แอปพลิเคชันชนิดกระจายอำนาจการควบคุมสำหรับการจัดเก็บทรัพย์สินทางปัญญาอย่างถาวรและปลอดภัยด้วยเทคโนโลยีบล็อกเชนและไอพีเอฟเอส

A Decentralized Application (DApp) for Securely and Permanently Storing Intellectual Property Using Blockchain Technology and IPFS

นางสาวดลนภา ฉิมสอาด

Dolnapa Chimsa-ard

นางสาวธมลวรรณ รังผึ้ง

Tamonwan Rangpueng

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยศรีนครินทรวิโรฒ ปีการศึกษา พ.ศ. 2561

คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ชื่อหัวข้อโครงงาน แอปพลิเคชันชนิดกระจายอำนาจการควบคุมสำหรับการจัดเก็บทรัพย์สินทางปัญญาอย่างถาวรและปลอดภัยด้วยเทคโนโลยีบล็อกเชนและไอพีเอฟเอส

A Decentralized Application (DApp) for Securely and Permanently Storing Intellectual Property Using Blockchain Technology and IPFS

นิสิต

นางสาวดลนภา	ฉิมสอาด	58102010806
นางสาวธมลวรรณ	รังผึ้ง	58102010809

ปริญญา **วิทยาศาสตรบัณฑิต (วท.บ.)**

ภาควิชา **วิทยาการคอมพิวเตอร์**

อาจารย์ที่ปรึกษาโครงงาน **ผู้ช่วยศาสตราจารย์ ดร. ศุภชัย ไทยเจริญ**

ลงชื่อ.....

(ผู้ช่วยศาสตราจารย์ ดร. ศุภชัย ไทยเจริญ)

Abstract

Blockchain is a decentralized digital ledger system. It allows different organizations and individuals to exchange information online securely and anonymously without any trusted third party. A key characteristic of blockchain technology is that data once successfully stored in the blockchain network is almost completely unalterable. Moreover, since an exchange of information in the blockchain network does not rely on trusted third parties, the performance of business processes can be improved with higher efficiency and lower processing costs. Various domains have been successfully taken advantage of this technology such as business, medicine, and manufacturing. In this paper, an application of blockchain technology to the event management industry, specifically academic and professional conference events, is presented. The proposed system is implemented as a decentralized application (DApp) based on Ethereum platform. Its goals are to facilitate communications between conference event organizers and attendees and provide channels for all involved parties to collaborate securely and efficiently. The core functionalities of the current version of the system include secure online registration, user authentication, article submission, and reviewers' evaluation. Some useful features for future versions are secure online payment, on-site verification, and attendee tracking.

สารบัญ

หัวข้อ **หน้า**

สารบัญตาราง

หัวข้อ **หน้า**

สารบัญรูปภาพ

หัวข้อ **หน้า**

สารบัญรูปภาพ

หัวข้อ **หน้า**

บทที่ 1
บทนำ

1.1. ที่มาและความสำคัญของโครงาน

สิทธิในทรัพย์สินทางปัญญานั้นจะแตกต่างจากสิทธิหรือการเป็นเจ้าของในสิ่งที่เป็นผลผลิตทางทรัพย์สินทางปัญญา เช่น ลิขสิทธิ์ในบทความวิจัยจะไม่ใช่เป็นสิ่งเดียวกันกับความเป็นเจ้าของบทความวิจัยซึ่งจับต้องได้ สิทธิบัตรในเรื่องอิเล็กทรอนิกส์จะแยกต่างหากจากความเป็นเจ้าของเครื่องมืออิเล็กทรอนิกส์ ดังนั้นเจ้าของบทความวิจัยหรือเครื่องมืออิเล็กทรอนิกส์จึงมีกรรมสิทธิ์ในการใช้บทความวิจัยหรือจัดการบทความวิจัยนั้นตามความประสงค์ แต่ไม่สามารถทำการใดๆ ซึ่งละเมิดต่อสิทธิของเจ้าของบทความวิจัยนั้น เช่น เจ้าของบทความวิจัยจะไม่สามารถทำบทความวิจัยขึ้นมาจำหน่ายเองโดยปราศจากความยินยอมของเจ้าของลิขสิทธิ์หรือผู้เขียนบทความวิจัย เนื่องจากสิทธิในการทำซ้ำเป็นสิทธิทางกฎหมายแต่เพียงผู้เดียวของของลิขสิทธิ์หรือผู้เขียนบทความวิจัยนั้น หรือผู้ซื้อซอร์ฟแวร์จะเป็นเจ้าของสินค้านี้เพื่อการนำไปใช้ประโยชน์ต่อ แต่จะไม่ได้รับอนุญาตให้ทำซอร์ฟแวร์นั้นขึ้นมาจำหน่ายเอง เว้นแต่ได้รับอนุญาตจากของลิขสิทธิ์หรือผู้เขียนบทความวิจัยก่อนเท่านั้น

ทรัพย์สินทางปัญญาจึงเป็นสิ่งผู้เขียนมีกรรมสิทธิ์ในการใช้หรือจัดการตามความประสงค์ของผู้เขียน หากมีผู้ใดทำการละเมิดสิทธิในการทำซ้ำหรือเผยแพร่โดยไม่ได้รับอนุญาต จะถือว่าความผิดทางกฎหมาย เว้นแต่จะรับการอนุญาตจากผู้เขียน

ในปัจจุบันเทคโนโลยี Blockchain ถูกนำมาใช้อย่างกว้างขวางในการจัดการสกุลเงิน Bitcoin หรือสกุลเงินในโลกดิจิทัลอื่นๆโดยสามารถเรียกว่า Cryptocurrency ซึ่งนอกจาก Bitcoin แล้วยังมีสกุลเงินอื่นๆ เช่น Ethereum, Ripple และ Zcash ที่เป็นที่รู้จักอย่างแพร่หลาย โดย Bitcoin และเทคโนโลยี Blockchain ไม่ใช่สิ่งเดียวกัน แต่มีความสัมพันธ์กันเป็นอย่างมาก เพราะเทคโนโลยี Blockchain จะทำงานเป็นเบื้องหลังของ Bitcoin ซึ่งจะถูกนำมาใช้เพื่อเพิ่มความปลอดภัยและความน่าเชื่อถือของข้อมูล

เนื่องจากเทคโนโลยี Blockchain มีความสามารถในการจัดเก็บข้อมูลได้อย่างปลอดภัย ทำให้เราต้องการนำเทคโนโลยี Blockchain มาใช้ในการจัดเก็บบทความวิจัย เพื่อเพิ่มความน่าเชื่อถือของบทความวิจัยว่าบทความวิจัยนี้ได้รับการอนุญาตจากผู้เขียนบทความวิจัยเรียบร้อยแล้ว สามารถนำไปประยุกต์ใช้ในด้านธุรกิจ เช่น ด้านธุรกิจเพลง ด้านอุตสาหกรรมการผลิต เป็นต้น

1.2. วัตถุประสงค์ของโครงงาน

- ศึกษาแนวคิดและหลักการทำงานของ Blockchian Technology
- ศึกษาหลักการทำงานของ IPFS
- ศึกษาวิธีการผนวกรวม Blockchain Technology เข้ากับ Web Application เพื่อพัฒนาเป็น Decentralized Application (DApp)
- เพื่อประยุกต์ใช้และพัฒนา Decentralized Application (DApp) สำหรับการจัดการการเข้าถึงบทความทางวิชาการ

1.3. ขอบเขตของโครงงาน

- ผู้ใช้งานสามารถเข้าสู่ระบบผ่าน MetaMask ได้
- ผู้ใช้งานสามารถกรอกรายละเอียดของบทความและบันทึกข้อมูลลงบล็อกเชนได้
- ผู้ใช้งานสามารถอัปโหลดบทความผ่าน IPFS ได้
- พัฒนาเป็น Web application ที่สามารถรันบนเว็บได้

1.4. ประโยชน์ที่คาดว่าจะได้รับ

- ได้ศึกษาและเข้าใจหลักการทำงานของ Blockchian Technology มากขึ้น
- ได้ศึกษาวิธีการพัฒนาแอปพลิเคชันด้วยภาษา Solidiity
- ได้ศึกษาและเรียนรู้เครื่องมือต่างๆที่ใช้ในการพัฒนา เช่น Truffle frameworks เป็นต้น
- ได้แอปพลิเคชันสำหรับการจัดการการเข้าถึงบทความทางวิชาการด้วย Blockchian Technology และ IPFS
- ผู้ใช้งานสามารถเข้าสู่ระบบผ่าน MetaMask ได้
- ผู้ใช้งานสามารถกรอกรายละเอียดของบทความและบันทึกข้อมูลลงบล็อกเชนได้
- ผู้ใช้งานสามารถอัปโหลดไฟล์บทความผ่าน IPFS ได้

บทที่ 2

การทบทวนวรรณกรรม

ตัวอย่างงานวิจัยที่เกี่ยวข้อง

1. **Bitcoin: A Peer-to-Peer Electronic Cash System**

   ผู้แต่ง: Satoshi Nakamoto

   บทความนี้กล่าวถึงการทำงานแบบ Peer-to-Peer โดยในบทความนี้ได้กล่าวถึงการชำระเงินออนไลน์โดยตรงจากฝ่ายหนึ่งไปอีกฝ่ายหนึ่งโดยไม่ต้องผ่านสถาบันทางการเงิน โดยนำลายเซ็นดิจิทัลเป็นส่วนหนึ่งในทางแก้ปัญหา

   เนื้อหาของบทความได้กล่าวถึงการทำงานโดยไม่ต้องผ่านบุคคลที่สามเพื่อตัดปัญหาการทำงานซ้ำซ้อน ซึ่งจะมีการพูดถึงการแลกเปลี่ยนของธุรกรรมต่างๆ ซึ่งเรียกว่า Transaction ซึ่งการเกิด Transaction ในแต่ละครั้งข้อมูลจะถูกบันทึกลงใน Block โดยเราสามารถถึงผู้ส่งจากลายเซ็นดิจิทัลและหลักการทำงานของการทำงานแบบ Peer-to-Peer โดยมีการบอกรายละเอียดในแต่ละบล็อกนั้นๆว่ามีการทำงานอย่างไร

   วิธีการดำเนินงานจะเน้นเป็นการอธิบายการทำงานแบบ Peer-to-Peer เป็นส่วนใหญ่ และมีการประเมินประสิทธิภาพของการทำงานแบบไม่ผ่านบุคคลที่สามว่ามีประสิทธิภาพดีแค่ไหน

2. **Open Peer-to-Peer Systems over Blockchain and IPFS: an Agent-Oriented Framework**

   ผู้แต่ง: Antonio Tenorio-Fornes, Samer Hassan, JuanPavon

   บทความนี้กล่าวถึงการทำงานระหว่าง IPFS กับ Blockchain ซึ่งเป็นรูปแบบที่มีอยู่ของ Peer-tp-Peer

   เนื้อหาของบทความนี้ได้กล่าวถึงการทำงานของ IPFS และ Blockchain ซึ่งเป็นการเข้าถึงข้อมูลและการไว้วางใจของข้อมูล IPFS เป็นนวัตกรรมของการจัดเก็บข้อมูลที่ไม่สามารถแก้ไข้ข้อมูลได้ จึงทำให้ได้นำ IPFS และ Blockchain มาผสมผสานกัน

3. **Building an Ethereum and IPFS-based Decentralized Social Network System**

ผู้แต่ง: Quanqing Xu, Zhiwen Song, Rick Siow Mong Goh, Yongjun Li

บทความนี้กล่าวถึงการพัฒนา DApps คือ แอปพลิเคชันที่มีการอ้างอิง Ethereum

เนื้อหาของบทความได้กล่าวถึงการจัดเก็บข้อมูลแบบกระจายอำนาจเพื่อลดต้นทุนของพื้นที่ของฮาร์ดแวร์ จึงได้นำ IPFS ขึ้นมาเพื่อแก้ไขปัญหานี้ โดย IPFS มีการเก็บข้อมูลแบบกระจาย และเก็บข้อมูลแบบถาวร จึงประยุกต์นำ IPFS และ Blockchain สำหรับ Social Network นั่นคือ Twitter

วิธีการดำเนินงานของบทความนี้คือ การอัพโหลดรูปภาพผ่าน IPFS และภาพจะถูกโพสต์บน Twitter โดยภาพจะถูกแปลงค่า Hash และจึงนำค่า Hash มาบันทึกลง Blockchain แทน

4. **Commercial Property Tokenizing With Smart Contracts**

ผู้แต่ง: Alex Norta, Chad Fernandez, Stefan Hickmott

บทความนี้กล่าวถึงการซื้อขายอสังหาริมทรัพย์เชิงพาณิชย์ โดยใช้ Smart Contract มาเกี่ยวข้อง เพื่อลดปัญหาของพ่อค้าคนกลาง

โดยเนื้อหาของบทความกล่าวถึงการพัฒนาแอปพลิเคชันแบบไม่รวมศูนย์ สำหรับการทำธุรกรรมแบบ Peer-to-Peer โดยเป็นแอปพลิเคชันสำหรับการเปิดใช้งานการซื้อขายอสังหาริมทรัพย์เชิงพาณิชย์โดยใช้ Blockchain

ส่วนวิธีการดำเนินงานของบทความนี้คือ การที่สร้าง Smart Contract เพื่อสร้างเงื่อนไขข้อตกลงของการตกลงซื้อขายอสังหาริมทรัพย์เชิงพาณิชย์ โดยจะทำการส่ง Token เมื่อมีการซื้อขาย

5. **MedRec: Using Blockchain for Medical Data Access and Permission Management**

ผู้แต่ง: Asaph Azaria, Ariel Ekblaw, Thiago Vieira and Andrew Lippman

บทความนี้กล่าวถึงการจัดเก็บข้อมูลทางการแพทย์บนเทคโนโลยีบล็อกเชนเพื่อทำให้ข้อมูลทางการแพทย์เกิดความน่าเชื่อถือ ข้อมูลโปร่งใสและสามารถรักษาความปลอดภัยข้อมูลทางการแพทย์

โดยเนื้อหาของบทความเป็นไอเดียการทำเว็บแอปพลิเคชัน โดยการนำเสนอการจัดเก็บข้อมูลบนเทคโนโลยีบล็อกเชนลงใน Ethereum โดยนำ Smart Contract มาใช้งานกับเว็บแอปพลิเคชันเพื่อเพิ่มความปลอดภัยของข้อมูลทางการแพทย์มากขึ้น

ส่วนวิธีการดำเนินงานเน้นการพัฒนาโครงสร้างหลักๆ ในส่วนของการจัดเก็บข้อมูลทางการแพทย์บนเทคโนโลยีบล็อกเชน

6. **Integrating Blockchain for Data Sharing and Collaboration in Mobile Healthcare Applications**

