

The background features a central dark blue hexagonal area with a faint grid pattern. This is surrounded by larger, overlapping hexagonal shapes in various colors: teal, orange, green, and purple. Some of these shapes have a textured, paper-like appearance, while others are solid or have a subtle gradient. The overall design is modern and geometric.

DEVELOPER SUMMIT

March 10–13

Sneak Peek

Into the next generation Android API



Presenters

Shelly Gill

Xueming Wu

Outline

- **Next release overview and goals**
- **Map, MapView, and Layers API**
- **Loadable pattern**
- **Security API**
- **Portal API**
- **Async pattern**
- **Geometry API**

Status

- Not even in beta
- Changes are inevitable
- Let us know
 - Give us feedback on everything you've seen here
 - Email us
 - sgill@esri.com, xwu@esri.com, or any one else on the Android SDK team
- Beta will be available
- More info at Road Ahead on Friday

PRE-BETA!

Overview and goals

of the next Runtime release



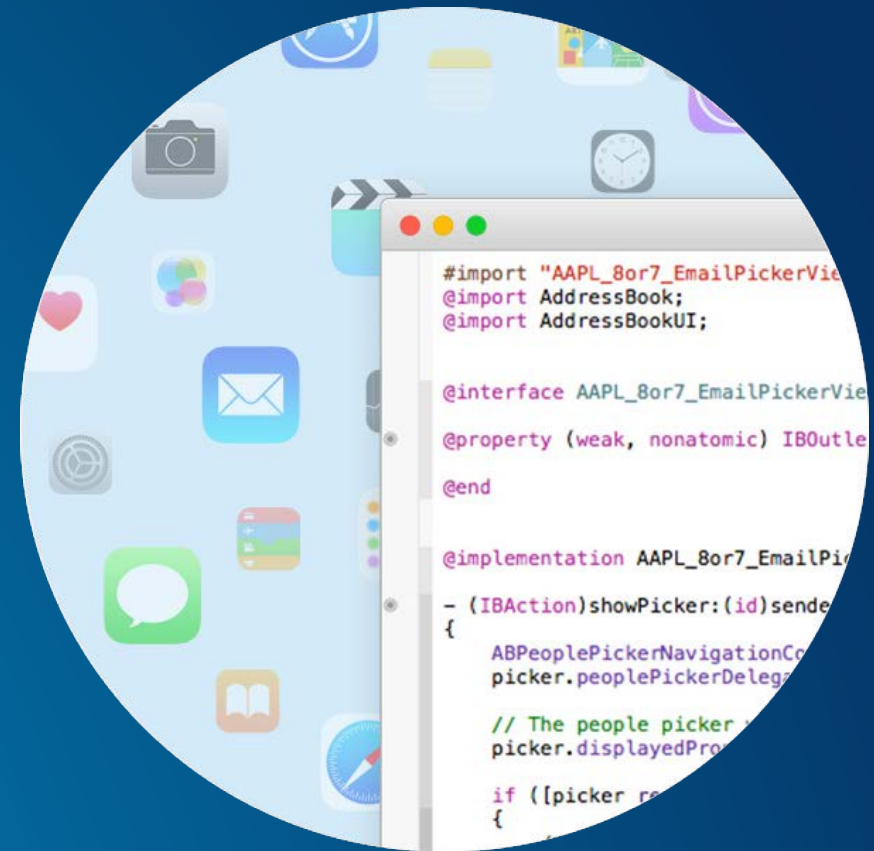
Next release

- **Major release**

- Many new capabilities
- New and changed APIs
- Improved internal architecture

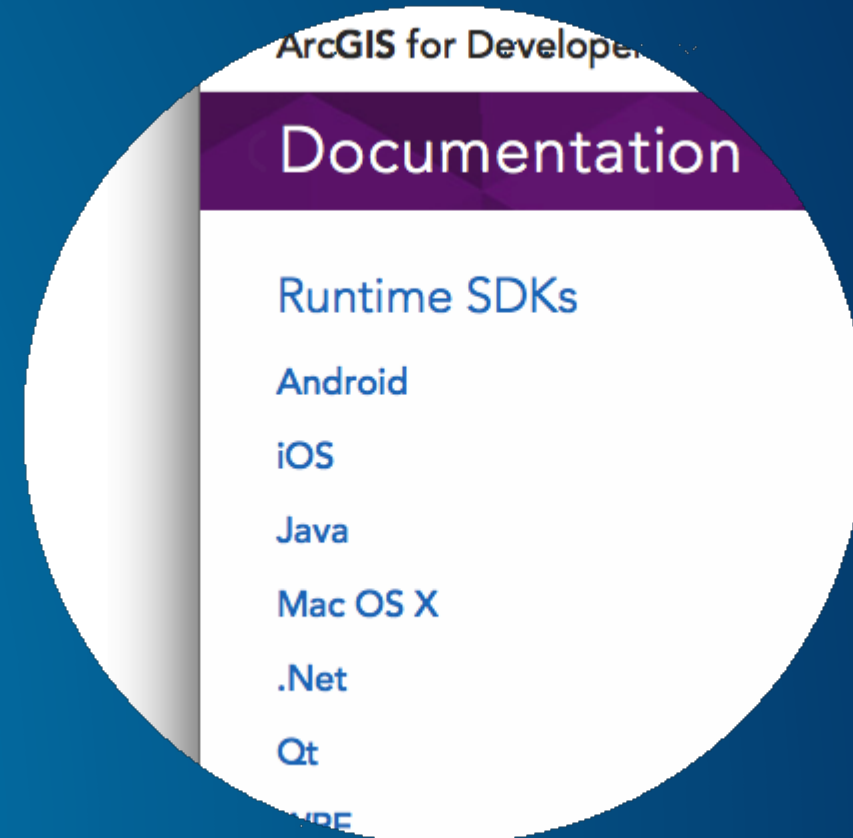
- **Goals**

- Support the ArcGIS Platform
- Move ArcGIS Engine developers to the ArcGIS Runtime
- Synchronize APIs across all ArcGIS Runtime platforms
- Support specific user workflows



Improved internal architecture offers consistent API across platforms

- **Core is growing/changing**
 - New capabilities such as 3D
- **More is moving to core**
 - e.g. map and layer class functionality
- **More consistently exposed across platforms**
 - Good if you develop apps for multiple platforms
 - Still maintaining platform specific conventions



General runtime goals for next release - 1



- **Working with maps**

- New Map object central to the API
- Create, edit and share maps across the ArcGIS Platform (Online, Portal, Pro)
- Offline maps
 - Take a map (and its layers) offline using a task
 - Author offline maps
