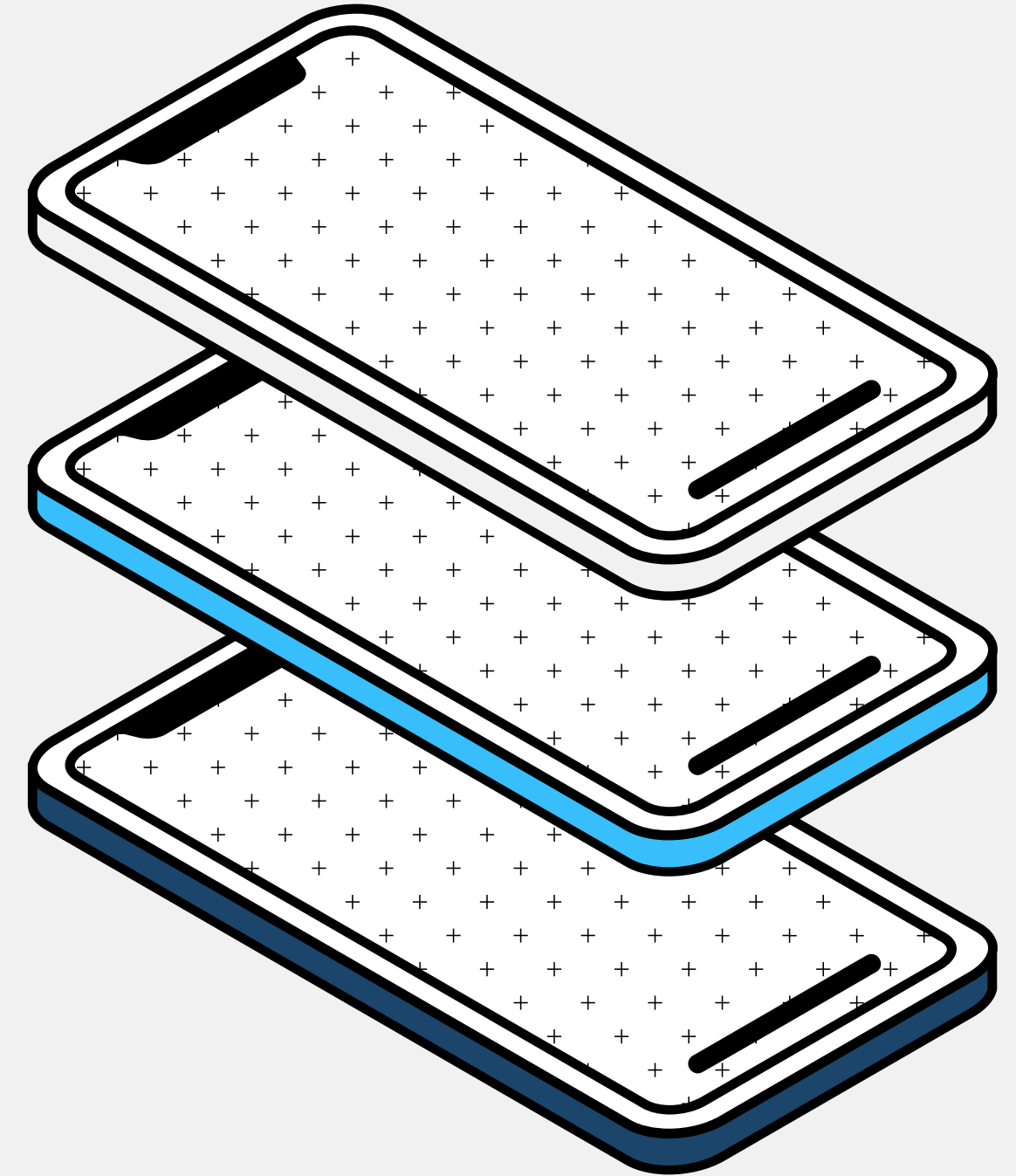


บทที่ 2

สถาปัตยกรรม อุปกรณ์เคลื่อนที่

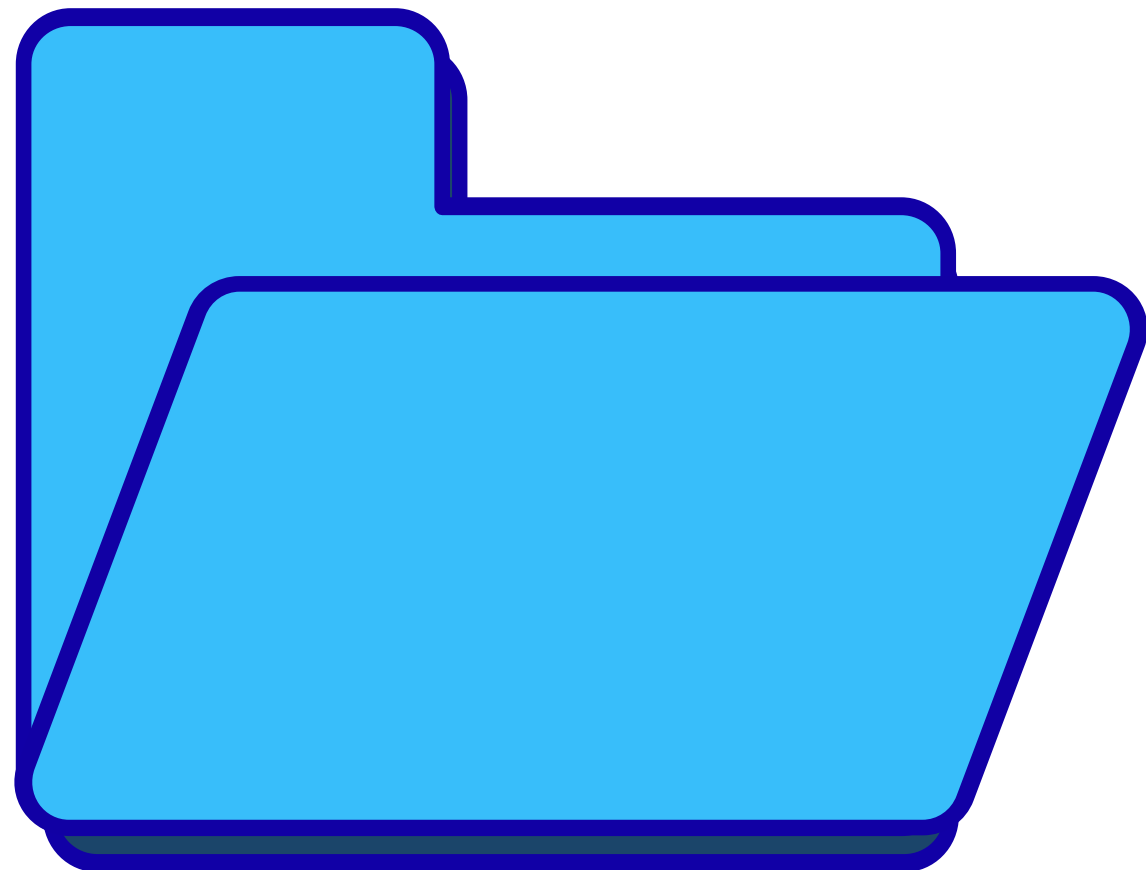
ดร.พงษ์ดนัย จิตตวิสุทริกุล





Topic

5 หัวข้อหลัก:



สถาปัตยกรรมไมโคร (Microarchitecture)

System-on-a-Chip (SoC)

