

การออกแบบสถาปัตยกรรมของระบบบริหารการบำรุงรักษาเครื่องจักร ตามแนวคิดของ SOA และคลาวด์คอมพิวติง

An Architectural Design of Computerized Maintenance Management System towards SOA and Cloud Computing

ชัยรัตน์ ศรีพงษ์ธนากุล¹ กิตติมา เมฆาปัญญากิจ²

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยี ไทย-ญี่ปุ่น กรุงเทพมหานคร

¹sr.chairat_st@tni.ac.th

²kittimame@tni.ac.th

บทคัดย่อ

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

คำสำคัญ: การออกแบบสถาปัตยกรรม, ระบบบริหารการบำรุงรักษา, สถาปัตยกรรมระบบเชิงบริการ, คลาวด์คอมพิวติง, เอพีไอ

ABSTRACT

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 offer operation flexibility. It is an adaptable resource 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 linked and communicated by API.

Keywords: Architectural Design, CMMS, SOA, Cloud Computing, API

1) บทนำ

ในภาคอุตสาหกรรม ซอฟต์แวร์ประเภทระบบการบริหารงานบำรุงรักษา (Computerized Maintenance Management System, CMMS) เป็นซอฟต์แวร์สำหรับการบริหารจัดการงานบำรุงรักษาและซ่อมบำรุง เพื่อช่วยให้ธุรกิจสามารถบริหารจัดการและการติดตามเครื่องจักร, เครื่องมือ หรือ อุปกรณ์ ยานพาหนะ หรือ สินทรัพย์ ต่างๆ ในการบำรุงรักษา และวางแผนเพื่อให้เครื่องมือและอุปกรณ์ต่างๆ ในภาคธุรกิจ สามารถใช้งานได้ต่อเนื่องอย่างมีประสิทธิภาพ และลดการสูญเสียโอกาสในธุรกิจ

ซอฟต์แวร์ระบบการบริหารงานบำรุงรักษา ถือเป็นระบบบริหารจัดการด้านเทคนิคที่ช่วยเหลือเจ้าหน้าที่ในหน่วยงานบำรุงรักษา ในการวางแผนวิเคราะห์เชิงประสบการณ์ เพื่อให้เครื่องจักรมีความพร้อมใช้งาน, ใช้งานเครื่องจักรได้อย่างปลอดภัย, ควบคุมต้นทุนการบำรุงรักษาให้เหมาะสม, เตรียมวางแผนซ่อมบำรุงเชิงป้องกัน, จัดการงานซ่อมประจำวัน รวมถึงทำการบันทึกข้อมูลการซ่อมบำรุง และสืบค้นประวัติงานซ่อมบำรุงพร้อมวิธีแก้ไขที่ผ่านมาแล้ว

ในปัจจุบันซอฟต์แวร์ระบบการบริหารงานบำรุงรักษา ยังคงมีโครงสร้างสถาปัตยกรรมแบบ Client/Server ซึ่งเป็นการใช้งานในรูปแบบการจัดการแบบรวมศูนย์ (Centralized System) โดยมีองค์ประกอบ คือ เครื่องผู้ให้บริการ (Server) และเครื่องผู้ให้บริการ (Client) เชื่อมต่อกันอยู่ในรูปแบบของ Client/Server นั้น

สำหรับเทคโนโลยีที่มีการพัฒนามาจนถึงปัจจุบันเป็นเทคโนโลยีระบบคลาวด์คอมพิวติง (Cloud Computing) มีคุณลักษณะเฉพาะที่มีความยืดหยุ่นสูงในการเพิ่มและลดจำนวนทรัพยากรของระบบตามความต้องการที่จะใช้งานจริง และยังลดต้นทุนในการบริหารจัดการทรัพยากรของผู้ใช้งาน ซึ่งเป็นเทคโนโลยีที่ใช้วิธีการประมวลผลโดย

อ้างอิงความต้องการของผู้ใช้ โดยระบบจะจัดสรรทรัพยากรและการให้บริการที่ตรงกับความต้องการของผู้ใช้งาน โดยระบบสามารถยืดหยุ่น [1] เพิ่มและลดจำนวนทรัพยากร เพื่อให้เหมาะสมกับความต้องการของผู้ใช้งานได้ตลอดเวลา

2) งานวิจัยที่เกี่ยวข้อง

2.1) CMMS ในปัจจุบัน

งานวิจัยของ Marquez และ Gupta [2] ได้ทบทวนแนวคิดต่างๆ กระบวนการและมาตรฐานที่เกี่ยวข้องกับการบริหารการบำรุงรักษา เนื่องจากความซับซ้อนมาก และเสนอกรอบแนวคิดของหน้าที่การบำรุงรักษาในองค์กร โดยมีการใช้ปัจจัยทางด้านคุณลักษณะต่างๆ (เช่น ระบบสารสนเทศที่ใช้ในการบริหารการบำรุงรักษา เทคโนโลยี กระบวนการและการบูรณาการ (Process Technology and Integration) และระบบบริหารการผลิต อย่าง JIT เป็นต้น) ซึ่งมีผลต่อความซับซ้อนในการบำรุงรักษาการดำเนินงาน งานวิจัยนี้ได้วิเคราะห์ทางด้านกลยุทธ์ ยุทธวิธี และทางด้านการดำเนินงานของการบำรุงรักษา แล้วจัดทำโครงสร้างเพื่อช่วยให้การทำงานสมบูรณ์ โดยมีกรอบแนวคิดนี้จัดการบริหารการบำรุงรักษาเป็น 3 ระดับ คือ ระดับกลยุทธ์ (Strategic Level) ระดับยุทธวิธี (Tactical Level) และระดับปฏิบัติการ (Operational Level)

