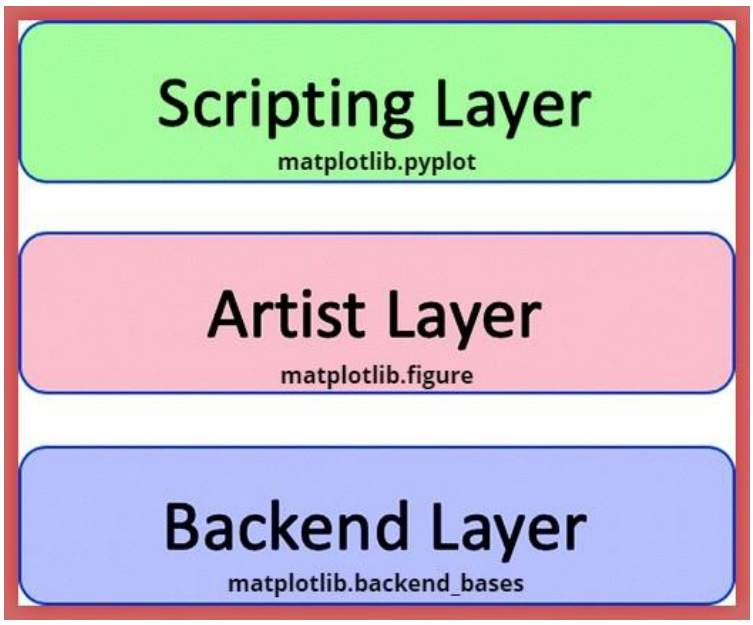
Matplotlib

วัตถุประสงค์ (Purpose):

Matplotlib เป็นไลบรารีที่ครอบคลุมสำหรับการสร้างการแสดงภาพแบบสแตติก (Static) ภาพเคลื่อนไหว (Animated) และแบบภาพจำลองที่สามารถโต้ตอบได้ (Interactive Visualization) ใน Python

รูปแบบสถาปัตยกรรม (Architectural Pattern):

รูปแบบสถาปัตยกรรมของ matplotlib เป็นแบบเลเยอร์ (Layer) แบ่งออกได้เป็น 3 เลเยอร์ ได้แก่ Scripting Layer, Artist Layer และ Backend Layer โดยเลเยอร์ชั้นบนจะสามารถติดต่อเลเยอร์ที่อยู่ระดับต่ำกว่าและติดกันได้ แต่จะไม่สามารถติดต่อกับเลเยอร์ที่อยู่ต่ำกว่านั้นได้



Matplotlib's Architectural Pattern

ref. Mastering Matplotlib: Part 1. Understanding Matplotlib Architecture… | by Lawrence Alaso Krukrubo | DataSeries | Medium

Backend Layer

เป็นเลเยอร์ที่อยู่ล่างสุดจะมีการใช้งาน Abstract interface class ประกอบด้วย FigrureCannvas, Renderer และ Event โดย

FigureCanvas (matplotlib.backend_bases.FigureCanvas): เป็นส่วนที่ทำหน้าที่เหมือนกับพื้นที่หรือพื้นผิวที่จะสำการวาดบางสิ่งลงไป (เช่น กระดาษ)

Renderer (matplotlib.backend_bases.Renderer): เป็นส่วนที่ทำการวาดบางสิ่ง (เช่น พู่กัน)

Event (matplotlib.backend_bases.Event): เป็นส่วนที่คอยรับ input ต่างๆเช่น คีย์บอร์ด เมาส์

Artist Layer

เป็นชั้นที่อยู่ถัดขึ้นมาจาก Backend Layer มี base class เป็น matplotlib.artist.Artist ใน Artist layer ไม่ได้ทำการวาดภาพโดยตัวเองแต่จะมีฟังก์ชัน draw ที่ไปทำการเรียก API Renderer จาก Backend Layer Artist layer แบ่งออกเป็น 2 ชนิดได้แก่

