

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง  
การพัฒนาแอปพลิเคชันด้วยเว็บเซอร์วิส  
APPLICATION DEVELOPMENT WITH WEB SERVICE



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาแอปพลิเคชันด้วยเว็บเซอร์วิส  
APPLICATION DEVELOPMENT WITH WEB SERVICE

นายธีรวัฒน์ ชนสุวรรณธาร  
นายนิติพล คุณากรรังสีมันต์  
นายปกรณ์ เรือนทอง

อาจารย์ที่ปรึกษา  
ดร. วรวัฒน์ ลิ้มโกศา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2549

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เว็บเซอร์วิส แอปพลิเคชัน ดีเวลอปเมนต์ II

**Web Services Application Development II**

ผู้จัดทำ

1. นายธีรวัฒน์ ชนสุวรรณธาร รหัสนักศึกษา 46010320
2. นายนิติพล คุณากรรังสีมันต์ รหัสนักศึกษา 46010364
3. นายปกรณ์ เรือนทอง รหัสนักศึกษา 46010396



อาจารย์ที่ปรึกษา

(วรวัฒน์ ลิ้มโกคา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การพัฒนาแอปพลิเคชันด้วยเว็บเซอร์วิส

นาย ชีรวัฒน์ ชนสุวรรณธาร 46010320  
นาย นิติพล คุณากรรังสีมันต์ 46010364  
นาย ปกรณ์ เรือนทอง 46010396  
ดร.วรวัฒน์ ลีมีโกคา อาจารย์ที่ปรึกษา  
ปีการศึกษา 2549

### บทคัดย่อ

ในการพัฒนาโครงการครั้งนี้ ได้นำเสนอการพัฒนาเว็บเซอร์วิสบนคอนเทนต์แพลตฟอร์ม โดยใช้ Microsoft .NET framework 3.0 อันประกอบไปด้วย Windows Communication Foundation, Windows Workflow Foundation และ Windows Presentation Foundation ซึ่งทางผู้จัดทำโครงการได้เลือกใช้ Windows Communication Foundation ในการพัฒนาเว็บเซอร์วิสแอปพลิเคชัน เนื่องจากรองรับการพัฒนาเว็บเซอร์วิสตามมาตรฐานต่างๆ ซึ่งกำหนดโดยองค์กร w3c, oasis และ WS-I รวมไปถึงการนำเอา Windows Workflow Foundation เพื่อเพิ่มความสามารถ ความยืดหยุ่นในการจัดการกับ Business Logic ซึ่งอาจมีการเปลี่ยนแปลงอยู่เป็นประจำ ในการพัฒนาโครงการ ได้มีการนำเอามาตรฐาน WS-Security, WS-Transaction และ WS-Reliable Messaging มาใช้สนองความต้องการในการรับประกันคุณภาพของระบบ นอกจากนี้ยังมีการใช้ Message transfer optimization mechanism (MTOM) มาใช้ในการเข้ารหัสข้อมูลเพื่อให้สามารถส่งข้อมูลที่มีขนาดใหญ่ๆได้ สำหรับเครื่องมือที่ทางผู้จัดทำโครงการเลือกใช้ขึ้นมาใช้ในการพัฒนา คือ Microsoft Visual Studio .NET 2005, Microsoft Windows SDK, SQL Server Management Studio และ SQL Developer โดยมีสภาพแวดล้อมในการพัฒนาดังนี้ SQL Server 2005, Oracle 10g, Windows XP SP2 พร้อม IIS5.1, Windows Server 2003 พร้อม IIS6.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# Web Service Application Development

Terawat Tanasuwantan 46010320

Nitipol Kunakornrangsimun 46010364

Pakorn Ruentong 46010396

Worawat Limpoka Advisor

Academic Year 2006

## Abstract

This report represents .NET Platform Web Service Application development by Microsoft .NET framework 3.0. It comprises Windows Communication Foundation, Windows Workflow Foundation and Windows Presentation Foundation. In this case, we choose Windows Communication Foundation for development since it supports the web service application development that follows by WS-\* Specification. Moreover, Windows Workflow Foundation has been implied to increase the flexibility of Business Logic management that regularly changed. Not only, our additional application entail WS-Security, WS-Transactions and WS-Reliable messaging to guarantee the quality of service, but also append Message transfer optimization mechanism (MTOM) which enhance encoding capability for large messages to stream over the network. The selected development tools are Microsoft Visual Studio .NET 2005, Microsoft Windows SDK, SQL Server Management Studio and SQL Developer within these environment; SQL Server 2005, Oracle 10g, Windows XP SP2 with IIS5.1 and Windows Server 2003 with IIS6.0

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ และคำปรึกษาจาก ดร.วรัฒน์ ลิ้มโกศา ซึ่งเป็นอาจารย์ผู้ควบคุมปริญญาโท ข้าพเจ้ารู้สึกซาบซึ้งในความอนุเคราะห์จากท่านอาจารย์ และขอขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชา ให้กับข้าพเจ้า

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำปริญญาโทฉบับนี้สำเร็จ ลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากปริญญาโทฉบับนี้ ข้าพเจ้าขอบแต่ผู้มีพระคุณทุกท่าน

นาย ชีรวัฒน์ ชนสุวรรณธาร  
นาย นิติพล คุณากรรังสีมันต์  
นาย ปกรณ์ เรือนทอง

## สารบัญ

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญรูป	IX
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์ของโครงการ	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ	2
1.4 ขอบเขตของโครงการ	2
1.5 วิธีการดำเนินงาน	3
1.6 ส่วนประกอบของวิทยานิพนธ์	3
บทที่ 2 ทฤษฎีพื้นฐานที่ใช้ในเว็บไซต์	5
2.1 หลักการของธุรกิจผ่านสื่ออิเล็กทรอนิกส์แบบไดนามิก	5
2.2 Web service	8
2.3 พื้นฐานการใช้เว็บไซต์	8
2.3.1 Extensible Markup Language (XML)	9
2.3.2 Simple Object Access Protocol (SOAP)	9
2.3.3 Web Service Description Language (WSDL)	13
2.3.4 Universal Description, Discovery and Integration (UDDI)	16
2.4 แบบจำลองการใช้เว็บไซต์	18
2.4.1) Service Provider	18
2.4.2) Service Requestor	18
2.4.3) Service Registry	18
2.4.4) UDDI	19
2.4.5) WSDL	19
2.5 คุณลักษณะของ Web service	19
2.6 ประเภทของ Web Service	19
2.7 ประโยชน์ของ Web Services	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

2.8 การเรียกใช้ Web services	20
บทที่ 3 ความปลอดภัย	21
3.1 นิยามความปลอดภัยในคอมพิวเตอร์	21
3.1.1 จุดประสงค์หลักของความปลอดภัยทางข้อมูล	21
3.2 รูปแบบการรักษาความปลอดภัยในเว็บเซอร์วิส	22
3.2.1 พอยท์-ทู-พอยท์ (Point-to-Point)	22
3.2.2 เอ็น-ทู-เอ็น (End-to-End)	22
3.3 ข้อกำหนดด้านการรักษาความปลอดภัยในเว็บเซอร์วิส	23
3.4 WS-Security	24
3.4.1 Authentication	25
3.4.2 Signing	26
3.4.3 Encryption	26
3.5 ไบรรับรองอิเล็กทรอนิกส์ (Certificate)	27
3.5.1 รูปแบบของไบรรับรองอิเล็กทรอนิกส์	28
3.5.2 ข้อมูลพื้นฐานของไบรรับรองอิเล็กทรอนิกส์	28
3.5.3 ตัวอย่างไบรรับรองอิเล็กทรอนิกส์	29
3.5.4 ชนิดของไบรรับรองอิเล็กทรอนิกส์	30
3.6 Secure Socket Layer (SSL)	32
3.6.1 กระบวนการในการเริ่มต้นการสื่อสารผ่านชั้น SSL	33
3.6.2 วิธีการแลกเปลี่ยนกุญแจในการเข้ารหัส (Key exchange method)	35
3.6.3 วิธีการเข้ารหัส	35
บทที่ 4 WS-Transaction	36
4.1 Transaction	36
4.1.1 Traditional Transaction	36
4.1.2 Transaction บนเว็บเซอร์วิส	37
4.2 WS-Coordination	38
4.2.1 Model	39
4.2.2 Extensibility	40
4.2.3 Coordination Context	40
4.2.4 Coordination Service (Coordinator)	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

4.2.5 Coordination Fault	47
4.2.6 Security Model	51
4.3 WS-AtomicTransaction	52
4.3.1 การcommit แบบสองระยะ (Two-Phase Commit Protocol)	52
4.3.2 Atomic Transaction Context	54
4.3.3 Atomic Transaction Protocols	54
4.3.4 Preconditions	55
4.3.5 Completion Protocol	55
4.3.6 ลักษณะของ Two-Phase Commit Protocol ใน WS-AtomicTransaction	56
4.3.7 Transaction Faults	59
บทที่ 5 Business Process Execution Language	60
5.1 ความหมายและหลักการ	60
5.2 โครงสร้างของBpel	61
5.3 การติดต่อกับเว็บเซอร์วิส	64
5.4 เครื่องมือที่ใช้ในการทดลอง	66
บทที่ 6 WS-Reliable Messaging	68
บทที่ 7 แบบจำลองระบบซื้อขายสื่ออิเล็กทรอนิกส์ผ่านเว็บเซอร์วิส	71
7.1 Use Case Diagram	71
7.2 Sequence Diagram	72
7.3 Activity Diagram	76
7.4 Database design	78
7.4.1 Agent database design	78
7.4.2 Provider database design	79
7.4.3 Payment database design	80
7.5 Security Model	81
7.5.1 ความต้องการความปลอดภัย	81
7.5.2 ความสัมพันธ์ระหว่าง Entity กับ Certificate Authority	83
7.5.3 การปฏิบัติในการรักษาความปลอดภัย	84
บทที่ 8 การทดลอง และผลการทดลอง	87
8.1 ฮาร์ดแวร์ และซอฟต์แวร์ที่ใช้	87

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

8.2 ผลการทดลอง	88
8.2.1 Register	89
8.2.3 Search	89
8.2.4 Browse	89
8.2.5 View Transaction	91
8.2.6 Buy Item	91
บทที่ 9 บทวิจารณ์และสรุป	93
9.1 บทวิจารณ์และสรุป	93
9.2 ปัญหาอุปสรรค และแนวทางการแก้ไข	93
9.2.1 ปัญหาการทำงานร่วมกันระหว่างเว็บเซิร์ฟเวอร์ที่พัฒนาโดย .NET และ JAVA	94
9.2.2 ปัญหาในการสืบค้นข้อมูลที่จำเป็นในการใช้งาน	94
9.2.3 ปัญหาในการเลือกใช้เทคโนโลยี และมาตรฐานที่เหมาะสม	95
9.2.4 ปัญหาในการใช้งาน WS-*	95
9.2.5 ปัญหาในการวิเคราะห์ระบบ	95
9.3 แนวทางในการพัฒนาต่อ	96
บรรณานุกรม	97
ภาคผนวก	99
ภาคผนวก ก	100
ภาคผนวก ข	102
ภาคผนวก ค	105
ภาคผนวก ง	108
ภาคผนวก จ	116

## สารบัญตาราง

ตารางที่ 2.1 อธิบายส่วนย่อยของการเชื่อมต่อ	14
ตารางที่ 3.1 แสดงเวอร์ชันและผู้พัฒนาเทคโนโลยี SSL	33
ตารางที่ 4.1 แสดงคุณสมบัติของ [Code] สำหรับ SOAP 1.2	48
ตารางที่ 7.1 แสดงความต้องการในด้านความปลอดภัย	82
ตารางที่ 8.1 แสดงรายละเอียดของฮาร์ดแวร์ที่ใช้	87
ตารางที่ 8.2 แสดงรายละเอียดของซอฟต์แวร์ที่ใช้	87
ตารางที่ 9.1 แสดงปัญหาทางด้าน addressing	94
ตารางที่ 9.2 แสดงปัญหาทางด้าน BPEL Engine	94
ตารางที่ ข.1 แสดงการตั้งค่า และคอมโพเนนท์ที่ใช้ในเว็บเซอร์วิสแต่ละตัว	102
ตารางที่ ข.2 แสดงรายละเอียดของคอมโพเนนท์แต่ละตัว	103
ตารางที่ ข.3 แสดงรายละเอียดของคอมโพเนนท์แต่ละตัว	103

## สารบัญรูป

รูปที่ 2.1 แสดงโครงสร้างพื้นฐานการทำธุรกิจผ่านสื่ออิเล็กทรอนิกส์	5
รูปที่ 1.2 แสดง SOA Model	6
รูปที่ 2.2 แสดง Web Service Architecture	7
รูปที่ 2.3 แสดง SOAP Message	11
รูปที่ 2.5 แสดง SOAP Message format	11
รูปที่ 2.4 แสดงหลักการทำงานของ SOAP อย่างง่าย	13
รูปที่ 2.7 แสดง WSDL Example	15
รูปที่ 2.8 แสดงหลักการทำงานของ WSDL อย่างง่าย	15
รูปที่ 2.9 แสดงความสัมพันธ์ ระหว่างผู้ให้บริการกับผู้ให้บริการ (Service Provider) และ UDDI	16
รูปที่ 2.10 แสดงหลักการทำงานของ UDDI อย่างง่าย	17
รูปที่ 2.11 แสดง Web Services with SOAP, UDDI and WSDL	18
รูปที่ 2.12 แสดง Web Service Model	18
รูปที่ 3.1 แสดงการรักษาความปลอดภัยแบบ พอยท์-ทู-พอยท์	22
รูปที่ 3.2 แสดงการรักษาความปลอดภัยแบบ เอ็น-ทู-เอ็น	23
รูปที่ 3.3 แสดงข้อกำหนดด้านการรักษาความปลอดภัยในเว็บเซอร์วิส	23
รูปที่ 3.4 แสดง Flow ของ WS-Security	25
รูปที่ 3.5 แสดง SOAP ที่เข้ารหัสแล้ว	27
รูปที่ 3.6 แสดงตัวอย่างใบรับรองอิเล็กทรอนิกส์	28
รูปที่ 3.7 แสดงตัวอย่าง Certificate	30
รูปที่ 3.8 แสดงกระบวนการเริ่มต้นการติดต่อสื่อสารของโพรโตคอล SSL	34
รูปที่ 4.1 แสดงตัวอย่างของ Transaction บนเว็บเซอร์วิส	37
รูปที่ 4.2 แสดงเซอร์วิสที่ประกอบภายในCoordinator	39
รูปที่ 4.3 แสดงCoordination Context	42
รูปที่ 4.4 แสดงแอปพลิเคชัน และ Coordinator	43
รูปที่ 4.5 แสดงCreate Coordination Context	44
รูปที่ 4.6 แสดงโครงสร้างของ message CreateCoordinationContext	45
รูปที่ 4.7 แสดงแสดงการคืนค่า Coordination Context	45
รูปที่ 4.8 แสดงรายละเอียดเพิ่มเติม	46
รูปที่ 4.9 แสดงการใช้จุดอ้างอิง Endpoint ระหว่างการลงทะเบียน	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่ 4.10 แสดง SOAP 1.2	49
รูปที่ 4.11 แสดงการทำงานของ การ Commit สองระยะในระบะที่หนึ่ง	53
รูปที่ 4.12 แสดงการทำงานของ การ Commit สองระยะในระบะที่สอง	54
รูปที่ 4.13 แสดงภาพรวมของ Completion Protocol	56
รูปที่ 4.14 แสดงภาพรวมของการ Commit แบบสองระยะ	57
รูปที่ 5.1 แสดง WS-Architecture	61
รูปที่ 5.2 แสดงโครงสร้างของ BPEL	64
รูปที่ 6.1 แสดงชั้นการทำงานตามข้อกำหนด WS-ReliableMessaging	69
รูปที่ 6.2 แสดงตัวอย่างการส่งเมสเสจด้วย WS-ReliableMessaging โปรโตคอล	69
รูปที่ 7.1 แสดง Use case diagram ของระบบ	71
รูปที่ 7.2 แสดง Sequence Diagram การล็อกอินของระบบ	72
รูปที่ 7.3 แสดง Sequence Diagram ในการเลือกไอเท็มของระบบ	73
รูปที่ 7.4 แสดง Sequence Diagram แสดงไอเท็มที่ได้ในระบบ	74
รูปที่ 7.5 แสดง Sequence Diagram การจ่ายเงินของระบบ	75
รูปที่ 7.6 แสดง Activity Diagram ของการล็อกอิน	76
รูปที่ 7.7 แสดง Activity Diagram ของการดูรายละเอียดของไอเท็มในระบบ	77
รูปที่ 7.8 แสดง Database Design ของทาง agent	78
รูปที่ 7.9 แสดง Database Design ของทาง Provider	79
รูปที่ 7.10 แสดง Database Design ของทาง Payment	80
รูปที่ 7.11 แสดงความสัมพันธ์ของส่วนต่างๆของระบบ	81
รูปที่ 7.12 แสดง Protocol ที่ใช้ในการติดต่อของระบบ	83
รูปที่ 7.13 แสดงความสัมพันธ์ระหว่าง CA กับส่วนต่างๆของระบบ	84
รูปที่ 7.14 แสดงวิธีการในการยืนยันสิทธิ์ในการดาวน์โหลดของ User ต่อ Provider	85
รูปที่ 8.1 แสดงหน้าแรกของ Olala Content Service	88
รูปที่ 8.2 แสดงหน้าต่างการ Login เมื่อทำการ login สำเร็จ	88
รูปที่ 8.3 แสดงหน้าต่างการ Register เข้าใช้งาน	88
รูปที่ 8.4 แสดงหน้า Search	89
รูปที่ 8.5 แสดงหน้าแรกของ Olala Content Service	89
รูปที่ 8.6 แสดงหน้า Browse ของ Olala Content Service	90
รูปที่ 8.7 แสดงหน้า Available artist ของ Olala Content Service	90

## สารบัญรูป (ต่อ)

รูปที่ 8.8 แสดงหน้าแรกของ Olala Content Service	91
รูปที่ 8.9 แสดงหน้าแรกของ Olala Content Service	91
รูปที่ 8.10 แสดงหน้าแสดงจำนวนของใน Cart ของ Olala Content Service	92
รูปที่ 8.11 แสดงหน้าแสดงใน cart ของ Olala Content Service	92
รูปที่ ก.1 แสดงความสัมพันธ์ของ MSDTC กับ Transaction	100
รูปที่ ก.2 แสดงการตั้งค่า Security Configuration ใน MSDTC	101
รูปที่ ข.1 แสดงโมเดลที่ใช้ในการทดลองการทำงานของเว็บเซอร์วิส ที่ใช้งานฐานข้อมูลคนละยี่ห้อ	102
รูปที่ ค.1 แสดงรูปแสดงหน้าต่างของ PayPal ที่ผู้ซื้อต้องทำการยืนยันตัวก่อนชำระเงิน	106
รูปที่ ค.2 แสดงรูปแสดงตัวอย่างของแมสเชกที่ใช้ในการพิสูจน์ตัวตนกับทาง PayPal	107
รูปที่ ง.1 แสดงหน้าต่าง Default Web Site Properties	108
รูปที่ ง.2 แสดงหน้าต่าง Web Server Certificate Wizard	109
รูปที่ ง.3 แสดงหน้าต่าง IIS Certificate Wizard	109
รูปที่ ง.4 แสดงหน้าต่าง IIS Certificate Wizard	110
รูปที่ ง.5 แสดงหน้าต่าง IIS Certificate Wizard	110
รูปที่ ง.6 แสดงหน้าต่าง IIS Certificate Wizard	111
รูปที่ ง.7 แสดงหน้าต่าง CA Service	111
รูปที่ ง.8 แสดงหน้าต่าง CA Service	112
รูปที่ ง.9 แสดงหน้าต่าง CA Service	112
รูปที่ ง.10 แสดงหน้าต่าง CA Service	113
รูปที่ ง.11 แสดงวิธีเลือก Certificate Authority	113
รูปที่ ง.12 แสดงการ Issue certificate	114
รูปที่ ง.13 แสดงหน้าต่าง CA Service	115
รูปที่ ง.14 แสดงหน้าต่าง CA Service	115
รูปที่ จ.1 แสดง .Net framework diagram	117

## บทที่ 1

### บทนำ

#### 1.1 ความสำคัญและที่มาของโครงการ

เป็นที่ยอมรับกันว่าเทคโนโลยีสารสนเทศและเครือข่ายอินเทอร์เน็ตเป็นเครื่องมือธุรกิจที่จำเป็นในการดำเนินธุรกิจยุคใหม่ซึ่งเว็บแอปพลิเคชันเป็นเครื่องมือที่สำคัญในการทำธุรกิจบนโลกอินเทอร์เน็ต แต่ในปัจจุบันเว็บแอปพลิเคชันกำลังจะกลายเป็นเพียงควัประอบเนื่องจากเว็บแอปพลิเคชันสามารถตอบสนองความคิด Distributed Processing ได้ในระดับหนึ่งเท่านั้น ด้วยเหตุนี้องค์กรธุรกิจจึงเริ่มหันมาให้ความสนใจกับเทคโนโลยีเว็บเซอร์วิส ซึ่งเข้ามาช่วยเพิ่มประสิทธิภาพให้กับระบบการดำเนินธุรกิจขององค์กร และยังสามารถทำงานร่วมกับเว็บแอปพลิเคชันเดิมได้เป็นอย่างดี

เว็บเซอร์วิสเป็นเทคโนโลยีที่ถูกนำมาใช้ในการพัฒนาอย่างรวดเร็วและแพร่หลาย เนื่องจากเว็บเซอร์วิสสามารถเข้าถึงได้ทุกที่ผ่านทางเว็บ ไม่ขึ้นกับแพลตฟอร์มและรูปแบบของภาษา โดยเว็บเซอร์วิสทำให้แอปพลิเคชันต่าง ๆ นั้นสามารถติดต่อกันได้โดยใช้ภาษา XML เป็นภาษากลางในการสื่อสารผ่านเครือข่ายคอมพิวเตอร์ เว็บเซอร์วิสสามารถที่จะทำงานในสภาวะแวดล้อมที่แตกต่างกันได้โดยผ่าน โพรโตคอลกลาง แต่เนื่องจากบางระบบจะต้องใช้ความปลอดภัยค่อนข้างสูงอย่างเช่น ระบบเกี่ยวกับการเงิน จึงจำเป็นต้องมีการรักษาความปลอดภัยให้กับข้อมูลด้วย เทคโนโลยีที่ใช้ในด้านความปลอดภัยมีหลายอย่างแต่เทคโนโลยีที่ในด้านความปลอดภัยที่ใช้แพร่หลายอยู่ในปัจจุบันนี้ คือ เอสเอสแอล แต่ก็ยังไม่สามารถตอบสนองความต้องการด้านความปลอดภัยในเว็บเซอร์วิสได้ทั้งหมด

จึงได้มีการจัดทำเว็บเซอร์วิสซีเคียวริตี้ขึ้นมา ซึ่งสามารถตอบสนองความต้องการในด้านรักษาความปลอดภัยบนเว็บเซอร์วิสได้มากกว่าแบบเดิม นอกจากนี้ยังมีเว็บเซอร์วิสบางตัวที่ต้องเรียกใช้เว็บเซอร์วิสอื่น เพื่อสร้างเซอร์วิสของตัวเอง เช่น เว็บเซอร์วิสของบริษัททัวร์ ที่ต้องเรียกใช้เว็บเซอร์วิสของ โรงแรมเพื่อจองห้อง, สายการบินเพื่อจองตั๋ว และธนาคารเพื่อตัดยอดเงิน

แต่เนื่องจากเว็บเซอร์วิสประเภทนี้อาจจะมีการเปลี่ยนแปลง Business logic อยู่บ่อยครั้งดังนั้นการผูกแต่ละเว็บเซอร์วิสไว้ด้วยการ coding แบบเก่า นั้น จึงไม่เหมาะกับการสร้าง business process ด้วยเว็บเซอร์วิสตามหลักการของ SOA ดังนั้นจึงมีการใช้ Business Process Execution Language (BPEL) เพื่อช่วยในการแก้ปัญหาดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อศึกษาการทำงาน, สถาปัตยกรรม และส่วนประกอบต่างๆของเว็บเซอร์วิส
- 1.2.2 เพื่อศึกษการรักษาความปลอดภัยของข้อมูลในการส่งผ่านเครือข่ายโดยใช้ WS-security และ SSL
- 1.2.3 เพื่อศึกษามาตรฐานของ WS-transaction
- 1.2.4 เพื่อศึกษาการสร้างเว็บเซอร์วิสที่พัฒนาโดยจาวารวมไปถึงวิธีการที่จะเรียกใช้เว็บเซอร์วิสดังกล่าว
- 1.2.5 เพื่อศึกษาการสร้าง Business Process Execution Language (BPEL) เพื่อช่วยเป็นตัวกลางในการเรียกใช้เว็บเซอร์วิส
- 1.2.6 เพื่อศึกษาส่งข้อมูลแบบ asynchronous โดยใช้ message queue

## 1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1.3.1 ได้รับความรู้ความเข้าใจเกี่ยวกับการสร้างเว็บเซอร์วิสและสามารถนำไปใช้อย่างถูกต้อง
- 1.3.2 เข้าใจหลักการดำเนินงานพื้นฐานของการสร้างความปลอดภัยให้กับข้อมูลที่มีในเว็บเซอร์วิส
- 1.3.3 เข้าใจหลักการและการทำงานของควมน่าเชื่อถือของข้อมูล
- 1.3.4 เข้าใจหลักการของ Business Process Execution Language (BPEL) และสามารถสร้าง BPEL เพื่อเรียกใช้เว็บเซอร์วิส
- 1.3.5 สามารถสร้างเว็บเซอร์วิสขึ้นมาใช้งานได้ถูกต้องและมีประสิทธิภาพ

## 1.4 ขอบเขตของโครงการ

โครงการนี้จะศึกษาเกี่ยวกับรายละเอียดของเว็บเซอร์วิสทั้งหมด ซึ่งประกอบไปด้วย การทำงานของเว็บเซอร์วิส, สถาปัตยกรรมของเว็บเซอร์วิส, เทคโนโลยีที่ใช้ในเว็บเซอร์วิส อันได้แก่ XML, SOAP, WSDL และ UDDI โดยดูว่าในแต่ละเทคโนโลยีที่นำมาใช้นั้นมีการนำมาใช้ในลักษณะใด แต่ละเทคโนโลยีมีความเกี่ยวเนื่องกันอย่างไร โครงสร้างของแต่ละเทคโนโลยีเป็นอย่างไร นอกจากนี้แล้วยังศึกษาในเรื่องของการสร้างเว็บเซอร์วิสด้วย Microsoft Visual Studio 2005 พร้อมทั้งศึกษาเกี่ยวกับความปลอดภัยของเว็บเซอร์วิสโดยใช้ WS-Security และศึกษาวิธีการสร้าง Business Process Execution Language (BPEL) เพื่อใช้เป็นตัวที่ใช้ในการเรียกใช้ Java Web Services และ ASP.net Web Services โดยนำส่วนของความปลอดภัยและเรื่องของ Transaction เข้ามาเกี่ยวข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.5 วิธีการดำเนินงาน

- 1.5.1 ศึกษาสถาปัตยกรรม และรายละเอียดของเว็บเซอร์วิสในส่วนต่างๆ เช่น XML, SOAP, WSDL, UDDI
- 1.5.2 เลือกเครื่องมือที่จะนำมาใช้ในการสร้างเว็บเซอร์วิส ซึ่งในที่นี้เลือกใช้ Microsoft Visual Studio 2005 และศึกษาวิธีการสร้างเว็บเซอร์วิสโดยใช้เครื่องมือนี้
- 1.5.3 ศึกษาการรักษาความปลอดภัยของข้อมูลในการส่งผ่านเครือข่าย โดยใช้ WS-security และ SSL
- 1.5.4 ทดลองสร้างเว็บเซอร์วิสโดยนำ WS-security มาใช้ในการสร้างความปลอดภัยให้กับข้อมูล
- 1.5.5 ศึกษารายละเอียดของ WS-transaction
- 1.5.6 ศึกษารายละเอียดและวิธีการสร้าง Business Process Execution Language (BPEL)
- 1.5.7 ทดลองสร้าง Business Process Execution Language (BPEL) เพื่อใช้เป็นตัวกลางในการไปเรียกใช้เซอร์วิสอื่น
- 1.5.8 ศึกษาและทดลองวิธีการสื่อสารระหว่างเว็บเซอร์วิส (WS-interoperability) ในส่วนของ Java และ Microsoft.Net ให้สามารถติดต่อสื่อสารกันได้อย่างมีประสิทธิภาพ

## 1.6 ส่วนประกอบของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 9 บทด้วยกันคือ

- บทที่ 1 กล่าวถึงความสำคัญและที่มาของโครงการ วัตถุประสงค์ของโครงการ ขอบเขตของโครงการ วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของวิทยานิพนธ์
- บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานของเว็บเซอร์วิสว่าใช้เทคโนโลยีอะไรบ้างในบทนี้จะกล่าวถึงความหมายของ SOA, SOAP, XML, WSDL, UDDI และเว็บเซอร์วิส และส่วนประกอบ
- บทที่ 3 บทนี้จะกล่าวถึงรายละเอียดของความปลอดภัยที่เกี่ยวข้องกับเว็บเซอร์วิส และมาตรฐานต่างๆที่เกี่ยวข้อง
- บทที่ 4 บทนี้จะกล่าวถึงรายละเอียดของ WS-Transaction ซึ่งเป็นส่วนที่คอยควบคุมการทำงานในเว็บเซอร์วิสให้การทำงานสำเร็จซึ่งและทำการทดลองขึ้นมาตรวจสอบ
- บทที่ 5 บทนี้จะกล่าวถึงรายละเอียดของ Business Process Execution Language ซึ่งเป็นภาษาที่นำมาใช้ในการสร้าง Business Process โดยการใช้งานเว็บเซอร์วิส
- บทที่ 6 บทนี้จะกล่าวถึง WS-ReliableMessaging ซึ่งนำมาใช้ในการสร้างรับรอบการส่งถึงของ SOAP ในการเรียกใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7 บทนี้จะกล่าวถึงแบบจำลองระบบซื้อขายสื่ออิเล็กทรอนิกส์ผ่านเว็บเซอรัวิสซึ่งเป็นระบบที่ใช้เป็นตัวอย่างในการนำเว็บเซอรัวิส BPEL และมาตรฐานต่างที่กล่าวถึงในบทก่อนหน้ามาประยุกต์ใช้

บทที่ 8 บทนี้จะแสดงถึงผลการทดลองจากระบบในบทที่ 7

บทที่ 9 สรุป ปัญหาต่างๆที่พบในการทำโครงการนี้



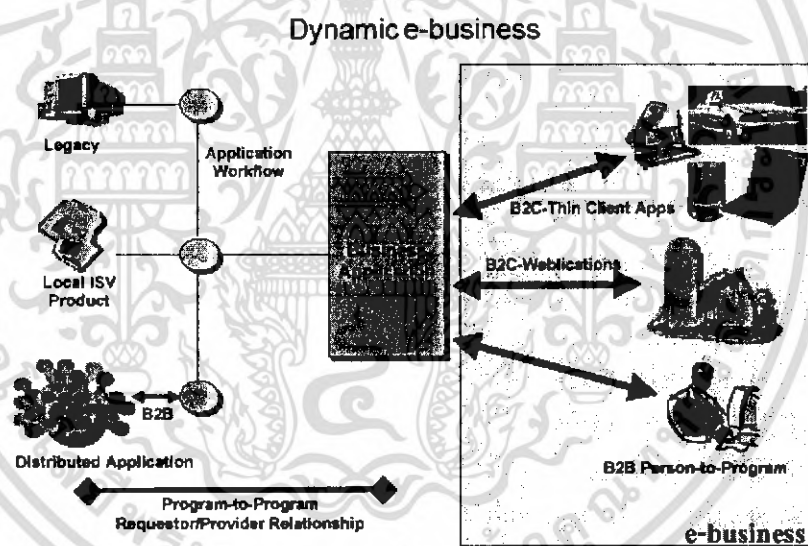
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีพื้นฐานที่ใช้ในเว็บเซอร์วิส

#### 2.1 หลักการของธุรกิจผ่านสื่ออิเล็กทรอนิกส์แบบไดนามิก

ไดนามิก-อีบิซเนสจะมุ่งความสนใจไปยังการยกระดับโครงสร้างพื้นฐาน และการบูรณาการด้านพาณิชย์อิเล็กทรอนิกส์ แบบเชื่อมโยงธุรกิจกับธุรกิจ (Business-to-Business (B2B)) โดยการใช้ประโยชน์จากมาตรฐานบนอินเทอร์เน็ต และโครงสร้างพื้นฐานของอินเทอร์เน็ตให้เกิดประสิทธิภาพสูงสุด ทั้งภายในองค์กรและระหว่างองค์กร ไดนามิก-อีบิซเนสเกิดขึ้นโดยมีความมุ่งหวังว่าธุรกิจที่ดำเนินการผ่านทาง อินเทอร์เน็ตจะสามารถทำการติดต่อสื่อสารกันแบบอัตโนมัติ โดยเป็นการติดต่อกันระหว่าง โปรแกรมกับ โปรแกรม (P2P) หรือระหว่างแอปพลิเคชันกับ แอปพลิเคชัน (A2A) ดังรูปที่ 2.1



รูปที่ 2.1 แสดงโครงสร้างพื้นฐานการทำธุรกิจผ่านสื่ออิเล็กทรอนิกส์

หลักการพื้นฐานที่จะช่วยให้เราสามารถจัดการความยุ่งยากซับซ้อนของการบูรณาการดำเนินธุรกิจแบบ B2B การรวมหรือการบูรณาการของซอฟต์แวร์ต่างระบบกันนั้น จะต้องอนุญาตให้แต่ละระบบเหล่านี้มีความ เป็นอิสระจากกัน (Loosely Coupled) อินเทอร์เน็ตทางด้านบริการของซอฟต์แวร์ที่จะนำมาทำการบูรณาการ ควรจะเผยแพร่สู่สาธารณชน และเปิด โอกาสของการเข้าถึงได้ง่าย

เมสเสจ (Message) ที่ใช้ติดต่อกันของการทำงานแบบ โปรแกรมกับ โปรแกรม (P2P) ต้องสอดคล้องกับมาตรฐาน เปิดบนอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

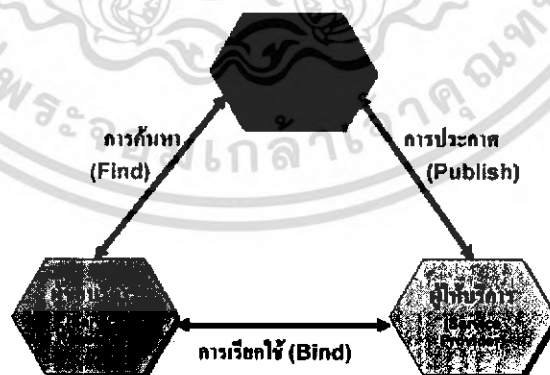
แอปพลิเคชันสามารถสร้างได้จากการใช้ซอฟต์แวร์คอมโพเนนต์ (Software Component) จากภายนอกองค์กร โดยยึดตามแนวทางการดำเนินธุรกิจหลักขององค์กร

แหล่งซอฟต์แวร์คอมโพเนนต์ (Software Component) ที่หาได้ง่าย ช่วยเพิ่มความยืดหยุ่น และเพิ่มคุณสมบัติ ส่วนตัวของกระบวนการทางธุรกิจ

การนำซอฟต์แวร์จากภายนอกองค์กรกลับมาใช้ใหม่ ช่วยให้เกิดการลดต้นทุนและ/หรือช่วยปรับปรุงประสิทธิภาพในการบริหารแก่ลูกค้า

ซอฟต์แวร์สามารถขายเป็นบริการได้ การที่ไคนามิกอีบิสซิเนสเป็นจริงได้ ต้องมีสถาปัตยกรรมพื้นฐานและมาตรฐานอินเทอร์เน็ตแบบเปิดเป็นตัวสนับสนุน สถาปัตยกรรมของธุรกิจผ่านสื่ออิเล็กทรอนิกส์แบบไคนามิก

การพัฒนากระบวนการธุรกิจผ่านสื่ออิเล็กทรอนิกส์อาจจะใช้สถาปัตยกรรมบริการในลักษณะที่เรียกว่า "Service- Oriented Architecture" (SOA) เป็นแนวคิดเบื้องต้นระบบและแอปพลิเคชันส่วนใหญ่ในโลกของธุรกิจที่ใช้งานในปัจจุบันเป็นแอปพลิเคชันและระบบย่อยที่ถูกสร้างขึ้นให้มีการทำงานที่ต้องสัมพันธ์กันอย่างแนบแน่น (Tightly Coupled) การเปลี่ยนแปลงการทำงานภายในระบบย่อยหรือแอปพลิเคชันใด ๆ จะมีผลกระทบกับทั้งระบบซึ่งส่งผลให้การบำรุงรักษามีต้นทุนที่สูงขึ้น รวมทั้งยังเป็นข้อจำกัดในการเชื่อมต่อกับระบบของคู่ค้าอื่นๆ SOA ไม่ใช่แนวคิดใหม่แต่ได้เกิดขึ้นมานานแล้ว ซึ่งอยู่ในจำนวนหนึ่งของแนวคิดทางการออกแบบระบบแบบกระจายศูนย์ (Distributed computing concepts) SOA เป็นแนวคิดที่ได้รับการยอมรับและประสบความสำเร็จ ในขณะที่แนวคิดอื่น ๆ นั้นประสบความสำเร็จเพียงเล็กน้อย การพัฒนา SOA นั้นทำบนมาตรฐานเปิดซึ่งได้รับการรับรองจากผู้ผลิตซอฟต์แวร์ชั้นนำเช่น บริษัทไอบีเอ็มและบริษัทไมโครซอฟต์เป็นต้นและบริษัทเหล่านี้ยังได้ร่วมมือกันรับรองมาตรฐานอย่างเช่น UDDI และ WSDL



รูปที่ 1.2 แสดง SOA Model

การพัฒนา Web Services แบบ encapsulate all level of business functionality คือการเอา web service หลายๆอันมาประกอบกันเป็น function ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Web service architecture มี 3 บทบาท provider requestor broker

Provider จะเป็นคนสร้าง Web Services และทำให้มัน Available สำหรับ Client ที่ต้องการเรียกใช้ซึ่งเป็นคนหรือองค์กรก็ได้ที่สร้าง Agent ขึ้นมา

Requestor คือ Client application ที่ใช้ (consume) Web service เป็น agent ที่ใช้แลกเปลี่ยน message ระหว่างตัวมันเองกับ provider's agent

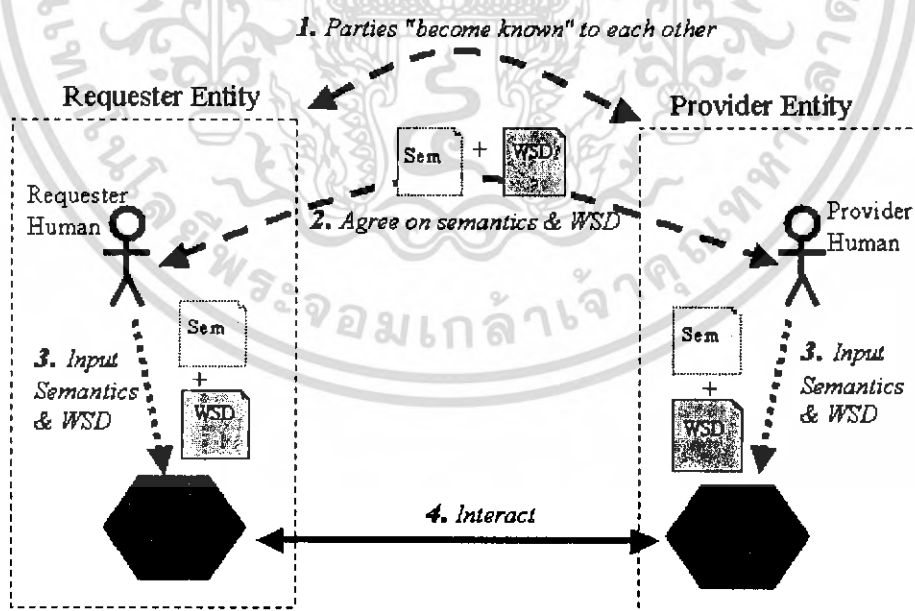
Broker คือ service registry มันจะจัดหาเส้นทางให้ provider กับ requestor ติดต่อกัน Provider จะแจ้งให้ broker ทราบว่า ผมเป็น Web service นะ มีตัวตนอยู่โดยการใช้ brokers publish interface เพื่อให้ service ของ provider สามารถใช้งานได้โดย client (accessible) ข้อมูลที่ถูกประกาศโดย provider จะอธิบาย service และบอกว่า service นี้อยู่ที่ไหน requestor จะถามไปยัง broker ว่า Web service ที่ถูกประกาศแล้ว มันอยู่ที่ไหน ข้อมูลที่ requestor ได้รับจาก broker เกี่ยวกับ Web service นั้น requestor จะผูกติดกับ Web service นั้น มีหลายทางที่ Requester จะใช้งาน Web service โดยทั่วไปจะทำดังนี้ตามรูปข้างล่าง

1. ทั้ง Requester และ Provider ต้องรู้จักกันก่อนหรืออย่างน้อย ฝ่ายใดฝ่ายหนึ่งต้องรู้จักอีกฝ่าย

2. ทั้ง Requester และ Provider ต้องมาตกลงร่วมกันถึงรายละเอียดในการสื่อสารกัน

3. ทั้ง Service description และ semantics ถูกทำให้เกิดขึ้นจริง โดย requestor และ provider

4. ต่อมาจึงเริ่มแลกเปลี่ยนข้อมูลกัน



รูปที่ 2.2 แสดง Web Service Architecture

Semantics ของ Web service คือ การคาดหวังร่วมกัน ถึง พฤติกรรมของบริการ ในส่วนของ การตอบสนองต่อ ข้อความที่ถูกส่งมา Web services จะนำมันมาเป็น พันธะสัญญา ระหว่าง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Requester entity และ Provider entity เพื่อทำตามวัตถุประสงค์และ ผลที่จะเกิดภายหลังได้ถูกต้องตามการคาดหมาย อย่างไรก็ตาม พันธะสัญญาที่ได้แสดงถึง ข้อตกลงทุกอย่าง ในเรื่องของทำอย่างไรและทำไมสำหรับ Agent ในการปฏิสัมพันธ์กัน

ในขณะที่ Service description แสดงถึงข้อตกลงในการดำเนินการติดต่อกัน Semantic จะแสดงถึงข้อตกลงที่ควบคุม ความหมายและวัตถุประสงค์ของ การเชื่อมต่อ อย่างไรก็ตาม การแบ่งเช่นนี้ก็ยังไม่ตายตัว และถือว่าไม่ใช่สาระสำคัญเท่าไร

## 2.2 Web service

Web service คือ อะไร

Web service คือ Application หรือ Program ซึ่งทำงานอย่างไรอย่างหนึ่งในลักษณะให้บริการ โดยจะถูกเรียกใช้งานจาก Application หรือ Program อื่นๆผ่าน Web ซึ่งการให้บริการของ Web Servicesจะมีเอกสารที่อธิบายคุณสมบัติของบริการกำกับไว้ และมีการนำเสนอให้สาธารณชนรับทราบ ผู้ใช้บริการจึงสามารถค้นหา Web services ได้โดยไม่ต้องรู้ที่อยู่จริงของ Application หรือ Program นั้น

**\*\* Application หรือ Program นั้นคือ DLL file นั้นเอง**

Web service นั้นจะต้องมี 2 ส่วนสำคัญคือ

- Web service ต้องมี interface ที่เป็นสาธารณะ ซึ่งเขียนโดย ภาษา XML โดยที่ interface นี้จะอธิบาย ทุกๆ method ที่ Client (ผู้ให้บริการ) สามารถใช้งานได้ การอธิบาย interface นี้จะใช้ WSDL (Web Service Description Language) เป็นตัวอธิบาย
- กลไกที่บอกถึง สถานที่ตั้งของ Web service นั้น กลไก นั้นคือ UDDI (Universal Description, Discovery and Integration)

