

Serverless Brewing IPAs

WoodWing

Wednesday, March 11, 2020
AWS MeetUp @ Bestseller, Amsterdam



WoodWing

2000

Established

100+

Countries

170+

Employees

1000+

Customers

60+

Business Partners

49,000

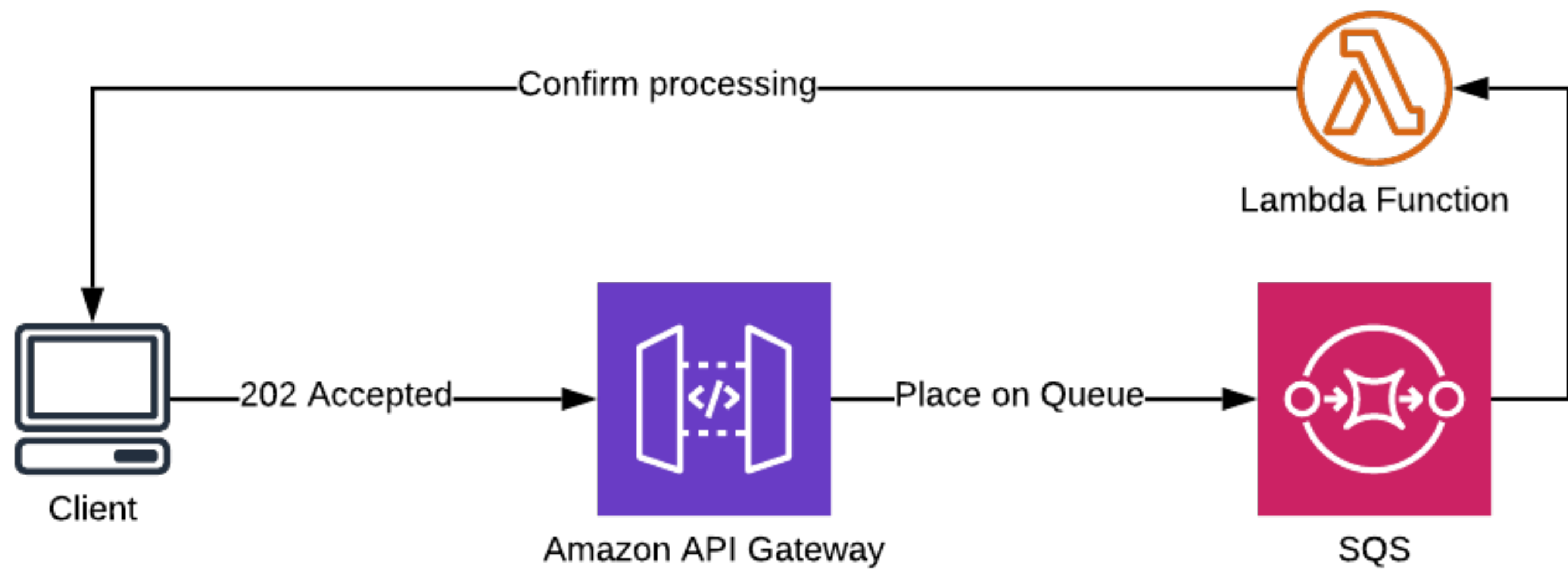
Users



The Architecture

API Integration

- Validate model at API Gateway
- Queue all requests
- Process the request
- Confirm processing



