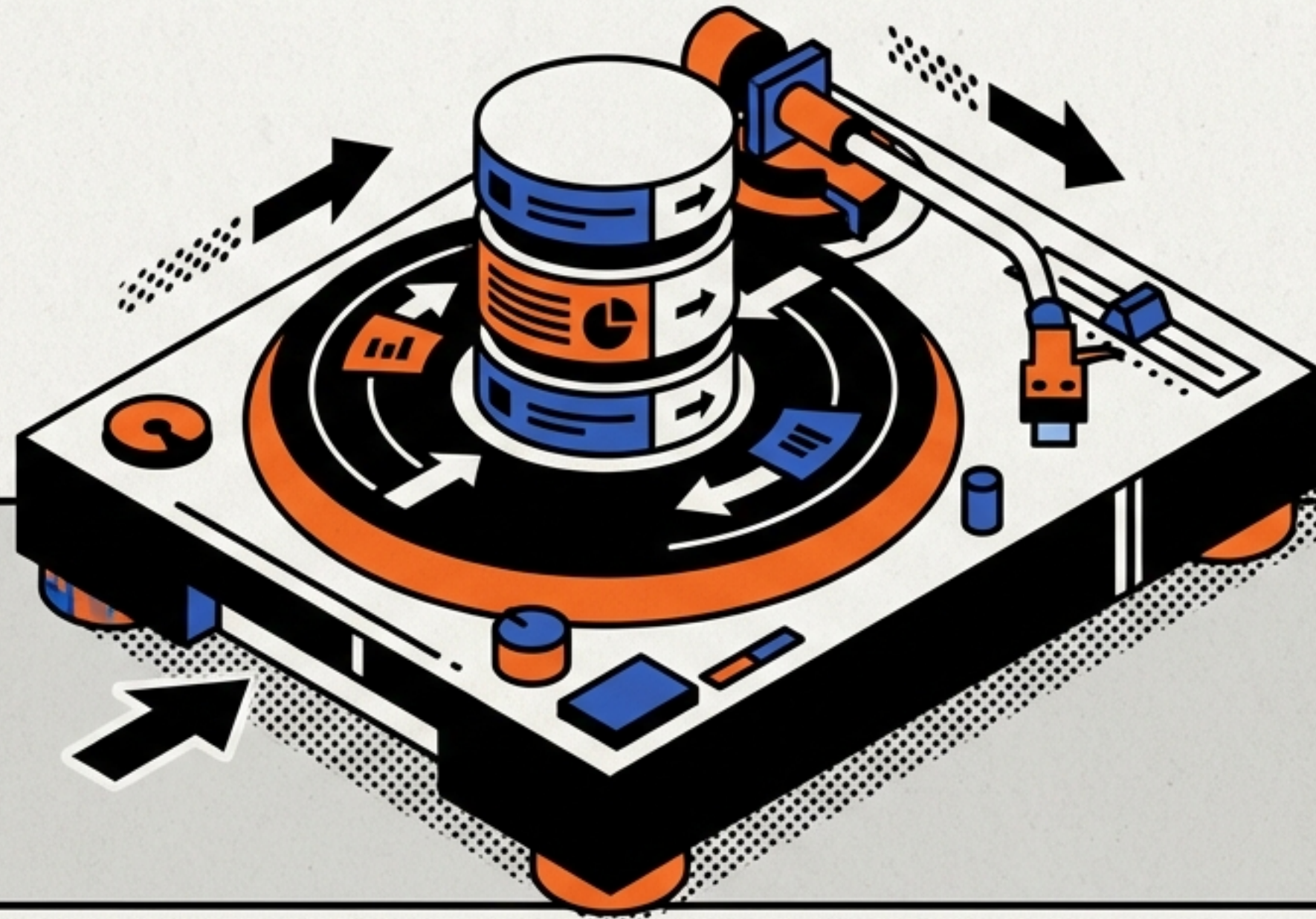


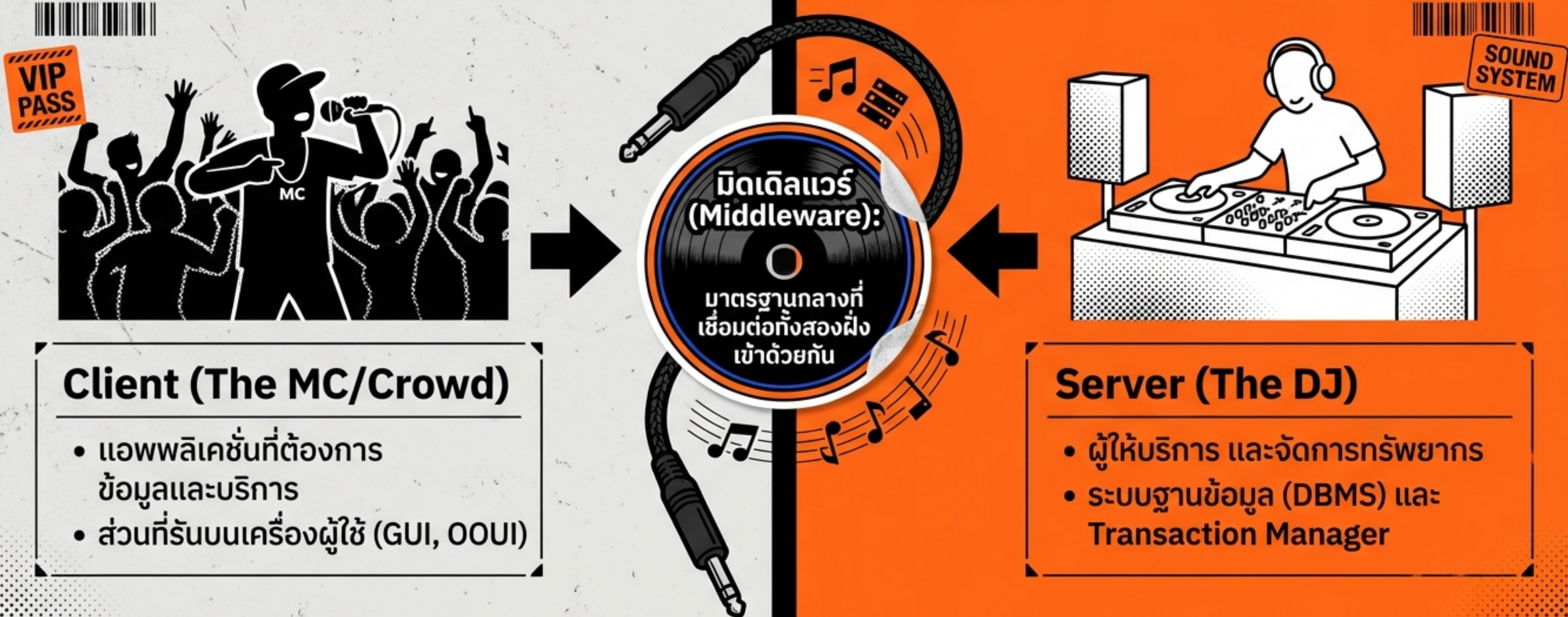
DATA CYPHER

สถาปัตยกรรมและการจัดการฐานข้อมูล (DBMS Architecture & Data Management)



