สพร. / CDV
ธันวาคม 2564

ห้ามใช้หรือยึดร่างนี้เป็นมาตรฐาน
มาตรฐานฉบับสมบูรณ์จะมีประกาศในราชกิจจานุเบกษา

ร่าง

มาตรฐานรัฐบาลดิจิทัล
Digital Government Standard

ว่าด้วย มาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ด้านการเชื่อมโยงข้อมูล เรื่อง ข้อกำหนดด้านโปรโตคอลระดับแอปพลิเคชัน เอนพอยน์ และการจัดการโทเคนและเซสชัน

THAILAND GOVERNMENT INFORMATION EXCHANGE STANDARD
SERIES: LINKAGE STANDARD
PART 3 : STANDARD REGULATIONS FOR APPLICATION PROTOCOL, END-POINT, TOKEN AND SESSION MANAGEMENT

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

ผู้ช่วยศาสตราจารย์โษฑศ์รัตต 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
ด้านการเชื่อมโยงข้อมูล
เรื่อง ข้อกำหนดด้านโปรโตคอลระดับแอปพลิเคชัน เอนพอยน์ และการจัดการโทเคนและเซสชัน

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 และบัญชีผู้ใช้งาน 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

สารบัญตาราง

มาตรฐานรัฐบาลดิจิทัล

ว่าด้วย มาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ด้านการเชื่อมโยงข้อมูล เรื่อง ข้อกำหนดด้านโปรโตคอลระดับแอปพลิเคชัน เอนพอยน์ และการจัดการโทเคนและเซสชัน

1. ขอบข่าย

การแลกเปลี่ยนข้อมูลระหว่างหน่วยงานเป็นพื้นฐานหลักที่จำเป็นต่อการพัฒนารัฐบาลดิจิทัล ในปัจจุบันประเทศไทยมีแพลตฟอร์มการแลกเปลี่ยนข้อมูลที่ให้บริการอยู่หลายแห่ง แพลตฟอร์มแต่ละแห่งมี แนวทางและพันธกิจในการดำเนินงานเป็นของตนเอง เป็นผลให้การบูรณาการข้อมูลภาครัฐจำเป็นต้อง ขับเคลื่อนด้วยการสร้างมาตรฐานหรือข้อตกลงร่วมกันในการแลกเปลี่ยนข้อมูลสำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน) ได้เล็งเห็นความสำคัญในจุดนี้ จึงมีความจำเป็นต้องจัดทำมาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ เพื่อใช้ในการ แลกเปลี่ยนข้อมูลระหว่างหน่วยงานของรัฐเพื่อให้เกิดการบูรณาการข้อมูลเกิดขึ้นอย่างเป็นรูปธรรม

เป้าประสงค์หลักของการใช้มาตรฐานฯ เป็นตัวขับเคลื่อนการบูรณาการข้อมูลภาครัฐ คือ การให้หน่วยงาน ของรัฐมีแนวทางในการพัฒนาสถาปัตยกรรมระบบสารสนเทศเพื่อใช้ในการแลกเปลี่ยนข้อมูลที่ชัดเจน มีความสอดคล้องในการเชื่อมต่อระหว่างกัน

ดังนั้นเพื่อให้บรรลุเป้าประสงค์หลักดังกล่าวเอกสารฉบับนี้จึงนำเสนอข้อกำหนดด้านโปรโตคอลระดับแอปพลิเคชัน เอนพอยน์ และการจัดการโทเคนและเซสชัน สำหรับประกอบเอกสารว่าด้วยมาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ เรื่องมาตรฐานสถาปัตยกรรมการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐระดับการเชื่อมโยงข้อมูลที่เหมาะสมกับบริบทของประเทศไทยเท่านั้น

2. นิยาม

นิยามคำศัพท์ที่เกี่ยวข้องกับมาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐด้านการเชื่อมโยงข้อมูล เรื่อง ข้อกำหนดด้านโปรโตคอลระดับแอปพลิเคชัน เอนพอยน์และการจัดการโทเคนและเซสชันที่ใช้เอกสารฉบับนี้มีดังนี้

2.1 Hypertext Transfer Protocol (HTTP) หมายความว่า โปรโตคอลในระดับชั้นเซสชัน (Session Layer) ของตัวแบบ OSI ทำหน้าที่สื่อสารผ่านระบบเครือข่ายคอมพิวเตอร์ใช้ในการรับและส่งข้อมูล ในการแลกเปลี่ยนข้อมูลระหว่างเครื่องลูกข่ายและเครื่องแม่ข่าย

2.2 Hypertext Transfer Protocol Secure (HTTPS) หมายความว่า โปรโตคอลในระดับชั้นเซสชัน (Session Layer) ของตัวแบบ OSI ทำหน้าที่สื่อสารผ่านระบบเครือข่ายคอมพิวเตอร์ใช้ในการรับและส่งข้อมูล ที่ช่วยรักษาความสมบูรณ์ของข้อมูลในการแลกเปลี่ยนข้อมูลระหว่างเครื่องลูกข่ายและเครื่องแม่ข่าย รวมทั้งเก็บข้อมูลนั้นไว้เป็นความลับ

2.3 Transport Layer Security (TLS) หมายความว่า โปรโตคอลในระดับชั้นเซสชัน (Session Layer) ของตัวแบบ OSI ใช้เทคโนโลยีการเข้ารหัสข้อมูล เพื่อเพิ่มความปลอดภัยในการส่งข้อมูลบนเครือข่ายอินเทอร์เน็ต ระหว่างเครื่องเซิร์ฟเวอร์กับเว็บเบราว์เซอร์หรือ แอปพิเคชันที่ใช้งาน เพื่อให้ข้อมูลปลอดภัยจากการเข้าถึงโดยแฮกเกอร์ วิธีการเรียกใช้งานจะเรียกผ่านโปรโตคอล HTTPS หรือโปรโตคอลความปลอดภัยอื่นๆ ตามแต่วิธีการใช้งาน

2.4 Transmission Control Protocol (TCP) หมายความว่า โปรโตคอลระดับชั้นทรานสปอร์ต (Transport Layer) ของตัวแบบ OSI ทำหน้าที่ควบคุมการรับส่งข้อมูลระหว่างผู้ส่งกับผู้รับ เพื่อใช้แลกเปลี่ยนข้อมูลระหว่างกัน โดยมีการตรวจสอบให้แน่ใจว่าทุกแพ็กเกจที่รับส่งมีความถูกต้อง

2.5 Application Programming Interface หรือ API หมายความว่า ช่องทางการเชื่อมโยงและการแลกเปลี่ยนข้อมูลภาครัฐระหว่างผู้ให้บริการและผู้ใช้บริการ

2.6 Representational State Transfer (REST API หรือ RESTful API) หมายความว่า ช่องทางการเชื่อมโยงและการแลกเปลี่ยนข้อมูลภาครัฐระหว่างผู้ให้บริการและผู้ใช้บริการตามมาตรฐาน TGIX

2.7 JavaScript Object Notation (JSON) หมายความว่า รูปแบบของโครงสร้างข้อมูลที่ใช้แลกเปลี่ยนผ่าน REST API

2.8 TGIX JSON Data Format หมายความว่า รูปแบบของมาตรฐานโครงสร้างข้อมูลที่ใช้แลกเปลี่ยนผ่าน REST API ตามมาตรฐาน TGIX

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

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

2.11 ผู้ให้บริการ TGIX Platform (TGIX Platform Provider) หมายความว่า ระบบสารสนเทศของหน่วยงานผู้ให้บริการ TGIX Platform เพื่อสนับสนุนดำเนินการเชื่อมโยงและการแลกเปลี่ยนข้อมูลให้เป็นไปตามมาตรฐาน TGIX

2.12 การบริการออกใบรับรอง (Certification Authority) หมายความว่า บริการของผู้ให้บริการ TGIX Platform ทำหน้าที่ออกใบรับรองอิเล็กทรอนิกส์ (Digital Certificate) ให้แก่สมาชิกในกลุ่ม TGIX

2.13 การลงลายมือชื่อดิจิทัล (Digital Signature) หมายความว่า การลงลายมือชื่อดิจิทัลโดยใช้ใบรับรองอิเล็กทรอนิกส์ (Certificate) ที่ระบุตัวบุคคล หรือองค์กรผู้เป็นเจ้าของลายมือชื่อ เพื่อแสดงว่าบุคคลหรือองค์กร ดังกล่าวยอมรับข้อความในข้อมูลอิเล็กทรอนิกส์นั้น

3. กฎหมายและแนวปฏิบัติที่เกี่ยวข้อง

การเชื่อมโยงและการแลกเปลี่ยนข้อมูลภาครัฐมีการบัญญัติไว้ในกฎหมายหรือแนวปฏิบัติที่เกี่ยวข้องดังนี้

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

3.2 พระราชบัญญัติการบริหารงานและการให้บริการภาครัฐผ่านระบบดิจิทัล พ.ศ. 2562 ในมาตรา 13 ระบุว่า เพื่อประโยชน์ในการบริหารราชการแผ่นดินและการให้บริการประชาชน ให้หน่วยงานของรัฐจัดให้มีการเชื่อมโยงและแลกเปลี่ยนข้อมูลดิจิทัลที่มีการจัดทำและครอบครองตามที่หน่วยงานของรัฐแห่งอื่นร้องขอ ที่จะเกิดการบูรณาการร่วมกัน

มาตรา 15 ระบุว่า ให้มีศูนย์แลกเปลี่ยนข้อมูลกลางทำหน้าที่เป็นศูนย์กลางในการแลกเปลี่ยนข้อมูลดิจิทัลและทะเบียนดิจิทัลระหว่างหน่วยงานของรัฐ เพื่อสนับสนุนการดำเนินการของหน่วยงานของรัฐในการให้บริการประชาชนผ่านระบบดิจิทัล และดำเนินการในเรื่องดังต่อไปนี้

