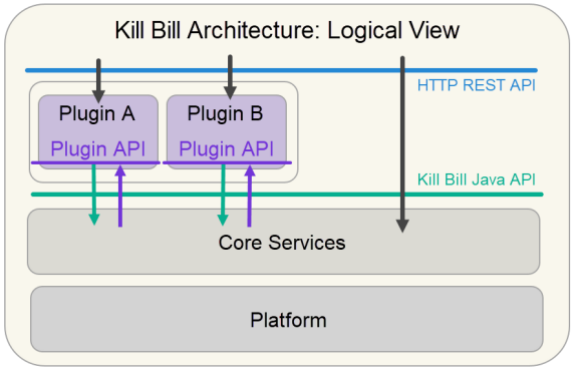
Purpose of the project : Kill Bill

open-source software ช่วยให้ผู้ใช้สร้าง Solution รูปแบบการเรียกเก็บเงินที่กำหนดเองได้ มี API เพื่อสร้างแบบจำลองการกำหนดราคาต่างๆ จัดการการสมัครรับข้อมูล สร้างใบแจ้งหนี้ รวมกับ Gatewayการชำระเงินต่างๆ ดูแลการสร้างใบแจ้งหนี้ จัดการเครดิต และเรียกชำระเงิน
ช่วยเชื่อมต่อ Gateway การชำระเงินและกระบวนการการสมัครสมาชิกเรียกเก็บเงิน
เช่น Netflix , Amazon , PayPal

Kill Bill Architecture

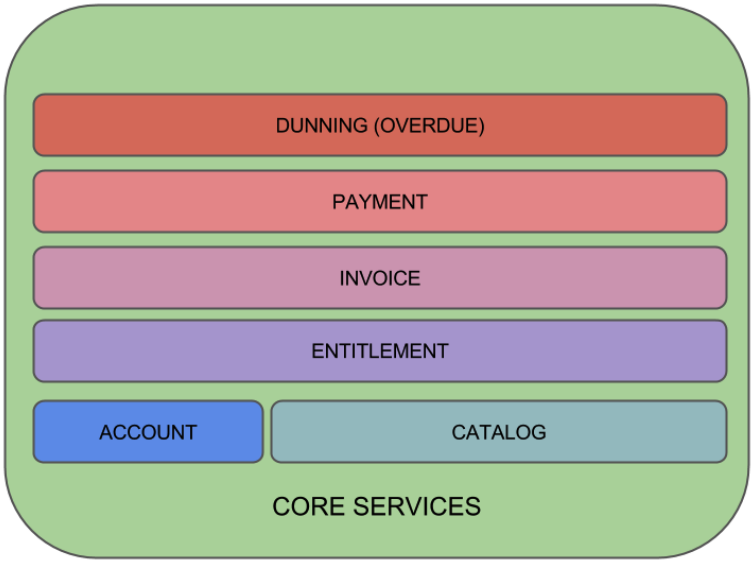


https://killbill.io/blog/kill-bill-platform-open-source-open-data-open-architecture/

1. Platform
   จัดการ Life cycle สำหรับ Core Services initializing, starting, stopping และอื่นๆ
   Event-based ทุกครั้งที่มีการเปลี่ยนแปลงในระบบ ระบบจะปล่อยเหตุการณ์ และทั้งระบบจะรับรู้ถึงเหตุการณ์นั้น
2. Core Services
   บริการหลักการเรียกเก็บชำระเงิน การสมัครสมาชิก การให้สิทธิ์ การออกใบแจ้งหนี้ การชำระเงิน การติดตามหนี้
3. Plugin
   Gateway การชำระเงิน หรือ Plugin ที่มีผลต่อธุรกิจ
4. JAX-RS Layer
   HTTP REST API

Kill Bill: Billing System Architecture (Core Services)