Brewing beer at home



Ingredients

Malt
Yeast
Hop
Brewery love



Hygiene

You need to keep your workplace as clean as possible

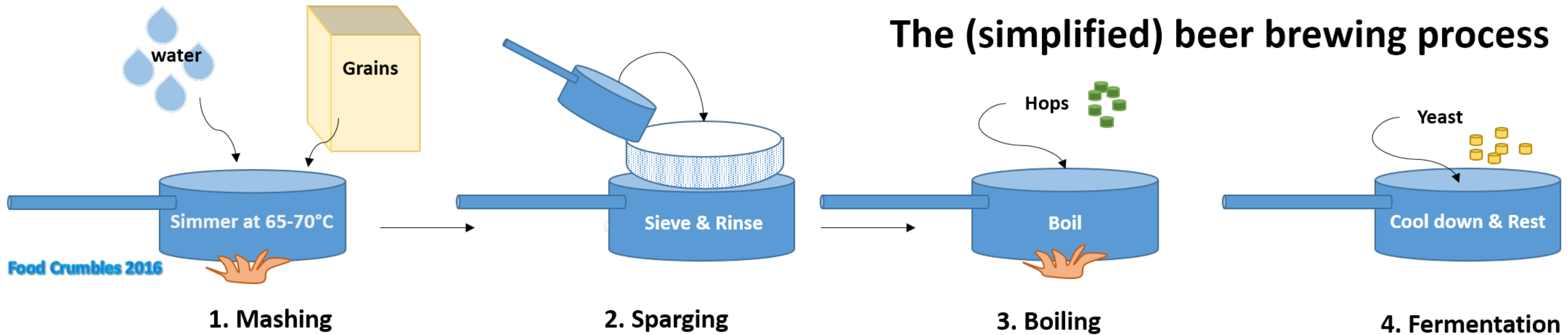


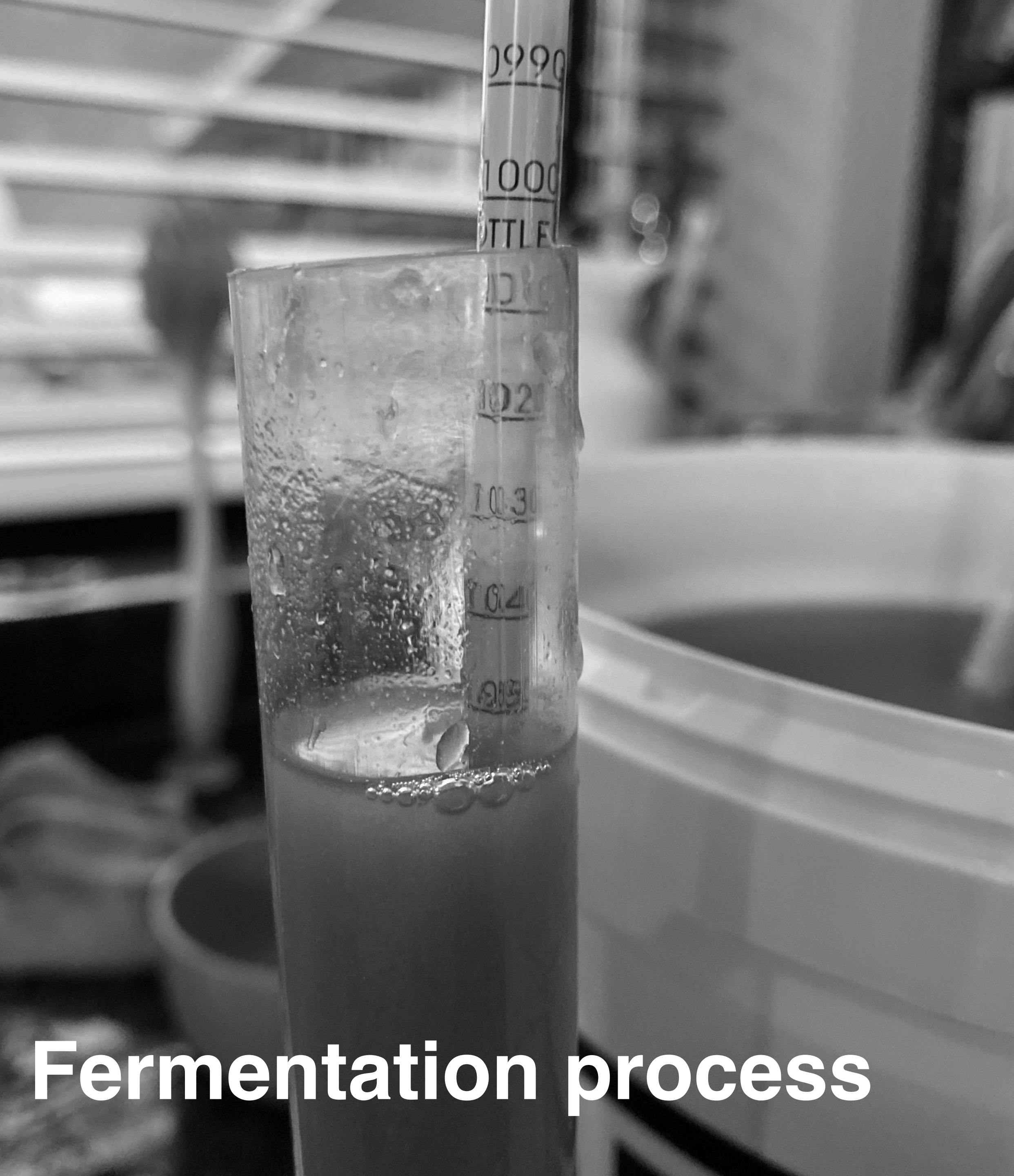
Tools

Brew pan
Pan with rinse water