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

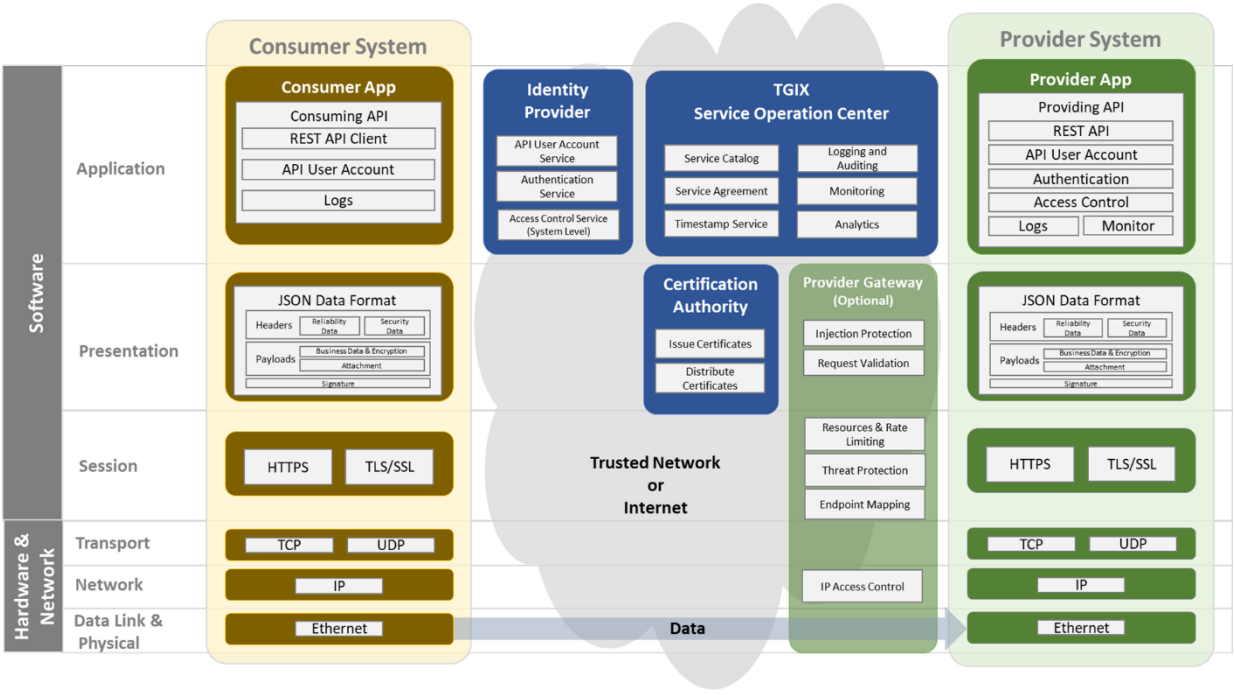
มาตรา 19 ระบุว่า ในวาระเริ่มแรก ให้สำนักงานดำเนินการให้มีศูนย์แลกเปลี่ยนข้อมูลกลางตามมาตรา 15 เป็นการชั่วคราวแต่ไม่เกินสองปี เมื่อครบกำหนดระยะเวลาดังกล่าว ให้คณะกรรมการพัฒนารัฐบาลดิจิทัลพิจารณาความจำเป็นและเหมาะสมเกี่ยวกับหน่วยงานของรัฐที่จะมาดำเนินการเกี่ยวกับศูนย์แลกเปลี่ยนข้อมูลกลาง ทั้งนี้ ในกรณีที่คณะกรรมการพัฒนารัฐบาลดิจิทัลเห็นควรให้หน่วยงานของรัฐแห่งอื่น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

4. ข้อกำหนดด้านโปรโตคอลระดับแอปพลิเคชัน เอนพอยน์และการจัดการโทเคน และเซสชัน

4.1 การทำงานของโปรโตคอล

การเชื่อมโยงและการแลกเปลี่ยนข้อมูลตามมาตรฐาน TGIX มีองค์ประกอบของสถาปัตยกรรมการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ดังรูปที่ 1 แสดงการเชื่อมโยงข้อมูลจาก REST API Client ของผู้ใช้บริการ API (Consumer System) ไปยังเอนพอยน์ (Endpoint URL) ของ REST API ของผู้ให้บริการ API (Provider System) ซึ่งเอนพอยน์ต้องมีโปรโตคอลเป็น Hypertext Transfer Protocol Secure (HTTPS) ใช้ร่วมกันกับโปรโตคอลสำหรับการรับรองความปลอดภัยในรูปแบบ Transport Layer Security (TLS) และ Secure Socket Layer (SSL) โดยขั้นตอนทั้งหมดที่กล่าวมาจะทำงานอยู่บน Transmission Control Protocol (TCP)

นอกจากนี้ข้อมูลที่ใช้ในการเชื่อมโยงและแลกเปลี่ยนทั้งหมดของ REST API ถูกรวมไว้ในรูปแบบ JSON Data Format ซึ่งประกอบด้วยข้อมูลเชิงธุรกรรม (Business Data) พร้อมทั้งข้อมูลที่เกี่ยวข้องกับความปลอดภัยเพิ่มเติม เช่น ลงลายมือชื่อดิจิทัล (Digital Signature) และ Online Certificate Status Protocol (OCSP) เพื่อใช้งานกับ Certification Authority เป็นต้น



รูปที่ 1 ภาพรวมองค์ประกอบของการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ

4.2 ข้อกำหนดด้านโปรโตคอลที่เกี่ยวข้องกับเอนพอยน์

ข้อกำหนดด้านโปรโตคอลที่เกี่ยวข้องกับเอนพอยน์ (Endpoint URL) ในการเชื่อมโยงและแลกเปลี่ยนข้อมูลตามมาตรฐาน TGIX มีดังต่อไปนี้

(1) กำหนดให้ผู้ใช้บริการ API (Consumer System) มีการเรียกใช้งาน Endpoint URL ของผู้ให้บริการ (Provider System) ผ่านโปรโตคอล HTTPS เท่านั้น

(2) กำหนดให้ผู้ให้บริการ (Provider System) และ ผู้ใช้บริการ API (Consumer System) มีการใช้งาน TLS version 1.2 เป็นอย่างน้อยสำหรับการใช้งาน TLS/SSL

(3) กำหนดให้ผู้ให้บริการ (Provider System) และ ผู้ใช้บริการ API (Consumer System) ใช้งาน Transmission Control Protocol (TCP) ผ่าน TLS/SSL เท่านั้น

4.3 ข้อกำหนดด้านโครงสร้าง TGIX JSON Data Format ตามมาตรฐาน TGIX

การกำหนดโครงสร้าง TGIX JSON Data Format ตามมาตรฐาน TGIX เป็นการกำหนดรูปแบบโครงสร้างการรับส่งข้อมูลผ่าน REST API ระหว่างผู้ใช้บริการ API (Consumer System) และ ผู้ให้บริการ API (Provider System) ซึ่งมีโครงสร้างดังรูปที่ 1 โดยโครงสร้างของ TGIX JSON Data Format สามารถแบ่งตามประเภทการแลกเปลี่ยนข้อมูลดังตารางที่ 1

ตารางที่ 1 ประเภทการแลกเปลี่ยนข้อมูล

ประเภทการแลกเปลี่ยน		Content Type	รายละเอียด
การแลกเปลี่ยนข้อมูลเชิงธุรกรรม	ข้อมูลเชิงธุรกรรมที่กำหนดลักษณะ Payload เป็นรูปแบบ JSON	กำหนด Content Type ประเภท JSON	กำหนดข้อความมีลักษณะเป็น JSON ทั้งหมด
	ข้อมูลเชิงธุรกรรมที่กำหนดลักษณะ Payload ไม่ได้เป็นรูปแบบ JSON	กำหนด Content Type ประเภท Multipart	เพื่อรองรับการการแลกเปลี่ยนข้อมูล Format อื่นๆ เช่น XML, ebXML เป็นต้น ทำให้มาตรฐาน TGIX สามารถทำงานร่วมกับ Data Format อื่นๆ ได้

ตารางที่ 2 ประเภทการแลกเปลี่ยนข้อมูล (ต่อ)

ประเภทการแลกเปลี่ยน		**Content Type**	**รายละเอียด**
การแลกเปลี่ยนข้อมูลที่เป็น File	ขนาดของ File ไม่เกิน 5 MB	กำหนด Content Type ประเภท Multipart	เพื่อรองรับการแลกเปลี่ยนข้อมูลแบบ File ที่มีขนาดไม่เกิน 5MB
	ขนาดของ File มากกว่า 5 MB ขึ้นไป	กำหนด Content Type ประเภท Multipart ร่วมกับ Resumable หากต้องการ	เพื่อรองรับการแลกเปลี่ยนข้อมูลแบบ File ที่มีขนาดเกิน 5MB รวมทั้งสามารถ Resume ได้กรณีการ Upload มีปัญหาตามมาตรฐาน Form-based File Upload in HTML: RFC-1867 [1] และ Hypertext Transfer Protocol (HTTP/1.1): Range Requests: RFC-7233 [2]

4.3.1 ส่วน TGIX Message Headers

ในส่วนของ Message จะประกอบด้วย HTTP Header และ HTTP Body โดยมีรายละเอียดดังตารางที่ 3 และตารางที่ 4

ตารางที่ 3 รายละเอียดโครงสร้างของ TGIX Message Headers ส่วน HTTP Header

พารามิเตอร์	ความจำเป็นต้องมี	รายละเอียด
HTTP Method	Required	กำหนด HTTP Method โดยรองรับ HTTP/1.1 [3] และโดยรองรับกำหนดค่าเป็น POST, GET, DELETE, PUT, OPTIONS และ PATCH
Authorization	Required	กำหนดรหัสการยืนยันตนของผู้ใช้งาน โดยกำหนดค่าเป็น Bearer เสมอ
Accept-Encoding	Required	กำหนดการเข้ารหัสข้อมูล
Accept-Language	Required	กำหนดภาษาในการตอบรับ
Accept	Required	กำหนดประเภทของเนื้อหา
Host	Required	กำหนด URL ปลายทาง
Cache-Control	Required	กำหนดคำสั่งชี้แนะว่าจะต้องทำตามกลไกการเก็บแคชทั้งหมดโดยตลอดทั้งการร้องขอและการตอบรับ
Connection	Required	กำหนดวิธีการเชื่อมต่อ
Content-Type	Required	กำหนดชนิดของเนื้อหาที่ร้องขอ
Content-Length	Required	กำหนดความยาวของข้อมูลเนื้อหา
Origin	Required	กำหนด URL ต้นทาง
messageVersion	Required	กำหนดเวอร์ชันของ API
MessageId	Required	กำหนดรหัสของข้อความ
Timestamp	Required	กำหนดเวลาในการร้องขอ
clientId	Required	กำหนดรหัสของผู้ใช้บริการ
event	Required	กำหนดรายละเอียดการที่จะดำเนินการ (Action)

