#SPSBE

Real-life Solutions with Microsoft Azure, Microsoft 365, and the Microsoft Graph

Paolo Pialorsi – PiaSys.com - @PaoloPia

BIWUG

SharePoint Saturday Belgium 2019

Thanks to our sponsors!

Platinum

Gold

Silver

Community

About me

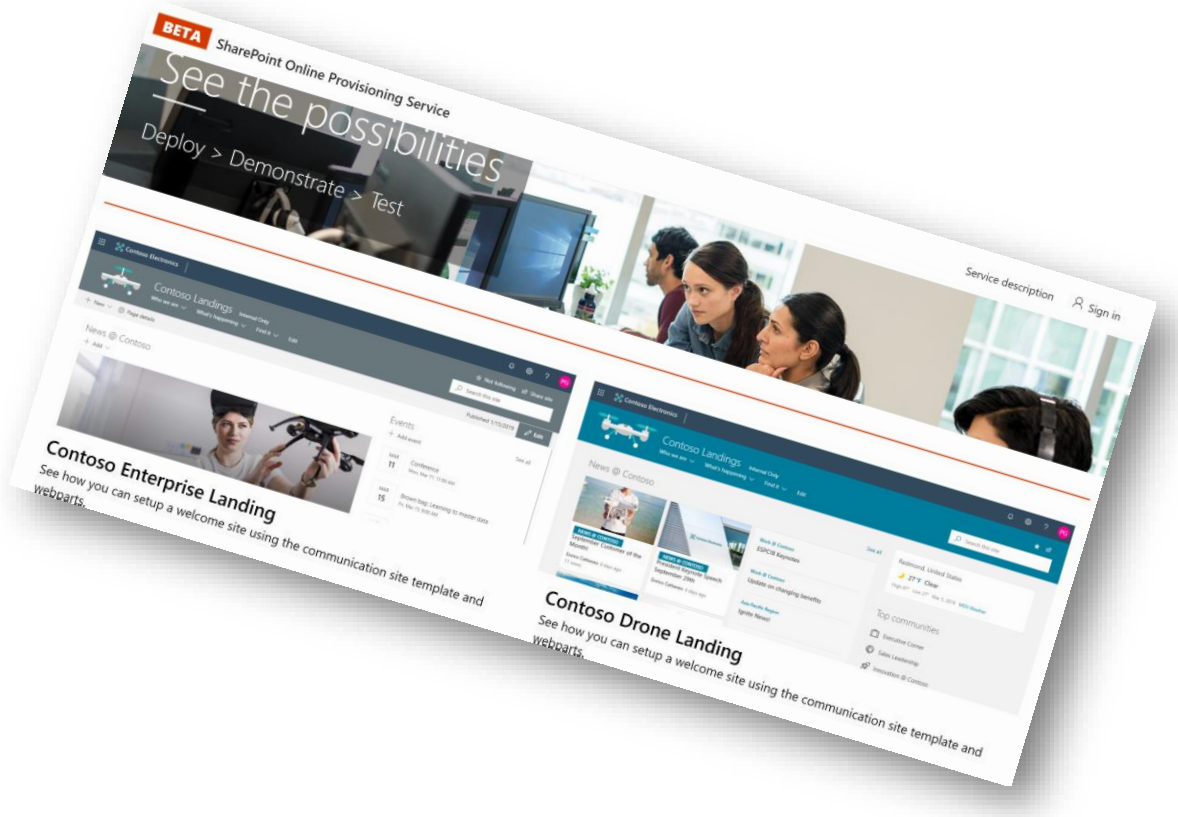
- Project Manager, Consultant, Trainer
  - PiaSys.com based in the USA and in Italy
- About 50 Microsoft certification exams passed
  - MCSM – Charter SharePoint
  - MVP Office Servers and Services
  - SharePoint PnP Core Team Member
- Focused on SharePoint and Office 365
- Author of many books about XML, SOAP, .NET, LINQ, SharePoint, and Office 365
- Speaker at main IT conferences worldwide
- Follow me @PaoloPia - https://piasys.com/TechBites

Topics

- Introducing SharePoint Online Provisioning Service
- Architecture of the solution
- Challenges and choices
- Inside the solution (source code and infrastructure)

