โมเดลการตรวจสอบสิทธิ์การเข้าถึงในห่วงโซ่การเรียกบริการแบบประสานงานภายใต้สภาพแวดล้อมของการเช่าหลายรายในการบริการบนคลาวน์

โดย

นาย ดนัย ทองแสง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์
ปีการศึกษา 2558

โมเดลการตรวจสอบสิทธิ์การเข้าถึงในห่วงโซ่การเรียกบริการแบบประสานงานภายใต้สภาพแวดล้อมของการเช่าหลายรายในการบริการบนคลาวน์

โดย

นายดนัย ทองแสง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิทยาการคอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์
ปีการศึกษา 2558

Calling Chain Coordination In Multi-tenant Authorization Model for Collaborative Cloud Services

BY

Mr. Danai Thongsang

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER DEGREE OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF SCIENCE AND TECHNOLOGY
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2015

มหาวิทยาลัยธรรมศาสตร์
คณะวิทยาศาสตร์และเทคโนโลยี

วิทยานิพนธ์

ของ

นายดนัย ทองแสง

เรื่อง

โมเดลการตรวจสอบสิทธิ์การเข้าถึงในห่วงโซ่การเรียกบริการแบบประสานงาน
ภายใต้สภาพแวดล้อมของการเช่าหลายรายในการบริการบนคลาวน์

ได้รับการตรวจสอบและอนุมัติ ให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต

เมื่อ วันที่ 27 ธันวาคม พ.ศ. 2558

ประธานกรรมการสอบวิทยานิพนธ์ _____
(ดร. กษิดิศ ชาญเชี่ยว)

กรรมการและอาจารย์ที่ปรึกษาวิทยานิพนธ์ _____
(รองศาสตราจารย์ ดร. เยาวดี เต็มธนาภัทร์)

กรรมการสอบวิทยานิพนธ์ _____
(ผู้ช่วยศาสตราจารย์ ดร. ทรงศักดิ์ รองวิริยะพานิช)

กรรมการสอบวิทยานิพนธ์ _____
(ผู้ช่วยศาสตรจารย์ ดร.เบญจพร ลิ้มธรรมาภรณ์)

คณบดี _____
(รองศาสตราจารย์ ปกรณ์ เสริมสุข)

หัวข้อวิทยานิพนธ์	โมเดลการตรวจสอบสิทธิ์การเข้าถึงในห่วงโซ่การเรียกบริการแบบประสานงานภายใต้สภาพแวดล้อมของการเช่าหลายรายในการบริการบนคลาวน์
ชื่อผู้เขียน	นายดนัย ทองแสง
ชื่อปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา/คณะ/มหาวิทยาลัย	สาขาวิทยาการคอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รองศาสตราจารย์ ดร.เยาวดี เต็มธนาภัทร์
ปีการศึกษา	2558

บทคัดย่อ

ปัจจุบันการบริการบนคลาวน์ได้ถูกใช้งานกันอย่างแพร่หลายมากขึ้น แต่กระบวนการการควบคุมสิทธิ์ในการเข้าถึงยังถูกจำกัด และมีการผูกเงื่อนไขในการกำหนดสิทธิ์แยกจากกันระหว่างการเช่าอย่างเด็ดขาด เพื่อให้การใช้ประโยชย์สูงสุดบนคลาวน์ การเช่าของผู้เช่าหลายรายนั้นก็มีระดับความเชื่อที่แตกต่างกันในการทำงานหรือใช้ทรัพยากรร่วมกัน

อย่างไรก็ตามการสร้างแอพลิเคชันที่มีการทำงานร่วมกันเป็นลักษณะของห่วงโซ่ คือมีการเรียกบริการระหว่างกันไปมา ที่ถูกพัฒนาบนระบบคลาวน์เดียวหรือหลายตัวได้พบความท้าทายในประเด็นเรื่องการควบคุมการเข้าถึง และจุดนี้เองได้เป็นประเด็นสำคัญของปัญหาในงานวิจัยชิ้นนี้ เพื่อให้เกิดความเหมาะสมในการตรวจสอบสิทธิ์การเข้าถึงแบบห่วงโซ่บริการ งานวิจัยชิ้นนี้นำเสนองานที่ถูกเพิ่มเติมต่อจาก “Multi-Tenant Authorization System Model (MTAS)” โดยใช้ชื่อว่า “โมเดลกระบวนการเพื่อตรวจสอบสิทธิ์การเข้าถึงในห่วงโซ่การเรียกบริการแบบประสานงานภายใต้สภาพแวดล้อมของการเช่าหลายรายในการบริการบนคลาวน์”

ในโมเดล MTAS การเรียกบริการในรูปแบบห่วงโซ่นั้นจำเป็นต้องแตกบทบาทของผู้เช่าออกเป็นบทบาทย่อย ๆ ในการกำหนดความเชื่อถือระหว่างผู้เช่าด้วยกัน ซึ่งทำให้จำนวนบทบาทถูกเพิ่มขึ้นอย่างมากโดยไม่ได้ตั้งใจ และบทบาทดังกล่าวนั้น ผู้ที่กำหนดขึ้นคือเจ้าของซอฟต์แวร์ในการใช้ซอฟต์แวร์นั้นแต่ละคน โดยแยกออกจากกันอย่างเด็ดขาด ดังนั้นลักษณะของความสัมพันธ์ของบทบาทหนึ่งสามารถใช้งานได้หลายการเช่าซอฟต์แวร์จึงผิดความหมาย

งานวิจัยนี้จึงเสนอให้มีการแยกองค์ประกอบของผู้เช่าออกมาเป็นอีกองค์ประกอบหนึ่ง เพื่อสร้างความชัดเจนและง่ายในการกำหนดบทบาทและสิทธิ์ของการเข้าถึงด้วยเรียกโมเดลใหม่นี้ว่า โมเดลการเรียกใช้บริการห่วงโซ่แบบประสานงานบนการทำงานในสภาพแวดล้อมของหลายผู้เช่า” (Chain of Calling Coordination in Multi-Tenancy Authorization: C-MTAS) โดยจุดเด่นของงานวิจัยชิ้นนี้เมื่อเทียบกับโมเดล MTAS นั้นผู้วิจัยได้ใช้สถานการณ์จำลองในการเปรียบเทียบการกำหนดบทบาทสำหรับสิทธิ์แบบเดียวกัน ซึ่งจะเห็นได้ว่าโมเดล C-MTAS นั้นมี

จำนวนบทบาทที่ต้องกำหนดน้อยกว่าในโมเดล MTAS นอกจากนี้ในงานวิจัยชิ้นนี้ยังนำโมเดลที่ออกแบบไว้มาอธิบายนโยบายตัวอย่างในรูปแบบของ XACML เพื่อสร้างเป็นระบบต้นแบบผู้วิจัยได้สร้างขึ้นเป็นแพลตฟอร์มการให้บริการในส่วนของการตรวจสิทธิ์การเข้าถึง (Authorization as a Service:AaaS) ซึ่งเป็นตัวกลางที่คั่นระหว่างการเรียกใช้บริการบนคลาวน์กับผู้ใช้งานซึ่งสามารถจะนำมาใช้กับผู้ให้บริการเดียว หรือข้ามกันระหว่างผู้ให้บริการก็ได้ สุดท้ายนี้ ได้ทดสอบประสิทธิ์ภาพการทำงานของตัวต้นแบบภายในฮาร์ดแวร์ที่มีขนาดแตกต่างกัน เพื่อวิเคราะห์เปรียบเทียบการขยายขนาดของฮาร์ดแวร์ว่ามีผลต่อการทำงานกับตัวต้นแบบอย่างไร จากการทดสอบพบว่าระบบสามารถรองรับการปรับขนาดของฮาร์ดแวร์ที่เพิ่มมากขึ้นได้เป็นอย่างดี

คำสำคัญ: การตรวจสอบสิทธิ์การเข้าถึง, ระบบการเช่าของผู้เช่าหลายราย, การบริการที่ทำงานร่วมกัน, MTAS, การเชื่อถือ, การประมวลผลบนกลุ่มเมฆ

Thesis Title	Calling Chain in Coordination for Multi-Tenant Collaborative Cloud Services
Author	Mr. Danai Thongsang
Degree	Master of Science Program Computer Science
Major Field/Faculty/University	Department of Computer Science Faculty of Science and Technology Thammasat University
Thesis Advisor	Assoc. Prof. Dr. Yaowadee Temtanapat
Academic Years	2015

ABSTRACT

Currently, a cloud service is widely available but its access control is usually limited and tied only to its tenancy in isolation. To take full advantage from cloud services, multiple tenancies with some level of mutual trust would seek to collaborate and share their resources. However, building a collaborative application from inter-related chain callings to various services on a single or multiple cloud systems encounters an access control challenge. It becomes a big barrier to its adoption. To provide an appropriate fine grained chain calling authorization, this paper proposes an extension to Multi-Tenant Authorization System Model (MTAS), named “Calling Chain Coordination in MTAS” (C-MTAS). In the MTAS, a service with several chain callings would require the model to break a tenant’s role into too many sub-roles with a limited trust scope. This would increase unintentional number of roles that could lead to breaches. It would be also hard to maintain. We, instead, propose to separate a tenant element to make a non-redundant, clear and simplified set of roles and permissions. The benefit of our model to the MTAS is shown by applying both models to the same concrete scenario. We found that our model gives a cleaner and smaller set of rules as compared to the MTAS’s. We also illustrate how to use our model via a practically feasible example policy in the XACML format. The prototype system is built as an Authorization as a Service (AaaS) platform, a middle layer on the part of the cloud services, which can be used by the same or across providers. Finally, it is tested on different hardware sets. The results showed that the model could be scalable.

Keywords: authorization, multi-tenancy, collaboration service MTAS, trust, cloud computing

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความช่วยเหลืออย่างดียิ่งจากรองศาสตราจารย์ ดร. เยาวดี เต็มธนาภัทร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้เสียสละเวลาอันมีค่าให้คำแนะนำเกี่ยวกับแนวทางการวิจัย และคำปรึกษาที่มีประโยชน์อย่างมากกับวิทยานิพนธ์นี้ ขอขอบพระคุณ ดร. กษิดิศ ชาญเชี่ยว ผู้ช่วยศาสตราจารย์ ดร. ทรงศักดิ์ รองวิริยะพานิช และผู้ช่วยศาสตราจารย์ ดร. เบญจพร ลิ้มธรรมาภรณ์ คณะกรรมการสอบวิทยานิพนธ์ ที่ท่านได้กรุณาให้คำแนะนำ และชี้แนะในการทำงานวิจัย รวมถึงตรวจสอบวิทยานิพนธ์ฉบับให้สำเร็จลุล่วงได้อย่างดี

สุดท้ายนี้ ข้าพเจ้าขอกราบขอบพระคุณบิดา มารดา และญาติพี่น้องทุกท่านในครอบครัว ที่สนับสนุนด้านการเงิน ให้ความห่วงใย และให้กำลังใจ รวมทั้ง นางสาวพัชรพรรณ ขุมแร่ ที่คอยให้กำลังใจ ร่วมให้คำปรึกษาในเรื่องต่าง ๆ จนวิทยานิพนธ์สำเร็จลุล่วงมาด้วยดีตลอด อีกทั้งบริษัท ไอเจนโก้ จำกัด ที่สละเวลาให้ข้าพเจ้าได้ลา จนมีเวลาทำวิทยานิพนธ์ชิ้นนี้ได้สำเร็จ

นายดนัย ทองแสง

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	(1)
บทคัดย่อภาษาอังกฤษ	(3)
กิตติกรรมประกาศ	(5)
สารบัญตาราง	(9)
สารบัญภาพ	(10)
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	11
1.3 ขอบเขตของการศึกษา	11
1.4 ประโยชน์ที่คาดว่าจะได้รับ	12
บทที่ 2 วรรณกรรมและงานวิจัยที่เกี่ยวข้อง	13
2.1 ทฤษฎีที่เกี่ยวข้อง	13
2.1.1 การประมวลผลบนกลุ่มเมฆ หรือคลาวน์ (Cloud Computing)	13
2.1.2 eXtensisible Access Control Markup Language (XACML)	15
2.2 งานวิจัยที่เกี่ยวข้อง	18
2.2.1 โมเดลการควบคุมการเข้าถึงตามบทบาท (Role-based Access Control: RBAC)	19
2.2.2 ลักษณะที่เกี่ยวข้องกระบวนการห่วงโซ่การเรียกใช้บริการแบบประสานงาน	32
2.2.3 การตรวจสอบกราฟที่มีลักษณะของการเกิด cycle	36

บทที่ 3 วิธีการวิจัย 38

3.1 การออกแบบการวิจัย 38
3.2 ขั้นตอนการวิจัย 39
3.3 ออกแบบโมเดลสำหรับการควบคุมการเข้าถึง เพื่อใช้ในตรวจสอบสิทธิ์การเข้าใช้เพื่อรองรับการใช้บริการแบบห่วงโซ่การเรียกใช้บริการแบบประสานงานกัน 40
3.3.1. องค์ประกอบของโมเดล C-MTAS 40
3.3.2. การอธิบายความสัมพันธ์ระหว่างองค์ประกอบทั้ง 5 เพื่อสร้างเซตในการอธิบายการตรวจสอบสิทธิ์การเข้าใช้บริการ 43
3.3.3. การบริการแบบห่วงโซ่การเรียกใช้แบบประสานงาน 49
3.4. พัฒนาระบบการตรวจสอบสิทธิ์จากโมเดลการตรวจสอบสิทธิ์ที่ได้กำหนดขึ้น 50
3.5. กระบวนการทดสอบหลังจากการ implement 55
3.6. วิเคราะห์ในเชิงการเปรียบเทียบ เรื่องของการปรับขนาดของฮาร์ดแวร์ในการรองรับการทำงานของ C-MTAS 59

บทที่ 4 ผลการวิจัยและอภิปรายผล 60

4.1. สภาพแวดล้อมที่ใช้ในการพัฒนาและทดสอบตัวจำลอง C-MTAS 60
4.1.1. ฮาร์ดแวร์ (Hardware) 60
4.1.1. ซอฟต์แวร์ (Software) 60
4.2. ตัวระบบที่ถูกจำลองจากโมเดล C-MTAS 61
4.2.1. การส่งการขอร้องเพื่อตรวจสอบสิทธิ์การเข้าถึง (input) 61
4.2.2. กระบวนการคำนวณการตรวจสอบสิทธิ์การเข้าถึง 61
4.2.3. ผลลัพธ์ขอการร้องขอ 63
4.3. สรุปผลการทดสอบในกรณีปกติ 64
4.4. สรุปผลการทดสอบในกรณีที่เกิดการเรียกบริการแบบห่วงโซ่การบริการแบบประสานงาน 65
4.5. สรุปผลการวิเคราะห์ในเชิงการเปรียบเทียบ เรื่องของการปรับขนาดของฮาร์ดแวร์ในการรองรับการทำงานของ C-MTAS 70
4.5.1. ค่าเวลาการตอบสนองเฉลี่ย (Average Response Time) 74
4.5.2. ค่าภาระการทำงานบนเซิร์ฟเวอร์ (Througtput) 75
4.5.3. กราฟฮิสโตแกรม โดยนำระยะเวลาที่ใช้ในการประมวลผลมาแสดงผล (Histrogram) 76

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ 81

5.1. สรุปผลการวิจัยจากผลการทดสอบการทำงานของ AaaS 82
5.2. สรุปผลการวิเคราะห์ในเชิงการเปรียบเทียบ เรื่องของการปรับขนาดของฮาร์ดแวร์ในการรองรับการทำงานของ C-MTAS 82
5.3. ข้อเสนอแนะในการทำวิจัยในอนาคต 83

รายการอ้างอิง 84

ภาคผนวก 86

ภาคผนวก ก กรณีทดสอบ ในกรณีปกติ 87

ประวัติผู้เขียน 103

สารบัญตาราง

ตารางที่ หน้า

1.1 อธิบายความสัมพันธ์ขององค์ประกอบใน MTAS 4

2.1 แสดงคำนิยามของ Core RBAC 20

2.2 แสดงความสัมพันธ์ของ RH 22

2.3 องค์ประกอบของโมเดล MTAS 24

2.4 ตารางแสดงความสัมพันธ์ต่าง ๆ ระหว่างองค์ประกอบของ AMTAS 25

3.1 องค์ประกอบของโมเดล C-MTAS 41

3.2 ความสัมพันธ์ในโมเดล C-MTAS 43

3.3 นิยามฟังก์ชันในการจัดการในการบริหารการควบคุมการเข้าถึง 47

4.1 ผลลัพธ์จากการรันทดสอบเชิงเปรียบเทียบการทำงานของ C-MTAS ตามสภาพแวดล้อมเครื่องจำลองที่แตกต่างกัน 72

4.2 ข้อมูลผลลัพธ์จากการรันเปรียบเทียบประสิทธิภาพการปรับ ขนาดของฮาร์ดแวร์ โดยแสดงค่าระยะเวลาการตอบสนองเฉลี่ย 74

4.3 ข้อมูลผลลัพธ์จากการรันเปรียบเทียบประสิทธิภาพการปรับ 75

ก.1 รายละเอียดกรณีทดสอบทั้งหมด ในกรณีปกติ 87

สารบัญภาพ

ภาพที่ หน้า

1.1 10 อันดับจาก 100 บริษัททั่วโลกที่มีรายได้จาก SaaS มากที่สุด และอัตราการเติบโตของบริษัท 2

1.2 โมเดลของระบบ MTAS 3

1.3 ความสัมพันธ์ระหว่างบทบาท(R) ประเด็น(I) และสิทธิ์/การเช่าใช้(P/T) ในโมเดล MTAS 6

1.4 ความสัมพันธ์ระหว่างบทบาท(R) ประเด็น(I) และสิทธิ์/การเช่าใช้(P/T) ในโมเดล MTAS โดยสลับ Tenent กับ Issuer ทำให้เห็นภาพของ 1 บทบาทอยู่ภายใต้หลายการเช่า 7

1.5 แสดงการเรียกใช้บริการที่เป็นห่วงโซ่การเรียกบริการแบบประสานงาน 8

1.6 การกำหนด File Policy ของ XACML ในโมเดล MTAS ในการจัดการ 1 บทบาท จะประกอบไปด้วย 3 ไฟล์ RPS TPS และ PPS 8

1.7 การกำหนด File Policy ของ XACML ในโมเดล MTAS ในการจัดการ 1 บทบาทที่ Cross-issuer ให้กับอีก iseuer หนึ่ง 9

2.1 แผนการจำแนกประเภทของประมวลผลบนคลาวน์ โดยแยกจากการใช้งาน 14

2.2 แผงผังการไหลผ่านของข้อมูลในการทำงานของ XACML 16

2.3 โมเดลสำหรับภาษาในการเขียนนโยบาย XACML 17

2.4 กระบวนการของ XACML 18

2.5 แสดงให้เห็นถึง Core RBAC 19

2.6 *แสดงให้เห็นถึง Hierarchai RBAC* 21

2.7 โมเดลของระบบ MTAS ที่มา: Multi-Tenancy Authorization Models for Collaborative Cloud Services. IEEE, CTS (2013) 23

2.8 แผนภาพอธิบายฟังก์ชันของการบริหารจัดการของ AMTAS ที่มา: Multi-Tenancy Authorization Models for Collaborative Cloud Services. IEEE, CTS (2013) 27

2.9 แผนผังแสดงองค์ประกอบ และความสัมพันธ์ของโมเดล RAMARS_RM 28

2.10 องค์ประกอบและความสัมพันธ์ของโมเดล WS-RBAC 29

2.11 องค์ประกอบและความสัมพันธ์ของโมเดล SWS-RBAC 30

2.12 องค์ประกอบและความสัมพันธ์ของโมเดล MT-RBAC 31

2.13 อธิบายความหมายในการใช้เครื่องหมาย $\otimes$ 32

2.14 อธิบายความหมายในการใช้เครื่องหมาย $\oplus$ 33

2.15 ตัวอย่างของการกำหนดในเหตุการณ์ของ Ownership Chain 35

2.16 แสดงตัวอย่างกราฟที่ไม่มีการวนกลับ ซึ่งแสดงให้เห็น worst case ของการใช้ระยะเวลาของ Tarjan 37

2.17 แสดงตัวอย่างกราฟที่มีการวนกลับ และมีเส้นทางที่ซับซ้อน ซึ่งแสดงให้เห็น worst case ของการใช้ระยะเวลาของ Tarjan 37
3.1 ความสัมพันธ์ระหว่างบทบาท (R) ประเด็น (I) และสิทธิ์/การเช่าใช้ (P/T) ในโมเดล C-MTAS เพื่อแก้ไขบทบาท 1 บทบาทเข้าถึงได้ 1 การเช่า 38
3.2 แผนผังองค์ประกอบโมเดล C-MTAS 42
3.3 ภาพแสดงห่วงโซ่การเรียกบริการแบบประสานงาน 49
3.4 อัลกอริทึมการตรวจสอบสิทธิ์การเข้าถึง 52
3.5 แผนผังสถาปัตยกรรมซอฟต์แวร์เพื่อใช้ในการตรวจสอบสิทธิ์การเข้าถึงของโมเดล C-MTAS 53
3.6 แผนภาพประกอบการอธิบาย XACML ที่ใช้ในการอธิบายโมเดล C-MTAS 54
3.7 แผนภาพแสดงเหตุการณ์จำลองเพื่อใช้ในกระบวนการทดสอบ 56
3.8 ความสัมพันธ์ของการกำหนดบทบาทและการอนุญาตในการเช่าของตนเอง (เส้นทึบ) รวมทั้งการกำหนดบทบาทในการข้ามกันของแต่ละการเช่า (เส้นประ) 58
3.9 กรณีทดสอบ ในห่วงโซ่การบริการแบบประสานงาน 58
4.1 องค์ประกอบ XML ในการร้องขอเพื่อใช้ในการตรวจสอบสิทธิ์ในระบบ C-MTAS 62
4.2 ภาพตัวอย่างการกำหนดเซตของ CCPR โดยใช้ XML ในการกำหนด 63
4.3 องค์ประกอบ XML สำหรับผลลัพธ์ในการตรวจสอบสิทธิ์ในระบบ C-MTAS 64
4.4 แสดงองค์ประกอบในการหาจำนวนกรณีทดสอบในกรณีแบบปกติ 64
4.5 ผลลัพธ์การรันกรณีทดสอบแบบปกติ 65
4.6 แสดงกรณีทดสอบที่ 4.3.1 66
4.7 แสดงกรณีทดสอบที่ 4.3.2 66
4.8 แสดงกรณีทดสอบที่ 4.3.3 67
4.9 แสดงกรณีทดสอบที่ 4.3.4 67
4.10 แสดงกรณีทดสอบที่ 4.3.5 68
4.11 แสดงกรณีทดสอบที่ 3.4.6 68
4.12 แสดงกรณีทดสอบที่ 4.3.7 69
4.13 ผลลัพธ์การรันกรณีทดสอบการเกิดห่วงโซ่การบริการแบบประสานงาน 69
4.14 ตัวอย่างหน้าจอของการรันโปรแกรม VitualBox ในการจำลอง 71
4.15 ตัวอย่างหน้าจอ GlassFish โดยติดตั้ง AssS 71
4.16 ตัวอย่างหน้าจอผลการรัน jMeter 73

4.17 กราฟแสดงระยะเวลาการตอบสนองเฉลี่ย ในแต่ละสภาพแวดล้อมของฮาร์ดแวร์ที่แตกต่างกัน 74
4.18 กราฟแสดงภาระการทำงาน ในแต่ละสภาพแวดล้อมของฮาร์ดแวร์ที่แตกต่างกัน 76
4.19 กราฟฮิสโตแกรมแสดงจำนวนการแจกแจงของระยะเวลา ที่ใช้ในการตอบสนองในกรณีของผู้ใช้งาน 10 ผู้ใช้งานพร้อมกัน โดยแยก 4 ชุดของฮาร์ดแวร์ 77
4.20 กราฟฮิสโตแกรมแสดงจำนวนการแจกแจงของระยะเวลา ที่ใช้ในการตอบสนองในกรณีของผู้ใช้งาน 100 ผู้ใช้งานพร้อมกัน โดยแยก 4 ชุดของฮาร์ดแวร์ 79
4.21 กราฟฮิสโตแกรมแสดงจำนวนการแจกแจงของระยะเวลาที่ใช้ในการตอบสนองในกรณีของผู้ใช้งาน 1000 ผู้ใช้งานพร้อมกัน โดยแยก 4 ชุดของฮาร์ดแวร์ 80

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันไม่อาจปฏิเสธได้ว่า “การประมวลผลบนกลุ่มเมฆหรือคลาวน์” เป็นหนึ่งในกระแสสำคัญที่มีอิทธิพลกับเทคโนโลยีสารสนเทศ เนื่องจากระบบการประมวลผลบนคลาวน์เป็นการใช้ทรัพยากรร่วมกัน ช่วยลดต้นทุนทางด้านฮาร์ดแวร์ ผู้ใช้สามารถเช่าใช้บริการตามความต้องการโดยค่าใช้จ่ายที่เกิดขึ้นจะถูกคำนวณตามทรัพยากรและระยะเวลาที่เช่าใช้ประกอบกัน นอกจากนี้องค์กรจำนวนมาก เช่น กูเกิล ไมโครซอฟต์ Amazon ที่เป็นองค์กรยักษ์ใหญ่ ผู้นำในด้านเทคโนโลยี ก็มีผลิตภัณฑ์ที่ออกมาในรูปของคลาวน์เพิ่มมากขึ้น เช่น ไมโครซอฟต์ได้ออกผลิตภัณฑ์ Azure ที่เป็นการบริการทั้งแฟลตฟอร์มของคลาวน์ หรือการบริการคลาวน์ กูเกิลเองได้มีการพัฒนาแพลตฟอร์มเพื่อใช้งานของกูเกิลเองโดยแพลตฟอร์มดังกล่าวอยู่ในรูปของการบริการคลาวน์ รวมทั้ง Amazon EC2 ก็เป็นการประมวลผลโดยใช้คลาวน์ ที่ให้ผู้ใช้บริการมาเช่าใช้บริการได้เช่นเดียวกัน จากตัวอย่างดังกล่าวนี้เป็นหนึ่งในหลาย ๆ องค์กรที่ได้หันมานำกระแสของคลาวน์ จึงหลีกเลี่ยงไม่ได้ว่า การบริการคลาวน์จึงมีบทบาทที่สำคัญเป็นอย่างยิ่งในทางด้านเทคโนโลยีสารสนเทศ

การประมวลผลบนคลาวน์ แบ่งออกเป็น 3 ประเภทตามการใช้งาน คือ Software as a Service (SaaS) เป็นการให้บริการคลาวน์ในด้านของแอพพลิเคชั่นและบริการทางด้านซอฟแวร์ Platform as a Service (PaaS) เป็นการให้บริการคลาวน์ในด้านของแพลตฟอร์ม ที่ไม่ได้เฉพาะเจาะจงในเรื่องของ ระบบปฏิบัติการ แต่เป็นการพูดถึงแฟลตฟอร์มอื่น ๆ ที่สามารถนำมาใช้งานบนโครงสร้างของคลาวน์ได้ อาทิเช่น Google Cloud Platform เป็นต้น และส่วนสุดท้ายคือ Infrastructure as a Service (IaaS) เป็นการให้บริการคลาวน์ในระดับของโครงสร้างของสถาปัตยกรรมคอมพิวเตอร์พื้นฐาน เป็นการกล่าวถึงฮาร์ดแวร์อย่างเดียว อาทิเช่น เครือข่าย พื้นที่ในการจัดเก็บข้อมูล เป็นต้น

จากทั้งสามประเภทของการประมวลผลบนคลาวน์นี้ SaaS เป็นประเภทหนึ่งที่มี ดังจะเห็นได้จากภาพที่ 1.1 ที่แสดงให้เห็นถึงการเติบโตของ SaaS จะเห็นได้ว่าการบริษัทใหญ่ ๆ มีรายได้จากการซื้อซอฟต์แวร์แบบ SaaS เห็นได้ว่ามีอัตราการเติบโตรวมทั้งสิ้น 17.5 เปอร์เซ็นในปี ค.ศ. 2011 และคาดการณ์ว่าจะเพิ่มขึ้นเป็น 24 เปอร์เซ็นในปีค.ศ. 2016

Figure 1: Top 10 SaaS revenues amongst the Global 100

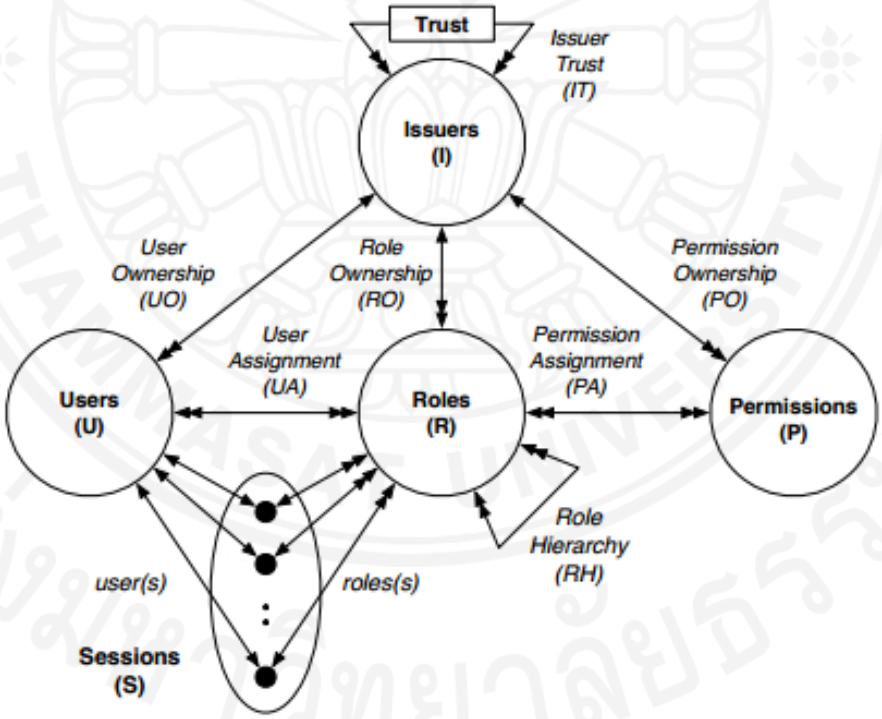
Company	Country HQ	2011 SaaS revenue (US$M)	2011 software revenue (US$M)	SaaS revenue as % of software revenue
Salesforce.com	US	$1,848	$2,008.7	92.0%
Intuit	US	$950	$2,456.5	38.7%
Cisco	US	$831	$1,796.9	46.3%
Microsoft	US	$788	$57,668.4	1.4%
Symantec	US	$572	$6,330.3	9.0%
Google Inc.	US	$462	$575.6	80.3%
Oracle	US	$446	$26,175.9	1.7%
Adobe	US	$410	$4,154.1	9.9%
Blackboard	US	$396	$411.7	96.2%
DATEV	Germany	$395	$974.2	40.5%
	Total	$7,098	$102,552.16	6.9%

ภาพที่ 1.1 10 อันดับจาก 100 บริษัททั่วโลกที่มีรายได้จาก SaaS มากที่สุด และอัตราการเติบโตของบริษัท
ที่มา: http://www.slideshare.net/marcos_serrao/pwc-global100softwareleaders

เมื่อการใช้งานบนคลาวน์ในรูปแบบของซอฟต์แวร์มีการเติบโตมากขึ้น ประเด็นปัญหาในด้านความมั่นคงของการใช้ข้อมูลที่อยู่บนคลาวน์จึงมีความท้าทาย และได้รับความสนใจมากขึ้นตามไปด้วย

สำหรับการอธิบายถึงกระบวนการใช้บริการซอฟต์แวร์บนคลาวน์นั้น เริ่มตั้งแต่เมื่อผู้ต้องการใช้บริการ SaaS บนคลาวน์ได้ลงทะเบียนการใช้งานไปยังผู้ให้บริการ และเมื่อผู้ให้บริการตอบรับ และส่งรหัสผู้ใช้งานไปยังผู้เช่าบริการ เราเรียกกระบวนการนี้ว่า “การเช่า” (Tenant) ด้วยลักษณะของ SaaS จะต้องสามารถรองรับการผู้ใช้งาน หรือเรียกอีกอย่างได้ว่าเจ้าของการเช่า (Tenancy) ได้หลายคน ดังนั้นลักษณะของกระบวนการควบคุมการเข้าถึง (Access Control) จะต้องรองรับการตรวจสอบการเข้าถึงของทรัพยากรต่าง ๆ ตามแต่ละผู้เช่า โดยไม่ให้ผู้เช่าอื่นเข้ามาจัดการของทรัพยากรของตนเองได้ โดยลักษณะของการเข้าใช้ลักษณะนี้ ถูกเรียกว่า “ผู้เช่าหลายราย” (Mulit-Tenancy) (Calero, Edwards, Kirschnick, Wilcock, and Wray, 2010, pp. 49)

หนึ่งในโมเดล ที่กล่าวถึงการควบคุมการเข้าถึงข้อมูล ที่เป็นพื้นฐานของหลาย ๆ โมเดล รวมทั้งยังคงนิยมใช้กันอยู่แพร่หลายในปัจจุบัน คือ โมเดลการควบคุมการเข้าถึงตามบทบาท (Role-Base Access Control: RBAC) โดยโมเดลดังกล่าวได้อธิบายการตัดสินใจในการตรวจสอบการเข้าถึงโดยนำนโยบายมาอธิบายโดยใช้ “บทบาท” (Role) เป็นข้อกำหนด เพื่อให้ผู้ใช้งาน (User) ที่ถูกกำหนดให้ใช้บทบาท ซึ่งบทบาทถูกอธิบายด้วยสิทธิ์ (Permission) ในการเข้าถึง object ดังนั้น RBAC โดยสรุปได้กล่าวถึงองค์ประกอบหลัก 3 ส่วน คือ ผู้ใช้งาน บทบาท และสิทธิ์การเข้าถึง และหากพิจารณานำโมเดล RBAC มาใช้ควบคุมการเข้าถึงในลักษณะของผู้เช่าหลายรายแล้วนั้น พบว่า ข้อมูลในแต่ละการเช่าจะถูกกำหนดเป็น Object และมีการกำหนดบทบาท ซึ่งถูกรวมกัน โดยไม่แบ่งตามผู้เช่า เมื่อมีการเพิ่มบทบาทใหม่ ซึ่งบทบาทนี้กำหนดให้กับผู้ใช้งานเฉพาะสิทธิ์การเช่าของตนเองเท่านั้น แต่อย่างไรก็ตามบทบาทนี้ยังสามารถกำหนดให้กับผู้ใช้งานสำหรับเช่าอื่นได้ เนื่องจากบทบาทไม่ได้ถูกแยกตามสิทธิ์ของการเข้า ด้วยเหตุผลนี้เองทำให้โมเดลของ RBAC ไม่รองรับการทำงานในสภาพแวดล้อมของคลาวน์ในผู้เช่าหลายราย



ภาพที่ 1.2 โมเดลของระบบ MTAS
ที่มา: “Multi-Tenancy Authorization Models for Collaborative Cloud Services,” by B. Tang, R. Sandhu, and Q. Li, pp. 229-238. Copyright 2013 by IEEE.

และอีกหนึ่งโมเดล ที่ได้ถูกเสนอเพื่อให้รองรับการทำงานบนคลาวน์ในรูปแบบของผู้เช่าหลายราย คือ MTAS ซึ่งได้ถูกต่อยอดมาจาก RBAC โดยได้เพิ่มองค์ประกอบที่จำเป็นในการแบ่งเจ้าของการเช่าออกอย่างชัดเจน เรียกว่า Issuer และกำหนดความสัมพันธ์เพื่อที่จะสามารถ

จำแนกบทบาท สิทธิ์การเข้าถึง ผู้ใช้งาน ตามเจ้าของการเช่าใช้ได้ ทำให้บทบาทไม่ได้ถูกรวมกัน เหมือนบทบาทของโมเดล RBAC และในโมเดลของ MTAS ยังสามารถกำหนดสิทธิ์การเข้าถึงที่เรียกว่า การให้สิทธิ์ข้ามกันระหว่างเจ้าของการเช่า (Cross-Tenancy) ได้อีกด้วย โดยดูจากความเชื่อถือ (Trust) ของเจ้าของการเช่าไปยังผู้ถูกเชื่อถือซึ่งเป็นอีกเจ้าของการเช่าหนึ่งให้สามารถเข้าถึงทรัพยากรของตนได้

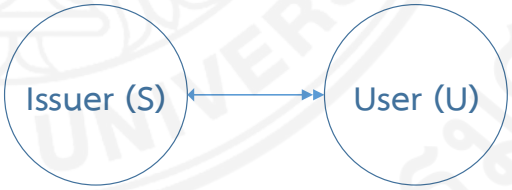
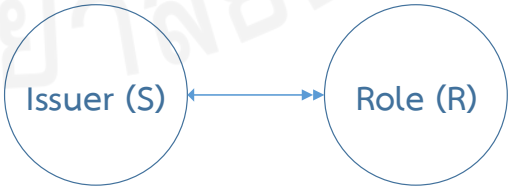
โดยในโมเดลของ MTAS แบ่งโมเดลออกเป็น 3 ส่วน

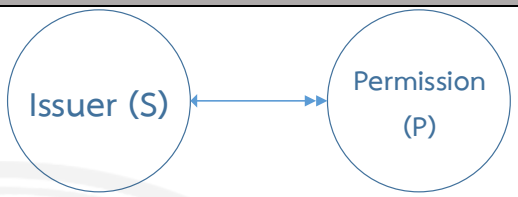
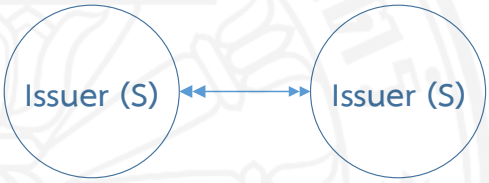
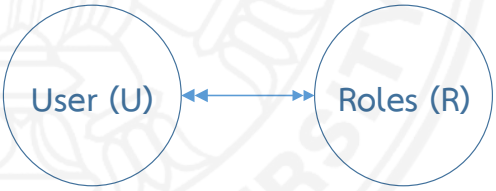
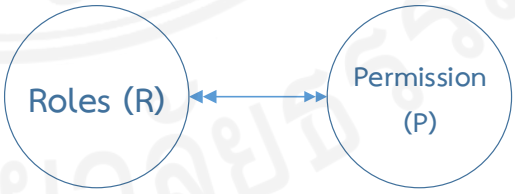
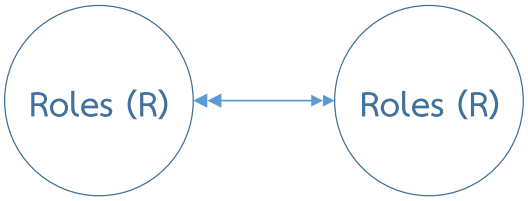
ส่วนแรกคือองค์ประกอบของโมเดลซึ่งได้แบ่งออกเป็น 5 องค์ประกอบ คือ เจ้าของการเช่า(Issuers) ผู้ใช้งาน(Users) การอนุญาต(Permissions) บทบาท (Roles) และเซสชั่น (Sessions) ตามภาพที่ 1.2

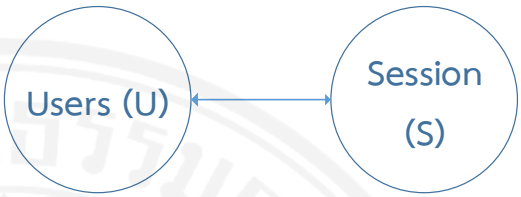
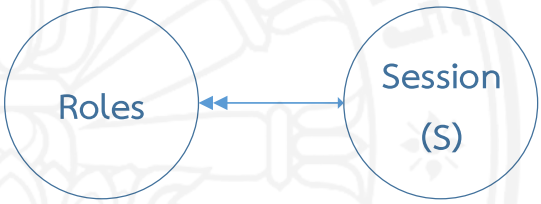
ส่วนที่สองอธิบายความสัมพันธ์ขององค์ประกอบทั้ง 5 ซึ่งแบ่งออกได้เป็น 9 ความสัมพันธ์ ดังตารางที่ 1.1

ตารางที่ 1.1

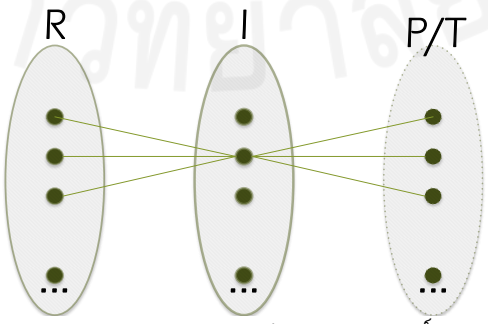
ความสัมพันธ์ขององค์ประกอบใน MTAS

ชื่อความสัมพันธ์	รายละเอียด
UserOwnersship (UO)	เซตที่แสดงถึง *Issuer* มีผู้ใช้งานอะไรบ้าง เป็นความสัมพันธ์แบบ 1-many 
RoleOwnership (RO)	เซตที่แสดงถึง *Issuer* มีบทบาทอะไรบ้าง เป็นความสัมพันธ์แบบ 1-many 
PermissionOwnership (PO)	เซตที่แสดงถึง *Issuer* มีสิทธิ์อะไรบ้าง เป็นความสัมพันธ์แบบ 1-many

ชื่อความสัมพันธ์	รายละเอียด
	
Issuers Trust (IT)	เซตความสัมพันธ์ระหว่าง *issuer* กับ *issuer* ในการกำหนดความเชื่อถือระหว่างกัน โดยใช้เครื่องหมาย " $\lesssim$ " แทนความเชื่อถือจากความเชื่อถือไปยังผู้ถูกเชื่อถือ เป็นความสัมพันธ์แบบ many-many 
UserAssignment (UA)	เซตที่แสดงถึงผู้ใช้งานสามารถใช้บทบางอะไรบ้าง เป็นความสัมพันธ์แบบ many-many 
Permission Assignment (PA)	เซตที่แสดงถึงบทบาทแต่ละบทบาทมีสิทธิ์อะไรบ้าง เป็นความสัมพันธ์แบบ many-many 
Role Hierarchy (RH)	เซตในการกำหนดชั้นของบทบาท เป็นความสัมพันธ์แบบ *many-many* 

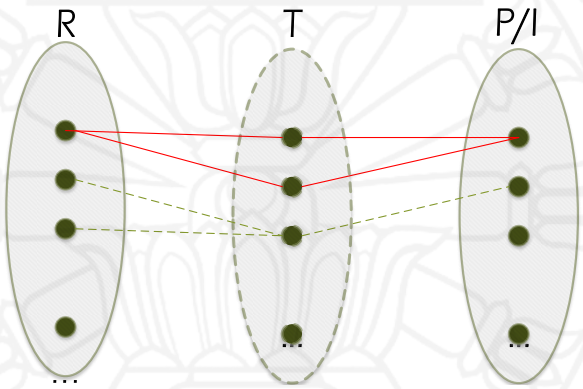
ชื่อความสัมพันธ์	รายละเอียด
$\text{user}(s: S) \rightarrow U$	ฟังก์ชันที่ตรวจสอบผู้ใช้งานมีอยู่ในเซสชั่นหรือไม่ โดยเป็นความสัมพันธ์แบบ *1-1* 
$\text{roles}(s: S) \rightarrow 2^R$	ฟังก์ชันที่ตรวจสอบบทบาทถูกใช้งานอยู่ในเซสชั่นหรือไม่ โดยเป็นความสัมพันธ์แบบ *many-1* 

ในการกำหนดการเข่าแต่ละซอฟต์แวร์สำหรับโมเดล MTAS นั้น ได้กำหนดไว้ภายใต้การกำหนด Permission ซึ่งมีส่วนประกอบ 3 ส่วนดังนี้ (ชื่อสิทธิ์, ชื่อการเช่า, วัตถุที่ให้สิทธิ์) ดังนั้นเมื่อแต่ละ issuers มีการเช่าหลายการเช่า Permission ที่แสดงข้อมูลการเช่าก็จะถูกรวมเก็บไว้โดยไม่ได้แบ่งแต่ละการเช่าออกจากกัน อีกทั้งเมื่อวิเคราะห์การกำหนดความสัมพันธ์ของ RoleOwnership (RO) และ PermissionOwnership (PO) ที่ RO เป็นความสัมพันธ์จาก Role ไปยัง Issuer แบบ many-1 และ PO เป็นความสัมพันธ์จาก Issuer ไปยัง Permission แบบ 1-many ซึ่งสำหรับแต่ละ Permission ครอบสิทธิ์ในการเข้าใช้ Tenant ดังนั้นหากแสดง Tenant ให้ปรากฏร่วมใน Permission จะทำให้เห็นถึงปัญหาชัดเจนขึ้น โดยนำความสัมพันธ์ของ Role Issuers และ การเช่า (Tenant) มาเขียนความสัมพันธ์ แสดงได้ดังในตามภาพที่ 1.3



