



**477- 403 โครงการระบบสารสนเทศ 2**  
**(Project in Information System II)**

เล่มทะเบียนรถบนบล็อกเชน  
( Car registration on the Blockchain )

**จัดทำโดย**

นางสาวณิชนน พรหมทอง รหัสนักศึกษา 5910513057  
นายวุฒิชัย คาลุนนิม รหัสนักศึกษา 5910513057

**อาจารย์ที่ปรึกษา**

ผู้ช่วยศาสตราจารย์จิราวรรณ ท่าอ่างศรี

ภาคการศึกษาที่ 2 ปีการศึกษา 2562

ภาควิชาบริหารธุรกิจ หลักสูตรระบบสารสนเทศทางธุรกิจ  
คณะวิทยาการจัดการ มหาวิทยาลัยสงขลานครินทร์

## คำนำ

รายงานฉบับนี้ จัดทำขึ้นมาเพื่อเป็นส่วนหนึ่งของรายวิชา โครงการระบบสารสนเทศ 2 (PROJECT IN INFORMATION SYSTEM II) ของนักศึกษาชั้นปีที่ 4 ปีการศึกษาที่ 2/2562 สาขา ระบบสารสนเทศ คณะวิทยาการจัดการ มหาวิทยาลัยสงขลานครินทร์ มีวัตถุประสงค์เพื่อศึกษาหาความรู้ โดยรายงานฉบับนี้มีเนื้อหาเกี่ยวกับเรื่องของ Hyperledger และ Blockchain ซึ่งประกอบไปด้วยเรื่องของการเตรียมเครื่องมือให้พร้อมกับการสร้าง Blockchain, ขั้นตอนการสร้าง Fabric Network, การทำงานของ Hyperledger Fabric และบล็อกเชน รวมถึงการใช้งานอุปกรณ์เครื่องมือตัวอื่น ๆ อย่างเช่น Hyperledger composer , node js , docker เป็นต้น

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่า รายงานฉบับนี้จะเป็นประโยชน์สำหรับผู้อ่านและผู้ที่สนใจ ทั้งยังสามารถนำความรู้ที่ได้ไปพัฒนาต่อในการทำงาน หรือการเรียนรู้ในการเรียนการสอนได้อีกด้วย หากรายงานฉบับนี้มีความผิดพลาดประการใด ต้องขออภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

20 กุมภาพันธ์ 2563

## สารบัญ

| เรื่อง                                                         | หน้า |
|----------------------------------------------------------------|------|
| คำนำ                                                           | ก    |
| สารบัญ                                                         | ๖    |
| บทคัดย่อ                                                       | 1    |
| บทที่ 1 บทนำ                                                   |      |
| 1. ที่มาและความสำคัญของโครงการ                                 | 2    |
| 2. วัตถุประสงค์ของการจัดทำโครงการ                              | 2    |
| 3. ประโยชน์ที่คาดว่าจะได้รับจากการทำโครงการ                    | 2    |
| 4. ขอบเขตของโครงการ                                            | 2    |
| 5. ระยะเวลาการทำโครงการ                                        | 2    |
| 6. สถานที่ดำเนินโครงการ                                        | 2    |
| 7. แผนการดำเนินงานตลอดโครงการ                                  | 3    |
| บทที่ 2 แนวคิด ทฤษฎีที่เกี่ยวข้องและเครื่องมือที่ใช้ในการพัฒนา |      |
| 1. Blockchain                                                  |      |
| 1.1. Blockchain คืออะไร                                        | 4    |
| 1.2. หลักการทำงานของบล็อกเชน                                   | 4    |
| 2. Hyperledger                                                 |      |
| 2.1. มารู้จักกับ Hyperledger                                   | 6    |
| 2.2. Hyperledger คืออะไร                                       | 7    |
| 2.3. เป้าหมายของ Hyperledger                                   | 7    |
| 2.4. Distributed Ledger คืออะไร                                | 7    |
| 2.5. Hyperledger Ecosystem ประกอบด้วย                          | 7    |
| 2.5.1. Hyperledger Fabric                                      | 8    |
| 2.5.2. Hyperledger Iroha                                       | 8    |
| 2.5.3. Hyperledger Sawtooth                                    | 8    |
| 2.5.4. Hyperledger Burrow                                      | 8    |
| 2.5.5. Hyperledger Indy                                        | 9    |
| 2.5.6. Hyperledger Cello                                       | 9    |
| 2.5.7. Hyperledger Explorer                                    | 9    |
| 2.5.8. Hyperledger Composer                                    | 9    |

## สารบัญ

| เรื่อง                                                                | หน้า |
|-----------------------------------------------------------------------|------|
| 3. Hyperledger Fabric                                                 |      |
| 3.1. Hyperledger Fabric คืออะไร                                       | 9    |
| 3.2. จุดเริ่มต้นของ Hyperledger Fabric                                | 10   |
| 3.3. สถาปัตยกรรม Hyperledger Fabric                                   | 10   |
| 4. การเตรียมเครื่องมือให้พร้อมกับการสร้าง Blockchain ด้วย Hyperledger |      |
| 4.1. NodeJS และ NPM                                                   | 11   |
| 4.1.1. NodeJS                                                         | 11   |
| 4.1.2. NPM                                                            | 11   |
| 4.2. ภาษา Golang                                                      |      |
| 4.2.1. มารู้จักกับภาษา Go                                             | 12   |
| 4.2.2. ขั้นตอนการเตรียมการ                                            | 12   |
| 4.2.3. เรียนรู้ภาษา Go ด้วยการสั่ง hello, world!                      | 12   |
| 4.2.4. ข้อดีของภาษา Go                                                | 12   |
| 4.3. Docker และ Docker Compose                                        |      |
| 4.3.1. Docker คืออะไร                                                 | 12   |
| - Docker image คืออะไร                                                | 13   |
| - Docker container คืออะไร                                            | 14   |
| - Docker Toolbox คืออะไร                                              | 14   |
| - ความน่าสนใจของ Docker                                               | 14   |
| 4.3.2. Docker Compose คืออะไร                                         | 15   |
| 5. ขั้นตอนการสร้าง Fabric Network                                     |      |
| 5.1. การติดตั้ง Docker Images ของ Fabric                              | 16   |
| 5.2. Build your first network                                         | 16   |
| 5.3. ส่วนประกอบของ Hyperledger Network เบื้องต้น                      | 21   |
| 6. Blockchain for Hyperledger Fabric v1.0                             |      |
| 6.1. บริการ IBM Blockchain for Hyperledger Fabric v1.0                | 21   |
| 7. Hyperledger Fabric 1.10                                            | 22   |
| 8. Hyperledger Fabric จัดการกับ Concurrency อย่างไร                   |      |
| 8.1. ลำดับการทำงานของ Fabric ในการอ่านและเขียนข้อมูล                  | 22   |
| 8.2. การทำงานแบบ Stateful ใน Hyperledger Fabric                       | 22   |

## สารบัญ

| เรื่อง                      | หน้า |
|-----------------------------|------|
| 8.3. R/W Set บน MVCC        | 22   |
| บทที่ 3 วิเคราะห์ระบบ       |      |
| 3.1. Use Case               | 25   |
| 3.2. Work Flow Diagram      | 26   |
| 3.3. Data Flow Diagram      | 27   |
| 3.4. Process Description    | 28   |
| บทที่ 4 ผลการดำเนินงาน      | 30   |
| บทที่ 5 สรุปผลและข้อเสนอแนะ |      |
| 5.1. ผลการดำเนินงาน         | 36   |
| 5.2. ปัญหาและอุปสรรค        | 36   |
| 5.3. ข้อเสนอแนะและการพัฒนา  | 36   |
| บรรณานุกรม                  | 37   |

## สารบัญภาพ

| เรื่อง                                                                         | หน้า |
|--------------------------------------------------------------------------------|------|
| รูปที่ 1 แสดง Infographic บล็อกเชนของเครือข่ายเงินดิจิทัล                      | 4    |
| รูปที่ 2 การแสดงข้อมูลของผู้อื่นเป็นสาธารณะ                                    | 5    |
| รูปที่ 3 การแสดงการแปลงข้อมูลโดยใช้ฟังก์ชันแฮชในบล็อกเชน                       | 5    |
| รูปที่ 4 แสดงการเปลี่ยนแปลงของ HASH เมื่อมีการเปลี่ยนแปลงข้อมูลที่ใกล้เคียงกัน | 6    |
| รูปที่ 5 Hyperledger Goals                                                     | 7    |
| รูปที่ 6 Hyperledger Modular Umbrella Approach                                 | 8    |
| รูปที่ 7 การทำงานของ cello                                                     | 9    |
| รูปที่ 8 แสดง Comparing Containers and Virtual Machines                        | 13   |
| รูปที่ 9 แผนภาพการจำลองการทำงานของ docker บน windows                           | 14   |
| รูปที่ 10 แสดง docker-compose.yml                                              | 15   |
| รูปที่ 11 Docker images                                                        | 16   |
| รูปที่ 12 เช็ค Directory                                                       | 16   |
| รูปที่ 13 เช็คไฟล์ใน Folder bin                                                | 17   |
| รูปที่ 14 แสดงการเรียกไฟล์จากทุก Path                                          | 17   |
| รูปที่ 15 แสดงการเรียกคำสั่ง cryptogen                                         | 18   |
| รูปที่ 16 การ clone repository มาไว้ที่ Workspace                              | 18   |
| รูปที่ 17 แสดงการเข้าไปใน first-network                                        | 19   |
| รูปที่ 18 แสดงการเข้าไฟล์ cryptogen และ configtxgen                            | 19   |
| รูปที่ 19 แสดง Spec ของ Orderer node และ Peer ต่าง ๆ ใน Organization           | 20   |

**ชื่อโครงการ:** เล่มทะเบียนรถบนบล็อกเชน ( Car registration on the Blockchain )

**อาจารย์ที่ปรึกษา:** ผู้ช่วยศาสตราจารย์จิราวรรณ ลำอาจศรี

**ผู้จัดทำ:** 1.นางสาวณิชนน พรหมทอง รหัสนักศึกษา 5910513057

2.นายวุฒิชัย ดาลุนนิม รหัสนักศึกษา 5910513075

### บทคัดย่อ

การจัดทำรายงานในครั้งนี้มีวัตถุประสงค์เพื่อ (1) สร้างเครือข่ายของ Fabric Network บน Blockchain ให้สามารถทำงานเชื่อมโยงกันได้ (2) ศึกษาโปรแกรมและภาษาที่ใช้สร้างเครือข่ายบนบล็อกเชน ซึ่งในที่นี้จะใช้ภาษา javascript เป็นหลัก และต้องติดตั้งโปรแกรมอื่น ๆ อย่างเช่น Docker , Ubuntu , npm และเครื่องมืออื่น ๆ (3) ศึกษาวิธีการดำเนินงานโครงการและแก้ปัญหาเมื่อเกิดข้อผิดพลาดได้

ผลการศึกษาและจัดทำโครงการพบว่า เล่มทะเบียนรถบนบล็อกเชน สามารถทำการแก้ไขข้อมูลส่วนตัวของผู้ใช้ และเจ้าหน้าที่สามารถดูข้อมูลของผู้ใช้และข้อมูลรายละเอียดได้ รวมถึงเจ้าหน้าที่จะเป็นผู้ทำการโอนรถและแก้ไขข้อมูลรถได้เพียงผู้เดียว ทั้งนี้ก่อนทำการเข้าสู่ระบบผู้ใช้จะต้องล็อกอินก่อนเป็นอันดับแรก การจัดทำโครงการครั้งนี้เพื่อตอบสนองการป้องกันการสวมรอยทะเบียนรถ ป้องกันสมุดคู่มือจดทะเบียนรถสูญหายหรือชำรุด และเพื่อเพิ่มความสะดวกในการใช้งานให้กับผู้ใช้และเจ้าหน้าที่กรมขนส่ง

## บทที่ 1 บทนำ

### 1. ที่มาและความสำคัญของการโครงการ

ปัจจุบันเทคโนโลยีบล็อกเชนได้เข้ามามีอิทธิพลต่อการจัดการข้อมูลมากขึ้น ทั้งในภาครัฐและภาคเอกชน เนื่องด้วยบล็อกเชนมีคุณสมบัติที่มีความปลอดภัยในระดับสูง ทั้งยังออกแบบมาให้ทนทานต่อการเปลี่ยนแปลงข้อมูลที่บันทึกแล้ว

ในปัจจุบันผู้ที่ถือกรรมสิทธิ์จะต้องเก็บสมุดคู่มือจดทะเบียนรถไว้ เพื่อยืนยันตัวตนกับรถหรือป้องกันรถที่ถูกขโมยมา ดังนั้นเมื่อสมุดคู่มือจดทะเบียนรถสูญหาย หรือเกิดการชำรุด ก็จะทำให้เกิดปัญหาและสร้างความยุ่งยากให้กับผู้ใช้ในการไปติดต่อทำสมุดคู่มือจดทะเบียนรถเล่มใหม่ ทั้งยังต้องเสียค่าธรรมเนียมอีกด้วย จึงทำให้มีการนำเอาเทคโนโลยีบล็อกเชนเข้ามาช่วยในการจัดการปัญหา โดยการบันทึกข้อมูลต่าง ๆ ลงในบล็อกเชนแทนการถือสมุดคู่มือจดทะเบียนรถ รวมถึงบันทึกประวัติการเปลี่ยนแปลงผู้ถือกรรมสิทธิ์รถเป็นปัจจุบัน ซึ่งผู้ใช้ก็สามารถดูสมุดคู่มือจดทะเบียนรถแบบออนไลน์ได้ ช่วยให้สะดวกและไม่ต้องกลัวสมุดเล่มทะเบียนรถจะสูญหายอีกด้วย

### 2. วัตถุประสงค์ของการจัดทำโครงการ

1. เพื่อป้องกันการสวมรอยทะเบียนรถ
2. เพื่อป้องกันสมุดคู่มือจดทะเบียนรถสูญหายหรือชำรุด
3. เพื่อเพิ่มความสะดวกในการใช้งาน

### 3. ประโยชน์ที่คาดว่าจะได้รับจากโครงการ

1. ป้องกันการสวมรอยทะเบียนรถได้
2. ป้องกันสมุดคู่มือจดทะเบียนรถสูญหายหรือชำรุด
3. เพิ่มความสะดวกในการใช้งาน

### 4. ขอบเขตของโครงการ

1. บันทึกข้อมูลรายละเอียดรถ
2. บันทึกประวัติผู้ถือกรรมสิทธิ์รถ
3. สามารถแก้ไขผู้ถือกรรมสิทธิ์รถได้

