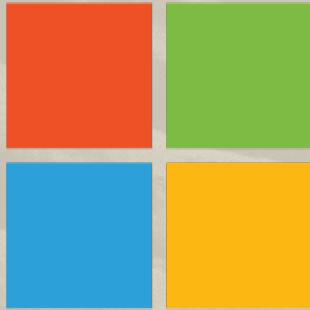


# Everybody Into The Pool!

Node.js For .Net Developers



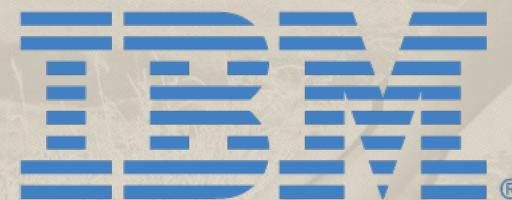




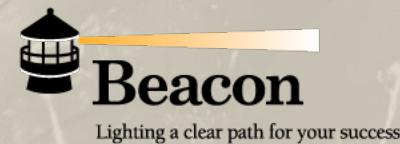
# Microsoft



**inrule**  
TECHNOLOGY®



**RENAISSANCE**  
LEARNING™

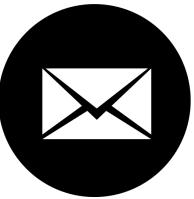


**COMPLY365**®

**Concurrency**

**tegile**

# The Pool's Lifeguard ... Matt



[msoucoup@codemilltech.com](mailto:msoucoup@codemilltech.com)



[codemilltech.com](http://codemilltech.com)



[@codemillmatt](https://twitter.com/codemillmatt)



[github.com/codemillmatt](https://github.com/codemillmatt)



<http://bit.ly/MSN-MOBILE>

# Everybody Into the Pool!

Overview

- Dipping Our Toes In
- Learning To Swim
- Into the Blue With Azure and Off the Ledge with Edge

# Everybody Into the Pool!

## Part 1: Dipping Our Toes In

- What Is Node.js?
- When Should It Be Used?
- When Shouldn't It Be Used?
- Why Would a .Net Developer Want to Learn Node.js?

# What Is Node.js?

Node.js is a platform built on Chrome's JavaScript runtime for building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

“JavaScript ... for fast, scalable network  
applications”

```
var name = 'Matt';  
name = 123;
```

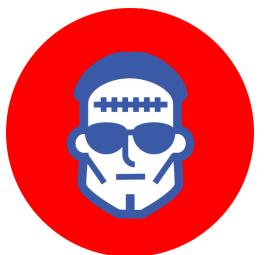
# “JavaScript ... for fast, scalable network applications”

- **Fast**
  - JavaScript runs in Chrome’s V8 engine
  - V8 and Node.js written in C
- **Scalable**
  - No multi-threaded overhead
- **Network**
  - http, dns, net, querystring, url



