บทที่ 2

แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

ในการวิจัยครั้งนี้ ผู้วิจัยได้รวบรวมแนวความคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง ในการจัดทำ งานวิจัยเรื่อง ระบบนายหน้าโดยใช้ออนโทโลจีเพื่อภาวะการดำเนินการร่วมกันของการคำนวณแบบคลาวด์หลายแพลตฟอร์ม โดยได้นำเสนอภายใต้หัวข้อดังต่อไปนี้

- การคำนวณแบบคลาวด์
- แพลตฟอร์มบริการโครงสร้างพื้นฐานคลาวด์
- การทำงานร่วมกันได้ของการคำนวณแบบคลาวด์
- วิศวกรรมออนโทโลจี
- งานวิจัยที่เกี่ยวข้อง

แนวคิดที่เกี่ยวข้อง

1.การคำนวณแบบคลาวด์

การคำนวณแบบคลาวด์ เป็นตัวแบบที่ทำให้การเข้าถึงแหล่งรวมทรัพยากรการคำนวณที่ใช้ร่วมกันและปรับแต่งโครงแบบได้อันได้แก่ เครือข่าย, เครื่องให้บริการ, หน่วยเก็บ, โปรแกรมประยุกต์, และบริการ สามารถดำเนินการผ่านเครือข่ายตามคำขอได้อย่างสะดวก ซึ่งทรัพยากรเหล่านี้สามารถจัดหาและคืนได้อย่างรวดเร็วโดยอาศัยการจัดการและการโต้ตอบกับผู้ให้บริการน้อยที่สุด (Mell และ Grance, 2011) องค์ประกอบสำคัญของการคำนวณแบบคลาวด์ได้แก่ บริการซอฟต์แวร์คลาวด์ และการคำนวณสาธารณูปโภคซึ่งหมายถึงการให้บริการโครงสร้างพื้นฐานคลาวด์หรือบริการแพลตฟอร์มคลาวด์ (Armbrust et al., 2009)

สถาบันมาตรฐานและเทคโนโลยีแห่งชาติของสหรัฐอเมริกา ได้นิยามไว้ว่าการคำนวณแบบคลาวด์ เป็นตัวแบบที่มีการเปิดใช้งานแบบยูบิควิตัส (Ubiquitous) ที่ไม่ว่าผู้ใช้งานอยู่ที่ไหน ก็สามารถเข้าถึงสภาพแวดล้อมของระบบได้ผ่านอุปกรณ์การสื่อสารต่างๆ โดยมีลักษณะที่ง่ายต่อการใช้งาน และมีการใช้งานโดยการใช้ทรัพยากรคอมพิวเตอร์ร่วมกันผ่านระบบเครือข่ายคอมพิวเตอร์แบบตามความต้องการของผู้ใช้งาน (on-demand) โดยที่การใช้งานในการบริหารจัดการสามารถทำได้ง่ายและสะดวกรวดเร็ว หรือการทำงานร่วมกันของผู้ใช้บริการบนระบบการคำนวณเดียวกัน โดยรูปแบบการคำนวณแบบคลาวด์นี้ประกอบด้วยลักษณะสำคัญดังนี้

การคำนวณแบบคลาวด์ ผู้ใช้บริการหรือผู้บริโภคสามารถจัดหาความสามารถด้านคอมพิวเตอร์เพียงฝ่ายเดียวเช่นเวลาของเครื่องให้บริการและเครือข่ายการจัดเก็บข้อมูลตามที่ต้องการ โดยอัตโนมัติโดยไม่ต้องมีการปฏิสัมพันธ์ของมนุษย์กับแต่ละผู้ให้บริการ (on-demand self-service.)

การคำนวณแบบคลาวด์ ต้องทำงานบนเครือข่ายคอมพิวเตอร์ ที่สามารถเข้าถึงได้ตามมาตรฐานการสื่อสาร ที่รองรับการใช้งานแพลตฟอร์มที่ต่างกันของลูกค้า เช่น การใช้งานด้วยโทรศัพท์มือถือ แท็บเล็ต แล็ปท็อป และเวิร์กสเตชัน

ในการคำนวณแบบคลาวด์ การรวมเครื่องคอมพิวเตอร์หลายๆ เครื่องไว้เป็นกลุ่มของเครื่องคอมพิวเตอร์สำหรับให้บริการ โดยส่วนใหญ่จะให้บริการเกี่ยวกับ การให้บริการประมวลผลข้อมูล การให้บริการพื้นที่จัดเก็บข้อมูล การให้บริการเชื่อมโยงเครือข่ายคอมพิวเตอร์ และการเข้าถึงการใช้งานอุปกรณ์ต่างๆ ซึ่งการคำนวณแบบคลาวด์จะตอบสนองต่อความต้องการของผู้ใช้งานโดยจำเป็นต้องมีคุณสมบัติ ที่สามารถขยายตัวได้อย่างยืดหยุ่นในการคอมพิวเตอร์ข้อมูลและการจัดเก็บข้อมูลโดยเน้นการจัดสรรทรัพยากรได้จากระยะไกลและง่ายต่อการเข้าถึง และให้บริการทรัพยากรในระบบคอมพิวเตอร์โดยสามารถปรับเปลี่ยนได้ ตามความต้องการของผู้ใช้งาน โดยการให้บริการคำนวณแบบคลาวด์ มีตัวแบบการบริการสามารถแบ่งได้ตามลักษณะการให้บริการเป็น 3 ตัวแบบดังนี้ (NIST, 2015)

1) ตัวแบบบริการแพลตฟอร์ม (Platform as a Service: PaaS) เป็นการให้บริการในการอำนวยความสะดวกสำหรับการสนับสนุนการสร้างและการทำงานของซอฟต์แวร์ที่สามารถทำงานในรูปแบบที่กำหนดโดยส่วนใหญ่จะให้บริการผ่านเทคโนโลยีเว็บ โดยการใช้งานจะต้องได้รับการพัฒนาด้วยแพลตฟอร์มที่ถูกกำหนด โดยสภาพแวดล้อมที่รองรับผู้เช่าหรือผู้ใช้งานได้หลายคนและ รองรับสถาปัตยกรรมแบบมัลติเทียร์ที่ปรับเปลี่ยนได้อย่างไม่มีขีดจำกัดโดยมีผู้ให้บริการตัวอย่างเช่น อาเมซอลเว็บเซอร์วิส (Amazon Web Services: AWS) กูเกิลแอพเอนจิน (Google App Engine) และไมโครซอฟต์อาเซอร์แอพเซอร์วิส (Microsoft Azure App Service)

2) ตัวแบบบริการซอฟต์แวร์ (Software as a Service: SaaS) เป็นการให้บริการโดยวิธีการเสนอซอฟต์แวร์ที่ได้รับอนุญาตให้เข้าถึงซอฟแวร์ได้หลายเครื่องคอมพิวเตอร์ลูกข่าย จากระยะไกลเป็นบริการโดยส่วนใหญ่เป็นการให้บริการในรูปแบบเว็บ (Web-based) โดยการให้บริการจะเป็นแบบการเรียกเก็บเงินตามการใช้งานจริง สภาพแวดล้อมที่รองรับผู้เช่าหรือผู้ใช้งานได้หลายคน รองรับสถาปัตยกรรมที่ปรับเปลี่ยนได้อย่างไม่มีขีดจำกัดโดยในปัจจุบันมีผู้ให้บริการ ตัวอย่างเช่น เซลล์ฟอร์ส (Salesforce.com), ไมโครซอฟต์ ออนไลน์ เซอร์วิส (Microsoft online services) และ ออราเคิล ซีอาร์เอ็ม ออนดีมานด์ (oracle CRM on demand)

3) ตัวแบบบริการโครงสร้างพื้นฐาน (Infrastructure as a Service : IaaS) เป็นการให้บริการของระบบคอมพิวเตอร์เป็นการให้บริการที่เกี่ยวกับการให้บริการพื้นฐานด้านเทคโนโลยีคอมพิวเตอร์สำหรับผู้ใช้บริการ โดยจะให้บริการแบบเสมือนจริงของอุปกรณ์ โดยที่ผู้ใช้บริการสามารถปรับเปลี่ยนขนาดได้ตามความต้องการใช้งาน โดยการคิดค่าบริการสามารถเรียกเก็บเงินตามการใช้งาน โดยระบบจะสามารถให้บริการได้หายผู้ใช้งานในเวลาเดียวกันสามารถจัดการให้บริการรองรับระบบปฏิบัติการ หรือ ซอฟต์แวร์ประยุกต์ได้หลายแพลตฟอร์มโดยมีผู้ให้บริการตัวอย่างการบริการโครงสร้างพื้นฐานคลาวด์เช่น อาเมซอลเว็บอีลาสติคคอมพิวคลาวด์ (Amazon elastic compute cloud: EC2) กูเกิลคอมพิวเอนจิน (Google Compute Engine) และ ไมโครซอฟต์อาเซอร์ (Microsoft azure) งานวิจัยนี้เน้นศึกษาตัวแบบบริการนี้

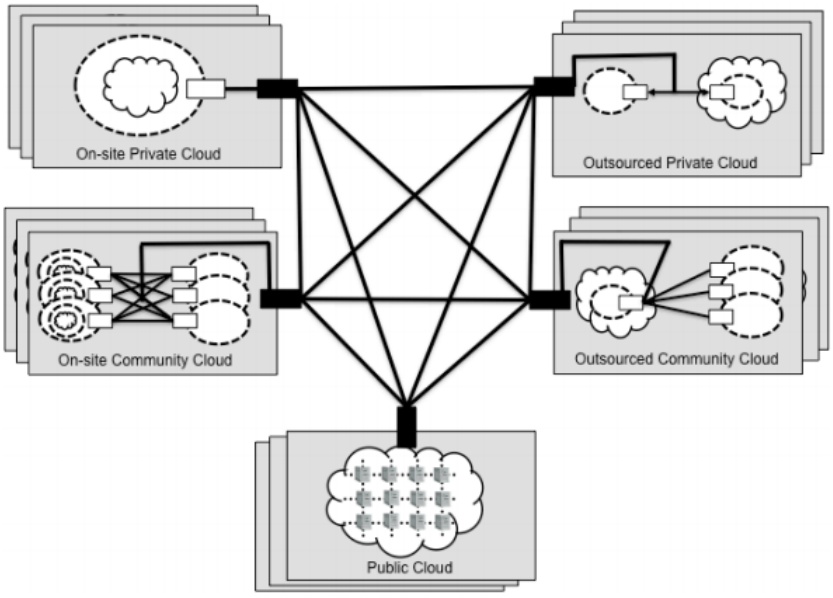
ในส่วนของการติดตั้งตัวแบบคลาวด์สามารถแบ่งเป็นสี่ตัวแบบการติดตั้งใช้งาน (deployment models) (NIST, 2015) ดังต่อไปนี้

1) ตัวแบบคลาวด์ชุมชน (Community Cloud) คือระบบการคำนวณแบบคลาวด์ที่มีการจัดเตรียมโครงสร้างพื้นฐานคลาวด์สำหรับการใช้งานโดยเฉพาะชุมชนที่เฉพาะเจาะจงในรูปแบบกลุ่มสมาชิกที่มีข้อตกลงร่วมกัน เช่น ภารกิจ นโยบาย คามต้องการความปลอดภัย และอื่นๆ ซึ่งอาจจะเป็นองค์การเดียวบริหารจัดการเองหรือมากกว่าหนึ่งขององค์การ โดยอาจะติดตั้งนอกที่ตั้ง (off-premise private cloud) หรือติตตั้งแบบในที่ตั้ง (on-premise private cloud) (NIST, 2015)

2) ตัวแบบคลาวด์สาธารณะ (Public Cloud) คือตัวแบบการติดตั้งใช้งานคลาวด์ที่โครงสร้างพื้นฐานคลาวด์ถูกจัดหามาสำหรับการใช้แบบเปิดโดยสาธารณะทั่วไป และอาจถูกเป็นเจ้าของ จัดการ และจัดดำเนินการโดยองค์การธุรกิจ สถาบันการศึกษา และภาครัฐ หรือแบบผสม และติดตั้งอยู่ที่ฝั่งผู้ให้บริการคลาวด์สาธารณะ (Mell และ Grance, 2011) โดยตัวแบบนี้มีการจัดเตรียมโครงสร้างพื้นฐานคลาวด์สำหรับการใช้งานแบบเปิดโดยประชาชนโดยทั่วไปสามารถเข้าถึงแหล่งรวมโครงสร้างพื้นฐานคลาวด์ที่ใช้ร่วมกันและปรับแต่งรูปแบบการใช้งานตามความต้องการใช้ทรัพยากรได้ (NIST, 2015) เช่น ทรัพยากรเครือข่ายคอมพิวเตอร์, หน่วยการประมวลผลกลาง, หน่วยความจำ, หน่วยเก็บข้อมูล, ระบบปฏิบัติการ และโปรแกรมประยุกต์ต่างๆ โดยสามารถดำเนินการผ่านเครือข่ายตามคำขอได้อย่างสะดวก และทรัพยากรเหล่านี้สามารถจัดหาและคืนได้อย่างรวดเร็วโดยอาศัยการจัดการและการโต้ตอบกับผู้ให้บริการน้อยที่สุด (Mell และ Grance, 2011) องค์ประกอบสำคัญของการคำนวณแบบคลาวด์ได้แก่ บริการซอฟต์แวร์คลาวด์ และการคำนวณสาธารณูปโภคซึ่งหมายถึงการให้บริการโครงสร้างพื้นฐานคลาวด์หรือบริการแพลตฟอร์มคลาวด์ (Armbrust et al., 2009)

3) ตัวแบบคลาวด์ส่วนบุคคล (Private cloud) คือตัวแบบการติดตั้งใช้งานคลาวด์ที่โครงสร้างพื้นฐานคลาวด์ถูกจัดหามาสำหรับการใช้ประโยชน์ขององค์การใดองค์การหนึ่งที่ประกอบด้วยผู้บริโภคหลายคน และอาจถูกเป็นเจ้าของ จัดการ และจัดดำเนินการโดยองค์การ บุคคลที่สาม หรือแบบผสม และอาจติดตั้งอยู่ที่ฝั่งผู้ให้บริการคลาวด์สาธารณะหรือในองค์การของผู้ใช้บริการเอง (Mell และ Grance, 2011) โดยตัวแบบนี้มีการจัดเตรียมโครงสร้างพื้นฐานคลาวด์สำหรับการใช้งานโดยเฉพาะองค์กรเดียวที่ประกอบไปด้วยผู้บริโภคหลาย ซึ่งอาจจะเป็นเจ้าของบริหารจัดการเองและดำเนินการโดยผู้ให้บริการ ในการใช้บริการขององค์การอาจจะใช้งานบริการคลาวด์ส่วนบุคคลแบบนอกที่ตั้ง (off-premise private cloud) ส่วนองค์การที่ต้องการบริหารจัดการคลาวด์ส่วนบุคคลเองอาจะติดตั้งแบบในที่ตั้ง (on-premise private cloud) (NIST, 2015)

4) ตัวแบบคลาวด์ลูกผสม (Hybrid Cloud) คือตัวแบบการติดตั้งใช้งานคลาวด์ที่โครงสร้างพื้นฐานคลาวด์เป็นองค์ประกอบของสองระบบหรือมากกว่าของแพลตฟอร์มคลาวด์ที่แตกต่างกัน โดยโครงสร้างพื้นฐานคลาวด์อาจเป็นคลาวด์ส่วนบุคคลหรือคลาวด์สาธารณะ ที่มีตัวแบบไม่ซ้ำกันแต่มีความผูกพันเข้าด้วยกันโดยใช้ข้อกำหนดของเทคโนโลยีมาตรฐานหรือเทคโนโลยีเฉพาะที่ช่วยเปิดการทำงานของข้อมูลและซอฟต์แวร์ประยุกต์สมารถทำงานรวมกัน (Mell และ Grance, 2011) งานวิจัยนี้เน้นศึกษาตัวแบบนี้



ภาพประกอบที่ 1 ตัวแบบคลาวด์ลูกผสม (Hybrid Cloud) (Mell และ Grance, 2011)

2.แพลตฟอร์มบริการโครงสร้างพื้นฐานคลาวด์

สถาบันมาตรฐานและเทคโนโลยีแห่งชาติของสหรัฐอเมริกา (NIST, 2015) ได้กล่าวถึงการให้บริการโครงสร้างพื้นฐานคลาวด์เป็นการรวบรวมฮาร์ดแวร์และซอฟต์แวร์ที่คุณสมบัติที่จำเป็นในการสนับสนุนการให้บริการการคำนวณแบบคลาวด์ โดยสามารถแบ่งออกเป็นระดับชั้นกายภาพ (physical layer) และระดับชั้นนามธรรม (abstraction layer) โดยระดับชั้นกายภาพประกอบด้วยทรัพยากรฮาร์ดแวร์ที่จำเป็นในการสนับสนุนการให้บริการการคำนวณแบบคลาวด์เช่น เครื่องบริการ หน่วยเก็บข้อมูล และอุปกรณ์เครือข่าย ส่วนระดับชั้นนามธรรมประกอบด้วยซอฟต์แวร์ที่มีคุณสมบัติที่สามารถให้บริการในรูปแบบเครื่องเสมือนโดยผู้ใช้บริการไม่ต้องทราบถึงตำแหน่งที่ตั้งของทรัพยากรฮาร์ดแวร์ Armbrust, et al., (2009) โดยการให้บริการโครงสร้างพื้นฐานคลาวด์มีลักษณะเฉพาะที่สำคัญของการให้บริการการคำนวณแบบคลาวด์ประกอบด้วยห้าลักษณะเฉพาะดังนี้ (NIST, 2015)

1) การบริการตามคำขอด้วยตนเอง (on-demand self-service) ผู้ใช้บริการสามารถจัดหาทรัพยากรได้ตามความต้องการ และสามารถปรับเปลี่ยนความต้องการได้ด้วยตนเองผ่านช่องทางที่กำหนดโดยระบบจะทำงานอัตโนมัติตามคำขอของผู้ใช้บริการ ซึ่งไม่ต้องแจ้งผู้ให้บริการจัดการให้

2) การเข้าถึงเครือข่ายวงกว้าง (broad network access) ระบบสามารถทำงานได้บนเครือข่ายการสื่อสารที่มีมาตรฐานและสามารถเข้าถึงได้จากทุกอุปกรณ์ เช่น คอมพิวเตอร์ตั้งโต๊ะ คอมพิวเตอร์วางตัก โทรศัพท์เคลื่อนที่ และแท็บเล็ต

3) แหล่งรวมทรัพยากร (resource pooling) ผู้ให้บริการสามารถให้บริการลูกค้าได้พร้อมกันและสามารถให้บริการได้หลายรูปแบบ ซึ่งใช้ทรัพยากรร่วมกันทางกายภาพ โดยการให้บริการจะเป็นการสร้างทรัพยากรเสมือนสำหรับลูก โดยที่ลูกค้าจะไม่ทราบถึงตำแหน่งที่ตั้งของอุปกรณ์ที่ใช้งาน ซึ่งการใช้บริการของลูกค้าไม่จำเป็นต้องมีความรู้เกี่ยวกับระบบก็สามารถใช้บริการได้ทั้งในสถานที่ตั้งและนอกสถานที่ตั้ง

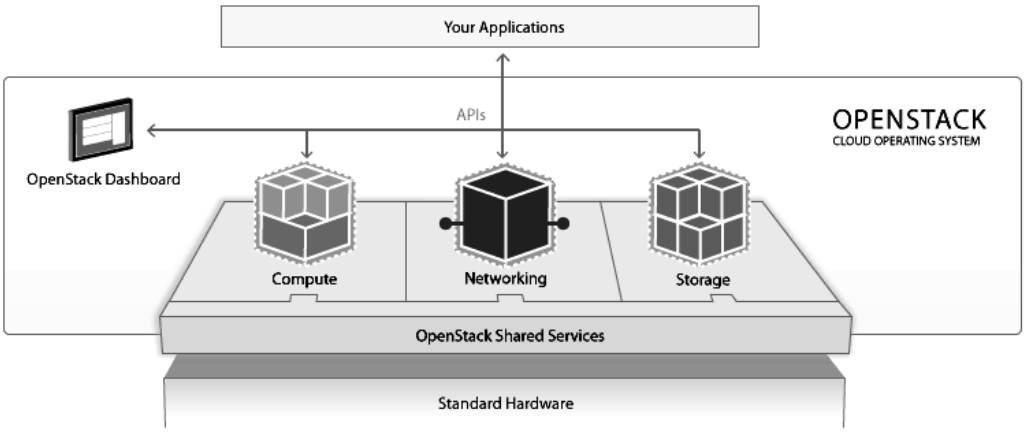
4) การให้บริการที่มีสภาพยืดหยุ่นและรวดเร็ว (rapid elasticity) ในการจัดเตรียมการใช้ทรัพยากรและการคืนทรัพยากรได้อย่างรวดเร็ว โดยไม่มีข้อจำกัดซึ่งลูกค้าสามารถปรับความต้องการใช้ทรัพยากรให้เหมาะสมกับปริมาณงาน ณ เวลาใดก็ได้

5) บริการที่วัดได้ (measured service) โดยระบบการคำนวณแบบคลาวด์เป็นระบบที่มีการควบคุมโดยอัตโนมัติ และใช้ทรัพยากรอย่างมีประสิทธิภาพ โดยใช้ประโยชน์จากความสามารถของเครื่องมือที่ใช้วัดการใช้งานทรัพยากร เช่น การจัดเก็บข้อมูล การประมวลผล แบนด์วิดธ์ และบัญชี