2.2) คลาวด์คอมพิวติง

คลาวด์คอมพิวติง แพร่หลายมากขึ้น เกิดจากการเปลี่ยนแปลงรูปแบบสถาปัตยกรรมคอมพิวเตอร์ จากเมนเฟรม มาเป็นสถาปัตยกรรมแบบ Client/Server จนมาถึงในปัจจุบันเป็นสถาปัตยกรรมคลาวด์คอมพิวติง ซึ่งมีคุณลักษณะเฉพาะที่มีความยืดหยุ่นสูงในการเพิ่มและลดจำนวนทรัพยากรของระบบตามความต้องการที่จะใช้งานจริง และยังลดต้นทุนในการบริหารจัดการทรัพยากรของผู้ใช้งาน ผู้ใช้งานจะสามารถใช้งานด้วยกระบวนการทำงานที่เหมือนเดิมได้ โดยรูปแบบบริการของคลาวด์คอมพิวติง เพื่อสะดวกในการเข้าถึงในเวลาที่ต้องการไปยังเครือข่ายของทรัพยากรคอมพิวเตอร์ (ซึ่งคอนฟิกร์ได้) องค์กรและบุคคลมีส่วนใช้งานร่วมกันได้ (เช่น เครือข่าย เซิร์ฟเวอร์ ระบบจัดเก็บข้อมูล แอปพลิเคชัน และบริการต่างๆ) ที่สามารถจัดเตรียมให้ผู้รับบริการได้อย่างรวดเร็ว และเปิดให้บริการด้วยต้นทุนของการบริหารจัดการ หรือภาระการดำเนินการของผู้ให้บริการน้อยที่สุด

การให้บริการคลาวด์มีรูปแบบของการส่งมอบบริการ (Delivery Models) 3 แบบ ดังนี้ [3-5]

Infrastructure as a Service (IaaS) : IaaS หรือ โครงสร้างพื้นฐานในรูปแบบของบริการ เป็นการให้บริการโครงสร้างพื้นฐานที่เป็นทรัพยากรที่จำเป็น รวมถึงพื้นที่ในการจัดเก็บข้อมูล การเชื่อมต่อเครือข่าย หน่วยความจำ หน่วยประมวลผลกลาง และทรัพยากรอื่นๆ ทางด้านฮาร์ดแวร์ที่จำเป็น ตัวอย่างของ IaaS ได้แก่ Google Compute Engine, Amazon EC2 และ Rackspace Cloud Servers เป็นต้น

Platform as a Service (PaaS) : PaaS หรือ แพลตฟอร์มในรูปแบบของบริการ เป็นการให้บริการเครื่องมือ ในการให้บริการในส่วนการสร้างโปรแกรมประยุกต์เพื่อสนับสนุน และเป็นเครื่องมือในการพัฒนาซอฟต์แวร์ ที่รองรับการพัฒนาทำงานร่วมกันได้ถึงแม้ว่าผู้ร่วม

พัฒนาจะอยู่ต่างสถานที่กัน ตัวอย่างของ PaaS ได้แก่ Google App Engine, Microsoft Azure, Heroku และ Salesforce เป็นต้น

Software as a Service (SaaS) : SaaS หรือ ซอฟต์แวร์ในรูปแบบของบริการ เป็นการให้บริการซอฟต์แวร์ที่มีการติดตั้งอยู่บนโครงสร้างพื้นฐานคลาวด์ ได้แก่ เครือข่าย เซิร์ฟเวอร์ ระบบปฏิบัติการ ระบบจัดเก็บข้อมูล และแอปพลิเคชันต่างๆ เป็นต้น ซึ่งจัดเตรียมโดยผู้ให้บริการคลาวด์คอมพิวติง ผู้ใช้บริการสามารถจะเข้าถึงบริการเหล่านั้นได้ด้วยอุปกรณ์ต่างๆ สำหรับบริการ SaaS สามารถแบ่งออกได้เป็น 2 กลุ่มบริการ ดังนี้

- บริการระดับองค์กร (Enterprise services) เช่น บริการทางด้านห่วงโซ่อุปทาน, บริการจัดการลูกค้าสัมพันธ์ และบริการค้นหา (Search) เป็นต้น
- แอปพลิเคชันในรูปของเว็บ 2.0 เช่น บริการจัดการเมตาดาต้า (Metadata management), บริการเครือข่ายสังคม (Social networking) และบริการเว็บล็อก เป็นต้น

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

- การขยายขนาด และความยืดหยุ่น (Scalability and flexibility)
- การเข้าถึงเครือข่าย (Broad network access)
- วิธีการชำระค่าบริการตามการใช้งาน (Pay per use)
- การลดต้นทุน (Cost effectiveness)