ภาพที่ 1.3 ความสัมพันธ์ระหว่างบทบาท (R) ประเด็น (I) และสิทธิ์/การเช่าใช้ (P/T) ในโมเดล MTAS

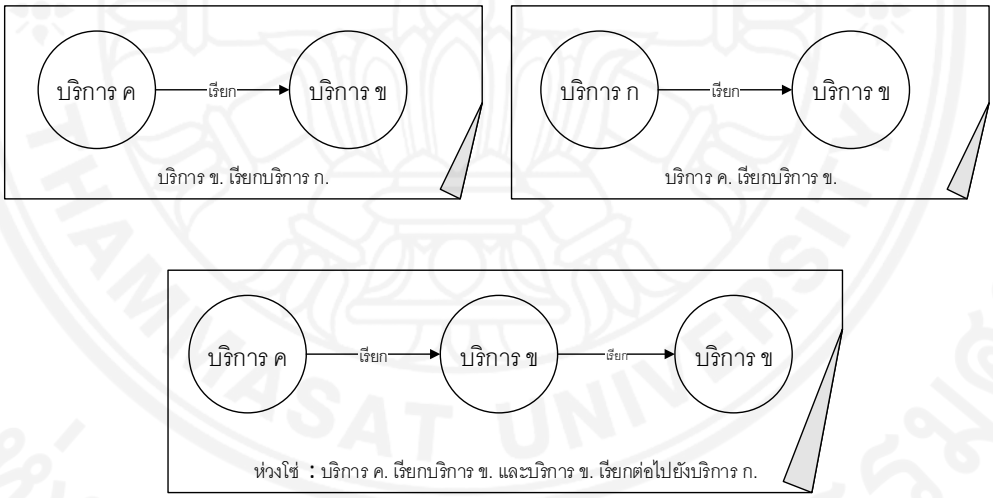
จากภาพที่ 1.3 จะเห็นได้ว่า 1 บาทบาทสามารถถูกกำหนดไว้ภายใต้ issuer ได้แค่ 1 issuer และจากการที่แต่ละ issuer สามารถมีการเช่า (Tenant) ใช้ซอฟต์แวร์ได้หลายตัว ซึ่งเมื่อดูความสัมพันธ์ระหว่าง Role กับการเช่า (Tenant) จะเห็นได้ว่า เป็นความสัมพันธ์แบบ 1-many คือ 1 บทบาทสามารถกำหนดสิทธิ์การใช้งานของการเช่าซอฟต์แวร์ได้มากกว่า 1 การเช่า ดังแสดงให้เห็นในเส้นทึบตามภาพที่ 1.4 แต่ทว่า Role หรือบทบาทดังกล่าวนั้น ผู้ที่กำหนดบทบาทที่จะให้แต่ละ Issuer ใช้งานได้นั้น เจ้าของซอฟต์แวร์แต่ละซอฟต์แวร์จะเป็นผู้กำหนดขึ้น ดังนั้นเมื่อความสัมพันธ์ที่ MTAS กำหนดไว้ว่า 1 Role สามารถเข้าใช้ได้หลาย Tenant ทำให้เกิดข้อขัดแย้งกันกับลักษณะของการกำหนดบทบาท จากปัญหาที่หนึ่งบทบาทสามารถกำหนดการเข้าถึงได้มากกว่าหนึ่งนี้จึงเป็นที่มาของงานวิจัยชิ้นนี้ เพื่อให้เจ้าของซอฟต์แวร์เป็นผู้กำหนดบทบาทอย่างแท้จริง ซึ่งบทบาทนั้นจะต้องกำหนดภายใต้ซอฟต์แวร์ของตนเท่านั้น



ภาพที่ 1.4 ความสัมพันธ์ระหว่างบทบาท (R) ประเด็น (I) และสิทธิ์/การเช่าใช้ (P/T) ในโมเดล MTAS โดยสลับ Tenent กับ Issuer ทำให้เห็นภาพของ 1 บทบาทอยู่ภายใต้หลายการเช่า

มากไปกว่านั้น เมื่อการบริการซอฟต์แวร์ หรือ SaaS เริ่มที่จะได้รับความนิยมกันมากขึ้น ผู้ให้บริการซอฟต์แวร์ต่าง ๆ เริ่มนำแต่ละซอฟต์แวร์ของตนมาให้บริการบนคลาวน์ และเป็นไปได้สูงที่ซอต์แวร์เหล่านั้นจะสามารถนำมาเชื่อมโยงกับกับซอฟต์แวร์ตัวอื่น ๆ เพื่อให้ความสามารถของซอฟต์แวร์เพิ่มขึ้นโดยนำความสามารถของซอฟต์แวร์อื่น ๆ ที่มีอยู่แล้ว นำมาก่อให้เกิดประโยชน์ มากกว่าการพัฒนาซอฟต์แวร์หนึ่ง ๆ ให้สามารถทำได้ทุกอย่าง ซึ่งต้องใช้ทรัพยากรด้านเงิน คน และเวลาอย่างมหาศาล โดยผู้วิจัยได้ขอเรียกลักษณะการใช้งานร่วมกันนี้ว่า “ห่วงโซ่การเรียกบริการแบบประสานงาน” กล่าวคือ การเช่าบริการหนึ่งเรียกไปยังอีกบริการหนึ่ง และเรียกต่อกันไปเรื่อยจนกระทั่งต้องการข้อมูลหรือทรัพยากรที่ต้องการแล้วตามภาพที่ 1.5 อาทิ ในปัจจุบันคือ Google เชื่อมโยงทรัพยากรระหว่าง Google Doc กับ Google Drive โดย Google Doc เป็นซอฟต์แวร์ที่สามารถจัดการเกี่ยวกับเอกสาร ไม่ว่าจะเป็น doc excel ppt หรือเอกสารอื่น ส่วน Google Drive เป็นซอฟต์แวร์ที่ใช้ในการจัดเก็บไฟล์เอกสารต่าง ๆ ของผู้ใช้บริการ โดยตัวอย่างของลักษณะของห่วงโซ่การเรียกบริการแบบประสานงาน คือ เมื่อมีการดึงข้อมูลเอกสารจาก Google Drive มาเปิดใน Google Doc แล้วแก้ไข แล้วบันทึกกลับไปยัง Google Drive ด้วยลักษณะของผู้ให้บริการเป็นคนเดียวกัน จึงสามารถจัดการในส่วนเรื่องของสิทธิ์การเข้าถึงได้ แต่

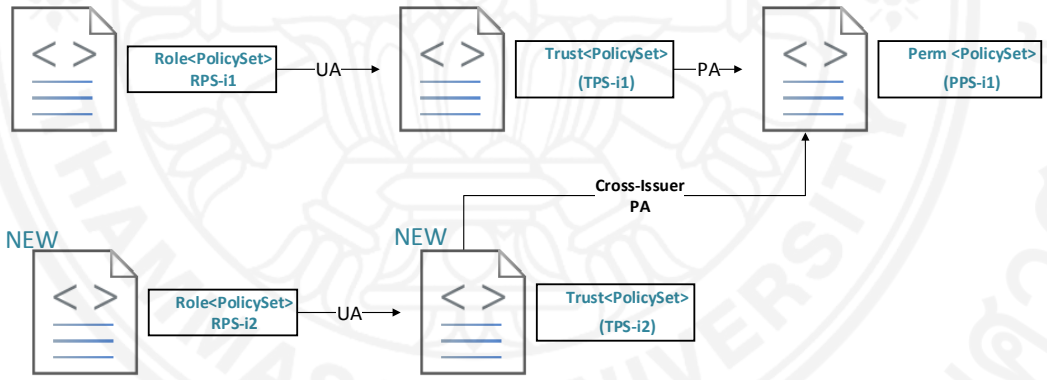
หากมีผู้ให้บริการอื่น ๆ เข้ามาร่วม จำเป็นต้องมีตัวบริการที่ทำหน้าที่เป็นชั้นกลางที่รองรับการทำงานในหลายการเช่า โดยในโมเดลที่รองรับลักษณะของผู้เช่าหลายราย อย่าง MTAS นั้น นโยบายที่รองรับการเรียกบริการแบบต่อเนื่อง หรือข้ามผู้ให้บริการ โดยในโมเดล MTAS ได้นิยามในเรื่องนี้ไว้ในเรื่องของการเชื่อถือ (trust) โดยเป็นการเชื่อกันระหว่างเจ้าของการเช่าให้สามารถใช้บทบาทที่เชื่อถือนั้นในการเข้าถึงข้อมูลของการเช่าของผู้เชื่อถือ (cross-issuer) (Tang, R. Sandhu, and Q. Li, 2013, pp.135) แต่ถ้าการ cross-issuer ที่มีลักษณะเป็นห่วงโซ่แบบประสานงาน โดยมีการเรียกบริการต่อเนื่องไปถึงทรัพยากรที่ต้องการนั้น ซึ่งไม่ได้กำหนดแค่ 1 issuer เชื่อกับคนละ issuer แต่เป็นลักษณะของการกำหนด trust ข้าม issuers ที่มีการเรียกผ่านไปหลาย ๆ ต่อกันไปนั้น ทำให้การกำหนดความเชื่อถือมีความยุ่งยากขึ้น เนื่องจากการกำหนดการเข้าถึง เป็นลักษณะของ XACML หรือ file policy โดย 1 บทบาทที่ถูกกำหนดขึ้น จะประกอบไปด้วย 3 ไฟล์ คือ RolePolicySet สำหรับการกำหนดชื่อบทบาทและ user ที่เข้าใช้บทบาทได้ TrustPolicySet สำหรับการกำหนด issuer ที่ใช้บทบาทนั้นได้ และ PermissionPolicySet สำหรับการกำหนดสิทธิ์ต่าง ๆ ที่จะสามารถใช้งานได้ ตามภาพที่ 1.6



ภาพที่ 1.5 แสดงการเรียกใช้บริการที่เป็นห่วงโซ่การเรียกบริการแบบประสานงาน

ภาพที่ 1.6 การกำหนด File Policy ของ XACML ในโมเดล MTAS ในการจัดการ 1 บทบาท จะประกอบไปด้วย 3 ไฟล์ RPS TPS และ PPS

และเมื่อต้องการกำหนดการจัดการ trust ข้ามผู้เช่าหรือ Cross-issuer เพิ่มเติม วิธีการคือ เจ้าของซอฟต์แวร์ต้องเพิ่มไฟล์ RolePolicySet และ TrustPolicySet และอ้างอิงจาก TrustPolicySet มายัง PermisionPolicySet เดิมที่ตนเองได้สร้างขึ้นไว้แล้ว ซึ่งจะเห็นได้ว่าต้องมีการสร้างไฟล์เพิ่มขึ้น 2 ไฟล์สำหรับการเพิ่ม cross-issuer 1 ครั้ง หากมีการเข้าถึงไฟล์ต่าง ๆ จำนวนมาก จะทำให้ต้องใช้ทรัพยากรสำหรับ IO เพิ่มมากขึ้นตามไปด้วย ส่งผลให้ประสิทธิภาพของการทำงานลดลง เพื่อจะตรวจสอบว่ามีการอนุญาตให้เข้าถึงหรือไม่ รวมทั้งซอฟต์แวร์ต่าง ๆ ที่เชื่อมโยงการเข้าถึงเข้าหากันจะมีรูปแบบการเรียกบริการที่คล้าย ๆ กัน หากเราแยกการกำหนดนโยบายสำหรับการเกิดห่วงโซ่การบริการแบบประสานงานออกจากนโยบาย และจัดการไว้เป็นอีกนโยบายหนึ่งโดยเฉพาะ ก็จะทำให้การใช้ทรัพยากร IO ในการเข้าถึงไฟล์นโยบายลดลง รวมทั้งการจัดการที่ง่ายขึ้นเพราะเป็นนโยบายที่ถูกกำหนดด้วยไฟล์เดียว ด้วยเหตุนี้ผู้วิจัยจึงได้ยกประเด็นของลักษณะห่วงโซ่การเรียกบริการแบบประสานงานเป็นหนึ่งในองค์ประกอบของงานวิจัยเพื่อเพิ่มประสิทธิภาพในการทำงานบนคลาวน์ และสามารถรวม SaaS ต่าง ๆ ที่มีอยู่มาใช้งานร่วมกันได้



ภาพที่ 1.7 การกำหนด File Policy ของ XACML ในโมเดล MTAS ในการจัดการ 1 บทบาทที่ Cross-issuer ให้กับอีก iseuer หนึ่ง

ดังนั้นงานวิจัยชิ้นนี้ ได้นำปัญหาในโมเดลของ MTAS เรื่องการกำหนดบทบาท ที่ผู้กำหนดบทบาทเป็นหน้าที่ของเจ้าของซอฟต์แวร์ที่ให้เช่าบริการ แต่ด้วยสัมพันธ์ของ Permission (P) ที่มีการกำหนด Tenant (T) ไว้ภายใต้การกำหนด Permission รวมทั้งความสัมพันธ์ของบทบาทไปยังสิทธิ์เป็นลักษณะของ many-many ทำให้เกิดความสัมพันธ์ของบทบาทหนึ่งถูกกำหนดให้เข้าถึงการเช่าซอฟต์แวร์ได้มากกว่า 1 การเช่าซอฟต์แวร์ ดังนั้นงานวิจัยชิ้นนี้จึงนำเสนอการแยก Tenant (T) ออกเป็นอีกองค์ประกอบหนึ่ง และกำหนดความสัมพันธ์ไปยังบทบาท สิทธิ์ และ Issuers ใหม่ เพื่อป้องกันไม่ให้เกิดการกำหนด Role ที่เข้าถึงการเช่าได้มากกว่า 1 การเช่านั้นเอง และในโมเดลดังกล่าวยังแก้ไขการกำหนดนโยบายสำหรับการเชื่อถือข้าม issuer ออกเป็นนโยบายแยกออกจากนโยบายต่าง ๆ เพื่อลดการแตกไฟล์ที่ยุ่งยากในการจัดการออกไป โดยได้ใช้

ชื่อโมเดลว่า “โมเดลการเรียกใช้บริการห่วงโซ่แบบประสานงานบนการทำงานในสภาพแวดล้อมของหลายผู้เช่า” (Chain of Calling Coordination in Multi-Tenancy Authorization: C-MTAS) ซึ่งได้อธิบายลักษณะของโมเดลในรูปแบบที่เป็นทางการ และได้นำโมเดลดังกล่าวมาปฏิบัติโดยใช้มาตรฐานของ eXtensible Access Control Markup Language (XACML) ซึ่งเป็นภาษา XML หนึ่งในการเขียนเพื่อการควบคุมการเข้าถึงของทรัพยากรต่าง ๆ ที่เป็นรูปแบบของ XML โดยมาตรฐานดังกล่าวรองรับการทำงานในลักษณะของ RBAC เบื้องต้น โดยผู้วิจัยได้ใช้พื้นฐาน XACML เวอร์ชั่น 3.0 และใช้ library ในการพัฒนาคือ Sun XACML เวอร์ชั่น 1.2

1.2 วัตถุประสงค์

จากบทเกริ่นนำในหัวข้อที่ 1.1 นั้น งานวิจัยชิ้นนี้จึงมีวัตถุประสงค์เพื่อ

(1) นำเสนอโมเดลที่รองรับการควบคุมการเข้าถึงในลักษณะของผู้เช่าหลายราย

(2) นำเสนอกระบวนการในการตรวจสอบการเข้าถึงในลักษณะของห่วงโซ่การเรียกแบบประสานงานในรูปแบบที่ไม่เกิด cycle

1.3 ขอบเขตของการศึกษา

ขอบเขตของการศึกษางานวิจัยชิ้นนี้

(1) ผู้วิจัยได้นำโมเดลของ MTAS มาเป็นพื้นฐาน และแก้ไของค์ประกอบของ Tenant โดยแยก Tenant ออกจากการกำหนด permission และกำหนดความสัมพันธ์ให้สอดคล้องใหม่

(2) โมเดลที่เสนอ มีทำงานในลักษณะของการใช้การบริการในรูปแบบของห่วงโซ่ของการเรียกใช้บริการแบบประสานโดย

ก. แยกองค์ประกอบของ Issuer (I) และผู้เช่า (T) ออกจากกัน และนิยามความสัมพันธ์ใหม่

ข. เพิ่มความชัดเจนในการกำหนดความเชื่อถือ ในฟังก์ชันของการจัดการความเชื่อถือ ให้กำหนดภายใต้บทบาทใดบทบาทหนึ่งด้วย

ค. นิยาม เซตของเจ้าของของผู้ใช้งาน จะนิยามโดยเป็นความสัมพันธ์ระหว่าง issuer และผู้ใช้งาน เนื่องจากจะต้องรองรับหากในแต่ละ issuer มีผู้เช่ามากกว่า 1 ผู้เช่า ทำให้ผู้ใช้งานภายใต้ issuer จะสามารถใช้งานผู้เช่าที่ภายใต้ issuer ได้ด้วยเช่นกัน

ง. นิยามฟังก์ชั่นในการตรวจสอบการเป็นเจ้าของในการเช่า รวมทั้งการตรวจสอบสิทธิ์ในการได้รับจากความเชื่อถือของเจ้าของในการเช่า เพื่อถ่ายทอดการกำหนดสิทธิ์ดังกล่าวให้กับผู้ถูกเชื่อถือสามารถกำหนดสิทธิ์ในการเข้าถึงได้

จ. รูปแบบของห่วงโซ่ของการเรียกใช้บริการแบบประสานนั้น เป็นรูปแบบที่ไม่เกิด Cycle โดยผู้วิจัยได้นำ Algorithm สำหรับการตรวจสอบ Cycle ในกราฟเส้นทางเดียว ของ Szwarcfiter (Szwarcfiter, 1974) มา implement ในการตรวจสอบการเกิด cycle

(3) ขอบเขตในส่วนของการพัฒนาระบบ ในโมเดลที่เสนอ จะพัฒนาโดยใช้ eXtensible Access Control Markup Language (XACML) โดยใช้เวอร์ชั่น XACML 3.0

ตามที่ OASIS (OASIS, 2010) ได้เสนอไว้ โดย XACML ถูกนำมาใช้เป็นนโยบายในการกำหนดควบคุมการเข้าถึง และนำคลังของ Java ชื่อ Sun XACML 1.2 เพื่อนำโมเดล C-MTAS มาปฏิบัติให้เห็นเป็นรูปธรรม

(4) โดยหลังจากที่นำ XACML มา implement แล้ว ผู้วิจัยได้ทดสอบการเรียกใช้บริการข้ามกันระหว่างเจ้าของ รวมทั้งในรูปแบบของห่วงโซ่การเรียกใช้บริการแบบประสานงานโดยใช้ตัวอย่างที่จะกล่าวถึงในบทที่ 3 ต่อไป

(5) ผู้วิจัยได้ทดลองในสภาพแวดล้อมของคลาวน์ ในลักษณะของการให้บริการในส่วนของซอฟแวร์ (SaaS) เท่านั้น

(6) สรุปผลวิเคราะห์ในปัจจัยของการปรับขนาดของฮาร์ดแวร์ในการ implement โดยเปรียบเทียบการเรียกใช้บริการในสภาพแวดล้อมของฮาร์ดแวร์ที่แตกต่างกัน เพื่อเปรียบเทียบขนาดของฮาร์ดแวร์ในรูปแบบใดที่มีประสิทธิภาพในเรื่องของการใช้ระยะเวลาเฉลี่ยในการตอบสนอง ค่าภาระการรองรับการทำงาน ในตัวอย่างที่นำมาศึกษา

1.4 ประโยชน์ที่คาดว่าจะได้รับ

(1) ทำให้เกิดโมเดลที่รองรับการกำหนดสิทธิในรูปแบบของผู้เช่าหลายรายในสภาพแวดล้อมบนคลาวน์ โดยบทบาทที่ถูกกำหนดขึ้น ผู้กำหนดคือเจ้าของซอฟต์แวร์

(2) ทำให้เกิดโมเดลที่รองรับการกำหนดสิทธิ์ในลักษณะของห่วงโซ่การบริการแบบประสานงาน

บทที่ 2
วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

ในบทนี้ จะแบ่งออกเป็น 2 ส่วนคือ

ส่วนที่ 1 เป็นการทบทวนทฤษฎีที่เกี่ยวข้อง ศึกษาความหมายของการประมวลผลบนคลาวน์ รวมทั้งการแบ่งประเภทต่าง ๆ ของคลาวน์ เพื่อให้เกิดความเข้าใจในการทำงานของการประมวลผลบนคลาวน์ และภาษา eXtensible Access Control Markup Language (XACML)

ส่วนที่ 2 เป็นการทบทวนงานวิจัยที่เกี่ยวข้อง แบ่งได้เป็น 2 ส่วน ส่วนแรกศึกษาการตรวจสอบสิทธิ์การเข้าถึงของโมเดลการควบคุมการเข้าถึงตามบทบาท (Role-based Access Control) และส่วนที่สอง ได้อธิบายลักษณะที่เกี่ยวข้องกับกระบวนการของห่วงโซ่การเรียกใช้บริการแบบประสานงาน โดยอธิบายลักษณะรูปแบบของห่วงโซ่ในรูปแบบต่าง ๆ รวมทั้งการจัดการกับการตรวจสอบสิทธิ์การเข้าถึงในรูปแบบของห่วงโซ่

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 การประมวลผลบนกลุ่มเมฆ หรือคลาวน์ (Cloud Computing)

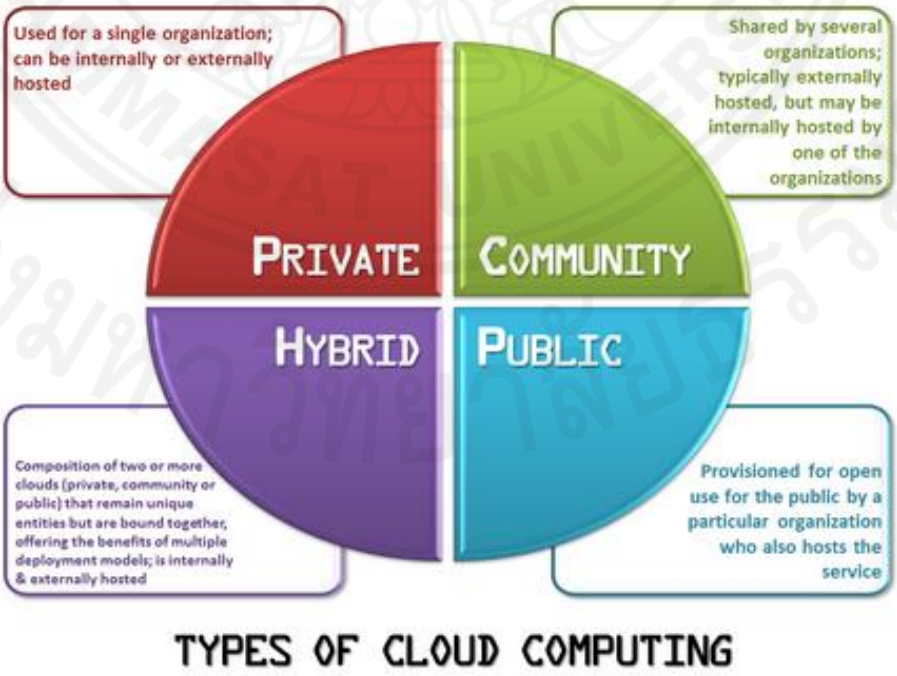
National Institute of Standards and Technology (NIST) (Mell & Grance, 2011) ได้ให้คำจำกัดความของการประมวลผลบนคลาวน์มีใจความว่า “เป็นตัวแบบที่ใช้งานกันอย่างแพร่หลาย เพื่อความสะดวกสบายในการใช้งานสำหรับเครือข่ายที่ใช้ร่วมกัน การใช้งานทรัพยากรทางด้านคอมพิวเตอร์ เช่น เครือข่าย เซิร์ฟเวอร์ พื้นที่ในการจัดเก็บข้อมูล แอพพลิเคชันและการบริการ เป็นต้น ที่สามารถที่จะดำเนินการจัดเตรียมเพื่อให้ผู้ใช้งานสามารถใช้งานได้อย่างรวดเร็วและเป็นไปตามที่ผู้ใช้งานต้องการ”

การประมวลผลบนคลาวน์นั้น เป็นการนำพื้นฐานกรอบความคิดในเรื่องของเวอร์ชัวไลเซชันและเว็บเซอร์วิสมาประกอบเข้าด้วยกัน โดยให้ผู้ต้องการใช้บริการคลาวน์ สามารถจัดการทรัพยากรที่ผู้ใช้ต้องการได้เอง ซึ่งการให้บริการบนคลาวน์นั้นสามารถจำแนกโดยแยกตามรูปแบบที่ให้บริการ ได้เป็น 3 ประเภทตาม NIST เพื่อที่จะสามารถอธิบายชั้นการทำงานของคลาวน์ได้ชัดเจน

(1) การให้บริการในส่วนของซอฟแวร์ หรือ Software as a Service (SaaS) เป็นการให้บริการกับผู้ใช้บริการที่ใช้งานโปรแกรมที่ทำงานบนโครงสร้างพื้นฐานของคลาวน์ โดยการใช้งานสามารถเข้าถึงได้โดยใช้อุปกรณ์ได้หลากหลาย ซึ่งอาจผ่านทางอินเทอร์เฟสของอุปกรณ์เอง เช่น เว็บเบราว์เซอร์ (อาทิเช่น จดหมายอิเล็กทรอนิกส์บนเว็บ) หรือผ่านทางอินเทอร์เฟสของโปรแกรม โดยผู้ใช้บริการไม่ได้จัดการหรือควบคุมโครงสร้างพื้นฐานของคลาวน์ ทั้งในด้านของเครือข่าย เซิร์ฟเวอร์ ระบบปฏิบัติการ หรือพื้นที่จัดเก็บข้อมูล

(2) การให้บริการในส่วนของแพลตฟอร์ม หรือ Platform as a Service (PaaS) เป็นการให้บริการกับผู้ใช้ที่บริการที่ต้องการใช้งานบนโครงสร้างพื้นฐานของคลาวน์ ในสิ่งที่ผู้ใช้งานสร้างไว้อยู่แล้ว หรือเป็นการใช้ภาษาสร้างโปรแกรม คลัง (library) หรือเครื่องมือที่สนับสนุนของผู้ให้บริการเอง โดยผู้ใช้บริการไม่ต้องจัดการและหรือควบคุมโครงสร้างพื้นฐานของคลาวน์ ในด้านของเครือข่าย เซิร์ฟเวอร์ ระบบปฏิบัติการ หรือพื้นที่จัดเก็บ แต่จะสามารถควบคุมการนำโปรแกรมของตนขึ้นมาใช้บนคลาวน์ รวมถึงในด้านของการตั้งค่าต่าง ๆ สำหรับสภาพแวดล้อมในการใช้งานโปรแกรมนั้น ๆ

(3) การให้บริการในส่วนของสถาปัตยกรรม หรือ Infrastructure as a Service (IaaS) เป็นการให้บริการกับผู้ใช้บริการ ในด้านของการจัดการการประมวลผล การจัดเก็บข้อมูล เครือข่าย และทรัพยากรอื่น ๆ ที่จำเป็นในการใช้คอมพิวเตอร์พื้นฐาน โดยผู้ใช้งานสามารถปรับใช้และเรียกใช้ซอฟต์แวร์ได้เอง ผู้ใช้บริการไม่ต้องจัดการหรือควบคุมโครงสร้างพื้นฐานของคลาวน์ แต่จะสามารถควบคุมระบบปฏิบัติการ พื้นที่ในการจัดเก็บ และนำระบบมาเข้าบนคลาวน์ได้ รวมทั้งถึงการควบคุมระบบเครือข่าย เช่น ไฟล์วอล



ภาพที่ 2.1 แผนการจำแนกประเภทของประมวลผลบนคลาวน์ โดยแยกจากการใช้งาน
ที่มา: http://ohioerc.org/?page_id=187

และสำหรับประเภทของการนำการประมวลผลบนคลาวน์ไปใช้งาน จะแยกได้เป็น 4 ประเภท (ดังแสดงในภาพที่ 2.1) กล่าวคือ

(1) Private Cloud เป็นโครงสร้างของคลาวน์ที่จัดตั้งขึ้นมาเพื่อรองรับการทำงานของ องค์กรใดองค์กรหนึ่ง ไม่ว่าจะเป็นการดูแลภายใต้ขององค์กรนั้น หรือบุคคลที่ 3 ก็ตามและไม่ว่าจะเป็นการเช่าไว้ภายในองค์กรเอง หรือนอกองค์กรก็ตาม

(2) Public Cloud เป็นโครงสร้างของคลาวน์ที่บริการ เมื่อผู้ร้องขอใช้งานผ่านเครือข่ายโดยผ่านอินเทอร์เน็ต ก็สามารถเข้าถึง Public Cloud ที่เปิดบริการได้ อาทิเช่น Amazon (AWS) Microsoft (Azure) และ Google

(3) Community Cloud เป็นโครงสร้างของคลาวน์ที่ใช้ร่วมกันระหว่างหลาย ๆ องค์กรที่มีความเกี่ยวข้องกัน อาทิเช่น ความมั่นคง กฎข้อบังคับ อำนาจ เป็นต้น ไม่ว่าจะเป็นการดูแลขององค์กรเหล่านั้นเอง หรือบุคคลที่ 3 และไม่ว่าจะเป็นการเช่าไว้ภายในองค์กร หรือนอกองค์กรก็ตาม

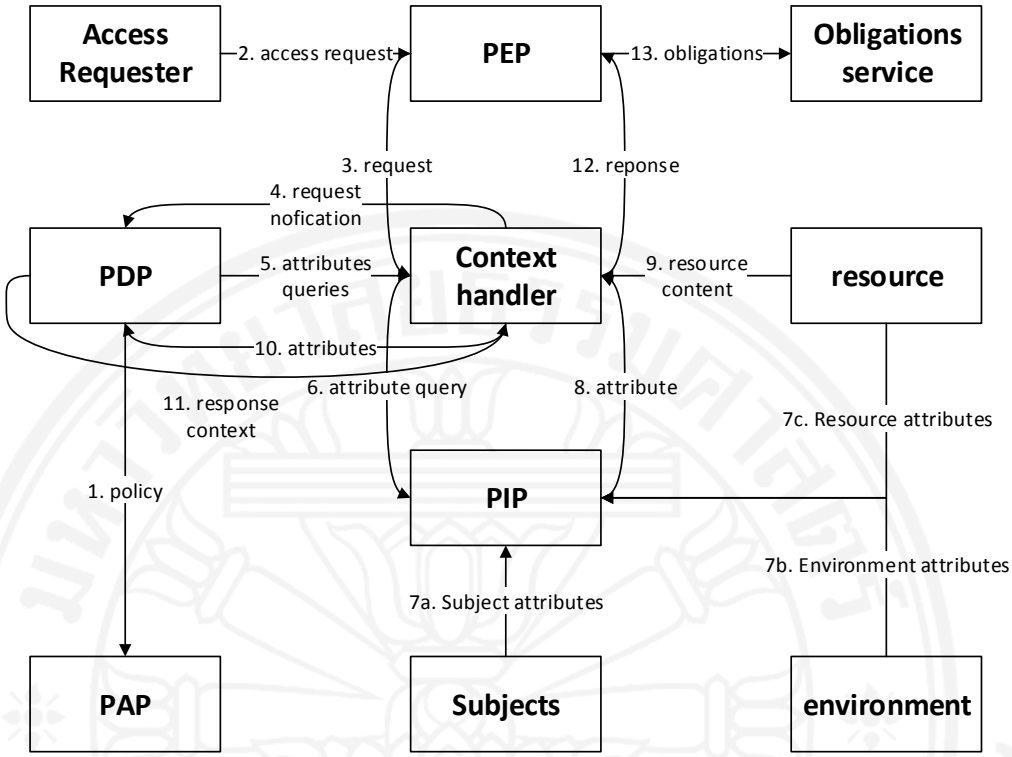
(4) Hybrid Cloud คือ คือ เป็นโครงสร้างของคลาวน์ที่ประกอบไปด้วยการร่วมกันของ ส่วนของ Private Cloud, Public Cloud หรือ Community Cloud จากผู้ให้บริการที่ต่างกัน หรือเป็นการนำระบบบางส่วนเข้าไปทำงานบนคลาวน์ก็ได้

จากประเภทที่แบ่งตามรูปแบบที่ให้บริการ งานวิจัยชิ้นนี้ได้สนใจในส่วนของการประมวลผลบนคลาวน์ในรูปแบบของซอฟต์แวร์ ซึ่งจากที่ผู้วิจัยได้เสนอในบทที่ 1 ว่าความต้องการที่เห็นได้ชัดในปัจจุบันของรูปแบบบนคลาวน์นั้น ทำให้ SaaS เป็นสิ่งที่น่าสนใจมากกว่า PaaS และ IaaS อีกทั้งความต้องการของผู้ใช้งานที่ต้องการใช้บริการซอฟต์แวร์ที่ใช้งานได้มากยิ่งขึ้นกว่าอดีต ซึ่งการยอมรับการใช้งานบนคลาวน์ในปัจจุบันจะเห็นได้ว่ามีการยอมรับเพิ่มมากขึ้น ดังจากเห็นได้จากการต่ออายุของการใช้ในภาพที่ 1.1

2.1.2 eXtensisible Access Control Markup Language (XACML)

XACML เป็นมาตรฐานที่ถูกกำหนดขึ้นมาเพื่อใช้ในการกำหนดนโยบายในการควบคุมการเข้าถึง ด้วยการใช้ XML เป็นภาษาพื้นฐาน โดยประกอบกันขึ้นเรียกว่า นโยบาย (Policy) เพื่อใช้ในการกำหนด Attribute Based Access Control (ABAC) หรือนำนโยบายมาประกอบกันมากขึ้น ซึ่งสามารถนำมาใช้ใน Role-based Access Control (RBAC) ได้ด้วยเช่นเดียวกัน (OASIS, 2010)

มาตรฐาน XACML ที่ถูกยอมรับและนำมาใช้งานกันนั้นเป็นการกำหนดมาตรฐานภายใต้องค์กร OASIS ซึ่งกระบวนการทำงานของ XACML นั้นประกอบขึ้นตามภาพที่ 2.2 โดยมีองค์ประกอบหลัก คือ



ภาพที่ 2.2 แผงผังการไหลผ่านของข้อมูลในการทำงานของ XACML

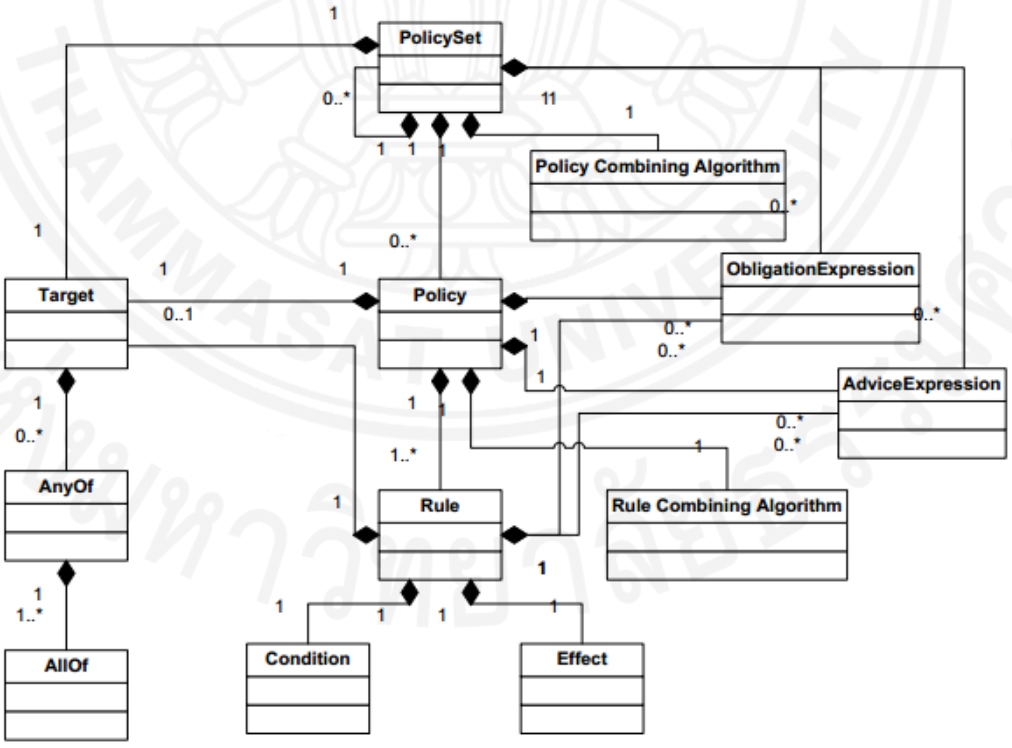
ที่มา: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf."

- **Access Requester** เป็นผู้ร้องขอในการตรวจสอบสิทธิ์การเข้าถึง
- **Policy Enforcement Point (PEP)** เป็นส่วนที่ไว้สำหรับการบังคับเพื่อให้มีการตรวจสอบสิทธิ์การเข้าถึง ตามที่ผู้ร้องส่งความต้องการเข้ามา
- **Context handler** เป็นกระบวนการในตัดสินใจที่ส่งไปยัง PDP และส่งทรัพยากรต่าง ๆ ที่ PDP ต้องการไปให้เพื่อให้ PDP ประกอบการตัดสินใจสำหรับการร้องขอของ access requester มา
- **Policy Decision Point (PDP)** เป็นส่วนที่ไว้ตัดสินใจในการร้องขอในการเข้าถึงทรัพยากรที่ต้องการ โดยเมื่อ context handler ส่งทรัพยากรทั้งหมดที่ใช้ประกอบการตัดสินใจให้กับ PDP ซึ่งประกอบการพิจารณาผลลัพธ์โดยใช้ policy เป็นหลัก
- **Policy Administration Point (PAP)** เป็นส่วนที่ระบบสร้างนโยบายหรือเซตของนโยบายเพื่อส่งมายัง PDP ตัดสินใจ

- **Policy Information Point (PIP)** เป็นส่วนที่ระบบดึงข้อมูลสำหรับคุณสมบัติของทรัพยากรต่าง ๆ เพื่อนำมาประกอบให้กับ PDP ในการตัดสินใจ โดยจะใช้ส่วนของ resource subject และ environment
- **Subject** แทนบุคคล หรือตัวกระทำที่ถูกเชื่อในการใช้งานระบบ
- **Resource or Action** แทนข้อมูล การบริการ หรือระบบที่ใช้งาน
- **Environment** แทนเซตของคุณลักษณะที่ใช้ในการประกอบการตัดสินใจในการตรวจสอบการเข้าถึง โดยใช้ความสัมพันธ์ของ subject resource หรือ action

สำหรับโครงสร้างของภาษา XACML นั้นการประกอบขึ้นมาของนโยบาย หรือ Policy จะใช้องค์ประกอบด้วย 3 ส่วนหลัก คือ Rule Policy และ Policy Set และได้ใช้ความสัมพันธ์ของส่วนต่าง ๆ ตามภาพที่ 2.3

โดย Rule เป็นการบอกถึงกฎสำหรับการเข้าถึงทรัพยากรใด ๆ ว่ามีการอนุญาต หรือไม่อนุญาต สำหรับ Policy เป็นการกำหนดนโยบายที่อ้างไปยัง Rule และสำหรับ PolicySet ถือการรวมขึ้นกันของ Policy นั้นเอง



ภาพที่ 2.3 โมเดลสำหรับภาษาในการเขียนนโยบาย XACML
ที่มา: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf.

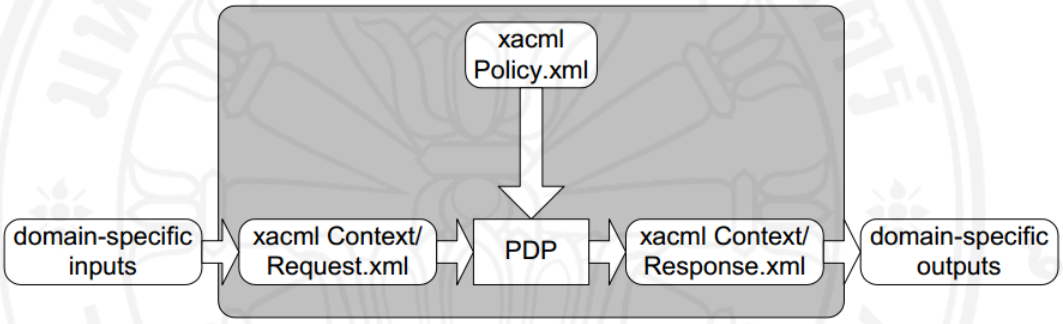
และในขั้นตอนการทำงานของ XACML ซึ่งสามารถสรุปเป็นขั้นตอนอย่างง่าย ได้กระบวนการตามภาพที่ 2.4 ซึ่งเริ่มต้นตั้งแต่เมื่อมีการร้องขอ เพื่อตรวจสอบการเข้าถึง จะถูกส่งไปให้ PDP ในการตัดสินใจ โดย PDP จะตัดสินใจจากนโยบายที่ถูกสร้างไว้ หรือที่เรียกว่า Policy หลังจากนั้น PDP จึงส่งผลของการตัดสินใจไปให้กับผู้ร้องขอ ซึ่งผลลัพธ์ จะแบ่งออกได้เป็น 4 อย่างคือ

(1) **Permit** หมายถึง การมีสิทธิ์ในการเข้าถึงตามที่ร้องขอ

(2) **Deny** หมายถึง ไม่มีสิทธิ์ในการเข้าถึงตามที่ร้องขอ

(3) **Not Applicable** หมายถึง ไม่สามารถตัดสินใจได้ เกิดขึ้นจากมีการกำหนดนโยบายบางประการที่ไม่ชัดเจน

(4) **Indeterminate** หมายถึง ไม่สามารถตัดสินใจได้ ไม่มีการกำหนดนโยบายที่เกี่ยวข้องตามที่ร้องขอ



ภาพที่ 2.4 กระบวนการของ XACML
ที่มา: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf."

