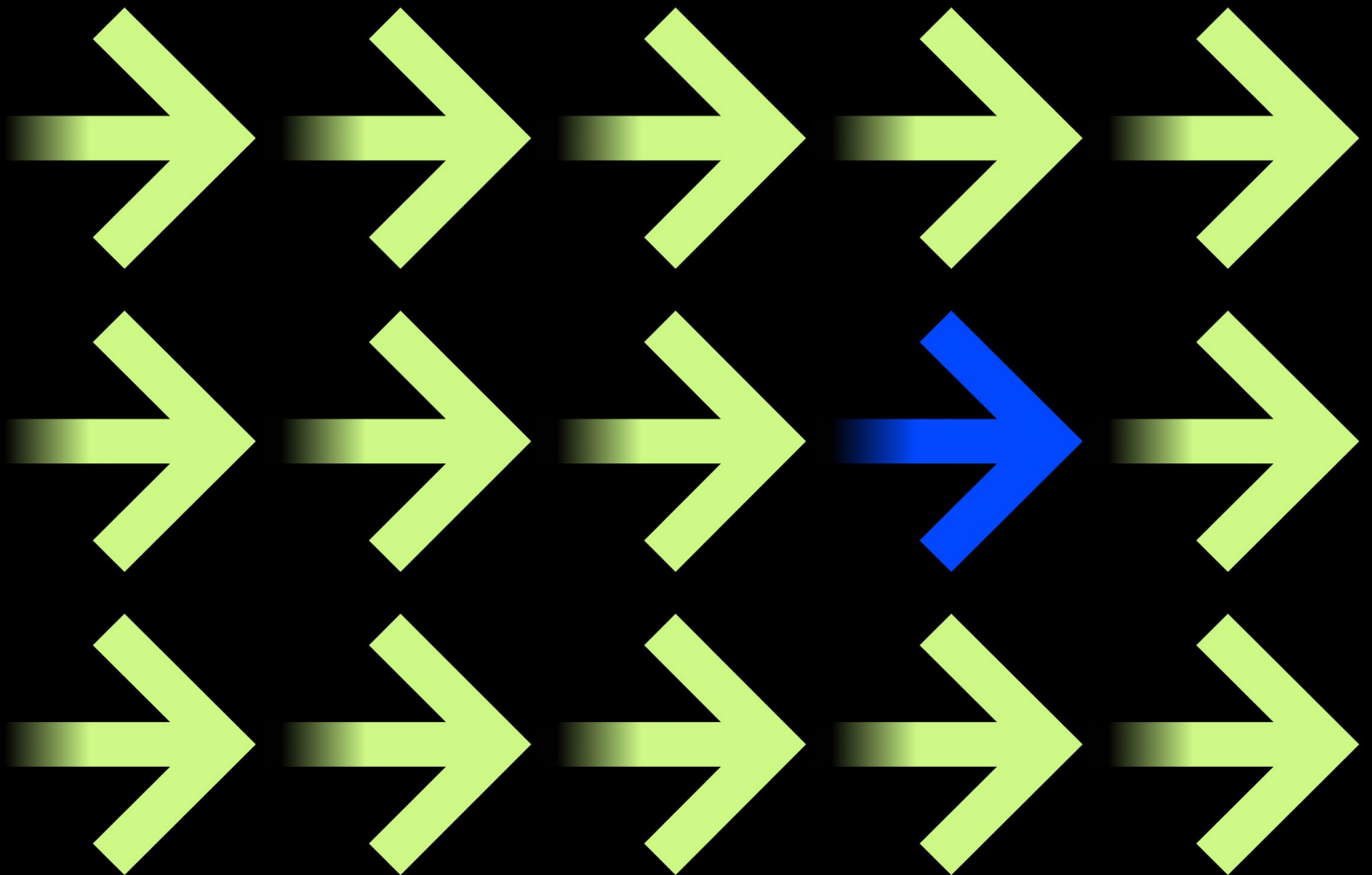


BigCommerce to Shopify

A proven migration framework for
technical decision makers moving
from BigCommerce to Shopify.



Contents

01	Executive summary	3	05	Shopify Migration from BigCommerce Checklist	38
02	The case for migration	4		A technical roadmap for platform transition	38
03	Key points of platform comparison	7		Migration strategy and planning	39
	Platform architecture: Modern foundations	7		Technical discovery	39
	BigCommerce: A distributed, microservices-first model	7		Data preparation	40
	Shopify: A modular monolith optimized for scale	8		Shopify store setup	40
	Extending Shopify: Tools that adapt with you	9		Theme and storefront rebuild	40
	Shopify Functions	9		Data migration	41
	Metaobjects + Online Store 2.0	9		SEO and redirects	41
	Shopify Flow	10		Integrations and extensibility	42
	Checkout Extensibility	10		Testing and QA	42
	Hydrogen + Oxygen	10		Go-live preparation	43
	Systems overview: Functional upgrades	12		Post-launch monitoring	43
	Terminology translation: The same language	14	06	Leap to Shopify with MTN Haus by your side	44
04	Navigating your migration	15			
	Consideration 1: Choosing the right migration path	16			
	CSV import or Matrixify	16			
	Custom API migration	17			
	Hybrid approach	17			
	Consideration 2: Ecosystem readiness	18			
	Tech ecosystem overview	18			
	Developer support	19			
	Consideration 3: Platform extensibility	20			
	Checkout customizations	22			
	Consideration 4: Timeline expectations	23			
	Consideration 5: Integration mapping	24			
	Consideration 6: Data migration	26			
	Schema comparison	26			
	Product data	26			
	Customer data	27			
	Order and transactional data	27			
	Collections vs. categories	28			
	Metafields and custom fields	28			
	Planning your data migration	29			
	Consideration 7: SEO redirects	30			
	Consideration 8: Customer account handling	32			
	DTC account migration	32			
	B2B account migration	32			
	Multi-store and internalization	33			
	Consideration 9: Post-migration QA	35			
	Consideration 10: Organizational impact	36			

01

Executive summary

This migration guide provides a comprehensive roadmap for transitioning from BigCommerce to Shopify. Beginning with a strategic analysis of the business case for migration, including total cost of ownership and operational efficiency gains, the guide establishes a clear rationale for platform transition.

This foundation continues with detailed systems comparisons—from PIM and OMS to payment infrastructure and analytics—alongside architectural differences between the platforms. We also include a precise terminology translation, ensuring all stakeholders share a common language throughout the migration process. This context then sets the stage for exploring ten critical migration considerations that shape successful platform transitions.

These considerations follow the natural progression of a migration project:

Selecting the optimal migration approach

Evaluating ecosystem capabilities

Understanding platform extensibility

Setting realistic timeline expectations

Mapping essential integrations

Structuring data migration effectively

Preserving SEO equity

Managing customer account transitions

Validating through quality assurance

Preparing for organizational change

Each section provides detailed technical guidance while maintaining clarity for non-technical stakeholders. Whether you're a CTO evaluating the transition or a team leader planning implementation, this guide offers strategic insight and practical direction for your migration to Shopify.

02

The case for migration

Every digital platform has its moment. And then, almost imperceptibly, the moment passes. What was once a competitive edge becomes a bottleneck. For many brands still operating on BigCommerce, that moment has arrived.

The urgency to move may not be acute for large enterprise retailers with extensive dev teams, long planning cycles, and deeply embedded platform customizations. However, for companies in a different phase—those growing with leaner teams, revenue accountability, and a mandate for operational efficiency—remaining on BigCommerce can introduce more friction than it resolves.

Across our ecosystem, we've seen a clear shift: brands are migrating to Shopify not just for modern tooling but because they can no longer afford the cost of staying put. The Shopify platform unlocks velocity—faster iteration, deployment, and customer experiences. And the economics support the switch. Brands that migrate see an average 12% lift in conversion, and Shopify's fully integrated infrastructure yields a 31% lower total cost of ownership (TCO) on average.*

“We had built so many weird little things into BigCommerce that it became a pseudo-ERP. At that point, we really couldn't do much with it... Shopify ticked all the boxes.”

George Phenix,
CTO of The Music Room

If you're a CTO evaluating this transition, understand that you're not just switching ecommerce platforms—you're choosing a new strategic foundation—one that reduces overhead, scales with your ambitions, and replaces technical drag with commercial lift.

Shopify vs. BigCommerce TCO comparison*

Research from a leading independent consulting firm found that **costs are 45% higher on BigCommerce** than on Shopify, on average. This, combined with the fact that **Shopify's checkout converts 12% better than BigCommerce**, paints a clear picture—it not only pays to be on Shopify, it costs you if you're not.

