

ออกแบบและพัฒนาสถาปัตยกรรมแบบไม่รวมศูนย์เพื่อจัดการควบคุมการเข้าถึงข้อมูลสุขภาพสำหรับเครือข่ายโรงพยาบาลโดยใช้บล็อกเชนส่วนบุคคล

ไตรศักดิ์ ไตรเสนีย์, อมิตา มงคลปริตยาไชย และ ธีรพงศ์ ลีลานุกภาพ

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
Emails: traisak_tra@outlook.com, amt_mpch@hotmail.com, teerapong@it.kmitl.ac.th

บทคัดย่อ

เทคโนโลยีบล็อกเชนมีศักยภาพเป็นอย่างมากในการนำไปสร้างโครงสร้างพื้นฐานต่าง ๆ สำหรับทุก ๆ ภาคส่วนในด้านเทคโนโลยีสารสนเทศซึ่งนำมาสู่การสร้างโซลูชันที่มีค่าใช้จ่ายน้อยลงและความปลอดภัยที่สูงขึ้น สำหรับงานในด้านสาธารณสุข ความก้าวหน้าด้านเทคโนโลยีสารสนเทศทำให้เกิดฐานข้อมูลด้านข้อมูลสุขภาพขนาดมหึมาและเปลี่ยนวิธีที่ผู้ใช้แต่ละคนมีการปฏิสัมพันธ์กับข้อมูลสุขภาพของตนเอง ไม่ว่าจะเป็นการบันทึกและการเข้าถึงข้อมูลเหล่านั้น ผ่านทางอุปกรณ์สวมใส่ด้านสุขภาพและมือถือสมาร์ทโฟนต่าง ๆ อีกทั้งสถานพยาบาลยังไม่มีส่วนที่รองรับในการเข้าถึงข้อมูลเหล่านี้ ดังนั้นระบบเทคโนโลยีสารสนเทศด้านสุขภาพจึงต้องการสถาปัตยกรรมใหม่ในการบันทึกข้อมูลสุขภาพ ซึ่งเปลี่ยนรูปแบบจากฐานข้อมูลแบบรวมศูนย์และบนกลุ่มเมฆไปเป็นแบบไม่รวมศูนย์กลาง

บทความนี้นำเสนอการวิเคราะห์และออกแบบสถาปัตยกรรมแบบไม่รวมศูนย์กลางสำหรับการเก็บบันทึกข้อมูลสุขภาพโดยใช้บล็อกเชนส่วนบุคคล โครงการนี้ได้รับความร่วมมือจากเครือข่ายโรงพยาบาลเอกชนที่ใหญ่ที่สุดในประเทศไทย ซึ่งผู้พัฒนาได้ใช้เป็นการศึกษาในการออกแบบและพัฒนาบทความนี้ได้นำเสนอการใช้งานบล็อกเชนส่วนบุคคลและสมาร์ทคอนแทรคท์เป็นผู้จัดการควบคุมการเข้าถึงข้อมูลสุขภาพซึ่งถูกจัดเก็บแบบกระจายศูนย์ นอกจากนี้บทความนี้ยังทำการเปรียบเทียบเครื่องมือต่างที่ใช้เทคโนโลยีบล็อกเชน โครงการนี้นำเสนอความท้าทายใหม่เพื่อพิสูจน์แนวคิดที่จะนำเอาบล็อกเชนมาใช้เพื่อเก็บข้อมูลด้านสุขภาพ

คำสำคัญ-- ข้อมูลทางการแพทย์; บล็อกเชน; อีเธอเรียม; สมาร์ทคอนแทรคท์; แอปพลิเคชันแบบไม่รวมศูนย์; การเข้ารหัส; การให้บริการบล็อกเชน

Abstract

Blockchain technology has a large potential to create new foundations for every segment in information technology (IT) with a lower-cost and higher-security solution. For health care, IT advancement has produced massive databases of health data and changed the way in which individuals interact with their own health data, by recording or viewing them, via wearable devices and smartphones. As a result, health IT systems require a new architecture to record health

information, shifting to a decentralized database from traditional centralized or cloud ones. In this paper, we describe the analysis and design of the decentralized architecture for health record using private Blockchain. We have collaborated with the largest group of private hospitals in Thailand and using its requirements as our case study. Our proposal involves the use of a private Blockchain and its smart contract as an access control manager to health records that are stored off Blockchain, whereas actual health records are stored in another native base layer of distributed storage. We compare and contrast different tools for Blockchain technologies. This work can be viewed as a new challenge to be the first proof-of-concept in implementing Blockchain in Thailand as Blockchain-as-a-service (BaaS) for medical records.

Keywords— Medical Record; Ethereum; Blockchain; Smart contract; Decentralized Application; Encryption; Blockchain as a Service

1. บทนำ

โรงพยาบาลภายในประเทศไทยส่วนใหญ่มีการจัดเก็บฐานข้อมูลแบบรวมศูนย์ (Centralized Database) อันเนื่องมาจากการดูแลรักษาที่ง่ายและความรวดเร็วในการจัดตั้งระบบ โดยแต่ละโรงพยาบาลมีการใช้รูปแบบโครงสร้างข้อมูลแตกต่างกันไปขึ้นอยู่กับผู้พัฒนา แต่รูปแบบการเก็บข้อมูลดังกล่าวอาจส่งผลให้เกิดปัญหาจากการเติบโตของข้อมูลในอนาคต ทั้งด้านค่าใช้จ่ายในการเพิ่มปริมาณฐานข้อมูลและด้านความเสถียรของระบบ หากเกิดข้อผิดพลาดกับคลังข้อมูล จะส่งผลให้ส่วนเชื่อมต่อที่เหลือทำงานผิดพลาดตามไปด้วย นอกจากนี้ระบบในปัจจุบันไม่ได้มีการอำนวยความสะดวกในการเข้าถึงจากผู้ใช้งานต่าง ๆ โดยตรง เช่น ผู้ป่วยซึ่งเป็นเจ้าของข้อมูลที่ต้องการนำข้อมูลไปใช้เข้ารับการรักษาที่โรงพยาบาลอื่น นักวิจัยที่ต้องการข้อมูลเพื่อประกอบงานวิจัยโดยไม่เกิดปัญหาทางจริยธรรมจากการละเมิดความเป็นส่วนตัวของผู้ป่วยและโรงพยาบาลอื่น ๆ ที่ต้องการประวัติการรักษาของคนไข้ย้อนหลัง

จากนั้นในปี 2016 ได้มีการเสนอแนวทางการใช้สมาร์ทคอนแทรคท์ในการจัดการเก็บข้อมูลทางการแพทย์แบบไม่รวมศูนย์ ดังเช่นงานวิจัยและพัฒนาตัวต้นแบบ [1], [2] เพื่อควบคุมการเข้าถึงข้อมูลทางการแพทย์โดยใช้บล็อกเชน การศึกษาระบบบิโณจันย์โรคอัตโนมัติและสร้างระบบการ

