

Architectural Patterns/Styles

1. Choose 2 project

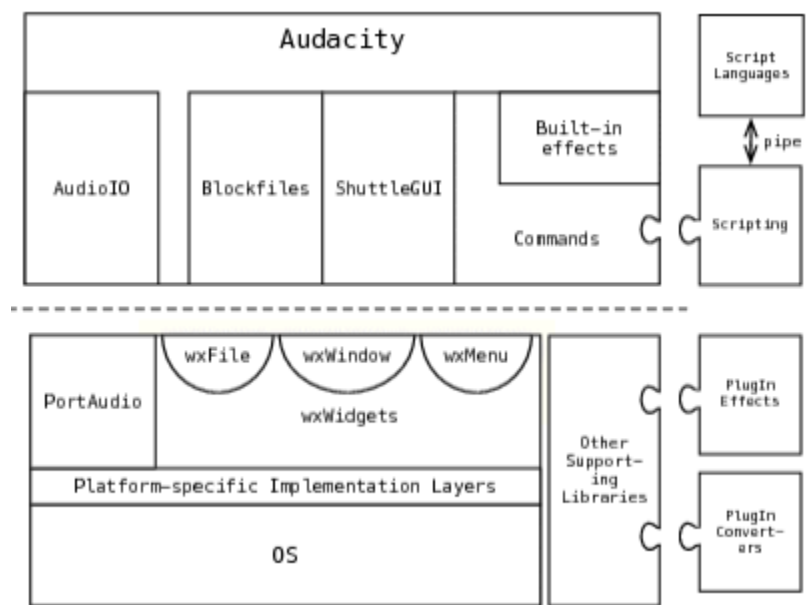
- Audacity

○ Purpose of the project

- ต้องการสร้างแพลตฟอร์มเพื่อปรับแต่งเสียงต่างๆ โดยพัฒนา effect เพื่อให้ใช้ในการแต่งเสียง รวมถึงการปรับระดับของเสียง
- พัฒนาเครื่องมือเพื่อแก้ปัญหาเกี่ยวกับอัลกอริทึมของการประมวลผลเสียงเพลง

○ Architectural patterns

- Audacity architectural patterns แบบ Microkernel (Plug-in) โดยแยกส่วน core system และส่วนที่เป็น Plug-in ของ core system ในรูปแบบ layout



○ Quality Attribute Scenarios

- Usability

Source:	User
Stimulus:	Learn to use
Environment:	Run Time
Response:	Help facilitate
Response Measure:	Satisfaction

- Integrability

Source:	User
Stimulus:	Want to add a plugin to Audacity?
Environment:	Deployment, Deployment, Runtime, Integration)
Response:	New Configuration
Response Measure:	There is a new plugin.

- Performance

Source:	Hacker
Stimulus:	Insecure libraries
Environment:	Plugin Online
Response:	Data, Resource
Response Measure:	Intrusion detection devices

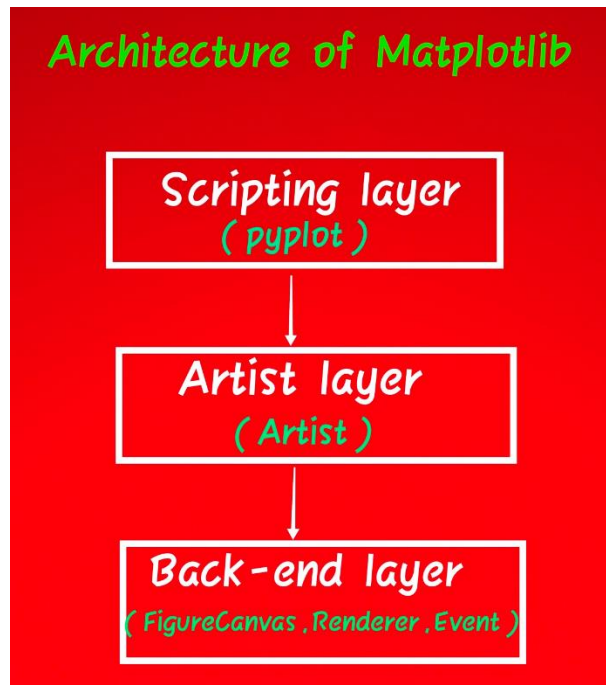
- Matplotlib

○ Purpose of the project

- เพื่อให้สามารถสร้างพล็อตที่มีคุณภาพที่มีความง่ายและความสะดวกมากยิ่งขึ้น
- เพื่อแสดงผลภาพแบบคงที่และแบบเคลื่อนไหวด้วยภาษา Python

