

In ecommerce modernization, the buzz around **experience-led delivery** agencies like DEPT is growing louder. Their strength lies in crafting customer-facing digital experiences that resonate. Yet, for backend-heavy ecommerce platforms, especially those relying on robust order management, inventory controls, and enterprise integrations, this focus sparks a question: *does an experience-first approach introduce risks or blind spots?*

Through the lens of composable commerce initiatives executed by firms like **Netguru**, DEPT, and **Codal**, this post explores the balance between frontend brilliance and backend rigor. We'll dissect the challenges of composable commerce risk, cost control through modular scope discipline, and the critical role of clear system boundaries, API-driven integrations, and frontend-backend tradeoffs.

## Experience-Led Delivery: What's in the Name?

Ask yourself this: "experience-led delivery" means starting with the end user's journey — crafting interfaces, touchpoints, and flows that engage customers seamlessly. Agencies like DEPT excel here, translating brand values into interactive storefronts using technologies such as headless storefronts and API-first architectures. This approach focuses on:

- Agile iteration on the customer experience
- Innovative frontend design with composable components
- Rapid A/B testing and personalization

The challenge emerges when this focus overshadows backend complexities — vital for scalability, reliability, and operational control in enterprise ecommerce setups.



## The Backend-Heavy Reality of Enterprise Ecommerce

Unlike smaller ecommerce shops where the frontend can often drive the business, larger, backend-heavy enterprises depend on foundational systems such as:

- Order management systems (OMS)
- Inventory and warehouse management

- Payment processing and fraud prevention
- Customer data platforms and CRM integrations
- Supplier and drop-shipping coordination

These capabilities require [improve ecommerce core web vitals](#) careful attention to system boundaries, data integrity, and legacy technology compatibility. Rushing ahead with just an experience-led frontend can risk operational disruptions.

## Composable Commerce Risk: Balancing Frontend and Backend

Composable commerce, relying on modular, API-driven components, promises flexibility and speed. But this approach introduces specific risks, particularly when experience-led delivery overshadows backend governance:



| Risk                        | Description                                                           | Mitigation Strategy                                                           |
|-----------------------------|-----------------------------------------------------------------------|-------------------------------------------------------------------------------|
| Over-Engineering Frontend   | Spending excessive time perfecting UI/UX without backend scalability. | Modular scope discipline — prioritize MVP backend capabilities before polish. |
| Integration Debt            | Accumulating brittle API connections prone to failure under load.     | Conduct thorough API contract reviews and plan for replaceability.            |
| Lack of Long-Term Ownership | One-off delivery mindset misses operational handoff requirements.     | Define ownership clearly for year two and beyond in contracts.                |
| Opaque System Boundaries    | Unclear demarcation leads to overlapping responsibilities and bugs.   | Explicitly document system boundaries and data flows.                         |

## Lessons from the Field: Netguru, DEPT, and Codal

### Netguru: Modular Scope Discipline in Practice

Netguru emphasizes careful **scope control** — breaking down projects into composable parts with strict prioritization on backend essentials before frontend flourishes. This was crucial in a mid-market replatform where rushed frontend expansions caused costly backend fixes post-launch.

### DEPT: Experience-Led Strengths with Backend Caveats

DEPT brings undeniable mastery in delivering exceptional customer experiences. However, their projects sometimes reveal the hidden costs tied to deep backend complexity — a classic “scope creep” trap when the backend architecture isn’t tightly governed alongside the customer journey.

## Codal: API-Driven Integration and Replaceability Advocates

Codal's approach highlights an **API-first architecture** philosophy paired with insisting on clear system replaceability. Their teams ask, “Who owns this integration in year two?” — a vital question often overlooked in experience-centric projects. This mindset reduces risk in evolving backend systems.

## Ownership and Evolution: Beyond One-Off Delivery

Experience-led delivery can sometimes imply a “one-off” model: deliver the storefront, hand it off, and move on. For backend-heavy ecommerce, that’s risky. The ecommerce stack is dynamic. APIs evolve, suppliers change, business rules shift. Without clear long-term ownership:

- Technical debt accumulates quietly.
- Operational teams struggle with opaque integrations.
- Scaling new channels adds disproportionate complexity.

Vendors specialized in backend integrations, like those engaged by Codal, stress the need for contractually defining ownership beyond launch — a vital step to controlling hidden costs.

## Clear System Boundaries and Replaceability

API-driven integrations are double-edged swords. While they enable the flexible assembly of best-of-breed components, they require disciplined system boundary definitions and fail-safes for component swaps. The best projects:

1. Map APIs exhaustively and formalize contracts
2. Build in monitoring for API latency and errors
3. Keep replacement strategies documented and ready

Ignoring this invites fragile integrations, particularly in headless storefronts where frontend expectations hinge on backend responses.

## API-First Architecture Supports Controlled Evolution

Headless commerce platforms exemplify the API-first model, separating frontend presentation from backend logic. This architecture facilitates experimentation on the experience side without disrupting order management or payment processing. However:

- **APIs must be stable and backward compatible.**
- **New frontend features need backend readiness.**
- **Real-time data consistency is critical for operations.**

Picking an agency adept mostly at frontend risks creating a disconnected system where the backend and integrations strain under unsupported frontend demands.

# Frontend vs Backend Tradeoffs: Making Smart Choices

Experience-led agencies often favor pushing frontend innovation first. But in backend-heavy projects, tradeoffs must be balanced:

|                    |                                                  |                 |                                                       |
|--------------------|--------------------------------------------------|-----------------|-------------------------------------------------------|
| Aspect             | Frontend-Focused                                 | Backend-Focused | Speed to Market                                       |
| Initial rollout    | Fast                                             | Longer          | Visible UX benefits                                   |
| Foundational build | Longer                                           | Delayed         | UX enhancements                                       |
| Risk               | Lower                                            | Higher          | Risk of operational glitches post-launch              |
| Iteration          | Slower                                           | Faster          | Risk of slower iteration and missed user expectations |
| Cost Control       | Potential hidden backend rework costs            | Better          | Predictability through modular backend scope          |
| Ownership          | May neglect backend maintenance responsibilities | Clear           | Clear long-term ownership, including backend ops      |

Collaborative models, where frontend experts like DEPT partner closely with backend integrators like Netguru or Codal, often yield the best balance. Each team respects the other's domain and co-owns the end-to-end delivery.

## Final Thoughts: Is DEPT's Experience-Led Approach Risky for Backend-Heavy Ecommerce?

The short answer: It depends. If DEPT or similar agencies operate without tight backend governance and modular scope discipline, risk inevitably rises. But with conscious collaboration, clear API-first architecture, ownership clarity, and rigorous replaceability planning, experience-led delivery can powerfully differentiate ecommerce experiences without sacrificing backend stability.

For companies embarking on composable commerce transformations, here's what to watch for:

- **Demand modular backend scope prioritization** alongside frontend vision.
- **Insist on explicit system boundaries and API contracts.**
- **Clarify vendor ownership for long-term operations.**
- **Don't let frontend allure eclipse backend realities.**

Done right, DEPT's experience-led style plus backend champions like Netguru and Codal together can unlock true ecommerce agility and resilience.