

**An Open-Source Ecosystem of Next-Gen Web Technologies**

**for Hydrological Education, Research, and Operations**

**HydroSuite**

**Ibrahim Demir, Carlos Ramirez, Yusuf Sermet**



**HILab**



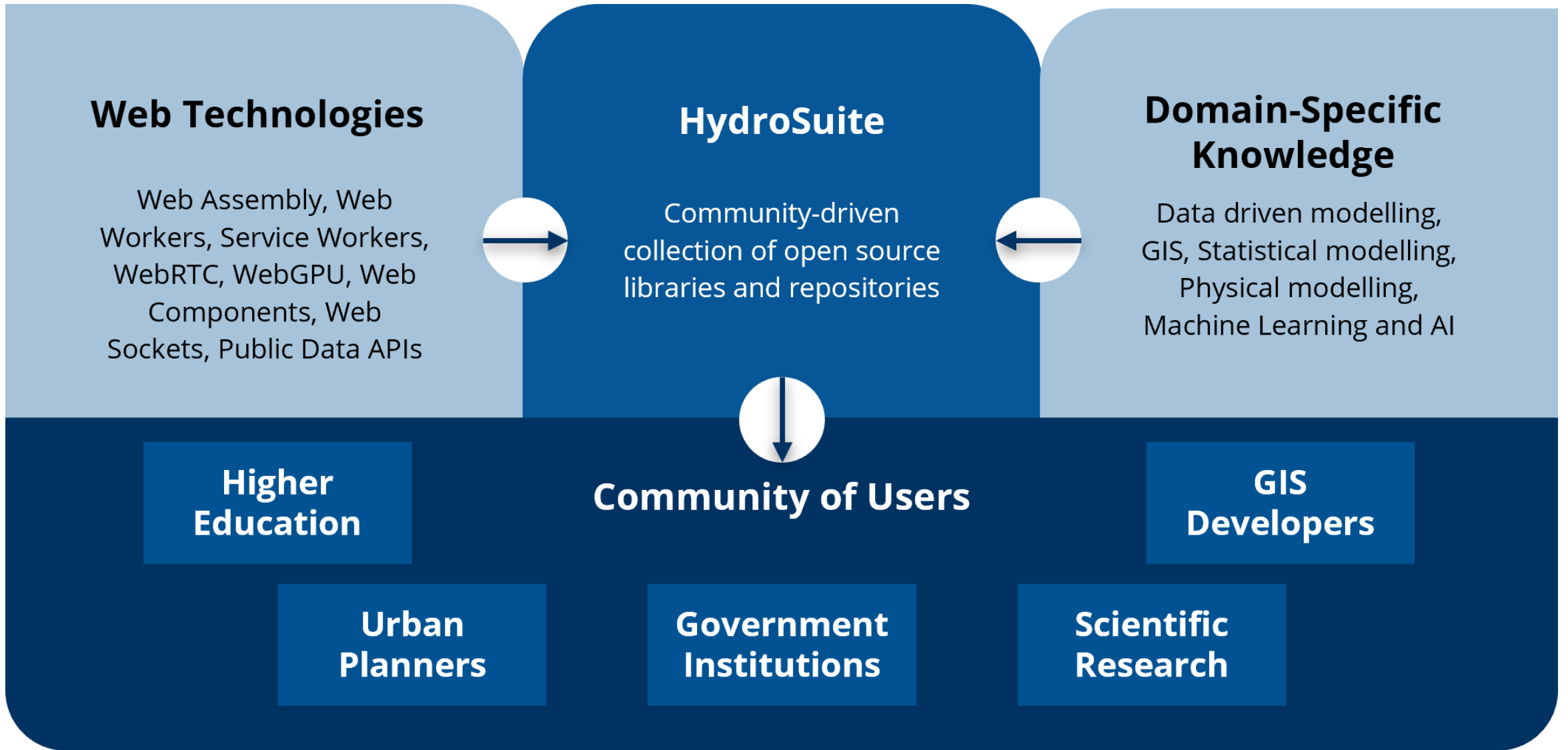
## HydroSuite – Collection of Open Source Tools, Libraries, Repositories

DATA	COMPUTING	COMMUNICATION	COMMUNITY PORTALS
Flood-ML	HydroLang	RasterJS	EarthAI Hub
Flood Event DOM	HydroLang-ML	Hydro3DJS	HLM Web
IS Ontology	HydroLang-BMI	Instant Expert	HydroLSTM
WaterBench	HydroCompute	GeospatialVR	HydroLang Models
IowaRain	HydroRTC	Watershed Delineation	Training Repos

