2014 Global Windows Azure BOOTCAMP

Welcome to the #GWAB 2014!

ASP.NET WebAPI & Azure WebSites

Mario Szpuszta
Principal Program Manager
Global Partner Engagement, Technical Evangelism & Development (TED)
Microsoft Corp. HQ

Lokale Sponsoren:

Compuware APM

Microsoft

www.IT-Visions.at
Dr. Holger Schwichtenberg

Using ASP.NET Web API with VS2013

Windows Azure

ASP.NET Web API

Build HTTP-based services

Ships with Visual Studio 2013

Available as NuGet packages (for .NET >= 4.5)

Details: www.asp.net/vnext

Open Source: http://aspnetwebstack.codeplex.com

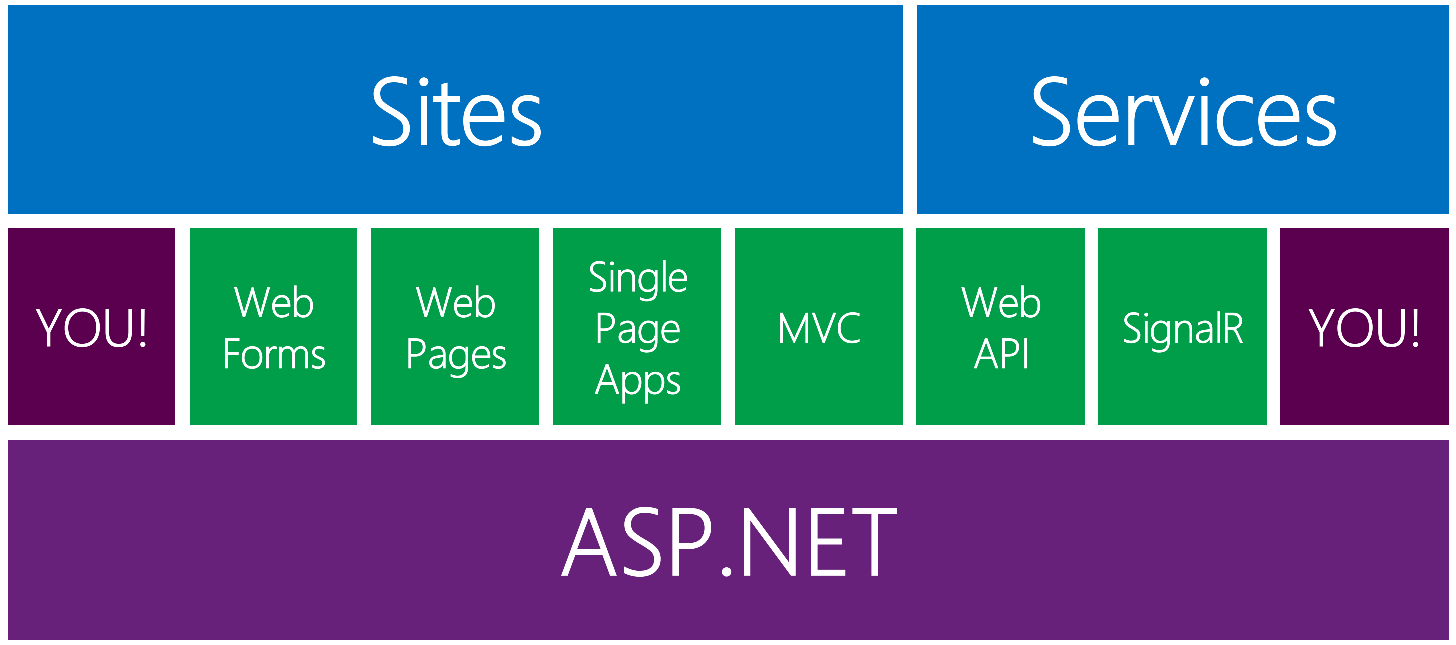
Specific Visual Studio 2013 Enhancements

„One ASP.NET"

Unified Scaffolding

Identity Integration

Mixing ASP.NET frameworks



Windows Azure

Note: Continuous Extensions!!

NuGet!!

Extension Manager!!

