



# ESRI DEVELOPER SUMMIT

10-12 November | Berlin, Germany



# Building Great Mobile Apps: Tips and Tricks

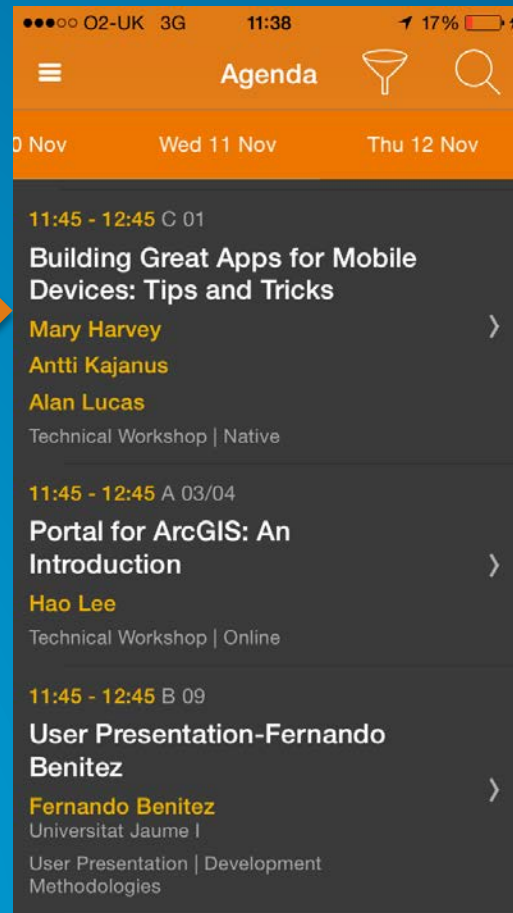
Antti Kajanus, Alan Lucas, Mary Harvey

# Please Take Our Survey!

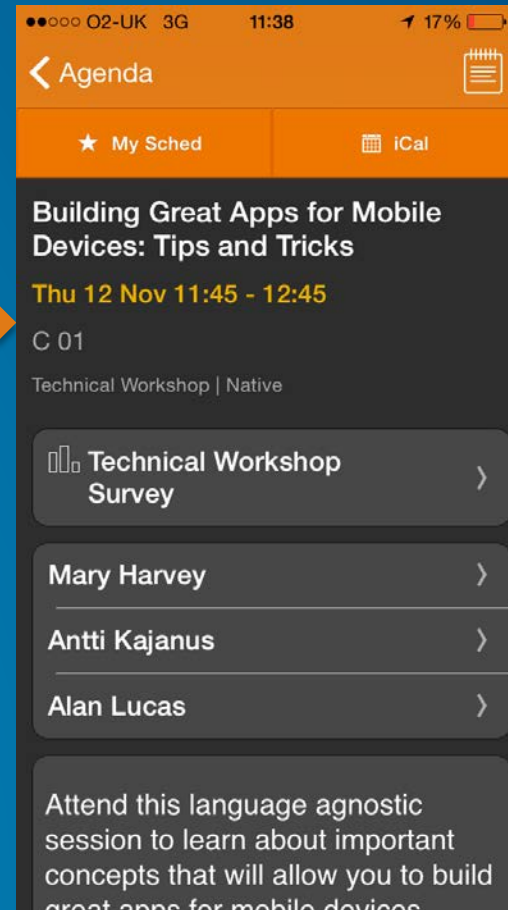
Download the Esri Events app  
and find your event



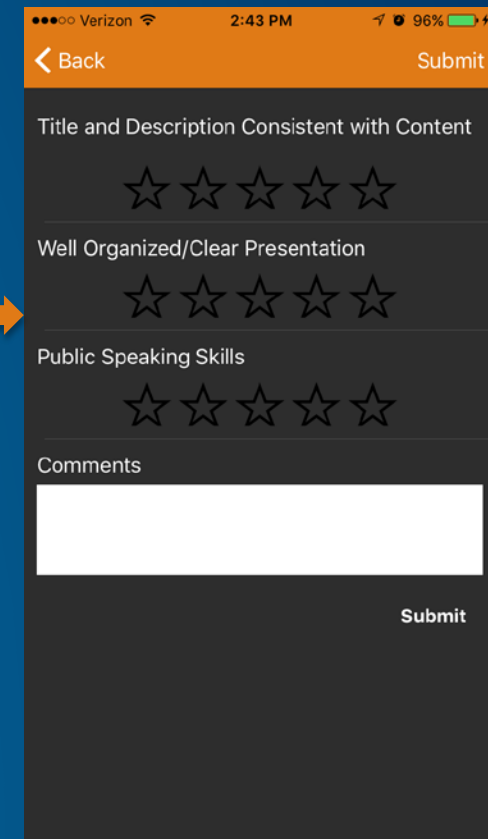
Select the session



Select  
"Technical Workshop Survey"



Complete Answers  
and Select "Submit"





# Building Great Mobile Apps: Tips and Tricks

# How does your app stand out from the crowd?

- **Must have product**
- **Intuitive**
- **Great user experience**
- **Fast**
- **Honorable**
- **Current data**
- **Correct price**
- **No bugs, crashes**
- **Marketing**
- ....



