Cross-Cloud Campaign Analytics Meets JavaScript: Building Scalable Frontends for Big Data Insights.

Sruthi Erra Hareram
Independent Researcher, Canada
Conf42.com JavaScript 2025

Presentation Agenda

The Modern Analytics Challenge

Cross-Cloud Architecture Overview

Data Engineering Stack Deep Dive

JavaScript Frontend Implementation

Performance & Optimization

Team Collaboration

Key Takeaways

The Modern Analytics Challenge

1

The Data Deluge: A Marketing Reality

Marketing teams face an overwhelming deluge of data from countless platforms. This volume makes it challenging to process, analyze, and extract real-time insights, often leading to information paralysis.

2

JavaScript: Unlocking Actionable Intelligence

JavaScript acts as a critical bridge, transforming complex data into accessible, interactive experiences. It empowers marketing teams to gain immediate, actionable intelligence for strategic decision-making.

Architecture at a Glance: Fueling Marketing Intelligence

Unified Data Ingestion

Harnessing the power of Amazon S3, we centralize vast reservoirs of raw campaign data, streamed directly from myriad marketing platforms, forming the bedrock of our analytics.

Intelligent Data Transformation

Our robust processing layer, powered by Azure Databricks and Data Factory, meticulously orchestrates complex ETL workflows, refining raw data into structured, ready-for-analysis intelligence.

Secure Analytics Delivery

Through a secure and scalable API Gateway, we provide seamless, real-time access to our meticulously processed analytics and critical marketing metrics via intuitive RESTful endpoints.

Empowering Frontend Experience

The heart of user interaction, our dynamic React frontend transforms complex data into stunning, real-time dashboards, empowering marketing teams with immediate, actionable campaign intelligence at their fingertips.

Why Cross-Cloud?

Unleash Best-of-Breed Tools

Leverage the unparalleled analytics power of Azure and the vast storage scalability of AWS, breaking free from vendor lock-in to drive innovation and flexibility.

Enhanced Data Locality & Performance

Strategically store data closest to its origin for optimized access, while processing it in the environment that offers peak efficiency and minimal latency.

Dynamic Cost Optimization

Intelligently select the most cost-effective storage and compute solutions for each specific workload, ensuring maximum value and superior return on investment across your infrastructure.

Amazon S3

Role: Data lake foundation

- Ingests raw campaign logs
- Partitioned by date and source
- Supports multi-region access

