SPIRAL'S ECHO

Workflow ฉบับละเอียด Google Sheets → RPG Platform → Custom LMS

แนวทางสถาปัตยกรรม ข้อมูล API ความปลอดภัย Offline Sync
การย้ายระบบ และแผนพัฒนาสำหรับ RPG-O-NET P.6

เอกสารออกแบบเชิงระบบสำหรับเจ้าของผลิตภัณฑ์ ทีมเกม ทีม LMS ครู และทีมปฏิบัติการ

เวอร์ชันเอกสาร	1.0 - 13 กรกฎาคม 2026
สถานะ	Architecture proposal - พร้อมใช้เป็นฐานสำหรับ ADR และ Backlog
หลักสำคัญ	Learning ไม่เท่ากับ Combat Power / Client ไม่น่าเชื่อถือ / ข้อมูลผู้เล่นต้องมี version

วิธีอ่านและขอบเขตเอกสาร

- ผู้บริหารหรือเจ้าของผลิตภัณฑ์: อ่านบทสรุป สถาปัตยกรรม เจ้าของข้อมูล แผนระยะ และความเสี่ยง
- ทีมเกม: อ่าน Runtime workflow, Game/Learning firewall, reward authority, offline queue และ migration
- ทีม LMS: อ่าน Identity/SSO, roster, assignment, API contracts, webhook และ reporting
- ครูและทีมเนื้อหา: อ่าน Google Sheets governance, question review, content publication และ teacher reports
- ทีม QA/Operations: อ่าน testing matrix, observability, incident handling, deployment และ checklist

เอกสารนี้เป็น Target Workflow ไม่ได้กำหนดให้ทำ full rewrite ระบบเดิม การเปลี่ยนทุกส่วนต้องทำหลัง feature flag และ adapter พร้อม rollback ตาม Project Constitution