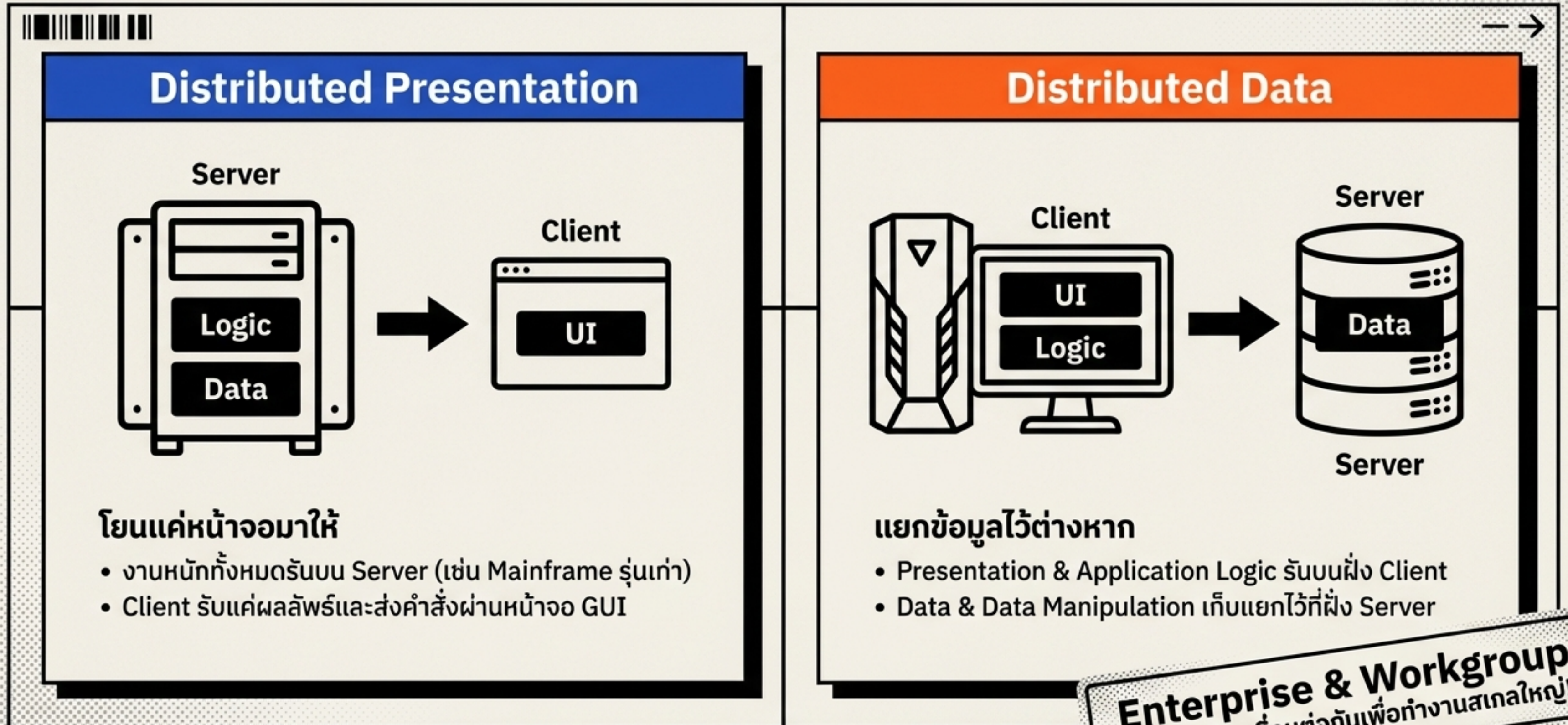
Modern Management and Information Technology | Client/Server • 3-Schema • Data Models

THE SETUP: Client / Server คืออะไร?



การแยกหน้าที่กันชัดเจนช่วยให้ใช้ทรัพยากร (Resources) ได้อย่างมีประสิทธิภาพสูงสุด!

THE ARCHITECTURE: รูปแบบการกระจายระบบ



โยนแค่นั้นมาให้

- งานหนักทั้งหมดรันบน Server (เช่น Mainframe รุ่นเก่า)
- Client รับแค่นั้นและส่งคำสั่งผ่านหน้าจอ GUI

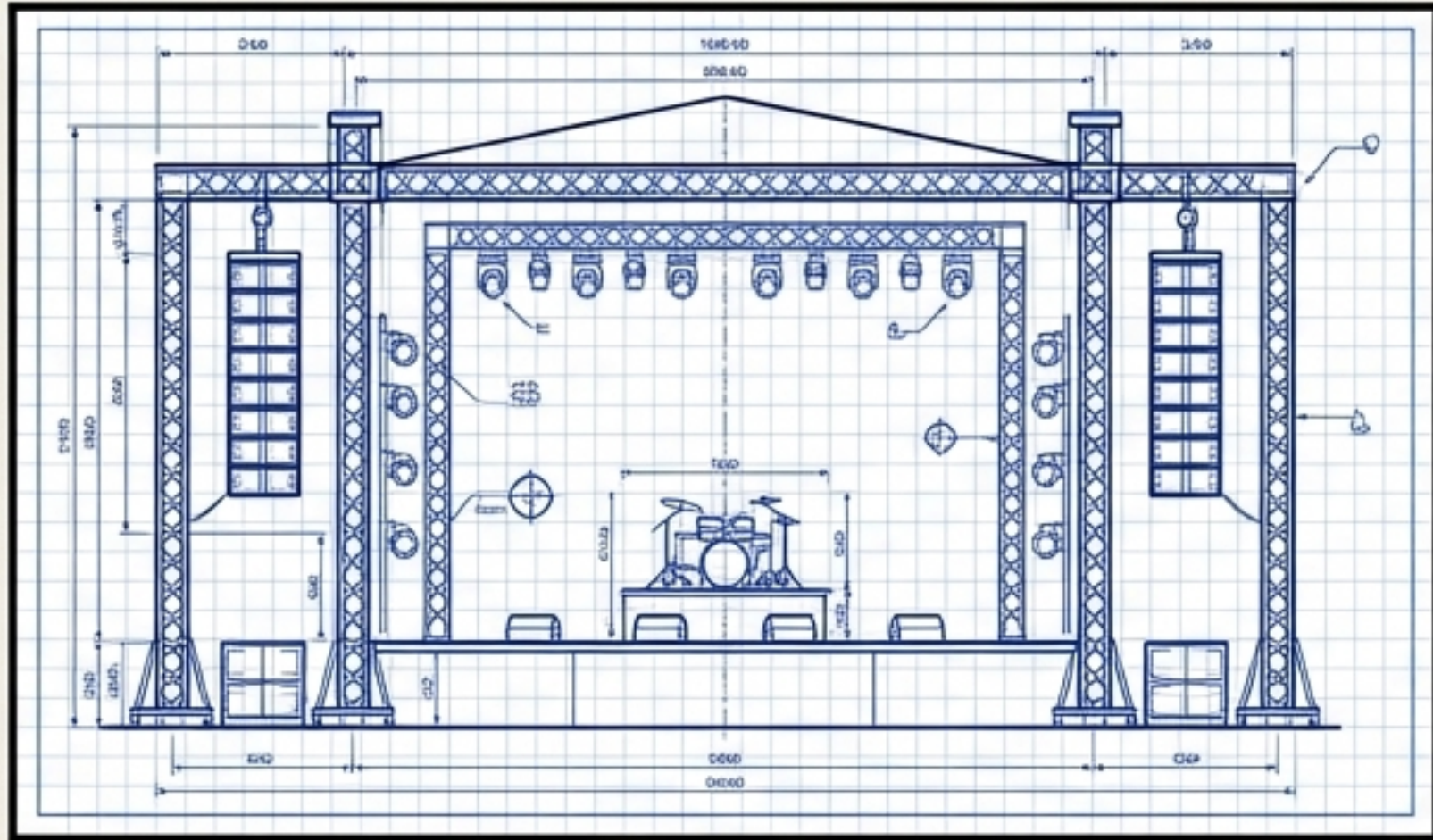
แยกข้อมูลไว้ต่างหาก

- Presentation & Application Logic รันบนฝั่ง Client
- Data & Data Manipulation เก็บแยกไว้ที่ฝั่ง Server

Enterprise & Workgroup
ระดับองค์กรเชื่อมต่อกันเพื่อทำงานสเกลใหญ่!

