การออกแบบสถาปัตยกรรมของระบบบริหารการบำรุงรักษาเครื่องจักร ตามแนวคิดของ SOA และคลาวด์คอมพิวติง

ชัยรัตน์ ศรีพงษ์ธนากุล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
บัณฑิตวิทยาลัย สาขาเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีไทย-ญี่ปุ่น
ปีการศึกษา 2559

AN ARCHITECTURAL DESIGN OF CMMS TOWARDS SOA AND CLOUD COMPUTING

Chairat Sripongtanakul

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Information Technology

Graduate School

Thai-Nichi Institute of Technology

Academic Year 2016

หัวข้อวิทยานิพนธ์	การออกแบบสถาปัตยกรรมของระบบบริหารการบำรุงรักษาเครื่องจักร ตามแนวคิดของ SOA และคลาวด์คอมพิวติง
โดย	ชัยรัตน์ ศรีพงษ์ธนากุล
สาขาวิชา	เทคโนโลยีสารสนเทศ
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ดร. กิตติมา เมฆาบัญชากิจ

บัณฑิตวิทยาลัย สถาบันเทคโนโลยีไทย-ญี่ปุ่น อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีบัณฑิตวิทยาลัย

(รองศาสตราจารย์ ดร. พิชิต สุขเจริญพงษ์)

วันที่.....เดือน.....พ.ศ.....

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(รองศาสตราจารย์ ดร. วีระ บุญจริง)

.....กรรมการ

(ดร. ภาสกร อภิรักษ์วรพินิต)

.....กรรมการ

(ดร. กันติชา กิตติพีรชล)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์

(ดร. กิตติมา เมฆาบัญชากิจ)

ชัยรัตน์ ศรีพงษ์ธนากุล : การออกแบบสถาปัตยกรรมของระบบบริหารการบำรุงรักษาเครื่องจักร ตามแนวคิดของ SOA และคลาวด์คอมพิวติง. อาจารย์ที่ปรึกษา : ดร. กิตติมา เมฆาบัญชากิจ, 42 หน้า.

ซอฟต์แวร์ประเภทระบบบริหารการบำรุงรักษาเครื่องจักร (CMMS) ที่ใช้งานอยู่บนสถาปัตยกรรมแบบ Client/Server ต้องอาศัยประสิทธิภาพของเครื่องผู้ให้บริการ เป็นหลัก เครื่องผู้ให้บริการจำเป็นต้องมีประสิทธิภาพที่ดีและมีทรัพยากรเพียงพอต่อการให้บริการ แต่ในปัจจุบันมีเทคโนโลยีระบบคลาวด์คอมพิวติง (Cloud Computing) ซึ่งมีความยืดหยุ่นได้ดีกว่า สามารถปรับตัวให้มีทรัพยากรที่เพียงพอต่อผู้ใช้งานได้อย่างเหมาะสม การปรับเปลี่ยนซอฟต์แวร์ให้สามารถใช้งานได้บนระบบคลาวด์โดยการพัฒนาซอฟต์แวร์เพื่อเปลี่ยนสถาปัตยกรรมทั้งหมดในคราวเดียวมีความเป็นไปได้และใช้เวลานานมากเกินไป งานวิจัยนี้นำเสนอแนวทางการออกแบบสถาปัตยกรรมเพื่อให้ซอฟต์แวร์ CMMS ที่อยู่บนสถาปัตยกรรมแบบ Client/Server ยังคงใช้งานได้ดังเดิม และมีทางเลือกให้สามารถใช้งานได้บนสถาปัตยกรรมคลาวด์ คอมพิวติงร่วมด้วยได้อย่างต่อเนื่อง การออกแบบสถาปัตยกรรมนี้เป็นแนวทางที่ช่วยในการปรับให้ซอฟต์แวร์ CMMS เข้ากับเทคโนโลยีคลาวด์คอมพิวติง โดยอาศัยสถาปัตยกรรมระบบเชิงบริการ (Service-Oriented Architecture, SOA) เพื่อให้ซอฟต์แวร์ CMMS ที่ทำงานทั้งสองสถาปัตยกรรมสามารถเชื่อมโยงข้อมูลถึงกันผ่าน API ได้

บัณฑิตวิทยาลัย ลายมือชื่อนักศึกษา .....
สาขาวิชา เทคโนโลยีสารสนเทศ ลายมือชื่ออาจารย์ที่ปรึกษา .....
ปีการศึกษา 2559

CHAIRAT SRIPONGTANAKUL : AN ARCHITECTURAL DESIGN OF CMMS TOWARDS SOA AND CLOUD COMPUTING. ADVISOR : DR. KITTIMA MEKHABUNCHAKIJ, 42 PP.

For the Software each type as use for preventive maintenance machine system. Active by refer data base form Server and the main parameter of operation efficiency depend on the performance of system and server include resources to service. But nowadays, novel Cloud Computing is a technology that can be flexible of operate. It can adaptable resources for the each user. Software modifications are available on cloud computing. To develop software to change all architectures at once. Possible but must take long time of operation. Development of Architecture for CMMS software running on architectures type Client/Server can operate under previous condition and there is an option to continue to use CMMS software with Cloud Computing Architecture. It is a great way to apply and improve CMMS software with cloud computing technology. By using Service-Oriented Architecture (SOA) purpose for CMMS software as operating under two architectures can be link and communicate by API.

Graduate School Student's Signature.....

Field of Study Information Technology Advisor's Signature.....

Academic Year 2016

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สาเร็จลงได้ด้วยดี ด้วยความกรุณาของ ดร. กิติมา เมฆาบัญชากิจ อาจารย์ที่ปรึกษาวิทยานิพนธ์ เป็นผู้ให้คำปรึกษาแนะนำแนวทาง และตรวจสอบงานวิทยานิพนธ์จนเสร็จสมบูรณ์ ผู้วิจัยขอกราบขอบพระคุณ และสานึกในพระคุณเป็นอย่างสูง

ผู้วิจัยขอกราบขอบพระคุณคณะกรรมการสอบทุกท่าน ได้แก่ รองศาสตราจารย์ ดร. วีระ บุญจริง ดร. ภาสกร อภิรักษ์วรพินิต และดร.กันติชา กิตติพีรชล ที่กรุณาให้ข้อแนะนำการทำวิจัย และตรวจแก้ไขวิทยานิพนธ์ฉบับนี้ให้สมบูรณ์ยิ่งขึ้น

ผู้วิจัยขอกราบขอบพระคุณบิดา มารดา และครอบครัวที่ให้โอกาสผู้วิจัยในการศึกษาต่อ รวมทั้งการทาวิจัยในครั้งนี้ และขอขอบคุณ พี่ๆ เพื่อนๆ และน้องๆ ที่คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีไทย-ญี่ปุ่น ที่เป็นกาลังใจและให้ความช่วยเหลือด้วยดีเสมอมา

สุดท้ายนี้ผู้วิจัยหวังว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์ในการศึกษา และนำไปประยุกต์ใช้ได้ต่อไปในอนาคต ขอขอบคุณอีกครั้ง

ชัยรัตน์ ศรีพงษ์ธนากุล

สารบัญ

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญรูป

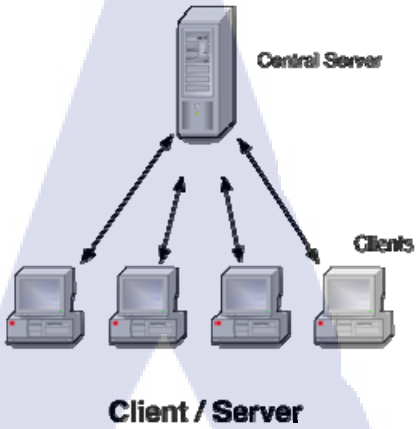
บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในภาคอุตสาหกรรม ซอฟต์แวร์ประเภทระบบการบริหารงานบำรุงรักษา (Maintenance Management Software: MMS) หรือที่เรียกว่า ระบบการบริหารสินทรัพย์องค์กร (Enterprise Asset Management: EAM) และ ระบบการบริหารงานบำรุงรักษา (Computerized Maintenance Management System: CMMS) เป็นซอฟต์แวร์สำหรับการการบริหารจัดการงานบำรุงรักษาและซ่อมบำรุง เพื่อช่วยให้ธุรกิจสามารถบริหารจัดการและการติดตาม เครื่องจักร เครื่องมือ หรือ อุปกรณ์ ยานพาหนะ หรือ สินทรัพย์ ต่างๆ ในการบำรุงรักษา และวางแผนเพื่อให้เครื่องมือและอุปกรณ์ต่างๆ ในภาคธุรกิจ สามารถใช้งานได้ต่อเนื่องอย่างมีประสิทธิภาพ และลดการที่จะไม่สามารถใช้งานเครื่องจักร เครื่องมือ หรือ อุปกรณ์ ซึ่งจะทำให้เกิดผลกระทบ และการสูญเสียโอกาสในธุรกิจ

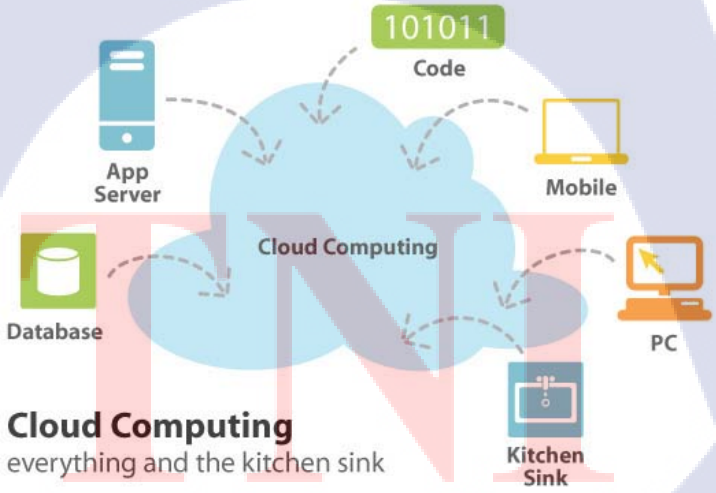
ซอฟต์แวร์ระบบการบริหารงานบำรุงรักษา ถือเป็นระบบบริหารจัดการด้านเทคนิคที่ช่วยเหลือเจ้าหน้าที่ที่อยู่ในหน่วยงานบำรุงรักษา โดยสามารถนำข้อมูลในระบบฯ มาช่วยในการวางแผนวิเคราะห์เชิงประสบการณ์ เพื่อให้การทำงานบรรลุผลเป็นไปตามเป้าหมายของหน่วยงานบำรุงรักษา เช่น ทำให้เครื่องจักรมีความพร้อมใช้งาน และใช้งานได้อย่างปลอดภัย เพื่อควบคุมต้นทุนการบำรุงรักษาที่เหมาะสม เพื่อวางแผนซ่อมบำรุงเชิงป้องกันประจำวัน เพื่อจัดการงานซ่อมประจำวัน เพื่อทำการบันทึกข้อมูลการซ่อมบำรุง และสืบค้นประวัติปัญหาที่พบและวิธีการซ่อมบำรุงที่ผ่านมาแล้ว

ในปัจจุบันซอฟต์แวร์ระบบการบริหารงานบำรุงรักษา ยังคงมีโครงสร้างสถาปัตยกรรมแบบ Client/Server ตามรูปที่ 1.1 ซึ่งเป็นการใช้งานในรูปแบบการจัดการแบบรวมศูนย์ (Centralized System) โดยมีองค์ประกอบ คือ เครื่องผู้ให้บริการ (Server) และเครื่องผู้ใช้บริการ (Client) เชื่อมต่อกันอยู่ ในรูปแบบของ Client/Server นั้น เครื่องผู้ใช้บริการได้จะทำการติดต่อร้องขอบริการจากเครื่องผู้ให้บริการ ซึ่งเครื่องผู้ให้บริการก็จะดำเนินการจัดการตามที่เครื่องผู้ขอใช้บริการทำการร้องขอข้อมูล แล้วส่งข้อมูลที่มีการขอข้อมูลไว้กลับไปให้ ซึ่งประสิทธิภาพของ เครื่องผู้ให้บริการ (Server) จะต้องมีประสิทธิภาพที่สูงขึ้น และมีทรัพยากรมากขึ้นจะแปรผันตามจำนวนเครื่องผู้ใช้บริการ (Client) เพื่อให้รองรับกับจำนวนของเครื่องผู้ใช้บริการ (Client) และเพียงพอต่อการให้บริการได้อย่างมีประสิทธิภาพ



รูปที่ 1.1 สถาปัตยกรรมแบบ Client/Server

สำหรับเทคโนโลยีที่มีการพัฒนามาจนถึงปัจจุบันเป็นเทคโนโลยีระบบคลาวด์คอมพิวติง (Cloud Computing) ตามรูปที่ 1.2 ซึ่งเป็นเทคโนโลยีที่ใช้วิธีการประมวลผลโดยอ้างอิงความต้องการของผู้ใช้ โดยระบบจะจัดสรรทรัพยากรและการให้บริการที่ตรงกับความต้องการของผู้ใช้งาน โดยระบบสามารถยืดหยุ่น เพิ่มและลดจำนวนทรัพยากร เพื่อให้เหมาะสมกับความต้องการของผู้ใช้งานได้ตลอดเวลา โดยผู้ใช้งานไม่จำเป็นต้องทราบการทำงานของระบบเบื้องหลังว่าเป็นอย่างไร



รูปที่ 1.2 สถาปัตยกรรมแบบ คลาวด์คอมพิวติง

เทคโนโลยีระบบคลาวด์คอมพิวติง ที่มีในปัจจุบันเป็นทางเลือกที่ดีและเป็นโอกาสในการพัฒนาซอฟต์แวร์ประเภทระบบการบริหารงานบำรุงรักษา (CMMS) โดยอาศัยความสามารถของ

ระบบคลาวด์คอมพิวติง ให้เป็นประโยชน์ เพื่อเพิ่มประสิทธิภาพในการใช้งาน และยังประโยชน์ให้อุตสาหกรรมหรือธุรกิจขนาดเล็กไปจนถึงขนาดกลางมีโอกาสได้ใช้งาน ซอฟต์แวร์ระบบการบริหารงานบำรุงรักษา อีกทั้งยังรองรับการเพิ่มขยายขนาดของธุรกิจที่จะเติบโตขึ้นได้โดยที่ไม่ต้องทำการปรับเปลี่ยนโครงสร้างสถาปัตยกรรมระบบแต่อย่างไร [1, 2]

1.2 วัตถุประสงค์การวิจัย

1. ศึกษาเกี่ยวกับกรรมวิธี และรูปแบบสถาปัตยกรรมคลาวด์คอมพิวติง
2. เพื่อออกแบบสถาปัตยกรรมคลาวด์คอมพิวติง สำหรับระบบการบริหารงานบำรุงรักษาโดยใช้ซอฟต์แวร์ผลิตภัณฑ์ที่มีอยู่มาปรับเข้ากับเทคโนโลยีคลาวด์

1.3 กรอบแนวคิดการวิจัย

1. SaaS (Software as a Service)

SaaS หรือ ซอฟต์แวร์ในรูปแบบของบริการ เป็นรูปแบบหนึ่งของการส่งมอบบริการ (Delivery Models) บนคลาวด์คอมพิวติง (ดูคำอธิบายเพิ่มเติมในหัวข้อ 2.2.2) วิทยานิพนธ์นี้จะใช้รูปแบบ SaaS เป็นหลักในการพัฒนาคลาวด์เซอร์วิสของระบบ CMMS บนคลาวด์เป็นต้นแบบ

2. สถาปัตยกรรมของอินเตอร์คลาวด์ (InterCloud) [3]

สถาปัตยกรรมและส่วนประกอบของอินเตอร์คลาวด์ (InterCloud) เป็นสถาปัตยกรรมคลาวด์คอมพิวติงแบบรวมกลุ่มพันธมิตรเพื่อการให้บริการโปรแกรมประยุกต์ที่รองรับการขยายตัวได้ (ดูคำอธิบายเพิ่มเติมในหัวข้อ 2.3.2) วิทยานิพนธ์นี้จะใช้โครงสร้างต้นแบบจากสถาปัตยกรรมของอินเตอร์คลาวด์ เพื่อรองรับการขยายตัวของผู้ให้บริการคลาวด์

1.4 ขอบเขตของการวิจัย

1. โครงสร้างสถาปัตยกรรมของซอฟต์แวร์ที่มีอยู่ซึ่งมีรูปแบบการทำงานเป็น Client/Server
2. เฉพาะซอฟต์แวร์ CMMS ที่ใช้ในการบำรุงรักษาเครื่องจักรแบบ Single Site

1.5 นิยามศัพท์เฉพาะ

1. คลาวด์คอมพิวติง (Cloud Computing) - รูปแบบของการพัฒนา การนำไปใช้ (Deployment) และการส่งมอบ (Delivery) บริการไอทีที่เกิดขึ้นใหม่ ซึ่งช่วยให้สามารถมีการส่งมอบผลิตภัณฑ์ บริการ และโซลูชัน ต่างๆ แบบ Real-Time บนอินเตอร์เน็ต (คือ ทำให้มีคลาวด์เซอร์วิส)