ระบบการเรียกเก็บชำระเงิน

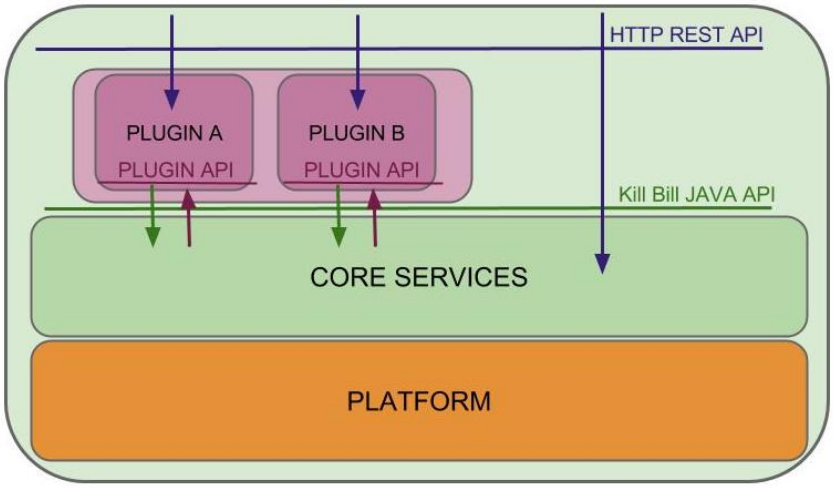


https://killbill.io/blog/kill-bill-billing-system-architecture/

1. Catalog
   Catalog service มีหน้าที่ให้ข้อมูลผลิตภัณฑ์ กฎการจัดตำแหน่ง ราคาที่เกี่ยวข้องกับผู้เช่า
   โดยใช้ API ดึงข้อมูล
2. Entitlement
   service API จัดการข้อมูลการให้สิทธิ์ทั้งหมด ที่เกี่ยวข้องกับการสมัครรับข้อมูล สถานะ (เริ่มต้น,หยุดชั่วคราว, ดำเนินการต่อ,หยุด) การเลือกกดรับข้อมูล การชำระเงินแบบ Subscription
   สามารถสร้าง Event Bus ให้ service อื่นที่มา Subscribe ได้
3. Invoice
   บริการใบแจ้งหนี้มี API เพื่อดึงใบแจ้งหนี้ หรือเรียกใบแจ้งหนี้ในอนาคต
   สามารถสร้าง Event Bus ให้ service อื่นที่มา Subscribe ได้
4. Payment
   ระบบการชำระเงินมี API เพื่อดึงข้อมูลการชำระเงิน/การคืนเงินที่ผ่านมา หรือเพื่อเรียกการชำระเงิน/คืนเงินใหม่ สามารถสร้าง Event Bus ให้ service อื่นที่มา Subscribe ได้
5. Dunning (Overdue)
   ระบบค้างชำระที่สามารถกำหนดได้ เมื่อเกิดเหตุการณ์ชำระเงินล้มเหลวหรือผู้ใช้ไม่ได้ชำระเงิน ดำเนินการยุติการให้บริการ แจ้งให้ผู้ใช้ทราบ
   สามารถสร้าง Event Bus ให้ service อื่นที่มา Subscribe ได้

จากระบบด้านบน จะมีการเรียกใช้ API Cross service กัน อาจจะทำให้เกิดปัญหา Loop ได้เพราะดังนั้น การเรียกใช้ API Cross service จะเรียกจาก Top Layer สู่ Bottom Layer เท่านั้น เช่น บริการชำระเงิน (Payment) สามารถเรียก API ไปยังระบบใบแจ้งหนี้ได้ (Invoice) แต่ไม่สามารถเรียกย้อนกลับได้ แต่ระบบนั้นมี Publish-Subscribe ให้แต่ละ service สามารถ Subscribe เพื่อรับ Event Bus จาก service อื่นๆ ได้ Event Bus จะรับประกันการส่งในแบบ Asynchronous เพื่อไม่ให้เกิดการเรียกแบบสายยาว long chain

Kill Bill: Plugins Architecture



https://killbill.io/blog/kill-bill-plugins-architecture/

ระบบแต่ละระบบต้องการความยืดหยุ่นและการปรับ Business logic อย่างมาก จึงมีความจำเป็นในการเลือกทำเป็น Module เพื่อตอบสนองต่อ Event ของระบบไปจนถึงการผสานการทำงานกับ Third party การออกแบบ Plugin framework แบบ OSGi ช่วยให้ Kill Bill เพิ่ม feature ได้มากมาย เช่น lifecycle, isolation การแยกตัว