สารบัญ

วิธีอ่านและขอบเขตเอกสาร 2

สารบัญ 3

1. บทสรุปสำหรับการตัดสินใจ 5

ข้อเสนอหลัก . . . 5

คำตัดสินเกียวกับระบบเดิม . . . 5

2. หลักออกแบบและข้อห้ามที่ต้องรักษา 6

3. System of Record และขอบเขตความเป็นเจ้าของข้อมูล 7

กฎการเขียนข้อมูลข้ามระบบ . . . 7

4. บทบาท Google Sheets และโครงสร้าง Workbook 8

สิ่งที่ห้ามเก็บเป็นระบบจริงใน Sheets . . . 8

Workbook governance . . . 8

5. Workflow การสร้าง ตรวจทาน และเผยแพร่คอนเทนต์ 9

Question publication gate . . . 9

6. Data Contract สำหรับคอนเทนต์ 10

ตัวอย่าง Question record หลังผ่านการ import . . . 10

Content manifest . . . 10

7. สถาปัตยกรรมการเชื่อม Custom LMS 11

LMS Integration Boundary . . . 11

8. Identity, SSO, Tenant และสิทธิ์ 12

SSO launch sequence . . . 12

Role permissions . . . 12

9. Roster, Classroom และ Assignment Workflow 13

Assignment contract . . . 13

กฎ Assignment . . . 13

10. Workflow ตั้งแต่เปิดเกมจนจบ Learning Session 14

Knowledge Break boundary . . . 14

11. Learning Event, Mastery และ Spaced Review 15

Mastery update rules . . . 15

12. การแยก Learning, Combat และ Economy 16

Reward idempotency example . . . 16

13. Offline-first และการ Sync อย่างปลอดภัย 17

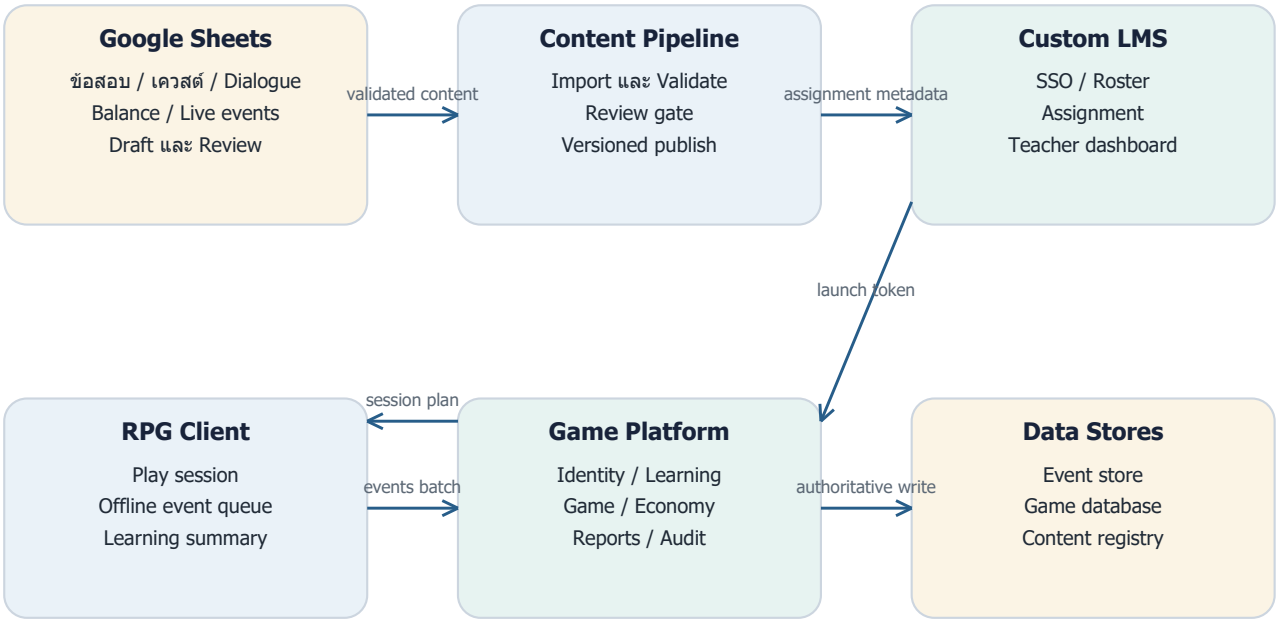
สถานการณ์ Sync . . . 17

14. API Contracts ที่แนะนำ 18

มาตรฐาน response/error . . . 18
15. Webhook และการแจ้งผลกลับ LMS **19**
Webhook security . . . 19
16. Teacher Reporting และ Google Sheets Export **20**
Sheets export flow . . . 20
17. ความเป็นส่วนตัว ความปลอดภัย และข้อมูลเด็ก **21**
Threat scenarios . . . 21
18. การย้ายจากระบบปัจจุบันโดยไม่ Full Rewrite **22**
Rollback points . . . 22
19. Environment, Deployment และ Configuration **23**
Configuration categories . . . 23
20. Observability, Audit และ Operations **24**
Logging rules . . . 24
21. Test Strategy และ Acceptance Gates **25**
Mandatory release gate . . . 25
22. Roadmap การพัฒนาแบบเป็นระยะ **26**
23. Ownership/RACI สำหรับการดำเนินงาน **27**
24. ความเสี่ยงและการตัดสินใจที่ต้องยืนยัน **28**
Owner decisions ก่อนเริ่ม implementation . . . 28
25. Checklist ก่อนเริ่มเชื่อม LMS **29**
Architecture . . . 29
Content . . . 29
Identity/Security . . . 29
Learning . . . 29
Game/Economy . . . 29
LMS/Reports . . . 29
Release . . . 29
ภาคผนวก A. ตัวอย่าง Learning Event Batch **30**
ภาคผนวก B. ตัวอย่าง Session Summary **31**
ภาคผนวก C. Definition of Done สำหรับ LMS Integration Pilot **32**
ภาคผนวก D. ข้อเสนอ Next Action ที่เป็นรูปธรรม **33**
ลำดับที่แนะนำ . . . 33

1. บทสรุปสำหรับการตัดสินใจ

เป้าหมายคือสร้างเส้นทางข้อมูลที่รองรับทั้งการเล่นเพื่อความสนุก การเรียนที่วัดผลได้ การทำงานของครู และการขยายไปยัง LMS โดยไม่ทำให้ระบบหนึ่งกลายเป็นฐานข้อมูลรวมที่เสี่ยงต่อความถูกต้องและข้อมูลเด็ก



ข้อเสนอหลัก

- Google Sheets เป็น authoring surface และ report destination ไม่ใช่ฐานข้อมูลผู้เล่น
- Custom LMS เป็นเจ้าของโรงเรียน ห้องเรียน roster assignment สิทธิ์ และ teacher-facing workflow
- Game Platform เป็นเจ้าของตัวละคร world progression combat inventory economy และ reward ledger
- Learning Service เป็นเจ้าของ attempt events mastery misconception review schedule และ learning summary
- ทุกการเชื่อมต่อผ่านสัญญา API/event ที่มี schemaVersion, idempotency และ audit trail
- ใช้นามแฝง learnerId ภายใน ลดการกระจายชื่อจริง และไม่ส่ง answer key ที่กำลังใช้งานให้ client/LMS โดยไม่จำเป็น

คำตัดสินเกี่ยวกับระบบเดิม

การส่ง player state จาก client ไป Apps Script โดยอิงชื่อผู้เล่นควรจัดเป็น legacy prototype เท่านั้น เมื่อเริ่ม LMS integration ให้หยุดใช้เป็นเซฟและ leaderboard ที่มีผลต่อรางวัล แล้วแทนด้วย GameSaveRepository, RewardLedger, ReportingAdapter และ LmsAdapter

2. หลักออกแบบและข้อห้ามที่ต้องรักษา

หลัก	ผลต่อการออกแบบ
Learning/Combat firewall	Mastery และ spaced review ไม่เพิ่ม ATK, HP, damage หรือ drop rate ระบบ Combat อาจร้องขอ Knowledge Break แต่ผลการเรียนถูกอัปเดตโดย Learning Service เท่านั้น
Server authority	ออนไลน์ต้องให้ server ตรวจผล combat inventory และ reward Client ส่ง intent/event ไม่ส่งผลสำเร็จที่เชื่อได้ทันที
Reviewed content only	คำถาม production ต้อง reviewed มีเฉลย explanation distractor reasoning taxonomy และ provenance
Version everything	Save, content, API, event และ report projection ต้องมีเวอร์ชันพร้อม migration/rollback
Idempotency	event, reward claim, export job และ webhook ต้องส่งซ้ำได้โดยไม่เกิดผลซ้ำ
Privacy by design	เก็บข้อมูลส่วนบุคคลเท่าที่จำเป็น แยก identity mapping ออกจาก learning events และกำหนด retention
Offline deterministic	เล่นออฟไลน์ได้ในขอบเขตที่อนุญาต เก็บ event queue และยืนยันกับ server เมื่อกลับออนไลน์
No full rewrite	สร้าง interface/adapter และเปิดทีละ feature flag เก็บ legacy path จน gate ผ่าน

3. System of Record และขอบเขตความเป็นเจ้าของข้อมูล

ระบบ	ข้อมูลที่เป็นเจ้าของ	แหล่งจริง
Custom LMS	โรงเรียน tenant, ห้องเรียน, ครู, roster, enrollment, assignment, due date, consent/permission reference	LMS Database
Identity Service	mapping ระหว่าง lmsSubject กับ internal learnerId, session/token, account link	Identity Database
Learning Service	attempt event, mastery, misconception, response time summary, next review, learning plan	Event Store + Learning Projection
Game Service	character, class, world/quest progression, encounter/session state	Game Database
Economy Service	currency, inventory, equipment, crafting, rewards, transaction ledger	Transactional Database
Content Registry	question/quest/dialogue/balance version ที่ผ่าน review	Versioned JSON/Object Store/Content DB
Reporting Service	รายงานที่สร้างจาก event/projection; ไม่เป็นต้นทางของ mastery	Reporting Store/Materialized Views
Google Sheets	draft content, review notes, optional report export	ไม่ใช่ authoritative runtime store
RPG Client	UI state, cached content, offline queue, local checkpoint	ชั่วคราวและไม่ authoritative ออนไลน์

กฎการเขียนข้อมูลข้ามระบบ

- LMS ไม่เขียน mastery หรือ inventory โดยตรง แต่ส่ง assignment/roster และอ่าน report projection
- เกมไม่เขียน roster และสิทธิ์ครู แต่ขอข้อมูลผ่าน LMS adapter
- Sheets ไม่เขียน player save หรือ reward ledger ทุกการ import ต้องผ่าน validation และ publish gate
- Reporting เป็น read model จึง rebuild ได้จาก learning events และ authoritative game records

4. บทบาท Google Sheets และโครงสร้าง Workbook

Sheets เหมาะกับงานทีมนุษย์ต้องร่วมกันแก้ไข ตรวจทาน เปรียบเทียบ และอนุมัติ แต่ไม่เหมาะกับธุรกรรมที่ต้อง atomic, realtime, authenticated หรือมีความเสี่ยงจากการส่งซ้ำ

