CREATE A WEBSITE AND API WITH AWS LAMBDA

A starting guide on the serverless stack

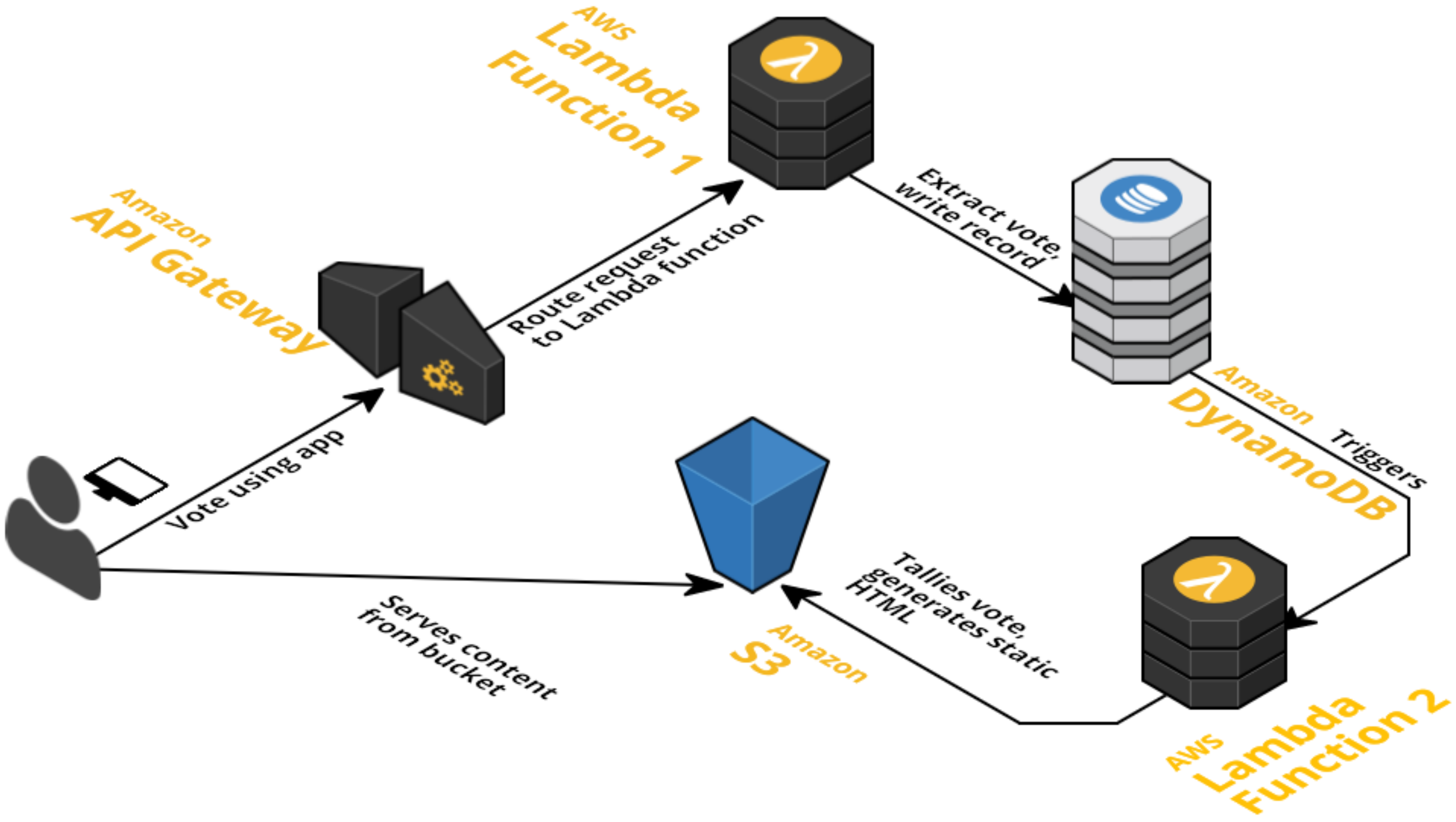
WHY SERVERLESS

- No server infrastructure management
- Scaling is simple and precise, only pay for what you use
- Decoupled architecture keeps you agile