2.3) การพัฒนาซอฟต์แวร์บนคลาวด์

2.3.1) การพัฒนาซอฟต์แวร์และวิศวกรรมซอฟต์แวร์

Singh และ Chana [6] ได้ศึกษาแนวทางการพัฒนาซอฟต์แวร์บนคลาวด์ โดยประยุกต์ใช้แนวคิดของแนวทางการพัฒนาซอฟต์แวร์แบบเอจายล์ (Agile Software Development) สนใจทางด้านวัฏจักรของการพัฒนาที่จะเกิดเป็นช่วงสั้นๆ ในการออกแบบ พัฒนา และทดสอบเป็นอย่างนี้ซ้ำๆ โดยในระหว่างกระบวนการต่างๆ ก็จะทำให้เกิดความรู้ในการที่จะใช้ในการแก้ไขปัญหาซ้ำๆ จากปัจจัยแวดล้อมที่มีผลกระทบ เช่น สภาพแวดล้อมที่จะพัฒนา เทคโนโลยีที่มีความต้องการ ผู้มีส่วนร่วมในการพัฒนา ผู้ขายซอฟต์แวร์ เป็นต้น นำมาซึ่งประสบการณ์ในการแก้ไขปัญหาที่เกิดขึ้น เป็นสถาปัตยกรรมที่สามารถนำมาใช้ใหม่ได้ โดยงานวิจัยของ Singh และ Chana นี้เสนอรูปแบบการพัฒนาคloudแบบปรับตัวได้ (Adaptive Cloud Development (ASD) Model) โดยคาดหวังความรวดเร็วในการนำแอปพลิเคชันไปใช้บนระบบคลาวด์ (Speed of Application Deployment) การใช้เวลาการเข้าสู่ตลาด (Time to market) สั้นลง การลดค่าใช้จ่ายการดำเนินงาน ความสามารถในการนำคอมโพเนนต์บนคลาวด์มาใช้ซ้ำ (Reusability) และง่ายต่อการบำรุงรักษาสำหรับผู้เช่าหรือผู้รับบริการ งานวิจัยเสนอรูปแบบการพัฒนา ASD พร้อมบูรณาการ การปฏิสัมพันธ์กับผู้บริการคลาวด์ (Cloud Provider) ในการยอมรับกันได้นบนคลาวด์คอมพิวติง มุ่งใช้รูปแบบ ASD ในการพัฒนาคloud ตามแนวคิดของ เอจายล์ เพื่อที่จะให้ต้นทุนการพัฒนาต่ำสุด และสามารถปรับปรุงความพึงพอใจของลูกค้าให้ดีขึ้น และเพิ่มการนำคอมโพเนนต์บนคลาวด์มาใช้ซ้ำ (Reusability) ได้

2.3.2) สถาปัตยกรรมระบบที่เกี่ยวข้อง

ในด้านการพัฒนาและใช้คลาวด์ในระบบโรงงานอุตสาหกรรม Tsai และคณะ [7] ได้ทำสำรวจและอิมพลีเมนต์โดยนำเสนอสถาปัตยกรรมคลาวด์คอมพิวติงที่เน้นการบริการ เรียกว่า SOCCA (Service-Oriented Cloud Computing Architecture) เพื่อให้คลาวด์หลายๆ คลาวด์สามารถทำงานร่วมกันได้ (Interoperation) , การออกแบบในระดับสูง (High Level Designs) เพื่อให้การสนับสนุนที่ดีขึ้นสำหรับคุณสมบัติผู้ใช้หลายรายบนแอปพลิเคชันเดียวกัน (Multi-Tenancy Feature)

สถาปัตยกรรมระบบที่เสนอโดยงานวิจัยนี้เอื้ออำนวยให้ แอปพลิเคชันหนึ่งๆ รันบนคลาวด์ได้หลายคลาวด์และทำงานประสานร่วม (Interoperate) กับคลาวด์อื่นๆ สถาปัตยกรรม SOCCA [7] ออกแบบโดยแบ่งเป็น 4 ระดับชั้น 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) โดยมุ่งเน้นให้เป็นแพลตฟอร์มเปิด (Open Platform) ที่สามารถควบคุมและบำรุงรักษาได้ด้วยมาตรฐานเปิด (Open Standards) และออนโทโลยี (Ontology)

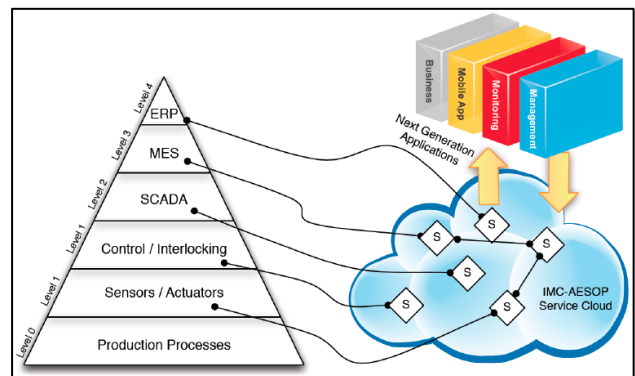
Schulte และคณะ [8] ได้กำหนดความต้องการของกรอบแนวคิดซอฟต์แวร์สำหรับ Cloud Manufacturing มีสมมติฐานว่า กรอบแนวคิดซอฟต์แวร์นั้น จำเป็นต้องสนับสนุนทุกเฟสของการบริหารกระบวนการธุรกิจ (BPM Lifecycle) คือ การวิเคราะห์และออกแบบ การคอนฟิก การกำหนดกฎ และการประเมินผล ความต้องการของกรอบแนวคิดซอฟต์แวร์สามารถจะอธิบายโดยย่อได้ดังนี้

