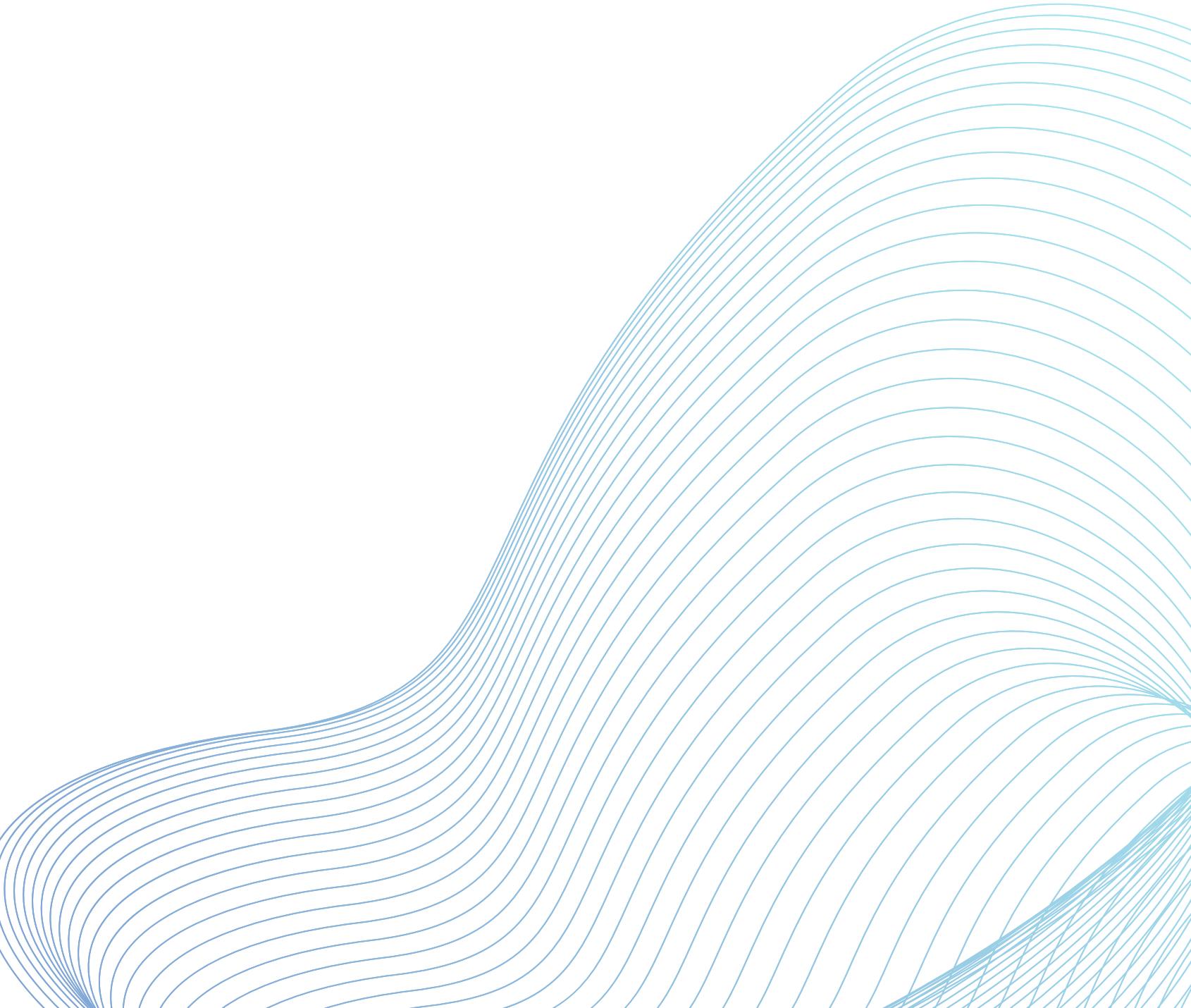




GOLANG ARCHITECTURE AND TDD

Manage AI Solution Co.,ltd





Keerati Rukmanee
Software Engineer
MANAGE AI SOLUTION CO.,LTD



EP 1.Function Programing & OOP

- Function programming คืออะไร ?
- ข้อดีและ ข้อเสียของ Function programming
- OOP คืออะไร ?
- ข้อดีและ ข้อเสียของ OOP

EP 2.Software design

- Software design คืออะไร ?
- Design Patterns คืออะไร ?

EP 3.Hexagonal Architecture

- Hexagonal Architecture คืออะไร ?
- PORT คืออะไร ?
- ADAPTER คืออะไร ?

EP 4.Test Driven Development (TDD) in Golang 1

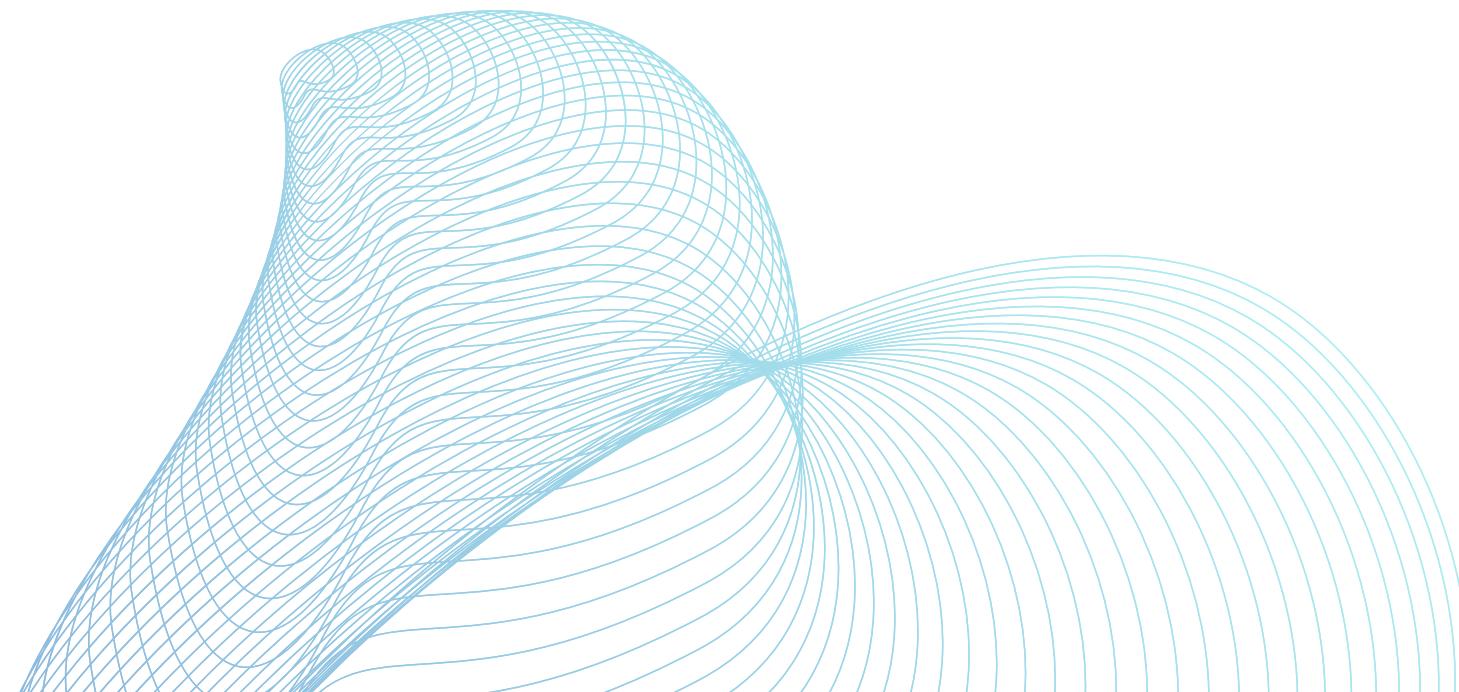
- Test Driven Development (TDD) คืออะไร
- โครงสร้างโปรเจคที่เหมาะสมสำหรับการพัฒนา TDD
- หลักการและแนวทางในการเขียนเทสแบบ Unit Test

