API Management

Sonu Sathyadas

Agenda

SYNERGETICS
GET IT RIGHT
Training | Consulting | Cloud Services | Staffing

- Understanding API Management
- Overview of Publisher and Developer portals
- Publish/Setup Web API in Azure (WADL, Swagger)
  - Use of Products, Analytics in Azure API Management
  - Configuring Policies at product or API or Operation level, API Caching
- Securing backend APIs
  - Implementing API securities using AAD, OAuth 2.0, OpenID Connect , Federated etc
  - Protecting and Optimizing APIs
- Customizing the Developer Portal
  - Working with API Errors and API monitoring techniques
  - Tracing using API inspector and Logging to Azure Event Hubs

Understanding API Management

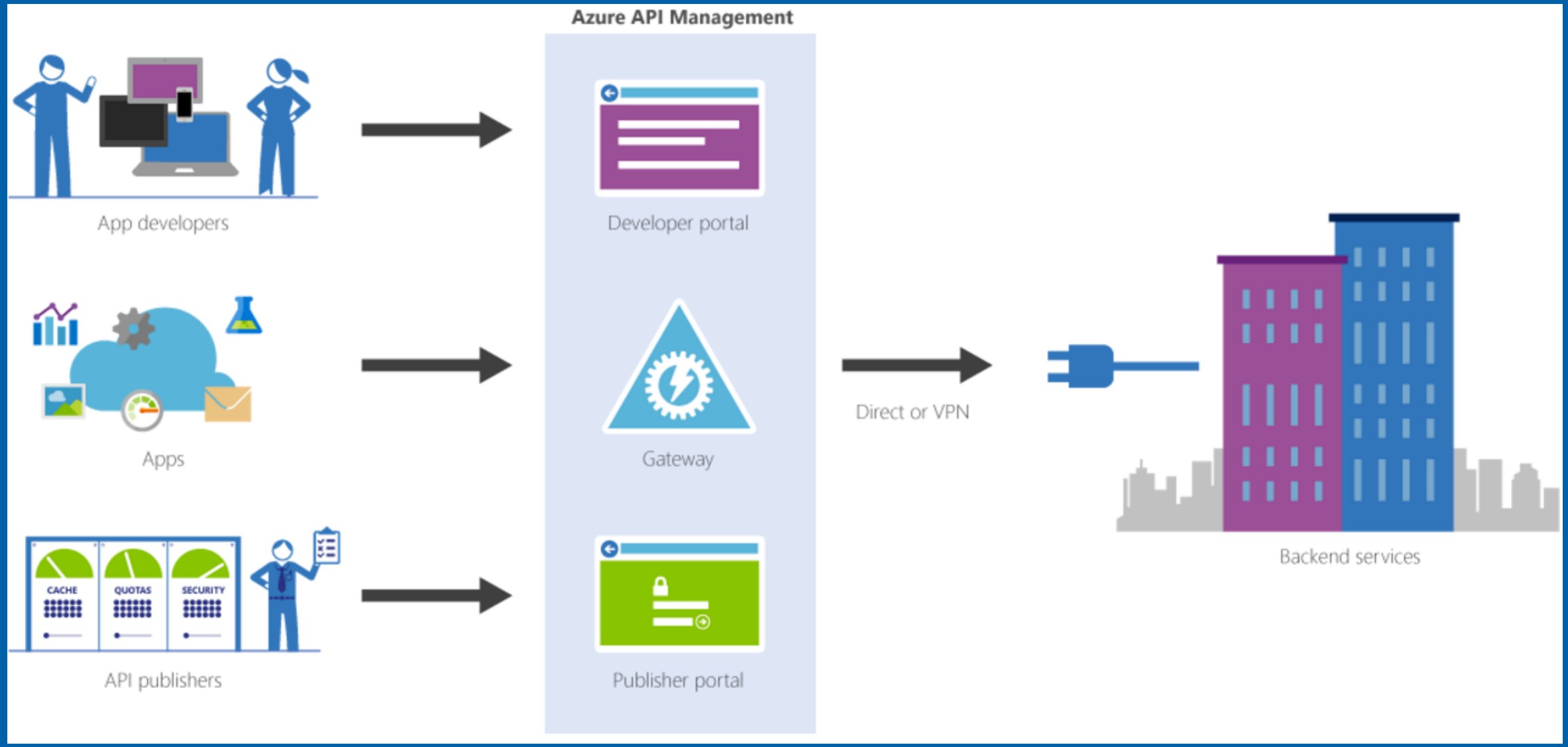
Azure API Management – Scenarios

- **Securing mobile infrastructure**
  - By gating access with API keys, preventing DOS attacks by using throttling, or using advanced security policies like JWT token validation.
- **Enabling ISV partner ecosystems**
  - By offering fast partner onboarding through the developer portal and building an API facade to decouple from internal implementations that are not ripe for partner consumption.
- **Running an internal API program**
  - By offering a centralized location for the organization to communicate about the availability and latest changes to APIs, gating access based on organizational accounts, all based on a secured channel between the API gateway and the backend.

Azure API Management

- Azure API Management allows organizations to publish APIs more securely, reliably, and at scale.
- Use API Management to drive API consumption among internal teams, partners and developers while benefiting from business and log analytics available in the admin portal.
- End-to-end API management—everything from provisioning user roles, creating usage plans and quotas, applying policies for transforming payloads, throttling, analytics, monitoring and alerts.

Azure API Management



Provide first-rate developer experience

- Self-service API key management
- Auto-generated API catalogue, documentation and code samples
- OAuth-enabled API console for exploring APIs without writing a line of code
- Sign in using popular Internet identity providers and Azure Active Directory

Protect and optimize your APIs

- Simplify and optimize requests and responses with transformation policies
- Secure APIs with key, JWT token validation and IP filtering
- Protect your APIs from overload and overuse with quotas and rate limits
- Use response caching for improved latency and scale

