สพร. / CDV

ธันวาคม 2564

ห้ามใช้หรือยึดร่างนี้เป็นมาตรฐาน

มาตรฐานฉบับสมบูรณ์จะมีประกาศในราชกิจจานุเบกษา

ร่าง

มาตรฐานรัฐบาลดิจิทัล

Digital Government Standard

ว่าด้วย มาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ด้านการเชื่อมโยงข้อมูล เรื่อง ข้อกำหนดด้านการยืนยันตัวตน การควบคุมสิทธิ์ และบัญชีการใช้งาน

THAILAND GOVERNMENT INFORMATION EXCHANGE STANDARD

SERIES: LINKAGE STANDARD

PART 2 : STANDARD REGULATIONS FOR AUTHENTICATION, AUTHORIZATION AND ACCOUNTING

สำหรับเสนอคณะกรรมการจัดทำร่างมาตรฐาน ข้อกำหนด และหลักเกณฑ์ ภายใต้พระราชบัญญัติการบริหารงานและการให้บริการภาครัฐผ่านระบบดิจิทัล พ.ศ. 2562

สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน)

ชั้น 17 อาคารบางกอกไทยทาวเวอร์ 108 ถนนรางน้ำ แขวงถนนพญาไท เขตราชเทวี กรุงเทพฯ 10400

หมายเลขโทรศัพท์: (+66) 0 2612 6000 โทรสาร: (+66) 0 2612 6011 (+66) 0 2612 6012

คณะกรรมการจัดทำร่างมาตรฐาน ข้อกำหนด และหลักเกณฑ์ ภายใต้พระราชบัญญัติการบริหารงานและการให้บริการภาครัฐผ่านระบบดิจิทัล พ.ศ. 2562

ประธานกรรมการ

ผู้ช่วยศาสตราจารย์ภุชงค์ อุทโยภาศ	มหาวิทยาลัยเกษตรศาสตร์

รองประธานกรรมการ

นายวิบูลย์ ภัทรพิบูล	สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน)

กรรมการ

ผู้ช่วยศาสตราจารย์โษฆศ์รัตต ธรรมบุษดี	มหาวิทยาลัยมหิดล
ผู้ช่วยศาสตราจารย์ณัฐวุฒิ หนูไพโรจน์	จุฬาลงกรณ์มหาวิทยาลัย
นายสุทธิศักดิ์ ตันตะโยธิน	สำนักงานคณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ
นายพณชิต กิตติปัญญางาม	สมาคมการค้าเพื่อส่งเสริมผู้ประกอบการเทคโนโลยีรายใหม่
นายมารุต บูรณรัช	ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ
นางสาวปศิญา เชื้อดี	สำนักงานคณะกรรมการข้อมูลข่าวสารของราชการ
นายศุภโชค จันทรประทิน	สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน)
นางสาวพลอย เจริญสม	
นางบุญยิ่ง ชั่งสัจจา	สำนักบริหารการทะเบียน กรมการปกครอง
นายณัฏฐา พาชัยยุทธ	สำนักงานคณะกรรมการพัฒนาระบบราชการ
นายพัชโรดม ลิมปิษเฐียร	สำนักงานคณะกรรมการกฤษฎีกา
นางสาวพัชรี ไชยเรืองกิตติ	
นางสาวสุภร สุขะตุงคะ	สำนักงานการตรวจเงินแผ่นดิน
นางสาวขนิษฐา ทัศนาพิทักษ์	
นายธีรวุฒิ ธงภักดิ์	สำนักงานคณะกรรมการดิจิทัลเพื่อเศรษฐกิจและสังคมแห่งชาติ
นายกฤษณ์ โกวิทพัฒนา	
นายทรงพล ใหม่สาลี	สำนักงานสถิติแห่งชาติ
นางกาญจนา ภู่มาลี	

กรรมการและเลขานุการ

นางสาวอุรัชฎา เกตุพรหม	สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน)

คณะทำงานเทคนิคด้านมาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ

ที่ปรึกษา

นายสุพจน์ เธียรวุฒิ	สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน)
ผู้ช่วยศาสตราจารย์ภุชงค์ อุทโยภาศ	มหาวิทยาลัยเกษตรศาสตร์
นายวิบูลย์ ภัทรพิบูล	สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน)

ประธานคณะทำงาน

ผู้ช่วยศาสตราจารย์ณัฐวุฒิ หนูไพโรจน์	จุฬาลงกรณ์มหาวิทยาลัย

รองประธานคณะทำงาน

ผู้ช่วยศาสตราจารย์มารอง ผดุงสิทธิ์	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

คณะทำงาน

นายธีรวุฒิ ธงภักดิ์	กระทรวงดิจิทัลเพื่อเศรษฐกิจและสังคม
นายกฤษณ์ โกวิทพัฒนา	
นางสาวนฤมล พันธุ์มาดี	
นายกิติพงษ์ จันทรสกุล	กรมการค้าต่างประเทศ
นายนริศร จินตวรรณ	
ผู้แทนกรมการค้าภายใน	
นางบุญยิ่ง ชั่งสัจจา	กรมการปกครอง
นางสาวมนทิพา เข่งพิมล	กรมพัฒนาธุรกิจการค้า
นายพงศกร ริยะมงคล	
นายกุลเชษฐ์ ชีวะไพบูลย์	
นายกำชัย จัตตานนท์	ผู้แทนกรมศุลกากร
นางสาวขนิษฐา สหเมธาพัฒน์	กรมสรรพากร
ผู้แทนสำนักงบประมาณ	
นายนฤทธิ์ หรั่งทอง	สำนักงานคณะกรรมการส่งเสริมการลงทุน
นางสาวณัฐพร วัฒนสุทธิ	
นายชาวันย์ สวัสดิ์-ชูโต	สำนักงานส่งเสริมวิสาหกิจขนาดกลางและขนาดย่อม
นางสาวณัฏฐา ตุณสุวรรณ	
นางสาวชมบุญ บุญคง	
นางสมศจี ศิกษมัต	ธนาคารแห่งประเทศไทย

นายอาศิส อัญญะโพธิ์ สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน)

คณะทำงานและเลขานุการ

นางสาวอุรัชฎา เกตุพรหม สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน)

นายเจษฎา ขจรฤทธิ์

วิเคราะห์และจัดทำมาตรฐานรัฐบาลดิจิทัล
มาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ
ด้าน การเชื่อมโยงข้อมูล
เรื่อง ข้อกำหนดด้านการยืนยันตัวตน การควบคุมสิทธิ์ และบัญชีการใช้งาน

นายเจษฎา ขจรฤทธิ์	สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน)
นายปราการ ศิริมา	สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน)
นายสุเมธ สุทธิกุล	สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน)

คำนำ

ตามแผนพัฒนารัฐบาลดิจิทัลของประเทศไทยในการผลักดันให้เกิดการเชื่อมโยงข้อมูลของส่วนราชการเข้ากับศูนย์ข้อมูลอื่นๆ รัฐบาลจึงกำหนดให้มีการนำธรรมาภิบาลข้อมูลภาครัฐ (Data Governance: DG) มาเป็นแกนสำคัญในการประยุกต์ใช้ Big Data ภาครัฐเพื่อเพิ่มประสิทธิผลของนโยบายในการพัฒนาประเทศระยะยาว สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน) หรือ สพร. จึงได้สร้างความร่วมมือกับหน่วยงานภาครัฐเพื่อดำเนินการจัดทำมาตรฐานการเชื่อมโยง และแลกเปลี่ยนข้อมูลภาครัฐ (Thailand Government Information Exchange: TGIX) โดยมีจุดประสงค์เพื่อให้เกิดมาตรฐานในการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ อันนำไปสู่การบูรณาการข้อมูล และการใช้ข้อมูลเพื่อขับเคลื่อนประเทศอย่างมีประสิทธิภาพ

มาตรฐานที่ทาง สพร. ดำเนินการจัดทำขึ้นประกอบด้วย 2 ส่วนที่มีความสอดคล้องกัน ได้แก่

(1) มาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ในระดับด้านความหมายข้อมูล (Semantic Standard) และ

(2) มาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ในระดับด้านการเชื่อมโยงข้อมูล (Linkage Standard)

มาตรฐานส่วน (2) เป็นมาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ในระดับด้านการเชื่อมโยงข้อมูล (Linkage Standard) ว่าด้วยเรื่องของสถาปัตยกรรมการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ และองค์ประกอบของสถาปัตยกรรม เช่น (1) การบริหารจัดการ Authentication และ Access Control และ บัญชีผู้ใช้งาน Accounting (2) การบริหารจัดการ Token และ Session (3) โปรโตคอล (Protocol) สำหรับการเชื่อมโยงและแลกเปลี่ยนข้อมูล (4) ความมั่นคงปลอดภัย (Security) และการเข้ารหัสข้อมูล (Encryption) (5) การบันทึกล็อค (Logging) และการติดตาม (Monitoring) (6) การกำหนด namespace ของระบบ เป็นต้น

สารบัญ

สารบัญรูป

มาตรฐานรัฐบาลดิจิทัล

ว่าด้วย มาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ
ด้านการเชื่อมโยงข้อมูล
เรื่อง ข้อกำหนดด้านการยืนยันตัวตน การควบคุมสิทธิ์ และบัญชีการใช้งาน

1. ขอบข่าย

การแลกเปลี่ยนข้อมูลระหว่างหน่วยงานเป็นพื้นฐานหลักที่จำเป็นต่อการพัฒนารัฐบาลดิจิทัล ในปัจจุบันประเทศไทยมีแพลตฟอร์มการแลกเปลี่ยนข้อมูลที่ให้บริการอยู่หลายแห่ง แพลตฟอร์มแต่ละแห่งมี แนวทางและพันธกิจในการดำเนินงานเป็นของตนเอง เป็นผลให้การบูรณาการข้อมูลภาครัฐจำเป็นต้อง ขับเคลื่อนด้วยการสร้างมาตรฐานหรือข้อตกลงร่วมกันในการแลกเปลี่ยนข้อมูล สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน) ได้เล็งเห็นความสำคัญในจุดนี้ จึงมีความจำเป็นต้องจัดทำมาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ เพื่อใช้ในการแลกเปลี่ยนข้อมูลระหว่างหน่วยงานของรัฐเพื่อให้เกิดการบูรณาการข้อมูลเกิดขึ้นอย่างเป็นรูปธรรม

