็นต้องถูก

Override เพื่อแก้ไขขั้นตอนในการคำนวณ ดังนั้นงานนี้จึงเลือกที่จะเสนอฟังก์ชันในการคำนวณ Royalty แบบใหม่แทนการ Override ระบบงานเดิมเพื่อลดความซ้ำซ้อนและความจำเป็นในการ Override โดยความสามารถที่นำมาแทน ERC2981 ใน DMS คือฟังก์ชัน listeningMusic ดังต่อไปนี้

```
function listeningMusic(address _sender) payable public {
    //IERC20(OnlyToken).approve(_sender,Amount);
    //IERC20(OnlyToken).allowance(_sender, address(this));
    IERC20(OnlyToken).transferFrom(_sender, address(this), Amount);
    uint i = 0;
    for (i = 1;i <= contri.length; i++){
    Contributor storage _contri =(contri[i]);
      _contri.account.transfer(msg.value/_contri.percent);
    }
    _baseURI;
}
```

รูปที่ 5 Function listeningMusic

จากรูปที่ 5 ฟังก์ชัน listeningToMusic ดังแสดงนี้มีความสามารถหลักคือการที่ระบบสามารถใช้คำสั่ง transferFrom ที่สามารถทำการดึงเหรียญจากผู้ขอฟังเพลงมาแล้วหลังจากนั้นทำการแบ่งจ่ายให้กับ Contributor ได้ตามแต่เปอร์เซ็นต์จองแต่ละ Contributor ที่กำหนดมาไว้ก่อนหน้า

ฟังก์ชันอื่นๆที่มีการเพิ่มเติมได้แก่ฟังก์ชันกลุ่มการตั้งค่าชื่อเพลง Playlist Metadata ต่างๆ ตาม รายละเอียดโค้ดทั้งหมดที่ใช้ในการพัฒนา DMS นั้นแสดงอยู่ในภาคผนวก ก. ท้ายเล่มนี้

4.2 การ Deploy Contract และการวัดประสิทธิภาพผลการใช้งาน

DMS Contract ดังที่กล่าวมาแล้วทั้งหมดนั้นได้ถูกทำการ Implement และ Deploy ขึ้น Blockchain Network จริงโดยอาศัย Remix IDE ของ Ethereum Foundation เพื่อประเมินถึงความเป็นไปได้ในการใช้งานจริงอย่างมีประสิทธิภาพ การประเมินค่า Gas Fee ที่ใช้ในการ Deploy Contract นั้นเป็นสิ่งจำเป็น โดยค่า Gas Fee ที่ใช้ในการ Deploy DMS นั้นจะสื่อถึงว่าขนาดของข้อมูลและชุดคำสั่งที่ส่วนขยายของ DMS นั้นมีปริมาณที่ยอมรับได้หรือสูงเกินกว่าจะทำการรับได้จริง

ในส่วนของ Delay นั้นเนื่องจากระบบของ Ethereum Blockchain Network นั้นในตอนที่ทำการ Deploy Contract นั้นจะต้องรอตรวจสอบธุรกรรมจากผู้ตรวจสอบขั้นตอนนี้เรียกว่า pending ซึ่งเมื่อมีการเรียกใช้งานฟังก์ชันใน DMS ซ้อนกันจะเกิด pending ขึ้นทำให้เกิดอาการล่าช้าขึ้นได้ตามจำนวนของการ pending ซึ่งจำนวนที่มีผลต่อเครือข่าย Ethereum จนสามารถเปลี่ยนสถานะของเครือข่ายได้นั้นจะต้องมีถึง 1000 pending จากที่กล่าวมางานเพลงจากที่กำหนดใน DMS นั้นจะต้องจ่ายค่า Royalty ไปยังผู้มีส่วนร่วมกับเพลง ซึ่งงานนี้ไม่ได้มีการกำหนดไว้ว่าสามารถรับรองได้กี่คน ในงานทดลองได้มีการ pending อยู่ 30 รายการซึ่งไม่มีผลต่อสถานะเครือข่าย

เพื่อทำการทดลองการวัดค่า Gas Fee เนื่องด้วยค่า Gas Fee นั้นมีโอกาสผันผวนขึ้นอยู่กับช่วงว่าเครือข่ายมีการใช้อยู่ในปริมาณสูงหรือไม่ มี Validator Node ที่พร้อมรับ Validate Block ใหม่มากหรือน้อยเพียงใด ในงานวิจัยนี้เพื่อลดการผันผวนของความหนาแน่นของเครือข่ายต่อค่า Gas จึงได้เลือกทำการทดลองกับเครือข่ายสองแบบ ดังต่อไปนี้

- Ethereum Ropsten Test Network

เครือข่ายนี้เป็นเครือข่ายของ Ethereum Foundation สำหรับใช้ทดสอบความถูกต้องของ Smart Contract ใหม่ที่พัฒนาโดยนักพัฒนาอิสระ เครือข่ายนี้จะสื่อถึงการใช้งานในเครือข่ายที่ใกล้เคียง Production จริง แต่ยังคงมีความหนาแน่นที่น้อยกว่าใน Ethereum Production Main Network จริง ซึ่งเป็นรูปแบบเครือข่ายที่มีรูปแบบข้อตกลงเหมือนกับเครือข่ายหลักจึงเลือกเครือข่ายนี้เพื่อให้ได้ผลลัพธ์ใกล้เคียงกับเครือข่ายหลักมากที่สุด

- SciNET Sandbox

เครือข่าย SciNET เป็น Blockchain เฉพาะที่พัฒนาภายในห้องปฏิบัติการของคณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ เครือข่าย SciNET นี้ออกแบบเพื่อใช้ทำการทดลองเครือข่าย Ethereum สำหรับภายในห้องปฏิบัติการ โดย Network จะประกอบด้วย Validator Node แบบ Proof of Authority 2 Node เพียงเท่านั้นที่สร้างโดย Opensource ชื่อ geth เนื่องด้วยการที่ SciNET นี้มี Traffic ที่น้อยกว่า Network แบบ Public Chain อื่นๆมากจึงได้มีการทดลองวัดค่า Gas บน SciNET นี้ด้วย

