The European Journal of Research and Development

journals.orclever.com/ejrnd

OR CLEVER Science & Research Group

Research Article

AI-Powered Multi-Agent Fashion Assistant for Personalized Retail Recommendations

Seza Dursun[1*], Sedat Çelik[2], Bahar Önel[3], Tülin Işıkkent[4], Mert Alacan[5]

[1] Boyner Orcid ID: https://orcid.org/0000-0003-1389-072X e-mail: seza.dursun@boyner.com.tr
[2] Boyner, Orcid ID: https://orcid.org/0009-0003-0335-6440 e-mail: sedat.celik@boyner.com.tr
[3] Boyner, Orcid ID: https://orcid.org/0009-0007-4597-6591 e-mail: bahar.onel@boyner.com.tr
[4] Boyner, Orcid ID: https://orcid.org/0009-0005-5775-0093 e-mail: tulin.isikkent@boyner.com.tr
[5] Boyner, Orcid ID : https://orcid.org/0000-0003-3893-6309 e-mail: mert.alacan@boyner.com.tr
*Corresponding author: e-mail: seza.dursun@boyner.com.tr; Tel.: (+90 533 614 59 23)

Received: 15 August 2025
Revised: 11 November 2025
2nd Revised: 26 November 2025
Accepted: 08 December 2025
Published: 24 December 2025

Reference: Dursun, S., Çelik, S., Önel, B., Işıkkent, T., & Alacan, M. (2025). AI-powered multi-agent fashion assistant for personalized retail recommendations. *The European Journal of Research and Development, 5*(1), 624–634.

Abstract

As fashion retail navigates a new era shaped by heightened consumer expectations and rapidly evolving digital interactions, the need for deeply personalized, stylistically coherent, and context-aware recommendation systems has become paramount. Traditional engines, reliant on static rules or collaborative filtering, often fall short in capturing the complexity of human taste and the visual-semantic richness inherent in fashion products. This paper introduces Boyner's AI-powered Multi-Agent Fashion Assistant, an enterprise-grade personalization platform architected on Microsoft Azure AI Foundry. The system orchestrates multiple specialized agents to deliver real-time, occasion-aware, and visually grounded fashion recommendations across omnichannel touchpoints. Leveraging multimodal embeddings, behavioral clustering, semantic search, and real-time trend signals, each agent operates with a distinct cognitive function, from silhouette-based outfit pairing to brand–season compatibility evaluation. Our implementation demonstrates how agentic AI systems can bridge the gap between algorithmic precision and stylistic intuition in large-scale fashion environments. The assistant not only enhances conversion and engagement

metrics but also redefines the digital shopping journey as an explainable, adaptive, and human-centric dialogue. By operationalizing multi-agent orchestration within a live retail environment, Boyner pioneers a new paradigm in AI-powered visual discovery, offering a scalable blueprint for next-generation personalization in the global fashion ecosystem.

Keywords: Visual Search, Multimodal AI, GroundingDINO, SigLIP, Milvus, Retail Intelligence, Semantic Search, AI in E-Commerce, Omnichannel Retail, Customer Experience

1. Introduction

The fashion retail industry is undergoing a profound transformation driven by the convergence of artificial intelligence, omnichannel commerce, and heightened consumer expectations. As digital experiences increasingly shape purchasing behavior, delivering scalable yet deeply personalized fashion recommendations has become a critical differentiator for modern retailers. Traditional recommender systems, primarily based on collaborative filtering or content-based heuristics, often fail to capture the stylistic nuance, contextual awareness, and evolving preferences essential for today's fashion consumers.

In this context, Boyner, one of the largest fashion retailers in Turkey, has developed the AI-powered Multi-Agent Fashion Assistant, an enterprise-grade intelligent system designed to redefine how fashion recommendations are generated and delivered at scale. Built on Microsoft Azure AI Foundry, the system leverages a modular, agent-based architecture where each agent is assigned a specialized cognitive task, ranging from real-time image-based search and silhouette-level outfit matching to behavioral segmentation and intent-aware personalization. The agents operate on top of real-time and historical customer interaction data, enabling contextual and stylistically coherent recommendations across both web and mobile channels.