แลกเปลี่ยนข้อมูลสาธารณสุขที่เชื่อถือได้โดยใช้บล็อกเชน การใช้โครงสร้างพื้นฐานข้อมูลลงลายเซ็นแบบไม่ใช้กุญแจ (Keyless Signature Infrastructure) [2] กับระบบ e-hospital ในประเทศเอสโตเนีย โดยผลที่ได้ตามมาจากงานวิจัยข้างต้นคือระบบที่ใช้รูปแบบโครงสร้างแบบไม่รวมศูนย์ในการควบคุมสิทธิให้เป็นไปตามที่แต่ละโรงพยาบาลกำหนด ช่วยลดต้นทุนค่าใช้จ่ายในด้านอุปกรณ์ ช่วยให้เกิดการเชื่อมต่อและทำงานร่วมกันได้ดีขึ้น มีความน่าเชื่อถือโปร่งใสมากขึ้นเนื่องจากตรวจสอบย้อนหลังได้ง่ายและยากต่อการลักลอบแก้ไขจากบุคคลภายนอก ส่งผลให้ข้อมูลมีความน่าเชื่อถือ

2. เทคโนโลยี เครื่องมือ และงานวิจัยที่เกี่ยวข้อง

ในส่วนนี้ได้ทำการอธิบายถึงเทคโนโลยีและเครื่องมือที่นำมาใช้ในการออกแบบสถาปัตยกรรมแบบกระจายศูนย์และหลักการพัฒนาระบบสถาปัตยกรรมเพื่อจัดการควบคุมการเข้าถึงข้อมูล รวมถึงงานวิจัยที่เกี่ยวข้องซึ่งนำเสนอการใช้เทคโนโลยีบล็อกเชนมาใช้ในการออกแบบสถาปัตยกรรมสำหรับข้อมูลสุขภาพของโรงพยาบาล

2.1. เทคโนโลยีที่เกี่ยวข้อง

2.1.1 Blockchain

บล็อกเชน [3] เป็นเทคโนโลยีสำหรับเก็บข้อมูลแบบไม่รวมศูนย์ซึ่งถูกออกแบบมาเพื่อเน้นความปลอดภัย ความถูกต้องของข้อมูล และจัดการข้อมูลที่มีแนวโน้มที่จะเพิ่มสูงขึ้นอย่างต่อเนื่อง โครงสร้างของการเก็บข้อมูลในบล็อกเชนนี้นั้นจะถูกจัดเก็บเป็นลักษณะของชุดข้อมูลที่มีหน่วยเป็น "บล็อก" ซึ่งมีการเก็บข้อมูลการลงเวลาที่แก้ไขล่าสุดและความสัมพันธ์ระหว่างบล็อก โดยแสดงการอ้างอิงของบล็อกข้อมูลก่อนหน้า ในปัจจุบันบล็อกเชนได้มีการพัฒนาอย่างต่อเนื่องจนถึงยุค 3.0

Blockchain 1.0 - Crypto Currency ในช่วงแรกที่เทคโนโลยีบล็อกเชนถูกสร้างขึ้นมานั้นมีประโยชน์ในวงการการเงินการธนาคารสูงมาก เนื่องจากมีธุรกรรมเกิดขึ้นในทุก ๆ ชั่วโมง แต่บัญชีของแต่ละบุคคลถูกเก็บเป็นความลับเสมอกันโดยมีเพียงธนาคารเท่านั้นที่ทราบ จึงถูกนำไปใช้ในรูปแบบของสกุลเงินที่ทุกคนสามารถทำธุรกรรมร่วมกันและช่วยกันตรวจสอบความถูกต้องได้อีกทั้งยังไม่สามารถสร้างข้อมูลที่เป็นเท็จอีกด้วย ซึ่งในช่วงนี้จะมีสกุลเงินที่รู้จักกันทั่วไป เช่น BitCoin MasterCoin Ripple

Blockchain 2.0 - Smart Contract ต่อมาเทคโนโลยีได้พัฒนาความสามารถในการเขียนโปรแกรมเข้าไปควบคุมเงื่อนไขของบล็อกเชนได้โดยไม่ต้องพึ่งตัวกลางใด ๆ เช่น ผู้ซื้อขายสินค้าออนไลน์และผ่านทางโซเชี่ยลมีเดีย สามารถลดความเสี่ยงของการชำระเงินลงได้ การโอนเงินข้ามประเทศจะสามารถสำเร็จภายในไม่กี่นาที จากเดิมที่ใช้เวลาหลายวัน การอนุมัติสินเชื่อที่สามารถทำได้อย่างรวดเร็ว หรือการเคลมประกันแบบอัตโนมัติโดยไม่จำเป็นต้องใช้ออกสาร รวมถึงยังเป็นการเพิ่มความโปร่งใสในการทำธุรกรรม ที่จะช่วยให้การกำกับดูแลของหน่วยงานภาครัฐเป็นไปได้ อย่างมีประสิทธิภาพ ซึ่งรูปแบบนี้สามารถบังคับใช้สิ่งที่กำหนดไว้ในสัญญาได้อัตโนมัติ มีความถูกต้อง โปร่งใส เพิ่มความน่าเชื่อถือและเพิ่มประสิทธิภาพการทำงาน ช่วยลดขั้นตอนการทำงานและลดข้อผิดพลาดที่อาจเกิดจาก

มนุษย์ (Human Error) ซึ่งกลายมาเป็นโครงสร้างพื้นฐานของแอปพลิเคชันแบบไม่รวมศูนย์ (Decentralized Application)

Blockchain 3.0 - แอปพลิเคชันแบบไม่รวมศูนย์ (Decentralized Application - Dapp) บล็อกเชนพัฒนาเข้าสู่ยุคที่นำเทคโนโลยีนี้มาพัฒนาในเชิงธุรกิจอย่างจริงจัง ซึ่งแอปพลิเคชันในยุคนี้จะต่างจากยุคก่อนที่ไม่จำเป็นต้องทำงานจากเซิร์ฟเวอร์ส่วนกลางที่ใดที่หนึ่ง แต่จะทำงานอยู่ตลอดเวลาไม่มีวันปิดตราปิดที่ยังมีบุคคลใดบุคคลหนึ่งบนโลกสั่งทำงาน

