



Building Mapping Solutions with Esri Open Source Projects

Andrew Turner - @ajturner
andrew@esri.com

CTO, Esri R&D, Washington, DC

Esri Developer Summit
Washington, DC



The screenshot shows a web browser window with the address bar displaying 'esri.github.io'. The page features a dark blue header with a satellite map background. The main content area is white and contains a search filter and a grid of project cards. Each card displays the project name, language, description, and GitHub statistics (forks and stars).

Esri is on GitHub!
We're excited about helping developers build and share software. Browse our open source code and get started with our powerful ArcGIS platform.

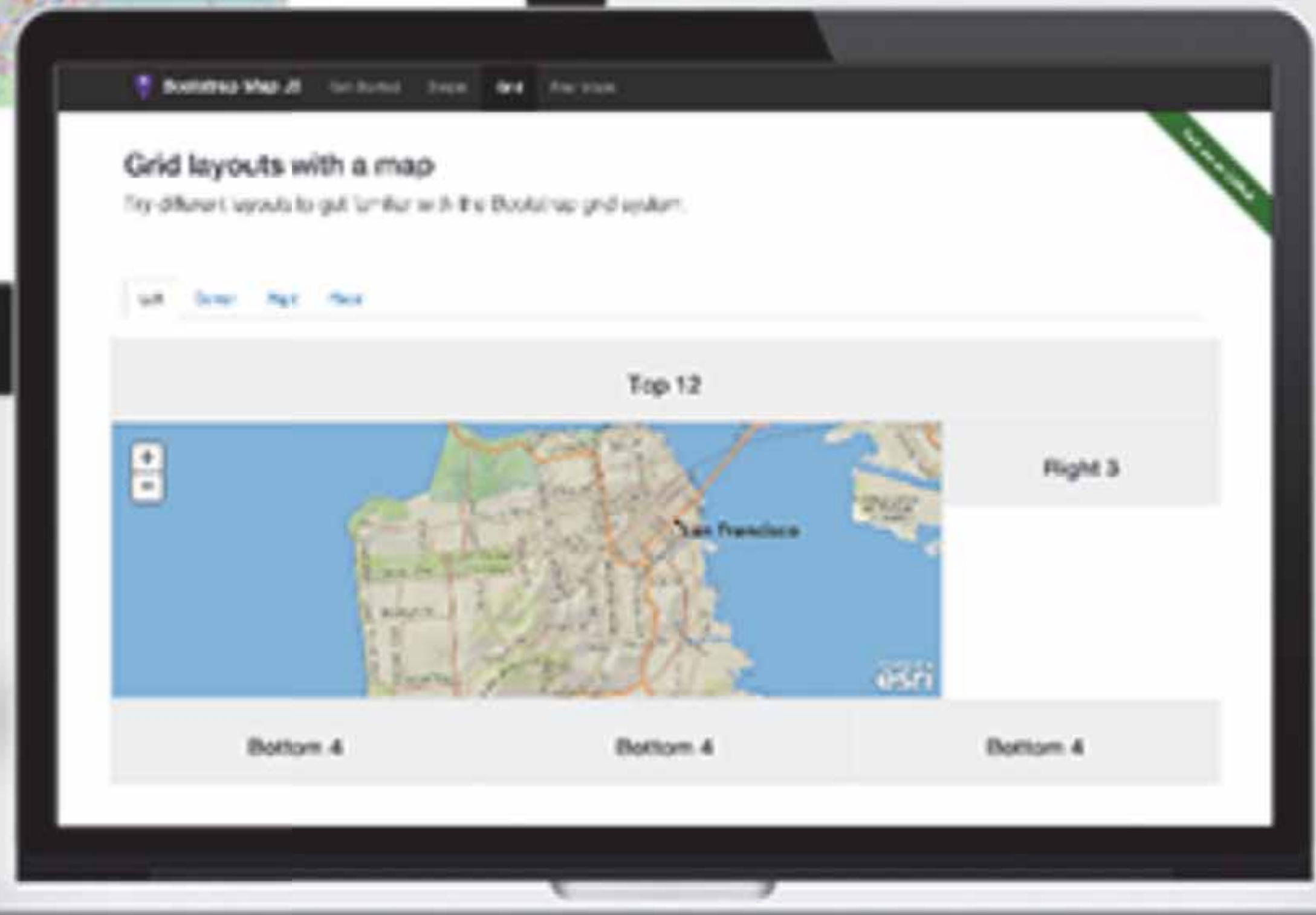
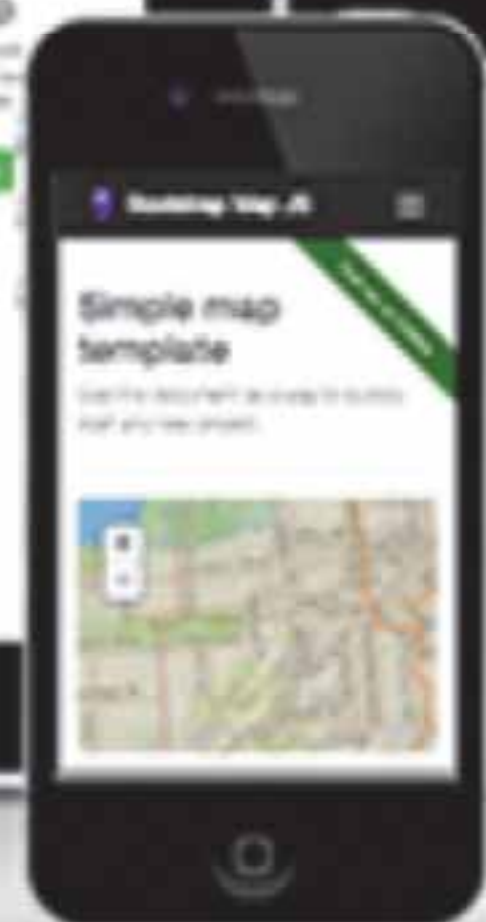
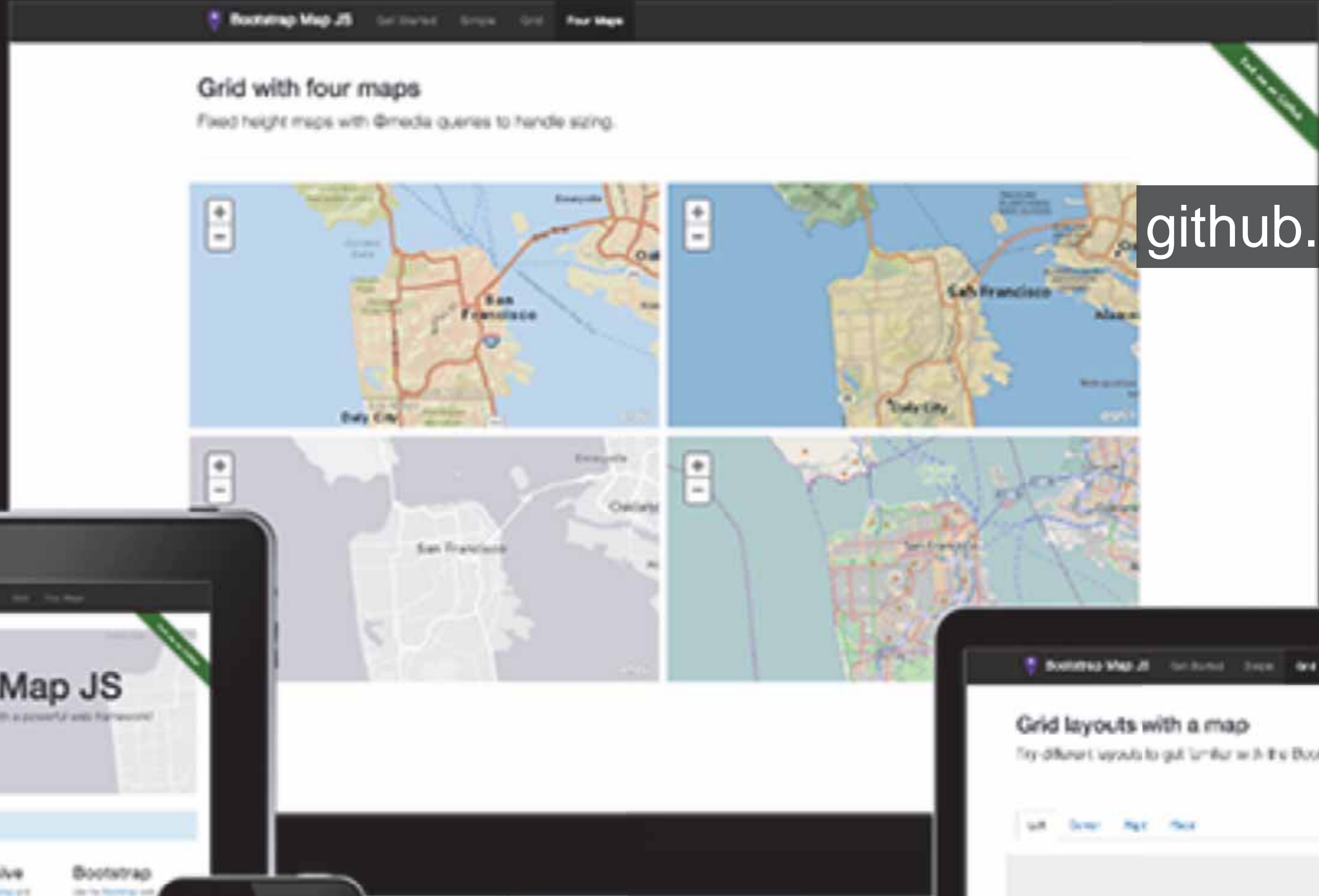
[BROWSE ON GITHUB](#)