2. คลาวด์เซอร์วิส (Cloud Services) - ผลิตภัณฑ์ บริการ และโซลูชัน ทางธุรกิจ ที่ถูกส่งมอบและบริโภคผ่านทางอินเตอร์เน็ตแบบ Real-Time

3. เซอร์วิส - ในวิทยานิพนธ์นี้ หมายถึง บริการบนคลาวด์คอมพิวติง ใช้สลับกับคำว่า “บริการ” ตามความเหมาะสม เพื่อแยกแยะกับบริการอื่น ๆ ที่ไม่อยู่บนคลาวด์

4. สถาปัตยกรรม SOA - หมายถึง สถาปัตยกรรมระบบเชิงบริการ (Service-Oriented Architecture) ซึ่งเป็นแพทเทิร์นของการออกแบบซึ่งคอมโพเนนท์ของแอปพลิเคชันมีการให้บริการแก่คอมโพเนนท์อื่น มีการสื่อสารระหว่างคอมโพเนนท์ของแอปพลิเคชัน ผ่านทางเครือข่าย [4]

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้แนวคิดการพัฒนาซอฟต์แวร์บนคลาวด์และการเปลี่ยนเทคโนโลยี

2. องค์กรผู้ใช้งานซอฟต์แวร์ CMMS มีทางเลือกมากขึ้น เป็นคลาวด์เซอร์วิส ซึ่งจะช่วยในการประหยัดค่าใช้จ่าย และความยืดหยุ่นในการเพิ่มและลดจำนวนทรัพยากรของระบบ (Scalability)

บทที่ 2
ทบทวนวรรณกรรม

2.1 CMMS ในปัจจุบัน

สำหรับโรงงานอุตสาหกรรม ซอฟต์แวร์ประเภทระบบการบริหารงานบำรุงรักษา (CMMS) ได้มีการพัฒนาเพื่อช่วยในการบริหารงานซ่อมบำรุง และเข้าถึงข้อมูลประวัติการทำงานของเครื่องจักร เครื่องมือ หรือ อุปกรณ์ ประวัติการซ่อมบำรุงรักษา รวมถึงการจัดตารางงานการซ่อมบำรุงเพื่อเป็นแผนงานการปฏิบัติงาน ซึ่งถือเป็นเครื่องมือที่ช่วยให้หน่วยงานซ่อมบำรุง สามารถทำงานในการซ่อมบำรุงได้อย่างเป็นระบบ โดยมีเป้าหมายเพื่อเพิ่มประสิทธิภาพของเครื่องจักร เครื่องมือ หรืออุปกรณ์ ให้สามารถทำงานได้อย่างเกิดประสิทธิภาพสูงสุด โดยข้อมูลทางด้านการซ่อมบำรุงนี้สามารถจะนำมาวิเคราะห์หาทางป้องกันในครั้งหน้า เพื่อนำไปปรับวิธีการซ่อมบำรุง และกำหนดแผนงานการซ่อมบำรุงเชิงป้องกัน

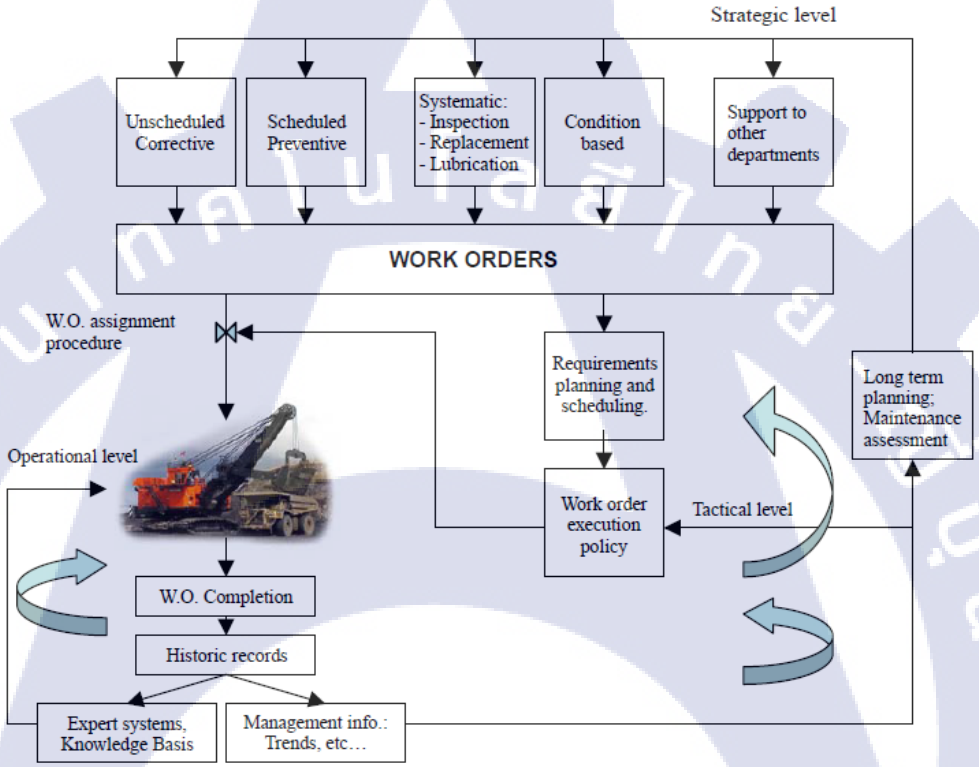
งานวิจัยของ Marquez & Gupta [5] ได้ทบทวนแนวคิดต่างๆ กระบวนการ และมาตรฐานต่างๆ ที่เกี่ยวของกับการบำรุงรักษาและการบริหารการบำรุงรักษา เนื่องจากการบริหารการบำรุงรักษามีความซับซ้อนมากขึ้น และเสนอกรอบแนวคิดของหน้าที่การบำรุงรักษาในองค์กร โดยมีการใช้ปัจจัยทางด้านคุณลักษณะต่างๆ (เช่น ระบบสารสนเทศที่ใช้ในการบริหารการบำรุงรักษา เทคโนโลยีกระบวนการและการบูรณาการ (Process Technology and Integration) และระบบบริหารการผลิต อย่าง JIT เป็นต้น) ซึ่งมีผลต่อความซับซ้อนในการบำรุงรักษาการดำเนินงาน งานวิจัยนี้ได้วิเคราะห์ทางด้านกลยุทธ์ ยุทธวิธี และทางด้านการดำเนินงานของการบำรุงรักษา แล้วจัดทำโครงสร้างเพื่อช่วยให้การทำงานสมบูรณ์ทั้งสามระดับดังกล่าว

รูปที่ 2.1 แสดงกรอบแนวคิดดังกล่าว โดยในที่นี้ Process หมายถึง กิจกรรมหรือการทำงานที่สัมพันธ์กับการบำรุงรักษา และ Framework ครอบคลุมโครงสร้างพื้นฐานสนับสนุน (Supporting Infrastructure) กรอบแนวคิดนี้จัดการบริหารการบำรุงรักษาเป็น 3 ระดับ คือ ระดับกลยุทธ์ (Strategic Level) ระดับยุทธวิธี (Tactical Level) และระดับปฏิบัติการ (Operational Level)

2.2 คลาวด์คอมพิวติง

คลาวด์คอมพิวติง เริ่มแพร่หลายในปัจจุบันมากยิ่งขึ้น ซึ่งเกิดจากการเปลี่ยนแปลงจากเดิมที่เป็นรูปแบบสถาปัตยกรรมคอมพิวเตอร์จากเมนเฟรม ที่มีการประมวลผลที่เครื่องส่วนกลาง มาเป็นสถาปัตยกรรมแบบ Client/Server ซึ่งมีการแบ่งการประมวลผลบางส่วนมาไว้ที่ Client บางส่วนไว้ที่

Server จนมาถึงในปัจจุบันเป็นสถาปัตยกรรมคลาวด์คอมพิวติง ซึ่งมีคุณลักษณะเฉพาะที่มีความยืดหยุ่นสูงกว่าในการเพิ่มและลดจำนวนทรัพยากรของระบบตามความต้องการที่จะใช้งานจริง และยังลดต้นทุนในการบริหารจัดการทรัพยากรของผู้ใช้งานไปเป็นภาระรับผิดชอบของผู้ให้บริการคลาวด์คอมพิวติงแทน ผู้ใช้งานจะสามารถใช้งานโดยยังคงมีกระบวนการทำงานที่เหมือนเดิมได้ โดยไม่จำเป็นต้องทราบว่ามีการใช้งานทรัพยากรมากน้อยเพียงใด และเหลือเพียงพอที่จะใช้งานต่อไปหรือไม่



รูปที่ 2.1 กระบวนการบำรุงรักษา (Maintenance Process) แนวทางการดำเนินการ และการปฏิบัติการสะท้อนกลับ (Feedback) ในสามระดับของกิจกรรมทางธุรกิจ [5]

2.2.1 คลาวด์คอมพิวติง

โดยทั่วไปมีการนิยามว่า คลาวด์คอมพิวติง เป็นโมเดลที่อำนวยความสะดวกในการใช้งานทรัพยากรร่วมกัน โดยมีความยืดหยุ่นในการแบ่งทรัพยากรให้มากน้อยตามความต้องการของผู้ใช้งาน

NIST [6] ได้นิยาม คลาวด์คอมพิวติงไว้ว่า

“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g.,

networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." [7]

คลาวด์คอมพิวติง เป็นรูปแบบที่จะเปิดใช้งานแพร่หลาย เพื่อความสะดวก การเข้าถึงในเวลาที่ต้องการไปยังเครือข่ายของทรัพยากรคอมพิวเตอร์ (ซึ่งคอนฟิกได้) ที่องค์กรและบุคคลร่วมกันใช้งานได้ (เช่น เครือข่าย เซิร์ฟเวอร์ ระบบจัดเก็บข้อมูล แอปพลิเคชัน และบริการต่างๆ) ที่สามารถจัดเตรียมให้ผู้รับบริการได้อย่างรวดเร็ว และเปิดให้บริการด้วยต้นทุนและความพยายามของการบริหารจัดการ หรือภาระการดำเนินการของผู้ให้บริการ น้อยที่สุด

2.2.2 รูปแบบการให้บริการคลาวด์คอมพิวติง

การให้บริการคลาวด์มีรูปแบบของการส่งมอบบริการ (Delivery Models) 3 แบบ ดังนี้ [7, 8, 9] (ดูรูปที่ 2.2)

(1) Infrastructure as a Service (IaaS)

IaaS หรือ *โครงสร้างพื้นฐานในรูปแบบของบริการ* เป็นการให้บริการโครงสร้างพื้นฐานที่เป็นทรัพยากรที่จำเป็น รวมถึง พื้นที่ในการจัดเก็บข้อมูล การเชื่อมต่อเครือข่าย หน่วยความจำ หน่วยประมวลผลกลาง ทรัพยากรอื่นๆ ทางด้านฮาร์ดแวร์ที่จำเป็น

ตัวอย่างของ IaaS ได้แก่ Google Compute Engine [10], Amazon EC2 [11], Amazon S3 [12], Rackspace Cloud Servers [13] และ FlexiScale [14] เป็นต้น

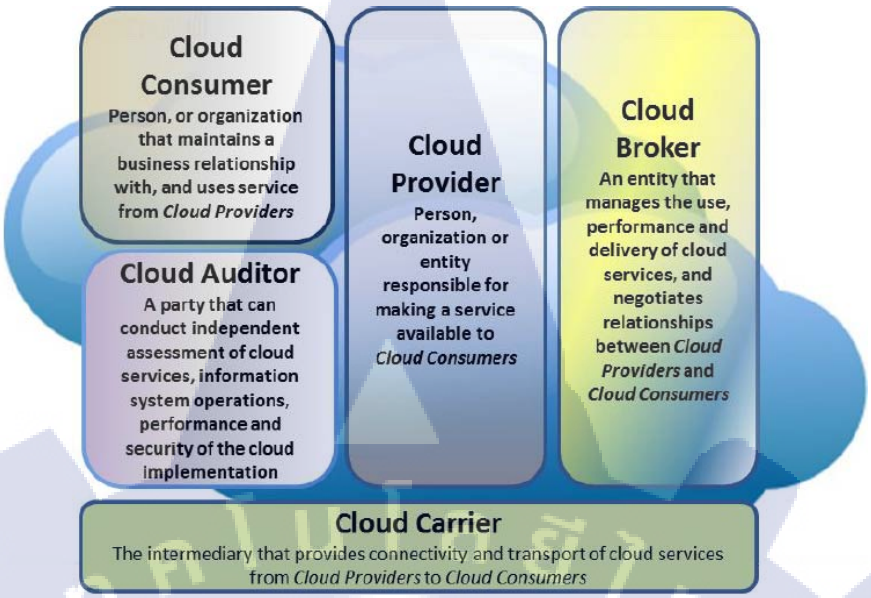
(2) Platform as a Service (PaaS)

PaaS หรือ *แพลตฟอร์มในรูปแบบของบริการ* เป็นการให้บริการเครื่องมือในการให้บริการในส่วนการสร้างโปรแกรมประยุกต์เพื่อสนับสนุน และเป็นเครื่องมือในการพัฒนาซอฟต์แวร์

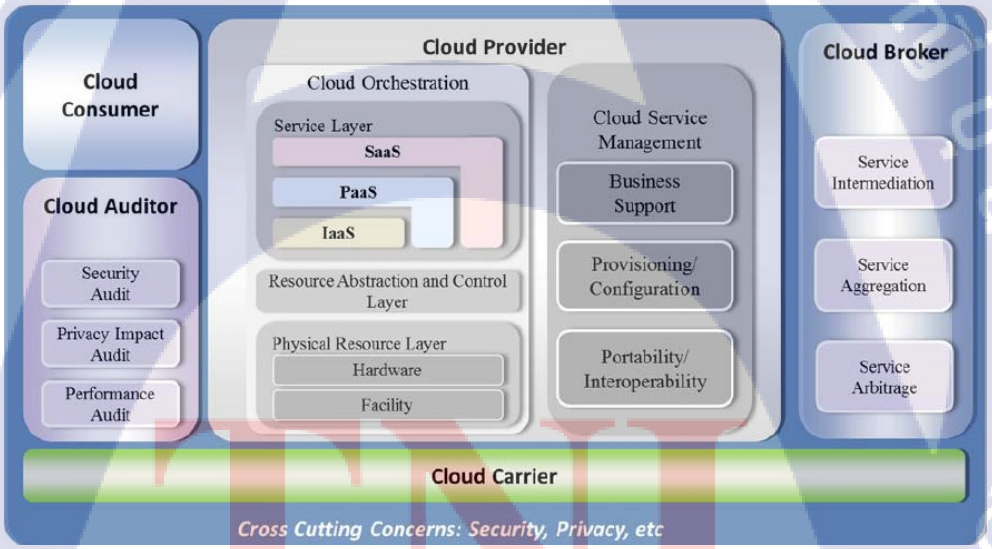
ตัวอย่างของ PaaS ได้แก่ Google App Engine [15], Apache Stratos [16], Microsoft Azure [17], Heroku [18] และ Salesforce [19] เป็นต้น

(3) Software as a Service (SaaS)

SaaS หรือ *ซอฟต์แวร์ในรูปแบบของบริการ* เป็นการให้บริการซอฟต์แวร์ที่มีการติดตั้งอยู่บนโครงสร้างพื้นฐานคลาวด์ ได้แก่ เครือข่าย เซิร์ฟเวอร์ ระบบปฏิบัติการ ระบบจัดเก็บข้อมูล และความสามารถของแอปพลิเคชันต่างๆ เป็นต้น ซึ่งจัดเตรียมโดยผู้ให้บริการคลาวด์คอมพิวติง ผู้ใช้บริการสามารถจะเข้าถึงบริการซอฟต์แวร์หรือแอปพลิเคชันเหล่านั้นได้ด้วยอุปกรณ์ต่างๆ ที่มีอินเตอร์เฟสของธินไคลเอนต์ (Thin Client Interface) เช่น เว็บบราวเซอร์ โดยผู้ใช้ไม่จำเป็นต้องบริหารและควบคุมโครงสร้างพื้นฐานคลาวด์นั้นเอง สำหรับบริการ SaaS สามารถแบ่งออกได้เป็น 2 กลุ่มบริการ ดังนี้



รูปที่ 2.2 (ก) นิยาม Actor หรือองค์กรหรือบุคคลผู้ร่วมดำเนินการบนคลาวด์



รูปที่ 2.2 (ข) Conceptual Reference Model (แบบจำลองอ้างอิงเชิงแนวคิดคลาวด์คอมพิวติง)

รูปที่ 2.2 แบบจำลองอ้างอิงเชิงแนวคิดของคลาวด์คอมพิวติง และนิยามผู้มี ของ NIST [7]

1. บริการระดับองค์กร (Enterprise Services) เช่น บริการจัดการเวร์กโฟลว์ (Workflow Management) บริการทางด้านห่วงโซ่อุปทาน (Supply Chain) บริการจัดการลูกค้าสัมพันธ์ (Customer Relationship Management: CRM) บริการซอฟต์แวร์ทางการเงิน และบริการค้นหา (Search) เป็นต้น

