Building Modern Applications with DynamoDB and Elasticache

Lee Hannigan, DynamoDB Specialist SA

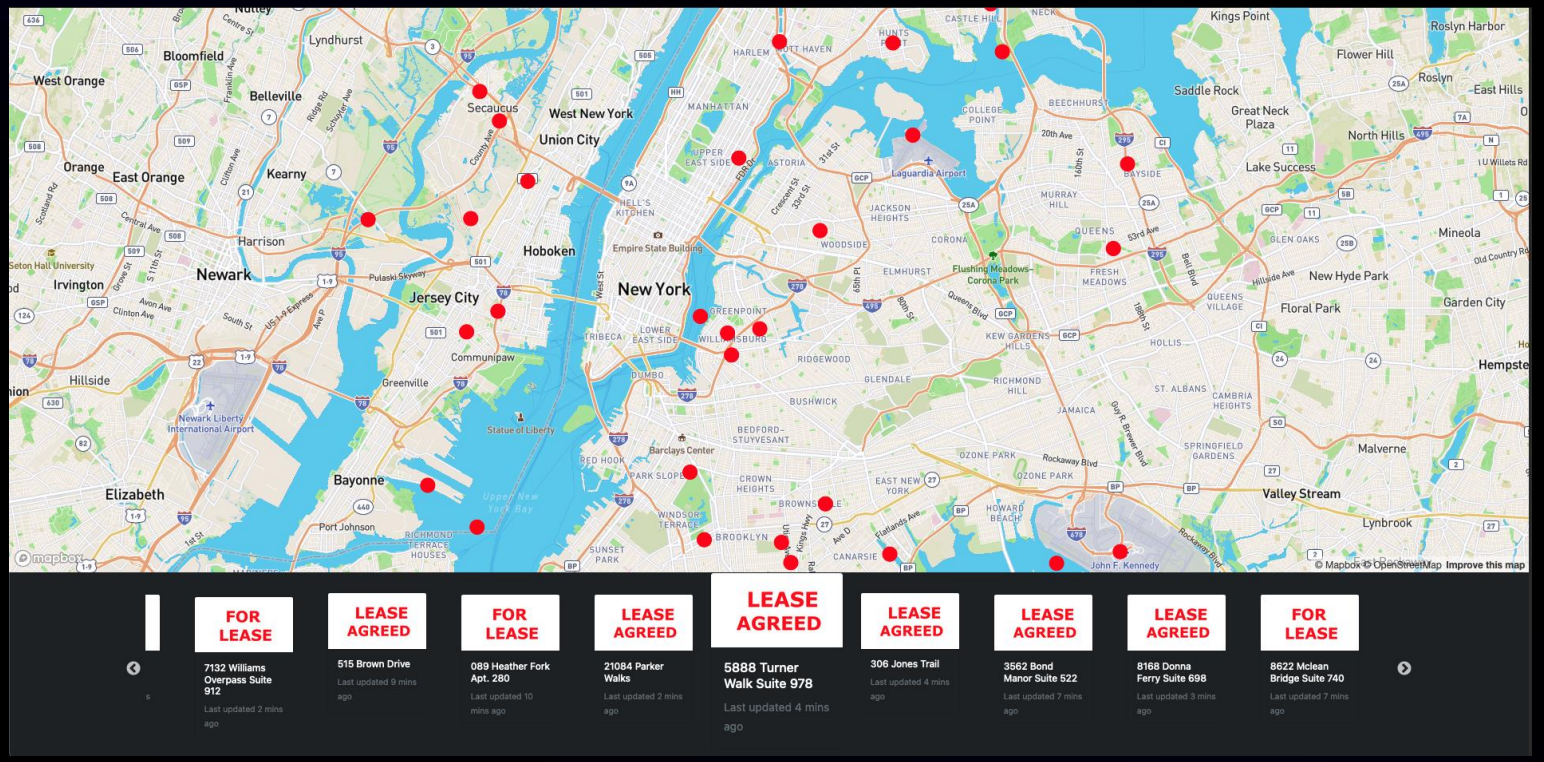
Jeff Duffy, Sr. Technical Advocate

What We Will Cover

- Use Case
- Data/UI/Application Architecture
- Architecture Alternatives
- Demo
- Detailed Implementation Walkthrough
- Q&A

