

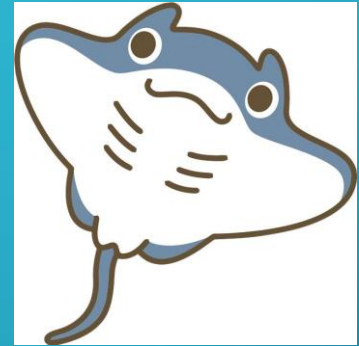
Roads Asset Editor

Tom Ryan
Fred Lott
Harkeerat Kang

WAGISA Conference
6.15.2023



What is Rae-Rae?



- Custom application for Roads Division
- Manage Roads asset inventory in unincorporated King County
- System of record
- Tracks 33 different type of Roads assets
 - ADA ramps, Barricades, Signs, Traffic cameras
- Add new assets or updating existing assets

Modernization Effort



- Replace Silverlight project
- Silverlight is deprecated by Microsoft
- Upgrade and rewrite application
- Improve application performance
- Responsive design (desktop/tablet/mobile)
- Improve functional operations

Technologies Used

- App built using React and Next.js
- ArcGIS Maps SDK for JavaScript 4.x
- Uses Azure stack
- Enterprise Geodatabase 10.9.1



NEXT.js



Why custom application?

- Using Linear Referencing
 - Route feature class with a defined measurement (m value)
