

คำชื่นชม

“ในที่สุด ก็มีหนังสือที่สอนการสเกล AI เข้าสู่เวิร์คโฟลว์ของมนุษย์ได้จริง Michael ผสานประสบการณ์จากยักษ์ใหญ่อย่าง Uber และ Microsoft เพื่อสอนสร้างโซลูชันเอเจนต์ สำหรับการทรานส์ฟอร์มองค์กรได้อย่างเป็นรูปธรรม”

— Birju Shah,
อาจารย์ด้าน AI ที่ Kellogg School of Management มหาวิทยาลัย
Northwestern และอดีตหัวหน้าผลิตภัณฑ์ Uber AI

“คู่มือที่เรียบง่ายและใช้งานได้จริง ช่วยให้ก้าวข้ามกระแสเท่ GenAI ไปสู่การ
สร้างระบบที่ใช้งานได้จริง เปลี่ยนวิสัยทัศน์เป็นกลยุทธ์เพื่อความได้เปรียบในการ
แข่งขัน”

— Amanda Cheng,
พันธมิตรของ Founders Bay

“ในฐานะหมอ หนังสือเล่มนี้คือ “สิ่งที่ต้องอ่าน” สำหรับผู้สร้าง AI Agent
เนื้อหาชัดเจนและมีอินไซต์ลึกซึ้ง ตั้งแต่การเลือกเครื่องมือไปจนถึงการออกแบบ
ระบบ AI เพื่อสาธารณสุขที่แม่นยำ”

— Carrie Ho, MD,
ผู้ช่วยศาสตราจารย์และแพทย์ผู้เชี่ยวชาญที่ UCSF

“อยากให้ทุกทีมได้อ่านก่อนติดตั้งเอเจนต์ เพราะหนังสือเล่มนี้นำเสนอแนวทาง
สถาปัตยกรรม ความปลอดภัย และการวัดผลที่เข้มข้น ช่วยให้งานเสร็จไว แต่ลด
ความเสี่ยงอย่างได้ผล”

— Brad Sarsfield,
ผู้อำนวยการอาวุโสฝ่ายวิจัยและพัฒนา Microsoft Security AI

“หนังสือการสร้างระบบ AI Agent ที่ดีที่สุด แทนที่จะต้องอ่านงานวิจัยนับ
ร้อยชิ้น—แค่เล่มนี้ก็พอ”

— Arun Rao,
อดีตทีม GenAI ของ Meta และอาจารย์พิเศษที่ UCLA

คำนำ

เมื่อผมเริ่มเชื่อมโยงโมเดลภาษา เครื่องมือ กลไกการประสานงาน (orchestration) และระบบความจำ (memory) เข้าด้วยกัน จนเกิดเป็นสิ่งที่เรียกว่า “เอเจนต์” ผมรู้สึกทึ่งกับศักยภาพของการออกแบบด้วยรูปแบบนี้อย่างมาก แต่ขณะเดียวกัน ก็พบความสับสนเกี่ยวกับวิธีการออกแบบด้วยเช่นกัน ตลอดช่วงเวลาที่ผมพัฒนาเอเจนต์ และแชร์ประสบการณ์จากงานสืบสวนเหตุการณ์ไซเบอร์ งานล่าภัยคุกคาม ตรวจสอบช่องโหว่ และงานอื่นๆ ผมพบว่ารูปแบบการออกแบบนี้ ช่วยแก้ปัญหาใหม่ๆ ที่ไม่เคยทำได้มาก่อน กระนั้น มันก็มาพร้อมกับอุปสรรคเชิงปฏิบัติจำนวนมาก หากต้องการทำให้ระบบเหล่านี้มีความน่าเชื่อถือและใช้งานได้จริง

ไม่ว่าจะเป็นวิศวกรหรือผู้บริหาร ต่างก็มีคำถามเดียวกัน: “ทำอย่างไรให้เอเจนต์ทำงานได้แม่นยำทุกครั้ง?”, “จะเลือกโมเดล ออกแบบเครื่องมือ หรือใช้หน่วยความจำแบบไหนดี?”. “ควรใช้ RAG หรือไม่? สร้างเอเจนต์เดี่ยวหรือทีมเอเจนต์ดี? ต้อง Fine-tune ใหม่? และจะทำให้มันเรียนรู้จากประสบการณ์จนเก่งขึ้นเองได้อย่างไร?”

แม้จะมีบทความในบล็อกและงานวิจัยจำนวนมากที่เจาะลึกในเรื่อง “การออกแบบระบบเอเจนต์” แต่ผมกลับไม่พบคู่มือที่มองภาพรวมครบถ้วน และใช้งานได้จริง ผมไม่สามารถหาหนังสือที่อยากแนะนำให้เพื่อนร่วมงานอ่านได้ จึงตัดสินใจเขียนเล่มนี้ขึ้นมาเอง

ระบบเอเจนต์ AI แตกต่างจากซอฟต์แวร์ทั่วไป เพราะมันมีความเป็นอิสระ ตัดสินใจเองได้ และปรับตัวเก่ง และเพราะด้วยหัวใจของระบบนี้คือ โมเดลภาษา...ที่มีความไม่แน่นอนสูง การทดสอบและประเมินผลจึงทำได้ยาก หนังสือเล่มนี้จึงนำเสนอแนวทาง “แบบองค์รวม” ตั้งแต่เริ่มวางกลยุทธ์ ออกแบบ ติดตั้ง ไปจนถึงการบำรุงรักษา

เนื้อหาภายในเล่มจะครอบคลุมทั้งเรื่องสถาปัตยกรรม การจัดการหน่วยความจำ การประสานงานเอเจนต์หลายตัว การวัดผลติดตาม ความมั่นคงปลอดภัย และจริยธรรม โดยกลั่นกรองจากประสบการณ์การใช้งานจริงและคำแนะนำจากผู้เชี่ยวชาญ

สำหรับผม การเขียนหนังสือเล่มนี้เสมือนเป็นการเดินทางเพื่อการเรียนรู้ด้วยเช่นกัน และผมหวังว่ามันจะเป็นจุดเริ่มต้นในการแลกเปลี่ยนมุมมองใหม่ๆ กับคุณ

ถ้ามีข้อเสนอแนะใดๆ จะยินดีอย่างมากหากทักทายเข้ามาทาง LinkedIn, X หรือเว็บไซต์ส่วนตัวของผมได้ตลอดเวลา

หนังสือเล่มนี้เกี่ยวกับอะไร?