การทดลองเพื่อวัดประสิทธิภาพและค่าใช้ Gas Fee ของ DMS นั้นมีขั้นตอนดังนี้

1. ทำการ Deploy DMS ลงบน Ropsten Test Network

2. ทำการทดลองเรียกใช้ฟังก์ชันดังต่อไปนี้ผ่าน Remix IDE 5 ครั้งเพื่อวัดค่าเฉลี่ยในการใช้ค่า Gas
    - setMusicName
    - addContributor
    - setServiceCharge
    - setPayToken
    - listeningMusic

3. ทำซ้ำขั้นตอนที่ 1 และ 2 โดยเปลี่ยนการทดลองจากบน Ropsten Test Network เป็น SciNET

4.2. ผลลัพธ์ที่ได้จากเครือข่ายทดสอบ

ตามที่อธิบายไว้ในส่วนก่อนหน้า เราปรับใช้ DMS ผ่านเครือข่าย Ropsten และ SciNET เพื่อวัด Gas Fee ผลลัพธ์จะได้รับในตารางที่ 3

ตารางที่ 3 *ผลการทดลอง: ค่าธรรมเนียมแก๊สที่จำเป็นสำหรับการดำเนินการ DMS*

ฟังก์ชัน	เครือข่ายทดสอบ	
	Ropsten	SciNET
setMusicName	0.00005094	0.00004447
addContributor	0.00013869	0.00012892
setServiceCharge	0.00004944	0.000042
setPayToken	0.00004944	0.00004383
listeningMusic	0.00004946	0.00002733

จากตารางที่ 3 ให้ค่าเฉลี่ยของการทดลองใช้ค่า Gas Fee โดยเป็นค่าเฉลี่ยของการเรียกใช้ฟังก์ชันครั้งที่จำเป็นสำหรับการทำสัญญาสำคัญห้าครั้ง ซึ่งต้องมีการดำเนินการค่าธรรมเนียมใน DMS ของเรา ในกรณีของเครือข่าย Ropsten เราทดลองและวัดผลลัพธ์ในช่วงเวลาที่เครือข่ายไม่มีการโต้แย้งเพื่อหลีกเลี่ยงความผันผวนของค่าธรรมเนียมก๊าซเนื่องจากขาดเครื่องมือตรวจสอบฟรี

ผลการทดลอง พบว่า แม้ว่าการ Deploy บน Ropsten Test Network และ SciNET จะมีผลค่า Gas ที่แตกต่างกันเล็กน้อย แต่ว่าแนวโน้มของค่า Gas ที่ใช้นั้นอยู่ในปริมาณที่ใกล้เคียงกัน กล่าวคือการทำงานของ DMS นั้นสามารถทำการใช้งานได้โดยที่ Contract ไม่มีขนาดที่มากหรือสูงเกินกว่าที่จะทำให้ค่าใช้งานบนเครือข่ายมีค่าบริการที่แพงเกินกว่าความจำเป็น

บทที่ 5

สรุปผลงานวิจัย ปัญหา อุปสรรค และ แผนการทดลองต่อยอด

5.1. สรุปผลการวิจัย

งานวิจัยชิ้นนี้ได้กล่าวถึงปัญหาและความต้องการในการทำการให้บริการ Music Streaming แบบกระจายโดยอาศัยเครือข่าย Blockchain เพื่อการดังกล่าว งานวิจัยนี้ได้เสนอสถาปัตยกรรมที่มีภาพรวมว่า (1) ระบบควรทำงานโดยอาศัย NFT เป็นหลักในการขับเคลื่อน และ (2) ต้นแบบของ Contract ที่เหมาะสมกับการใช้งานที่ต่อยอดจาก NFT มาสำหรับธุรกิจ Music Streaming โดยเฉพาะที่ชื่อว่า Decentralized Music System (DMS) Contract

ใน DMS และ ทุกระบบที่อิงตามสัญญาอัจฉริยะ โดยไม่คำนึงถึงโดเมนของแอปพลิเคชัน ค่าธรรมเนียมเป็นสิ่งสำคัญสำหรับความสำเร็จของระบบ ขนาดสัญญาที่ใหญ่ขึ้นและผู้ใช้สัญญาจำนวนมากอาจนำไปสู่ความยากลำบากในการค้นหาผู้ตรวจสอบความถูกต้องเพื่อรับและดำเนินการธุรกรรม ในงานของเรา จะเห็นได้ว่า DMS ใช้ค่าน้ำมันเพียงเล็กน้อยทั้งใน Ropsten และเครือข่าย SciNET ที่ควบคุมในห้องปฏิบัติการของเราภายใต้เงื่อนไขในสถานะเครือข่ายที่เสถียร จะเห็นได้ว่าฟังก์ชันการตั้งค่าส่วนใหญ่มีค่าธรรมเนียมคล้ายกัน ยกเว้นการเพิ่มผู้มีส่วนร่วมในลิขสิทธิ์เพลง เพื่อรวบรวมข้อมูลต่าง ๆ เกี่ยวกับผู้เกี่ยวข้องและนำไปเรียกเก็บค่าลิขสิทธิ์ ในการปรับใช้ DMS ในทางปฏิบัติ นอกจากการออกแบบและการปรับใช้สัญญาแล้ว จำเป็นต้องทำสัญญาในตลาดกลาง นอกจากนี้ การที่เราเสนอให้เพิ่มการทำงานไปยัง ERC721 และ ERC2981 ที่มีอยู่ หมายความว่าการใช้งานกระเป๋าเงินคริปโตเคอเรนซีที่มีอยู่ จะไม่รองรับ DMS ของเราตั้งแต่แรก อาจจำเป็นต้องเสนอการใช้งาน DMS แบบกำหนดเองบนกระเป๋าเงินดิจิตอลที่มีอยู่ ทั้งสองด้านดังกล่าวจะเป็นทิศทางโฟกัสของเราในการทำงานในอนาคต