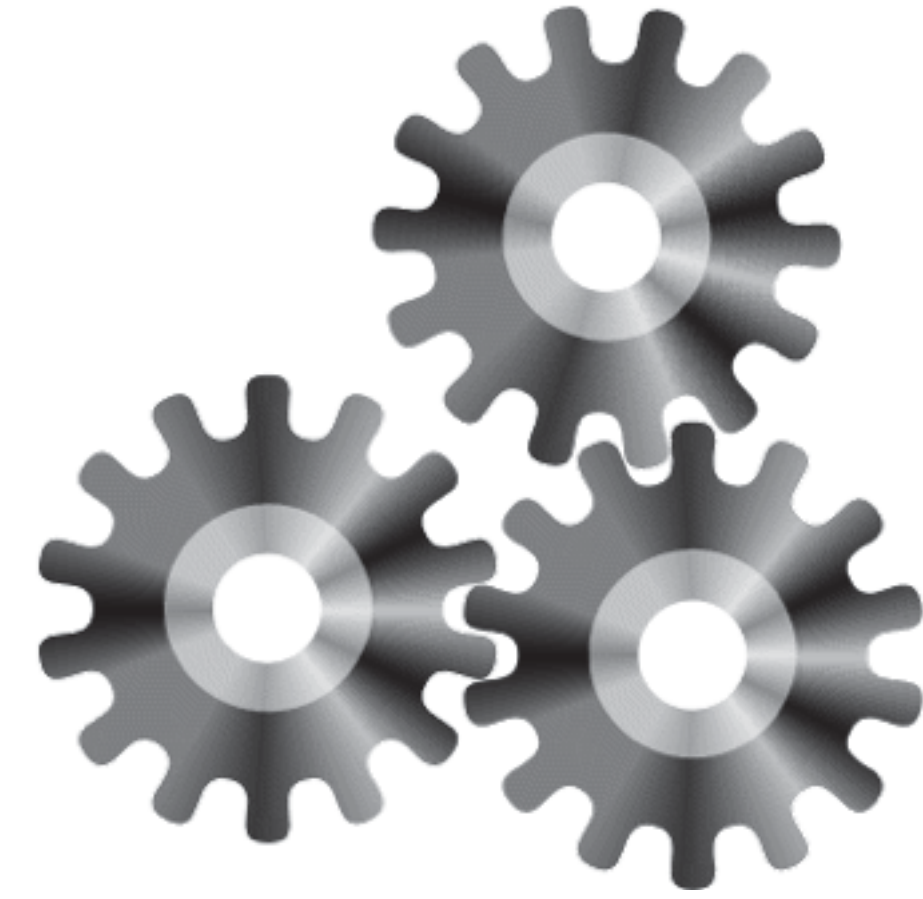
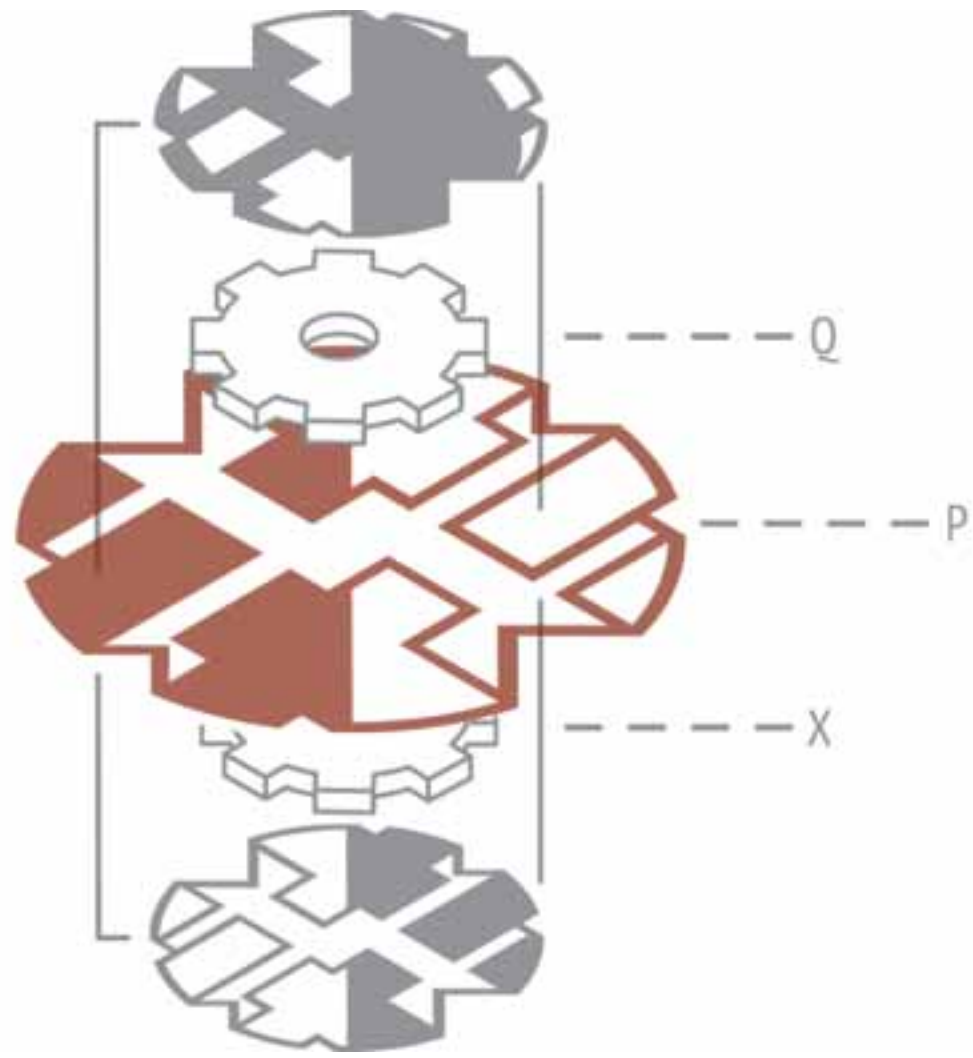
Need an [ArcGIS subscription](#)? Start developing today for free.

Filter by language or keyword:

Project Name	Language	Description	Forks	Stars
esri-leaflet	JavaScript	A lightweight set of tools for working with ArcGIS services with Leaflet.	83	133
terraformer	JavaScript	A geometric toolkit for dealing with geometry, geography, formats, and building geo databases	41	124
arcgis-viewer-flex	ActionScript	Source code for ArcGIS Viewer for Flex – a great application framework for web applications.	116	119
geometry-api-java	Java	The Esri Geometry API for Java enables developers to write custom applications for analysis of spatial data. This API is used in the Esri GIS Tools for Hadoop and other 3rd-party data processing solutions.	28	58
gis-tools-for-hadoop	Python	The GIS Tools for Hadoop are a collection of GIS tools for spatial analysis of big data.	28	57
geoportal-server	Java	Geoportal Server is a standards-based, open source product that enables discovery and use of geospatial resources including data and services.	51	44
geojson-utils	JavaScript			