# Agenda

- **User Interface and User Experience (Alan Lucas)**
- **Data considerations (Mary Harvey)**
- **Performance (Antti Kajanus)**
- **Questions**

# User Interface and Experience

Alan Lucas

# What is a good UI/UX for a location based app?

- **Who are you building your app for?**
  - GIS analysts - tools, tools, tools on a toolbar
  - Knowledge workers - minimum tool set to get the job done
  - Consumers - just what they need and make it look good
- **Where are your users using it?**
  - Office based - wifi, indoors
  - Outdoors - offline, glare, gloves?
- **Whatever you do, don't make it slow or annoying!**
  - Efficient tooling, fast screen loads, fast lists, fast maps!
  - Make alerts appropriate for usage





## Some more questions to ask before you start cranking code...

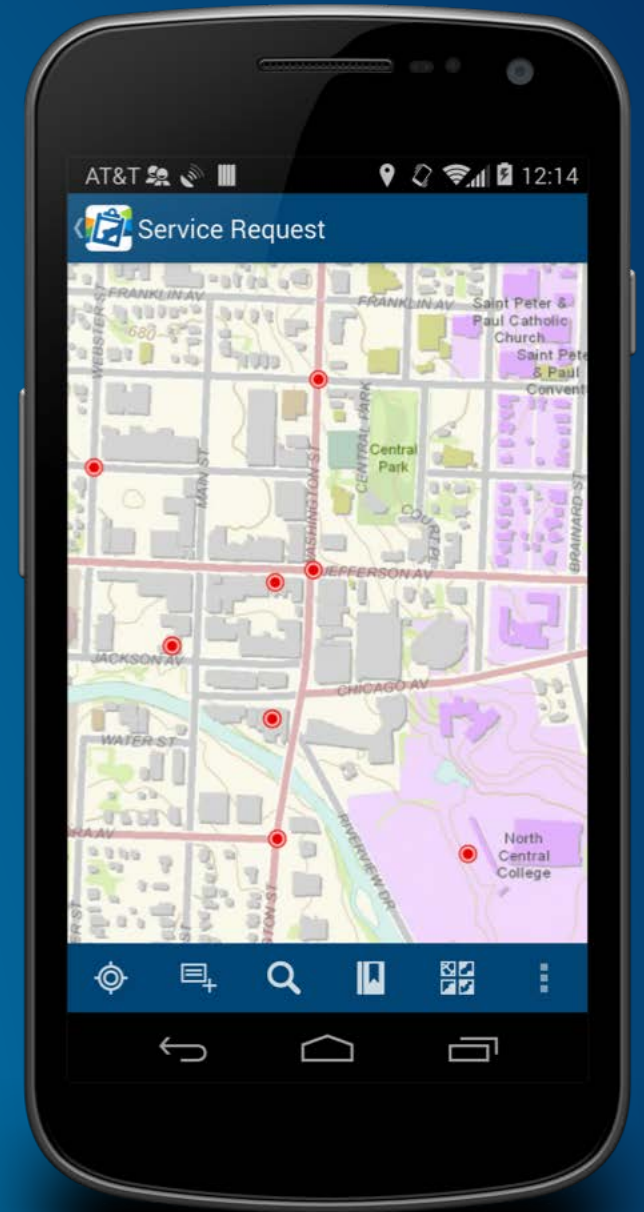
- What sort of location based app are you building?
- What value does location have to the user?
- How will the user interact with location?
- How does the app consume location?
- What are the patterns I can learn from?

