

Full Stack Development Base on Reference Data Model

**นำเสนอด้วย นายสิทธิพงศ์ จำรัสกุลธิรังค์
อาจารย์ที่ปรึกษา รองศาสตราจารย์ ดร.วรา วรावิทย์**

เส้นทางการเรียนรู้ ก่อนเริ่มงานจริง

01

ศึกษา Full Stack
Cloud Native
(Docker,
Postgres, PHP,
React)

02

เรียนรู้ Data Model
จาก Data Model
Resource Book

03

อ่านแบบ Database
ด้วย Visual
Paradigm

04

สร้าง CRUD App
จัดการ Person ด้วย
Database จาก
หนังสือ

ເສັນຖານກາຣເຣຍນຮ່ງ ກ່ອນເຣີມທານຈົງ

05

ສ້າງ CRUD App ຈັດກາ
Person ດ້ວຍ Database
ຈາກ Visual Paradigm

06

ສ້າງ CRUD App ຈັດກາ
Person ດ້ວຍ MVC
(React ເປັນ View)

07

ສ້າງ CRUD App
ຈັດກາ Geo Data
(Laravel, Artisan)

08

ໃຊ້ Pandas ETL(extract
transform load) ບໍາ Dataset
Geo ຈາກກຽມກາປົກຄອງເຂົາສູ່
app

ເສັ້ນທາງກາຣເຮືຍນັ້ນ ກ່ອນເຮີມທານຈົດ

09

ພັບປາ Dynamic
Dropdown
(Country, Province,
District)

10

ປັບປຸງ Laravel
App (Response
Time: 7s → 1s)

11

ສ້າງ CRUD App
ຈັດກາຣ Person-
Organization ດ້ວຍ
Slim PHP

12

ພັບປາ CRUD App
ເພີ່ມ Role,
Relationship,
Communication
Event

เส้นทางการเรียนรู้ ก่อนเริ่มงานจริง

13

สร้าง CRUD App
เดิม เพิ่ม JWT
Authentication
ด้วย FastAPI

14

ปรับ UX/UI ตามคำ
แนะนำจาก
ผู้ช่วยศาสตราจารย์
ภานุศรี ณ รังษี
ครุสาขาออกแบบ

15

ปรับปรุง UI ตามข้อเสนอ
แนะนำผู้ใช้ Gen Z

16

สร้าง Role-Based
CRUD App (6 บทบาท)

ເສັ້ນທາງກາຣເຣຍນຮູ ກ່ອນເຣີມຫານຈົດ

17

ອອກແບບ Diagram ປະກອບຊອບົດແວ່ນ (Activity, ERD,
Sequence, State, Use Case)

ปัญหาคืออะไร?