github.com/Esri/bootstrap-map-js





Leaflet 



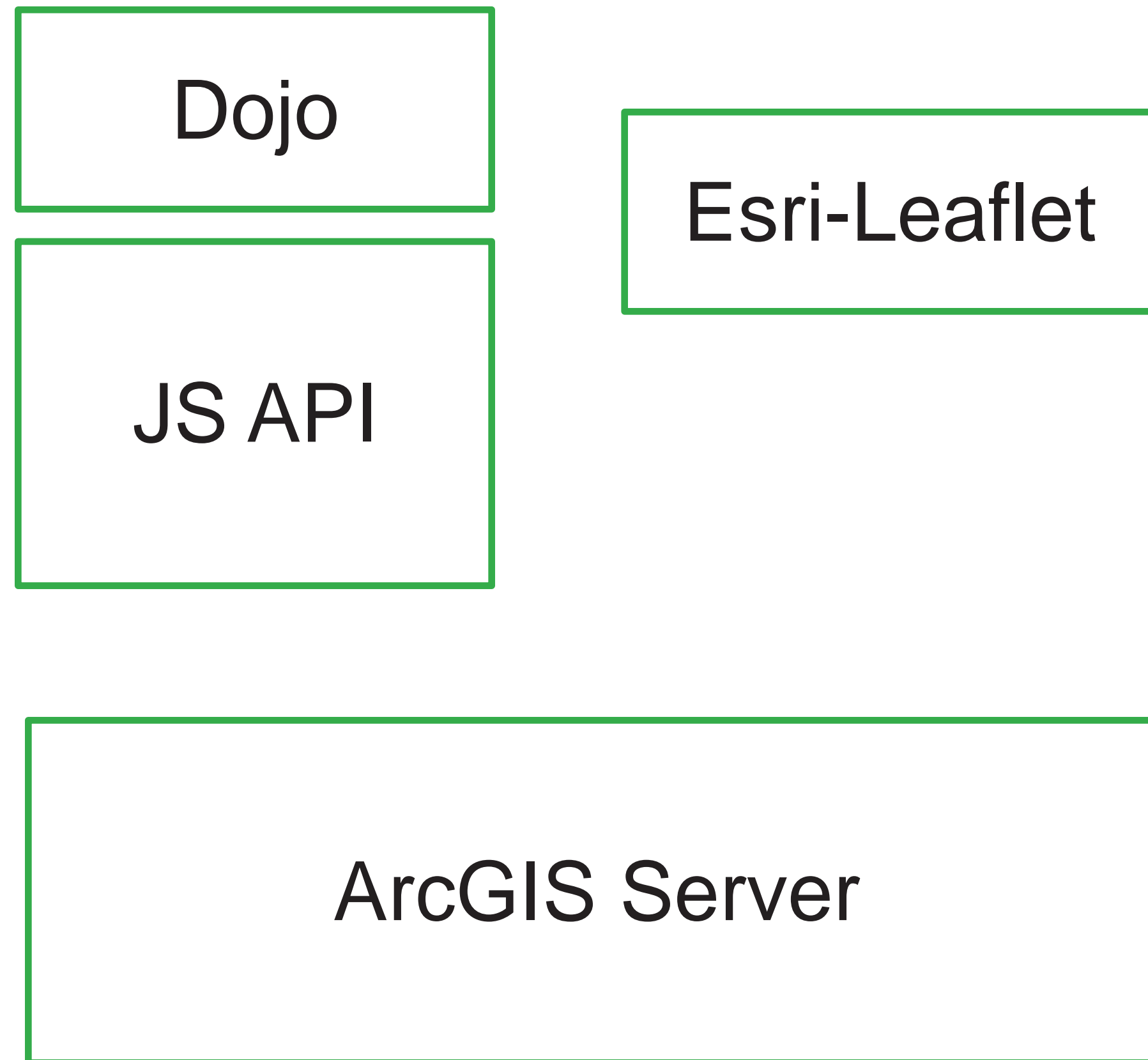
Open Patterns

Dojo

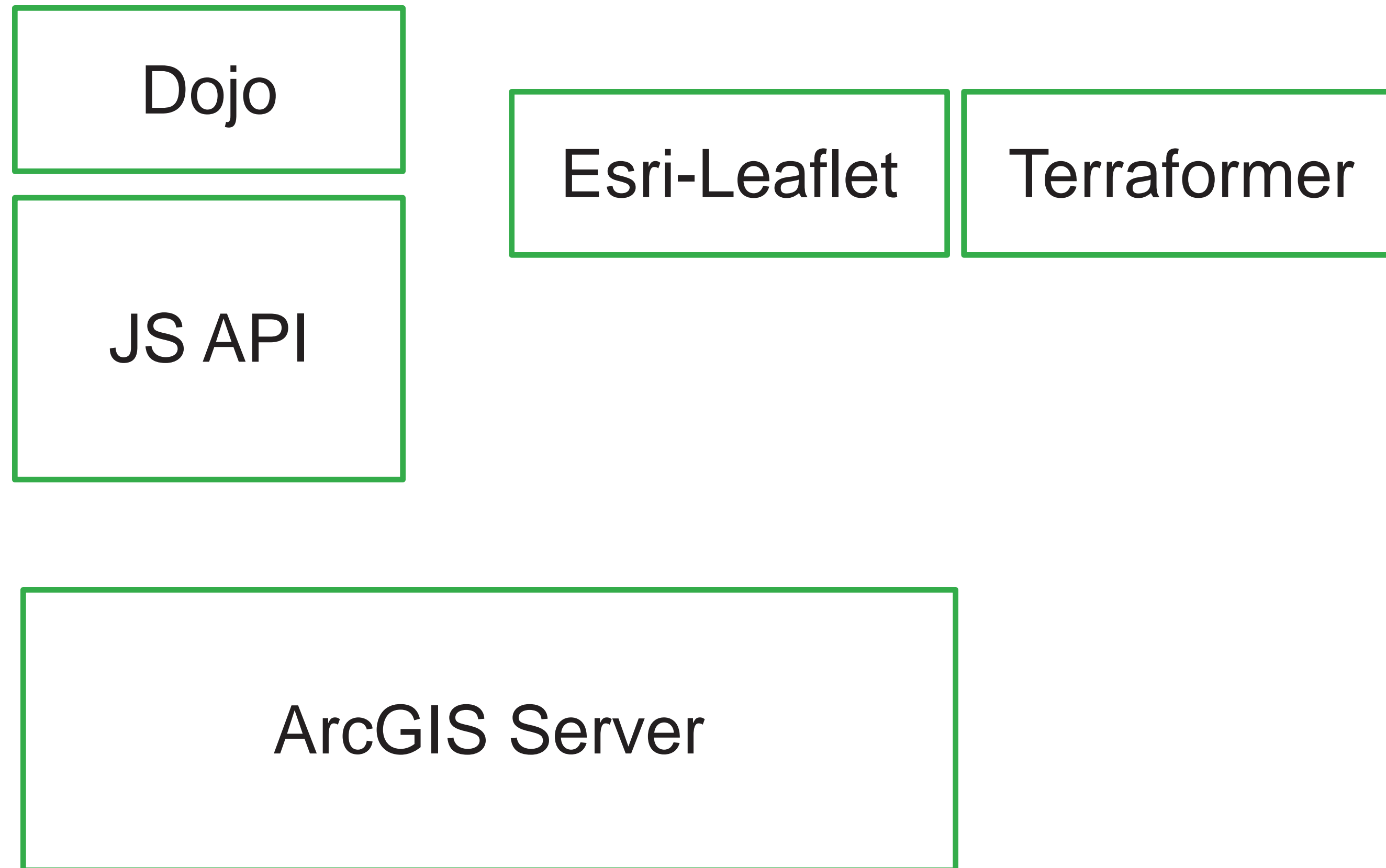
JS API

ArcGIS Server

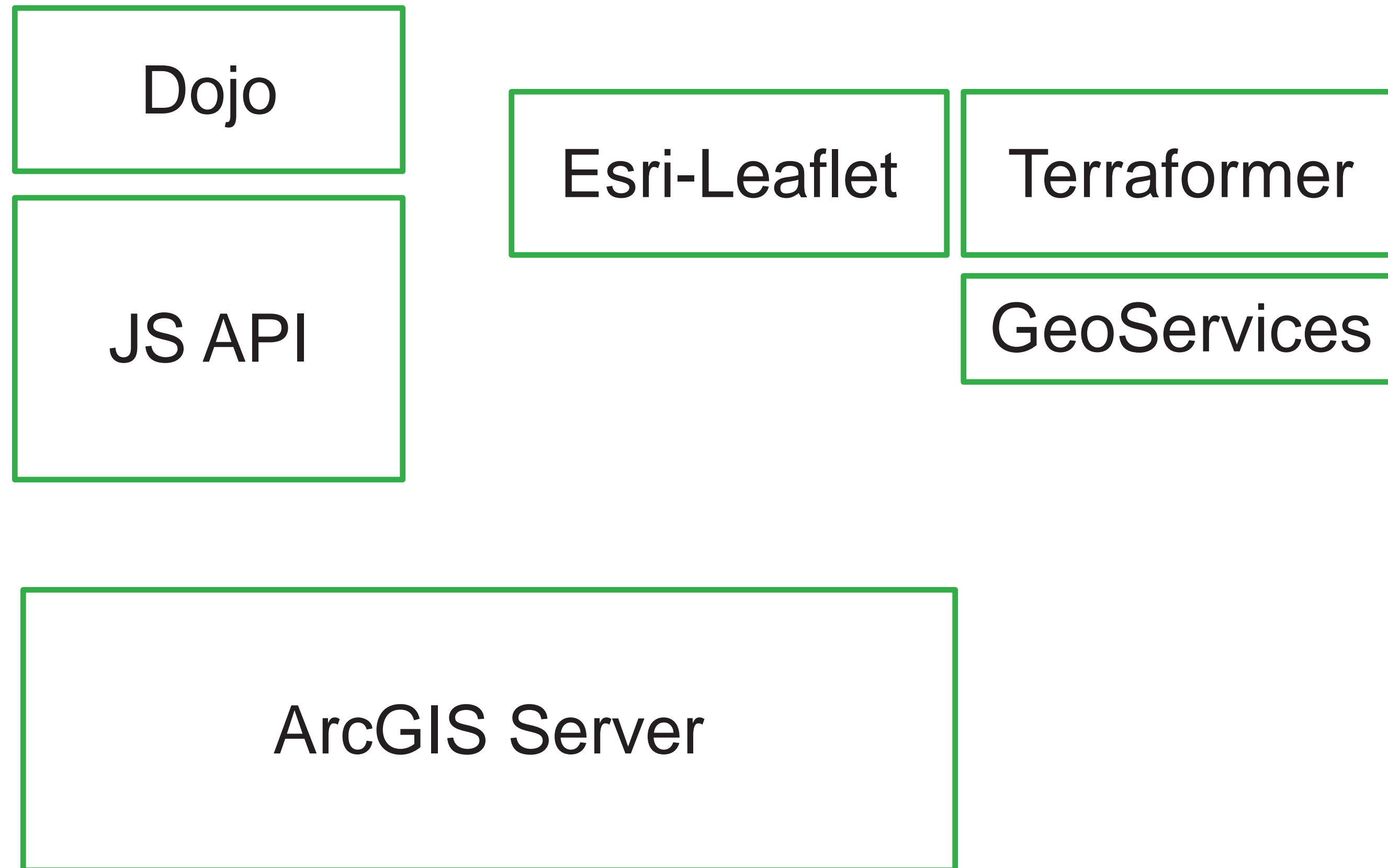
Open Patterns



Open Patterns



Open Patterns



Open Patterns

Dojo

JS API

Esri-Leaflet

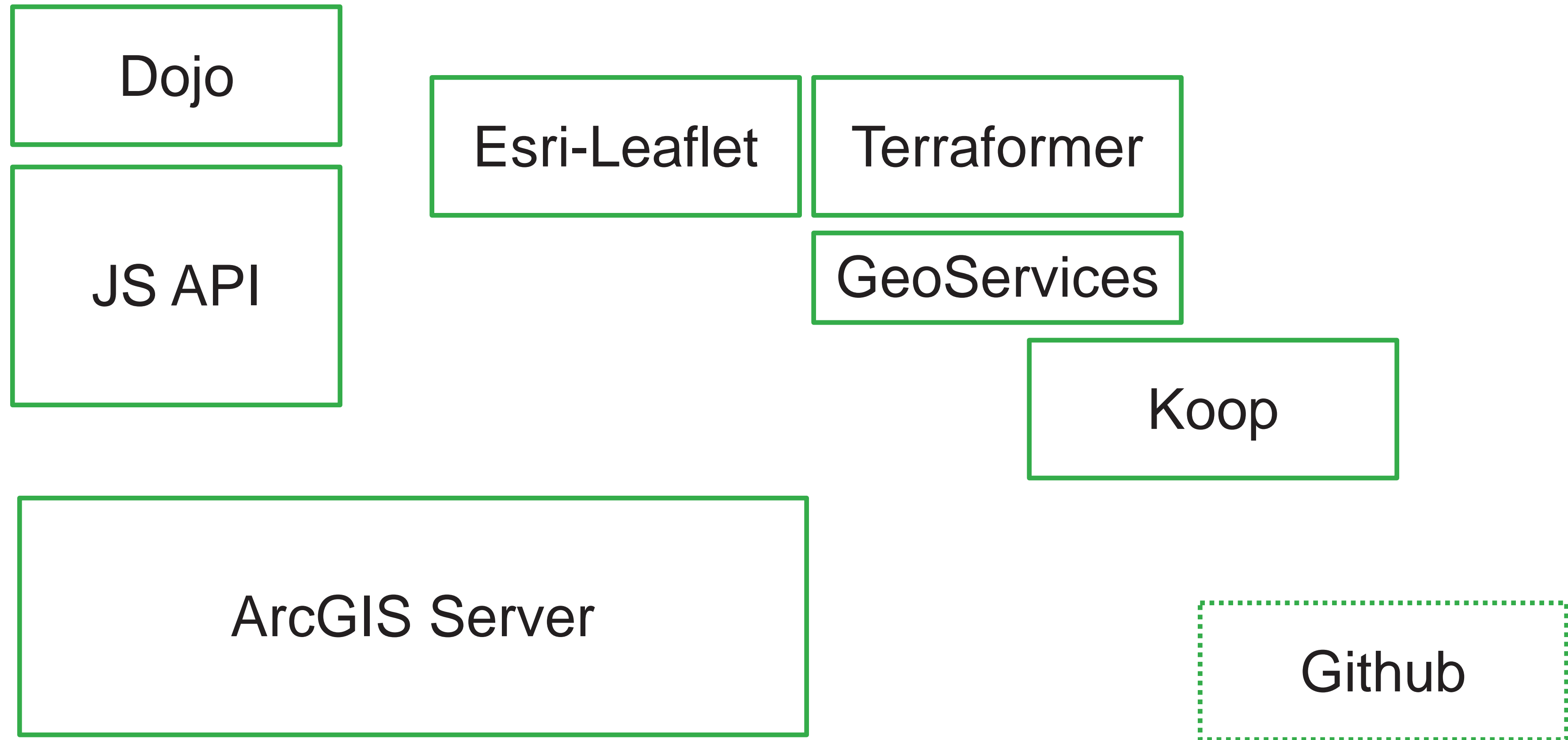
Terraformer

GeoServices

ArcGIS Server

Github

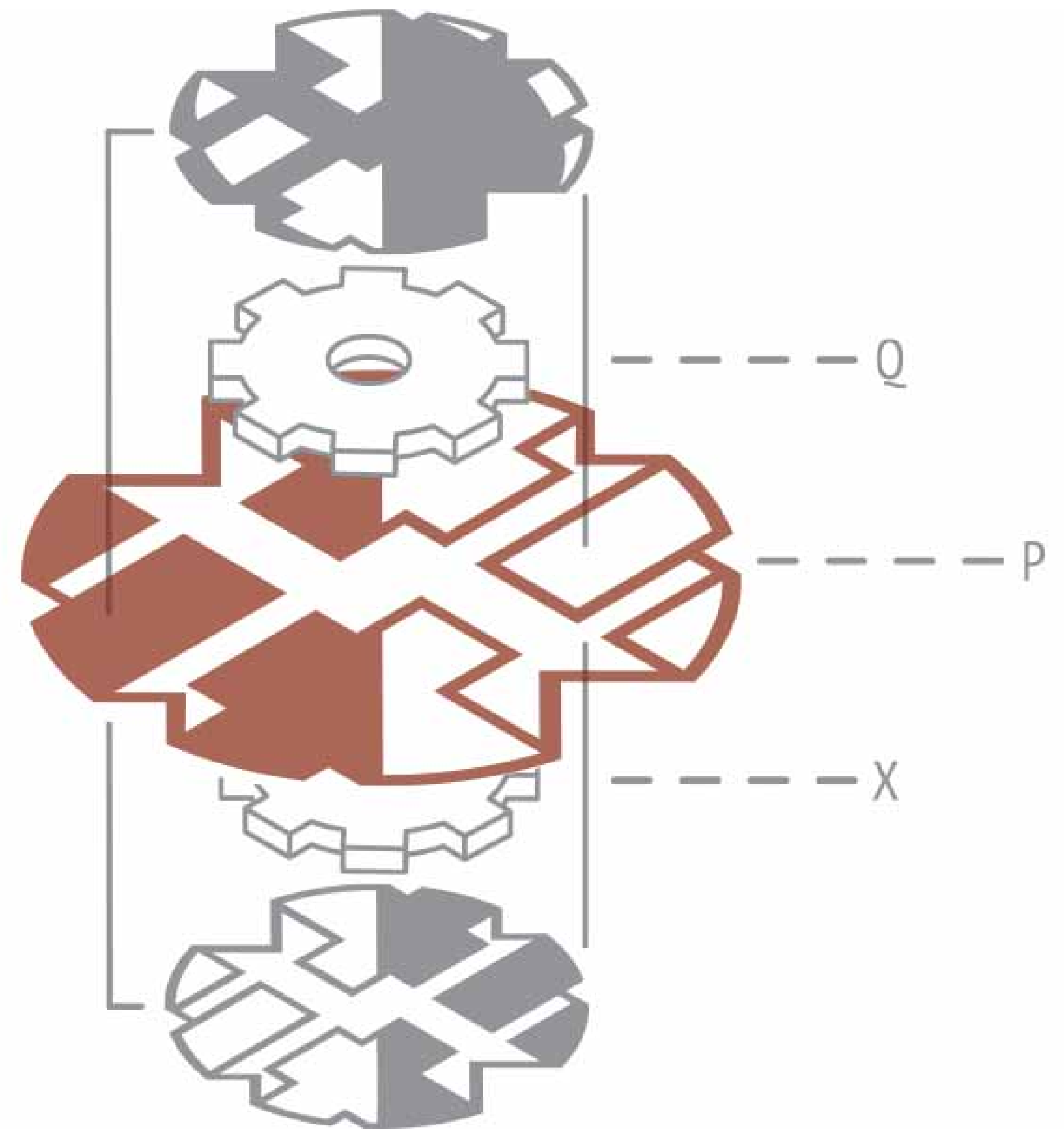
Open Patterns



Terraformer

Geodata JavaScript Library

<http://terraformer.io/>



Terraformer

- Open source geometry and geodata library
- Key features
 - Geometry format conversions (GeoJSON)
 - Geometry operations
 - Coordinate system conversion
 - Store and access data
- Node.js and client-side JavaScript

github.com/Esri/Terraformer

Terraformer Modules

- Terraformer
- terraformer-arcgis-parser
- terraformer-wkt-parser
- terraformer-geostore
- terraformer-geostore-rtree
- terraformer-geostore-memory
- terraformer-geostore-localstorage

Terraformer: Geometry and Features

terraformer.js

```
// create a typed primitive from GeoJSON
var point = new Terraformer.Primitive({ "type": "Point", "coordinates": [ 100, 1 ] });

// create a Geometry from coordinates or GeoJSON
var point = new Terraformer.Point( [ 10, 10 ] );
var ls = new Terraformer.LineString([ [ 10, 10 ], [ 20, 20 ]]);
var poly = new Terraformer.Polygon([
  [[100.0, 0.0], [101.0, 0.0], [101.0, 1.0], [100.0, 1.0]]]);
var circle = new Terraformer.Circle([-122.6764, 45.5165], 1000);

// creates a feature from a valid GeoJSON Object
var feature = new Terraformer.Feature({ "type": "Point", "coordinates": [ 10, 10 ]},
  "properties": { "prop0": "value0" });
```

Terraformer: Geometric Operations

terraformer.js

```
// to Web Mercator and WGS84
primitive.toMercator();
primitive.toGeographic();

var box = poly.bbox;
var ev = polygon.envelope();

multi.addPoint([ 10, 10 ]);
multi.insertPoint([ 10, 10 ],1);
multi.removePoint(1);
multi.get(1);

polygon1.within(polygon2);
polygon1.intersects(line);
polygon1.contains(point);
circle.contains(point);
```


Terraformer: WKT Conversion

[terraformer-wkt-parser.js](#)

```
// take a WKT representation and convert it into a primitive
<script>
  var primitive = Terraformer.WKT.parse('LINESTRING (30 10, 10 30, 40 40)');
</script>

// take a primitive and convert it into a WKT representation
var polygon = Terraformer.WKT.convert(
  {
    "type": "Polygon",
    "coordinates": [
      [ [100.0, 0.0], [101.0, 0.0], [101.0, 1.0], [100.0, 1.0], [100.0,
0.0] ],
      [ [100.2, 0.2], [100.8, 0.2], [100.8, 0.8], [100.2, 0.8], [100.2,
0.2] ]
    ]
  }
);
```

Terraformer: ArcGIS JSON to GeoJSON

[terraformer-arcgis-parser.js](#)

```
<script>
  // take ArcGIS JSON and convert to Primitive or GeoJSON
var primitive = Terraformer.ArcGIS.parse({
  x: "-122.6764",
  y: "45.5165",
  spatialReference: {
    wkid: 4326
  }
});

// take a Primitive or GeoJSON and convert it to ArcGIS JSON
var point = Terraformer.ArcGIS.convert({
  "type": "Point",
  "coordinates": [45.5165, -122.6764]
});
</script>
```

Terraformer: GeoStore

[terraformer-geostore.js](#) and [terraformer-rtree.js](#)

```
// In-memory geostore. Requires id property.
var store = new Terraformer.GeoStore({
  store: new Terraformer.GeoStore.Memory(),
  index: new Terraformer.RTree()
});

store.add(geojson, function(err, resp){
  // callback
});

store.update(geojson, function(err, resp){
  // callback
});