เป้าประสงค์หลักของการใช้มาตรฐานฯ เป็นตัวขับเคลื่อนการบูรณาการข้อมูลภาครัฐ คือ การให้หน่วยงานของรัฐมีแนวทางในการพัฒนาสถาปัตยกรรมระบบสารสนเทศเพื่อใช้ในการแลกเปลี่ยนข้อมูลที่ชัดเจน มีความสอดคล้องในการเชื่อมต่อระหว่างกัน

ดังนั้น เพื่อให้บรรลุเป้าประสงค์หลักดังกล่าว เอกสารฉบับนี้จึงนำเสนอข้อกำหนดด้านการยืนยันตัวตน การควบคุมสิทธิ์ และบัญชีการใช้งาน สำหรับประกอบเอกสารว่าด้วยมาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ เรื่องมาตรฐานสถาปัตยกรรมการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐระดับการเชื่อมโยงข้อมูลที่เหมาะสมกับบริบทของประเทศไทยเท่านั้น

2. นิยาม

นิยามคำศัพท์ที่เกี่ยวข้องกับมาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ด้านการเชื่อมโยงข้อมูล เรื่อง ข้อกำหนดด้านการยืนยันตัวตน การควบคุมสิทธิ์ และบัญชีการใช้งานที่ใช้ในเอกสารฉบับนี้มีดังนี้

2.1 การยืนยันตัวตน (Authentication) หมายความว่า กระบวนการที่ผู้ใช้บริการ API ยืนยันตัวตนกับ ผู้พิสูจน์และยืนยันตัวตนว่าเป็นเจ้าของไอเดนทิตีที่กล่าวอ้างด้วยการใช้สิ่งที่ใช้ยืนยันตัวตน

2.2 ผู้ให้บริการ API (Provider System) หมายความว่า ระบบสารสนเทศของหน่วยงานที่เปิดให้บริการ API สำหรับเชื่อมโยงและการแลกเปลี่ยนข้อมูลภายในแพลตฟอร์มที่ใช้มาตรฐาน TGIX

2.3 ผู้ใช้บริการ API (Consumer System) หมายความว่า ระบบสารสนเทศของหน่วยงานมีการใช้บริการ API สำหรับเชื่อมโยงและการแลกเปลี่ยนข้อมูลภายในแพลตฟอร์มที่ใช้มาตรฐาน TGIX

2.4 ผู้ให้บริการแพลตฟอร์ม TGIX (TGIX Platform Provider) หมายความว่า ระบบสารสนเทศของหน่วยงานผู้ให้บริการ TGIX Platform เพื่อสนับสนุนดำเนินการเชื่อมโยงและการแลกเปลี่ยนข้อมูลให้เป็นไปตามมาตรฐาน TGIX

2.5 การยืนยันตัวตนด้วยวิธีการใช้ API Key (API Key Authentication Method) หมายความว่า กระบวนการยืนยันตัวตนด้วยวิธีการใช้ API Key ซึ่งเป็นค่าที่สร้างขึ้นแบบไม่ซ้ำกันโดยผู้ให้บริการ API (Provider System) แล้วส่งมอบให้ผู้ใช้บริการ API (Consumer System) เก็บไว้ใช้ในการยืนยันตัวตนระหว่างเรียกใช้งานบริการแบบ REST API ของผู้ให้บริการ API (Provider System)

2.6 การยืนยันตัวตนด้วยมาตรฐาน OAuth 2.0 หมายความว่า กระบวนการยืนยันตัวตนด้วยวิธีการใช้ OAuth 2.0 ซึ่งผู้ใช้บริการ API (Consumer System) ดำเนินการยืนยันตัวตนก่อนรับบริการ API จากผู้ให้บริการ API (Provider System) ตามมาตรฐาน TGIX

2.7 การยืนยันตัวตนด้วยมาตรฐาน Open ID Connect หมายความว่า กระบวนการยืนยันตัวตนด้วยด้วยวิธีการใช้ Open ID Connect ซึ่งผู้ใช้บริการ API (Consumer System) ดำเนินการยืนยันตัวตนก่อนรับบริการ API จากผู้ให้บริการ API (Provider System) ตามมาตรฐาน TGIX

2.8 ประเภทการให้สิทธิ์ (Grant Type) หมายความว่า ประเภทการให้สิทธิ์ระหว่างผู้ใช้บริการ API (Consumer System) และผู้พิสูจน์และยืนยันตัวตน (Identity Provider) เมื่อมีการเลือกใช้มาตรฐาน OAuth 2.0 ในการยืนยันตัวตนก่อนเรียกใช้บริการ API ไปยังผู้ให้บริการ API (Provider System)

2.9 การควบคุมสิทธิ์ในการเข้าถึง API (API Access Control) หมายความว่า การควบคุมสิทธิ์ในการเข้าถึง API ด้วยวิธีการควบคุมสิทธิ์และการตรวจสอบสิทธิ์ตามบทบาทและหน้าที่ของผู้ใช้งาน API อ้างอิงจาก INCITS 359-2012[R2017]

2.10 การควบคุมสิทธิ์การเข้าถึงด้วยวิธี Role-Based Access Control (Role-Based Access Control) หมายความว่า กระบวนการในการควบคุมสิทธิ์การเข้าถึง API ของผู้ให้บริการ API (Provider System) ในการเข้าถึง API

2.11 การบริหารจัดการบัญชีผู้ใช้งาน API (API User Account) หมายความว่า ระเบียบปฏิบัติในการที่ผู้ใช้บริการ API (Consumer System) ใช้สำหรับยืนยันตัวตนเพื่อใช้บริการ API ของผู้ให้บริการ API (Provider System) แบ่งประเภทบัญชีได้ตามประเภทการยืนยันตัวตนได้ 3 ประเภทบัญชี คือ บัญชีใช้งานประเภท API Key ใช้สำหรับการยืนยันตัวตนด้วย API Key บัญชีใช้งาน Identity Provider ที่รองรับมาตรฐาน OAuth 2.0 และบัญชีใช้งาน Identity Provider ที่รองรับมาตรฐาน Open ID Connect

2.12 ผู้พิสูจน์และยืนยันตัวตน (Identity Provider) หมายความว่า ระบบสารสนเทศของหน่วยงานหรือของผู้ให้บริการแพลตฟอร์ม TGIX สำหรับผู้รับลงทะเบียนและผู้พิสูจน์ตัวตน และการบริหารจัดการสิ่งที่ใช้รับรองตัวตน ซึ่งเชื่อมโยงไอเดนทิตีเข้ากับสิ่งที่ใช้ยืนยันตัวตนของผู้ใช้บริการโดยผู้พิสูจน์และยืนยันตัวตนอาจบริหารจัดการสิ่งที่ใช้รับรองตัวตนเพื่อใช้ภายในองค์กรหรือใช้ภายนอกองค์กรก็ได้

3. กฎหมายและแนวปฏิบัติที่เกี่ยวข้อง

การเชื่อมโยงและการแลกเปลี่ยนข้อมูลภาครัฐมีการบัญญัติไว้ในกฎหมายหรือแนวปฏิบัติที่เกี่ยวข้อง ดังนี้

3.1 รัฐธรรมนูญแห่งราชอาณาจักรไทย พ.ศ. 2560 ในมาตรา 59 ระบุว่า รัฐต้องเปิดเผยข้อมูลหรือข่าวสารสาธารณะในครอบครองของหน่วยงานของรัฐที่มิใช่ข้อมูลเกี่ยวกับความมั่นคงของรัฐหรือเป็นความลับของทางราชการตามที่กฎหมายบัญญัติ และต้องจัดให้ประชาชนเข้าถึงข้อมูลหรือข่าวสารดังกล่าวได้โดยสะดวก

3.2 พระราชบัญญัติการบริหารงานและการให้บริการภาครัฐผ่านระบบดิจิทัล พ.ศ. 2562 ในมาตรา 13 ระบุว่าเพื่อประโยชน์ในการบริหารราชการแผ่นดินและการให้บริการประชาชน ให้หน่วยงานของรัฐจัดให้มีการเชื่อมโยงและแลกเปลี่ยนข้อมูลดิจิทัลที่มีการจัดทำและครอบครองตามที่หน่วยงานของรัฐแห่งอื่นร้องขอ ที่จะเกิดการบูรณาการร่วมกัน

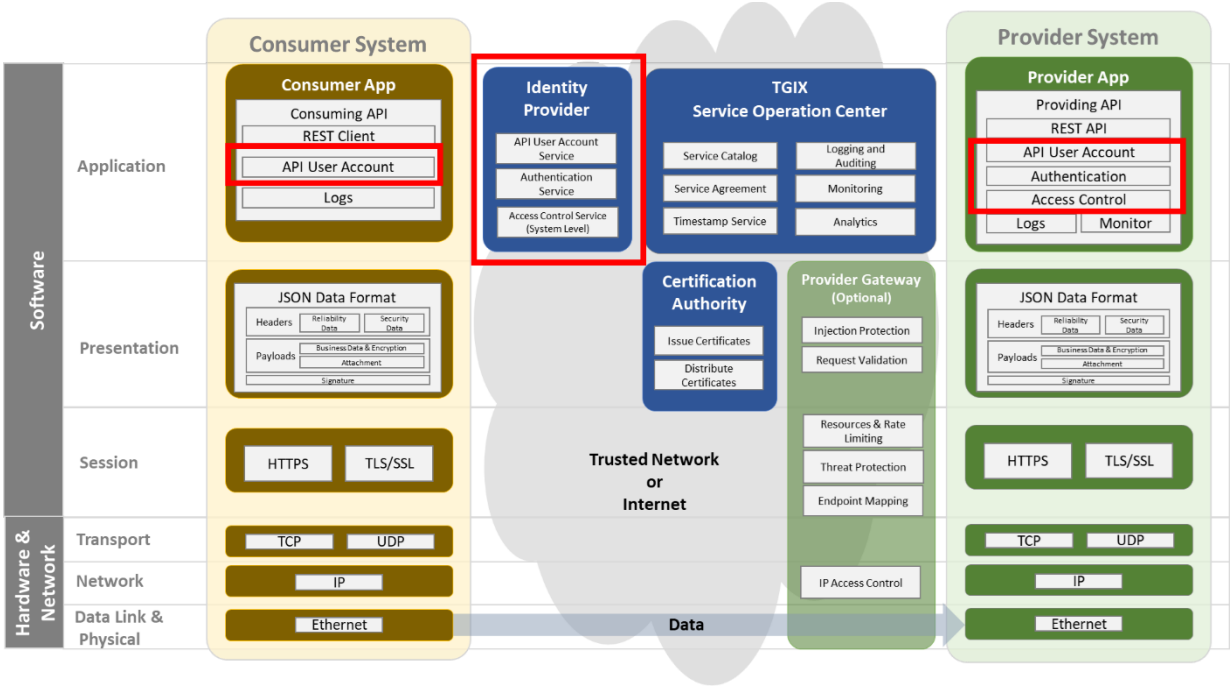
มาตรา 15 ระบุว่า ให้มีศูนย์แลกเปลี่ยนข้อมูลกลางทำหน้าที่เป็นศูนย์กลางในการแลกเปลี่ยนข้อมูลดิจิทัลและทะเบียนดิจิทัลระหว่างหน่วยงานของรัฐ เพื่อสนับสนุนการดำเนินการของหน่วยงานของรัฐในการให้บริการประชาชนผ่านระบบดิจิทัล และดำเนินการในเรื่องดังต่อไปนี้

