



# AWS Lambda in the Browser

Javascript Interface and Node.js App Without a Single Server



Olivier Klein

Solutions Architect, AWS

17<sup>th</sup> May 2015

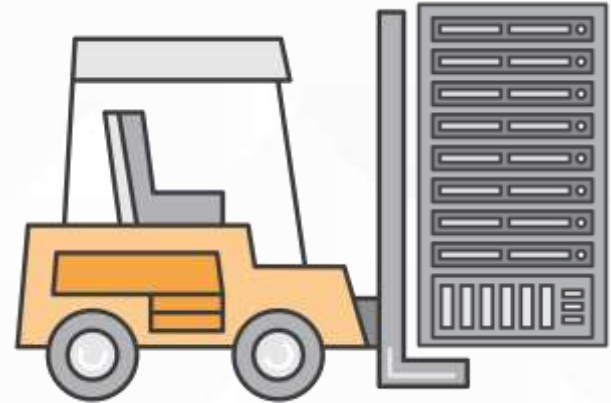
Modern Web Conference Taipei



**What do we expect from modern web applications?**

# Modern Web Applications Expectations

- What do we expect from web applications?
  - Highly Availability
  - Scalability
  - Predictable Responsiveness
  - Fault Tolerance
- How can we achieve this?
  - Multiple datacenter facilities
  - Auto-Scalable databases
  - Auto-Scalable server infrastructure
  - Load-Balancing
  - etc.