○ Architectural patterns

- แบ่ง layer ของตัว library ออกเป็น 3 ส่วน ดังนี้
 1. Scripting Layer เป็น Layer ชั้นบนสุดที่ออกแบบมาเพื่อให้ Matplotlib ทำงานเหมือนสคริปต์ MATLAB เป็นชุดของฟังก์ชันรูปแบบคำสั่ง ดังนั้นจึงถือว่าเป็นเลเยอร์ที่ใช้งานง่ายที่สุด
 2. Artist Layer เป็น Layer ที่ช่วยให้สามารถควบคุมและปรับแต่งองค์ประกอบต่างๆได้มากที่สุด Layer นี้ประกอบด้วยวัตถุหลักหนึ่งชิ้นคือ Artist ที่ช่วยให้คุณปรับแต่งได้มากขึ้นเมื่อเทียบกับ Scripting Layer และ สะดวกกว่าสำหรับพล็อตขั้นสูง
 3. Backend Layer เป็น Layer ที่จัดการงานทั้งหมดผ่านการสื่อสารกับชุดเครื่องมือ เช่น wxPython หรือ PostScript และ Layer นี้เป็นชั้นที่ซับซ้อนที่สุดของ Matplotlib



○ Quality Attribute Scenarios

- Modifiability

Source:	Developer
Stimulus:	Wishes to modify 3D function
Artifact:	Code
Environment:	Development Time
Response:	Modification is made with no side effects
Response measure:	In Three hours

- Portability

Source:	OS
Stimulus:	Wishes to run on another OS
Artifact:	Resource

Environment: Runtime

Response: Can run without error occurs

Response measure: In 30 minutes

- Testability

Source: Tester

Stimulus: Performs end to end test

Artifact: Complete application

Environment: At deployment time

Response: Perform a test sequence

Response measure: Path coverage of 85% is achieved within three hours

Choose 1 project

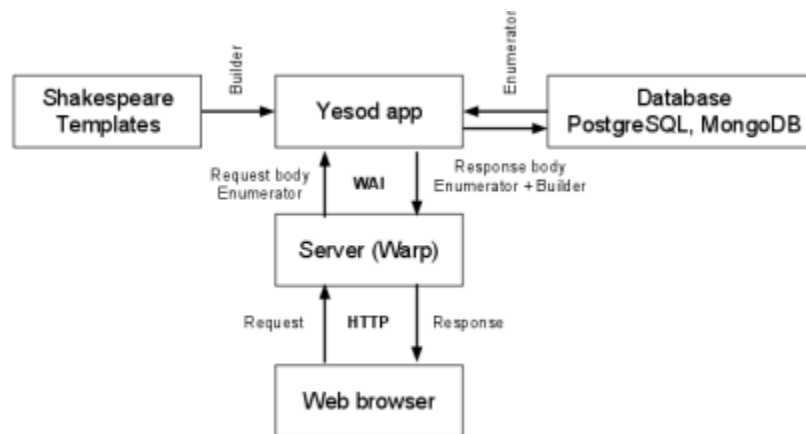
- Yesod

○ Purpose of the project

- การขยายจุดแข็งของ Haskell ไปสู่การพัฒนาเว็บ Yesod เน้นให้โค้ด กระชับที่สุดตรวจสอบโค้ดทุกบรรทัด ตอนเวลาคอมไพล์ให้มากที่สุดเท่าที่เป็นไปได้ เพื่อที่จะลดการใช้ไลบรารีขนาดใหญ่

○ architectural patterns

- client-server & N-Tier



○ quality attribute scenarios

- Performance : ลดการเรียกใช้ระบบและลดการสำนัปปเฟอร์รี่ให้น้อยที่สุด
 1. Buffer ไม่มาก
 2. หน่วยความจำ เหลือใช้
 3. application จะไม่ช้าลง
- Security : การทดสอบ WAI
 1. Req
 2. WAI test

3. access fake req
 4. WAI test
 5. not access
- Reliability : มีการใช้กฎจำนวนมากในฐานะข้อมูลเพื่อป้องกันไม่ให้ข้อมูลสูญหายและแจ้งข้อมูลความผิดพลาดกลับ
 1. เกิดการผิดพลาด
 2. กฎข้อกำหนดป้องกันข้อมูล
 3. ข้อมูลจะไม่สูญหา