 - Provision offline maps (from ArcGIS Pro and other Runtime Apps) with Mobile Map Packages

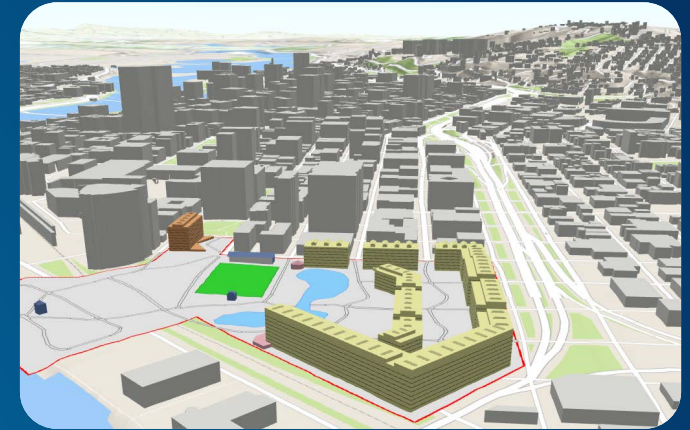
- **Working with Portals**

- Find, upload, share and configure
- Authentication made easy



General runtime goals for next release - 2

- **Working with scenes for 3D (Beta for Android)**
 - Scenes work everywhere
- **Working with layers**
 - **New layer types (some Runtimes already have these)**
 - Raster layer, scene layer (Beta), KML layer
 - **Feature layer support for local GIS data**
 - Mobile geodatabase (from ArcGIS Pro), geopackage, shapefile
 - **Richer symbology**
 - New renderers - heatmap, dot density, scale dependent, dictionary
 - New symbology model - proportional size, color, transparency
 - Advanced symbology – from ArcGIS Pro
 - **Analysis**
 - On the fly visual analysis – viewshed, line of sight
 - Raster layer with a Function Raster



What's similar or the same?

- **FeatureLayer and related classes**
- **Offline workflows**
- **Most tasks**
 - Locator, GeodatabaseSyncTask, ExportTileCacheTask, RouteTask, ClosestFacilityTask, etc
- **Local shapefile, geopackage, raster layer**
- **MapView built-in gestures**
- **SpatialReference, transformations**



Map and MapView Pattern



Map and MapView

- Content and presentation are separated
- Map is a separate class
 - Map contains layers, content
 - Edit maps and save them - portal or json
- Mapview
 - Presentation layer
 - Contains the map
 - Will help with material design apps
 - Extends `android.view.ViewGroup`
 - Viewpoints, including navigation completed handler



Layers



Layers

- **Basemap is first class citizen**
 - Map has separate collections of basemap layers and operational layers
- **Other layers mostly similar to before**
 - KMLLayer, RasterLayer get the new consistent API
- **Graphics are stored in a GraphicsOverlay**
 - It's not a layer, because it doesn't apply the same rules
 - Added to the MapView (not the Map)

Loadables

What are they and what are they used for?

