ArcGIS Runtime: Authenticating Your Apps with the ArcGIS System

Nimesh Jarecha, Koushik Hajra, Xueming Wu

2021 ESRI
DEVELOPER SUMMIT

Purpose

- When is Authentication Required?
- Goals of Runtime Authentication APIs
- Authenticating a User
- Authentication Manager
- Authentication Challenge
- Credential Cache
- API key Overview

When is Authentication Required?

What is Authentication?

When is Authentication Required?

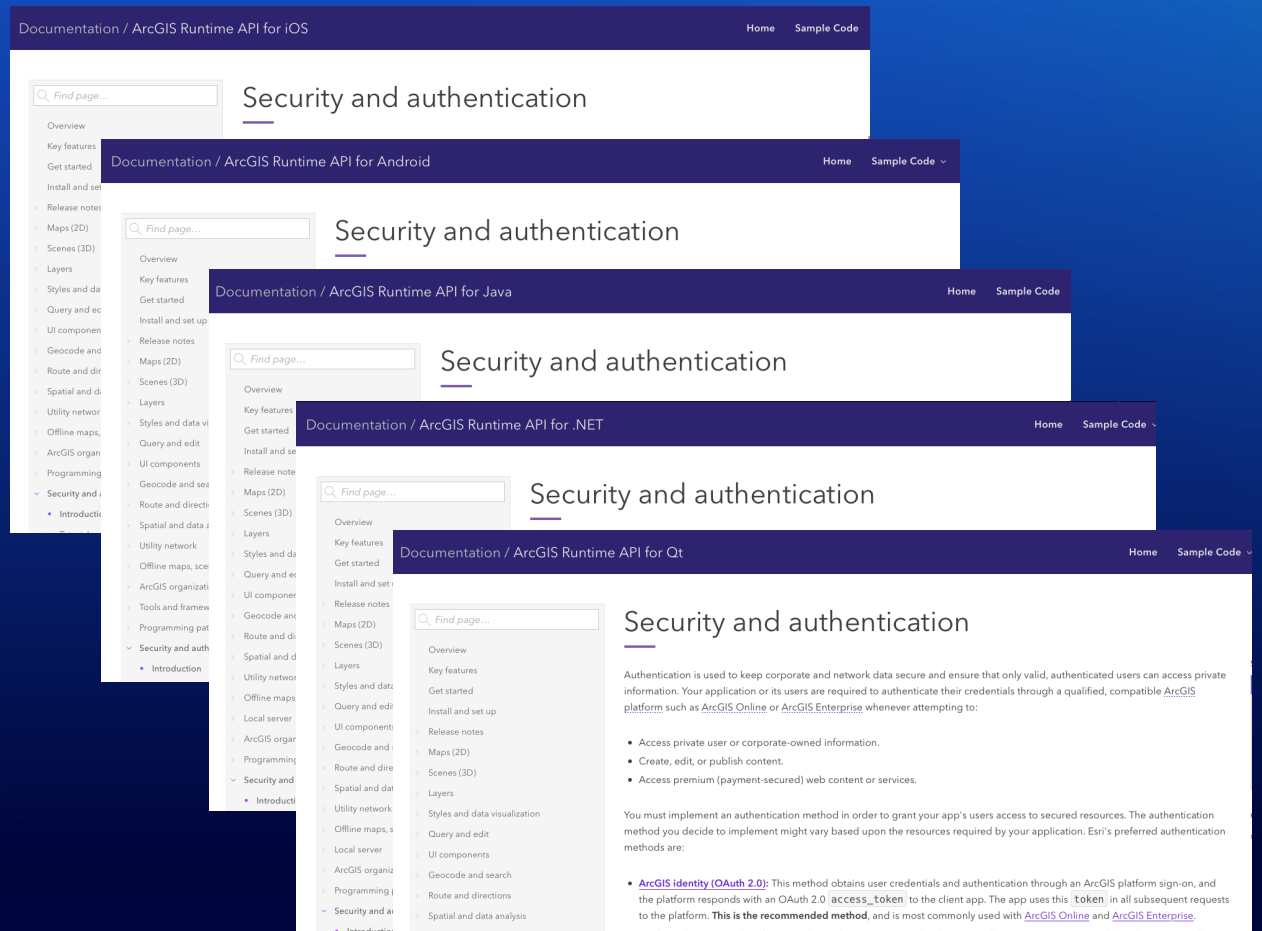
- Access a user's secured/private resource
- Add or update a user's data to server
- Access ArcGIS Platform services