THE BLUEPRINT vs. THE LIVE SHOW

โครงสร้างข้อมูล (Schema) vs. เนื้อข้อมูล (Instance)



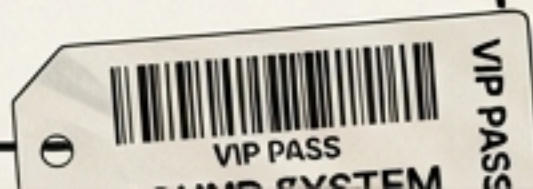
SCHEMA (โครงสร้าง)

- การออกแบบและนิยามฐานข้อมูล (เช่น กำหนดว่าต้องมีรหัสนักศึกษา, ชื่อ, สาขา)
- Vibe Check: เปลี่ยนแปลงยาก นานๆ เปลี่ยนที (เหมือนการสร้างโครงสร้างตึก หรือวาดแปลนเวที)



INSTANCE (เนื้อข้อมูล)

- ข้อมูลจริงๆ ที่เก็บอยู่ในระบบ ณ วินาทีนั้น (เช่น เรคคอร์ด 54370001, แดง ใจดี)
- Vibe Check: เปลี่ยนแปลงตลอดเวลา มีการ Insert/Delete ทุกนาที (เหมือนคนดูที่เดินเข้า-ออกฮอลล์คอนเสิร์ต)

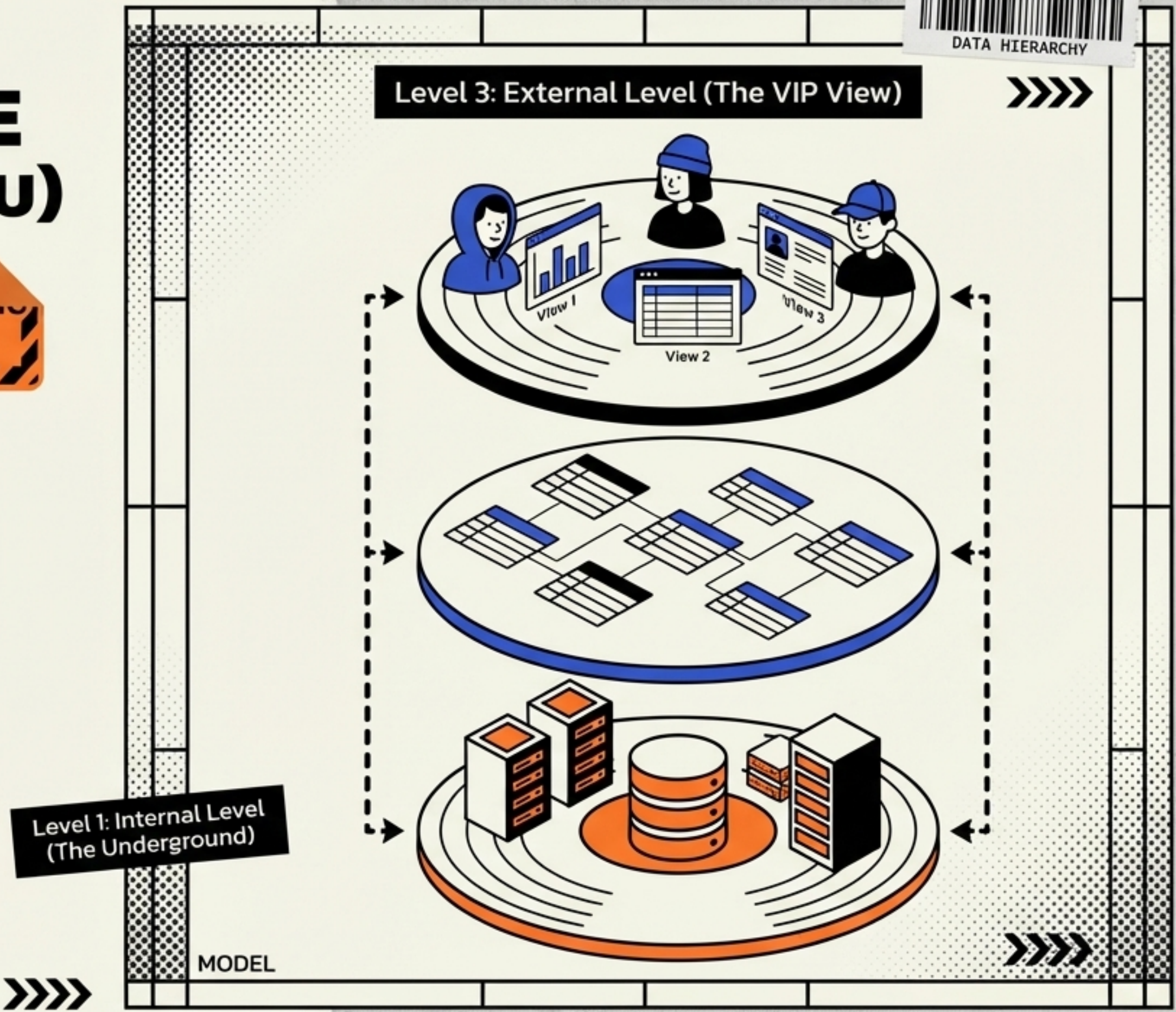


3-LEVEL ARCHITECTURE (สถาปัตยกรรม 3 ระดับ)

มาตรฐาน ANSI/SPARC - แยก Application ของผู้ใช้ออกจากฐานข้อมูลทางกายภาพ



เป้าหมายหลัก: ข้อมูลชุดเดียว แต่มองเห็นได้หลายมุมมอง โดยไม่ต้องยุ่งกับระบบหลังบ้าน!



LEVEL 1: INTERNAL SCHEMA

Physical Layer - ข้อมูลถูกเก็บในฮาร์ดดิสก์จริงๆ อย่างไร?

The Job:

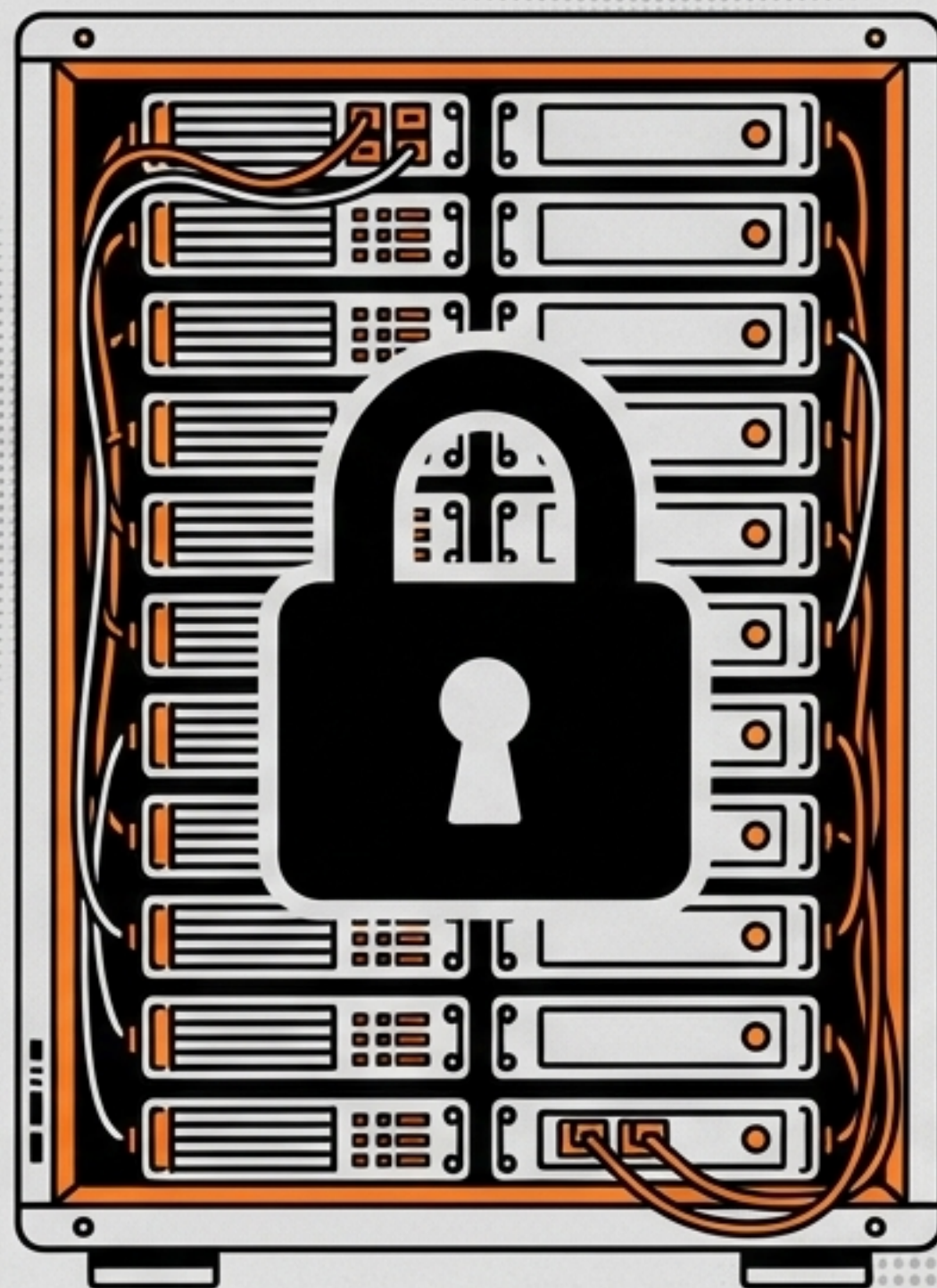
จัดการระบบการเก็บข้อมูลจริง
(Physical Storage Structure)
ระดับพีลีสัก