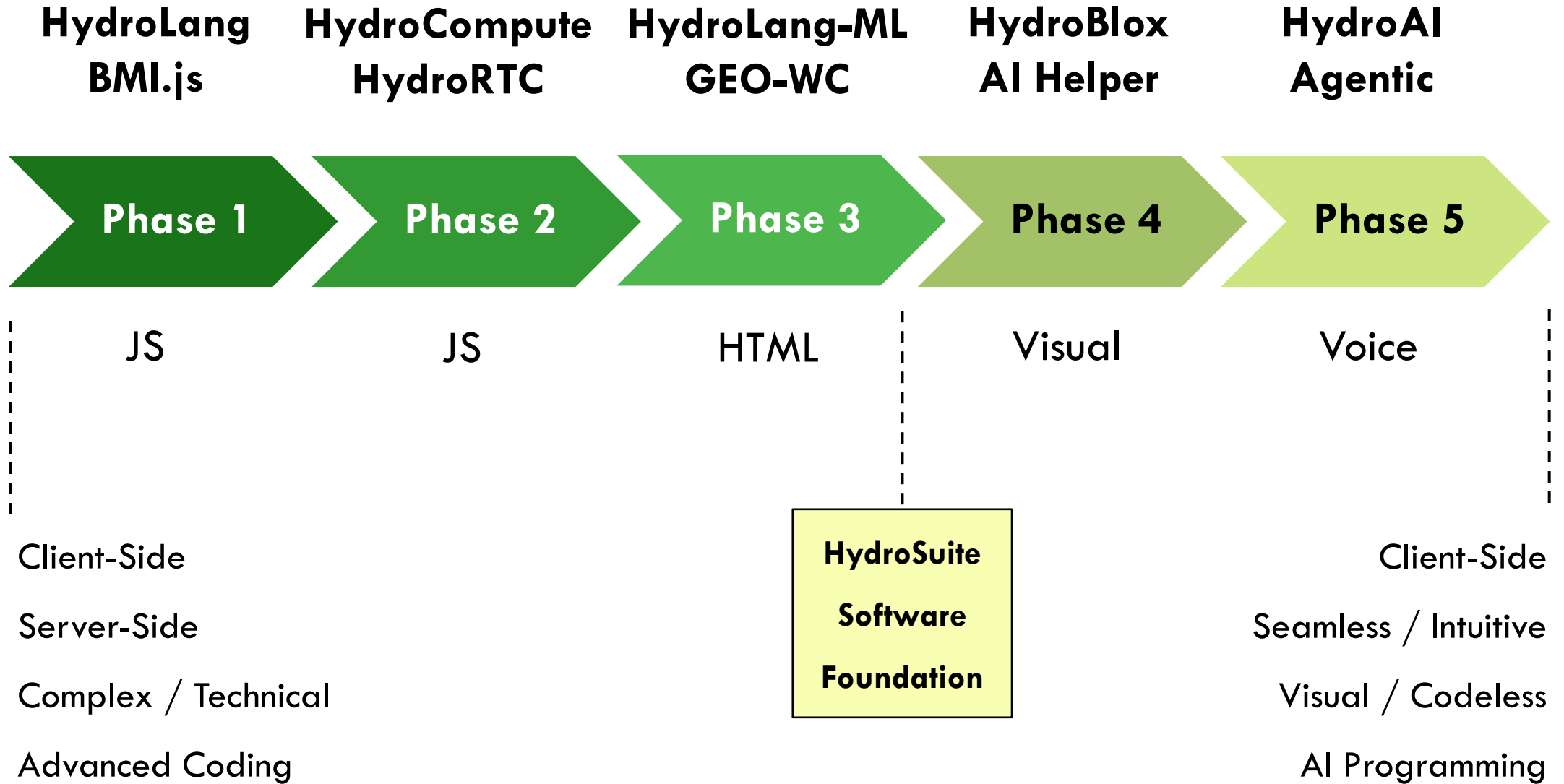
140+ software tools and libraries



Open-Source Ecosystem



# HydroSuite Roadmap



# BMI-JS

- ✓ The Basic Model Interface (BMI) is a standardized set of functions that allows coupling of models to models and models to data.
- ✓ The BMI is developed by the Community Surface Dynamics Modeling System (CSDMS).