2.2 งานวิจัยที่เกี่ยวข้อง

ในส่วนของงานวิจัยที่เกี่ยวข้อง ผู้วิจัยได้ศึกษาใน 2 องค์ประกอบ คือ ส่วนของโมเดลการควบคุมการเข้าถึงตามบทบาท (Role-based Access Control) และลักษณะที่เกี่ยวข้องกับห่วงโซ่การเรียกบริการแบบประสานงาน

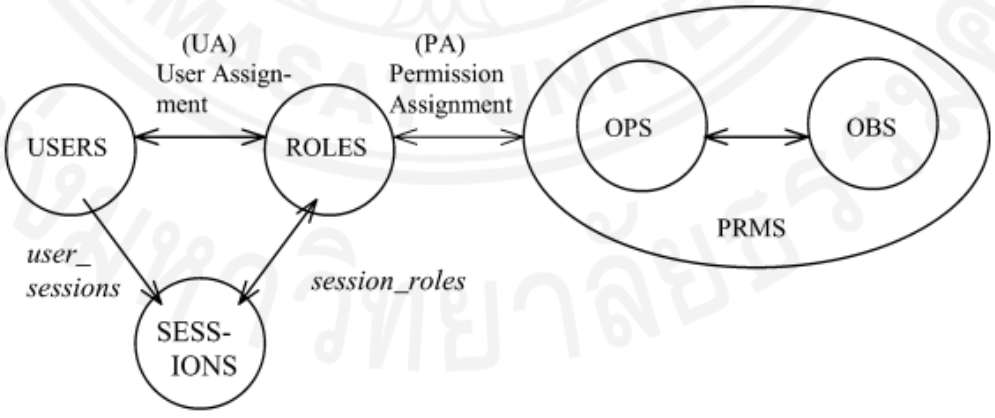
2.2.1 โมเดลการควบคุมการเข้าถึงตามบทบาท (Role-based Access Control: RBAC)

โมเดลการควบคุมการเข้าถึงตามบทบาท หรือที่เรียกกันว่า RBAC มีการนำโมเดลนี้มาใช้ในส่วนของการควบคุมการเข้าถึง (Access Control) กันอย่างแพร่หลายและเป็นที่ยอมรับในวงกว้าง (Sandhu, Ferraiolo, and Gavrila, 2000, pp. 224-274) โดยในปีค.ศ. 2001 ได้มีการนำเสนอมาตรฐานของ RBAC ถูกนำเสนอโดย NIST ซึ่งได้อธิบายองค์ประกอบ และความสัมพันธ์ต่าง ๆ ของโมเดล RBAC รวมทั้งมีการกำหนดลำดับชั้นของบทบาท และข้อจำกัดเพื่อไม่ให้โมเดลเกิดความขัดแย้งกันเองในการกำหนดบทบาท และต่อมาเมื่อสภาพแวดล้อมของการใช้งาน มีความซับซ้อนและความหลากหลายของทรัพยากรที่ต้องจัดการมากขึ้น ทำให้จากพื้นฐานของโมเดล RBAC ถูกนำมาเพิ่มเติมเพื่อให้รองรับความต้องการเหล่านั้น โดยผู้วิจัยได้นำงานวิจัยบางส่วนในการดัดแปลงโมเดล RBAC เพื่อต่อเติมให้รองรับความต้องการเหล่านั้นในบทนี้ โดยแบ่งออกเป็น 3 หัวข้อ คือ NIST_RBAC MTAS เป็นโมเดลที่ผู้วิจัยได้นำมาเป็นพื้นฐานในการแก้ไขปัญหาที่เกิดขึ้น ดังที่อธิบายในบทที่ 1 และสุดท้ายคืองานวิจัยบางส่วนที่ได้นำเสนอโดยใช้โมเดล RBAC เป็นพื้นฐาน

2.2.1.1 NIST RBAC

โดยงานวิจัยชิ้นนี้ได้นำโมเดลการควบคุมการเข้าถึงตามบทบาท (RBAC) โดยอธิบาย NIST RBAC ในโมเดลของ NIST RBAC นั้นเป็นการอธิบาย 4 ส่วน คือ องค์ประกอบหลักของ RBAC(Core RBAC) ลำดับชั้นใน RBAC ข้อบังคับของ RBAC (Constraint RBAC) และฟังก์ชันที่ใช้ในการจัดการกับองค์ประกอบทั้งหมดของ RBAC

ส่วนที่หนึ่ง องค์ประกอบหลักของ RBAC (Core RBAC)



ภาพที่ 2.5 แสดงให้เห็นถึง Core RBAC
ที่มา: “The NIST Model for Role – Based Access Control,”by D.F. Ferraiolo, R. Sandhu, and S.Gavrila, pp. 224-274. Copyright 2001 by ACM.

สำหรับ Core RBAC กล่าวถึงองค์ประกอบ และความสัมพันธ์ตามที่กำหนดไว้ในภาพที่ 2.5 ซึ่งประกอบไปด้วย 5 องค์ประกอบหลัก คือ ผู้ใช้งาน (Users) บทบาท (Roles) เซสชั่น (Sessions) วัตถุ (OBS) และการอนุญาต (OPS) และความสัมพันธ์ 2 ส่วนคือ User Assignment (UA) เป็นความสัมพันธ์ระหว่างผู้ใช้งานไปยังบทบาท และ Permission Assignment (PA) เป็นความสัมพันธ์ระหว่างบทบาทไปการอนุญาตสำหรับวัตถุต่าง ๆ ซึ่งอธิบายในรูปแบบทางการได้ตามตารางที่ 2.1

ตารางที่ 2.1

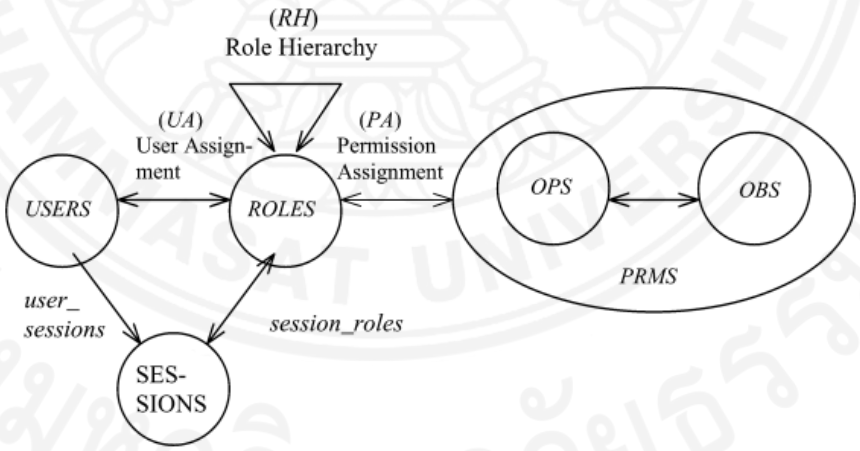
แสดงคำนิยามของ Core RBAC

คำนิยาม	**ความหมาย**
$USERS$	เซตของผู้ใช้งาน ย่อด้วย U
$ROLES$	เซตของบทบาท ย่อด้วย R
OPS	เซตของโอเปอเรชั่นที่ผู้ใช้งานสามารถทำได้
OBS	เซตของวัตถุที่เกี่ยวข้องกับโอเปอเรชั่น
$SESSIONS$	เซตที่เก็บเซสชั่นการเข้าใช้งานของผู้ใช้งาน
$UA \subseteq USERS \times ROLES$	เซตความสัมพันธ์ระหว่างผู้ใช้งานกับบทบาท
$assigned\_users: (r: ROLES) \rightarrow 2^{ROLES}$	ฟังก์ชันในการเซตของผู้ใช้งานที่มีบทบาท r โดยเขียนได้เป็น $assigned\_users\ (r) = \{u \in USERS \mid (u, r) \in UA\}$
$PRMS = 2^{(OPS \times OBS)}$	เซตที่กำหนดการอนุญาตโดยระบุความสัมพันธ์ระหว่างโอเปอเรชั่นและวัตถุที่เกี่ยวข้องกับโอเปอเรชั่น
$PA \subseteq PRMS \times ROLES$	เซตความสัมพันธ์ระหว่างสิทธิ์ในการใช้กับบทบาท
$assigned\_permissions: (r: ROLES) \rightarrow 2^{PRMS}$	ฟังก์ชันในการหาเซตของ PA ที่สามารถใช้บทบาท r ได้ โดยเขียนได้เป็น $assigned_{permissions}(r) = \{p \in PRMS \mid (p, r) \in PA\}$

คำนิยาม	ความหมาย
$Ob(p: PRMS) \rightarrow \{op \subseteq OPS\}$	ฟังก์ชันในการหาความสัมพันธ์ของ permission กับโอเปอเรชั่นได้รับการอนุญาต
$Ob(p: PRMS) \rightarrow \{ob \subseteq OBS\}$	ฟังก์ชันในการหาความสัมพันธ์ของ permission กับวัตถุที่ได้รับการอนุญาต
$user\_sessions\ (u: USERS) \rightarrow 2^{SESSIONS}$	ฟังก์ชันการหาผู้ใช้งาน (u) ที่ใช้งานอยู่
$session\_role\ (s: SESSIONS) \rightarrow 2^{ROLES}$	ฟังก์ชันการหาบทบาท (r) ที่ใช้งานอยู่
$avail\_session\_perms\ (s: SESSIONS) \rightarrow 2^{PRMS}$	ฟังก์ชันการหาสิทธิ์ (p) ที่ใช้งานอยู่

ส่วนที่สอง ลำดับชั้นใน RBAC (Hierarchal RBAC)

เป็นการอธิบายเซตของลำดับชั้นในการกำหนดบทบาทที่ถูกนำบทบาทที่เหมือนกัน มากำหนดในรูปแบบของการสืบทอด จากบทบาทที่มีอยู่แล้ว แทนที่จะกำหนดบทบาทที่มีความซ้ำซ้อนกัน โดยอธิบายได้ในลักษณะตามภาพที่ 2.6



ภาพที่ 2.6 แสดงให้เห็นถึง Hierarchai RBAC

ที่มา: "The NIST Model for Role – Based Access Control," by D.F. Ferraiolo, R. Sandhu, and S.Gavrila, pp. 224-274. Copyright 2001 by ACM.

ตารางที่ 2.2

แสดงความสัมพันธ์ของ RH

คำนิยาม	ความหมาย
$RH \subseteq ROLES \times ROLES$	ความสัมพันธ์ในการกำหนดลำดับชั้นของบทบาท โดยใช้เครื่องหมาย $\geq$ เมื่อ $r_1 \geq r_2$ แสดงว่าการอนุญาตทั้งหมดของ r_2จะสืบทอดไปเป็นการอนุญาตทั้งหมดของ r_1และผู้ใช้งานทั้งหมดที่ใช้ r_1 เป็นผู้ใช้งานของ r_2 ด้วย โดยเขียนได้ว่า $r_1 \geq r_2 => authroized_{permissions(r_2)} \subseteq authorized_{permissions(r_1)} \wedge authorization_{users(r_1)} \subseteq authorization_{users(r_2)}$
$authoized\_users(r: ROLES) \rightarrow 2^{USERS}$	ฟังก์ชันในการหาผู้ใช้งาน ที่สามารถใช้บทบาท r ที่อยู่ในกำหนดลำดับชั้นนของ RH ได้โดยเขียนได้เป็น $authoized\_users(r) = \{u \in USERS \mid r' \geq r\ (u, r') \in UA\}$

สำหรับ RH นั้นในได้ถูกแบ่งออกเป็น 2 ประเภท คือ limited role hierarchy คือเป็นการจำกัด เพื่อให้การสืบทอดบทบาทไปได้แค่บทบาทเดียวเท่านั้น และ general role hierarchy เป็นการกำหนดลำดับชั้นของบทบาทโดยไม่มีการจำกัดในการสืบทอด

ส่วนที่สาม ข้อบังคับของ RBAC (Constraint RBAC) เพื่อเป็นข้อบังคับในการใช้งานบน RBAC โดยมีรายละเอียดแบ่งออกเป็น 2 ความสัมพันธ์ คือ

- **Static Separation of Duty Relations (SSD)**

เป็นกฎข้อบังคับเพื่อใช้ในการการกำหนดความสัมพันธ์ของ UA และ RH ไม่ให้เกิดการขัดแย้งกันเองของการให้บทบาท กล่าวได้คือ ถ้าหากมีการกำหนดเป็นเซตของ SSD แล้ว โดยผู้ใช้งานที่ได้รับบทบาทหนึ่ง ซึ่งบทบาทดังกล่าวมีกำหนดว่าจะไม่สามารถใช้บทบาทอีกบทบาทหนึ่งได้ ทำให้ในขณะที่ให้บทบาทนั้นกับผู้ใช้งาน จะไม่สามารถให้อีกบทบาทที่ขัดแย้งดังกล่าวได้

- **Dynamic Separation of Duty Relations (DSD)**

เป็นกฎข้อบังคับเพื่อใช้ในการกำหนดความสัมพันธ์ในเซตของ Session ที่กำหนดไว้ในเซตของ DSD เพื่อตรวจสอบบทบาทเพื่อถูกกำหนดนั้น จะต้องไม่ให้สามารถใช้ภายใต้บทบาทพร้อมกันได้ในช่วงระยะเวลาเดียวกัน

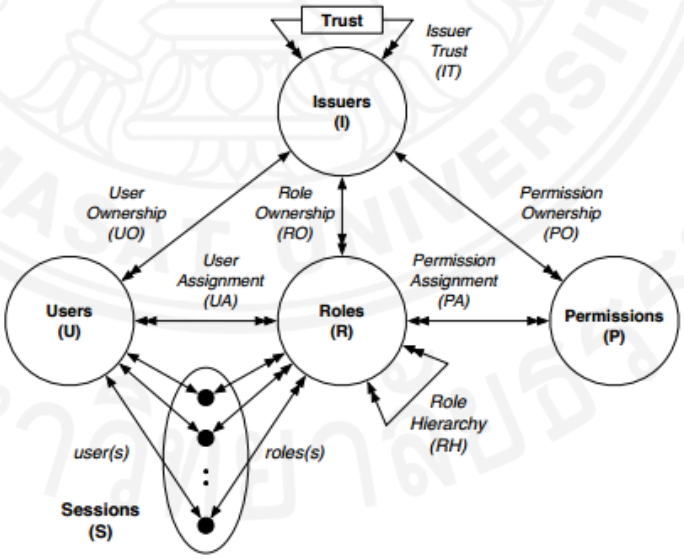
ส่วนที่สี่ ฟังก์ชันในการจัดการขององค์ประกอบและความสัมพันธ์ต่าง ๆ จากทั้งสามองค์ประกอบหลัก คือ Core RBAC, Hierarchal RBAC และข้อบังคับของ RBAC โดยในส่วนที่ 4 นี้ ได้แสดงถึงฟังก์ชันต่าง ๆ ที่ทำหน้าที่ในการจัดการเพื่อเงื่อนไขสำหรับกำหนดองค์ประกอบและความสัมพันธ์ต่าง ๆ

2.2.1.2 Administrative Multi-Tenant Authorization System (AMTAS)

ได้กล่าวถึงโมเดลของ MTAS ในลักษณะที่มีรูปแบบที่เป็นทางการ รวมทั้งได้กำหนดฟังก์ชันในการจัดการพื้นฐานของโมเดล MTAS ซึ่งเรียกโมเดลใหม่นี้ว่า AMTAS โดยแบ่งเนื้อหาออกได้เป็น 3 ส่วน คือ

ส่วนที่หนึ่ง การอธิบายลักษณะของโมเดล MTAS

โดยระบบ MTAS จะประกอบไปด้วย 5 องค์ประกอบ คือ ประเด็น (Issuers:I) ผู้ใช้งาน (Users:U) การอนุญาต (Permissions:P) บทบาท (Roles:R) และเซสชั่น (Session:S) รวมทั้ง 1 ความสัมพันธ์ที่ถูกเพิ่มเติมจาก RBAC คือ ความเชื่อถือของ Issuer (Issuers Trust:IT) รายละเอียดตามตารางที่ 2.3



ภาพที่ 2.7 โมเดลของระบบ MTAS

ที่มา: “Multi-Tenancy Authorization Models for Collaborative Cloud Services,” by B. Tang, R. Sandhu, and Q. Li, pp. 229-238. Copyright 2013 by IEEE.

ตารางที่ 2.3

องค์ประกอบของโมเดล MTAS

คำนิยาม	คำอธิบาย
Issuers	**เจ้าของการเช่า** ใช้แทนองค์กร หรือบุคคลที่ใช้บริการคลาวน์ โดยบริการจะสร้างอินเทอร์เฟส เรียกว่า ผู้เช่า (tenant) ในแต่ละ issuer ดังนั้นข้อมูลและการกระทำของในแต่ละ issuer จะถูกแยกออกจาก issuer อื่น ๆ อย่างชัดเจน
Users	**ผู้ใช้งาน** แทนถึงตัวบุคคล โดยการพิสูจน์ตัวจริงจะถูกทำจากภายนอกเพื่อให้ผู้ใช้งานไม่ซ้ำกันของทุก ๆ เจ้าของการเช่า (Issuers) เป็นการใช้แทนองค์กร หรือบุคคลที่ใช้บริการคลาวน์ โดยบริการจะสร้างอินเทอร์เฟส ซึ่งเรียกว่า ผู้เช่า (tenant) ในแต่ละ issuer ดังนั้นข้อมูลและการกระทำของในแต่ละ issuer จะถูกแยกออกจาก issuer อื่น ๆ อย่างชัดเจน
Permissions	**การอนุญาต** เป็นข้อจำกัดในการให้สิทธิ์ เพื่อเข้าใช้ในวัตถุ ของผู้เช่า ซึ่งจะทำงานผ่านอินเทอร์เฟส โดยการอนุญาต จะประกอบกันด้วย 3 ส่วนคือ สิทธิ์, ผู้เช่า, วัตถุที่ให้สิทธิ์ โดยการอนุญาตสำหรับผู้เช่าหนึ่ง ๆ นั้นจะต้องอยู่ภายใต้ของ issuer เพียง issuer เดียวเท่านั้น และทุก ๆ การอนุญาตจะเกี่ยวข้องเพียง issuer เดียวเท่านั้นด้วย แต่ใน issuer หนึ่ง ๆ สามารถมีการอนุญาตได้มากกว่าหนึ่งด้วยเช่นกัน
Roles	**บทบาท** เป็นฟังก์ชันของหน้าที่ หรือจะได้กล่าวว่าเป็นชื่อของหน้าที่ ของ issuer นั้น ๆ โดยการแทนของบทบาทจะประกอบไปด้วย role (issuer , ชื่อบทบาท) ซึ่งบทบาทหนึ่งบทบาทจะนิยามขึ้นใช้ได้เพียงหนึ่ง issuer เท่านั้น แต่ issuer หนึ่ง ๆ มีได้หลายบทบาทเช่นกัน
Sessions	**เซสชั่น** เป็นค่าคงที่ ที่ถูกเก็บกิจกรรมต่าง ๆ ของการเข้าใช้งานของผู้ใช้งาน
Issuer Trust (IT)	**ความเชื่อมั่นในแต่ละ issuer** ซึ่งเกิดจากเซตของ I คูณกับ I โดยมีเงื่อนไขความเชื่อมั่นเป็นกลไกในการบอกความสัมพันธ์ ซึ่งเขียนในรูปสมการได้เป็น ($IT \subseteq I \times I$, โดยจะเขียนในรูปของ " $\lesssim$ ") โมเดลได้ถูกนิยามเรื่องความเชื่อมั่นจาก $\forall i_r, i_e, i_f \in I, IT$ สามารถเขียนความสัมพันธ์ได้เป็น $i_r \lesssim i_r$ - ไม่สามารถถ่ายทอดได้ $i_r \lesssim i_e \wedge i_r \lesssim i_f \nRightarrow i_r \lesssim i_f$ - ไม่มีกฎของการสมมาตร $i_r \lesssim i_e \nRightarrow i_e \lesssim i_r$ - ไม่มีถ่ายทอดการสมมาตร $i_r \lesssim i_e \wedge i_e \lesssim i_r \nRightarrow i_r = i_e$

นิยามที่ 1 : เมื่อ A และ B แทน 2 issuer โดยมีกฎของการเชื่อมั่น ดังนี้ $A \lesssim \mathrm{B}$ ซึ่งหมายถึง B เชื่อมั่นใน A กฎของการแตกลำดับของบทบาท และบทบาทของ B ที่ซึ่ง B เป็นไปตามการ (Tang et al., 2013, pp. 135) ให้ดังนี้

1) ให้การอนุญาตของ B กับบทบาทของ A และ
2) ให้บทบาทของ B เป็นบทบาทย่อยของ A

ส่วนที่สอง การอธิบายโมเดล MTAS ในรูปแบบทางการ

นิยามที่ 2 : โมเดลการตรวจสอบสิทธิ์การเข้าใช้ มีองค์ประกอบดังนี้ (Tang et al., 2013, pp. 136)

ตารางที่ 2. 4

ตารางแสดงความสัมพันธ์ต่าง ๆ ระหว่างองค์ประกอบของ MTAS

คำนิยาม	คำอธิบาย
$UO \subseteq U \times I$	เป็นความสัมพันธ์ที่เกิดขึ้นจากหลายความสัมพันธ์ของผู้ใช้งาน ไปยังความสัมพันธ์เดียวของ issuer เรียกว่า เซตของเจ้าของการเช่า
$RO \subseteq R \times I$	เป็นความสัมพันธ์ที่เกิดขึ้นจากหลายความสัมพันธ์ของบทบาท ไปยังความสัมพันธ์ของ issuer เรียกว่า เซตของเจ้าของของบทบาท แต่ละบทบาทจะต้องเป็นไปตามฟังก์ชันที่ตรวจสอบบทบาทว่าอยู่ในแต่ละ issuer หรือไม่ $roleOwner(r: R) \rightarrow I$ หรืออาจจะเขียนได้ว่า $roleOwner(r) \in \{i \in I \mid (r, i) \in RO\}$
$PO \subseteq P \times I$	เป็นความสัมพันธ์ที่เกิดขึ้นจากหลายความสัมพันธ์ของการอนุญาตไปยังความสัมพันธ์ของ issuer เรียกว่า เซตของเจ้าของของการอนุญาต แต่ละการอนุญาตเป็นไปตามฟังก์ชันที่ตรวจสอบการอนุญาตว่าอยู่ในแต่ละ issuer หรือไม่ $permOwner(p: P) \rightarrow I$ หรืออาจจะเขียนได้ว่า $roleOwner(p) \in \{i \in I \mid (p, i) \in PO\}$
$IT \subseteq I \times I$	เป็นความสัมพันธ์ระหว่าง I ที่มีเรื่องของความเชื่อถือเป็นสำคัญ โดยสามารถเขียนได้จากเครื่องหมาย " $\lesssim$ "
$canUse(r: R) \rightarrow 2^i$	เป็นฟังก์ชันที่ตรวจสอบเซตของบทบาท ว่ามีความสัมพันธ์กับ issuer หรือไม่ เขียนได้เป็น $canUse(r) = \{i \in I \mid roleOwner\ (r) \lesssim i\}$

คำนิยาม	คำอธิบาย
$UA \subseteq U \times R$	เป็นความสัมพันธ์ที่เกิดขึ้นจากหลายผู้ใช้งาน ไปยังหลายความสัมพันธ์ของบทบาท
$PA \subseteq P \times R$	เป็นความสัมพันธ์ที่เกิดขึ้นจากหลายการอนุญาต ไปยังหลายความสัมพันธ์ของบทบาท
$RH \subseteq R \times R$	เป็นความสัมพันธ์ เพื่อสร้างลำดับชั้นของบทบาท ซึ่งใช้แทนเครื่องหมาย $\geq$ และจำเป็นต้องเขียนได้เป็น $r \geq r_1$ โดยในความสัมพันธ์ของ RH จะต้องตรวจสอบว่า $roleOwner\ (r_1) \in canUse\ (r)$
$user(s{:}\ S) \rightarrow U$	เป็นฟังก์ชันที่ตรวจสอบระหว่างเซสชั่น ว่ามีผู้ใช้งานดังกล่าวอยู่ในเซตของผู้ใช้งานหรือไม่
$roles(s{:}\ S) \rightarrow 2^R$	ในเซตย่อยของบทบาท โดยจะต้องเป็นไปตาม $roles(s) \subseteq \{r \mid \exists r_2 \geq r[(user(s), r_2) \in UA \wedge userOwner(user(s)) \in canUse(r)]\}$

นิยามที่ 3 : การกำหนดการบริหารจัดการ MTAS หรือโมเดล AMTAS (Tang et al., 2013, pp. 136) เป็นฟังก์ชันที่กำหนดขึ้นเพื่อใช้ในการเพิ่มหรือลบองค์ประกอบของ user permission RH และ Trust โดยมีเงื่อนไขในการตรวจสอบตามคอลัมน์ Condition หากตรวจสอบแล้วเป็นจริง จึงดำเนินการเพิ่ม/ลบองค์ประกอบตามคอลัมน์ Update

- ทรัพยากรที่ผู้ร้องขอ A จะต้องตอบสนองในการจัดการความสัมพันธ์ของการเชื่อถือโดยการนิยามจาก $A \lesssim \mathrm{B}$
- ทรัพยากรของเจ้าของ B จะตอบสนองในการจัดการการให้กับบทบาทของผู้ร้องขอ A

Function	Condition	Update
$assignUser\ (i, r, u)$	$i = roleOwner(r) \wedge u \in U$	$UA' = UA \cup \{u \rightarrow r\}$
$revokeUser\ (i, r, u)$	$i = roleOwner(r) \wedge u \in U \wedge u \rightarrow r \in UA$	$UA' = UA \setminus \{u \rightarrow r\}$
$assignPerm\ (i, r, p)\ )$	$i = permOwner(p) \wedge i \in canUse\ (r)$	$PA' = PA\ \cup \{p \rightarrow r\}$
$revokePerm\ (i, r, p)$	$t = permOwner(p) \wedge i \in canUse\ (r) \wedge \mathrm{p} \rightarrow \mathrm{r} \subset \mathrm{PA}$	$PA' = PA \setminus \{p \rightarrow r\}$

Function	Condition	Update
$assignRH$ $(i, r1, r)$	$i = roleOwner(r) \wedge i \in canUse(r1) \wedge$ $\neg\,(r1 \gg r) \wedge \neg\,(r \geq r1)^{a}$	$\geq' = \geq \sqcup \{r2, r3{:}\, R \mid r2$ $\geq r1 \wedge r$ $\geq r3$ $\wedge roleOwner\,(r3)$ $\in canUse\,(r2) \cdot r2$ $\rightarrow r3)$
$revokeRH$ $(i, r1, r)$	$i = roleOwner(r) \wedge i$ $\in canUse(r1)\ \wedge\ (r1 \gg r^{b})$	$\geq' =\ (\geq \backslash \{r1 \rightarrow r)^{*c}$
$assignTrust$ (i, i_1)	$i_1 \in I$	$\leq' = \leq\ U$ $\{i \rightarrow i_1\}$
$revokeTrust$ (i, i_1)	$i_1 \in I \wedge i\ \leq i_1 \wedge i \neq i_1$	$\leq' = \leq \backslash$ $\{i \rightarrow i_1\}$

a. This condition avoids cycle creation in the role hierarchy.

b. It requires r_1 to be an immediate ascendant of r.

c. Implied relations are preserved after revocation.

d. By revoking the trust relation, the $canUse$ () function of I's roles automatically updates accordingly, same as ***PA*** and ***RH***.

ภาพที่ 2.8 แผนภาพอธิบายฟังก์ชันของการบริหารจัดการของ AMTAS
ที่มา: "Multi-Tenancy Authorization Models for Collaborative Cloud Services," by B. Tang, R. Sandhu, and Q. Li, pp. 229-238. Copyright 2013 by IEEE.

ส่วนที่สาม การอธิบายรูปแบบของความเชื่อถือที่เพิ่มเติมขึ้น (Tang et al., 2013, pp. 136-137)

นิยามที่ 4 : การเชื่อถือโดยใช้บทบาทสาธารณะกลาง (TCPR) โดยเปลี่ยนได้ดังนี้

- $P_T(i{:}\, I) \rightarrow 2^R$ เป็นฟังก์ชันที่ใช้ในการตรวจสอบ issuer กับเซตของบทบาทสาธารณะกลาง โดย i เป็นบทบาทที่แตกออกไปให้กับผู้ถูกเชื่อถือ และ
- $canUse(r{:}\, R) \rightarrow 2^T$ เปลี่ยนเป็น $CanUse(r) = \{i\} \cup \{i_i \in i \mid i \lesssim i_1 \wedge r \in P_T(i)\}\,;\ i = roleOwer\ (r)$

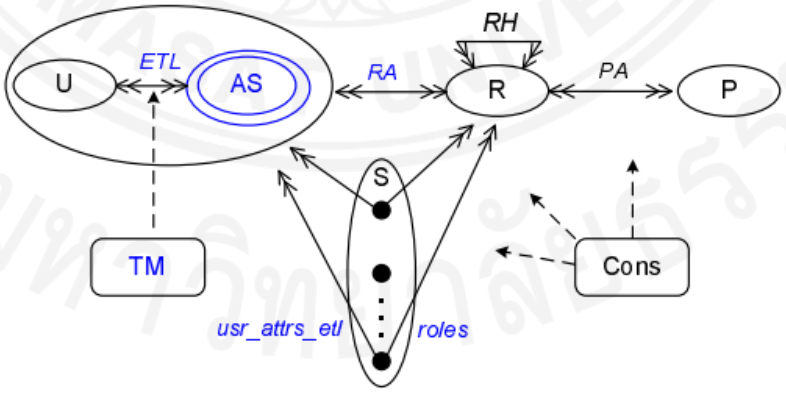
นิยามที่ 5 : การเชื่อถือโดยใช้ความสัมพันธ์ของบทบาทสาธารณะกลาง (PCPR) โดยเปลี่ยนได้ดังนี้

- $P_R(i: IT) \rightarrow 2^R$ เป็นฟังก์ชันที่ใช้ในการตรวจสอบความเชื่อถือระหว่าง issuer กับความเชื่อของเซตของบทบาทสาธารณะกลาง
- $canUse(r: R) \rightarrow 2^T$ เปลี่ยนเป็น $CanUse(r) = \{i\} \cup \{i_1 \in I | i \lesssim i_1 \wedge r \in P_R(i \lesssim i_1)\}$; $i = roleOwer\ (r)$

2.2.1.3 งานวิจัยอื่น ๆ ที่นำโมเดล RBAC มาต่อยอด

จากโมเดล RBAC นั้นก็ได้มีการนำพื้นฐานมาใช้ในการต่อยอด โดยผู้วิจัยได้ศึกษางานที่ได้นำ RBAC มาต่อยอดเพิ่มเติม เพื่อให้เห็นภาพของการทำงานของ RBAC ที่มีลักษณะที่แตกต่างกัน สรุปได้ดังนี้

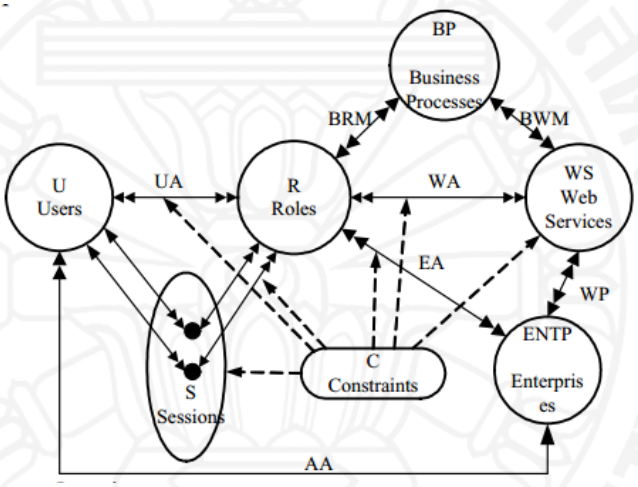
RAMARS_RM (Jin, Ahn, Shehab, and Hu, 2007) เป็นโมเดลที่นำพื้นฐานขององค์ประกอบผู้ใช้งาน, บทบาท และการอนุญาตมาต่อยอด เพื่อเพิ่มเติมความสามารถในกำหนดความการเชื่อถือ (Trust) โดยเพิ่มเซตของ Attributes Set (AS) เพื่อเป็นความสัมพันธ์เพิ่มระหว่างผู้ใช้งานกับคุณลักษณะบางประการ เพื่อนำมาเป็นเซต ในการตรวจสอบหาความเชื่อในการถึงเข้าระบบ และจึงกำหนดนโยบายเพื่อตรวจสอบการเข้าถึง จากคุณลักษณะที่ผูกกับผู้ใช้งานดังกล่าวโดยใช้องค์ประกอบเป็น (Aname, Value) เช่น (เชื้อชาติ, ไทย) เป็นต้น และนำนโยบายที่เขียนโดยการอ้างกับ AS เพื่อให้เกิดการตรวจสอบสิทธิ์การเข้าถึงนั้นเอง



ภาพที่ 2.9 แผนผังแสดงองค์ประกอบ และความสัมพันธ์ของโมเดล RAMARS_RM

ที่มา: “Towards Trust-aware Access Management for Ad-hoc Collaborations,” by J. Jin, G–J. Ahn, M. Shehab, and H. Hu., pp.41-48. Copyright 2007 by IEEE.

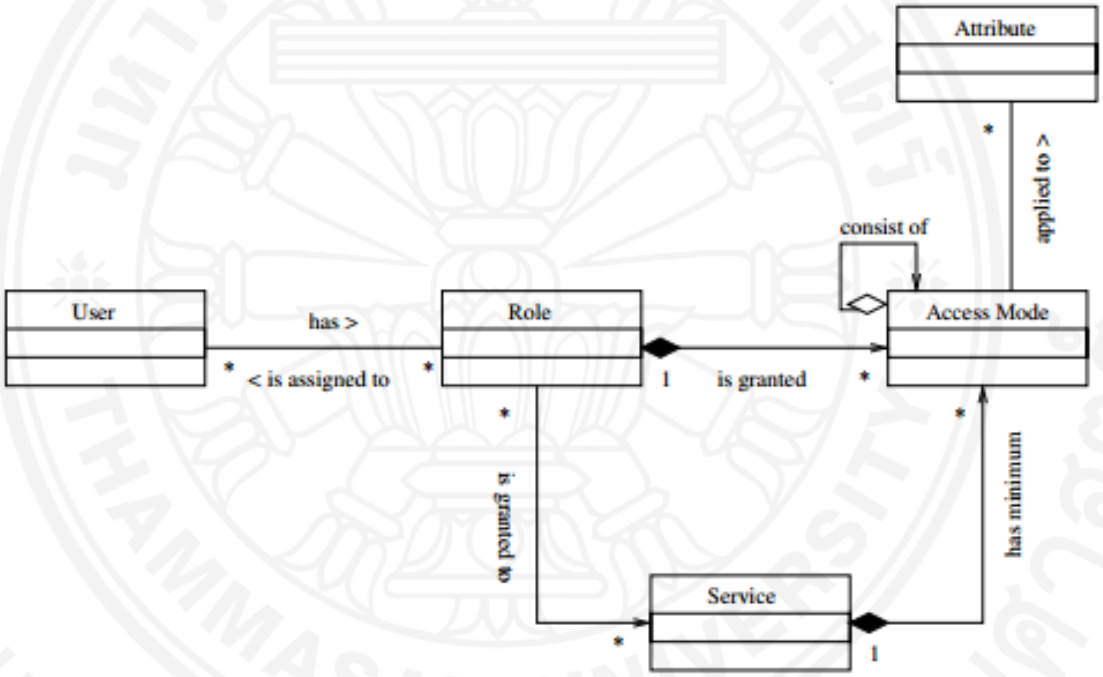
WS-RBAC (Liu and Chen, 2010) เป็นโมเดลที่นำ NIST RBAC มาพัฒนาต่อยอดให้มีความสามารถในการใช้งานบนสภาพแวดล้อมของเว็บบริการได้ โดยได้เพิ่มองค์ประกอบ 2 ส่วน คือ ส่วนของการกำหนดนโยบายข้อกำหนดทางธุรกิจ (Business Process) โดยใช้ BPEL4WS ในการอธิบาย ซึ่งเป็น XML ที่ใช้ในการกำหนดข้อบังคับทางธุรกิจ ซึ่งได้นำมาเชื่อมกับการกำหนดบทบาท เพื่อใช้ในการตรวจสอบความสอดคล้องตามข้อบังคับทางธุรกิจหรือไม่ และส่วนที่ 2 คือ Enterprise เป็นเซตที่กำหนดองค์กรที่ใช้งานเว็บบริการได้ และได้ดัดแปลงเซตของวัตถุ ให้เปลี่ยนเป็นเซตของเว็บบริการแทน (Web Service) เพื่อกำหนดว่ามีเว็บบริการที่สามารถใช้ให้กับบทบาทได้ ในโมเดลดังกล่าวได้ตัดการอธิบายในส่วนของการกำหนดลำดับชั้นของบทบาทออก ซึ่งองค์ประกอบและความสัมพันธ์เป็นไปตามภาพที่ 2.10



ภาพที่ 2.10 องค์ประกอบและความสัมพันธ์ของโมเดล WS-RBAC

ที่มา: “An Access Control Model for Web Services in Business Process,” by P. Liu, and Z. Chen, pp. 49-56. Copyright 2004 by IEEE.

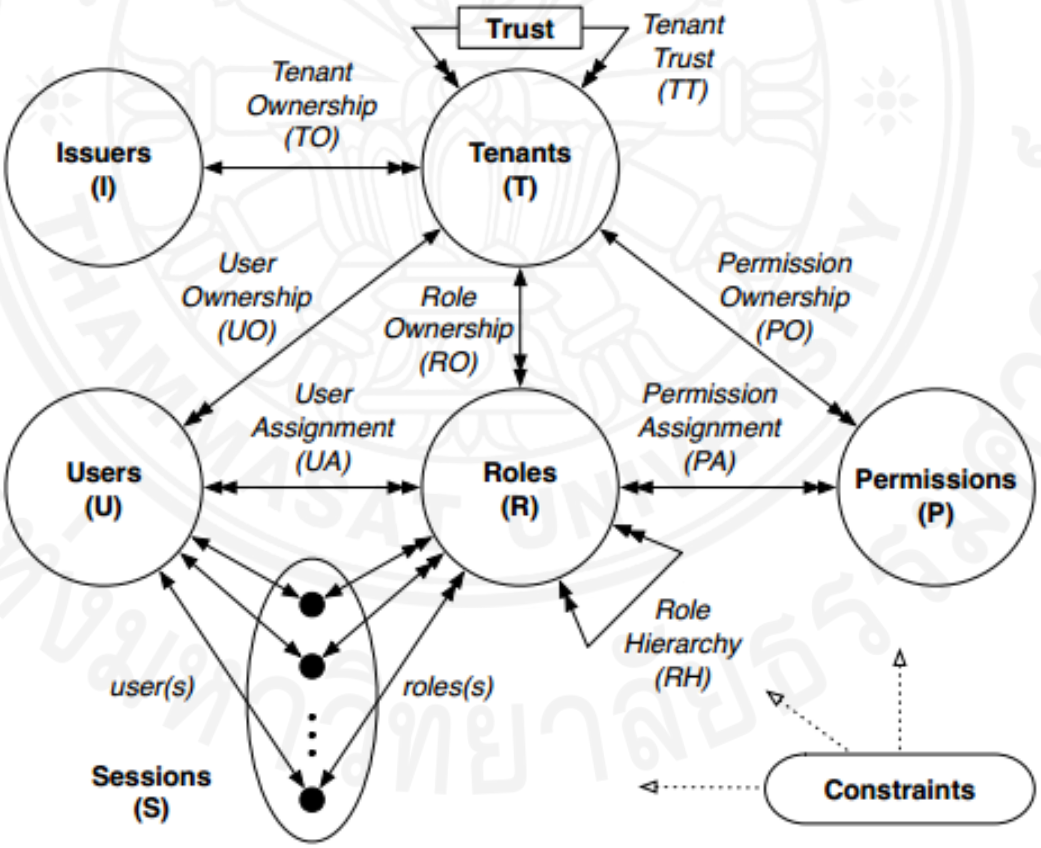
SWS-RBAC (Roosdiana Wonohoesodo, Zahir Tari, 2004) ได้ใช้พื้นฐานของโมเดล RBAC มาใช้ในสภาพแวดล้อมของเว็บบริการ (Web Service) เดียว ซึ่งได้เพิ่มปรับปรุงองค์ประกอบของวัตถุ ออกเป็น 3 องค์ประกอบ คือ Access Mode เพื่อเก็บเซตของวิธีการทำงาน เช่น อ่าน เขียน เป็นต้น เซตของคุณลักษณะเพื่อใช้แทนการเก็บวัตถุที่สามารถเข้าถึงได้ และ service เพื่อเก็บเซตของรายชื่อของเว็บบริการที่มีในระบบ ซึ่งทั้งสามนี้อาจจะเรียกได้ว่าเป็นองค์ประกอบของวัตถุ (object) ในโมเดลของ RBAC เพื่อใช้ในการอธิบายความสัมพันธ์ของการตรวจสอบสิทธิ์การเข้าถึงของเว็บเซอร์วิส โดยองค์ประกอบและความสัมพันธ์เป็นไปตามภาพที่ 2.11



ภาพที่ 2.11 องค์ประกอบและความสัมพันธ์ของโมเดล SWS-RBAC

ที่มา: “A Role Based Access Control for Web Services,” by R. Wonohoesodo, and Z. Tari, pp. 292-298. Copyright 2004 by IEEE.

MT-RBAC (Tang, Li, and Sandhu, 2013, pp. 229-238) ซึ่งเป็นงานวิจัยที่ต่อยอดจาก MTAS โดยเพิ่มการแยกองค์ประกอบของ Tenant เพื่อให้เกิดความชัดเจนในเรื่องของการกำหนดความสัมพันธ์ระหว่างการเช่ากับบทบาทมากยิ่งขึ้น แต่ด้วยความสัมพันธ์ของ UserOwnership (UO) ที่กำหนดความสัมพันธ์จากหลายความสัมพันธ์ไปยังความสัมพันธ์เดียวของผู้ใช้งานไปยังการเช่า ทำให้การกำหนดผู้ใช้งานในเจ้าของเดียวกัน ที่ใช้งานข้ามระหว่างการเช่าของตนเอง ทำให้ต้องเกิดผู้ใช้งานหลายบัญชี ซึ่งก่อให้เกิดความซ้ำซ้อนในการตั้งชื่อบัญชีภายใต้ในแต่ละการเช่า แต่ผู้วิจัยได้กำหนดความสัมพันธ์ของ UserOwnership (UO) ในงานวิจัยที่กำลังศึกษาที่แตกต่างจาก MT-RBAC ในการกำหนดจากผู้ใช้งานไปยัง issuer แทน ซึ่งทำให้เกิดความสัมพันธ์ของผู้ใช้งานกับ issuer ที่ใช้สื่อความหมายแทนเจ้าของของการใช้งาน ดังนั้นทำให้บัญชีผู้ใช้งานในเจ้าของเดียวกัน สามารถใช้งานในหลายการเช่าได้ โดยองค์ประกอบ และความสัมพันธ์ของโมเดล MT-RBAC เป็นไปตามภาพที่ 2.12



ภาพที่ 2.12 องค์ประกอบและความสัมพันธ์ของโมเดล MT-RBAC

ที่มา: “A Multi-Tenant RBAC Model for Collaborative Cloud Services,” by B. Tang, Q. Li, and R. Sandhu, pp. 132-138. Copyritght 2013 by IEEE.

2.2.2 ลักษณะที่เกี่ยวข้องกระบวนการห่วงโซ่การเรียกใช้บริการแบบประสานงาน

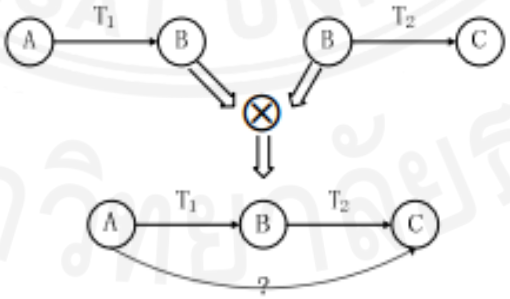
โดยส่วนแรกที่นำมาอธิบายคือ รูปแบบของการกำหนดความเชื่อถือ โดยงานพื้นฐานในเรื่องของความน่าเชื่อถือนั้น เป็นการอธิบายถึงความสัมพันธ์ระหว่างสองวัตถุ ซึ่งวัตถุแรกเป็นวัตถุที่ทำให้เกิดความเชื่อถือ (Trust) ซึ่งจะเรียกว่า ผู้ถูกเชื่อถือ (trustee) โดยต่อไปผู้วิจัยจะใช้คำว่า “ผู้ถูกเชื่อถือ” และวัตถุที่สองคือ วัตถุที่ถูกทำให้เกิดเชื่อถือ ซึ่งจะเรียกว่า ผู้เชื่อถือ (trustor) โดยต่อไปผู้วิจัยจะใช้คำแทนว่า “ผู้เชื่อถือ” โดยรูปแบบของการทำให้เกิดความเชื่อถือก็มีหลากหลายรูปแบบ อาทิเช่น เป็นเส้นทางเส้นเดียว หรือเป็นกราฟที่มีหลายเส้นทาง โดยในการที่จะอธิบายความเชื่อถือ ให้เข้าใจง่ายนั้น

โดยได้ถูกแบ่งออกเป็น 2 หัวข้อ คือ การอธิบายพีชคณิตในเรื่องของความเชื่อถือ และ โมเดลของความเชื่อถือ

ส่วนที่ 1 การอธิบายพีชคณิตในเรื่องของความเชื่อถือ

ซึ่งได้กล่าว รูปแบบการคิดสำหรับการคำนวณความเชื่อถือออกเป็น 2 รูปแบบ คือ

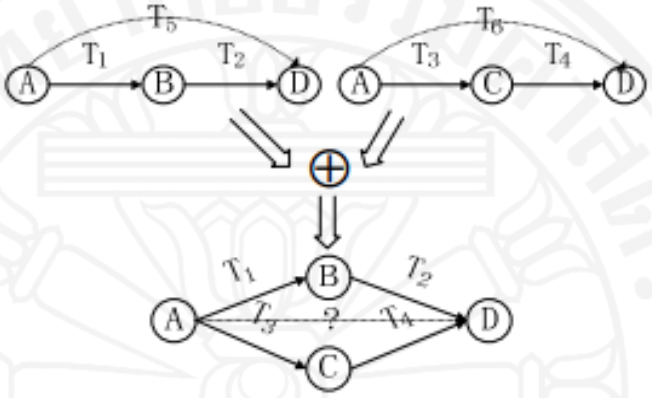
รูปแบบที่ 1) กราฟในการอธิบายความเชื่อถือนั้น เครื่องหมาย $\otimes$ ถูกแทนความหมายในเชิงของความเชื่อถือทางอ้อม ที่ใช้ในการหาค่าของความเชื่อถือเพื่อไปยังจุดของเป้าหมายในเส้นทางเดียวกัน ซึ่งสามารถอธิบายได้ตามภาพที่ 2.13 โดยจะเห็นได้ว่า T1 เป็นความเชื่อถือโดยตรงของ A ที่ให้กับ B และ T2 เป็นความเชื่อถือโดยตรงของ B ที่ให้กับ C เช่นกันเดียวกัน ดังนั้นจะเกิดคำถามในเรื่องความเชื่อถือจาก A ไปยัง C นั้นมีความเชื่อถืออย่างไร จึงนิยามในสมการได้เป็น $T_1 \otimes T_2$



ภาพที่ 2.13 อธิบายความหมายในการใช้เครื่องหมาย$\otimes$

ที่มา: “A General Trust Model Based on Trust Algebra,” by Y. Wenzhong, H.Chuanhe, W.Bo, And Z.Zhenyu, pp.125-129. Copyright 2009 by IEEE.