 - Event tables contains info about assets, conditions and events located along route features
- Dynamic segmentation
 - Uses a process to display assets from an event table on a map using route features measure
 - The routes are no longer needed to be split each time an attribute value changes.
 - You dynamically locate the asset on the segment
- No SaaS or COTS applications available

Agile Methodology

- Two week sprints followed by demos
- Scrum Master and Project Manager from business
- Product owner from Roads
- Technical lead
- Project development team
 - Yuko Caras, Matt Kozleski, Fred Lott, Tom Ryan, Harkeerat Kang
 - Lisa Owen, Jeff Gregg
 - Adam Cabrera
 - Michelle VanCooten, James Bach

Personas

- Tina the Traffic Engineer



Roads Asset Supervisors
Roads Data Stewards
Roads Work Planning

- Randy the Roads Asset Installer



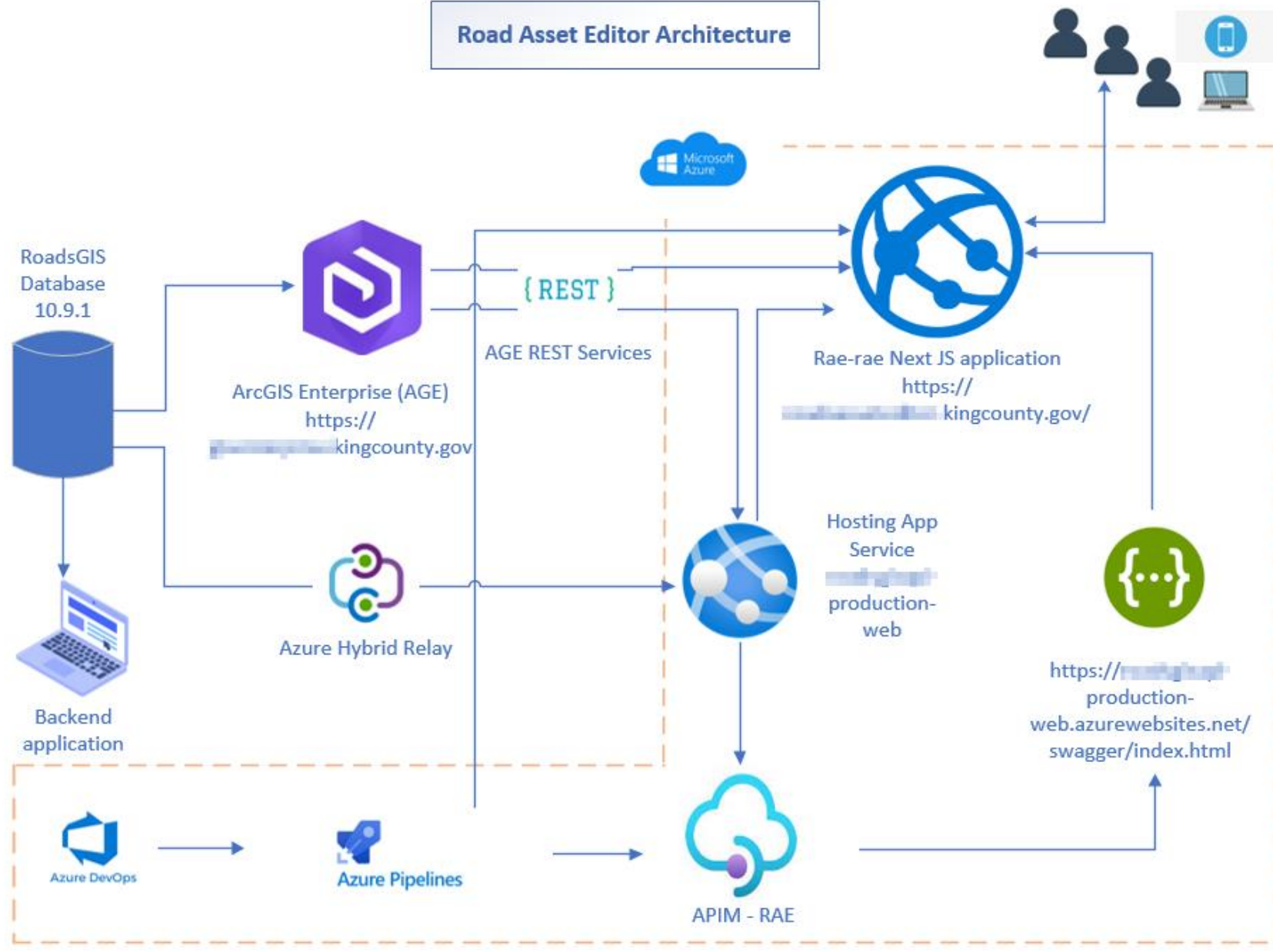
Sign Installers
Signal Technician
Pavement Striper
Sidewalk & Curb Installer
Landscape Architect