### 5. ระยะเวลาการทำโครงการ

ระยะเวลาที่ใช้ในการทำโครงการเริ่ม 15 สิงหาคม 2562 ถึง 30 เมษายน 2562

### 6. สถานที่ดำเนินโครงการ

มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ คณะวิทยาการจัดการ ตึก Fms





## บทที่ 2 แนวคิด ทฤษฎีที่เกี่ยวข้องและเครื่องมือที่ใช้ในการพัฒนา

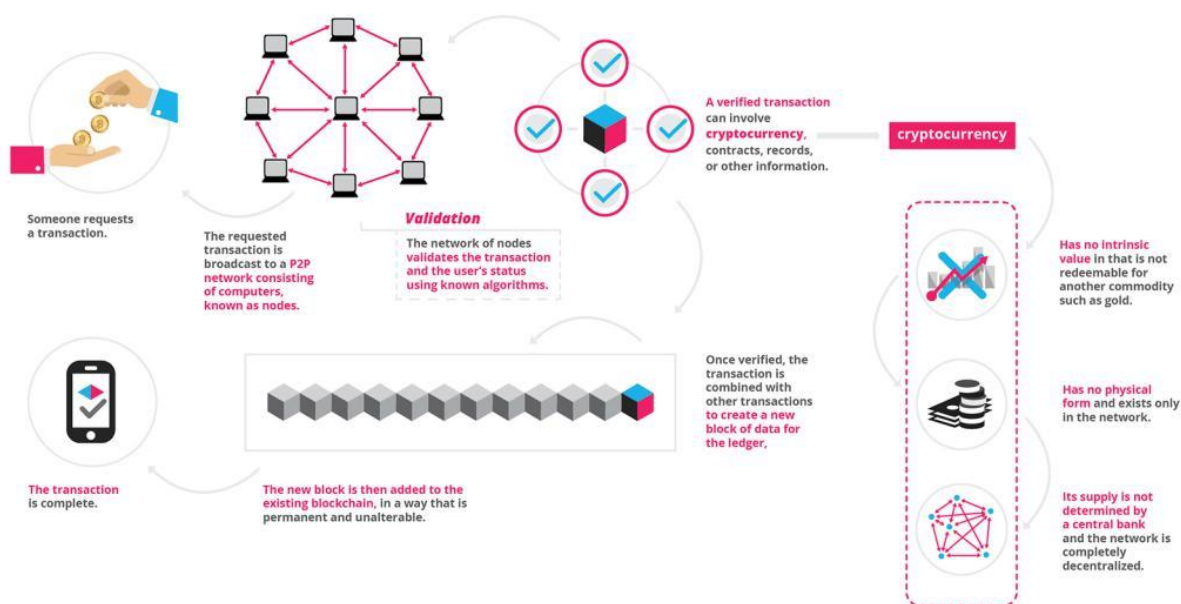
### 1. Blockchain

#### 1.1. Blockchain คืออะไร

บล็อกเชน คือ บัญชีดิจิทัลแยกประเภทที่บันทึกธุรกรรมที่ไม่อาจซ่อนโง่ได้ และไม่ได้บันทึกเพียงแค่ธุรกรรมทางการเงินเท่านั้น แต่ยังบันทึกได้ทุกธุรกรรมที่มีมูลค่า ซึ่งเป็นนวัตกรรมที่ฉลาดมากที่รวบรวมเอา 3 เทคโนโลยีที่มีมานานแล้วมาใช้งานร่วมกันอย่างแบบยล ดังนี้

- 1) Cryptographic keys คือการเข้ารหัสลับ
- 2) ระบบ peer to peer network หมายถึง เครือข่ายที่ไม่มี Server กลางใช้ในการบันทึก และเผยแพร่บัญชีการทำธุรกรรม
- 3) โปรแกรมในการให้แรงจูงใจกับเครือข่ายในการให้บริการเก็บรักษาข้อมูลและความปลอดภัย

#### 1.2. การทำงานของบล็อกเชน



รูปที่ 1 แสดง Infographic บล็อกเชนของเครือข่ายเงินดิจิทัล (cr.Blockgeeks)

โดยบล็อกเชน ก่อการณโดยกลุ่มคนหรือบุคคลที่สร้างเหรียญบิตคอยน์ขึ้นมา เรียกคนกลุ่มนี้ว่า นากาโมระ ปรากฏครั้งแรกในปี 2008 โดยใช้เทคโนโลยีบล็อกเชนเป็นเครื่องมือบันทึกธุรกรรมของเงินดิจิทัลสกุลบิตคอยน์ จากความสำเร็จของเหรียญบิตคอยน์ส่งผลให้เทคโนโลยีบล็อกเชนเติบโตขึ้นอย่างรวดเร็วและได้ถูกนำไปใช้ในหลาย ๆ อุตสาหกรรม โดยเฉพาะอุตสาหกรรมเกี่ยวกับการเงินและธนาคาร โดยเทคโนโลยีบล็อกเชนมีจุดเด่นอยู่ 3 อย่าง คือ

### 1) การกระจายอำนาจ

ข้อมูลจะไม่ถูกจัดเก็บโดยหน่วยงานใดหน่วยงานหนึ่งแต่ทุกคนในเครือข่ายกลับเป็นเจ้าของข้อมูลในเครือข่าย หากเราต้องการโต้ตอบกับเพื่อนเราสามารถทำได้โดยตรงโดยไม่ต้องผ่านบุคคลที่สาม นั่นคือคุณสมบัติหลักที่อยู่เบื้องหลัง Bitcoins และเราคนเดียวเท่านั้นที่ เป็นผู้ดูแล และสามารถส่งเงินของเราให้กับทุกคนที่เราต้องการโดยไม่ต้องผ่านธนาคาร

### 2) ความโปร่งใส

ตัวตนที่แท้จริงของบุคคลอื่นจะถูกซ่อนไว้ผ่านการเข้ารหัสที่ซับซ้อนและจะแสดงเพียงที่อยู่สาธารณะ ของพวกเขาเท่านั้น ดังนั้นหากเราต้องการค้นหาประวัติการทำธุรกรรมของบุคคลอื่นเราจะไม่เห็นข้อมูลเป็น “ Bob ส่ง 1 BTC” แต่เราจะเห็นข้อมูลสาธารณะเป็น “1MF1bhsFLkBzzz9vpFYEmvwT2TbyCt7NZJ ส่ง 1 BTC” จะเห็นได้ว่าข้อมูลจะถูกแปลงเป็นตัวอักษรภาษาอังกฤษและเป็นตัวเลขแทน เพื่อป้องกันไม่ให้ผู้อื่นนำข้อมูลไปใช้ในทางที่ไม่ดี

| TxHash               | Block   | Age         | From                | To                   | Value                   | [Tx Fee]   |
|----------------------|---------|-------------|---------------------|----------------------|-------------------------|------------|
| 0x2d055e4585ae2a...  | 5629306 | 16 secs ago | 0x003e3655080890... | 0x2bdc9181de5c1b...  | 0,004741591554641 Ether | 0.000294   |
| 0xb4d37c791ff4cde... | 5629306 | 16 secs ago | 0x6c3b4fa413e0e4... | 0xf14cb3acac7b230... | 0,744767225 Ether       | 0.000294   |
| 0x9979410dcb5f4c...  | 5629306 | 16 secs ago | 0x99bcd75abba05...  | 0x2d42ee86390c5b...  | 0,016294 Ether          | 0.000294   |
| 0x189c4cf4aee09be... | 5629306 | 16 secs ago | 0x175cd602b2a1e7... | 0xd39681bb0586b...   | 0,01 Ether              | 0.000294   |
| 0xda0e9bb11fb77...   | 5629306 | 16 secs ago | 0x73a065367d111c... | 0x01995788f14357...  | 0 Ether                 | 0.00150007 |
| 0x6be498fafad9acb... | 5629306 | 16 secs ago | 0xa3eb206871124a... | 0x8a91cac422e55e...  | 0,029594 Ether          | 0.000294   |

รูปที่2 การแสดงข้อมูลของผู้อื่นเป็นสาธารณะ

### 3) ข้อมูลไม่มีการเปลี่ยนแปลง

ความสามารถไม่เปลี่ยนแปลงในบริบทของบล็อกเชน หมายความว่าเมื่อมีบางสิ่งบางอย่างเข้าสู่ blockchain แล้วจะไม่สามารถแก้ไขได้ โดยคุณสมบัติการเข้ารหัสลับเป็นคลาสแฮชฟังก์ชันพิเศษซึ่งมีคุณสมบัติต่าง ๆ ทำให้เหมาะสำหรับการเข้ารหัส และมีคุณสมบัติบางอย่างที่จำเป็นต้องมีฟังก์ชันแฮชเพื่อที่จะได้รับการพิจารณาว่าปลอดภัย ซึ่งทำงานโดยการรับ input ของข้อความใด ๆ มาแล้วดำเนินการผ่านอัลกอริทึมให้ output ของความยาวนั้นคงที่โดยใช้ SHA-256 (อัลกอริทึม Hashing ที่ปลอดภัย 256)

| INPUT                                         | HASH                                                              |
|-----------------------------------------------|-------------------------------------------------------------------|
| Hi                                            | 3639EFCDD08ABB273B1619E82E78C29A7DF02C1051B1820E99FC395DCAA3326B8 |
| Welcome to blockgeeks. Glad to have you here. | 53A53FC9E2A03F9B6E66D84BA701574CD9CF5F01FB498C41731881BCDC68A7C8  |

รูปที่ 3 การแสดงการแปลงข้อมูลโดยใช้ฟังก์ชันแฮชในบล็อกเชน

อย่างที่เรเห็นในกรณีของ SHA-256 ไม่ว่า input ของเราจะเล็กหรือใหญ่ก็ตาม output จะมีความยาว 256 บิตคงที่เสมอ สิ่งนี้กลายเป็นสิ่งสำคัญเมื่อเราต้องรับมือกับข้อมูลและธุรกรรมจำนวนมาก โดยพื้นฐานแล้วแทนที่จะจดจำข้อมูล input ที่อาจมีขนาดใหญ่มากเราสามารถจดจำแฮชและติดตามได้และหากเราต้องการจะเปลี่ยนแปลงข้อมูลให้มีความผิดเพี้ยนหรือคล้ายคลึงกับข้อความเดิม ก็ไม่สามารถทำได้เพราะแฮชที่ได้จะมีความยาวต่างกันมาลองทดสอบโดยใช้ SHA-256:

| INPUT          | HASH                                                             |
|----------------|------------------------------------------------------------------|
| This is a test | C7BE1ED902FB8DD4D48997C6452F5D7E509FBCDBE2808B16BCF4EDCE4C07D14E |
| this is a test | 2E99758548972A8E8822AD47FA1017FF72F06F3FF6A016851F45C398732BC50C |

รูปที่ 4 แสดงการเปลี่ยนแปลงของHASH เมื่อมีการเปลี่ยนแปลงข้อมูลที่ใกล้เคียงกัน

จะเห็นว่าการปรับแต่งขนาดเล็กนี้เป็นสิ่งที่ทำให้บล็อกเชนน่าเชื่อถือ จึงทำให้มั่นใจได้ว่าข้อมูลที่ได้รับไม่ได้มีการปลอมแปลงมาก่อนหรือหากเราถูกแฮกเกอร์โจมตีบล็อก 3 และพยายามเปลี่ยนข้อมูลเนื่องจากคุณสมบัติของฟังก์ชันแฮชการเปลี่ยนแปลงข้อมูลเล็กน้อยจะเปลี่ยนแฮชอย่างมาก ซึ่งหมายความว่า การเปลี่ยนแปลงเล็กน้อยในบล็อก 3 จะเปลี่ยนแฮชที่เก็บไว้ในบล็อก 2 ตอนนี้จะเปลี่ยนข้อมูลและแฮชของบล็อก 2 ซึ่งจะส่งผลให้มีการเปลี่ยนแปลงในบล็อก 1 เป็นต้น ตรงนี้เป็นวิธีการที่บล็อกเชนแสดงการไม่เปลี่ยนรูปของข้อมูล

## 2. Hyperledger

### 2.1. มารู้จักกับ Hyperledger

Blockchain มีผลกระทบต่อโลกในอนาคตอย่างมากการมีโปรเจก Open sourced Blockchain ซึ่งอยู่ภายใต้การดูแลของ Linux Foundation ที่ต้องการจะทำให้ทุกคนเข้าถึง Blockchain ได้โดยไม่ต้องจ่ายเงินในราคาแพง และสามารถนำมาประยุกต์ใช้กับธุรกิจได้โดยไม่ต้องเสียค่าใช้จ่ายในการสร้าง Blockchain ขึ้นมาเองทั้งหมดเราเรียกมันว่า Hyperledger

Hyperledger เป็นการรวบรวมเอา Platform ที่มีความโดดเด่นและช่วยเหลือวงการ Open sourced Blockchain เข้ามาไว้ด้วยกันเพื่อใช้สนับสนุนด้านเงินทุน การพัฒนาและการตลาด โดย Platform เหล่านี้ถูกพัฒนาโดยบริษัทใหญ่ ๆ มากมายอย่างเช่น IBM, Hitachi และ Intel เป็นต้น

## 2.2. Hyperledger คืออะไร

คือกลุ่มของ Open sourced Framework ที่ใช้ในการสร้าง Blockchain Linux Foundation



รูปที่ 5 Hyperledger Goals

## 2.3. เป้าหมายของ Hyperledger

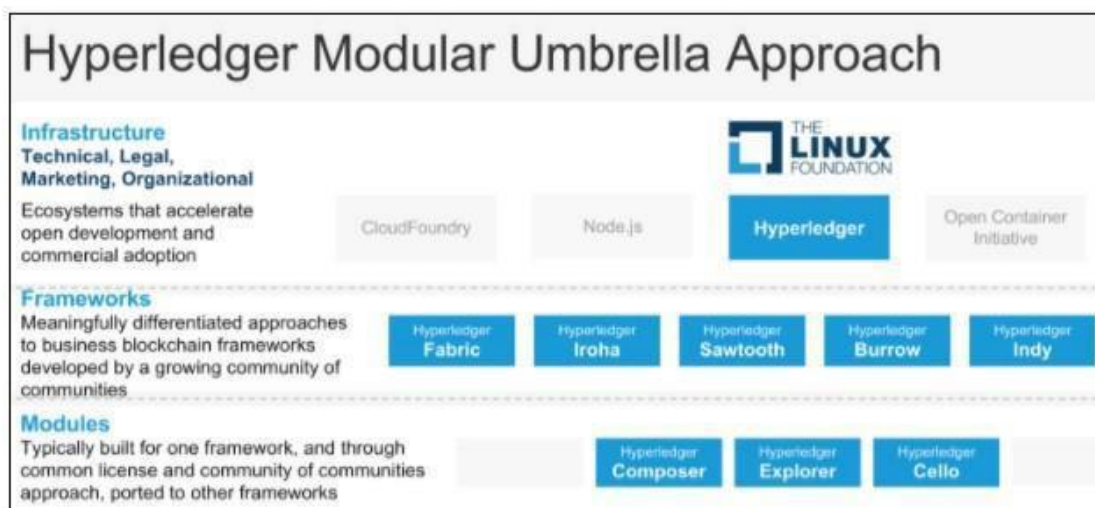
คือ การสร้างแพลตฟอร์ม distributed ledger ที่เปิดและสามารถนำไปใช้ได้หลาย ๆ อุตสาหกรรม จึงเน้นออกแบบให้มีความ modular สูงและเลือกใช้ได้ตามความต้องการที่แตกต่างกัน

## 2.4. Distributed Ledger คืออะไร

Hyperledger มักจะใช้คำว่า Distributed Ledger แทนคำว่า Blockchain ซึ่งเรียกได้ว่าเป็นคำที่ครอบคลุมการประยุกต์ใช้ที่มากกว่า Blockchain ซึ่งเป็นส่วนหนึ่งของ Distributed Ledger และ Platform ที่ใช้ทำ Distributed Ledger ประกอบไปด้วยส่วนประกอบ 3 อย่างคือ Data Model, Language of Transactions และ Protocol

## 2.5. Hyperledger Ecosystem

Hyperledger เป็น โปรเจกต์ที่ประกอบด้วย 5 Platform ในการสร้าง Blockchain และเครื่องมืออีก 4 อย่างในการสนับสนุนการทำงานบน Blockchain ดังนี้