2.1.2 Decentralized Storage

การจัดเก็บข้อมูลแบบไม่รวมศูนย์ [4] คือระบบฐานข้อมูลที่มีการเก็บข้อมูลไว้ในระบบคอมพิวเตอร์หลาย ๆ เครื่อง ซึ่งเครื่องที่ติดตั้งนั้นอาจตั้งอยู่ตามที่แตกต่างกัน โดยระบบคอมพิวเตอร์เหล่านั้นจะสื่อสารกันผ่านระบบเครือข่ายคอมพิวเตอร์ ซึ่งผู้ใช้สามารถเรียกใช้ข้อมูลจากฐานข้อมูลที่อยู่ในเครื่องใดก็ได้ และผู้ใช้ไม่จำเป็นต้องรับรู้ข้อมูลที่ต้องการนั้นจัดเก็บอยู่บนเครื่องใดและมองเห็นเป็นระบบการจัดเก็บข้อมูลแบบรวมศูนย์

ลักษณะสำคัญของการจัดเก็บข้อมูลแบบไม่รวมศูนย์คือมีความเป็นอิสระของแต่ละส่วนงาน ไม่ขึ้นกับส่วนกลาง ทำให้สามารถขยายระบบได้ง่ายขึ้นและสามารถลดข้อผิดพลาดเรื่องจุดเดียวของความล้มเหลว (Single point of failure) ได้ดีกว่าแบบรวมศูนย์

2.1.3 Encryption

เทคโนโลยีการเข้ารหัสข้อมูล [5] เป็นกระบวนการสำหรับการแปรรูปข้อมูลอิเล็กทรอนิกส์ธรรมดาให้อยู่ในรูปแบบที่บุคคลทั่วไป ไม่สามารถอ่านเข้าใจได้ โดยทั่วไปการเข้ารหัสจะกระทำก่อนการจัดเก็บข้อมูลหรือก่อนการส่งข้อมูล โดยการนำข้อมูลอิเล็กทรอนิกส์ธรรมดากับกุญแจ (Key) ซึ่งเป็นตัวเลขซึ่งถูกสุ่ม มาผ่านกระบวนการทางคณิตศาสตร์ และผลที่ได้คือข้อมูลที่ถูกเข้ารหัส ขั้นตอนที่กำลังจะเรียกว่า "การเข้ารหัส" (Encryption) และเมื่อต้องการอ่านข้อมูล จะต้องนำเอาข้อมูลที่ถูกเข้ารหัสและกุญแจมาผ่านกระบวนการทางคณิตศาสตร์ จะได้ผลลัพธ์ที่เป็นข้อมูลดั้งเดิม ขั้นตอนนี้จะเรียกว่า "การถอดรหัส" (Decryption) ระบบเข้ารหัสสามารถแบ่งตามวิธีการใช้กุญแจได้เป็น 2 วิธีคือ ระบบเข้ารหัสแบบกุญแจสมมาตรและอสมมาตร

ระบบเข้ารหัสแบบกุญแจสมมาตร (Symmetric-key cryptography) คือการเข้ารหัสข้อมูลด้วยกุญแจเดียว ทั้งผู้ส่งและผู้รับใช้กุญแจแบบเดียวกันทั้งการเข้ารหัสและถอดรหัส โดยวิธีการนี้ผู้ส่งและผู้รับจะต้องตกลงกุญแจในการเข้ารหัสข้อมูล

ระบบเข้ารหัสแบบกุญแจอสมมาตร (Asymmetric-key cryptography หรือ Public Key Technology) การเข้ารหัสวิธีนี้จะใช้หลักกุญแจคู่ทำการเข้ารหัสและถอดรหัส โดยกุญแจจะประกอบไปด้วย กุญแจส่วนตัว (Private key) และกุญแจสาธารณะ (Public key) โดย หากใช้กุญแจใดเข้ารหัส จะต้องใช้อีกกุญแจหนึ่งถอดรหัส สำหรับการเข้ารหัสและถอดรหัสด้วยกุญแจคู่นี้จะใช้ฟังก์ชันทางคณิตศาสตร์เข้ามาช่วยโดยที่ฟังก์ชันทางคณิตศาสตร์ที่นำมาใช้ได้รับการพิสูจน์แล้วว่าจะมีเฉพาะกุญแจคู่ของกันเท่านั้นที่จะสามารถถอดรหัสได้ ไม่สามารถนำกุญแจคู่อื่นมาถอดรหัสได้อย่างเด็ดขาด

Hash Function คือการเข้ารหัสแบบกุญแจสมมาตรประเภทหนึ่ง ซึ่งเป็นการเข้ารหัสทางเดียว โดยจะสร้างลายเซ็นดิจิทัล (Digital Signature) ของข้อมูลอิเล็กทรอนิกส์ที่ *ไม่* สามารถถอดรหัสกลับมาได้ คุณสมบัติหลักของ Hash Function คือค่า Hash ของข้อความเดิมจะต้องเหมือนกันเสมอในทางตรงกันข้าม ข้อความที่ไม่เหมือนกันจะไม่มีทาง Hash ได้ค่าเดียวกัน ความยาวของข้อความที่ผ่านการ Hash จะต้องเท่ากันไม่ว่าจะเป็นเพียงข้อความสั้น ๆ หรือทั้งไฟล์ก็ตาม ซึ่งค่า Hash ที่ได้มาจะ *ไม่* สามารถนำไปถอดรหัสกลับเพื่อหาข้อความเดิมได้ Bitcoin ได้นำเอาคุณสมบัติเหล่านี้มาใช้ในการตรวจสอบความถูกต้องของรายการโดยการสร้างบัญชีแยกประเภท (Ledger Address) ของแต่ละบัญชีเพื่อใช้แทนเลขที่บัญชี โดยเลขที่บัญชีของ Bitcoin จะสร้างจาก Public key ที่นำมาผ่านกระบวนการ Hash Function

2.2. เครื่องมือที่เกี่ยวข้อง

2.2.1. อีเธอเรียมบล็อกเชน (Ethereum blockchain)