ตารางที่ 4 รายละเอียดโครงสร้างของ TGIX Message Headers ส่วน HTTP Body

พารามิเตอร์	ความจำเป็นต้องมี	รายละเอียด
ExpirationTime	Required	กำหนดเวลาที่หมดอายุของข้อความ
RequestId	Required	กำหนดรหัสของการร้องขอสำหรับตอบกลับ
Action: Method	Required	กำหนด HTTP Method ในการร้องขอ
Action: Path	Required	กำหนด Context Path ในการร้องขอ
Action: URL	Required	กำหนด URL ในการร้องขอ
Action: Parameter	Optional	กำหนดพารามิเตอร์ เพิ่มเติม โดยต้องกำหนด “parameterName”: “value” เสมอ
messageStatus	Required	กำหนดสถานะของข้อความ
messageStatus: status	Required	กำหนดสถานะตามมาตรฐาน HTTP Status
messageStatus: description	Required	กำหนดรายละเอียดสถานะ
messageStatus: error	Required	กำหนดรายละเอียดกรณีร้องขอไม่สำเร็จ โดยผู้ให้บริการเป็นผู้กำหนดเอง
error: code	Required	กำหนดรหัสของ Error
error: message	Required	กำหนดข้อความที่ต้องการแสดง Error
apiKey	Optional	กำหนดรหัสของ API
Headers	Optional	กำหนด Header เพิ่มเติมของข้อความ
attachMents	Optional	กำหนด Metadata ของเอกสารแนบ โดยรองรับการแนบเอกสารได้มากกว่า 1 ไฟล์ แต่ขนาดรวมกันไม่เกิน 5 เมกะไบต์
attachMents: referenceId	Required	กำหนดที่อยู่ของเอกสารแนบ
attachMents: name	Required	กำหนดชื่อของเอกสารแนบ
attachMents: mimeType	Required	กำหนดประเภทของเอกสารแนบ
attachMents: sequence	Required	กำหนดลำดับของเอกสารแนบ
attachMents: description	Optional	กำหนดรายละเอียดของเอกสารแนบ

4.3.2 ส่วน TGIX Message Payloads

ในส่วนของ Message ประกอบด้วย HTTP Body เป็นการกำหนดรูปแบบการรับส่งข้อมูลแบบ Multipart Content-Type เพื่อรองรับการรับส่งข้อมูลจากผู้ขอใช้บริการข้อมูลได้หลายรูปแบบ แต่อยู่ภายใต้ Header ที่เป็นเป็น JSON โดยมีรายละเอียดดัง ตารางที่ **5**

ตารางที่ 5 รายละเอียดโครงสร้างของ TGIX Message Payloads ส่วน HTTP Body

พารามิเตอร์	ความจำเป็นต้องมี	รายละเอียด
Content-Type	Required	กำหนด Body ของเนื้อหา โดยสามารถกำหนดได้หลายรูปแบบ (Any MimeType) เช่น JSON, XML และ File เป็นต้น

4.3.3 ส่วน TGIX Message Signature

ในส่วนของ Message จะประกอบด้วย HTTP Body เป็นการตรวจสอบความถูกต้องและครบถ้วนของข้อมูลที่รับส่งระหว่างผู้ใช้บริการและผู้ให้บริการ โดยการนำข้อมูล Header และ Payloads ที่ได้รับมาไปเข้ารหัสเทียบกับค่าของ sigValue ที่ถอดรหัสแล้ว ถ้าตรงกันจะถือว่าข้อมูลที่รับส่งนั้นครบถ้วนสมบูรณ์ โดยมีรายละเอียดดังตารางที่ 6

ตารางที่ 6 รายละเอียดโครงสร้างของ TGIX Message Payloads ส่วน HTTP Body

พารามิเตอร์	ความจำเป็นต้องมี	รายละเอียด
alg	Required	กำหนดอัลกอริธึมของกุญแจ เช่น RS256, RFC-7518
cert	Required	กำหนด Public Key ของลายมือชื่อดิจิทัล
sigValue	Required	กำหนดกุญแจที่ใช้ในการลงลายมือชื่อดิจิทัล

4.3.4 ตัวอย่างโครงสร้าง TGIX JSON Data Format

4.3.4.1 โครงสร้าง TGIX JSON Data Format กรณีการแลกเปลี่ยนข้อมูลเชิงธุรกรรม

(1) การแลกเปลี่ยนข้อมูลเชิงธุรกรรม โดยกำหนดลักษณะ Payload เป็น JSON ทั้งข้อความ โดยกำหนด Content Type เป็นประเภท JSON ดังตัวอย่างรูปที่ 2

```
// ============================================================
// Request Header
// ============================================================
   POST https://oneweb.tgix.com/api/v1/sendmessage HTTP/1.1
```

```
    Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxM
jM0NTY3ODkwIiwibmFtZSI6IlRHSVggREdBIiwiaWF0IjoxNTE2MjM5MDIyfQ.gs6d4E
8CIkIs8FC-IysF8p7aBbE2gapTBT0e5xwUHug
    Accept-Encoding: gzip, deflate
    Accept-Language: en-US,en;q=0.5
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
    Host: localhost:8000
    Cache-Control: no-cache
    Connection: keep-alive
    Content-Type: application/json;charset=UTF-8
    Content-Length: 5125
    Origin: localhost.com
// ============================================================
// Message Signature: TGIX-Object
// ============================================================
{
 "TGIXHeader": {
    "messageVersion": "V1.0.0",
    "MessageId": "M1633151789",
    "Timestamp": "2021-10-08T08:10:12.24+07:00",
    "clientId": "452435",
    "apiKey": "",
    "event":"SendMessage",
    "RequestId":"423984729387",
    "Headers":"",
    "Host":"localhost",
    "InitiatorId":"",
    "ConversationId":"1",
    "SourceAddress":"http://localhost:8081/callService",
    "DestinationAddress":"http://localhost:8080/xmlService",
    "ResponseAddress":"",
    "FaultAddress":"",
```

```
    "ExpirationTime":"1643151789",
    "SentTime":"",
    "Action":{
        "Protocol": "http",
        "Method": "POST",
        "Path": "/searchJuristic",
        "URL": "http://localhost:8081"
    }
},
 "TGIXPayload": {
"JuristicID": "0133552005772"
},
  " TGIXSignature":{
            "alg" : "RS256",
            " cert" : "<<public key of signer Alice>>",
            " sigValue" : "<<signature Alice>>"
         }
}
```

รูปที่ 2 ตัวอย่างการแลกเปลี่ยนข้อมูลเชิงธุรกรรม โดยกำหนดลักษณะ Payload เป็น JSON

(2) การแลกเปลี่ยนข้อมูลเชิงธุรกรรม โดยกำหนดลักษณะ Payload ไม่ได้เป็น JSON โดยกำหนด Content Type เป็นประเภท Multipart เพื่อรองรับ XML Message หรือการแนบ File ขนาดไม่เกิน 5 MB ดังตัวอย่างรูปที่ 3

```
// ============================================================
// Request Header
// ============================================================
    POST https://oneweb.tgix.com/api/v1/sendmessage HTTP/1.1
    Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxM
jM0NTY3ODkwIiwibmFtZSI6IlRHSVggREdBIiwiaWF0IjoxNTE2MjM5MDIyfQ.gs6d4E
8CIkIs8FC-IysF8p7aBbE2gapTBT0e5xwUHug
    Accept-Encoding: gzip, deflate
    Accept-Language: en-US,en;q=0.5
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
    Host: localhost:8000
    Cache-Control: no-cache
    Connection: keep-alive
    Content-Type: multipart/related; boundary=tgix_message
    Content-Length: 5125
    Origin: localhost.com
// ============================================================
// Message: TGIX-Object
// ============================================================
--tgix_message
Content-Type: application/json; charset=UTF-8
{
 "TGIXHeader": {
    "messageVersion": "V1.0.0",
    "MessageId": "M1633151789",
    "Timestamp": "2021-10-08T08:10:12.24+07:00",
    "clientId": "452435",
    "apiKey": "",
    "event":"SendMessage",
```

```
"RequestId":"423984729387",
"Headers":"",
"Host":"localhost",
"InitiatorId":"",
"ConversationId":"1",
"SourceAddress":"http://localhost:8081/callService",
"DestinationAddress":"http://localhost:8080/xmlService",
"ResponseAddress":"",
"FaultAddress":"",
"ExpirationTime":"1643151789",
"SentTime":"",
"Action":{
    "Protocol": "http",
    "Method": "POST",
    "Path": "/searchJuristic",
    "URL": "http://localhost:8081"
},
"attachMents":[
    {
        "mimeType": "text/xml",
        "contentId": "0",
        "name": "Screen Shot 2564-10-08 at 10.41.34.png",
        "referenceId": "170e80bc-281e-11ec-9621-0242ac130000",
        "sequence": "1"
      },
      {
        "mimeType": "image/jpeg",
        "contentId": "1",
        "name": "Screen Shot 2564-10-08 at 10.07.10.png",
        "referenceId": "170e80bc-281e-11ec-9621-0242ac130001",
        "sequence": "1"
      }
```

```
    ],
    "messageStatus":
    {
        "status": "",
        "description": "",
        "error": {
            "code": "",
            "message": ""
        }
    }
},
 "TGIXPayload": {
          "JuristicID": "0123456789012"
},
  " TGIXSignature":{
            "alg" : "RS256",
            " cert" : "<<public key of signer Alice>>",
            " sigValue" : "<<signature Alice>>"
         }
}
```

รูปที่ 3 ตัวอย่างการแลกเปลี่ยนข้อมูลเชิงธุรกรรม โดยกำหนดลักษณะ Payload ไม่ได้เป็น JSON

