

تقرير التحليل المعماري واستراتيجية التنفيذ لمنصة Delivery Ways: مراجعة خطة Manus والنماذج المرنة "Uber" معايير

الملخص التنفيذي والسياق الاستراتيجي 1.

تحول النموذج الهندسي: من الكود اليدوي إلى التوليد المعزز بالذكاء الاصطناعي 1.1

يشهد المشهد الهندسي للبرمجيات حاليًا تحولًا زلزاليًا، يتميز بالانتقال من كتابة الأكواد البرمجية يدويًا بشكل كامل إلى توليد البنية التحتية كما هو محدد في التذكرة، (DW) "Delivery Ways" المعمارية بمساعدة وكلاء الذكاء الاصطناعي المتقدمين. يمثل مشروع وهو وكيل ذكاء اصطناعي - "Manus" مثالًا جوهريًا لهذا النموذج الحديث. من خلال الاستفادة من، "ANALYSIS-PLAN-001" (Prototype) عالي القدرة مصمم لتنفيذ المهام المعقدة وتوليد الأكواد¹ - يهدف المشروع إلى تسريع مساره من مرحلة النموذج الأولي "Uber" إلى منصة جاهزة للإنتاج بمستوى يضاهي.

يقدم مجموعة فريدة من التحديات المتعلقة بحوكمة الكود، والنزاهة Manus ومع ذلك، فإن استخدام وكلاء الذكاء الاصطناعي مثل (Refactoring) على عكس إعادة الهيكلة التي يقوم بها البشر. (Legacy Systems) المعمارية، والتكامل مع الأنظمة القديمة على حساب (Localized Optimization) غالبًا ما تعطي الخطط التي يولدها الذكاء الاصطناعي الأولوية للسرعة والتحسين المحلي المتضمنة في Manus إن تحليل خطة. (Hard Constraints) "الاتساق النظامي الشامل، ما لم يتم تقييدها بصرامة بـ" قيود صلبة ليس مجرد مراجعة للكود؛ بل هو تدقيق معماري Delivery Ways - Project Development Plan.zip الملف المضغوط لمخرجات وكيل مستقل مقابل المعايير الصارمة لبرمجيات التنقل المؤسسية.

وتقييمها مقابل الوضع، (Tracks A, B, C, D) يقدم هذا التقرير المخرجات الشاملة المطلوبة، حيث يقوم بتفكيك المسارات المقترحة "Shim" وتجميع خارطة طريق تحترم القيود غير القابلة للتفاوض للمشروع: معماريات، DW_workspace الحالي لمساحة العمل مباشرة في طبقة التطبيق، والحفاظ على مجموعة الاختبارات الحالية SDK الصارمة، منع استيراد حزم.

تعريف المعيار "Uber": طموح الوصول لمستوى 1.2

يعني الالتزام بمجموعة محددة من المتطلبات غير الوظيفية التي تتجاوز مجرد تحسين "Uber"³ إن الطموح للوصول إلى مستوى يعني هذا، Delivery Ways واجهة المستخدم. في سياق مشروع

1. القدرة على التعامل مع قطاعات خدمات متنوعة (نقل الركاب، الطعام، **(Operational Flexibility) المرونة التشغيلية**) الذين يجادلون،⁴ Rungtusanatham et al. دون هشاشة معمارية. يعكس هذا النتائج الأكاديمية التي توصل إليها بأن البساطة المنضبطة في العمليات الأساسية هي شرط أساسي للمرونة التشغيلية.
2. يجب أن تدير المنصة الحالة **(Event-Driven Real-Time Consistency) الاتساق الآني المعتمد على الأحداث** يقاس بالمللي ثانية (Latency) (موقع السائق، حالة الطلب) عبر عملاء موزعين بوقت استجابة دون المخاطرة بحدوث تراجع في "Ride" القدرة على نشر تحديثات لقطاع **(Modular Isolation) العزل المعياري** يرتبط هذا بمفهوم الوحدات في معماريات عائلات المنتجات⁵، حيث تتفاعل الوحدات المتميزة من خلال واجهات "Parcels". "Shims"، صارمة - في هذه الحالة

1.3 منهجية التحليل: تثليث البيانات

الصارمة. يتم التثليث بين ثلاث نقاط بيانات متميزة لاستخلاص الرؤى (Gap Analysis) "يعتمد هذا التقرير منهجية" تحليل الفجوة

- D. إلى A مقسمة إلى مسارات من Manus، الحالة المثالية التي اقترحها وكيل الذكاء الاصطناعي **(Manus) النية (خطة)**
- الحالة الحالية للكود، والتي تتميز بتنفيذات جزئية وديون تقنية موروثة **(DW_workspace) الواقع**
- التي تعمل (Clean Architecture, Banned Imports) **(المعيار (القيود الصلبة): القواعد الهندسية غير القابلة للتغيير** كمعايير قبول لأي كود يتم دمجها في المستودع

2. الإطار النظري: المرونة الصارمة وهندسة الوكلاء المستقلين

2.1 كأساس معماري "Rigid Flexibility" نموذج

"Rigid" إلى ما يُعرف في أدبيات إدارة العمليات بنموذج Delivery Ways تستند الفلسفة المعمارية التي يجب أن تحكم مشروع في العمليات الأساسية لا (Discipline) يقترح هذا النموذج، الذي أعيد فحصه تجريبيًا، أن الفرض الصارم للانضباط⁴. "Rigid Flexibility" يحد من المرونة، بل على العكس، هو الممكن الرئيسي لها.