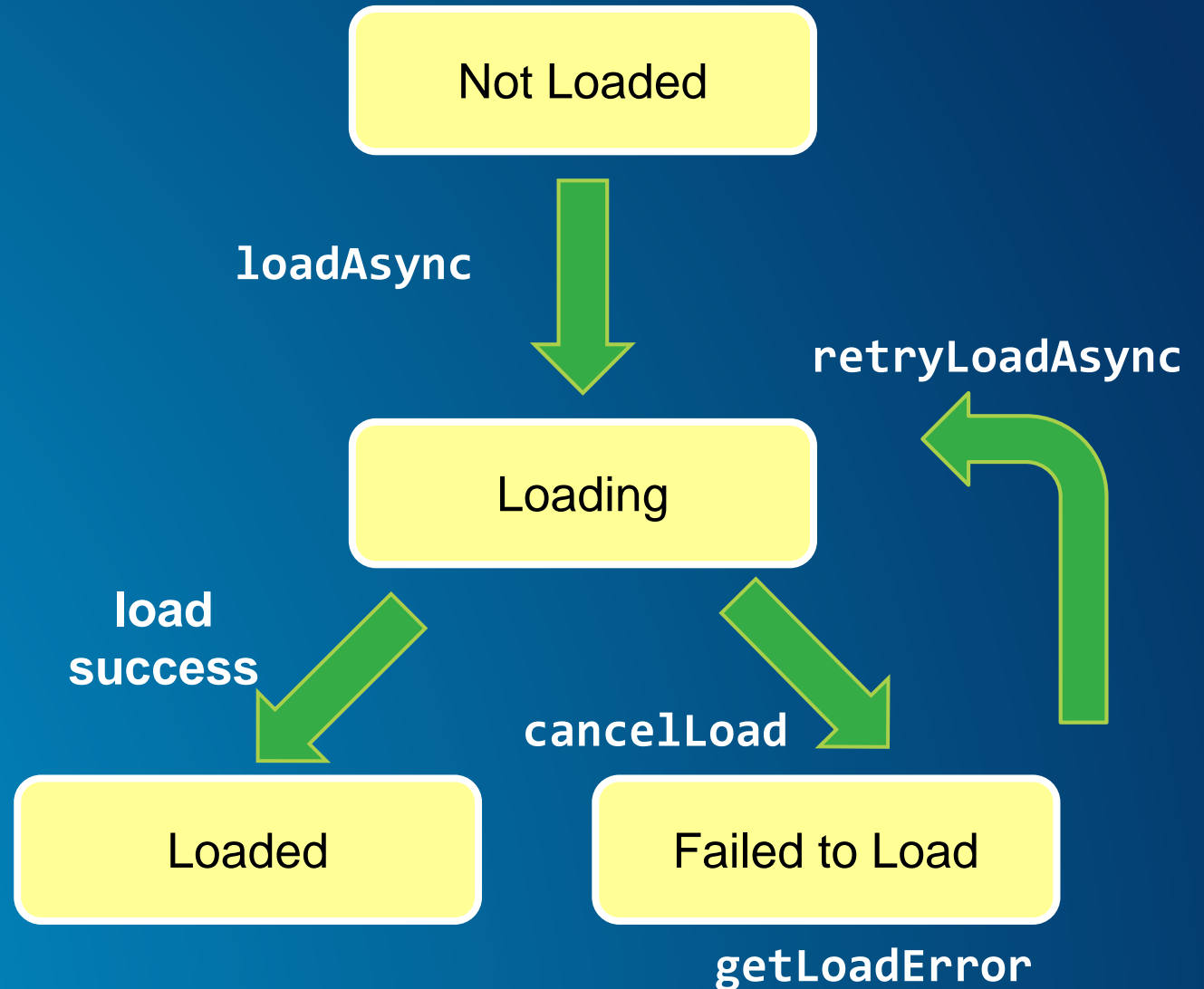


Loadables

- **Unified design for resources or metadata that take time to become initialised and communicate progress**
 - Layers, maps, tables, portal items, thumbnails, etc
- **Deal with retry**
 - network interruption, missing resource, etc
- **Load status information available**
- **Allow cancellation**

Loadable behaviour

- `getLoadStatus` reports state
 - Starts in Not Loaded
 - Moves through Loading
 - Ends in Loaded or Failed To Load
- `loadAsync` method starts the load
 - Or listens to existing calls
- `retryLoadAsync` moves back to Loading
- `cancelLoad` moves to Failed To Load
- `getLoadError` reports most recent error



Loadable

- **Load dependencies**
 - When a loadable depends on other loadable resources
 - For example, loading a Map from a PortalItem
 - The loadable calls load on its dependencies automatically
- **Handle concurrent and repeated requests for same resource**
- **Properties or methods may not be available before load completion**
 - Generally it's expected that you wait for completion on loadables
- **Overriding properties before completion**
 - Values honored once object completes
 - Start loading a map, change layer visibility, change is respected