- Mary the Maintenance Specialist



General maintenance

Road Asset Editor Architecture

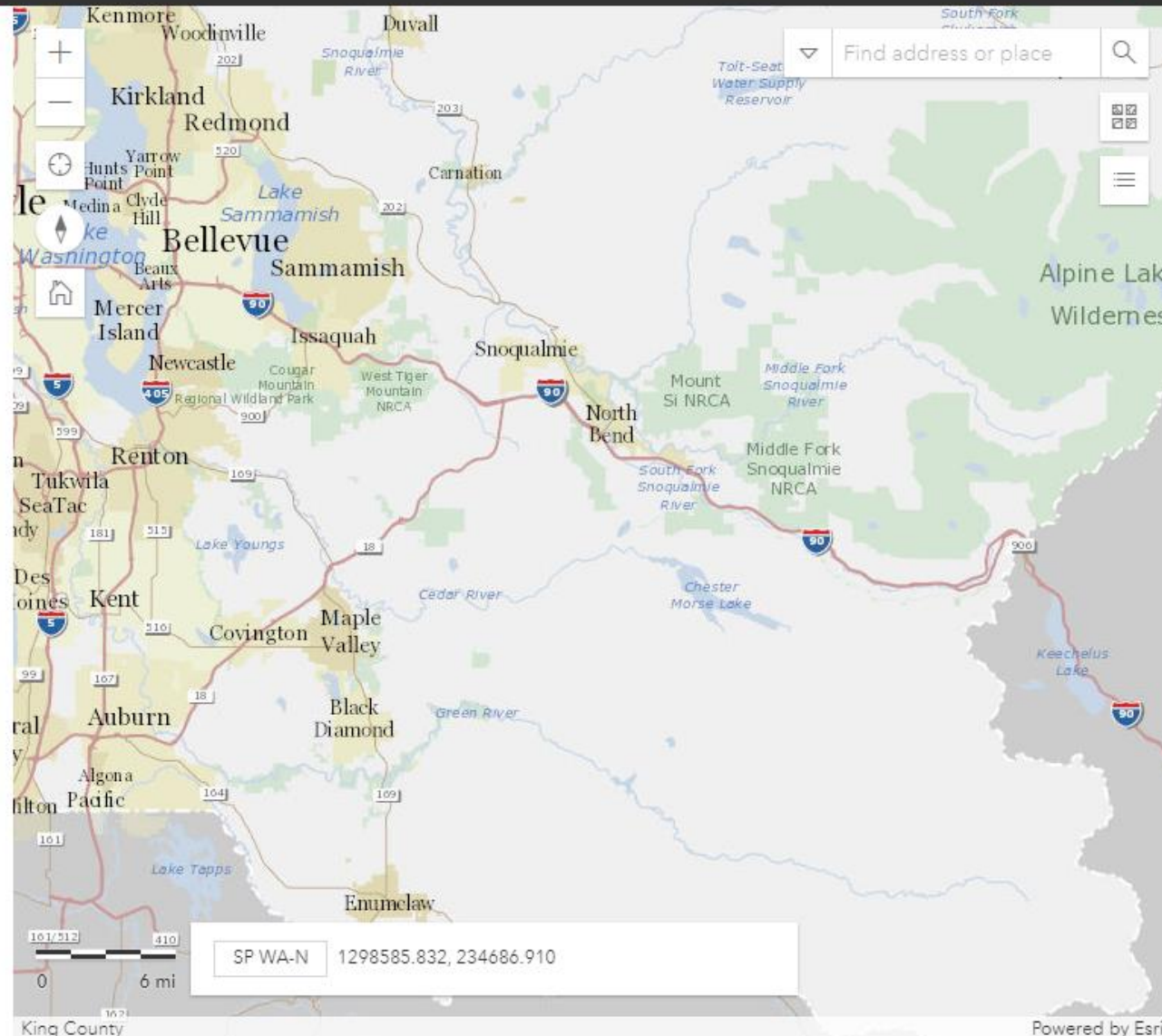


Interface tour

Select Asset Type

Select the asset type you want to add or edit.