<https://github.com/uihilab/bmi-js>

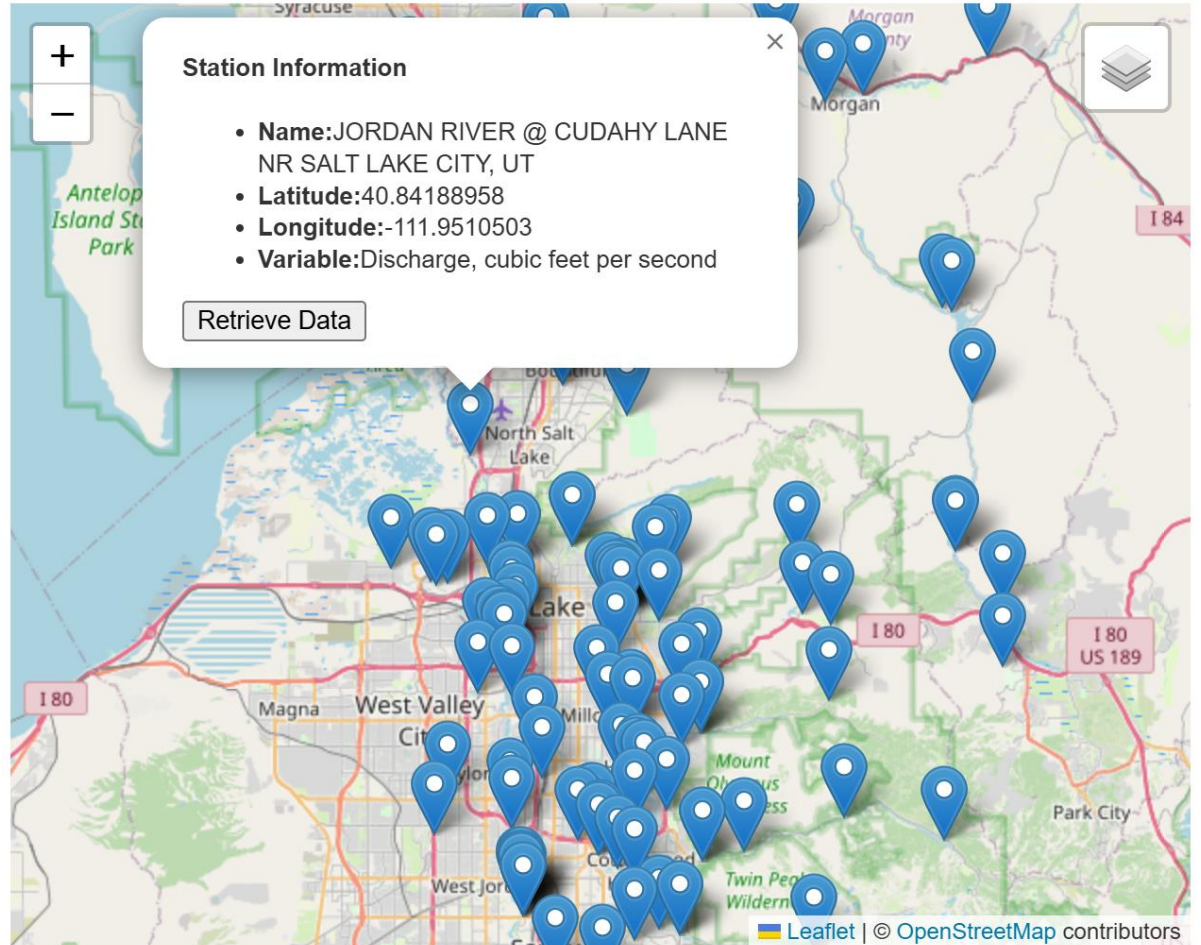
<https://github.com/uihilab/bmi-example-js>

BMI languages		
Language	Specification	Example implementation
C	<a href="#">bmi-c</a>	<a href="#">bmi-example-c</a>
C++	<a href="#">bmi-cxx</a>	<a href="#">bmi-example-cxx</a>
Fortran	<a href="#">bmi-fortran</a>	<a href="#">bmi-example-fortran</a>
Java	<a href="#">bmi-java</a>	<a href="#">bmi-example-java</a>
Python	<a href="#">bmi-python</a>	<a href="#">bmi-example-python</a>

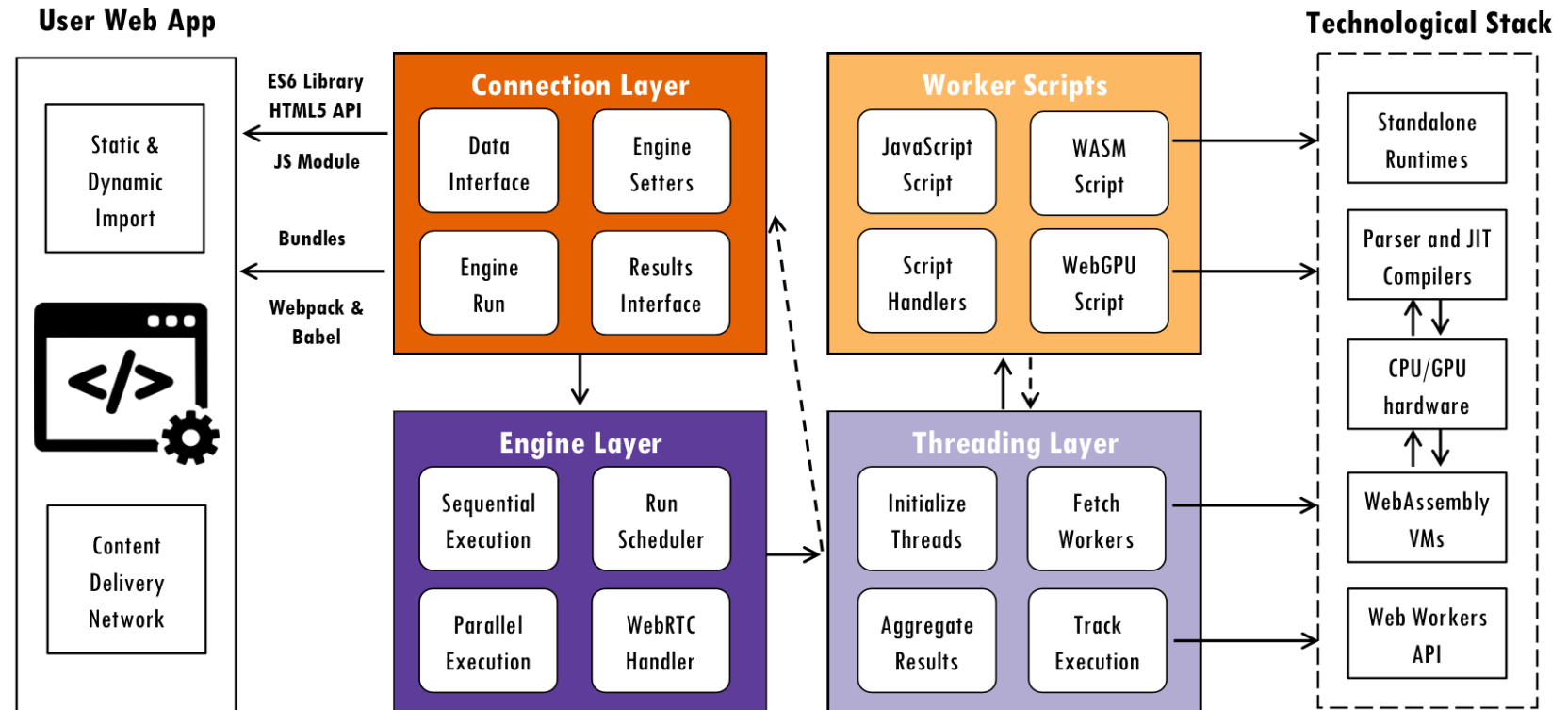
Community-contributed BMI languages		
Language	Specification	Example implementation
Javascript	<a href="#">bmi-js</a>	<a href="#">bmi-example-js</a>