The Mechanics:

กำหนดโครงสร้างข้อมูล (Data
Structure), การจัดระเบียบแฟ้มข้อมูล
(File Organization), และ Indexing
เพื่อให้เข้าถึงข้อมูลไวที่สุด


The Players:

ทำงานร่วมกับ Operating System (OS) เพื่อเขียนข้อมูลลงสื่อบันทึก
(Secondary Storage)



RESTRICTED AREA: ซ่อนความซับซ้อนไว้ ผู้ใช้ทั่วไปไม่มีสิทธิ์เข้ามายุ่ง (พื้นที่ของ DBA / ผู้ดูแลระบบเท่านั้น)

LEVEL 2: CONCEPTUAL SCHEMA

Logical Layer - ภาพรวมของข้อมูลทั้งระบบสัมพันธ์กันอย่างไร? 

The Job:

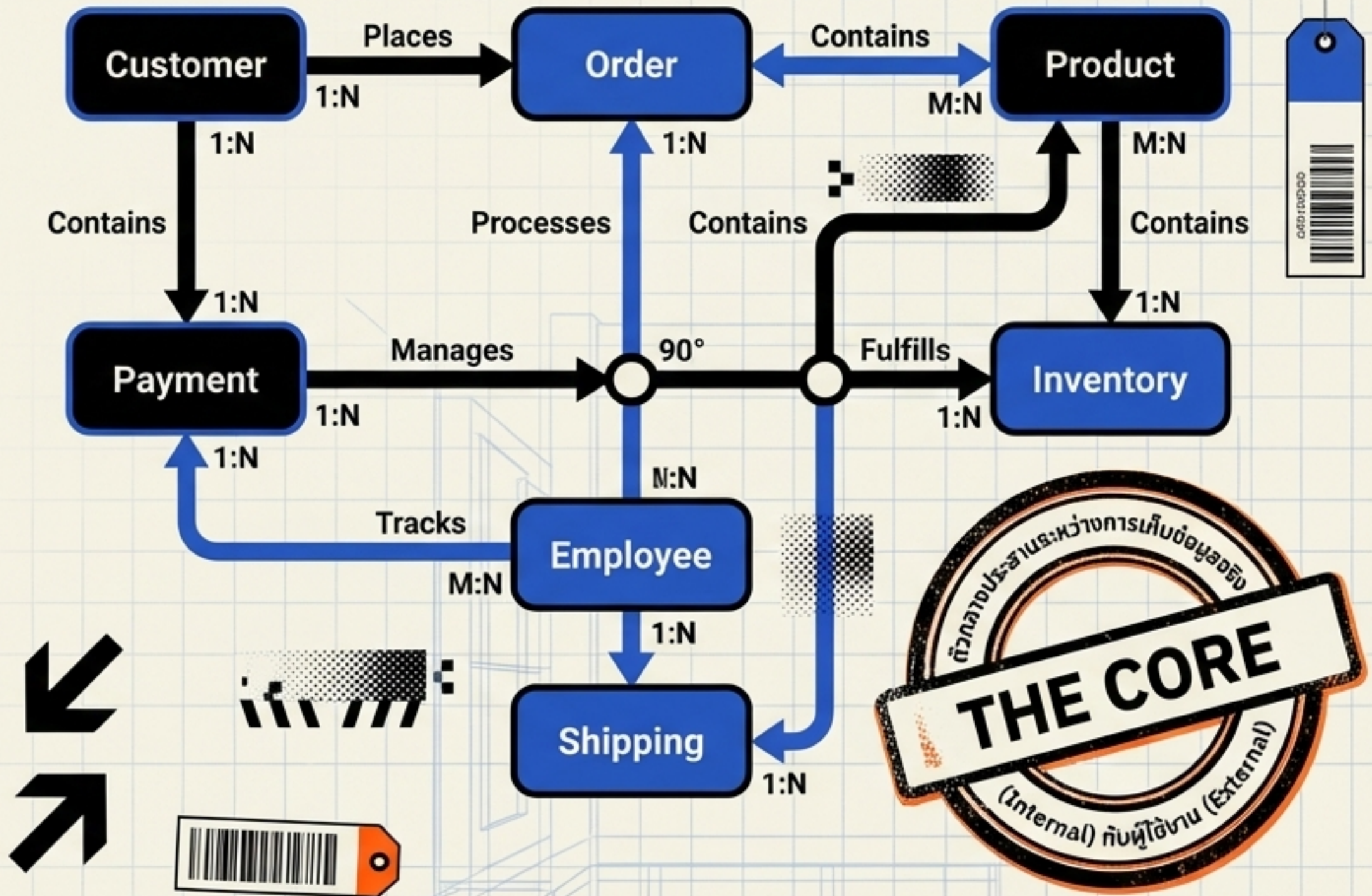
อธิบาย ตรรกะ ของฐานข้อมูลทั้งหมด (Logical Schema) โดยซ่อนรายละเอียดการเก็บข้อมูลระดับ Physical (ฮาร์ดดิสก์) ไว้ทั้งหมด

What's Inside:

อธิบายเอนทิตี (Entities), ชนิดข้อมูล (Data Types), ความสัมพันธ์ (Relationships), และกฎความปลอดภัย (Security Constraints)