Beyond its technical architecture, the system also reflects a broader organizational shift toward AI operationalization, integrating seamlessly with Boyner's search engine, product catalog, and customer analytics ecosystem. This initiative represents not only an innovative application of large language models (LLMs), multimodal embeddings, and semantic filtering, but also a step toward establishing a retail-specific agentic framework that supports explainability, experimentation, and continuous learning.

This paper presents the design principles, implementation pipeline, and key learnings from Boyner's Multi-Agent Fashion Assistant. We aim to contribute to the growing body of applied research on AI-native recommender systems while providing practical insights for retailers seeking to transition from monolithic, black-box models to modular, interpretable, and real-time recommendation architectures.

2. Materials and Methods

The development of the Multi-Agent Fashion Assistant system was grounded in a modular and scalable architecture designed to meet the needs of a high-velocity retail environment. This architecture combines cloud-native components with advanced AI capabilities for multimodal understanding, vector-based retrieval, and dynamic user interaction. Leveraging Microsoft Azure AI Foundry as its core infrastructure, the system orchestrates various autonomous agents that perform specialized tasks, such as outfit generation, search optimization, trend detection, and contextual ranking [4][11]. Each agent is built upon foundation models and task-specific pipelines that incorporate both textual and visual signals from Boyner's extensive product catalog and user behavior logs [3][6].

The overall pipeline follows a retrieval-then-generation strategy, enriched with vector similarity search and real-time feedback loops to enable adaptive personalization [2][14]. Integration with existing enterprise systems such as CRM, PIM (Product Information Management), and real-time search APIs ensures operational consistency and facilitates continuous learning across touchpoints [10][17].

The following subsections outline the main technological and methodological building blocks of the system.

2.1. Cloud-Native Architecture and Infrastructure

The Multi-Agent Fashion Assistant is built upon a fully cloud-native architecture deployed within the Microsoft Azure ecosystem, ensuring high scalability, resilience, and modularity required for real-time fashion recommendation scenarios. The system leverages Azure AI Foundry as its orchestration layer, enabling the deployment of autonomous agents through Azure Container Apps, each independently responsible for a specific cognitive task, such as generating outfits, detecting seasonal trends, or ranking products by user affinity [4][10].

At the data storage level, Azure Blob Storage is utilized for structured and unstructured data, including product catalogs, enriched metadata, and multimodal embeddings. These storage containers support versioning and access control mechanisms essential for enterprise-grade AI pipelines [6]. Audio-visual inputs (e.g., lifestyle photos, lookbook imagery) and structured product data (e.g., PIM exports, fabric details) are ingested and stored in separate containers for downstream embedding and retrieval.

The application layer is exposed via Azure API Management to serve real-time personalization endpoints to the Boyner.com.tr frontend, the mobile app, and CRM dashboards. This allows seamless integration with the existing search engine, customer segmentation platform, and recommendation widgets across Boyner's digital touchpoints [10].

To handle LLM-based reasoning, the platform integrates Azure OpenAI Service, which securely calls task-specific GPT-4 models for creative outfit generation, persona-based captioning, or occasion-aware filters. The system uses fine-grained role-based access control (RBAC) and prompt-level safety layers to ensure compliant and context-aware generations, especially when deployed in production retail environments [1][13].

The infrastructure supports continuous deployment and monitoring through GitHub Actions and Azure Monitor, enabling A/B testing of agents, rollout of improved ranking models, and error tracking through telemetry dashboards. This DevOps integration allows data scientists and product teams to iterate on new features and agents with minimal friction and ensures observability across the full lifecycle of an AI fashion interaction [11].

This modular approach allows the Fashion Assistant to grow organically: new agents can be deployed as microservices without interrupting existing operations, and updates to embeddings or prompts can be tested in isolated sandboxes before full deployment, reflecting a mature MLOps mindset within fashion retail innovation [3][4][12].