ผู้ใช้งานที่กำลังใช้งานระบบ โดยการใช้ระบบการคำนวณแบบคลาวด์ สามารถตรวจสอบ ควบคุม และรายงานการใช้งานได้อย่างโปร่งใส ทั้งผู้ให้บริการและผู้บริโภคที่ใช้บริการ

การให้บริการโครงสร้างพื้นฐานคลาวด์นอกจากต้องใช้ฮาร์ดแวร์แล้วยังต้องมีซอฟต์แวร์แพลตฟอร์มบริการโครงสร้างพื้นฐานคลาวด์ช่วยในการจัดสรรทรัพยากรโครงสร้างพื้นฐานคลาวด์ เช่นโอเพนสแตก (wiki.openstack.org, 2015: 1), อาปาเช่คลาวด์สแตก (cloudstack.apache.org, 2015: 1) และยูคาลิตัส (eucalyptus.com, 2015: 1) ซอฟต์แวร์แพลตฟอร์มบริการโครงสร้างพื้นฐานคลาวด์มีองค์ประกอบสำคัญคือผู้จัดการเครื่องเสมือน (Virtual Machine Manager : VMM) หรือไฮเพอร์ไวเซอร์ (Hypervisor) ซึ่งแบ่งได้เป็น 2 ประเภทคือแบร์เมทัล (bare metal) และ โฮสเตด (hosted) โดยแบบแบร์เมทัลมีการติดตั้งไฮเพอร์ไวเซอร์โดยตรงกับฮาร์ดแวร์ซึ่งทำให้ประสิทธิภาพในการประมวลผลทำได้ดีกว่าแบบโฮสเตดซึ่งจะต้องมีการติดตั้งระบบปฏิบัติการ (operating systems) ก่อนจึงติดตั้งไฮเพอร์ไวเซอร์ (Mell และ Grance, 2011) ในงานวิจัยนี้เลือกใช้แพลตฟอร์มโอเพนสแตก และอาปาเช่คลาวด์สแตก ซึ่งประกอบด้วยโฮสเตดไฮเพอร์ไวเซอร์ ในการทดสอบระบบนายหน้า

2.1 แพลตฟอร์มโอเพนสแตก คือซอฟต์แวร์โอเพ่นซอร์ส (open source) ที่ผู้ใช้สามารถพัฒนาและบริหารจัดการบริการโครงสร้างพื้นฐานคลาวด์ได้ พัฒนาขึ้นโดยมีวัตถุประสงค์เพื่อช่วยให้องค์กรสามารถใช้บริการของระบบการคำนวณแบบคลาวด์ที่สามารถติดตั้งบนฮาร์ดแวร์ที่อยู่ในมาตรฐานปัจจุบัน โดยแพลตฟอร์มโอเพนสแตกได้ทำการเปิดตัวโดยทำการติดตั้งแพลตฟอร์มบริการโครงสร้างพื้นฐานคลาวด์กับศูนย์ข้อมูล (data center) ที่เป็นที่ยอมรับเช่น แร็คสเปซ โฮสติ้ง (Rackspace Hosting) องค์การบริหารการบินและอวกาศแห่งชาติ สหรัฐอเมริกา (NASA) (OpenStack, 2015) บริษัทเอทีแอนด์ที (AT&T) องค์การยุโรปเพื่อการวิจัยนิวเคลียร์ (CERN) มหาวิทยาลัยเกษตรศาสตร์ และ องค์กรชั้นนำอื่นๆ โดยแพลตฟอร์มโอเพนสแตกได้เปิดตัวเมื่อกรกฎาคม ค.ศ. 2010 โดยใช้ชื่อว่าออสติน (Austin) และได้มีการพัฒนามาอย่างต่อเนื่อง โดยสถาปัตยกรรมแพลตฟอร์มโอเพนสแตกโดยการออกแบบสถาปัตยกรรมแบ่งออกเป็นสามตัวแบบ คือ คอมพิว (compute) เน็ตเวิร์คกิ้ง (networking) และสตอเร็จ (storage) ที่ทำงานรวมกันบนระบบเครือข่ายและมี แดชบอร์ดโอเพนสแตก (openstack dashboard) เป็นส่วนประสานกับผู้ใช้งานผ่านทางเอพีไอ (APIs) ของระบบโดยแสดงสถาปัตยกรรมของโอเพนสแตกดังภาพประกอบที่ 2



ภาพประกอบที่ 2 สถาปัตยกรรมของโอเพนสแตก (OpenStack, 2015)

แดชบอร์ด มีชื่อรหัส (code name) ว่าโนวา (Horizon) เป็นโครงการของโอเพนสแตกที่พัฒนาขึ้นเพื่อเพิ่มความสามารถในการให้บริการการเข้าถึง ของผู้จัดการระบบและผู้ใช้งานระบบผ่านเว็บ (web based) โดยผู้ใช้บริการสามารถจัดการได้ด้วยตนเอง เช่นการสร้างอินสแตนซ์ (instance) การกำหนดหมายเลขไอพี (IP Address) และ การกำหนดการควบคุมการเข้าถึง

คอมพิว มีชื่อรหัสว่าโนวา (Nova) เป็นโครงการของโอเพนสแตกที่พัฒนาขึ้นเพื่อเพิ่มความสามารถในการปรับขนาดการใช้ทรัพยากรโครงสร้างพื้นฐานคลาวด์ได้ตามความต้องการที่ร้องขอ (on-demand) ใช้งานด้วยตนเองผ่าน เอพีไอ ที่รองรับบริการของเรสท์ฟูล (RESTful HTTP) งานวิจัยนี้เน้นศึกษาเอพีไอนี้ (OpenStack, 2015)

เน็ตเวิร์คกิ้ง มีชื่อรหัสว่านิวตรอน (Neutron) เป็นโครงการของโอเพนสแตกที่พัฒนาขึ้นเพื่อเพิ่มความสามารถในการบริการเชื่อมต่อเครือข่ายของทุกบริการของโอเพนสแตก เช่น การสนับสนุนการทำงานของคอมพิวที่มีการกำหนดระบบเครือข่ายและเทคโนโลยีเครือข่ายที่ต้องการผ่านเอพีไอของโอเพนสแตก (OpenStack, 2015)

สตอเร็จ มีชื่อรหัสว่าสวิฟ (Swift) เป็นโครงการของโอเพนสแตกที่พัฒนาขึ้นเพื่อจัดการหน่วยเก็บข้อมูล โดยมีหน้าที่สำหรับการจัดเก็บข้อมูลและการสืบค้นข้อมูลจำนวนมากโดยปรับขนาดพื้นที่การใช้งานเก็บข้อมูลที่ดีที่สุดซึ่งเหมาะสมกับความต้องการใช้งาน การใช้งานสามารถทำได้โดยผ่านเอพีไอพื้นฐานของโอเพนสแตก (OpenStack, 2015)

แพลตฟอร์มโอเพนสแตกมีคุณลักษณะเด่นที่ครอบคลุมการบริการโครงสร้างพื้นฐานคลาวด์หลายประการซึ่งผู้วิจัยเลือกเฉพาะที่เกี่ยวกับงานวิจัยดังนี้

แพลตฟอร์มโอเพนสแตกในคอมพิวของชื่อรหัสโนวา มีการสนับสนุนวีเอ็มแวร์ (VMware) ที่รองรับโดยตัวขับคอมพิว (compute drivers) สามารถรองรับการปลุกเครื่อง (boot) จาก

ไอเอสโออิมเมท (ISO image) และ เคสอิมเมท (cached images) และนอกจากวีเอ็มแวร์แพลตฟอร์มโอเพนสแตกยังสนับสนุน Hyper-V, Libvirt (KVM) และ XenServer (OpenStack, 2015)

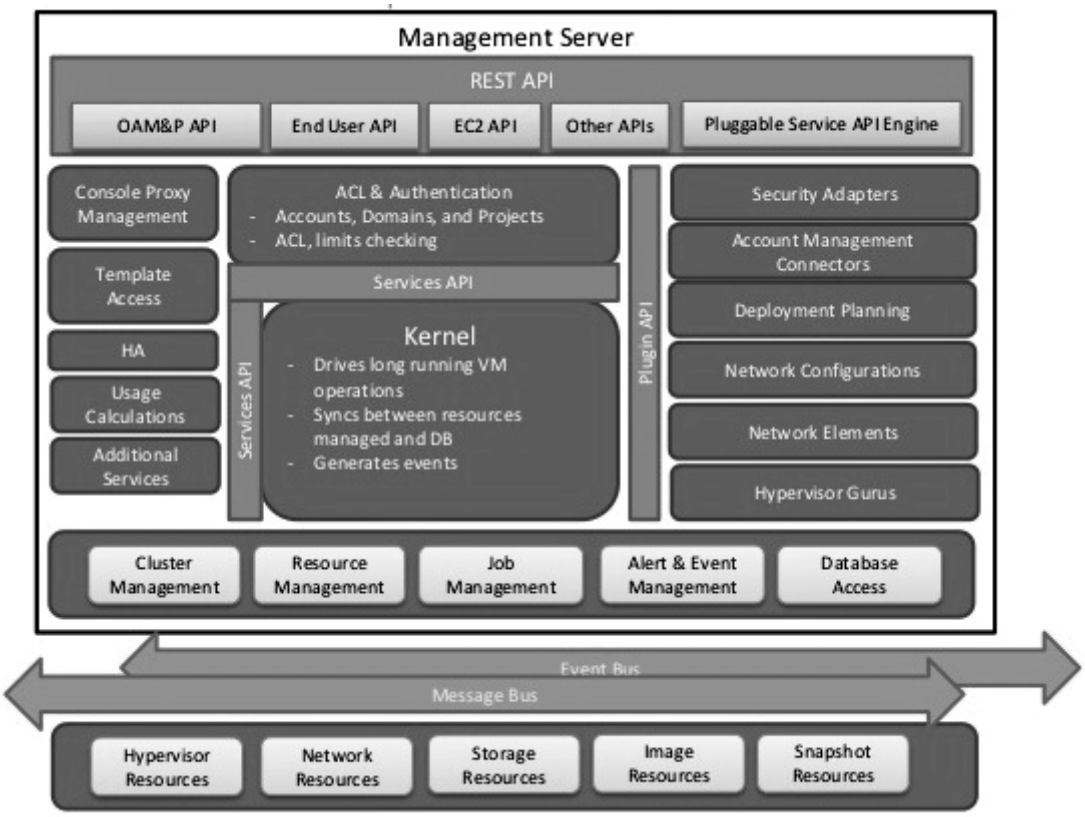
แพลตฟอร์มโอเพนสแตกมีการสนับสนุนบริการฐานข้อมูล (Database service) โดยสมารถบริหารจัดการผู้ใช้งานและเค้าร่างฐานข้อมูล (User/Schema management) ด้วยเอพีไอโทว์ (Trove API) ที่สนับสนุนการทำงานมายเอสคิวแอล (MYSQL) และสามารถเพิ่มลดขนาด (volume resizes) พื้นที่ในการจัดเก็บฐานข้อมูลโดยไม่มีผลกระทบ สามารถสำรองและกลับคืน (backups and restore) ฐานข้อมูล และรองรับระบบจัดการฐานข้อมูลได้เต็มรูปแบบเช่น เอสคิวแอล และ เพอร์โคนา (percona) และมีฐานข้อมูลที่รองรับแบบไม่เต็มรูปแบบเช่น มังโกดีบี (MongoDB) รีดิส (redis) แคสแซนดา (cassandra) และ คัสเบส (couchbase) (OpenStack, 2015)

แพลตฟอร์มโอเพนสแตกสามารถอนุญาตให้ผู้พัฒนาโปรแกรมสามารถ ตั้งค่าและปรับแต่งแผ่นแบบ (Template) ของโครงสร้างทรัพยากรพื้นฐานคลาวด์เช่น คอมพิว สตอเร็จ และเน็ตเวิร์คกิ้ง เพื่อรองรับการใช้งานการประยุกต์คลาว์ (cloud application) ของผู้ใช้ และรองรับคุณสมบัติการปรับเปลี่ยนขนาดโดยอัตโนมัติ (autoscaling) (OpenStack, 2015)

แพลตฟอร์มโอเพนสแตกสามารถรองรับการทำงานร่วมกับเอพีไอคลาวด์ (cloud APIs) อื่นๆ โดยมีการสนับสนุนอาเมซอลอีซีทู (Amazon EC2) ในรูปแบบอีซีทูคิวรีเอพีไอ (EC2 Query API) มีการสนับสนุนกูลเกิลคอมพิวเอนจิน (Google Compute Engine) ในรูปแบบจีซีไอเอพีไอ (GCE API) และรองรับอาเมซอลเอสสาม (Amazon S3) โดยผ่านมิลเดิลแวร์ของโครงการสวิฟสาม (Swift3)

2.2 อาปาเช่คลาวด์สแตก ซอฟต์แวร์โอเพ่นซอร์สที่ผู้ใช้สามารถพัฒนาและบริหารจัดการบริการโครงสร้างพื้นฐานคลาวด์ได้ พัฒนาขึ้นโดยมูลนิธิซอฟต์แวร์อาปาช่ (Apache Software Foundation: ASF) ซึ่งเป็นโครงการที่มีความสำคัญสุดของมูลนิธิ โดยอาปาเช่คลาวด์สแตกมีวัตถุเพื่อพัฒนาซอฟต์แวร์โอเพ่นซอร์สสำรับบริหารจัดการบริการโครงสร้างพื้นฐานคลาวด์ สำหรับภาครัฐและเอกชน โดยอาปาเช่คลาวด์สแตก โดยแพลตฟอร์มอาปาเช่คลาวด์สแตกได้ถูกพัฒนาครั้งแรกด้วยคลาวด์ดอทคอม (cloud.com) เมื่อพฤษภาคม ค.ศ. 2010 โดยใช้ชื่อว่าวีแอ็มอ๊อฟ (VMOps) และได้มีการพัฒนามาอย่างต่อเนื่อง แพลตฟอร์มโอเพนสแตกมีผู้ใช้บริการที่มีชื่อเสียงในวงการคอมพิวเตอร์เช่น แอปคารา (Appcara Inc) แอปเปิ้ล (Apple) เดล (Dell) ดิสนีย์ (Disney) หัวเว่ย (Huawei) และ อื่นๆ (Apache CloudStack, 2015) โดยแสดงสถาปัตยกรรมของอาปาเช่คลาวด์สแตก แบ่งเป็นสองตัวแบบคือ ตัวแบบเครื่องบริการสำหรับบริการจัดการทรัพยากรโครงสร้างพื้นฐานคลาวด์ (management server) และตัวแบบเครื่องบริการทรัพยากรโครงสร้างพื้นฐานคลาวด์

(resources server) ซึ่งมีการเชื่อมต่อผ่านอีเว่นบัส (event bus) และเมสเสจบัส (message bus) ดังภาพประกอบที่ 3



ภาพประกอบที่ 3 สถาปัตยกรรมของอาปาเช่คลาวด์สแตก (apache cloudstack, 2015)

ตัวแบบเครื่องบริการสำหรับบริการจัดการทรัพยากรโครงสร้างพื้นฐานคลาวด์ (management server) ประกอบด้วยส่วนต่อประสานโปรแกรมประยุกต์ที่สามารถเข้าถึงและจัดการตัวแบบต่างๆ ได้อย่างมีประสิทธิภาพ ประกอบด้วยเรทส์เอพีไอ (REST API) ที่มีการเชื่อมประสานกับผู้ใช้งาน อุปกรณ์ หรือ บริการอื่นที่เกิดจากภายนอกผ่านเรทส์เทคโนโลยี ส่วนที่สองเซอร์วิสเอพีไอ (service API) ที่มีการเชื่อมประสานกับส่วนบริการการควบคุมและจัดการคลาวด์สแตก (OpenStack Orchestration) มีหน้าที่บริการมนุษย์หรือเครื่องจักรในการวงจรชีวิตทั้งหมดของจัดการทรัพยากรโครงสร้างพื้นฐานคลาวด์และการประยุกต์ใช้งานต่างๆ ภายในคลาวด์สแตก ส่วนที่สาม ปลั๊กอินเอพีไอ (plugin API) สำหรับเสริมการทำงานของคลาวด์สแตกเช่น การกำหนดค่าของระบบเครือข่าย (network configurations) การวางแผนการปรับใช้ทรัพยากรโครงสร้างพื้นฐานคลาวด์ (deployment planning)

แพลตฟอร์มอาปาเช่คลาวด์สแตก มีคุณลักษณะเด่นที่ครอบคลุมการบริการโครงสร้างพื้นฐานคลาวด์หลายประการดังนี้

แพลตฟอร์มอาปาเช่คลาวด์สแตกมีส่วนต่อประสานโปรแกรมประยุกต์สมบูรณ์แบบโดยเรียกว่า เอพีไอเพาเวอร์ฟูล (Powerful API) ที่มีการบริหารจัดการในการสร้างการรวม และการใช้งานโปรแกรมประยุกต์ได้ผ่านเอพีไอคลาวด์สแตก (CloudStack API) ซึ่งทำให้สามารถเข้าถึงและปรับแต่งตัวแบบต่างๆ อาปาเช่คลาวด์สแตกได้อย่างมีประสิทธิภาพ โดยสามารถสั่งงานจากภายนอกผ่านเรทส์เอพีไอ หรือทำการติดตั้งปลั๊กอินเอพีไอของคลาวด์สแตกเพื่อช่วยเพิ่มประสิทธิภาพในการทำงานต่างๆ ของคลาวด์สแตกเช่น การเชื่อมต่อไปยังระบบการคำนวณแบบคลาวด์ภายนอกเช่น อาเมซอลอีซีทู (Amazon EC2) และเอสทรีเอพไอ (S3 APIs.)

แพลตฟอร์มอาปาเช่คลาวด์สแตกมีรูปแบบของส่วนต่อประสานผู้ใช้งานที่สามารถบริหารจัดการได้ทุกส่วนโปรแกรมโดยเรียกว่าริชเมเนจเมนท์ยูสเซอร์อินเตอร์เฟส (Rich Management User Interface) โดยมีการพัฒนาตามแนวคิดนอกกรอบ (out-of-the-box) ที่มีการสร้างส่วนต่อประสานกับผู้ใช้งานในระดับบนเอพีไอคลาวด์สแตก (CloudStack API) ที่สามารถบริหารจัดการโครงสร้างพื้นฐานคลาวด์ได้อย่างมีประสิทธิภาพ โดยการพัฒนาใช้เทคโนโลยีเอแจ็กซ์ (AJAX) ที่รองรับการทำงานผ่านเว็บเบราเซอร์ (web browser) และยังสามารถรวมกับเว็บพอร์ทอลภายนอกได้ (existing portal)

แพลตฟอร์มอาปาเช่คลาวด์สแตกสามารถทำงานทำงานร่วมกับไฮเปอร์ไวเซอร์รายใหญ่ในตลาด เช่น วีเอ็มแวร์ (VMware) เควีเอ็ม (KVM) ซิทริกซ์ซรีเซิร์ฟเวอร์ (Citrix XenServer) และ ซิทริกซ์ซรีคลาว์แพลตฟอร์ม (Citrix Xen Cloud Platform)

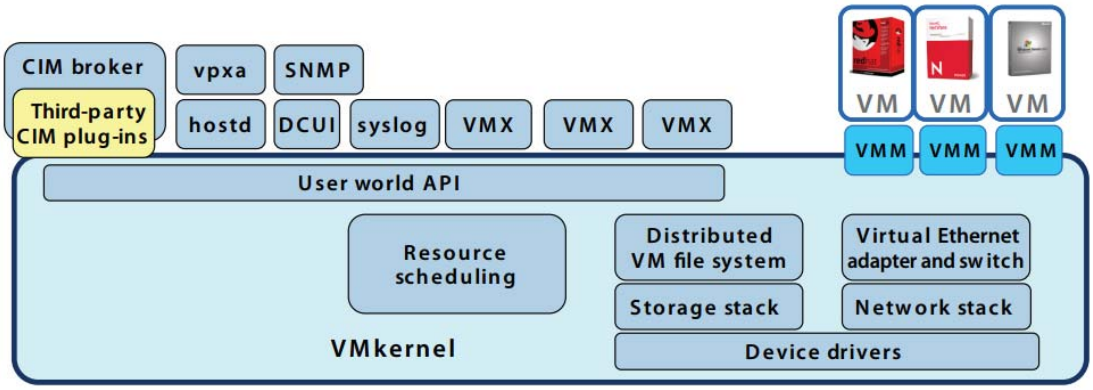
แพลตฟอร์มอาปาเช่คลาวด์สแตกสามารถรองรับการทำงานของระบบปฏิบัติการได้หลายรูปแบบขึ้นอยู่กับการกำหนดไฮเปอร์ไวเซอร์ โดยระบบปฏิบัติการที่รองรับประกอบด้วยวินโดว์ (Windows) ลีนุกส์ (Linux) และ บีเอสดี (BSD) แพลตฟอร์มอาปาเช่คลาวด์สแตกเพิ่มความปลอดภัยของการให้บริการในการสร้างเครื่องเสมือนที่รองรับระบบปฏิบัติการต่างๆ ด้วยการสุ่มรหัสผ่านสำหรับระบบปฏิบัติการทั่วไป (Random Password for Guest OS) โดยจะสร้างรหัสผ่านที่ไม่ซ้ำกันของเครื่องเสมือนโดยผู้ใช้งานมีความปลอดภัยในการเข้าถึงและการเปลี่ยนรหัสผ่าน