- ออกแบบแบบจำลองกระบวนการ (Process Models) และเตรียมแบบจำลองกระบวนการ
- การคอนฟิกกระบวนการผลิต (Configuration of Manufacturing processes) ขั้นตอนการที่จะอิมพลีเมนต์
- การเฝ้าระวังเหตุการณ์วิกฤต ในขั้นตอนของกระบวนการ
- มีการเก็บรวบรวมข้อมูลการทำงาน เพื่อใช้ในการวิเคราะห์และปรับรูปแบบจำลองกระบวนการได้
- BPMS สำหรับ Cloud Manufacturing จำเป็นที่จะต้องมีความรู้เพื่อการตัดสินใจในแต่ละขั้นตอนของ Lifecycle ความรู้เกี่ยวกับทรัพยากรและสินทรัพย์ภายในสามารถถูกเก็บไว้ในฐานข้อมูลขององค์กร ในขณะที่ความรู้เกี่ยวกับสินทรัพย์ภายนอกอาจมีให้บริการโดย Marketplaces ซึ่งเป็นส่วนหนึ่งของระบบเว็บธุรกิจทางการผลิต (Manufacturing Business Web) ที่มีสภาพแวดล้อมคลาวด์ซึ่งนำองค์กรคู่ค้าในห่วงโซ่อุปทานมาพบกันโดยมีการจัดเตรียมโครงสร้างพื้นฐานคลาวด์แอปพลิเคชัน คอนเทนต์ และการเชื่อมโยงถึงกัน ไว้ให้

นอกจากนี้ งานวิจัยของ Schulte และคณะ [8] ยังรวบรวมหัวข้อที่เกี่ยวข้องและน่าสนใจ ที่มีความท้าทายต่อการวิจัยในอนาคต เกี่ยวกับเรื่องต่อไปนี้

- การอธิบายบริการและการจำลองกระบวนการ (Process Modeling and Service Descriptions)
- ตลาดการบริการ (Service Marketplaces)
- การเฝ้าติดตามกระบวนการ (Process Monitoring)
- ความวางใจและความปลอดภัยของข้อมูล (Trust and Data Security)

ในระบบอัตโนมัติอุตสาหกรรม (Industrial Automation systems) Karnouskos และคณะ [9] ชี้ปัญหาด้านสถาปัตยกรรมระบบในเรื่องระบบอัตโนมัติอุตสาหกรรม และคาดการณ์ว่า ระบบจะมีความซับซ้อน ซึ่งจะช่วยให้ระบบมีความสามารถในยุคหน้าในด้านการนำไปประยุกต์และด้านเซอร์วิสต่างๆ โรงงานอุตสาหกรรมในอนาคตจะพึ่งพาระบบนิเวศขนาดใหญ่ของระบบ (Ecosystem of Systems) ซึ่งเกิดการร่วมประสานกันในระดับสูงและปริมาณมาก ("collaboration at large scale") ทั้งนี้ก็คาดการณ์ได้ว่า อนาคตจะมีระบบอัตโนมัติแบบกระจายที่ฉลาดขึ้น ใช้ซ้ำได้ ทนทานต่อความผิดพลาด (Fault-Tolerant) ที่เกิดขึ้น ในด้านกำลังความสามารถ (Capabilities, Functionalities and Structural Characteristics) เป็นเซอร์วิสบนคลาวด์ ดังแสดงในรูปที่ 1



รูปที่ 1: ภาพรวมของระบบอุตสาหกรรมในอนาคต ที่มีการประกอบเซอร์วิสบนคลาวด์ แหล่งที่มา : Karnouskos และคณะ [9]

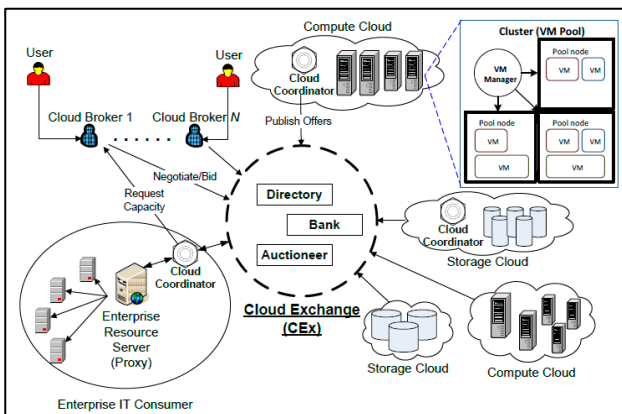
นอกจากนี้ มีสถาปัตยกรรมและส่วนประกอบของ InterCloud ที่น่าสนใจ ที่กล่าวไว้ในงานวิจัยของ Buyya และคณะ [10] ได้นำเสนอสถาปัตยกรรมคลาวด์คอมพิวติงแบบรวมกลุ่มพันธมิตรเพื่อให้บริการโปรแกรมประยุกต์ที่รองรับการขยายตัวได้ ในภาพรวม ได้นำเสนอการขยายตัวของผู้ให้บริการคลาวด์ โดยมีสถานที่ตั้งของศูนย์ข้อมูลที่อยู่ในภูมิภาคต่างๆกัน ให้สามารถมารวมกลุ่มกันเพื่อให้บริการมีตัวกลางในการเชื่อมโยงการบริการ ระหว่างผู้ให้บริการคลาวด์และผู้ใช้งาน โดยผ่าน 3 องค์ประกอบหลักทำงานร่วมกันได้อย่างสอดคล้องกัน ตามการใช้งานระหว่างผู้ใช้งานและผู้ให้บริการคลาวด์ ให้คลาวด์แต่ละคลาวด์สามารถทำงานร่วมกันได้ โดยผ่านการจัดการโดย ผู้แลกเปลี่ยนคลาวด์ (CEX) เป็นผู้ให้การสนับสนุนสำหรับการจัดการความต้องการของผู้ใช้งาน รวมถึงการจัดการรายการการให้บริการของผู้ให้บริการคลาวด์ โดยผู้ใช้งานสามารถเรียกใช้งานโปรแกรมประยุกต์