(non-intrusive extensions☺)

Web Essentials
A Visual Studio extension

Windows Azure

ASP.NET Web API 2

- Attribute routing
- OWIN integration
- Easier Unit-testing (IHttpActionResult)
- Portable Web API clients (HttpClient)
- Odata-integration ($select, $expand, $batch)
- Request-batching
- CORS (cross origin resource sharing)
- Oauth 2.0 integration

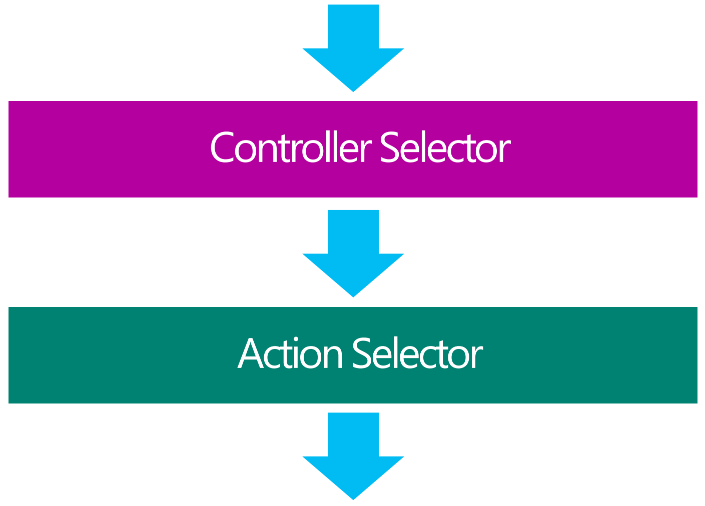
Attribute Routing in Web API 2

Web API Routing so far

```
config.Routes.MapHttpRoute(
    name: “DefaultRoute",
    routeTemplate: "api/{controller}/{id} ",
    defaults: new {
            controller = “home”,
            action = “Get” }
);
```

Attribute-based routing

```
config.MapHttpAttributeRoutes();
```



```
public IEnumerable<TodoItem> GetTodos()
{ … }
```

```
[HttpGet("api/todolists/{id}/todos")]
public IEnumerable<TodoItem> GetTodos(int id)
{ … }
```

Attribute routing

Optional values

```
[HttpGet(“Demographics/{zipcode?}")]
public Demographics Get(int? zipcode) { … }
```

Default values

```
[HttpGet("Demographics/{zipcode=98052}")]
public Demographics Get(int zipcode) { … }
```

Inline constraints

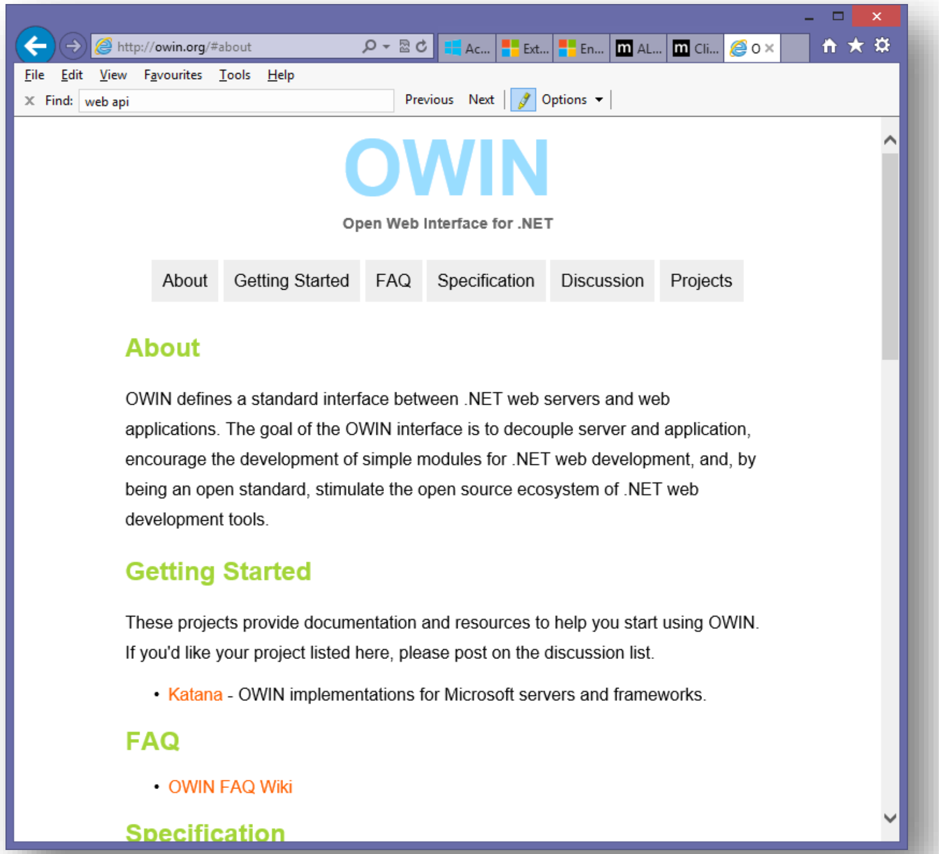
```
[HttpGet("people/{id:int}")]
public Person Get(int id) { … }

[HttpGet("people/{name:alpha}")]
public Person Get(string name) { … }
```