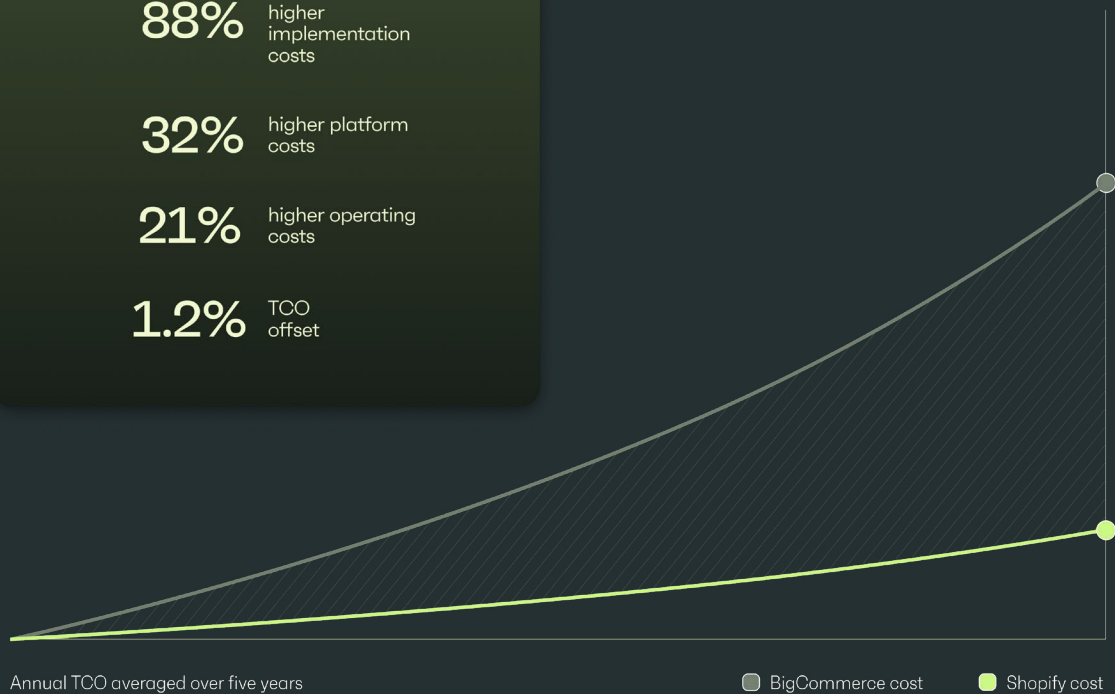
45% higher TCO compared to Shopify

88% higher implementation costs

32% higher platform costs

21% higher operating costs

1.2% TCO offset



This guide outlines the implications of migration, system by system, with enough technical depth for your engineers and enough clarity for your operators. If you're evaluating the next decade of your digital commerce strategy, this is where it begins.

*According to research commissioned by Shopify from a leading independent consulting firm to study TCO across major platforms in North America and understand Shopify's relative positioning based on objective research methods.



MTN Haus is the ideal partner for companies migrating from BigCommerce to Shopify due to our deep expertise, strategic approach, and proven track record. We specialize in seamless migrations across various platforms, ensuring businesses experience minimal disruption while enhancing performance, user experience, and scalability. Our team combines technical proficiency with a customer-centric mindset, focusing on preserving SEO, optimizing design, and integrating third-party apps to maximize Shopify's capabilities. Whether you're moving from BigCommerce, Magento, or custom-built platforms, we tailor our migration process to your unique needs, allowing for a smooth transition and future-proof growth. At MTN Haus, we empower brands with the tools, insights, and support they need to thrive on Shopify, helping them scale quickly, improve operations, and stay ahead in the competitive ecommerce landscape. We don't just migrate—We transform.

For more details about why brands are moving to Shopify, check out our in-depth comparison of BigCommerce and Shopify.

[Get details](#)

03

Key points of platform comparison

Platform architecture: Modern foundations

The architecture of your commerce platform defines more than just technical capabilities—it shapes how quickly you can move, how easily you can scale, and how confidently you can plan. BigCommerce and Shopify approach architecture from fundamentally different perspectives, with profound implications for brands in expansion mode.

BigCommerce: A distributed, microservices-first model

BigCommerce promotes an “open SaaS” model based on MACH principles—microservices, API-first, cloud-native, and headless. It exposes nearly all platform capabilities via GraphQL APIs and runs a loosely coupled system of services behind the scenes. This allows brands to customize freely across the stack, especially if they prefer complete control over every function and integration. BigCommerce’s architecture is flexible but also places more operational responsibility on the merchant’s team or partners:

01

Teams often manage more integration logic directly.

02

Checkout is customizable via scripts and SDKs, but requires developer intervention.

03

Infrastructure is abstracted, but less opinionated: you’ll configure caching, optimize theme speed, and build or maintain middleware for more profound logic.

The microservices model provides isolation between features but introduces versioning challenges and higher coordination overhead as complexity grows.

Shopify: A modular monolith optimized for scale

Shopify runs on a modular monolith architecture: a single, versioned codebase internally split into well-defined components (e.g., checkout, catalog, fulfillment). This design offers deep internal cohesion while allowing each domain team at Shopify to manage and evolve their slice of the platform independently. It enables coordinated scaling, streamlined testing, and predictable behavior across feature boundaries.

At the infrastructure level, Shopify operates on a modern, cloud-native foundation designed for elasticity, resilience, and global scale. The platform is engineered to absorb massive traffic spikes, such as Black Friday, without requiring merchant-side tuning or DevOps intervention. Shopify handles key aspects of compliance like PCI, data encryption, infrastructure security, fraud monitoring, audits, and more, so businesses don't have to. Core operational concerns like scaling, caching, and deployment automation are fully managed by Shopify as well. All of this allows teams to focus on product and growth, not infrastructure.

What this means in practice:

01

You don't maintain infrastructure, configure CDNs, or patch services.

02

Shopify's back end doesn't fragment across teams or integrations.

03

Updates to components don't break downstream behavior.

04

Checkout, storefront, and admin performance are stable under extreme load.

In short, it's designed to remove complexity that doesn't create value. Both platforms support extensibility but take different paths to get there. Shopify emphasizes maintainability, ecosystem alignment, and platform-wide stability. BigCommerce prioritizes customization and flexibility, but often at the cost of integration complexity. In practice, this results in two operating patterns:

Shopify: Developers build features, not foundations. Most extensibility—checkout logic, content modeling, workflow triggers—happens through native tooling. Hydrogen (React/Remix) and Functions offer full-stack flexibility within Shopify's managed environment for complex use cases.

BigCommerce: Developers spend more time integrating. The architecture supports modularity, but orchestration and reliability fall to the business or partner team. Flexibility comes with operational overhead.

Both are modern, cloud-native platforms. However, Shopify's architecture is built for long-term operational efficiency and team velocity. That difference, especially for lean teams or those focused on speed, reliability, and scale, compounds over time.

Extending Shopify: Tools that adapt with you

Migrating to Shopify isn't just a back-end upgrade—it's access to a suite of extensibility tools that reduce the need for infrastructure ownership. These tools are fully integrated, API-accessible, and built to address real commerce problems across checkout, content, logic, and hosting. Instead of coding around platform gaps, you build from solid ground, with native tooling that scales.

Below are the foundational capabilities that differentiate Shopify post-migration and support long-term agility.

Shopify Functions

Shopify Functions let you customize back-end logic like shipping, discounts, and payment methods without modifying Shopify's core. These are deployed as apps, configured in the admin, and executed at scale during checkout. There is no middleware, no template overrides.

For brands needing conditional logic, tiered pricing, or customer-based payment options, Functions offer native control, which is secured within Shopify's infrastructure and update cycles.

Metaobjects + Online Store 2.0

Storefronts aren't static anymore. Shopify's metaobjects allow you to define structured content types (e.g., fit guides, campaign modules) that plug directly into templates. Combined with Online Store 2.0's block-based architecture, this turns your theme into a flexible layout engine. Developers define schemas once, and operators gain UI-level control of dynamic content, creating a foundation for scalable content management.

Marketers can schedule content drops, seasonal refreshers, or promotions without developer intervention.

Building on this architectural foundation, Shopify brings significant improvements to day-to-day content operations. Marketers can schedule content drops, seasonal refreshers, or promotions without developer intervention. BigCommerce's content tooling is more limited,

often requiring outside platforms or custom staging environments to manage the same workflows. The result on Shopify is faster, more autonomous content management without sacrificing structure or consistency.