สถาปัตยกรรมระบบที่นำเสนอโดยงานวิจัยนี้ทำให้โปรแกรมประยุกต์หนึ่งๆ สามารถให้บริการได้จากหลายๆคลาวด์ โดยสถาปัตยกรรมนี้จะสนับสนุนการให้บริการโดยมีการขยายตัวของคลาวด์และผู้ใช้งาน ได้มากขึ้นโดยมีความยืดหยุ่นสูง และรองรับการขยายตัว

สูงขึ้น รูปที่ 2 แสดงให้เห็นถึงกลุ่มเครือข่ายการให้บริการโดยผู้ให้บริการคลาวด์ โดยมีตัวกลางเป็น ผู้แลกเปลี่ยนคลาวด์ ตามความต้องการใช้งานของผู้ใช้งานแต่ละคนที่มีความต้องการใช้งานโปรแกรมประยุกต์ที่แตกต่างกันออกไป และแสดงให้เห็นถึงองค์ประกอบอื่นๆ ที่มีในอินเทอร์เน็ตคลาวด์ (InterCloud)

Buyya และคณะ [10] ได้กำหนดองค์ประกอบในสถาปัตยกรรมของ InterCloud สามารถจะอธิบายตามองค์ประกอบที่มีได้ดังนี้

- ผู้ประสานงานคลาวด์ (Cloud Coordinator, CC) มีบทบาทเป็นผู้โต้ตอบในการให้บริการ โดยจะเป็นผู้รับผิดชอบในส่วนของการทรัพยากรบนคลาวด์ที่รับผิดชอบ ทำการจัดเตรียมโปรแกรมประยุกต์ และสภาพแวดล้อมของโปรแกรมประยุกต์นั้นๆ จากความต้องการที่มีการร้องขอที่ส่งผ่านมาจากผู้แลกเปลี่ยนคลาวด์ (CEX)
- นายหน้าคลาวด์ (Cloud Broker, CB) มีบทบาทเป็นผู้ระบุผู้ให้บริการคลาวด์ที่เหมาะสมให้กับผู้ใช้งานที่มีการเรียกใช้บริการ โดยทำการสื่อสารผ่านผู้แลกเปลี่ยนคลาวด์ (Cloud Exchange, CEX)
- ผู้แลกเปลี่ยนคลาวด์ (CEX) มีบทบาทเป็นผู้รวบรวมข้อมูลของการให้บริการคลาวด์จากผู้ให้บริการคลาวด์ที่อยู่ในกลุ่มพันธมิตร และทำการจับคู่การให้บริการ โดยให้ข้อมูลกับ นายหน้าคลาวด์ (CB) เพื่อทำการค้นหาทรัพยากรที่เหมาะสมจากข้อมูลผู้ให้บริการคลาวด์ที่มีข้อมูลอยู่



รูปที่ 2: เครือข่ายกลุ่มให้บริการคลาวด์ โดยมีตัวกลางเป็น ผู้แลกเปลี่ยนคลาวด์
แหล่งที่มา : Rajkumar Buyya และคณะ [10]

3) การวิเคราะห์และออกแบบสถาปัตยกรรม ตามแนวคิด SOA

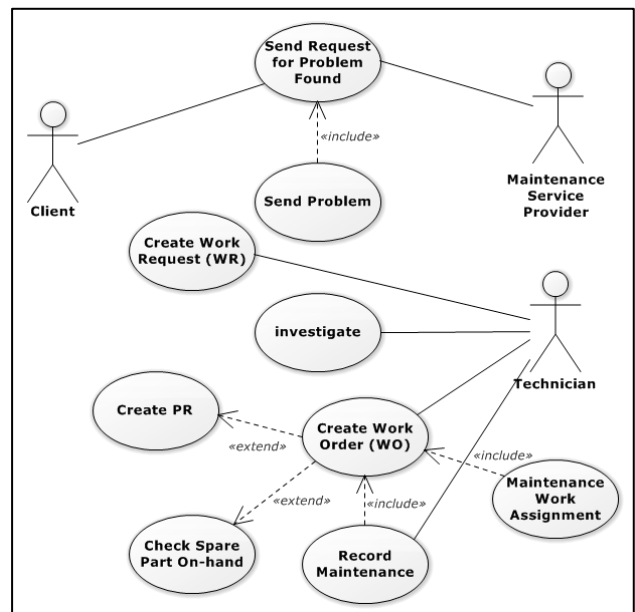
3.1) การดำเนินการศึกษาและวิเคราะห์ระบบงาน CMMS

การดำเนินการศึกษาระบบงาน CMMS ที่มีอยู่เดิมพบว่ามีการพัฒนาด้วยภาษาวิซวลเบสิก (Visual Basic) และมีโครงสร้างสถาปัตยกรรมแบบ Client/Server ซึ่งในกระบวนการหนึ่งในการบริหารการบำรุงรักษา คือ WO (Work Order) โดย WO นั้นมีผู้ใช้งาน (Actor) 3 ประเภท คือ ผู้รับบริการหรือลูกค้า (Client) ผู้ให้บริการงานซ่อมบำรุง (Maintenance Service Provider) และช่างเทคนิค (Technician) กระบวนการทำงานต่างๆ ตั้งแต่เริ่มต้นจนจบกระบวนการทั้งหมด จะมีユースケースต่างๆ ซึ่งสามารถสามารถอธิบายกรณีการใช้ระบบปกติ (Scenario) ได้ 3 กรณี ดังนี้

