



Client-Server Architecture

System Architecture :

สถาปัตยกรรมระบบ ประกอบด้วย

- ฮาร์ดแวร์ (Hardware)
- ซอฟต์แวร์ (Software)
- ระบบเครือข่าย (Network)
- กระบวนการประมวลผล (Data Process)

ผลลัพธ์ที่ได้มาจาก

System Design Specification





Client-Server Architecture

Software Architecture :

เป็น Abstract view ของระบบซอฟต์แวร์ที่ต่างจากรายละเอียดของการสร้างอัลกอริทึม และการนำเสนอข้อมูล สถาปัตยกรรมเป็นส่วนหนึ่งของกลยุทธ์ทางธุรกิจขององค์กรที่กำลังมีความสำคัญเพิ่มมากขึ้น สถาปัตยกรรมซอฟต์แวร์สามารถ

- ❑ ช่วยให้ซอฟต์แวร์มีความยืดหยุ่นและปรับเปลี่ยนได้ง่าย เพื่อสอดคล้องต่อธุรกิจที่มีการเปลี่ยนแปลงอยู่ตลอดเวลา
- ❑ ทำให้เกิดความร่วมมือในการทำงานร่วมกันระหว่างภาคธุรกิจ อุตสาหกรรม
- ❑ ช่วยให้มีการนำไปใช้ต่อในภาคธุรกิจ อุตสาหกรรม
- ❑ ช่วยให้นักพัฒนาให้ความสำคัญไปทำงานที่มีความเฉพาะเจาะจงยิ่งขึ้น ซึ่งมีประโยชน์ต่อการแข่งขันในภาคธุรกิจ อุตสาหกรรม





Client-Server Software Development

Software Architecture : (cont.)

- ☐ สามารถที่จะใช้เป็นเครื่องมือทางด้านการขายและทางการตลาด
- ☐ ช่วยลดค่าใช้จ่ายในการดูแลรักษาและการพัฒนา โดยใช้เทคโนโลยีที่เหมาะสม พร้อมทั้งจะขยายได้ในอนาคต
- ☐ ช่วยในการจัดการทรัพยากรบุคคล และ สำหรับการบริหารและติดตามโครงการ
- ☐ สร้างกลุ่มคำศัพท์ที่มีความหมายที่เป็นที่เข้าใจในองค์กร
- ☐ ช่วยลดเวลาในการเรียนรู้และพัฒนา (reuse component)
- ☐ ช่วยในการเขียน TOR และประเมินผลก่อนการลงทุนหรือว่าจ้างผู้พัฒนา
- ☐ ตอบสนองความต้องการด้าน Functional Requirement





Client-Server Architecture

Software Architecture : (cont.)

- ❑ ตอบสนองความต้องการด้าน Non-Functional Requirement หรือบางที่เรียกว่า Quality Attributes ซึ่งประกอบด้วย
 - ความปลอดภัย (Security)
 - ประสิทธิภาพ (Performance)
 - ความง่ายต่อการใช้งาน (Usability)
 - การที่ต้องทดสอบได้ในทุกๆส่วน (Testability)
 - การนำกลับมาใช้ใหม่ (Reusability)
 - ความเชื่อถือได้ของระบบ (Reliability)
 - การโยกย้ายระบบ (Portability) เช่นการเปลี่ยนฐานข้อมูลจาก MS SQL ไปเป็น Oracle
 - ความง่ายในการบำรุงรักษา (Maintainability)
 - การทำงานร่วมกับระบบอื่น (Interoperability)





Client-Server Architecture

Software Architecture : (cont.)

- ❑ ตอบสนองความต้องการด้าน Non-Functional Requirement
 - ความสามารถในการประยุกต์ (Adaptability) เช่นการประยุกต์ใช้กับงานได้หลากหลายโดยการออกแบบและพัฒนาเพียงครั้งเดียว
 - ความถูกต้องของระบบและข้อมูลในระบบ (Correctness)
 - ความง่ายในการเปลี่ยนแปลงแก้ไข (Flexibility)
 - ความสามารถในการรองรับปริมาณการใช้งาน (Robustness)





Client-Server Architecture

Software Architecture : (cont.)