Super-Apps في سياق هندسة البرمجيات لتطبيقات الـ

- تتمثل في القواعد التي تمنع المطورين (أو وكلاء الذكاء الاصطناعي) من استيراد مكتبات خارجية **(Rigidity) الصرامة** هذا القيد يبدو معيّنًا للسرعة في البداية. (App Layer) مباشرة داخل طبقة التطبيق

- أو (Mapbox إلى Google Maps تظهر عندما تقرر الشركة استبدال مزود الخرائط (مثلاً من **(Flexibility) المرونة** يمكن إجراء هذا التغيير الجذري دون إعادة كتابة "Shims" تغيير مزود الدفع. بفضل الصرامة الأولية في استخدام طبقة العمل (Business Logic) سطر واحد في منطق العمل).

أي القدرة - (Mix Flexibility) ومرونة المزيج (Volume Flexibility) تشير الأبحاث إلى أن محاولة تحقيق المرونة في الحجم على تقديم خدمات متنوعة مثل الركاب والطرود - تتطلب استراتيجيات تصنيع (أو برمجة) تفصل بين المكونات الثابتة والمكونات يجب أن تُقيم بناءً على مدى التزامها بهذا الفصل، وليس فقط بناءً على سرعة الإنجاز Manus المتغيرة.⁶ وبالتالي، فإن خطة

2.2 في التنفيذ (Manus) دور وكيل الذكاء الاصطناعي

يتجاوز مجرد تقديم الإجابات النصية إلى تنفيذ المهام المعقدة (Action Engine) وفقاً للمواد البحثية¹، محركاً للفعل Manus، يُعد إلى تحديات محددة⁸ Manus الخاصة بـ (Context Engineering) "وأتمتة سير العمل. ومع ذلك، تشير أدبيات "هندسة السياق

- (Stable) على سياق متراكم. إذا لم يتم تغذيته بـ "القيود الصلبة" بشكل متكرر ومستقر Manus نافذة السياق والذاكرة: يعتمد فإنه قد "ينسى" القواعد المعمارية لصالح الحلول الأسرع والأكثر شيوعاً في بيانات تدريبه (والتي غالباً ما تكون أكواداً، Prefix) (Spaghetti Code) غير معيارية أو
- إلى المخرجات (Input Tokens) نسبة المدخلات إلى المخرجات: في الوكلاء المستقلين، تكون نسبة الرموز المميزة للمدخلات عالية جداً (100:1). هذا يعني أن تكلفة "تذكير" الوكيل بالقواعد المعمارية عالية، ولكنها ضرورية لتجنب الديون التقنية
- ليس كوصفة نهائية، بل كـ "مسودة أولية" تتطلب مراجعة بشرية صارمة لضمان أنها لم Manus الاستنتاج: يجب معاملة خطة "تتجاوز حدود" الصرامة المطلوبة لتحقيق "المرونة".

3. البنية التحتية ونظام التصميم: (Track A) تحليل المسار أ.

3.1 أكثر من مجرد واجهة: (App Shell) قشرة التطبيق

يعتبر هذا المسار هو Design System & App Shell Implementation على (A) يركز المسار Manus، في خطة لا تعمل "قشرة التطبيق" مجرد حاوية للتنقل، بل هي "نظام التشغيل، Super-Apps الأساس الحرج للمشروع بأكمله. في تطبيقات الـ (Verticals) المصغر " الذي يدير التطبيقات الفرعية

تحليل الفجوة بين الخطة والواقع:

- تشير البيانات إلى أن نظام التصميم "مفعل جزئياً". هذا يعني وجود حالة خطيرة تعرف بـ DW_workspace الواقع في (Hard-coded) حيث توجد مكونات قديمة تعتمد على قيم لونية ومسافات ثابتة، (Design Drift) "انحراف التصميم ومكونات جديدة تحاول استخدام المتغيرات
- Manus طموح (Production-Ready Implementation) "يهدف المسار إلى" تطبيق جاهز للإنتاج

- يجب أن تكون معزولة تمامًا عن منطق الأعمال، (Rides, Food) الرؤية المعمارية: لكي تنجح القشرة في دعم مسارات متعددة الخاص بتلك المسارات. يجب أن لا تعرف القشرة "كيف" يتم حجز رحلة، بل تعرف فقط "كيف" تعرض واجهة الرحلات.