- กรณีที่ 1 กระบวนการเพื่อการซ่อมบำรุงโดยไม่เบี่ยงอะไหล่
- กรณีที่ 2 กระบวนการเพื่อการซ่อมบำรุงโดยมีการเบี่ยงอะไหล่คงคลัง
- กรณีที่ 3 กระบวนการทำงานเพื่อการซ่อมบำรุงโดยมีการใช้อะไหล่คงคลังแต่ไม่เพียงพอ

เหตุการณ์ที่เกิดขึ้นของกระบวนการทำงานของการบันทึกคำสั่งงานซ่อม นั้นจะมีกระบวนการหลัก ที่อย่างน้อยต้องดำเนินการสรุปได้เป็น 6 ยูสเคสเป็นอย่างน้อยในการดำเนินงานของการบันทึกคำสั่งงานซ่อม ให้เสร็จตั้งแต่ต้นจนจบกระบวนการ ในรูปที่ 3 คือ

- ผู้รับบริการหรือลูกค้าส่งคำร้องขอเมื่อพบปัญหา (Send Request for Problem Found)
- ช่างเทคนิคบันทึกคำร้องขอ (Create Work Request)
- ช่างเทคนิคตรวจสอบและวินิจฉัยเบื้องต้น (Investigate)
- ช่างเทคนิคบันทึกคำสั่งงานซ่อม (Create Work Order หรือ WO)
- ช่างเทคนิคมอบหมายงานเพื่อดำเนินการซ่อมบำรุง (Maintenance Work Assignment)
- ช่างเทคนิคบันทึกการซ่อมบำรุง (Record Maintenance)



รูปที่ 3: แผนภาพ Use Case สำหรับกระบวนการคำสั่งซ่อม (Work Order Process)

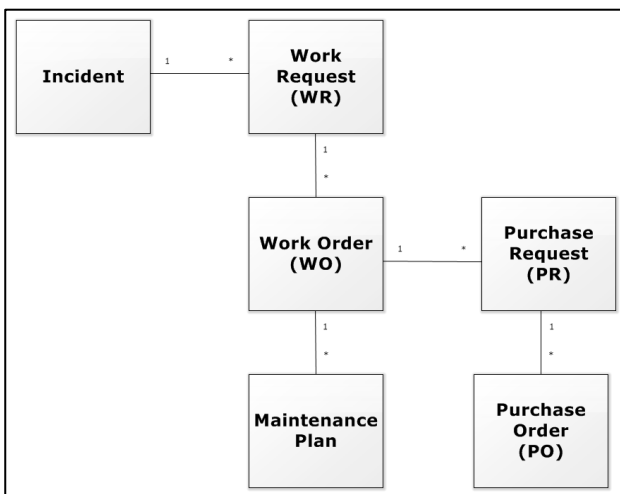
3.2) การพิจารณา CMMS เพื่อปรับตัวขึ้นไปใช้งานบนคลาวด์

ในกรณีของ CMMS ซึ่งต้องการปรับตัวขึ้นไปใช้งานบนคลาวด์ มีการตัดสินใจในการเลือกโมดูลบันทึกคำสั่งงานซ่อม (WO) จากที่พบว่า มี Domain Class ต่างๆ จำนวนมากแต่ที่สนใจสำหรับ Class ที่จะใช้บนคลาวด์ได้นั้น มีเพียง 6 Domain Class เท่านั้นในรูปที่ 4 ที่มีในส่วนของ WO โดยจากที่เนื่องจากการบันทึกคำสั่งงานซ่อม เป็นการระบุความต้องการให้กับช่างเทคนิคกลุ่มที่มีความเชี่ยวชาญด้านใดในปัญหานั้น ให้เข้าแก้ไขและซ่อมบำรุงงานนั้นๆ ให้ลุล่วงไปด้วยดี อีกทั้งกรณีที่องค์กรไม่ได้มีเจ้าหน้าที่เทคนิคในงานแก้ไขนั้นๆ ก็จะทำให้สามารถส่งคำสั่งซ่อม (WO) ไปยังผู้ให้บริการงานซ่อมบำรุง (Maintenance

Service Provider) ได้อย่างสะดวก แต่ทั้งนี้โมดูลอื่นๆ ที่เกี่ยวข้องในระบบ CMMS ก็สามารถที่จะปรับตัวไปใช้งานบนคลาวด์ได้เช่นเดียวกัน