Authentication Methods

- OAuth 2.0
- ArcGIS Token
- Client Certificate or Public Key Infrastructure (PKI)
- HTTP (Integrated Windows Authentication (IWA), Basic, Digest)

Goals of Runtime Authentication APIs

- Less Code
- Central Logic
  - Avoid different failure points
  - Do it once in one place
- Handle all Authentication Methods
- Common Patterns in Runtime SDKs



Runtime Authentication APIs

Class
AuthenticationManager

Properties

credentialCache: CredentialCache *read only*
OAuthConfigurations: Array<OAuthConfiguration> *read only*
trustedHosts: Array<String> *read only*

Methods

authenticationChallengeReceivedAsync(): AuthenticationChallenge
sharedAuthenticationManager(): AuthenticationManager

Enumeration
AuthenticationChallengeType

Enumeration Values

unknown (0)
usernamePassword (1)
OAuth (2)
clientCertificate (3)
untrustedHost (4)

Class
AuthenticationChallenge

Properties

authenticatingHost: String *read only*
error: Error *read only*
failureCount: int32 *read only*
portalURL: URL *read only*
proposedCredential: UserCredential *read only*
type: AuthenticationChallengeType *read only*

Methods

cancel(): void
continueWithCredential(): UserCredential
continueWithDefaultHandling(): void
isRemoteResourceFederatedWithPortalURL(portalURL: URL): bool
trustHostAndContinue(): void

Class
OAuthConfiguration

Constructors

OAuthConfiguration(portalURL: URL, clientId: String, redirectURL: String)

Properties

clientId: String *read only*
culture: String?
portalURL: URL *read only*
redirectURL: String *read only*
refreshTokenExchangeInterval: int32
refreshTokenExpirationInterval: int32
showCancelButton: bool
showSocialLogins: bool
userInterfaceStyle: UserInterfaceStyle