3.2 SDKs وحظر الـ Shims معضلة الـ

(Ports and Adapters) "هذا نمط معماري يُعرف بـ "المنافذ والمحولات". "No SDKs in app/lib" أحد أهم القيود المذكورة في التذكرة هو. أو العمارة النظيفة (Adapters)

3.2.1 الضرورة النظرية والعملية

مباشرة في واجهات المستخدم. هذا يخلق اقتراءً وثيقاً package:firebase_auth التقليدي، من الشائع استيراد Flutter في تطوير (Tight Coupling). تغيير مزود المصادقة، فإن التطبيق بأكمله يحتاج إلى إعادة كتابة Uber إذا قررت.

Shim Layer: تقترح الخطة (ويجب أن تفرض) طبقة

- (Abstract Classes) تُعرف واجهات. Flutter النقي وواجهات Dart تحتوي فقط على منطق (app/lib): طبقة التطبيق (مثل IMapProvider) لما تحتاجه (مثل).
- فعلية SDKs تنفذ هذه الواجهات باستخدام (packages/shims): طبقة المحولات.

مخاطر التنفيذ عبر الذكاء الاصطناعي:

(Leaky) "في سعيه لسرعة الإكمال، قد يولد "تجريدات مسربة، Manus هو أن وكيل (A) الخطر الرئيسي في المسار (Abstractions). فإن التجريد، GoogleMapController تحتوي على دالة تُرجع IMapProvider مثال: إذا كانت واجهة. (moveCamera(TargetLocation)) وهمي وفاشل. يجب أن تكون الواجهة محايدة تمامًا (مثلاً)

3.3 نظام التصميم: الرموز مقابل القيم الثابتة

"Design Tokens" على نظام (A) في التناسق البصري، يجب أن يعتمد المسار Uber للوصول إلى مستوى

- يجب أن (C و B) في المسارات اللاحقة Manus التوصية: يجب أن يتضمن تقرير المراجعة بنذًا يفرض أن أي كود يولده context.theme.primaryBrandColor بل يجب استخدام Colors.blue، يستهلك هذه الرموز. لا يُقبل

4. (Ride Vertical) عمود التنقل والرحلات: (Track B) تحليل المسار ب

4.1 التنقل "MVP" التعقيد الكامن في

الحد الأدنى للمنتج القابل للتطبيق)) MVP في عالم تطبيقات التنقل، مصطلح "Ride Vertical MVP" تنفيذ (B) يتناول المسار "مضلل. التطبيق الذي يفشل في تحديد الموقع بدقة أو حساب السعر لا يعتبر "قابلاً للتطبيق

4.2 آلة الحالة المحدودة (Finite State Machine - FSM)

جوهر تطبيق التنقل هو "آلة حالة" معقدة تدير دورة حياة الرحلة:

Draft -> Quoting -> Requesting -> Finding Driver -> Driver Accepted -> Driver Arrived -> In Progress -> Payment -> Complete.

تحليل الفجوة:

- **الوضع الحالي:** تشير المدخلات إلى وجود "واجهة مستخدم بسيطة + هيكل تتبع". هذا يعني أن المكونات المرئية موجودة، لكن هش أو مفقود (State Machine) "المنطق" الذي يربطها بآلة الحالة.
- حيث الشبكة متصلة دائماً والسائق يقبل الطلب (Happy Path) "غالباً" المسار السعيد Manus يفترض **Manus** افتراضات فوراً.
- وليس في واجهات المستخدم (Domain Layer) **التحدي الهندسي:** يجب أن يكون منطق إدارة الحالة موجوداً في طبقة المجال (Unit Tests)، هذا يسمح باختبار الحالات المعقدة (مثل انقطاع الشبكة أثناء الرحلة) عبر اختبارات الوحدة (UI Widgets)، "مما يساهم في الحفاظ على هدف "158 اختباراً".

4.3 Shims تحدي الخرائط الجغرافية وتكامل الـ

(A) التي تم تأسيسها في المسار Shims يعتبر التعامل مع الخرائط هو الاختبار الحقيقي لمعمارية الـ

- تنفيذ خريطة تفاعلية عبر طبقة عزل هو أمر صعب تقنياً. الخرائط بطبيعتها أدوات مرئية وتفاعلية **Map Shim مشكلة**
- للأحداث (مثل النقر على الخريطة) ومصارف (Streams) واجهة تعتمد على التدفقات Shim **الحل المقترح:** يجب أن يقدم الـ هو الوحيد الذي يضمن فصلاً حقيقياً (Reactive Pattern) للأوامر (مثل تحريك الكاميرا). هذا النمط التفاعلي (Sinks)

4.4 محركات التسعير (Pricing Engines)

من الخادم (Quote) "يتم طلب" عرض سعر (Client-side) لا تحسب الأسعار على العميل Uber تطبيقات مستوى

- تحاكي منطق "Stub Service" إنشاء (B) جاهزاً، يجب أن يتضمن المسار (Backend) **الفجوة**: إذا لم يكن الخادم الخلفي "التسعير وتأخير الشبكة". هذا ضروري لاختبار تجربة المستخدم في حالات "ارتفاع الطلب" أو "فشل الاتصال".

