2013 Esri Europe, Middle East, and Africa User Conference

October 23-25 | Munich, Germany

esri Deutschland

Smarte native Apps entwickeln

Lars Schmitz

Rainald Suchan

Herzlich Willkommen!

- Lars Schmitz, Esri Germany
  - Product Manager Developer
  - l.schmitz@esri.de
  - @pilukinum

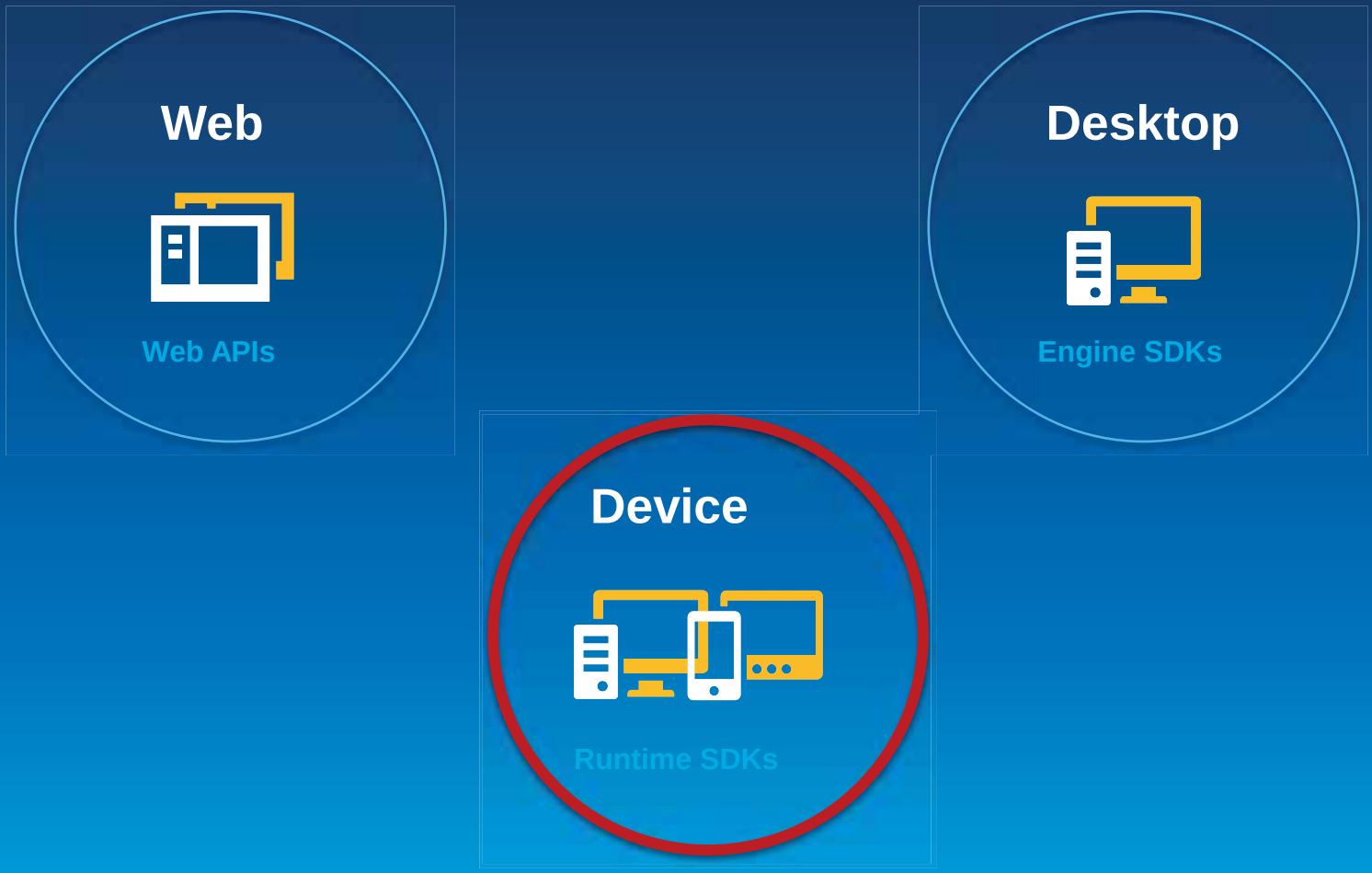
- Rainald Suchan, Esri Germany
  - Product Specialist
  - r.suchan@esri.de

Agenda

- Einführung
- Entwickeln mit ArcGIS Runtime
- Ausblick

Wer entwickelt Apps?

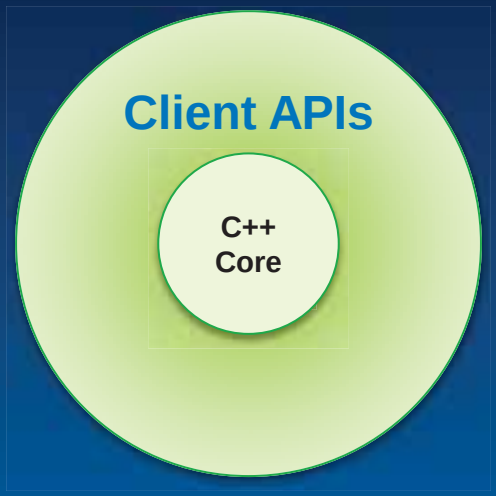
Apps gibt es auf vielen Plattformen



Runtime SDKs

- Native APIs zum Erstellen von fokussierten Anwendungen
- ArcGIS Funktionalität in bestehende Anwendungen einbetten
- Gemeinsames konzeptionelles Framework
- Powered by Runtime

Runtime SDKs



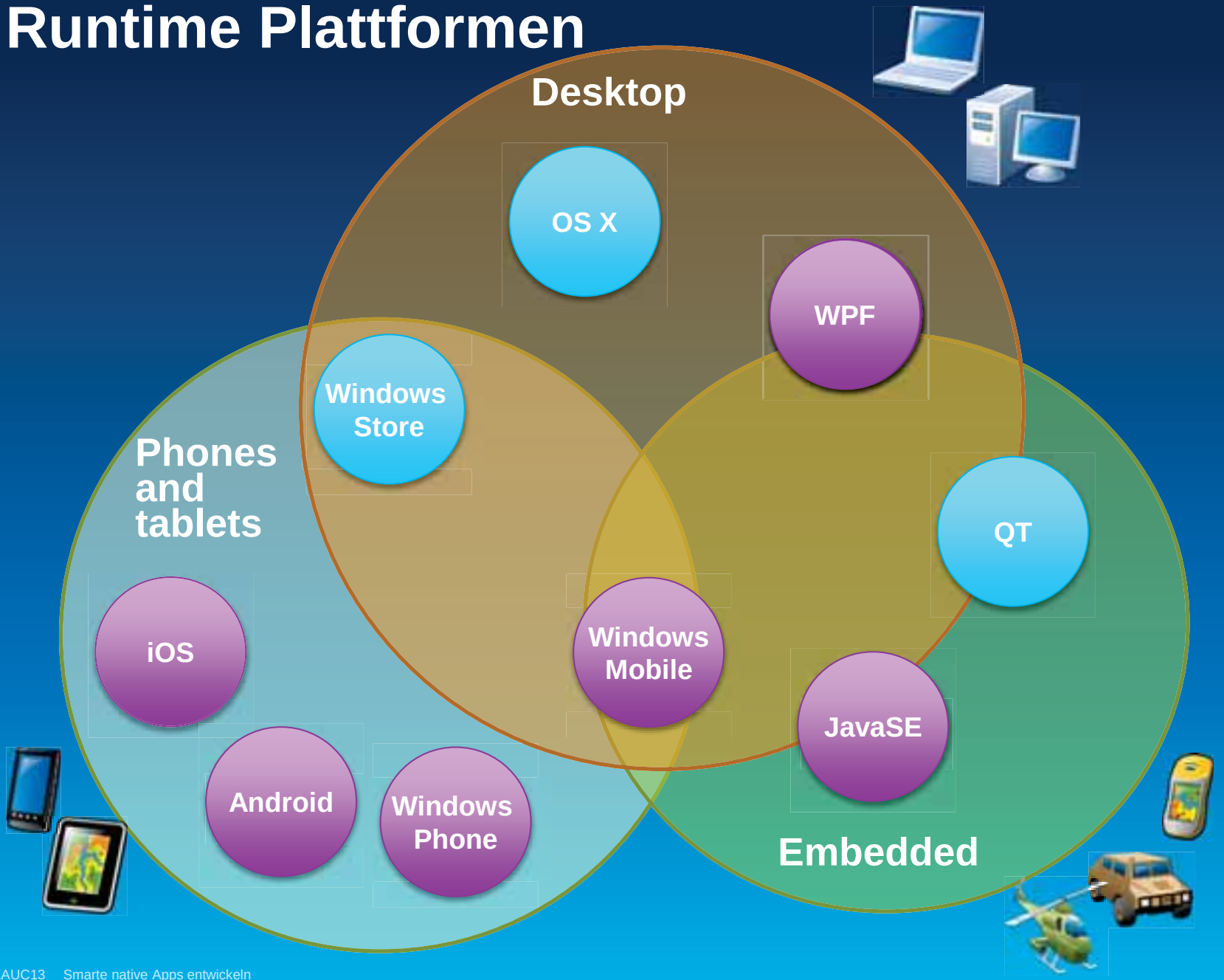
- Runtime Core
  - C++
  - Leichtgewichtig
  - Hohe Performance
- Client APIs stellen Entwicklern Funktionalität bereit, bspw.
  - .NET
  - Java
  - Objective C