แพลตฟอร์มอาปาเช่คลาวด์รองรับการทำงานการบริหารจัดการภาระงานแบบพลวัต (Dynamic Workload Management) มีการรองรับภาระงานกระจายงานให้ทรัพยากรโครงสร้างพื้นฐานในระดับกายภาพเช่น คอมพิว เครือข่าย และ หน่วยเก็บข้อมูล โดยอัตโนมัติในการกระจายภาระงานจะพิจารณาจาก นโยบายของโหลดบาลานซ์ (load balancing) ความปลอดภัยของข้อมูล (data security) และ ข้อกำหนดที่ต้องปฏิบัติตาม (compliance)

แพลตฟอร์มอาปาเช่คลาวด์มีบริการเครือข่ายเสมือน (virtual network) เช่น เราเตอร์เสมือน (virtual routers), ไฟวอลเสมือน (virtual firewalls) และโหลดบาลานซ์เสมือน (virtual load balancers) โดยแพลตฟอร์มอาปาเช่คลาวด์สามารถบริการและกำหนดค่าการใช้งานของเครือข่าย

เสมือนที่ช่วยให้เกิดความปลอดภัย และยืดหยุ่นต่อการใช้งานในสภาพแวดล้อมของคำนวณแบบคลาวด์ โดยมีการรวมเทคโนโลยีของ เอฟไฟว์เน็ตเวิร์ค (F5 Network) และ เน็ตสเกลล่า (Netscaler)

2.3 วีเอ็มแวร์ อีเอสเอ็กซ์ไอ ถูกพัฒนาโดยมีวัตถุเพื่อพัฒนาซอฟต์แวร์สำรับบริหารจัดการบริการโครงสร้างพื้นฐานคลาวด์ ในระดับองค์กร โดยบริษัท วีเอ็มแวร์ จำกัด ซึ่งวีเอ็มแวร์ อีเอสเอ็กซ์ไอเป็นนวตกรรมใหม่ที่เหมาะกับระบบปฏิบัติการทั่วไป ที่มีความปลอดภัยและหน้าเชื่อถือ สามารถจัดการได้ง่าย และสถาปัตยกรรมที่มีขนาดกะทัดรัดโดยมีสถาปัตยกรรมตามภาพประกอบที่ 4



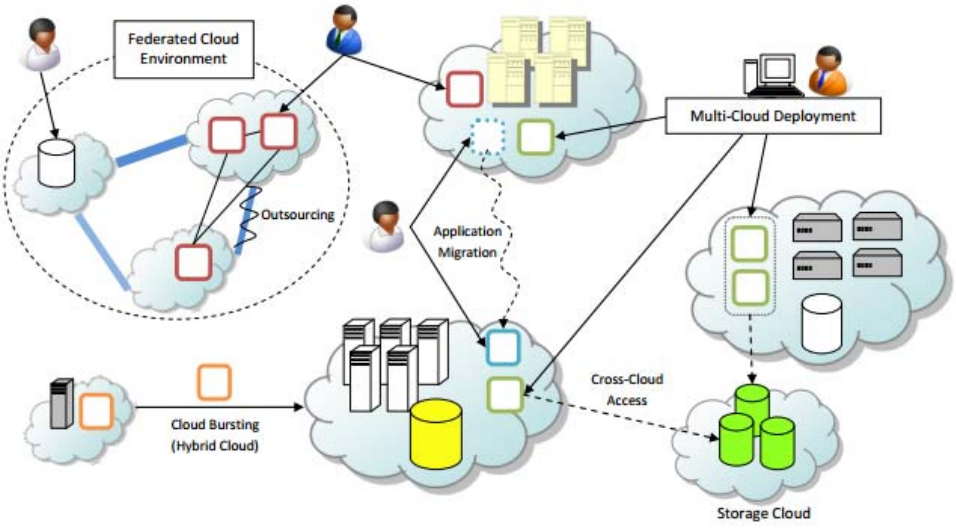
ภาพประกอบที่ 4 สถาปัตยกรรมของวีเอ็มแวร์ อีเอสเอ็กซ์ไอ (VMware, Inc. 2016)

จากภาพประกอบที่ 4 สถาปัตยกรรมของวีเอ็มแวร์ อีเอสเอ็กซ์ไอประกอบด้วยระบบปฏิบัติการที่รองรับเรียกว่า วีเอ็มเคอร์เนลที่ทำหน้าที่ควบคุมเครื่องเสมือน ฮาร์ดแวร์ และทรัพยากรสำหรับโปรแกรมประยุกต์โดย โดยวีเอ็มเคอร์เนลมีกระบวนการหลักประกอบด้วย 1) ส่วนต่อประสานกับผู้ใช้สำหรับการเฝ้าคุมโดยตรง (Direct Console User Interface: DCUI) ทำหน้าที่บริหารจัดการและกำหนดโครงแบบระดับต่ำผ่านทางส่วนเฝ้าคุมขอลเครื่องบริการ 2) ส่วนเฝ้าสังเกตเครื่องเสมือน (virtual machine monitor) ทำหน้าที่เป็นการตรวจสอบการทำงานของเครื่องเสมือนประกอบด้วย วีเอ็มเอกซ์ (VMX) และ วีเอ็มเอ็ม (VMM) 3) ตัวกระทำต่างๆ (Various agents) สำหรับการควบคุมวีเอ็มแวร์จากระยะไกลด้วยโปรแกรมประยุกต์ และ ตัวแบบสารสนเทศทั่วไป (Common Information Model: CIM) เป็นระบบที่ช่วยควบคุมผ่านส่วนต่อประสานโปรแกรมประยุกต์มาตรฐานจากระยะไกล

3.การทำงานร่วมกันได้ของการคำนวณแบบคลาวด์ (cloud interoperability)

Toosi, et. al., (2014) กล่าวว่าปัจจุบันมีการใช้ระบบอินเตอร์เน็ตสำหรับการสื่อสารและการให้บริการการคำนวณแบบคลาวด์อย่างแพร่หลาย และมีความต้องการใช้งานร่วมกันของการคำนวณแบบคลาวด์ เช่น มีการใช้บริการหลายการคำนวณแบบคลาวด์ (multi-cloud deployment) โดยมีด้วยเหตุของประสิทธิภาพการประมวลผลข้อมูลที่ไม่เท่ากันของแต่ละแพลตฟอร์ม หรือบ้าง

แพลตฟอร์มมีการใช้บริการเก็บข้อมูลบนการคำนวณแบบคลาวด์ในรูปแบบหน่วยเก็บข้อมูล (storage cloud) เดียวกันในรูปแบบการเข้าถึงคลาวด์แบบข้ามระบบคลาวด์ (cross-cloud access) ผู้ใช้บริการการคำนวณแบบคลาว์อาจมีความต้องการย้ายโปรแกรมประยุกต์ (application migration) จากผู้ให้บริการรายหนึ่งไปผ้ให้บริการรายอื่น ผู้ใช้บริการมีความต้องการใช้การคำนวณแบบคลาวด์ที่มีสภาพแวดล้อมรวมเป็นสหพันธ์ (federated cloud environment) ที่มีการใช้การคำนวณแบบคลาวด์ร่วมกันเป็นกลุ่มบนสภาพแวดล้อมเดียวกัน หรือการใช้งานร่วมกันของคลาวด์ลูกผสมที่มีการใช้บริการการคำนวณแบบคลาวด์แบบแยกแผ่น (cloud bursting) ที่สามารถใช้งานทรัพยากรโครงสร้างพื้นฐานคลาวด์ ได้เกินขีดจำกัดของระบบการคำนวณแบบคลาวด์หลัก โดยการขยายการให้บริการโครงสร้างพื้นฐานคลาวด์ของผู้ให้บริการรายอื่น โดยแสดงสถานการณ์ต่างๆ ตามภาพประกอบที่ 5

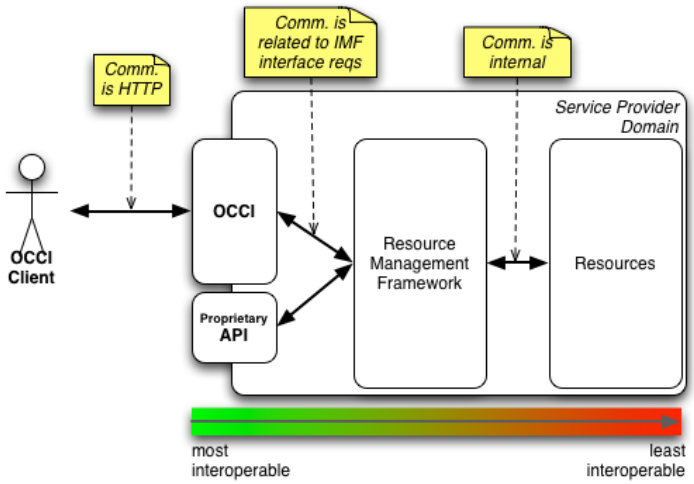


ภาพประกอบที่ 5 แสดงสถานการณ์ของการทำงานร่วมกันของระบบการคำนวณแบบคลาวด์และอินเตอร์คลาวด์ (Inter-Cloud) (Toosi, et. al., 2014)

Open Data Center Alliance (2012) ได้กล่าวถึงการทำงานร่วมกันของระบบการคำนวณแบบคลาวด์ในภาพรวมว่ามีองค์ประกอบที่สำคัญสองประการคือ การเคลื่อนย้ายง่ายของการบริการ (portability) โดยสามารถเข้าใช้งาน การย้าย การยกเลิกการใช้บริการที่สามารถเปลี่ยนแปลงได้ตลอดเวลาซึ่งสมารถกำหนดค่าได้โดยง่ายหรือโดยอัตโนมัติตามภาระงาน ประการสองการเชื่อมต่อระหว่างระบบ (interconnectability) ที่มีทรัพยากรที่ให้อยู่หลายที่มีความซับซ้อนในการเชื่อมต่อถึงกัน ซึ่งเป็นเหตุผลสำคัญในการทำงานร่วมกันของระบบการคำนวณแบบคลาวด์ เมื่อพิจารณาในระบบธุรกิจพบว่าการทำงานร่วมกันของระบบการคำนวณแบบคลาวด์เป็นกุญแจสำคัญสำหรับการบริหารจัดการระบบการคำนวณแบบคลาวด์ที่ยืดหยุ่นตามความต้องการของตลาดที่มีความผันผวน

ของลูกค้า อย่างไรก็ตามการทำงานร่วมกันของการคำนวณแบบคลาวด์ในระบบอินเทอร์เน็ตจำเป็นต้องมีมาตรฐานการทำงานร่วมกันของระบบการคำนวณแบบคลาวด์โดย Loutas, et. al., (2011) ได้กล่าวถึงการพัฒนามาตรฐานการทำงานร่วมกันของระบบการคำนวณแบบคลาวด์เป็นการร่วมมือของ นักวิชาการ หน่วยงานรัฐบาล และ ผู้ใช้บริการคลาวด์ เพื่อออกมาตรฐานสำหรับเป็นข้อกำหนดในการออกแบบคำนวณแบบคลาวด์ มีการดำเนินการพัฒนาแนวทางการทำงานร่วมกันของการคำนวณแบบคลาวด์ 3 แนวทางดังต่อไปนี้

1) เอพีไอมาตรฐาน (standardized API) และ ส่วนต่อประสานคลาวด์ (cloud interface) สำหรับการทำงานร่วมกันของการคำนวณแบบคลาวด์เช่น คลาวด์ดาต้าแมเนจเม้นท์อินเตอร์เฟส (Cloud Data Management Interface) หรือ ซีดีเอ็มไอ (CDMI) เป็นการกำหนดมาตรฐานสำหรับนักพัฒนาโปรแกรมใช้ในการดำเนินการและการใช้งานหน่วยเก็บข้อมูลคลาวด์ (cloud storage) และการจัดการข้อมูลที่เก็บไว้ โดยการกำหนดส่วนต่อประสานสำหรับทำการ สร้างข้อมูล ปรับปรุงข้อมูล แก้ไขข้อมูล และลบข้อมูล ที่เป็นส่วนประกอบของการคำนวณแบบคลาวด์ และกำหนดส่วนต่อประสานสำหรับการบริการจัดการ บัญชีผู้ใช้ ความปลอดภัยในการเข้าถึง และการตรวจสอบการเรียกเก็บเงินจากการใช้บริการข้อมูล (SNIA, 2016) และโอเพ่นคลาวด์คอมพิวติ้งอินเตอร์เฟส (Open Cloud Computing Interface) หรือ โอซีซีไอ (OCCI) เป็นการกำหนดมาตรฐานการจัดการบริการโครงสร้างพื้นฐานการคำนวณแบบคลาวด์ในระยะไกลโดยใช้เอพีไอ เพื่อให้สามารถพัฒนาเครื่องมือสำหรับการทำงานร่วมกันของการคำนวณแบบคลาวด์ รวมถึงการใช้งานแบบปรับอัตโนมัติตามความต้องการ และการตรวจสอบการทำงานร่วมกัน (Open Grid Forum, 2016) โดยมีสถาปัตยกรรมของโอซีซีไอดังภาพประกอบที่ 6



ภาพประกอบที่ 6 สถาปัตยกรรมของโอซีซีไอที่ปรับใช้กับการให้บริการโครงสร้างพื้นฐานคลาวด์ (Open Grid Forum, 2016)

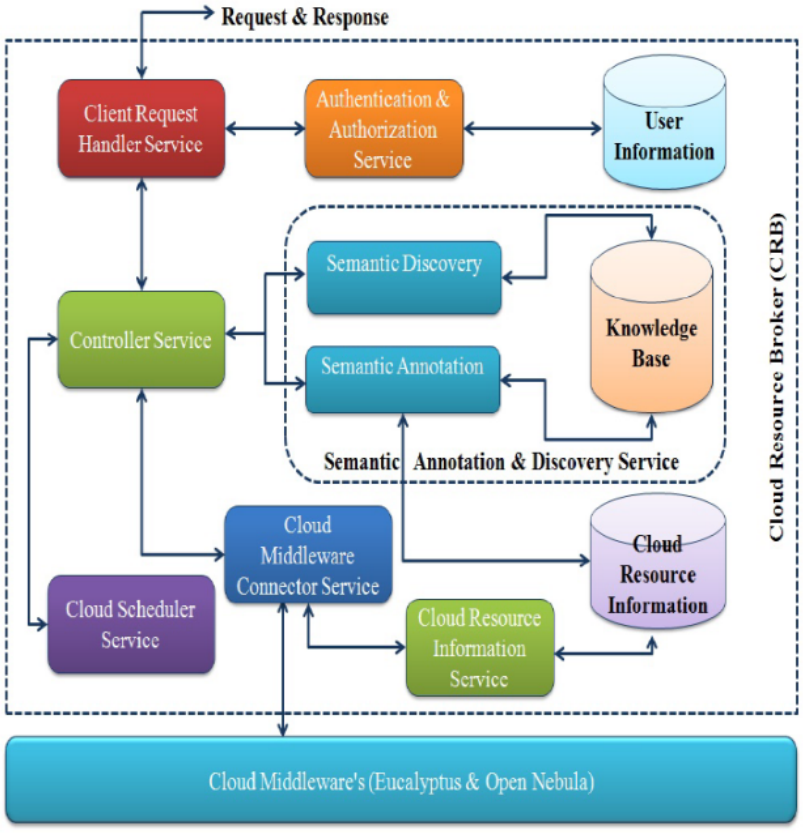
จากภาพประกอบที่ 6 แสดงขอบเขตของโปรโตคอลสำหรับผู้ให้บริการโครงสร้างพื้นฐานคลาวด์ เพื่อการบริหารจัดการภายในของบริการโครงสร้างพื้นฐานคลาวด์ โดยโอซีซีไอกำหนดวิธีการสื่อสารระหว่าง โอซีซีไอไคล์แอนด์กับโอซีซีไอ ผ่านโปรโตคอลเอชทีทีพี โดยใช้เรสท์ฟูลเอพีไอ โดยกำหนดรูปแบบชุดคำสั่งของเรสท์ฟูลเอพีไอ ในการจัดการโครงสร้างพื้นฐานคลาวด์ โดยการติดตั้งโครงแบบ การบริหารจัดการ และการคืนทรัพยากร โดยส่งชุดคำสั่งให้กับส่วนกรอบบริหารจัดการทรัพยากร (resource management framework) ของการคำนวณแบบคลาวด์ เพื่อส่งชุดคำสั่งสำหรับการจัดการทรัพยากรภายใน

2) ตัวแบบข้อมูลร่วมกันมาตรฐาน (standardized common data model) ของการคำนวณแบบคลาวด์ เช่น คลาวด์แมเนจเม้นท์เวิร์คกรุ๊ป (Cloud Management Working Group :CMWG) ที่กำหนดมาตรฐานดำเนินงานและคุณลักษณะของวงจรชีวิตของบริการคลาวด์ผ่านการทำงานบนอินเตอร์เฟซการจัดการโครงสร้างพื้นฐานคลาวด์ (CIMI) โดยกำหนดโปรโตคอลสำหรับจัดการระหว่างสองบริการโครงสร้างพื้นฐานคลาวด์ และการบริการลูกค้าในการใช้งานทรัพยากรโครงสร้างพื้นฐานคลาวด์ (Distributed Management Task Force, 2016), โอเพ่นวิชวลไลเซชันฟอร์แมต (Open Virtualization Format : OVF) เป็นมาตรฐานให้อุตสาหกรรมในการกำหนดรูปแบบมาตรฐานของบรรจุซอฟแวร์บนพื้นฐานของระบบเสมือนเพื่อแก้ปัญหาความต้องการใช้งานบริการการคำนวณแบบคลาวด์ของผู้ผลิตซอฟต์แวร์และผู้ให้บริการการคำนวณแบบคลาวด์ ซึ่งเวโอเอฟสามารถนำไปใช้รวมกับมาตรฐานไอเอสโอ 17203 (ISO 17203) (Distributed Management Task Force, 2016) และการจัดการโปรแกรมประยุกต์สำหรับแพลทฟอร์ม การคำนวณแบบคลาวด์ (Cloud Application Management for Platforms: CAMP) ที่กำหนดมาตรฐานใช้ในการจัดการใช้โปรแกรมสำเร็จในการพัฒนาระบบการคำนวณแบบคลาวด์และการปรับใช้โปรแกรมประยุกต์สำหรับการคำนวณแบบคลาวด์ซึ่งสามารถชื่อประสาน การจัดเตรียม การตรวจสอบ และ การควบคุม ของระบบการคำนวณแบบคลาวด์ บนพื้นฐานของเรทส์ที่มีการใช้เครื่องมือ คลังโปรแกรม (libraries) ปลั๊กอิน (plugins) และ กรอบการทำงาน(frameworks) ร่วมกัน (Organization for the Advancement of Structured Information Standards, 2016)

3) การใช้ประโยชน์จากตลาดหรือนายหน้า (utilization of a marketplace/broker) ในมาตรฐานการใช้ประโยชน์จากตลาดหรือนายหน้า สถาบันมาตรฐานและเทคโนโลยีแห่งชาติของสหรัฐอเมริกาได้กล่าวไว้ในแผนการพัฒนาเกี่ยวกับมาตรฐานการคำนวณแบบคลาวด์ของสถาบันมาตรฐานและเทคโนโลยีแห่งชาติของสหรัฐอเมริกา (NIST Cloud Computing Standards Roadmap) ได้กล่าวถึงระบบนายหน้าคลาวด์เป็นตัวกลางระหว่างผู้บริโภคและผู้ให้บริการ ซึ่งจะช่วยให้ผู้บริโภคสามารถใช้งานบริการการคำนวณแบบคลาวด์ที่ขั้นตอนทรัพซ้อนได้โดยง่ายและยังเป็นแนวทางในการเพิ่ม