5.2. ปัญหาและอุปสรรคในการวิจัย

เนื่องจาก ในการทดสอบตัวแปรเปลี่ยนแปลงที่สำคัญในงานนี้คือ สถานะปัจจุบันของเครือข่ายทดสอบอย่าง Ropsten ซึ่งเมื่อจะเทียบผลลัพธ์กับ เครือข่ายทดสอบที่สร้างขึ้นมาเองอย่าง SciNET แล้วจะต้องมีสถานะในขณะนั้นอยู่ที่ เสถียร (stable) จึงจะสามารถเปรียบเทียบได้ ฉะนั้นจึงต้องใช้เวลาทดสอบที่มากขึ้นเพื่อติดตามผลของสถานะเครือข่ายในขณะนั้น

5.3. ข้อเสนอแนะ และ ทิศทางการดำเนินงานต่อในอนาคต

งานนี้เกี่ยวข้องกับการใช้สถาปัตยกรรมของแพลตฟอร์มการสตรีมเพลงแบบกระจายอำนาจโดยใช้บล็อกเชน มีการเสนอการออกแบบเบื้องต้นของสถาปัตยกรรม DMS สัญญาที่กำหนดเองของเราที่เขียนการทำงานเพิ่มเติมจาก ERC721 และ ERC2981 เพื่อรองรับตรรกะทางธุรกิจการสตรีมเพลงได้รับการพัฒนา สัญญา DMS ของเราผ่านการตรวจสอบผ่านการวัดค่าธรรมเนียม ผลการศึกษาพบว่า ค่าน้ำมันที่ต้องจ่ายจากการเพิ่มเติมของเรานั้นไม่แพงและนำไปปฏิบัติได้จริง ในการปรับใช้การศึกษาของเราในทางปฏิบัติ ยังคงต้องมีการออกแบบและนำเสนอส่วนหน้าของตลาดและกระเป๋าเงิน cryptocurrency แบบกำหนดเอง ทั้งสองด้านจะเน้นไปที่งานในอนาคตของเรา

นอกจากนี้ ฟังก์ชันส่วนใหญ่เพื่อความยืดหยุ่นในการเขียนโค้ดเพื่อให้สามารถปรับแต่งได้ง่ายจึงไม่ได้เขียนการจำกัดความมากน้อยในการตั้งค่าต่างๆไว้เช่น จำนวนของผู้เกี่ยวข้องกับงานเพลง ซึ่งอาจจะมีผลต่อสถานะเครือข่าย ไม่ได้ทดสอบไปมากกว่า 30 รายการ จึงยังไม่ทราบว่าฟังก์ชันที่เกี่ยวข้องกับ ผู้ที่เกี่ยวข้องนั้น ได้มีการใช้ค่าธรรมเนียมเปลี่ยนแปลงไปหรือไม่ ทำให้งานนี้ไม่เหมาะกับเพลงแนว Symphonic Orchestra/Choir เพราะไม่อยู่ในขอบเขตเบื้องต้นของงานวิจัยซึ่งเน้นที่การแก้ไขปัญหาระบบงานแบบเก่าและทำให้กระจายค่าลิขสิทธิ์ไปยังผู้เกี่ยวข้องได้ ซึ่งหากทำให้สามารถกระจายค่าลิขสิทธิ์ด้วยวิธีอื่นได้หรือทำให้สามารถลดการใช้ค่าธรรมเนียมที่ใช้ในการตั้งค่าได้ซึ่งสามารถต่อยอดต่อไปได้ในงานวิจัยในอนาคต

บรรณานุกรม

[1] Mark Mulligan, *music streaming Reports from MIDiA Research*. midiaresearch, 2021. Accessed: Jan. 31, 2022. [Online]. Available: https://www.midiaresearch.com/reports/category/music-streaming

[2] D. Vujičić, D. Jagodić, and S. Randić, "Blockchain technology, bitcoin, and Ethereum: A brief overview," *2018 17th International Symposium on INFOTEH-JAHORINA, INFOTEH 2018 - Proceedings*, vol. 2018-January, pp. 1–6, Apr. 2018, doi: 10.1109/INFOTEH.2018.8345547.

[3] H. Dai, Z. Zheng, Y. Z.-I. I. of T. Journal, and undefined 2019, "Blockchain for Internet of Things: A survey," *ieeexplore.ieee.org*, Accessed: Jan. 31, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8731639/

[4] L. Kugler, "Non-fungible tokens and the future of art," *Commun ACM*, vol. 64, no. 9, pp. 19–20, Sep. 2021, doi: 10.1145/3474355.

[5] S. Hong, Y. Noh, and C. Park, "Design of extensible non-fungible token model in hyperledger fabric," *SERIAL 2019 - Proceedings of the 2019 3rd Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers*, pp. 1–2, Dec. 2019, doi: 10.1145/3366611.3368142.

[6] J. Singh and P. Singh, "Distributed Ownership Model for Non-Fungible Tokens," *Smart and Sustainable Intelligent Systems*, pp. 307–321, Mar. 2021, doi: 10.1002/9781119752134.CH22.

[7] M. A. Hassan and Z. Shukur, "Review of Digital Wallet Requirements," *2019 International Conference on Cybersecurity, ICoCSec 2019*, pp. 43–48, Sep. 2019, doi: 10.1109/ICOCSEC47621.2019.8970996.

[8] G. A. Pierro and H. Rocha, "The influence factors on ethereum transaction fees," *Proceedings - 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain, WETSEB 2019*, pp. 24–31, May 2019, doi: 10.1109/WETSEB.2019.00010.

[9] G. W.-E. project yellow paper and undefined 2014, "Ethereum: A secure decentralised generalised transaction ledger," *files.gitter.im*, Accessed: Jan. 31, 2022. [Online]. Available: https://files.gitter.im/ethereum/yellowpaper/VIyt/Paper.pdf

[10] E. Erturk, M. Dogan, U. Kadiroglu, and E. Karaarslan, "NFT based Fundraising System for Preserving Cultural Heritage: Heirloom," pp. 699–702, Oct. 2021, doi: 10.1109/UBMK52708.2021.9559006.