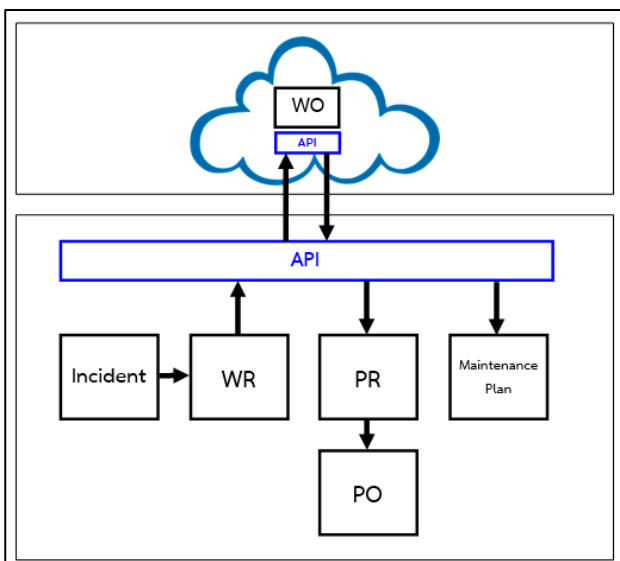
3.3) การออกแบบสถาปัตยกรรม ตามแนวคิด SOA

การดำเนินงานในส่วนของโมดูลบันทึกคำสั่งงานซ่อม (WO) ต้องเตรียมการในส่วนของการพัฒนา คือ ปรับปรุงความสามารถของระบบ CMMS เดิมให้สามารถใช้งานร่วมกับโมดูลบันทึกคำสั่งงานซ่อม (WO) ที่นำไปใช้งานบนคลาวด์ โดยเตรียมตัวกลางที่ทำการเชื่อมต่อระหว่างระบบ CMMS เดิมที่อยู่บนสถาปัตยกรรมแบบ Client/Server และโมดูลบันทึกคำสั่งงานซ่อมที่จะปรับขึ้นไปใช้งานบนคลาวด์ ที่มีสถาปัตยกรรมระบบเชิงบริการ (Service-Oriented Architecture) งานวิจัยจึงได้พัฒนา API ในรูปที่ 5 เพื่อการเชื่อมต่อและส่งผ่านการทำงานให้เข้ากันได้และเหมาะสมระหว่างทั้งสองระบบ [11-12]



รูปที่ 4: Domain Class Diagram สำหรับ CMMS

ในส่วนของการดำเนินงานหลังจากนั้นจะต้องทำการทดสอบเพื่อให้แน่ใจว่าหลังจากที่นำโมดูลบันทึกคำสั่งงานซ่อมที่เป็นส่วนหนึ่งของระบบ CMMS ให้สามารถใช้งานได้บนคลาวด์แล้ว โมดูลบันทึกคำสั่งงานซ่อมที่อยู่บนระบบ CMMS ก็จะต้องทำงานได้โดยไม่ติดปัญหาเช่นกัน



รูปที่ 5: Cloud Based Component Development

3.4) การออกแบบ API

จากการศึกษาได้ออกแบบ API ซึ่งสนับสนุนการทำงานของทั้ง 8 ยูสเคส (Use Case) ในตารางที่ 1 เพื่อให้การทำงานของซอฟต์แวร์ CMMS ที่ทำงานอยู่บนทั้งสองสถาปัตยกรรมเชื่อมโยงข้อมูลถึงกันได้

ตารางที่ 1: API ของระบบ CMMS

Use Case	API
1. ผู้รับบริการหรือลูกค้าส่งคำร้องขอเมื่อพบปัญหา (Send Request for Problem Found)	incident.receiveProblem (CompanyID, MachineID, IncidentDate, IncidentTitle, FromStaff, ProblemMsg)
2. ช่างเทคนิคบันทึกคำร้องขอ (Create Work Request)	workrequest.createWR (CompanyID, RefIncID)
3. ช่างเทคนิคตรวจสอบและวินิจฉัยเบื้องต้น (Investigate)	workrequest.investigateProb (RefWRID) workrequest.closeWR (RefWRID)
4. ช่างเทคนิคบันทึกคำสั่งงานซ่อม (Create Work Order)	workorder.createWO (RefWRID)
5. ช่างเทคนิคตรวจสอบอะไหล่คงเหลือ (Check Spare Part On-Hand)	workorder.createPartList (RefWOID, PartList)
6. ช่างเทคนิคการจัดทำคำสั่งซื้ออะไหล่ต่อไป (Create Purchase Request)	workorder.createPR (RefWOID, PartListO)
7. ช่างเทคนิคมอบหมายงานเพื่อดำเนินการซ่อมบำรุง (Maintenance Work Assignment)	workorder.assignStaff (RefWOID, StaffID)
8. ช่างเทคนิคบันทึกการซ่อมบำรุง (Record Maintenance)	workorder.closeWO (RefWOID)

จากการออกแบบ API ที่จะใช้งาน ให้รองรับกระบวนการซ่อมบำรุงตามที่มียูสเคสจำนวน 8 ยูสเคส ของการสั่งงานซ่อม (Create Work Order (WO)) เป็นดังนี้

- ผู้รับบริการหรือลูกค้าส่งคำร้องขอเมื่อพบปัญหา (Send Request for Problem Found)
- ช่างเทคนิคบันทึกคำร้องขอ (Create Work Request หรือ WR)
- ช่างเทคนิคตรวจสอบและวินิจฉัยเบื้องต้น (Investigate)
- ช่างเทคนิคบันทึกคำสั่งงานซ่อม (Create Work Order หรือ WO)
- ช่างเทคนิคตรวจสอบอะไหล่คงเหลือ (Check Spare Part On-Hand)
- ช่างเทคนิคการจัดทำคำสั่งซื้ออะไหล่ต่อไป (Create Purchase Request)
- ช่างเทคนิคมอบหมายงานเพื่อดำเนินการซ่อมบำรุง (Maintenance Work Assignment)
- ช่างเทคนิคบันทึกการซ่อมบำรุง (Record Maintenance)