01. การพัฒนา Full Stack ขาด
โนเดลมาตรฐาน

02. การจัดการข้อมูล Person-
Organization ซับซ้อน

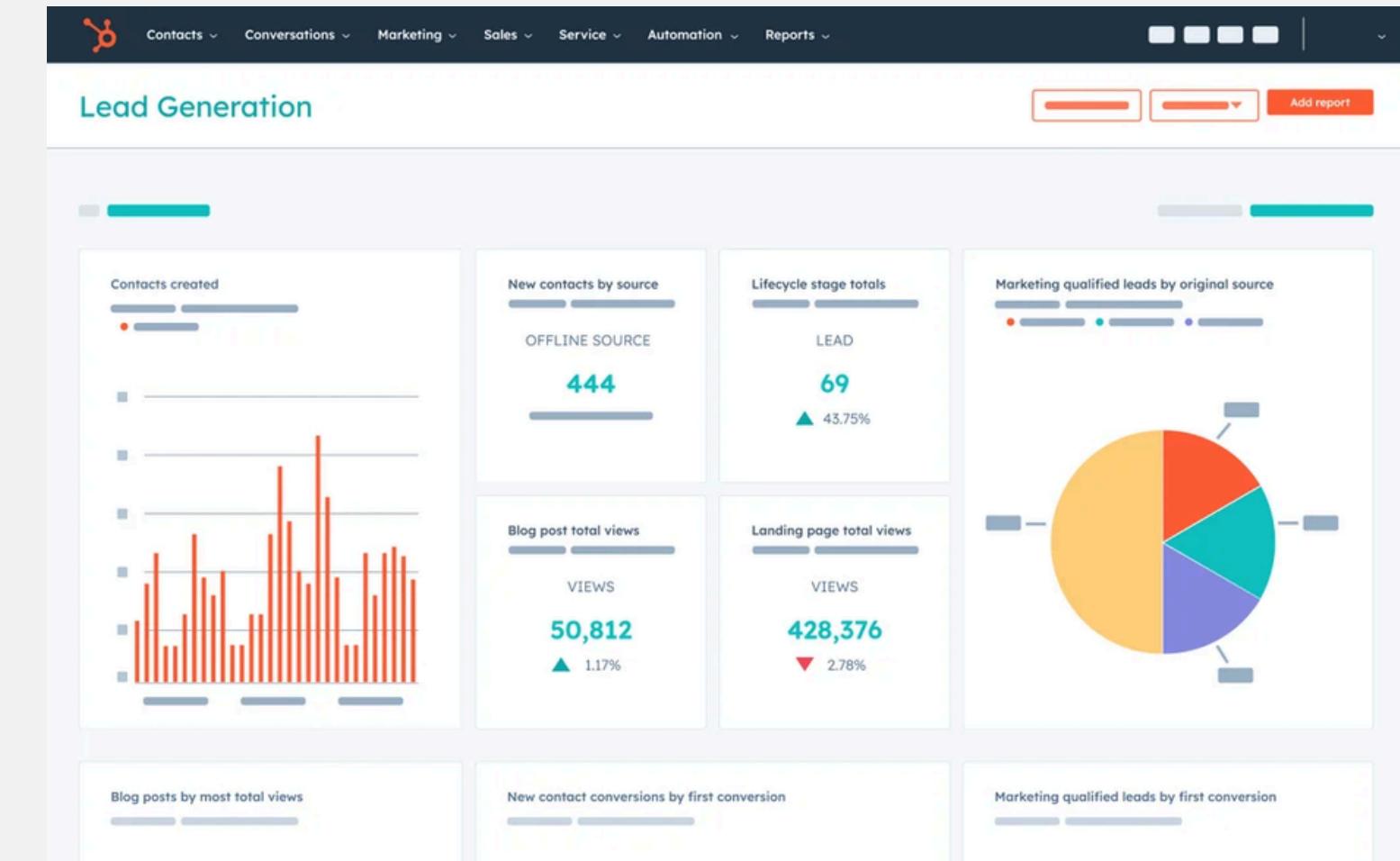
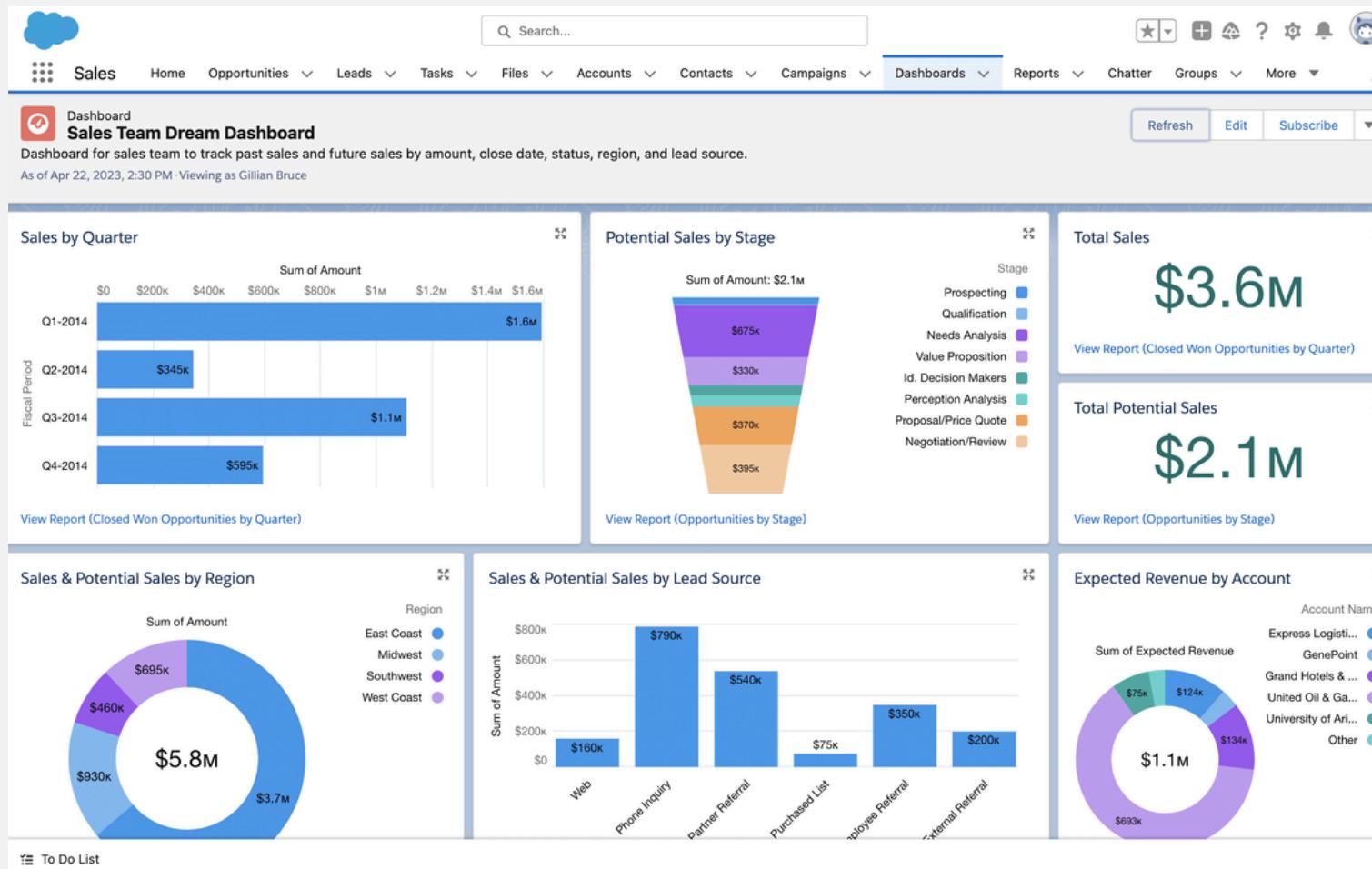
03. การติดต่อสื่อสารขาดการ
จัดเก็บเป็นระบบ

04. ผู้ใช้หลายบทบาทต้องการ
จัดการที่ยืดหยุ่น

ผู้อ่อนทำอย่างไร?