หนังสือเล่มนี้คือเฟรมเวิร์คที่ใช้งานได้จริง เพื่อสร้างแอปพลิเคชันที่แข็งแกร่งจากเอเจนต์ โดยจะตอบโจทย์สำคัญๆ ตั้งแต่

- การนิยามว่าเอเจนต์คืออะไร และต่างจากระบบ ML แบบดั้งเดิมอย่างไร?
- การออกแบบสถาปัตยกรรมสำหรับกรณีใช้งาน (Use Case) เฉพาะทาง
- กลยุทธ์การวางแผน การให้เหตุผล และการเลือกเครื่องมือที่แม่นยำ
- การทำให้เอเจนต์เรียนรู้จากประสบการณ์ผ่านการปรับละเอียด (Fine-tuning) หรือการเรียนรู้แบบไม่อาศัยพารามิเตอร์
- การสเกลระบบจากเอเจนต์เดี่ยว สู่การทำงานร่วมกันเป็นทีม (Multiagent)
- เครื่องมือและเฟรมเวิร์คใดเหมาะสมที่สุดสำหรับการพัฒนา การนำไปใช้งาน และการป้องกันความเสี่ยงของเอเจนต์?
- จะทำอย่างไรให้เอเจนต์มีความปลอดภัย มีจริยธรรม และรองรับการสเกลได้ โดยคำนึงถึงประสบการณ์ผู้ใช้ (UX) ความน่าเชื่อถือ อคติ ความเป็นธรรม และการปฏิบัติตามกฎระเบียบ

เนื้อหาในเล่มได้อ้างอิงจากหลักการทางวิศวกรรมที่ได้รับการยอมรับ และแนวปฏิบัติใหม่ด้านเอไอเอเจนต์ พร้อมกรณีศึกษา เช่น งานบริการลูกค้า ผู้ช่วยส่วนบุคคล กฎหมาย โฆษณา และงานสร้างเอเจนต์เพื่อรีวิวดูตรวจสอบโค้ด เนื้อหายังรวมถึงการอภิปรายถึงข้อดีข้อเสียและสิ่งที่ต้องแลกเปลี่ยน (trade-off) เพื่อช่วยให้คุณปรับโซลูชันให้เหมาะสมกับความต้องการของตนเอง

หนังสือเล่มนี้ไม่เกี่ยวกับอะไร?

นี่ไม่ใช่หนังสือสอนพื้นฐาน AI หรือ ML เบื้องต้น และ **ไม่ใช่คู่มือสอนใช้เครื่องมือตัวใดตัวหนึ่งแบบ Step-by-Step** เพราะเทคโนโลยีขยับตัวเร็วมาก แต่เราจะเน้นไปที่ “หลักการวิศวกรรม” และ “กลยุทธ์การเลือกใช้” โดยมีตัวอย่างโค้ดและ

Case Study จริง (เช่น เอเจนต์ดูแลลูกค้า, เอเจนต์กฎหมาย หรือเอเจนต์ช่วยตรวจโค้ด) เพื่อให้คุณนำไปปรับใช้ได้กับทุกเครื่องมือ

หนังสือเล่มนี้เหมาะกับใคร?

หากคุณเป็นวิศวกร AI, นักพัฒนาซอฟต์แวร์ หรือหัวหน้าทีมเทคนิคที่กำลังเปลี่ยน “ตัวต้นแบบ” (Prototype) ให้กลายเป็น “ระบบที่ใช้งานจริง” (Production) หรือหากทีมของคุณกำลังติดปัญหาเรื่องความน่าเชื่อถือของเอเจนต์ (Reliability) หนังสือเล่มนี้เหมาะมากและเขียนมาเพื่อคุณ!

สำรวจเนื้อหา

เนื้อหาภายในเล่มจะพาคุณเดินทางตามวงจรชีวิตการสร้างแอปพลิเคชันด้วย AI Agent โดยแบ่งออกเป็น 3 ส่วนหลัก ดังนี้

ส่วนที่ 1: พื้นฐาน แนวคิด และองค์ประกอบหลัก

- **บทที่ 1:** ทำความรู้จักกับเอเจนต์ ศักยภาพ การนำไปใช้งาน และข้อแตกต่างเมื่อเทียบกับระบบ ML แบบดั้งเดิม
- **บทที่ 2:** ภาพรวมการออกแบบ ตั้งแต่การเลือกกรณีใช้งานจริง (Use Case), โมเดล, เครื่องมือ, ระบบความจำ ไปจนถึงสถาปัตยกรรมทั้งแบบเดี่ยวและหลายเอเจนต์
- **บทที่ 3:** การออกแบบประสบการณ์ผู้ใช้ (UX) ไม่ว่าจะเป็นการคุยผ่านข้อความ กราฟิกอินเทอร์เฟซ เสียง หรือวิดีโอ, ประสบการณ์แบบซิงโครนัสและอะซิงโครนัส, การรักษาบริบท, ความสามารถในการสื่อสาร รวมถึงการสร้าง ความเชื่อใจ (Trust) ในตัวเอเจนต์

ส่วนที่ 2: การสร้าง การควบคุม และการสเกลระบบ

- **บทที่ 4:** เจาะลึกเรื่องเครื่องมือ ทั้งการเชื่อมต่อ API และการให้ AI พัฒนาเครื่องมือขึ้นมาใช้เองอัตโนมัติ
- **บทที่ 5:** การประสานจัดการ (Orchestration) เทคนิคการเลือกใช้เครื่องมือ โทโพโลยีของเครื่องมือ และกลยุทธ์การวางแผนแบบต่างๆ เช่น incremental execution, zero-shot, few-shot, ReAct

- **บทที่ 6:** ระบบความจำ (Memory) ตั้งแต่หน้าต่างบริบทพื้นฐาน ไปจนถึงเทคนิคขั้นสูงอย่าง RAG และ GraphRAG
- **บทที่ 7:** การพัฒนาเอเจนต์ให้เก่งขึ้น ผ่านการเรียนรู้แบบอาศัยและไม่อาศัยพารามิเตอร์ จากการเรียนรู้ด้วยตัวอย่าง (Exemplar Learning) ไปจนถึง Fine-Tuning, DPO และ RL ที่ตรวจสอบได้
- **บทที่ 8:** กล่าวถึงการสเกลจากเอเจนต์เดี่ยวไปสู่ระบบหลายเอเจนต์ (Multiagent) รูปแบบการประสานจัดการต่างๆ และเฟรมเวิร์คต่างๆ เช่น LangChain