First type of plugin

Payment plugin Kill Bill ไม่ทราบเกี่ยวกับ Payment gateway แต่ละบัญชีมีวิธีการชำระเงินหลายแบบ (บัตรเครดิต บัญชี PayPal) โดยวิธีการชำระเงินแต่ละวิธีจะเชื่อมโยงไป plugin ที่กำหนด ใครชำระเงินรูปแบบไหน ก็จะใช้ Plugin แบบนั้นมาทำงาน

Second type of plugin

Notification plugin ใช้ตอบสนองต่อ Event ของระบบ เช่น การสร้างบัญชีหรือข้อผิดพลาดในการชำระเงิน ตัวอย่างเช่น

- Zendesk plugin จะรับฟังการสร้างบัญชีและอัปเดต event และ copy ข้อมูล Kill Bill ลงใน Zendesk ทั้ง Kill Bill และ Zendesk ระบบจะ sync กันอยู่เสมอ โดยผู้ดูแล Zendesk สามารถเข้าถึงข้อมูลของ Kill Bill ได้ตามเฉพาะที่กำหนด

ทั้ง Plugin 2 ประเภทนี้

สามารถเข้าถึง Kill Bill API ได้ครบชุด ทำให้ผู้พัฒนาสามารถเขียน Plugin ต่างๆได้มากมาย

Attribute scenarios

Testability

Kill Bill มี Tools ในการสร้างแบบจำลองการเรียกชำระเงิน หรือเกี่ยวกับการชำระเงินต่างๆ เพื่อนำมาทดสอบได้

Dev test new Plugin

- Source of stimulus: Unit testers, System testers, Integration testers testes
- Stimulus: ทำ Unit tests, System tests, Integration tests

  Unit test Validate การเรียกเก็บเงินวันที่ 31 กุมภาพันธ์

  Validate system functions

  Validate Plugin functions
- Artifacts: Code, Plugin, Service , the entire system
- Environment: Run time
- Response: ผลการ test
- Response measure: เวลาที่ใช้ test , fail error ของ Plugin , Core Services ทำงานได้เต็มประสิทธิภาพ

Integrability

การเชื่อมต่อระบบ Core Services กับ Plugin payment

- Source of stimulus: new Plugin
- Stimulus: Integrate new Plugin with existing service
- Artifacts: Plugin
- Environment: Integration time
- Response: Changes are integrated
- Response measure: ค่าใช้จ่าย , เวลา

Modifiability

Kill Bill ถูกพัฒนาตามหลัก OSGi Module / Plugin มีหน้าที่ในการทำงาน เมื่อการชำระเงินนั้นมีหลากหลายรูปแบบทั้งการโอน บัตรเครดิต บัญชี PayPal ผู้พัฒนาก็สามารถพัฒนา Plugin การชำระเงินรูปแบบนั้นมาเชื่อมต่อกับ Core Service ได้

Increase Cohesion Split module

Developer , Project Owner ต้องการ add new Plugin ที่เกี่ยวกับธุรกิจตัวเอง

- Source of stimulus: Developer , Project Owner
- Stimulus: add functionality
- Artifacts: Code
- Environment: Run time
- Response: make modification , Core Services รองรับการทำงานของ Plugin
- Response measure: ค่าใช้จ่าย , เวลา , ความซับซ้อนของระบบ