5. (Parcels & Food) التوسع اللوجستي: تحليل المسار ج

5.1 (Domain Separation) الوحدات النمطية وفصل المجالات

من المغري إعادة استخدام (Code Contamination) "خدمات" الطرود والطعام". الخطر هنا هو "تلوث الكود (C) يقدم المسار لكنهما مختلفان جوهرياً. الرحلة لها "ركاب"، والطرود لها "أبعاد" و"تعليمات"، ParcelDelivery لتمثيل RideRequest كائن "استلام".

(Mix Flexibility) رؤية نظرية: مرونة المزيج

بالعودة إلى المراجع 5، تعني مرونة المزيج قدرة النظام على التبديل بين أنواع مختلفة من المنتجات بأقل تكلفة

- (Null Fields) واحدة مليئة بالحقول الفارغة Order **العمارة السينية**: فئة
- قد تشترك في (RideOrder, FoodOrder, ParcelOrder) كيانات مجال متميزة: **Manus** **العمارة المستهدفة (هدف)** لسجل الطلبات، ولكنها تظل منفصلة في تدفق العمل النشط (BaseOrder) سمات خفيفة

5.2 (Super App Dashboard) لوحة تحكم التطبيق الشامل

الشاشة الرئيسية تحتاج إلى توجيه المستخدمين بذكاء إلى هذه المسارات

- (Demo Screens) "الواقع": شاشات تجريبية
- **Manus** خطة تنفيذ المنطق
- هذا يتطلب (C) يجب أن تكون متاحة في المسار (B) **الفجوة**: يجب معالجة "الموارد المشتركة". طريقة الدفع المختارة في المسار تقع تحت التطبيقات العمودية (Horizontal Module) أن تكون ميزة "الدفع" وحدة مشتركة

6. (Auth & Onboarding) بوابات العبور والهوية الرقمية: تحليل المسار د

6.1 إدارة الهوية كطبقة عزل (Identity Shim)

AWS أو Firebase Auth هي أهم طبقة عزل. يجب ألا يعرف التطبيق ما إذا كان يستخدم (Authentication) المصادقة Cognito.

- (DSR) نطاق المسار: التسجيل، المصادقة، وطلب خدمة السائق.
- الخطر: الوضع الحالي يشير إلى أنها "موجودة جزئيًا في الكود/الدراسات". هذا هو أخطر وضع، حيث أن كود المصادقة نصف المكتوب هو ثغرة أمنية.
- لأنه لا يمكن، (A) أو حتى بالتوازي مع المسار (C) ليكون قبل المسار (D) استراتيجية التنفيذ: يقترح التقرير نقل تنفيذ المسار (User ID) بدون معرف مستخدم (B حجز رحلة (المسار

6.2 والتحقق (DSR) طلب خدمة السائق

(تتضمن هذه الميزة تدفقات معقدة لرفع المستندات (رخصة القيادة، صور السيارة

- (التعامل مع الوسائط: يتطلب رفع الصور التعامل مع الأذونات (الكاميرا/المعصر
- في permission_handler أي استدعاء مباشر لمكتبة PermissionShim. فحص القيود: يجب التعامل مع الأذونات عبر واجهة المستخدم يعد خرقًا للمعمارية
- حيث تطلب المعلومات، (Progressive Profiling) "التميط التدريجي" Uber معيار تجربة المستخدم: تستخدم تطبيقات للتأكد من اتباع هذا النمط Manus على مراحل. يجب مراجعة خطة

7. تحليل الفجوات الشامل (Comprehensive Gap Analysis)

والواقع، وتُستخدم لتوليد ملف Manus تهدف الجداول التالية إلى تلخيص الفجوات بين خطة docs/analysis/manus_tracks_gap_analysis.md.

7.1 جدول مقارنة المسار A (Design & Shell)

المعيار	Manus افتراضات	واقع DW_workspace	(Gap) الفجوة	الإجراء التصحيحي

نظام التصميم	نظام مكتمل وجاهز للاستخدام	مفعل جزئيًا، خليط من القديم والجديد	عدم وجود استراتيجية لتحويل الشاشات القديمة	إضافة مهمة "Migration Sprint" لتحويل الشاشات القديمة
الهيكلية (Navigation)	استخدام go_router أو Shell مشابه مع Route	كود ملاحظة قديم (Legacy Navigator)	تعارض في إدارة الـ Deep Links وحالة القشرة	إعادة كتابة RouterDelegate في Track A
الاستقلالية	القشرة لا تعتمد على التفاصيل	القشرة قد تكون حاليًا Auth مقترنة بـ	القشرة تحتاج لمعرفة "حالة الدخول" لعرض القوائم	فصل AuthStatusStream كواجهة وحيدة للقشرة

7.2 جدول مقارنة المسار B (Ride Vertical)