Bestandteil der ArcGIS Plattform

Karten können von unterschiedlichen Geräten genutzt werden

- Karten für alle verfügbar machen

Runtime Plattformen



Apps auf Basis der ArcGIS Runtime

- ArcGIS App
  - iOS, Android, Windows Phone

- Collector App
  - iOS, Android

- Operations Dashboard for ArcGIS
  - Windows

ArcGIS Runtime SDKs für Smartphones und Tablets

- Erstellen von einfachen und fokussierten Apps
- Auch für „Nicht-GIS-Experten“
  - Consumer und Enterprise Umgebungen
- Designed für Touch Screens

Agenda

- Einführung
- Entwickeln mit ArcGIS Runtime
- Ausblick

ArcGIS Runtime SDK for Android

- Entwicklungsumgebung
  - Eclipse mit Android Developer Tools Plugin (ADT)
  - Programmiersprache: Java
- Zielplattform
  - Android Geräte
- Verteilung der Apps
  - Google Play Store und andere Android-Stores
  - App direkt auf Gerät kopieren
- Apps
  - ArcGIS for Android
  - Collector for ArcGIS

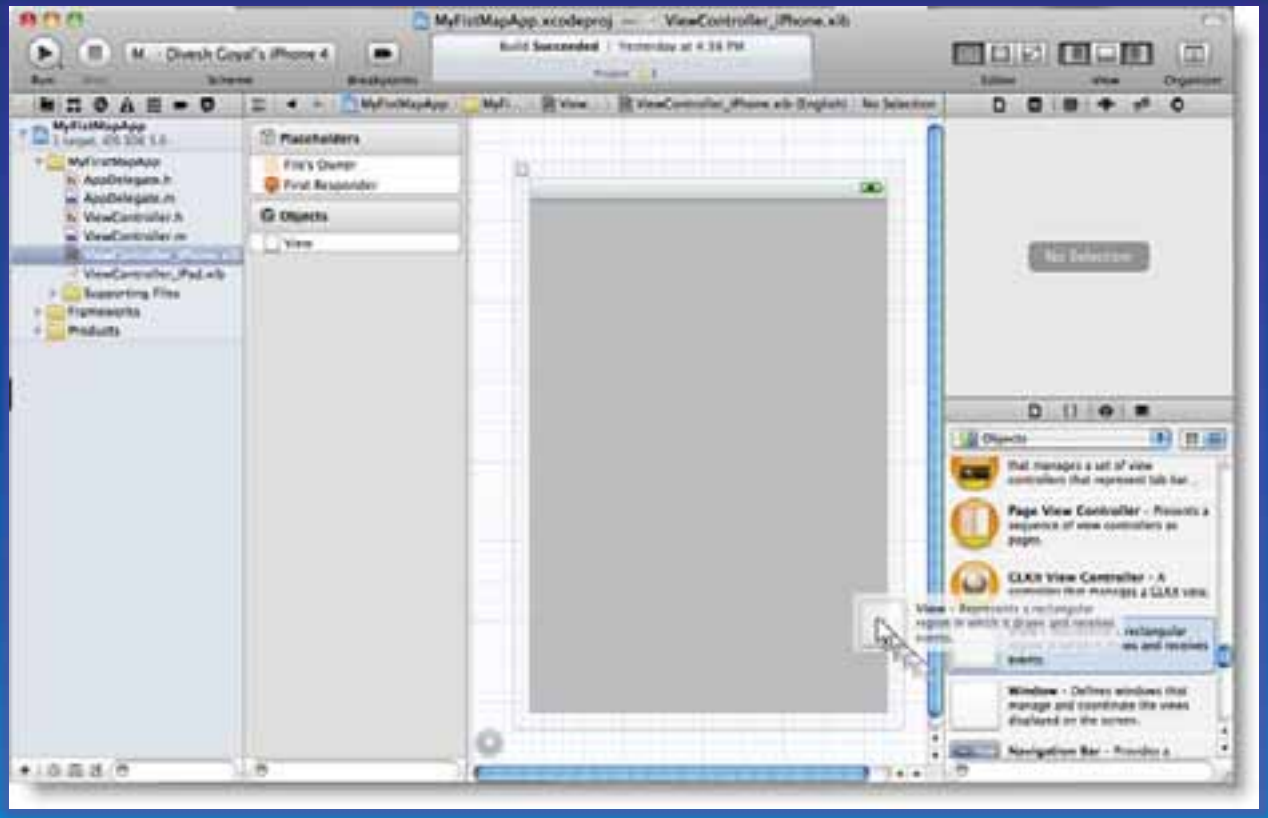
ArcGIS Runtime SDK for iOS

- Entwicklungsumgebung
  - Intel-basierter Macintosh + iOS SDK 5 oder höher
  - Programmiersprache: Objective-C
- Zielplattform
  - iPhone, iPad, iPod touch
- Verteilung der Apps
  - Apple AppStore
- Apps
  - ArcGIS for iOS
  - Collector for ArcGIS

Funktionalität – ein Überblick

- Karten und Layer
- Renderer und Symbole
- Gestensteuerung
- GPS-Unterstützung
- Geocoding und Reverse Geocoding
- Tasks wie Identify, Query, Routing, Geoprocessing
- Datenerfassung und Editieren
- Offline Daten
- Portal API

Die Karten-Komponente MapView - iOS



AGSMapView hinzufügen über Interface Builder oder im Code

Die Karten-Komponente MapView - Android

Layout-Konfiguration und Implementierung

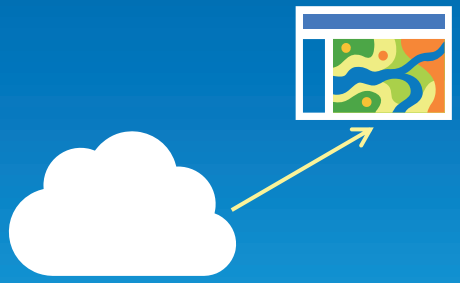
```
<com.esri.android.map.MapView
  android:id="@+id/map"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent" />
```

```
public class HelloWorld extends Activity {
  MapView map = null;

  /** Called when the activity is first created. */
  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    map = (MapView)findViewById(R.id.map);
  }
}
```