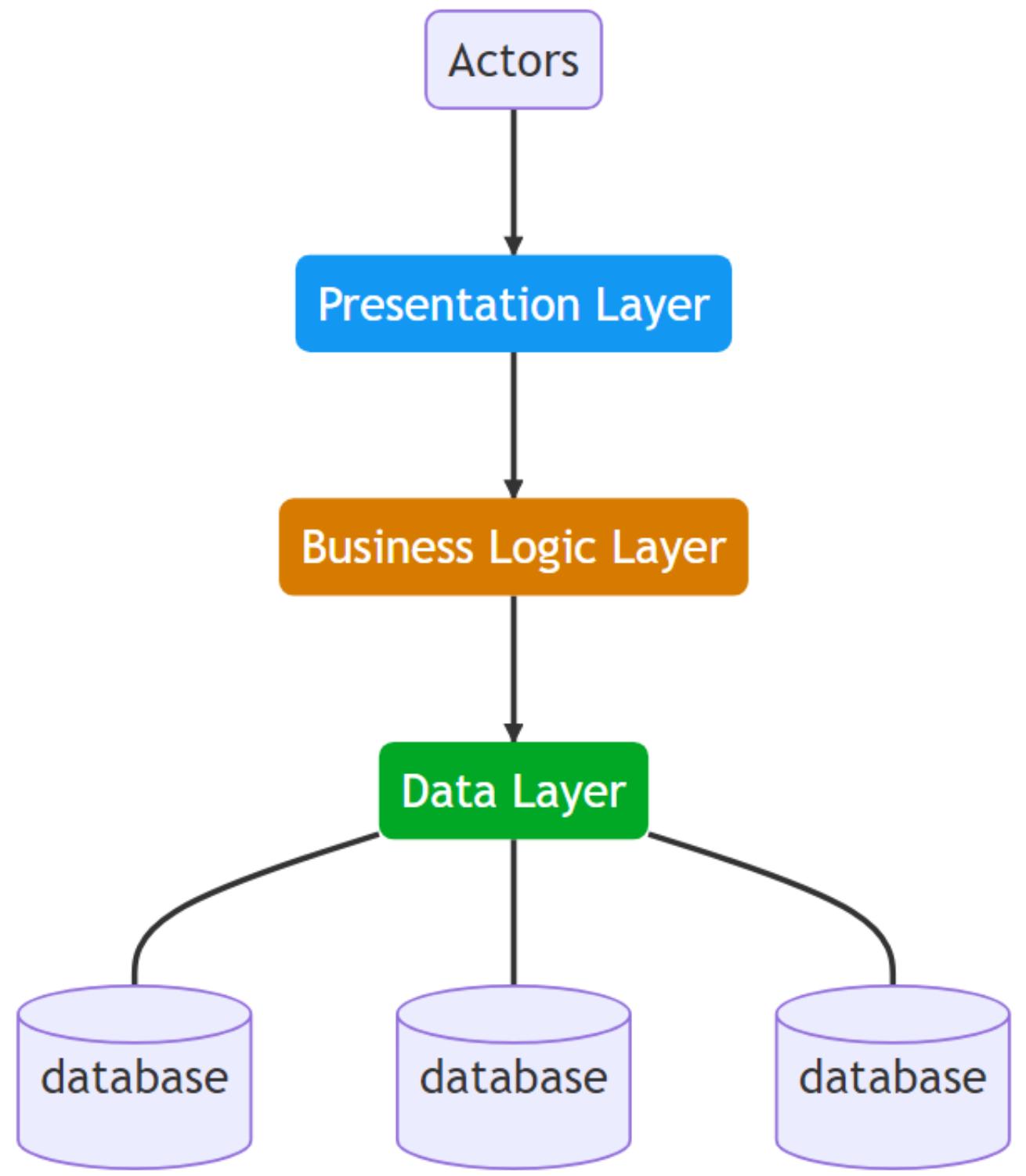
EP 5.Test Driven Development (TDD) in Golang 2

- การเทสแบบ Integration Test
- การเทสแบบ End-to-End Test
- การใช้ Test Doubles เพื่อทำ TDD
- แนวทางและการเขียนเทสตามหลัก TDD

EP 3.Hexagonal Architecture

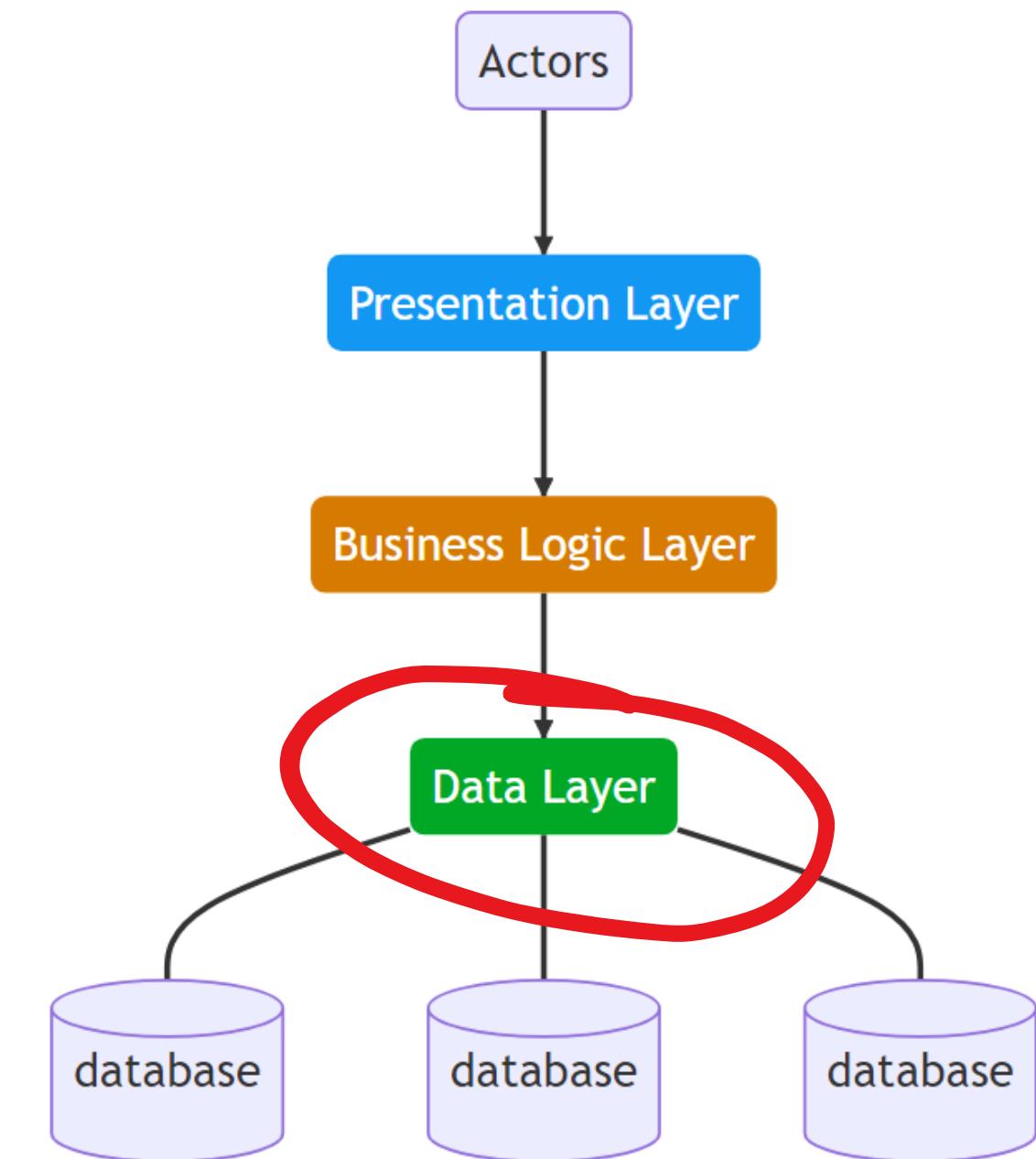
- Hexagonal Architecture คืออะไร ?
- PORT คืออะไร ?
- ADAPTER คืออะไร ?





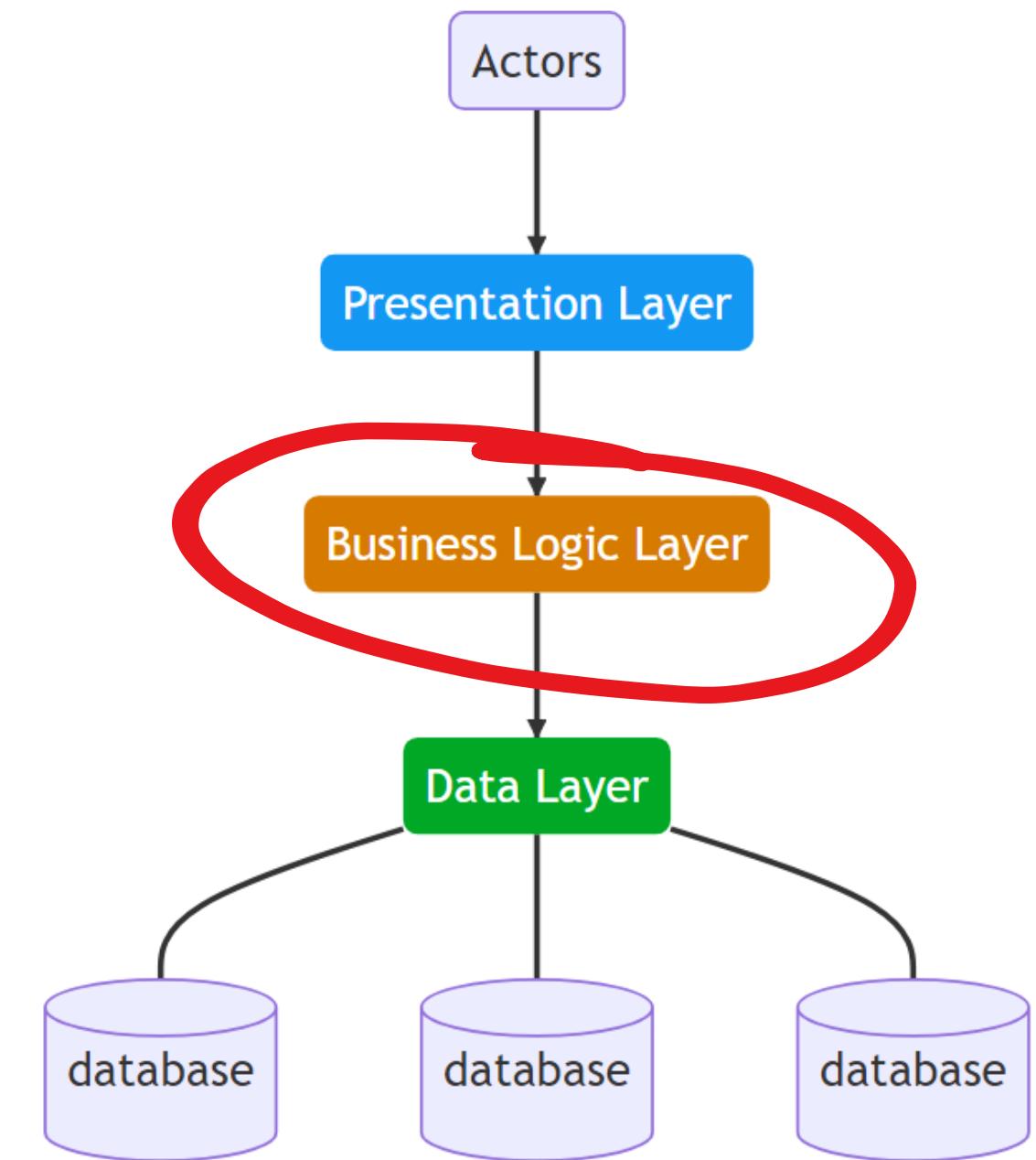
Data layer

เป็นส่วนที่ใช้ในการเก็บข้อมูล หรือเป็นส่วนที่ใช้ในการติดต่อกับ Techonogy ที่ใช้ในการเก็บข้อมูล หรือจะเรียกว่าส่วนที่ใช้ในการติดต่อ Database ก็ได้ เช่น หากว่าเราจะใช้ MongoDB หรือ Mysql เพื่อที่จะเก็บข้อมูลของ Application จะมีเพียง Data Layer เท่านั้นที่จะเชื่อมต่อ หรือรับถึงการมีอยู่ของ Database ที่กำลัง Connect อยู่ใน Application