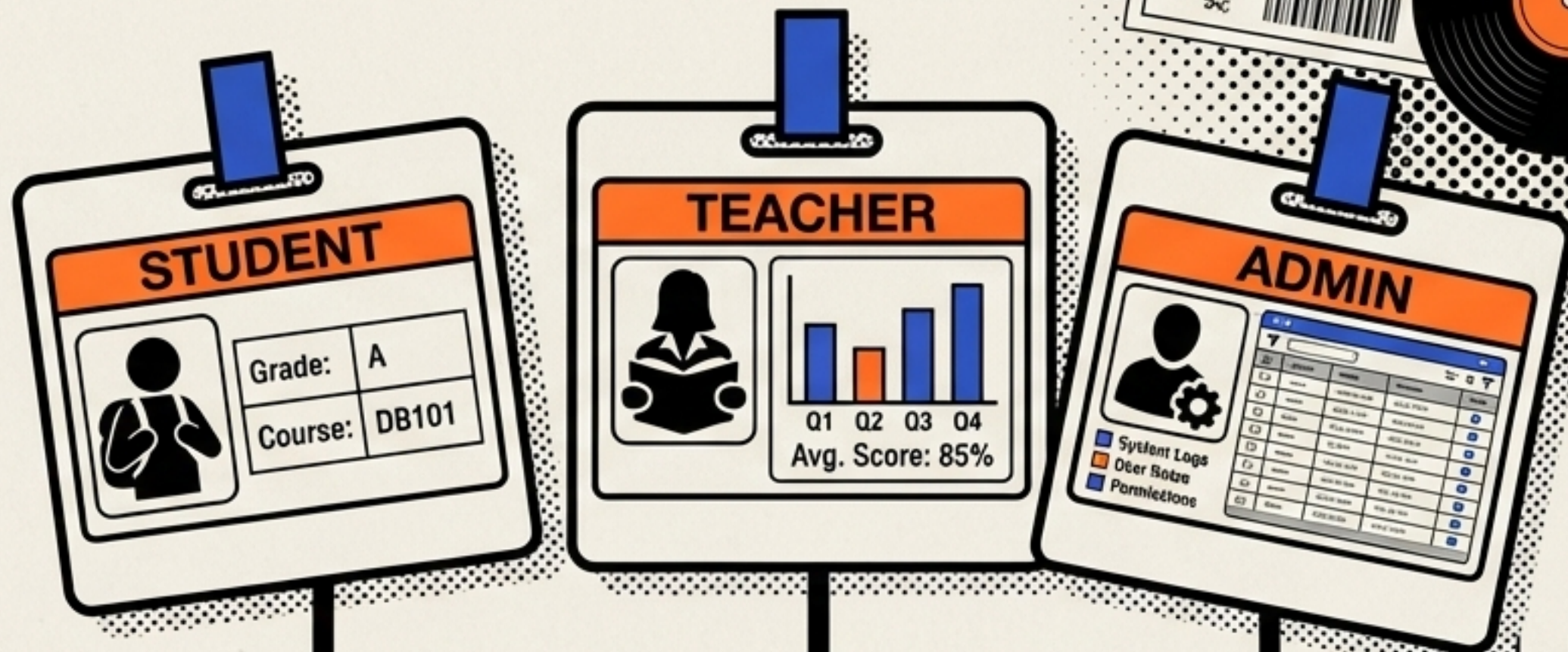
The Players:

พื้นที่ทำงานหลักของ Programmers และ Database Administrators (DBA)



LEVEL 3: EXTERNAL SCHEMA

View Layer - สิ่งที่ใช้ใช้งานแต่ละคน
(End Users) มองเห็น



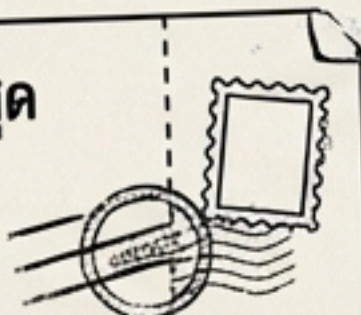
The Job	The Feature	The Magic
สร้าง วิว (View) หรือหน้าต่างข้อมูลย่อยๆ ที่ผู้ใช้แต่ละกลุ่มมีสิทธิ์เข้าถึงหรือสนใจ	ผู้ใช้ต่างกลุ่มกัน มองเห็นข้อมูลจากฐานข้อมูลเดียวกันในรูปแบบที่ต่างกัน (เช่น ฝ่ายบัญชีเห็นเงินเดือน, ฝ่าย HR เห็นแค่วันลา)	ข้อมูลบางอย่างไม่ได้ถูกเก็บจริงในดิสก์ แต่เกิดจากการคำนวณโชว์แว้ขึ้นมาบนหน้าจอแบบเรียลไทม์!



MASTER DATABASE



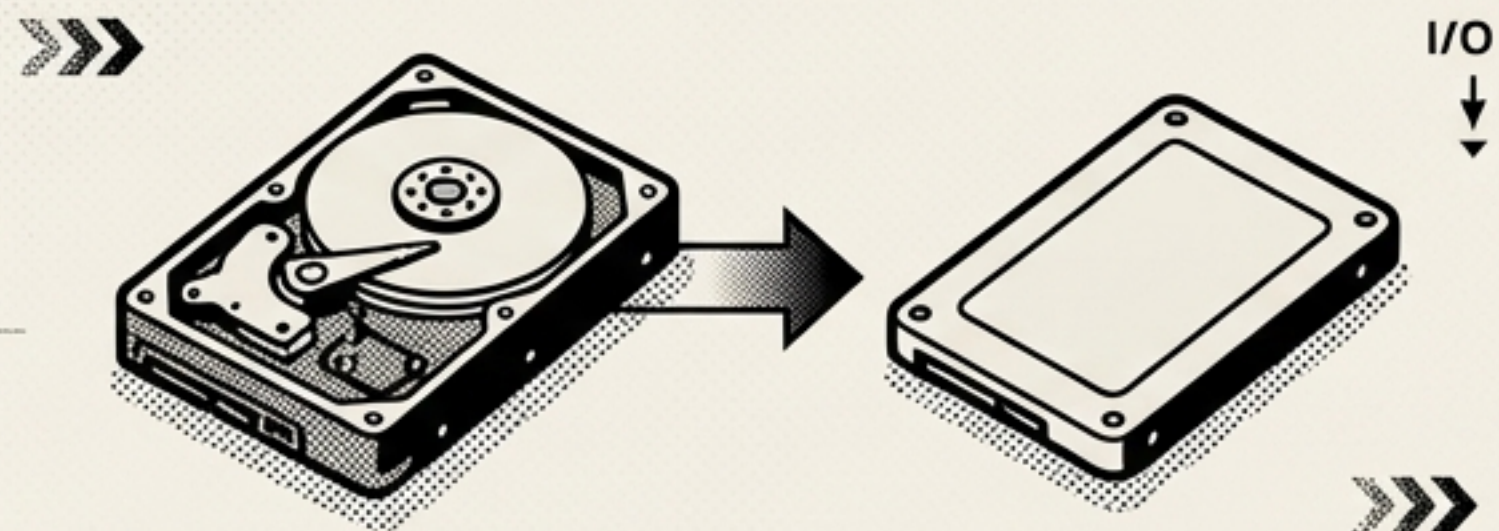
หน้าต่างที่ใกล้ชิดผู้ใช้งานมากที่สุด โดยซ่อนความซับซ้อนทั้งหมดของระบบหลังบ้านไว้!



DATA INDEPENDENCE (ความเป็นอิสระของข้อมูล)

แอปเกรดระดับล่าง โดยไม่ต้องรู้โปรแกรมระดับบน!

Physical Data Independence

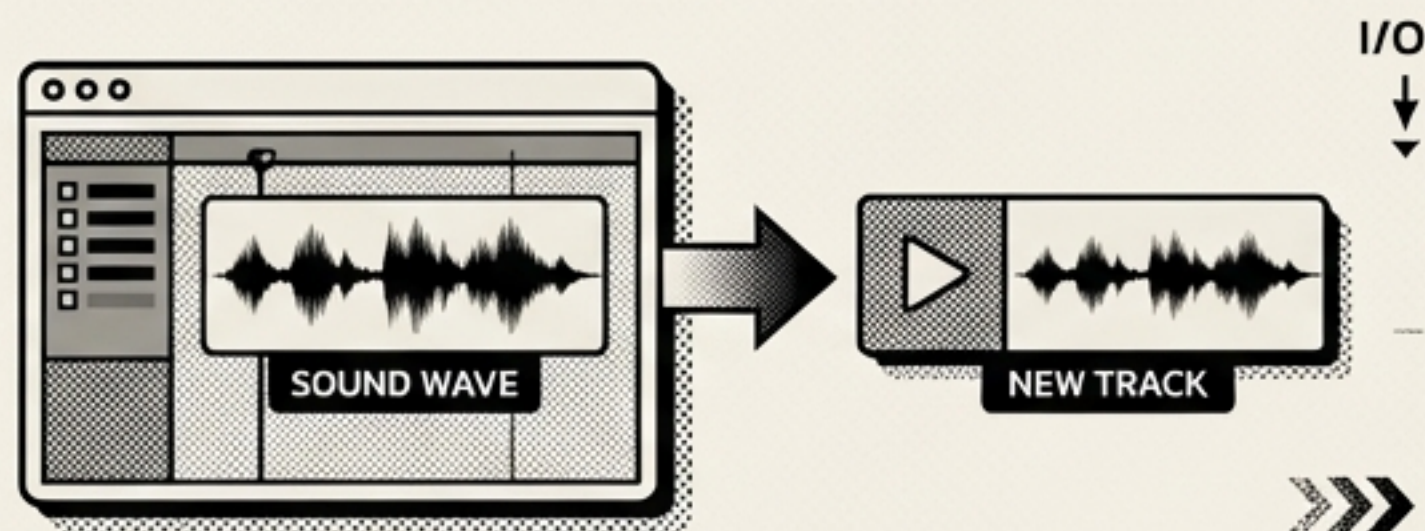


เปลี่ยนแปลงระดับ Internal (เช่น แอปเกรดเปลี่ยน Harddisk เป็น SSD) -> โครงสร้างตาราง (Conceptual Level) ยังคงทำงานได้ปกติโดยไม่ต้องแก้ไขโค้ด!