ในทางสถาปัตยกรรมซอฟต์แวร์แล้วการวิเคราะห์มีความสำคัญมาก ดังเห็นได้จากวงจรการพัฒนาซอฟต์แวร์ที่ต้องมี

'Analysis and Design'

นั่นคือ เมื่อสถาปนิกซอฟต์แวร์ออกแบบสถาปัตยกรรมซอฟต์แวร์แล้ว ไม่ได้หมายความว่าสำเร็จสมบูรณ์ แต่ผลงานการออกแบบนั้นจำเป็นต้องนำวิเคราะห์เพื่อประเมินด้านคุณภาพและความคุ้มค่าในการลงทุนหรือวิเคราะห์ประโยชน์จากผลการออกแบบว่าตอบสนองต่อความต้องการทางธุรกิจ (Business Goals) มากน้อยเพียงใด

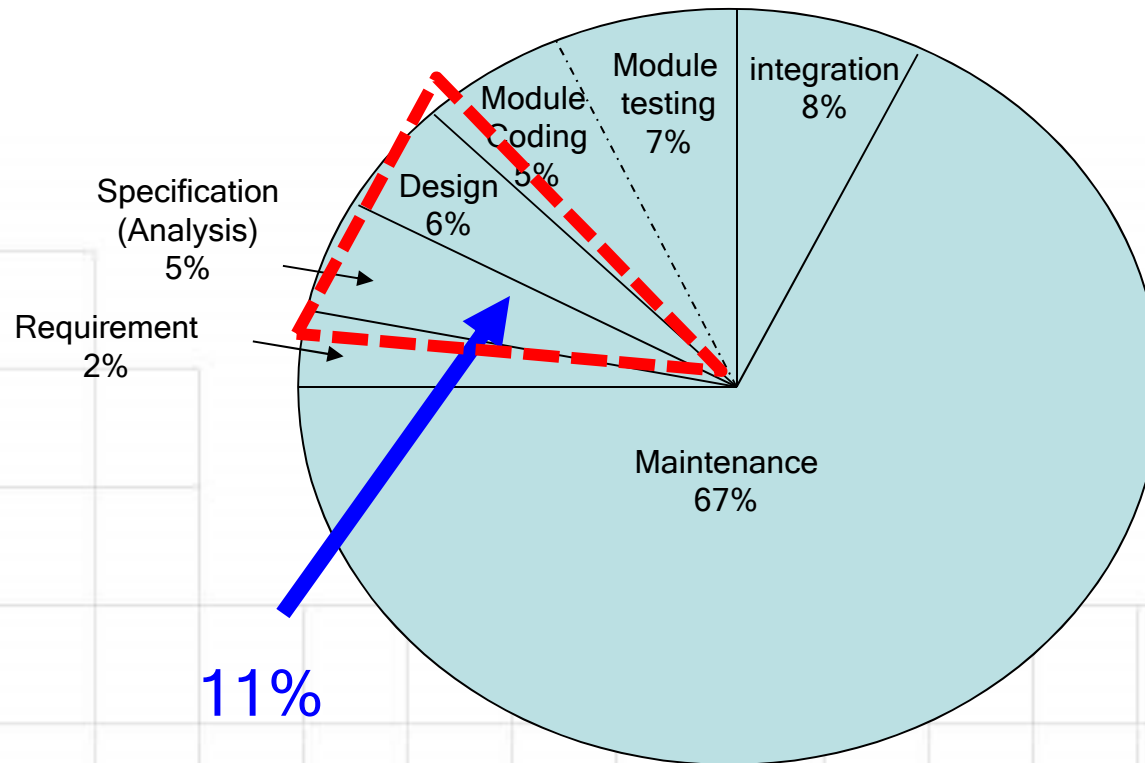




Client-Server Architecture

วงจรการพัฒนาซอฟต์แวร์

Approximate relative costs of the phases of the software life cycle.
(Stephen R. Schach, 1996, P.11)



Maintenance is so important, a major aspect of software engineering consists of those techniques, tools, and practices that lead to a reduction in maintenance cost.