Sieve
Thermometer
Hydrometer



The process





Fermentation process

iSpindel

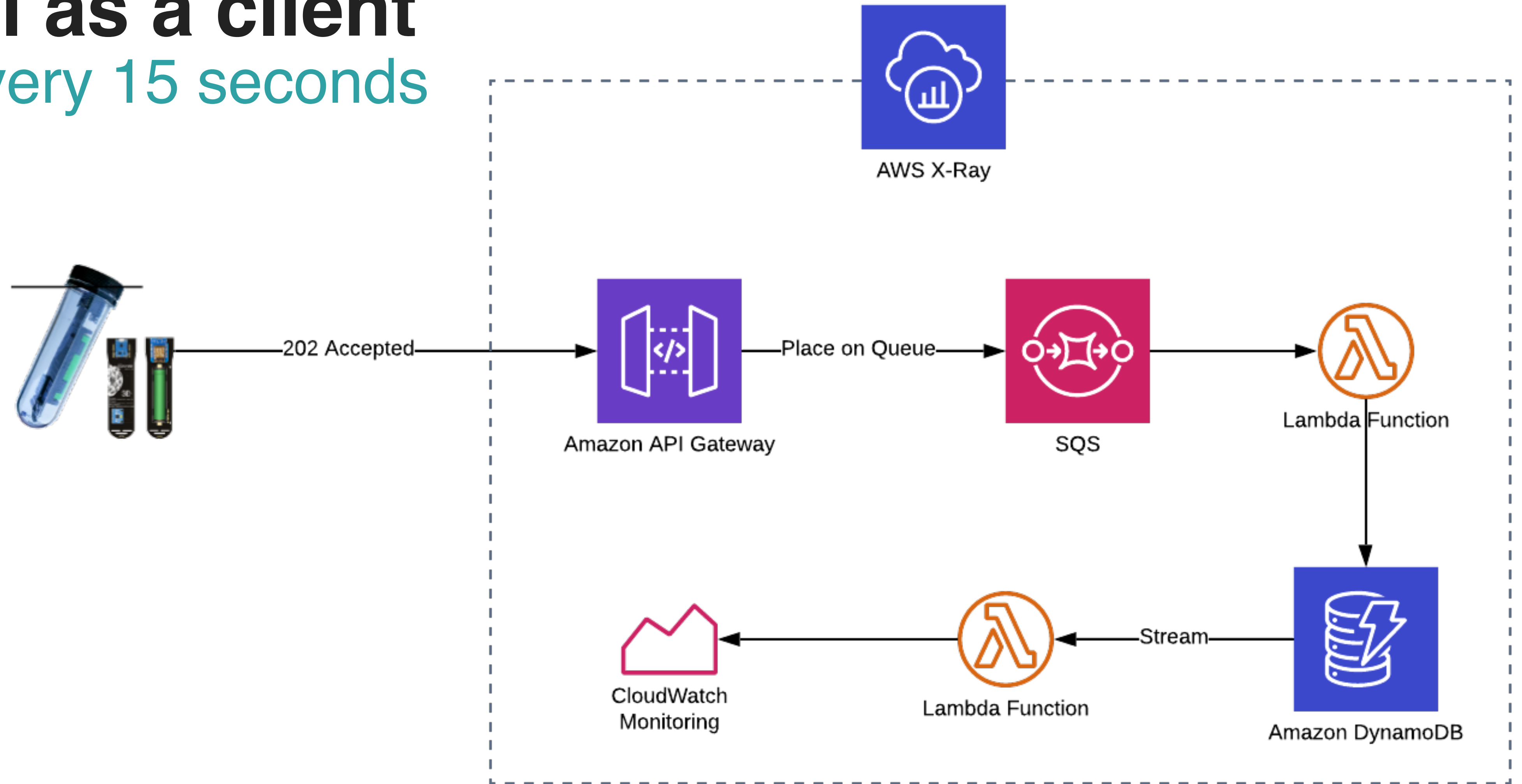
An ESP8266 based free-floating sensor for brewers who want a real-time update about the brewing process via Wifi.