มูลค่าสำหรับบริการการคำนวณแบบคลาวด์ โดยผู้บริโภคคลาวด์สามารถร้องขอบริการคลาวด์จากผู้ให้บริการระบบคลาวด์โดยตรงหรือผ่านนายหน้าคลาวด์ ซึ่งนายหน้าคลาวด์อาจเป็นนิติบุคคลที่จัดการการใช้งาน, การทำงาน, และการส่งมอบบริการคลาวด์และเจรจาความสัมพันธ์ระหว่างผู้ให้บริการคลาวด์และระบบคลาวด์ของผู้บริโภค นายหน้าคลาวด์มีความจำเป็นเพราะการคำนวณแบบคลาวด์เป็นการรวมกลุ่มของบริการคลาวด์อาจกลายเป็นความซับซ้อนมากสำหรับผู้บริโภคในการจัดการระบบคลาวด์ ซึ่งทำให้ผู้บริโภคสามารถร้องขอบริการจากนายหน้าคลาวด์ได้โดยไม่จำเป็นต้องติดต่อไปที่ผู้ให้บริการคลาวด์โดยตรง นายหน้าคลาวด์จะเป็นการให้บริการแบบจุดเดียวที่สามารถให้บริการได้หลายแพลตฟอร์มการบริการคลาวด์จากหลายผู้ให้บริการ ข้อกำหนดความแตกต่างระหว่างนายหน้าคลาวด์กับผู้ให้บริการคลาวด์ โดยพิจารณาจากส่วนต่อประสานเพื่อวัตถุประสงค์ทางธุรกิจหรือทางเทคนิคโดยทั่วไปนายหน้าคลาวด์มีการให้บริการในสามประเภทได้แก่ 1)การเป็นสื่อกลาง (intermediation) นายหน้าคลาวด์ช่วยเพิ่มการบริการที่ได้รับจากการปรับปรุงความสามารถการให้บริการที่มีมูลค่าเพิ่มให้กับผู้บริโภคคลาวด์เช่น การปรับปรุงที่สามารถจัดการการเข้าถึงบริการคลาวด์ การจัดการเอกลักษณ์ การรายงานผลการปฏิบัติงาน การรักษาความปลอดภัยที่เพิ่มขึ้น และอื่นๆ 2)ภาพรวมกลุ่ม (aggregation) นายหน้าคลาวด์เป็นการรวมและบูรณาการ หลายการบริการเพื่อให้เกิดบริการใหม่ นายหน้าให้ข้อมูลและการบริการสำหรับลูกค้าที่ใช้บริการคลาวด์เพื่อทราบถึงความเคลื่อนไหวระหว่าง ลูกค้าที่ใช้บริการคลาวด์กับหลายผู้ให้บริการคลาวด์ เพื่อสร้างความเชื่อมั่นและความปลอดภัยในการใช้บริการ และ 3)การอนุญาโตตุลาการ(arbitrage) นายหน้าคลาวด์สามารถให้ผู้ใช้บริการสามารถเลือกบริการจากหลายผู้ให้บริการ ที่มีบริการที่คล้ายคลึงกันโดยคำนึงถึงปัจจัยต่างๆ เช่น งบการเงินที่ไม่คงที่ของผู้ใช้บริการสมารถมีความยืดหยุ่นในการเลือกผู้ให้บริการที่เหมาะสมจากหลายผู้ให้บริการ (NIST, 2013) และมีการกล่าวถึงมาตรฐานการใช้ประโยชน์จากตลาดหรือนายหน้าเช่น คลาวด์อินดัสทรีฟอรั่ม (Cloud Industry Forum) หรือ ซีไอเอฟ (CIF) ที่มีการกล่าวถึงมาตรฐานขององค์การธุรกิจหรืออุตสาหกรรมที่ต้องการใช้คลาวด์ลูกผสมในการดำเนินธุรกิจโดยจะมีการประเมินองค์กรธุรกิจและแนะนำในการแนวทางการแก้ไขปัญหาตามข้อกำหนดของมาตรฐาน โดยเมื่อผ่านมาตรฐานแล้วจะได้รับการรับรองที่น่าเชื่อถือ และมีความสามารถในการให้บริการที่มีความปลอดภัยและมีประสิทธิภาพโดยซีไอเอฟให้กำหนดรหัสสำหรับการปฏิบัติ (code of practice) ของผู้ให้บริการการคำนวณแบบคลาวด์ ประกอบด้วยรหัสเอความโปร่งใส (A. Transparency) ในการให้บริการซึ่งมีการกำหนดเรื่อง การเปิดเผยความจริงของสารสนเทศต่อสาธารณะและเรื่องการทำสัญญาในการเปิดเผยความจริงของสารสนเทศต่อสาธารณะในข้อกำหนดที่เป็นความลับของผู้ใช้บริการ รหัสบีความสามารถ (B. Capability) ของผู้ให้บริการการคำนวณแบบคลาวด์ ในการแสดงให้เห็นถึงการจัดการสารสนเทศที่มีประสิทธิภาพและโปร่งใส่ และ รหัสซีความรับผิดชอบ (C. Accountability)

โดยผู้ให้บริการการคำนวณแบบคลาวด์ จะยืนยันว่าพวกเขาจะปฏิบัติตามและรับผิดชอบในการปฏิบัติตามหลักจรรยาบรรณและพฤติกรรมของพวกเขากับลูกค้า โดยการได้รับใบรับรองของซีไอเอฟจำเป็นต้องลงทะเบียนและปฏิบัติตามรหัสสำหรับการปฏิบัติและผ่านการตรวจสอบจากคณะกรรมการของซีไอเอฟ (cloud industry forum, 2016) ที่กล่าวมาข้างต้นเป็นการกำหนดมาตรฐานการใช้ประโยชน์จากตลาดหรือนายหน้าที่เป็นข้อกำหนดสำหรับผู้ให้บริการ แต่มีความพยายามจัดทำข้อกำหนดสำหรับซอฟต์แวร์ที่ให้บริการในลักษณะนายหน้าเช่น ตัวแบบนายหน้าสำหรับการจัดการทรัพยากรการคำนวณแบบคลาวด์ (cloud resource broker) หรือ ซีอาบี (CRB) ที่มีการกำหนดกรอบในการจัดทำซอฟต์แวร์สำหรับการบริหารจัดการทรัพยากรโครงสร้างพื้นฐานคลาวด์ โดยมีสถาปัตยกรรมตัวแบบซีอาบีตามภาพประกอบที่ 7



ภาพประกอบที่ 7 สถาปัตยกรรมของตัวแบบซีอาบี (Somasundaram, et. al., 2012)

สถาปัตยกรรมของตัวแบบซีอาบีเป็นการกำหนดกรอบแนวคิดในการพัฒนาซอฟต์แวร์สำหรับจัดการทรัพยากรโครงสร้างพื้นฐานคลาวด์โดยมีการเชื่อมประสานระหว่าง บริการโครงสร้างพื้นฐานคลาวด์กับชุดคำสั่งที่ร้องขอหรือตอบสนองจากภายนอกผ่านตัวแบบบริการจัดการการร้องขอจากไคล์แอนด์ (client request handler service) ซึ่งจะนำชุดข้อมูลมาตัวแบบบริการการพิสูจน์ตัวจริงด้วยตัวแบบและอนุญาต (authentication & authorization service) เพื่อความปลอดภัยในการระบุตัวตนจากสารสนเทศผู้ใช้งาน (user information) ที่เก็บไว้ เมื่อชุดคำสั่ง

พิสูจน์ตัวตนได้ จะถูกส่งไปยังตัวแบบบริการโปรแกรมควบคุม (controller service) แล้วทำการเปรียบเทียบกับ ฐานความรู้ (knowledge base) โดยการเปรียบเทียบเชิงความหมาย ตามตัวแบบบริการเชิงความหมายแบบความคิดเห็นประกอบและการค้นพบ (semantic annotation & discovery service) ในบางกรณีที่ต้องตรวจสอบทรัพยากรโครงสร้างพื้นฐานคลาวด์จะตรวจสอบจากสารสนเทศทรัพยากรโครงสร้างพื้นฐานคลาวด์ (cloud resource information) ที่ถูกจัดเก็บไว้ กรณีที่มีการจัดกำหนดการของความต้องการใช้งานโครงสร้างพื้นฐานคลาวด์ จะกำหนดจากตัวแบบบริการจัดกำหนดการคลาวด์ (cloud scheduler service) เพื่อส่งชุดคำสั่งที่ต้องการใช้งานตามกำหนดการไปยังตัวแบบบริการโปรแกรมควบคุมเหมือนกัน เมื่อตัวแบบบริการโปรแกรมควบคุมประมวลผลเสร็จแล้วจะส่งชุดคำสั่งไปยังตัวแบบบริการตัวตัวเชื่อมต่อมิดเดิลแวร์คลาวด์ (cloud middleware connector service) ซึ่งจะทำการส่งชุดคำสั่งไปยังแพลตฟอร์มบริการโครงสร้างพื้นฐานคลาวด์ ในกรณีที่มีการร้องขอใช้ทรัพยากร และ การคืนทรัพยากรข้อมูลชุดคำสั่งจะถูกส่งไปยังตัวแบบบริการสารสนเทศทรัพยากรคลาวด์ (cloud resource information service) เพื่อจัดเก็บที่สารสนเทศทรัพยากรโครงสร้างพื้นฐานคลาวด์ต่อไป (Somasundaram, et. al., 2012)

ซึ่งงานวิจัยนี้เน้นที่การใช้ตัวแบบนายหน้าสำหรับการจัดการทรัพยากรการคำนวณแบบคลาวด์หลายแพลตฟอร์ม

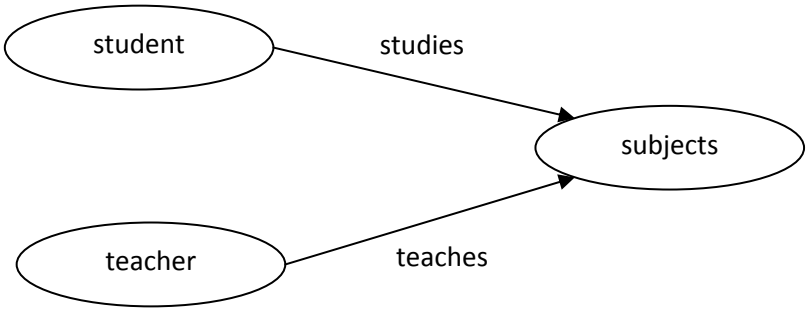
4.วิศวกรรมออนโทโลจี

Staab และ Studer (2009) กล่าวว่าออนโทโลจีเป็นองค์ความรู้ที่ถูกกล่าวถึงอย่างมากในการพัฒนาปัญญาประดิษฐ์ (AI) ที่ช่วยทำให้ระบบคอมพิวเตอร์สามารถคิดและตัดสินใจได้อย่างมีประสิทธิภาพ โดยมีการกำหนดข้อมูลแบบสามัญการ (generalization) หรือ ข้อมูลการชำนาญเฉพาะวิชา (specialization) ที่เป็นลำดับขั้นของแนวคิด และออนโทโลจีเป็นชุดตัวแบบโดเมนขององค์ความรู้ (model a domain of knowledge) ที่ประกอบด้วยสมาชิกของข้อมูลในรูปแบบ มโนทัศน์ (concept) หรือคลาส (classes) ที่สามารถอธิบายคุณลักษณะของข้อมูลด้วยคุณสมบัติ (properties) (Gruber, 2009) โดยสามารถสรุปส่วนประกอบที่สำคัญของออนโทโลจีได้ดังนี้

1) คลาสหรือมโนทัศน์ เป็นขอบเขตของความรู้ (knowledge) หรือความคิดทั่วไป ที่สามารถอธิบายรายละเอียดได้ โดยคลาสจะมีความสัมพันธ์ระหว่างสมาชิกของคลาส ซึ่งแต่ละคลาสจะถูกกำหนดไว้อย่างชัดเจนและสามารถอธิบายความหมายของข้อมูลได้ถูกต้อง เช่น บุคคล คำศัพท์ ส่วนประกอบคอมพิวเตอร์ และการให้บริการโครงสร้างพื้นฐานคลาวด์

2) พร็อพเพอร์ตี้ (properties) หรือแอททริบิว (attributes) เป็นสิ่งที่ใช้ประกอบการอธิบายลักษณะเฉพาะของคลาส และมีความสัมพันธ์กันในคลาส หรือระหว่างคลาส เช่น คลาสอาจารย์ประกอบด้วย ตำแหน่ง วิชาที่สอน สถานที่ทำงาน ซึ่งเรียกว่าดาต้าพร็อพเพอร์ตี้ (data properties)

กรณีที่มีความสัมพันธ์ระหว่างคลาสจะใช้อ็อบเจกต์พร็อพเพอร์ตี้ (object properties) ในการอธิบาย เช่น คลาสนักเรียนมีความสัมพันธ์กับคลาสวิชาด้วยการเรียนเป็นอ็อบเจกต์พร็อพเพอร์ตี้ คลาสอาจารย์มีความสัมพันธ์กับคลาสวิชาด้วยอ็อบเจกต์พร็อพเพอร์ตี้การสอน โดยแสดงตัวอย่างตามภาพประกอบที่ 8



ภาพประกอบที่ 8 แสดงความสัมพันธ์ของออบเจ็คพร็อพเพอร์ตี้ระหว่างคลาส

3) อินสแตนท์ (instances) หรืออินดิวิดวล (Individuals) เป็นส่วนประกอบพื้นฐานของออนโทโลจี ซึ่งจะกล่าวถึงวัตถุที่เป็นรูปธรรม (concrete objects) เช่น นักเรียนชื่อสมชาย รหัส 580001 เรียนปี 1 คณะเทคโนโลยีสารสนเทศ โดยจะถูกระบุตามพร็อพเพอร์ตี้ของคลาส student โดยหนึ่งอินสแตนท์แทนด้วยชุดข้อมูลหนึ่งคน

โดยส่วนประกอบที่กล่าวมาข้างต้นกล่าวได้ว่าออนโทโลจีเป็นอธิบายข้อมูลให้มีความชัดเจนโดยแบ่งข้อมูลเป็นลำดับขั้นและสามารถทำความสัมพันธ์กับออนโทโลจีอื่นโดยอาจกำหนดสัมพันธ์ของคลาสหรือของอินสแตนท์ ขึ้นอยู่กับการประยุกต์ใช้งานซึ่งออนโทโลจีสามารถประยุกต์ใช้งานอย่างกว้างขว้างในปัจจุบัน โดย Natalya และ Deborah (2014) ได้กล่าวถึงเหตุผลของการนำออนโทโลจีมาใช้งานประกอบด้วย

1) เมื่อต้องการที่จะแบ่งปันความเข้าใจร่วมกันของโครงสร้างของข้อมูลในหมู่คนหรือตัวแทนซอฟแวร์ เช่น สมมติว่าหลายเว็บไซต์ที่แตกต่างกันมีข้อมูลทางการแพทย์หรือการให้บริการ อีคอมเมิร์ซ (e-commerce) ทางการแพทย์ ถ้าเว็บไซต์นี้แบ่งปันและเผยแพร่ออนโทโลจีพื้นฐานเดียวกัน ตัวแทนคอมพิวเตอร์สามารถสกัดข้อมูลโดยรวมจากเว็บไซต์ที่แตกต่างกันเหล่านี้โดยง่ายและถูกต้องตามความหมายของข้อมูล

2) เพื่อเปิดใช้งานการนำกลับมาใช้ใหม่ของโดเมนความรู้เช่น หากเรามีออนโทโลจีจำนวนมากที่มีโดเมนต่างกัน แล้วต้องการสร้างออนโทโลจีขนาดใหญ่โดยครอบครุมออนโทโลจีที่มีอยู่เราสามารถนำออนโทโลจีที่มีอยู่กลับมาใช้ใหม่เพื่ออธิบายโดเมนที่เราสนใจ

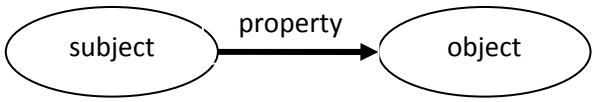
3) เพื่อทำให้สมมติฐานโดเมนมีความชัดเจนเช่น หากเรามีเว็บไซต์ที่มีข้อมูลที่สนใจและต้องการนำมาใช้งานใหม่จะเป็นเรื่องยากเพราะการแสดงข้อมูลของเว็บไซต์จะถูกกำหนดจากรหัสโปรแกรมซึ่งทำให้ยากต่อการทำเข้าใจในข้อมูลเราจำเป็นต้องใช้ออนโทโลจีเข้ามาช่วยในการอธิบายข้อมูลในโดเมนที่กำหนดซึ่งจะทำให้ข้อมูลมีความชัดเจนและตอบสมมติฐานได้ถูกต้องโดยสามารถใช้งานข้อมูลจากใครก็ได้

4) ต้องการสกัดโดเมนความรู้จากความรู้ในการปฏิบัติงานเช่น เราสามารถสร้างออนโทโลจีที่เป็นโดเมนความรู้ของคอมพิวเตอร์ส่วนบุคคลจากส่วนประกอบคอมพิวเตอร์และขั้นตอนวิธีในการกำหนดโครงแบบคอมพิวเตอร์

5) ต้องการวิเคราะห์โดเมนของความรู้เช่น การนำออนโทโลจีที่มีอยู่ซึ่งจะแสดงคุณสมบัติเฉพาะของข้อมูลและข้อตกลง แล้วทำการวิเคราะห์เพื่อหาผลลัพธ์ หรือเพื่อสร้างองค์ความรู้ใหม่

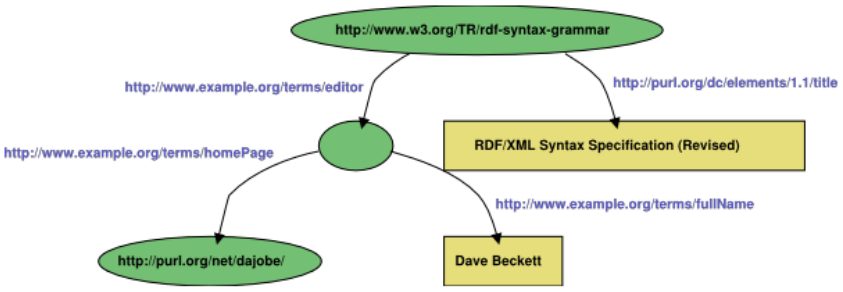
การพัฒนาออนโทโลจีจะคล้ายกับการกำหนดชุดข้อมูลและโครงสร้างข้อมูลที่สนใจสำหรับการแก้ปัญหา หรือให้บริการองค์ความรู้สำหรับโปรแกรมอื่นๆ ที่จะมาใช้งาน โดยการพัฒนาออนโทโลจีมีภาษาที่ใช้พัฒนาหลายรุ่น ซึ่งสมาคมเวิลด์ไวด์เว็บ (World Wide Web Consortium, 2016) หรือดับบลิวสามซี (W3C) ได้กล่าวถึงการสร้างออนโทโลจีจำเป็นต้องอาศัยภาษาที่มีโครงสร้าง ตัวอย่างเช่น

1) อาร์ดีเอฟ (RDF) หรือกรอบแนวคิดการพรรณนาทรัพยากร (Resource Description Framework) เป็นกรอบแนวคิดในการอธิบายสารสนเทศบนเว็บโดยสามารถเชื่อมโยงไวยากรณ์ของนามธรรมเพื่อสื่อถึงความหมายของสิ่งต่างๆ โดยรูปแบบจากเรื่องที่สนใจและมีพร็อพเพอร์ตี้เป็นอ็อบเจกต์ในการอธิบายเรื่องนั้นๆ โดยสามารถอธิบายในรูปแบบกราฟตามภาพประกอบที่ 9



ภาพประกอบที่ 9 แสดงตัวแบบความสัมพันธ์ของข้อมูลด้วยภาษาอาร์ดีเอฟ (w3c, 2004)

อาร์ดีเอฟสามารถเขียนในรูปแบบไวยกรณ์ภาษาเอ็กเอ็มแอล(XML syntax) เปรียบเทียบกับกราฟแสดงความสัมพันธ์ของสารสนเทศดังภาพประกอบที่ 10



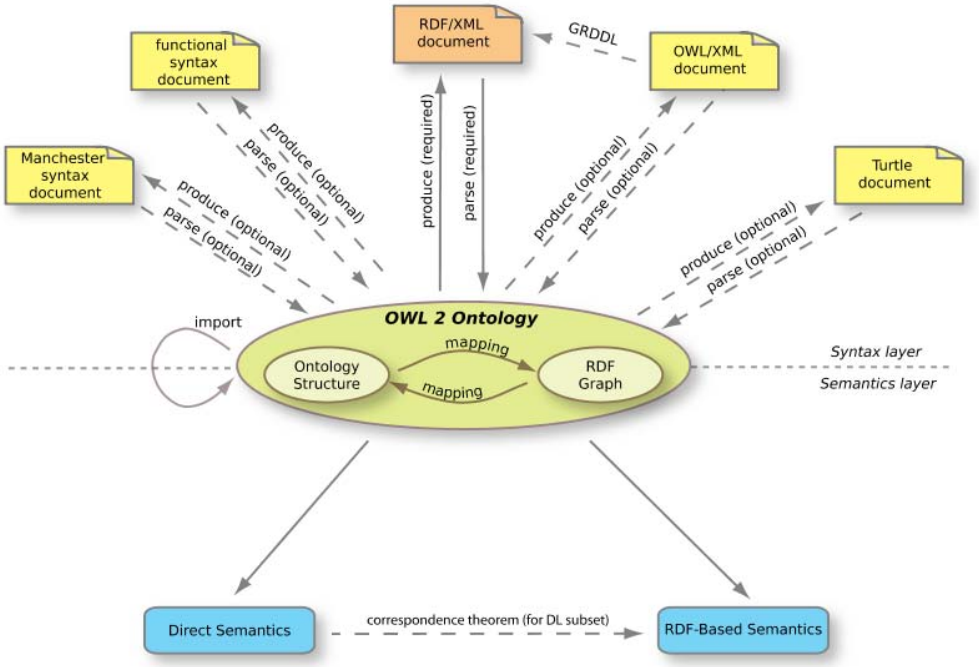
XML syntax for RDF

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
  <ex:editor>
    <rdf:Description>
      <ex:homePage>
        <rdf:Description rdf:about="http://purl.org/net/dajobe/">
        </rdf:Description>
      </ex:homePage>
    </rdf:Description>
  </ex:editor>
</rdf:Description>

<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
  <ex:editor>
    <rdf:Description>
      <ex:fullName>Dave Beckett</ex:fullName>
    </rdf:Description>
  </ex:editor>
</rdf:Description>

