

Esri Developer Summit

March 8–11, 2016 | Palm Springs, CA



Building Great Android App UIs and UXs

Will Crick

Dan O'Neill

Sabine Barrera

Rama Chintapalli

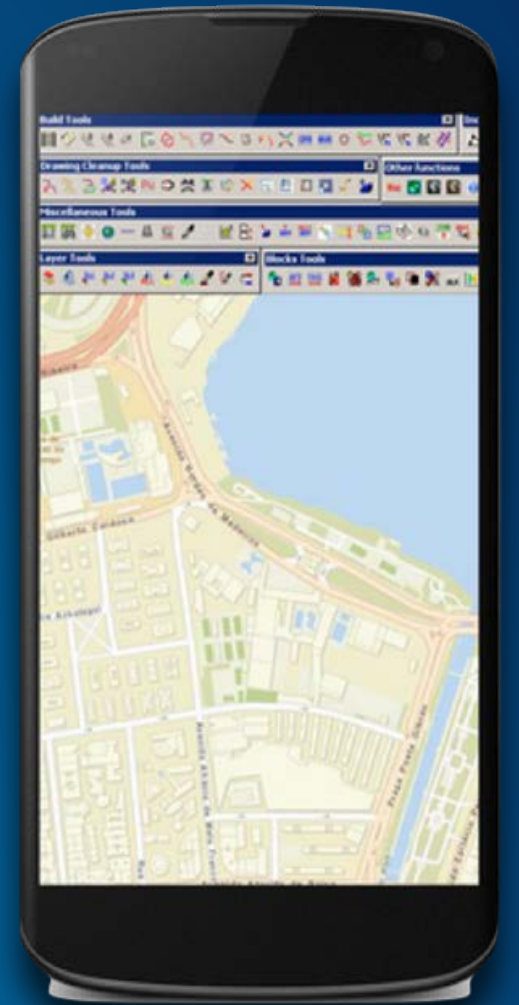
What we will cover

- **Material design**
- **Good UI/UX process**
- **Building a map focused app**
- **Searching for data**
- **Working with your portal**
- **Providing location based alerts**

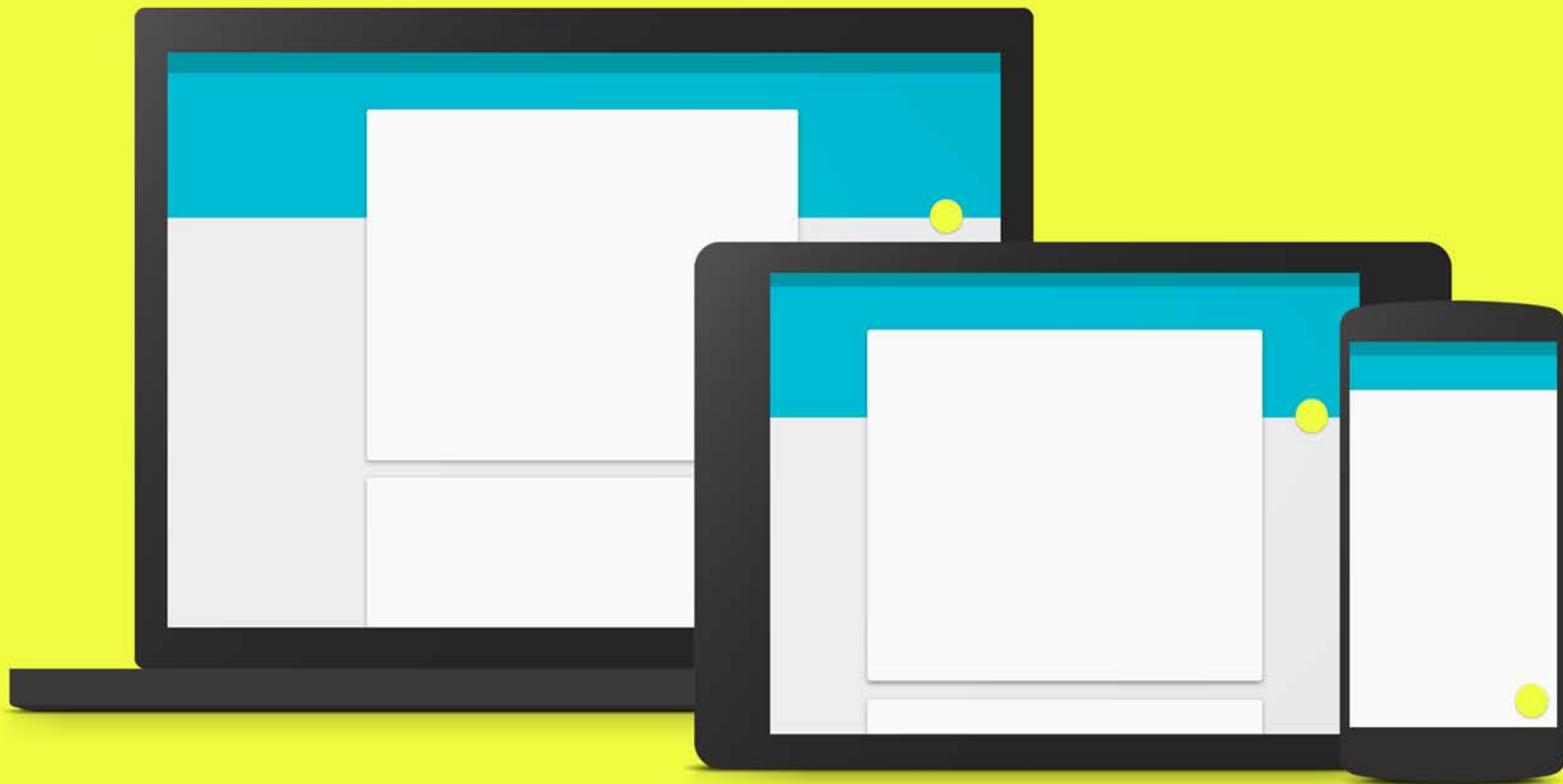
Before you start cranking code...

Ask yourself some questions...

- **What sort of location based app are you building?**
- **Location Based Experience**
 - What value does location have to the user?
 - How will the user interact with location?
 - How does the app consume location?
- **What are the patterns i can learn from?**
 - <https://developers.arcgis.com/android/guide/determine-your-app-map-pattern.htm>



Material design?



Google's answer to Apple's design dominance...

“...a visual language for our users that synthesizes the classic principles of good design with the innovation and possibility of technology and science.”

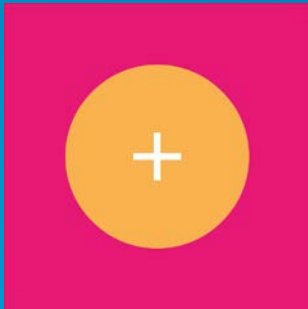
Unified experience across platforms, device sizes and human computer interaction (touch, voice, mouse & keyboard)

Creating a Google brand AND an Android brand

Principles



Material is the metaphor – modern tech inspired by paper and ink
(not skeuomorphic – does not replicate)



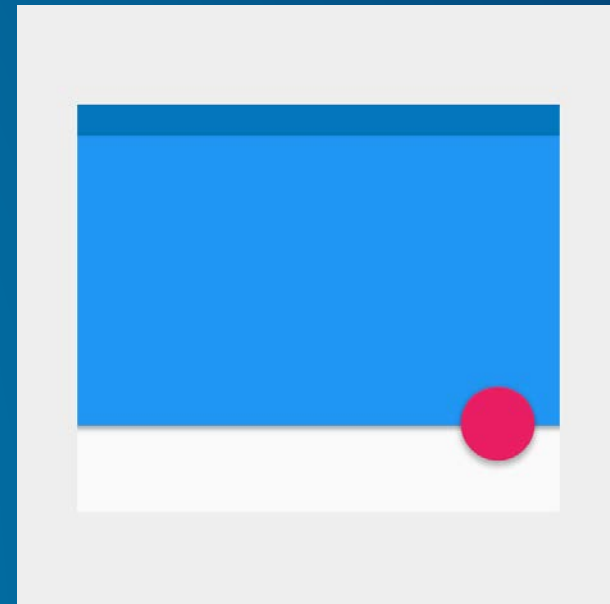
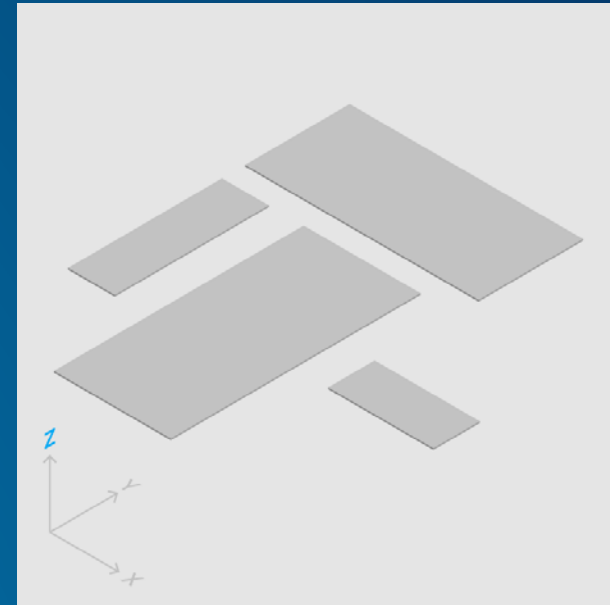
Bold, graphic, intentional – print based design guides
visual treatment



Motion provides meaning – user actions initiate movement

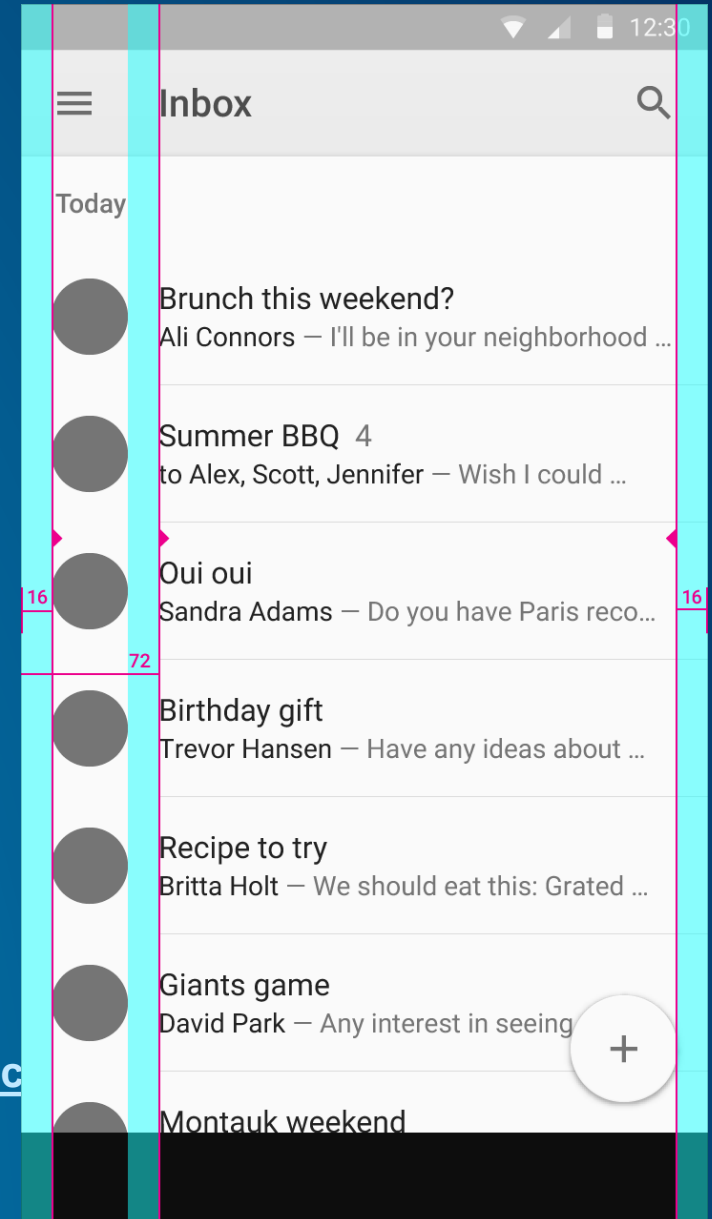
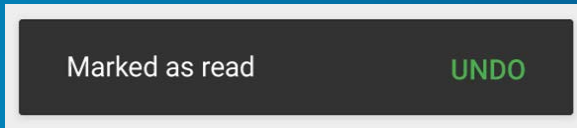
Structure

- **3d layering of content**
 - Elevation of elements important – conveyed by shadows
- **Movement of surfaces under each other**
 - Grouping of objects, siblings should move the same
- **Floating action buttons**



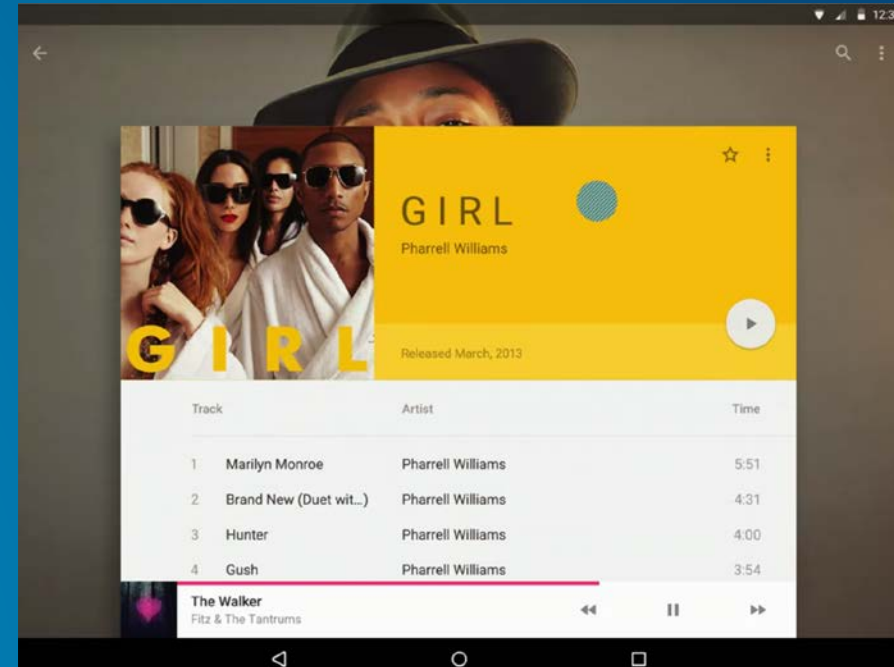
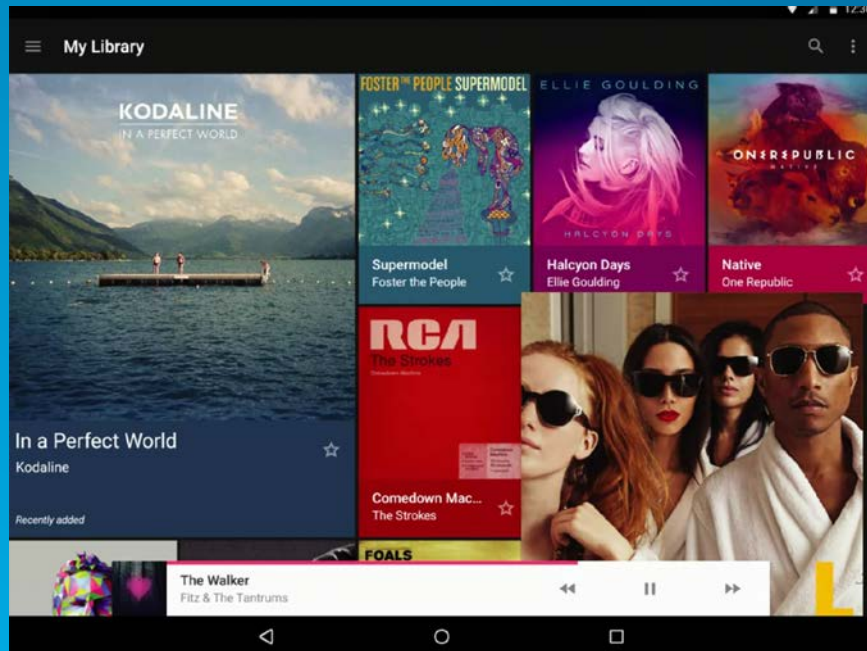
Layout and color