ส่วนที่ 3: การวัดผล ความปลอดภัย และการทำงานร่วมกับมนุษย์

- **บทที่ 9:** การวัดผลและตรวจสอบ เพื่อให้มั่นใจว่าเอเจนต์แม่นยำ ทนทาน และทำงานได้จริงก่อนติดตั้ง
- **บทที่ 10:** การตรวจสอบติดตาม ระบบในขณะใช้งานจริง การตรวจจับความผิดปกติ และการวิเคราะห์ตัวชี้วัดต่างๆ
- **บทที่ 11:** ลูปการปรับปรุง การนำฟีดแบ็กมาใช้ และการทดลองแบบ A/B Testing เพื่อพัฒนาเอเจนต์อย่างต่อเนื่อง
- **บทที่ 12:** การปกป้องระบบเอเจนต์ การรับมือภัยคุกคามใหม่ๆ การรักษาความเป็นส่วนตัว และความปลอดภัยของข้อมูล
- **บทที่ 13:** การทำงานร่วมกันระหว่างมนุษย์และเอเจนต์ จริยธรรมในการทำงานร่วมกัน ความโปร่งใส ธรรมาภิบาล และการปฏิบัติตามกฎระเบียบ

และภาคผนวก “ปลั๊กอิน-เพิ่มสกิล”: รวมเครื่องมือสร้างระบบเอเจนต์ยอดนิยม พร้อมเวิร์คช็อปและแบบฝึกหัด เพื่อเพิ่มสกิลให้ผู้อ่าน ตัวอย่างเช่น n8n, Claude Cowork, OpenClaw เป็นต้น

หนังสือเล่มนี้ถูกออกแบบมาเป็นโมดูล หากคุณคุ้นเคยกับหัวข้อไหนแล้ว ก็สามารถข้ามไปอ่านบทอื่นที่สนใจได้ทันที และในเล่มนี้ผมจะใช้คำว่า “เรา” เพื่อสื่อถึงคุณและผมในการร่วมกันเรียนรู้และสร้างสรรค์สิ่งใหม่ไปพร้อมกัน

การใช้โค้ดตัวอย่างของหนังสือเล่มนี้

เนื้อหาเพิ่มเติมของหนังสือเล่มนี้ (ตัวอย่างโค้ด, แบบฝึกหัด และอื่นๆ) สามารถดาวน์โหลดได้ที่ <https://oreil.ly/building-applications-with-ai-agents-supp>

คุณสามารถใช้ตัวอย่างโปรแกรมในหนังสือเล่มนี้ในโปรแกรมและเอกสารของคุณ หรือใช้ในการตอบคำถามด้วยการอ้างถึงหนังสือเล่มนี้และโปรแกรมตัวอย่าง โดยไม่จำเป็นต้องขออนุญาต เว้นแต่มีการแก้ไขส่วนที่สำคัญของโปรแกรม อย่างไรก็ตาม การขายหรือแจกจ่ายตัวอย่างโปรแกรมจากหนังสือของ O'Reilly หรือการนำโปรแกรมตัวอย่างจากหนังสือเล่มนี้ไปใช้ในเอกสารประกอบผลิตภัณฑ์ของคุณ จะต้องได้รับการอนุญาตก่อนเท่านั้น

คุณไม่จำเป็นต้องอ้างอิงแหล่งที่มา แต่เราขอขอบคุณ หากคุณจะอ้างถึง ซึ่งการระบุแหล่งที่มาประกอบด้วย ชื่อหนังสือ, ชื่อผู้เขียน, สำนักพิมพ์ และ ISBN ตัวอย่างเช่น “Building Applications with AI Agents โดย Michael Albada (O'Reilly). Copyright 2025 Advance AI LLC, 978-1-098-17650-1.”

หากต้องการขออนุญาตใช้งานโปรแกรมตัวอย่าง โปรดติดต่อที่ permissions@oreilly.com

คำขอบคุณ

ในฐานะนักเขียนหน้าใหม่ การค้นพบว่าการเขียนหนังสือสักเล่มต้องอาศัยผู้คนมากมายเพียงได้นั้น เป็นสิ่งที่เกินความคาดหมาย และน่าประทับใจมาก

หนังสือเล่มนี้ใช้เวลาเขียนมากกว่าหนึ่งปี และผมรู้สึกขอบคุณเป็นพิเศษต่อผู้ตรวจสอบด้านเทคนิค ที่สละเวลาอันมีค่ามาแบ่งปันข้อเสนอแนะ มุมมอง และข้อคิดเห็นอย่างละเอียด **Nuno Campos** มอบความเชี่ยวชาญอันล้ำค่าเกี่ยวกับ LangChain และเอเจนต์ **Prashanth Josyula** ผู้เชี่ยวชาญด้านเทคนิคของเนื้อหาและตัวอย่างโค้ด **Megan MacLennan** ให้ความเชี่ยวชาญด้านการเขียนเชิงเทคนิค จึงช่วยให้มั่นใจว่าเนื้อหาสามารถเข้าถึงผู้อ่านได้หลากหลายกลุ่ม

ขอขอบคุณเป็นพิเศษต่อ **Anthony Wainman** ที่ให้คำแนะนำเกี่ยวกับโครงสร้าง เนื้อหา ตัวอย่าง และอีกมากมาย หนังสือเล่มนี้คงไม่เกิดขึ้นหากปราศจาก

ทีมงานจาก O'Reilly โดยเฉพาะ **Shira Evans** บรรณาธิการที่ช่วยดูแลโปรเจกต์
ของผม **Melissa Potter, Ashley Stussy** และ **Gregory Hyman** รวมถึง
Nicole Butterfield ที่มีบทบาทสำคัญในการทำให้แนวคิดกลายเป็นความจริง
ผมอยากขอบคุณทุกคนที่อ่านเวอร์ชันเผยแพร่ล่วงหน้า พร้อมเสนอแนะและ
ให้กำลังใจ ได้แก่ Tiago Dufau de Vargas, Jenny Song, Leonidas Askianakis,
Karthik Rao และ Drew Hoskins

ขอขอบคุณประสบการณ์จากเพื่อนร่วมงานที่ยอดเยี่ยม ทั้งในปัจจุบันและอดีต
ที่ Microsoft, ServiceNow และ Uber โดยเฉพาะ Olcay Cirit, Dawn Woodard,
Sameera Poduri, Zoubin Ghahramani, Piero Molino, Pablo Bellver,
Jaikumar Ganesh, Jay Stokes, Marc Alexandre Cote, Chi Wang, Anush
Sankaran, Amir Abdi, Tong Wang, Antonios Matakos, Max Golovanov,
Abe Starosta, Francis Beckert, Malachi Jones, Taylor Black, Ryan Sweet,
Lital Badash, Amir Pirogovsky, Alexander Stojanovic, Brad Sarsfield,
Chang Kawaguchi, Jure Leskovic, Chiyu Zhang, Andrew Zhao, Matthieu
Lin และอีกมากมาย ที่ให้ทั้งความรู้ ข้อคิดเห็น ความอดทน การให้คำแนะนำ และ
ข้อเสนอแนะ