4) สรุป

จากการศึกษาการออกแบบสถาปัตยกรรมของระบบบริหารการบำรุงรักษาเครื่องจักร ตามแนวคิดของ SOA และคลาวด์คอมพิวติง เพื่อให้เป็นแนวคิดสำหรับระบบบริหารการบำรุงรักษาเครื่องจักร ที่มีโครงสร้างสถาปัตยกรรมแบบ Client/Server ซึ่งเป็นการใช้งานในรูปแบบการจัดการแบบรวมศูนย์ให้สามารถเปลี่ยนเทคโนโลยีไปใช้งานได้บนสถาปัตยกรรมคลาวด์คอมพิวติง เพื่อมุ่งหวังให้เกิดทางเลือกที่มากขึ้น โดยใช้คลาวด์เซอร์วิส ที่จะช่วยให้มีความยืดหยุ่นในการใช้งานในการเพิ่มและลดจำนวนทรัพยากรที่ใช้โดยระบบ (Scalability) อีกทั้งยังสามารถยืดหยุ่นในการติดตั้งระบบงาน โดยพิจารณาบางกระบวนการในระบบงาน เพื่อให้ใช้งานบนคลาวด์ได้ โดยการทำงานของระบบบริหารการบำรุงรักษาเครื่องจักร ยังสามารถใช้งานได้สอดคล้องต่อเนื่องกัน ระหว่างระบบงานที่ทำงานบนโครงสร้างสถาปัตยกรรมแบบ Client/Server และระบบงานที่ทำงานบนสถาปัตยกรรมคลาวด์คอมพิวติง

จากการศึกษาได้นำแนวคิดของ SOA ในการพัฒนาซอฟต์แวร์ CMMS สามารถมีแนวทางสำหรับการพัฒนาเพื่อปรับเปลี่ยนซอฟต์แวร์ CMMS จากโครงสร้างสถาปัตยกรรมแบบ Client/Server ไปยังสถาปัตยกรรมคลาวด์คอมพิวติง เพื่อรองรับเทคโนโลยีที่เปลี่ยนแปลงในการออกแบบสถาปัตยกรรมคลาวด์คอมพิวติงสำหรับระบบการบริหารงานบำรุงรักษา ซึ่งเป็นผลิตภัณฑ์ซอฟต์แวร์ที่มีอยู่มาปรับเข้ากับเทคโนโลยีคลาวด์ การวิเคราะห์การคลาวด์อธิบายได้ด้วย UML โดยแบ่งออกเป็น 5 ขั้นตอนของการพัฒนาสถาปัตยกรรม ดังนี้

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

กิตติกรรมประกาศ

ผู้วิจัยขอขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ รองศาสตราจารย์ ดร. วีระ บุญจริง, ดร. ภาสกร อภิรักษ์วรพินิต และดร. กันติชา กิตติพิรัชล ที่ให้คำแนะนำ และตรวจแก้ไขงานวิจัยให้สมบูรณ์ยิ่งขึ้น

เอกสารอ้างอิง

- [1] Jinn-Shing C., Huang E., and Chuan-Lang L., "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, 2012.
- [2] A. Crespo Marquez and J. N. D. Gupta, "Contemporary maintenance management: process, framework and supporting pillars," *Omega*, vol. 34, no. 3, pp. 313-326, Jun. 2006.
- [3] Marinescu D. C., "Cloud computing delivery models and services," in *Cloud Computing Theory and Practice*, Waltham: Elsevier, 2013, pp. 11-14.

- [4] F. Liu et al., "NIST Cloud Computing Reference Architecture," *Special Publication (NIST SP) - 500-292*, 2011.
- [5] S. Zhang, S. Zhang, X. Chen, and X. Huo, "Cloud Computing Research and Development Trend," in *2010 Second International Conference on Future Networks*, Jeju Island, Korea (South), Jun. 2010, pp. 93-97.
- [6] Singh S. and Chana I., "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, 2013.
- [7] W. T. Tsai, X. Sun, and J. Balasooriya, "Service-Oriented Cloud Computing Architecture," in *2010 Seventh International Conference on Information Technology: New Generations*, Las Vegas, NV, USA, 2010, pp. 684-689.
- [8] Schulte S., Hoenisch P., Hochreiner C., Dustdar S., Klusch M. And Schuller D., "Towards Process Support for Cloud Manufacturing," in *2014 IEEE 18th International Enterprise Distributed Object Computing Conference*, 2014, pp. 142-149.
- [9] Karnouskos S., Colombo A. W., Bangemann T., Manninen K., Camp R., Tilly M., Stluka P., Jammes F., Delsing J. and Eliasson J., "A SOA-base architecture for empowering future collaborative cloud-based industrial automation," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society (IECON 2012)*, Montreal, 25-28 Oct 2012, pp. 5766-5772.
- [10] R. Buyya, R. Ranjan, and R. N. Calheiros, "InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services," presented at the *International Conference on Algorithms and Architectures for Parallel Processing*, 2010, pp. 13-31.
- [11] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud Migration Research: A Systematic Review," *IEEE Transactions on Cloud Computing*, vol. 1, no. 2, pp. 142-157, Jul. 2013.
- [12] L. Mei, W. K. Chan, and T. H. Tse, "A Tale of Clouds: Paradigm Comparisons and Some Thoughts on Research Issues," in *2008 IEEE Asia-Pacific Services Computing Conference*, 2008, pp. 464-469.