المعيار	Manus افتراضات	واقع DW_workspace	(Gap) الفجوة	الإجراء التصحيحي
إدارة الحالة (State)	(Happy Path) تدفق سعيد	واجهة مستخدم بسيطة دون منطق ربط	غياب التعامل مع حالات الحافة (Edge Cases) وانقطاع الشبكة	فرض اختبارات "Chaos Testing" على الـ FSM
(Maps) الخرائط	(Plug & Play) تكامل جاهز	Map لا يوجد حقيقي Shim	الخرائط الحالية قد تكون Hard-coded	بناء MapInterface UI قبل أي كود
(Pricing) التسعير	واجهة برمجة (API) تطبيقات جاهزة	غير معروف / غير موجود	التطبيق سيفشل عند محاولة جلب السعر	بناء MockPricingService داخل Dart لمحاكاة الخادم

8. توصيات التنفيذ وخارطة الطريق (Execution Roadmap)

والحاجة إلى ضوابط بشرية، نوصي بتعديل استراتيجية التنفيذ Manus بناءً على التحليل أعلاه، وبالنظر إلى الطبيعة "المستقلة" لوكيل

8.1 ترتيب التنفيذ المقترح (Re-sequencing)

بدون "مستخدم (B) "الذي تقترحه الخطة يحمل مخاطر عالية. لا يمكن اختبار "حز الرحلة (A -> B -> C -> D) الترتيب الخطي (D) "مسجل

الترتيب الموصى به:

1. **الأساس):** بناء القشرة ونظام التصميم. هذا شرط مسبق لا يمكن تجاوزه (Track A).
2. **مستقبلي API حقيقي للتعامل مع أي AuthToken الهوية - مُقدم):** تنفيذ المصادقة الأساسية. نحتاج إلى (Track D).
3. **تنفيذ القيمة الأساسية للمشروع (Rides - جوهر العمل):** Track B.
4. **يمكن تأجيله أو تنفيذه بالتوازي إذا توفرت الموارد (Logistics - التوسع):** Track C.

8.2 إدارة المخاطر والأسئلة المفتوحة

manus_execution_recommendations.md) للتنفيذ، يجب حسم النقاط التالية (لتضمنها في Manus قبل إطلاق

- **مخاطر "الديون الخفية" للذكاء الاصطناعي:**
 - للاستيراد aliasing عن طريق عمل "No SDKs" بتوليد كود يعمل ظاهرياً ولكنه يخرق قاعدة Manus الخطر: قد يقوم أو وضعه في ملفات غير مراقبة
 - عند اكتشاف أي (Build Failure) ليرفض البناء analysis_options.yaml في ملف custom_lint الحل: تكوين استيراد محظور، مما يجبر الوكيل على التصحيح الذاتي
- **Backend: سؤال حول جاهزية الـ**
 - بكتابة Manus جاهزة؟ إذا كانت الإجابة "لا"، يجب تكليف RideMatching لخدمة WebSocket هل واجهات الـ B. كجزء من المسار Dart بلغة (Mock Server) "" خادم محاكاة
- **(Feature Flags): استراتيجية الميزات**
 - التكوين (Track A) "في الإطلاق الأول؟ إذا نعم، يجب أن تدعم "القشرة (Flag) خلف ميزة "Parcels" هل نريد إخفاء لإخفاء أيقونة الطرود (Remote Config) عن بعد

8.3 التوافق مع المعايير الداخلية

"Clean B Workspace": تضمن هذه الخطة المعدلة الالتزام بمعايير

- فقط: يتم عزل كل مكتبة خارجية **Shims** التكامل عبر الـ
- يقلل النسبة يتم رفضه PR وليس السقف. أي (Floor) الحفاظ على الاختبارات: يتم اعتبار رقم "158 اختباراً" هو الأرضية
- لإدارة الحزم المتعددة، مما يفرض الفصل المادي بين الكود melos نظافة المستودع: يتم استخدام

9. الخاتمة

باستخدام قدرات الذكاء الاصطناعي المتقدمة "Uber" فرصة فريدة للقفز نحو مستوى "Delivery Ways" يواجه مشروع ومع ذلك، يُظهر هذا التحليل أن الاعتماد الأعمى على الخطة المقترحة دون حوكمة معمارية سيؤدي إلى منتج "سريع" (Manus).
"الانكسار" بدلاً من "مرن".

وفصل المجالات - هو الطريق الوحيد Shims من خلال فرض قيود صارمة على الـ - "Rigid Flexibility" إن تطبيق نموذج (D) لا تتحول إلى ديون تقنية طويلة الأمد. من خلال إعادة ترتيب المسارات لتقديم المصادقة Manus لضمان أن السرعة التي يوفرها وتوضيح استراتيجية محاكاة الخدمات الخلفية، يمكن للفريق تحويل هذه الخطة النظرية إلى واقع برمجي قوي وقابل للتوسع.

