JOURNAL OF INTERNATIONAL CRISIS AND RISK COMMUNICATION RESEARCH
ISSN: 2576-0017
2026, VOL 9, NO 2

Anatomy Of Real-Time Ad Delivery Systems: Behind The Scenes Of Modern Advertising Infrastructure

Rishi Kanth Alapati

University of Southern California, Los Angeles, USA

Abstract

The key infrastructure of ad delivery systems in the digital advertising age is real-time systems, which integrate advanced engineering and machine learning to drive monetization on both web and mobile platforms. These multifaceted systems must operate within strict latency limits, with responsiveness measured in milliseconds, and evaluate hundreds of possible advertisements based on a set of targeting parameters. This article reviews the complex design of such high-performance systems, including routing of requests with load balancers, and specialized subsystems that deal with eligibility, ranking, budgetary, and creative selection. It examines how multi-layered caching and parallel processing can be used to support sub-millisecond performance at large scale, and the analytical pipelines that promote continual improvement. The article also discusses the major engineering issues that these systems encounter, such as global scale pressures, distributed consistency problems, optimisation of cold-start issues, and changing privacy policies. The knowledge of these system design patterns and limitations can provide useful learning to the engineers in other fields of high performance, other than the advertising technology.

Keywords: Real-Time Advertising Infrastructure, Latency Optimization, Distributed Caching Strategies, Machine Learning Ranking Systems, Privacy-Compliant Ad Technology.

1. Introduction

The real-time advertisement systems are the keystone to modern digital advertising stock as they represent the high-tech technical framework fueling monetization plans online and on mobile applications. Such systems make detailed decision-making in a matter of milliseconds, managing numerous constraints and optimization targets simultaneously.

The design of such systems holds a level of amazing complexity, as Sharon Horsky explains in the analysis of Google Ads infrastructure, where he goes on to detail how the platform has to process high numbers of candidate advertisements per request while keeping response times below hard latency limits to sustain the quality of the user experience [1]. This is a computational problem that scales exponentially with size, as indicated in the industry-scale infrastructure initiatives reported by Sharon Horsky, where they cite the evolution of advertising platforms from being mere placement systems to complex real-time decisioning engines that handle millions of requests at once during periods of high traffic volume [2].

Modern ad delivery systems are built as multi-tiered architectures ranging from initial request routing to geographically dispersed load balancers to optimized data centers for low latency. Yodgorbek Komilov explains how Google's architecture uses multiple redundant paths for routing for the sake of high availability, with traffic management systems that change dynamically based on network performance and server capacity [1]. The backend core orchestrates sophisticated workflows across many specialized

subsystems, each operating on a different piece of the ad selection process, such as eligibility determination, machine learning-based ranking, budget allocation, and creative selection.

In order to ensure performance under stringent latency requirements, such platforms have complex caching mechanisms in place at multiple layers. Yodgorbek Komilov describes how Google's data infrastructure leverages tiered caching methods that favor increasingly accessed content, with multiple storage systems optimized for certain access patterns and consistency demands [1]. Such caching implementations are augmented by the organizational frameworks outlined by Sharon Horsky, who details how advertisement technology firms developed specialized departments dedicated solely to performance optimization and data center reliability to underpin these mission-driven systems [2].

This article explores the architecture, elements, and engineering issues of these high-speed systems that are the backbone of the global digital advertising industry, learning from both technical deployments and organizational designs that have developed to enable this infrastructure of great importance.

2. The Critical Time Constraint

The defining feature of contemporary ad delivery systems is their capacity to serve ad requests inside very tight latency budgets—often less than 100 milliseconds. All this in a short span of time, when the system needs to serve the request that comes in, filter out hundreds of available ads, apply targeting and eligibility rules, compute auction ranks, pick optimal creative form, enforce frequency cap rules, and serve the final ad creative. This is because ad selection is usually done as part of a comprehensive page or app load process, where latency directly affects user experience.

From the findings of Growth-onomics, these latency requirements have become more important as consumers are showing decreasing tolerance with digital experiences that are slow or unresponsive [3]. From their breakdown of performance metrics for ad platforms, user engagement notably degrades when delays in loading ads are beyond the 100-ms barrier, with significantly high effects on conversion and session time. This immediate correlation between system performance and business results has caused advertising technology providers to make response time optimization a primary engineering goal.