2. แอปพลิเคชันในรูปของเว็บ 2.0 เช่น บริการจัดการเมทาเดต้า (Metadata Management) บริการเครือข่ายสังคม (Social Networking) บริการเว็บบล็อก บริการเว็บวิกิ และบริการเว็บพอร์ทัล (Portal Services) เป็นต้น

ตัวอย่างของ SaaS ได้แก่ Gmail, Google Apps, Apple iCloud, YouTube และ Facebook เป็นต้น

2.2.3 เทคโนโลยีที่เกี่ยวข้อง (Related Technologies)

(1) กริดคอมพิวติง (Grid Computing)

เป็นการกระจายการประมวลผลโดยแบ่งปันทรัพยากรในเครือข่ายให้ช่วยกันในการประมวลผลให้สำเร็จ โดยความคล้ายคลึงกันของ คลาวด์คอมพิวติง และกริดคอมพิวติง คือ มีการแบ่งปันการใช้ทรัพยากร

(2) เวอร์ชวลไลเซชัน (Virtualization)

เป็นระบบจำลองเครื่องเสมือน โดยทำงานบนเครื่องเสมือน และใช้งานทรัพยากรเสมือน ซึ่งจะสนับสนุนการทำงานของซอร์ฟแวร์ โดยไม่มีการแสดงให้เห็นถึงรายละเอียดการใช้งานทางด้านกายภาพของฮาร์ดแวร์แต่อย่างใด

(3) ออโตนอมิกคอมพิวติง (Autonomic Computing)

เป็นการสร้างระบบประมวลผลที่สามารถจัดการความต้องการทรัพยากรต่างๆ ที่ให้บริการได้ด้วยตัวเองแบบทันทีทันใด ซึ่งจะมีข้อแตกต่างจาก คลาวด์คอมพิวติง คือ Autonomic Computing มีเป้าหมายที่จะลดระบบลง แต่คลาวด์คอมพิวติงมีเป้าหมายในการลดต้นทุนทางด้านทรัพยากร

2.2.4 ประเภทของคลาวด์คอมพิวติง

คลาวด์คอมพิวติงสามารถแบ่งออกเป็น 4 ประเภท ตามรูปแบบการนำไปใช้งาน (Deployment Models) ดังนี้ [8]

(1) คลาวด์แบบส่วนตัว (Private Cloud)

เป็นการใช้บริการโดยใช้ในองค์กรของผู้ใช้งานเอง แบบเป็นส่วนตัว แต่เพียงผู้เดียว โดยที่ผู้ใช้งานจะใช้งานและบริหารจัดการบนเครื่องแม่ข่ายที่เป็นของตัวเอง หรือจากผู้ให้บริการรายอื่นก็ได้

(2) คลาวด์แบบสาธารณะ (Public Cloud)

เป็นการใช้บริการโครงสร้างพื้นฐานจากผู้ให้บริการ โดยผู้ให้บริการจะให้บริการแบบสาธารณะ โดยมีการแบ่งส่วนให้ผู้ใช้งานได้ใช้งานตามความต้องการ และผู้ใช้งานจะมีการชำระค่าบริการตามปริมาณที่มีการใช้งานจริง

(3) คลาวด์แบบชุมชน (Community Cloud)

เป็นการแบ่งการใช้งานทรัพยากรพื้นฐานจากกลุ่มที่รวมตัวกันเพื่อแบ่งทรัพยากรระหว่างกันไว้ และนำมาให้ผู้ใช้งานได้ใช้งานในส่วนนี้ โดยโครงสร้างพื้นฐานนี้จะถูกจัดการโดยองค์กรต่างๆ ที่เข้าร่วมอยู่ในกลุ่ม

(4) คลาวด์แบบผสม (Hybrid Cloud)

การใช้งานตามโครงสร้างพื้นฐาน โดยเป็นการใช้งานประเภทของคลาวด์คอมพิวติงคลาวด์แบบส่วนตัว หรือ คลาวด์แบบสาธารณะ หรือ คลาวด์แบบชุมชน ร่วมกันอย่างน้อย 2 ประเภท เพื่อให้เกิดความเหมาะสมสูงสุด ซึ่งพิจารณาการแบ่งการใช้งานตามความสำคัญของข้อมูล โดยข้อมูลบางส่วนอาจจะอยู่บนคลาวด์แบบส่วนตัว เพราะว่าเป็นข้อมูลที่มีความสำคัญสูงสำหรับหน่วยงาน และแบ่งข้อมูลบางส่วนไว้บนคลาวด์แบบสาธารณะ หรือ คลาวด์แบบชุมชน เพื่อให้เหมาะสมและประหยัดทรัพยากร

2.2.5 ประโยชน์ของคลาวด์คอมพิวติง

คลาวด์คอมพิวติงให้ประโยชน์ต่อองค์กรทางด้านธุรกิจอุตสาหกรรม ได้แก่

(1) การขยายขนาด และความยืดหยุ่น (Scalability and Flexibility)

คลาวด์คอมพิวติงสามารถขยายขนาดเพื่อรองรับการใช้งานขนาดใหญ่ และยืดหยุ่นให้เพิ่มลดการใช้งานทรัพยากรได้ตามการใช้งานจริงที่เกิดขึ้น

(2) การเข้าถึงเครือข่าย (Broad Network Access)

คลาวด์คอมพิวติงสนับสนุนให้ผู้ใช้สามารถเข้าถึงโดยใช้อุปกรณ์ได้หลากหลาย ได้แก่ คอมพิวเตอร์ส่วนบุคคล คอมพิวเตอร์แบบพกพา หรืออุปกรณ์สมาร์ทโฟน

(3) วิธีชำระค่าบริการตามการใช้งาน (Pay Per Use)

ผู้ให้บริการคลาวด์จะจัดเตรียมทรัพยากรไว้ให้เพียงพอต่อการใช้งานของผู้ใช้งาน โดยผู้ใช้งานจะชำระค่าบริการตามการใช้งานที่มีใช้ไปเท่านั้น

(4) การลดต้นทุน (Cost Effectiveness)

จากเดิมที่ผ่านมาองค์กรจะต้องมีการลงทุนทางด้านฮาร์ดแวร์เพื่อให้รองรับได้เพียงพอต่อการดำเนินธุรกิจ และเป็นไปตามเทคโนโลยีที่มีการปรับเปลี่ยนอยู่เสมอ การใช้งานคลาวด์คอมพิวติงสามารถจะช่วยลดภาระต้นทุนทางด้านนี้และให้เปลี่ยนไปเป็นหน้าที่ของผู้ให้บริการแทน โดยองค์กรที่ใช้งานก็จะมีต้นทุน เพียงค่าบริการตามการใช้งานที่จำเป็นต้องใช้เพียงเท่านั้น

2.2.6 การโยกย้ายระบบไปบนคลาวด์คอมพิวติง

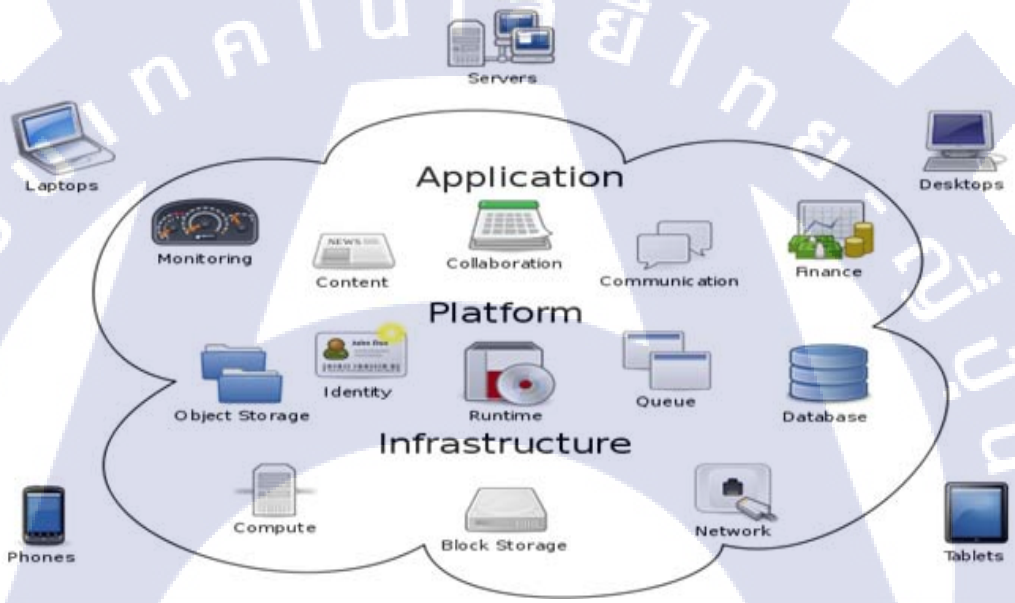
ในการยกระดับระบบให้เป็นคลาวด์เซอร์วิส องค์กรสามารถจะ Deploy ระบบซอฟต์แวร์ขององค์กรบน ทรัพยากรต่างๆ อย่างไรก็ตาม องค์กรต้องพึงพาระบบที่สำคัญๆ (Business-Critical Systems) ซึ่งมีการพัฒนามาเป็นเวลานาน แอปพลิเคชันเก่าเหล่านี้ก็ยังคงถูกใช้งานอยู่ได้เป็นเวลานานตามที่เคยตั้งกฎเกณฑ์ของระบบและกระบวนการเดิม แม้มีงานวิจัยต่างๆ เกี่ยวกับการโยกย้ายระบบไปเป็นคลาวด์ (Cloud Migration) งานวิจัยของ Jamshidi และคณะ [20] จำแนกคำศัพท์และการตีความ (Taxonomy) และเปรียบเทียบ เกี่ยวกับการโยกย้ายระบบไปเป็นคลาวด์โดยทบทวนวรรณกรรมอัตโนมัติ (Systematic Literature Review) กับงานวิจัย 23 งานที่ตีพิมพ์ในปี ค.ศ.2010 – 2013 มีการเปรียบเทียบตามกรอบแนวคิดการกำหนดคุณลักษณะเฉพาะ (Characterization Framework) ผลลัพธ์ของจากการสังเคราะห์งานวิจัยและการศึกษาดังกล่าวเป็นฐานความรู้ของโซลูชันปัจจุบัน ในการโยกย้ายระบบเก่าไปบนคลาวด์คอมพิวติง (Legacy-To-Cloud Migration) งานวิจัยของ Jamshidi และคณะ [20] พบว่า การวิจัยเกี่ยวกับการโยกย้ายระบบเก่าไปบนคลาวด์คอมพิวติงยังคงอยู่ในขั้นเริ่มต้น แต่กำลังมีความก้าวหน้า และสรุปว่า ยังคงมีความต้องการการวิจัยเพื่อค้นหากรอบแนวคิดการโยกย้ายระบบ (Migration Framework) เพื่อช่วยปรับปรุงระดับความสามารถและความไว้วางใจของการโยกย้ายระบบไปบนคลาวด์

2.3 การพัฒนาซอฟต์แวร์บนคลาวด์

2.3.1 การพัฒนาซอฟต์แวร์และวิศวกรรมซอฟต์แวร์

Singh และ Chana [2] ได้ศึกษาแนวทางการพัฒนาซอฟต์แวร์บนคลาวด์ ในรูปที่ 2.3 โดยประยุกต์ใช้แนวคิดของแนวทางการพัฒนาซอฟต์แวร์แบบเอจายล์ (Agile Software Development) [21] สนใจทางด้านวัฏจักรของการพัฒนาที่จะเกิดเป็นช่วงสั้นๆ ในการออกแบบ พัฒนา และทดสอบเป็นอย่างนี้ซ้ำๆ โดยในระหว่างกระบวนการต่างๆ ก็จะทำให้เกิดความรู้ในการที่จะใช้ในการแก้ไขปัญหาซ้ำๆ จากปัจจัยแวดล้อมที่มีผลกระทบ เช่น สภาวะแวดล้อมที่จะพัฒนา เทคโนโลยีที่มี ความต้องการ ผู้มีส่วนร่วมในการพัฒนา ผู้ขายซอฟต์แวร์ เป็นต้น นำมาซึ่งประสบการณ์ในการแก้ไขปัญหาที่เกิดขึ้นได้ เป็นสถาปัตยกรรมที่สามารถนำมาใช้ใหม่ได้โดยพัฒนามาจากรูปแบบสถาปัตยกรรมที่สามารถนำมาใช้ใหม่ งานวิจัยของ Singh และ Chana นี้เสนอรูปแบบการพัฒนาคลาวด์แบบปรับตัว

ได้ (Adaptive Cloud Development (ASD) Model) โดยคาดหวังความรวดเร็วในการนำแอปพลิเคชันไปใช้บนระบบคลาวด์ (Speed of Application Deployment) การใช้เวลาการเข้าสู่ตลาด (Time To Market) สั้นลง การลดค่าใช้จ่ายการดำเนินงาน ความสามารถในการนำ คอมโพเนนท์บนคลาวด์มาใช้ซ้ำ (Reusability) และความง่ายต่อการบำรุงรักษาสำหรับผู้เช่าหรือผู้รับบริการ งานวิจัยเสนอรูปแบบการพัฒนา ASD พร้อมกับบูรณาการการปฏิสัมพันธ์กับผู้บริการคลาวด์ (Cloud Provider) ในการยอมรับกันได้บนคลาวด์คอมพิวติง มุ่งให้รูปแบบ ASD เป็นแนวทางในการพัฒนาคลาวด์ตามแนวคิดของเอจายล์ เพื่อที่จะให้ต้นทุนการพัฒนาต่ำสุด และสามารถปรับปรุงความพึงพอใจของลูกค้าให้ดีขึ้น และเพิ่มการนำคอมโพเนนท์บนคลาวด์มาใช้ซ้ำ (Reusability) ได้

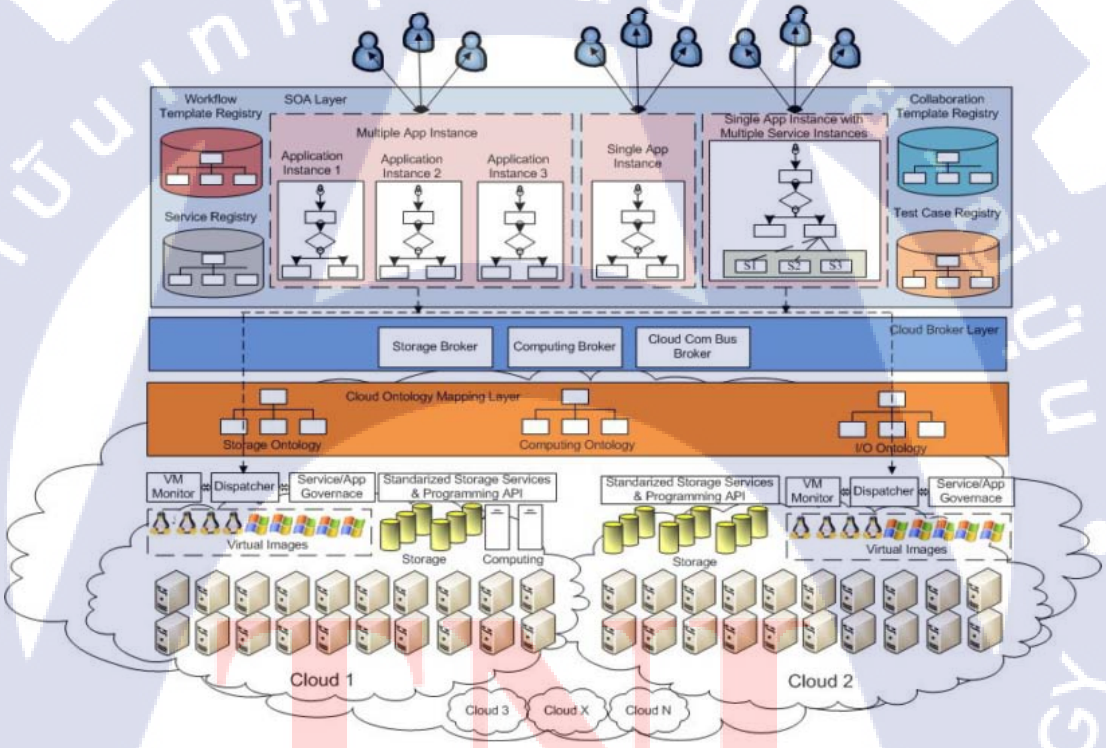


รูปที่ 2.3 คลาวด์คอมพิวติง และบริการ [2]

2.3.2 สถาปัตยกรรมระบบที่เกี่ยวข้อง

ในด้านการพัฒนาและใช้คลาวด์ในระบบโรงงานอุตสาหกรรม Tsai และคณะ [22] ได้ทำสำรวจสถาปัตยกรรมคลาวด์คอมพิวติงในภาพรวม ถกปัญหาเกี่ยวกับการอิมพลีเมนต์บนคลาวด์ และนำเสนอสถาปัตกรรมบนคลาวด์คอมพิวติงที่เน้นการบริการ เรียกว่า SOCCA (Service-Oriented Cloud Computing Architecture) เพื่อให้คลาวด์แต่ละคลาวด์สามารถทำงานประสานร่วมกับ (Interoperation) คลาวด์อื่นได้ นอกจากนี้ SOCCA เสนอการออกแบบในระดับสูง (High Level Designs) ที่ให้การสนับสนุนที่ดีขึ้นในคุณสมบัติด้านความสามารถที่ผู้ใช้หลายรายใช้แอปพลิเคชันเดียวกันได้ (Multi-Tenancy Feature) ของคลาวด์คอมพิวติง สถาปัตยกรรมระบบที่เสนอโดย