รูปที่ 6 Hyperledger Modular Umbrella Approach

### 1. Hyperledger Fabric

Fabric ถูกออกแบบมาให้เป็น Distributed Ledger ระหว่างองค์กร เพราะแต่ละองค์กรต่างมีความลับบางอย่างที่ไม่สามารถแชร์ให้ผู้ที่ไม่เกี่ยวข้องรับรู้ได้ เช่น บริษัทขายส่งที่มีสัญญาราคาสินค้ากับผู้ซื้อสินค้าแต่ละรายไม่เหมือนกันและผู้ซื้อแต่ละรายอาจไม่จำเป็นต้องทราบปริมาณสินค้าทั้งหมดที่มี ส่วนผู้ตรวจสอบของรัฐบาลอาจต้องการตรวจสอบปริมาณของสินค้าที่มีทั้งหมดแต่ไม่จำเป็นต้องทราบราคา

โดย Fabric จะใช้แนวคิดของ Channel ซึ่งเป็นช่องทางการส่งข้อมูล ทุกคนที่อยู่ใน Channel เดียวกัน จะเห็นข้อมูลและทำงานกับข้อมูลได้เหมือน ๆ กัน แต่คนที่อยู่นอก Channel ถึงแม้จะเข้าถึง Blockchain ได้ แต่ก็ไม่สามารถถอดรหัสข้อมูลได้ นอกจากนี้ Fabric ยังสามารถเปลี่ยนแปลง Consensus และระบบ Login แต่ละองค์กรได้อย่างง่ายดาย

### 2. Hyperledger Iroha

Iroha เป็น Platform ที่ถูกสร้างและใช้งานในภาษา C++ และออกแบบมาให้ใช้งานกับโทรศัพท์มือถือ อย่างเช่น iOS และ Android เป็นหลัก Iroha ใช้ Consensus ที่ชื่อว่า YAC ซึ่งเป็น Voting based consensus algorithm

### 3. Hyperledger Sawtooth

Hyperledger Sawtooth คือ Framework ที่ใช้ในการสร้าง Distributed Ledger โดยมีเป้าหมายในการจัดการ Supply Chain และการทำงานร่วมกันกับ Internet Of Things โดยออกแบบให้เพิ่มผู้มีส่วนได้ส่วนเสียในระบบได้ง่ายและใช้พลังงานและงบประมาณน้อย โดยมี Consensus ของตัวเองที่เรียกว่า Proof of Elapsed Time (PoET)

### 4. Hyperledger Burrow

Burrow มีนิยามว่าเป็น Permissionable Smart Contract Machine ซึ่งเป็นเหมือนซอฟต์แวร์ที่สามารถรัน Smart Contract ของ Ethereum ได้แต่จะรันบน Apache-licensed EVM และเป็นระบบปิด

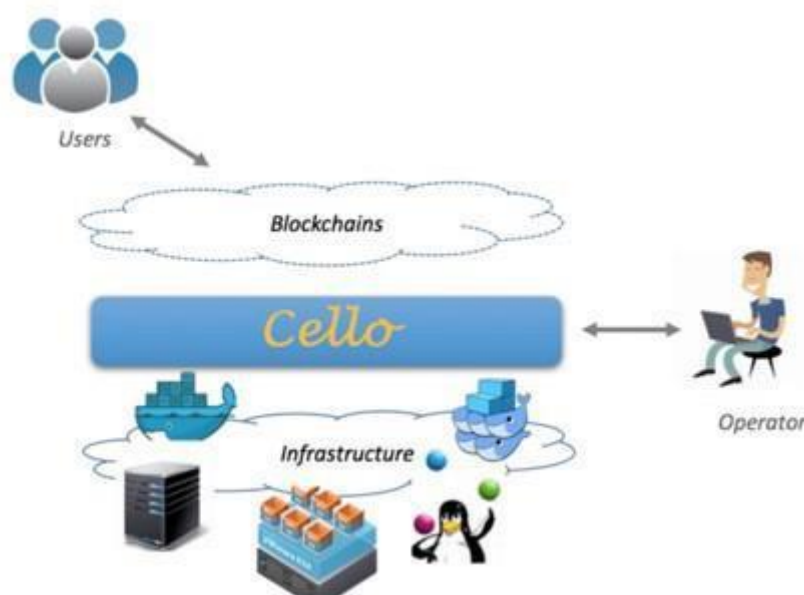
Burrow นั้นสามารถนำไปใช้บน Sawtooth ทำให้ Sawtooth สามารถรัน Smart Contract ของ Ethereum ได้อีกด้วย

### 5. Hyperledger Indy

Indy เป็นระบบที่ใช้งานกับการยืนยันตัวตนของบุคคลแบบกระจายศูนย์คล้ายกับระบบ Login by Facebook ซึ่งออกแบบให้นำไปใช้กับระบบอื่นได้โดยง่าย

### 6. Hyperledger Cello

Cello นั้นช่วยให้ผู้พัฒนา Blockchain สามารถสร้าง จัดการและปล่อย Blockchain ขึ้นไปบน Cloud ได้อย่างง่ายดาย



รูปที่ 7 แสดงการทำงานของ cello

7. Hyperledger Explorer เป็นตัวที่ใช้ทำ Visualization ให้กับ Blockchain ต่าง ๆ

### 8. Hyperledger Composer

เป็น Platform ที่ใช้ JavaScript ในการทำ Blockchain โดยทำให้เราสามารถขึ้นโครงสร้าง และ API ให้กับ Blockchain ของเราอย่างรวดเร็ว และสามารถ Deploy Blockchain นั้นขึ้นไปบน Fabric ได้อีกด้วย

## 3. Hyperledger Fabric

### 3.1. Hyperledger Fabric คืออะไร

Hyperledger Fabric เป็น framework บล็อกเชนแบบแยกส่วน ซึ่งทำหน้าที่เป็นพื้นฐานสำหรับการพัฒนาผลิตภัณฑ์โซลูชันและแอปพลิเคชันที่ใช้บล็อกเชน โดยใช้ส่วนประกอบแบบ plug-and-play ที่มีวัตถุประสงค์เพื่อใช้ภายในองค์กรเอกชน

### 3.2. จุดเริ่มต้นของ Hyperledger Fabric

Hyperledger Fabric ริเริ่มโดย Digital Asset และ IBM และปัจจุบันกลายเป็นกิจการร่วมทุนระหว่างอุตสาหกรรม ซึ่งปัจจุบันอยู่ภายใต้การดูแลของ Linux Foundation และโครงการ Hyperledger หลายโครงการ Fabric เป็นโครงการแรกที่เปิดตัวและบรรลุขั้นตอน “ใช้งาน” ในเดือนมีนาคม 2560

เมื่อเครือข่าย blockchain แบบดั้งเดิมไม่สามารถรองรับธุรกรรมส่วนตัวและสัญญาที่เป็นความลับซึ่งเป็นกุญแจสำคัญสำหรับธุรกิจ Hyperledger Fabric จึงได้รับการออกแบบให้เป็นโครงสร้างพื้นฐานที่ปรับขนาดได้และปลอดภัยสำหรับการเสนอโซลูชันบล็อกเชนอุตสาหกรรม

Hyperledger Fabric ถูกกำหนดให้เป็นเครื่องมือ open source สำหรับ blockchain ธุรกิจและดูแลคุณสมบัติที่สำคัญที่สุดสำหรับการประเมินและการใช้ blockchain สำหรับกรณีการใช้งานทางธุรกิจในที่นี้จะกล่าวถึงภายในเครือข่ายอุตสาหกรรมเอกชน การตรวจสอบตัวตนของผู้เข้าร่วมจัดเป็นข้อกำหนดหลักของ Hyperledger Fabric ที่รองรับการเป็นสมาชิกที่ได้รับอนุญาต ซึ่งต้องการให้ผู้เข้าร่วมเครือข่ายทั้งหมดมีข้อมูลการทำธุรกรรมหรือการใช้ข้อมูลภาคธุรกิจจำนวนมาก เช่น การดูแลสุขภาพและการเงิน มีข้อผูกพันข้อบังคับการปกป้องข้อมูลที่ได้รับคำสั่งในการรักษาข้อมูลเกี่ยวกับผู้เข้าร่วมที่หลากหลายและการเข้าถึงข้อมูลต่าง ๆ Fabric รองรับการเป็นสมาชิกตามการอนุญาต

### 3.3. สถาปัตยกรรม Hyperledger Fabric

เป็นสถาปัตยกรรมแบบแยกส่วนของ Hyperledger Fabric โดยแยก workflow การประมวลผลธุรกรรมออกเป็น 3 ขั้นตอน

- 1) Smart Contract หรือสัญญาอัจฉริยะ Hyperledger ใน Fabric เรียกว่า เซนโค้ดที่ประกอบด้วยการประมวลผลเชิงตรรกะแบบกระจาย และข้อตกลงของระบบการสั่งซื้อธุรกรรม การทำงาน แยกส่วน ทำให้มีประโยชน์หลายประการ
- 2) ลดจำนวนระดับความน่าเชื่อถือและการตรวจสอบที่ช่วยให้เครือข่ายและการประมวลผลที่ปราศจากความยุ่งเหยิง
- 3) ความสามารถในการขยายเครือข่ายที่ดีขึ้นและประสิทธิภาพโดยรวมที่ดีขึ้น

นอกจากนี้การสนับสนุน Hyperledger Fabric สำหรับ plug-and-play สำหรับส่วนประกอบต่าง ๆ ช่วยให้นำคุณสมบัติที่มีอยู่กลับมาใช้ใหม่ได้ง่ายและบูรณาการ readymade ของโมดูลต่าง ๆ ตัวอย่างเช่น หากมีฟังก์ชันที่ยืนยันตัวตนของผู้เข้าร่วมอยู่แล้วเครือข่ายระดับองค์กรก็จำเป็นต้องเปิดเผยและนำโมดูลที่มีอยู่นี้กลับมาใช้ใหม่แทนการสร้างฟังก์ชันเดียวกันใหม่

- ผู้ใช้ในเครือข่ายมีบทบาทที่แตกต่าง 3 ประการ ได้แก่

- 1) ผู้สนับสนุน
- 2) ผู้สับเปลี่ยน
- 3) ผู้ให้ความช่วยเหลือ



โดยสรุปข้อเสนอการทำธุรกรรมจะถูกส่งไปยังผู้สนับสนุนตามนโยบายการรับรองที่กำหนดล่วงหน้าเกี่ยวกับจำนวนผู้ลงนามที่ต้องการหลังจากการรับรองที่เพียงพอ โดยผู้ลงนามชุดหรือบล็อกของการทำธุรกรรมจะถูกส่งไปยังผู้ส่ง ผู้ยอมรับตรวจสอบว่ามีการปฏิบัติตามนโยบายการรับรองและไม่มีธุรกรรมที่ขัดแย้งกัน เมื่อทำการตรวจสอบทั้งสองแล้วธุรกรรมจะถูกส่งไปยังบัญชีแยกประเภท

#### 4.การเตรียมเครื่องมือให้พร้อมกับการสร้าง Blockchain ด้วย Hyperledger

จะต้องติดตั้งสิ่งต่อไปนี้บนแพลตฟอร์มที่จะใช้งานหรือพัฒนาบน Hyperledger ก่อน

##### 4.1. NodeJS และ NPM (แนะนำเป็น NodeJS Version 6)

4.1.1. NodeJS คือ JavaScript มีการ Compiled เป็น Byte Code ด้วย V8 Engine ของ Google ปัจจุบัน NodeJS ถูกนำมาทำเป็น Web Server , Mobile Hybrid , IOT , Webkit , TVOS ,OS และอื่น ๆ อีกมาก เรียกได้ว่าเข้าถึงได้หลากหลายเทคโนโลยี

4.1.2. NPM คือ การจัดการภาครัฐแนวใหม่ หมายถึง รูปแบบการบริหารที่นำมาปรับใช้กับภาครัฐ เพื่อลดปัญหาต่าง ๆ โดยเน้นการทำงานที่ยืดหยุ่นเป็นหลัก มีการวัดผลลัพธ์และค่าใช้จ่ายอย่างเป็นรูปธรรม เป้าหมายสุดท้าย คือ ระบบราชการทำงานที่มีประสิทธิภาพ ประสิทธิผลสูง เทียบเท่ามาตรฐานสากล เจ้าหน้าที่ที่มีศักยภาพสูง และมีการบริหารจัดการทรัพยากรของภาครัฐอย่างประหยัดคุ้มค่าเหมาะสม ทำให้ระบบมีความคล่องตัว ยืดหยุ่น ปรับตัวได้ทันต่อการเปลี่ยนแปลงของสังคมและประชาคมโลก ตอบสนองความต้องการและนำบริการที่ดีมีคุณภาพไปสู่ประชาชน

##### 4.2. ภาษา Golang

###### 4.2.1. มารู้จักกับ Go

Golang หรือ Go เป็นภาษาโปรแกรมภาษาหนึ่งถูกสร้างขึ้นโดยทีมของ Google เป็นภาษาใหม่ที่ Open source และมีฟีเจอร์มากมายอย่าง memory safety, garbage collection, structural typing and CSP-style concurrent programming

###### 4.2.2. ขั้นตอนการเตรียมการ

1) ทำการติดตั้ง Go Programming Language ให้เราสามารถเรียกใช้ go เพื่อการ build run test get สิ่งต่าง ๆ มาใช้ได้และเมื่อทำการติดตั้งเสร็จต้องเช็คค่า Environment Variable Path ของ %GOPATH% %GOROOT% ว่าถูกตั้งค่าเรียบร้อยหรือยังหลักจากนั้นลองสั่ง go version ดู ถ้ามีบางอย่างออกมาก็แสดงว่าเรียบร้อย

2) ใช้ Visual Studio Code ในการเขียนเพราะเครื่องมือนี้เหมาะมากมาพร้อมกับ Extension Go ที่จะช่วยให้การเขียน Go เป็นไปอย่างราบรื่น

3) ใช้ Git Bash อันนี้เป็นทางเลือกเพราะเขียนบน Windows และ cmd มันก็ยุ่งยากเกินไป และที่สำคัญ Git Bash เชื่อมกับ VSCode ได้ดีด้วยการ Settings

“terminal.integrated.shell.windows”: “C:\\Program Files\\Git\\bin\\bash.exe”

#### 4.2.3. เรียนรู้ภาษา Go ด้วยการสั่ง hello, world!

1. เริ่มต้นด้วยการสร้างไฟล์ main.go บรรทัดแรกเป็นการตั้งชื่อ package ชื่อว่า main
2. ทำการ import “fmt” ซึ่งเป็น package หนึ่งของ go ในที่นี้ต้องการใช้ Println จาก fmt
3. ทำการประกาศฟังก์ชัน main() โดยภายในสั่ง Println(“hello, world!”)
4. ทำการสั่ง run โปรแกรมด้วยได้ผลลัพธ์คำว่า hello, world!

```
$ go run main.go
hello, world!
```

#### 5. ทำการสั่ง build โปรแกรมและได้ผลลัพธ์เป็นไฟล์ main.exe

```
$ go build main.go$ ./main.exe
hello, world!
```

#### 6. ทำการสั่ง run โปรแกรมผ่าน main.exe ได้ผลลัพธ์คำว่า hello, world! เช่นกัน

#### 4.2.4. ข้อดีของภาษา Go

1. รองรับทุก os (window, linux, mac)
2. ใช้ระยะเวลาศึกษาไม่นาน
3. หา component ต่าง ๆ เพื่อให้สะดวกต่อการพัฒนาได้ง่ายและติดตั้งง่าย
4. ทำงานเร็ว ลดเวลาการ compile ทุกครั้งที่มีการเรียกใช้