Source:

https://stackshare.io/stackups/killbill-vs-stripe

Kill Bill: plugins architecture

https://killbill.io/blog/kill-bill-plugins-architecture/

Kill Bill: Billing System Architecture

https://killbill.io/blog/kill-bill-billing-system-architecture/

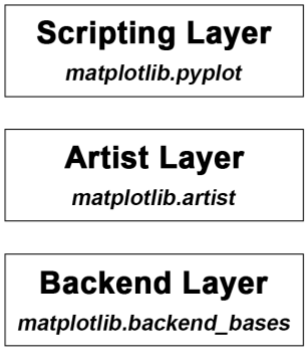
Testability

https://docs.killbill.io/latest/features.html

Purpose of the project : matplotlib

เป็น Library Python ในการสร้างภาพจากข้อมูล Plot 2D 3D ทำ Data visualization และการ Plot graph มีประโยชน์สำหรับการวิเคราะห์ข้อมูล และ Machine Learning Matplotlib ถูกออกแบบมาให้ผู้ใช้งานสามารถสร้าง plot ได้ง่ายเพียงไม่กี่คำสั่ง ให้มีความยืดหยุ่นหลากหลายวิธีในการสร้าง plot

Matplotlib Architecture



Three main layers in Matplotlib architecture. Source: Jun Ye's Blog

ประกอบด้วย 3 main layers

1. Backend Layer

ทำหน้าที่จัดการ งานหนัก ๆ ทั้งหมด โดยสื่อสารไปยังชุด drawing toolkits บนเครื่อง

เป็น Layer ที่มีความซับซ้อนมากที่สุด

มี 3 abstract interface classes

1.1 FigureCanvas (แผ่นกระดาษเปล่า)

matplotlib.backend_bases.FigureCanvasBase เหมือนแผ่นผ้าใบที่รอการวาด

1.2 Renderer (แปรง)

matplotlib.backend_bases.RendererBase abstract base class เพื่อจัดการการวาดหรือแสดงผลลงบน FigureCanvas

1.3 Event (กิจกรรม จะวาดอะไร)

matplotlib.backend_bases.Event จัดการ Input ของผู้ใช้ เช่นการคลิปเมาส์หรือคีย์บอร์ด

2. Artist Layer

ส่วนที่ให้ผู้ใช้สามารถควบคุมและเลือกปรับแต่งองค์ประกอบต่าง ๆ ของแบบ Figure

Artist ซึ่งใช้ Renderer เพื่อใช้วาดภาพบน canvas

Artist สามารถปรับค่า figure ได้ เช่น Line2D , Rectangle , Circle , Axis แกน x,y , Axes

3. Scripting Layer

Top Layer ที่ถูกออกแบบมาเพื่อให้ทำงานคล้าย Script ของ MATLAB

เป็นชุดของฟังก์ชั่นคำสั่ง ที่ใช้งานได้ง่าย ทำให้รวมทุกอย่างเข้าด้วยกันโดยอัตโนมัติ

ดังนั้นจึงใช้งานได้ง่ายกว่า Artist Layer

Attribute scenarios

Testability

matplotlib ใช้ pytest framework

การมีส่วนร่วมในการพัฒนา matplotlib ทุก code ที่เปลี่ยนแปลงทั้งหมดจะต้องผ่านการทดสอบ

- Source of stimulus: Contributor
- Stimulus: test set code
- Artifacts: A unit of code
- Environment: Development setup
- Response: Test result , code ที่เพิ่มเข้ามาไม่ส่งผลกระทบต่อส่วนอื่น
- Response measure: เวลา, ความซับซ้อน

Usability

ผู้ใช้สามารถควบคุม plot ที่สร้างขึ้นได้อย่างเต็มที่ มีเครื่องมือให้ผู้ใช้สร้างได้ง่าย

ผู้ใช้สามาใช้งานการสร้าง plot ได้หลากหลาย matplotlib สามารถตอบสนองความต้องการได้

ผู้ใช้ต้องการนำข้อมูล มาสร้าง Scatter plot เพื่อทำ Data visualization

- Source of stimulus: User, Data Science
- Stimulus: นำข้อมูลมากสร้าง plot
- Artifacts: Code
- Environment: run time ,system configuration time
- Response: สร้าง Scatter plot ให้
- Response measure: User ใช้งานได้เครื่องมือง่าย ความพึงพอใจของ User