ผู้แต่ง: Xueping Liang, Juan Zhao, Sachin Shetty, Jihong Liu1, Danyi Li

บทความนี้กล่าวถึงการประยุกต์ใช้เทคโนโลยีบล็อกเชนสำหรับการแลกเปลี่ยนข้อมูลสุขภาพส่วนบุคคลและการร่วมมือกันจัดการข้อมูลสุขภาพส่วนบุคคลระหว่างผู้ใช้งาน โรงพยาบาล และบริษัทประกันภัยบนสมาร์ทโฟน

โดยเนื้อหาของบทความจะพัฒนาแอปพลิเคชันบนสมาร์ทโฟน ด้วย Hyperledger Fabric ซึ่งถูกออกแบบมาให้เป็น Distributed Ledger ระหว่างองค์กร โดยแต่ละองค์กรอาจจะมีข้อมูลบางอย่างที่ไม่สามารถแชร์ให้ผู้ที่ไม่เกี่ยวข้องรับรู้ได้ ทำให้ผู้ใช้งานสามารถมั่นใจได้ว่าข้อมูลสุขภาพส่วนบุคคลที่แชร์ออกไปนั้นมีความปลอดภัยและสามารถตรวจสอบได้ มีการกำหนดสิทธิการเข้าถึงของข้อมูล เช่น โรงพยาบาลสามารถอัพเดตข้อมูลฝั่งของโรงพยาบาลได้ และยังสามารถเรียกดูข้อมูลที่เกี่ยวข้องในฝั่งของบริษัทประกันภัยได้แต่ไม่สามารถทำการอัพเดตข้อมูลฝั่งบริษัทประกันภัยได้

ส่วนวิธีการทำงานผู้ใช้งานจะสามารถใส่ข้อมูลสุขภาพส่วนบุคคลลงในแอปพลิเคชันของตนเองได้ โดยผู้ใช้งานจะเป็นศูนย์กลางในการกำหนดสิทธิการเข้าถึงข้อมูลต่างๆ เพื่อทำการซิงค์ข้อมูลไปยังบล็อกเชนแล้ว จะไม่สามารถทำการแก้ไขข้อมูลได้ ทำให้สามารถป้องกันการแอบแก้ไขข้อมูลภายหลังได้ จึงทำให้ข้อมูลมีความน่าเชื่อถือ ปลอดภัย และสามารถตรวจสอบได้

7. **Blockchain-based Personal Health Data Sharing System Using Cloud Storage**

ผู้แต่ง: Xiaochen Zheng, Raghava Rao Mukkamala, Ravi Vatrapu, Joaqun Ordieres-Mere

บทความนี้กล่าวถึงการแลกเปลี่ยนข้อมูลสุขภาพส่วนบุคคล โดยใช้เทคโนโลยีบล็อกเชนและทำการนำข้อมูลสุขภาพส่วนบุคคลที่ได้จากอุปกรณ์ที่สวมใส่และอุปกรณ์เคลื่อนที่สามารถเก็บข้อมูลทางสุขภาพ เช่น สมาร์ทโฟน, Apple Watch และ Smart Watch ขึ้นอยู่บน Cloud Storage

โดยเนื้อหาของบทความกล่าวถึงการเก็บข้อมูลทางสุขภาพส่วนบุคคล จากอุปกรณ์ที่สามารถเก็บข้อมูลทางสุขภาพส่วนบุคคลได้ และยังมีการกำหนดสิทธิการเข้าถึงข้อมูล ทำให้เจ้าของข้อมูลสุขภาพสามารถควบคุมได้ด้วยตนเองและปลอดภัยยิ่งขึ้น

ส่วนวิธีการดำเนินงานของบทความนี้ได้มีการจำแนกข้อมูลสุขภาพส่วนบุคคลออกเป็นหมวดหมู่ต่างๆตามลักษณะข้อมูลเช่น เพศ, กรุ๊ปเลือด, ลายนิ้วมือ เป็นต้น และวิธีการเก็บข้อมูลนั้นจากบทความนั้นได้เสนอการเก็บจากอุปกรณ์ที่สวมใส่และอุปกรณ์เคลื่อนที่แล้วจึงนำข้อมูลที่ได้มาเก็บรักษาบนบล็อกเชนและ Cloud Storage

8. **Design of A Blockchain-based Lottery System for Smart Cities Applications**

ผู้แต่ง : Da-Yin Liao, XueHong Wang

บทความนี้กล่าวถึงการออกแบบระบบลอตเตอรี่ด้วยเทคโนโลยีบล็อกเชนสำหรับแอปพลิเคชันเมืองอัจฉริยะ เพื่อแก้ปัญหาการทุจริตในการซื้อขายลอตเตอรี่

โดยเนื้อหาของบทความกล่าวถึงการนำ Smart Contract ของ Ethereum และรูปแบบการเข้ารหัสของ Blockchain มาใช้ในการออกแบบระบบลอตเตอรี่ที่มีชื่อว่า FairLotto

ส่วนวิธีการดำเนินการของบทความนี้คือ การออกแบบระบบลอตเตอรี่บนแอปพลิเคชัน เพื่อให้มุ่งเน้นให้คุณภาพชีวิตของประชาชนดีขึ้น และสร้างความมั่นใจ ความโปร่งใสของการชำระเงิน การออกตั๋ว โดยที่ไม่ผ่านบุคคลที่สาม

บทที่ 3
ทฤษฎีและแนวคิดที่ใช้ในการทำโครงงาน

3.1. องค์ความรู้เกี่ยวกับเทคโนโลยีบล็อกเชน (Blockchain Teachnology)

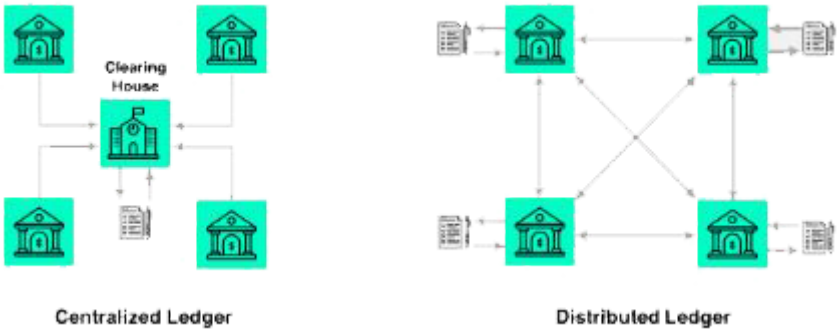
Blockchain คือ รูปแบบการจัดการเก็บข้อมูล (Database) อย่างหนึ่งของระบบที่ไม่มีศูนย์กลาง ซึ่งข้อมูลเหล่านั้นจะอยู่ ในกล่องขนาดเดียวกันเป็นบล็อก (Block) เชื่อมต่อกันเป็นโซ่ (Chain) หรือที่เรียกว่า Blockchain โดยวิธี Hash Function เป็นการทำข้อมูลให้ย่อลงแต่มีลักษณะจำเพาะของข้อมูลนั้น เปรียบเสมือนลายนิ้วมือของข้อมูลที่ใช้ในการ Verify ยืนยันความถูกต้องตามข้อกำหนด หรือกฎที่ตั้งไว้ ข้อมูลจะถูกกระจายไปยังที่อยู่ต่างๆในระบบ หากมีข้อมูลที่สร้างใหม่จะต้องได้รับการเห็นชอบจากผู้ใช้คนอื่นๆ ในห่วงโซ่ผ่านข้อตกลงที่มีรวมกันก่อนหน้านี้ และจะมีการตรวจสอบเพื่อให้เกิดความเชื่อมั่น การทำงานของ Blockchain จึงเหมือนกับการให้ทุกคนถือเอกสารที่มีข้อมูลชุดเดียวกันตรวจสอบกันเองได้เสมอ เมื่อมีการอัปเดตก็จะอัปเดตด้วยกัน โดยใช้เทคโนโลยีมาควบคุมทำให้ระบบมีความโปร่งใส ตรวจสอบง่าย ยากต่อการโกงจนสามารถมั่นใจได้ว่า ข้อมูลเหล่านั้นเชื่อถือได้มีความแน่นอน ไม่มีการปลอมแปลง มีความปลอดภัยและความถูกต้องสูง

ภาพที่ 3-1 ลักษณะการทำงานของ Blockchain
(ที่มา: https://www.aware.co.th/ลักษณะการทำงานของ-blockchain/)

เดิมเทคโนโลยีการบันทึกข้อมูล (Data) จะเป็นการบันทึกในที่เดียวแบบรวมศูนย์ (Centralized Ledger) ซึ่งหมายถึงว่า ผู้ที่ต้องการจะใช้ข้อมูลที่บันทึกไว้ที่ระบบกลางนี้ต้องเชื่อมั่นและไว้ใจในระบบกลางว่าเก็บเฉพาะข้อมูลที่เป็นจริง ระบบกลางบันทึกไว้ว่าอย่างไรสิ่งนั้นก็ถือว่าเป็นสิ่งที่เกิดขึ้นจริง

อย่างไรก็ตามเทคโนโลยี Blockchain จะต่างออกไปโดยจะเปลี่ยนรูปแบบจาก Centralize Ledger มาเป็น Distributed Ledger เพราะการบันทึกข้อมูลแบบกระจายศูนย์ (Distributed Ledger) ที่ไม่มีตัวกลางโดยที่ผู้ที่เกี่ยวข้องทุกคน (Peer) หรือผู้ที่ต้องการเก็บการทำรายการจะเก็บบันทึกข้อมูลต่างๆ ไว้ด้วยตนเองและช่วยกันตรวจสอบยืนยันและทำสำเนาข้อมูลเก็บไว้ ไม่ใช่เพียงโดยผู้ใดผู้หนึ่งเหมือนอย่างระบบรวมศูนย์

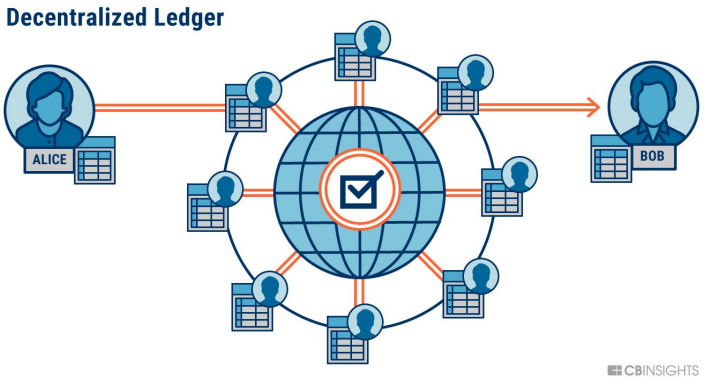
โดยข้อมูลใน Blockchain จะไม่มีใครสามารถเปลี่ยนแปลง แก้ไขหรือลบได้ หากไม่ได้รับความยินยอมจากสมาชิกทุกคนในเครือข่าย แม้แต่กระทั่งการจะเจาะระบบทุกคนเพื่อล้วงเข้าไปเปลี่ยนแปลงข้อมูลก็ไม่สามารถทำได้ เนื่องจากจะต้องใช้ขุมกำลังคอมพิวเตอร์มหาศาลจนไม่คุ้มค่ากับเวลาและทรัพยากรที่ลงไป นอกจากนั้นข้อมูลที่บันทึกไว้โดย Blockchain นี้ สามารถเชื่อมต่อแลกเปลี่ยนกัน (Distributed) ได้แบบ Peer-to-Peer โดยไม่มีศูนย์กลาง



ภาพที่ 3-2 ลักษณะความแตกต่างระหว่างการบันทึกข้อมูลแบบ Centralized Ledger และการบันทึกข้อมูลแบบ Distributed Ledger
(ที่มา: https://tradeix.com/distributed-ledger-technology/)

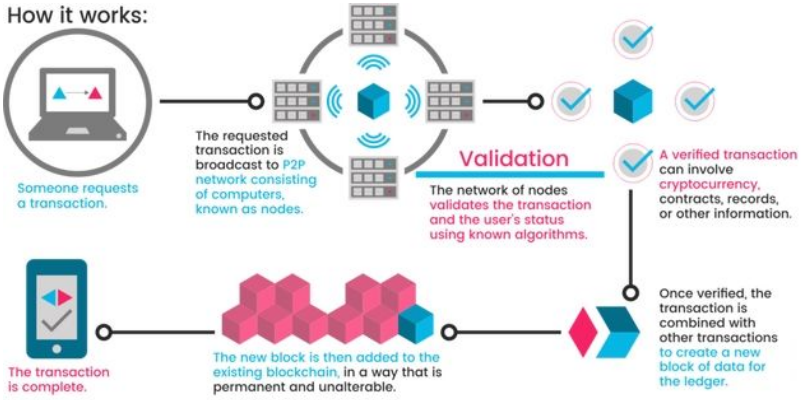
การที่ข้อมูลถูกบันทึกแบบ Distributed Ledger หรือรายการเดินบัญชีแบบกระจายตัวที่ผู้ใช้ทุกคนจะต้องมีเป็นของตัวเอง และมีการเชื่อมต่อกันแลกเปลี่ยนมีการยืนยันความถูกต้องร่วมกันแบบนี้ ส่งผลให้มีความโปร่งใสของข้อมูลและสร้างความเชื่อมั่นแบบ Distributed Trust ซึ่งจะช่วยลดการฉ้อโกงและข้อผิดพลาด

แต่ในโครงสร้างของ BlockChain จะมีส่วนสำคัญคือ Node ที่เชื่อมระหว่างกันจำนวนมากและแต่ละ Node จะมี Ledger ของผู้ใช้ทุกคนในเครือข่าย โดยแสดงผลผ่าน Address ที่ไม่ระบุตัวตน ทำให้ทุกคนในเครือข่ายสามารถเห็นการเดินบัญชีทางการเงินของผู้ใช้รายอื่นๆทั้งหมดแต่ไม่สามารถระบุได้ว่าแต่ละ Address คือผู้ใช้รายใด



ภาพที่ 3-3 ลักษณะการส่งข้อมูลแบบ Distributed Ledger
(ที่มา: https://www.cbinsights.com/research/what-is-blockchain-technology/)

พื้นฐานของ Blockchain จึงกำเนิดสกุลเงินดิจิทัล เรียกว่า Cryptocurrency (เงินดิจิทัล) ซึ่งใน Cryptocurrency นี้ก็มีสกุลเงินดิจิทัลในระบบกว่า 1,400 สกุล โดยมีสกุลเงินดิจิทัลยอดฮิต ได้แก่ Bitcoin, Ethereum, Ripple, Litecoin และ Cardano เงินดิจิทัลเหล่านี้ ไม่สามารถจับต้องได้เหมือนเหรียญหรือธนบัตร มูลค่าสกุลของเงินดิจิทัลในตลาดขึ้นอยู่กับความต้องการและปริมาณในตลาด และที่สำคัญมีบันทึกทุกการใช้จ่ายอยู่ในระบบโดยใช้เทคโนโลยี Blockchain เข้ามาควบคุมตรวจสอบบันทึกข้อมูล เห็นได้ชัดว่า Bitcoin นั้นไม่ใช่ Blockchain ในปัจจุบันนี้มีบางประเทศยอมรับให้ใช้จ่ายเงินสกุลดิจิทัลสามารถนำไปแลกเปลี่ยนเป็นเงินจริงได้หรือซื้อสินค้าได้จริง เช่น ที่ Big Camera ในประเทศญี่ปุ่น Subway ในประเทศสหรัฐอเมริกา สามารถใช้สกุลเงิน Bitcoin ได้ และ Burger King ที่รัสเซียใช้เงินสกุล Whopper Coin เป็นต้น แต่ในประเทศไทยยังไม่เป็นที่ยอมรับเท่าไหร่นัก