(Simulated Output Files) الملحقات: محتوى ملفات المخرجات

docs/analysis/ جاهز للحفظ في Markdown فيما يلي المحتوى التفصيلي للملفات الثلاثة المطلوبة، مصاعاً بتنسيق

الملف الأول: docs/analysis/manus_plan_overview.md

Delivery لمشروع Manus نظرة عامة على خطة تطوير Ways

الملخص التنفيذي

من نموذج أولي إلى منصة تنقل متعددة الخدمات جاهزة DW_workspace التطويرية إلى تحويل مساحة العمل Manus تهدف خطة مع التركيز على إنشاء قشرة تطبيق، (A, B, C, D) تقوم الخطة بتفكيك العمل إلى أربعة مسارات رئيسية. ("Uber للإنتاج" مستوى صلبة ونظام تصميم قبل التوسع في التطبيقات العمودية المعقدة. الخطة طموحة وتفترض جاهزية عالية للخدمات الخلفية (App Shell) وتتطلب التزاماً صارماً بالقيود المعمارية لضمان الجودة، (Backend).

(Tracks Breakdown) هيكلية المسارات

المسار (Track)	العنوان في الخطة	الهدف الاستراتيجي	أهم المخرجات (Deliverables)
A	Design System & App Shell	تأسيس البنية التحتية المرئية والملاحية.	<ul style="list-style-type: none">قشرة تطبيق جاهزة للإنتاج. ((ملاحظة، توجيهنظام تصميم قائم على الرموز (Tokens)محرك الثيمات (Theme Engine)
B	Ride Vertical MVP	تنفيذ الوظائف الأساسية لخدمة النقل.	<ul style="list-style-type: none">(FSM) آلة حالة الرحلةShim تكامل خرائط عبرواجهة التسعير والحجز
C	Parcels & Food MVP	توسيع المنصة لتشمل الخدمات اللوجستية.	<ul style="list-style-type: none">معالجات نماذج الطرود (Wizards)واجهة كتالوج الطعاملوحة تحكم متعددة الخدمات
D	Onboarding, Auth,	إدارة الهوية وجانب العرض	<ul style="list-style-type: none">مصادقة آمن Shim (OIDC/OAuth)

	DSR	((السائقين)).	<ul style="list-style-type: none"> تدفق طلب خدمة السائق (DSR) إدارة الملف الشخصي
--	-----	---------------	--

القيود الصلبة (Hard Constraints)

تم تحديد القيود التالية كإجابات لا يمكن تجاوزها لضمان نجاح المشروع

- **غير قابلة للتفاوض Shims عمارة الـ:**
 - داخل مجلد (Firebase, Google Maps, Stripe) القيد: يمنع منعًا باتًا استيراد حزم البنية التحتية الخارجية (مثل app/lib).
 - معرفة في طبقة المجال (Interfaces) المتطلب: يجب أن تمر جميع التفاعلات مع الطرف الثالث عبر واجهات مجردة packages/shims ومطبقة في (Domain).
- **خط الأساس للاختبارات:**
 - القيد: يجب أن تنتج مجموعة الاختبارات الحالية (158 اختبارًا) في جميع الأوقات.
 - يجب أن تحافظ على نسبة التغطية أو تزيدها (Tracks B/C) المتطلب: الميزات الجديدة.
- **الصارمة Linter قواعد:**
 - عند وجود استيرادات محظورة (Errors) لإطلاق أخطاء analysis_options.yaml القيد: يجب تكوين.
- **صحة البناء (Build Health):**
 - المستقرة دون الاعتماد على مسارات محلية مطلقة Flutter القيد: يجب أن يتم بناء المشروع على قنوات.

مع الواقع الحالي v3.2.1 مقارنة حالة المشروع

- منفذة جزئيًا " (RBAC) إلى أن "الإشعارات" و"التحكم في الوصول v3.2.1 نطاق التقرير: يشير تقرير
- **التعارض:** مساحة العمل الحالية تحتوي على هذه الميزات في حالة "مفاهيمية" ولكنها غير مدمجة بالكامل في قشرة التطبيق الجديدة A. المقترحة في المسار
- الرحلات) بشكل كبير على الإشعارات لتحديثات الحالة (مثل "وصل السائق"). إذا لم تكن الإشعارات B **التبعية:** يعتمد المسار B. سيفشل المسار، Shim مغلفة بطبقة

الملف الثاني: docs/analysis/manus_tracks_gap_analysis.md

Manus تحليل الفجوات التفصيلي لمسارات

DW_workspace والواقع الفعلي لمستودع Manus يقارن هذا المستند بين الافتراضات الواردة في خطة

