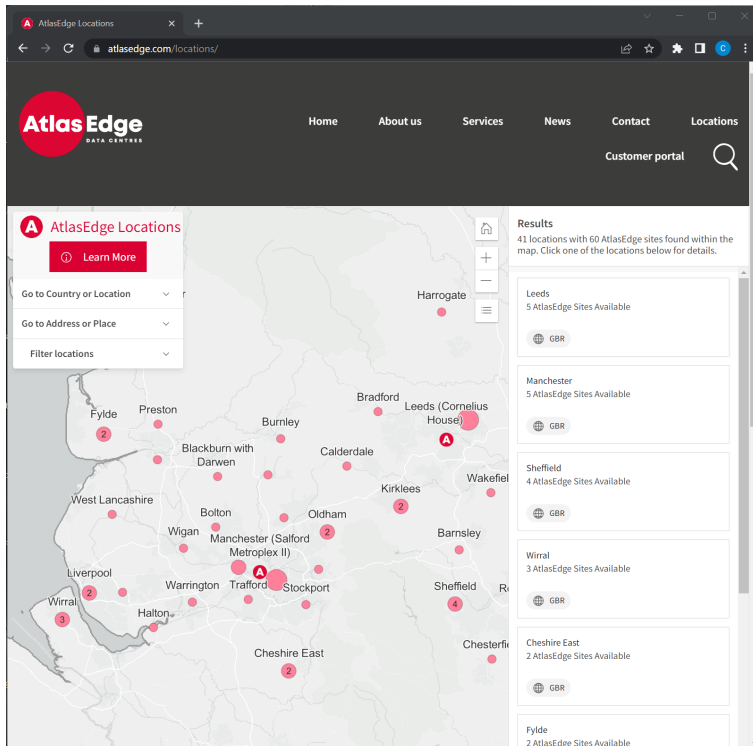
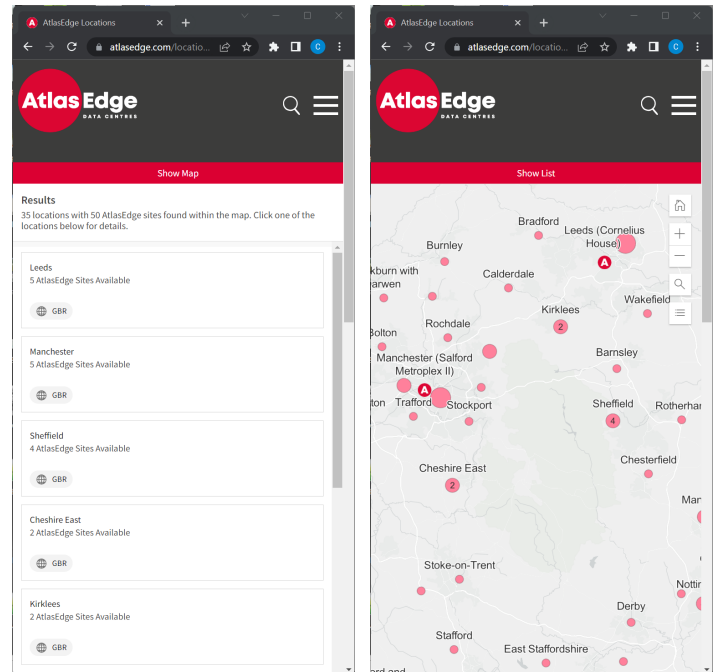


Sales-Support Web Application		Live Demo:	www.eichcorp.com
Purpose:	Implementation:	Source:	Available with purchase
This map-centric, web application enables customers and prospective customers to identify potential data center locations.		This application was built using the React JavaScript framework and Esri's Calcite web components. The mapping front-end uses the ArcGIS API for JavaScript as ES modules (@arcgis/core). Code is organized into reusable React Components. See our Web R Mapping solution for more information.	