#### 4.3. Docker และ Docker Composer

##### 4.3.1. Docker คืออะไร

Docker คือ engine ตัวหนึ่งที่มีการทำงานในลักษณะจำลองสภาพแวดล้อมขึ้นมาบนเครื่อง server เพื่อใช้ในการ run service ที่ต้องการ มีการทำงานคล้ายคลึงกับ Virtual Machine เช่น

VMWare, VirtualBox, XEN, KVM แต่ข้อแตกต่างที่ชัดเจนคือ Virtual Machine ที่รู้จักกันก่อนหน้านั้นเป็นการจำลองทั้ง OS เพื่อใช้งานและหากต้องการใช้งาน service ใด ๆ ต้องทำการติดตั้งเพิ่มเติมบน OS นั้น ๆ แต่สำหรับ docker จะใช้ container ในการจำลองสภาพแวดล้อมขึ้นมาเพื่อใช้งานสำหรับ 1 service ที่ต้องการใช้งานเท่านั้น โดยไม่ต้องมีส่วนของ OS เข้าไปเกี่ยวข้องเหมือน Virtual Machines อื่น ๆ นอกจากนี้ docker ยังใช้งานได้อย่างสะดวกและตอบสนองความต้องการของผู้พัฒนาโปรแกรม (Developer) หรือผู้ดูแลระบบ System (admin)

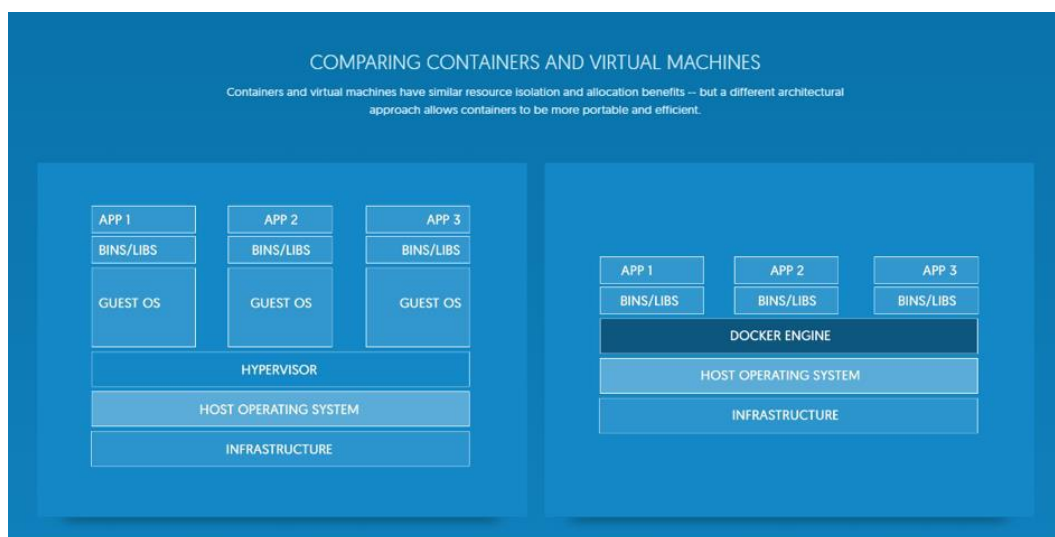
สำหรับการติดตั้ง Docker เวอร์ชัน 17.06.2-ce หรือสูงกว่าเป็นสิ่งจำเป็นและการติดตั้ง Docker บน Mac ,Windows หรือ Docker Toolbox จะติดตั้ง Docker Compose ด้วย หากติดตั้ง Docker ไว้แล้ว ควรตรวจสอบว่า Docker Compose เป็นเวอร์ชัน 1.14.0 หรือสูงกว่าหรือไม่ ถ้าไม่ใช่แนะนำให้ติดตั้ง Docker รุ่นใหม่กว่า

- การตรวจสอบรุ่นของ Docker ที่ติดตั้งด้วยคำสั่งต่อไปนี้จากหน้าจอพร้อมรับเทอร์มินัล

```
docker --version
```

- Docker image คืออะไร

Docker image เป็นเหมือนตัวต้นแบบของ container ซึ่งภายในจะประกอบด้วย application ต่าง ๆ ที่มีการติดตั้งไว้เพื่อใช้งานสำหรับ service นั้น ๆ รวมทั้งมีการ config ค่าต่าง ๆ ไว้เรียบร้อยแล้ว จากนั้นก็นำมาสร้างเป็น docker image บน registry เพื่อนำไปใช้งาน ทั้งนี้ผู้ใช้งานยังสามารถสร้าง docker image สำหรับใช้งานเองได้อีกด้วย



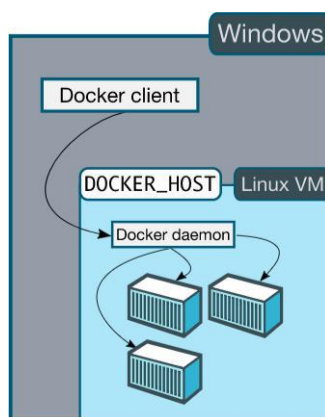
รูปที่ 8 แสดง Comparing Containers and Virtual Machines

- Docker container คืออะไร

Docker container สามารถมองได้เสมือนกล่อง ซึ่งนำ docker image มาติดตั้ง เพื่อให้สามารถใช้งาน service ที่ต้องการจาก image นั้น ๆ ได้ โดยใน container แต่ละตัวจะมีการใช้งาน RAM, CPU, ไฟล์ config ต่าง ๆ เป็นของตัวเองและยังสามารถสั่ง start, stop ได้ที่ container นั้น ๆ อีกด้วย

- Docker Toolbox คืออะไร

เป็นตัวช่วยสำหรับการติดตั้ง Docker ที่เวอร์ชันต่ำกว่าที่รองรับโดยหลักการคร่าวๆของ docker toolbox คือ จะสร้าง Linux VM ขึ้นมาตัวหนึ่งเพื่อใช้รัน docker ตามภาพข้างล่างนี้



รูปที่ 9 แผนภาพการจำลองการทำงานของ docker บน windows

#### • Docker Toolbox ประกอบด้วย

- Docker CLI client ไว้สำหรับรันคำสั่ง Docker Engine (docker) เพื่อสร้าง images และ containers
- Docker Machine (docker-machine) เพื่อสร้าง Linux VM ตัวเล็ก ๆ ที่มี Docker Engine รันอยู่
- Docker Compose สำหรับรันคำสั่ง docker-compose
- Kitematic ที่เป็น Docker GUI
- Docker QuickStart shell ที่มีการทำ preconfigured สำหรับ Docker command-line environment
- Oracle VM Virtual Box

#### • ความน่าสนใจของ docker

- Docker engine สามารถใช้งานได้บนหลาย platform ทั้งบน Linux, Mac และ Windows
- Docker มีขนาดเล็ก สามารถใช้งานและติดตั้งได้อย่างรวดเร็ว และสะดวกในการ start / stop หรือ แม้แต่การย้ายไปใช้งานสำหรับเครื่อง server อื่นที่มีการ run docker engine ก็สามารทำได้โดยไม่ซับซ้อน
- ผู้ใช้งาน docker ไม่จำเป็นต้องติดตั้ง OS อีกครั้งเพื่อติดตั้ง container รวมทั้งไม่จำเป็นต้อง config เพิ่มเติมในส่วนที่ไม่จำเป็นอีกด้วย
- Docker มีความต้องการในการใช้ CPU, RAM และพื้นที่น้อยกว่า Virtual Machine ทั้งนี้ในทรัพยากรที่มีเท่ากัน docker สามารถใช้งาน container ได้มากกว่า Virtual Machine
- เนื่องจากผู้ใช้งานสามารถสร้าง docker image ได้เองจาก dockerfile ดังนั้นการใช้งาน docker ยังช่วยลดปัญหาสภาพแวดล้อมที่ต่างกันมักพบเมื่อบาง application สามารถทำงานได้บน development server แต่ไม่สามารถใช้งานบน production server ได้
- Docker ยังมี docker registry ซึ่งผู้ใช้งานสามารถเลือก pull image ต่าง ๆ ที่มีการสร้างไว้ให้แล้วมาใช้งาน โดยมี Docker Hub เป็น registry หลักในการเรียกใช้ image

### 4.3.2. Docker Compose คืออะไร

Docker Compose คือ tool อีกตัวหนึ่งของ docker ที่จะมาช่วยจัดการการสร้าง container หลาย ๆ ตัวให้เป็นเรื่องที่ยง่าย โดยการสร้างไฟล์ที่ชื่อว่า docker-compose.yml ที่เปรียบเสมือนไฟล์ที่ใช้ในการควบคุมการสร้าง container และการกำหนด option ต่าง ๆ ที่ใช้ในการรัน container แต่ละตัว

```

1  version: '2.2'
2  services:
3    hello-frontend:
4      build:
5        context: hello-frontend/
6      ports:
7        - 8888:80
8      container_name: hello-frontend
9    hello-springboot:
10     build:
11       context: hello-springboot/
12     ports:
13       - 8889:8080
14     container_name: hello-springboot

```

รูปที่ 10 แสดง docker-compose.yml

- version = คือ config กำหนด docker compose เวอร์ชันที่ต้องการใช้งาน
- services = scope ที่ใช้บอกว่าจะมี service อะไรบ้าง
- hello-frontend, hello-springboot = ชื่อ service (1 service สามารถทำได้ตั้งแต่ pull/build image, create/start container และกำหนดค่าต่าง ๆ เช่น port, config, environment variable ของ container ตัวนั้น ๆ)
- build = ระบุว่า จะ build ของจากที่ไหน (ที่ ๆ มี Dockerfile อยู่) สามารถกำหนดเป็น path ที่นี้เลยก็ได้ว่าจะใช้หรือจะใช้
- context = กำหนด path ที่จะไปเรียก Dockerfile หากไม่มี Dockerfile สามารถใช้ config อีกตัวแทน build ได้ นั่นคือ image เป็น config สำหรับระบุ image ที่ต้องการ pull มาใช้งาน
- ports = การ mapping port ระหว่าง ภายนอก:container (ภายนอกเปรียบง่าย ๆ คือ localhost)
- container\_name = กำหนดชื่อให้กับ container

- ตรวจสอบรุ่นของ Docker Compose ที่คุณติดตั้งด้วยคำสั่งต่อไปนี้จากหน้าจอพร้อมเทอร์มินัล

docker-compose --version

## 5. ขั้นตอนการสร้าง Fabric Network

### 5.1. การติดตั้ง Docker Images ของ Fabric

ควรเป็นเวอร์ชัน 1.0.5 หรือสูงกว่าและ Binary Files ที่จำเป็น โดยมีสคริปต์ที่รวบรวมทุกอย่างไว้ให้แล้วโดยการรันคำสั่งที่ home directory

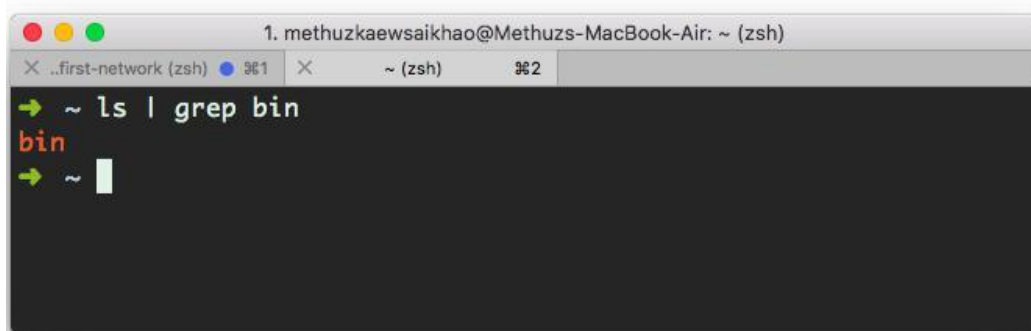
```
cd ~ && curl -SSL https://goo.gl/byy2Qj | bash -s 1.0.5
```

5.1.1. เราจะได้ Docker images ทั้งหมดมา ให้เช็คด้วยคำสั่ง dockerimages เช็คดูว่าทุก Images ได้ติด tag latest แล้วหรือยัง โดยถ้าติดแล้วจะมี imageid เดียวกัน 2 อัน อันหนึ่งเป็น tag:version อีกอันหนึ่งเป็น tag:latest ถ้ายังไม่ติดให้ติดให้หมด

| Image Name                   | Tag          | Image ID     | Age         | Size   |
|------------------------------|--------------|--------------|-------------|--------|
| hyperledger/fabric-tools     | latest       | 6a8993b718c8 | 4 weeks ago | 1.33GB |
| hyperledger/fabric-tools     | x86_64-1.0.5 | 6a8993b718c8 | 4 weeks ago | 1.33GB |
| hyperledger/fabric-couchdb   | latest       | 9a58db2d2723 | 4 weeks ago | 1.5GB  |
| hyperledger/fabric-couchdb   | x86_64-1.0.5 | 9a58db2d2723 | 4 weeks ago | 1.5GB  |
| hyperledger/fabric-kafka     | latest       | b8c5172bb83c | 4 weeks ago | 1.29GB |
| hyperledger/fabric-kafka     | x86_64-1.0.5 | b8c5172bb83c | 4 weeks ago | 1.29GB |
| hyperledger/fabric-zookeeper | latest       | 68945f4613fc | 4 weeks ago | 1.32GB |
| hyperledger/fabric-zookeeper | x86_64-1.0.5 | 68945f4613fc | 4 weeks ago | 1.32GB |
| hyperledger/fabric-orderer   | latest       | 368c78b6f03b | 4 weeks ago | 151MB  |
| hyperledger/fabric-orderer   | x86_64-1.0.5 | 368c78b6f03b | 4 weeks ago | 151MB  |
| hyperledger/fabric-peer      | latest       | c2ab022f0bdb | 4 weeks ago | 154MB  |
| hyperledger/fabric-peer      | x86_64-1.0.5 | c2ab022f0bdb | 4 weeks ago | 154MB  |
| hyperledger/fabric-javaenv   | latest       | 50890cc3f0cd | 4 weeks ago | 1.41GB |
| hyperledger/fabric-javaenv   | x86_64-1.0.5 | 50890cc3f0cd | 4 weeks ago | 1.41GB |
| hyperledger/fabric-ccenv     | latest       | 33feadb8f7a6 | 4 weeks ago | 1.28GB |
| hyperledger/fabric-ccenv     | x86_64-1.0.5 | 33feadb8f7a6 | 4 weeks ago | 1.28GB |
| hyperledger/fabric-ca        | latest       | 002c9089e464 | 4 weeks ago | 238MB  |
| hyperledger/fabric-ca        | x86_64-1.0.5 | 002c9089e464 | 4 weeks ago | 238MB  |

