Authentication in Azure Active Directory (Marc de Fontenay)
Definitions

Cloud : *Cloud computing is a general term for the delivery of hosted services over the internet. Cloud computing enables companies to consume a compute resource, such as a virtual machine (VMs), storage or an application, as a utility, rather than having to build and maintain computing infrastructures in house. Put simply, Cloud computing means that instead of all the computer hardware and software you're using sitting on your desktop, or somewhere inside your company's network, it's provided for you as a service by another company and accessed over the Internet, usually in a completely seamless way.*

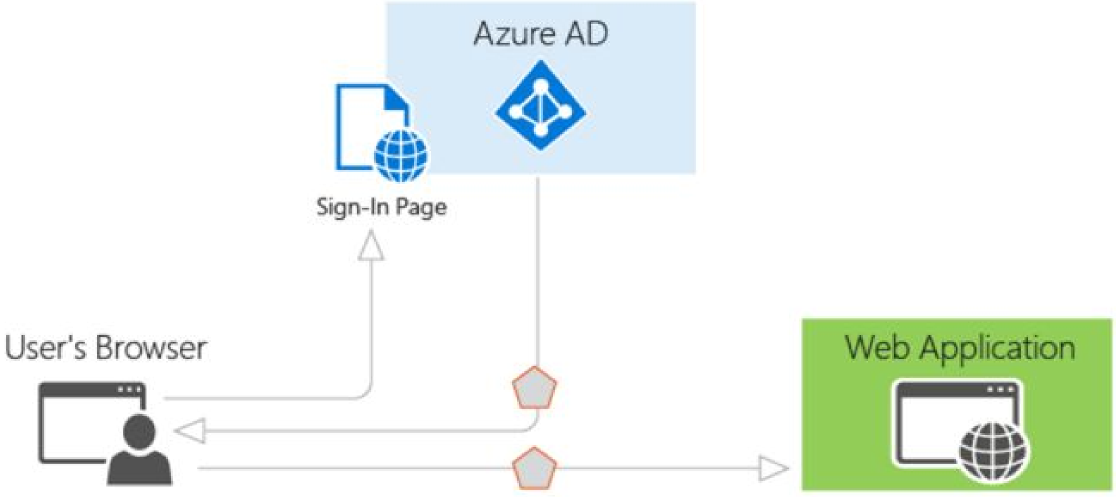
Authentication : *In the context of computer systems, authentication is a process that ensures and confirms a user's identity. Since Access Control is normally based on the identity of the User who requests access to a resource, Authentication is essential to effective Security.*

OAuth : *An open protocol to allow secure authorization in a simple and standard method from web, mobile and desktop applications. It is used as a way for Internet users to log in to third party websites through an access token mechanism, using their accounts at Google, Facebook, Microsoft, Twitter, etc.—but without exposing their password.*

Microsoft Azure : Microsoft's public cloud computing platform. It provides a range of cloud services, including those for compute, analytics, storage and networking. Users can pick and choose from these services to develop and scale new applications, build advanced analytics in the cloud by creating machine learning experiments, or run existing applications, in the public cloud.

Azure Active Directory : *Microsoft's multi-tenant cloud based directory and identity management service. It is a powerful identity and access management service (IDaaS) for on-premises and cloud-based apps and provides administrators with the ability to manage end user identities and access privileges.*

Azure Active Directory (Azure AD) simplifies authentication for developers by providing identity as a service, with support for industry-standard protocol **OAuth 2.0.**
The various components of the authentication process can be represented as such:



An application that wants to outsource authentication to Azure AD **must be registered in Azure AD**, which registers and uniquely **identifies the app in the directory**.

Azure AD is the **identity provider**, responsible for **verifying the identity** of users and applications that exist in an organization's directory, and ultimately **issuing security tokens** upon successful authentication of those users and applications.
Once a user has been authenticated, the application must **validate the user's security token** to ensure that authentication was successful.

In the case of OAuth 2.0 protocol token security :

OAuth makes extensive use of bearer tokens. A bearer token is a lightweight security token that grants the "bearer" access to a protected resource. Bearer tokens do not have counter-interception mechanisms and **must be transported in a secure channel** such as transport layer security (**HTTPS**).

PRIORITY : REGISTERING THE APPLICATION WITH AZURE AD

- Any application that outsources authentication to Azure AD must be registered in a directory
- Provide Azure AD with information about the application, including :
    - the URL where it's located
    - the URL to send replies after authentication
    - the URI to identify your application

This is to ensure that Azure AD can communicate with the application when handling sign on or exchanging tokens.

Application ID URI: The identifier for an application. This value is sent to Azure AD during authentication to indicate which application the caller wants a token for.

Reply URL and Redirect URI: *In the case of a web API or web application*, the Reply URL is the location to which Azure AD will send the authentication response, including a token if authentication was successful. *In the case of a native application*, the Redirect URI is a unique identifier to which Azure AD will redirect the user-agent in an OAuth 2.0 request.

→ Determine whether we are dealing with a web application or a native application

Client ID and Key: The ID for an application, which is generated by Azure AD when the application is registered. When requesting an authorization code or token, the client ID and key are sent to Azure AD during authentication.

There are **two categories** of applications that can be developed and integrated with Azure AD:

Single tenant application: A single tenant application is intended for use in one organization.

Multi-tenant application: A multi-tenant application is intended for use in many organizations, not just one organization.

If you are currently developing a single tenant application but want to make it available to many organizations, you can easily make changes to the application and its configuration in Azure AD to make it multi-tenant capable.

→ Determine whether we are dealing with a Single tenant application or a multi-tenant application

These are the **five primary application scenarios** supported by Azure AD:

1 - Web Browser to Web Application: A user needs to sign in to a web application that is secured by Azure AD.
2 - Single Page Application (SPA): A user needs to sign in to a single page application that is secured by Azure AD.
3 - Native Application to Web API: A native application that runs on a phone, tablet, or PC needs to authenticate a user to get resources from a web API that is secured by Azure AD.
4 - Web Application to Web API: A web application needs to get resources from a web API secured by Azure AD.
5 - Daemon or Server Application to Web API: A daemon application or a server application with no web user interface needs to get resources from a web API secured by Azure AD.

→ Determine what scenario we have to deal with

collected from

https://azure.microsoft.com/en-us/documentation/articles/active-directory-authentication-scenarios/#basics-of-authentication-in-azure-ad

https://azure.microsoft.com/en-us/documentation/articles/active-directory-protocols-oauth-code/#register-your-application-with-your-ad-tenant

https://oauth.net/

https://tools.ietf.org/html/draft-denniss-oauth-device-flow-00

https://aaronparecki.com/2012/07/29/2/oauth2-simplified