Class
CredentialCache

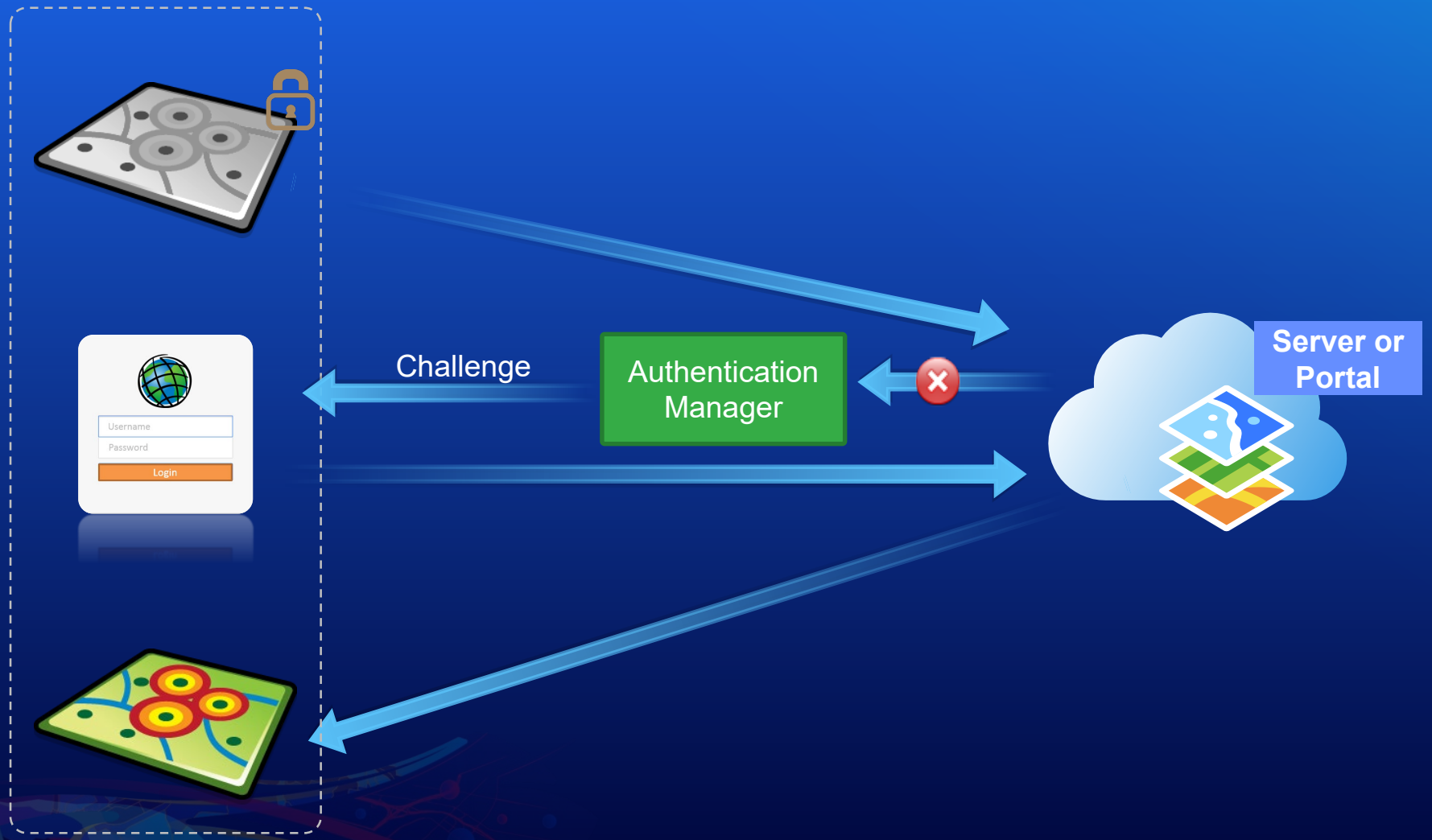
Properties

autoSyncToIdentityStore: bool *read only*
persistence: CredentialPersistence?