“เปลี่ยนสายลำโพง แต่เพลงยังบิตเดิม”

POWER

Logical Data Independence



เปลี่ยนแปลงระดับ Conceptual (เช่น เพิ่มคอลัมน์ใหม่ในตาราง) -> โปรแกรม/หน้าจอผู้ใช้ (External Level) ที่ไม่ได้ใช้คอลัมน์นั้น ยังคงทำงานได้ปกติ!

“รีมิกซ์บิตใหม่” แต่เนื้อร้องที่คนดูร้องตามยังเหมือนเดิม

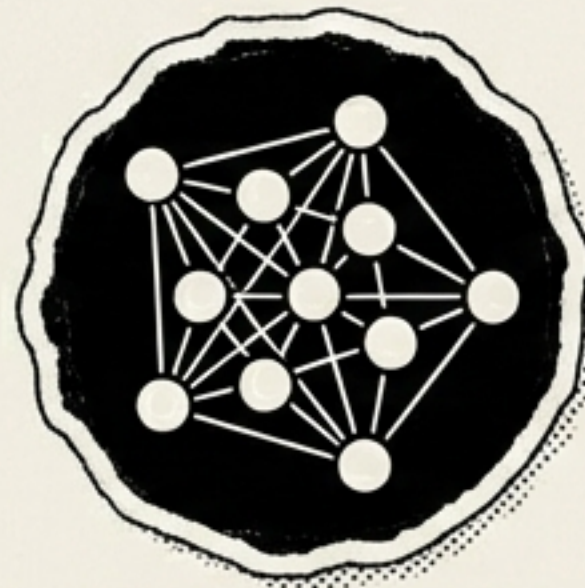
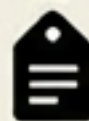
นี่คือจุดเด่นที่ทำให้ DBMS ทรงพลังกว่าการเก็บเป็นแฟ้มข้อมูล (File System) ธรรมดา!

THE 4 FLOWS: แบบจำลองข้อมูล (DATA MODELS)

การออกแบบโครงสร้างฐานข้อมูล (Implementation Models) ที่นิยมใช้ มี 4 สไตล์หลัก:



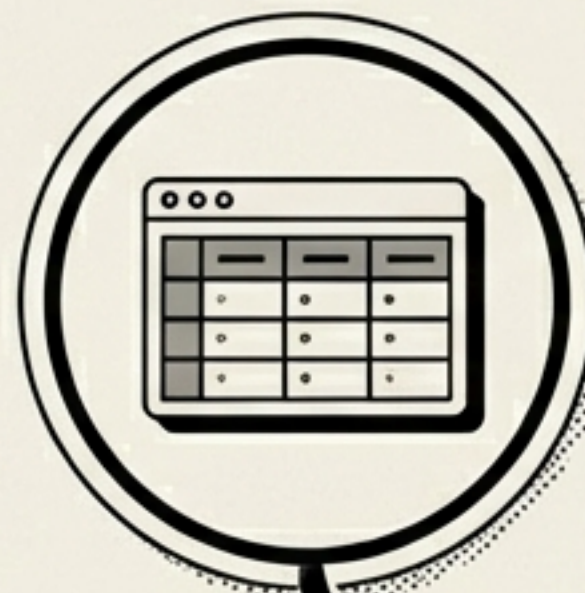
1. Hierarchical:
ต้นไม้แบบลำดับชั้น
(Tree)



2. Network:
โยงใยแบบเครือข่าย
(Net)



3. Relational:
ตารางแบบสัมพันธ์
(Table)



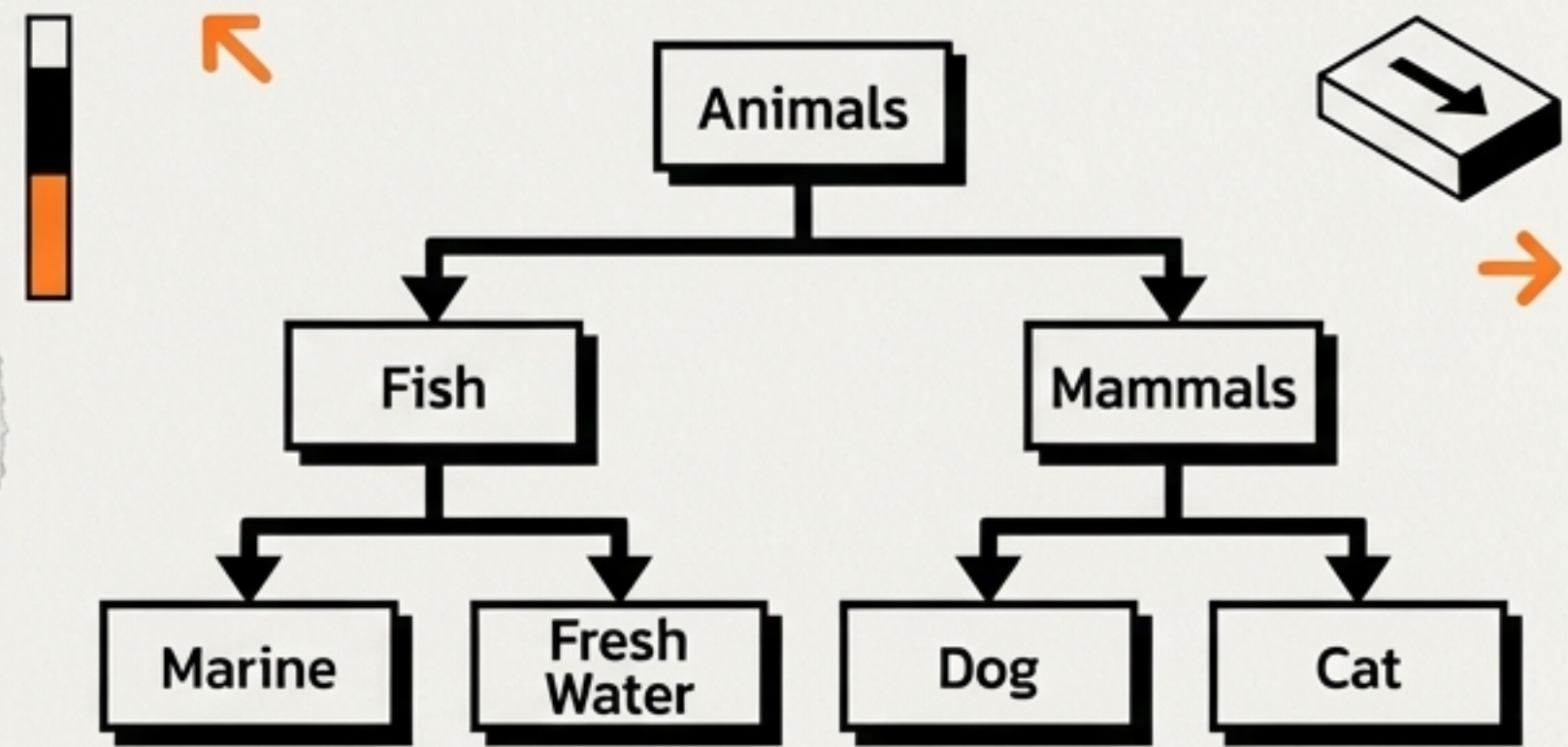
4. Flat File:
แฟ้มเดียวเรียบง่าย
(Simple Table)



การเลือกสไตล์
ขึ้นอยู่กับการวิเคราะห์ความสัมพันธ์
(Relationships) ของข้อมูล!

