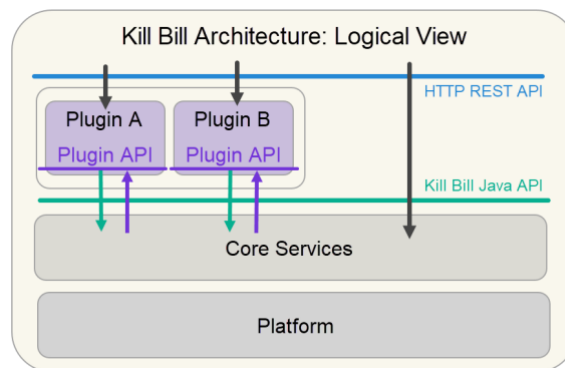


## Purpose of the project : Kill Bill

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

## Kill Bill Architecture

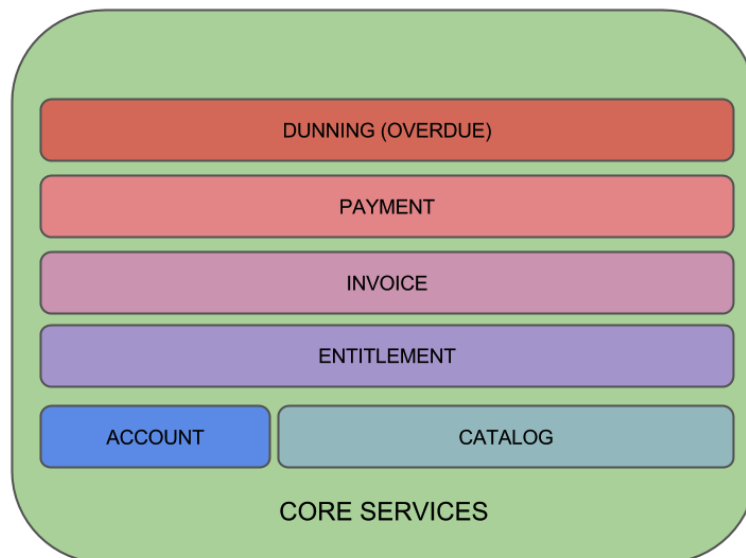


<https://killbill.io/blog/kill-bill-platform-open-source-open-data-open-architecture/>

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) การเลือกกรับข้อมูล 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

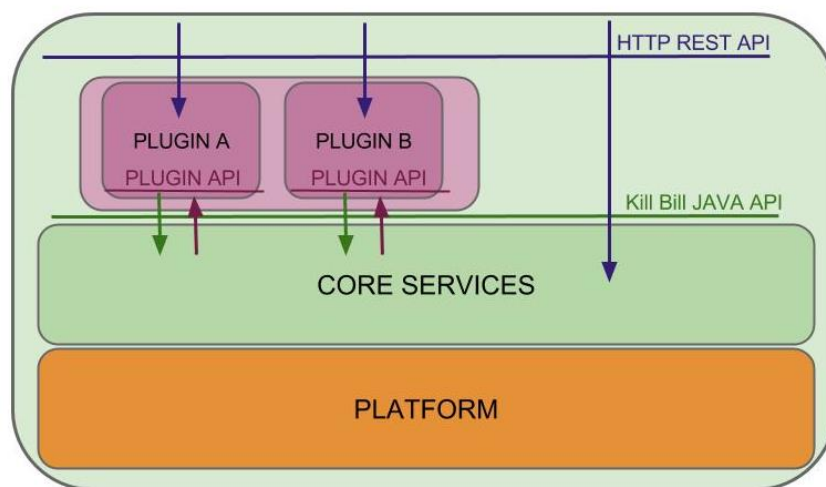
ระบบการชำระเงินมี API เพื่อดึงข้อมูลการชำระเงิน/การคืนเงินที่ผ่านมา หรือเพื่อเรียกการชำระเงิน/คืนเงินใหม่ สามารถสร้าง Event Bus ให้ service อื่นที่มา Subscribe ได้

### 5. Dunning (Overdue)

ระบบค้างชำระที่สามารถกำหนดได้ เมื่อเกิดเหตุการณ์ชำระหนี้ล้มเหลวหรือผู้ใช้ไม่ได้ชำระเงิน ดำเนินการยุติการให้บริการ แจ้งให้ผู้ใช้ทราบ สามารถสร้าง Event Bus ให้ service อื่นที่มา Subscribe ได้

จากระบบด้านบน จะมีการเรียกใช้ API Cross service กัน อาจจะทำให้เกิดปัญหา Loop ได้เพราะฉะนั้น 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 ประเภ่นี้

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>