# “Non-blocking I/O model”

Blocking Example:  
The ATM



# “Non-blocking I/O model”

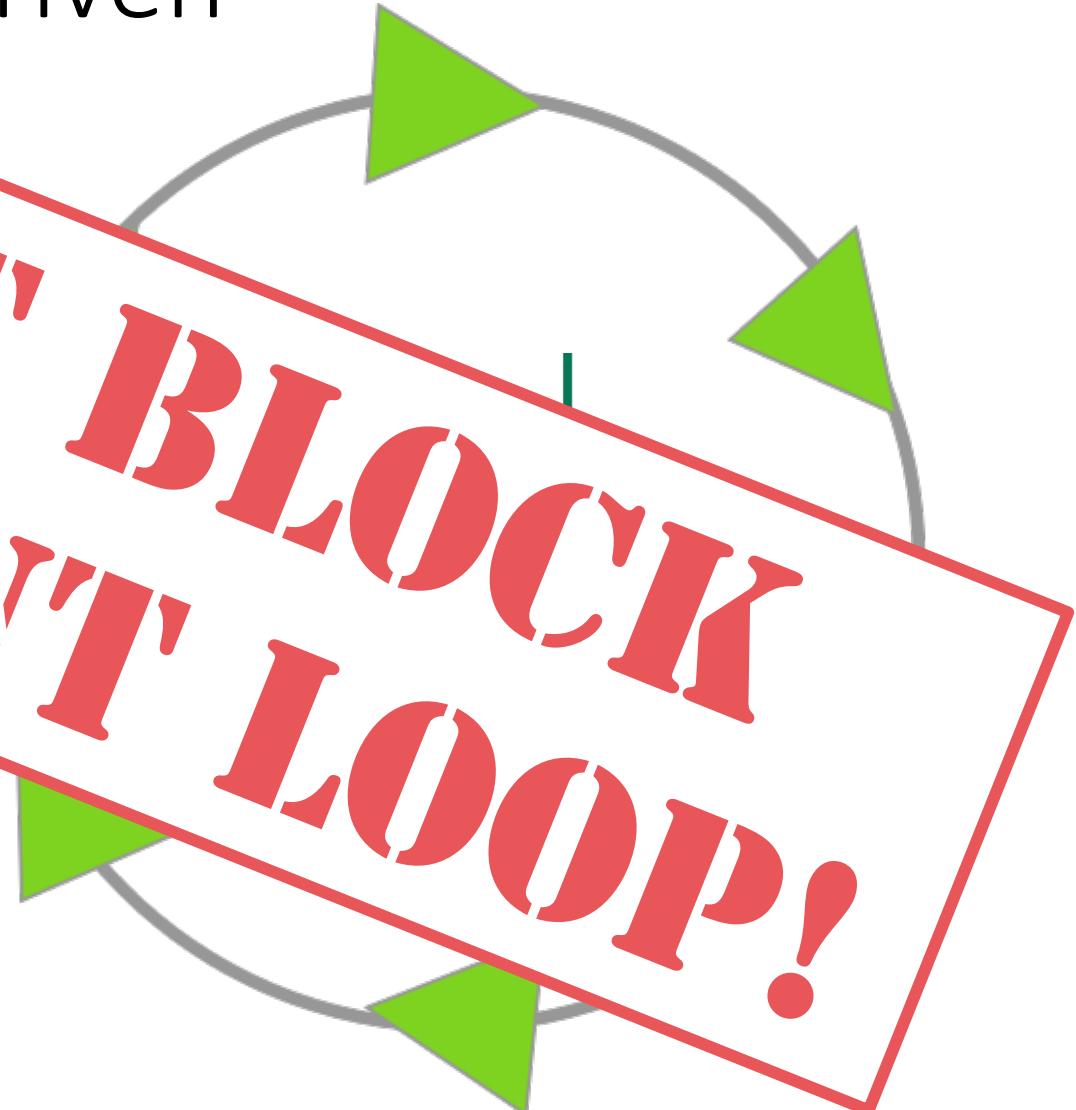
Non-Blocking Example:  
Coffee Shop



- Order coffee
- Pick up coffee “back”
- Whole coffee shop is an “event loop”

“Event-Driven”

**DO NOT BLOCK  
THE EVENT LOOP!**



# Node.js in a Nutshell...

- Fast, scalable, network apps
  - Chrome's V8
  - Modules focused on scalable networking
- Non-blocking
  - Designed to be asynchronous
- Event-driven
  - Events, call-backs and event loops

# Use Cases

## What is it good for?

- Real-time web
  - Chat apps, stock tickers
- The glue
  - REST servers, message queue
- Streaming data
  - File uploading
- Serve large amounts of clients
  - Ad server

## What isn't it good for?

- CPU intensive work
- Non-I/O
- Just because it's cool

# Why Node.js For .Net Developers?

- Teaches us how to handle async
- Javascript is lingua-franca
- Microsoft & keep existing investments
- Great community developing node modules
- Give clients more choice



# Everybody Into the Pool!

## Part 2: Learning To Swim

- Installing Node.js
- npm
- Node.js Tools For Visual Studio
- Jump Into the Water With a Demo