แนวคิดอีเธอเรียมบล็อกเชน [6] เริ่มพัฒนาโดย Vitalik Buterin นักพัฒนาชาวรัสเซียในปี 2013 ซึ่ง Vitalik เคยเป็นผู้ร่วมพัฒนา Bitcoin มาก่อนแต่พบข้อบกพร่องและข้อจำกัดของ Bitcoin ทำให้เขาเริ่มสร้างอีเธอเรียมโดยสิ่งสำคัญที่เพิ่มเข้ามาคือสมาร์ทคอนแทรคท์ เทคโนโลยีที่ใช้ในอีเธอเรียมบล็อกเชน ประกอบไปด้วย (1) การเข้ารหัสข้อมูลด้วยกุญแจสาธารณะ (2) การเข้ารหัสด้วย Hash-function (3) เครือข่ายแบบ Peer-to-peer (4) EVM (Ethereum Virtual Machine) และ (5) Proof-of-stake โดยเมื่อผู้ใช้ทำการสร้าง Transaction รายการ Transaction นั้นจะถูกส่งไปยังเครือข่าย Peer-to-peer ที่สมาชิกในเครือข่ายจะเก็บข้อมูลเดียวกันและมีการใช้งานเครือข่ายจำลองของอีเธอเรียม (EVM) เพื่อที่จะคอยตรวจสอบการเปลี่ยนแปลง และสถานะต่าง ๆ ในเครือข่าย จากนั้นกลไก Proof-of-stake [7] ที่รวมอยู่ในระบบจะทำหน้าที่ในเลือกผู้ที่เหมาะสมมาทำการรับรองความถูกต้องโดยใช้หลักการ Merkle Tree [8] ผู้ที่ทำการรับรองความถูกต้องและสามารถดำเนินการคำสั่งตามสมาร์ทคอนแทรคท์ได้ จะได้รับค่าธรรมเนียมเป็นสิ่งตอบแทน ผู้ที่ได้รับค่าตอบแทนจะทำการจัดเก็บ Transaction ดังกล่าวลงในบล็อกพร้อมกับนำข้อมูลส่วนหัวของบล็อกก่อนหน้า (Previous block header) มาผ่านกระบวนการ Hash Function และทำการประกาศออกไปในเครือข่ายให้สมาชิกทำสำเนาข้อมูลรูปแบบเดียวกันขึ้นมาเพื่อป้องกันการลักลอบแก้ไขทั้งนักพัฒนาและผู้ใช้ทั่วไปสามารถที่จะใช้ประโยชน์จากเครือข่ายแบบไม่รวมศูนย์ที่ช่วยกระจายความสามารถและอำนาจการควบคุมออกจากระบบแบบศูนย์กลาง เพื่อลดค่าใช้จ่ายจากการทำงานแบบ proof-of-work [7] ที่ต้องใช้อุปกรณ์ประมวลผลจำนวนมาก ในขณะที่สามารถเพิ่มขีดจำกัดด้านความเร็วและยืดหยุ่น ซึ่งสิ่งเหล่านี้คือจุดประสงค์หลักของแนวคิดการสร้างบล็อกเชนแบบอีเธอเรียม

2.2.2. Test RPC (Simulated blockchain private-network)

ในการพัฒนาและทดสอบคอนแทรคท์นั้นจำเป็นที่จะต้องใช้ Blockchain network ในการติดตั้งเพื่อเรียกใช้งาน แต่การทดสอบคอนแทรคท์ในเครือข่ายสาธารณะของอีเธอเรียม (Public network) หรือเครือข่ายสาธารณะ

สำหรับทดสอบ (Public test network) ที่มีผู้ใช้งานอยู่จริงนั้นมีความซับซ้อนสูงและต้องใช้กำลังในการประมวลผลสูงมาก จึงมีการพัฒนาเครือข่ายแบบจำลองขึ้นมาเพื่อใช้ทดสอบคอนแทรคท์โดยเฉพาะ ซึ่ง Test RPC เป็น Node.js โดยมีพื้นฐานจากอีเธอเรียมที่ช่วยจำลองสภาวะการทำงานของเครือข่ายแบบเต็มรูปแบบและมีคำสั่งเบื้องต้นในการจัดการกับโหนดของอีเธอเรียม เพื่อให้สามารถตั้งค่าสภาวะของ Network ได้ตามต้องการ

2.2.3. Web3 library และ Web3 framework

Web3 API ถูกกำหนดให้เป็น Library หลักในการพัฒนา Dapp เพื่อสร้างปฏิสัมพันธ์กับ Ethereum node โดย Web3 library มีการพัฒนาให้รองรับหลายภาษาและมี Framework ให้เลือกใช้หลากหลายตามจุดประสงค์ของการสร้าง นอกจากนี้การใช้ Framework จะช่วยเพิ่มประสิทธิภาพด้านการทดสอบและความรวดเร็วในการพัฒนาเนื่องจาก Framework ส่วนมากมีคำสั่งพื้นฐานที่สำคัญต่อการสร้าง Dapp เช่น Migration, Compiler, Deploy, Testing เป็นต้น

2.3. งานวิจัยที่เกี่ยวข้อง

2.3.1. MedRec

ในปี ค.ศ.2016 นักวิจัยจากสถาบันเทคโนโลยีแมสซาชูเซตส์ ได้พัฒนาระบบต้นแบบ เพื่อทดสอบความเป็นไปได้และความเหมาะสมในการใช้บล็อกเชนร่วมกับข้อมูลด้านสาธารณสุข อีกทั้งเพื่อศึกษาวิธีการทำงานของ โครงสร้างแบบไม่รวมศูนย์ในบล็อกเชนว่าสามารถเพิ่มประสิทธิภาพด้านความปลอดภัยและใช้งานร่วมกับระบบอื่น ๆ ที่เกี่ยวข้องได้อย่างไร โดยนักวิจัยเลือกได้เลือกใช้อีเธอเรียมบล็อกเชนที่มีจุดเด่นเรื่องสมาร์ทคอนแทรคท์ซึ่งสามารถนำมาพัฒนาแอปพลิเคชันแบบไม่รวมศูนย์ได้ แตกต่างจากบล็อกเชนรูปแบบอื่นที่มีจุดประสงค์หลักเพื่อใช้ในการทำธุรกรรม นอกจากนี้ สมาร์ทคอนแทรคท์ยังมีชุดคำสั่งที่เอื้อต่อการสร้างระบบควบคุมการเข้าถึงข้อมูลระหว่างแหล่งข้อมูลกับส่วนเชื่อมต่อผู้ใช้และสามารถจัดทำแหล่งเก็บข้อมูลแบบไม่รวมศูนย์ที่มีข้อดีในด้านความพร้อมใช้งาน ความสอดคล้องของข้อมูลและป้องกันข้อมูลสูญหายได้อย่างมีประสิทธิภาพ ผลที่ได้จากการพัฒนาข้างต้นคือระบบ MedRec [1] ที่ช่วยให้ข้อมูลการรักษา ข้อมูลการใช้ยา และผลการทดสอบในห้องทดลอง สามารถตรวจสอบและทำงานร่วมกันได้ อีกทั้งยังสามารถกำหนดสิทธิ์การเข้าถึงข้อมูลได้อย่างเหมาะสม นอกจากนี้ผู้ป่วยซึ่งเป็นเจ้าของข้อมูลจะมีสิทธิ์ในการควบคุมการใช้งานข้อมูลอย่างที่เราจะเป็นและสนับสนุนให้นักวิจัยสามารถนำข้อมูลส่วนตัวที่ได้รับอนุญาตจากผู้ป่วยไปใช้ได้โดยไม่เกิดปัญหาทางจริยธรรมในการละเมิดความเป็นส่วนตัว