Performance: เกี่ยวกับ เวลาในการตอบสนอง / การประมวลผล

ผู้ใช้ต้องการสร้าง plot จากข้อมูลจำนวนมาก และข้อมูลเปลี่ยนแปลงได้ง่าย

- Source of Stimulus: จำนวนของข้อมูล
- Stimulus: Flow rate ของข้อมูล, ข้อมูลที่เปลี่ยนแปลง
- Artifacts: system
- Environment: Normal mode, Peak mode
- Responds: ประมวลผลตอบสนองได้ทัน, ผล Plot graph มีความถูกต้อง
- Respond Measures: วัดการทำงาน Latency, ปริมาณ/จำนวนงานที่ผ่านเข้าระบบได้

Source :

https://subscription.packtpub.com/book/application-development/9781847197900/1/ch01lvl1sec01/merits-of-matplotlib

Testability

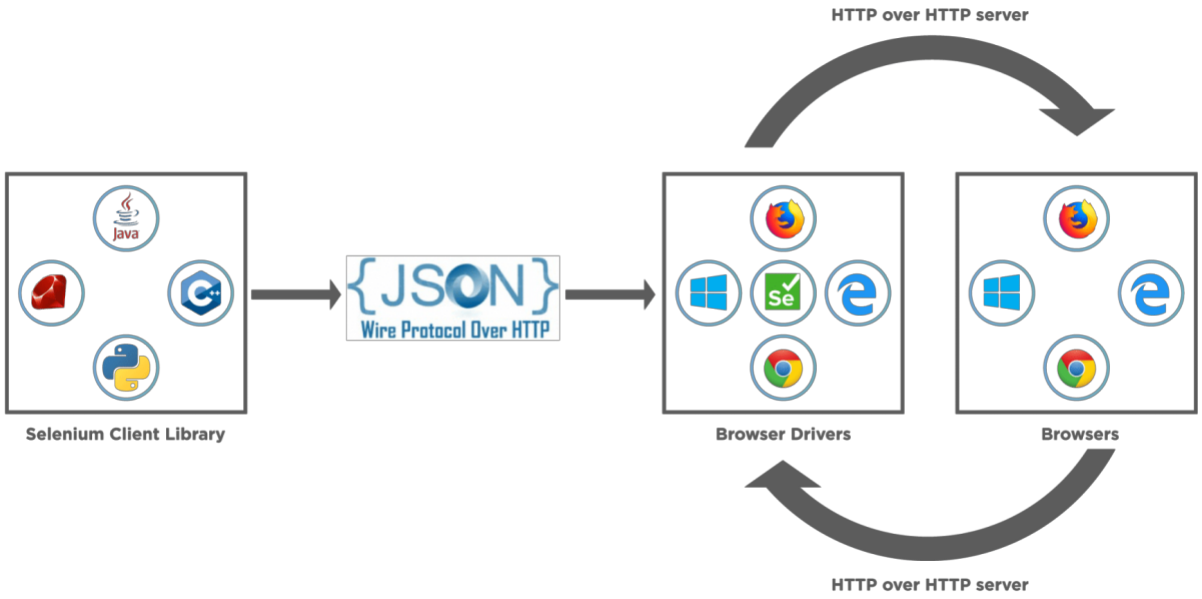
https://matplotlib-org.translate.goog/stable/devel/testing.html?_x_tr_sl=en&_x_tr_tl=th&_x_tr_hl=th&_x_tr_pto=op,wapp

Purpose of the project : Selenium WebDriver

โดยถ้าเราทำก็ต้องนั่งป้อน Input Event ต่างๆ Test ทุกๆ Requirement ด้วยมือของเราเอง ก็จะทำให้เสียเวลาและอาจเกิดข้อผิดพลาดของมนุษย์ในการ Test ได้

Selenium WebDriver เป็น open-source framework ที่สามารถใช้งานเพื่อทำการทดสอบเว็บแอปพลิเคชันโดยอัตโนมัติ มีความยืดหยุ่นสูงในการทดสอบกรณีทดสอบ function และ regression test cases เขียนได้ด้วยหลายภาษา และนำไปเทสทดสอบใน เบราว์เซอร์ต่างๆ เช่น Chrome, Safari, Firefox, Opera สามารถใช้งานได้ทั้ง Windows , MacOS , Linux

