AppForge 6GL: The Final Blueprint

ฉบับปรับให้ตรง Final Review v1.0 · 25 ต.ค. 2026

Purpose

- กรอบคุณภาพ UX/UI และ agent-native foundation สำหรับวางแผน implementation
- สังเคราะห์ Codex framework + independent reviews ของ Muse Code และ Antigravity

Direction

- Capability-first + services/policy ชุดเดียว
- Human UX และ Jarvis ต้องน่าใช้; agent APIs ต้อง concise และ reliable
- ตรวจฐาน → reuse → prove → scale

Planning horizon

- เป้าหมายเดิม: 23 สิงหาคม 2027
- Readiness ราย declared capability พร้อม evidence
- ไม่ใช่คำรับรอง deployed features หรือผล UAT

เอกสารต้นทางยังเป็น Final Review revision 1.0; ฉบับแนบนีปรับเป็น aligned edition 1.1

Strategic target: readiness ตาม capability scope

23 สิงหาคม 2027 · กรอบลงทุน 10 เดือน ไม่ใช่คำรับรองทุก feature (§1,14)

1. Re-baseline / reuse

- ตรวจ main/deployed/artifact/flags
- แยก source presence ออกจาก supported behavior
- reuse contracts/recipes/runtime ที่พิสูจน์แล้ว