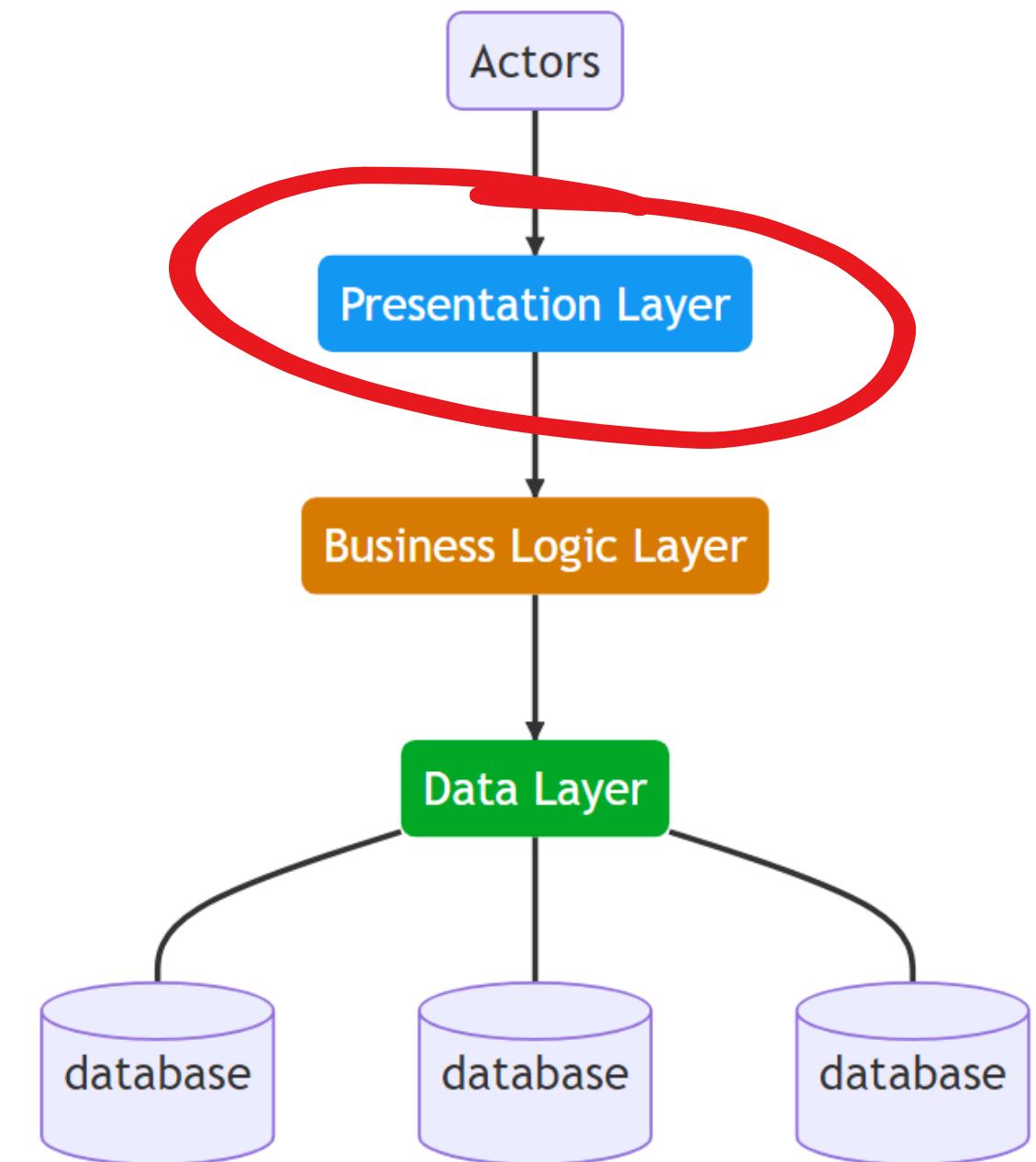
Business Logic Layer

เป็นส่วนหลักในการทำงานของระบบ โดยการทำงานในส่วนนี้จะยึดโยงกับ Requirement ของ Application เป็นส่วนที่ใช้ในการ Process งานต่างๆ เช่น เมื่อมีการเพิ่ม User เข้ามาในระบบ ส่วนของ Business Logic Layer จะต้องทำการ Process ว่า ข้อมูลของ User มีความถูกต้อง มีความครบถ้วน เกิดข้อผิดพลาด หรือเกิดปัญหาใดๆ อย่างเช่น มี User นั้นอยู่ในระบบอยู่แล้ว การ Handler Error ต่างๆ การทำ Logger จะอยู่ใน Layer นี้ ก็คงหมด

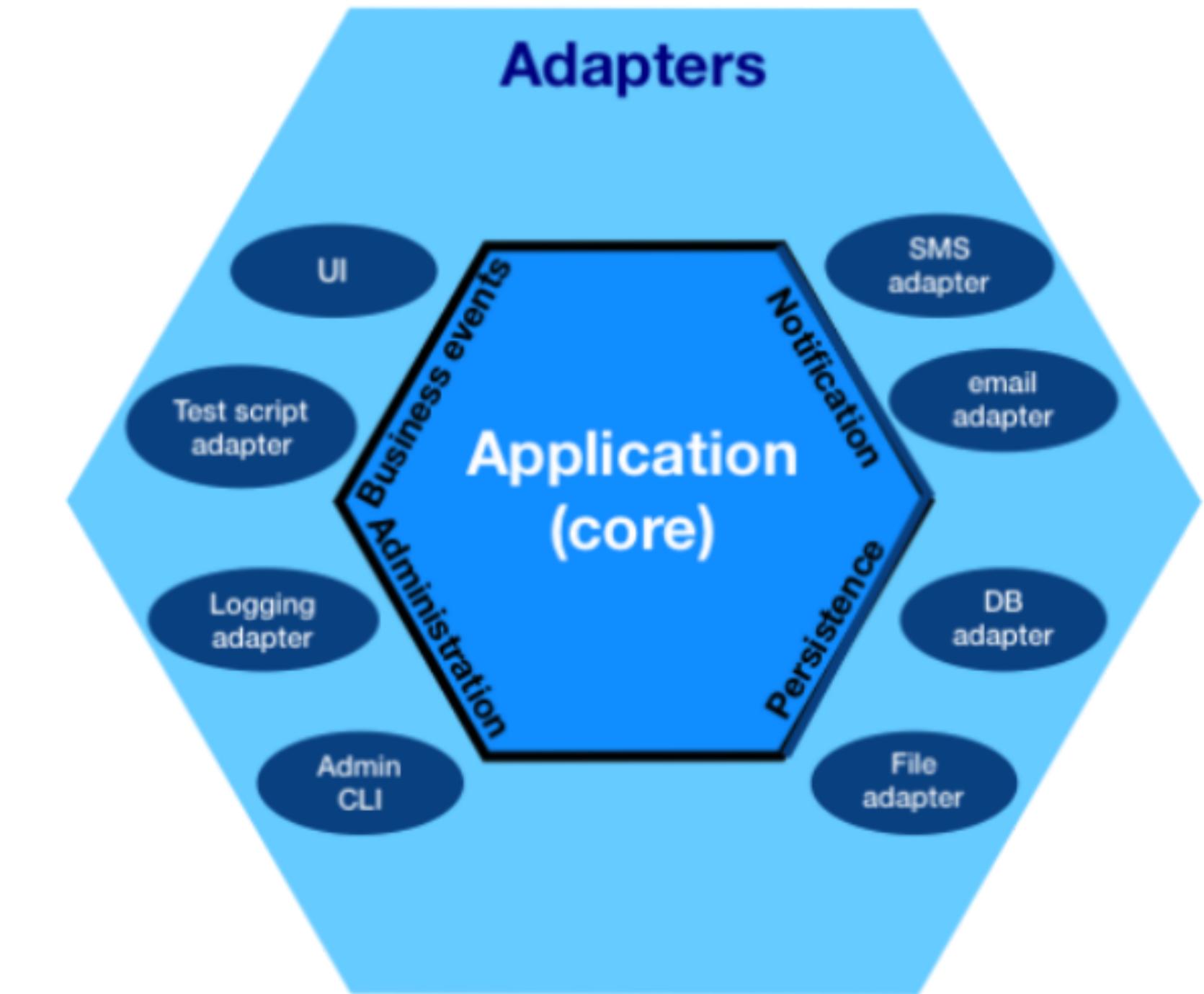


Presentation Layer

เป็นส่วนที่ใช้ในการติดต่อกับผู้ใช้งาน โดยการใช้งานนั้นมีหลายรูปแบบขึ้นอยู่กับ Technology เช่น RESTfull , Websocket, MessageQuery หรือ Cli โดยหน้าที่หลักๆ ของ layer นี้จะเป็นช่องทางที่ทำให้ User สามารถใช้งานระบบได้ เช่น หากต้องการ API ที่ใช้ในการเพิ่ม User ในระบบ ตัว Presentation Layer ก็จะใช้ libary หรือ Framework ที่สามารถ ทำ API Server ได้ เช่น net/http,fiber หรือ echo เพื่อสร้างช่องทางที่จะรับคำสั่งและข้อมูลจาก User