<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
  <dc:title>RDF/XML Syntax Specification (Revised)</dc:title>
</rdf:Description>
```

ภาพประกอบที่ 10 แสดงไวยกรณ์ภาษาเอ็กเอ็มแอลสำหรับภาษาอาร์ดีเอฟ (w3c, 2004)

จากภาพประกอบที่ 10 แสดงถึงส่วนย่อยโหนด (Node Elements) และส่วนย่อยพร็อพเพอร์ตี้ (Property Elements) ที่มีความสัมพันธ์ของข้อมูลมีการอธิบายข้อมูลแต่ละชุดว่าเกี่ยวกับเรื่องอะไรด้วยรูปวงรี และอธิบายพร็อพเพอร์ตี้ด้วยเส้นโค้ง ซึ่งทั้งโหนดและพร็อพเพอร์ตี้จะถูกอ้างอิงด้วยยูอาร์ไอ (URI) ที่ไม่ซ้ำกัน และรูปสี่เหลี่ยมอธิบายถึงข้อมูลที่อธิบายตามแอททริบิวที่กำหนดไว้

2) ภาษาเว็บออนโทโลจี (Web Ontology Language) หรือ โอดับบลิวแอล (OWL) ซึ่งเป็นภาษาที่พัฒนาต่อยอดมาจากอาร์ดีเอฟสำหรับออนโทโลจีและเว็บเชิงความหมาย (Semantic Web) ทำให้สามารถกำหนดความหมายของคำ และความสัมพันธ์ของคำได้ชัดเจนยิ่งขึ้น โดยโอดับบลิวแอลประกอบด้วย คลาส (classes) พร็อพเพอร์ตี้ (properties) อินดิวิดวล (individuals) และค่าของข้อมูล (data values) โดยทั้งหมดจะถูกเก็บไว้ในรูปแบบเอกสารเว็บเชิงความหมาย (Semantic Web documents) และโอดับบลิวแอลสามารถนำออนโทโลจีที่เขียนในรูปแบบภาษาอาร์ดีเอฟมาใช้งานรวมกันได้ ซึ่งปัจจุบันทางองค์การดับบลิวสามซี (W3C) ได้ออกโอดับบลิวแอล 2 (OWL 2) เพื่อปรับปรุงการอำนวยความสะดวกในการพัฒนาออนโทโลจีและการแบ่งปันออนโทโลจีผ่านงานเว็บ โดยภาษาโอดับบลิวแอล 2 มีโครงสร้างดังภาพประกอบที่ 11

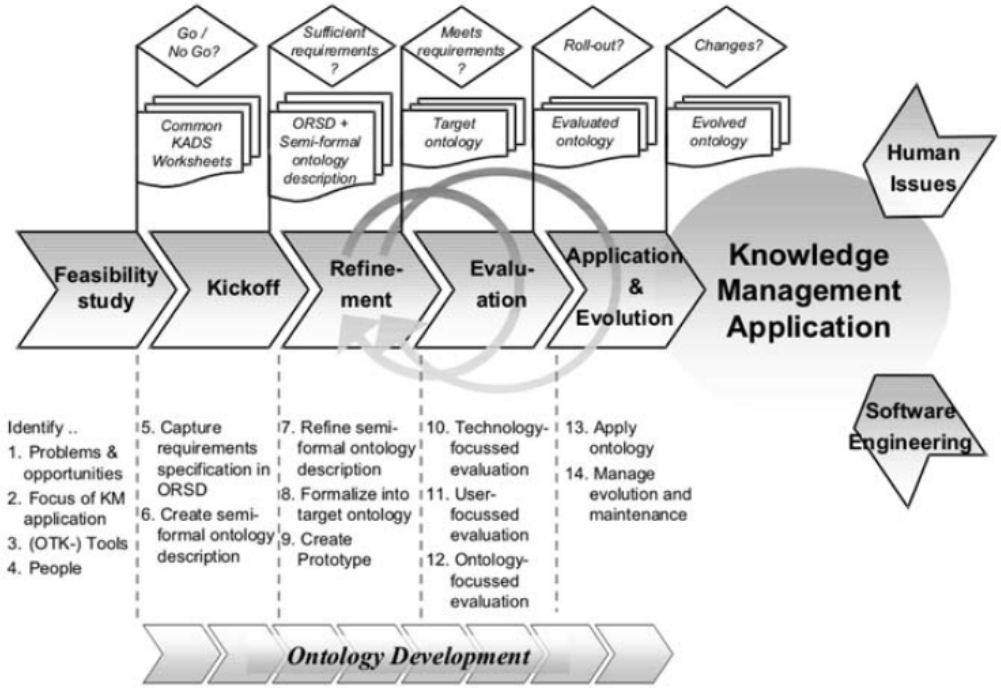


ภาพประกอบที่ 11 แสดงโครงสร้างของโอดับบลิวแอล 2 (W3C, 2012)

จากภาพประกอบที่ 11 เป็นการนำเสนอโครงสร้างของภาษาโอดับบลิวแอล 2 โดยในวงรีที่อยู่กึ่งกลางแสดงออนโทโลจี และการนำออนโทโลจีที่เขียนในรูปแบบภาษาอาร์ดีเอฟแบบกราฟมาใช้งานร่วมกัน ซึ่งในระดับระดับวากยสัมพันธ์ (syntax layer) เป็นการสร้าง หรือสามารถวิเคราะห์คำจากเอกสารต่างๆ เช่นเอกสารโอดับบิวเอลในรูปแบบเอ็กเอ็มแอล (OWL/XML document) เอกสารอาร์ดีเอฟในรูปแบบเอ็กเอ็มแอล (RDF/XML document) เอกสารเทอเลาน์ (turtle document) เอกสารวากยสัมพันธ์แมนเชสเตอร์ (manchester syntax document) และเอกสารฟังก์ชันนัล (functional document) ส่วนระดับเชิงความหมาย (semantics layer) เป็นการระบุความหมายโดยตรง (direct semantics) หรืออาจใช้การระบุความหมายบนฐานอาร์ดีเอฟ (RDF-Based semantics) ซึ่งงานวิจัยนี้เน้นการใช้ ภาษาโอดับบลิวแอล แบบเอกสารโอดับบลิวแอล/เอ็กเอ็มแอล ในการสร้างออนโทโลจี เพราะถูกออกแบบมาเพื่อเว็บเชิงความหมายโดยเฉพาะ

Staab และ Studer (2009) กล่าวว่าการพัฒนาออนโทโลจีมีความจำเป็นต้องมีขั้นตอนและวิธีการพัฒนาโดยมีหลักเกณฑ์ที่น่าเชื่อถือทางวิศวกรรมเรียกว่า วิศวกรรมออนโทโลจี (ontology engineering) การทำวิศวกรรมออนโทโลจีจำเป็นต้องกำหนดการประยุกต์ใช้การจัดการความรู้ (knowledge management application) และพิจารณาอีก 3 กระบวนการย่อย ด้านบุคคล (Human issues) ซึ่งเป็นการยอมรับในองค์ความรู้และสภาพแวดล้อมที่ต้องการใช้องค์ความรู้ ด้านวิศวกรรมซอฟต์แวร์ (software engineering) เพื่อการจัดการความรู้โดยมีข้อกำหนดของซอฟต์แวร์ กระบวนการผลิตซอฟต์แวร์ การประเมิน และการบำรุงรักษา ให้เป็นไปตามหลักวิศวกรรม และ

ด้านกระบวนการเมต้าความรู้ (knowledge meta process) ที่เป็นกระบวนการหลักในการพัฒนาออนโทโลจี (ontology development) ดังแสดงในภาพประกอบที่ 12



ภาพประกอบที่ 12 กระบวนการของวิศวกรรมออนโทโลจี (Staab และ Studer , 2009)

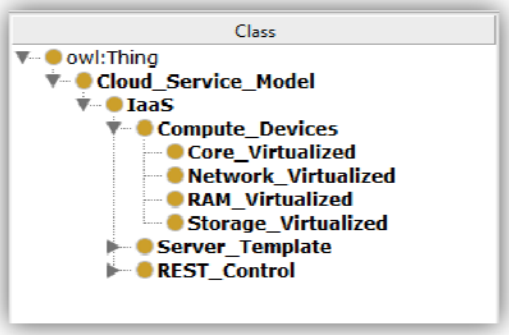
กระบวนการเมต้าความรู้ประกอบด้วย 5 ขั้นตอนหลักเพื่อการพัฒนาออนโทโลจี โดยเริ่มจาก 1) ศึกษาความเป็นไปได้ (feasibility study) ด้านปัญหา เครื่องมือ วิธีการ รวมถึงสภาพแวดล้อม 2) คิ๊กอ๊อฟ (kickoff) เป็นการเริ่มพัฒนาออนโทโลจีจากความต้องการที่รวบรวม 3) การแบ่งละเอียด (refinement) เป็นการปรับแต่งรายละเอียดก่อนที่จะสร้างออนโทโลจีต้นแบบ 4) การประเมิน (evaluation) เป็นการกำหนดวิธีการประเมินที่สามารถมุ่งเน้นด้านเทคโนโลยี ด้านการใช้งาน หรือด้านออนโทโลจี และ 5) การประยุกต์ใช้และวิวัฒนาการ (application and evolution) ซึ่งในขั้นตอนนี้เน้นที่การประยุกต์ใช้งานออนโทโลจีสำหรับสถานการณ์จริง และวิวัฒนาการที่มีลักษณะหรือข้อกำหนดในการปรับปรุงออนโทโลจีให้มีความคงทนและใช้งานในสภาพแวดล้อมในปัจจุบันได้อย่างมีประสิทธิภาพ จากที่กล่าวมาทั้งห้าขั้นตอนสามารถอธิบายรายละเอียดในการพัฒนาออนโทโลจีได้ดังนี้

1) การศึกษาความเป็นไปได้ เป็นการศึกษาประเด็นปัญหา เครื่องมือ วิธีการเทคโนโลยี รวมถึงสภาพแวดล้อมที่นำไปใช้ โดยคำนึงถึงปัจจัยความสำเร็จหรือความล่มเหลวของการพัฒนาองค์

ความรู้ ซึ่งประกอบด้วยสี่ขั้นตอน เริ่มจากการกำหนดปัญหาและโอกาส (problem & opportunity) ของขอบเขตปัญหาที่สนใจ โดยพิจารณาจากปัจจัยต่างๆ เช่น การประหยัดค่าใช้จ่าย ข้อมูลทางเทคนิคที่รองรับ และความเป็นไปได้ของโครงการ จากนั้นกำหนดความสนใจเฉพาะเคเอ็มที่จะนำไปใช้งาน (focus of KM application) แล้วทำการใช้เครื่องมือวันทูโนวเลจ (On-To-Knowledge Tools) หรือเครื่องมือโอทีเค (OTK-tools) เพื่อทำการรวบรวมข้อมูลสำหรับจัดทำออนโทโลจีโดยพิจารณาที่สนับสนุนการทำงานร่วมกันและการนำกลับมาใช้ใหม่ของออนโทโลจี ขั้นตอนการบุคคล (people) โดยกำหนดบุคคลที่เกี่ยวข้องหรือผู้เชี่ยวชาญตรวจสอบข้อมูลที่ได้จากการรวบรวมก่อนนำมาใช้ในการสร้างออนโทโลจี

2) การคิ๊กอ๊อฟ เป็นขั้นตอนการเริ่มสร้างออนโทโลจีโดยการแคปเจอร์ออนโทโลจี (ontology capture) เพื่อกำหนดคุณลักษณะของออนโทโลจีในรูปแบบเอกสารโออาร์เอสดี (ORSD) หรือเอกสารข้อกำหนดคุณลักษณะออนโทโลจี (ontology requirements specification document) จากนั้นทำการสร้างเซมิฟอร์มอลรายละเอียดของออนโทโลจี (crate semi-formal ontology description) ซึ่งอาจใช้มายด์แมพ (mind maps) หรือ ยูเอ็มแอล (UML) ในการอธิบายข้อกำหนดคุณลักษณะออนโทโลจี

3) การแบ่งละเอียด เป็นขั้นตอนการปรับแต่งรายละเอียดก่อนที่จะสร้างออนโทโลจีต้นแบบโดยนำออนโทโลจีแบบกึ่งทางการ (semi-formal ontology) มาพิจารณาจากแหล่งความรู้ที่เกี่ยวข้องเพื่อปรับแต่งโดยเพิ่มเข้าหรือตัดออกจากออนโทโลจี เพื่อได้ออนโทโลจีที่มีคุณภาพสูงทางวิศวกรรม โดยขั้นตอนการปรับแต่งใช้วิธีการพิจารณาเซมิฟอร์มอลออนโทโลจีจากบนลงล่าง (top–down) และพิจารณาจากล่างขึ้นบน (bottom–up) เพื่อพิจารณาองค์ความรู้ที่เกี่ยวข้องแล้วทำการปรับปรุงโดยทำซ้ำจนกว่าจะได้ออนโทโลจีที่มีคุณภาพสูง จากนั้นเซมิฟอร์มอลที่ได้แปลงเป็นออนโทโลจีตามจุดมุ่งหมาย (target ontology) โดยการกำหนดความหมายเช่น “is-a” ของแต่ละคลาส โดยพิจารณาจากโนทัศน์ คุณสมบัติ และความสัมพันธ์ของความหมายของคำ แล้วทำการสร้างต้นแบบ (create prototype) ซึ่งสามารถใช้ซอฟต์แวร์ประยุกต์เช่น ซอฟต์แวร์โพรทีเจ ช่วยการสร้างโดยการเข้ารหัสออนโนโลจี (ontology coding) และเลือกภาษาที่รองรับออนโทโลจีเช่น ภาษาโอดับบลิวแอล เพื่อกำหนดโครงสร้างโดเมนขององค์ความรู้ตามตัวแบบมโนทัศน์ (conceptual model) โดยจะมีลักษณะเป็นออนโทโลจีแบบลำดับชั้น (hierarchical ontology) ดังภาพประกอบที่ 13



ภาพประกอบที่ 13 แสดงออนโทโลจีแบบลำดับชั้น

4) การประเมิน เป็นกระบวนการในการตัดสินคุณค่าของออนโทโลจีที่สร้างขึ้น ในการประเมินออนโทโลจีมีประเด็นที่ให้ความสนใจ ตัวอย่างเช่น

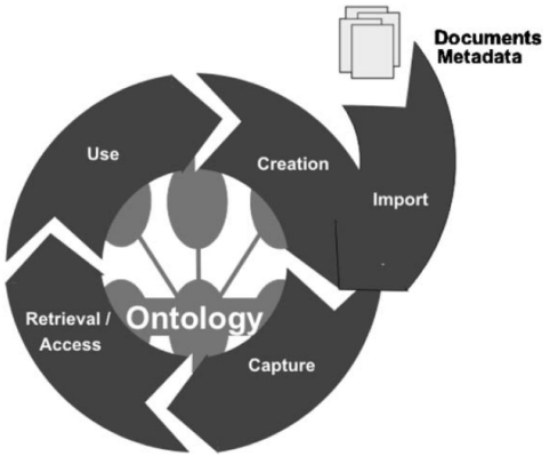
การประเมินการใช้ออนโทโลจีในโปรแกรมประยุกต์ (Evaluate Use Of Ontology In An Application) เป็นการประเมินออนโทโลจีตามกรอบที่ต้องนำไปใช้ประโยชน์โดยกำหนดคำสั่งที่ต้องการทดสอบในรูปแบบภาษาสอบถาม (query language) เพื่อทำการทดสอบความสามารถของออนโทโลจีในการอธิบายองค์ความรู้ที่นำเสนอเช่น ครอบคลุมเนื้อหาในโดเมนที่สนใจ สามารถนำออนโทโลจีกลับมาใช้ใหม่ ความยืดหยุ่นของออนโทโลจี และความสะดวกในการใช้ ซึ่งการประเมินตามที่กล่าวจะต้องกำหนดกรณีที่ต้องการใช้งาน ที่จะอธิบายลักษณะความต้องการของความรู้เป้าหมาย เช่นกรณีที่ต้องการออนโทโลจีแบบเฉพาะเจาะจง จะใช้ตำรา หรือ เอกสารอื่นๆ เพื่อกำหนดความต้องการในการประเมิน (Obrst, et al., 2007)

การประเมินเปรียบเทียบออนโทโลจีกับแหล่งที่มาของเขตข้อมูล (Comparison of Ontology Against a Source of Domain Data) เป็นการประเมินออนโทโลจีจากการเปรียบเทียบข้อมูลกำแหล่งที่มาของข้อมูลโดยการจำแนกประเภทข้อมูลออกเป็นประเด็นต่างๆ แล้วทำการเปรียบเทียบกับออนโทโลจีเพื่อวัดความครอบประเด็นที่สนใจ ซึ่งออนโทโลจีต้องสามารถจับคู่กับข้อมูลประเด็นที่สนใจได้โดยอัตโนมัติและถูกต้อง (Obrst, et al., 2007)

การประเมินโดยมนุษย์กับชุดของเกณฑ์ เป็นการประเมินที่มีการใช้อย่างกว้างขวางในการสร้างออนโทโลจี โดยชุดของคำถามที่ใช้ในการประเมินโดยส่วนใหญ่มาจากความคิดเห็นร่วมกันของผู้เชี่ยวชาญหรือผู้ที่ต้องการใช้ออนโทโลจี เพื่อประเมินการตอบโจทย์ของออนโทโลจีตามวัตถุประสงค์ที่ตั้งไว้ หรือประเมินความสัมพันธ์ของคลาสในออนโทโลจี เช่น is_a และ part_of (Obrst, et al., 2007)

งานวิจัยนี้เน้นการประเมินการใช้ออนโทโลจีในโปรแกรมประยุกต์ซึ่งอ้างอิงแหล่งความรู้จากเอกสารคู่มือของคลาวด์แพลตฟอร์ม และงานวิจัยที่เกี่ยวข้อง

5) การประยุกต์ใช้และวิวัฒนาการ เป็นการประยุกต์ออนโทโลจี (apply ontology) ที่ได้เพื่อใช้งานตามวัตถุประสงค์ โดยมีการจัดการวิวัฒนาการและการบำรุงรักษา (manage evolution and maintenance) ตามกระบวนการของกระบวนการความรู้ (knowledge process) ที่มีวงรอบของกระบวนการตามภาพประกอบที่ 14



ภาพประกอบที่ 14 กระบวนการความรู้ (Staab และ Studer , 2009)

จากภาพประกอบที่ 14 แสดงขั้นตอนกระบวนการความรู้เริ่มจากการการนำเข้าเอกสารเมตดาต้าเช่น เอกสารโอดับบิวเอลในรูปแบบเอ็กเอ็มแอล จากนั้นทำการแคปเจอร์เพื่อบันทึกองค์ความรู้เพื่อเตรียมพร้อม จากนั้นทำการเข้าถึงและการค้นคืนองค์ความรู้ตามคำร้องขอ เพื่อนำความรู้ที่ได้ไปใช้งานแล้วได้ความรู้ใหม่แล้วทำตามวงรอบของกระบวนการความรู้ต่อไป

งานวิจัยนี้เน้นใช้วิศวกรรมออนโทโลจีในการออกแบบองค์ความรู้สำหรับเป็นฐานความรู้ในการให้บริการโครงสร้างพื้นฐานคลาวด์หลายแพลตฟอร์ม

5.งานวิจัยที่เกี่ยวข้อง

ผู้วิจัยได้ทำการศึกษางานวิจัยที่เกี่ยวข้องของการทำงานร่วมกันของบริการการคำนวณแบบคลาวด์หลายแพลตฟอร์ม โดยสามารถจำแนกเป็น 4 กลุ่มดังนี้

1) ออนโทโลจีการคำนวณแบบคลาวด์ (cloud ontology) คืองานวิจัยที่กล่าวถึงแนวคิด และวิธีการสร้างออนโทโลจีสำหรับบริการของการคำนวณแบบคลาวด์โดยมีรายละเอียดของงานวิจัยที่เกี่ยวข้องดังนี้

Alfazi, et. al., (2015) ได้เสนอสถาปัตยกรรมโปรแกรมค้นหาบริการการคำนวณแบบคลาวด์ (Cloud Service Search Engine) โดยใช้ออนโทโลจีสำหรับการระบุประเภทของบริการการคำนวณแบบคลาวด์ โดยมุ่งเน้นการระบุและจัดประเภทของบริการการคำนวณแบบคลาวด์โดยอัตโนมัติ ซึ่งงานวิจัยดังกล่าวได้เสนอแนวคิดให้ลงทะเบียนผู้ให้บริการกับระบบดังกล่าว จากนั้นผู้ใช้บริการป้อนข้อคำถาม (Query) ของความต้องการใช้ผ่านส่วนต่อประสานที่กำหนด ระบบจะทำ

การระบุหมวดหมู่ความต้องการใช้บริการ โดยพิจารณาจาก 3 กรณี 1) ตัวประมวลผลที่คล้ายคลึงกัน (similarity processor) ที่มีบริการตรงความต้องการของผู้ใช้บริการ 2) แนวคิดของการถ่วงน้ำหนัก (concepts weight) ซึ่งพิจารณาจากความถี่ที่มีการใช้บริการ และ 3) แนวคิดการตรวจสอบ (concepts investigator) ซึ่งจะมีการบันทึกการใช้บริการเพื่อทำการตรวจสอบว่ามีบริการใดพร้อมใช้งานอยู่ในขณะนั้นแล้วนำมาเปรียบเทียบกับออนโทโลจีที่สร้างไว้ ซึ่งงานวิจัยดังกล่าวได้เสนอการสร้างออนโทโลจีที่ครอบคลุมทั้งสามตัวแบบบริการการคำนวณแบบคลาวด์ประกอบด้วย 1) คลาสบริการซอฟต์แวร์คลาวด์ที่ประกอบด้วยซับคลาสดังนี้ เดสก์ท็อปเสมือน (Virsual_Desktop) ซอฟต์แวร์ (Software) และเว็บเบสอีเมล (web_based_email) 2) คลาสบริการแพลตฟอร์มคลาวด์ประกอบด้วยซับคลาสดังนี้ ส่วนต่อประสานโปรแกรมประยุกต์ (application_program_interface) เครื่องมือที่ใช้พัฒนาซอฟต์แวร์(software_development_tools) สภาพแวดล้อมการพัฒนาแบบแบบบูรณาการ (integrated_development_environment) และ 3) คลาสบริการโครงสร้างพื้นฐานคลาวด์ประกอบด้วย ซับคลาสดังนี้ โครงสร้างพื้นฐานที่อยู่ในรูปแบบเครื่องเสมือน หน่วยเก็บข้อมูล หน่วยประมวลผล และเครือข่ายคอมพิวเตอร์ ซึ่งรายละเอียดของออนโทโลจีดังกล่าวจะถูกนำมาใช้ในการออกแบบออนโทโลจีสำหรับการวิจัยครั้งนี้

Ngo, et. al., (2015) ได้เสนอออนโทโลจีสำหรับการกำหนดการเข้าถึงการใช้บริการโครงสร้างพื้นฐานคลาวด์ที่สามารถสร้างนโยบายการเข้าถึงด้วยลักษณะประจำ (attribute) โดยอัตโนมัติที่มีความสัมพันธ์กับรายละเอียดทรัพยากรโครงสร้างพื้นฐานคลาวด์ของผู้ให้บริการที่สนับสนุนการให้บริการการคำนวณแบบคลาวด์และอินเตอร์คลาวด์ (intercloud) โดยมีการให้บริการผู้เช่าหลายคน ซึ่งงานวิจัยดังกล่าวใช้ออนโทโลจีไอเอ็นดีแอล (INDL: infrastructure and network description language) (Ghijsen, et. al., 2012) สำหรับจัดการโครงสร้างพื้นฐานคลาวด์ อย่างไรก็ตามงานวิจัยดังกล่าวเป็นการประยุกต์ใช้ออนโทโลจีไอเอ็นดีแอลกับการกำหนดการเข้าถึงการใช้บริการ ซึ่งต่างจากการวิจัยครั้งที่ที่เป็นการจัดการทรัพยากรโครงสร้างพื้นฐานคลาวด์หลายเพลตฟอร์ม