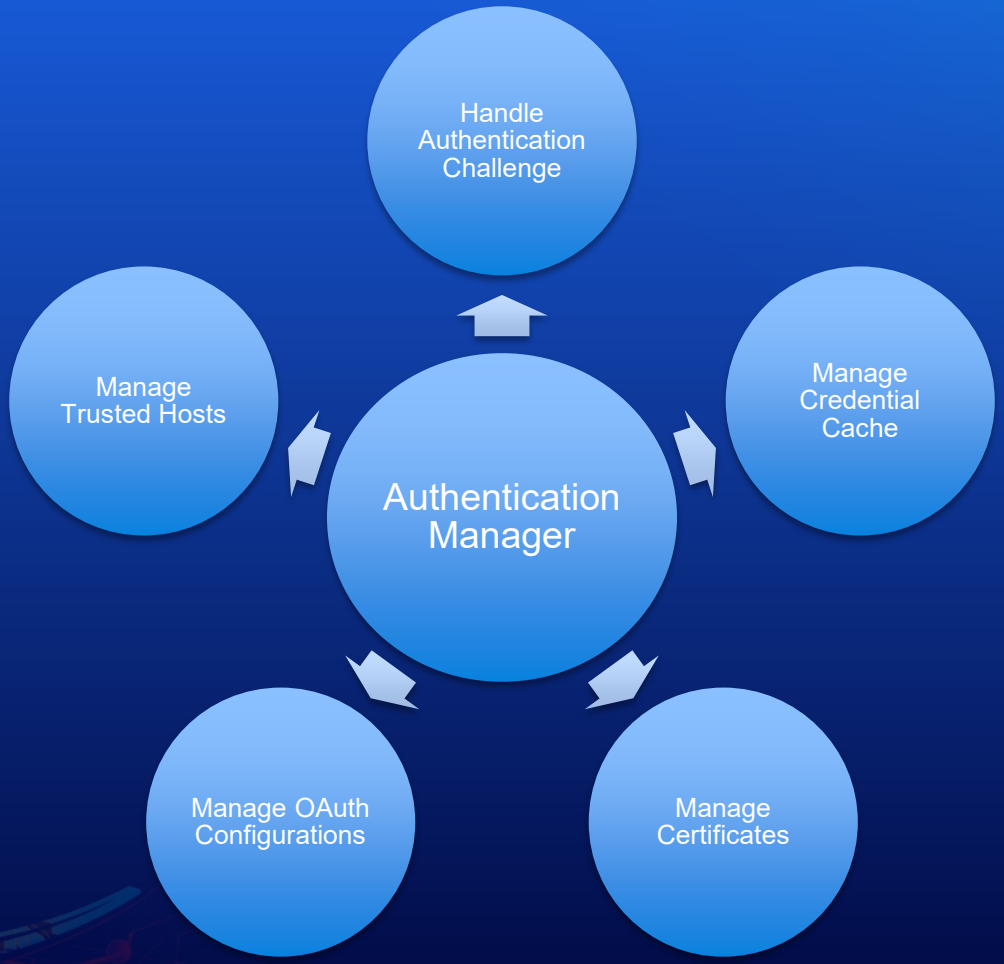
Methods

disableAutoSyncToIdentityStore(): void
enableAutoSyncToIdentityStore(identifier: String, acrossApps: String, acrossDevices: bool): void
removeAllCredentials(): void
removeAndRevokeAllCredentialsAsync(): Future<Dictionary<Credential, Error>>
removeAndRevokeCredentialAsync(credential: Credential): Future<void>
removeItemFromIdentityStoreAsync(identifier: String): void

Authenticating a User



Authentication Manager



Authentication Challenge

Authentication Challenge

- Created when server responds with one or more demands for credentials
- Has type, error, failure count, proposed credential, authenticating host etc.
- Can be controlled using RequestConfiguration

Types

- Username Password
- OAuth
- Client Certificate
- Untrusted Host

Actions

- Continue with Credentials
- Trust Host and Continue
- Cancel

Handlers

- Default
- Custom

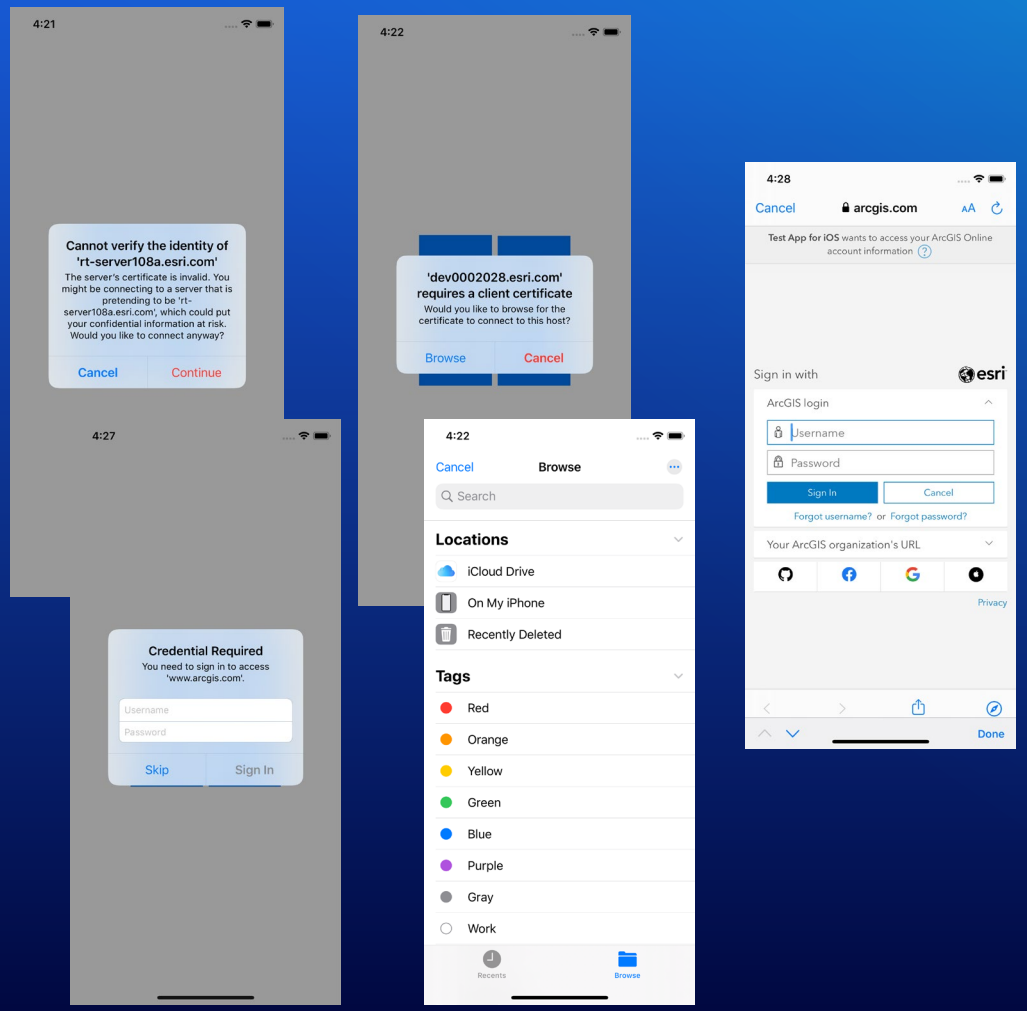
Default Authentication Challenge Handler