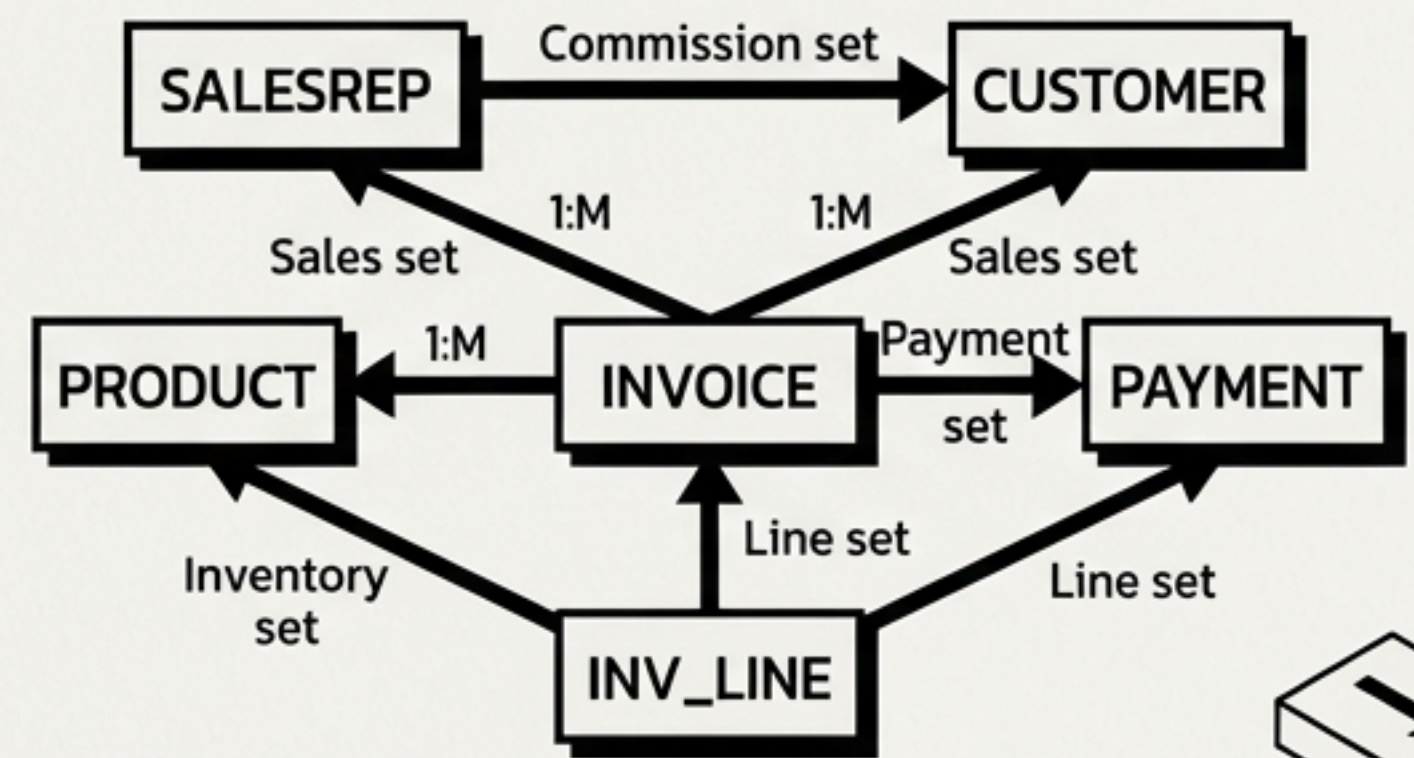
THE OLD SCHOOL FLOW

Hierarchical Model (แบบลำดับชั้น)



- ความสัมพันธ์แบบ 1 ต่อหลาย (1:M)
- พ่อแม่มีลูกได้หลายคน แต่ลูกมีต้นกำเนิด (Parent) ได้เพียงแปมเดียว เท่านั้น!
- ยุคเมนเฟรม (เช่น IMS/VS)

Network Model (แบบเครือข่าย)



- ความสัมพันธ์แบบ หลายต่อหลาย (M:M)
- พัฒนาต่อยอดให้ยืดหยุ่นขึ้น ลูก (Child) สามารถมีต้นกำเนิด (Parent) ได้มากกว่า 1 แปมข้อมูล!
- ภาษาที่ใช้: (เช่น IDMS)



THE MODERN ERA

Relational Model (ตารางสัมพันธ์) [Most Popular]

Items

itemid	orderid	item	amount
5	1	Chair	200.00
6	1	Table	200.00
7	1	Lamp	123.12

Customers

customerid	name	email
5	Rosalyn Rivera	rosalyn@adatum.com
6	Jayne Sargent	jayne@contoso.com
7	Dean Luong	dean@contoso.com

Orders

orderid	customerid	date	amount
1	4	11/1/17	523.12
2	3	11/15/17	32.99
3	1	11/21/17	23.99

- ข้อมูลอยู่ในรูป ตาราง (Table) ที่เป็นอิสระต่อกัน เชื่อมโยงกันผ่านข้อมูลที่มีอยู่ตรงกัน (Attribute Attribute / Key)
- ภาษาที่ใช้คือ SQL (เป็นมาตรฐานอุตสาหกรรมในปัจจุบัน!)

Flat File Model (แบบรายแฟ้ม)

ProductID	ProductName	Price	Qty	Total
3	Product 3	300	3	900.00
1	Product 1	100	1	100.00
4	Product 4	400	1	400.00

- ตารางเดี่ยวๆ ไม่มีความสัมพันธ์ (No relationship) กับใคร
- เรียบง่ายสุด เหมาะกับข้อมูลน้อยๆ (เช่น CSV, Address Book)