[11] K. B. Muthe, K. Sharma, and K. E. N. Sri, "A Blockchain Based Decentralized Computing and NFT Infrastructure for Game Networks," *2020 2nd International Conference on Blockchain Computing and Applications, BCCA 2020*, pp. 73–77, Nov. 2020, doi: 10.1109/BCCA50787.2020.9274085.

[12] Y. Chen, H. Li, K. Li, and J. Zhang, "An improved P2P file system scheme based on IPFS and Blockchain," *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, vol. 2018-January, pp. 2652–2657, Jul. 2017, doi: 10.1109/BIGDATA.2017.8258226.

[13] J. Benet, "IPFS - Content Addressed, Versioned, P2P File System," Jul. 2014, Accessed: Mar. 04, 2022. [Online]. Available: https://arxiv.org/abs/1407.3561v1

[14] S. Zhao and D. O'mahony, "BMCProtector: A Blockchain and Smart Contract Based Application for Music Copyright Protection," *Proceedings of the 2018 International Conference on Blockchain Technology and Application - ICBTA 2018*, 2018, doi: 10.1145/3301403.

[15] M. Marescotti, M. Blicha, A. E. J. Hyvärinen, S. Asadi, and N. Sharygina, "Computing Exact Worst-Case Gas Consumption for Smart Contracts," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11247 LNCS, pp. 450–465, 2018, doi: 10.1007/978-3-030-03427-6_33.

[16] W. Chen, Z. Zheng, J. Cui, E. Ngai, P. Zheng, and Y. Zhou, "Detecting ponzi schemes on ethereum: Towards healthier blockchain technology," *The Web*

Conference 2018 - Proceedings of the World Wide Web Conference, WWW 2018, vol. 4, pp. 1409–1418, Apr. 2018, doi: 10.1145/3178876.3186046.

[17] "Music Streaming Application using Blockchain | IEEE Conference Publication | IEEE Xplore." https://ieeexplore.ieee.org/abstract/document/8991275 (accessed Mar. 02, 2022).

[18] A. Kim and M. Kim, "A Study on Blockchain-based Music Distribution Framework: Focusing on Copyright Protection," *International Conference on ICT Convergence*, vol. 2020-October, pp. 1921–1925, Oct. 2020, doi: 10.1109/ICTC49870.2020.9289184.

[19] M. O'Dair and R. Owen, "Monetizing new music ventures through blockchain: Four possible futures?:," *https://doi.org/10.1177/1465750319829731*, vol. 20, no. 4, pp. 263–276, Feb. 2019, doi: 10.1177/1465750319829731.

[20] K. Chi and H. Kim, "The Impact of Blockchain Technology on the Music Industry," *International journal of advanced smart convergence*, vol. 8, no. 1, pp. 196–203, 2019, doi: 10.7236/IJASC.2019.8.1.196.

[21] Z. Cai, "Usage of deep learning and blockchain in compilation and copyright protection of digital music," *IEEE Access*, vol. 8, pp. 164144–164154, 2020, doi: 10.1109/ACCESS.2020.3021523.

[22] B. Shrestha, M. N. Halgamuge, and H. Treiblmaier, "Using Blockchain for Online Multimedia Management: Characteristics of Existing Platforms," pp. 289–303, 2020, doi: 10.1007/978-3-030-44337-5_14.

[23] H. Li, "Piano automatic computer composition by deep learning and blockchain technology," *IEEE Access*, vol. 8, pp. 188951–188958, 2020, doi: 10.1109/ACCESS.2020.3031155.

[24] I. Adjei-Mensah, I. O. Agyemang, C. Sey, L. D. Fiasam, and A. A. Salako, "Securing music sharing platforms: A Blockchain-Based Approach," Oct. 2021, doi: 10.48550/arxiv.2110.05949.

[25] A. Gomaa, "A DRM Solution for Online Content Using Blockchain - A Music Perspective," *SSRN Electronic Journal*, Dec. 2018, doi: 10.2139/SSRN.3351542.

[26**]** C. Adjovu and E. Fabian, **"**Blockchain-mediated Licensing**:** Legal Engineering for Artist Empowerment,**"** *SSRN Electronic Journal*, Jun. 2020, doi**:** 10.2139/SSRN.3625317.

[27**]** W. Gao and G. Zhang, **"**Research on decentralized music sharing model based on consortium blockchain,**"** *Communications in Computer and Information Science*, vol. 1122 CCIS, pp. 613–624, 2019, doi**:** 10.1007/978-981-15-1301-5_48/COVER/.

[28**]** Z. Guochao, T. Huayun, C. Jianhai, S. Rui, H. Qinming, and H. Butian, **"**Digital music copyright management system based on blockchain,**"** *Journal of Computer Applications*, vol. 41, no. 4, p. 945, Apr. 2021, doi**:** 10.11772/J.ISSN.1001-9081.2020111731.

[29**]** Zach Burks, James Morgan, Blaine Malone, and James Seibel, **"**EIP-2981**:** NFT Royalty Standard.**"** https**://**eips.ethereum.org/EIPS/eip-2981 **(**accessed Jun. 12, 2022**).**

[30**]** L. Peng, W. Feng, Z. Yan, Y. Li, X. Zhou, and S. Shimizu, **"**Privacy preservation in permissionless blockchain**:** A survey,**"** *Digital Communications and Networks*, vol. 7, no. 3, pp. 295–307, Aug. 2021, doi**:** 10.1016/J.DCAN.2020.05.008.

[31**]** R. Lai and D. Lee Kuo Chuen, **"**Blockchain – From Public to Private,**"** *Handbook of Blockchain, Digital Finance, and Inclusion*, vol. 2, pp. 145–177, Jan. 2018, doi**:** 10.1016/B978-0-12-812282-2.00007-3.

[32**]** Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, **"**Consortium blockchain for secure energy trading in industrial internet of things,**"** *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3690–3700, Aug. 2018, doi**:** 10.1109/TII.2017.2786307.

[33**]** L. S. Sankar, M. Sindhu, and M. Sethumadhavan, **"**Survey of consensus protocols on blockchain applications,**"** *2017 4th International Conference on Advanced Computing and Communication Systems, ICACCS 2017*, Aug. 2017, doi**:** 10.1109/ICACCS.2017.8014672.