งานวิจัยนี้เอื้ออำนวยให้แอปพลิเคชันหนึ่งๆ รันบนคลาวด์ได้หลายคลาวด์และทำงานประสานร่วม (Interoperate) กับคลาวด์อื่นๆ รูปที่ 2.4 แสดงสถาปัตกรรม SOCCA ซึ่งแบ่งเป็น 4 ชั้น เพื่อสนับสนุนทั้ง SOA และคลาวด์คอมพิวติง โดยอ้างว่า SOCCA จะสนับสนุนการโยกย้ายแอปพลิเคชัน (Application Migration) จากคลาวด์หนึ่งไปยังอีกคลาวด์หนึ่ง และการให้บริการโดย Deploy ซ้ำ (Service Redeployment) บนคลาวด์หลายระบบได้ วิธีของ SOCCA คือ การแบ่งแยกบทบาทหน้าที่ของผู้ให้บริการตรรกะของเซอร์วิส (Service Logic Provider) และผู้ให้บริการคลาวด์หรือโฮสติง (Service Hosting/Cloud Providers) โดยมุ่งเน้นให้เป็นแพลทฟอร์มเปิด (Open Platform) ที่สามารถควบคุมและบำรุงรักษาได้ด้วยมาตรฐานเปิด (Open Standards) และออนโทโลยี (Ontology)



รูปที่ 2.4 สถาปัตยกรรม SOCCA [22]

ในมุมมองทางด้านธุรกิจ มีโครงการหลายโครงการและหลายงานวิจัยที่ได้เฝ้าสังเกตการณ์การประกอบกระบวนการการผลิต (Composition of Manufacturing Processes) จากเซอร์วิสต่างๆ ที่แทนสินทรัพย์ทางการผลิตเดี่ยว หรือกระบวนการจริงในการผลิต (Single Manufacturing Assets or Real-World Manufacturing Processes) อาทิเช่น การประยุกต์ใช้แนวคิด “Virtual Factories” หรือ “Virtual Manufacturing Enterprises” โดยทั่วไปตามหลักการแล้ว ถ้าประยุกต์

การประกอบระบบจากบริการต่าง (Service-Based Composition) กับกระบวนการผลิต กระบวนการเหล่านั้นสามารถจะอยู่บนสถาปัตยกรรม SOA วิธีการค้นพบเมื่อไม่นานนี้นำกระบวนการผลิตเชิงบริการ (Service-Oriented Manufacturing Processes) ไปใช้ในระดับถัดไปโดยเสนอให้โยกจาก สาขาคลาวด์คอมพิวติง ไปยังกระบวนการผลิต (Real-World Manufacturing Processes) แล้วให้สนับสนุนกระบวนการเหล่านี้โดย ซอฟต์แวร์บนคลาวด์คอมพิวติงและโครงสร้างพื้นฐานของไอที (IT Infrastructure) วิธีการนี้เรียกกันว่า การผลิตและสนับสนุนบนคลาวด์ (Cloud Manufacturing and Supports) [23]

Schulte และคณะ ได้กำหนดความต้องการของกรอบแนวคิดซอฟต์แวร์ สำหรับ Cloud Manufacturing โดยมีสมมติฐานว่า กรอบแนวคิดซอฟต์แวร์นั้นจำเป็นต้องสนับสนุนทุกเฟสของการบริหารกระบวนการธุรกิจ (BPM Lifecycle) (คือ การวิเคราะห์และออกแบบ การคอนฟิก การกำหนดกฎ และการประเมินผล ความต้องการของกรอบแนวคิดซอฟต์แวร์สามารถจะอธิบายโดยย่อได้ดังนี้

(1) จำเป็นต้องออกแบบแบบจำลองกระบวนการ (Process Models) และเตรียมให้มีแบบจำลองเหล่านั้น โดยปกติบริษัทผู้ผลิตจะมีแบบจำลองกระบวนการไว้ให้

(2) การคอนฟิกกระบวนการผลิต (Configuration of Manufacturing Processes) มีความจำเป็นที่จะอิมพลีเมนต์ขั้นตอนต่างๆ ของกระบวนการหนึ่ง บริการซอฟต์แวร์สามารถแทนขั้นตอนหนึ่ง ๆ ของกระบวนการที่มีคำอธิบายแหล่งข้อมูล (Data Sources (เช่น เซนเซอร์) การคอนฟิกมีการกำหนดตารางเวลา (The Scheduling and Resource Allocation of Services) ให้กับสินทรัพย์การผลิตและทรัพยากรทางการประมวลผลหรือคำนวณ (Computational Resources) กล่าวคือ กรอบแนวคิดซอฟต์แวร์และ BPMS ที่ใช้ในการจัดการกระบวนการจำเป็นต้องยืดหยุ่นและสามารถลดขยายได้ (Scalable)

(3) มีการเฝ้าระวังเหตุการณ์วิกฤต โดย Monitor ข้อมูลบริการซอฟต์แวร์ได้ในระดับขั้นตอนของกระบวนการ (เช่น การใช้ CPS หรือเซนเซอร์)

(4) มีการเก็บข้อมูลการทำงานเพื่อใช้ในการวิเคราะห์เพื่อปรับแบบจำลองกระบวนการได้

(5) BPMS สำหรับ Cloud Manufacturing จำเป็นที่จะต้องมีฐานความรู้เพื่อการตัดสินใจในแต่ละขั้นตอนของ Lifecycle ความรู้เกี่ยวกับทรัพยากรและสินทรัพย์ภายในสามารถถูกเก็บไว้ในฐานข้อมูลขององค์กร ในขณะที่ความรู้เกี่ยวกับสินทรัพย์ภายนอก อาจมีให้บริการโดย Marketplaces ซึ่งเป็นส่วนหนึ่งของระบบเว็บธุรกิจทางการผลิต (Manufacturing Business Web) ที่มีสภาพแวดล้อมคลาวด์ซึ่งสามารถนำองค์กรคู่ค้าในห่วงโซ่อุปทานมาพบกันโดยมีการจัดเตรียมโครงสร้างพื้นฐานคลาวด์ แอปพลิเคชัน คอนเทนต์ และการเชื่อมโยงถึงกัน ไว้ให้

นอกจากนี้ งานวิจัยของ Schulte และคณะ [23] ยังรวบรวมหัวข้อที่เกี่ยวข้องและน่าสนใจ มีความท้าทายต่อการวิจัยในอนาคต เกี่ยวกับเรื่องต่อไปนี้

(1) การอธิบายบริการและการจำลองกระบวนการ (Process Modeling and Service Descriptions)

(2) ตลาดการบริการ (Service Marketplaces)

(3) การเฝ้าติดตามกระบวนการ (Process Monitoring)

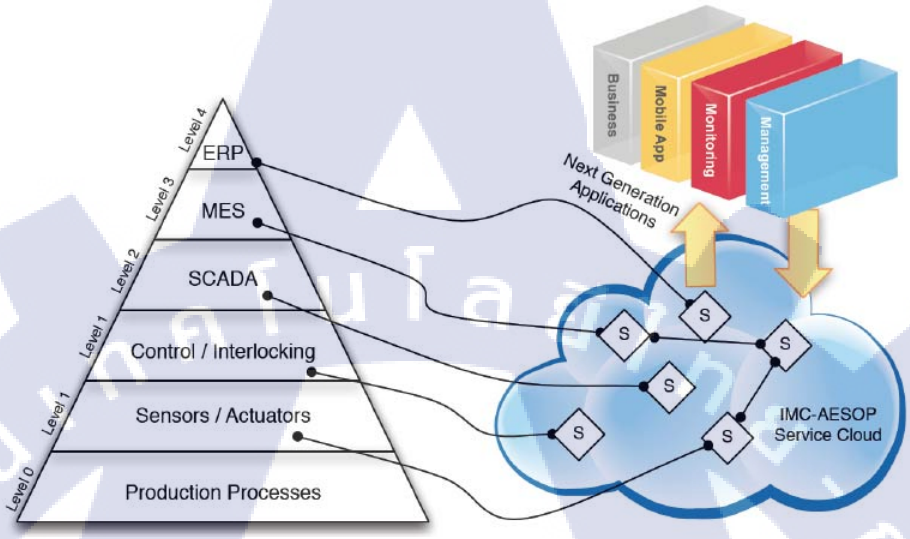
(4) ความวางใจและความปลอดภัยของข้อมูล (Trust and Data Security)

ในด้านระบบอัตโนมัติอุตสาหกรรม (Industrial Automation Systems) [24] ได้ชี้ให้เห็นถึงปัญหาทางด้านสถาปัตยกรรมระบบในเรื่องระบบอัตโนมัติอุตสาหกรรม และคาดการณ์ว่าระบบจะมีความซับซ้อนซึ่งจะช่วยให้ระบบมีกำลังความสามารถในยุคหน้าในด้านการนำไปประยุกต์และด้านเซอร์วิสต่างๆ โรงงานอุตสาหกรรมในอนาคตจะพึ่งพาระบบนิเวศขนาดใหญ่ของระบบ (Ecosystem of Systems) ซึ่งมีเกิดการร่วมประสานกันในระดับสูงและปริมาณมาก (“collaboration at large scale”) ทั้งนี้ก็สามารถคาดการณ์ได้ว่า อนาคตจะมีระบบอัตโนมัติแบบกระจาย ที่ชาญฉลาดมากขึ้น ใช้ซ้ำได้ ทนทานต่อความผิดพลาด (Fault-Tolerant) ซึ่งจะเกิดขึ้นใหม่ๆ ในด้านกำลังความสามารถ (Capabilities, Functionalities and Structural Characteristics) เป็นเซอร์วิสบนคลาวด์

ในปัจจุบัน โรงงานอุตสาหกรรมต่างๆ มีส่วนประกอบและโครงสร้างของหลายระบบที่ทำงานร่วมกันในลักษณะลำดับชั้น (Hierarchy) ตามข้อกำหนดของสถาปัตยกรรมแบบองค์กรที่มีมาตรฐาน (Standard Enterprise Architectures) แต่เนื่องจากมีการใช้สถาปัตยกรรมแบบ SOA อยู่ด้วย หน้าที่ของระบบที่มีการออกแบบไว้ให้แต่ละระบบหรือแต่ละอุปกรณ์สามารถถูกเสนอให้ใช้เป็นบริการเดียวหรือหลาย ๆ บริการ แตกต่างกันในด้านความซับซ้อน ซึ่งอาจถูกจัดเก็บหรือโฮสต์ไว้ในคลาวด์ และประกอบกันโดยบริการอื่น (ซึ่งเป็นไปได้ว่า ข้ามชั้นของสถาปัตยกรรม) ก็ได้ ดังแสดงในรูปที่ 2.5

โดยทั่วไป ระบบการผลิต (โรงงานและกระบวนการ) ในปัจจุบันถูกจัดโครงสร้างเป็นลำดับชั้นในรูปแบบลำดับชั้น 5 ชั้น (5-Level Hierarchical Model) มาตรฐานสากล IEC 62264 [25] ยังกำหนดรูปแบบของการบริหารการดำเนินงานการผลิต (Manufacturing Operations Management Model) มาตรฐานนี้กำหนดหน้าที่ต่างๆ ซึ่งอยู่ในลำดับชั้นที่ 3 และ 4 และสิ่งต่างๆ ที่มีการแลกเปลี่ยนกันได้ คุณลักษณะเฉพาะและคุณสมบัติ กิจกรรม และหน้าที่ (Characteristics and Attributes, Activities and Functions) ที่เกี่ยวข้องกับการบริหารโรงงานอุตสาหกรรมหนึ่งๆ ในทางปฏิบัติของการนำไปใช้จะขึ้นอยู่กับความต้องการของลูกค้าแต่ละรายและกลยุทธ์ของผู้ผลิตเครื่องมืออุตสาหกรรม ยกตัวอย่างเช่น การดำเนินงานบริหารการบำรุงรักษาโดยทั่วไปอาจถูก

มอบหมายให้ใช้กับระบบ CMMS, Manufacturing Execution System (MES) (ซึ่งทั้งสองมักเป็นเครื่องมือในระดับ 3) แต่ก็สามารถกำหนดให้กับระบบ ERP (Enterprise Resource Planning) หรือระบบควบคุมแบบกระจาย (Distributed Control System) ด้วย



รูปที่ 2.5 ภาพรวมของระบบอุตสาหกรรมในอนาคต ที่มีการประกอบเซอร์วิสบนคลาวด์ [24]

รูปที่ 2.6 แสดงสถาปัตยกรรมเชิงบริการในภาพรวม ตามแนวคิดในงานวิจัยของ Karnouskos และคณะ [24] ซึ่งใช้สัญลักษณ์อ้างอิง FMC [26] สถาปัตยกรรมเชิงบริการนี้เน้นที่การพิจารณาความต้องการ 2 ส่วนที่พึงพิจารณา ดังนี้

(1) บทบาทผู้ใช้ระบบ (User Roles)

บทบาทผู้ใช้ระบบถูกจัดแบ่งตามบทบาทของผู้ใช้ในการทำงานกับสถาปัตยกรรม ทั้งทางตรงหรือทางอ้อม ในโรงงานอุตสาหกรรมหนึ่งๆ พนักงานและผู้บริหารภายในโรงงานจะเป็นผู้ทำงานตามบทบาทหน้าที่ซึ่งกำหนดด้วยบทบาทผู้ใช้ระบบ (ดูกล่องทางด้านซ้ายในรูปที่ 2.6)

- บทบาท Business หมายถึง บทบาทหน้าที่ในการดำเนินธุรกิจ การบริหารจัดการการผลิตและการควบคุม เป็นต้น หน้าที่เหล่านี้ต้องการให้มีการวางแผนเชิงกลยุทธ์ในระยะยาวที่มีประสิทธิผล. จากมุมมองทางด้านไอที ถือว่าบทบาทเหล่านี้อยู่ในชั้น Enterprise ของโรงงาน และมีปฏิสัมพันธ์กับระบบสนับสนุน อาทิเช่น ระบบ ERP (Enterprise Resource Planning) และ ระบบ EAM เป็นต้น

- บทบาท Operations หมายถึง บทบาทหน้าที่ในการดำเนินงานปกติของโรงงานอุตสาหกรรม ซึ่งต้องมีประสิทธิภาพ มีกระบวนการที่น่าเชื่อถือได้ และมีความปลอดภัย บทบาทนี้อยู่

ที่ชั้น Operations และใช้ระบบสนับสนุน อาทิเช่น OCS (Operations Control System) เพื่อควบคุมและติดตามโครงสร้างพื้นฐานของกระบวนการ (Process Infrastructure) และ Process Optimization Systems ของโรงงาน

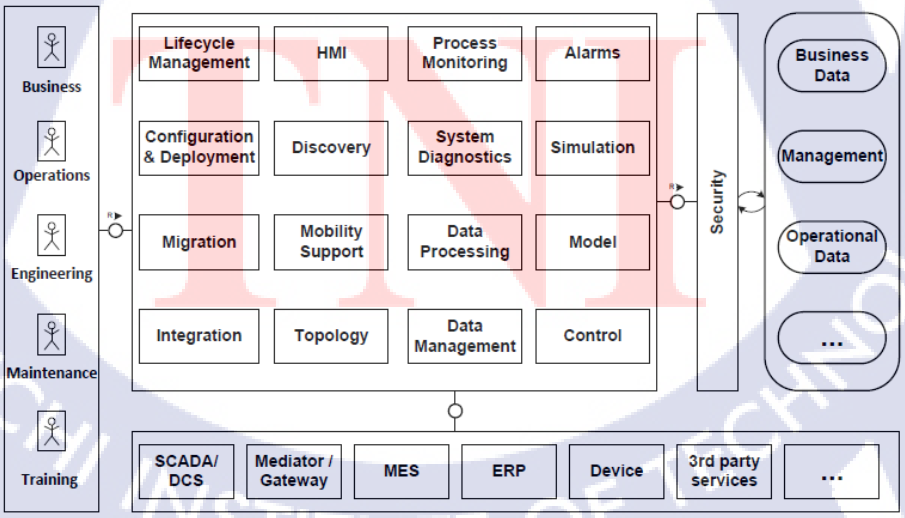
- บทบาท Engineering หมายถึง หน้าที่ทางด้านวิศวกรรม ได้แก่ หน้าที่ของวิศวกรในทางวิศวกรรมกระบวนการ (Process Engineering) และวิศวกรรมระบบ (System Engineering)

