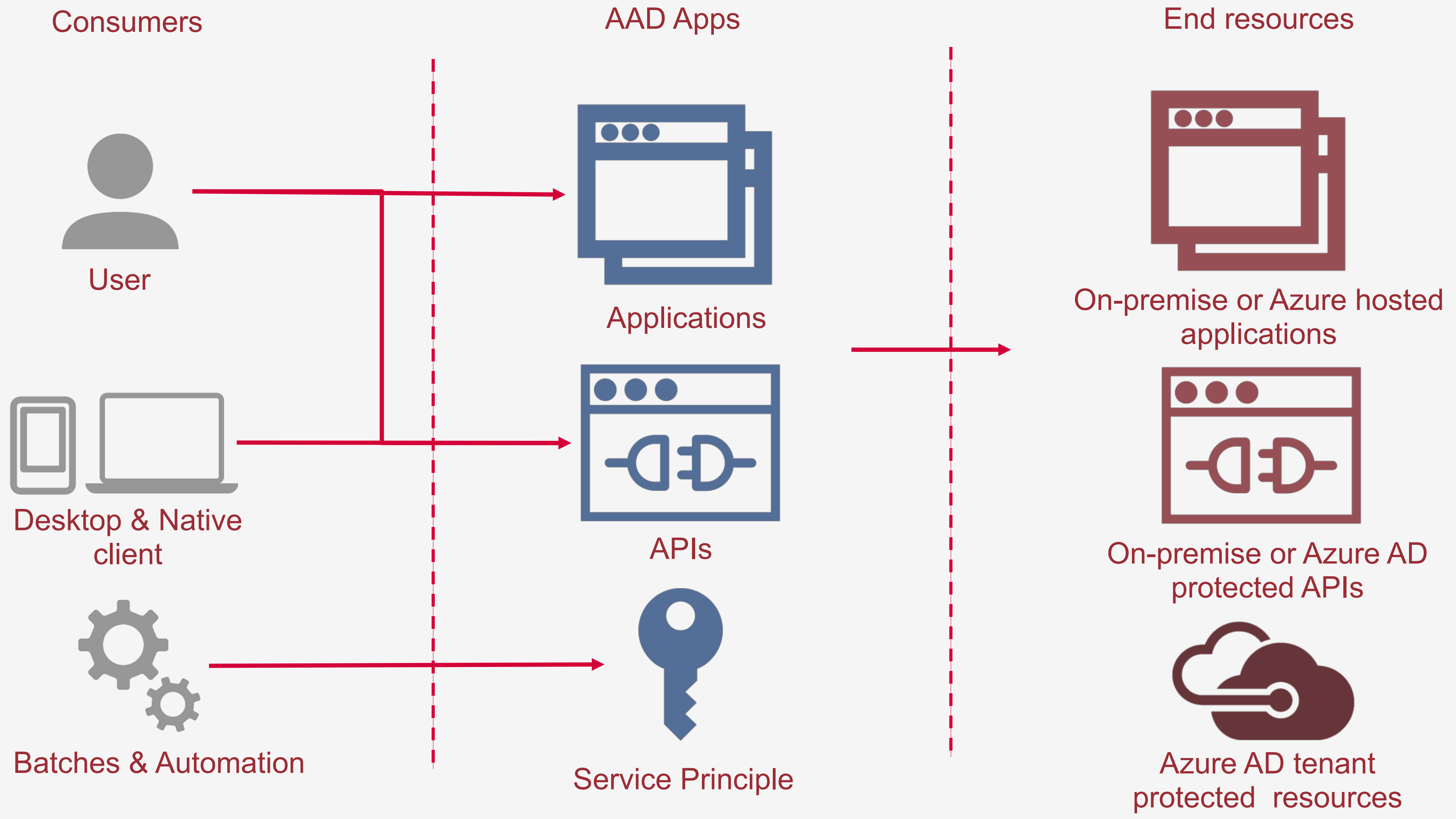
Introduction to Azure AD App Registrations

App registration types


Consumers
AAD Apps
End resources
User
Applications
On-premise or Azure hosted applications
Desktop & Native client
APIs
On-premise or Azure AD protected APIs
Batches & Automation
Service Principle
Azure AD tenant protected resources

Application scenarios supported by Azure AD

These are the five primary application scenarios supported by Azure AD

- **Single-page application (SPA) -** A user needs to sign in to a single-page application that is secured by Azure AD.
- **Web browser to web application -** A user needs to sign in to a web application that is secured by Azure AD.
- **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.
- **Web application to web API -** A web application needs to get resources from a web API secured by Azure AD.
- **Server application to web API -** A server application with no web user interface needs to get resources from a web API secured by Azure AD.

Application registration object types

When you register an Azure AD application in the Azure portal, two objects are created in your Azure AD tenant

- **Application object -** Application objects describe the application to Azure AD and can be considered as the definition of the application, allowing the service to know how to issue tokens to the application based on its settings. The application object will only exist in its home directory

- **Service principal object -** When an application is given permission to access resources in a tenant (upon registration or consent), a service principal object is created.

Types of permissions

- **Delegated permissions** - Are used by apps that have a signed-in user present. For these apps, either the user or an administrator consents to the permissions that the app requests and the app is delegated permission to act as the signed-in user when making calls to an API. Depending on the API, the user may not be able to consent to the API directly and would instead require an administrator to provide "admin consent".

- **Application permissions** - Are used by apps that run without a signed-in user present; for example, apps that run as background services or daemons. Application permissions can only be consented by an administrator because they are typically powerful and allow access to data across user-boundaries, or data that would otherwise be restricted to administrators.