Asset Type

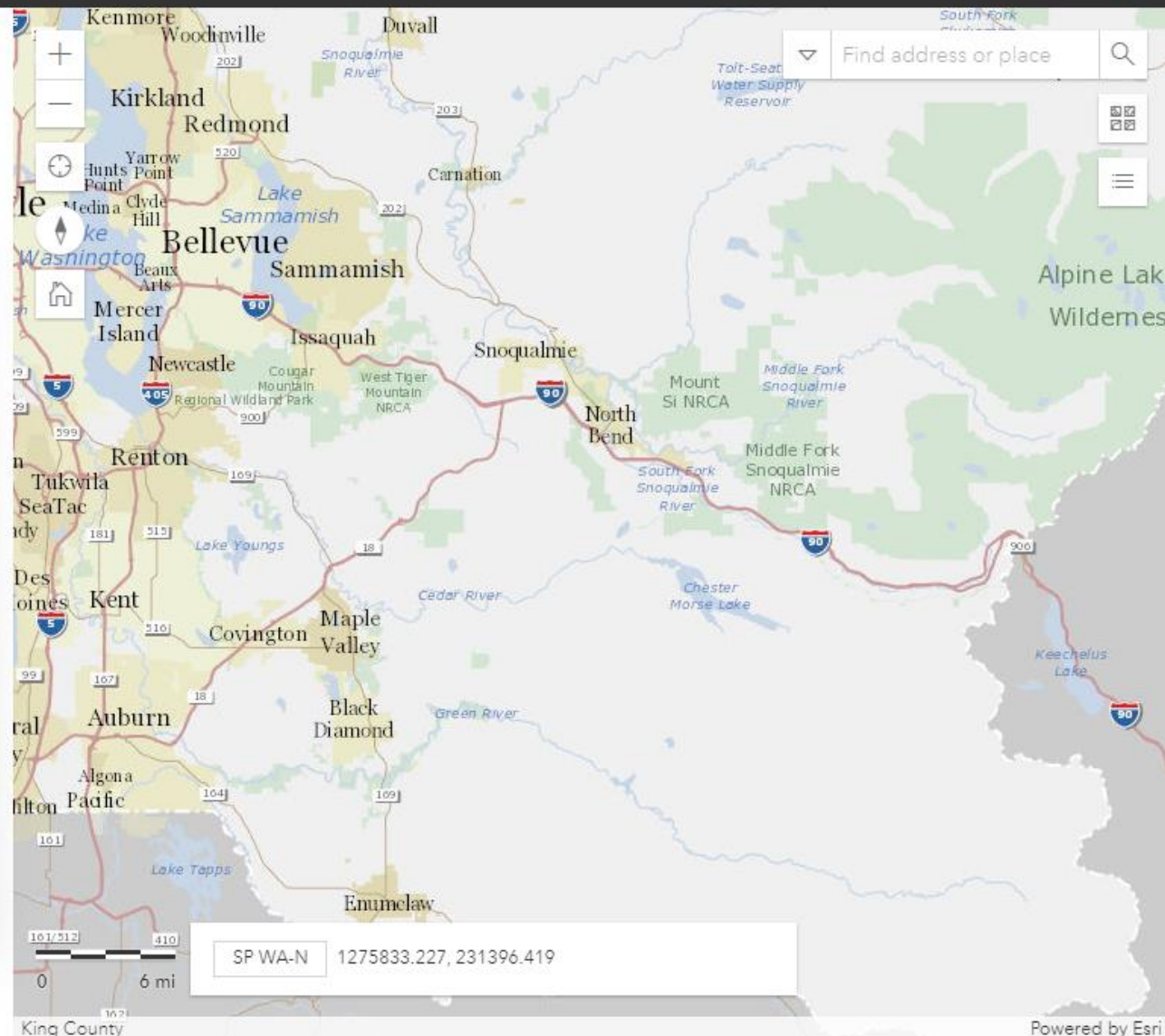


Select Asset Type

Select the asset type you want to add or edit.

Asset Type

- ADA Ramp
- Barricade
- CM Sign
- Curb
- Delineator
- Fence
- Flasher
- Gutter
- Headwall
- Jersey Barrier
- Luminaire
- Mowable Slope
- Pavement Marking
- Planter Box
- Planter Strip
- Radar Speed Sign
- RD Facility
- Retaining Wall
- Roads CIP



Edit Asset

[Help](#) ?

Add or edit asset information.

Asset Type
Barricade

 ADD ASSET



Edit Asset

[Help ?](#)

Add or edit asset information.

Asset Type
Barricade

CREATE MUTCD

+ SELECT LOCATION

Barricade W16-92R Details

SAVE



RoutelD
63340Measure
185Offset
30

MUTCD_Code
W16-92R

Owner
Roads

CurrentStatus
Active

InstallDate
10/16/2007 12:00 AM

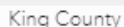
InventoryDate
mm/dd/yyyy --:-- --



[Help](#)

Asset Type
Barricade

 SELECT LOCATION

 **SAVE**

Edit Asset

[Help ?](#)

Add or edit asset information.

Asset Type
Curb



SELECT LOCATION

Curb New Details



SAVE

+

-

📍

🏠

📏

📐

📊

Find address or place

🔍

🗺️

☰

SE 187th St

57927

933

320

366

35384

42333

100 ft

SP WA-N

1313634.964, 161437.047

Choose action

⌵

✕

There are the following options for adding a linear asset.
Click ... if not all options are visible.

- 📏 adds the asset for the entire segment.
- 📐 adds the asset from the start of the segment (0) to the selected measure (52)
- 📊 adds the asset from the selected measure (52) to the end of the segment (933).

Edit Asset

Add or edit asset information.