store.contains(geojson, function(err, resp){
  // callback
});
```

Terraformer: GeoStore

alternate indexes

```
// create an index on the properties.street value
gs.addIndex({
  property: "street",
  index: new BTree()
});
```

```
// create an index on the properties.crime value
gs.addIndex({
  property: "crime",
  index: new BTree()
});
```

Terraformer: GeoStore

alternate indexes

```
// create an index on the properties.street value
gs.addIndex({
  property: "street",
  index: new BTree()
});
```

```
// create an index on the properties.crime value
gs.addIndex({
  property: "crime",
  index: new BTree()
});
```

```
gs.within(
  geojson,
  {
    "name": {
      "equals": "Main"
    },
    "crime": {
      "equals": "Arson"
    }
  },
  function (err, res) {
    // node.js style callback
  }
);
```

Geoservices-js

ArcGIS REST Service API



ArcGIS Server

Geoservices-js

ArcGIS REST Service API

Terraformer

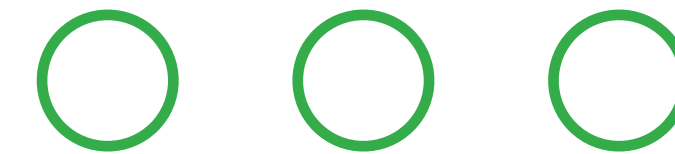


ArcGIS Server

Geoservices-js

ArcGIS REST Service API

Terraformer



GeoServices-js

ArcGIS Server

Geoservices-js

- Open source Geoservices REST API
- Communicate with ArcGIS REST services
- Key features
 - Light-weight, pure JavaScript
 - Browser and Node.js
 - Built on the [Geoservices REST specification](#)

esri.com/opengeoservices

github.com/Esri/geoservices-js

Geoservices-js: Getting Started

geoservices.js

```
// Browser
<script src="browser/geoservices.js"></script>
<script>
  var client = new Geoservices();
</script>
```

```
// Node.js
var Geoservices = require('geoservices');
var client = new Geoservices();
```

Geoservices-js: FeatureService Info

[geoservices.js](#)

```
// Define parameters for a feature service
var params = {
  catalog: 'http://server6.arcgisonline.com/arcgis/rest/services',
  service: 'Census',
  type: 'MapServer',
  layer: 3
};

// Make request to the service
client.FeatureService( params , function (err, result) {
  if (err) {
    console.error("ERROR: " + err);
  } else {
    console.log("Got the FeatureService Metadata: ", result );
  }
});
```

Geoservices-js: FeatureService Query

[geoservices.js](#)

```
// Define query parameters
var query_params = {
  f: 'json',
  returnGeometry: true,
  where: '1=1',
  outSR: '4326'
};

// Request features
var fs = client.FeatureService( params , function(err, data){
  fs.query( query_params, function( err, result ){
    if (err) {
      console.error("ERROR: " + err);
    } else {
      console.log("Features: ", result );
    }
  });
});
```

Geoservices-js: Geocoding

geoservices.js

```
// Geosearch
client.geocode({ text: "920 SW 3rd Ave, Portland, OR 97201" },
function (err, result) {
  if (!err) {
    console.log(result.locations[0].feature.geometry.y + ", "
result.locations[0].feature.geometry.x);
  }
});

// Reverse-geocoding
client.geocode.reverse({ location: "-122.67633,45.51673" },
function (err, result) {
  if (!err){
    console.log(result.address.Address + ", " + result.address.City);
  }
});
```

Geoservices-js: Batch Geocoding

geoservices.js

```
// Simple authentication only!
var client = new Geoservices();
client.authentication.authenticate('username', 'password', { /*
optional options */ }, callback);

// Batch geocoding
var batch = new client.geocode.Batch();

// add addresses to geocode
batch.geocode("123 Fake Street");
batch.geocode("456 Other Street");

// run the batch
batch.run(function (err, results) {
    console.dir(results);
});
```