Contribution by
Tim McCall
http://attributerouting.net

OWIN integration

- OWIN = Open Web Interface for .NET
  - (http://owin.org)
  - Common interface to decouples web apps from web servers
  - Inspired by the likes of node.js, Rack, WSGI
- Middleware pipeline
  - Brings "runtimes" into "the host"
- ASP.NET 4.5.1 integrates with OWIN
  - Ex. run authenticating middleware during the Authenticate ASP.NET pipeline stage
- Run your Web APIs on any OWIN compliant host


http://owin.org/#about
File Edit View Favourites Tools Help
Find: web api
Previous Next Options
OWIN
Open Web Interface for .NET
About Getting Started FAQ Specification Discussion Projects

About

OWIN defines a standard interface between .NET web servers and web applications. The goal of the OWIN interface is to decouple server and application, encourage the development of simple modules for .NET web development, and, by being an open standard, stimulate the open source ecosystem of .NET web development tools.

Getting Started

These projects provide documentation and resources to help you start using OWIN. If you'd like your project listed here, please post on the discussion list.

- Katana - OWIN implementations for Microsoft servers and frameworks.

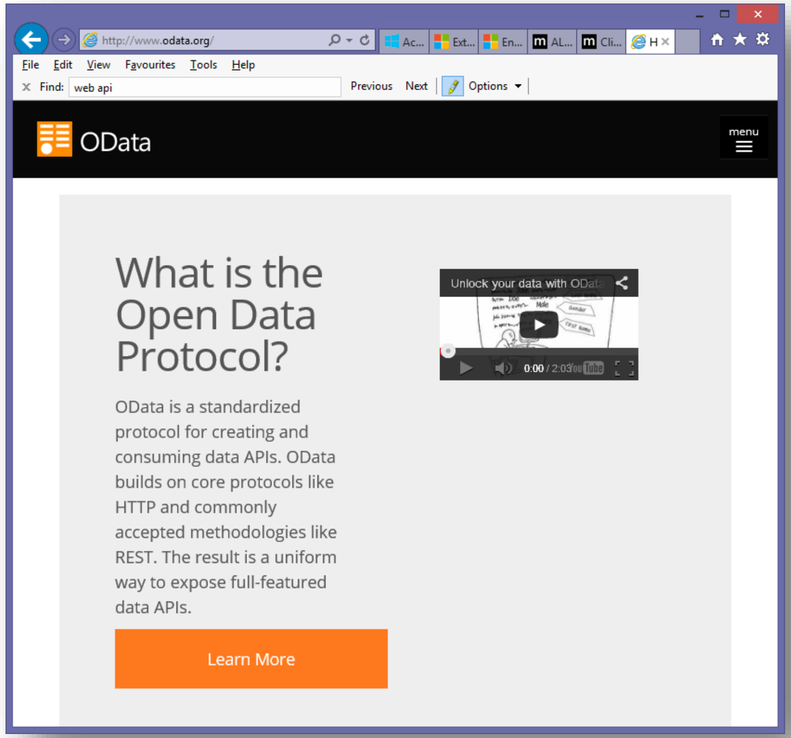
FAQ

- OWIN FAQ Wiki

Specification

Odata Filtering

- Odata implementation via Web API
  - www.odata.org
  - Possible since Visual Studio 2012 Update 2
- Web API 2 – Odata part of framework
  - Based on "ODataLib"
- Allows basic Odata operations for non-Odata Web APIs
  - $select, $expand, $batch, $filter



What is the Open Data Protocol?

OData is a standardized protocol for creating and consuming data APIs. OData builds on core protocols like HTTP and commonly accepted methodologies like REST. The result is a uniform way to expose full-featured data APIs.

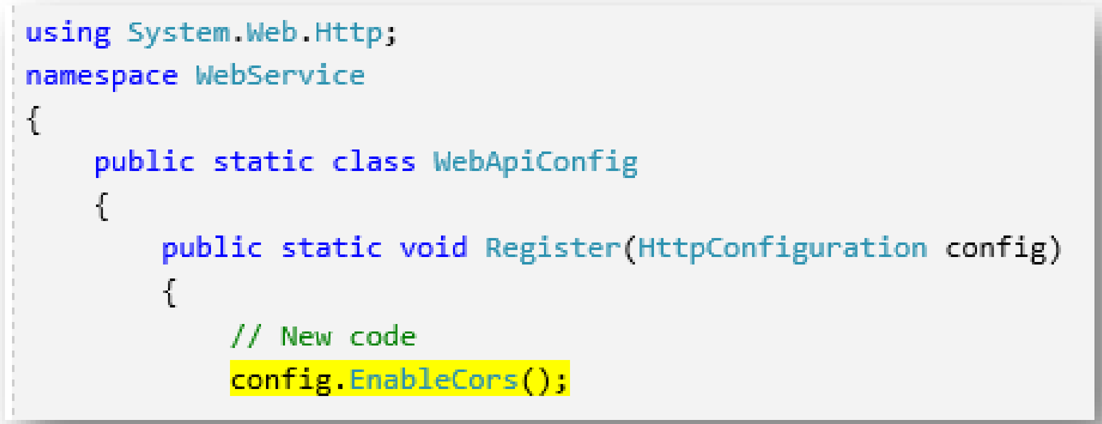
Cross Origin Resource Sharing (CORS)