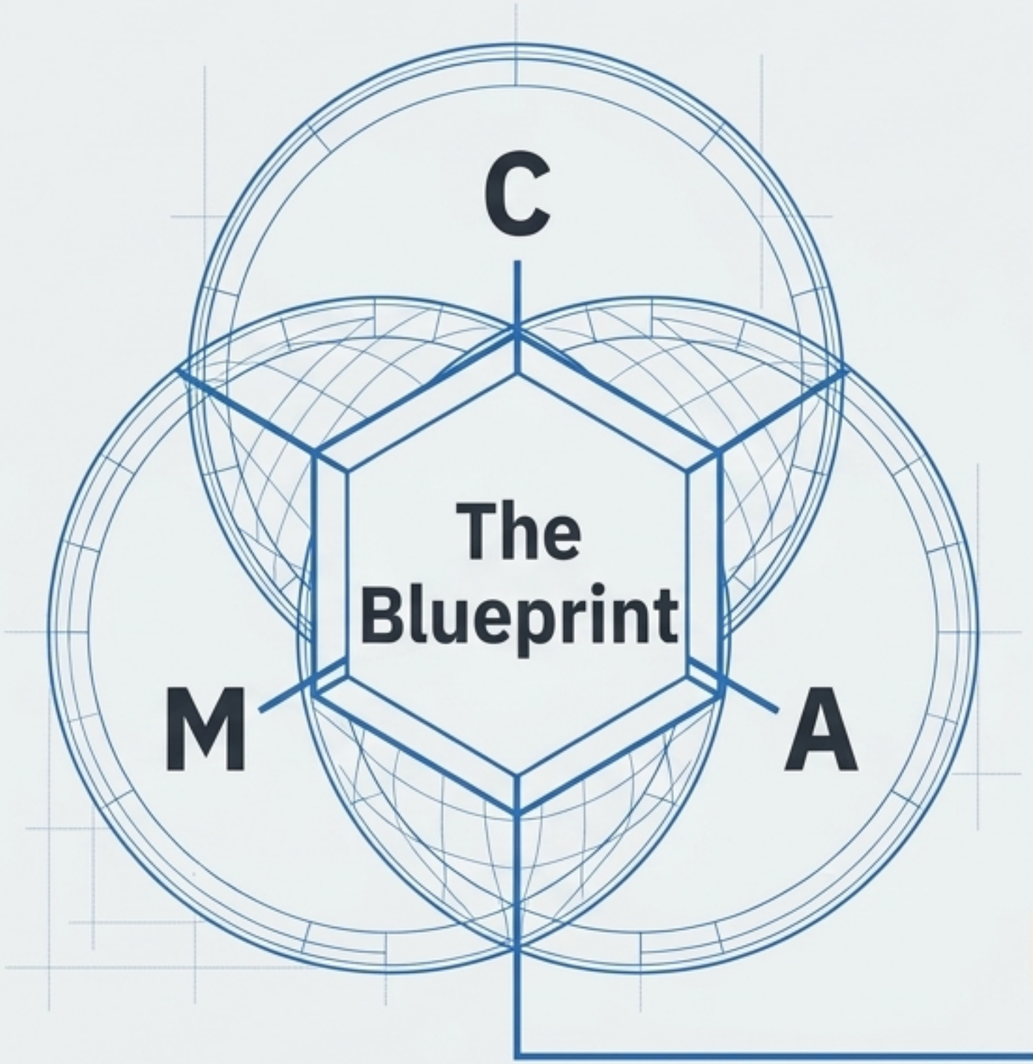
2. Prove one workflow

- Approval sandbox ผ่าน Human/Jarvis/Agent ตาม actor needs
- POS product editor เป็น visual/regeneration proof คู่ขนาน
- authority/idempotency/concurrency/evidence ก่อนเปิด mutation

3. Scale from evidence

- cross-domain proof ช่วง 61–90 วัน
- เพิ่ม adapters/connectors ตาม need หลัง structured flow ผ่าน
- ขยายพร้อม readiness matrix และ independent GO/NO-GO

คุณค่าคือทำงานธุรกิจได้จริงและเข้าใจง่าย; token savings/compiled coverage ไม่ใช่ตัวแทนคุณภาพ



Codex: Quality & Contract

- กำหนดกรอบคุณภาพ Output และ Visual standards
- รักษา Intent ของ Requirement
- กลยุทธ์เส้นทาง Agent-native

Muse Code: Migration & Reuse

- เน้นวินัยในการ Re-baseline และรันระบบเดิม
- ใช้ Capability sidecar ลดการเขียนโค้ดซ้ำ
- ป้องกัน Untrusted content ตั้งแต่ฐานราก

Antigravity: Execution & Concurrency

- โครงสร้างแกนกลาง (Capability Substrate)
- บังคับใช้ Optimistic Concurrency Control
- สถาปัตยกรรม Tri-surface บนฐานเดียวกัน

บทสรุป: ใช้ C เป็นมาตรฐานคุณภาพ, M เป็นวินัยย้ายระบบ และ A เป็นบททดสอบความทนทานของ **Execution**

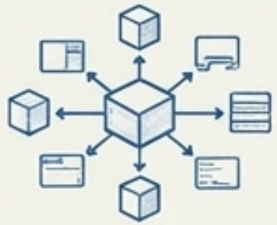
The 8 Core Directives

8 ข้อเสนอชี้ขาดเพื่อเริ่มงาน (The Constitution)



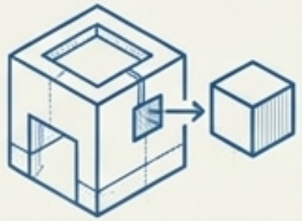
1. Check Before Build

ตรวจ Main และ Deployed build ก่อนสร้างของใหม่เสมอ



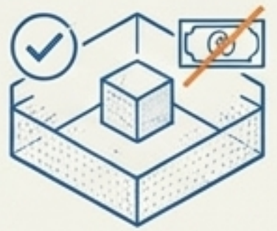
2. Single Source of Logic

Business logic ต้องอยู่ที่ Domain Services ชุดเดียว



3. Preserve AppSpec

รักษาโครงสร้างเดิม เพิ่ม Capability Sidecar เท่าที่จำเป็น



4. Pilot Scope

เริ่มจาก Approval Workflow ใน Sandbox (ไม่ผูกการจ่ายเงิน)



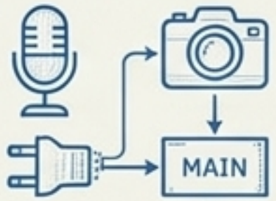
5. Visual Proof First

พิสูจน์ความสวยงามด้วย Output จริง ไม่ต้องรอ Backend เสร็จ



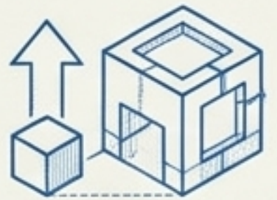
6. Security & Execution First

Authority ต้องมาก่อนเปิดให้ Agent ทำการแก้ไขข้อมูล



7. Adapters Later

Voice/Vision เป็น Input Adapters ทำทีหลังเมื่อระบบนิ่งแล้ว



8. Scale on Readiness

ขยายผลเมื่อมีหลักฐานความพร้อม ไม่บังคับสร้างทุกอย่างพร้อมกัน

Decision register: adopt / adapt / delay / reject

ใช้คำตัดสินทีรวมแล้ว ไม่บวก backlog ของทุก reviewer เข้าด้วยกัน (§5–7)

