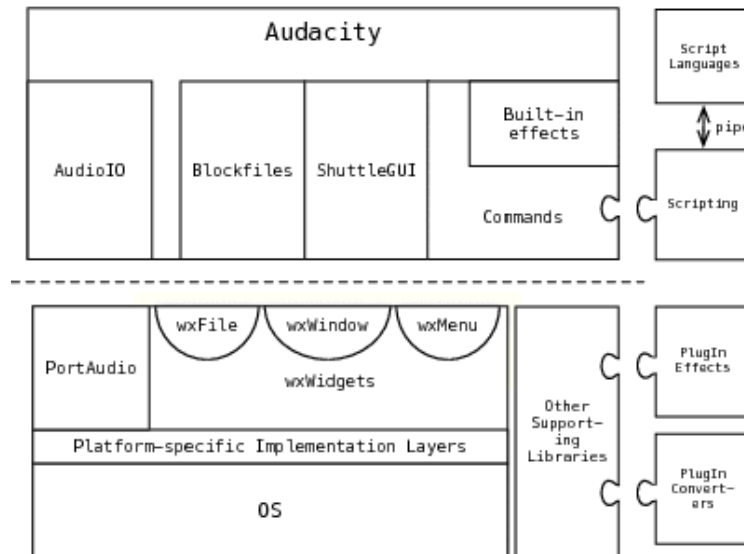


1. Audacity

Audacity คือ free software ที่ใช้สำหรับตัดต่อ, อัด, แก้ไข เสียง และ import/export แปลงไฟล์ไป-มาได้หลาย format

Architectural Patterns/Styles



รูป 1.0 diagram ที่แสดงถึงบางส่วนของ layers และ modules ใน Audacity

ซึ่ง Audacity นั้นใช้รูปแบบสถาปัตยกรรมแบบ Layer architectural โดยใน Audacity นั้น

จะแบ่ง Layer ตาม libraries โดยจากรูปที่ 1.0 จะมี libraries ที่สำคัญที่สุด 2 ตัว ได้แก่

1. wxWidgets GUI Library (provides GUI components in a cross-platform way) คือ

Library เดียวที่เป็น user interface โดย GUI จะถูกแบ่งออกมาเป็นหลายๆส่วน เช่น Blockfiles, ShuttleGUI

ซึ่งทำหน้าที่รับและแสดงinput และ output ของผู้ใช้

2. PortAudio Library (provides a low-level audio interface in a cross-platform way) คือ

Library เสียงที่ช่วยให้ Audacity สามารถเล่นและบันทึกเสียงในรูปแบบ cross-platform ได้ หากไม่มี

PortAudio Library โปรแกรม Audacity ก็จะไม่สามารถใช้ sound card ของอุปกรณ์ที่กำลังทำงานอยู่ได้

Quality attribute scenarios

- **Scenarios 1: Useability**

- Source (User)
- Stimulus (Export files)
- Artifact (Data, Interface)
- Environment (Run time)
- Response (หาก Export files สำเร็จจะมีการแจ้งเตือน)
- Response Measure (เวลาที่ใช้ในการ Export files)

- **Scenarios 2: Integrability**

- Source (User)
- Stimulus (ต้องการเพิ่ม Plugin)
- Artifact (System ทั้งหมด)
- Environment (Deployment, Runtime, Integration)
- Response (เพิ่ม Plugin โดยไม่มี errors)
- Response Measure (ผลกระทบที่มีต่อ component อื่นๆ)

- **Scenarios 3: Security**

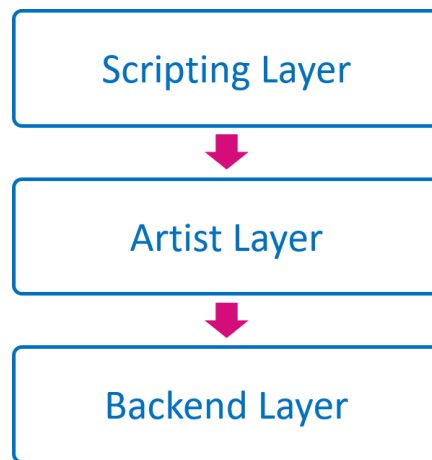
- Source (Hacker)
- Stimulus (นำเข้า Library ที่ไม่ปลอดภัย)
- Artifact (System services)
- Environment (Plugin Online, Runtime)
- Response (มีการปฏิเสธหากมีการนำเข้า Library ที่ไม่ปลอดภัย)
- Response Measure (ความครอบคลุมในการตรวจสอบ Library)

Source : <https://aosabook.org/en/audacity.html>

2. Matplotlib

Matplotlib คือ library ที่ครอบคลุม creating static, animated, and interactive visualizations ด้วย Python.