## 2.3 พื้นฐานการใช้เว็บเซอร์วิส

Service Directory	:	UDDI
Service Description	:	WSDL
Service Interaction	:	SOAP
Data Format	:	XML
Communication Protocol:		HTTP
Communication Network:		Internet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.1 Extensible Markup Language (XML)

เนื่องจาก XML เป็นภาษาที่พัฒนาขึ้นมาจาก HTML ดังนั้นการอธิบายในหัวข้อนี้จะเน้นในเรื่องของความแตกต่างระหว่าง XML และ HTML

XML นั้นใช้อธิบายข้อมูล คือ บอกว่าข้อมูลนั้นคืออะไร ส่วน HTML นั้นใช้อธิบายการแสดงผล คือ บอกว่าข้อมูลนั้นจะแสดงผลในหน้าตาแบบไหน

สำหรับ Tag ใน XML นั้นจะมีรูปแบบคล้ายกับ HTML แต่ต่างกันที่ Tag ของ XML นั้นเกิดจาก Element ที่มีการนิยามมาแบบสำเร็จรูปจากองค์กรที่วางมาตรฐาน เช่น W3C (<http://www.w3c.org>) แล้วเราสามารถนำไปใช้งานได้เลย ในขณะที่ Tag ของ XML นั้น เราต้องทำการนิยาม Element ก่อนจึงจะเอาไปใช้ได้

XML มีความสามารถในการอธิบายความหมายของตัวเองได้ จึงทำให้ Element ของ XML มีความหลากหลายมาก เนื่องจากเรานิยาม Element ได้ด้วยตัวเอง

HTML ภาษาที่ใช้ในการเขียน Web มากที่สุดนั้นเป็นเพราะมีรูปแบบที่ง่ายต่อการแสดงผลของ Browser เนื่องจาก มี Tag ตายตัวที่สามารถบอกได้ว่าเมื่อเจอ Tag นี้จะแสดงผลอย่างไร เช่น เมื่อเจอ tag <B>...</B> ในเอกสารก็ให้แสดงข้อความที่อยู่ระหว่าง tag เป็นตัวหนา แต่จะสังเกตได้ว่าคอมพิวเตอร์จะไม่เข้าใจว่าข้อความนั้นคืออะไร เพียงแต่รู้ว่าจะแสดงผลอย่างไร นั่นแสดงว่าไม่สามารถนำข้อมูลภายใน Tag เหล่านี้ไปทำการประมวลใดๆ ได้เลย

XML เป็นภาษาที่มีลักษณะเป็น Tag คล้าย HTML แต่ไม่ได้มุ่งที่การแสดงผล XML มุ่งที่การสื่อความหมายโดยอนุญาตให้ผู้ใช้สามารถกำหนด Tag ขึ้นได้เองเพื่อให้สื่อความหมายทางภาษาของมนุษย์ แต่คอมพิวเตอร์เองก็เข้าใจเช่นกัน ทำให้ข้อมูลระหว่าง Tag สามารถนำไปประมวลผลต่อได้

### 2.3.2 Simple Object Access Protocol (SOAP)

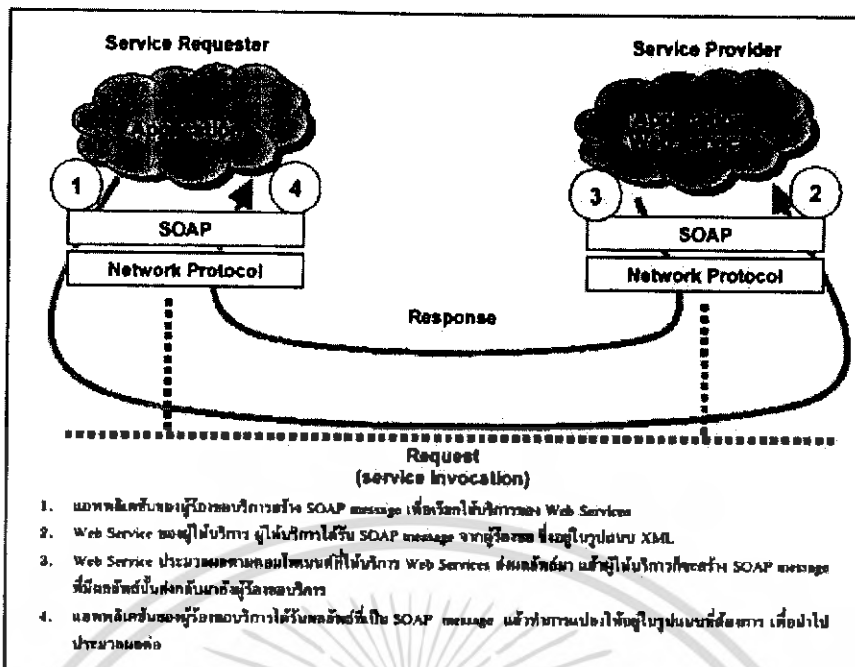
SOAP เป็น protocol ที่ใช้ในการรับส่ง message ระหว่างคอมพิวเตอร์ ซึ่งข้อความของ SOAP นั้น เป็น XML SOAP จะให้คำจำกัดความของ Format of XML Message แต่ไม่ใช้การบอกถึงเนื้อหาและวิธีการส่ง SOAP จะบอกว่า จะส่ง SOAP message โดยหาเส้นทางใน Http อย่างไร ถึงแม้ว่า SOAP ถูกใช้งานได้ในระบบการส่งข้อความหลายๆแบบ และสามารถส่งผ่าน Protocol ได้หลายๆแบบ แต่ส่วนใหญ่จะเป็นการทำงานบน HTTP

เนื่องจากระบบปฏิบัติการสำคัญๆ ทุกตัว เช่น วินโดวส์ Unix Linux และ Macintosh ต่างก็สามารถใช้งานเว็บได้ ฉะนั้นด้วย HTTP ซึ่งเป็น Protocol ที่ Firewall ยอมให้ผ่านและ XML ที่สามารถเข้าใจได้ง่ายในทุก Platform ทำให้เราสามารถแก้ปัญหาเรื่องการติดต่อสื่อสารระหว่างโปรแกรมบนเครือข่ายที่รันบนระบบปฏิบัติการที่ต่างกันได้

SOAP จะระบุวิธีในการเข้ารหัสส่วนหัว (Header Encoding) ของทั้ง HTTP และไฟล์ XML ไว้อย่างชัดเจนทั้งในส่วนของการติดต่อไปยังคอมพิวเตอร์เครื่องหนึ่งและส่งผ่านข้อมูลไปให้ รวมถึงระบุวิธีที่โปรแกรมซึ่งถูกเรียกนั้นจะส่งค่าคืนกลับมาด้วย

สิ่งที่ทำให้ SOAP เป็นที่จับตามองคือ มันใช้โพรโทคอล HTTP ในการสื่อสาร ฉะนั้นจึงไม่ค่อยมีปัญหาในการเข้าออกระบบเครือข่ายที่ Firewall เพราะว่า Firewall ส่วนใหญ่ต่างก็อนุญาตให้ข้อมูลที่ใช่ HTTP (พอร์ต 80) ผ่านได้อยู่แล้ว ดังที่กล่าวในข้างต้น นอกจากนี้มันยังถูกออกแบบให้สามารถใช้งานร่วมกับ SMTP และ MIME ได้ด้วย ฉะนั้นการทำงานระหว่างโปรแกรมต่างเครื่องต่างเครือข่ายก็ทำได้สะดวกขึ้น

SOAP กลายเป็นสิ่งที่มีความสำคัญสำหรับ Web Services อย่างรวดเร็ว เป็นโพรโทคอลที่ผู้จัดทำ Web Services เลือกใช้ที่จะส่ง message ระหว่าง Web Services SOAP เป็น Transport Protocol ที่มี XML เป็นพื้นฐานและใช้ HTTP เป็นโพรโทคอลร่วมในการส่งผ่านเครือข่าย เนื่องจากจุดประสงค์หลักของการใช้งานเว็บเซอร์วิส เราต้องการให้แอปพลิเคชัน มีการทำงานกับแอปพลิเคชันที่ทำงานอยู่ในเครื่องอื่น โดยผ่านทางเครือข่าย ซึ่งต่างก็มีปัญหาในด้านการนำมาใช้งานในแง่ของความเข้ากันได้ของการเรียกใช้งานข้ามเทคโนโลยี เนื่องจากเป็นเทคโนโลยีเฉพาะของแต่ละค่าย ซึ่งในแต่ละค่ายต่างก็มี Protocol ที่ใช้ในการติดต่อข้ามเครื่องต่างกัน เมื่อนำมาทำงานร่วมกัน จึงเกิดปัญหาในส่วนของ Firewalls และ Proxy Server ด้วยเนื่องจากโดยปกติ Server จะปิดการสื่อสารที่ไม่ใช่โพรโทคอล HTTP ออกไป เพื่อความปลอดภัยของระบบที่มีการติดต่อสื่อสารกับภายนอก ดังนั้นทางเลือกของการสื่อสารที่จะนำมาใช้ในการให้บริการเว็บเซอร์วิส ก็คือให้ทำงานอยู่บน โพรโทคอล HTTP ซึ่ง SOAP นอกจากจะทำงานบนโพรโทคอล HTTP แล้วยังเป็นมาตรฐานเปิดที่จะทำให้สามารถติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ที่มีความแตกต่างกันทั้งระบบปฏิบัติการ, เทคโนโลยี รวมไปถึงภาษาที่ใช้ในการพัฒนาด้วยก็ได้



รูปที่ 2.3 แสดง SOAP Message

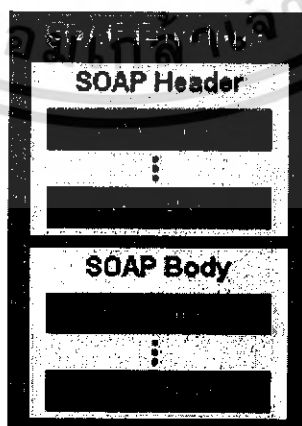
**โครงสร้างของ SOAP**

SOAP เป็นเรื่องที่เราเข้าใจได้ง่ายซึ่งมีโครงสร้างในรูปแบบ XML ส่วนของ SOAP จะมี envelope เป็น Root element ซึ่งหมายความว่า ใน 1 SOAP document จะมีส่วน envelope โดยส่วน envelope นี้จะมี header และ body ในส่วน header จะมี routing and context data ส่วน body จะมี message

SOAP envelope เป็นเสมือน ซองจดหมายที่ห่อหุ้มทุกอย่างๆ

SOAP header เป็น Optional จะมีหรือไม่มีก็ได้

SOAP body ประกอบด้วยเนื้อหาของ SOAP มีเมธอด และ การส่ง parameter



รูปที่ 2.5 แสดง SOAP Message format

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลองดูตัวอย่าง เป็น SOAP ฟังทาง requester เรียกไปหาฝั่ง provider ไปขอพยากรณ์อากาศ

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getWeather
      xmlns:ns1="urn:examples:weatherservice"
      SOAP-ENV:encodingStyle=" http://www.w3.org/2001/09/soap-encoding
      <zipcode xsi:type="xsd:string">10016</zipcode>
    </ns1:getWeather>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

มีการใช้ XML namespaces และ XML schema ในส่วนของ Body จะมี เมธอด getWeather และ รายการ parameter ในที่นี้ก็คือ zip code

ตัวอย่างถัดมาเป็น SOAP ที่ตอบกลับมาจากฝั่ง provider ของบริการ weather service

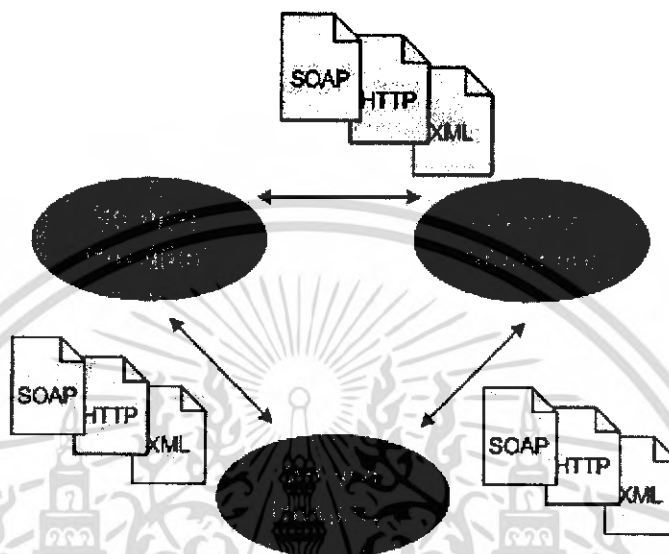
```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getWeatherResponse
      xmlns:ns1="urn:examples:weatherservice"
      SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/soap-encoding">
      <return xsi:type="xsd:int">65</return>
    </ns1:getWeatherResponse>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
</SOAP-ENV:Body>
```

```
</SOAP-ENV:Envelope>
```

ในส่วนนี้สังเกตจะเห็นการ Return ค่ากลับมาเป็น integer นั่นคือ ส่วน ของ อุดมภูมิ



รูปที่ 2.4 แสดงหลักการทำงานของ SOAP อย่างง่าย

### 2.3.3 Web Service Description Language (WSDL)

WSDL คือ XML-based language เป็นภาษาที่มีพื้นฐานมาจากภาษา XML มันถูกใช้เพื่อให้คำจำกัดความ หรือ นิยามของ Web service และทำอะไรเพื่อที่จะ Access Web service นั้น WSDL ถูกใช้เพื่อ พรรณนา หรือ บรรยาย data และ message ของ Web service ดังนั้น developer ต้องไปศึกษา WSDL Document ของ Web service เพื่อจะได้รู้ว่า method ใดใช้งานได้ และจะเรียกใช้งาน โดยผ่าน parameter อย่างไรให้ถูกต้อง หากเปรียบ Server ของเราเป็นเหมือนบ้าน Fire wall ก็จะเหมือนตัวปิดเปิดประตูและ web services ก็เหมือนขามที่คอยเฝ้าประตูใดประตูหนึ่งไว้ (ส่วนมากจะเป็นประตู 80 กับ 8080) เวลาใครต้องการทำอะไร ก็จะมาถามที่ขามว่าเค้าทำอะไรให้เราได้บ้าง ขามก็จะตอบกลับ ไปด้วย (WSDL) หลังจากรู้ว่าขามทำอะไรได้บ้างแล้ว แยกของเราก็จะส่งสนุ่ (SOAP) ไปให้ขาม ขามเห็นสนุ่ ก็จะตีความหมาย และนำเข้าไปในบ้าน อันนี้ขามจะไปทำอะไรเราก็ไม่รู้ แต่เราคาดหวังผลได้ว่าเค้าน่าจะทำตามที่เรา WSDL บอกไว้ หลังจากที่ยามทำเสร็จเค้าก็จะเอาผลลัพธ์มาแปลงเป็นสนุ่อีกก้อน แล้วส่งกลับมาให้ผู้ใช้งานประตูเดิม \*\*ขามในที่นี้ คือ Web application ครับ

WSDL สามารถแบ่งได้เป็น 2 กลุ่ม หนึ่งคือ กลุ่มที่เรียกว่า Abstract Definitions ซึ่ง

ประกอบไปด้วย types, message และ portType ซึ่งกลุ่มนี้เป็นส่วนที่เกี่ยวข้องกับการเชื่อมต่อค้ำเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมซึ่งทำให้เราติดต่อกับ code ของเราได้ และอีกกลุ่ม เรียกว่า Concrete Descriptions ประกอบไปด้วย binding และ service ซึ่งในกลุ่มนี้อธิบายเกี่ยวกับวิธีที่ส่วนเชื่อมต่อด้าน abstract จับคู่กับข้อมูล ซึ่งรายละเอียดของแต่ละส่วนย่อยเป็นดังตาราง

ตารางที่ 2.1 อธิบายส่วนย่อยของการเชื่อมต่อ

Element Name	Description
types	อธิบายชนิดข้อมูลที่เว็บเซอร์วิสใช้ เพื่อความเป็นกลาง WSDL ใช้ XML Schema Syntax ในการระบุชนิดข้อมูล
message	อธิบายเกี่ยวกับข้อมูล ( Data Elements ) ที่ส่งเข้าและออกจาก Operation เปรียบได้กับ Parameter ของ Function
portType	เป็นส่วนที่สำคัญที่สุดใน WSDL อธิบายว่ามี Operation ที่เว็บเซอร์วิสมี ให้บริการและ Messages ที่เกี่ยวข้อง เปรียบได้กับ Function Library หรือ Module หรือ Class ในการเขียนโปรแกรม
binding	อธิบายรูปแบบของ Message และรายละเอียดของ โปรโตคอล(Protocol Details) ในแต่ละ Port
service	เป็นส่วนที่บรรจุ <port> ที่เป็นส่วนที่ระบุถึง URL ของเว็บเซอร์วิสที่ให้บริการ โดยที่จะมีหนึ่งหรือมากกว่าหนึ่ง Port ก็ได้
operation	อธิบาย Method ที่ให้บริการเปรียบได้กับ Function ที่อยู่ใน Class โดยเว็บเซอร์วิสหนึ่งจะมี Method จำนวนกี่ Method ก็ได้

ตามทฤษฎีแล้ว ไฟล์เอกสาร WSDL แต่ละไฟล์ สามารถอธิบายคุณลักษณะของบริการ Web Services ได้มากกว่า 1 บริการ โดยแต่ละ Web Services จะมี port สื่อสารเฉพาะตัว ซึ่งบ่งบอกไว้ในเอกสาร WSDL อยู่แล้ว

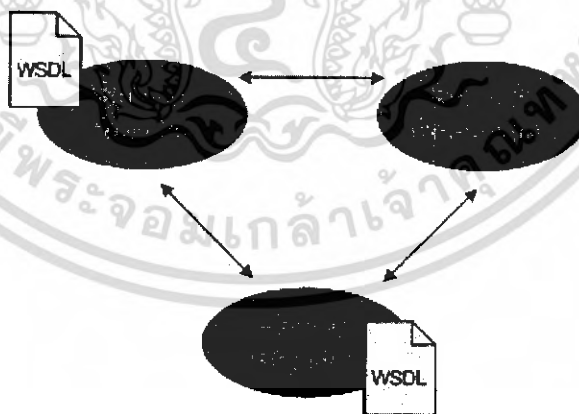
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<?xml version="1.0" ?>
- <definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
  ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:si="http://soapinterop.org/xsd"
  xmlns:tns="http://soapinterop.org/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://soapinterop.org/">
- <message name="helloWorldRequest">
  <part name="message" type="xsd:string" />
</message>
- <message name="helloWorldResponse">
  <part name="return" type="xsd:string" />
</message>
- <portType name="helloWorldPortType">
  - <operation name="helloWorld">
    <input message="tns:helloWorldRequest" />
    <output message="tns:helloWorldResponse" />
  </operation>
</portType>
- <binding name="helloWorldBinding" type="tns:helloWorldPortType">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
  - <operation name="helloWorld">
    <soap:operation soapAction="http://localhost/nusoap/server.php" style="rpc" />
  - <input>
    <soap:body use="encoded" namespace="http://soapinterop.org/"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  - <output>
    <soap:body use="encoded" namespace="http://soapinterop.org/"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
</binding>
- <service name="helloWorld">
  - <part name="helloWorldPort" binding="tns:helloWorldBinding">
    <soap:address location="http://localhost/nusoap/server.php" />
  </part>
</service>
</definitions>

```

รูปที่ 2.7 แสดง WSDL Example



รูปที่ 2.8 แสดงหลักการทำงานอย่างง่ายของ WSDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

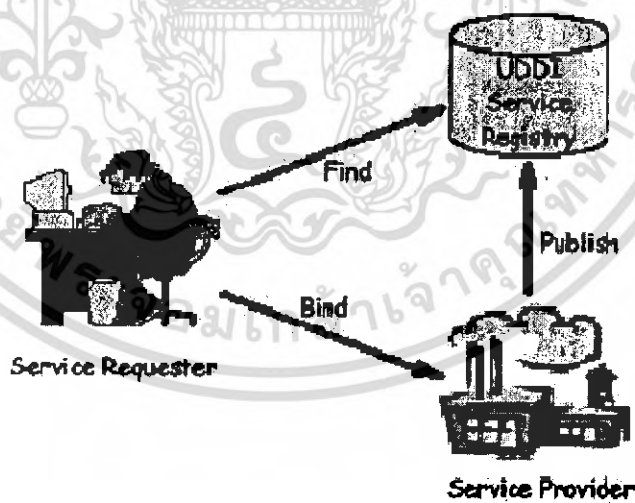
### 2.3.4 Universal Description, Discovery and Integration (UDDI)

UDDI เป็นมาตรฐานที่สร้างขึ้นมาเพื่อ การบรรยายพรรณนา ประกาศ และ ค้นหา Web Services ซึ่งเปิดให้บริการแล้ว โดยเป็นรายละเอียดสำหรับ ข้อมูลที่ลงทะเบียนไว้แล้ว และพร้อมจะเผยแพร่เปิดเผย เมื่อ Web Services ถูกพัฒนาขึ้นมาและ WSDL ถูกสร้างขึ้นมา ยังจำเป็นที่จะต้องมีส่วนหรือวิธีการที่จะนำเอา WSDL information มาให้ user ใช้งาน เมื่อ Web service ลงทะเบียน UDDI และถูกประกาศแล้ว user ก็จะมองหา และเรียนรู้ Web Services นั้นเพื่อนำไปใช้

UDDI เป็นมาตรฐานที่ให้ชุดพื้นฐาน APIs (Application Programming Interface) ของ SOAP ที่สามารถนำมาใช้ในการพัฒนาเป็นตัวแทนของผู้ให้บริการ (Service broker or Service Registry) เป็นที่เก็บรวบรวม Web Services ต่างๆ ในอินเทอร์เน็ต ไว้ในแหล่งเดียวกันเพื่อให้ผู้ใช้บริการสามารถค้นหาได้ง่าย หากเปรียบเทียบก็คงเหมือนสมุดหน้าเหลืองที่เราใช้ในการเปิดคูเบอร์โทรศัพท์

โดยสรุปแล้ว

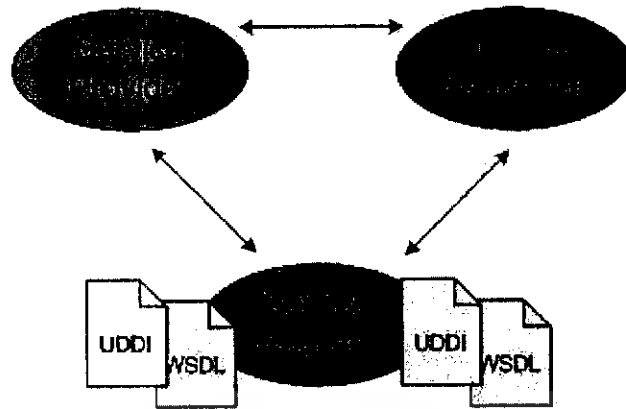
- UDDI ถูกสร้างขึ้นมาเพื่อเป็นมาตรฐานในการค้นหาบริการของ Web service
- UDDI เป็นเสมือน โครงสร้างพื้นฐานที่รองรับการค้นหาข้อมูลแบบใหม่ โดยใช้ตัวเลขหรือคำสามัญแทน Domain name แต่สำหรับ Domain name เองยังคงอยู่ไม่ได้หายไปไหน เพียงแต่ UDDI จะมาเป็นอีกมาตรฐานหนึ่งในการติดต่อกับ IP Address เหมือนที่เราเคยมี Domain name หรือ URL ในการทำให้ผู้ที่เข้าสู่เว็บไซต์รู้จักและจดจำได้โดยง่าย



รูปที่ 2.9 แสดงความสัมพันธ์ระหว่างผู้ขอบริการกับผู้ให้บริการ (Service Provider) และ UDDI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

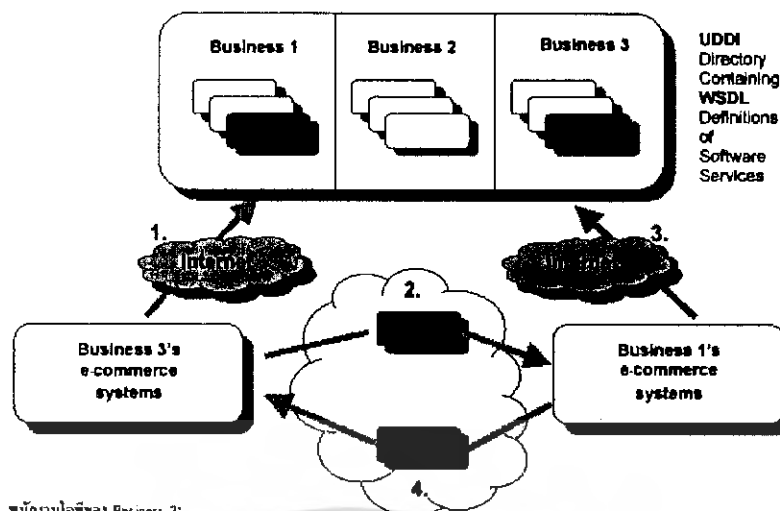


รูปที่ 2.10 แสดงหลักการทำงานอย่างง่ายของ UDDI

เราได้ทราบเกี่ยวกับ UDDI ว่าเป็นวิธีการมาตรฐานสำหรับจัดเก็บและรวบรวมบริการต่าง ๆ ที่ให้บริการในรูปของ Directory service และทราบเกี่ยวกับ WSDL ว่าเป็นมาตรฐานที่ใช้อธิบายคุณลักษณะของการเรียกใช้บริการของ Web Services และวิธีการติดต่อกับ Web Services แต่ UDDI จะเกิดขึ้นได้ต้องอาศัยผู้ให้บริการ (Service provider) จำนวนมาก เสนอบริการทางด้านซอฟต์แวร์หรือโปรแกรมของตนเอง แล้วเราจะต้องประกาศ (publish) บริการเหล่านี้ไปบนอินเทอร์เน็ต การให้บริการทางด้านซอฟต์แวร์ต่าง ๆ ของผู้ให้บริการ (Service providers) บนอินเทอร์เน็ตนี้รู้จักกันในนามของ "Web Service" กล่าวคือ Web Service คือ Software Component ที่สามารถนำมาสร้างเป็น Application สำหรับให้บริการการทำงาน ๆ หนึ่งให้แก่ผู้ร้องขอบนอินเทอร์เน็ต หรือสามารถที่จะนำ Web Service แต่ละตัวมาประกอบกันตามกระบวนการทางธุรกิจ เพื่อร่วมกันทำงานในลักษณะ Interoperability รวมกันเป็น "Web Services" ซึ่งสามารถจำลองภาพการทำงานของ Web Services ที่เกิดจากการทำงานร่วมกันของ SOAP, UDDI และ WSDL ดังแสดงในรูป

72925

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

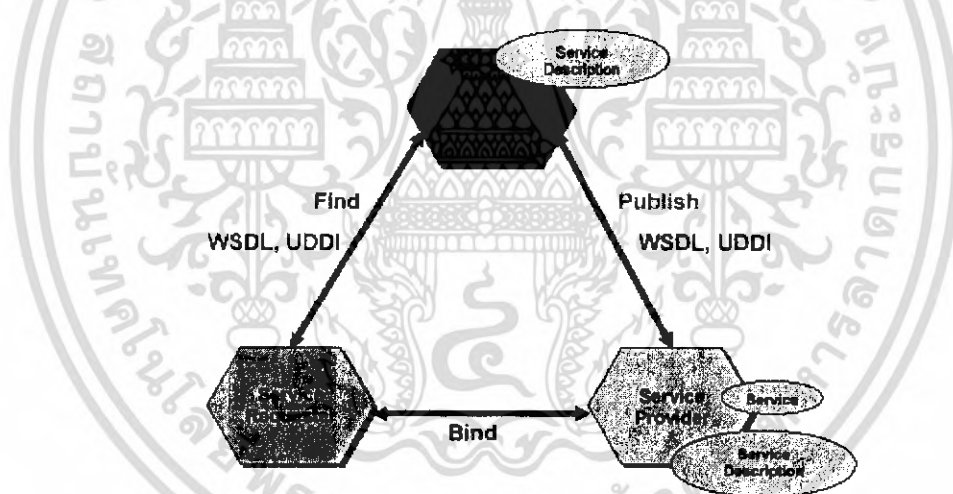


พนักงานอยู่ที่ของ Business 3:

1. ค้นหาการให้บริการใน WSDL ผ่านอินเทอร์เน็ตของ Business 1
2. ติดต่อผ่านอินเทอร์เน็ตเพื่อขอแมพพิพจน์ที่ให้บริการ (ตัว SOAP) ผ่านพหุอันเทอมนี่
- 3,4. Business 1 รวมหรือบูรณาการกับ Business 3 ตัวอีกตัวหนึ่ง

รูปที่ 2.11 แสดง Web Services with SOAP, UDDI and WSDL

## 2.4 แบบจำลองการใช้เว็บเซอร์วิส



รูปที่ 2.12 แสดง Web Service Model

2.4.1) Service Provider: ผู้ให้บริการที่ทำการประกาศบริการ (publish) ไปยัง Directory ที่เกี่ยวข้องกับบริการ

2.4.2) Service Requestor: ผู้ขอใช้บริการจะทำการค้นหา (find) บริการที่ต้องการ และเมื่อพบก็จะทำการเรียกใช้ (bind) ไปยังผู้ให้บริการนั้น

2.4.3) Service Registry: เปรียบเสมือนตัวแทนของผู้ให้บริการหรือที่อาจเรียกว่า “Directory of Service”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4) UDDI เป็นวิธีการมาตรฐานสำหรับจัดเก็บและรวบรวมบริการต่าง ๆ ที่ให้บริการในรูปของ Directory service แต่ UDDI จะเกิดขึ้นได้ต้องอาศัยผู้ให้บริการ (Service provider) จำนวนมาก เสนอบริการทางด้านซอฟต์แวร์หรือโปรแกรมของตนเอง แล้วเราจะต้องประกาศ (publish) บริการเหล่านี้ไปบน อินเทอร์เน็ต

2.4.5) WSDL เป็นมาตรฐานที่ใช้อธิบายคุณลักษณะของการเรียกใช้บริการของ Web Services และวิธีการติดต่อกับ Web Services

## 2.5 คุณลักษณะของ Web Services

2.5.1) รายละเอียดในการสร้างและพัฒนาเว็บเซอร์วิสจะถูกซ่อนไว้เพื่อไม่ให้มองเห็นได้จากภายนอก ผู้เรียกใช้เว็บเซอร์วิสจะรู้จักเพียง Interface ที่ผู้ให้บริการประกาศไว้เท่านั้น

2.5.2) ซอฟต์แวร์ที่ทำงานบนเว็บเซอร์วิสสามารถนำมาแก้ไขรายละเอียดภายในได้โดยไม่ต้องส่งผลกระทบต่อผู้ใช้ ทำให้การออกแบบซอฟต์แวร์เป็นไปได้ง่าย และผู้ใช้ที่ปลายทางไม่จำเป็นต้องโหลดซอฟต์แวร์ไว้มากเกินความจำเป็น

2.5.3) โปรแกรมที่เรียกใช้เว็บเซอร์วิสจะรับรู้ได้เองว่าเซอร์วิสที่กำลังจะเรียกใช้นั้นมีการกำหนดพารามิเตอร์อินพุตและเอาต์พุตอย่างไร ผ่านทาง WSDL file

2.5.4) อยู่บนพื้นฐานอยู่บนภาษา XML

2.5.5) สามารถแก้ไขได้ตลอดเวลา จึงสามารถอัปเดตได้อย่างรวดเร็ว และไม่จำเป็นต้องเรียก Application ที่แก้ไขแล้วมารันใหม่

เทคโนโลยีในการกระจายข่าวสารข้อมูลทาง Internet ในปัจจุบันก็คือ Web page แต่การที่มันมีความสามารถที่จะทำงานได้ด้วยการรวมภาษา เช่นภาษา VBScript, Java Script หรือ ASP, PHP, JSP นั้นทำให้ Web page มีลักษณะคล้าย Application จึงถูกเรียกรวมกันว่า Web Application

## 2.6 ประเภทของ Web Services

2.6.1) เว็บเซอร์วิสอย่างง่าย (Simple Web Services) ทำหน้าที่เพียงส่งและตอบรับคำขอ บริการต่าง ๆ แต่ไม่สามารถประมวลผลข้อมูลทางธุรกรรมได้ด้วยตนเองและไม่มียระบบรักษาความปลอดภัย ลักษณะเรียลไทม์ที่ใช้ UDDI, WSDL SOAP เพื่อการค้นหา และสั่งการ ยังไม่มีขีดความสามารถเพียงพอที่จะรองรับ การสื่อสาร และการเชื่อมต่อการใช้งานแอปพลิเคชันระหว่างธุรกิจกับธุรกิจที่มีมูลค่าสูงๆ ได้

2.6.2) เว็บเซอร์วิสที่มีโครงสร้างซับซ้อน (Complex Web Services) ใช้ข้อได้เปรียบของโปรโตคอลระดับสูง เพื่อรองรับการสื่อสารข้อมูลระหว่างระบบ ที่ซับซ้อนมากขึ้น ธุรกรรมที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องใช้เวลาทำรายการนาน ซึ่งเกี่ยวข้องกับมาตรการรักษาความปลอดภัยที่ทันสมัย เช่น ระบบยืนยันการส่ง-รับข้อมูล (non-repudiation) และลายเซ็นอิเล็กทรอนิกส์ (Digital signatures)

## 2.7 ประโยชน์ของ Web Services

2.7.1) พันธมิตรทางการค้าสามารถเกิดขึ้นได้ตลอดเวลา โดยการค้นหาจาก UDDI

2.7.2) การดำเนินธุรกิจการค้าและบริการสามารถเป็นไปได้แบบอัตโนมัติในระดับของ Application to Application (A2A) โดยการแลกเปลี่ยนข้อมูลผ่าน Web Services

2.7.3) ง่ายต่อการนำไปใช้งานเนื่องจากในปัจจุบันมีเครื่องมือมากมายที่ช่วยเหลือในการพัฒนา Web Services

2.7.4) ลดต้นทุนในการพัฒนาระบบบางอย่างที่ไม่จำเป็น โดยขอบริการจาก Web Services ของพันธมิตรทางการค้า

## 2.8 การเรียกใช้ Web Services

1. ใช้ Web service client API เช่น WSIF, WSE เป็นต้น ก็คือเราเขียน application (ด้วย Java หรือ .NET) ของเราไปตามปกติเมื่อถึงเวลาต้องเรียกใช้ WS ก็เรียกใช้ API แทนเหมือนกับการเขียนโปรแกรมปกติทั่วไป วิธีนี้เหมาะกับ end user client ที่เป็นผู้ใช้ปลายทางจริงๆ เพราะไม่ค่อยมีการเปลี่ยนแปลงบ่อย และมักเรียกเพียงไม่กี่ service

อย่างไรก็ตามวิธีนี้ไม่เหมาะกับการสร้าง Composite WS เพราะเรามักใช้ Composite WS เพื่อสร้าง Business process (business process คือ บริการที่เห็นหรือสัมผัสได้โดยตรงจากผู้หรือลูกค้า และให้ผลตอบแทนกับองค์กร นั่นก็คือ service นอกสุดที่ให้บริการลูกค้าโดยตรงนั่นเอง) ซึ่งมี Business logic ซึ่งอาจเปลี่ยนแปลงบ่อยๆ นอกจากนี้ Business goal ของการสร้าง Application ด้วย Web Services ก็เพื่อความคล่องตัว (agility) ปรับเปลี่ยนง่าย จะได้สอดคล้องกับสภาพธุรกิจปัจจุบันที่มีการแข่งขันสูง ดังนั้นการผูกแต่ละ Web Services ไว้ด้วยการเขียนโปรแกรมแบบเก่านั้นจึงไม่เหมาะกับการสร้าง Business process ด้วย Web Services ตามหลักการของ SOA

2. ใช้ Web Services ตัวกลาง (mediator) มาเรียกใช้ Sub-WS นั่นคือใช้ BPEL (Business Process Execution Language) นั่นเอง

## บทที่ 3

### ความปลอดภัย

#### 3.1 นิยามความปลอดภัยในคอมพิวเตอร์

ในปัจจุบันระบบคอมพิวเตอร์ได้ถูกคุกคามมากขึ้นทั้งจากไวรัสคอมพิวเตอร์หรือจากผู้ไม่ประสงค์ดี ซึ่งความมั่นคงปลอดภัยคอมพิวเตอร์ช่วยปกป้องเครื่องคอมพิวเตอร์รวมถึงอุปกรณ์ต่างๆที่เกี่ยวข้อง และที่สำคัญยังสามารถช่วยปกป้องข้อมูลที่ได้จัดเก็บไว้ภายในระบบหรือใช้ในความหมายความปลอดภัยทางข้อมูลสารสนเทศ

##### 3.1.1 จุดประสงค์หลักของความปลอดภัยทางข้อมูล

- การรักษาความลับ (Confidentiality) คือการรับรองว่าจะมีการเก็บข้อมูลไว้เป็นความลับ และผู้สิทธิเท่านั้นจึงจะเข้าถึงข้อมูลนั้นได้
- การรักษาความสมบูรณ์ (Integrity) คือการรับรองว่าข้อมูลจะไม่ถูกเปลี่ยนแปลงหรือทำลายไม่ว่าจะเป็นโดย อุบัติเหตุหรือโดยเจตนา
- การห้ามปฏิเสธความรับผิดชอบ (Non-Repudiation) คือวิธีการสื่อสารซึ่งผู้ส่งข้อมูลได้รับหลักฐานว่าได้มีการส่งข้อมูลแล้วและผู้รับก็ได้รับการยืนยันว่าผู้ส่งเป็นใคร ดังนั้นทั้งผู้ส่งและผู้รับจะไม่สามารถปฏิเสธได้ว่าไม่มีความเกี่ยวข้องกับข้อมูลดังกล่าวในภายหลัง
- การกำหนดสิทธิ์ (Authorization) คือขั้นตอนในการอนุญาตให้แต่ละบุคคลสามารถเข้าถึงข้อมูลหรือระบบใดได้บ้าง ก่อนอื่นต้องทราบก่อนว่าบุคคลที่กล่าวอ้างนั้นคือใครตามขั้นตอนการพิสูจน์ตัวตนและต้องให้แน่ใจด้วยการพิสูจน์ตัวตนนั้นถูกต้อง
- การตรวจสอบ (Audit) การตรวจสอบ คือการตรวจสอบหลักฐานทางอิเล็กทรอนิกส์ ซึ่งสามารถใช้ในการติดตามการดำเนินการเพื่อตรวจสอบความถูกต้องและแม่นยำ ตัวอย่างเช่นการตรวจสอบบัญชีผู้ใช้ โดยผู้ตรวจบัญชี ซึ่งการตรวจสอบความถูกต้องของการดำเนินการเพื่อให้แน่ใจว่าหลักฐานทางอิเล็กทรอนิกส์นั้นได้ถูกสร้างและสั่งให้ทำงาน โดยบุคคลที่ได้รับอนุญาต และในการเชื่อมต่อเหตุการณ์เข้ากับบุคคลจะต้องทำการตรวจสอบหลักฐานของบุคคลนั้นด้วย ซึ่งถือเป็นหลักการพื้นฐานของขั้นตอนการทำงานของการพิสูจน์ตัวตนด้วย

### 3.2 รูปแบบการรักษาความปลอดภัยในเว็บเซิร์ฟเวอร์

การรักษาความปลอดภัยในเว็บเซิร์ฟเวอร์คือการรักษาความปลอดภัยของเอกสาร โชนิปที่ส่งระหว่างกัน โดยการแบ่ง โมเดลของการรักษาความปลอดภัยในเว็บเซิร์ฟเวอร์ตามสถาปัตยกรรม เซิร์ฟเวอร์โอเรียลเต็ลจะแบ่งออกเป็น 2 ประเภทคือ

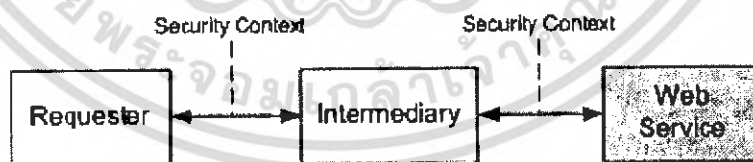
#### 3.2.1 พอยท์-ทู-พอยท์ (Point-to-Point)

การรักษาความปลอดภัยแบบ พอยท์-ทู-พอยท์ คือการรักษาความปลอดภัยแบบระหว่างหน่วยที่ละหน่วย กล่าวคือซีเคียวริตี้คอนเทกซ์(Security context) ระหว่างรีเคเวสเตอร์และอินเทอร์มีเดียร์กับซีเคียวริตี้คอนเทกซ์ระหว่างอินเทอร์มีเดียร์และเว็บเซิร์ฟเวอร์เป็นคนละตัวกัน การรักษาความปลอดภัยในโมเดลนี้ได้แก่การใช้ Secure Socket Layer (SSL) เป็นต้น

การใช้ SSL ในการรักษาความปลอดภัยเป็นการรักษาความปลอดภัยในชั้นทรานสปอร์ตเลเยอร์ (Transport Layer) หรือที่เรียกว่าทรานสปอร์ตเลเยอร์ซีเคียวริตี้ (Transport Layer Security) หรือ TLS

ข้อดีของ SSL/TLS คือมีประสิทธิภาพในการประมวลผลสูงกว่าการรักษาความปลอดภัยในแบบอื่นเนื่องจากไม่ต้องมีส่วนเพิ่มเติมในเอกสาร โชนิปและเป็นเทคโนโลยีที่เป็นเทคโนโลยีพื้นฐานในปัจจุบันอยู่แล้ว

การรักษาความปลอดภัยแบบพอยท์-ทู-พอยท์ ใช้ได้ดีในระบบที่ไม่มีตัวกลาง ในระบบที่มีความซับซ้อนจำเป็นที่จะต้องมีส่วนกลางเข้ามาเกี่ยวข้องอย่างหลีกเลี่ยงไม่ได้เช่น อาจมีการใช้พรอกซี เซอเวอร์ เร้าเตอร์ ในระบบด้วย ถ้าเราใช้การรักษาความปลอดภัยแบบพอยท์-ทู-พอยท์ จะทำให้เกิดความยุ่งยากในการติดตั้งระบบและขาดความยืดหยุ่นต่อการปรับปรุงเปลี่ยนแปลงระบบในภายหลัง

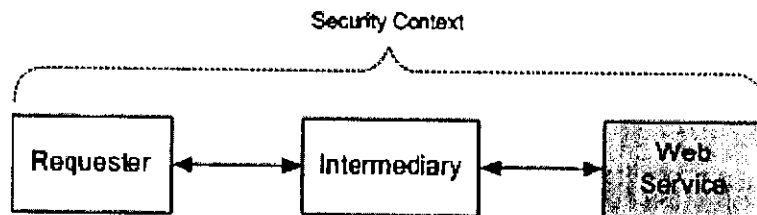


รูปที่ 3.1 แสดงการรักษาความปลอดภัยแบบ พอยท์-ทู-พอยท์

#### 3.2.2 เอ็น-ทู-เอ็น (End-to-End)

การรักษาความปลอดภัยแบบเอ็น-ทู-เอ็น คือการรักษาความปลอดภัยจากแอฟพลิเคชันที่ทำการเรียกเว็บเซิร์ฟเวอร์ถึงตัวเว็บเซิร์ฟเวอร์ที่เป็นตัวสุดท้าย จากรูปการรักษาความปลอดภัยในแบบเอ็น-ทู-เอ็น จากรีเคเวสเตอร์ถึง โพรไวเดอร์จะมีซีเคียวริตี้คอนเทกซ์เดียวกัน ถึงแม้ว่าจะมีส่วนกลางอยู่ก็ตาม เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

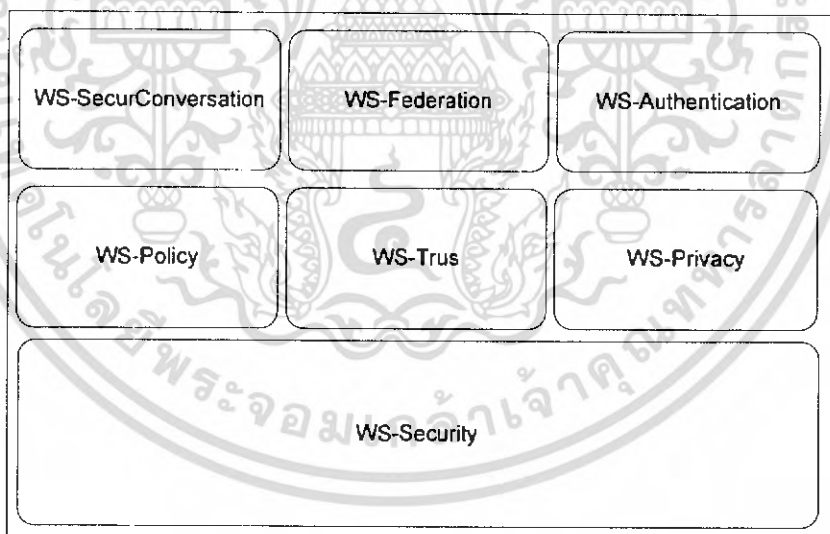
การรักษาความปลอดภัยในแบบนี้จะทำให้มีความสะดวกในการทำงานจริงคือไม่ต้องคำนึงถึงตัวกลางว่าจะมีกี่หน่วย การรักษาความปลอดภัยในแบบนี้จะกำกระทำในใช้เมสเสจ ซึ่งมีข้อกำหนดต่างของเว็บเซอร์วิสเป็นตัวกำหนดว่าจะมีการรักษาความปลอดภัยอย่างไรซึ่งจะพูดในหัวข้อถัดไป



รูปที่ 3.2 แสดงการรักษาความปลอดภัยแบบ เอ็น-ทู-เอ็น

### 3.3 ข้อกำหนดด้านการรักษาความปลอดภัยในเว็บเซอร์วิส

การรักษาความปลอดภัยจะทำตามข้อกำหนดเพื่อก่อให้เกิดมาตรฐานในการติดต่อระหว่างที่ทำงานร่วมกัน ข้อกำหนดแต่ละอันจะถูกสร้างขึ้นมาจากจุดประสงค์ละการใช้ที่ต่างกัน การเลือกปฏิบัตินั้นขึ้นอยู่กับความต้องการของแอปพลิเคชัน เพราะการปฏิบัติบางอย่างอาจทำให้ประสิทธิภาพการทำงานของแอปพลิเคชันลดลง นอกจากนี้ยังต้องคำนึงถึงการรองรับของแพลตฟอร์มที่ใช้ในการสร้างแอปพลิเคชันด้วย



รูปที่ 3.3 ข้อกำหนดด้านการรักษาความปลอดภัยในเว็บเซอร์วิส

จากรูปแสดงกลุ่มของข้อกำหนดด้านความปลอดภัยของเว็บเซอร์วิสในชั้นล่างสุด WS-Security จะเป็นข้อกำหนดที่เป็นพื้นฐานในข้อกำหนดอื่นๆต่อมา ในชั้นถัดมา WS-Policy จะเป็นข้อกำหนดที่มีเพื่อแสดงนโยบายในการทำงานเอ็นท์พอยท์แต่ละเอ็นท์พอยท์ WS-Trust เป็นข้อกำหนดถึงtrust model WS-Privacy จะกำหนด privacy model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างปฏิบัติตามข้อกำหนดดังกล่าวมาเป็นพื้นฐานของการรักษาความปลอดภัยแบบ  
ร่วมกันระหว่างองค์กรคือ WS-Federation, WS-SecureConversation และ WS-Authentication

เราสามารถแบ่งข้อกำหนดด้านการรักษาความปลอดภัยออกเป็นสองประเภทคือข้อกำหนด  
พื้นฐานและข้อกำหนดที่นำข้อกำหนดที่นำข้อกำหนดพื้นฐานมาใช้

ข้อกำหนดพื้นฐาน

WS-Security

WS-Policy

WS-Trust

WS-Privacy

ข้อกำหนดที่นำข้อกำหนดอื่นมาใช้

WS-SecureConversation

WS-Federation

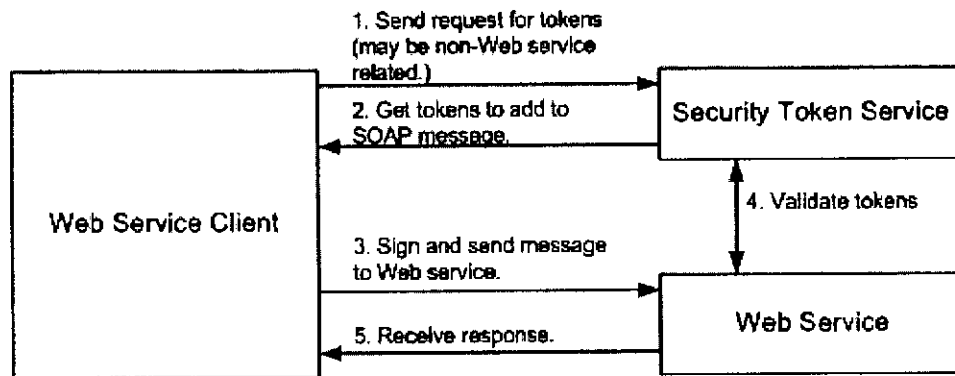
WS-Authentication

### 3.4 WS-Security

ข้อกำหนดนี้เป็นข้อกำหนดที่อธิบายส่วนเพิ่มเติมของSOAP เมสเสจเพื่อใช้ในการสร้าง  
ความปลอดภัยให้กับตัวเมสเสจ WS-Security ทำให้ SOAPเมสเสจที่ส่งระหว่างเซิร์ฟเวอร์เป็นไปตาม  
ความต้องการในการรักษาความปลอดภัย 3 ข้อคือ

1. การปกปิดความลับของข้อมูล ข้อกำหนดกำหนดให้ใช้ XML-Encryption ในการ  
เข้ารหัสSOAP การเข้ารหัสโดยใช้ XML-Encryption จะเป็นการเข้ารหัส ตามXML  
infoset ซึ่งไม่ใช่การเข้ารหัสเมสเสจทั้งหมด การใช้วิธีนี้ทำให้เกิดการรักษาความ  
ปลอดภัยแบบเอ็น-ทู-เอ็น
2. การยืนยันความถูกต้องของข้อมูล การรักษาความถูกต้องของข้อมูลใช้ XML-Signature  
ในการตรวจสอบเมสเสจ โดย WS- Security จะกำหนดวิธีในการใส่ตัวรายเซ็นไว้ใน  
SOAPเมสเสจ
3. การยืนยันตัวตน WS-Security กำหนดถึงวิธีในการใส่ซีเคียวริตี้โทเค้นไว้ใน  
SOAP เพื่อใช้ในการยืนยันตัวตน ซึ่งซีเคียวริตี้โทเค้นนั้นสามารถใช้ได้หลายชนิด  
เช่น Username Token, PKI(X.509), Kerberos เป็นต้น

WS-Security เป็นข้อกำหนดที่นำมามาตรฐานการรักษาความปลอดภัยที่อยู่แล้วในปัจจุบันมา  
ใช้โดยที่ไม่ได้กำหนดขึ้นมาใหม่รูปที่ 3.4 แสดงการไหลของเมสเสจใน WS-Security



รูปที่ 3.4 แสดง Flow ของ WS-Security

จากรูปแสดงขั้นตอนของการรักษาความปลอดภัยในเว็บเซอร์วิสโดยใช้ WS-Security ประกอบไปด้วยสามส่วนหลักคือ เว็บเซอร์วิสไคลเอ็นท์คือ รีแควสเตอร์ที่เป็นผู้เรียกใช้บริการจากเว็บเซอร์วิส เว็บเซอร์วิสหรือ โพรไวเดอร์เป็นผู้ให้บริการ ซึ่งเคียวรีตีโทเคนเซอร์วิสคือส่วนที่ให้บริการโทเคนกับผู้เรียกใช้บริการ

ในขั้นที่แรกฝั่งไคลเอ็นต์ต้องร้องขอโทเคนจากซีเคียวรีตีโทเคนเซอร์วิสหลังจากนั้นซีเคียวรีตีโทเคนเซอร์วิสจะให้โทเคนแก่ไคลเอ็นท์ในขั้นที่ 2 ซึ่งการกระทำใน 2 ขั้นตอนนี้อาจจะไม่ใช่ว่าขั้นตอนที่เป็นเว็บเซอร์วิสก็ได้เช่น การร้องขอซีเคียวรีตีโทเคนอาจใช้วิธีการทางอีเมลและส่งซีเคียวรีตีโทเคนมาให้ทางอีเมลเป็นต้น ซีเคียวรีตีโทเคนอาจเป็นได้ทั้ง UsernameToken, X.509 certificate, หรือ Kerberos ticket

ต่อมาการให้บริการฝั่งรีแควสเตอร์จะต้องนำโทเคนที่ได้รวมเข้ากับ SOAP แล้วทำการส่งมาที่เว็บเซอร์วิสในขั้นที่ 3 เมื่อเว็บเซอร์วิสได้รับ SOAP จะทำโทเคนที่ไปตรวจสอบ ในขั้นที่ 4 ถ้าการยืนยันตัวบุคคลสำเร็จเว็บเซอร์วิสจะรีเทิร์นค่ากลับมาที่ไคลเอ็นท์ในขั้นที่ 5 ต่อไป

#### 3.4.1 Authentication

การยืนยันตัวบุคคลใน WS-Security ทำได้โดยตรวจสอบจาก credential ที่แนบมากับ SOAP Message WS-Security ไม่ได้กำหนดตายตัวว่ามีวิธีหรือชนิดของ Credential ที่ใช้ในการยืนยันตัวบุคคลทั้งหมดกี่วิธี แต่ใน WS-Security กำหนดวิธีในการยืนยันตัวบุคคลไว้ 3 ชนิดคือ

1. Username/Password
2. Public Key Infrastructure (X.509)
3. Kerberos

### 3.4.2 Signing

ในการส่งข้อมูลเมื่อมีการ Sign เมสเสจที่จะทำการส่ง โอกาสที่เมสเสจจะถูกแอบอย่างด้วยผู้อื่นนั้นเป็นไปได้น้อยมาก การใช้ลายเซ็นใน SOAP ฝ่ายรับสามารถตรวจสอบได้ว่าเมสเสจไม่ได้ถูกแก้ไขในระหว่างการส่ง WS-Security กำหนดให้ใช้ XML-Signature ในการ sign เมสเสจก่อนทำการส่ง วิธีในการยืนยันตัวตนทั้ง 3 วิธีที่กำหนดไว้ใน WS-Security ได้กำหนดวิธีในการ Sign ไว้ภายในตัวเองด้วย

Username/Password จะใช้ Password ในการ sign เมสเสจ, X.509 ใช้ Private Key ในการ sign เมสเสจและ Kerberos ใช้ session key ในการ sign ข้อมูล

### 3.4.3 Encryption

การปกปิดความลับของข้อมูลจะใช้ XML-Encryption ในการเข้ารหัสข้อมูล การเข้ารหัสข้อมูลจะต้องเลือกว่าจะใช้ key แบบใดระหว่าง symmetric key หรือ Asymmetric key WS-Security ไม่ได้กำหนดถึงวิธีในการที่เราจะเข้ารหัสข้อมูล เราสามารถเลือกใช้ได้เองเช่น triple DES เป็นต้น