Azure Data Factory

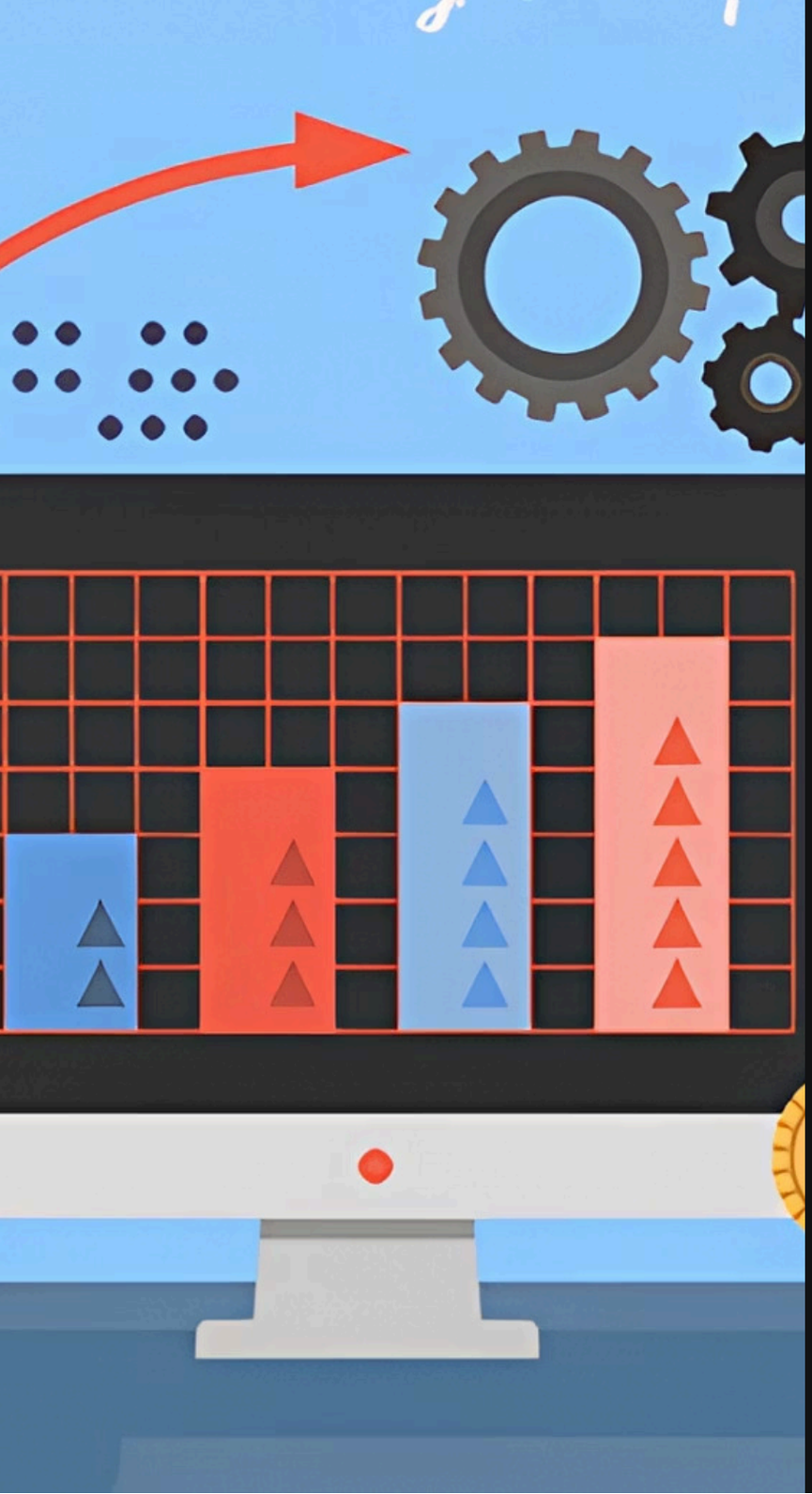
Role: Orchestration engine

- Schedules ETL pipelines
- Manages data movement
- Monitors job health

Azure Databricks

Role: Processing powerhouse

- Spark-based transformations
- Machine learning models
- Real-time aggregations



Processing Scale in Numbers

Daily Data Volume

Campaign logs processed from multiple marketing platforms every 24 hours

Events Per Day

Individual user interactions tracked and analyzed across all campaigns

Query Response

Average API response time for complex aggregation queries

Uptime SLA

Guaranteed availability for mission-critical marketing insights

From Backend to Browser

The frontend challenge wasn't just about displaying data; it was about transforming an overwhelming flood of billions of raw data points into something truly comprehensible and actionable for marketing professionals. This wasn't merely a cosmetic task, but a critical translation layer.

We faced the complex task of designing interfaces that could fluidly handle immense volumes of real-time analytics – from campaign logs and user interactions to performance metrics across diverse platforms. The goal was to empower users to navigate, filter, and extract meaningful insights without being bogged down by the underlying complexity.

JavaScript emerged as the indispensable cornerstone for bridging this gap. Leveraging its dynamic capabilities, we built interactive dashboards and intuitive visualization tools. It enabled us to process and aggregate data client-side, render sophisticated charts and graphs, and provide instantaneous feedback, effectively turning raw numbers into strategic marketing intelligence that drives informed decisions.

React Dashboard Architecture

01

Component Library

Reusable chart components built on D3.js and Recharts for consistent visualization patterns

02

State Management

Redux toolkit managing complex filter states, user preferences, and cached query results

03

API Integration

Axios interceptors with retry logic, request queuing, and intelligent caching strategies

04

Performance Layer

React.memo, virtual scrolling, and web workers for smooth rendering of large datasets

Handling Large Datasets Client-Side

The Problem

Marketing dashboards need to display thousands of data points. Loading everything freezes the browser. Traditional pagination breaks the user's mental model of exploring data.