Koop

Node.js GeoServices
REST Implementations

Dojo

JS API



Koop

Node.js GeoServices
REST Implementations

Dojo

Esri-Leaflet

JS API



Koop

Node.js GeoServices
REST Implementations

Dojo

JS API

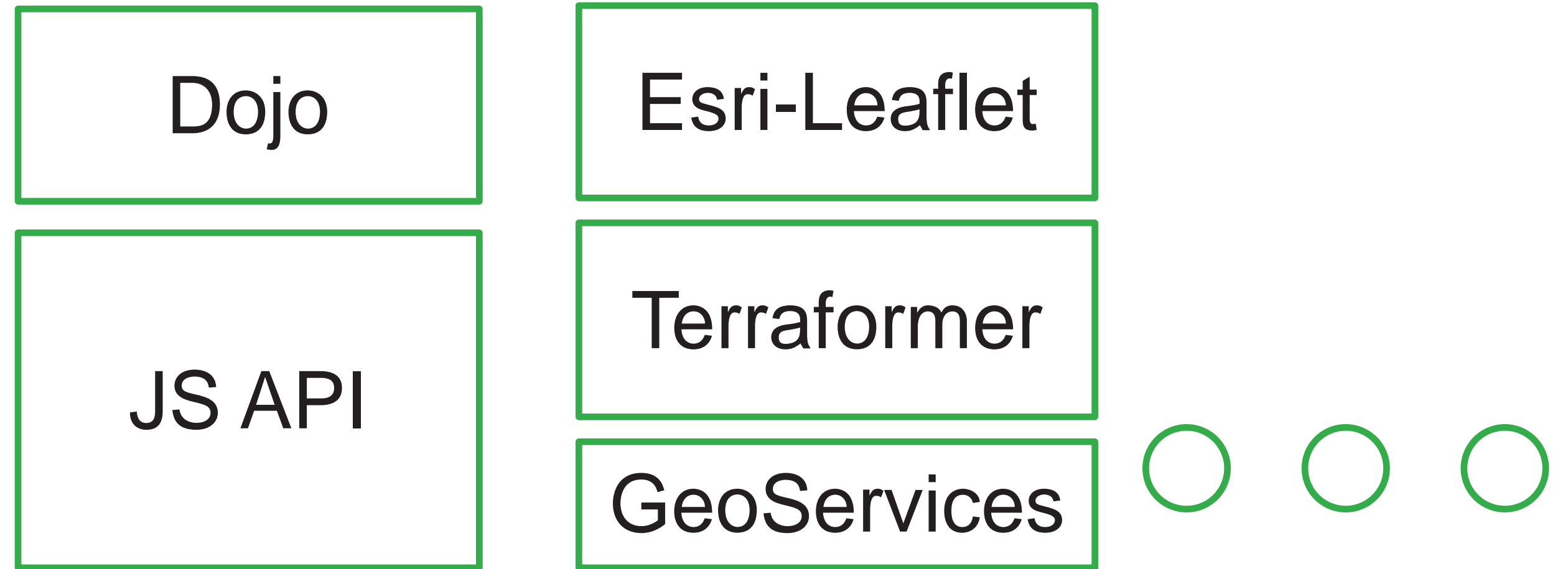
Esri-Leaflet

Terraformer



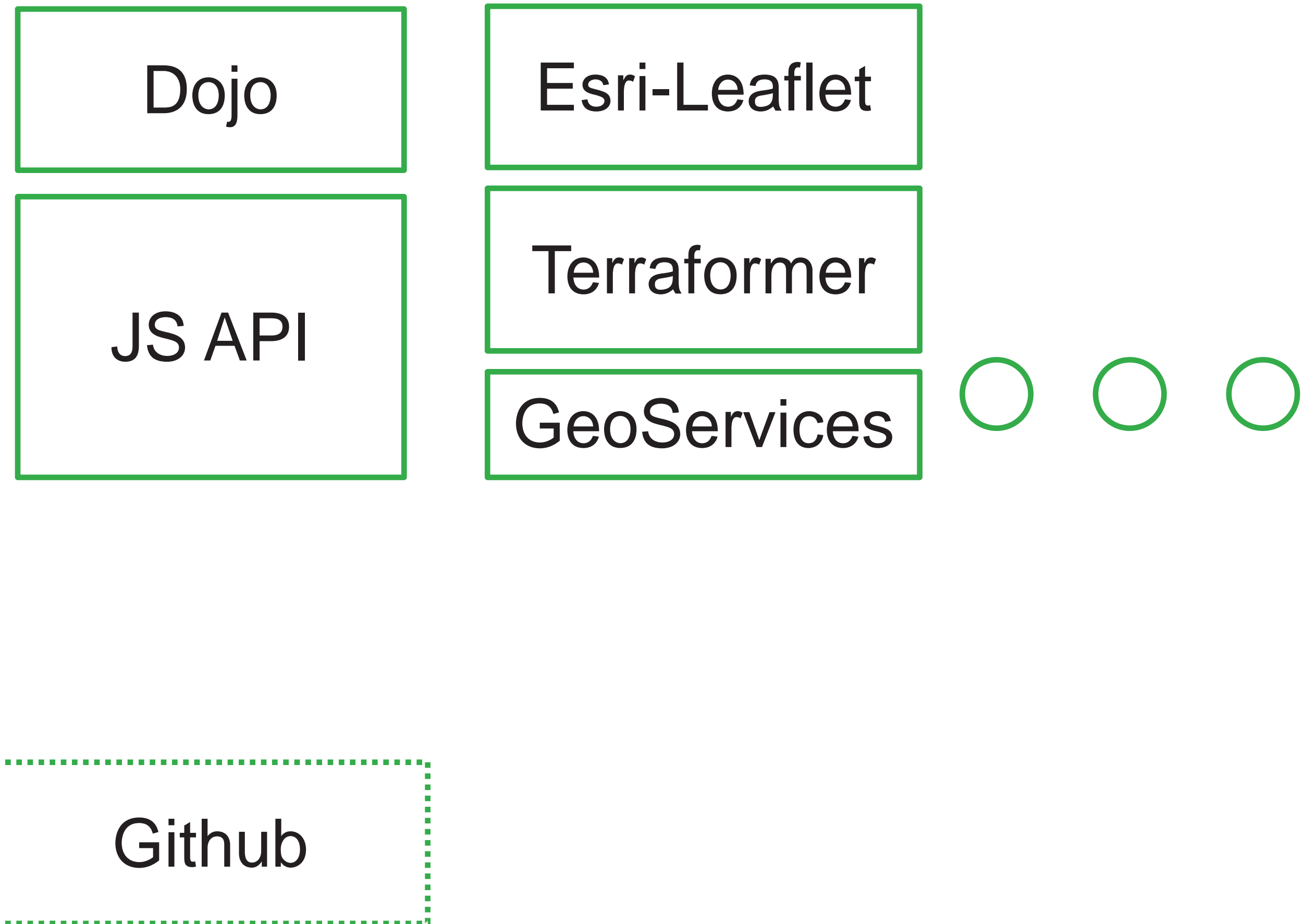
Koop

Node.js GeoServices
REST Implementations



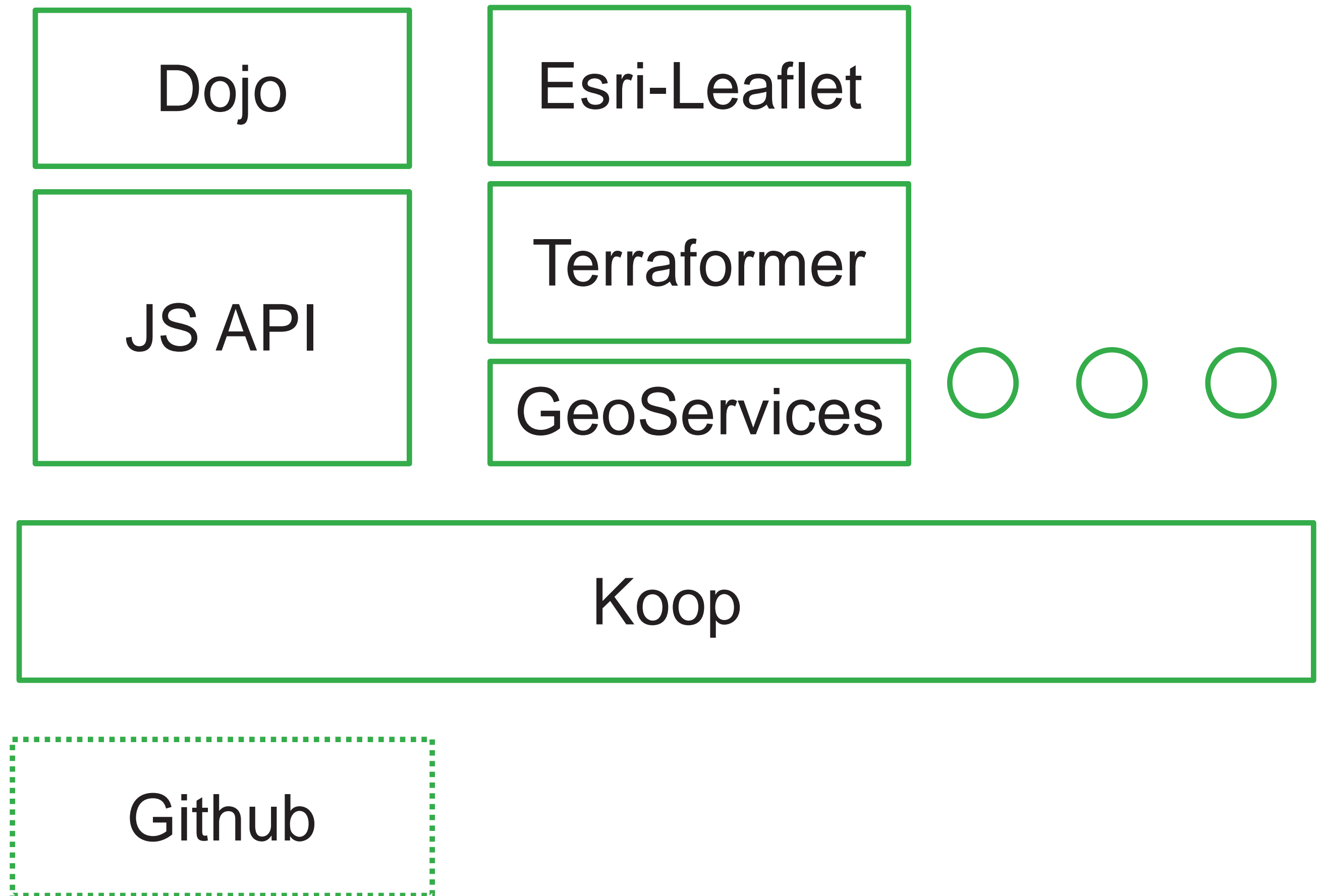
Koop

Node.js GeoServices
REST Implementations



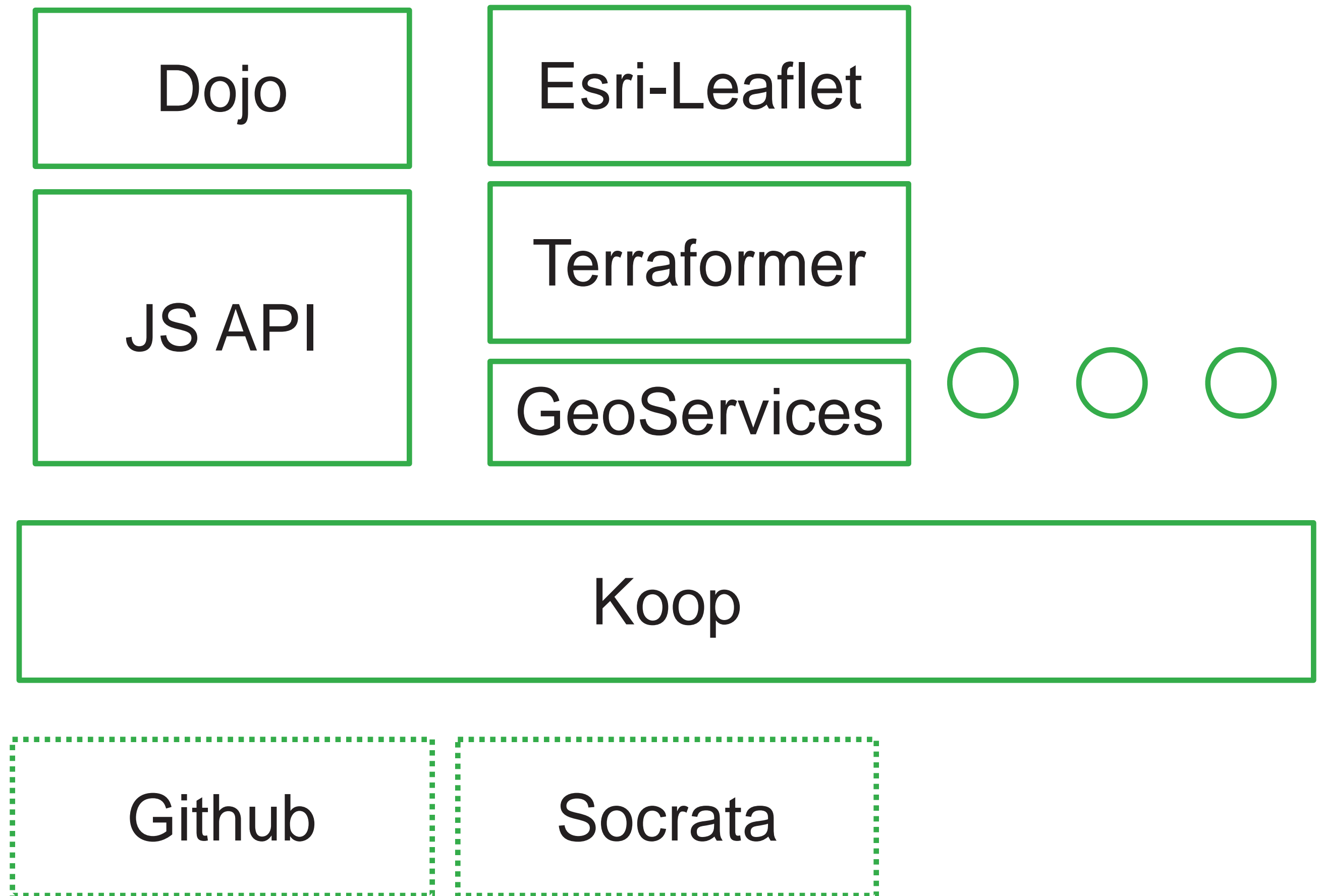
Koop

Node.js GeoServices
REST Implementations



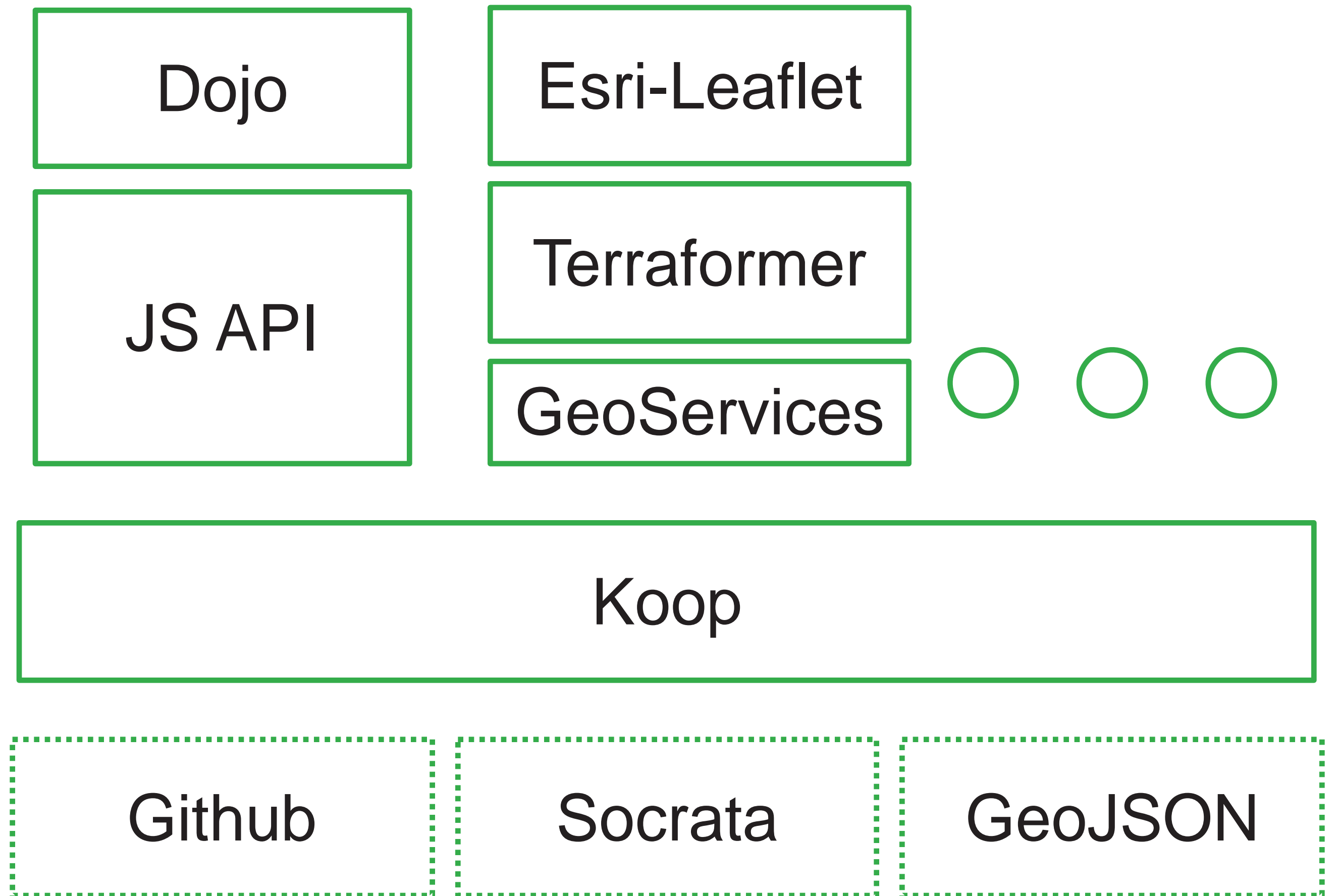
Koop

Node.js GeoServices
REST Implementations

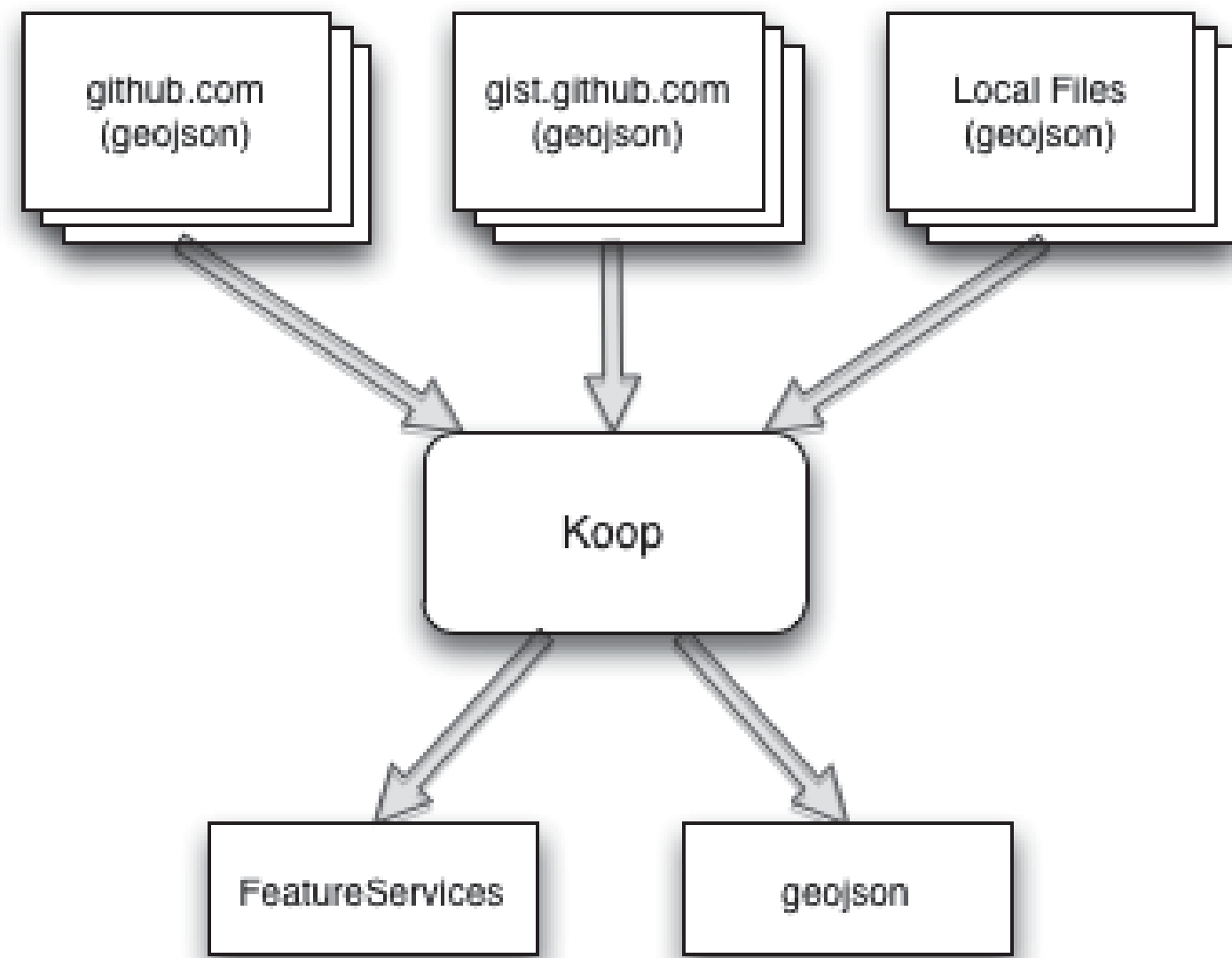


Koop

Node.js GeoServices
REST Implementations



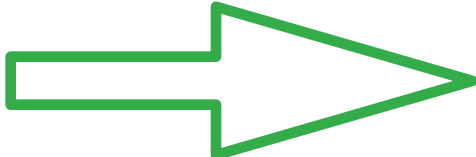
Koop



- Open source ArcGIS REST provider
- Expose “your” service as an ArcGIS service
- Node.js implementation
- Can consume “any” service

Koop: From the Client-side

- Consumable by different ArcGIS clients
- ArcGIS map viewer
- Store as ArcGIS Online “items”
- Follows the [Geoservices REST](#) specification

<http://heorb.it/githubphl>  <http://heorb.it/koophl>

Koop: GitHub Provider Example

[Github.js](#)

MVC

```
// routes/index.js
```

```
module.exports {  
  'get /github/:user/:repo/FeatureServer/:layer/:method': {  
    controller: 'github',  
    action: 'featureservice'  
  },  
}
```

...

```
// model/github.js
```

```
var Geohub = require('geohub');
```

```
module.exports = {  
  find: function( user, repo, file, options, callback ){  
    var key = [ user, repo, file ].join('/'),  
        type = 'Github';  
  }  
}
```

...

```
// controller/index.js
```

```
module.exports = {  
  getRepo: function( req, res ){  
    var _send = function( err, data ){  
      ...  
    }  
  }  
}
```

...