แท็บ	หน้าที่	ลำดับ
Questions	ข้อสอบ draft/reviewed/retired พร้อม taxonomy, explanation, distractor reasoning, provenance	สูงสุด
Question_Review	ผู้ตรวจ สถานะ review issue severity หมายเหตุ approvedAt	สูงสุด
Taxonomy	domain/skill/subskill prerequisite และ misconception vocabulary	สูงสุด
Quests	questId, steps, prerequisites, dialogue keys, rewards references	สูง
NPC_Dialogue	ข้อความ speaker, locale, condition, next node และ accessibility notes	สูง
Monsters/Items/Sigils	ค่าตั้งต้นและ tuning โดย export ผ่าน schema	สูง
Loot/Crafting	ตารางออกแบบ ไม่ใช่ transaction จริง	สูง
Daily_Expeditions	template, learning tags, duration, anti-repeat group	กลาง
Live_Events	draft schedule และ configuration ก่อน publish	กลาง
Teacher_Report_Export	ผลสรุปที่ลดข้อมูลส่วนบุคคลและสร้างตามคำขอ	กลาง
QA_Playtest	บัก device/browser steps severity และหลักฐาน	กลาง
Asset_Catalog	asset ID, path, license, owner, status และ biome	กลาง

สิ่งที่ห้ามเก็บเป็นระบบจริงใน Sheets

- password, token หรือ secret
- ตัวละครและ player save ที่ใช้งานจริง
- inventory/currency/reward claim
- realtime combat/room/position
- raw chat/moderation evidence
- learning event log ปริมาณมาก
- leaderboard ที่แจกของรางวัล
- ข้อมูลระบุตัวเด็กเกินความจำเป็น

Workbook governance

- จำกัดสิทธิ์ edit ตามบทบาท; reviewer และ publisher ไม่ควรเป็นคนเดียวกันในงานสำคัญ
- ใช้ protected range กับ ID, status, contentVersion, reviewer และ publication metadata
- ห้ามแก้ ID หลังเผยแพร่; การเปลี่ยน semantic content สร้าง revision/version ใหม่
- เก็บ import report ทุกครั้ง: จำนวนแถว ผ่าน เตือน ไม่ผ่าน duplicate และ checksum
- Production ไม่เรียก Google Sheets API ระหว่าง session; ใช้ content registry ที่ publish แล้ว

5. Workflow การสร้าง ตรวจทาน และเผยแพร่คอนเทนต์

1. **สร้างหรือแก้ Draft**
   ผู้เขียนกรอกข้อมูลใน Sheets โดย questionId/questId มาจาก ID generator
2. **ดึง Snapshot**
   Importer อ่าน revision ที่ระบุและสร้าง immutable source snapshot
3. **Schema Validation**
   ตรวจชนิดข้อมูล required fields enum ความยาว และ referential integrity
4. **Learning Validation**
   ตรวจ taxonomy เฉลย explanation distractors misconception และ provenance
5. **Quality Analysis**
   ตรวจ duplicate, answer position bias, reading level และความครบของ locale
6. **Human Review**
   ครู/learning reviewer ตรวจความถูกต้อง ความเหมาะสม และลิขสิทธิ์
7. **Approval Gate**
   publisher เปลี่ยน reviewed พร้อม reviewerId, reviewedAt และ source checksum
8. **Build Content Version**
   สร้าง data/*.json หรือ content bundle ที่มี version และ manifest
9. **Automated Test**
   unit/schema/reference/selection tests และ production reviewed-only scan
10. **Publish & Rollback**
    เผยแพร่แบบ staged/canary; rollback ไป content version ก่อนหน้าได้

Question publication gate

Gate	เงื่อนไขผ่าน
Identity	questionId ไม่ซ้ำและไม่เปลี่ยนหลังเผยแพร่
Review	status=reviewed, reviewerId และ reviewedAt มีค่า
Pedagogy	correctAnswer, explanation และเหตุผลตัวลวงทุกตัวครบ
Curriculum	domainId/skillId/subskillId มีจริงและ prerequisite ถูกต้อง
Provenance	ระบุ authored หรือ knowledge-pattern; ห้ามคัดลอกข้อสอบต้นฉบับ
Safety	ไม่มีข้อมูลส่วนบุคคล เนื้อหาไม่เหมาะสม หรือ stereotype
Technical	choice indexes ถูกต้อง references ครบ JSON สร้างได้
Analysis	ไม่มี duplicate และ answer-position bias เกินเกณฑ์ที่ทีมกำหนด

6. Data Contract สำหรับคอนเทนต์

ตัวอย่าง Question record หลังผ่านการ import


```
{
  "questionId": "q-g6-reading-mainidea-0001",
  "schemaVersion": 1,
  "contentVersion": "2026.07.1",
  "status": "reviewed",
  "domainId": "reading",
  "skillId": "reading-comprehension",
  "subskillId": "reading.main-idea",
  "difficulty": 2,
  "prompt": "...",
  "choices": ["...", "...", "...", "..."],
  "correctChoice": 1,
  "explanation": "...",
  "distractorReasoning": {"0":"...", "2":"...", "3":"..."},
  "misconceptionTags": ["detail-not-main-idea"],
  "provenance": {"source":"knowledge-pattern", "patternId":"..."},
  "review": {"reviewerId":"teacher-opaque-id", "reviewedAt":"..."}
}
```


Content manifest

- contentVersion: เวอร์ชัน bundle ที่ client/server ใช้ตรงกัน
- schemaVersion: เวอร์ชันรูปแบบข้อมูล
- generatedAt และ sourceSnapshotId: ตรวจย้อนกลับได้ว่า export จากอะไร
- checksums: ป้องกันไฟล์ไม่ครบหรือแก้หลัง publish
- minimumClientVersion: ป้องกัน client เก่าอ่าน schema ใหม่ผิด
- rollbackVersion: เวอร์ชันก่อนหน้าที่ผ่าน gate

7. สถาปัตยกรรมการเชื่อม Custom LMS

แนะนำใช้ Anti-Corruption Layer ชื่อ LmsAdapter อยู่ระหว่าง domain ของเกมกับ LMS เพื่อไม่ให้โครงสร้างภายใน LMS แพร่เข้า game code ทุกจุด

