สรุปวัตถุประสงค์ของโปรเจกต์นี้

เรากำลังหาอะไร (What)

หาว่า "เครือข่ายที่ไม่สมบูรณ์แบบ" (network impairment) ส่งผลกระทบต่อการทำงานร่วมกันของ AI หลายตัว (multi-agent LLM) มากแค่ไหน และแบบไหน
พูดง่ายๆ คือทดลองจำลองสถานการณ์ "วิทยุสื่อสารมีปัญหา" (delay, packet loss, jitter, bandwidth จำกัด) แล้วดูว่าทีม AI (Planner → Worker → Reviewer) ที่คุยกันผ่านเครือข่ายนั้น จะ:

- ทำงานสำเร็จได้ไหม (success rate)
- ช้าลงแค่ไหน (latency)
- ต้องพูดซ้ำ/retry บ่อยแค่ไหน (retry/timeout)
- เสีย token/ค่าใช้จ่ายเพิ่มขึ้นแค่ไหน (cost)
- มีจุดวิกฤต **(threshold)** ที่ระบบเริ่มพังกะทันหันตรงไหน

เพื่ออะไร (Why / So what)

เพื่อตอบคำถามที่ยังไม่มีใครตอบชัดเจน: "ถ้าจะเอา multi-agent LLM ไปใช้งานจริงในสภาพแวดล้อมที่เครือข่ายไม่สมบูรณ์แบบ (เช่น edge device, cross-region cloud, พื้นที่ signal ไม่ดี) — มันจะยังทำงานได้ดีไหม หรือพังตรงไหน?"
เพราะตอนนี้:

- คนส่วนใหญ่ทดสอบ/deploy multi-agent LLM บน cloud เดียวกันหมด (สมมติว่า network เร็วสมบูรณ์แบบเสมอ)
- แต่ในความเป็นจริง ถ้าจะขยายไปใช้งานแบบ distributed หรือ edge computing เครือข่ายจะไม่มีทางสมบูรณ์แบบ
- **ยังไม่มีตัวเลข/ข้อมูลอ้างอิงที่ชัดเจน** ว่าเครือข่ายแย่ระดับไหนที่ยังรับได้ กับระดับไหนที่ระบบเริ่มพัง

ผลลัพธ์สุดท้ายที่อยากได้ (Deliverable)

1. **ตัวเลข/threshold ที่ชัดเจน** เช่น "ถ้า delay เกิน 500ms หรือ packet loss เกิน 10% ระบบ multi-agent เริ่มพังแบบเห็นได้ชัด"
2. **คำแนะนำเชิงปฏิบัติ (Practical Recommendation)** ที่คนทำงานจริงเอาไปใช้ตัดสินใจได้ เช่น วิศวกรที่จะ deploy ระบบควรวาง infrastructure แบบไหน

สรุปเป็นประโยคเดียว

"เราอยากรู้ว่า AI หลายตัวที่ทำงานร่วมกัน จะทนทานต่อเครือข่ายที่ไม่สมบูรณ์แบบได้แค่ไหน และอยากทำตารางคำแนะนำที่บอกคนหน้างานว่า ควรวางระบบแบบไหนให้รอดพ้นปัญหานี้"
ถ้าตรงนี้ตรงกับที่คุณคิดไว้ในใจ เราสามารถต่อยอดไปเขียนเป็น **Research Question / Objective section** ของ paper ได้เลยครับ อยากให้ช่วยร่างเป็นข้อความทางการ (formal) สำหรับใส่ใน paper ไหม?