<https://developers.arcgis.com/android/guide/determine-your-app-map-pattern.htm>

# Map-centric apps

Its all about the map

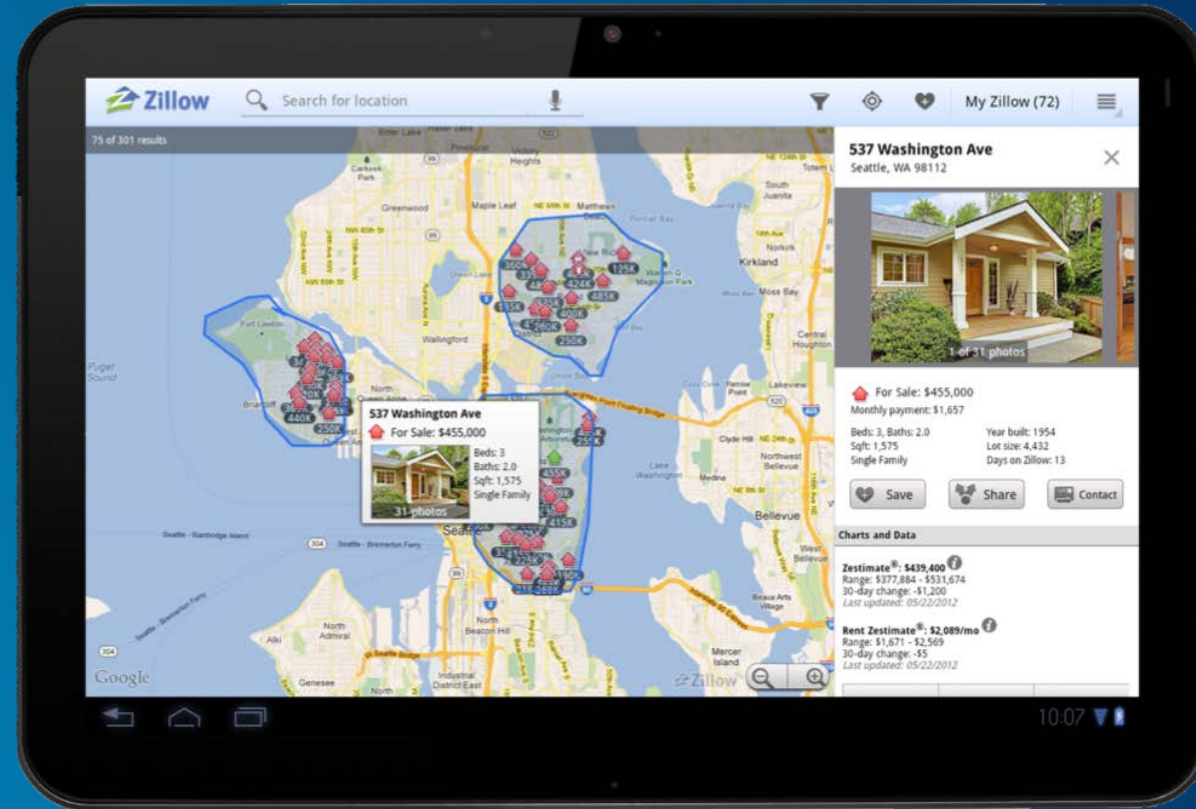
- **Get to the map quickly**
- **Care about cartography (as always)**
  - legend should be last resort
- **Save the state of the map**
  - last used, extent, active tools etc...
- **Full screen map on mobile, side panel on tablet?**
- **Provide tools in appropriate ways**
  - Use gestures, default tap action
- **Offline maps?**
  - a) Take data offline from an online map
  - b) Pre-prepare offline data ready for app startup



# Map-as-navigation apps

The map is an app navigation tool

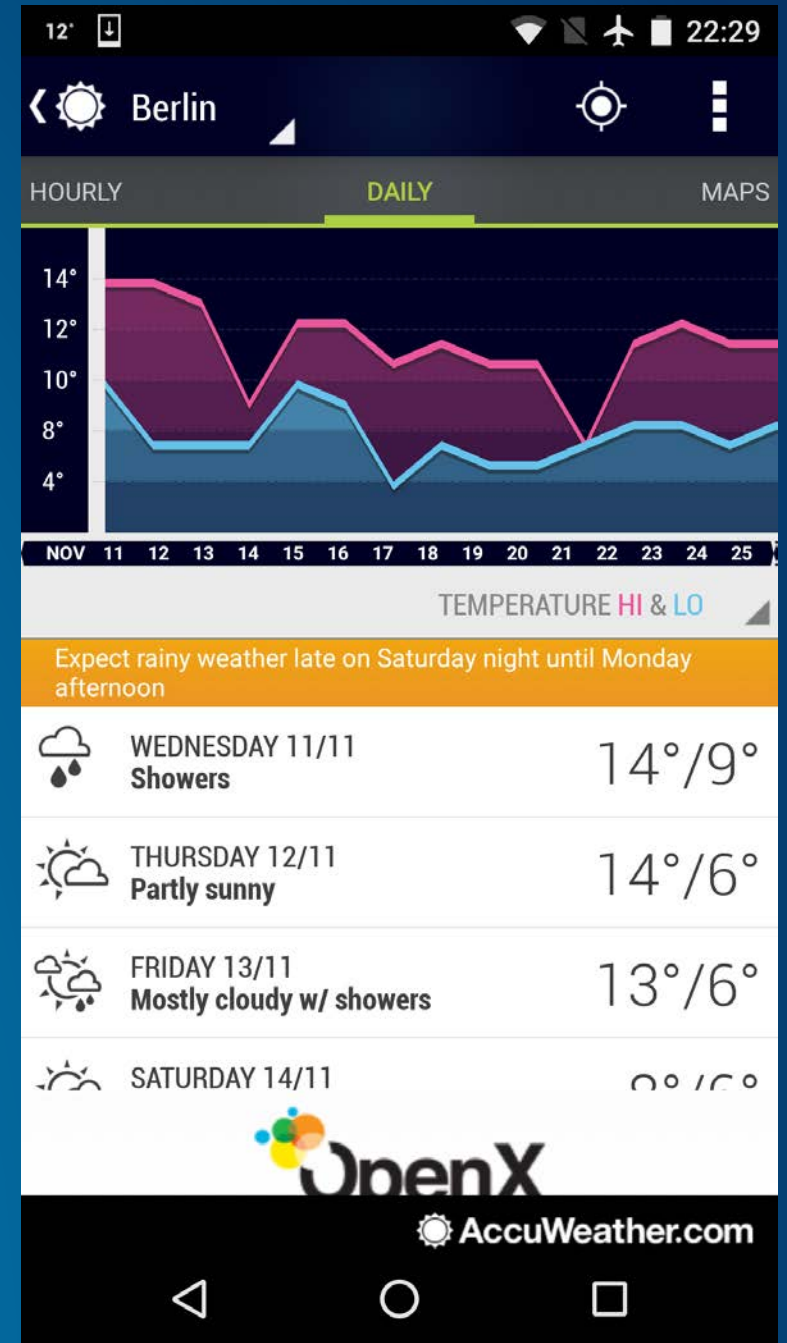
- Map is just part of the apps screen flow
  - has little value on its own
  - e.g. sketch graphic for use in spatial query on a layer
- Focused toolset
  - Simple, no distractions
  - Default actions only?
- Simple cartography
  - Intuitive, no distractions
- Consider the use of popups to show information