[34**]** B**.** Chase and E**.** MacBrough, **“**Analysis of the XRP Ledger Consensus Protocol,**”** Feb**.** 2018, doi**:** 10**.**48550/arxiv**.**1802**.**07242**.**

[35**]** W**.** Zou *et al***.**, **“**Smart Contract Development**:** Challenges and Opportunities,**”** *IEEE Transactions on Software Engineering*, vol**.** 47, no**.** 10, pp**.** 2084–2106, Oct**.** 2021, doi**:** 10**.**1109/TSE**.**2019**.**2942301**.**

[36**]** T**.** Roughgarden, **“**Transaction Fee Mechanism Design for the Ethereum Blockchain**:** An Economic Analysis of EIP**-**1559,**”** Dec**.** 2020, doi**:** 10**.**48550/arxiv**.**2012**.**00854**.**

ภาคผนวก

ภาคผนวก ก.

Source code ของสัญญา DMS

สามารถหาข้อมูลได้ทางออนไลน์ผ่านลิงค์ต่อไปนี้

https://github.com/Maste12etrun/DMS

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721Burnable.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/token/ERC20/IERC20.sol";

contract TestDMS is ERC721, ERC721URIStorage, ERC721Burnable, Ownable {
    constructor() ERC721("TestDMS", "TDMS") {}

    function safeMint(address to, uint256 tokenId, string memory uri)
        public
        onlyOwner
    {
        _safeMint(to, tokenId);
        _setTokenURI(tokenId, uri);
    }

    function _burn(uint256 tokenId) internal override(ERC721, ERC721URIStorage) {
        super._burn(tokenId);
    }
```

```
function tokenURI(uint256 tokenId)
    public
    view
    override(ERC721, ERC721URIStorage)
    returns (string memory)
{
    return super.tokenURI(tokenId);
}

//ลิงค์เพลงของสัญญานี้
function _baseURI() internal pure override returns (string memory) {
    return "https://ipfs.io/ipfs/QmUFdYEH934SuVtHKcvUYfDHqFQiwvToJQzNLqvsM51Tnk";
}
////////////////////////////////////////////////////////////////////////////////////////////////////
// การคิดค่าบริการเมื่อมีผู้ต้องการฟังเพลงจาก contract
// ตั้งค่าว่าในการเข้ามาฟังเพลงต้องใช้สกุลเงินอะไรในการแลกเปลี่ยน
////////////////////////////////////////////////////////////////////////////////////////////////////

//เป็น String ที่แสดงถึงชื่อเพลง
string musicName;
function setMusicName(string memory name) public onlyOwner {
    musicName = name;
}
//ดูชื่อเพลงที่ตั้งไว้
function getMusicName() public view returns (string memory) {
    return musicName;
}

//รายละเอียดการเก็บข้อมูลของผู้มีส่วนร่วมในลิขสิทธิ์
struct Contributor {
    string name;
    string position;
    address payable account;
    uint percent;
}

//ดูรายละเอียดของผู้มีส่วนร่วมในผลงานเพลงโดยใส่เลขเริ่มจาก 0
Contributor[] public contri;
```

```
//เพิ่มผู้มีส่วนร่วมในลิทสิทธิ์เพลง
//การคิดค่าเปอร์เซ็นต์ส่วนแบ่งของภาษา Solidity เบื้องต้น
//100; == 1 Percent
//500; == 5 percent
//2000; == 20 Percent
function addContributor(string memory _name,string memory _position,
        address payable _account,uint _percent) public onlyOwner {
    Contributor memory newCon = Contributor({
        name : _name,
        position : _position,
        account : _account,
        percent : _percent
    });
    contri.push(newCon);
}

//ใช้ในการแก้ไขรายละเอียดของผู้มีส่วนร่วมในเพลง
function editContributor(uint assetId,string memory _name,string memory _position,
        address payable _account,uint _percent) public onlyOwner{
    Contributor storage _contri =(contri[assetId]);
    _contri.name = _name;
    _contri.position = _position;
    _contri.account = _account;
    _contri.percent = _percent;
}
   //ใช้เพื่อดูจำนวนผู้มีส่วนร่วมในลิขสิทธิ์เพลงว่ามีกี่คน
   function recordsArrayLenth() public view returns (uint256){
      return contri.length;
   }
/*
   //ดูรายละเอียกของผู้มีส่วนร่วมในผลงานเพลง
   function getContributor(uint assetId) public view returns (string memory,address,uint) {
      return (contri[assetId].name,contri[assetId].account,contri[assetId].percent);
   }
*/
   event Received(address, uint);
   receive() external payable {
      emit Received(msg.sender, msg.value);
   }
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////
// การคิดค่าบริการเมื่อมีผู้ต้องการฟังเพลงจาก contract
// ตั้งค่าว่าในการเข้ามาฟังเพลงต้องใช้อะไรบ้างเช่น ค่าบริการ หน่วยเงินหรือ ercที่ใช้แลกเปลี่ยน
////////////////////////////////////////////////////////////////////////////////////////////////////

//เจ้าของสัญญา
address Owner;
function setOwner(address _owner) public onlyOwner {
   Owner = _owner;
}
//ค่าบริการในการเข้ามาฟังเพลง
uint Amount;
function setServiceCharge(uint _amount) public {
   Amount = _amount;
}
//ตั้งค่าให้รับเฉพาะ ERC20 ที่กำหนด
address OnlyToken;
function setToken(address _token) public onlyOwner {
   OnlyToken = _token;
}

 struct someToken {
    string symbol;
    address payable token;
 }
 someToken[] public token;
 //เพิ่ม ERC20 ที่สามารถใช้ได้
 function addToken(string memory _sym,address payable _token) public onlyOwner {
    someToken memory newToken = someToken({
       symbol : _sym,
       token : _token
    });
    token.push(newToken);
 }
```