Track A: Design System & App Shell Implementation

- متعددة الخدمات، مدعومة بنظام "Super App" الجذري ومنطق التنقل لدعم هيكلية MaterialApp إعادة بناء **TL;DR:** تصميم قائم على الرموز.
- **Manus (What Manus Assumes):** افتراضات
1. بسهولة دون كسر جميع الشاشات الحالية (Root Widget) أنه يمكننا إعادة هيكلة عنصر واجهة المستخدم الجذري.
2. (Tokens) وليس نموذج حوكمة للرموز، Widgets أن "نظام التصميم" هو مجرد مجموعة من الـ.
- **(DW_workspace Reality):** الواقع الحالي
1. (Hard-coded) نظام تصميم جزئي حيث تستخدم بعض العناصر الرموز والبعض الآخر يستخدم ألوانًا ثابتة.
2. أو بنية القشرة المقترحة go_router مكدس تنقل قديم قد لا يكون متوافقًا مع.
- **(Gaps & Contradictions):** الفجوات والتناقضات
○ لأكثر من 50 شاشة موجودة. إذا قام (Migration Strategy) "الفجوة": لا تذكر الخطة صراحة "استراتيجية الترحيل (Franken-UI) ببناء مكونات جديدة فقط، فسننتهي بواجهة مستخدم هجينة A المسار والوضع الليلي، لكن العينات المقدمة (Accessibility) التناقض: مصطلح "جاهز للإنتاج" يعني دعم إمكانية الوصول وتركز فقط على الشكل الجمالي.
- **عمل ثقيل vs مكاسب سريعة:**
○ ورموز الألوان فورًا ThemeData مكاسب سريع: تعريف.
○ عمل ثقيل: إعادة هيكلة منطق التنقل لفصل القشرة تمامًا عن التطبيقات العمودية.

Track B: Ride Vertical MVP Implementation

- **TL;DR:** منطق العمل الأساسي. حجز رحلة، تتبع السائق، وإدارة جلسة الرحلة.
- **Manus:** افتراضات
1. والتحكم بالكاميرا (Polylines) والخطوط (Markers) يعمل ويدعم العلامات "Map Shim" وجود.
2. WebSocket/Stream يدفع تحديثات "حالة الرحلة" عبر backend وجود.
- **الواقع الحالي:**
1. "شاشات واجهة مستخدم بسيطة لعملية" الحجز.
2. (Mock) من المحتمل أن يكون ثابتًا أو يعتمد على بيانات وهمية (Tracking) "هيكل" تتبع.

- 3. (مباشرة في الكود القديم Google Maps Widget بمستوى الإنتاج (غالبًا ما يتم استخدام Map Shim لا يوجد).
- **الفجوات والتناقضات:**
 - يحتاج التطبيق للتعامل مع أحداث "انتهاء صلاحية السعر" أو "زيادة" (Quote Engine) "الفجوة: منطق "محرك التسعير Uber. الأسعار". الخطة الحالية تشير لتدقيق بسيط "عرض السعر - حجز"، وهو ساذج جدًا لمستوى (Rehydration) "إذا تم إغلاق التطبيق أثناء الرحلة، فإن منطق "إعادة التميؤ" (State Loss) الخطر: فقدان الحالة MVP. لحالة الرحلة من الخادم عند بدء التشغيل معقد وغالبًا ما يتم تجاهله في خطط ال
- **عمل ثقيل vs مكاسب سريعة:**
 - باستخدام نظام التصميم الجديد (Booking Sheet) "مكسب سريع: تحسين واجهة "ورقة الحجز
 - الذي يتعامل مع أكثر من 15 حالة لدورة حياة الرحلة بقوة RideBloc و RideRepository عمل ثقيل: بناء

Track C: Parcels & Food MVP Implementation

- **TL;DR:** إضافة القدرات اللوجستية والتجارية.
- **Manus:** افتراضات
 1. (يمكن للطرود والطعام مشاركة نفس هيكل "الطلب" (افتراض خطير
 2. مطابق للرحلات (Checkout) تدفق الدفع
- **الواقع الحالي:**
 1. للطرود (Demo) شاشات تجريبية
- **الفجوات والتناقضات:**
 - للطعام. الرحلات ليس لديها سلة؛ الطعام لديه. هذا يتطلب آلية تخزين (Basket Management) "الفجوة: إدارة السلة "محلية جديدة عامة لـ "السلة"
- **عمل ثقيل vs مكاسب سريعة:**
 - C. لعنوان التسليم في المسار B من المسار (Location Picker) "مكسب سريع: إعادة استخدام "منتقي الموقع"

Track D: Onboarding, Auth, and DSR

- **TL;DR:** الهوية، الأمان، واكتساب السائقين
- **Manus:** افتراضات
 1. "المصادقة هي مجرد "شاشة تسجيل دخول
 2. "هو مجرد "نموذج DSR
- **الواقع الحالي:**
 1. (منطق مصادقة منفذ جزئيًا (غالبًا بدون معالجة تجديد التوكن الكاملة
- **الفجوات والتناقضات:**
 - API إذا فشل استدعاء Uber في تطبيق بمستوى (Token Refresh Rotation) الفجوة: تدوير تجديد التوكن هذا ينتمي (Interceptor) "برمز 401، يجب أن يقوم التطبيق بتجديد التوكن بصمت وإعادة المحاولة. منطق "المعتزض ولكنه يؤثر على المصادقة HTTP Shim لطبقة

