**🟦 MobUpps**

**חלק א׳ – תכנון ארכיטקטורה למערכת A/B Testing**

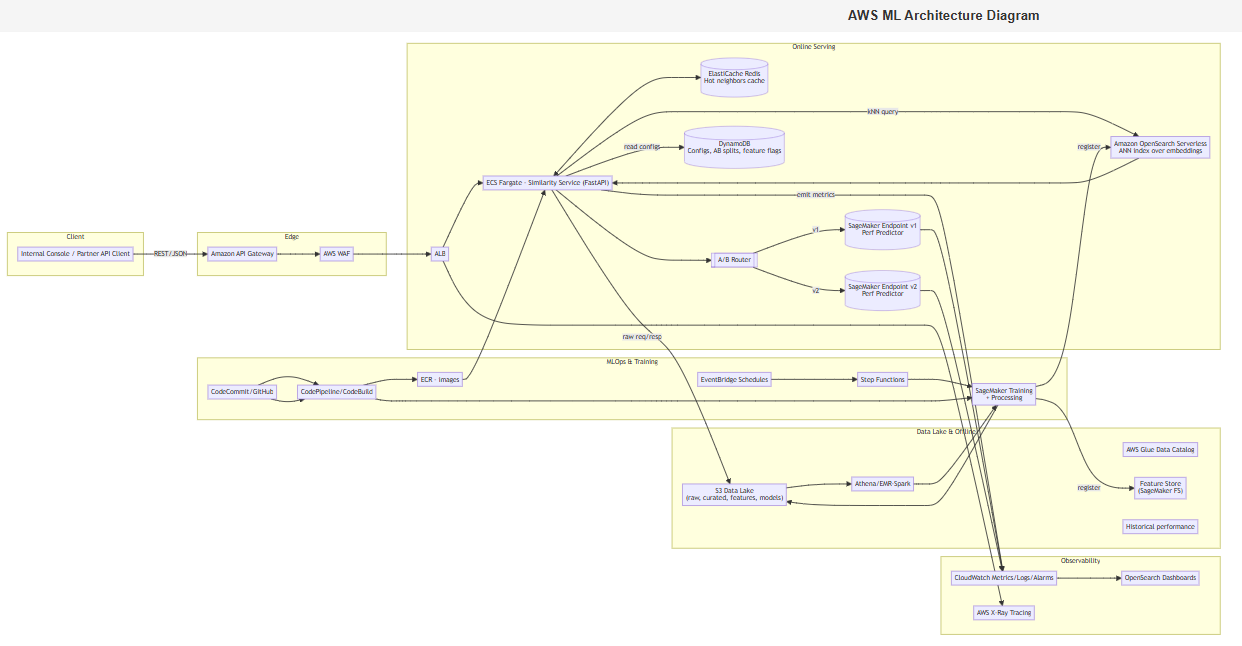
**שם:** אורטל לסרי  
**מטלת הגשה לתפקיד:** ML Engineer – AI Team  
**תאריך הגשה:** 19.10.25

**🧭 תיאור כללי – Use Case**

המערכת של MobUpps נועדה לחזות את **ביצועי ההצלחה הצפויים** של אפליקציה חדשה בשוק.  
תהליך החיזוי כולל שלושה שלבים עיקריים:

1. קבלת **מטא־דאטה** של האפליקציה החדשה (קטגוריה, אזור, מחיר, פיצ׳רים וכו׳).
2. מציאת **10–20 אפליקציות דומות** באמצעות חיפוש וקטורי (ANN).
3. חיזוי **performance score** (ציון הצלחה צפוי) ו־**user segments** (קהל יעד רלוונטי).

בנוסף, המערכת כוללת **שני מודלים שונים (v1 ו-v2)** שעליהם מתבצע **A/B Testing אוטומטי** כדי לזהות איזה מודל מדויק יותר.

  
  
  
  
  
**🗺️ תרשים גרפי – תיאור מילולי**

**לקוח → API Gateway + WAF → ALB → ECS (שירות FastAPI)**  
ECS אחראי על שליפת embedding, חיפוש אפליקציות דומות ב־OpenSearch, והעברת הבקשה ל־A/B Router.  
ה־Router מחליט לאיזה מודל לשלוח (v1 או v2 ב־SageMaker).  
התחזית + הפילוח מוחזרים ללקוח, ונשמרים ב־S3 לצורך ניתוח עתידי.  
מאוחר יותר, נתוני ה־ground truth נכנסים ל־S3, משם ל־Athena/Glue → SageMaker retraining.