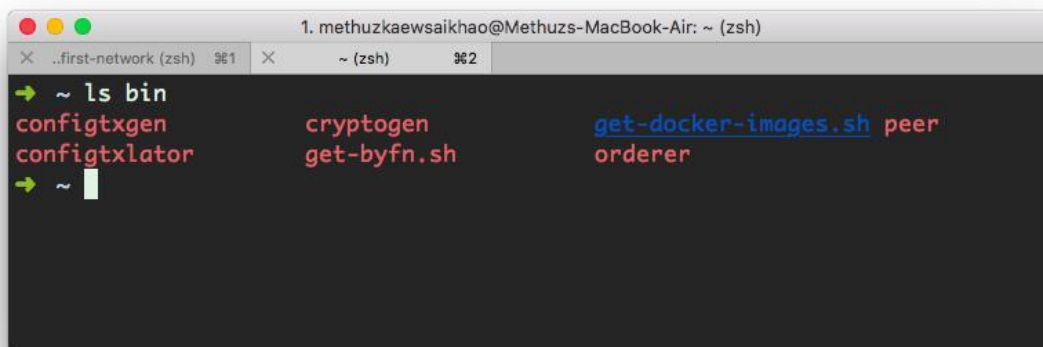
รูปที่ 11 Docker images

### 5.1.2. เช็คที่ Directory ปัจจุบันจะพบว่ามี Folder bin อยู่



รูปที่ 12 เช็ค Directory

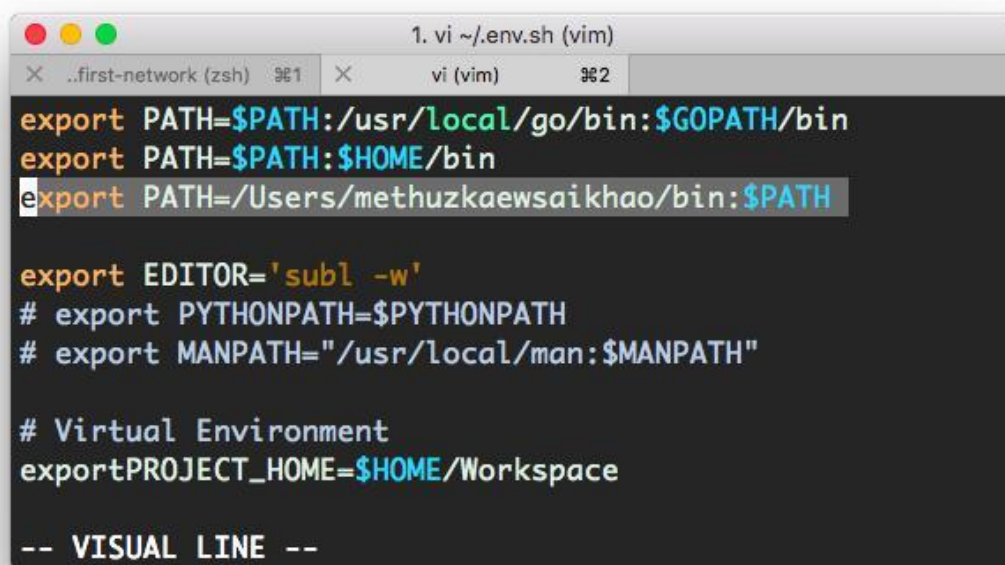
5.1.3. คูใน bin จะมีไฟล์ที่ใช้ทำงานเกี่ยวกับ hyperledger อยู่ดังนี้



```
1. methuzkaewsai khao@Methuzs-MacBook-Air: ~ (zsh)
X ..first-network (zsh) 961 X ~ (zsh) 962
-> ~ ls bin
configtxgen          cryptogen             get-docker-images.sh peer
configtxlator        get-byfn.sh           orderer
-> ~
```

รูปที่ 13 เช็คไฟล์ใน Folder bin

5.1.4. จากนั้นจะทำให้สามารถเรียกไฟล์เหล่านี้จากทุก Path โดยการเพิ่ม Path นี้เข้าไปใน Environment โดยพิมพ์ pwd เพื่อหาชื่อ path แล้วนำไปใส่ไว้ใน ~/.bashrc หรือ ~/.env.sh



```
1. vi ~/.env.sh (vim)
X ..first-network (zsh) 961 X vi (vim) 962
export PATH=$PATH:/usr/local/go/bin:$GOPATH/bin
export PATH=$PATH:$HOME/bin
export PATH=/Users/methuzkaewsai khao/bin:$PATH

export EDITOR='subl -w'
# export PYTHONPATH=$PYTHONPATH
# export MANPATH="/usr/local/man:$MANPATH"

# Virtual Environment
export PROJECT_HOME=$HOME/Workspace

-- VISUAL LINE --
```

รูปที่ 14 แสดงการเรียกไฟล์จากทุก Path

5.1.5. จากนั้นปิด-เปิด tab ใหม่หนึ่งครั้งหรือรันคำสั่ง source /path/to/your/.bashrc แล้วลองดูว่าสามารถเรียกคำสั่ง cryptogen ได้หรือเปล่า ถ้าได้ต่อไปจะมาลองสร้าง Fabric network แบบ Hello World



```

1. methuzkaewsai khao@Methuzs-MacBook-Air: ~ (zsh)
X ..first-network (zsh) 1 X ~ (zsh) 2
→ ~ cryptogen
usage: cryptogen [<flags>] <command> [<args> ...]

Utility for generating Hyperledger Fabric key material

Flags:
  --help  Show context-sensitive help (also try
          --help-long and --help-man).

Commands:
  help [<command>...]
    Show help.

  generate [<flags>]
    Generate key material

  showtemplate
    Show the default configuration template

  version
    Show version information

→ ~

```

รูปที่ 15 แสดงการเรียกคำสั่ง cryptogen

## 5.2. Build your first network

โดยให้ clone repository ต้นต้นของ Hyperledger Fabric มาไว้ที่ Workspace

```
git clone -b master https://github.com/hyperledger/fabric-samples.git
```

ใน folder จะมีไฟล์ดังนี้

```

1. methuzkaewsai khao@Methuzs-MacBook-Air: ~/Workspace/hyperledger/fa...
X ..first-network (zsh) 1 X ..abric-samples (zsh) 2
→ fabric-samples git:(release) ls
LICENSE                               chaincode-docker-devmode
MAINTAINERS.md                       fabcar
README.md                            first-network
balance-transfer                     high-throughput
basic-network                        scripts
chaincode

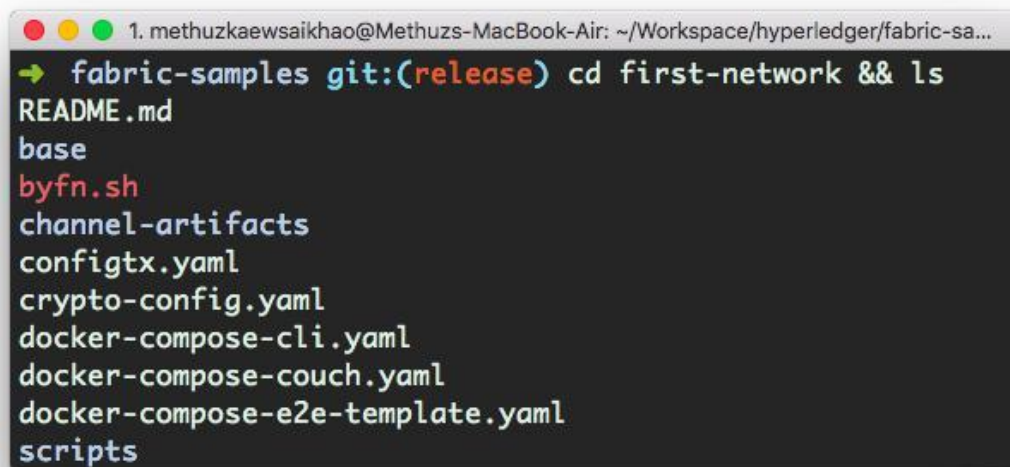
```

รูปที่ 16 การ clone repository มาไว้ที่ Workspace



5.2.1. ให้เข้าไปใน first-network จะมีไฟล์ดังนี้

```
cd first-network && ls
```



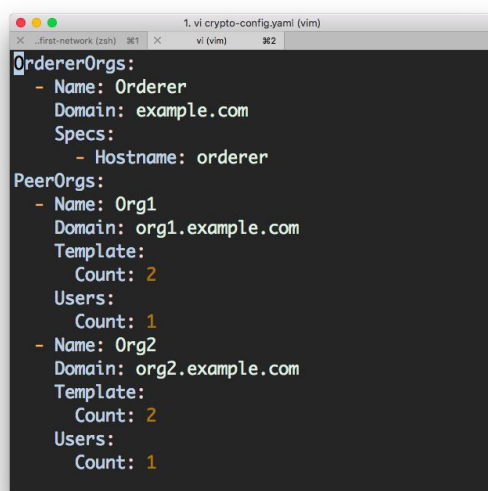
```
1. methuzkaewsaikhao@Methuzs-MacBook-Air: ~/Workspace/hyperledger/fabric-sa...
→ fabric-samples git:(release) cd first-network && ls
README.md
base
byfn.sh
channel-artifacts
configtx.yaml
crypto-config.yaml
docker-compose-cli.yaml
docker-compose-couch.yaml
docker-compose-e2e-template.yaml
scripts
```

รูปที่ 17 แสดงการเข้าไปใน first-network

5.2.2. จากนั้นจะใช้ byfn.sh ซึ่งเป็น script ที่ช่วยรันคำสั่งต่าง ๆ ทั้งหมดให้เราในการสร้าง ลบ หรือรีเซตเน็ตเวิร์กของเราให้รัน

```
./byfn -m generate
```

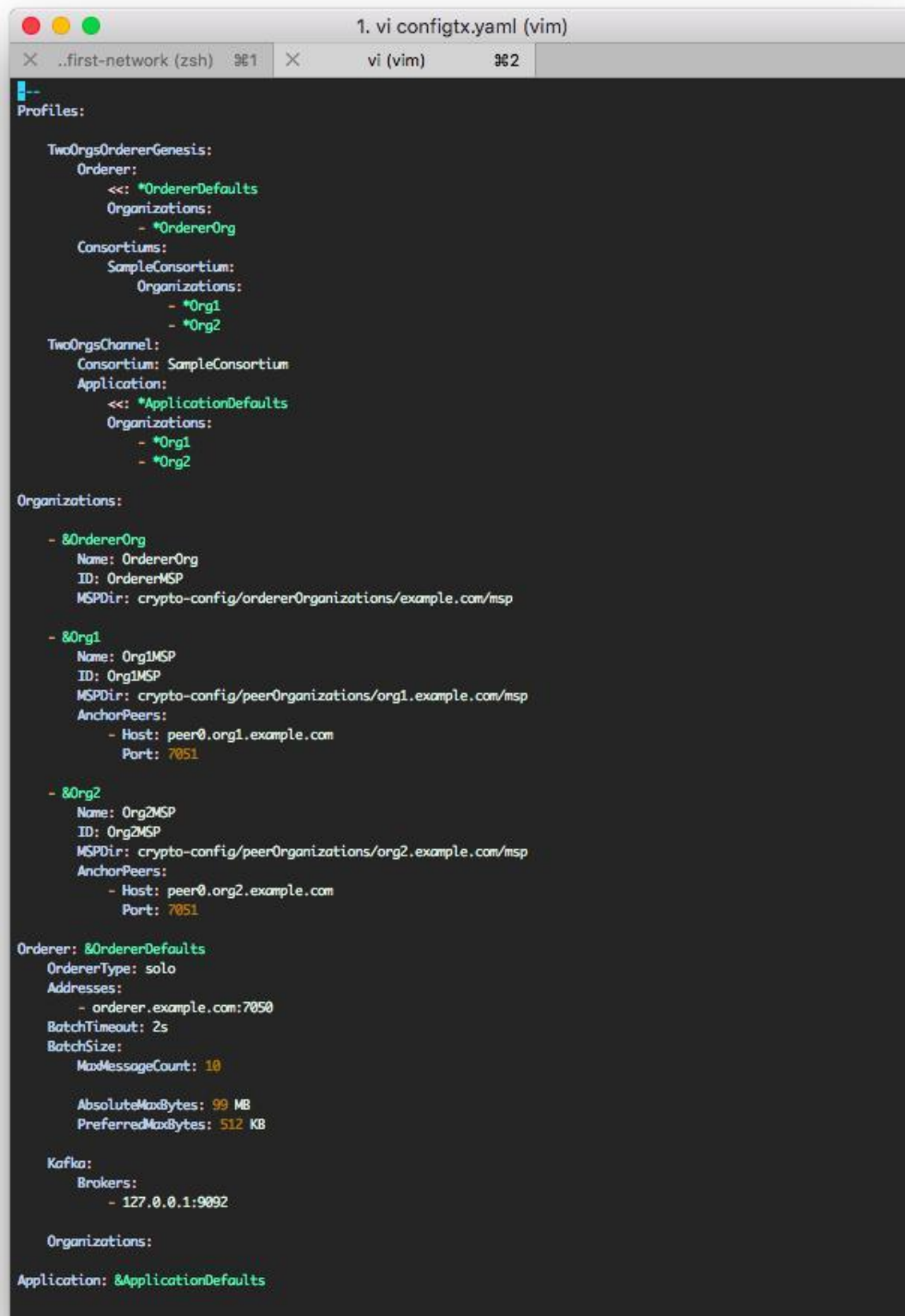
ซึ่งคำสั่งนี้จะทำการรับไฟล์ crypto-config.yaml และ configtx.yaml เข้าไปในคำสั่ง cryptogen และ configtxgen ตามลำดับ



```
1. vi crypto-config.yaml (vim)
OrdererOrgs:
- Name: Orderer
  Domain: example.com
  Specs:
  - Hostname: orderer
PeerOrgs:
- Name: Org1
  Domain: org1.example.com
  Template:
    Count: 2
  Users:
    Count: 1
- Name: Org2
  Domain: org2.example.com
  Template:
    Count: 2
  Users:
    Count: 1
```

รูปที่ 18 แสดงการเข้าไฟล์ cryptogen และ configtxgen

5.2.3. cryptoconfig file จะแสดง Spec ของ Orderer node และ Peer ต่าง ๆ ใน Organization โดย configtx.yaml จะเป็น spec ในการเชื่อมต่อกันระหว่าง Organization และ Orderer



```

--
Profiles:

  TwoOrgsOrdererGenesis:
    Orderer:
      <<: *OrdererDefaults
      Organizations:
        - *OrdererOrg
    Consortiums:
      SampleConsortium:
        Organizations:
          - *Org1
          - *Org2

  TwoOrgsChannel:
    Consortium: SampleConsortium
    Application:
      <<: *ApplicationDefaults
      Organizations:
        - *Org1
        - *Org2

Organizations:

  - &OrdererOrg
    Name: OrdererOrg
    ID: OrdererMSP
    MSPDir: crypto-config/ordererOrganizations/example.com/msp

  - &Org1
    Name: Org1MSP
    ID: Org1MSP
    MSPDir: crypto-config/peerOrganizations/org1.example.com/msp
    AnchorPeers:
      - Host: peer0.org1.example.com
        Port: 7051

  - &Org2
    Name: Org2MSP
    ID: Org2MSP
    MSPDir: crypto-config/peerOrganizations/org2.example.com/msp
    AnchorPeers:
      - Host: peer0.org2.example.com
        Port: 7051

Orderer: &OrdererDefaults
  OrdererType: solo
  Addresses:
    - orderer.example.com:7050
  BatchTimeout: 2s
  BatchSize:
    MaxMessageCount: 10

    AbsoluteMaxBytes: 99 MB
    PreferredMaxBytes: 512 KB

  Kafka:
    Brokers:
      - 127.0.0.1:9092

Organizations:

Application: &ApplicationDefaults