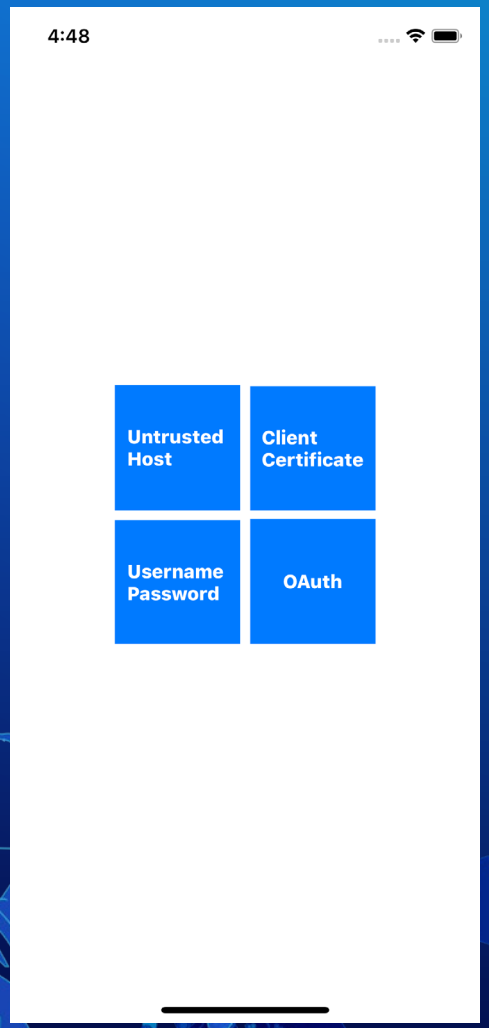
UI presented by SDK to handle different type of authentication challenges

Platform Specific Workflows

- iOS: Default
- Java and Android: Opt-In

```
val handler = DefaultAuthenticationChallengeHandler(this)
AuthenticationManager.setAuthenticationChallengeHandler(handler)
```





Default Authentication Challenge Handler

Nimesh Jarecha

Custom Authentication Challenge Handler

Platform Specific Workflows

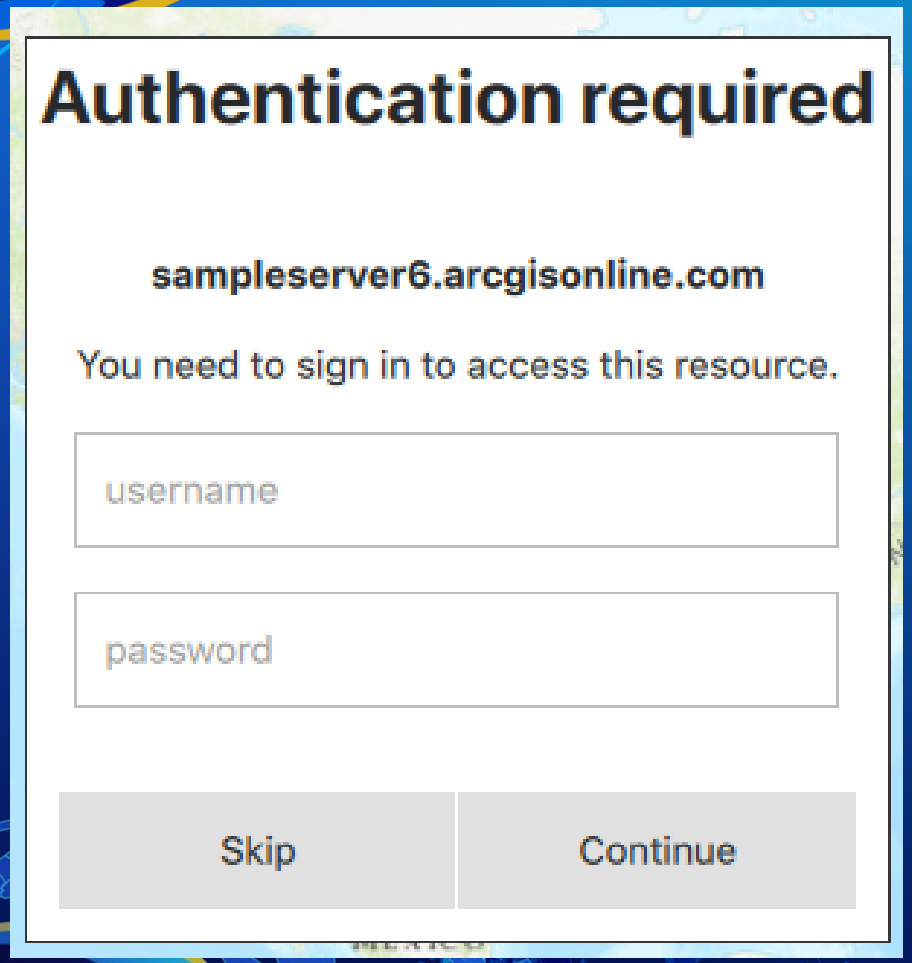
- Platform specific APIs to handle authentication challenge