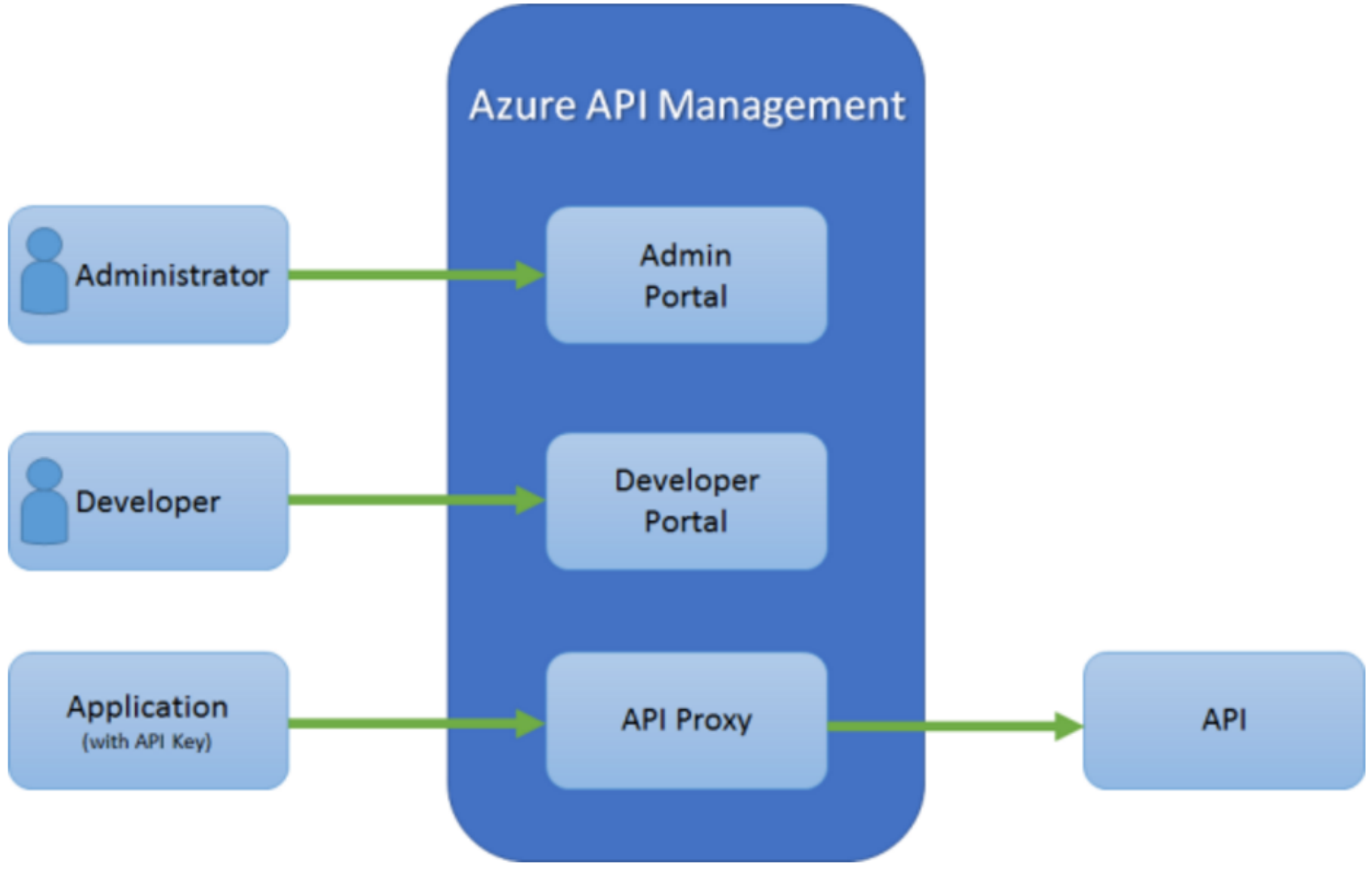
Manage all your APIs in one place

- Expose all APIs behind a single static IP and domain
- View near real-time usage, performance and health analytics
- Automate management and integrate using REST API, PowerShell and Git
- Provision API Management and scale it on demand in one or more geographical regions

System is made up of the following components

- **API gateway** is the endpoint that:
  - Accepts API calls and routes them to your backends.
  - Verifies API keys, JWT tokens, certificates, and other credentials.
  - Enforces usage quotas and rate limits.
  - Transforms your API on the fly without code modifications.
  - Caches backend responses where set up.
  - Logs call metadata for analytics purposes.
- **Publisher portal** is the administrative interface where you set up your API program. Use it to:
  - Define or import API schema.
  - Package APIs into products.
  - Set up policies like quotas or transformations on the APIs.
  - Get insights from analytics.
  - Manage users.
- **Developer portal** serves as the main web presence for developers, where they can:
  - Read API documentation.
  - Try out an API via the interactive console.
  - Create an account and subscribe to get API keys.
  - Access analytics on their own usage.