Adapter	หน้าที่	การใช้
LocalLmsAdapter	โหมดพัฒนา/ออฟไลน์ ส่ง roster และ assignment จำลอง	Phase แรก
HttpLmsAdapter	เรียก Custom LMS API พร้อม token และ retry policy	Production
StandardsAdapter	ทางเลือกภายหลัง เช่น LTI/xAPI โดยแปลงเข้า contract กลาง	Optional

LMS Integration Boundary

- LMS ส่ง identity assertion, tenant/class context และ assignment intent
- Game Platform คืน launch session, completion, learning summary และ report links
- LMS ไม่ได้รับ raw save, inventory transaction หรือ active answer key
- Game ไม่เก็บข้อมูลโรงเรียนซ้ำทั้งหมด เก็บเพียง external references และ snapshot ที่จำเป็นต่อ audit
- การลบ/ระงับ enrollment ต้องส่งผลต่อสิทธิ์ทันที แต่ไม่ลบ audit/history โดยไม่มี retention workflow

8. Identity, SSO, Tenant และสิทธิ์

Identity model ต้องรองรับหลายโรงเรียนและป้องกัน external ID ชนกัน จึงใช้ composite identity: provider + tenantId + subject แล้ว map ไป internal learnerId

ฟิลด์	หน้าที่
tenantId	ขอบเขตโรงเรียน/องค์กร ใช้แบ่งข้อมูลและสิทธิ์
lmsSubject	opaque ID จาก LMS ไม่ใช่ชื่อหรืออีเมล
learnerId	ID ภายใน Game Platform ใช้ใน learning/game records
classId	external reference + tenant scope
role	student, teacher, school-admin, content-reviewer, platform-admin
launchSessionId	อ้างอิงการเปิดเกมหนึ่งครั้ง มีอายุและสถานะ

SSO launch sequence

1. **LMS Login**
   LMS ยืนยันผู้ใช้และตรวจ enrollment/role
2. **Create Launch Assertion**
   สร้าง assertion แบบ signed, audience เฉพาะเกม และหมดอายุเร็ว
3. **Token Exchange**
   backend เกมตรวจ signature, issuer, audience, nonce, expiry และ replay
4. **Identity Mapping**
   ค้น/สร้าง learnerId ภายใน tenant ที่ถูกต้อง
5. **Issue Game Token**
   ออก access token อายุสั้นและ refresh strategy ที่เหมาะสม
6. **Create Session**
   ผูก assignment/class/contentVersion และ audit launch context

ห้ามวาง token, ชื่อ, อีเมล หรือ answer key ใน query string เนื่องจากอาจติด browser history, proxy logs และ analytics

Role permissions

บทบาท	สิทธิ์หลัก
Student	เล่น ดูผลของตนเอง และคำอธิบายที่อนุญาต
Teacher	ดูห้องที่ได้รับมอบหมาย สร้าง assignment และ export report
School admin	จัด roster/teacher mapping และ retention ภายใน tenant
Content reviewer	อ่าน/อนุมัติคอนเทนต์ ไม่มีสิทธิ์ดู player inventory
Platform admin	ดูระบบ/incident ตาม least privilege และต้องมี audit log

9. Roster, Classroom และ Assignment Workflow

LMS เป็น system of record ของ roster เกมควร cache เฉพาะข้อมูลที่จำเป็นและใช้ sync cursor/version เพื่อหลีกเลี่ยงการโหลดทั้งหมดทุกครั้ง

Assignment contract


```
{
  "assignmentId": "asg-2026-001",
  "tenantId": "school-001",
  "classId": "class-6-1",
  "title": "Reading: Main Idea",
  "subskillIds": ["reading.main-idea", "reading.supporting-detail"],
  "targetMinutes": 20,
  "mode": "learning-focus",
  "dueAt": "2026-07-20T16:00:00+07:00",
  "contentPolicy": {"minimumStatus":"reviewed"},
  "completionRule": {"minimumAttempts":12, "minimumActiveMinutes":10},
  "version": 3
}
```


กฎ Assignment

- มอบหมายด้วย skill/subskill และเวลาหรือกิจกรรม ไม่ควรล็อกกับ questionId จำนวนมากเพราะทำให้การปรับ adaptive จำกัด
- completion แยกจาก mastery: ทำครบไม่ได้แปลว่าเชี่ยวชาญ และ mastery ต่ำไม่ควรทำให้สถานะงานหาย
- การแก้ assignment หลังเริ่มต้องเพิ่ม version และเก็บ snapshot ที่ session เดิมใช้
- ยกเลิก assignment ไม่ควรลบ attempt history; เปลี่ยนสถานะและกำหนด visibility
- นักเรียนย้ายห้องต้องคง learning history ภายใต้ learnerId แต่สิทธิ์ครูเก่าต้องสิ้นสุดตาม policy

10. Workflow ตั้งแต่เปิดเกมจนจบ Learning Session

1. **Launch**
   ตรวจ game token, tenant, role และ assignment context
2. **Load profile**
   โหลด character/save ผ่าน versioned repository และ migration registry
3. **Resolve content**
   เลือก contentVersion ที่ compatible และดาวน์โหลด manifest/cache
4. **Build plan**
   Learning Service เลือก reviewed questions ตาม assignment + mastery + due review
5. **Start session**
   สร้าง sessionId, mode, duration, seed และ initial checkpoint
6. **Play**
   สำรวจ ต่อสู้ ทำเควสต์ และ trigger Knowledge Break ตาม cadence
7. **Record intent**
   Client สร้าง answer event/interaction event พร้อม eventId และ monotonic sequence
8. **Validate**
   Server ตรวจ session, question version, timing sanity, duplicate และ permission
9. **Project learning**
   MasteryUpdater/ReviewScheduler อัปเดต learning projection
10. **Resolve game**
    Combat domain ประมวลผลผลเกมโดยไม่อ่าน mastery เป็น damage modifier
11. **Reward**
    Economy รับ RewardRequest ที่ตรวจแล้วและบันทึก ledger แบบ idempotent
12. **Complete**
    สร้าง session summary, assignment progress และส่ง webhook/report projection

Knowledge Break boundary

- Combat ส่ง trigger reason เช่น elite armor break, boss phase, rune door หรือ treasure challenge
- Learning เลือกคำถามที่ reviewed และเหมาะกับ mastery/assignment
- ผลผิดหนึ่งครั้งไม่ทำให้ตายทันที; feedback สั้น อ่านได้ และนัด review ในอนาคต
- ถ้าไม่มีคำถามที่ผ่านเงื่อนไข เกมต้องดำเนินต่อด้วย no-question fallback