รูปแบบที่ 2) เครื่องหมาย $\oplus$ เป็นเครื่องหมายที่ใช้แทนการอนุมานความเชื่อถือทางอ้อม แต่ในการคำนวณค่าเชื่อถือในทางอ้อม ภายใต้เส้นทางที่หลากหลายในกราฟที่แสดงความเชื่อถือจากต้นทางไปยังปลายทางนั้น ยกตัวอย่างได้ตามภาพที่ 2.14 กล่าวได้คือ จะมีอยู่ 2 เส้นทางคือที่แสดงความเชื่อถือจาก A เชื่อถือไปยัง D ซึ่งค่าของความเชื่อถือ จะต้องพิจารณาทั้ง 2 กรณี คือ T5 และ T6 โดยทั้งสองกรณีดังกล่าวจะเห็นได้ว่าเป็นความเชื่อถือที่อยู่ในลักษณะทางอ้อมทั้งคู่ ซึ่งสามารถแทนสมการได้เป็น $T_5 \oplus T_6$ โดยที่ $T_5 = T_1 \otimes T_2, T_6 = T_3 \otimes T_4$



ภาพที่ 2.14 อธิบายความหมายในการใช้เครื่องหมาย $\oplus$

ที่มา: “A General Trust Model Based on Trust Algebra,” by Y. Wenzhong, H.Chuanhe, W.Bo, And Z.Zhenyu, pp.125-129. Copyright 2009 by IEEE.

ส่วนที่ 2 การอธิบายโมเดลความเชื่อถือ

นิยามที่ 2 : เป็นการอธิบายเอนโทรปี ใช้ในการคำนวณค่าความเชื่อถือ ได้เป็น

$$T(< trustor, trustee, action >) = \begin{Bmatrix} 0.5H(p), 0 \le p < 0.5 \\ 0.5(1 - H(p)), 0.5 \le p \le 1 \end{Bmatrix}$$

เมื่อ $H(p) = -p\log(p) - (1-p)\log(1-p)$ และ $p = P\ (< trustor, trustee, action >)$ โดยค่าของความเชื่อถือจะอยู่ระหว่าง [0,1] และเมื่อค่าของ p=0 แสดงได้ว่าไม่มีความเชื่อถือของผู้เชื่อถือไปยังผู้ถูกเชื่อถือ และเมื่อค่า p=1 แสดงได้ว่ามีความเชื่อถือขอผู้เชื่อถือไปยังผู้ถูกเชื่อถือ

เมื่อค่าความเชื่อถือ จะถูกลดลงเมื่อมีความสัมพันธ์ระหว่างความเชื่อถือถูกประมาณการณ์ในเส้นทางเดียว โดยใช้ $\otimes$ สามารถนิยามได้เป็น

$$\forall a, b \in S, a \otimes b = a \times b$$

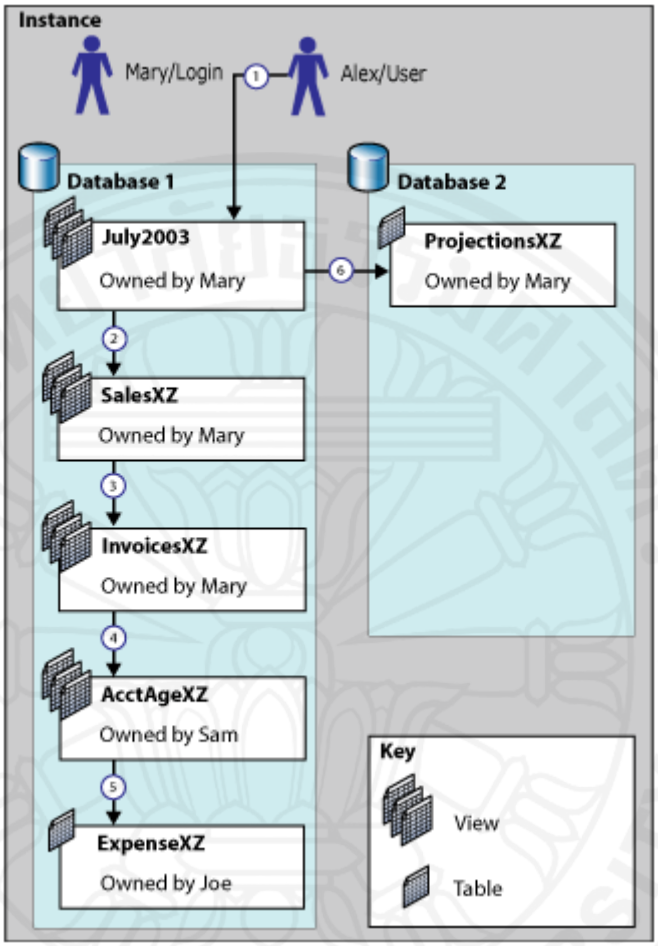
และในเส้นทางที่มีหลากหลาย โดยใช้ $\oplus$ สามารถนิยามได้เป็น

$$\forall a, b \in S, a \oplus b = \max(a, b)$$

ซึ่งจากนิยามดังกล่าว สามารถสรุปได้ว่า เมื่อเข้ามาคำนวณหาค่า T ซึ่งค่า T คือ ค่าของความเชื่อถือระหว่างความสัมพันธ์ของผู้ถือเชื่อ และผู้ถูกเชื่อ ในลักษณะทั้งการสืบทอดการเชื่อถือ และลักษณะของการหาค่าของการสืบทอดว่าเส้นทางใดมีค่าของความเชื่อถือมากกว่ากัน

โดยในงานวิจัย (Wenzhong, Chanhe, Bo, and Zhenyu, 2009, pp.125-129) ผู้วิจัยได้นำรูปแบบของความเชื่อถือดังกล่าวมาใช้ในการอธิบายการเกิดของห่วงโซ่การเรียกใช้บริการแบบประสานงาน ซึ่งจากภาพที่ 2.13 ถือได้ว่าเป็นห่วงโซ่แบบเส้นทางเดียว และในภาพที่ 2.14 แสดงให้เห็นถึงห่วงโซ่แบบหลายเส้นทาง จึงจะเห็นได้ว่ากระบวนการนำมาวิเคราะห์จะมีความแตกต่างกัน แต่ในด้วยลักษณะของ *SaaS* ที่ผู้วิจัยกำลังศึกษา จะเป็นการเกิดในรูปแบบแรก คือ เส้นทางเดียว เนื่องจากการเรียกบริการต่อไปยังผู้ให้บริการหนึ่ง หากเกิดในลักษณะของหลายเส้นทาง กระบวนการตรวจสอบจะมีความซับซ้อน และการส่งผลลัพธ์กลับมายังผู้ให้บริการที่เรียกบริการนั้นเป็นไปได้ยาก ซึ่งหากเกิดเหตุการณ์เช่นนั้น กระบวนการแก้ไขคือ เรียกไปทีละบริการ และเมื่อส่งกลับมา ก็เรียกต่อไปอีกบริการหนึ่งก็ถือได้ว่าเป็นการแก้ไข เพื่อไม่ให้เกิดเส้นทางที่เป็นในลักษณะของหลายเส้นทาง หรือแบบกราฟนั้นเอง

และองค์ประกอบอีกประการหนึ่ง ที่ผู้วิจัยได้นำส่วนของการตรวจสอบการเข้าถึงมาใช้ในกระบวนการของห่วงโซ่การเรียกใช้บริการแบบประสานงาน เพื่อลดขั้นตอนในการตรวจสอบ โดยใช้ปัญหาของการกำหนดสิทธิ์การเข้าถึง view และ table ของ MsSQL 2008R2 ในลักษณะของ Ownership Chain ที่เป็นปัญหาในการกล่าวถึงการเป็นเจ้าของทรัพยากรหนึ่ง ๆ แล้ว เมื่อให้สิทธิ์กับผู้ใช้งานอื่นในการเข้าถึงทรัพยากรที่เป็น view หรือ table หรือ function ใด ๆ ที่ถูกอนุญาตให้เข้าถึงแล้ว แต่ทรัพยากรที่ดังกล่าวยังต้องใช้ข้อมูลของ view หรือ table อื่น ๆ อีก ดังนั้นในส่วนของสิทธิ์จะมีกระบวนการอย่างไรในการตรวจสอบ ซึ่งใน Ownership Chain ได้กล่าวไว้ว่า “หากเป็นเจ้าของเดียวกันกับที่ให้สิทธิ์ไปแล้ว แล้วทรัพยากรที่ถูกเรียกใช้ต่อจากต้นทาง ให้เสมือนได้สิทธิ์เมื่อวัตถุนั้นเป็นเจ้าของเดียวกัน แต่หากเป็นเจ้าของอื่น ก็จะต้องตรวจสอบต่อไปว่าวัตถุที่กำลังเข้าถึงนั้นเจ้าของวัตถุมีการกำหนดการอนุญาตให้กับผู้ใช้งานต่อไปหรือไม่ ซึ่งหากให้สิทธิ์ ถือว่าได้ว่ามีสิทธิ์ แต่หากไม่มีการกำหนดสิทธิ์ ถือว่าได้ไม่มีสิทธิ์เข้าถึงวัตถุนั้น และมีผลต่อไปยังวัตถุต้นทางถือว่าไม่มีสิทธิ์เข้าถึงด้วย” ซึ่งตัวอย่างที่ (Ownership Chain Database Microsoft) ได้กล่าวถึงไว้ เป็นไปตามภาพที่ 2.15



ภาพที่ 2.15 ตัวอย่างของการกำหนดในเหตุการณ์ของ Ownership Chain
ที่มา: https://technet.microsoft.com/en-us/library/ms188676%28v=sql.105%29.aspx

โดยจากภาพที่ 2.15 เป็นการอธิบายลักษณะของ Ownership Chain โดยกำหนดตัวอย่างคือ view ที่ชื่อว่า July2003 โดยมี Mary เป็นเจ้าของ และต้องการให้ผู้ใช้งาน Alex เข้าใช้ view July2003 จึงกำหนดสิทธิ์ให้ Alex สามารถเข้าถึง July2003 ได้ แต่ด้วยลักษณะของ view นี้มีการใช้ข้อมูลของ view และ table อื่น ๆ เพื่อให้ได้ข้อมูล โดยมีการใช้ view และ table ของผู้ใช้งานอีก 2 คน คือ Sam และ Joe ซึ่งกระบวนการในการตรวจสอบสิทธิ์ตามตัวอย่างดังกล่าว เมื่อ Mary ให้สิทธิ์กับ Alex เข้าใช้ July2003 ซึ่ง July2003 เรียกต่อไปยัง view SalesXZ โดยเจ้าของ view SalesXZ คือ Mary ดังนั้นจึงถือได้ว่า Alex สามารถเข้าถึง view SalesXZ และ InvoicesXZ ได้ เนื่องจาก view ทั้งสามมีเจ้าของคนเดียวกันคือ Mary และ Mary ให้สิทธิ์ของ alex ในการเข้าถึง July2003 แล้ว แต่เมื่อ view InvoicesXZ เรียกต่อไปยัง AcctAgeXZ แต่เจ้าของของ view ดังกล่าว คือ Sam ดังนั้นจะต้องตรวจสอบสิทธิ์ว่า Sam มีการกำหนดสิทธิ์เข้าถึง view AcctAgeXZ ให้กับ Alex หรือไม่ หากมีการกำหนดให้ Alex เข้าใช้ได้ จึงถือว่า Alex มีสิทธิ์ในการเข้าถึงต่อไป แต่หากไม่กำหนด ถือว่า Alex ไม่สามารถใช้ view July2003 ในการเข้าถึงทรัพยากรทั้งหมดได้

และหาก Sam ได้กำหนดสิทธิ์ให้กับ Alex เข้าถึง view AccAgeXZ แล้วก็จะเรียกต่อไปยัง table ExpenseXZ ซึ่งเจ้าของของ table ดังกล่าวคือ Joe ซึ่งกระบวนตรวจสอบจะตรวจสอบคล้ายกับในกรณีของ view AccAgeXZ ของเจ้าของ Sam ซึ่งหาก Joe กำหนดให้ Alex เข้าถึง table AccAgeXZ จะถือได้ว่า Alex จะใช้ view July2003 ของ Mary ได้อย่างสมบูรณ์ต่อไป

โดยในตัวอย่างที่ผู้วิจัยยกมาให้เห็นนี้ ซึ่งได้นำกระบวนการตรวจสอบดังกล่าว มาใช้ในห่วงโซ่ของการเรียกแบบประสานงานต่อไป เพื่อลดขั้นตอนในการกำหนดสิทธิ์ในการเข้าถึงในกรณีที่เป็นเจ้าของเดียวกันด้วยเช่นกัน เพื่อไม่ให้เจ้าของทรัพยากรที่ต้องการให้ผู้อื่นมาใช้นั้น ไม่ต้องกำหนดการเข้าถึงทั้งหมดของทรัพยากรที่ถูกนำมากล่าวอ้างในการเข้าถึงนั้นเอง

2.2.3 การตรวจสอบกราฟที่มีลักษณะของการเกิด cycle

ในงานวิจัยชิ้นนี้ ได้เพิ่มเติมในการหาห่วงโซ่การเรียกบริการแบบประสานงาน เมื่อนำมาใช้ในการบริการบนคลาวน์ ซึ่งมีการร้องขอใช้บริการอยู่ตลอดเวลา อาจจะก่อให้เกิดการเรียกที่ไม่มีที่สิ้นสุดสำหรับการบริการที่เกิด cyclic ซึ่งจะทำให้ตัวบริการอาจจะเกิดการจองการใช้ทรัพยากรตลอดเวลา จนผู้ใช้บริการอื่น ๆ ไม่สามารถขอใช้การบริการตรวจสอบสิทธิ์ได้ ผู้วิจัยจึงนำการตรวจสอบกราฟที่เกิดเป็นวงจร (Cyclic Graph) มาใช้ เพื่อตรวจสอบว่ามีการเรียกบริการที่ก่อให้เกิดในลักษณะของ cycle หรือไม่ ซึ่งหากมีลักษณะของการเรียกบริการแบบ cycle ซึ่งเป็นไปได้ว่าผู้ใช้บริการเรียกบริการที่ก่อให้เกิดลักษณะการจองทรัพยากรอยู่ตลอดเวลา ระบบจึงส่งผลลัพธ์กลับไปให้ผู้ขอร้องว่ามีการ Error เกิดขึ้น และให้ขอใช้บริการใหม่ โดยผู้วิจัยได้เห็นกระบวนการตรวจสอบการเกิด cycle ของ Tiernan (Tiernan, 1970, pp. 722-726) ซึ่งใช้กระบวนการในการสำรวจเส้นทางทุกเส้นทางของกราฟที่มีการวิ่งไปได้ ซึ่งหากไม่มีเส้นทางเดินได้แล้ว จึงย้อนกลับไปจุดก่อนหน้า และทำตามกระบวนการเดิม ซึ่งสามารถนิยามได้ดังนี้ จุดของกราฟจะเป็นตัวเลข ตั้งแต่ 1 จนถึง V โดยขั้นตอนวิธีจะสร้างเส้นทางที่เป็นได้ไป $p = (v_1, v_2, \dots, v_k)$ โดย $v_1 < v_i$ สำหรับ $s \leq i \leq k$ ซึ่งจะถูกเริ่มต้นจาก จุด v_1 โดยเลือกเส้นทางที่สามารถจะผ่านได้ไปจากเส้นทาง $v_2 > v_1$ และเดินตามเส้นทางไปจนกระทั่งไม่พบจุดที่จะสามารถเดินต่อไปได้ จึงใช้การย้อนกลับไปยังจุดก่อนหน้าและเลือกเส้นทางใหม่ในการเดินต่อไป ซึ่งถ้า v_1 เป็นจุดที่ต่อจาก v_k จะถือได้ว่าเป็นเส้นทางในการเกิดการวนกลับ $(v_1, v_2, \dots, v_k, v_1)$ ซึ่งเมื่อดูเวลาที่ใช้ในกรณีที่แย่ที่สุด คือเป็นลักษณะของ Exponential ของจำนวนที่เกิดการวนกลับ

Tarjan (Tarjan, 1973, pp. 146-160) ได้ปรับปรุงวิธีขั้นตอนของ Tiernan โดยใช้วิธีท่องกราฟในแนวลึก (Depth first Search) และเพิ่มสแตก 2 ตัว ในการเก็บเส้นทางที่เคยผ่านซึ่งเรียกว่า “mark stack” และเส้นทางปัจจุบันเรียกว่า “point stack” ซึ่งจะทำให้ไม่ต้องไล่เส้นทางที่ไม่จำเป็นเหมือนกับแนวคิด Tiernan ที่ไล่เส้นทางที่เป็นได้ ทำให้ระยะเวลาที่ใช้ของ Tarjan เป็น O ((V+E) (C+1)) โดยที่ V เป็นจำนวนจุดทั้งหมด E เป็นจำนวนเส้นทาง และ C คือจำนวนเส้นทางของวงกลับที่เกิดขึ้น

อย่างไรก็ตาม แนวคิดของ Tarjan ก็ยังพบปัญหาในบางกรณี ที่ทำให้ระยะเวลาที่ใช้เป็นแบบเชิงเส้น (Linear Algorithm) ตามภาพที่ 2.17 ซึ่งจะพบว่า เป็นเส้นทางที่ไม่เจอการวนกลับ แต่จะทำให้การเข้าถึงเส้นทางที่สามารถเดินทางได้เป็นเส้นทางแรก (1,2,..., n) เส้นทางที่สอง (2,3,...,n) ซึ่งเป็นเส้นทางที่ซ้ำกัน และเช่นเดียวกันกับ ภาพที่ 2.17 ซึ่งจะพบว่าเส้นทางหลังจาก 3,... , n ก็จะถูกเดินแล้ว และยังไม่มีเส้นทางที่วนกลับ ซึ่งทำให้เกิดเส้นทางที่ไม่เป็นจำเป็น ดังนั้น Szwarcfiter (Szwarcfiter, 1974) จึงได้แก้ปัญหาดังกล่าวเพิ่มเติมต่อจาก Tarjan ซึ่งเพิ่มเงื่อนไขเพื่อตรวจสอบเส้นทางที่เคยผ่านมาแล้ว และไม่พบการวนกลับ เส้นทางส่วนนั้นจะไม่ต้องนำมาตรวจสอบอีก ทำให้ระยะเวลาสำหรับวิธีขั้นตอนดังกล่าวเป็น O (V+E) ในแต่ละรอบของการพบการวนกลับ โดยที่ N เป็นจำนวนของจุด และ M เป็นจำนวนของเส้นทาง

ภาพที่ 2.16 แสดงตัวอย่างกราฟที่ไม่มีการวนกลับ ซึ่งแสดงให้เห็น worst case ของการใช้ระยะเวลาของ Tarjan

ที่มา: “Finding the elementary cycles of a directed graph,” by Szwarcfiter, Jayme Luiz, Technicla Report Series no#60. Copyright 1974 by Newcastle upon Tyne, England.

ภาพที่ 2.17 แสดงตัวอย่างกราฟที่มีการวนกลับ และมีเส้นทางที่ซับซ้อน ซึ่งแสดงให้เห็น worst case ของการใช้ระยะเวลาของ Tarjan

ที่มา: “Finding the elementary cycles of a directed graph,” by Szwarcfiter, Jayme Luiz, Technicla Report Series no#60. Copyright 1974 by Newcastle upon Tyne, England.

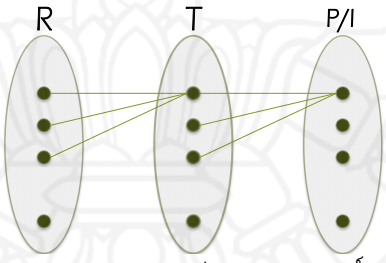
จากทั้ง 3 แนวความคิดในการหากราฟที่มีการวนกลับ ผู้วิจัยได้นำแนวคิดของ Szwarcfiter มาใช้ในการหา เนื่องจาก Tiernan นั้นระยะเวลาที่เป็นลักษณะของ exponential และสำหรับแนวคิดของ Tarjan ก็ยังพบหลายสถานการณ์ที่ทำให้ระยะเวลาที่ใช้เป็นแบบเชิงเส้น (Linear Algorithm) โดย Szwarcfiter ได้แก้ไขรูปแบบที่มีลักษณะระยะเวลาที่เป็นเชิงเส้น โดยลักษณะของห่วงโซ่การบริการแบบประสานงานส่วนใหญ่จะมีลักษณะตามภาพที่ 2.16 ในข้อมูลที่ยังไม่เกิดการวนกลับ และเมื่อมีลักษณะของการวนกลับระยะเวลาที่ใช้ ก็ยังเป็น O (V+E) ซึ่งจะสามารถทำให้ประสิทธิภาพในการตรวจสอบดีขึ้นอีกด้วย

บทที่ 3
วิธีการวิจัย

3.1 การออกแบบการวิจัย

ในกระบวนการออกแบบงานวิจัยชิ้นนี้ จากปัญหาที่ได้กล่าวถึงในบทที่ 1 สามารถสรุปได้เป็น 2 ประเด็น คือ

(1) การกำหนดบทบาท ผู้ที่สามารถกำหนดแต่ละบทบาทขึ้นมา ก็คือ เจ้าของซอฟต์แวร์แต่ละซอฟต์แวร์ แต่ในโมเดล MTAS นั้นมีความสัมพันธ์ที่ทำให้เกิดการกำหนดบทบาท 1 บทบาทสามารถที่จะเข้าถึงได้หลายซอฟต์แวร์ ทำให้เกิดความขัดแย้งกัน ดังนั้นเพื่อป้องกันไม่ให้เกิดลักษณะของการกำหนดบทบาทเข้าถึงได้มากกว่า 1 การใช้ซอฟต์แวร์ ผู้วิจัยจึงได้แยกการเช่าออกมาเป็นอีกองค์ประกอบหนึ่ง และกำหนดความสัมพันธ์จากบทบาทไปยังการเช่าในลักษณะของ many-1 และจากการเช่าไปยัง issuer เป็นลักษณะของ many-1 ดังภาพที่ 3.1 ซึ่งจะช่วยป้องกันไม่ให้บทบาทหนึ่ง มีการกำหนดการเข้าถึงการเช่าซอฟต์แวร์ได้มากกว่าหนึ่งซอฟต์แวร์ได้ทันที



ภาพที่ 3.1 ความสัมพันธ์ระหว่างบทบาท (R) ประเด็น (I) และสิทธิ์/การเช่าใช้ (P/T) ในโมเดล C-MTAS เพื่อแก้ไขบทบาท 1 บทบาทเข้าถึงได้ 1 การเช่า

(2) ปัญหาในด้านของการเรียกใช้บริการในรูปแบบของห่วงโซ่การเรียกใช้บริการแบบประสานงาน ซึ่งเมื่อการบริการมีจำนวนมากขึ้นเรื่อย ๆ จึงก่อให้เกิดการเรียกใช้บริการระหว่างกันเองในลักษณะของห่วงโซ่มากยิ่งขึ้น ซึ่งทำให้การกำหนดนโยบายเกิดการแตกไฟล์สำหรับการกำหนดสิทธิ์การเข้าถึงที่เพิ่มขึ้นไปเรื่อย ๆ ดังนั้นหากมีการปรับรูปแบบการอ่านไฟล์สำหรับสิทธิ์ในห่วงโซ่การบริการแบบประสานงานลง โดยแยกการกำหนดสิทธิ์ดังกล่าวออกไปเป็นอีกหนึ่งไฟล์ ที่ไม่ได้รวมอยู่ในนโยบายการเข้าถึงแบบปกติ ก็จะช่วยให้เข้าใจและจัดการนโยบายดังกล่าวง่ายขึ้น รวมทั้งการใช้ IO ของการอ่านไฟล์น้อยลงด้วยเช่นกัน ดังนั้นผู้วิจัยจึงได้นำการกำหนดสิทธิ์ให้ห่วงโซ่การบริการแบบประสานแยกออกเป็นนโยบายต่างหาก เพื่อช่วยใน 2 ปัจจัยดังกล่าว

ในเรื่องของโมเดลพื้นฐานนี้ผู้วิจัยได้สนใจศึกษาและปรับปรุง กีคือโมเดล MTAS ซึ่งมีลักษณะที่รองรับการตรวจสอบสิทธิ์ในสภาพแวดล้อมของผู้เช่าหลายราย และปัญหาในข้อ 1) ที่ได้กล่าวในข้างต้น ผู้วิจัยจึงดำเนินการปรับปรุงเพื่อไม่ให้เกิดความสัมพันธ์ของการกำหนดบทบาท 1 บทบาทเข้าถึงซอฟต์แวร์ในการเช่าได้หลายซอฟต์แวร์พร้อมกัน โดยเพิ่มเติมในการแยกองค์ประกอบการเช่า (Tenant) ออกจากการกำหนดภายใต้ของการอนุญาตอย่างชัดเจน และกำหนด

ความสัมพันธ์จาก Tenant ไปยัง Permission เป็นลักษณะของ 1-many เพื่อไม่ให้เกิดการเข้าถึงในบทบาทหนึ่ง ๆ ได้สำหรับการเช่าในหลายการเช่า

และในรูปแบบของการเรียกบริการแบบห่วงโซ่การเรียกแบบประสานงานนั้น เป็นรูปแบบที่ผู้วิจัยได้สนใจอีกประการหนึ่ง โดยการแยกนโยบายออกมาจากการกำหนดนโยบายในรูปแบบปกติ เพื่อลดการกำหนดไฟล์ที่มีความยุ่งยาก อีกทั้งผู้วิจัยได้นำกระบวนการตรวจสอบการเกิด cycle ในกราฟที่เป็นเส้นตรงของ Szwarcfiter มาตรวจสอบการเกิดบริการที่มีลักษณะวนกลับไปยังบริการเดิมที่เคยเรียกแล้ว เพื่อป้องกันการเรียกการบริการที่อาจจะไม่รู้จบได้ เนื่องจากกระบวนการตรวจสอบสิทธิ์นี้ จำเป็นต้องทำงานในลักษณะของชั้นกลางที่ต้องรองรับการร้องขอบริการด้วยปริมาณมาก ๆ หากมีการครองทรัพยากรการเรียกเพื่อตรวจสอบสิทธิ์จนเกิดการ down time จนที่ผู้ร้องขออื่นๆ ไม่สามารถใช้งานได้ เพื่อป้องกันไม่ให้มีการเรียกใช้บริการแบบ cycle ผู้วิจัยจึงตรวจสอบหากมีการเรียกบริการที่เกิด cyclic จึงตัดการขอร้องขอ เพื่อรองรับให้ผู้อื่นสามารถร้องขอได้ โดยใช้กระบวนการในงานวิจัยชิ้นนี้ว่า “กระบวนการตรวจสอบการเข้าถึงในสภาพแวดล้อมของห่วงโซ่การเรียกบริการแบบประสานงานในความหลากหลายในการเช่า” หรือ Calling Chain Coordination in Multi-Tenancy Authorization System (C-MTAS) เพื่อรองรับการตรวจสอบการเข้าถึงในลักษณะของผู้เช่าหลายรายในสภาพแวดล้อมบนคลาวน์ รวมทั้งให้การรองรับการทำงานในลักษณะของห่วงโซ่การเรียกบริการแบบประสานงานที่จัดการง่ายขึ้น

3.2 ขั้นตอนการวิจัย

สำหรับขั้นตอนในการวิจัย สามารถสรุปขั้นตอนได้เป็น 6 ขั้นตอน ได้ดังนี้

3.2.1 ศึกษากระบวนการ และโมเดลการตรวจสอบการเข้าถึง โดยการใช้บทบาท (RBAC) มาเป็นตัวกำหนด ซึ่งผู้วิจัยได้นำโมเดลพื้นฐานที่ใช้ในงานวิจัย มาจากโมเดลของ AMTAS

3.2.2 ศึกษานิยามและความหมายของความเชื่อถือ และลักษณะการเกิดของการเรียกบริการในรูปแบบต่าง ๆ ที่มีลักษณะเป็นการเรียกแบบต่อเนื่องกัน เพื่อนำมาใช้ในการอธิบายในรูปแบบของห่วงโซ่การเรียกใช้บริการแบบประสานงาน

3.2.3 ปรับปรุงโมเดลสำหรับการตรวจสอบการเข้าใช้ จากโมเดลของ MTAS โดยการแยกการเช่าออกจากการกำหนดภายใต้ของการอนุญาต เพื่อลดช่องโหว่สำหรับข้อจำกัดในการแตกบทบาทให้รองรับสำหรับการเช่าหลาย ๆ การเช่าที่อยู่ในแต่ละ issuer และรวมทั้งอธิบายกระบวนการเพื่อให้โมเดลรองรับการเรียกใช้บริการในรูปแบบของห่วงโซ่การทำงานแบบประสานงาน

3.2.4. พัฒนาระบบตามโมเดลข้อ 3.2.3. โดยการ Implement โมเดลดังกล่าว ด้วยการใช้ XACML 3.0 รวมทั้งนำ Library ของ Java Sun XACML 1.2 มาใช้ในการพัฒนา

3.2.5. ทดสอบโมเดลตามข้อ 3.2.4 โดยตั้งกรณีทดสอบที่ได้เป็น 2 กรณี คือ กรณีปกติในการตรวจสอบสิทธิ และกรณีของการบริการที่เกิดห่วงโซ่การบริการแบบประสานงาน

3.2.6. วิเคราะห์ในเชิงการเปรียบเทียบ เรื่องของการปรับขนาดของฮาร์ดแวร์ในการรองรับการทำงานของ C-MTAS ว่าเมื่อมีการขยายขีดจำกัดของฮาร์ดแวร์จะทำให้โมเดล C-MTAS มีประสิทธิ์ภาพการทำงานที่มากขึ้นด้วยหรือไม่ เนื่องจากตัวโมเดลจะต้องเป็นตัวกลางที่รองรับการร้องขอของผู้ตรวจสอบสิทธิ์เป็นจำนวนมาก เมื่อถูกใช้งานบนคลาวน์ โดยใช้เครื่องจำลอง 4 โมเดลของฮาร์ดแวร์ มาวัดในเรื่องของเวลาการตอบสนอง ค่าภาระการรองรับการโหลด และกราฟฮิตโตแกรมของระยะเวลาการตอบสนองทั้งหมด จะมีผลอย่างไรบ้าง โดยฮาร์ดแวร์ทั้ง 4 ชุด เป็นไปตามนี้

(1) CPU ขนาด 1 core/ Ram ขนาด 1024 Mb

(2) CPU ขนาด 2 core/ Ram ขนาด 1024 Mb

(3) CPU ขนาด 2 core/ Ram ขนาด 2048 Mb

(4) CPU ขนาด 4 core/ Ram ขนาด 4096 Mb

3.3 ออกแบบโมเดลสำหรับการควบคุมการเข้าถึง เพื่อใช้ในตรวจสอบสิทธิ์การเข้าใช้ เพื่อรองรับการใช้บริการแบบห่วงโซ่การเรียกใช้บริการแบบประสานงานกัน

ในส่วนของการออกแบบโมเดลสำหรับการควบคุมการเข้าถึง โดยจากการนำพื้นฐานของ AMTAS มาใช้ในการออกแบบ ซึ่งจากจุดบกพร่องในโมเดลดังกล่าว ก็คือเรื่องของการอธิบายการเช่าให้อยู่ภายใต้ประเด็น (I) ทำให้เกิดปัญหาดังที่กล่าวไว้ใน 3.1 แล้ว ซึ่งในขั้นตอนของกระบวนการออกแบบโมเดลสำหรับการควบคุมการเข้าถึง ในการตรวจสอบสิทธิ์การเข้าใช้ สามารถอธิบายได้เป็น 3 ส่วน คือ

ส่วนที่หนึ่ง คือ การอธิบายองค์ประกอบของโมเดล C-MTAS

ส่วนที่สอง คือ การอธิบายความสัมพันธ์ระหว่างองค์ประกอบทั้ง 5 ในรูปแบบทางการตามโมเดลที่อธิบายไว้ในส่วนที่หนึ่ง

และส่วนที่สาม คือ การกำหนดลักษณะของการเรียกใช้บริการแบบประสานงาน

3.3.1. องค์ประกอบของโมเดล C-MTAS

โมเดลในการควบคุมการเข้าถึงการใช้บริการ โดยเป็นกลไกในการอธิบายสิทธิ์ในการเข้าใช้นั้น จะประกอบไปด้วยองค์ประกอบ 5 องค์ประกอบ อันได้แก่ ประเด็น (Issuers:I) การเช่าใช้ (Tenant:T) ผู้ใช้งาน (Users:U) บทบาท (Roles:R) การอนุญาต (Permission:P) เซสชั่น (Session) จาก 6 องค์ประกอบในข้างต้น สามารถเขียนความสัมพันธ์ดังในภาพที่ 3.1 และรายละเอียดตามตารางที่ 3.1

นิยามที่ 1 : องค์ประกอบของโมเดล C-MTAS

ตารางที่ 3.1

องค์ประกอบของโมเดล C-MTAS

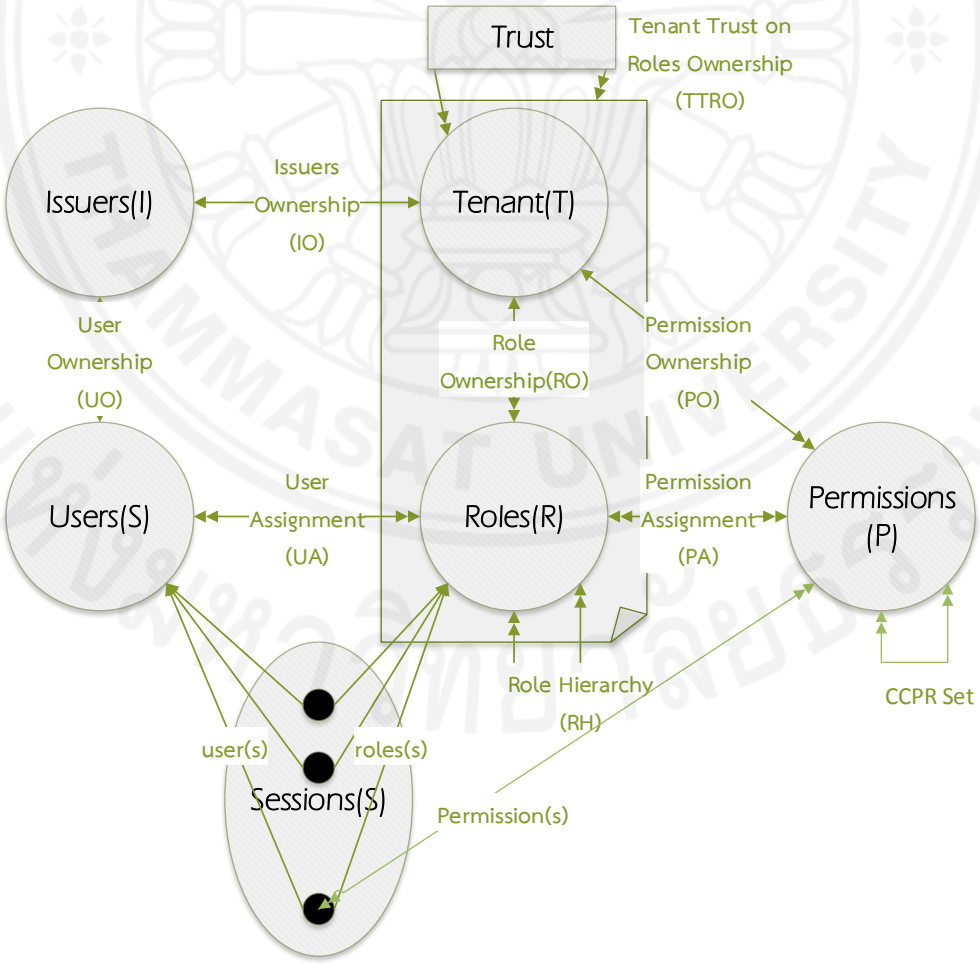
คำนิยาม	คำอธิบาย
Issuer	**ประเด็น** จะแทนองค์กรหรือบุคคลที่ใช้บริการบนคลาวน์ ซึ่งการแทน issuer อย่างเดียวจะยังไม่สามารถอธิบายองค์ประกอบที่ใช้บริการบนคลาวน์ได้อย่างครบถ้วน ดังนั้นจะต้องแยกการเช่า (T) ออกมาเป็นอีกองค์ประกอบหนึ่ง ซึ่งเทียบเท่าได้กับการเป็นเจ้าของผู้เช่าบริการ
Tenant	**การเช่า** จะแทนบริการที่ issuer ได้เช่าบริการไว้ ซึ่งจากโครงสร้างดังกล่าวจะเห็นได้ว่า องค์กรหรือบุคคลจะสามารถเช่าใช้ได้หลายบริการ
User	**ผู้ใช้งาน** จะแทนบุคคลที่เข้าใช้บริการ
Role	**บทบาท** จะแทนการนิยามของฟังก์ชันในการทำงานสำหรับแต่ละการเช่า โดยใช้แทนการอธิบายได้ว่า “บทบาท (ชื่อการเช่า, ชื่อบทบาท)” โดยชื่อการเช่า จะเป็นค่าที่อยู่เซตของการเช่า และชื่อบทบาทเป็นการอธิบายบทบาทที่ต้องการ อาทิเช่น บทบาท (การเช่าเอ, พัฒนาระบบ) ใช้แทนบทบาทที่มีชื่อว่า พัฒนาระบบ ซึ่งเป็นบทบาทของการเช่าที่ชื่อว่า การเช่าเอ เป็นต้น ในการกล่าวถึงความสัมพันธ์เบื้องต้น ของบทบาทกับการเช่า จะสามารถกล่าวได้ว่าหนึ่งการเช่าใช้จะมีได้หลายบทบาท
Permission	**การอนุญาต** จะแทนสิทธิ์สิ่งใดที่จะอนุญาติให้ทำได้ภายใต้ฐานะในแต่ละการเช่า โดยการอนุญาต จะใช้องค์ประกอบ 3 ส่วนเพื่อใช้ในการอธิบายการอนุญาต ก็คือ (สิทธิ์ การเช่า วัตถุที่ให้สิทธิ์) โดยสิทธิ์ จะเสมือนแทนได้ว่าสามารถอ่าน สามารถแก้ไข สามารถลบ เป็นต้น สำหรับการเช่า ก็คือการเช่าที่อยู่ในเซตของ T และวัตถุที่ให้สิทธิ์ ก็คือ วัตถุที่ได้พูดถึงสิทธิ์นั้น เช่น แฟ้ม ทรัพยากรต่าง ๆ เป็นต้น ในการกล่าวถึงความสัมพันธ์เบื้องต้น ของการอนุญาตกับการเช่า จะสามารถกล่าวได้ว่าหนึ่งการเช่าใช้จะมีได้หลายการอนุญาต เช่นเดียวกันกับบทบาท
Session	**เซสชั่น** เป็นกลุ่มของข้อมูลที่ระบบนำมาใช้สำหรับตรวจสอบผู้ใช้งาน ว่าสามารถใช้งานได้อยู่หรือไม่ และอีกส่วนหนึ่ง เพื่อเก็บเซตของบทบาทย่อยที่จะถูกให้สามารถใช้งานได้กับผู้ใช้งานด้วยเช่นกัน โดยในสภาพแวดล้อมของการทำงานภายใต้คลาวน์นั้น จะสามารถบอกได้ว่า ผู้ใช้งานและบทบาท

คำนิยาม	คำอธิบาย
	ที่ได้รับ อาจจะไม่ได้มาจาก issuer เดียวกันก็ได้ ยกตัวอย่างเช่น เมื่อมีการเรียกใช้บริการห่วงโซ่แบบประสานงานกันแล้ว การตรวจสอบการจะเป็นไปในรูปแบบของการข้ามไประหว่างบริการก็เป็นได้

ในการอธิบายความเชื่อถือนั้น งานวิจัยชิ้นนี้ได้ใช้เครื่องหมาย “$\lesssim$” เป็นการอธิบายว่า เมื่อ การบริการ ก มีความเชื่อถือสำหรับ การบริการ ข เพื่อที่จะให้ใช้ตามบทบาทเอ จะสามารถเขียนแทนด้วยสัญลักษณ์ได้เป็น การบริการ ข $\lesssim$ การบริการ $ก_{บทบาทเอ}$ เขียนในรูปความเชื่อถือได้เป็น

$$b \lesssim a_{r1}$$

โดยทั่วไป สำหรับ $b \lesssim a_{r1}$ จะกล่าวได้ว่า a เป็นการเช่าที่เชื่อถือ และ b เป็นการเช่าที่ถูกเชื่อถือ และ r1 เป็นบทบาทที่ a เชื่อถือให้ b เป็นผู้มีสิทธิ์ใช้บริการตามที่บทบาท r1 กำหนดไว้ได้



ภาพที่ 3.2 แผนผังองค์ประกอบโมเดล C-MTAS

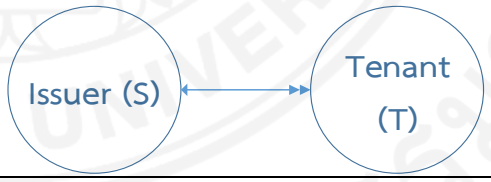
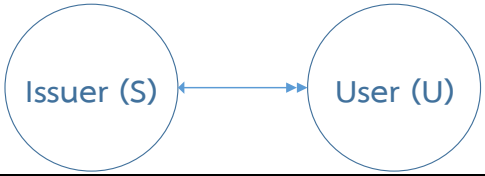
3.3.2. การอธิบายความสัมพันธ์ระหว่างองค์ประกอบทั้ง 5 เพื่อสร้างเซตในการอธิบายการตรวจสอบสิทธิ์การเข้าใช้บริการ

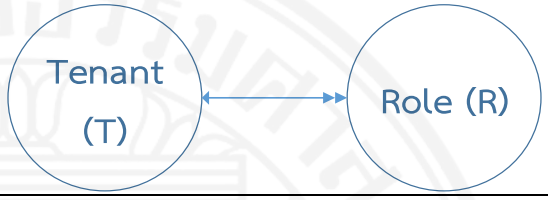
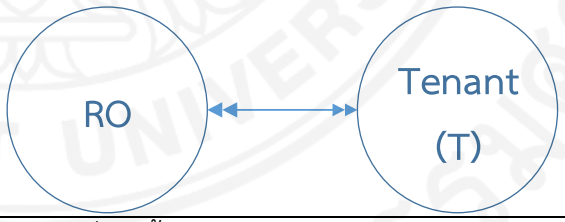
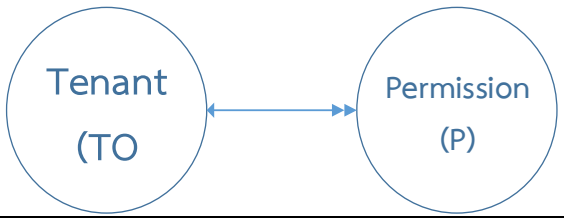
จากภาพที่ 3.2 ซึ่งเป็นการอธิบายโมเดลที่ใช้ในการอธิบายการตรวจสอบสิทธิ์การเข้าถึงนี้ จะเห็นได้ว่า องค์ประกอบต่าง ๆ จะมีความสัมพันธ์ระหว่างกัน ซึ่งความสัมพันธ์ระหว่างกันนี้ จะถูกแสดงในรูปของเซต เพื่อนำเซตจากองค์ประกอบหลักทั้ง 6 องค์ประกอบ คือ issuer การเช่า ผู้ใช้งาน บทบาท การอนุญาต และเซสชั่น นำมาประกอบเป็นความสัมพันธ์ใหม่ ซึ่งจะถูกนิยามขึ้นในนิยามที่ 2

