Summary from Guest Speaker

1. **การพิจารณาย้ายจาก Microservices ไป Monolithic**: บริษัทหลายแห่งกำลังย้ายกลับจาก Microservices ไป Monolithic เพื่อลดความซับซ้อนและปรับปรุงกระบวนการในการพัฒนา
2. **ลด Operation Cost ของ Monolithic**: ด้วยการรวมบริการเข้าด้วยกัน บริษัทต่างๆ จะเห็นการประหยัดต้นทุนได้อย่างมากในแง่ของ Structure and Maintain cost
3. **Improved Reliability and Simplicity ของ Monolithic**: Monolithic จัดการได้ง่ายกว่า, ลด Overhead ในการคุยกันระหว่าง service เหมือน Microservice.
4. **การ Scalability ที่ต่างกัน**: Microservice สามารถทำการ Scale บาง Service ได้ ไม่จำเป็นต้อง Scale ทั้งหมด ต่างจาก Monolithic ที่จำเป็นต้อง Scale ทั้งระบบไม่สามารถแยกได้
5. **ความง่ายในการ Deployment ของ Monolithic**: Monolithic สามารถ deply ได้ง่ายกว่า Microservice ที่จำเป็นต้อง deploy แยก service กันเพื่อคงข้อดีของความเป็น Microservice
6. **การจัดทรัพยากรของ Monolithic**: ใน Micrservice อาจมีการใช้หรือเรียกใช้ข้อมูลส่วนเดียวกัน แต่ไม่สามารถใช้ร่วมกันได้ แต่ Monolithic สามารถ share memory และ data ได้
7. **การ Maintenance ของ Monolithic**: การจัดการ, maintain, และการ debug ของ Monolithic ทำได้ง่ายเพราะทำอยู่บนที่เดียว แต่ Microservice มีการแยก service กันทำให้ maintain และ debug ได้ยาก
8. **บริษัทที่ใช้ Monolithic**: เช่น Amazon, Segment, and InVision เนื่องจากต้องการลก overhead และ cost ต่าง ๆ ในการ maintain microservices.
9. **Green Software Principles**: The shift aligns with principles of green software, focusing on energy efficiency, hardware efficiency, and carbon awareness to reduce environmental impact.
10. **Carbon Emissions in Software Development**: Tools like Software Carbon Intensity (SCI) scores help companies measure the carbon footprint of their software and make data-driven adjustments to improve efficiency.