2.3.2. Estonia e-health และ e-prescription

สาธารณรัฐเอสโตเนียเป็นประเทศรัฐบาลที่ให้ความสำคัญกับระบบสารสนเทศอย่างมาก มีการก่อตั้งมูลนิธิ e-Health foundation [2] ที่เป็นส่วนหนึ่งของ e-government system ตั้งแต่ปี ค.ศ.1997 โดยระบบนี้ช่วยให้ประชาชนสามารถเข้าถึงทุกบริการและทุกหน่วยงานของภาครัฐ และมีการบังคับใช้รูปแบบการจัดการข้อมูลของหน่วยงานต่างภายใต้การควบคุม ให้ความสำคัญ

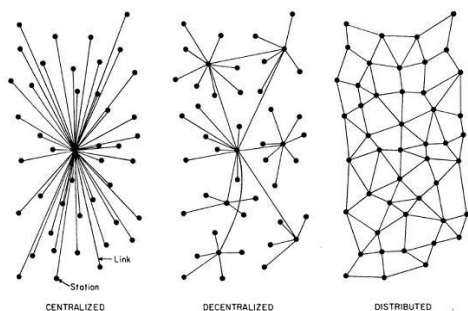
ปลอดภัย สอดคล้องกัน และสามารถนำไปใช้งานร่วมกันได้ ในด้านสาธารณสุขก็เช่นกัน ประชาชนสามารถเข้าถึงประวัติการรักษา ข้อมูลการใช้จ่ายของตนเองที่เคยมีการบันทึกไว้จากสถานพยาบาลต่าง ๆ ได้ด้วย บัตรประชาชนที่มีการฝังชิปเอาไว้ นอกจากนี้แพทย์ผู้รักษาจากสถานพยาบาลต่าง ๆ รวมถึงนักวิจัยที่ต้องการข้อมูลสามารถทำการขอสิทธิในการใช้งานได้

หลังจากนั้น ในปี ค.ศ. 2016 มูลนิธิ e-health มีการประกาศความร่วมมือกับบริษัทด้านความปลอดภัยในการปรับปรุงรูปแบบการเก็บข้อมูลโดยใช้ Keyless Signature Infrastructure (KSI) Blockchain เพื่อป้องกันการลักลอบแก้ไขข้อมูลการรักษาอย่างผิดกฎหมาย และสามารถตรวจสอบเส้นทางในการทุจริตได้

3. การเปรียบเทียบ

3.1. การเปรียบเทียบรูปแบบสถาปัตยกรรมแบบรวมศูนย์ไม่รวมศูนย์ และกระจายศูนย์

ผู้คนส่วนมากมักคุ้นเคยกับคำว่า “แอปพลิเคชัน” ซึ่งแต่ละแอปพลิเคชันมีจุดประสงค์ที่ต่างกันออกไป ปัจจุบันมีมากกว่าหนึ่งพันล้านแอปพลิเคชันที่ใช้อยู่ในปัจจุบันและส่วนใหญ่มีอยู่ในรูปแบบของเว็บแอปพลิเคชันที่เป็นสถาปัตยกรรม client-server แบบรวมศูนย์หรือกระจายศูนย์บางส่วน และส่วนน้อยที่เป็นแบบไม่รวมศูนย์ ตามรูปที่ 1 แสดงภาพรวมของ 3 แบบสถาปัตยกรรมสำหรับแอปพลิเคชัน



รูปที่ 1. สถาปัตยกรรมแบบรวมศูนย์ ไม่รวมศูนย์และกระจายศูนย์

ซอฟต์แวร์แอปพลิเคชันแบบรวมศูนย์ซึ่งเป็นที่นิยมมากที่สุดในปัจจุบัน จะควบคุมการทำงานของแต่ละหน่วยงานและการไหลของข้อมูลโดยตรงจากศูนย์กลางเดียว ทุกหน่วยงานจะขึ้นอยู่กับอำนาจกลางในการรับและการส่งข้อมูลจากคำสั่งที่ได้รับ การจัดเก็บแบบรวมศูนย์ช่วยให้สามารถรวบรวมทรัพยากรการจัดเก็บข้อมูลได้ไม่ว่าจะอยู่ในระดับไฟล์หรือระดับบล็อก เพื่อให้สามารถใช้ที่เก็บข้อมูลได้อย่างมีประสิทธิภาพเนื่องจากเซิร์ฟเวอร์ทั้งหมดสามารถเข้าถึงพื้นที่เก็บข้อมูลเดียวกันได้ การจัดการและการสำรองข้อมูลทำได้ง่ายกว่าเนื่องจากเป็นศูนย์กลางและถูกมองว่าเป็นพื้นที่จัดเก็บข้อมูลเดียว อย่างไรก็ตาม ฐานข้อมูลแบบรวมศูนย์จะขึ้นอยู่กับ การเชื่อมต่อเครือข่าย การเชื่อมต่ออินเทอร์เน็ตที่ช้าลงจะต้องใช้เวลาในการเข้าถึงฐานข้อมูลมากขึ้น อาจเกิดคอขวด (Bottlenecks) ขึ้นจากการเข้าชมที่สูง ซึ่งหากไม่มีการตั้งค่าป้องกันความผิดพลาดและเกิดเสียหายของฮาร์ดแวร์ขึ้น ข้อมูลทั้งหมดภายในฐานข้อมูลจะสูญหายไป

สถาปัตยกรรมแบบไม่รวมศูนย์เป็นการประมวลผลที่เกิดจากการแบ่งการทำงานให้กับคอมพิวเตอร์ต่าง ๆ ภายในองค์กรหรือเครือข่ายเดียวกัน โดยแต่ละหน่วยงานหรือแต่ละคอมพิวเตอร์สามารถจัดการเครื่องคอมพิวเตอร์ของตนเองและไม่ต้องมีปฏิสัมพันธ์กับหน่วยงานอื่น ประเด็นหลักสำคัญเกี่ยวกับระบบแบบไม่รวมศูนย์คือการไม่มีศูนย์กลางควบคุมที่สำคัญ (central point of control) ซึ่งลักษณะการเก็บข้อมูลเป็นไปตามหัวข้อย่อยที่ 2.1.2.

