Journal of Experimental and Computational Methods in Applied Sciences, pp. 1–14

Constructing a Unified Data Layer for Scalable 1:1 Personalization in B2C Digital Commerce Environments

Lucas Henrique da Silva [1], Rafael Augusto Pereira [2], Bruno Carvalho dos Santos [3]

1. *Department of Business Administration, University of São Paulo, 1200 Avenida Professor Luciano Gualberto, Butantã, São Paulo 05508-010, Brazil*
2. *Department of Finance and Economics, Federal University of Rio de Janeiro, 219 Avenida Pasteur, Urca, Rio de Janeiro 22290-240, Brazil*
3. *Faculty of Management and Innovation, Pontifical Catholic University of Minas Gerais, 500 Rua do Rosário, Centro, Belo Horizonte 30180-132, Brazil*

Abstract

B2C digital commerce environments operate over heterogeneous transactional, behavioral, and content streams that evolve under high traffic, dynamic assortments, and rapidly changing customer intent. Existing personalization stacks are frequently fragmented, coupling models tightly to application surfaces, and producing inconsistent user experiences as channels proliferate. In this context, a unified data layer that consolidates identity, events, product knowledge, real-time signals, and model outputs into a coherent substrate offers a structured way to support scalable 1:1 personalization without privileging any specific algorithm or channel. This paper examines the construction of such a unified data layer targeted at high-throughput commerce platforms, focusing on architectural primitives, formal data contracts, and latency-aware serving constraints. The discussion emphasizes how this layer can support heterogeneous personalization workloads, including ranking, generation, pricing assistance, and content orchestration, while maintaining tractable guarantees on correctness, observability, and governance. Rather than optimizing for a singular notion of performance, the analysis considers trade-offs between modeling flexibility, operational simplicity, and robustness across a wide variety of traffic patterns and organizational contexts. The resulting formulation aims to clarify how a unified data substrate enables consistent decision-making for 1:1 personalization, under conditions of incomplete information, strict privacy requirements, and incremental system evolution, without prescribing a single algorithmic paradigm or fixed technology stack.

Introduction

B2C digital commerce environments have undergone a sustained transition from static, manually curated storefronts toward highly dynamic, data-driven ecosystems in which almost every surface can, in principle, be individualized [1]. Homepages, search results, category pages, recommendations, banners, email campaigns, push notifications, and conversational agents are all potential carriers of tailored content, pricing hints, and interaction flows. This expansion of opportunities coincides with growth in assortments, diversification of fulfillment options, richer media representations, and integration with external traffic sources. The resulting systems operate under high request volumes, non-stationary demand, heterogeneous user intent distributions, and constraints related to latency, reliability, and regulation. Within this context, 1:1 personalization is less a single algorithmic technique and more an ongoing process of mapping evolving signals to decisions that remain coherent across channels and time.

In practice, most organizations have not designed their personalization capabilities as unified systems from the outset. Instead, they accumulate functionality incrementally [2]. A recommendation widget is deployed on the homepage using a model trained on a particular snapshot of click and purchase logs. A separate team develops search ranking features using another representation of queries, sessions, and products. Email and push messaging adopt segmentation rules that rely on yet another schema for events and identifiers. Promotions and merchandising teams maintain their own audience definitions and product groupings. Experimentation tools write assignments

and outcomes into custom log structures. Over time, each component becomes dependent on its local version of realities such as what constitutes an impression, how a user is identified, how an item is keyed, or which time windows define recency.

This incremental evolution is understandable, given organizational pressures and the heterogeneity of technology stacks, but it introduces structural fragmentation [3]. Fragmentation appears as duplicate or conflicting identity resolution mechanisms, inconsistent event semantics, loosely documented catalog mappings, and parallel feature pipelines whose definitions drift apart. The same user may be treated as distinct entities in different channels; the same product may be categorized differently in recommendation systems and in search; and the same action may be logged under incompatible schemas by various clients. As additional personalization initiatives are layered on, these discrepancies become harder to diagnose. When performance degrades or behaves unexpectedly, teams often attribute issues to model choices while overlooking misalignments in underlying data representations and contracts.

The notion of a unified data layer emerges as a response to this condition. Instead of treating data engineering, feature computation, experimentation logging, and application integration as largely independent concerns for each personalization surface, the unified layer posits a logically centralized substrate that encodes shared concepts, schemas, and access methods. In this view, identity, events, entities, and decisions are modeled once with explicit versioning and governance, then exposed through stable interfaces to all online and offline consumers [4]. Models read features and write outputs in terms of this shared substrate; applications integrate with personalization via contracts that reference unified keys and event types rather than bespoke encodings. The intention is not to introduce a monolithic storage system or to prescribe a singular machine learning framework, but to reduce uncontrolled divergence in how different components interpret and manipulate the same underlying environment.

Conceptually, the unified data layer separates logical structure from physical implementation. Logically, it defines how user and session identifiers map to stable internal keys; which event types exist and which attributes they contain; how catalog entities, content units, and promotional constructs are represented; and how personalization decisions and experimental assignments are recorded. Physically, it may be realized over a combination of technologies such as event logs, columnar warehouses, key-value feature stores, document databases, search indices, and caches, distributed across regions for latency and resilience. The unifying element is not a single database but a set of invariants: stable identifiers, coherent temporal semantics, carefully controlled schema evolution, and mandatory decision logging. Provided these invariants hold, individual components can be replaced, scaled, or optimized independently without silently altering what upstream events or downstream outputs mean. [5]

The relevance of this perspective becomes clearer when considering typical failure modes in personalization. One common pattern involves train-serving skew: models are trained on datasets produced by one set of pipelines and then served using features generated differently or with different freshness guarantees. Another involves silent schema drift, where new fields are added, encodings change, or semantics are reinterpreted without coordinated updates to all consumers. A third involves unobservable decision processes, where it is difficult to retrospectively determine which model or rule produced a given outcome for a given user under specific conditions. These phenomena do not depend on any particular algorithm; they arise from misalignment between parts of the system. A unified data layer aims to constrain such misalignment by making data transformations explicit, auditable, and reusable, and by requiring that all production personalization decisions pass through interfaces that produce structured records suitable for evaluation.

At the same time, constructing and operating a unified data layer introduces its own tensions [6]. Excessive centralization can create bottlenecks, where changes to schemas or feature definitions become slow and contentious. Overly rigid constraints on how teams consume or contribute data can discourage experimentation and lead to parallel, unofficial pipelines. Conversely, minimal governance may allow rapid iteration but reintroduces fragmentation and undermines the very benefits of unification. The design challenge is to define a substrate that is sufficiently opinionated to prevent ambiguous semantics and uncontrolled duplication, yet sufficiently flexible to support diverse modeling approaches, surface-specific requirements, and incremental extension. Achieving this balance is partly technical and partly organizational.

Scalability and latency considerations further complicate the picture. B2C systems must respond within tight time budgets at peak loads, while incorporating as much relevant context as possible [7]. Users expect responsive interfaces, and business constraints limit the computational resources that can be allocated per request. A unified data layer that naively centralizes all features and joins might introduce unacceptable delays or network overhead. For this reason, the logical unification must be compatible with physically distributed deployments, including region-local replicas, sharded feature stores, and hierarchical caches. The abstraction offered to personalization services is that of fast, key-based access to well-defined feature sets and candidate pools, even though these may be served from multiple coordinated infrastructures. A viable design must reconcile the aspiration for coherent semantics with the realities of distributed systems engineering.