WHAT IS THE SERVERLESS STACK

- API Gateway provides http(s) endpoints
- Lambda executes business logic
- DynamoDB or RDS provides database
- S3 Hosts Static Web Content
- CloudFront provides CDN/HTTPS
- Route53 provides DNS

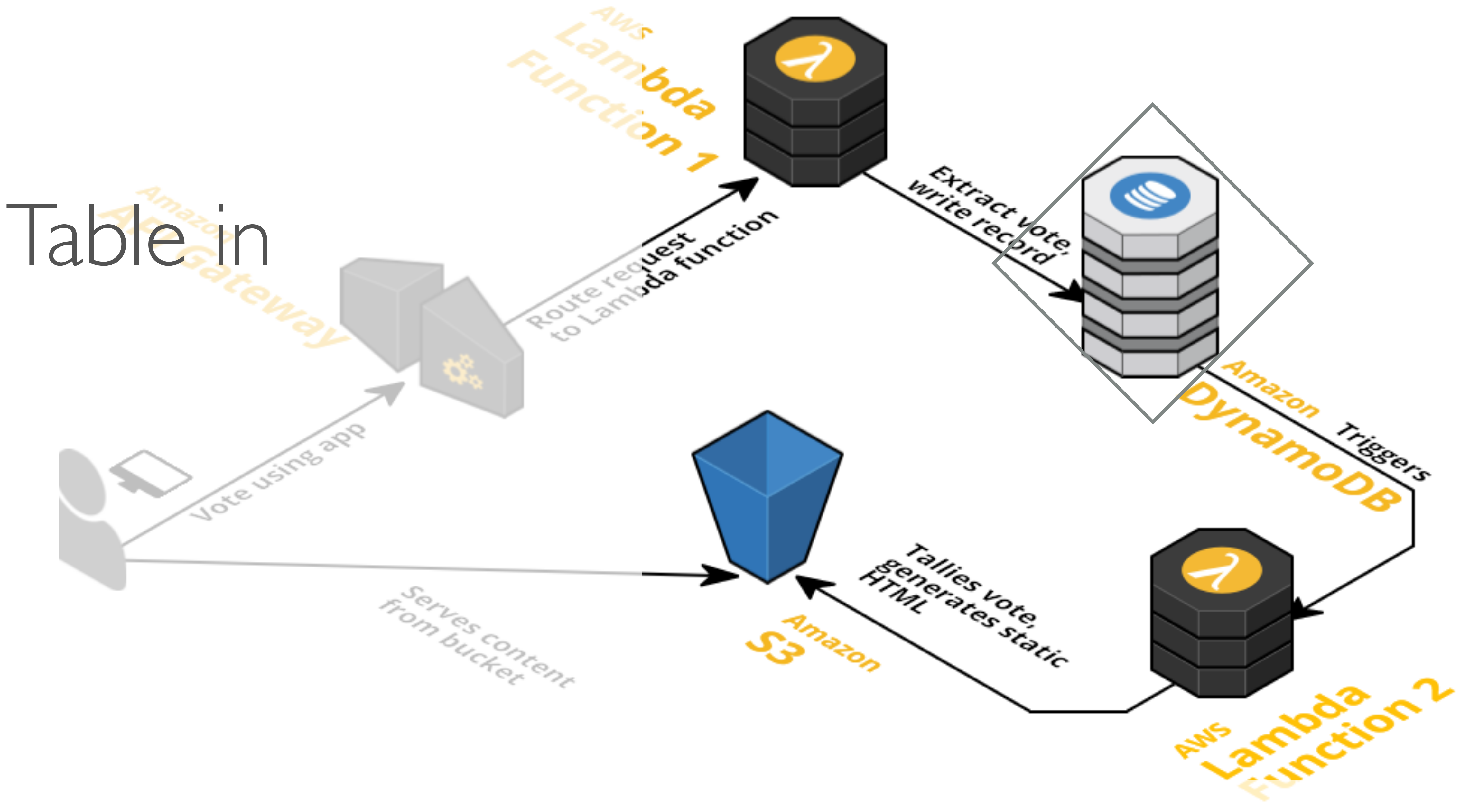
VOTING APP ARCHITECTURE



Created with https://cloudcraft.co/

STEP 1: SET UP DYNAMODB

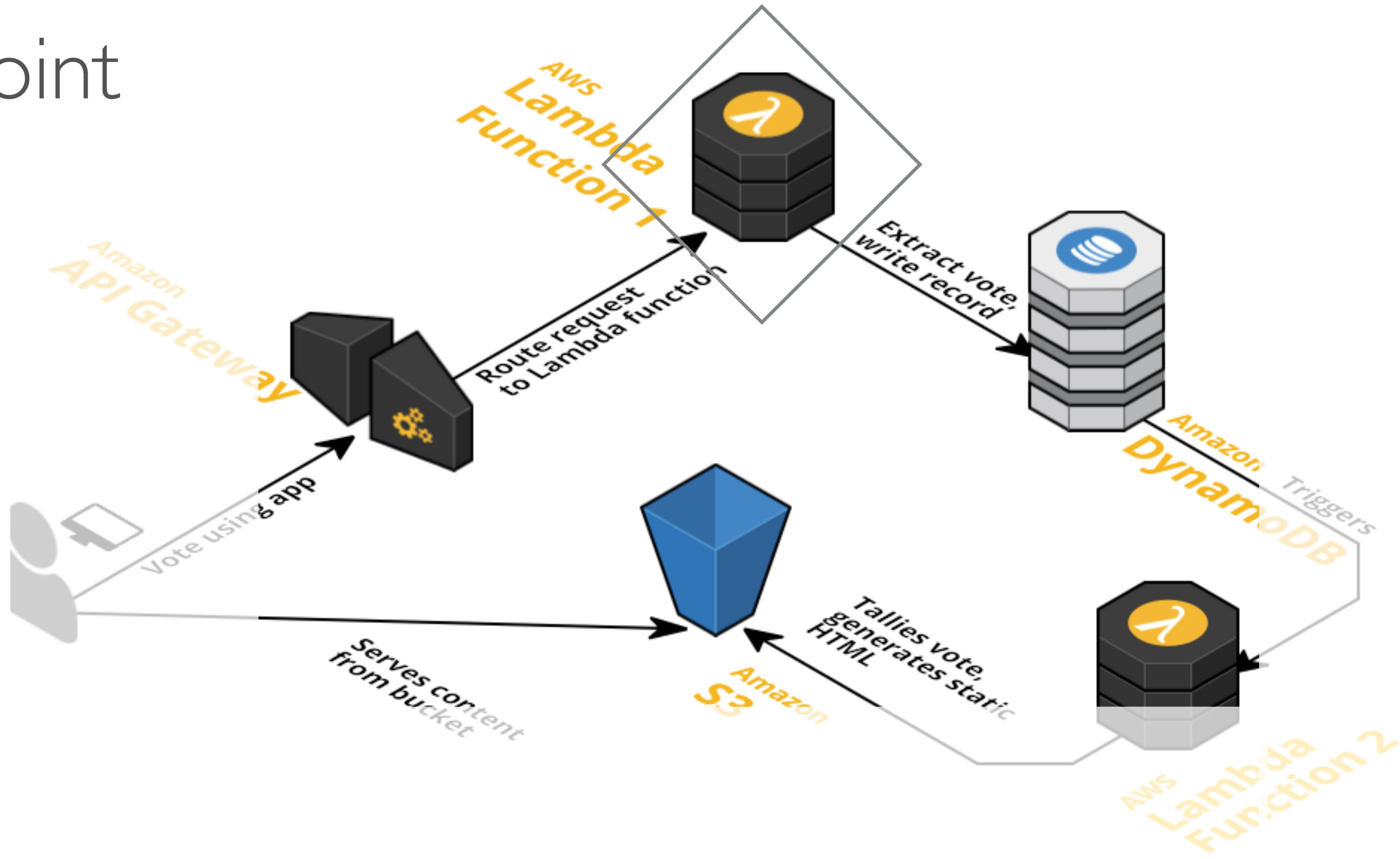