The amount of computational work required to meet these tight time frames is considerable. As described by MediaCulture in their exhaustive study of programmatic ad infrastructure, contemporary platforms have to consider and screen massive candidate pools while applying advanced targeting criteria and business rules in under a millisecond [4]. Their work details how sophisticated programmatic systems have developed specialized processing pipelines that filter and rank candidates successively through progressively refined stages, enabling effective allocation of computational resources while strictly enforcing performance limits.

The multi-phase assessment demanded within this limited time window has essentially influenced system design decisions. MediaCulture captures the manner in which the industry moved from centralized processing paradigms to dispersed architectures that take advantage of microservices, edge computing, and specialty hardware acceleration to deliver the required performance attributes [4]. Such an architectural shift allows both horizontal scaling to support traffic variation and bespoke optimization of specific processing elements, establishing dramatic competitive advantages for organizations possessing advanced engineering competencies.

Perhaps most strikingly, these performance demands need to be sustained at an enormous scale. Growth-onomics observes that enterprise-class ad platforms typically process millions of requests per second at peak times, with the largest systems processing hundreds of billions of ad requests per day across worldwide infrastructure [3]. Sustaining steady sub-100ms performance at this size demands advanced load distribution mechanisms, multi-layered caching practices, and graceful degradation features that sustain critical functionality even under conditions of maximum load.

Table 1: Critical Time Constraints in Ad Delivery Systems: Performance Metrics and Business Impact [3, 4]

Performance Factor	Value/Impact
Latency Budget	<100 milliseconds
User Engagement Degradation Threshold	100 milliseconds
Request Volume (Peak)	Millions per second
Daily Request Volume (Large Systems)	Hundreds of billions
Processing Steps	7 (request processing, ad filtering, targeting application, auction ranking, creative selection, frequency capping, final delivery)
Impact of Exceeding Latency Budget	Significant decrease in conversion rates and session duration

3. System Architecture Overview

3.1 Request Routing Infrastructure

An ad request's journey starts with routing on an advanced load-balancing infrastructure. The routing layer forwards incoming traffic to geo-distributed data centers, balancing for proximity to the user and operational capacity. Current systems use Layer 7 load balancers to handle HTTP/HTTPS traffic, BGP anycast routing for geographic distribution efficiency, active health checks to identify and route around failures, and a traffic splitting feature to roll out features gradually.

As described by Moparthi in his technical analysis of real-time bidding infrastructure, the routing layer is the key entry point that affects both performance and reliability aspects of the overall system [5]. His review of industry deployments shows how advanced ad platforms use edge presence in many different worldwide regions to reduce network latency, with smart traffic distribution systems that dynamically reroute based on a variety of factors such as server capacity, network congestion, and even forecasted traffic behavior. These routing systems dynamically track health statistics throughout the infrastructure, rerouting traffic automatically when performance degradation is detected in order to provide service availability during even temporary outages.

3.2 Core Delivery Backend

Inside the data center, the request reaches the delivery backend service—the orchestration layer that handles the ad selection pipeline. This piece of software handles incoming requests by parsing and validating them, triggering concurrent sub-requests to specialized services, handling timeout budgets for downstream components, enforcing business logic and policy, and constructing the final response.

As per GeoEdge's in-depth industry research, the delivery backend is the central nervous system of contemporary ad platforms that orchestrates the intricate interactions between many specialized subsystems, which together decide on ad selection [6]. In their study, they capture how top-tier platforms have transitioned from monolithic to advanced service-oriented design, allowing for both development agility and operational stability. These systems utilize thoughtful timeout handling with explicit latency budgets assigned to each downstream dependency such that overall response times are kept within acceptable ranges even when individual components suffer from performance degradation.

3.3 Specialized Subsystems

The delivery backend aligns with various specialized subsystems:

- **Eligibility Engines:** Determine what ads can be displayed based on user attributes and segments, geo-targeting, device and platform requirements, and content context and safety considerations. Moparthi's technical explanation brings to the fore eligibility determination as the most computationally demanding part of the ad selection process [5]. His work explains how contemporary platforms utilize multi-stage filtering pipelines that iteratively reduce the candidate set through increasingly sophisticated criteria. Such systems make use of customized data structures

that are optimized for fast evaluation of intricate targeting expressions, allowing large-scale processing of large numbers of targeting conditions over enormous inventories at stringent performance profiles.