4.3.4.2 โครงสร้าง TGIX JSON Data Format กรณีการ File ขนาดเกิน 5 MB

การแลกเปลี่ยนข้อมูล File ขนาดเกิน 5 MB มีตัวอย่างดังตัวอย่างรูปที่ 4

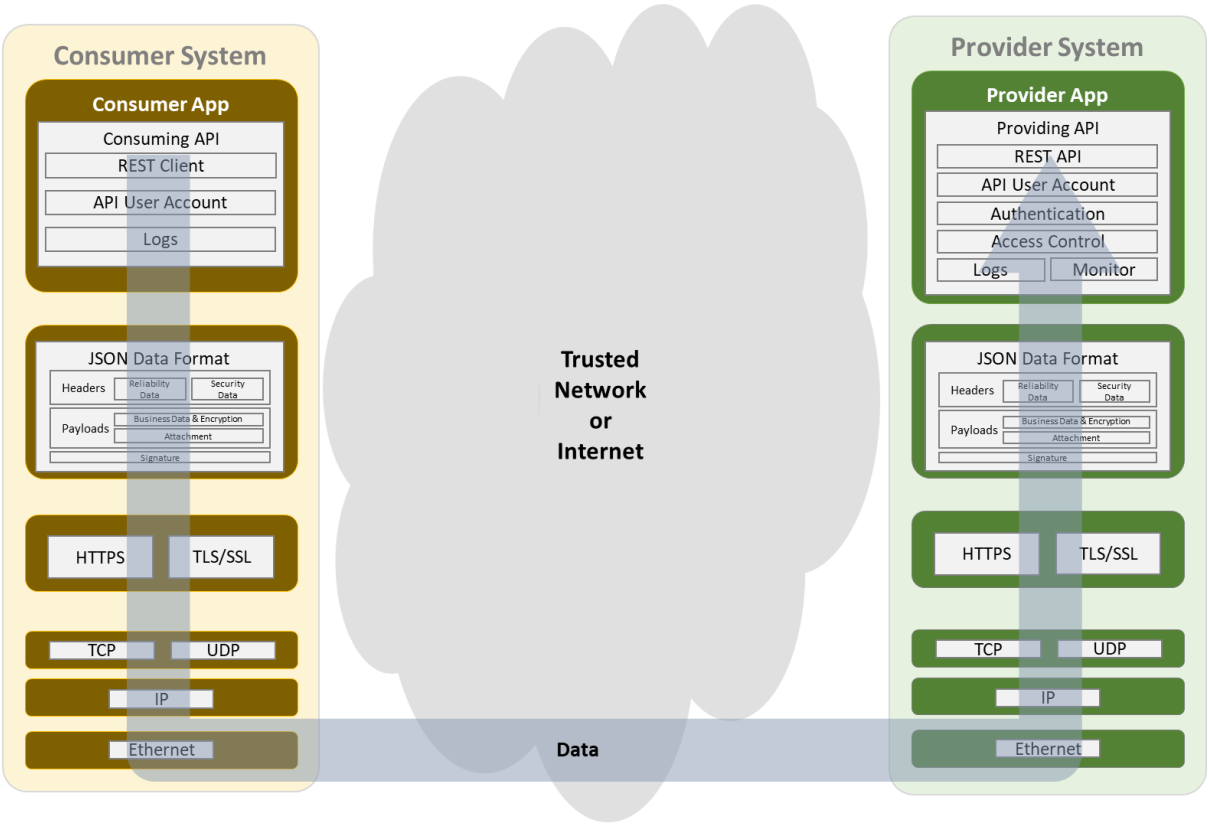
```
//Consumer System ส่ง Request ไปแจ้งว่าจะมีการ Upload File พร้อมแจ้งขนาดและจำนวน Chunk ที่จะส่ง
PATCH /document HTTP/1.1
Content-Type: multipart/byteranges; boundary=THIS_STRING_SEPARATES

//Consumer System แตก File ออกเป็น Chunk ย่อยๆ แล้วทยอย Upload ทีละ File จนสำเร็จ
--THIS STRING SEPARATES
Content-Type: text/plain
Content-Range: bytes 10-21/22

1234567890
--THIS_STRING_SEPARATES--
```

รูปที่ 4 ตัวอย่างการการแลกเปลี่ยนข้อมูลเชิงธุรกรรม โดยกำหนดลักษณะ File

4.4 การบริหารจัดการ Session

4.4.1 การใช้งานเซสชัน (Session)

เซสชันตามมาตรฐาน TGIX คือกลุ่มของข้อมูลใช้สำหรับการโต้ตอบระหว่างผู้ใช้บริการ (Consumer System) และผู้ให้บริการ (Provider System) ที่เกิดขึ้นภายในช่วงเวลาที่กำหนด ดังรูปที่ 5



รูปที่ 5 การส่งข้อมูลระหว่างผู้ใช้บริการ (Consumer System) และผู้ให้บริการ (Provider System)

เซสชัน (Session) เดียวสามารถมีได้หลายกิจกรรม ซึ่งเซสชันทั้งหมดเก็บไว้ชั่วคราวในขณะที่ผู้ใช้เชื่อมต่ออยู่ ตามมาตรฐาน TGIX มีการกำหนดให้มีการใช้การยืนยันตัวตน ด้วย Open ID Connect 1.0 (OIDC) เมื่อผู้ใช้เข้าสู่ระบบจะมีการสร้างเซสชันสำหรับผู้ใช้ที่รับรองความถูกต้องผ่านแอปพลิเคชัน และเมื่อต้องการตรวจสอบสิทธิ์สามารถใช้ข้อมูลในเซสชันช่วยเป็นตัวกำหนดว่าผู้ใช้ต้องได้รับการตรวจสอบทุกครั้งที่มีการขอเพื่อเข้าใช้งานบริการข้อมูลของผู้ให้บริการ API (Provider System) โดยมีการใช้งานร่วมกับโทเค็น (Token) ที่มีการส่งไปพร้อมกับข้อมูลขอใช้บริการ รายละเอียดโทเค็นจะกล่าวถึงในส่วนการบริหารจัดการโทเค็นต่อไป เซสชันสามารถแบ่งออกได้เป็น 3 ส่วน คือ

(1) แอปพลิเคชันเซสชัน (Application Session)

ส่วนนี้เป็นเซสชันภายในแอปพลิเคชัน ใช้สำหรับติดตามผู้ใช้งานมีการลงชื่อเข้าใช้งานหรือไม่ โดยการจัดเก็บข้อมูลไว้ในคุกกี้ (Cookie) หรือมีประโยชน์ในการเก็บข้อมูลกิจกรรมที่เกิดขึ้นในแอปพลิเคชันว่ามีการเข้าใช้งานระบบส่วนไหนบ้าง

(2) เซสชันการอนุญาต (Authorization Session)

เซสชันการอนุญาต เป็นการเก็บเซสชันบนเซิร์ฟเวอร์การให้สิทธิ์สำหรับผู้ใช้และจัดเก็บข้อมูลผู้ใช้ไว้ในคุกกี้ เซสชันนี้ใช้เพื่อให้ครั้งต่อไปที่ผู้ใช้ถูกเปลี่ยนเส้นทาง (Redirect) ไปยังโปรแกรมบริการเพื่อเข้าสู่ระบบข้อมูลของผู้ใช้จะถูกจดจำสำหรับการใช้งานลงชื่อเพียงครั้งเดียว (SSO)

(3) เซสชันผู้ให้บริการข้อมูลประจำตัว (Identity Provider Session)

เซสชันนี้เกิดขึ้นเมื่อผู้ใช้ลงชื่อเข้าใช้งานระบบโดยใช้ผู้ให้บริการข้อมูลประจำตัว เช่น Identity Provider Server และมีการลงชื่อเข้าใช้ถูกต้องอยู่แล้ว ผู้ให้บริการข้อมูลประจำตัวจะสร้างเซสชันขึ้นเพื่อเก็บข้อมูลประจำตัว เมื่อผู้ใช้มีการใช้งานข้อมูลประจำตัวจะไม่ได้รับแจ้งให้ลงชื่อเข้าใช้อีก

4.4.2 การกำหนดอายุของเซสชัน (Session Lifetime Limits)

มาตรฐาน TGIX แนะนำให้มีการกำหนดค่าในส่วนนี้ เพื่อเป็นการกำหนดว่าผู้ให้บริการควรจะเก็บเซสชันไว้นานเท่าไหร่ก่อนจะทำการออกจากระบบโดยอัตโนมัติ โดยผู้ให้บริการที่พัฒนาระบบตามมาตรฐาน OAuth 2.0 นั้นจะต้องมีการกำหนดค่าหมดเวลาไม่ใช้งาน (Inactivity timeout) คือกรอบเวลาหลังจากที่เซสชันของผู้ใช้จะหมดอายุหากไม่ได้โต้ตอบกับเซิร์ฟเวอร์การให้สิทธิ์ จะถูกทำให้ออกจากระบบหากเกินเวลาที่กำหนด และค่าต้องเข้าสู่ระบบหลังจากเวลาที่กำหนด (Require log in after) คือ กรอบเวลาที่กำหนดให้ผู้ใช้งานจะต้องเข้าสู่ระบบอีกครั้ง

4.4.3 การล้างเซสชัน

ส่วนของการล้างเซสชันมาตรฐาน TGIX แนะนำให้ทำการล้างเซสชันเมื่อผู้ใช้งานออกจากระบบหรือแอบพลิเคชันนั้นๆ

4.4.3.1 การล้างเซสชันระดับแอปพลิเคชัน

เซสชันในส่วนปกติแล้วจะเกิดขึ้นเมื่อมีผู้ใช้งานเข้ามาใช้งานแอปพลิเคชันของผู้ขอใช้บริการ (Consumer) จะมีการสร้างเซสชันขึ้นมาเพื่อใช้งาน เมื่อผู้ใช้งานออกจากระบบ การล้างเซสชันในส่วนนี้จะต้องเป็นหน้าที่ของแอปพลิเคชันที่ต้องทำการล้างเซสชันทั้งหมดที่เกิดขึ้น โดยการบริหารจัดการในส่วนนี้สามารถทำได้ดังนี้

(1) การกำหนดอายุของเซสชัน (Session Expiration)

เพื่อลดระยะเวลาที่ผู้โจมตีสามารถเริ่มการโจมตีในเซสชันที่ใช้งานอยู่และ ขโมยเซสชันเหล่านั้น จำเป็นต้องกำหนดอายุสำหรับทุกเซสชัน โดยกำหนดระยะเวลาที่เซสชันจะยังคงทำงานอยู่ การกำหนดอายุของเซสชันที่นานเกินความจำเป็นสำหรับเว็บแอปพลิเคชันจะเพิ่มช่องโหว่ของการโจมตีตามเซสชันได้ โดยผู้โจมตีจะสามารถใช้ ID เซสชันที่ถูกต้องโจมตีซ้ำๆ ได้อยู่ ยิ่งการกำหนดช่วงเซสชันสั้นลงเท่าใด ผู้โจมตีก็จะมีโอกาสใช้รหัสเซสชันที่ถูกต้องน้อยลงเท่านั้น ดังนั้นการกำหนดเวลาหมดอายุของเซสชันตามมาตรฐาน TGIX แนะนำมีการกำหนดค่าให้สอดคล้องกับวัตถุประสงค์และลักษณะการใช้งานหรือให้บริการของเว็บแอปพลิเคชัน โดยคำนึงถึงความปลอดภัยและการใช้งาน เพื่อให้ผู้ใช้สามารถดำเนินการภายในเว็บแอปพลิเคชันให้เสร็จสิ้นได้อย่างสะดวกสบายโดยที่เซสชันหมดอายุบ่อยครั้งจนเกินไป