**🧩 רכיבי המערכת העיקריים**

**Edge & Security**

* **API Gateway + WAF** – ניהול גישה, אימות (JWT), הגנה מפני עומסים ו־rate limiting.
* **Load Balancer (ALB)** – ניתוב פנימי לשירותים.

**Online Serving**

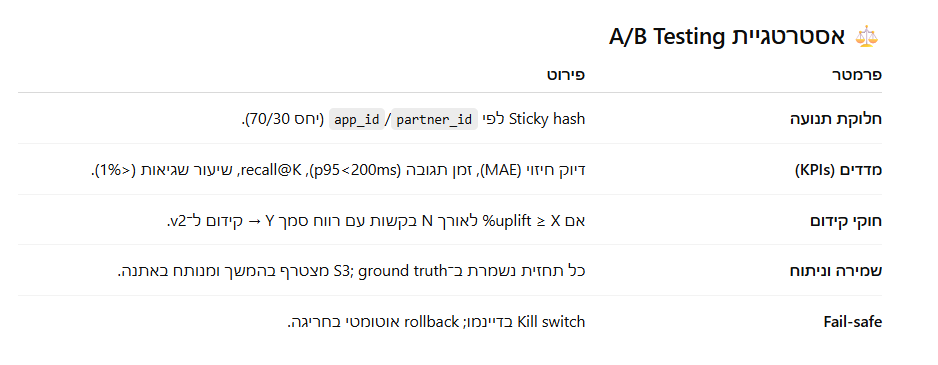
* **ECS Fargate (FastAPI)** – שרת חיפוש דומים וחיזוי בזמן אמת.
* **ElastiCache Redis** – קאש לשאילתות נפוצות (neighbors).
* **OpenSearch Serverless** – אינדקס וקטורי (HNSW/IVF-PQ) עבור embeddings v1/v2.
* **DynamoDB** – ניהול ניסויים ו־feature flags ל־A/B.
* **SageMaker Endpoints (v1/v2)** – מודלים בזמן אמת לחיזוי ביצועים ו־user segments.

**Offline & MLOps**

* **S3 + Glue + Athena** – מאגר נתונים גולמיים, תחזיות, ונתוני אמת.
* **SageMaker Training + Step Functions** – pipelines לאימון מחדש והערכת ביצועים.
* **Feature Store** – ניהול פיצ׳רים משותפים לאון-ליין ואוף-ליין.

**Monitoring**

* **CloudWatch + X-Ray** – ניטור ביצועים, זמני תגובה, שגיאות ו־trace מלא.
* **OpenSearch Dashboards** – דשבורדים לניטור ה־A/B.

  
  
**🔁 Feedback Loop**

1. תחזיות נרשמות ב־S3 (raw predictions).
2. נתוני אמת (ground truth) נוספים לאחר מכן (historical\_performance.csv).
3. אתנה/Glue מחברים בין התחזיות לאמת.
4. SageMaker מחשב ביצועים ומתרגל מודלים חדשים.
5. Step Functions מבצע rollout גרסה חדשה (v3).

**⚙️ CI/CD**

* **קוד:** GitHub → CodeBuild/CodePipeline.
* **בדיקות:** יחידה, אינטגרציה, עומס.
* **תוצרים:** Docker images ל־ECR; מודלים ל־S3.
* **פריסה:** Terraform/CDK → ECS & SageMaker (blue/green).
* **ניהול גרסאות:** tag לפי model\_version.

**📈 Scaling ו־Cost Optimization**

* **ECS Fargate** – autoscale לפי CPU/memory; Stateless.
* **OpenSearch** – הוספת shards/replicas; דחיסת PQ לוקטורים.
* **SageMaker Serverless** – שימוש רק בשעות עומס.
* **S3 Lifecycle** – העברת נתונים ישנים ל־Glacier.
* **CloudWatch Logs** – דגימת לוגים לצמצום עלות.

**🔐 אבטחת מידע**

* **IAM Least Privilege**, הצפנה ב־KMS לכל שכבות המידע.
* **PrivateLink / Security Groups** לפי צורך.
* **WAF Rules** והגבלת תעבורה לפי IP.