- **Ranking Models:** Systems that are based on machine learning to predict click-through rates (CTR), conversion probability estimates, expected value calculations for advertisers, and platform revenue maximization. GeoEdge's industry analysis names the development of ranking abilities as a hallmark trend in ad tech innovation [6]. They document the progression from basic rule-based ranking to advanced implementations of machine learning that rely on vast historical and contextual information to make educated guesses about engagement probability and value. These systems rely more and more on ensemble methods that take different specialized models with disparate areas of focus for performance prediction and combine them to provide more detailed and accurate judgments of anticipated advertisement value.
- **Budget Controllers:** Impose financial limits such as daily and lifetime campaign spends, pacing algorithms for uniform distribution, bid optimizations based on auction dynamics, and spend tracking and throttling. Budgeting is an essential part of advertising infrastructure, as outlined in Moparthi's survey of real-time bidding systems [5]. His work explains how contemporary platforms use distributed budget monitoring with high consistency guarantees to keep precise expenditure records in geographically spread infrastructure. Such systems utilize advanced pacing algorithms that modulate delivery rates on the basis of past performance trends, anticipated traffic levels, and competitive situations, optimizing budget usage across campaign lifecycles.
- **Creative Selection:** Ad format-to-inventory matching systems, creative adaptation to varying screen sizes, personalization where appropriate, and brand safety compliance. In GeoEdge's holistic review of ad tech evolution, the abilities of creative selection have become more advanced as digital formats have mushroomed [6]. Their report verifies how contemporary platforms have evolved sophisticated systems for creative optimization that take into consideration not just technical compatibility but also performance traits, personalization possibilities, and brand safety needs. These systems more and more include dynamic creative optimization methods that can automatically change visual components, messages, and calls to action depending on user context and historical performance.

Table 2: Architectural Components of Modern Ad Delivery Systems: Functions and Interactions [5, 6]

System Component	Key Elements	Primary Functions
Request Routing Infrastructure	● Layer 7 load balancer ● BGP anycast routing ● Active health checking ● Traffic splitting	● Geographic distribution ● Latency optimization ● Failure detection ● Gradual feature rollout
Core Delivery Backend	● Request parsing/validation ● Parallel sub-requests ● Timeout management ● Business logic enforcement	● Orchestration ● Workflow management ● Response assembly ● Policy application
Specialized Subsystems	● Eligibility Engines ● Ranking Models ● Budget Controllers ● Creative Selection	● Targeting application ● Performance prediction ● Financial constraint enforcement ● Format optimization

4. Latency Management Strategies

4.1 Multi-Layered Caching

To get sub-100ms performance, ad delivery systems utilize advanced caching techniques: Hot In-Memory Caches (local process caches for data of highest frequency), Distributed Cache Clusters (frameworks like Memcached or Redis for shared state), Specialized Data Stores (custom solutions like Facebook's TAO for effective graph data access), and Probabilistic Prediction Caches (pre-computed model outputs for frequent cases).

As explained in the seminal paper on Facebook's TAO data store by Bronson and others, caching is a key system architectural element for high-performance advertising systems at web scale [7]. Their work illustrates how custom caching infrastructures allow platforms to provide consistent low-latency responses even with extremely high volumes of data and intricate access patterns. The multi-level design that is outlined in their paper emphasizes the progression from straightforward key-value caches to complex distributed systems that are tailored to the distinct needs of social graph data access patterns. These systems walk a balance among a variety of conflicting concerns—access performance, consistency guarantees, and failure resilience—using thoughtful hierarchical designs that make use of alternative storage mechanisms for various kinds of data.

The intricacies of these cache strategies continue to refine beyond generic-purpose solutions to highly specialized deployments. Contemporary advertising platforms have built upon these core ideas to create increasingly bespoke caching infrastructures optimized for particular patterns of data access prevalent in advertising workloads. These bespoke strategies frequently involve probabilistic methods for prediction caching, with typical model input combinations being pre-calculated and cached for immediate reuse. By carefully placing various caching layers along the request processing pipeline, such systems can significantly minimize mean latency while keeping the capability to manage intricate, dynamic data needs intrinsic in personalized ad delivery.

4.2 Parallel Processing

Such systems optimize throughput by parallelizing: Fan-out requests to multiple services in parallel, using non-blocking asynchronous I/O models, using partial response aggregation wherever feasible, and using adaptive timeout management for critical path operations.