(1) กำหนดนโยบายและมาตรฐานเกี่ยวกับการเชื่อมโยงและแลกเปลี่ยนข้อมูลเสนอต่อคณะกรรมการพัฒนารัฐบาลดิจิทัลให้ความเห็นชอบ
(2) ประสานและให้ความช่วยเหลือแก่หน่วยงานของรัฐในการเชื่อมโยงและแลกเปลี่ยนข้อมูลดิจิทัลระหว่างกัน รวมทั้งกำกับติดตามให้การดำเนินการดังกล่าวเป็นไปในแนวทางและมาตรฐานเดียวกันตามหลักเกณฑ์ วิธีการ และเงื่อนไขที่คณะกรรมการพัฒนารัฐบาลดิจิทัลกำหนด
(3) จัดทำคำอธิบายชุดข้อมูลดิจิทัลของภาครัฐ และจัดเก็บบันทึกหลักฐานของการเชื่อมโยงและแลกเปลี่ยนข้อมูลดิจิทัล
(4) เรื่องอื่นๆ ตามที่คณะกรรมการพัฒนารัฐบาลดิจิทัลมอบหมาย

มาตรา 19 ระบุว่า ในวาระเริ่มแรก ให้สำนักงานดำเนินการให้มีศูนย์แลกเปลี่ยนข้อมูลกลางตามมาตรา 15 เป็นการชั่วคราวแต่ไม่เกินสองปี เมื่อครบกำหนดระยะเวลาดังกล่าว ให้คณะกรรมการพัฒนารัฐบาลดิจิทัลพิจารณาความจำเป็นและเหมาะสมเกี่ยวกับหน่วยงานของรัฐที่จะมาดำเนินการเกี่ยวกับศูนย์แลกเปลี่ยนข้อมูลกลาง ทั้งนี้ ในกรณีที่คณะกรรมการพัฒนารัฐบาลดิจิทัลเห็นควรให้หน่วยงานของรัฐแห่งอื่นใดทำหน้าที่แทนสำนักงาน ให้เสนอแนวทางการดำเนินการ การโอนภารกิจ งบประมาณ ทรัพย์สินและหนี้สิน ภาระผูกพัน และบุคลากรไปยังหน่วยงานของรัฐแห่งอื่นนั้นต่อคณะรัฐมนตรีเพื่อพิจารณา

4. ข้อกำหนดด้านการยืนยันตัวตน การควบคุมสิทธิ์ และบัญชีการใช้งาน

การยืนยันตัวตน (Authentication) การควบคุมสิทธิ์ (Access Control) และบัญชีการใช้งาน (Accounting) เป็นองค์ประกอบสำคัญของสถาปัตยกรรมการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ดังรูปที่ 1 องค์ประกอบเหล่านี้ช่วยให้ผู้ให้บริการ API (Provider System) และผู้ใช้บริการ API (Consumer System) เชื่อมโยงและแลกเปลี่ยนข้อมูลได้อย่างปลอดภัย



รูปที่ 1 ภาพรวมองค์ประกอบของการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ

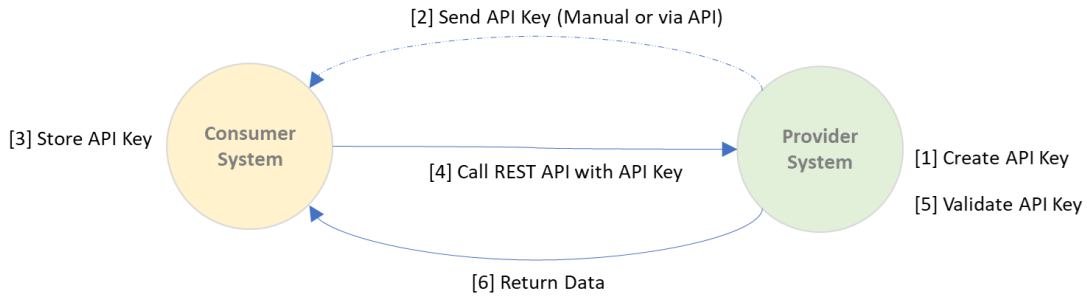
4.1. การยืนยันตัวตน (Authentication)

การยืนยันตัวตนเพื่อขอใช้บริการการแลกเปลี่ยนข้อมูล หมายถึง กระบวนการที่ผู้ใช้บริการ API (Consumer System) ทำการยืนยันตัวตนเพื่อขอใช้บริการ API ของผู้ให้บริการ API (Provider System) ที่เป็น REST API ซึ่งแนวทางในการยืนยันตัวตน สามารถแบ่งได้เป็น 3 วิธีคือ API Key, OAuth 2.0 และ Open ID Connect

4.2. การยืนยันตัวตนด้วย API Key

API Key ใช้เพื่อยืนยันว่าผู้ใช้บริการ API (Consumer System) ต้องการขอเข้าถึง API แบบ REST API ของผู้ให้บริการ API (Provider System) แต่ไม่ได้ต้องการการยืนยันตัวระดับตัวบุคคลที่ใช้งานในระบบของผู้ใช้บริการ API (Consumer System) ดังนั้น ในด้านความปลอดภัยจะเพียงพอสำหรับการเข้าถึง API ที่เป็นบริการ API ทั่วไปในหน่วยงานของผู้ให้บริการ API (Provider System) โดยข้อมูลเหล่านั้นสามารถเข้าถึงด้วย API โดยที่ไม่ต้องยืนยันตัวตนระดับบุคคล

ในด้านเทคนิคนั้น API Key เป็นค่าที่สร้างขึ้นแบบไม่ซ้ำกันโดยผู้ให้บริการ API (Provider System) แล้วส่งมอบให้ผู้ใช้บริการ API (Consumer System) เก็บไว้ใช้ในการยืนยันตัวตนระหว่างเรียกใช้งาน REST API ของผู้ให้บริการ API (Provider System) ดังรูปที่ 2



รูปที่ 2 ขั้นตอนการดำเนินการเพื่อยืนยันตัวตนด้วย API Key

ผู้ให้บริการ API (Provider System) และผู้ใช้บริการ API (Consumer System) มีขั้นตอนการดำเนินการดังต่อไปนี้

4.2.1 ขั้นตอนที่ 1: การสร้าง API Key (Create API Key)

ในขั้นตอนนี้ ผู้ให้บริการ API (Provider System) ต้องดำเนินการสร้าง API Key ดังรูปที่ 2 ซึ่งในแต่ละ API นั้น ผู้ให้บริการ API (Provider System) ต้องดำเนินการสร้าง API Key ให้มีค่าไม่ซ้ำกัน โดยวิธีการที่แนะนำคือการสุ่มด้วยวิธีการ Secure Random จากภาษาโปรแกรมที่ใช้พัฒนาระบบ เช่น โปรแกรมภาษา Java ดังนี้

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import org.apache.commons.codec.binary.Base64; public class SampleGenerateAPIKey {
    public static void main(String[] args) {
        try {
            SampleGenerateAPIKey s = new SampleGenerateAPIKey();
            String prefix = SampleGenerateAPIKey.getSecureRandom(32).substring(0,7),key =
SampleGenerateAPIKey.getSecureRandom(32);
            System.out.println("API key: " + s.genAPIKey(prefix, key));
        } catch (NoSuchAlgorithmException e) {
        }
    }

    public static String filter(String source) {
        return source.replaceAll("/", "").replaceAll("\\+", "").replaceAll("=", "").replaceAll("\\\\", "");
    }

    public static String getSecureRandom(int bytesLength) throws NoSuchAlgorithmException{
        byte[] sRandomBytes = new byte[bytesLength];
        SecureRandom.getInstance("SHA1PRNG").nextBytes(sRandomBytes);
```

```
        return filter((new Base64()).encodeToString(sRandomBytes));
    }

    public String genAPIKey(String prefix, String key) throws NoSuchAlgorithmException{
        return prefix + "." + filter(
                (new Base64())
                        .encodeToString(MessageDigest
                                .getInstance("SHA-256")
                                .digest((prefix + key)
                                        .getBytes(StandardCharsets.UTF_8))));
    }
}
```

ผลลัพธ์ที่ได้จากตัวอย่างข้างต้นคือ API Key ที่มีค่าดังนี้

API Key : Lhyz7fW.0MFHlBmWWVhoLZWSmNXBW8lugbOwkTtHy76BEQ ซึ่งประกอบด้วย Prefix คือ Lhyz7fW และ Key คือ 0MFHlBmWWVhoLZWSmNXBW8lugbOwkTtHy76BEQ

หลังจากได้ค่า API Key แล้ว ผู้ให้บริการ API (Provider System) ควรเก็บรักษา API Key ไว้ในที่ปลอดภัย เช่น เก็บไว้ในฐานข้อมูลโดยทำการใส่ Prefix และ Hash ค่าของ API Key ด้วยภาษาโปรแกรมที่ใช้พัฒนาระบบ ดังนี้