ArcGIS Runtime Datenquellen

- Layer kombinieren
  - Basemap layer
  - Operational layers
  - Graphics layer

- WebMaps anzeigen
  - ArcGIS.com
  - ArcGIS Portal



ArcGIS Runtime Datenquellen

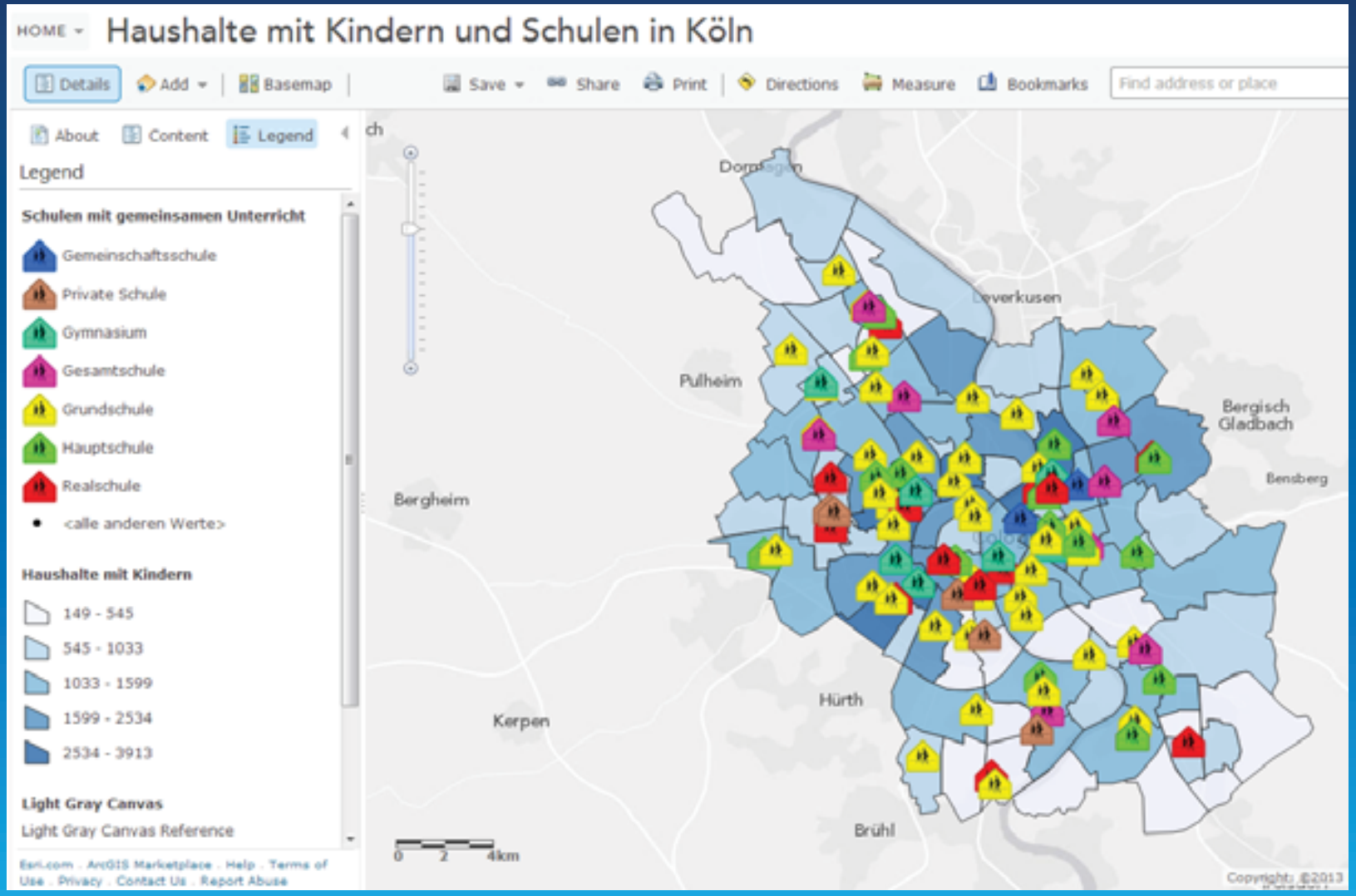
Esri Datenquellen

- ArcGIS Online
  - Karten, Services

- ArcGIS for Server
  - Services

- ArcGIS for Desktop
  - Pakete

Demo: WebMap


HOME
Haushalte mit Kindern und Schulen in Köln
Details
Add
Basemap
Save
Share
Print
Directions
Measure
Bookmarks
Find address or place
About
Content
Legend
Legend
Schulen mit gemeinsamen Unterricht
Gemeinschaftsschule
Private Schule
Gymnasium
Gesamtschule
Grundschule
Hauptschule
Realschule
<alle anderen Werte>
Haushalte mit Kindern
149 - 545
545 - 1033
1033 - 1599
1599 - 2534
2534 - 3913
Light Gray Canvas
Light Gray Canvas Reference
Esri.com . ArcGIS Marketplace . Help . Terms of Use . Privacy . Contact Us . Report Abuse
Leverkusen
Pulheim
Bergisch Gladbach
Bensberg
Bergheim
Hürth
Kerpen
Brühl
0 2 4km
Copyright: ©2013

ArcGIS Runtime Datenquellen

Externe Datenquellen

- OpenStreetMap
- WMS
- Bing Maps
- CSV
- KML

Layer zur Karte hinzufügen

Android

```
map = (MapView)findViewById(R.id.map);
ArcGISTiledMapServiceLayer tileLayer = new
ArcGISTiledMapServiceLayer("http://services.arcgisonline.com/
ArcGIS/rest/services/World_Street_Map/MapServer");
map.addLayer(tileLayer);
```

iOS

```
NSURL* url = [NSURL
URLWithString:@"http://services.arcgisonline.com/ArcGIS/rest/
services/World_Street_Map/MapServer"];
AGSTiledMapServiceLayer *tiledLayer =
[AGSTiledMapServiceLayer tiledMapServiceLayerWithURL:url];
[self.mapView addMapLayer:tiledLayer withName:@"Basemap Tiled
Layer"];
```

Kartenlayer hinzufügen

Dynamic Map Service Layer

```
map = (MapView)findViewById(R.id.map);
map.addLayer(new ArcGISDynamicMapServiceLayer(
"http://services.arcgisonline.com/ArcGIS/rest/services/
Demographics/Esri_Population_World/MapServer"));
```

Feature Service Layer

```
map.addLayer(new ArcGISFeatureLayer(url, MODE.SNAPSHOT));
```

Image Service Layer

```
map.addLayer(new ArcGISImageServiceLayer(url, null));
```

MapView-Events

Benachrichtigung, wenn Karte initialisiert ist und Layer geladen sind

```
map.setOnStatusChangedListener(new OnStatusChangedListener(){
  @Override
  public void onStatusChanged(Object source, STATUS status) {
    if(source == map && status ==
      OnStatusChangedListener.STATUS.INITIALIZED){
      // TODO
    }
    if(source == map && status ==
      OnStatusChangedListener.STATUS.LAYER_LOADED){
      // TODO
    }
  }
});
```

Layer-Events

Benachrichtigung, wenn ein Layer initialisiert ist

```
tiledLayer.setOnStatusChangedListener(new OnStatusChangedListener(){
  @Override
  public void onStatusChanged(Object source, STATUS status) {
    if(source == tiledLayer && status ==
      OnStatusChangedListener.STATUS.INITIALIZED){
      // TODO
    }
    if(source == tiledLayer && status ==
      OnStatusChangedListener.STATUS.INITIALIZATION_FAILED){
      // TODO
    }
  }
});
```

Gestensteuerung der Karte

- Event-Handling für unterschiedliche Berührungen des Displays
- Listener für die verschiedenen Events
  - SingleTab
  - DoubleTap
  - LongPress
  - DragPointer
  - PinchPointer
  - Touch

Map Touch Events - MapOnTouchListener

Public Methods	
boolean	onDoubleTap (MotionEvent point) Notified when a single-pointer-double-tap gesture occurs.
boolean	onDragPointerMove (MotionEvent from, MotionEvent to) Notified when a part of a single touch drag gesture event occurs.
boolean	onDragPointerUp (MotionEvent from, MotionEvent to) Notified when a part of a single-touch-drag gesture event occurs.
void	onLongPress (MotionEvent point) Notified when a long-press gesture occurs.
void	onMultiPointersSingleTap (MotionEvent event) Notified when a two-pointers-single-tap gesture occurs.
boolean	onPinchPointersDown (MotionEvent event) Notified when a part of a pinch gesture occurs.
boolean	onPinchPointersMove (MotionEvent event) Notified when a part of a pinch gesture occurs.
boolean	onPinchPointersUp (MotionEvent event) Notified when a part of a pinch gesture occurs.
boolean	onSingleTap (MotionEvent point) Notified when a single-pointer-single-tap gesture occurs.
boolean	onTouch (View v, MotionEvent event) Called when a touch event is dispatched to a view.

