PIASYS
PIALORSI SISTEMI

Publishing SharePoint Apps on Microsoft Azure

Paolo Pialorsi – paolo@pialorsi.com - @PaoloPia

About Me

- Project Manager, Consultant, Trainer
- More than 40 Microsoft certification exams passed, including MC(S)M
- Focused on SharePoint since 2002
- Author of 10 books about XML, SOAP, .NET, LINQ, and SharePoint
- Speaker at main IT conferences worldwide

- http://www.piasys.com/

PIASYS
PIALORSI SISTEMI

Agenda

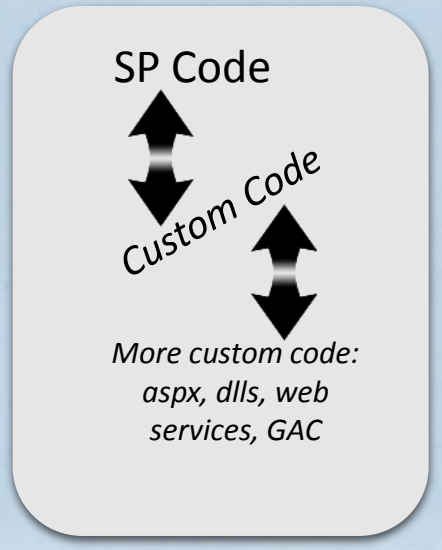
- SharePoint App Model (quick recap)
- Hosting Models
- Microsoft Azure
  - Web Sites
  - Cloud Service
  - SQL Database
- Provider-hosted apps on Microsoft Azure

SharePoint App Model

Quick recap

Evolution of SharePoint Customizations

Full Trust Solutions
No real control

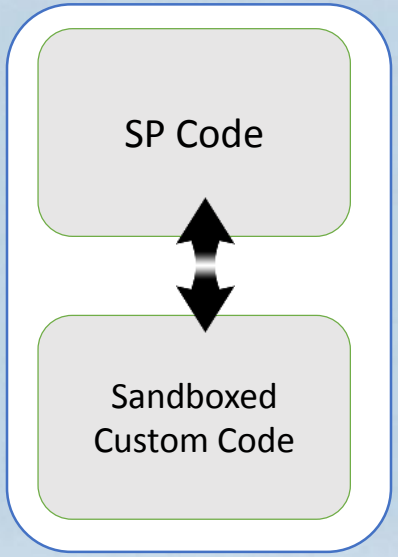


Support is a nightmare

Upgrade is quite a challenge

Securing code to run in hosted environments is effectively impossible

Sandbox
Partial control

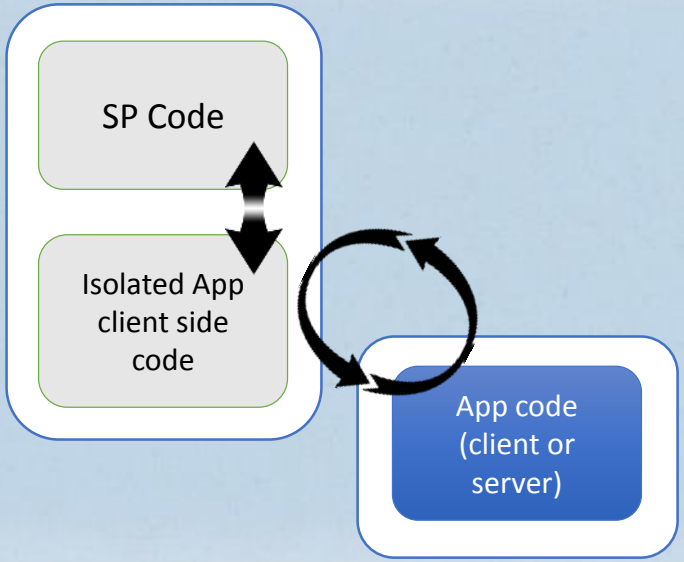


Way too strict for developers

Hard to maintain and expand

Managed by your self

App Model
Control, Trust, Manage



Host/language independent

Management and update easily doable per app

Emprases reusability

No server side sandbox, improved CSOM

Customization Development and Deployment Options

Farm

- Full trust solutions
- Customizations to file system of servers
- Hosted in same process as SharePoint
- Server side SharePoint API access
- Classic model from 2007