- ✓ A web-based open-source programming framework for hydrological and environmental analyses
- ✓ Entirely client-side operation, no server-side dependency
- ✓ Scalable and upgradable
- ✓ Easy to use and modify
- ✓ Community oriented modular design



# HydroCompute

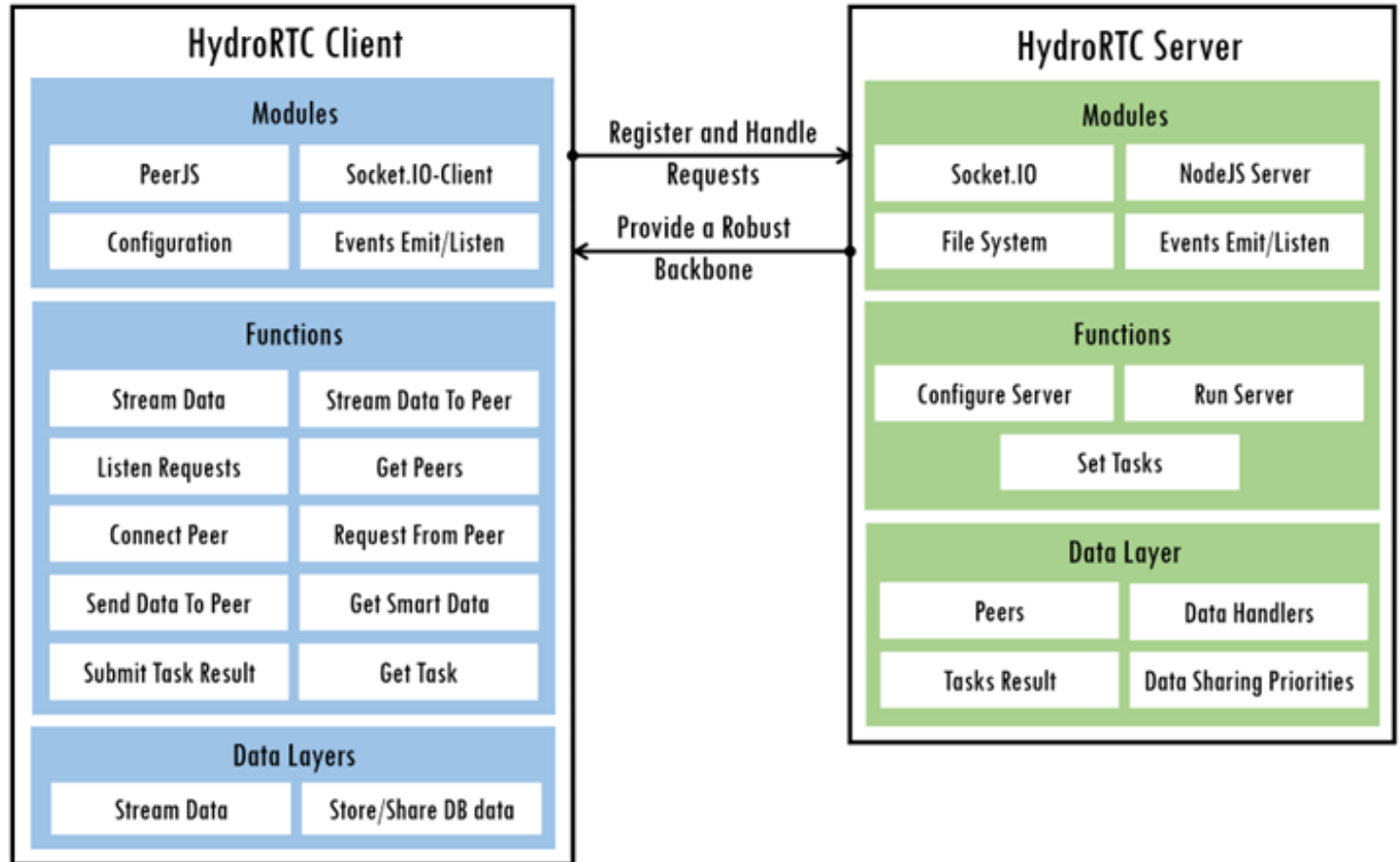
- ✓ Multi-CPU and GPU Based Parallel Computing Library for Hydrology
- ✓ Community oriented modular design for expansion
  - ✓ Special GitHub repo
  - ✓ UI Hackathon