Map-centric Interactive Web Application



Responsive Mobile Layout

Fiber-Optic Network Management System

Live Demo:

[Request Here](#)

Purpose:

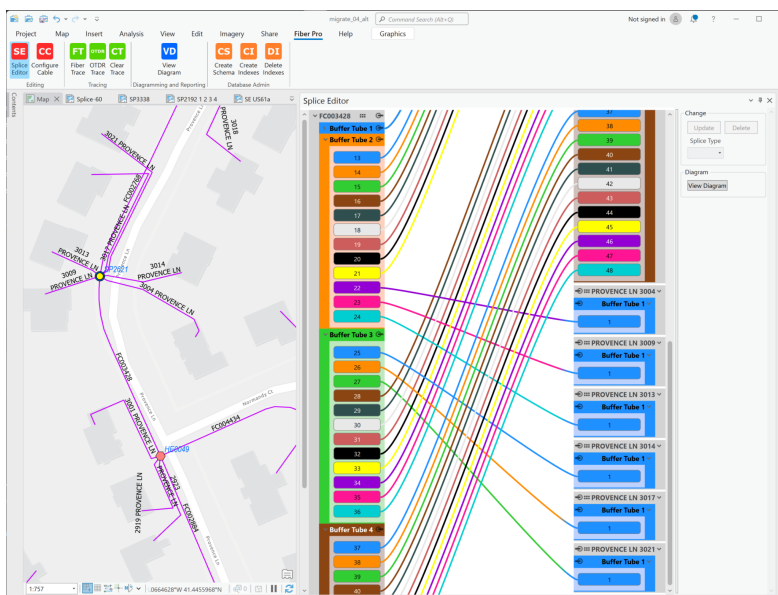
mPower FiberPro enables design engineers to manage fiber object networks using an intuitive, and hyper-usable tool that extends ArcGIS Pro, Esri's flagship GIS desktop application. Engineers use FiberPro to view and update fiber-optic splicing information, design and manage fiber networks, and generate information-rich splice diagrams.

Implementation:

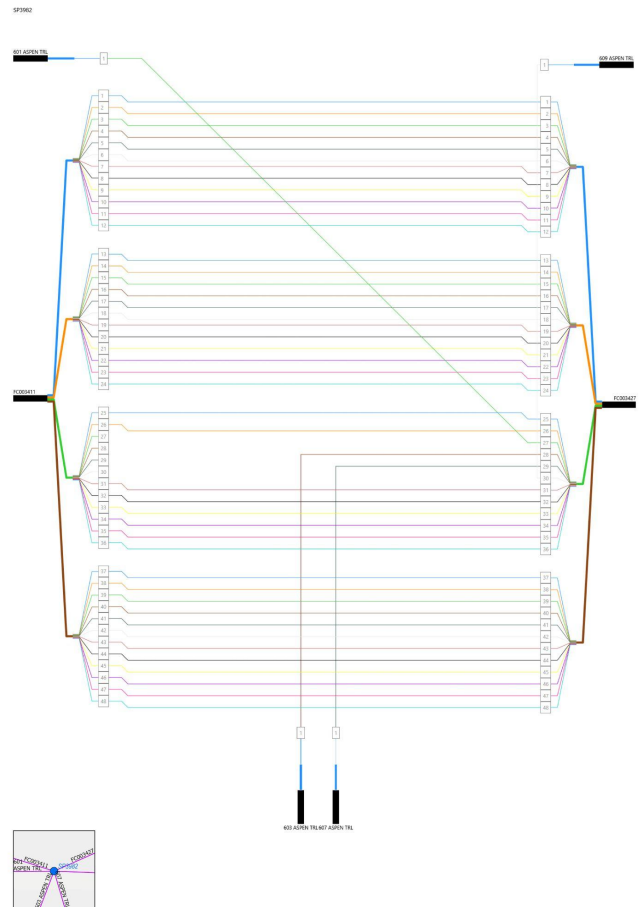
Eichcorp is mPower's [Software Development Partner](#) and handled all aspects of the development of FiberPro, from design through implementation. FiberPro is an ArcGIS Pro Add-In built using Microsoft .NET/C# and utilizes data stored in file and enterprise Esri GIS geodatabases (e.g SQL Server). Eichcorp also developed workflows and automated Python scripts to migrate data from other systems to FiberPro.