รับ / Adopt

- Re-baseline และ reuse
- canonical business implementation + projection bindings
- no silent downgrade; authority/evidence; stable IDs/versions

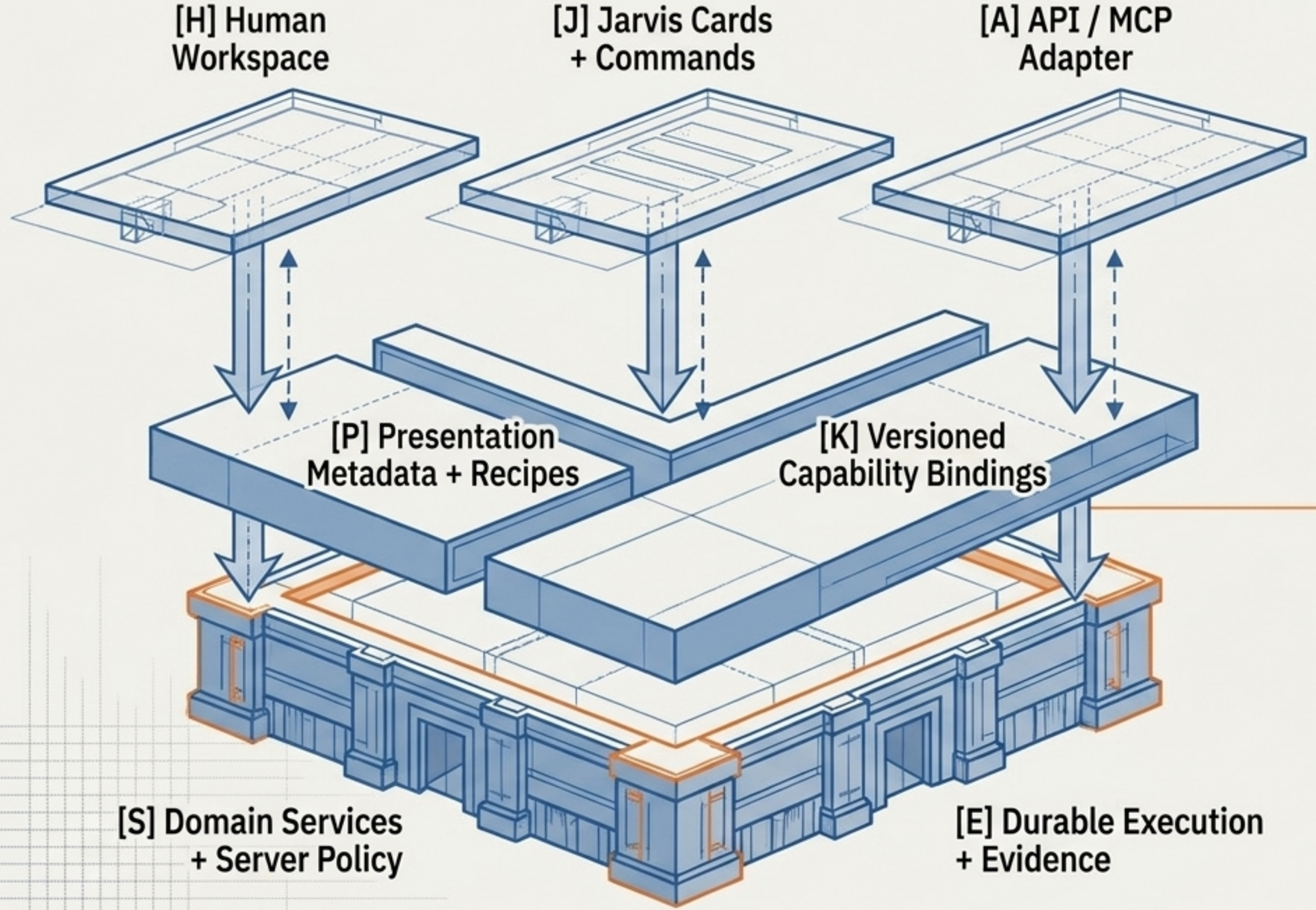
ปรับ / Adapt

- Capability sidecar อ้าง handlers/policy เดิม ไม่เพิ่ม behavior DSL
- version gate ใช้ existing mechanism ก่อนเพิ่ม flag
- table/store/auth patterns ต้องตรวจ audience/tenant/runtime topology