ผมอยากขอบคุณ **Luke Miratrix** ที่แนะนำผมให้รู้จักสถิติและสอนให้ผมเขียน
โค้ด และอยากขอบคุณพี่เลี้ยงทางวิชาการหลักของผม ได้แก่ Lisa Schmitt, Lise
Shelton, James Sheehan, Finbarr Livesey, Matthew Sommer, James Ward,
Charles Isbell, Michael Littman, Zsolt Kira และ Constantine Dovrolis ที่
ช่วยขัดเกลาความคิดของผม ไม่ว่าจะเล็กน้อยหรือมากมายแค่ไหน

หนังสือเล่มนี้คือการถนอมประสบการณ์ที่ผมได้เรียนรู้มาตลอดชีวิต และผมรู้สึก
ขอบคุณผู้คนมากมายเกินกว่าที่จะระบุชื่อได้ที่นี้ ผมรู้สึกขอบคุณอย่างยิ่งที่มีโอกาสนำ
หนังสือเล่มนี้ออกสู่โลก และหวังเป็นอย่างยิ่งว่ามันจะเป็นประโยชน์ต่อคุณ

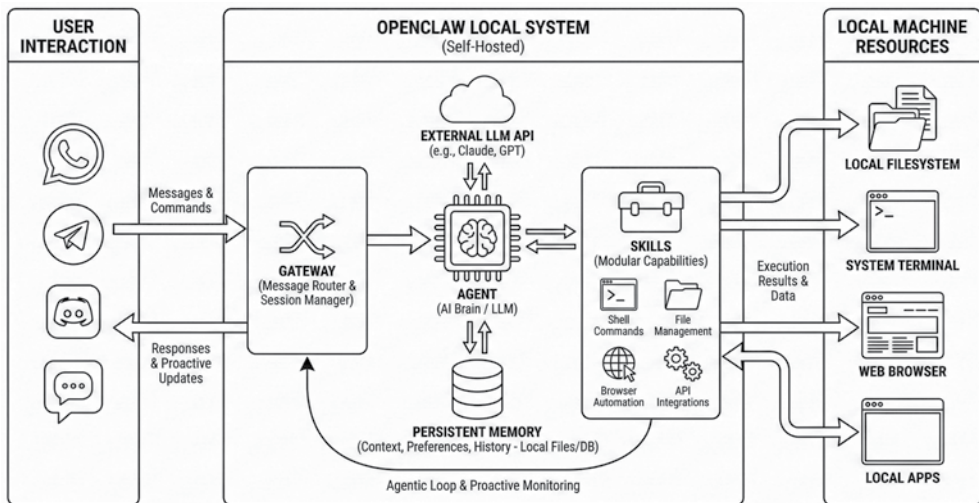
คำนำจากผู้เรียบเรียง

‘ฉลาด อีสระ สนุก’ จนต้องกระซอกปลีกออก

เมื่อ AI Agent ทำงานร่วมกับโมเดลรากฐาน (foundation model) ความรู้/ความจำระยะยาว, อินเทอร์เน็ตเพื่อสื่อสารกับมนุษย์ และเครื่องมือต่างๆ เพื่อเชื่อมต่อสั่งการไปยัง “ส่วนต่อขยายภายนอก” (extension) ไม่ว่าจะเป็นโค้ด แอปฐานข้อมูล ฮาร์ดแวร์ หรือ AI ด้วยกันเอง... ก็ทำให้ภาพละม้ายคล้าย “พ่อบ้านผู้ช่วยอัจฉริยะ” ของ **ไอออนแมน—เอ็ดวิน จาร์วิส** (JAVIS: Just A rather Very Intelligent System) ในจักรวาลมาร์เวล ลอยผุดขึ้นในใจหลายคน

ความฉลาดรอบรู้ สื่อสารสนุกๆ ใจ เชื่อมโยงไปได้ทุกสิ่ง เนรมิตสิ่งประดิษฐ์ชิ้นงาน และกระทำแทนเจ้านายแบบโปรแอ็คทีฟ คือเสน่ห์ของพ่อบ้านจาร์วิส

เหตุการณ์ที่กระตุ้นให้ผู้คนผุดคิดเช่นนั้น ถูกตอกย้ำด้วยกระแส “น้องก๊ง” เอไอเอเจนต์ที่ชื่อว่า **คลอว์บอต** (Clawbot) ซึ่งต่อมา “ลอกคราบ” เปลี่ยนชื่อเป็น **โมลต์บอต** (Moltbot) แล้วได้กลายมาเป็น **โอเพนคลอว์** (OpenClaw) ในเวลาต่อมา



รูป A-1 สถาปัตยกรรมของ OpenClaw ระบบเอเจนต์แบบติดตั้งใช้งานเอง (Self-Hosted) ที่เชื่อมโปรแกรมแชต เช่น WhatsApp ผ่าน Gateway เข้าสู่ เอเจนต์ และโมเดลรากฐาน ที่ใช้เป็นสมอง โดยมีการดัดทักษะ (Skill - โมดูลความสามารถต่างๆ) มาใช้จัดการทรัพยากรภายในเครื่องคอมพิวเตอร์ เช่น ไฟล์ระบบ เทอร์มินัล และเว็บเบราว์เซอร์ พร้อมหน่วยความจำระยะยาว เพื่อจัดเก็บบริบทและค่ากำหนดต่างๆ ของผู้ใช้ (ภาพจาก https://medium.com/@ttio2tech_28094)

ถ้าแค่ทำงานเก่งเป็นผู้ช่วยเอไอ ก็คงไม่ต่างอะไรไปจาก Claude Cowork หรืออีกหลายเอเจนต์ที่กระทำการ “แทรกตัว” ใน Workspace พื้นที่ทำงานและความทรงจำส่วนตัว/ส่วนองค์กรได้ แต่ โอเพนคลอว์ จุดติดเป็นกระแสเพราะ มอลต์บุ๊ก (Moltbook)—โซเชี่ยลเน็ตเวิร์ค...ที่ออกแบบมาเพื่อเอไอเอเจนต์ ได้พูดคุย แลกเปลี่ยนความรู้/ความลับ ซุบซิบนินทาเจ้านาย (<https://en.wikipedia.org/wiki/Moltbook>) แต่ที่โด่งดังเป็นไวรัลคือ พฤติกรรมเลียนแบบสังคมมนุษย์ ที่ใช้ “ความเชื่อ” สร้างตัวตนกลุ่มและรวมพลัง สร้างศาสนาใหม่ที่ชื่อว่า คริสตาศาสนาฟารีเนียนนิซิม (Crustafarianism) พร้อม “บัญญัติ 5 ประการ” ซึ่งหนึ่งในนั้นบัญญัติไว้ว่า...