Use Case: Rental Property Search

- A startup needs a system to store and query data for rental properties



- Agents need to add/change/remove property details
- Customers need to search for rental properties near them

Rental Property Search Workload

Cost efficient during periods of low utilization

- Property updates tend to be infrequent

Low latency, high performance reads

- Property searches traffic is extremely bursty

Fast geospatial queries of property data

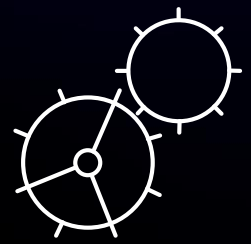
- Property searches are location based

Scales easily

- Coverage area expansion plans, needs to handle going viral

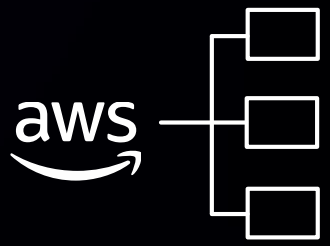
Amazon DynamoDB

Fast and flexible NoSQL database service for any scale



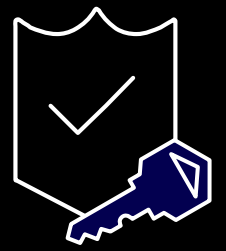
Performance at scale

- Handles millions of requests per second
- Delivers single-digit-millisecond latency
- Automated global replication



No servers to manage

- Maintenance free
- Auto scaling
- On-demand capacity mode
- Change data capture for integration with AWS Lambda



Enterprise ready

- ACID transactions
- Encryption at rest
- Continuous backups (PITR), and on-demand backup and restore

Cost efficient during periods of low utilization

Scales easily

aws

Amazon ElastiCache

Fully-managed In-memory Data Store

Redis & Memcached compatible

Fully compatible with open source Redis and Memcached

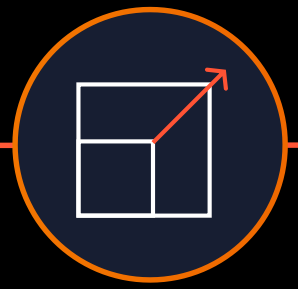
Extreme performance

In-memory data store and cache for microsecond response times

Secure and reliable

Network isolation, encryption at rest/transit, HIPAA, PCI, FedRAMP, multi AZ, global datastore, and automatic failover

Easily scales to massive workloads



Scale reads and writes with sharding and replicas

Low latency, high performance reads

Supports geospatial queries of property data

Scales easily

aws

Property Search Data Architecture

Amazon DynamoDB

System of Record

Amazon ElastiCache for Redis

Geospatial Search

Read Cache

Data Store Alternatives

System of Record

Amazon RDS

Amazon Aurora

Read Cache

DynamoDB Accelerator (DAX)