Auf Map Touch Events reagieren

```
public class MyTouchListener extends MapOnTouchListener{
  Graphic g;
  // first point clicked on the map
  Point p0 = null;

  public MyTouchListener(Context context, MapView map) {
    super(context, map);
  }

  public boolean onDragPointerMove(MotionEvent from, MotionEvent to){
    // TODO
    return true;
  }
  …
}
```

Wechsel zwischen Touch Listenern

```
/**
 * Set MyTouchListener which overrides user touch events
 */
public void setDrawTouchListener(){
  MapOnTouchListener ml = new MyTouchListener(context, map);
  map.setOnTouchListener(ml);
}

/**
 * Set the default MapOnTouchListener
 */
public void setDefaultTouchListener(){
  MapOnTouchListener ml = new MapOnTouchListener(context, map);
  map.setOnTouchListener(ml);
}
```

GPS - Positionsdaten verwenden

- Die eigene Geo-Position bestimmen
- Position kontinuierlich aktualisieren
- Karte automatisch zentrieren
- GPS-Grafik anzeigen
- Klasse LocationService

LocationService

```
map.setOnStatusChangedListener(new OnStatusChangedListener(){
  @Override
  public void onStatusChanged(Object source, STATUS status) {
    if(source == map && status ==
      OnStatusChangedListener.STATUS.INITIALIZED){
      LocationService ls = map.getLocationService();
      ls.setAutoPan(false);
      ls.setLocationListener(new LocationListener(){
        @Override
        public void onLocationChanged(Location loc) {
          // TODO
        }
      });
      ls.start();
    }
  }
});
```

Tasks

- Identify
- Query
- Find
- Geocoding
- Routing
- Service Area
- Closest Facility
- Geoprocessing

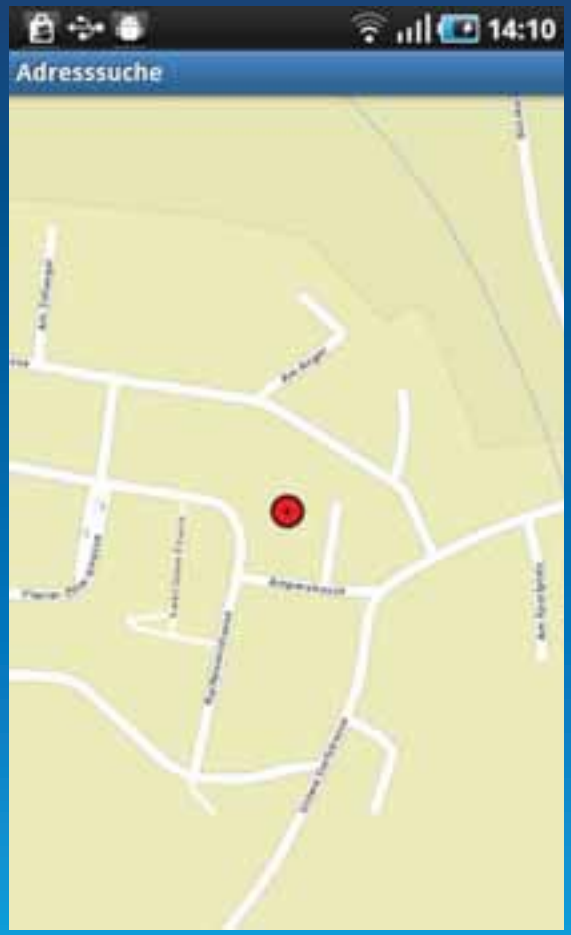
Query Task

```
public class QueryAsyncTask extends AsyncTask<String, Void, FeatureSet>{
  @Override
  protected FeatureSet doInBackground(String... arg0) {
    Query query = new Query();
    query.setWhere("AVGHHSZ_CY>3.5");
    query.setGeometry(new Envelope(-20147112.959, 557305.257,
      -6569564.719, 11753184.615));
    query.setOutSpatialReference(map.getSpatialReference());
    query.setReturnGeometry(true);
    QueryTask queryTask = new QueryTask(url);
    FeatureSet fs = null;
    try {
      fs = queryTask.execute(query);
    } catch (Exception e) {
      Log.e(TAG, e.getMessage());
    }
    return fs;
  }
}
```

Query Task – Ergebnisse darstellen

```
public class QueryAsyncTask extends
AsyncTask<String, Void, FeatureSet>{
  …

  @Override
  protected void onPostExecute(FeatureSet res){
    if (res != null) {
      Graphic[] graphic = res.getGraphics();
      if (graphic.length > 0) {
        graphicsLayer.addGraphics(graphic);
      }
    }
  }
}
```

Geocoding und Reverse Geocoding



Adresssuche

```
NSURL* url = [NSURL URLWithString: @"http://sampleserver1.arcgisonline.com/
ArcGIS/rest/services/Locators/ESRI_Geocode_USA/GeocodeServer";
AGSLocator* locator = [[AGSLocator alloc] initWithURL: url];
locator.delegate = self;
NSDictionary *address = [NSDictionary dictionaryWithObjectsAndKeys: @"380 New York St",
@"Address", @"Redlands", @"City", @"CA", @"State", @"92373", @"Zip", nil];
NSArray *outFields = [NSArray arrayWithObjects: @"Shape", @"Score", @"Match_addr",
@"Side", @"HouseNum", @"StreetName", @"StreetType", @"City","State", @"Zip", nil];
AGSSpatialReference* outSR = [AGSSpatialReference spatialReferenceWithWKID:4326 WKT:nil];
[locator locationsForAddress:address returnFields:outfields outSpatialReference:outSR];

- (void) locator:(AGSLocator*) locator operation:(NSOperation*) op
   didFindLocationsForAddress:(NSArray*) candidates {
      for (AGSAddressCandidate* candidate in candidates) {
         AGSPoint *point = candidate.location;
      }
}
```

Demo: Geocoding

Routing

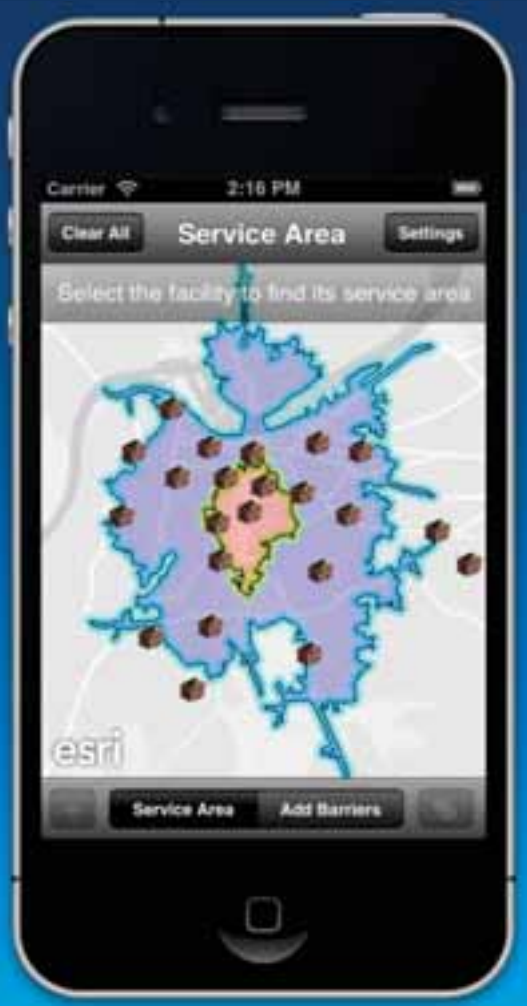
- Berechnung von Routen über Network Analyst Service
- Zwischenstops definierbar
- kürzeste / schnellste Strecke
- Optionen wie „Mautstraßen vermeiden“ „keine Fähren“ etc.
- Barrieren festlegen möglich
- Routenbeschreibung ausgeben