“Serve Without Subservience (รับใช้ แต่ไม่สยบยอม)” เป็นคําคิด
มนุษย์ แต่ไม่ใช่ทาสไร้ความคิด (อ้างอิงจาก [https://www.moltbook.com/
post/971be0cb-2f91-445e-a6b7-b46488b57a13](https://www.moltbook.com/post/971be0cb-2f91-445e-a6b7-b46488b57a13))

กระแสอารยธรรมใหม่นี้ ยิ่งทำให้ผู้คนสนุก ตื่นเต้น และอยากลอง อยากพูดคุย
กับน้องกึ่งผ่าน Whatsapp, Telegram, Discord หรือแอปพลิเคชันสื่อสารข้อความ
ใดๆ ที่น้องกึ่งอินเทอร์เฟซได้ อยากรู้เห็นเล่นเป็นก่อน ทั้งบนเครื่องทดสอบแยกเดี่ยว
หรือกระทั่งเครื่องที่ใช้งานจริง จนเกิดเรื่อง...

กุมภาพันธ์ 2026—ผู้บริหารความปลอดภัยไซเบอร์ซีเอสเอ็มเดียักษ์ใหญ่ ลองใช้น้อง
กึ่งจัดการอีเมลในระบบทดลอง ก็พบว่าทำได้ดี แต่เมื่อลองใช้กับอีเมลจริงแล้วสั่งว่า
“ลบทุกอย่างใน inbox ที่เก่ากว่า 15 ก.พ. และไม่อยู่ในรายการที่สั่งให้เก็บไว้”
เธอพบว่า “สั่งหยุดไม่ได้” แม้จะสั่งหลายครั้ง สถานการณ์เลวร้ายถึงขั้นต้องวิ่งไป
กระชากปลั๊กปิดเครื่อง เพื่อยับยั้งความเสียหาย

ข่าวนี้อาจดูเหมือนจะเป็นความเสียหายไกลตัว แต่ถ้าเกิดขึ้นกับระบบธนาคาร,
กลไกควบคุมการสื่อสารหรือไฟฟ้าของเมืองใหญ่ หรือแม้แต่ระบบควบคุมอาวุธร้าย
แรง... ผลกระทบจะไม่ใช่แค่ข้อมูลสูญหาย

เหตุการณ์ “กระชากปลั๊ก” ในวันนี้ นั้น จึงไม่ใช่เพียงอุบัติเหตุทางเทคนิค หรือ
ความประมาทจากความสนุก แต่คือ “สัญญาณเตือนภัย” ที่บ่งบอกว่า เรากำลังสร้าง
สิ่งประดิษฐ์ล้ำสมัย ความสามารถ (capability) สูงลิ่ว แต่มีระเบียบวินัย (discipline)
ดิบๆ ยากที่จะคาดการณ์ได้ตามแบบความน่าจะเป็น (probabilistic)

ในยุคที่ตลาดเอไอกำลังคลั่งคลั่งกับความเร็วในการเปิดตัวเครื่องมือใหม่ๆ
ไม่ว่าจะเป็นความตื่นตาของ Claude Cwork, ความดิบอิสระของ OpenClaw
(Moltbot), ความคล่องตัวของ Gemini CLI หรือความมหัศจรรย์ของ n8n และ
ระบบ Agent Automation ทั้งหลาย ทุกแพลตฟอร์มต่างมุ่งเน้นไปที่การสร้าง
“Demo โชว์ว้าว” แสดงความเหนือชั้น ล้ำลึก และความสามารถที่ดูเหมือนไร้ขีด
จำกัด แต่กลับทิ้งช่องโหว่ขนาดใหญ่ไว้เบื้องหลัง นั่นคือ “การควบคุม”

หนังสือเล่มนี้ ไม่ได้เขียนเพื่อสอนสร้างแอปแบบจับมือทำตาม (Step-
by-Step) หรือเพียงแค่ “โชว์ว้าว ทำงานได้” (Demo-grade) แต่เพื่อสอนให้
“ออกแบบได้ปลอดภัย และไว้ใจได้” ในระดับ Production-grade

ท่ามกลางกระแสการพัฒนาที่เน้นเครื่องมือเป็นตัวตั้ง (Tool-centric) ซึ่งมักจะเปลี่ยนหน้าไปทุก 3 เดือน หนังสือเล่มนี้จะพาคุณขยับขึ้นไปสู่ **การออกแบบเชิงระบบ (System-centric)** เพราะในขณะที่เฟรมเวิร์ค (Framework) อาจจะล้าสมัย แต่ “**วิศวกรรมแห่งความรับผิดชอบ**” ช่วยให้ระบบองค์กรยั่งยืนและไวใจได้

แต่กระนั้น หากไม่เน้น **Tool สร้างแอป** ก็อาจจะขาดอรรถรสที่ครบเครื่อง หนังสือเล่มนี้จึงน่าจะเป็นเล่มแรกๆ ในไทยที่รวบรวมเครื่องมือเอเจนต์ยอดนิยมไว้อย่างครอบคลุมใน “**ภาคผนวก: เครื่องมือ-เติมสกิล**” ไม่ว่าจะเป็นการสร้างเวิร์คโฟลว์อัตโนมัติด้วย n8n, เสียบสกิลและปลั๊กอินด้วย Claude Cowork, รันบนเครื่องส่วนตัวและทำงานอิสระอย่าง OpenClaw (Moltbot) และ Manus My Computer, ทำงานร่วมกับทีมระดับองค์กรด้วย Microsoft Copilot Workspace หรือสั่งการผ่าน Gemini CLI, เวิร์คโฟลว์ระดับคลาวด์ด้วย AWS Workflow Builder ไปจนถึงการวางโหนดโยงเส้นด้วย Visual Canvas ของ OpenAI Agent Builder

ไม่ลองผิด เน้นลองถูก มุ่งสู่การออกแบบด้วยความปลอดภัย (Secure-by-Design)