Amazon MemoryDB for Redis

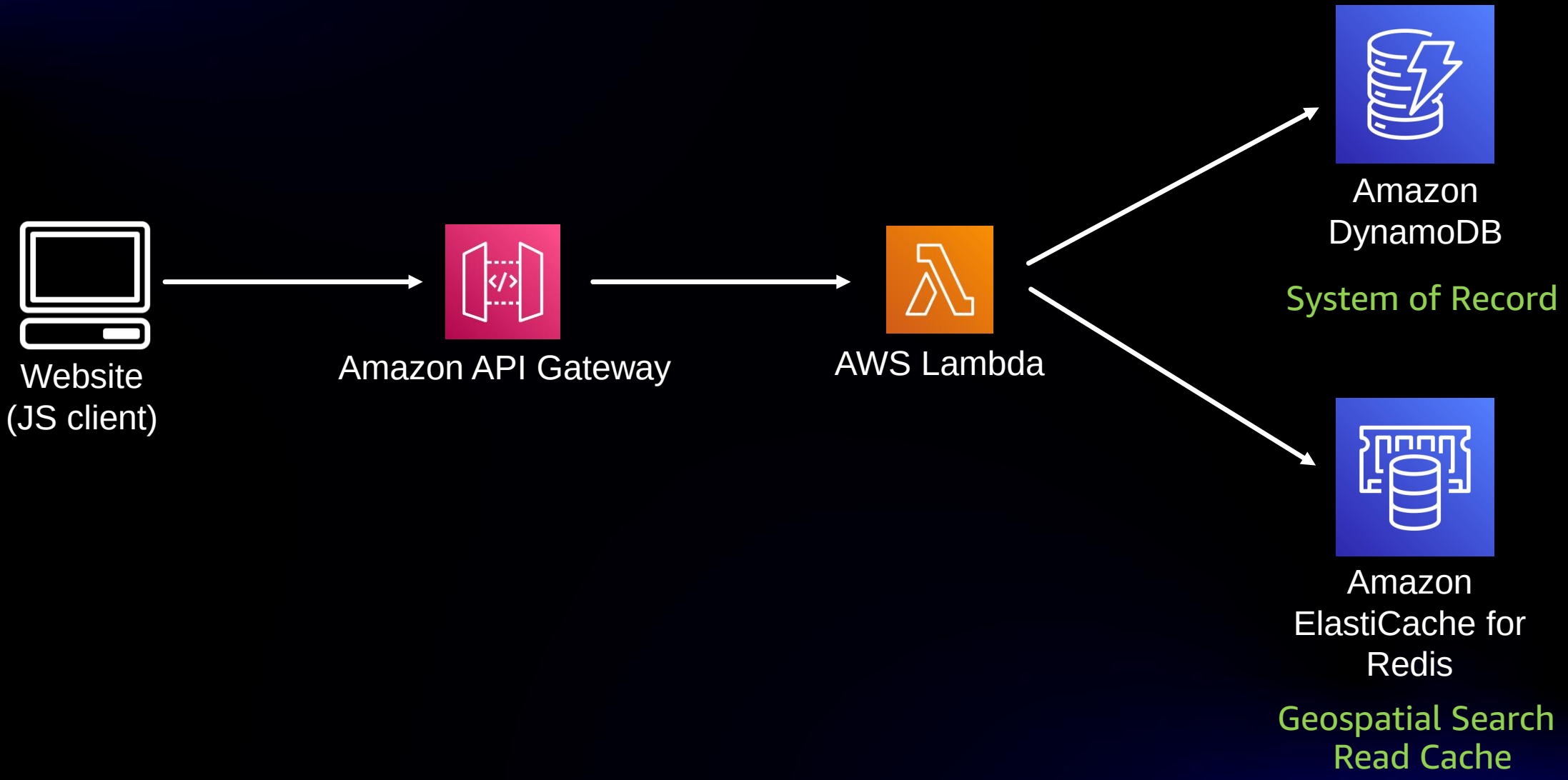
Geospatial Search Engine

Amazon DocumentDB (with MongoDB Compatibility)