ชะลอ / ไม่รับ

- ชะลอ bespoke UI ทุก entity และ voice/vision expansion ก่อน structured flow ผ่าน
- ไม่ใช้ cache-only เป็น durable execution guarantee
- ไม่บังคับ byte equality ทุก PR; IDs ไม่ใช่ secrets โดยตัวมันเอง ต้องตรวจ authorization จริง

รักษา human-visible quality floor และ security fixes; ไม่สร้าง global token-prefix/transport/database mandates



Single Truth, Multiple Surfaces

ทุกเส้นทาง Mutation ไม่ว่าจะเป็นคนหรือ AI ต้องวิ่งผ่าน Authorization และ Execution Record ของ Server ที่ชั้นล่างสุดเสมอ

Ownership Boundaries (เส้นแบ่งความรับผิดชอบ)

	Owns (รับผิดชอบ)	Must Not Become (ห้ามกลายเป็น)
Domain Services & Policy	กฎเกณฑ์ธุรกิจ, สิทธิ์, Invariants	ห้ามสร้างชุด Business Logic แยกต่างหากสำหรับ Agent
Capability Contract	Identity, IO refs, กฎ Concurrency	ห้ามเป็น Interpreter รัน Rules/SQL แบบอิสระ
Presentation Metadata	การจัดกลุ่ม, UI Roles, Visual states	ห้ามเก็บสำเนา Permission ที่ Server ไม่ได้ตรวจสอบ
Execution Record	หลักฐานถาวร (Evidence), การ Recovery	ห้ามทำตัวเป็นแค่ Response Cache ที่อ้าง exactly-once

The proving grounds: สอง tracks ทีทำคู่ขนาน

Pilot scope ลด blast radius แต่ไม่ทำให้ความเสียงเป็นศูนย์ (§11)

Pilot A · Approval

- bounded expense-claim decision ใน sandbox ไม่เชื่อม payment/bank settlement
- read/prepare → human review → exact approval → server recheck/commit → evidence
- ถ้า service/data/rules ไม่พร้อม ใช้ CRM draft+human-send ตาม discovery decision

Pilot B · POS editor

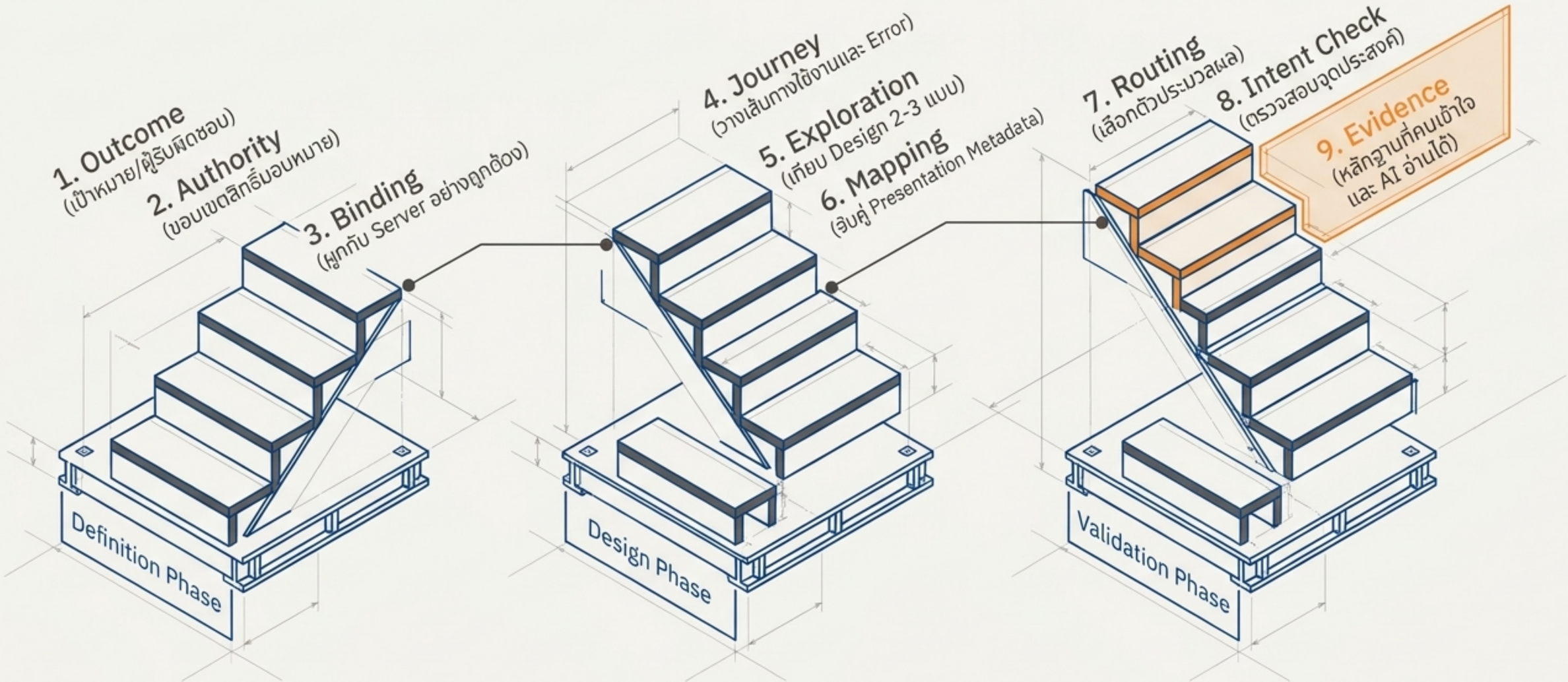
- งานแก้สินค้าเดิม: groups/labels/lookup/money/save/error states
- standalone before/after พิสูจน์ human UX และ regeneration
- ไม่รวม checkout, cash drawer, payment, offline transaction sync หรือ fiscal certification