The regulatory and normative environment also shapes requirements. Data protection regulations, platform policies, and internal risk tolerances constrain how identifiers can be linked, how long events may be retained, and how

Aspect	Description	Examples
Personalization Surfaces	Dynamic digital commerce touchpoints	Homepages, search results, banners, emails, conversational agents
Challenges	Data heterogeneity, latency, regulation	Non-stationary demand, multiple identifiers, schema drift
Goal	Coherent individualized experience	Consistent decisions across channels and time

Table 1: Scope and challenges of personalization in B2C digital commerce.

Problem Type	Root Cause	Impact
Train-serving skew	Divergent pipelines for train vs. serve features	Model degradation, inconsistent predictions
Schema drift	Uncoordinated changes in event or feature definitions	Breakage in downstream systems
Unobservable decisions	Missing or incomplete logs of actions	Low auditability, hindered diagnostics

Table 2: Typical failure modes in large-scale personalization systems.

sensitive attributes can be used in modeling [8]. These constraints interact directly with the design of identity graphs, event schemas, and decision logging structures. Building privacy and governance controls into the unified data layer, rather than layering them on top of fragmented pipelines, can facilitate consistent enforcement and auditing. It can also clarify which personalization strategies are feasible given available data and permitted operations, reducing misunderstandings between legal, product, and engineering stakeholders.

Within this landscape, the unified data layer is not proposed as a narrowly scoped data warehouse or feature store, nor as a generic enterprise data model. Its scope is more specific: to support 1:1 personalization, broadly construed, in environments where users, items, and contexts interact through repeated, measurable decisions. The layer is concerned with those aspects of data and computation that bear directly on how personalized experiences are constructed, evaluated, and governed. This includes the representation of identities at the granularity relevant for user-level and session-level decision making, the consistent capture of interaction events and outcomes, the timely reflection of catalog and content changes, and the ability to reconstruct which signals and models informed a particular decision at a particular time. [9]

From an analytical standpoint, such a unified substrate enables a more systematic formulation of personalization as a policy learning problem under uncertainty and constraints. When state representations, action spaces, and reward signals are definable in terms of shared schemas and logs, different modeling paradigms can be compared more directly. For instance, a heuristic ranking rule, a supervised learning-to-rank model, a contextual bandit policy, and a constrained reinforcement learning algorithm can all be expressed as mappings from standardized feature sets to actions, evaluated against outcomes recorded in a common structure, and monitored for compliance with the same exposure and fairness constraints. The unified layer does not dictate which method should be preferred, but it provides the conditions under which such methods can be deployed, iterated, and audited without each introducing its own incompatible data subset.

In organizational terms, the introduction of a unified data layer changes how different roles interact. Application engineers integrate with personalization through documented APIs that return results keyed to unified entities; they no longer need separate logic for each underlying model or data source. Data engineers focus on reliable ingestion, storage, and transformation pipelines that populate the unified views, with clear checks and balances [10]. Data scientists design, train, and evaluate models based on declared feature sets that are aligned with what is available online, reducing friction when promoting models to production. Experimentation and analytics teams interpret outcomes through metrics built on the same definitions of events and identities used by models and serving systems. Governance stakeholders see a clearer mapping from regulatory requirements to technical controls embedded in schemas and interfaces. While frictions and trade-offs remain, the shared substrate becomes a common reference point that structures these negotiations.

Background and Problem Formulation

Consider a B2C commerce environment characterized by a set of users, a catalog of items, and a set of channels. Let the user set be denoted by

$$U = \{u_1, \ldots, u_{|U|}\},$$

the item catalog by [11]

$$I = \{i_1, \ldots, i_{|I|}\},$$

Unified View	Logical Role	Key Relations
Identity	Resolve raw identifiers to stable keys	$R_{\text{id}} \subseteq Z \times U$
Event	Record user/system actions as logs	$R_E = \{(k, t, a, p)\}$
Entity	Represent catalog or content entities	$R_I = \{(i, \psi(i))\}$
Decision	Capture personalization outputs	$R_P = \{(q, x, a, \hat{r}, p)\}$

Table 3: Core logical views of the unified data layer.

Layer Property	Purpose	Example Mechanism
Schema Versioning	Ensure consistent interpretation	Explicit feature definitions with timestamps
Temporal Semantics	Maintain coherence over time	Event time ordering, watermark control
Decision Logging	Enable audit and replay	Structured logging of actions and model versions
Governance	Align with regulatory rules	Consent enforcement and retention policies

Table 4: Design invariants and governance mechanisms of the unified data layer.

and the set of channel surfaces by

$$C = \{c_1, \ldots, c_{|C|}\}.$$

User interaction over time generates an event stream in which each event can be represented as a tuple

$$e = (u, i, c, t, a, s),$$

where $u \in U$, $i \in I \cup \{\}$, $c \in C$, t is a timestamp, a is an action type (for example view, click, add-to-cart, purchase), and s is a context vector encoding state such as device information, referrer, or campaign attributes. To maintain the width constraint, the tuple structure can be regarded as consisting of identifiers and a bounded-dimensional context vector without elaborating extended fields in a single expression.

The fundamental objective in 1:1 personalization for such an environment can be described as learning a decision policy π that, for each request, maps from an observable state to a distribution over actions, where actions include ranked lists of items, content variants, prices, and interaction flows. Formally, for a request at time t, define the observable state

$$x_t = (u_t, h_t, s_t), [12]$$

where u_t is the user or anonymous session identifier, h_t is a bounded history summary derived from past events associated with u_t, and s_t is the current context. The decision space may be large: for ranking, it is the set of permutations of candidate items; for messaging, it is the set of templates and delivery channels; for pricing, it is the set of allowed price points for each item. A general stochastic policy can be written as

$$\pi(a \mid x_t),$$

where a denotes an abstract personalization action.

The system seeks to select actions that optimize one or more business-aligned reward signals [13]. Let the reward associated with action a at state x_t be $r(a, x_t)$, which may reflect short-term outcomes such as click or conversion as well as longer-horizon metrics such as repeat purchase or churn reduction. The canonical objective is to find a policy π^* that maximizes the expected cumulative reward

$$\mathbb{E}\Big[\sum_{t=1}^{T} \gamma^{t-1} r(a_t, x_t)\Big],$$

with discount factor $0 < \gamma \leq 1$, subject to constraints such as latency, fairness, exposure diversity, budget caps, and regulatory limitations on feature usage.

In operational deployments, direct optimization of such an objective is not performed in isolation. Multiple teams maintain specialized models for different products, campaigns, and lifecycle stages. Each model consumes input data from bespoke pipelines, often derived from partially overlapping sources. This leads to three structural issues [14]. First, semantic inconsistency arises when distinct representations of the same user or item diverge across systems, leading to incompatible features and disagreement in downstream decisions. Second, observability becomes fragmented, hindering root-cause analysis when behavior changes. Third, constraints on data residency, consent, and retention are difficult to enforce consistently across heterogeneous stacks.