Demo

Map, MapView, and Loadable

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
  
    // Create a local tiled layer from the TPK file.  
    ArcGISLocalTiledLayer localTiledLayer = new ArcGISLocalTiledLayer(getDataPath());  
  
    // Create a Basemap containing the local tiled layer.  
    final Basemap basemap = new Basemap(localTiledLayer);  
  
    // Create a Map using the Basemap.  
    mMap = new Map(basemap);  
  
    // Create a MapView (or use Layout).  
    mMapView = new MapView(this);  
    setContentView(mMapView);  
  
    // Bind the Map to the MapView  
    mMapView.setMap(mMap);  
  
    // No need to explicitly load the Map or Basemap. The MapView loads when it  
    // is displayed. In turn, it knows it needs to load the Map, and the Map  
    // knows to load the Basemap and any other layers it contains.
```

```
basemap.loadAsync();
basemap.addDoneLoadingListener(new Runnable() {
    @Override
    public void run() { // If already done, listener will be called immediately).
        LoadStatus loadStatus = basemap.getLoadStatus();
        if (loadStatus == LoadStatus.LOADED) {
            // We can be sure that the properties of the Basemap are available.
            Log.i(TAG, "Basemap LoadStatus: " + loadStatus);
            Log.i(TAG, "BasemapLayer count=" + basemap.getBaseLayers().size());
            Log.i(TAG, "ReferenceLayer count=" + basemap.getReferenceLayers().size());
        } else {
            // Deal with errors that occurred during loading
            showBasemapFailedToLoadMessage();
            Throwable error = basemap.getLoadError();
            if (error instanceof CancellationException) {
                // User cancelled
            } else if (error instanceof LoadException) {
                // Deal with error...
            }
        }
    }
});
```

Feedback for us?

Do these patterns cover your workflows?

Let us know! At the showcase, email, beta program

Security



AuthenticationChallenge

- Is created when due to inability to authenticate
- Contains information such as exception, credentials, url and number of trials
- Is issued to an AuthenticationChallengeHandler