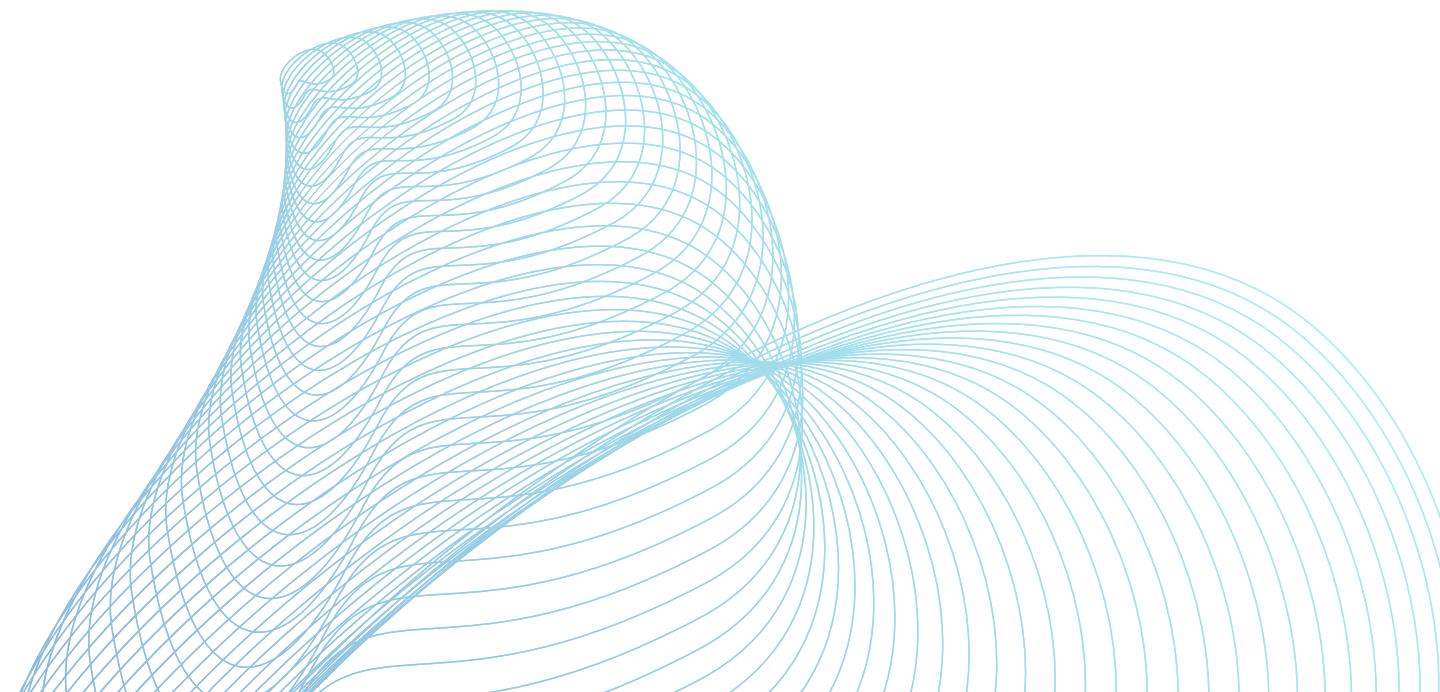


Hexagonal Architecture



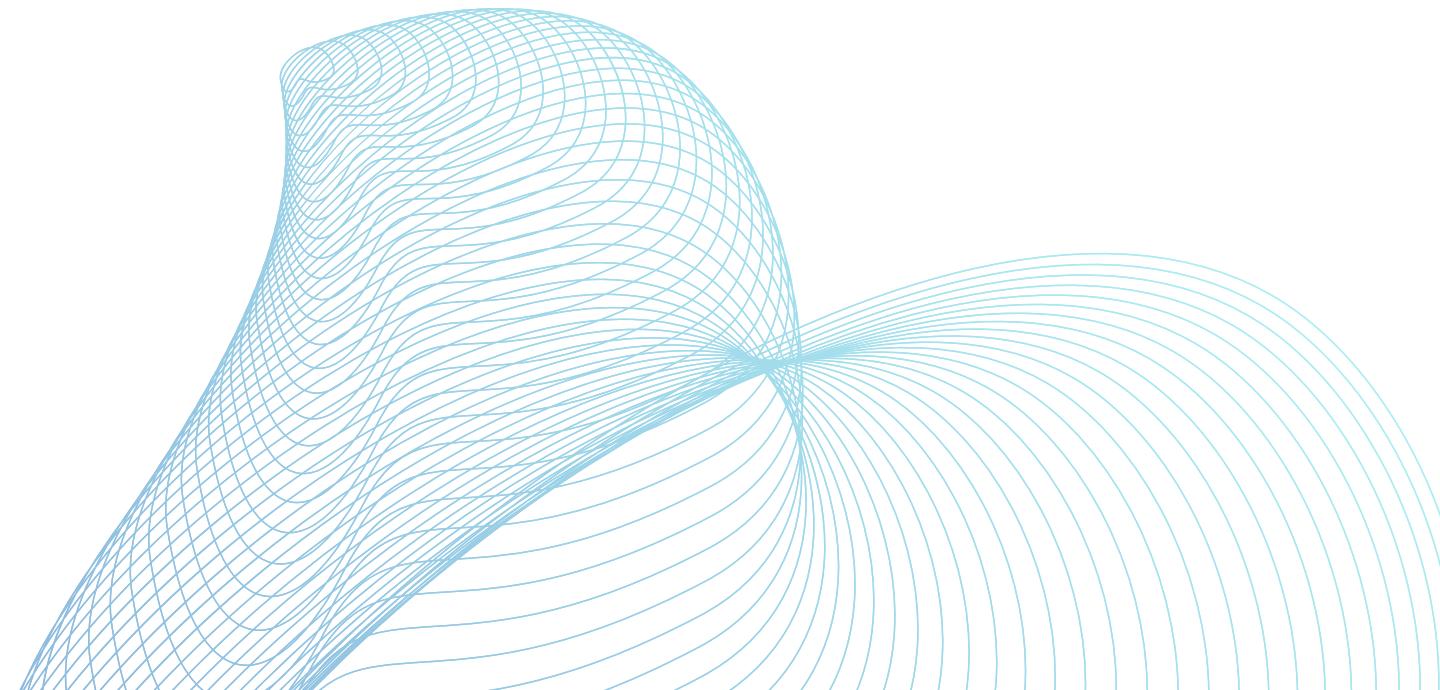
Hexagonal architecture

Hexagonal architecture หรือ Ports and Adapters architecture เป็นรูปแบบของการออกแบบระบบที่ใช้ในกระบวนการ Software Design มีจุดมุ่งหมายเพื่อสร้างส่วนประกอบของ Application โดยเชื่อมต่อกัน อย่างหลวມๆ(Loose coupling) เพื่อให้สามารถเปลี่ยนแปลงแก้ไข Environment ได้โดยง่าย โดยใช้ Concept Port & Adapter เพื่อกำหนดความสามารถเปลี่ยน Environment ได้ในทุกๆ Layer และ ง่ายต่อการ Test



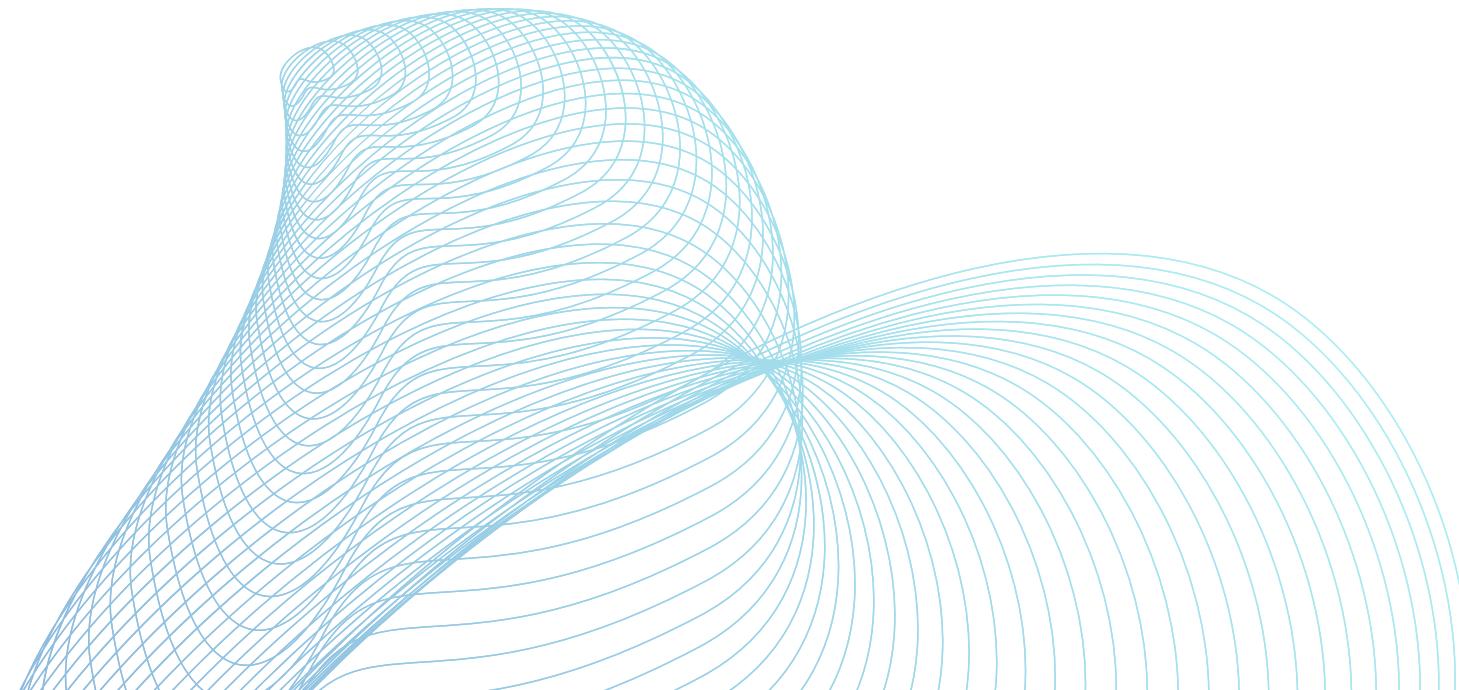
Hexagonal architecture

Hexagonal architecture ถูกคิดค้นโดย Alistair Cockburn และได้เริ่มเผยแพร่ ในปี 2005 เพื่อที่จะแก้ไขปัญหาด้าน Software Design เช่นการที่มีส่วนของ Business logic ไปอยู่ใน Data layer ซึ่งจะทำให้เกิดปัญหาว่าหากเปลี่ยน Data layer ใหม่ ตัว Business logic ก็อาจทำงานไม่สมบูรณ์ ไม่เหมือนเดิม เป็นต้น