Architectural Patterns/Styles



รูป 2.0 ภาพแสดงส่วนประกอบของ Matplotlib

รูปแบบสถาปัตยกรรมที่ Matplotlib ใช้คือ Layer architectural ซึ่งประกอบด้วย

- **Scripting layer** เป็น layer ที่ใช้สำหรับการเขียนโปรแกรม โดยในส่วนนี้จะเป็นการเขียนโปรแกรมเพื่อใช้งานกับ Matplotlib โดยจะเขียนโปรแกรมเพื่อสร้างกราฟ
- **Artist Layer** เป็น layer ที่ช่วยให้ควบคุมและปรับแต่งองค์ประกอบต่างๆ ของ figure เช่น spines, tick direction, tick label size, tick label font, tick color. โดยในส่วนนี้จะเป็นการสร้าง figure และจะเป็นส่วนที่ผู้ใช้งานสามารถเขียนโปรแกรมเพื่อใช้งานกับ Matplotlib ได้
- **Backend Layer** เป็น layer ที่ซับซ้อนที่สุด ใช้สำหรับการแสดงผลของ figure โดยในส่วนนี้จะเป็นการแสดงผลของ figure ที่สร้างขึ้นมา

Quality attribute scenarios

- **Scenarios 1: 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 sours)

- **Scenarios 2: Portability**

- Source (OS)
- Stimulus (Wishes to run on another OS)
- Artifact (Resource)
- Environment (Run time)
- Response (Can run without error occurs)
- Response Measure (In 30 minutes)

- **Scenarios 3: 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)

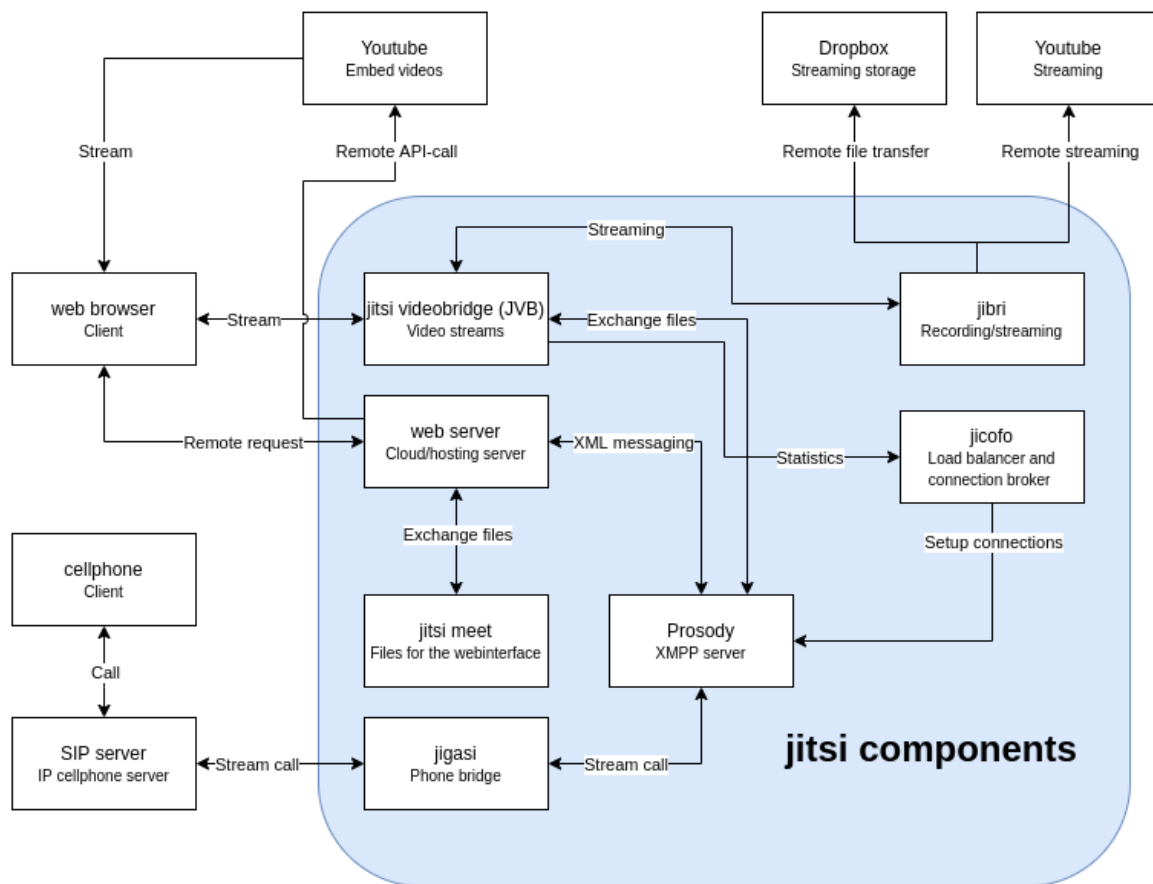
Source :