11. Learning Event, Mastery และ Spaced Review

Event เป็นข้อเท็จจริงทีเกิดขึนแล้วและไม่แก้ทับ ส่วน mastery เป็น projection ทีคำนวณใหม่ได้ การแยกสองส่วนทำให้ตรวจย้อนหลัง แก้สูตร และสร้างรายงานใหม่ได้

ฟิลด์	ความหมาย
eventId	UUID/ULID ที unique ใช้ idempotency
schemaVersion	เวอร์ชัน event contract
tenantId/learnerId	ขอบเขตและตัวตนแบบ opaque
classId/assignmentId	context ทีเป็น optional ตาม session
sessionId/sequence	จัดกลุ่มและตรวจลำดับ
questionId/contentVersion	พิสูจน์ว่าเด็กเห็นข้อใด revision ใด
subskillId	แกนสำหรับ mastery
outcome	correct, incorrect, timeout, skipped ตาม enum
selectedChoice	เก็บตาม policy; ไม่ส่งออกกว้างโดยไม่จำเป็น
responseTimeMs	ผ่าน sanity bounds และไม่ใช้ลงโทษเด็ก
occurredAt/receivedAt	เวลาจาก client และ server เพือรองรับ offline
source	online, offline-sync, teacher-session พร้อม client version

Mastery update rules

- อัปเดตเฉพาะ event ที validated และยังไม่ประมวลผล
- เก็บความมันใจ ความเสถียร จำนวนหลักฐาน misconception และ nextReview แยกตาม subskill
- รองรับ deterministic replay โดย version สูตร masteryUpdaterVersion
- เมือเปลียนสูตร ให้ rebuild projection หรือ migrate อย่างบันทึกเวอร์ชัน ไม่แก้ event ต้นฉบับ
- การตอบเร็ว/ช้าเป็นสัญญาณประกอบ ไม่ควรเป็นตัวตัดสินความสามารถเพียงอย่างเดียว

12. การแยก Learning, Combat และ Economy

เส้นทาง	ข้อมูลที่อนุญาต
Learning → Combat	ส่งเพียง outcome/effect ที่ออกแบบ เช่น เปิดเกราะหรือคงสถานะ ไม่ส่ง mastery เป็น damage multiplier
Combat → Learning	ส่ง answer context และ trigger reason ไม่เขียน mastery store
Combat → Economy	สร้าง RewardRequest หลัง encounter result ผ่าน validation
Economy → Combat	คืน transaction result/claim receipt ไม่ให้ UI เพิ่มของเอง
LMS → Learning	ส่ง assignment intent และ learner context ไม่เขียนคะแนน mastery โดยตรง
Learning → LMS	ส่ง summary/progress projection ไม่ส่ง raw answer key

Reward idempotency example

```
RewardRequest {
  requestId: "reward:<encounterId>:<learnerId>",
  learnerId: "...",
  encounterId: "...",
  validatedResultId: "...",
  rewardTableVersion: "world1-2026.07.1"
}

UNIQUE(requestId) + atomic transaction
=> retry ได้ แต่ gold/item เพิ่มเพียงครั้งเดียว
```

13. Offline-first และการ Sync อย่างปลอดภัย

Offline mode ต้องกำหนดขอบเขตชัด: การเรียนและ single-player deterministic ทำได้ แต่ online-critical inventory/trade/party reward อาจต้องรอ server ยืนยัน

ส่วน	ข้อกำหนด
Local checkpoint	SaveEnvelope แบบ versioned พร้อม timestamp, checksum และ backup ก่อน migration
Event queue	append-only records พร้อม eventId, sequence, retryCount และ nextRetryAt
Content cache	manifest + bundle checksum; ไม่ใช้ draft content
Sync batch	ส่งตามลำดับ/ขนาดจำกัด รับ ack ราย eventId
Conflict policy	attempt events merge แบบ append; profile settings ใช้ field policy; economy server wins
Clock handling	ใช้ occurredAt เป็นข้อมูลประกอบและ receivedAt เป็นเวลาที่เชื่อถือได้
Recovery	ห้ามลบ queue ก่อน server ack; malformed record quarantine และแจ้ง diagnostic

สถานการณ์ Sync

เหตุการณ์	การจัดการ
ส่ง event ซ้ำ	Server ตอบ accepted-already-processed และไม่อัปเดต mastery/reward ซ้ำ
content version เก่า	Server ตรวจว่ายังยอมรับได้; ถ้าไม่ได้ให้ mark rejected พร้อมเหตุผล ไม่เดาคำตอบ
token หมดอายุ	หยุดส่ง ขอ re-auth แล้ว retry โดยคง eventId เดิม
ออกจากห้องแล้ว	ตรวจว่า event เกิดตอน enrollment ยัง valid ผ่าน session snapshot
queue เสียบางรายการ	quarantine เฉพาะรายการ ไม่ reset save ทั้งก้อน

14. API Contracts ที่แนะนำ

Endpoint	หน้าที่
POST /api/v1/lms/auth/exchange	แลก signed LMS assertion เป็น game token
POST /api/v1/game/sessions	สร้าง game/learning session พร้อม assignment context
GET /api/v1/learners/me/assignments	อ่านงานที่มีสิทธิ์และสถานะล่าสุด
GET /api/v1/content/manifests/:version	อ่าน manifest ที่ publish แล้ว
POST /api/v1/learning/events/batch	ส่ง events แบบ offline-safe
POST /api/v1/game/encounters/:id/intents	ส่ง combat intent ในโหมด authoritative
POST /api/v1/rewards/claims	ภายใน server boundary หรือ trusted service only
POST /api/v1/game/sessions/:id/complete	ปิด session และสร้าง summary
GET /api/v1/reports/classes/:classId	teacher class overview
GET /api/v1/reports/learners/:learnerId	รายงานรายบุคคลตามสิทธิ์
POST /api/v1/reports/exports	สร้าง CSV/JSON/Sheets export job

มาตรฐาน response/error

```
{
  "requestId": "req-...",
  "ok": false,
  "error": {
    "code": "CONTENT_VERSION_UNSUPPORTED",
    "message": "ข้อความสำหรับผู้ใช้/ผู้ดูแลที่ไม่เปิดเผยข้อมูลลับ",
    "retryable": false,
    "details": {"supportedMinimum": "2026.07.1"}
  }
}
```

- ทุก mutation รับ Idempotency-Key หรือมี domain eventId/requestId
- ใช้ pagination/cursor กับ roster, events และ reports
- กำหนด request/response schema และ contract tests ระหว่างทีม LMS กับเกม
- ไม่คืน stack trace, secret, internal database ID หรือ active answer key