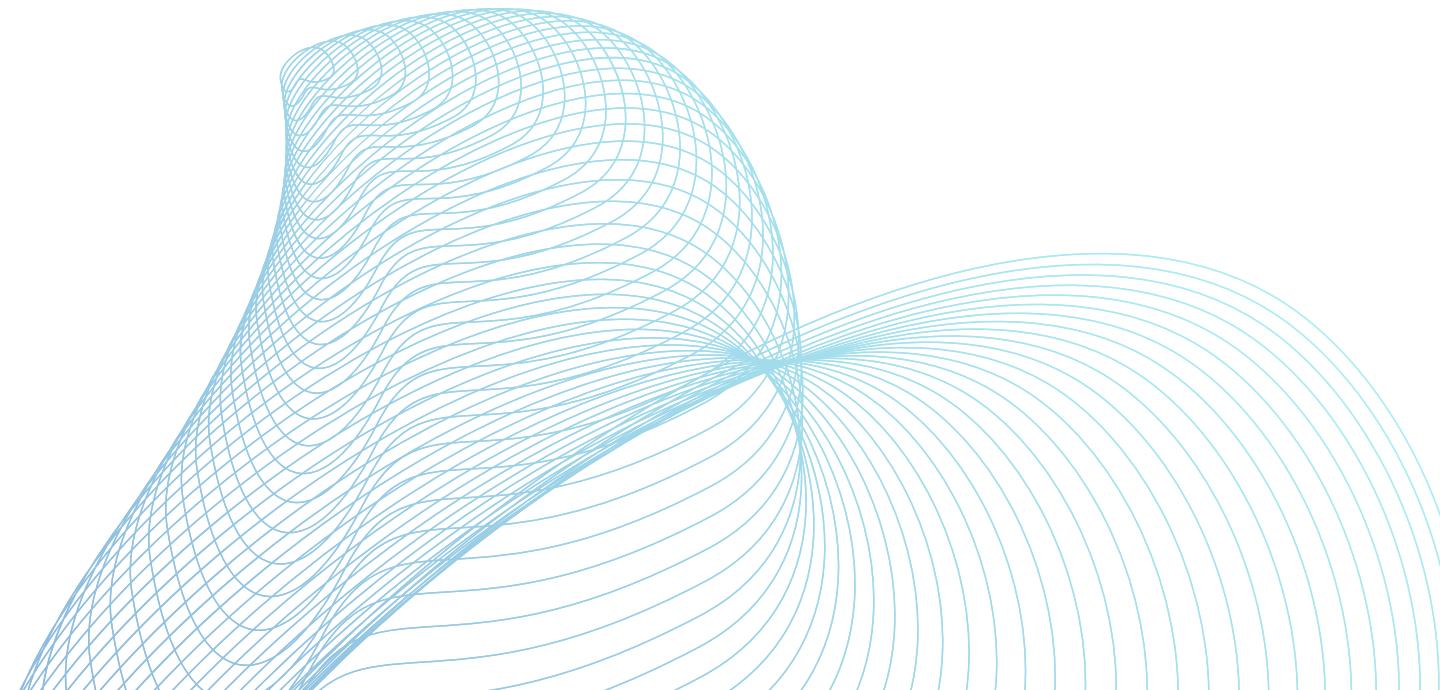
Hexagonal architecture

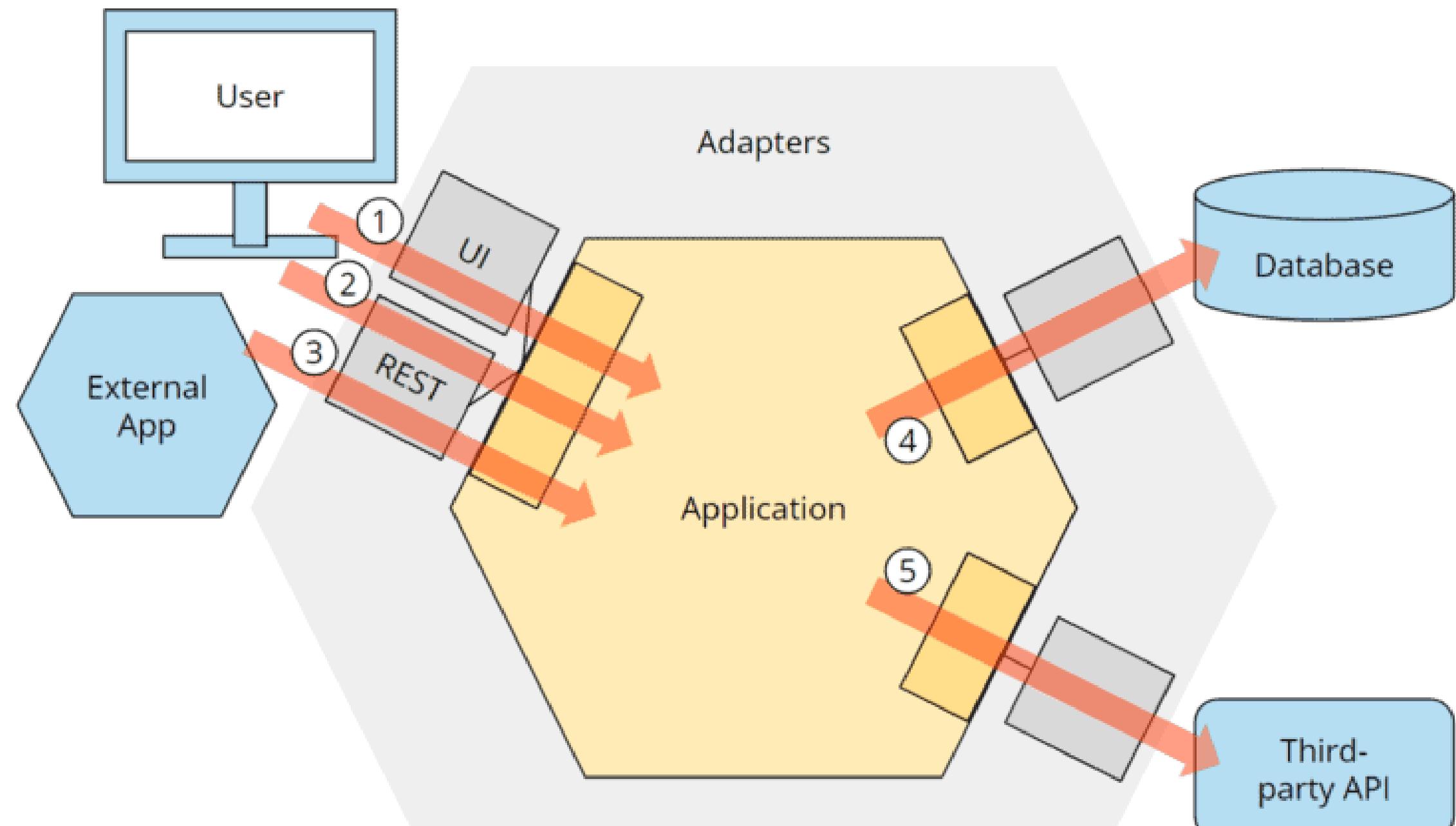
คำว่า "Hexagonal" มาจากภาพของ cell ที่มี 6 เหลี่ยม แต่ไม่ได้หมายถึงว่าใน Software นั้นจะมีแค่เพียง 6 ช่องทาง(port) ที่จะออกสู่ภายนอก แต่เพื่อที่ต้องการจะบอกว่ามีพื้นที่ สำหรับการแบ่ง Component และ Interface ต่างๆ ที่จำเป็น