สำหรับสถาปัตยกรรมแบบกระจายศูนย์ ข้อมูลจะถูกจัดเก็บไว้ในหลาย ๆ เครื่องและจะเชื่อมต่อเข้าด้วยกันผ่านระบบเครือข่าย โดยแต่ละส่วนจะมีระบบจัดการฐานข้อมูลเป็นของตนเองและสามารถที่จะทำงานได้ด้วยตนเองหรือทำงานร่วมกันก็ได้ โดยคอมพิวเตอร์แต่ละเครื่องจะเรียกว่า “โหนด” (Node) แต่ละโหนดจะสามารถสื่อสารกันได้ซึ่งผู้ใช้สามารถเรียกใช้ข้อมูลจากฐานข้อมูลที่มีอยู่จากโหนดใดก็ได้และผู้ใช้ไม่จำเป็นต้องทราบว่ามีข้อมูลที่ต้องการนั้นจัดเก็บอยู่บนเครื่องใด หลักการพื้นฐานของฐานข้อมูลแบบกระจาย ใต้ระบุไว้ว่า “สำหรับผู้ใช้ นั้น ระบบกระจายควรจะให้เขามองเห็นเหมือนกันระบบที่ไม่ได้กระจาย” กล่าวอีกอย่างหนึ่งก็คือ ผู้ใช้ฐานข้อมูลในระบบกระจาย ควรจะสามารถกระทำสิ่งต่าง ๆ ได้เหมือนกับว่าระบบที่ใช้อยู่ นั้นไม่ใช้ระบบกระจาย

ในขณะที่ฐานข้อมูลแบบรวมศูนย์นั้นเก็บข้อมูลไว้ในอุปกรณ์จัดเก็บข้อมูลที่อยู่ในตำแหน่งเดียวซึ่งเชื่อมต่อกับหน่วยประมวลผลเดียว ระบบฐานข้อมูลแบบกระจายศูนย์จะเก็บข้อมูลในอุปกรณ์จัดเก็บข้อมูลที่อยู่ในตำแหน่งที่ต่างกัน ฐานข้อมูลแบบรวมศูนย์จะง่ายต่อการรักษาและปรับปรุงให้ดีขึ้น นอกจากนี้ยังง่ายต่อการรักษาความสมบูรณ์ของข้อมูลและหลีกเลี่ยงความซ้ำซ้อนของข้อมูล แต่ค่าของทั้งหมดที่เข้ามาเพื่อต้องการเข้าถึงข้อมูลจะได้รับการประมวลผลโดยเอนทิตีเดียว เช่น เมนเฟรมเดียว จึงสามารถทำให้เกิดคอขวดได้อย่างง่ายดาย แต่ด้วยฐานข้อมูลแบบกระจายศูนย์ สามารถหลีกเลี่ยงคอขวดนี้ได้เนื่องจากฐานข้อมูลแบบขนานทำให้โหลดสมดุลระหว่างหลายเซิร์ฟเวอร์ แต่การเก็บรักษาข้อมูลให้ทันสมัยอยู่เสมอในระบบฐานข้อมูลแบบกระจายจำเป็นต้องใช้งานเพิ่มจึงทำให้ต้นทุนการบำรุงรักษาและความซับซ้อนเพิ่มขึ้นและต้องใช้ซอฟต์แวร์เพิ่มเติมเพื่อการนี้ด้วย นอกจากนี้การออกแบบฐานข้อมูลสำหรับฐานข้อมูลแบบกระจายมีความซับซ้อนมากกว่าฐานข้อมูลแบบรวมศูนย์

3.2. การเปรียบเทียบเครื่องมือพัฒนา

การพัฒนาแอปพลิเคชันแบบไม่รวมศูนย์โดยใช้สมาร์ทคอนแทรคสามารถนำเฟรมเวิร์คที่มีชุดคำสั่งสำเร็จรูปในการทดสอบ คอมไพล์และเรียกใช้งานมาใช้เพื่อผลต่อการพัฒนาทั้งเรื่องความเร็วและความง่ายต่อการแก้ไขแอปพลิเคชัน นอกจากนี้ยังมีเฟรมเวิร์คหลายรูปแบบให้เรียกใช้ซึ่งมีข้อดีแตกต่างกันไป ดังตารางที่ 1 โดยผู้พัฒนาเลือกใช้ Truffle framework เนื่องด้วยความสะดวกในการแก้ไขและใช้งานคอนแทรค การสนับสนุนทุกระบบปฏิบัติการ ส่วนจัดการแพ็คเกจของ Node.js ส่วนเชื่อมต่อกับฐานข้อมูลแบบไม่รวมศูนย์ เช่น Ethereum Swarm และการที่มีผู้ใช้งานจำนวนมาก ทำให้มีตัวอย่างวิธีการแก้ไขปัญหาและการสนับสนุนจากกลุ่มผู้พัฒนาอย่างต่อเนื่อง

ตารางที่ 1. ตารางเปรียบเทียบเครื่องมือที่ใช้พัฒนา

Topic	Truffle	Embark	Dapple
Operation System	MacOS, Windows, Linux	Only Linux	MacOS, Linux
Package management	npm	npm	Nix
Dashboard	no	yes	no
Decentralize storage	Swarm, IPFS	Swarm, IPFS	IPFS
Automate testing	Mocha, Chai	no	DS-test

4. การวิเคราะห์ความต้องการจากกรณีศึกษา

ในการวิจัยและพัฒนาการศึกษาถึงบริบทของโรงพยาบาลเพื่อหาแนวทางการออกแบบและพัฒนาสถาปัตยกรรมแบบกระจายศูนย์เพื่อจัดการควบคุมการเข้าถึงข้อมูลสุขภาพสำหรับเครือข่ายโรงพยาบาลโดยใช้บล็อกเชนส่วนบุคคล ซึ่งได้ทำความร่วมมือกับเครือข่ายโรงพยาบาลเอกชนแห่งหนึ่งในประเทศไทยเพื่อรวบรวมข้อมูลความต้องการของระบบ โดยโรงพยาบาลมีความตั้งใจที่จะพัฒนาระบบเพื่อที่จะแลกเปลี่ยนข้อมูลสุขภาพของผู้ป่วยระหว่างโรงพยาบาลในเครือและหน่วยงานที่เกี่ยวข้อง ในปัจจุบันการที่คนไข้จากโรงพยาบาลหนึ่ง ต้องเปลี่ยนไปรักษาที่โรงพยาบาลในเครืออีกแห่งหนึ่ง จะต้องนำเอกสารจากโรงพยาบาลแรกไปยังโรงพยาบาลที่สองด้วยตนเอง จึงค่อนข้างลำบากและล่าช้าเนื่องจากยังไม่มีระบบมารองรับ และจากเดิมที่ทางโรงพยาบาลมีระบบฐานข้อมูลแบบรวมศูนย์ ปริมาณข้อมูลที่เพิ่มขึ้นในทุกวันจึงเป็นเรื่องยากในการดูแลและขยายระบบเพราะค่าใช้จ่ายด้านโครงสร้าง (Infrastructure) จะสูงขึ้นตามไปด้วย