Client-Server Architecture

INTERNET & INTRANET

Intranet

คือ ระบบเครือข่ายภายในองค์กร เป็นบริการ และการเชื่อมต่อคอมพิวเตอร์ เหมือนกันอินเทอร์เน็ต แต่จะเปิดให้ใช้เฉพาะสมาชิกในองค์กรเท่านั้น เช่น อินทราเน็ตของธนาคารแต่ละแห่ง หรือระบบเครือข่ายมหาดไทย ที่เชื่อมตาสถาบันกลางทั่วประเทศ เป็นต้น เป็นการสร้างระบบบริการข้อมูลข่าวสาร ซึ่งเปิดบริการ คล้ายกับอินเทอร์เน็ตเกือบทุกอย่าง แต่ยอมให้เข้าถึงได้เฉพาะคนในองค์กรเท่านั้น เป็นการจำกัดขอบเขตการใช้งาน ดังนั้นระบบอินเทอร์เน็ตในองค์กร ก็คือ "อินทราเน็ต" นั่นเอง แต่ในช่วงที่ชื่อนี้ยังไม่เป็นที่นิยม ระบบอินทราเน็ต ถูกเรียกในหลายชื่อ เช่น Campus network, Local internet, Enterprise network เป็นต้น

ระบบปิด (Close System)





Client-Server Architecture

INTERNET & INTRANET

Internet

เครือข่ายคอมพิวเตอร์นานาชาติ ที่มีสายตรงเชื่อมต่อไปยังสถาบันหรือหน่วยงานต่าง ๆ เพื่ออำนวยความสะดวกให้แก่ผู้ใช้ทั่วโลก. ผู้ใช้เครือข่ายนี้สามารถสื่อสารถึงกันได้ทางอีเมล สามารถสืบค้นข้อมูลและสารสนเทศ รวมทั้งคัดลอกเพิ่มข้อมูลและโปรแกรมมาใช้ได้. อย่างไรก็ตาม มีผู้เปรียบเทียบว่า อินเทอร์เน็ตเป็นเหมือนทางหลวงระหว่างประเทศ แต่ละประเทศจะต้องมีถนนเข้ามาเชื่อมต่อเข้าไปในประเทศ กล่าวคือ จะต้องมีการข่ายภายในรับช่วงต่ออีกทอดหนึ่ง (เช่น เครือข่ายภายในมหาวิทยาลัย, องค์กร หรือเครือข่ายของผู้ให้บริการอินเทอร์เน็ต) มิฉะนั้นก็จะใช้ไม่ได้ผล

ระบบเปิด (Open System)





Client-Server Architecture

สิ่งสำคัญประการหนึ่งคือการตัดสินใจว่าจะนำ Application ไปวางไว้ใน Server หรือ Client ดี

1. Fat Client (Traditional)

- ติดตั้งโปรแกรมที่ทำหน้าที่ต่าง ๆ ลงบนเครื่อง Client (มากกว่า Server)
- เป็น Model Client-Base / Client-Server แบบดั้งเดิม
- การบำรุงรักษายากลำบาก (Admin ปวดหัว)

2. Fat Server

- ติดตั้งโปรแกรมที่ทำหน้าที่ต่าง ๆ ลงบนเครื่อง Server
- บ่อยครั้งที่ Client ใช้เพียง Web browser
- ง่ายในการจัดการเพราะมี Software อยู่เพียงที่เดียวคือ Server ทำให้เปลี่ยน/แก้ไขได้ง่ายกว่า Fat Client ที่มีปริมาณมาก