- can be turn on/off through RequestConfiguration
- Provides a default AuthenticationChallengeHandler

AuthenticationManager

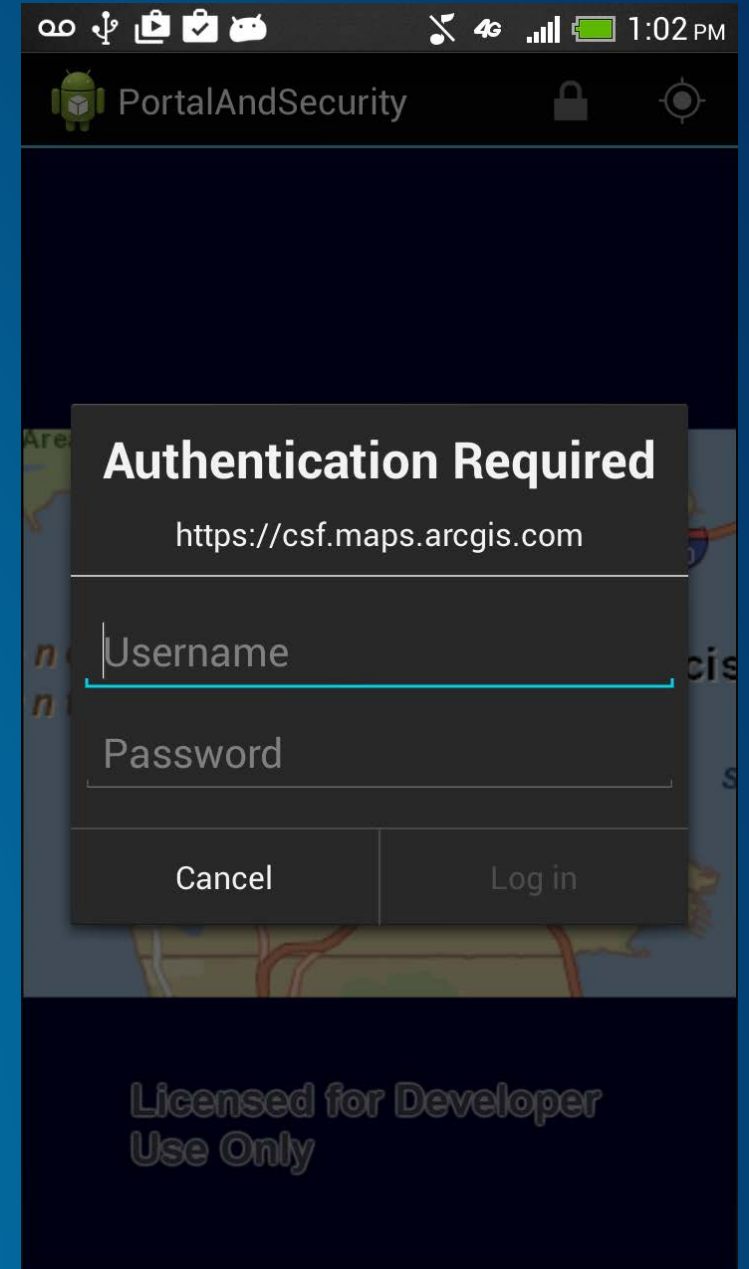
- Provides a single entry point to handle
 - AuthenticationChallenge using default or custom handler
 - self-signed certificate such as trust/clear certificate, set/clear key store
 - Credential cache: turn on/off and clear cache
 - OAuth

DefaultAuthenticationChallengeHandler

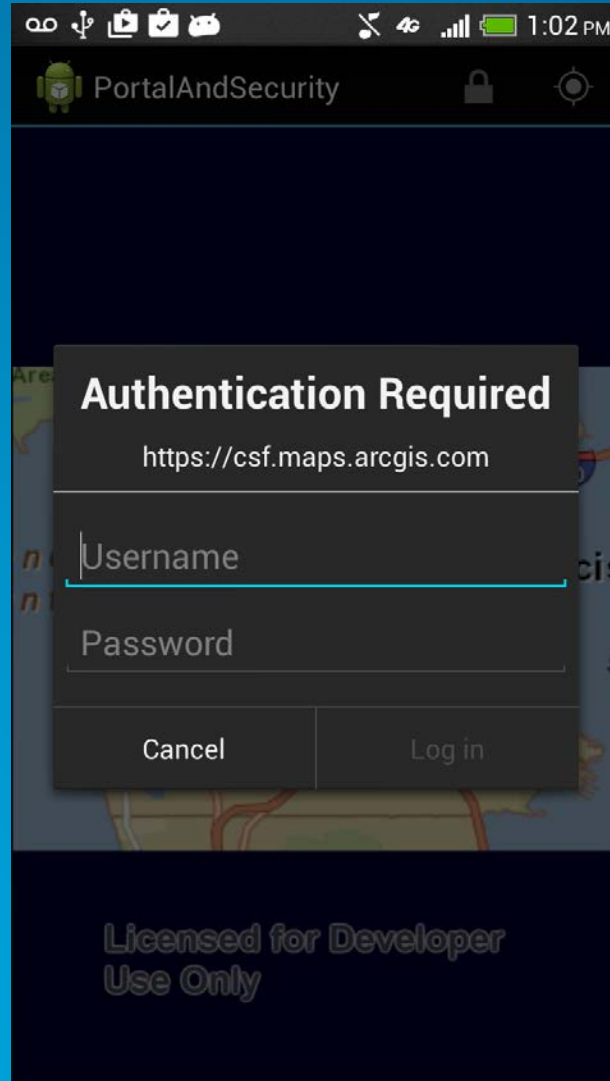
Supporting workflows:

- credentials (http, token-based)
- certificate (PKI)
- self signed certificate

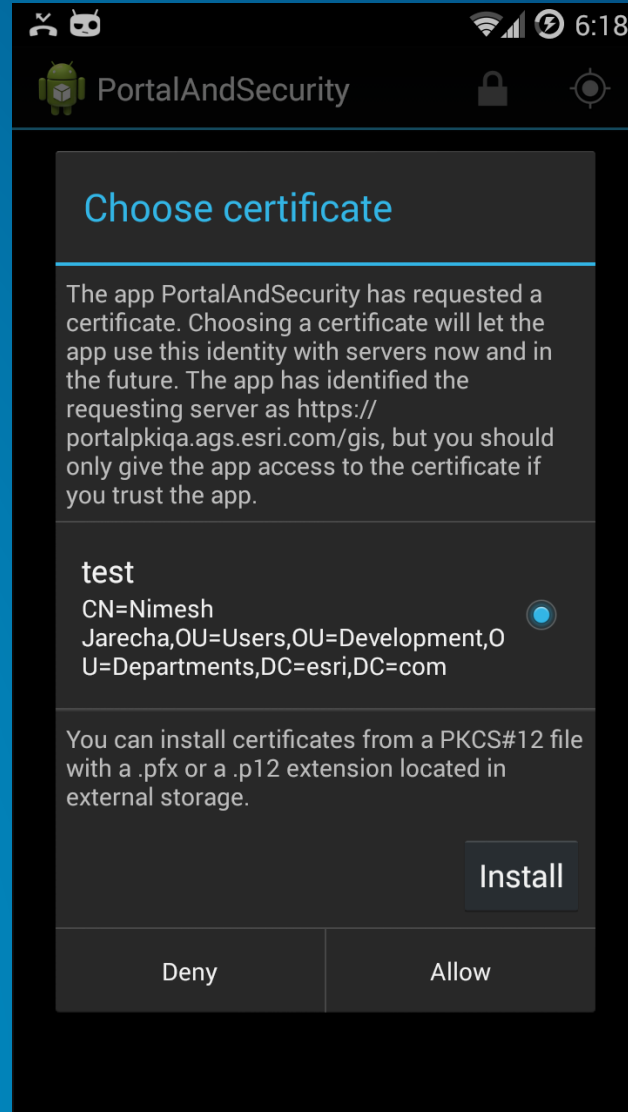
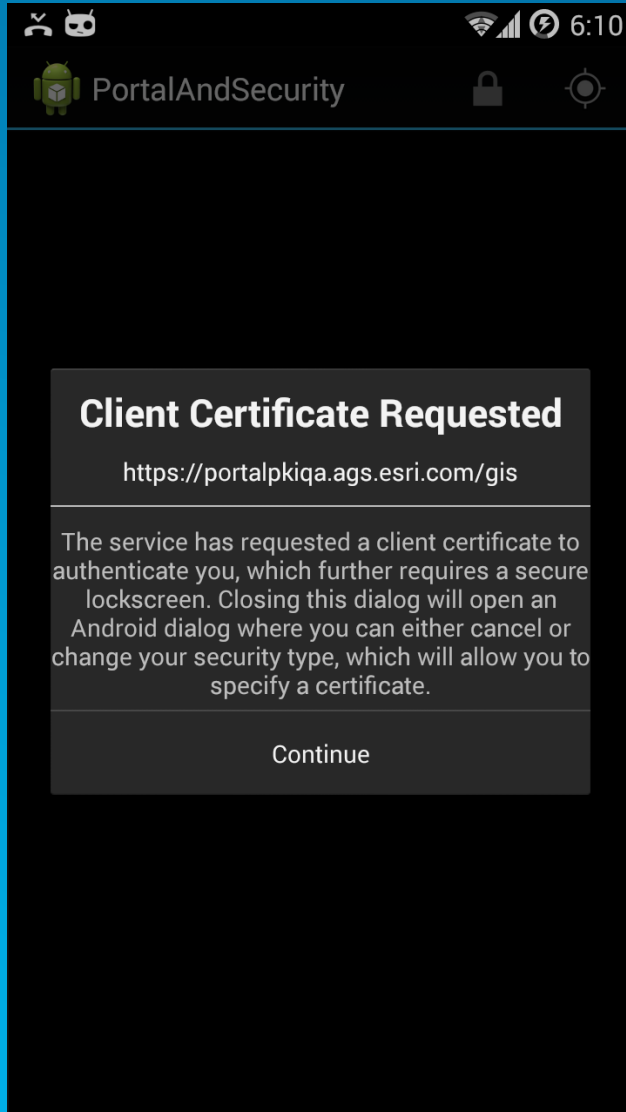
```
AuthenticationManager.setAuthenticationChallengeHandler(  
    new DefaultAuthenticationChallengeHandler(this));
```



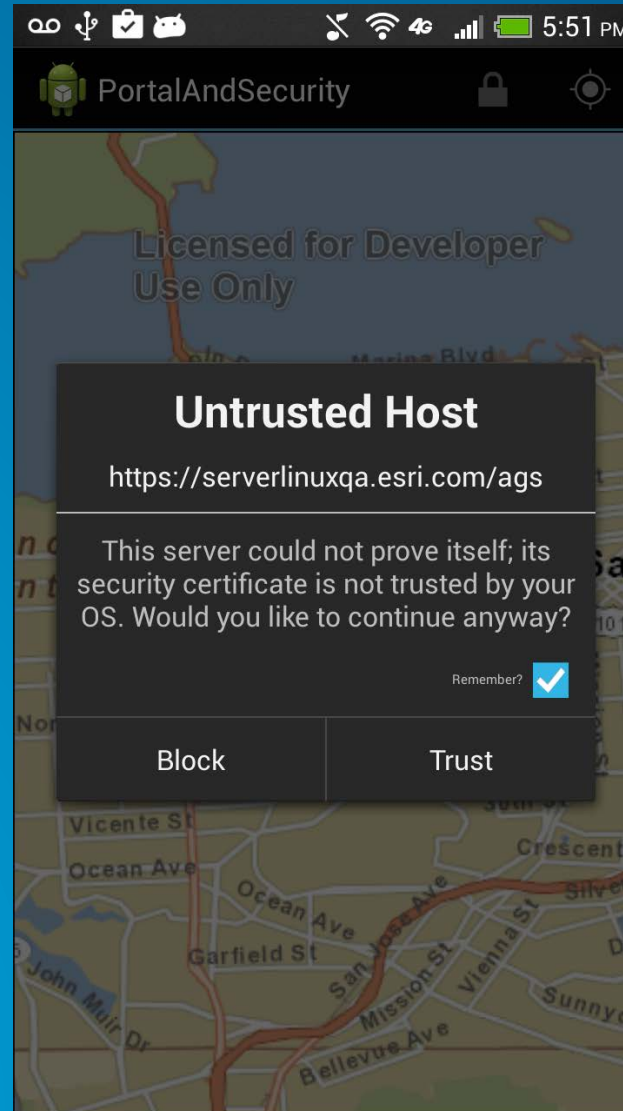
DefaultAuthenticationChallengeHandler - credentials



DefaultAuthenticationChallengeHandler - certificate



DefaultAuthenticationChallengeHandler - self-signed certificate



DefaultAuthenticationChallengeHandler - customization