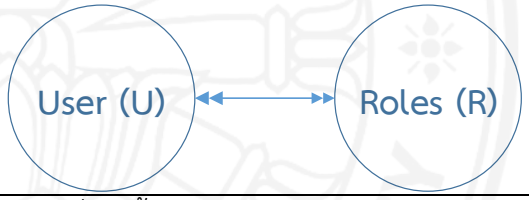
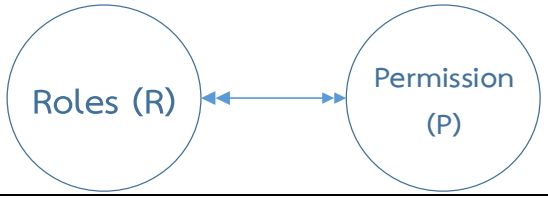
นิยามที่ 2 : โมเดลการตรวจสอบสิทธิ์การเข้าใช้ จะเป็นตามองค์ประกอบดังนี้

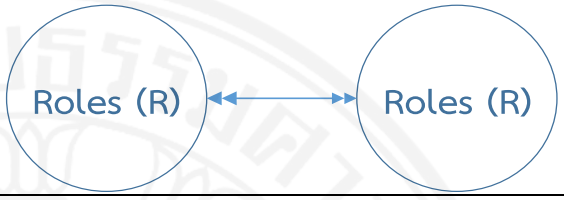
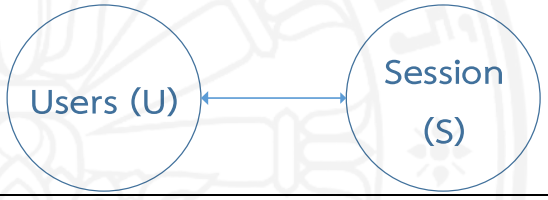
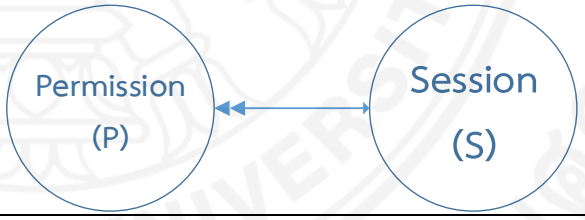
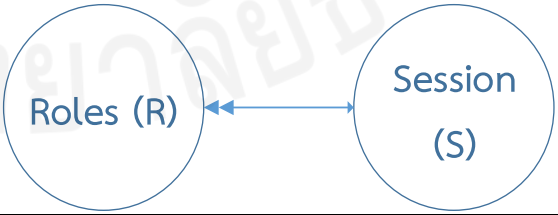
ตารางที่ 3.2

ความสัมพันธ์ในโมเดล C-MTAS

คำนิยาม	คำอธิบาย
$IO \subseteq T \times I$	ความสัมพันธ์ที่เกิดขึ้นจากหลายความสัมพันธ์ของการเช่า ไปยังความสัมพันธ์เดียวของ issuer เรียกว่า เซตของเจ้าของการเช่า $issuerOwner(t) \in \{t \in T \mid (t,i) \in IO\}$ 
$UO \subseteq U \times I$	ความสัมพันธ์ที่เกิดขึ้นจากหลายความสัมพันธ์ของผู้ใช้งาน ไปยังความสัมพันธ์เดียวของ issuer เรียกว่า เซตของเจ้าของของผู้ใช้งาน โดยมีฟังก์ชัน $userOwner(i) \in \{i \in I \mid (u,i) \in UO\}$ ในการตรวจสอบว่า issuer นั้นมี user อะไรบ้าง 
$RO \subseteq R \times T$	ความสัมพันธ์ที่เกิดขึ้นจากหลายความสัมพันธ์ของบทบาท ไปยังความสัมพันธ์ของการเช่า เรียกว่า เซตของเจ้าของของ

คำนิยาม	คำอธิบาย
	บทบาท โดยมีฟังก์ชัน $roleOwner(r: R) \rightarrow T$ ซึ่งเป็นฟังก์ชันที่หาว่าบทบาทดังกล่าว การเช่าใดเป็นเจ้าของ ซึ่งเขียนในรูปแบบของเซตได้เป็น $roleOwner(r) \in \{t \in T \mid (r,t) \in RO\}$ 
$TTRO'$ $= TTRO\ U\ \{(role1, t2, t1)\}$	ความสัมพันธ์ของการเกิดความเชื่อถือระหว่างการเช่า ในการใช้บทบาทใดบทบาทหนึ่ง กล่าวได้คือ $t1_{role1} \lesssim t2$ ก็ต่อเมื่อ $\forall t1, t2 \in T, \forall\ role1, role2 \in R$, และ $\{(role1, t1),$ $(role2, t2)\} \in RO$ ฟังก์ชัน $roleTrustee(r: R) \rightarrow T$ เป็นฟังก์ชันในการตรวจสอบบทบาทกล่าวกล่าว r มีการเช่าใดบ้างที่ถูกให้ความเชื่อถือในการใช้บทบาทนั้น ซึ่งเขียนในรูปแบบของสมการได้เป็น $roleTrustee(r) \in \{t \in T \mid (r, t, roleOwner(r)) \in TTRO\}$ 
$PO \subseteq P \times T$	ความสัมพันธ์ที่เกิดขึ้นจากหลายความสัมพันธ์ของการอนุญาตไปยังความสัมพันธ์ของการเช่า เรียกว่า เซตของเจ้าของของการอนุญาต และมีฟังก์ชัน $permOwner(p: P) \rightarrow T$ ในการตรวจสอบการอนุญาตนั้นว่าอยู่ในการเช่าหรือไม่ หรืออาจจะเขียนได้ว่า $permOwner(p) \in \{t \in T \mid (p, t) \in PO\}$ 

คำนิยาม	คำอธิบาย
$canUse(r: R) \rightarrow 2^T$	ฟังก์ชันที่ตรวจสอบเซตของบทบาท ว่าบทบาท r สามารถใช้ได้หรือไม่ โดยมีฟังก์ชัน $P_R(t: TTRO, r: R) \rightarrow T$ เป็นฟังก์ชันที่ใช้ในการตรวจสอบความเชื่อถือระหว่างการเช่า กับความเชื่อของเซตของบทบาทสาธารณะ โดยนิยามฟังก์ชันได้เป็น $CanUse(r) = \{t\} \cup \{t1 \in T \mid t1 \in P_R(t,r)\}$ เมื่อ $t = roleOwer(r)$ และ $P_R(t,r) = \{t1 \in T \mid (r, t1, t) \in TTRO\}$
$UA \subseteq U \times R$	ความสัมพันธ์ที่เกิดขึ้นจากหลายผู้ใช้งาน ไปยังหลายความสัมพันธ์ของบทบาท มีฟังก์ชัน $\mathrm{userGetRole(u: U)} \rightarrow \mathrm{R}$ เป็นฟังก์ชันที่ใช้ในการตรวจผู้ใช้งานสามารถใช้บทบาทใดได้บ้าง ซึ่งเขียนเป็นฟังก์ชันได้เป็น $\mathrm{userGetRole(u) = \{r \in R \mid (u, r) \in UA\}}$ 
$PA \subseteq P \times R$	ความสัมพันธ์ที่เกิดขึ้นจากหลายความสัมพันธ์ของการอนุญาตไปยังหลายความสัมพันธ์ของบทบาท ซึ่งมีฟังก์ชัน $PermGetRole(p: P) \rightarrow R$ เป็นฟังก์ชันที่ใช้ในการตรวจสอบว่าการอนุญาตดังกล่าวอยู่ในบทบาทใดได้บ้าง สามารถเขียนได้ว่า $PermGetRole(p) = \{r \in R \mid (p, r) \in PA\}$ และมีฟังก์ชันในการตรวจสอบการอนุญาตดังกล่าวอยู่ในการเช่าที่ถูกผู้เชื่อถือจากเจ้าของการเช่าอื่นหรือไม่ $permTrustee(p: P) \rightarrow T$ โดยเขียนเป็นสมการได้เป็น $permTrustee(p) = roleTrustee\ (PermGetRole(p))$ 

คำนิยาม	คำอธิบาย
$RH \subseteq R \times R$	ความสัมพันธ์ เพื่อสร้างลำดับชั้นของบทบาท ซึ่งใช้แทนเครื่องหมาย $\geq$ โดยเขียนได้เป็น $r \geq r_1$ และการเกิดลำดับชั้นมีเงื่อนไข $roleOwner\ (r_1) \in canUse\ (r)$ เป็นจริงด้วย 
$user(s: S) \rightarrow U$	ฟังก์ชันที่ตรวจสอบระหว่างเซสชั่น ว่ามีผู้ใช้งานดังกล่าวอยู่ในเซตของผู้ใช้งานหรือไม่ 
$permission(s: S) \rightarrow P$	ฟังก์ชันที่ใช้ในการตรวจสอบเซสชั่นของการอนุญาตที่ผู้ใช้งานได้ใช้งานการอนุญาตอะไรบ้าง 
$roles(s: S) \rightarrow 2^R$	ในเซตย่อยของบทบาท โดยจะต้องเป็นไปตาม $roles(s) \subseteq \{r \mid \exists r_2 \geq r[(user(s), r_2) \in UA \wedge issuerOwner(userOwner(user(s))) \in canUse(r)]\}$ 

ตามที่ได้นิยามที่ 2 นั้น จะเป็นการอธิบายความสัมพันธ์ระหว่างกัน ภายในโมเดลการตรวจสอบการเข้าถึง และสำหรับการตรวจสอบสิทธิ์นั้น ก็จะนำความสัมพันธ์ของเซตของ PO, RO, RH, และ U มาใช้ในการอธิบาย โดยเริ่มต้นจากดูว่า วัตถุที่ต้องตรวจสอบสิทธิ์การเข้าถึงนั้น ถูกนำมาใช้การบทบาทใดได้บ้าง ซึ่งรวมทั้งในบทบาทในเซตของ RO และบทบาทที่ถูกถ่ายทอดให้กับผู้ใช้งานที่มีความเชื่อถือรองรับ โดยเมื่อทราบถึงบทบาทใดบ้างแล้วนั้น ก็นำมาหาความสัมพันธ์ต่อว่าบทบาทนั้นสามารถนำมาใช้กับผู้ใช้งานใดได้บ้าง และเซตของผู้ใช้งานที่มีความสัมพันธ์กับบาทบาทดังกล่าว ตรงกับผู้ใช้งานที่กำลังเข้าใช้งานอยู่หรือไม่ ซึ่งถ้าหากตรง ก็แสดงได้ว่า ผู้ใช้งานดังกล่าวมีสิทธิ์ที่ใช้งานกับวัตถุที่ต้องการใช้ได้ แต่หากไม่พบความสัมพันธ์ระหว่างผู้ใช้งาน, บาทบาท และการอนุญาตแล้ว ในโมเดลดังกล่าว ได้อธิบายลักษณะที่ไม่มีความสัมพันธ์ได้ว่า “ไม่มีสิทธิ์ที่จะเข้าใช้งานกับวัตถุนั้น ๆ”

นิยามที่ 3 : ฟังก์ชันในการจัดการในการบริหารการควบคุมการเข้าถึง

เป็นการนิยามฟังก์ชันต่าง ๆ เพื่อใช้ในการจัดการกับการควบคุมการเข้าถึง โดยสรุปได้เป็นดังนี้

ตารางที่ 3.3

นิยามฟังก์ชันในการจัดการในการบริหารการควบคุมการเข้าถึง

ฟังก์ชัน	เงื่อนไข	การกระทำ
$assignUser$ (t,r,u)	$u \in U \ \wedge t \in$ $(\ roleOwner(r) \vee roleTrustee(r)\ )$	$UA' = UA \cup$ $\{u \to r\}$
$revokeUser$ (t,r,u)	$u \in U \ \wedge$ $(\ roleOwner(r) \vee roleTrustee(r)\ ) \wedge$ $u \to r \in UA$	$UA' = UA \setminus \{u \to r\}$
$assignPerm$ $(t,r,p)\ )$	$t = permOwner\ (p) \wedge t \in canUse\ (r)$	$PA' = PA\ \cup\ \{p \to r\}$
$revokePerm$ (t,r,p)	$t = permOwner(p) \wedge t \in canUse(r)$ $\wedge \mathrm{p} \to \mathrm{r} \subset \mathrm{PA}$	$PA' = PA \setminus \{p \to r\}$
$assignRH$ $(t,r1,r)$	$t = roleOwner(r) \wedge i \in canUse(r1) \wedge$ $\neg\ (r1 \gg r) \wedge \neg\ (r \geq r1)^a$	$\geq' = \geq \sqcup \{r2, r3: R \mid r2$ $\geq r1 \wedge r$ $\geq r3$ $\wedge roleOwner\ (r3)$ $\in canUse\ (r2) \cdot r2$ $\to r3)$

ฟังก์ชัน	เงื่อนไข	การกระทำ
$revokeRH$ $(i, r1, r)$	$t = roleOwner(r) \wedge i \in canUse(r1) \wedge (r1 \gg r^b)$	$\geq' = (\geq \backslash \{r1 \rightarrow r)^{*c}$
$assignTrust$ $(r1, t1, t)$	$t1 \in T \wedge t \in T \wedge r1 \in R \wedge t = roleOwner(r1) \wedge t1 \notin roleTrustee\ (r1)$	$TTRO' = TTRO\ U$ $\{(r1, t1, t)\}$
$revokeTrust$ $(r1, t1, t)$	$t1 \in T \wedge t \in T \wedge r1 \in R \wedge t = roleOwner(r1) \wedge t1 \in roleTrustee\ (r1)$	$TTRO' = TTRO \backslash$ $\{(r1, t1, t)\}$

นิยามที่ 4 : ฟังก์ชันในการตรวจสอบการเข้าถึง

เป็นการนิยามฟังก์ชัน เพื่อใช้ตรวจสอบสิทธิ์การเข้าถึง จากผู้ใช้งาน และการอนุญาต ซึ่งนิยามได้เป็น $authorize(u: U, r: R, p: P) \rightarrow \{True, False\}$ ซึ่งฟังก์ชันดังกล่าวเขียนเป็นสมการได้เป็น

$$authorize(u, r, p) \Rightarrow \{PermGetRole(p) \cap (userGetRole(u) \cap \{r\})\} \neq \emptyset$$

ซึ่งหากผลลัพธ์เป็น $\emptyset$ แสดงว่าผู้ใช้งาน *U* นั้นไม่มีสิทธิ์ในการเข้าถึงสำหรับ *P* ดังกล่าว แต่หากไม่ใช่ $\emptyset$ แสดงว่าผู้ใช้งาน *U* มีสิทธิ์ในการเข้าถึงสำหรับ *P* นั้น

นิยามที่ 5 : เซตสำหรับการเรียกบริการแบบห่วงโซ่การเรียกบริการแบบประสานงาน

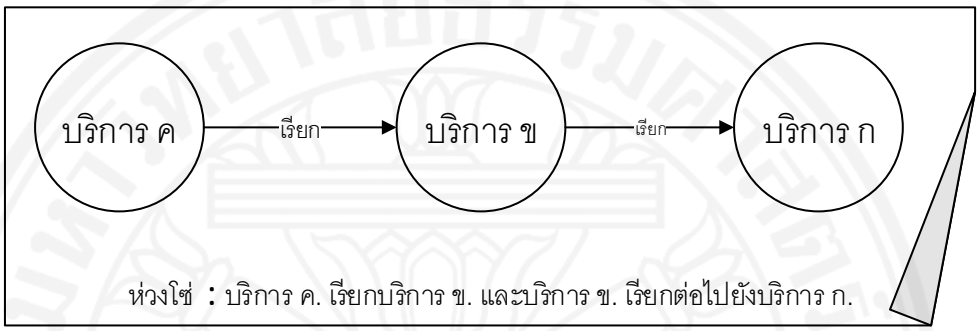
เมื่อมีการกำหนดให้มีห่วงโซ่การเรียกบริการแบบประสานงาน จำเป็นจะต้องมีข้อกำหนดเพื่อบางถึงห่วงโซ่ที่สามารถเกิดขึ้นได้ ดังนั้นผู้วิจัยจึงได้กำหนด Chain Calling Permission Relation หรือ CCPR

$$CCPR \subseteq P \times P$$

เป็นการสร้างความสัมพันธ์จากหนึ่งความสัมพันธ์ไปยังหนึ่งความสัมพันธ์ของการอนุญาตไปยังการอนุญาต เพื่อบอกถึงเซตที่สามารถเกิดการเรียกบริการแบบประสานงานเกิดขึ้นได้ โดยมีฟังก์ชันเพื่อใช้ในการตรวจสอบคือ $getPermissionToChain(p) \rightarrow P$ หรือสามารถเขียนได้เป็น $getPermissionToChain(p1) = \{p \in P | (p, p1) \in CCPR\}$

3.3.3. การบริการแบบห่วงโซ่การเรียกใช้แบบประสานงาน

ในรูปแบบของห่วงโซ่การเรียกบริการแบบประสานงานนั้น เป็นการเรียกใช้บริการที่สืบเนื่องไปยังบริการอื่น ๆ เพื่อต้องการข้อมูลหรือเข้าถึงทรัพยากรที่ผู้เรียกใช้บริการต้องการ โดยในสภาพแวดล้อมที่อยู่บนความหลากหลายในการเช่านั้น ซึ่งความสัมพันธ์ในการเรียกใช้บริการจะเป็นไปตามภาพที่ 3.3



ภาพที่ 3.3 ภาพแสดงห่วงโซ่การเรียกบริการแบบประสานงาน

จากปัญหาของการกำหนด cross-issuers ที่หากมีการกำหนดเพิ่มขึ้น ต้องมีการเพิ่มไฟล์ *2* ไฟล์ ซึ่งทำให้การจัดการไฟล์มีความยุ่งยาก รวมทั้งเมื่อไฟล์นโยบายมีการโตขึ้น จะก่อให้เกิดการใช้ทรัพยากร IO ในการตรวจสอบที่มากขึ้น ดังนั้น เพื่อแก้ปัญหาดังกล่าว ผู้วิจัยจึงได้กำหนดไฟล์สำหรับการแยกการกำหนดนโยบายการเข้าถึงสำหรับการเกิดห่วงโซ่การบริการแบบประสานงานออกมา โดยกำหนดจาก CCPRSET ดังที่กำหนดในนิยามที่ *5* และเพื่อป้องกันไม่ให้การบริการการตรวจสอบมีการจองทรัพยากรการตรวจสอบที่เกินความจำเป็น ซึ่งหนึ่งในนั้นคือการบริการที่มีการวนไม่รู้จบ ผู้วิจัยจึงดำเนินการตรวจสอบจากข้อมูลในเซสชั่นว่าหากมีการเรียกบริการไปยัง *permission* ที่ร้องขอมานั้นเกิด *cycle* หรือไม่ ซึ่งหากเกิด *cycle* จึงตัดการขอร้องนั้น ให้ผู้ร้องขอเรียกบริการมาใหม่ เพื่อที่จะสามารถรองรับการเรียกการร้องขอได้ ซึ่งได้สรุปการตรวจสอบสิทธิ์การเข้าถึงในห่วงโซ่การบริการแบบประสานงานเป็นตามนิยามที่ *6* ดังนี้

นิยามที่ 6 : การตรวจสอบสิทธิ์การเข้าถึงเมื่อมีลักษณะของห่วงโซ่การเรียกแบบประสานงาน

เพื่อตรวจสอบสิทธิ์จากเจ้าของการเช่าว่ามีการให้สิทธิ์แล้วหรือไม่ $authorizeChainCall(u{:}\,U, s{:}\,S, p{:}\,P) \rightarrow \{True, False\}$ โดยเขียนเป็นฟังก์ชันได้เป็น

$$authorizeChainCall(u, s,\ p) \Rightarrow \{\ (\ user(s) \cap \{u\}\ ) \cap (getPermissionToChain(p) \cap (permission(s)_{last})\} \neq \emptyset$$

ถ้าผลลัพธ์เป็น is $\emptyset$ แสดงว่าจะต้องใช้ฟังก์ชัน $\mathbf{authorize(u, r, p)}$ ในการตรวจสอบสิทธิ์ต่อไป แต่ถ้าไม่เป็น $\emptyset$ ถือได้ว่า ผู้ใช้งานดังกล่าว สามารถใช้บทบาทในการเข้าถึงการอนุญาตที่ร้องขอนั้นมาได้ ด้วยเหตุผลว่าเจ้าของการเช่าบริการที่ร้องขอนั้น ได้เคยให้สิทธิ์ของผู้ร้องขอดังกล่าวไว้แล้วในกระบวนการของห่วงโซ่การเรียกบริการแบบประสานงานแล้ว

3.4. การพัฒนาระบบการตรวจสอบสิทธิ์จากโมเดลการตรวจสอบสิทธิ์ที่ได้กำหนดขึ้น

ในขั้นตอนการของพัฒนาจากโมเดล C-MTAS เพื่อทดลองให้เห็นว่าโมเดล C-MTAS นั้นสามารถนำไปใช้งานได้จริงตามรูปแบบโมเดลที่ถูกนำเสนอในงานวิจัยชิ้นนี้ โดยผู้วิจัยได้นำข้อกำหนดเพื่อสร้างนโยบาย eXtensible Access Control Markup Language (XACML) เวอร์ชั่น 3.0 มาใช้เป็นพื้นฐานในการพัฒนา

โดยจากพื้นฐานของ XACML ที่ได้อธิบายวากยสัมพันธ์ ในการเขียน XML เพื่ออธิบายความสัมพันธ์เพื่อใช้ในการตรวจสอบการเข้าถึง โดยใช้องค์ประกอบระหว่าง Users, Roles, Objects, Operations, Permissions เพื่อประกอบกันเป็นนโยบายสำหรับการตรวจสอบสิทธิ์การเข้าถึง

โดยกระบวนการเพื่อใช้ในการตรวจสอบการเข้าถึงสำหรับการ implementation ในโมเดล C-MTAS นี้ ในงานวิจัยชิ้นนี้ได้ใช้ชื่อว่า Authorization as a Services (AaaS) ทำหน้าที่การบริการเพื่อใช้ในการตรวจจับเมื่อมีผู้ร้องขอบริการ ในการตรวจสอบสิทธิ์การเข้าถึงของทรัพยากรที่ร้องขอมา โดยผลลัพธ์จากการร้องขอนั้น จำแนกได้เป็น 4 ผลลัพธ์ กล่าวได้คือ

- ***Permit :*** *อนุญาตให้ผู้ร้องขอเข้าถึงทรัพยากรที่ร้องขอมาได้*
- ***Deny :*** *ไม่อนุญาตให้ผู้ร้องขอเข้าถึงทรัพยากรที่ร้องขอมาได้*
- ***Not Applicable :*** *มีการกำหนดแต่ไม่ชัดเจน ซึ่งไม่ได้ผลลัพธ์อนุญาตหรือไม่*
- ***Indeterminate :*** *พบนโยบายที่กำหนดในการตัดสินใจ แต่ระหว่างการตัดสินใจนั้นเกิด Error ขึ้น*

และขั้นตอนในการตรวจสอบการเข้าถึงใน AaaS จะเป็นไปตามกระบวนการในภาพที่ 3.3 โดยมีขั้นตอนดังนี้

เริ่มต้นจากผู้ต้องการใช้บริการร้องขอเพื่อเข้ามายังการตรวจสอบของ AaaS แล้ว โดย AaaS ส่งการร้องขอดังกล่าวเข้ามายัง Policy Enforcement Point (PEP) โดย PEP จะส่งต่อมายัง context handler ซึ่งในส่วนนี้แบ่งการตรวจสอบออกเป็น 2 ขั้นตอน คือ

(2.1) ขั้นตอนที่หนึ่ง ตรวจสอบข้อบังคับ ในการตรวจสอบลักษณะของ Cycle โดยตรวจสอบจากเซสชั่น ซึ่งเมื่อเซสชั่นมีลักษณะ Cycle ที่ผู้วิจัยได้นำอัลกอริทึม SzwarcfiterLauer มาใช้ในการตรวจสอบ เพื่อสร้างเป็นข้อจำกัดของ CCPR โดยเมื่อมีการสร้างเซสชั่นของการเรียกของบทบาทกรณีหากเกิดห่วงโซ่ที่มีพฤติกรรมที่ส่อในทางการในลักษณะของ Cyclic โดยตรวจสอบใน Session ของผู้ใช้งานว่ามีการเรียกใช้ผ่าน permission เดิมหรือไม่ หากเกิดใน permission เดิม จึงถือได้ว่าอาจจะเกิดลักษณะของ cycle จึงส่งผลลัพธ์กลับไปยังผู้ร้องขอว่าเป็น “Indeterminate”

(2.2) ขั้นตอนที่สอง เมื่อตรวจสอบข้อบังคับทั้ง 2 เป็นการตรวจสอบการร้องขอที่ส่งมาหากเป็นแบบห่วงโซ่การเรียกใช้บริการแบบประสานงาน จะตรวจสอบว่าเจ้าของการเช่าได้เคยอนุญาตให้กับผู้ร้องขอแล้วหรือไม่ หากมีการให้สิทธิ์แล้ว ถือได้ว่าผลลัพธ์จะเป็น “Permit”

```
Context handler considers the request

//case "chain calling coordination"

Var session_ID := get current Session ID;

Var roundCalling := Loop count in Session Table Role+1;

Loop in Session Table Role

// step1 validate cyclic calling

  If Search Graph Session has Cycle (SzwarcfiterLauer Algorithm)
  for Role request and Permission request

        Return Indeterminate;

// step2 search ownership

If Search Tenant Owner to request has Match &&

   Permission request in constraint CCPR

        Return permit;

// other case "normal calling", send to Normal PEP to evaluate

Return send request to PDP
```

ภาพที่ 3.4 อัลกอริทึมการตรวจสอบสิทธิ์การเข้าถึง

(1) เมื่อ Context handler ตรวจสอบในเรื่องของการเรียกแบบห่วงโซ่แล้ว ตรวจสอบเจ้าของการเช่ายังไม่มีการตรวจสอบ หรือเป็นการเรียกบริการใหม่ จะถูกส่งมาตรวจสอบต่อในส่วนของ Policy Decision Point (PDP) ซึ่งในส่วนนี้จะตรวจสอบจาก policy ที่ถูกกำหนดไว้จาก 3 ส่วน คือ

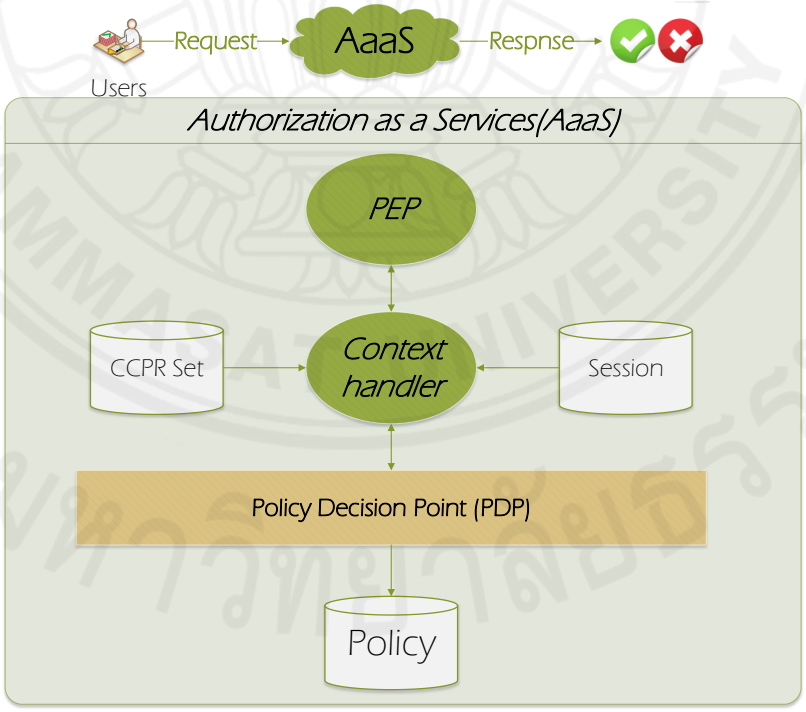
(3.1) Trust Policy เป็นนโยบายที่กำหนดขึ้นเพื่อแสดงความสัมพันธ์ของความเชื่อถือระหว่างกันของแต่ละการเช่า โดยจะประกอบไปด้วย (บทบาท ผู้เชื่อถือ ผู้ถูกเชื่อถือ)

(3.2) Tenant Policy เป็นนโยบายที่กำหนดขึ้นเพื่อแสดงความสัมพันธ์ของ issuer การเช่า และผู้ใช้งาน เพื่อใช้ในการตรวจสอบหาเจ้าของของการเช่าสำหรับในแต่ละผู้ใช้งาน

(3.3) Authorization Policy เป็นนโยบายที่กำหนดขึ้นเพื่อแสดงความสัมพันธ์ของบทบาท ผู้ใช้งาน การอนุญาต เพื่อใช้ในการตรวจสอบการเข้าถึงของทรัพยากรที่ผู้ใช้งานร้องขอเพื่อใช้ในการตรวจสอบการเข้าถึง

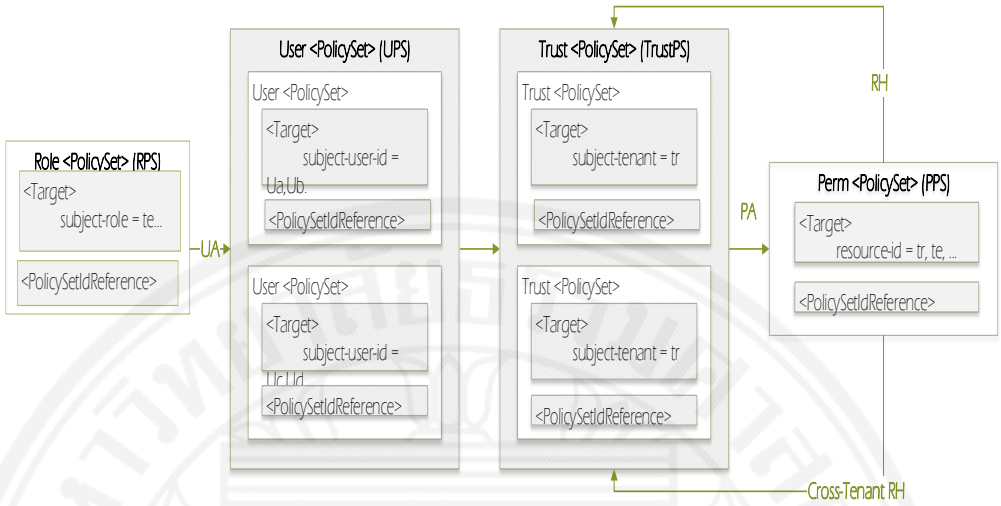
(4) เมื่อ PDP(Policy Decision Point) ตัดสินใจจากนโยบาย ได้ผลลัพธ์ของการตรวจสอบสิทธิ์การเข้าถึงจะดำเนินส่งผลลัพธ์กลับไปให้กับ PEP เพื่อตอบสนองไปยังผู้ร้องขอต่อไป

(5) ในกรณีที่การตอบสนองไปยังผู้ร้องขอการตรวจสอบ ผลลัพธ์เป็น Permit PEP จะเก็บเซสชั่นในส่วนของบทบาท โดยเก็บ (Session_id User_name seq Tenant Role Permission Reference_permission Reference_tenant Status) ไว้ใน Table Session Role



ภาพที่ 3.5 แผนผังสถาปัตยกรรมซอฟต์แวร์เพื่อใช้ในการตรวจสอบสิทธิ์การเข้าถึงของโมเดล C-MTAS

และจากโครงสร้างพื้นฐานของ XACML นั้น ผู้วิจัยได้นำมาประกอบ เพื่อใช้ในการอธิบายโมเดล ได้เป็นตามภาพที่ 3.5



ภาพที่ 3.6 แผนภาพประกอบการอธิบาย XACML ที่ใช้ในการอธิบายโมเดล C-MTAS

จากภาพที่ 3.6 แสดงให้เห็นถึงนโยบายของโมเดล C-MTAS ซึ่งเป็นการอธิบายด้วย XACML โดยแบ่งออกได้เป็น

(1) **การทำงานในสภาพปกติ** ที่ยังไม่ได้กำหนดในรูปแบบของการข้ามการเช่า โดยเริ่มต้นจาก RPS ตามภาพที่ 3.6 โดยเมื่อผู้ใช้งานร้องขอโดยอ้างอิงผ่านบทบาท te เข้ามา และ RPS ตรวจสอบว่าผู้ใช้งานดังกล่าวสามารถใช้บทบาท te ได้โดยส่งไปยัง UPS ในการตรวจสอบเซ็ตของผู้ใช้งานที่ร้องขอ และส่งมาตรวจสอบต่อไปยัง TrustPS เพื่อตรวจสอบความเชื่อถือระหว่างกัน โดยในที่นี้บทบาทดังกล่าวเป็นการกำหนดในการใช้งานของการเช่า tr เองจึงทำให้ tr ถูกเชื่อถือตัวเองอยู่แล้ว จะส่งต่อมายัง PPS เพื่อตรวจสอบต่อไปว่าบทบาทดังกล่าวสามารถใช้งานในทรัพยากรที่ผู้ใช้งานร้องขอมาได้หรือไม่ หากพบทรัพยากรดังกล่าวอยู่ในการอนุญาตของ PPS จึงถือได้ว่าผู้ใช้งานมีสิทธิ์ในการเข้าถึงทรัพยากรที่ร้องขอมา แต่หากไม่พบจะถือได้ว่าผู้ใช้งานไม่มีสิทธิ์ในการใช้ทรัพยากรที่ร้องขอมา

ในกรณีที่ PPS มีการสืบทอดบทบาทต่อกันไป สำหรับ XACML จะมีการใช้ PolicySetIdReference เพื่ออ้างอิงบทบาทไปจนถึงบทบาทย่อยสุด เพื่อหาการอนุญาตต่อไป โดยในภาพที่ 3.6 จะอยู่ในเส้นของ RH โดย PPS จะเชื่อมกันไปยัง TrustPS เพื่อหาบทบาทย่อยสุด แล้วจึงนำมาตรวจสอบการอนุญาตใน PPS ต่อไป

(2) **การทำงานในสภาพที่ยังมีการกำหนดบทบาทให้สามารถใช้งานในการข้ามของการเช่าได้** โดยเริ่มต้นจาก RPS ตามภาพที่ 3.6 โดยเมื่อผู้ใช้งานร้องขอโดยอ้างอิงผ่านบทบาท tr เข้ามา และ RPS ตรวจสอบว่าผู้ใช้งานดังกล่าวสามารถใช้บทบาท tr ได้ โดย RPS จะดำเนินการส่งต่อมายัง UPS เพื่อดำเนินการตรวจสอบหาการเช่าที่เป็นเจ้าของ โดยในที่นี้คือการเช่า te และส่งมาตรวจสอบต่อไปยัง TrustPS ว่าบทบาท tr เพื่อตรวจสอบความเชื่อถือระหว่างกัน จากตัวอย่างใน TrustPS ในส่วนด้านล่างคือการกำหนดความเชื่อถือระหว่างการเช่า te กับ tr ในการใช้บทบาท tr ดังนั้นในส่วนของ TrustPS จะถือได้ว่ามีการกำหนดไว้อย่างถูกต้อง จึงดำเนินการส่งต่อมายัง PPS โดยในกรณีนี้ PPS ถือได้ว่ามีการกำหนดในรูปแบบของ RH ที่เป็นการข้ามการเช่า (Cross-tenant RH) ดังนั้นจะส่งกลับไปอ้างอิงกับ TurstPS เพื่อหาบทบาทที่ย่อยที่สุด และตรวจสอบการอนุญาตว่ามีการกำหนดการเข้าถึงหรือไม่ โดยหากมีกำหนดการเข้าถึงทรัพยากรดังกล่าวอยู่ในการอนุญาตของ PPS จึงถือได้ว่าผู้ใช้งานมีสิทธิ์ในการเข้าถึงทรัพยากรที่ร้องขอมา แต่หากไม่พบจะถือได้ว่าผู้ใช้งานไม่มีสิทธิ์ในการใช้ทรัพยากรที่ร้องขอมานั้นเอง

3.5. กระบวนการทดสอบหลังจากการ implement

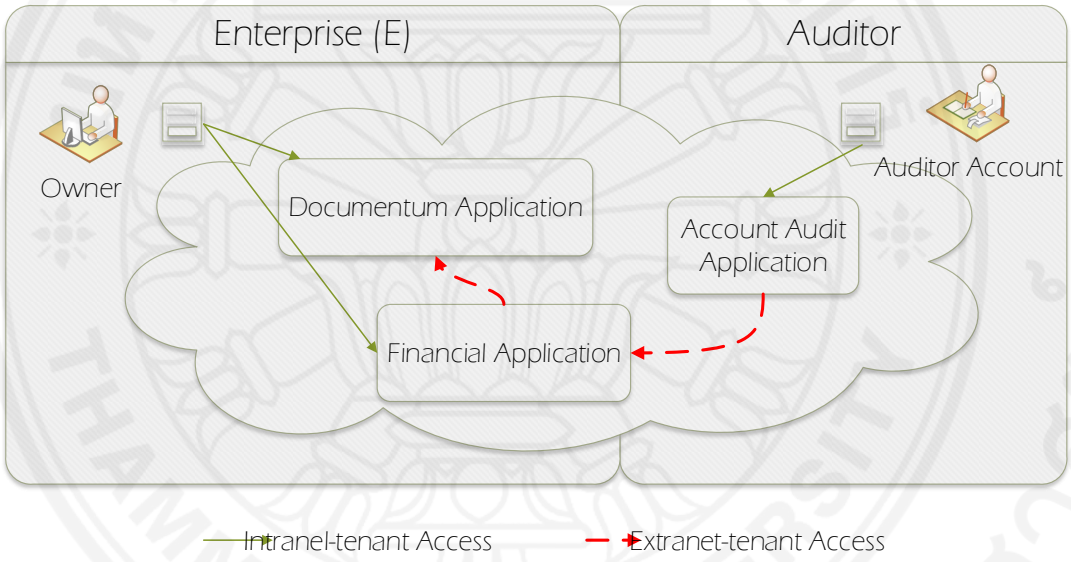
เมื่อ Implement จากส่วนในข้อ 3.4 เรียบร้อยแล้ว ผู้วิจัยจึงดำเนินการทดสอบด้วย 2 ขั้นตอนในการทดสอบ กล่าวได้คือ

3.5.1. ขั้นตอนการทดสอบการทำงานของ C-MTAS โดยนำตัวอย่างมาใช้ในการทดสอบ

สำหรับกระบวนการเพื่อใช้ในการทดสอบถูกต้องหลังขั้นตอนของการ implement นี้ ในงานวิจัยชิ้นนี้ได้เน้นถึงขั้นตอนของการตรวจสอบสิทธิ์การเข้าถึง (Authorization) ดังนั้นเมื่อผู้ร้องขอการตรวจสอบสิทธิ์นี้ ในโมเดลจึงไม่ได้มีการพิสูจน์ตัวตน (Authentication) ได้ดำเนินการเพื่อการตรวจสอบตามนโยบายการเข้าถึงที่เจ้าของซอฟต์แวร์เป็นผู้ตั้งบทบาทให้กับผู้เช่าเท่านั้น และผู้วิจัยได้จำลองโดยใช้ตัวอย่างโดยมีรายละเอียดของการจำลองตามภาพที่ 3.7 ซึ่งแบ่งเป็นองค์ประกอบได้เป็น 2 องค์ประกอบ คือ Enterprise (E) ซึ่งคือบริษัท โดยมีผู้ใช้งาน คือ Alice Bob Charles Dan และบริษัทนี้มีการจำลองการใช้ระบบ Documentum Application (DocApp) และระบบบัญชีการเงิน Financial Application (FinanApp) โดยมีการทำงานดังนี้ ระบบ Documentum เป็นระบบที่ใช้ในการจัดเก็บเอกสาร ซึ่งบริษัทนำมาใช้ในการจัดเก็บเอกสารสำหรับข้อมูลทางการเงินและบัญชี และระบบ Financial Application จะต้องมีสิทธิ์เข้าใช้ในงานในการจัดเก็บเอกสาร เพื่อแนบไฟล์ แก้ไขไฟล์ และดูไฟล์ได้ ในการแนบไฟล์เช่น ใบเสร็จรับเงิน ใบจ่ายเงิน เช็คต้นฉบับ สำเนาเอกสารที่ใช้รับเช็ค-สั่งจ่ายต่าง ๆ โดยมี Alice เป็นเจ้าหน้าที่ดูแลระบบทำหน้าที่สร้างและแก้ไขไดเร็กทอรี่ต่าง ๆ ในการจัดเก็บ Bob เป็นเจ้าหน้าที่ทางการเงิน และ Charles เป็นเจ้าหน้าที่การบัญชี ในทุก ๆ สิ้นปี บริษัทต้องแจ้งกับบริษัทตรวจสอบบัญชี โดยได้จ้าง Dan เข้ามาตรวจสอบและเซ็นรับรองบัญชีบริษัท ซึ่งบริษัทตรวจสอบบัญชีใช้ Account Audit Application (AuditApp) เข้ามาทำรายงาน รวมทั้งตรวจทานยอดทางบัญชี จึงต้องมีข้อมูลเบื้องต้นจากระบบ Financial Application สำหรับการประมวลผลตรวจทานบัญชี โดยผู้ใช้งานตาม

ภาพที่ 3.7 มีบทบาทในการทำงานที่แตกต่างกันดังที่ได้กล่าวไว้แล้ว และก็มีสิทธิ์ใช้งานข้ามกันระหว่างการเช่าบริการตามที่เห็นในเป็นเส้นประสีแดงในภาพ โดยมีความต้องการในการกำหนด External-tenant Access กล่าวได้คือ

- ระบบ Documentum จะต้องให้ความเชื่อถือกับระบบ Financial ในการใช้บทบาท Upload File (R2) และ Preview File (R3) สำหรับผู้ใช้งาน Bob และ Charles แต่ไม่มีสิทธิ์ในบทบาท Manage Directory (R1)
- ระบบ Financial จะต้องให้ความเชื่อถือกับระบบ Account Audit ในการใช้บทบาท Financial audit (R6) ในการตรวจทานทางบัญชีให้กับผู้ใช้งาน Dan ของระบบ Account Audit ได้