- Sizing of components
 - toolbars = 56dp / 64dp
- Never split sheets with another sheet!
- Grids and templates – take care of your dp!!!
- Cards for related data (not homogenous content)
- Color & Themes - material theme
 - 3 from primary, one from accent palette
 - <https://www.materialpalette.com/>
- Chips, Snackbars and Toasts, *Tasty...*
- Imagery - don't use stock images
 - Use Hero images!
- Maximise your content, take up the screen real estate
- <http://www.google.com/design/spec/style/imagery.html#imagery-best-practices>



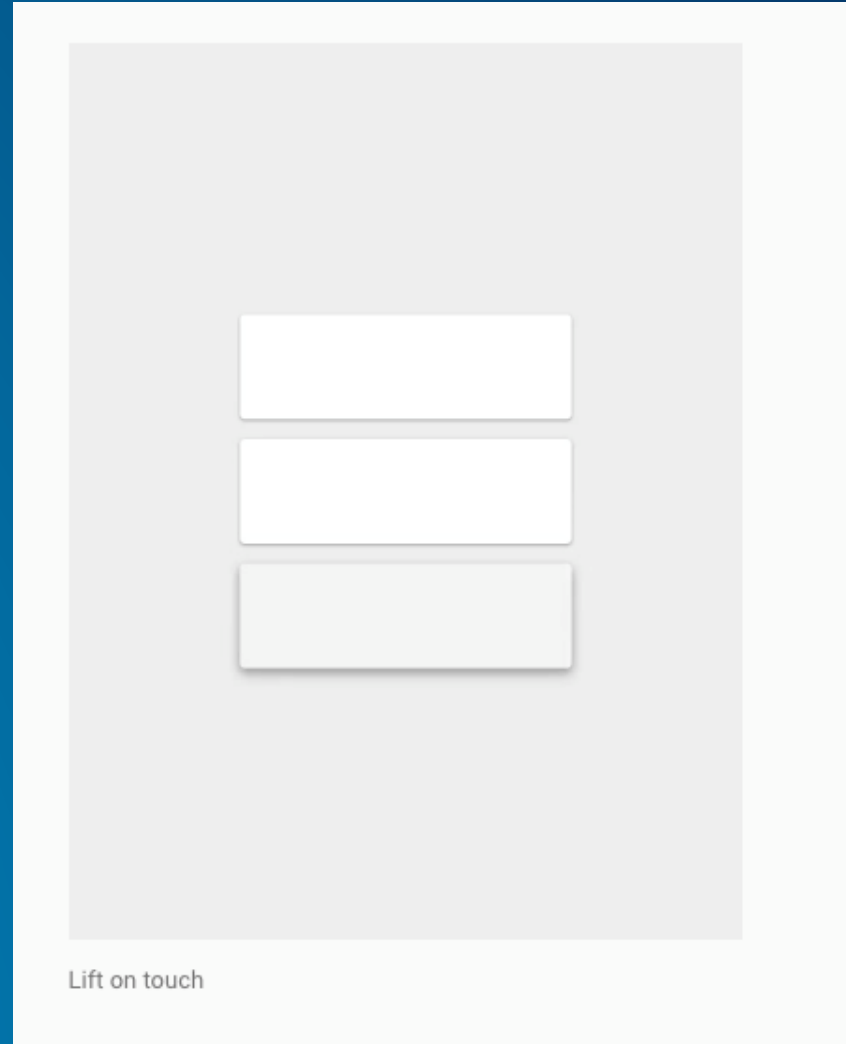
Meaningful transitions

- Content grows/shrinks from its source to destination
- Help organize the information architecture of the app



Interaction feedback

- **Highlight/selection of paper**
- **Effects**
 - **Surface reaction – touch ripple**
 - **Material response – lift up when touched**
- **Helps guide user in what to do next**



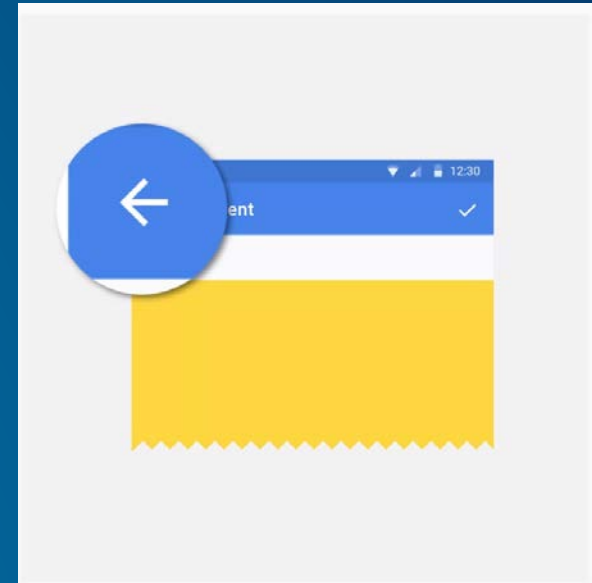
Animations

- **Authentic motion and consistent choreography**

- <http://www.google.com/design/spec/animation/authentic-motion.html#authentic-motion-mass-weight>

- **Delightful details**

- <http://www.google.com/design/spec/animation/delightful-details.html>



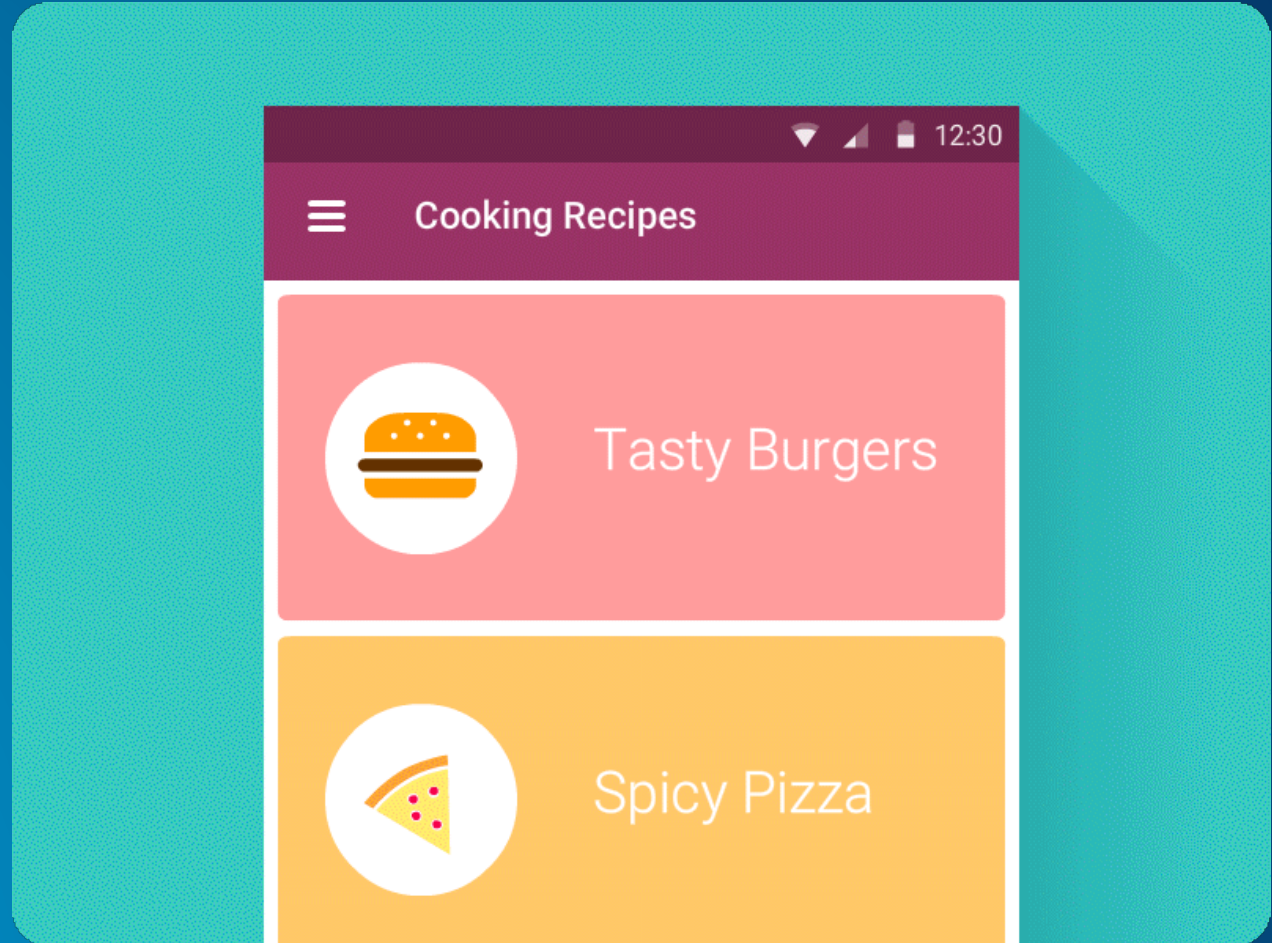
Iconography

Simple, flat, consistent

Longest page in the spec, so beware....



Take reasonable opportunities to do something interesting.

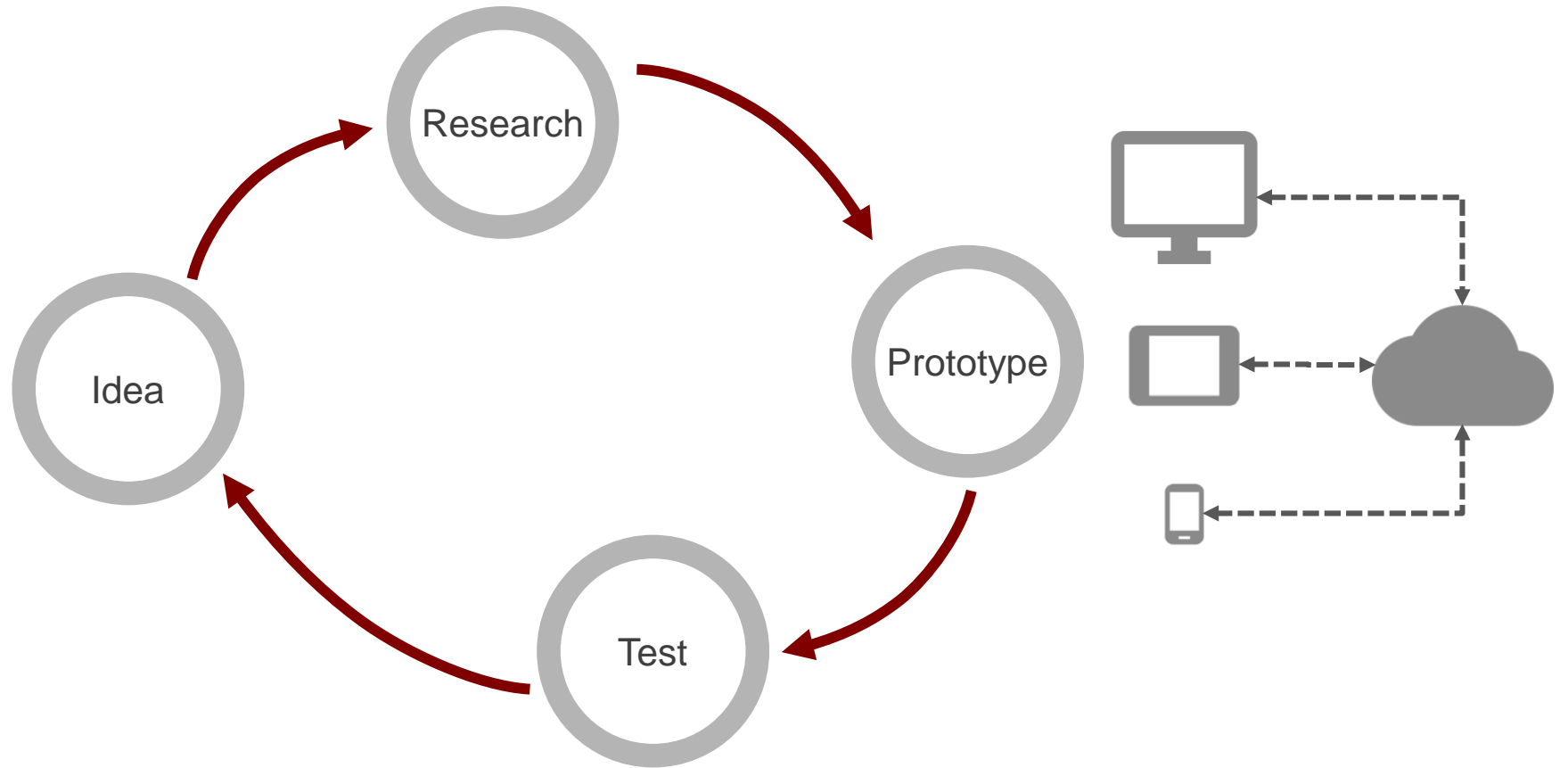


What is a good design process?

SABINE

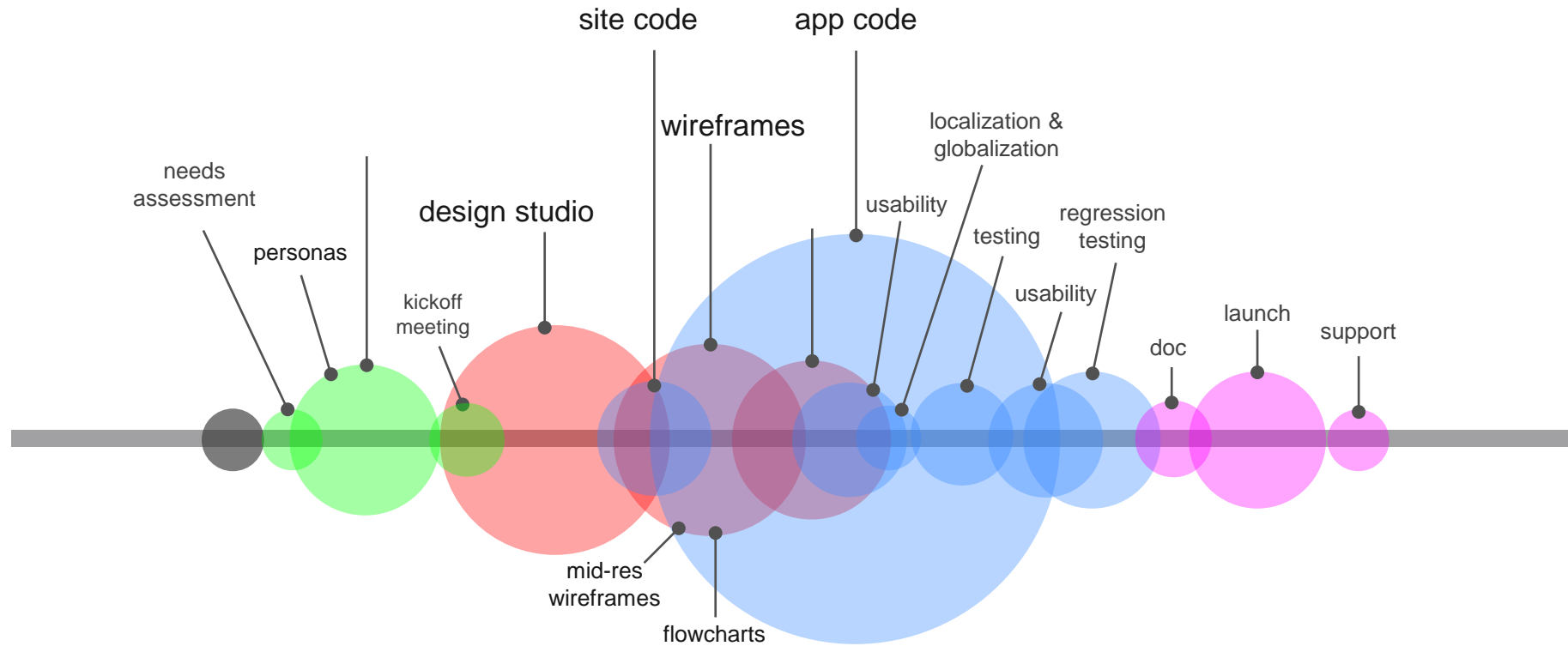
Esri's apps project process

WORKFORCE CASE



Esri's apps project process - detailed

WORKFORCE CASE



Defining Personas

Workforce is comprised of 3 components:

1. Field Worker mobile app



- Used by workers in field
- May be 1 worker or a crew (leader)
- Phone or tablet
- Views and completes assignments
- Shares status

2. Dispatcher

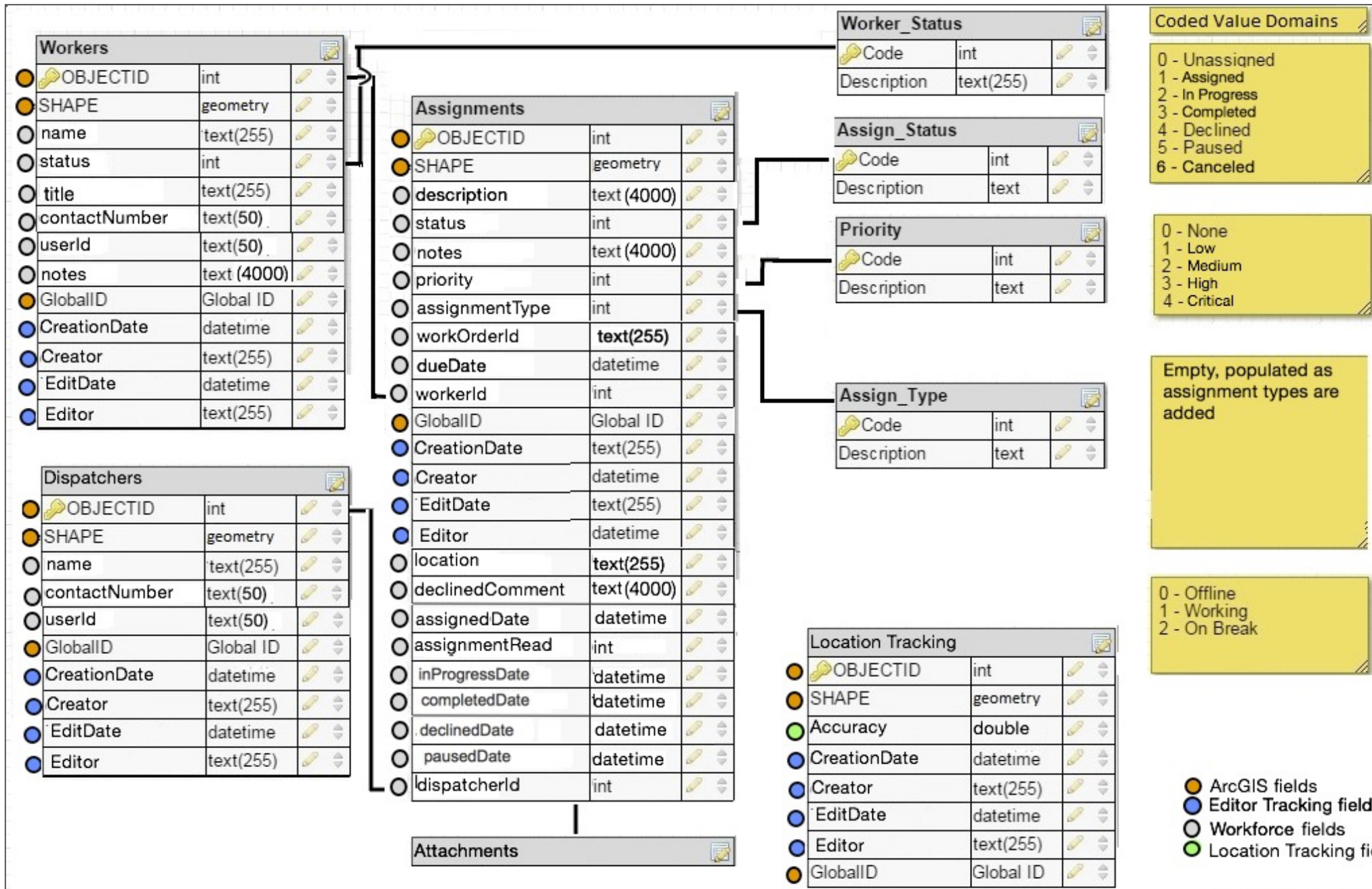


- Sits in an office at a desktop
- Assigns and prioritizes work
- Manages adhoc/emergencies
- Views work and worker status

3. Administrative app and Service



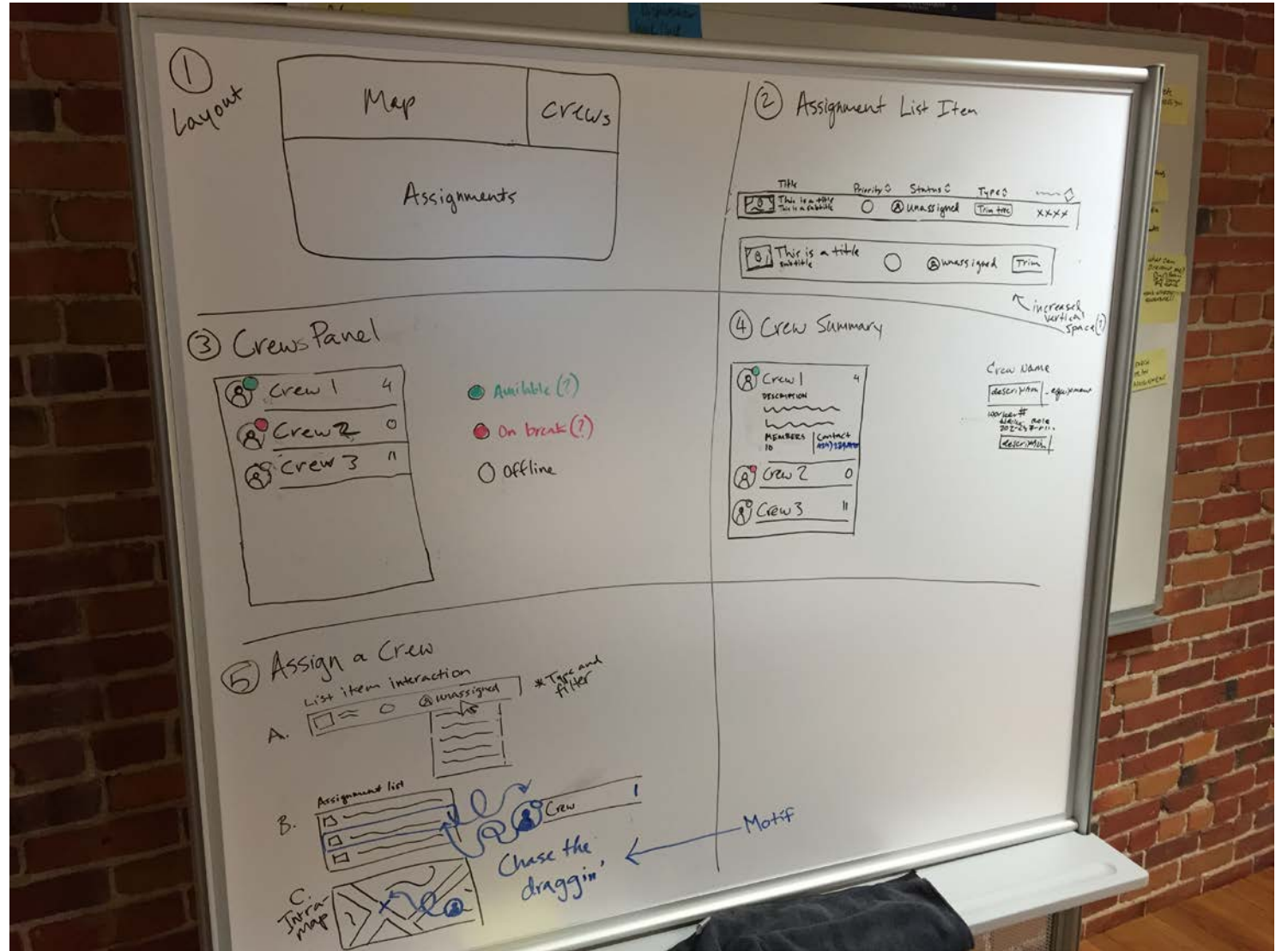
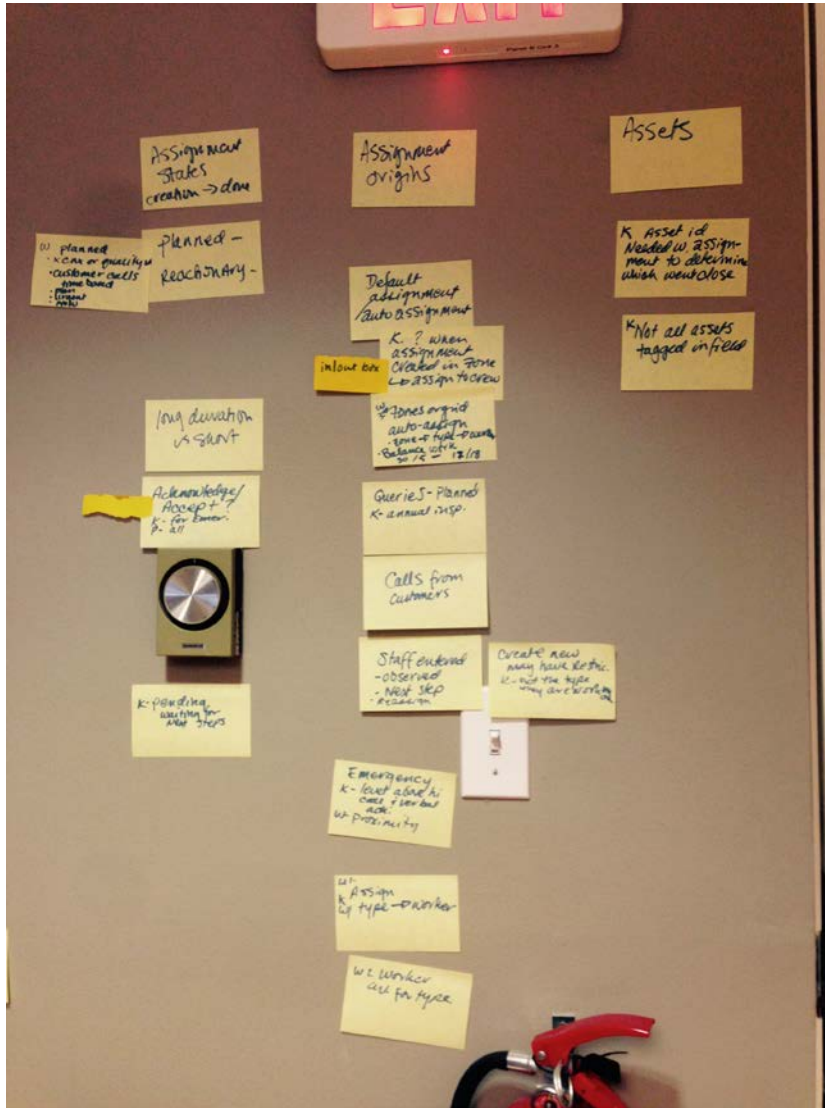
- Sits in an office at a desktop
- Sets up the system
- Manages users and roles
- Generates/sets up reports



Kickoff Meeting - Design Studio



Design Studio - Backoffice Web App



Design Studio - Mobile App

Roadmap

- Lightweight assignments integ. via systems
- planned assignments
- Reporting NICE
 - Benefit
 - Need to invest more
 - ops dashboard
- App Integration
 - Collector
 - Nav. - pass list & create route
 - Audit trail NICE
 - History tracking
 - Tracking
- Projects
 - Map
 - Dispatcher
 - Worker
 - Admin controls
 - Priority
 - Do we include CR - YES
 - Custom - NO
 - Assignment table
 - ID - assist or work?
 - keep
 - to map
 - to assign
 - to add
 - How does disp. get populated to store non item info?
 - Read or Acknowledged
 - planned vs Ad hoc?
- in/out box
 - Default assignments zones/due date? time?
 - worker NICE
 - work flow
 - reasons
 - Create in field
 - for barcode
 - same type
 - diff table

TEAM LIFT

Assignments List

- Assign to you
- This is a title 10 Jan 2015 | ID#
- This is a title 10 Jan 2015 | ID#
- This is a title 10 Jan 2015 | ID#

Assignments Detail

- This is a title 10 Jan 2015 | ID#
- Joe Franko started
- tasks
- ACCEPT | REJECT
- This is a title 10 Jan 2015 | ID#
- This is a title 10 Jan 2015 | ID#

Details

- This is a title 10 Jan 2015 | ID#
- Description
- ACCEPT | REJECT

MAP

- GPS Location
- Assignment pin w/ warning symbol
- reflects status

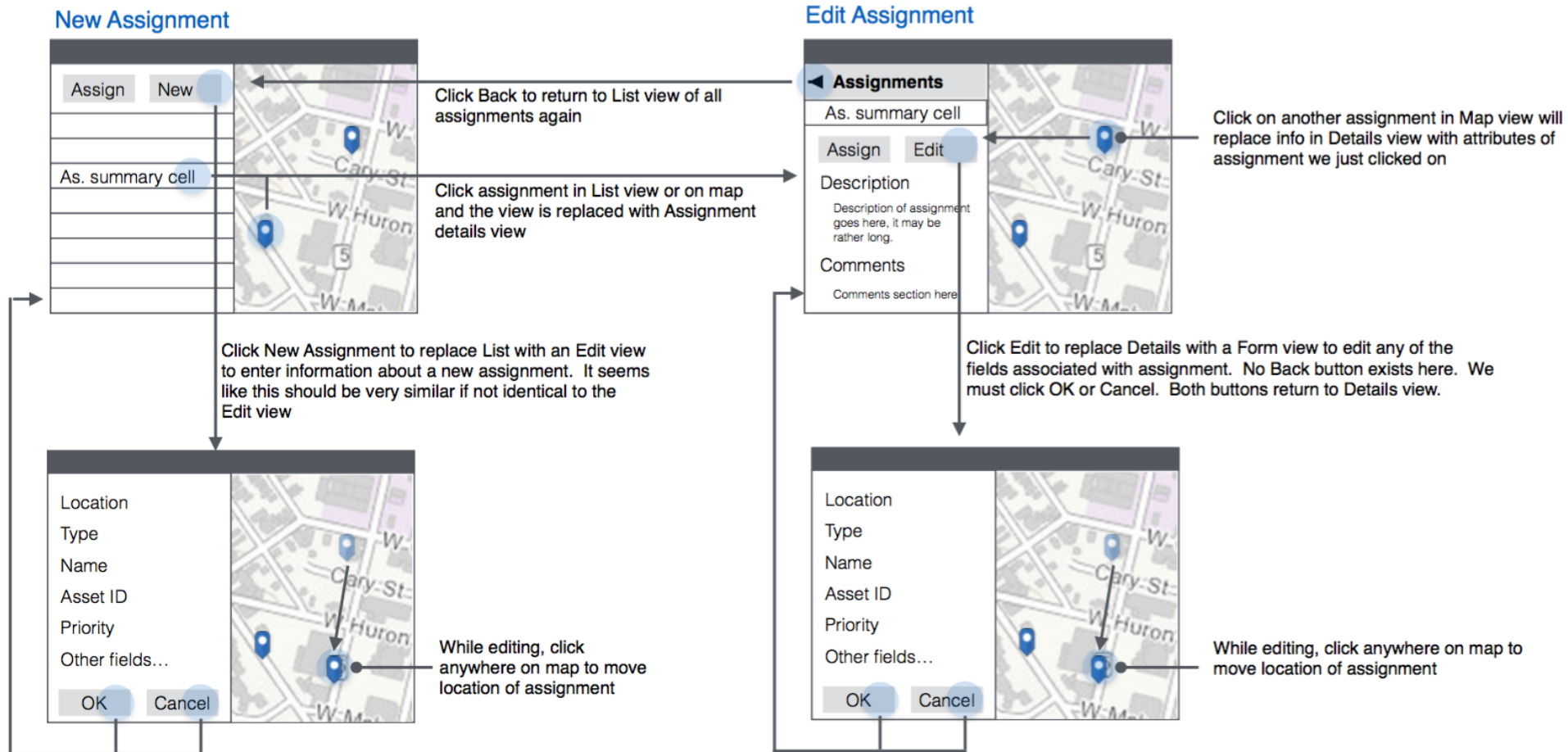
Today extension in Notification center

- Good Morning
- steps are clear and sunny 10%

fine am... temperature aware, workload aware



Mid-res Backoffice - Navigation Design

Details, assignment creation & editing



***Note** - there will be some additional design work for creating a new assignment from an existing asset on map. This design will be updated when Identify/mini-panel designs are done.