Abdullah, et. al., (2014) ได้เสนอฐานความรู้ออนโทโลจีสำหรับตัวแบบบริการคลาวด์ที่มีวัตถุประสงค์เพื่อแสดงองค์ประกอบแนวคิดของบริการการคำนวณแบบคลาวด์สำหรับการจัดการการคำนวณแบบคลาวด์ได้อย่างมีประสิทธิภาพ และสามารถนำองค์ความรู้ออนโทโลจีกลับมาใช้ใหม่ได้ โดยงานวิจัยได้สร้างออนโทโลจีด้วยภาษาอาร์ดีเอฟ (RDF) และได้กล่าวถึงตัวแบบบริการของการคำนวณแบบคลาวด์สามโดเมนหลักที่ครอบคลุมการทำงานของระบบการคำนวณแบบคลาวด์โดยทั่วไปประกอบด้วย ตัวแบบบริการแพลตฟอร์ม (PaaS) ตัวแบบบริการซอฟต์แวร์ (SaaS) และตัวแบบบริการโครงสร้างพื้นฐาน (IaaS) ที่มีรายละเอียดของตัวแบบบริการโครงสร้างพื้นฐานคลาวด์ประกอบด้วยโดเมนต่างๆ คือโดเมนของคอมพิว (compute) ที่กำหนดให้เป็นวิชวล

ไลเซชัน (virtualization) โดเมนของเครือข่าย (network) รูปแบบโพรโทคอล (protocol) ที่ใช้ในการติดต่อสื่อสาร และโดเมนของหน่วยเก็บ (storage) ที่เก็บข้อมูลในรูปแบบต่างๆ ซึ่งรายละเอียดเหล่านี้จะถูกนำมาใช้ในการออกแบบออนโทโลจีสำหรับการวิจัยครั้งนี้

Maheswari และ Karpagam (2014) ได้เสนอสถาปัตยกรรมการค้นพบบริการคลาวด์ (service discovery) ด้วยออนโทโลจีในภาวะฉุกเฉิน โดยมีจุดมุ่งหมายในการจัดสรรบริการการคำนวณแบบคลาวด์สำหรับการจัดการภัยพิบัติและให้การสนับสนุนการปฐมพยาบาลที่ต้องใช้ระบบสารสนเทศ ซึ่งงานวิจัยดังกล่าวได้ทำการสร้างออนโทโลจีสำหรับบริการการคำนวณแบบคลาวด์ด้วยซอฟต์แวร์โพรเตเจ (protégé) และใช้ภาษาโอดับบิวเอล โดยออนโทโลจีที่เสนอไว้ประกอบด้วย 4 โดเมนหลักดังนี้ 1)โดเมนความต้องการใช้งานในภาวะฉุกเฉินเช่น การพยาบาล การบริการอาหาร และการบริการยานพาหนะ 2) บริการซอฟต์แวร์คลาวด์ 3) บริการแพลตฟอร์มคลาวด์ เช่น และ 4) บริการโครงสร้างพื้นฐานคลาวด์ จากนั้นนำออนโทโลจีไปใช้ในสถาปัตยกรรมการค้นพบบริการคลาวด์เพื่อช่วยเปรียบเทียบเชิงความหมายความต้องการใช้งานที่ร้องขอโดยผู้ใช้บริการและกำหนดการเลือกบริการคำนวณแบบคลาวด์ที่ลงทะเบียนไว้ให้ผู้ใช้บริการ ซึ่งแนวคิดการประยุกต์ใช้ออนโทโลจีช่วยในการเปรียบเทียบเชิงความหมายเพื่อเลือกใช้บริการการคำนวณแบบคลาวด์ที่ตอบสนองต่อความต้องการของผู้ใช้บริการของงานวิจัยที่กล่าวมานี้นำมาประยุกต์ใช้กับงานวัจัยในครั้งนี้ในส่วนของการเปรียบเทียบเชิงความหมาย

Ghijsen, et. al., (2013) ได้เสนอตัวแบบส่วนประกอบโหนดภายใน (internal node components) โดยใช้ออนโทโลจีไอเอ็นดีแอล (Ghijsen, et. al., 2012) สำหรับการเฝ้าสังเกตการทำงานและการจัดการทรัพยากรโครงสร้างพื้นฐานคลาวด์เพื่อการประหยัดพลังงาน ซึ่งไอเอ็นดีแอลได้อธิบายทรัพยากรโครงสร้างพื้นฐานคลาวด์ในรูปแบบของเว็บเชิงความหมายโดยจำแนกโดเมนของทรัพยากรโครงสร้างพื้นฐานคลาวด์ ออกเป็นโหนดโดยแต่ละโหนดประกอบด้วย หน่วยเก็บ (storage) ตัวประมวลผล (processor) และหน่วยความจำ (memory) โดยแต่ละซับคลาสจะเก็บข้อมูลของของ ขนาด ความเร็ว แกนการประมวลผล และค่าอื่นๆ ซึ่งจัดเก็บในรูปแบบของดาต้าพร็อพเพอร์ตี้ (data properties) ของอินสแตนซ์ (instance) ซึ่งแนวคิดการลงทะเบียนทรัพยากรโครงสร้างพื้นฐานคลาวด์ในอินสแตนซ์ของออนโทโลจีที่เสนอโดยงานวิจัยข้างต้นได้ถูกนำมาใช้สำหรับงานวิจัยในครั้งนี้

Quinton, et. al., (2013) ได้เสนอตัวแบบสำหรับการปรับแต่งมัลติคลาวด์โดยคำนึงถึงลักษณะเด่นของระบบการคำนวณแบบคลาวด์ และความต้องการใช้ทรัพยากรโครงสร้างพื้นฐานคลาวด์ที่แตกต่างกันของผู้ใช้งานซึ่งมีความเปลี่ยนแปลงอยู่ตลอดเวลา โดยตัวแบบที่เสนอได้สร้างออนโทโลจีตัวแบบฟีเจอร์ (Feature Model) หรือเรียกย่อว่าเอฟเอ็ม (FM) ซึ่งจะมีการกำหนดฟีเจอร์ของระบบการคำนวณแบบคลาวด์ของแต่ละแพลตฟอร์มโดยพิจารณาจาก ภาษาโปรแกรมที่รองรับ

บริการเว็บเซอร์เวอร์ บริการฐานข้อมูล ที่มีความแตกต่างกันแต่ละแพลตฟอร์ม ซึ่งมีความสัมพันธ์กับทรัพยากรโครงสร้างพื้นฐานของคลาวด์ที่ให้บริการอยู่ ซึ่งแนวคิดการกำหนดความต้องการใช้ฟีเจอร์ที่ต่างกันของบริการคลาวด์ถูกนำมาใช้ในการออกแบบออนโทโลจีสำหรับงานวิจัยในครั้งนี้ด้วย

Ghijsen, et. al., (2012) ได้เสนอออนโทโลจีไอเอ็นดีแอล โดยมีจุดมุ่งหมายเพื่ออธิบายทรัพยากรโครงสร้างพื้นฐานคลาวด์ ประกอบด้วย 4 โดเมนหลักดังนี้ 1) จุดต่อ (node) ประกอบด้วยทรัพยากรเสมือน (virtual resource) ทรัพยากรตรรกะ (logical resource) และแหล่งรวมทรัพยากร (resource pool) 2) ส่วนโปรแกรมจุดต่อ (node component) ประกอบด้วย หน่วยความจำ หน่วยประมวลผลกลาง หน่วยเก็บข้อมูล ระบบปฏิบัติการ และส่วนโปรแกรมการสลับสาย (switching component) 3) บริการ (service) และ 4) ส่วนย่อยเครือข่าย (network element) งานวิจัยดังกล่าวนี้ใช้ภาษาเอ็กเอ็มแอล (XML) ในการสร้างออนโทโลจีและได้ถูกนำไปประยุกต์ใช้ในสองโครงการของอียู-เอ็ฟพี7 (EU-FP7) ชื่อโนวี (NOVI: Network Innovation over Virtualized Infrastructures) และกีย์เซอร์ (GEYSERS: **G**reen n**e**tworked data centres as energ**y** pro**s**um**e**r**s** in sma**r**t city environments) เพื่อแสดงให้เห็นถึงความยืดหยุ่นของออนโทโลจีที่ออกแบบว่ารองรับการจัดการทรัพยากรโครงสร้างพื้นฐานคลาวด์ ซึ่งแนวคิดของการกำหนดโดเมนเหล่านี้ถูกนำมาดัดแปลงใช้ในการออกแบบออนโทโลจีสำหรับบริการโครงสร้างพื้นฐานคลาวด์สำหรับงานวิจัยในครั้งนี้

Zhang, et al., (2012) ได้เสนอตัวแบบออนโทโลจีการคำนวณแบบคลาวด์ (Cloud Computing Ontology) หรือเรียกย่อว่าโคโคออน (CoCoOn) ที่เป็นออนโทโลจีสำหรับบริการโครงสร้างพื้นฐานคลาวด์ โดยใช้ภาษาโอดับบิวเอล (OWL) ในการสร้างออนโทโลจี โดยการกำหนดโดเมนที่สนใจของบริการโครงสร้างพื้นฐานคลาวด์เป็นสามโดเมนหลักประกอบด้วยโดเมนของคอมพิว (compute) ที่กำหนดให้เป็นวิชวลไลเซชัน (virtualization) โดเมนของเครือข่าย (network) ที่สามารถกำหนดความกว้างแถบความถี่ (bandwidth) และรูปแบบโพรโทคอล (protocol) ที่ใช้ในการติดต่อสื่อสาร และ โดเมนของหน่วยเก็บ (storage) ที่เก็บข้อมูลในรูปแบบต่างๆ ซึ่งออนโทโลจีที่สร้างขึ้นสามารถรองรับบริการโครงสร้างพื้นฐานคลาวด์หลายแพลตฟอร์มที่ให้บริการเช่น อาเมซอล (amazon) วินโดว์อาเชอร์ (windows azure) และโกกริด (gogrid) ซึ่งแนวคิดของการกำหนดโดเมนเหล่านี้ถูกนำมาใช้ในการออกแบบออนโทโลจีสำหรับบริการโครงสร้างพื้นฐานคลาวด์สำหรับงานวิจัยในครั้งนี้

Kang, et. al., (2011) ได้เสนอกรอบแนวคิดในการนำเว็บเชิงความหมายมาช่วยในการจัดการการคำนวณแบบคลาวด์ โดยสร้างออนโทโลจีแบบโอดับบลิวแอล (OWL) ด้วยภาษาเอ็กเอ็มแอล/อาร์ดีเอฟ (XML/RDF) ประกอบด้วย 4 ออนโทโลจีได้แก่ 1) ออนโทโลจีกระบวนการ (process ontology) ซึ่งเป็นการอธิบายภารกิจของผู้ใช้ 2) โลจีมัลติโดเมน (multi domain ontology)

เป็นการกำหนดการจำแนกประเภทค่าพารามิเตอร์ของกระบวนการที่เป็นนามธรรมซึ่งจะเชื่อมโยงเทคโนโลยีต่างๆ เข้าด้วยกัน 3) โดเมนออนโทโลจี (domain ontology) เป็นการกำหนดเอนทิตีของโปรแกรมประยุกต์และความสัมพันธ์ของบริการการคำนวณแบบคลาวด์ และ 4) ออนโทโลจีโอดับบลิวเอส (OWL-S ontology: OWL for Services ontology) ที่เก็บข้อมูลการบริการของการคำนวณแบบคลาวด์ ซึ่งในงานวิจัยดังกล่าวนี้ได้พัฒนาโปรแกรมประยุกต์ที่ใช้งานออนโทโลจีทั้ง 4 เพื่อทำการจับคู่โดเมนและเลือกบริการการคำนวณแบบคลาวด์ จากนั้นทำการส่งชุดคำสั่งที่ได้ผ่านบริการของเว็บเซอร์วิส (web services) ที่ประกอบด้วยยูดีดีไอ (UDDI) ดับบลิวเอสดีแอล (WSDL) และเอสโอเอพี (SOAP) ซึ่งจะส่งข้อมูลการปฏิบัติเพื่อให้บังเกิดผลกับการคำนวณแบบคลาวด์ ในงานวิจัยครั้งนี้ได้นำแนวคิดการใช้เว็บเซอร์วิสในการส่งชุดคำสั่งสำหรับการจัดการบริการการคำนวณแบบคลาวด์มาใช้แต่เป็นเว็บเซอร์วิส แบบเรสท์ (REST) เพราะเป็นเทคโนโลยีที่ใหม่กว่าและรองรับโดยคลาวด์แพลตฟอร์มทั้ง โอเพนสแตก และอาปาเช่คลาวด์สแตก

Taekgyeong และ Kwang (2010) ได้เสนอระบบซีเอสดีเอส (CSDS: Cloud service discovery system) ที่มีจุดมุ่งหมายเพื่อสนับสนุนผู้ใช้การคำนวณแบบคลาวด์ที่ช่วยในการหาบริการคลาวด์ผ่านอินเทอร์เน็ต โดยซีเอสดีเอสใช้ออนโทโลจีในตรวจสอบความคล้ายคลึงกันของสองบริการคลาวด์ โดยใช้ตัวกระทำซีเอสอาร์เอส (CSRA: Cloud service reasoning agent) ในการเปรียบเทียบซึ่งมีสองฟังก์ชันคือ การใช้เหตุผล (Reasoning) เป็นการเปรียบเทียบความต้องการใช้งานกับออนโทโลจี และการกำหนดอัตรา (Rating) โดยการถ่วงน้ำหนักจากจำนวนเว็บเพ็จทั้งหมดที่เคยสร้างไว้กับรูปแบบความต้องการ และงานวิจัยดังกล่าวนี้ได้เสนอออนโทโลจีสำหรับการคำนวณแบบคลาวด์ประกอบด้วย ตัวแบบบริการแพลตฟอร์ม ตัวแบบบริการซอฟต์แวร์ และตัวแบบบริการโครงสร้างพื้นฐาน ที่มีรายละเอียดของตัวแบบบริการโครงสร้างพื้นฐานคลาวด์ที่ประกอบด้วยโดเมนต่างๆ คือโดเมนซีเอเอเอส (CaaS: Communications as a Service) ที่กำหนดการบริการเครือข่าย โดเมนดีเอเอเอส (DaaS: Data as a Service) ที่กำหนดการบริการข้อมูล อินฟราซอฟต์แวร์ (InfraSoftware) ที่กำหนดบริการของซอฟต์แวร์ต่างๆ และคอมพิวติ้งรีซอร์ส (ComputingResourcse) ที่กำหนดทรัพยากรที่ให้บริการ เช่น หน่วยความจำ หน่วยประมวลผลกลาง และ เครื่องเสมือน ฟังก์ชั่นการใช้เหตุผลและโดเมนเหล่านี้ถูกนำมาดัดแปลงใช้ในการออกแบบออนโทโลจีสำหรับบริการโครงสร้างพื้นฐานคลาวด์สำหรับงานวิจัยในครั้งนี้

งานวิจัยที่เกี่ยวข้องเรื่องออนโทโลจีการคำนวณแบบคลาวด์สามารถสรุปเปรียบเทียบในภาพรวมประกอบด้วย ภาษาที่ใช้ในการพัฒนาออนโทโลจี แพลตฟอร์มบริการโครงสร้างพื้นฐานคลาว์ และโดเมนของทรัพยากรบริการโครงสร้างพื้นฐานคลาวด์ โดยแสดงรายละเอียดในตารางที่ 1

ตารางที่ 1 เปรียบเทียบงานวิจัยที่เกี่ยวข้องกับเรื่องออนโทโลจีการคำนวณแบบคลาวด์

Related work	Cloud ontology																			
	Ontology language		IaaS cloud platform								Domain of resource IaaS									
	OWL	RDF	Cloud Stack	Open Stack	Vm ware Esxi	Open nebula	Eucalyptus	azure	AmazonEC2	other	VM	CPU	memory	storage	network	cost	SLA (uptime)	server function	software product	OS platform
Alfazi, et. al., (2015)		x							x		x	x		x	x					
Ngo, et. al., (2015)	x	x								x	x	x	x	x	x					x
Abdullah, et. al., (2014)		x							x		x	x		x	x					x
Maheswari and Karpagam (2014)	x						x				x		x	x	x				x	
Quinton, et. al., (2013)									x			x	x	x					x	
Ghijsen, et. al., (2013)	x	x		x		x						x	x	x						
Ghijsen, et. al., (2012)	x	x		x		x						x	x	x						
Gurparkash, et. al., (2012)	x	x		x		x					x				x					
Kang, et al., (2011)	x	x						x	x	x	x			x	x	x				
Taekgyeong and Kwang (2010)		x								x	x	x	x	x	x					x
proposed research	x	x	x	x	x						x	x	x	x	x	x	x	x	x	x

จากตารางที่ 1 เป็นการเปรียบเทียบงานวิจัยที่เกี่ยวข้องกับเรื่องออนโทโลจีการคำนวณแบบคลาวด์ซึ่งในงานวิจัยนี้จะมีข้อแตกต่างจากงานวิจัยอื่นสองประเด็นคือ 1)ประเด็นเรื่องแพลตฟอร์มบริการ โครงสร้างพื้นฐานคลาวด์ที่ศึกษา เป็นการทำงานของการจัดการทรัพยากรโครงสร้างพื้นฐานคลาวด์ที่ครอบคลุมทั้งสามแพลตฟอร์มคือ โอเพนสแตก อาปาเช่คลาวด์สแตก และวีเอ็มแวร์ อีเอสเอ็กซ์ไอ และ 2) ประเด็นของทรัพยากรโครงสร้างพื้นฐานคลาวด์ประกอบด้วยเครื่องเสมือน ตัวประมวลผล หน่วยความจำ หน่วยเก็บข้อมูล เครือข่าย ค่าใช้บริการต่ำสุด ช่วงเวลาให้บริการ ฟังก์ชั่นเครื่องให้บริการ ผลิตภัณฑ์ซอฟต์แวร์ และ แพลตฟอร์มระบบปฏิบัติการ

2) การทำงานร่วมกันของการคำนวณแบบคลาวด์ (cloud interoperability) คืองานวิจัยที่กล่าวถึงแนวคิดและวิธีการการทำงานร่วมกันของการคำนวณแบบคลาวด์โดยมีรายละเอียดของงานวิจัยที่เกี่ยวข้องดังนี้

Leonard, et. al., (2016) ได้เสนอขั้นตอนวิธีบีอาร์เคจีเอ-เอ็มซี (BRKGA-MC: Biased Random Key Genetic Algorithm) สำหรับแก้ปัญหาของการจัดการบริการโครงสร้างพื้นฐานคลาวด์ในสภาพแวดล้อมที่มีการใช้บริการจากผู้ให้บริการมากกว่าหนึ่งรายอย่างมีประสิทธิภาพ โดยพิจารณาจากวงเงินงบประมาณ ค่าใช้จ่ายที่เกิดขึ้น และระยะเวลาที่ใช้บริการ ซึ่งมีความสัมพันธ์ความต้องการใช้บริการโครงสร้างพื้นฐานคลาวด์ที่มีการเปลี่ยนแปลงแบบเวลาจริง (real-time) โดยมีระบบนายหน้าในการจัดการเลือกผู้ให้บริการที่ลงทะเบียนไว้ จะได้ผลลัพธ์เป็นโครงแบบสำหรับการใช้บริการโครงสร้างพื้นฐานคลาวด์ แล้วทำการส่งค่าโครงแบบไปยังผู้ให้บริการคลาวด์เพื่อทำการสร้างเครื่องเสมือนสำหรับการใช้งานของผู้ใช้บริการ โดยงานวิจัยดังกล่าวนี้เลือกศึกษาเฉพาะอาเมซอนเว็บเซอร์วิสอีซีทู กูเกิลคอมพิวเอนจิน และไมโครซอฟต์อาเซอร์ ที่เป็นผู้ให้บริการโครงสร้างพื้นฐานคลาวด์ ซึ่งงานวิจัยดังกล่าวเป็นแนวคิดในการสร้างระบบนายหน้าที่มีการทำงานแบบอัตโนมัติและมีการตอบสนองต่อความต้องการใช้บริการแบบเวลาจริง