Azure API Management



Overview of Publisher and Developer portals

Basic Calc Demo

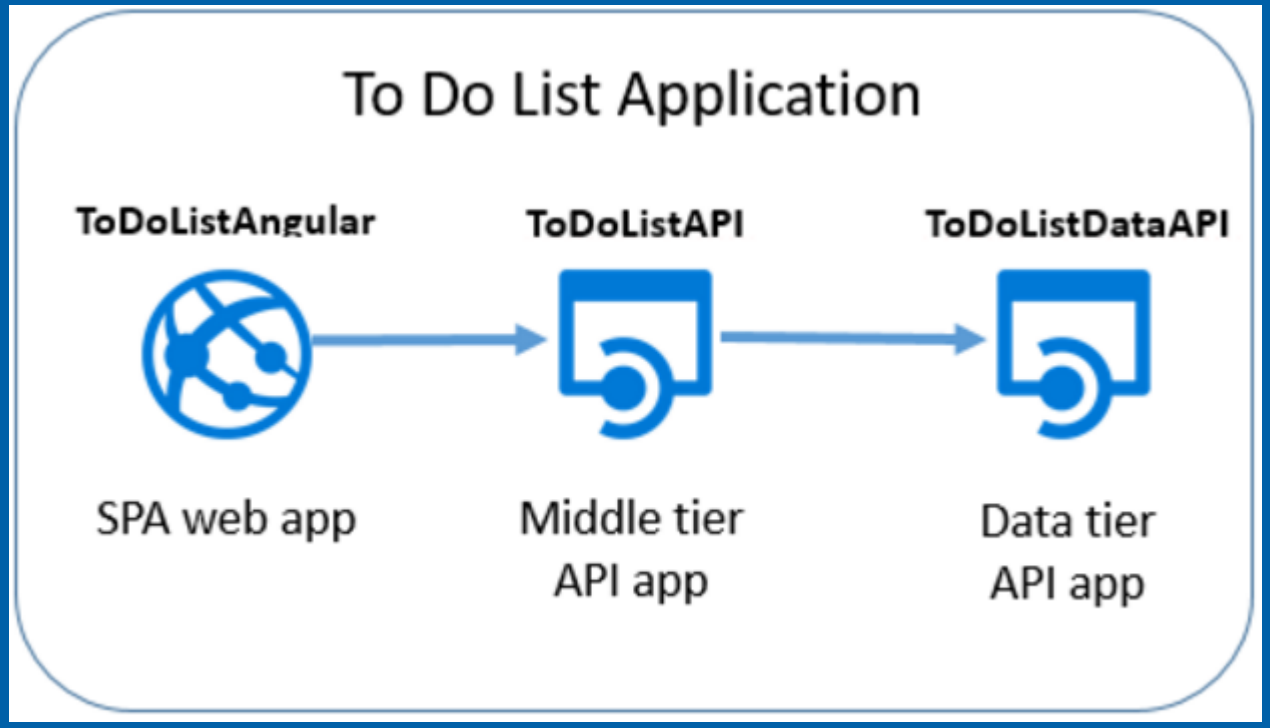
Complete Walkthrough Steps

- Develop ASP.Net Web API Project
- Host/Deploy to Azure
- Create instance of API Management
- Add Created API
- Add Operations to API
- Create subscribers
- Test APIs

- http://www.codeproject.com/Articles/890435/Getting-Started-with-Azure-API-Management

Publish API App

Publish/Setup Web API in Azure - Scenario



Publish/Setup Web API in Azure - Steps

- Download Sample Application
- Use Swagger API
- Create API App and deploy on Azure
- API definition blade on Azure Portal
- Generate client code for Data Tier
- Create API app to host middle tier
- Configure the middle tier to call the data tier

Publish/Setup Web API in Azure

- Building End-to-End Web API solution
  - https://azure.microsoft.com/en-in/documentation/articles/app-service-api-dotnet-get-started/ - HOL
  - https://azure.microsoft.com/en-in/documentation/articles/web-sites-dotnet-rest-service-aspnet-api-sql-database/ - HOL with DB

Products and Analytics in API Management

Use of Products, Analytics in Azure API Management

- A product contains one or more APIs as well as a usage quota and the terms of use.
- Once a product is published, developers can subscribe to the product and begin to use the product's APIs.
- Operations are added and configured to an API in the publisher portal. To access the publisher portal, click **Manage** in the Azure Classic Portal for your API Management service.
- Analytics of each API call can be viewed on the portal as a graphical representation

Training | Consulting | Cloud Services | Staffing

Walkthrough

https://azure.microsoft.com/en-in/documentation/articles/api-management-howto-add-products/

Configuring Policies

Configuring Policies at product or API or Operation level

- Policies are a powerful capability of the system that allow the publisher to change the behavior of the API through configuration.
- Policies are a collection of Statements that are executed sequentially on the request or response of an API.
- Popular Statements include format conversion from XML to JSON and call rate limiting to restrict the amount of incoming calls from a developer. Many more policies are available out of the box.

Configuring Policies at product or API or Operation level

- Policies are applied inside the gateway which sits between the API consumer and the managed API.
- The gateway receives all requests and usually forwards them unaltered to the underlying API. However a policy can apply changes to both the inbound request and outbound response.
- Policy expressions can be used as attribute values or text values in any of the API Management policies, unless the policy specifies otherwise. Some policies such as the Control flow and Set variable policies are based on policy expressions.