- <https://medium.datadriveninvestor.com/data-visualization-with-python-matplotlib-architecture-6b05af533569>
- <http://aosabook.org/en/matplotlib.html>

3. Jitsi

Jitsi คือ คอลเล็กชันของ Open-Source projects สำหรับการสร้าง video conference และ chat โดยใช้ WebRTC และ XMPP ซึ่งเป็นโปรเจกต์ที่เปิดโอกาสให้ผู้ใช้งานสามารถเข้ามาเพิ่มเติม แก้ไข และ ปรับปรุงโปรเจกต์ได้

Architectural Patterns/Styles



รูปที่ 3.0 แสดงส่วนประกอบของ Jitsi

รูปแบบสถาปัตยกรรมที่ Jitsi ใช้คือ Layer architectural ซึ่งประกอบด้วย

- **Jitsi Meet** เป็น แอปพลิเคชัน JavaScript ที่เข้ากันได้กับ WebRTC ซึ่งใช้ Jitsi Videobridge เพื่อ

จัดการประชุมทางวิดีโอคุณภาพสูงและปรับขนาดได้ สร้างจาก React และ React Native

- **Jitsi Videobridge (JVB)** เป็น เซิร์ฟเวอร์ที่เข้ากันได้กับ WebRTC ซึ่งออกแบบมาเพื่อกำหนดเส้นทาง

สตรีมวิดีโอระหว่างผู้เข้าร่วมในการประชุม

- **Jitsi Conference Focus (jicofo)** เป็น focus component ฝั่งเซิร์ฟเวอร์ที่ใช้ในการประชุมของ

Jitsi Meet ซึ่งจัดการ media sessions และทำหน้าที่เป็น load balancer ระหว่างผู้เข้าร่วมแต่ละคน

กับ videobridge.

- **Jitsi Gateway to SIP (jigasi)** เป็น server-side application ที่อนุญาตให้ regular SIP clients

เข้าร่วมการประชุมของ Jitsi Meet

- **Jitsi Broadcasting Infrastructure (jibri)** เป็นชุดเครื่องมือสำหรับบันทึกและ/หรือสตรีมการประชุม

ของ Jitsi Meet ที่ทำงานโดยเปิดใช้ Chrome instance ที่แสดงผลในเฟรมบัพเฟอร์เสมือน ซึ่งบันทึกและเข้ารหัส

เอาต์พุตด้วย ffmpeg

Quality attribute scenarios

● Scenarios 1: Useability

- Source (User)
- Stimulus (ทำการ video conference ผ่าน Jitsi)
- Artifacts (Interface)
- Environment (Runtime)
- Response (มีการแจ้งเตือนหากเชื่อมต่อสำเร็จ)
- Response Measure (เวลาที่ใช้ในการเชื่อมต่อ)

● Scenarios 2: Modifiability

- Source (Developer)
- Stimulus (ต้องการเพิ่ม screen sharing function)
- Artifacts (Code)
- Environment (Development time)
- Response (สามารถเพิ่มฟังก์ชันได้โดยไม่กระทบระบบส่วนอื่น)
- Response Measure (ผลกระทบที่เกิดขึ้นในระบบ)

● Scenarios 3: Security

- Source (User)
- Stimulus (ต้องการเข้าร่วม video conference ผ่าน Jitsi โดยไม่ได้รับอนุญาต)
- Artifacts (System services)
- Environment (The System is either online)
- Response (ป้องกันการเข้าใช้โดยไม่ได้รับอนุญาต)
- Response Measure (ไม่สามารถเข้าร่วมได้โดยไม่ได้รับอนุญาต)

Source :

- <https://jitsi.github.io/handbook/docs/architecture/?fbclid=IwAR0jx3OqNPMmjdZRENO8Fiwz7gsYvDWbRWSJR9nSKZ1GwHNwoKUHUYEvR9A>
- <http://aosabook.org/en/jitsi.html>