17th International Conference on Advances in Databases, Knowledge, and Data Applications

DBKDA 2025

March 9, 2025 to March 13, 2025 - Lisbon, Portugal

A Low-Code Approach for Creating Dynamic Map-Based Web Applications Using W3C Web Components

Andreas Schmidt[1,3] and Tobias Münch[2, 4]

(1)
Karlsruhe Institute of Technology (KIT)
Germany

(2)
Münch Ges. für IT Solutions mbH,
Gewerbering 1, 49393 Lohne, Germany

(3)
Karlsruhe University of Applied Sciences
Germany

(4)
Chemnitz University of Technology
Germany

Outline

- Introduction
  - Leaflet Library
  - W3C Web-Components
- Our Concept
  - Architecture
  - Implemented Components
  - Examples
- Summary & Outlook

Introduction - Leaflet library

- Javascript library to build web-mapping applications
- Open Source
- Features
  - Allows to display tiled web maps and overlays (WMTS)
  - Supports Markers, Polylines, Polygons, Circles, Rectangles, Popups
  - Supports GeoJSON format
  - Allows various interactions with maps
  - lightweight (< 50KB)
  - huge community (many extensions)
  - very good documentation
  - supports many desktop/mobile browsers
- Widespread usage (i.e. Flickr, pinterest, github, FourSquare)

Introduction - Web-Components

- Composite W3C standard
- Create your own HTML-Elements with arbitrary semantics
- Run in the browser
- Implementation in ES2015 Javascript class (inherit from HTMLElement)
  - Implementation of callback methods
- Parameter can be passed in native way as attributes
- Attribute values are monitored for changes
- Shadow DOM to encapsulate Web-Component
- Templates (inactive HTML fragments that can be repeatedly used within a component)
- Slots (placeholder)

Introduction - Web-Components

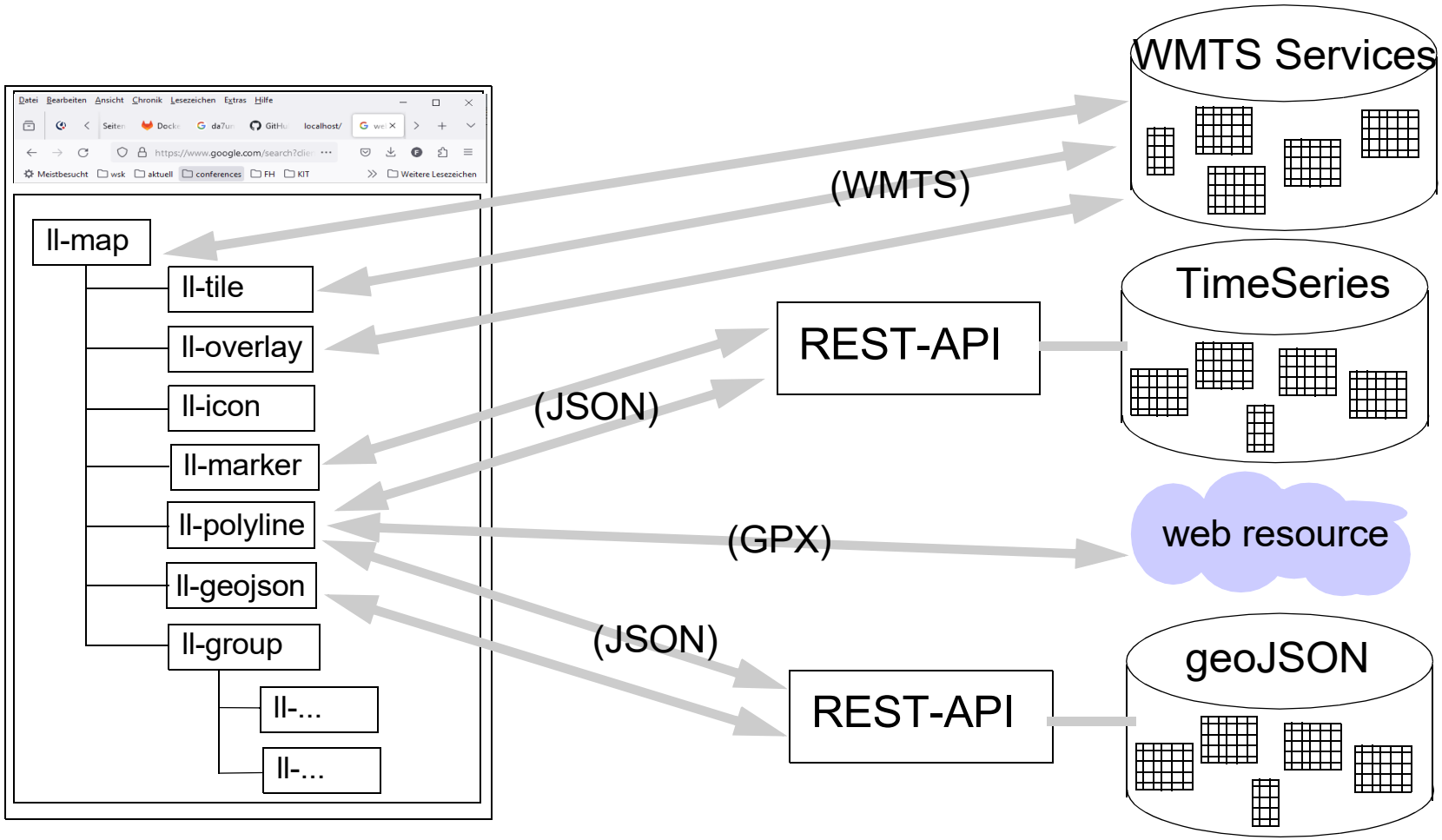
- Composite W3C standard
- Create your own HTML-Elements with arbitrary semantics
- Run in the browser
- Implementation in ES2015 Javascript class (inherit from HTMLElement)
  - Implementation of callback methods
- Parameter can be passed in native way
- Shadow DOM to encapsulate Web-Component
- Templates (inactive HTML fragments that can be repeatedly used within a component)
- Slots (placeholder)