The unified data layer is introduced as a design mechanism to address these issues by centralizing the logical representation of identities, events, features, and decisions while permitting physically distributed storage and computation. Formally, the layer is defined as a set of

Role	Interaction with Unified Layer	Benefit
Application Engineers	Integrate APIs using unified keys	Simplified personalization interfaces
Data Engineers	Maintain ingestion and feature pipelines	Reduced semantic drift, improved lineage
Data Scientists	Train models on unified features	Consistent train/serve behavior
Governance Teams	Audit decisions and compliance	Traceability and enforceable constraints

Table 5: Organizational roles and interactions in unified data layer operations.

typed relations and access operations that satisfy a series of invariants. Let $\mathcal{S}$ denote the schema family, including base relations such as R_U for user profiles, R_I for items, R_E for events, and R_F for materialized features; and derived relations such as R_M for model scores and R_P for applied personalization policies. The schemata are subject to stable keys and versioning rules that ensure that different consumers interpret attributes consistently over time.

A key property is that the unified data layer is not a monolithic storage system but a contract: it defines the canonical shapes and semantics of data accessible to and produced by personalization systems [15]. Implementations may rely on hybrid components, such as streaming logs for raw events, columnar stores for analytical features, key-value stores for online features, search indices for retrieval, and cache layers for low-latency access. The problem becomes one of designing these components, and their interaction, so that the resulting system approximates the behavior of a logically unified, temporally coherent data substrate underpinning 1:1 decision-making.

Unified Data Layer Architecture

The unified data layer is modeled here as a composition of four principal logical views: identity, events, entities, and decisions. Each view is exposed through constrained interfaces optimized for both batch and online consumption. The layer enforces contracts on write and read paths, limiting the introduction of ambiguous fields and ensuring that evolutions are explicit.

The identity view maintains a mapping between raw identifiers and stable, pseudonymous user keys. Represent raw identifiers such as cookies, device IDs, email hashes, and account IDs as elements of a set Z [16]. The identity graph is a relation

$$R_{\text{id}} \subseteq Z \times U,$$

where each pair indicates an association inferred by deterministic rules or probabilistic models with configurable confidence thresholds. Ambiguity in identity resolution is managed by associating edge weights or scores, but each online decision is required to operate on either a single resolved user key or an anonymous key with bounded uncertainty. The unified layer provides deterministic resolution semantics at serving time, for example selecting the highest-confidence cluster under specified constraints, thereby avoiding divergent interpretations across services.

The event view records all user and system actions with minimal loss of fidelity. Events are ingested as an append-only log and projected into normalized and denormalized forms. The base event relation [17]

$$R_E = \{(k, t, a, p)\},$$

includes a stable key k (user or session), timestamp t, action a, and payload p that remains bounded in size. To support modeling, the layer defines deterministic transformations from R_E into feature relations. For each feature family f, an offline projection generates a materialized feature

$$R_F^f = \{(k, \phi_f(k))\},$$

where $\phi_f(k)$ is a function of events up to a cutoff time. Similarly, an online feature view maintains incremental updates of $\phi_f(k)$ using streaming computation. By enforcing that both batch and streaming implementations approximate the same mathematical transform within a defined tolerance, the layer constrains train-serving skew.

The entity view covers products, categories, content assets, and other items exposed to users [18]. Let each item i be associated with attributes in a relation

$$R_I = \{(i, \psi(i))\},$$

where $\psi(i)$ includes static fields, dynamic inventory signals, and embeddings derived from content or behavioral co-occurrence. The unified data layer enforces canonical identifiers for items across systems, preventing duplication due to differing catalog feeds. Incremental updates are captured through change events that maintain temporal consistency between operational catalog systems and personalization consumers.

The decision view captures the outputs of personalization policies and models as first-class entities. A decision instance is represented as

$$d = (q, x, a, \hat{r}, p),$$

where q is a request identifier, x is the observed state, a is the chosen action, $\hat{r}$ is a vector of predicted metrics, and p is a serialized policy descriptor encoding model versions, constraints, and randomization seeds. Logging each decision into a relation [19]

$$R_P = \{d\},$$

closes the loop for evaluation and enables counterfactual analysis, off-policy estimation, and auditability.

From an architectural perspective, the unified data layer constrains interactions as follows. All personalization components consume identity, event, and entity views through stable APIs or query interfaces with bounded latency objectives. All components that apply personalization write their decision outputs into the decision view with required metadata. No model is permitted to introduce latent, private state outside these interfaces for production decisions; instead, stateful components persist their state within or against the unified layer using agreed schemas. This constraint simplifies reasoning about data lineage while allowing technical heterogeneity underneath.

To satisfy latency requirements, the layer adopts a dual-path design [20]. An offline path computes heavy-weight features, embeddings, and aggregates from historical data with relaxed latency tolerances. An online path maintains incremental views of recency-sensitive signals and exposes low-latency retrieval endpoints. Consistency between the two paths is preserved through well-defined watermarks and versioned feature definitions. At any evaluation time, consumers know which feature versions and data freshness intervals are in effect, reducing ambiguity in interpreting model behavior.

Observability is embedded as a first-class design concern. For each relation in the unified data layer, the system maintains compact summary statistics and anomaly signals over distributions of keys and values. Feature definitions are registered along with ownership metadata [21]. All writes to critical relations are subject to schema validation and invariant checks that can reject or quarantine malformed updates. These mechanisms reduce the probability that silent data drift or accidental schema changes propagate unchecked into personalization decisions.

Modeling 1:1 Personalization over the Unified Data Layer

Given the unified data layer, personalization policies can be expressed with greater precision. Consider a generic personalization request at time t for user key k in context s_t. The serving system retrieves identity-resolved attributes, historical features, and candidate entities from the unified layer. Denote by $\phi(k)$ the concatenation of user-level features from the relevant R_F^f relations, and by $\psi(i)$ the item attributes and representations from R_I. For ranking scenarios, define a scoring function [22]

$$g_\theta(k, i, s_t),$$

parameterized by θ, which maps the joint representation into a scalar utility estimate. The ranking policy then induces a permutation by sorting candidate items according to

$$g_\theta(k, i, s_t).$$

To treat personalization as a policy learning problem, introduce a reward function $r(k, i, t)$ describing the realized outcome. A simplified empirical risk objective can be written as

$$\min_\theta \ \frac{1}{N} \sum_{j=1}^{N} \ell\big(g_\theta(k_j, i_j, s_j), y_j\big),$$

where (k_j, i_j, s_j, y_j) are drawn from historical interactions in R_E joined with R_P, and ℓ is an appropriate loss function [23]. Because the unified data layer standardizes how decisions and outcomes are logged, counterfactual estimators such as inverse propensity weighting can be incorporated without additional instrumentation outside the layer.

Beyond single-objective ranking, many B2C environments require policies that balance multiple metrics and constraints. Let each candidate action a at state x have a vector of predicted outcomes

$$\hat{r}(a, x) \in \mathbb{R}^m,$$

for example revenue, margin, satisfaction proxies, or return risk. A constrained optimization formulation can be expressed as

$$\max_\pi \ \mathbb{E}[w^\top \hat{r}(a, x)]$$

subject to

$$\mathbb{E}[c_j(a, x)] \le \kappa_j$$

for constraint functions c_j and thresholds κ_j [24]. To comply with the width constraint, the expressions remain compact; each constraint is evaluated using features and statistics available from the unified layer, such as exposure counts for regulated categories or budgets for promotional impressions.