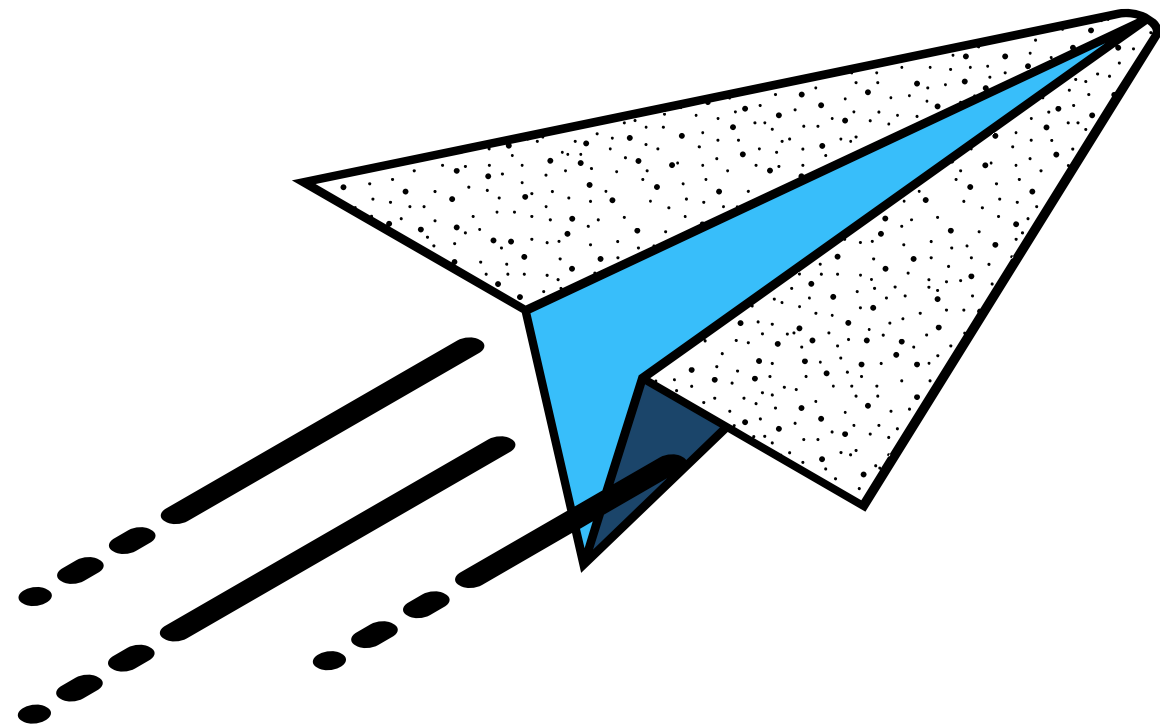
สถาปัตยกรรมระบบปฏิบัติการ

การจัดการทรัพยากรที่สำคัญ

เซ็นเซอร์ (Sensors)

Objective behavior

6 วัตถุประสงค์เชิงพฤติกรรม:



อธิบายความแตกต่างระหว่าง CISC และ RISC

ระบุหน้าที่ของส่วนประกอบหลักใน SoC

เปรียบเทียบสถาปัตยกรรมแบบชั้น Android & iOS

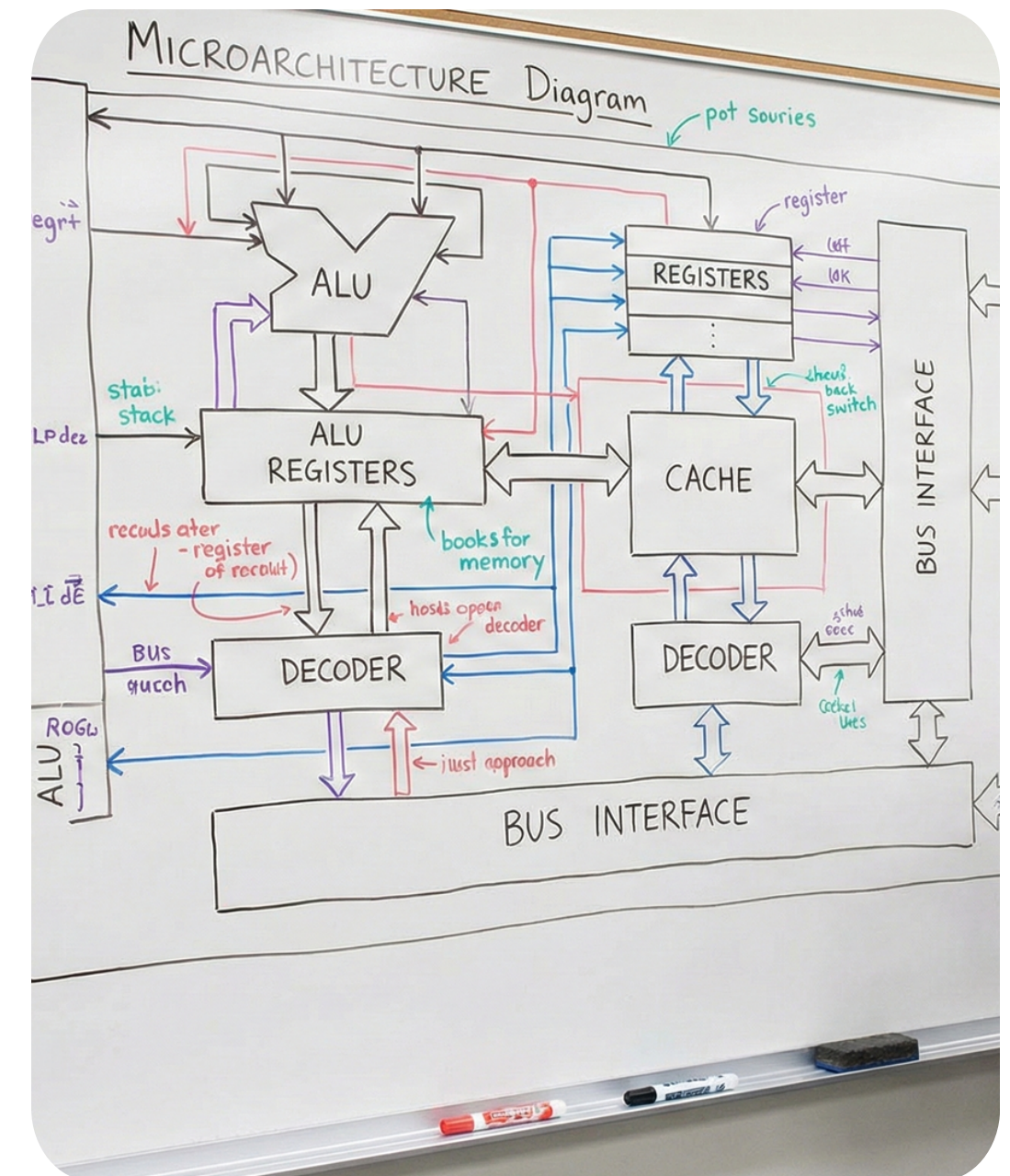
อธิบายผลกระทบของปรัชญาที่แตกต่างกัน

เปรียบเทียบกลไกการจัดการหน่วยความจำ

เปรียบเทียบกลยุทธ์การจัดการพลังงาน

สถาปัตยกรรมไมโคร (Microarchitecture): พิมพ์เขียวของหน่วยประมวลผล

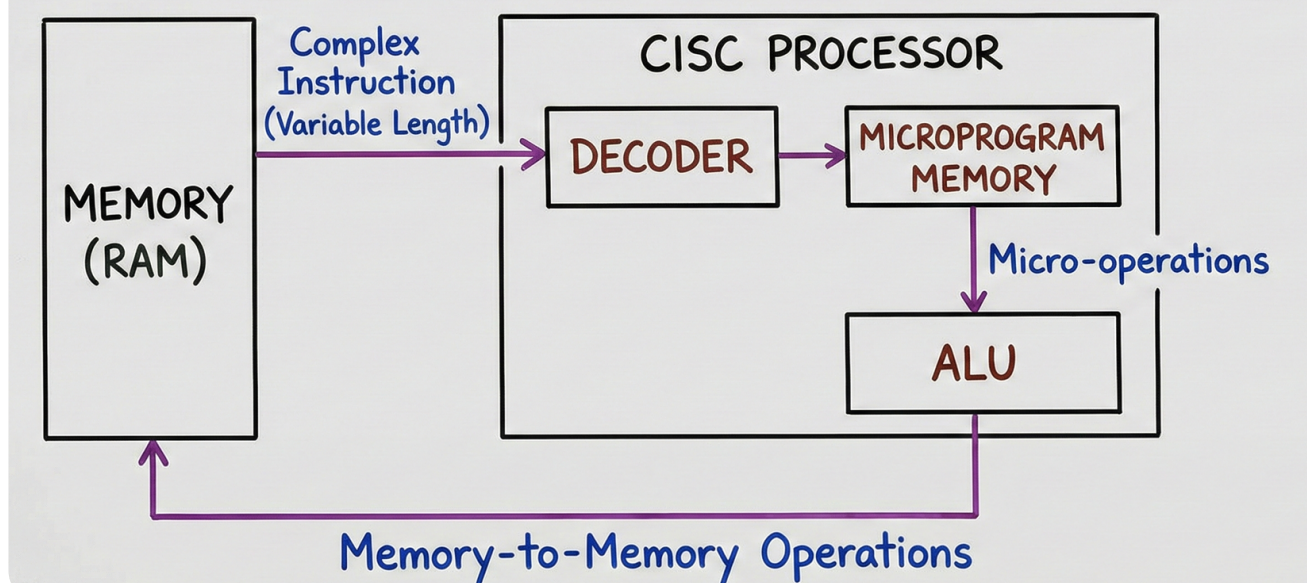
- การออกแบบเชิงตรรกะให้ CPU มีประสิทธิภาพสูงสุด
- ส่วนประกอบหลัก: ALU, CU, register และลำดับชั้นของหน่วยความจำแคช
- ชุดคำสั่ง: รายการเมนูอาหาร
- สถาปัตยกรรมไมโคร: การออกแบบห้องครัว
- CISC (ซับซ้อนแต่ช้า) VS RISC (เรียบง่ายแต่เร็ว)



CISC: คำสั่งเดียวครอบจักรวาล

- หนึ่งคำสั่ง สามารถทำงานที่ซับซ้อนได้หลายขั้นตอน
- ลักษณะเด่น: คำสั่งซับซ้อน และจำนวนคำสั่งน้อย
- ตัวอย่าง: x86 Intel & AMD
- ข้อแลกเปลี่ยน: ใช้เวลาทำงานแต่ละคำสั่งนาน และใช้พลังงานสูง
- เปรียบเสมือน: พ่อครัวหัวป่าก์