# Maps-as-context apps

Map is an auxiliary view that MAY add information

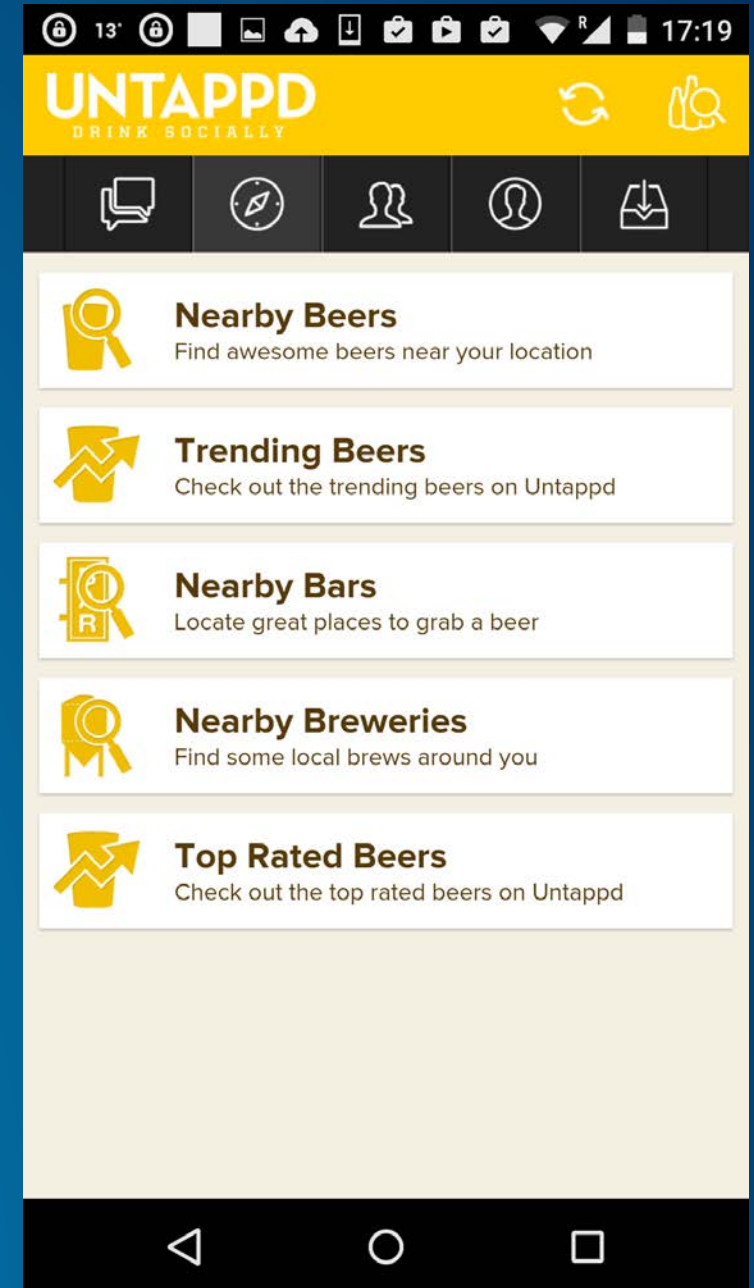
- Map embedded in other information
- Map is driven by other content, e.g. location of weather forecast
- Maybe never shown
- Few or no tools
- Must not distract from the primary content