Backoffice Web App

 **Redlands Tree Service** Community  Nitro

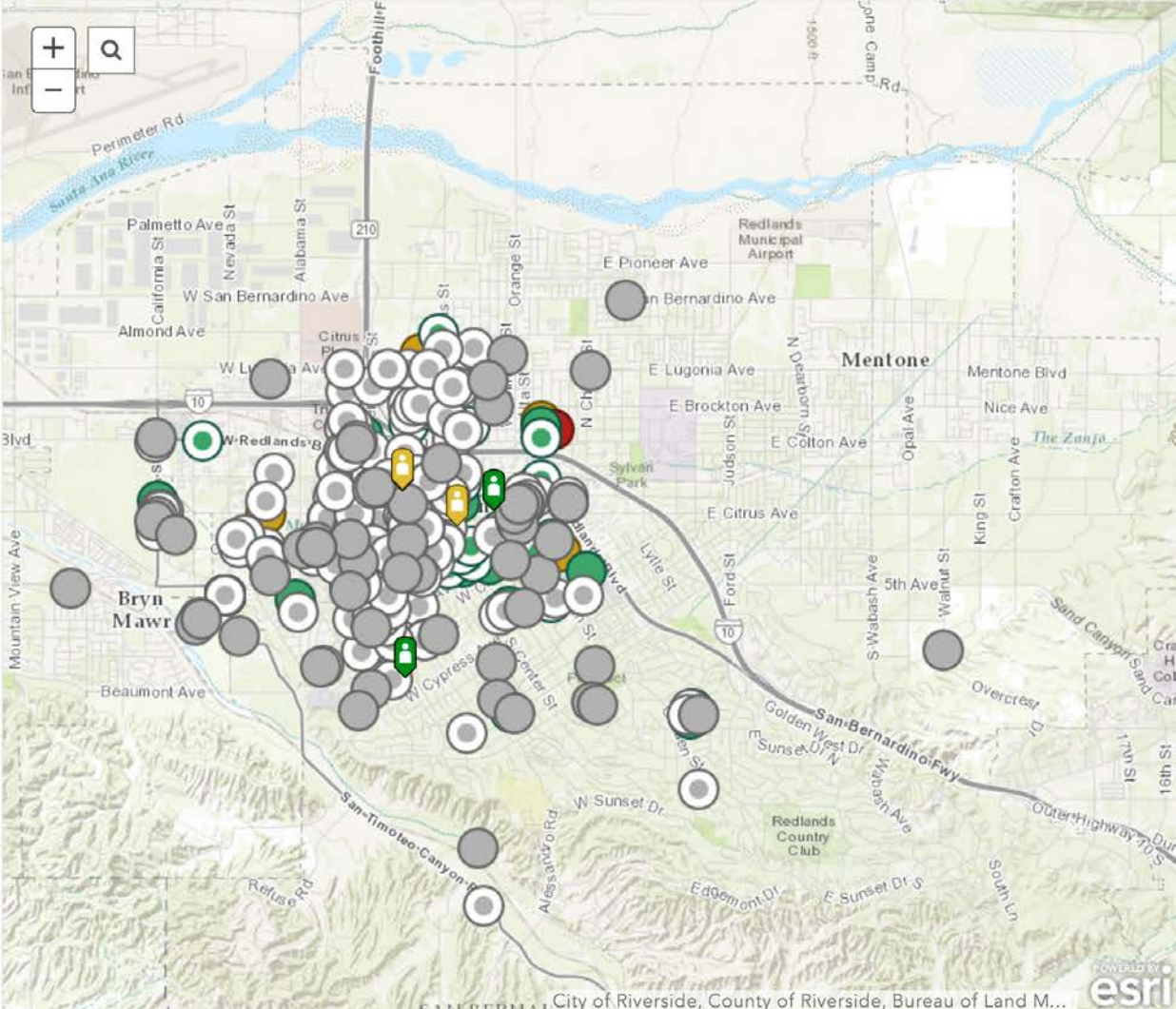
Assign [+ Assignment](#)


Status Due Priority Assignee Sort

539 assignments Showing 1-50 < >

- Tree Planting**
416 Tennessee St, Redlands, California, 92373
Low Priority
- Tree Planting**
135 Belmont Ct, Redlands, California, 92373
Critical
- Tree Planting**
1102 W Fern Ave, Redlands, California, 92373
Critical | ⌚ Due 6 months ago
- Tree Trimming**
606 Texas St, Redlands, California, 92374
Critical
- Tree Removal**
10 E State St, Redlands, California, USA
Critical | ⌚ Due 6 months ago
- Brick Over Tree Hole**
10997 California St, Redlands, California, 92373
Medium Priority
- Tree Planting**
10997 California St, Redlands, California, 92373
Medium Priority

Assignments Workers

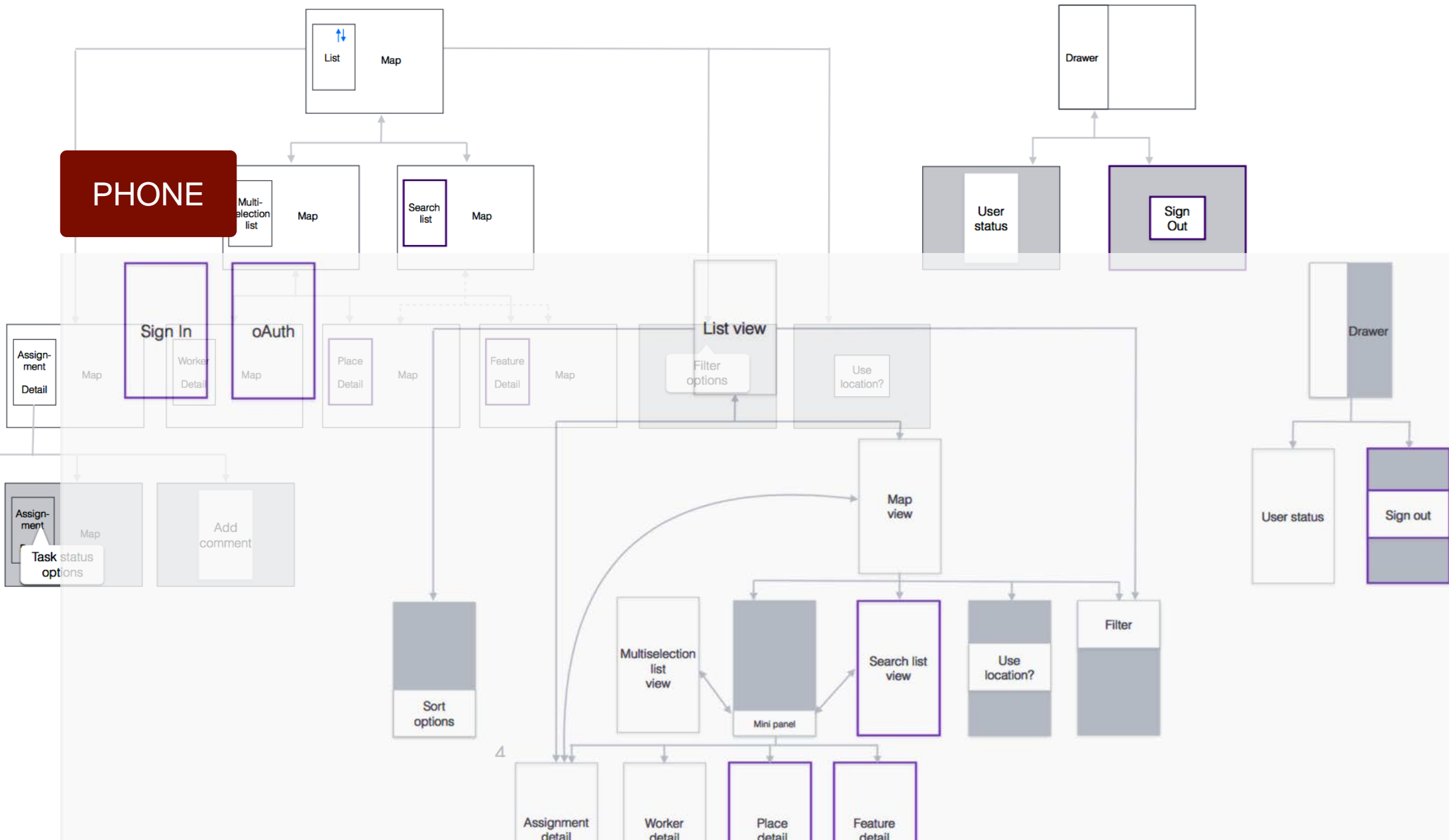
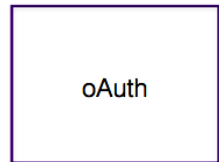


City of Riverside, County of Riverside, Bureau of Land M... 

TABLET

App's Navigational Structure

PHONE



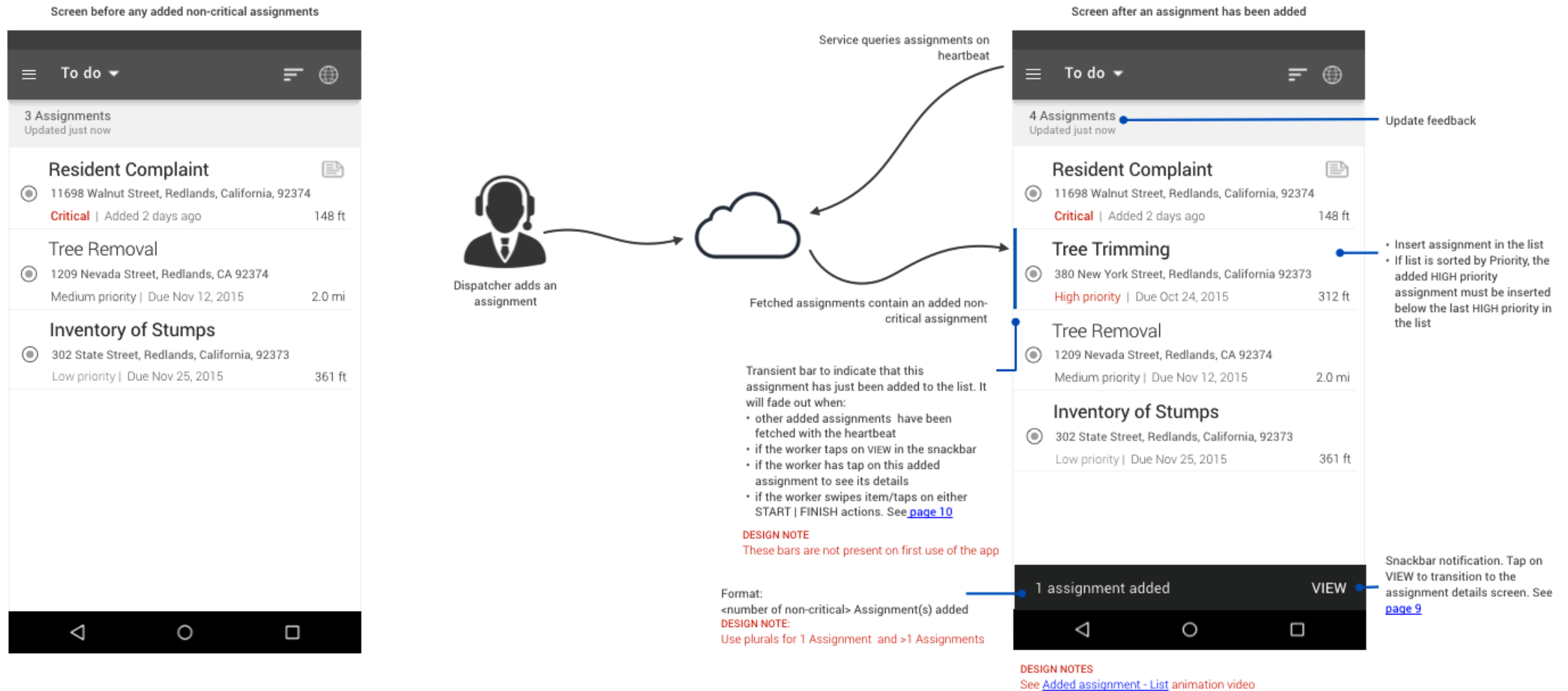
= Leverage existing framework

PHONE - Mid-res Mobile App

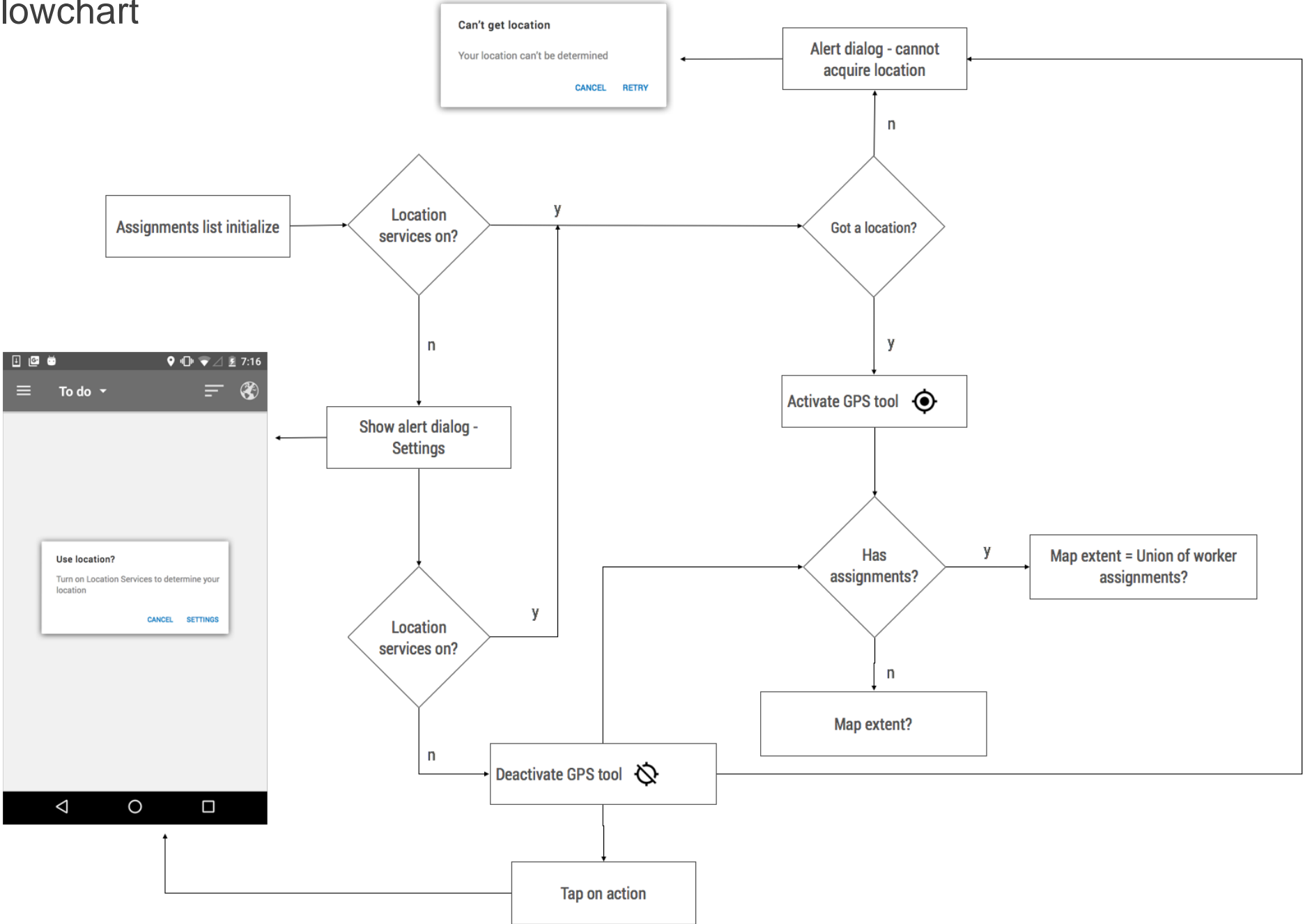
Assignments Messaging - List Screen - Non-critical