Sandbox

- Declarative elements
- Partially trusted code service still included for limited server side support
- Hosted in isolated process
- Limited server SharePoint API access

Code based sandbox solutions deprecated in SP2013

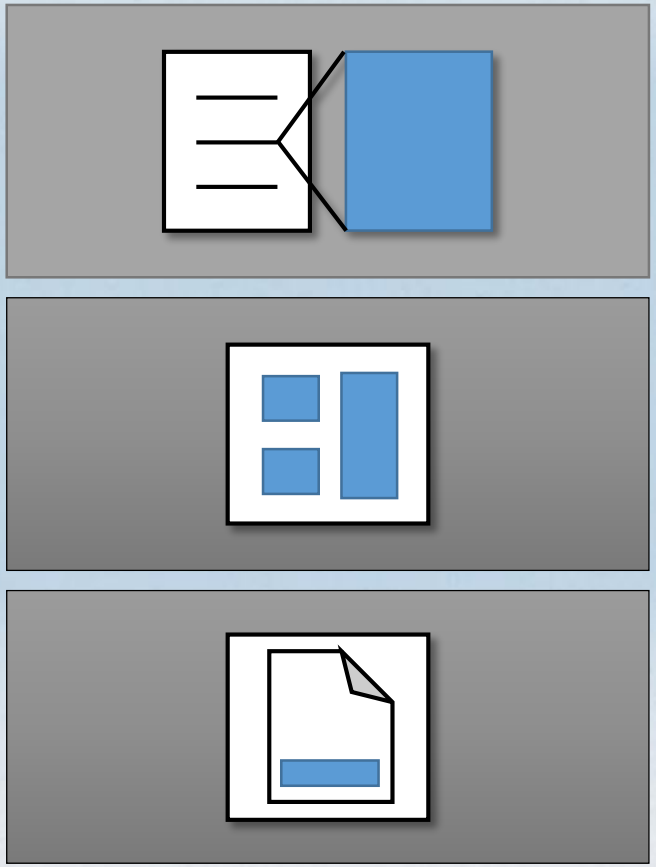
SP Apps

- New Apps model
- Deployed from corporate catalog or SharePoint store
- Manage permission and licenses specifically
- Simple install and upgrade process
- Preferred option

Redefining application models for SharePoint

Classic - Full trust solutions	Client Side Solutions	SP Apps
• ISV solutions • Platform level customizations to on-premises • Custom service applications • Custom WCF services • SharePoint customizations, not customer-specific customizations	• Server side controls as JavaScript on page layouts and master pages • Remote provisioning for elements • Embracing un-ghosted model • SP App dimension with provider-hosted apps to provide new capabilities • Customer-specific customizations	• App catalog based solution • Packaged reusable solutions built for specific functionality • Not only for marketplace or store, but also as platform for customer-specific customizations
~~Office 365~~	Office 365	Office 365

App Shapes for SharePoint



Full page

Implement complete app experiences to satisfy business scenarios

Parts

Create app parts that can interact with the SharePoint experience

UI Command extensions

Add new commands to the ribbon and item menus

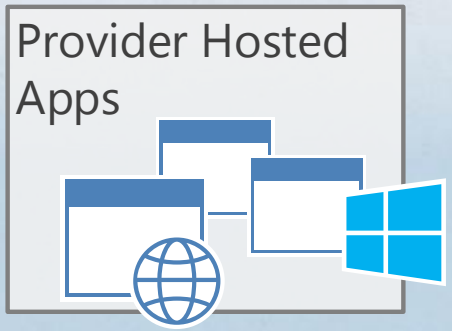
Available tools for .NET Developers

- .NET Framework
- ASP.NET and ASP.NET MVC
- Client Side Object Model for SharePoint (.NET)
- HTML5/jQuery/CSS3
- Client Side Object Model for SharePoint (JavaScript)
- SharePoint REST API
- Office 365 REST API

Hosting Models

For SharePoint Apps

SP App Hosting Options



Oauth +
REST or client object models

SharePoint

Existing sites and services

App web (optional)

The app and all resources are hosted by the provider in any environment suitable for the app.