Esri-Leaflet

ArcGIS Services Plug-in



Leaflet

- Open source mapping library
- Pure JavaScript – 31kb
- Simple, easy to use, mobile friendly
- Many plug-ins

www.leafletjs.com

Leaflet Functionality

What's there?

- Draw map tiles
- Add pop-ups
- Read GeoJSON
- Add graphics (layers)
- Symbolize features
- Control layers
- Add other controls
- Plugins...

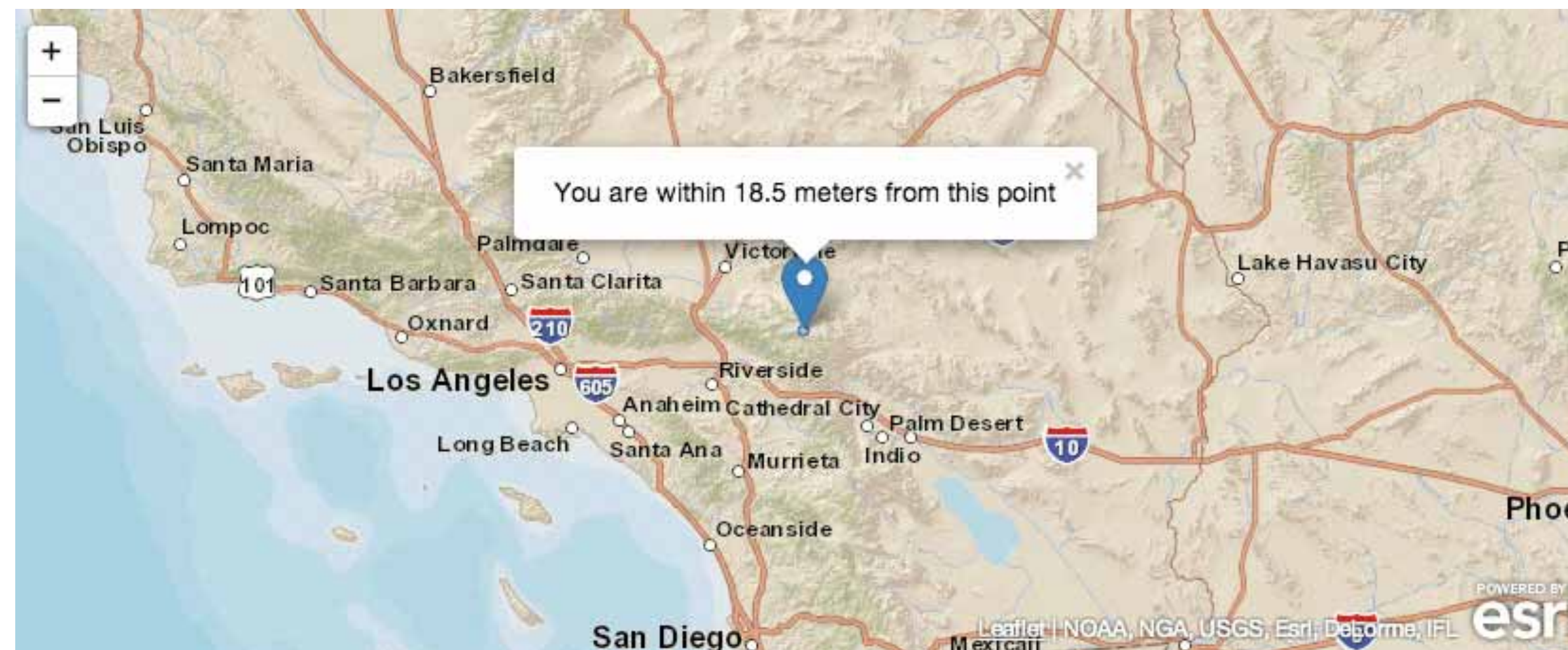
What's missing?

- Basemaps
- ArcGIS support
 - Basemaps
 - Feature Services
 - Other Services
- Widgets
- Webmaps
- Cloud storage

Esri-Leaflet

ArcGIS Online Services Plug-in

- Open source plug-in for ArcGIS Online services
- Extends L.class and namespace = L.esri.xxx
- Dependence on Terraformer



Esri-Leaflet: Getting Started

Reference `L.esri.xxx` Library

```
<!DOCTYPE html>
<html>
  <head>
    <title>Esri Leaflet</title>
    <link rel="stylesheet" href="/the/path/to/leaflet.css" />
  <style>
    html, body, #map { width: 100%; height: 100%; }
  </style>
  <script src="/the/path/to/leaflet.js"></script>
  <script src="/the/path/to/esri-leaflet.min.js"></script>
</head>
<body>
  <div id="map"></div>
  <script>
    var map = L.map('map');
    L.esri.basemapLayer("Streets").addTo(map);
    map.setView([38.97993, -104.9794], 12);
  </script>
</body>
</html>
```

Esri-Leaflet: ArcGIS Basemaps

```
L.esri.BasemapLayer = L.TileLayer.extend({...
```

```
// Load an ArcGIS basemap
```

```
var map = L.map('map').setView([37.75, -122.45], 12);
```

```
L.esri.basemapLayer("Topographic").addTo(map);
```

```
// Supported basemap types
```

```
//L.esri.basemapLayer("Streets").addTo(map);
```

```
//L.esri.basemapLayer("Oceans").addTo(map);
```

```
//
```

```
L.esri.basemapLayer("NationalGeographic").addTo(map)
```

```
;
```

```
//L.esri.basemapLayer("Gray").addTo(map);
```

```
//L.esri.basemapLayer("GrayLabels").addTo(map);
```

```
//L.esri.basemapLayer("Imagery").addTo(map);
```

```
//L.esri.basemapLayer("ImageryLabels").addTo(map);
```

Esri-Leaflet: ArcGIS FeatureServices

```
L.esri.FeatureLayer = L.GeoJSON.extend({...
```

```
    // Access ArcGIS FeatureService  
    var map = L.map('map').setView([45.52963623111275,  
    -122.67389774322508], 12);  
    L.esri.basemapLayer("Topographic").addTo(map);  
  
    var url = 'http://services.arcgis.com/  
    rOo16HdIMeOBI4Mb/arcgis/rest/services/stops/  
    FeatureServer/0'  
  
    L.esri.featureLayer(url);
```


Esri-Leaflet: Symbols

```
// Create FeatureLayer and define styles
L.esri.featureLayer(url, {
  style: function (feature) {
    return getStyle(feature);
  }).addTo(map);

function getStyle(feature) {
  var c,o = 0.5;
  switch (feature.properties.BIKEMODE) {
    case "Low traffic through street":
      c = "#007D7D";
      break;
    case "Bike boulevard":
      c = "#00FF3C";
      break;
    ...
  }
  return {color: c, opacity: o};
}
```

Esri-Leaflet: Popups

```
// Create FeatureLayer and bind to popup
L.esri.featureLayer(featureServiceUrl, {
    onEachFeature: createPopup
}).addTo(map);

// Define popup content - show all fields and values
function createPopup(geojson, layer) {
    if (geojson.properties) {
        var popupText = "<div style='max-height:200px;'>";
        for (prop in geojson.properties) {
            var val = geojson.properties[prop];
            if (val) {
                popupText += "<b>" + prop + "</b>: " + val + "<br>";
            }
        }
        popupText += "</div>";
        layer.bindPopup(popupText);
    }
}
```

Esri-Leaflet: DynamicMapLayer

```
// ArcGIS Server Dynamic Map Service - Hurricane Tracks
dynLayer = L.esri.dynamicMapLayer("http://tmservices1.esri.com/arcgis/rest/services/
LiveFeeds/Hurricane_Recent/MapServer", { layers:[0,1] });

// Identifying Dynamic Map Service Features
map.on("click", function(e) {
  dynLayer.identify(e.latlng, {
    layerDefs: {
      0: "STORMNAME='ANDREA'",
      1: "STORMNAME='ANDREA'"
    }
  }, function(data) {
    popupText = "<center><b>" +
      data.results[0].attributes.STORMNAME + "</b><br>" +
      data.results[0].attributes.STORMTYPE + "</center>";
    L.popup().setLatLng(e.latlng).setContent(popupText).openOn(map);
  }
});
});
```

Esri-Leaflet: ClusterFeatureLayer

esri-leaflet.js + clustered-feature-layer.js

```
// Reference cluster plug-in and esri feature layer
<script src="lib/markercluster/leaflet.markercluster.js"></script>

<script src="lib/esri-leaflet/extras/clustered-feature-layer.js"></
script>

// Create and add a new feature cluster layer
var fl = L.esri.clusteredFeatureLayer("http://services.arcgis.com/
rOo16HdIMEOBI4Mb/arcgis/rest/services/stops/FeatureServer/0", {
  cluster: new L.MarkerClusterGroup() ,
  onEachMarker: function(geojson, marker) {
    marker.bindPopup(" <h3>" +
      geojson.properties.stop_name+" </h3><p>
      Stop ID: "+geojson.properties.stop_id+" </p><p>"
+geojson.properties.stop_desc+" </p>" )
    }
  }
}).addTo(map);
```

Esri-Leaflet: FeatureService Query

esri-leaflet.js + geoservices.js

```
// Access feature service directly and query (geoservices.js)
var fs = new GeoServices.FeatureService ({url:featureServiceUrl}, function (err, results) {
  var queryOptions = document.getElementById("query");
  var query = queryOptions.text;
  var queryEnvelope =
    JSON.stringify(L.esri.Util.boundsToExtent(map.getBounds()));
  // Build query parameters
  var params = {
    f:"json", where: query,
    geometry: queryEnvelope,
    spatialRel: "esriSpatialRelIntersects",
    returnGeometry:true, outSR: 4326, outFields:"*"
  };
  // Query the feature service
  fs.query(params, function (err, results) {
addFeaturesToMap(results); // Manual
  })
}
```

Esri-Leaflet: Geocoding

esri-leaflet.js + geoservices.js

```
// Reference geoservices.js
<script src="lib/geoservices/geoservices.js"></script>
...
var GeoServices = new Geoservices.Geoservices({});
var options = {
    text:searchString,
    outFields: "Loc_name,Place_addr",
    bbox: mapBounds }

// Add geocodes to map
GeoServices.geocode(options, function (err,result) {
    for (var i = 0; i < result.locations.length; i++) {
        var place = result.locations[i];
        var pt = new L.LatLng(place.feature.geometry.y, place.feature.geometry.x);
        var marker = L.marker(pt).bindPopup(place.name + "<br>" +
place.feature.attributes.Place_addr);
        layerPlaces.addLayer(marker);
    }
}
```

Esri-Leaflet: Holy Grail?