Asset Type
Flasher

ADD ASSET

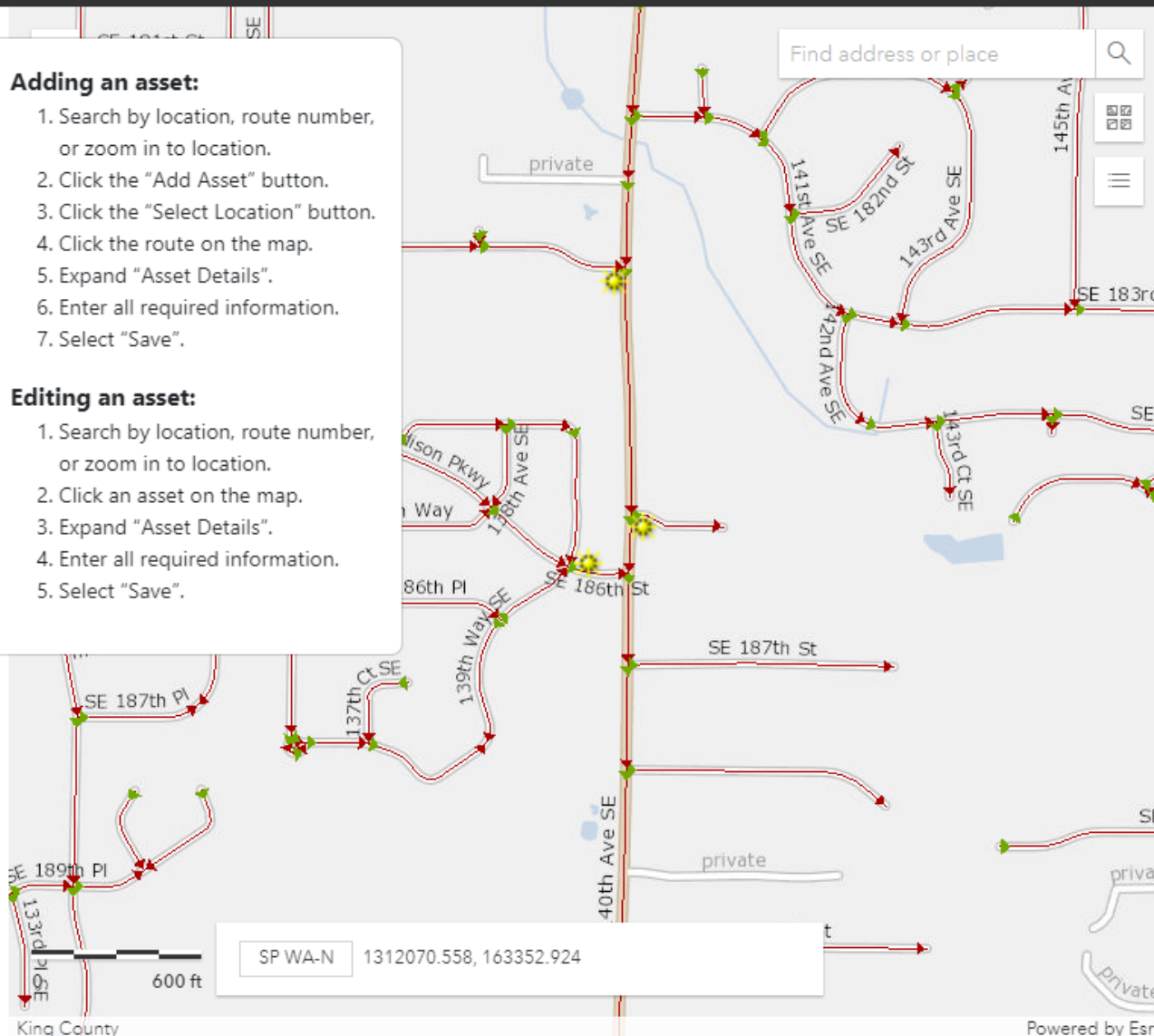
[Help](#) ?

Adding an asset:

1. Search by location, route number, or zoom in to location.
2. Click the "Add Asset" button.
3. Click the "Select Location" button.
4. Click the route on the map.
5. Expand "Asset Details".
6. Enter all required information.
7. Select "Save".

Editing an asset:

1. Search by location, route number, or zoom in to location.
2. Click an asset on the map.
3. Expand "Asset Details".
4. Enter all required information.
5. Select "Save".