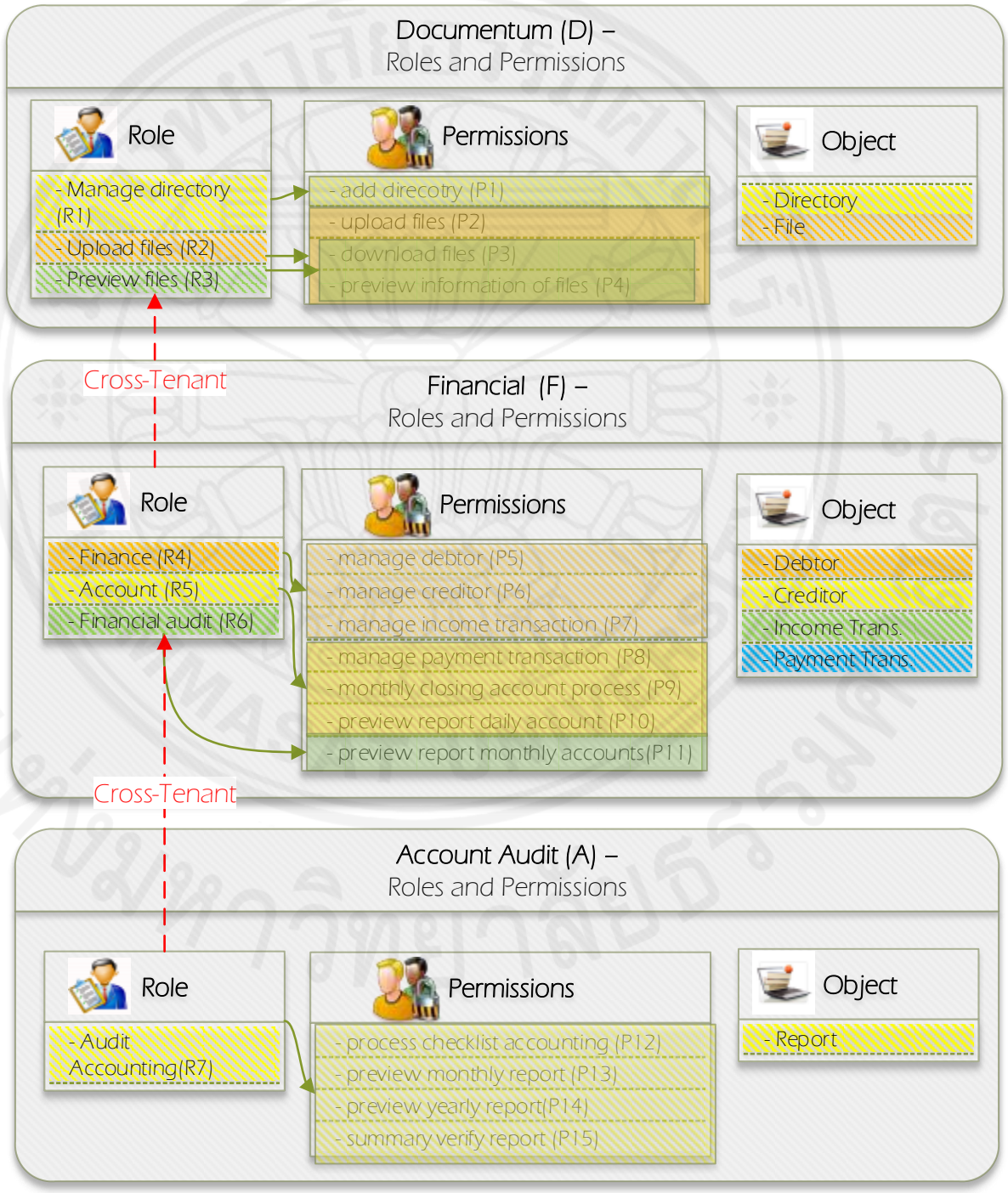
ภาพที่ 3.7 แผนภาพแสดงเหตุการณ์จำลองเพื่อใช้ในกระบวนการทดสอบ

และในส่วนของการกำหนดบทบาท และการอนุญาตผู้วิจัยได้สรุปเป็นไปตามภาพที่ 3.7 ซึ่งถูกแบ่งบทบาทออกเป็น 7 บทบาท โดยแต่ละบทบาทก็เชื่อมไปยังสิทธิ์ในของการอนุญาตในสิทธิ์การเข้าถึงแบบปกติทั่วไป ได้แสดงให้เห็นในเส้นของสีเขียว และในส่วนของการกำหนดสิทธิ์ข้ามระหว่างเจ้าของการบริการ ในการแสดงให้เห็นเป็นเส้นสีแดง

จากที่ผู้วิจัยได้กล่าวถึงตัวอย่างเพื่อใช้ในการทดสอบนั้น ผู้วิจัยต้องนำบทบาทและการอนุญาตใน

ภาพที่ *3.8* เขียนออกมาเป็นความสัมพันธ์เพื่อสร้างเป็น XACML file ตามที่ผู้วิจัยได้กำหนดไว้ในภาพที่ 3.5 และหลังจากนั้นผู้วิจัยจึงเขียนกรณีทดสอบจากการกำหนดการเข้าถึงของผู้ใช้งานทั้ง 4 ผู้ใช้งานคือ Alice Bob Charles Dan เพื่อทดสอบการเข้าถึงบทบาททั้ง 7 บทบาท และการอนุญาตทั้ง 15 การอนุญาต โดยผู้วิจัยได้เขียนกรณีทดสอบทั้งหมดออกได้เป็น 420 กรณี

ทดสอบ โดยผู้วิจัยได้เขียนผลลัพธ์ในการทดสอบไว้ในแต่ละกรณีทดสอบ เพื่อทดสอบกระบวนการตรวจสอบสิทธิ์ที่ผู้วิจัยได้ implement ขึ้นเพื่อทดสอบเปรียบเทียบกับกรณีทดสอบทั้งหมดและสรุปผลการทดสอบต่อไป ซึ่งในกรณีการทดสอบดังกล่าว ยังไม่รวมในส่วนของการเรียกบริการแบบประสาน เนื่องจากเป็นการตรวจสอบสิทธิ์ในการเรียกบริการแค่ 1 บริการ ยังไม่มีการกำหนดเพื่อเกิดการเรียกบริการแบบประสานงานกันเกิดขึ้น

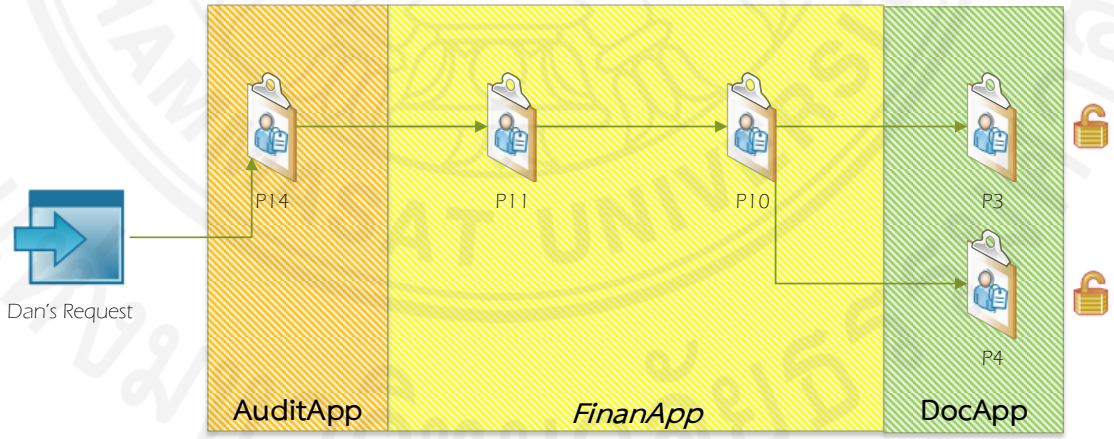


ภาพที่ 3.8 ความสัมพันธ์ของการกำหนดบทบาทและการอนุญาตในการเช่าของตนเอง (เส้นทึบ) รวมทั้งการกำหนดบทบาทในการข้ามกันของแต่ละการเช่า (เส้นประ)

3.5.2. ขั้นตอนการทดสอบการทำงานของ C-MTAS ในการตรวจสอบสิทธิ์ในรูปแบบของห่วงโซ่การเรียกบริการแบบประสานงาน

ในกรณีทดสอบที่ใช้สำหรับการทดสอบในเรื่องของห่วงโซ่การเรียกบริการแบบประสานงานนั้น ผู้วิจัยได้ใช้จากตัวอย่างเดียวกันจากขั้นตอน 3.5.1 แต่ได้เพิ่มเงื่อนไขในส่วนของแอพพลิเคชั่น เพื่อให้เกิดการห่วงโซ่การเรียกบริการแบบประสานงาน โดย Dan มีสิทธิ์ใช้บาทบาท R7 ในการเรียกใช้จาก AuditApp ซึ่งได้ถูกกำหนดเพื่อให้ใช้ในการเรียกรายงานเพื่อตรวจสอบดูข้อมูลทางบัญชีรายปี โดยการบริการนี้ จำเป็นต้องดึงข้อมูลในการบริการของระบบ FinanApp ในการใช้ข้ามบทบาทไปยัง R6 เพื่อใช้สิทธิ์ของ P11 ซึ่งเป็นสิทธิ์ในการเรียกรายงานทางบัญชีในแต่ละเดือนที่ได้ถูกปิดงวดแล้ว โดยในกระบวนการนี้เองจำเป็นต้องใช้ข้อมูลเพื่อดูรายงานทางบัญชีประจำวันต่อดังนั้นจึงในสิทธิ์ของ P10 และระหว่างการตรวจสอบรายงานประจำวันอยู่ ผู้ตรวจสอบจำเป็นต้องใช้เอกสารเพิ่มเติมที่เกี่ยวข้องกับการรับจ่ายประจำวัน ซึ่งได้ถูกเก็บไว้ในระบบ DocApp ในการใช้สิทธิ์ข้ามการเช่าของบทบาท R3 เพื่อใช้สิทธิ์ P3 ในการเพื่อดาวน์โหลดไฟล์มาดำเนินการตรวจสอบ หรือต้องการตรวจสอบข้อมูลเกี่ยวกับไฟล์ในสิทธิ์ของ P4 ก็ได้ ซึ่งจากกฎทางธุรกิจที่กล่าวถึง สามารถเขียนภาพแสดงความสัมพันธ์ของห่วงโซ่การเรียกใช้บริการได้ตาม

ภาพที่ 3.9



ภาพที่ 3.9 กรณีทดสอบ ในห่วงโซ่การบริการแบบประสานงาน

ซึ่งขั้นตอนการทดสอบนั้น ผู้วิจัยได้เขียนกรณีทดสอบจากการตรวจสอบการเข้าถึงของ Dan ทั้งหมดและรันแอพพลิเคชั่น เพื่อตรวจสอบจากกรณีทดสอบทั้งหมด ซึ่งจะต้องตรงตามความคาดหวังของกรณีทดสอบ จึงถือว่าการทดสอบเรียบร้อย

3.6. วิเคราะห์ในเชิงการเปรียบเทียบ เรื่องของการปรับขนาดของฮาร์ดแวร์ในการรองรับการทำงานของ C-MTAS

หลังกระบวนการทดสอบได้ทดสอบเรียบร้อยแล้ว ผู้วิจัยได้วิเคราะห์ในแง่มุมหนึ่งเพื่อเปรียบเทียบปริมาณในการรองรับ AaaS ของโมเดล C-MTAS สำหรับขนาดในเชิงฮาร์ดแวร์ เพื่อหาฮาร์ดแวร์ที่เหมาะกับการทำงานใน AaaS นี้ที่สุด โดยเปรียบเทียบขนาดของฮาร์ดแวร์จำนวน 4 ชุดด้วยกัน โดยได้อาศัยการทำการเครื่องจำลอง (Visualization Machine) โดยผู้วิจัยได้นำการทำเครื่องจำลองมาจากโปรแกรม VirtualBox 4.3.10 จากเครื่องฮาร์ดแวร์หลัก ที่มีทรัพยากรดังนี้

- CPU intel Core I5
- RAM 8 GB
- Hard disk 500 GB

และซึ่งจำลองฮาร์ดแวร์ในการทำเครื่องจำลอง ได้เป็นดังนี้

- CPU ขนาด 1 core/ Ram ขนาด 1024 Mb
- CPU ขนาด 2 core/ Ram ขนาด 1024 Mb
- CPU ขนาด 2 core/ Ram ขนาด 2048 Mb
- CPU ขนาด 4 core/ Ram ขนาด 4096 Mb

และเมื่อได้จำลองเครื่องทั้ง 4 เครื่องเรียบร้อยแล้ว ผู้วิจัยจึงนำ AaaS ที่ได้พัฒนาขึ้นมาติดตั้งทั้ง 4 เครื่องเพื่อเก็บระยะเวลาในการเรียก AaaS โดยเก็บจากข้อมูลของการเรียกใช้บริการในการเข้าถึงพร้อมกัน 10 ผู้ใช้งาน 100 ผู้ใช้งาน และ 1,000 ผู้ใช้งาน นำมาทดสอบจำนวน 1,000 รอบ และจึงนำผลของระยะเวลาที่ใช้จากทั้ง 4 เครื่องจำลองมาเปรียบเทียบหา ระยะเวลาการตอบสนองเฉลี่ย ค่าภาระงาน และวิเคราะห์ฮิสโตแกรม เพื่อดูแนวโน้มการแจกแจงความถี่ของระยะเวลาที่ใช้ในการประมวลผล

บทที่ 4
ผลการวิจัยและอภิปรายผล

ผลการศึกษา โดยจำแนกผลการศึกษาได้เป็นดังนี้

4.1. สภาพแวดล้อมที่ใช้การพัฒนาตัวจำลอง C-MTAS

4.2. ตัวระบบที่ถูกจำลองจากโมเดล C-MTAS

4.3. สรุปผลการทดสอบในกรณีทดสอบในกรณีปกติ

4.4. สรุปผลการทดสอบในกรณีทดสอบในกรณีของการเกิดการเรียกบริการแบบประสานงาน

4.5. สรุปผลการวิเคราะห์ในเชิงการเปรียบเทียบ เรื่องของการปรับขนาดของฮาร์ดแวร์ในการรองรับการทำงานของ C-MTAS

4.1. สภาพแวดล้อมที่ใช้ในการพัฒนาและทดสอบตัวจำลอง C-MTAS

4.1.1. ฮาร์ดแวร์ (Hardware)

ฮาร์ดแวร์ที่ใช้ในการพัฒนาตัวจำลอง C-MTAS ประกอบด้วย

(1) เครื่องคอมพิวเตอร์โน้ตบุ๊ค หน่วยประมวลผล Intel (R) Core (TM) i5-2450M CPU @ 2.5GHz

(2) หน่วยความจำหลัก (RAM) 8 กิกะไบต์

(3) ฮาร์ดดิส (Harddisk) 500 กิกะไบต์

4.1.1. ซอฟต์แวร์ (Software)

ซอฟต์แวร์ที่ใช้ในการพัฒนาตัวจำลอง C-MTAS ประกอบด้วย

(1) ระบบปฏิบัติการ (Operating System) ไมโครซอฟท์วินโดวส์ 8 โฟรเฟซชันแนล (Microsoft Windows 8 Professional 64 bit)

(2) ระบบจัดการฐานข้อมูล (Database management system) MySQL เวอร์ชั่น 5.0.67-community-nt

(3) พัฒนาตัวจำลอง C-MTAS ด้วยภาษาจาวา โดยใช้โปรแกรม Netbean IDE 8.0 โดยใช้ library

- JDK 1.7
- Sun XACML เวอร์ชั่น 1.2 เพื่อใช้ในการรันพื้นฐานของ XACML
- JGrapht-core เวอร์ชั่น 0.9.0 เพื่อใช้ในการตรวจสอบการเกิด Szwarcfiter Cycle Alogirthm

- Junit เพื่อใช้ในรันการทดสอบจากกรณีทดสอบที่จัดเตรียมไว้

(4) เครื่องมือที่ใช้ในการจัดการฐานข้อมูล MySQL Workbench เวอร์ชั่น 6.2 Community

(5) เว็บเซิร์ฟเวอร์ GlassFish Server 7.0 ในการรันเว็บเซอร์วิส

(6) เครื่องมือในการตรวจสอบการเชื่อมต่อเว็บเซอร์วิส Soap UI เวอร์ชั่น 5.0.0

(7) เครื่องมือในการตรวจสอบประสิทธิภาพของการรันเว็บเซอร์วิส Jmeter เวอร์ชั่น 2.9 r1437961

4.2. ตัวระบบที่ถูกจำลองจากโมเดล C-MTAS

หลังกระบวนการพัฒนาระบบจากโมเดล C-MTAS แล้ว ซึ่งมีองค์ประกอบตาม ภาพที่ 3.5 แล้วนั้น เมื่อมองกระบวนการทำงาน (input process output) สามารถแบ่งออกได้เป็น 3 ส่วนคือ

4.2.1. การส่งการขอร้องเพื่อตรวจสอบสิทธิ์การเข้าถึง (input)

จากภาพที่ 4.1 แสดงถึงองค์ประกอบในการขอร้องเพื่อใช้ในการตรวจสอบสิทธิ์ของการเข้าถึง ผ่าน AaaS โดยมีองค์ประกอบดังนี้

(1) ROLE_ID เพื่อบอกถึงบทบาทที่ขอร้อง

(2) TENANT_ID เพื่อบอกถึงการเช่าที่ขอร้อง

(3) USER_ID เพื่อบอกถึงผู้ใช้งานที่ร้องขอ

(4) PERMISSION_ID เพื่อบอกถึงสิทธิ์ที่ร้องขอ

โดย C-MTAS ที่ทำงานผ่าน AaaS นั้นจะทำการคำนวณ โดยใช้องค์ประกอบทั้ง 4 เพื่อตรวจสอบว่าผู้ใช้งานที่ร้องขอมานั้น มีสิทธิ์ในการทำงานตาม Permission หรือไม่

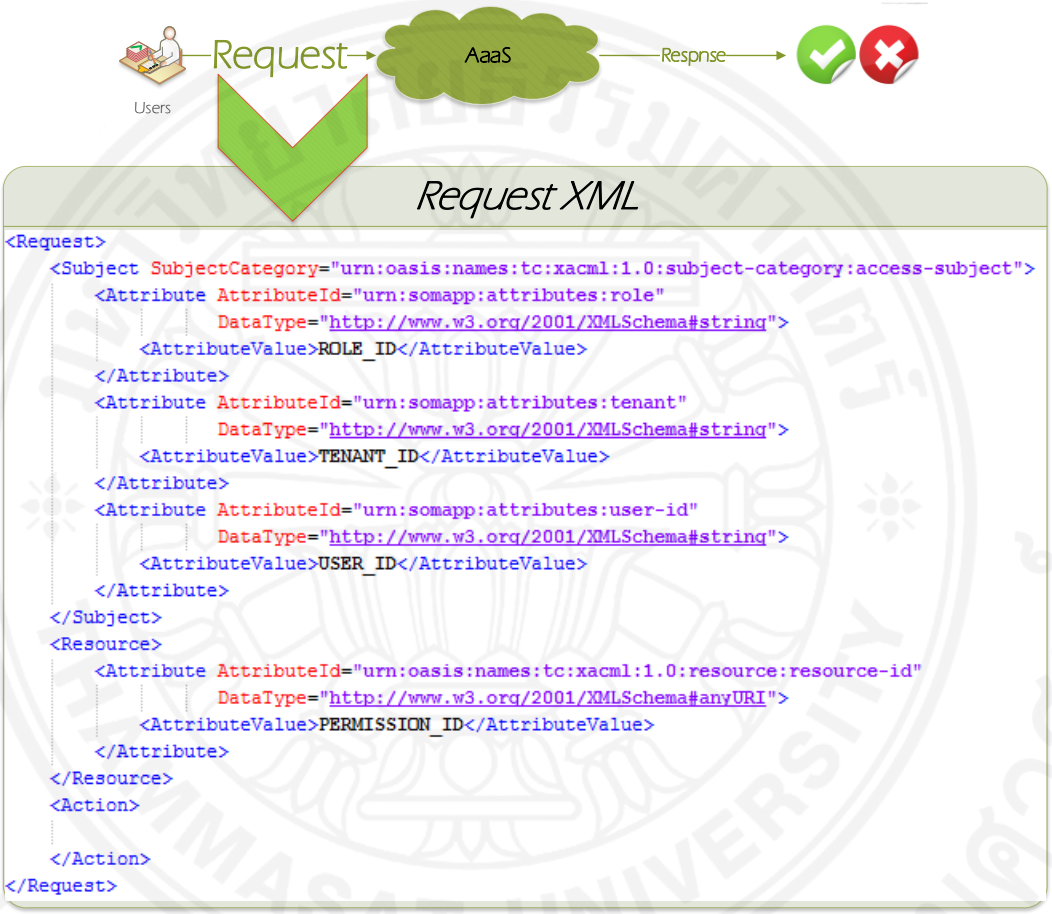
4.2.2. กระบวนการคำนวณการตรวจสอบสิทธิ์การเข้าถึง

จากภาพที่ 3.5 ได้แสดงถึงองค์ประกอบของโมเดล C-MTAS โดยประกอบด้วย 3 องค์ประกอบ โดยกระบวนการตรวจสอบสิทธิ์นั้นเป็นไปตามภาพที่ 3.4 ที่แสดงอัลกอริทึมการทำงาน โดยมีองค์ประกอบและรายละเอียด ดังนี้

1. **PEP (Policy Enforcement Point)** เป็นตัวกลางที่รับค่าพารามิเตอร์ในการนำเข้ามาตรวจสอบสิทธิ์การเข้าถึง

2. **ContextHandler** เป็นตัวที่รับส่งมาจาก PEP ในการตัดสินใจและตรวจสอบว่าสิทธิ์ที่ขอร้องเข้ามาเป็นห่วงโซ่การบริการแบบประสานงานหรือไม่ โดยนำมาตรวจสอบกับเซตที่กำหนดใน CCPR เซต โดยในภาพที่ 4.2 กล่าวได้คือ เป็นเซตของ ccprset โดย

กำหนดคุณสมบัติของรหัสผู้เช่า จากตัวอย่างคือ AuditApp ซึ่งมีเซตของสิทธิ์ในการเกิดห่วงโซ่การบริการแบบประสานงาน โดยเริ่มจาก previewReportYealy () สามารถไปใช้สิทธิ์ของ previewReportMonthlyAccount () โดยจากคุณสมบัติของ first คือเป็นจุดเริ่มต้นของต้นไม้ของ CCPRSET ดังกล่าว



ภาพที่ 4.1 องค์ประกอบ XML ในการร้องขอเพื่อใช้ในการตรวจสอบสิทธิ์ในระบบ C-MTAS

เมื่อสิทธิ์ที่ร้องขอเป็นห่วงโซ่ตามที่ได้กำหนดไว้ใน CCPRSET ตัวโมเดลดังกล่าว จะตอบสนองผลลัพธ์กับไปให้กับผู้ร้องขอเป็น “Permit” นั่นแสดงได้ว่าอนุญาตให้มีสิทธิ์ต่อไปได้

3. แต่หากสิทธิ์ไม่มีเกิดใน CCPRSET นั่นจึงส่งต่อให้กับ PDP ในการตัดสินใจสิทธิ์ที่ผู้ขอขอให้ตรวจสอบสิทธิ์มา โดย PDP จะตรวจสอบนโยบายที่ถูกกำหนดไว้ใน 3 ส่วน คือ Authorization Policy , Trust Policy และ Tenant Policy ตามภาพที่ 3.5

```xml
<?xml version="1.0"?>
<ccprsets>
    <ccprset tenant_id="AuditApp">
        <permsets>
            <permset first="true">
                <form>previewReportYearly()</form>
                <to>previewReportMonthlyAccount()</to>
            </permset>
            <permset>
                <form>previewReportMonthlyAccount()</form>
                <to>previewReportDailyAccount()</to>
            </permset>
            <permset>
                <form>previewReportDailyAccount()</form>
                <to>downloadFile()</to>
            </permset>
            <permset>
                <form>previewReportDailyAccount()</form>
                <to>previewFile()</to>
            </permset>
        </permsets>
    </ccprset>
</ccprsets>
```

ภาพที่ 4.2 ภาพตัวอย่างการกำหนดเซตของ CCPR โดยใช้ XML ในการกำหนด

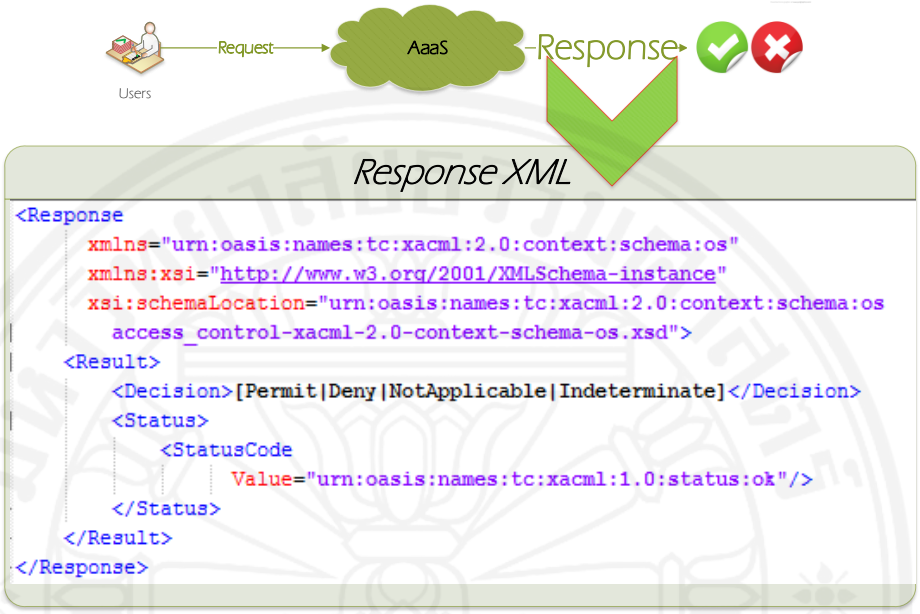
4.2.3. ผลลัพธ์ขอการร้องขอ

จากภาพที่ 3.6 แสดงถึงองค์ประกอบใน XML ที่เป็นผลลัพธ์หลังจากการตรวจสอบสิทธิ์ของการเข้าถึง ผ่าน AaaS โดยมีองค์ประกอบดังนี้

(1) Result เพื่อบอกถึงผลลัพธ์จากการคำนวณจากนโยบายที่ถูกกำหนดขึ้น โดยมี 4 ผลลัพธ์ ดังนี้

- Permit โดยเมื่อนำนโยบายที่ถูกกำหนดขึ้นสำหรับตรวจสอบ หากการตรวจสอบพบว่านโยบายดังกล่าวมีการกำหนดให้อนุญาต กับข้อมูลที่ร้องขอมา ระบบจึงถือว่า “อนุญาตให้เข้าถึง”
- Deny โดยเมื่อนำนโยบายที่ถูกกำหนดขึ้นสำหรับตรวจสอบ หากการตรวจสอบพบว่านโยบายดังกล่าวมีการกำหนดไม่ให้อนุญาต กับข้อมูลที่ร้องขอมา ระบบจึงถือว่า “ไม่อนุญาตให้เข้าถึง”
- Not Applicable โดยเมื่อนำนโยบายที่ถูกกำหนดขึ้นสำหรับตรวจสอบ หากการตรวจสอบพบว่านโยบายดังกล่าวไม่ชัดเจน พอเพียงที่จะตอบได้ว่าอนุญาตหรือไม่อนุญาต กับข้อมูลที่ร้องขอมา ระบบจึงถือว่า Not Applicable

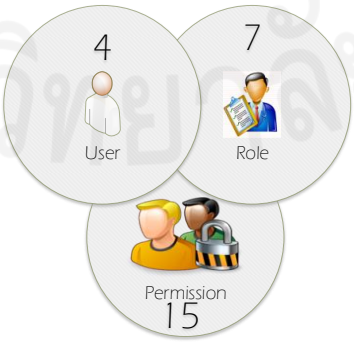
● Indeterminate เกิดข้อผิดพลาดระหว่างตรวจสอบนโยบาย ซึ่งไม่พบนโยบายที่ถูกอ้างอิง ระบบจึงถือว่าเกิดข้อผิดพลาด



ภาพที่ 4.3 องค์ประกอบ XML สำหรับผลลัพธ์ในการตรวจสอบสิทธิ์ในระบบ C-MTAS

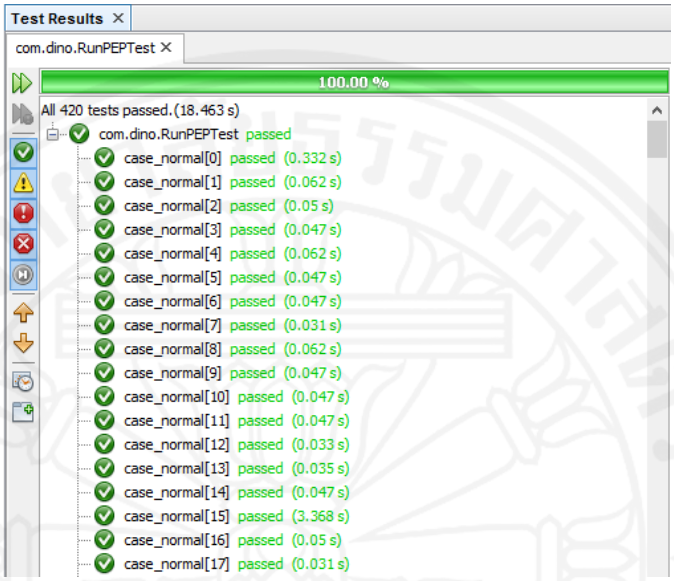
4.3. สรุปผลการทดสอบในกรณีปกติ

ในกรณีของการทดสอบในกรณีปกตินั้น ผู้ใช้ได้นำตัวอย่างที่กล่าวไว้ในบทที่ 3 ภาพที่ *3.8* ซึ่งได้ใช้การคำนวณกรณีทดสอบจากองค์ประกอบ 3 ส่วน คือ จำนวนผู้ใช้งาน 4 ผู้ใช้งาน จำนวนบทบาท 7 บทบาท และจำนวนสิทธิ์การเข้าถึง 15 สิทธิ์ มาใช้ในการหากรณีที่เป็นไปได้ทั้งหมด ซึ่งรวมแล้ว ได้เป็น 420 กรณีทดสอบ โดยกรณีทดสอบในชุดแรก ได้แสดงรายละเอียดในภาคผนวก ก. ในตารางที่ ก-1



ภาพที่ 4.4 แสดงองค์ประกอบในการหาจำนวนกรณีทดสอบในกรณีแบบปกติ

โดยการรันผลลัพธ์ ด้วยการใช้เครื่องมือในการทดสอบที่ชื่อว่า Junit นั้นพบว่ากรณีทดสอบจำนวน 420 กรณี ผลลัพท์ตรงตามผลลัพธ์ที่ต้องการทั้งหมด ดังนั้นจึงสรุปผลการทดสอบในกรณีปกติได้เป็น “มีความถูกต้อง” ทั้งหมด ตามภาพที่ 4.5



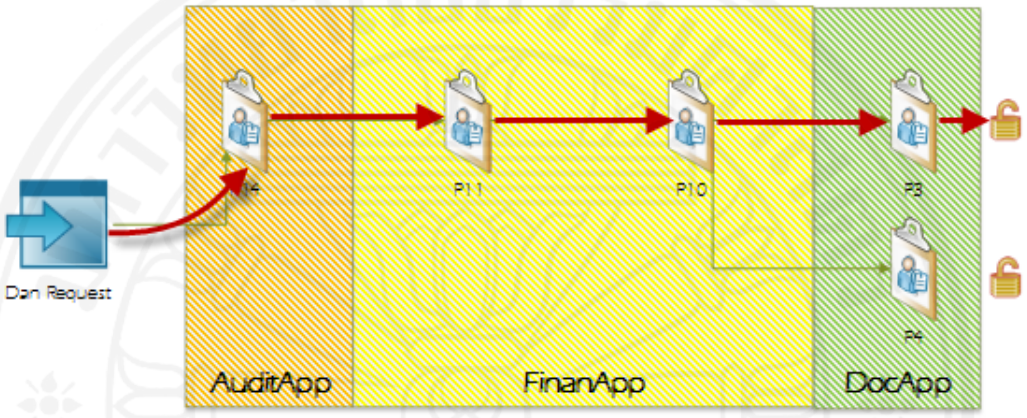
ภาพที่ 4.5 ผลลัพธ์การรันกรณีทดสอบแบบปกติ

4.4. สรุปผลการทดสอบในกรณีที่เกิดการเรียกบริการแบบห่วงโซ่การบริการแบบประสานงาน

ในกรณีของการทดสอบในกระบวนการเกิดการเรียกบริการแบบห่วงโซ่การบริการแบบประสานงานนั้น ผู้วิจัยได้นำกรณีที่เกิดขึ้นได้จาก

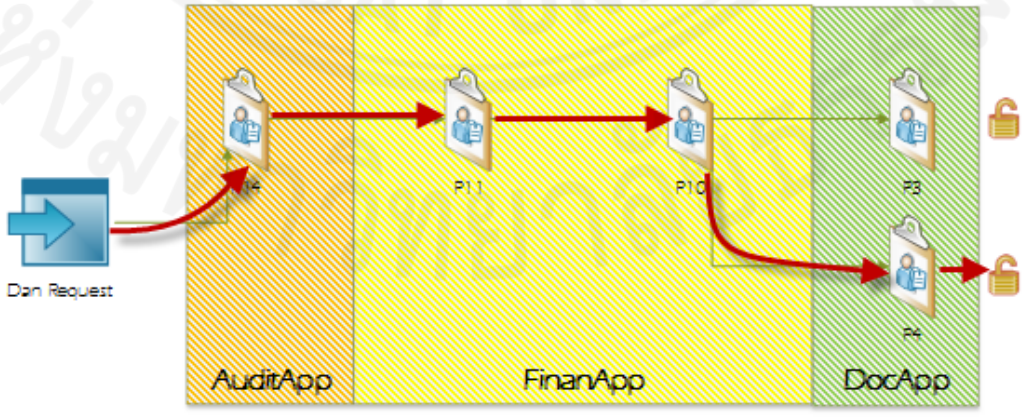
ภาพที่ 3.8 ซึ่งสามารถเกิดกรณีทดสอบของการเกิดการเรียกบริการแบบห่วงโซ่การบริการแบบประสานงานได้เป็น 7 กรณีทดสอบ ได้ดังนี้

4.3.1. การเกิดกรณีที่ผู้ใช้งาน Dan เรียกใช้บริการจากสิทธิ์ P14 : previewReportYearly () เพื่อเรียกใช้บริการจากสิทธิ์ P11 : previewReportMonthlyAccount () แล้วจึงเรียกใช้บริการจากสิทธิ์ P10 : previewReportDailyAccount () เพื่อเรียกใช้บริการในการดาวน์โหลดไฟล์ที่ถูกเก็บมาใช้งาน จากสิทธิ์ P3 : downloadFile () ซึ่งผลลัพธ์จากการเกิดห่วงโซ่ดังกล่าวเป็น Permit คือ Dan มีสิทธิ์ในการเข้าถึงในการดาวน์โหลดไฟล์ที่ถูกเก็บในการตรวจสอบรายงานผลประจำปีได้ ตามภาพที่ 4.6



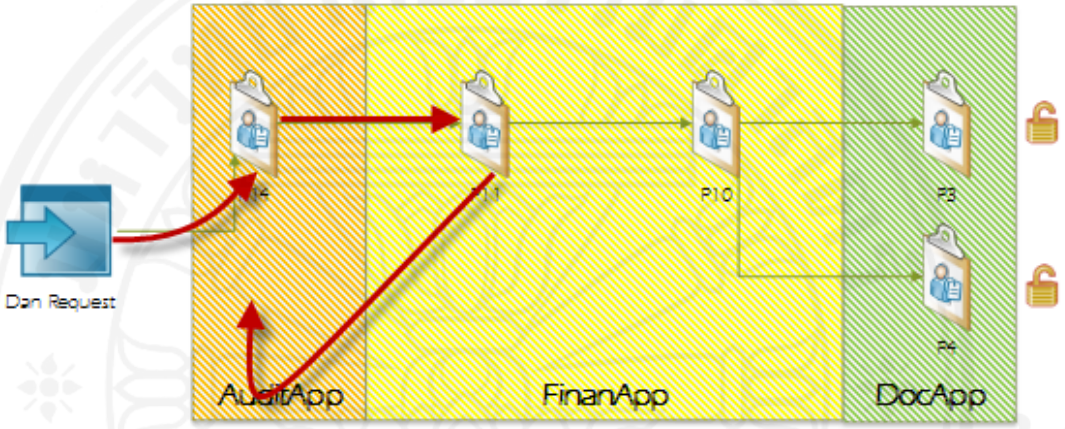
ภาพที่ 4.6 แสดงกรณีทดสอบที่ 4.3.1

4.3.2. การเกิดกรณีที่ผู้ใช้งาน Dan เรียกใช้บริการจากสิทธิ์ P14 : previewReportYearly () เพื่อเรียกใช้บริการจากสิทธิ์ P11 : previewReportMonthlyAccount () แล้วจึงเรียกใช้บริการจากสิทธิ์ P10 : previewReportDailyAccount () เพื่อเรียกใช้บริการในการดูไฟล์อย่างเดียว จากสิทธิ์ P4 : previewFile () ซึ่งผลลัพธ์จากการเกิดห่วงโซ่ดังกล่าวเป็น Permit คือ Dan มีสิทธิ์ในการเข้าถึงในการเรียกดูเพื่อตรวจสอบไฟล์ที่ถูกเก็บในการตรวจสอบรายงานผลประจำปีได้ตามภาพที่ 4.7



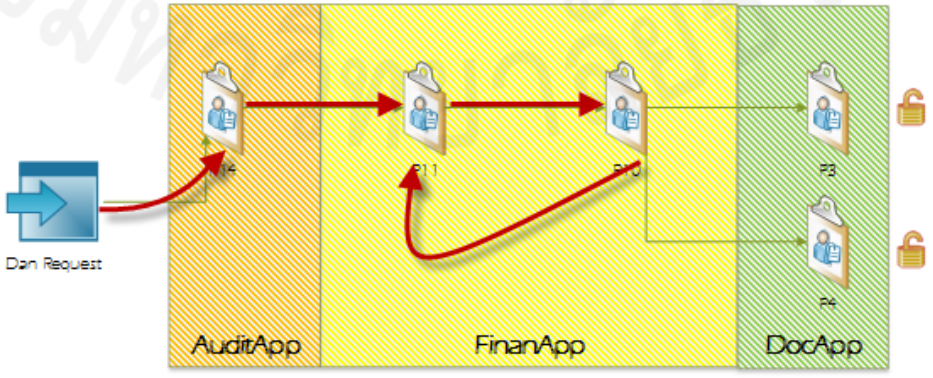
ภาพที่ 4.7 แสดงกรณีทดสอบที่ 4.3.2

4.3.3. การเกิดกรณีที่ผู้ใช้งาน Dan เรียกใช้บริการจากสิทธิ์ P14 : previewReportYearly () เพื่อเรียกใช้บริการจากสิทธิ์ P11 : previewReportMonthlyAccount () แต่ผู้ใช้ได้ดำเนินการกลับมาเรียกดูข้อมูลใหม่จากการใช้บริการจากสิทธิ์ P14 : previewReportYearly () อีกครั้ง ซึ่งเสมือนกับการเรียกห่วงโซ่การบริการแบบประสานงานเริ่มต้นใหม่อีกครั้ง ซึ่งผลลัพธ์จะไม่ถือว่าเป็นการเกิด cycle และผลลัพธ์ในการตรวจสอบสิทธิ์คือ Permit ตามภาพที่ 4.8



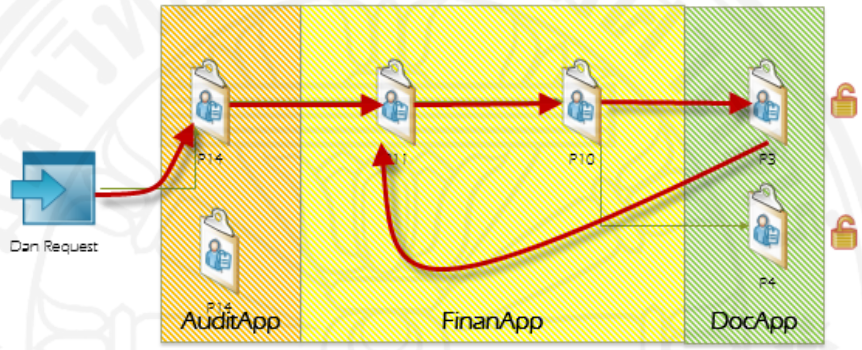
ภาพที่ 4.8 แสดงกรณีทดสอบที่ 4.3.3

4.3.4. การเกิดกรณีที่ผู้ใช้งาน Dan เรียกใช้บริการจากสิทธิ์ P14 : previewReportYearly () เพื่อเรียกใช้บริการจากสิทธิ์ P11 : previewReportMonthlyAccount () แล้วจึงเรียกใช้บริการจากสิทธิ์ P10 : previewReportDailyAccount () แต่กลับมาเรียกสิทธิ์ของจากสิทธิ์ P11 : previewReportMonthlyAccount () ซึ่งทำให้ขั้นตอนของการนำไปสู่การเกิด cyclic ได้ของการเรียกบริการแบบประสานงาน ซึ่งในเคสดังกล่าวอาจจะเกิดการเรียกต่อกันไม่รู้จบ ทำให้ผู้บริการจะเกิดปัญหาได้ ดังนั้นในงานวิจัยชิ้นนี้ เมื่อเกิด cycle ดังกล่าว จึงถือว่าเกิดผลลัพธ์เป็น Indeterminate ตามภาพที่ 4.9



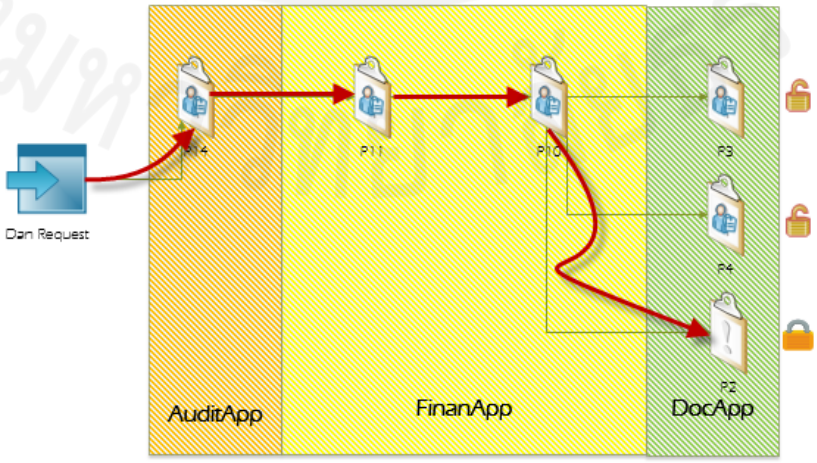
ภาพที่ 4.9 แสดงกรณีทดสอบที่ 4.3.4

4.3.5. การเกิดกรณีที่ผู้ใช้งาน Dan เรียกใช้บริการจากสิทธิ์ P14 : previewReportYearly () เพื่อเรียกใช้บริการจากสิทธิ์ P11 : previewReportMonthlyAccount () แล้วจึงเรียกใช้บริการจากสิทธิ์ P10 : previewReportDailyAccount () เพื่อเรียกใช้บริการในการดาวน์โหลดไฟล์ที่ถูกเก็บมาใช้งาน จากสิทธิ์ P3 : downloadFile () และผู้ใช้งานกลับย้อนไปใช้บริการจากสิทธิ์ P11 : previewReportMonthlyAccount () ซึ่งในเคสกรณีดังกล่าวถือว่าเกิด cyclic เช่นกัน ดังนั้นระบบจึงได้ผลลัพธ์เป็น Indeterminate ตามภาพที่ 4.10



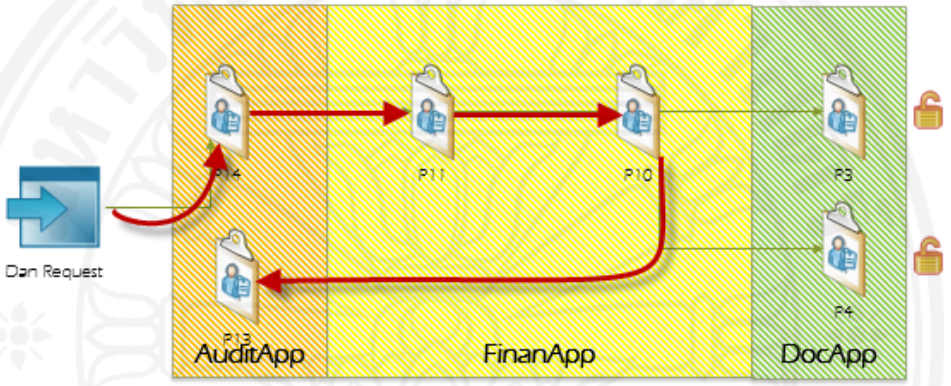
ภาพที่ 4.10 แสดงกรณีทดสอบ 4.3.5

4.3.6. การเกิดกรณีที่ผู้ใช้งาน Dan เรียกใช้บริการจากสิทธิ์ P14 : previewReportYearly () เพื่อเรียกใช้บริการจากสิทธิ์ P11 : previewReportMonthlyAccount () แล้วจึงเรียกใช้บริการจากสิทธิ์ P10 : previewReportDailyAccount () แต่กลับมาเรียกสิทธิ์ของจากสิทธิ์ P2 : uploadFile () ซึ่งไม่ได้มีอยู่ในการกำหนดการเกิดในรูปแบบของห่วงโซ่การบริการแบบประสานงาน จึงถือว่าเป็นกระบวนการตรวจสอบในรูปแบบปกติ และผู้ใช้งาน Dan ก็ไม่ได้ถูกกำหนดในสิทธิ์การอัพโหลดไฟล์ได้จากเจ้าของการเช่า DocApp ซึ่งผลลัพธ์ของการร้องขอให้ห่วงโซ่การบริการแบบประสานงานดังกล่าวจึงได้ผลเป็น Not Applicable ตามภาพที่ 4.11