In scenarios where contextual bandit or reinforcement learning approaches are applied, the unified data layer supports policy learning by providing factored representations of state and curated logs of past actions and outcomes. For a contextual bandit with context x_t, action set A_t, and reward r_t, the unified layer ensures that the logged probability $\pi_b(a_t \mid x_t)$ of the behavior policy is captured in R_P. This allows estimation of a new policy π via objectives of the form

$$\hat{V}(\pi) = \frac{1}{N} \sum_{t=1}^{N} \frac{\pi(a_t \mid x_t)}{\pi_b(a_t \mid x_t)} r_t,$$

with stabilized variants as required. Because both π_b and π operate against the same unified schema, the risk of mismatched feature definitions is mitigated. [25]

For sequence-aware personalization, such as session-based recommendations or conversational flows, the unified data layer captures event sequences in R_E and enables consistent construction of sequence features. A sequence model may embed the ordered event history for user k up to time t as a vector

$$h_t = f_{\text{seq}}(e_{1:t}),$$

where f_{seq} represents a recurrent, attention-based, or convolutional architecture. The resulting representation is written back as part of a feature relation so that different decision policies can reuse it without redefining the underlying event transformations.

Generative models may also operate over the unified layer for tasks such as personalized messaging or landing page synthesis. Input conditioning vectors, containing user intent summaries, item attributes, and contextual signals, are assembled through unified queries. Generated outputs and associated metadata, such as safety filters and model versions, are logged into the decision view. This approach maintains traceability for content presented to individual users without requiring each consuming team to develop separate logging standards. [26]

The central modeling consequence of the unified data layer is that all personalization methods, whether based on supervised learning, bandits, reinforcement learning, or generative modeling, share a common set of primitives for state construction, candidate representation, and decision logging. This reduces the space of silent incompatibilities and provides a stable foundation for incremental improvement while accommodating a wide range of algorithmic choices.

Scalability, Consistency, and Latency Analysis

To support 1:1 personalization at scale, the unified data layer must operate under stringent performance and reliability constraints. Let the peak request rate be λ_q queries per second and the latency budget per personalization call be $L_{\max}$. The end-to-end processing time can be decomposed into contributions from identity resolution, feature retrieval, candidate retrieval, scoring, and post-processing. If T_{id}, T_{feat}, T_{cand}, and T_{score} denote the respective service times, then a simple bound requires

$$T_{\text{id}} + T_{\text{feat}} + T_{\text{cand}} + T_{\text{score}} \leq L_{\max}.$$

The unified data layer influences these components by placing data structures close to computation, minimizing cross-region round trips, and constraining feature joins to pre-materialized views.

Horizontal scalability can be analyzed by modeling each logical view as a sharded service. Suppose the identity and feature stores are partitioned by user key, while item data are partitioned by item key [27]. Under an approximately uniform distribution of keys, the load per shard for a view with S shards is λ_q/S. To ensure capacity under peak load with utilization threshold $\rho_{\max}$, each shard must satisfy

$$\frac{\lambda_q}{S\mu} \leq \rho_{\max},$$

where μ is the service rate. This relationship can be inverted to determine the minimal number of shards needed for a given performance envelope, assuming service times remain stable with added capacity. Because all personalization components depend on these views, their scaling properties directly shape feasible policy complexity.

Consistency considerations arise from the temporal alignment of features and events. In distributed settings, strong consistency for all reads may be impractical due to latency and availability constraints. The unified data layer instead defines explicit consistency contracts: for example, event streams may guarantee at-least-once delivery and monotonic ordering within partitions, while feature views may specify a freshness bound Δ such that all reads observe data at least up to time $t - \Delta$ [28]. For many personalization tasks, bounded staleness with known Δ is sufficient, provided that both offline training and online serving respect the same semantics. Deviations can be modeled as perturbations in the input space, and robustness analyses can estimate the sensitivity of policies to such perturbations.

To formalize the effect of staleness, consider a feature vector ϕ_t that ideally would be computed from all events up to time t, but in practice is computed using events only up to $t - \Delta$. The error induced by staleness can be expressed as

$$\epsilon_t = \phi_t - \phi_{t-\Delta}.$$

If the scoring function g_θ is Lipschitz continuous with constant L_g in its feature arguments, then the change in predicted utility satisfies

$$\left| g_\theta(\phi_t) - g_\theta(\phi_{t-\Delta}) \right| \leq L_g \|\epsilon_t\|.$$

This provides a way to reason about the impact of delayed updates on decision quality and to select acceptable freshness bounds for different feature classes. [29]

Fault tolerance is addressed by designing the unified data layer so that failures in individual views or components degrade gracefully rather than causing systemic outages. For example, if online feature retrieval fails for a fraction of requests, the serving system can fall back to cached or baseline features that are still derived from the unified schema. The performance effect can be modeled by analyzing the mixture of high-fidelity and fallback responses and their associated reward distributions. Similarly, progressive rollouts of schema changes and feature definitions are orchestrated through versioned contracts, preventing incompatible updates from being applied simultaneously across training and serving paths.

Privacy, Governance, and Robustness

Any unified data layer for 1:1 personalization in B2C contexts must operate under regulatory and ethical constraints

related to privacy, consent, transparency, and fairness. Treating these constraints as afterthoughts at the application layer leads to inconsistent enforcement. Embedding them directly into the structure of the unified data layer provides a more systematic mechanism for constraining model behavior. [30]

Privacy is approached through a combination of pseudonymization, minimization, and controlled joins. The identity view ensures that stable user keys are generated using irreversible transformations of direct identifiers, and that access to linkage between stable keys and sensitive identifiers is restricted to a narrowly scoped subsystem. Feature definitions are expressed in a way that avoids unnecessary inclusion of sensitive attributes. For example, instead of exposing raw location or full birthdate, the unified layer exposes coarser or derived attributes whose informational contribution to personalization can be evaluated more easily.

Governance is implemented via schema-level access policies and lineage. Each relation and attribute in the unified data layer carries metadata describing allowable use cases and retention periods. Personalization models declare which attributes they consume, and automated checks validate that these declarations are compatible with attribute-level policies before deployment [31]. Because all decision outputs are logged with policy descriptors, auditors can reconstruct which attributes and models influenced specific decisions without reconstructing ad hoc traces from disparate systems.

Robustness considerations extend beyond infrastructure reliability to include distributional shifts and adversarial interactions. The unified data layer supports robust modeling by maintaining historical distributions and drift indicators for key features and targets. When shifts are detected, retraining or policy adaptation strategies can be executed using consistent datasets derived from the same substrate that serves production traffic. This alignment reduces discrepancies between evaluation and deployment conditions.

To discourage overfitting to narrow segments or transient artifacts, policies can be regularized using constraints expressed over aggregates computed within the unified layer. For instance, exposure diversity requirements can be encoded as soft penalties or hard bounds on the proportion of traffic allocated to particular items or categories over defined horizons [32]. Mathematically, if x indexes contexts and a indexes items, a simple diversity constraint might require that

$$\mathbb{E}[\mathbf{1}\{a = i\}] \leq \delta_i,$$

for specified upper bounds δ_i. Enforcement can rely on online counters derived from decision logs, thus integrating seamlessly with the decision view.

The integration of privacy, governance, and robustness into the schema and interfaces of the unified data layer does not eliminate the need for careful organizational practices, but it provides concrete mechanisms to express and verify constraints. Since all models and applications interface through the layer, policy changes can be enacted and audited centrally, reducing the risk of inconsistent interpretations across teams.

Operationalization Patterns and Cross-Functional Integration

Operationalizing a unified data layer for scalable 1:1 personalization requires careful alignment between abstract architectural principles and the routines of engineering, data science, product management, and governance functions. While the unified layer is defined as a logical substrate, its practical effectiveness depends on how organizations encode experiments, manage rollout strategies, translate high-level objectives into machine-readable constraints, and reconcile heterogeneous priorities that arise in large B2C environments [33]. This section examines operational patterns that emerge when the unified layer is treated as the central medium for observation and control, emphasizing neutral trade-off analysis rather than asserting prescriptive best practices.