ภาพที่ 3-4 ลักษณะการทำงานบน Blockchain
(ที่มา: https://dzone.com/articles/blockchain-solutions-how-to-transform-your-busines)

จากภาพที่ 3-4 แสดงให้เห็นถึงลักษณะการทำงานเมื่อผู้รับต้องการทำธุรกรรมบางอย่างธุรกรรมที่ปลายทางจะถูกประกาศลงบนเครือข่ายข้อมูล ซึ่งเรียกว่า Node เป็นคอมพิวเตอร์สำหรับการเก็บข้อมูลทาง

ธุรกรรมนี้ จากนั้น Node จะทำการตรวจสอบธุรกรรมและสถานะต่างๆของผู้ใช้งาน โดยการใช้ Algorithm ต่างๆในการตรวจสอบธุรกรรมนี้ เมื่อทำการตรวจสอบเรียบร้อยธุรกรรมจะอยู่ในรูปแบบ Cryptocurrency หรือสัญญาของธุรกรรมต่างๆ ต่อมาข้อมูลในธุรกรรมก็จะถูกนำข้อมูลบันทึกลงไว้ใน Block บน Blockchain เพื่อเป็น Block สำหรับข้อมูลใหม่ ซึ่งข้อมูลใน Block ก็จะคงอยู่แบบนี้ตลอดไป จะไม่สามารถถูกดัดแปลงหรือแก้ไขข้อมูลได้ จะทำลักษณะแบบนี้เรื่อยๆ เมื่อเกิดข้อมูลใหม่ ซึ่งในเครือข่าย Blockchain ของ Bitcoin ตั้งอยู่บนพื้นฐานของความเห็นพ้องต้องกันของทุกฝ่าย (Consensus) จะมีการตรวจสอบข้อมูลในทุกๆ 10 นาที ระบบจะตรวจสอบธุรกรรมที่เกิดขึ้นในช่วงระยะเวลา 10 นาทีนี้ การทำธุรกรรมในแต่ละครั้งดังกล่าวจะถูกเก็บไว้ในกล่องที่เราเรียกว่า Block

ข้อดีของ Blockchain คือ ทุกคนมีรายการบัญชีของตัวเองและสามารถดูรายการบัญชีของทุกคนได้ในระบบผ่าน Address ที่ไม่ระบุตัวตน ทำให้มีความโปร่งใสในทุกขั้นตอน และการทำธุรกรรมทุกครั้งจะต้องผ่านการตรวจสอบจากเครือข่ายของ Node ต่างๆ ก่อนว่าข้อมูลตรงกันหรือไม่ ทำให้มีโอกาสฉ้อโกงนั้นมีน้อยและในการทำธุรกรรมนี้ยังมีการเข้ารหัสลับขั้นสูง ถึงแม้ว่าจะเห็นรายการบัญชีทั้งหมดและก็ไม่สามารถรู้ได้ว่าใครคือเจ้าของบัญชีนั้น ซึ่งผู้ที่จะสามารถแก้ไขรายการบัญชีได้คือ เจ้าของที่มี Private key เท่านั้น จึงส่งผลให้มีความปลอดภัยขั้นสูง โดยธุรกรรมที่เกิดขึ้นนั้นจะอยู่ในรูปแบบของ Block ที่ต่อกันเรื่อยๆ และไม่สามารถแก้ไขข้อมูลย้อนหลังได้ และการกระจายของการบันทึกข้อมูลที่ไม่มีศูนย์กลางนี้ จะช่วยให้ระบบสามารถดำเนินการต่อได้ต่างจากการบันทึกข้อมูลแบบมีศูนย์กลางที่หากระบบล่ม ก็จะดำเนินการต่อไม่ได้

Hash Function

Hash Function (แฮชฟังก์ชัน) คือ การเข้ารหัสทางเดียวโดยจะสร้าง Digital Signature ของข้อมูล Digital ที่ไม่สามารถถอดรหัสกลับมาได้ และใช้เป็นตัวแทนของข้อมูลนั้นๆ โดยใช้หลักการของ Private key และ Public key

ขั้นตอนการทำงานของ Hash Fucnction

สมมุติว่าจะเข้ารหัสข้อความ ***"I am Blockchain"***

- ทำการเข้ารหัสด้วย Function Hash โดยใช้ Private Key ของผู้ส่ง ออกมาเป็น Digital Signature
- เมื่อได้ Digital Signature มาก็จะส่งให้กับผู้รับ พร้อมกับ Public Key
- ผู้รับตรวจสอบ Digital Signature ที่ได้โดยใช้ Public Key ของผู้ส่งถ้าได้ค่า Hash ที่ตรงกัน ตรวจสอบได้ว่าเป็นข้อความที่ถูกต้อง เชื่อถือได้

คุณสมบัติหลักของ Hash Function ประกอบด้วย

- ค่า Hash ของข้อความเดิมจะต้องเหมือนกันเสมอ ไม่ว่าจะ Hash กี่รอบ ข้อความ "I am Blockchain" ก็ต้องได้ค่า Hash ที่เหมือนเดิมเสมอ
- Collision-Free : ค่า Hash ของข้อมูลที่ไม่เหมือนกันจะไม่มีทางเหมือนกัน เช่น ถ้าเข้ารหัสข้อความ "I am Blockchain" กับ "i am blockchain" ก็จะไม่มีทางได้ค่าที่เหมือนกัน
- Hiding : ไม่สามารถนำค่า Hash ที่ได้มาเพื่อถอดรหัสกลับหาค่าข้อความเดิมได้
- ค่า Hash จะมีความยาวเท่ากันเหมือนไม่ว่า ข้อความนั้นจะสั้นหรือยาว กล่าวคือถ้าเรา Hash ข้อความข้างต้น หรือใส่ไปทั้งไฟล์ ค่า Hash ที่ได้ก็จะมีความยาวเท่ากัน

ภาพที่ 3-5 ลักษณะข้อมูลแบบ Hash function
(ที่มา: https://medium.com/@iyawatkongmalai/blockchain-101-เข้าใจ-blockchain-แบบง่าย)

Bitcoin ได้นำเอาคุณสมบัติเหล่านี้มาใช้ในการตรวจสอบความถูกต้องของรายการ โดยสร้างเป็น Ledger Address ของแต่ละบัญชี เพื่อใช้แทนเลขที่บัญชี โดย Address ของ Bitcoin จะสร้างจาก

1. Private Key ซึ่งเป็น User/ Password หรือเป็น Secret words ที่แต่ละคนจะกำหนดไว้ ซึ่งทุกคนต้องจำให้ได้ ถ้าหากจำไม่ได้ก็จะไม่สามารถเข้าใช้งานบัญชีตัวเองได้
2. สร้าง Public Key จาก Private Key ขึ้นมา
3. เมื่อได้ Public Key ก็จะทำการคำนวณหาค่า Hash (SHA-256) เพื่อให้ได้ค่าที่มีความยาว 33–34 Bytes (Base58 string) และใช้แทนเลขที่บัญชี

3.2. องค์ความรู้เกี่ยวกับ Ethereum & Smart Contract

3.2.1 Ethereum

Ethereum (ETH) เป็นหนึ่งในสกุลเงินดิจิทัล (Cyptocurrency) ที่ใช้เทคโนโลยี Blockchain ทำงานอยู่เบื้องหลัง เช่นเดียวกับอีกหลาย ๆ สกุลเงิน เช่น Bitcoin, Ethereum, Ripple, Litecoin และ Cardano เป็นต้น ที่เกิดจากการร่วมระดมทุนแบบ ICO ในปี 2014 ทำให้มีการสร้างเหรียญขึ้นมา 60 ล้านเหรียญในระบบที่แตกต่างจากเหรียญอื่น ถึงแม้ Ethereum จะสามารถทำธุรกรรมการซื้อ ขาย โอนเงิน แลกเปลี่ยนเหมือนสกุลเงินอื่นๆ

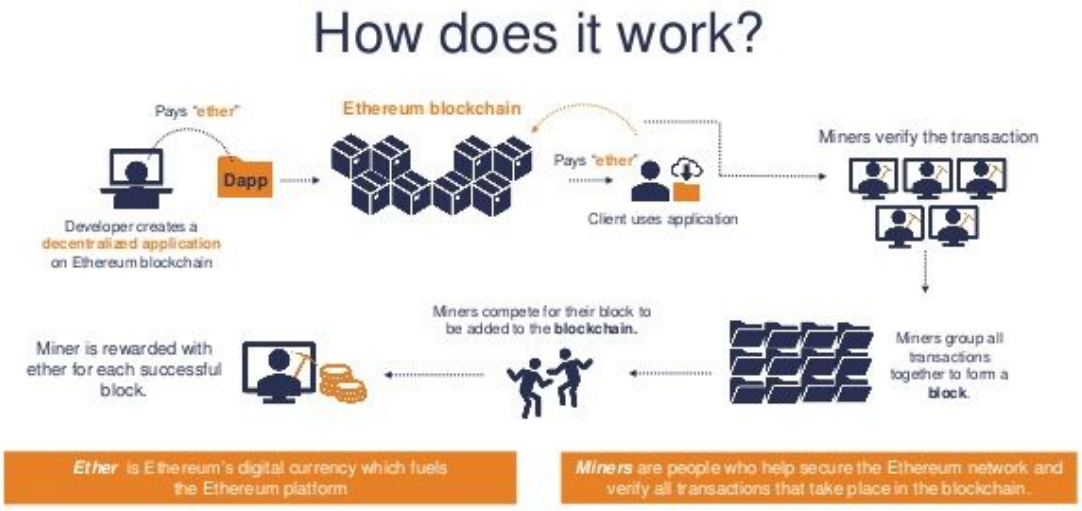
โดย ICO หรือ Initial Public Offering เป็นรูปแบบการระดมทุนแนวใหม่ที่กำลังได้รับความนิยมอย่างมาก โดยจะเป็นการออกเหรียญดิจิทัลชนิดใหม่ขึ้นมา แล้วเปิดขายให้ผู้ที่สนใจนำเงินมาลงทุน เพื่อนำเงินที่ระดมทุนได้ไปต่อยอดธุรกิจ หรือโครงการที่อยากจะทำ และด้วยความสามารถ Smart Contract จึงทำให้นักพัฒนาสามารถใช้ Ethereurm ออกเหรียญชนิดใหม่เป็นของตัวเองขึ้นมาได้ ซึ่งแน่นอนว่ายิ่งมีคนใช้ Ethereurm ระดมทุนทำ ICO มากเท่าไหร่ ก็ยิ่งเป็นผลดีต่อมูลค่าของ Ethereurm มากเท่านั้น

นอกจากนี้ Ethereum คือยังเป็นแพลตฟอร์มแบบเปิดของ Blockchain ที่ทำให้ทุกคนสามารถสร้างและใช้งานแอปพลิเคชันแบบกระจายข้อมูล (Decentralized) ซึ่งทำงานบนเทคโนโลยี Blockchain ได้ Ethereum มีความคล้าย Bitcoin ตรงที่ไม่มีใครสามารถควบคุมหรือเป็นเจ้าของ Ethereum ได้ เนื่องจาก Ethereum เป็นโครงการแบบโอเพนซอร์ส (Open-source project) ที่สร้างขึ้นโดยผู้คนเป็นจำนวนมากจากทั่วโลก แต่ Ethereum มีความแตกต่างจากโปรโตคอล Bitcoin เนื่องจาก Ethereum ถูกออกแบบมาเพื่อให้สามารถปรับตัวได้และมีความยืดหยุ่น ซึ่งคุณสมบัตินั่นคือ Smart Contract โดยหลักการของ Smart Contract บน Ethereum คือ ระบบพิเศษที่อนุญาตให้เขียนโปรแกรมลงไปในข้อมูลของสกุลเงิน และโปรแกรมนั้นจะทำงานโดยอัตโนมัติเมื่อตรงตามเงื่อนไขที่กำหนด ด้วยเหตุนี้การพัฒนา Application จากระบบการทำงานของ Ethereum จึงเป็นไปได้และถือเป็นความแปลกใหม่ต่างจาก Bitcoin ที่มีหน้าที่เดียวคือการเป็นสกุลเงินเท่านั้น

Ethereum เป็น Blockchain ที่เขียนชุดคำสั่งได้ ซึ่งแทนที่จะให้ผู้ใช้ชุดปฏิบัติการที่ถูกกำหนดไว้ล่วงหน้าแล้ว (Pre-defined operations) เช่น ธุรกรรมเกี่ยวกับ Bitcoin จึงทำให้ Ethereum กลับช่วยให้ผู้ใช้สามารถสร้างปฏิบัติการให้มีความซับซ้อนตามที่ผู้ใช้ต้องการได้ ด้วยวิธีนี้ Ethereum จะทำหน้าที่เป็นแพลตฟอร์มสำหรับแอปพลิเคชัน Blockchain ที่มีรูปแบบหลากหลาย ซึ่งสามารถใช้สกุลเงินดิจิตอลได้อย่างไม่จำกัด

Ethereum ในความหมายอย่างแคบ (Narrow sense) หมายถึงชุดของโปรโตคอลที่กำหนดแพลตฟอร์มสำหรับแอปพลิเคชันแบบกระจายข้อมูล หัวใจสำคัญของมันคือ Ethereum Virtual Machine (“EVM”) ซึ่งสามารถประมวลผลซึ่งมีอัลกอริทึมที่ซับซ้อนได้เอง ในแง่ของวิทยาการคอมพิวเตอร์ Ethereum นั้นอยู่ในระดับ “Turing complete” (แก้ปัญหาได้เทียบเท่าเครื่องคอมพิวเตอร์) นักพัฒนาซอฟต์แวร์สามารถสร้างแอปพลิเคชันที่ทำงานบน EVM โดยใช้ภาษาโปรแกรมที่รองรับได้ซึ่งมีตัวแบบเป็นภาษาที่มีอยู่ เช่น JavaScript และ Python

Ethereum ยังมีโปรโตคอลเครือข่ายแบบ peer-to-peer เช่นเดียวกับ Blockchain ทั้งนี้ ฐานข้อมูล Blockchain ของ Ethereum ถูกเก็บรักษาและพัฒนาปรับปรุงโดยโหนดหลายโหนดที่เชื่อมต่อกันเป็นเครือข่าย โหนดแต่ละโหนดในเครือข่ายจะเรียกใช้ EVM และใช้คำสั่งเดียวกัน ด้วยเหตุผลนี้ Ethereum จึงถูกอธิบายว่าเป็น “คอมพิวเตอร์ระดับโลก (“World computer)”