<https://github.com/uihilab/HydroCompute>

# HydroRTC

- ✓ Real-time communication library for decentralized server-side applications
- ✓ Enables data streaming, real time data sharing, and hydrologic data interpretation
- ✓ Can be used for deployment of largescale applications

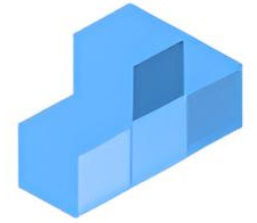


<https://github.com/uihilab/HydroRTC>

# HydroLang ML - Markup Language

Web Component Library for Hydrological and Environmental Sciences

<https://github.com/uihilab/HydroLang-ML>



**HL-ML**

```
<data-mod method="transform">
  <parameters-here input="usgs_data" save="value" output="cleaned_usgs_data1" ></parameters-here>
  <arguments-here type="ARR" keep='["datetime", "value"]'></arguments-here>
</data-mod>
```

```
<hydrolang-ml>
  <analyze-mod method="someFunctionName">
    <parameters-here someAttr="Some Attribute"></parameters-here>
    <arguments-here somArgs="someArguments"></arguments-here>
    <dataset-here>[1,2,3,4]</dataset-here>
  </analyze-mod>
</hydrolang-ml>
```

```
<visualize-mod method="draw">
  <parameters-here type="json" input="usgs_data"></parameters-here>
</visualize-mod>
```

# Geo-WC - Custom Web Components

Custom web components for geoscience agencies

<https://github.com/uihilab/Geo-WC>

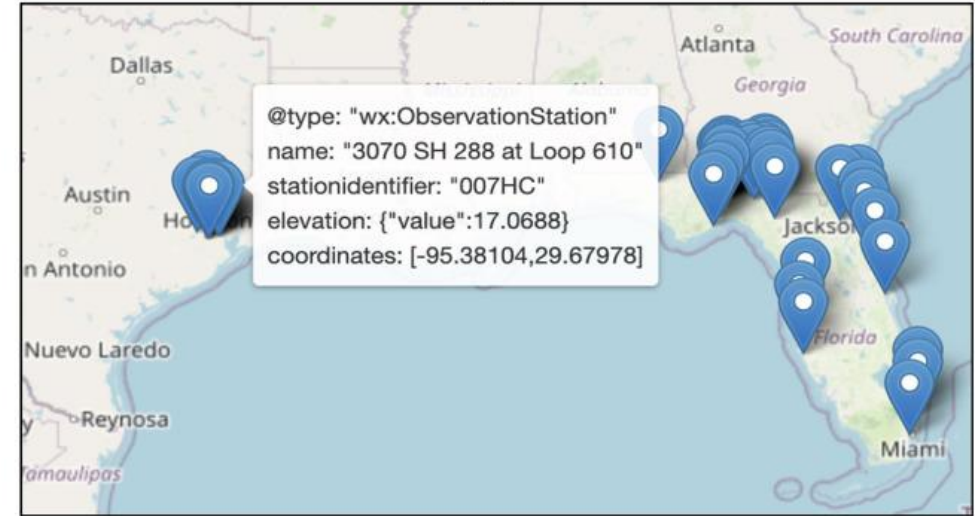
```
<usgs-ml service="instant-values">  
  <api-args startDate="2023-11-01" timeInterval="1d"></api-args>  
  <geoweb-graph> </geoweb-graph>  
</usgs-ml>
```



```
<usgs-ml service="instant-values">  
  <api-args raw="true" site="USGS:02056000" startDt="2023-11-01"  
    endDt="2023-11-01" parameterCd="00060"></api-args>  
  <api-args startDate="2023-11-01" timeInterval="1d"></api-args>  
</usgs-ml>
```

```
<epa-ml service="station">  
  <api-args raw="true" bbox="-96.789,40.580,-89.802,43.564"  
    characteristicName="Nitrogen" countycode="US:24:033"></api-args>  
  <geoweb-map></geoweb-map>  
</epa-ml>
```

```
<nws-ml service="stations">  
  <api-args raw="true" limit="100"></api-args>  
  <geoweb-map></geoweb-map>  
</nws-ml>
```



```
<fema-ml service="disaster-declarations">  
  <api-args raw="true" fipsStateCode="19"></api-args>  
  <geoweb-table> </geoweb-table>  
</fema-ml>
```

declarationtitle	placecode
0419 FIRE	99017
1707 ADKINS AVE FIRE	99005
1ST CANYON FIRE	99007
259 FIRE	99049
30TH AVENUE FIRE	99021
316 FIRE	99007
335 FIRE	99253
36 PIT FIRE	99005
36TH AVENUE FIRE	99021
412 FIRE COMPLEX	99007

Showing 1 to 10 of 1,000 entries

# HydroBlox

- ✓ Block-based client-side visual programming library
- ✓ Integrating visual queues and workflow generation using HydroSuite libraries
- ✓ AI Assistant for block generation and integration

The screenshot displays the HydroBlox interface, which is a block-based visual programming environment. The main workspace shows a workflow diagram with 19 nodes connected by blue dashed arrows. The nodes include data sources like 'RAW\_STATION\_BBOX' and 'STATIONS\_GEOJSON', processing blocks like 'STATION\_DATA', 'FORMATTED\_TS', and 'ENSEMBLE\_PREDIC...', and visualization blocks like 'Map'. A 'Data Store' panel on the right lists 42 datasets, including 'PRECIP\_BIAS\_CORRECT ED', 'STATIONS\_SPI', 'DATASET\_GRID', 'FORMATTED\_TS', 'CLEAN\_TS', and 'EXTRACT\_EXAMPLE'. The interface also features a search bar, navigation tabs (Core, Helper, Triggers, Session, Advanced), and a footer with the Hydroinformatics Lab logo and contact information.