15. Webhook และการแจ้งผลกลับ LMS

Event	Payload ที่ควรมี
game.session.completed	จบ session พร้อม active minutes และ summary reference
learning.assignment.progressed	progress เปลี่ยนตามเกณฑ์ที่กำหนด
learning.assignment.completed	ทำ completion rule ครบ ไม่เท่ากับ mastery สำเร็จ
learning.review.recommended	มี subskills ถึงกำหนดทบทวน
identity.link.changed	mapping หรือสถานะ link เปลี่ยน

Webhook security

- ลงลายเซ็น payload และมี timestamp/nonce ป้องกัน replay
- มี deliveryId unique; LMS deduplicate ก่อนเขียน
- retry แบบ exponential backoff พร้อม dead-letter queue
- ตอบรับเร็วแล้วประมวลผล async; ห้ามให้ webhook failure rollback session นักเรียน
- มีหน้า operations สำหรับดู pending/failed/replay โดยบันทึก audit

16. Teacher Reporting และ Google Sheets Export

Reporting Service อ่าน event/projection แบบ read-only และสร้างมุมมองตาม tenant/class permission ครูควรเห็นแนวโน้มและคำแนะนำ ไม่ใช่ข้อมูลเกมหรือเฉลยที่ทำให้ข้อสอบรั่ว

รายงาน	รายละเอียด
Class overview	จำนวนผู้เรียน active, completion, accuracy/mastery distribution ราย subskill
Learner profile	mastery movement, misconceptions, due review, session history
Assignment report	assigned/started/completed, active minutes, attempts, coverage
Content quality	difficulty, distractor selection, timeout, possible ambiguity สำหรับ reviewer
Operational	sync failures, unsupported client/content version โดยไม่เปิดเผย PII เกินจำเป็น

Sheets export flow

1. **Request Export**
   ครูเลือกห้อง ช่วงเวลา และชนิดรายงาน
2. **Authorize**
   ตรวจ tenant, class assignment และ scope
3. **Create Snapshot**
   freeze report cutoff และ query version
4. **Minimize Data**
   ใช้ learner code/alias แทนชื่อ ถ้าไม่จำเป็น
5. **Write Sheet**
   service account เขียนเฉพาะปลายทางที่อนุญาต
6. **Audit**
   เก็บผู้ขอ เวลา scope ปลายทาง จำนวนแถว และ expiry
7. **Expire/Revoke**
   รองรับลบหรือยกเลิก link ตาม retention policy

รายงานต้องไม่เป็นทางกลับเพื่อเขียน mastery หากครูต้องแก้ข้อมูล ให้ใช้ workflow correction ที่มีเหตุผล ผู้อนุมัติ และ audit event แยกต่างหาก

17. ความเป็นส่วนตัว ความปลอดภัย และข้อมูลเด็ก

มาตรการ	ข้อกำหนด
Data minimization	เก็บเฉพาะข้อมูลที่จำเป็นต่อการเรียน เกม และหน้าที่ตามสิทธิ์
Pseudonymization	learning events ใช้ learnerId; identity mapping อยู่ store แยก
Encryption	TLS ระหว่างระบบ และเข้ารหัสข้อมูลสำคัญที่ rest ตาม infrastructure
Least privilege	ครูเห็นเฉพาะห้อง reviewer ไม่เห็น player data และ service account จำกัด scope
Audit	login, export, content publish, permission change, reward adjustment และ admin access
Retention	กำหนดระยะของ raw event, report, audit, backup และ export อย่างชัดเจน
Deletion workflow	ตรวจสิทธิ์ ลบ/ทำให้ไม่ระบุตัวตน และบันทึก tombstone โดยไม่ทำลาย ledger
Incident response	จำแนก ตรวจ containment แจ้งผู้รับผิดชอบ recover และ postmortem

Threat scenarios

ภัยคุกคาม	การควบคุม
นักเรียนปลอม floor/level/gold	Server ไม่รับ player state จาก client; ตรวจ intent และ ledger
ขโมย launch URL	token อายุสั้น audience-bound nonce/replay check และไม่อยู่ query string
ครูเปิดรายงานห้องอื่น	tenant/class authorization ทุก request และ audit denied access
Apps Script URL รั่ว	ยุติ save endpoint; ใช้ authenticated backend และ rotate credentials
แก้ Sheet หลัง publish	runtime ใช้ immutable content bundle/checksum ไม่อ่าน live Sheet
ส่ง event offline ปลอมจำนวนมาก	session/content validation, rate limit, sequence/sanity และ anomaly flags

18. การย้ายจากระบบปัจจุบันโดยไม่ Full Rewrite

ปัจจุบันมี useSheetsSync ที่ส่ง player state ไป Apps Script และมี local Pinia persistence แนวทางย้ายต้องรักษาเซฟเดิม ทำ adapter และเปิดทีละกลุ่มผู้ใช้

ระยะ	งาน
M0 - Inventory	บันทึก schema/data flow ของ Players และ Leaderboard Sheet; ปิดการเพิ่ม field ใหม่
M1 - Interfaces	สร้าง PlayerRepository, RewardLedger, ReportingSink และ LmsAdapter โดย legacy adapter ยังทำงาน
M2 - Versioned save	อ่าน legacy localStorage → backup → migrate SaveEnvelope แบบ idempotent
M3 - Reporting only	เปลี่ยน Sheets write จาก player save เป็น report snapshot ที่ลดข้อมูล
M4 - Backend shadow	เขียน backend พร้อมเปรียบเทียบผลโดยไม่เป็น authoritative
M5 - Authority cutover	เปิด server-authoritative save/reward เป็น cohort ผ่าน feature flag
M6 - LMS pilot	เปิด token exchange, roster และ assignment สำหรับหนึ่งห้อง
M7 - Retire legacy	หยุด Apps Script save/load หลัง compat, backup, rollback และ audit ผ่าน

Rollback points

- ทุก phase มี feature flag และ adapter กลับ legacy path
- ก่อน save migration สร้าง backup envelope ที่อ่านกลับได้
- content publish เก็บ previous stable version และ cache manifest
- authority cutover แยกตาม tenant/cohort ไม่เปิดทั้งระบบพร้อมกัน
- ห้าม rollback ด้วยการ reset ข้อมูลผู้เล่น; ต้องใช้ migration/reconciliation ที่ตรวจสอบได้