Hexagonal architecture

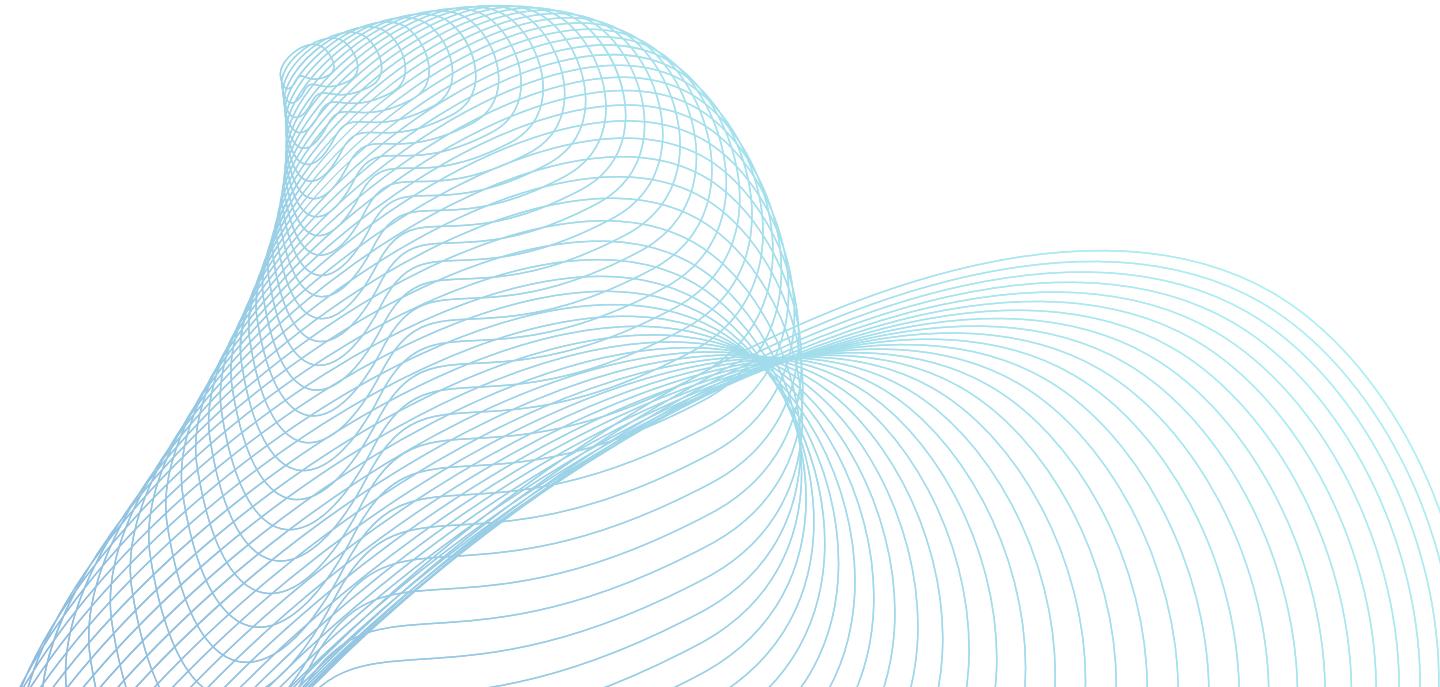
Hexagonal นั้นจะแบ่งส่วนต่างๆ ประกอบไปด้วย การทำงานหลักของระบบ(Core) ส่วนที่ใช้ในการติดต่อฐานข้อมูล(Database) ส่วนติดต่อผู้ใช้(user interface) ส่วนที่ใช้สำหรับการทดสอบ(Testing) และส่วนที่ใช้ในการติดต่อกับระบบอื่นๆ โดยในแต่ละส่วนนั้นจะเชื่อมต่อกันแบบหลวມๆ ซึ่งจะต่างจากการเขียนโปรแกรมแบบเดิม





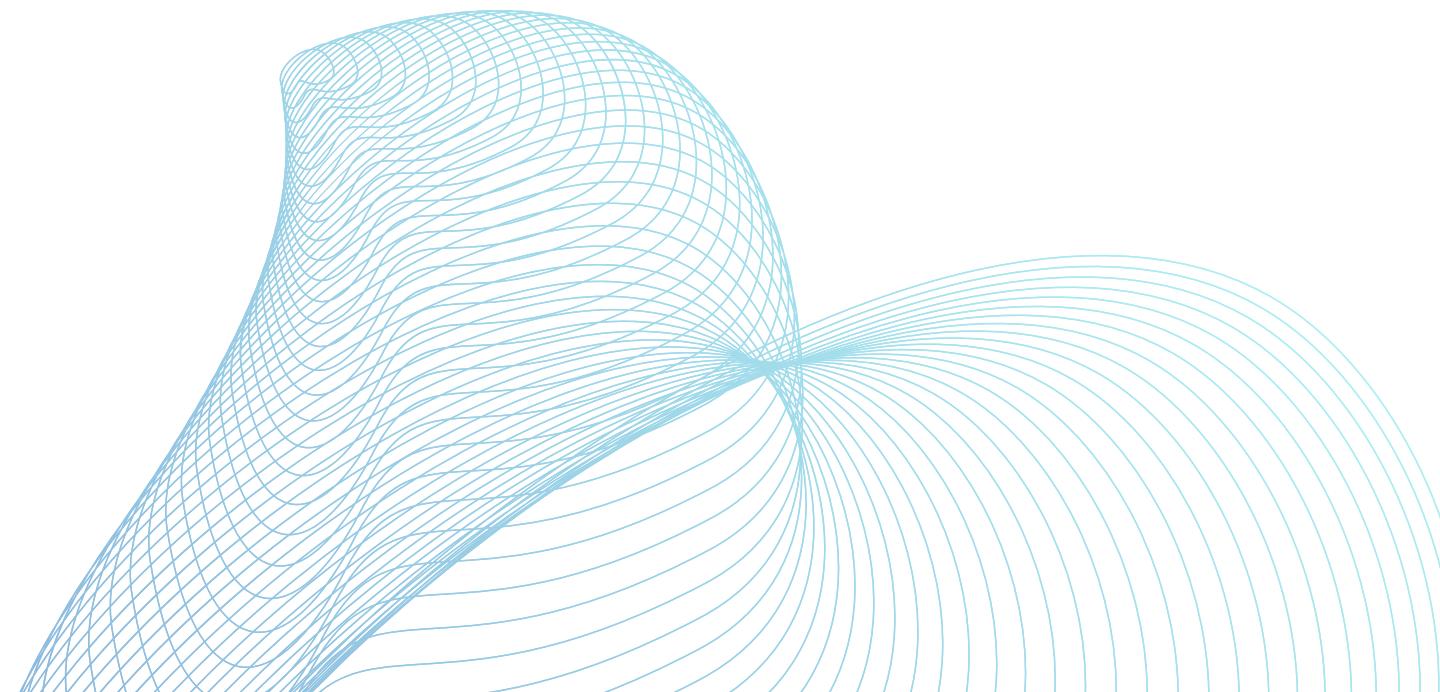
Port

แต่ละองค์ประกอบจะเชื่อมต่อกับส่วนอื่น ๆ ผ่าน สิ่งที่เรียกว่า "Port"
ซึ่งการสื่อสารผ่าน Port จะเป็นไปตาม Protocols ที่กำหนด ขึ้นอยู่กับวัตถุประสงค์ และหน้าที่ของ
Port นั้นๆ ซึ่ง Port นั้นจะเพียงแค่เป็นตัวที่ใช้ในการกำหนดรูปแบบของ Protocols ที่จะใช้ในการสื่อสาร