```
<style name="arcgisruntime_theme_auth_dialog" parent="android:Theme.... ">  
    .....
```

DefaultAuthenticationChallengeHandler - customization

```
<resources>
  <string name="arctgisruntime_http_auth_request_dialog_title">Authentication Required</string>
  <string name="arctgisruntime_http_auth_request_dialog_username_hint">Username</string>
  <string name="arctgisruntime_http_auth_request_dialog_password_hint">Password</string>
  <string name="arctgisruntime_http_auth_request_dialog_button_positive">Log in</string>
  <string name="arctgisruntime_http_auth_request_dialog_button_negative">Cancel</string>
  <string name="arctgisruntime_self_signed_dialog_untrusted_title">Untrusted Host</string>
  <string name="arctgisruntime_self_signed_dialog_mismatched_title">Mismatched Hostname</string>
  <string name="arctgisruntime_self_signed_dialog_untrusted_host">This server could not prove itself; its security
certificate is not trusted by your OS. Would you like to continue anyway?</string>
  <string name="arctgisruntime_self_signed_dialog_mismatched_host">This server could not prove itself; its
security certificate is from another domain. This may caused by a misconfiguration or an attacker intercepting your
connection. Would you like to continue anyway?</string>
  <string name="arctgisruntime_self_signed_dialog_remember">Remember?</string>
  <string name="arctgisruntime_self_signed_dialog_button_positive">Trust</string>
  <string name="arctgisruntime_self_signed_dialog_button_negative">Block</string>
  <string name="arctgisruntime_client_cert_dialog_title">Client Certificate Requested</string>
  <string name="arctgisruntime_client_cert_dialog_message">The service has requested a client certificate to
authenticate you, which further requires a secure lockscreen. Closing this dialog will open an Android dialog where
you can either cancel or change your security type, which will allow you to specify a certificate.</string>
  <string name="arctgisruntime_client_cert_dialog_button_positive">Continue</string>
</resources>
```


Custom AuthenticationChallengeHandler

```
public interface AuthenticationChallengeHandler {  
  
    /**  
     * Handles the incoming AuthenticationChallenge, returning a response that contains an action and  
     * potentially a parameter with which to carry out the action.  
     *  
     * @param challenge the authentication challenge to handle  
     * @return the response that should be taken  
     * @since 10.3  
     */  
    public AuthenticationChallengeResponse handleChallenge(AuthenticationChallenge challenge);  
  
}
```

Portal



Portal

- **Add and delete portal items**
- **Sharing portal items**
- **Save a map to a portal**
- **Embrace loadable pattern**
- **Provides pairs of sync and async methods**

Portal - pairs of sync and async methods

```
public void initialize()
public ListenableFuture<Void> initializeAsync()
public PortalQueryResultSet<PortalItem> findItems(PortalQueryParams queryParams)
public ListenableFuture<PortalQueryResultSet<PortalItem>> findItemsAsync(
    final PortalQueryParams queryParams)
public PortalQueryResultSet<PortalGroup> findGroups(PortalQueryParams queryParams)
public ListenableFuture<PortalQueryResultSet<PortalGroup>> findGroupsAsync(
    final PortalQueryParams queryParams)
```

Portal - pairs of sync and async methods

```
new Thread(new Runnable() {
    @Override
    public void run() {
        try {
            Portal portal = new Portal(url, creds);
            portal.initialize();
            PortalInfo info = portal.getPortalInfo();
            PortalUser user = portal.getPortalUser();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}).start();
```

```
final Portal portal = new Portal(url, creds);
final ListenableFuture<Void> future = portal.initializeAsync();
future.addDoneListener(new Runnable() {
    @Override
    public void run() {
        try {
            future.get();
            PortalInfo info = portal.getPortalInfo();
            PortalUser user = portal.getPortalUser();
        } catch (InterruptedException e) {
            e.printStackTrace();
        } catch (ExecutionException e) {
            e.printStackTrace();
        }
    }
});
```

Async patterns



Why

- Execute sync on the current thread or async on background thread
- Returns result or throws exception
- Cancel an async computation

Limitation of async pattern in 10.2.x

- Two ways of retrieving result - `Future.get()` or `onCallback`
- Two ways of error handling - `Future.get()` or `onError`
- Confusion of what happens if calling `Future.cancel`

```
public interface CallbackListener<T> {  
    public abstract void onCallback(T result);  
    public abstract void onError(Throwable e);  
}
```

```
public Future<Status> initialize(final CallbackListener<Status> callback)  
  