support declarative web-mapping development

Our Concept

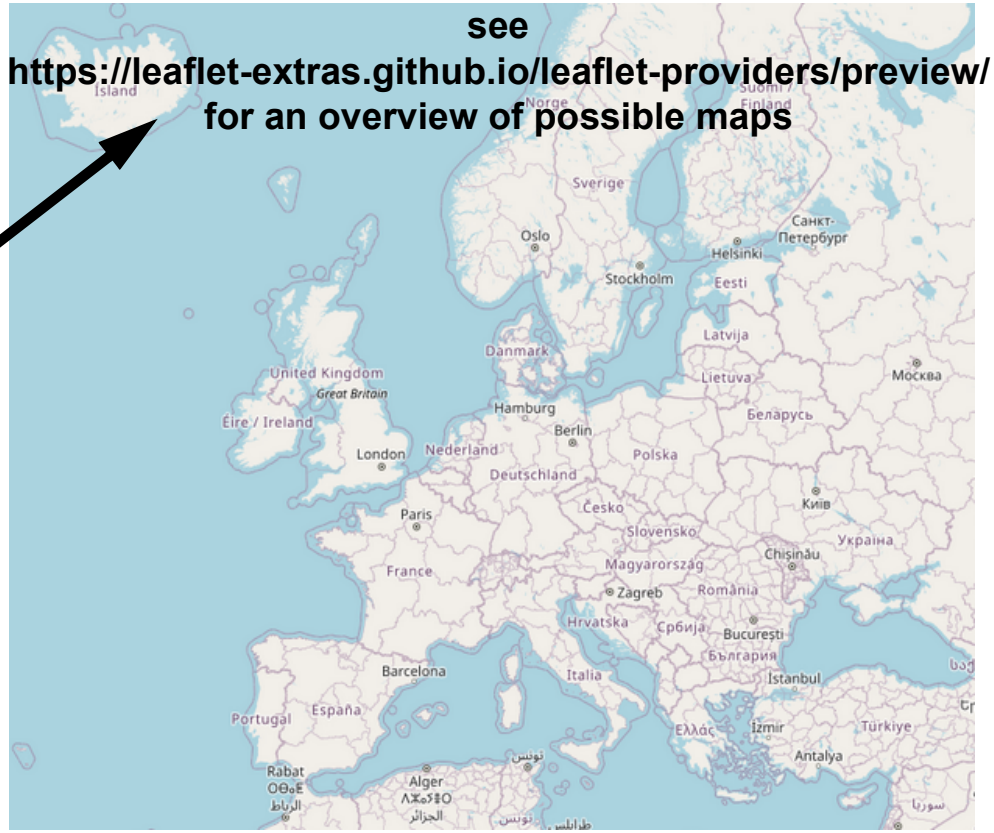
- The general idea is to enable the visualization of maps together with user-specific content in a HTML page.
- Build a web-component for each of leaflets key concepts
  - maps, tiles
  - overlays
  - markers (point)
  - Polyline
  - icons
  - geojson objects
  - groups
- Interaction of maps, tiles, overlays, groups, and geojson objects with leaflets *layer control* (switch on/off a group of artefacts)
- Support of time series databases, GPX and GeoJSON datasources
- Easy connection of leaflet web-compoment attributes with other elements in the page (i.e. select-box, input-box, ...)