Primitive Artist: เปรียบเสมือน object ที่เรามองเห็นในการแสดงภาพ เช่น Line2D, Rectangle, Circle และ Text

Composite Artist: เป็นส่วนประกอบของ Artist เช่น Axis, Trick, Axes และ Figure

Scripting Layer

เป็นเลเยอร์ที่อยู่บนสุดอยู่ในรูปแบบของ interface matplotlib.pyplot ที่เรียกใช้ API จากชั้น Artist Layer โดยการรวบรวมเอาชุดคำสั่งต่างๆจาก Artist Layer ทำให้ใช้งานง่ายเหมาะสำหรับโปรแกรมเมอร์ทั่วไป

Quality Attribute Scenario

- **Modifiability:**

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

แหล่งอ้างอิง:

- Mastering Matplotlib: Part 1. Understanding Matplotlib Architecture… | by Lawrence Alaso Krukrubo | DataSeries | Medium
- https://www.aosabook.org/en/matplotlib.html

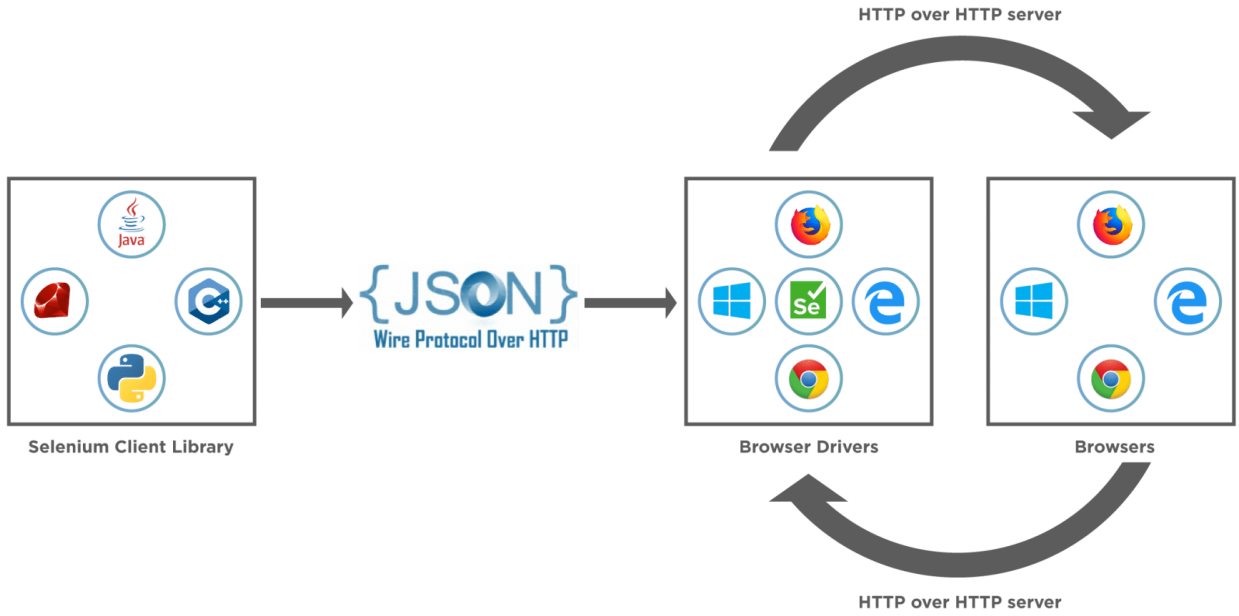
Selenium

จุดประสงค์ (Purpose):

Selenium เป็น browser automation ที่ส่วนใหญ่มักจะใช้ในการทำการทดสอบแบบ end-to-end สำหรับ web application โดย Selenium จะทำงานแบบเข้าไปควบคุม browser ให้ทำงานนั้นซ้ำๆ

รูปแบบสถาปัตยกรรม (Architectural Pattern):