- ใช้ Data Model ทั่วไป ไม่มีมาตรฐาน
- เก็บบันทึกการสื่อสารแบบแยกส่วน
- ระบบ CRM เน้นการขาย
- ไม่ครอบคลุมการจัดการบทบาทหลากหลาย

SAAS ໃກລ້ເຄີຍງັບ App ທີ່ຜມທຳ



Salesforce

ທີ່ 2 ຕັ້ງທີ່ກລ່າວມາ ມີ CRM (Customer Relationship Management)

HubSpot

เบนวคิด App

01. พัฒนา Full Stack ด้วย
Reference Data Model

02. จัดเก็บ Person,
Organization,
Communication Event
เป็นระบบ

03. App เวพาทาง บันทึก
การสื่อสารอย่างมี
โครงสร้าง

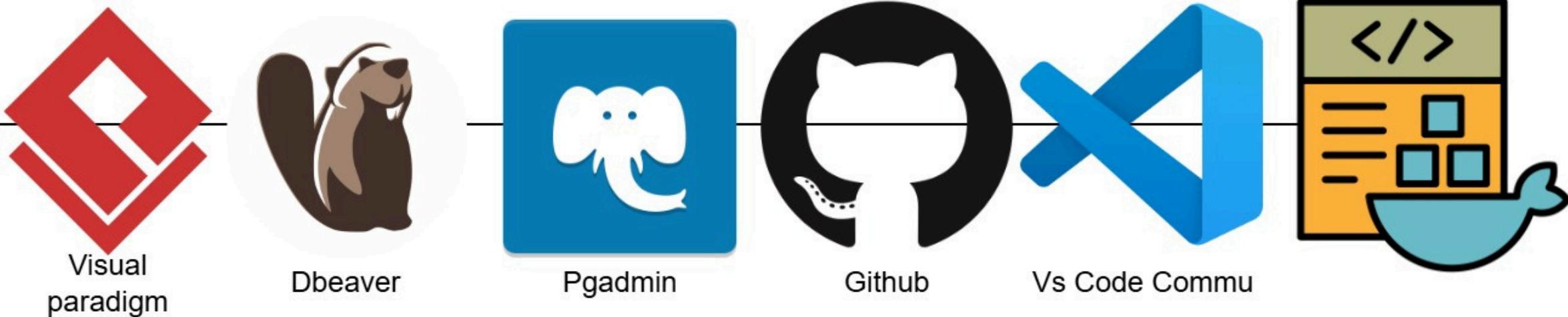
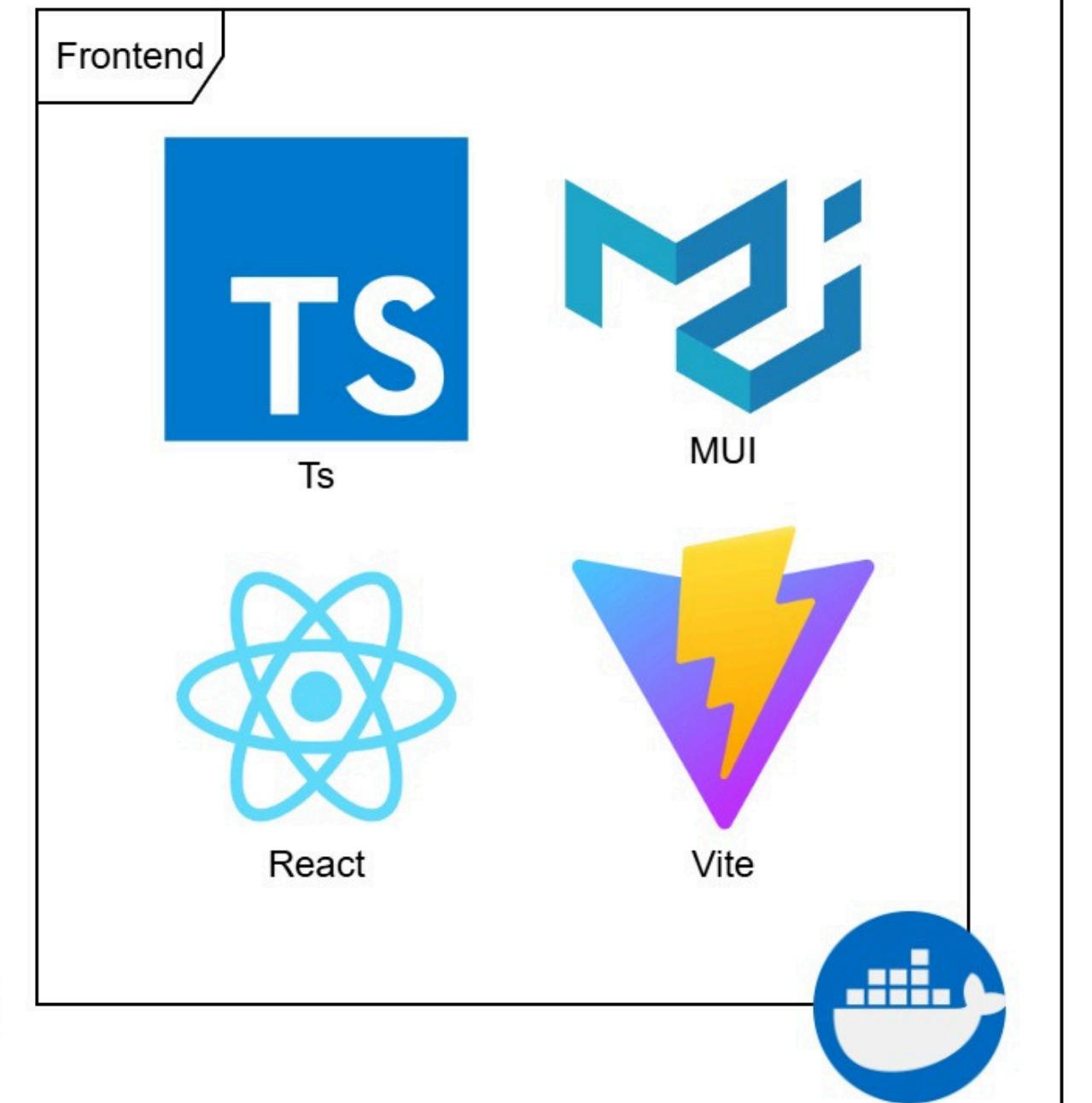
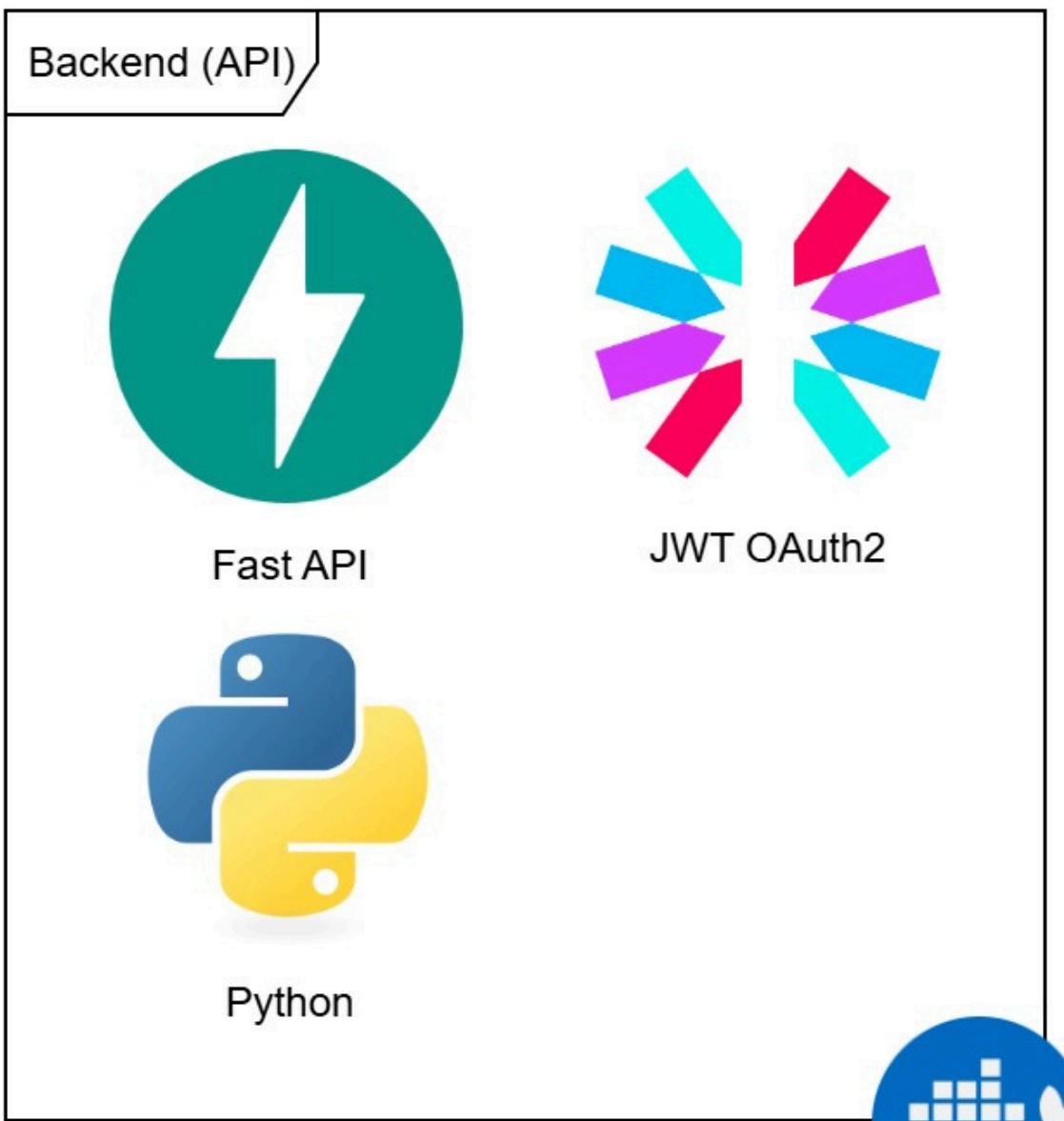
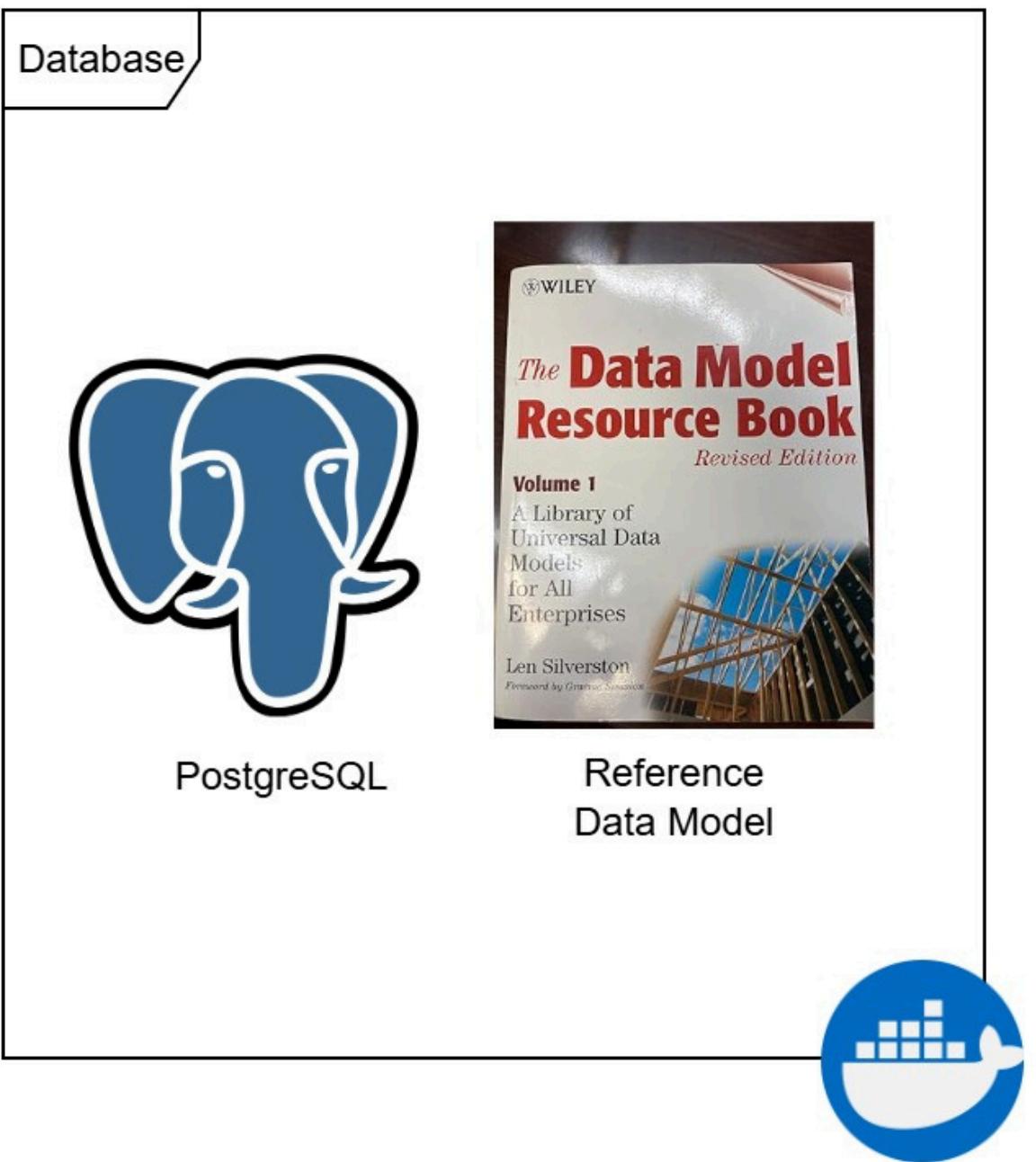
04. รองรับ 6 ภาษาผู้ใช้
ใช้งานยืดหยุ่น