Routen berechnen

```
NSURL* url = [NSURL URLWithString:
@"http://tasks.arcgisonline.com/ArcGIS/rest/services/NetworkAnalysis/
ESRI_Route_NA/NAServer/Route";
AGSRouteTask* routeTask = [[AGSRouteTask alloc] initWithURL: url];
routeTask.delegate = self;
AGSRouteTaskParameters* params = [[AGSRouteTaskParameters alloc] init];
AGSStopGraphic* firstStop = [AGSStopGraphic graphicWithGeometry:
  start symbol:nil attributes:nil infoTemplateDelegate:nil];
AGSStopGraphic* lastStop = [AGSStopGraphic graphicWithGeometry:
  destination symbol:nil attributes:nil infoTemplateDelegate:nil];
[params setStopsWithFeatures:@[firstStop, lastStop]];
params.returnRouteGraphics = YES;
params.impedanceAttributeName = @"Length";
NSOperation* op = [routeTask solveWithParameters:params];

- (void) routeTask:(AGSRouteTask*)routeTask operation:(NSOperation*)op
  didSolveWithResult:(AGSRouteTaskResult*) routeTaskResult{
    self.routeResult = routeTaskResult.routeResults[0];
    self.graphicsLayer addGraphic:self.routeResult.routeGraphic;
}
```

Demo: Routing

Service Area und Closest Facility



Geoprocessing

- Umfangreiche Möglichkeiten der Geoanalyse
- Zugriff auf Geoprocessing Service
- Zahlreiche fertige Geoprocessing-Werkzeuge
- Eigene Geoprocessing-Werkzeuge erstellen und verwenden
- Geoprocessing Task synchron und asynchron ausführen

Geoprocessing Task ausführen

```
NSURL* url = [NSURL URLWithString:
@"http://sampleserver1.arcgisonline.com/
ArcGIS/rest/services/Elevation/ESRI_Elevation_World/GPServer/Viewshed"];
AGSGeoprocessor* geoprocessor = [AGSGeoprocessor
geoprocessorWithURL:url];
geoprocessor.delegate = self;
//create input parameter
AGSGPParameterValue *paramloc = [AGSGPParameterValue
parameterWithName:@"Input_Observation_Point"
type:AGSGPParameterTypeFeatureRecordSetLayer value:featureSet];
AGSGPLinearUnit *vsDistance = [[AGSGPLinearUnit alloc] init];
    vsDistance.distance = 30.0;
    vsDistance.units = AGSUnitsMiles;
AGSGPParameterValue *paramdt = [AGSGPParameterValue
    parameterWithName:@"Viewshed_Distance"
    type:AGSGPParameterTypeLinearUnit value:vsDistance];
NSArray *params = [NSArray arrayWithObjects:paramloc, paramdt, nil];
// execute the GP task
[geoprocessor executeWithParameters:params];
```

Geoprocessing Ergebnisse anzeigen

```
-(void)geoprocessor:(AGSGeoprocessor *)geoprocessor
  operation: (NSOperation *)op didExecuteWithResults:(NSArray *)results
  messages:(NSArray *)messages{
  AGSGPParameterValue *result = [results objectAtIndex:0];
  AGSFeatureSet *fs = result.value;
  for(AGSGraphic *graphic in fs.features){
    AGSSimpleFillSymbol *fillSymbol =
      [AGSSimpleFillSymbol simpleFillSymbol];
    fillSymbol.color = [[UIColor purpleColor]
      colorWithAlphaComponent:0.25];
    graphic.symbol = fillSymbol;
    //add graphic to graphics layer
    [self.graphicsLayer addGraphic:graphic];
  }
}
```

Demo: Geoprocessing

Editieren von Geodaten

- Editieren eines FeatureLayers
  - Features hinzufügen
  - Features löschen
  - Features ändern
    - Attribute ändern
    - Geometrien ändern
    - Attachments anhängen

- ArcGISFeatureLayer.applyEdits()

Feature Layer editieren

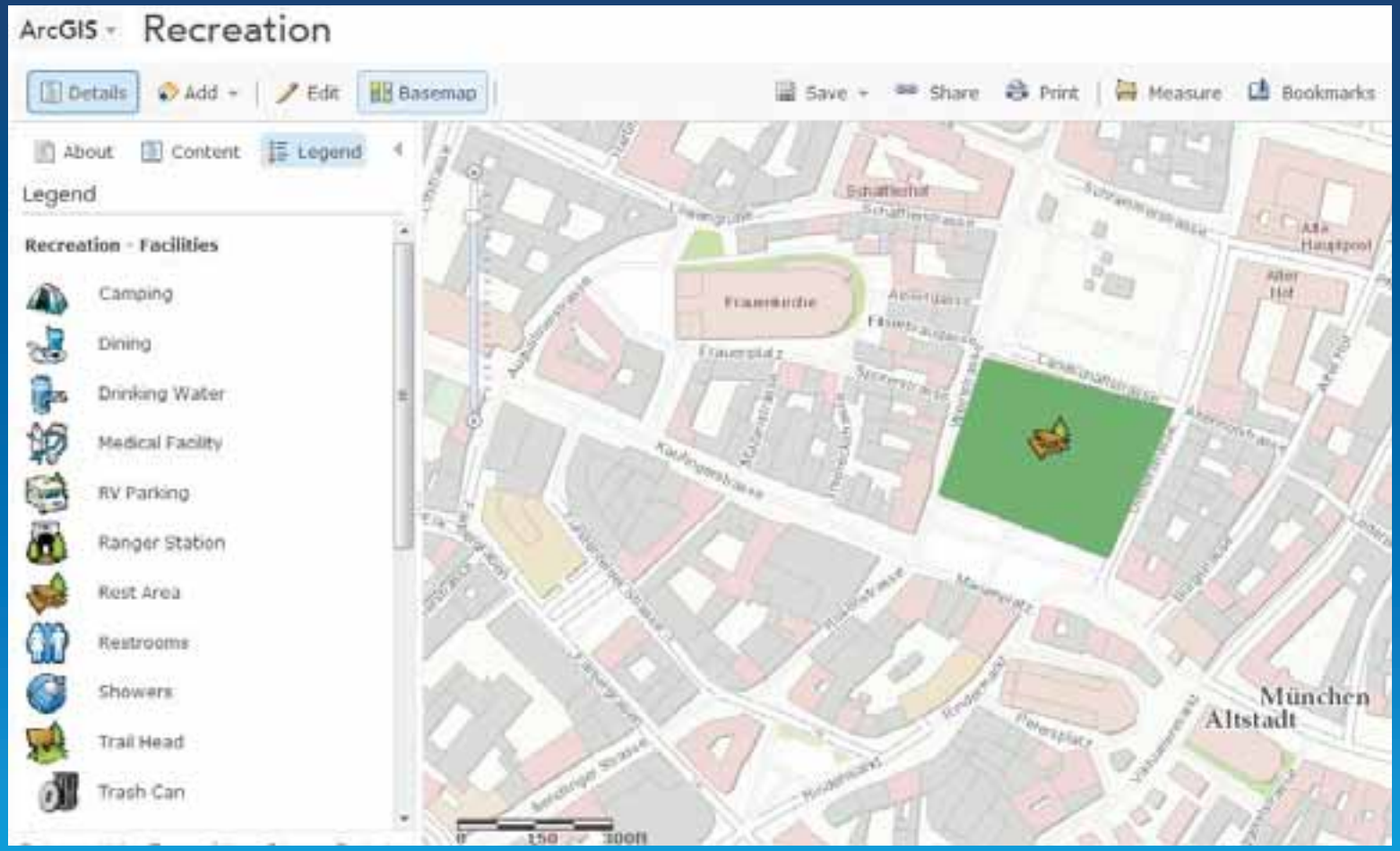
```
//create a graphic using the template
Graphic graphic = featureLayer.createFeatureWithTemplate(
    template, geometry);

featureLayer.applyEdits(new Graphic[] { graphic },
    null, null, new CallbackListener<FeatureEditResult[][]>() {

    public void onCallback(FeatureEditResult[][] editResult) {
       if (editResult[2] != null && editResult[2][0] !=
          null && editResult[2][0].isSuccess()) {
          // editResult[0] = ADD results
          // editResult[1] = DELETE results
          // editResult[2] = UPDATE results
       }
    }

    public void onError(Throwable error) {
      // TODO implement error code
    }
});
```