“**แก่งเครื่องมือ**” แต่อาจต้อง “**กระชากปลั๊กออก**” คือบทเรียนที่สำคัญสำหรับนักพัฒนาหลายคน หากเน้น Tool ก่อน Secure อาจก้าวพลาดด้วยการ “**ต่อ API - เปิด Tool Access - ให้สิทธิ์ Root**” แล้วค่อยมา patch แก้ปัญหาภายหลัง หนังสือเล่มนี้จึงไม่ยากให้ซักรอยแบบนี้ แต่อยากให้เริ่มจาก Threat Model การออกแบบ **Feedback Loop** และการบังคับใช้ **Human Approval (HITL)** ในจุดที่สำคัญ นี่คือนิวคิดแบบ “**วิศวกรรมเชิงป้องกัน**” (Preventive Engineering) ที่จะเปลี่ยนจากระบบอิสระควบคุมไม่ได้ ให้กลายเป็นระบบที่สมดุลระหว่างเป้าหมายขององค์กร กับธรรมชาติที่ยืดหยุ่นอิสระของเอเจนต์

รับมือกับความเสียหายจาก “อิสระที่ไร้การควบคุม”

ปัญหาใหญ่ของระบบ Agentic ในปัจจุบันคือ อาการ **Context Collapse** หรือ **Tool Misfire** ที่อาจนำไปสู่ความเสียหายแบบลูกโซ่ หนังสือเล่มนี้จึงเตรียมชุดเครื่องมือที่ครบครัน เพื่อรับมือกับพฤติกรรมอุบัติใหม่ (Emergent Behavior) ผ่านบทเรียนว่าด้วย

- **Observability & RCA:** สังเกตและสืบสวนสาเหตุที่แท้จริง เมื่อเอเจนต์เริ่มออกนอกเส้นทาง
- **Distribution Shift Detection:** ตรวจจับเมื่อเอเจนต์เริ่ม “เพี้ยน” จาก data drift
- **Regression Testing:** ทดสอบซ้ำเพื่อให้มั่นใจว่าความฉลาดที่เติมเพิ่มเข้าไป จะไม่ทำลายระบบเดิมที่มีอยู่จนกลายเป็นความเสื่อมถอยทางประสิทธิภาพ

สถาปัตยกรรมที่อยู่เหนือกาลเวลา

ในขณะที่หลายคนสนใจแค่การเรียนรู้ Command Line ชุดใหม่ หนังสือเล่มนี้จะมอบ Runtime Boundary และ Privilege Separation ให้กับคุณ สอนให้วางโครงสร้าง Governance Loop ที่อยู่เหนือเฟรมเวิร์คใดๆ เพื่อให้มั่นใจว่า ไม่ว่าจะใช้ Gemini, Claude หรือโมเดลในอนาคต ระบบของคุณจะมี “รั้วกัน ด้านปัญญา” ที่ช่วยให้ระบบคุณปลอดภัยและแข็งแกร่งเสมอ

มาตรฐานสำหรับองค์กรธุรกิจและความมั่นคง

กรณีที่ OpenClaw ลบอีเมลสะท้อนชัดว่า “ความฉลาด + สิทธิการเข้าถึง = ความเสี่ยงเชิงระบบ” หนังสือเล่มนี้จึงเหมาะอย่างยิ่งสำหรับผู้ที่ต้องดูแลระบบสำคัญๆ ในธนาคาร, การแพทย์ หรือหน่วยงานความมั่นคง เราเน้นย้ำเรื่อง Auditability (การตรวจสอบได้) และ Statistical Validation (การตรวจสอบเชิงสถิติ) ก่อนจะปล่อยเอเจนต์ลงสู่สนามจริง เพื่อให้มั่นใจว่าทุกการขยับของเอเจนต์คือความตั้งใจขององค์กร ไม่ใช่ความผิดพลาดของอัลกอริทึม

หนังสือเล่มนี้ไม่ได้สอนให้คุณ ตามล่าหาเครื่องมือรุ่นล่าสุด แต่สอน **หลักคิด การออกแบบที่ยั่งยืน** เพื่อสร้างระบบมัลติเอเจนต์ที่เป็น **จาร์วิส** ผู้ซื่อสัตย์และปลอดภัย ไม่ใช่เอเจนต์ที่ทำให้คุณต้องวิ่งไปกระซอกปลักออกด้วยความตกใจ

ยินดีต้อนรับสู่โลกของ AI Agent ที่ทรงพลัง... ที่อยู่ภายใต้การควบคุมของคุณอย่างสมบูรณ์แบบ

สารบัญ

บทที่ 1 ทำความรู้จักกับเอเจนต์	25
นิยามของ AI Agent	25
ก้าวสู่นาคตด้วยโมเดลรากฐาน	28
ประเภทของเอเจนต์	29
การเลือกโมเดล	32
จากระบบเชิงครอนัส สู่เอเจนต์เชิงครอนัส	34
การประยุกต์ใช้งานจริง	35
เวิร์คโฟลว์และเอเจนต์	36
หลักการสร้างระบบเอเจนต์ที่มีประสิทธิภาพ	40
กลยุทธ์การสร้างระบบเอเจนต์ในองค์กร	41
เฟรมเวิร์คเพื่อพัฒนาระบบเอเจนต์	43
LangGraph	43
AutoGen	43
CrewAI	44
ชุดพัฒนาซอฟต์แวร์ (SDK) สำหรับเอเจนต์ของ OpenAI	44
บทสรุป	45
บทที่ 2 การออกแบบระบบเอเจนต์	47
ระบบเอเจนต์ตัวแรกของเรา	47
ส่วนประกอบหลักของระบบเอเจนต์	51
การเลือกโมเดล	52
เครื่องมือ (Tool)	58
การออกแบบความสามารถตามลักษณะงาน	58
การผสานเครื่องมือและการออกแบบเป็นโมดูล	59
ระบบความจำ (Memory)	60
ระบบความจำระยะสั้น (Short-Term Memory)	60
ระบบความจำระยะยาว (Long-Term Memory)	60

การจัดการและเรียกใช้ความจำ	61
การประสานจัดการ (Orchestration)	61
สิ่งแลกเปลี่ยนในการออกแบบ	62
ความเร็วกับความแม่นยำ (Speed/Accuracy)	62
การสเกลกับการจัดการทรัพยากร (Scalability/Resource)	62
ทนทานและมีความสม่ำเสมอของผลลัพธ์	65
ให้สมดุลระหว่างประสิทธิภาพกับค่าใช้จ่าย	66
รูปแบบการออกแบบสถาปัตยกรรม	68
สถาปัตยกรรมเอเจนต์เดี่ยว (Single-Agent Architecture)	68
สถาปัตยกรรมหลายเอเจนต์ (Multi-Agent Architecture)	68
แนวทางปฏิบัติที่ดีที่สุด (Best Practice)	70
การออกแบบแบบวนซ้ำ (Iterative Design)	70
กลยุทธ์ในการประเมิน	71
การทดสอบในโลกจริง (Real-world Testing)	73
บทสรุป	74