Thanks to Hen Peretz



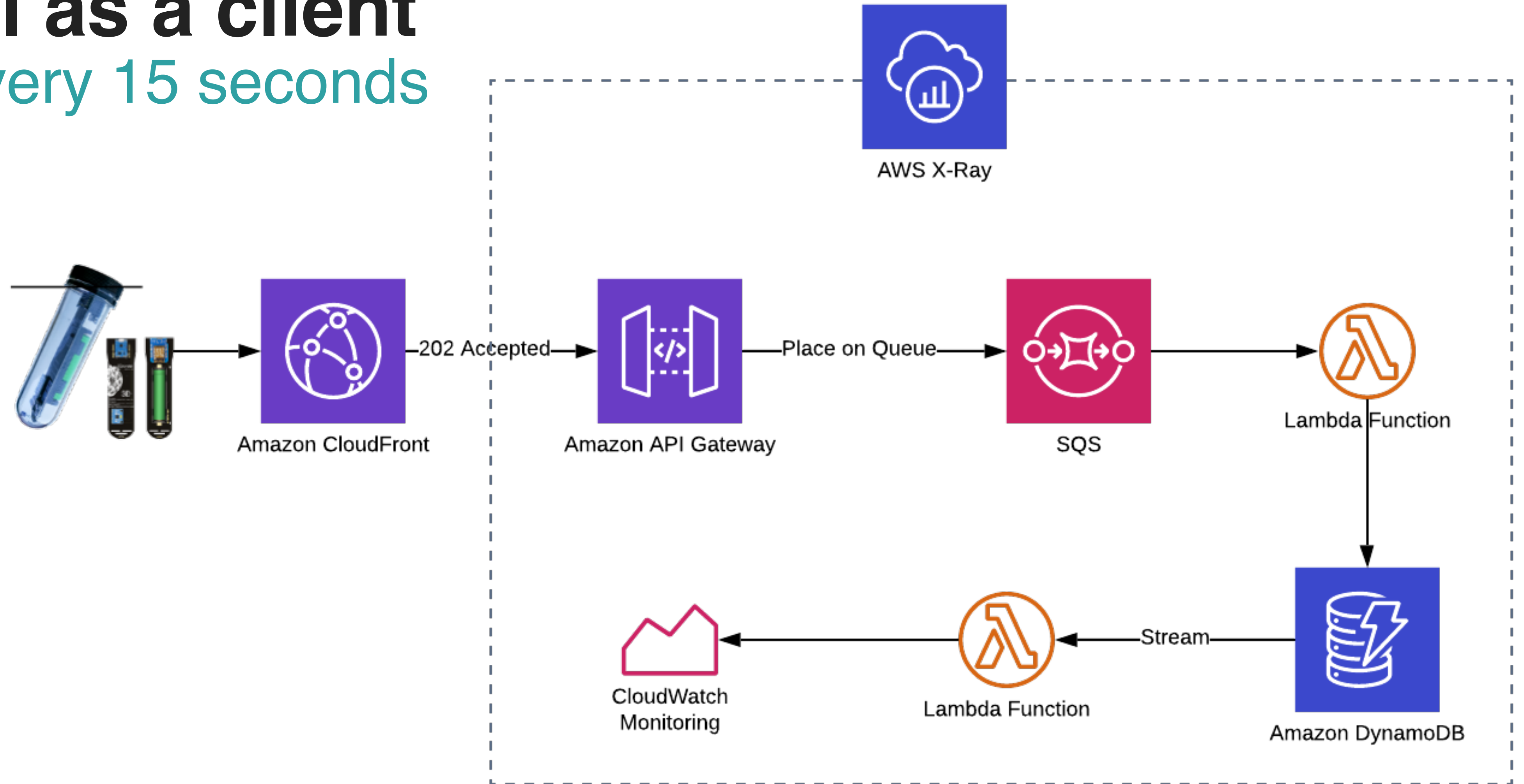
iSpindel as a client

Metrics every 15 seconds



iSpindel as a client

Metrics every 15 seconds



Demo

BrewAPI



Lessons Learned



Split your API definition from your CloudFormation template

See: <https://github.com/OAI/OpenAPI-Specification> (also known as Swagger)

Api:

Type: AWS::Serverless::Api

Properties:

...

DefinitionBody:

Fn::Transform:

Name: AWS::Include

Parameters:

Location: resources/openapi.yaml

\$ sam package

\$ sam deploy



Refs and GetAtt will work in your included API

Only not the shorthand version, that will fail (cfn-lint will not catch it!)

paths:

iSpindel:

post:

x-amazon-apigateway-integration:

credentials: !GetAtt APIGatewayRole.Arn

paths:

iSpindel:

post:

x-amazon-apigateway-integration:

credentials:

Fn::GetAtt: APIGatewayRole.Arn



API Model Validation

iSpindelPayload:

type: object

required: [name, ID, token, angle, temperature, temp_units, battery, gravity, interval]

properties:

name: { **type:** string }

ID: { **type:** integer, **format:** int32 }

token: { **type:** string }

angle: { **type:** number, **format:** float }

temperature: { **type:** number, **format:** float }

temp_units: { **type:** string }

battery: { **type:** number, **format:** float }

gravity: { **type:** number, **format:** float }

interval: { **type:** integer, **format:** int32 }



Usage of Usage plans

Limit the amount of calls made by a client or device:

- **Use the X-API-Key header to send the API key**
- **Use a custom authoriser to return a API key**



SAM Deploy vs AWS Console deploy

When you add `AddDefaultAuthorizerToCorsPreflight` to the `AWS::Serverless::Api` resource it will remove the `AWS_IAM` authorization defined in the OpenAPI?

Auth:

`AddDefaultAuthorizerToCorsPreflight: true`

`ApiKeyRequired: true`

Adding the “DefaultAuthorizer” or removing the “AddDefaultAuthorizerToCorsPreflight” will solve that.

Auth:

`AddDefaultAuthorizerToCorsPreflight: true`

`DefaultAuthorizer: AWS_IAM`

`ApiKeyRequired: true`



Lambda Destinations

Lambda destinations only work with asynchronous invocations and not with synchronous invocations, this means destinations will not work when:

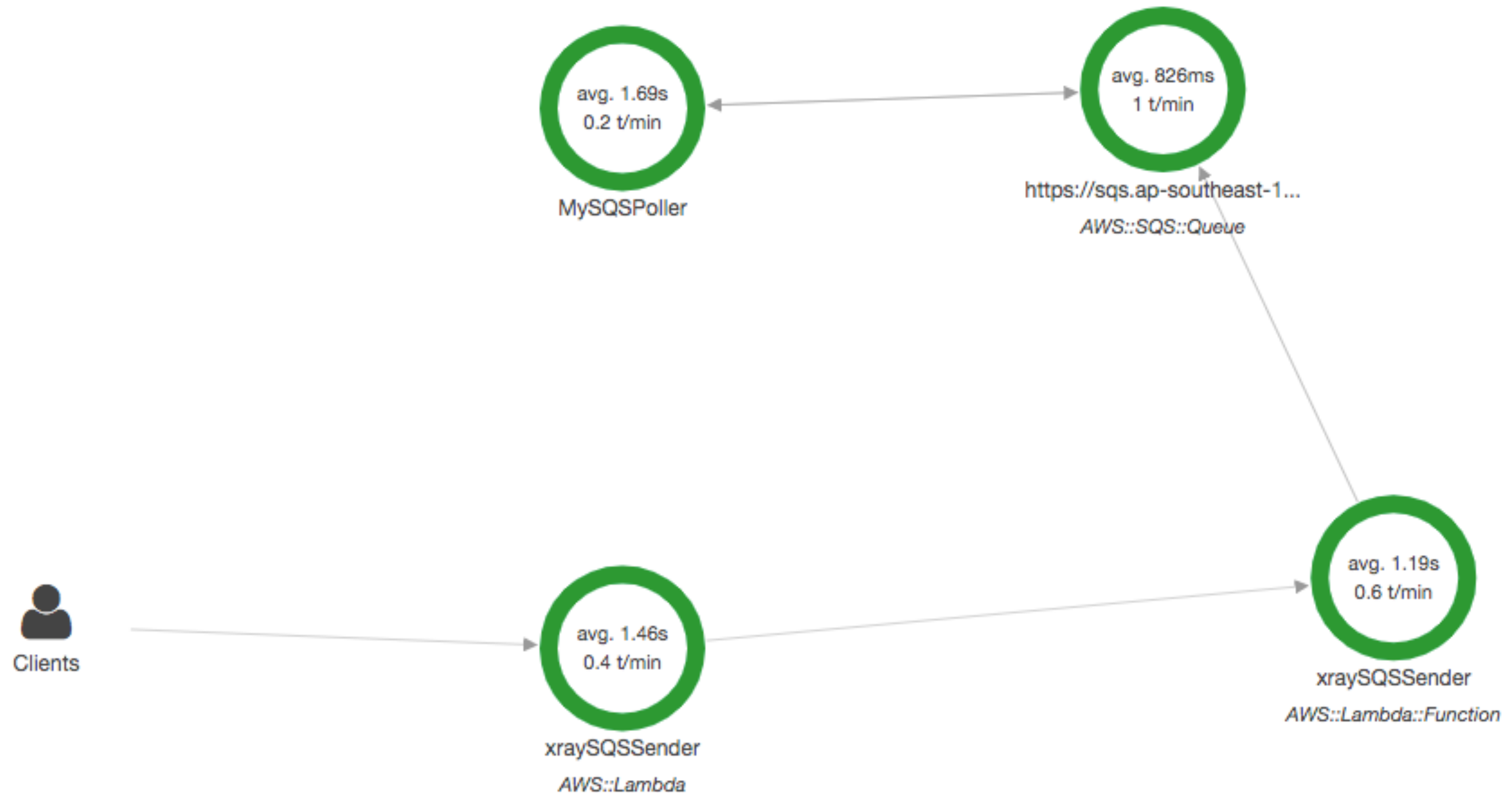
- **When you test a Lambda from the AWS Console**
- **You have a SQS Trigger on your AWS::Serverless::Function resource**

Placing a SNS Topic in between would make it asynchronous again...



Amazon SQS and AWS X-Ray

<https://docs.aws.amazon.com/xray/latest/devguide/xray-services-sqs.html>



Send the AWSTraceHeader from the API Gateway to SQS

x-amazon-apigateway-integration:

uri:

Fn::Sub: arn:aws:apigateway:\${AWS::Region}:sqs:path/\${AWS::AccountId}/\${IncomingMeasurementQueue.QueueName}

httpMethod: POST

type: aws

requestParameters:

integration.request.header.Content-Type: "application/x-www-form-urlencoded"

requestTemplates:

application/json: |

&Action=SendMessage##

&MessageSystemAttribute.1.Name=AWSTraceHeader##

&MessageSystemAttribute.1.Value.DataType=String##

&MessageSystemAttribute.1.Value.StringValue=\$util.urlEncode(\$method.request.header.X-Amzn-Trace-Id)##

&MessageBody=\$util.urlEncode(\$input.json('\$'))##



SQS and Encryption

When you send a message to an encrypted SQS Queue the role needs to have the following IAM policies:

- kms:Decrypt
- kms:GenerateDataKey



Thank you!

Questions?

Are you looking for a new challenge? <https://www.woodwing.com/jobs>