RULES

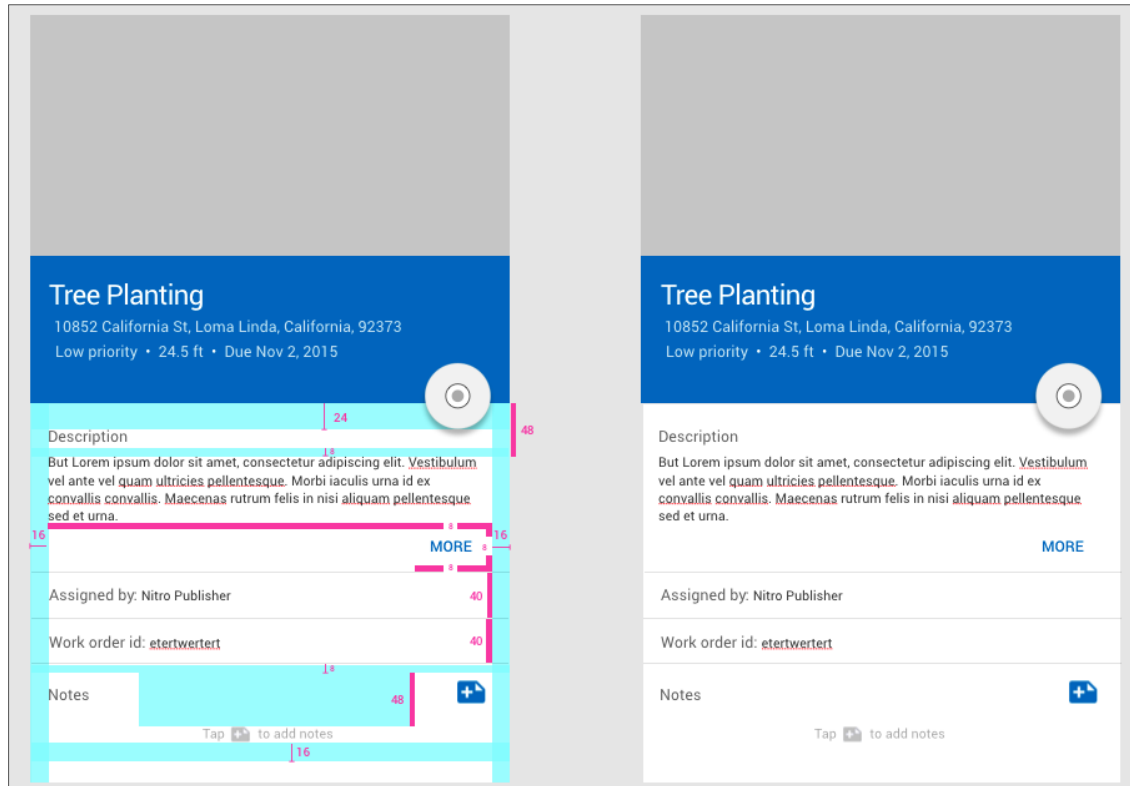
- when app is open must first cancel another sync requests
- it must schedule periodic syncs using the app's heartbeat
- a transient snackbar will notify the worker that has assignment(s) has been added to her/his TO DO list during that heartbeat period
- snackbar timeout is LENGTH_SHORT



Location Services - Flowchart



Material Design - Metrics & Keylines



Assignments details: Metrics/keylines & colors #248

Closed sabine opened this issue on Nov 13, 2015 - 4 comments

sabine commented on Nov 13, 2015

The *Assignment Details* design doc page 8 specifies the anatomy of the **details panel**. Some of this specs were ignored during feature implementation:

DETAILS HEADER

- ☑ Top padding: 18dp
- ☑ Bottom padding: 24dp
- ☑ Left padding: 16dp
- ☑ Right padding: 16dp

The diagram shows a grey header with the word 'MAP' in white. Below it is a blue bar with 'Tree Planting' in white, followed by the address '10852 California St, Loma Linda, California, 92373' and status 'Low priority • 24.5 ft • Due Nov 2, 2015'. A white circular button with a plus sign is on the right. Red and blue lines indicate padding: 16dp on the left and right, 18dp on the top, and 24dp on the bottom.

FAB

The diagram shows a red circular FAB with a white plus sign. A red square icon is shown inside a white circle, with a red line indicating a diameter of 24dp. A larger red circle indicates a diameter of 56dp.

Interior icon: 24 x 24dp
Floating action button circle: 56 x 56dp

Assignment type

The diagram shows a blue bar with the title 'Tree Planting', address '10852 California St, Loma Linda, California, 92373', and status 'Low priority • 24.5 ft • Due Nov 2, 2015'. A white circular button with a plus sign is on the right. Red and blue lines indicate padding: 16dp on the left and right, 18dp on the top, and 24dp on the bottom.

Note: Padding has been already added in the **details body**

Dividers

- ☐ Width: 1dp
- ☐ Color: #1F000000 (black 12% alpha)

Dispatcher id & Work order id

- ☐ Height: 40dp (no padding)
- Attribute name (the IDs)
 - ☐ textAppearance: @style/TextAppearance.AppCompat.Body1
 - ☐ textColor: #8A000000 (black 54% alpha)
 - ☐ textSize: 15sp
- Value
 - ☐ textAppearance: @style/TextAppearance.AppCompat.Body1
 - ☐ textColor: #B3000000 (black 70% alpha)

The diagram shows two white rectangular fields. The top field contains 'Assigned by: Nitro Publisher' and the bottom field contains 'Work order id: etertwertert'. Red and blue lines indicate padding: 16dp on the left and right, 40dp on the top and bottom.

Notes text

- ☐ textAppearance: @style/TextAppearance.AppCompat.Body1
- ☐ textSize: 15sp
- ☐ textColor: #8A000000 (black 54% alpha)
- ☐ String must be **Notes** (no upper case letters please)

Add notes action

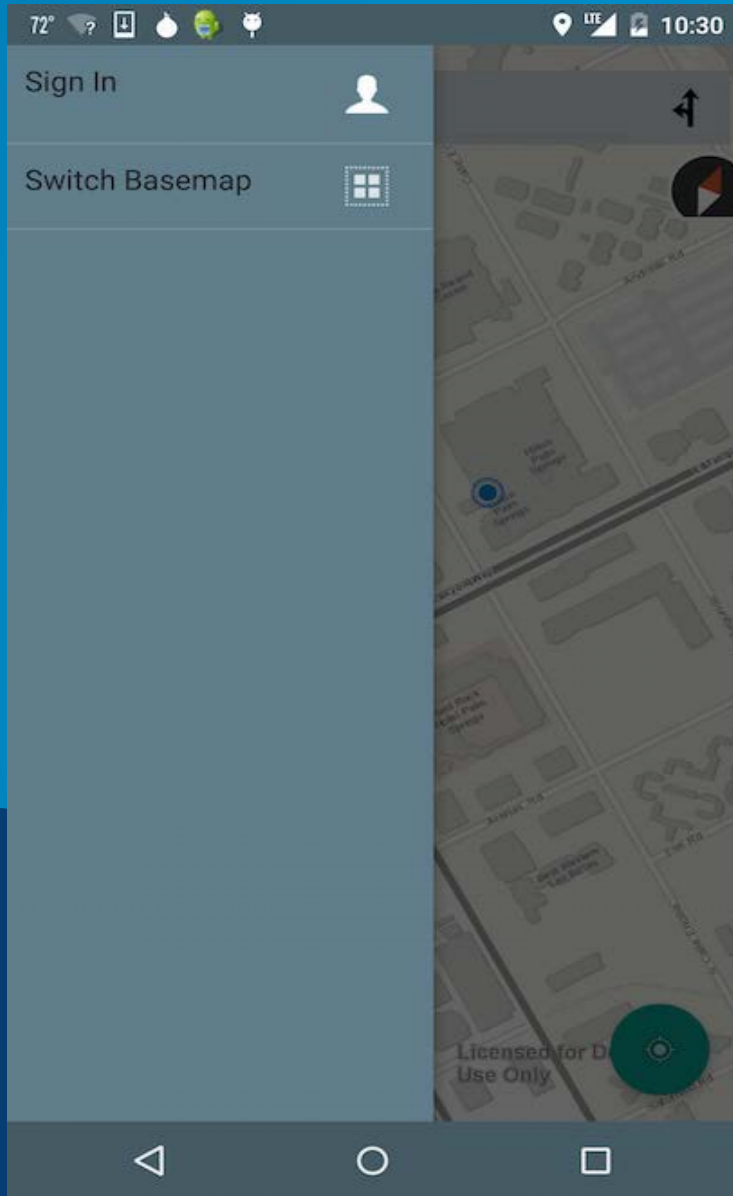
- ☐ Width & height: 36 dp
- ☐ Use assets from Google icon library - ic_note_add_white_24dp

Notes helper text

- ☐ Top padding 48dp
- ☐ textAppearance: @style/TextAppearance.AppCompat.Body1
- ☐ textColor: #4D000000
- ☐ use same asset as mentioned above black with 30% transparency

Added notes by worker

- ☐ textAppearance: @style/TextAppearance.AppCompat.Body1
- ☐ textColor: #8A000000



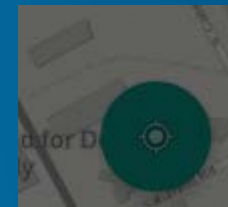
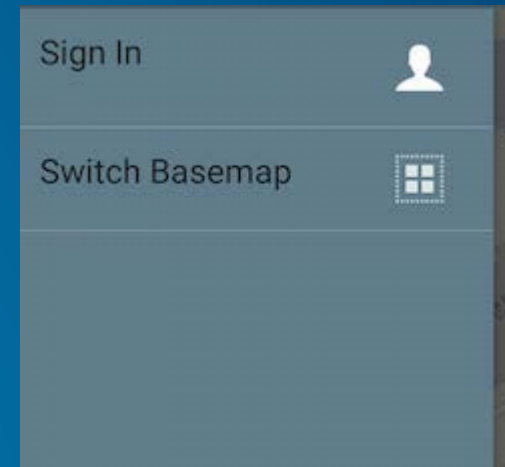
Building a map-centric app

Maps app demo - Dan

Maps App



- **Material**
 - **Navigation Drawer**
- **Search Widget**
- **Floating Compass Button**
- **Fragments**
 - **Routing Dialog**
 - **Directions Dialog**
 - **Basemaps Dialog**



Maps App Design

- **Working with webmaps**
- **Setting initial extent**
 - **Zooming to a location**
 - **Base of orientation for UX**
- **Dealing with device rotation**
 - <https://developers.arcgis.com/android/guide/handle-device-configuration-changes.htm>
- **Adding map tools / FABs**
 - **Compass/GPS**
 - **Floating Action Button**
 - **Routing**
 - **Automatic Routing from SearchBar**
 - **Point to point routing in Dialog**
 - **Using navigation draws**
 - **Log In**
 - **Switch Basemaps**

Working with webmaps

- **Webmaps are an interactive display of geographic information**
 - **Basemap**
 - **Set of data layers**
 - **Can include popup windows for information about data**
 - **Extent**
- **Hosted and shared through ArcGIS Online**
 - **Layers and basemaps can be referenced externally**

Dealing with device rotation

- **Patterns**

- **Retain Fragment instance**

- Use fragments to hold views in your app
- `setRetainState` prevents fragment from being destroyed
- Recommended only if you do not use WebMaps

- **Handle configuration changes yourself**

- Smooth transition
- MapViews do not need to be recreated
- Only way with WebMaps

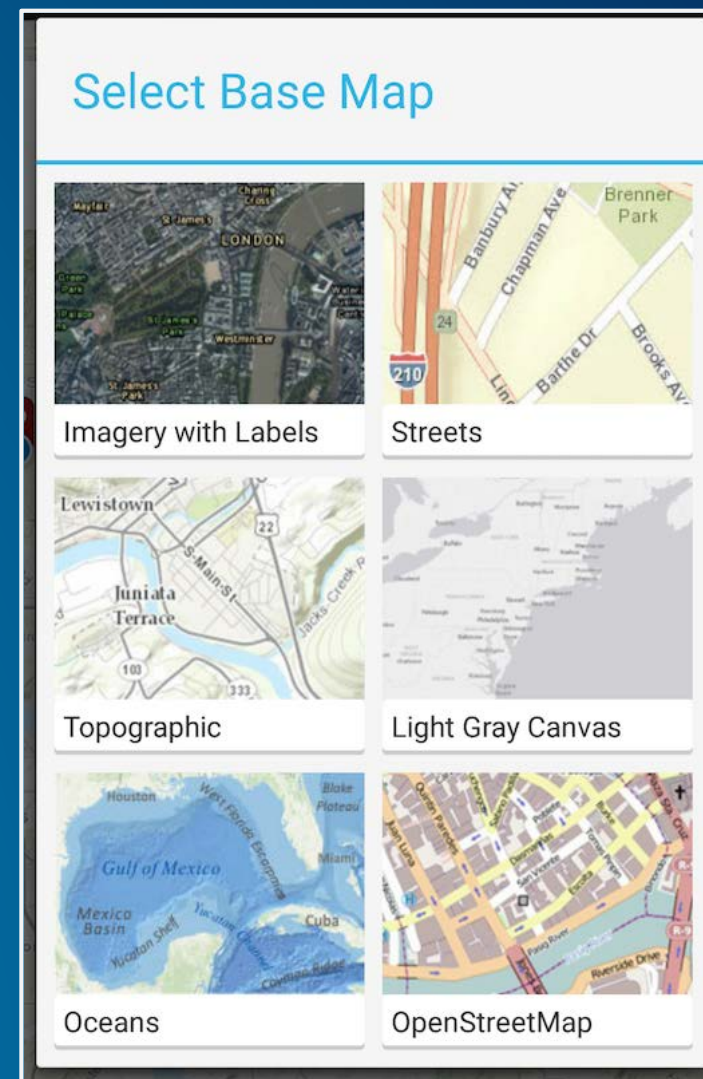
- **Recreate all objects**

- Create new mapping data objects each time an activity is created
- Most straightforward option
- Worst user experience
 - MapView must be entirely recreated
 - Fetch data from server every time recreated

```
<activity android:name=".ActivityName"  
    android:label="@string/app_name"  
    android:configChanges="orientation|screenSize|keyboard|keyboardHidden"  
    ...  
>
```