- الرحلات) يحتاج لمستخدم موثق لإنشاء حجز) B لكن المسار ،(D) التناقض: تضع الخطة هذا المسار أخيرًا

docs/analysis/manus_execution_recommendations.md: الملف الثالث

توصيات التنفيذ وخارطة الطريق

1. مقترح ترتيب التنفيذ (Execution Order Proposal)

التسلسل الموصى به: Track A → Track D → Track B → Track C

- الأساس):** لا يمكننا بناء ميزات بدون القشرة ونظام التصميم. إنه يفرض القواعد المعمارية) Track A
- صالح (Auth Token) وتوكن مصادقة (User Entity) الهوية/المصادقة): تم تقديمه. نحتاج إلى كيان مستخدم) Track D
بمستخدمين وهميين يخلق ديونًا تقنية B لأي تطبيق عمودي. بناء المسار Backend للتفاعل مع الـ
- (Sockets) ومقابس الوقت الحقيقي Map Shim الرحلات): القيمة الأعلى. يتحقق من صحة MVP) Track B
- B. الطرود/الطعام): توسع في التعقيد. يمكن العمل عليه بالتوازي بمجرد ترسيخ الأنماط في المسار) Track C

2. المخاطر والأسئلة المفتوحة (Risks & Open Questions)

المخاطر الاستراتيجية:

- أغلفة للخرائط، المدفوعات، المصادقة) يستغرق وقتًا أطول بمرتين من التكامل) Shims تطوير الـ "Shim": عنق زجاجة الـ المباشر.
 - الميزات Sprints قبل Shims بنية تحتية" محددة فقط لتطوير الـ Sprints" التخفيف: تخصيص
- إذا كانت الجلسة طويلة جدًا "Banned Imports" قد يفقد وكيل الذكاء الاصطناعي سياق قاعدة Manus: حدود سياق
يخرق القاعدة، مما يجبر الوكيل على تصحيح نفسه PR لرفض أي (CI/CD) صارم Linter التخفيف: استخدام

أسئلة مفتوحة للحسم:

- Feature Flag استراتيجية الطرود والطعام: هل نريد إتاحتها للجمهور فورًا، أم يجب إخفاؤها خلف (enable_logistics_vertical)؟

2. جاهزة؟ إذا لم تكن كذلك، يجب بناء RideMatching Socket API هل واجهة Backend: جاهزية الـ B. لفك الحظر عن تطوير المسار Dart في MockRideService
3. الآن لدعم التبديل بينهما لاحقاً Shim ؟ يجب تصميم واجهة الـ Mapbox مقابل Google Maps مزود الخرائط: هل قررنا
4. ؟ إذا كانت (Admin Panel) مؤتمتة بالكامل أم تتطلب لوحة إدارة (Track D) هل عملية انضمام السائقين DSR: التحقق من يدوية، نحتاج لمعرفة من سيتحقق من السائقين أثناء الاختبار

3. التوافق مع المعايير الداخلية (Alignment with Internal Standards)

"Clean B Workspace" تفرض خطة التنفيذ هذه بصرامة معايير

- infrastructure. أبداً على app لا يعتمد UI و domain على app قاعدة التبعية: يعتمد
- اختبارات وحدة. خط الأساس "158 اختباراً" هو A-D يتم إنشاؤه للمسارات PR الحفاظ على الاختبارات: يجب أن يتضمن كل الحد الأدنى
- مما يتضمن الفصل، (app, shims, features/*) لإدارة الحزم المنفصلة melos سيتم استخدام Monorepo صحة الـ المادي للوحدات على نظام الملفات

Works cited

1. Manus AI: General AI Agent, accessed November 27, 2025, <https://manus.so/>
2. #1 Manus AI - The World's First General-Purpose AI Assistant Launching in 2025, accessed November 27, 2025, <https://www.manusai.io/>
3. How to Build a Taxi App like Uber in 2024: Features and Cost - Django Stars, accessed November 27, 2025, <https://djangostars.com/blog/how-much-does-it-cost-to-build-a-taxi-app-like-uber/>
4. Effects of simplicity and discipline on operational flexibility: An empirical reexamination of the rigid flexibility model | Request PDF - ResearchGate, accessed November 27, 2025, https://www.researchgate.net/publication/247174124_Effects_of_simplicity_and_discipline_on_operational_flexibility_An_empirical_reexamination_of_the_rigid_flexibility_model
5. The Concepts and Determinants of Manufacturing Flexibility | Request PDF - ResearchGate, accessed November 27, 2025, https://www.researchgate.net/publication/362942471_The_Concepts_and_Determinants_of_Manufacturing_Flexibility
6. Considerations on a contemporary Flexibility Approach | Request PDF - ResearchGate, accessed November 27, 2025, https://www.researchgate.net/publication/234092839_Considerations_on_a_contemporary_Flexibility_Approach
7. Manus: Hands On AI, accessed November 27, 2025, <https://manus.im/>

8. Context Engineering for AI Agents: Lessons from Building Manus, accessed November 27, 2025,
<https://manus.im/blog/Context-Engineering-for-AI-Agents-Lessons-from-Building-Manus>