Architecture



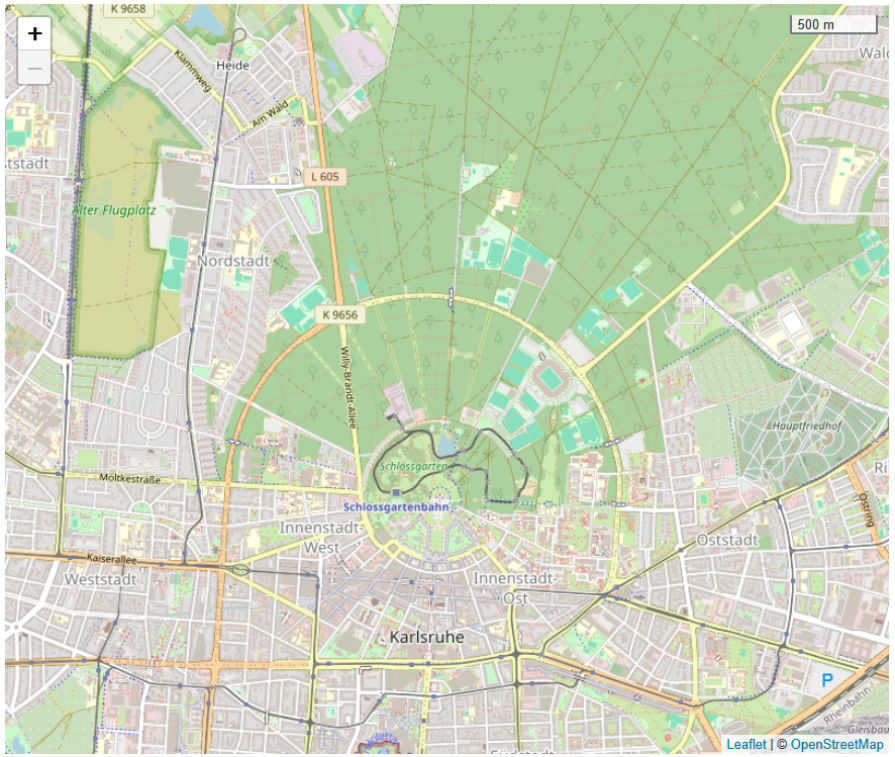
<ll-map>

- Central web-component, responsible for displaying a map in the browser.
- All other elements are subelements of ll-map.
- Specification of general features like ...
  - tile-server (look and feel of base-map)
  - central coordinate and zoom level
    or
  - fixed bounding box
  - Dynamic bounding box (based on subelements)
  - further options