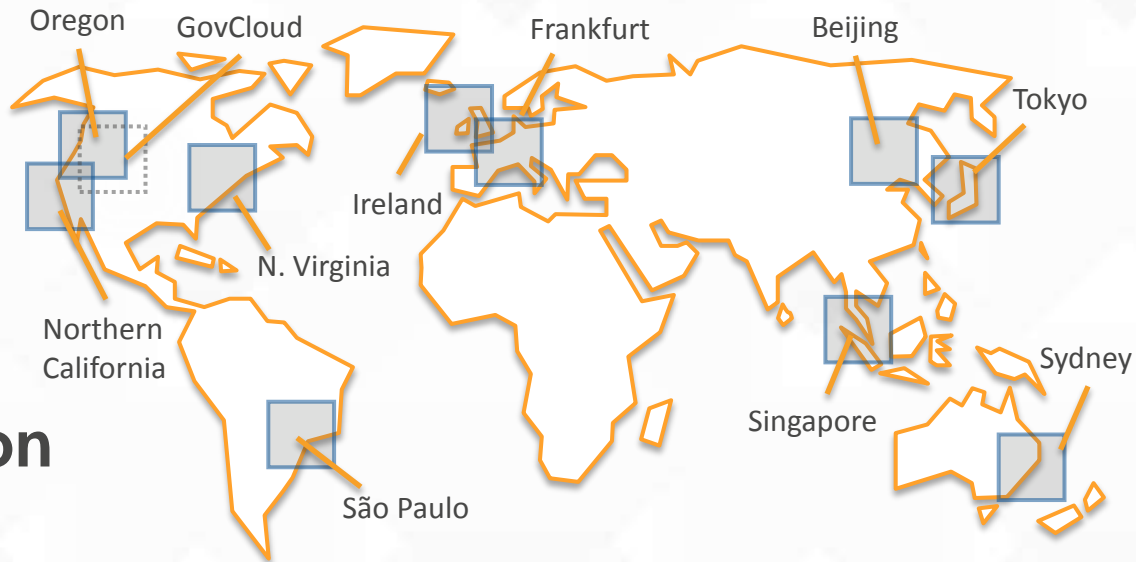
# AWS Global Infrastructure

**11 Regions**

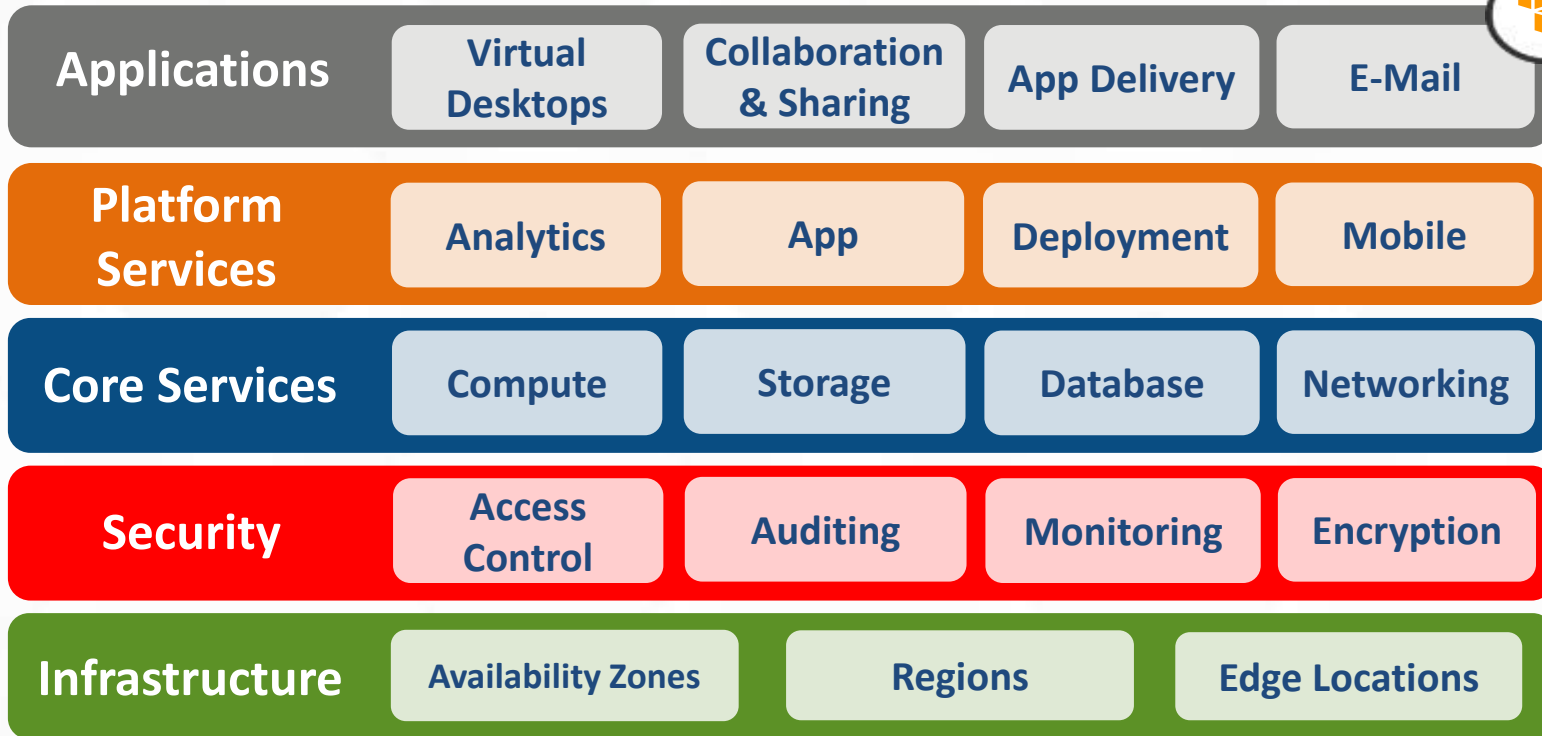
**29 Availability Zones**

**53 Edge Locations**

**Continuous Expansion**



# Web Services



A  
P  
I  
&  
S  
D  
K  
S



**So what's this AWS Lambda?**

# AWS Lambda

“Run your code in the cloud in response to events and scale without any servers to manage”

1

Zero Administration



Focus on business logic, not infrastructure. Upload your code; AWS Lambda handles everything else

2

Auto Scaling  
(Never under or over provision)



Lambda scales the infrastructure as needed to match the event rate and pay as you go