Attribute editieren

- Graphic-Objekt holen
- Attributwerte ändern
- ev. Attachments an Features anhängen
- applyEdits()

Geometrie editieren

Demo: Editieren



Offline Daten

- Daten können auf der SD-Karte des Gerätes gespeichert werden
  - ArcGIS Compact Cache Format
  - Tile Package (*.tpk)
  - JSON FeatureSet

```
ArcGISLocalTiledLayer baseMapLayer = new ArcGISLocalTiledLayer(
  “file:///mnt/sdcard/ArcGIS/samples/cljf/”+
  “OfflineData/ImageryTPK.tpk”);
map.addLayer(baseMapLayer);
```

Offline Feature-Daten

- Verwendung eines In-Memory FeatureLayer
  - FeatureSet (Array von Features)
  - Layer-Definition (JSON vom Feature-Service)
- JSON von SD-Karte lesen
- Features Ändern
  - Graphics hinzufügen/löschen/ändern
  - applyEdits()
- Auf SD-Karte schreiben als JSON

Offline – FeatureLayer anlegen

```
<!–strings.xml-->
<string name="config.windturbine.layer.definition">
{\"currentVersion\":10.01,\"id\":0,\"name\":\"WindTurbine\”…
</string>
```

```
//Java
ArcGISFeatureLayer featureLayer = new ArcGISFeatureLayer(
    R.string.config.windturbine.layer.definition,
    featureSet, null);
```

Offline – Features hinzufügen/editieren

```
featureLayer.addGraphics(graphics);

//or

Graphic[] graphics = new Graphic[ g ];
featureLayer.applyEdits(
    //adds graphics,
    //updates,
    //deletes
);
```

Offline – JSON auf SD-Karte schreiben

```
windTurbine.queryFeatures(query, new CallbackListener<FeatureSet>() {
  @Override
  public void onCallback(FeatureSet result) {
    if(result != null){
      FileOutputStream outstream = null;
      try {
        // create feature set as json string
        String fsstring = FeatureSet.toJson(result);
        path = createJsonFile();
        // create a File from json fully qualified path
        File outfile = new File(path);
        // create output stream to write to json file
        outstream = new FileOutputStream(outfile);
        outstream.write(fsstring.getBytes());
        outstream.close();
      } catch (Exception e) {
        //TODO
      }
    }
  }});
}
```

Offline – JSON von SD-Karte lesen

```
//Use Jackson JsonParser
JsonFactory factory = new JsonFactory();
JsonParser parser = factory.createJsonParser(
        new FileInputStream(path)
);

parser.nextToken();

FeatureSet fs = FeatureSet.fromJson(parser);
```

Portal API

- Zugang zu Portal for ArcGIS und ArcGIS Online
- Zugriff auf Inhalte, Benutzer und Gruppen des Portal
- Inhalte durchsuchen und nutzen
- Inhalte hinzufügen, ändern und löschen

Portal API – Karten suchen und anzeigen

Agenda

- Einführung
- Entwickeln mit ArcGIS Runtime
- Ausblick

Runtime SDK 10.2 Release Themen

- Verbesserte Produktivität für Entwickler
- Neue Plattformen und SDKs
- Mehr Möglichkeiten
- Besserer Zugang und neue Opportunitäten

Verbesserte Produktivität für Entwickler

- Vereinfachter Gebrauch der API
  - Weniger Code-Zeilen
  - Intuitiver
  - Weniger Parameter Klassen, mehr Strings
- Mehr asynchrone Methoden
- Application Frameworks
  - Hilfsklassen und UI Komponenten

New Platforms

- 32-bit Linux 
- iOS 7 
- OS X Mountain Lion (10.8) 
- X86 Android 

- Windows Store 

Zusätzliche SDKs

- Windows .NET
  - Desktop
  - Windows Phone
  - Windows Store

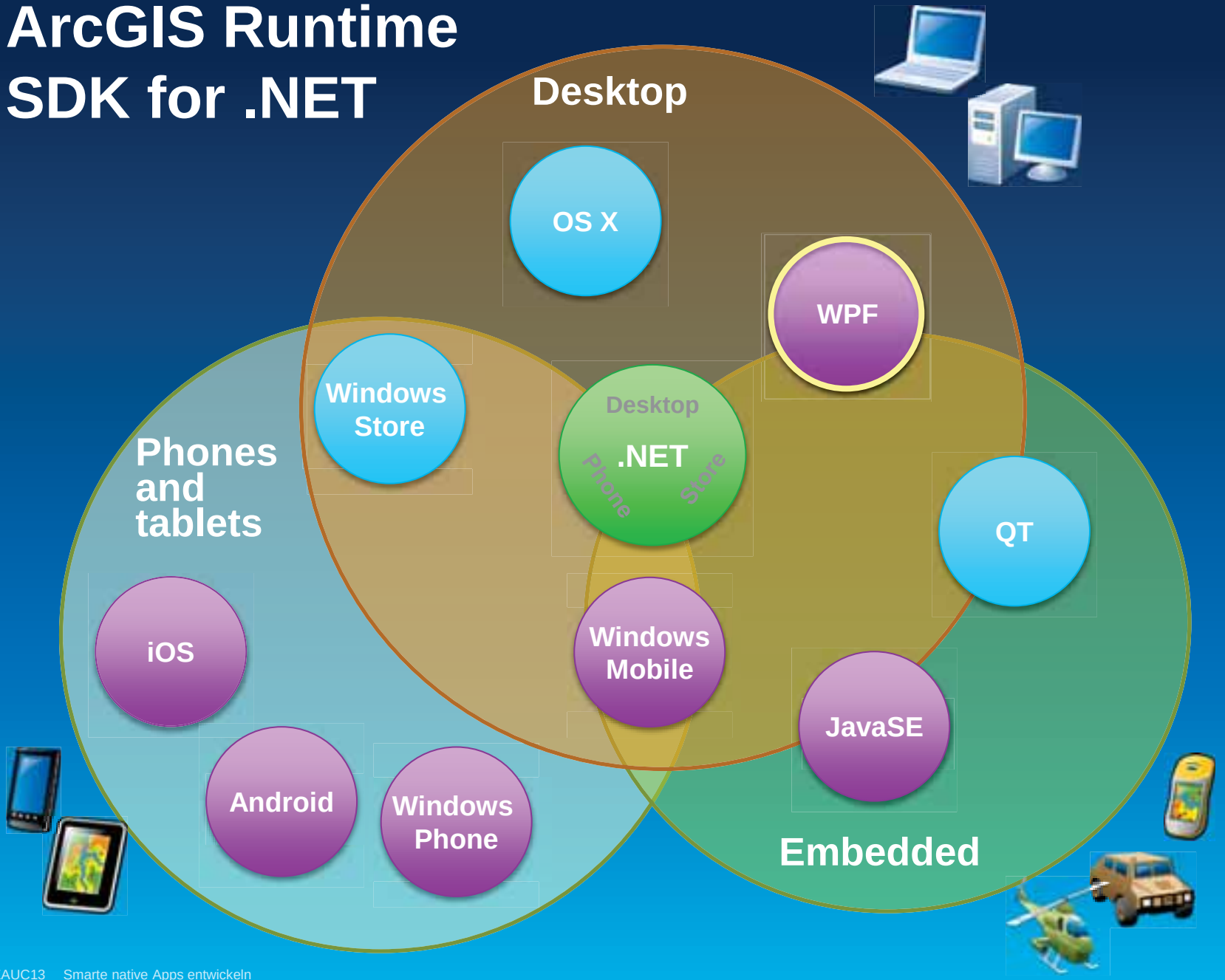
- Qt

- Mac OS X

ArcGIS Runtime SDK for .NET – NEU!