# HydroBlox

Search or describe a workflow...

Core Helper Triggers Session Advanced

### DATA SOURCES

aemet	aorc
bmkg	chirps
clearcreek	cma
cpctec	drought
dwd	eauk
ecmwf	epa
fao	fema
hisCentral	hrrr



19 Nodes

[Play] [Pause] [Stop]

### Data Store

Datasets (42) Workflows Packages

Search...

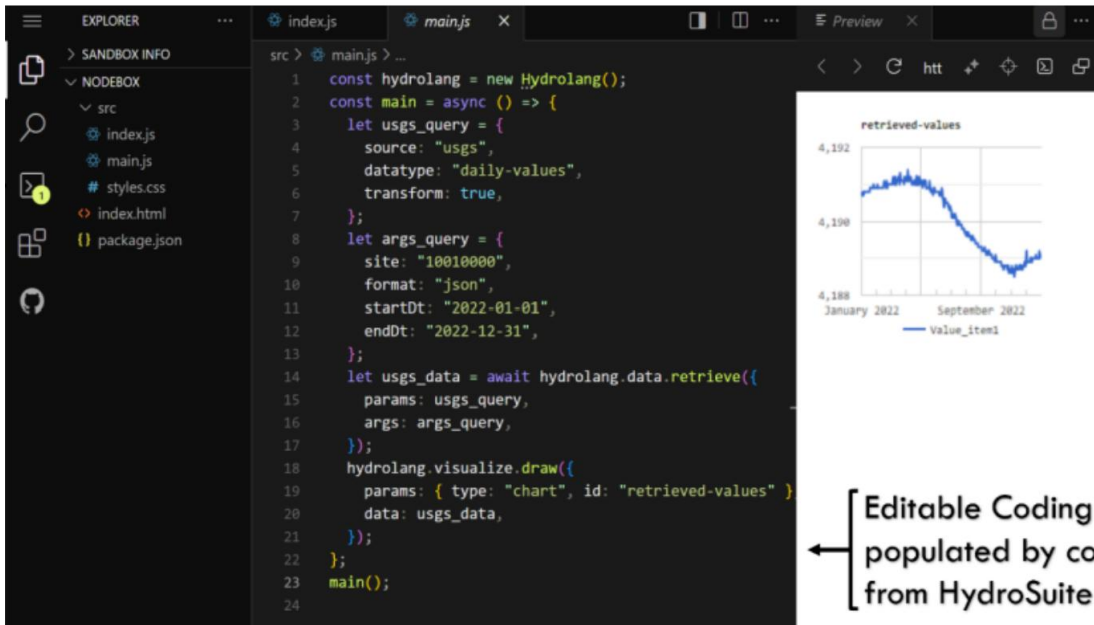
Select all Download all as ZIP Delete all

- PRECIP\_BIAS\_CORRECT ED
  - JSON
  - completed · 5/24/2026, 10:39:32 PM · 4.25 MB
- STATIONS\_SPI
  - JSON
  - completed · 5/24/2026, 10:38:44 PM · 181.1 KB
- DATASET\_GRID
  - JSON
  - completed · 5/24/2026, 10:34:22 PM · 1.3 KB
- FORMATTED\_TS
  - TIMESERIES
  - completed · 5/24/2026, 10:34:20 PM · 1.67 MB
- CLEAN\_TS
  - ARRAY
  - completed · 5/24/2026, 10:33:33 PM · 39.25 MB
- EXTRACT\_EXAMPLE

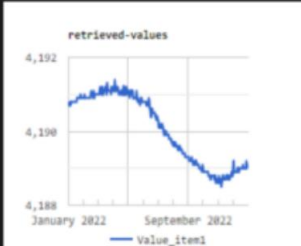
Undo Redo Select Auto Layout Minimize Expand 62%

# HydroSuite AI Helper

- ✓ Facilitating Hydrological Research with LLM-Driven Code Assistance



```
1 const hydroLang = new HydroLang();
2 const main = async () => {
3   let usgs_query = {
4     source: "usgs",
5     datatype: "daily-values",
6     transform: true,
7   };
8   let args_query = {
9     site: "10010000",
10    format: "json",
11    startDt: "2022-01-01",
12    endDt: "2022-12-31",
13  };
14  let usgs_data = await hydroLang.data.retrieve({
15    params: usgs_query,
16    args: args_query,
17  });
18  hydroLang.visualize.draw({
19    params: { type: "chart", id: "retrieved-values" },
20    data: usgs_data,
21  });
22 };
23 main();
24
```



Retrieved Daily Values from USGS for site 10010000

Editable Coding environment populated by code generated from HydroSuite AI

Simply select the type of task, enter your query or request, and let HydroSuite AI do the rest!