Our Techniques:

- Virtual scrolling for table views
- Progressive data loading
- Client-side aggregation for drill-downs
- Web Workers for heavy computations

Power BI Integration Pattern

1 Authentication

Azure AD SSO passes user context to embedded reports

2 Embedding

Power BI JavaScript SDK loads reports in React components

3 Interaction

Custom filters sync between React state and Power BI visuals

4 Export

Programmatic export to PDF/Excel via JavaScript API calls

The Power BI JavaScript SDK enabled deep integration—React components could control filters, handle events, and even trigger exports programmatically. This created a seamless experience where Power BI felt native to the application.

Visual Storytelling with Data

Start with the "Why"

Craft compelling chart headlines that immediately reveal the core insight, transforming raw metrics into actionable narratives.

Guide the Discovery

Present critical high-level KPIs first, then enable intuitive drill-downs so users explore details only when their curiosity is piqued.

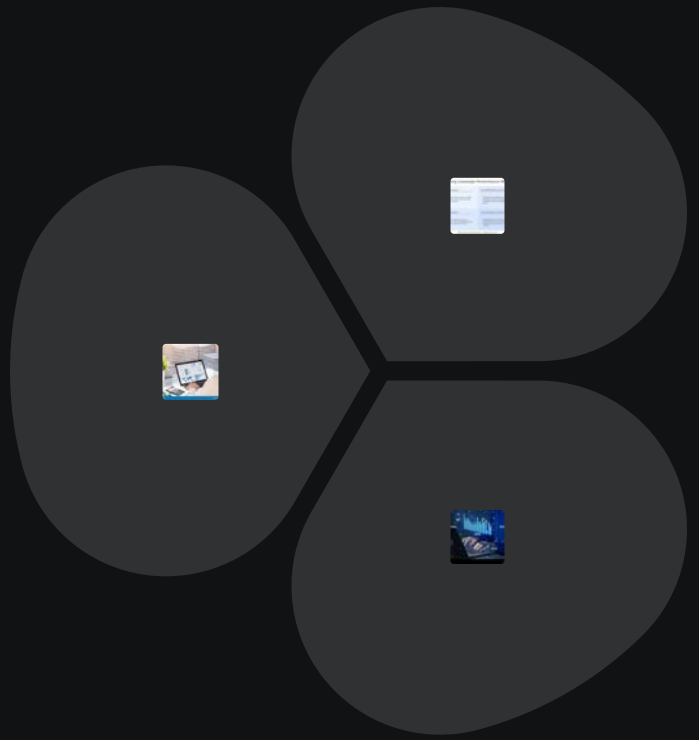
Illuminate with Context

Never leave data in isolation. Showcase powerful trends, crucial comparisons, and relevant benchmarks to unlock deeper understanding and strategic insights.

Role-Based Analytics Views

Executive View

Empowering executives with a crystal-clear, high-level strategic overview. Instantly grasp critical KPIs, identify key trends, and monitor budget pacing to make swift, informed decisions. Maximum clarity for maximum impact.



Campaign Manager

Equip campaign managers with the tools to dominate. Dive deep into campaign performance, analyze A/B test outcomes, and receive intelligent optimization recommendations to maximize ROI and drive superior results.

Analyst View

Unleash the power of data for analysts. Gain unrestricted access to raw data, leverage intuitive custom query builders, and export insights seamlessly for in-depth exploration and uncovering hidden opportunities.

Performance Optimization Strategies

Code Splitting

Lazy-load dashboard modules only when users navigate to them, significantly reducing initial bundle size for faster loading

Smart Caching

Cache API responses with TTL strategies static data cached longer, real-time data refreshed frequently

Data Pagination

Server-side pagination with client-side caching enables browsing extensive datasets smoothly

Debouncing & Throttling

Limit API calls during rapid filter changes, reducing server load and improving responsiveness

Collaboration Between Teams

Breaking Down Silos

Success required tight collaboration between data engineers and frontend developers. Weekly syncs aligned API contracts with UI needs. Shared documentation in Confluence kept everyone on the same page.

Key Success Factors:

- API-first design with OpenAPI specs
- Shared testing environments
- Joint performance reviews
- Cross-functional standups

Key Takeaways

JavaScript is the bridge

Modern analytics platforms need JavaScript to translate complex data infrastructure into intuitive user experiences

Performance is non-negotiable

Client-side optimization techniques—virtual scrolling, caching, web workers are essential when handling large datasets

Cross-functional teams win

Frontend engineers and data engineers must collaborate closely from day one to build scalable, maintainable analytics platforms

Thank You!