19. Environment, Deployment และ Configuration

Environment	ลักษณะ
Local	LocalLmsAdapter, local repositories, seeded content, debug diagnostics
Development	shared backend, test tenant, draft/review sandbox, synthetic learners
Staging	production-like auth, reviewed test content, migration rehearsal, load/security tests
Production	strict reviewed content, secrets manager, backups, monitoring, restricted admin

Configuration categories

- Secrets: signing keys, database credentials, Sheets service account - เก็บนอก repository
- Deploy config: endpoint, issuer, audience, timeouts, retention - versioned และแยก environment
- Gameplay config: balance/content - publish ผ่าน validation pipeline
- Feature flags: LMS launch, backend save, Sheets report export - มี owner และ expiry date
- Protocol/content minimum versions: ป้องกัน client/server/content ไม่เข้ากัน

20. Observability, Audit และ Operations

หมวด	ตัวชี้วัด
Availability	API success rate, p95 latency, auth exchange failure
Sync	queue age, batch rejection, duplicate rate, unsupported content version
Learning	event validation rate, projection lag, no-question fallback rate
Economy	duplicate prevented, reconciliation mismatch, negative currency invariant
LMS	webhook delivery lag/failure, roster sync age, assignment fetch error
Content	import errors, review lead time, publish rollback, invalid reference

Logging rules

- ใช้ requestId/sessionId/eventId เพื่อ trace แต่ไม่ log token หรือ full answer payload โดยไม่จำเป็น
- แยก security audit log จาก application log และจำกัดผู้เข้าถึง
- alerts ต้องผูกกับ action: ใครรับผิดชอบ ตรวจ dashboard ใด และ rollback อย่างไร
- เก็บ runbook สำหรับ webhook backlog, event projection lag, content rollback และ save migration failure

21. Test Strategy และ Acceptance Gates

ชุดทดสอบ	กรณีหลัก
Contract	LMS/Game request-response, webhook signatures, backward compatibility
Content	schema, reviewed-only, duplicate, taxonomy/reference, answer distribution
Save	legacy migration, repeat migration, corruption recovery, backup restore
Learning	deterministic selection/mastery/review, weak skills recur, no combat coupling
Economy	atomic reward, duplicate/replay, no negative currency, reconciliation
Offline	disconnect/reconnect, duplicate batch, expired token, partial ack, queue corruption
Authorization	cross-tenant/class denial, role matrix, export permission
Performance	class dashboard, batch ingestion, content delivery, concurrent sessions
Accessibility	keyboard/touch, timing, reduced motion, readable feedback
Manual pilot	new/returning learner, teacher assignment/report, admin roster changes

Mandatory release gate

- production build และ automated tests ผ่าน
- ไม่มี unreviewed question selectable
- save migration/rollback rehearsal ผ่าน
- event/reward replay ไม่สร้างผลซ้ำ
- cross-tenant access tests ถูกปฏิเสธ
- offline queue recover ได้โดยไม่ reset ผู้เล่น
- teacher report ไม่เปิด active answer key และเข้าใจได้
- mobile/accessibility/manual smoke ผ่าน
- มี independent review และ unresolved risks ถูกจัดระดับตรงจริง

22. Roadmap การพัฒนาแบบเป็นระยะ

ระยะ	ขอบเขต	ประมาณเวลา	Exit criterion
R0 Architecture contract	ADR, ownership matrix, event/API schemas, privacy map	1-2 สัปดาห์	ไม่มี runtime change
R1 Content pipeline	Sheets template, importer, validator, review/publish	2-4 สัปดาห์	Questions production จาก reviewed bundle
R2 Learning events	event queue/store, mastery projection, reporting seed	3-5 สัปดาห์	offline-safe attempts
R3 Backend persistence	save repository, reward ledger, migration/shadow	4-8 สัปดาห์	server authority pilot
R4 LMS identity	SSO exchange, tenant, roster, role permissions	3-6 สัปดาห์	หนึ่ง test tenant
R5 Assignment/report	assignment API, summaries, dashboard/export	4-7 สัปดาห์	teacher pilot
R6 Reliability/security	load, backup, DR, monitoring, abuse testing	3-6 สัปดาห์	production gate
R7 Scale-out	multi-school ops, standards adapters, advanced analytics	ต่อเนื่อง	หลังมี measured evidence

หมายเหตุ: ระยะเวลาเป็น planning range สำหรับทีมขนาดเล็กถึงกลาง ต้องปรับจากทีมจริง infrastructure และ scope ของ LMS ไม่ใช่ commitment

23. Ownership/RACI สำหรับการดำเนินงาน

งาน	Responsible	Accountable	Consulted
Architecture/API	Game Architect	LMS Lead	Security, Learning, QA
Question taxonomy/content	Learning Architect	Teacher Reviewer	Content Ops, QA
Save/migration	Save Migration Engineer	Game Architect	QA, Operations
Combat	Combat Engineer	Game Architect	Learning เฉพาะ boundary
Economy/rewards	Economy Engineer	Game Architect	Security, QA
LMS/identity	LMS Lead	Security Lead	Game Architect, QA
Reports/privacy	Learning + LMS	Product/Data Owner	Security, Teachers
Release	QA/Release	Product Owner	ทุก domain owner

24. ความเสี่ยงและการตัดสินใจที่ต้องยืนยัน

ระดับ	ความเสี่ยง	ผล	การควบคุม
สูง	ใช้ Sheets ต่อเป็น player save	โกง/ทับชื่อ/ไม่มี migration	หยุดเป็น authoritative; ทำ backend + adapter
สูง	ตัวตนข้ามโรงเรียนชนกัน	ข้อมูลผิดคนหรือข้าม tenant	composite identity + tenant isolation tests
สูง	Reward จาก client	duplication/economy เสีย	server validation + unique ledger key
สูง	ข้อมูลเด็กกระจาย	privacy incident	pseudonymization + retention + export audit
กลาง	LMS/Game schema เปลี่ยน	integration พัง	versioned contract + compatibility window
กลาง	offline event ล่าช้า/ซ้ำ	mastery ผิด	eventId + ack + deterministic projection
กลาง	ครูแก้ Sheet หลัง publish	content ไม่ตรงรายงาน	immutable bundle + checksum/version
กลาง	รายงานตีความ mastery เป็นเกรด	การใช้ผลผิดวัตถุประสงค์	แสดง evidence/confidence และคำอธิบาย