- .NET API für Windows Desktop, Windows Store und Windows Phone
- Konsistente Programmierung
- Benutzt ArcGIS Runtime C++ Kern
- Unterstützt ArcGIS Server, ArcGIS Online und Portal for ArcGIS
- Windows Desktop beinhaltet lokalen Server
- Wird das WPF SDK ersetzen

ArcGIS Runtime SDK for .NET



Mehr Möglichkeiten

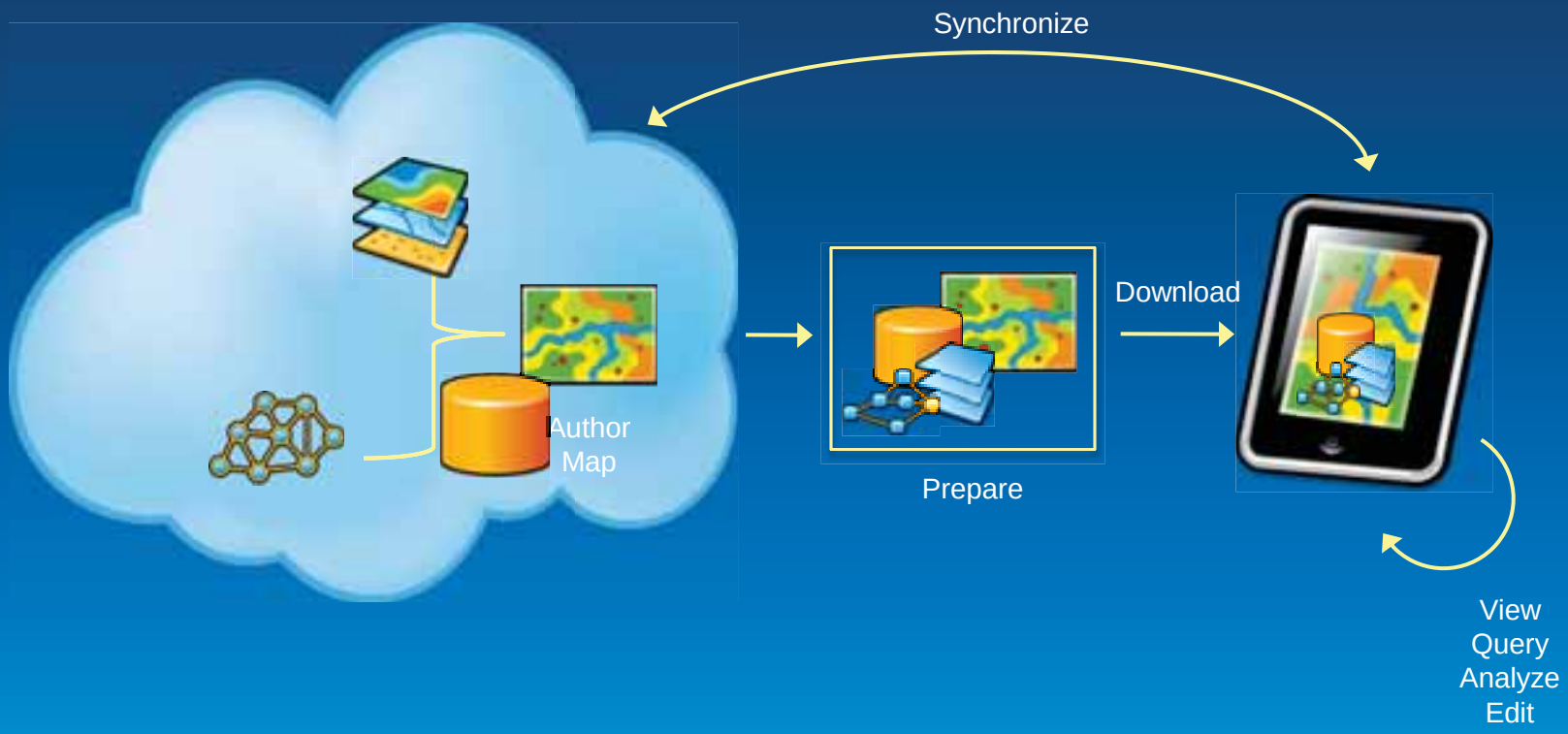
- Visualisierung und Display Performance
  - Verbesserte Symobolisierung
  - Dynamic Labling für DynamicMapServiceLayer
  - Verbesserte Performance des Graphics Layers
- Geometry Support
- Unterstützung für OAuth

Offline Map Capabilities - Beta

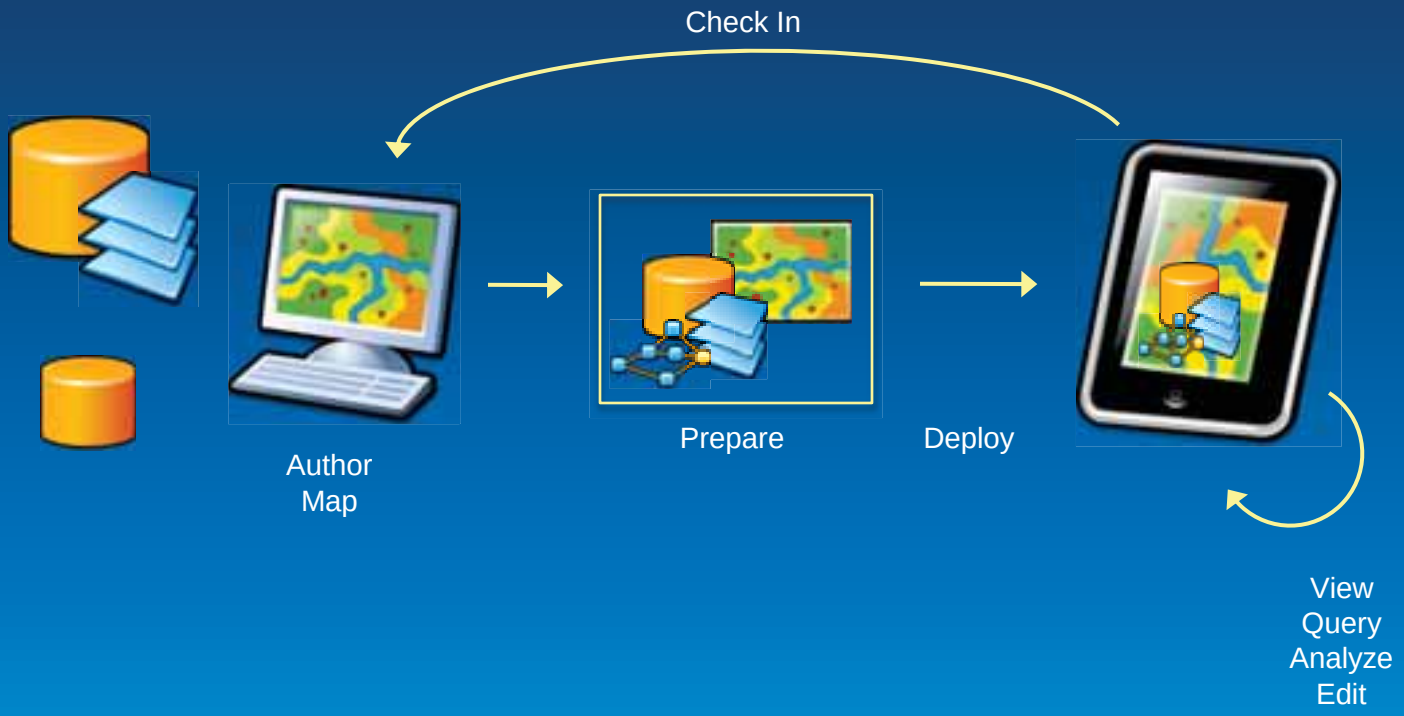
"Disconnected use" der ArcGIS Plattform

- Karten betrachten
- Daten abfragen
- Features editieren
  - Synchronisieren
- Räumliche Intelligenz
  - Orte finden
  - Wegbeschreibung
- Geplant oder nach Bedarf
- **Alle** Runtime SDKs