```
API Key : {prefix}.{hash_of_whole_api_key}
```

นอกจากนี้ ผู้ให้บริการ API (Provider System) ควรระบุได้ว่ามีการส่งมอบ API Key ให้กับผู้ใช้บริการ API (Consumer System) ใดแล้วบ้าง พร้อมทั้งมีการกำหนดวันหมดอายุของ API Key สามารถกำหนดค่าตั้งต้นให้ไม่มีวันหมดอายุ แต่ควรสามารถปรับเปลี่ยนให้มีวันหมดอายุตามความเหมาะสมของ API ได้ นอกจากนี้ ควรสร้าง API Key ได้ใหม่เมื่อมีการร้องขอจากผู้ใช้บริการ API (Consumer System)

4.2.2 ขั้นตอนที่ 2: การส่งมอบ API Key (Send API Key)

ในขั้นตอนนี้ ผู้ให้บริการ API (Provider System) ต้องดำเนินการส่งมอบ API Key ให้กับผู้ใช้บริการ API (Consumer System) ดังรูปที่ 2 ซึ่งการส่งมอบ API Key นั้น เกิดขึ้นหลังจากผู้ให้บริการ API (Provider System) ทำข้อตกลงเพื่อใช้บริการ API (Service Agreement) กับผู้ใช้บริการ API (Consumer System) ที่ TGIX Service Operation Center ซึ่งดูแลโดยหน่วยงานผู้บริการ TGIX Platform เรียบร้อยแล้ว โดยมีรายละเอียดตามมาตรฐานสถาปัตยกรรมการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐระดับการเชื่อมโยงข้อมูล ซึ่งผู้ให้บริการ API (Provider System) สามารถเลือกดำเนินการส่ง API Key ผ่านช่องทางต่างๆ ได้ตามความเหมาะสมและงบประมาณของหน่วยงาน เช่น

- ส่งผ่านอีเมล

- สร้าง API สำหรับส่ง API Key
- พัฒนาหน้าจอสำหรับให้ผู้ใช้บริการ API (Consumer System) มารับ API Key

4.2.3 ขั้นตอนที่ 3: การเก็บรักษา API Key (Store API Key)

ในขั้นตอนนี้ ผู้ใช้บริการ API (Consumer System) ต้องดำเนินการเก็บรักษา API Key ดังรูปที่ 2 ซึ่งหลังจากได้รับมอบ API Key แล้วนั้น ผู้ใช้บริการ API (Consumer System) ควรเก็บรักษา API Key ไว้ในที่ปลอดภัย ไม่ควรกำหนด API Key ไว้ใน Source Code ซึ่งอาจจะเกิดความผิดพลาดขณะแชร์ Source Code ให้กับบุคคลอื่นได้ ผู้ใช้บริการ API (Consumer System) ควรเก็บ API Key ไว้ใน Environment Variable หรือ File หรือที่อื่นๆ ที่ไม่อยู่ใน Source Code หลัก รวมทั้งทำการ Hash ของข้อมูล API Key ก่อนเก็บเสมอ

4.2.4 ขั้นตอนที่ 4: การยืนยันตัวตนและเรียก API ด้วย API Key (Call REST API with API Key)

ในขั้นตอนนี้ ผู้ใช้บริการ API (Consumer System) ต้องดำเนินการการยืนยันตัวตนเพื่อเรียกใช้บริการ REST API ด้วย API Key ดังรูปที่ 2 ซึ่งเกิดขึ้นเมื่อได้รับ API Key แล้วผู้ใช้บริการ API (Consumer System) จะต้องดำเนินการส่งข้อมูล API Key เพื่อยืนยันตัวตนระหว่างเรียกใช้บริการ REST API ไปยังผู้ให้บริการ API (Provider System) โดยผู้ใช้บริการ API (Consumer System) สามารถดำเนินการตามที่ได้ตกลงไว้กับผู้ให้บริการ API (Provider System) จากวิธีต่อไปนี้

(1) ส่งข้อมูล API Key เป็นส่วนหนึ่งของ Authorization Header ขณะเรียกใช้ REST API ตัวอย่างเช่น

```
Authorization: Apikey 1234567890abcdef
```

(2) ส่งข้อมูล API Key เป็นส่วนหนึ่งของ Basic Authentication ของ REST API ตัวอย่าง Curl Command เช่น

```
curl -X GET \
  'https://provider_server/endpoint/' \
-H 'authorization: Basic 1234567890abcdef '
```

(3) ส่งข้อมูล API Key เป็นส่วนหนึ่งของ Body Data ขณะเรียกใช้ REST API ตัวอย่างเช่น

```
curl -X POST \
  "https://provider_server/endpoint/" \
  -H 'content-type: application/json' \
  -d '{
          "api_key": "1234567890abcdef "
```

```
        }'
```

(4) ส่งข้อมูล API Key เป็นส่วนหนึ่งของ Query String ขณะเรียกใช้ REST API ตัวอย่างเช่น

```
curl -X GET "https://provider_server/api_endpoint/?api_key=1234567890abcdef '
```

4.2.5 ขั้นตอนที่ 5: การตรวจสอบความถูกต้องของ API Key (Validate API Key)

ในขั้นตอนนี้ ผู้ให้บริการ API (Provider System) ต้องดำเนินการตรวจสอบความถูกต้องของ API Key ดังรูปที่ 2 ซึ่งเกิดขึ้นเมื่อผู้ให้บริการ API (Provider System) ได้รับการขอใช้บริการ API พร้อมด้วย API Key หลังจากนั้น ผู้ให้บริการ API (Provider System) ต้องดำเนินการตรวจสอบความถูกต้องแล้วดำเนินการให้บริการ API ตามข้อมูลที่ถูกร้องขอ หรือปฏิเสธการให้บริการหาก API Key ไม่ถูกต้อง ซึ่งขั้นตอนนี้ผู้ให้บริการ API สามารถดำเนินการได้ตามความเหมาะสมของภาษาโปรแกรมที่ใช้พัฒนาระบบ

4.2.6 ขั้นตอนที่ 6: การตอบกลับผลการให้บริการ API (Return Data)

ในขั้นตอนนี้ ผู้ให้บริการ API (Provider System) ต้องดำเนินการตอบกลับผลการให้บริการ API ดังรูปที่ 2 ซึ่งเกิดขึ้นเมื่อผู้ให้บริการ API (Provider System) ตรวจสอบความถูกต้องของ API Key แล้วดำเนินการให้บริการ API ตามข้อมูลที่ถูกร้องขอสำเร็จ ควรตอบกลับด้วย HTTP Code 200 ตามตัวอย่างนี้ หรือ HTTP Code อื่นๆ ตามความเหมาะสม

```
        “messageStatus": {
         “status": “200",
         “description": " REST API successfully carried out the client requested"
}
```

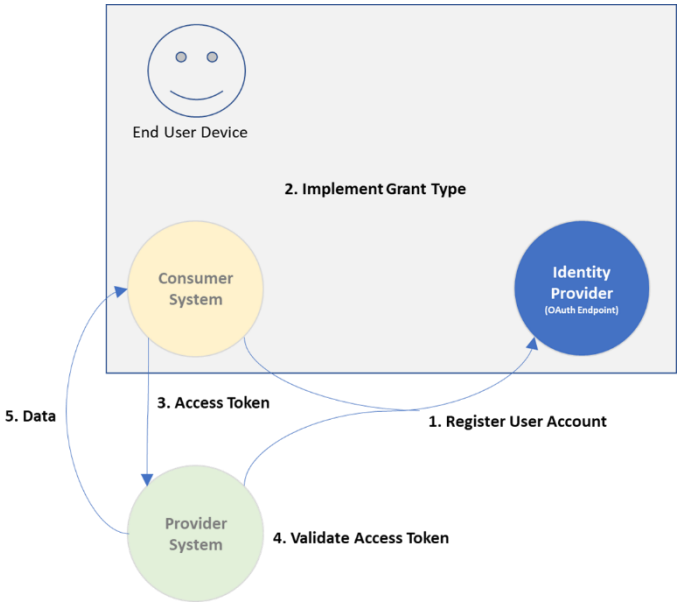
กรณีต้องการปฏิเสธการให้บริการ เนื่องจาก API Key ไม่ถูกต้อง ควรตอบกลับด้วย HTTP Code 401 ตามตัวอย่างต่อไปนี้