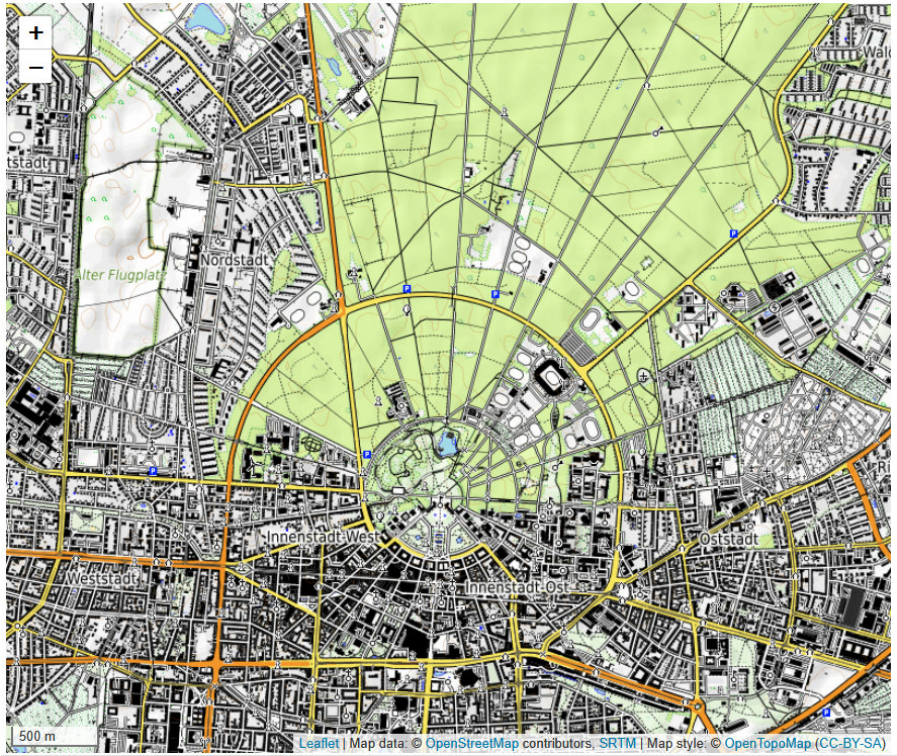
<ll-map> - Example

```
<!DOCTYPE html>
<html>
   <head>
     <script type="module" src="llwc2.js"></script>
   </head>
   <body>
     <div id="map"></div>
     <ll-map
         ref="map"
         height="300px"
         width="100vw"
         lat="49.02"
         lng="8.405"
         zoom="14"
         options='{
                 "min-zoom":14,
                 "max-zoom":16,
                 "scale-position": "topright"
                 }'>
     </ll-map>
   </body>
</html>
```



<ll-map> - Example

```
<!DOCTYPE html>
<html>
   <head>
     <script type="module" src="llwc2.js"></script>
   </head>
   <body>
     <div id="map"></div>
     <ll-map
         ref="map"
         height="300px"
         width="100vw"
         lat="49.02"
         lng="8.405"
         zoom="14"
         options='{
                 "min-zoom":14,
                 "max-zoom":16
                 }'>
         base-url="https://{s}.tile.opentopomap.org/{z}/
{x}/{y}.png"
         attribution='Map data: &copy; ...'
     </ll-map>
   </body>
</html>
```