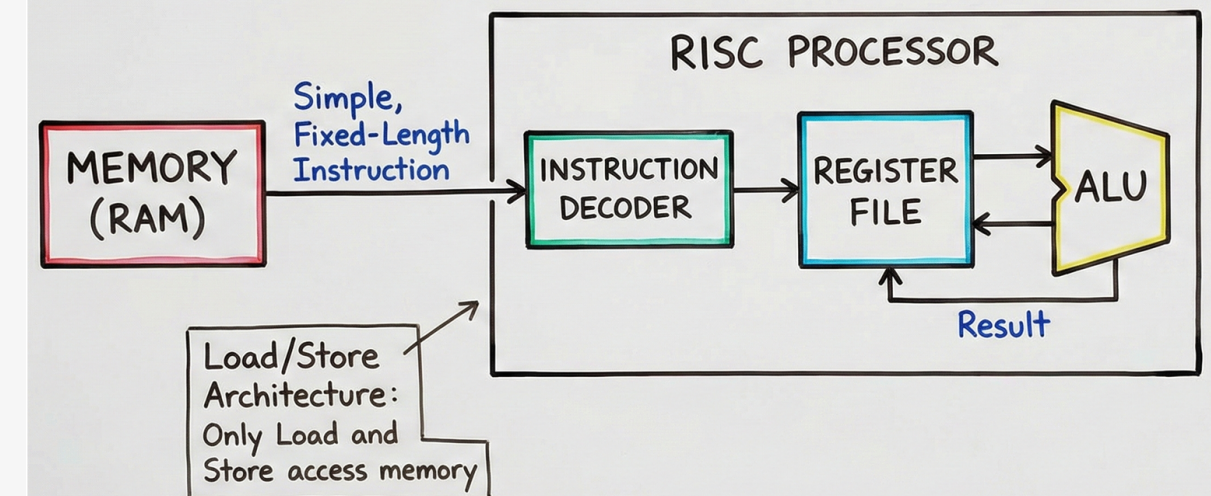
CISC ARCHITECTURE DIAGRAM

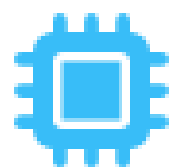


RISC: เรียบง่าย รวดเร็ว และประหยัดพลังงาน

- คำสั่งง่าย ๆ ทำงานพื้นฐาน มีหลายคำสั่ง
- ลักษณะเด่น: คำสั่งพื้นฐาน จำนวนคำสั่งมาก และความเร็วต่อคำสั่ง
- ตัวอย่าง: ARM
- เปรียบเทียบ: กระบวนการผลิตทรานซิสเตอร์
- ข้อแลกเปลี่ยน: ใช้เวลาทำงานแต่ละคำสั่งสั้น ใช้พลังงานน้อย แต่มีหลายคำสั่ง
- เปรียบเสมือน: ครั้วร้านฟาสต์ฟู้ด

RISC ARCHITECTURE DIAGRAM





CISC (X86)



พ่อครัวหัวป่าก์

RISC (ARM)



ร้านฟาสต์ฟู้ด

ปรัชญา

คำสั่งซับซ้อน (Complex)

คำสั่งเรียบง่าย (Reduced)

ชุดคำสั่ง

1 คำสั่งทำหลายอย่าง (หลาย Cycle)

เน้นความเร็ว (1 Cycle)

ขนาดโค้ด

สั้นกว่า

ยาวกว่า

การใช้
พลังงาน

สูง

ต่ำ

ความซับซ้อน
ของฮาร์ดแวร์

ซับซ้อน ใช้ทรานซิสเตอร์จำนวนมาก

เรียบง่าย ใช้ทรานซิสเตอร์จำนวนน้อย

การใช้งาน
หลัก

คอมพิวเตอร์เดสก์ท็อป เซิร์ฟเวอร์

อุปกรณ์เคลื่อนที่ ระบบฝังตัว



-
- 7TE (1) +5A USE ONLY (3) PERITEL ONLY VERSION
- IC18 - IC20
R07 - R09
C41 - C47
X2
- (2) -2A USE ONLY
- IC100, TR100
R100 - R118
C48, C100 - C111
L100, L101
- R28-R30, R44, R46-R56, R58-R60
C22-C26, C28, C29, C32, C33, C49, C50
L2-L3
V1 5E 5A/10 1/2 570p
- PERITEL RGB
- CASSETTE PCB
- JOYSTICKS
- KEYPAD
- RS232/MIDI
- CENTRONICS
- 765-FDC
- MC1376
- MC1488
- MC1489
- AY-3-8912
- 40077
- GATE ARRAY
- IC1
- IC4
- IC5
- IC6
- IC9
- IC10
- IC11
- IC12
- IC13
- IC14
- IC16
- TEA2000