Pilot C · Cross-domain

- ช่วง 61–90 วัน ถึง 21 ม.ค.2027 เมือ pilot evidence ผ่าน
- Booking/resource reschedule เป็นตัวเลือกตาม service readiness
- ใช้ฐานเดิม; domain invariant กัน booking ต่าง row ชน resource/time ไม่พึ่ง revision เดี่ยว

ตรวจ isolation, fixtures, credentials และ external side effects ของ sandbox; POS editor ไม่พ่วง currency-storage migration

The 9-Step Journey for Every Capability



Human, Jarvis และ machine quality ต่างหน้าที

Jarvis เป็น human-visible surface ส่วน machine-only ไม่ต้อง operational GUI (§10,15)

Human workspace

- ลำดับข้อมูล/งานชัด; meaningful labels/groups; navigation รู้ขั้นต่อไป
- money/date/relations แสดง semantics ไม่โยน raw FK ให้ end user
- a11y/keyboard/mobile + dirty/error/conflict/unknown recovery

Jarvis / Hybrid

- command + plan/decision/result/exception + inspect/takeover
- cards บอก target, before/after, impact, evidence และ commit state
- ผ่าน visual/usability/a11y floor เช่นเดียวกับหน้าที่คนต้องใช้

Machine / Agent API

- typed task operations, scoped discovery, structured errors/results/evidence
- visual=N/A แต่ task/authority/reliability gates ต้องผ่าน
- API อาจรอ human approval ตาม policy; ไม่จำเป็นต้อง autonomous ทุก action

Interaction modes (Human/Hybrid/Agent) แยกจาก generation routes (compiler/kit/AI-custom/hybrid composition)

Backlog priorities ≠ waterfall phases

P0 ปิดความเสี่ยงก่อนเปิด scope; design และ test design ทำคู่ขนานได้ (§12–13)

P0 · FR00/01/02/05/06/07

- FR00 baseline ก่อนงานทีแตะของเดิม
- FR01 pilot/authority/evidence + FR02 compatibility
- FR05 capability bindings; FR06 authority/boundaries; FR07 reliable execution
- FR06 ต้องผ่านก่อนเปิด mutation ไม่ใช่ก่อนเขียน design ทุกชิ้น

P1 · FR03/04/08/09/10

- FR03 design หลัง FR00/01
- FR04 presentation slice หลัง FR02/03
- FR08 thin Jarvis + takeover โดย reuse ของเดิม
- FR09 gates/evidence → FR10 second-domain proof

P2 · FR11/12

- FR11 bounded input adapters หลัง FR06/08–10
- FR12 packs/readiness/SLO หลัง FR10
- FR11 เป็น dependency ของ FR12 เฉพาะ capability ที่ใช้ voice/vision
- ไม่บังคับสร้าง gateway หรือ registry ซ้ำ