เมื่อเซสชันหมดอายุ เว็บแอปพลิเคชันต้องดำเนินการเพื่อทำให้เซสชันเป็นโมฆะทั้งสองด้าน ทั้งไคลเอ็นต์และเซิร์ฟเวอร์

สำหรับกลไกการแลกเปลี่ยนเซสชันส่วนใหญ่ การดำเนินการฝั่งไคลเอ็นต์เพื่อทำให้ ID เซสชันใช้งานไม่ได้จะขึ้นอยู่กับการล้างค่าในโทเค็น ตัวอย่างเช่น หากต้องการทำให้คุกกี้ใช้งานไม่ได้ ข้อแนะนำคือ ให้ระบุค่าว่างสำหรับรหัสเซสชัน และตั้งค่าแอตทริบิวต์ Expires (หรือ Max-Age) เป็นวันที่ในอดีต

(2) การทำให้เซสชันหมดอายุอัตโนมัติ (Automatic Session Expiration)

หมดเวลาที่ไม่ได้ใช้งาน (Idle Timeout) ข้อแนะนำคือ เซสชันทั้งหมดควรมีการกำหนดค่าหมดเวลาเมื่อไม่มีการใช้งาน ค่าหมดเวลานี้เป็นการกำหนดระยะเวลาที่เซสชันจะยังคงอยู่และสามารถใช้งานได้ ในกรณีที่ไม่มีกิจกรรมในเซสชัน จะถูกทำให้เซสชันเป็นโมฆะเมื่อเลยช่วงเวลาที่กำหนดไว้โดยนับตั้งแต่คำร้องขอ HTTP ล่าสุดได้รับ จากเว็บแอปพลิเคชัน การกำหนดค่าหมดเวลาเมื่อไม่มีการใช้งานจะเป็นการจำกัดโอกาสที่ผู้โจมตีใช้ ID เซสชันที่ถูกต้องในการโจมตี หากผู้โจมตีสามารถขโมยเซสชันได้ มาตรฐาน TGIX แนะนำให้มีการกำหนดค่าหมดเวลาของเซสชันและการหมดอายุในฝั่งเซิร์ฟเวอร์

(3) การทำให้หมดอายุของเซสชันด้วยตนเอง (Manual Session Expiration)

เว็บแอปพลิเคชันจะต้องมีกลไกที่อนุญาตให้ผู้ใช้สามารถปิดเซสชันของตนได้ เมื่อใช้งานเว็บแอปพลิเคชันเสร็จแล้ว โดยเว็บแอปพลิเคชันต้องมีปุ่มล็อกเอาต์ ออกจากระบบที่มองเห็นและเข้าถึงได้ง่าย ซึ่งอยู่ในส่วนหัวหรือเมนูของแอปพลิเคชันเว็บ และสามารถเข้าถึงได้จากทุกหน้า เพื่อให้ผู้ใช้สามารถปิดเซสชันด้วยตนเองเวลาใดก็ได้

4.4.3.2 การล้างเซสชันการอนุญาต

โดยปกติแอปพลิเคชันผู้ให้บริการให้สิทธิ์จะมี ฟังก์ชันให้เรียกใช้งานอยู่แล้วเพื่อล้างเซสชัน ขึ้นอยู่กับเครื่องมือที่นำมาใช้ในการพัฒนาแอปพลิเคชัน การล้างเซสชันในส่วนนี้สามารถทำได้โดยจะต้องเรียกฟังก์ชันล้างเซสชันของแอปพลิเคชันผู้ให้บริการให้สิทธิ์

4.4.3.3 การล้างเซสชันผู้ให้บริการข้อมูลประจำตัว

สำหรับการล้างเซสชันในส่วนนี้ไม่มีความจำเป็นต้องนำเนินการใด สำหรับแอปพลิเคชันของผู้ขอใช้บริการ (Consumer) ตามมาตรฐาน TGIX แนะนำให้ปฏิตามขั้นตอนการล้างเซสชันระดับแอปพลิเคชัน และการล้างเซสชันการพิสูจน์ตัวตน ก็เพียงพอสำหรับการทำงาน ประกอบกับการกำหนดอายุของเซสชัน (Session Lifetime Limits) ผู้ให้บริการพิสูจน์และยืนยันตัวตน ในกรณีที่ผู้ใช้งานไม่ได้ทำการออกจากระบบ จะถูกทำให้ออกจากระบบโดยอัตโนมัติ เมื่อถึงเวลาที่กำหนด

4.4.4 การตรวจจับการโจมตีเซสชัน (Session Attacks Detection)

โดยปกติแอปพลิเคชันจะต้องมีการออกแบบ และพัฒนาโดยคำนึงถึงความปลอดภัยของแอปพลิเคชัน ดังนั้นฟังก์ชันการทำงานด้านความปลอดภัยพื้นฐานตามมาตรฐาน TGIX แนะนำควรจะมีดังต่อไปนี้

(1) การเดารหัสเซสชันและการตรวจจับการบังคับเซสชันที่ถูกต้อง (Session ID Guessing and Brute Force Detection)

ในกรณีที่ผู้โจมตีพยายามคาดเดาหรือบังคับ ID เซสชันที่ถูกต้อง จำเป็นต้องเรียกใช้คำร้องขอจำนวนหลายรายการกับเว็บแอปพลิเคชันเป้าหมายโดยใช้ ID เซสชันที่แตกต่างกันจากที่อยู่ IP Address เดียว นอกจากนี้ ยังรวมถึงผู้โจมตีพยายามวิเคราะห์การคาดการณ์ของ ID เซสชัน เช่น การใช้การวิเคราะห์ทางสถิติก็จำเป็นต้องเรียกใช้คำร้องขอจำนวนหลายรายการจากที่อยู่ IP Address เดียวกันเพื่อนำข้อมูลมาเปรียบเทียบและรวบรวมข้อมูลให้เพียงพอสำหรับการสร้างเซสชัน ID ใหม่ที่ถูกต้อง

ข้อแนะนำเว็บแอปพลิเคชันต้องมีฟังก์ชันที่สามารถตรวจพบทั้งสองสถานการณ์ได้ โดยตรวจสอบจากตามจำนวนครั้งที่พยายามร้องขอข้อมูล จะต้องมีการแจ้งเตือนและบล็อกที่อยู่ IP Address ที่ละเมิดได้

(2) การตรวจจับความผิดปกติของรหัสเซสชัน (Detecting Session ID Anomalies)

เว็บแอปพลิเคชันควรเน้นที่การตรวจจับความผิดปกติที่เกี่ยวข้องกับ ID เซสชัน แนะนำให้มีการพัฒนาเว็บแอปพลิเคชันโดยดูคำแนะนำจาก OWASP เป็นกรอบแนวทางและนำวิธีการต่างๆ มาปรับใช้เพื่อให้มีความสามารถในการตรวจจับการบุกรุก โดยเว็บแอปพลิเคชันควรเน้นไปที่การตรวจจับความผิดปกติและพฤติกรรมที่ไม่คาดคิด แทนที่จะใช้การป้องกันจากองค์ประกอบภายนอกเช่น fire wall บางครั้งรายละเอียดข้อมูลต่างๆ ที่ได้จากภายในเว็บแอปพลิเคชันเท่านั้นที่จะสร้างจุดสังเกต และตรวจจับที่เกี่ยวข้องกับเซสชันได้หลายจุด เช่น เมื่อมีการแก้ไขคุกกี้ สร้างคุกกี้ใหม่ การใช้รหัสเซสชันจากผู้ใช้รายอื่นซ้ำ หรือเมื่อ User-Agent เปลี่ยนแปลงไปจากเดิมในช่วงระหว่างการสื่อสาร

(3) การผูก ID เซสชันกับคุณสมบัติผู้ใช้อื่น (Binding the Session ID to Other User Properties)

ในการตรวจจับพฤติกรรมที่ไม่เหมาะสมของผู้ใช้และการขโมยเซสชัน ข้อแนะนำ ให้ผูก ID เซสชันกับผู้ใช้หรือคุณสมบัติของไคลเอ็นต์ เช่น IP Address ของไคลเอ็นต์, User-Agent, ใบรับรองดิจิทัล หากเว็บแอปพลิเคชันตรวจพบการเปลี่ยนแปลงหรือความผิดปกติของคุณสมบัติของไคลเอ็นต์ต่างๆ ในช่วงการสื่อสารของ

เซสชันที่สร้างขึ้นจะเป็นจุดสังเกตสำหรับการจัดการความพยายามในการขโมยบัญชี และสามารถใช้ข้อมูลนี้เพื่อแจ้งเตือนและยุติเซสชันที่ไม่ถูกต้อง การใช้คุณสมบัติเหล่านี้เพื่อป้องกันการโจมตีเซสชันเป็นเพียงการเพิ่มความสามารถในการตรวจจับ ของเว็บแอปพลิเคชันเท่านั้น

4.4.5 การป้องกันและจัดการเซสชันโดยใช้ Web Application Firewalls

ในกรณีที่การตรวจจับการบุกรุก โดยเว็บแอปพลิเคชันยังไม่สามารถครอบคลุมได้ เพื่อเป็นการเสริมการป้องกันเว็บแอปพลิเคชัน โดยมีเป้าหมายเพื่อให้เว็บแอปพลิเคชันมีความปลอดภัยมากขึ้น ข้อแนะนำให้ใช้การป้องกันภายนอก เช่น Web Application Firewalls (WAFs) ที่สามารถป้องกันภัยคุกคามการจัดการเซสชันที่อธิบายไว้แล้วข้างต้น โดยความสามารถในการตรวจจับ และป้องกันการโจมตีตามเซสชัน สำหรับ WAF เป็นการบังคับให้มีการใช้แอตทริบิวต์ความปลอดภัยบนคุกกี้ เช่น แฟล็ก Secure และ HttpOnly โดยใช้เป็นกฎพื้นฐานบนส่วนหัว Set-Cookie สำหรับการตอบกลับของเว็บแอปพลิเคชันทั้งหมดจะต้องมีการใช้แอตทริบิวต์ความปลอดภัยบนคุกกี้ ความสามารถของ WAF สามารถติดตามเซสชันและ ID เซสชันที่เกี่ยวข้อง และใช้ป้องกันการแก้ไขเซสชัน โดยตรวจสอบความสัมพันธ์ระหว่าง ID เซสชันและคุณสมบัติของไคลเอ็นต์ เช่น IP Address หรือ User-Agent หรือจัดการการหมดอายุของเซสชัน โดยบังคับให้ไคลเอ็นต์และเว็บแอปพลิเคชันสิ้นสุดเซสชัน