Training | Consulting | Cloud Services | Staffing

Walkthrough

https://azure.microsoft.com/en-in/documentation/articles/api-management-howto-policies/

Training | Consulting | Cloud Services | Staffing

API Caching and Security

API Caching

- Response caching can significantly reduce API latency, bandwidth consumption, and web service load for data that does not change frequently.
- Azure API Management service has built-in support for HTTP response caching using the resource URL as the key.
- The key can be modified by request headers using the vary-by properties. This is useful for caching entire HTTP responses (aka representations), but sometimes it is useful to just cache a portion of a representation.

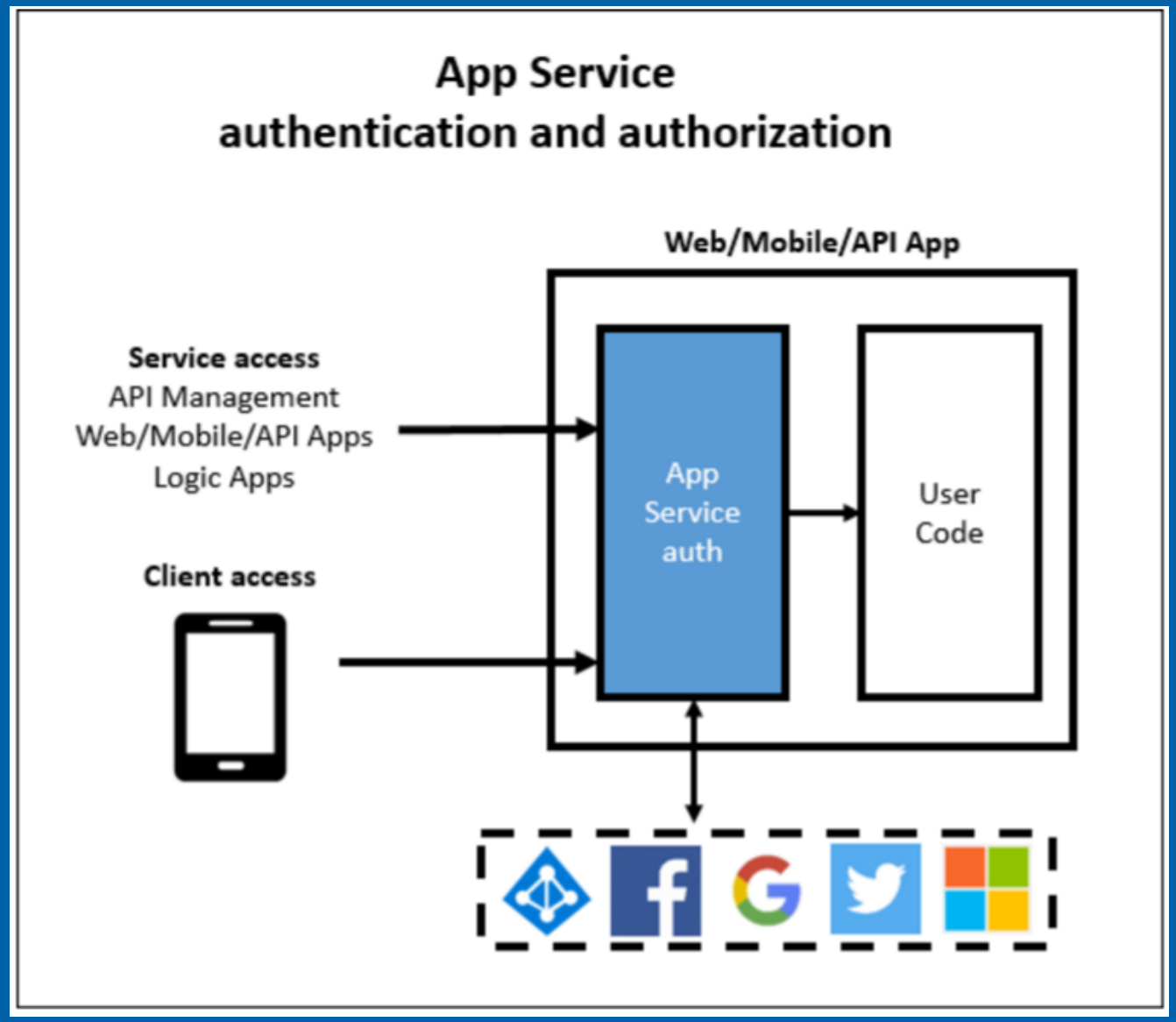
API caching

- The new cache-lookup-value and cache-store-value policies provide the ability to store and retrieve arbitrary pieces of data from within policy definitions. This ability also adds value to the previously introduced send-request policy because you can now cache responses from external services.
- API Management service uses a shared per-tenant data cache so that, as you scale up to multiple units you will still get access to the same cached data.
- When working with a multi-region deployment there are independent caches within each of the regions. Due to this, it is important to not treat the cache as a data store, where it is the only source of some piece of information. If you did, and later decided to take advantage of the multi-region deployment, then customers with users that travel may lose access to that cached data.

Securing backend APIs

- API Management provides the capability to secure access to the back-end service of an API using client certificates.
- https://azure.microsoft.com/en-in/documentation/articles/api-management-howto-mutual-certificates/ - HOL

Implementing API securities



Implementing API securities using AAD, OAuth 2.0, OpenID Connect , Federated, third party like Ping Identity etc

- Language Agnostic
- Multiple Protection Option
- Service Account Authentication
- https://azure.microsoft.com/en-in/documentation/articles/api-management-howto-protect-backend-with-aad/-HOL

Training | Consulting | Cloud Services | Staffing

https://azure.microsoft.com/en-in/documentation/articles/api-management-customize-portal/

Customizing the Developer Portal

Working with API Errors and API monitoring techniques

Working with API Errors and API monitoring techniques

- The API Management service provides many capabilities to enhance the processing of HTTP requests sent to your HTTP API.