A starting point for operationalization is the formalization of objectives in a way that can be implemented consistently across multiple decision surfaces. Let a configuration of personalization behavior at time t be represented by parameters Θ_t, encoding model weights, thresholding rules, constraints, and traffic allocations. For a given horizon, define an aggregate objective

$$J(\Theta_t) = \mathbb{E}\big[R(\Theta_t, Z_t)\big],$$

where Z_t denotes the stochastic environment comprising user arrivals, catalog states, and exogenous conditions, and R is a reward function capturing metrics such as conversion, margin, and engagement. In practice, organizations operate with several such metrics that cannot be collapsed into a single scalar without loss of nuance. The unified data layer enables representation of these metrics as jointly observed quantities attached to decision records in R_D, such that each deployed configuration Θ_t is associated with empirical distributions of outcomes [34]. Operational decision making then becomes an iterative adjustment of Θ_t informed by these distributions, rather than discrete and opaque code changes in isolated systems.

Stable experimentation is central to this adjustment process. Consider an experiment where traffic is randomly partitioned into variants indexed by $v \in \{0, 1, \dots\}$, each associated with a policy π_v. Assignment is implemented as a deterministic function of keys and experiment definitions, and is logged in the decision schema. For request context x_t, variant v is chosen, action a_t is drawn from π_v, and realized reward vector r_t is attributed. Because all relevant elements are recorded within the unified layer, estimates of differences

$$\Delta_v = \mathbb{E}[r_t \mid v] - \mathbb{E}[r_t \mid 0]$$

Model Type	Core Mechanism	Unified Layer Support
Supervised Ranking	Learn $g_\theta(k, i, s_t)$ from logged events	Consistent joins between R_E, R_I, R_P ensure coherent labels
Contextual Bandits	Estimate $\hat{V}(\pi)$ via importance weighting	Unified logs capture $\pi_b(a_t\|x_t)$ for unbiased evaluation
Sequence Models	Encode ordered user events $h_t = f_{\text{seq}}(e_{1:t})$	Sequential event access and shared feature views
Generative Models	Produce text or layout conditioned on context	Input conditioning and traceable logging via decision view

Table 6: Personalization modeling paradigms unified through shared data schemas.

Optimization Type	Objective	Constraint Mechanism
Single Objective	$\min_\theta \frac{1}{N}\sum_j \ell(g_\theta(k_j, i_j, s_j), y$	Standard empirical risk
Multi-Objective	$\max_\pi \mathbb{E}[w^\top \hat{r}(a, x)]$	Enforced $\mathbb{E}[c_j(a, x)] \leq \kappa_j$ via unified metrics
Fairness	Maintain balanced exposure	Online counters within R_P monitor compliance
Budget Control	Cap promotion or spend	Constraints applied through aggregate statistics

Table 7: Policy optimization formulations leveraging unified data layer statistics.

can be computed in a reproducible manner, with shared definitions of cohorts and time windows. The absence of duplicated logging schemes reduces ambiguity in cross-team interpretations [35]. Over time, such experimentation histories inform priors on which classes of models or policies are productive under specific traffic and catalog conditions, but these inferences remain grounded in data that share schemas and lineage.

Cross-functional integration becomes salient when product teams articulate qualitative objectives, such as promoting specific assortments, avoiding excessive repetition, or aligning with brand guidelines, which must be translated into quantitative constraints. Within the unified framework, such constraints act on relations in the data layer. For example, a requirement that a fraction of personalized surface exposures correspond to items from a designated campaign set $\mathcal{C}$ can be modeled as a bound on

$$\frac{1}{N}\sum_{t=1}^{N} \mathbf{1}\{a_t \in \mathcal{C}\},$$

while an upper limit on repetition for the same user and item over a period can be modeled via constraints on counts derived from R_D. These expressions allow qualitative goals to be enforced as policies evaluated through unified logs instead of ad hoc checks per application. Trade-offs between such constraints and primary objectives are handled explicitly during model selection and configuration tuning.

An important pattern involves the design and management of shared feature definitions [36]. In the absence of a unified layer, teams frequently re-implement feature logic, leading to subtle divergences. Under the unified approach, feature definitions are specified as declarative transformations over event and entity relations, each with an identifier, owner, and version history. When a definition changes, both training pipelines and online feature services adopt the new version in a coordinated way, and decision records include references to the versions used at serving time. Let $\phi^{(k)}$ denote a particular versioned feature map. A model g_θ that relies on this map can be represented as $g_\theta \circ \phi^{(k)}$. When migrating to a revised feature map $\phi^{(k+1)}$, experiments can compare $g_\theta \circ \phi^{(k)}$ and $g_\theta \circ \phi^{(k+1)}$ under controlled conditions. This notation becomes operational when encoded into configuration metadata and validated at deployment, ensuring that changes to feature semantics are tracked and interpretable.

Rollout strategies for new personalization components benefit from the observability afforded by the unified layer. A typical pattern is progressive allocation of traffic while monitoring key indicators not only of business outcomes but also of system performance and distributional stability [37]. Define a rollout schedule where variant v receives a time-varying traffic share $\alpha_v(t)$. The unified decision view records these assignments, enabling post hoc reconstruction of rollout dynamics. In addition to standard convergence metrics, teams can compute divergence measures between feature or score distributions under old and new policies. For instance, a bounded change constraint may require that for specific monitored features, their empirical means under new policy remain within a tolerance band relative to a baseline during early rollout. Such constraints can be formalized using simple inequalities that refer only to relations in the unified layer and evaluated without custom instrumentation.

Another operational theme is incident response. When anomalous behavior is detected, for example a sudden drop in click-through on a surface or increased latency, the unified data layer provides a structured basis for diagnosis [38]. Observability tools can traverse from

Performance Aspect	Symbol / Bound	Implication
Latency Budget	$T_{\text{id}}+T_{\text{feat}}+T_{\text{cand}}+T_{\text{score}} \leq L_{\max}$	Tight response constraint per request
Shard Load	$\lambda_q/(S\mu) \leq \rho_{\max}$	Determines minimal shard count
Staleness Error	$\epsilon_t = \phi_t - \phi_{t-\Delta}$	Captures feature freshness deviation
Prediction Sensitivity	$\lvert g_\theta(\phi_t) - g_\theta(\phi_{t-\Delta})\rvert \leq L_g\lVert\epsilon_t\rVert$	Quantifies latencyaccuracy tradeoff

Table 8: Scalability and latency analysis using unified layer parameters.

Reliability Feature	Description	Outcome
Fallback Paths	Cached or baseline features for partial outages	Graceful degradation
Schema Versioning	Versioned feature definitions	Safe progressive rollouts
Consistency Contracts	Defined freshness bounds Δ	Predictable staleness semantics
Monitoring	Drift and anomaly detection on relations	Early fault detection and rollback

Table 9: Mechanisms ensuring operational robustness of the unified data layer.

symptoms expressed in R_D and downstream metrics back to potential causes in event ingestion, identity mappings, catalog updates, feature computations, or scoring services. Because each of these components interacts through registered schemas and transformations, their outputs can be inspected for violations of documented invariants, such as unexpected null rates or key mismatches. Let Q_t denote a vector of quality indicators derived from unified relations at time t. Operational health targets can be expressed as bounds on expectations or quantiles of Q_t. When these are violated, alerts reference the shared contracts rather than isolated implementations, improving coordination among teams responsible for different segments of the pipeline.