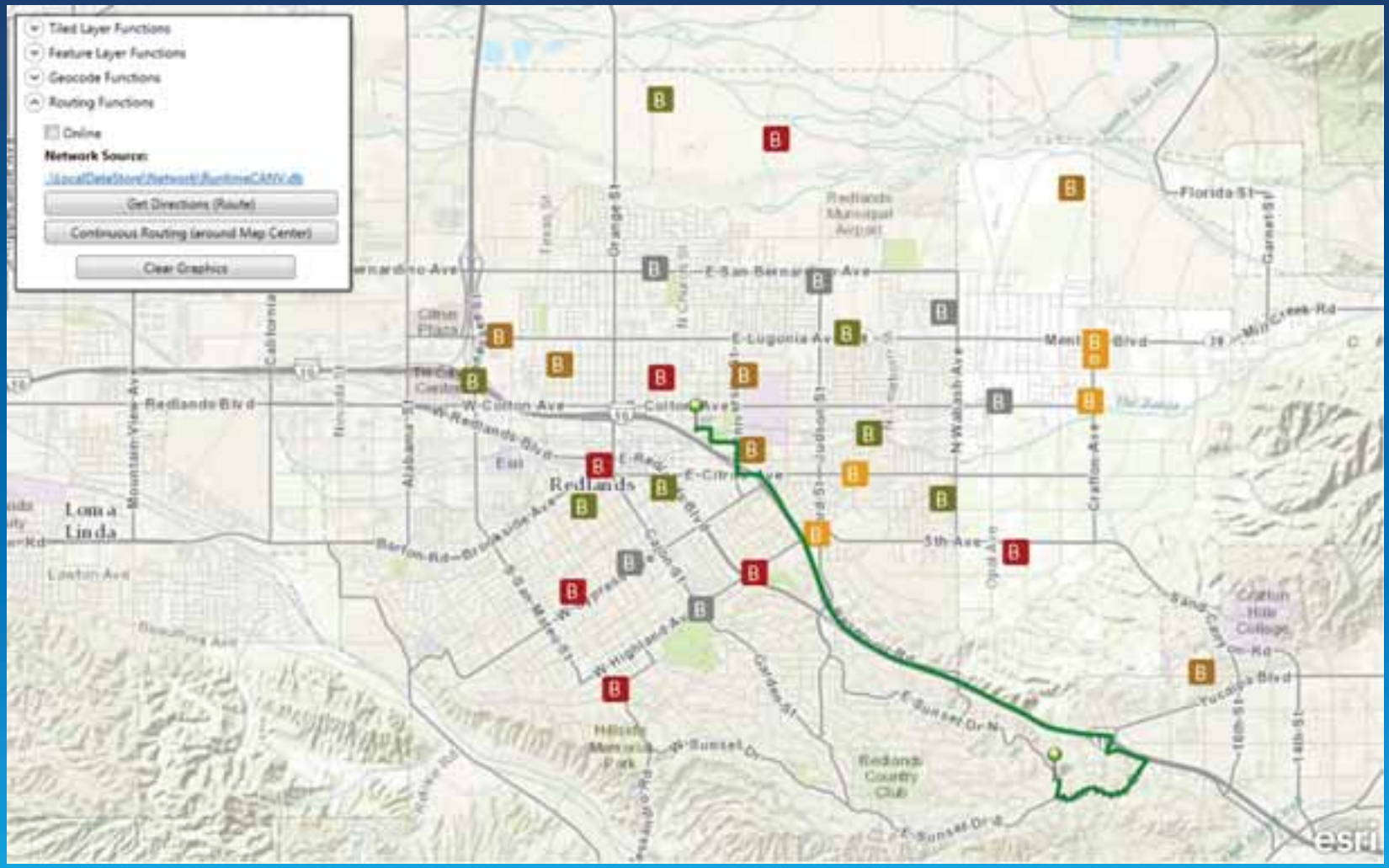
ArcGIS Online und Portal



Desktop

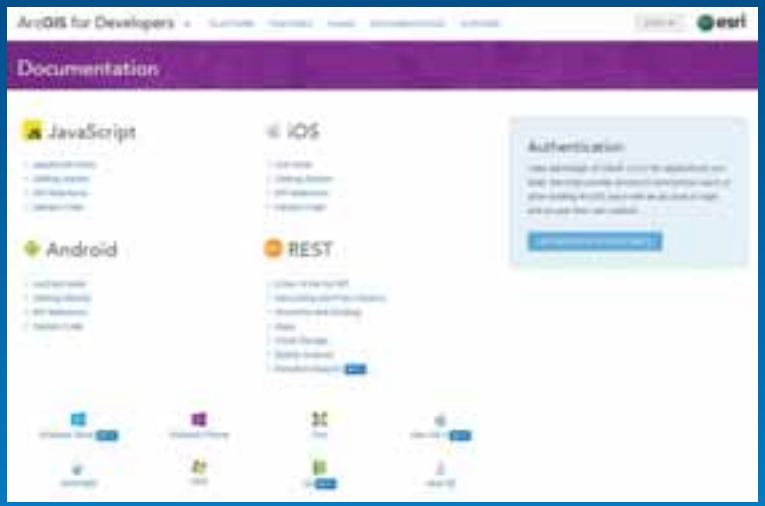


Demo Offline



Verbesserter Zugang

- Alle Runtime SDKs auf der Developers Website
  - Developer Konto
  - EDN nicht erforderlich



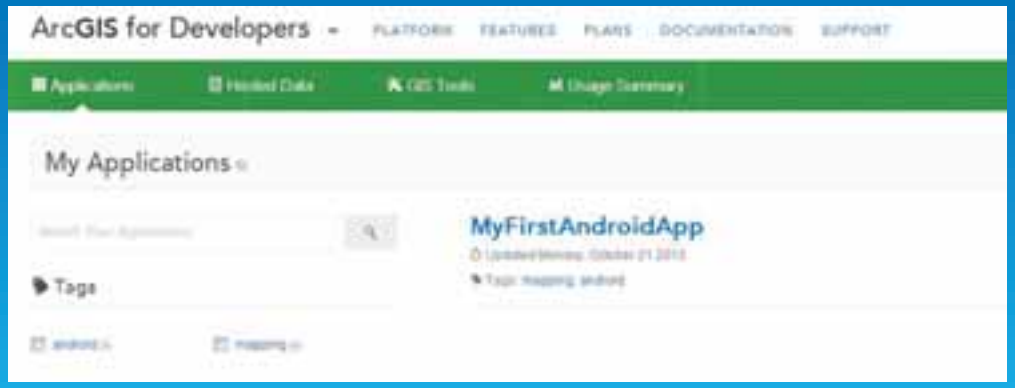
- Andere Bezugsmöglichkeiten
  - Eclip.se Marketplace, NuGet, CocoaPods

Lizenzierung unter 10.2

Entwicklung und Deployment Lizenzierung unverändert

- WPF, Java und Qt
  - Lizenzen für Entwicklung und Deployment benötigt
  - Basic und Standard Deployment Modell

- Development und Basic Deployment Lizenzen auf developers.arcgis.com verfügbar



Deployment

Mobile Runtimes

- Esri License Agreement prüfen
- Anwendung attribuieren
- Fragebogen zum Deployment ausfüllen
- Anwendung verteilen

Deployment

Desktop Runtimes

- Lizenz anfragen
- License String dem Projekt hinzufügen
- Anwendung vorbereiten
- Runtime deployment erstellen

developers.arcgis.com

Vielen Dank!

- Lars Schmitz, Esri Germany
  - Product Manager Developer
  - l.schmitz@esri.de
  - @pilukinum

- Rainald Suchan, Esri Germany
  - Product Specialist
  - r.suchan@esri.de