2.2. Vector Search and Multimodal Embedding Layer

At the core of the Fashion Assistant lies a vector-based semantic matching system designed to transcend the limitations of traditional keyword search and rule-based filtering in fashion retail. This layer enables personalized, stylistically consistent, and context-aware recommendations by utilizing embeddings generated from multimodal data, text, images, and structured metadata, mapped into a shared high-dimensional vector space [2][6].

To represent textual data, including product titles, descriptions, and GPT-generated outfit captions, the system employs OpenAI's ADA-002 model, which provides efficient and accurate embeddings aligned with GPT-4-based semantic understanding [13]. These embeddings capture fine-grained semantic relations between concepts such as "flowy summer dress" and "boho chic outfit," essential for styling tasks beyond exact term matches.

For visual data, the assistant integrates CLIP-based image embeddings, applied to product photos, model shots, and lookbook images. These embeddings enable the system to understand visual cues like color palette, silhouette, and texture, a critical advantage for fashion domains where visual language often conveys more meaning than product names [3][8]. All image embeddings are preprocessed via a vision encoder and normalized before indexing.

Structured product attributes from Boyner's Product Information Management (PIM) system, such as category, season, brand, and material, are tokenized and embedded separately to preserve their influence on outfit cohesion and customer preferences. These

embeddings are fused with textual and visual vectors using weighted late fusion methods to generate a comprehensive representation of each item [10].

All vectors are indexed using Milvus, a high-performance vector database optimized for approximate nearest neighbor (ANN) search, enabling millisecond-level retrieval over millions of embeddings [9]. The vector index supports multiple partitions, allowing segmented queries by gender, occasion, brand affinity, or seasonality. The ANN engine is tuned to optimize recall-to-latency tradeoffs appropriate for real-time user interactions on e-commerce platforms.

The retrieval logic is orchestrated by an agent that dynamically selects the most relevant retrieval mode: visual similarity, style coherence, or hybrid personalization. For instance, a "smart casual" look for a winter occasion would prioritize color and texture alignment while also considering the user's past style ratings and preferred silhouettes [7][12].

This multimodal vector infrastructure ensures that recommendations are not only accurate but also emotionally resonant, visually appealing, and explainable to the end user, bridging the gap between search intent and fashion creativity in a scalable, production-ready system [4][5][13].

2.3. Multi-Agent Orchestration Layer

To manage the complexity of dynamic fashion personalization, the Fashion Assistant architecture incorporates a modular multi-agent orchestration layer deployed on Microsoft Azure AI Foundry [13]. This layer enables the sequential and conditional execution of specialized agents, each responsible for a key function in the recommendation pipeline. The following subsections outline the three major components of this orchestration process.

2.3.1. Search and Style Assembly

The orchestration begins with the Search Agent, which executes a real-time semantic retrieval task using Milvus vector database. It receives user queries transformed into embeddings via OpenAI's text-embedding-3-small model [9], and returns a filtered product shortlist based on cosine similarity to relevant product vectors stored in the database [6].

These retrieved candidates are then forwarded to the Style Assembly Agent, which leverages GPT-4 Turbo to construct visually and thematically coherent outfit combinations. The agent factors in contextual signals, such as occasion type, season, and gender, and performs prompt chaining with curated style templates. This enables the assistant to build full looks rather than isolated product suggestions, which is critical in fashion use cases where harmony and coherence drive conversion [4][5].

2.3.2. Persona Filtering and Trend Adjustment

The output of the style assembly phase is refined through the Persona Filter Agent, which applies behavioral segmentation to ensure alignment with customer archetypes. These segments, such as "casual minimalist," "bold trendsetter," or "professional classic" are generated through unsupervised clustering over user embeddings derived from CRM and transaction data [11].

Following this, the Trend Adjustment Agent dynamically adjusts rankings based on commercial and trend-based factors. It incorporates live data feeds such as SKU-level sales velocity, marketing promotion flags, and inventory availability to ensure the final output is not only stylistically aligned but also strategically relevant to current business priorities [2][12]. This agent also incorporates feedback loops, allowing marketing teams to intervene via campaign-driven re-ranking models.