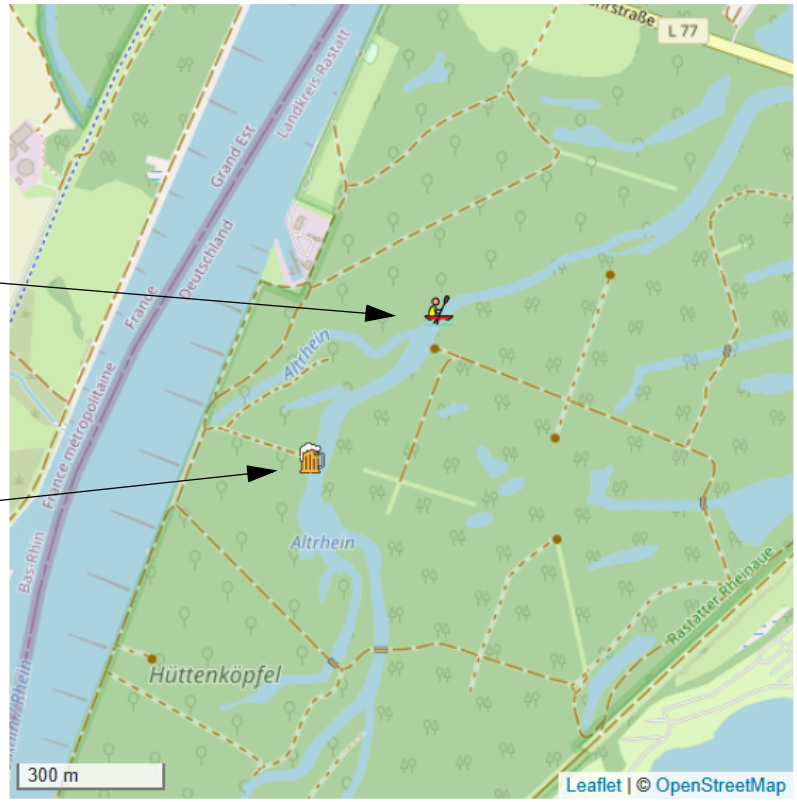
<ll-marker>

- Represents a point on a map
- Can be customized with an icon and size
- coordinates are specified
  - hardcoded
  - obtained from other elements at runtime
  - read from a time series database (json-format) or geoJSON resource

```
<ll-marker id="m1"
      lat="48.875215"
      lng="8.1356334"
      icon-size="20"
      icon="../icons/kanu.png"
      tooltip="Smiff">
</ll-marker>

   <ll-marker id="m2"
         lat="{{latitude.value}}"
         lng="{{longitude.value}}"
         icon-size="20"
         icon="../icons/sailing-boat.png"
         tooltip="Thomas">
   </ll-marker>

<ll-marker id="m3"
      url="./REST/getActualPosition.php?id=m3"
      lat-path="result.lat"
      lng-path="result.long"
      icon-size="20"
      icon="../icons/canoe.png"
      tooltip="Ingo"
      path-color="blue">
</ll-marker>
```

<ll-marker> Example

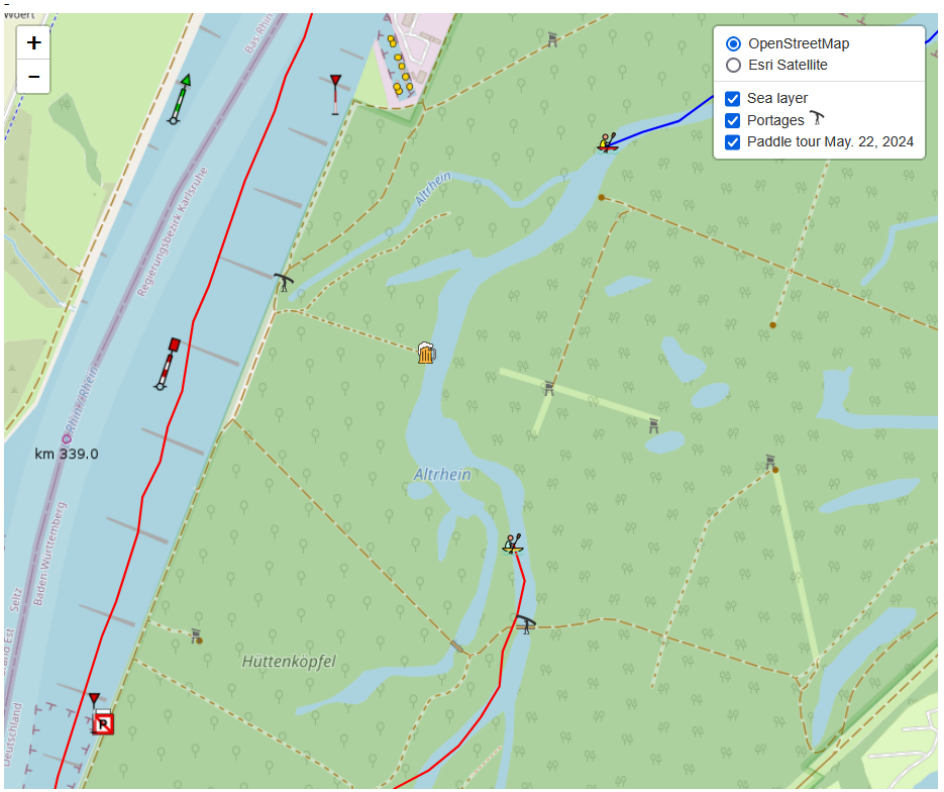
```
<ll-map
     ref="map2"
     zoom="15">
     <ll-marker
          lat="48.880352"
          lng="8.1374788"
          icon="../icons/canoe.png"
          icon-size="20"/>

     <ll-markerlat="48.87767129477924"
          lng="8.133959770202638"
          icon-size="20"
          icon="../icons/beer.png"/>
</ll-map>
```