Shopify Flow

Flow is a no-code automation tool that links events, filters, and actions. Without scripting, you can trigger segmented emails, tag orders, pause risky checkouts, or sync CRM data.

For operations and CX teams, this replaces brittle Zapier workflows or internal cron jobs with centralized logic that lives inside the platform.

Checkout extensibility

Shopify's checkout used to be fixed. Now, Shopify Extensions in checkout and Shopify Functions enable secure customization—fields, logic, branding—without code forking.

This is critical for brands with B2B workflows, upsells, or regulatory data collection. Everything is versioned and upgrade-safe, so there is no risk of breaking core checkout.

Hydrogen + Oxygen

Hydrogen is Shopify's headless React framework, prewired with cart, routing, and product logic. Oxygen is its zero-config global edge hosting layer. Together, they power composable storefronts without infrastructure overhead.

“I had a tech team embedded in BigCommerce. Now, they’re doing meaningful development on Shopify. We reclaimed our talent.”

— George Phenix, CTO, Music Room

Hydrogen offers faster time to value for teams prioritizing custom UX, performance, or omnichannel use cases than building from scratch. Oxygen eliminates DevOps from the equation.

Shopify's architectural flexibility isn't theoretical—it's operational. Whether you're extending back-end logic, modeling dynamic content, automating workflows, or deploying headless storefronts, these tools minimize technical friction while maximizing optionality. Businesses move faster, operate leaner, and spend less time maintaining baseline functionality.

The result is a platform that scales from startup to enterprise without requiring periodic rebuilds. Shopify's architecture isn't just current. It's durable.

This architectural foundation is a key reason Shopify is able to evolve continuously. While BigCommerce moves conservatively, which benefits risk-averse brands but potentially slows time to market for those pursuing modern innovation, Shopify regularly releases platform upgrades that keep pace with modern commerce. From AI assistants like Shopify Magic to native bundles and visual merchandising tools, Shopify operates as a product company with a roadmap aligned to how brands need to operate in the future.

The takeaway? With Shopify, you're not betting on a rigid system or narrow vision. You're choosing a foundation that invites iteration, welcomes integration, and scales alongside your ambitions. And in a landscape where adaptability is survival, Shopify gives you the tools to stay ahead without forcing your developers to reinvent the basics every time the market shifts.

Your platform shouldn't be a limitation. Shopify ensures it never will be.

Systems overview: Functional upgrades

Shopify's architectural advantages translate directly into practical system improvements. Migration isn't just a front-end redesign—it's a comprehensive platform evolution that touches every core system. Here's how Shopify extends and enhances each fundamental component:

COMPONENT	BIGCOMMERCE	SHOPIFY
Product information management (PIM)	• Uses "option sets" to define product variants • Requires manual assignment • Limited logic reuse	• Native variant management • Metaobject support • Enterprise PIM integration via API or apps • Simplified import and merchandising
Order management system (OMS)	• Basic order flows • Less unified order dashboard • Limited automation	• Enhanced automation via Shopify Flow • Native centralized dashboard with filtering, tagging, notes, and order tracking
Shipping rules (delivery dates)	• Ability to show delivery dates only through third-party apps; no native display	• Native support for both automatic and manual delivery date display at checkout
Shipping rules (local delivery)	• No native local delivery option; requires third-party apps or custom setup	• Native local delivery feature, including area selection
Smart order routing and multi-location fulfillment	• Limited native multi-location support; often requires custom workflows or apps	• Native support for routing orders to optimal fulfillment locations across up to 1,000 locations
Bulk operations (labels, packing slips, fulfillment)	• Available via third-party apps or custom development	• Native bulk label printing, packing slips, and bulk order edits
Shipping rules (delivery dates)	• Ability to show delivery dates only through third-party apps; no native display	• Native support for both automatic and manual delivery date display at checkout
Shipping rules (local delivery)	• No native local delivery option; requires third-party apps or custom setup	• Native local delivery feature, including area selection

COMPONENT	BIGCOMMERCE	SHOPIFY
Smart order routing and multi-location fulfillment	Limited native multi-location support; often requires custom workflows or apps	Native support for routing orders to optimal fulfillment locations across up to 1,000 locations
Bulk operations (labels, packing slips, fulfillment)	Available via third-party apps or custom development	Native bulk label printing, packing slips, and bulk order edits
Customer relationship management (CRM)	- Basic customer information storage - Minimal segmentation	- Native segmentation tools - Personalization capabilities - Lifecycle stage tracking - Direct integration with HubSpot and Salesforce
Payment infrastructure	- Broad gateway support - Requires third-party management	- Simplified processing via Shopify Payments 50% faster checkout with Shop Pay - Improved cash flow velocity - Unified payment experience
Analytics and insights	- Basic reporting - Limited extensibility	- Financial dashboards - Conversion data - Retention cohorts - ShopifyQL for custom queries - Native product performance
Hosting and infrastructure	- SaaS platform - Manual optimization needed	- SaaS platform - Global edge network (Oxygen) - Automatic scaling and caching - React-based storefronts (Hydrogen) - Zero infrastructure management

Terminology translation: The same language

When moving from BigCommerce to Shopify, aligning technical and non-technical teams on the language used within each platform is essential. While the underlying concepts are often similar, the terminology can vary, leading to confusion during implementation. Here's a side-by-side breakdown to level-set the migration effort. A few important distinctions:

CONCEPT	BIGCOMMERCE TERM	SHOPIFY TERM
Product variants	Option Sets + Modifiers	Variants
Custom attributes	Custom Fields(text only)	Metafields / Metaobjects (structured data)
Product categorization	Categories (hierarchical, static)	Collections (manual or rule-based)
Scheduled content	Native (limited)	Launch Pad (promotions/themes) / Native (new content)
Multi-storefront	Native multi-storefront support (single admin)	Shopify Markets / Markets Pro (multi-region, same storefront)
Page content management	Page Builder (manual)	Metaobjects / Themes
Payment rules	Custom code	Shopify Functions
Checkout customization	Template overrides	Checkout UI Extensions

Understanding and mapping this terminology in advance reduces ambiguity in your migration plan and helps cross-functional teams work more efficiently during implementation.

04

Navigating your migration

Migrating from BigCommerce to Shopify is not a lift and shift—it's an intentional reset of your digital foundation. It allows your team to reduce technical debt, streamline operations, and modernize your customer experience. But it requires forethought.

To help you navigate this transition successfully, we've organized this guide around 10 critical migration considerations. Each section provides detailed guidance on:

- | | | | | |
|--|--|---|--|---|
| 01
Choosing the right migration path:
Evaluating and selecting your optimal approach | 02
Ecosystem readiness:
Identifying your platform capabilities | 03
Platform extensibility:
Maximizing customization potential | 04
Timeline expectations:
Understanding project phases and resource requirements | 05
Integration mapping:
Ensuring continuity of your technical ecosystem |
| 06
Data migration:
Structuring your data for optimal performance | 07
SEO redirects:
Preserving your discoverability | 08
Customer account handling: Managing user transitions seamlessly | 09
Post-migration QA: Validating your implementation | 10
Organizational impact:
Preparing your team for change |

These considerations reflect patterns MTN Haus has observed across successful migrations. By addressing each systematically, you can minimize risk, maintain momentum, and maximize the value of your transition to Shopify.

CONSIDERATION 01

Choosing the right migration path

Your choice of migration path sets the foundation for your entire Shopify journey. Understanding the available approaches—and which best suits your business—is crucial for success. Working with a Shopify agency or systems integrator (SI) provides valuable guidance and risk mitigation for teams without in-house Shopify expertise. These partners bring experience handling edge cases, managing SEO transitions, implementing redirects, and developing custom logic. Their managed approach helps navigate common pitfalls while ensuring best practices. In this section, we'll examine the two primary migration paths and the factors that may influence your decision.

CSV import or Matrixify

This is the fastest path for static, well-structured data. Shopify supports native CSV imports for basic resources (e.g., products, customers), but this is limited in scope and formatting. Most teams use Matrixify to manage bulk data import/export across various objects.

How it works: Matrixify allows the import of products, variants, meta fields, redirects, orders, collections, and content pages using structured CSV or Excel files. It supports predefined templates, field mapping, and validation. Import jobs can be scheduled or run on demand, and no custom code is required. This approach is ideal when:

In terms of limitations, this is a batch-based process. Inventory, pricing, or order data changed in BigCommerce post-export will not sync unless manually reimported. It also doesn't natively support conditional logic, relationship mapping, or delta detection.