```

รูปที่ 19 แสดง Spec ของ Orderer node และ Peer ต่าง ๆ ใน Organization

### 5.3. ส่วนประกอบของ Hyperledger Network เบื้องต้น

ในระบบของ Hyperledger Fabric จะประกอบไปด้วย Server หลัก 3 ส่วนคือ

1. Application ซึ่งเป็นเหมือน Web Server ที่ยังคำสั่งเข้าไปใน Endorsing Peer เพื่อรับหรือแก้ไขข้อมูลต่าง ๆ
2. Peer ซึ่งเป็นตัวแทนขององค์กรหรือ Organization ซึ่งจะรับคำสั่งจาก Application โดยจะแบ่งเป็น 2 ประเภทคือ

2.1 Endorsing Peer ทำหน้าที่เช็ค Signature และจำลองการทำงานของคำสั่งจาก Application ออกมาเป็น RW (ReadWrite) Set ส่งกลับไปให้ Application

2.2 Committing Peer ทำหน้าที่ Verify RW Set ว่าตรงกับ World State ปัจจุบันของ Blockchain หรือไม่

- ถ้าใช่จะ Commit Block ใหม่โดยทำงานตาม RW Set นั้นและแก้ไข World State
- แต่ถ้าไม่ใช่จะ Commit Block ใหม่โดยรวมเอา RW Set นั้นเข้าไปด้วยแต่จะ Mark ว่า Invalid และไม่แก้ไข World State

3. Orderer เป็นตัวจัดการข้อตกลงร่วมหรือ Consensus ทั้งหมดของระบบซึ่งเป็นตัวหลักแยกออกมาต่างหาก ข้อดีของมันคือทำให้สามารถแก้ไขเปลี่ยนแปลง Consensus ภายหลังได้ง่าย Orderer จะรับคำสั่งของ Application ที่ถูก Endorsed จาก Endorsing Peer แล้ว ส่งไปให้ Committing Peer

ทำการ Verify โดย Consensus ที่มีในปัจจุบันคือ

- Solo ซึ่งใช้ในระหว่าง Development
- Kafka ใช้ในระบบจริง
- Simplified Byzantine Fault Tolerance (SBFT) กำลังพัฒนาแต่ยังไม่เสร็จ
- Raft มีแผนที่จะพัฒนาเร็ว ๆ นี้

## 6. Blockchain for Hyperledger Fabric v1.0

### 6.1. บริการ IBM Blockchain for Hyperledger Fabric v1.0

มีจุดเด่นที่น่าสนใจ ดังนี้

- สามารถป้องกันการโจมตีจากภายในองค์กรเองได้ โดยเฉพาะภัยคุกคามที่เกิดขึ้นจากผู้ใช้งานซึ่งในระดับ System Administrator
- ได้รับมาตรฐานความมั่นคงปลอดภัย Evaluation Assurance Level Certification EAL5+
- ทำระบบ Blockchain ให้อยู่ในรูปของ Virtual Appliance ซึ่งมีความปลอดภัยสูง ปกป้องทั้งโค้ดและข้อมูลไม่ให้บุคคลภายนอกเข้าถึงได้
- มี Hardware Security Module ที่ได้รับมาตรฐาน FIPS 140-2 Level 4 สามารถป้องกันการปลอมแปลงข้อมูลได้ในระดับสูง
- การทำงานทั้งหมดสามารถ Audit ได้เสมอ

โดยการเปิดตัว IBM Blockchain Service ที่ใช้ Hyperledger Fabric นี้ทำให้ IBM กลายเป็นผู้ให้บริการ Cloud รายแรกที่รองรับการให้บริการ Cloud Blockchain ที่มีความมั่นคงปลอดภัยและประสิทธิภาพระดับสูงสำหรับองค์กรได้ และมีลูกค้าผู้ใช้งานกลุ่มแรกทั้งจากธุรกิจ Startup ไปจนถึงสถาบันการเงิน

## 7. Hyperledger Fabric 1.1

ทีมพัฒนา Hyperledger Fabric โครงการระบบ Open Source Distributed Ledger ได้ออกมาประกาศเปิดตัว Hyperledger Fabric 1.1 โดยมีฟีเจอร์ใหม่ๆ ที่น่าสนใจดังนี้

- รองรับ Node.js Chaincode
- เพิ่ม Channel based event service เข้ามา
- สามารถใส่ CouchDB Index เข้าไปใน Chaincode เพื่อเพิ่มประสิทธิภาพของระบบได้
- สามารถสร้าง Certificate Revocation List (CRL) ได้
- สามารถทำการอัปเดต Client Identity และ Affiliation ได้แบบ Dynamic
- มี Node.js SDK Connection Profile เพื่อให้การเชื่อมต่อไปยัง Fabric Node ง่ายขึ้น
- ทำ Mutual TLS ระหว่างแต่ละ Fabric Node และระหว่าง Client มายัง Node
- สามารถเข้ารหัสข้อมูล Ledger ได้ด้วย Chaincode Encryption Library
- มี Attributed-based Access Control ใน Chaincode
- มี Chaincode API สำหรับใช้เข้าถึงค่า Client Identity ได้
- ปรับปรุงประสิทธิภาพ ทำให้มี Transaction Throughput สูงขึ้น Response Time น้อยลง

ทั้งนี้เหล่าเอกสารคู่มือและ Tutorial ต่างๆ ถูกปรับปรุงให้ดีขึ้น เชื่อว่าน่าจะทำให้ นักพัฒนา สามารถศึกษาเทคโนโลยีนี้ได้ง่ายขึ้น และ Hyperledger Fabric 1.1 นี้รองรับการอัปเดตขึ้นมาจากรุ่น 1.0 ได้ด้วย โดยโครงการ Hyperledger Fabric นี้เป็นความร่วมมือระหว่าง Alibaba, Belltane, BlocLedger, IBM, Oracle, SecureKey และบริษัทอื่น ๆ อีกมากมาย

## 8. Hyperledger Fabric จัดการกับ Concurrency อย่างไร

### 8.1. ลำดับการทำงานของ Fabric ในการอ่านและเขียนข้อมูล

1. เมื่อ Client ต้องการแก้ไขข้อมูลจะทำการร้องขอไปที่ Peer
2. Endorsing Peer จะทำการจำลองการทำงานของธุรกรรมว่าถูกต้องไหมตาม Chaincode/Smartcontract
3. ถ้าทุกอย่างถูกต้อง Endorsing Peer จะคืนสิ่งที่เรียกว่า R/WSet (Read Write Set) กลับมาให้ Client
4. Client จะทำการเช็คธุรกรรมถูกรับรอง Endorsed) ( ครบตามนโยบายของ Chaincode หรือไม่)
5. ถ้าครบ Client จะส่ง R/W Set ไปให้ Orderer (Kafka หรือ Solo)

6. Orderer จะทำการเช็ค R/W Set มีเวอร์ชันของข้อมูลที่ถูกต้องหรือเปล่า ถ้าถูกจะ Queue ไว้และส่งไปให้ Committing Peer (ด้วยวิธี Polling) ถ้าครบ Interval (เช่น 2 วินาที)
7. Committing Peer จะทำการบันทึก Block แล้วส่งไปบอก Client ผ่าน EventHub

## 8.2. การทำงานแบบ Stateful ใน Hyperledger Fabric

Hyperledger Fabric แม้จะมีการทำงานแบบ Stateful แต่มีการเช็คความปลอดภัยของการแก้ไข 2 ชั้นคือเช็คด้วย Chaincode หรือ Smart Contract ที่ Endorsing Peer และเช็คด้วยเวอร์ชันของข้อมูล (MVCC) เพื่อให้แน่ใจว่าใช้ข้อมูลเวอร์ชันล่าสุดเพื่อประกอบการทำงาน ทำให้สามารถสร้าง Transaction ที่มีความสามารถ Atomicity และ Isolation โดยยังคงความเร็วในการอ่านไว้ได้ แต่แลกมาด้วยการที่ต้องเก็บข้อมูลมากขึ้น

ในการทำระบบเพื่อให้รองรับลูกค้าหรือผู้ใช้งานจำนวนมากการรองรับปริมาณธุรกรรมจำนวนมากเป็นสิ่งสำคัญโดยเฉพาะอย่างยิ่งการทำระบบที่เกี่ยวข้องกับการเงินทั้งต้องรองรับธุรกรรมจำนวนมากและที่ต้องไม่มีข้อผิดพลาดในเวลาเดียวกัน ซึ่ง Hyperledger Fabric เองเป็นแพลตฟอร์มที่นำมาใช้สร้างฐานข้อมูลแบบ Blockchain ระหว่างองค์กรได้อย่างหลากหลายซึ่งเรานำข้อมูลอะไรไปเก็บไว้ก็ได้ เพียงแต่ต้องกำหนดรูปแบบของข้อมูลให้ดีและการที่เราจะเก็บข้อมูลอะไรสักอย่างไว้ใน Fabric เราจะต้องเรียกฟังก์ชัน

```
setState(<key>,<value>)
```

### 8.2.1. Stateful จดจำการติดต่อ

Stateful หรือ Connection-oriented คือการที่ Server จดจำ Client ที่เข้ามายังเว็บไซต์ เทคนิคที่นำมาใช้ในการจดจำผู้ใช้แต่ละคนประกอบด้วย

- Cookies คือข้อมูลขนาดเล็กที่ถูกส่งมาจาก Server เพื่อเก็บลงในเครื่องผู้ใช้โดยมี Web Browser เป็นผู้จัดการ
- Session คือ object ที่ถูกสร้างขึ้นบน Server เพื่อเก็บข้อมูลของผู้ใช้แต่ละคน โดยจะเกิดในระหว่างที่มีการติดต่อกับผู้ใช้ในระยะเวลาสั้นๆ Web Server จะเป็นผู้จัดการ

### 8.2.2. การควบคุมการเข้าถึงข้อมูลพร้อมกัน

Concurrency คือ การที่มีทรานแซกชันหลาย ๆ ทรานแซกชันต้องการเรียกใช้ข้อมูลเดียวกันในเวลาเดียวกันจากฐานข้อมูลเพื่อทำงานของแต่ละทรานแซกชัน

Concurrency Control คือ กระบวนการของการจัดการกับ operations ต่าง ๆ ที่ประมวลผลในเวลาเดียวกัน ดังนั้นความถูกต้องของข้อมูลต้องถูกรักษาไว้ operations เหล่านั้นและต้องไม่ขัดขวางหรือขัดแย้งกันเองภายในระบบฐานข้อมูลแบบมีผู้ใช้หลายคน (Multiuser environment)

### 8.3. R/W Set บน MVCC

Multiversion Currency Control (MVCC) คือเป็นวิธีการจัดการธุรกรรมอย่างหนึ่งที่เข้ามาแทนที่การ Lock ถ้าใครเคยใช้ MongoDB เราอาจจะคุ้นเคยกับการถูกล็อค Collection เพื่อป้องกันการเขียนที่ผิดพลาดในขณะที่ถูก Lock ไม่มีใครสามารถอ่านข้อมูลได้

MVCC จึงเริ่มเป็นที่นิยมมากขึ้น (ถูกใช้ในPostgresql) เพราะถึงแม้เราจะต้องการให้การเขียนข้อมูลถูกต้องแต่เรายังอยากอ่านข้อมูลด้วยความเร็วสูงด้วย หลักการคร่าว ๆ ของ MVCC คือการให้ข้อมูลทุกอย่างมีเวอร์ชันของตัวเอง และเมื่อมีการส่งข้อมูลให้ Client จะแนบ Version ไปกับข้อมูลด้วย ทำให้ Client ยังสามารถอ่านข้อมูลได้เสมอ และเมื่ออัปเดตสามารถเช็คกับ Version ได้ว่าเราอัปเดตกับข้อมูลล่าสุดหรือไม่

Hyperledger ใช้หลักการของ MVCC มารวมเข้ากับ Blockchain เพื่อป้องกันการผิดพลาดในการแก้ไขข้อมูลและยังคงการอ่านข้อมูลด้วยความเร็วสูงไว้ โดยใช้ BlockHeight เป็นเลขเวอร์ชันของข้อมูลแต่ละตัว แต่ Hyperledger ได้ทำเหนือไปกว่านั้นคือการจัดการกับ R/W Set ผ่าน Orderer เพื่อทำการป้องกันความผิดพลาดโดยที่ผู้พัฒนา Chaincode ไม่จำเป็นต้องปวดหัวกับการจัดการ Version

ถ้าเราเขียนฟังก์ชันซึ่งมีการอ่านและเขียนข้อมูล Endorsing เมื่อ Peer ทำการจำลองการทำงานเสร็จเรียบร้อยตาม Chaincode มันจะส่ง R/W Set กลับมาที่ Client เมื่อข้อมูลถูกส่งไปที่ Orderer (สไลด์ที่6) ตัว Orderer จะทำการเช็คเวอร์ชันโดยเฉพาะของ **Read Set** เพื่อป้องกันการทำงานที่ผิดพลาดจากการอ่านค่าที่ผิด Version ส่วน Chaincode เราจะ Write อะไรนั้น Orderer ไม่ได้สนใจ แต่ถ้าเราได้ข้อมูลผิดเวอร์ชันไปทำงานก็มีโอกาสสูงมากที่ข้อมูลถูกเขียนผิด

ตัวอย่าง Chaincode ส่ง R/W Set กลับมาที่ Client

```
<TxReadWriteSet>
  <NsReadWriteSet name="chaincode1">
    <read-set>
      <read key="userA", version="1">
      <read key="userB", version="1">
    </read-set>
    <write-set>
      <write key="userA", value="0">
      <write key="userB", value="20">
    </write-set>
  </NsReadWriteSet>
</TxReadWriteSet>
```

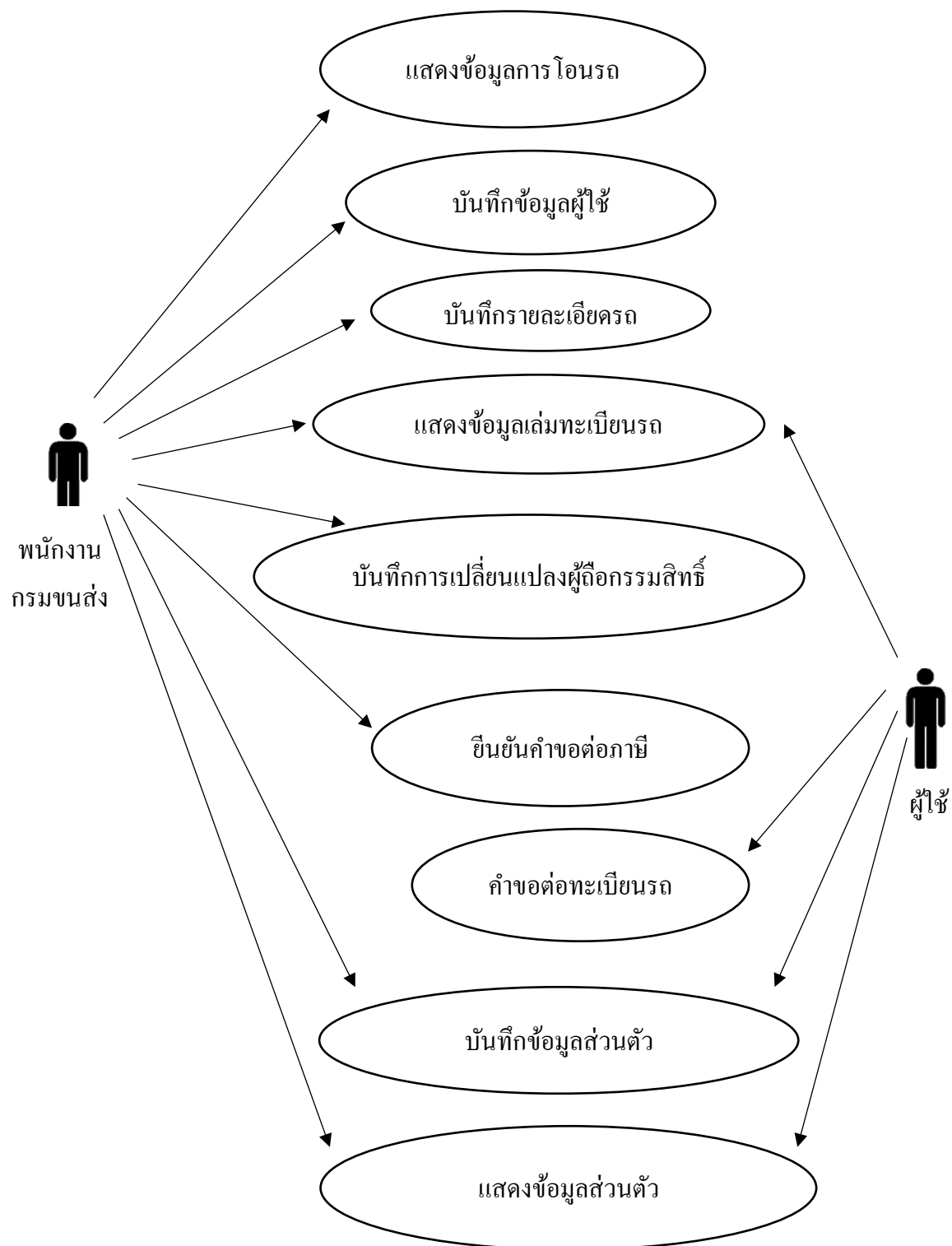
ด้วยหลักการนี้หลักการเดียวทำให้ Hyperledger Fabric สามารถทำให้ Chaincode/Smartcontract ของเราสร้าง Transaction ที่มีความสามารถ Atomicity และ Isolation ได้โดยอัตโนมัติ

