# Deployment of the model

## Objectives

* Deploy an App Engine endpoint to serve recommendations.
* Deploy a Cloud Composer environment to orchestrate model training updates.

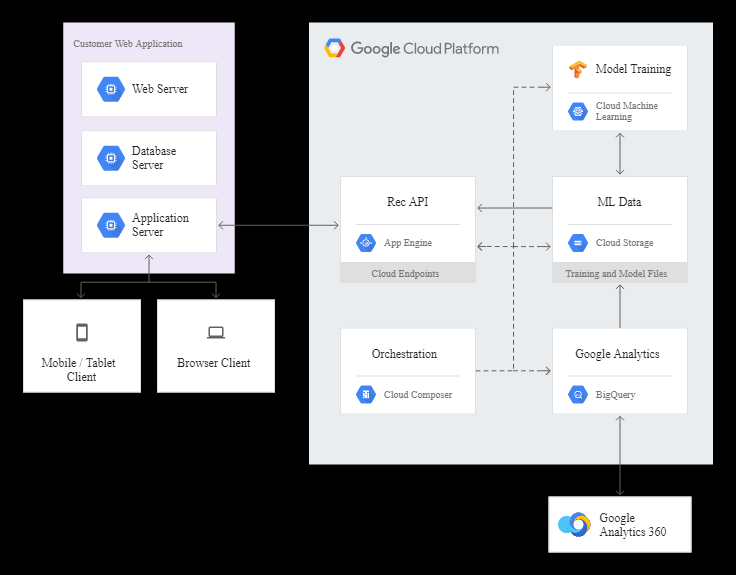
To make recommendations available, you can provide a REST API endpoint that serves the recommendations to a client application. This REST endpoint takes a Google Analytics client ID and number of recommendations **K** as inputs, and returns **K** number of article IDs that are recommended for that user.

To incorporate new website articles and user-click data into the recommendations, you must periodically retrain and redeploy the model. This involves these steps:

* Query training data from Google Analytics tables in Big Query.
* Export that data to Cloud Storage as a CSV file.
* Train a new recommendation model on the data.
* Deploy the updated model to the API.

You can use [Cloud Composer](https://cloud.google.com/composer) to implement and manage this sequence. Cloud Composer is a managed service for [Apache Airflow](https://airflow.apache.org/) on Google Cloud. Apache Airflow is an orchestration manager for sequences of tasks in which each step involves an operation performed by a cloud service, and in which each step depends on the successful completion of the previous step. The tutorial shows you how to deploy a Cloud Composer environment and provides sample configuration of a workflow represented in a directed acyclic graph ([DAG](http://airflow.apache.org/concepts.html#dags)) to perform the model update steps.

The following diagram shows the complete architecture of the deployed solution.



## Deploying the recommendation service

1. Open a command shell on a computer where you've installed the Cloud SDK and the tutorial code.
2. Change to the top-level directory for the solution code.
3. Make a Cloud Storage bucket with the name recserve\_[YOUR-PROJECT-ID]

### Set up WALS model package and model data

The training task in the Airflow DAG requires the code package for the model to be deployed in a static location in Cloud Storage.

1. Create a distributable package for the model code using the setup script in the model directory
2. Copy the package to the code folder in the bucket you created
3. When the recommendation service is launched, it requires the [NumPy](http://www.numpy.org/) array model files. To create these initial model files, run an AI Platform job on the initial training data, with an output directory argument pointing to your Cloud Storage bucket.

### Install the API endpoint and application

1. Create the App Engine application.
2. Prepare to deploy the API endpoint service.
3. Run the command that is output by the prepare\_deploy\_api.sh script
4. Prepare to deploy the App Engine app.
5. Run the command that is output by the prepare\_deploy\_app.sh script

## Using the recommendation service

### Recommendation endpoint service

The [OpenAPI](https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md" \t "github) service specification file for the recommendation endpoint is app/openapi.yaml. For more information on using Endpoints see the [documentation](https://cloud.google.com/endpoints/docs/openapi/about-cloud-endpoints).

To test the recommendation service endpoint, you can use the query\_api.sh script.

The output of that script shows a sample call to the API, with the JSON return value:

curl "https://[PROJECT\_ID].appspot.com/recommendation?userId=5448543647176335931&numRecs=5"

{

"articles": [

"299941605",

"299800704",

"298299208",

"299800661",

"299928862"

]

}

----------------------------------------------------------------------------------------------------------------