ส่วนประกอบหลักใน SoC: ทีมผู้เชี่ยวชาญใน SoC

- CPU ผู้จัดการทั่วไป
- GPU ศิลปิน
- ISP ช่างภาพ
- DSP วิศวกรเสียงและเซนเซอร์
- NPU สมอง AI
- Modem เจ้าหน้าที่สื่อสาร

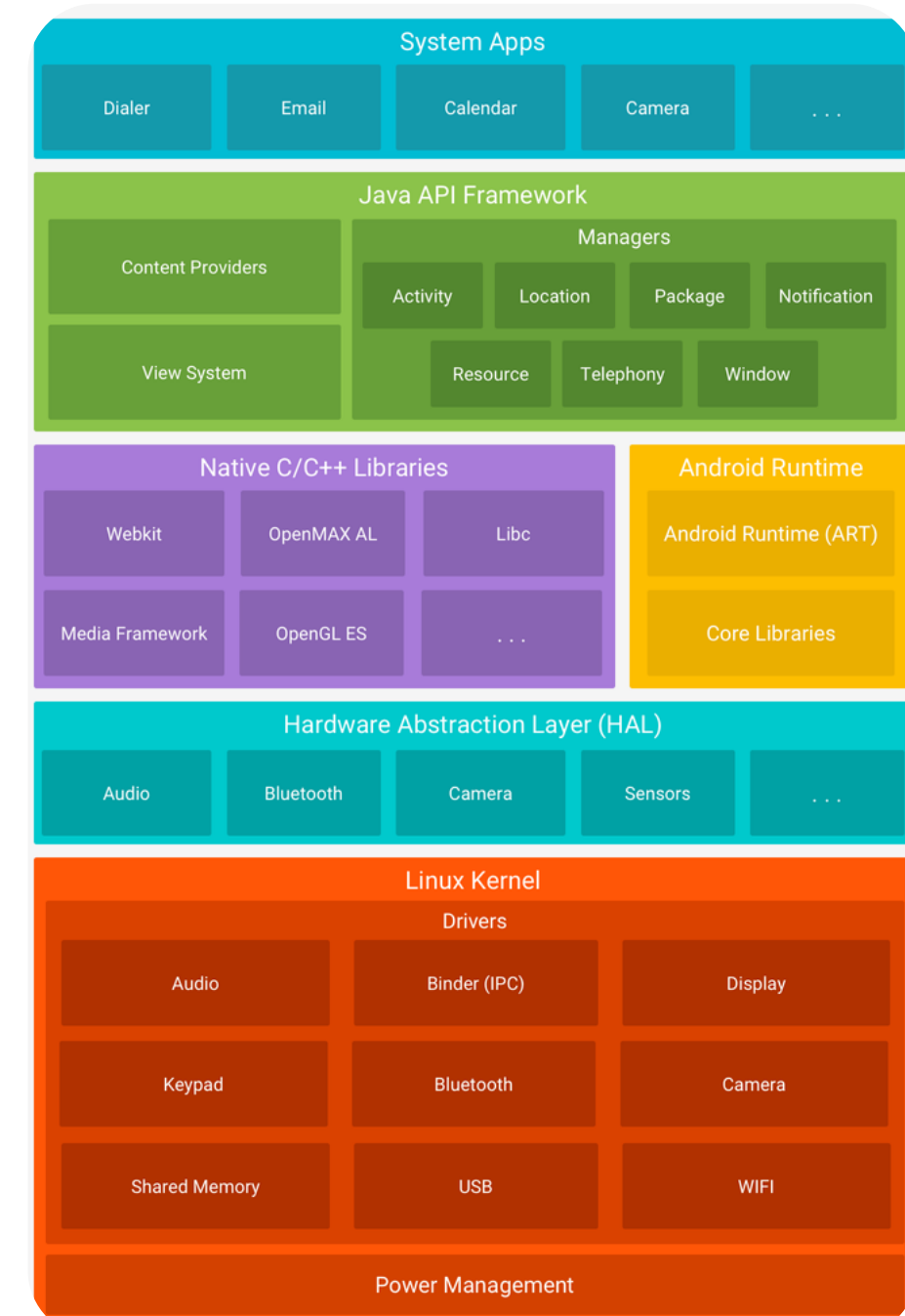




สถาปัตยกรรมแบบชั้นของ Android

- Linux Kernel: ฐานรากที่จัดการฮาร์ดแวร์
- HAL (Hardware Abstraction Layer): "นักแปล" ที่ทำให้ Android รันบนมือถือยี่ห้อไหนก็ได้
- ART (Runtime): ตัวรันโค้ดแอปพลิเคชัน
- Java API Framework: เครื่องมือที่เราใช้เขียนแอป
- System Apps & User Apps: แอปทั้งหมดที่มีในเครื่อง

"ความเปิดกว้าง คือหัวใจของ Android"



สถาปัตยกรรมแบบชั้นของ iOS

- Core OS / XNU Kernel: รากฐานที่จัดการฮาร์ดแวร์
- Core Services: บริการพื้นฐาน (Location, Database)
- Media: กราฟิกและเสียงคุณภาพสูง (Metal)
- Cocoa Touch: ชั้น UI ที่เราสัมผัส

"การบูรณาการแนวดิ่ง เพื่อประสบการณ์ที่ลื่นไหล"





การจัดการหน่วยความจำ

Android Garbage Collection (GC) "รถเก็บขยะ"

ระบบจะหยุดแอปชั่วคราว (Stop-the-world) เพื่อเดินเก็บกวาดหน่วยความจำที่ไม่ใช้แล้วอัตโนมัติ

ข้อดี: Dev สบาย