ภาพที่ 3-6 ลักษณะการทำงานของ Ethereum

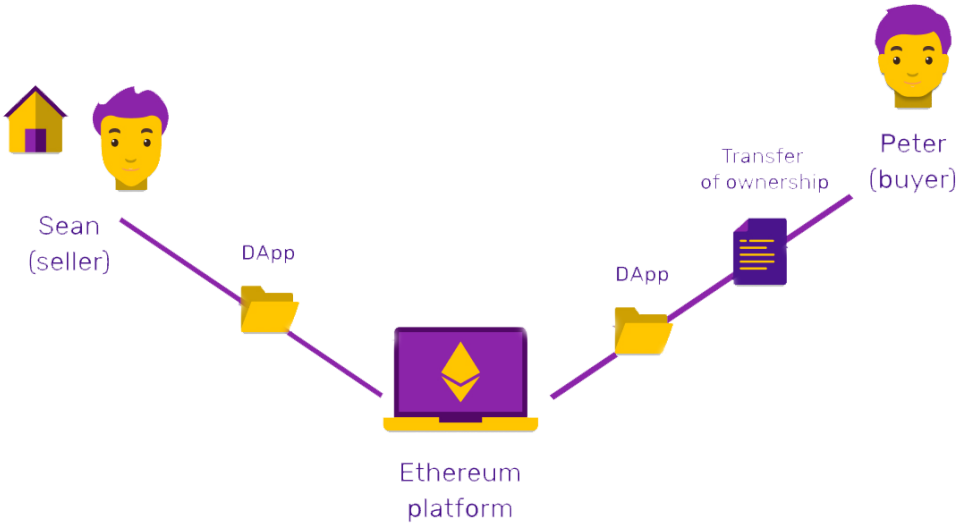
(ที่มา: https://www.slideshare.net/cordieliea/binarycom-what-is-ethereum-and-how-does-it-work)

จากภาพที่ 3-6 แสดงให้เห็นถึงหลักการทำงานของ Ethereum คือ เมื่อนักพัฒนาทำการสร้าง Appilcation และผู้ใช้งานทำการใช้ Appilcation จะมีการจ่าย Ether (ETH) ไปยัง Ethereum Blockchain จากนั้น Miner จะทำการตรวจสอบ Transaction เมื่อตรวจสอบ Transaction เรียบร้อยแล้ว ข้อมูลจะถูกบันทึกลง Blockchain โดยสุดท้าย Miner ที่ตรวจสอบ Block สำเร็จเป็นคนแรก จะได้รับรางวัลเป็น Ether (ETH)

3.2.2 Smart Contracts

Smart Contract หรือ สัญญาอัจฉริยะ คือ Distributed ledgers หรือฐานข้อมูลแบบกระจายทำให้รหัส (Code) ที่อยู่ในสัญญาเกิดผลบังคับใช้ทันทีเมื่อมีการทำตามเงื่อนไขที่ได้ระบุไว้ในสัญญา Ethereum ก็เลยถูกสร้างขึ้นเพื่อการนี้โดยเฉพาะ โดย Smart Contract สามารถตั้งโปรแกรมให้ทำงานได้เมื่อบรรลุเงื่อนไขตามที่กำหนดในสัญญา เช่น ตราสารจะสามารถสั่งจ่ายได้ต่อเมื่อเครื่องมือทางการเงิน (Financial instruments) เป็นไปตามเกณฑ์มาตรฐานที่กำหนดซึ่งเทคโนโลยี Blockchain และ Bitcoin จะทำให้การจ่ายเงินเป็นไปโดยอัตโนมัติ เงื่อนไขหรือข้อตกลงของสัญญาต่างๆ ที่ถูกจัดเก็บไว้ในรูปแบบ Code คอมพิวเตอร์ ซึ่ง Code นี้จะถูกเก็บไว้ในเครือข่ายของ Blockchain อีกที โดยถ้าหากมีคำสั่งที่ตรงตามเงื่อนไขข้อตกลงที่วางไว้ ระบบจะดำเนินการทำธุรกรรมต่างๆ ตามข้อตกลงโดยอัตโนมัติ ความคิดในการทำ Smart Contract นี้เกิดจากนักคอมพิวเตอร์คนหนึ่งชื่อว่า Nick Szabo ซึ่งเป็นผู้ริเริ่มความคิดนำเอา Blockchain มาใช้กับการทำสัญญา ซึ่งอย่างที่ทราบกันดีว่าการใช้เทคโนโลยี Blockchain นั้น ไม่จำเป็นต้องมีคนกลาง หรือพนักงานมานั่งตรวจสอบเอกสาร โดยทุกอย่างเป็นการทำงานของคอมพิวเตอร์ทั้งหมด อีกทั้งยังเป็นการเพิ่มความปลอดภัยของข้อมูลด้วยการเก็บข้อมูลในรูปแบบ Distributed ledger หรือการกระจายข้อมูลให้ทุกคนในเครือข่าย ทุกคนในเครือข่ายจึงมีข้อมูลที่เหมือนกันทั้งหมด และเป็นพยานว่าสัญญานี้เกิดขึ้นและบรรลุจริงๆ ทำให้ไม่สามารถทุจริตได้ มีคนอธิบายไว้ว่า Smart Contract มีหลักการทำงานแบบ **“if-this-then-that”** หรือการ

ทำงานแบบป้อนคำสั่งในรูปแบบเงื่อนไขล่วงหน้าไว้ว่า “ถ้าเป็นแบบนี้ แล้วจึงทำแบบนั้น" โดยอัตโนมัติ ตั้งแต่ขั้นตอนแรกจนถึงขั้นตอนสุดท้าย



ภาพที่ 3-7 ลักษณะการทำงานของ Smart Contract
(ที่มา: https://blockspoint.com/guides/ethereum/what-is-a-smart-contract)

จากภาพที่ 3-7 แสดงให้เห็นถึงหลักการทำงานของ Smart Contract เมื่อผู้ใช้สองคนต้องการแลกเปลี่ยนสินค้า โดยผู้ขายจะนำข้อมูลสินค้าลงใน DApp และมีการกำหนดข้อตกลงของการแลกเปลี่ยนสินค้าชิ้นนี้ ซึ่งข้อมูลถูกบันทึกลงใน Ethereum และหากมีผู้ซื้อต้องการแลกเปลี่ยนสินค้าชิ้นนี้ โดยมีคุณสมบัติหรือเงื่อนไขตรงตามที่ผู้ขายกำหนดไว้ในข้อตกลง ผู้ขายก็จะทำการแลกเปลี่ยนสินค้ากับผู้ซื้อได้ ทำให้เกิด Transaction โดยอัตโนมัติและข้อมูลจะถูกบันทึกลงใน Blockchain

อย่างที่กล่าวไว้ว่า Smart Contract ทำงานบนระบบของ Blockchain จึงทำให้มีคุณสมบัติบางอย่างของ Blockchain นั่นคือ

- Immutable : เปลี่ยนรูปไม่ได้ ซึ่งหมายความว่าสัญญาไม่สามารถเปลี่ยนแปลงได้และไม่มีใครสามารถเข้าไปแก้ไขหรือทำลายสัญญาได้
- Distributed : สัญญาถูกกระจายออกไป ซึ่งหมายความว่าผลลัพธ์ของสัญญาก็จะได้รับการตรวจสอบจากทุกคนในเครือข่ายหากมีผู้ไม่หวังดีทำการเปลี่ยนแปลงผลลัพธ์ก็จะถูกตรวจพบและทำเครื่องหมายไว้ว่าไม่ถูกต้อง

3.3. ภาษา Solidity

ภาษา Solidity เป็นภาษาสำหรับการสร้าง Smart Contract เป็นภาษาที่ได้รับอิทธิพลมาจาก C++, Python และ JavaScript ที่สำคัญคือเป็นภาษาชนิดที่ Statically Typed และเป็นภาษาแบบ Object Oriented (OO) เนื่องจากมีคุณสมบัติของการสืบทอดและการทำ struct เป็นต้น

ชนิดของตัวแปร (Value Types)

1. Booleans : bool (true and false)
2. Integers : int/uint สามารถกำหนดขนาดที่ใช้ได้โดยมีขนาดตั้งแต่ 8-256 bits เช่น int8 และ uint16 และหากไม่ได้ได้ระบุขนาดของ bits จะมีขนาด 256 โดยอัตโนมัติ เช่น int หรือ uint นั้นหมายความว่ามีค่าเท่ากับ int256 หรือ uint256
3. Bytes : bytes มีขนาดตั้งแต่ 1–32 bytes เช่น bytes8 หรือ bytes32 และหากไม่กำหนดขนาดก็จะเป็น Array Dynamic Size
4. Strings : string ไม่ให้กำหนดขนาดของ bytes หมายความว่าจะมองเป็น Array Dynamic Size ซึ่งมีความแตกต่างจากการใช้ bytes ที่มีการกำหนดขนาดตรงที่จำนวนของ gas ที่ใช้ strings จะใช้ gas มากกว่า
5. Address : address มีค่าอยู่ที่ 20 byte ตามขนาดของ Ethereum address

รูปแบบการเขียน (Syntax)

1. การประกาศ Contract : เป็นการสร้างเงื่อนไขว่าต้องการให้เป็นแบบไหน ซึ่งการสร้าง Contract ต้องมี Keyword Contract เหมือนกับ Keyword Class ในการสร้างคลาสใน Java

```
syntax : contract name {...}

----------------------------------------------------------------------

pragma solidity ^0.4.17;

contract MyContract {

    ...

}
```

ภาพที่ 3-8 รูปแบบการประกาศ Contract

(ที่มา: https://medium.com/20scoops-cnx/มารู้จักกับ-solidity-ขั้นพื้นฐานกัน-6f713b3fb64))

จากภาพที่ 3-8 แสดงรูปแบบการประกาศชื่อ Contract และบรรทัด pragma solidity ^0.4.17 แสดงถึงเวอร์ชันของภาษา Solidity ซึ่งในที่นี้คือเวอร์ชัน 0.4.17 ซึ่งเวอร์ชันเหล่านี้มีการอัปเดตตลอดเวลา

2. การประกาศตัวแปร : การประกาศตัวแปรว่าต้องการเก็บตัวแปรเป็นรูปแบบชนิดใด (Value types) ดังภาพที่ 3-9

```
syntax : value_types name = value;

----------------------------------------------------------------------

pragma solidity ^0.4.17;

contract ExampleContract {

    bool myBool = true;

    int myInt = 5;
    uint myUint = 999;

    uint8 myUint8 = 255; // uint8 range 0 to 255
    int16 myInt16 = 32767; // int16 range -32,768 to 32,767

    bytes myBytes = "20scoops CNX";
    string myStr = "Chiang Mai, the most beautiful city";

    address myAddress = 0x72ba7d8e73fe8eb666ea66babc8116a41bfb10e2;

    uint[] myArr = [1,3,5,7,9];

    struct User {
        string name;
        int age;
   }
}
```

ภาพที่ 3-9 รูปแบบการประกาศตัวแปรของ Contract
(ที่มา: https://medium.com/20scoops-cnx/มารู้จักกับ-solidity-ขั้นพื้นฐานกัน-6f713b3fb64)

3. เงื่อนไขและการวนลูป : วิธีการเงื่อนไข If-else, For, While, Do-While ซึ่งวิธีการเขียนคล้ายๆกับภาษา Java ดังภาพที่ 3-10

```
pragma solidity ^0.4.17;

contract ExampleContract {

    uint[] myArr = [1,2,3,4,5,6,7,8,9,10];

    function testCondition(uint n) public view returns (string) {
        if (n%2 == 0) {
            return "is even number";
        } else {
            return "is odd number";
        }
    }

    function testFor() public view returns (uint) {
        uint sum = 0;
        for(uint i=0; i<myArr.length; i++) {
            sum += myArr[i];
        }
        return sum;
    }

    function testWhile() public view returns (uint) {
        uint sum = 0;
        uint i = 0;
        while (i<myArr.length) {
            sum += myArr[i];
            i++;
        }
        return sum;
    }

    function testDoWhile() public view returns (uint) {
        uint sum = 0;
        uint i = 0;
        do {
            sum += myArr[i];
            i++;
        } while (i<myArr.length);
        return sum;
    }
}
```

ภาพที่ 3-10 รูปแบบเงื่อนไขและการวนลูปของ Contract
(ที่มา: https://medium.com/20scoops-cnx/มารู้จักกับ-solidity-ขั้นพื้นฐานกัน-6f713b3fb64)

4. การประกาศฟังก์ชัน : ในการประกาศฟังก์ชันต้องขึ้นต้นด้วย Keyword เป็น function ตามด้วยชื่อของฟังก์ชัน ต่อด้วย parameter และลักษณะการเข้าถึงฟังก์ชัน ดังภาพที่ 3-11

```
function calculate1(int _x, int _y, int _z, bool _flag)
    returns (int _alpha, int _beta, int _gamma) {  //#A
    _alpha = _x + _y; //#B
    _beta = _y + _z;  //#B
    if ( flag)
        _gamma = _alpha / _beta; //#B
    else
        _gamma = _z; //#B
}
```

ภาพที่ 3-11 รูปแบบการประกาศฟังก์ชันของ Contract
(ที่มา: https://medium.com/20scoops-cnx/มารู้จักกับ-solidity-ขั้นพื้นฐานกัน-6f713b3fb64)

5. การสืบทอด Contract : คุณสมบัติในการสืบทอดในภาษา Solidity มี Keyword คือ is ซึ่งเหมือนกับ Keyword คือ extends ในภาษา Java ดังภาพที่ 3-12

```
syntax : contract is contractParent

-----------------------------------------------------------------------

pragma solidity ^0.4.17;

contract Animal {

    string internal nameAnimal;

    function setName(string name) public {
        nameAnimal = name;
    }

    function getName() public view returns (string) {
        return nameAnimal;
    }

    function speak(string sound) public pure returns (string) {
        return sound;
    }

    function somethingAnimal() private { }
}

contract Cat is Animal {

    function something() public { }
}
```

ภาพที่ 3-12 รูปแบบการสืบทอด Contract
(ที่มา: https://medium.com/20scoops-cnx/มารู้จักกับ-solidity-ขั้นพื้นฐานกัน-6f713b3fb64)

- การเรียกใช้ Contract : การเรียกใช้งานจาก Contract หนึ่งไปอีก Contract หนึ่งซึ่งคล้ายกับการประกาศ Object ของภาษา Java โดยมี Keyword คือ new ดังภาพที่ 3-13