Parallel processing is a basic architectural style in contemporary ad delivery systems, which has been reported by CelerData's in-depth review of parallelization methods within high-performance computing areas [8]. They track the evolution of parallel processing paradigms from specialized scientific computing to become central architectural styles in commercial platforms, including ad technology platforms. Current ad delivery systems take advantage of these methods via complex request distribution schemes that allow for parallel processing across dimensions—considering alternate ad candidates in parallel, handling multiple targeting criteria in parallel, and distributing workloads across geo-regions to maximize resource utilization.

The utilization of efficient parallel processing in advertising systems goes beyond straightforward task partitioning to involve advanced workflow management, as per CelerData's market analysis [8]. Their explanation of how contemporary platforms utilize event-driven architectures that use non-blocking I/O models with high throughput even when dealing with many downstream dependencies, adaptive timeout control with dynamic budget assignment based on observed performance and request criticality, is apt. And most importantly, perhaps, they include graceful degradation mechanisms with partial response aggregation so that the system can provide the best available solution under the time limit instead of totally failing when some of its components are delayed.

Table 3: Performance Optimization Techniques in Real-Time Ad Delivery Systems [7, 8]

Strategy	Implementation	Benefits	Key Technologies
Multi-Layered Caching	● Hot In-Memory Caches	● Reduced response time	● Process-local memory ● Memcached/Redis

	• Distributed Cache Clusters • Specialized Data Stores • Probabilistic Prediction Caches	• Consistent performance • Support for complex data patterns	• Custom solutions (e.g., TAO) • Pre-computed predictions
Parallel Processing	• Request fan-out • Asynchronous I/O • Partial response aggregation • Adaptive timeout management	• Increased throughput • Maximized resource utilization • Graceful degradation	• Event-driven architectures • Non-blocking I/O models • Dynamic budget allocation • Distributed workloads

5. Analytics and Feedback Loops

5.1 Metrics Pipeline

Ad delivery systems have full instrumentation: Impression logging with delivery metadata, click and conversion tracking, latency and error rate monitoring, and budget consumption and pacing metrics.

As explained in Straive's in-depth examination of marketing analytics infrastructure, metric collection constitutes the base layer on which all optimization and enhancement processes within advertising platforms are established [9]. Their study expounds on how advanced delivery systems have multi-phased event processing pipelines that can process enormous volumes of events with preserved data integrity and minimal latency effects on core service paths. These pipelines commonly use distributed streaming architectures that buffer events in memory prior to asynchronously processing them through multiple enrichment and aggregation phases, avoiding instrumentation overhead from impacting the performance-critical ad delivery path. The output data streams record fine-grained information throughout the entire delivery workflow, starting from initial request parameters to final rendering and interaction results.

Their complexity just keeps on growing as ad platforms are developed to accommodate more advanced campaign goals and attribution mechanisms. Straive's analysis states that today's platforms now measure many different distinct metrics per impression, ranging from technical performance measures, business performance metrics, to user activity signals [9]. These systems need to carefully balance mass data gathering with privacy and infrastructure limitations, using complex sampling methods and multi-tiered storage mechanisms to cope with the unprecedented amounts of telemetry generated by production systems running at a global scale, yet ensuring compliance with changing regulatory needs.

5.2 Real-Time Dashboards

Operational visibility is sustained through: Service health monitoring, campaign performance monitoring, anomaly detection systems, and revenue impact visualization.

The process of converting raw metrics into actionable insights is reliant on advanced monitoring infrastructure, as evidenced by Gordon and co-authors in their observational survey of Facebook advertising practices [10]. Their findings detail how contemporary platforms utilize layered monitoring systems that address various stakeholder needs, ranging from engineering teams with specific system health metrics to business stakeholders concerned with campaign performance and return on investment metrics. These monitoring systems increasingly incorporate real-time processing functions that offer near-instant insight into platform health as well as campaign performance, allowing for immediate response to technical problems and timely optimization of ad strategy.

Anomaly detection is a very important function within these monitoring systems, Gordon's analysis of ad platform operations indicates [10]. Their work explains how advanced platforms have developed from mere threshold-based notification to deploying statistical models that automatically detect unusual patterns in

many dimensions of performance. Such systems tend to generate dynamic baselines including information regarding known traffic patterns, seasonality, and projected growth trends so that they can more accurately identify really problematic deviations and reduce false positives that may overwhelm operations staff or mislead advertisers about campaign performance.

5.3 Model Retraining Loops

The performance of the system is enhanced by: Ongoing model training with new impressions and click data, A/B testing infrastructure for algorithmic updates, feature store refreshes for targeting optimization, and performance reviews across various segments of inventory.