Auto-hosted

Oauth +
REST or client object models

Existing sites and services

App web (optional)

The app is hosted in the cloud. Windows Azure and SQL Server Azure components are provisioned automatically when an app is installed. (Available for SharePoint Online only)

Removed

Windows Azure & SQL Azure

SharePoint hosted

Existing sites and services

App web

The app and all resources are hosted in your organization's SharePoint farm. Relies on client side technologies. Your IT organization supports the app.

demo

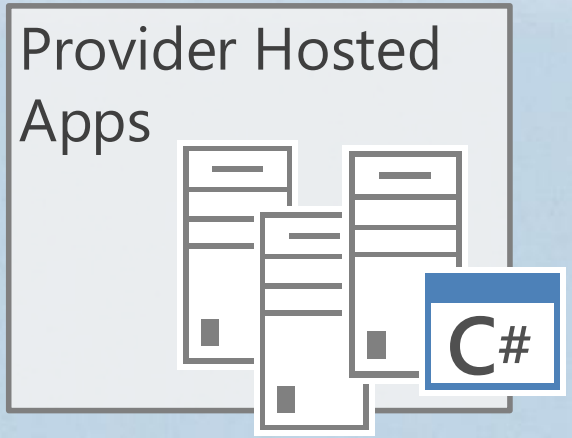
Some real-life examples

Provider hosted app hosting patterns

- There are 3 basic patterns for setting up the provider hosting environments:
  - **Shared**: Apps are hosted as separate web sites on a shared IIS servers farm (on-premises) or Azure Web Sites
  - **Dedicated**: Each app is hosted on its own dedicated IIS servers farm (on-premises) or own Azure Web Roles
  - **Per-LOB**: Each LoB has its own dedicated IIS server farm (or Azure Web Roles) and all apps belonging to the LoB are hosted on that farm (Web Roles)

General rules for provider hosted on-premises

- Load balanced Windows Server hosting ASP.NET application in IIS
- Low trust (ACS) OAuth model requires Internet connectivity for server side for Office365 Dedicated app environment
- Typical web server hardware requirements
  - Minimum: 8 GB, 4 cores
  - Recommended: 16 GB, 8 cores as a typical setup
- Optional SQL Server instance for possible databases used in provider hosted apps
- Each provider hosted app has specific own domain which is routed in DNS
  - Domain depends on policies, but quite often look something like spapp-appname.group.Contoso.com
- Actual layout is dependent on usage model, but we should be fine with load balanced 3 servers per location
  - Each app is hosted from these same servers



Microsoft Azure

For SharePoint Apps

Useful Services

- Azure Web Sites
- Azure Cloud Service
- Azure SQL Database

Azure Web Sites

- Easy to use web hosting model
  - Multi-language support (.NET, PHP, Java, Node.js, Python)
  - Multi-backend support (MS-SQL, MySQL, MongoDB, Azure Data Services, etc.)
  - Really easy to make it Highly Available (just a few clicks)
- Supports SSL for free with default host names
  - sitename.azurewebsites.net (* SSL for *.azurewebsites.net)
- Supports on-premises data sources
  - That's a cool new feature! Really useful for LOB SharePoint apps!
- Easy development and management
  - Support staging and production environment
  - Scheduled scaling or automatic scaling
  - Health monitoring
  - Automated backup
- Available for single web sites only
  - No multi-tenancy, or not for free ☺