```

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
  <soap:Header
    xmlns:wssc="http://schemas.xmlsoap.org/ws/2002/07/secext"
    xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility">
    <wsu:Timestamp>
      <wru:Created
        wsu:Id="Id-2beeb885-16a4-4b65-b14c-0cfe6ad26800"
        >2002-08-22T00:26:15Z</wsu:Created>
      <wsu:Expires
        wsu:Id="Id-10c46143-cb53-4a8a-9e83-ef374e40aa54"
        >2002-08-22T00:31:15Z</wsu:Expires>
    </wsu:Timestamp>
    <wssc:Security soap:mustUnderstand="1" >
      <xenc:ReferenceList>
        <xenc:DataReference
          URI="#{EncryptedContent-f6f50b24-3458-41d3-aac4-390f476f2e51}" />
        </xenc:ReferenceList>
        <xenc:ReferenceList>
          <xenc:DataReference
            URI="#{EncryptedContent-666b184a-a388-46cc-a9e2-06582b9d43b6}" />
          </xenc:ReferenceList>
        </wssc:Security>
      </soap:Header>
      <soap:Body>
        <xenc:EncryptedData
          Id="EncryptedContent-f6f50b24-3458-41d3-aac4-390f476f2e51"
          Type="http://www.w3.org/2001/04/xmlenc#Content">
          <xenc:EncryptionMethod Algorithm=
            "http://www.w3.org/2001/04/xmlenc#tripledes-cbc" />
          <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
            <KeyName>Symmetric Key</KeyName>
          </KeyInfo>
          <xenc:CipherData>
            <xenc:CipherValue
              >Im35XQcBV5UiT... Y7RV2QgnPpZYMg</xenc:CipherValue>
            </xenc:CipherData>
          </xenc:EncryptedData>
        </soap:Body>
      </soap:Envelope>

```

รูปที่ 3.5 แสดง SOAP ที่เข้ารหัสแล้ว

### 3.5 ใบรับรองอิเล็กทรอนิกส์ (Certificate)

ปัจจุบันการติดต่อสื่อสารมักทำผ่านระบบเครือข่ายอินเทอร์เน็ต โดยที่บุคคลหรือองค์กรที่ติดต่อด้วยนั้น อาจจะไม่เคยมีความสัมพันธ์หรือรู้จักกันมาก่อน ก่อให้เกิดความไม่มั่นใจว่าบุคคลหรือองค์กรที่ติดต่อด้วยคือใคร มีตัวตนจริงหรือไม่ ฉะนั้น จึงจำเป็นที่จะต้องมีการยืนยันตัวตนบุคคลสำหรับโลกอิเล็กทรอนิกส์ สิ่งที่ใช้ในการยืนยันก็คือ ใบรับรองอิเล็กทรอนิกส์ โดยใบรับรองอิเล็กทรอนิกส์ดังกล่าวจะถูกออกโดยหน่วยงานที่เรียกว่า ผู้ให้บริการออกใบรับรอง ซึ่งข้อมูลในใบรับรองอิเล็กทรอนิกส์นั้น บ่งบอกถึงตัวตนในโลกแห่งอิเล็กทรอนิกส์ ดังนั้น การที่จะนำใบรับรองอิเล็กทรอนิกส์ไปใช้งาน ผู้ใช้ควรที่จะมีความรู้ความเข้าใจเกี่ยวกับใบรับรองอิเล็กทรอนิกส์ บทความนี้จะนำเสนอรูปแบบของใบรับรองอิเล็กทรอนิกส์ และรายการข้อมูล ซึ่งเป็นส่วนประกอบในใบรับรองอิเล็กทรอนิกส์ ตลอดจนความหมายของรายการข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5.1 รูปแบบของใบรับรองอิเล็กทรอนิกส์ (Certificate Profile)

รูปแบบของใบรับรองอิเล็กทรอนิกส์ อิงตามมาตรฐาน X.509 Certificate ซึ่งเป็นมาตรฐานของใบรับรองอิเล็กทรอนิกส์ที่กำหนดโดย ITU-T X.509 International Standard ที่บอกถึงรูปแบบของใบรับรองฯ โดยประกอบไปด้วยข้อมูลหลักๆ 2 ส่วน คือ ข้อมูลพื้นฐานของใบรับรองอิเล็กทรอนิกส์ (Basic Certificate Fields) และข้อมูลเพิ่มเติมของใบรับรองอิเล็กทรอนิกส์ (Certificate Extension) ดังรูป

Certificate format version		
Certificate serial number		
Signature algorithm identifier for CA		
Issuer X.500 name		
Validity period		
Subject X.500 Name		
Subject public key information		
Type	Criticality	Value
Type	Criticality	Value
CA Signature		

Extensions

รูปที่ 3.6 ตัวอย่างใบรับรองอิเล็กทรอนิกส์

### 3.5.2 ข้อมูลพื้นฐานของใบรับรองอิเล็กทรอนิกส์ (Basic Certificate Fields)

ใบรับรองอิเล็กทรอนิกส์ประกอบไปด้วยข้อมูลพื้นฐาน ดังนี้

#### Version

เวอร์ชันของใบรับรองอิเล็กทรอนิกส์ที่ใช้ตามมาตรฐาน X.509 Certificate ซึ่งมีทั้งหมดด้วยกัน 3 เวอร์ชัน คือเวอร์ชัน 1, 2 และ 3 โดยที่เวอร์ชัน 3 จะรองรับการใช้งานใบรับรองอิเล็กทรอนิกส์ที่มีข้อมูลเพิ่มเติม (Certificate Extension)

#### Serial Number

หมายเลขที่ผู้ให้บริการออกใบรับรอง ทำการกำหนดให้กับใบรับรองอิเล็กทรอนิกส์แต่ละใบ เพื่อป้องกันการซ้ำกันของใบรับรองฯ ในกรณีที่ใบรับรองอิเล็กทรอนิกส์ถูกเพิกถอน หมายเลขดังกล่าวจะปรากฏอยู่ในรายการเพิกถอนใบรับรองฯ โดยที่รายการเพิกถอนใบรับรองฯ นี้จะถูกรับรองโดยผู้ให้บริการออกใบรับรอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Signature

อัลกอริทึมที่ผู้ให้บริการออกใบรับรองฯ ใช้ในการลงลายมือชื่อดิจิทัลและใช้ในการย่อยข้อมูล(Hash Function) เพื่อทำการรับรองใบรับรองอิเล็กทรอนิกส์ เช่น

Sha1WithRSAEncryption,

Md5withRSAEncryption เป็นต้น

### Issuer

ชื่อของผู้ให้บริการออกใบรับรอง ที่ทำการรับรองและออกใบรับรองอิเล็กทรอนิกส์ ซึ่งจะใช้รูปแบบของ Distinguished Name (DN) ตามมาตรฐาน X.500 เช่น c=TH, o=GOV เป็นตัวอย่างของ DN ซึ่งหมายถึงผู้ให้บริการออกใบรับรองที่ทำการออกใบรับรองอิเล็กทรอนิกส์ให้กับหน่วยงานที่เป็นภาครัฐในประเทศไทย

### Validity

ช่วงเวลาที่สามารถใช้งานใบรับรองอิเล็กทรอนิกส์ โดยระบุถึงวัน-เวลาเริ่มต้นและสิ้นสุดของการใช้งานใบรับรองฯ ซึ่งมี 2 มาตรฐานในการกำหนดวัน-เวลาดังกล่าว คือ

Universal time (UTCTime)

ซึ่งมีรูปแบบของวัน-เวลา ดังนี้ YYMMDDHHMMSSZ

Generalized time (GeneralizedTime)

ซึ่งมีรูปแบบของวัน-เวลา ดังนี้YYYYMMDDHHMMSSZ โดยปีคริสต์ศักราชที่น้อยกว่าปีคริสต์ศักราช 2049 ใช้รูปแบบของวัน-เวลาเป็น UTCTime ส่วนปีคริสต์ศักราช 2050 เป็นต้นไป ใช้รูปแบบของวัน-เวลาเป็น GeneralizedTime

### Subject

ชื่อของผู้ที่เป็นเจ้าของใบรับรองอิเล็กทรอนิกส์ ซึ่งจะใช้รูปแบบของ Distinguished Name (DN) ตามมาตรฐาน X.500 เช่น c=TH, o=GOV, cn=Nuntana Podjananuntakul เป็นตัวอย่างของ DN ซึ่งบ่งบอกถึงชื่อของผู้ที่เป็นเจ้าของใบรับรองฯ (นันทนา พจนานันทกุล) ที่อยู่ภายใต้หน่วยงานภาครัฐในประเทศไทย

### Subject Public Key Info

กุญแจสาธารณะของผู้ที่เป็นเจ้าของใบรับรองอิเล็กทรอนิกส์ และอัลกอริทึมที่ใช้ในการสร้างกุญแจสาธารณะดังกล่าว ตัวอย่างอัลกอริทึม เช่น RSA Encryption, Digital SignatureAlgorithm เป็นต้น

### 3.5.3 ตัวอย่างใบรับรองอิเล็กทรอนิกส์

ใบรับรองอิเล็กทรอนิกส์ประกอบไปด้วยข้อมูลต่างๆ ซึ่งองค์ประกอบหลักคือ ข้อมูลพื้นฐาน ส่วนข้อมูลเพิ่มเติมนั้นไม่จำเป็นที่จะต้องมีการเข้ารหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Certificate:**  
**Data:**  
**Version:** 3 (0x2)  
**Serial Number:** 63 (0x3F)  
**Signature Algorithm:** sha1WithRSAEncryption  
**Issuer:** C=TH, S=Bangkok, O=GITS, OU=PKI, CN=GITSCA Internal  
**Validity**  
 Not Before: Jun 27 04:10:50 2002 GMT  
 Not After: Dec 1 01:00:00 2002 GMT  
**Subjects:** C=TH, O=GOV, OU= National Science and Technology Development Agency,  
 OU = National Electronics and Computer Technology Center, CN= Nuntana  
 Podjananuntakul ; Email= nuntana.podjananuntakul@gits.net.th  
**Subject Public Key Info:**  
**Public Key Algorithm:** rsaEncryption  
**RSA Public Key:** (1024 bit)  
**Modulus:** (1024 bit):  
 00:cb:8a:c4:cb:e6:55:23:13:f4:ee:48:99:46:21:  
 78:2b:d5:93:07:37:83:c4:70:00:33:4a:a9:d2:8f:  
 ...  
 6c:4e:5f:9c:b5:83:5c:f9:79  
**Exponent:** 65537 (0x10001)  
**X.509v3 extensions:**  
**X.509v3 Basic Constraints:**  
 CA: FALSE  
**X.509v3 Subject Key Identifier**  
 95 :85:e6:83:99:51:2B:72:61:ES:4DF0:41:93:8C:  
 85 :DE:28:8C:25  
**X.509v3 Authority Key Identifier**  
 KeyID:FC:B9:14:62:02:12:3B:F2:33:3C:D3:BA:71:D6:38:06:32:7A:  
 :9:1:1C  
 DirName:/C=TH/S=Bangkok/O=GITS/OU=PKI/CN=GITSCA  
 Internal  
 Serial:00  
**Signature Algorithm:** sha1WithRSAEncryption  
 ab:a2:26:79:79:a0:00:b1:a0:f8:a:c6:7:4c:56:54:91:56:53:  
 2d:05:ic5:3d:54:66:ad:fa:2e:f8:ef:1d:f:d0:c:d:f:88:de:95:  
 ...  
 b7:5a:3f:50:b1:8d:dc:4a

รูปที่ 3.7 แสดงตัวอย่าง Certificate

### 3.5.4 ชนิดของใบรับรองอิเล็กทรอนิกส์

#### 3.5.4.1 Personal Certificate

ใบรับรองอิเล็กทรอนิกส์สำหรับบุคคล เป็นใบรับรองที่ใช้ในการยืนยันตัวตนบุคคลทางโลก อิเล็กทรอนิกส์ ทำให้ผู้ประกอบธุรกรรมอิเล็กทรอนิกส์สามารถมั่นใจได้ว่า บุคคลที่ติดต่อดังนี้มีตัวตนและเป็นบุคคลที่อ้างถึงจริง รวมทั้งยังก่อให้เกิดความปลอดภัยของข้อมูลที่สื่อสารระหว่างกัน ทำให้ผู้อื่น ไม่สามารถอ่านข้อมูลดังกล่าวได้ ใบรับรองอิเล็กทรอนิกส์สำหรับบุคคลสามารถนำไปประยุกต์ใช้ได้ ในหลายลักษณะงาน เช่น การรับ-ส่งจดหมายอิเล็กทรอนิกส์แบบปลอดภัย (Secure e-mail) การยืนยันตัวตนในการเข้าใช้งานเว็บไซต์ผ่านทางเว็บเบราว์เซอร์ (Client Authentication) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

• สิ่งที่ได้จากการใช้งานใบรับรองอิเล็กทรอนิกส์

การพิสูจน์ตัวตนจริง (Authentication): เป็นการระบุตัวตนแท้จริงของผู้ส่งข้อมูลอิเล็กทรอนิกส์

การรักษาความครบถ้วนของข้อมูล (Data Integrity): สามารถตรวจสอบได้ว่า ข้อมูลที่ได้รับนั้น มีความถูกต้องครบถ้วนและไม่ถูกเปลี่ยนแปลงแก้ไข

การรักษาความลับของข้อมูล (Data Confidentiality): เพื่อป้องกันมิให้ข้อมูลถูกเปิดเผยโดยบุคคลซึ่งมิได้รับอนุญาตหรือไม่มีสิทธิ

การห้ามปฏิเสธความรับผิดชอบ (Non-repudiation): เป็นการป้องกันไม่ให้บุคคลผู้ส่งปฏิเสธว่าตนไม่ได้ส่งข้อมูลอิเล็กทรอนิกส์

• การประยุกต์ใช้งาน Personal Certificate

ระบบจดหมายอิเล็กทรอนิกส์แบบปลอดภัย (Secure Electronic Mail System)

ระบบจดหมายอิเล็กทรอนิกส์ สามารถนำใบรับรองอิเล็กทรอนิกส์มาใช้ในการเข้ารหัสลับและลงลายมือชื่อดิจิทัล เพื่อเป็นกายืนยันตัวตนผู้ส่ง ยืนยันความถูกต้องครบถ้วนของข้อมูลรวมทั้งยังสามารถรักษาความลับของข้อมูลในจดหมายให้อ่านได้เฉพาะผู้รับที่ถูกระบุไว้ได้อีกด้วย

การยืนยันตัวตนของผู้ใช้บริการ (Client Authentication)

ผู้ให้บริการสามารถใช้ใบรับรองอิเล็กทรอนิกส์ ในการยืนยันตัวตนก่อนเข้าใช้บริการเว็บไซต์ได้ (ในกรณีที่เว็บไซต์นั้นต้องการยืนยันตัวบุคคล) เพื่อเป็นการยืนยัน/ระบุตัวตนของผู้ใช้บริการ อีกทั้งยังเป็นการสร้างช่องทางการสื่อสารแบบปลอดภัยระหว่างเครื่องให้บริการ (Server) และเครื่องใช้บริการ (Client) ด้วย

การประยุกต์ใช้งานกับแอปพลิเคชันอื่น ๆ

ใบรับรองอิเล็กทรอนิกส์สามารถนำไปประยุกต์ใช้งานกับแอปพลิเคชันต่างๆ

นอกเหนือจากที่ได้กล่าวข้างต้น โดยพิจารณาว่าส่วนใดของแอปพลิเคชันที่ต้องการความปลอดภัยของข้อมูลก็สามารถนำเทคโนโลยีโครงสร้างพื้นฐานของระบบกฎหมายสาระณะไปผนวกกับส่วนนั้นๆ ซึ่งการประยุกต์ในลักษณะดังกล่าว จะต้องมีการพัฒนาแอปพลิเคชันเฉพาะ เพื่อให้สามารถทำงาน

ร่วมกับใบรับรองอิเล็กทรอนิกส์ได้

### 3.5.4.2 Web Server Certificate

ใบรับรองอิเล็กทรอนิกส์สำหรับเครื่องให้บริการ (เช่น เว็บเซิร์ฟเวอร์) เป็นใบรับรองที่ใช้ในการยืนยันเครื่องให้บริการ ทำให้ผู้ประกอบการอิเล็กทรอนิกส์สามารถมั่นใจได้ว่า เครื่องให้บริการที่ติดต่อด้วยนั้นเป็นเครื่องที่อ้างถึงจริง รวมทั้งยังก่อให้เกิดความปลอดภัยของข้อมูลที่

รับส่งระหว่างเครื่องให้บริการ (Server) และเครื่องใช้บริการ (Client)

• สิ่งที่ได้รับจากการใช้งาน Web Server Certificate

การระบุตัวตนที่แท้จริงของเครื่องให้บริการ ทำให้ผู้ใช้บริการมั่นใจได้ว่าเป็นเครื่องใช้บริการซึ่งอ้างถึงจริง

ช่องทางสื่อสารแบบปลอดภัย เพื่อการรักษาความลับของข้อมูลที่รับส่งระหว่างเครื่องใช้บริการและเครื่องใช้บริการ

• การประยุกต์ใช้งาน Web Server Certificate

- การยื่นแบบและชำระภาษีผ่านอินเทอร์เน็ต
- การจัดซื้อจัดจ้างอิเล็กทรอนิกส์
- การจดทะเบียนนิติบุคคลแบบออนไลน์
- การขออนุมัติงบประมาณแบบออนไลน์
- การประกอบธุรกรรมทางการเงินต่างๆ ของธนาคารแบบออนไลน์
- การสั่งซื้อสินค้าแบบออนไลน์

ใบรับรองอิเล็กทรอนิกส์และกุญแจสามารถนำมาใช้งาน เพื่อก่อให้เกิดความเชื่อมั่นในการรักษาความปลอดภัยของข้อมูล ในแง่การรักษาความลับ (Confidentiality) ความถูกต้องของข้อมูล (Integrity) การระบุตัวตน (Authentication) และการป้องกันการปฏิเสธความรับผิดชอบ (Non-repudiation) ถ้าผู้ใช้งานใบรับรองอิเล็กทรอนิกส์มีความรู้ความเข้าใจเกี่ยวกับกระบวนการในการใช้งาน รูปแบบ และข้อมูลในใบรับรองอิเล็กทรอนิกส์ ทำให้การใช้งานมีประสิทธิภาพยิ่งขึ้น อีกทั้งยังเป็นประโยชน์แก่ตัวผู้ใช้งานด้วย

### 3.6 Secure Socket Layer (SSL)

Secure Sockets Layer (SSL) เริ่มพัฒนาโดย Netscape Communications เพื่อใช้ในโปรโตคอลระดับแอปพลิเคชันคือ Hypertext Transfer Protocol (HTTP) ซึ่งเป็นการสื่อสารผ่านเว็บให้ปลอดภัย พัฒนาในช่วงต้นของยุคการค้าอิเล็กทรอนิกส์กำลังได้รับความนิยมในโลกอินเทอร์เน็ต SSL ทำให้เกิดการสื่อสารอย่างปลอดภัยระหว่างไคลเอนต์และเซิร์ฟเวอร์ โดยการอนุญาตให้มีกระบวนการพิสูจน์ตัวตนกับการใช้งานลายเซ็นดิจิทัลสำหรับการรักษาความถูกต้องของข้อมูล และการเข้ารหัสข้อมูลเพื่อป้องกันความเป็นส่วนตัวระหว่างการสื่อสารข้อมูล โปรโตคอล SSL อนุญาตให้สามารถเลือกวิธีการในการเข้ารหัส วิธีสร้างไคเจสต์ [\*1] และลายเซ็นดิจิทัล ได้อย่างอิสระก่อนการสื่อสารจะเริ่มต้นขึ้น ตามความต้องการของทั้งเว็บเซิร์ฟเวอร์และบราวเซอร์ ทั้งนี้เพื่อเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพิ่มความยืดหยุ่นในการใช้งาน เปิดโอกาสให้ทดลองใช้วิธีการในการเข้ารหัสวิธีใหม่ รวมถึงลดปัญหาการส่งออกวิธีการเข้ารหัสไปประเทศที่ไม่อนุญาต Netscape เริ่มพัฒนา SSL เวอร์ชันแรกคือเวอร์ชัน 2.0 และเวอร์ชันถัดมาเป็น 3.0 ซึ่งสนับสนุนความสามารถด้านความปลอดภัยมากขึ้น และเป็นเวอร์ชันสุดท้ายก่อนที่จะเป็นมาตรฐานกลางของโพรโตคอลบนอินเทอร์เน็ต โดยเปลี่ยนชื่อเป็น Transport Layer Security หรือ TLS ซึ่งดูแลมาตรฐานโดย Internet Engineering Task Force (IETF) อธิบายเวอร์ชันของ SSL และผู้พัฒนาได้ตามตาราง

ตารางที่ 3.1 แสดงเวอร์ชันและผู้พัฒนาเทคโนโลยี SSL

เวอร์ชัน	ผู้พัฒนา	จุดเด่น	เบราว์เซอร์ที่สนับสนุน
SSL v2.0	Netscape Corp. [SSL2]	โพรโตคอล SSL รุ่นแรกที่พัฒนาบนเบราว์เซอร์	NS Navigator 1.x/2.x MS IE 3.x Lynx/2.8 + OpenSSL
SSL v3.0	Netscape Corp. เป็น Internet Drafted รุ่นก่อนเป็นมาตรฐานกลาง [SSL3]	ปรับปรุงใหม่เพิ่มความปลอดภัยมากขึ้น สนับสนุนการใช้ non-RSA ciphers ในการเข้ารหัส และห่วงโซ่ Certificate[*2]	NS Navigator 2.x/3.x/4.x MS IE 3.x/4.x
TLS v1.0	IETF กำลังเสนอให้เป็นมาตรฐานโพรโตคอลบนอินเทอร์เน็ต (Proposed Internet Standard)	ปรับปรุงจาก SSL v3.0 สนับสนุนการทำงานในชั้น MAC และ HMAC เพิ่ม Padding ชนิด Block และวิธีการจัดลำดับข้อมูล และเพิ่มระดับการแข็งแกร่ง	Lynx/2.8 + OpenSSL

[\*1] โดเจสต์ (Digest) คือข้อความที่เกิดจากการเข้ารหัสข้อมูลด้วยฟังก์ชันแฮชเช่น MD5หรือSHA-1

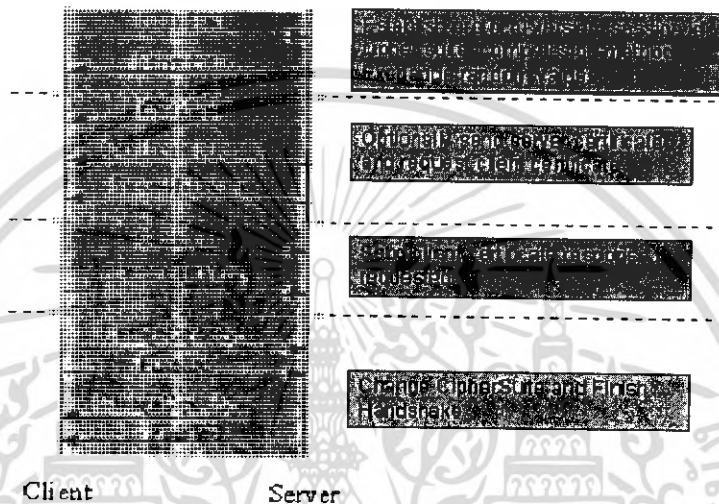
[\*2] ห่วงโซ่ Certificate (Certificate Chain) คือการเพิ่มข้อมูล Certificate ที่เกี่ยวเนื่องกันเมื่อใช้ขั้นตอนแลกเปลี่ยนข้อมูล ซึ่งจะช่วยลดเวลาในการค้นหา Certificate จากผู้ให้บริการ Certificate Authority (CA) ที่เกี่ยวเนื่องกันมากกว่า 1 ชั้นไป

### 3.6.1 กระบวนการในการเริ่มต้นการสื่อสารผ่านชั้น SSL

แบ่งเป็น 4 ขั้นตอนคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1 ประกาศชุดวิธีการเข้ารหัส โคลเจสต์ และลายเซ็นดิจิทัลที่สนับสนุนของทั้งไคลเอ็นต์และเซิร์ฟเวอร์
- 2 การพิสูจน์ตัวตนของเซิร์ฟเวอร์ต่อไคลเอ็นต์
- 3 การพิสูจน์ตัวตนของไคลเอ็นต์ต่อเซิร์ฟเวอร์ ถ้าจำเป็น
- 4 ไคลเอ็นต์และเซิร์ฟเวอร์ตกลงชุดวิธีการเข้ารหัส การสร้างโคลเจสต์ และการใช้ลายเซ็นดิจิทัล ตามรูป



รูปที่ 3.8 แสดงกระบวนการเริ่มต้นการติดต่อสื่อสารของโพรโตคอล SSL

ขั้นตอน 1: ประกาศชุดวิธีการเข้ารหัส โคลเจสต์ และลายเซ็นดิจิทัลที่สนับสนุนของทั้งไคลเอ็นต์และเซิร์ฟเวอร์

ไคลเอ็นต์และเซิร์ฟเวอร์ส่งข้อความเริ่มต้นการสื่อสาร (Hello message) ซึ่งประกอบไปด้วยเวอร์ชันของโพรโตคอลที่ใช้ วิธีการเข้ารหัสที่เว็บเซิร์ฟเวอร์และไคลเอ็นต์สนับสนุน หมายเลขระบุการสื่อสาร (Session identifier) รวมถึงวิธีการบีบอัดข้อมูลในการสื่อสารที่สนับสนุน หมายเลขระบุการสื่อสารที่เกิดขึ้น ใช้สำหรับตรวจสอบการเชื่อมต่อระหว่างไคลเอ็นต์และเซิร์ฟเวอร์ ถ้ามีการเชื่อมต่อก่อนหน้านี้เกิดขึ้น แสดงว่าได้มีการตกลงวิธีการสื่อสารแล้ว สามารถเริ่มต้นส่งข้อมูลได้ทันที เป็นการลดเวลาติดต่อสื่อสารลง

ขั้นตอน 2: การพิสูจน์ตัวตนของเซิร์ฟเวอร์ต่อไคลเอ็นต์

ถัดมาเว็บเซิร์ฟเวอร์ทำการส่ง Certificate หรือ ใบยืนยันความมีตัวตนของเซิร์ฟเวอร์ ไคลเอ็นต์จะทำการตรวจสอบ Certificate กับผู้ให้บริการ Certificate Authority ที่ได้ตั้งค่าไว้ เพื่อยืนยันความถูกต้องของ Certificate ของเซิร์ฟเวอร์

ขั้นตอน 3: การพิสูจน์ตัวตนของไคลเอ็นต์ต่อเซิร์ฟเวอร์ ถ้าจำเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซิร์ฟเวอร์สามารถร้องขอ Certificate จาก ไคลเอ็นต์เพื่อตรวจสอบความถูกต้องของ Client ด้วยก็ได้ ใช้ในกรณีที่มีการจำกัดการใช้งานเฉพาะไคลเอ็นต์ที่ต้องการเท่านั้น ซึ่ง SSL สนับสนุนการตรวจสอบได้จากทั้งเซิร์ฟเวอร์และไคลเอ็นต์ ขึ้นอยู่กับการเลือกใช้งานในขณะติดต่อดังกล่าวที่เกิดขึ้น

ขั้นตอน 4: ไคลเอ็นต์และเซิร์ฟเวอร์ตกลงชุดวิธีการเข้ารหัส การสร้างไคเจสต์ และการใช้ลายเซ็นดิจิทัลขั้นตอนการตรวจสอบ Certificate ที่เซิร์ฟเวอร์ร้องขอจากไคลเอ็นต์จะมีหรือไม่มีก็ได้ ขึ้นอยู่กับการตั้งค่าบนเซิร์ฟเวอร์ หลังจากขั้นตอนการตรวจสอบเสร็จสิ้น เซิร์ฟเวอร์และไคลเอ็นต์จะตกลงการใช้งานวิธีการเข้ารหัสระหว่างกัน โดยใช้ค่าที่ได้จากการประกาศในขั้นตอนแรก

### 3.6.2 วิธีการแลกเปลี่ยนกุญแจในการเข้ารหัส (Key exchange method)

คือการกำหนดกลไกการแลกเปลี่ยนกุญแจที่ใช้ในการเข้ารหัสระหว่างการสื่อสาร โดยทั้งไคลเอ็นต์และเซิร์ฟเวอร์จะใช้กุญแจนี้ในการเข้ารหัสและถอดรหัสข้อมูล ใน SSL เวอร์ชัน 2.0 จะสนับสนุนวิธีการแลกเปลี่ยนกุญแจแบบ RSA ส่วน SSL เวอร์ชัน 3.0 ขึ้นไปจะสนับสนุนวิธีการอื่นๆ เพิ่มเติมเช่นการใช้ RSA ร่วมกับการใช้ Certificate หรือ Diffie-Hellman เป็นต้น

### 3.6.3 วิธีการเข้ารหัส

ในปัจจุบันแบ่งเป็นสองวิธีคือ การใช้กุญแจเดียวกันในการเข้ารหัสและถอดรหัส อาจเรียกว่า Session key หรือ Secret key ส่วนอีกวิธีการคือ การใช้กุญแจคนละตัวในการเข้ารหัสและถอดรหัส ประกอบไปด้วยกุญแจสาธารณะและกุญแจส่วนตัวซึ่งเป็นที่คู่กันเสมอ การเข้ารหัสด้วยกุญแจใด จะต้องถอดรหัสด้วยกุญแจที่คู่กันและตรงกันข้ามเท่านั้น มักใช้วิธีการเข้ารหัสด้วยกุญแจคนละตัวมาใช้ในการเข้ารหัส Session key และส่งไปให้ฝั่งตรงข้ามก่อนการสื่อสารจะเกิดขึ้นรวมเรียกว่าวิธีการแลกเปลี่ยนกุญแจในการเข้ารหัส

SSL ใช้วิธีการเข้ารหัสด้วยกุญแจสมมาตร หรือกุญแจเดียวในการเข้ารหัสและถอดรหัสตามที่กล่าวข้างต้น วิธีการเข้ารหัสคือ การเข้ารหัสด้วย DES และ 3DES (Data Encryption Standard), วิธีการเข้ารหัสด้วย IDEA ส่วน RC2 และ RC4 เป็นวิธีการเข้ารหัสของ RSA รวมถึงวิธีการเข้ารหัสแบบ Fortezza สำหรับความยาวของการเข้ารหัสที่ใช้คือ 40 บิต, 96 บิต และ 128 บิต

การสร้าง Message Authentication Code (MAC) เพื่อใช้สำหรับการยืนยันความถูกต้องของข้อมูลระหว่างการสื่อสารและป้องกันการปลอมข้อมูล ส่วนฟังก์ชันสร้างไคเจสต์ที่ SSL สนับสนุนและเลือกใช้ได้ในปัจจุบันคือ MD5 ขนาด 128 บิต และ SHA-1 (Secure Hash Algorithm) ขนาด 160 บิต ซึ่งจะได้รับการที่ทั้งสองฝ่ายสนับสนุนและเหมาะสมซึ่งเป็นขั้นตอนสุดท้ายก่อนการสื่อสารที่มีการเข้ารหัสจะเริ่มต้นขึ้น

## บทที่ 4

### WS-Transaction

#### 4.1 Transaction

Transaction คือ หน่วยของงานที่ประกอบไปด้วยหน่วยของงานย่อยๆ โดยเราจะพิจารณา transaction เป็น 2 ประเภท คือ Traditional Transaction และ Transaction in web service context

##### 4.1.1 Traditional Transaction

สำหรับ Transaction ประเภทนี้มักจะหมายถึงความถึง Transaction บนฐานข้อมูล โดยมีจุดประสงค์เพื่อให้คงไว้ซึ่งความถูกต้องของข้อมูลในฐานข้อมูล โดย Transaction ประเภทนี้จะมีคุณสมบัติของ ACID Properties ซึ่งมีรายละเอียด ดังนี้

- Atomicity ทุกๆคำสั่งภายใต้ Transaction ต้องทำได้สำเร็จ ถ้ามีเพียงคำสั่งใดคำสั่งหนึ่งทำไม่สำเร็จหรือเกิดข้อผิดพลาดขึ้น ทุกคำสั่งที่ทำมาต้องเสมือนว่าไม่เคยเกิดขึ้นมาก่อนบนระบบ

- Consistency ผลลัพธ์สุดท้ายของ Transaction ต้องไม่ผิดกฎของ database
- Isolation ถ้าในการทำงานมีหลาย Transaction ทำงานพร้อมๆกัน Transaction เหล่านั้นต้องไม่สามารถเข้าถึงทรัพยากรของอีกฝ่ายได้

- Durability เมื่อ transaction มีการ commit แล้วถือว่าการเปลี่ยนแปลงที่ถาวรแล้ว ไม่ว่าจะมียุทธศาสตร์เกิดขึ้นตามมาภายหลังหรือไม่

- Traditional Transaction จะใช้เวลาในการทำงานสั้น ทรัพยากรที่ Transaction ต้องการจะถูกจองไว้ให้กับ Transaction จนกว่าจะทำงานเสร็จแล้วจึงมีการคืนทรัพยากรกลับมาให้ระบบ ซึ่งจะพบว่าถ้าเรานำการใช้งาน Transaction ประเภทนี้มาใช้ร่วมกับเว็บเซอร์วิส จะพบว่า

- เนื่องจากเว็บเซอร์วิสแต่ละตัวมีการทำงานที่แยกจากกัน โดยชัดเจน ดังนั้นเว็บเซอร์วิสแต่ละตัวจึงมีการจัดการ Transaction ภายใต้อุปกรณ์ของ Transaction ในตัวมันเอง ซึ่งเราไม่สามารถนำการจัดการ Transaction บนฐานข้อมูลมาใช้ใน Transaction บนเว็บเซอร์วิสได้

- Transaction บนฐานข้อมูลมีการจองทรัพยากรจนกว่า Transaction จะทำงานเสร็จซึ่งบนเว็บเซอร์วิสนั้นเป็นการทำงานกันบนเครือข่ายซึ่งถ้าการเชื่อมต่อล้มเหลว เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Transaction อาจจะไม่ปล่อยทรัพยากรคืนกลับให้ระบบได้

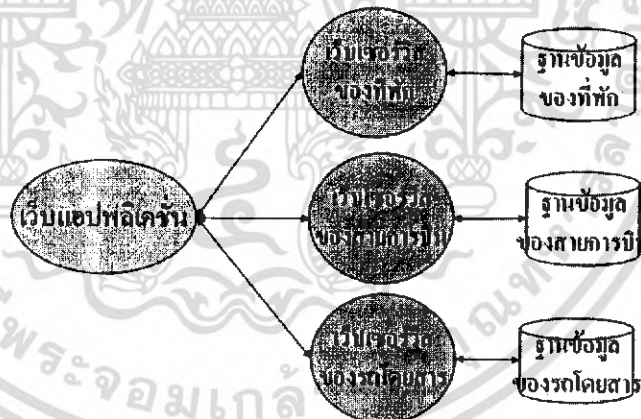
จากการที่ Transaction บนฐานข้อมูลจะทำการจองทรัพยากรจนกว่าจะทำงานเสร็จ ทำให้เป็นช่องทางในการโจมตีจากผู้อื่นโดยวิธี DoS (Denial of service) ได้

#### 4.1.2 Transaction บนเว็บเซอร์วิส

Transaction บนเว็บเซอร์วิสจะเป็นการมองว่าการทำงานแต่ละเซอร์วิสเป็นการทำงานหนึ่งคำสั่ง เพราะฉะนั้นภายใต้ขอบเขตของ Transaction จะแทนที่ด้วยกลุ่มของเซอร์วิสที่มาประกอบกันเป็นหนึ่ง Transaction ตัวอย่างเช่น ถ้าผู้ใช้ต้องการทำการจองตั๋วเว็บแอปพลิเคชันของบริษัทที่รับผิดชอบตั๋ว จะต้องทำการจองดังต่อไปนี้

- ทำการจองที่พัก
- ทำการจองสายการบิน
- ทำการจองรถที่จะใช้ในการเดินทาง

ซึ่งทั้งที่พัก สายการบิน และรถ ต่างก็มีเว็บเซอร์วิสเป็นของตัวเอง ดังนั้น เว็บแอปพลิเคชันทำหน้าที่เพียงแค่จัดการการจองทั้งหมดเป็น Transaction หนึ่ง ซึ่งถ้าการจองอย่างใดอย่างหนึ่งไม่สามารถทำการจองได้ แสดงว่าการจองทั้งหมดต้องถูกยกเลิก



รูปที่ 4.1 แสดงตัวอย่างของ Transaction บนเว็บเซอร์วิส

การจัดการ Transaction บนฐานข้อมูลจึงไม่เหมาะที่จะนำมาใช้งานกับ Transaction บนเว็บเซอร์วิสทำให้มีการกำหนดคุณสมบัติของ Transaction บนเว็บเซอร์วิส ดังนี้

- Transaction อาจจะมีระยะเวลาการทำงานที่ยาวนาน อาจจะเป็นชั่วโมง วัน หรือมากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Participant จะไม่ได้รับอนุญาตให้ทำการจองทรัพยากรของระบบไปเป็นของตัวเองเป็นระยะเวลานานๆ
  - การเชื่อมต่อระหว่าง participants อาจจะไม่มีความน่าเชื่อถือ (การเชื่อมต่ออาจล้มเหลวได้)
  - ACID property อาจจะมีการผ่อนปรนได้บ้างในบางกรณี บางแอปพลิเคชันอาจจะไม่ต้องการให้มีคุณสมบัติของ ACID ก็ได้
  - Transaction สามารถทำงานสำเร็จถึงแม้ว่าจะมีการ cancel โดย participant ใดๆ (แต่ต้องมีการ confirm โดย participant อื่นๆ ซึ่งหมายความว่าไม่มีคุณสมบัติของ ACID)
  - เนื่องจาก Transaction บนเว็บเซอร์วิสไม่จำเป็นต้องอยู่ภายใต้กลุ่มของความน่าเชื่อถือเดียวกัน Participant จึงสามารถเลือกใช้ Transaction Manager ของตัวเองได้เพื่อความปลอดภัยในการเข้าถึงทรัพยากรของตน
  - ทุกๆกิจกรรมจะต้องถูกบันทึกไว้
- เพื่อที่จะจัดการกับลักษณะของ Transaction ที่เกิดขึ้นบนเว็บเซอร์วิส จึงได้มีการกำหนดข้อกำหนดขึ้นมาเพื่อจัดการกับ Transaction บนเว็บเซอร์วิสซึ่งก็คือ WS-Transaction
- ภายใต้ WS-Transaction จะประกอบไปด้วยข้อกำหนดอีก 3 ข้อกำหนดดังนี้
- WS-Coordination ทำการอธิบายถึงโครงร่างของการทำงานร่วมกัน
  - WS-Atomic Transaction อธิบายถึงโปรโตคอลที่ใช้ใน WS-Coordination ซึ่งมีคุณสมบัติของ ACID
  - WS-Business Activity อธิบายถึงโปรโตคอลที่ใช้ใน WS-Coordination ซึ่ง
- ไม่มีคุณสมบัติของ ACID สำหรับในโครงงานนี้เราจะไม่ทำการพิจารณาถึง WS-Business Activity

#### 4.2 WS-Coordination

สำหรับ WS-Coordination จะเป็นการอธิบายถึงโปรโตคอลที่เกี่ยวข้องกับการทำงานในเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอปพลิเคชันแบบกระจาย (Distributed Application) โดย โครงสร้างที่ถูกกำหนดใน WS-Coordination จะทำให้แอปพลิเคชันมีความสามารถในการสร้าง Context ที่จำเป็นในการประกาศ Activity ตัวเองให้กับ แอปพลิเคชัน Service อื่นๆ และการ register สำหรับ coordination protocols (ขั้นตอนในการทำให้สามารถทำงานร่วมกันได้)

โครงสร้างใน WS-Coordination นี้ให้ความสามารถแก่ Transaction processing, workflow และ ระบบอื่นๆสามารถทำงานร่วมกันโดยซ่อนกลุ่มของโปรโตคอลที่ตัวมันต้องใช้ และการทำงานในสถานะแวดล้อมที่แตกต่างกัน

WS-Coordination ยังบรรยายถึงคำจำกัดความของ โครงสร้างของ Context และ requirement ของการเผยแพร่ Context ระหว่าง Service ที่ทำงานร่วมกัน

สำหรับ Protocol ของ Web service ที่ถูกจำกัดความไว้ใน WS-Coordination ควรที่จะถูกใช้เมื่อมีความต้องการที่จะทำงานระหว่าง platform ที่แตกต่างกัน Protocol ใน WS-Coordination จึงสามารถที่จะนำไปรวมกับกลุ่มของ protocol อื่นๆได้ภายใต้ แอปพลิเคชัน เดียวกัน

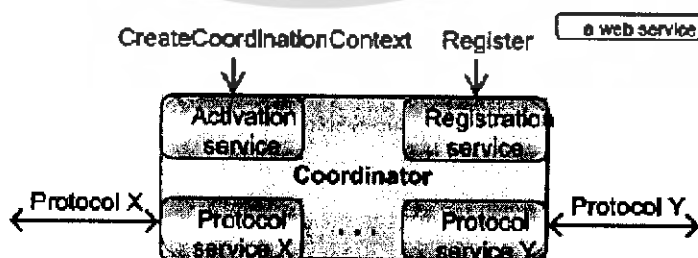
#### 4.2.1 Model

WS-Coordination ได้ทำการบรรยายถึง โครงสร้างของ Coordination Service (Coordinator) เอาไว้ซึ่งประกอบไปด้วย Service ดังนี้

- Activation Service, ซึ่งเป็นกระบวนการในการสร้าง Coordination instance (Context)
- Registration Service, ซึ่งเป็นกระบวนการในการให้ แอปพลิเคชัน register เข้ากับ

Coordination protocol

- Coordination Type ซึ่งเป็นเซตของ Coordination protocol



รูปที่ 4.2 แสดงเซอร์วิสที่ประกอบภายในCoordinator

แอปพลิเคชัน มีการใช้ Activation Service ในการที่จะสร้าง Coordination Context สำหรับเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Activity ของตัวเอง เมื่อ Coordination Context ถูกนำไปโดย แอปพลิเคชัน อื่น Coordination Context จะถูกส่งไปด้วยอะไรก็ตามที่เหมาะสมสำหรับ แอปพลิเคชัน อื่นๆนั้น

Context จะประกอบไปด้วยข้อมูลที่สำคัญในการที่จะให้ แอปพลิเคชัน อื่นๆนั้นสามารถที่จะเข้ามา Register สำหรับ Activity ได้โดยการกำหนดพฤติกรรมของ Coordination ที่ แอปพลิเคชัน นั้นต้องทำตาม

นอกจากนั้น แอปพลิเคชัน ที่ได้รับ Coordination Context ยังสามารถที่จะใช้ Registration Service กับตัวมันเอง หรืออาจจะใช้กับ Trusted Coordinator อื่นก็ได้ ในกรณีนี้กลุ่มของ Web Service อาจจะมาสัมพันธ์กันในกระบวนการเชื่อมต่อ

#### 4.2.2 Extensibility

โครงสร้างของ WS-Coordination มีความสามารถในการที่จะต่อขยายรวมไปถึงมีความยืดหยุ่นสำหรับการใช้งาน โดยโครงสร้างจะอนุญาตให้

- สามารถทำการตีพิมพ์ เผยแพร่ Coordination Protocols ใหม่ๆได้
- Protocol ที่ถูกเลือกมาจาก Coordination Type และจำกัดความของส่วนที่ต่อขยาย

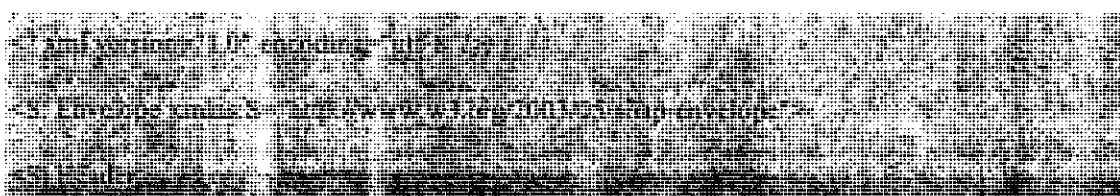
สามารถทำการเพิ่มเข้าไปใน Protocol และ message flows

ส่วนที่ทำการต่อขยายเข้ามาสามารถใช้ในการแลกเปลี่ยนข้อมูล ได้บน Message flow ที่ถูกจำกัดความไว้ใน WS-Coordination ซึ่งนำไปสู่ความต้องการในการแลกเปลี่ยนข้อมูลใน Isolation-level หรือใน Business-Level Coordination protocols ข้อมูลที่แลกเปลี่ยนกันสามารถทำการบันทึกไว้ได้เพื่อทำการตรวจสอบในภายหลัง

#### 4.2.3 Coordination Context

Coordination Context คือ ชนิดของ Context ชนิดหนึ่งที่ใช้ในการส่งข้อมูลของ Coordination ให้กับ Participants อื่นๆทุกตัวที่เกี่ยวข้องใน Coordination Service (Coordinator) ซึ่งตัว Coordination Context จะถูกใส่รวมไว้กับ แอปพลิเคชัน (Conveying a context-flowing the context)

*Coordination Context* จะให้ความสามารถในการ coordination registration service, Coordination Type และ relevant extensions ตัวอย่างของ *Coordination Context*