Future initTask = table.initialize(new CallbackListener<GeodatabaseFeatureServiceTable.Status>()  
{  
  
    @Override  
    public void onError(Throwable e) {  
  
    }  
  
    @Override  
    public void onCallback(Status status) {  
  
    }  
});
```


New async pattern

- Extends Java Future with the addition of an addCompleteListener
- Can add multiple DoneListeners

```
public interface ListenableFuture<V> extends Future<V> {  
  
    /**  
     * Adds a listener to run on successful or unsuccessful completion of the asynchronous  
     * computation. The listener will  
     * run immediately if the computation is already complete when this method is called.  
     * <p>  
     * If running on Android, the listener will be run on the thread that this method is called on.  
     * Otherwise the listener  
     * will be run on the background thread the computation runs on, unless the computation is  
     * already complete in which  
     * case it runs on the current thread.  
     *  
     * @param listener a Runnable to run when the computation is done  
     * @throws IllegalArgumentException if listener is null  
     * @since 10.3  
     */  
    void addCompleteListener(Runnable listener);  
}
```

Asyn pattern - pairs of sync and async methods

```
public void initialize()
public ListenableFuture<Void> initializeAsync()
public PortalQueryResultSet<PortalItem> findItems(PortalQueryParams queryParams)
public ListenableFuture<PortalQueryResultSet<PortalItem>> findItemsAsync(
    final PortalQueryParams queryParams)
public PortalQueryResultSet<PortalGroup> findGroups(PortalQueryParams queryParams)
public ListenableFuture<PortalQueryResultSet<PortalGroup>> findGroupsAsync(
    final PortalQueryParams queryParams)
```

Asyn pattern - getting the result

- `ListenableFuture.get()`

```
final Portal portal = new Portal(url, creds);
final ListenableFuture<Void> future = portal.initializeAsync();
future.addDoneListener(new Runnable() {
    @Override
    public void run() {
        try {
            future.get();
            PortalInfo info = portal.getPortalInfo();
            PortalUser user = portal.getPortalUser();
        } catch (InterruptedException e) {
            e.printStackTrace();
        } catch (ExecutionException e) {
            e.printStackTrace();
        }
    }
});
```

Asyn pattern - cancelling a computation

- `ListenableFuture.cancel()`
 - task has not started: task never starts; `cancel()` returns true; done listener is run; `get()` throws exception.
 - task has started not completed: interrupt request is made; `cancel()` returns true; done listener is run; `get()` throws exception.
 - task has completed: `cancel()` returns false; done listener have been run; `get()` returns result.

Asyn pattern - threads of DoneListener

- Runs on the same thread of `ListenableFuture.addDoneListener()`
- One exception: a computation is cancelled, the same thread that call `ListenableFuture.cancel()`

Feedback for us?

Do you need async methods?

Do you use AsyncTask?

Have you used Futures?

What about ListenableFuture?

Do you need synchronous methods?

Do these patterns cover your workflows?

Let us know! At the showcase, email, beta program

Geometry



Geometry

- Performance and planning for the future
- Geometries are immutable
- You use builders to create and edit them
 - Typical use of builders is in interactive editing workflows (see current .NET API)
- They have a spatial reference!
- Polys are made of segments
 - So they can get curves (in future!)
 - But have point-based API helper methods

Related sessions - next release

**The Road Ahead: ArcGIS
Runtime**

Fri 8:30 – 9:30am

Primrose A

**Sneak Peek into the next
generation iOS and OS X API**

Fri 1:00-2:00pm

Primrose C/D

Related sessions - current Android SDK

Collaborate with ArcGIS Runtime SDK for Android	Weds 1:00 – 2:00 pm	Demo Theater 1 – Oasis 1
ArcGIS Runtime SDK for Android: Hit the Ground Running	Weds 2:30-3:30 pm	Smoketree A-E
Building Great Android App Uis and UXs	Weds 4:00 – 5:00 pm	Mojave Learning Center
Adding GIS to Android Wearables: What's Possible, What's a Good Idea	Thurs 10:30-11:00 am	Demo Theater 2 - Oasis 1
Android Material Design: its Impact on Location-based Apps	Thurs 3:00-3:30 pm	Demo Theater 1 - Oasis 1

Summary

- Next release goals - 3D, creating sharing maps, layer types, consistent APIs
- APIs - Map, MapView, Layers, Portal, Geometry
- Consistent async patterns across the SDK
- Loadables for predictable, consistent behaviour loading resources asynchronously
- Major release. Lots same or similar
 - *...but this will be a breaking change*
- Look out for the beta, give it a try, *and most important send us feedback!*



**Please rate this
session at**
[www.esri.com/
RateMyDevSummitSession](http://www.esri.com/RateMyDevSummitSession)

Or on the dev summit website,
go to Agenda > Session Rater

Thankyou!



Understanding our world.