```
“messageStatus": {
         “status": “401",
         “description": " Unauthorized - ApiKey invalid or ApiKey not found"
}
```

4.3. การยืนยันตัวตนด้วยมาตรฐาน OAuth 2.0

OAuth 2.0 เป็นการรวมกระบวนการยืนยันตัวตนและจัดการสิทธิ์ให้เข้าถึงข้อมูลเข้าด้วยกัน ตามมาตรฐาน The OAuth 2.0 Authorization Framework: Bearer Token Usage: RFC-6749 [1] , RFC-6750 [2] มาตรฐาน

OAuth 2.0 สามารถใช้ยืนยันตัวตนระดับผู้ใช้งานระบบได้ ดังนั้นจึงเหมาะสมในการเข้าถึง API ที่เป็นบริการเข้าถึงข้อมูลส่วนบุคคลหรือข้อมูลสำคัญของผู้ให้บริการ API (Provider System) รวมทั้ง API เชิงธุรกรรมประเภทที่เป็นการสร้าง ลบหรือแก้ไขข้อมูล โดยหลักการของ OAuth 2.0 จะเป็นการยืนยันตัวตนผู้ใช้งานในระบบ (End User) ของผู้ใช้บริการ API (Consumer System) กับระบบพิสูจน์และยืนยันตัวตน (Identity Provider) ที่รองรับ OAuth 2.0 เพื่อให้ได้ Access Token ซึ่งกระบวนการดังกล่าวเรียกว่าการยืนยันตัวตนและให้สิทธิ์ (Grant Type) หลังจาก ผู้ใช้บริการ API (Consumer System) ได้รับ Access Token แล้วจะนำมาใช้ในการเข้าถึง API ของผู้ให้บริการ API (Provider System) ขั้นตอนการดำเนินการมีดังนี้



รูปที่ 3 ภาพรวมของการยืนยันตัวตนด้วย OAuth 2.0

4.3.1 ขั้นตอนที่ 1 การลงทะเบียนบัญชีผู้ใช้งาน (Register User Account)

ในขั้นตอนนี้ ผู้ให้บริการ API (Provider System) และผู้ใช้บริการ API (Consumer System) ดำเนินการแจ้งความประสงค์ขอลงทะเบียนบัญชีใช้งานที่ผู้พิสูจน์และยืนยันตัวตน (Identity Provider) ซึ่งดูแลโดยหน่วยงานผู้บริการ TGIX Platform ดังรูปที่ 3 ขั้นตอนนี้จะเกิดขึ้นหลังจากที่ผู้ให้บริการ API (Provider System) และผู้ใช้บริการ API (Consumer System) ทำข้อตกลงเพื่อใช้บริการ API (Service Agreement) ไว้ที่ TGIX Service Operation Center เรียบร้อยแล้ว

วิธีดำเนินการในขั้นตอนนี้ขึ้นอยู่กับ Identity Provider ที่สมาชิกในกลุ่ม TGIX และหน่วยงานผู้บริการ TGIX Platform ตกลงกันเลือกใช้ดำเนินการเพื่อบริหารจัดการบัญชีผู้ใช้งาน

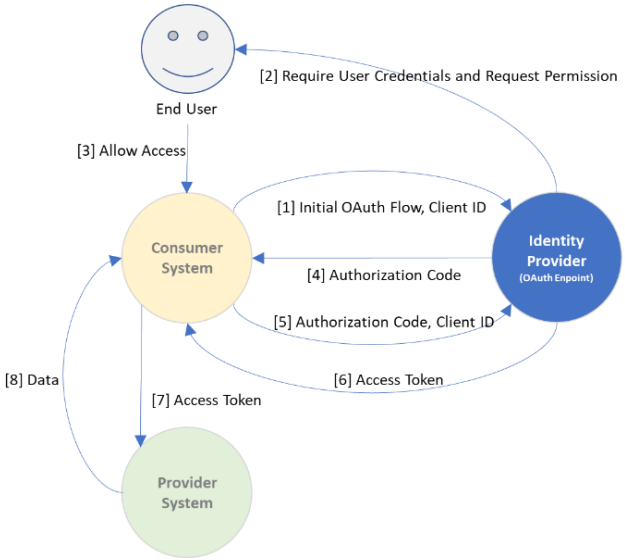
4.3.2 ขั้นตอนที่ 2: การยืนยันตัวตนเพื่อให้ได้ Access Token (Implement Grant Type)

ขั้นตอนนี้ ผู้ใช้บริการ API (Consumer System) ต้องดำเนินการยืนยันตัวตนกับผู้พิสูจน์และยืนยันตัวตน (Identity Provider) เพื่อให้ได้ Access Token ดังกรอบสีเทาในรูปที่ 3 ซึ่งในมาตรฐาน OAuth 2.0 มีประเภท

การยืนยันตัวตนและให้สิทธิ์ (Grant Type) ซึ่งผู้ใช้บริการ API (Consumer System) สามารถเลือกดำเนินการได้ตามความเหมาะสมของภาษาโปรแกรมที่ใช้พัฒนาและงบประมาณที่มี โดยเลือกได้จาก 4 ประเภท ได้แก่

(1) การยืนยันตัวตนและให้สิทธิ์ (Grant Type) ประเภท Authorization Code

Authorization Code เป็น Grant type ประเภทที่ผู้ใช้บริการ API (Consumer System) ยืนยันตัวตนกับผู้พิสูจน์และยืนยันตัวตน (Identity Provider) โดยนำ Authorization Code มาแลกเปลี่ยนเป็น Access Token

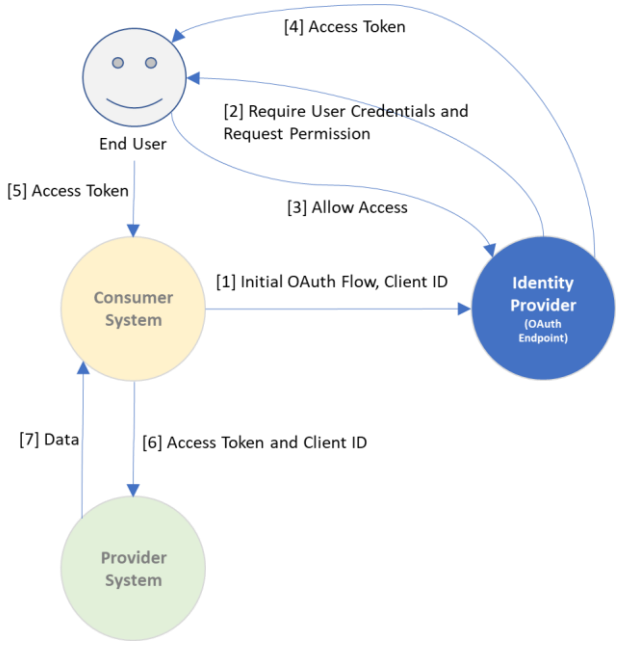


รูปที่ 4 ขั้นตอนการยืนยันตัวตนด้วย Grant Type ประเภท Authorization Code

จากลำดับที่ [1] – [6] ในรูปที่ 4 นั้นผู้ใช้บริการ API (Consumer System) สามารถเลือกดำเนิน การเพื่อให้ได้ Access Token ตามขั้นตอนในเอกสาร The OAuth 2.0 Authorization Framework (RFC-6749): Section-1.3.1 [3]

(2) การยืนยันตัวตนและให้สิทธิ์ (Grant Type) ประเภท Implicit

Implicit จะมีความคล้ายกับแบบ Authorization Code แต่ต่างกันที่ผู้ใช้บริการ API (Consumer System) ไม่ต้องดำเนินการส่ง Authorization Code แล้วไปขอ Access Token อีกที แต่จะได้ Access Token กลับมาผ่านทาง Query String จากผู้พิสูจน์และยืนยันตัวตน (Identity Provider) ในคราวเดียว

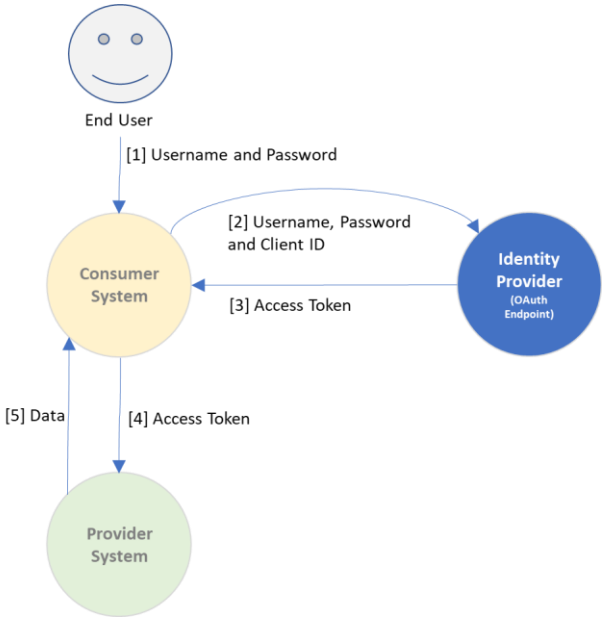


รูปที่ 5 ขั้นตอนการยืนยันตัวตนด้วย Grant Type ประเภท Implicit

จากลำดับที่ [1] – [5] ในรูปที่ 5 นั้นผู้ใช้บริการ API (Consumer System) สามารถเลือกดำเนิน การเพื่อให้ได้ Access Token ตามขั้นตอนในเอกสาร The OAuth 2.0 Authorization Framework (RFC-6749): Section-1.3.2 [4]

(3) การยืนยันตัวตนและให้สิทธิ์ (Grant Type) ประเภท Resource Owner Password

เป็นการยืนยันตัวตนและขอสิทธิ์โดย ผู้ใช้งานระบบของผู้ใช้บริการ API (Consumer System) จะให้ Username และ Password กับผู้ใช้บริการ API (Consumer System) โดยตรง เพื่อนำไปขอ Access Token จากผู้พิสูจน์และยืนยันตัวตน (Identity Provider) ดังนี้

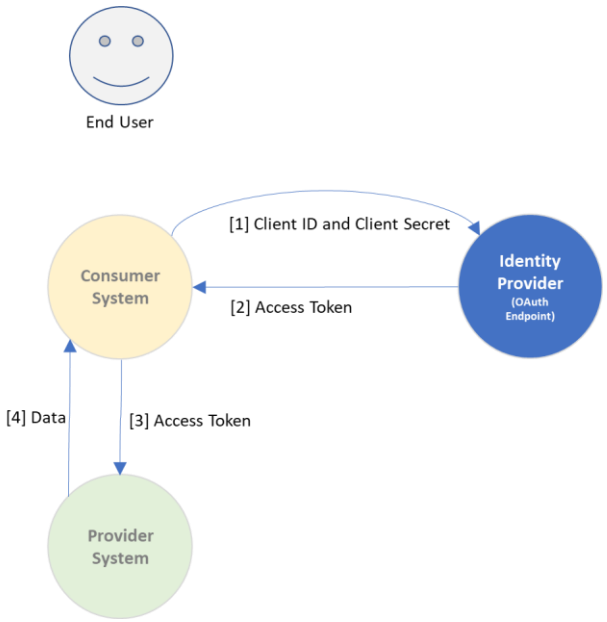


รูปที่ 6 ขั้นตอนการยืนยันตัวตนด้วย Grant Type ประเภท Resource owner password

จากลำดับที่ [1] – [3] ในรูปที่ 6 นั้นผู้ใช้บริการ API (Consumer System) สามารถเลือกดำเนิน การเพื่อให้ได้ Access Token ตามขั้นตอนในเอกสาร The OAuth 2.0 Authorization Framework (RFC-6749): Section-1.3.3 [5]

(4) การยืนยันตัวตนและให้สิทธิ์ (Grant Type) ประเภท Client Credentials

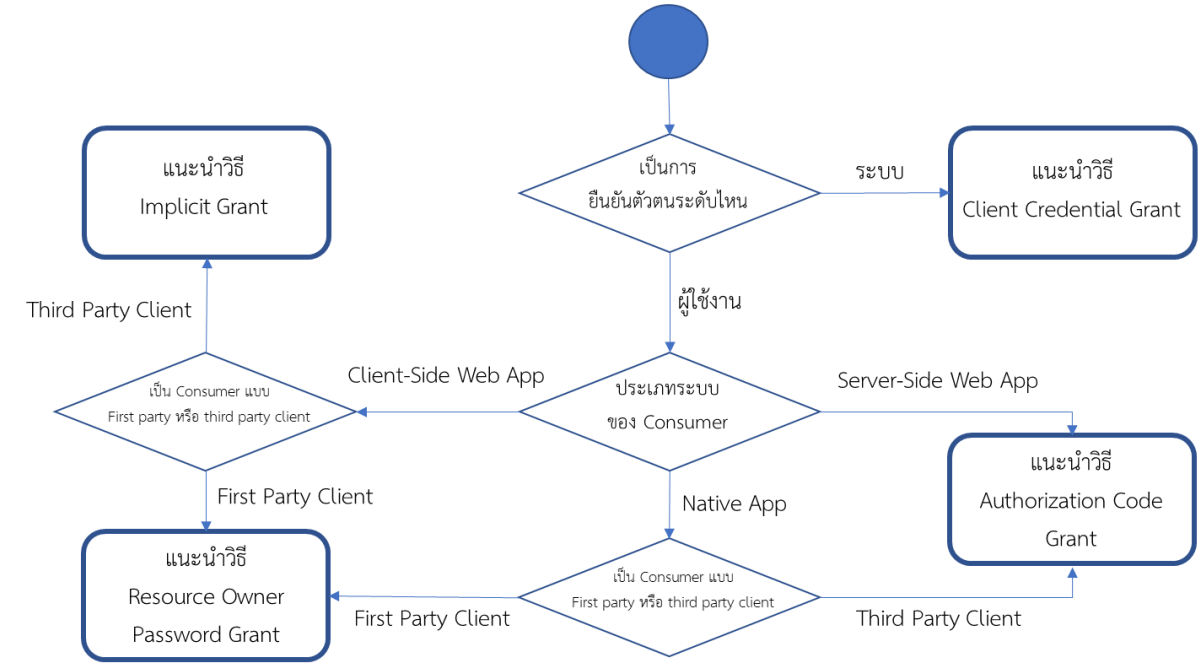
เป็นการยืนยันตัวตนและขอสิทธิ์โดยผู้ใช้บริการ API (Consumer System) จะใช้ Client ID และ Client Secret ในการส่งไปขอ Access Token ที่ผู้พิสูจน์และยืนยันตัวตน (Identity Provider) โดยจะเป็นการขอระหว่าง Server ไปยัง Server โดยตรง



รูปที่ 7 ขั้นตอนการยืนยันตัวตนด้วย Grant Type ประเภท Client Credentials

จากลำดับที่ [1] – [2] ในรูปที่ 7 นั้นผู้ใช้บริการ API (Consumer System) สามารถเลือกดำเนินการเพื่อให้ได้ Access Token ตามขั้นตอนในเอกสาร The OAuth 2.0 Authorization Framework (RFC-6749): Section-1.3.4 [6]

แนวทางในการเลือกประเภท Grant Type ขึ้นอยู่กับประเภทข้อมูลต่างๆ ของผู้ใช้บริการ API (Consumer System) ได้แก่ ผู้ถือข้อมูลการยืนยันตัวตน และลักษณะ Application ของผู้ใช้บริการ API (Consumer System) เป็น Web Application หรือ Native Application ดังรูปที่ 8



First Party Client คือระบบ Consumer ที่มีความน่าเชื่อถือเพียงพอให้ผู้ใช้งานกรอกข้อมูล Password จาก IDP ที่ระบบ
Third Party Client คือระบบ Consumer ที่จะ Redirect ไปให้ผู้ใช้งานกรอกข้อมูล Password ที่ IDP ซึ่งเป็นวิธีที่นิยมอย่างแพร่หลาย

รูปที่ 8 แนวทางการเลือกดำเนินการ Grant Type

เมื่อผู้ใช้บริการ API (Consumer System) ดำเนินการเสร็จเรียบร้อย ผลลัพธ์ที่ผู้ใช้บริการ API (Consumer System) ได้รับคือ Access Token ที่ได้รับการเข้ารหัสด้วย JSON Web Tokens (JWT) เพื่อใช้ส่งให้กับผู้ให้บริการ API (Provider System) ในขณะเรียกใช้บริการ REST API ของผู้ให้บริการ API (Provider System) ตัวอย่าง Access Token ดังนี้

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
{
  "access_token":"
eyJhbGciOiJIUzIINilsInR5cCI6IkpXVCJ9.eyJpc3MiOiodHRwOi8vbXktZG9tYWluLmF1dGgwLmNvSl
sInN1Yil6ImF1dGgwfDEyMzQ1NilsimF1ZCI6IjEyMzRhYmNKZWYiLCJleHAiOjEZMTEyODESNzAsIm
lhdCIGMTMxMTI4MDk3MCwibmFtZS|6lkphbmUgRGOlliwiZ2|2ZW5fomFtZS|6lkphbmUiLCJmYW
pbHfomFtZSI6IkRvZSJ9.bqljxtG9B_bielkqOnjTY9DI9FilFb6IMQINXoYs",
    "token_type":"example",
    "expires_in":3600,
```

```
"example_parameter":"example_value"
```