- บทบาท Maintenance หมายถึง บทบาทหน้าที่ในการบำรุงรักษา เพื่อให้การทำงานกับระบบมีประสิทธิภาพดีที่สุด (Optimum Performance) และระบบและอุปกรณ์ต่างๆ ของโรงงานมีสถานะทำงานได้เต็มที่ น่าเชื่อถือได้ และปลอดภัย การดำเนินงานบำรุงรักษายังเป็นส่วนหนึ่งของชั้นการดำเนินงานไอที (Operations IT Layer) ของโรงงานด้วย ระบบต่างๆ ที่สนับสนุนการทำงานต่างๆ ที่ทำในบทบาทในการบำรุงรักษาจะเกี่ยวข้องกับ การตรวจสอบโดยการประเมินความเสี่ยง (Risk Based Inspections: RBI), การเฝ้าติดตามระบบ (Systems Monitoring) และการวินิจฉัยและการควบคุม (Diagnostics and Control) เป็นต้น

- บทบาท Training หมายถึง หน้าที่การฝึกอบรมและฝึกปฏิบัติ เพื่อนำไปใช้ในการปรับปรุงทักษะการทำงาน รวมถึง การวางแผนการฝึกอบรมและฝึกปฏิบัติ เพื่อให้สอดคล้องกับกลยุทธ์ทางการบริหาร

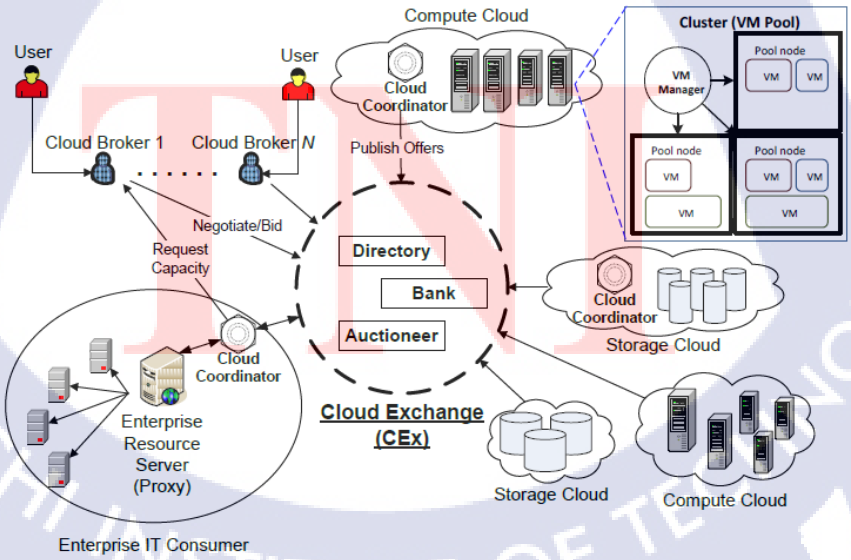
(2) ภาพรวมของกลุ่มบริการ (Service Group Overview)

ดังแสดงในรูปที่ 2.6 การจัดกลุ่มบริการหลากหลาย ซึ่งถือว่าทุกบริการมีความสำคัญกับระบบอัตโนมัติแบบร่วมกันทำงานด้วยคลาวด์ (Cloud-Based Collaborative Automation Systems) ในยุคหน้า



รูปที่ 2.6 ภาพรวมสถาปัตยกรรมเชิงบริการ ซึ่งใช้สัญลักษณ์อ้างอิง FMC [24]

นอกจากนี้มีสถาปัตยกรรมและส่วนประกอบของอินเตอร์คลาวด์ (InterCloud) ที่น่าสนใจที่กล่าวโดย Rajkimar Buyya และคณะ [3] ได้นำเสนอสถาปัตยกรรมคลาวด์คอมพิวติงแบบรวมกลุ่มพันธมิตรเพื่อการให้บริการโปรแกรมประยุกต์ที่รองรับการขยายตัวได้ ในภาพรวม ได้นำเสนอการขยายตัวของผู้ให้บริการคลาวด์ โดยที่มีสถานที่ตั้งของศูนย์ข้อมูลที่อยู่ในภูมิภาคต่างๆกัน ให้สามารถมารวมกลุ่มกันเพื่อให้บริการโดยมีตัวกลางในการเชื่อมโยงการการบริการ ระหว่างผู้ให้บริการคลาวด์และผู้ใช้งาน โดยผ่าน 3 องค์ประกอบหลักคือ ผู้ประสานงานคลาวด์ (Cloud Coordinator: CC), นายหน้าคลาวด์ (Cloud Broker: CB) และ ผู้แลกเปลี่ยนคลาวด์ (Cloud Exchange: CEx) ซึ่งทั้ง 3 องค์ประกอบทำงานเพื่อให้การบริการทำงานร่วมกันได้อย่างสอดคล้องกัน ตามการใช้งานระหว่างผู้ใช้งานและผู้ให้บริการคลาวด์ ให้คลาวด์แต่ละคลาวด์สามารถทำงานร่วมกันได้ โดยผ่านการจัดการโดย ผู้แลกเปลี่ยนคลาวด์ (Cloud Exchange: CEx) เป็นผู้ให้การสนับสนุนสำหรับการจัดการความต้องการของผู้ใช้งาน และรวมถึงการจัดการรายการการให้บริการของผู้ให้บริการคลาวด์ โดยผู้ใช้งานสามารถเรียกใช้งานโปรแกรมประยุกต์ได้โดยไม่จำเป็นต้องทราบว่าได้รับบริการจากคลาวด์ที่ใด สถาปัตยกรรมระบบที่นำเสนอโดยงานวิจัยนี้ทำให้โปรแกรมประยุกต์หนึ่งๆ สามารถให้บริการได้จากหลายๆคลาวด์ โดยสถาปัตยกรรมนี้จะสนับสนุนการให้บริการโดยมีการขยายตัวของคลาวด์ และผู้ใช้งาน ได้มากขึ้นโดยมีความยืดหยุ่นสูง และรองรับการขยายตัวสูงขึ้น รูปที่ 2.7 แสดงให้เห็นถึงกลุ่มเครือข่ายการให้บริการโดยผู้ให้บริการคลาวด์ โดยมีตัวกลางเป็น ผู้แลกเปลี่ยนคลาวด์ ตามความต้องการใช้งานของผู้ใช้งานแต่ละคนที่มีความต้องการใช้งานโปรแกรมประยุกต์ที่แตกต่างกันออกไป และแสดงให้เห็นถึงองค์ประกอบอื่นๆ ที่มีในอินเตอร์คลาวด์ (InterCloud)

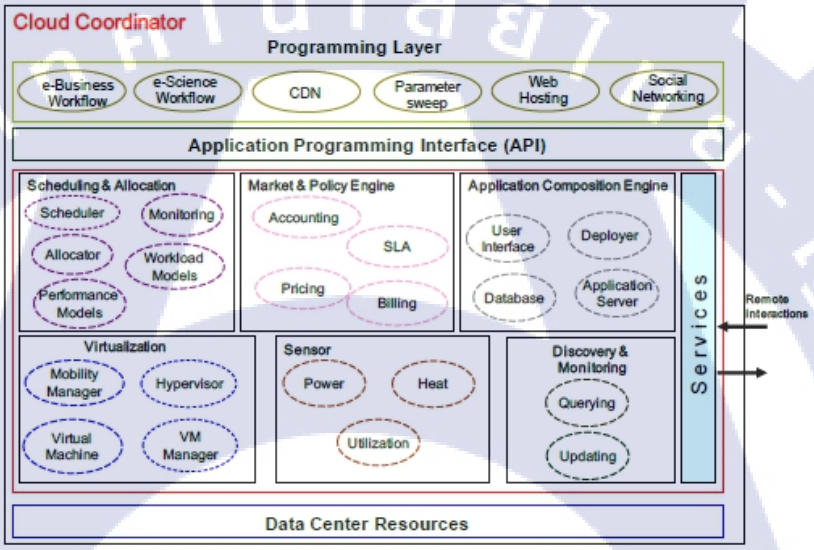


รูปที่ 2.7 เครือข่ายกลุ่มให้บริการคลาวด์ โดยมีตัวกลางเป็น ผู้แลกเปลี่ยนคลาวด์ [3]

Rajkimar Buyya และคณะ ได้กำหนดองค์ประกอบในสถาปัตยกรรมของอินเตอร์คลาวด์ สามารถจะอธิบายตามองค์ประกอบที่มีได้ดังนี้

(1) ผู้ประสานงานคลาวด์ (Cloud Coordinator: CC)

มีบทบาทเป็นผู้โต้ตอบในการให้บริการ โดยจะเป็นผู้รับผิดชอบในส่วนของทรัพยากรบนคลาวด์ที่รับผิดชอบ ทำการจัดเตรียมโปรแกรมประยุกต์ และสภาพแวดล้อมของโปรแกรมประยุกต์นั้นๆ จากความต้องการที่มีการร้องขอที่ส่งผ่านมาจาก ผู้แลกเปลี่ยนคลาวด์ (Cloud Exchange) รูปที่ 2.8 แสดงให้เห็นถึงรายละเอียดส่วนประกอบในการจัดการทรัพยากรบนคลาวด์ในส่วนของผู้ประสานงานคลาวด์ให้บริการ



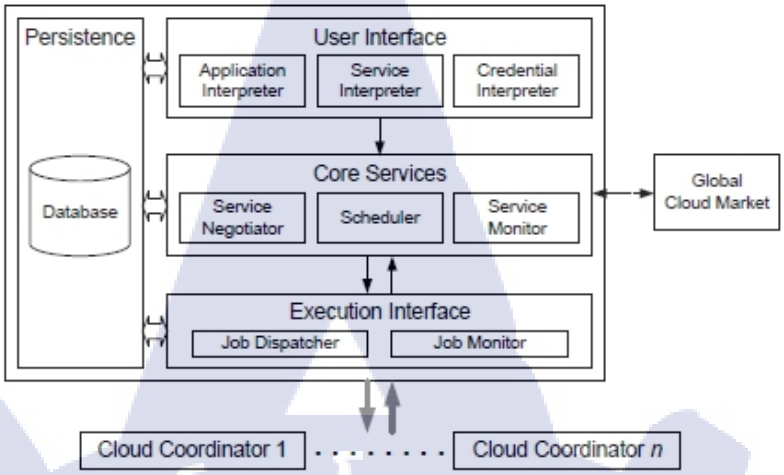
รูปที่ 2.8 สถาปัตยกรรมซอฟต์แวร์ของผู้ประสานงานคลาวด์ [3]

(2) นายหน้าคลาวด์ (Cloud Broker: CB)

มีบทบาทเป็นผู้ระบุผู้ให้บริการคลาวด์ที่เหมาะสมให้กับผู้ใช้งานที่มีการเรียกใช้บริการ โดยทำการสื่อสารผ่าน ผู้แลกเปลี่ยนคลาวด์ (Cloud Exchange) รูปที่ 2.9 แสดงให้เห็นถึงรายละเอียดส่วนประกอบในการจัดการทรัพยากรบนคลาวด์ในส่วนของผู้ประสานงานคลาวด์ให้บริการ

(3) ผู้แลกเปลี่ยนคลาวด์ (Cloud Exchange: CEx)

มีบทบาทเป็นผู้รวบรวมข้อมูลของการให้บริการคลาวด์จากผู้ให้บริการคลาวด์ที่อยู่ในกลุ่มพันธมิตร และทำการจับคู่การให้บริการ โดยให้ข้อมูลกับ นายหน้าคลาวด์ (Cloud Broker) เพื่อทำการค้นหาทรัพยากรที่เหมาะสมจากข้อมูลผู้ให้บริการคลาวด์ที่มีข้อมูลอยู่



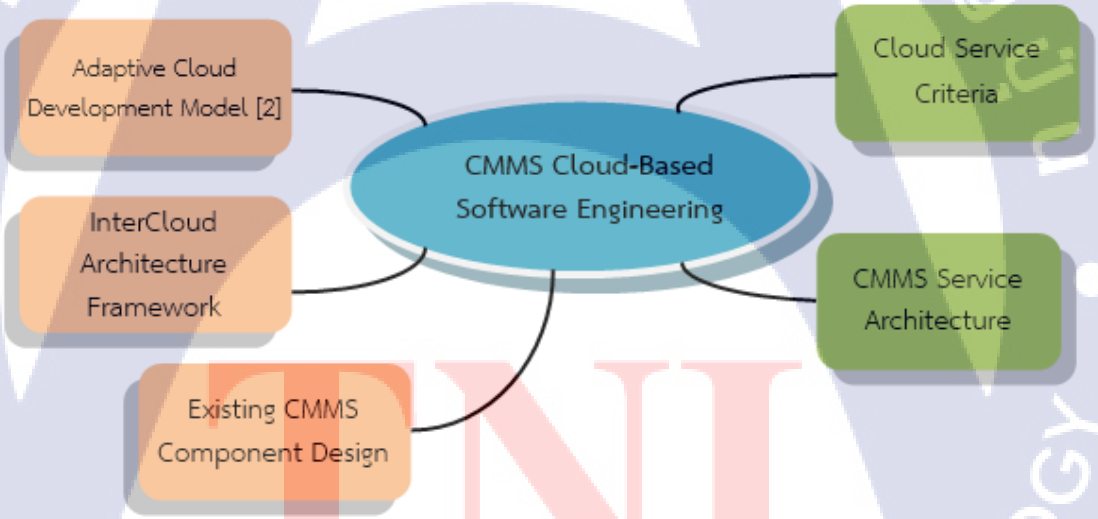
รูปที่ 2.9 สถาปัตยกรรมขั้นสูงของการให้บริการโดยนายหน้าคลาวด์ [3]

บทที่ 3
วิธีดำเนินการวิจัย

งานวิจัยนี้จะการดำเนินงานภายใต้กรอบแนวคิดที่พัฒนาจากที่ผู้วิจัยได้ศึกษาค้นคว้าขั้นต้นในด้านเทคโนโลยีคลาวด์คอมพิวติง และสถาปัตยกรรมซอฟต์แวร์ที่เกี่ยวข้อง เพื่อเป็นแนวทางในการออกแบบสถาปัตยกรรมคลาวด์คอมพิวติง สำหรับระบบการบริหารงานบำรุงรักษา (CMMS)

3.1 กรอบแนวคิดการวิจัย

ผู้วิจัยได้พัฒนากรอบแนวคิดของการวิจัย ดังรูปที่ 3.1 เพื่อที่ค้นหาแนวทางในการออกแบบสถาปัตยกรรมคลาวด์คอมพิวติงสำหรับระบบการบริหารงานบำรุงรักษา (CMMS) ตามแนวคิดของ Karnouskos [24] และเกณฑ์คุณภาพของเซอร์วิส (QoS) ในกระบวนการหนึ่งในกระบวนการบริหารการบำรุงรักษา แล้วพัฒนาคลาวด์เซอร์วิสเพื่อทดสอบสถาปัตยกรรม ตามวิธีการ Adaptive Cloud Development [2]

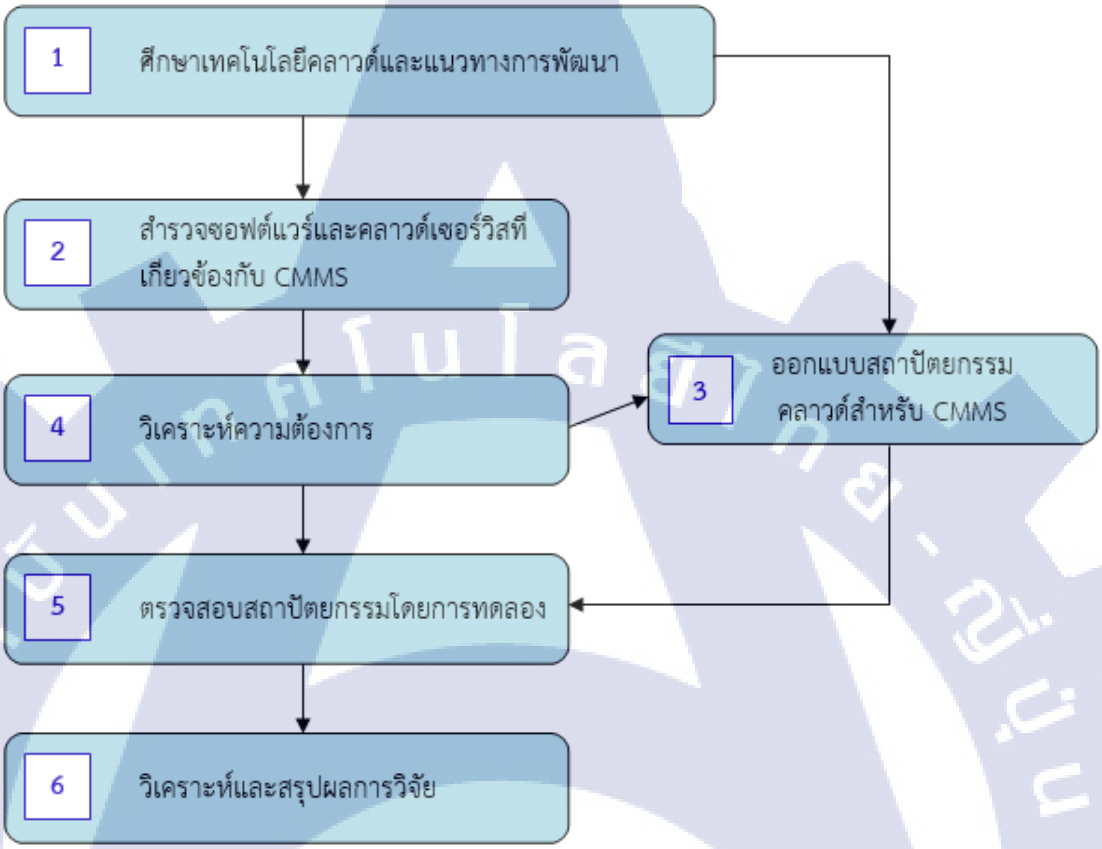


รูปที่ 3.1 กรอบแนวคิดของการวิจัย

3.2 ขั้นตอนการวิจัย

ผู้วิจัยได้พัฒนากรอบแนวคิดของการวิจัย ดังรูปที่ 3.1 เพื่อที่ค้นหาแนวทางในการออกแบบสถาปัตยกรรมคลาวด์คอมพิวติงสำหรับระบบการบริหารงานบำรุงรักษา (CMMS) ตามแนวคิดของ Karnouskos [24] และเกณฑ์คุณภาพของเซอร์วิส (QoS) ในกระบวนการหนึ่งในกระบวนการบริหาร