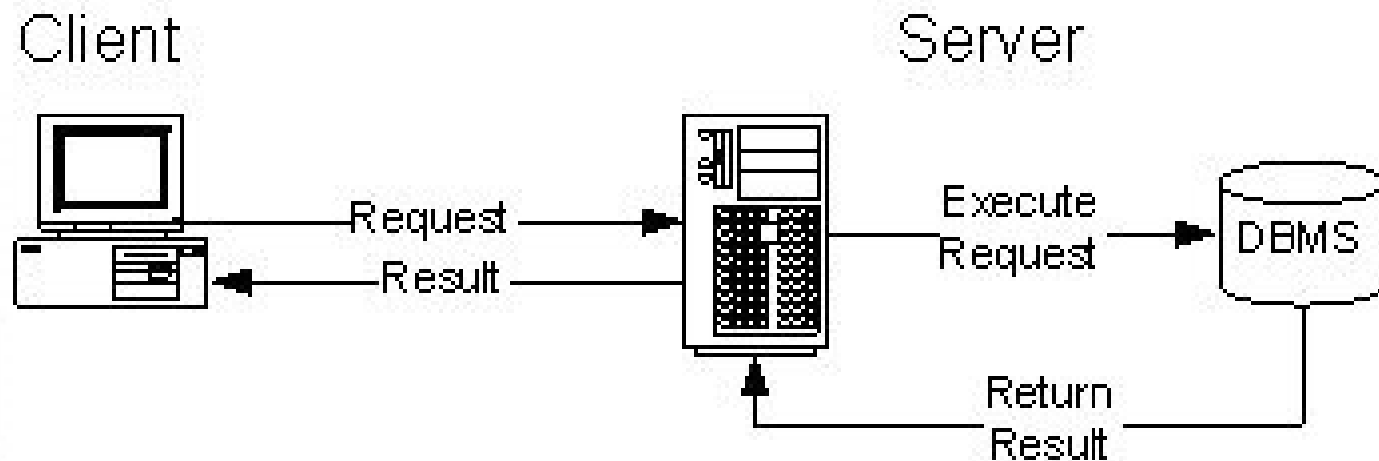




Client-Server Architecture

สิ่งสำคัญที่ต้องพิจารณาคือ การใช้ Tiers ในการทำ Client/Server App.

1. 2 Tier



(ตัวอย่างของ 2 Tier เช่น File Server, Web Server)





Client-Server Architecture

สิ่งสำคัญที่ต้องพิจารณาคือ การใช้ Tiers ในการทำ Client/Server App.

1. 2 Tier (ต่อ)

ประโยชน์ของ 2 Tier

1. ง่ายในการออกแบบ และ ดำเนินการ
2. มี Visual builder tools หลายตัวที่ง่ายในการออกแบบ Interface โปรแกรม Client / Server แบบ 2 Tier

ปัญหาของ 2 Tier

1. ไม่สามารถ Scale up หากมีการเพิ่ม Client or Hardware โปรแกรมจะทำงานได้ดีถ้าจำนวนผู้ใช้ไม่มาก แต่จะล้มเหลวเมื่อต้องให้บริการกับผู้ใช้จำนวนมาก
2. การบริหารจัดการ และการ Deployment จะเป็นเหมือนฝันร้ายถ้าในฝั่ง Client Tier มีส่วนของโปรแกรมที่ทำหน้าที่สำคัญ (ทางแก้ปัญหาคือ ทำเป็น 3 Tiers)





Client-Server Architecture

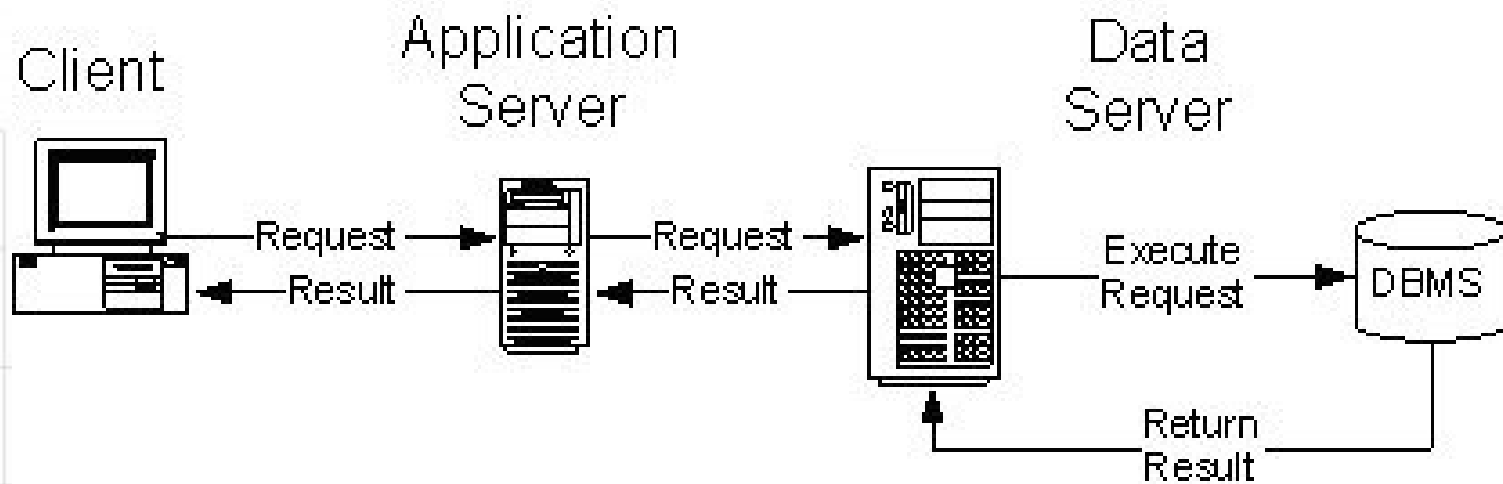
สิ่งสำคัญที่ต้องพิจารณาคือ การใช้ Tiers ในการทำ Client/Server App.