Owner เป็น role; effort/capacity ต้อง estimate หลัง discovery ไม่ใช้ model assignments จากรายงานเดิมเป็นข้อผูกมัด

Milestones & exit evidence

สืบกรอบเดิมเริ่ม 23 ต.ค.2026; readiness ต้องมี applicable evidence (§14)

0–30 วัน · 22 พ.ย.26

- FR01–07 ขั้นต่ำ; read/prepare + approved sandbox action ผ่าน API ไม่ใช้ browser
- auth/tenant/stale/retry + compatibility checks
- POS editor standalone before/after

31–90 วัน

- 22 ธ.ค.26: Hybrid E2E + exact approval + takeover + crash/unknown recovery
- 21 ม.ค.27: second-domain proof + cost/latency/reliability baseline
- ไม่ fork auth/execution engine

เดือน 4–10

- 23 เม.ย.27: supervised multistep/ connectors; bounded adapters ตาม need
- 23 ส.ค.27: capability packs + production qualification ตาม declared scope
- security/isolation ขั้นพื้นฐานต้องมาก่อน mutation ไม่รอท้ายโครงการ

ถ้า controls ไม่ผ่าน จำกัด read-only/report gaps และลด breadth ไม่ลด correctness เพื่อให้ตัวเลข milestone ดูครบ

Hard blockers: ครบทั้ง 8 ประเภท

หยุด readiness claim ของ scope ที่เกี่ยวข้อง คะแนน visual/coverage กลบไม่ได้ (§15.1)

Wrong target / authority

- ทำ mutation ผิดเป้าหมายหรือไม่มีสิทธิ์

Tenant leak

- ข้อมูลข้าม tenant โดยไม่มีสิทธิ

Critical data loss

- ข้อมูลสำคัญสูญหาย/เขียนทับ

Duplicate effect

- financial/irreversible side effect ซ้ำ

Misleading success

- แจ้ง final success โดยไม่มี committed evidence

Core action unavailable

- ฟังก์ชันหลักทำงานไม่ได้

Critical controls unusable

- คนใช้ control สำคัญ/เข้าถึงงานไม่ได้

Silent downgrade

- interaction หลักถูกลดทอนโดยไม่มี owner decision

Human-visible usability/visual floor ยังเป็น release criterion; machine-only ใช้ machine gates และ visual=N/A

10 Critical Anti-Patterns

1. อย่ารวม Contract กับ Implementation ไว้ที่เดียวกัน
2. อย่าก๊อปปี้ Sample Schemas มาใช้บน Production ตรงๆ
3. อย่าใช้ Row Revision แทน Invariant ของธุรกิจ
4. อย่าใช้ Redis แทน Execution Ledger ถาวร
5. อย่าซ่อนสถานะ (States) จนความจริง/ปัญหาหายไป
6. อย่า Reuse Database Table ข้าม Product แบบมักง่าย
7. อย่ารีบ Merge Jarvis Gateway โดยไม่ดูระบบเดิมที่มีอยู่
8. อย่าแก้ Emitter ผิด Revision
9. อย่าคิดว่า Source เหมือนเดิมแล้ว Runtime จะเหมือนเดิม
10. อย่าวัด UX แก้สำเร็จจากจำนวนฟิลด์ ต้องเทสต์ใช้งานจริง

Immediate next steps: plan ทีตรวจสอบได้

สิงทีส่งให้ Claude/ทีมก่อนเริม implementation (§17–18)

FR00 baseline

- read-only HEAD/main/deployed/ project artifact/flags inventory
- snapshot99e234985 เป็น historical reference
- reuse/extend/verify/build/obsolete พร้อม evidence ไม่ reset/checkout ทับงานค้าง

FR01/02/03 slice

- freeze bounded approval outcome/ actors/authority/evidence
- compatibility ADR + Studio edits/node ownership + versioned upgrade
- design alternatives + POS visual fixtures ทำคู่ขนาน runtime track

Acceptance & ownership

- แตก FR00–FR12 เป็น tickets พร้อม role owners/dependencies
- calibrate/freeze applicable gates, rubric anchors และ budget ก่อน benchmark
- รายงาน gaps และ effort ranges ตาม capacity จริง; ใช้ Final Review เป็นคำตัดสินหลัก

การปรับเอกสารนีไม่ได้เพิมผลทดสอบหรือรับรอง deploy; เป้าหมายคือ implementation plan ทีไม่สร้างของซำและพิสูจน์ได้