2.3.3. Explanation Generation and Orchestration Flow

The final stage involves the Explanation Generator Agent, which enhances interpretability through natural language justifications of recommendations. It utilizes a RAG (Retrieval-Augmented Generation) setup, drawing from a curated knowledge base of fashion principles, trend articles, and seasonal style guides [1][4]. Generated explanations help bridge the gap between algorithmic suggestions and human-like stylistic reasoning (e.g., "Monochrome layering for an effortless urban chic appeal").

All these agents are orchestrated using Azure Durable Functions, where an orchestrator process determines execution logic based on system rules and user intent [13]. Each function's output is passed to the next in the flow, logged via Azure Application Insights, and version-controlled for performance monitoring and rapid iteration.

This architecture enables flexibility, maintainability, and rapid experimentation, allowing new agents (e.g., sustainability scoring, region-specific biasing) to be introduced with minimal disruption [3][7].

2.4. User Persona Modeling and Behavioral Signals

2.4.1. Multimodal Behavioral Embedding

At the heart of the persona modeling pipeline lies a multimodal embedding process, where structured (e.g., CRM demographics, transaction history) and unstructured (e.g., search queries, product reviews, clickstream data) signals are encoded into dense vector representations. This is achieved via a hybrid embedding strategy: textual data is processed using OpenAI's text-embedding-3-small model [9], while structured tabular signals are embedded through neural multilayer perceptrons trained on contrastive learning objectives [3].

These embeddings are normalized and concatenated to create a holistic user vector that captures both preference signals (e.g., frequent category purchases, price sensitivity) and stylistic leanings (e.g., affinity for color palettes, seasonal trends) [6].

2.4.2. Clustering and Dynamic Segment Assignment

The resulting user vectors are periodically clustered using unsupervised learning techniques, primarily K-Means and Gaussian Mixture Models (GMM), to identify latent user segments. Each cluster is mapped to a dynamic persona label (e.g., "smart casual explorer," "elegant classicist") which is used to drive filtering and personalization throughout the recommendation pipeline [11].

Unlike static personas, these labels are updated in real-time using a stream processing engine on Azure Synapse, enabling the system to reflect evolving user behavior, such as a shift from summer to fall shopping preferences or increased interest in formalwear due to an upcoming event [13].

2.4.3. Contextual Augmentation and Feedback Loops

In order to ensure that personas remain relevant within a specific recommendation session, the system integrates contextual augmentation strategies. For instance, if a user searches for "beachwear," the system temporarily boosts signals associated with resort and summer collections, regardless of the user's primary persona cluster [8].

In addition, explicit and implicit feedback loops are built into the system. Clicks, dwell time, wishlist additions, and cart activity are logged and used to adjust vector weights, re-train embeddings, and refine clustering boundaries on a weekly basis. This architecture supports rapid adaptation to micro-trends and short-term intent changes while preserving long-term stylistic fidelity [4][10].

Once the product region is identified, the image is cropped to eliminate background noise and standardize the input for the visual similarity search stage. This not only improves retrieval quality but also aligns the format of user images with catalog images, which are typically taken in controlled studio environments. The use of GroundingDINO therefore ensures consistency and precision in object localization, forming a crucial bridge between free-form user inputs and structured catalog data [11].

2.5. Evaluation and Real-World Feedback Loops

The performance and generalizability of the Fashion Assistant system are fundamentally shaped by the quality of its training data, the rigor of its evaluation metrics, and the design of its feedback loop. This section outlines the pipeline's underlying data sources, preprocessing strategies, and the evaluation framework used to monitor model effectiveness over time.

The system is trained on a diverse and multi-source dataset comprising historical behavioral logs (e.g., clickstream patterns, wishlist events, cart additions, and purchase conversions), structured product catalog metadata (including categorical taxonomy, brand features, price segments, and visual embeddings derived from multimodal encoders), and direct user feedback signals such as explicit ratings, post-interaction surveys, and Net Promoter Score (NPS) responses [7][12].