InventoryDate
mm/dd/yyyy --:-- --

ActiveDate
11/18/2005 12:00 AM

RetireDate
mm/dd/yyyy --:-- --


AssetID
1828

Jurisdiction
King County

ModifiedBy
Automated

DBCreateDate
04/17/2012 08:00

DBModDate
04/17/2012 08:09:24 AM

 SAVE

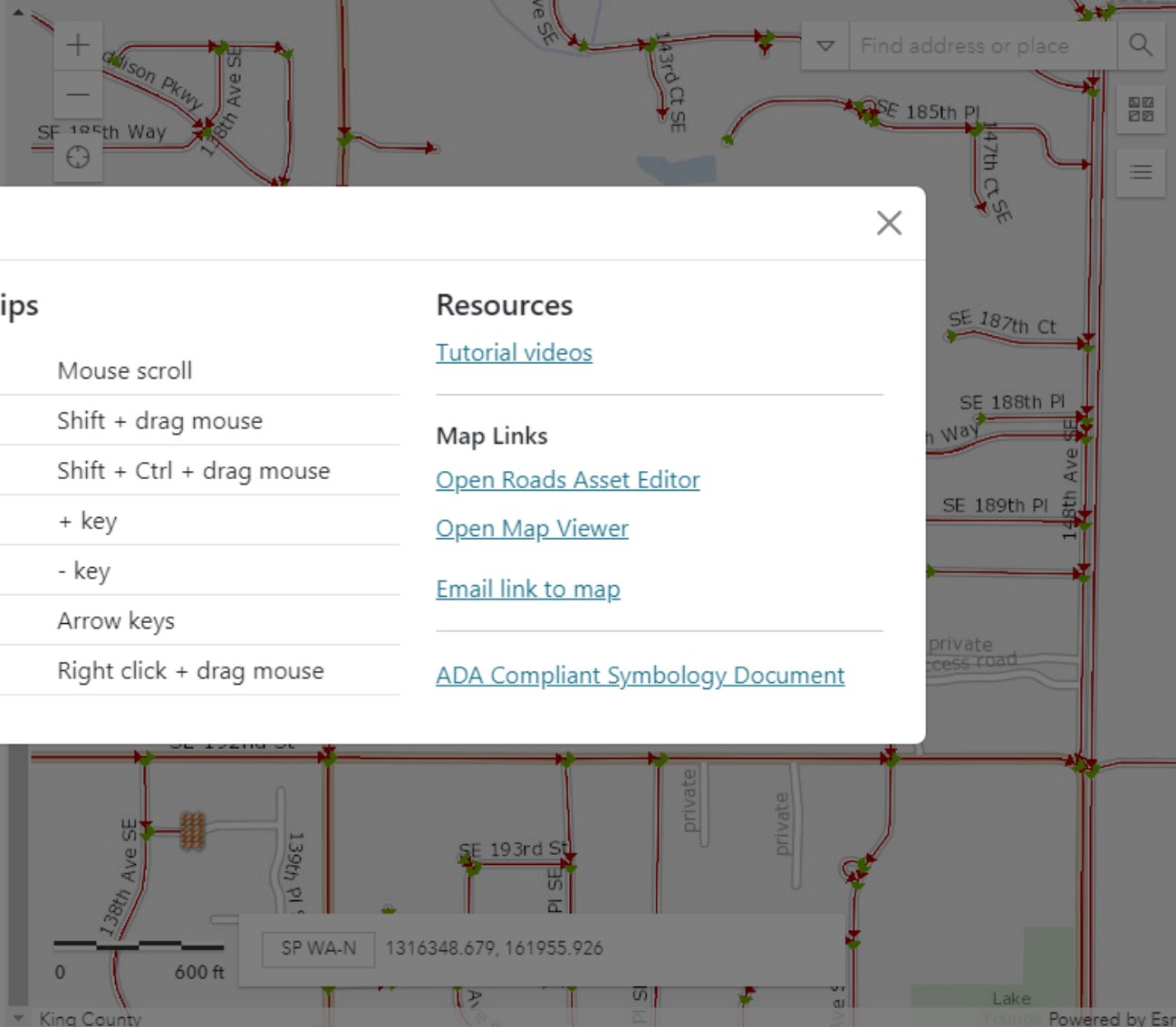
Map Navigation tips

Zoom in / out:	Mouse scroll
Zoom in:	Shift + drag mouse
Zoom out:	Shift + Ctrl + drag mouse
Zoom in one level:	+ key
Zoom out one level:	- key
Pan:	Arrow keys
Rotate:	Right click + drag mouse

Resources

[Tutorial videos](#)

Map Links

[Open Roads Asset Editor](#)
[Open Map Viewer](#)
[Email link to map](#)
[ADA Compliant Symbology Document](#)


