



FeedMe

A Restaurant Engagement Manager

Vamsi Gadiraju, Harsh Yadav, Vishruth Nair

Today's Discussion

Current Problem and Motivation

Our Solution

Our System Architecture

Key Technologies

Why Cloud?

Demo

**Why did God give me two ears
and one mouth? So that I will
hear more and talk less.**

LEO ROSTEN

\$365 billion will be spent around the world on food delivery by 2030

Increasingly popular to use virtual kitchens (delivery-only restaurants)

Extremely difficult for restaurants to actively engage with customers and drive user reviews and feedback



SPUR NEW USER REVIEWS

through personally-tailored messages and websites for reviews and user interaction



AGGREGATE MESSAGES AND SEARCH

to create a one stop shop for user response management and engagement analytics across your restaurants



INTEGRATE CHAT THROUGH TEXT

on our web-based chatting platform and maintain persistent chats with users

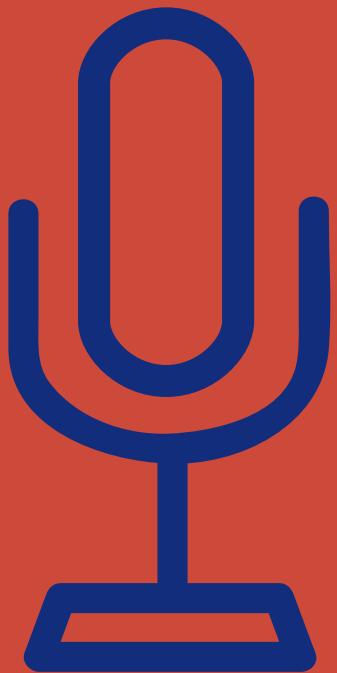
Our Goals and Uniqueness

Key Parts



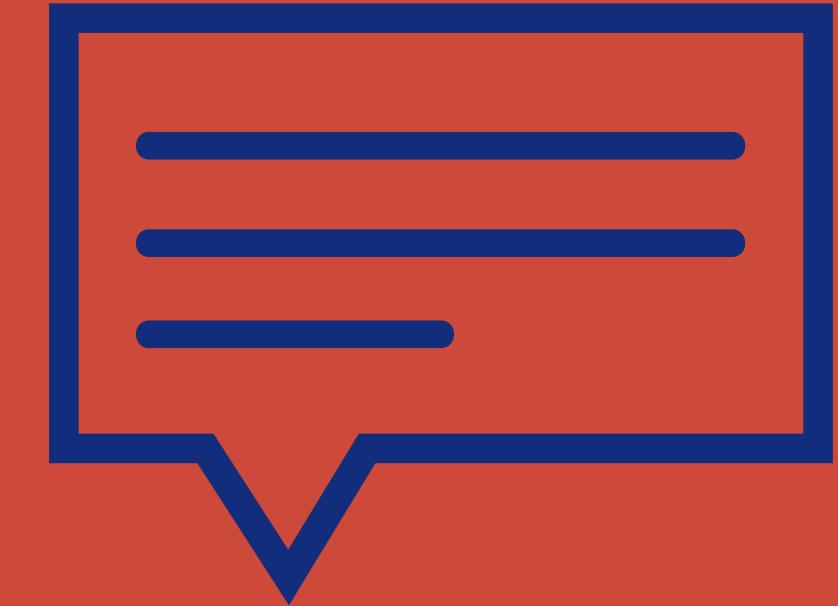
USER DASHBOARD FOR KEY ANALYTICS

that shows the number of overall reviews, stars, and growth metrics for engagement



ENGAGEMENT FEEDS AND AUTOMATED MESSAGES

to have a catalog of User reviews. Restaurant can also see the message left, any conversational information