**📋 הנחות ושאלות פתוחות**

* חישוב embedding נעשה באמצעות lookup מה־Feature Store (לא מודל כבד).
* ground truth מתקבל בדיליי של מספר ימים.
* אין מידע אישי (PII).
* יעד ביצועים: זמני תגובה p95 < 200ms, זמינות 99.9%.

**✅ סיכום**

מערכת זו משלבת בין Online Prediction ל־Offline Feedback Loop,  
מאפשרת השוואה בזמן אמת בין שני מודלים שונים (v1/v2),  
ובונה תהליך אוטומטי לשיפור מתמיד של תחזיות הביצועים, תוך ניטור, סקיילינג, ובקרה הדוקה על עלויות.  
  
**☁️ רשימת שירותי AWS בפרויקט**

| **שכבה** | **שירות AWS** | **תפקיד במערכת** |
| --- | --- | --- |
| **Edge / API Layer** | **Amazon API Gateway** | חשיפה מאובטחת של ה־API החיצוני (/find-similar, /predict) למשתמשים פנימיים או שותפים. |
|  | **AWS WAF (Web Application Firewall)** | הגנה על ה־API מפני התקפות (SQLi, rate limiting, XSS וכו׳). |
| **Compute Layer** | **AWS Fargate (ECS)** | הרצת ה־FastAPI (microservice שלך) במכולות ללא צורך לנהל שרתים. |
|  | **Application Load Balancer (ALB)** | ניתוב תעבורה בין קונטיינרים של Fargate; מאפשר סקיילינג אוטומטי. |
| **Model Inference** | **Amazon SageMaker Endpoint (v1, v2)** | אירוח שני המודלים שלך (גרסה 1 ו־2) בזמן אמת, לטובת ה־A/B Testing. |
| **Similarity Search** | **Amazon OpenSearch Serverless (Vector Index)** | אחסון embeddings וביצוע חיפוש דמיון (ANN) על אפליקציות קיימות. |
| **Cache Layer** | **Amazon ElastiCache (Redis)** | שמירת תוצאות חמות של neighbors כדי להאיץ בקשות חוזרות. |
| **Configuration & Flags** | **Amazon DynamoDB** | אחסון של ניסויים, אחוז חלוקת זרועות A/B, feature flags וכו׳. |
| **Data Lake / Storage** | **Amazon S3** | אחסון קלט/פלט: embeddings, תוצאות תחזיות, ground truth, נתוני אימון חוזר. |
|  | **AWS Glue Data Catalog** | קטלוג מבנה הקבצים ב־S3 לצורך שאילתות נוחות. |
|  | **Amazon Athena** | הרצת שאילתות על נתוני התחזיות/ביצועים לצורך ניתוח offline. |
| **Pipeline & Automation** | **AWS Step Functions** | orchestration לתהליכי retraining, evaluation ו־re-deploy. |
|  | **AWS CodeBuild / CodePipeline / ECR** | CI/CD מלא לקוד שלך: בנייה, בדיקה, הפצה ו־container registry. |
| **Observability** | **Amazon CloudWatch** | ניטור מדדים (latency, error rate, throughput), לוגים ו־alarms. |
|  | **AWS X-Ray** | ניתוח בקשות מקצה לקצה (trace של בקשה דרך כל השירותים). |
|  | **OpenSearch Dashboards** | תצוגה של דוחות/מדדים (dashboard של A/B Test). |
| **Security & IAM** | **AWS KMS** | הצפנת דאטה (S3, DynamoDB, OpenSearch וכו׳). |
|  | **AWS IAM** | בקרת גישה והרשאות מינימליות בין השירותים. |
| **Scheduling / Eventing** | **Amazon EventBridge** | הפעלת תהליכים מתוזמנים (retrain, evaluation, cleanup). |

**🔎**

נספח ל**חלק א׳ – תכנון ארכיטקטורה למערכת A/B Testing**”  
כל השירותים האלה מופיעים ב:

“## Architecture (AWS)”  
ובתרשים ה־Mermaid, מתחת ל־subgraphים:

* Edge
* Online Serving
* Data Lake & Offline
* MLOps & Training
* Observability

תרשים mermaid קיים ב repo aws-assignment-mobupps בגיטהאב תחת architecture-diagram.md כשלוחצים בגיטהאב אפשר לראות את התרשים הגרפי בצורה מאוד יפה וברורה.