SharePoint Online Provisioning Service

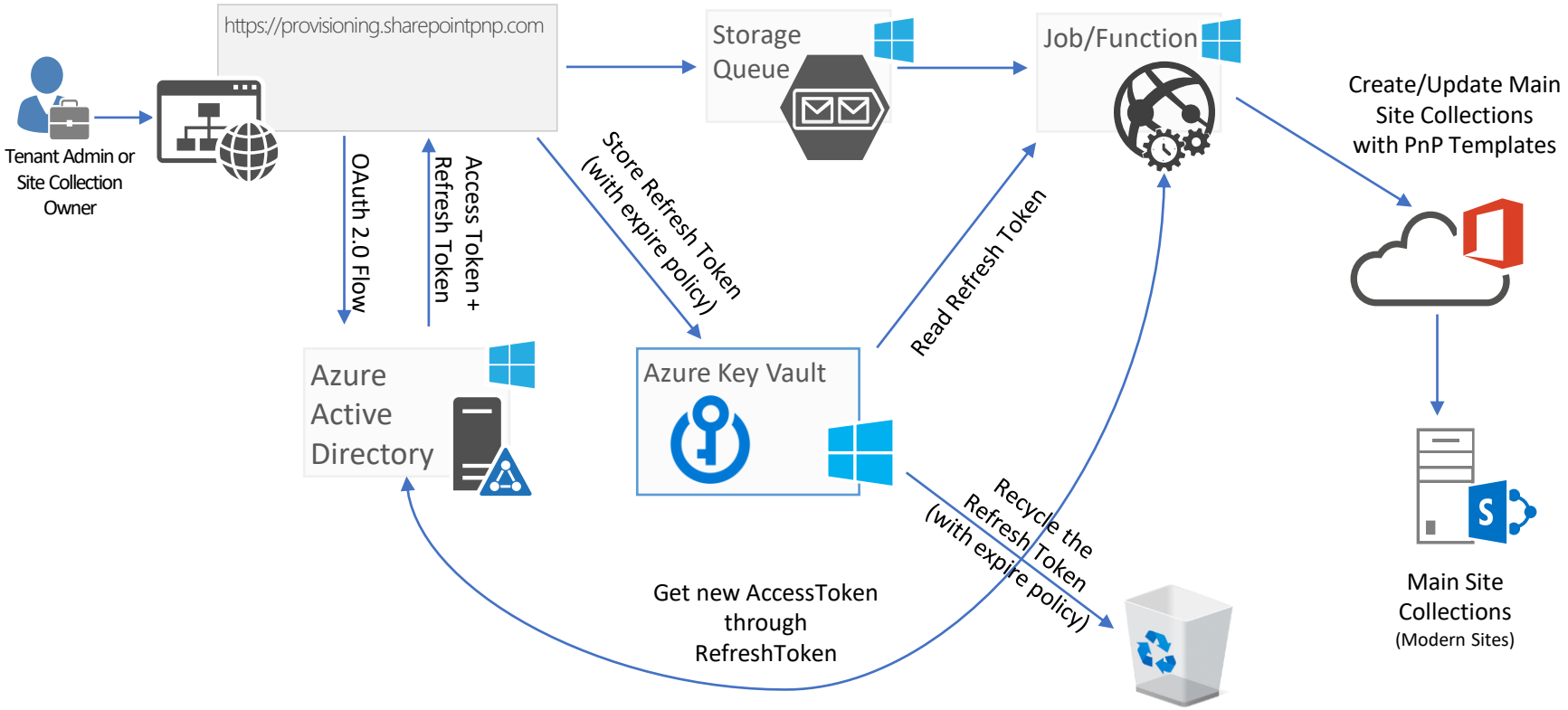
- It’s “PnP Provisioning as a Service”
- AAD multi-tenant application
- Built on Microsoft Azure
  - Azure Web App
  - Azure Web Job
  - Azure SQL Database
  - Azure Blob Storage
  - Azure Key Vault
  - Azure Function
  - Application Insights
  - GitHub as the templates repository
- It is open source!



Demo

Lap around the SharePoint Online Provisioning Service

Architectural Schema



Challenges

- Multiple environments
- Templates Repository
- Content Pages
- Global and Templates Settings
- High Availability
- Asynchronous Processing
- Reusable Components (CSOM and PnP)
- Security: Authentication and Authorization
- OAuth Security Layer
- Monitoring

Multiple Environments

- Functional requirements:
  - Support for DEV, TEST, PROD environments
  - Different versions of the engine across different environments
  - Same set of templates/packages
  - Different access rules
- Technical solution:
  - Microsoft Azure deployment slots for web apps/jobs/functions
  - Dedicated Azure infrastructure for every single environment
  - Multiple AAD app registrations
  - Unique content repository with “settings” for access rules
  - Maintainable list of allowed tenants for DEV, TEST
  - Shared security storage repository (Azure Key Vault) with different keys