To uphold ethical and regulatory compliance, all user-related data undergo strict anonymization in accordance with both GDPR and KVKK protocols. Textual attributes are processed through a normalization pipeline involving lemmatization, tokenization, and lowercasing, while numerical outliers are addressed via interquartile range-based filtering techniques to prevent skewed learning [2].

Evaluation of the agent ensemble adopts a multi-objective lens. Precision@K is employed as the primary relevance metric, measuring the share of relevant products appearing in the top-K recommendations based on recent user interactions [6]. Category coverage is used to quantify the breadth of recommendations across subcategories, ensuring that personalization does not overfit to narrow product bands. In addition, a stylistic diversity score is computed using cosine similarity across item embeddings to evaluate the semantic variety of the recommended sets, encouraging exploration without sacrificing relevance.

Feedback signals, both explicit and implicit, are continuously integrated into the retraining process via scheduled batch learning cycles. These signals are weighted based on source reliability and recency, and their impact is measured through A/B testing frameworks conducted in live environments. Such iterative fine-tuning ensures that the system evolves with seasonal trends, shifting consumer preferences, and catalog updates, preserving alignment with real-world retail dynamics [4][13].

3. Results

The deployment of the Multi-Agent Fashion Assistant in Boyner's e-commerce ecosystem led to tangible operational improvements across the customer journey, spanning discovery, engagement, and conversion stages. These advancements reinforce the applicability of agentic AI systems in dynamic and style-sensitive environments such as fashion retail [1][2].

Behavioral signals collected post-deployment indicated a more fluid and engaging browsing experience. Users interacted more frequently with personalized outfit suggestions, particularly those driven by occasion-based and trend-aware logic. The assistant's ability to offer contextually relevant recommendations enhanced the perceived coherence of suggested ensembles [4][6][11].

Customer interaction patterns further suggested a stronger alignment between search intent and recommendation outcomes, especially in curated categories like "business

casual" and "seasonal trends." These improvements stemmed from the assistant's multimodal semantic matching engine, which combines image, text, and behavioral embeddings to capture latent style preferences [5][14].

From an operational standpoint, the integration of generative agents into the merchandising workflow significantly reduced manual workload for styling and marketing teams. Automated tasks such as outfit bundling, occasion tagging, and style classification were aligned with internal product information management (PIM) structures, enabling faster adaptation to inventory and promotional cycles [3][10].

The agentic architecture demonstrated seamless interoperability with Boyner's core digital infrastructure, including CRM, PIM, real-time search, and personalization surfaces. Dynamic scheduling via the orchestration layer allowed for low-friction experimentation, supporting an agile innovation cycle [7][9].

In addition to these qualitative advancements, quantitative performance indicators highlighted the system's robustness and production-readiness. The index hit rate, representing the percentage of recommendations successfully retrieved from the real-time vector index, reached 98%, as monitored via Milvus-based embedding retrievals [13]. The slot accuracy, denoting the correct mapping between user intent and agent routing, was observed at 95%, enabled by orchestrated agent pipelines in Azure Durable Functions [4][10]. Moreover, the assistant preserved gender consistency in 92% of recommendations, ensuring alignment with user demographics and fashion expectations, a result validated through internal A/B testing scenarios [6][11]. Despite the system's multi-agent complexity, the average latency per recommendation call remained within a responsive 4-second window, thanks to lightweight vector search and optimized prompt chaining mechanisms [5][9]. Finally, internal post-deployment evaluations yielded a user satisfaction score of 4.7 out of 5, based on stakeholder interviews, heuristic evaluations, and internal feedback loops during pilot phases [1][17].

4. Discussion and Conclusion

The development and deployment of the Multi-Agent Fashion Assistant represent a significant step forward in operationalizing explainable and context-aware artificial intelligence in fashion retail. By combining real-time behavioral signals, multimodal embeddings, and agentic orchestration, the system addresses key limitations of traditional recommendation engines, namely their lack of stylistic nuance, contextual awareness, and interpretability [2][5].

One of the core innovations of this system lies in its agent-oriented modularity, which allows each component, whether for trend mapping, occasion-based recommendation, or visual search, to evolve independently without disrupting the overall experience. This architecture not only supports scalability and experimentation but also offers a

foundation for lifelong learning and adaptive recommendation pipelines in high-velocity retail environments [1][8][14].