4.4.6 การจัดการเซสชันในสถาปัตยกรรม Stateless

แนวคิดของสำหรับการจัดการเซสชันในสถาปัตยกรรม Stateless สำหรับนักพัฒนาที่ใช้เซสชันการเก็บสถานะ เพื่อให้เห็นประโยชน์และสามารถพัฒนาได้อย่างไร นอกจากนี้ยังอธิบายรายละเอียดของ JWT ตามมาตรฐาน OAuth 2.0 ซึ่งจะกล่าวถึงในหัวข้อถัดไป

เนื่องจากการรับรองความถูกต้องจะต้องมีการเก็บสถานะเป็นเวลานาน สำหรับกรณี stateful เมื่อผู้ใช้ทำการเข้าใช้งานจะต้องป้อนข้อมูลประจำตัว จากนั้นแอปพลิเคชันจะสร้าง ID เซสชันที่ไม่ซ้ำกัน เก็บไว้ที่เซิร์ฟเวอร์ และส่งคืน ID เซสชันกลับให้ผู้ใช้ โดยข้อมูลรายละเอียดผู้ใช้ทั้งหมดจะถูกเก็บไว้ที่เซิร์ฟเวอร์ ทุกบริการที่ต้องใช้ข้อมูลบางอย่างเกี่ยวกับผู้ใช้จะต้องติดต่อกับฐานข้อมูลที่จัดเก็บข้อมูล ซึ่งวิธีการนี้มีข้อดี ในกรณีที่ข้อมูลผู้ใช้ถูกรวมศูนย์ ทำให้ยากต่อการปลอมแปลงข้อมูลได้ และข้อมูลยังปรับปรุงให้ทันสมัยอยู่เสมอ ทุกอย่างถูกเก็บไว้ในฐานข้อมูลศูนย์กลาง สำหรับสถาปัตยกรรมรูปแบบ stateful การดึงข้อมูลจากส่วนกลางเพื่อดำเนินการบางอย่างอาจก่อให้เกิดปัญหา button neck ได้ ในกรณีที่มีความต้องการในการเข้าถึงข้อมูลจำนวนมากการดำเนินการทั้งหมดที่กล่าวมาจะไม่เกิดปัญหา หากการพิสูจน์ตัวตนและการอนุญาตเป็นแบบ stateless เนื่องจากการร้องขอแต่ละครั้งจะมีข้อมูลที่จำเป็นทั้งหมดที่อยู่แล้ว โดยการสร้าง “รหัสเซสชัน” พิเศษซึ่งเป็นผลมาจากการเข้ารหัสข้อมูลด้วยรหัสลับที่เก็บไว้เซิร์ฟเวอร์ ทำให้สามารถส่งข้อมูลนั้นให้กับ

ผู้ใช้งาน โดยไม่ต้องกังวลว่าจะถูกแก้ไขข้อมูล หรือกล่าวอีกนัยหนึ่งคือ สามารถรักษาคุณลักษณะของข้อมูลในรหัสเซสชันไว้เหมือนเดิม และในขณะเดียวกันก็สามารถที่จะเพิ่มข้อมูลต่าง ๆ เพิ่มเติมได้ ซึ่งทำให้เข้าถึงข้อมูลฝั่งเซิร์ฟเวอร์ไม่จำเป็นต้องมีการดึงข้อมูลจากฐานข้อมูลหรือที่จัดเก็บข้อมูลอีก เซิร์ฟเวอร์ทำแค่ถอดรหัสข้อมูลที่มีอยู่ใน "รหัสเซสชัน" นั้น จากนี้ไปจะกล่าวถึงองค์ประกอบเหล่านี้ว่าเป็นโทเค็น ซึ่งจะกล่าวถึงในหัวข้อถัดไป

ข้อดีของ stateless คือ สามารถใช้ทั้งสองวิธีพร้อมกันได้ หากปัจจุบันสถาปัตยกรรมที่ใช้รหัสเซสชันสามารถเพิ่ม JWT ลงไปได้ หรือสามารถฝังรหัสเซสชันในโทเค็นและให้ API Gateway ทำการดึงข้อมูลเหล่านั้นแล้วส่งต่อไปยังผู้บริการต่อไปได้ สามารถเพิ่มการรับรองความถูกต้องสมัยใหม่ให้กับแอปพลิเคชันรุ่นเก่า โดยการปรับให้มีการสร้าง JWT เป็นคุกกี้ กล่าวอีกนัยหนึ่งคือการใช้ JWT เป็นรหัสเซสชันที่ไม่ซ้ำกันแทน ID เซสชันเดิม ในกรณีที่สถาปัตยกรรมอนุญาตให้ใช้รูปแบบ ID เซสชันได้หลากหลาย วิธีการนี้เป็นตัวเลือกที่ดีสำหรับนักพัฒนา

4.5 การบริหารจัดการ Token

4.5.1 มาตรฐานการสร้าง Token

การให้บริการข้อมูล ผู้ให้บริการจะต้องมีการกำหนดการรูปแบบการรักษาความปลอดภัยในการรับส่งข้อมูล ซึ่งมีมาตรฐานที่แตกต่างกันมาก ยกตัวอย่างเช่น การใช้การลงนามข้อความตาม OAuth 1.0 การตรวจสอบสิทธิ์และการอนุญาตที่ใช้ OAuth 2.0 ซึ่งมาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ด้านการเชื่อมโยงข้อมูล ได้กำหนดให้ใช้รูปแบบการตรวจสอบสิทธิ์และการอนุญาตที่ใช้ Open ID Connect 1.0 (OIDC) มาตรฐานดังกล่าวได้มีการกล่าวถึงโทเค็น และไอดีโทเค็น ซึ่งรายละเอียดของโทเค็นและไอดีโทเค็นได้อ้างถึงใน มาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ด้านการเชื่อมโยงข้อมูล เรื่อง ข้อกำหนดด้านการยืนยันตัวตน การกำหนดสิทธิ์ และบัญชีการใช้งาน โดยทั้งโทเค็น และไอดีโทเค็นได้มีการกำหนดรูปแบบการรักษาความปลอดภัยสำหรับโทเค็น โดยอ้างอิงตามมาตรฐาน JSON Web Tokens (JWT): RFC-7519 [4]

มาตรฐาน JSON Web Token (JWT): RFC-7519 [4] ซึ่งมีข้อกำหนดที่กะทัดรัดและครอบคลุมเหมาะสำหรับการรับส่งข้อมูลเจสันออบเจ็ค เนื่องจากมีขนาดค่อนข้างเล็ก จึงสามารถส่งผ่านพารามิเตอร์ POST หรือสามารถส่งภายในส่วนหัว HTTP ได้ อีกทั้ง JWT มีข้อมูลที่จำเป็นเพียงพอที่ผู้รับตรวจสอบความถูกต้องของโทเค็นได้โดยไม่ต้องเรียกเซิร์ฟเวอร์ และเพื่อหลีกเลี่ยงการสืบค้นฐานข้อมูลมากกว่าหนึ่งครั้งสำหรับการเรียกดูข้อมูลที่เกี่ยวกับเอนติตี้ ประโยชน์ของการใช้งานมาตรฐาน JWT มีดังต่อไปนี้

(1) JWT มีขนาดที่เล็กกว่าเมื่อเทียบกับ โทเค็น SAML ซึ่งเป็นโทเค็นที่เกิดจากการนำ XML มาผ่านกระบวนการแฮชและการลงลายมือชื่อดิจิทัล เนื่องจาก JSON มีความละเอียดของส่วนขยายน้อยกว่า XML ดังนั้นเมื่อมีการเข้ารหัส JWT จะมีขนาดที่เล็กกว่าโทเค็น SAML สิ่งนี้ทำให้ JWT เป็นตัวเลือกที่ดีในการส่งผ่านโปรโตคอล Hyper Test Transfer Protocol

(2) มีความปลอดภัยสูง JWT สามารถใช้คู่คีย์สาธารณะ/ส่วนตัวในรูปแบบของใบรับรอง X.509 สำหรับการลงนาม JWT ยังสามารถเซ็นชื่อด้วยคีย์แบบสมมาตรโดยข้อมูลลับที่ใช้ร่วมกันโดยใช้อัลกอริธึม HMAC

(3) มีความนิยมใช้งานกันอย่างแพร่หลาย เนื่องจากภาษาที่ใช้ในการพัฒนาในปัจจุบันรองรับรูปแบบข้อมูลเจสันออบเจ็คอยู่แล้วจึงทำให้ใช้งานได้ง่ายกว่าเมื่อเทียบกับ XML

โครงสร้างของ โทเค็นตามมาตรฐาน JSON Web Token (JWT) จะประกอบไปด้วย 3 ส่วนหลักๆ และขั้นด้วย “.” โดยมีลักษณะดังนี้

```
[header].[payload].[signature]
```

ส่วนที่ 1 Header เป็นส่วนที่บอกรายละเอียดของ JWT ส่วนมากจะประกอบไปด้วย algorithm (alg), type(typ), Key ID(kid) ตัวอย่างดังรูปที่ 6

รูปที่ 6 แสดงตัวอย่าง Header ของ JWT

ส่วนที่ 2 Payload เป็นส่วนที่เก็บข้อมูลเบื้องต้นของผู้ใช้งาน และข้อมูลเพิ่มเติมที่ต้องการโดยการเก็บข้อมูลจะเป็นแบบเจสันออบเจ็ค ประกอบไปด้วย Issuer(iss), Audience(aud), Expiration Time(exp), nonce, Access Token hash value(at_hash), Code hash value(c_hash), Issued At(iat), Time when the authentication occurred(auth_time), not before(nbf), Subject(sub), JWT ID(jti) รายละเอียดความหมายของแต่ละฟิลด์สามารถอ่านเพิ่มเติมได้ที่ภาคผนวก ก.