UseCase

Multi-region cloud Deployment
สภาพแวดล้อม: บริษัทที่กระจาย agent ข้าม cloud region เพื่อลดต้นทุน GPU หรือทำตาม data compliance (เช่น agent วิเคราะห์อยู่ US แต่ agent เก็บข้อมูลอยู่ Asia)
ทำไมต้องใช้: ไม่ได้เกี่ยวกับ "สัญญาณแย่" แบบกายภาพ แต่เป็น **network latency ข้าม region ที่หลีกเลี่ยงไม่ได้** (เช่น US↔Asia อาจ 150-300ms) — นี่คือ use case ที่ mainstream ที่สุดและใช้ได้จริงเร็วที่สุด

ปัญหาของ "solution ที่มีอยู่แล้ว" คือมันเป็นแค่ Best Practice ไม่ใช่

Evidence

สังเกตว่า Async Execution และ State Hydration/Caching เป็น**สถาปัตยกรรมทั่วไป** ที่ใช้แก้ latency ได้กับระบบ distributed ทุกชนิด (ไม่ใช่แค่ LLM agent) — เป็นความรู้ทางวิศวกรรมที่มีมาก่อน LLM agent จะเกิดขึ้นด้วยซ้ำ (Kafka, SQS, caching มีมาเป็นสิบปีแล้ว)

คำถามที่ solution เหล่านี้ตอบไม่ได้:

- ควร cache แค่ไหน? cache ผิดพลาดกี่ % ถึงจะทำให้ agent ตัดสินใจพลาด?
- Async queue ช่วยลด "ผลกระทบ" ได้จริงแค่ไหน เมื่อ agent ต้องรอผลจาก agent อื่นก่อนตัดสินใจต่อ (dependency chain)?
- ที่ latency เท่าไหร่ถึงจะ**คุ้มที่จะลงทุนทำ** async architecture (เพราะมันมี complexity cost)? ถ้า latency แค่ 50ms อาจไม่คุ้มที่จะทำระบบซับซ้อนขนาดนี้
- แล้วถ้าเป็น packet loss หรือ jitter ล่ะ (ไม่ใช่แค่ delay คงที่) — async queue ช่วยได้เท่าเดิมไหม?

เปรียบเทียบง่ายๆ

คือเหมือนมีคนบอกว่า "ถ้าจะขับรถในหมอกให้เปิดไฟตัดหมอก" (solution ที่มีอยู่แล้ว) — แต่ยังไม่มีใครวัดว่า **"หมอกหนาแค่ไหนถึงจำเป็นต้องเปิดไฟตัดหมอก" หรือ "เปิดไฟตัดหมอกแล้วลดอุบัติเหตุได้กี่ %"** งานคุณคือการวัดตัวเลขพวกนี้ ไม่ใช่การคิดค้น solution ใหม่

งานคุณเติมเต็มช่องว่างตรงไหน

สิ่งที่มีอยู่แล้ว	สิ่งที่งานคุณเติม
Solution/pattern ทั่วไป (async, caching)	**ตัวเลข threshold** ว่าต้องใช้ solution นี้เมื่อไหร่
Best practice จาก experience วิศวกร	**Empirical evidence** จากการทดลองจริงกับ LLM agent
ใช้ได้กับ distributed system ทั่วไป	เจาะจงกับ**พฤติกรรม LLM multi-agent** (hallucination, retry, token cost เพิ่มยังไง) ซึ่งต่างจาก traditional distributed system
บอกว่า "ควรทำอะไร"	บอกว่า "**ทำไมถึงควรทำ และควรทำระดับไหน**"

สรุป

ไม่ใช่ว่า "ปัญหานี้แก้ไปแล้ว" แต่คือ **"มี tool สำหรับแก้ปัญหาแล้ว แต่ยังไม่มีใครบอกว่าเมื่อไหร่ควรใช้ tool ไหน และช่วยได้แค่ไหนจริงๆ สำหรับ LLM agent โดยเฉพาะ"** — งานคุณสามารถอ้างอิง solution พวกนี้ใน paper แล้วบอกว่า "งานวิจัยของเราให้ข้อมูลเชิงประจักษ์ที่ช่วยตัดสินใจว่าเมื่อไหร่ควรลงทุนใช้ solution เหล่านี้"

ส่วนคำถามที่ AI อีกตัวถามคุณ (Data Anonymization / Orchestration tools) นั่นเป็นคนละหัวข้อกับ core research คุณ — เป็นเรื่อง implementation detail ของการ deploy จริง ถ้าอยากให้ผมช่วยแยกว่าอันไหนควรอยู่ใน paper อันไหนไม่เกี่ยว บอกได้ครับ