4.1. ขอบเขตของข้อมูลทดสอบเบื้องต้น

```

296289  },
296290  {
296291    "id": "3afd589c1ff624828444c0b52d5a50b9",
296292    "Employee": "",
296293    "E": "c47e1a8a04ac701524e57f46bfd76ad4",
296294    "Year": 2017,
296295    "Date": "2/28/2017",
296296    "Nation": "THAI",
296297    "Sex": "Male",
296298    "Age": 53,
296299    "Location": "HPC Site 1",
296300    "Weight": 76.9,
296301    "Height": 169,
296302    "BMI": 26.925,
296303    "Systolic": 143,
296304    "Diastolic": 93,
296305    "Glucose (Fasting)": 140,
296306    "Glucose (POCT)": 0,
296307    "HbA1C": 6.3,
296308    "LDL": 171,
296309    "HDL": 59,
296310    "Cholesterol": 228,
296311    "Triglycerides": 151,
296312    "Hypertension": 0,
296313    "Diabetes": 0,
296314    "Vascular Disease": 0,
296315    "Smoking": "",
296316    "DOB": "6/12/1964",
296317    "Waist circumference": 0,
296318    "Waist size (cm)": 95,
296319    "ICD10 Hypertension": 0,
296320    "ICD10 Diabetes": 1
296321  },
296322  {
296323    "id": "57ab30b4deea3b4997123b01f4a228a1",
296324    "Employee": "Not Relevant",

```

รูปที่ 2. ตัวอย่างข้อมูลทดสอบเบื้องต้น

จากการประชุมเพื่อรวบรวมข้อมูลความต้องการของระบบกับทางเครือข่ายโรงพยาบาล โดยขอบเขตของระบบคือ ระบบที่สามารถทำการเก็บข้อมูลผู้ป่วยและข้อมูลธุรกรรมต่าง ๆ ไว้ในบล็อกเชน สามารถกำหนดสิทธิ์การเข้าถึงข้อมูลและทำการโอนย้ายข้อมูลผู้ป่วยระหว่างโรงพยาบาลในเครือ โดยข้อมูลดังกล่าวจะต้องได้รับการยินยอมสิทธิ์จากผู้ป่วยก่อนเสมอ เพื่อพิสูจน์แนวคิดเบื้องต้น ผู้พัฒนาได้ตกลงกับทางโรงพยาบาลในการใช้ข้อมูลการตรวจสุขภาพของผู้ป่วย (check-up data) เป็นการทดลองเบื้องต้นในการทดสอบการจัดเก็บและแลกเปลี่ยนข้อมูล ซึ่งมีรูปแบบการจัดเก็บข้อมูลดังรูปที่ 2 ซึ่งข้อมูลที่ใช้ในการศึกษาเหล่านี้จะไม่สามารถระบุชื่อผู้ป่วยได้ (Unidentified)

4.2. การกำหนดสิทธิ์การเข้าถึงข้อมูล

จากการวิเคราะห์ความต้องการของระบบพบว่าข้อมูลที่นำมาทดสอบเบื้องต้นแต่ละส่วนจะถูกควบคุมการกำหนดสิทธิ์เข้าถึงข้อมูลโดยแต่ละบุคคลจะมีสิทธิ์เข้าถึงข้อมูลต่างกันไปดังตารางที่ 2

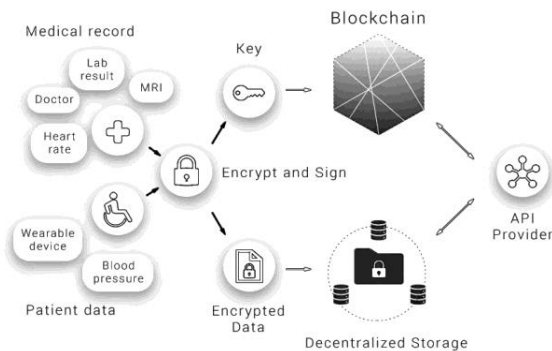
ตารางที่ 2. ตารางกำหนดสิทธิ์การเข้าถึงข้อมูล

Data	Role					
	Physician	Nurse	Pharmacist	Technician	Lab staff	Medical Record staff
Lab result	CRUD	- R - -	- - - -	- R - -	- R U -	- R - -
Prescription	CRUD	- R - -	- R U -	- - - -	- - - -	- R - -
X-ray	CRUD	- R - -	- - - -	- R U -	- - - -	- R - -
Patient record	CRUD	CRUD	CRUD	CRUD	CRUD	- R - -

กรณีศึกษานี้จะแบ่งผู้เกี่ยวข้องออกเป็น 6 บทบาท ได้แก่ แพทย์ (Physician) พยาบาล (Nurse) เภสัชกร (Pharmacist) นักเทคนิคการแพทย์ (Technician) เจ้าหน้าที่ตรวจผลการรักษาเชิงปฏิบัติ (Lab staff) และเจ้าหน้าที่ตรวจผลการรักษา (Medical Record staff) แบ่งประเภทของข้อมูลออกเป็น 4 ประเภท ได้แก่ ผลการรักษาเชิงปฏิบัติ (Lab result) ใบสั่งยา (Prescription) ผลภาพฉายรังสี (X-ray) และข้อมูลผู้ป่วย (Patient Record) แต่ละบทบาทสามารถเข้าถึงข้อมูลแต่ละประเภทได้แตกต่างกันออกไปตามสิทธิ์ของตนที่ถูกกำหนดไว้ตามตัวอักษร C R U และ D ซึ่งตัวอักษรเหล่านี้หมายถึง สิทธิ์ในการสร้างข้อมูล สิทธิ์ในการอ่านข้อมูล สิทธิ์ในการแก้ไขข้อมูล และสิทธิ์ในการลบหรือยกเลิกข้อมูลตามลำดับ เช่น พยาบาลสามารถอ่านข้อมูลผลการรักษาเชิงปฏิบัติ ใบสั่งยา ผลภาพฉายรังสี และสามารถสร้าง อ่าน แก้ไข และลบข้อมูลผู้ป่วยได้

5. ผลการออกแบบสถาปัตยกรรมของระบบ

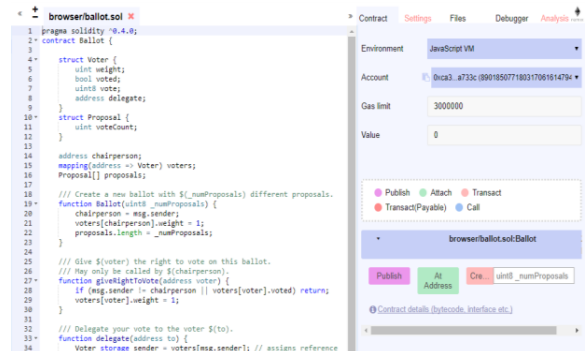
จากการวิเคราะห์เทคโนโลยีบล็อกเชนและความต้องการของเครือข่ายโรงพยาบาลเป็นกรณีศึกษา ผู้พัฒนาได้สร้างแผนภาพแสดงรูปแบบสถาปัตยกรรมของระบบดังรูปที่ 3