Full Stack Development Based on Reference Data Model

- อ้างอิง Data Model จาก Data Model Resource Book
 - พัฒนา Full Stack Cloud Native App
 - จัดเก็บ Person, Organization, Communication Event
 - ใช้ Authentication รองรับ 6 ภาษาผู้ใช้
 - มี UX / UI กีสวยงาม ใช้งานง่าย
- 

Docker Compose



Data Model ໃຫ້ນັ້ງສູ່

60 Chapter 2

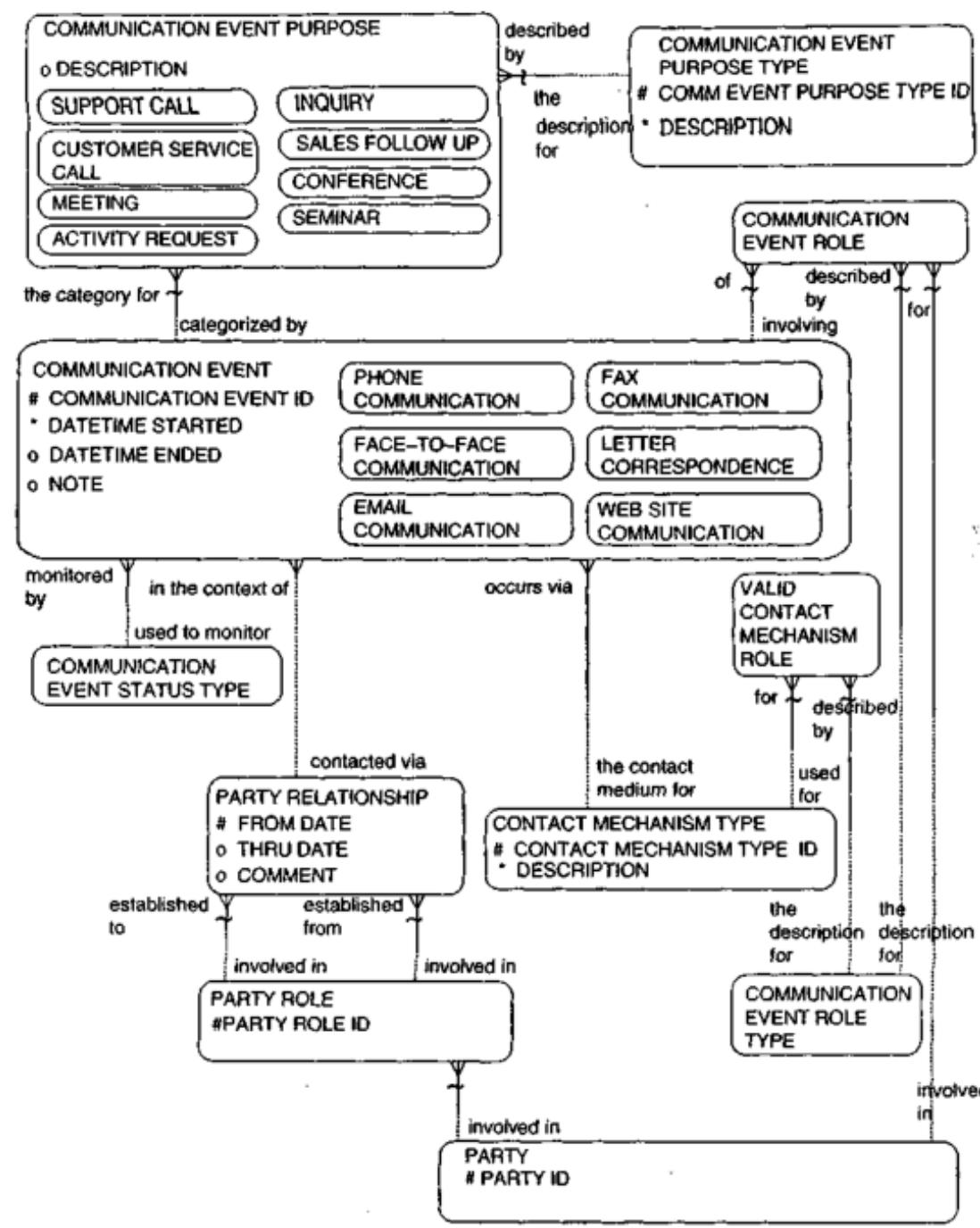


Figure 2.12 Communication event.