Tutorials

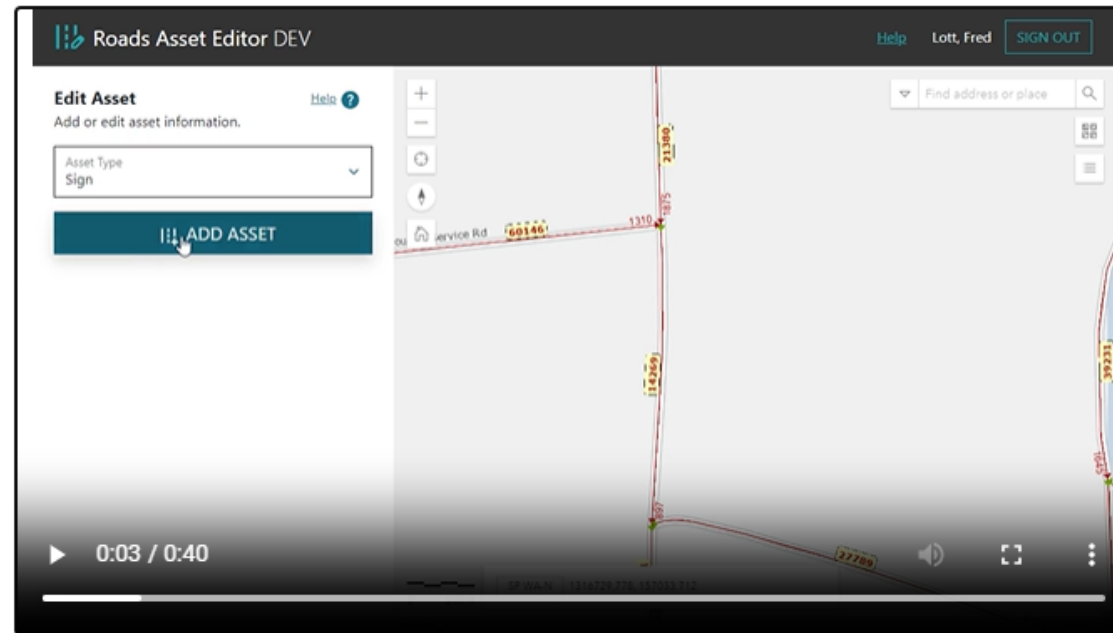
General

- [Signing in](#)
- [Signing out](#)
- [Map navigation](#)
- [Search](#)
- [Basemaps](#)
- [Map legend](#)
- [Finding your location](#)
- [Rotating the map](#)
- [Resetting the map view](#)
- [Selecting an asset type](#)

Editing

- [Creating a new point asset](#)
- [Creating a new line asset](#)
- [Creating a new complex asset](#)
- [Updating an existing point asset](#)
- [Updating an existing line asset](#)
- [Updating an existing complex asset](#)
- [Adding an asset to an existing mount structure](#)
- [Updating location using the map](#)
- [Switching an existing asset to a pole mount structure](#)

Creating a new complex asset



Implementation details

Why Azure API Management (APIM)?

- Meet KCIT security and compliance requirements
- Put all the backend architecture and complexity of ArcGIS Enterprise and Enterprise Geodatabase on-premise access behind the API
- Securely expose API operations for our front-end application
- Simplify and reduce the cost of application integration
- Make data and services re-usable and easily accessible