```
//ใช้เพื่อส่ง ERC20 มายังสัญญานี้
//และเมื่อมีการส่ง ERC20 มาแล้วตามกำหนดจะส่ง URL ที่เป็นที่อยู่ของเพลงให้
function listeningMusic(address _sender) payable public {
   //IERC20(OnlyToken).approve(_sender,Amount);
   //IERC20(OnlyToken).allowance(_sender, address(this));
   IERC20(OnlyToken).transferFrom(_sender, address(this), Amount);
   uint i = 0;
   for (i = 1;i <= contri.length; i++){
   Contributor storage _contri =(contri[i]);
      _contri.account.transfer(msg.value/_contri.percent);
   }
   _baseURI;
}

//ใช้เพื่อดู ERC20 ที่มีอยู่
//function getBalanceToken(address _address) public view returns (unit) {
//    return ERC20(_address).balanceOf(address(this));
//}

// ตั้งมูลค่าของเพลงเพื่อทำการขายเพลงไปยังบัญชีอื่นถาวร
// จำนวนแสดงถึง ERC20 ที่ต้องจ่าย
uint SalePrice;
function setSalePrice (uint _salePrice) public {
   SalePrice = _salePrice;
}
```

```
//ตั้งค่าลิขสิทธิ์ที่ได้จากการขายขาด
uint feeDenominator;
function _feeDenominator(uint _fee) public {
    feeDenominator = _fee;
}

struct RoyaltyInfo {
    address receiver;
    uint96 royaltyFraction;
}

RoyaltyInfo private _defaultRoyaltyInfo;
mapping(uint256 => RoyaltyInfo) private _tokenRoyaltyInfo;