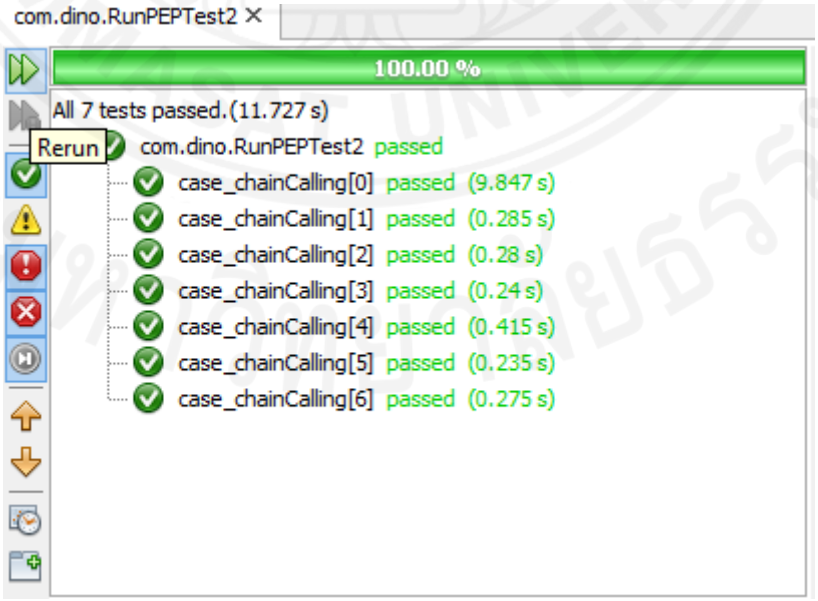
ภาพที่ 4.11 แสดงกรณีทดสอบที่ 3.4.6

4.3.7. การเกิดกรณีที่ผู้ใช้งาน Dan เรียกใช้บริการจากสิทธิ์ P14 : previewReportYearly () เพื่อเรียกใช้บริการจากสิทธิ์ P11 : previewReportMonthlyAccount () แล้วจึงเรียกใช้บริการจากสิทธิ์ P10 : previewReportDailyAccount () และกลับไปเรียกการใช้บริการจากสิทธิ์ P13: verifyReport () ซึ่งไม่ได้เกิดเป็นห่วงโซ่การบริการแบบประสานงานแล้ว แต่เป็นการเริ่มใช้บริการใหม่ ซึ่งในกรณีนี้จะถือได้ว่าเป็นตรวจสอบสิทธิ์ในรูปแบบปกติ เช่นเดียวกับ 4.3.6 ตามภาพที่ 4.12



ภาพที่ 4.12 แสดงกรณีทดสอบที่ 4.3.7

โดยการรันผลลัพธ์ ผู้วิจัยได้ใช้ Junit เพื่อทดสอบกรณีทดสอบทั้ง 7 กรณีทดสอบดังกล่าว และพบว่าทั้ง 7 กรณีตรงตามผลลัพธ์ที่ต้องการทั้งหมด ดังนั้นจึงสรุปผลการทดสอบในกรณีที่เป็นห่วงโซ่การบริการแบบประสานงานเป็น “ถูกต้อง” ทั้งหมด ตามภาพที่ 4.13



ภาพที่ 4.13 ผลลัพธ์การรันกรณีทดสอบการเกิดห่วงโซ่การบริการแบบประสานงาน

4.5. สรุปผลการวิเคราะห์ในเชิงการเปรียบเทียบ เรื่องของการปรับขนาดของฮาร์ดแวร์ในการรองรับการทำงานของ C-MTAS

จากการวิเคราะห์ในเชิงการเปรียบเทียบ เรื่องของการปรับขนาดของฮาร์ดแวร์ในการรองรับทำการจากแบบจำลอง C-MTAS นั้น ผู้วิจัยได้จำลอง C-MTAS ขึ้นทำงานเป็นเว็บเซอร์วิสไว้ใน Virtual Machine (VM) ซึ่งผู้วิจัยได้ใช้โปรแกรมจำลอง VM ของ VirtualBox 4.3.10 จากเครื่องฮาร์ดแวร์หลัก ที่มีทรัพยากรดังนี้

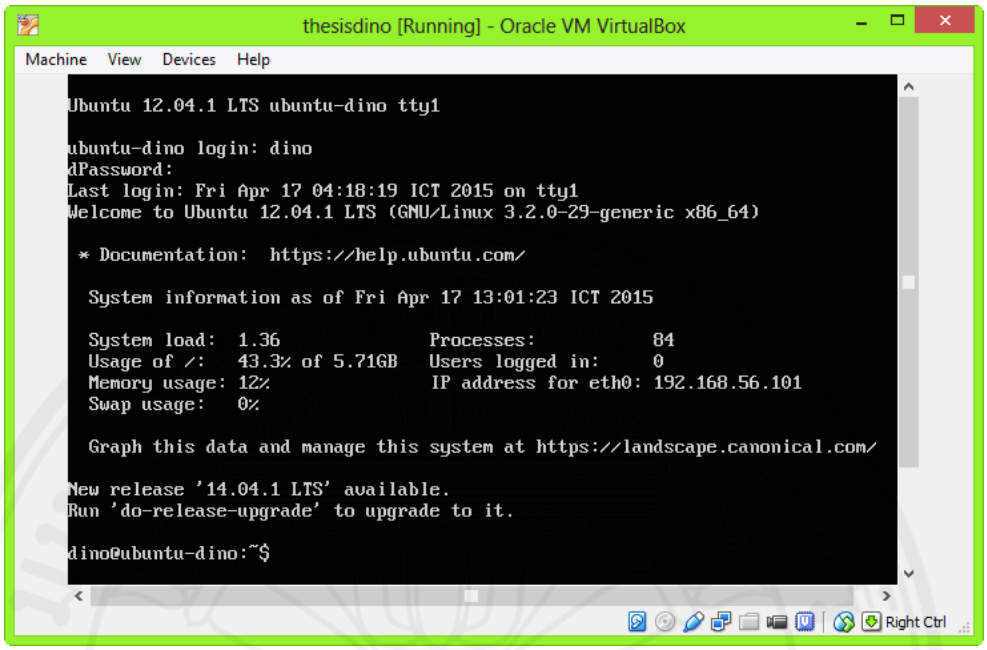
- CPU intel Core I5
- RAM 8 GB
- Hard disk 500 GB

และจำลองฮาร์ดแวร์ในการทำเครื่องจำลอง ที่มีสภาพแวดล้อมของ CPU และ RAM แตกต่างกัน เพื่อทดสอบประสิทธิ์การรัน ซึ่งสภาพแวดล้อมสามารถแบ่งออกได้เป็นดังนี้

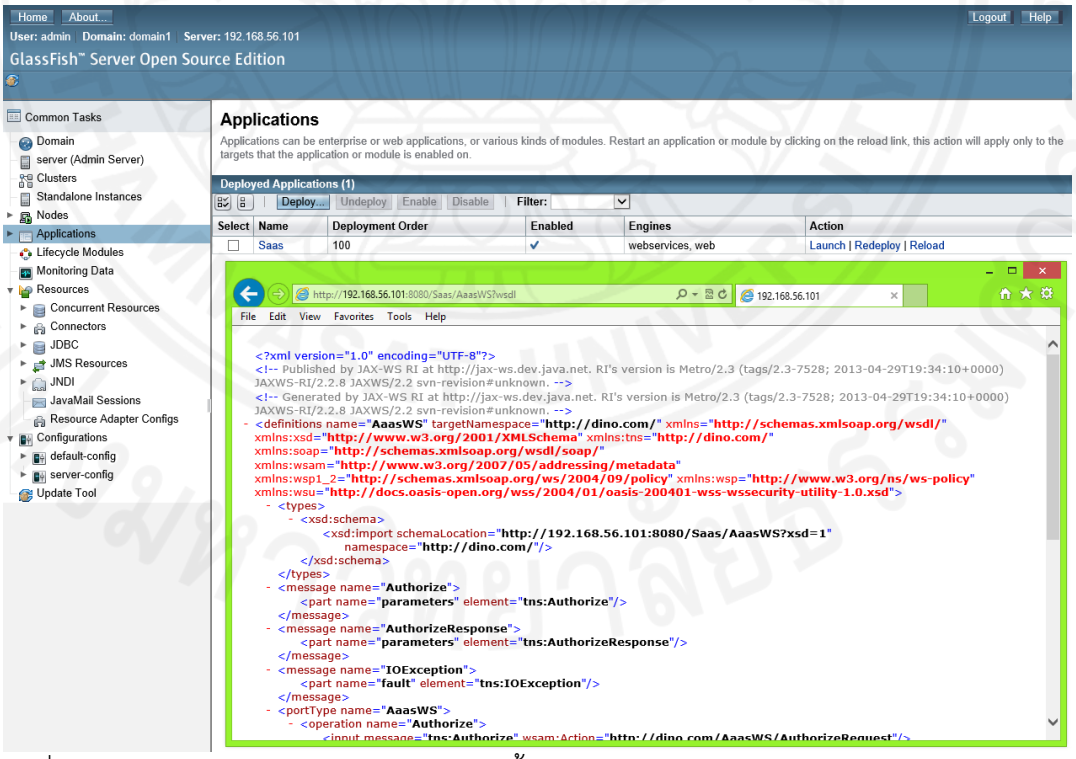
- CPU ขนาด 1 core/ Ram ขนาด 1024 Mb
- CPU ขนาด 2 core/ Ram ขนาด 1024 Mb
- CPU ขนาด 2 core/ Ram ขนาด 2048 Mb
- CPU ขนาด 4 core/ Ram ขนาด 4096 Mb

โดยเครื่องจำลองที่จำลองขึ้นมานั้น ผู้วิจัยได้ติดตั้งตั้งโปรแกรม เพื่อทดสอบการทำงานของ C-MTAS โดยประกอบด้วย

- ระบบปฏิบัติการเป็น Ubuntu Server 12.04.1 LTS
- เว็บเซิร์ฟเวอร์ Glassfish 4.0
- ระบบฐานข้อมูล MySQL 5.5



ภาพที่ 4.14 ตัวอย่างหน้าจอของการรันโปรแกรม VitualBox ในการจำลอง



ภาพที่ 4.15 ตัวอย่างหน้าจอ GlassFish โดยติดตั้ง AssS

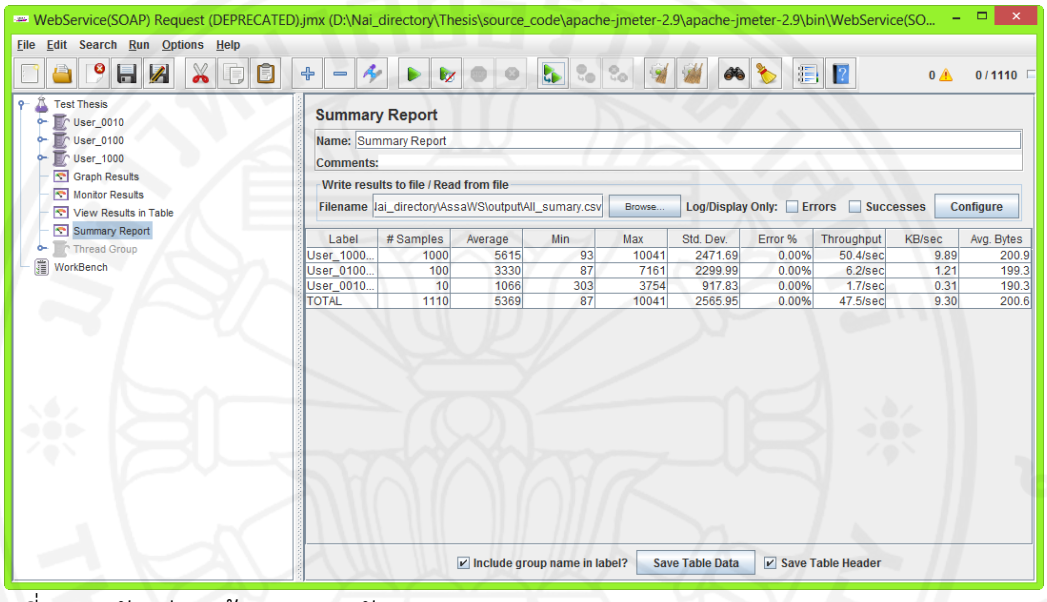
และได้ใช้เครื่องมือ jMeter 2.9 เพื่อใช้ในการตรวจสอบประสิทธิภาพการทำงานในสภาพแวดล้อมที่แตกต่างกันไปตามเครื่องจำลองทั้ง 4 กรณี

ตารางที่ 4.1

ผลลัพธ์จากการรันทดสอบเชิงเปรียบเทียบการทำงานของ C-MTAS ตามสภาพแวดล้อมเครื่องจำลองที่แตกต่างกัน

Hardware	Sample	Concurrent user	Average (ms)	Min (ms)	Max (ms)	SD (ms)	ERROR (%)
1 CPU, 1024 Mb	10000	10	62.67	16	6000	34.21	0.000
1 CPU, 1024 Mb	100000	100	42.60	7	3131	211.40	0.000
1 CPU, 1024 Mb	1000000	1000	144.62	7	20496	2551.76	0.000
2 CPU, 1024 Mb	10000	10	59.05	15	5048	30.50	0.000
2 CPU, 1024 Mb	100000	100	34.22	7	1991	173.64	0.000
2 CPU, 1024 Mb	1000000	1000	51.12	8	7621	1706.58	0.000
2 CPU, 2048 Mb	10000	10	41.47	10	3211	35.26	0.000
2 CPU, 2048 Mb	100000	100	32.98	7	6656	106.25	0.000
2 CPU, 2048 Mb	1000000	1000	48.41	7	11788	1295.83	0.000
4 CPU, 4096 Mb	10000	10	40.1581	13	455	28.98	0.000
4 CPU, 4096 Mb	100000	100	29.12	8	930	137.65	0.000
4 CPU, 4096 Mb	1000000	1000	36.08	7	7945	1170.7	0.000

ซึ่งผลลัพธ์ในการทดสอบในการรันจากเครื่องมือ jMeter 2.9 โดยได้ทำการรันตามการเข้าใช้งานพร้อมกัน (Concurrent user) ที่แตกต่างกัน 3 กรณี คือ 10 ผู้ใช้งาน 100 ผู้ใช้งาน และ 1000 ผู้ใช้งาน โดยในแต่ละกรณีที่รันจำนวน 1,000 รอบ ซึ่งตัวอย่างของผลการรันได้เป็นตามตารางที่ 4.1 โดยสามารถจำแนกสรุปผลได้จาก 3 ปัจจัย ดังนี้



ภาพที่ 4.16 ตัวอย่างหน้าจอผลการรัน jMeter

4.5.1. ค่าเวลาการตอบสนองเฉลี่ย (Average Response Time)

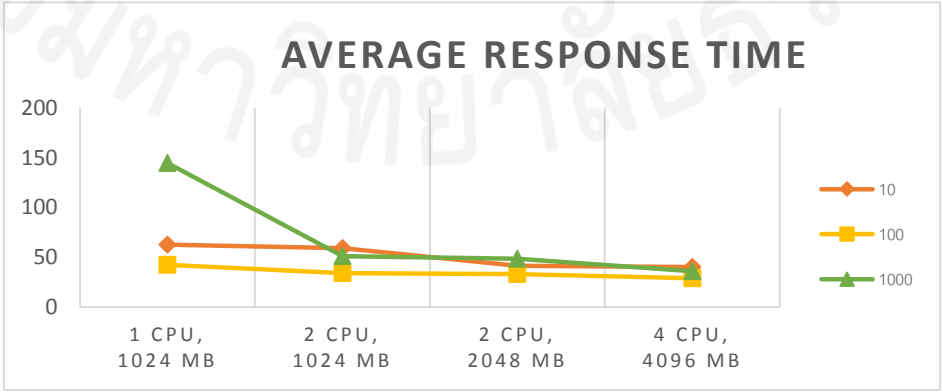
ผลจากการรันเปรียบเทียบประสิทธิภาพ ในเรื่องของค่าระยะเวลาตอบสนองเฉลี่ย นั้นจากสามารถสรุปได้อย่างชัดเจนว่าขนาดของฮาร์ดแวร์เมื่อมีการปรับขนาดของ CPU และ RAM ที่เพิ่มขึ้น เมื่อเปรียบเทียบจากฮาร์ดแวร์ชุดที่ 1 กับชุดที่ 2 ของจำนวนผู้เข้าใช้พร้อมกัน 1000 ผู้ใช้งาน พบว่ามีแนวโน้มที่ลดลงมาถึง คือ 63.9% ตามตารางที่ 4.2 และสำหรับในส่วนอื่น ๆ ก็ยังพบว่าแนวโน้มของเส้นทั้ง 3 ของ 10 100 และ 1000 ผู้ใช้งานพร้อมกันมีลักษณะลดลงเช่นเดียวกัน

และเมื่อข้อมูลของการทดสอบ 1000 ผู้ใช้งานพร้อมกัน มาวิเคราะห์ค่าเบี่ยงเบนมาตราฐานของฮาร์ดแวร์ชุด ก. ข. ค. และ ง. ตามลำดับได้ค่าดังนี้ 2,551.76 1,706.58 1,295.8 และ 1,170.7 ซึ่งพบว่าค่าเบี่ยงเบนมาตราฐานมีค่าที่ลดลง แสดงให้เห็นว่าเมื่อเพิ่มขนาดของฮาร์ดแวร์ขึ้น การกระจายของข้อมูลจะน้อยลง

ตารางที่ 4.2

ข้อมูลผลลัพธ์จากการรันเปรียบเทียบประสิทธิภาพการปรับขนาดของฮาร์ดแวร์ โดยแสดงค่าระยะเวลาการตอบสนองเฉลี่ย

Sum of average (ms)	Concurrent User		
Row Labels	10	100	1000
1 CPU, 0512 Mb	62.67	42.60	144.62
1 CPU, 1024 Mb	59.05	34.22	51.12
2 CPU, 2048 Mb	41.47	32.98	48.41
4 CPU, 4096 Mb	40.16	29.16	36.08



ภาพที่ 4.17 กราฟแสดงระยะเวลาการตอบสนองเฉลี่ย ในแต่ละสภาพแวดล้อมของฮาร์ดแวร์ที่แตกต่างกัน

4.5.2. ค่าภาระการทำงานบนเซิร์ฟเวอร์ (Througtput)

ค่าภาระการทำงาน แสดงให้เห็นถึงความสามารถในการรับโหลดบนเครื่องประมวลผลได้ ซึ่งหากค่ามีแนวโน้มที่สูงสุด แสดงให้เห็นว่าประสิทธิภาพในการรับการโหลดข้อมูลจะมากไปด้วย ตามเครื่องมือ jMeter ได้คำนวณค่าภาระการทำงานบนเซิร์ฟเวอร์ ได้เป็นดังนี้

Throughput = (number of requests) / (total time)

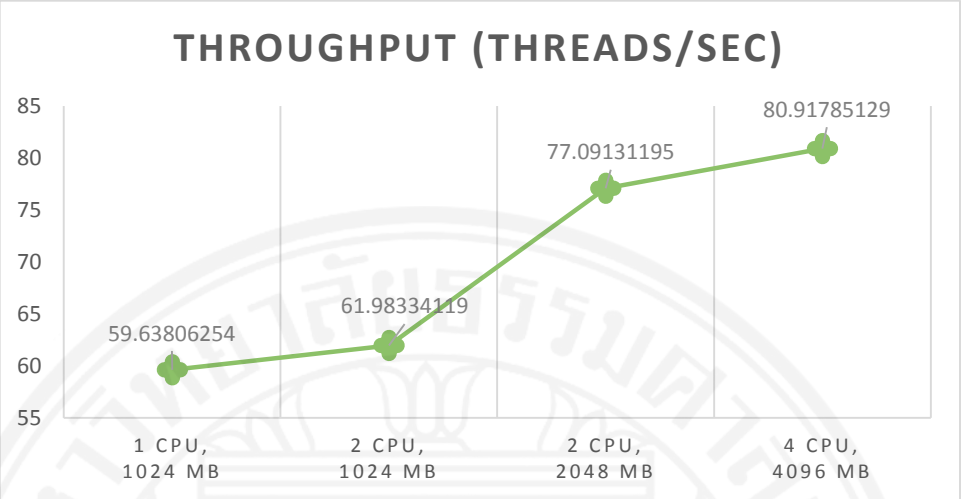
โดยผู้วิจัยได้เลือกทดลองจากจำนวนผู้ใช้งานเข้าใช้พร้อมกัน 1,000 ผู้ใช้งานเป็นตัวแปรต้น เนื่องจากในสภาพแวดล้อมการทำงานบนคลาวน์ สำหรับการรองรับประสิทธิภาพในการตรวจสอบการเข้าถึงผู้ใช้งานพร้อมกัน 10 ผู้ใช้งาน และ 100 ผู้ใช้งาน อาจจะไม่น้อยไป ดังนั้นผู้วิจัยจึงเลือกจำนวนผู้ใช้งานเข้าใช้พร้อมกัน 1,000 ผู้ใช้งานมาเป็นตัวทดสอบดังกล่าว และวิธีการตรวจสอบผู้วิจัยได้รันผลลัพธ์ โดยใช้เวลา 5 นาทีในแต่ละฮาร์ดแวร์ และจับค่าภาระงาน ซึ่งผลของภาระงานเป็นไปตามตารางที่ 4.3 ซึ่งชี้ให้เห็นได้อย่างชัดเจนว่าขนาดของฮาร์ดแวร์เมื่อมีการปรับขนาดของ CPU และ RAM ที่เพิ่มขึ้น โดยสามารถดูค่าภาระการทำงานก็จะเพิ่มขึ้นด้วยเช่นกัน

ซึ่งสามารถสรุปผลได้ว่า สำหรับฮาร์ดแวร์ชุดแรกค่าภาระการทำงานอยู่ที่ 59.6 รอบต่อวินาที ชุดฮาร์ดแวร์ที่สองค่าภาระการทำงานอยู่ที่ 62 รอบต่อวินาที ชุดฮาร์ดแวร์ที่สามค่าภาระการทำงานอยู่ที่ 77.1 รอบต่อวินาที และสำหรับฮาร์ดแวร์ชุดที่สี่ค่าภาระการทำงานอยู่ที่ 80.9 รอบต่อวินาที ซึ่งดูแนวโน้มของค่าจากฮาร์ดแวร์ทั้ง 4 ชุด มีค่าที่สูงขึ้นด้วย นั่นแสดงได้ว่า เมื่อฮาร์ดแวร์มีการปรับขนาดที่ดีขึ้นของจำนวน CPU รวมทั้ง RAM ทำให้ค่าภาระการทำงานของ SaaS มีค่าที่สูงขึ้นด้วยเช่นกัน

ตารางที่ 4.3

ข้อมูลผลลัพธ์จากการรันเปรียบเทียบประสิทธิภาพการปรับขนาดของฮาร์ดแวร์ โดยแสดงค่าภาระการทำงาน

Row Labels	Samples	Througphut (Threads/sec)
1 CPU, 1024 Mb	19,071	59.6
2 CPU, 1024 Mb	19,720	62.0
2 CPU, 2048 Mb	24,216	77.1
4 CPU, 4096 Mb	88,383	80.9



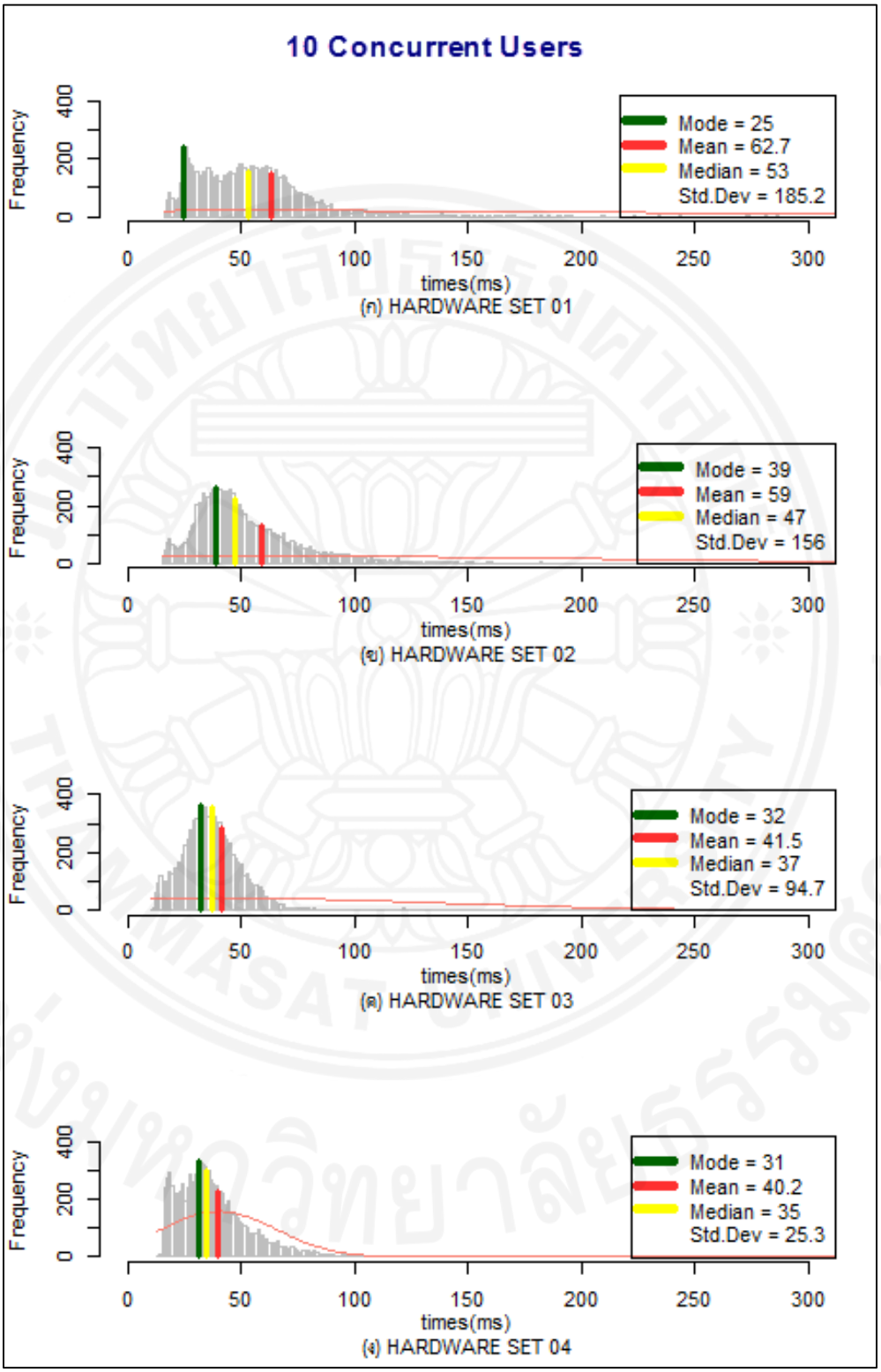
ภาพที่ 4.18 กราฟแสดงภาระการทำงาน ในแต่ละสภาพแวดล้อมของฮาร์ดแวร์ที่แตกต่างกัน

4.5.3. กราฟฮิสโตแกรม โดยนำระยะเวลาที่ใช้ในการประมวลผลมาแสดงผล (Histrogram)

กราฟฮิสโตแกรม โดยผู้วิจัยได้ทำการเปรียบเทียบนำกราฟฮิสโตแกรมมาแสดงการแจกแจงความถี่ของระยะเวลาในการตอบสนองโดยแบ่งออกเป็น 3 ส่วนคือ เปรียบเทียบระหว่าง 10 ผู้ใช้งานพร้อมกัน เปรียบเทียบระหว่าง 100 ผู้ใช้งานพร้อมกัน และเปรียบเทียบระหว่าง 1000 ผู้ใช้งานพร้อมกัน ในแต่ละฮาร์ดแวร์ ดังนี้

4.5.3.1. กราฟแสดงความถี่ของระยะเวลาที่ใช้ สำหรับจำนวนผู้ใช้งาน 10 ผู้ใช้งานพร้อมกัน

จากภาพที่ 4.19 หลังจากได้ตัดจำนวนเวลาที่ตอบสนองที่มากกว่า 300 ms ออกซึ่งแสดงให้เห็นว่าจากภาพที่ 4.19 (ก) กราฟมีลักษณะที่มีการกระจายตัวมาก เมื่อมาเปรียบเทียบกับกับภาพที่ 4.19 (ข) และการกระจายตัวของภาพที่ 4.19 (ค) เมื่อเปรียบเทียบกับภาพที่ 4.19 (ข) ก็มีการลดลงดังจะเห็นได้จากภาพที่ 4.19 (ค) นั้นช่วงระยะเวลาตอบสนอง 50-100 ms มีความถี่น้อยกว่ามาก และข้อมูลมียังการกระจายตัวในช่วง 0-50 ms และเมื่อนำภาพที่ 4.19 (ค) มาเปรียบเทียบกับภาพที่ 4.19 (ง) จะเห็นได้ว่าภาพที่ 4.19 (ง) นั้นลักษณะของเส้นโค้งแห่งความถี่ที่มีความสูงที่ชัดเจนขึ้นในช่วง 40-45 ms ดังนั้นจึงสรุปได้ว่าเมื่อมีการปรับขนาดของฮาร์ดแวร์ทั้ง CPU และ RAM จึงทำให้ประสิทธิภาพของระยะเวลาการตอบสนองที่ดีขึ้น



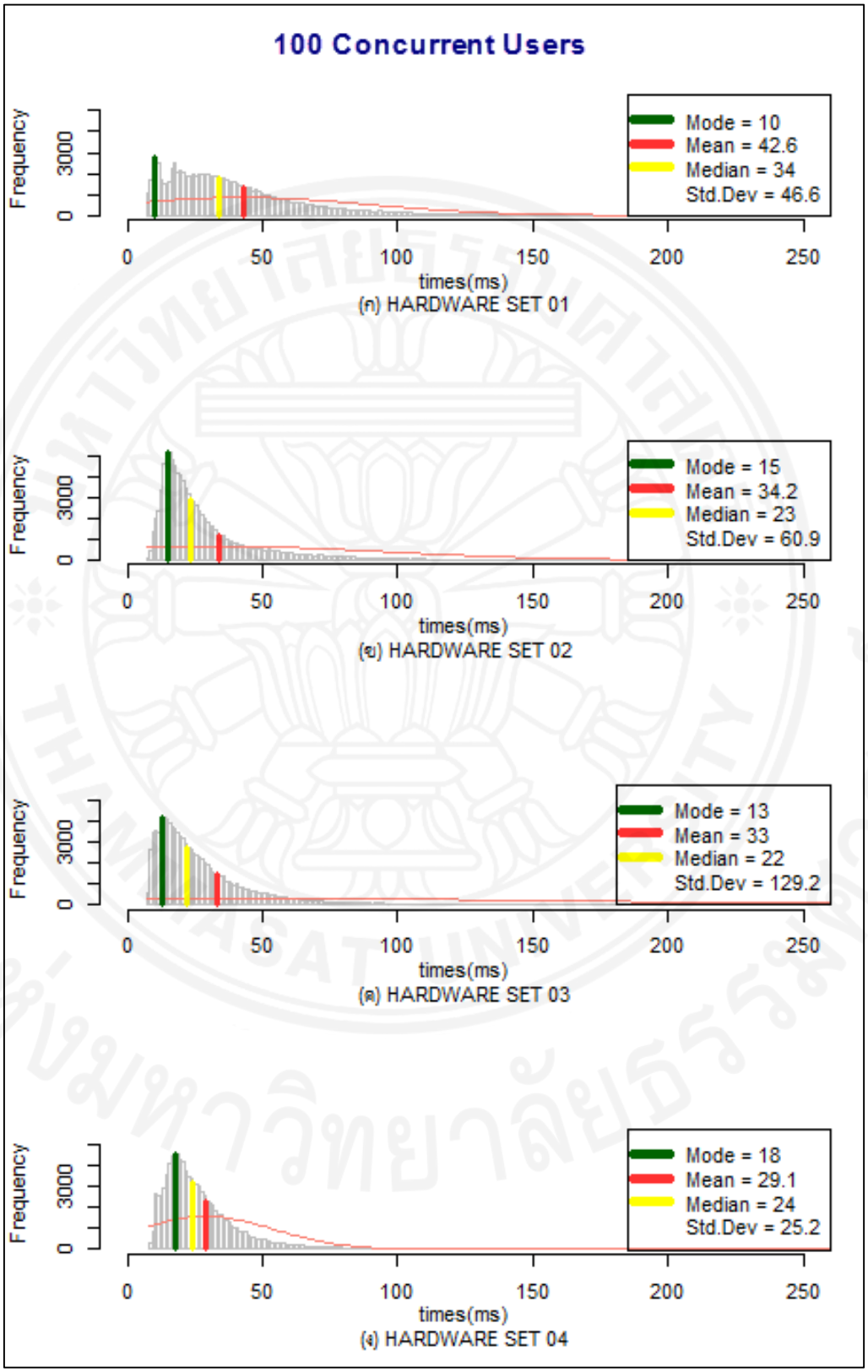
ภาพที่ 4.19 กราฟฮิสโตแกรมแสดงจำนวนการแจกแจงของระยะเวลา ที่ใช้ในการตอบสนองในกรณีของผู้ใช้งาน 10 ผู้ใช้งานพร้อมกัน โดยแยก 4 ชุดของฮาร์ดแวร์

4.5.5.2. กราฟแสดงความถี่ของระยะเวลาที่ใช้ สำหรับจำนวนผู้ใช้งาน 100 ผู้ใช้งานพร้อมกัน

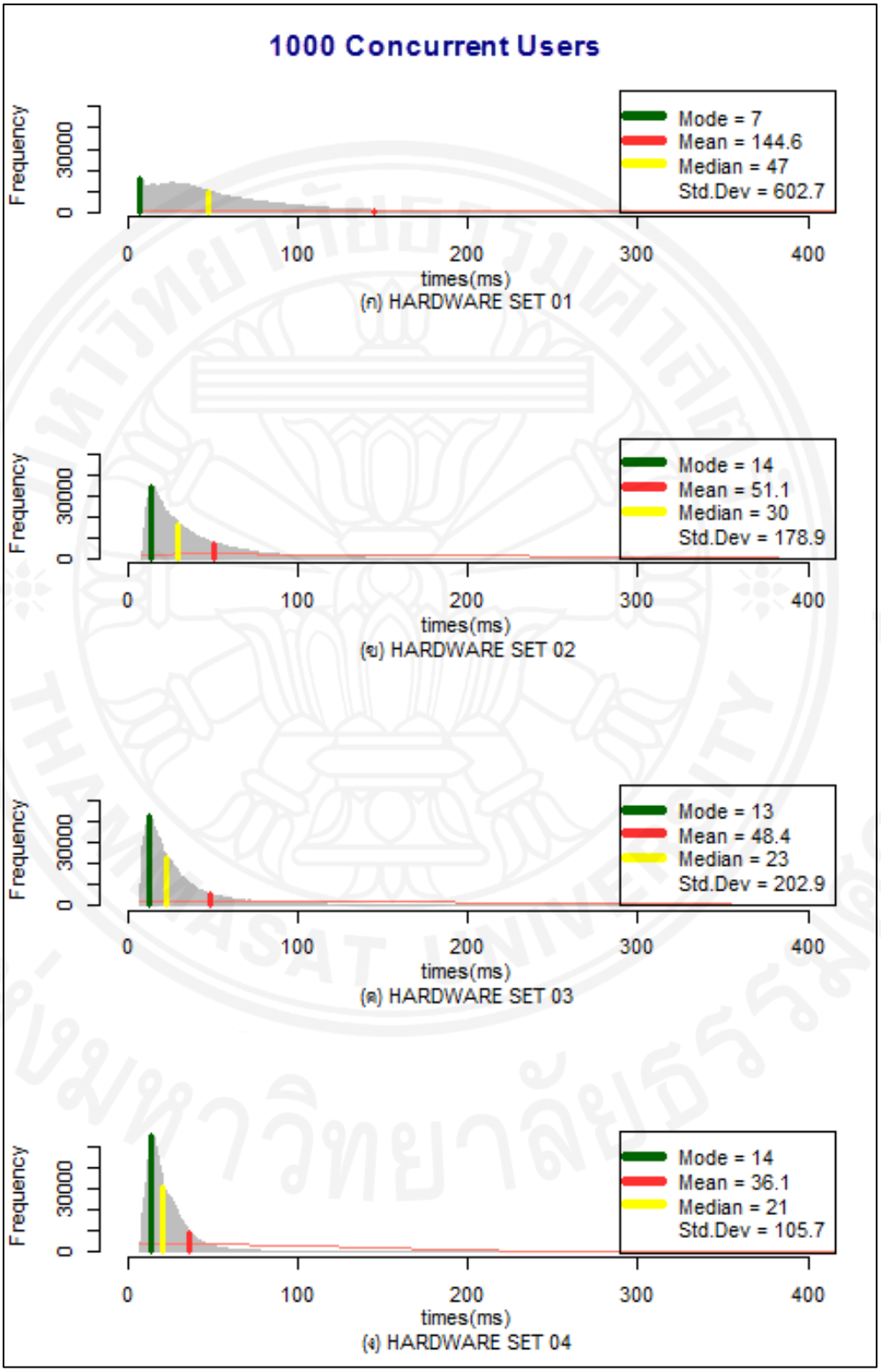
จากภาพที่ 4.20 หลังจากได้ตัดจำนวนเวลาที่ตอบสนองที่มากกว่า 250 ms ออกซึ่งแสดงให้เห็นว่าจากภาพที่ 4.20 (ก) เมื่อเปรียบเทียบกับภาพที่ 4.20 (ข) ข้อมูลเกาะกลุ่มในช่วง 15-25 ms ซึ่งถือว่ามีการกระจายตัวของข้อมูลที่ต่ำกว่าภาพที่ 4.20 (ก) เมื่อนำภาพที่ 4.20 (ข) มาเปรียบเทียบกับภาพที่ 4.20 (ค) มีลักษณะที่ไม่แตกต่างกันมากขึ้น ดังจะเห็นได้ว่าจากค่าฐานนิยม มัธยมฐาน รวมทั้งค่าเฉลี่ยเลขคณิตในภาพที่ 4.20 (ค) น้อยกว่าภาพที่ 4.20 (ง) เล็กน้อย และเมื่อนำภาพที่ 4.20 (ค) มาเปรียบเทียบกับภาพที่ 4.20 (ง) จะเห็นได้ว่าภาพที่ 4.20 (ง) นั้นลักษณะของเส้นโค้งแห่งความถี่ที่มีความสูงที่ชัดเจนขึ้นในช่วง 40-45 ms ดังนั้นจึงสรุปได้ว่าเมื่อมีการปรับขนาดของฮาร์ดแวร์ทั้ง CPU และ RAM จึงทำให้ประสิทธิภาพของระยะเวลาการตอบสนองที่ดีขึ้น และรวมทั้งค่าของส่วนเบี่ยงเบนมาตรฐานก็ยังพบว่าที่มีแนวน้อมที่น้อยลงอย่างมากจาก 129.2 ลงมาเป็น 25.2 ด้วย

4.5.5.3. กราฟแสดงความถี่ของระยะเวลาที่ใช้ สำหรับจำนวนผู้ใช้งาน 1000 ผู้ใช้งานพร้อมกัน

จากภาพที่ 4.21 หลังจากได้ตัดจำนวนเวลาที่ตอบสนองที่มากกว่า 400 ms ออกซึ่งแสดงให้เห็นว่าจากภาพที่ 4.21 (ก) พบว่าระยะเวลาการตอบสนองมีการกระจายตัวค่อยข้างมาก ซึ่งดูได้จากค่า SD ซึ่งมีค่า 602.7 และเมื่อมาเปรียบเทียบกับภาพที่ 4.21 (ข) มีการกระจายตัวลดลง รวมทั้งค่าเฉลี่ยเลขคณิตมีการลดลงอย่างมากจาก 144.6 เป็น 51.5 และเมื่อเปรียบเทียบจากภาพที่ 4.21 (ข) กับภาพที่ 4.21 (ค) พบได้ว่าลักษณะกราฟคล้ายกัน คือกราฟมีลักษณะของการเบ้ขวา แต่ภาพที่ 4.21 (ค) พบว่าความถี่ของฐานนิยมมากกว่า 42,000 ส่วนภาพที่ 4.21 (ข) พบว่าความถี่ของฐานนิยมอยู่ที่ 35,000 รวมทั้งค่าเฉลี่ยเลขคณิตก็ลดลงจาก 51.1 เป็น 48.4 และจากภาพที่ 4.21 (ค) เมื่อเปรียบเทียบกับภาพที่ 4.21 (ง) พบว่าค่าเฉลี่ยเลขคณิตก็ลดลงจาก 48.4 เป็น 36.1 ซึ่งแสดงให้เห็นว่าเมื่อมีการปรับขนาดของฮาร์ดแวร์ทั้ง CPU และ RAM จึงทำให้ประสิทธิภาพของระยะเวลาการตอบสนองที่ดีขึ้น



ภาพที่ 4.20 กราฟฮิสโตแกรมแสดงจำนวนการแจกแจงของระยะเวลา ที่ใช้ในการตอบสนองในกรณีของผู้ใช้งาน 100 ผู้ใช้งานพร้อมกัน โดยแยก 4 ชุดของฮาร์ดแวร์



ภาพที่ 4.21 กราฟฮิสโตแกรมแสดงจำนวนการแจกแจงของระยะเวลาที่ใช้ในการตอบสนองในกรณีของผู้ใช้งาน 1000 ผู้ใช้งานพร้อมกัน โดยแยก 4 ชุดของฮาร์ดแวร์

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

ปัจจุบันการบริการคลาวน์มีการเติบโต และได้รับความนิยมอย่างเพิ่มขึ้น แต่เมื่อการบริการคลาวน์ในลักษณะของซอฟต์แวร์ มีการนำซอฟต์แวร์หลาย ๆ ผู้บริการมาเชื่อมโยงเข้าด้วยกันเสมือนเป็นหนึ่งเจ้าของการเช่า แต่ด้วยลักษณะเฉพาะของซอฟต์แวร์ ซึ่งเมื่อนำความเฉพาะของซอฟต์แวร์ที่แตกต่างไปจากการบริการคลาวน์ในลักษณะของแพลตฟอร์ม รวมทั้งสถาปัตยกรรมที่ผู้บริการไม่ต้องมีความหลากหลายในเชิงของการเข้าใจเหมือนอย่างซอฟต์แวร์ ทำให้การบริการคลาวน์ในลักษณะของซอฟต์แวร์ยังไม่ได้ถูกนำมารวมศูนย์การบริการอย่างแท้จริง ซึ่งจำเป็นต้องมีกระบวนการในการตรวจสอบสิทธ์ที่ต้องรองรับในการทำงานของผู้บริการหลายเจ้าของ และเมื่อมองถึงผู้เช่าเอง ที่มีลักษณะของการเช่าใช้ซอฟต์แวร์ที่เป็นไปได้หลากหลายด้วย จากลักษณะดังกล่าวได้ถูกเรียกว่า “ผู้เช่าหลายราย (Mulit-Tenant)” ซึ่งก็มีตัวแบบหนึ่งที่ได้ทำการรองรับในลักษณะของหลายผู้เช่า คือ MTAS แต่ด้วยลักษณะของตัวแบบดังกล่าว และปัญหาจากความสัมพันธ์ของโมเดล MTAS ที่ทำให้เกิดความสัมพันธ์ระหว่าง 1 บทบาทสามารถเข้าถึงได้มากกว่า 1 การเช่า ซึ่งผู้กำหนดบทบาทเป็นเจ้าของซอฟต์แวร์เกิดความขัดแย้งกัน ผู้วิจัยจึงแก้ไขเพื่อป้องกันไม่ให้เกิดความสัมพันธ์ที่ก่อให้เกิดการกำหนดบทบาทไปยังหลายความสัมพันธ์ได้ เพื่อรองรับในการตรวจสอบสิทธิ์การเข้าถึงในสภาพแวดล้อมของผู้เช่าหลายรายได้อย่างมีประสิทธิภาพ