Source:

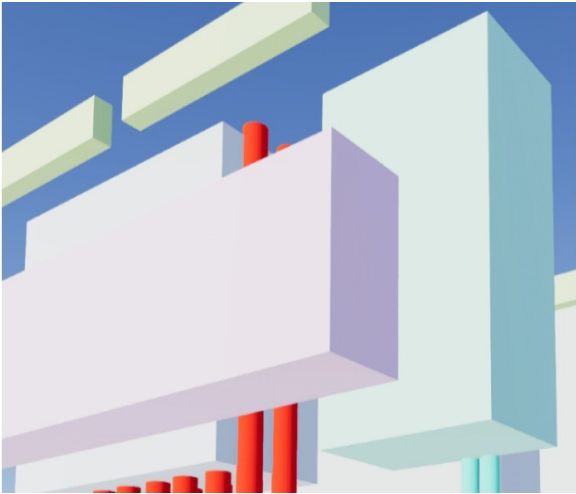
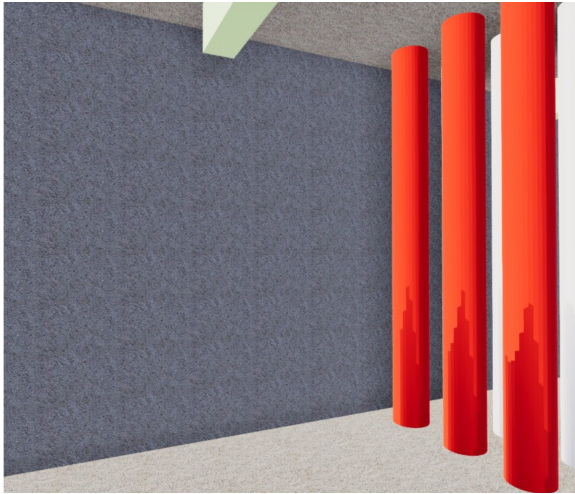
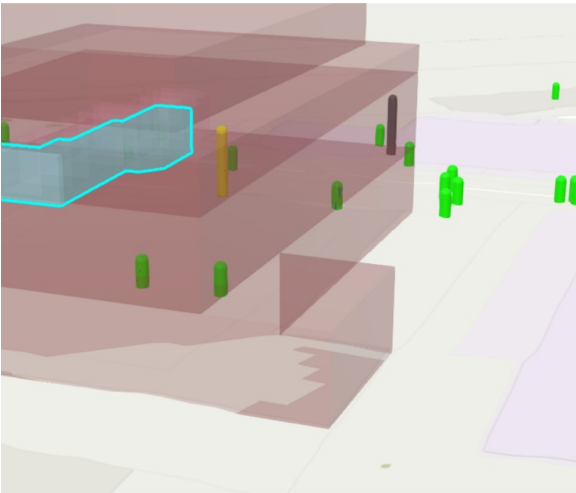
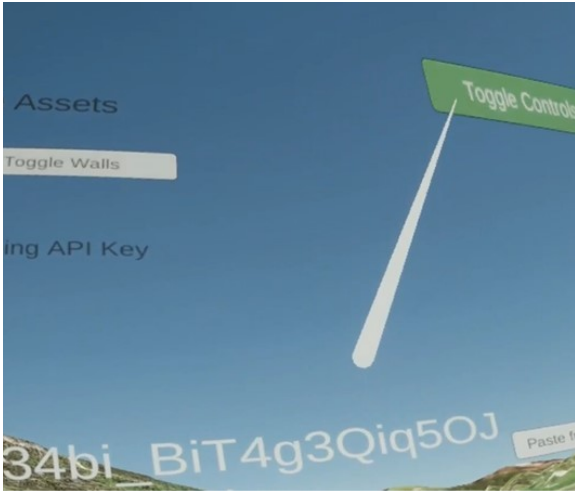
Not available