function royaltyInfo(uint256 _tokenId) public view returns (address, uint256) {
    RoyaltyInfo memory royalty = _tokenRoyaltyInfo[_tokenId];
    if (royalty.receiver == address(0)) {
        royalty = _defaultRoyaltyInfo;
    }
    uint256 royaltyAmount = (SalePrice * royalty.royaltyFraction) / feeDenominator;
    return (royalty.receiver, royaltyAmount);
}
//ขายขาดเพลง
function sellOutMusic(address _receiver, address payable _token) public view returns (string memory) {
   royaltyInfo;
   ERC721(_token).transferFrom(Owner, _receiver);
}
```

ภาคผนวก ข.

บทความตีพิมพ์

Conference : ITC-CSCC 2022 “The 37th international Technical Conference on Circuit/Systems, Computers and Communications ”

วันที่ 5-8 กรกฎาคม 2565 ณ Duangjitt Resort & Spa จังหวัดภูเก็ต

DMS : An Architecture of a Decentralized-based Music Streaming Platform using Blockchain.

Suppakit Yamwaja
Division of Computational Science, Faculty of Science,
Prince of Songkla University, Songkhla, Thailand
Email: 6210220024@psu.ac.th

Chinnapong Angsuchotmetee
Division of Computational Science, Faculty of Science,
Prince of Songkla University, Songkhla, Thailand
Email: chinnapong.a@psu.ac.th

***Abstract*—Managing royalty fees in the music business is the task that is done by record labels, music retailers, and performance-right organization firms. Royalty fees are mostly distributed in a centralized manner. A centralized-based approach may be straightforward for managing a music business. However, a centralized-based approach prevents artists to be able to set their revenue model for their music and getting paid promptly. To allow an artist to have freedom in setting their revenue model, a decentralized-based approach may need to be proposed. In this work, we propose to apply Blockchain and Smart Contract in the music business to propose a decentralized-based music streaming service that allows artists can independently design their royalty fee model and get paid promptly. Our proposal is based on an extension of ERC20 and ERC721 standards on writing a smart contract. Our proposal is deployed and tested in a private Ethereum-based blockchain in a laboratory-controlled environment to determine the feasibility and gas fee required. The result suggests that the business logic required in our contract does not consume a high amount of gas fee. Hence, it can be deemed that our contract design is suitable for deploying in an actual decentralized-based music streaming service.**

***Index Terms*—Keywords- Blockchain ; decentralized ; music streaming ; smart contracts**

I. Introduction

Nowadays, music streaming becomes a mainstream online service. People prefer an online subscription service over buying physical copies of music albums. Music streaming users are growing rapidly. said at least 523.9 million people, an increase of 26.4% over the same period of 2020 [1]. Well-known services nowadays include Spotify, Apple iTunes, Google Music, etc.

Although the music streaming system has greatly grown the music business. But the exclusivity of each platform also requires artists and their associated parties to agree to the terms of their respective payouts, which may be delayed or proportional to the artist's non-negotiable compensation. The mechanisms and contracts for compensation to those involved in the music business are shown in Figure 1

Fig. 1. Mechanisms and compensation contracts in the music business of the legacy system

To resolve the monopoly of platforms using centralized architecture in the aforementioned manner The technology designed to solve these problems is blockchain technology, which has three key features: (1) the written data cannot be changed (2) all parties in the system are jointly owned by the system. together There is no central authority to maintain the system and (3) there is a Smart Contract system that allows system users to write any programs for users of the system to run together. At present, many systems were originally centralized systems, moved to use in distributed architecture instead, such as BitSong, Musicoin, ROCKI, OPUS, etc.

Blockchain allows musicians to easily approve and manage their music copyright with distributed ledger technology. Rights-holders can automatically and immediately receive a share of royalties from the music industry, even if no broker is involved in the distribution process. to deal with piracy issues using encryption and watermarking methods. However, it has been mentioned in [2] that introduce a versioning mechanism to update the data in the smart contract for correcting mistakes made by the publisher, By distributing and managing

978-1-6654-8559-3/22/$31.00 © 2022 IEEE

music using the proposed model, can provide all transaction information and related tasks in the music market safely and transparently.

All of the above The main goal of this research is to propose a blockchain and smart contract implementation to deliver music streaming systems based on a distributed architecture. Our study is based on an Ethereum-based network which means that your proposal is compatible with Ethereum and its derivative (e.g., Binance). Our proposal is named *Decentralized Music System* (DMS) contract. DMS contract is written in solidity language. DMS contract along with our proposed architecture design can help an artist get paid for their music without having to go through a central authority.

This paper is organized as follows. Section 2 describes the related works. Section 3 introduces the design of our proposed contract. Section 4 describes the deployment and validation of our contract in our test-bed. The result is given in Section 5. The discussion, conclusion, and future work are described in Section 6 and Section 7.

II. Related Studies

Studies that are related to our work can be categorized into 2 groups which are (i) Application of Blockchain in the music business and (ii) Ethereum Request for Comment (ERC) standard. They are described as follows.

A. Application of Blockchain in Music Business

Studies in this group are mostly aimed to apply Blockchain in managing music copyright. A significant study in this field is in [2]. This study focuses on proposing a new music copyright protection scheme that not only assures the rights holder's income stream but also considers the protection and tracks the copyright itself after a song has been purchased. this work concludes that the right of artists to reproduce, perform and distribute a musical work. However, there are still some issues of concern in the digital music industry, including rights and licensing management, levying fees from intermediaries, and piracy problems. This is particularly important because many countries around the world have published related laws and rules to help protect this kind of right, but there are still some business and political problems that may block users from taking part. For example, some companies do not want to lose control of the process. Not every artist wants to disclose their income to the public. Other similar works in [3] and [4] also yielded a similar conclusion. To date, public blockchain-based music platforms, through providing aligned incentives, radically innovate the digital music business towards a decentralized and artist-driven business model and urge established organizations to follow suit.it can solve the limitations of the current music copyright monopoly management and can accept the newly emerged various types of works in an expanded form, but gas consumption [5] and delay have not yet been investigated.

B. Ethereum Request for Comment (ERC) standard

Deploying a new contract over the Ethereum network or its derivative requires programming a smart contract in Solidity language. A solidity-based [6] contract over the Ethereum network is considered to be *Turing complete* which means that any kind of business logic can be programmed into the network without any limitation. To avoid having too many types of contracts over the network, the community of Ethereum developers gathered and proposed a set of standard contract codes so-called *Ethereum Request for Comment (ERC)*. One of the most famous ERC standards is the ERC20 [7]. ERC20 is used as an *Initial Coin Offering (ICO)* for most businesses on the Ethereum network. Another important ERC is the ERC721 [8]. ERC721 is designed for offering a *Non-Fungible Token (NFT)* to Ethereum users. An NFT is a token that is associated with a piece of asset, mostly an image. An NFT can be used for representing ownership of a piece of an asset over Blockchain network. The concept of NFT through ERC721 is later extended to ERC2981 [9] which is designed for collecting royalty fees for the usage of a given NFT.

Due to the fact that ERC721 and ERC2981 are not designed and designated for the music business, they may not be able to cover every aspect of the requirement of proposing a decentralized-based music business despite the fact that it can already be used for modeling ownership of a single music asset. The requirements of a decentralized-based music business that we identified, compared with ERC721, and ERC2981 features are described in Table I.

TABLE I
Requirement of a decentralized-based music business against ERC721 and ERC2981 feature.

Requirement	Feature Coverage	
	ERC721	ERC2981
Set the song title, details of the artist, composer, record label, etc.	Partial	Not covered
Set up service charge Listening to music from the contract.	Partial	Not covered
Set a song's value in order to sell it to another account.	Partial	Not covered
Set up contracts to include songs or playlists.	Partial	Not covered
Set the ERC20 currency which symbol will be used for the payout.	Partial	Not covered
Set up an allocation of percentages earned from music service fees and paid music royalty payments to music contributors.	Not covered	Partial

III. Prosposal : DMS Architecture & DMS Contract

In this study, we propose an architecture of a decentralized-based music streaming platform over a blockchain network through the usage of the Ethereum and Smart Contract mechanism. The architecture and contract are both named *Decentralized-based Music Streaming (DMS)*. The proposed architecture is depicted in Figure 2.

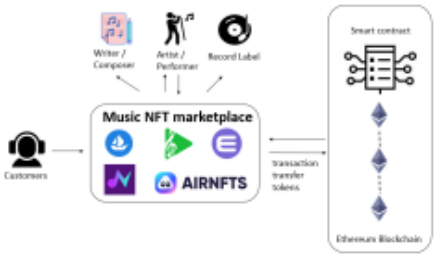


Fig. 2. DMS : Architecture Design

Figure 2 depicts that our proposal *DMS Platform* are based on an idea of a simple NFT-based marketplace. However, the major difference lies in the usage of our own DMS contract which can help an artist to be able to set their own payment scheme without relying on PROs. Our contract is named *DMS*.

DMS contract is developed through extending ERC721 and ERC2981. We choose to base our implementation on ERC721 and ERC2981 in order to ensure that our contracts are still compatible with the existing NFT-based system. The extension that we propose are (i) 'setMusicName' for setting details of the artist on music NFT asset (ii) 'addContributor' function was used to add variables definition into a struct that holds any variable such as name, position, account, and percentage.T (iii) 'set service charge' for setting up payment model (iv) 'addPayToken'and 'free denominator' for setting up an allocation of percentages earned from music service fees and paid music royalty payments to music contributors and (v) extension metadata for *baseURI* property of ERC721 to support music playlist, in the case of providing music album service. The metadata extension is depicted as follows.

```
{
    {"musicName": "......",
    "song-writer": "....",
    "Instrumental": "....",
    "Vocal lyrics": "....",
    "description": "....",
    "music": "......"
    },
    .
    .
    .
    },
```

Another essential extension that DMS proposed is the *listeningToMusic* function.This function takes all parameters and payment schemes that users set to pay all contributors promptly for each music access. The function is implemented as followed.

```
function listeningMusic(address _sender)
    payable public {
    IERC20(OnlyToken).
        transferFrom(_sender,
            address(this),
            Amount);
    uint i = 0;
    for (i = 1;i <= contri.length; i++){
        Contributor storage _contri
            =(contri[i]);
        _contri.account.
            transfer(msg.value/
                _contri.percent);
    }
    _baseURI;
}
```

The function *listeningMusic* uses IERC20 token adapter to ensure that the payment scheme to contributors can support any kind of ERC20-based token. The function works by looping through the list of contributors that are added from *addContributor* function beforehand and paying all of them before returning baseURL back to the music playback request. Each contributor is defined as a struct as follows.