Cross-origin resource sharing (CORS) is a mechanism that allows JavaScript on a web page to make XMLHttpRequests to another domain, not the domain the JavaScript originated from.[1] Such "cross-domain" requests would otherwise be forbidden by web browsers, per the same origin security policy. CORS defines a way in which the browser and the server can interact to determine whether or not to allow the cross-origin request.[2] It is more powerful than only allowing same-origin requests, but it is more secure than simply allowing all such cross-origin requests.

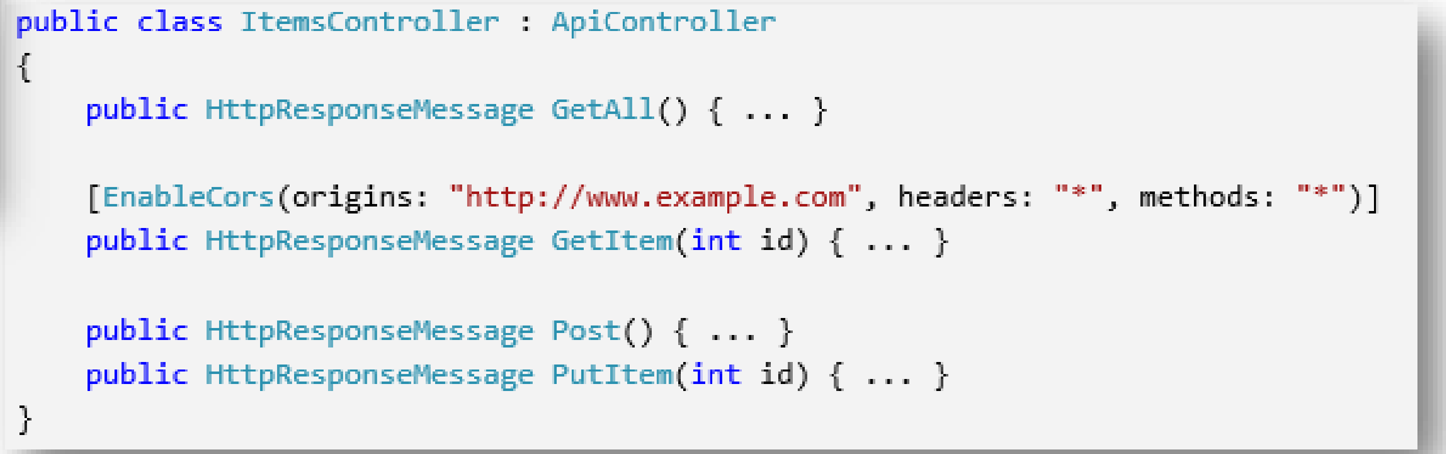
ASP.NET Web API 2 and CORS

Install NuGet Package

- Microsoft.AspNet.WebApi.Cors
- Note: Install-Package with „-Pre“ Option



```
using System.Web.Http;
namespace WebService
{
    public static class WebApiConfig
    {
        public static void Register(HttpConfiguration config)
        {
            // New code
            config.EnableCors();
```



```
public class ItemsController : ApiController
{
    public HttpResponseMessage GetAll() { ... }

    [EnableCors(origins: "http://www.example.com", headers: "*", methods: "*")]
    public HttpResponseMessage GetItem(int id) { ... }

    public HttpResponseMessage Post() { ... }
    public HttpResponseMessage PutItem(int id) { ... }
}
```

Windows Azure

OAuth 2.0 Bearer token support

```
public class Startup
{
    public void ConfigureAuth(IAppBuilder app)
    {
        // Enable the application to use OAuth 2.0 bearer tokens to authenticate users
        app.UseOAuthBearerAuthentication(new OAuthBearerAuthenticationOptions());
    }
}
```