30 Chapter 2

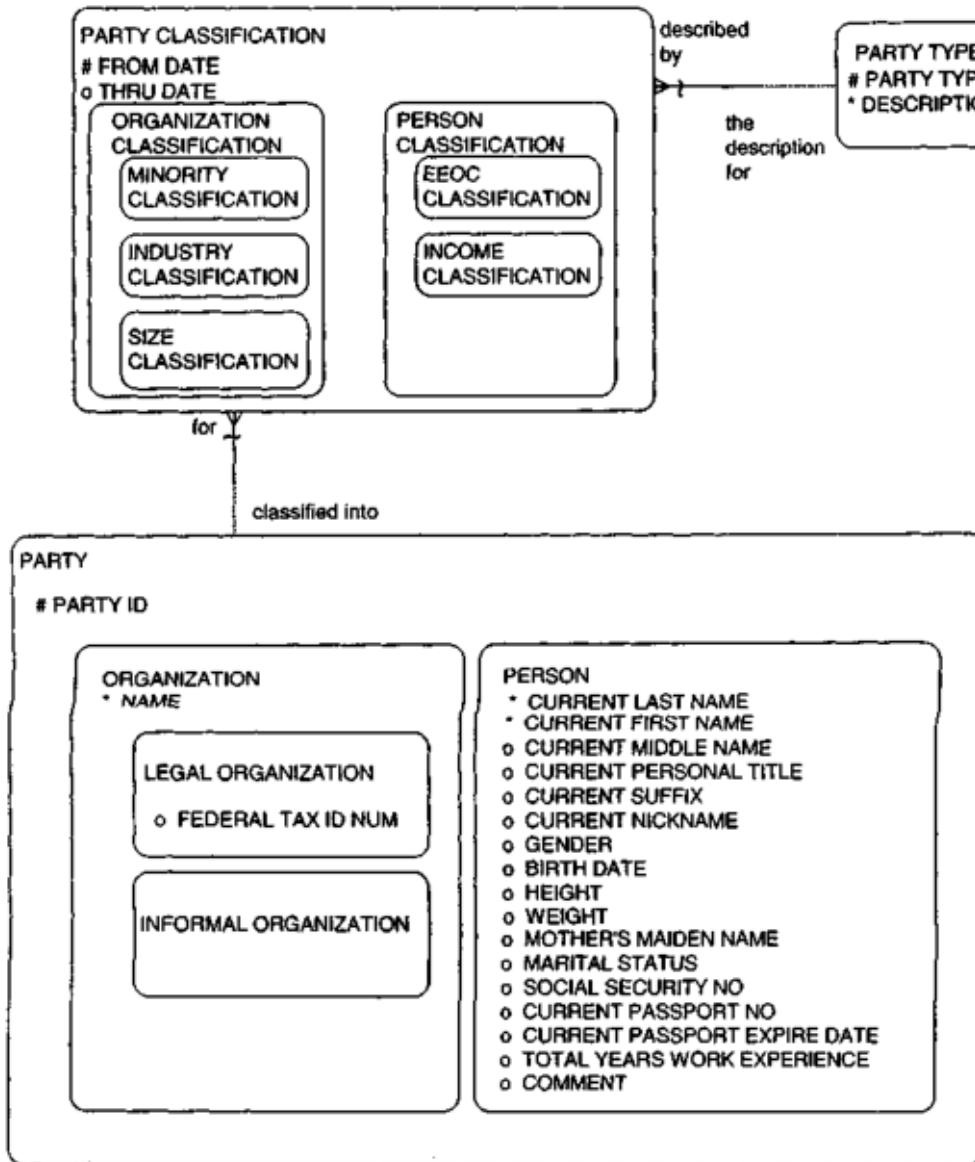


Figure 2.3 Party.

Therefore, Figure 2.3 shows a superentity named PARTY that has as its two subtypes PERSON and ORGANIZATION. This PARTY entity will enable storage of some of the common characteristics and relationships that people and organizations share.