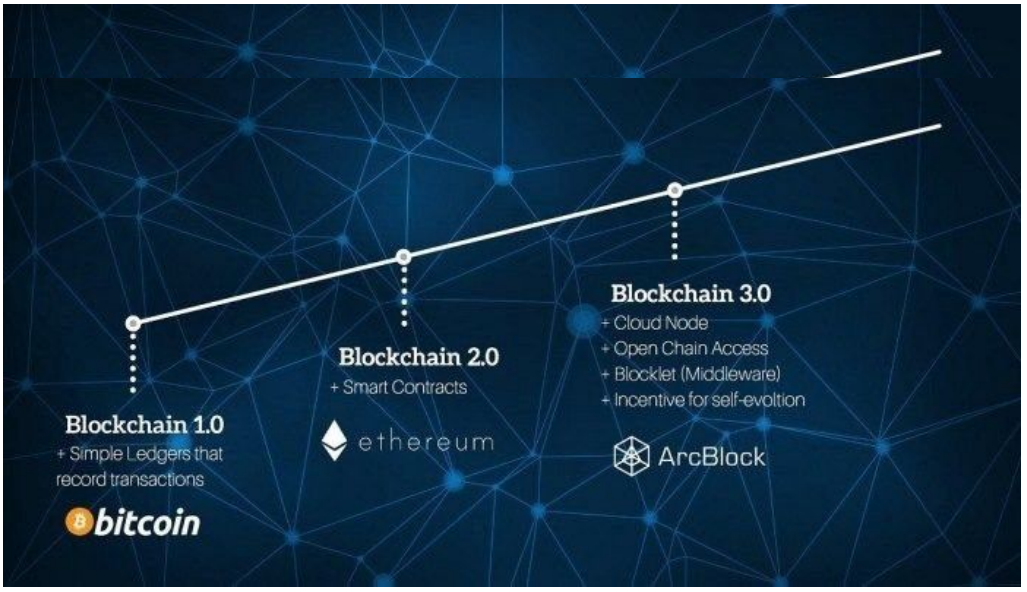
```
syntax : ContractName contract = new ContractName();

----------------------------------------------------------------------

contract A {
    function printName() public pure returns (string) {
        return "20scoops CNX";
    }
}

contract B {

    A a = new A();

    function printString() public view returns (string) {
        return a.printName();
    }
}
```

ภาพที่ 3-13 รูปแบบการเรียกใช้ Contract

(ที่มา: https://medium.com/20scoops-cnx/มารู้จักกับ-solidity-ขั้นพื้นฐานกัน-6f713b3fb64)

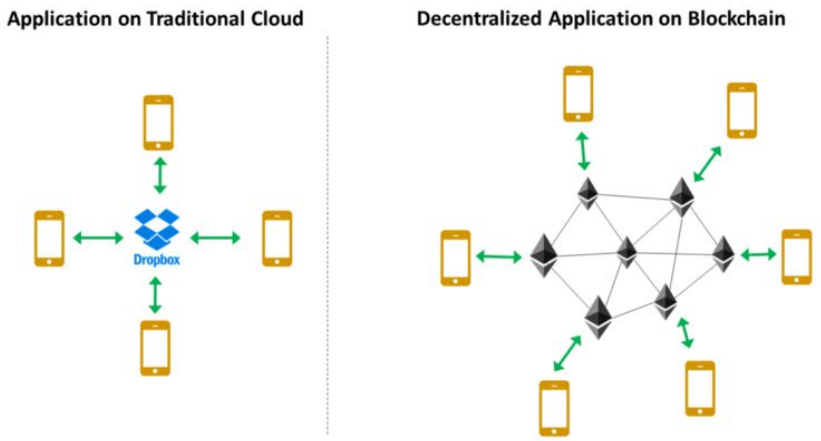
3.4. องค์ความรู้เกี่ยวกับ DApp (Decentralized Application)

หลังจากที่การพัฒนา Smart Contract ในยุคของ Blockchain 2.0 เริ่มเป็นรูปธรรมมากขึ้น เทคโนโลยี Blockchain ที่พัฒนาอย่างไม่หยุดหย่อนได้ก้าวเข้าสู่ยุคของ Blockchain 3.0 เป็นยุคของ Decentralized application หรือ Dapp ถือเป็นการพัฒนาแบบพลิกโฉมของ Blockchain แบบเดิมๆ เพราะเป็นการเปิดประตูไปสู่อุตสาหกรรมอื่น ๆ อีกมากมาย โดยผู้ประกอบการสามารถนำ Smart contract มาสร้างเป็น Application ใช้สนับสนุน และแก้ปัญหาในธุรกิจรูปแบบเดิมๆ อย่างเป็นรูปธรรมเช่น การเงินการธนาคาร, ธุรกิจด้านอสังหาริมทรัพย์, การวิจัยและการศึกษา, การแพทย์และสาธารณสุข และอื่นๆ อีกมากมาย



ภาพที่ 3-14 ตัวอย่างประกอบการอธิบายการพัฒนาของ Blockchain
(ที่มา: http://medium.com/@prick.aunt/ตอนที่4-blockchain-3-0-เมื่อเงินดิจิตอลไม่ใช่แค่การเก็งกำไรอีกต่อไป-7a30d606b104)

เมื่อเปรียบเทียบจุดเด่นและจุดด้อยระหว่าง Centralized และ Decentralized



ภาพที่ 3-15 การเปรียบเทียบ Application แบบ Centralized และ Decentralized
(ที่มา: https://medium.com/@chawansit/ย้อนเวลาหา-ethereum-บทความปูพื้น-76cc97cbda13)

Centralized Application

- ทำงานได้รวดเร็วกว่า Decentralized มาก
- รองรับปริมาณการเรียกใช้งานได้มหาศาล โดยอาศัยเทคนิค Geo-location load balancing + Content Delivery Network (CDN)
- การปรับปรุงระบบสามารถค่อยๆทำได้ ไม่จำเป็นต้องทำพร้อมกันทั้งหมด

Decentralized Application

- ปลอดภัยกว่าระบบ Centralized
- ทำงานได้ช้ากว่า Centralized มากๆ เพราะจะสามารถทำได้เร็วที่สุดเท่ากับสามารถของแค่ 1 Node เท่านั้น
- ในปัจจุบันยังรองรับปริมาณธุรกรรมได้น้อย
- การปรับปรุงระบบต้องทำพร้อมกันทั้งหมด

3.5. องค์ความรู้เกี่ยวกับ IPFS (Interplanetary File System)

IPFS ย่อมาจาก Interplanetary File System เป็นโปรโตคอลการกระจายสื่อหลายมิติที่มีวัตถุประสงค์เพื่อรองรับการจัดเก็บและการแบ่งปันไฟล์ในรูปแบบการกระจายอำนาจ ไฟล์จะไม่ได้รับการจัดเก็บอยู่ในตำแหน่งเดียว แต่จะถูกแบ่งออกเป็นบล็อกๆและกลายเป็น IPFS objects และการจายออกไปในเครือข่าย

ภาพที่ 3-16 สัญลักษณ์ของ IPFS

คุณสมบัติหลักๆของ IPFS ประกอบด้วย

1. การกระจายตัว ซึ่งคุณสมบัตินี้มีขึ้นเพื่อแก้ปัญหาเว็บที่มีเว็บเซิร์ฟเวอร์เป็นศูนย์กลางในการทำงาน ถึงแม้ว่ามีการถูกวางไว้แยกเป็นหลายเว็บแต่การทำงานจริงๆนั้นยังคงทำในที่เดียว โดย IPFS สามารถแก้ปัญหาได้โดยวิธีการกระจายข้อมูลที่เหมือนกันไปยังหลายที่
2. สามารถทำงานได้แม้ไม่ได้เชื่อมต่ออินเทอเน็ต โดยเมื่อมีใครได้ข้อมูลนั้นแล้วต้องการให้บริการข้อมูลนั้นกับคนอื่นแล้วยังสามารถทำงานออฟไลน์ได้ด้วย
3. แก้ปัญหาการโดยบล็อคหรือเซนเซอร์ไม่ให้เข้าเว็บได้ เพราะไม่มีใครที่เป็นจุดเดียวที่จะบล็อคได้นั้นเอง
4. การเก็บข้อมูลแบบถาวร ไม่มีการลบและแก้ไขไม่ได้
5. ใช้ได้กับเทคโนโลยีเว็บปัจจุบัน ผู้ใช้ไม่จำเป็นต้องลงโปรแกรมเฉพาะเพื่อเรียกข้อมูลเรียกผ่าน URL ได้บนเว็บบราวเซอร์ได้เลย ตัวอย่างเช่น
   https://gateway.ipfs.io/ipfs/QmYwAPJzv5CZsnA625s3Xf2nemtYgPpHdWEz79ojWnPbdG

จึงนิยมใช้ IPFS ใช้เป็นที่เก็บข้อมูลสำหรับ Ethereum ควบคู่กับ Blockchain เพราะข้อมูลไม่สามารถเปลี่ยนแปลงได้ และการเก็บข้อมูลจำนวนมากบน Blockchain นั้นมีการสิ้นเปลืองค่า Gas มาก ดังนั้นการนำ IPFS มาใช้แทนแล้วจึงนำค่า Hash ใส่ใน Blockchain แทน ซึ่งจะทำให้สูญเสียค่า Gas น้อยกว่า

บทที่ 4
วิธีการดำเนินงาน

4.1. วิธีการดำเนินงานวิจัย

- **การวางแผนโครงงาน**
    - เลือกหัวข้อโครงงาน
    - นำเสนอหัวข้อโครงงาน
    - วางแผนการดำเนินงาน
    - ศึกษาแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง
    - ศึกษาภาษาและเครื่องมือที่ใช้ในการพัฒนา
- **การออกแบบระบบ**
    - กำหนดขอบเขตของระบบ
    - ออกแบบการทำงานของระบบ
    - ออกแบบฟังก์ชันการทำงานของแอปพลิเคชัน
    - ออกแบบ Interface ของแอปพลิเคชัน
- **การพัฒนาระบบ**
    - พัฒนาฟังก์ชันเข้าสู่ระบบผ่าน MetaMask
    - พัฒนาฟังก์ชันสำหรับกรอกรายละเอียดบทความและบันทึกข้อมูลลงบล็อกเชน
    - พัฒนาฟังก์ชันอัปโหลดไฟล์บทความผ่าน IPFS
- **การทดสอบและแก้ไขแอปพลิเคชัน**
    - ทดสอบการทำงานของแอปพลิเคชัน
    - แก้ไขข้อผิดพลาดของแอปพลิเคชัน
    - ปรับปรุงแอปพลิเคชันให้สมบูรณ์
- **การวิจัยประเมินผล**
    - ประเมินผลการทำงานของแอปพลิเคชัน
    - สรุปผลการดำเนินงาน
    - จัดทำรูปเล่มงานวิจัย

4.2. แผนการดำเนินงานตลอดโครงงาน

แผนการดำเนินงานจะแสดงในตารางที่ 4-1

ลำดับที่	แผนการดำเนินงาน	2561					2562				
		ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1	เลือกหัวข้อโครงงาน										
2	นำเสนอหัวข้อโครงงาน										
3	วางแผนการดำเนินงาน										
4	ศึกษาแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง										
5	ศึกษาภาษาและเครื่องมือที่ใช้ในการพัฒนา										
6	กำหนดขอบเขตโครงงาน										
7	ออกแบบการทำงานของระบบ										
8	ออกแบบฟังก์ชันการทำงานของแอปพลิเคชัน										
9	ออกแบบ Interface ของแอปพลิเคชัน										
10	พัฒนาฟังก์ชันเข้าสู่ระบบผ่าน MetaMask										
11	พัฒนาฟังก์ชันสำหรับกรอกรายละเอียดของบทความและบันทึกข้อมูลลงบล็อกเชน										
12	พัฒนาฟังก์ชันอัปโหลดไฟล์บทความผ่าน IPFS										

ลำดับที่	แผนการดำเนินงาน	2561					2562				
		ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
13	ทดสอบการทำงานของแอปพลิเคชัน										
14	แก้ไขข้อผิดพลาดของแอปพลิเคชัน										
15	ปรับปรุงแอปพลิเคชันให้สมบูรณ์										
16	สรุปผลการดำเนินงาน										
17	จัดทำรูปเล่มงานวิจัย										

ตารางที่ 4-1 แผนการดำเนินงานตลอดโครงงาน

4.3. อุปกรณ์และเครื่องมือที่ใช้ในงานวิจัย

4.3.1. ฮาร์ดแวร์ (Hardware)

- **เครื่องคอมพิวเตอร์โน๊ตบุ๊ค : ใช้สำหรับพัฒนาแอปพลิเคชัน**
  - Processor : Intel(R) Core i7-7700HQ
  - Memory : 8.00 GB

4.3.2. ซอฟท์แวร์ (Software)

- **Visual Studio Code**
  - ใช้สำหรับพัฒนาแอปพลิเคชัน
- **Adobe Photoshop CS6**
  - ใช้สำหรับการตกแต่งภาพในแอปพลิเคชัน

4.3.3. ภาษาที่ใช้ในการพัฒนา

- **Front-End** : Javascript , HTML, CSS
- **Back-End :** Solidity, Javascript

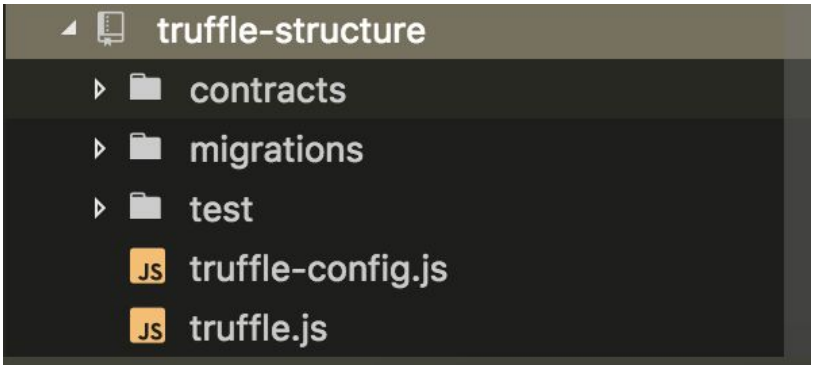
4.3.4. เครื่องมือที่ใช้ในการพัฒนา (Tools)

1. **Truffle**

ภาพที่ 4-1 สัญลักษณ์ของ Truffle Framework

Truffle เป็น Framework ที่ใช้สำหรับพัฒนา Ethereum โดยมีลักษณะดังนี้

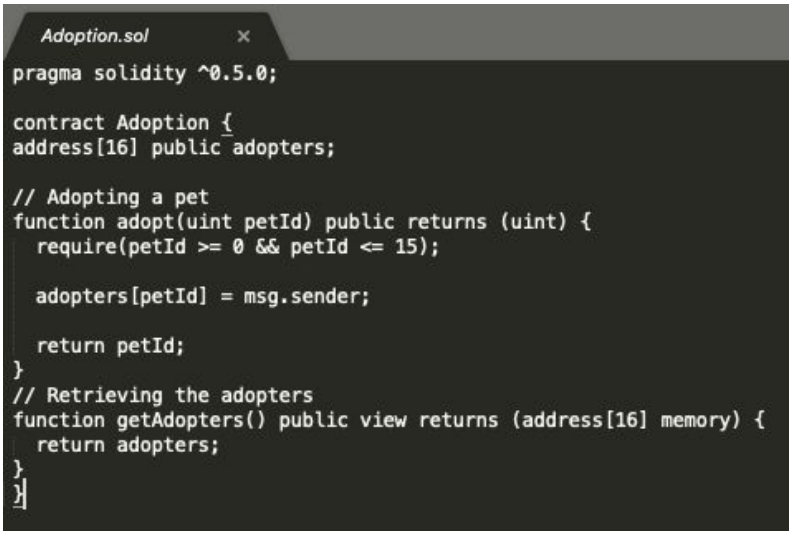
- รวบรวม Smart Contracts ที่สร้างขึ้นและ Deploy ขึ้นไปบน Ethereum Platform หรือ Deploy ใน Localhost ของเครื่องตนเอง
- เขียน Test ทดสอบความถูกต้องของ Smart Contracts
- สามารถ Log ดูการทำงานใน Smart Contracts ได้