- Widgets
- Symbols
- Renderers
- Editing
- Geometry operations and types
- Webmaps
- Accessing all ArcGIS Services
-

OAuth 2.0

Authenticating ArcGIS Apps



What is OAuth 2.0?

- OAuth 2.0 is a framework for allowing apps to securely access data on behalf of users
- A more secure alternative to asking for a user's password

Types of OAuth

- Client-side
 - Best for browser-based applications
 - No `client_secret` required, so it's safe to use in Javascript apps
- Server-side
 - Requires `client_secret`, so cannot be used from a Javascript app without a proxy

Simple OAuth 2.0 Example

- Use the “implicit” grant type to generate an access token directly in the browser
- No server-side code required
- Browser makes API calls directly to secure services with the access token

Step 1: ArcGIS Developer Subscription

- Sign up for a subscription
- Register your apps
- Enable for OAuth

Registering your apps

Credentials

ArcGIS for Developers PLATFORM FEATURES PLANS DOCUMENTATION SUPPORT razzberry

Applications Hosted Data GIS Tools Usage Summary Licensing Downloads Account Settings

< esri-leaflet VIEW IN ARCGIS ONLINE

- Application Details
- API Access
- Redirect URIs
- Usage Summary

DELETE APP

OAuth Credentials

Client ID: sSR3ZF4K3GILBqsl

Client Secret: [REDACTED]

Redirect URLs

< esri-leaflet VIEW IN ARCGIS ONLINE

- Application Details
- API Access
- Redirect URIs
- Usage Summary

DELETE APP

http://esri-leaflet.dev/oauth/callback.html

http://esri.github.io/esri-leaflet/oauth/callback.html

http://your-site.com/redirect-uri

Adding redirect URIs to your application will allow users with ArcGIS online subscriptions to login to your application via OAuth 2. If you are building an application for users outside of ArcGIS online you dont need redirect uris.

UPDATE REDIRECT URIS

Step 2: Create a login page

```
https://www.arcgis.com/sharing/oauth2/authorize?  
&client_id=YOUR_CLIENT_ID&response_type=token  
&redirect_uri=REDIRECT_URI
```

```
var clientID = 's5R3ZF4K3GILBqsI';  
var url = "https://www.arcgis.com/sharing/oauth2/authorize?client_id=";  
var uri = encodeURIComponent(window.location.origin)+"%2Foauth%2Fcallback.html";  
  
function startAGOOAuth() {  
    window.open(url + clientID  
        &redirect_uri=" + uri,  
        +"&response_type=token&expiration=20160  
        "oauth-window", "height=400,width=600");  
}
```

The authorization page

OAuth Test wants to access your account information

Sign In esri

Username

Password

[SIGN IN](#) [CANCEL](#)

[Forgot Username or Password?](#)

Step 3: Acquire a token

On success, user is redirected back to your site with the access token in the fragment

```
https://example.com/auth#token=ACCESS_TOKEN
```

On error, user is redirected back to your site with error code

```
https://example.com/auth#error=access_denied
```


Using the Access Token

- Parse out the access token
- Make API requests using the token

```
var accessToken;
```

```
window.oauthCallback = function(token) {  
    accessToken = token;  
}
```

```
// Access services with token
```

```
L.esri.get("http://route.arcgis.com/arcgis/rest/services/  
World/Route/NAserver/Route_World/solve", {  
    token: accessToken,  
    stops: s.lng+", "+s.lat+"; "+e.lng+", "+e.lat,  
    outputLines: 'esriNAOutputLineTrueShape'  
}, function(response){
```

```
...
```

esri-leaflet/directions

Licensing

ArcGIS Developer Subscriptions



Licensing

- Free ArcGIS Developer Subscription
 - Testing and development
 - Public deployments (non-commercial)
 - 50 credits
- Paid ArcGIS Developer or ArcGIS Organization Subscription
 - Private deployments
 - Commercial deployments (generates revenue)

esri.github.io

The screenshot shows the homepage of esri.github.io. At the top, there's a navigation bar with the text "Esri GitHub | Open Source" and the URL "esri.github.io". Below this is a large banner image of a satellite map of a coastal area. The banner contains the text "Esri is on GitHub!" followed by "We're excited about helping developers build and share software. Browse our open source code and get started with our powerful ArcGIS platform." and a "BROWSE ON GITHUB" button. Below the banner, there's a search bar with the text "Filter by language or keyword:". Below the search bar, there's a grid of project cards. Each card displays the project name, language, description, and star/fork counts.

Esri is on GitHub!

We're excited about helping developers build and share software. Browse our open source code and get started with our powerful ArcGIS platform.

[BROWSE ON GITHUB](#)

Need an [ArcGIS subscription](#)? Start developing today for [free](#).

Filter by language or keyword:

- esri-leaflet**
JavaScript
A lightweight set of tools for working with ArcGIS services with Leaflet.
🔼 83 ★ 133
- terraformer**
JavaScript
A geometric toolkit for dealing with geometry, geography, formats, and building geo databases.
🔼 41 ★ 124
- arcgis-viewer-flex**
ActionScript
Source code for ArcGIS Viewer for Flex – a great application framework for web applications.
🔼 116 ★ 119
- geometry-api-java**
Java
The Esri Geometry API for Java enables developers to write custom applications for analysis of spatial data. This API is used in the Esri GIS Tools for Hadoop and other 3rd-party data processing solutions.
🔼 28 ★ 58
- gis-tools-for-hadoop**
Python
The GIS Tools for Hadoop are a collection of GIS tools for spatial analysis of big data.
🔼 28 ★ 57
- geoportal-server**
Java
Geoportal Server is a standards-based, open source product that enables discovery and use of geospatial resources including data and services.
🔼 51 ★ 44
- geojson-utils**
JavaScript

- github.com/chelm
- github.com/driskull

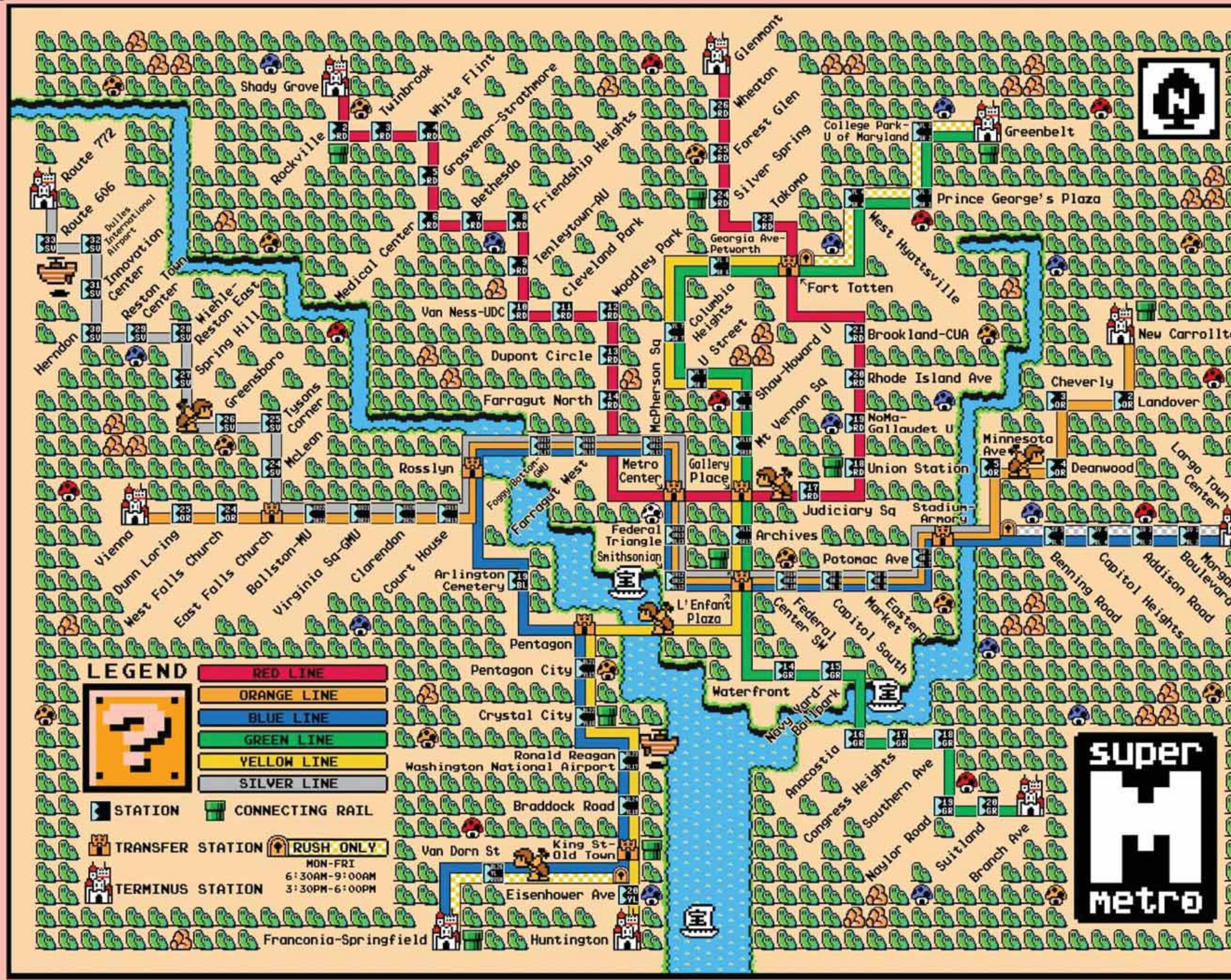
Other libraries

Andrew Turner

@ajturner

andrew@esri.com

github.com/ajturner





Understanding our world.