Ongoing model updating is an essential loopback in today's advertising systems, as explained in Straive's examination of analytics infrastructure [9]. Their study describes how advanced platforms use automated training pipelines that continuously feed in new interaction data to update prediction accuracy and respond to changing user behaviors and market conditions. These systems usually keep centralized feature stores that facilitate consistent access to high-quality training signals by multiple model development workstreams. The models resulting from these experiments are subjected to aggressive testing against holdout sets before deployment into production environments, with extensive monitoring for any drop in performance after release.

Experimentation systems are key to this ongoing improvement process, as Gordon and co-authors' wide-ranging survey of advertising practices found [10]. Their study records how Facebook's platform supports advanced experimentation capabilities that facilitate controlled evaluation of algorithm modifications over statistically significant traffic segments. These frameworks facilitate concurrent testing of many variations with rigorous isolation to avoid interaction effects, allowing for rapid iteration on model refinement without strict control over possible adverse effects. The resulting experiment data passes into extensive analysis systems that measure performance by broad categories of advertisers, audiences, and placements, so that optimization improves overall ecosystem health instead of preferring to optimize for limited traffic segments at the cost of global platform health.

Table 4: Continuous Improvement Cycle in Ad Delivery Systems: From Data Collection to Model Refinement [9, 10]

Component	Key Elements	Purpose	Implementation
Metrics Pipeline	● Impression logging ● Click/conversion tracking ● Latency/error monitoring ● Budget consumption metrics	● Data collection ● Performance measurement ● System optimization	● Distributed streaming architectures ● Asynchronous processing Multi-phase event processing
Real-Time Dashboards	● Service health monitoring ● Campaign performance tracking ● Anomaly detection ● Revenue visualization	● Operational visibility ● Issue identification ● Performance optimization	● Layered monitoring systems ● Statistical models ● Dynamic baselines ● Near-immediate updates

Model Retraining	• Continuous model training • A/B testing frameworks • Feature store updates • Performance analysis	• Prediction refinement • Algorithm improvement • Targeting enhancement	• Automated training pipelines • Controlled experimentation • Holdout testing • Cross-segment evaluation

6. Engineering Challenges

Scaling and keeping these systems updated pose some challenges: Scale (serving billions of requests daily on global infrastructure), Consistency (accurate budget tracking across distributed systems), Cold Start (processing new campaigns with sparse historical data), and Privacy Regulations (keeping up with changing law requirements such as GDPR and CCPA).

As Abhishek Shetty notes in his in-depth review of self-serve ad platform systems, the size of contemporary ad delivery systems is one of the most daunting engineering challenges confronting technology organizations today [11]. His study records how these platforms need to handle unprecedented request volumes on globally distributed infrastructure while meeting very high performance and reliability requirements. As advertising ecosystems grow, these infrastructures are increasingly under pressure to grow horizontally across many geographic areas, into more and more sophisticated distributed architectures that balance responsiveness in the local area against global consistency requirements. The resultant infrastructure commonly uses advanced partitioning techniques that spread processing load while keeping intelligent routing facilities to route requests optimally based on capacity, proximity, and service health.

The consistency problem is especially severe for budget management elements, according to Abhishek Shetty's work [11]. They explain how ad platforms need to keep up-to-date spending records among distributed infrastructure elements while handling high volumes of transactions in real time. These systems use custom consistency protocols that are designed to keep spending decisions up to date even when elements crash or network conditions worsen. The ensuing budget tracking mechanisms are a fine balance between enforcing strict consistency constraints for financial precision and the performance needs of real-time decision making, typically utilizing domain-specific data structures tailored to this application alone.

The cold start issue poses special challenges for machine learning parts in ad serving systems, as per Abhishek Shetty's in-depth analysis of optimization methods in self-serve ad platforms [11]. The work documents how new campaigns arrive in the system with little historical data to learn from in order to make targeting and bidding choices, resulting in high uncertainty in early delivery stages. Contemporary platforms overcome this through advanced bootstrapping methods that draw on data from comparable campaigns, contextual information, and well-controlled exploration plans in order to quickly build performance models for new inventory. Such methods generally use probabilistic models that contain uncertainty explicitly at the beginning stages of a campaign and smoothly transition to more deterministic approaches with accumulated performance data.