dynamic determination of center of the map and zoom level, based on subelements

<ll-tile> & <ll-overlay>

- Add further maps
  - alternate base-maps
  - overlay maps (on top of base-map)
- If more than one basemap (via <ll-tile>) is specified, you can choose one from the upper part of the layer-control
- zero or more overlays can be selected from the lower part of the layer control

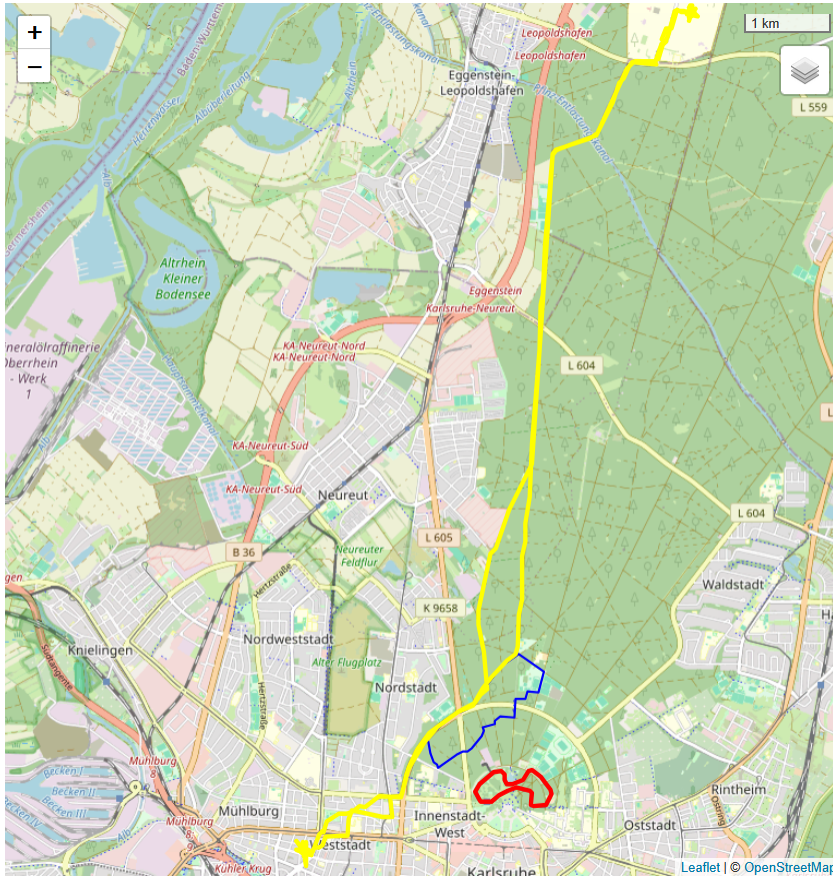


<ll-tile> & <ll-overlay>

```
<ll-map ref="map2"
        lat="48.880352"
        lng="8.1374788"
        zoom="16">
    <ll-tile url='http://.../World_Imagery/MapServer/tile/{z}/{y}/{x}'>
        &copy; <a href="http://www.esri.com/">Esri Satellite</a>
    </ll-tile>

    <ll-overlay url="http://t1.openseamap.org/seamark/{z}/{x}/{y}.png"
                label="Sea Layer">
        &copy; <a href="http://t1.openseamap.org/copyright">OpenSeaMap</a>
    </ll-overlay>

    ...
</ll-map>
```

<ll-polyline>

- Draws a multiline
- Sources:
  - json- array holding lat, lng pairs
  - gpx-file (one trk)
  - Time series database (json format)
  - geoJSON resource



<ll-polyline> Examples

```
<ll-polyline color="red" weight="2"
             points="[
                      [48.890512743964791, 8.148843431845307],
                      [48.890063557773829, 8.148915097117424],
                      [48.889614371582866, 8.148628436028957]
                      ...
                     ]">
</ll-polyline>

<ll-polyline color="blue" weight="2"
             gpx="../data/Schlossgartenpfad.gpx">
</ll-polyline>

<ll-polyline color="yellow" weight="4"
             url="http://localhost/gps-location/readFromInflux.php/smiff?from=2025-01-
20T6:00:00.000Z&to=2025-01-20T23:00:00.000Z&mmnt=dbkda-demo"
             lat-path="result.lat"
             lng-path="result.lon">
</ll-polyline>
```

<ll-geojson>

- Support of import from GeoJSON objects
- Single object or **multiple** objects import possible
- <ll-geojson> element can also act as group (if attribute „label“ is set) and its members can activated/deactivated via layer control.
- Example:

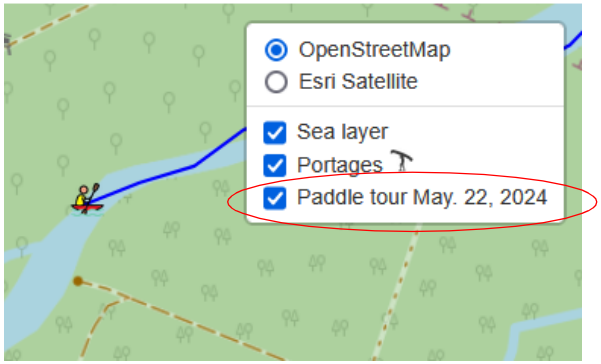
```
<ll-geojson
    label="Portages &lt;img src=&apos;icons/portage.png&apos; width=&apos;15&apos;&gt;"
    icon-url="./icons/portage.png"
    url="http://localhost/llwc/readGeoJSON.php?file=data/portages.json">
    path="result">
</ll-geojson>
```

see portages on Slide 13

<ll-group>

- Group multiple elements (i.e. markers, polylines, circles, ...) together, so that they can be shown/hidden together

- groups appears in the lower part of the layer-control (see attribute „label“) and so the elements in this group can be activated/ deactivated together




```
<ll-group label="Paddle tour May 22, 2024">
    <ll-marker
       icon="icons/kanu2.png" icon-size="20"
       url="./REST/getActualPosition.php?id=m3"
       lat_path="result[0].lat"
       lng_path="result[0].lon"
       path-color="blue">
    </ll-marker>

    <ll-marker icon="icons/kanu.png" icon-size="20"
       tooltip="Andreas"
       url="./REST/getActualPosition.php?id=m7"
       lat_path="result[0].lat"
       lng_path="result[0].lon"
       path-color="red">
    </ll-marker>
    ...
</ll-group>
```

Karlsruhe Institute of Technology

münch IT SOLUTIONS

UNIVERSITY OF TECHNOLOGY IN THE EUROPEAN CAPITAL OF CULTURE CHEMNITZ

Hochschule Karlsruhe University of Applied Sciences
Fakultät für Informatik und Wirtschaftsinformatik

Summary

- Library of web-components to build web-mapping applications
- Encapsulation of Javascript code behind intuitive markup interface
- Low-code environment that allows the creation of interactive map-applications without coding (markup only)
- Automatic coupling of components
- connection of the components to external databases, like GeoJSON sources and time series databases
- Can easily be extended with supplemental javascript code to build application specific bahaviour (access to the leaflet instances of map, markers, groups, geoJSON objects, ...)

Outlook

- Evaluate the usefulness of our components
  - Test our components in real-world scenarios (at IAI)
  - Build online game, based on our components (course at University of Applied Sciences)
- Extend functionality
  - move, drag & drop of markers, geoJSON objects
  - integrate leaflet draw functionality
- Filter functionality, based on properties of imported GeoJSON objects
- Wizard to build markup code
- Reimplementation using Lit-library [3]
- Declarative interface to the leaflet event system.

Resources

[1] Volodymyr Agafonkin, leaflet-library: https://github.com/Leaflet/Leaflet

[2] Michael Dorman, Introduction to Web Mapping, CRC Press, 2020

[3] Lit library. https://lit.dev/

[4] Gary Sherman, Leaflet Cookbook: Recipes for Creating Dynamic Web Maps, Locate Press, 2019