People and Organizations 43

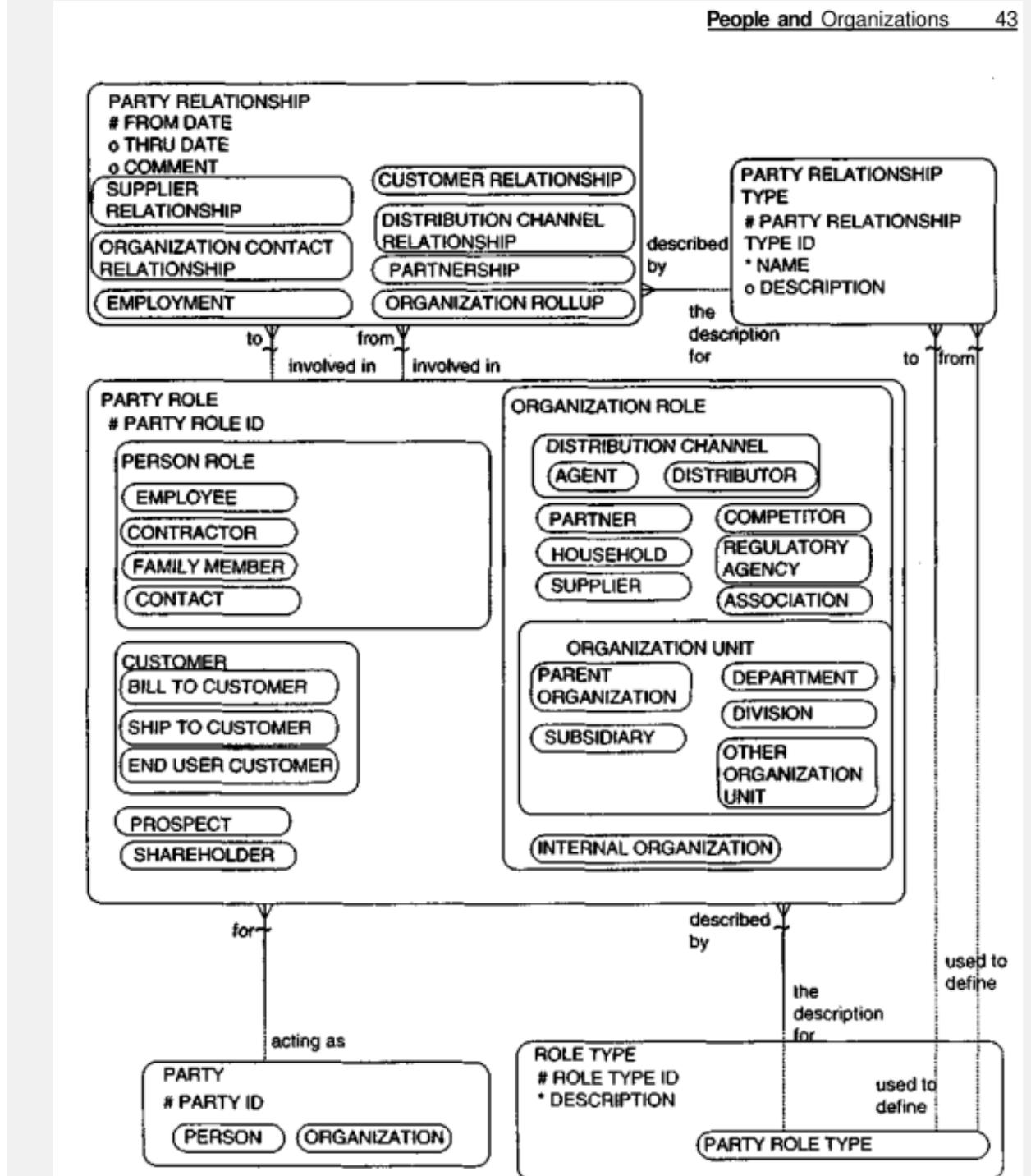
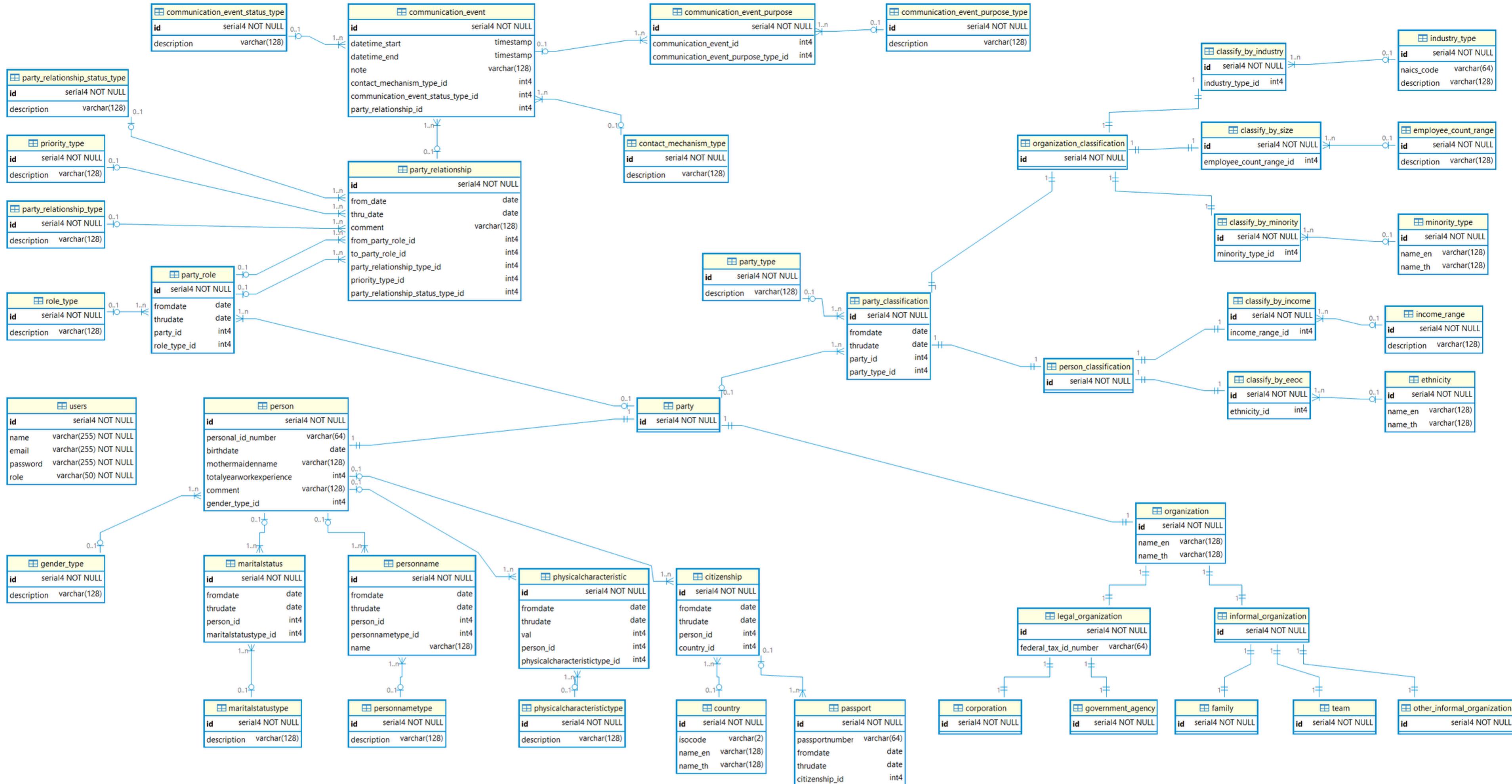
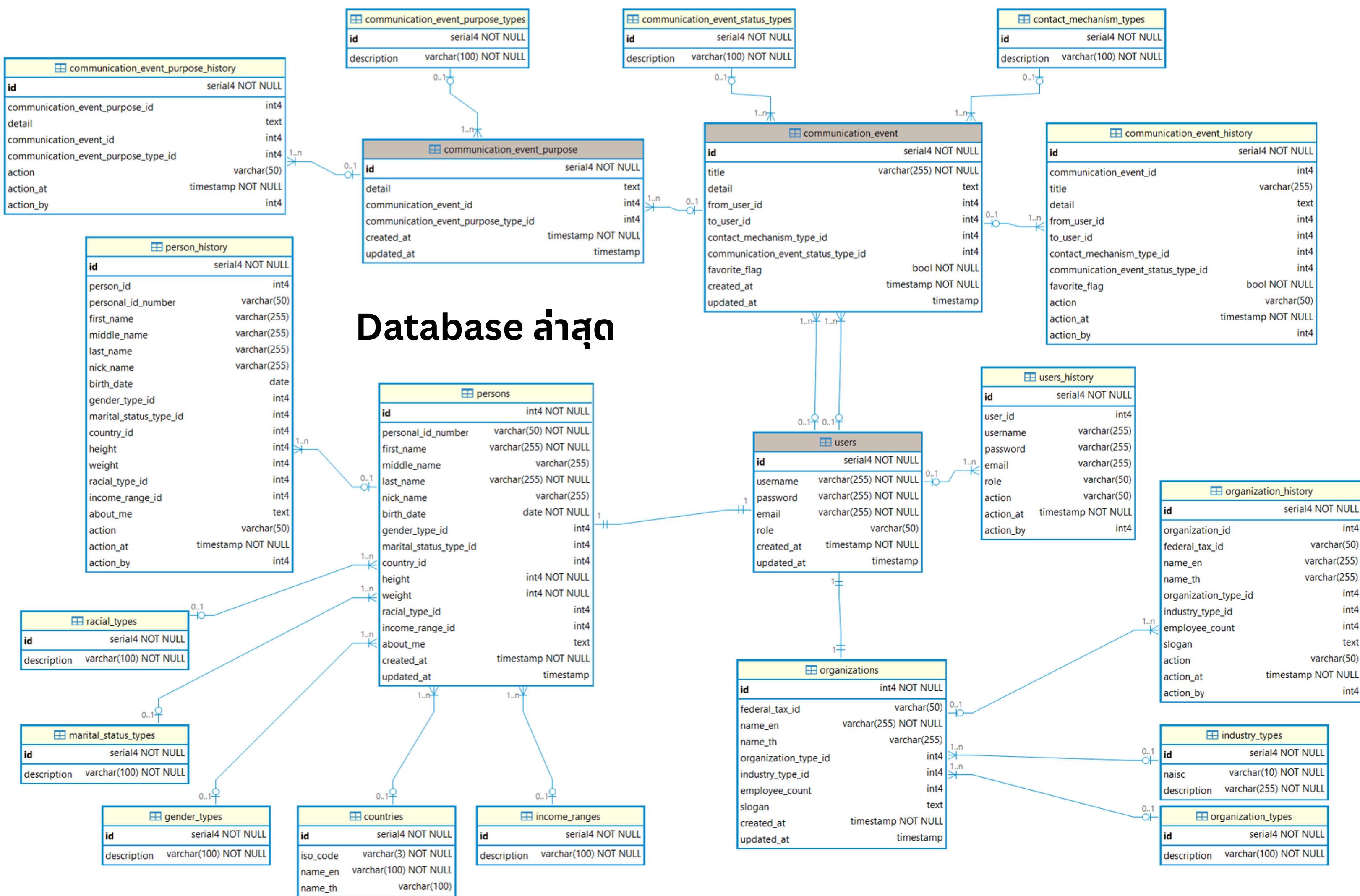