```

<wscoor:CoordinationContext
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:wscoor="http://schemas.xmlsoap.org/ws/2004/10/wscoor"
  xmlns:myApp="http://fabrikam123.com/myApp"
  S:mustUnderstand="true">
  <wscoor:Identifier>
    http://Fabrikam123.com/SS/1234
  </wscoor:Identifier>
  <wscoor:Expires>3000</wscoor:Expires>
  <wscoor:CoordinationType>
    http://schemas.xmlsoap.org/ws/2004/10/wsa
  </wscoor:CoordinationType>
  <wscoor:RegistrationService>
    <wsa:Address>
      http://Business456.com/mycoordination-service/registration
    </wsa:Address>
    <wsa:ReferenceProperties>
      <myApp:BetaMark> ... </myApp:BetaMark>
      <myApp:EBDCode> ... </myApp:EBDCode>
    </wsa:ReferenceProperties>
  </wscoor:RegistrationService>
  <myApp:IsolationLevel>
    Repeatable Read
  </myApp:IsolationLevel>
</wscoor:CoordinationContext>
</S:Header>
</S:Body>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

</S: Body >

</S: Envelope >

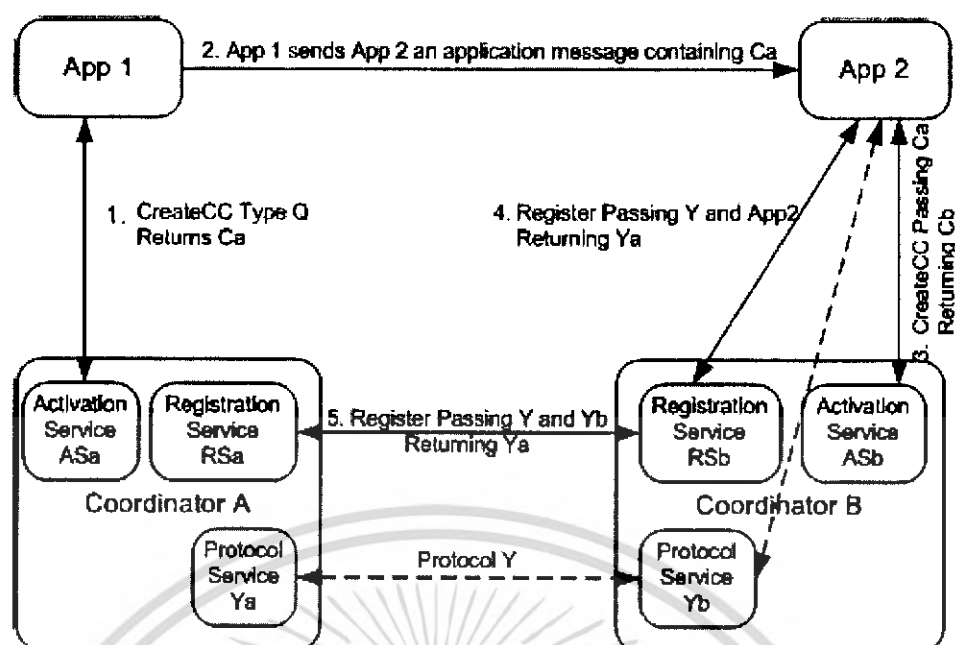
### รูปที่ 4.3 แสดง Coordination Context

แอปพลิเคชัน ต้องการใช้ Activity ที่เหมาะสมจาก Coordination Service (Coordinator) แอปพลิเคชัน ต้องมีการรวมเอา Coordination Context เอาไว้ใน message ด้วย Context ถูกแลกเปลี่ยนกันในลักษณะของ SOAP header 필드 mustUnderstand จะต้องถูกเซทให้มีค่าเป็นจริง เพื่อให้ แอปพลิเคชัน ต้องทำการรับ SOAP header (เนื่องจาก SOAP header เป็นเพียงตัวเลือก แอปพลิเคชัน อาจจะไม่ต้องสนใจก็ได้ แต่ถ้ามีการเซท mustUnderstand ให้เป็นจริง แอปพลิเคชัน จะต้องรับ SOAP header)

#### 4.2.4 Coordination Service (Coordinator)

Coordination Service (Coordinator) เป็นการประกอบไปด้วย Service ดังต่อไปนี้

- Activation Service มีการกำหนดให้ CreateCoordinationContext เป็นการสร้าง Coordination Context โดย Coordination Service อาจจะรองรับ Activation Service
- Registration Service เป็นการกำหนดขั้นตอนวิธีการของการ Register ซึ่งมีส่วนในการอนุญาติให้ Web Service เข้าร่วมใน Coordination protocol โดย Coordination service ต้องรองรับ Registration Service
- เซทของ Coordination protocol สำหรับแต่ละ Coordination type ที่รองรับ ซึ่งทั้งหมดนี้จะถูกกำหนดในข้อกำหนดในส่วนของ Coordination Type



รูปที่ 4.4 แสดงแอปพลิเคชัน และ Coordinator

จากรูปที่ 4.4 จะเป็นการแสดงการที่ แอปพลิเคชัน Services (App1 และ App2) กับ Coordinator ของตัวเอง (Coordinator A และ Coordinator B) ติดต่อกัน โดยมี Protocol Y และ Services Ya และ Yb เป็น Coordination Type ซึ่งไม่ได้ระบุใน WS-Coordination โดยจะมีขั้นตอนดังนี้

- App1 ทำการส่ง CreateCoordinationContext สำหรับ Coordination Type Q โดยได้รับ Context Ca กลับมาซึ่งประกอบไปด้วยตัวระบุถึง Activity (Activity identifier) A1 โดยที่ Coordination type Q และมี Endpoint ที่อ้างอิงถึง Registration Service ของ Coordinator A (RSa)
- จากนั้น App1 จะส่งแอปพลิเคชัน message ไปยัง App2 ซึ่งประกอบไปด้วย Context Ca
- แต่เนื่องจาก App2 ต้องการใช้ CoordinatorB เพราะฉะนั้น App2 จึงใช้ Create Coordination Context โดยมี Ca เป็น input แล้วทำการเสนอให้กับ CoordinatorB จากนั้น CoordinatorB จึงทำการสร้าง Coordination Context Cb ของมันเองซึ่งประกอบไปด้วยตัวระบุถึง Activity (Activity identifier) เช่นเดียวกัน และ Coordination Type เหมือนกับ Ca แต่เป็นการ Register เข้ากับ Service ของตัวมันเอง RSb

- App2 ทำการพิจารณา Coordination protocol ที่รองรับของ Coordination type Q จากนั้นทำการ Register สำหรับ Coordination protocol Y ที่ CoordinatorB จากนั้นทำการเปลี่ยนจุด Endpoint ให้อ้างอิงถึง App2 และ Protocol Service Yb ซึ่งจะทำให้เกิดการเชื่อมต่อเสมือนระหว่างจุด Endpoint ที่ Protocol Y สามารถใช้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การ Register นี้จะทำให้ CoordinatorB ทำการส่งต่อการ register ไปยัง Registration Service ของ Coordinator A (RSa) และทำการแลกเปลี่ยนจุด Endpoint สำหรับ Yb และ Protocol Service Ya ซึ่งจะช่วยให้เสมือนเกิดการเชื่อมต่อระหว่าง Endpoint ที่ protocol Y สามารถใช้ได้

#### Activation Service

สำหรับ Activation Service จะทำการสร้าง Activity ใหม่และคืนค่าเป็น Coordination Context ของมันกลับมา  
แอปพลิเคชัน ส่ง CreateCoordinationContext จากนั้น Activation Service จะทำการคืนค่าของ CreateCoordinationContextRespond

- CreateCoordinationContext การร้องขออันนี้จะทำให้มีการสร้าง coordination context ที่มีการรองรับ Coordination type (service ที่จัดหาเซทของ coordination protocol เช่น WS-Atomic Transaction) ซึ่งคำสั่งที่มีความจำเป็นเมื่อมีความต้องการที่จะเข้าถึงเครือข่ายที่มีสภาวะแวดล้อมแตกต่างกัน (ต่าง platform กัน)



รูปที่ 4.5 แสดง CreateCoordinationContext

/CreateCoordinationContext/CoordinationType

ตัวบ่งบอกเฉพาะที่แสดงถึง Coordination type ที่ต้องการสำหรับ activity (เช่น Atomic Transaction)

/CreateCoordinationContext/Expires

แสดงถึงเวลาที่หมดอายุ โดยอยู่ในรูปของตัวเลขที่ไม่มีเครื่องหมายในหน่วย Milliseconds ซึ่งอาจจะมีหรือไม่มีก็ได้

/CreateCoordinationContext/CurrentContext

อาจจะใช้ได้อย่างหลากหลายรวมไปถึงการ Recovery และ subordinate ซึ่งอาจจะมีหรือไม่มีก็ได้

/CreateCoordinationContext / {any}

Element ที่อาจจะใช้สำหรับรายละเอียดเพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

/CreateCoordinationContext /@{any}

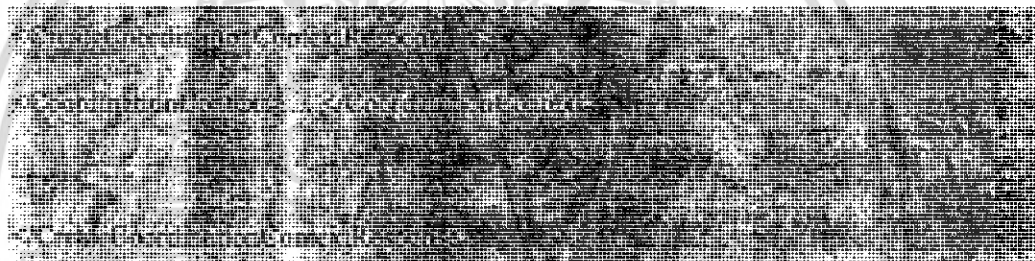
Attributes ที่อาจจะใช้สำหรับรายละเอียดเพิ่มเติม

ซึ่ง Message CreateCoordinationContext อาจจะมี โครงสร้างง่ายๆ ดังนี้ก็ได้



รูปที่ 4.6 แสดงโครงสร้างของ message CreateCoordinationContext

- CreateCoordinationContextResponse จะเป็นการคืนค่า CoordinationContext ที่ถูกสร้างขึ้นมา โดยจะมีโครงสร้างดังนี้



รูปที่ 4.7 แสดงการคืนค่า CoordinationContext

/CreateCoordinationContext/CoordinationContext

ส่วนนี้คือ Coordination Context ที่ถูกสร้างขึ้นมา

/CreateCoordinationContext / {any}

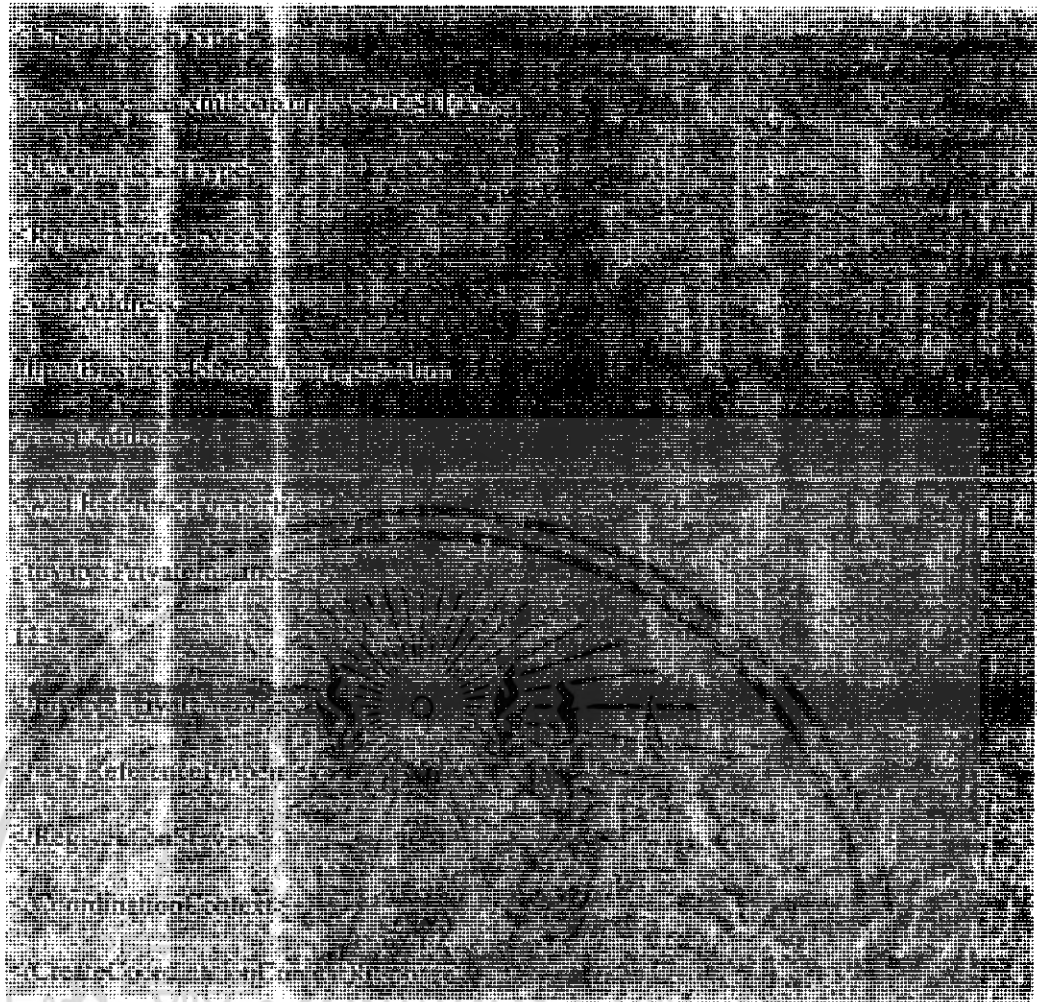
Element ที่อาจจะใช้สำหรับรายละเอียดเพิ่มเติม

/CreateCoordinationContext /@{any}

Attributes ที่อาจจะใช้สำหรับรายละเอียดเพิ่มเติม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

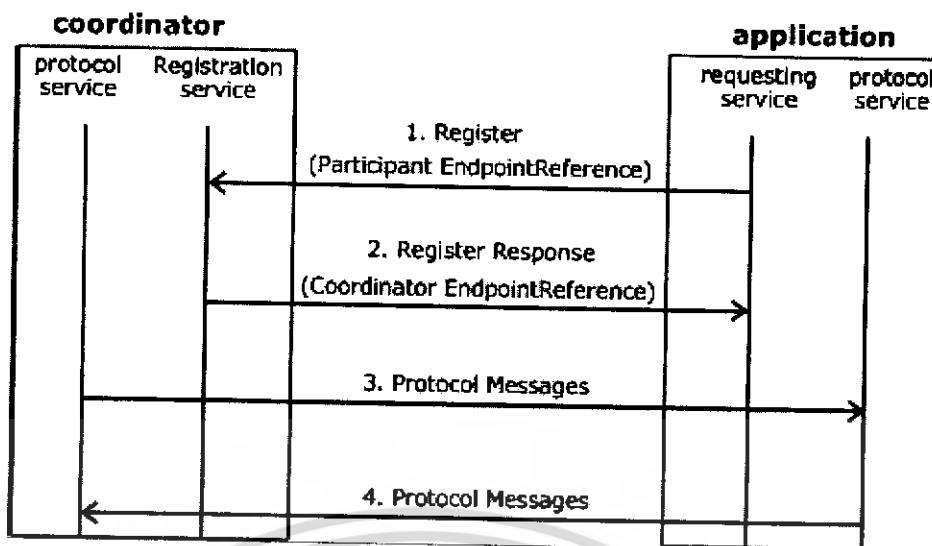


รูปที่ 4.8 แสดงรายละเอียดเพิ่มเติม

#### Registration Service

หลังจากที่ แอปพลิเคชัน มี Coordination context จาก coordinator ที่ถูกเลือก แอปพลิเคชัน จะสามารถเข้าไป register สำหรับ Activity นั้นๆ ได้ โดย interface ที่ถูกจัดทำให้ สำหรับการ register ของ แอปพลิเคชัน เพื่อเข้าถึง activity และสำหรับ coordinator ในการ register เพื่อเข้าถึง activity นั้นเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงการใช้จุดอ้างอิง Endpoint ระหว่างการลงทะเบียน

จากรูปที่ 21 coordinator จัดหาจุดอ้างอิง endpoint ในการ registration ให้ใน Coordination Context ในขั้นตอน CreateCoordinationContext ซึ่ง service ที่ทำการร้องขอจุดอ้างอิง endpoint จะได้รับใน CoordinationContext ซึ่งอยู่ใน แอปพลิเคชัน message

- register message จะทำการอ้างถึงจุดอ้างอิง endpoint นี้ และทำการรวมเอาจุดอ้างอิง endpoint ของ service participant protocol เอาไว้เป็น parameter
- RegisterResponse ทำการรวมเอาจุดอ้างอิง endpoint ของ protocol service ของ coordinator
- ในจุดนี้ทั้งสองฝั่งต่างมีจุดอ้างอิง endpoint ของ protocol service ของกันและกัน ดังนั้น Protocol message สามารถ target ไปยังอีกฝั่งได้

จุดอ้างอิง Endpoint เหล่านี้อาจจะมี ws:ReferenceProperties เป็นส่วนประกอบด้วยก็ได้ จากกฎของการ mapping ที่ถูกระบุไว้ใน WS-Addressing คุณสมบัติการอ้างอิงทั้งหมดต้องถูก copy เหมือนเป็น header ในทุกๆ message ที่ target ไปยัง endpoint

- Register Message
- RegistrationResponse Message

#### 4.2.5 Coordination Fault

WS-Coordination Fault ต้องมีการ include URI: <http://schemas.xmlsoap.org/ws/2004/10/wxcoor/fault>

ws/2004/10/wxcoor/fault ใน WS-Coordination จะมีการนิยามความผิดพลาดไว้โดยมีคุณสมบัติ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนี้

[Code]	Fault code
[Subcode]	Fault subcode
[Reason]	ส่วนของเหตุผล
[Detail]	รายละเอียดต่างๆ ถ้าว่างคือไม่มีรายละเอียดของความผิดพลาด

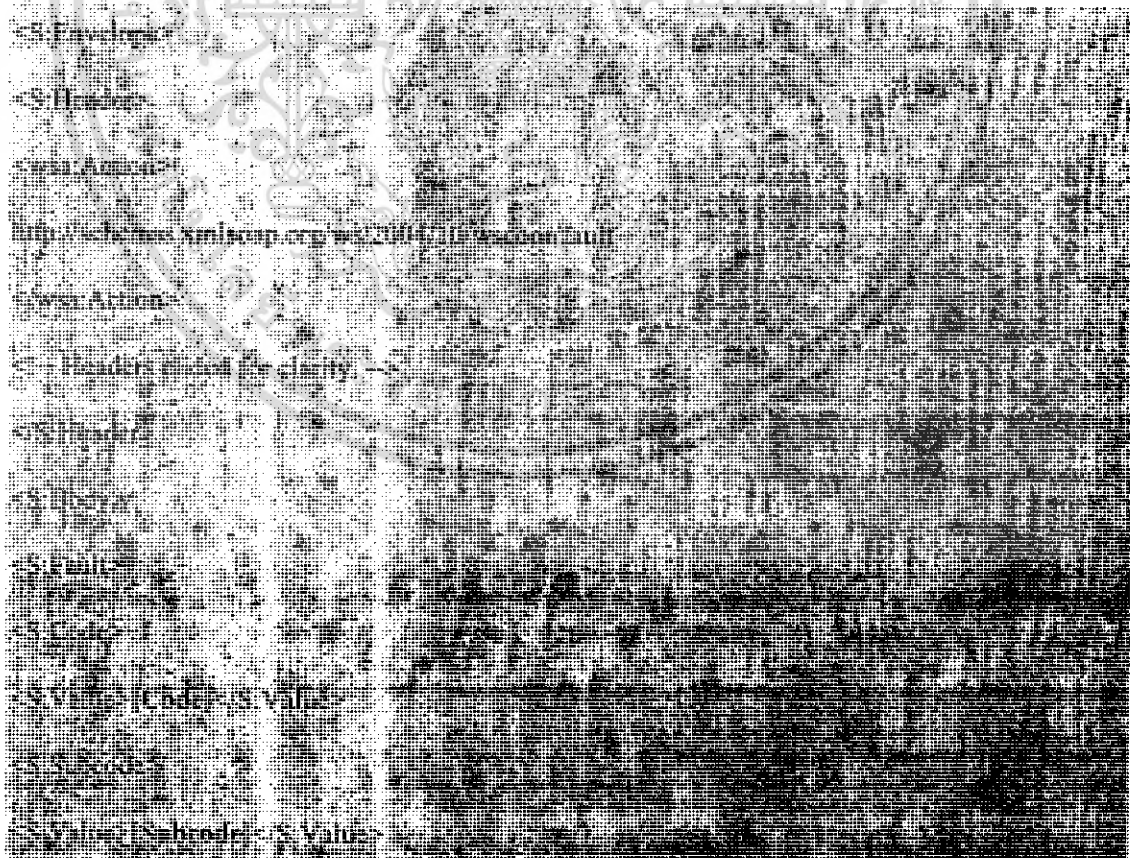
สำหรับ SOAP 1.2 [Code] ต้องเป็น Sender หรือ Receiver โดยคุณสมบัติเหล่านี้จะ

ถูกทำให้เป็น XML ดังตาราง

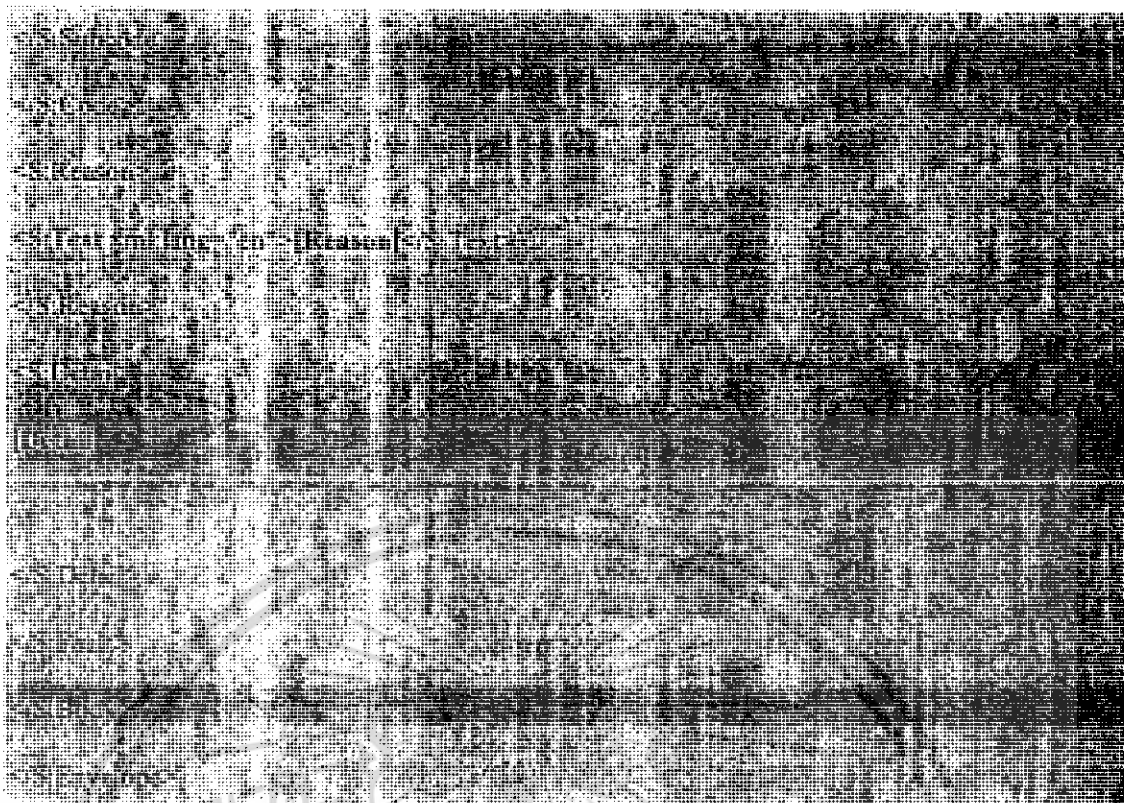
ตารางที่ 4.1 แสดงคุณสมบัติของ [Code] สำหรับ SOAP 1.2

SOAP VERSION	SENDER	RECEIVER
SOAP 1.2	S:SENDER	S:RECEIVER

โดยสามารถนำเอาคุณสมบัติในตารางข้างต้นมาเชื่อมเป็น SOAP 1.2 ได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แสดง SOAP 1.2

**Invalid State**

สำหรับความผิดพลาดประเภทนี้จะถูกส่งได้ทั้ง Coordinator หรือ Participant เพื่อจะกำหนดจุด Endpoint ที่ทำให้เกิดความผิดพลาดนั้น ซึ่งได้รับ *message* ที่ไม่ *valid* สำหรับสถานะตอนนั้นของมัน ซึ่งสถานการณ์นี้ไม่สามารถทำการกู้คืนได้

## คุณสมบัติ

[Code]	Sender
[Subcode]	wscor:InvalidState
[Reason]	The message was invalid for the current state of the activity.
[Detail]	unspecified

**Invalid Protocol**

สำหรับความผิดพลาดประเภทนี้จะถูกส่งได้ทั้ง Coordinator หรือ Participant เพื่อจะกำหนดจุด Endpoint ที่ทำให้เกิดความผิดพลาดนั้น ซึ่งได้รับ *message* ที่ถูกส่งมาด้วย *Protocol* ที่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ถูกต้องซึ่งสถานการณ์นี้ไม่สามารถทำการกู้คืนได้

คุณสมบัติ

[Code] Sender

[Subcode] wscor:InvalidProtocol

[Reason] The protocol is invalid or is not supported by the coordinator

[Detail] unspecified

### Invalid Parameters

สำหรับความผิดพลาดประเภทนี้จะถูกส่งได้ทั้ง Coordinator หรือ Participant เพื่อจะกำหนดจุด Endpoint ที่ทำให้เกิดความผิดพลาดนั้น ซึ่งได้รับ message ที่มี parameter ที่ invalid ซึ่งสถานการณ์นี้ไม่สามารถทำการกู้คืนได้

คุณสมบัติ

[Code] Sender

[Subcode] wscor:InvalidParameters

[Reason] The message contained invalid parameters and could not be processed

[Detail] unspecified

### No Activity

สำหรับความผิดพลาดประเภทนี้จะถูกส่งได้จาก Coordinator ถ้า participant ไม่มีการติดต่อมาเป็นระยะเวลาที่นานเกินไป

คุณสมบัติ

[Code] Sender

[Subcode] wscor:ContextRefused

[Reason] The coordination context that was provided could not be accepted

[Detail] unspecified

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Context Refused

สำหรับความผิดพลาดประเภทนี้จะถูกส่งไปยัง Coordinator เพื่อจะกำหนดจุด Endpoint ที่ไม่สามารถทำการรับ Context ที่ถูกส่งมาได้

คุณสมบัติ

[Code] Sender

[Subcode] wscor: ContextRefused

[Reason] The coordination context that was provided could not be accepted

[Detail] unspecified

### Already Registered

สำหรับความผิดพลาดประเภทนี้จะถูกส่งไปยัง Participant ถ้า coordinator ตรวจพบว่า participant พยายามที่จะ Register สำหรับ protocol ตัวเดียวกันให้กับ activity ตัวเดียวกันมากกว่า 1 ครั้ง

คุณสมบัติ

[Code] Sender

[Subcode] wscor: AlreadyRegistered

[Reason] The participant has already registered for the same protocol

[Detail] unspecified

#### 4.2.6 Security Model

จุดประสงค์หลักของ Security โดยอ้างอิงถึง WS-Coordination คือ

- ให้แน่ใจว่ามีเพียงขั้นตอนที่ได้รับการอนุมัติแล้วทำการสร้าง Coordination context
- ให้แน่ใจว่ามีเพียงขั้นตอนที่ได้รับการอนุมัติแล้วทำการ Register เข้ากับ Activity
- ให้แน่ใจว่ามีเพียง Coordination Context ที่ถูกต้องตามกฎหมายทำการ Register
- ทำให้การทำงานของ Security ที่มีอยู่มีประสิทธิภาพมากขึ้น
- อนุญาตให้มีการจำกัดสิทธิ์โดยอ้างอิงถึง federated identities

จุดประสงค์หลักเหล่านี้ถูกกำหนดขึ้นมาจากความต้องการของ Security เพื่อความ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถูกต้องของข้อมูล การรักษาความลับ และการแสดงความเป็นเจ้าของ ซึ่งความต้องการแต่ละอย่างจะถูกจัดทำให้โดยองค์กรณ์ที่สร้างข้อกำหนดเกี่ยวกับ Web Service Security เช่น WS-Security และ WS-Trust

### 4.3 WS-Atomic Transaction

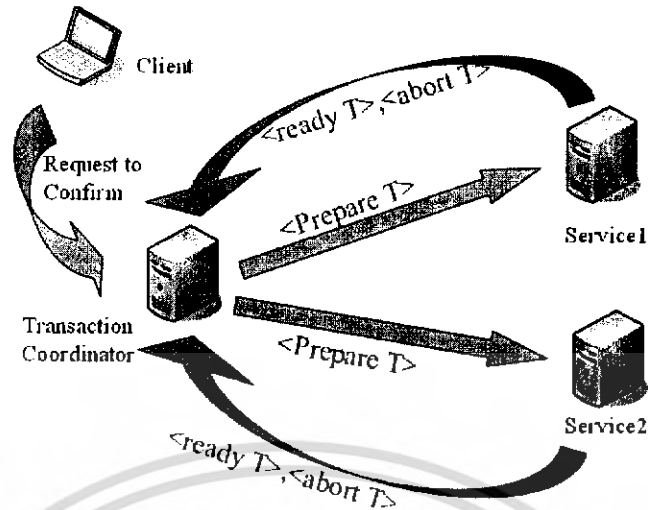
WS-Atomic Transaction จะอธิบายถึง Atomic transaction ซึ่งเป็น Coordination Type แบบหนึ่งที่สามารถนำมาใช้ใน โครงร่างที่อธิบายโดย WS-Coordination สำหรับใน WS-Atomic Transaction นี้จะทำการอธิบายถึง Coordination Protocol 3 ข้อตกลงที่ใช้สำหรับ Atomic transaction coordination type คือ Completion, Volatile two-phase commit และ Durable two-phase commit โดยที่เราสามารถนำเอา Protocol ตัวใดตัวหนึ่ง หรือทั้งหมดก็ได้มาใช้เมื่อเราต้องการสร้างแอปพลิเคชัน ที่ต้องการข้อตกลงของผลลัพธ์ที่ได้จาก Activity ที่ใช้เวลาสั้น และมีคุณสมบัติ all-or-nothing (ถ้าสำเร็จต้องสำเร็จทั้งหมด ถ้าไม่สำเร็จต้องไม่สำเร็จทั้งหมด)

#### 4.3.1 การcommit แบบสองระยะ (Two-Phase Commit Protocol)

การCommit แบบสองระยะเป็นกลไกที่จะช่วยให้การทำงานของแอปพลิเคชันแบบกระจายสามารถทำการ Commit Transaction ร่วมกันได้ และถ้ามีความผิดพลาดใดๆเกิดขึ้น Transaction นั้นๆจะต้องเหมือนกับไม่เคยเกิดขึ้นมาก่อน (Rollback) ซึ่งเป็นการช่วยให้คงความถูกต้องของข้อมูล การCommit แบบสองระยะจะประกอบไปด้วยการทำงานสองระยะ ดังนี้