ภาพที่ 4-2 โครงสร้างการทำงานของ Truffle Framework

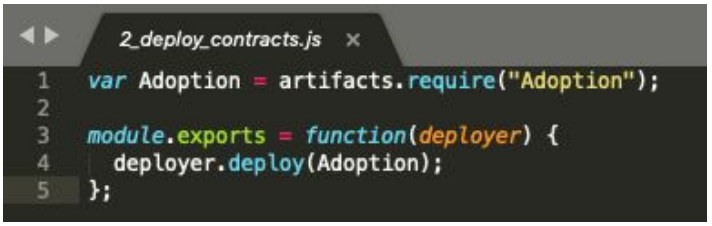
จากภาพที่ 4-2 จะเห็นได้ว่าโครงสร้างการทำงานของ Truffle Framework แบ่งออกเป็น 3 ส่วนดังนี้

1.1. Contracts ใช้สำหรับการพัฒนา Contract เพื่อสร้างฟังก์ชันการใช้งานในแอปพลิเคชัน โดยใช้ภาษา Solidity ในการพัฒนา



ภาพที่ 4-3 ลักษณะการเขียนภาษา Solidity

1.2. Migrations เป็นส่วนการ Deploy Contract สำหรับการนำไฟล์ขึ้นบน Server หรือ Localhost บนเครื่องตนเอง



ภาพที่ 4-4 ลักษณะการเขียนสำหรับการ Deploy

1.3. Test ใช้สำหรับการเขียน Test เพื่อตรวจสอบฟังก์ชันการใช้งานให้เกิดความถูกต้องยิ่งขึ้นและป้องกันการผิดพลาดของการใช้แอปพลิเคชัน โดยสามารถใช้ภาษา Solidity หรือ JavaScript ก็ได้

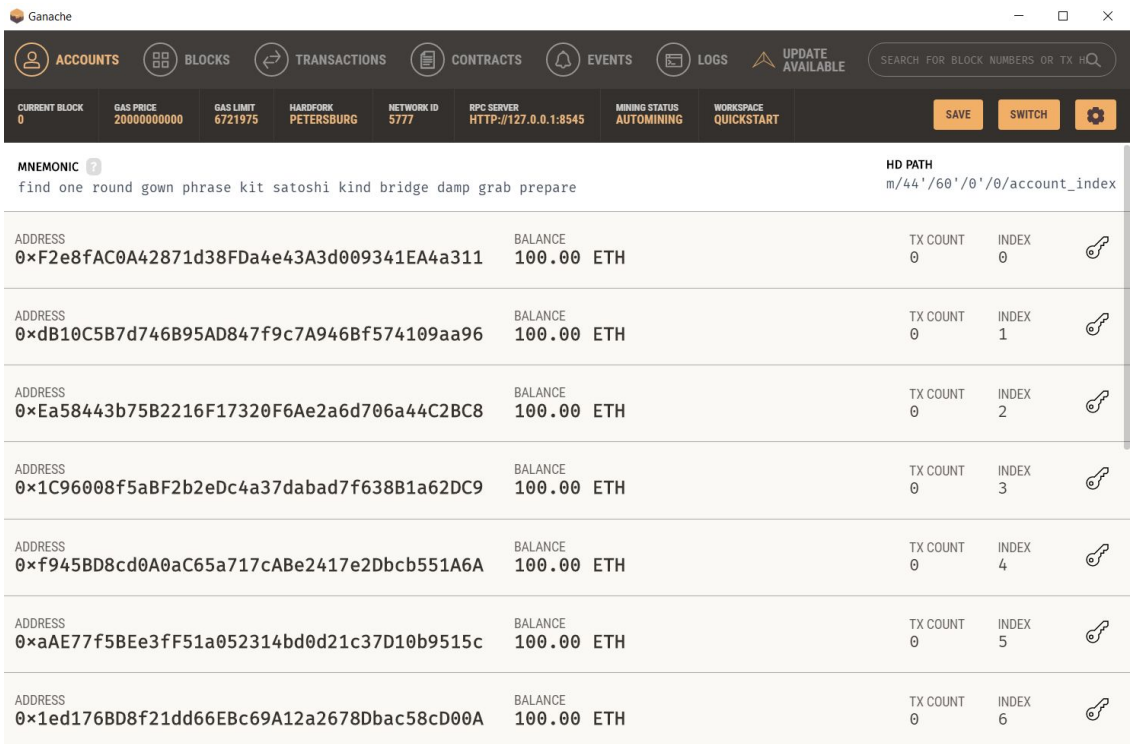
```
TestAdoption.sol
pragma solidity ^0.5.0;

import "truffle/Assert.sol";
import "truffle/DeployedAddresses.sol";
import "../contracts/Adoption.sol";

contract TestAdoption {
 // The address of the adoption contract to be tested
 Adoption adoption = Adoption(DeployedAddresses.Adoption());

 // The id of the pet that will be used for testing
 uint expectedPetId = 8;

 //The expected owner of adopted pet is this contract
 address expectedAdopter = address(this);

 // Testing the adopt() function
function testUserCanAdoptPet() public {
  uint returnedId = adoption.adopt(expectedPetId);

  Assert.equal(returnedId, expectedPetId, "Adoption of the expected pet should match what is returned.");
}
// Testing retrieval of a single pet's owner
function testGetAdopterAddressByPetId() public {
  address adopter = adoption.adopters(expectedPetId);

  Assert.equal(adopter, expectedAdopter, "Owner of the expected pet should be this contract");
}
// Testing retrieval of all pet owners
function testGetAdopterAddressByPetIdInArray() public {
  // Store adopters in memory rather than contract's storage
  address[16] memory adopters = adoption.getAdopters();

  Assert.equal(adopters[expectedPetId], expectedAdopter, "Owner of the expected pet should be this contract");
}
}
```

ภาพที่ 4-5 ลักษณะการเขียนทดสอบความถูกต้องของแอปพลิเคชัน

2. Ganache

ภาพที่ 4-6 สัญลักษณ์ของ Ganache

Ganache จะช่วยให้สร้าง Private Ethereum Blockchain สำหรับทดสอบ Tests และตรวจสอบสถานะโดยไม่มีค่าใช้จ่าย จึงใช้ Ganache เพื่อทดสอบความถูกต้องของการพัฒนา Smart Contracts ในระหว่างการพัฒนา



ภาพที่ 4-7 หน้าจอของ Ganache ที่แสดง Address และ Ether

โดยจากภาพที่ 4-7 แสดงถึงการ Generate Address จำนวน 10 Address 10 Private key และ Ether จำนวน 100 ETH เพื่อใช้ในการทดสอบความถูกต้องในการทำ Transaction และ Smart Contract ซึ่งจากภาพได้มีการกำหนด Server คือ 127.0.0.1:8545

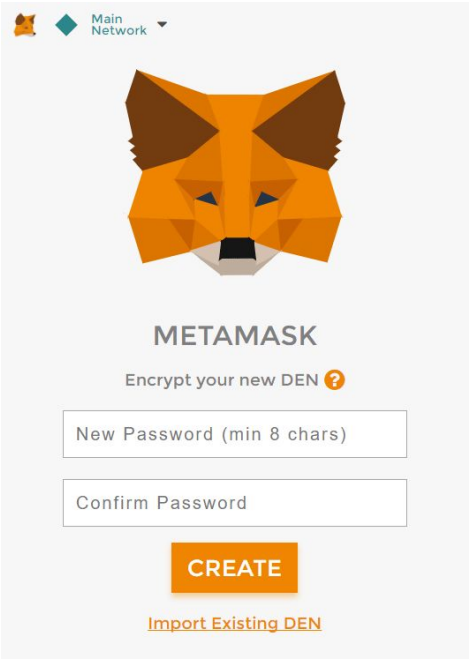
3. MetaMask

ภาพที่ 4-8 สัญลักษณ์ของ MetaMask

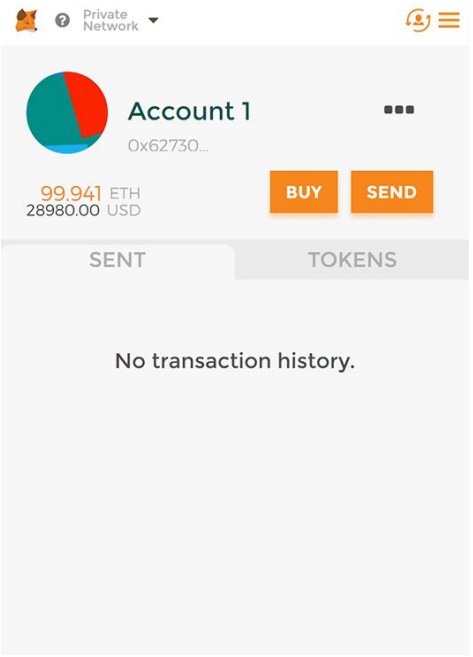
MetaMask เป็นกระเป๋าเก็บคริปโตแบบ ERC-20 ช่วยให้สามารถรัน Ethereum DApp โดยตรงใน Browser ของเราเอง โดยไม่ต้องใช้ Node Ethereum แบบเต็มรูปแบบ

ERC-20 คือ ข้อกฎต่างๆที่เหรียญ Ethereum-based token ทุกๆตัวตัวต้องทำตาม ซึ่งหมายความว่านักพัฒนาที่ต้องการสร้างเหรียญให้เพื่อทำงานบน Ethereum จะต้องทำตามกฎต่างๆตามที่ Protocol ของ Ethereum ระบุไว้เพื่อให้ทำงานร่วมกัน

สามารถติดตั้ง MetaMask Plugin ใน Chrome, Firefox, Opera เพื่อให้สามารถพัฒนาได้สะดวกขึ้น



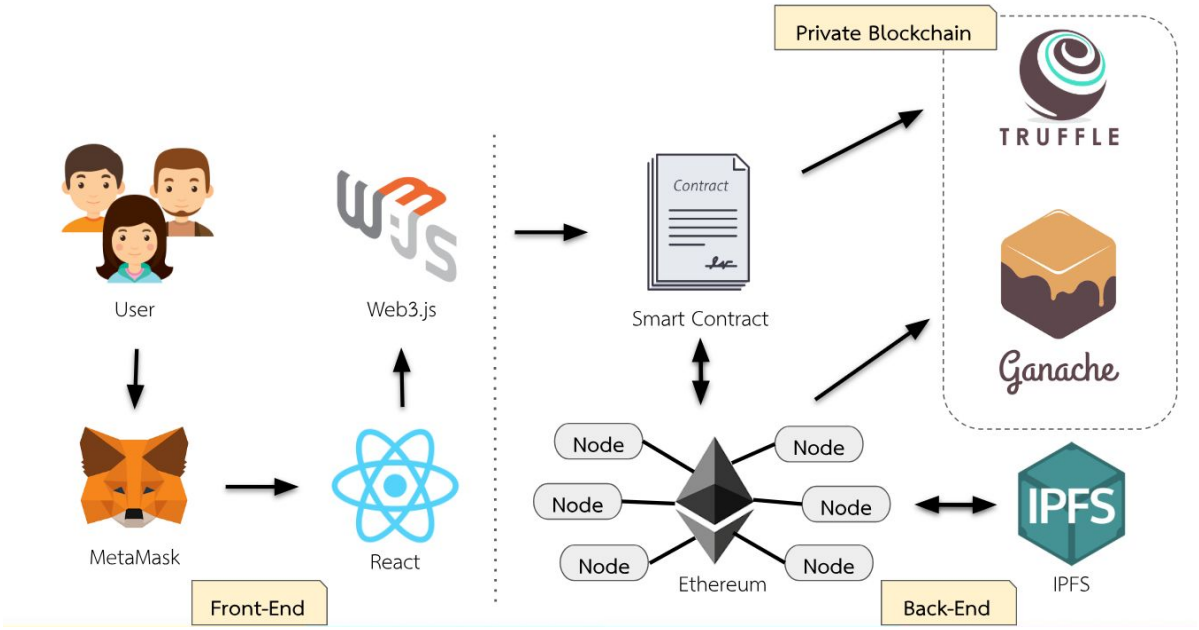
ภาพที่ 4-9 หน้าจอของ MetaMask Plugin



ภาพที่ 4-10 หน้าจอแสดง Account ของ MetaMask

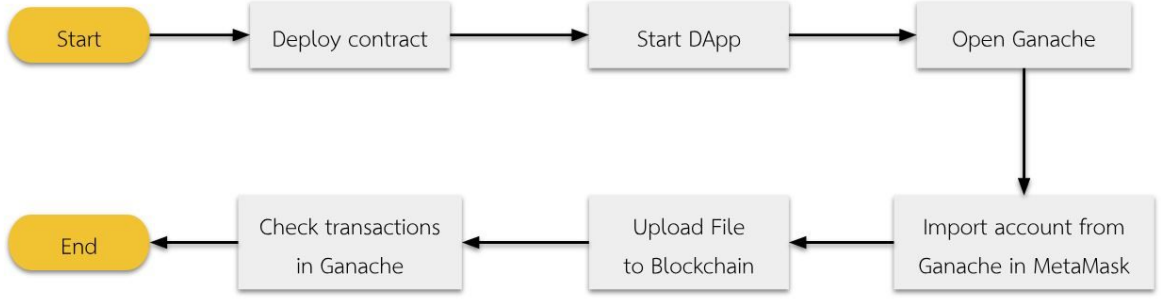
4.4. ขั้นตอนการออกแบบระบบ

4.4.1. โครงสร้างของระบบ



ภาพที่ 4-11 โครงสร้างของระบบ

4.4.2. ขั้นตอนการทำงานของระบบ



ภาพที่ 4-12 ขั้นตอนการทำงานของระบบ

บทที่ 5
ผลการดำเนินงาน

5.1. ผลการดำเนินงาน

5.1.1 หน้าแรกของ Application

ภาพที่ 5-1 หน้าแรกของ Application

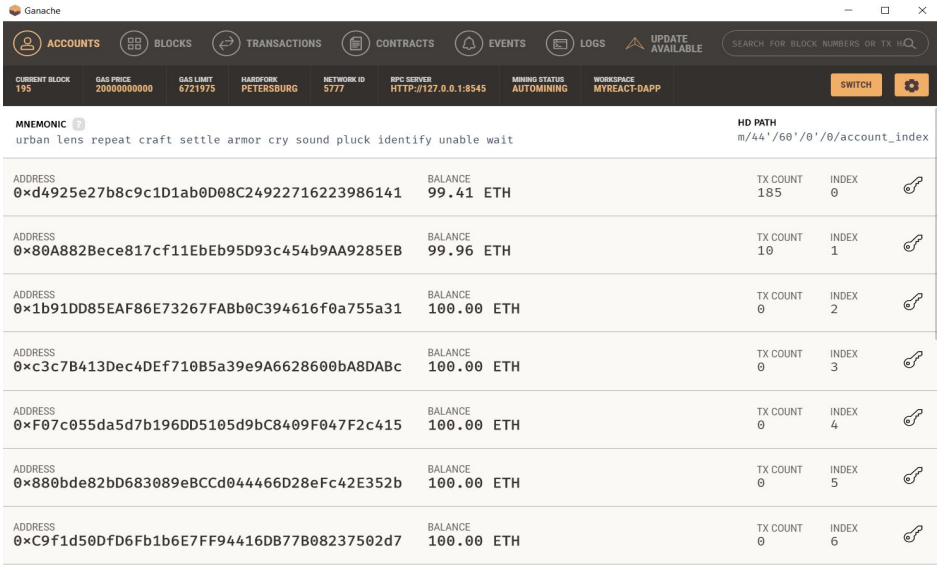
เมื่อทำการ Deploy Contract เสร็จแล้วจะได้ Contract Address โดย Contact Address จะอยู่ในบล็อกเชน แล้วทำการเปิดใช้งาน Application โดยใช้คำสั่ง npm start ผ่าน Command Line ซึ่งเป็นคำสั่งสำหรับการเปิดใช้ Application ของ React Framework และเมื่อใช้คำสั่งเรียบร้อยแล้ว Application จะถูกเปิดโดยอัตโนมัติ