รวมทั้งเมื่อนำซอฟต์แวร์มารวมกัน ดังนั้นจึงมีความเป็นไปได้ว่า ซอฟต์แวร์จะมีการเชื่อมโยงเข้ามากันได้ ซึ่งการกำหนดสิทธิ์ในลักษณะที่มีการเชื่อมโยงในการเรียกบริการหนึ่งไปยังอีกบริการหนึ่ง ซึ่งผู้วิจัยได้เรียกลักษณะดังกล่าวว่า “ห่วงโซ่การบริการแบบประสานงาน” ซึ่งในตัวแบบ MTAS ไม่มีการรองรับในลักษณะของห่วงโซ่อย่างชัดเจน แต่สามารถกำหนดได้ คือใช้นโยบาย เขียนขึ้นมาและใช้ความเชื่อถือระหว่างการเช่ามากำหนดเป็นสิทธิ์ ซึ่งการทำงานในลักษณดังกล่าวอาจจะมีความยุ่งยากและซับซ้อนขึ้นในเชิงของการกำหนดนโยบาย ดังนั้นในงานวิจัยนี้จึงแก้ปัญหาดังกล่าวโดยสร้างตัวแบบที่มีชื่อว่า C-MTAS ซึ่งได้กำหนดเซตความสัมพันธ์หนึ่งที่ชื่อว่า CCPR มาตัวกำหนดสิทธิ์การเข้าถึงหนึ่งไปยังอีกสิทธิ์การเข้าถึงหนึ่งของการเรียกบริการซอฟต์แวร์ต่างเจ้าของการเช่าได้เพื่อช่วยแก้ไขการกำหนดโนบายในลักษณะของห่วงโซ่การบริการแบบประสานงานได้

หลังจากการกำหนดตัวแบบ C-MTAS ผู้วิจัยจึง Implement จากตัวแบบดังกล่าวได้ออกมาเป็น AaaS ที่เป็นตัวบริการในการตรวจสอบสิทธิ์การเข้าถึง แล้วจึงได้ทดสอบกรณีทดสอบต่าง ๆ เพื่อตรวจสอบความถูกต้องของการทำงาน และในการนำการตัวแบบดังกล่าวจะใช้งานได้จริงในการบริคลาวน์ ผู้วิจัยจึงได้ทดสอบในเชิงเปรียบเทียบ เรื่องของการปรับขนาดของฮาร์ดแวร์ ซึ่ง AaaS มีประสิทธิภาพที่ดีขึ้นในการขยายขนาดของ CPU และ RAM ซึ่งการกระจายตัวของเวลาตอบสนองมีการกระจายตัวที่น้อยลง ซึ่งถือว่าประสิทธิภาพดีขึ้น เพื่อนำมาประยุกต์ในการใช้งานได้จริงในสภาพแวดล้อมบนคลาวน์ได้

5.1. สรุปผลการวิจัยจากผลการทดสอบการทำงานของ AaaS

5.1.1 กรณีทดสอบในกรณีปกติ

ในขั้นตอนการทดสอบการทำงานของ AaaS ในกรณีปกติ ซึ่งหมายถึงการตรวจสอบสิทธิ์การเข้าถึงของการร้องขอของการใช้บริการแบบชั้นเดียว ซึ่งจากตัวอย่างในงานวิจัยตาม

ภาพที่ 3.8 ซึ่งสามารถจำแนกกรณีทดสอบได้ออกเป็น 420 กรณี ซึ่งผลการทดสอบในกรณีปกติมีความถูกต้อง ร้อยเปอร์เซ็นตามผลลัพธ์ที่คาดหวังไว้

5.1.2. กรณีทดสอบในกรณีการเกิดห่วงโซ่การบริการแบบประสานงาน

ในขั้นตอนการทดสอบการทำงานของ AaaS ในกรณีที่เกิดห่วงโซ่การบริการแบบประสานงาน ซึ่งหมายถึงการตรวจสอบสิทธิ์การเข้าถึงของการร้องขอจากสิทธิ์หนึ่งเรียกไปยังอีกสิทธิ์หนึ่งในผู้บริการเดียวกัน หรือต่างผู้บริการได้ ซึ่งจากตัวอย่างในงานวิจัยตาม

ภาพที่ 3.9 ซึ่งมีความเป็นไปได้ใน 7 กรณีทดสอบ ซึ่งผลการทดสอบในกรณีที่เกิดห่วงโซ่การบริการแบบประสานงานมีความถูกต้องร้อยเปอร์เซ็นตามผลลัพธ์ที่คาดหวังไว้

5.2. สรุปผลการวิเคราะห์ในเชิงการเปรียบเทียบ เรื่องของการปรับขนาดของฮาร์ดแวร์ในการรองรับการทำงานของ C-MTAS

ในส่วนนี้ผู้วิจัยได้ใช้ระยะเวลาในการตอบสนองนำมาวิเคราะห์ ซึ่งสามารถสรุปปัจจัยในการวิเคราะห์ได้เป็น 3 ปัจจัย คือ

ค่าตอบสนองเฉลี่ย สำหรับการวิเคราะห์ในข้อมูลดังกล่าวผลปรากฏว่ามีแนวโน้มของกราฟที่แสดงได้ว่าฮาร์ดแวร์เมื่อมีการปรับขนาดในส่วนของ CPU และ RAM ที่มากขึ้น ระยะเวลาในการตอบสนองมากที่สุดกลับน้อยลง

ค่าภาระการทำงานบนเซิร์ฟเวอร์ พบว่าผลการทดสอบ บ่งชี้ได้ว่าเมื่อฮาร์ดแวร์มีการปรับขนาดในส่วนของ CPU และ RAM ที่เพิ่มมากขึ้น ค่าภาระการทำงานก็จะมากยิ่งขึ้นด้วย แสดงได้ว่าประสิทธิภาพการทำงานของ C-MTAS ก็มากขึ้นด้วยเช่นกัน

และสุดท้ายคือ การนำ**กราฟแสดงการแจกแจงความถี่ของระยะเวลาที่ใช้ในการตอบสนอง**มาวิเคราะห์ จึงสรุปได้ว่าเมื่อมีการปรับขนาดของฮาร์ดแวร์ทั้ง CPU และ RAM จึงทำให้ประสิทธิภาพของระยะเวลาการตอบสนองที่ดีขึ้น และยังส่งผลให้ประสิทธิภาพการทำงานของ C-MTAS ดีขึ้นด้วยเช่นกัน

5.3. ข้อเสนอแนะในการทำวิจัยในอนาคต

5.3.1. นำตัว authentication มาเป็นองค์ประกอบหนึ่งที่สำคัญด้วย เนื่องจากการจะตรวจสอบสิทธิ์ได้ ควรจะต้องมีระบบยืนยันตัวบุคคลเข้ามา ทำให้การทำงานที่อยู่ในสภาพแวดล้อมทั่วไปและบนคลาวน์มีสมบูรณ์มากยิ่งขึ้น และระบบการพิสูจน์ตัวตนนั้น จะต้องเป็นระบบที่รองรับในสภาพแวดล้อมของผู้เช่าหลายรายได้

5.3.2. ในเรื่องของการกำหนดนโยบาย (Policy) รวมทั้งกำหนดนโยบายเชิงความสัมพันธ์ของห่วงโซ่การบริการแบบประสานงาน (CCPR Set) ซึ่งจากตัวอย่างในบทที่ 3 ผู้วิจัยได้กำหนดนโยบายขึ้นมา ซึ่งในทางที่พัฒนาต่อไป มี API ที่ช่วยในการจัดนโยบายเหล่านั้นให้ง่ายและมีประสิทธิภาพ ทำให้สามารถนำโมเดลดังกล่าวมาใช้งานในสภาพแวดล้อมจริงด้วย

5.3.3. ในเรื่องของคอขวดของการเก็บและดึงข้อมูลในเซสชั่น ปรับปรุงรูปแบบข้อมูลในการเก็บเซสชั่น ให้รองรับสภาพการทำงานบนคลาวน์ที่ดีกว่าการใช้ฐานข้อมูล MySQl ซึ่งผู้วิจัยเพียงต้องการทดสอบการทำงานเบื้องต้นของ C-MTAS จึงทำนำฐานข้อมูล MySQL มาเก็บข้อมูลเซสชั่นในการทำงานเท่านั้น

รายการอ้างอิง

บทความวารสาร

Calero, J. M., Edwards, N., Kirschnick, J., Wilcock, L., & Wray, M. (2010). Toward a multi-tenancy authorization system for cloud services. *IEEE Security & Privacy*, (pp.48-55).

Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, R., Chandramouli, R. (2001). Proposed NIST Standard for Role - Based Access Control. *ACM, TISSEC August 2001 v.4 n.3,* (pp.224-274).

Jin, J., & Ahn, A.-J. (2006). Role-based Access Management for Ad-hoc Collaborative Sharing. *ACM SACMAT June 2006*, (pp.200-209).

Jin, J., Ahn, A.-J., Shehab, M., & Hu, H. (2007). Towards Trust-aware Access Management for Ad-hoc Collaborations. *IEEE, CollaborateCom*, (pp.41-48).

Liu, P., Chen, Z. (2004). An Access Control Model for Web Services in Business Process. *IEEE, WI*, (pp.292-298).

Mell, P., Grance, T., (2011). The NIST Definition of Cloud Computing. *NIST Special Publication 800-145.*

Sandhu, R., Ferraiolo, D., Kuhn, R. (2001). The NIST Model for Role – Based Access Control: Towards A Unified Standard. *ACM*, (pp.224-274).

Szwarcfiter, J. L., & Lauer, P. E. (1974). Finding the elementary cycles of a directed graph in O (N+M) per cycle. *Computing Laboratory. Technicla Report Series no#60. , Univ. of Newcastle upon Tyne, Newcastle upon Tyne, England*.

Tarjan, R. (1973). Enumberation of the Elementary circuits of a Directed Graph. *SIAM J. Comput.,3,* (pp.146-160).

Tang, B., Li, Q., Sandhu, R. (2013). A Multi-Tenant RBAC Model for Collaborative Cloud Services. *IEEE, PST*, (pp.229-238).

Tang, B., Li, Q., Sandhu, R. (2013). Multi-Tenancy Authorization Models for Collaborative Cloud Services. IEEE, CTS, (pp.132-138).

Tiernan, J. C. (1970). An Efficient Search Algorithm to Find the Elementary Circuits of a Graph. *ACM Volume 13*, (pp.722-726).

Wenzhong, Y., Cauanhe, H., Bo, W., & Zhenyu, Z. (2009). A General Trust Model Based on Trust Algebra. *IEEE.,* (pp.125-129).

Wonohoesodo, R., Tari, Z. (2004). A Role based Access Control for Web Services. *IEEE, SCC' 04*, (pp.49-56).

สื่ออิเล็กทรอนิกส์

eXtensible Access Control Markup Language (XACML) Version 3.0, OASIS Standard, Available from http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf.

Ownership Chain Database, Microsoft, Availabel from http://technet.microsoft.com/en-us/library/ms188676.

XACML v.3.0 core and Hierarchical Role Based Access Control (RBAC) Profile Version 1.0, Available from http://docs.oasis-open.org/xacml/3.0/rbac/v1.0/xacml-3.0-rbac-v1.0.pdf

Security Guidance for Critical Areas of Focus in Cloud Computing V 3.0 [2011] Available from https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf

ภาคผนวก

ภาคผนวก ก
กรณีทดสอบ ในกรณีปกติ

ตารางที่ ก-1

รายละเอียดกรณีทดสอบทั้งหมด ในกรณีปกติ

case id	Role id	Tenant	User id	Resource	Result
1	R1	AuditApp	Dan	addDirectory ()	Not Applicable
2	R1	AuditApp	Dan	downloadFile ()	Not Applicable
3	R1	AuditApp	Dan	manageCreditor ()	Not Applicable
4	R1	AuditApp	Dan	manageIncomeTrans ()	Not Applicable
5	R1	AuditApp	Dan	managePaymentTrans ()	Not Applicable
6	R1	AuditApp	Dan	managerDebtor ()	Not Applicable
7	R1	AuditApp	Dan	previewFile ()	Not Applicable
8	R1	AuditApp	Dan	previewReportDailyAccount ()	Not Applicable
9	R1	AuditApp	Dan	previewReportMonthly ()	Not Applicable
10	R1	AuditApp	Dan	previewReportMonthlyAccount ()	Not Applicable
11	R1	AuditApp	Dan	previewReportYearly ()	Not Applicable
12	R1	AuditApp	Dan	processChecklist ()	Not Applicable
13	R1	AuditApp	Dan	processCloseAccountMonth ()	Not Applicable
14	R1	AuditApp	Dan	uploadFile ()	Not Applicable
15	R1	AuditApp	Dan	verifyReport ()	Not Applicable
16	R1	DocApp	Alic	addDirectory ()	Permit
17	R1	DocApp	Alic	downloadFile ()	Not Applicable
18	R1	DocApp	Alic	manageCreditor ()	Not Applicable
19	R1	DocApp	Alic	manageIncomeTrans ()	Not Applicable
20	R1	DocApp	Alic	managePaymentTrans ()	Not Applicable
21	R1	DocApp	Alic	managerDebtor ()	Not Applicable
22	R1	DocApp	Alic	previewFile ()	Not Applicable

case id	Role id	Tenant	User id	Resource	Result
23	R1	DocApp	Alic	previewReportDailyAccount ()	Not Applicable
24	R1	DocApp	Alic	previewReportMonthly ()	Not Applicable
25	R1	DocApp	Alic	previewReportMonthlyAccount ()	Not Applicable
26	R1	DocApp	Alic	previewReportYearly ()	Not Applicable
27	R1	DocApp	Alic	processChecklist ()	Not Applicable
28	R1	DocApp	Alic	processCloseAccountMonth ()	Not Applicable
29	R1	DocApp	Alic	uploadFile ()	Not Applicable
30	R1	DocApp	Alic	verifyReport ()	Not Applicable
31	R1	FinanApp	Bob	addDirectory ()	Not Applicable
32	R1	FinanApp	Charles	addDirectory ()	Not Applicable
33	R1	FinanApp	Bob	downloadFile ()	Not Applicable
34	R1	FinanApp	Charles	downloadFile ()	Not Applicable
35	R1	FinanApp	Bob	manageCreditor ()	Not Applicable
36	R1	FinanApp	Charles	manageCreditor ()	Not Applicable
37	R1	FinanApp	Bob	manageIncomeTrans ()	Not Applicable
38	R1	FinanApp	Charles	manageIncomeTrans ()	Not Applicable
39	R1	FinanApp	Bob	managePaymentTrans ()	Not Applicable
40	R1	FinanApp	Charles	managePaymentTrans ()	Not Applicable
41	R1	FinanApp	Bob	managerDebtor ()	Not Applicable
42	R1	FinanApp	Charles	managerDebtor ()	Not Applicable
43	R1	FinanApp	Bob	previewFile ()	Not Applicable
44	R1	FinanApp	Charles	previewFile ()	Not Applicable
45	R1	FinanApp	Bob	previewReportDailyAccount ()	Not Applicable
46	R1	FinanApp	Charles	previewReportDailyAccount ()	Not Applicable
47	R1	FinanApp	Bob	previewReportMonthly ()	Not Applicable
48	R1	FinanApp	Charles	previewReportMonthly ()	Not Applicable
49	R1	FinanApp	Bob	previewReportMonthlyAccount ()	Not Applicable
50	R1	FinanApp	Charles	previewReportMonthlyAccount ()	Not Applicable

case id	Role id	Tenant	User id	Resource	Result
51	R1	FinanApp	Bob	previewReportYearly ()	Not Applicable
52	R1	FinanApp	Charles	previewReportYearly ()	Not Applicable
53	R1	FinanApp	Bob	processChecklist ()	Not Applicable
54	R1	FinanApp	Charles	processChecklist ()	Not Applicable
55	R1	FinanApp	Bob	processCloseAccountMonth ()	Not Applicable
56	R1	FinanApp	Charles	processCloseAccountMonth ()	Not Applicable
57	R1	FinanApp	Bob	uploadFile ()	Not Applicable
58	R1	FinanApp	Charles	uploadFile ()	Not Applicable
59	R1	FinanApp	Bob	verifyReport ()	Not Applicable
60	R1	FinanApp	Charles	verifyReport ()	Not Applicable
61	R2	AuditApp	Dan	addDirectory ()	Not Applicable
62	R2	AuditApp	Dan	downloadFile ()	Not Applicable
63	R2	AuditApp	Dan	manageCreditor ()	Not Applicable
64	R2	AuditApp	Dan	manageIncomeTrans ()	Not Applicable
65	R2	AuditApp	Dan	managePaymentTrans ()	Not Applicable
66	R2	AuditApp	Dan	managerDebtor ()	Not Applicable
67	R2	AuditApp	Dan	previewFile ()	Not Applicable
68	R2	AuditApp	Dan	previewReportDailyAccount ()	Not Applicable
69	R2	AuditApp	Dan	previewReportMonthly ()	Not Applicable
70	R2	AuditApp	Dan	previewReportMonthlyAccount ()	Not Applicable
71	R2	AuditApp	Dan	previewReportYearly ()	Not Applicable
72	R2	AuditApp	Dan	processChecklist ()	Not Applicable
73	R2	AuditApp	Dan	processCloseAccountMonth ()	Not Applicable
74	R2	AuditApp	Dan	uploadFile ()	Not Applicable
75	R2	AuditApp	Dan	verifyReport ()	Not Applicable
76	R2	DocApp	Alic	addDirectory ()	Not Applicable
77	R2	DocApp	Alic	downloadFile ()	Not Applicable
78	R2	DocApp	Alic	manageCreditor ()	Not Applicable

case id	Role id	Tenant	User id	Resource	Result
79	R2	DocApp	Alic	manageIncomeTrans ()	Not Applicable
80	R2	DocApp	Alic	managePaymentTrans ()	Not Applicable
81	R2	DocApp	Alic	managerDebtor ()	Not Applicable
82	R2	DocApp	Alic	previewFile ()	Not Applicable
83	R2	DocApp	Alic	previewReportDailyAccount ()	Not Applicable
84	R2	DocApp	Alic	previewReportMonthly ()	Not Applicable
85	R2	DocApp	Alic	previewReportMonthlyAccount ()	Not Applicable
86	R2	DocApp	Alic	previewReportYearly ()	Not Applicable
87	R2	DocApp	Alic	processChecklist ()	Not Applicable
88	R2	DocApp	Alic	processCloseAccountMonth ()	Not Applicable
89	R2	DocApp	Alic	uploadFile ()	Not Applicable
90	R2	DocApp	Alic	verifyReport ()	Not Applicable
91	R2	FinanApp	Bob	addDirectory ()	Not Applicable
92	R2	FinanApp	Charles	addDirectory ()	Not Applicable
93	R2	FinanApp	Bob	downloadFile ()	Not Applicable
94	R2	FinanApp	Charles	downloadFile ()	Not Applicable
95	R2	FinanApp	Bob	manageCreditor ()	Not Applicable
96	R2	FinanApp	Charles	manageCreditor ()	Not Applicable
97	R2	FinanApp	Bob	manageIncomeTrans ()	Not Applicable
98	R2	FinanApp	Charles	manageIncomeTrans ()	Not Applicable
99	R2	FinanApp	Bob	managePaymentTrans ()	Not Applicable
100	R2	FinanApp	Charles	managePaymentTrans ()	Not Applicable
101	R2	FinanApp	Bob	managerDebtor ()	Not Applicable
102	R2	FinanApp	Charles	managerDebtor ()	Not Applicable
103	R2	FinanApp	Bob	previewFile ()	Not Applicable
104	R2	FinanApp	Charles	previewFile ()	Not Applicable
105	R2	FinanApp	Bob	previewReportDailyAccount ()	Not Applicable
106	R2	FinanApp	Charles	previewReportDailyAccount ()	Not Applicable

case id	Role id	Tenant	User id	Resource	Result
107	R2	FinanApp	Bob	previewReportMonthly ()	Not Applicable
108	R2	FinanApp	Charles	previewReportMonthly ()	Not Applicable
109	R2	FinanApp	Bob	previewReportMonthlyAccount ()	Not Applicable
110	R2	FinanApp	Charles	previewReportMonthlyAccount ()	Not Applicable
111	R2	FinanApp	Bob	previewReportYearly ()	Not Applicable
112	R2	FinanApp	Charles	previewReportYearly ()	Not Applicable
113	R2	FinanApp	Bob	processChecklist ()	Not Applicable
114	R2	FinanApp	Charles	processChecklist ()	Not Applicable
115	R2	FinanApp	Bob	processCloseAccountMonth ()	Not Applicable
116	R2	FinanApp	Charles	processCloseAccountMonth ()	Not Applicable
117	R2	FinanApp	Bob	uploadFile ()	Not Applicable
118	R2	FinanApp	Charles	uploadFile ()	Not Applicable
119	R2	FinanApp	Bob	verifyReport ()	Not Applicable
120	R2	FinanApp	Charles	verifyReport ()	Not Applicable
121	R3	AuditApp	Dan	addDirectory ()	Not Applicable
122	R3	AuditApp	Dan	downloadFile ()	Not Applicable
123	R3	AuditApp	Dan	manageCreditor ()	Not Applicable
124	R3	AuditApp	Dan	manageIncomeTrans ()	Not Applicable
125	R3	AuditApp	Dan	managePaymentTrans ()	Not Applicable
126	R3	AuditApp	Dan	managerDebtor ()	Not Applicable
127	R3	AuditApp	Dan	previewFile ()	Not Applicable
128	R3	AuditApp	Dan	previewReportDailyAccount ()	Not Applicable
129	R3	AuditApp	Dan	previewReportMonthly ()	Not Applicable
130	R3	AuditApp	Dan	previewReportMonthlyAccount ()	Not Applicable
131	R3	AuditApp	Dan	previewReportYearly ()	Not Applicable
132	R3	AuditApp	Dan	processChecklist ()	Not Applicable
133	R3	AuditApp	Dan	processCloseAccountMonth ()	Not Applicable
134	R3	AuditApp	Dan	uploadFile ()	Not Applicable

case id	Role id	Tenant	User id	Resource	Result
135	R3	AuditApp	Dan	verifyReport ()	Not Applicable
136	R3	DocApp	Alic	addDirectory ()	Not Applicable
137	R3	DocApp	Alic	downloadFile ()	Permit
138	R3	DocApp	Alic	manageCreditor ()	Not Applicable
139	R3	DocApp	Alic	manageIncomeTrans ()	Not Applicable
140	R3	DocApp	Alic	managePaymentTrans ()	Not Applicable
141	R3	DocApp	Alic	managerDebtor ()	Not Applicable
142	R3	DocApp	Alic	previewFile ()	Permit
143	R3	DocApp	Alic	previewReportDailyAccount ()	Not Applicable
144	R3	DocApp	Alic	previewReportMonthly ()	Not Applicable
145	R3	DocApp	Alic	previewReportMonthlyAccount ()	Not Applicable
146	R3	DocApp	Alic	previewReportYearly ()	Not Applicable
147	R3	DocApp	Alic	processChecklist ()	Not Applicable
148	R3	DocApp	Alic	processCloseAccountMonth ()	Not Applicable
149	R3	DocApp	Alic	uploadFile ()	Not Applicable
150	R3	DocApp	Alic	verifyReport ()	Not Applicable
151	R3	FinanApp	Bob	addDirectory ()	Not Applicable
152	R3	FinanApp	Charles	addDirectory ()	Not Applicable
153	R3	FinanApp	Bob	downloadFile ()	Not Applicable
154	R3	FinanApp	Charles	downloadFile ()	Not Applicable
155	R3	FinanApp	Bob	manageCreditor ()	Not Applicable
156	R3	FinanApp	Charles	manageCreditor ()	Not Applicable
157	R3	FinanApp	Bob	manageIncomeTrans ()	Not Applicable
158	R3	FinanApp	Charles	manageIncomeTrans ()	Not Applicable
159	R3	FinanApp	Bob	managePaymentTrans ()	Not Applicable
160	R3	FinanApp	Charles	managePaymentTrans ()	Not Applicable
161	R3	FinanApp	Bob	managerDebtor ()	Not Applicable
162	R3	FinanApp	Charles	managerDebtor ()	Not Applicable

case id	Role id	Tenant	User id	Resource	Result
163	R3	FinanApp	Bob	previewFile ()	Not Applicable
164	R3	FinanApp	Charles	previewFile ()	Not Applicable
165	R3	FinanApp	Bob	previewReportDailyAccount ()	Not Applicable
166	R3	FinanApp	Charles	previewReportDailyAccount ()	Not Applicable
167	R3	FinanApp	Bob	previewReportMonthly ()	Not Applicable
168	R3	FinanApp	Charles	previewReportMonthly ()	Not Applicable
169	R3	FinanApp	Bob	previewReportMonthlyAccount ()	Not Applicable
170	R3	FinanApp	Charles	previewReportMonthlyAccount ()	Not Applicable
171	R3	FinanApp	Bob	previewReportYearly ()	Not Applicable
172	R3	FinanApp	Charles	previewReportYearly ()	Not Applicable
173	R3	FinanApp	Bob	processChecklist ()	Not Applicable
174	R3	FinanApp	Charles	processChecklist ()	Not Applicable
175	R3	FinanApp	Bob	processCloseAccountMonth ()	Not Applicable
176	R3	FinanApp	Charles	processCloseAccountMonth ()	Not Applicable
177	R3	FinanApp	Bob	uploadFile ()	Not Applicable
178	R3	FinanApp	Charles	uploadFile ()	Not Applicable
179	R3	FinanApp	Bob	verifyReport ()	Not Applicable
180	R3	FinanApp	Charles	verifyReport ()	Not Applicable
181	R4	AuditApp	Dan	addDirectory ()	Not Applicable
182	R4	AuditApp	Dan	downloadFile ()	Not Applicable
183	R4	AuditApp	Dan	manageCreditor ()	Not Applicable
184	R4	AuditApp	Dan	manageIncomeTrans ()	Not Applicable
185	R4	AuditApp	Dan	managePaymentTrans ()	Not Applicable
186	R4	AuditApp	Dan	managerDebtor ()	Not Applicable
187	R4	AuditApp	Dan	previewFile ()	Not Applicable
188	R4	AuditApp	Dan	previewReportDailyAccount ()	Not Applicable
189	R4	AuditApp	Dan	previewReportMonthly ()	Not Applicable
190	R4	AuditApp	Dan	previewReportMonthlyAccount ()	Not Applicable

case id	Role id	Tenant	User id	Resource	Result
191	R4	AuditApp	Dan	previewReportYearly ()	Not Applicable
192	R4	AuditApp	Dan	processChecklist ()	Not Applicable
193	R4	AuditApp	Dan	processCloseAccountMonth ()	Not Applicable
194	R4	AuditApp	Dan	uploadFile ()	Not Applicable
195	R4	AuditApp	Dan	verifyReport ()	Not Applicable
196	R4	DocApp	Alic	addDirectory ()	Not Applicable
197	R4	DocApp	Alic	downloadFile ()	Not Applicable
198	R4	DocApp	Alic	manageCreditor ()	Not Applicable
199	R4	DocApp	Alic	manageIncomeTrans ()	Not Applicable
200	R4	DocApp	Alic	managePaymentTrans ()	Not Applicable
201	R4	DocApp	Alic	managerDebtor ()	Not Applicable
202	R4	DocApp	Alic	previewFile ()	Not Applicable
203	R4	DocApp	Alic	previewReportDailyAccount ()	Not Applicable
204	R4	DocApp	Alic	previewReportMonthly ()	Not Applicable
205	R4	DocApp	Alic	previewReportMonthlyAccount ()	Not Applicable
206	R4	DocApp	Alic	previewReportYearly ()	Not Applicable
207	R4	DocApp	Alic	processChecklist ()	Not Applicable
208	R4	DocApp	Alic	processCloseAccountMonth ()	Not Applicable
209	R4	DocApp	Alic	uploadFile ()	Not Applicable
210	R4	DocApp	Alic	verifyReport ()	Not Applicable
211	R4	FinanApp	Bob	addDirectory ()	Not Applicable
212	R4	FinanApp	Charles	addDirectory ()	Not Applicable
213	R4	FinanApp	Bob	downloadFile ()	Not Applicable
214	R4	FinanApp	Charles	downloadFile ()	Not Applicable
215	R4	FinanApp	Bob	manageCreditor ()	Permit
216	R4	FinanApp	Charles	manageCreditor ()	Not Applicable
217	R4	FinanApp	Bob	manageIncomeTrans ()	Permit
218	R4	FinanApp	Charles	manageIncomeTrans ()	Not Applicable

case id	Role id	Tenant	User id	Resource	Result
219	R4	FinanApp	Bob	managePaymentTrans ()	Not Applicable
220	R4	FinanApp	Charles	managePaymentTrans ()	Not Applicable
221	R4	FinanApp	Bob	managerDebtor ()	Permit
222	R4	FinanApp	Charles	managerDebtor ()	Not Applicable
223	R4	FinanApp	Bob	previewFile ()	Not Applicable
224	R4	FinanApp	Charles	previewFile ()	Not Applicable
225	R4	FinanApp	Bob	previewReportDailyAccount ()	Not Applicable
226	R4	FinanApp	Charles	previewReportDailyAccount ()	Not Applicable
227	R4	FinanApp	Bob	previewReportMonthly ()	Not Applicable
228	R4	FinanApp	Charles	previewReportMonthly ()	Not Applicable
229	R4	FinanApp	Bob	previewReportMonthlyAccount ()	Not Applicable
230	R4	FinanApp	Charles	previewReportMonthlyAccount ()	Not Applicable
231	R4	FinanApp	Bob	previewReportYearly ()	Not Applicable
232	R4	FinanApp	Charles	previewReportYearly ()	Not Applicable
233	R4	FinanApp	Bob	processChecklist ()	Not Applicable
234	R4	FinanApp	Charles	processChecklist ()	Not Applicable
235	R4	FinanApp	Bob	processCloseAccountMonth ()	Not Applicable
236	R4	FinanApp	Charles	processCloseAccountMonth ()	Not Applicable
237	R4	FinanApp	Bob	uploadFile ()	Not Applicable
238	R4	FinanApp	Charles	uploadFile ()	Not Applicable
239	R4	FinanApp	Bob	verifyReport ()	Not Applicable
240	R4	FinanApp	Charles	verifyReport ()	Not Applicable
241	R5	AuditApp	Dan	addDirectory ()	Not Applicable
242	R5	AuditApp	Dan	downloadFile ()	Not Applicable
243	R5	AuditApp	Dan	manageCreditor ()	Not Applicable
244	R5	AuditApp	Dan	manageIncomeTrans ()	Not Applicable
245	R5	AuditApp	Dan	managePaymentTrans ()	Not Applicable
246	R5	AuditApp	Dan	managerDebtor ()	Not Applicable

case id	Role id	Tenant	User id	Resource	Result
247	R5	AuditApp	Dan	previewFile ()	Not Applicable
248	R5	AuditApp	Dan	previewReportDailyAccount ()	Not Applicable
249	R5	AuditApp	Dan	previewReportMonthly ()	Not Applicable
250	R5	AuditApp	Dan	previewReportMonthlyAccount ()	Not Applicable
251	R5	AuditApp	Dan	previewReportYearly ()	Not Applicable
252	R5	AuditApp	Dan	processChecklist ()	Not Applicable
253	R5	AuditApp	Dan	processCloseAccountMonth ()	Not Applicable
254	R5	AuditApp	Dan	uploadFile ()	Not Applicable
255	R5	AuditApp	Dan	verifyReport ()	Not Applicable
256	R5	DocApp	Alic	addDirectory ()	Not Applicable
257	R5	DocApp	Alic	downloadFile ()	Not Applicable
258	R5	DocApp	Alic	manageCreditor ()	Not Applicable
259	R5	DocApp	Alic	manageIncomeTrans ()	Not Applicable
260	R5	DocApp	Alic	managePaymentTrans ()	Not Applicable
261	R5	DocApp	Alic	managerDebtor ()	Not Applicable
262	R5	DocApp	Alic	previewFile ()	Not Applicable
263	R5	DocApp	Alic	previewReportDailyAccount ()	Not Applicable
264	R5	DocApp	Alic	previewReportMonthly ()	Not Applicable
265	R5	DocApp	Alic	previewReportMonthlyAccount ()	Not Applicable
266	R5	DocApp	Alic	previewReportYearly ()	Not Applicable
267	R5	DocApp	Alic	processChecklist ()	Not Applicable
268	R5	DocApp	Alic	processCloseAccountMonth ()	Not Applicable
269	R5	DocApp	Alic	uploadFile ()	Not Applicable
270	R5	DocApp	Alic	verifyReport ()	Not Applicable
271	R5	FinanApp	Bob	addDirectory ()	Not Applicable
272	R5	FinanApp	Charles	addDirectory ()	Not Applicable
273	R5	FinanApp	Bob	downloadFile ()	Not Applicable
274	R5	FinanApp	Charles	downloadFile ()	Not Applicable

case id	Role id	Tenant	User id	Resource	Result
275	R5	FinanApp	Bob	manageCreditor ()	Not Applicable
276	R5	FinanApp	Charles	manageCreditor ()	Not Applicable
277	R5	FinanApp	Bob	manageIncomeTrans ()	Not Applicable
278	R5	FinanApp	Charles	manageIncomeTrans ()	Not Applicable
279	R5	FinanApp	Bob	managePaymentTrans ()	Not Applicable
280	R5	FinanApp	Charles	managePaymentTrans ()	Permit
281	R5	FinanApp	Bob	managerDebtor ()	Not Applicable
282	R5	FinanApp	Charles	managerDebtor ()	Not Applicable
283	R5	FinanApp	Bob	previewFile ()	Not Applicable
284	R5	FinanApp	Charles	previewFile ()	Not Applicable
285	R5	FinanApp	Bob	previewReportDailyAccount ()	Not Applicable
286	R5	FinanApp	Charles	previewReportDailyAccount ()	Permit
287	R5	FinanApp	Bob	previewReportMonthly ()	Not Applicable
288	R5	FinanApp	Charles	previewReportMonthly ()	Not Applicable
289	R5	FinanApp	Bob	previewReportMonthlyAccount ()	Not Applicable
290	R5	FinanApp	Charles	previewReportMonthlyAccount ()	Not Applicable
291	R5	FinanApp	Bob	previewReportYearly ()	Not Applicable
292	R5	FinanApp	Charles	previewReportYearly ()	Not Applicable
293	R5	FinanApp	Bob	processChecklist ()	Not Applicable
294	R5	FinanApp	Charles	processChecklist ()	Not Applicable
295	R5	FinanApp	Bob	processCloseAccountMonth ()	Not Applicable
296	R5	FinanApp	Charles	processCloseAccountMonth ()	Permit
297	R5	FinanApp	Bob	uploadFile ()	Not Applicable
298	R5	FinanApp	Charles	uploadFile ()	Not Applicable
299	R5	FinanApp	Bob	verifyReport ()	Not Applicable
300	R5	FinanApp	Charles	verifyReport ()	Not Applicable
301	R6	AuditApp	Dan	addDirectory ()	Not Applicable
302	R6	AuditApp	Dan	downloadFile ()	Not Applicable

case id	Role id	Tenant	User id	Resource	Result
303	R6	AuditApp	Dan	manageCreditor ()	Not Applicable
304	R6	AuditApp	Dan	manageIncomeTrans ()	Not Applicable
305	R6	AuditApp	Dan	managePaymentTrans ()	Not Applicable
306	R6	AuditApp	Dan	managerDebtor ()	Not Applicable
307	R6	AuditApp	Dan	previewFile ()	Not Applicable
308	R6	AuditApp	Dan	previewReportDailyAccount ()	Not Applicable
309	R6	AuditApp	Dan	previewReportMonthly ()	Not Applicable
310	R6	AuditApp	Dan	previewReportMonthlyAccount ()	Not Applicable
311	R6	AuditApp	Dan	previewReportYearly ()	Not Applicable
312	R6	AuditApp	Dan	processChecklist ()	Not Applicable
313	R6	AuditApp	Dan	processCloseAccountMonth ()	Not Applicable
314	R6	AuditApp	Dan	uploadFile ()	Not Applicable
315	R6	AuditApp	Dan	verifyReport ()	Not Applicable
316	R6	DocApp	Alic	addDirectory ()	Not Applicable
317	R6	DocApp	Alic	downloadFile ()	Not Applicable
318	R6	DocApp	Alic	manageCreditor ()	Not Applicable
319	R6	DocApp	Alic	manageIncomeTrans ()	Not Applicable
320	R6	DocApp	Alic	managePaymentTrans ()	Not Applicable
321	R6	DocApp	Alic	managerDebtor ()	Not Applicable
322	R6	DocApp	Alic	previewFile ()	Not Applicable
323	R6	DocApp	Alic	previewReportDailyAccount ()	Not Applicable
324	R6	DocApp	Alic	previewReportMonthly ()	Not Applicable
325	R6	DocApp	Alic	previewReportMonthlyAccount ()	Not Applicable
326	R6	DocApp	Alic	previewReportYearly ()	Not Applicable
327	R6	DocApp	Alic	processChecklist ()	Not Applicable
328	R6	DocApp	Alic	processCloseAccountMonth ()	Not Applicable
329	R6	DocApp	Alic	uploadFile ()	Not Applicable
330	R6	DocApp	Alic	verifyReport ()	Not Applicable

case id	Role id	Tenant	User id	Resource	Result
331	R6	FinanApp	Bob	addDirectory ()	Not Applicable
332	R6	FinanApp	Charles	addDirectory ()	Not Applicable
333	R6	FinanApp	Bob	downloadFile ()	Not Applicable
334	R6	FinanApp	Charles	downloadFile ()	Not Applicable
335	R6	FinanApp	Bob	manageCreditor ()	Not Applicable
336	R6	FinanApp	Charles	manageCreditor ()	Not Applicable
337	R6	FinanApp	Bob	manageIncomeTrans ()	Not Applicable
338	R6	FinanApp	Charles	manageIncomeTrans ()	Not Applicable
339	R6	FinanApp	Bob	managePaymentTrans ()	Not Applicable
340	R6	FinanApp	Charles	managePaymentTrans ()	Not Applicable
341	R6	FinanApp	Bob	managerDebtor ()	Not Applicable
342	R6	FinanApp	Charles	managerDebtor ()	Not Applicable
343	R6	FinanApp	Bob	previewFile ()	Not Applicable
344	R6	FinanApp	Charles	previewFile ()	Not Applicable
345	R6	FinanApp	Bob	previewReportDailyAccount ()	Not Applicable
346	R6	FinanApp	Charles	previewReportDailyAccount ()	Not Applicable
347	R6	FinanApp	Bob	previewReportMonthly ()	Not Applicable
348	R6	FinanApp	Charles	previewReportMonthly ()	Not Applicable
349	R6	FinanApp	Bob	previewReportMonthlyAccount ()	Not Applicable
350	R6	FinanApp	Charles	previewReportMonthlyAccount ()	Not Applicable
351	R6	FinanApp	Bob	previewReportYearly ()	Not Applicable
352	R6	FinanApp	Charles	previewReportYearly ()	Not Applicable
353	R6	FinanApp	Bob	processChecklist ()	Not Applicable
354	R6	FinanApp	Charles	processChecklist ()	Not Applicable
355	R6	FinanApp	Bob	processCloseAccountMonth ()	Not Applicable
356	R6	FinanApp	Charles	processCloseAccountMonth ()	Not Applicable
357	R6	FinanApp	Bob	uploadFile ()	Not Applicable
358	R6	FinanApp	Charles	uploadFile ()	Not Applicable

case id	Role id	Tenant	User id	Resource	Result
359	R6	FinanApp	Bob	verifyReport ()	Not Applicable
360	R6	FinanApp	Charles	verifyReport ()	Not Applicable
361	R7	AuditApp	Dan	addDirectory ()	Not Applicable
362	R7	AuditApp	Dan	downloadFile ()	Not Applicable
363	R7	AuditApp	Dan	manageCreditor ()	Not Applicable
364	R7	AuditApp	Dan	manageIncomeTrans ()	Not Applicable
365	R7	AuditApp	Dan	managePaymentTrans ()	Not Applicable
366	R7	AuditApp	Dan	managerDebtor ()	Not Applicable
367	R7	AuditApp	Dan	previewFile ()	Not Applicable
368	R7	AuditApp	Dan	previewReportDailyAccount ()	Not Applicable
369	R7	AuditApp	Dan	previewReportMonthly ()	Permit
370	R7	AuditApp	Dan	previewReportMonthlyAccount ()	Not Applicable
371	R7	AuditApp	Dan	previewReportYearly ()	Permit
372	R7	AuditApp	Dan	processChecklist ()	Permit
373	R7	AuditApp	Dan	processCloseAccountMonth ()	Not Applicable
374	R7	AuditApp	Dan	uploadFile ()	Not Applicable
375	R7	AuditApp	Dan	verifyReport ()	Permit
376	R7	DocApp	Alic	addDirectory ()	Not Applicable
377	R7	DocApp	Alic	downloadFile ()	Not Applicable
378	R7	DocApp	Alic	manageCreditor ()	Not Applicable
379	R7	DocApp	Alic	manageIncomeTrans ()	Not Applicable
380	R7	DocApp	Alic	managePaymentTrans ()	Not Applicable
381	R7	DocApp	Alic	managerDebtor ()	Not Applicable
382	R7	DocApp	Alic	previewFile ()	Not Applicable
383	R7	DocApp	Alic	previewReportDailyAccount ()	Not Applicable
384	R7	DocApp	Alic	previewReportMonthly ()	Not Applicable
385	R7	DocApp	Alic	previewReportMonthlyAccount ()	Not Applicable
386	R7	DocApp	Alic	previewReportYearly ()	Not Applicable

case id	Role id	Tenant	User id	Resource	Result
387	R7	DocApp	Alic	processChecklist ()	Not Applicable
388	R7	DocApp	Alic	processCloseAccountMonth ()	Not Applicable
389	R7	DocApp	Alic	uploadFile ()	Not Applicable
390	R7	DocApp	Alic	verifyReport ()	Not Applicable
391	R7	FinanApp	Bob	addDirectory ()	Not Applicable
392	R7	FinanApp	Charles	addDirectory ()	Not Applicable
393	R7	FinanApp	Bob	downloadFile ()	Not Applicable
394	R7	FinanApp	Charles	downloadFile ()	Not Applicable
395	R7	FinanApp	Bob	manageCreditor ()	Not Applicable
396	R7	FinanApp	Charles	manageCreditor ()	Not Applicable
397	R7	FinanApp	Bob	manageIncomeTrans ()	Not Applicable
398	R7	FinanApp	Charles	manageIncomeTrans ()	Not Applicable
399	R7	FinanApp	Bob	managePaymentTrans ()	Not Applicable
400	R7	FinanApp	Charles	managePaymentTrans ()	Not Applicable
401	R7	FinanApp	Bob	managerDebtor ()	Not Applicable
402	R7	FinanApp	Charles	managerDebtor ()	Not Applicable
403	R7	FinanApp	Bob	previewFile ()	Not Applicable
404	R7	FinanApp	Charles	previewFile ()	Not Applicable
405	R7	FinanApp	Bob	previewReportDailyAccount ()	Not Applicable
406	R7	FinanApp	Charles	previewReportDailyAccount ()	Not Applicable
407	R7	FinanApp	Bob	previewReportMonthly ()	Not Applicable
408	R7	FinanApp	Charles	previewReportMonthly ()	Not Applicable
409	R7	FinanApp	Bob	previewReportMonthlyAccount ()	Not Applicable
410	R7	FinanApp	Charles	previewReportMonthlyAccount ()	Not Applicable
411	R7	FinanApp	Bob	previewReportYearly ()	Not Applicable
412	R7	FinanApp	Charles	previewReportYearly ()	Not Applicable
413	R7	FinanApp	Bob	processChecklist ()	Not Applicable
414	R7	FinanApp	Charles	processChecklist ()	Not Applicable

case id	Role id	Tenant	User id	Resource	Result
415	R7	FinanApp	Bob	processCloseAccountMonth ()	Not Applicable
416	R7	FinanApp	Charles	processCloseAccountMonth ()	Not Applicable
417	R7	FinanApp	Bob	uploadFile ()	Not Applicable
418	R7	FinanApp	Charles	uploadFile ()	Not Applicable
419	R7	FinanApp	Bob	verifyReport ()	Not Applicable
420	R7	FinanApp	Charles	verifyReport ()	Not Applicable

ประวัติผู้เขียน

ชื่อ	นาย ดนัย ทองแสง
วันเดือนปีเกิด	5 พฤษภาคม 2530
ตำแหน่ง	นักวิเคราะห์และออกแบบระบบ บริษัท ไอเจนโก้ จำกัด

ผลงานทางวิชาการ

A Chain Calling in Coordination for Multi-Tenant Collaborative Cloud Services: , 2014 18th International Computer Science and Engineering Conference (ICSEC)