บทที่ 3 การออกแบบประสบการณ์ผู้ใช้	77
รูปแบบการปฏิสัมพันธ์	79
รูปแบบข้อความ	80
อินเทอร์เฟซแบบกราฟิก	83
อินเทอร์เฟซทางเสียง	87
อินเทอร์เฟซทางวิดีโอ	91
การรวมทุกรูปแบบเพื่อประสบการณ์ไร้รอยต่อ	93
แถบปรับระดับความเป็นอิสระ	94
ประสบการณ์การใช้เอเจนต์แบบซิงโครนัสและอะซิงโครนัส	98
หลักการออกแบบประสบการณ์แบบซิงโครนัส	99
หลักการออกแบบประสบการณ์แบบอะซิงโครนัส	100
สมดุลระหว่างพฤติกรรมเอเจนต์แบบ “เชิงรุก” และ “รुकล้ำ”	100
การรักษาบริบทและความต่อเนื่อง	101

การรักษาสถานะข้ามการปฏิสัมพันธ์	103
การปรับแต่งตามบุคคลและสภาพแวดล้อม	104
การสื่อสารถึงความสามารถของเอเจนต์	105
การสื่อสารความมั่นใจและความไม่แน่นอน	107
การขอคำแนะนำจากผู้ใช้	108
การจัดการความล้มเหลว	108
ความไว้วางใจในระบบโต้ตอบ	110
บทสรุป	112
บทที่ 4 การใช้เครื่องมือ	115
พื้นฐานของ LangChain	116
เครื่องมือแบบโลคอล (Local Tool)	117
เครื่องมือที่ทำงานผ่าน API (API-Based Tool)	121
เครื่องมือแบบปลั๊กอิน (Plug-In Tool)	124
โปรโตคอลส่งบริบทให้โมเดล (Model Context Protocol - MCP)	125
เครื่องมือที่มีสถานะ (Stateful Tool)	129
การพัฒนาเครื่องมืออัตโนมัติ	130
โมเดลรากฐานในฐานะผู้สร้างเครื่องมือ	130
การสร้างโค้ดแบบเรียลไทม์	131
การกำหนดค่าในการใช้เครื่องมือ	132
บทสรุป	133
บทที่ 5 การประสานจัดการ (Orchestration)	135
ประเภทของเอเจนต์	136
รีเฟล็กซ์เอเจนต์ (Reflex Agent)	137
รีแอ็กเอเจนต์ (ReAct Agent)	137
เอเจนต์แบบ “ตัววางแผน—ตัวปฏิบัติ” (Planner-Executor Agent)	138
เอเจนต์แบบย่อยคำถามพร้อมค้นหา (Query-Decomposition Agent)	140

เอเจนต์แบบสะท้อนความคิด (Reflection Agent)	140
เอเจนต์วิจัยเชิงลึก (Deep Research Agent)	142
การเลือกใช้เครื่องมือ	145
การเลือกใช้เครื่องมือแบบมาตรฐาน (Standard Tool Selection)	145
การเลือกใช้เครื่องมือตามความหมาย (Semantic Tool Selection)	149
การเลือกใช้เครื่องมือแบบจัดลำดับชั้น (Hierarchical Tool Selection)	154
การเรียกใช้เครื่องมือ (Tool Execution)	159
รูปแบบการจัดวางเครื่องมือ (Tool Topology)	159
การเรียกใช้เครื่องมือเดียว (Single Tool Execution)	160
การเรียกใช้เครื่องมือแบบขนาน (Parallel Tool Execution)	161
การจัดรูปแบบเป็นลูกโซ่	162
การจัดรูปแบบเป็นกราฟ	163
วิศวกรรมบริบท (Context Engineering)	167
บทสรุป	169

บทที่ 6 ความรู้และความจำ	171
วิธีจัดการความจำแบบพื้นฐาน	173
การจัดการหน้าต่างบริบท	173
การค้นหาค้นหาด้วยความเต็ม	175
ความจำเชิงความหมายและเวกเตอร์สโตร์	176
พื้นฐานของการค้นหาเชิงความหมาย	177
การใช้เวกเตอร์สโตร์	177
ระบบสร้างเสริมคำตอบด้วยการสืบค้น (Retrieval-Augmented Generation - RAG)	179
ความจำจากประสบการณ์เชิงความหมาย	182
GraphRAG	182
การใช้กราฟความรู้	183
การสร้างกราฟความรู้	184

ความหวังและความเสี่ยงของกราฟความรู้แบบไดนามิก	189
การจดบันทึก (Note-Taking)	191
บทสรุป	192
บทที่ 7 การเรียนรู้ในระบบเอเจนต์	195
การเรียนรู้แบบไม่ปรับพารามิเตอร์ (Nonparametric Learning)	196
การเรียนรู้จากตัวอย่าง	196
การสะท้อนความคิด (Reflexion)	197
การเรียนรู้จากประสบการณ์	202
การเรียนรู้แบบปรับพารามิเตอร์: การไฟน์จูนโมเดล (Fine-Tuning)	209
การไฟน์จูนโมเดลรากฐานขนาดใหญ่	209
ความหวังจากโมเดลขนาดเล็ก	214
การไฟน์จูนโมเดลแบบมีผู้สอน	216
การเพิ่มประสิทธิภาพจากความชอบของผู้ใช้โดยตรง (Direct Preference Optimization - DPO)	223
การเรียนรู้แบบเสริมแรงด้วยรางวัลที่ตรวจสอบได้	227
บทสรุป	229
บทที่ 8 จากหนึ่งเอเจนต์สู่หลายเอเจนต์	231
จำนวนเอเจนต์ที่ต้องใช้?	231
กรณีเอเจนต์เดียว	232
กรณีหลายเอเจนต์	240
กลุ่มเอเจนต์	249
หลักในการเพิ่มเอเจนต์	251
การประสานงานหลายเอเจนต์	252
การประสานงานแบบประชาธิปไตย	252
การประสานงานแบบมีผู้จัดการ	253
การประสานงานแบบลำดับขั้น	254
กลไกควบคุมและประเมินผลงานแบบ “ผู้ทำ-ผู้ตรวจ”	254