โดย Application ของเราจะมีฟังก์ชันการทำงานหลัก 3 ฟังก์ชัน ได้แก่

- ฟังก์ชันการเข้าสู่ระบบ
- ฟังก์ชันการกรอกรายละเอียดของบทความ
- ฟังก์ชันอัปโหลดบทความผ่าน

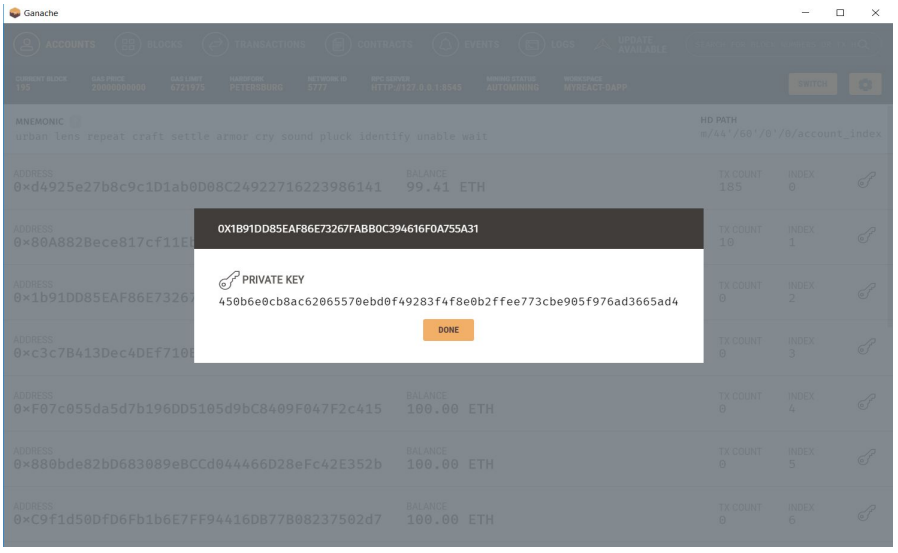
ซึ่ง Application ของเราทำการทดสอบโดย Ethereum Test Network

ฟังก์ชันการเข้าสู่ระบบ

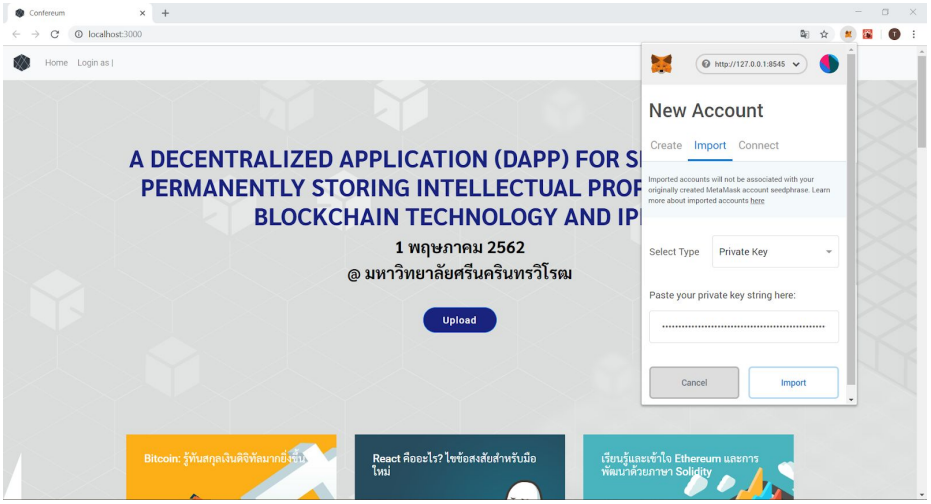


ภาพที่ 5-2 หน้าตาของ Ganache

จากภาพที่ 5-2 ผู้ใช้ทำการเปิด Ganache เพื่อรัน Application ขึ้นบน Ethereum Test Network ซึ่ง Ganache มีหน้าที่เป็นแบบจำลอง Ethereum Test Network

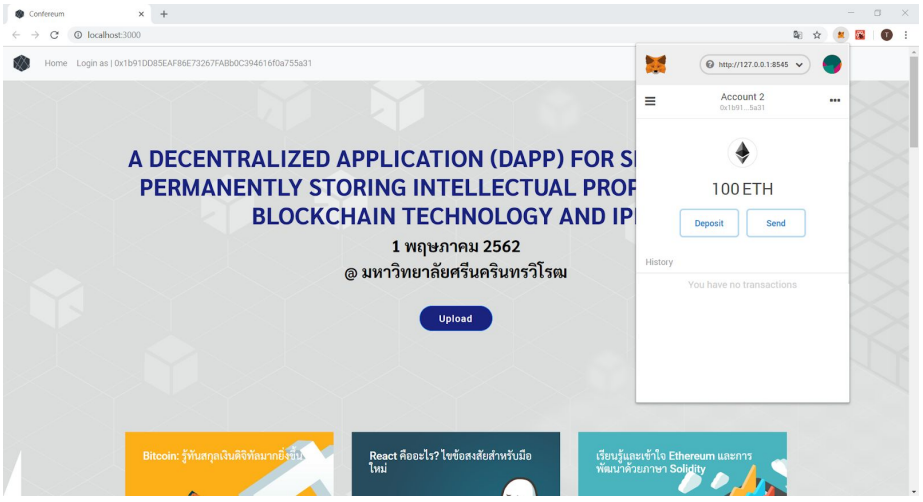


ภาพที่ 5-3 แสดง Private key ใน Ganache



ภาพที่ 5-4 แสดงการนำ Private key ใน Ganache เพื่อสร้าง Account

ผู้ใช้นำ Account จาก Ganache เพื่อทำการเข้าสู่ระบบผ่าน MetaMask ซึ่งในการเข้าสู่ระบบสำหรับการสร้าง Account ใหม่จึงต้องใช้ Private key จาก Ganache เพื่อทำการ Import Account ซึ่งเมื่อทำการ Import Account เรียบร้อยแล้ว



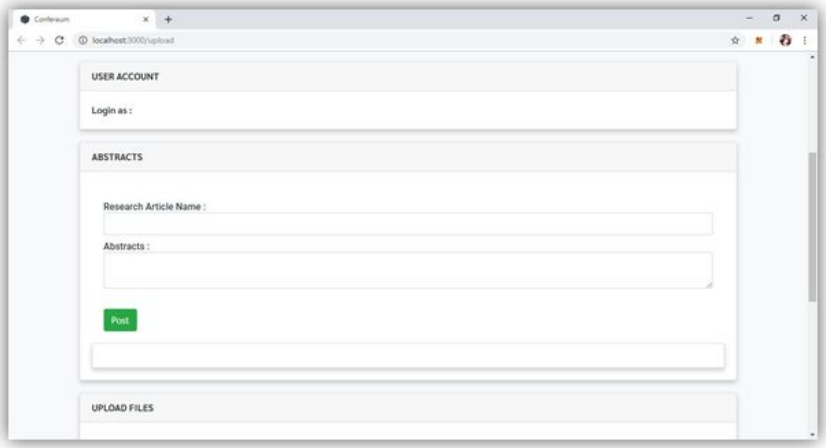
ภาพที่ 5-5 การเข้าสู่ระบบอย่างสมบูรณ์

โดยผู้ใช้งานสามารถตรวจสอบ Account ที่ทำการเข้าสู่ระบบซึ่ง Account จาก หน้า Application และ MetaMask จะต้องเป็น Accountเดียวกัน

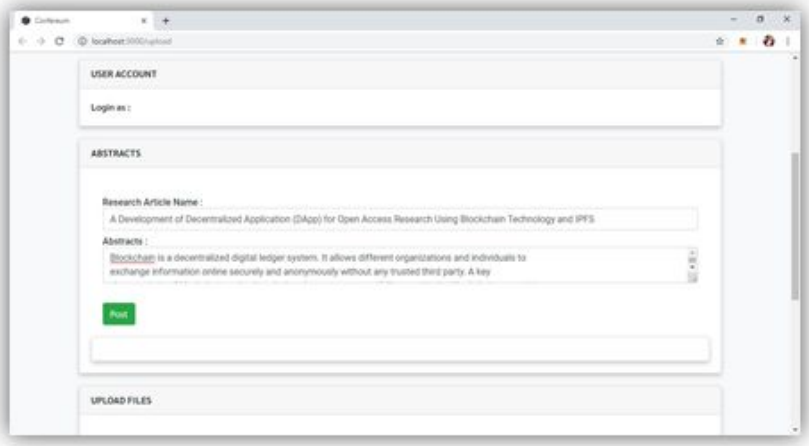
ฟังก์ชันการกรอกรายละเอียดของบทความ

รายละเอียดของการกรอกบทความมี 2 อินพุตได้แก่

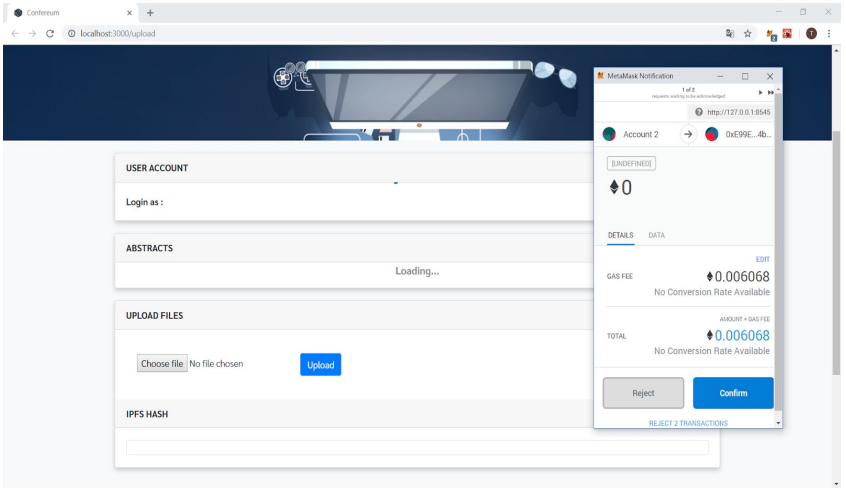
- ชื่อของบทความ
- บทคัดย่อของบทความ



ภาพที่ 5-6 ฟอร์มการกรอกรายละเอียดของบทความ

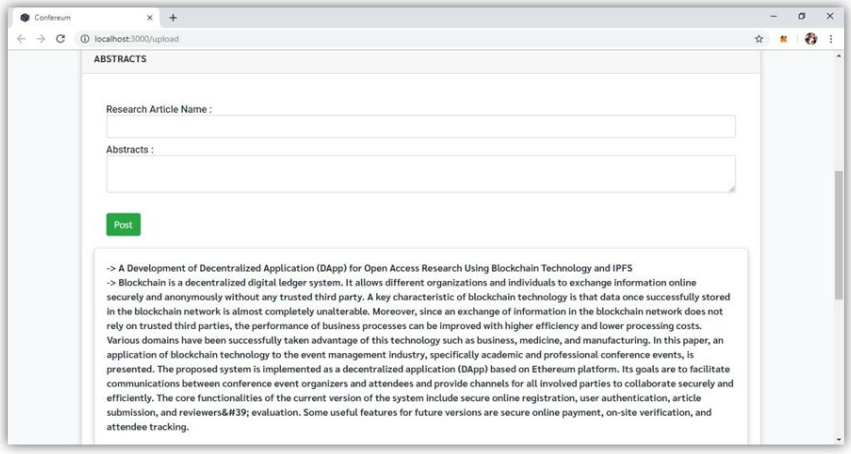


ภาพที่ 5-7 แสดงถึงการกรอกรายละเอียดของบทความ



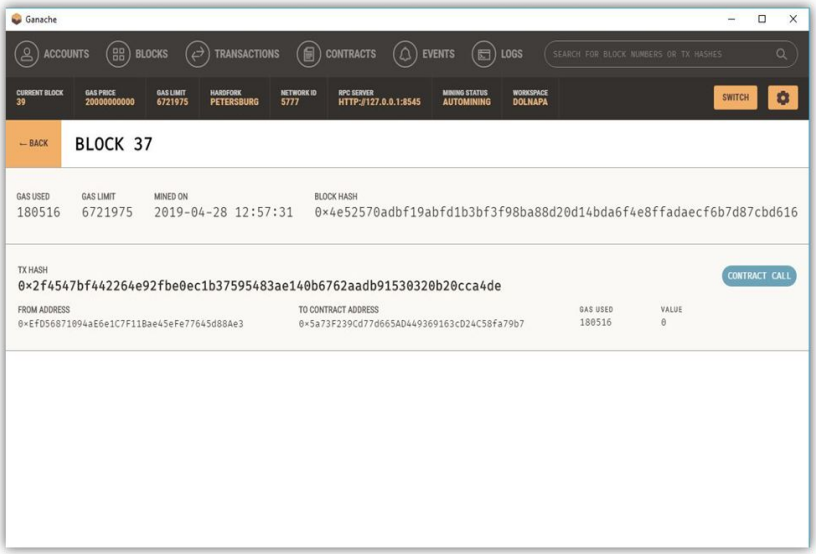
ภาพที่ 5-8 การยืนยัน Transaction ของการกรอกรายละเอียดของบทความ

จากภาพที่ 5-6,5-7,5-8 ผู้ใช้กรอกรายละเอียดของบทความเรียบร้อยแล้วทำการกดปุ่ม Post และ MetaMask จะแสดง Pop-up เพื่อการยืนยัน Transaction ของชื่อบทความและบทคัดย่อของบทความ

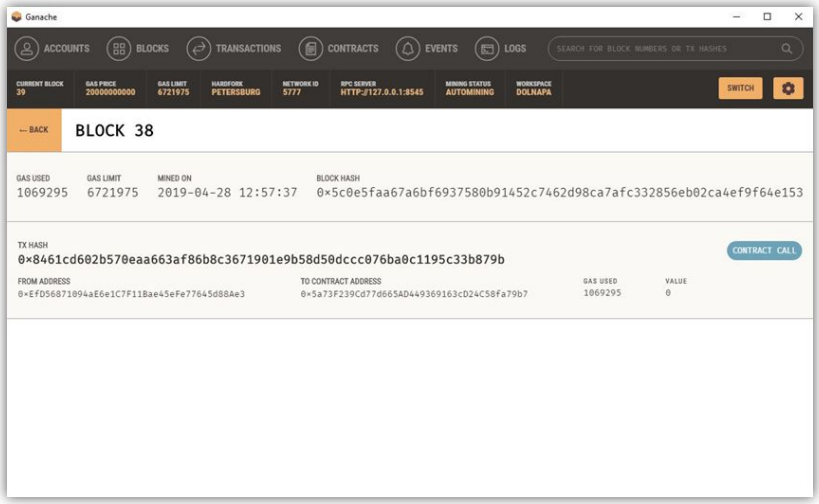


ภาพที่ 5-9 การแสดงของรายละเอียดบทความ

ดังนั้นข้อมูลของบทความถูกบันทึกลงในบล็อกเชน โดยผู้ใช้สามารถตรวจสอบ Tranaction ต่างๆ จาก Ganache ดังภาพ 5-10 และ ภาพ 5-11

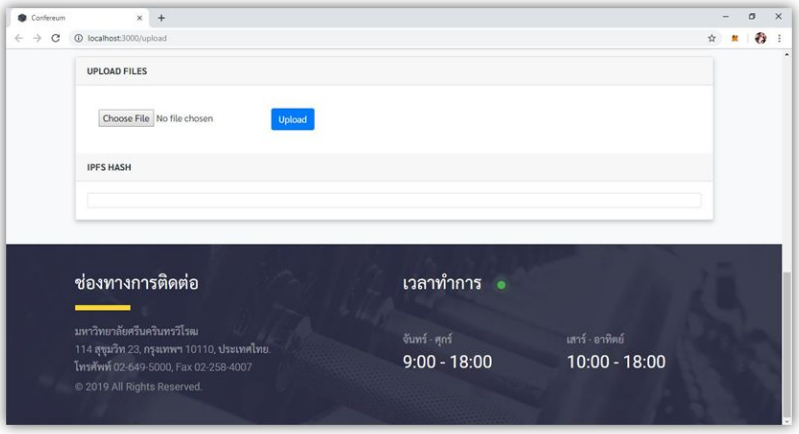


ภาพที่ 5-10 รายละเอียดของ Transaction Block ที่ 37

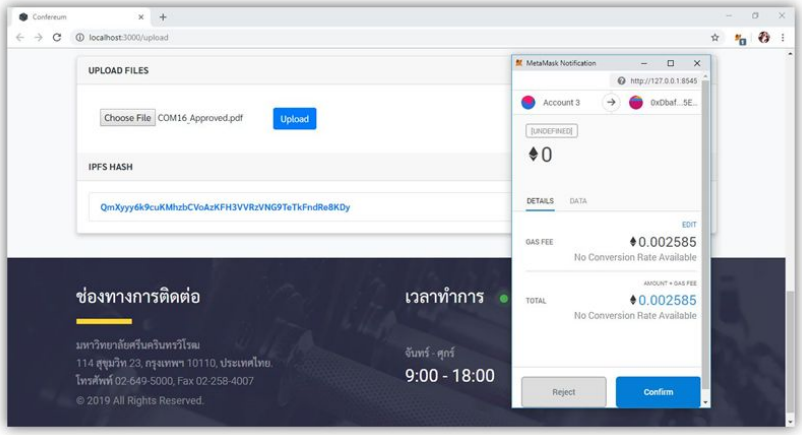


ภาพที่ 5-11 รายละเอียดของ Transaction Block ที่ 38

ฟังก์ชันอัปโหลดบทความ

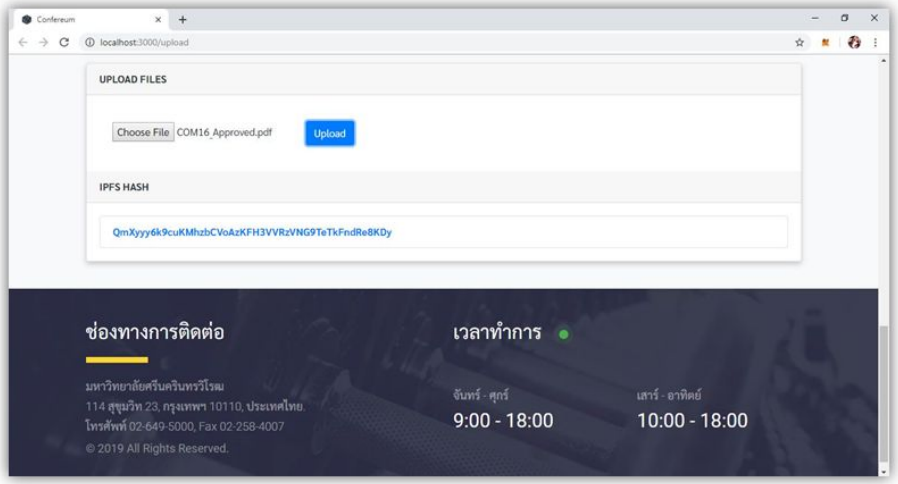


ภาพที่ 5-12 ฟอร์มการอัปโหลดของบทความ



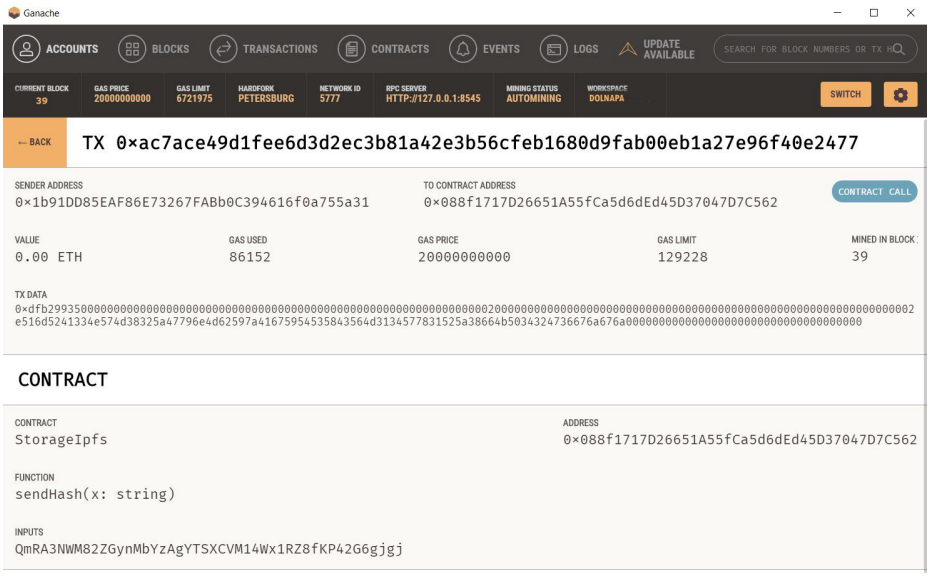
ภาพที่ 5-13 แสดงถึงการอัปโหลดบทความ

จากภาพที่ 5-13 เมื่อผู้ใช้ทำการเลือกไฟล์บทความที่ต้องการอัปโหลด และ Metamask จะแสดง Notification เพื่อยืนยันการอัปโหลดบทความ เมื่อผู้ใช้คลิกปุ่ม Upload ไฟล์ที่อัปโหลดจะถูกแปลงเป็น Hash สำหรับบทความที่อัปโหลดไว้ ดังภาพ 5-14



ภาพที่ 5-14 แสดงค่า Hash ของไฟล์ที่อัปโหลด

ภาพที่ 5-15 แสดงถึงไฟล์อัปโหลดโดยผ่าน IPFS



ภาพที่ 5-16 รายละเอียดของ Transaction ของไฟล์ที่อัปโหลด

จากภาพที่ 5-16 เมื่อทำการอัปโหลดไปแล้วข้อมูลถูกบันทึกลงในบล็อกเชน ซึ่งโดยปกติไฟล์ที่อัปโหลดมีขนาดค่อนข้างใหญ่ลงบนบล็อกเชน เราจึงใช้ Dirtributed File System นั่นคือ IPFS โดยเราทำการอัปโหลดเข้า IPFS และนำ Hash ที่เชื่อมโยงกับไฟล์ที่อัปโหลดสำหรับการอัปโหลดลงบล็อกเชนแทน

บทที่ 6
สรุปผล อภิปราย และข้อเสนอแนะ

6.1. สรุปผลการศึกษาค้นคว้า

แอปพลิเคชันชนิดกระจายอำนาจการควบคุมสำหรับการจัดเก็บทรัพย์สินทางปัญญาอย่างถาวรและปลอดภัยด้วยเทคโนโลยีบล็อกเชนและไอพีเอฟเอสนี้ ผู้ใช้สามารถเข้าสู่ระบบ Decentralized Application (DApp) ผ่าน MetaMask ได้ ผู้ใช้สามารถกรอกรายละเอียดของบทความและบันทึกข้อมูลลงบล็อกเชนได้ ผู้ใช้สามารถอัปโหลดไฟล์บทความผ่าน IPFS ได้ ตามที่ได้ออกแบบการพัฒนาระบบไว้ข้างต้น ซึ่งบทความที่ทำการอัปโหลดนั้น จะไม่สามารถเปลี่ยนแปลงเนื้อหาของบทความได้ จึงมั่นใจได้ว่าบทความนี้ผู้ใช้งานที่ซึ่งเป็นผู้เขียนบทความนี้อนุญาตให้ผู้อื่นเห็นบทความนี้ได้ เนื่องจากผู้ใช้งานเป็นผู้ทำการอัปโหลดบทความนี้ด้วยตนเอง หากมีบทความที่คล้ายกันผู้ใช้งานที่ซึ่งเป็นผู้เขียนบทความนี้ จะสามารถตรวจสอบ Account ของผู้ใช้งานที่อัปโหลดบทความนั้นได้

6.2. ปัญหาและอุปสรรคในการดำเนินงาน

เนื่องจากภาษา Solidity เป็นภาษาใหม่จึงใช้เวลาศึกษานานและเวอร์ชันของ Solidity มีการอัปเดตเรื่อยๆ ทำให้ต้องปรับวิธีการเขียนบ่อยครั้ง

6.3. ข้อเสนอแนะ

ในอนาคตสามารถนำไปต่อยอดได้ เช่น ให้ผู้ใช้ที่ต้องการอ่านบทความวิจัยสามารถเข้ามาอ่านบทความที่ผู้ใช้คนอื่นอัปโหลดบทความไว้ได้ ซึ่งมั่นใจได้ว่าบทความนี้ได้รับอนุญาตจากผู้ใช้เจ้าของบทความจริงๆ

บรรณานุกรม

[1] A. E. T. V. L. Asaph Azaria, "MedRec: Using Blockchain for Medical Data Access and Permission Management," 2nd International Conference on Open and Big Data, 2016.

[2] J. Z. S. S. J. L. D. L. Xueping Liang, “Integrating Blockchain for Data Sharing and Collaboration in Mobile Healthcare Applications,” IEEE, 2017.

[3] R.R.M.R. J.O.M. Xiaochen Zheng, “Blockchain-based Personal Health Data Sharing System Using Cloud Storage,” IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom), 2018.

[4] A. C. Alex Norta, "Commercial Property Tokenizing With Smart Contracts," 2018 IEEE, 2018.

[5] D. Yin Liao, W. XueHong, “Design of A Blockchain-based Lottery System for Smart Cities Applications,” 2017 IEEE 3rd International Conference on Collaboration and Internet Computing
,2017

[6] Blockchain Technology. สืบค้นเมื่อ 7 พฤศจิกายน 2561, จาก https://thaipublica.org/2018/06/kkp-financial-literacy-21/ , http://www.maruey.com/article/contentinjournal0024.html, https://en.wikipedia.org/wiki/Blockchain

[7] Decentralized application. สืบค้นเมื่อ 26 มกราคม 2562, จาก http://medium.com/@prick.aunt/ตอนที่4-blockchain-3-0-เมื่อเงินดิจิตอลไม่ใช่แค่การเก็งกำไรอีกต่อไป-7a30d606b104, https://medium.com/@chawansit/ย้อนเวลาหา-ethereum-บทความปูพื้น-76cc97cbda13, https://en.wikipedia.org/wiki/Decentralized_application

[8] Ethereum. สืบค้นเมื่อ 7 พฤศจิกายน 2561, จาก https://gawao.com/ethereum-คืออะไร/

[9] Event Management. สืบค้นเมื่อ 20 มกราคม 2562, จาก https://www.swisseducation.ac/event-management/

[10] Ganache. สืบค้นเมื่อ 26 มกราคม 2562, จาก https://ethereum.stackexchange.com/questions/58093/difference-between-ganache-and-truffle, https://medium.com/coinmonks/ganache-truffle-framework-64b01f4ca200

[11] Hash Function. สืบค้นเมื่อ 22 พฤศจิกายน 2561, จาก https://medium.com/@iyawatkongmalai/blockchain-101-เข้าใจ-blockchain-แบบง่าย-3429082ff96

[12] MetaMask. สืบค้นเมื่อ 26 มกราคม 2562, จาก https://metamask.io/

[13] Pet Shop. สืบค้นเมื่อ 25 มกราคม 2562, จาก

https://truffleframework.com/tutorials/pet-shop,
https://medium.com/@lippoldt331/dapp-deployment-petshop-now-and-from-scratch-aeca69e7

[14] Smart Contract. สืบค้นเมื่อ 14 พฤศจิกายน 2561, จาก
http://dv.co.th/blog-en/smart-contract-blockchain/,
https://siamblockchain.com/2017/06/04/blockchain-คืออะไร/,
https://en.wikipedia.org/wiki/Smart_contract

[15] Truffle. สืบค้นเมื่อ 26 มกราคม 2562, จาก https://medium.com/dcen/ethereum-smart-contracts-with-next-js-part-0-รู้จักสิ่งต่างๆ-ในการพัฒนา-b3f3adcc2ac1

[16] IPFS. สืบค้นเมื่อ 20 มีนาคม 2562, จาก https://ipfs.io/,
https://medium.com/@artiya4u/what-is-ipfs-2d54241a7b4b