01

Catalog and customer data can be staged and uploaded in batches

02

Real-time updates aren't required during cutover.

03

Data mapping is straightforward or can be preprocessed offline.

Custom API migration

Required for more complex implementations. API-driven migrations use scripts or middleware to extract data from BigCommerce and push it into Shopify using the Admin API. This approach supports:

01

Real-time or scheduled sync during parallel operations

02

Transformation logic (e.g., customer segmentation, dynamic metafield assignment)

03

System integrations (ERP, OMS, CRM) where Shopify is not the system of record

04

Developers can fetch and transform data programmatically, manage retry logic, and control pacing to meet rate limits. This allows for higher data integrity and operational continuity during go-live.

Considerations: API-based migrations require engineering effort, testing, and error handling. Setup time is longer but offers complete control and extensibility—critical for B2B scenarios, high-SKU catalogs, or live operational systems.

Hybrid approach

Most projects combine both approaches. Bulk data is loaded via Matrixify, followed by targeted API syncs for live orders, inventory, or customer records during final cutover. This balances speed and precision without overengineering the early stages.

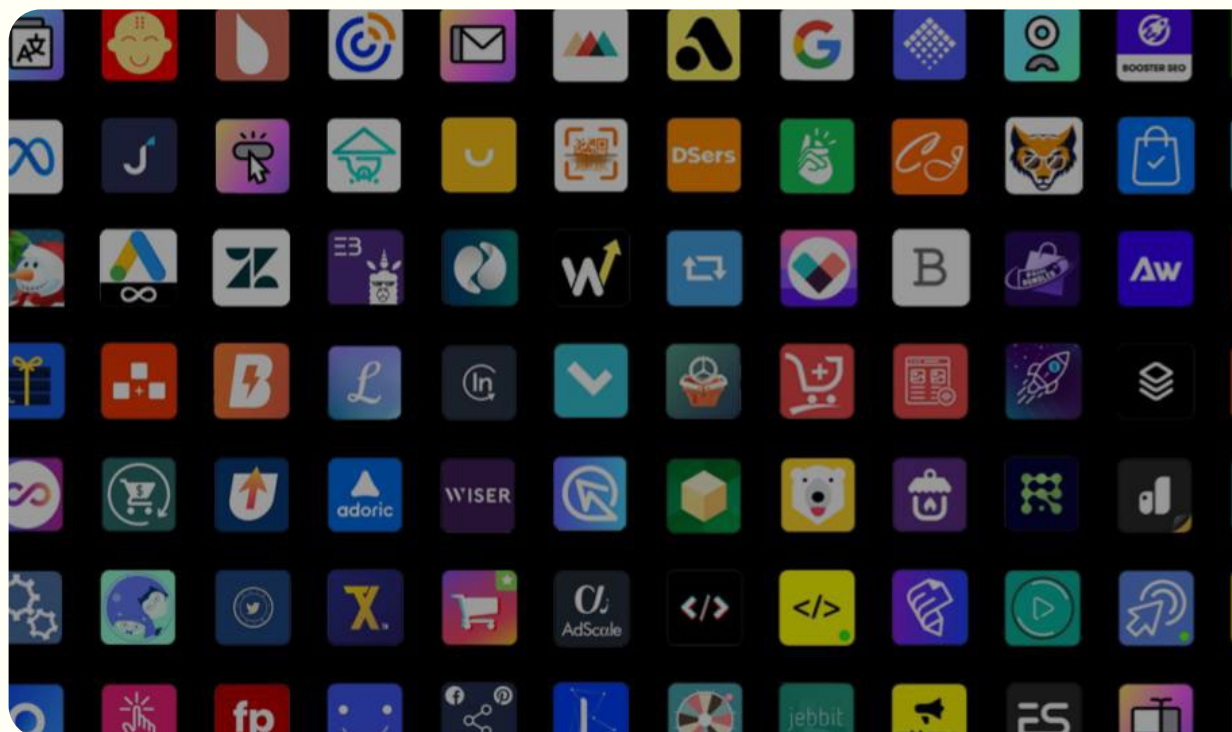
CONSIDERATION 02

Ecosystem readiness

When choosing a commerce platform, you're not just solving for today's requirements—you're investing in tomorrow's opportunities. This is where Shopify's ecosystem and extensibility genuinely shine.

Tech ecosystem overview

Shopify's technical ecosystem is built specifically for extensibility at scale. With over 8,000 apps in its marketplace—backed by a mature API surface, versioned developer tools, and a standardized UI framework—the platform doesn't just support third-party integrations, it accelerates them. Whether subscriptions, loyalty, fraud prevention, fulfillment orchestration, or AI-powered search, apps are deployed faster, perform more predictably, and integrate cleanly with Shopify's core data model. BigCommerce provides a smaller app ecosystem but compensates with more prebuilt, platform-native features. That can reduce initial configuration effort but limits optionality over time. Features are built-in, but customization and innovation lag what's available through Shopify's partner ecosystem, especially for enterprises that rely on specific tools to drive omnichannel growth, complex workflows, or regional market adaptation.



A common concern in replatforming discussions is, “Will I need more apps on Shopify?” In practice, the answer is yes—but that’s intentional. Shopify’s architecture assumes modularity, giving you access to best-in-class vendors rather than locking you into slower-moving native features. Loyalty is a prime example. Most enterprise brands implement solutions like Yotpo, LoyaltyLion, or Smile.io regardless of platform. The difference is that on Shopify, these tools are preintegrated, version-aligned, and supported natively across APIs and admin surfaces.

A common concern in replatforming discussions is, “Will I need more apps on Shopify?”

Integration depth is another differentiator. Shopify APIs are versioned, deeply adopted across the ecosystem, and increasingly GraphQL-first. Tools like Klaviyo, ReCharge, and Celigo are architected around Shopify’s extensibility model, often with prebuilt connectors and admin UIs that eliminate custom middleware or manual sync logic. For example, Celigo’s NetSuite integration on Shopify includes automated error handling, schema mapping, and real-time sync jobs. On BigCommerce, similar integrations typically require more implementation effort and ongoing maintenance.

At scale, Shopify’s ecosystem behaves more like a developer platform than a plugin marketplace. It’s composable by default—but with managed infrastructure, prevetted apps, and centralized permissions. That model pays dividends when evolving architecture over time, onboarding new tools, or rolling out changes across markets.

Developer support

The developer support structure highlights another key difference. Where BigCommerce developers often face API limitations and a smaller partner ecosystem, Shopify maintains a global network of certified Experts, Plus Partners, and developer agencies specializing in integrations, custom development, and UX optimization. This matters because complex builds—ERP syncs, subscription logic, B2B workflows—can’t always be solved with an app alone.

CONSIDERATION 03

Platform extensibility

Platform flexibility emerges as a clear differentiator. While both platforms offer modern commerce capabilities, their approaches to flexibility and customization differ significantly. Although BigCommerce offers built-in features, those features often become constraints. Because they are not easily extensible, what's built-in today may become outdated tomorrow. Shopify, on the other hand, takes a modular approach. Its platform is intentionally lean at the core, empowering businesses to build what they need through a curated app ecosystem and extensible APIs.

This difference in philosophy shows up everywhere:

Flexibility:

Shopify powers front-end and back-end extensibility through Functions, Hydrogen, and metaobjects. BigCommerce requires custom development or workarounds for changes beyond what's offered out of the box.