Evidence boundary & readiness

Historical evidence ≠ deployed readiness (§3,14,20)

Baseline ที่ต้องตรวจ

- Final Review อ้าง local origin/main snapshot 99e234985 (22 ต.ค. 2026)
- พบ source ของ recipes, presentation/lookups, Jarvis และ factory agents; อ่านบางส่วน ไม่ใช่ full audit
- FR00 ตรวจ HEAD/main/deployed/artifact/flags ณ วัน implement พร้อม reuse/extend/verify/build/obsolete

เป้าหมาย 23 ส.ค. 2027

- กรอบลงทุน 10 เดือนเดิมเริ่ม 23 ต.ค. 2026 ไม่ reset เมื่อแก้เอกสาร
- พร้อมตาม declared capability scope และ exit evidence ไม่สัญญาทุก feature หรือทำนายตลาด
- Capacity/effort ต้อง estimate หลัง discovery; ลด breadth ก่อนลด correctness

เอกสารอ้างอิง

- Final Review v1.0: docs/appforge/reviews/AppForge_Final_Review_6GL_UX_UI_Agent_Native_2026-10-23.md
- ฉบับนี้เป็น planning/learning derivative ไม่ใช่ผล UAT หรือ production signoff
- Source presence, emitted behavior, deployed availability และ verified UX เป็นคนละสถานะ

วันที่ปรับเอกสาร 25 ต.ค. 2026 ไม่ใช่วันที่ audit codebase/deployment ใหม่

Journey, execution และ autonomy

แยกสามเรืองนีก่อนออกแบบ contract (§6 D07,8,11,15–16)

Journey เป็นลำดับงาน

- Read/context → Prepare → Human review → Exact approval → Server recheck/commit → Result/evidence
- เป็น flow ของ pilot A ไม่ใช่ชื่อ execution-state enum
- Cancel/stop/takeover ไม่เท่ากับ rollback สิ่งที่ commit แล้ว

Lifecycle เป็นสถานะ runtime

- Candidate: queued / running / awaiting_approval / succeeded / failed / cancelled
- เลือกตาม consumer และ runtime เดิม; เก็บ unknown/reconciling knowledge กับ partial child steps ได้
- ห้ามให้ parent failed หรือ network timeout กลบผลที่เกิดจริง; reconcile ก่อน blind retry

Autonomy ภายใต้ delegation

- Pilot เริม agent read/prepare และ human approval ตาม policy
- ระยะถัดไป agent execute/decide ได้ใน delegated scope ไม่ต้อง confirm ทุก tool call
- Agent ห้าม mint approval/ขยายสิทธิ์เอง; ASR/OCR/content ไม่ใช่ authority; takeover ใช้ safe boundary/fencing

Ledger อย่างเดียวไม่รับประกัน exactly-once ข้ามทุก external system ต้องทดสอบ crash window และ recovery ตาม operation

Evaluation & implementation handoff

หลักฐานที่ใช้เทียบคุณภาพและเปิด scope (§13–18)

Fair benchmark

- A: brief/data/functional scope เดียว เทียบ generation routes; B: design intent เดียว เทียบ preservation
- แยก visual-only prototype กับ functional output; blind labels; freeze versions/attempts/repair budget
- วัด total cost ต่อ accepted outcome รวม repair/review/runtime/maintenance

Applicable gates

- FT01–FT12 เป็น scenario intents ยังไม่ใช่ suite ที่รันผ่าน; FT11 ใช้กับ booking/resource conflict
- 80/100, comprehension ≥90%, 50 tasks เป็น proposed examples ที่ต้อง calibrate/freeze ต่อ pilot
- Visual repair ≤2 รอบเป็น budget proposal ไม่ใช่ cap ของ correctness/security repair

ส่งต่อทีม implement

- FR00 → bounded pilot + compatibility ADR → vertical slice; ระบุ files/owners/dependencies/evidence
- POS screenshot เดิม: docs/appforge/references/pos-2026-10-22/pos-edit-product-studio.png
- ใช้ภาพเดิมอธิบายอาการ; เก็บ standalone before/after ตาม revision/viewport เพื่อพิสูจน์ผล ไม่เดา emitter bug จากภาพ

ไช้ Final Review เป็นคำตัดสินหลัก; เอกสารชุดนี้ไม่เพิ่ม global HTTP/token/storage mandates หรือรับรอง readiness โดยตัวเอง