การดำเนินการแบบละเอียดเกี่ยวกับ JSON Web Tokens (JWT) มีรายละเอียดในเอกสารมาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ด้านการเชื่อมโยงข้อมูล เรื่องข้อกำหนดด้านโปรโตคอลระดับแอบพลิเคชันเอนพอยน์และการจัดการโทเคนและเซสชัน

4.3.3 ขั้นตอนที่ 3: การเรียกใช้ REST API ด้วย Access Token (Call API with Access Token)

ขั้นตอนนี้ ผู้ใช้บริการ API (Consumer System) ต้องดำเนินการเรียกใช้บริการ REST API ของผู้ให้บริการ API (Provider System) ด้วย Access Token ดังรูปที่ 3 ซึ่งขั้นตอนนี้เกิดขึ้นหลังจากที่ผู้ใช้บริการ API (Consumer System) ได้รับ Access Token จากผู้พิสูจน์และยืนยันตัวตน (Identity Provider) แล้วผู้ใช้บริการ API (Consumer System) สามารถนำ Access Token มาประกอบเป็นส่วนหนึ่งของ HTTP Authorization Request Header เพื่อเรียกใช้บริการ REST API จากผู้ให้บริการ API (Provider System) ดังตัวอย่างต่อไปนี้

```
GET /resource HTTP/1.1
    Host: server.example.com
    Authorization: Bearer
eyJhbGciOiJIUzIINilsInR5cCI6IkpXVCJ9.eyJpc3MiOiodHRwOi8vbXktZG9tYWluLmF1dGgwLmNvSlsI
nN1Yil6ImF1dGgwfDEyMzQ1NilsimF1ZCI6IjEyMzRhYmNKZWYiLCJleHAiOjEZMTEyODESNzAsImlh
dCIGMTMxMTI4MDk3MCwibmFtZS|6lkphbmUgRGOlliwiZ2|2ZW5fomFtZS|6lkphbmUiLCJmYWpb
HfomFtZSI6IkRvZSJ9.bqljxtG9B_bielkqOnjTY9DI9FilFb6IMQINXoYs
```

4.3.4 ขั้นตอนที่ 4: การตรวจสอบความถูกต้องของ Access Token (Validate Access Token)

ในขั้นตอนนี้ ผู้ให้บริการ API (Provider System) ต้องดำเนินการตรวจสอบความถูกต้องของ Access Token ว่าเป็น Access Token ที่ออกจากผู้พิสูจน์และยืนยันตัวตน (Identity Provider) และเป็น Access Token ที่ไม่หมดอายุ แล้วดำเนินการให้บริการ API ตามที่ถูกร้องขอเมื่อ Access Token ถูกต้อง ดังรูปที่ 3 ซึ่งการตรวจสอบ Access Token มีรายละเอียดในเอกสารมาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ด้านการเชื่อมโยงข้อมูล เรื่องข้อกำหนดด้านโปรโตคอลระดับแอบพลิเคชัน เอนพอยน์และการจัดการโทเคนและเซสชัน

4.3.5 ขั้นตอนที่ 5: การตอบกลับผลการให้บริการ API (Return Data)

เมื่อผู้ให้บริการ API (Provider System) ตรวจสอบความถูกต้องของ Access Token แล้วดำเนินการให้บริการ API ตามข้อมูลที่ถูกร้องขอสำเร็จควรตอบกลับด้วย HTTP Code 200 ตามตัวอย่างนี้ หรือ HTTP Code อื่นๆ ตามความเหมาะสม ดังรูปที่ 3

```
“messageStatus": {
```

```
        “status": “200",
        “description": " REST API successfully carried out the client requested"
}
```

แต่หากผู้ให้บริการ API (Provider System) ต้องการปฏิเสธการให้บริการกรณี Access Token ไม่ถูกต้องควรตอบกลับด้วย HTTP Code 401 ตามตัวอย่างต่อไปนี้

```
“messageStatus": {
        “status": “401",
        “description": " Unauthorized - Access Token invalid or Access Token not found"
}
```