# Installing Node.js

- <http://nodejs.org>
- Download installer package



# Installing Node.js

- <http://chocolatey.org>
- Machine Package Manager
  - NuGet for applications!



```
C:\> choco install nodejs.install
```

# npm Package Management

- Package manager
- ~ NuGet for JavaScript
- 170,000+ packages





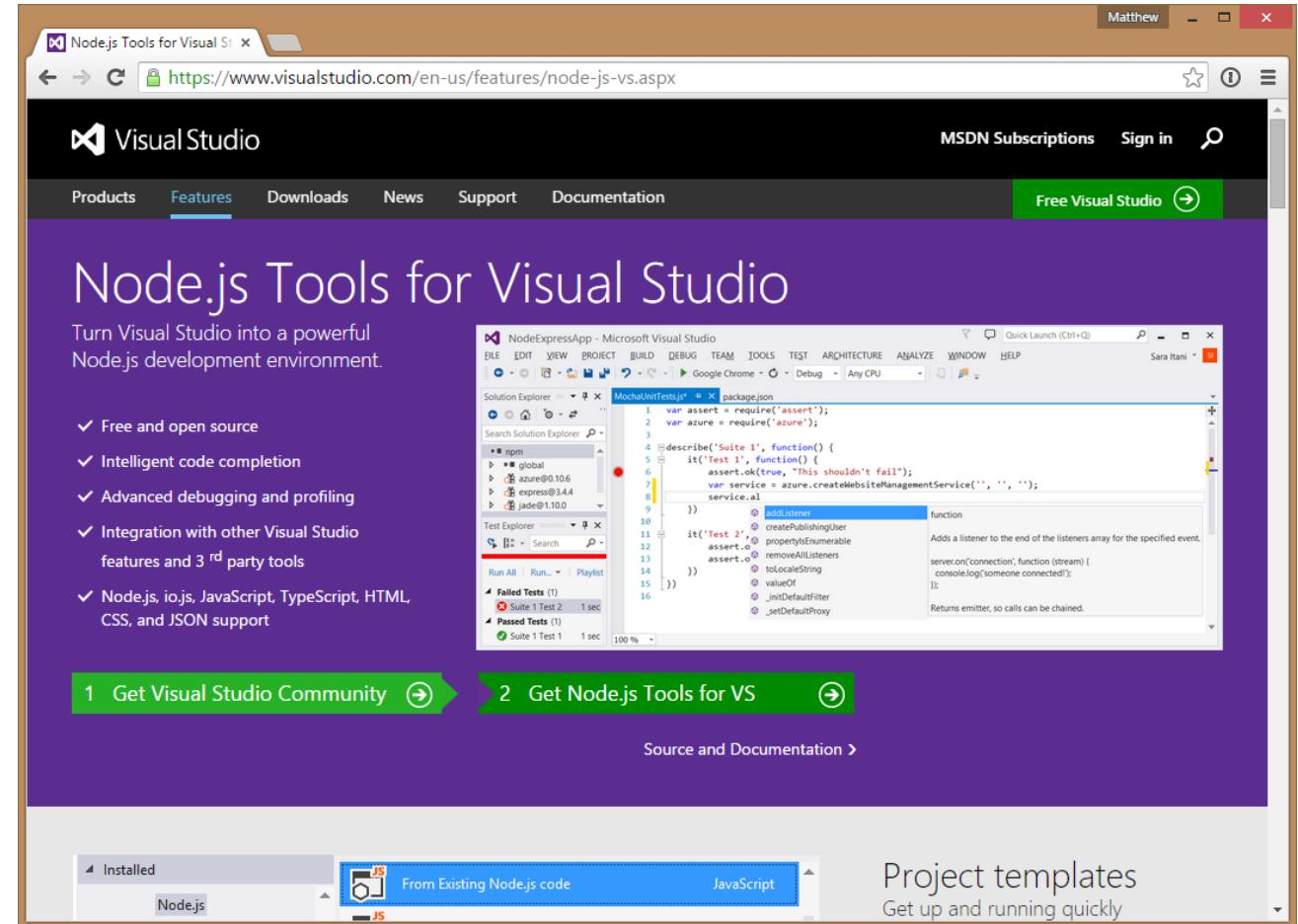
# Handy Modules For the .Net Dev

- express
  - Middleware
- restify
  - REST services
- edge
  - Call .Net functions from Node.js and Node.js functions from .Net
- node-sqlserver
  - Microsoft's driver to access SQL Server from Node.js
- azure
  - Access to many of Azure's services

# Node.js Tools for Visual Studio

- Project templates