##### ระยะที่ 1

- Coordinator ทำการส่ง <Prepare T> ไปให้กับเซอร์วิสที่เกี่ยวข้องใน Transaction ทั้งหมด
- จากนั้นเซอร์วิสจะทำการดูว่างานที่ได้รับมานั้นสามารถทำงานได้สำเร็จหรือไม่
  - a. ถ้าสำเร็จเซอร์วิสจะทำการส่ง <Ready T> ไปให้กับ Coordinator
  - b. ถ้าไม่สำเร็จเซอร์วิสจะทำการส่ง <Abort T> ไปให้กับ Coordinator
- Coordinator รอจนกระทั่งได้รับเมสเซจจากทุกๆเซอร์วิสครบแล้ว



รูปที่ 4.11 แสดงการทำงานของ การ Commit สองระยะในระยะเวลาที่หนึ่ง

### ระยะที่ 2

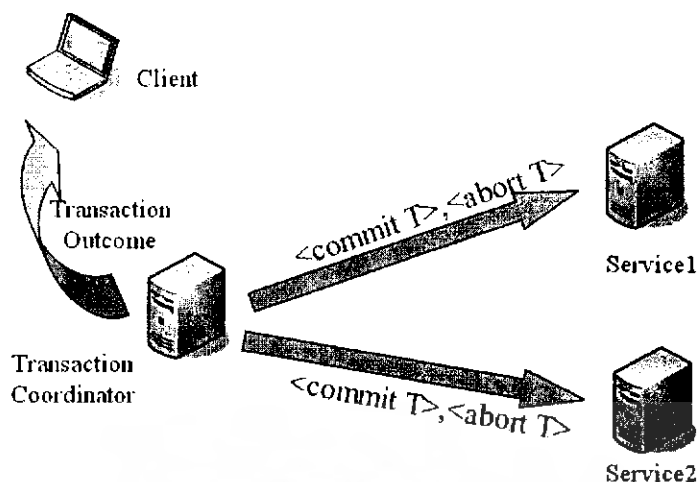
ถ้าทุกๆ เมชเซจที่ Coordinator ได้รับคือ <Ready T> แล้ว

- Coordinator ทำการเขียน <Commit> ลงในไฟล์บันทึก
  - Coordinator ทำการส่ง <Commit T> ไปให้กับเซอร์วิสทุกตัวที่เกี่ยวข้อง
- ใน Transaction
- เซอร์วิสที่เกี่ยวข้องทุกตัวทำการ Commit งานของตัวเองจากนั้นทำการคืนทรัพยากรทั้งหมดที่ใช้ให้กับระบบ

- เซอร์วิสทุกตัวทำการตอบรับการ Commit ไปยัง Coordinator
- Coordinator สิ้นสุดการทำงานใน Transaction
- ถ้า Coordinator ได้รับ <Abort T> จากเซอร์วิสตัวใดตัวหนึ่ง
- Coordinator ทำการเขียน <Rollback> ลงในไฟล์บันทึก
- Coordinator ทำการส่ง <Rollback T> ไปให้กับเซอร์วิสทุกตัวที่

เกี่ยวข้องกับ Transaction

- เมื่อเซอร์วิสได้รับ <Rollback T> จะทำการ Rollback งานที่ทำของตนแล้วตอบรับกลับไปยัง Coordinator
- Coordinator สิ้นสุดการทำงานใน Transaction



รูปที่ 4.12 แสดงการทำงานของ การ Commit สองระยะในระยะเวลาที่สอง

#### 4.3.2 Atomic Transaction Context

Atomic Transaction สร้างขึ้นมาเพื่อทำการอธิบายถึง Coordination Type ที่ได้รับการกำหนดไว้บน WS-Coordination

Atomic Transaction Coordination Context ต้องส่งผ่านอยู่ในทุก แอปพลิเคชัน Message ที่เกี่ยวข้องกับ Transaction นอกจากนั้นยังมีการเพิ่มความหมายของขั้นตอน CreateCoordinationContext ในส่วนของ Activation Service

- ถ้า request มีการรวมเอาส่วนของ CurrentContext Coordinator จะทำการแทรกเข้าไปเหมือนกับ subordinate ให้กับ coordinator ใน CurrentContext
- ถ้า request ไม่มีการรวมเอาส่วนของ CurrentContext Coordinator จะทำการสร้าง Transaction ใหม่ที่จะทำหน้าที่เหมือนเป็น root

Coordination Context อาจจะมี Expires attribute ประกอบอยู่ด้วยซึ่งเป็นการกำหนดเวลาที่เร็วที่สุดที่ Transaction อาจจะถูก terminate ได้จากการใช้เวลานานเมื่อถึงจุดนั้นแล้ว Transaction Manager อาจจะมีการทำการเลือกว่าจะทำการ roll back transaction นั้นหรือไม่ ถ้ายังไม่มีการ Transmit, Commit หรือ Prepare notification

Atomic Transaction protocol ถูกระบุโดย Coordination Type:

<http://schemas.xmlsoap.org/ws/2004/10/wsat>

#### 4.3.3 Atomic Transaction Protocols

ในส่วนของ WS-AtomicTransaction จะทำการระบุถึง Protocol ต่อไปนี้สำหรับการทำ Atomic Transaction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Completion: protocol นี้จะทำการเริ่มต้นกระบวนการในการ commit โดยอ้างจากแต่ละ protocol ที่ถูก register โดย participants ซึ่ง Coordinator จะเริ่มต้นด้วย Volatile 2PC จากนั้นจะเข้าไปสู่ Durable 2PC โดยผลลัพธ์สุดท้ายจะถูกส่งให้กับ initiator
  - Two-Phase Commit (2PC): 2PC Protocol จะกำหนดให้ participant จะทำการ commit หรือว่า abort และให้แน่ใจว่า participant ทุกตัวได้รับการแจ้งผลลัพธ์สุดท้าย โดย 2PC protocol ประกอบไปด้วย
    - Volatile 2PC: Participant ที่มีการจัดการกับ Volatile resource เช่น Cache ควรจะทำการ register เข้ากับ Protocol
    - Durable 2PC: Participant ที่มีการจัดการกับ Durable resource เช่น Database ควรจะทำการ register เข้ากับ Protocol
- Participant สามารถทำการ register กับ Protocol ได้มากกว่า 1 protocol ซึ่งทำได้ โดยการส่ง Register message หลายๆ message

#### 4.3.4 Preconditions

ในการที่ Protocol จะทำงานได้ตามขั้นตอนอย่างถูกต้องจำเป็นต้องมีการทำงานก่อนที่จะเริ่มต้นการประมวลผลดังนี้

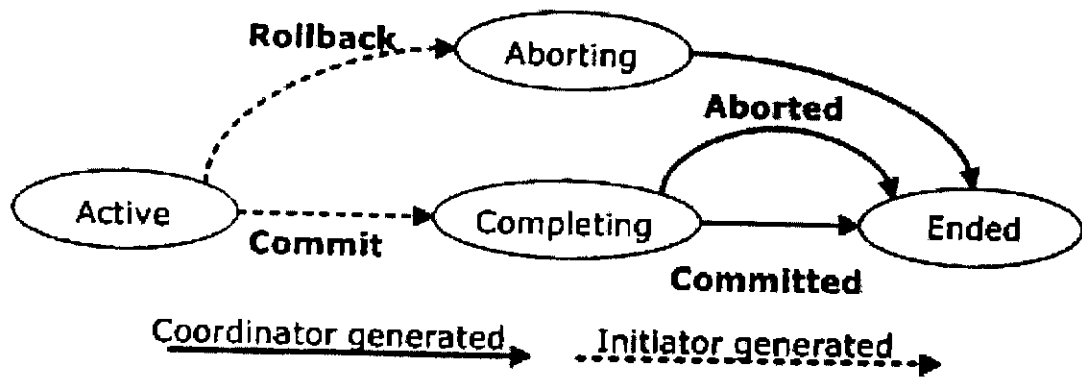
- Source ต้องรู้ว่ากฎของ destination เป็นอย่างไร และถ้ามี source ต้องมีความสามารถในการที่จะสร้าง message ให้สอดคล้องตามกฎของ destination
- ถ้ามีความต้องการของ secure exchange of message ดังนั้น source และ destination ต้องมี Security Context

#### 4.3.5 Completion Protocol

Completion Protocol ถูกใช้โดย แอปพลิเคชัน เพื่อทำการบอก coordinator ว่าจะพยายาม Commit หรือจะ Abort Atomic Transaction หลังจากที่ transaction นั้นทำงานจนเสร็จแล้วสถานะของมันจะถูกส่งกลับมาให้ยัง แอปพลิเคชัน

Initiator register ของ Protocol นี้ใช้ Protocol identifier ดังนี้:

<http://schemas.xmlsoap.org/ws/2004/10/wsat/Completion>



รูปที่ 4.13 แสดงภาพรวมของ Completion Protocol

Coordinator accepts:

- Commit - หลังจากที่ได้รับแจ้งมาทำให้ Coordinator รู้ว่า Participant มีความสามารถในการที่จะทำงานให้เสร็จ และมันควรพยายามที่จะ Commit Transaction นั้น
- Rollback - หลังจากที่ได้รับแจ้งมา Coordinator รู้ว่า Participant ได้ทำการ terminate การประมวลผลของการทำงานใน แอปพลิเคชัน นั้น และมันควรที่จะทำการ Abort Transaction

Initiator accepts:

- Committed - หลังจากที่ได้รับแจ้งมา Initiator รู้ว่า coordinator ถึงเวลาที่ ต้องทำการ Commit transaction
- Aborted - หลังจากที่ได้รับแจ้งมา Initiator รู้ว่า Coordinator ถึงเวลาที่ ต้องทำการ abort transaction

#### 4.3.6 ลักษณะของ Two-Phase Commit Protocol ใน WS-AtomicTransaction

Two-Phase Commit Protocol (2PC) เป็น Coordination Protocol ที่ทำการกำหนดว่า Participants หลายตัวจะทำงานร่วมกันตามข้อตกลงของผลลัพธ์ใน Atomic Transaction ได้ อย่างไร 2PC Protocol แบ่งออกได้เป็น Volatile 2PC และ Durable 2PC

##### Volatile Two-Phase Commit Protocol

หลังจากที่ได้รับแจ้ง Commit มาในส่วนของ Completion Protocol จากนั้น Root Coordinator จะทำการเตรียมพร้อมให้ทุกๆ participant ทำการ register

สำหรับ Volatile 2PC Protocol ทุกๆ Participant ที่ถูก register เข้า Protocol (Volatile 2PC) นี้จะต้องทำการ respond ก่อนที่จะ Register เข้ากับ Durable 2PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Volatile Recipient จะไม่มีการรับรองว่าจะได้รับการแจ้งบอกเมื่อมีผลลัพธ์ของ Transaction

Participant ที่ register กับ Volatile 2PC สามารถใช้ Protocol identifier

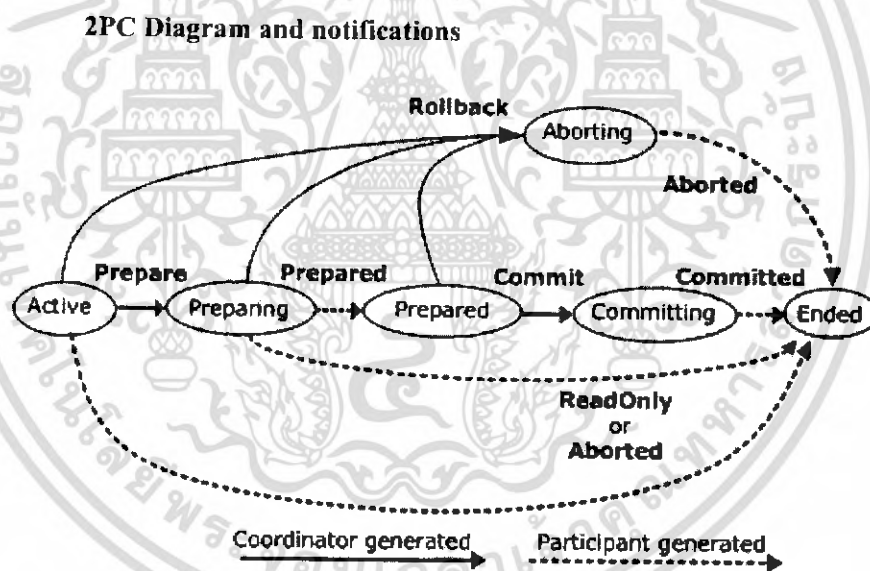
ดังนี้ : <http://schemas.xmlsoap.org/ws/2004/10/wsac/Volatile2PC>

### Durable Two-Phase Commit Protocol

หลังจากที่ได้รับการแจ้ง Commit ในส่วนของ Completion Protocol และเสร็จสิ้นจากการเตรียมสถานะของ Volatile 2PC แล้ว Root Coordinator จะเริ่มทำการเตรียมสำหรับให้ Participant Register เข้าสู่ Durable 2PC Protocol โดยจะต้องมีการ respond ว่า Prepare หรือ ReadOnly ก่อนที่จะมีการแจ้ง Commit ให้กับ participant เพื่อให้มีการ register กับ protocol ตัวใดตัวหนึ่ง

Participant ที่ register กับ Durable 2PC สามารถใช้ Protocol identifier

ดังนี้ : <http://schemas.xmlsoap.org/ws/2004/10/wsac/Durable2PC>



รูปที่ 4.14 แสดงภาพรวมของการ Commit แบบสองระยะ

*Participant accepts:*

#### Prepare

หลังจากที่ได้รับการแจ้งมา Participant จะรู้ว่าต้องเข้าสู่ Phase 1 และทำการ vote ให้กับผลลัพธ์ของ Transaction ถ้า Participant ไม่รู้เกี่ยวกับ Transaction มันต้องทำการ abort แต่ถ้า Participant นั้นได้ทำการ vote ไปแล้วมันควร จะทำการ resend ค่าที่เคย vote ให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Rollback

หลังจากที่ได้รับการแจ้งมา Participant จะรู้ว่าต้องทำการ abort และ ทำการทิ้ง Transaction นั้นไป การแจ้งนี้สามารถส่งได้ทั้ง phase 1 และ phase 2 เมื่อมีการส่งแล้ว coordinator จะไม่สนใจ และ clear ที่ทั้งหมดของ Transaction

### Commit

หลังจากที่ได้รับการแจ้งมา Participant จะรู้ว่าต้องทำการ commit Transaction โดยการแจ้งนี้จะสามารถส่งได้หลังจาก phase 1 และ Participant ทำการ vote commit ถ้า participant ไม่รู้เกี่ยวกับ Transaction มันต้องทำการส่งการแจ้งว่า committed ให้กับ Coordinator

*Coordinator accepts:*

### Prepared

หลังจากที่ได้รับการแจ้งมา Coordinator จะรู้ว่า participant ได้ถูกเตรียม และทำการ commit Transaction

### ReadOnly

หลังจากที่ได้รับการแจ้งมา Coordinator จะรู้ว่า participant ได้ทำการ commit Transaction แล้ว และไม่สนใจใน Transaction อีก Participant ไม่ต้องการที่จะเข้ามามีส่วนร่วมใน phase 2

### Aborted

หลังจากที่ได้รับการแจ้งมา Coordinator จะรู้ว่า participant ได้ทำการ abort แล้ว และไม่สนใจใน Transaction อีก

### Committed

หลังจากที่ได้รับการแจ้งมา Coordinator จะรู้ว่า Participant ได้ทำการ committed transaction แล้วทำให้ Participant สามารถไม่ต้องสนใจกับ Transaction อีก

### Replay

หลังจากที่ได้รับการแจ้งมา Coordinator จะทำการสรุปได้ว่า Participant มีความผิดพลาดขึ้นซึ่งไม่สามารถทำการกู้คืนได้ Coordinator จึงควรที่จะ resend Protocol ที่เหมาะสมล่าสุด

## 4.3.7 Transaction Faults

WS-AtomicTransaction fault ต้องมีการ include เอา [action] จาก URI:

<http://schemas.xmlsoap.org/ws/2004/10/wsat/fault>

Faults จะเน้นไปที่ destination endpoint ซึ่งเกี่ยวข้องกับกฎของการรองรับการเกิด faults ซึ่งได้รับการระบุไว้ใน WS-Addressing

Faults ในส่วนนี้จะถูกระบุไว้ด้วยคุณสมบัติดังนี้

[Code] Fault code

[Subcode] Fault subcode

[Reason] เหตุผลของการเกิด fault (ภาษาอังกฤษ)

[Detail] รายละเอียดต่างๆ ถ้าว่างคือไม่มีรายละเอียดของความผิดพลาด

#### InconsistentInternalState

Fault ประเภทนี้จะถูกส่งโดย Participant เพื่อทำการบอกว่าไม่สามารถทำตามวัตถุประสงค์ได้ ซึ่งจะเป็นการบอกถึง Global consistency failure และเป็นสถานะที่ไม่สามารถทำการกู้คืนได้

[Code] Sender

[Subcode] wsat: InconsistentInternalState

[Reason] A global consistency failure has occurred. This is an unrecoverable condition.

[Detail] unspecified

## บทที่ 5

### Business Process Execution Language

#### 5.1 ความหมายและหลักการ

ในปัจจุบันเว็บเซอร์วิสได้เข้ามามีบทบาทมากขึ้น และทุกวันนี้เราจะเขียน โปรแกรม โดยไปเรียกเซอร์วิสอื่นๆมาช่วยซึ่งช่วยให้ผู้เขียนโปรแกรมสามารถลดภาระที่จะต้องเขียน โปรแกรมได้ แต่ในโลกของธุรกิจมีการแข่งขันตลอดเวลาจึงมีความจำเป็นต้องเปลี่ยนแปลงตลอดเวลา จึงต้องแก้ไขโค้ดโปรแกรมที่ใช้เขียนเซอร์วิสให้เหมาะสม จึงเกิดความยุ่งยากในการเขียน โปรแกรมซึ่งแก้ไขโดยการสร้างตัวกลางขึ้นมาซึ่งเรียกว่า BPEL โดย BPEL จะมองตัวมันเองเป็นเหมือนเซอร์วิสตัวหนึ่งแต่ภายในเซอร์วิสนั้นจะแสดงขั้นตอนกระบวนการทำงานของระบบธุรกิจ ซึ่งช่วยให้นักวิเคราะห์ธุรกิจสามารถแก้ไขได้ง่าย และเนื่องจากbpel ถูกเขียนมาในลักษณะแบบกราฟจึงสามารถแก้ไขเปลี่ยนแปลงได้โดยง่ายซึ่งช่วยให้ผู้เขียน โปรแกรมเขียนโปรแกรมได้ง่ายขึ้น

Business Process Execution Language คือ ภาษาที่ใช้ในการอธิบายความหมายและการดำเนินการทางกระบวนการทางธุรกิจ โดยจะอยู่บนสุดของสถาปัตยกรรมเว็บเซอร์วิส bpel ใช้ xml เป็นภาษาพื้นฐานที่สนับสนุนการเทคโนโลยีเว็บเซอร์วิส รวมไปถึง soap, uddi, wsdl bpel ถูกสร้างมาจากเทคโนโลยีที่สำคัญของทั้ง 2 ค่ายได้แก่ Microsoft และ IBM คือ xlang, wsfl ซึ่งได้นำข้อดีของทั้ง 2 เทคโนโลยีมารวมกันคือนำข้อดีของการเขียน โปรแกรมแบบโฟลว์ และการเขียนโปรแกรมเป็นบล็อก มารวมกันโดยภาษานี้จะสนับสนุนการใช้งาน 2 แบบ คือ

- Abstract เป็นโปรโตคอลที่ใช้ในการติดต่อกระบวนการทางธุรกิจโดยจะระบุรูปแบบการติดต่อของข้อความ
- Executable ระบุถึงรายละเอียดการทำงานในกระบวนการทางธุรกิจ

Bpel เกี่ยวข้องกับ มาตรฐานเว็บเซอร์วิสตัวอื่นๆ โดย Bpel เกี่ยวข้องกับ ws -transaction และ ws -coordination โดยขึ้นอยู่กับ wsdl และทั้งหมดนี้ขึ้นอยู่กับ xml และ xsd

Choreography - CDL4WS			Business Processes
Orchestration - BPEL4WS			
WS-Reliability	WS-Security	Transaction	Quality of Service
		Coordination	
		Context	
UDDI			Discovery
WSDL			Description
SOAP			Message
XML			
HTTP, IIOP, JMS, SMTP			Transport

รูปที่ 5.1 แสดง WS-Architecture

จากรูปที่ 5.1 ใน Layer ของ Business Process จะมีอยู่ 2 ส่วน ด้วยกัน คือ CDL4WS กับ BPEL4WS โดย BPEL จะอยู่ในส่วนของ Orchestration

Orchestration เป็นการติดต่อภายในกระบวนการทางธุรกิจเดียวกัน

Choreography เป็นการติดต่อระหว่างกระบวนการทางธุรกิจ

Bpel ประกอบด้วยส่วนประกอบต่างๆที่สำคัญคือ

- BPEL file - BPEL เป็นภาษาที่ใช้กำหนด business process ซึ่งจริงๆ แล้ว เป็นภาษา xml ลักษณะของ BPEL คือ เป็น procedural language คล้ายกับ flow chart ทำหน้าที่กำหนดว่าจะเรียก WS ไหน, เมื่อไหร่ และอาจเก็บตัวแปรด้วย  
การทำงานจะไปข้างหน้าเรื่อยๆ จนจบไฟล์ (ลองนึกถึง file build.xml ของ Ant อาจเข้าใจ การทำงานคล้ายอย่างนั้นเลย)
- BPEL engine คือ ตัวที่จะมาอ่าน BPEL ที่เราเขียน และ สร้าง composite WS ให้ทำงานตามที่กำหนดใน BPEL นั้นเอง  
client ที่เรียกใช้จะเห็น composite WS ที่สร้างขึ้นเป็นเหมือน WS ทั่วไป

## 5.2 โครงสร้างของ Bpel

Bpel ประกอบด้วยขั้นตอนต่างๆ โดยแต่ละขั้นตอนจะเรียกว่า activity โดยจะแบ่งได้ออกเป็น 2 ประเภทคือ

Primitive จะเป็น activity ขั้นพื้นฐานที่ใช้ใน activities มีอยู่ด้วยกันหลายตัวด้วยกันคือ

Invoke นำเว็บเซอร์วิสตัวอื่นมาใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<invoke partner Link="ncname"
  port Type="qname"
  operation="ncname"
  input Variable="ncname"
  output Variable="ncname">
</invoke>

```

Receive รอ โคล์เอนท์ที่ invoke ส่งข้อความมาให้

```

<receive partner Link="ncname"
  port Type="qname"
  operation="ncname"
  variable="ncname"
  createInstance="yes|no">
</receive>

```

Reply สร้างข้อความตอบกลับแบบในเวลาเดียวกัน

```

<reply partnerLink="ncname"
  portType="qname"
  operation="ncname"
  variable="ncname"
  faultName="qname">
</reply>

```

Wait รอเวลาช่วงหนึ่ง

```
<wait (for="duration-expr" | until="deadline-expr") />
```

Structured ใช้ในการอธิบายโครงสร้างของกระบวนการทางธุรกิจ

Sequence ทำการจัดลำดับ activity

```
<Sequence>
```

```
  activities
```

```
</sequence>
```

Switch ใช้ในการสร้างเงื่อนไขของกระบวนการ

```
<Switch>
```

```
<case condition="bool-expr">
```

```
  activity
```

```
</case>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<otherwise>
  activity
</otherwise>
</switch>
While เป็นการวนเงื่อนไขในลูปนั้นจนกว่าจะเป็นจริง
  <while condition="bool-expr">
    activity
  </while>

```

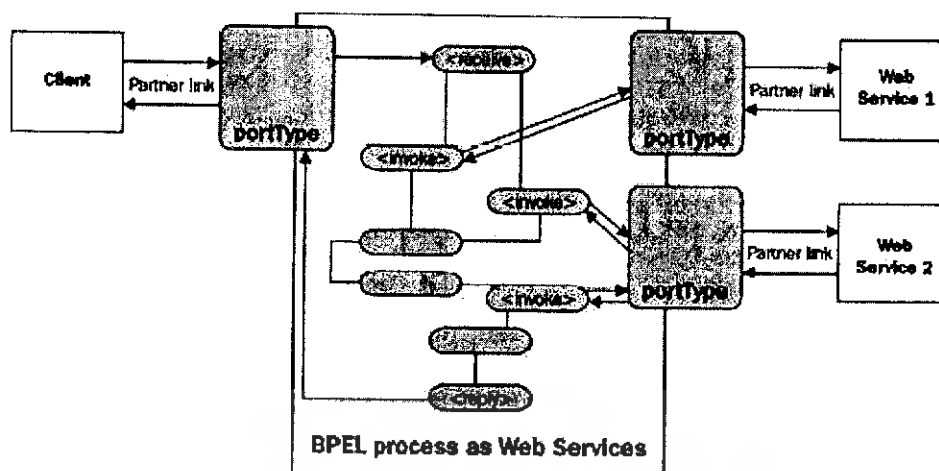
Pick เป็นวิธีการที่จะใช้เลือกเส้นทางใดเส้นทางหนึ่ง

```

  pick create Instance="yes|no">
    <onMessage partnerLink="nicknamed"
      portType="qname"
      operation="ncname"
      variable="ncname">
      activity
    </onMessage>
    <onAlarm (for="duration-expr" | until="deadline-expr")>
      activity
    </onAlarm>
  </pick>

```

Bpel สามารถใช้ได้แบบมีการจับสัญญาณาฬิกา และ ไม่มีการจับแบบสัญญาณาฬิกา Bpel แบบมีการจับสัญญาณาฬิกาจะทำการบล็อกไคล์เอนท์จนกว่ากระบวนการนั้นจะเสร็จสิ้น และคืนค่ากลับสู่ไคล์เอนท์นั้นไป Bpel แบบไม่มีการจับสัญญาณาฬิกาจะไม่บล็อกไคล์เอนท์จนกว่าจะมีคำสั่งจากไคล์เอนท์ไปอีกที โดยปรกติเราจะใช้ Bpel แบบไม่มีการจับสัญญาณาฬิกาในกระบวนการที่มีระยะเวลาที่ยาวและใช้ Bpel แบบจับสัญญาณาฬิกาในความสัมพันธ์ระยะสั้นในการที่จะใช้ Bpel นั้น web service ที่เข้ามาติดต่อกับจะทำการติดต่อผ่าน port Type ดังรูปข้างล่าง



รูปที่ 5.2 แสดงโครงสร้างของ BPEL

### 5.3 การติดต่อกับเว็บเซอร์วิส

ในกระบวนการของ Bpel ได้มีการติดต่อกับเว็บเซอร์วิสได้ 2 แบบ คือ

- กระบวนการ Bpel ไปเชื่อมต่อกับงานอื่นๆผ่านทางเว็บเซอร์วิส
- กระบวนการ Bpel รับข้อมูลจากไคลเอนท์โดย 1 ใน ไคลเอนท์คือผู้ใช้

Bpel ได้เรียกสิ่งๆที่ไปเชื่อมต่อกับเซอร์วิสอื่นๆว่า partner links โดย partner links สามารถเชื่อมไปยังเว็บเซอร์วิสต่างๆที่เชื่อมต่อกัน โดยกระบวนการของ Bpel นั้นจะมี 1 ไคลเอนท์ partner links เป็นอย่างน้อย โดยเหตุผลที่ Bpel ทำสิ่งๆที่ไคลเอนท์เป็น partner links นั้นมีอยู่ 2 เหตุผลด้วยกันคือ สามารถสนับสนุนการเชื่อมต่อแบบไม่จับสัญญาณนาฬิกา และ เหตุผลอีกข้อนั้นสนับสนุนความจริงที่ว่า Bpel นั้นต้องมีเซอร์วิสเข้ามาเกี่ยวข้องจะเข้ามาทางPortType โดยสามารถแยกแยะความแตกต่างที่เข้ามาได้เช่นในระบบประกันภัย สามารถแยกได้ว่าอันไหนเป็นประกันรถ อันไหนเป็นประกันชีวิต

ในภาพรวมนั้นเราสามารถสรุปได้ว่า Partner links นั้นจะใช้อธิบายลิงค์ของ Partner ขณะที่ Partner อาจจะเป็น เซอร์วิสที่เกี่ยวข้องกับกระบวนการ เซอร์วิสที่ถูกเกี่ยวข้องในกระบวนการ เซอร์วิสที่เป็นได้ทั้ง 2 บทบาท ตัวอย่างการนำ bpel ไปใช้

เพื่อให้เกิดความเข้าใจในการนำ Bpel ไปใช้งานในกระบวนการทางธุรกิจโดยจะยกตัวอย่างง่ายคือในระบบประกันเราจะใช้ Bpel มาเพื่อช่วยในการตรวจสอบว่าราคาค่าประกันของบริษัทประกันประเภทไหนราคาถูกกว่ากัน

ประกันประเภทที่ 1 InsuranceA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประกันประเภทที่ 2 InsuranceB

```
<?xml version='1.0' encoding='utf-8'?>
<process name='insuranceSelectionProcess'
targetNamespace='http://packtpub.com/bpel/example/'
xmlns='http://schemas.xmlsoap.org/ws/2003/05/business-process/'
xmlns:ins='http://packtpub.com/bpel/insurance/'
xmlns:com='http://packtpub.com/bpel/company/' >

<partnerLinks>

<partnerLink name='client'
partnerLinkType='com:selectionLT'
myRole='insuranceSelectionService' />

<partnerLink name='insuranceA'
partnerLinkType='ins:insuranceLT'
myRole='insuranceRequester'
partnerRole='insuranceService' />

<partnerLink name='insuranceB'
partnerLinkType='ins:insuranceLT'
myRole='insuranceRequester'
partnerRole='insuranceService' />

</partnerLinks>
```

จากนั้นเราจะประกาศตัวแปรในการรีเควส Insurance Request และ response ของทั้ง 2 Insurance Response. Insurance Response และคำตอบในการเลือก InsuranceSelectionResponse

```
<variables>
<!-- Input for BPFL process -->
<variable name='insuranceRequest'
messageType='ins:insuranceRequestMessage' />
<!-- output from Insurance A -->
<variable name='insuranceAResponse'
messageType='ins:insuranceResponseMessage' />
<!-- output from Insurance B -->
<variable name='insuranceBResponse'
messageType='ins:insuranceResponseMessage' />
<!-- output from BPFL process -->
<variable name='insuranceSelectionResponse'
messageType='ins:insuranceResponseMessage' />
</variables>
```

ท้ายที่สุดเราจะระบุขั้นตอนการทำงานของกระบวนการนี้ ขึ้นแรกเราจะรีเควสจากไคลเอนท์เข้ามา จากนั้น เราจะทำการเชื่อมต่อกับเซอร์วิสประกันภัยทั้ง 2 ซึ่งจะทำการเชื่อมต่อแบบขนาน จากนั้น จะทำการตรวจสอบราคาประกันที่ถูกที่สุดและส่งค่ากลับมายังไคลเอนท์

```
<sequence>
<!-- Receive the initial request from client -->
<receive partnerLink='client'
portType='com:insuranceSelectionPT'
operation='SelectInsurance'
variable='insuranceRequest'
createInstance='yes' />

<!-- Make concurrent invocations to Insurance A and B -->
<flow>

<!-- Invoke Insurance A web service -->
<invoke partnerLink='insuranceA'
portType='ins:computeInsurancePremiumPT'
operation='ComputeInsurancePremium'
inputVariable='insuranceRequest'
outputVariable='insuranceAResponse' />