Selenium WebDriver Architecture



core selenium webdriver architecture and the major selenium components

1. Selenium WebDriver Client Libraries / Language Bindings
   รองรับการการใช้งานได้หลายภาษา เช่น Ruby , Python , Java
   มีการพัฒนาเชื่อมโยงหลายภาษา ส่วนที่เขียนโค้ด test
2. JSON WIRE PROTOCOL Over HTTP Client Top Layer
   ช่วยอำนวยความสะดวกในการสื่อสารทั้งหมดที่เกิดขึ้นใน Selenium ระหว่าง Browser และ Code (Selenium WebDriver Client Libraries) โดยหัวใจหลักคือ สื่อสำหรับการถ่ายโอนข้อมูลโดยใช้ RESTful (Representational State Transfer) API ซึ่งมีกลไกการส่งและกำหนด RESTful โดยใช้ JSON ผ่าน HTTP สร้างการเชื่อมต่อระหว่าง Browser Drivers และ client libraries

3. Browser Drivers

   Brower แต่ละอัน มี Driver แยกต่างหากเฉพาะตัวเอง โดย Browser Drivers จะสื่อสารติดต่อโดยที่ไม่ต้องรู้ Logic function การทำงานของ Browser นั้น เมื่อ Browser Drivers ได้รับคำสั่งใด ๆ คำสั่งนั้นจะถูกดำเนินการบน Browser นั้น ๆ และการตอบกลับไปในรูปแบบของ HTTP

4. Browser

   การทดสอบจะทำได้ก็ จำเป็นต้องมีการติดตั้ง Browser นั้นบน local หรือ server เครื่องที่เราต้องการทดสอบ

Attribute scenarios

Modifiability

รองรับการใช้งานได้หลากหลาย Browser ผู้ใช้ต้องการ test ระบบเว็ปไซต์ของตน ก็สามารถเลือกใช้ Selenium ไปทดสอบได้ ทั้งรองรับการพัฒนาที่หลากหลายภาษา และ test ได้หลาย Browser

- Software tester ต้องการ test บน website ต่าง Browser
- Source of stimulus: Software tester
- Stimulus: ต้องการ test script บน website
- Artifacts: Code test case
- Environment: Development time, Browser ต่างๆ
- Response: Test results
- Response measure: สามารถ test ได้ทุก test case, Test result ของ Brower ต่างๆ

Usability

Tester ต้องการเขียน test case script เพื่อนำไปใช้ทดสอบกับ website

- Source of stimulus: Software tester
- Stimulus: ต้องการเขียน test case script
- Artifacts: Code test case script
- Environment: Run time
- Response: มอบ Tools, features ให้ตามที่ Tester ต้องการ
- Response measure: Tester ใช้งานได้เครื่องมือง่าย ความพึงพอใจของ Tester

Testability

จุดที่เราต้องทดสอบ test script แบบอัตโนมัติ Selenium Grid ช่วยลดปัญหาได้ Selenium Grid สามารถสั่ง execution test script แบบ parallel ใน Brower ต่างๆ สามารถใช้ Selenium Grid บน Cloud และทำงานกับ Remote WebDriver เพื่อดำเนินการทดสอบกับคอมพิวเตอร์ระยะทางไกลได้

- Source of stimulus: Software tester
- Stimulus: ต้องการเขียน test case script แบบอัตโนมัติ ในหลาย ๆ Browser
- Artifacts: Code test case script
- Environment: Development time
- Response: test case script ถูก execute เพื่อทดสอบ test case ได้แบบอัตโนมัติในหลาย ๆ Browser
- Response measure: จำนวนข้อผิดพลาด , จำนวน test script ที่ถูกรัน , เวลา

Source:

https://www.linkedin.com/pulse/guide-selenium-webdriver-getting-started-test-amanda-d-cruz?trk=articles_directory

https://medium.com/edureka/selenium-webdriver-architecture-565e2db26dd5

https://www.toolsqa.com/selenium-webdriver/selenium-webdriver-architecture/

Usability

https://www.toolsqa.com/software-testing/what-is-usability-testing/

https://docs.testable.io/selenium/remote/overview.html