# Location-as-search apps

Use location to drive a list of items

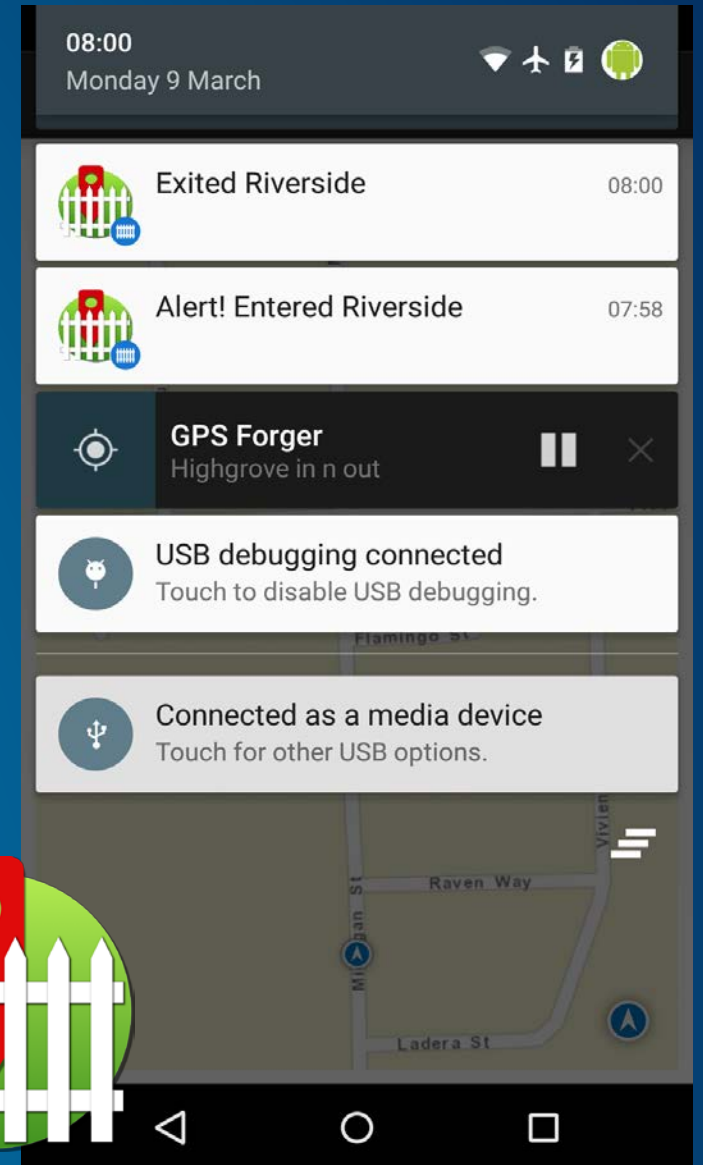
- **Current location provides basis for a search from data/service**
- **Retrieve location from the device location manager**
  - One time request
- **Sorting of the list?**
  - Nearest, largest etc...
- **Add spatial information to the list**
  - Distance, route time, size, pictures etc...



# Location-as-alerts apps

Use location to alert the user about proximity

- Builds on location-as-search apps
- Current location provides basis for search from data/service
- Retrieve location from the device location manager
  - What is your strategy?
  - Be kind on the battery...
  - What is your tolerance?
- Provide alerts - background service, system notifications
- Work offline
  - typically yes - needs to work wherever!



## UI/UX Summary

- Ask lots of questions before you start
- Consider which app pattern(s) apply to your app
- Have a read of our guide doc:

<https://developers.arcgis.com/android/guide/determine-your-app-map-pattern.htm>

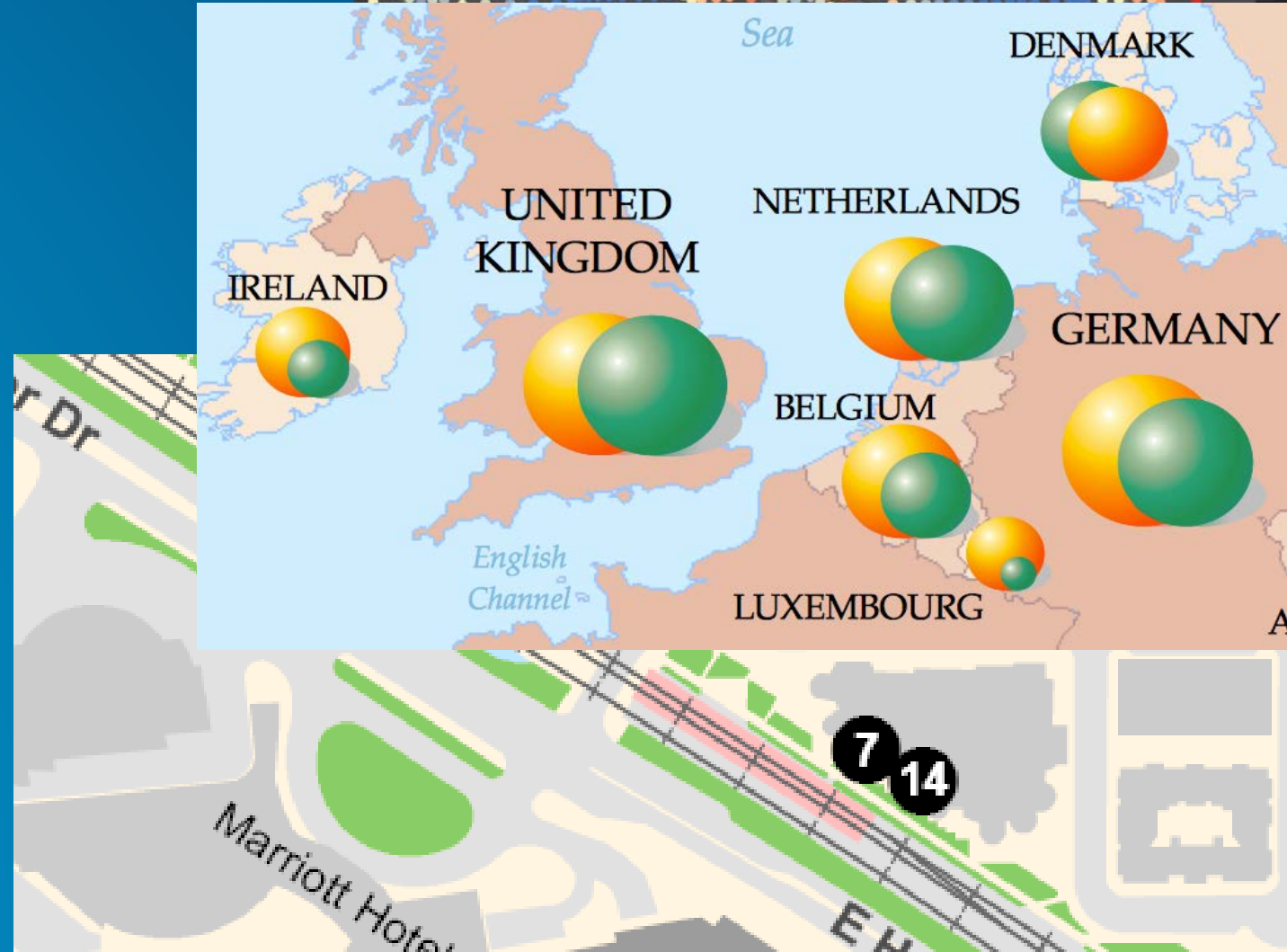
# Data Considerations

Mary Harvey



# Great map apps have data that

- Looks good
- Communicates clearly
- Display quickly
- Responds immediately
- Current
- It's a balance
- What can we do?



# USING ONLINE DATA

- **ALWAYS** access the most up-to-date data
  - Weather, Aircraft location, Emergency Services, Crowd control, Traffic problems
  - Network Connectivity Required
  - Server CPU
  - Bandwidth



- **MINIMIZE** Data Transfer
  - Tiled basemap layers
    - Choice of basemap – minimize tile sizes and LODS
  - Dynamic map services
    - Limit visible layers – minimize server side drawing
  - Feature services
    - Limit data – attribute fields and definition expressions
    - Generalize geometry by query – Maximum Allowable Offset
    - Subject to server limit – default 1000

## ONLINE FEATURE LAYER – FeatureRequestMode [**Java/Android only**]

- GeodatabaseFeatureServiceTable



• **MANUAL\_CACHE** - populateFromService and cached

- Great if small subset of large data is required



• **ON\_INTERACTION\_NO\_CACHE** – every map navigation, select, query

- Very frequently changing data
- BUT Server and bandwidth impact






• **ON\_INTERACTION\_CACHE** – as above cached for duration of session

- Less network use compared to no-cache

## ONLINE FEATURE LAYER – Mode [iOS/OS X/.NET only]


- Feature Layer

-  **SNAPSHOT** – requested all features once when layer is loaded into the map
  - Great if small datasets
-  **ON\_DEMAND** – every map navigation, select, query
  - Very frequently changing data
  - BUT Server and bandwidth impact
-  **SELECTION** – selected features only retrieved
  - Reduced number of features returned

# OFFLINE PATTERNS

- **Deliberately OFFLINE**
  - Prepare Data
    - Provision to device in advance
    - ArcMap CreateRuntime Content – performance steps
    - Reduces server workload
  - App - Planned Sync and Download (API)
- **Occasionally OFFLINE**
  - Normally connected, but network may drop
  - Work with offline data and sync if connected
- **Options?**

## Option 1: Bundle Data

- Android – APK and expansion files
  - iOS – App bundle
  - Windows – App bundle and packages
- 
- 
- Small datasets
  - In advance
  - Store updates controlled by publisher!
  - Internally – good option?

Platform	App size limit
Android (3.2+)	100MB APK (Sept 2015) + 2Gb x 2 expansion files
iOS	4GB iTunes
Windows	Windows 10 – 25GB 8.1 Store – 8GB 8.1 Phone – 4GB

## Option 2: Sideload – Copy/Paste

- **Windows and Android**
- **Manually drag and drop data directly on device**
- **Some android / windows devices have memory expansion slots**
  - **Can simply slot in pre-filled flash memory card**
- **Easy to achieve, flexible**



## Option 2: Sideloading - iTunes

- iOS – iPad, iPhone
- In iTunes: Device -> Apps, File Sharing, drag-and-drop files from computer to the selected app
- Matches platform expectations, user-driven updates



The screenshot shows the iTunes interface for an iPad. The top bar includes navigation icons and the device name 'iPad'. The left sidebar shows the 'Settings' menu with 'Apps' selected. The main area is titled 'File Sharing' and contains two sections: 'Apps' and 'Keynote Documents'. The 'Apps' section lists Keynote, Numbers, and Pages. The 'Keynote Documents' section shows a table with two entries: 'Getting Started.key' (668 KB) and 'Getting Started.pdf' (160 KB). At the bottom right of the 'Keynote Documents' section are 'Add...' and 'Save to...' buttons.