Custom UI

- Take desired action on authentication challenge
- Present UI matching other Application UI

Runtime Qt Custom Handler

- Challenge handling done using toolkit
- Toolkit is open sourced



Custom Authentication Challenge Handler (Qt)

Koushik Hajra

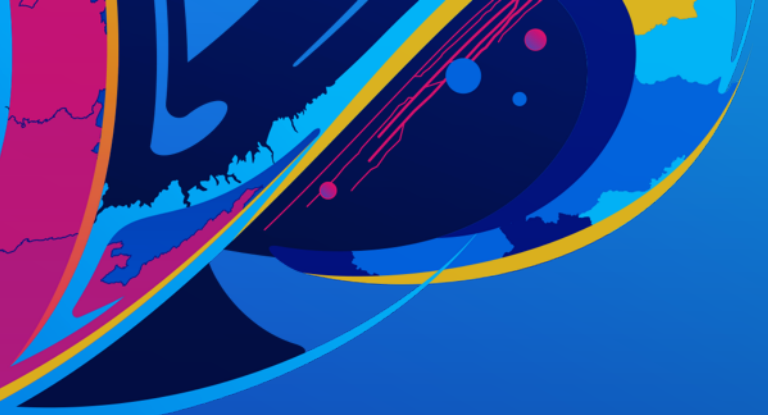
Credential Cache

In-memory Cache

- Enabled by default
- Reuse credentials for requests in the same domain
- Accessed via AuthenticationManager
- Add/Update/Remove credentials
- Revoke OAuth tokens

Persistence

- Between User Sessions
- Between Different Apps
- Between Different Devices
- Platform specific solutions
  - iOS – Keychain
  - Android – SharedPreferences, or json
  - Java - json
  - Qt - json
  - .Net – Keystore, Keychain, or Data Protection



```
var json = {
    "portalUrl": portalUrl,
    "refreshToken": refreshToken,
    "tokenServiceUrl": tokenServiceUrl,
    "oAuthClientInfoClientId": clientInfo.clientId,
    "oAuthClientInfoOAuthMode": clientInfo.oAuthMode
}

console.log(oAuthRefreshTokenFileName, clientInfo.oAuthMode);
oAuthRefreshTokenFile.writeJsonFile(oAuthRefreshTokenFileName, json)
```

Credential Cache

Xueming Wu

Offline Content Security

Unsecured Mobile Map/Scene Package (MMPK/MSPK)

Unsecured extracted geodatabase on disk

API key Overview

Access ArcGIS Platform services

- Metered Basemaps
- Geocoding & POI search
- Routing, Service Area, Closest Facility
- Demographics