การบำรุงรักษา แล้วพัฒนาคลาวด์เซอร์วิสเพื่อทดสอบสถาปัตยกรรม ตามวิธีการ Adaptive Cloud Development [2]



รูปที่ 3.2 ขั้นตอนการวิจัย

การวิจัยจะดำเนินงานโดยแบ่งออกเป็น 6 ขั้นตอน ดังแสดงในรูปที่ 3.2

(1) ศึกษาเทคโนโลยีคลาวด์และแนวทางการพัฒนา

ขั้นตอนนี้จะทำการศึกษาเทคโนโลยีคลาวด์ และรูปแบบของคลาวด์เซอร์วิสที่มีอยู่จากผู้ให้บริการในขอบเขตของระบบ CMMS

(2) สำรวจซอฟต์แวร์และคลาวด์เซอร์วิสที่เกี่ยวข้องกับ CMMS

ขั้นตอนนี้จะสำรวจซอฟต์แวร์ประเภทระบบบริหารการบำรุงรักษาเครื่องจักร ว่ามีการสนับสนุนการทำงานในส่วนต่างๆ เพื่อทราบเป็นข้อมูลที่ในการออกแบบสถาปัตยกรรมคลาวด์ในขั้นตอนที่ 3 และการพัฒนาเชิงทดลองในขั้นตอนที่ 4

(3) ออกแบบสถาปัตยกรรมคลาวด์สำหรับ CMMS

ขั้นตอนนี้จะเป็นการออกแบบสถาปัตยกรรมคลาวด์คอมพิวติง สำหรับระบบการบริหารงานบำรุงรักษา (CMMS) เพื่อเอื้ออำนวยให้ซอฟต์แวร์สามารถที่จะทำงานบนคลาวด์ได้ โดยมุ่งเน้นไปในรูปแบบของการให้บริการ และยังคงมีคุณสมบัติการทำงานของซอฟต์แวร์ที่มีอยู่เดิม สถาปัตยกรรมจะมีเกณฑ์ความต้องการขั้นต้นตามแนวคิดของ Karnouskos [24] และเกณฑ์คุณภาพของเซอร์วิส (QoS) ในด้านการนำไปใช้และการคอนฟิก (Configuration & Deployment) การโยกย้ายระบบหรือบริการ (Migration) การบูรณาการระบบ (Integration) การบริหารข้อมูล (Data Management)

(4) วิเคราะห์ความต้องการ

ขั้นตอนนี้จะเป็นวิเคราะห์ความต้องการคลาวด์เซอร์วิส ตามแนวคิดของ Adaptive Cloud Development Model [2] เพื่อการบริหารกระบวนการที่คัดเลือก กระบวนการหนึ่งในการบริหารการบำรุงรักษา คือ WO (Work Order) เพื่อให้ทำงานประสานกับระบบ CMMS ที่มีอยู่เดิม

(5) ตรวจสอบสถาปัตยกรรมโดยการทดลอง

การตรวจสอบสถาปัตยกรรมว่าเหมาะสมหรือไม่ จะใช้วิธีทดลองพัฒนาคลาวด์เซอร์วิสตามข้อกำหนดความต้องการที่ได้เป็นลัพธ์จากขั้นตอนที่ 4 การพัฒนาจะใช้แนวคิดของ Adaptive Cloud Development Model [2] แล้วจึงประเมินผลลัพธ์ที่ได้จากการพัฒนาเชิงทดลอง (คลาวด์เซอร์วิส)

การออกแบบ API ตามแนวคิดของ SOA เพื่อสนับสนุนการพัฒนาโครงสร้างระบบบนคลาวด์อ้างอิงมาตรฐานระบบการจัดการคลาวด์ ในที่นี้ คือ InterCloud

(6) วิเคราะห์และสรุปผลการวิจัย

การวิเคราะห์และสรุปการวิจัยจะใช้ผลของการทดลองพัฒนาคลาวด์เซอร์วิส CMMS และประเมินผลลัพธ์เทียบกับเกณฑ์การประเมินเซอร์วิส ที่ตั้งไว้ในขั้นตอนที่ 3

บทที่ 4
ผลการดำเนินงาน

งานวิจัยนี้ได้ดำเนินการวิเคราะห์ความต้องการของระบบงาน CMMS ที่มีอยู่เดิม (Requirement Analysis) ที่มีการพัฒนาด้วยภาษาวิชวลเบสิก (Visual Basic) และมีโครงสร้างสถาปัตยกรรมแบบ Client/Server ซึ่งเป็นการใช้งานในรูปแบบการจัดการแบบรวมศูนย์ (Centralized System) ซึ่งในกระบวนการหนึ่งในการบริหารการบำรุงรักษา (CMMS) คือ WO (Work Order) โดย ส่วนการทำงานของ WO นั้นมีการแบ่งแยกผู้ใช้งาน (Actor) ออกเป็น 3 ส่วน คือ ผู้รับบริการหรือลูกค้า (Client) ผู้ให้บริการงานซ่อมบำรุง (Maintenance Service Provider) และช่างเทคนิค (Technician) โดยผู้ใช้งานแต่ละประเภทมีหน้าที่การทำงานที่มีความสัมพันธ์สอดคล้องกัน (ดูรูปที่ 4.1)

4.1 การวิเคราะห์บริการคลาวด์

4.1.1 Work Order Process และ Use Case Diagram

กระบวนการซ่อมบำรุงเพื่อที่จะสั่งงานซ่อม (Create Work Order: WO) มียูสเคส (Use Case) ที่สำคัญๆ ดังนี้ (ดูรูปที่ 4.1)

(1) ผู้รับบริการหรือลูกค้าส่งคำร้องขอเมื่อพบปัญหา (Send Request for Problem Found) ผู้รับบริการหรือลูกค้า แจ้งปัญหาที่พบ เพื่อให้ดำเนินการตรวจสอบและแก้ไข ต่อไป

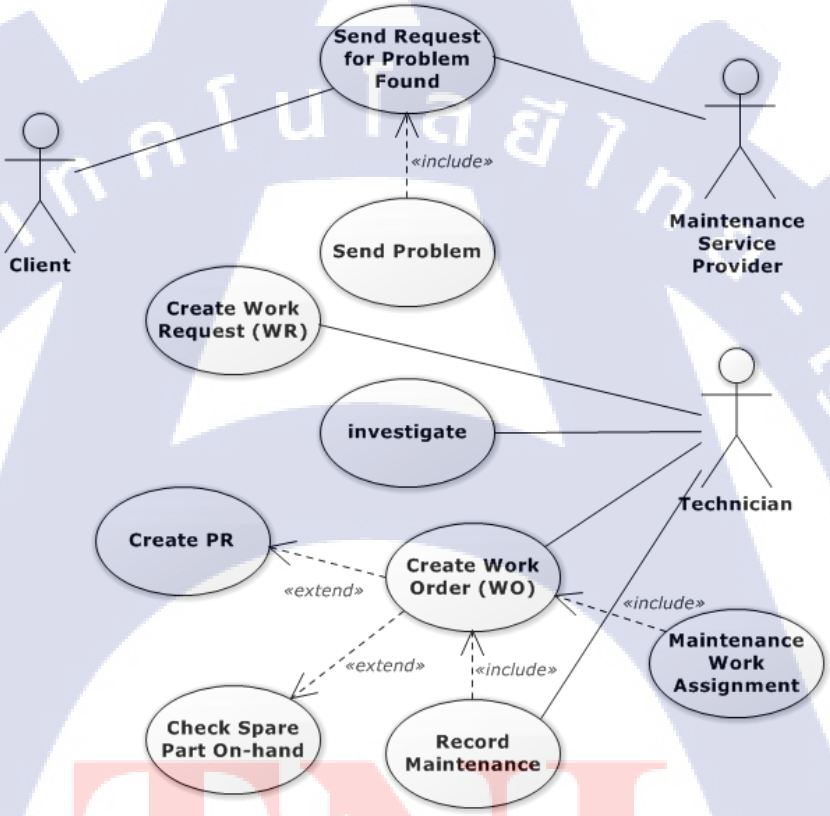
(2) ช่างเทคนิคบันทึกคำร้องขอ (Create Work Request: WR) ช่างเทคนิคทำการบันทึกปัญหาที่ได้รับแจ้งเข้ามา

(3) ช่างเทคนิคตรวจสอบและวินิจฉัยเบื้องต้น (Investigate) ช่างเทคนิคดำเนินการเข้าตรวจสอบและวินิจฉัยเบื้องต้น สำหรับปัญหาที่แจ้งเข้ามา

(4) ช่างเทคนิคบันทึกคำสั่งงานซ่อม (Create Work Order: WO) ช่างเทคนิคทำการลงบันทึกคำสั่งงานซ่อม โดย ระบุถึงวิธีการดำเนินการซ่อมบำรุงปัญหาที่รับแจ้งเข้ามา ถ้าการดำเนินการซ่อมนั้น ไม่สามารถดำเนินการได้โดยเจ้าหน้าที่ของหน่วยงานก็ส่งเรื่องไปยังผู้ให้บริการงานซ่อมบำรุง (Maintenance Service Provider) แต่ถ้าหากสามารถทำได้โดยเจ้าหน้าที่ของหน่วยงานก็ต้องพิจารณาว่างานซ่อมนั้นมีความต้องการเบิกใช้อะไหล่ทดแทนหรือไม่ กรณีต้องมีการเปลี่ยนอะไหล่ ก็ต้องดำเนินการตรวจสอบว่ามีอะไหล่คงคลังเพียงพอกับการใช้งานหรือไม่หากไม่เพียงพอ ก็ดำเนินการจัดทำคำสั่งขอซื้ออะไหล่ต่อไป (Create Purchase Request: PR)

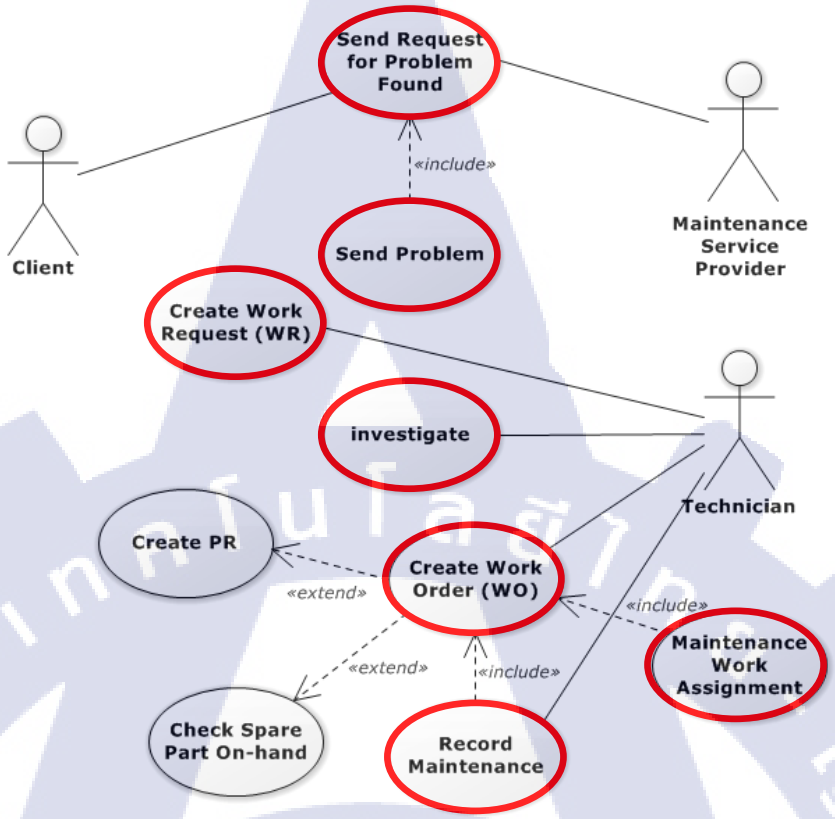
(5) ช่างเทคนิคมอบหมายงานเพื่อดำเนินการซ่อมบำรุง (Maintenance Work Assignment) ช่างเทคนิคทำการระบุเจ้าหน้าที่เทคนิคเข้าดำเนินการในการบำรุงรักษาตามคำสั่งงานซ่อมนั้นๆ

(6) ช่างเทคนิคบันทึกการซ่อมบำรุง (Record Maintenance) ช่างเทคนิคทำการบันทึกผลการดำเนินงานซ่อมบำรุงที่ดำเนินการ ว่ามีขั้นตอนการดำเนินการอย่างไรบ้าง เพื่อเก็บเป็นวิธีการดำเนินการครั้งต่อไปหากพบปัญหาลักษณะที่เกิดขึ้นเช่นเดียวกันในภายหน้า



รูปที่ 4.1 แผนภาพ Use Case สำหรับกระบวนการคำสั่งซ่อม (Work Order Process)

กระบวนการทำงานของการบันทึกคำสั่งงานซ่อม (Create Work Order: WO) ตั้งแต่เริ่มต้นจนจบกระบวนการทั้งหมด จะมียูสเคส ต่างๆ ซึ่งสามารถสามารถอธิบายกรณีการใช้ระบบปกติ (Scenario) ได้ 3 กรณี ดังนี้



รูปที่ 4.2 กรณีการใช้ระบบที่ 1 กระบวนการเพื่อการซ่อมบำรุงโดยไม่เบิกอะไหล่

กรณีการใช้ระบบที่ 1

วัตถุประสงค์

การซ่อมบำรุงโดยไม่ต้องเบิกอะไหล่ในการซ่อมบำรุงครั้งนั้น มีการทำงานตั้งแต่การส่งคำร้องของานซ่อม จนถึงการบันทึกการซ่อม

ลำดับการใช้ระบบ (ดูรูปที่ 4.2)

1) ผู้รับบริการหรือลูกค้าส่งคำร้องขอเมื่อพบปัญหา (Send Request for Problem Found)

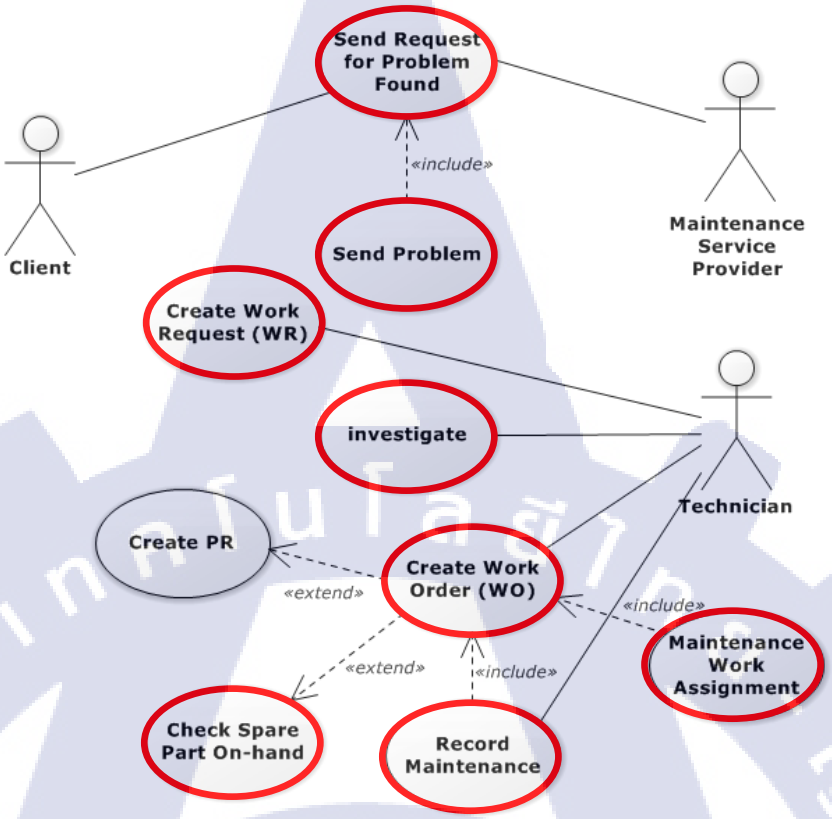
2) ช่างเทคนิคบันทึกคำร้องขอ (Create Work Request: WR)