- Create Votes Table in DynamoDB



STEP 2: SET UP LAMBDA FUNCTION 1, THE VOTE CASTER

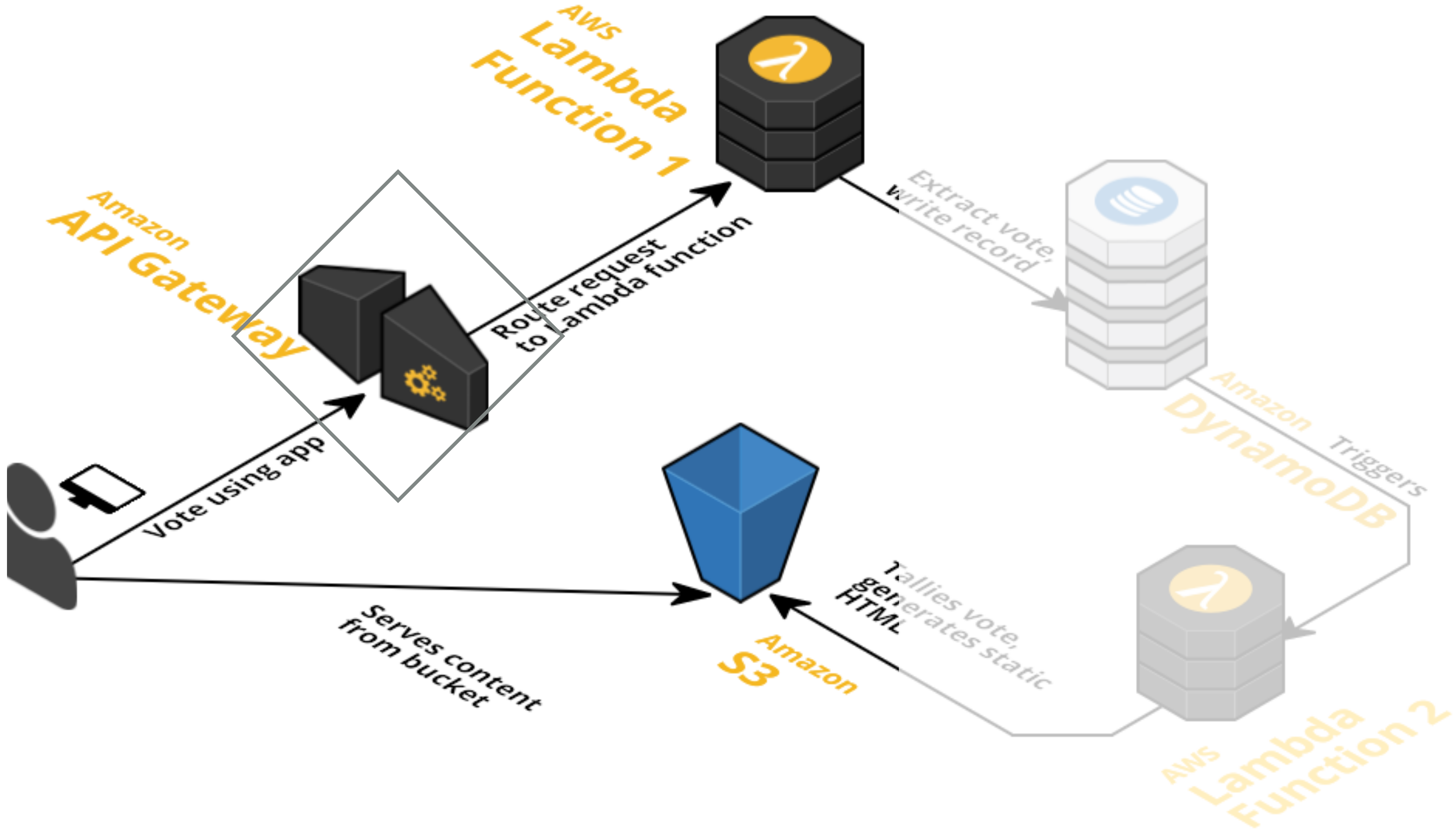
- Create a new Lambda with microservice-http-endpoint blueprint