Panda, et. al., (2015) ได้เสนอตัวแบบและขั้นตอนวิธีการจัดตารางเวลางานสำหรับคลาวด์ลูกผสมที่ต่างกัน เพื่อมุ่งเน้นการบริหารจัดการความต้องการใช้บริการโครงสร้างพื้นฐานคลาวด์ของผู้ใช้บริการในสภาพแวดล้อมของผู้ให้บริการคลาวด์ที่ต่างกัน โดยตัวแบบประกอบด้วยสามส่วนหลักคือ 1) ลูกค้า (customer) หรือผู้ใช้บริการมีความต้องการใช้บริการโครงสร้างพื้นฐานคลาวด์ที่ต่างกันซึ่งเป็นข้อมูลนำเข้า 2) ระบบผู้จัดการคลาวด์ (cloud manager system) ซึ่งทำหน้าที่ในการจัดการความต้องการใช้บริการโครงสร้างพื้นฐานคลาวด์ซึ่งจะเลือกใช้เครื่องเสมือนจากผู้ให้บริการโดยพิจารณาจากข้อกำหนดเช่น หน่วยเก็บข้อมูล เครือข่าย และจำนวนเครื่องเสมือนที่ลงทะเบียนไว้ที่มีความสามารถในการประมวลผลที่ต่างกัน และ 3) ผู้ให้บริการคลาวด์ (cloud service provider: CPS) เป็นผู้ให้บริการโครงสร้างพื้นฐานในรูปแบบเครื่องเสมือน ที่มีกลยุทธ์การกำหนดเวลางานที่ต่างกัน ในงานวิจัยดังกล่าวได้เสนอสามขั้นตอนวิธีเอทีเอส (ATS: allocation-aware task scheduling) ในการเลือกใช้บริการ เริ่มจากขั้นแรกทำการจับคู่ที่พิจารณาตามลำดับของงานแบบคิวจี (Qg: Global queue of all the tasks) และความต้องการใช้บริการที่เข้ามาในระบบ ขั้นที่สองเป็นการพิจารณากลยุทธ์จัดสรรตารางเวลางานของเพลตฟอร์มที่ให้บริการ และขั้นที่สามเป็นการดำเนินการจัดสรรตารางเวลางานตามขั้นตอนวิธีที่กำหนดไว้ ซึ่งงานวิจัยครั้งนี้ได้นำตัวแบบการจัดตารางเวลางานมาประยุกต์ใช้ให้ตรงตามความต้องการของผู้ใช้บริการของคลาวด์ส่วนบุคคล

Wang, et. al., (2015) ได้เสนอตัวแบบซีบีอาร์เอสเอ็ม (CBRSM: cloud service broker-based service selection model) สำหรับการคัดเลือกบริการคลาวด์แบบพลวัตโดยใช้กลไกการเรียนรู้แบบปรับตัว (adaptive learning mechanism) ในสภาพแวดล้อมคลาวด์ลูกผสม เพื่อมุ่งเน้นการเพิ่มประสิทธิภาพการเลือกใช้บริการคลาวด์ ภายใต้เงื่อนไขของบริการคลาวด์ที่มีคุณภาพสูงและราคาการใช้บริการที่ต่ำที่สุด ซึ่งตัวแบบซีบีอาร์เอสเอ็มสามารถแบ่งได้ 3 ระดับชั้นประกอบด้วย

1) ระดับชั้นผู้ใช้บริการ (user layer) ที่รับความต้องการใช้บริการโครงสร้างพื้นฐานคลาวด์ผ่านตัวกระทำผู้ใช้งาน (UA: user agents) ที่รองรับผู้ใช้งานได้หลายคน 2) ระดับชั้นนายหน้าผู้ใช้บริการการคำนวณแบบคลาวด์ (cloud service broker layer) ซึ่งทำหน้าที่ในการเลือกใช้บริการโครงสร้างพื้นฐานคลาวด์ที่ให้บริการอยู่ขณะนั้น ซึ่งใช้วิธีการการคัดเลือกบริการคลาวด์แบบพลวัตโดยใช้กลไกการเรียนรู้แบบปรับตัว และ 3) ระดับชั้นบริการทรัพยากรคลาวด์ (cloud service resource layer) เป็นบริการโครงสร้างพื้นฐานคลาวด์ที่พร้อมใช้งาน ซึ่งมีการเฝ้าสังเกตบริการด้วยตัวแบบเอสเอ็ม (SM:service monitors) และส่งข้อมูลไปยังระดับชั้นนายหน้าผู้ใช้บริการการคำนวณแบบคลาวด์เพื่อเป็นข้อมูลช่วยในการตัดสินใจแบบการเรียนรู้แบบปรับตัวจากบริการทรัพยากรโครงสร้างพื้นฐานคลาวด์ที่มีการเปลี่ยนแปลงอยู่ตลอดเวลา ซึ่งเป็นแนวคิดสำหรับงานวิจัยครั้งนี้ในการนำข้อมูลทรัพยากรพร้อมใช้งานของบริการโครงสร้างพื้นฐานคลาวด์ที่ให้บริการมาเป็นปัจจัยที่ใช้ในการตัดสินใจการเลือกใช้บริการโครงสร้างพื้นฐานคลาวด์

Hioual และ Hemam (2015) ได้เสนอสถาปัตยกรรมสำหรับการจัดทำเว็บเซอร์วิสในสภาพแวดล้อมของการคำนวณแบบคลาวด์หลายระบบ โดยมีจุดมุ่งหมายเพื่อลดค่าใช้จ่ายและปัญหาของการได้ดุลภาระ (Load Balancing) ซึ่งสถาปัตยกรรมดังกล่าวประกอบด้วยส่วนซีเอ (CA: composer agent) ซึ่งเป็นส่วนกำหนดพิกัดและตัวแทนในการดำเนินงานที่จำเป็นและยังเป็นส่วนในการเริ่มกระบวนการส่งข่าวสาร (message) ไปยังส่วนเอ็มเอ (MA: manager agent) ที่เป็นส่วนควบคุมบริการการคำนวณแบบคลาวด์ และค่าใช้จ่าย โดยถูกติดตั้งประจำไว้ที่ระบบการคำนวณแบบคลาวด์ที่ให้บริการ อย่างไรก็ตามงานวิจัยดังกล่าวเป็นการศึกษามุ่งเน้นเพื่อลดค่าใช้จ่ายและปัญหาของการได้ดุลภาระ มิใช่การทำงานร่วมกันของการคำนวณแบบคลาวด์

Copil, et. al., (2014) ได้เสนอตัวแบบสำหรับการควบคุมระบบการคำนวณแบบคลาวด์หลายระบบแบบยืดหยุ่นด้วยภาษาเอสวายบีแอล (SYBL: simple yet beautiful language) ซึ่งตัวแบบดังกล่าวประกอบด้วยปลั๊กอินคลาวด์ (plugin cloud) เฉพาะที่ลงทะเบียนไว้ของแต่ละบริการการคำนวณแบบคลาวด์ ซึ่งแต่ละบริการการคำนวณแบบคลาวด์จะประกอบด้วยส่วนประสานงานการคำนวณแบบคลาวด์ (cloud orchestrator) ที่รับคำสั่งผ่านเอพีไอจากนั้นจะส่งชุดคำสั่งไปยัง ส่วนการควบคุมเฉพาะที่อาร์เอสวายบีแอล (rSYBL local control) ซึ่งจะถูกกำหนดข้อบังคับขึ้นอยู่กับเครื่องเสมือนเฉพาะที่ซึ่งจะแตกต่างกัน อย่างไรก็ตามงานวิจัยดังกล่าวเน้นการควบคุมภาวะความยืดหยุ่น (elasticity) ในสภาพแวดล้อมของระบบการคำนวณแบบคลาวด์หลายเพลตฟอร์มที่ต่างกัน

Ahmad, et. al., (2014) ได้เสนอตัวแบบซีคลาวด์ (SeaClouds) สำหรับบริหารจัดการโปรแกรมประยุกต์บนระบบการคำนวณแบบคลาวด์หลายระบบ โดยมุ่งเน้นการจัดการการทำงานร่วมกันของบริการแพลตฟอร์มคลาวด์ที่แตกต่างกันโดยใช้ตัวแบบซีคลาวด์ ซึ่งตัวแบบดังกล่าวประกอบด้วย 4 ส่วนหลักดังนี้ 1) การจัดการ (management) ทำหน้าที่จัดการความต้องการใช้

บริการการคำนวณแบบคลาวด์ของผู้ใช้บริการ 2) เฝ้าสังเกต (monitoring) ทำหน้าที่ตรวจสอบบริการคลาวด์ที่ให้บริการ 3) การปรับตัวสำหรับส่วนการประสานงานโดยอัตโนมัติ (orchestration adaptation) ซึ่งมีความสัมพันธ์กับส่วนการจัดการ และ 4) การค้นพบ (discovery) ทำหน้าที่ค้นหาบริการการคำนวณแบบคลาวด์ที่พร้อมให้บริการซึ่งมีความสัมพันธ์กับส่วนเฝ้าสังเกตเพื่อเสนอบริการการคำนวณแบบคลาวด์ที่พร้อมใช้งาน อย่างไรก็ตามงานวิจัยดังกล่าวเป็นการศึกษาเฉพาะบริการแพลตฟอร์มคลาวด์มิใช่บริการโครงสร้างพื้นฐานคลาวด์

Balamurugan, et. al., (2014) ได้เสนอสถาปัตยกรรมการคำนวณแบบคลาวด์สามัญสำหรับการทำงานร่วมกันของการคำนวณแบบคลาวด์ ระหว่างคลาวด์ส่วนบุคคลและคลาวด์สาธารณะโดยเสนอคลาวด์มิดเดิลแวร์เป็นตัวกลางในการประสานการทำงานผ่านเครือข่ายอินเทอร์เน็ต โดยทำการทดสอบคลาวด์ส่วนบุคคลด้วยแพลตฟอร์มยูคาลิปตัส (eucalyptus) และคลาวด์สาธารณะใช้อาเมซอนเว็บเซอร์วิสอีซีทู (AWS EC2)โดยงานวิจัยดังกล่าวได้เสนอกรอบการทำงานที่มีประสิทธิภาพสำหรับการทำงานร่วมกันของการคำนวณแบบคลาวด์ และการรักษาความปลอดภัยในกันทำงานเช่น ทุกคนต้องมีการลงทะเบียนก่อนใช้งาน มีการควบคุมการเข้าถึงโดยใช้ออนโทโลจีมาช่วยในการในการกำหนดสิทธิการเข้าใช้งานทรัพยากร และมีการตรวจสอบแบบสุ่มจากคลาวด์มิดเดิลแวร์ ซึ่งเป็นแนวคิดสำหรับการจัดการทำงานร่วมกันของการคำนวณแบบคลาวด์ผ่านระบบเครือข่ายอินเทอร์เน็ตโดยใช้คลาวด์มิดเดิลแวร์เป็นเครื่องสำหรับการจัดการการทำงานร่วมกัน

Rafique, et. al., (2014) ได้เสนอตัวแบบมิดเดิลแวร์สำหรับการทำงานร่วมกันของคลาวด์ลูกผสม ซึ่งได้สร้างต้นแบบมิดเดิลแวร์ ด้วยภาษาจาวาที่สามารถควบคุมโปรแกรมประยุกต์ผ่านเอพีไอในสภาพแวดล้อมคลาวด์ลูกผสม สำหรับจัดการหน่วยเก็บข้อมูล หน่วยเก็บข้อมูลแบบบล๊อบ (Blob : Binary Large Object) และการปฏิบัติตามงานแบบไม่ประสานเวลา (asynchronous task execution) ซึ่งมิดเดิลแวร์จะทำงานในตัวแบบบริการแพลตฟอร์มของคลาวด์ลูกผสม เช่น เจบอสคลัสเตอร์ลิง (JBoss Clustering) กูเกิลแอพเอนจิน (Google App Engine) และเรทแฮทโอเพ่นชิป (Red Hat OpenShift) ซึ่งเป็นแนวคิดสำหรับการจัดการทำงานร่วมกันของการคำนวณแบบคลาวด์ผ่านเอพีไอ อย่างไรก็ตามงานวิจัยดังกล่าวนี้เป็นการศึกษาการทำงานร่วมกันของตัวแบบบริการแพลตฟอร์ม ในขณะที่งานวิจัยในครั้งนี้มุ่งเน้นการทำงานร่วมกันของบริการโครงสร้างพื้นฐานคลาวด์

Tamas และ Attila (2013) ได้เสนอการใช้หน่วยเก็บข้อมูลบนโครงสร้างพื้นฐานคลาวด์หลายคลาวด์ส่วนบุคคลร่วมกัน ซึ่งทำทดสอบการใช้งานผ่านโทรศัพท์เคลื่อนที่และคลาวด์บุคคลใช้แพลตฟอร์มโอเพ่นเนบิวล่า (OpenNebula) เป็นมิดเดิลแวร์ (middleware) เพื่อการจัดการเลือกพื้นที่สำหรับจัดเก็บข้อมูลกับผู้ให้บริการคลาวด์ที่มีบริการโครงสร้างพื้นฐานคลาวด์ที่มีบริการหลายเครื่องเสมือนให้สามารถทำงานร่วมกัน โดยมีการพัฒนาโปรแกรมประยุกต์สำหรับ

ระบบปฏิบัติการแอนดรอยด์ (Android) สำหรับการทดสอบในประเด็น การแสดงรายการ (list) การบรรจุลง (download) การปรับขนาด (resize) การสร้าง (create) และการบรรจุขึ้น (upload) สำหรับไฟล์รูปภาพซึ่งการทดสอบสามารถจัดเก็บข้อมูลที่เครื่องเสมือนต่างๆ เป็นไปตามสมุติฐาน แต่งานวิจัยดังกล่าวนี้เป็นการศึกษาการทำงานร่วมกันของคลาวด์แพลตฟอร์มเพียงหนึ่งแพลตฟอร์มซึ่งต่างจากงานวิจัยครั้งนี้มุ่งศึกษาการทำงานร่วมกันระหว่างหลายแพลตฟอร์มคลาวด์

Canali และ Lancellotti (2013) ได้เสนอวิธีการสำหรับการจัดกลุ่มเครื่องเสมือนอัตโนมัติโดยใช้วิธีการทางสถิติบาตตาชายา (Bhattacharyya) ในสภาพแวดล้อมคลาวด์ลูกผสม โดยมีจุดมุ่งหมายเพื่อเพิ่มประสิทธิภาพในการจัดเก็บข้อมูล โดยงานวิจัยดังกล่าวได้เสนอตัวแบบสำหรับจัดการโดยแบ่งออกเป็น 2 ตัวแบบ 1) เครื่องจัดการกลุ่ม (clustrering enging) สำหรับจัดการใช้งานทรัพยากรของเครื่องเสมือน และจัดการกลุ่มตัวแทนของเครื่องเสมือน 2) ตัวควบคุมคลาวด์ลูกผสม (multi cloud controller) สำหรับทำการรวบรวมข้อมูลที่ใช้ในการตัดสินใจ และตัดสินใจเลือกเครื่องเสมือนที่ตรงกับความต้องการใช้งานโดยใช้สถิติบาสตาชายาในการตรวจสอบความคล้ายคลึงกันของเครื่องเสมือนที่ให้บริการของแต่ละศูนย์ข้อมูลคลาวด์ ซึ่งทั้งสองตัวแบบจะทำงานรวมกันเพื่อช่วยจัดการเครื่องเสมือนของศูนย์ข้อมูลคลาวด์ (cloud data center) ผ่านตัวควบคุมคลาวด์ (cloud controller) ที่ให้บริการของแต่ละศูนย์ข้อมูลคลาวด์ ซึ่งการส่งข้อมูลให้ชุดคำสั่งสำหรับจัดการเครื่องเสมือนจะส่งผ่านเอพีไอไฮเปอร์ไวเซอร์ (hypervisor) ของแต่ละศูนย์ข้อมูลคลาวด์ แม้งานวิจัยนี้ไม่มุ่งเน้นการทำงานร่วมกันของการคำนวณแบบคลาวด์ แต่แนวคิดตัวควบคุมคลาวด์จะถูกนำมาใช้ในงานวิจัยในครั้งนี้

Rafael, et. al., (2010) ได้เสนอการใช้งานมัลติคลาวด์ในรูปแบบการคำนวณแบบคลัสเตอร์สำหรับโปรแกรมประยุกต์ที่มีการคำนวณหลายภารกิจบนคลาวด์คู่ขนานที่เป็นอิสระกันเพื่อแก้ปัญหาการขยายระบบการคำนวณคลาวด์ ประเมินสมรรถนะและค่าใช้จ่ายในการใช้งานระบบที่ต่างกันโดยทำการทดสอบการทำงานร่วมกันระหว่างคลาวด์ส่วนบุคคลที่อยู่ในองค์การซึ่งใช้แพลตฟอร์มโอเพ่นเนบิวล่า (OpenNebula) เพื่อเชื่อมต่อออกไปยังคลาวด์สาธารณะของอาเมซอนอีซีทูที่สหรัฐอเมริกา และอาเมซอนอีซีทูที่ยุโรป อย่างไรก็ตามงานวิจัยงานวิจัยดังกล่าวนี้เป็นการศึกษาเฉพาะแพลตฟอร์มโอเพ่นเนบิวล่าและอาเมซอนอีซีทูเท่านั้น

งานวิจัยที่เกี่ยวข้องเรื่องการทำงานร่วมกันของการคำนวณแบบคลาวด์งานวิจัยที่กล่าวมาสามารถสรุปเปรียบเทียบในภาพรวมประกอบด้วยแพลตฟอร์มบริการโครงสร้างพื้นฐานคลาว์และบริการคลาวด์ โดยมีรายละเอียดตามตารางที่ 2

ตารางที่ 2 เปรียบเทียบงานวิจัยที่เกี่ยวข้องกับเรื่องการทำงานร่วมกันของการคำนวณแบบคลาวด์

Related work	Cloud interoperability											
	IaaS cloud platform									Cloud service		
	Cloud Stack	Open Stack	VM ware Esxi	Open nebula	Eucalyptus	Microsift azure	Amazon EC2	Google compute	other	IaaS	SaaS	PaaS
Leonard, et. al., (2016)						x	x	x		x		
Panda, et. al., (2015)						x	x			x		
Wang, et. al., (2015)						x	x	x	x	x		
Hioual and Hemam (2015)						x	x			x		
Copil, et. al., (2014)									x	x		
Ahmad, et. al., (2014)									x			x
Balamurugan, et. al., (2014)					x		x				x	
Rafique, et. al., (2014)								x	x	x		x
Tamas and Attila (2013)				x						x		
Canali and Lancellotti (2013)									x	x		
Rafael, et. al., (2010)				x			x			x		
proposed research	x	x	x							x		

จากตารางที่ 2 เป็นงานวิจัยที่เกี่ยวข้องเรื่องการทำงานร่วมกันของการคำนวณแบบคลาวด์ ซึ่งในงานวิจัยนี้จะเน้นที่การจัดการทรัพยากรโครงสร้างพื้นฐานคลาวด์ และมีข้อแตกต่างจากงานวิจัยอื่นเรื่องแพลตฟอร์มบริการโครงสร้างพื้นฐานคลาวด์ที่ศึกษา เป็นการทำงานของการจัดการทรัพยากรโครงสร้างพื้นฐานคลาวด์ที่ครอบคลุมทั้งสามแพลตฟอร์มคือโอเพนสแตก อาปาเช่คลาวด์สแตก และวีเอ็มแวร์ อีเอสเอ็กซ์ไอ

3) การทำงานร่วมกันของการคำนวณแบบคลาวด์โดยใช้ออนโทโลจี (cloud interoperability using ontology) คืองานวิจัยที่กล่าวถึงแนวคิดและวิธีการการทำงานร่วมกันของการคำนวณแบบคลาวด์ต่างแพลตฟอร์มโดยอาศัยออนโทโลจีมาช่วยในการอธิบายความหมายของสภาพแวดล้อมบริการคลาวด์สำหรับการตัดสินใจเลือกใช้บริการคลาวด์ โดยมีรายละเอียดของงานวิจัยที่เกี่ยวข้องดังนี้

Galan, et. al., (2015) ได้เสนอตัวแบบบริการโครงสร้างพื้นฐานเอฟเอ็ม (IaaS FM: Infrastructure as a Service Feature Model) ในรูปแบบออนโทโลจีสำหรับบริการโครงสร้างพื้นฐานคลาวด์ ซึ่งมีการกำหนดลักษณะเฉพาะของหน่วยเก็บข้อมูล และแพลตฟอร์มของระบบปฏิบัติการ ที่มีความสัมพันธ์ระหว่างขนาดความจุที่ต้องการใช้งานกับค่าใช้จ่ายในการใช้บริการต่อเดือน ตัวแบบเอเออเอฟเอ็ม (AAFM: Automated Analysis of Feature Models) ถูกใช้ในการจัดการและการวิเคราะห์หาค่าของการกำหนดโครงแบบ (configuration) โดยอัตโนมัติ โดยอาศัยมอดูลแรกคือการลงทะเบียนโครงแบบมาตรฐานแล้วเก็บในตัวแบบบริการโครงสร้างพื้นฐานเอฟเอ็ม โดยงานวิจัยดังกล่าวเลือกใช้อาเมซอนเว็บเซอร์วิสอีซีทูเป็นกรณีศึกษาซึ่งประกอบด้วยบริการต่างๆ ที่มีค่าใช้จ่ายที่ต่างกันและสัมพันธ์กับขนาดของหน่วยเก็บข้อมูล จากนั้นมอดูลที่สองจะทำการวิเคราะห์ความต้องการของผู้ใช้งานเปรียบเทียบกับตัวแบบบริการโครงสร้างพื้นฐานเอฟเอ็ม ซึ่งจะได้ผลลัพธ์เป็นค่าโครงแบบที่ตรงกับความต้องการของผู้ใช้บริการเพื่อส่งไปยังผู้ให้บริการคลาวด์เพื่อทำการสร้างเครื่องเสมือนต่อไป ซึ่งงานวิจัยครั้งนี้นำแนวคิดในการจัดการทรัพยากรโครงสร้างพื้นฐานคลาวด์โดยการกำหนดโครงแบบโดยอัตโนมัติมาใช้ อย่างไรก็ตามงานวิจัยดังกล่าวเป็นการศึกษาเฉพาะแพลตฟอร์มอาเมซอนเว็บเซอร์วิสอีซีทู