<!-- Invoke Insurance B web service -->
<invoke partnerLink='insuranceB'
portType='ins:computeInsurancePremiumPT'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        operation="ComputeInsurancePremium"
        inputVariable="InsuranceRequest"
        outputVariable="InsuranceBResponse" />
    </flow>
    <!-- Select the best offer and construct the response -->
    <switch>
        <case condition="bpw:age(VariableData('InsuranceAResponse',
            'confirmationData',/confirmationData/Amount))
            <= bpw:age(VariableData('InsuranceBResponse',
            'confirmationData',/confirmationData/Amount))">
            <!-- Select Insurance A -->
            <assign>
                <copy>
                    <from variables="InsuranceAResponse" />
                    <to variable="InsuranceSelectionResponse" />
                </copy>
            </assign>
        </case>
        <otherwise>
            <!-- Select Insurance B -->
            <assign>
                <copy>
                    <from variables="InsuranceBResponse" />
                    <to variable="InsuranceSelectionResponse" />
                </copy>
            </assign>
        </otherwise>
    </switch>
    <!-- Send a response to the client -->
    <reply partnerLink="client"
        portType="com:InsuranceSelectionPT"
        operation="SelectInsurance"
        variable="InsuranceSelectionResponse" />
</sequence>
</process>

```

#### 5.4 เครื่องมือที่ใช้ในการทดลอง

ในการทดลองนี้เราจะทำการทดลองสร้าง BPEL ขึ้นมาโดยใช้เครื่องมือดังนี้

Microsoft Visio

Microsoft Visual Studio 2005

Microsoft SQL Server 2005

Microsoft orchestration Data Business Activity

Microsoft .Net Framework 3.0

Microsoft Windows Workflow Extension

โดยในการทดลองนี้เราจะใช้ Windows Workflow Foundation เป็นตัวที่รันเอนจินของ BPEL ขึ้นมา โดย Window Workflow Foundation เป็นหนึ่งในสถาปัตยกรรมของ Microsoft .Net Framework 3.0 ซึ่งจะมีอยู่ 3 ส่วนด้วยกัน คือ

1. Window Communication Foundation คือเฟรมเวิร์คที่ใช้ในการพัฒนาเว็บเซิร์ฟวิส โดยหน้าที่หลักคือจะทำตัวเป็นตัวกลางสื่อสารระหว่างเว็บเซอร์วิสหรือเทคโนโลยีอื่นๆ ที่ใช้ร่วมกันกับเว็บเซอร์วิส

2. Window Workflow Foundation คือเฟรมเวิร์คที่ทำหน้าที่ในการควบคุมการทำงานของโปรแกรม การทำงานระหว่างโปรแกรมที่เกี่ยวข้องและในโปรเจกต์นี้เราได้นำมาใช้ในส่วนของการควบคุมการทำงานของเว็บเซอร์วิสให้เป็นไปตามโครงสร้างที่เรากำหนดไว้ โดยลักษณะเด่นของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Window Workflow Foundation คือสามารถกำหนด นโยบาย (Policy) ที่ใช้ได้ โดยเรานำนโยบายมาใช้ในส่วนของการทำโปรโมชัน คือเราสามารถกำหนดรายการโปรโมชันไปได้เลยว่าจะให้รายการสินค้าใดบ้างที่มีการจัดโปรโมชันหรือ ทำเป็นแพ็คเกจเนื่องจาก Window Workflow Foundation สามารถนำ BPEL มาใช้ได้จึงทำให้เราสามารถปรับปรุงแก้รายการได้ตลอดเวลา

3. Window Presentation Foundation เป็นส่วนที่เอามาใช้ในการแสดงผลของการทำงานในโปรแกรม ซึ่งในโครงการนี้ไม่ได้ใช้ เนื่องจากเราเน้นการศึกษาไปที่ WCF กับ WWF



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

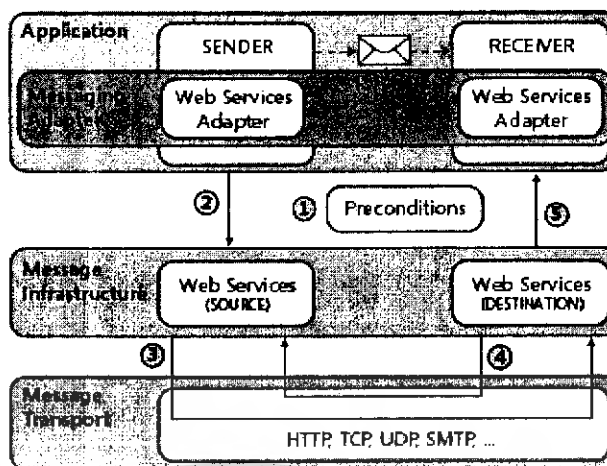
### WS-Reliable Messaging

ในการทำงานของเว็บเซอร์วิส การส่ง 메시지를ผ่านเครือข่ายสามารถเกิดข้อผิดพลาดที่มาขัดขวางการรับ-ส่งข้อความระหว่างเซอร์วิสได้หลายชนิดอาจทำให้เกิดการสูญหายของข้อความระหว่างการรับ-ส่งหรือการส่งข้อความซ้ำเป็นต้น การเกิดความผิดพลาดในการรับ-ส่งข้อความระหว่างเซอร์วิสอาจทำให้เว็บเซอร์วิสทำงานล้มเหลวหรืออาจทำให้การทำงานของเว็บเซอร์วิสผิดพลาดได้ เพื่อแก้ปัญหาในกรณีดังกล่าวจึงมีการกำหนดโปรโตคอลขึ้นมาเพื่อใช้ในการติดต่อระหว่างเว็บเซอร์วิสเพื่อให้การสื่อสารระหว่างเว็บเซอร์วิสมีความน่าเชื่อถือมากขึ้น

WS-ReliableMessaging คือโปรโตคอลที่สร้างขึ้นมาเพื่อทำให้การสื่อสารระหว่างเว็บเซอร์วิสมีความน่าเชื่อถือแบบ เอ็น-ทู-เอ็น โดย WS-ReliableMessaging สร้างขึ้นจากการรับรองการรับส่งข้อความ 4 ข้อคือ

1. แต่ละข้อความทำการส่งได้อย่างมากหนึ่งครั้ง ฝ่ายรับจะได้รับข้อความเพียงครั้งเดียว โดยไม่มีการได้รับข้อความซ้ำ อาจเกิดการสูญหายของข้อความจากการทำการส่งได้
2. แต่ละข้อความทำการส่งได้น้อยหนึ่งครั้ง ฝ่ายรับอาจได้รับข้อความซ้ำได้แต่ต้องมีการส่งข้อความที่สำเร็จอย่างน้อยหนึ่งครั้ง
3. แต่ละข้อความทำการส่งได้ในครั้งเดียว การรับส่งข้อความจะทำได้สำเร็จภายในครั้งเดียวคือไม่มีความผิดพลาดหรือส่งเมสเสจซ้ำเกิดขึ้นเลย
4. ข้อความส่งถึงเรียงลำดับก่อนหลัง ฝ่ายรับจะได้ข้อความเรียงลำดับก่อนหลังโดยเรียงตามการส่งข้อความ

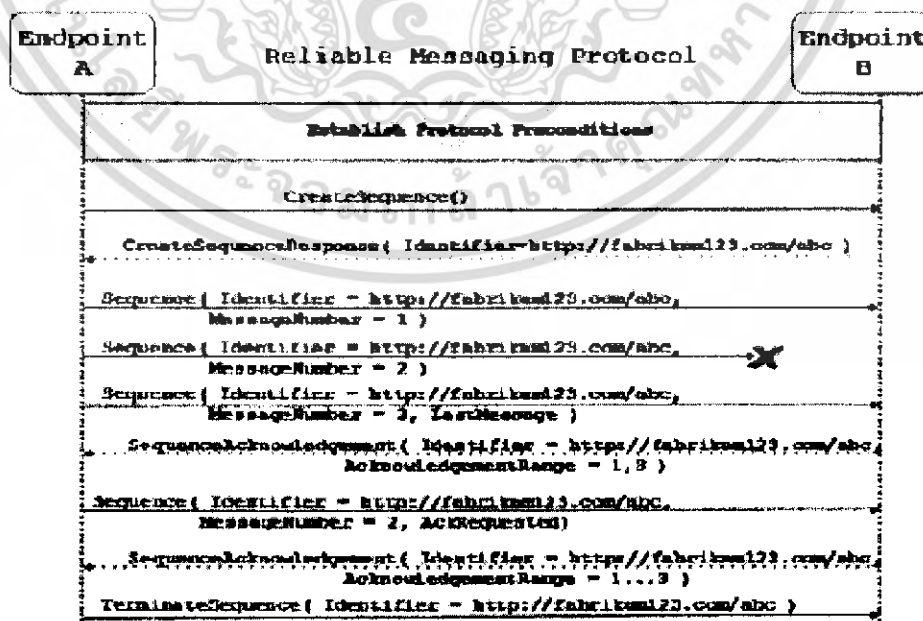
WS-ReliableMessaging ไม่ขึ้นอยู่กับทรานสปอร์ตโปรโตคอลที่ใช้ว่าจะใช้แบบใด เนื่องจากสถาปัตยกรรมของเว็บเซอร์วิสออกแบบมาเพื่อให้มีความอิสระในการใช้ทรานสปอร์ตโปรโตคอล เราอาจใช้โปรโตคอลเพียงชนิดเดียวหรืออาจใช้ร่วมกันก็ได้



รูปที่ 6.1 แสดงชั้นการทำงานตามข้อกำหนด WS-ReliableMessaging

จากรูปที่ 6.1 ในขั้นตอนที่ 1 จะเป็นการสร้างพรีคอนดิชันสำหรับการแลกเปลี่ยนเมสเสจ ระหว่าง SENDER และ RECEIVER. SENDER จะทำการส่งเมสเสจในชั้นที่ 2 ในชั้นเมสเสจอินฟราสตรัคเจอร์ เมสเสจจะทำการเปลี่ยนแปลงรูปแบบของเมสเสจให้เป็นไปตามชั้นเมสเสจทรานสปอร์ตแล้วทำการส่งในชั้นที่ 3 ในการรับส่งจะมีการตอบกลับของของ RECEIVER เพื่อจัดการในเรื่องข้อผิดพลาดที่เกิดในการรับส่งเมสเสจในชั้นที่ 4 และขั้นสุดท้าย เมสเสจอินฟราสตรัคเจอร์ฝั่ง RECEIVER จะทำการเปลี่ยนแปลงรูปแบบของเมสเสจให้เหมาะสมกับตัวแอปพลิเคชันส่งต่อไปในชั้นที่ 5

ตัวอย่างการรับส่งเมสเสจ



รูปที่ 6.2 ตัวอย่างการส่งเมสเสจด้วย WS-ReliableMessaging โปรโตคอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 6.2 เป็นการส่งเมสเสจระหว่างสองแอปพลิเคชัน โดยให้แอปพลิเคชันที่ทำการส่งเมสเสจมีเอ็นพอยคือ เอ็นพอยท์ A ส่วนแอปพลิเคชันที่รับเมสเสจมีเอ็นพอยท์คือเอ็นพอยท์ B แอปพลิเคชัน A ต้องการส่งเมสเสจเรียงลำดับกัน 3 เมสเสจ ในการเริ่มการติดต่อ แอปพลิเคชันที่ทำการส่งเมสเสจจะทำการสร้างพรีคอนดิชันขึ้นมาระหว่างเอ็นพอยท์ทั้งสอง หลังจากนั้นฝั่งฝ่ายส่งจะทำการส่งเมสเสจไปที่ฝ่ายรับเพื่อร้องขอที่จะสร้างลำดับในการส่งเมสเสจ ฝ่ายรับจะทำการตอบกลับ แล้วฝ่ายส่งจะเริ่มทำการส่งโดยเรียงลำดับที่ละเมสเสจ ในกรณีที่เมสเสจเกิดสูญหายในการส่ง ฝ่ายส่งจะทำการจะทราบได้จากเมสเสจที่ฝ่ายรับตอบกลับและจะทำการส่งใหม่อีกครั้ง

การส่งเมสเสจระหว่างเอ็นพอยท์ทั้งสองกระที่ชั้นเมสเสจอินฟราสตรักเจอร์ โดยที่ผู้พัฒนาแอปพลิเคชัน ไม่จำเป็นต้องจัดการด้วยตัวเอง

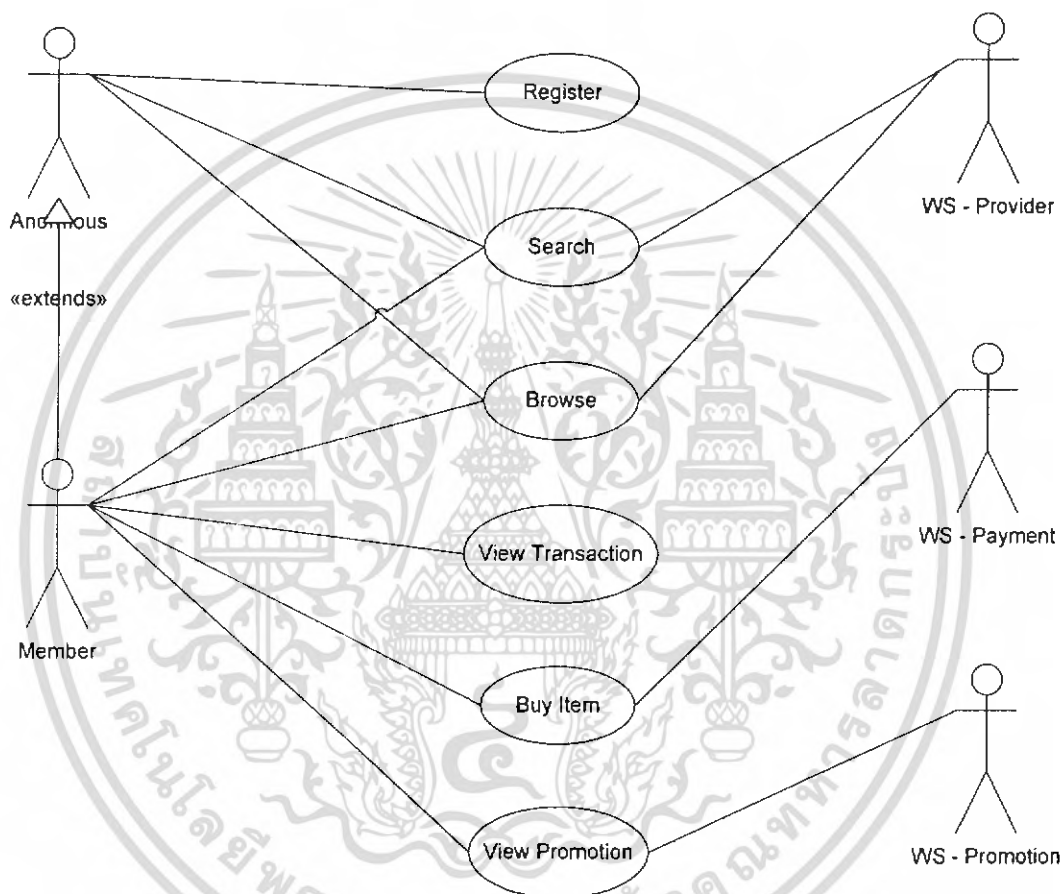


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 7

### แบบจำลองระบบซื้อขายสื่ออิเล็กทรอนิกส์ผ่านเว็บไซต์

#### 7.1 Use Case Diagram

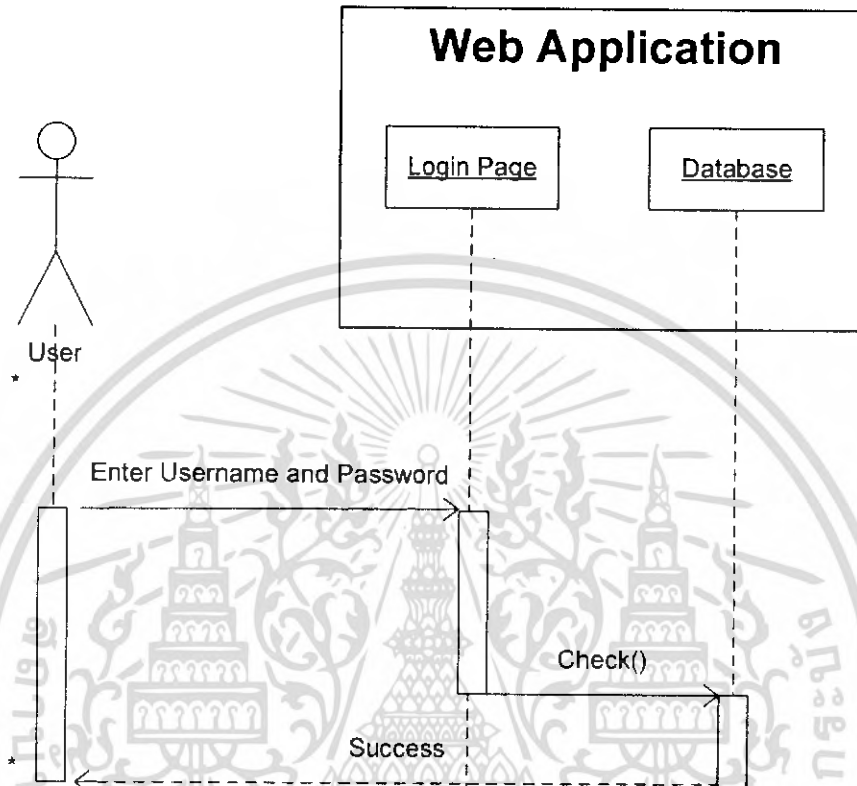


รูปที่ 7.1 แสดง Use case diagram ภาพรวมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 7.2 Sequence Diagram

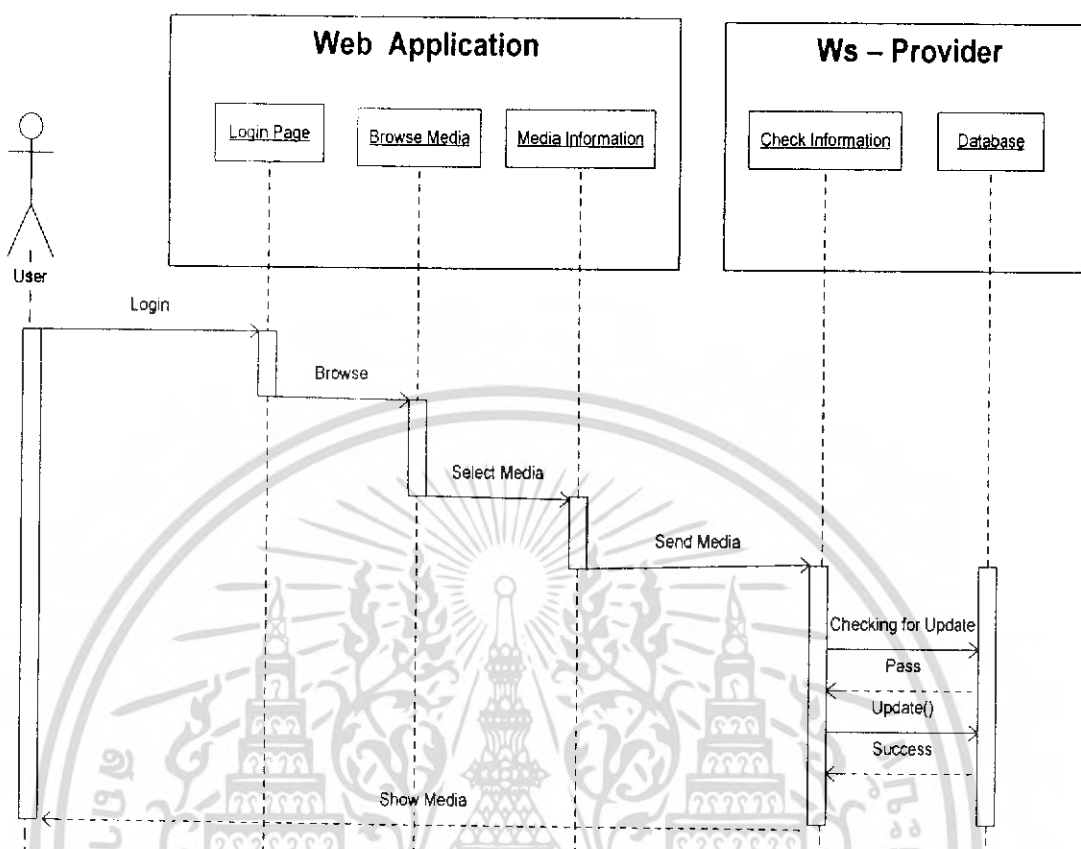
### Sequence Diagram :Login



รูปที่ 7.2 แสดง Sequence Diagram การล็อกอินของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

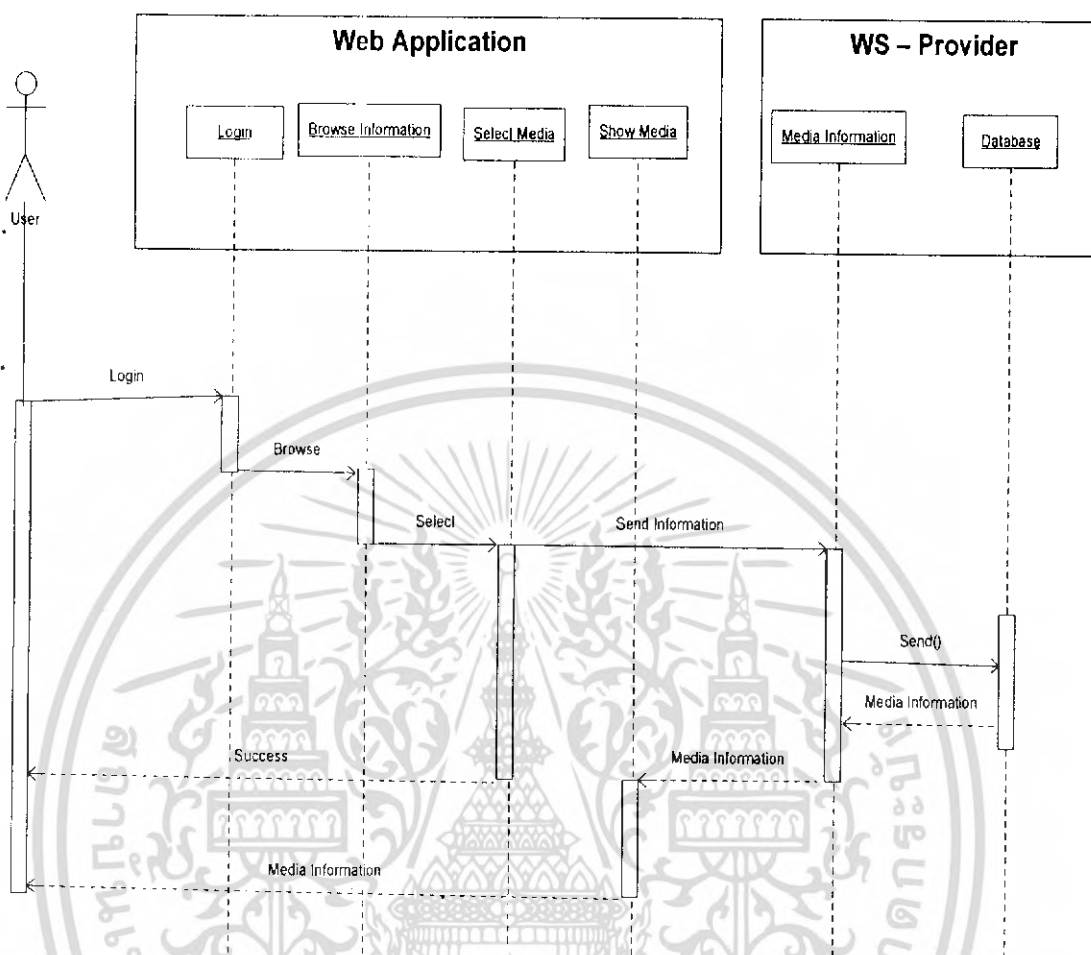
## Select Media Entertain



รูปที่ 7.3 แสดง Sequence Diagram ในการเลือกไอเท็ม ของระบบ

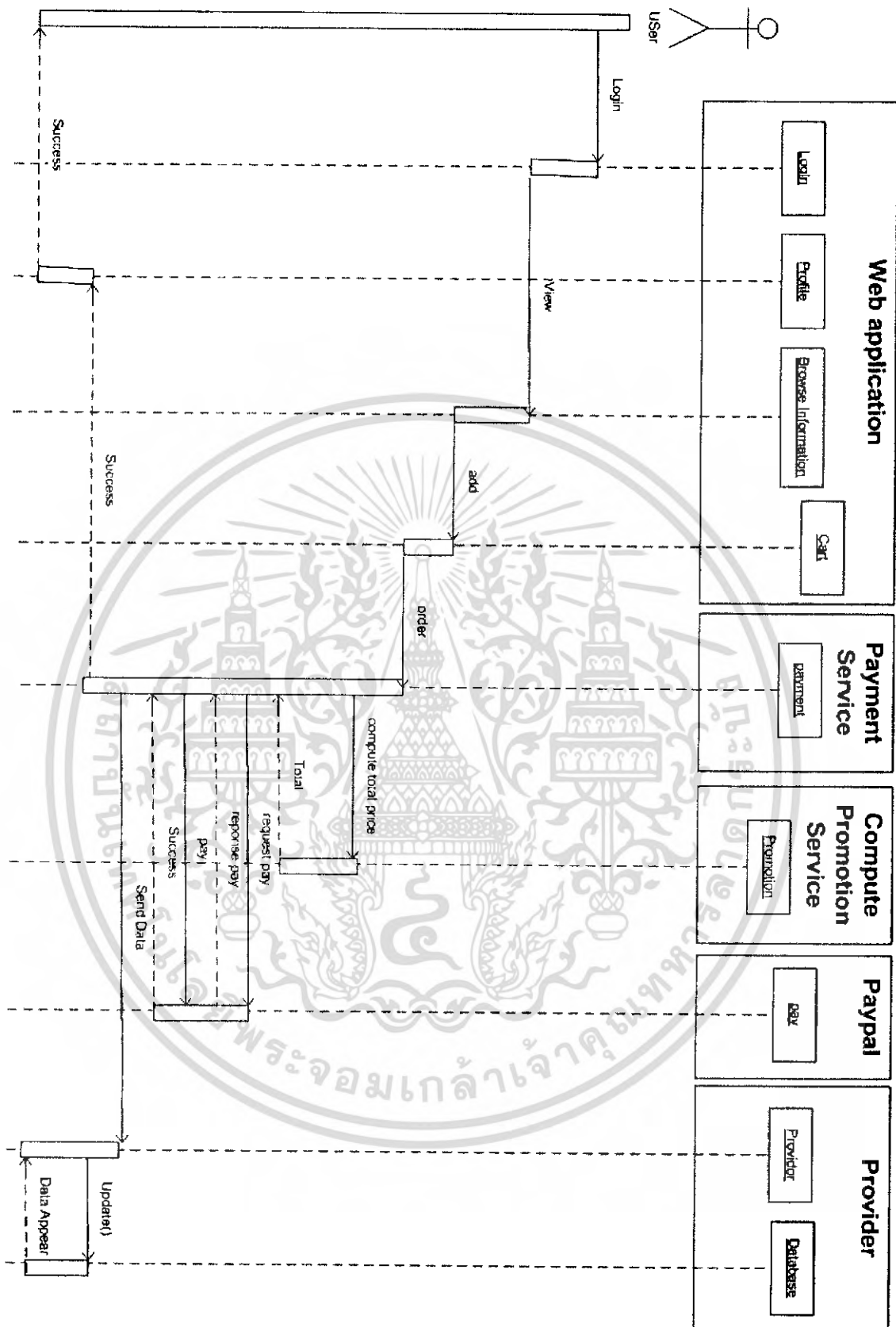
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sequence Diagram : Select Content Media



รูปที่ 7.4 แสดง Sequence Diagram แสดงไอเท็มที่ได้ในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.5 แสดง Sequence Diagram ของการจ่ายเงิน ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 7.3 Activity Diagram

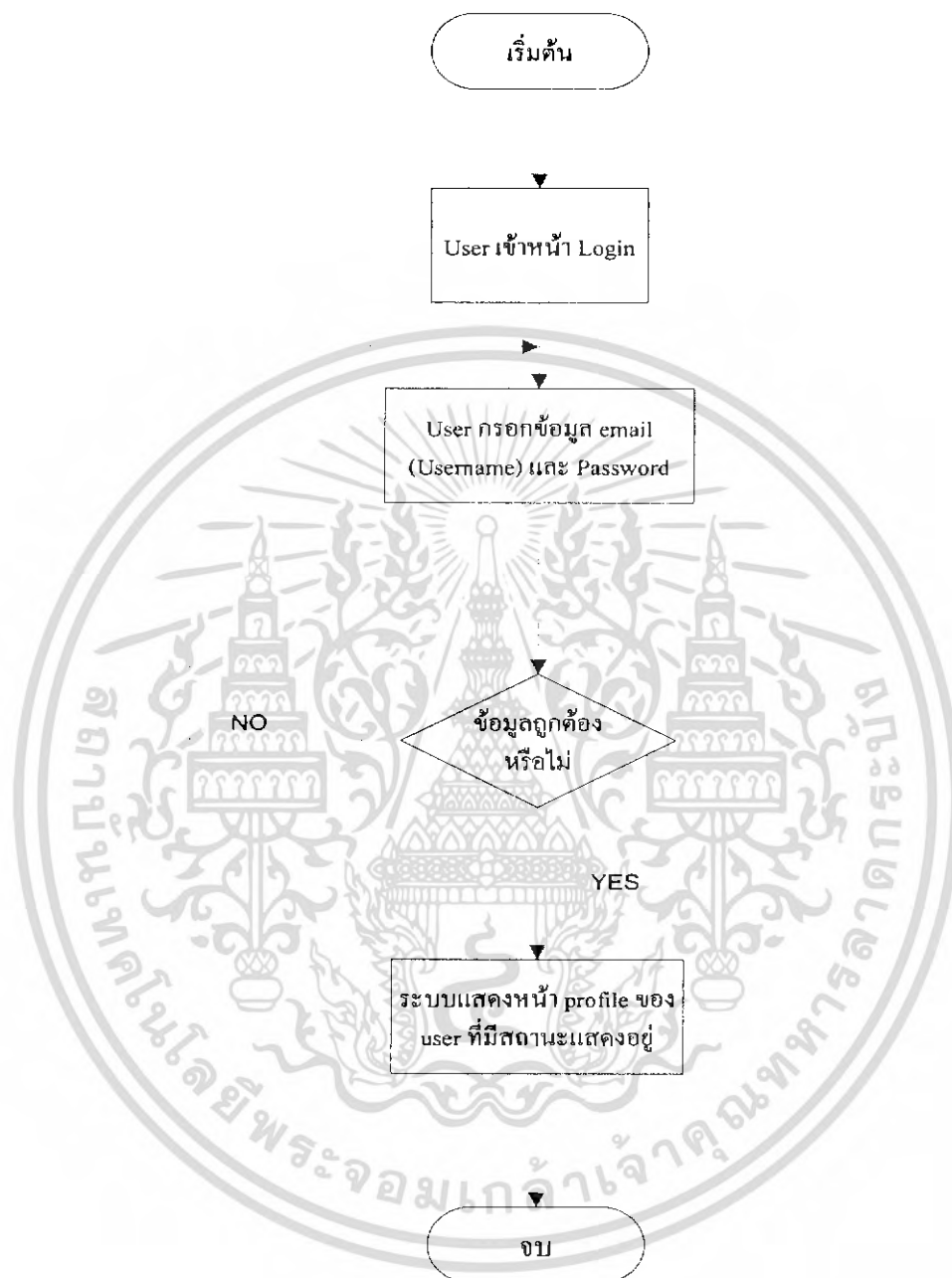
## Activity Diagram :LOGIN



รูปที่ 7.6 แสดง Activity Diagram ของการล็อกอิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## VIEW Content

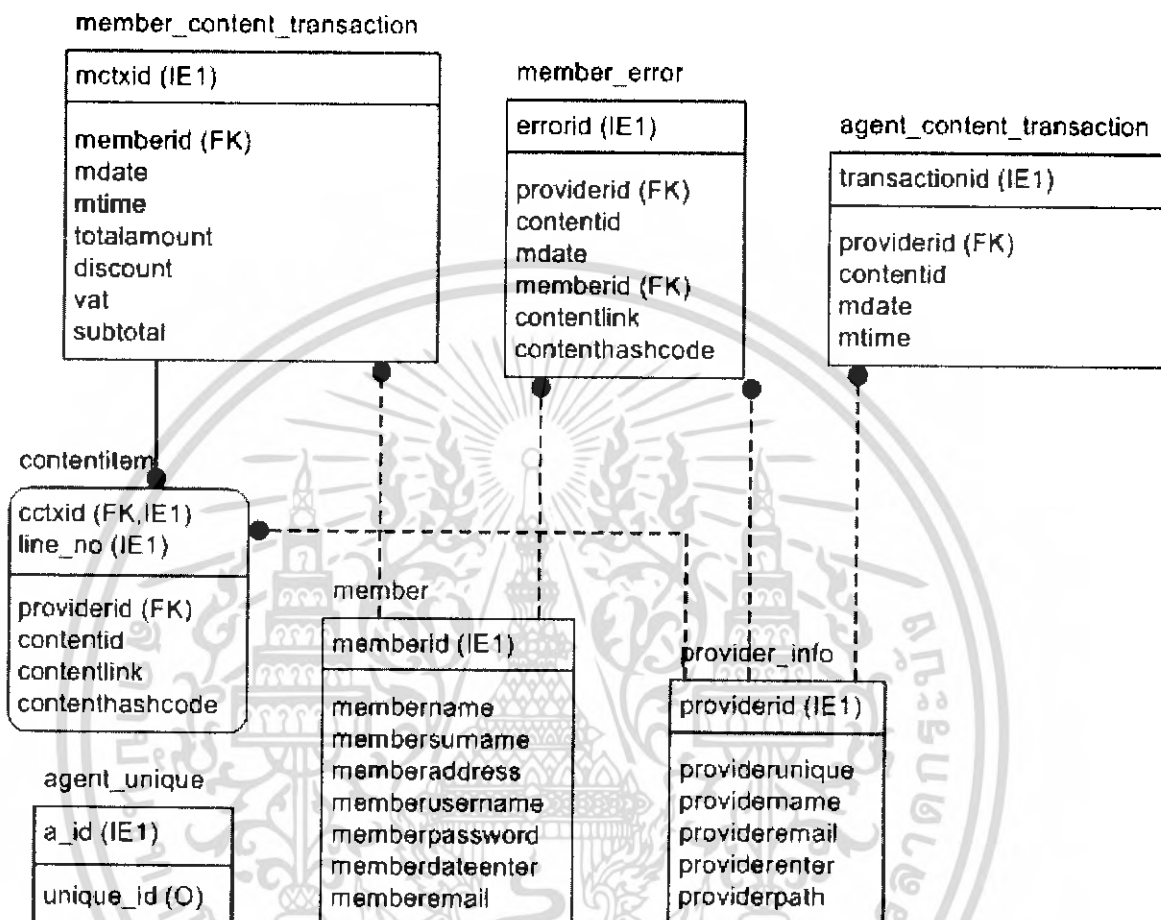


รูปที่ 7.7 แสดง Activity Diagram การดูรายละเอียดของไอเท็มในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 7.4 Database design

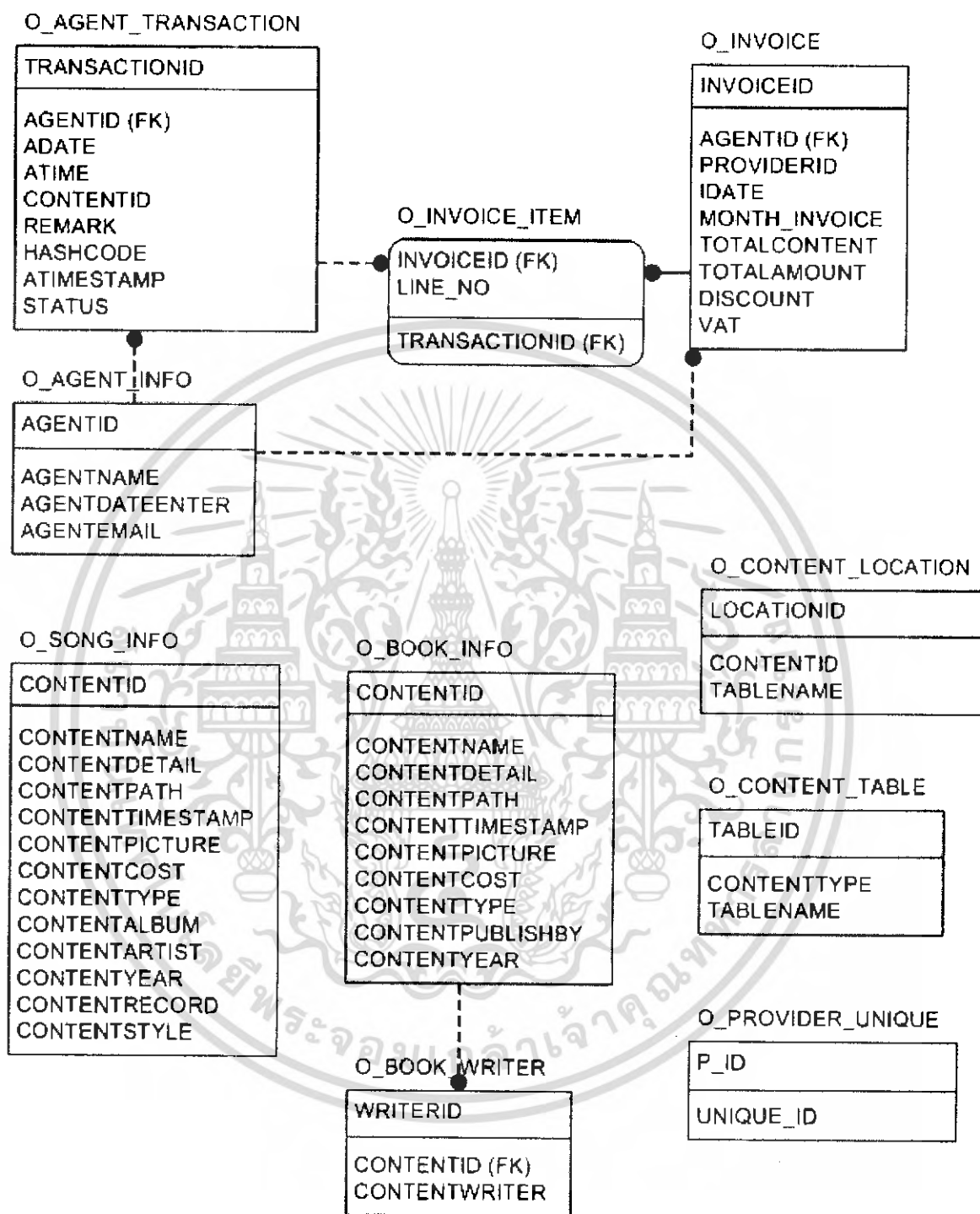
### 7.4.1 Agent database design



รูปที่ 7.8 แสดง Database Design ของทาง agent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

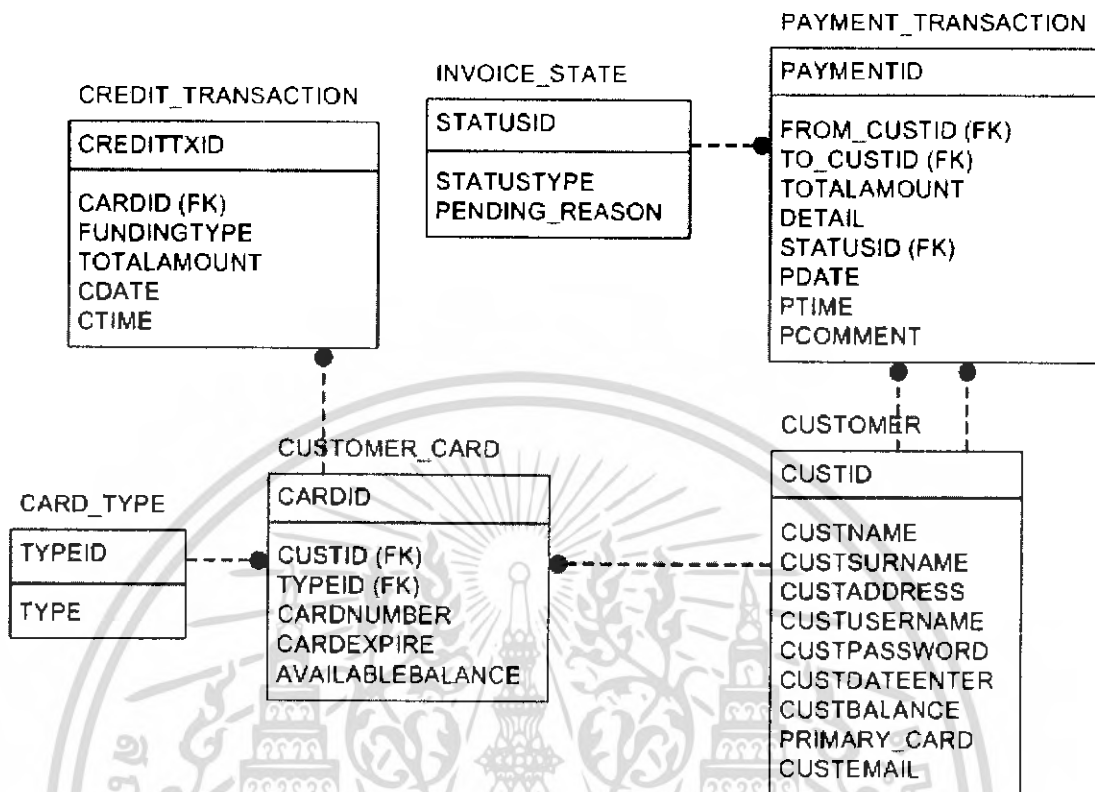
## 7.4.2 Provider database design



รูปที่ 7.9 แสดง Database Design ของทาง Provider

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 7.4.3 Payment database design



รูปที่ 7.10 แสดง Database Design ของทาง Payment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

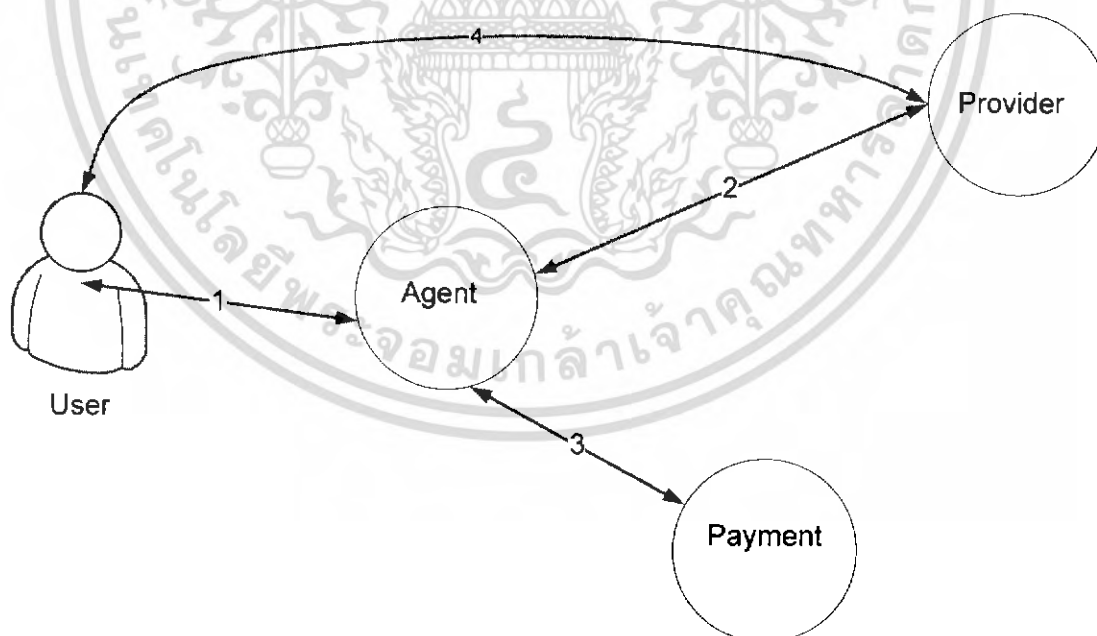
## 7.5 Security Model

เนื่องจากระบบมีการทำธุรกรรมผ่านทางเครือข่ายอินเทอร์เน็ต ความปลอดภัยของระบบจึงมีความสำคัญอย่างยิ่ง ทั้งนี้เพื่อเป็นการปกป้องลูกค้าตลอดจนสร้างความมั่นใจให้กับผู้ใช้บริการและผู้ร่วมธุรกิจ ในหัวข้อนี้จะกล่าวถึงความต้องการในด้านความปลอดภัยในจุดต่างๆ และเทคโนโลยีในการรักษาความปลอดภัยที่เลือกใช้โดยจะเน้นไปที่การยืนยันตนเอง ซึ่งมีความสำคัญอย่างยิ่งในการดำเนินธุรกรรม

### 7.5.1 ความต้องการความปลอดภัย

ระบบบริการคาว์โพลด์สื่ออิเล็กทรอนิกส์ประกอบไปด้วย 4 Entity คือ

1. **User** คือผู้ที่ทำการซื้อและคาว์โพลด์สื่ออิเล็กทรอนิกส์
2. **Agent** คือองค์กรที่เป็นตัวแทนจำหน่ายมีหน้าที่ในการจัดหาข้อมูลและทำการขายสื่ออิเล็กทรอนิกส์ของ Provider
3. **Payment** คือองค์กรที่ทำหน้าที่ในการเก็บข้อมูลทางการเงินและชำระเงินของ User
4. **Provider** คือผู้ที่เป็นเจ้าของสื่ออิเล็กทรอนิกส์มีหน้าที่ในการจัดหาข้อมูลและเป็นจัดหาสื่ออิเล็กทรอนิกส์ให้ User ทำการคาว์โพลด์สื่ออิเล็กทรอนิกส์



รูปที่ 7.11 แสดงความสัมพันธ์ของส่วนต่างๆของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 7.11 แสดงให้เห็นถึงการติดต่อระหว่าง Entity ต่างๆในระบบ ซึ่งสามารถแบ่งการติดต่อได้เป็น 4 จุดคือ

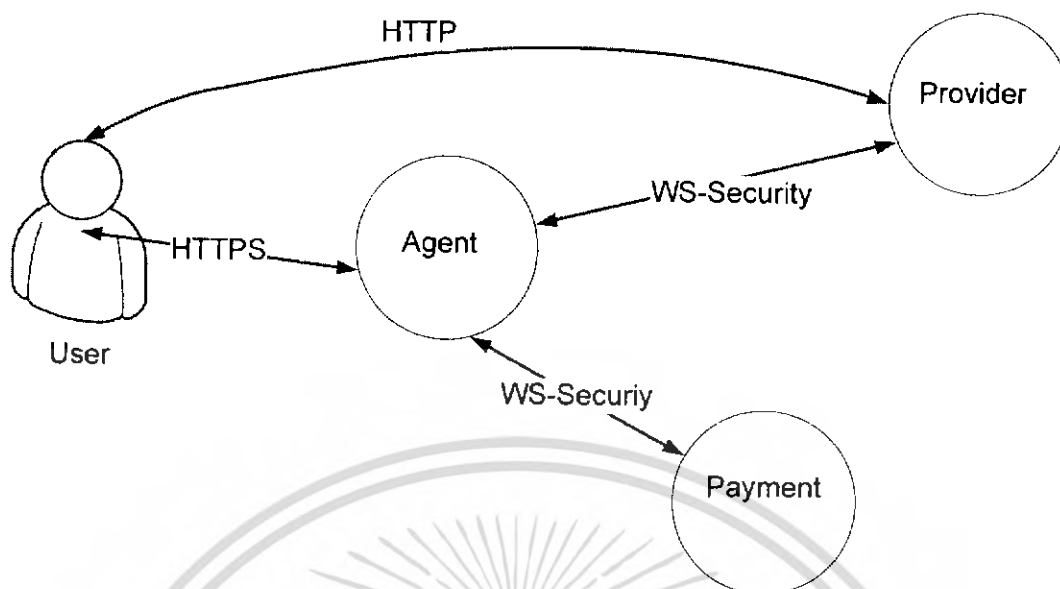
1. User ติดต่อกับ Agent
2. Agent ติดต่อกับ Provider
3. Agent ติดต่อกับ Payment
4. User ติดต่อกับ Provider

ความต้องการในด้าน Security ของแต่ละจุดและเทคโนโลยีที่เลือกใช้แสดงได้ในตารางต่อไปนี้

ตารางที่ 7.1 แสดงความต้องการในด้านความปลอดภัย

จุดเชื่อมต่อ	ความต้องการทาง Security	เทคโนโลยีหรือวิธีการที่เลือกใช้
1. User ติดต่อกับ Agent	<ul style="list-style-type: none"> <li>- การยืนยันตัวตนของ User ต่อ Agent</li> <li>- การยืนยันตัวตนของ Agent ต่อ User</li> <li>- การปกป้องความลับของข้อมูล ที่ User ส่งให้ Agent</li> </ul>	<ul style="list-style-type: none"> <li>- ใช้ Username และ Password ในการยืนยันตัวของ User</li> <li>- ใช้ SSL ในการยืนยันตัวตนของ Agent และเข้ารหัสข้อมูล</li> </ul>
2. Agent ติดต่อกับ Provider	<ul style="list-style-type: none"> <li>- การยืนยันตัวตนของ Agent ต่อ Provider</li> <li>- การยืนยันตัวตนของ Provider ต่อ Agent</li> </ul>	<ul style="list-style-type: none"> <li>- ใช้ WS-Security แบบ Mutual certificate ซึ่งใช้ certificate ในการยืนยันตัวเองของทั้งฝั่ง Agent และ Provider</li> </ul>
3. Agent ติดต่อกับ Payment	<ul style="list-style-type: none"> <li>- การยืนยันตัวตนของ Agent ต่อ Payment</li> <li>- การยืนยันตัวตนของ Payment ต่อ Agent</li> </ul>	<ul style="list-style-type: none"> <li>- ใช้ WS-Security แบบ Mutual certificate ซึ่งใช้ certificate ในการยืนยันตัวเองของทั้งฝั่ง Agent และ Payment</li> </ul>
4. User ติดต่อกับ Provider	<ul style="list-style-type: none"> <li>- การยืนยันสิทธิ์ของ User</li> </ul>	<ul style="list-style-type: none"> <li>- ใช้ key ซึ่งสร้างโดย Agent ในการยืนยันสิทธิ์ของ User</li> </ul>

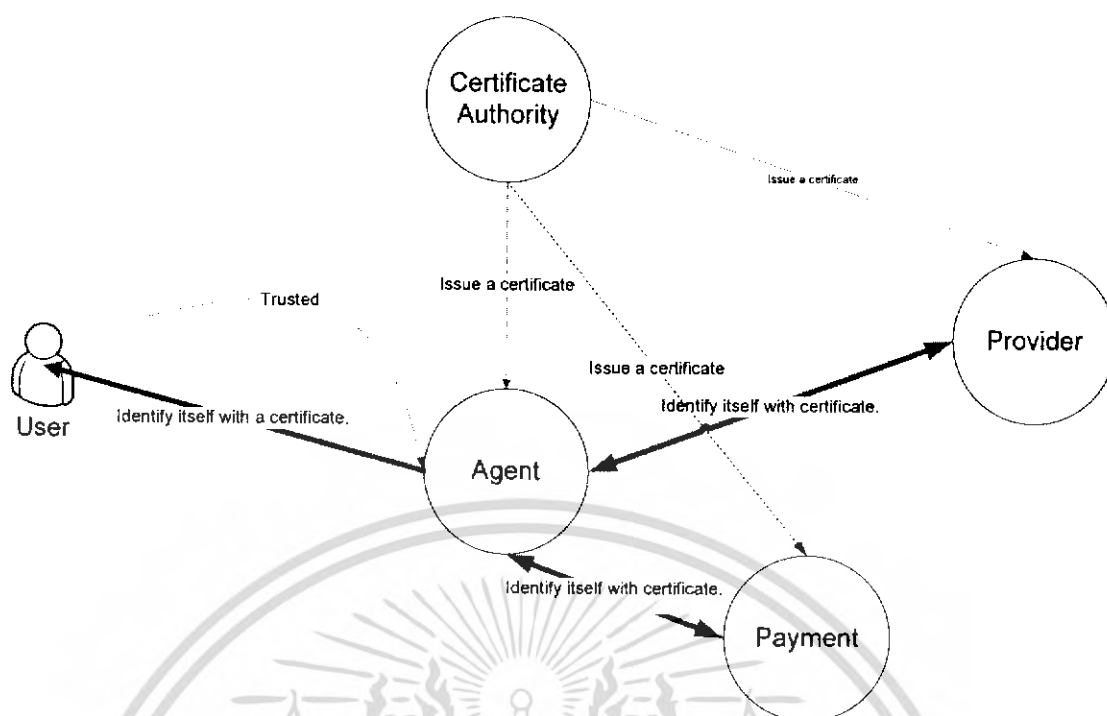
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.12 แสดง Protocol ที่ใช้ในการติดต่อของระบบ

#### 7.5.2 ความสัมพันธ์ระหว่าง Entity กับ Certificate Authority

เนื่องจากในระบบมีการใช้ Certificate ในการยืนยันตัวตนของส่วนต่างๆ ดังนั้นจึงมีการตั้ง Certificate Authority (CA) ขึ้นเพื่อใช้ในการจัดทำ certificate ให้กับส่วนต่างๆของระบบ ความสัมพันธ์ระหว่าง CA กับส่วนต่างๆของระบบแสดงได้ในรูปที่ 7.13



รูปที่ 7.13 แสดงความสัมพันธ์ระหว่าง CA กับส่วนต่างๆของระบบ

### 7.5.3 การปฏิบัติในการรักษาความปลอดภัย

การรักษาความปลอดภัยในระบบแบ่งออกเป็น 2 ขั้นตอนดังนี้

#### 7.5.3.1 การติดตั้ง

1. Provider Agent และ Payment ทำการขอ X.509 certificate จาก CA
2. Provider และ Payment ทำการแจกจ่าย Certificate ของตนเองให้แก่ Agent
3. User ทำการ Register

#### 7.5.3.2 การใช้งาน

##### การยืนยันตนเองของ Agent ต่อ Provider

1. Agent นำ key จาก certificate ของ Provider มารหัส SOAP message ที่ทำการส่ง
2. Agent ยืนยันตนเองด้วย certificate
3. Provider ตรวจสอบ certificate ของ Agent โดยการตรวจสอบ CA ของ Agent
4. ถ้า CA ของ Agent ได้รับการเชื่อถือ (trusted) ,provider จะทำการตรวจสอบสิทธิ์ของ Agent ว่ามีสิทธิ์ใช้หรือไม่

##### การยืนยันตนเองของ Provider ต่อ Agent

การยืนยันตนเองของ Provider จะถูกกระทำตั้งแต่ขั้นตอนการติดตั้งแล้วเนื่องจากต้องมีการ

นำ certificate ของ Provider มาไว้ที่ Agent ซึ่งจะมีการตรวจสอบแล้ว

เอกสารนี้เป็นเอกสารที่สงวนเวลาสำหรับการใช้งานเพื่อการศึกษาดูเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### การยืนยันตนเองของ Agent ต่อ Payment

1. Agent นำ key จาก certificate ของ Payment มารหัส SOAP message ที่ทำการส่ง
2. Agent ยืนยันตนเองด้วย certificate
3. Payment ตรวจสอบ certificate ของ Agent โดยการตรวจสอบ CA ของ Agent
4. ถ้า CA ของ Agent ได้รับการเชื่อถือ (trusted), payment จะทำการตรวจสอบสิทธิ์ของ Agent ว่ามีสิทธิ์ใช้หรือไม่

### การยืนยันตนเองของ Payment ต่อ Agent

การยืนยันตนเองของ Payment จะถูกกระทำตั้งแต่ขั้นตอนการติดตั้งแล้วเนื่องจากต้องมีการนำ certificate ของ Payment มาไว้ที่ Agent ซึ่งจะได้รับการตรวจสอบแล้ว

### การยืนยันตนเองของ User ต่อ Agent

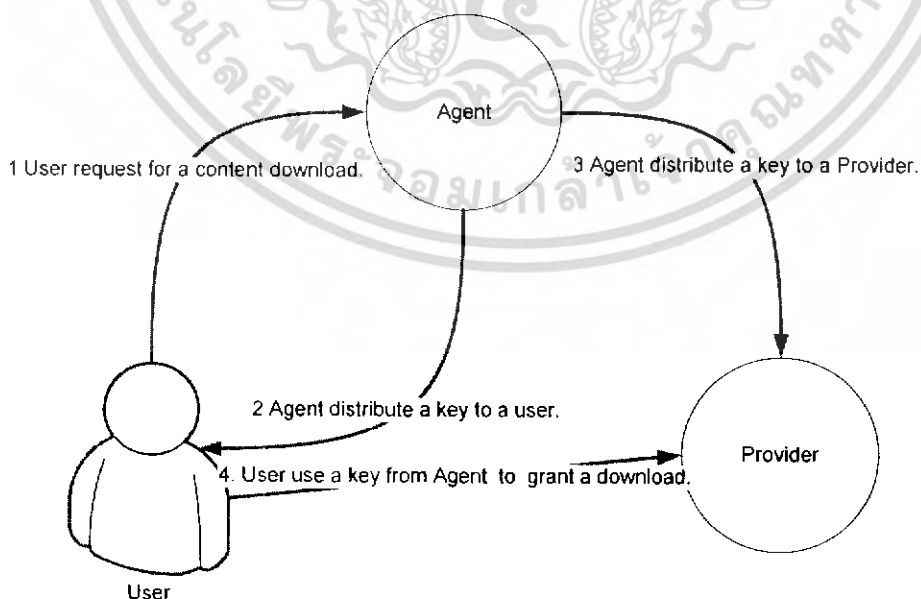
1. User ยืนยันตนเองโดยใช้ Username และ Password ( User ต้องทำการ Register ก่อนนั้นแล้ว )
2. Agent ทำการตรวจสอบ username และ password ว่าเป็น user ตัวจริงหรือไม่

### การยืนยันตนเองของ Agent ต่อ User

1. Agent แสดง certificate ของตนเอง User
2. User ทำการตรวจสอบ certificate ของ Agent

### การยืนยันตนเองของ User ต่อ Provider

การยืนยันตนเองของ User ต่อ Provider แสดงโดยรูปต่อไปนี้



รูปที่ 7.14 แสดงวิธีการในการยืนยันสิทธิ์ในการดาวน์โหลดของ User ต่อ Provider

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 7.14 สามารถอธิบายตามขั้นตอนได้ดังต่อไปนี้

1. User ทำการสั่งซื้อสื่ออิเล็กทรอนิกส์จาก Agent โดยในขั้นตอนนี้จะถือว่าการร้องขอการ download จาก Agent
2. ในขั้นที่ 2 และ 3 Agent ทำการสร้าง key แล้วแจกจ่ายให้กับ User และ Provider ที่เป็นเจ้าของ สื่อที่ User ต้องการทำการดาวน์โหลด
3. User ทำการดาวน์โหลดโดยส่ง key ที่ได้จาก Agent ให้ Provider
4. Provider ทำการตรวจสอบ key ที่ได้จาก user ว่าได้รับอนุญาตให้ดาวน์โหลดได้หรือไม่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 8

### การทดลอง และผลการทดลอง

#### 8.1 ฮาร์ดแวร์ และซอฟต์แวร์ที่ใช้

ตารางที่ 8.1 แสดงรายละเอียดของฮาร์ดแวร์ที่ใช้

รายละเอียด	Computer 1	Computer 2	Computer 3	Computer 4
Operating Systems	Windows XP SP2	Windows XP SP2	Windows XP SP2	Windows Server 2003
Database Management System	Oracle10g SQL Server 2005	SQL Server2005	-	-
Web Application Server(Web Server)	Internet Information System 5.1 (IIS5.1)	Internet Information System 5.1 (IIS5.1)	Internet Information System 5.1 (IIS5.1)	Internet Information System 6.0 (IIS6.0)
Development Tools	Visual Studio 2005	Visual Studio 2005	Visual Studio 2005	-
.NET Framework	3.0	3.0	3.0	3.0

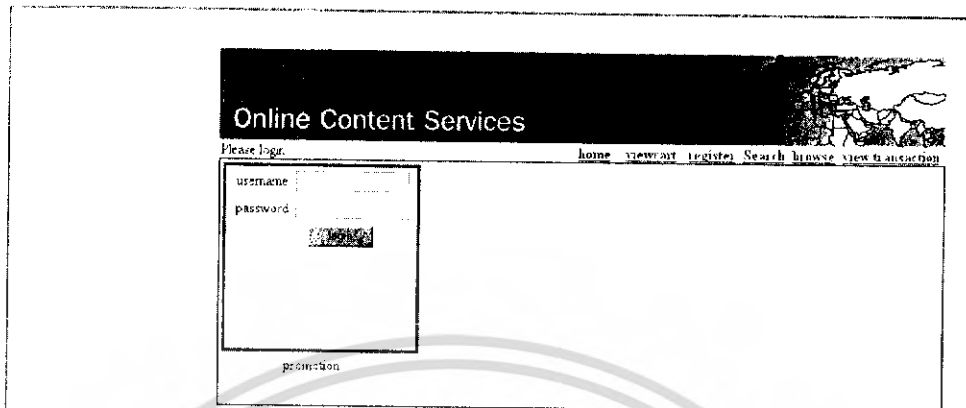
ตารางที่ 8.2 แสดงรายละเอียดของซอฟต์แวร์ที่ใช้

รายละเอียด	Computer 1	Computer 2	Computer 3	Computer 4
Processor	Pentium 1.7M	Pentium	Pentium	Pentium II 450 MHz
RAM	1.25GB	512MB	1.51GB	256 MB
HDD	60 GB	60GB	80GB	30GB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 8.2 ผลการทดลอง

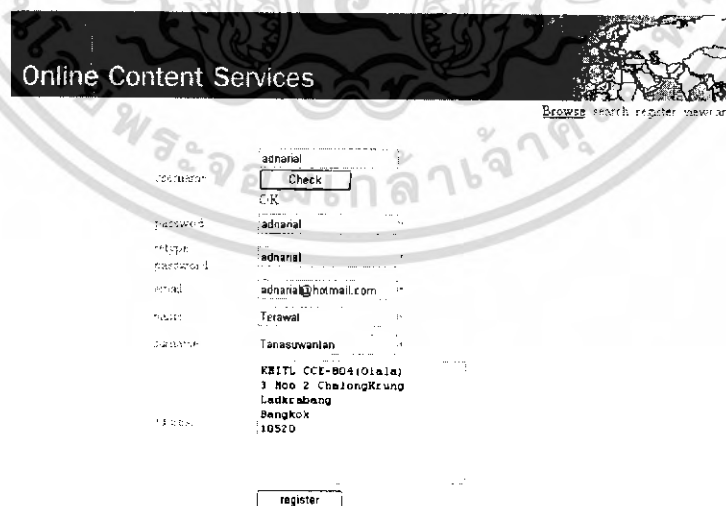
### 8.2.1 Register



รูปที่ 8.1 แสดงหน้าแรกของ Olala Content Service



รูปที่ 8.2 แสดงหน้าดั่งการ Login เมื่อทำการ login สำเร็จ

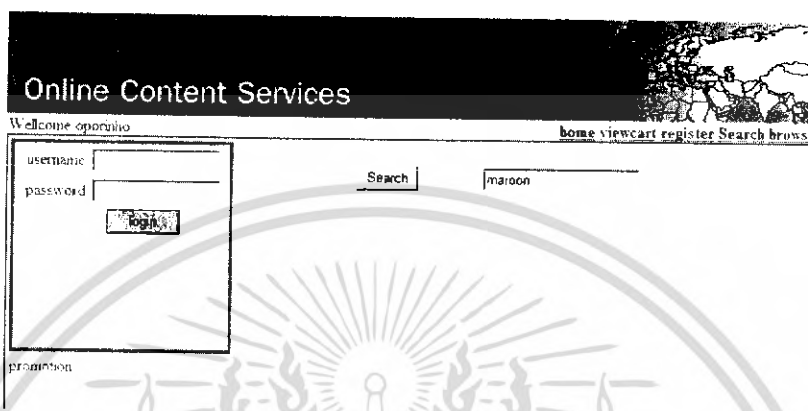


รูปที่ 8.3 แสดงหน้าดั่งการ Register เข้าใช้งาน

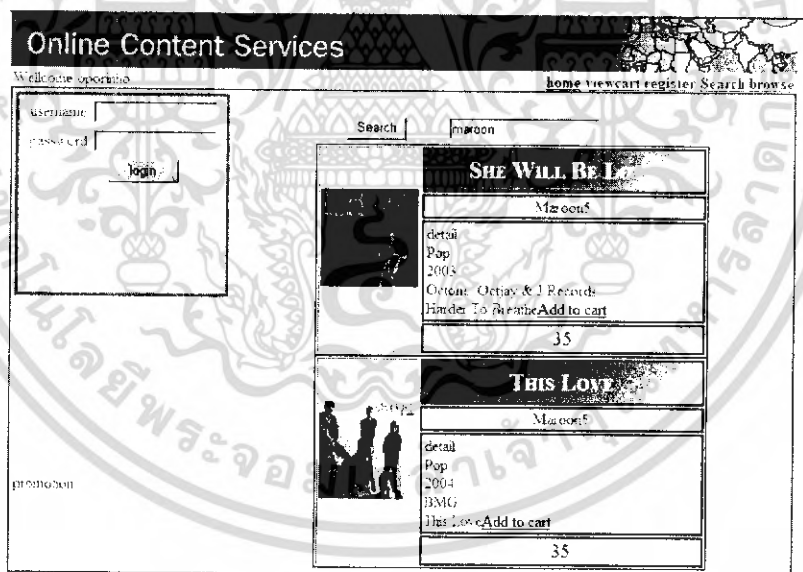
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 8.2.3 Search

User สามารถ Search ชื่อของ content ที่จะทำการ Download ได้



รูปที่ 8.4 แสดงหน้า Search

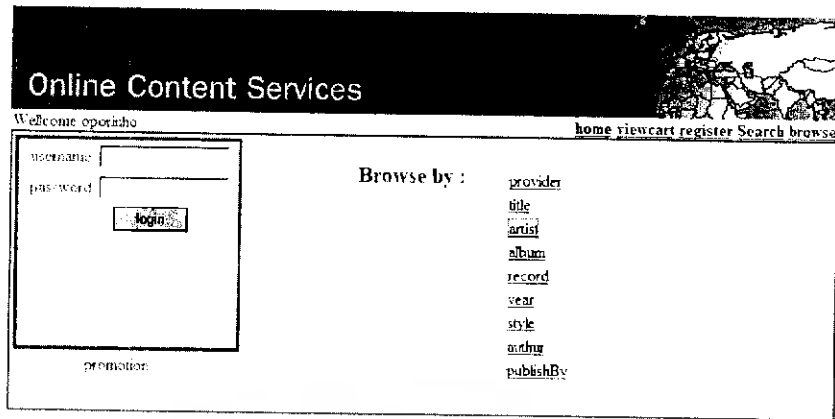


รูปที่ 8.5 แสดงหน้าแรกของ Olala Content Service

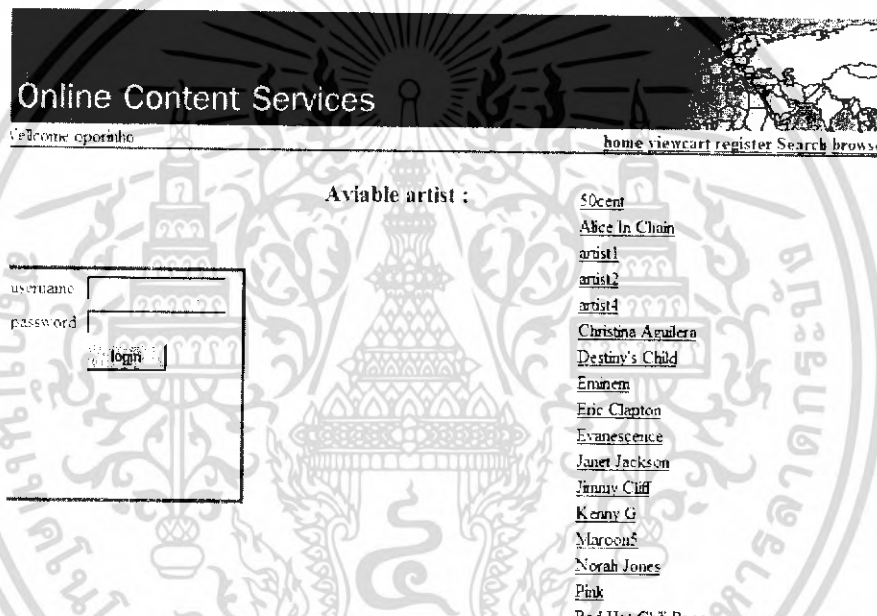
### 8.2.4 Browse

User สามารถเลือกชมสินค้าได้ตามประเภทของ content

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

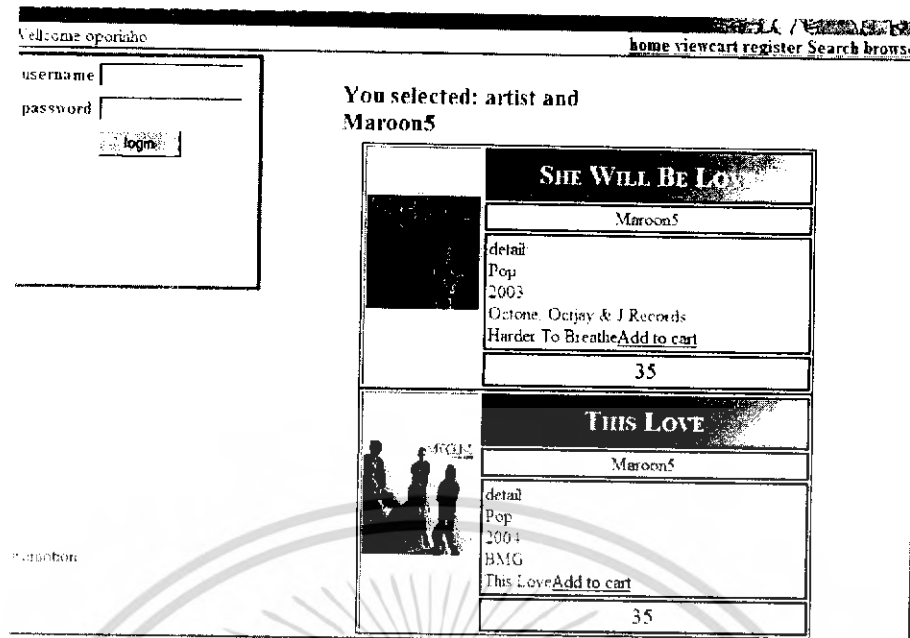


รูปที่ 8.6 แสดงหน้า Browse ของ Olala Content Service



รูปที่ 8.7 แสดงหน้า Aviable artist ของ Olala Content Service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.8 แสดงหน้าแรกของ Olala Content Service

#### 8.2.5 View Transaction

User สามารถ ดูรายการที่ซื้อไว้แล้วได้โดยเลือก view transaction ดังรูป

Online Content Services

home viewcart register Search browse view transaction

TransactionID	mdate	mtime	totalamount	paymentid
7	9/2/2550	18:04	500	asdf
8	9/2/2550	18:07	500	000
9	9/2/2550	21:32	1800	281
13	9/2/2550	22:44	1800	304
14	9/2/2550	22:45	1800	305
15	9/2/2550	22:47	0	306
16	9/2/2550	22:48	0	307
17	9/2/2550	22:49	1800	308
46	10/2/2550	10:7	0	361
47	10/2/2550	10:7	1800	362

1 2

รูปที่ 8.9 แสดงหน้าแรกของ Olala Content Service

#### 8.2.6 Buy Item

User สามารถเลือกซื้อได้โดยการ Search หรือ Browse แล้วทำการ Addcart ดังรูป

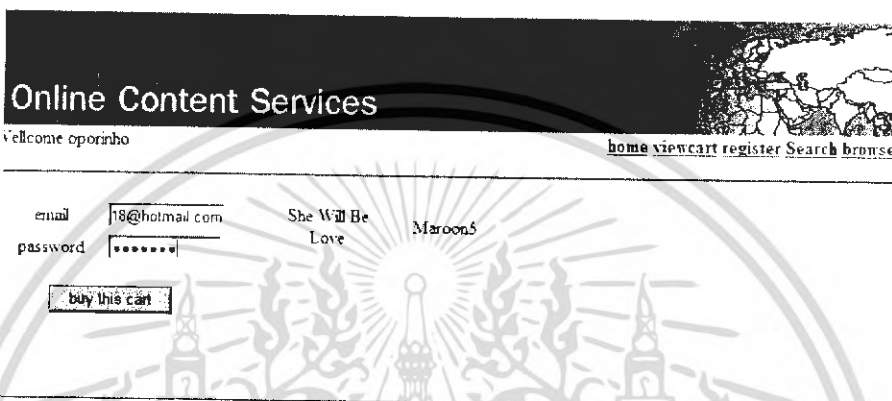
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Online Content Services

[home](#) [viewcart](#) [register](#) [Search](#) [browse](#) [view transaction](#)  
You have 1 items in cart

### รูปที่ 8.10 แสดงหน้าแสดงจำนวนของใน Cart ของ Olala Content Service

User สามารถดู Item ที่ทำการสั่งซื้อได้โดยทำการเลือกเมนู view cart



### รูปที่ 8.11 แสดงหน้าแสดงใน cart ของ Olala Content Service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 9

### บทวิจารณ์และสรุป

#### 9.1 บทวิจารณ์และสรุป

จากการที่ทางไมโครซอฟท์ได้ทำการเน้นถึงเทคโนโลยี SOA (Service oriented architecture) และการใช้งานเอกสาร XML ว่าเป็นแนวทาง และสิ่งจำเป็นสำหรับการพัฒนา Application ในอนาคตไม่ว่าจะเป็น web-based หรือ Application-based ซึ่งทางไมโครซอฟท์เองได้พยายามนำเสนอมาตรฐาน และเครื่องมือที่ใช้ในการพัฒนาออกมา ตั้งแต่ .NET Framework 2.0 มาจนถึง .NET Framework 3.0 ซึ่งมีการเพิ่มความสามารถในการใช้งานมาตรฐาน WS-\* ไม่ว่าจะเป็น ws-security, ws-coordination, ws-AtomicTransaction เป็นต้น ซึ่งทางผู้จัดทำก็ได้ทำการเลือกใช้ .NET Framework 3.0 อันประกอบไปด้วย WCF(Windows Communication Foundation), WWF(Windows Workflow Foundation) และ WPF (Windows Presentation Foundation)