Maps App - Fragments Basemap

- **Basemap Dialog is a fragment**
 - Floats on top of Activity Window
- **Backed by a Basemaps Adapter**
 - Registers the listener for clicks on thumbnails
- **Listens for Click Events**
 - Opens the Basemap



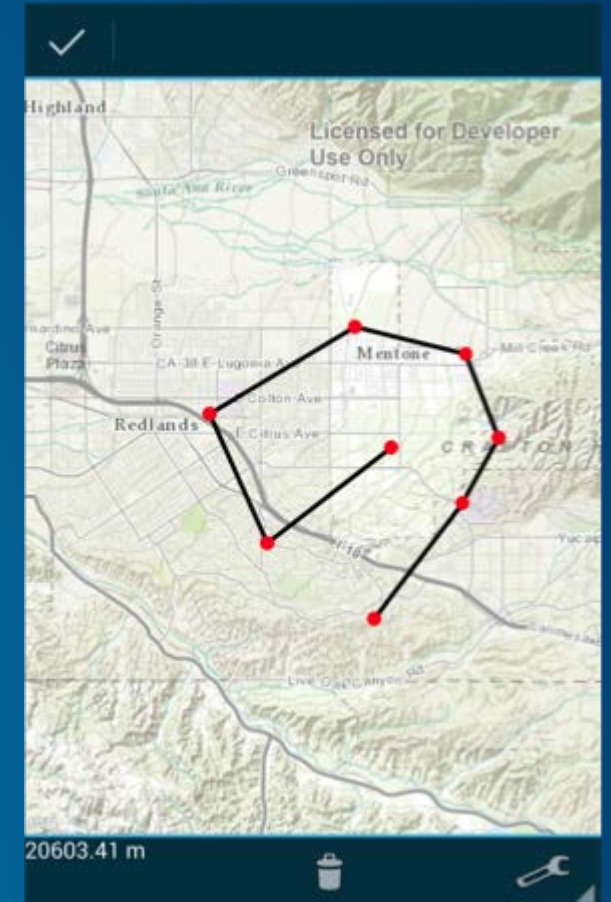
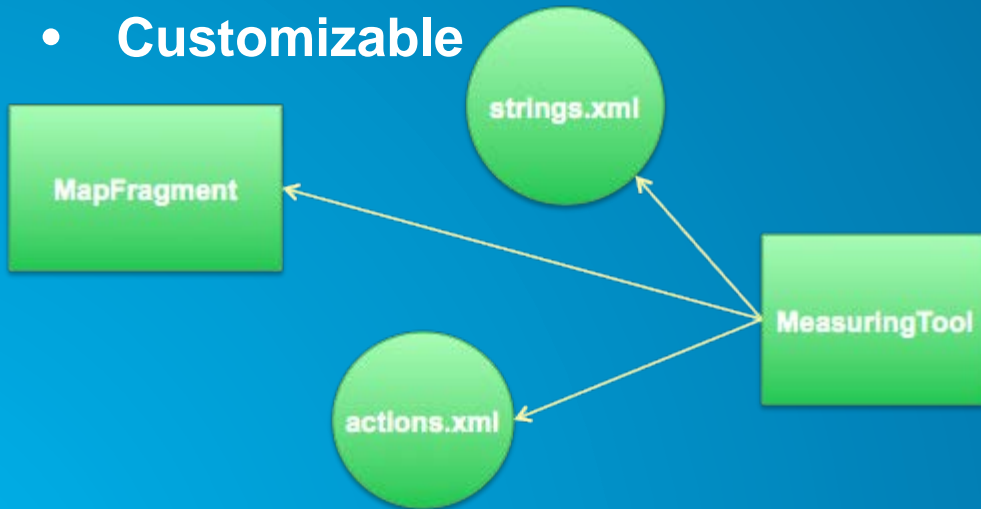
Maps App - Basemap Fragment

- Inflate layout
- Create View
- Register listener
- Set title

```
public View getView(final int position, View convertView, ViewGroup parent) {  
  
    // Inflate view unless we have an old one to reuse  
    View newView = convertView;  
    if (convertView == null) {  
        LayoutInflater inflater = (LayoutInflater)  
            mContext.getSystemService(Context.LAYOUT_INFLATER_SERVICE);  
        newView = inflater.inflate(R.layout.basemap_image, null);  
    }  
  
    // Create view for the thumbnail  
    ImageView image = (ImageView)  
        newView.findViewById(R.id.basemap_grid_item_thumbnail_imageview);  
    image.setImageBitmap(items.get((position)).itemThumbnail);  
  
    // Register listener for clicks on the thumbnail  
    image.setOnClickListener(new OnClickListener() {  
        @Override  
        public void onClick(final View view) {  
            mListener.onBasemapItemClicked(position);  
        }  
    });  
  
    // Set the title and return the view we've created  
    TextView text = (TextView) newView.findViewById(R.id.basemap_grid_item_title_textview);  
    text.setText(items.get((position)).item.getTitle());  
    return newView;  
}
```


Maps App - Configurable Measure Tool

- Measure Tool provided by ArcGIS Android Toolkit
- Contextual action bar
- Works in map centric apps
- Easy to integrate into your own app
- Customizable



Maps App - Configurable Measure Tool

```
// initialize some resources for the measure tool, optional.
Unit[] linearUnits = new Unit[] {
    Unit.create(LinearUnit.Code.CENTIMETER),
    Unit.create(LinearUnit.Code.METER),
    Unit.create(LinearUnit.Code.KILOMETER),
    Unit.create(LinearUnit.Code.INCH),
    Unit.create(LinearUnit.Code.FOOT),
    Unit.create(LinearUnit.Code.YARD),
    Unit.create(LinearUnit.Code.MILE_STATUTE) };
SimpleMarkerSymbol markerSymbol = new SimpleMarkerSymbol(
    Color.BLUE, 10,
    com.esri.core.symbol.SimpleMarkerSymbol.STYLE.DIAMOND);
SimpleLineSymbol lineSymbol = new SimpleLineSymbol(Color.YELLOW, 3);
SimpleFillSymbol fillSymbol = new SimpleFillSymbol(Color.argb(100,
    0, 225, 255));
fillSymbol.setOutline(new SimpleLineSymbol(Color.TRANSPARENT, 0));
```

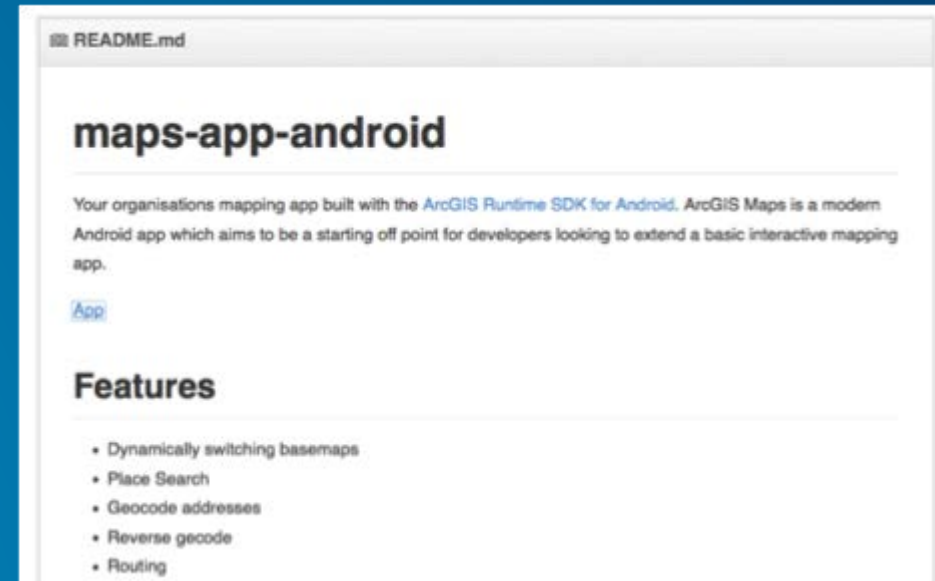
Maps App - Configurable Measure Tool

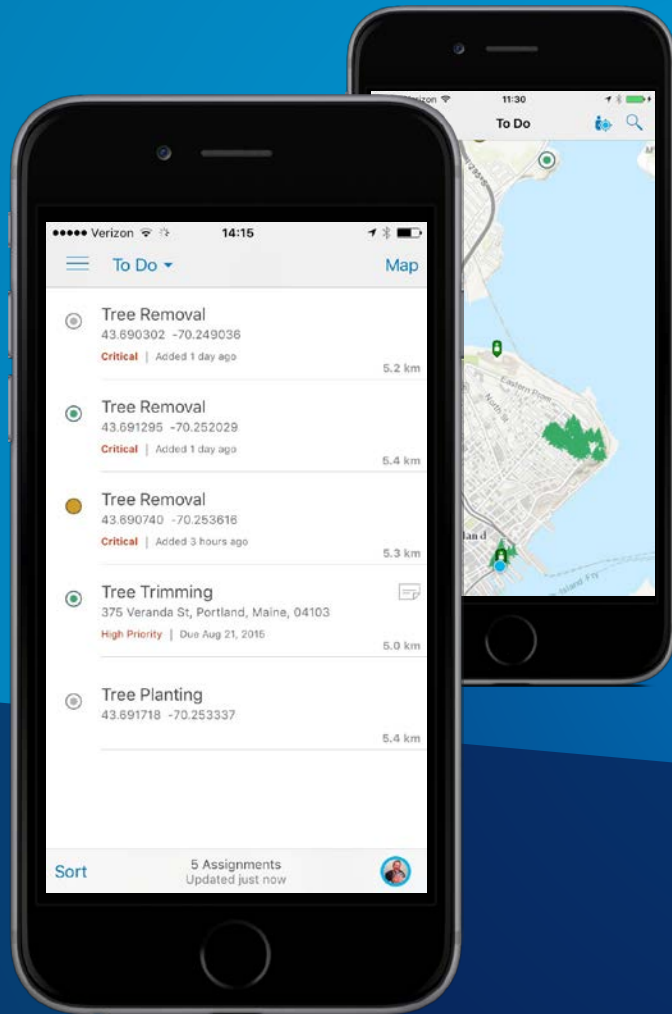
```
// create the tool, required.  
MeasuringTool measuringTool = new MeasuringTool(MapFragment.mMapView);  
// customize the tool, optional.  
measuringTool.setLinearUnits(linearUnits);  
measuringTool.setMarkerSymbol(markerSymbol);  
measuringTool.setLineSymbol(lineSymbol);  
measuringTool.setFillSymbol(fillSymbol);  
  
// fire up the tool, required.  
startActionMode(measuringTool);
```


Maps App on GitHub



- <https://github.com/Esri/maps-app-android>
- Get involved
- Report Issues
- Contribute Code
 - Fork it
 - Clone it
 - Configure remotes
 - Send pull requests





Searching for
features
Will

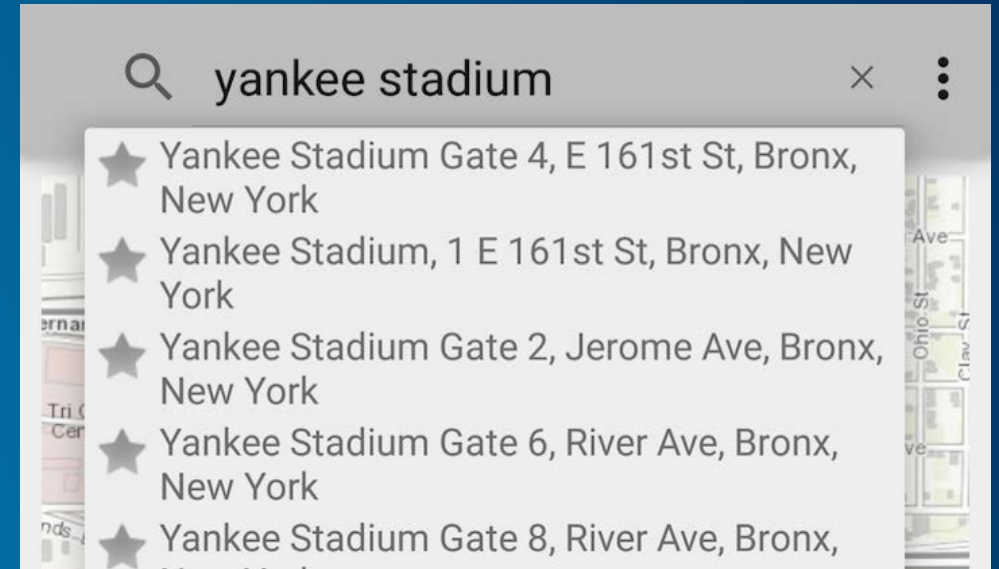
Geocoding

UX

- Use in Map Centric apps
 - Android search widget or custom widget
- Use in map-less apps
 - Location first apps, use location and geocode results to drive the rest of the app

API

- **Geocode using the LocatorTask**
 - Using an ArcGIS Geocode Service or local Locator (from ArcGIS Desktop/Pro)
- Geocode point from address or POI
- Reverse geocode a point to an address
- Supports suggestions



Code to know: Locators

- Use ArcGIS Online World Locator
- Can find:
 - Street addresses
 - Administrative place names
 - Postal / zip codes
 - Points of interest
 - Businesses
- Locator.find
- Locator.suggest

```
mLocator = Locator.createOnlineLocator();
```

```
LocatorFindParameters fParams =  
    new LocatorFindParameters(searchFor);  
fParams.setSearchExtent(searchExtent, mMapSr);  
fParams.setOutSR(mMapSr);  
fParams.setOutFields(mOutFields);  
if(mLDM != null) {  
    Point currentPoint = getAsPoint(mLDM.getLocation());  
    fParams.setLocation(currentPoint, mMapSr);  
}  
mLocator.find(fParams, findCallback);
```

Maps App - Search Widget

- MapFragment - Search widget is an instance of **SearchView**

```
// Setup the listener when the search button is pressed on the keyboard
mSearchview.setOnQueryTextListener(new OnQueryTextListener() {

    @Override
    public boolean onQueryTextSubmit(String query) {
        onSearchButtonClicked(query);
        mSearchview.clearFocus();
        return true;
    }

    @Override
    public boolean onQueryTextChange(String newText) {
        if(mLocator == null)
            return false;
        getSuggestions(newText);
        return true;
    }
});
```

