**VPBank Technology Hackathon 2025 – Senior Track**

General Brief

Please fill up this table and use this document as a template to write your proposal.

| **Challenge Statement** | Modernized Data Platform for Promotion Campaigns — Uplift Engine |
| --- | --- |
| **Team Name** | Uplift Engine |

**Team Members**

**Content Outline**

**Solutions Introduction**

Uplift Engine is a modern, configurable promotion data platform that turns promotions into profit by targeting persuadable customers—those who convert because of the treatment. It combines a no-code Rule Engine (for business users) with Causal AI uplift modeling, low-latency decisioning, and transparent explanations.

Key features:

* Business-user configurability: Campaign Configuration UI (React/Amplify) + AppSync GraphQL + DynamoDB for versioned rules; publish/rollback without deployments.
* Explainability-by-design: Every decision includes human-readable reason codes (e.g., "Customer spent 5M VND in Dining this week, reached Gold tier"). TreeSHAP for tree models; async SHAP fallback for complex learners.
* Full-spectrum processing: Batch (Glue/EMR), near real-time (Kinesis/MSK + Lambda), and real-time (<100ms) via API Gateway + Lambda + SageMaker Feature Store + Endpoint + Redis cache.
* AWS-native, serverless-first: Lambda (Provisioned Concurrency), Step Functions, SageMaker (Feature Store, Training, Endpoints), DynamoDB, Redshift, ElastiCache Redis, Kinesis/MSK.
* Optimizer & guardrails: Knapsack optimizer for budget allocation; do-no-harm guardrails (DNC, confidence lower bound, frequency capping).

**Impact of Solution**

Business impact:

* Focus squad/product: Initial pilot with Credit Cards squad (or CASA) to maximize measurable Net Profit Uplift and speed up adoption across squads.
* Higher ROI, lower waste: Focus budget on Persuadables; avoid Sure Things/Lost Causes; reduce wasted spend (up to ~70%) and improve ROI (simulated +308%), while preserving brand trust via guardrails.
* Faster time-to-market: Business users change rules without code; publish instantly with audit/versioning.
* Personalization at scale: Real-time eligibility + uplift scoring with P95 < 100ms; near real-time streaming updates for contests/challenges.

Why this solution is better:

* Causal AI advantage: Optimizes net profit uplift, not propensity—aligns with true business outcomes.
* Consistency by design: SageMaker Feature Store removes training-serving skew; same features across offline/online.
* Operational excellence: Redis caching, observability with CloudWatch, and performance targets/benchmarks included.

Differentiators (USP):

* Explainability as a core MVP feature in inference (reason codes/SHAP).
* No-code Rule Engine tightly integrated with uplift/optimizer.
* Tiered architecture (Glue/EMR) for cost-performance balance.

**Deep Dive into Solution**

End-to-end flows:

* Data ingestion & features: Structured (profiles, transactions) and unstructured (text/clickstream) data handled via Glue/EMR pipelines; point-in-time correctness enforced; features stored in SageMaker Feature Store (Offline/Online).
* MLOps: Step Functions orchestrates feature jobs, parallel training (CatBoostUplift, DR-Learner, CausalForest), best model selection by Profit@K, registration/deployment to SageMaker Endpoint.
* Real-time decisioning: API Gateway → Lambda (Provisioned Concurrency) → Online Feature Store (GetRecord) → Uplift Endpoint → Guardrails → Optimizer → Response with explanation; exposures logged to Kinesis/MSK and fed to Redshift for analytics.
* Rule Engine: Business UI → AppSync → DynamoDB (draft/publish) → Lambda CompileRule → Redis cache of compiled rules; Decision Lambda evaluates eligibility in ~1–5ms via Redis.
* Streaming & near real-time: Kinesis/MSK updates aggregates and triggers eligibility/contest logic (leaderboards in Redis/Redshift).

Contract examples:

* Request (client → API): { customerId, context }
* Response: decision, offer, uplift\_score, uplift\_std\_error, explanation.primary\_factors, threshold\_met

Demo scenarios (prototype-ready):

* Batch cashback: Weekly/monthly cap + rule threshold; batch evaluation via Glue/EMR; outputs to Redshift with explanations.
* Real-time sales contest: First N customers meeting spend within a time window; streaming updates + real-time rule evaluation and leaderboard refresh.

**Architecture of Solution**

AWS services and how they’re used:

* API Gateway + Lambda (Provisioned Concurrency): Low-latency decision API; cold-start eliminated.
* SageMaker Feature Store: Offline (S3/Parquet) for training/analytics; Online (DynamoDB) for <10ms feature reads.
* SageMaker Endpoints: Real-time uplift inference; TreeSHAP inline where applicable; autoscaling by RPS/CPU.
* AppSync + Cognito + S3/CloudFront (UI): Business Rule Builder UI with auth; GraphQL mutations to author/publish rules.
* DynamoDB + Redis: Versioned rule storage and 1–5ms compiled rule cache; invalidation on publish/rollback.
* Kinesis/MSK + Firehose: Exposure/outcome streams; sinks to S3/Redshift for analytics/monitoring.
* Glue/EMR Serverless: Tiered batch/feature engineering pipeline with cost/performance optimization.
* Redshift: Analytics warehouse for campaign KPIs, eligibility batches, explainability logs, and historical attribution.

Performance & Benchmarks (per 4.2.2):

* Real-time decisioning: P95 < 100ms, P99 < 150ms (API Gateway + Lambda + Feature Store + Endpoint + Rule eval + Guardrails).
* Batch processing: ≥ 1,000,000 transactions in ≤ 6h on Glue/EMR Serverless; scalable to 10,000,000/day with autoscaling and partitioning.
* Streaming ingest: Kinesis/MSK ≥ 50,000 events/sec; Feature Store writes ≥ 10,000 records/sec with parallel writers.

Processing modes covered as required: Batch, Near real-time, and Real-time.

Architecture diagram:

