

Adventures in Ocean Mapping

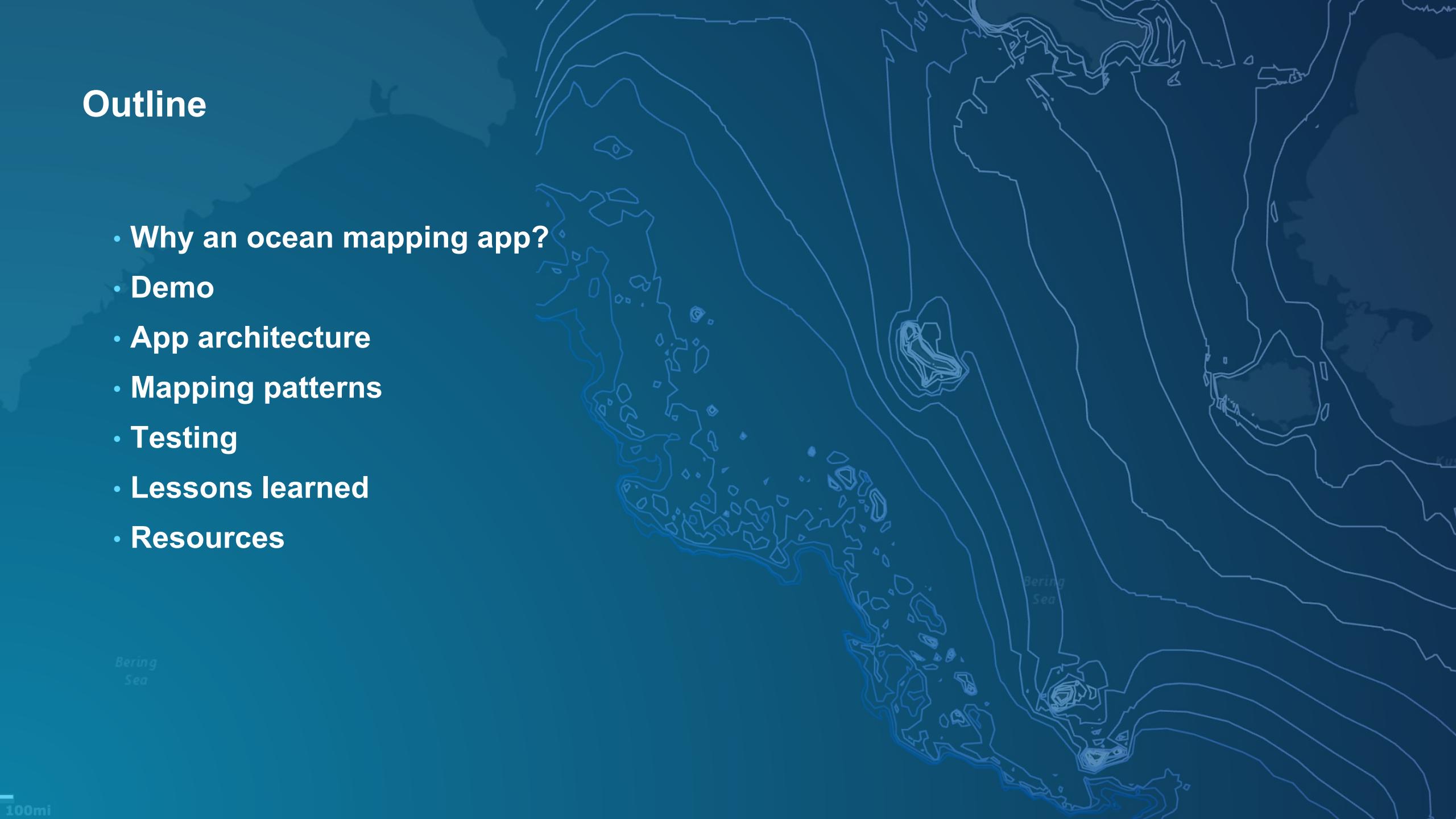
Sandie Peters

Bering
Sea

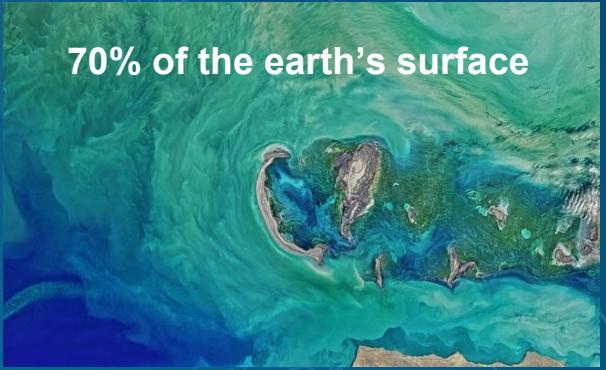
100mi

Outline

- Why an ocean mapping app?
- Demo
- App architecture
- Mapping patterns
- Testing
- Lessons learned
- Resources



Oceans



Overfishing



Only 5% has
been explored



Pollution



Lack of detail about the ocean hinders conservation and management efforts to protect this valuable ecosystem

Photos from <https://www.pexels.com/>

From Ocean Data to Ocean Maps



A 3-D Mesh Framework

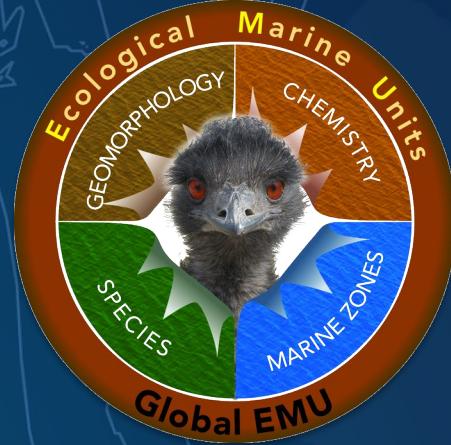
Why was this app built?

1. To raise awareness
2. To serve as a mobile counterpart
3. To showcase the ArcGIS Android SDK.



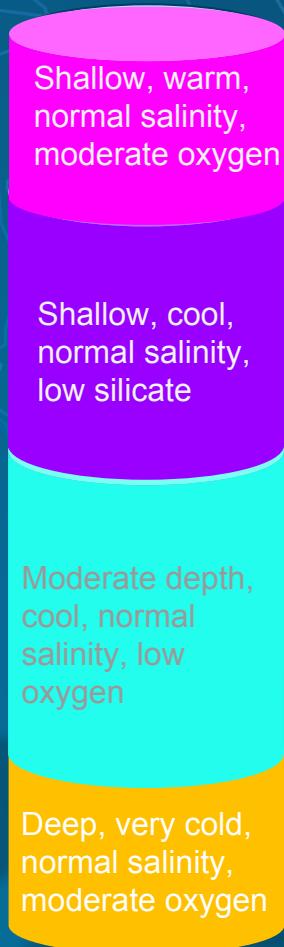
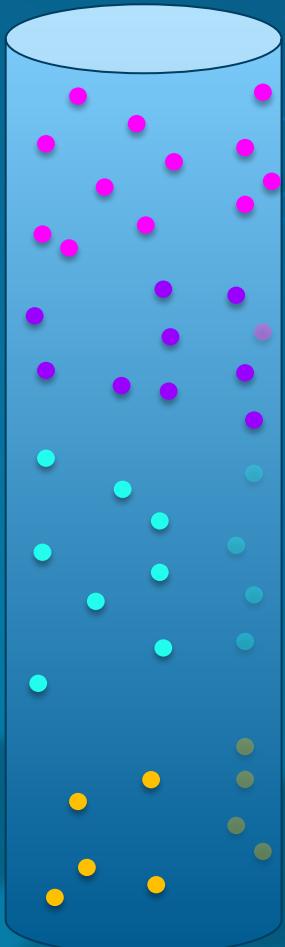
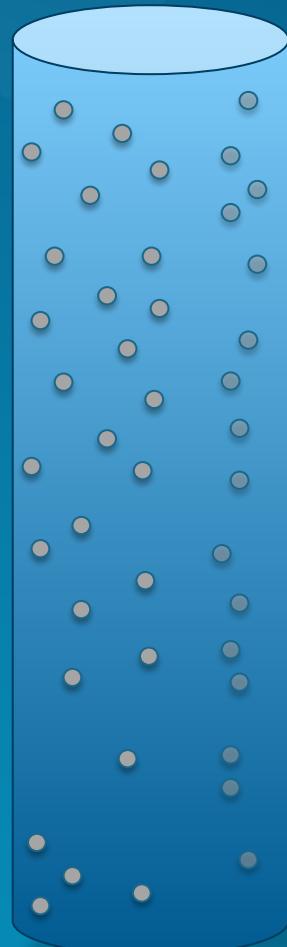
NOAA's 52 million data points collected every 27 km across ocean surface and at regular depth intervals.

Ecological Marine Units are ocean regions of similar temperature, salinity, oxygen and nutrient levels.

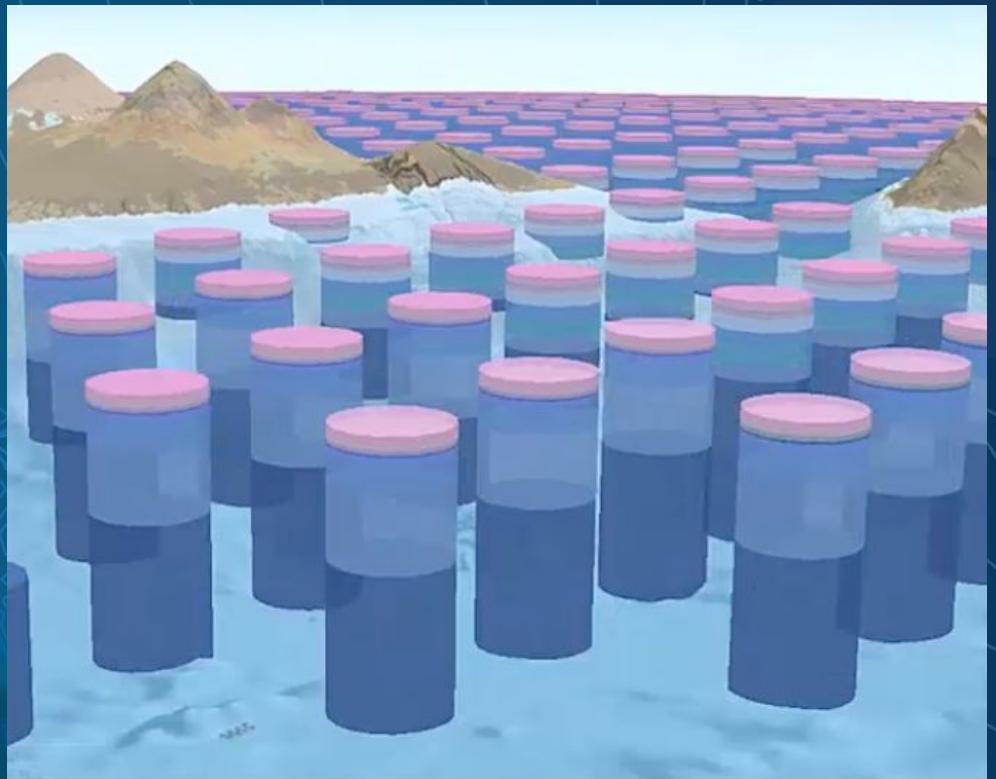


Ocean Measurements Clustered into Statistical Groupings

Ocean surface



Ocean floor



Demo

Get the code on GitHub or download from Google Play

GitHub



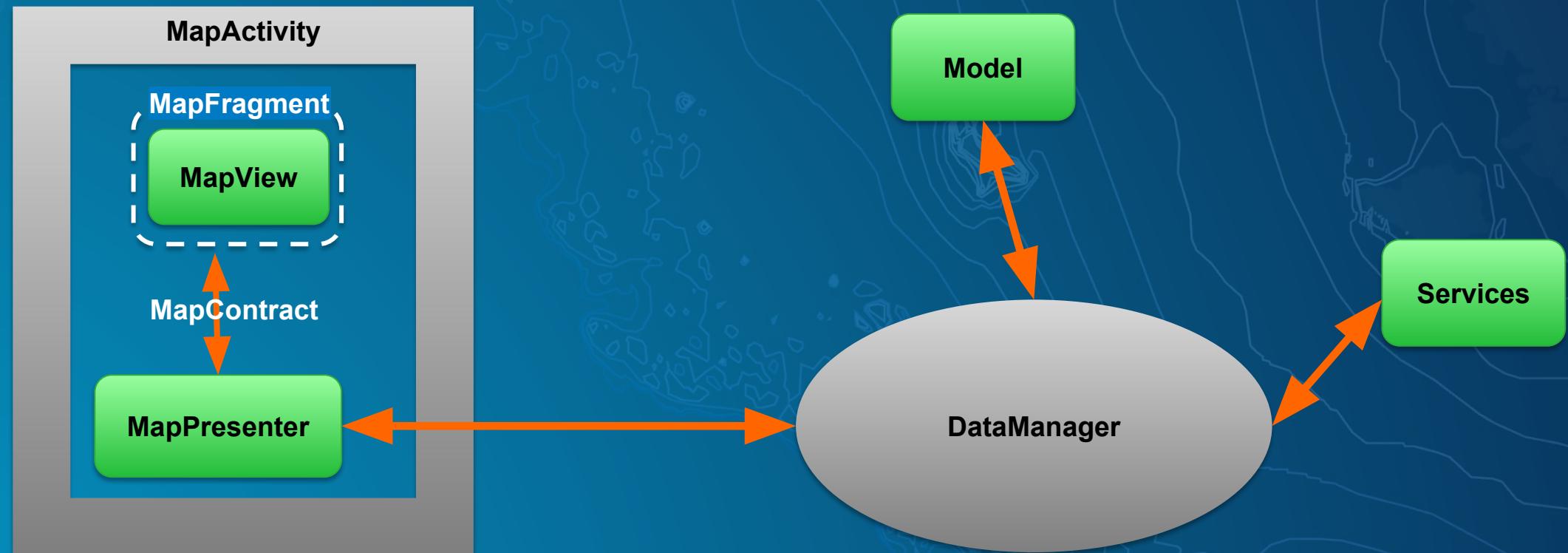
App Development Patterns - MVP



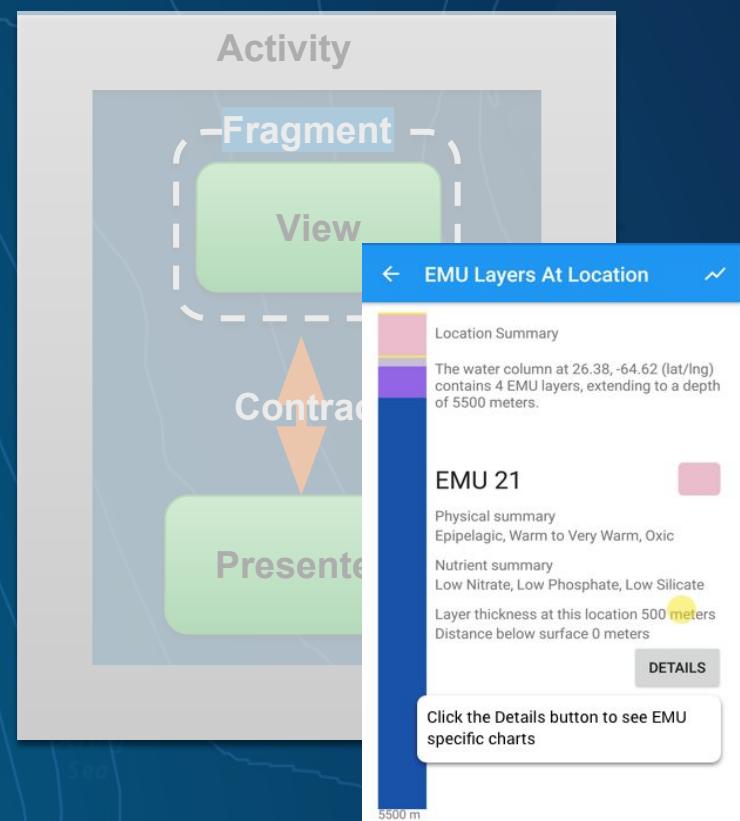
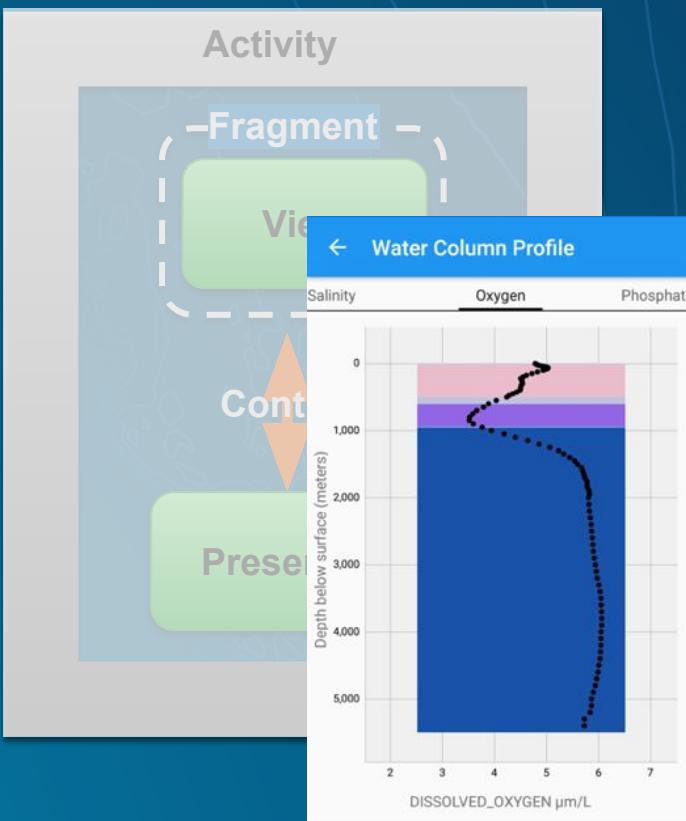
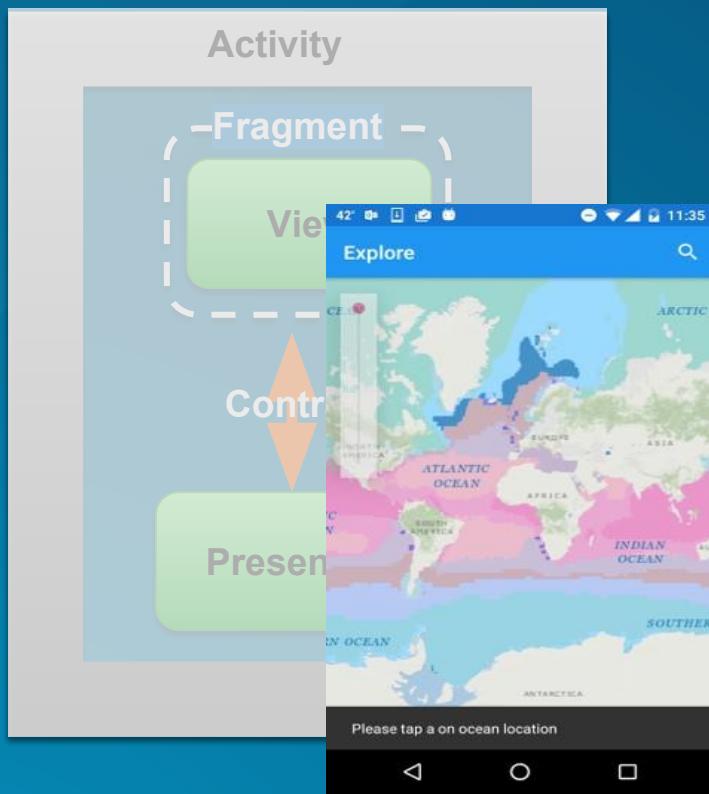
Bering
Sea

100mi

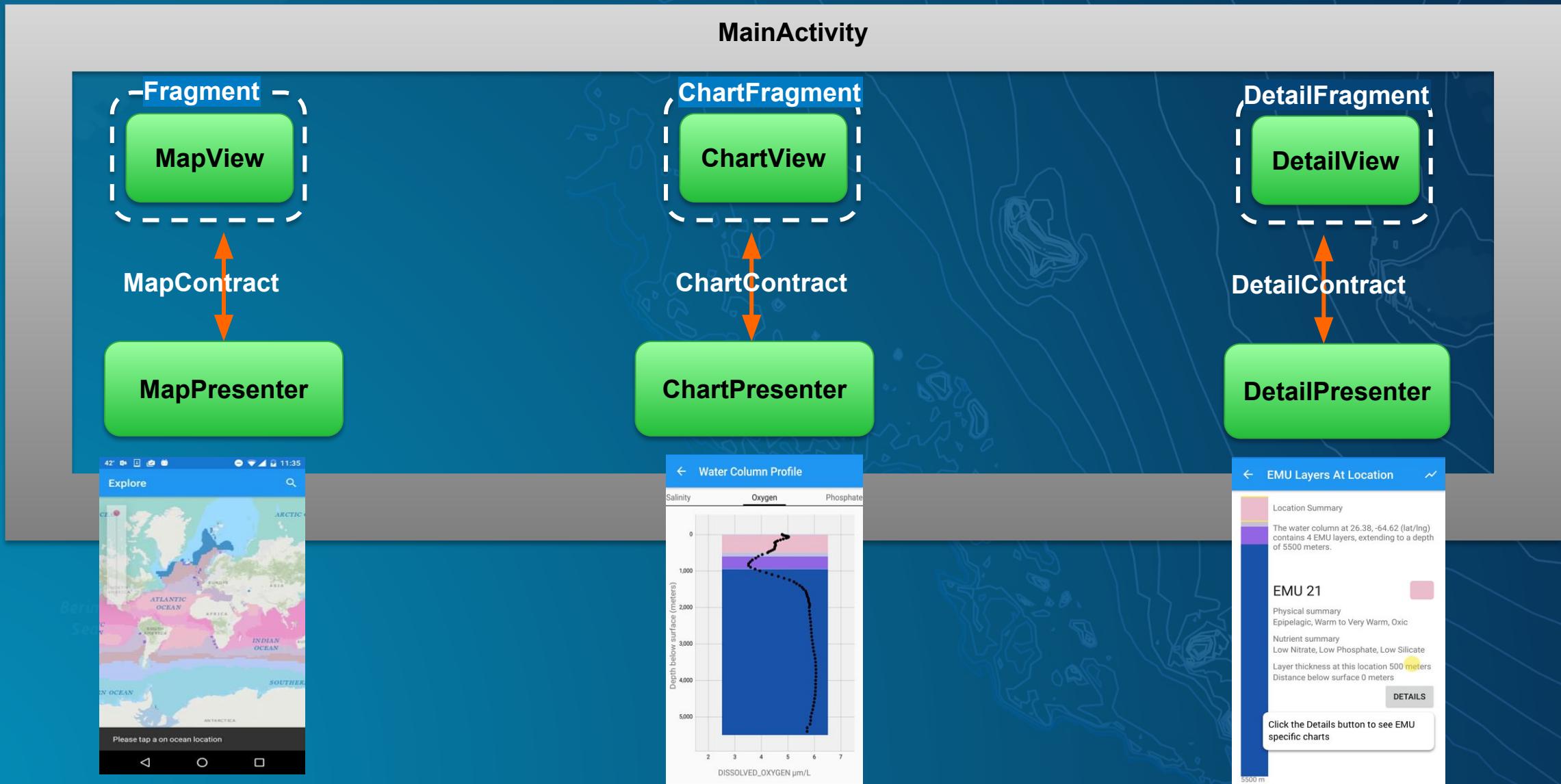
App Development Patterns - MVP



App Development Patterns - MVP



App Development Patterns - MVP



Mapping Patterns - MVP Architecture

Map View

XML or programmatically created
Responds to touch events
Loads data

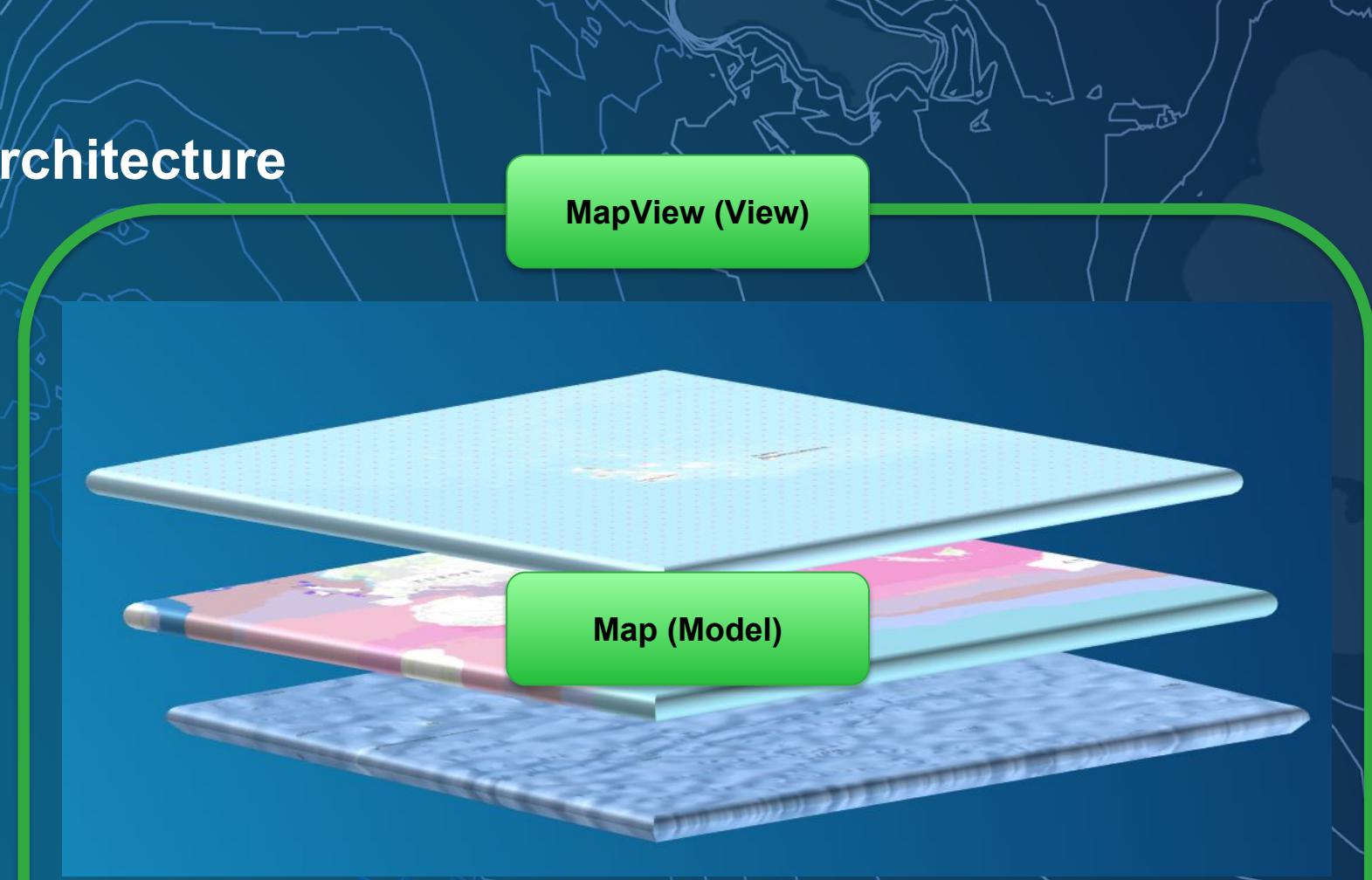
Map

Location as first class data citizen
Specifies how data is organized
Data organized into “layers”
Spatial and non-spatial data

MapView (View)

Map (Model)

Data

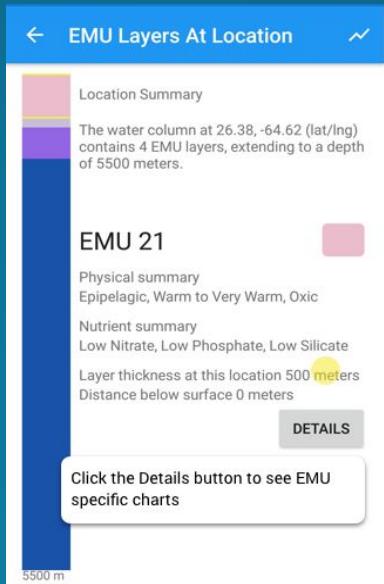


Mapping Patterns – Data Services

Services Backed By Spatial and Non-Spatial Tables

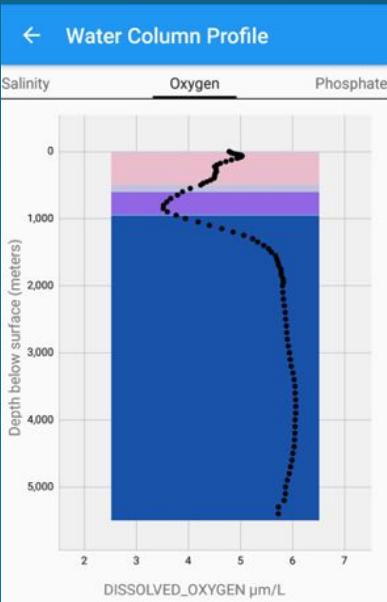
EMU Water Column Summary

Points



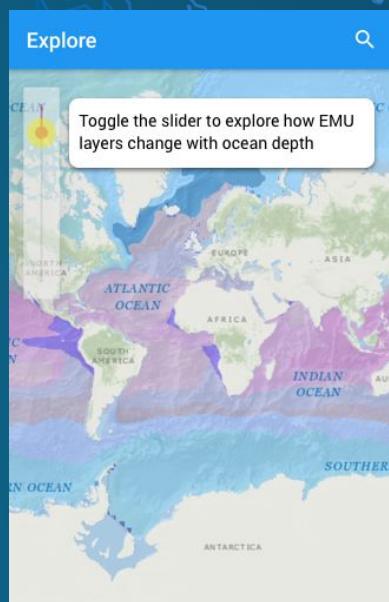
EMU Water Column Measurements

Points



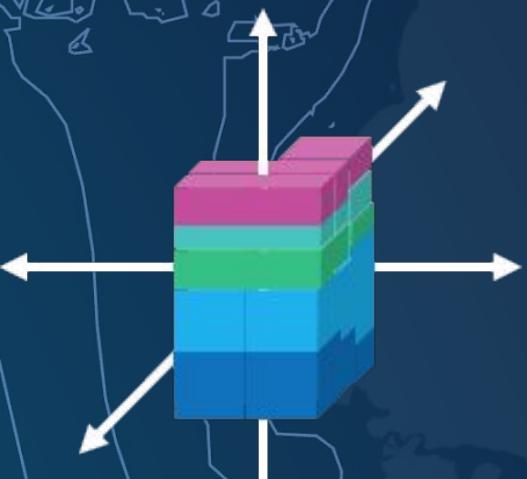
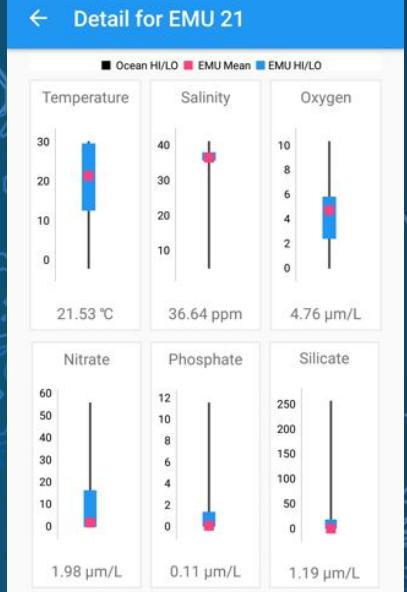
EMU Polygons by Depth

Polygons



EMU Statistics

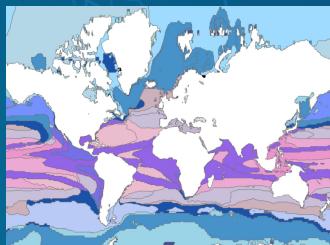
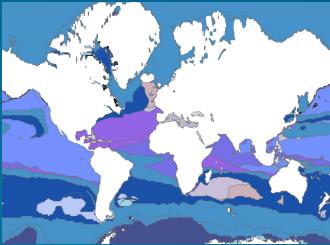
Non-Spatial



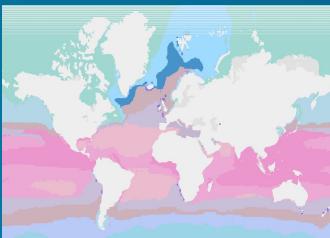
Mapping Patterns

- Adding a map to your app

EMU data by depth



EMU surface data

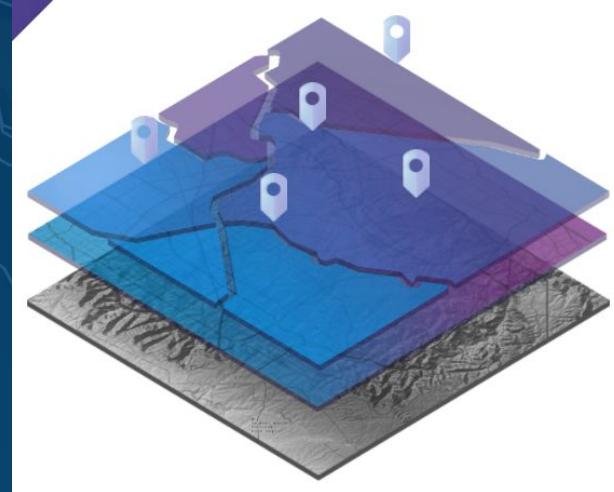


Ocean basemap



This data is added to the map
the first time the slider is moved

The two data layers get added to map on app
startup



Mapping Patterns – Setting up the Map

- Adding a map to your app

```
// Start with an ocean basemap, at a zoom level of 1
ArcGISMap map = new ArcGISMap(Basemap.Type.OCEANS, 0, 0, 1);

// Attach the map to the MapView
mapView.setMap(map)

// Define the EMU Ocean Surface layer
ArcGISTiledLayer layer =
    new ArcGISTiledLayer("http://esri.maps.arcgis.com/home/item.html?id=d2db1dbd6d2742a38fe69506029b83ac");
// Add the operational layer to the map
map.getOperationalLayers().add(layer);
```

```
// Start listening to touch interactions on the map
final View.OnTouchListener mapTouchListener = new MapTouchListener(getActivity().getApplicationContext(), mMapView);
mMapView.setOnTouchListener(mapTouchListener);
```

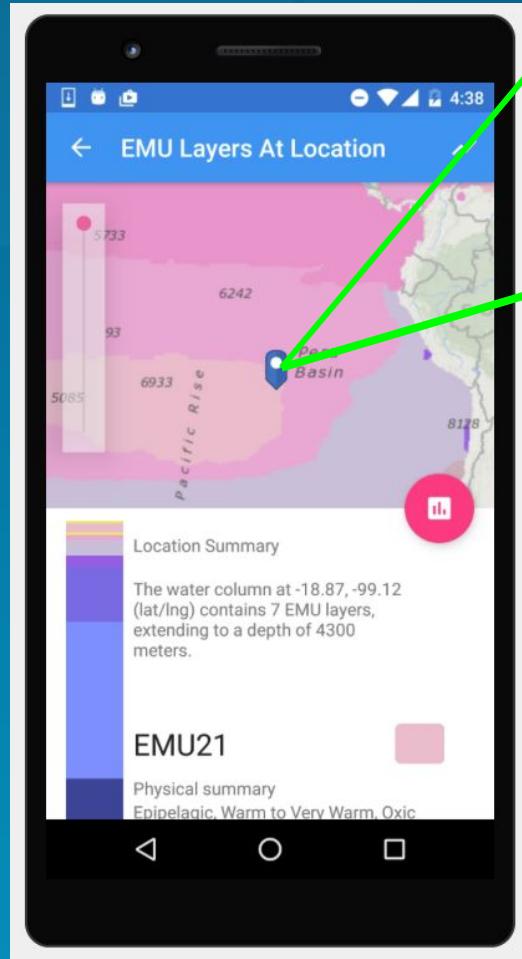
Mapping Patterns – Communicating With Geometry

- Translating Touch Event to Map Geometry

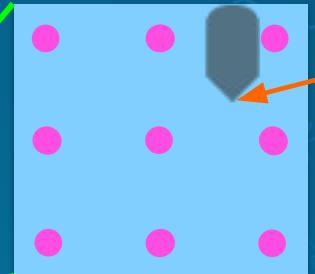
```
public class MapTouchListener extends DefaultMapViewOnTouchListener{  
    /**  
     * Instantiates a new DrawingMapViewOnTouchListener with the specified  
     * context and MapView.  
     *  
     * @param context the application context from which to get the display  
     *                 metrics  
     * @param mapView the MapView on which to control touch events  
     */  
    public MapTouchListener(final Context context, final MapView mapView) { super(context, mapView); }  
    @Override  
    public boolean onSingleTapConfirmed(final MotionEvent motionEvent) {  
        super.onSingleTapConfirmed(motionEvent);  
  
        // Get the screen x,y coordinates  
        final android.graphics.Point screenPoint = new android.graphics.Point((int) motionEvent.getX(),  
            (int) motionEvent.getY());  
  
        // Convert the screen point to a geometry point  
        Point mapPoint = mMapView.screenToLocation(screenPoint);  
  
        // Hand off for further processing  
        mPresenter.setSelectedPoint(mapPoint);  
        return true;  
    }  
}
```



Mapping Patterns - Querying

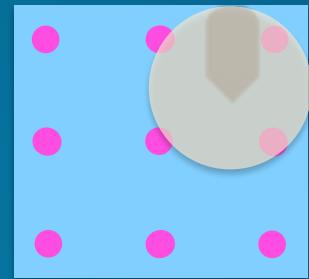


● Individual EMU data point



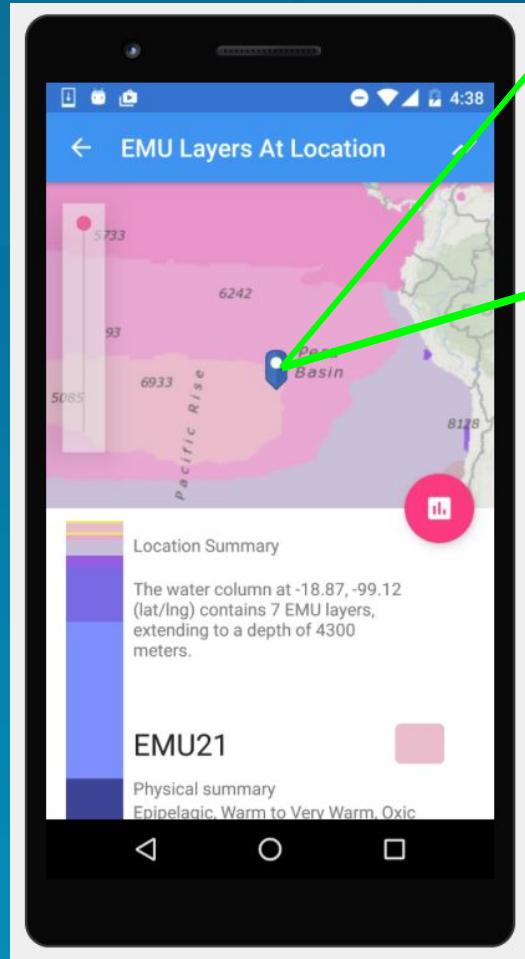
Tapped location translated to map location

```
Polygon bufferedLocation = GeometryEngine.buffer(geoPoint, BUFFER_SIZE);  
PolygonBuilder builder = new PolygonBuilder(bufferedLocation);  
Envelope envelope = builder.getExtent();
```

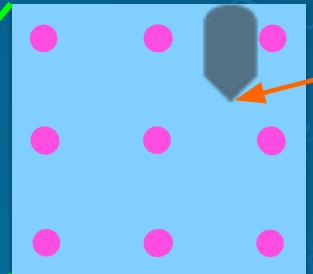


```
final QueryParameters queryParameters = new QueryParameters();  
  
// Use the buffered polygon to constrain the query  
queryParameters.setGeometry(envelope);  
  
// Load the service table  
mMeshClusterTable =  
    new ServiceFeatureTable(mContext.getString(R.string.service_emu_mesh_cluster));  
  
// Asynchronously execute the query  
final ListenableFuture<FeatureQueryResult> futureResult =  
    mMeshClusterTable.queryFeaturesAsync(queryParameters,  
        ServiceFeatureTable.QueryFeatureFields.LOAD_ALL);  
  
// Handle the future and callback  
processQueryForEmuAtLocation(envelope, futureResult, callback);
```

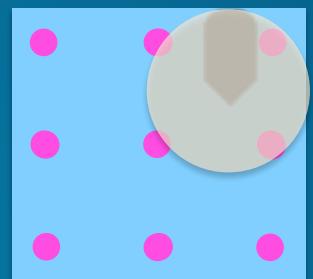
Mapping Patterns - Calculating Distance



● Individual EMU data point



Tapped location translated to map location



```
// Get the center of the buffered polygon
final Point center = envelope.getCenter();

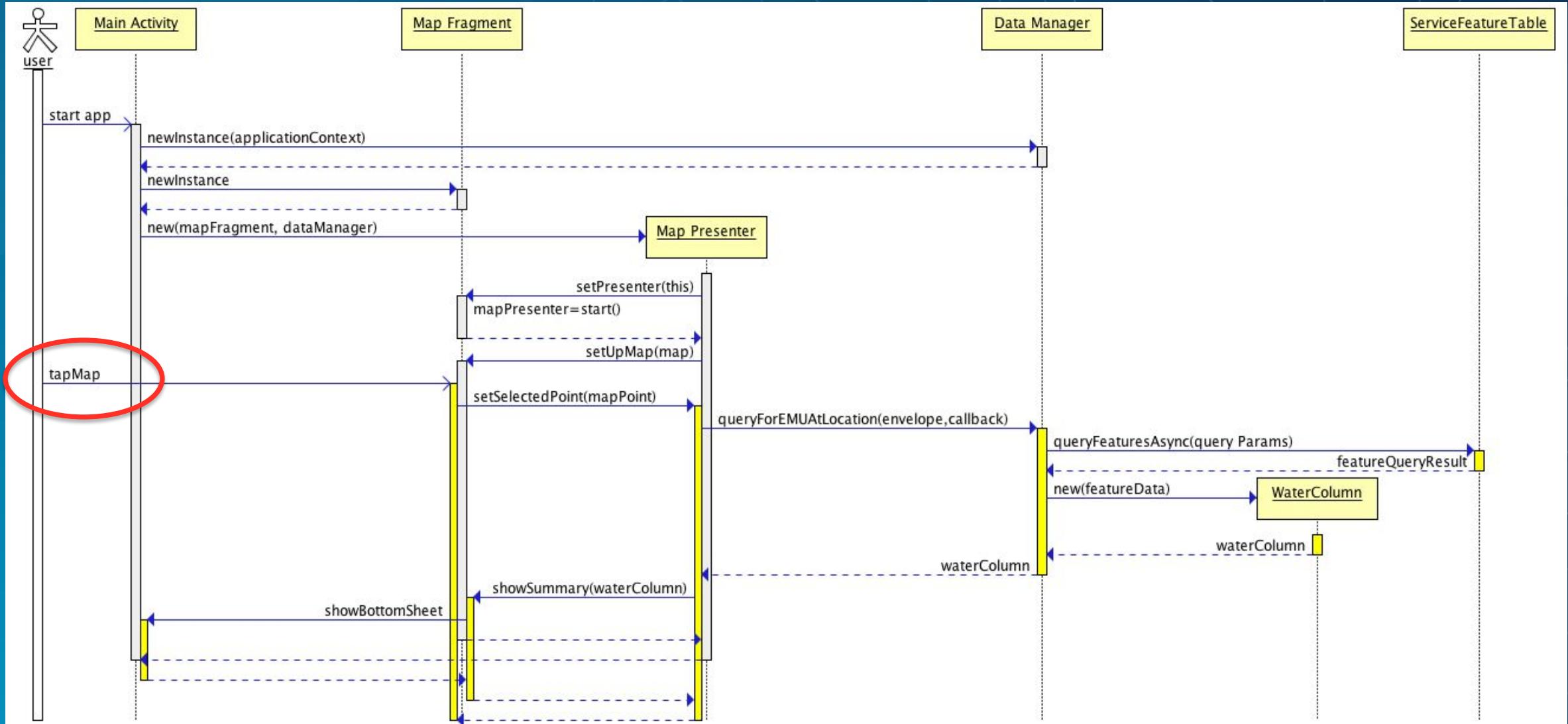
// Instantiate the returned Point for the water column
final Point waterColumnPoint = new Point(point.getX(), point.getY(),
    center.getSpatialReference());

// Calculate the distance between the center and the point
final GeodeticDistanceResult geodeticDistanceResult =
    GeometryEngine.distanceGeodetic(center, waterColumnPoint,
        linearUnit, angularUnit, GeodeticCurveType.GEODESIC);

final double calculatedDistance = geodeticDistanceResult.getDistance();

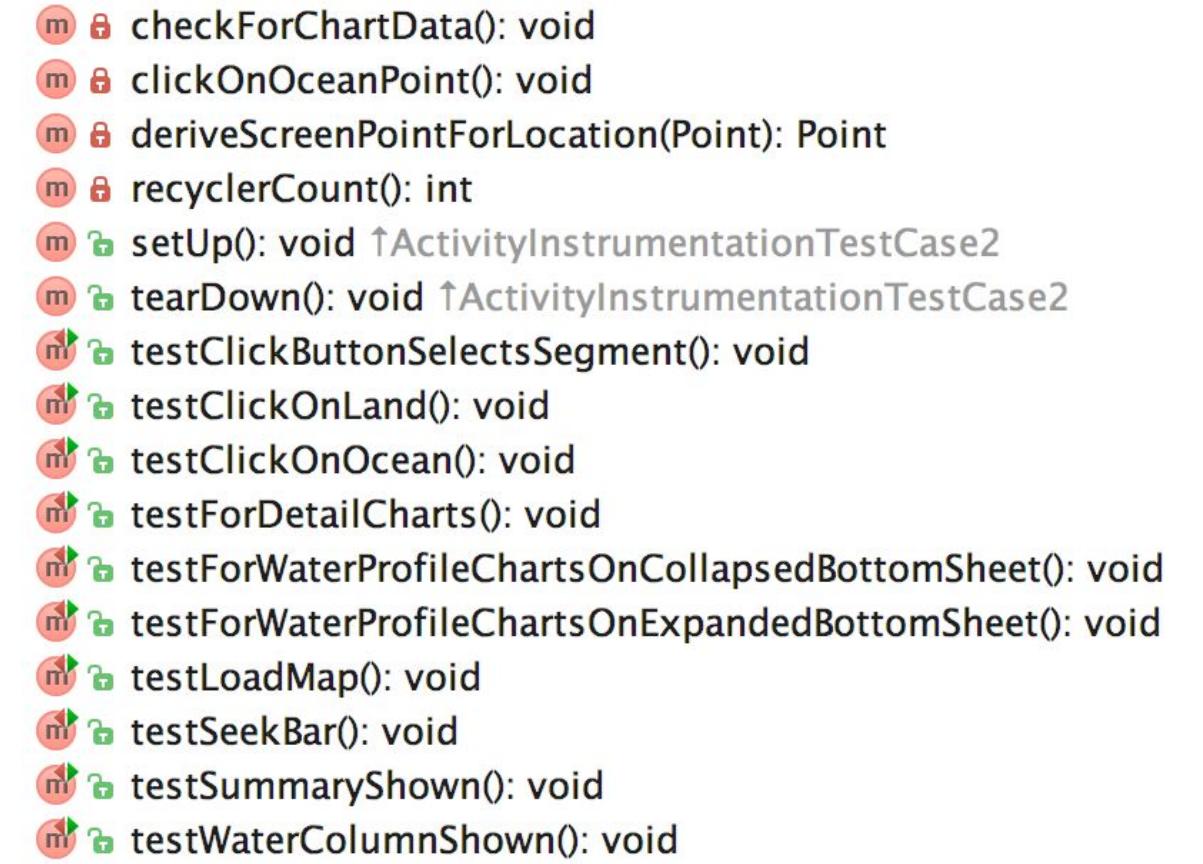
// Set the distance
waterColumn.setDistanceFrom(calculatedDistance);
```

Map and App Patterns



Testing

- Robotium (v 5.6.3) extends Android test framework
 - Black box testing
 - Relatively easy to write UI tests
 - Screenshots
-
- No explicit testing of app logic
 - Network latency and complex map views
 - Dependency management
 - Mocking



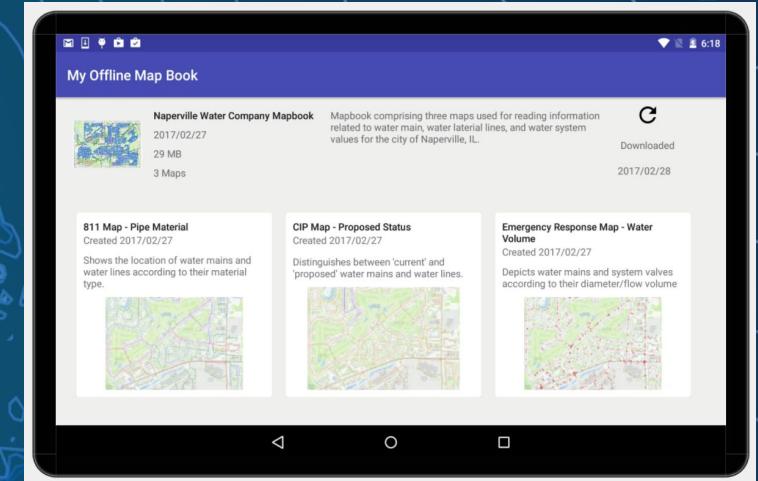
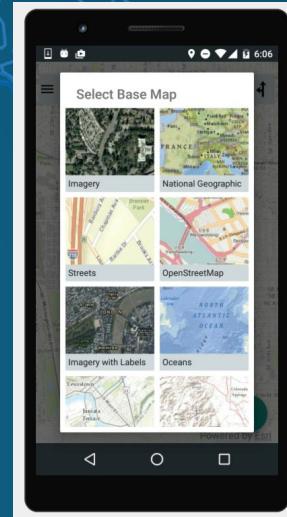
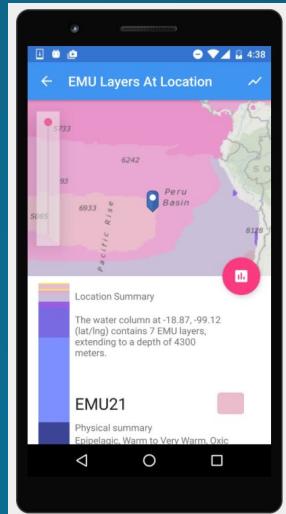
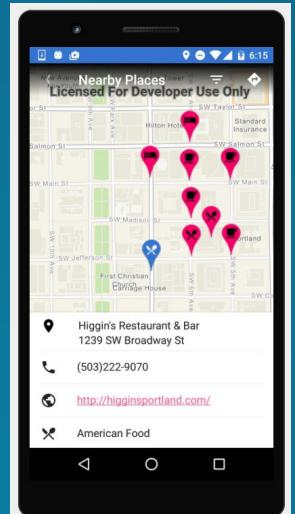
The screenshot shows a list of test methods in an Android test class. The methods are color-coded with red icons for private methods and green icons for public methods. Most methods have a lock icon, indicating they are protected.

- m checkForChartData(): void
- m clickOnOceanPoint(): void
- m deriveScreenPointForLocation(Point): Point
- m recyclerCount(): int
- m setUp(): void ↑ActivityInstrumentationTestCase2
- m tearDown(): void ↑ActivityInstrumentationTestCase2
- m testClickButtonSelectsSegment(): void
- m testClickOnLand(): void
- m testClickOnOcean(): void
- m testForDetailCharts(): void
- m testForWaterProfileChartsOnCollapsedBottomSheet(): void
- m testForWaterProfileChartsOnExpandedBottomSheet(): void
- m testLoadMap(): void
- m testSeekBar(): void
- m testSummaryShown(): void
- m testWaterColumnShown(): void

Android Mapping Apps

Open source
Best practices
Extensible

<https://github.com/Esri/ecological-marine-unit-android>
<https://github.com/Esri/nearby-android>
<https://github.com/Esri/maps-app-android>
<https://github.com/Esri/mapbook-android>

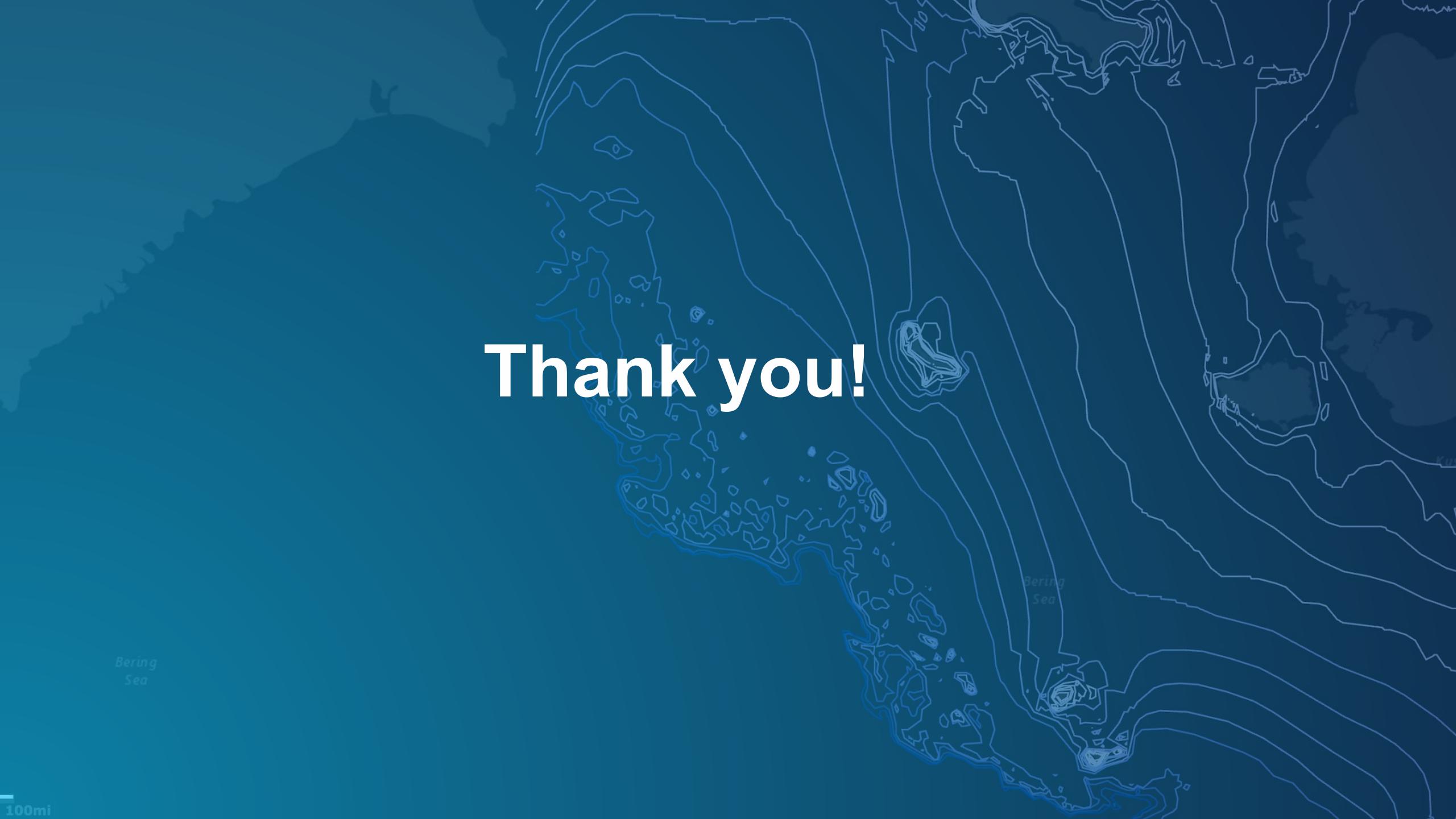


Lessons Learned

1. The geo-spatial analysis of ocean conditions provides us with maps to help manage ocean ecosystems.
2. Apps centered around complex views can remain responsive by using multiple Fragments managed by one Activity, but there is a cost of maintaining and coordinating view switching.
(What other strategies would work?)
3. The Android SDK for maps and MapView can be used to display millions of location-based data and contextual information.
4. Comprehensive automated app testing is difficult in Android. MVP architecture can ease the way, but it's only one of many considerations.

Resources

- **MP Android charting library** - <https://github.com/PhilJay/MPAndroidChart>
- **Android Blueprint Architecture** - <https://github.com/googlesamples/android-architecture>
- **Robotium** - <https://github.com/RobotiumTech/robotium>
- **ArcGIS Developer's Site** - <https://developers.arcgis.com/android/>
- **Android ArcGIS Map Samples** - <https://github.com/Esri/arcgis-runtime-samples-android>



Thank you!

Bering
Sea

100mi