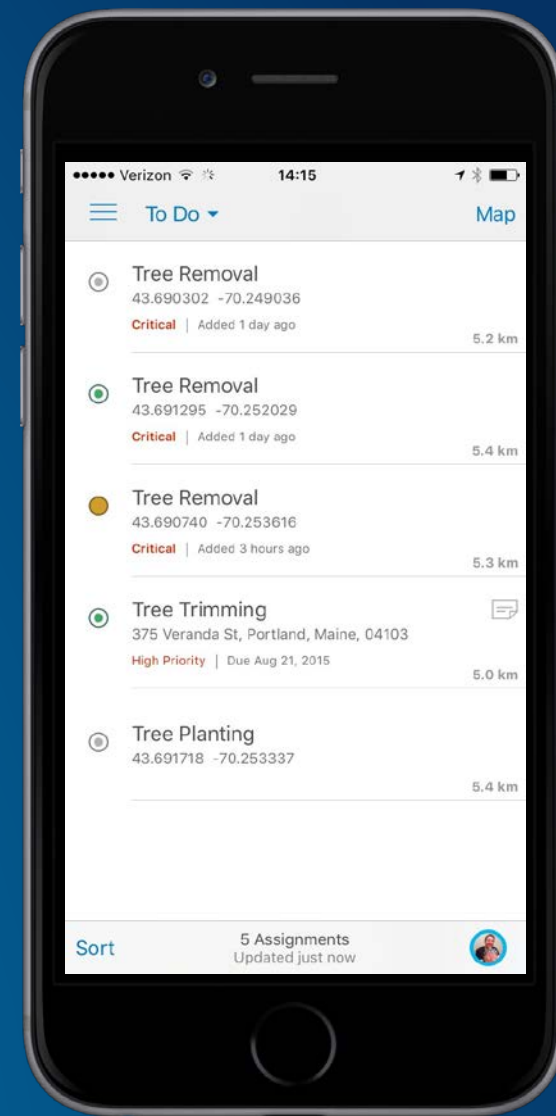
Searching for features

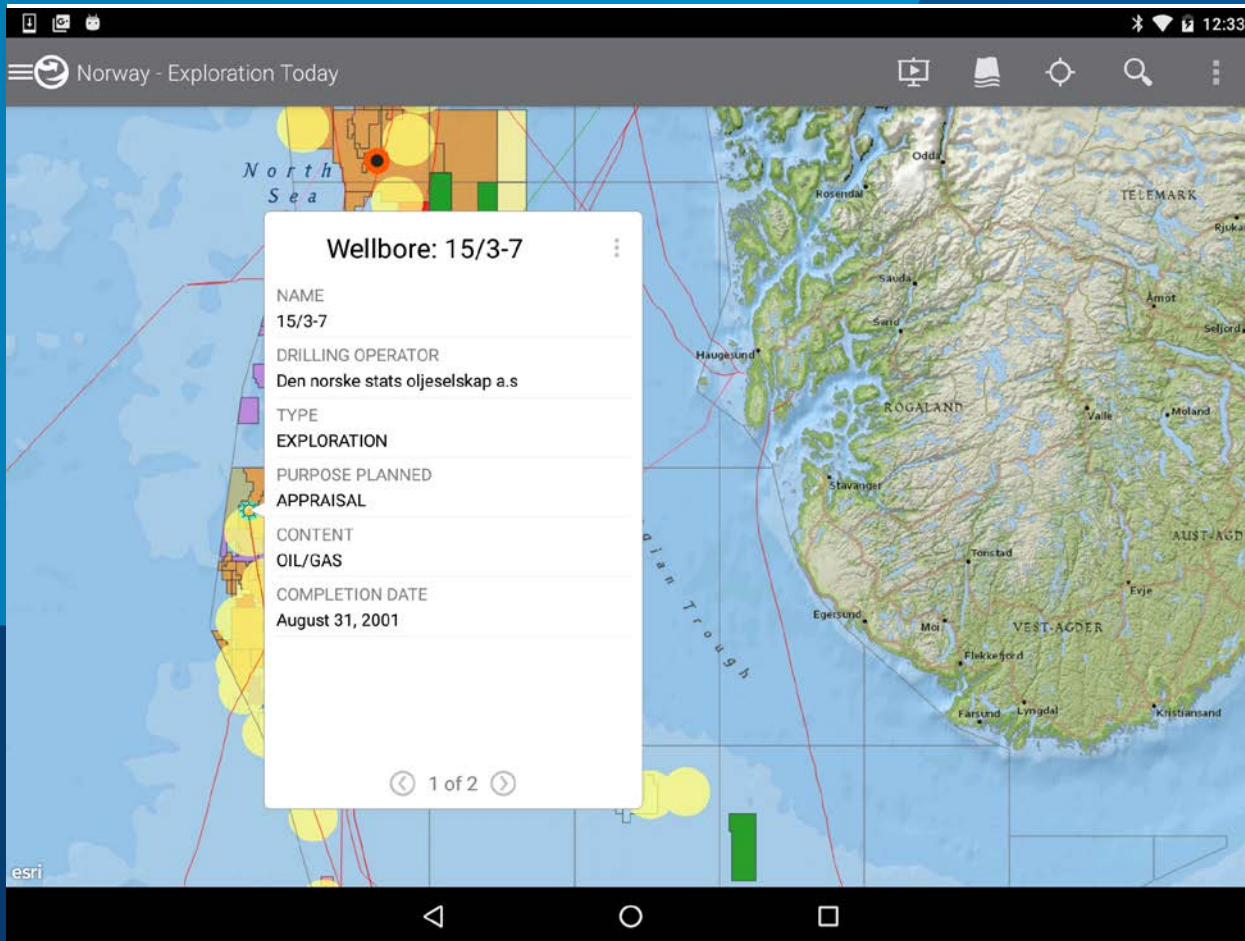
UX

- Search widget in a map centric app
- List driven app
- MapView interaction (onSingleTapListener)

API

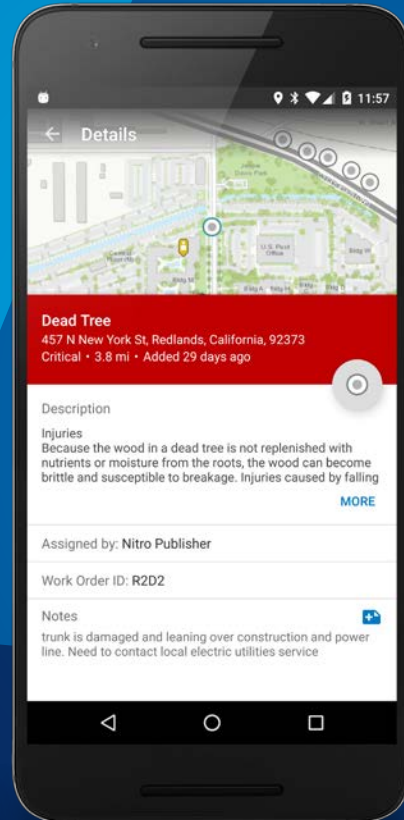
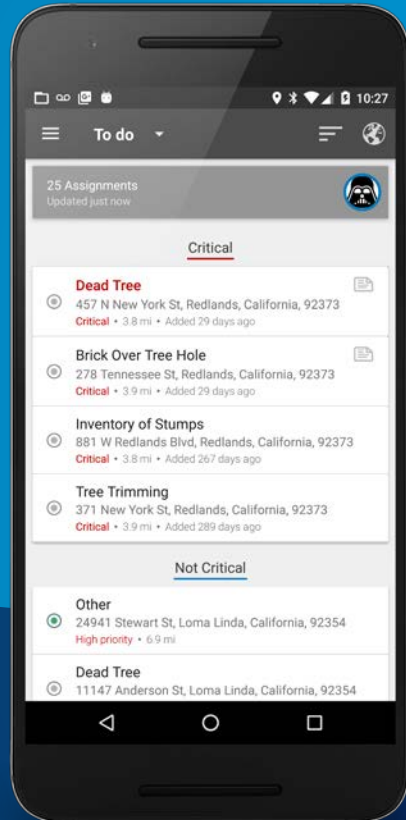
1. Search for features in a feature layer - Using `FeatureTable.query()`
 - Use when you have a feature layer in your map
2. Search for features using the `QueryTask`
 - Use when you don't have a map
 - Use for sublayers in an `ArcGISMapServiceLayer` or `ArcGISTiledLayer`
3. Select features in feature layers and graphics layers using `fl.getFeatureIds()` or `gl.getGraphicIds()`
 - Hit test or identify for features
 - Use when you have screen interaction





Searching for features in explorer

Sabine – use of FeatureTable query



Searching for features in Workforce

Sabine – use of QueryTask

Code to know: FeatureTable queries

- Set up QueryParameters
- Execute with callback or an AsyncTask (or your preferred async pattern)
- ServiceFeatureTable example below uses a callback

```
// Build query predicates to construct a query where clause from selected values.
String damageType = String.valueOf(mDamageSpinner.getSelectedItem());
String primCause = String.valueOf(mCauseSpinner.getSelectedItem());
String whereClause = "typedamage LIKE '" + damageType + "' AND primcause LIKE '" + primCause + "'";

// Create query parameters, based on the constructed where clause.
QueryParameters queryParams = new QueryParameters();
queryParams.setWhere(whereClause);

// Execute the query and create a callback for dealing with the results of the query.
featureServiceTable.queryFeatures(queryParams, new CallbackListener<FeatureResult>() {

    @Override
    public void onError(Throwable ex) {
        // Highlight errors to the user.
        showToast("Error querying FeatureServiceTable");
    }

    @Override
    public void onCallback(FeatureResult objs) {
```

Code to know: QueryTask

- Use QueryTask within AsyncTask.doInBackground()

```
String url = queryArray[0];
QueryParameters qParameters = new QueryParameters();
String whereClause = queryArray[1];
SpatialReference sr = SpatialReference.create(102100);
qParameters.setGeometry(mMapView.getExtent());
qParameters.setOutSpatialReference(sr);
qParameters.setReturnGeometry(true);
qParameters.setWhere(whereClause);

QueryTask qTask = new QueryTask(url);

try {
    return qTask.execute(qParameters);
}
```

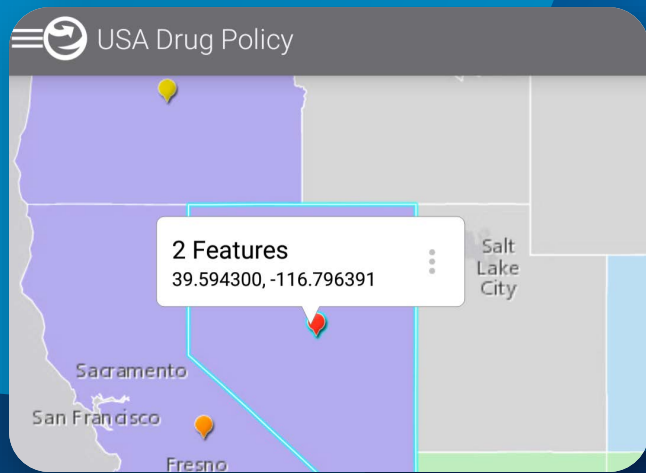
- Show results in a list from onPostExecute()

```
if (results != null) {
    int size = (int) results.featureCount();
    for (Object element : results) {
        progress.incrementProgressBy(size / 100);
        if (element instanceof Feature) {
            Feature feature = (Feature) element;
            // turn feature into graphic
            Graphic graphic = new Graphic(feature.getGeometry(),
                feature.getSymbol(), feature.getAttributes());
            // add graphic to layer
            graphicsLayer.addGraphic(graphic);
        }
    }
}
```


Code to know: feature and graphic selection

- Set `MapView.setOnSingleTapListener`
- Use parameters to find graphics/features at screen coordinates and select them
- Use attributes to show information

```
final OnSingleTapListener mapTapListener = (x, y) -> {  
  
    int[] graphicIDs = mResultsLayer.getGraphicIDs(x, y, 25);  
    if (graphicIDs != null && graphicIDs.length > 0) {  
        if (graphicIDs.length > 1){  
            int id = graphicIDs[0]; // Only select first  
            graphicIDs = new int[] { id };  
        }  
  
        mResultsLayer.clearSelection();  
        mResultsLayer.setSelectedGraphics(graphicIDs, true);  
        Graphic gr = mResultsLayer.getGraphic(graphicIDs[0]);  
        updateContent(gr.getAttributes());  
    }  
}
```

Show feature data

Dan

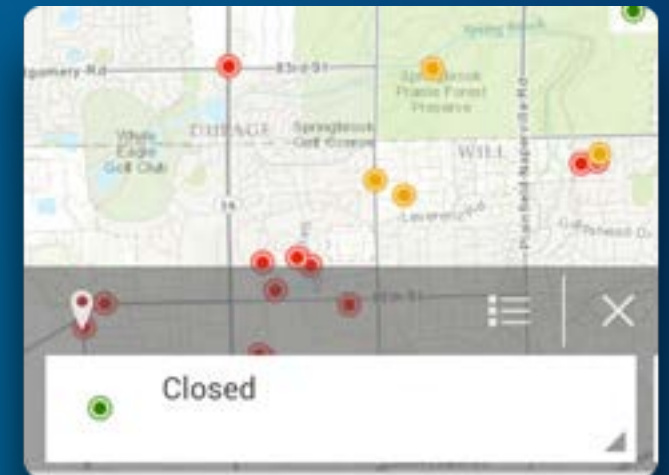
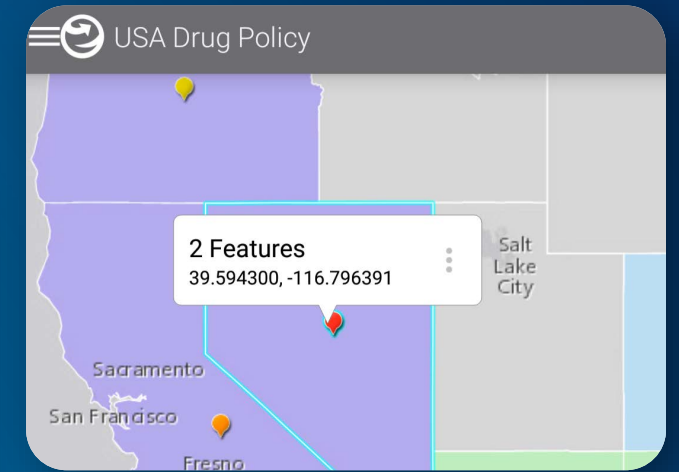
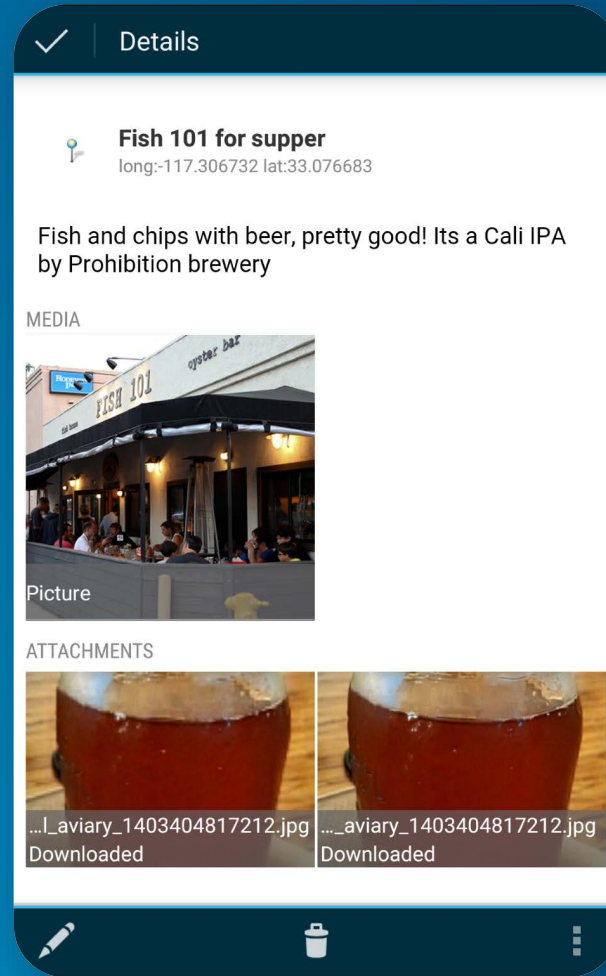
Showing feature data