Document Name	Date	Size
Getting Started.key	Today 15:53	668 KB
Getting Started.pdf	Today 15:54	160 KB



## Option 3: ArcGIS Runtime APIs

- Sync Task - `GenerateGeodatabase` -> mobile geodatabase
- `ExportTileCache` Task - `ExportTileCache` -> local tile cache
- Often used for field editing and later syncing – two-way sync is supported
- Requires network for period
- Time taken to sync data
- Same workflow across all platforms
- Controlled by the app
- Not supported for locators, networks



## Quartz – mobile map packages

- ArcGIS Pro can create mobile map packages
- Includes: basemap, operational data, locator, network, **MAP!**
- Quartz Runtime API will consume Mobile Map Packages
- Package can be uploaded to AGOL as an item
- Downloaded using a Task in API
- Can be sideloaded



# Performance

Antti

# Rendering modes

- **Static**
  - Tiled Layer, Dynamic Map Service Layer, Feature Layer, Graphics Layer/Overlay (option)
  - Textures are rendered on GPU
- **Dynamic**
  - Feature Layers and Graphic Layer/Overlays (option), Labels, GPS Layer
  - Symbols are converted to textures
  - Geometries are converted to triangles and texturized at render time

## Why different modes?

- We are building a general GIS solution
- We don't know the data ahead of time
- Must support a variety of use cases and workflows
- Performance is paramount
  - You choose the right mode for your use case and device

**DEMO**

rendering mode

## Static mode

- **Rendering primarily done through CPU**
- **Designed for cartographic quality**
  - Whole graphic is rendered as a path
- **Scales up well**
  
- **Can be system memory intensive**
- **Can drain your battery**
- **Updates to graphic require a redraws of the image**
- **View changes require full or partial updated to image**

# Dynamic Mode

- Entire graphic representation lives on the GPU
- Some graphic changes can be applied directly to the GPU state
  - For example moving or animating
- Number and complexity of graphics can impact GPU resources
- Symbology could look a little different
- Performance is the priority



## So... which mode I should use?

- I love the answer... it depends
- It depends on...
  - amount of data
  - symbology
  - user experience
  - exact cartography presentation needed
  - your device

## So... which mode I should use?

- **Core will consolidate layers when possible for maximum performance**
- **Dynamic mode**
  - Number and order of the layers isn't that important
  - All the resources are shared between all layers on the GPU
- **Static Mode**
  - Each layer renders to an image
  - Layer ordering can affect resource sharing

# Static Mode layer consolidation

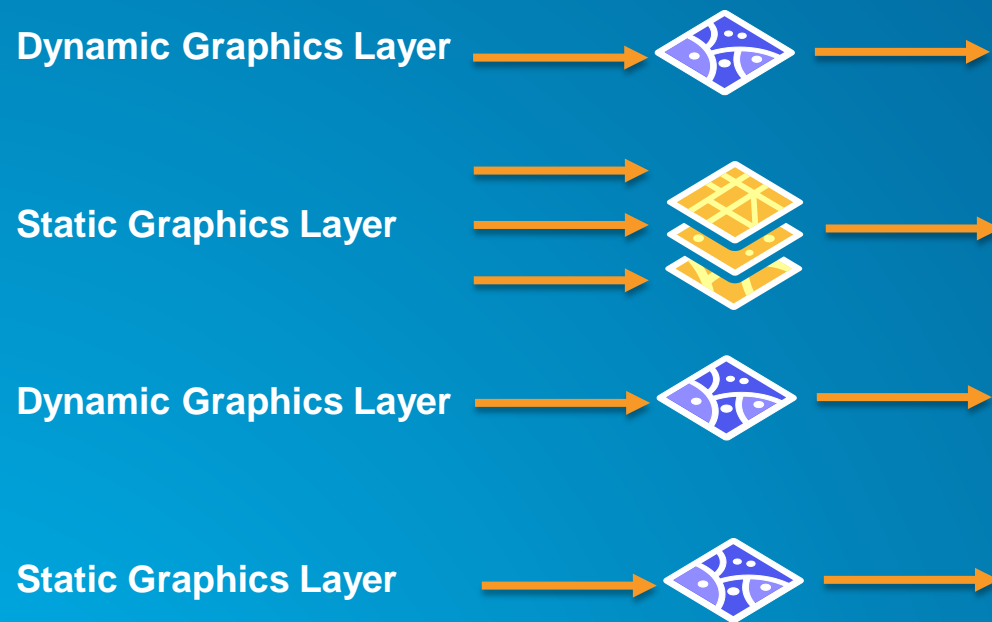


Image	Texture
	Set of textures needed
An image	Broken into textures
	Shared with all dynamic layers
An image	Broken into textures

# Tips how to improve performance

- **Use AddRange to add graphics**
- **Go asynchronous**
  - Keep the ui responsive
  - Quartz is build heavily around this approach
- **Background threading**
  - Push work off from the UI thread to keep UI thread work limited
- **Parallel execution**
  - Don't let "convenient coding" take over performance
  - Examine when going parallel is good and when not

```
for (int n = 0; n < numGraphics; ++n)
{
    graphicsLayer.Graphics.Add(CreateRandomGraphic());
}
```

```
_graphics = new List<Graphic>();
for (int n = 0; n < numGraphics; ++n)
{
    _graphics.Add(CreateRandomGraphic());
}
graphicsLayer.Graphics.AddRange(_graphics);
```