Figure 2.6a Common party relationships.

Data Model ที่ได้แรงบันดาลใจจากหนังสือ





ปัญหาที่เจอ

- การทำ Database รองรับการต่อ�อด และประสิทธิภาพดี เป็นเรื่องยาก
- Full Stack Cloud Native App เป็นเรื่องยาก ต้องเรียนรู้แยกกันส่วน
- หนังสือ Data Model Resource Book เก่า และเข้าใจยาก
- การทำ JWT Authentication ให้รองรับ 6 บทบาทผู้ใช้เป็นเรื่องยาก
- การจัดการ hashed password ในระบบ login เป็นเรื่องยาก
- มี UX / UI ที่เน้นใช้งานง่าย เหมาะกับ Gen Z เป็นเรื่องยาก
- การจัดการข้อมูลประจำวันเวลา เป็นเรื่องยาก
- การจัดเก็บประวัติข้อมูล ว่าข้อมูลอะไร ถูกทำอะไร เมื่อไหร่ โดยใคร เป็นเรื่องยาก

แนวทางต่อยอด

- พัฒนา Full Stack App ให้รองรับทุกภาษา
- ใช้ Reference Data Model อื่นๆ กี่ชั้บช้อนและมีประสิทธิภาพมากกว่าเดิม
- UX/UI ตั้งค่าเปลี่ยน theme เว็บได้ตามต้องการ
- เพิ่ม AI/ML วิเคราะห์ Communication
- ขยายสู่ Mobile App (ใช้ React Native และ via bus)
- นำ App ที่ได้ไป Deploy ในระบบ Cloud Computing อื่นๆ

Q & A