สำหรับในโครงงานของทางผู้จัดทำได้ทำการศึกษา .NET Framework 3.0 ในส่วนของ WCF (Windows Co Communication Foundation) และ WWF (Windows Workflow Foundation) เพื่อนำมาใช้ในการพัฒนา Web Application ให้ได้ตรงตามเป้าหมาย คือ มีความปลอดภัย มีการทำงานแบบ all-or-nothing และมีการจัดการการใช้งานร่วมกันของเว็บเซอร์วิสให้มีประสิทธิภาพ

#### 9.2 ปัญหาอุปสรรค และแนวทางการแก้ไข

จากการศึกษา และทำการทดลองการพัฒนาเว็บเซอร์วิสตามมาตรฐาน WS-\* ต่างๆ โดยใช้ Microsoft Visual Studio 2005 เป็นเครื่องมือที่ใช้ในการทำการทดลองพัฒนาเว็บเซอร์วิสทางผู้จัดทำได้ประสบปัญหาต่างๆ และได้ทำการแบ่งออกเป็นหมวดหมู่ดังนี้

1. ปัญหาการทำงานร่วมกันระหว่างเว็บเซอร์วิสที่พัฒนาโดย .NET และ JAVA
2. ปัญหาในการสืบค้นข้อมูลที่จำเป็นในการใช้งาน
3. ปัญหาในการเลือกใช้เทคโนโลยี และมาตรฐานที่เหมาะสม
4. ปัญหาในการใช้งาน WS-\*
5. ปัญหาในการวิเคราะห์ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 9.2.1 ปัญหาการทำงานร่วมกันระหว่างเว็บเซอร์วิสที่พัฒนาโดย .NET และ JAVA

จากการศึกษา และทดลองพัฒนาระบบทำให้สามารถสรุปได้ว่าปัญหาที่ทำให้เว็บเซอร์วิสที่พัฒนาบน .NET และ JAVA ไม่สามารถทำงานร่วมกันได้เกิดมาจาก ปัญหาด้าน addressing และ การใช้งาน BPEL Engine

ตารางที่ 9.1 แสดงปัญหาทางด้าน addressing

รายการ	.NET	JAVA
การใช้งาน WSDL	สามารถเข้าใจ WSDL ได้แต่ไม่สามารถทำการใช้งานแมธอดได้	ไม่สามารถเข้าใจ WSDL ของทาง .NET ได้(เมื่อมีการใช้งาน WS-*)
การรองรับมาตรฐาน WS-Addressing	ใช้มาตรฐานที่แตกต่างเวอร์ชันกัน	ใช้มาตรฐานที่แตกต่างเวอร์ชันกัน

ตารางที่ 9.2 แสดงปัญหาทางด้าน BPEL Engine

รายการ	Active BPEL	BizTalk 2006	WWF (Windows Workflow Foundation)
การเรียกใช้ JAVA web service	ได้ แต่ไม่สามารถทำตาม WS-*	ไม่สามารถทำการเชื่อมต่อ service ได้	ได้ กับ BasicBinding
การเรียกใช้ .NET web service	ได้ แต่ไม่สามารถทำตาม WS-*	ได้ กับ BasicBinding	ได้ และสามารถทำได้ตาม WS-*

สำหรับปัญหาทางด้าน addressing ยังไม่สามารถทำการแก้ไขได้เนื่องจากต้องรอการเพิ่มเติมเครื่องมือที่ใช้ในการพัฒนาของทั้งสองค่าย

ปัญหาด้าน BPEL Engine สำหรับ Biztalk 2006 มีแนวทางการแก้ไขคือการใช้ WCF Adapter แต่เนื่องจาก .NET Framework 3.0 ยังเป็นเวอร์ชันทดลองอยู่ทำให้ Adapter ไม่สามารถทำการพัฒนาได้ทันที ซึ่งหากเปลี่ยนไปใช้ .NET Framework 3.0 ที่มีเวอร์ชันต่ำกว่าจะประสบกับข้อผิดพลาดของการพัฒนามากกว่าเวอร์ชันปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 9.2.2 ปัญหาในการสืบค้นข้อมูลที่จำเป็นในการใช้งาน

เนื่องจาก .NET Framework 3.0 ยังเป็นเวอร์ชันทดลอง ซึ่งมีการเปลี่ยนแปลงอยู่ตลอดเวลา ทำให้ไม่มีเอกสารที่ประกอบการใช้งานอย่างครบถ้วนสมบูรณ์ หรือมีเอกสารแต่เป็นของเวอร์ชันก่อนหน้าซึ่งมีการตั้งค่าที่แตกต่างกัน และไม่มีเอกสารอยู่ในระบบของ Microsoft ทำให้ต้องคอยตรวจสอบกับทางเว็บไซต์ของตัวผู้พัฒนา .NET Framework 3.0 อยู่ตลอดเวลา ซึ่งบางครั้งก็ไม่ค่อยมีการเปลี่ยนแปลงของข้อมูลมากนัก

### 9.2.3 ปัญหาในการเลือกใช้เทคโนโลยี และมาตรฐานที่เหมาะสม

ในการทดลองพัฒนาในทอมแรกทางผู้จัดทำได้ทำการศึกษาถึงมาตรฐานของ BPEL และเครื่องมือที่จะใช้ในการพัฒนาซึ่งก็คือ BizTalk Server 2006 แต่ภายหลังจากการที่ได้พัฒนาเว็บเซอร์วิสขึ้นมาบน .NET Framework 3.0 ทำให้พบว่า BizTalk Server 2006 ไม่สามารถทำงานร่วมกับ .NET Framework 3.0 ได้ซึ่งต้องอาศัยการใช้งาน Adapter เข้ามาช่วย แต่ตัว Adapter เองก็ยังไม่ได้รับการพัฒนาให้มีความเข้ากันได้กับ .NET Framework 3.0

สำหรับแนวทางในการแก้ไขทางผู้จัดทำได้เลือกใช้ WWF (Windows Workflow Foundation) ขึ้นมาใช้งานแทน BizTalk Server 2006 ซึ่งทำให้เสียเวลาในการศึกษาการใช้งาน BizTalk Server 2006 ไป

### 9.2.4 ปัญหาในการใช้งาน WS-\*

จากการศึกษาการใช้งาน WS-\* (WS-Security, WS-Coordination, WS-Atomic Transaction, WS-Business Activity, WS-Addressing, WS-policy) พบว่าการศึกษาในตัวมาตรฐานของ WS-\* นั้นมีแต่เอกสารเชิงเทคนิคที่อธิบายเพียงความต้องการต่างๆในการทำงาน แต่ไม่มีเอกสารรับรองว่าจะนำมาใช้งานในระบบได้อย่างไร ทำให้ทางผู้จัดทำต้องทำการสืบค้นจากเว็บไซต์และบล็อกของผู้พัฒนาเป็นหลัก โดยอาศัยการนำเสนอตัวอย่าง และทดลองตั้งค่าด้วยตัวเอง ทำให้ต้องใช้เวลาในการเรียนรู้ซึ่งเป็นผลให้การเลือกใช้เครื่องมือ และการตั้งค่าต่างๆของทาง .NET มีความล่าช้าเกิดขึ้น

นอกจากนั้นยังต้องทำการอัปเดต .NET Framework ให้เป็น .NET Framework 3.0 ซึ่งยังเป็นตัวทดสอบที่เผยแพร่ไว้ในกลุ่มของผู้พัฒนา ซึ่งยังไม่มีเอกสารรับรองอย่างเป็นทางการจาก Microsoft ทำให้การสืบค้นข้อมูลต่างๆเป็นไปด้วยความยากลำบาก เช่น เอกสารที่เกี่ยวข้องกับการใช้ WWF (Windows Workflow Foundation) ที่เพิ่งมีการตีพิมพ์ได้เพียง 2 สัปดาห์ก่อนการออกแบบระบบที่จะพัฒนาในโครงการเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 9.2.5 ปัญหาในการวิเคราะห์ระบบ

จากการที่ทางผู้จัดทำได้ทำการพัฒนาระบบการซื้อขายข้อมูลแบบดิจิทัลผ่านทางอินเทอร์เน็ต รวมไปถึงการพัฒนาระบบการชำระเงินผ่านอินเทอร์เน็ต ทำให้ต้องมีการศึกษาว่าในปัจจุบันมีการพัฒนาระบบดังกล่าวอย่างไร และต้องนำมาออกแบบ รวมถึงพัฒนา โดยอาศัยการทำงานของเว็บเซอวิส และเว็บแอปพลิเคชันร่วมกัน ซึ่งการทำงานบางกระบวนการยังไม่มีการใช้งานเว็บเซอวิสทำให้ทางผู้จัดทำต้องทำการออกแบบระบบขึ้นมา และประสบกับข้อผิดพลาดในระบบธุรกิจ ทำให้ต้องมีการออกแบบระบบซ้ำอีกหลายครั้ง เพื่อพัฒนาโมเดลของระบบที่ใกล้เคียงกับการทำงานจริงโดยมีพื้นฐานการใช้งานเว็บเซอวิสเป็นองค์ประกอบหลักของแอปพลิเคชัน

### 9.3 แนวทางการพัฒนาต่อ

ในโครงการนี้ได้เสนอเทคโนโลยีของเว็บเซอวิสที่อยู่ในสถาปัตยกรรมของเว็บเซอวิส ซึ่งถือว่าเป็นเรื่องใหม่ยังไม่เป็นที่นิยมในประเทศและเนื่องจากเอกสารทั้งหมดเป็นภาษาอังกฤษและเป็นเชิงเทคนิคซึ่งยากแก่การทำความเข้าใจสำหรับผู้เริ่มต้นศึกษา รวมไปถึงไม่มีลำดับขั้นตอนที่ชัดเจนในการศึกษา โครงการชุดนี้จึงเป็นทางเลือกที่จะช่วยให้ผู้สนใจได้เข้ามาศึกษาและนำการทดลองที่อยู่ในเอกสารชุดนี้ โดยที่ผู้สนใจสามารถนำไปพัฒนาต่อได้โดยเพิ่มขีดความสามารถให้กับเว็บเซอวิสมากขึ้นอาจจะทำได้โดยการนำเทคโนโลยีต่างๆมารวมกัน หรืออาจจะทำเทคโนโลยีใดเทคโนโลยีหนึ่งให้มีความซับซ้อนขึ้นมาโดยโครงการชุดนี้มีแนวทางการพัฒนาต่อดังนี้

1. การพัฒนาในส่วนติดต่อกับผู้ใช้ของเว็บแอปพลิเคชันให้มีความน่าใช้ยิ่งขึ้น เพื่อดึงดูดผู้ใช้ให้เข้ามาใช้งานในระบบ
2. พัฒนาการทำงานร่วมกันของ .NET และ JAVA ซึ่งอาจจะมองได้ว่าเป็นแนวทางหลักที่สำคัญอย่างยิ่งเพราะถ้าสามารถทำให้ .NET และ JAVA ทำงานร่วมกันได้จะสามารถขยายขอบเขตการใช้งานของเว็บเซอวิสเพิ่มขึ้นได้อีกมาก
3. พัฒนาระบบชำระเงิน จากการที่ในโครงการนี้ได้ทำการศึกษาระบบการทำธุรกรรมการเงินของ PayPal และได้ทำการพัฒนาขึ้นมาเป็นเว็บเซอวิสโดยอิงตามการทำธุรกรรมการเงินแบบชำระเงินจำนวนเงินสุทธิ ซึ่งควรมีการพัฒนาให้สามารถทำงานได้เหมือนกับ PayPal และมีการบริการเว็บเซอวิส ซึ่งสามารถขยายขีดความสามารถในการให้บริการการทำธุรกรรมการเงินขึ้นได้อีกมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

เอกสารอ้างอิงที่เป็น Web-site

[1] World Wide Web Consortium,

<http://www.w3c.org>

[2] UDDI.org

<http://uddi.org>

[3] CMSThailand Web Services,

<http://ws.cmsthailand.com>

[4] MSDN Home Page,

<http://msdn.microsoft.com>

[5] Authentication as the Foundation for eBusiness,

<http://www.securityfocus.com/printable/infocus/1513>

[6] Authentication Methods,

<http://www.datastreamconsulting.com/Authenticationmthds.htm>

[7] RSA SecurID Authentication,

[http://www.rsasecurity.com/products/secuid/whitepapers/BVBROI\\_WP\\_1201.pdf](http://www.rsasecurity.com/products/secuid/whitepapers/BVBROI_WP_1201.pdf)

[8] The Simplest Security: A Guide To Better Password Practices,

<http://www.securityfocus.com/printable/infocus/1537>

[9] Independent .NET Programmers Reference.

<http://www.dotnet247.com>

[10] .NET Framework, ASP.NET, C#, VB.NET, .NET Compact Framework, and SQL Server

<http://www.eggheadcafe.com>

[11] The Key Concepts of Biometric,

[http://hydria.u-strasbg.fr/~norman/BAS/key\\_concepts.htm#Biometric%20Authentication](http://hydria.u-strasbg.fr/~norman/BAS/key_concepts.htm#Biometric%20Authentication)

[12] Authentication Methods,

<http://www.datastreamconsulting.com/Authenticationmthds.htm>

[13] Authentication Reference Guide,

<http://www.securecomputing.com/index.cfm?sKey=665>

[14] Configure computers for user authentication,

<http://www.cert.org/security-improvement/practices/p069.html>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[15] Distributed Authentication,

<http://distauth.ucdavis.edu/overview>

[16] On the Front Lines of Network Security,

<http://www.unicenteradvisor.com/doc/05184>

[17] IMail - How to create a P12 Client SSL Certificate.htm

<http://support.ipswitch.com/kb/IM-20040331-DM03.htm>

[18] GITS - Government Certificate Service

<http://gits.neetec.or.th/services/govca/>

[19] Using .NET pfx

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncapi/html/pkcs12.asp>

เอกสารอ้างอิงที่เป็นปฏิญญานิพนธ์

[20] สมเกียรติ อภิสุทธิไมตรี “ระบบการซื้อขายวีดีโอบนเว็ลด์ไวด์เว็บ” ปฏิญญานิพนธ์

วิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ ปีการศึกษา 2542 คณะ

วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

[21] วิษณุ ขอบเขต “ลายมือชื่อดิจิตอล” ปฏิญญานิพนธ์วิศวกรรมศาสตรบัณฑิต ภาควิชา

วิศวกรรมคอมพิวเตอร์ ปีการศึกษา 2545 คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอม

เกล้าเจ้าคุณทหารลาดกระบัง

[22] วรเมธ วุฑฒะวนิช, วิชาติ อ่ำไพญญูญกุล “ระบบสารสนเทศผ่านเว็บเซอร์วิส” ปฏิญา

นิพนธ์วิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ ปีการศึกษา 2547 คณะ

วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารอ้างอิงที่เป็นหนังสือ

[23] ศุภชัย สมพานิช 2546 “คู่มือการเขียนโปรแกรม Visual C# .NET ฉบับโปรแกรมเมอร์”

สำนักพิมพ์อินโฟเพรส กรุงเทพฯ

[24] สุวัฒน์ สุธสมจินตน์ 2546 “คัมภีร์การใช้ Visual C# .NET ฉบับสมบูรณ์” สำนักพิมพ์ซีเอ็ด

ยูเคชั่น กรุงเทพฯ

[25] ยิ่งยศ สันติประเสริฐ 2545 “เรียนรู้ ASP.NET ด้วยภาษา C#” สำนักพิมพ์ซีเอ็ดยูเคชั่น

กรุงเทพฯ

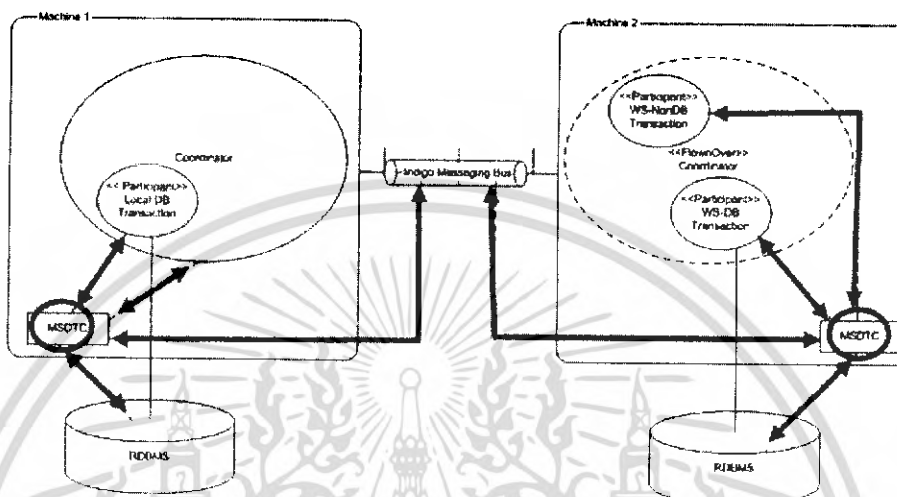
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

# การปรับแต่ง Windows XP SP2 ให้สามารถใช้งานทรานแซกชันแบบกระจาย ระหว่างเว็บเซอร์วิสได้

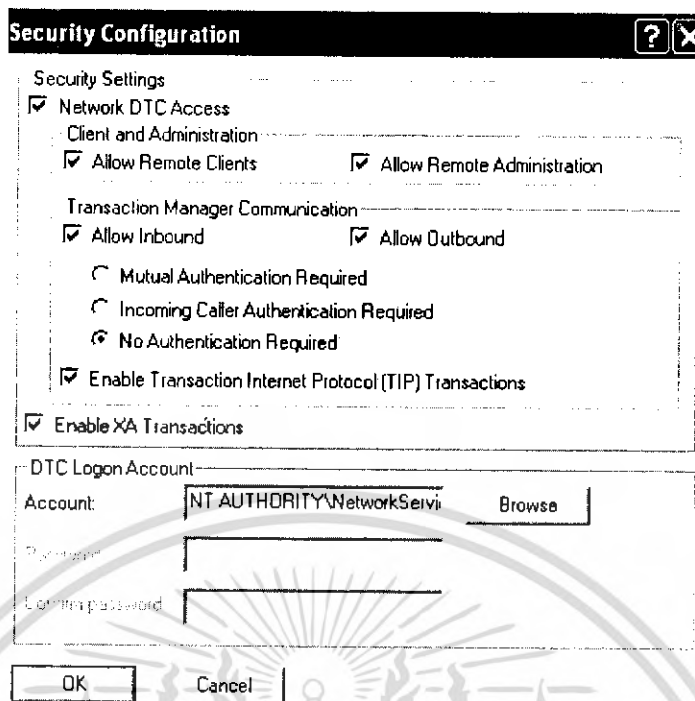


รูปที่ ก.1 แสดงความสัมพันธ์ของ MSDTC กับ Transaction

สำหรับการใช้งาน Transaction ที่ต้องทำงานระหว่าง Site นั้นเราต้องมีการตั้งค่าของ MSDTC (Microsoft Distributed Transaction Coordinator) ซึ่งก็คือ Coordinator ที่ทำหน้าที่รับผิดชอบ Transaction ระหว่าง Site ซึ่งทำได้โดยเปิดการทำงานของ MSDTC และทำการตั้งค่า Firewall ของ Windows ให้ MSDTC สามารถผ่านได้

การเปิดการทำงานของ MSDTC

1. คลิกที่ Start>Setting>Control Panel>Administrative Tools>Component Service
2. ทำการคลิกขวาที่ My Computer แล้วคลิกที่ Property
3. จากนั้นเลือกที่แท็บ MSDTC แล้วเลือก Security Configuration ใน Transaction Configuration
4. ถ้ามีการเชื่อมต่อระหว่าง Windows Server 2000 และ Windows XP ที่ Windows XP ต้องเลือก No Authentication Required (ภายใต้ Transaction Manager Communication)
5. ใน Security Setting ให้ทำการเลือกคังรูป



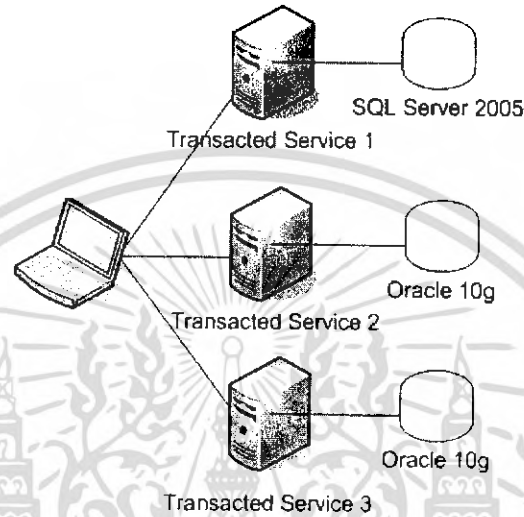
รูปที่ ก.2 แสดงการตั้งค่า Security Configuration ใน MSDTC

6. จากนั้นคลิกที่ ok เป็นอันเสร็จสิ้นการเปิดการทำงานของ MSDTC การตั้งค่าให้ MSDTC สามารถผ่าน Firewall ได้

1. คลิกที่ Start>run แล้วพิมพ์ว่า Wscui.cpl จากนั้นคลิกที่ ok
2. ใน Windows Security Center คลิกที่ Windows Firewall
3. คลิกที่ Add Program
4. จะปรากฏรายการของโปรแกรมให้คลิกที่ Browse แล้วไปยัง %WINDIR%\System32\Msdct.exe
5. คลิกที่ ok
6. คลิกที่ Add Port จากนั้นเลือก Open
7. ในกล่อง Name ให้พิมพ์ว่า Port 135 for MSDTC
8. ในกล่อง Port Number ให้พิมพ์ 135 และเลือก TCP
9. คลิกที่ ok เป็นอันเสร็จสิ้นการตั้งค่าให้ MSDTC สามารถผ่าน Firewall ได้

## ภาคผนวก ข

## การใช้งาน WS-Atomic Transaction บนฐานข้อมูล Oracle 10g และ SQL Server2005



รูปที่ ข.1 โมเดลที่ใช้ในการทดลองการทำงานของเว็บเซอร์วิสที่ใช้งานฐานข้อมูลคนละยี่ห้อ การตั้งค่าของเซอร์วิสต่างๆ และคอมโพเนนท์ที่ใช้ในการติดต่อกับฐานข้อมูล

ตารางที่ ข.1 แสดงการตั้งค่า และคอมโพเนนท์ที่ใช้ในเว็บเซอร์วิสแต่ละตัว

Configuration	Service 1	Service 2	Service 3
Binding Type	wsHttpBinding	wsHttpBinding	wsHttpBinding
Transaction Protocol	OleTx	OleTx	OleTx
DBMS	SQL Server 2005	Oracle 10g	Oracle 10g
Data Access Component	System.Data.SqlClient Version: 3.0.3600.0 Runtime: 2.0.50727 <b>Provider: Microsoft</b>	Oracle.DataAccess Version: 2.102.2.20 Runtime: 2.0.50727 <b>Provider: Oracle</b>	Oracle.DataAccess Version: 2.102.2.20 Runtime: 2.0.50727 <b>Provider: Oracle</b>
Transaction Handler	System.Transactions - TransactionScope	System.Transactions - TransactionScope	System.Transactions - TransactionScope

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในการทดลองจะใช้ทำการสร้างเว็บเซอร์วิสขึ้นมาทำการติดต่อกับฐานข้อมูลแต่ละตัว โดยให้ทำการอัปเดตค่าในฐานข้อมูลแต่ละตัว และใช้คำสั่งคิวรีที่ผิดพลาดเพื่อให้ทำการ โรลล์แบคฐานข้อมูลทั้งสามตัว

### การติดต่อกับฐานข้อมูล

สำหรับการติดต่อกับฐานข้อมูลจะมีด้วยกัน 3 วิธีหลักๆเรียงตามเก่าไปใหม่

1. การติดต่อด้วย ODBC
2. การติดต่อด้วย DSNLess
3. การติดต่อด้วย OLEDB

ซึ่งการติดต่อกับทั้ง 3 วิธีนั้นต้องติดต่อกับคอมโพเนนต์ที่โดยมีดังต่อไปนี้ (บน WCF)

ตารางที่ ข.2 แสดงรายละเอียดของคอมโพเนนต์แต่ละตัว

#	Provider	Component	Version	Runtime	OLEDB	ODBC
#1	Microsoft	System.Data.SqlClient	3.0.3600.0	v2.0.50727	Yes	Yes
#2	Microsoft	System.Data.OracleClient	2.0.0.0	v2.0.50727	Yes	Yes
#3	Oracle	Oracle.DataAccess	2.102.2.20	v2.0.50727	Yes	No
#4	Oracle	Oracle.DataAccess	10.2.0.100	v1.0.3705	n/a	n/a

จากตารางที่ ข.1 การใช้งาน WS-AtomicTransaction นั้น จะใช้งานผ่าน System.Transactions ซึ่งสร้างขึ้นมาจาก System.Service.Enterprise และใช้งานได้บนรันไทม์ v2.0.xxxxx ขึ้นไปเท่านั้นทำให้ คอมโพเนนต์ #4 ไม่สามารถใช้ในการทดลองได้

ถึงแม้ว่าจะเปลี่ยนไปใช้การติดต่อกับฐานข้อมูลแบบ ODBC แต่สิ่งที่มีผลในการทำงานตาม WS-AtomicTransaction คือ คอมโพเนนต์ ที่มีใน .NET ทำให้การเปลี่ยนไปใช้ ODBC Connection ให้ผลลัพธ์ไม่ต่างกัน (ในกรณีคอมโพเนนต์ #1 และ #2)

เพราะฉะนั้น สำหรับหากต้องการที่จะใช้งาน WS-Atomic Transaction กับเว็บเซอร์วิสที่ติดต่อกับฐานข้อมูล Oracle 10g ควรจะต้องอิงตาม คอมโพเนนต์ ที่สามารถทำงานร่วมกับ System.Transactions ซึ่งมีการติดต่อกับ DTC (Distributed Transaction Coordinator) ได้ซึ่งทำให้คอมโพเนนต์ที่เลือกขึ้นมาใช้คือคอมโพเนนต์ที่ 1 สำหรับ SQL Server 2005 ส่วน Oracle 10g ใช้คอมโพเนนต์ที่ 2 และ 3

ตารางที่ ข.3 แสดงรายละเอียดของคอมโพเนนต์แต่ละตัว

Test Case	Service 1	Service 2	Service 3	Tx State	Rollback
(valid,valid,valid)	Commit	Commit	Commit	Committed	No

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางธุรกิจ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(valid,valid,invalid)	Commit	Commit	Aborted	Aborted	Yes
(valid,invalid,valid)	Commit	Aborted	-	Aborted	Yes
(invalid,valid,valid)	Aborted	-	-	Aborted	Yes



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

### การศึกษาธุรกรรมบน PayPal

จากการที่ในโปรเจกมีเว็บเซอร์วิสที่เกี่ยวข้องกับการทำธุรกรรมบนอินเทอร์เน็ตทำให้ทางคณะผู้จัดทำได้ทำการศึกษาการทำธุรกรรมบน PayPal ซึ่งจากการค้นคว้าพบว่าเป็นช่องทางที่ได้รับความนิยมสูง ในการทำธุรกรรมบนอินเทอร์เน็ตทั้งยังเป็นช่องทางที่ผู้ค้าขายบนเว็บไซต์ Ebay ใช้กันมากที่สุด

PayPal เป็นธุรกิจที่ให้ผู้ใช้งานสามารถทำธุรกรรมบนอินเทอร์เน็ตได้ โดยทำหน้าที่ดูแลกระบวนการการทำธุรกรรมระหว่างผู้ขายออนไลน์ เช่น Ebay รวมไปถึงองค์กรหรือบริษัทที่ต้องการเพิ่มทางเลือกในการชำระเงินให้กับลูกค้า ซึ่ง PayPal จะหากำไรจากค่าธรรมเนียมที่ใช้ในการทำธุรกรรมจากผู้ขาย ซึ่งผู้ซื้อไม่จำเป็นต้องเสียค่าธรรมเนียมใดๆ ทำให้ PayPal ได้รับความนิยมในกลุ่มของผู้ซื้ออย่างมาก และจำนวนผู้ใช้บริการขยายตัวอย่างต่อเนื่อง ในการที่จะเข้าไปถือครองบัญชีของ PayPal ได้นั้น มีเงื่อนไขอยู่ว่า ผู้ใช้ต้องอายุไม่ต่ำกว่า 18 ปี และต้องมีบัตรเครดิตเดบิต หรือบัญชีธนาคาร และอีเมลล์

สำหรับทาง PayPal ได้เสนอลักษณะแนวคิดในการทำธุรกรรมผ่านทางอินเทอร์เน็ตให้กับทางผู้ขาย โดยแบ่งออกได้เป็น 3 ลักษณะด้วยกัน ดังนี้

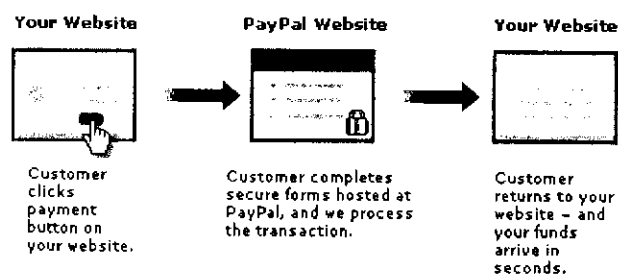
1. การทำธุรกรรมผ่านทางเว็บไซต์
  - 1.1. เพื่อให้เว็บไซต์ของผู้ขายสามารถรับชำระด้วยบัตรเครดิตได้
  - 1.2. เรียกใช้งานได้ผ่านทางแท็กแบบ HTML
2. การทำธุรกรรมผ่านทางอีเมลล์
3. เพิ่มทางเลือกในการชำระเงินให้กับลูกค้า

ซึ่งลักษณะของการทำธุรกรรมที่เลือกขึ้นมาใช้คือการทำธุรกรรมตามลักษณะที่ 1 โดยทาง PayPal จะทำการแบ่งออกเป็น 5 ประเภทด้วยกัน

1. ผู้ขายมีระบบการขายสินค้าอยู่แล้ว และใช้ PayPal ในการชำระมูลค่ารวมของทั้งรายการ
2. ผู้ขายไม่มีระบบการขาย และต้องการให้ PayPal ทำหน้าที่แสดงรายการย่อยของรายการ
3. การขายสินค้าทีละรายการ
4. การชำระค่าสมาชิกรายเดือน
5. รับการบริจาค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Customers shop on your website and pay on PayPal.



### รูปที่ ค.1 รูปแสดงหน้าต่างของ PayPal ที่ผู้ซื้อต้องทำการยืนยันตัวก่อนชำระเงิน

สำหรับขั้นตอนในการทำธุรกรรมโดยผ่านทาง PayPal จะมีขั้นตอนดังนี้

1. ผู้ใช้ทำการคลิกที่ปุ่ม "ชำระเงิน"
2. ผู้ใช้ทำการยืนยันตนบนเว็บไซต์ของ PayPal
3. ทำการยืนยันการชำระเงิน
4. กลับสู่หน้าเว็บไซต์ของผู้ขาย

### PayPal เว็บเซอร์วิส

นอกจากนี้ทาง PayPal ยังมีเว็บเซอร์วิสให้บริการกับผู้ขายที่ต้องการพัฒนาระบบให้ใช้งาน

กับทาง PayPal ได้อย่างมีประสิทธิภาพ โดยผู้ซื้อไม่จำเป็นต้องเข้าไปที่หน้ายืนยันตนของ PayPal ก่อนแต่สามารถทำการชำระเงินโดยผ่านทางเว็บไซต์ของผู้ขายได้ทันทีซึ่งผู้ซื้อสามารถทำการเลือกได้ว่า จะชำระเงินผ่านทางบัตรเครดิต หรือผ่านทางบัญชีของ PayPal

ในการชำระผ่านทางบัตรเครดิตผู้ซื้อจะต้องทำการกรอกข้อมูลซึ่งเสมือนการสมัครเข้าใช้บริการกับทาง PayPal โดยประกอบไปด้วย

1. ข้อมูลของผู้ซื้อ (ชื่อ, นามสกุล, ที่อยู่, อีเมลล์ และรหัสผ่านส่วนตัว)
2. ข้อมูลทางการเงิน (รหัสบัตรเครดิต, วันหมดอายุ, รหัสยืนยันบัตร และประเภทของบัตร)

ส่วนการชำระผ่านทางบัญชีของ PayPal ผู้ซื้อเพียงกรอกอีเมลล์ที่ใช้ในการสมัคร PayPal และรหัสส่วนตัว โดยระบบจะตัดเงินจากบัญชีของ PayPal ก่อน แล้วจึงทำการตัดเงินจากบัตรเครดิตต่อไป

### การรักษาความปลอดภัย และยืนยันตนกับทางเว็บเซอร์วิสของ PayPal

ทาง PayPal ได้กำหนดว่าในการที่จะทำการเรียกใช้งานว่าต้องเป็นไปตามการรักษาความปลอดภัยของข้อมูล 4 ข้อ เพื่อให้แน่ใจว่าผู้ที่เข้ามาใช้งานเว็บเซอร์วิสคือผู้ที่มีสิทธิ์ใช้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์โดย PayPal และอาจมีการเปลี่ยนแปลงโดยไม่ต้องแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ชื่อผู้ใช้ และรหัสผ่าน ซึ่งจะได้มาหลังจากที่ทำการสมัครเสร็จสิ้น
2. กลไกการพิสูจน์ตัวตน ซึ่งเป็นอันใดอันหนึ่งดังต่อไปนี้
  - a. Client side request signing ผ่านทาง PayPal API Certificate
  - b. Request Authentication ผ่านทาง API Signature (รวมอยู่ใน request)
3. กลไกการพิสูจน์ตัวตนจากองค์กรภายนอก (อาจจะมี หรือ ไม่มีก็ได้)
4. การส่งข้อมูลแบบ เอสเอสแอล (SSL)