Azure Cloud Service

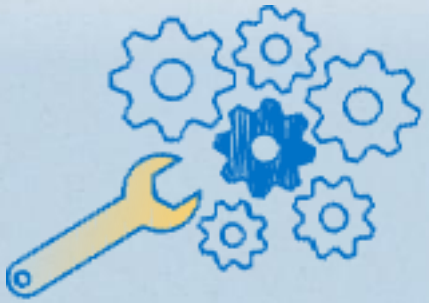
- Multi-language (.NET, Java, Node.js, PHP, Python, or Ruby)
- Provides
  - Worker Role
  - Web Role
- Focus on code and business data, not on hardware/physical resources
  - It is a “Platform as a Service” (PaaS) offering
- Extensible: you can deploy additional packages/libraries
- Provides staging/testing environment and production environment
  - Easy swap between test and production, and revert back, as well
- Health monitoring and alerts
- Auto-scale to optimize cost and performance
- Good option for multi-tenancy, multi-app, dedicated hosting

Azure SQL Database

- Microsoft SQL Server as a service
- Highly Available:
  - SLA: Up to 99,99% (52.56min/year!)
  - Continuous replica, even geographically (Premium)
- Automatic or scheduled backups
  - Premium automatically holds up to 35 days of backup
- Certified Security (HIPAA BAA, ISO/IEC 27001:2005, FedRAMP, EU Model Clauses)

Provider-hosted Apps on Microsoft Azure

Publishing SharePoint Apps on Azure

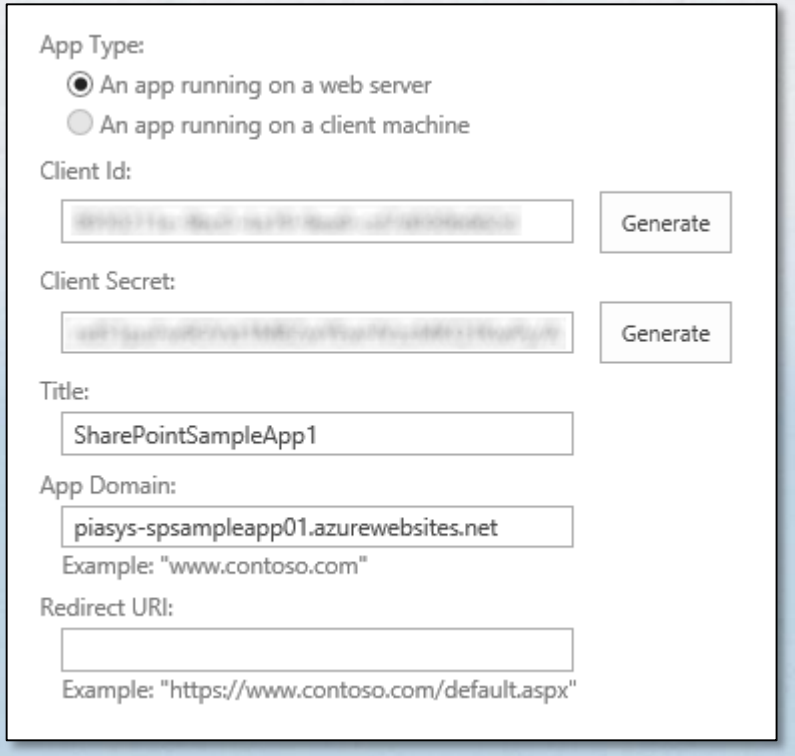


demo

Creating a provider-hosted SharePoint App on Microsoft Azure

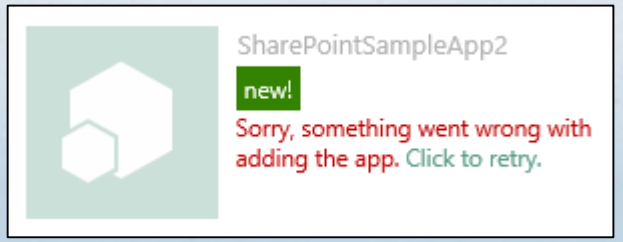
Steps Recall

- Create the SharePoint App project
  - Target Office 365 or SharePoint on-premises
  - Choose your preferred model (WebForms or MVC)
  - Choose ACS for Office 365, or S2S for on-premises
- Develop and debug locally (IIS Express)
  - F5 and debug
- Publish the SharePoint app on Azure
  - Create ClientID and ClientSecret (/_layouts/15/AppRegNew.aspx)
  - Test profile
  - Production profile
- Release using the .app file
  - Corporate App Catalog
  - Office Store