Templates Repository

- Functional requirements:
  - Something “easy” to update and manage
  - Something “accessible” to MS people
  - Something “accessible” to community members
  - Version and history tracking
  - Two main clusters of templates/packages: Tenant-wide and Site Collection
- Technical solution:
  - Dedicated GitHub repository
  - Folders for clusters (tenant, site)
  - Replicated on SQL Azure Database + Azure Blob Storage Account
  - Synchronization job running on schedule (or manually triggered)
  - Leveraging GitHub APIs

Content Pages

- Functional requirements:
  - Something “easy” to update and manage
  - Something “accessible” to MS people
  - Something “accessible” to community members
  - Version and history tracking
  - Based on HTML or HTML-like syntax
- Technical solution:
  - MD files in the GitHub repository
  - Replicated on SQL Azure Database + Media in Azure Blob Storage Account
  - Synchronization job running on schedule (or manually triggered)
  - Leveraging GitHub APIs + Conversion from MD to HTML (native in GitHub APIs)

System and Template Settings

- Functional requirements:
  - Being able to configure different settings for system and templates/packages
  - Easy to configure and maintain
  - No need to access Azure for the admin users
  - Content based on “open” syntax
- Technical solution:
  - System configuration files: categories.json, platforms.json, and tenants.json
  - Template configuration files: settings.json
  - Replicated on SQL Azure Database + Azure Blob Storage Account
  - Synchronization job running on schedule (or manually triggered)
  - Leveraging GitHub APIs

Highly Scalable Solution

- Functional requirements:
  - Support for thousands of tenants
  - Support for thousands of provisioning actions every day
  - Capability to “resume” failed provisioning actions, if needed
  - History of failed provisioning actions
  - Resilient to throttling
- Technical solution:
  - Azure Blob Storage Queue
  - Azure Web Job queue-triggered
  - Configured for 1 action retry only + poison queue for analysis and eventually retries
  - Out of the box web-based UI to monitor running actions and history

Why a WebJob?

- Easy support for .NET Classic
  - CSOM is not available yet for .NET Core
  - As such, PnP Sites Core and PnP Provisioning Engine are not available for .NET Core, either
  - Azure Functions v2 target .NET Core and PowerShell Core only
- We already have web applications
  - So we can reuse exactly the same resources (app service, app service plan, deployment slots)

Authentication and Authorization

- Functional requirements:
  - Available to Tenant Admins as well as to Site Collection Admins
  - Provision sites and artifacts on behalf of the requesting user
- Technical solution:
  - Two different AAD applications (1 for Tenant Admins, 1 for Site Collection Admins)
  - Two different web applications (2 virtual directories in Azure Web App)
  - OAuth flow with Authorization Code
  - Access Token and Refresh Token via Authorization Code
  - Refresh Token stored in secure and highly available cache: Azure Key Vault

Monitoring

- Functional requirements:
  - We need to monitor numbers and performance (stats)
  - We need to monitor exceptions (strength)
- Technical solution:
  - Anonymized Stats on Azure SQL Database (packages provisioned and exceptions count)
  - Azure Application Insights (number of provisionings, exceptions with stack trace)
  - PnP Correlation ID to keep track of items and provisioning flows

Demo

Let’s dig into the solution

https://github.com/SharePoint/sp-provisioning-service

Technologies Involved

- Microsoft Azure
  - Azure Active Directory
    - Apps registration
    - OAuth 2.0
  - Azure SQL Database
    - Data storage
  - Azure Storage Account
    - Blob Storage
    - Queue
  - Azure App Service
    - WebApps
    - WebJobs
  - Azure Key Vault
    - Secure store for keys and passwords
  - Azure Functions
    - Reporting
  - Azure Application Insights
    - Monitoring and tracking
- .NET Framework 4.7.1
  - ASP.NET MVC
  - Entity Framework 6 – Code First + Migrations
- GitHub
  - Repository for content and pages
- Microsoft Graph
  - API for provisioning
- SharePoint Online
  - API for provisioning
- Microsoft 365
  - The target platform
- SharePoint PnP
  - PnP Sites Core
  - Provisioning Service

Q&A

#SPSBE

http://spsbe.be

Please rate this session!

BIWUG

#SPSBE

BIWUG

SharePoint Saturday Belgium 2019