Cross-functional integration also manifests in the handling of exceptions and edge cases, such as cold-start users, sparse regions of the catalog, or temporal spikes in demand. In the unified framework, these conditions are defined relative to the same underlying relations [39]. A cold-start user may be defined as a key with no entries in R_E before time t or with limited feature availability in registered views. Strategies for such users, including popularity-based or contextual-only recommendations, are encoded as policies with explicit conditions on observed data. Similarly, long-tail items may be classified via counts in event or order relations, and specialized treatment (for example constrained exploration or protective exposure thresholds) can be registered as rules whose indicators are computed centrally. Because definitions of cold-start and long-tail derive from shared data, product, analytics, and engineering stakeholders can discuss them concretely without relying on divergent thresholds scattered across codebases.

The unified data layer also influences collaboration around compliance and legal review. When questions arise regarding how a particular decision was made for a user or group of users, organizations need to reconstruct relevant inputs, transformations, and models. With unified decision records, identity mappings, and feature schemas, this reconstruction is less dependent on fragile forensic queries over disparate systems [40]. A single structured query can often enumerate, for a defined cohort, the policies active, the classes of features accessed, and the realized outcomes over a time window. Let $H(u, t_1, t_2)$ denote the set of decisions involving user key u between times t_1 and t_2. Under the unified design, H is computable by standard joins over R_D and relevant relations, yielding an auditable narrative of personalization behavior. Legal and compliance teams can review these narratives against documented policies, and their feedback can be translated into updated constraints or schema annotations that apply uniformly going forward.

From a resource allocation perspective, the unified data layer creates a common platform on which infrastructure investment decisions can be made. Since personalization workloads across surfaces draw on the same underlying stores and transformations, projected increases in traffic or catalog size can be modeled as changes in shared parameters such as λ_q, feature cardinalities, or replication factors. Capacity planning models estimate the required scaling of storage and serving clusters to maintain target latencies and quality metrics [41]. The relationship between resource usage and decision performance becomes more visible because both are expressed in terms of unified constructs. Teams proposing new personalization capabilities can evaluate their incremental load on shared systems and justify optimizations in terms recognized by other stakeholders.

Governance Aspect	Implementation	Benefit
Privacy	Pseudonymized identifiers, minimized joins	Reduced reidentification risk
Access Control	Schema-level policies, metadata lineage	Enforced compliance per attribute
Auditability	Decision logs with model descriptors	Full traceability of personalization actions
Fairness & Robustness	Exposure and drift monitoring via R_P	Stable behavior under shift or constraints

Table 10: Integrated privacy, governance, and robustness controls in the unified data layer.

In addition, the unified layer provides a foundation for knowledge transfer and onboarding. Documentation of schemas, feature definitions, and decision contracts serves as a stable reference for practitioners entering the organization or moving between teams. Rather than learning multiple competing event taxonomies or identity schemes, new contributors engage with a single coherent model. Examples of training datasets, evaluation scripts, and experiment analyses built on top of the unified relations illustrate idioms that can be reused and adapted. The presence of unifying abstractions does not eliminate domain-specific complexity, but it structures it in a way that is more accessible and less reliant on tacit knowledge embedded solely in legacy code or isolated reports. [42]

Finally, operationalization over a unified data layer encourages a more deliberate approach to change management. Because schemas, transformations, and policies are treated as versioned artifacts, proposals for modification can be reviewed for compatibility with established contracts. Experiments can be scoped explicitly to evaluate the impact of proposed changes before they are generalized. Where negative interactions are discovered, for example between certain feature updates and downstream models, these are documented in the same framework, enabling future projects to anticipate similar interactions. Over time, this disciplined approach can lead to personalization ecosystems that favor incremental, observable adjustments over abrupt and opaque shifts, while still accommodating innovation and adaptation.

In summary, operational patterns and cross-functional practices that leverage a unified data layer revolve around using shared abstractions as the substrate for experimentation, configuration management, incident response, constraint enforcement, regulatory alignment, and capacity planning. Rather than elevating the architecture itself as a guarantee of performance, these patterns highlight how explicit contracts and consistent data representations can support the practical coordination required to sustain scalable 1:1 personalization in complex B2C digital commerce settings. [43]

Evaluation Methodology and Discussion

Evaluating a unified data layer for scalable 1:1 personalization requires examining both system-level and decision-level performance. System-level evaluation focuses on latency, throughput, availability, and data quality metrics associated with the core views. Decision-level evaluation examines the impact of personalized policies on target outcomes while monitoring for unintended effects.

A coherent offline evaluation pipeline can be constructed by sampling request and context records from R_P and joining them with corresponding features and outcomes from R_E and R_F. Because the logging and feature computation are standardized, counterfactual estimators can be applied more reliably than in fragmented environments. For example, policies trained on historical data can be evaluated using held-out logs without reconstructing bespoke datasets for each experiment. The unified layer therefore enables more consistent offline estimation, although it does not obviate the need for cautious interpretation. [44]

Online evaluation is conducted through controlled experiments where traffic is partitioned between baseline and treatment policies that all consume data from the unified layer. The experimental assignments and policy variants are logged explicitly in R_P, allowing for reconstruction of experiment cohorts and analysis windows. Metrics of interest may include click-through, conversion, revenue, margin, session length, and longer-term engagement proxies, along with operational measures such as latency distributions and error rates. Since the data definitions are shared, changes in metric behavior can be linked more directly to policy changes rather than confounding shifts in data extraction.

One aspect of evaluation involves quantifying the effect of the unified data layer itself, separate from specific modeling techniques. Comparative analysis can consider conditions before and after adoption of the unified schema, controlling for obvious confounders. Changes in deployment lead times, the frequency and severity of data incidents, and the variance of model performance across channels can provide indirect evidence of the layer's influence on overall system behavior [45]. However, such evaluations are inherently context-dependent and must be interpreted with care, avoiding over-attribution to architectural factors alone.

A further dimension involves assessing how well the unified data layer supports heterogeneous policy requirements. For example, lifecycle messaging, on-site ranking, and in-session search assistance may require different optimization horizons and risk tolerances. Evaluating whether the same substrate provides suitable features, latency, and observability for each use case helps identify structural gaps. Where deficiencies exist, they can often be addressed by refining feature definitions, extending schema metadata, or adjusting caching and replication strategies, rather than introducing new siloed pipelines.

Discussion of limitations is important for a balanced view. A unified data layer introduces coordination overhead and requires careful governance to avoid becoming a bottleneck [46]. Overly rigid schema control can slow innovation if all changes must traverse centralized review. Conversely, overly permissive evolution can reintroduce inconsistency. The architecture described here assumes that organizations can maintain disciplined contracts and invest in platform capabilities to support them. In environments where such discipline is impractical, partial or hybrid approaches may be more feasible, applying unified principles to critical components while allowing local variation elsewhere.