- The existence of the requests and responses are transient.
- The request is made and it flows through the API Management service to your backend API. Your API processes the request and a response flows back through to the API consumer.
- The API Management service keeps some important statistics about the APIs for display in the Publisher portal dashboard, but beyond that, the details are gone.
- By using the log-to-eventhub policy in the API Management service you can send any details from the request and response to an Azure Event Hub.
- There are a variety of reasons why you may want to generate events from HTTP messages being sent to your APIs. Some examples include audit trail of updates, usage analytics, exception alerting and 3rd party integrations.

Tracing using API inspector

- To use API Inspector, add an **ocp-apim-trace: true** request header to your operation call, and then download and inspect the trace using the URL indicated by the **ocp-apim-trace-location**response header. This can be done programatically, and can also be done directly from the developer portal.
- To review the values in the trace, download the trace file from the **ocp-apim-trace-location**URL

SYNERGETICS
GET IT RIGHT
Training | Consulting | Cloud Services | Staffing

API Inspector- - Demo

https://azure.microsoft.com/en-in/documentation/articles/api-management-howto-api-inspector/

Training | Consulting | Cloud Services | Staffing

Logging to Azure Event Hubs

Logging to Azure Event Hubs

- Azure Event Hubs is a highly scalable data ingress service that can ingest millions of events per second so that you can process and analyze the massive amounts of data produced by your connected devices and applications.
- Event Hubs acts as the "front door" for an event pipeline, and once data is collected into an event hub, it can be transformed and stored using any real-time analytics provider or batching/storage adapters.
- Event Hubs decouples the production of a stream of events from the consumption of those events, so that event consumers can access the events on their own schedule.

Training | Consulting | Cloud Services | Staffing

Walkthrough

https://azure.microsoft.com/en-in/documentation/articles/api-management-howto-log-event-hubs/

Training | Consulting | Cloud Services | Staffing

End-To-End Event Hub Sample

https://azure.microsoft.com/en-us/documentation/articles/api-management-log-to-eventhub-sample/

Thank you

omprakashpandey@synergetics-india.com