Privacy legislation is a growing challenge for ad delivery systems, as evidenced by Abhishek Shetty's regulatory impact analysis on advertising technology [11]. The study explains how regulatory policies such as GDPR and CCPA place significant limitations on data collection, processing, and storage practices within the advertising ecosystem. Contemporary platforms have reacted with architectural shifts that inculcate privacy-by-design principles such as data minimization, limitation of purpose, and explicit consent management. These adaptations tend to necessitate large-scale changes to established data streams and processing patterns, adding extra computational overhead and architectural complexity to ensure compliance at the expense of core functionality. Top platforms have become more aggressive in applying methods such as differential privacy and on-device processing to balance personalization features with privacy safeguards, effectively rearchitecting elements of the classic ad delivery architecture.

Conclusion

Real-time ad delivery systems are an advanced convergence of distributed systems engineering, machine learning, and financial technology. Such platforms exhibit impressive architectural sophistication, with competing goals of performance, accuracy, reliability, and scale, and with very tight constraints on the latency at which they can be operated. The tiered design patterns used, intelligent request routing, up to dedicated subsystems of eligibility, ranking, and budget management, demonstrate engineering principles that can be used in many highly performing areas. With the ongoing development of advertising ecosystems, the systems are challenged with an ever-growing difficulty to scale horizontally without loss in their performance, and keep up with the ever-tougher privacy regulations, as well as the integration of more advanced methods of artificial intelligence. The solutions devised to overcome these obstacles (such as sophisticated caching systems, parallel processing models, and graceful degradation systems) offer useful templates to engineers building fault-tolerant applications in other fields than advertising. The knowledge of these systems provides important perspectives regarding the future of the large-scale, latency-sensitive architectures as they evolve within the context of the shifting technological landscapes and user demands.

References

[1] Yodgorbek Komilov, "Unveiling the System Design of Google Ads: Architecture, Scalability, and Reliability," Medium, 2024. [Online]. Available: https://medium.com/@YodgorbekKomilo/unveiling-the-system-design-of-google-ads-architecture-scalability-and-reliability-a8f4adeb947d

[2] Sharon Horsky, "The Changing Architecture of Advertising Agencies," ResearchGate, 2006. [Online]. Available: https://www.researchgate.net/publication/227442424_The_Changing_Architecture_of_Advertising_Agencies

[3] Miltos George, "Real-Time Ad Spend Optimization with AI," Growth-onomics, 2025. [Online]. Available: https://growth-onomics.com/real-time-ad-spend-optimization-with-ai/

[4] MediaCulture, "Programmatic Advertising: The Future of Precision in Performance Marketing," 2024. [Online]. Available: https://www.mediaculture.com/insights/programmatic-advertising-the-future-of-precision-in-performance-marketing

[5] Ajith Moparthi, "Building a Scalable, Real-Time Ad Bidding Platform," Medium, 2025. [Online]. Available:https://medium.com/@ajithmoparthi/building-a-scalable-real-time-ad-bidding-platform-88a9e66bc3d7

[6] Eliana Vuijsje, "The Evolution of Ad Technology Explained," GeoEdge. [Online]. Available: https://www.geoedge.com/the-evolution-of-ad-technology-explained/

[7] Eunji Lee et al., "Caching Strategies for High-Performance Storage Media," ACM Transactions on Storage (TOS), Volume 10, Issue 3, 2014. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/2633691

[8] CelerData, "How Parallel Processing Shaped Modern Computing," 2025. [Online]. Available: https://celerdata.com/glossary/how-parallel-processing-shaped-modern-computing

[9] Vijay Nair, "How to Build a Scalable Marketing Analytics Stack for Growth?" 2025. [Online]. Available: https://www.straive.com/blogs/how-to-build-a-scalable-marketing-analytics-stack-for-growth/

[10] Julian Runge, Steve Geinitz, and Simon Ejdemyr, "Experimentation and Performance in Advertising: An Observational Survey of Firm Practices on Facebook," Meta Research, 2020. [Online]. Available: https://research.facebook.com/publications/experimentation-and-performance-in-advertising-an-observational-survey-of-firm-practices-on-facebook/

[11] Abhishek Shetty, "Optimizing Digital Marketing: Strategies and Challenges in Deploying Self-Serve Advertising Platform," ResearchGate, 2023. [Online]. Available: https://www.researchgate.net/publication/383884338_Optimizing_Digital_Marketing_Strategies_and_Challenges_in_Deploying_Self-Serve_Advertising_Platform