3) ช่างเทคนิคตรวจสอบและวินิจฉัยเบื้องต้น (Investigate)

4) ช่างเทคนิคบันทึกคำสั่งงานซ่อม (Create Work Order: WO)

5) ช่างเทคนิคมอบหมายงานเพื่อดำเนินการซ่อมบำรุง (Maintenance Work Assignment)

6) ช่างเทคนิคบันทึกการซ่อมบำรุง (Record Maintenance)



รูปที่ 4.3 กรณีการใช้ระบบที่ 2 กระบวนการเพื่อการซ่อมบำรุงโดยมีการเบิกอะไหล่คงคลัง

กรณีการใช้ระบบที่ 2

วัตถุประสงค์

การทำงานตั้งแต่ต้นกระบวนการจนจบกระบวนการ โดยการซ่อมบำรุงต้องใช้อะไหล่ในการซ่อมบำรุง และ มีอะไหล่คงเหลือเพียงพอ มีกระบวนการดังนี้

ลำดับการใช้ระบบ (ดูรูปที่ 4.3)

1) ผู้รับบริการหรือลูกค้าส่งคำร้องขอเมื่อพบปัญหา (Send Request for Problem Found)

2) ช่างเทคนิคบันทึกคำร้องขอ (Create Work Request: WR)

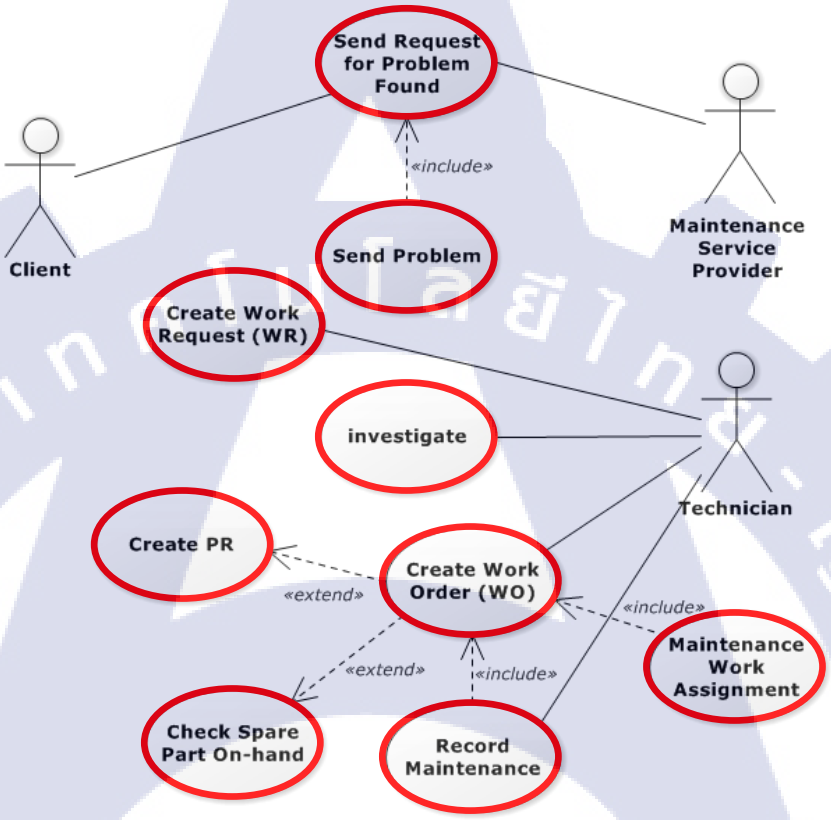
3) ช่างเทคนิคตรวจสอบและวินิจฉัยเบื้องต้น (Investigate)

4) ช่างเทคนิคบันทึกคำสั่งงานซ่อม (Create Work Order: WO)

5) ช่างเทคนิคตรวจสอบอะไหล่คงเหลือ (Check Spare Part On-hand) เมื่อการซ่อมบำรุงต้องทำการเปลี่ยนอะไหล่ และอะไหล่คงเหลือเพียงพอ พร้อมเบิก

6) ช่างเทคนิคมอบหมายงานเพื่อดำเนินการซ่อมบำรุง (Maintenance Work Assignment)

7) ช่างเทคนิคบันทึกการซ่อมบำรุง (Record Maintenance)



รูปที่ 4.4 กรณีการใช้ระบบที่ 3 กระบวนการทำงานเพื่อการซ่อมบำรุงโดยมีการใช้อะไหล่คงคลังแต่ไม่เพียงพอ

กรณีการใช้ระบบที่ 3

วัตถุประสงค์

การทำงานตั้งแต่ต้นกระบวนการจนจบกระบวนการ โดยการซ่อมบำรุงต้องใช้อะไหล่ในการซ่อมบำรุง และ อะไหล่คงเหลือไม่เพียงพอ ต้องทำการขอซื้อก่อนดำเนินการต่อไป มีกระบวนการดังนี้

ลำดับการใช้ระบบ (ดูรูปที่ 4.4)

1) ผู้รับบริการหรือลูกค้าส่งคำร้องขอเมื่อพบปัญหา (Send Request for Problem Found)

2) ช่างเทคนิคบันทึกคำร้องขอ (Create Work Request: WR)

3) ช่างเทคนิคตรวจสอบและวินิจฉัยเบื้องต้น (Investigate)

4) ช่างเทคนิคบันทึกคำสั่งงานซ่อม (Create Work Order: WO)

5) ช่างเทคนิคตรวจสอบอะไหล่คงเหลือ (Check Spare Part On-hand)

6) ช่างเทคนิคการจัดทำคำสั่งขอซื้ออะไหล่ต่อไป (Create Purchase Request: PR)

7) ช่างเทคนิคมอบหมายงานเพื่อดำเนินการซ่อมบำรุง (Maintenance Work Assignment)

8) ช่างเทคนิคบันทึกการซ่อมบำรุง (Record Maintenance)

เหตุการณ์ที่เกิดขึ้นของกระบวนการทำงานของการบันทึกคำสั่งงานซ่อมนั้นจะมีกระบวนการหลัก ที่อย่างน้อยต้องดำเนินการสรุปได้เป็น 6 ยูสเคสเป็นอย่างน้อยในการดำเนินงานของการบันทึกคำสั่งงานซ่อม ให้เสร็จตั้งแต่ต้นจนจบกระบวนการ คือ

1) ผู้รับบริการหรือลูกค้าส่งคำร้องขอเมื่อพบปัญหา (Send Request for Problem Found)

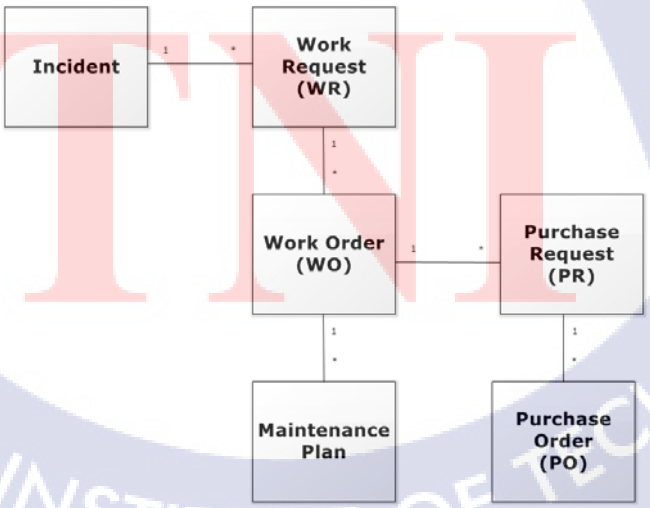
2) ช่างเทคนิคบันทึกคำร้องขอ (Create Work Request: WR)

3) ช่างเทคนิคตรวจสอบและวินิจฉัยเบื้องต้น (Investigate)

4) ช่างเทคนิคบันทึกคำสั่งงานซ่อม (Create Work Order: WO)

5) ช่างเทคนิคมอบหมายงานเพื่อดำเนินการซ่อมบำรุง (Maintenance Work Assignment)

6) ช่างเทคนิคบันทึกการซ่อมบำรุง (Record Maintenance)



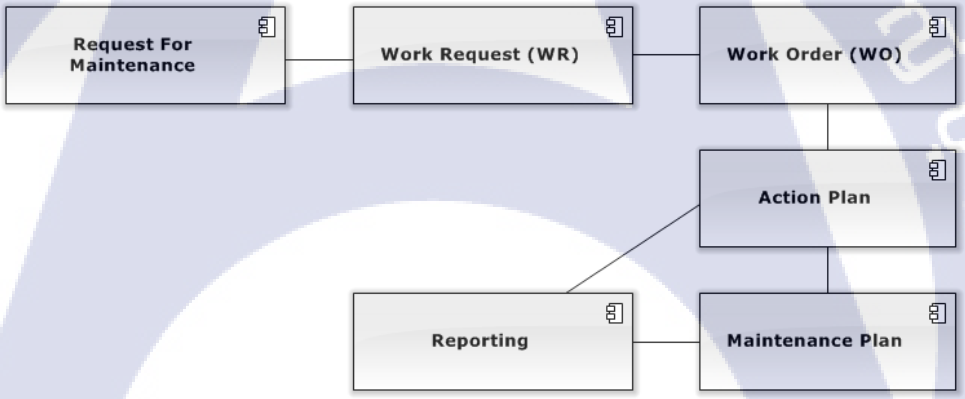
รูปที่ 4.5 Domain Class Diagram สำหรับ CMMS

4.1.2 Domain Class Diagram

หลังจากได้พัฒนา Use Case แล้ว (ดูรูปที่ 4.1) จึงพัฒนาขั้นตอนต่อไป คือการวิเคราะห์ Domain Class ซึ่งในระบบเดิม มีคลาสจำนวนมาก แต่ที่สนใจ คลาสที่จะใช้บนคลาวด์ได้นั้น (ดูรูปที่ 4.5) และ UML Component Diagram (ดูรูปที่ 4.6) จะมีการปรับปรุงแก้ไขอย่างไร

เมื่อพิจารณาการใช้งานบนคลาวด์สำหรับระบบบริหารการบำรุงรักษาเครื่องจักร จะสามารถแยกส่วนการทำงานออกเป็นหลายๆ คลาวด์ เพื่อให้การทำงานสอดคล้องประสานกันได้โดยอาศัยแนวคิดของสถาปัตยกรรม InterCloud เพื่ออธิบายการทำงานว่ามีส่วนการทำงานที่แต่ละคลาวด์ สามารถทำหน้าที่เหมือนกันได้ โดยไม่จำเป็นต้องอยู่ภายในคลาวด์เดียวกัน สามารถเชื่อมโยงโดยมีการจัดการผ่าน Cloud Coordinator หรือ Cloud Broker ทำหน้าที่จัดการและสื่อสารกับตัวกลางที่เป็น Cloud Exchange (CEx) (ดูรูปที่ 4.7)

4.1.3 UML Component Diagram For CMMS



รูปที่ 4.6 UML Component Diagram

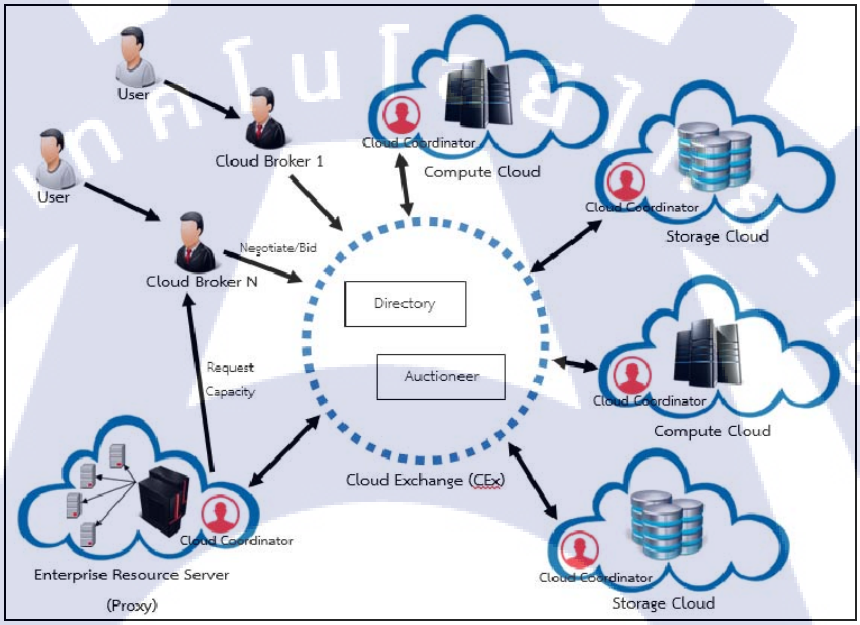
4.1.4 สถาปัตยกรรมระบบและส่วนประกอบของ InterCloud

สถาปัตยกรรมระบบของ InterCloud ตามรูปที่ 4.7 นั้นจะสามารถทำให้การทำงานร่วมกันระหว่างโดเมนที่แยกจากกันได้อย่างลงตัว ทำให้สามารถแบ่งการทำงานกระจายออกไปยังโดเมนที่เกียวข้องกันให้ทำงานร่วมกันได้โดยผ่าน Cloud Coordinator หรือ Cloud Broker ของแต่ละโดเมนเอง ทำหน้าที่จัดการ โดยมี Cloud Exchange เป็นตัวกลางช่วยเหลือ Cloud Coordinator หรือ Cloud Broker ในการสื่อสารกัน

Cloud Coordinator (CC) เป็นส่วนที่ทำหน้าที่ในการจัดการภายใต้คลาวด์ใดคลาวด์หนึ่ง เพื่อจัดการทรัพยากร เพื่อการบริการภายในคลาวด์นั้นๆ

Cloud Broker (CB) เป็นส่วนที่ทำหน้าที่ในการจัดการส่วนของผู้ใช้งาน ให้ได้รับบริการจากผู้ให้บริการคลาวด์ ผ่าน Cloud Exchange (CEx) ในการที่เป็นตัวกลาง

Cloud Exchange (CEx) เป็นส่วนที่ทำหน้าที่ ในการจัดเก็บข้อมูลของแต่ละผู้ให้บริการคลาวด์ และจัดการจัดสรรทรัพยากรของผู้ให้บริการคลาวด์ กับผู้ใช้งานตามความต้องการ ที่ต้องการใช้งานให้ได้อย่างเหมาะสม



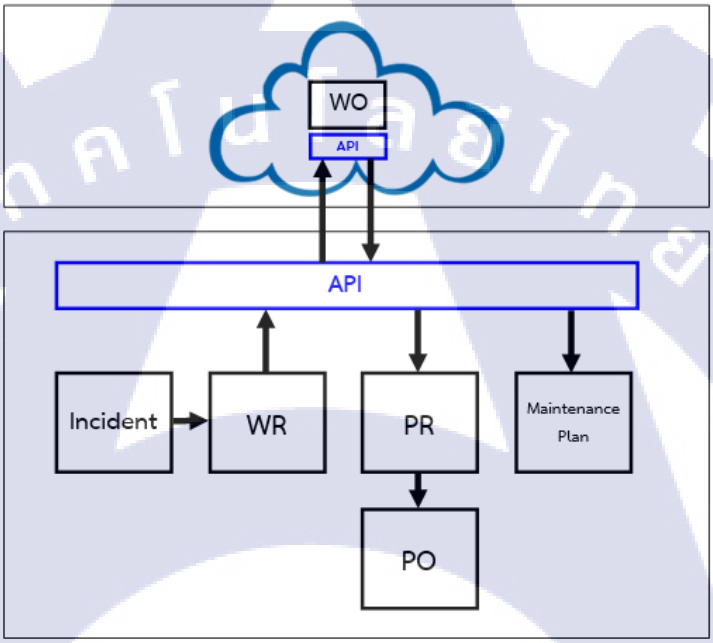
รูปที่ 4.7 Cloud Exchange ที่เป็นไปได้ และ Component

4.1.5 Cloud Based Component Development

ในกรณีของ CMMS ซึ่งต้องการปรับตัวขึ้นไปใช้งานบนคลาวด์ มีการตัดสินใจในการเลือกโมดูลบันทึกคำสั่งงานซ่อม (WO) เนื่องจากการบันทึกคำสั่งงานซ่อม เป็นการระบุว่าต้องการให้กับช่างเทคนิคกลุ่มที่มีความเชี่ยวชาญด้านใดในปัญหานั้น ให้เข้าแก้ไขและซ่อมบำรุงงานนั้นๆให้ลุล่วงไปด้วยดี อีกทั้งกรณีที่องค์กรไม่ได้มีเจ้าหน้าที่เทคนิคในงานแก้ไขนั้นๆ ก็จะทำให้สามารถส่งคำสั่งซ่อม (WO) ไปยังผู้ให้บริการงานซ่อมบำรุง (Maintenance Service Provider) ได้อย่างสะดวก แต่ทั้งนี้โมดูลอื่นๆ ที่เกี่ยวข้องในระบบ CMMS ก็สามารถที่จะปรับตัวไปใช้งานบนคลาวด์ได้เช่นเดียวกัน