รูปที่ 3. ผลการออกแบบสถาปัตยกรรมของระบบ

โดยระบบนี้จะทำการรับข้อมูลผู้ป่วยจากแหล่งต่าง ๆ ไม่ว่าจะเป็น อุปกรณ์สวมใส่ด้านสุขภาพหรือผลตรวจจากทางโรงพยาบาล มาทำการเข้ารหัสและจัดเก็บแบบไม่รวมศูนย์ในเครือข่ายเฉพาะ (Private network) ที่เป็นการนำอีเธอเรียมบล็อกเชน มาใช้เนื่องจากเป็นบล็อกเชนแพลตฟอร์มที่มุ่งเน้นให้นำบล็อกเชนมาใช้กับอุตสาหกรรมอื่นนอกเหนือจากการเงิน เช่น ธุรกิจอาหารมาร์ทรี ธุรกิจประกันภัย รวมถึงด้านการแพทย์ โดยมีสมาร์ทคอนแทรคท์ และ EVM เป็นหัวใจสำคัญซึ่งผู้พัฒนาต้องทำการปรับแต่งข้อกำหนดและการตั้งค่าต่าง ๆ ในไฟล์ตั้งต้น (Genesis file) ให้เข้ากับบริบทขององค์กร จากนั้นนำคีย์ในการเข้าถึงข้อมูลมาผูกกับบัญชีของผู้ป่วยในบล็อกเชน ซึ่งในส่วนนี้สามารถควบคุมสิทธิ์การเข้าถึงและติดตามการใช้งานข้อมูลได้ด้วยสมาร์ทคอนแทรคท์ ซึ่งผู้พัฒนาต้องการออกแบบให้ระบบสามารถเรียกใช้ได้โดยแอปพลิเคชันที่จะถูกใช้งานจากแผนกต่าง ๆ ของโรงพยาบาล และรองรับการเชื่อมต่อกับโรงพยาบาลอื่น ๆ ในอนาคต จึงทำการออกแบบตามแนวคิด REST API [9] เนื่องจากสามารถทำงานบน HTTP โปรโตคอลและสนับสนุนรูปแบบข้อมูลแบบ JSON นอกจากนี้ยังวางตัวเป็นผู้ให้บริการ API สำหรับบล็อกเชนด้านข้อมูลทางการแพทย์ พร้อมกับจัดทำเครือข่ายบล็อกเชนส่วนบุคคล (Private blockchain) ไว้สำหรับรับรองความถูกต้องของการทำการรายการและเอาไว้จัดเก็บข้อมูล โดย API เบื้องต้นที่ได้จัดเตรียมไว้ประกอบไปด้วย การสร้างบัญชีผู้ใช้งาน การกำหนดสิทธิ์การเข้าถึงข้อมูล การเพิ่มข้อมูล การเข้ารหัสข้อมูล การเรียกดูข้อมูล และการแก้ไขข้อกำหนดการใช้ข้อมูล ทั้งนี้การพัฒนาแอปพลิเคชันแบบกระจายศูนย์ต้องใช้เวลาในการศึกษาและทดลองค่อนข้างมาก การใช้ API เหล่านี้จะช่วยระยะเวลาในการพัฒนา เพื่อให้ผู้ที่ต้องการจะนำไปใช้ไม่จำเป็นต้องศึกษารายละเอียดการทำงานของบล็อกเชนหรือจัดทำเครือข่ายบล็อกเชนเอง แต่ยังสามารถใช้ความรู้พื้นฐานในการทำแอปพลิเคชันแล้วปฏิบัติตามแนวทางการใช้งานของแต่ละ API ที่ได้อ้างอิงไว้ และยังคงประโยชน์ของโครงสร้างแบบกระจายศูนย์ไว้ได้ทั้งหมด

โครงสร้างระบบแบบอีเธอเรียมบล็อกเชนออกแบบมาให้ใช้สมาร์ทคอนแทรคท์เป็นตัวกำหนดบทบาทของผู้ใช้ภายในเครือข่ายบล็อกเชนเดียวกันเสมือนเป็นข้อตกลงร่วมกันในเรื่องวิธีการทำงานของแอปพลิเคชัน โดยผู้พัฒนาเลือกใช้ Remix IDE ดังรูปที่ 4 ในการเขียนสมาร์ทคอนแทรคท์ เนื่องจากมี Online real-time compiler และมีการจำลองสภาพแวดล้อมของเครือข่ายบล็อกเชนพร้อมกับ log การทำงานของ contract ให้เห็นถึงวิธีการในการสร้าง Transaction โดย account ในระบบ



รูปที่ 4. หน้าต่างบราวเซอร์ Remix IDE

6. บทสรุป

บทความนี้เป็นกรอบการออกแบบและพัฒนาสถาปัตยกรรมแบบกระจายศูนย์เพื่อจัดการควบคุมการเข้าถึงข้อมูลสุขภาพสำหรับเครือข่ายโรงพยาบาล ซึ่งได้จัดเตรียมบล็อกเชนสำหรับข้อมูลด้านการแพทย์ไว้ให้บริการ (Blockchain as a Service) เพื่อความสะดวกต่อในการที่แต่ละโรงพยาบาลสามารถนำบริการนี้ไปพัฒนาระบบใหม่หรือเชื่อมต่อเข้ากับระบบเดิม โดยได้เลือกใช้เทคโนโลยีที่มีศักยภาพอย่างบล็อกเชนในการพัฒนา เนื่องจากเป็นวิธีที่สามารถสร้างความปลอดภัยที่สูงและสามารถกำหนดสิทธิ์การเข้าถึงได้อย่างอิสระ เหมาะอย่างยิ่งสำหรับข้อมูลด้านการแพทย์

เอกสารอ้างอิง

- [1] A. Ekblaw, A. Azaria, J. D. Halamka, MD, A. Lippman, "A Case Study for Blockchain in Healthcare: "MedRec" prototype for electronic health records and medical research data," pp. 1 - 10, Aug. 2016.
- [2] D. Palmer, "Blockchain Startup to Secure 1 Million e-Health Records in Estonia" [online], Retrieved from <https://www.coindesk.com/blockchain-startup-aims-to-secure-1-million-estonian-health-records>, Mar. 2016.
- [3] M. Swan, "Blockchain," 1st edition, pp. ix – xi, Feb. 2015.
- [4] S. Raval, "Decentralized Application," 1st edition, pp. 3 - 38, Aug. 2016.
- [5] T. R. Peltier, "Information Security Fundamentals," 2nd edition, pp. 29 – 42, 2014.
- [6] V. Butein, "A Next Generation Smart Contract & Decentralized Application Platform" [online], Retrieved from <https://github.com/ethereum/wiki/wiki/White-Paper>, 2017
- [7] V. Morabito, "Business Innovation Through Blockchain," 1st edition, pp. 10 - 13, 2017
- [8] V. Buterin, "Merkling in Ethereum" [online], Retrieved from <https://blog.ethereum.org/2015/11/15/merkle-in-ethereum/>, Nov. 2015
- [9] M. Massé, "REST API Design Rulebook," 1st edition, pp. 5 – 33, 2012