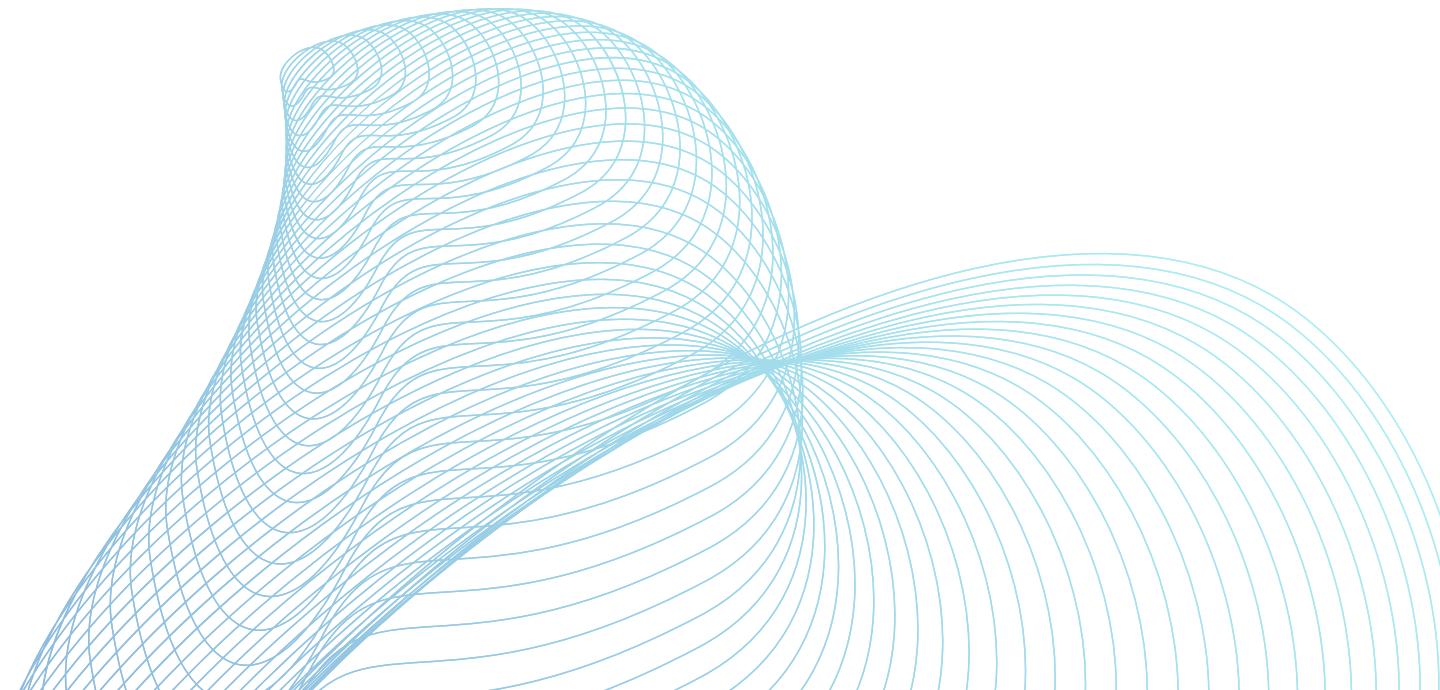
Port

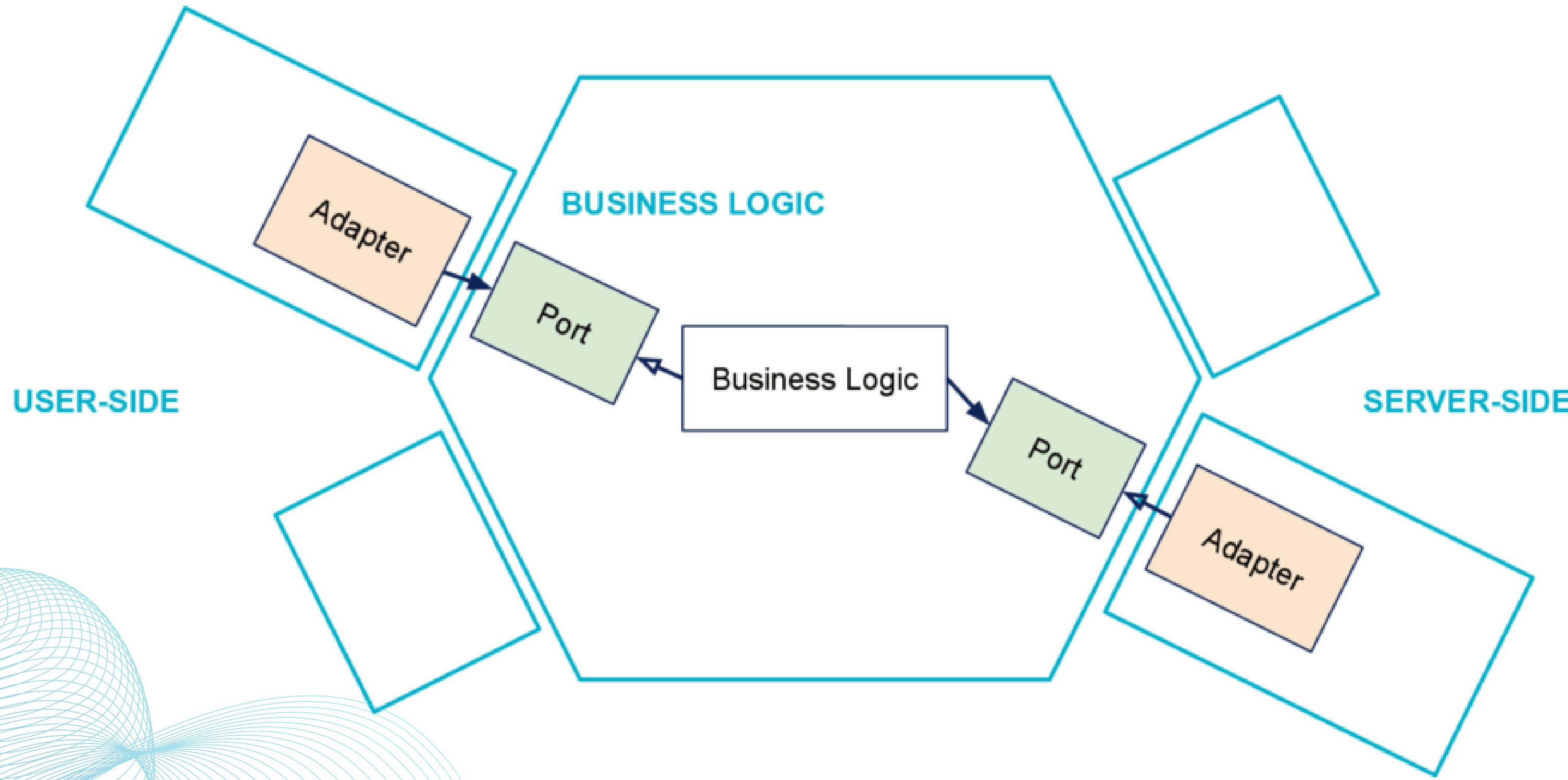
โดยที่ความลະເວີດແລະຈຳນວນຂອງ Port ໄມໄດ້ຄູກຈຳກັດໄວ້ ໃນບາງຄຣນີ ກາຣທີ່ເຮັນນັ້ນມີເພື່ອງ Port ເດີຍວົງເພື່ອ ແຕ່ໂດຍກ່ຽວປະເທດ Port ຕາມ Action ທີ່ Port ນັ້ນຈະກຳ ເຊັ່ນ ກາຣເຊື່ອມຕ່ວ MongoDB ແລະ Redis ເຮົາຈະສັງເກດ ເພື່ອ Port ເດີຍວແລ້ວ Connect ກັ້ງ 2 ຕັ້ງເລຍກີໄດ້ ແຕ່ຫາກເຮົາກຳໃໝ່ມັນລະເວີດຂຶ້ນ ແລະແຍກ Action ອອກຈາກກັນເຮົາຈະແຍກໄດ້ເປັນ 2 Port ຄື່ວ່າ Port ສຳຮັບ Connect MongodB ແລະ Port ສຳຮັບ Connect Redis ເປັນຕົ້ນ

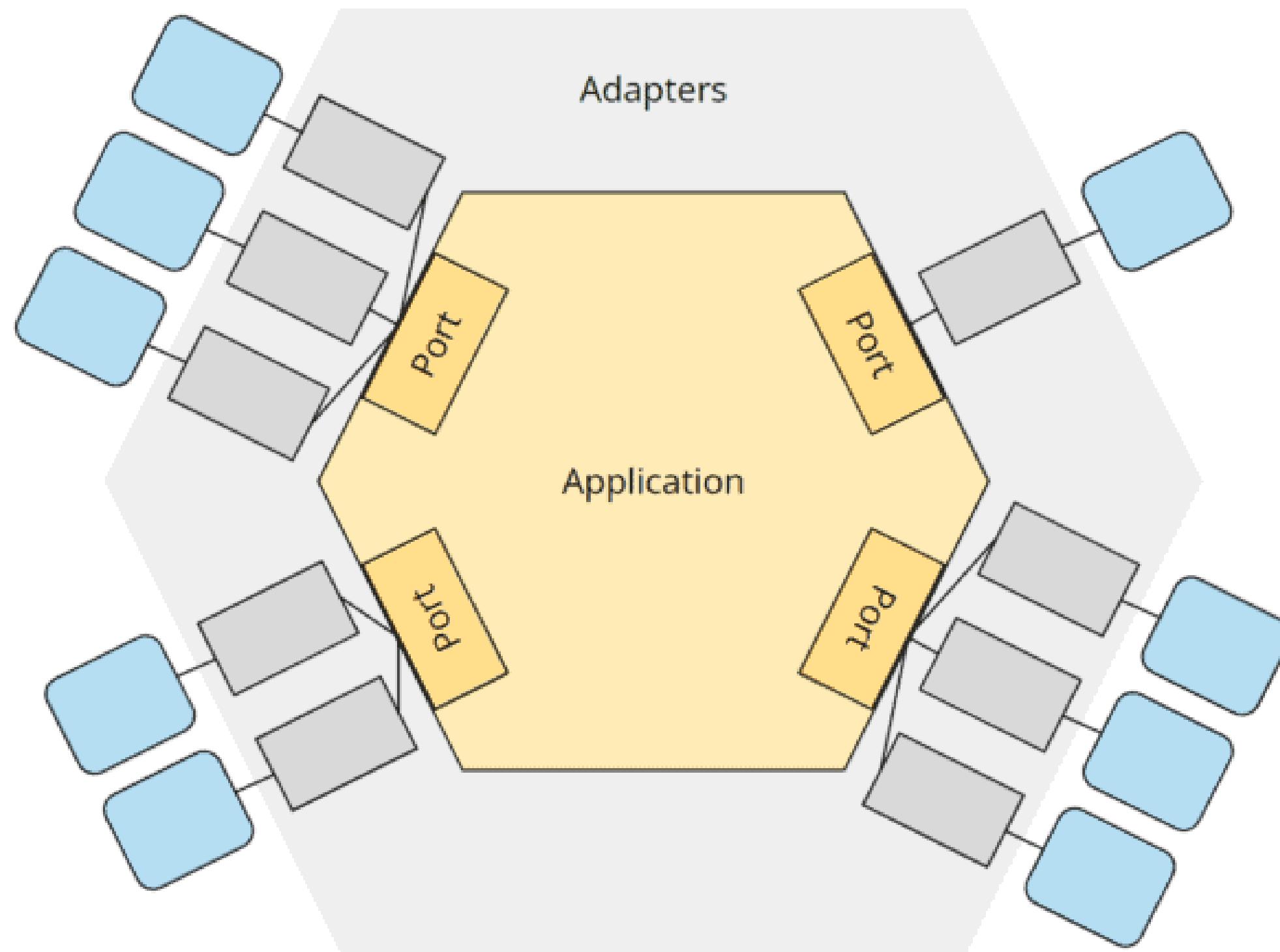
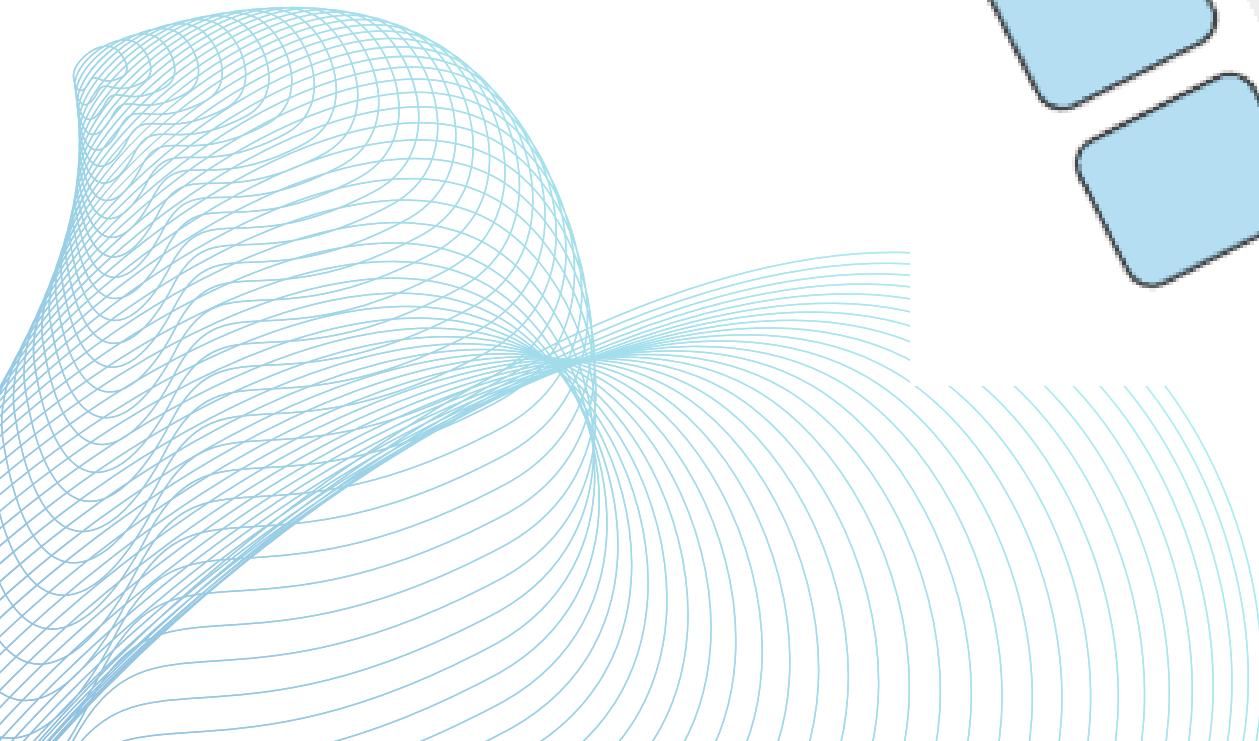