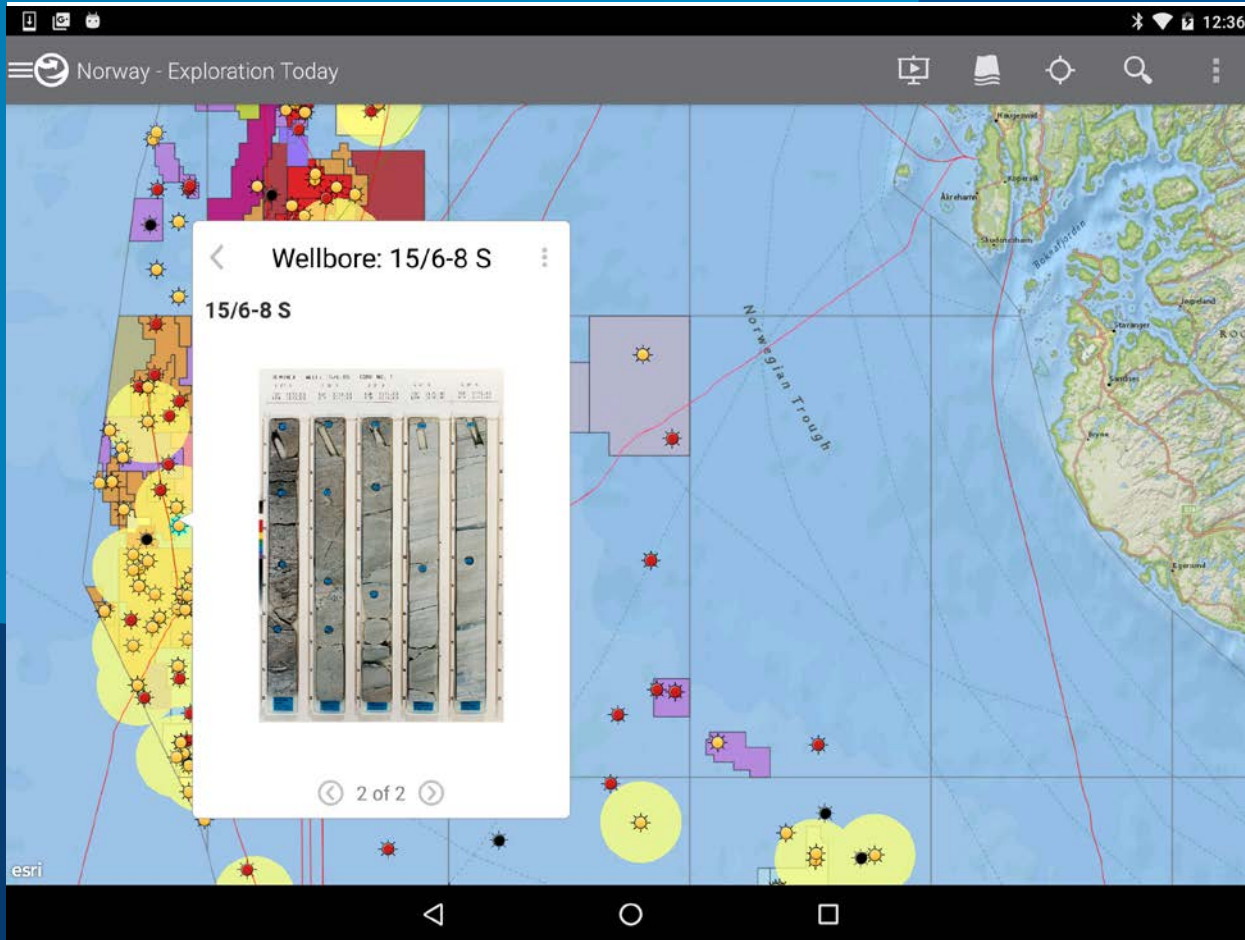
UX

- Over the map
- In a separate view from the map

API

- Callout
- Popup
- On your own... ;)





Explorer demo

Sabine

Callout Pattern Workflow

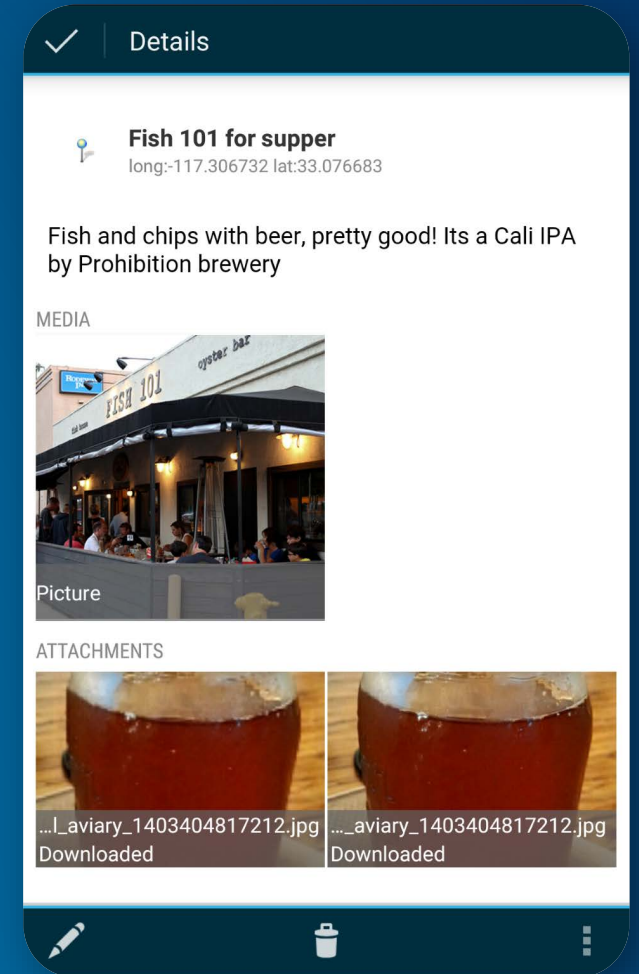
- Create an XML style file
 - Specific callout style attributes available
 - Set the title color, text size, background color
- Create an Android Layout
 - Define text and image elements
- Set the style and content of callout
 - Inflate the layout
 - Populate it with info retrieved from Graphic
 - Set layout of callout

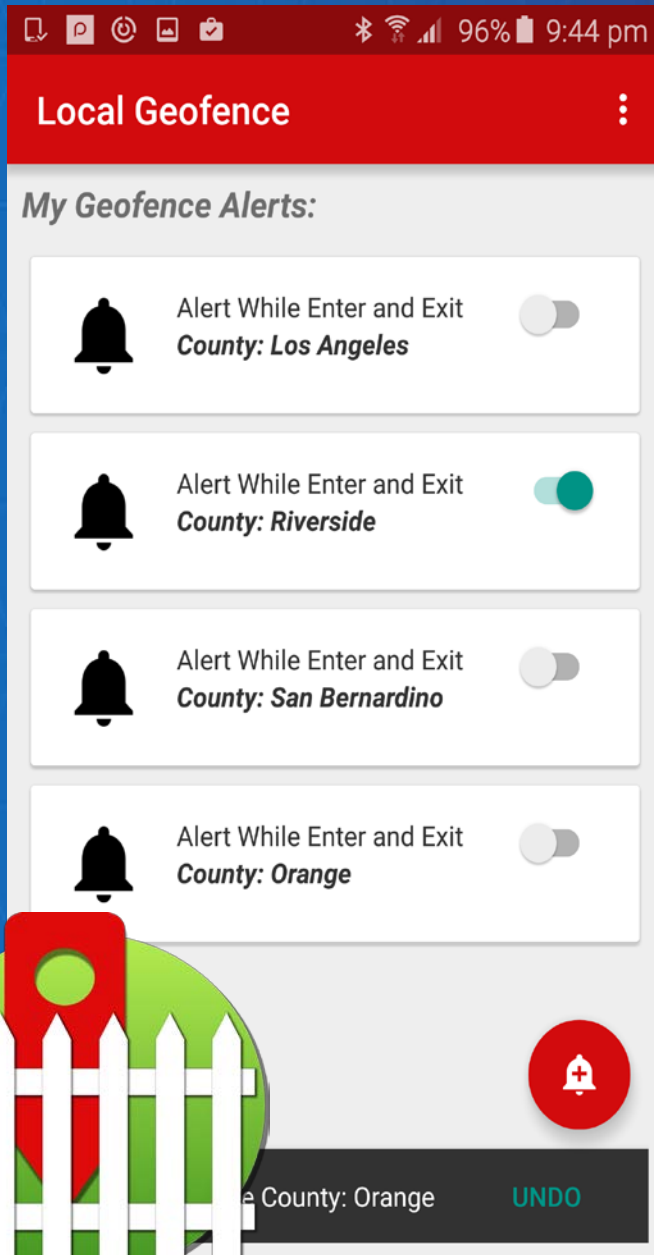


```
callout.setStyle(R.xml.countypop);  
String countyName = (String) gr.getAttributeValue("NAME");  
String countyPop = gr.getAttributeValue("POP07_SQMI").toString();  
// Sets custom content layout to callout.
```


Popups Pattern workflow

- Display information using pop-ups
 - Instantiate PopupContainer
 - Create a Popup and add it to the PopupContainer
 - Display PopupContainerView
- Handle user interaction
- Edit information using pop-ups
- Customize UI





Location-as-alerts

Local Geofence app demo
& code

Rama Chintapalli

<https://github.com/Esri/arcgis-runtime-demos-android/tree/master/2016-DS/LocalGeofence>

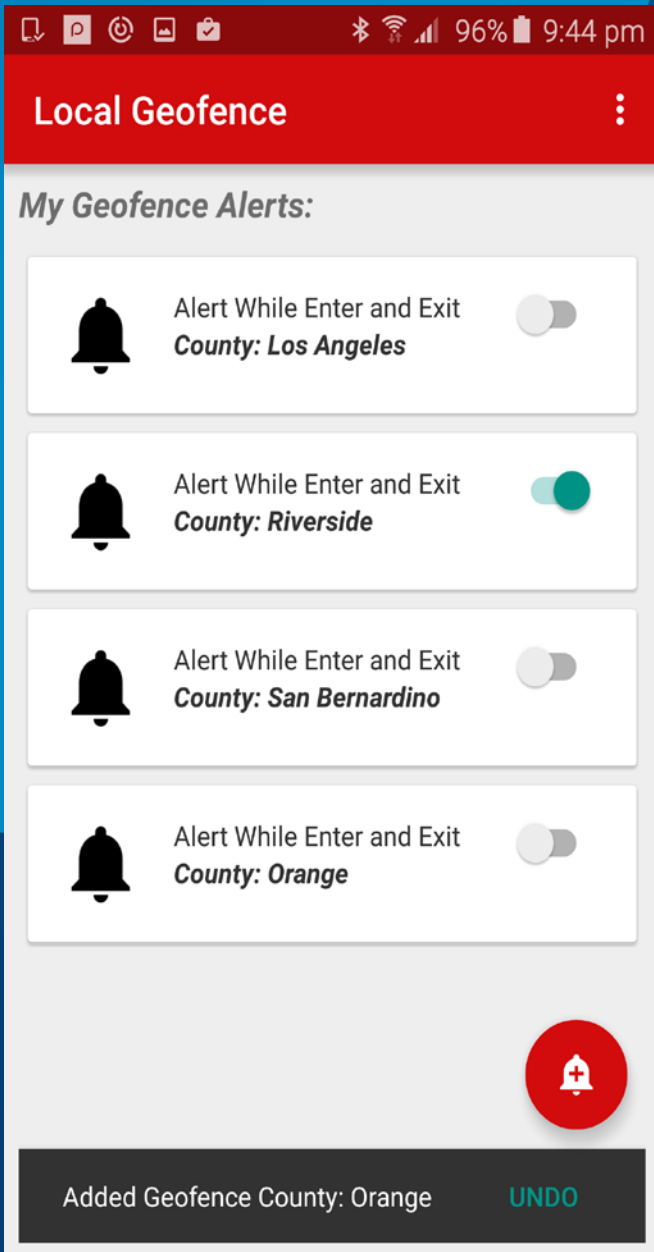
Location as alerts

UX

- **No map required!**
- **Notifications for the user run in the background**

API

- **Option 1: DIY**
 - **Androids LocationServices**
 - **Geometry engine**
 - **Geodatabase and FeatureTable**
- **Option 2: Android Geofencing API**
 - **Android LocationServices**
 - **Android GeofencingAPI**
 - **Geodatabase and FeatureTable**



Local geofence demo

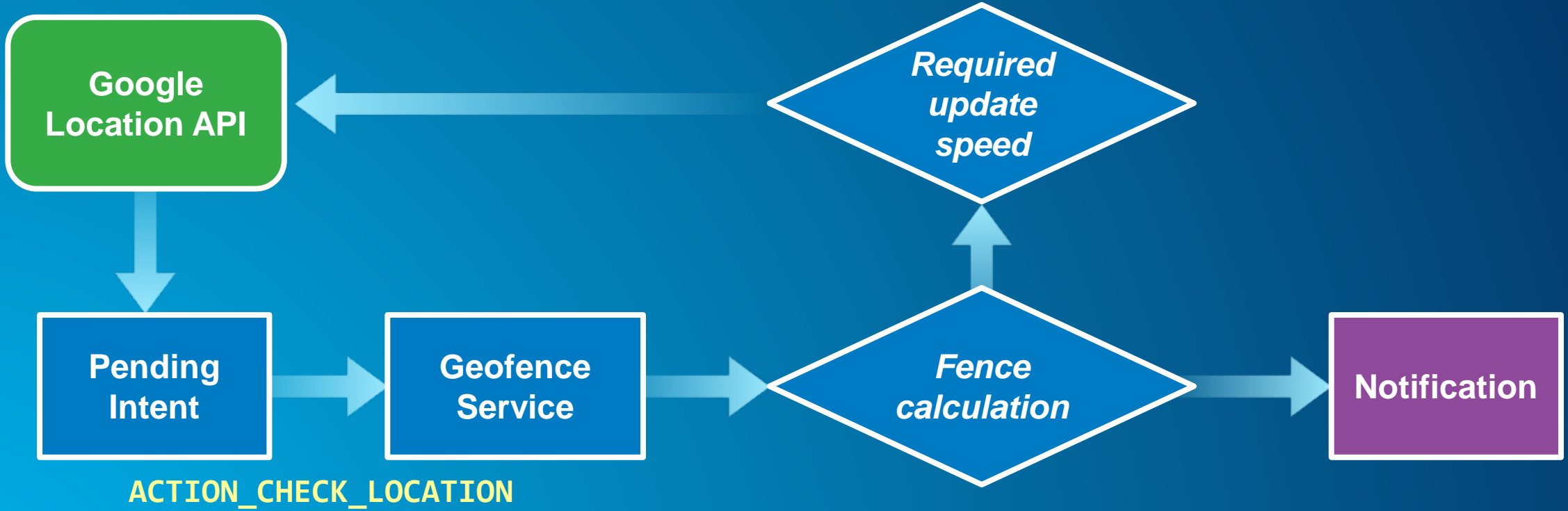
Rama

Local Geofence App

- **Uses offline geodatabase to store geofence features**
- **Notifications for alerting user when entering or exiting a polygon geofence**
 - Always available, outside app
 - User can control priority, sensitivity (lock screen)
- **Service for monitoring location updates**
 - Combines Google Fusion API and GeometryEngine
 - Increase update frequency when close by
- **Activity based UX allows user to choose fence**
 - Recycler View, Card View, Floating Action Button, Snack Bar

Geofence app process

`FusedLocationApi.requestLocationUpdates`



Code to know: GeometryEngine

- Proximity2DResult gets nearest location on fence boundary
- geodesicDistance
- within
- project of fence feature to WGS84

```
private static boolean isCloseToFence(Point location) {
    Proximity2DResult proximity =
        GeometryEngine.getNearestCoordinate(
            mFenceWgs84, location, true);
    double distanceGeodesic =
        GeometryEngine.geodesicDistance(location,
            proximity.getCoordinate(), mWgs84Sr,
            mProximityUnits);

    return (distanceGeodesic < IS_CLOSE_DISTANCE);
}
```

Code to know: Notifications

- Build and send notification using Feature attribute
- Notification API changes within SDK supported SDK versions

```
Notification.Builder nBuilder = new Notification.Builder(this);
nBuilder.setContentTitle("Alert! Entered " + LocalGeofence.getFeatureName());
nBuilder.setContentText(LocalGeofence.getSubtitle());
nBuilder.setSmallIcon(R.drawable.ic_fence_simple);

Bitmap largeIcon = BitmapFactory.decodeResource(getResources(),
    R.drawable.ic_geofence_bright);
nBuilder.setLargeIcon(largeIcon);

// Support Android SDK version 14 and up: API changes at 16
if (Build.VERSION.SDK_INT < 16) {
    notificationManager.notify(NOTIFICATION_ID, nBuilder.getNotification());
}
else {
    notificationManager.notify(NOTIFICATION_ID, nBuilder.build());
}
```