รูปที่ 7 แสดงตัวอย่าง Payload ของ JWT

ส่วนที่ 3 Signature เป็นส่วนที่เกิดจากการนำเอา ส่วนของ Header และ Payload มาทำการเข้ารหัสด้วยวิธีการ Base64 ของแต่ล่ะส่วนจากนั้นนำเอามาต่อกันและขั้นด้วยจุด จากนั้นนำไปเข้ารหัสด้วยวิธีการที่กำหนดอยู่ในส่วนของ Header จากนั้นจะได้ค่าเอาต์พุตให้นำไปเข้ารหัสด้วยวิธีการ Base64

4.5.2 ข้อกำหนดสำหรับพารามิเตอร์ตามมาตรฐาน JWT

ข้อกำหนดในส่วนนี้จะเป็นการอธิบายถึงรายละเอียดของแต่ละฟิลด์ที่มีการใช้งานหรือเป็นทางเลือกสำหรับการใช้งานตามมาตรฐาน TGIX ที่อ้างอิงจากมาตรฐานของ JWT โดยจะอธิบายรายละเอียดและระบุข้อกำหนดเพิ่มเติมเฉพาะส่วนของ TGIX สำหรับการใช้งานส่วนอื่นที่ไม่ได้กล่าวถึงให้ยึดตามมาตรฐาน JWT

ฟิลด์ที่จำเป็นสำหรับส่วน Header (Required Headers) ได้แก่ Algorithm(alg) เป็นพารามิเตอร์ส่วนหัวของ JWT ที่ระบุถึงข้อมูลอัลกอลิธึมในการเข้ารหัสข้อมูลที่ใช้ใน JWT ข้อกำหนดตามมาตรฐาน TGIX กำหนดให้ใช้งานอัลกอลิธึม ดังต่อไปนี้ "RS256" ตามมาตรฐาน JSON Web Algorithms (JWA): RFC-7518 [5] โดยข้อมูลอัลกอลิธึมทั้งหมดสามารถอ้างอิงจากได้จาก มาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐด้านการเชื่อมโยงข้อมูล เรื่องข้อกำหนดของความน่าเชื่อถือและความมั่นคงปลอดภัย

4.5.3 การใช้งาน JWT สำหรับมาตรฐาน TGIX

การประยุกต์ใช้งาน JWT ใน มาตรฐาน TGIX นั้น นอกจากข้อกำหนดต่างๆ ที่ได้กล่าวไปข้างต้น มาตรฐานได้มีการกล่าวถึงขั้นตอนหรือรูปแบบการใช้งาน ต่างๆ ที่เหมาะสม โดยในเนื้อหาส่วนนี้จะกล่าวถึงกระบวนการต่างๆ ที่มีความจำเป็นต้องปฏิบัติตาม เพื่อเป็นแนวทางให้นักพัฒนานำไปพัฒนาระบบงานได้อย่างถูกต้องตรงตามมาตรฐานการรักษาความปลอดภัยในการรับส่งข้อมูลโดยแยกเป็นหัวข้อต่างๆ ดังนี้

4.5.3.1 การตรวจสอบโทเค็น (Validate JSON Web Tokens)

การตรวจสอบโทเค็นมาตรฐาน TGIX แนะนำให้ผู้ให้บริการจะต้องทำการตรวจสอบโทเค็นที่ได้รับมาเสมอ โดยการตรวจสอบนั้นสามารทำได้หลายวิธีขึ้นอยู่กับวิธีการพัฒนาและภาษาที่ใช้ในการพัฒนา มาตรฐาน TGIX เปิดกว้างให้ผู้ให้บริการสามารถเลือกเทคโนโลยีและภาษาในการพัฒนาระบบให้บริการข้อมูลได้ตามความชำนาญของผู้ให้บริการ การพัฒนาฟังก์ชันการตรวจสอบสามารถแยกออกเป็น 3 กลุ่มดังนี้

(1) ผู้ให้บริการใช้งานจากฟังกชันพื้นฐานที่มีใน Framwork ที่ใช้งาน

เนื่องจากปัจจุบัน Framwork ที่เป็นที่นิยมใช้งานต่างก็มีฟังก์ชันรองรับการตรวจสอบโทเค็น และเป็นไปตามมาตรฐาน JSON Web Token (JWT): RFC-7519 [4] มาตรฐาน TGIX อนุญาตให้นักพัฒนาของผู้ให้บริการสามารถเลือกใช้งานได้ตามความเหมาะสม

(2) ใช้ Third-Party Libs ในการพัฒนาฟังก์ชันการตรวจสอบ

ในกรณีที่ผู้พัฒนาต้องการพัฒนาฟังก์ชันการตรวจสอบโทเค็น สามารถดาวน์โหลด Libs ที่ช่วยในการตรวจสอบโทเค็นได้จากเว็บ www.jwt.org ซึ่งมี Libs ที่ถูกพัฒนาในภาษาต่างๆ

มาตรฐาน TGIX แนะนำให้นักพัฒนาใช้เกณฑ์ในการเลือกใช้งานโดย ต้องขึ้นอยู่กับภาษาที่ใช้งานและอัลกอลิธึมที่เป็นไปตามข้อกำหนดของ TGIX

(3) ผู้ให้บริการทำการพัฒนาการตรวจสอบด้วยตนเอง

โทเค็นที่มีการใช้งานในมาตรฐาน TGIX นั้นอ้างอิงตามมาตรฐาน JWT ผู้พัฒนาระบบงานให้บริการต้องพัฒนาฟังก์ชันการตรวจสอบโดยทำตามมาตรฐาน JSON Web Algorithms (JWA): RFC-7518 [5] หัวข้อ 7.2 การตรวจสอบ JWT ให้ครบถ้วน

หัวข้อการตรวจสอบโทเค็น นอกจากจะทำการตรวจสอบโทเค็นแล้ว ตามมาตรฐาน TGIX แนะนำให้ผู้ให้บริการจะต้องทำการตรวจสอบเคลมด้วย โดยให้ทำการตรวจสอบโทเค็นออเดียน(Token audience) และ Nonce สำหรับ Implicit Flow

การตรวจสอบโทเค็นออเดียน (Token audience) ค่านี้ต้องตรงกับรหัสไคลเอ็นต์ของแอปพลิเคชันตามที่กำหนดไว้ในการตั้งค่าแอปพลิเคชัน

การตรวจสอบ nonce แนะนำให้ส่ง nonce ในคำขอโทเค็น เพื่อช่วยป้องกันการโจมตีซ้ำ โดยค่า nonce ในโทเค็นต้องตรงกับ nonce เดิมที่ส่งในคำขอ

4.5.4 การถอดถอนโทเค็น (Revoke Token)

เนื่องจากเมื่อมีการออกโทเค็นแล้ว ทั้งโทเค็นการเข้าถึง (Access Token) และโทเค็น ID (ID Token) จะไม่สามารถเพิกถอนได้เหมือน ID เซสชัน (Session ID) สำหรับฝั่งเซิร์ฟเวอร์ ดังนั้นเพื่อเหตุผลด้านความปลอดภัยจึงต้องมีการจำกัดเวลาที่ค่อนข้างสั้น และให้ใช้กระบวนการรีเฟรชโทเค็นเป็นระยะแทน แต่อย่างไรก็ตามส่วนของการถอดถอนโทเค็นก็ยังสามารถทำได้โดยการถอดถอนรีเฟรชโทเค็นแทน จะทำให้ Client นั้นไม่สามารถทำการต่ออายุโทเค็นได้ ตามมาตรฐาน TGIX แนะนำให้มีการพัฒนาฟังก์ชันการถอดถอนรีเฟรชโทเค็นเพื่อใช้ในกรณีที่ต้องการยกเลิกการใช้งานโทเค็นนั้น

4.5.5 การรีเฟรชโทเค็น (Refresh Token)

ในกรณีที่โทเค็นการเข้าถึงหมดอายุ จะต้องมีกระบวนการในการรีเฟรชโทเค็นที่เหมาะสมเช่น ไม่ควรเรียกปลายทางเพื่อรับโทเค็นการเข้าถึงใหม่ทุกครั้ง ในการขอรีเฟรชโทเค็น ให้ส่งคำขอ POST ไปยังปลายทาง /oauth/token โดยระบุ grant_type=refresh_token เพื่อเป็นการเปลี่ยนโทเค็นการเข้าถึง ตามมาตรฐาน TGIX แนะนำให้รีเฟรชโทเค็นต้องมีการกำหนดอายุการใช้งาน โดยจะขึ้นอยู่กับผู้ให้บริการผู้พิสูจน์และยืนยันตัวตน (Identity Provider) โดยค่าของอายุการใช้งานสามารถกำหนดได้ตั้งแต่ 30 วันจนถึง 6 เดือน

ภาคผนวก ก. รายละเอียดฟิลด์ตามมาตรฐาน JWT

ตารางที่ 7 รายละเอียดฟิลด์ตามมาตรฐาน JWT

ฟิลด์ของเฮดเดอร์		
ฟิลด์	คำอธิบายรายละเอียด	การใช้งาน
“alg”	อัลกอลิธึม เป็นพารามิเตอร์ส่วนหัวของ JWT ที่ระบุถึงข้อมูลอัลกอลิธึมในการเข้ารหัสข้อมูลที่ใช้ใน JWT ข้อกำหนดตามมาตรฐาน TGIX กำหนดให้ใช้งานอัลกอลิธึม ดังต่อไปนี้ "RS256" ตามมาตรฐาน JSON Web Algorithms (JWA): RFC-7518 [5] โดยข้อมูลอัลกอลิธึมทั้งหมดอ้างอิงจาก มาตรฐานการเชื่อมโยงและแลกเปลี่ยนข้อมูลภาครัฐ ด้านการเชื่อมโยงข้อมูล เรื่องข้อกำหนดของความน่าเชื่อถือและความมั่นคงปลอดภัย	ฟิลด์จำเป็นสำหรับเฮดเดอร์
“typ”	ไทป์ (type) เป็นพารามิเตอร์ส่วนหัวของ JWT ข้อกำหนดตามมาตรฐาน ให้ระบุเป็นค่า “JWT” ตามมาตรฐาน JSON Web Token (JWT): RFC-7519 [4]	ฟิลด์ทางเลือกสำหรับเฮดเดอร์
“kid”	คีย์ไอดี (key ID) เป็นพารามิเตอร์ส่วนหัวของ JWT โดยเป็นค่าที่บ่งบอกว่าใช้กุญแจไหน ในการเข้ารหัส โทเค็น JWT ตามมาตรฐาน JSON Web Signature (JWS): RFC-7515 [6]	ฟิลด์ทางเลือกสำหรับเฮดเดอร์
ฟิลด์ของเคลม		
“iss”	ผู้ออกสิทธิ์ (Issuer) เป็นค่าที่ระบุตัวตนผู้ที่ทำการรับรองการอ้างสิทธิ์ โดยค่าข้อมูลเป็นแบบ case sensitive URL ที่แสดงถึงรูปแบบ เช่น host, port number และ path	ฟิลด์จำเป็นสำหรับเคลม
“aud”	ผู้ชม (Audience) กลุ่มเป้าหมายที่ ID Token นี้มีไว้สำหรับ ต้องมี OAuth 2.0 client_id ของ Relying Party เป็นค่าผู้ชม นอกจากนี้ยังอาจมีตัวระบุสำหรับผู้ชมอื่นๆ ในกรณีทั่วไป ค่า aud คืออาร์เรย์ของสตริงที่คำนึงถึงขนาดตัวพิมพ์ ในกรณีพิเศษทั่วไปเมื่อมีผู้ชมหนึ่งราย ค่า aud อาจเป็นสตริงที่ละเอียดอ่อนตัวพิมพ์เล็กและตัวพิมพ์ใหญ่	ฟิลด์จำเป็นสำหรับเคลม