Conclusion

This paper has presented a technical formulation of a unified data layer to support scalable 1:1 personalization in B2C digital commerce environments. By treating identity, events, entities, and decisions as structured views connected through explicit contracts, the unified layer provides a coherent substrate for diverse personalization policies without prescribing a particular algorithmic agenda. The analysis has outlined how shared schemas and interfaces can reduce semantic discrepancies, enable more reliable offline and online evaluation, and support the integration of supervised, bandit, reinforcement learning, and generative approaches against consistent data. [47]

The mathematical formulations introduced for policy learning, constrained optimization, and robustness under bounded staleness illustrate how the unified data layer enables clearer reasoning about trade-offs between latency, accuracy, and compliance. The consideration of horizontal scalability and availability highlights that such a layer can be realized through distributed components, provided that their behavior collectively approximates a logically centralized system with predictable performance. Integration of privacy, governance, and robustness constraints into the data model demonstrates how regulatory and ethical requirements can be expressed as first-class properties rather than external impositions.

At the same time, the unified data layer is not positioned as a singular solution to all personalization challenges. Its effectiveness depends on careful schema design, disciplined implementation of feature definitions, and alignment between technical and organizational practices. Future work in specific deployments can focus on empirical characterization of the observed gains and costs associated with this architectural pattern and on refining mechanisms that facilitate evolution of schemas, models, and policies while preserving the stability of the underlying substrate. Within these considerations, a unified data layer offers a structured basis for developing and operating 1:1 personalization capabilities in a manner consistent with the operational and regulatory constraints typical of contemporary B2C digital commerce platforms [48].

Conflict of interest

Authors state no conflict of interest.

References

[1] S. H. Kukkuhalli, "Enabling customer 360 view and customer touchpoint tracking across digital and non-digital channels," *Journal of Marketing & Supply Chain Management*, vol. 1, no. 3, 2022.

[2] D. R. G. Anand and D. M. K. M, "A study on the rise of e-commerce from indian perspective- flipkart and its liquidation market," *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, no. 8, pp. 715–723, Aug. 31, 2022. DOI: 10.22214/ijraset.2022.46264

[3] M. H. Nguyen, D. Pojani, D. Q. Nguyen-Phuoc, and B. N. Thi, "What if delivery riders quit? challenges to last-mile logistics during the covid-19 pandemic.," *Research in transportation business & management*, vol. 47, pp. 100 941–100 941, Dec. 20, 2022. DOI: 10.1016/j.rtbm.2022.100941

[4] A. A. Ozok and J. Wei, "An empirical comparison of consumer usability preferences in online shopping using stationary and mobile devices: Results from a college student population," *Electronic Commerce Research*, vol. 10, no. 2, pp. 111–137, Apr. 21, 2010. DOI: 10.1007/s10660-010-9048-y

[5] Y. Li, X. Lu, K.-M. Chao, Y. Huang, and M. Younas, "The realization of service-oriented e-marketplaces," *Information Systems Frontiers*, vol. 8, no. 4, pp. 307–319, Oct. 28, 2006. DOI: 10.1007/s10796-006-9006-3

[6] C. Liao, P. Palvia, and H.-N. Lin, "Stage antecedents of consumer online buying behavior," *Electronic Markets*, vol. 20, no. 1, pp. 53–65, Feb. 24, 2010. DOI: 10.1007/s12525-010-0030-2

[7] F. Liébana-Cabanillas, F. Muñoz-Leiva, and J. Sánchez-Fernández, "A global approach to the analysis of user behavior in mobile payment systems in the new electronic environment," *Service Business*, vol. 12, no. 1, pp. 25–64, Feb. 8, 2017. DOI: 10.1007/s11628-017-0336-7

[8] A. Wieneke and C. Lehrer, "Generating and exploiting customer insights from social media data," *Electronic Markets*, vol. 26, no. 3, pp. 245–268, Jun. 9, 2016. DOI: 10.1007/s12525-016-0226-1

[9] H.-L. Chu, Y.-S. Deng, and M.-C. Chuang, "Investigating the persuasiveness of e-commerce product pages within a rhetorical perspective," *International Journal of Business and Management*, vol. 9, no. 4, pp. 31–, Mar. 24, 2014. DOI: 10.5539/ijbm.v9n4p31

[10] L. Kaovská and V. Bumberová, "Study of smes from electrical engineering industry providing smart services in the czech republic," *SHS Web of Conferences*, vol. 92, pp. 05 013–, Jan. 13, 2021. DOI: 10.1051/shsconf/20219205013

[11] G. erniauskait, J. Sabaityt, and E. Leonaviien, "Use of generational theory for the assessment of involvement in international electronic commerce activities," *Challenges to national defence in contemporary geopolitical situation*, vol. 2020, no. 1, pp. 272–281, Oct. 16, 2020. DOI: 10.47459/cndcgs.2020.36

[12] S. N. Marzuki and I. M. Yasin, "Factors influencing online shoppers to shop at marketplace or website: A qualitative approach," *Asian Journal of Business and Management*, vol. 9, no. 4, Nov. 3, 2021. DOI: 10.24203/ajbm.v9i4.6736

[13] M. P. Pieroni, T. C. McAloone, and D. C. A. Pigosso, "Configuring new business models for circular economy through productservice systems," *Sustainability*, vol. 11, no. 13, pp. 3727–, Jul. 8, 2019. DOI: 10.3390/su11133727

[14] D. M. Constantin, F. N. Valentina, D. Anisoara, and G. Raluca, "Improving the relationships between organizations and their customers using digital multichannel communication and mathematical simulation," *ECONOMIC COMPUTATION AND ECONOMIC CYBERNETICS STUDIES AND RESEARCH*, vol. 53, no. 1/2019, pp. 265–280, Mar. 19, 2019. DOI: 10.24818/18423264/53.1.19.17

[15] D. S. Johnson, D. Sihi, and L. Muzellec, "Implementing big data analytics in marketing departments: Mixing organic and administered approaches to increase data-driven decision making," *Informatics*, vol. 8, no. 4, pp. 66–, Sep. 26, 2021. DOI: 10.3390/informatics8040066

[16] S. H. Kukkuhalli, "Improving digital sales through reducing friction points in the customer digital journey using data engineering and machine learning," *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences*, vol. 10, no. 3, 2022.

[17] R. R. Ahmed, D. Streimikiene, G. Berchtold, J. Vveinhardt, Z. A. Channar, and R. H. Soomro, "Effectiveness of online digital media advertising as a strategic tool for building brand sustainability: Evidence from fmcgs and services sectors of pakistan," *Sustainability*, vol. 11, no. 12, pp. 3436–, Jun. 21, 2019. DOI: 10.3390/su11123436

[18] R. Misra, R. Mahajan, N. Singh, S. Khorana, and N. P. Rana, "Factors impacting behavioural intentions to adopt the electronic marketplace: Findings from small businesses in india.," *Electronic markets*, vol. 32, no. 3, pp. 1639–1660, Aug. 23, 2022. DOI: 10.1007/s12525-022-00578-4

[19] L. Hoeckesfeld, A. B. Sarquis, A. T. Urdan, and E. D. Cohen, "Contemporary marketing practices approaches in the professional services industry in brazil," *Revista Pensamento Contemporâneo em Administração*, vol. 14, no. 1, pp. 56–75, Mar. 31, 2020. DOI: 10.12712/rpca.v14i1.38890

[20] N. Chawla and B. Kumar, "E-commerce and consumer protection in india: The emerging trend," *Journal of business ethics : JBE*, vol. 180, no. 2, pp. 1–24, Jul. 9, 2021. DOI: 10.1007/s10551-021-04884-3

[21] A. Azmy and null null, "Optimization of technology mastery as part of increasing employee performance in the e-commerce industry," *Zenodo (CERN European Organization for Nuclear Research)*, Feb. 5, 2022. DOI: 10.5281/zenodo.5977629