2. 3 Tier

Tier 1 : คือ Client

Tier 2 : คือ Server ที่ให้บริการ application

Tier 3 : คือ Server ที่ให้บริการข้อมูล หรือทรัพยากรอื่น ๆ





Client-Server Architecture

สิ่งสำคัญที่ต้องพิจารณาคือ การใช้ Tiers ในการทำ Client/Server App.

2. 3 Tier (ต่อ)

- more scalable (รองรับการขยายขนาด)
- more robust (รองรับงานปริมาณมากได้)
- more flexible (มีความยืดหยุ่นสูง)
- easier to manage / maintain (ง่ายในการบริหารจัดการ / บำรุงรักษา)
- easier to deploy (ง่ายในการ deploy)
- can integrate data from multiple sources (สามารถรวมข้อมูลจากหลาย ๆ แหล่งรวมถึง application เก่า)
- can provide better security (ความปลอดภัยดีขึ้น)

สรุป 3-Tier application ก็มีความยากในการพัฒนา
อย่างไรก็ตาม ปัจจุบันมีเครื่องมือสนับสนุนมากมาย





Client-Server Architecture

สิ่งสำคัญที่ต้องพิจารณาคือ การใช้ Tiers ในการทำ Client/Server App.

3. N Tier

- สามารถพัฒนา Applications เป็นชิ้นเล็ก ๆ
- สามารถ Reuse Components จากโปรแกรม / transactions ที่ต่างกันได้
- ปลอดภัยมากขึ้น
- มีความยืดหยุ่นมากขึ้น และสามารถบำรุงรักษาได้ง่ายขึ้น