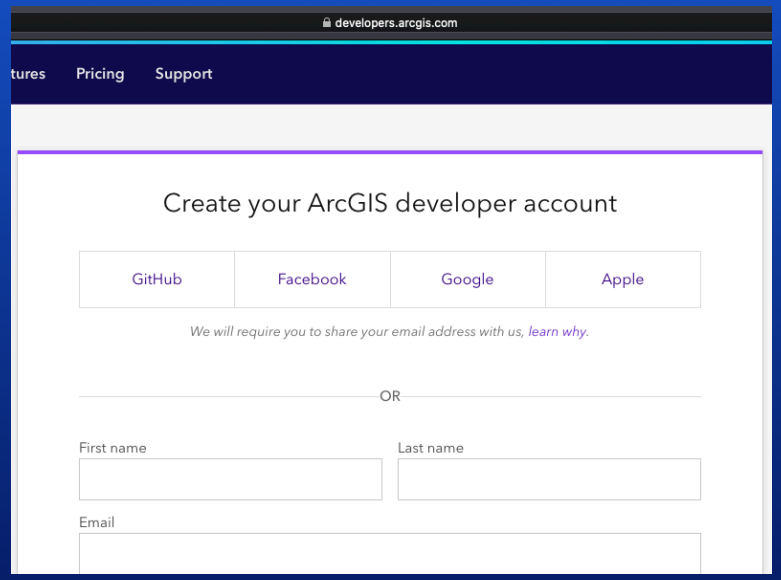
API key Benefits

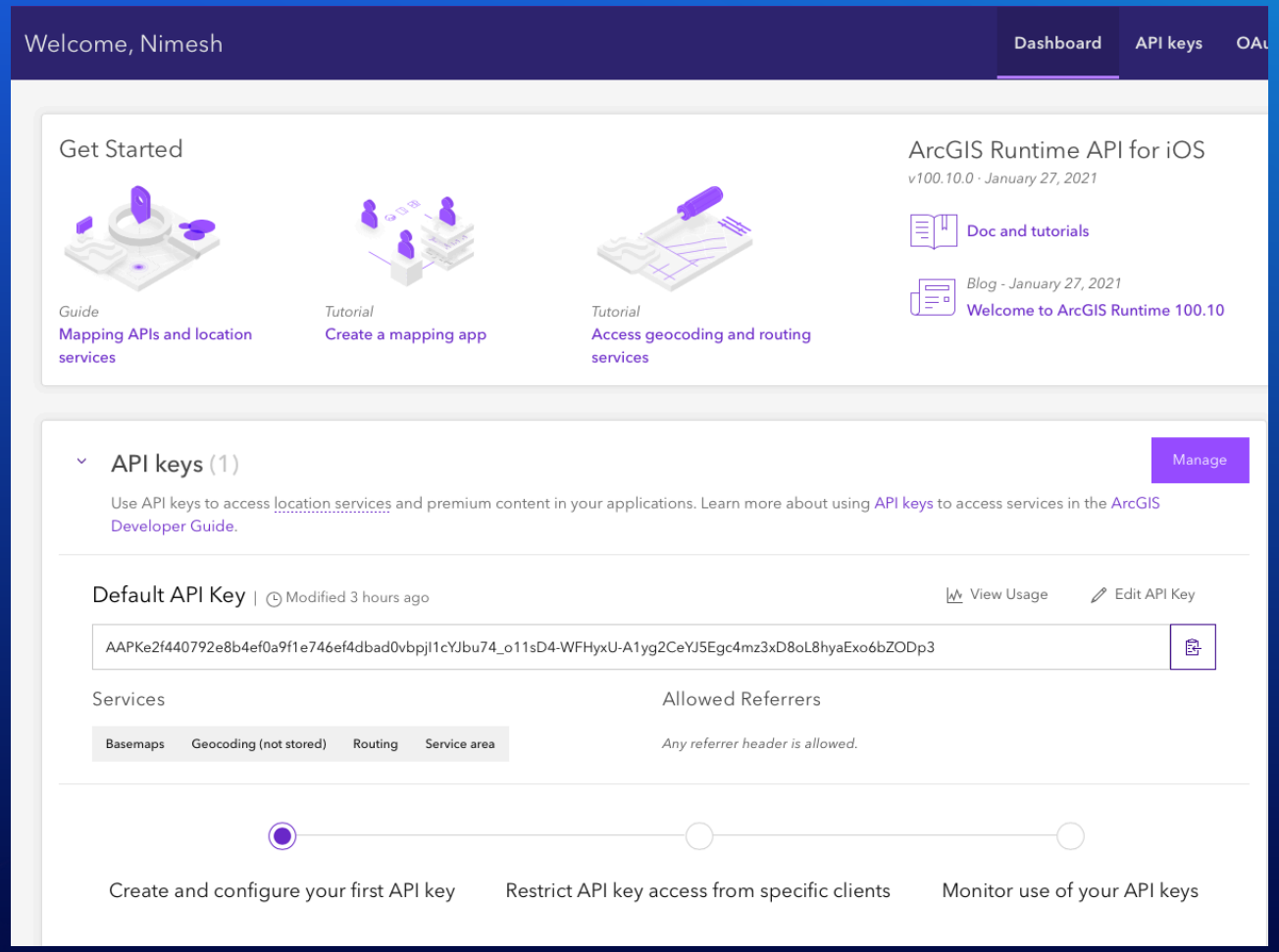
- Scope to specific services
- Monitor usage per key
- Restrict access by referrer heading
- Permanent access until API key is updated or deleted.
- Use with your own sign-in/user identity, outside of ArcGIS.