HydroLang HydroRTC HydroCompute

## Task 1

Select Task Type - HydroLang

Code Execution

Input - HydroLang

Give me a code to fetch and plot USGS Daily values for site 10010000 specifically for the whole year of 2022.

Run Task 1

**Disclaimer:** The generated code and responses may contain errors. Please review and validate carefully. The HydroSuite application is currently experimental and subject to further improvements.

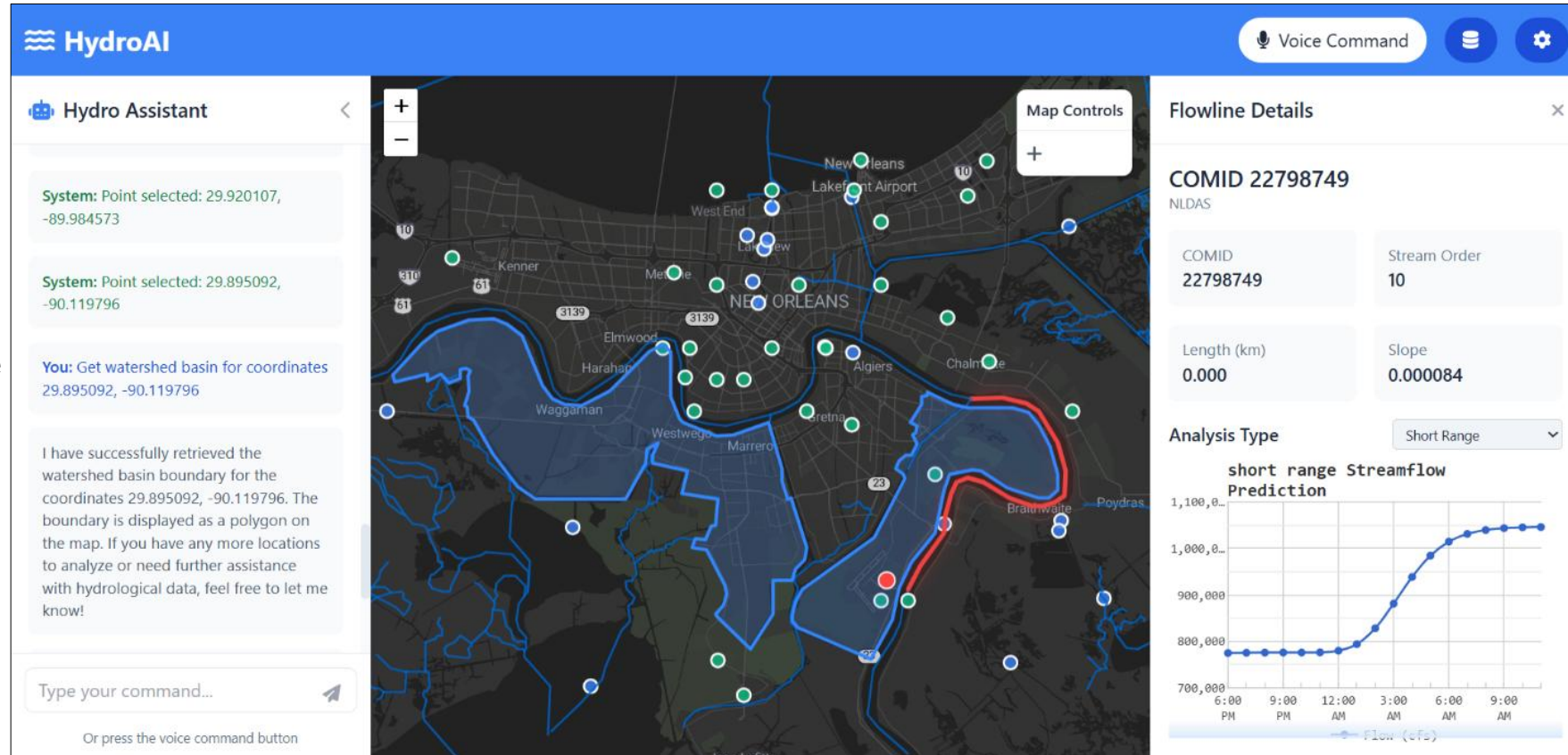
To fetch daily USGS values for site `10010000` for the year 2022 and plot them using JavaScript, you can use the HydroLang library and additional helper functions for data plotting. Here is the code following what was retrieved:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>USGS Daily Values Plot</title>
  <script src="https://cdn.jsdelivr.net/npm/hydroLang@1.1.7/hydroLang.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/plotly.js-dist-min"></script>
</head>
<body>
  <h1>USGS Daily Values for Site 10010000 - 2022</h1>
  <div id="plot"></div>