OAuth 2.0 authorization server support

- On-Premise Options (examples)
  - Visual Studio “Single Page Application Template”
    - Contains example template code
  - Windows Server 2012 R2 ADFS
    - http://www.cloudidentity.com/blog/2013/07/30/securing-a-web-api-with-windows-server-2012-r2-adfs-and-katana/
  - Thinktecture Identity Server
    - http://thinktecture.github.io/Thinktecture.IdentityServer.v2/
- Cloud Options (examples)
  - Windows Azure Active Directory
  - Windows Azure Active Directory Access Control Service

web sites

Windows Azure

Windows Azure Web Sites

Shared web hosting in the cloud

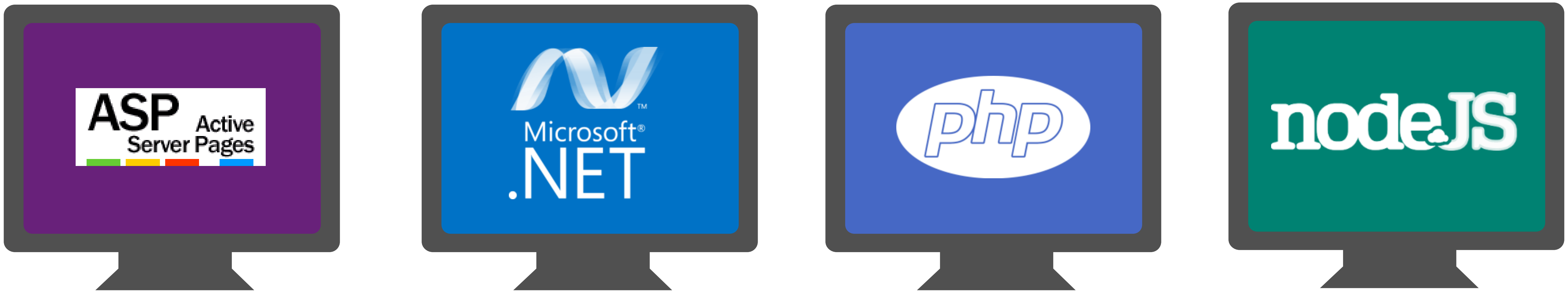
Simple web app or web service

With a Azure SQL DB or MySQL database

Easy deployment (FTP, web deploy, TFS, *GIT*)

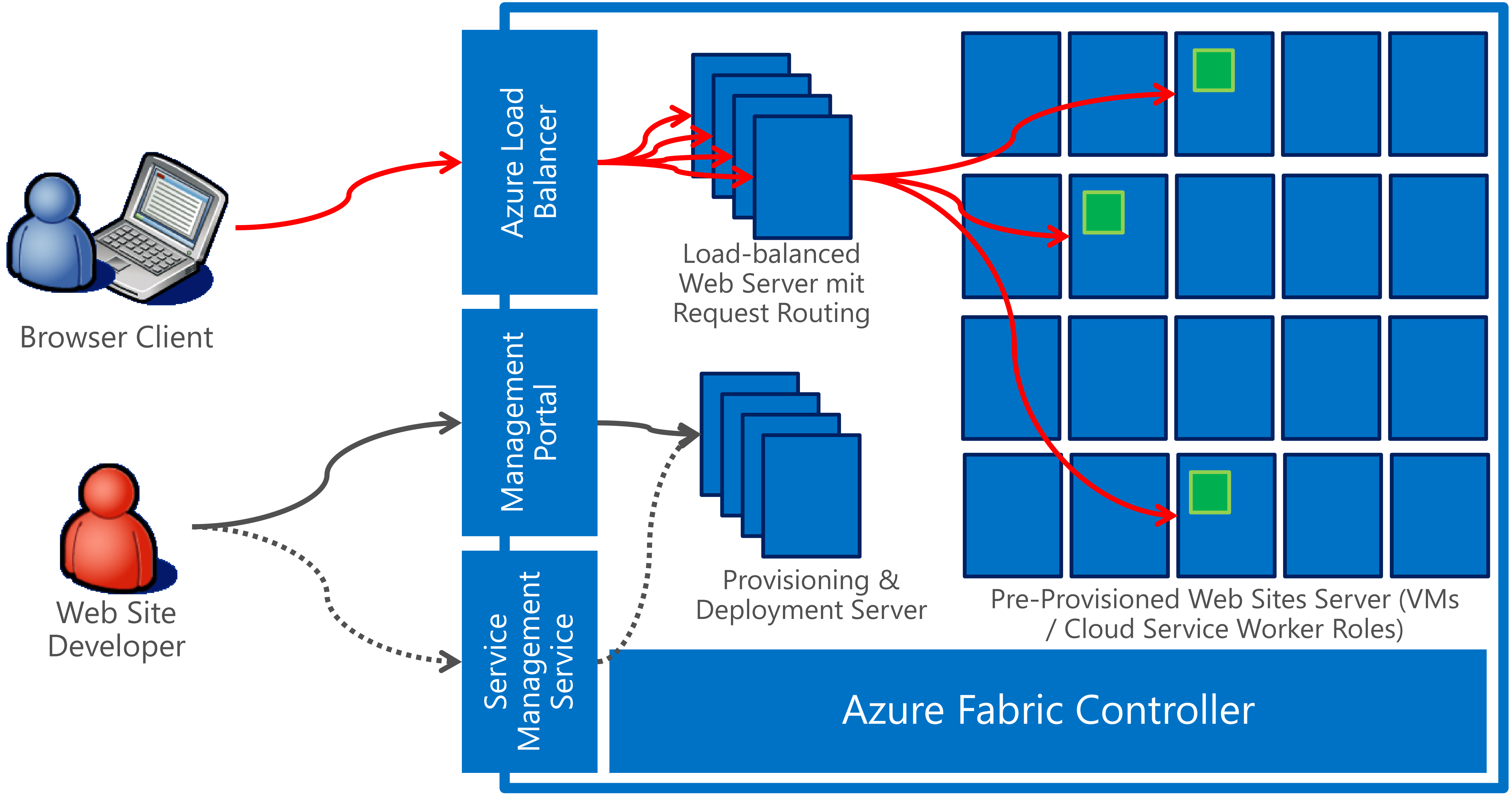
Built on-top of cloud services!!