IN-APP MESSAGING THROUGH TEXT

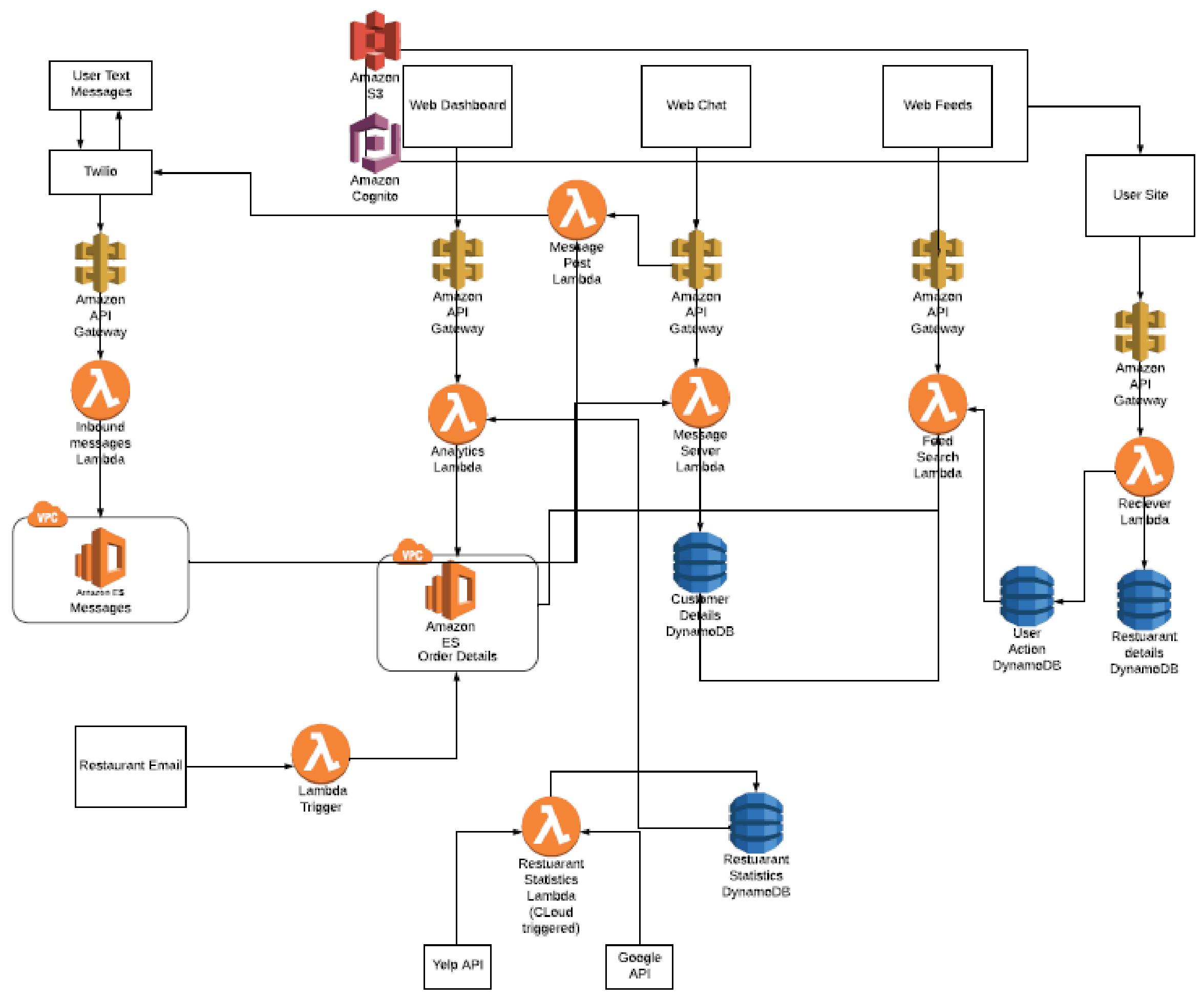
where customers reply using text messaging and the restaurant can use the wbe platform to respond.