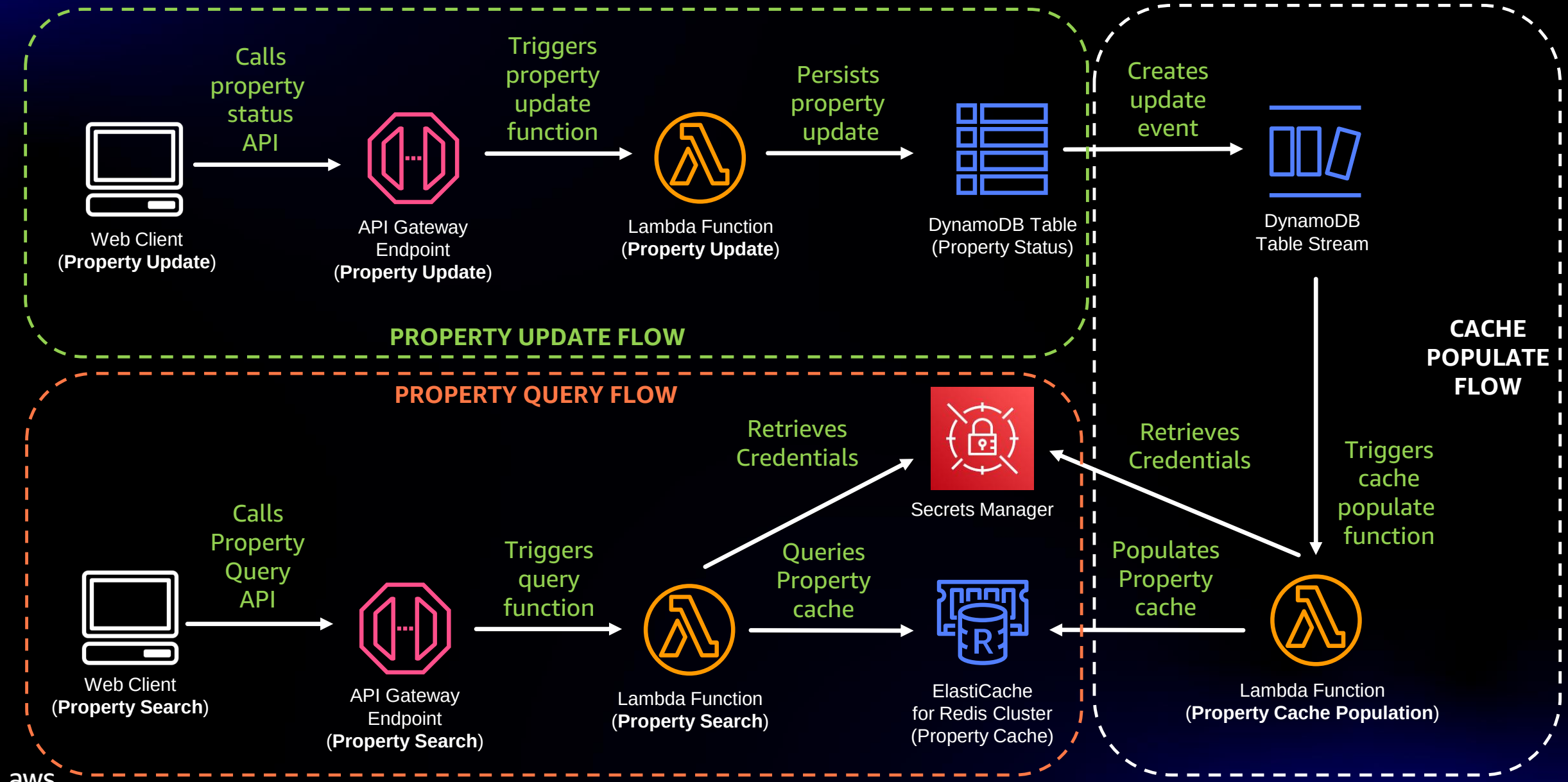
Amazon OpenSearch Service

Amazon DynamoDB Geospatial Library

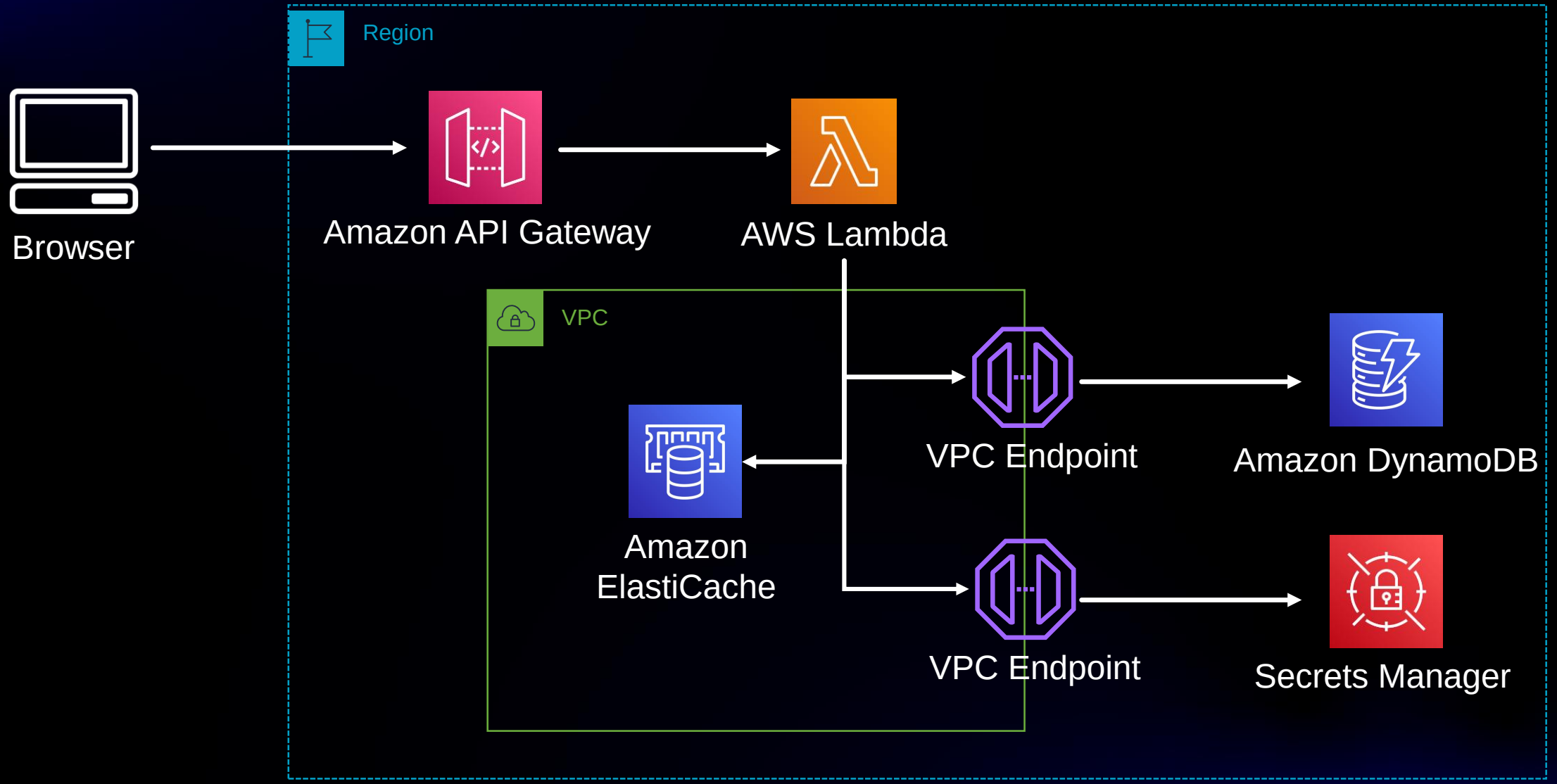
Property Search UI Architecture



Property Search Complete Architecture



Property Search Network Architecture



DynamoDB Table Schema

Primary Key (Must be unique) =
Partition Key + (optional) Sort Key

Partition Key (pk): **agency_id + "#" + agent_id**
Example: “584-64-5919#533-11-2090”

Sort key (sk): **state + "#" + city+ "#" + address**
Example: “New York#New York#20 W 34th S“

DynamoDB Item Schema

All scalar values to accommodate Redis

```
{
    "pk": {
        "S": "072-03-8357#483-98-0366"
    },
    "sk": {
        "S": "Tupelo#Gordonton#74856 Megan Passage"
    },
    ...
    "property_id": {
        "S": "8fc7ec75-7d4e-4062-bcd7-50cf83fde6c1"
    },
    "address": {
        "S": "74856 Megan Passage"
    }
}
```

Redis Keys (Storing Data)

Property Key: pk + "#" + sk

```
property_key = “584-64-5919#533-11-2090# New York#New York#20 W 34th S”
hmset(property_key,
        mapping=property_details)
```

Geospatial Index Key: “properties”

```
geoadd(“properties”,
        (property_details["longitude"], property_details["latitude"],
         property_key)
```

Redis Keys (Retrieving Data)

Geospatial Search

```
geosearch("properties",  longitude="-72.27814", latitude="42.93369",
                radius=10, withcoord=True, withdist=True)
```

```
[['584-64-5919#533-11-2090#Keene#Thomastown#11715 Powers Divide Suite 996',
  0.208, (-72.27813810110092, 42.93369125011038)],
...
]
```

Retrieve Details by Property Key

```
hgetall(property_key)
```

```
{'agent_id': '533-11-2090',
 'agency': 'Torres Ltd',
 ...
 'price': '1561.34', 'updated_at': '1646853523'}
```

aws

GitHub Repo

github.com/aws-samples/**dynamodb-elasticache-geospatial-workshop**

Q&A

Thank you!

Lee Hannigan, DynamoDB Specialist SA

Jeff Duffy, Sr. Technical Advocate