```

<SOAP-ENV:Header>
  <RequesterCredentials xmlns="urn:ebay:api:PayPalAPI"
    xsi:type="ebl:CustomSecurityHeaderType">
    <Credentials xmlns="urn:ebay:api:eBLBaseComponents"
      xsi:type="ebl:UserIdPasswordType">
      <Username>api_username</Username>
      <Password>api_password</Password>
      <Signature>api_signature</Signature>
      <Subject>authorizing_account_emailaddress</Subject>
    </Credentials>
  </RequesterCredentials>
</SOAP-ENV:Header>

```

รูปที่ ค.2 รูปแสดงตัวอย่างของเมสเซจที่ใช้ในการพิสูจน์ตัวตนกับทาง PayPal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

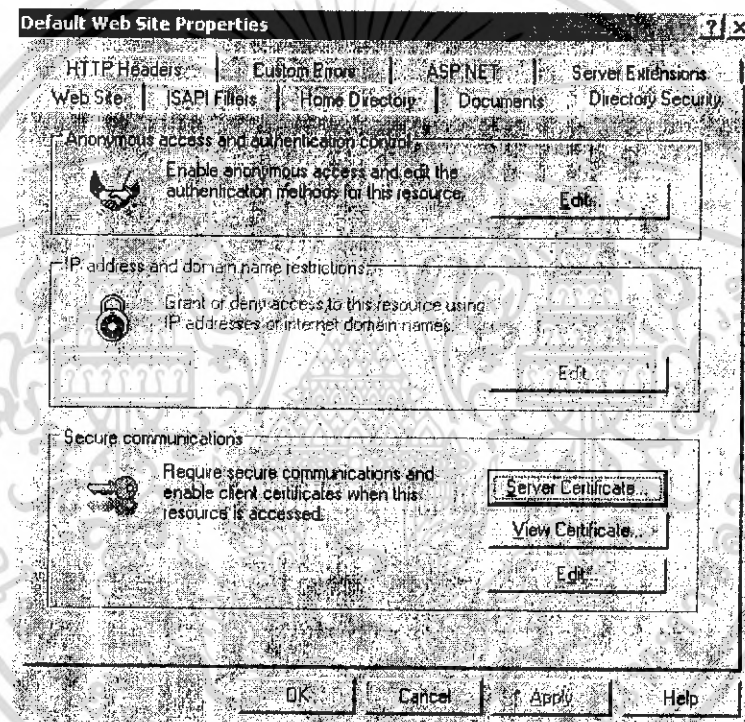
## ภาคผนวก ง

### Security configuration

#### การร้องขอ Certificate

เนื่องจากโครงการนี้ได้มีการจัดตั้ง Certificate authority ขึ้นเอง โดยใช้ component ของ Windows server 2003 การร้องขอ Certificate กระทำได้ดังนี้

1. ที่เครื่องที่ต้องการขอ certificate เลือก Administrative tool>IIS>Default website>Properties>Directory Security>Server Certificate



รูปที่ ง.1 แสดงหน้าต่าง Default Web Site Properties

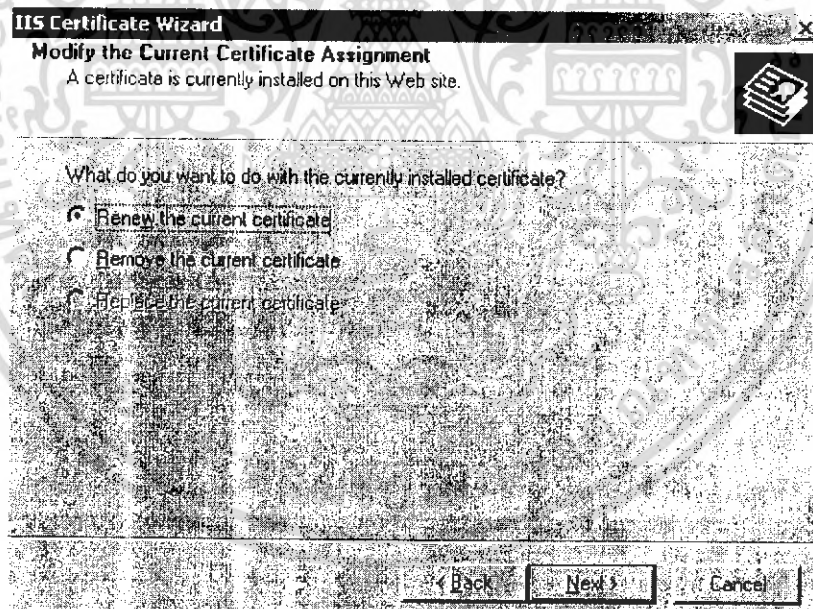
2. จากนั้นแสดง Wizard ขึ้นมาเลือก Next

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ง.2 แสดงหน้าต่าง Web Server Certificate Wizard

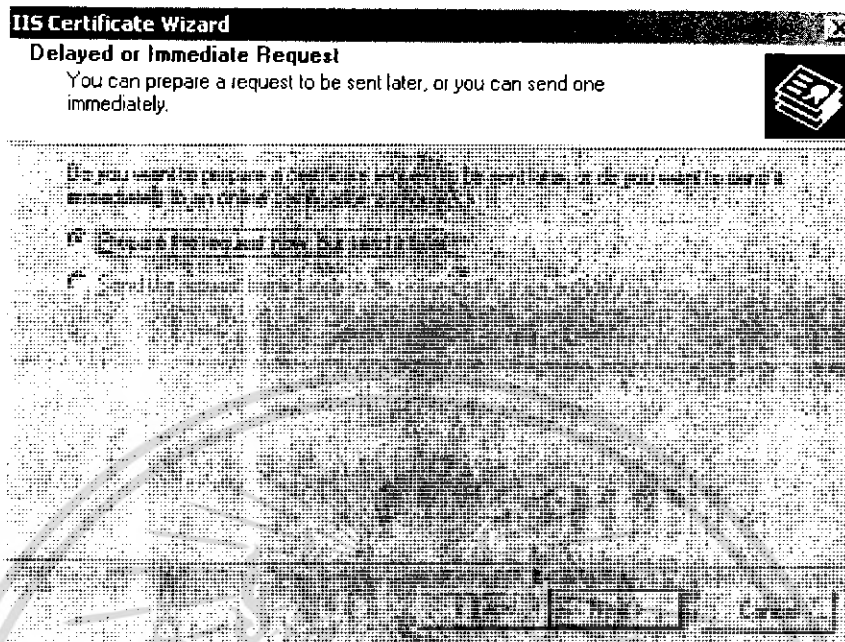
3. เลือก Renew the current certificate แล้วเลือก Next



รูปที่ ง.3 แสดงหน้าต่าง IIS Certificate Wizard

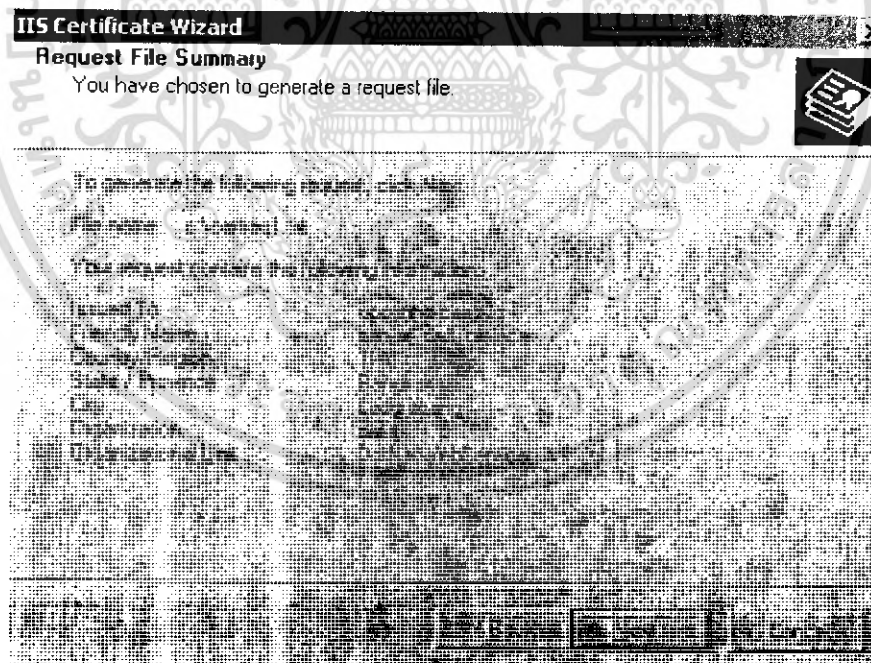
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4. เลือก Prepare the request now แล้วเลือก next



รูปที่ 3.4 แสดงหน้าต่าง IIS Certificate Wizard

#### 5. เลือก Next



รูปที่ 3.5 แสดงหน้าต่าง IIS Certificate Wizard

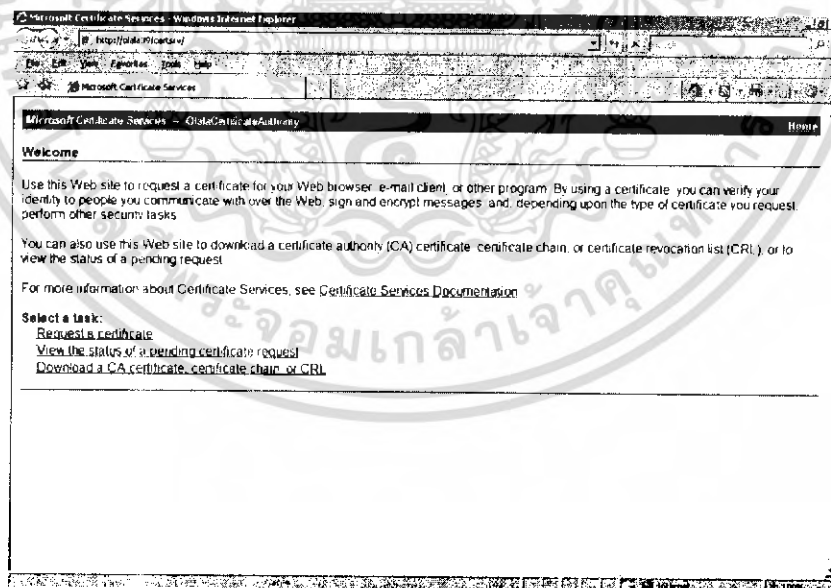
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6 Windows จะสร้าง Certificate request ซึ่งมีข้อมูลดังรูปหลังจากนั้นเมื่อขึ้น finish หมายถึงขั้นตอนนี้เสร็จสิ้นแล้ว



รูปที่ 3.6 แสดงหน้าต่าง IIS Certificate Wizard

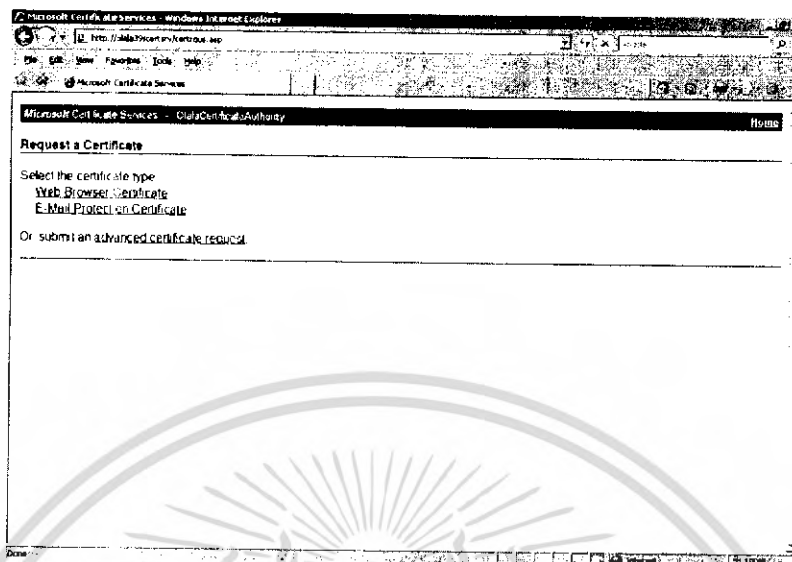
7 เปิดหน้าต่าง Web Browser แล้วเปิด URL ของ CA Service (ตัวอย่าง <http://olala39/certsrv>) แล้วเลือก Request a certificate



รูปที่ 3.7 แสดงหน้าต่าง CA Service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

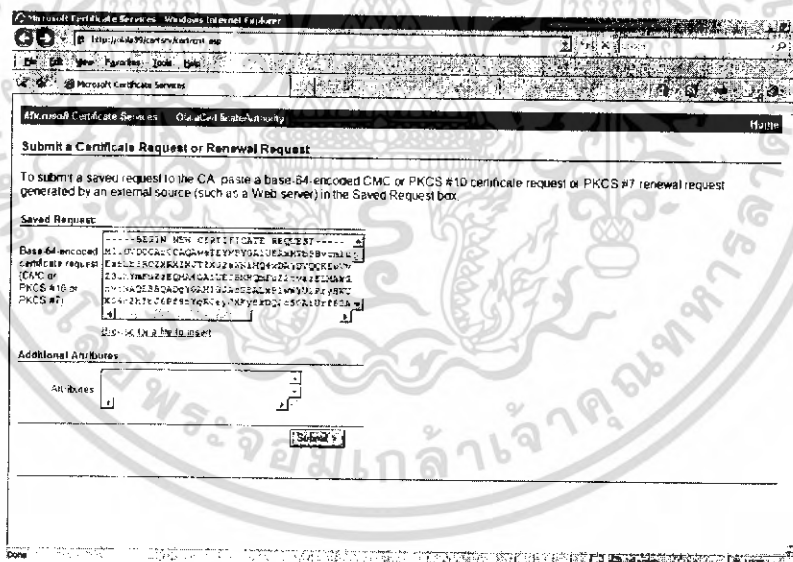
8 เลือก Advanced certificate request.



รูปที่ ง.8 แสดงหน้าต่าง CA Service

9 เปิด text file ที่ได้จากขั้นตอนที่ 1 - 6 แล้วนำข้อความที่ได้มาใส่ใน Text box ดังรูปคลิกที่

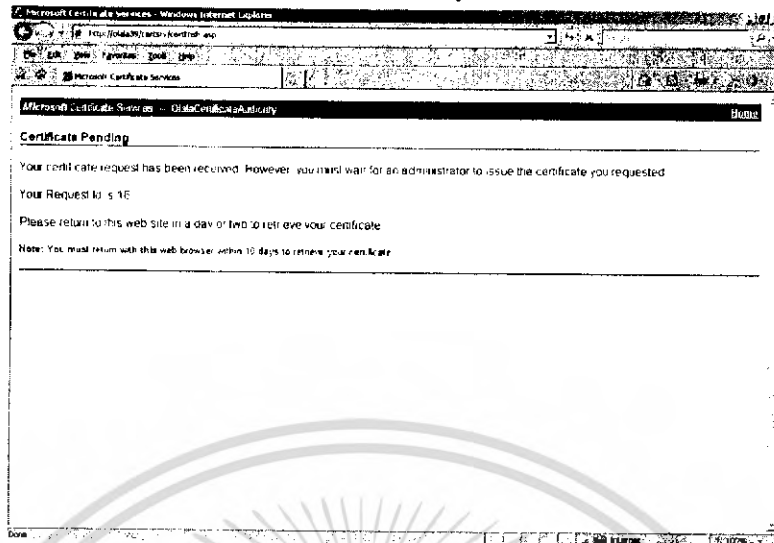
Submit



รูปที่ ง.9 แสดงหน้าต่าง CA Service

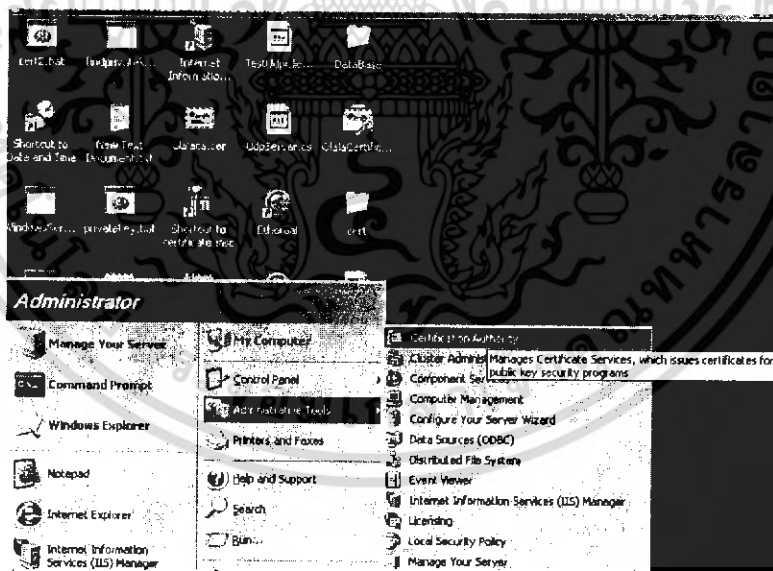
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 10 หลังจากคลิก Submit แล้วจะแสดงหน้าในรูปที่ ง.10



รูปที่ ง.10 แสดงหน้าต่าง CA Service

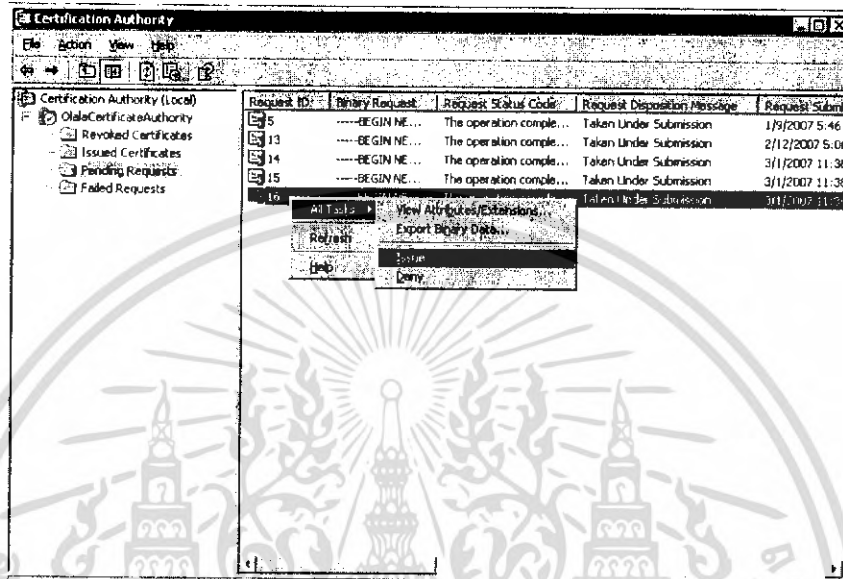
11 ขั้นตอนต่อไปเป็นการ Issue certificate ซึ่งผู้ที่กระทำได้ต้องเป็นผู้มีสิทธิ์ในการ Issue ซึ่งสามารถกำหนดได้ในขั้นตอนการติดตั้ง การเปิด Certificate Authority สามารถเปิดได้โดยเข้าไปที่ Start > Administrative tool > Certificate Authority



รูปที่ ง.11 แสดงวิธีเลือก Certificate Authority

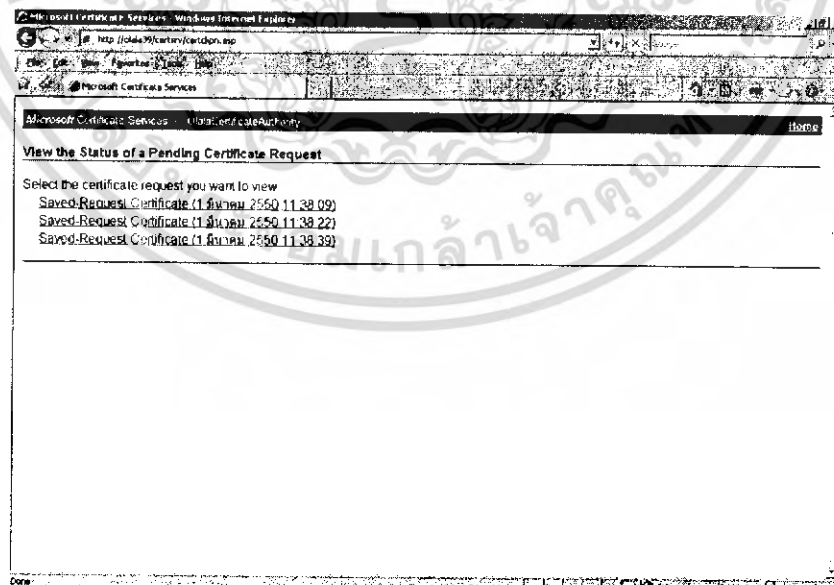
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12 เมื่อเข้ามาที่ Certificate authority แล้วให้ Pending Requests ซึ่งจะมารายการของ Certificate request ที่รอการ Issue จาก CA อยู่ให้คลิกขวาที่ Request ที่เราจะ Issue แล้วเลือก All Tasks > Issue



รูปที่ ง.12 แสดงการ Issue certificate

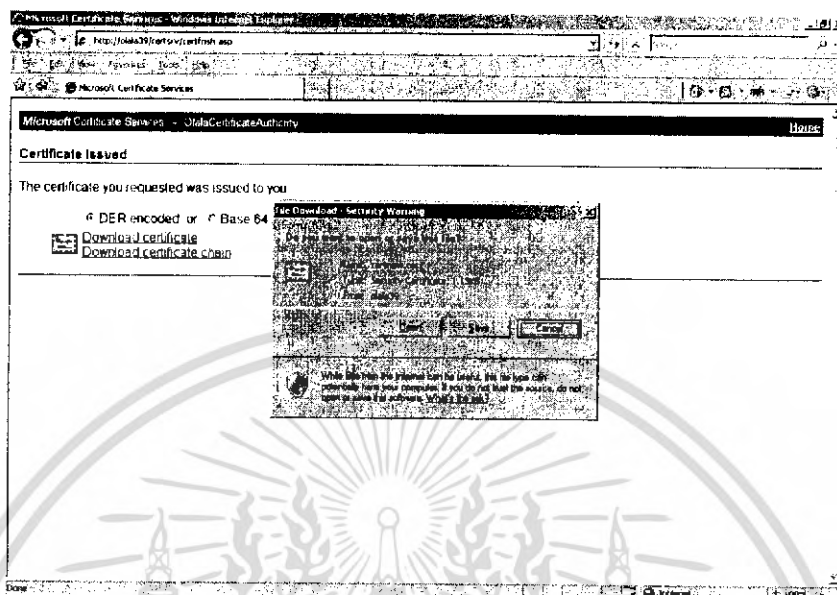
13 เปิด Web browser แล้วเข้าที่ CA Service หลังจากนั้นเลือก View pending certificate แล้วเลือก Certificate ที่เรา Request ไว้ในขั้นตอนที่ 10



รูปที่ ง.13 แสดงหน้าต่าง CA Service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14 เลือก Download certificate แล้วจะขึ้น Dialog ให้ Save (หรืออาจเป็นโปรแกรมช่วยดาวน์โหลดอื่นๆ)



รูปที่ 14 แสดงหน้าต่าง CA Service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก จ

### .NET Framework

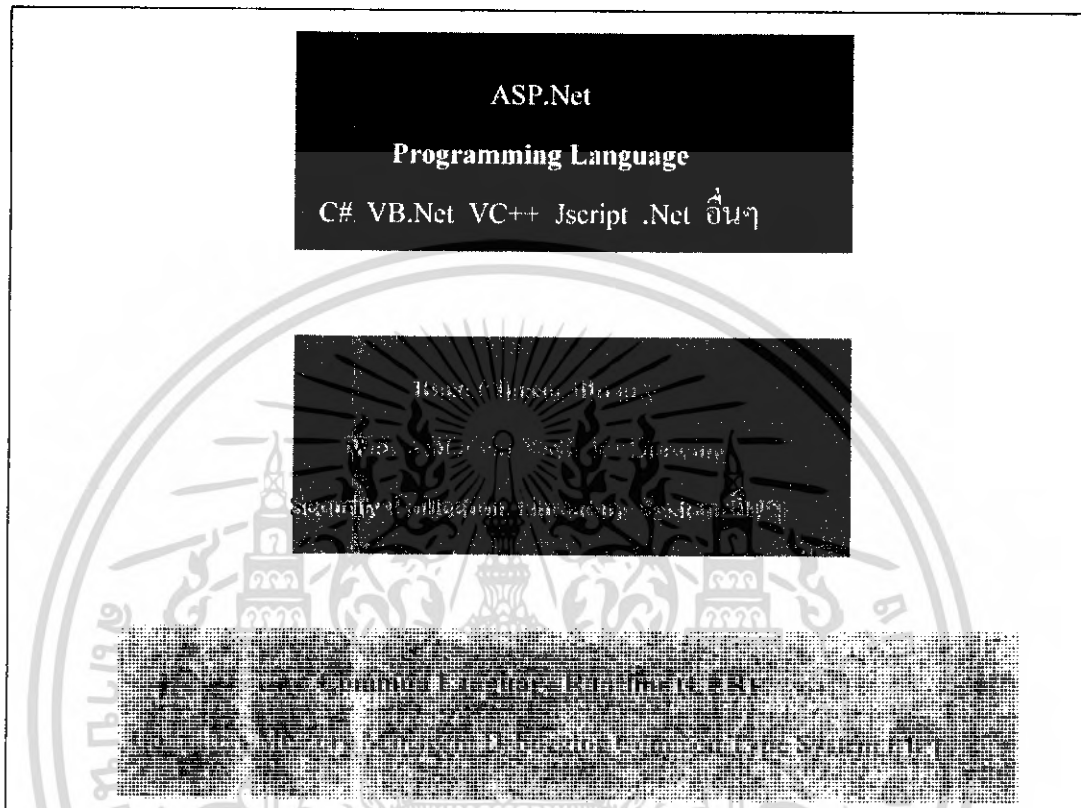
.Net Framework คือ โครงสร้างการพัฒนาโปรแกรมคอมพิวเตอร์ ที่ถูกออกแบบมาเพื่อให้อำนวยความสะดวกในการพัฒนาโปรแกรมสมัยใหม่ ที่ใช้งานในระบบเครือข่าย (Internet, Intranet, Mobile Devices, ฯลฯ) Bill Gates และ Steve Ballmer ได้บรรยายสรุปวิสัยทัศน์ ที่เกี่ยวกับ .Net เอาไว้ 3 ข้อหลัก ๆ ได้แก่

1. การพัฒนาโปรแกรมในรูปแบบของ Web Service จะเป็นหัวใจหลักในการขับเคลื่อนโปรแกรมต่าง ๆ ที่ใช้งานบน Internet. Web Service จะช่วยให้การติดต่อสื่อสารระหว่าง application บน Internet นั้นง่ายขึ้น และเป็นระบบมากยิ่งขึ้น
2. Web Service ขึ้นพื้นฐานเช่น การตรวจสอบ user ที่ log in เข้าสู่ระบบ จะถูกพัฒนาให้เป็นมาตรฐาน และสามารถนำไปใช้ได้ทั่วไปบน Internet
3. PC (desktop, notebook) และ Mobile Device ที่ต่อเชื่อมกับ Internet ได้ เช่น PDA และ โทรศัพท์มือถือ จะมีบทบาท และประโยชน์มากขึ้นไปอีก เมื่อสามารถติดต่อใช้งาน โปรแกรมต่าง ๆ บน Internet ได้

Microsoft ได้พัฒนารูปแบบการพัฒนาระบบคอมพิวเตอร์ขึ้นมา เรียกว่า .Net Framework (อันที่จริงแล้ว Microsoft ไม่ได้เป็นผู้คิดค้นเรื่องพวกนี้ขึ้นมาแต่เพียงผู้เดียว สิ่งต่าง ๆ เหล่านี้มีผู้คิดค้นพัฒนาจากหลายบริษัท หลายหน่วยงาน ยกตัวอย่างเช่น Sun Microsystems, IBM, ฯลฯ หากแต่ว่า Microsoft นำแนวคิดเหล่านั้นมาออกแบบให้อยู่ในรูปแบบที่สามารถต่อเชื่อมกันได้ ง่ายขึ้น เป็นระบบมากขึ้น) เพราะฉะนั้น ถ้าจะให้นิยามคำว่า .Net คงไม่สามารถชี้เฉพาะได้ว่า .Net คืออะไร เพราะจริงๆ แล้ว .Net ประกอบไปด้วยส่วนประกอบต่าง ๆ หลายส่วนด้วยกัน ส่วนประกอบเหล่านี้ถูกออกแบบมาเพื่อให้ทำงานได้เข้ากันได้ดียิ่งขึ้น

Net Framework นั้นประกอบไปด้วยอะไรบ้าง

ส่วนประกอบหลัก ๆ ของ .Net Framework แบ่งเป็นชั้น ๆ ได้ดัง diagram ต่อไปนี้



รูปที่ จ.1 แสดง .Net framework diagram

1. **Common Language Runtime (CLR)** เป็นส่วนพื้นฐานที่ติดต่อกับระบบปฏิบัติการ Windows ทำหน้าที่เป็น run-time environment ให้กับโปรแกรมที่เขียนขึ้นสำหรับใช้บน .Net CLR มีส่วนของ compiler ทั้งที่เป็นแบบปกติ (compile ก่อนที่จะนำโปรแกรมไปใช้) และแบบ Just-In-Time (compile เมื่อจะใช้โปรแกรมนั้น ๆ) มีส่วนของ Memory Management ที่เอาไว้สำหรับจัดสรรหน่วยความจำของเครื่องให้กับ โปรแกรม รวมไปถึงการทำ Garbage Collection (การเรียกคืนหน่วยความจำที่ไม่ได้ใช้อีกต่อไป) ส่วนของ Common Type Systems (CTS) ทำให้ภาษาต่าง ๆ ที่เขียนขึ้นบน .Net สามารถทำงานร่วมกันได้ เพราะขนาด และรูปแบบของข้อมูลที่เก็บไว้เป็นรูปแบบเดียวกัน

2. **Base Classes** เป็น class library พื้นฐาน ที่โปรแกรมต่าง ๆ ไม่ว่าจะเขียนด้วยภาษาใดบน .Net ก็สามารถใช้ร่วมกันได้ เช่น การติดต่อบริบบฐานข้อมูล (ADO.Net), การติดต่อกับ file system ของ server (IO), ฯลฯ

เอกสารนี้มีลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3. Programming Languages** เป็นเซตของ ภาษาคอมพิวเตอร์ ที่ถูกออกแบบมาเพื่อการเขียน โปรแกรมบน .Net Framework ไมโครซอฟท์นั้นเน้นไปที่ 3-4 ภาษาหลัก ๆ ได้แก่ VB.Net ซึ่งเป็น ตัวที่พัฒนาต่อมาจาก VB, C# ซึ่งเป็นภาษาใหม่ที่มี syntax ใกล้เคียงกับ Java และ C++, Visual C++, และ JScript.Net ส่วนภาษาอื่น ๆ นั้น มีบริษัท หรือหน่วยงานอื่น ๆ เป็นผู้พัฒนาขึ้น ซึ่งคาดว่าจะมี เป็นสิบ ๆ ภาษา สำหรับ .Net Framework นั้นไม่ว่าจะเขียน โปรแกรมด้วยภาษาใดก็ตาม Compiler ใน CLR ก็จะ compile โปรแกรมนั้นให้อยู่ในรูปของ Intermediate Language (IL) ซึ่งจะถูกนำไป แปลเป็นภาษาเครื่อง (Native Code) อีกทีเมื่อตอนที่นำไปใช้

**4. ASP.Net** เป็นภาษา script ที่พัฒนาต่อมาจาก ASP ตัวเก่า เพื่อให้โปรแกรมเมอร์สามารถพัฒนา web application ให้ใช้ .Net ได้สะดวกขึ้น ASP.Net นี้ถึงแม้จะอ้างอิงมาจาก ASP ตัวเก่า แต่ก็มี syntax หลายส่วนที่เปลี่ยนแปลง

จะเห็นได้ว่า .Net Framework กับ Sun's J2EE นั้นมีความคล้ายคลึงกันมาก สรุปได้ ดังนี้ CLR = JVM (Java Virtual Machine), IL = Java Bytecode, .Net base classes = Java Class Library, .Net Programming Languages = Java Language, ASP.Net = Java Server Page (JSP). ข้อแตกต่างในด้าน Architecture หลัก ๆ ก็คือ IL ของ .Net นั้นต้อง run บน Windows เท่านั้น แต่ Java Bytecode นั้นสามารถrun บน OS ไหนก็ได้ที่มี JVM ส่วนในด้าน Performance นั้น ไมโครซอฟท์ ได้ทำการเปรียบเทียบโดยพัฒนาโปรแกรม Pet Shop ด้วย .Net โปรแกรม Pet Shop นี้เป็น reference application ที่พัฒนาบน J2EE โดย Sun ไมโครซอฟท์แสดงให้เห็นว่า .Net นั้นทำให้ code สั้นลง หลายเท่าตัว ทำให้โปรแกรมทำงานเร็วขึ้นหลายเท่า เป็นต้น

#### ข้อดี .Net Framework

1. มีระบบไลบรารีที่เป็นมาตรฐานเดียวกัน เนื่องจากมีไลบรารีที่เป็นมาตรฐานเดียวกัน ทั้งหมด ทำให้เราไม่ต้องคอยกังวลว่าภาษาที่เราจะใช้เขียนนั้นมีไลบรารีตัวนั้นหรือตัวนี้หรือไม่ รวมทั้งไม่ต้องคอยระวังว่าจะใช้ไลบรารีของภาษาหนึ่งแล้วอีกภาษาหนึ่งจะไม่มีไลบรารีตัวนี้
2. ไม่ขึ้นกับระบบปฏิบัติการ เนื่องจากระบบปฏิบัติการที่แต่ละบุคคลหรือองค์กรใช้นั้นย่อม ไม่เหมือนกัน แต่การเขียนโปรแกรมเพื่อให้รันภายใต้ .NET นั้นขอเพียงแค่ว่า .NET Framework ติดตั้งอยู่บนโปรแกรมที่เขียนก็สามารถทำงานบนระบบปฏิบัติการนั้นได้ กล่าวคือ .NET Framework ช่วยให้เราสามารถนำโปรแกรมไปใช้งานได้ในทุกระบบปฏิบัติการ
3. ใช้ภาษาในการพัฒนาได้ทุกภาษา ทำให้เราไม่ต้องคอยมาศึกษาภาษาใหม่ๆ เมื่อต้องการ สร้างโปรแกรมในแต่ละครั้ง นอกจากนี้เรายังสามารถเลือกใช้ภาษาที่เราถนัดที่สุดในการพัฒนา โปรแกรมต่างๆ ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. มีการควบคุมสภาพแวดล้อมในการทำงานเป็นอย่างดี เนื่องจากเป็นระบบที่เป็นมาตรฐาน ทำให้การควบคุมจัดสรรระบบต่างๆ ทำได้ง่าย ไม่ว่าจะเป็นการจัดสรรหน่วยความจำ การใช้งานเครื่องจะสามารถกระทำได้อย่างรวดเร็วขึ้น ลดโอกาสที่เครื่องจะแฮงค์ได้เป็นอย่างดี

5. ความปลอดภัยที่มีมากขึ้น .NET Framework สามารถกำหนดสิทธิการใช้งาน หรือ permission ของผู้ใช้งานได้มากขึ้น ทำให้สามารถกำหนดได้ว่าโปรแกรมในส่วนใดใช้งานได้หรือไม่ได้ แล้วแต่เฉพาะบุคคลไป รวมทั้งมีฟังก์ชันไว้สำหรับถอดและเข้ารหัสข้อมูลอีกด้วย

ทั้งหมดนี้เป็นเพียงประโยชน์ส่วนหนึ่งในแนวความคิดของไมโครซอฟท์ที่กำลังจะพัฒนาให้สำเร็จเท่านั้น บางข้ออาจจะทำได้สำเร็จแล้ว แต่บางข้อก็ยังไม่สำเร็จ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้