Owner decisions ก่อนเริ่ม implementation

- LMS จะเป็นระบบบัญชีหลักหรือเชื่อม identity provider ภายนอก
- ข้อมูลใดจำเป็นต้องแสดงชื่อจริง และใครเห็นได้
- retention ของ attempts, reports, audit และ exports
- assignment completion rule และความหมายของ mastery ในรายงาน
- offline scope: อนุญาตระบบเกม/รางวัลใดก่อน sync
- hosting/database/backup/monitoring budget และเขตที่จัดเก็บข้อมูล
- ขั้นตอนให้ความยินยอมและการลบข้อมูลตามนโยบายองค์กร

25. Checklist ก่อนเริ่มเชื่อม LMS

Architecture

- ☐ ADR ระบบเจ้าของข้อมูลได้รับการอนุมัติ
- ☐ LmsAdapter/Repository/Event contracts ถูกกำหนด
- ☐ legacy rollback path มีอยู่
- ☐ protocol/content/save versions ชัดเจน

Content

- ☐ Sheets template และ protected ranges พร้อม
- ☐ review roles แยกจาก publisher
- ☐ import/validation report ทำซ้ำได้
- ☐ production reviewed-only test ผ่าน

Identity/Security

- ☐ tenant model และ ID mapping ชัด
- ☐ token issuer/audience/expiry/nonce กำหนด
- ☐ role matrix/cross-tenant tests พร้อม
- ☐ secret storage/audit/retention พร้อม

Learning

- ☐ taxonomy/subskills stable version
- ☐ event schema และ idempotency พร้อม
- ☐ mastery updater deterministic/versioned
- ☐ Learning ไม่เชื่อมกับ combat stats

Game/Economy

- ☐ save envelope + migrations + backup
- ☐ reward ledger unique/atomic
- ☐ client ไม่ claim victory/reward เอง
- ☐ offline conflict policies มี tests

LMS/Reports

- ☐ roster/assignment contracts ผ่าน contract test
- ☐ webhook signing/retry/DLQ พร้อม
- ☐ reports มี permission filters
- ☐ Sheets export ลด PII และมี audit

Release

- ☐ staging เทียบ production
- ☐ build/test/security/load ผ่าน
- ☐ pilot cohort และ rollback owner ชัด
- ☐ runbooks/monitoring/on-call พร้อม

ภาคผนวก A. ตัวอย่าง Learning Event Batch

```
POST /api/v1/learning/events/batch
Idempotency-Key: batch-device123-0042

{
  "schemaVersion": 1,
  "deviceId": "device-opaque-id",
  "batchId": "batch-device123-0042",
  "events": [{
    "eventId": "evt-01J...",
    "tenantId": "school-001",
    "learnerId": "lrn-opaque",
    "classId": "class-6-1",
    "assignmentId": "asg-001",
    "sessionId": "ses-001",
    "sequence": 7,
    "questionId": "q-g6-reading-mainidea-0001",
    "contentVersion": "2026.07.1",
    "subskillId": "reading.main-idea",
    "outcome": "incorrect",
    "selectedChoice": 2,
    "responseTimeMs": 12400,
    "occurredAt": "2026-07-13T10:30:00+07:00"
  }]
}

Response:
{
  "accepted": ["evt-01J..."],
  "duplicates": [],
  "rejected": [],
  "projectionVersion": "mastery-v2"
}
```

ภาคผนวก B. ตัวอย่าง Session Summary


```
{
  "summaryId": "sum-001",
  "sessionId": "ses-001",
  "learnerId": "lrn-opaque",
  "assignmentId": "asg-001",
  "startedAt": "2026-07-13T10:00:00+07:00",
  "completedAt": "2026-07-13T10:22:00+07:00",
  "activeMinutes": 19,
  "attempts": 18,
  "correct": 13,
  "timeouts": 1,
  "skills": [{
    "subskillId": "reading.main-idea",
    "attempts": 8,
    "masteryBefore": 0.46,
    "masteryAfter": 0.54,
    "confidence": 0.61,
    "nextReviewAt": "2026-07-16T08:00:00+07:00"
  }],
  "assignmentProgress": {"status":"completed", "percent":100},
  "gameSummary": {"expeditionCompleted":true, "bossDefeated":false}
}
```

ภาคผนวก C. Definition of Done สำหรับ LMS Integration Pilot

- นักเรียนหนึ่งห้องเปิดเกมจาก LMS โดยไม่ login ซ้ำและไม่เปิดเผย token ใน URL
- เกมโหลด assignment ถูกคน ถูกห้อง และสร้าง adaptive plan จาก reviewed content เท่านั้น
- ตอบคำถามออนไลน์และออฟไลน์แล้ว sync โดยไม่มี event/mastery/reward ซ้ำ
- Learning state ไม่เปลี่ยน combat stats และ combat result ไม่เขียน mastery โดยตรง
- ครูเห็น report เฉพาะห้องของตนและ export ได้โดยมี audit
- ถอน enrollment แล้วเข้าถึงใหม่ไม่ได้ตาม policy แต่ historical audit ไม่เสีย
- save เก่า migrate ได้ ทำซ้ำได้ และ restore backup ได้
- Apps Script player save ถูกปิดสำหรับ pilot cohort โดย rollback ได้
- build/tests/manual smoke/mobile/accessibility/security checks ผ่าน
- มี owner, monitoring, runbook และ rollback decision สำหรับช่วง pilot

ภาคผนวก D. ข้อเสนอ Next Action ที่เป็นรูปธรรม

งานถัดไปที่ควรทำก่อนเขียนระบบเชื่อมจริงคือ Architecture Contract Package หนึ่งชุด ประกอบด้วย ADR system of record, identity model, learning event schema, assignment schema, API/OpenAPI draft, privacy data map, migration plan และ pilot acceptance matrix จากนั้นจึงแตก backlog ให้เจ้าของแต่ละ domain ตาม constitution

ลำดับที่แนะนำ

- 1) ปิดการขยาย Apps Script save schema และประกาศเป็น legacy prototype
- 2) ออกแบบ Google Sheets Questions/Review/Taxonomy template ให้ตรง schema ปัจจุบัน
- 3) สร้าง importer + validation report โดยยังไม่แตะ runtime
- 4) กำหนด LearningEvent และ LmsAdapter interfaces พร้อม contract tests
- 5) สร้าง LocalLmsAdapter เพื่อให้เกมพัฒนาได้ก่อน LMS จริงเสร็จ
- 6) สร้าง backend persistence/reward ledger แบบ shadow และทำ save migration rehearsal
- 7) เชื่อม Custom LMS ใน staging แล้ว pilot หนึ่งห้องก่อนขยาย

จบเอกสาร - ใช้เป็นฐานสำหรับการประชุม Architecture, Data Governance และ LMS Integration Planning