Key Technologies

Lambda
DynamoDB
API Gateway
Cognito
ElasticSearch
Twilio

VPC
Sagemaker
S3
CloudWatch
SQS
SES

Architecture



Lambdas

1. Text Message Webhook
2. Review Aggregator
3. Chat handler
4. Dashboard Formatter
5. Order Creator
6. Message Link Handler
7. Analytics Processor
8. Machine Learning Engine
9. Review Handler

Data Model

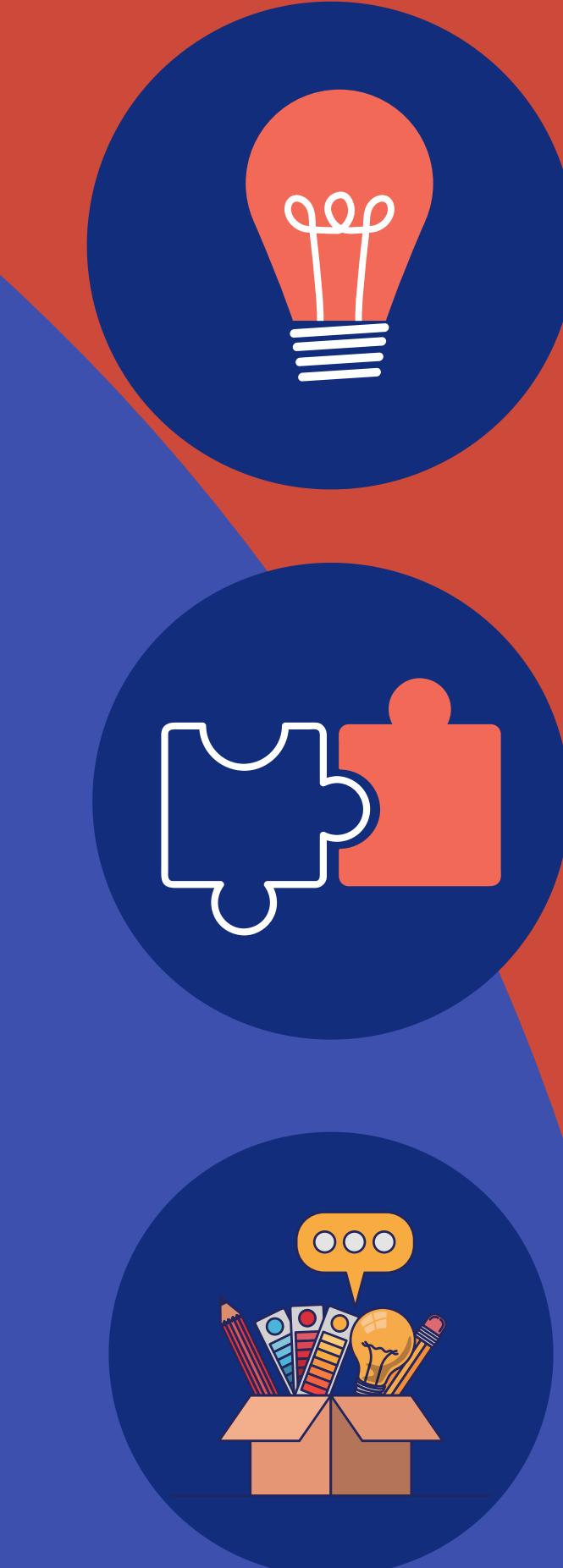
DynamoDB

1. Customer Details
2. Restaurant Details
3. User Actions
4.

ElasticSearch

1. MessagesMetadata
2. OrderMetadata
3. UserAnalytics
4. Restaurant Analytics

Why Cloud?



Use Existing Robust Resources

to collectively form and extremely powerful application without having to deal with the peripheral maintenance/development overhead.

Modularize Project Components

to promote CI/CD framework and allow for segmented development and ancillary rollout

High Scalability With Low Overhead

through our serverless architecture and highly scalable data storage/search technologies



Demo Link

Thanks!