4.4. ขั้นตอนการดำเนินการเพื่อยืนยันตัวตนด้วยมาตรฐาน Open ID Connect

Open ID Connect (OIDC) เป็นมาตรฐานการยืนยันตัวตนที่ทำงานอยู่บนมาตรฐาน OAuth 2.0 โดยมีจุดเด่นคือการให้ระบบงานใช้ยืนยันตัวตนของผู้ใช้งานเพียงครั้งเดียวแล้วสามารถเข้าไปใช้งานระบบอื่นๆ ได้หลายระบบ (Single Sign On) พร้อมทั้งสามารถบริหารจัดการข้อมูลผู้ใช้งานโดยใช้ผู้พิสูจน์และยืนยันตัวตน (Identity Provider) ที่รองรับมาตรฐาน Open ID Connect ในขณะที่มาตรฐาน OAuth 2.0 จะเน้นการยืนยันตัวตนเพื่อให้สิทธิ์ในการเข้าถึงทรัพยากรต่างๆ เช่น API เป็นต้น

ดังนั้น ผู้ให้บริการ API (Provider System) และผู้ใช้บริการ API (Consumer System) ที่ใช้มาตรฐาน TGIX แล้วมีความต้องการยืนยันตัวตนผู้ใช้งานเพียงครั้งเดียวแล้วสามารถเข้าไปใช้งานระบบของหน่วยงานอื่นๆ ในกลุ่มได้หลายระบบ (Single Sign On) สามารถเลือกยืนยันตัวตนสำหรับการเชื่อมโยงและแลกเปลี่ยนข้อมูลผ่าน API ด้วยผู้พิสูจน์และยืนยันตัวตน (Identity Provider) ที่รองรับมาตรฐาน Open ID Connect ด้วยเช่นกัน ซึ่งมีขั้นตอนการดำเนินการเดียวกันกับที่ใช้ในมาตรฐาน OAuth 2.0 ตามที่กล่าวในหัวข้อก่อนหน้านี้

สิ่งที่แตกต่างกับมาตรฐาน OAuth 2.0 ที่เพิ่มขึ้นมาในมาตรฐาน Open ID Connect คือผู้ให้บริการ API (Provider System) และผู้ใช้บริการ API (Consumer System) สามารถเรียกดูข้อมูลพื้นฐานของผู้ใช้งานด้วย ID Token ซึ่งเป็น Token ที่ผ่านการเข้ารหัสด้วย JSON Web Tokens (JWT) ทั้งนี้ขึ้นอยู่กับ ผู้พิสูจน์และยืนยันตัวตน (Identity Provider) ที่สมาชิกในกลุ่มเลือกใช้ดำเนินการ เช่น ID Token ที่เข้ารหัสด้วย JSON Web Token (JWT) ของ Microsoft Identity Platform [7]