ข้อเสีย: อาจกระตุก

iOS Automatic Reference Counting (ARC) "เคาน์เตอร์นับจำนวน"

นับจำนวนคนใช้งาน Object ถ้าเหลือ 0 ลบทันที ไม่ต้องรอรถขยะ

ข้อดี: ลื่นไหล
ข้อเสีย: Dev ต้องระวัง Cycle



การจัดการพลังงาน

Android Doze Mode

เมื่อวางเครื่องไว้นิ่งๆ ระบบจะ "จิบหลับ" ตัดเน็ต ตัด CPU และตื่นมาทำงานเป็นรอบๆ (Maintenance Window)

iOS Suspension

ข่มงวดมาก! เข้า Background ปูบ แช่แข็ง (Suspend) ปูบ ยกเว้นแอปพิเศษ (เพลง, GPS)

Sensors: ประเภทสัมผัสของสมาร์ทโฟน

Motion

- Accelerometer ตรวจสอบการเคลื่อนไหวและความเร่ง (การหมุนจอ การก้าวเดิน)
- Gyroscope ตรวจสอบการหมุนและทิศทาง (เกม AR/VR ระบบกันสั่น)

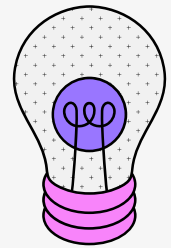
Position

- GPS ระบุตำแหน่งบนพื้นโลก
- Magnetometer ตรวจสอบสนามแม่เหล็ก (เข็มทิศ)

Environment

- Light Sensor ตรวจสอบแสงรอบข้าง (ปรับความสว่างหน้าจอ)
- Barometer ตรวจสอบวัดความกดอากาศ (ระบุความสูง)

Questions & Discussion



"การเข้าใจสถาปัตยกรรม
(Architecture) คือ รากฐานของ
การเขียนแอปที่มีประสิทธิภาพ
(Performance) และประหยัด
พลังงาน (Battery Efficiency)"

