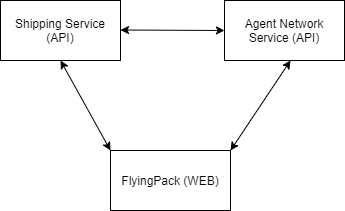
**ภาพรวมระบบ FlyingPack**



**Shipping Service**

* ระบบบ user จำแนกสิทธิ์ต่างๆ
* ระบบ authentication
  + Login และออก JWT ให้
* ระบบจัดการขนส่ง เรื่อง Booking
  + ตอนนี้เชื่อมต่อกับ Shippop เป็นหลัก
  + มี implement ของ Flash ไปแล้วแต่ไม่ได้ใช้งาน

**Agent Network Service**

* ระบบดูแลออร์เดอร์จากฝั่งตัวแทน
* บริการ Open API
  + Authentication ด้วย apiKey ที่แอดมิน JNA ออกให้
  + API สร้าง order จะไปเรียก API ของ Shipping Service ต่ออีกที

**Authentication**

* Shipping Service ออก token
* ใช้ JSON Web Token ด้วย HMAC
* แต่ละ service จะถือ secret key เดียวกันและแยก verify token แต่ละ service

**FlyingPack**

* เว็บสำหรับแสดงผล เรียกใช้งาน API ทั้งสอง service

**Note**

* ไม่มี API gateway เนื่องจาก service น้อยแค่สอง service

**รายละเอียดด้านเทคนิคต่างๆ**

* FlyingPack (Web)
  + Framework: Vue.js
  + Programming Language: JavaScript
  + เพิ่มเติม: <https://github.com/jna-distribution/web-flyingpack>
* Shipping Service (API)
  + Programming Language: Golang
  + ใช้ standard library เป็นหลัก
  + Database: PostgreSQL
  + เพิ่มเติม: <https://github.com/jna-distribution/service-shipping>
* Agent Network Service (API)
  + Programming Language: TypeScript
  + Framework: express
  + Database: PostgreSQL
  + เพิ่มเติม: <https://github.com/jna-distribution/service-agent-network>
  + Document สำหรับ Open API
    - ใช้ Docbox ในการ build
    - <https://github.com/jna-distribution/docs-agent-network-open-api-builder>
* Infrastructure
  + Heroku
* Version Control
  + Git และ Github

**การติดตั้งสำหรับพัฒนาและทดสอบ**

* Clone Repository <https://github.com/jna-distribution>

```

git clone <https://github.com/jna-distribution/service-shipping>

git clone <https://github.com/jna-distribution/service-agent-network>

git clone <https://github.com/jna-distribution/web-flyingpack>

git clone <https://github.com/jna-distribution/flyingpack-some-script>

git clone <https://github.com/jna-distribution/docs-agent-network-open-api-builder>

git clone <https://github.com/jna-distribution/docs-an>

```

* Config
  + เปลี่ยน .env.example เป็น .env และปรับค่า
* Database
  + pg\_restore -d database\_name -f db\_shippin.dump
  + pg\_resotre -d database\_name -f db\_an.dump
* Shipping service

```

go run cmd/restapi/main.go

```

* Agent network service

```

npm run dev

```

* Web Flyingpack

```

npm run serve

```

**API document (Postman Collection)**

* Service (Shipping, Agent Network, Robot)
  + Service Shipping.postman\_collection.json
  + Service AgentNetwork.postman\_collection.json
  + Service Robot.postman\_collection.json
* Provider (Shippop, Flash)
  + FlassExpress.postman\_collection.json
  + ShippopBusiness.postman\_collection.json

**การ Deploy**

* Branching Model
  + ตามนี้ <https://nvie.com/posts/a-successful-git-branching-model/>
  + แต่ไม่ได้ยึดติดมาก ยังไงก็ได้
  + ส่วนเลข Version กำหนดได้ตามใจเลย
* การ Deploy Service หลักๆ ดูได้ที่ Heroku (Agent Network, Shipping, and Web Flyingpack)
  + <https://dashboard.heroku.com/apps/service-agent-network/deploy/heroku-git>
  + <https://dashboard.heroku.com/apps/service-shipping/deploy/heroku-git>
  + <https://dashboard.heroku.com/apps/web-flyingpack/deploy/heroku-git>
* Agent Network Open API document

```

cd docs-agent-network-open-api-builder

npm run build

cd ..

cp ./docs-agent-network-open-api-builder/build/home.html ./docs-an/home.html

cd ./docs-an

git push heroku master:master

```