- Modify code to update Votes table



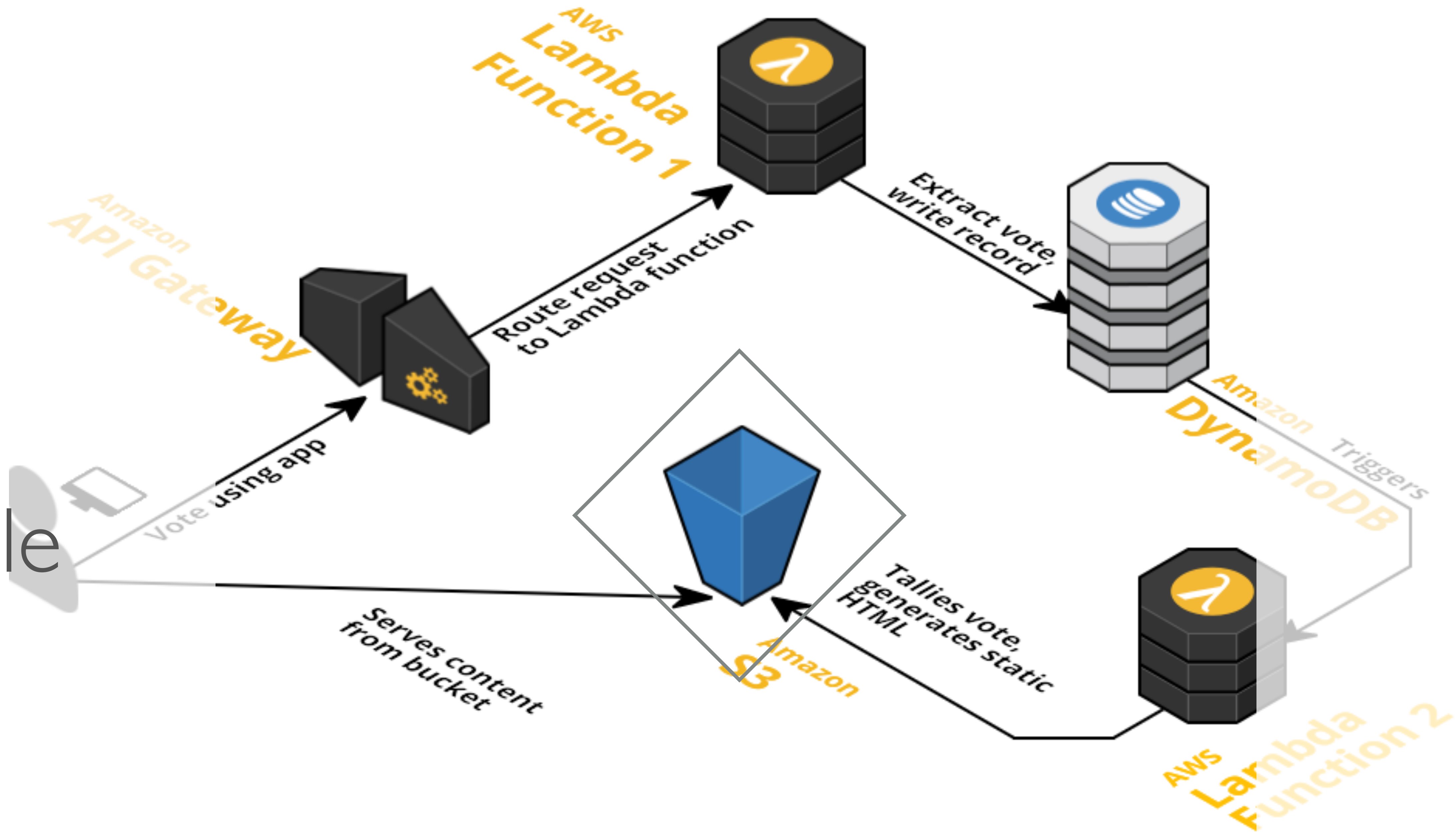
STEP 3: CONFIGURE API GATEWAY

- The endpoint provided by the Lambda blueprint is broken
- Delete the given endpoint and create one from scratch
- Create Body Mapping Template to transform x-www-form-url-encoded request
- Can add security layer with Incognito User Pools and logging to CloudWatch