Supported Web Frameworks

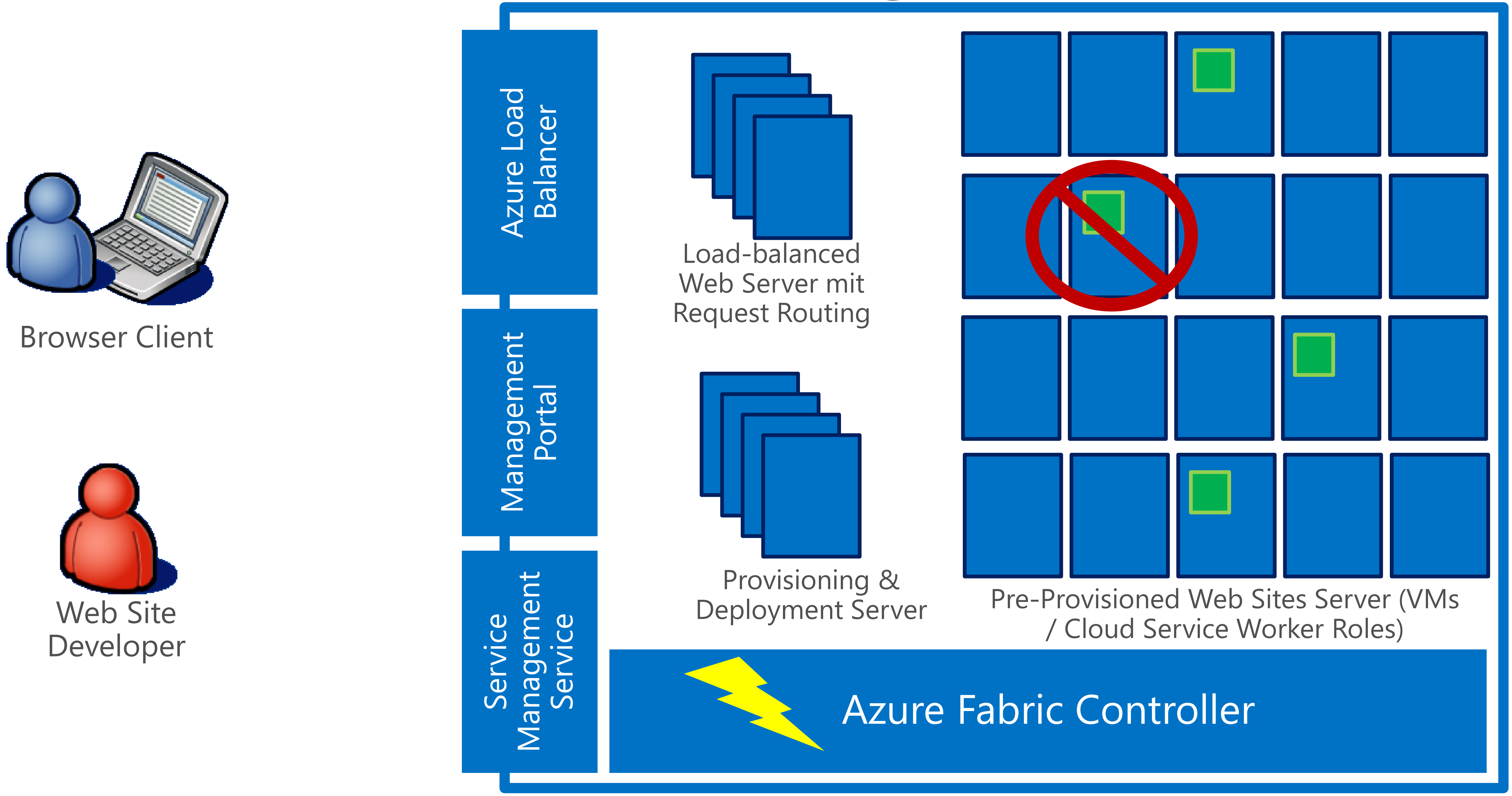


more coming soon…

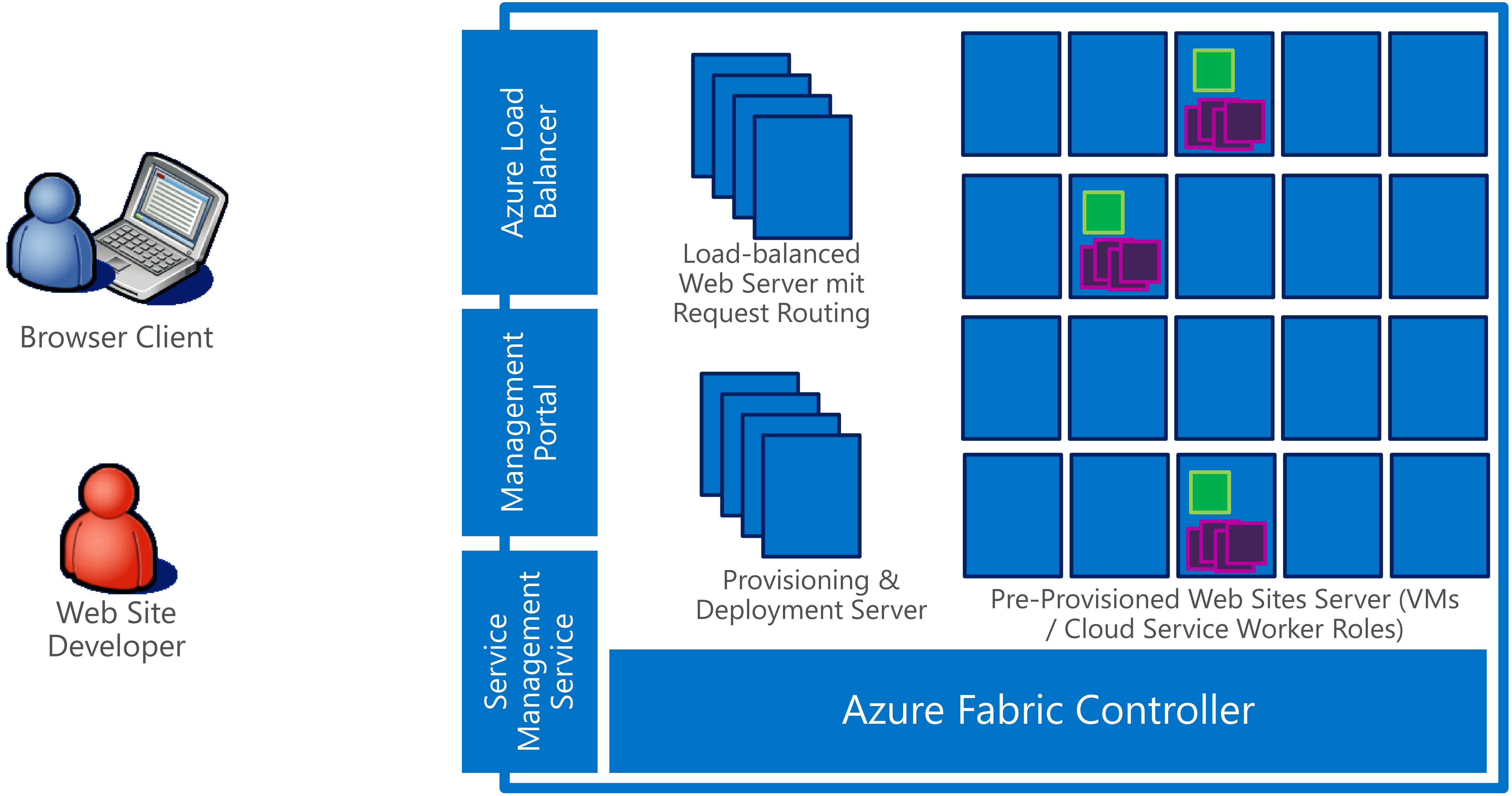
Windows Azure Web Sites Architecture


Browser Client
Web Site Developer
Azure Load Balancer
Management Portal
Service Management Service
Load-balanced Web Server mit Request Routing
Provisioning & Deployment Server
Pre-Provisioned Web Sites Server (VMs / Cloud Service Worker Roles)
Azure Fabric Controller