เราอนุมานว่า Selenium WebDriver ไม่ใช่เครื่องมือทดสอบแบบ standalone แต่ประกอบไปด้วยแต่ประกอบไปด้วยส่วนประกอบที่หลากหลายในการที่จะใช้งานเครื่องมือได้แก่ Selenium WebDriver Client Libraries, JSON Wire Protocol, Browser Drivers และ Browsers



Selenium's Architectural Pattern

Ref. [What is Selenium WebDriver Architecture? How Does it works? (toolsqa.com)]

Selenium WebDriver Client Libraries/Language Bindings:

เป็น library ที่ประกอบไปด้วยคำสั่งต่างๆในรูปแบบของไฟล์ .jar ที่สามารถทำงานร่วมกับ Selenium protocol/W3C Selenium protocol ได้ โดย Selenium Client Library แบ่งออกได้เป็น 2 อย่างได้แก่

WebDriver Protocol clients:

เป็นเสมือนชั้นบางๆที่ห่อหุ้มรอบ WebDriver protocol HTTP โดยขึ้นอยู่กับภาษาที่ผู้ใช้งานเลือกใช้ ไลบรารีจะสามารถเพิ่มเข้ามาในตอนเริ่มต้นโปรเจคได้

WebDriver-based tools:

เป็นไลบรารีระดับสูง (higher-level libraries) ที่เราใช้ในการทำ WebDriver Automation โดยเครื่องมือนี้จะทำงานโดยผ่านฟังก์ชันไลบรารี่ระดับต่ำ (lower-level libraries)

Selenium WebDriver รองรับหลากหลายภาษาเช่น Java, C#, Python, Ruby, PHP

JSON WIRE PROTOCOL Over HTTP Client:

Selenium WebDriver เป็น REST API มีหน้าที่ในการสื่อสารระหว่างระบบที่แตกต่างกัน (Heterogeneous system) กล่าวคือแต่ละ browser มี HTTP protocol ของตัวเอง โดยใช้ JSON ในการติดต่อสื่อสารระหว่าง client libraries และ drivers JSON requests ที่ถูกส่งโดย client จะถูกเปลี่ยนเป็น HTTP requests เพื่อที่จะถูกส่งไปยัง server และจะถูกเปลี่ยนกลับไปเป็น JSON อีกทีเมื่อถูกส่งกลับไปยัง client อีกครั้ง

Browser Drivers:

Browser Driver เปรียบเสมือนสะพานเชื่อมในการสื่อสารระหว่าง Selenium libraries กับ browser ที่จะช่วยให้คำสั่งของ Selenium สามารถทำงานได้บน browser ได้ โดยการสื่อสาร Browser Driver กับ Browser จะไม่มีการเปิดเผยลอจิกภายในของฟังก์ชัน เมื่อ Browser Driver ได้รับคำสั่งจะดำเนินการตามคำสั่งนั้นแล้วตอบกลับเป็นรูปแบบ HTTP response

Browser:

Selenium รองรับหลาย browser เช่น Chrome, Safari, Firefox, Opera, Internet Explorer และรองรับหลายระบบปฏิบัติการ (Operating System) เช่น Windows, Mac OS, Linux, Solaris

Quality Attribute Scenarios:

- **Modifiability:**

Source:	Developer
Stimulus:	Wishes to add functionality to support more languages.
Artifact:	Code
Environment:	Developer Time

Response: Modifications were made without errors and worked correctly

Response measure: In 5 hours

- **Portability:**

  Source: OS

  Stimulus: Wishes to run on another OS

  Artifact: Resource

  Environment: Runtime

  Response: Can run without error occurs

  Response measure: In 1 hour

- **Usability:**

  Source: Users

  Stimulus: Minimize impact of errors

  Artifact: System

  Environment: Runtime

  Response: Wishes to cancel current operations

  Response measure: Cancellation takes less than 1 second

แหล่งอ้างอิง:

- What is Selenium WebDriver Architecture? How Does it works? (toolsqa.com)
- Selenium Architecture - Detailed Explanation - InterviewBit
- Selenium WebDriver Architecture | Software Testing Material - Software Testing Material
- The Architecture of Open Source Applications: Selenium WebDriver (aosabook.org)

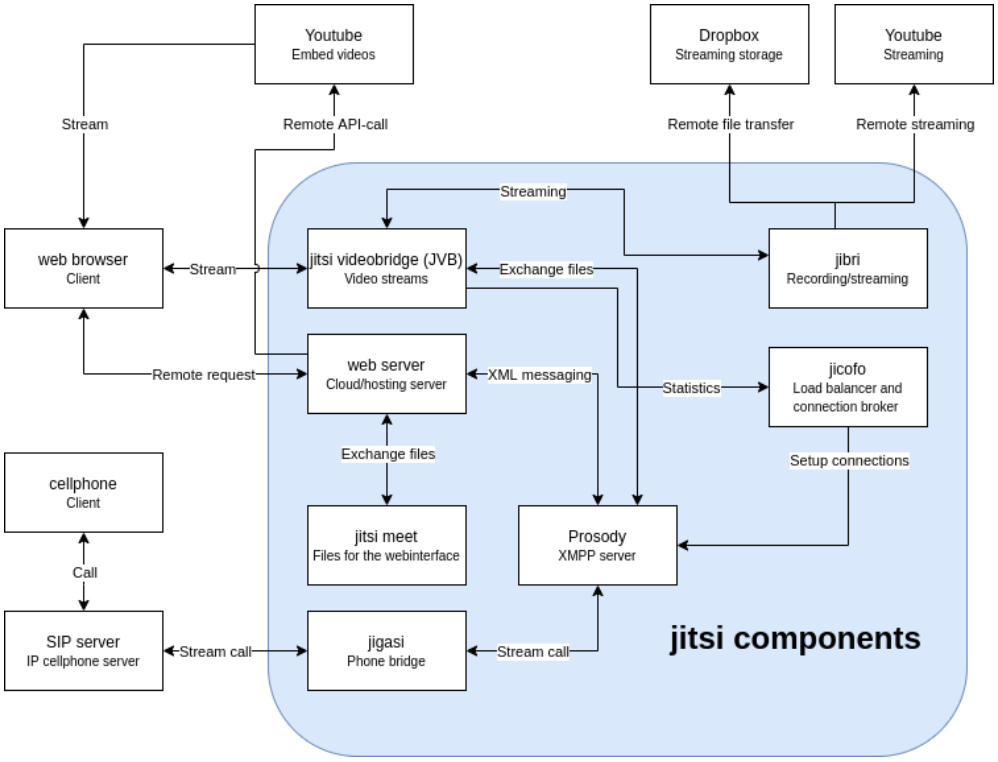
Jitsi

จุดประสงค์ (Purpose):

Jitsi เป็นแอปพลิเคชันที่ให้ผู้ใช้งานเริ่มการสนทนาผ่านวีดีโอและเสียง สตรีมหน้าจอของตนเอง แลกเปลี่ยนไฟล์ และข้อความกันได้ แอปฯอนุญาตให้ผู้ใช้งานดำเนินการสิ่งต่างๆผ่านโปรโตคอลต่างๆ ตั้งแต่ XMPP มาตรฐาน (Extensible Messaging and Presence Protocol) และ SIP (Session Initiation Protocol) ไปจนถึงกรรมสิทธิ์เช่น Yahoo! และ Windows Live Messenger (MSN)

รูปแบบสถาปัตยกรรม (Architectural Pattern):