# Demo

Threading

# Symbology considerations

- **Renderers versus Symbols**
  - Using renderers gives better resource reuse
  - Simple symbols over complex ones
- **Scale ranges**
  - What we do need to see?
  - Same data – several presentations
- **Do you know where your symbols comes from?**

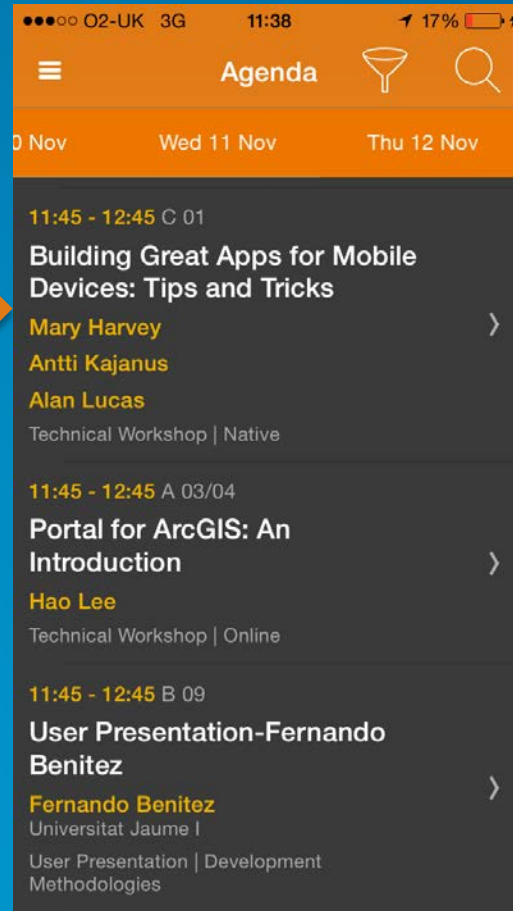


# Please Take Our Survey!

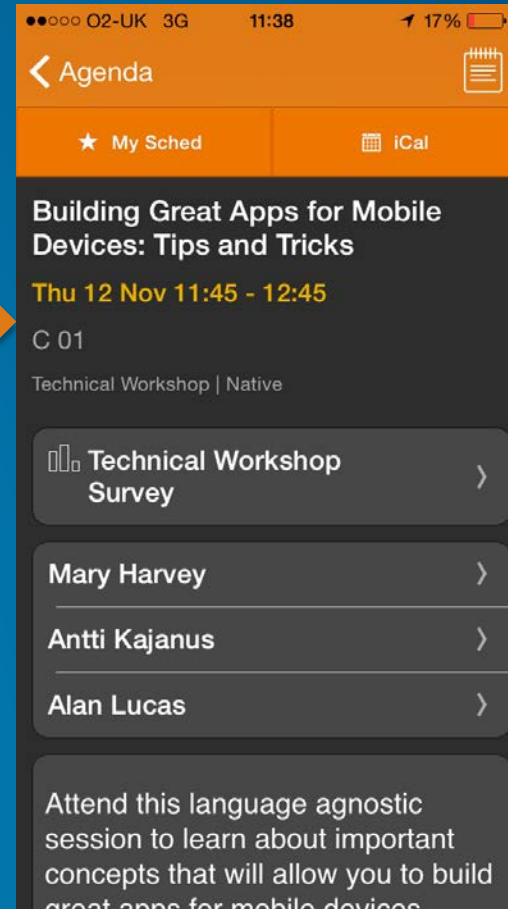
Download the Esri Events app  
and find your event



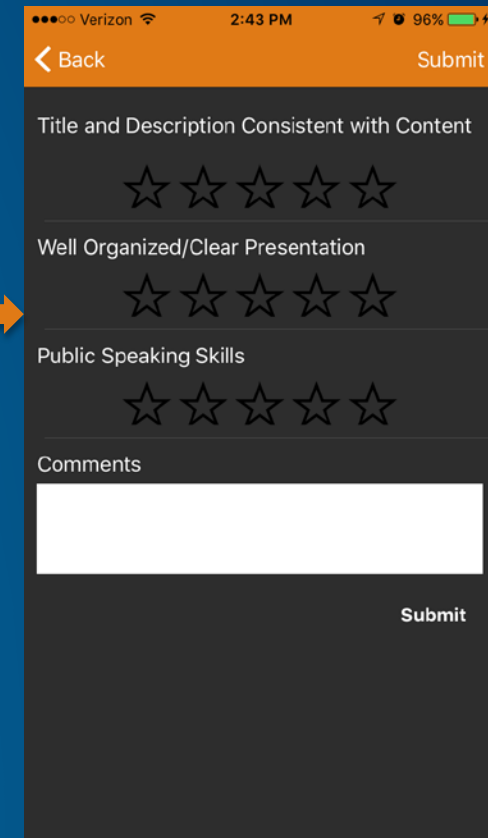
Select the session



Select  
"Technical Workshop Survey"



Complete Answers  
and Select "Submit"



**QUESTIONS**



Thank You to Our  
Gold Sponsor

con•terra



esri®

BERLIN

Prenzlauer Berg

Friedrichshagen

Friedrichshagen

Friedrichshagen