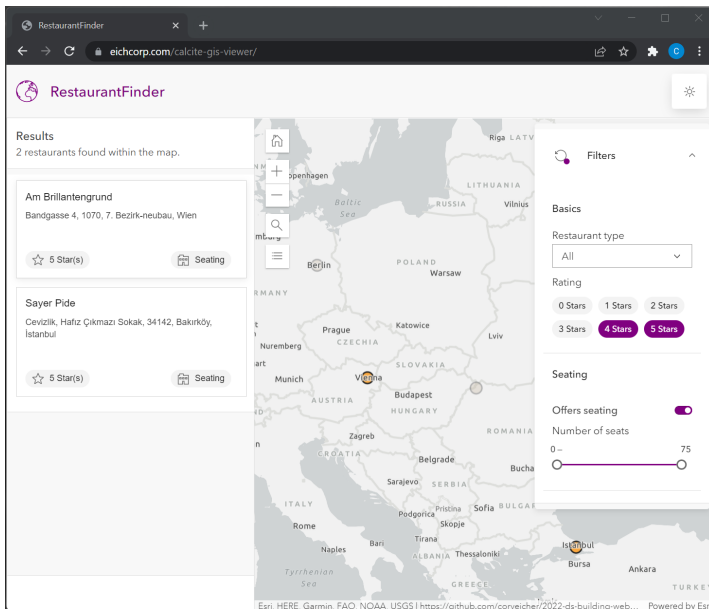
Splice Editing in FiberPro



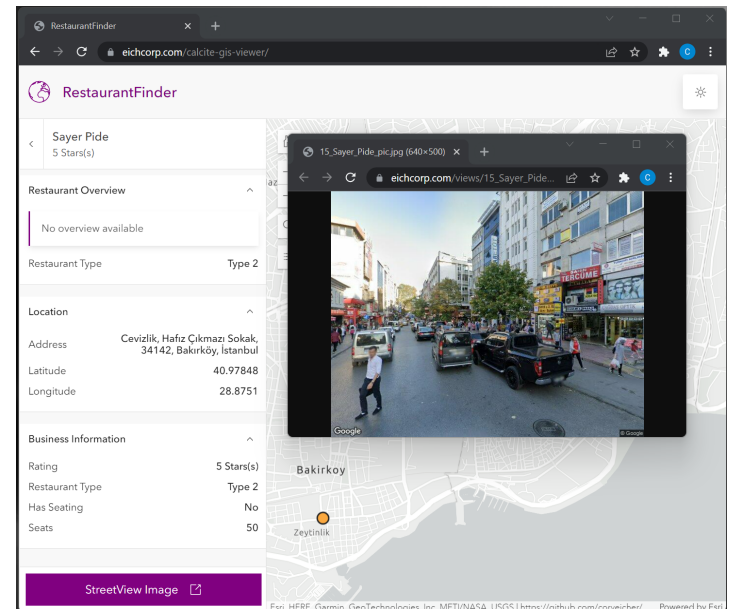
FiberPro Splice Diagramming

Unity-ArcGIS VR Digital Twin Application		Live Demo:	Not available
Purpose:	Implementation:	Source:	Not available
This virtual reality application allows end-users to visualize and navigate commercial property digital twins using Oculus VR goggles. The immersive VR experience offers a unique perspective of both interior and exterior physical assets, and the implementation supports future use cases where data are frequently updating		This Oculus Android native application was developed using the Unity gaming engine and the Esri ArcGIS Maps SDK for Unity. The app leverages 3D GIS data provided via ArcGIS Online web services. 3D GIS data and services are additionally reusable in other VR/non-VR applications. For additional details visit linkedin.com/company/eichcorp	
 Cutaway View of Interior Assets		 Interior View	
 Exterior Assets with Highlighted Room		 Example VR User Interface	

React/Calcite Web GIS Viewer		Live Demo:	eichcorp.com/business-viewer
Purpose:	Implementation:	Source:	react-calcite-viewer
This application allows end users to visualize commercial property locations in a web browser in both map and list form. Users can: perform faceted-searches to narrow down the map/list, drill in to data points to see more detail, and click links to display Google Street View images.		This application was built using the React JavaScript framework and Esri's Calcite web components. The mapping front-end uses the ArcGIS API for JavaScript as ES modules (@arcgis/core). The application is a port of an existing calcite application to React. The project implementation differs from this sample: Code is organized into React Components, the application is more configurable to different data, and additional reusable forms/mapping functionality have been added. Learn More	



Visualizing faceted-search results



Drilling in to view details and link to Street View image

Python Google Street View Image Harvester		Live Demo:	Not available
Purpose:	Implementation:	Source:	Available by request
This solution automatically captures Google Street View photos for a list of commercial properties. The solution additionally allows users to override any individual photo to use an improved view of the property (based on camera position). These StreetView images are utilized in the above case study Web GIS Viewer.	This solution was implemented as a Python script which is configured by a .csv file. By default, photos are captured via the Street View Static API using each row's Latitude and Longitude values. Then, to allow capturing images which put the best "face" on these locations, the camera's "viewport" parameters (position, angle, zoom, etc.) can be overridden in the .csv. Viewport overrides function in a way which allows a human to use Street View to refine/choose the camera position, then copy the viewport parameters directly from the browser url and paste into the spreadsheet. This allows exact camera positioning and consistent re-capturing in the future.		



Example Image 1: Before and after



Example Image 2: Before and after