Jrad, et. al., (2015) ได้เสนอตัวแบบข้อตกลงระดับบริการ (Service Level Agreement: SAL) สำหรับการจัดการกระแสงานด้านสุขภาพขนาดใหญ่บนคลาวด์ลูกผสม โดยงานวิจัยดังกล่าวได้ยกกรณีศึกษากระแสงานลำดับดีเอ็นเอ (DNA: Deoxyribonucleic Acid) ที่ใช้ทรัพยากรโครงสร้างพื้นฐานคลาวด์แบบคลาวด์ลูกผสมในการประมวลผลงานขนาดใหญ่ โดยมีขั้นตอนวิธี 4 ขั้นตอนเริ่มจาก 1) ทำการจับคู่กัน (match) ระหว่างเครื่องประมวลกระแสงาน (workflow engine) กับบริการโครงสร้างพื้นฐานคลาวด์ที่ลงทะเบียนไว้ที่ออนโทโลจีที่ถูกจัดเก็บบนระบบนายหน้าให้บริการคลาวด์ (cloud service broker system) ซึ่งออนโทโลจีดังกล่าวได้ถูกออกแบบมาเพื่อรองรับการทำงานของบริการโครงสร้างพื้นฐานคลาวด์ประกอบด้วย ออนโทโลจีสำหรับผู้ใช้บริการ และออนโทโลจีสำหรับผู้ให้บริการ ซึ่งออนโทโลจีทั้งสองมีโครงสร้างคลาสหลักเหมือนกันคือคลาสบริการโครงสร้างพื้นฐานคลาวด์ (IaaS_Service) จะมีคลาสย่อยเป็น หน่วยเก็บข้อมูล (Storage) และ เครื่องเสมือน (VM) ส่วนคลาสที่ต่างกันของออนโทโลจีทั้งสองเช่น ออนโทโลจีสำหรับผู้ใช้บริการจะประกอบด้วยคลาสคำร้องขอใช้บริการโครงสร้างพื้นฐานคลาวด์ (IaaS service request) และซับคลาสเป็นความต้องการคุณภาพการให้บริการของผู้ใช้งาน (Qos Requirements) ซึ่งประกอบด้วยเวลาตอบสนอง (Response time) ปริมาณงาน (Throughput) สภาพพร้อมใช้งาน (Availability) และเวลาแฝง (Latency) ส่วนออนโทโลจีสำหรับผู้ให้บริการจะประกอบด้วยคลาสผู้ให้บริการโครงสร้างพื้นฐานคลาวด์ (IaaS cloud provider) และซับคลาสเป็นตัววัดคุณภาพการให้บริการ (Qos Metrics) ซึ่งประกอบด้วย เวลาตอบสนอง ปริมาณงาน และ

สภาพพร้อมใช้งาน 2) ทำการติดตั้งใช้งาน (deploy) โครงสร้างพื้นฐานคลาวด์ในรูปแบบหน่วยเก็บข้อมูลคลาวด์ (cloud storage) ที่มีการทำงานร่วมกันของหลายศูนย์ข้อมูล (data center) ร่วมกัน 3) กระทำการ (execute) โดยการถ่ายโอนกระแสงานจากเครื่องประมวลกระแสงานไปยังหน่วยเก็บข้อมูลคลาวด์เพื่อทำการประมวลผลบนคลาวด์ลูกผสม และ 4) รวบรวมผลลัพธ์ (gather results) ส่งกลับมาที่เครื่องประมวลกระแสงาน ซึ่งงานวิจัยในครั้งนี้ได้นำแนวคิดการออกแบบออนโทโลจีสำหรับผู้ให้บริการของงานวิจัยดังกล่าว เข้ามาช่วยทำการจับคู่กันของความต้องการใช้บริการโครงสร้างพื้นฐานคลาวด์ กับทรัพยากรโครงสร้างพื้นฐานที่ลงทะเบียนไว้

Pérez and Hern'ndez (2012) ได้เสนอสถาปัตยกรรมการจัดตารางเวลาการขอใช้ทรัพยากรโครงสร้างพื้นฐานคลาวด์แบบซีแมนติก (semantic) สำหรับคลาวด์ลูกผสมที่มีการทำงานร่วมกันระหว่างคลาวด์ส่วนบุคคล กับคลาวด์สาธารณะโดยทำการทดสอบคลาวด์ส่วนบุคคลด้วยแพลตฟอร์มโอเพ่นนีบูลล่า (OpenNebula) และคลาวด์สาธารณะใช้อาเมซอนเว็บเซอร์วิสอีซีทู (AWS EC2) ซึ่งมีสร้างออนโทโลจีโดยอ้างอิงจากยูซีไอโครงการออนโทโลจี (UCI project ontologies) บนฐานของยูซีไอโอดับบลิวแอล (uci.owl) แล้วทำการสร้างออนโทโลจีแยกตามองค์ประกอบของข้อมูลเช่น คลาวด์โอดับบลิวแอล (Cloud.owl) อธิบายความหมายขององค์ประกอบของการคำนวณแบบคลาวด์ โอเพ่นนีบูลล่า (OpenNebula.owl) อธิบายความหมายขององค์ประกอบของการแพลตฟอร์มโอเพ่นนีบูลล่า และ อีซีทูโอดับบลิวแอล (EC2.owl) อธิบายความหมายขององค์ประกอบของการแพลตฟอร์มอาเมซอนเว็บเซอร์วิสอีซีทู แล้วทำการจับคู่ออนโทโลจี (Ontology Matching) ตามความต้องการใช้งานเช่นจำนวนซีพียูที่ต้องการใช้ และรูปแบบเครื่องเสมือนที่ต้องการใช้ที่ลงทะเบียนไว้ในออนโทโลจี ซึ่งใช้ข้อคำถามเอสพีเออาร์คิวแอล (SPARQL Queries) ในการเรียกใช้งานออนโทโลจีสำหรับจับคู่ความหมายตามตารางเวลางานที่สร้างไว้สำหรับการสร้างเครื่องเสมือนตามความต้องการของผู้ใช้บริการที่ถูกบันทึกในตารางงาน โดยงานวิจัยที่กล่าวมามุ่งเน้นการลดค่าใช้จ่ายในการใช้บริการโดยการกำหนดตารางการทำงานของชุดคำสั่งในการจัดการโครงสร้างพื้นฐานคลาวด์ตามความต้องใช้งานที่มีการเปลี่ยนแปลงตลอดเวลา อย่างไรก็ตามงานวิจัยงานวิจัยดังกล่าวนี้เป็นการศึกษาเฉพาะแพลตฟอร์มโอเพ่นเนบิวล่าและอาเมซอนอีซีทูเท่านั้น และการนำออนโทโลจีของยูซีไอที่ไม่ได้ถูกออกแบบมาโดยตรงสำหรับงานวิจัยในครั้งนี้อาจส่งผลให้ประสิทธิภาพของระบบลดลง

Somasundaram, et. al., (2012) ได้เสนอตัวแบบการทำงานร่วมกันของบริการโครงสร้างพื้นฐานคลาวด์แบบคลาวด์ส่วนบุคคล ภายใต้กรอบแนวคิดของตัวแบบนายหน้าที่ทำงานเป็นมิดเดิลแวร์ (middleware) และทำหน้าที่จัดสรรทรัพยากรโครงสร้างพื้นฐานคลาวด์แบบอัตโนมัติ โดยใช้องค์ความรู้ออนโทโลจีช่วยในการตัดสินใจความต้องการใช้บริการของผู้ใช้งานที่ต่างกัน โดยระบบถูกออกแบบสำหรับคลาวด์ส่วนบุคคลดังนี้ ยูคาลิปตัส (eucalyptus) และโอเพ่นนีบูลล่า

(opennebula) ซึ่งใช้ตัวแบบซีอาบี มาช่วยในการจัดการความต้องการใช้บริการโครงสร้างพื้นฐานคลาวด์ของทั้งสองแพลตฟอร์ม โดยการตัดสินใจเลือกแพลตฟอร์มบริการโครงสร้างพื้นฐานคลาวด์จะเลือกจากฐานความรู้ออนโทโลจีที่ออกแบบไว้ หลังจากตัดสินใจเลือกใช้บริการระบบจะทำการส่งชุดคำสั่งในรูปแบบเรสท์ฟูล (restful) ไปยังคลาวด์แพลตฟอร์มที่ต้องการใช้บริการได้อย่างมีประสิทธิภาพของสองแพลตฟอร์ม งานวิจัยดังกล่าวไม่รองรับแพลตฟอร์ม คลาวด์สแตก โอเพ่นสแตก และ วีเอ็มแวร์ ซึ่งงานวิจัยครั้งนี้ได้นำตัวแบบซีอาร์บีไปใช้เป็นแนวทางในการออกแบบระบบนายหน้าที่เป็นซอฟต์แวร์ด้วย

Amin, et. al., (2012) ได้เสนอตัวแบบไอซีเอ็มอี (ICME: Inter-cloud message exchange) สำหรับการทำงานร่วมกันของอินเตอร์คลาวด์ ซึ่งมีการทำงานในรูปแบบมิดเดิลแวร์ที่มีการลงทะเบียนโดเมนของการคำนวณแบบคลาวด์ที่ต้องการให้ทำงานร่วมกันผ่านตัวแบบจีดีเอส (GDS: cloud domain over global data space) โดยจีดีเอสจะทำการลงทะเบียนโปรแกรมประยุกต์ของตัวแบบไอซีเอ็มที่ถูกสร้างขึ้นแยกตามอินเตอร์คลาวด์ที่ลงทะเบียน ในการจัดการทรัพยากรโครงสร้างพื้นฐานคลาวด์ของแต่ละอินเตอร์คลาวด์งานวิจัยที่กล่าวนี้ได้นำออนโทโลจีเข้ามาช่วยในการอธิบายความหมายของบริการคลาวด์ โดยใช้ภาษาโอดับบิวเอล แบบอาร์ดีเอฟ ซึ่งประกอบด้วยตัวแบบบริการแพลตฟอร์ม ตัวแบบบริการซอฟต์แวร์ และตัวแบบบริการโครงสร้างพื้นฐานที่กล่าวถึง 3 โดเมนคือ 1) ทรัพยากรโครงสร้างพื้นฐานคลาวด์ในรูปแบบเครื่องเสมือน 2) การสื่อสารข้อมูลแบบคิวโอเอส (QoS: quality of service) ที่ทำงานอยู่บนเครื่องเสมือน และ 3) หน่วยเก็บข้อมูลที่กล่าวถึงข้อมูลในรูปแบบ ระบบแฟ้มข้อมูล และฐานข้อมูลอาร์ดีบีเอ็มเอส (RDBMS) อย่างไรก็ตามงานวิจัยงานวิจัยดังกล่าวนี้เป็นการศึกษาเฉพาะอินเตอร์คลาวด์ที่ทำงานบนเครือข่ายอินเตอร์เน็ต และตัวแบบไอซีเอ็มอีที่ใช้ในการจัดการต้องติดตั้งทุกอินเตอร์คลาวด์ที่มีการเพิ่มขึ้นมาใหม่ อาจส่งผลให้ประสิทธิภาพของระบบลดลง

งานวิจัยที่เกี่ยวข้องเรื่องการทำงานร่วมกันของการคำนวณแบบคลาวด์โดยใช้ออนโทโลจี งานวิจัยที่กล่าวมาสามารถสรุปเปรียบเทียบในภาพรวมประกอบด้วยแพลตฟอร์มบริการโครงสร้างพื้นฐานคลาว์ บริการคลาวด์ ภาษาที่ใช้ในการพัฒนาออนโทโลจี และโดเมนของทรัพยากรบริการโครงสร้างพื้นฐานคลาวด์ โดยมีรายละเอียดตามตารางที่ 3

ตารางที่ 3 เปรียบเทียบงานวิจัยที่เกี่ยวข้องกับเรื่องการทำงานร่วมกันของการคำนวณแบบคลาวด์โดยใช้ออนโทโลจี

Related work	Cloud interoperability using ontology																					
	IaaS cloud platform							Cloud service			Ontology language		Domain of resource IaaS									
	Cloud Stack	Open Stack	VM ware Esxi	Open nebula	Eucalyptus	Amazon EC2	other	IaaS	SaaS	PaaS	OWL	RDF	VM	CPU	memory	storage	network	cost	SLA (uptime)	server function	software product	OS platform
Galán, et. al., (2015)						x		x			x		x	x	x	x		x				x
Jrad, et. al., (2015)						x	x	x			x		x		x	x			x			
Pérez and Hern'ndez (2012)				x		x		x			x	x		x	x	x						x
Somasundaram, et. al., (2012)				x	x			x			x	x		x	x	x					x	x
Amin, et. al., (2012)				x		x	x	x			x	x		x	x	x						x
proposed research	x	x	x					x			x	x	x	x	x	x	x	x	x	x	x	x

จากตารางที่ 3 เป็นงานวิจัยที่เกี่ยวข้องเรื่องการทำงานร่วมกันของการคำนวณแบบคลาวด์โดยใช้ออนโทโลจีซึ่งในงานวิจัยนี้จะเน้นที่การจัดการทรัพยากรโครงสร้างพื้นฐานคลาวด์ซึ่งใช้ภาษาโอดับบลิวแอลและภาษาอาร์ดีเอฟ ในการพัฒนาออนโทโลจี และมีข้อแตกต่างจากงานวิจัยอื่นสองประเด็นคือ 1)ประเด็นเรื่องแพลตฟอร์มบริการโครงสร้างพื้นฐานคลาวด์ที่ศึกษาเป็นการทำงานของการจัดการทรัพยากรโครงสร้างพื้นฐานคลาวด์ที่ครอบคลุมทั้งสามแพลตฟอร์มคือโอเพนสแตก อาปาเช่คลาวด์สแตก และวีเอ็มแวร์ อีเอสเอ็กซ์ไอ และ 2) ประเด็นของทรัพยากรโครงสร้างพื้นฐานคลาวด์ประกอบด้วยเครื่องเสมือน ตัวประมวลผล หน่วยความจำ หน่วยเก็บข้อมูล เครือข่าย ค่าใช้บริการต่ำสุด ช่วงเวลาให้บริการ ฟังก์ชั่นเครื่องให้บริการ ผลิตภัณฑ์ซอฟต์แวร์ และแพลตฟอร์มระบบปฏิบัติการ

4) เวกเตอร์สเปซ (cloud interoperability using ontology) คืองานวิจัยที่กล่าวถึงแนวคิดและวิธีการใช้เวกเตอร์สเปซ มาใช้ในการวัดวัดความคล้ายคลึงเชิงมุมโคไซน์ โดยมีรายละเอียดของงานวิจัยที่เกี่ยวข้องดังนี้

Jovita, et al., (2015) ได้เสนอระบบตอบคำถามที่ตอบสนองต่อความต้องการของผู้ใช้โดยใช้วิธีการวัดความคล้ายคลึงเชิงมุมโคไซน์มาหาคำตอบที่ตรงคำถามมากสุด ซึ่งงานวิจัยดังกล่าวได้เสนอตัวแบบระเบียบวิธีใช้เวกเตอร์สเปซ โดยมีขั้นตอนการทำงานเริ่มจากการส่งข้อคำถามผ่านกระบวนการวิเคราห์คำถามเพื่อจัดรูปแบบชุดคำถามให้อยู่ในรูปแบบที่ต้องการจากนั้นนำไปเปรียบเทียบกับเอกสารคำตอบด้วยคำหลัก (keyword) จะได้ชุดคำตอบที่เข้าชิงตำแหน่ง (candidate answer) จากนั้นใช้เวกเตอร์สเปซวัดความคล้ายคลึงเชิงมุมโคไซน์ระหว่างข้อคำถามกับชุดคำตอบที่เข้าชิงตำแหน่งถ้าคำตอบไหนมีค่าความคล้ายคลึงเชิงมุมโคไซน์จะมีค่าเท่ากับ 1 ถ้าเวกเตอร์ทั้งสองอยู่ในแนวเดียวกันซึ่งหมายความว่ามีความคล้ายคลึงกันสูงสุดหรือเหมือนกัน ค่าความคล้ายคลึงเชิงมุมโคไซน์จะมีค่าต่ำสุดคือ 0 ซึ่งงานวิจัยที่เสนอใช้เวลาในการหาคำตอบ 29 วินาที จาก 25 คำถาม ในการป้อนผ่านระบบเปรียบเทียบกับ 150 ชุดคำตอบที่เก็บฐานข้อมูลที่ออกแบบไว้ ซึ่งงานวิจัยครั้งนี้ได้นำแนวคิดการใช้เวกเตอร์สเปซวัดความคล้ายคลึงเชิงมุมโคไซน์เพื่อหาคำตอบที่ความคล้ายคลึงมากสุดมาใช้เป็นแนวทางในการพัฒนาขั้นตอนวิธีของระบบนายหน้า

ZhiHao, et al., (2012) ได้เสนอแบบจำลองเวกเตอร์สเปซมาช่วยในการเพิ่มประสิทธิภาพในการค้นหาข้อมูลของบริการเว็บความหมายโดยการจัดลำดับตามความเหมือนคล้ายของบริการเว็บความหมายโดยใช้การวัดความคล้ายคลึงเชิงมุมโคไซน์โดยกำหนดเวกเตอร์ในรูปสามมิติประกอบด้วย Veql Vsub และ Vrel ซึ่งแต่ละเวกเตอร์นำมาจากออนโทโลจีของเว็บความหมายที่ถูกเลือกอย่างมีนัยสำคัญ และทำการปรับปรุงความถูกต้องของการค้นหาของผู้ใช้งานตามความถี่และประสบการณ์ของผู้ใช้ อย่างไรก็ตามงานวิจัยดังกล่าวเป็นการจัดลำดับของบริการเว็บความหมายซึ่งต่างจากงานวิจัยที่เสนอใช้เวกเตอร์สเปซมาช่วยจับคู่แผ่นแบบเครื่องบริการที่เก็บในออนโทโลจี

Liu, et al., (2010) ได้นำแบบจำลองเวกเตอร์สเปซมาวัดความเกี่ยวข้องระหว่างคำในโครงสร้างแบบต้นไม้ โดยกำหนดให้แต่ละเวกเตอร์แทนแต่ละโหนดในต้นไม้ แล้วทำการวัดความคล้ายคลึงเชิงมุมโคไซน์ เพื่อหาความคล้ายคลึงเชิงมุมโคไซน์ระหว่างโหนดในต้นไม้เพื่อหาความสัมพันธ์ของข้อมูลโดยค่าความคล้ายคลึงกันสูงสุดของแต่ละโหนดในต้นไม้ ซึ่งการใช้แบบจำลองเวกเตอร์สเปซมาจะมีความถูกต้องมากกว่าการค้นหาตามเงื่อนไขแบบเก่า อย่างไรก็ตามงานวิจัยดังกล่าวเป็นการค้นหาข้อมูลแบบโครงสร้างแบบต้นไม้ซึ่งต่างจากต่างจากงานวิจัยที่เสนอใช้เวกเตอร์สเปซมาช่วยจับคู่แผ่นแบบเครื่องบริการที่เก็บในออนโทโลจี

Eidoon, et al., (2007) ได้เสนอตัวแบบการจับคู่ออนโทโลจีด้วยวิธีเวกเตอร์สเปซซึ่งใช้การวัดความคล้ายคลึงเชิงมุมโคไซน์ของแบบจำลองเวกเตอร์สเปซเพื่อทำการจับคู่ออนโทโลจีสองชนิดที่มีความคล้ายคลึงกันโดยการกำหนดค่าน้ำหนักที่ไม่ใช้ศูนย์เป็นตัวแทนความสัมพันธ์และการสืบทอดของออนโทโลจี ซึ่งในงานวิจัยดังกล่าวได้กำหนดให้โทโลจีที่อยู่ในรูปแบบของเวกเตอร์โดย

กำหนดคลาสและซับคลาสของออนโทโลจีทั้งหมดที่ต้องการจับคู่แล้ววัดความคล้ายคลึงเชิงมุมโคไซน์ระหว่างออนโทโลจีเพื่อเพิ่มประสิทธิภาพของตัวแบบการจับคู่มีความถูกต้องและแม่นยำ ซึ่งงานวิจัยครั้งนี้ได้นำแนวคิดการการจับคู่มาใช้การการจับคู่แผ่นแบบเครื่องบริการที่ถูกจัดในออนโทโลจีของระบบนายหน้า

อย่างไรก็ตามงานวิจัยที่เกี่ยวข้องเหล่านี้มิได้เป็นการออกแบบขั้นตอนวิธีเฉพาะสำหรับการเลือกบริการโครงสร้างพื้นฐานคลาวด์แบบหลายแพลตฟอร์ม