การเรียกใช้งานใน Clean Architecture จะเป็นไปตามลำดับการเรียกใช้ของแต่ละชั้น โดยทั่วไปแล้วจะมีลำดับดังนี้:

1. **Frameworks and Drivers (External Interfaces)**: (public\_box\_api)
   * Controllers, HTTP endpoints, UI components เป็นต้น ซึ่งจะรับคำขอจากผู้ใช้หรือระบบภายนอกและส่งคำขอไปยัง Use Cases.
2. **Use Cases (Application Layer)**: (public\_box\_api.application)
   * Use Cases จะเป็นตัวกลางในการประมวลผลคำขอ โดยจะใช้ Interface Adapters เพื่อทำงานกับ Entities และส่งคำขอไปยัง Data Sources.

การ validate ข้อมูลใน Application Layer มักจะเกิดขึ้นก่อนที่จะดำเนินการต่อไปยัง Interface Adapters (Infrastructure Layer) เพื่อการจัดเก็บหรือการแสดงผล โดยตรวจสอบข้อมูลตามเงื่อนไขทางธุรกิจหรือความสมเหตุสมผลก่อนที่จะดำเนินการใด ๆ ต่อไป.

ตัวอย่างของการ validate ที่อาจเกิดขึ้นใน Application Layer ได้แก่:

1. **ความสมเหตุสมผลของข้อมูล**: เช่น ตรวจสอบว่าชื่อผู้ใช้มีความยาวที่ถูกต้อง หรือว่าอีเมลล์มีรูปแบบที่ถูกต้อง.
2. **การตรวจสอบสิทธิ์**: ตรวจสอบว่าผู้ใช้มีสิทธิ์ในการดำเนินการนั้น ๆ หรือไม่ เช่น ตรวจสอบว่าผู้ใช้เป็นผู้ที่มีสิทธิ์ในการสร้างข้อมูลหรือทำการแก้ไข.
3. **การตรวจสอบความสมบูรณ์ของข้อมูล**: ตรวจสอบว่าข้อมูลที่ส่งเข้ามาครบถ้วนและถูกต้องตามที่ต้องการ.
4. **Interface Adapters (Infrastructure Layer)**: (public\_box\_api.Infrastructure)
   * Interface Adapters จะรับคำขอจาก Use Cases และใช้ Data Sources เพื่อดำเนินการตามคำขอนั้น ๆ และส่งผลลัพธ์กลับไปยัง Use Cases.
5. **Entities (Domain Layer)**: (public\_box\_api.domain)
   * Entities เป็นตัวแทนของข้อมูลหลักในโดเมนของแอปพลิเคชัน ซึ่งอาจถูกใช้โดยทั้ง Use Cases และ Interface Adapters ในกระบวนการประมวลผลคำขอ.

ดังนั้นลำดับการเรียกใช้งานโดยทั่วไปจะเป็นดังนี้: Frameworks and Drivers -> Use Cases -> Interface Adapters -> Entities

แต่ละชั้นก็สามารถเรียกใช้งานชั้นอื่น ๆ ภายในขอบเขตของตนเองได้ตามความเหมาะสมของการออกแบบและการประยุกต์ใช้งานของแต่ละโปรเจค ยกตัวอย่างเช่น Use Cases อาจจะเรียกใช้ Interface Adapters เพื่อดึงข้อมูลหรือ Entities เพื่อประมวลผลคำขอ และ Interface Adapters ก็สามารถเรียกใช้ Frameworks and Drivers เพื่อการสื่อสารกับระบบภายนอกได้ เป็นต้น โดยลำดับนี้จะขึ้นอยู่กับโครงสร้างและการออกแบบของแต่ละโปรเจค.

Top of Form