Customization:

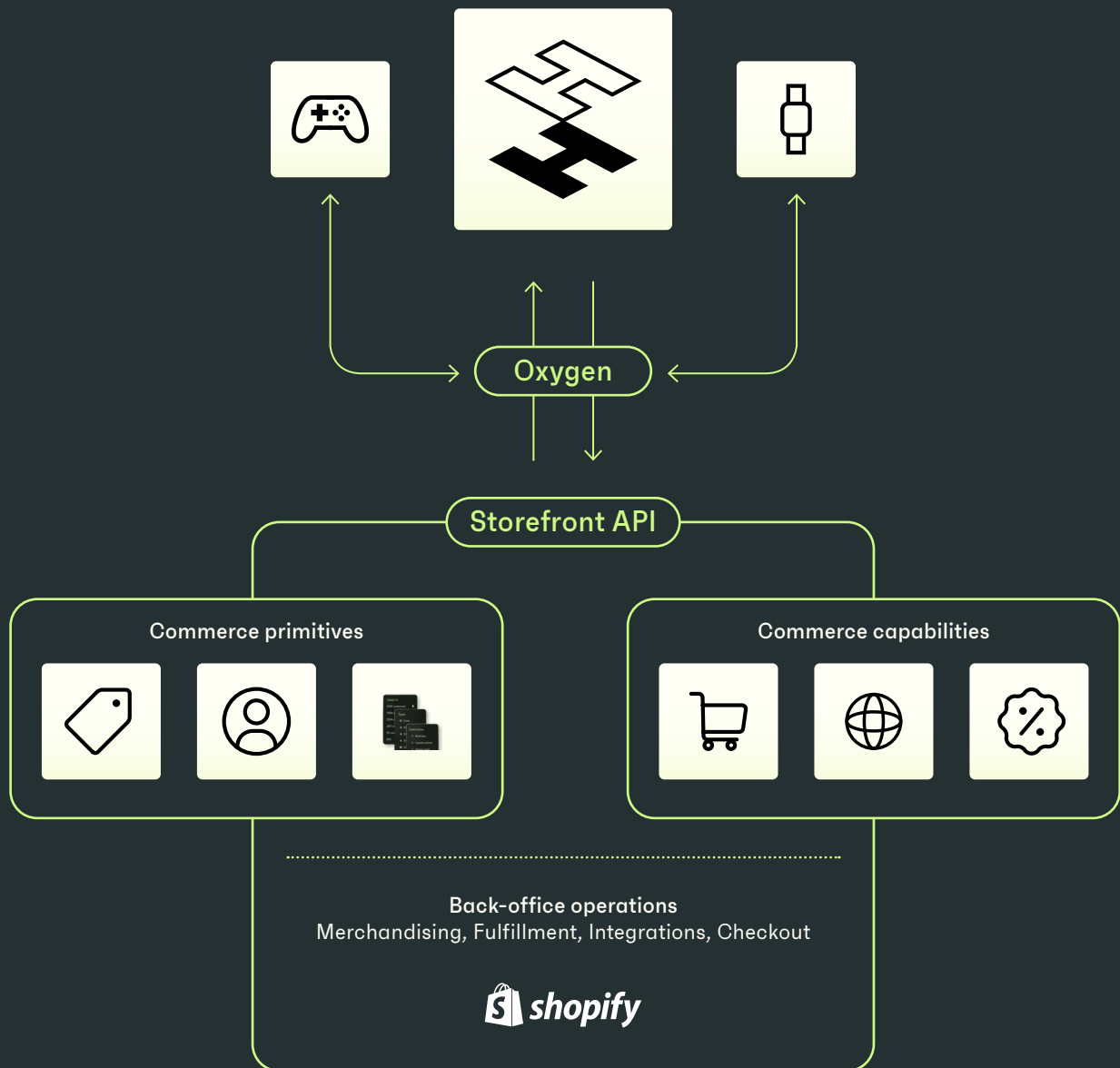
Shopify's APIs are versioned, well-documented, and widely supported, giving your team confidence to build for the long term. BigCommerce's APIs are functional but often inconsistent in handling data, especially across different modules.

Implementation vs. maintenance:

Shopify's Polaris design system and Online Store 2.0 architecture create more predictable development experiences. This minimizes tech debt and future-proofing costs. BigCommerce's architecture—while more open—can lead to one-off solutions that break over time.

Customer touchpoints

Shopify offers front-end flexibility through the Storefront API, allowing brands to use the framework of their choice—like Hydrogen with hosting included on Oxygen.



Checkout customizations

Checkout is where performance, compliance, and extensibility meet operational risk. How your platform handles it will impact your velocity, stability, and margin.

BigCommerce provides a few checkout options across plans, including one-page, open, redirected, embedded, and custom headless checkouts. Developers can override templates, inject logic, and modify flows at will. This suits edge-case requirements—multi-step tax forms, advanced shipping logic, or branded flows—but pushes complexity onto your team. Not only does every customization become a code dependency, but the limited ecosystem of tools and paid apps can start to take a toll. Plus, PCI compliance partially shifts to your court—any updates to the platform can risk regressions on your end. Integrations require constant validation across SDKs and vendor APIs.

Shopify minimizes operational risk in checkout management. Its checkout is centrally versioned, hosted, and PCI DSS-compliant by default. Businesses don't manage security patches, test core updates, or audit third-party customizations. Shopify introduces Shopify Extensions in checkout—native tools that let teams conditionally surface payment methods, enforce business logic, or customize layout without modifying core code. This keeps checkout extensible but hardened.

Shopify isolates that risk. Its checkout is centrally versioned, hosted, and PCI DSS-compliant by default. Businesses don't manage security patches, test core updates, or audit third-party customizations. Shopify introduces Shopify Extensions in checkout—native tools that let teams conditionally surface payment methods, enforce business logic, or customize layout without modifying core code. This keeps checkout extensible but hardened.

The architectural distinction is clear: BigCommerce offers control with exposure. Shopify offers adaptability with guardrails. At scale, it's the difference between systems that degrade quietly and those that fail predictably.

For teams managing high AOV, compliance mandates, or rapid iteration cycles, this layer isn't just a UI problem—it's a systems reliability question. Shopify's model contains that surface area, so your team doesn't have to.

CONSIDERATION 04

Timeline expectations

Planning for a realistic migration timeline is another vital step for project success. Your timeline will depend on catalog complexity, integration requirements, and internal resource availability. Let's break down what to expect at each stage.

Migration projects typically fall into three categories:

Small/clean catalogs: 2–4 weeks

Straightforward product structure
Minimal custom integrations
Standard theme implementation
Single market focus

Mid-size with some complexity: 6–8 weeks

Moderate catalog complexity
Multiple third-party integrations
Custom theme requirements
Basic multi-market needs

Enterprise or B2B: 12+ weeks

Complex product relationships
Custom integration development
Custom theme requirements or
(optional) headless implementation
Multi-market orchestration

The most successful migrations go beyond just data transfer. Your timeline should also account for:

- Discovery and planning
- Integration mapping and development
- Theme customization and testing
- Content migration and QA
- Staff training and system validation

This becomes especially critical when implementing headless front ends or managing multiple markets, where additional complexity requires careful staging and testing.

As part of your timeline, MTN Haus will recommend extending beyond the initial launch to include a stabilization period. After going live, monitor:

- Integration performance (timeout, error rates)
- Site speed optimization (especially on apps that inject JavaScript)
- Checkout flow consistency
- System integration health

This monitoring phase typically requires 2–4 weeks of dedicated attention post-launch to ensure all systems perform optimally.

CONSIDERATION 05

Integration mapping

A successful migration preserves your existing functionality and upgrades how your integrations operate within a more extensible, better-documented framework. For CTOs, the priority is to ensure that every operational dependency is accounted for and reimaged within Shopify's ecosystem.

Inventory your stack

Start by auditing every third-party service, integration, and custom script in your BigCommerce environment. Include:

- ERP and OMS platforms
- Shipping and logistics tools
- Personalization, loyalty, and review apps
- Payment processors and gateways
- CRM and marketing automation platforms
- Custom scripts and integrations

Evaluate Shopify capability

Shopify's ecosystem includes native integrations for most modern SaaS tools, including:

- NetSuite, via Celigo and Patchworks
- Salesforce, via API connectors or middleware
- Klaviyo, Yotpo, Recharge, Loop, Attentive, Gorgias, Avalara

The most popular BigCommerce-compatible tools are also available for Shopify. In fact, due to Shopify's popularity, they are often built with Shopify in mind first and can offer more features, better stability, and faster implementation times.

For unsupported tools, Shopify's mature ecosystem typically offers comparable native alternatives. If a standard integration doesn't meet your needs, Shopify's robust API framework (Admin API, Storefront API, Functions API) allows your dev team or partner agency to create custom middleware solutions. Shopify's versioned APIs and webhook support make it easier to maintain these solutions over time.

Partner ecosystem*

Our network of partners is second to none.

Shopify
2,380

BigCommerce
1,237

App ecosystem*

We help our customers do more, go bigger, and sell wider.

Shopify
8,000

BigCommerce
1,208

Choosing the right Shopify apps

Shopify maintains rigorous standards for all apps listed in its App Store, including mandatory security, performance, and functionality reviews.

Apps awarded the “Built for Shopify” badge have undergone additional vetting, meeting Shopify’s highest standards for reliability, performance, security, and user experience. While apps without this badge still meet Shopify’s baseline quality and security requirements, the badge provides an extra indicator of excellence and close alignment with Shopify’s platform.

When evaluating apps, consider the following:

- Regular updates and active development practices
- Positive user reviews and comprehensive support documentation
- Compatibility with Shopify’s Online Store 2.0 and metafield structure

For more information on Shopify’s app vetting and certification processes, check out [Partner Program Technology Track](#).