- Intellisense
- Debugging
- Profiling
- REPL
- Unit testing
- npm integration

Visual Studio 2012 +



[http://visualstudio.com/en-us/features/node-js-vs.aspx](https://www.visualstudio.com/en-us/features/node-js-vs.aspx)

# Look mom, I can swim!



# Everybody Into the Pool!

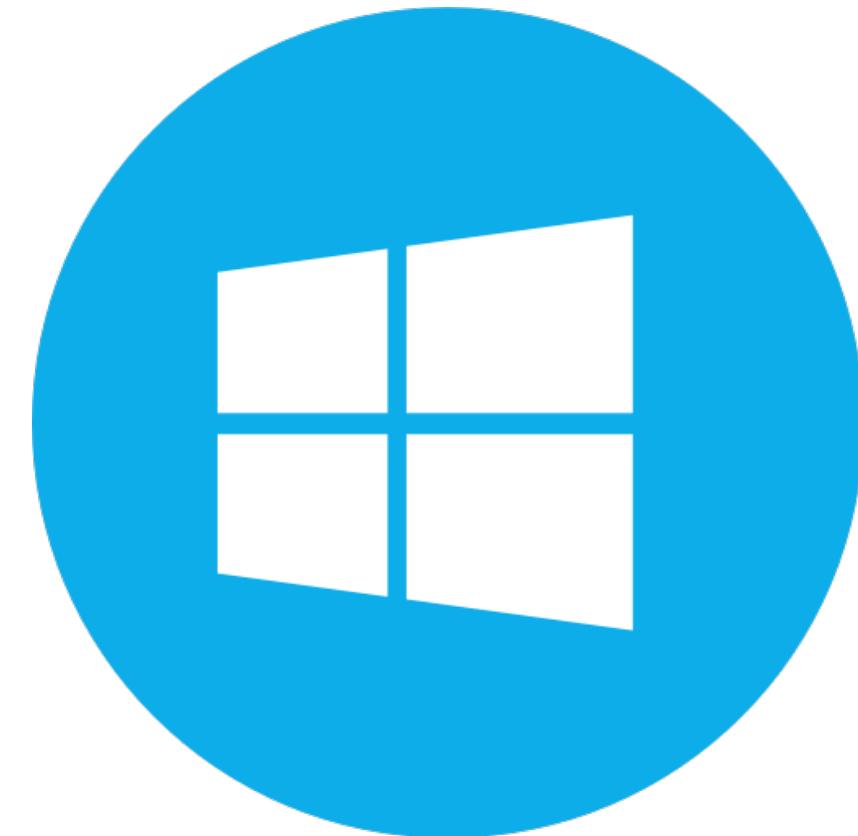
Part 3: Into the Blue With Azure and Off the Ledge with Edge

- Azure's Node.js Functionality
- Deploying Web Apps Into Azure
- Edge.js – Invoking .Net From Node.js

# What Can Azure Do With Node.js?

## Host Node.js Applications

- Web Apps/Sites
  - Azure Mobile Services
  - Cloud Services
  - Virtual Machines
- 
- Including npm modules!
  - Specific Azure modules!