การดำเนินงานในส่วนของโมดูลบันทึกคำสั่งงานซ่อม (WO) ต้องมีการเตรียมการในส่วนของการพัฒนาโดยต้องปรับปรุงเพิ่มเติมความสามารถของระบบ CMMS เดิม ให้สามารถใช้งาน

ร่วมกับโมดูลบันทึกคำสั่งงานซ่อม (WO) ที่นำไปใช้งานบนคลาวด์ โดยต้องทำการเตรียมในส่วนของตัวกลางที่ทำการเชื่อมต่อ ระหว่าง ระบบ CMMS เดิมที่อยู่บนโครงสร้างสถาปัตยกรรมแบบ Client/Server และ โมดูลบันทึกคำสั่งงานซ่อมที่จะปรับขึ้นไปใช้งานบนคลาวด์ ที่มีสถาปัตยกรรมเป็น สถาปัตยกรรมระบบเชิงบริการ (Service-Oriented Architecture) โดยพัฒนา API เพื่อเป็นตัวเชื่อมต่อและส่งผ่านการทำงานให้เข้ากันได้และเหมาะสมระหว่างทั้งสองระบบ [27] รูปแบบของ Cloud Based Component Development แสดงในรูปที่ 4.8



รูปที่ 4.8 Cloud Based Component Development

ในส่วนของการดำเนินงานหลังจากนั้นจะต้องทำการทดสอบเพื่อให้แน่ใจว่าหลังจากที่นำโมดูลบันทึกคำสั่งงานซ่อมที่เป็นส่วนหนึ่งของระบบ CMMS ให้สามารถใช้งานได้บนคลาวด์แล้ว โมดูลบันทึกคำสั่งงานซ่อมที่อยู่บนระบบ CMMS ก็จะต้องทำงานได้โดยไม่ติดปัญหาเช่นกัน

4.2 การออกแบบ API

การออกแบบ API ที่จะใช้งาน จากกระบวนการซ่อมบำรุง ตามที่มียูสเคสจำนวน 8 ยูสเคส (Use Case) ของการสั่งงานซ่อม (Create Work Order: WO) ดังนี้

1) ผู้รับบริการหรือลูกค้าส่งคำร้องขอเมื่อพบปัญหา (Send Request for Problem Found)

2) ช่างเทคนิคบันทึกคำร้องขอ (Create Work Request: WR)

3) ช่างเทคนิคตรวจสอบและวินิจฉัยเบื้องต้น (Investigate)

4) ช่างเทคนิคบันทึกคำสั่งงานซ่อม (Create Work Order: WO)

5) ช่างเทคนิคตรวจสอบอะไหล่คงเหลือ (Check Spare Part On-hand)

6) ช่างเทคนิคการจัดทำคำสั่งขอซื้ออะไหล่ต่อไป (Create Purchase Request: PR)

7) ช่างเทคนิคมอบหมายงานเพื่อดำเนินการซ่อมบำรุง (Maintenance Work Assignment)

8) ช่างเทคนิคบันทึกการซ่อมบำรุง (Record Maintenance)

จากนั้นจึงออกแบบ API ซึ่งประกอบด้วย Class และ Method ที่สนับสนุนการทำงานของทั้ง 8 ยูสเคส (Use Case) ที่ได้ แสดงผลลัพธ์ในตารางที่ 4.1

ตารางที่ 4.1 API ของระบบ CMMS

Use Case	API
1. ผู้รับบริการหรือลูกค้าส่งคำร้องขอเมื่อพบปัญหา (Send Request for Problem Found)	incident.receiveProblem (CompanyID, MachineID, IncidentDate, IncidentTitle, FromStaff, ProblemMsg) """ Receive incident about the problem and create ID: Parameters: CompanyID = ID of company to founding the problem MachineID = ID of machine to founding the problem IncidentDate = Date-time to sending the problem IncidentTitle = Title of incident FromStaff = ID of staff incompany to sending the problem ProblemMsg = Problem detail """
2. ช่างเทคนิคบันทึกคำร้องขอ (Create Work Request: WR)	workrequest.createWR (CompanyID, RefIncID) """ Create the Work Request can refer to incident: Parameters: CompanyID = ID of company to founding the problem RefIncID = ID of incident to reference """

ตารางที่ 4.1 API ของระบบ CMMS (ต่อ)

Use Case	API
3. ช่างเทคนิคตรวจสอบและวินิจฉัยเบื้องต้น (Investigate)	workrequest.investigateProb (RefWRID, ProblemLevel, ResultMsg) """ Record the first result to prepare next process: Parameters: RefWRID = Refer to ID of Work Request ProblemLevel = Critical Level for the problem """ ResultMsg = Result after investigate """ workrequest.closeWR (RefWRID) """ Update the status of WR to closed: Parameters: RefWRID = Refer to ID of Work Request """
4. ช่างเทคนิคบันทึกคำสั่งงานซ่อม (Create Work Order: WO)	workorder.createWO (RefWRID, WOTitle) """ Convert WR to WO to assign staff for process: Parameters: RefWRID = Refer to ID of Work Request WOTitle = Title of WO """
5. ช่างเทคนิคตรวจสอบอะไหล่คงเหลือ (Check Spare Part On-hand)	workorder.createPartList (RefWOID, PartList) """ Checking the parts to repair on WO Record: Parameters: RefWOID = Refer to ID of Work Order PartList = Refer to part number of this Work Order """
6. ช่างเทคนิคการจัดทำคำสังขอซื้ออะไหล่ต่อไป (Create Purchase Request: PR)	workorder.createPR (RefWOID, PartListO) """ Create Purchase request with parts that out of stock to use in WO: Parameters: RefWOID = Refer to ID of Work Order PartListO = Refer to part number of this Work Order """
7. ช่างเทคนิคมอบหมายงานเพื่อดำเนินการซ่อมบำรุง (Maintenance Work Assignment)	workorder.assignStaff (RefWOID, StaffID) """ Update the StaffID to specify Technician to repair on WO: Parameters: RefWOID = Refer to ID of Work Order StaffID = Specify the technician to repair on WO """

ตารางที่ 4.1 API ของระบบ CMMS (ต่อ)

Use Case	API
8. ช่างเทคนิคบันทึกการซ่อมบำรุง (Record Maintenance)	workorder.closeWO (RefWOID) """ Record the action results and update status to closed on WO: Parameters: RefWOID = Refer to ID of Work Order """

จากตารางที่ 4.1 สามารถอธิบายการทำงานของแต่ละ API ได้เป็นดังนี้

1) incident.receiveProblem เป็นการรับเรื่องปัญหาที่ได้รับแจ้งเข้ามาโดยการบันทึกปัญหา โดยต้องระบุที่มา เป็น หน่วยงาน เครื่องจักรหรืออุปกรณ์ที่พบปัญหา วันที่แจ้ง หัวเรื่องที่แจ้ง ผู้ที่ทำการแจ้ง และ ระบุรายละเอียดปัญหาที่พบ โดยการรับเรื่องปัญหาจะสร้างหมายเลข Incident เพื่อไว้ใช้อ้างอิงต่อไป

2) workrequest.createWR เป็นการสร้าง WR โดยอ้างอิงไปยังปัญหาที่ได้รับแจ้ง จากหมายเลข Incident เพื่อเป็นหมายเลขของการสร้างงานตามการแจ้งเข้ามา และใช้อ้างอิงถึงการบันทึกการปฏิบัติงานต่อไปของเจ้าหน้าที่ซ่อมบำรุง

3) workrequest.investigateProb เป็นการบันทึกผลการตรวจสอบเบื้องต้นของ WR เพื่อบันทึกว่าเป็นอย่างไร ปัญหาของงานนี้มีผลกระทบมากหรือน้อยอย่างไร

4) workrequest.closeWR เป็นการบันทึกสถานะของ WR เป็นปิดเรื่อง เมื่อปัญหาที่พบสามารถปิดเรื่องได้แล้ว

5) workorder.createWO เป็นการสร้าง WO เพื่อใช้ในการบันทึกงานทีละงาน ที่ส่วนที่ต้องซ่อมบำรุง และดำเนินการ

6) workorder.createPartList เป็นการบันทึกรายการอะไหล่ ที่ต้องนำมาใช้ในการซ่อมบำรุงสำหรับ WO แต่ละงาน

7) workorder.createPR เป็นการบันทึกคำสั่งขอซื้อรายการอะไหล่ สำหรับกรณีที่ตรวจสอบอะไหล่จากคลังจัดเก็บแล้วมีไม่เพียงพอ และต้องทำการขอซื้อต่อไปเพื่อนำมาซ่อมบำรุงในงาน

8) workorder.assignStaff เป็นการระบุ และมอบหมายงานให้เจ้าหน้าที่ซ่อมบำรุงเข้าปฏิบัติงานตามงานซ่อมบำรุงตามหมายเลข WO แต่ละงาน

9) workorder.closeWO เป็นการบันทึกสถานะของ WO เป็นปิดเรื่อง เมื่อปัญหาที่พบได้ทำการซ่อมบำรุงจนเรียบร้อยแล้ว

บทที่ 5
สรุปผลงานวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้เป็นการออกแบบสถาปัตยกรรมของระบบบริหารการบำรุงรักษาเครื่องจักร ตามแนวคิดของ SOA และคลาวด์คอมพิวติง เพื่อให้เป็นแนวคิดสำหรับระบบบริหารการบำรุงรักษาเครื่องจักร ที่มีโครงสร้างสถาปัตยกรรมแบบ Client/Server ซึ่งเป็นการใช้งานในรูปแบบการจัดการแบบรวมศูนย์ (Centralized System) ให้สามารถเปลี่ยนเทคโนโลยีไปใช้งานได้บนสถาปัตยกรรมคลาวด์คอมพิวติง เพื่อมุ่งหวังให้เกิดทางเลือกที่มากขึ้น โดยใช้คลาวด์เซอร์วิส ที่จะช่วยให้มีความยืดหยุ่นในการใช้งานในการเพิ่มและลดจำนวนทรัพยากรที่ใช้โดยระบบ (Scalability) อีกทั้งยังสามารถยืดหยุ่นในการติดตั้งระบบงาน โดยพิจารณากระบวนการบางกระบวนการในระบบงานเพื่อให้ใช้งานบนคลาวด์ได้ โดยการทำงานของระบบบริหารการบำรุงรักษาเครื่องจักร ยังสามารถใช้งานได้สอดคล้องต่อเนื่องกัน ระหว่าง ระบบงานที่ทำงานบน โครงสร้างสถาปัตยกรรมแบบ Client/Server และ ระบบงานที่ทำงานบนโครงสร้างสถาปัตยกรรมคลาวด์คอมพิวติง

แนวคิดของงานวิจัยนี้นำแนวคิดของ SOA ทำให้การพัฒนาซอฟต์แวร์ CMMS สามารถมีแนวทางสำหรับการพัฒนาเพื่อปรับเปลี่ยนซอฟต์แวร์ CMMS จากโครงสร้างสถาปัตยกรรมแบบ Client/Server ไปยังสถาปัตยกรรมคลาวด์คอมพิวติง เพื่อรองรับเทคโนโลยีที่เปลี่ยนแปลงและรองรับอนาคตต่อไปได้

งานวิจัยได้นำเสนอการวิเคราะห์บริการคลาวด์ เพื่อออกแบบสถาปัตยกรรมคลาวด์คอมพิวติงสำหรับระบบการบริหารงานบำรุงรักษา โดยใช้ซอฟต์แวร์ผลิตภัณฑ์ที่มีอยู่มาปรับเข้ากับเทคโนโลยีคลาวด์ การวิเคราะห์บริการคลาวด์อธิบายได้ด้วย UML ซึ่งแบ่งเป็น 5 ขั้นของการพัฒนาสถาปัตยกรรม คือ

- การพัฒนาแผนภาพ Use Case และอธิบายกรณีการใช้ระบบ (Scenario) สำคัญๆ 3 กรณี
- การพัฒนาแผนภาพ Class ซึ่งอธิบายความต้องการของแอปพลิเคชัน
- การพัฒนาแผนภาพ Component
- การอ้างอิงมาตรฐานระบบการจัดการคลาวด์ ในที่นี้ คือ InterCloud
- การพัฒนาโครงสร้างระบบบนคลาวด์และ API

บรรณานุกรม

บรรณานุกรม

[1] J.-S. Cheng et al., "An e-book hub service based on a cloud platform," *The International Review of Research in Open And Distance Learning,* vol. 13, no. 5, pp. 39-55, December 2012.

[2] S. Singh and I. Chana, "Introducing agility in cloud based software development through ASD," *International Journal of u- and e- Service,* Science and Technology, vol. 6, no. 5, pp. 191-202, October 2013.

[3] R. Buyya et al., "InterCloud: utility-oriented federation of cloud computing environments for scaling of application services," *Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing,* ICA3PP 2010, Busan, South Korea, May 21-23, 2010, pp. 13-31.

[4] Wikipedia, *"Service-Oriented Architecture,"* [Online]. Available: http://en.wikipedia.org/wiki/Service-oriented_architecture. [Accessed: March 24, 2015].

[5] A. C. Marquez and J. N. Gupta, "Contemporary maintenance management: process, framework and supporting pillars," *The International of Management Science,* vol. 34, no. 3, pp. 313-326, June 2006.

[6] NIST, *"NIST,"* [Online]. Available: http://www.nist.gov/. [Accessed: March 21, 2015].

[7] F. Liu et al., *NIST Cloud Computing Reference Architecture: Recommendations of the National Institute of Standards and Technology (Special Publication 500-292),* USA: CreateSpace Independent Publishing Platform, 2012.

[8] D. C. Marinescu, *Cloud Computing: Theory and Practice,* Waltham: Elsevier MK, 2013.

[9] S. Zhang et al., "Cloud computing research and development trend," *2010 Second International Conference on Future Networks,* ICFN 2010, Sanya, China, January 22-24, 2010, pp. 93-97.

[10] Google, *"Google Compute Engine,"* [Online]. Available: https://cloud.google.com/compute/. [Accessed: March 24, 2015].

[11] Amazon, *"Amazon EC2,"* [Online]. Available: http://aws.amazon.com/ec2/. [Accessed: March 24, 2015].

[12] Amazon, *"Amazon S3,"* [Online]. Available: http://aws.amazon.com/s3/. [Accessed: March 24, 2015].

[13] Rackspace, *"Rackspace Cloud Servers,"* [Online]. Available: http://www.rackspace.com/cloud/servers. [Accessed: March 24, 2015].

[14] FlexiScale, *"FlexiScale,"* [Online]. Available: http://www.flexiscale.com/. [Accessed: March 24, 2015].

[15] Google, *"Google App Engine,"* [Online]. Available: https://cloud.google.com/appengine/. [Accessed: March 24, 2015].

[16] Apache, *"Apache Stratos,"* [Online]. Available: http://stratos.apache.org/. [Accessed: March 24, 2015].

[17] Microsoft, *"Microsoft Azure,"* [Online]. Available: http://azure.microsoft.com/en-us/. [Accessed: March 24, 2015].

[18] Salesforce, *"Heroku,"* [Online]. Available: https://www.heroku.com/. [Accessed: March 24, 2015].

[19] Salesforce, *"Salesforce,"* [Online]. Available: https://www.salesforce.com/paas/. [Accessed: March 24, 2015].

[20] P. Jamshidi et al., "Cloud migration research: a systematic review," *IEEE Transactions on Cloud Computing,* vol. 1, no. 2, pp. 142-157, October 2013.

[21] Wikipedia, *"การพัฒนาระบบแบบเอจายล์,"* [Online]. Available: http://th.wikipedia.org/wiki/เอจายล์_(การพัฒนาซอฟต์แวร์). [Accessed: March 24, 2015].

[22] W.-T. Tsai et al., "Service-oriented cloud computing architecture," *2010 Seventh International Conference on Information Technology,* ITNG 2010, Las Vegas, USA, April 12-14, 2010, pp. 684-689.

[23] S. Schulte et al., "Towards process support for cloud manufacturing," *2014 IEEE 18th International Enterprise Distributed Object Computing Conference,* EDOC 2014, Ulm, Germany, September 1-5, 2014, pp. 142-149.

[24] S. Karnouskos et al., "A SOA-based architecture for empowering future collaborative cloud-based industrial automation," *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society,* IECON 2012, Montreal, Canada, October 25-28, 2012, pp. 5,766-5,772.

[25] Wikipedia, *"มาตรฐานสากล IEC 62264 สำหรับการบูรณาการระบบควบคุมขององค์กร (Enterprise-control system integration),"* [Online]. Available: http://en.wikipedia .org/wiki/IEC_62264. [Accessed: March 24, 2015].

[26] FMC, *"FMC (Fundamental Modeling Concepts),"* [Online]. Available: http://www.fmc-modeling.org/notation_ref erence. [Accessed: March 24, 2015].

[27] L. Mei et al., "A tale of clouds: paradigm comparisons and some thoughts on research issues," *2008 IEEE Asia-Pacific Services Computing Conference,* APSCC 2008, Yilan, Taiwan, December 9-12, 2008, pp. 464-469.