CONSIDERATION 06

Data migration

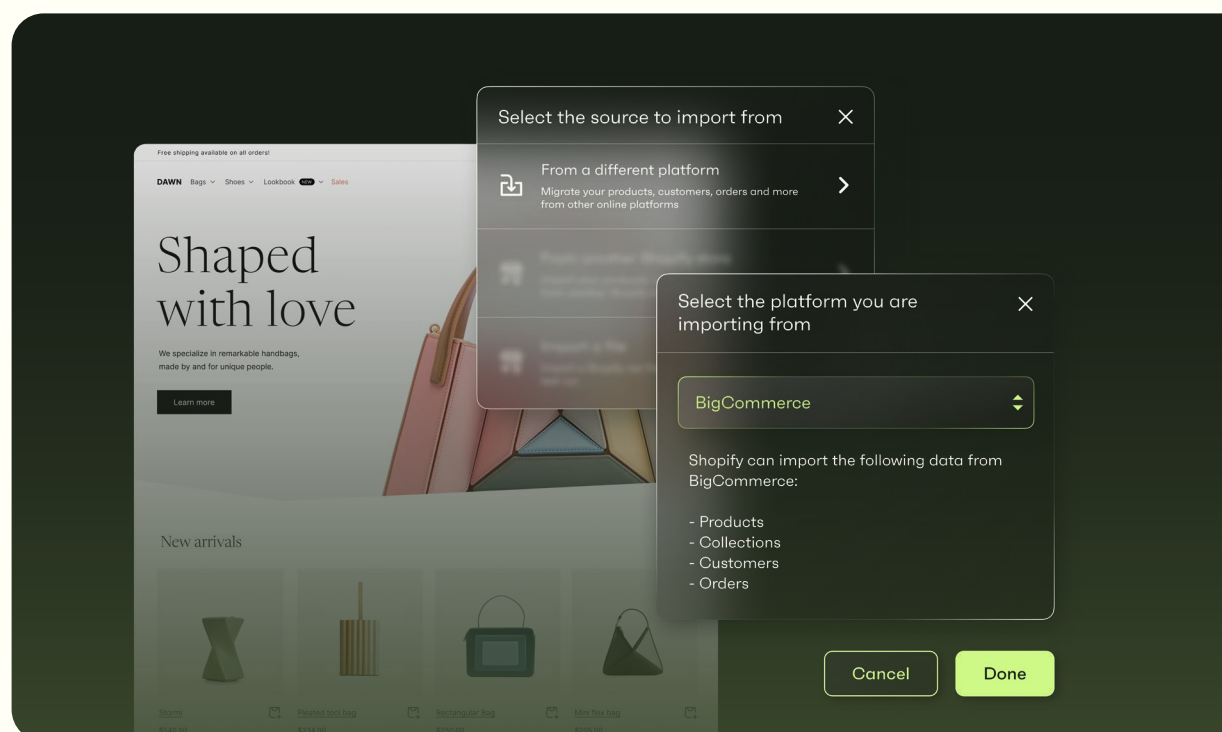
One of the most overlooked challenges in platform migration is how each system handles and structures data. Shopify and BigCommerce support core entities—products, customers, orders—but their models diverge in ways that materially affect implementation, performance, and future extensibility.

Schema comparison

Migrating between platforms isn't just about functionality—it's also data fidelity. How each system structures product, customer, and operational data plays a defining role in how well your business can scale and adapt post-migration. Here's a detailed breakdown of how BigCommerce and Shopify differ in data schemas and what that means for your migration strategy.

→ Product data

BigCommerce organizes product variations through option sets and modifiers, requiring brands to manually define and assign options (like size or color) separately from the product. This approach creates unnecessary complexity, reduces reusability, and results in a disjointed



data model with extra configuration steps. Option sets are not portable across storefronts or feeds, and modifier fields are not included in most standard data exports, requiring workarounds during migration.

Shopify, by contrast, treats each variant as a structured part of the product object. Up to 100 variants are supported within the Admin UI (with up to 2,000 via API), each including its price, SKU, inventory, and metafields—all editable in a single admin view. This creates a clean, intuitive, and developer-friendly data structure. Variant relationships are declared in line with the product definition, simplifying import/export operations and reducing friction when building custom storefronts or syncing data to third-party platforms.

Plus, Shopify's metaobjects introduce a new paradigm for managing structured content. These reusable schema entities (e.g., fabric guides, sustainability scores) enable schema-based content management that scales across localization and merchandising use cases. For developers, this means fewer workarounds; for content managers, it means faster workflows across the entire catalog. Metaobjects can be defined centrally and linked across products, enabling consistency and reusability. They are instrumental in headless builds, dynamic filtering, or region-specific merchandising.

→ [Customer data](#)

While both platforms capture standard customer fields (name, email, order history), Shopify's model goes significantly further. Shopify natively supports tagging, segmentation, and metafields at the customer level, enabling brands to enrich records without relying on external systems. These fields are queryable via the API and manageable in the admin, allowing flexible workflows across personalization, support, and marketing. For B2B use cases, Shopify introduces structured objects like Companies and Locations, supporting multi-user accounts, role-based permissions, payment terms, and shared order history—all built into Shopify Plus.

By comparison, BigCommerce supports customer groups and custom fields, but automated customer segmentation is inaccessible in the UI and lacks schema enforcement. More advanced segmentation, tagging, and account-level structures require custom development or third-party apps. BigCommerce's B2B Edition adds company-level features, but this functionality exists outside the core platform. This difference affects how identity, segmentation logic, and permission models are mapped when migrating, especially for hybrid or wholesale businesses with complex account hierarchies.

→ Order and transactional data

Shopify provides a more comprehensive native order management system than BigCommerce. While both platforms support standard orders, Shopify extends this functionality with Draft Orders and Abandoned Checkouts tracking built directly into the platform. Shopify also includes native tools for managing complex workflows, such as preorders or custom quotes, without requiring additional solutions.

BigCommerce businesses typically need to rely on third-party applications or external ERP systems to achieve this level of control, which introduces additional integration risks and operational overhead to your technology stack. This distinction becomes particularly important when planning your migration timeline and resource allocation.

→ Collections vs. categories

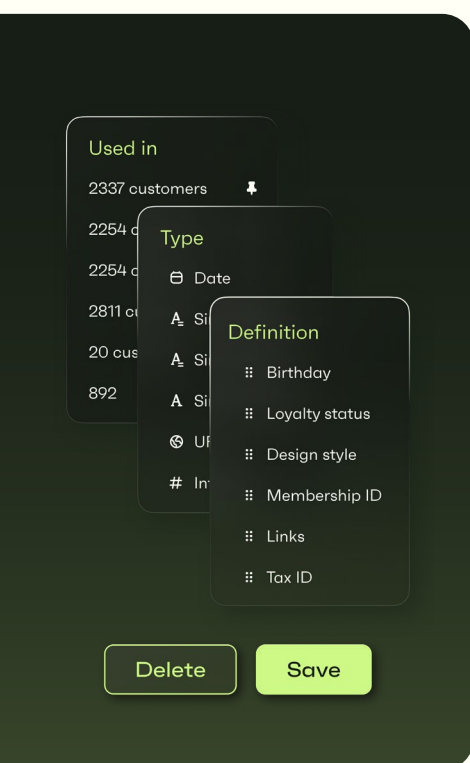
The organizational structure for products differs significantly between the two platforms. BigCommerce relies on categories, which function as rigid hierarchical structures that can limit flexibility. When your merchandising needs change or you need to reorganize your product catalog, categories often require developer intervention and technical resources to implement changes.

Shopify, in contrast, utilizes collections, which offer substantially more flexibility and control. Collections can be manually curated or automatically populated through rule-based logic (for example, all products with a tag equal to “summer”). This fundamental difference allows faster merchandising decisions and more dynamic product organization that can adapt to seasonal changes or promotional needs without technical bottlenecks. For technical teams managing the migration, this represents an opportunity to reduce ongoing maintenance requirements while providing marketing teams greater autonomy.

→ Metafields and custom fields

Data extensibility represents a critical difference between the two platforms. BigCommerce provides basic custom fields functionality, which is limited to plain-text values without structured data types or native UI support. This limitation often forces development teams to build custom validation logic and administrative interfaces. Shopify’s metafields system offers a more sophisticated approach with support for JSON schema definitions and built-in validation rules.