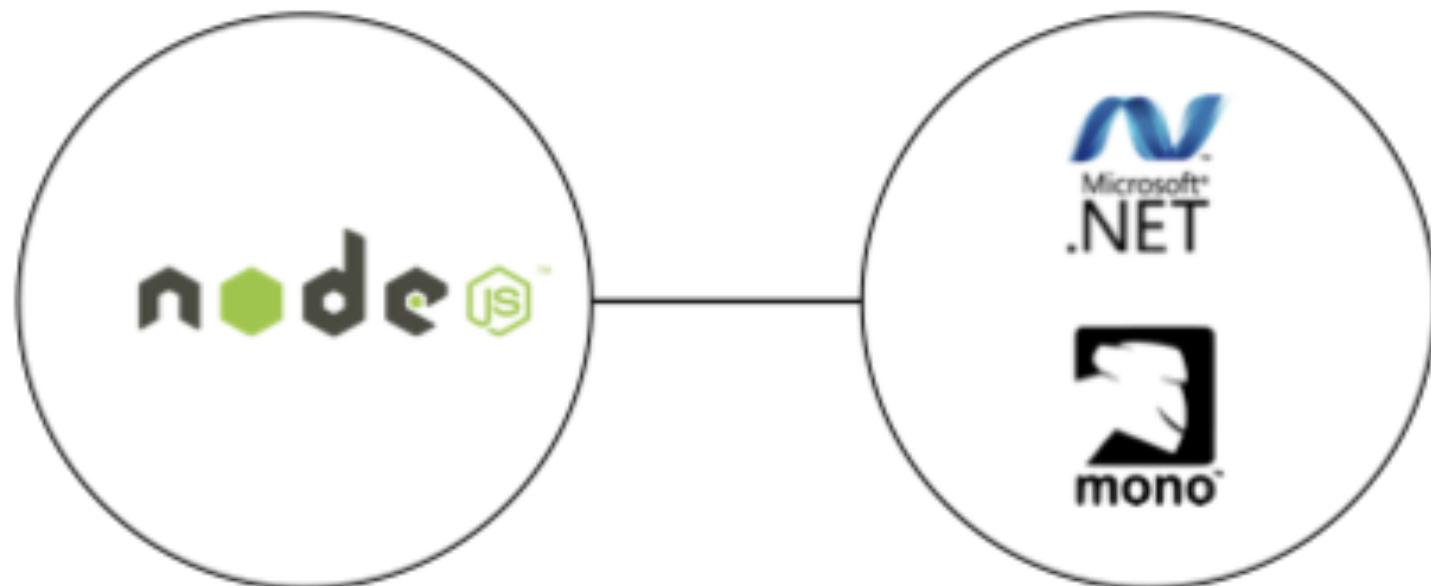


# Where Is Matt Swimming ... In Azure!!



# Edge.js

- Allows us to invoke .Net code from Node.js (and vice versa)
- Inline C#
- C# files
- Compiled DLLs



## node.exe

CLR thread  
CLR thread  
CLR thread  
CLR thread

Func<object, Task<object>>



v8 thread (singleton)

```
function (payload, callback) {
  callback(error, result);
}
```



# Dropping Off The Ledge ... With Edge.js



# Where To Go From Here?

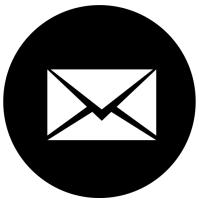
- Dive In!
- Node.js can be the glue
- Node.js can be microservice
- Node.js can invoke .Net



# Toweling Off

- Node.js – fast, non-blocking, event-driven
- Full support in Visual Studio
- Azure
- Edge and .Net code reuse
- (Don't need to dunk head into node right away – can transition slowly with Edge & calling web services, etc).

# Reach Out!



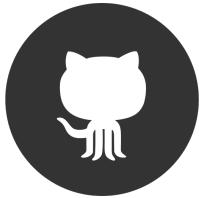
[msoucoup@codemilltech.com](mailto:msoucoup@codemilltech.com)



[codemilltech.com](http://codemilltech.com)



[@codemillmatt](https://twitter.com/codemillmatt)



[github.com/codemillmatt](https://github.com/codemillmatt)