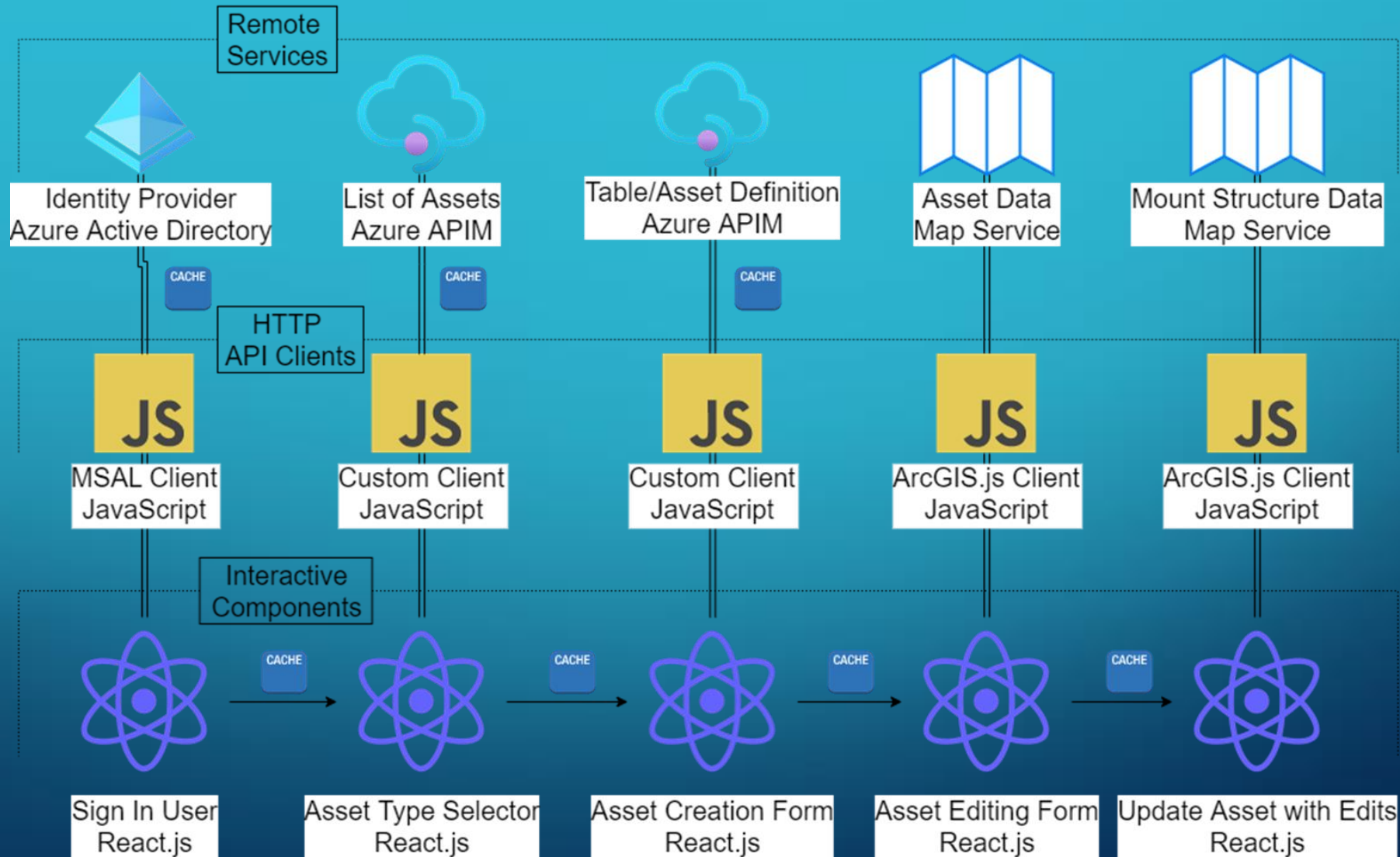
API Endpoints

GET	/isuserauthorized	Verify that a user is authorized.
GET	/applicationsettings	Get application settings
GET	/editableassets	Get editable assets by user
GET	/layerfields	Get layer fields metadata
GET	/MUTCDs	Get MUTCD codes by asset
GET	/roadwaybyroutexy	Get roadway by route and XY
GET	/roadwaybyrouteid	Get roadway by route ID

Client-side Interactivity using React.js

- ArcGIS.js is a traditionally structured Object-Oriented library.
- React.js enforces a functional programming paradigm.
- To resolve this conceptual mismatch, we isolated ArcGIS.js.
- Data is provided to the map component using `useState()` hooks and `useEffect()` event handlers.
- The map returns data to the other React components by calling the `set()` function exposed by each `useState()` hook when events (like a click) occur.

HTTP Requests and Response Consumers



Closing Thoughts

Lessons learned

- Thorough testing before launching any new features or updates
- Use UAT scripts to test all functionality in test environment
- Don't push any updates before a big event
- Keep documentation current
- Make use of UX designer in the beginning

Questions ?

Tom Ryan tryan@kingcounty.gov

Fred Lott fred.lott@kingcounty.gov

Harkeerat Kang harkeerat.kang@kingcounty.gov

King County GIS Center

<https://kingcounty.gov/gis/>

Extra slides