[22] M. Dachyar, T. Y. M. Zagloel, and L. R. Saragih, "Enterprise architecture breakthrough for telecommunications transformation: A reconciliation model to solve bankruptcy.," *Heliyon*, vol. 6, no. 10, e05273–, Oct. 19, 2020. DOI: 10.1016/j.heliyon.2020.e05273

[23] null null, null null, and null null, "Disintermediation a threat or opportunity for traditional pakistani travel agencies (a case study on travel mate)," *Zenodo (CERN European Organization for Nuclear Research)*, Jan. 3, 2022. DOI: 10.5281/zenodo.5814644

[24] G. Börühan, P. Ersoy, and I. O. Yumurtaci, "What is wrong with private shopping sites? evidence from turkey," *Pressacademia*, vol. 4, no. 3, pp. 401–401, Sep. 30, 2015. DOI: 10.17261/pressacademia.2015313062

[25] O. Zinchenko, I. Privarnikova, and A. Samoilenko, "Adaptive strategic management in a digital business environment," *Baltic Journal of Economic Studies*, vol. 8, no. 3, pp. 78–85, Sep. 30, 2022. DOI: 10.30525/2256-0742/2022-8-3-78-85

[26] B. Song and Z.-h. Zhao, "Online holiday marketings impact on purchase intention: Chinas double-11 shopping carnival," *Journal of Management and Humanity Research*, vol. 1, pp. 11–24, Jun. 18, 2019. DOI: 10.22457/jmhr.v01a02102p

[27] null Sailaja, P. R. P, and S. A, "Business intelligence and analytics: Recent trends and benefits in retailing," *Journal of Management and Science*, vol. 6, no. 1, pp. 11–26, Jun. 30, 2017. DOI: 10.26524/jms.2016.3

[28] M. H. Mussa, "The impact of artificial intelligence on consumer behaviors an applied study on the online retailing sector in egypt,", vol. 50, no. 4, pp. 293–318, Dec. 9, 2020. DOI: 10.21608/jsec.2020.128722

[29] "Article abstracts," *Interactive Marketing*, vol. 5, no. 3, pp. 288–302, Jan. 1, 2004. DOI: 10.1057/palgrave.im.4340246

[30] H. Brdulak and A. Brdulak, "Challenges and threats faced in 2020 by international logistics companies operating on the polish market," *Sustainability*, vol. 13, no. 1, pp. 359–, Jan. 3, 2021. DOI: 10.3390/su13010359

[31] B.-N. Yan, T.-S. Lee, and T.-P. Lee, "Analysis of research papers on e-commerce (2000—2013): Based on a text mining approach," *Scientometrics*, vol. 105, no. 1, pp. 403–417, Aug. 13, 2015. DOI: 10.1007/s11192-015-1675-6

[32] A. Tyagi, "A study on customer satisfaction with reference to myntraă.," *International Journal of Social Science and Economic Research*, vol. 6, no. 7, pp. 2211–2243, Jul. 30, 2021. DOI: 10.46609/ijsser.2021.v06i07.014

[33] P. Braun, "Networking tourism smes: E-commerce and e-marketing issues in regional australia.," *Information Technology & Tourism*, vol. 5, no. 1, pp. 13–23, Jan. 1, 2002. DOI: 10.3727/109830502108751028

[34] R. Z. Szabo et al., "Industry 4.0 implementation in b2b companies: Cross-country empirical evidence on digital transformation in the cee region," *Sustainability*, vol. 12, no. 22, pp. 9538–, Nov. 16, 2020. DOI: 10.3390/su12229538

[35] M. Dehghani, A. M. Abubakar, and M. Pashna, "Market-driven management of start-ups: The case of wearable technology," *Applied Computing and Informatics*, vol. 18, no. 1/2, pp. 45–60, Jul. 20, 2020. DOI: 10.1016/j.aci.2018.11.002

[36] M. Barna and B. Semak, "Main trends of marketing innovations development of international tour operating," *Baltic Journal of Economic Studies*, vol. 6, no. 5, pp. 33–41, Dec. 2, 2020. DOI: 10.30525/2256-0742/2020-6-5-33-41

[37] T. T. Haile and M. Kang, "Mobile augmented reality in electronic commerce: Investigating user perception and purchase intent amongst educated young adults," *Sustainability*, vol. 12, no. 21, pp. 9185–, Nov. 4, 2020. DOI: 10.3390/su12219185

[38] A. F. Delawari, "Online-business in afghanistan, current trend and challenges ahead: A conceptual study," *Journal of Emerging Economies and Islamic Research*, vol. 7, no. 2, pp. 44–, May 31, 2019. DOI: 10.24191/jeeir.v7i2.8764

[39] Y. Shang et al., "The nexuses between social media marketing activities and consumers' engagement behaviour: A two-wave time-lagged study.," *Frontiers in psychology*, vol. 13, pp. 811 282–, Apr. 20, 2022. DOI: 10.3389/fpsyg.2022.811282

[40] R. ELBadrawy, S. ElKheshin, and N. ELEssawy, "Assess the effect of service quality on customer satisfaction in facebook social commerce in egypt," *International Journal of Managing Information Technology*, vol. 12, no. 3, pp. 9–24, Aug. 31, 2020. DOI: 10.5121/ijmit.2020.12302

[41] Y. Horiashchenko, "Marketing support of business, consumer and state interaction," *Economic journal of Lesya Ukrainka Volyn National University*, vol. 1, no. 29, pp. 67–75, Apr. 1, 2022. DOI: 10.29038/2786-4618-2022-01-67-75

[42] H.-M. Ma, "Analysis and strategy study on the conflict of omni-channel retailing in mobile internet era," *DEStech Transactions on Social Science, Education and Human Science*, no. icss, May 9, 2017. DOI: 10.12783/dtssehs/icss2016/8924

[43] R. J. Reddy and null Rojarani, "Online shopping-its growth and status in india," *International Journal of Engineering and Management Research*, vol. 8, no. 03, pp. 28–36, May 5, 2018. DOI: 10.31033/ijemr.8.3.3

[44] D. Fu, Y. Hong, K. Wang, and W. Fan, "Effects of membership tier on user content generation behaviors: Evidence from online reviews," *Electronic Commerce Research*, vol. 18, no. 3, pp. 457–483, Jul. 18, 2017. DOI: 10.1007/s10660-017-9266-7

[45] S. A. Olaleye, S. S. Oyelere, I. T. Sanusi, and J. F. Agbo, "Experience of ubiquitous computing technology driven mobile commerce in africa: Impact of usability, privacy, trust, and reputation concern," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 12, no. 3, pp. 4–20, Jul. 20, 2018. DOI: 10.3991/ijim.v12i3.7905

[46] S. H. Kukkuhalli, "Increasing digital sales revenue through 1:1 hyper-personalization with the use of machine learning for b2c enterprises," *Journal of Artificial Intelligence, Machine Learning and Data Science*, vol. 1, no. 1, 2023.

[47] T. Saira and S. Yessimzhanova, "Modern aspects and trends of customer intelligence development," *The economy: strategy and practice*, vol. 15, no. 2, pp. 107–113, Apr. 8, 2020. DOI: 10.51176/jesp/issue_2_t9

[48] C.-A. Tsai and C.-W. Chang, "Development of a partial shipping fees pricing model to influence consumers' purchase intention under the covid-19 pandemic," *Energies*, vol. 15, no. 5, pp. 1846–1846, Mar. 2, 2022. DOI: 10.3390/en15051846