The results observed during live deployment at Boyner substantiate the assistant's impact both on customer-facing metrics and internal workflows. The measurable improvements in session length, click-through rate, cart composition, and stylist productivity point to the assistant's efficacy in enhancing digital engagement and operational efficiency simultaneously [4][6][10].

Moreover, the system demonstrated strong alignment with strategic enterprise goals such as faster inventory turn, more effective bundling of items, and customer satisfaction through personalized discovery. These outcomes reinforce the notion that fashion personalization must evolve beyond generic similarity-based matching into narrative-driven, occasion-aware, and style-coherent interactions, a capability made possible by agentic AI systems [11][13].

Looking ahead, future iterations of the Fashion Assistant will integrate reinforcement learning mechanisms, social signal modeling, and customer sentiment feedback to refine agent behavior in a fully closed loop. The system's architecture is also being prepared for multi-market deployment, enabling cultural adaptation and regional style tuning across Boyner's expanding international footprint [3][9][17].

In conclusion, the Multi-Agent Fashion Assistant exemplifies how vision-aligned AI transformation can materialize through modular intelligence, operational agility, and user-centered design. Its success offers a replicable blueprint for retailers aiming to embrace AI not as a feature layer, but as a foundational layer of strategic capability [7][15].

5. Acknowledge

The authors would like to express their sincere gratitude to the Boyner AI Team for their dedicated work in designing, implementing, and iterating the Multi-Agent Fashion Assistant system. Special thanks go to the Data Science, Data Engineering, and IT Architecture teams for enabling seamless integration with enterprise systems and ensuring operational scalability.

We also acknowledge the valuable contributions of Microsoft Azure AI Foundry in providing the technical infrastructure that empowered our modular, cloud-native deployment. Their collaboration was instrumental in accelerating experimentation, agent development, and deployment across production environments.

Finally, we thank the Boyner.com.tr Product and Marketing Teams for their close collaboration in aligning AI outputs with stylistic and commercial objectives, and for their feedback throughout the real-world testing cycles.

References

[1] T. Schick et al. (2023). "Toolformer: Language Models Can Teach Themselves to Use Tools," arXiv preprint, arXiv:2302.04761.

[2] Y. Jiang et al. (2023). "Agents: An Open Platform for Language Models as Autonomous Agents," arXiv preprint, arXiv:2308.08155.

[3] OpenAI. (2024). "GPT-4 Technical Report".

[4] Microsoft. (2024). "Azure AI Agents and Foundry Documentation," [Online]. Available: https://learn.microsoft.com/en-us/azure/ai-services/

[5] X. Han et al. (2017). "Automatic spatially-aware fashion concept discovery," in Proc. ICCV.

[6] M. Vasileva et al. (2018). "Learning Type-Aware Embeddings for Fashion Compatibility," in Proc. ECCV.

[7] T. Lin et al. (2021). "FashionBERT: Text and Image Matching with Adaptive Loss for Cross-modal Retrieval," arXiv preprint, arXiv:2004.03688.

[8] Amazon Science, "Outfit Recommendation with GCN and Contrastive Learning," 2021.

[9] Gartner. (2023). "Operationalizing AI in Retail: From Pilots to Scale,".

[10] McKinsey & Company. (2022). "The State of AI in Retail,".

[11] IDC. (2024). "The Future of AI Agents in the Enterprise,".

[12] RTIH. (2023). "AI in Fashion Retail Innovation Report,".

[13] A. Neelakantan et al. (2022). "Text and Code Embeddings with OpenAI's ADA-002 Model," OpenAI.

[14] Zilliz. (2023). "Milvus: Open-source Vector Database for Scalable AI,"[Online]. Available: https://milvus.io/.

[15] J. Devlin et al. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT.

[16] TÜSİAD. (2023). "Türkiye'de Perakende ve Dijitalleşme Raporu,".

[17] Boyner AI Initiative. (2024). "Internal Documentation & Use Case Landscape," Boyner Holding.