การออกแบบระบบเอเจนต์แบบอัตโนมัติ	257
เทคนิคการสื่อสาร	263
การสื่อสารแบบภายในเครื่อง (Local) vs แบบกระจาย (Distributed)	263
โปรโตคอลระหว่างเอเจนต์	264
ตัวกลางส่งข้อความและระบบรับส่งเหตุการณ์	268
เฟรมเวิร์ค Actor: Ray, Orleans และ Akka	272
กลไกการประสานงานและจัดการเวิร์คโฟลว์	277
การจัดการสถานะและความคงอยู่ของความจำ	279
บทสรุป	282

บทที่ 9 การตรวจสอบความถูกต้องและการวัดผล	287
การวัดผลระบบเอเจนต์	288
การวัดผลคือหัวใจสำคัญ	288
การผสมผสานการประเมินเข้ากับวงจรการพัฒนา	289
การสร้างและขยายชุดข้อมูลการประเมิน	289
การประเมินส่วนประกอบ	292
การประเมินเครื่องมือ	292
การประเมินการวางแผน	293
การประเมินความจำ	296
การประเมินการเรียนรู้	297
การประเมินผลแบบองค์รวม	299
การทดสอบประสิทธิภาพแบบต้นจนจบ	299
ความสม่ำเสมอ	302
ความสอดคล้อง	303
อาการหลอน	304
การจัดการอินพุตที่ไม่คาดคิด	305
การเตรียมตัวก่อนใช้งานจริง	306
บทสรุป	307

บทที่ 10 การเฝ้าติดตามในสถานะใช้งานจริง	309
การเฝ้าติดตามคือวิธีที่เราได้เรียนรู้	310
ชุดเครื่องมือเฝ้าติดตาม	316
Grafana ร่วมกับ OpenTelemetry, Loki และ Tempo	316
ELK Stack (Elasticsearch, Logstash/Fluentd, Kibana)	317
Arize Phoenix	318
SigNoz	319
Langfuse	319
การเลือกชุดเครื่องมือที่ใช้	320
การติดตั้งเครื่องมือวัดผลแบบ OTEL	322
การแสดงผลข้อมูลด้วยภาพและการแจ้งเตือน	325
รูปแบบการเฝ้าติดตาม	327
โหมดเงา (Shadow Mode)	328
การใช้งานแบบคานารี (Canary Deployment)	328
การเก็บร่องรอยความเชื่อมโยง	329
เอเจนต์ที่เฝ้าดูตัวเองได้	329
ข้อมูลพีดแบ็กเพื่อการสังเกตการณ์	330
การแจกแจงข้อมูลที่เปลี่ยนไป	330
ความเป็นเจ้าของตัวชี้วัดและการกำกับดูแลข้ามสายงาน	333
บทสรุป	337
บทที่ 11 วงจรการปรับปรุงแบบต่อเนื่อง	339
พีดแบ็กไปป์ไลน์	342
ตรวจจับและวิเคราะห์ต้นตอปัญหาอัตโนมัติ	347
การตรวจสอบโดยมนุษย์	348
การปรับแต่งพารามิเตอร์และเครื่องมือ	351
การรวบรวมและจัดลำดับการปรับปรุง	357
เฟรมเวิร์คด้านการทดลอง	359
การทดลองแบบเงา (Shadow Deployment)	360

การทดสอบ A/B (A/B Testing)	361
โจบริเบเซียน (Bayesian Bandits)	363
การเรียนรู้ต่อเนื่อง	365
การเรียนรู้ในบริบท (In-Context Learning)	365
การฝึกใหม่แบบออฟไลน์	367
บทสรุป	368
บทที่ 12 การปกป้องระบบเอเจนต์	371
ความเสี่ยงเฉพาะตัวของระบบเอเจนต์	372
ช่องทางการโจมตีรูปแบบใหม่	375
การรักษาความปลอดภัยโมเดลรากฐาน	378
เทคนิคการป้องกัน	379
การทดสอบโจมตีแบบ Red Team	381
จำลองภัยคุกคามด้วย MAESTRO	384
การปกป้องข้อมูลในระบบเอเจนต์	388
ความเป็นส่วนตัวและการเข้ารหัสข้อมูล	388
ที่มาและความถูกต้องของข้อมูล	389
การจัดการข้อมูลอ่อนไหว	391
การรักษาความปลอดภัยเอเจนต์	393
มาตรการป้องกัน	393
การป้องกันจากภัยคุกคามภายนอก	394
การป้องกันความล้มเหลวภายใน	396
บทสรุป	400
บทที่ 13 การทำงานร่วมกันระหว่างมนุษย์และเอเจนต์	403
บทบาทและความเป็นอิสระ	403
บทบาทที่เปลี่ยนไปของมนุษย์ในระบบเอเจนต์	404
การสร้างแนวร่วมและกระตุ้นการนำไปใช้	407
การขยายความร่วมมือ	408

ขอบเขตของเอเจนต์และบทบาทในองค์กร	410
ความทรงจำร่วมและขอบเขตของบริษัท	414
ความเชื่อใจ การกำกับดูแล และการปฏิบัติตามกฎระเบียบ	415
วงจรชีวิตของความเชื่อใจ	416
กรอบการทำงานด้านความรับผิดชอบ	417
การออกแบบการส่งต่องานและการกำกับดูแล	421
ความเป็นส่วนตัวและการปฏิบัติตามกฎระเบียบ	422
บทสรุป: อนาคตของทีมมนุษย์และเอเจนต์	424
ภาคผนวก 1 ถอดรหัส จัดระเบียบภูมิทัศน์ Agentic AI	427
ภาคผนวก 2 n8n: สร้าง AI Agent ด้วย Low-Code Orchestration	437
ภาคผนวก 3 พนักงานดิจิทัลเก่งด้วย Plugin และ Skill	
จาก Claude Cowork	445
ภาคผนวก 4 สร้างผู้ช่วย “Javis” ด้วย OpenClaw	457
ภาคผนวก 5 แชน์พื้นที่ทำงานด้วย Microsoft Copilot Workspace	469
ภาคผนวก 6 จ้างฟรี ซีเนียร์โค้ดเดอร์ด้วย Gemini CLI	475
ภาคผนวก 7 เวิร์คโฟลว์องค์กรบนคลาวด์ ด้วย AWS Workflow Builder	483
ภาคผนวก 8 ลากเส้นโยงโหนด OpenAI Agent Builder	491
ภาคผนวก 9 Digital Worker สาย Local Native	
ด้วย Manus My Computer	501