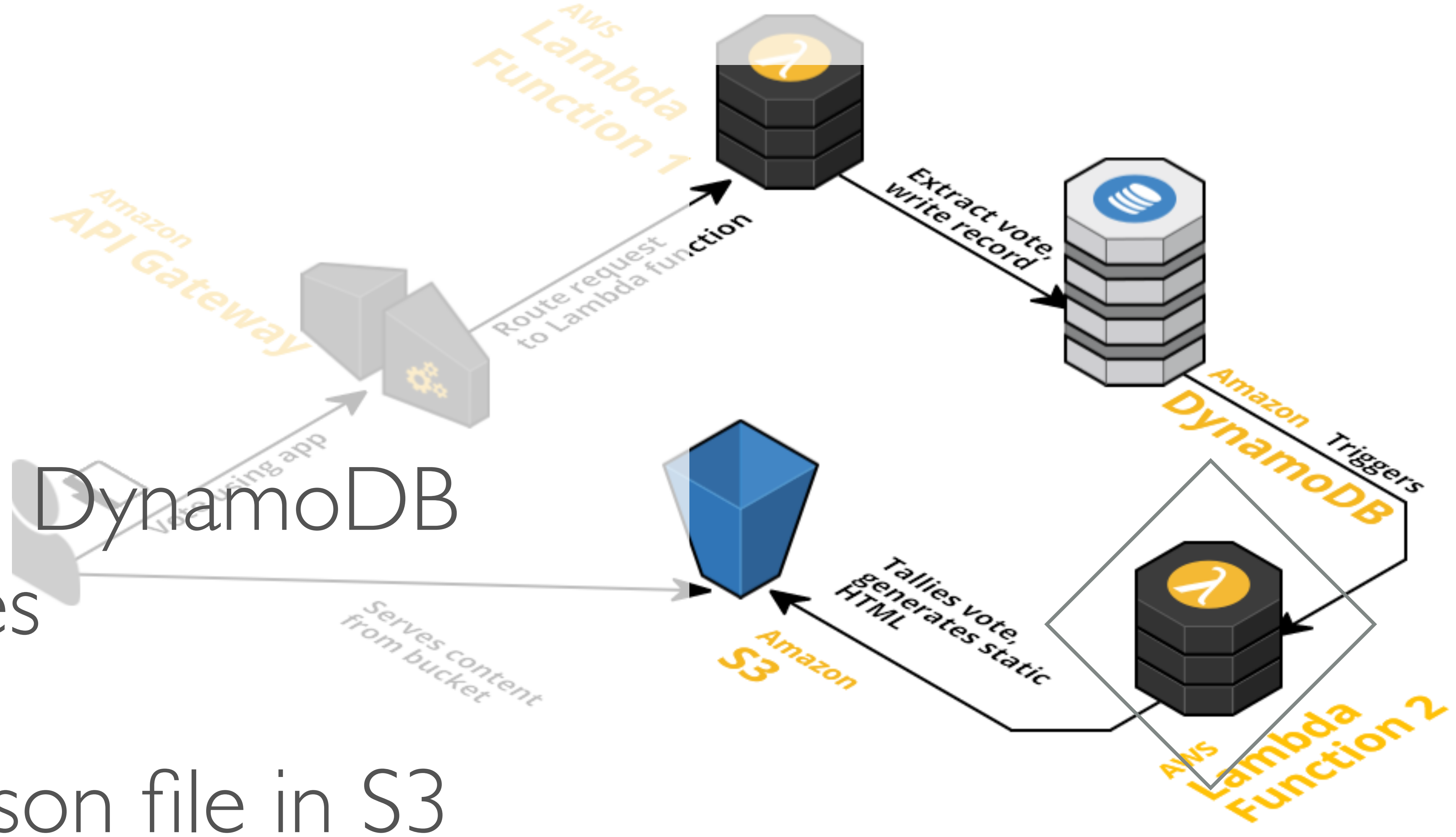


STEP 4: SET UP S3 BUCKET

- Enable Static Website Hosting
- Set bucket policy to enable public access
- Upload html file


AWS Lambda Function 1
Amazon API Gateway
Route request to Lambda function
Extract vote, write record
Amazon DynamoDB
Triggers
Vote using app
Serves content from bucket
Amazon S3
Tallies vote, generates static HTML
AWS Lambda Function 2

STEP 5: LAMBDA FUNCTION 2, THE VOTE TALLY READER



- Trigger from DynamoDB table updates

- Write data.json file in S3

FRAMEWORKS

- Serverless: https://serverless.com/
- apex: http://apex.run/
- chalice (python): https://github.com/awslabs/chalice
- aws-serverless-express: https://github.com/awslabs/aws-serverless-express

THANK YOU!

QUESTIONS?

Dan Rusk
danrusk@thorntech.com