HAZARD

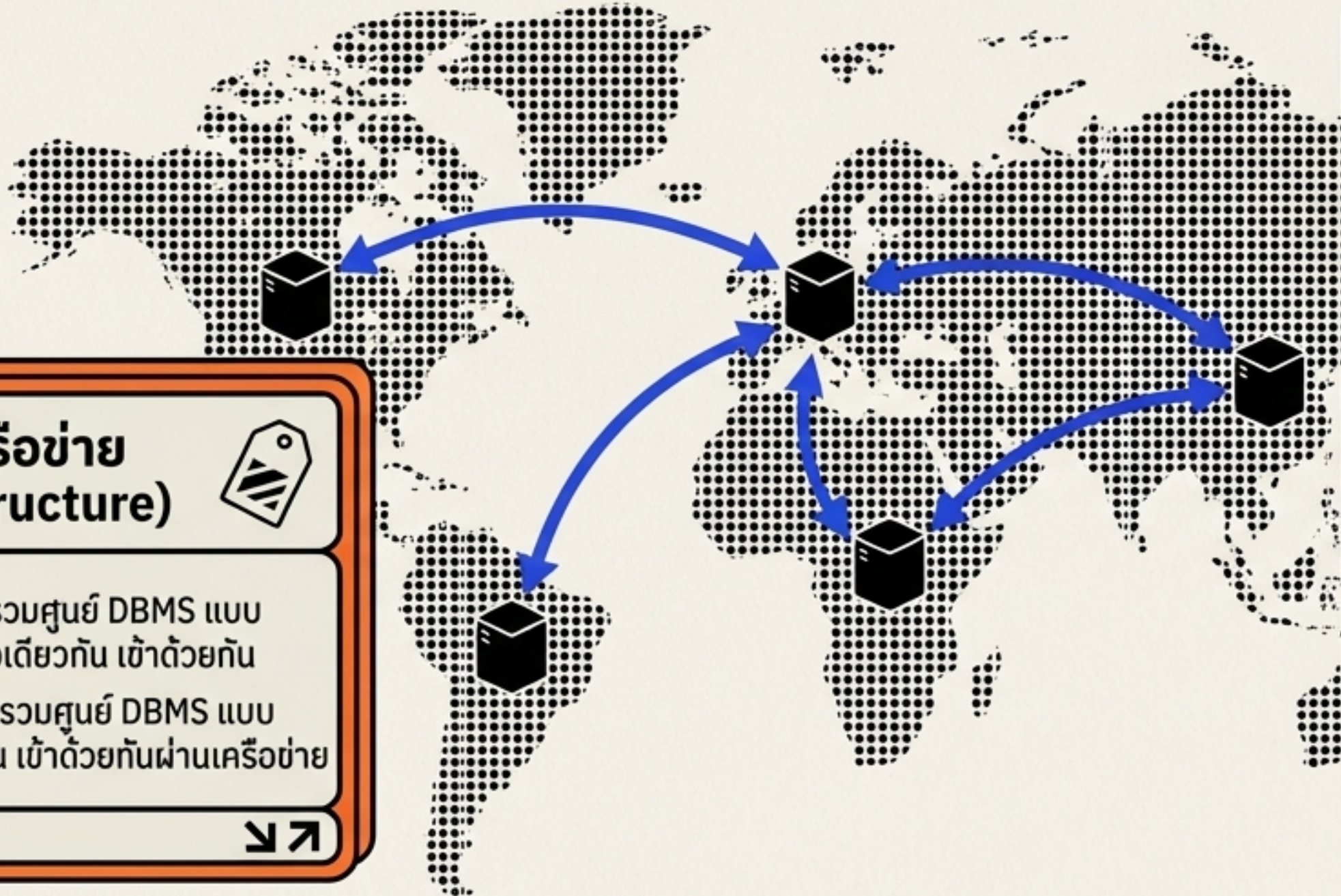
THE CYPHER: Data Models Matrix

แบบจำลอง (Model)	โครงสร้าง (Structure)	ความสัมพันธ์ (Relationship)	ข้อควรจำ (Vibe)
Hierarchical	Tree (ต้นไม้)	1:M (มีพ่อได้คนเดียว)	คลาสสิก ยุค Mainframe
Network	Web/Net (ใยแมงมุม)	M:M (มีพ่อได้หลายคน)	ซับซ้อนแต่ยืดหยุ่นกว่า Tree
Relational	Tables (ตาราง)	โยงด้วย Attribute กลาง	ราชาองค์ปัจจุบัน! (ใช้ SQL) ยืดหยุ่นสูงสุด
Flat File	Single Table (แฟ้มเดียว)	ไม่มี (No Relationship)	เรียบง่ายที่สุด ไฟล์เดี่ยวจบ

Relational Model คือผู้ชนะในยุคปัจจุบัน
เพราะการแยกเป็นตารางย่อยช่วยลดความซ้ำซ้อนของข้อมูลได้ดีที่สุด!

DISTRIBUTED DATABASE (ระบบฐานข้อมูลแบบกระจาย)

Taking it Global (MPMD: Multiple-site Processing, Multiple-site Data)



โครงสร้างเครือข่าย (Network Structure)



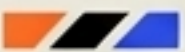
- Homogeneous: รวมศูนย์ DBMS แบบระบบเดียวกัน/ฮาร์ดแวร์เหมือนกัน เช้าด้วยกัน
- Heterogeneous: รวมศูนย์ DBMS แบบต่างแพลตฟอร์มกัน เช้าด้วยกันผ่านเครือข่าย



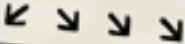
การทำซ้ำข้อมูล (Data Replication)



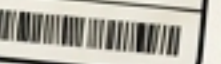
- ก๊อปปี้ฐานข้อมูล (บางส่วนหรือทั้งหมด) ไปเก็บไว้หลายๆ Server ทั่วโลก!
- ทำไปทำไม?
 1. ลดโหลดแบนด์วิดท์เครือข่าย (เข้าถึงข้อมูลจาก Server ที่ใกล้ที่สุด)
 2. ทำงานทดแทนกันได้ทันที (Failover) เมื่อมี Server ใดล่ม
- Note: ต้องพิจารณาขนาดข้อมูลและ ความถี่ในการใช้งาน



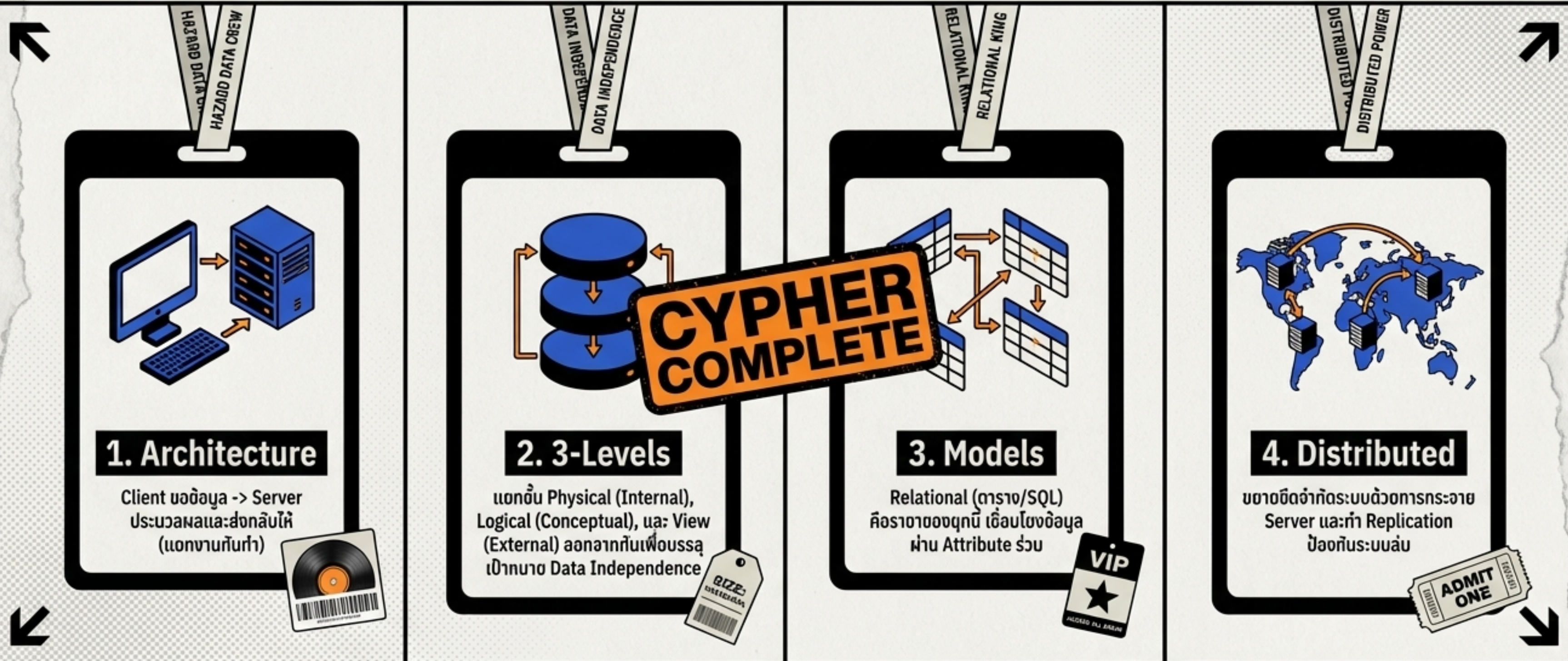
VIP



HAZARD



MASTER THE DATA GAME



ข้อมูลคือบิตของระบบ การจัดการฐานข้อมูลได้ คือการควบคุมสแตจ!