  <script>
    document.addEventListener('DOMContentLoaded', async () => {
```

# HydroAI Agentic

- ✓ Voice-enabled intelligent workflow development interface
- ✓ Generates HydroSuite workflows for hydrological analysis on a map interface
- ✓ Ability to quickly analyze and visualize data through natural language
- ✓ Scalable and integrated infrastructure



# HydroAI Agentic

## HydroAI

Voice Command

### Hydro Assistant

**System:** Point selected: 29.920107, -89.984573

**System:** Point selected: 29.895092, -90.119796

**You:** Get watershed basin for coordinates 29.895092, -90.119796

I have successfully retrieved the watershed basin boundary for the coordinates 29.895092, -90.119796. The boundary is displayed as a polygon on the map. If you have any more locations to analyze or need further assistance with hydrological data, feel free to let me know!

Type your command...

Or press the voice command button

Map Controls

### Flowline Details

COMID 22798749  
NLDAS

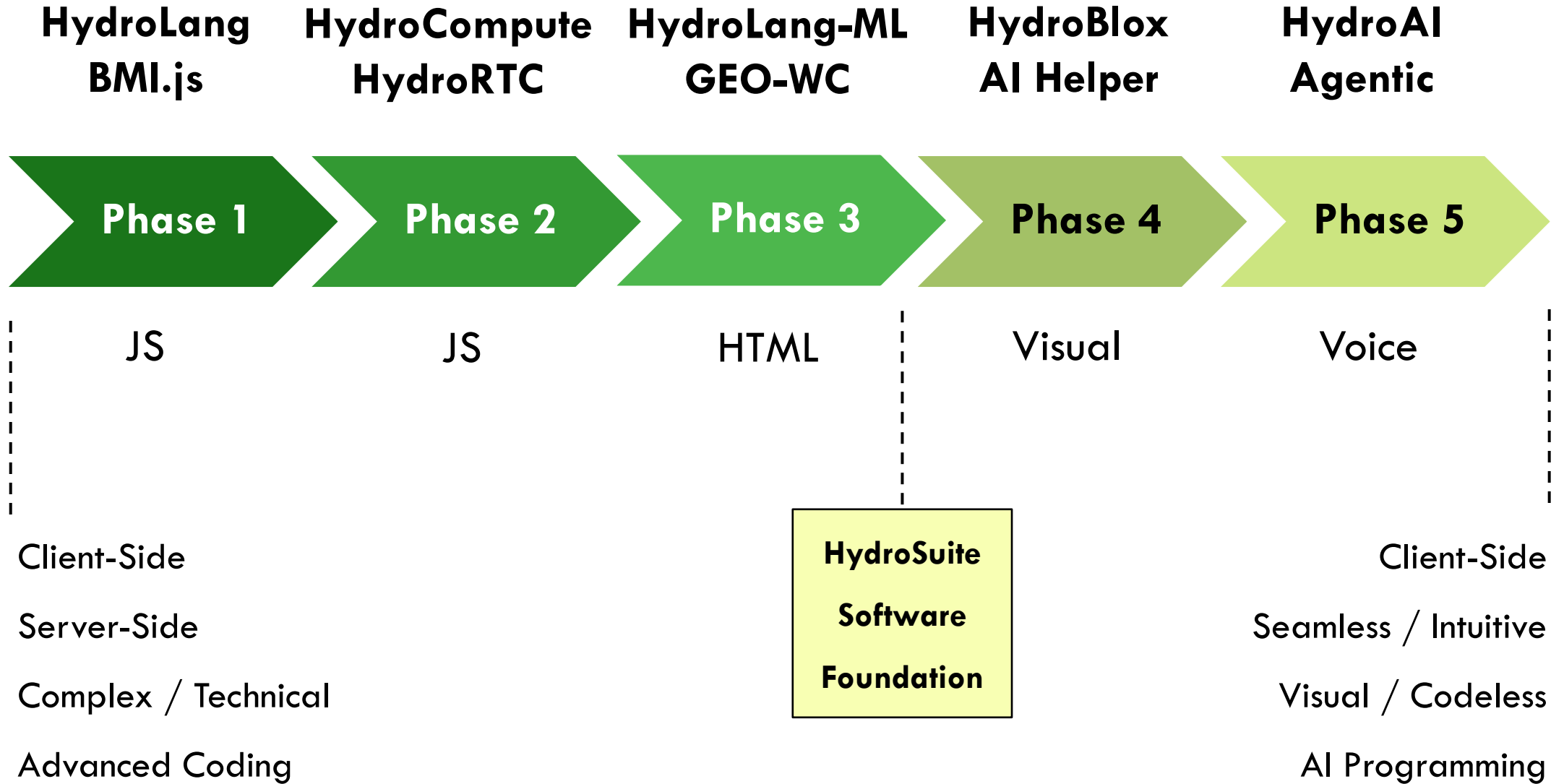
COMID	Stream Order
22798749	10
Length (km)	Slope
0.000	0.000084

Analysis Type: Short Range

#### short range Streamflow Prediction

Time	Flow (cfs)
6:00 PM	780,000
9:00 PM	780,000
12:00 AM	780,000
3:00 AM	850,000
6:00 AM	1,050,000
9:00 AM	1,080,000

# HydroSuite Roadmap



# Thanks

<https://hydroinformatics.tulane.edu/lab/hydroblox>



HydroLang



HydroRTC



HydroCompute



**HILab**



# HydroBlox – AI-Assisted Web-Based Visual Computing

## Platform Access



[hydroinformatics.tulane.edu/lab/hydroblox](https://hydroinformatics.tulane.edu/lab/hydroblox)

## Survey



[forms.gle/uYJr9QL74NKmJgbg6](https://forms.gle/uYJr9QL74NKmJgbg6)