Jitsi meet ใช้สถาปัตยกรรมแบบ Bridge ซึ่งใช้การแยกส่วนประกอบ (Components) ต่างๆ ออกเป็นส่วนย่อย เพื่อให้ง่ายเมื่อต้องการที่จะแก้ไขเปลี่ยนแปลงส่วนใดส่วนหนึ่งโดยไม่ส่งผลต่อแอปพลิเคชันทั้งหมด Jitsi Meet ประกอบด้วยส่วนประกอบหลายอย่างที่พัฒนาขึ้นเองทั้งหมด ไม่มีการใช้คอมโพเนนต์ของบุคคลที่สาม (Third Party Component) ยกเว้น web server ที่ใช้ Jitsi ประกอบไปด้วย Jitsi Meet, Jitsi videobridge (JVB), Jicofo, Jibri, เซิร์ฟเวอร์ SIP, Jigasi และ Prosody



Jitsit's Architectural Pattern

ref. [Architecture | Jitsi Meet]

Jitsi Meet:

ไฟล์สำหรับเว็บอินเตอร์เฟส (web interface) เข้าถึงผ่านไฟล์ที่ให้บริการโดยเว็บเซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์เริ่มต้นคือ Prosody

Jitsi videobridge (JVB):

บริการเชื่อมต่อวิดีโอที่ให้บริการสตรีมวิดีโอแก่ผู้เข้าร่วมทั้งหมด รวมถึงอัลกอริธึมสำหรับการกระจายคุณภาพของวิดีโอ JVB เป็นเซิร์ฟเวอร์รีเลย์อัจฉริยะที่ทดสอบแบนด์วิดท์ที่พร้อมใช้งานสำหรับ client เพื่อกำหนดว่าสตรีมใดที่จะถ่ายทอด โดยขึ้นอยู่กับสิ่งที่ client กำลังดูอยู่ รวมถึงการปิดสตรีมบางรายการเนื่องจากข้อจำกัดแบนด์วิดท์

Jicofo:

Jitsi Conference Focus (JICOFO) ดำเนินการ server side และสร้างขึ้นสำหรับการเริ่มต้นการเชื่อมต่อระหว่างผู้ใช้และบริดจ์ (bridge) วิดีโอ

Jibri:

ชุดเครื่องมือสำหรับบันทึก และ/หรือ สตรีมการประชุม Jitsi Meet ที่ทำงานโดยเปิดใช้ Chrome ในเฟรมบัฟเฟอร์เสมือน (virtual framebuffer) และบันทึกและเข้ารหัสเอาต์พุตด้วย ffmpeg

SIP server:

อนุญาตให้ผู้ใช้ที่มีการเชื่อมต่ออินเทอร์เน็ตที่ช้า สามารถเข้าร่วมการสนทนาผ่านเสียงด้วยการเชื่อมต่อโทรศัพท์มือถือ / โทรศัพท์พื้นฐาน

Jigasi:

การเชื่อมต่อระหว่าง videobridge และเซิร์ฟเวอร์ SIP

Prosody:

เซิร์ฟเวอร์ XMPP ฟรีภายนอกที่ทำหน้าที่เป็นฐานของการตั้งค่าหลังบ้าน (back-end)

Quality Attribute Scenarios:

- **Usability:**

Source:	Developer
Stimulus:	Wishes to add screen sharing function
Artifact:	Code
Environment:	Development time
Response:	Modifications were made without side effects
Response measure:	In 6 hours

- **Modifiability:**

Source:	Developer
Stimulus:	Wishes to add screen sharing function
Artifact:	Code
Environment:	Development time
Response:	Modifications were made without side effects
Response measure:	In 6 hours

- **Portability:**

Source:	iOS
Stimulus:	Wishes to run on iOS
Artifact:	Resource
Environment:	Runtime
Response:	Can run on iOS without error occurs
Response measure:	In 30 minutes

แหล่งอ้างอิง:

- Jitsi Meet: The Architectural Decisions - DESOSA
- Jitsi Meet Architecture Description - Developers - Jitsi Community Forum - developers & users
- The Architecture of Open Source Applications: Jitsi (aosabook.org)