ตารางที่ 7 รายละเอียดฟิลด์ตามมาตรฐาน JWT (ต่อ)

ฟิลด์	คำอธิบายรายละเอียด	การใช้งาน
“exp”	เวลาหมดอายุ (expiration time) ระบุเวลาหมดอายุในหรือหลังจากนั้น JWT ต้องไม่ได้รับการยอมรับสำหรับการประมวลผล การประมวลผลการอ้างสิทธิ์ "exp" กำหนดให้วันที่/เวลาปัจจุบันต้องอยู่ก่อนวันที่/เวลาหมดอายุที่ระบุไว้ในการอ้างสิทธิ์ "exp" ผู้ดำเนินการอาจให้เวลาเล็กน้อย โดยปกติไม่เกินสองสามนาที เพื่อพิจารณา Leap Seconds ค่าจะต้องเป็นตัวเลขที่มีค่า NumericDate ตามมาตรฐาน JSON Web Token (JWT): RFC-7519 [4] หมายเหตุ: NumericDate จะมีการใช้งานโดยพารามิเตอร์ exp, iat, และข้อมูลส่วนอื่นๆ ที่เกี่ยวข้องกับเคลม โดยค่าข้อมูลนี้จะนำเสนอจำนวนตัวเลขหน่วยเป็นวินาทีเริ่มนับจากเวลาเริ่มต้นคือ วันที่ 1970-01-01T00:00:00Z UTC จนถึงปัจจุบันไม่นับรวม leap seconds	ฟิลด์จำเป็นสำหรับเคลม
“nonce”	นอนซ (nonce) ค่าสตริงที่ใช้ในการเชื่อมโยงเซสชันไคลเอ็นต์กับโทเค็น ID และเพื่อลดการโจมตีซ้ำ (Replay Attacks) ค่าจะถูกส่งผ่านแบบไม่แก้ไขจากคำขอการตรวจสอบสิทธิ์ไปยัง ID Token หากมีอยู่ใน ID Token ลูกค้าต้องตรวจสอบว่า nonce Claim Value เท่ากับค่าของพารามิเตอร์ nonce ที่ส่งในคำขอการตรวจสอบสิทธิ์ หากมีอยู่ในคำขอการตรวจสอบสิทธิ์ เซิร์ฟเวอร์การให้สิทธิ์ต้องมีการอ้างสิทธิ์ nonce ในโทเค็น ID โดยมีค่าการอ้างสิทธิ์เป็นค่า nonce ที่ส่งในคำขอการตรวจสอบสิทธิ์ เซิร์ฟเวอร์การให้สิทธิ์ไม่ควรดำเนินการอื่นใดกับค่า nonce ที่ใช้ ค่า nonce เป็นสตริงที่คำนึงถึงขนาดตัวพิมพ์	ฟิลด์จำเป็นสำหรับเคลม
“at_hash”	ค่าแฮชโทเค็น (Access Token hash value) คือการเข้ารหัส base64 ของครึ่งซ้ายสุดของแฮช โดยที่อัลกอริธึมแฮชที่ใช้ คืออัลกอริธึมแฮชที่ใช้ในพารามิเตอร์ส่วนหัว alg ของส่วนหัว JOSE ของ ID Token ตัวอย่างเช่น ถ้า alg เป็น RS256 ให้แฮชค่า access_token ด้วย SHA-256 จากนั้นใช้ 128 บิตที่อยู่ทางซ้ายสุดและ base64url เข้ารหัสค่า at_hash เป็นสตริงที่คำนึงถึงขนาดตัวพิมพ์	ฟิลด์ทางเลือกสำหรับเคลม

ตารางที่ 7 รายละเอียดฟิลด์ตามมาตรฐาน JWT (ต่อ)

ฟิลด์	คำอธิบายรายละเอียด	การใช้งาน
“c_hash”	ค่าแฮชโค๊ด (Code hash value) คือการเข้ารหัส base64url ของครึ่งซ้ายสุดของแฮช โดยที่อัลกอริธึมแฮชที่ใช้ คืออัลกอริธึมแฮชที่ใช้ในพารามิเตอร์ alg Header ของส่วนหัว JOSE ของ ID Token ตัวอย่างเช่น ถ้า alg เป็น HS512 ให้แฮชค่าโค้ดด้วย SHA-512 จากนั้นใช้ 256 บิตที่อยู่ทางซ้ายสุดและ base64 เข้ารหัสค่า c_hash เป็นสตริงที่คำนึงถึงขนาดตัวพิมพ์	ฟิลด์ทางเลือกสำหรับเคลม
“iat”	(Issued At) เป็นค่าที่ระบุเวลาที่ออก JWT สามารถใช้เพื่อกำหนดอายุของ JWT ค่าจะต้องเป็นตัวเลขที่มีค่า NumericDate ใช้การอ้างสิทธิ์นี้เป็นทางเลือก หมายเหตุ: NumericDate จะมีการใช้งานโดยพารามิเตอร์ exp, iat, และข้อมูลส่วนอื่นๆ ที่เกี่ยวข้องกับเคลม โดยค่าข้อมูลนี้จะนำเสนอจำนวนตัวเลขหน่วยเป็นวินาทีเริ่มนับจากเวลาเริ่มต้นคือ วันที่ 1970-01-01T00:00:00Z UTC จนถึงปัจจุบันไม่นับรวม leap seconds	ฟิลด์ทางเลือกสำหรับเคลม
“auth_time”	เวลาที่เกิดการตรวจสอบสิทธิ์ผู้ใช้ (Time when the authentication occurred) คือค่าจำนวนตัวเลขหน่วยเป็นวินาทีเริ่มนับจากเวลาเริ่มต้นคือ วันที่ 1970-01-01T00:00:00Z UTC จนถึงเวลาที่มีการร้องขอการตรวจสอบสิทธิ์ผู้ใช้	ฟิลด์ทางเลือกสำหรับเคลม
“nbf”	ก่อนเวลาที่กำหนด (not before) เป็นการระบุเวลา ถ้าเหตุการณ์เกิดขึ้นก่อนเวลาที่กำหนด JWT ต้องไม่ได้รับการยอมรับสำหรับการประมวลผล การประมวลผลคำร้อง "nbf" กำหนดให้วันที่/เวลาปัจจุบันต้องอยู่หลังหรือเท่ากับวันที่/เวลาก่อนหน้าที่ระบุไว้ในการอ้างสิทธิ์ "nbf"	ฟิลด์ทางเลือกสำหรับเคลม
“sub”	หัวเรื่อง (Subject) ระบุหัวเรื่องของ JWT โดยไม่ซ้ำกัน	ฟิลด์ทางเลือกสำหรับเคลม

ตารางที่ 7 รายละเอียดฟิลด์ตามมาตรฐาน JWT (ต่อ)

ฟิลด์	คำอธิบายรายละเอียด	การใช้งาน
“jti”	เจดับเบิลยูไอดี (JWT ID) เป็นค่าที่ใช้สำหรับระบุตัวระบุเฉพาะสำหรับ JWT ค่าตัวระบุต้องถูกกำหนดในลักษณะที่ทำให้มั่นใจว่ามีความเป็นไปได้เล็กน้อยที่ค่าเดียวกัน หากแอปพลิเคชันมีการใช้ผู้ออกหลายรายจะต้องป้องกันการซ้ำกันระหว่างค่าที่สร้างโดยผู้ออกที่แตกต่างกันเช่นกัน การอ้างสิทธิ์ "jti" สามารถใช้เพื่อป้องกันไม่ให้เล่น JWT ซ้ำ ค่า "jti" เป็นสตริงที่คำนึงถึงขนาดตัวพิมพ์ ตามมาตรฐาน JSON Web Token (JWT): RFC-7519 [4] หัวข้อ 4.1.7	ฟิลด์ทางเลือก สำหรับเคลม

บรรณานุกรม

[1] E. Nebel. (1995,พฤศจิกายน). Form-based File Upload in HTML. [ออนไลน์]. เข้าถึงได้จาก: https://www.ietf.org/rfc/rfc1867.txt. (วันที่ค้นข้อมูล: 9 กันยายน 2021)

[2] R. Fielding. (2014, มิถุนายน). Hypertext Transfer Protocol (HTTP/1.1): Range Requests. [ออนไลน์]. เข้าถึงได้จาก: https://www.rfc-editor.org/rfc/rfc7233#section-4.2. (วันที่ค้นข้อมูล: 9 กันยายน 2021)

[3] Hypertext Transfer Protocol -- HTTP/1.1. (1999, มิถุนายน). [ออนไลน์]. เข้าถึงได้จาก: https://www.w3.org/Protocols/rfc2616/rfc2616.html. (วันที่ค้นข้อมูล: 9 กันยายน 2021)

[4] M. Jones. (2015,พฤษภาคม) JSON Web Token (JWT). [ออนไลน์]. เข้าถึงได้จาก: https://datatracker.ietf.org/doc/html/rfc7519#section-7.2. (วันที่ค้นข้อมูล: 9 กันยายน 2021)

[5] M. Jones. (2015,พฤษภาคม) JSON Web Algorithms (JWA). [ออนไลน์]. เข้าถึงได้จาก: https://datatracker.ietf.org/doc/html/rfc7518. (วันที่ค้นข้อมูล: 9 กันยายน 2021)

[6] M. Jones. (2015, พฤษภาคม) JSON Web Signature (JWS). [ออนไลน์]. เข้าถึงได้จาก: https://datatracker.ietf.org/doc/html/rfc7515. (วันที่ค้นข้อมูล: 9 กันยายน 2021)