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjFMVE16YWtpaGlSbGFfOHoyQkVKVlhlV01xbyJ9
.eyJ2ZXIiOiIyLjAiLCJpc3MiOiJodHRwczovL2xvZ2luLm1pY3Jvc29mdG9ubGluZS5jb20vOTEyMjA0
MGQtNmM2Ny00YzViLWIxMTItMzZhMzA0YjY2ZGFkL3YyLjAiLCJzdWIiOiJBQUFBQUFBQUFBQUFB
```

QUFBQUFBQUFJa3pxRlZyU2FTYUZIeTc4MmJidGFRIiwiYXVkIjoiNmNiMDQwMTgtYTNmNS00NmE3LWI5OTUtOTQwYzc4ZjVhZWYzIiwiZXhwIjoxNTM2MzYxNDExLCJpYXQiOjE1MzYyNzQ3MTEsIm5iZiI6MTUzNjI3NDcxMSwibmFtZSI6IkFiZSBMaW5jb2xuIiwicHJlZmVycmVkX3VzZXJuYW1lIjoiQWJlTGlAbWljcm9zb2Z0LmNvbSIsIm9pZCI6IjAwMDAwMDAwLTAwMDAtMDAwMC02NmYzLTMzMzJlY2E3ZWE4MSIsInRpZCI6IjkxMjIwNDBkLTZjNjctNGM1Yi1iMTEyLTM2YTMwNGI2NmRhZCIsIm5vbmNlIjoiMTIzNTIzIiwiYWlvIjoiRGYyVVZYTDFpeCFsTUNXTVNPSkJjRmF0emNHZnZGR2hqS3Y4cTVnMHg3MzJkUjVNQjVCaXN2R1FPN1lXQnlqZDhpUURMcSFlR2JJRGFreXA1bW5PcmNkcUhlWVNubHRlcFFtUnA2QUlaOGpZIn0.1AFWW-Ck5nROwSlltm7GzZvDwUkqvhSQpm55TQsmVo9Y59cLhRXpvB8n-55HCr9Z6G_31_UbeUkoz612I2j_Sm9FFShSDDjoaLQr54CreGIJvjtmS3EkK9a7SJBbcpL1MpUtlfygow39tFjY7EVNW9plWUvRrTgVk7lYLprvfzw-CIqw3gHC-T7IK_m_xkr08INERBtaecwhTeN4chPC4W3jdmw_lIxzC48YoQ0dB1L9-ImX98Egypfrlbm0IBL5spFzL6JDZIRRJOu8vecJvj1mq-IUhGt0MacxX8jdxYLP-KUu2d9MbNKpCKJuZ7p8gwTL5B7NlUdh_dmSviPWrw

เมื่อถอดรหัสแล้วจะมีข้อมูลคือ

{
  "typ": "JWT",
  "alg": "RS256",
  "kid": "1LTMzakihiRla_8z2BEJVXeWMqo"
}.{
  "ver": "2.0",
  "iss": "https://login.microsoftonline.com/9122040d-6c67-4c5b-b112-36a304b66dad/v2.0",
  "sub": "AAAAAAAAAAAAAAAAAAAAAIkzqFVrSaSaFHy782bbtaQ",
  "aud": "6cb04018-a3f5-46a7-b995-940c78f5aef3",
  "exp": 1536361411,
  "iat": 1536274711,
  "nbf": 1536274711,
  "name": "Abe Lincoln",
  "preferred_username": "AbeLi@microsoft.com",
  "oid": "00000000-0000-0000-66f3-3332eca7ea81",
  "tid": "9122040d-6c67-4c5b-b112-36a304b66dad",

"nonce": "123523",
"aio":
"Df2UVXL1ix!lMCWMSOJBcFatzcGfvFGhjKv8q5g0x732dR5MB5BisvGQO7YWByjd8iQDLq!eGblDaky
p5mnOrcdqHeYSnltepQmRp6AIZ8jY"
}.[Signature]

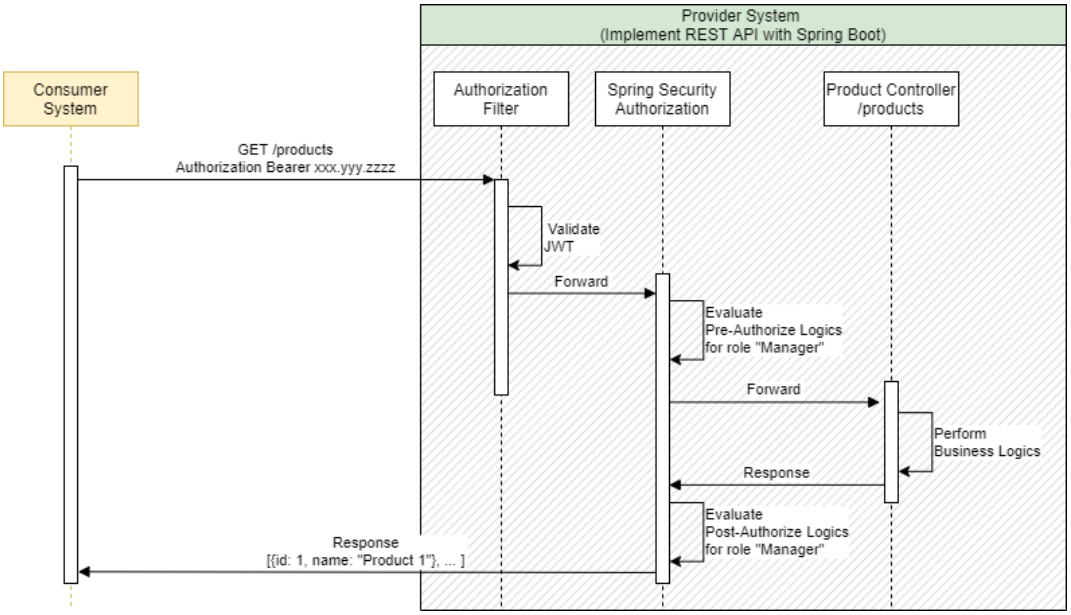
การเข้ารหัสด้วย JSON Web Token (JWT) มีรายละเอียดในเอกสารมาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ด้านการเชื่อมโยงข้อมูล เรื่องข้อกำหนดด้านโปรโตคอลระดับแอบพลิเคชัน เอนพอยน์และการจัดการโทเคนและเซสชัน

4.5. การควบคุมสิทธิ์ในการเข้าถึง API (API Access Control)

การควบคุมสิทธิ์ในการเข้าถึง API ตามมาตรฐาน TGIX มีจุดประสงค์เพื่อให้ผู้ให้บริการ API (Provider System) มั่นใจว่าเฉพาะระบบหรือบุคคลที่ได้รับอนุญาตเท่านั้นถึงจะเข้าถึง API ได้ มีข้อกำหนดดังต่อไปนี้

(1) ผู้ให้บริการ API (Provider System) ต้องตรวจสอบว่า มีเฉพาะผู้ใช้บริการ API (Consumer System) ที่ผ่านการยืนยันตัวตนเท่านั้นที่มีสิทธิ์เข้าถึง API ของผู้ให้บริการ API (Provider System) ตามรายละเอียดการยืนยันตัวตน

(2) ผู้ให้บริการ API (Provider System) ควรออกแบบและพัฒนาการควบคุมสิทธิ์และการตรวจสอบสิทธิ์ตามบทบาทและหน้าที่ของผู้ใช้งาน API ด้วยวิธี Role-Based Access Control (RBAC) อ้างอิงจาก INCITS 359-2012[R2017] Information technology - Role Based Access Control [8] ผู้ใช้งาน API ที่กล่าวถึงนั้นสามารถเป็นระดับของผู้ใช้บริการ API (Consumer System) หรือ ระดับบุคคลผู้ใช้งาน (End User) ในระบบของผู้ใช้บริการ API (Consumer System) ทั้งนี้ขึ้นอยู่กับความต้องการทางธุรกิจ (Business Requirement) ของ API และภาษาโปรแกรมที่ใช้พัฒนา API ดังตัวอย่างขั้นตอนตรวจสอบสิทธิ์แบบ Role-Based Access Control (RBAC) ของ REST API ด้วยภาษาจาวาโดยใช้ Spring Boot และ JWT



รูปที่ 9 ตัวอย่างขั้นตอนตรวจสอบสิทธิ์ภาษาจาวาโดยใช้ Spring Boot และ JWT

ในตัวอย่างข้างต้น เมื่อผู้ใช้บริการ API (Consumer System) เรียกใช้ API GET /products ผู้ให้บริการ API (Provider System) จะตรวจสอบสิทธิ์แล้วพบว่าเป็น Role ชื่อ Manager จึงดำเนินการตามเงื่อนไขทางธุรกิจ แล้วส่งข้อมูลกลับไปให้ผู้ใช้บริการ API (Consumer System)

4.6. การบริหารจัดการบัญชีการใช้งาน (API Accounting)

บัญชีการใช้งานหมายถึงบัญชีที่ผู้ใช้บริการ API (Consumer System) ใช้สำหรับยืนยันตัวตนเพื่อใช้บริการ API ของผู้ให้บริการ API (Provider System) แบ่งประเภทบัญชีได้ตามประเภทการยืนยันตัวตนได้ 3 ประเภท บัญชี คือ

4.6.1 บัญชีใช้งานประเภท API Key ใช้สำหรับการยืนยันตัวตนด้วย API Key

เมื่อผู้ให้บริการ API (Provider System) กำหนดให้ API มีการยืนยันตัวตนด้วยบัญชีการใช้งานประเภท API Key ทั้งผู้ให้บริการ API (Provider System) และผู้ใช้บริการ API (Consumer System) ต้องดำเนินการบริหารจัดการบัญชีการใช้งานให้มีความปลอดภัย ทั้งระหว่างการจัดเก็บและการรับส่งข้อมูล API Key โดยมีแนวทางปฏิบัติดังนี้

(1) ผู้ให้บริการ API (Provider System) ต้องเก็บรักษา API Key ไว้ในที่ปลอดภัย เช่น เก็บไว้ฐานข้อมูล โดยทำการใส่ Prefix และ Hash ค่าของ API Key ดังที่กล่าวไว้ในข้อ 4.2.1

(2) ผู้ใช้บริการ API (Consumer System) ไม่ควรกำหนด API Key ไว้ใน Source Code ซึ่งอาจจะเกิดความผิดพลาดขณะแชร์ Source Code ให้กับบุคคลอื่นได้ ให้เก็บไว้ใน Environment Variable

หรือ File หรือที่อื่นๆ ที่ไม่อยู่ใน Source Code หลัก รวมทั้งทำการ Hash ของมูล API Key ก่อนเก็บเสมอ

(3) ผู้ให้บริการ API (Provider System) ควรออกแบบและพัฒนา API ให้สามารถกำหนด Access Control ของแต่ละ API Key ที่มอบให้แก่ผู้ขอใช้บริการ API ได้

(4) ผู้ให้บริการ API (Provider System) ควรออกแบบและพัฒนา API ให้สามารถกำหนดวันหมดอายุของ API Key ได้

(5) ผู้ให้บริการ API (Provider System) ควรใช้ API Key ใน API ประเภทที่เป็นการอ่านข้อมูลเท่านั้น เนื่องจาก API Key ส่วนข้อมูลประเภทที่เป็นการสร้าง ลบหรือแก้ไขข้อมูล ควรใช้การยืนยันตัวตนระดับบุคคลร่วมด้วย เช่น OAuth 2.0 เป็นต้น

(6) ผู้ให้บริการ API ควรให้บริการ REST API ผ่าน HTTPS (SSL) เท่านั้น

(7) ทั้งผู้ให้บริการ API (Provider System) และผู้ใช้บริการ API (Consumer System) ควรมีการทดสอบความปลอดภัยของระบบเพื่อหาช่องโหว่ที่เกิดจากการใช้ API Key ก่อนการใช้งานจริง เช่น ทดสอบตามหัวข้อ API2:2019 Broken User Authentication ของ OWASP API Security [9]

4.6.2 บัญชีใช้งาน Identity Provider ที่รองรับมาตรฐาน OAuth 2.0

เมื่อผู้ให้บริการ API (Provider System) กำหนดให้ API มีการยืนยันตัวตนด้วยบัญชีการใช้งานประเภท OAuth 2.0 ทั้งผู้ให้บริการ API (Provider System) ผู้ใช้บริการ API (Consumer System) และหน่วยงานผู้บริการ TGIX Platform เพื่อเชื่อมโยงและแลกเปลี่ยนข้อมูล จะต้องดำเนินการดังต่อไปนี้

(1) หน่วยงานผู้บริการ TGIX Platform ดำเนินการจัดเตรียมระบบพิสูจน์และยืนยันตัวตน (Identity Provider) ที่รองรับมาตรฐาน OAuth 2.0

(2) หน่วยงานผู้บริการ TGIX Platform ดำเนินการรับลงทะเบียนบัญชีผู้ใช้งานตามที่ระบุไว้ในหัวข้อ 4.3.1

(3) ผู้ให้บริการ API (Provider System) ควรให้บริการ REST API ผ่าน HTTPS (SSL) เท่านั้น

(4) Access Token ควรมีระยะเวลาการใช้งานได้จำกัด ซึ่งผู้ขอใช้บริการ API จะต้องเรียกใช้บริการ API ก่อนที่ Access Token จะหมดอายุ มีรายละเอียดในเอกสารมาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ด้านการเชื่อมโยงข้อมูล เรื่องข้อกำหนดด้านโปรโตคอลระดับแอบพลิเคชัน เอนพอยน์และการจัดการโทเคนและเซสชัน

(5) เมื่อ Access Token ใกล้หมดอายุผู้ให้บริการ API สามารถเรียก Refresh Token เพื่อขอต่ออายุ Access Token ได้ มีรายละเอียดในเอกสารมาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ด้านการเชื่อมโยงข้อมูล เรื่องข้อกำหนดด้านโปรโตคอลระดับแอบพลิเคชัน เอนพอยน์และการจัดการโทเคนและเซสชัน

4.6.3 บัญชีใช้งาน Identity Provider ที่รองรับมาตรฐาน Open ID Connect

กรณีที่ผู้ให้บริการ API (Provider System) กำหนดให้ API มีการยืนยันตัวตนด้วยบัญชีการใช้งานประเภท Open ID Connect ทั้งผู้ให้บริการ API (Provider System) ผู้ใช้บริการ API (Consumer System) และหน่วยงานผู้บริการ TGIX Platform จะมีการดำเนินการเหมือนกับใช้มาตรฐาน OAuth 2.0

บรรณานุกรม

[1] D. Hardt. (2012, ตุลาคม). The OAuth 2.0 Authorization Framework. [ออนไลน์]. เข้าถึงได้จาก: https://datatracker.ietf.org/doc/html/rfc6749. (วันที่ค้นข้อมูล: 26 ตุลาคม 2021)

[2] M. Jones. (2012, ตุลาคม). The OAuth 2.0 Authorization Framework: Bearer Token Usage. [ออนไลน์]. เข้าถึงได้จาก: https://datatracker.ietf.org/doc/html/rfc6750. (วันที่ค้นข้อมูล: 26 ตุลาคม 2021)

[3] D. Hardt. (2012, ตุลาคม). The OAuth 2.0 Authorization Framework (RFC-6749): Section-1.3.1. [ออนไลน์]. เข้าถึงได้จาก: https://datatracker.ietf.org/doc/html/rfc6749#section-1.3.1. (วันที่ค้นข้อมูล: 26 ตุลาคม 2021)

[4] D. Hardt. (2012, ตุลาคม). The OAuth 2.0 Authorization Framework (RFC-6749): Section-1.3.2. [ออนไลน์]. เข้าถึงได้จาก: https://datatracker.ietf.org/doc/html/rfc6749#section-1.3.2. (วันที่ค้นข้อมูล: 26 ตุลาคม 2021)

[5] D. Hardt. (2012, ตุลาคม). The OAuth 2.0 Authorization Framework (RFC-6749): Section-1.3.3. [ออนไลน์]. เข้าถึงได้จาก: https://datatracker.ietf.org/doc/html/rfc6749#section-1.3.3. (วันที่ค้นข้อมูล: 26 ตุลาคม 2021)

[6] D. Hardt. (2012, ตุลาคม). The OAuth 2.0 Authorization Framework (RFC-6749): Section-1.3.4. [ออนไลน์]. เข้าถึงได้จาก: https://datatracker.ietf.org/doc/html/rfc6749#section-1.3.4. (วันที่ค้นข้อมูล: 26 ตุลาคม 2021)

[7] Microsoft. (2021). Microsoft identity platform ID tokens. [ออนไลน์]. เข้าถึงได้จาก: https://docs.microsoft.com/en-us/azure/active-directory/develop/id-tokens. (วันที่ค้นข้อมูล: 26 ตุลาคม 2021)

[8] Information Technology Industry Council. (2017). Information technology - Role Based Access Control. [ออนไลน์]. เข้าถึงได้จาก:

https://standards.incits.org/apps/group_public/project/details.php?project_id=1906. (วันที่ค้นข้อมูล: 26 ตุลาคม 2021)

[9] OWASP Foundation, Inc. (2019). OWASP API Security. [ออนไลน์]. เข้าถึงได้จาก: https://owasp.org/www-project-api-security/. (วันที่ค้นข้อมูล: 26 ตุลาคม 2021)

[10] Using API keys. (2021). [ออนไลน์]. เข้าถึงได้จาก: https://cloud.google.com/docs/authentication/api-keys. (วันที่ค้นข้อมูล: 26 ตุลาคม 2021)

[11] Hypertext Transfer Protocol (HTTP) Authentication Scheme Registry. (2021, ตุลาคม). [ออนไลน์]. เข้าถึงได้จาก: https://www.iana.org/assignments/http-authschemes/http-authschemes.xhtml#authschemes. (วันที่ค้นข้อมูล: 26 ตุลาคม 2021)