### 8.3.1. ข้อเสียของ MVCC

จำเป็นต้องเก็บข้อมูลมากกว่าปกติ เพราะต้องเก็บข้อมูลหลายเวอร์ชันของแต่ละ Document

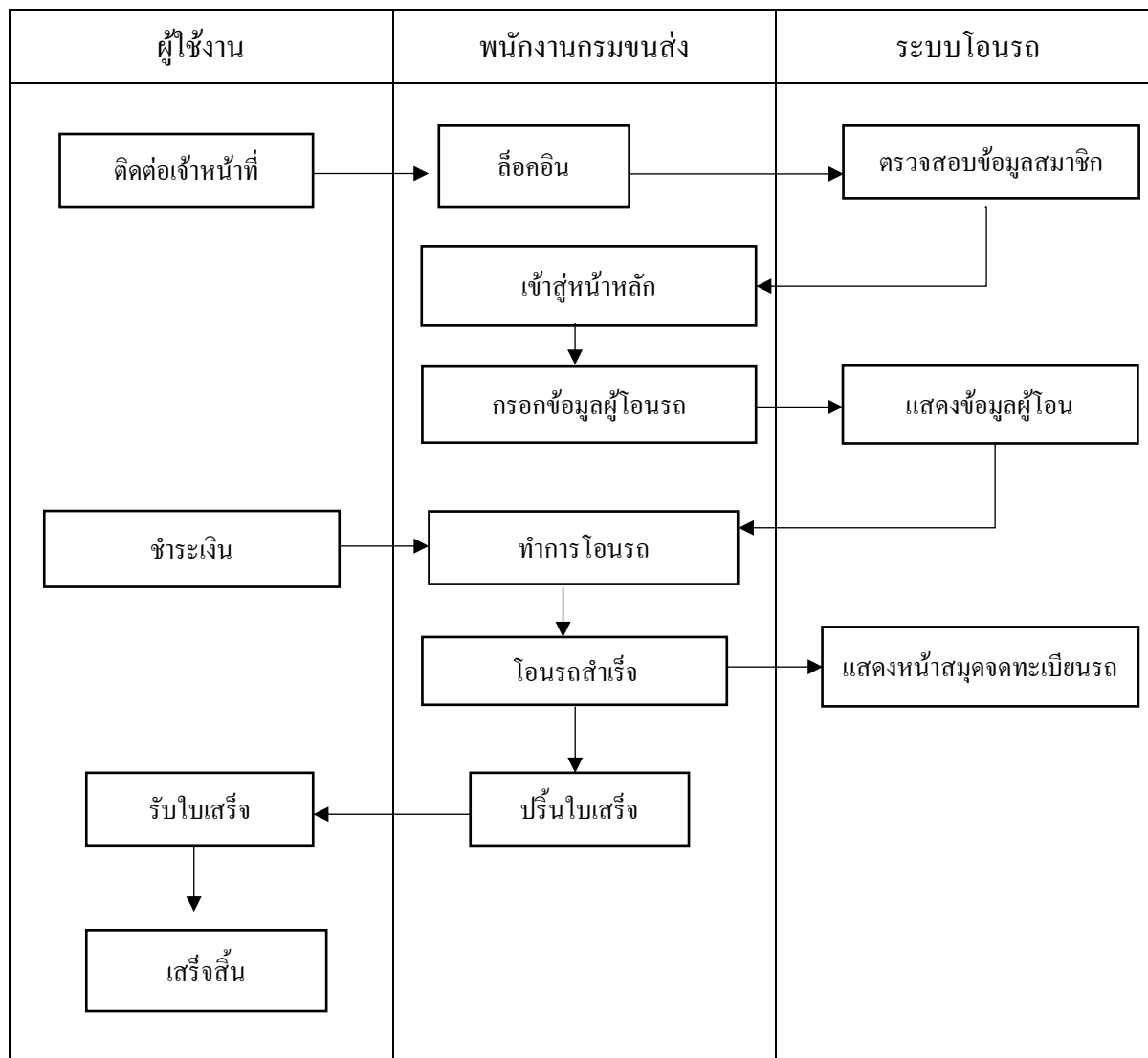
### บทที่ 3 การวิเคราะห์ระบบ

#### 3.1. Use Case Diagram



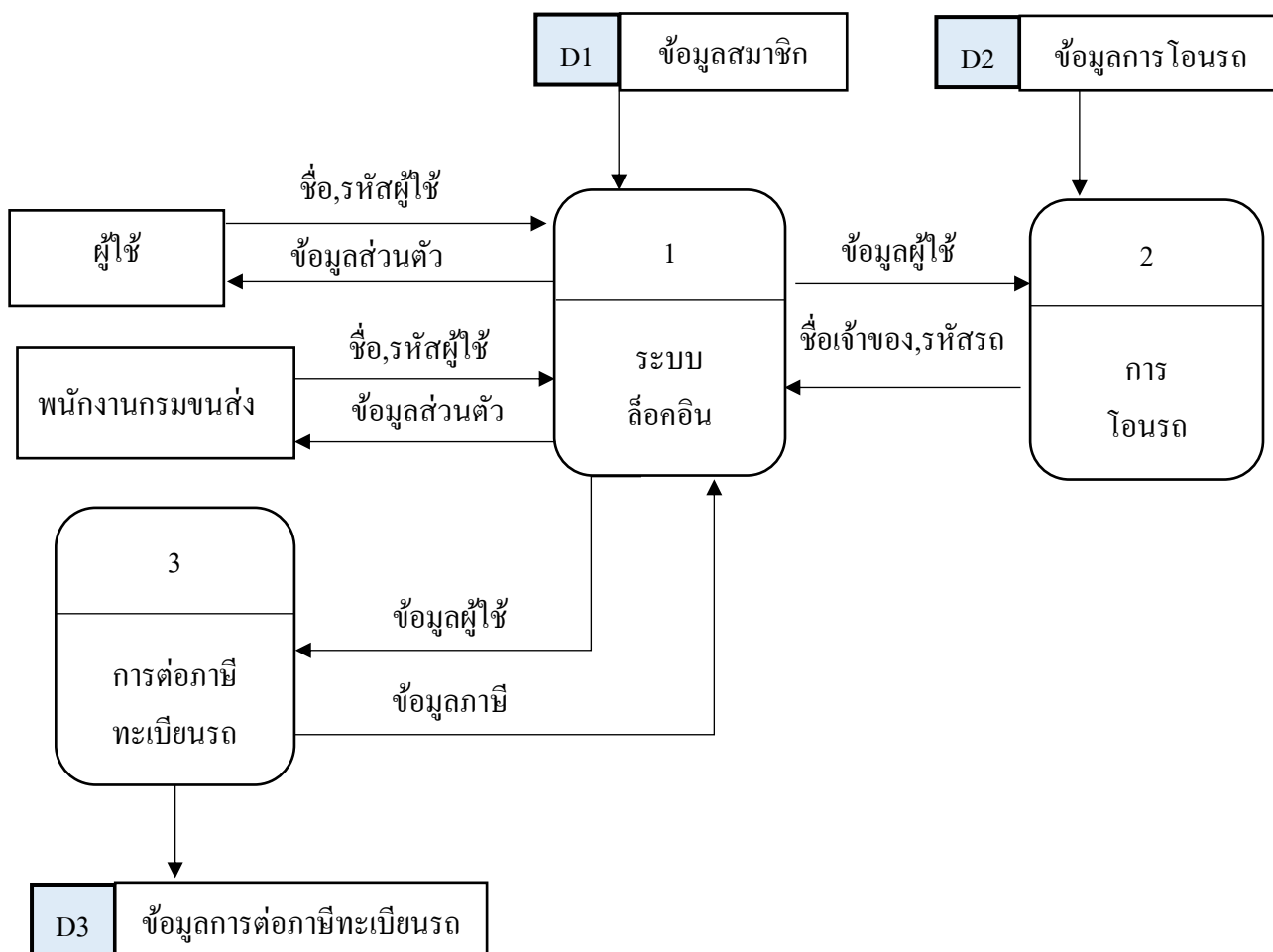
### 3.2. Work Flow Diagram

#### 3.2.1. ขั้นตอนการโอนรถ





## 3.3. Data Flow Diagram



### 3.4. Process Description

#### 3.4.1. ระบบล็อกอิน

| Process Description |                                                                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| System              | เล่มทะเบียนรถบนบล็อกเชน                                                                                                                                                               |
| DFD number          | 1                                                                                                                                                                                     |
| Process name        | ระบบล็อกอิน                                                                                                                                                                           |
| Input Data Flow     | ชื่อผู้ใช้, รหัสผู้ใช้                                                                                                                                                                |
| Output Data Flow    | ผลการเข้าสู่ระบบ                                                                                                                                                                      |
| Data stored user    | D1 ข้อมูลสมาชิก                                                                                                                                                                       |
| Description         | เป็นโปรเซสสำหรับล็อกอินเพื่อเข้าสู่ระบบ โดยต้องทำการกรอกข้อมูลชื่อบัญชีผู้ใช้งานและรหัสผ่านให้ถูกต้อง จึงจะสามารถเข้าสู่ระบบได้ หากกรอกข้อมูลผิดระบบจะแจ้งให้ทราบและให้กรอกข้อมูลใหม่ |

#### 3.4.2. การโอนรถ

| Process Description |                                                                                                                                           |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| System              | เล่มทะเบียนรถบนบล็อกเชน                                                                                                                   |
| DFD number          | 2                                                                                                                                         |
| Process name        | การโอนรถ                                                                                                                                  |
| Input Data Flow     | ชื่อผู้ใช้, รหัสผู้ใช้, รหัสรถ                                                                                                            |
| Output Data Flow    | ข้อมูลการถือครองรถ                                                                                                                        |
| Data stored user    | D2 ข้อมูลการโอนรถ                                                                                                                         |
| Description         | เป็นโปรเซสสำหรับเปลี่ยนแปลงผู้ถือครองกรรมสิทธิ์รถ โดยผู้โอนและผู้รับโอนต้องทำการกรอกข้อมูลส่วนตัวและรายละเอียดข้อมูลรถ จึงจะทำการโอนรถได้ |

## 3.4.3. การต่อภาษีทะเบียนรถ

| Process Description |                                                                                                                                          |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| System              | เล่มทะเบียนรถบนบล็อกลูกเซน                                                                                                               |
| DFD number          | 3                                                                                                                                        |
| Process name        | การต่อภาษีทะเบียนรถ                                                                                                                      |
| Input Data Flow     | ชื่อผู้ใช้, รหัสผู้ใช้, รหัสรถ, ป้ายทะเบียน                                                                                              |
| Output Data Flow    | ใบพรบ. ประจำปี                                                                                                                           |
| Data stored user    | D3 ข้อมูลการต่อภาษีทะเบียนรถ                                                                                                             |
| Description         | เป็นโปรเซสสำหรับการต่อภาษีทะเบียนรถประจำปี โดยเจ้าของรถต้องทำการยื่นเรื่องไปยังกรมการขนส่งก่อน แล้วชำระเงินที่เคาร์เตอร์ จึงเสร็จสมบูรณ์ |

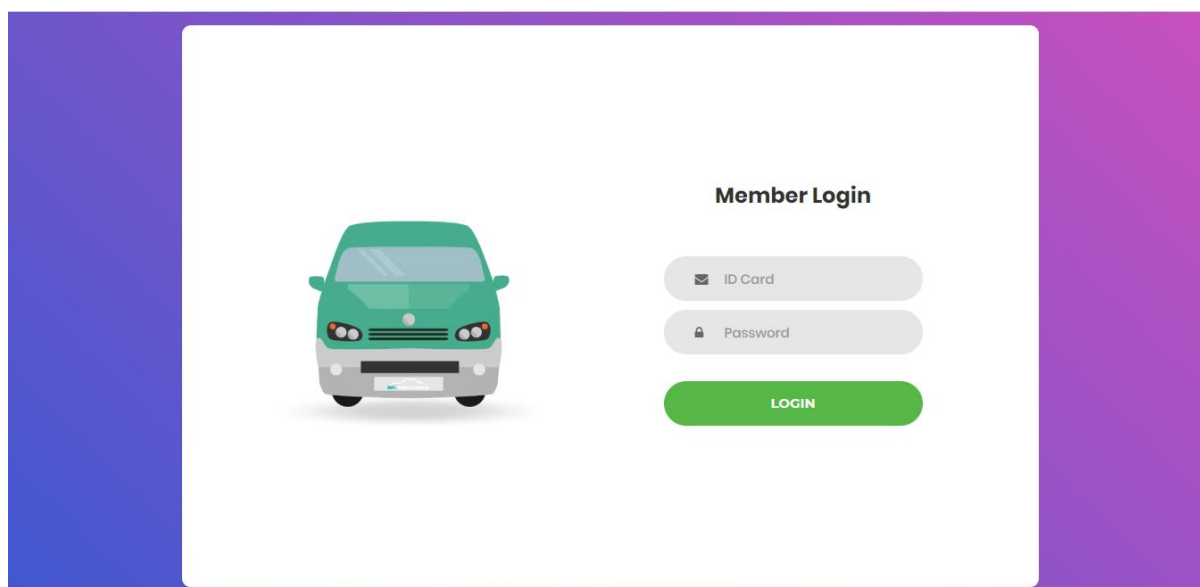
## บทที่ 4 ผลการดำเนินการ

### 4.1. ผลการดำเนินงาน

จากการดำเนินงานกลุ่มผู้จัดทำได้ทำการสร้างหน้าเว็บหลัก เพื่อให้ผู้ใช้งานสามารถใช้งานได้ดังนี้