ArcGIS Developer Account

- Pay as you go access to ArcGIS Platform services
- Access to the own read only private hosted data services

ArcGIS Developer Account





Use API key with Runtime SDKs

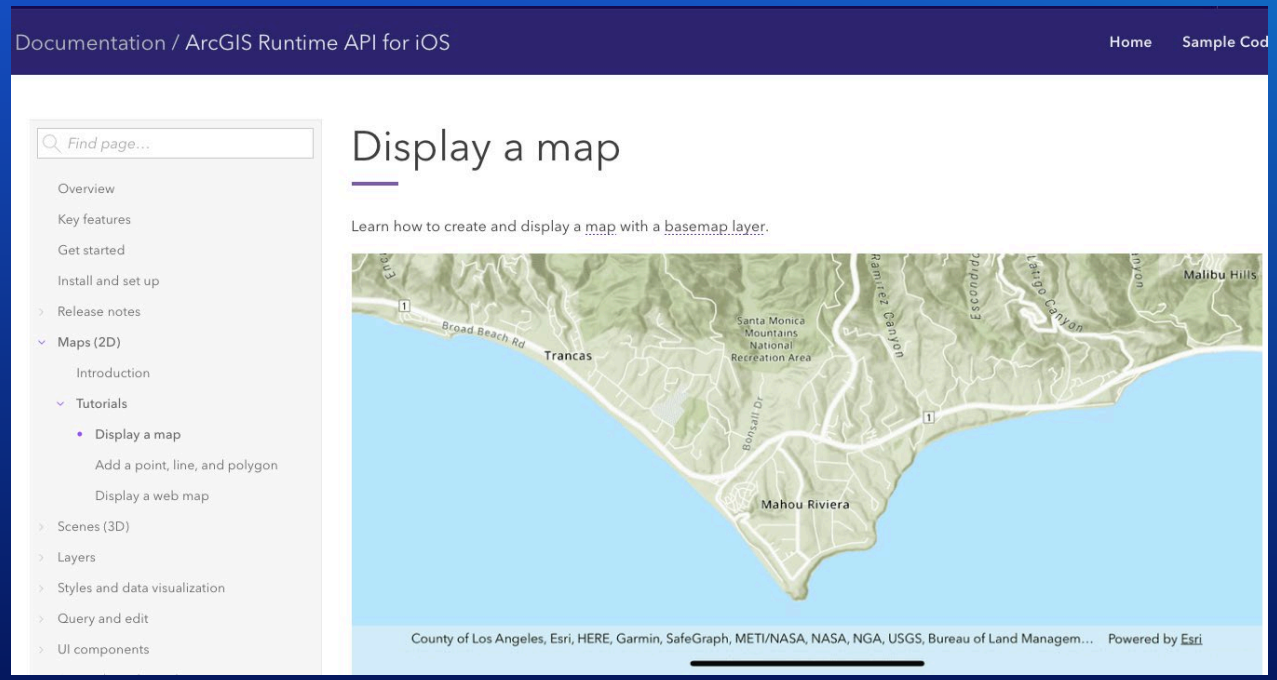
ArcGIS Runtime Environment

Layers

- ArcGIS Tiled Layer
- ArcGIS Vector Tiled Layer
- ArcGIS Scene Layer
- Service Feature Table

Tasks

- Route Task
- Locator Task
- Service Area Task
- Closest Facility Task
- Export Tile Cache Task
- Export Vector Tiles Task
- Geodatabase Sync Task



Display a Map

Nimesh Jarecha

Summary

- When is Authentication Required?
- Goals of Runtime Authentication APIs
- Authenticating a User
- Authentication Manager
- Authentication Challenge
- Credential Cache
- API key Overview

- Android: https://developers.arcgis.com/android/security-and-authentication/
- iOS: https://developers.arcgis.com/ios/security-and-authentication/
- Java: https://developers.arcgis.com/java/security-and-authentication/
- .Net: https://developers.arcgis.com/net/security-and-authentication/
- Qt: https://developers.arcgis.com/qt/security-and-authentication/

Thank you!

Copyright © 2021 Esri. All rights reserved.

Please provide your feedback for this session by clicking on the session survey link directly below the video.