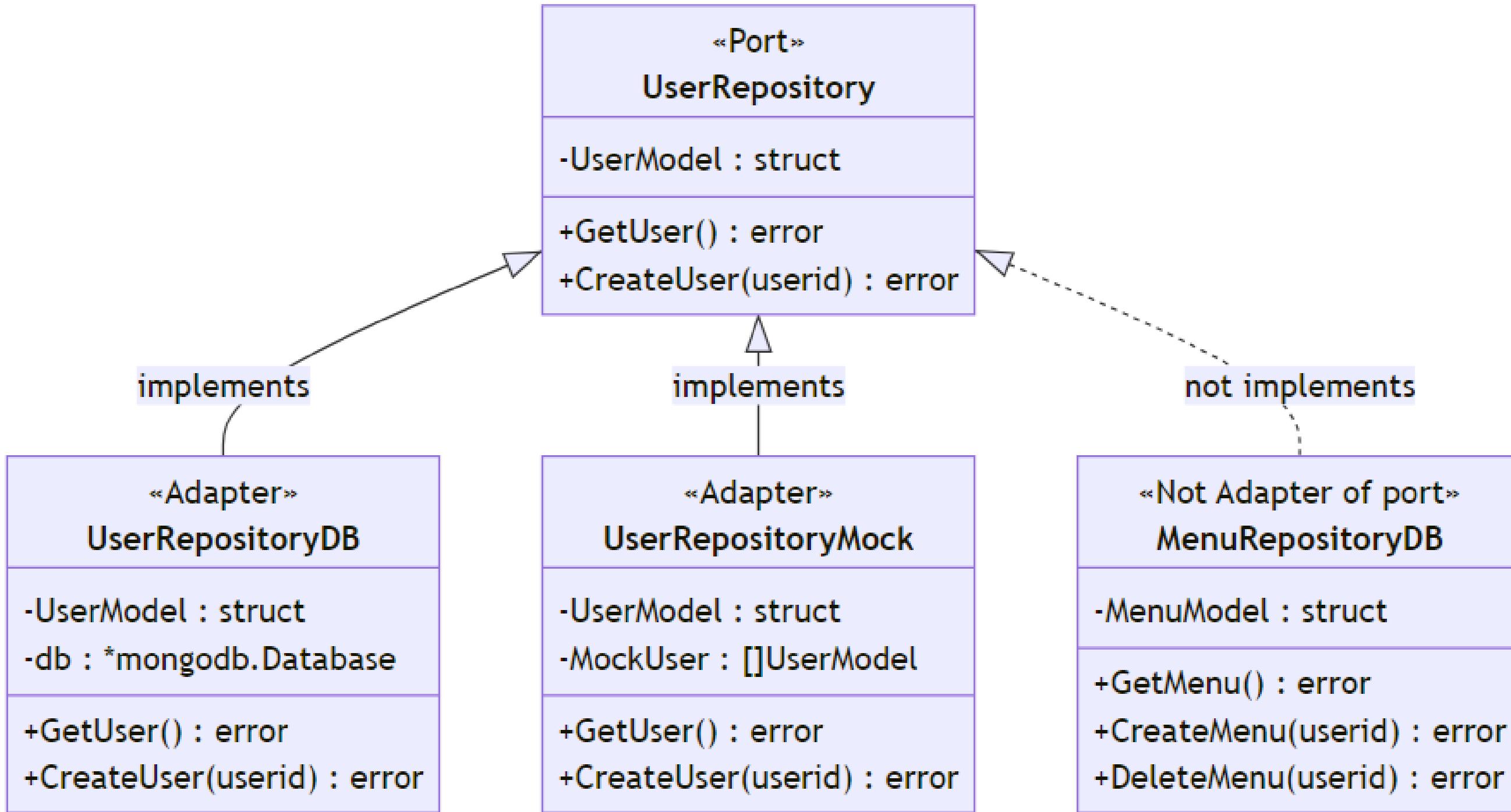
Adapter

ส่วนที่เรียกว่า Adapter นั้นเป็นตัวที่เข้ามาเชื่อมต่อระหว่างระบบของเรา กับส่วนต่างๆภายนอก เพื่อให้เกิดการทำงาน ตามที่ระบบได้ถูกออกแบบไว้ ซึ่งการสร้าง Adapter นั้นจะถูกกำหนดด้วย Port ว่าต้องใช้งาน Protocols แบบไหนจึงสามารถเชื่อมต่อกับระบบของเราได้ Port เดียวอาจจะมี Adapter ได้หลายตัว เช่น Adapter ที่เชื่อมต่อกับ Database และ Adapter ที่เป็น Mock Data เพื่อใช้ในการทดสอบ เวลาที่เราต้องการเทสในส่วนต่างๆ ของ Application จะได้ไม่กระทบกับข้อมูลจริงในระบบ



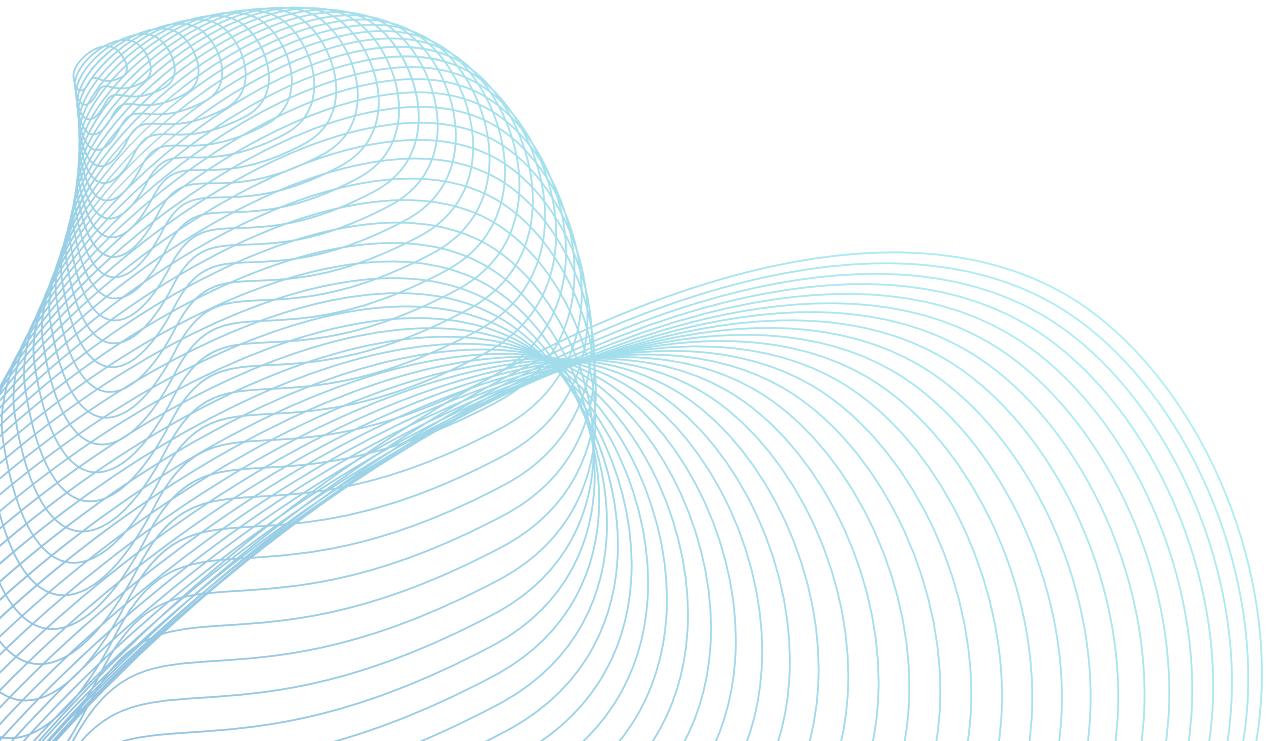






จาก Diagram จะเห็นว่า UserRepositoryDB และ UserRepositoryMock นั้น Conform ตาม UserRepository เพราะฉะนั้นเราสามารถเชื่อมต่อตัว Data Source ได้ทันที แต่ MenuRepositoryDB ไม่ได้ Conform ตาม UserRepository จึงไม่สามารถเชื่อมต่อ Adapter ได้

Coding!!



THANK YOU

Credit pic :

<https://github.com/ukritfongsomboon/Hexagonal>

https://en.wikipedia.org/wiki/Hexagonal_architecture_%28software%29

<https://www.happycoders.eu/software-craftsmanship/hexagonal-architecture/>

<https://blog.octo.com/hexagonal-architecture-three-principles-and-an-implementation-example/>