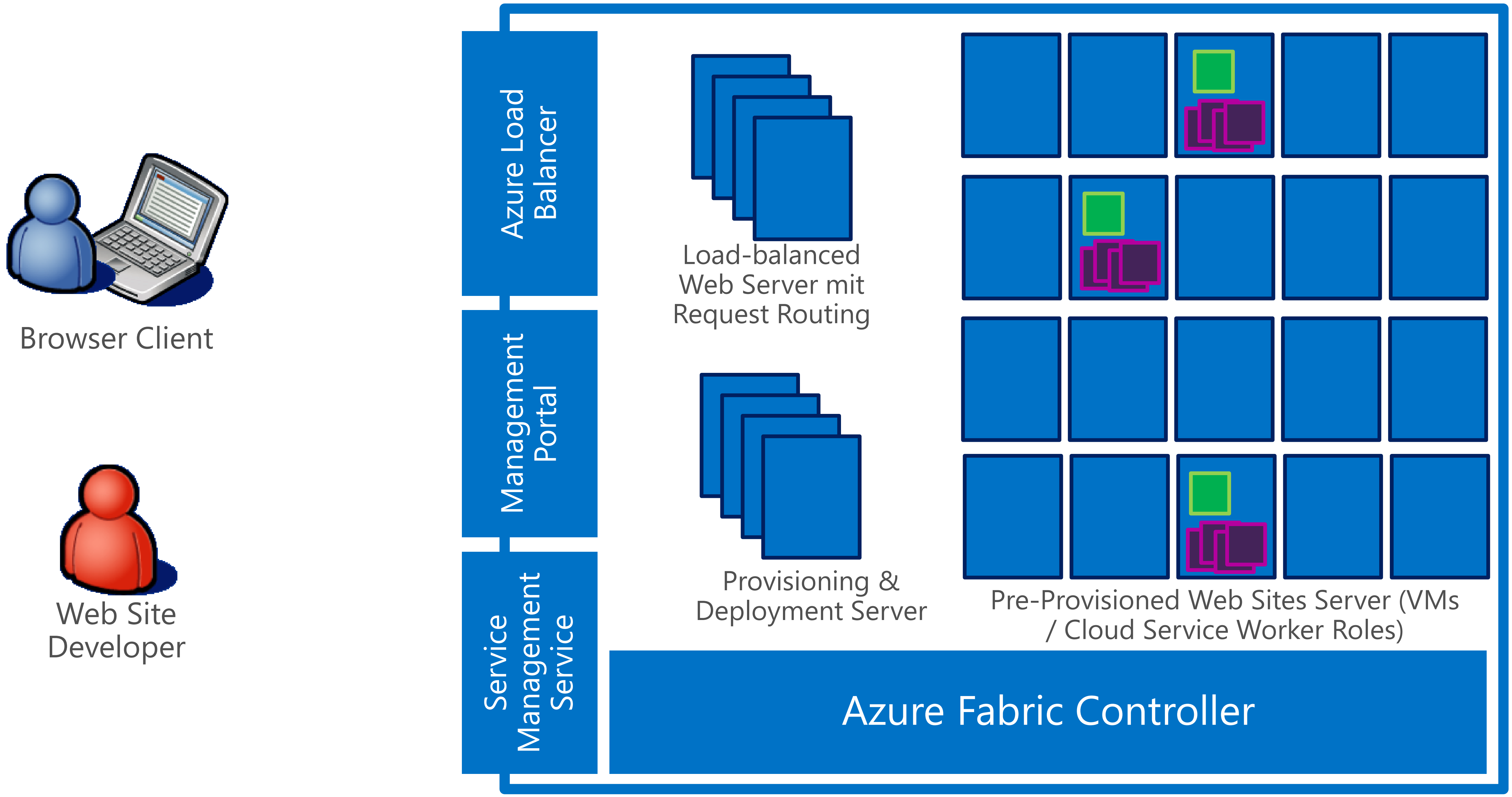
Web Sites – Auto-Management


Browser Client
Web Site Developer
Azure Load Balancer
Management Portal
Service Management Service
Load-balanced Web Server mit Request Routing
Provisioning & Deployment Server
Pre-Provisioned Web Sites Server (VMs / Cloud Service Worker Roles)
Azure Fabric Controller

Web Sites – Free & Shared


Browser Client
Web Site Developer
Azure Load Balancer
Management Portal
Service Management Service
Load-balanced Web Server mit Request Routing
Provisioning & Deployment Server
Pre-Provisioned Web Sites Server (VMs / Cloud Service Worker Roles)
Azure Fabric Controller

Web Sites – Standard (dedicated instances)


Browser Client
Web Site Developer
Azure Load Balancer
Management Portal
Service Management Service
Load-balanced Web Server mit Request Routing
Provisioning & Deployment Server
Pre-Provisioned Web Sites Server (VMs / Cloud Service Worker Roles)
Azure Fabric Controller

Summary

ASP.NET Web API 2

Attribute routing
OWIN integration
Easier Unit-testing (IHttpActionResult)
Portable Web API clients (HttpClient)
Odata-integration ($select, $expand, $batch)
Request-batching
CORS (cross origin resource sharing)
Oauth 2.0 integration

Windows Azure

Thank You!!

http://blog.mszcool.com

Windows Azure

Microsoft