รูปที่ 1 หน้าหลัก



รูปที่ 2 หน้าล็อกอิน

สวัสดี คุณNichamon Promthong

| ค้นหาทะเบียนรถ |                    |        |
|----------------|--------------------|--------|
| ID Card        | หมายเลขบัตรประชาชน | Submit |
| ป้ายทะเบียน    | ex. กง-9999        | Submit |
| เลขทะเบียนรถ   | หมายเลขทะเบียนรถ   | Submit |

| ค้นหาผู้ใช้งาน |                                    |        |
|----------------|------------------------------------|--------|
| ID Card        | หมายเลขบัตรประชาชน                 | Submit |
| ชื่อ-นามสกุล   | ชื่อ(ไม่ต้องมีคำนำหน้า)    นามสกุล | Submit |

| ดำเนินการ      |                |                   |
|----------------|----------------|-------------------|
| เพิ่มผู้ใช้งาน | เพิ่มทะเบียนรถ | ตรวจสอบข้อมูลภาษี |

รูปที่ 3 หน้าหลักของแอดมิน

| ข้อมูลรถ             |               |                   |         |          |          |      |      |                      |           |      |               |       |
|----------------------|---------------|-------------------|---------|----------|----------|------|------|----------------------|-----------|------|---------------|-------|
| เลขเดิม<br>ทะเบียนรถ | วันที่โอน     | ป้าย<br>ทะเบียนรถ | จังหวัด | ประเภทรถ | ยี่ห้อรถ | รุ่น | ปี   | ผู้ถือ<br>กรรมสิทธิ์ | ดำเนินการ |      |               |       |
| i8wou9h9             | 30.April.2020 | dd 2222           | สงขลา   | asdf     | asdf     | asdf | 2019 | 3075                 | ประวัติรถ | ภาษี | แก้ไขข้อมูลรถ | โอนรถ |

รูปที่ 4 ข้อมูลรถหน้าแอดมิน

ประวัติรถ

| เลขเดิม<br>ทะเบียนรถ | วันที่โอน  | ป้าย<br>ทะเบียนรถ | จังหวัด | ประเภทรถ | ยี่ห้อรถ | รุ่น | ปี   | สี      | หมายเลข<br>รถ | ประเภท<br>เครื่องยนต์ | จำนวน<br>ลูกสูบ | ขนาด<br>เครื่อง | แรงม้า | หมายเลข<br>เครื่อง | ประเภท<br>เชื้อ<br>เพลิง |
|----------------------|------------|-------------------|---------|----------|----------|------|------|---------|---------------|-----------------------|-----------------|-----------------|--------|--------------------|--------------------------|
| i8wou9h9             | 2020-04-30 | dd 2222           | สงขลา   | asdf     | asdf     | asdf | 2019 | sdfasdf | 151sdf        | asdf                  | 4               | 400             | 4000   | asdf               | sdf                      |
| i8wou9h9             | 2020-04-30 | dd 1111           | sdfsdf  | asdf     | asdf     | asdf | 2019 | sdfasdf | 151sdf        | asdf                  | 4               | 400             | 4000   | asdf               | sdf                      |
| i8wou9h9             | 2020-04-30 | dd 1111           | sdfsdf  | asdf     | asdf     | asdf | 2019 | sdfasdf | 151sdf        | asdf                  | 4               | 400             | 4000   | asdf               | sdf                      |

รูปที่ 5 ประวัติรถ

## ประวัติการต่อภาษีประจำปี

| หมายเลขคำร้อง | ป้ายทะเบียนรถ | วันที่ขอ      | วันสิ้นสุด | ผู้ดำเนินการ    |
|---------------|---------------|---------------|------------|-----------------|
| p3j0afa       | กง-1111       | 30 April 2020 | April 2021 | ชัยชนะ คล่องแจะ |

## รูปที่ 6 ประวัติการต่อภาษีประจำปี

## แก้ไขข้อมูลรถ

| เลขเดิมทะเบียนรถ | ป้ายทะเบียนรถ | จังหวัด | ประเภทรถ | ยี่ห้อรถ | รุ่น | ปี   | สี | ผู้ถือกรรมสิทธิ์ |
|------------------|---------------|---------|----------|----------|------|------|----|------------------|
| 0hwexopq         | กง 1111       | สงขลา   | ส่วนตัว  | nis      | gtr  | 2019 | ดำ | 19098            |

| เปลี่ยนหมายเลขป้ายทะเบียนรถ           |                                          |
|---------------------------------------|------------------------------------------|
| หมายเลขป้ายทะเบียนเดิม                | กง-1111                                  |
| หมายเลขป้ายทะเบียนใหม่                | <input type="text" value="ex. กก-1111"/> |
| จังหวัดเดิม                           | สงขลา                                    |
| จังหวัดใหม่                           | <input type="text" value="ex. สงขลา"/>   |
| <input type="button" value="Submit"/> | Status:                                  |

| เปลี่ยนสีรถ                           |                                           |
|---------------------------------------|-------------------------------------------|
| สีเดิม                                | ดำ                                        |
| สีใหม่                                | <input type="text" value="ex. ดำ - ขาว"/> |
| <input type="button" value="Submit"/> | Status:                                   |

## รูปที่ 7 การแก้ไขข้อมูลรถ

| โอนรถ                                 |                                             |
|---------------------------------------|---------------------------------------------|
| หมายเลขทะเบียนรถ                      | 0hwexopq                                    |
| ผู้รับโอน                             | <input type="text" value="เลขบัตรประชาชน"/> |
| ผู้ครอบครอง                           | <input type="text" value="เลขบัตรประชาชน"/> |
| เลขที่สัญญา                           | <input type="text" value="เลขที่สัญญา"/>    |
| <input type="button" value="Submit"/> |                                             |

## รูปที่ 8 การโอนรถ

## ข้อมูลผู้ใช้

| ID Card | ชื่อ  | นามสกุล | วันเกิด | เดือนเกิด | ปี   | หมายเลขโทรศัพท์ | Email             | สัญชาติ | ที่อยู่ |
|---------|-------|---------|---------|-----------|------|-----------------|-------------------|---------|---------|
| 19098   | สมชาย | เลิศดี  | 12      | 05        | 2000 | 0985462518      | somchai@gmail.com | THAI    | 122/30  |

## รูปที่ 9 ข้อมูลการติดต่อของผู้ใช้

| เพิ่มผู้ใช้งาน   |                    |
|------------------|--------------------|
| ID Card          | หมายเลขบัตรประชาชน |
| Password         | Password           |
| ชื่อ-นามสกุล     | ชื่อ นามสกุล       |
| ประเภทผู้ใช้     | ประเภทผู้ใช้       |
| วัน-เดือน-ปีเกิด | ex. 11-11-2011     |
| Email            | ex. abc@gmail.com  |
| เบอร์โทรศัพท์    | ex. 0812345678     |
| สัญชาติ          | กรอกสัญชาติ        |
| ที่อยู่          | ที่อยู่ผู้ใช้      |
| Submit           | Status:            |

## รูปที่ 10 การเพิ่มผู้ใช้งาน

| ตรวจสอบคำขอต่อภาษี |             |
|--------------------|-------------|
| หมายเลขทะเบียนรถ   | ex. กก-9999 |
| Submit             |             |

## รูปที่ 11 ตรวจสอบคำขอต่อภาษี

| เพิ่มทะเบียนรถ        |                     |
|-----------------------|---------------------|
| เลขป้ายทะเบียน        | ex. กก-9999         |
| จังหวัด               | ex. สงขลา           |
| ประเภทรถ              | ex. รถยนต์ส่วนบุคคล |
| Brand                 | ex. Nissan          |
| Model                 | ex. GTR-R34         |
| ปี                    | ex. 2019            |
| สี                    | ขาว-ดำ              |
| หมายเลขตัวรถ          | ex. 15D56F1D        |
| ตำแหน่งหมายเลขตัวรถ   | ex. ขวา             |
| ประเภทเครื่องยนต์     | ex. RB26            |
| จำนวนลูกสูบ           | ex. 6               |
| ขนาดเครื่องยนต์       | ex. 2568cc          |
| แรงม้า                | ex. 280 แรงม้า      |
| หมายเลขเครื่อง        | ex. 1R5EE69E        |
| ตำแหน่งหมายเลขเครื่อง | ex. ขวาเครื่อง      |
| เชื้อเพลิง            | ex. เบนซิน          |
| น้ำหนักตัวรถ          | ex. 1490 กก         |
| จำนวนที่นั่ง          | ex. 4               |
| ผู้ถือกรรมสิทธิ์      | กรอกหมายเลขบัตร ปชช |
| ผู้ครอบครอง           | กรอกหมายเลขบัตร ปชช |
| สัญญาซื้อ             | หมายเลขสัญญาซื้อ    |
| Submit                | Status:             |

รูปที่ 12 การเพิ่มทะเบียนรถ




## สวัสดี คุณชัยชนะ คลองแงะ

| ออกใบตรวจสอบสภาพรถ |             |
|--------------------|-------------|
| หมายเลขทะเบียนรถ   | ex. กก-9999 |
| Submit             | Status:     |

รูปที่ 13 การออกใบตรวจสอบสภาพรถ

 or number
 ออกจากระบบ



สวัสดี คุณWuttichai Dalunchim

| เลขเล่มทะเบียนรถ | วันที่จดทะเบียน | ป้ายทะเบียนรถ | จังหวัด | ประเภทรถ | ยี่ห้อรถ | รุ่น | ปี   | สี      | ดำเนินการ                            |
|------------------|-----------------|---------------|---------|----------|----------|------|------|---------|--------------------------------------|
| i8wou9h9         | 30.April.2020   | dd 2222       | สงขลา   | asdf     | asdf     | asdf | 2019 | sdfasdf | <div>ประวัติรถ</div> <div>ภาษี</div> |

ติดต่อ

สามารถติดต่อได้ที่นี้

↑

รูปที่ 14 แสดงข้อมูลรถที่ครอบครอง

## บทที่ 5 สรุปผลการดำเนินงาน และข้อเสนอแนะ

### 5.1. สรุปผลการดำเนินงาน

ผลการศึกษาและจัดทำโครงการพบว่า จากการดำเนินงานผู้จัดทำสามารถจัดทำโครงการได้อย่างทันตามเวลาและตามแผนงานที่ได้ระบุไว้ โดยเว็บที่สร้างขึ้นมาผู้ใช้สามารถเพิ่มและแก้ไขข้อมูลส่วนตัวได้ และสามารถดูข้อมูลกรรมสิทธิ์การถือครองรถของตนเองได้ แต่ในเรื่องของการโอนการถือครองจากคนหนึ่งไปยังอีกคนหนึ่งมีเพียงเจ้าหน้าที่แอดมินเท่านั้นที่สามารถทำได้ ซึ่งแอดมินสามารถดูข้อมูลทุกอย่างของลูกค้าผู้ถือครองกรรมสิทธิ์รถได้หมดทุกคน และสามารถทำการโอนรถและแก้ไขข้อมูลได้เพียงผู้เดียว

### 5.2. ปัญหาและอุปสรรค

จากการศึกษาพบว่า Blockchain เป็นเทคโนโลยีใหม่สำหรับคณะผู้จัดทำมาก ทำให้ต้องศึกษาหาความรู้ถึงวิธีการติดตั้งอุปกรณ์การใช้งาน และศึกษาขั้นตอนวิธีการใช้งานกันพอสมควร ทำให้งานมีความล่าช้าอยู่บ้าง และเมื่อเว็บเกิดปัญหา error ทำให้ผู้จัดทำต้องไล่การเขียน Code เพื่อหาว่าปัญหาเกิดจากตรงไหน และต้องใช้เวลาในการหาวิธีการแก้ไข เพื่อให้เว็บกลับมาใช้งานได้อย่างสมบูรณ์ที่สุด

### 5.3. ข้อเสนอแนะและแนวทางในการพัฒนา

จากการจัดทำโครงการเรื่องเล่มทะเบียนรถบนบล็อกเชน ( Car registration on the Blockchain ) พบว่า สามารถพัฒนาระบบให้สามารถใช้งานได้จริง โดยการเพิ่มระบบการจ่ายเงินเข้ามาเพื่อให้ผู้ใช้และเจ้าหน้าที่ทำงานได้อย่างรวดเร็วและประหยัดเวลา ทั้งยังทำให้การดำเนินงานมีประสิทธิภาพทั้งเรื่องของการต่อทะเบียนรถ ต่อพรบ. ชำระค่าภาษีประจำปี เป็นต้น ป้องกันข้อมูลที่ผิดพลาดได้เป็นอย่างดี ทั้งยังได้ข้อมูลที่น่าเชื่อถืออีกด้วย

นอกจากนี้เทคโนโลยีบล็อกเชนยังสามารถพัฒนามาใช้ในเรื่องของการบันทึกทะเบียนการแพทย์, การจัดการบริหารทะเบียนแบบอื่น ๆ เช่น การจัดการผู้มีสิทธิเข้าถึงระบบ (identity management), การประมวลผลธุรกรรม, การสร้างเอกสารแสดงความเป็นเจ้าของ, การตามรอยการผลิตและขนส่งอาหาร, หรือการใช้สิทธิออกเสียง เป็นต้น

## บรรณานุกรม

**Docker คืออะไร ใช้งานอย่างไร.**[เว็บบล็อก]สืบค้นจาก.

<https://medium.com/@rachatatongpagdee/docker-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B8%AD%E0%B8%A2%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B9%84%E0%B8%A3-7e77145967b6>.

**Hyperledger Fabric 1.1 ออกแล้ว รองรับNode.js Chaincode.** [เว็บบล็อก].

สืบค้นจาก <https://www.techtalkthai.com/hyperledger-fabric-1-1-is-released-with-nodejs-chaincode-support/>.

**Hyperledger Fabric.** [เว็บบล็อก][เว็บบล็อก].สืบค้นจาก.

<https://www.investopedia.com/terms/h/hyperledger-fabric.asp> .

**[Day #1] 7 วัน ฉันจะเขียนGolang ให้ได้!**. [เว็บบล็อก]สืบค้นจาก.

<https://medium.com/grean-developers-family/day-1-7-%E0%B8%A7%E0%B8%B1%E0%B8%99-%E0%B8%89%E0%B8%B1%E0%B8%99%E0%B8%88%E0%B8%B0%E0%B9%80%E0%B8%82%E0%B8%B5%E0%B8%A2%E0%B8%99-golang-%E0%B9%83%E0%B8%AB%E0%B9%89%E0%B9%84%E0%B8%94%E0%B9%89-fc2e6ebb3365>.

**มารู้จักกับDockerfile และ Docker Compose ฉบับบ้านๆกันเถอะ.**[เว็บบล็อก].

สืบค้นจาก <https://medium.com/@limjoonae/dockerfile-and-dockercompose-greeting-f910290eeb3a>.

**Concurrency ใน Hyperledger Fabric.**[เว็บบล็อก]สืบค้นจาก..

<https://medium.com/cochain/concurrency-%E0%B9%83%E0%B8%99-hyperledger-fabric-29ff3ef2b7e7>.

**Tutorials.** (2019). [ออนไลน์]. ได้จาก:

<https://hyperledger.github.io/composer/latest/tutorials/tutorials.html?fbclid=IwAR1QAX1JY24L908n0FJDVku6GjPbUjRixOSbmjM3OvfA-ISdT5DQ5Bbnwfs>.

**Hyperledger Composer - Product Auction Network With Events.** [ออนไลน์]. ได้จาก:

<https://github.com/IBM/BlockchainEvents-CompositeJourney>.