```
struct Contributor {
    string name;
    string position;
    address payable account;
    uint percent; }
```

As given above, each contributor is defined using four main values which are (i) contributor name (ii) their position (iii) payable account, and (iv) percentage that they can gain for each playback. The source code and further description of DMS can be found online[1].

IV. Validation: DMS Gas Fee Measurement

In every Ethereum-based blockchain network, the *Gas Fee* reflects the size of a smart contract and computational power required for executing the contract. The fee has to be paid by using the native utility coin of a given network. For example, in the Ethereum network, the fee has to be paid in ETH, while Binance requires the fee to be paid in BSC. In this work, we propose a new contract design. Hence, ensuring the fee is minimal, while

[1]https://github.com/Maste12etrun/DMS

covering all the requirements of music business logic is essential. Therefore, we choose gas fee measurement as a metric for validating our study.

To validate our study, we deploy DMS onto two Ethereum networks which are (i) the Ropsten test network of Ethereum and (ii) our laboratory-controlled SciNET, a private ethereum network deployed using two GETH validator nodes. Testing on Ropsten represents the actual usage when the network is deployed in the large scale network while deploying on SciNET represents the measurement of the gas fee over a low-contention network case. We deploy DMS and execute our contract to both networks five times. The average gas fees on both Ropsten and SciNET are measured.

V. Experiment Results

As described in the previous section, we deploy DMS over Ropsten and SciNET network to measure the gas fee of our contract execution. The results are given in Table II

TABLE II
EXPERIMENT RESULT: GAS FEE REQUIRED FOR EXECUTING DMS

function	Test Network	
	Ropsten)	SciNET
setMusicName	0.00005094	0.00004447
addContributor	0.00013869	0.00012892
setServiceCharge	0.00004944	0.000042
setPayToken	0.00004944	0.00004383
listeningMusic	0.00004946	0.00002733

Table II gives the average of five trials of the gas fee required for executing five essential contracts which require a gas fee to be executed in our DMS. In the Ropsten network case, we experiment and measure the result during the period in which networks do not have contention to avoid the gas fee fluctuation due to the lack of a free validator. The result shows that our extension from ERC721 and ERC2981 to propose DMS does not make a contract to be too complicate and too expensive to the point that gas fees are not feasible to be used.

VI. Discussion

In every smart contract-based system, regardless of the application domain, a gas fee is essential for the success of the system itself. Larger the contract size and a large number of users of the contract itself can lead to the difficulty in finding a validator to pick up and execute the transaction [10].

In our work, it can be seen that DMS consumes very little gas fee in both Ropsten and our laboratory-controlled SciNET network. It can be seen that most of the setup functions have similar gas fees, except for adding contributors to the music copyright. to collect various information about the contributors and bring to charge royalties. To deploy DMS in practice, besides the design and deployment of the contract, the marketplace front-end is needed to be signed. Also, the fact that we propose an extension to an existing ERC721 and ERC2981 means that the existing implementation of a cryptocurrency wallet [11] will not support our DMS natively. A custom implementation of DMS onto an existing cryptocurrency wallet may need to be proposed. Both aspects as mentioned will be our focus direction of the future work.

VII. Conclusion and Future Work

This work is concerned with implementing an architecture of a decentralized-based music streaming platform using blockchain. The preliminary design of DMS architecture is proposed. Our custom contract programmed by extending from ERC721 and ERC2981 to support music streaming business logic is developed. Our DMS contract is validated through gas fee measurement. The study reveals that the gas fees required from our extension are not expensive and feasible to be used in practice.

References

[1] M. Mulligan, *music streaming Reports from MIDiA Research*. midiaresearch, 12 2021.

[2] S. Zhao and D. O'mahony, "Bmcprotector: A blockchain and smart contract based application for music copyright protection," *Proceedings of the 2018 International Conference on Blockchain Technology and Application - ICBTA 2018*, 2018.

[3] A. Kim and M. Kim, "A study on blockchain-based music distribution framework: Focusing on copyright protection," *International Conference on ICT Convergence*, vol. 2020-October, pp. 1921–1925, 10 2020.

[4] A. Torbensen and R. Ciriello, "Tuning into blockchain: Challenges and opportunities of blockchain-based music platforms," 6 2019.

[5] M. Marescotti, M. Blicha, A. E. Hyvärinen, S. Asadi, and N. Sharygina, "Computing exact worst-case gas consumption for smart contracts," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11247 LNCS, pp. 450–465, 2018.

[6] M. Wohrer and U. Zdun, "Smart contracts: Security patterns in the ethereum ecosystem and solidity," *2018 IEEE 1st International Workshop on Blockchain Oriented Software Engineering, IWBOSE 2018 - Proceedings*, vol. 2018-January, pp. 2–8, 3 2018.

[7] F. Vogelsteller and V. Butterin, "Erc20: A standard interface for tokens," *EIP: Ethereum Improvement Proposals*, 2015.

[8] D. S. William Entriken (@fulldecent), "Eip-721: Non-fungible token standard," Jan 2018.

[9] J. M. j. Zach Burks (@vexycats), "Eip-2981: Nft royalty standard," Sep 2020.

[10] G. A. Pierro and H. Rocha, "The influence factors on ethereum transaction fees," *Proceedings - 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain, WETSEB 2019*, pp. 24–31, 5 2019.

[11] M. A. Hassan and Z. Shukur, "Review of digital wallet requirements," *2019 International Conference on Cybersecurity, ICoCSec 2019*, pp. 43–48, 9 2019.

ประวัติผู้เขียน

ชื่อ สกุล นายศุภกฤต แย้มวาจา

รหัสประจำตัวนักศึกษา 6210220024

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)	มหาวิทยาลัยราชภัฏภูเก็ต	2562

การตีพิมพ์เผยแพร่ผลงาน

S. Yamwaja and C. Angsuchotmetee. 2022. DMS: An Architecture of a Decentralized-based Music Streaming Platform using Blockchain,ITC-CSCC 2022 “The 37th international Technical Conference on Circuit/Systems, Computers and Communications, Duangjitt Resort & Spa, Kathu, Phuket,Thailand