3

Bring Your Own Code



Write your code in JavaScript just like with a Node.js app



# Event-Driven Compute

- Stateless, request-driven code called *Lambda* functions
- Invoke Lambda functions straight from JavaScript or even your Mobile app
- Triggered functions by events (state transitions) in other services:
  - New object in your object store
  - New item in your NoSQL database
  - New message in your notification service



# AWS JavaScript SDK

```
<script src="js/aws-sdk-2.1.26.min.js"></script>

// Construct a service object
var lambda = new AWS.Lambda();

// Making requests
lambda.invoke({FunctionName: 'myfunc'}, function(err, data) {
  if (err) console.log(err, err.stack);
  else {
    var result = JSON.parse(data.Payload);
    // Process result
  }
});
```

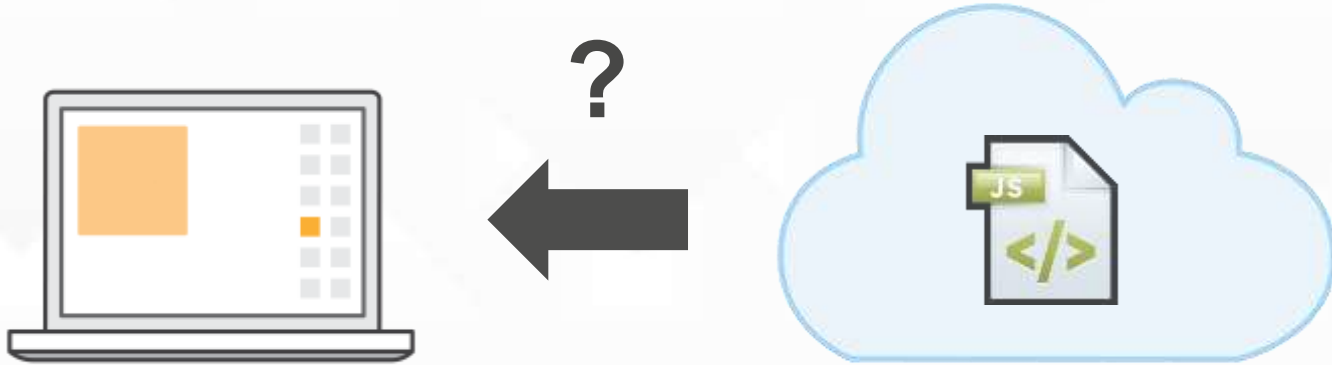


JavaScript SDK



**Great, but you said no servers?**

# How to get the JavaScript to the browser?

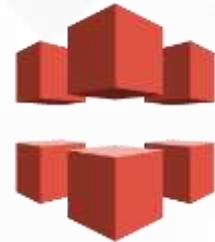


# Amazon S3 & Amazon CloudFront

- Amazon S3
  - Highly available object storage
  - Designed for 99.999999999% durability
  - Offers HTTP / HTTPS endpoint to objects
- Amazon CloudFront
  - Content Delivery Network with 53 edge locations across the world
  - Caches content on edge location for low latency