This structured approach ensures your extended data remains clean, consistently formatted, and searchable across your commerce



ecosystem. Metafields are fully integrated with Shopify's administrative tools and can seamlessly incorporate into storefront logic without additional development work. For technical teams, this means reduced custom code maintenance and fewer data integrity issues during and after migration. The structured nature of metafields also enables more powerful filtering, searching, and merchandising capabilities that would require custom development in BigCommerce.

Planning your data migration

A successful data migration requires more than just data transfer—it demands strategic mapping and cleanup. BigCommerce's option sets, modifiers, and custom fields must be thoughtfully translated into Shopify's product variables, metafields, and collections. Tools like Matrixify and Shopify's Admin APIs allow for smooth transitions, but the real catalyst happens when brands embrace Shopify's data structure instead of replicating BigCommerce's.

The most significant benefit of moving to Shopify may emerge after migration: data becomes more flexible and accessible. Schema changes that once required developer tickets can often be implemented directly by merchandisers or marketers using the admin UI or schema templates. This democratizes access to structured data and reduces overhead across teams.

In other words: don't bring your baggage. Bring your data—and let it move faster, cleaner, and more flexibly in Shopify's world.

Your data isn't just migrated. It's upgraded.

CONSIDERATION 07

SEO redirects

AUTOMATIC CANONICAL TAGS

XML SITEMAP GENERATION

STRUCTURED DATA (JSON-LD)

CLEAN SEMANTIC MARKUP

SEO preservation is one of the most overlooked aspects of platform migration, yet it's crucial for maintaining your digital presence and organic traffic. A thoughtful SEO migration strategy protects your search rankings and ensures customer pathways remain intact.

Shopify and BigCommerce differ in how they structure URLs, metadata, and templating logic—differences that materially impact how search engines crawl and interpret content. Shopify uses a flatter, standardized URL structure (/products/, /collections/, /pages/), compared to BigCommerce's deeper or query-based paths. This requires intentional 301 redirect mapping to preserve link equity and prevent crawl errors. All legacy URLs—product, category, and content—should be resolved before launch.

Shopify's default stack covers key SEO elements: automatic canonical tags, XML sitemap generation, structured data (JSON-LD), and clean semantic markup. Use these defaults as a baseline, then extend as needed through theme customization or metaobject-driven pages. Metaobjects are especially useful for structured landing pages, brand content, and campaign assets, surfacing schema-rich, crawlable URLs that improve indexing and relevance.

Speed is part of SEO. Fast, stable pages improve crawl efficiency, boost indexing, and are prioritized in search rankings, especially on mobile.

Speed is part of SEO. Fast, stable pages improve crawl efficiency, boost indexing, and are prioritized in search rankings, especially on mobile. Shopify's infrastructure and default optimizations—lazy loading, image compression, structured data, and metaobject-based content rendering—help ensure SEO best practices are built into the performance layer.

Shopify also supports modern SEO workflows through Shopify Magic. This AI-powered toolset can generate product descriptions, meta tags, and alt text at scale. This is especially useful for extensive catalogs or multi-language setups. Native JSON-LD output enhances eligibility for rich search results such as product snippets, FAQ, and review stars.

Expect short-term volatility in rankings post-migration, particularly if redirects, structured data, or metadata are misaligned. With proper planning, Shopify SEO experts estimate that most sites can regain baseline organic traffic within 30 to 90 days, though traffic recovery can take longer with more pages and more non-branded traffic. SEO outcomes are shaped by theme decisions and how teams approach redirects, data modeling, content ownership, and third-party app selection.

Remember, migration isn't the end of integration work, nor is it just about preserving SEO equity—it's the opportunity to do it right, at scale, to enhance your technical SEO foundation for future growth.

CONSIDERATION 08

Customer account handling

Managing customer accounts during migration requires careful attention to both consumer and business relationships. A well-planned approach ensures continuity of service while upgrading account management capabilities.

DTC account migration

Shopify supports passwordless login via secure one-time email links, allowing customers to reauthenticate without resetting credentials. This eliminates friction while maintaining account integrity and security. Customer records—including names, emails, address history, and tags—can be imported via API or CSV during migration.

Shopify’s native customer model reduces the need for app-layer segmentation or personalization logic, which is often required in BigCommerce.

Custom fields from BigCommerce can be translated into Shopify’s metafields, catalyzing richer segmentation, loyalty logic, or internal workflows post-launch. Store credit and gift card balances are also portable, assuming gateway support for data export. Shopify’s native customer model reduces the need for app-layer segmentation or personalization logic, which is often required in BigCommerce.

B2B account migration

BigCommerce’s B2B Edition offers functionality like shared accounts, price lists, and quote requests—but it operates via an app layered on top of the core platform, requiring separate management and integration upkeep.

Shopify consolidates this logic natively. B2B entities are represented as companies with associated users, locations, and net terms. Native features include:

- Multiuser company accounts with permission roles
- Shared carts and order history at the company level
- B2B-specific checkout, account portal, and draft orders
- Company-specific catalogs and pricing via price lists
- Payment terms enforcement (e.g., Net 30) at checkout

These capabilities are fully integrated into Shopify's admin, API, and storefront logic, eliminating the need for external systems to support core B2B functions. This reduces integration complexity and creates a more sustainable foundation for expanding wholesale operations.

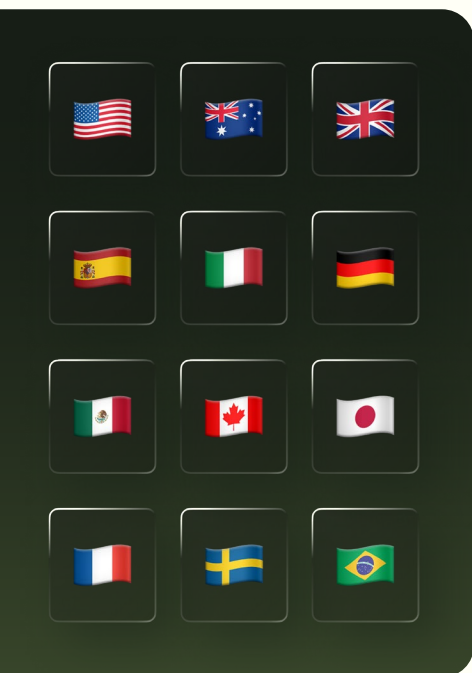
Migrating B2B data from BigCommerce requires mapping customer groups, pricing structures, and account hierarchies into Shopify's B2B model. Import tooling is available for Companies and Price Lists, but care is needed to preserve user roles, company relationships, and payment terms during the transition. As B2B use cases grow in complexity, having this logic managed natively simplifies both operations and long-term extensibility.

Multi-store and internalization

BigCommerce supports true multi-storefronts from a single admin, allowing businesses to manage multiple front ends—each with its own domain, theme, catalog, and pricing. This model is often used by businesses operating distinct regional stores or separate brand divisions that require full visual or structural separation.

Shopify takes a more unified approach. With Shopify Markets (and Managed Markets), a single storefront can support multiple markets through localized experiences, domain management, and content targeting. Key capabilities include:

- Automatic country and language detection with market-based routing
- Market-specific product visibility and content control
- Centralized analytics, order management, and content workflows
- Local pricing with currency conversion, rounding rules, and duties
- SEO-friendly domain structure and hreflang tag generation



If your current architecture includes multiple BigCommerce storefronts, migrating to Shopify may involve consolidating them into a single or reduced number of store instances. This shift often results in lower platform overhead, improved customer data consistency, and streamlined content governance, but it also introduces new planning requirements. You'll need to:

- Redefine catalog architecture to support market-specific visibility rules
- Model localized content using Shopify's native i18n support or metaobjects
- Align SEO strategy across domains or subfolders, depending on market structure

Customer account migration isn't just about preserving access. It's about enhancing the relationship management capabilities for both DTC and B2B customers.

The underlying architectural models differ: BigCommerce is optimized for storefront-level separation, while Shopify emphasizes centralized control with built-in localization. The result is fewer administrative surfaces to manage, greater consistency across regions, and tighter integration between content, commerce, and analytics, which provides initial planning accounts for market-specific catalog and governance needs.

Customer account migration isn't just about preserving access. It's about enhancing the relationship management capabilities for both DTC and B2B customers. Migrating from BigCommerce presents an opportunity to replicate existing functionality and redesign for global scale with fewer moving parts.

CONSIDERATION 09

Post-migration QA

Quality assurance isn't just a final box to be ticked. It's a critical phase that ensures your migration delivers on its promise of improved performance and reliability. Shopify's architecture replicates this process, but success demands disciplined attention to detail. It starts with prelaunch validation. Before flipping the DNS switch, ensure that data fidelity, front-end responsiveness, app functionality, payment workflows, and analytics tracking are all functioning. Run actual transactions in a test environment, simulating real-world customer journeys from browse to purchase.