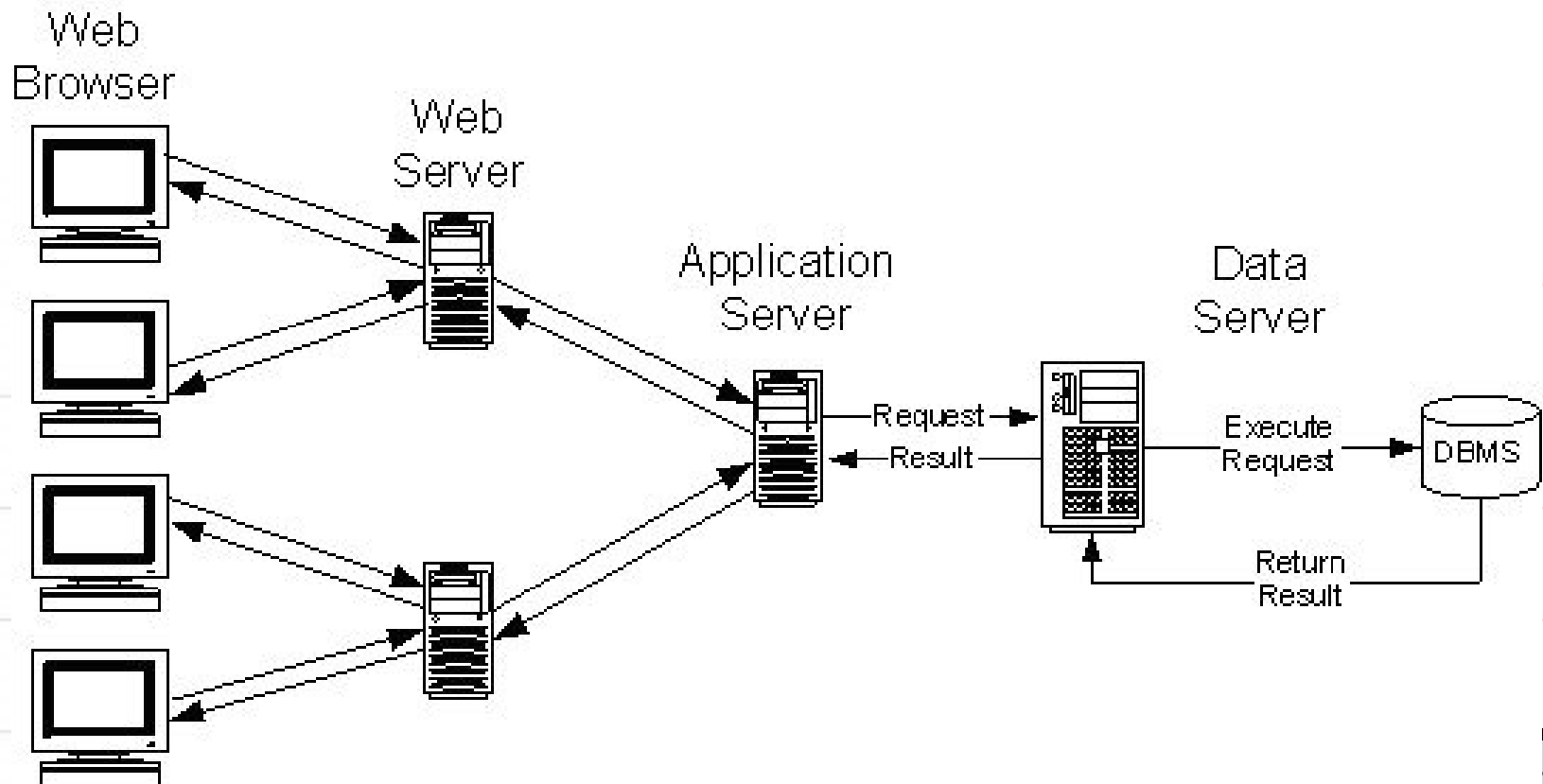




Client-Server Architecture

สิ่งสำคัญที่ต้องพิจารณาคือ การใช้ Tiers ในการทำ Client/Server App.

3. N Tier (ต่อ)





ใบงาน

ไฟล์งานชื่อ **Work1_19Nov19.docx**
Google Drive : [paijitwork->](#) ตามรหัสนักศึกษา

1. จงเขียนอธิบายความแตกต่างระหว่างสถาปัตยกรรมแบบเทอร์มินัลในแต่ละรูปแบบมาพอสังเขป พร้อมทั้งยกตัวอย่างการใช้งานตามแต่ละสถาปัตยกรรม (ตามความเข้าใจของนักศึกษา)
2. นักศึกษาคิดว่าในการพัฒนาระบบด้วยสถาปัตยกรรมแบบ Client / Server ควรมีการกำหนดรูปแบบของสถาปัตยกรรมให้เป็นแบบ Fat Server หรือ Fat Client หรือรูปแบบอื่น จงเขียนอธิบายพอสังเขป
3. จงยกตัวอย่างธุรกิจในปัจจุบันที่มีการวิเคราะห์และออกแบบด้วยสถาปัตยกรรมแบบระบบข้อมูลเป็นศูนย์กลาง และแนบรูปแบบสถาปัตยกรรมที่นำไปใช้ พร้อมอธิบายพอสังเขป (ตามความเข้าใจ)

//ฝากแจ้งเพื่อนที่ไม่เข้าเรียน ให้จัดส่งภายหลังโดยระบุรหัสไว้หลังชื่อไฟล์