Amazon S3



Amazon  
CloudFront



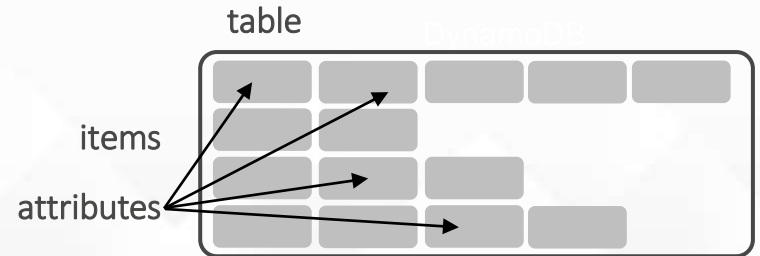


**But Lambda is stateless, where to store data?**

# Amazon DynamoDB

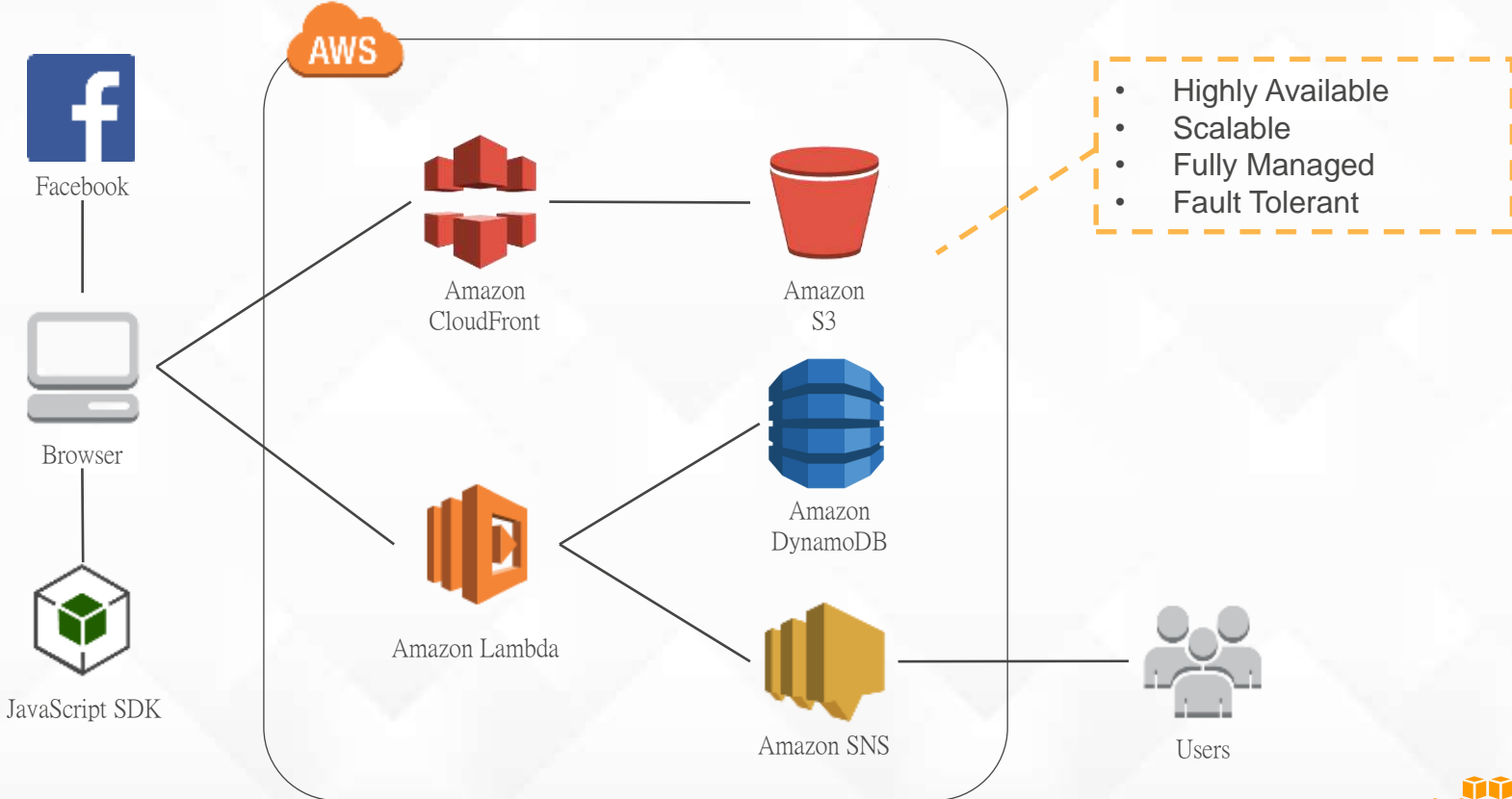
## Non-Relational Managed NoSQL Database Service

- Schemaless Data Model
- Consistent low latency performance
- Predictable provisioned throughput
- Seamless Scalability
- No storage limits
- High durability and availability
- Replicated across 3 Facilities
- Fully Managed and Low Cost





# Example Architecture



- Highly Available
- Scalable
- Fully Managed
- Fault Tolerant



**Demo Time!**



---

# Thank you!

Olivier Klein  
olivierk@amazon.com