Every theme and app is evaluated for load impact, which is why major retailers rely on Shopify for product drops, flash sales, and global traffic surges.

Next, move on to performance testing. Shopify has invested heavily in storefront performance, from a global CDN and built-in lazy loading to image compression and app-level performance reviews. Every theme and app is evaluated for load impact, which is why major retailers rely on Shopify for product drops, flash sales, and global traffic surges.

Focus on the key drivers: template speed, JavaScript payloads, and mobile responsiveness. Use Shopify's Web Performance Dashboard to monitor real-world metrics like Interaction to Next Paint and Largest Contentful Paint. Prioritize testing on high-traffic templates—home, product, cart, and checkout—and defer or eliminate scripts that block render or delay interactivity. And once live, it's time for post-launch monitoring:

- Integration performance (timeout, error rates)
- Site speed (especially on apps that inject JavaScript)
- Checkout flow consistency

Keep in mind that while Shopify provides the infrastructure for reliable performance, maintaining it requires ongoing attention and proactive monitoring.

CONSIDERATION 10

Organizational impact

Migration to Shopify isn't just about replatforming—it's about realigning your business to scale without baggage. This impact extends across teams, workflows, and operational models.

Commerce operations on BigCommerce are often centralized. Developers manage theme updates, integrations, app logic, and staging pipelines. Content changes, product configurations, and UX adjustments frequently require technical intervention or custom development. In this model, throughput is limited by technical bandwidth, and experimentation is constrained by release cycles.

Business users can configure promotions, update content, launch pages, and run A/B tests directly through the admin.

Shopify introduces a distributed model. Business users can configure promotions, update content, launch pages, and run A/B tests directly through the admin. Online Store 2.0, app-based workflows, and dynamic metafields shift day-to-day control from engineers to operators. This reduces context switching, shortens feedback loops, and unlocks parallel workstreams between development, marketing, and merchandising teams.

The governance model also evolves. Instead of managing platform health through release gates and code reviews, teams operate within Shopify's guardrails: app permissions, theme schemas, and standardized API behaviors. Hosting, scaling, and deployment are fully managed, enabling faster execution with fewer infrastructure dependencies.



This shift has a measurable operational impact. For example, 4ocean reduced developer reliance and recovered tens of thousands in unnecessary overhead after returning to Shopify from BigCommerce. Initiatives that once required third-party vendors—site updates, campaign launches, and SKU management—were handled internally. “On Shopify, the marketing team can just do what they need to do. If they want to change a banner or run a new promotion, they can do that in a few clicks and get data back quickly, versus relying on multiple people and external vendors to drive the business,” said cofounder Alex Schulze.

For Skullcandy, the move accelerated delivery and reduced technical debt. Product pages went live in under 30 minutes. Homepage load time dropped from 2.6 seconds to 0.8 seconds. Experimentation became routine, not risky. “We now have a testing strategy that allows us to iterate, learn, and improve in a highly agile way, whether through Shopify’s native tools or by integrating third-party apps,” shared their VP of ecommerce. Teams shifted from maintenance to innovation—focused less on uptime, more on growth.

The long-term effect is a leaner, more responsive commerce organization. Less engineering effort is spent maintaining table-stakes functionality. Platform updates and ecosystem changes are absorbed without custom intervention. With the proper implementation, teams see efficiency gains across development hours, deployment velocity, and time to market, setting the stage for scalable operations built on configuration, not code.

05

Shopify migration checklist from BigCommerce

A technical roadmap for platform transition

Migration strategy and planning →

1

Technical discovery →

2

Data preparation →

3

Shopify store setup →

4

Theme and storefront rebuild →

5

Data migration →

6

SEO and redirects →

7

Integrations and extensibility →

8

Testing and QA →

9

Go-live preparation →

10

Post-launch monitoring →

11

1 Migration strategy and planning



Evaluate migration approach: manual, automated (Matrixify, Cart2Cart), or via partner agency.



Define business case: cost of ownership, infrastructure offload, velocity gains.



List all store features, apps, and customizations.



Align stakeholders: CTO, product owner, engineering, ops.



Define timeline, phases, and responsibilities.



Establish QA criteria and success metrics.



Document the current platform architecture and dependencies.

2 Technical discovery



Audit all third-party integrations (payment, shipping, ERP, CRM, PIM, analytics).



Identify custom APIs, webhooks, and serverless functions.



Document URL schema, redirects, SEO structure.



Review theme structure and page builder dependencies.



Inventory metafields/custom fields and product option logic.



Map BigCommerce terms to Shopify equivalents (e.g., custom fields → metafields).



Capture product taxonomy (categories → collections).

3 Data preparation



Clean and export all relevant data:

- Products
(+ variants, images, videos, files)
- Collections and categories
- Customers and addresses
- Orders (historical and active)
- Discount codes and gift cards
- Blog posts and pages
- Reviews
(from Stamped.io, Yotpo, etc.)



Prepare CSVs or use migration tools
(Matrixify recommended for flexibility).



Validate encoding, formatting, and
attribute mapping.

4 Shopify store setup



Choose a Shopify plan.



Configure settings (taxes, checkout,
shipping, notifications).



Set up staff accounts
and permissions.



Enable Shopify Payments or
integrate third-party payment
gateways.



Install essential apps (reviews,
search, personalization, ERP
connectors).

5 Theme and storefront rebuild



Purchase a prebuilt theme from the
Shopify Theme Store or customize
the Dawn theme.



Recreate navigation, page layouts,
and banners.



Implement dynamic content with
metaobjects and custom templates.



Ensure accessibility, performance,
and mobile responsiveness.



Integrate analytics tags
(GA4, Meta Pixel, etc.).

6 Data migration



Products and variants → with metafields and media



Customers → Turn off notifications before import



Orders → define order prefixes, avoid sending invoices



Pages and blog posts → match structure and apply redirects



Discounts, gift cards → validate expiration and usage limits



Reviews → migrate into a native or third-party app



Test imports, validate integrity before bulk sync

7 SEO and redirects



Prepare a redirect map (old → new URLs).



Set up Shopify redirects or use apps for large-scale mapping.



Reassign canonical URLs, meta titles/descriptions.



Resubmit the sitemap to Google Search Console.



Retain alt tags, structured data, and OpenGraph metadata.



Verify crawlability and page speed.

8 Integrations and extensibility



Reconnect or rebuild APIs for ERP, PIM, CRM.



Use Shopify Flow or Shopify Functions for automation.



Confirm compatibility of third-party tools (custom vs. app store).



Validate webhook behavior (e.g., order create/update).

9 Testing and QA



Validate data accuracy across all entities.



Test storefront UX across breakpoints.



Verify checkout flow, tax/shipping rates, and discount logic.



Conduct order placement and fulfillment testing.



Run performance tests and Lighthouse audits.



Review email notifications and transactional flows.

10 Go-live preparation



Perform a final data sync: customers, orders, redirects.



Disable password protection.



Update DNS records and point the domain to Shopify.



Verify site availability.



Send bulk account invite emails via third-party apps.



Launch communication (email, banner, social).

11 Post-launch monitoring



Monitor analytics for traffic drops, bounce rates, and 404s.



Run a daily error crawl with Screaming Frog or similar.



Collect customer feedback.



Review app performance and renewals.



Schedule regular backups and theme versioning.



Iterate on content, SEO, and conversion opportunities.

06

Leap to Shopify with MTN Haus by your side

Replatforming isn't a project—it's a pivot. For many CTOs and digital operators, the question isn't whether Shopify is technically superior. It's whether now is the right time to leap. For companies still pushing for efficient growth, the answer is increasingly evident.

The cost of waiting is more than a budget line. It's a tax on velocity, experimentation, and your team's ability to build and iterate without roadblocks. Shopify doesn't just offer a better platform and a better operating model—one where performance is native, integrations are streamlined, and business teams can move as fast as their strategy requires.

The next phase of commerce demands composability, performance, and strong ecosystems. Shopify is the foundation for that future. To the teams considering migration: You're not alone, and you're not early. You're right on time. Work with an experienced partner. Build the migration roadmap with both technical precision and business outcomes in mind. And when you make the switch, don't just replicate what you had. Rethink what's possible. This isn't just a replatform. It's a relaunch. Let's build what's next.

If you're ready to learn more about migrating from BigCommerce to Shopify, get in touch today.

[Contact Shopify](#)

And if you're ready to get started today with a partner who knows what it takes, let's get to work.

[Contact MTN Haus](#)