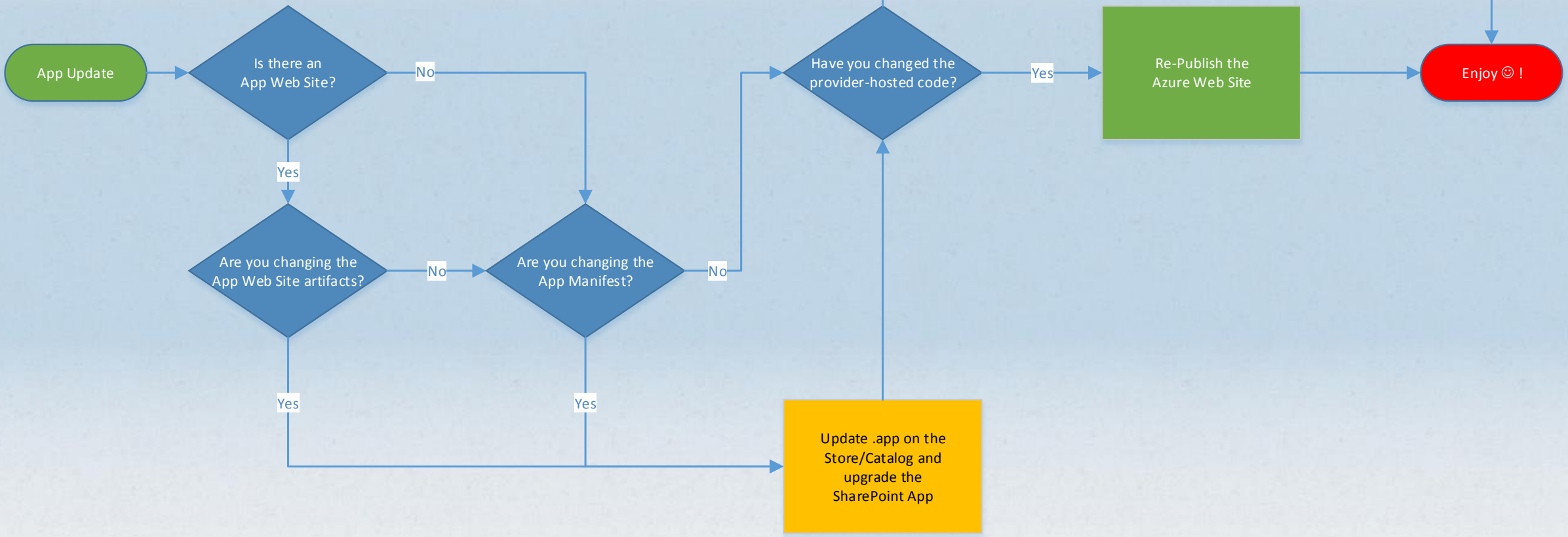


Common Issues

- Life could be better ☺ …
- … but even worst
- App fails to install
- App fails to update/remove
- Sometime the only solution is to recreate the ApplicationID
  - Ouch! ☹



Maintenance and Upgrade



Best Practices

- Deploy both test and production environments on Azure
  - Use separate Azure Web Site and Azure SQL Database, if any
- But share the same tenant to share the same platform version
  - And the same Azure AD under the cover
- Remember that you can update Azure Web Sites independently from the SharePoint environment

- Don’t forget to check, monitor and use Office 365 Developer PnP
  - https://github.com/OfficeDev/PnP/

That’s all folks! Thank YOU!