

HACKATHON 2022

WAY@TiDB GIS

Team Intro – Leonardo

“In 16th century, one genius transformed mapping from art to science;

Today, we are going to transform mapping from data to digital economy.”

// Author: Leonardo, Date: Oct.23,2022

// Project: WAY - where are you

namespace TiDB;

new GIS(spatial data);



Fig. Leonardo transformed mapping from art to science

Team Members

- Team Lead:

Yves Zhang [@yiwen92](#) from CN

- Team Members:

Daniël van Eeden [@dveeden](#) from NL

Elwyn Wen [@madwyn](#) based in UK

Mattias Jonsson [@mjonss](#) based in NL



- Project Advisor:

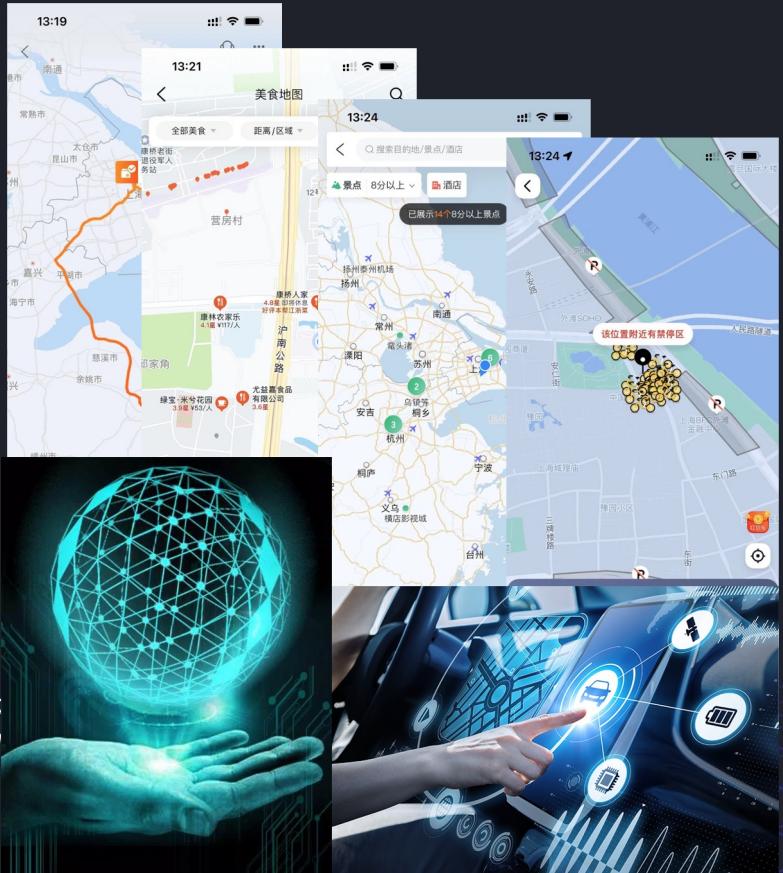
Zhidong Chen [@sillydong](#) from CN



Why we do this?

Why GIS?

- **Spatial data matters!**
- Widely used in daily life 衣食住行 样样不离！
- IOT/Mobile(14 Billion, 90%, 80%*) 设备增长 数据爆炸
- Promising future 自动驾驶/车联网，元宇宙/全息投影



* Franklin, Carl and Paula Hane, "An introduction to GIS: linking maps to c

** Cisco's Internet Business Solutions Group (IBSG), "The Internet of Thin

Why TiDB?

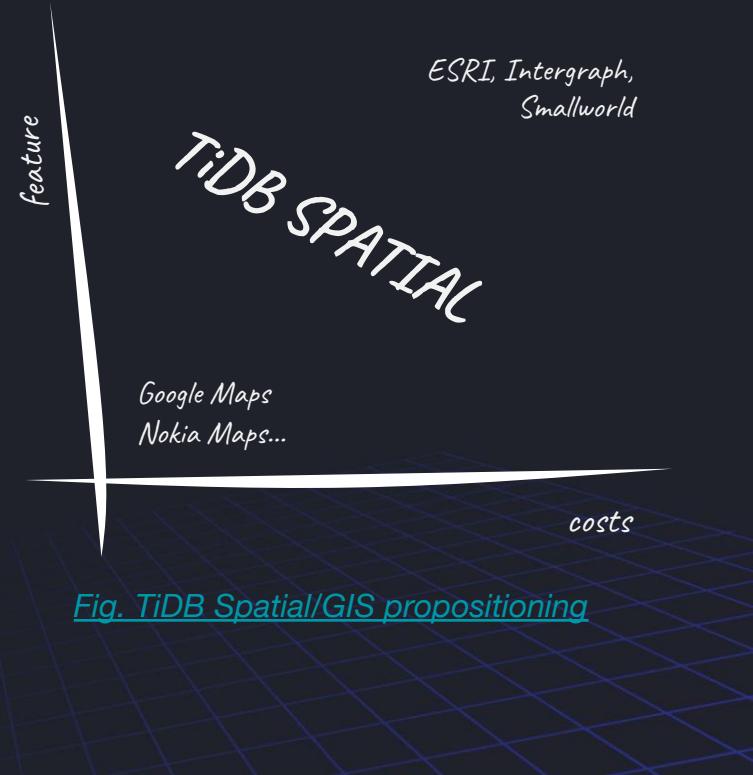
- **TiDB has solid foundations!**
- Widely-used install-base (Scenarios 场景)
- Business run on TiDB (Transaction data 数据)
- Distributed + Scalability + HTAP (Technique 技术)



Fig. TiDB strong foundations of scenario, biz and tech

GIS + TiDB

- Huge potential for TiDB!
- Simplify architecture to provide business agility
- Improve usability, visualization and interactivity
- Extend TiDB capability to adapt new scenarios
- Explore collaborative and innovative ecosystem
- 简化、敏捷、融合、创新



What we do?

Framework

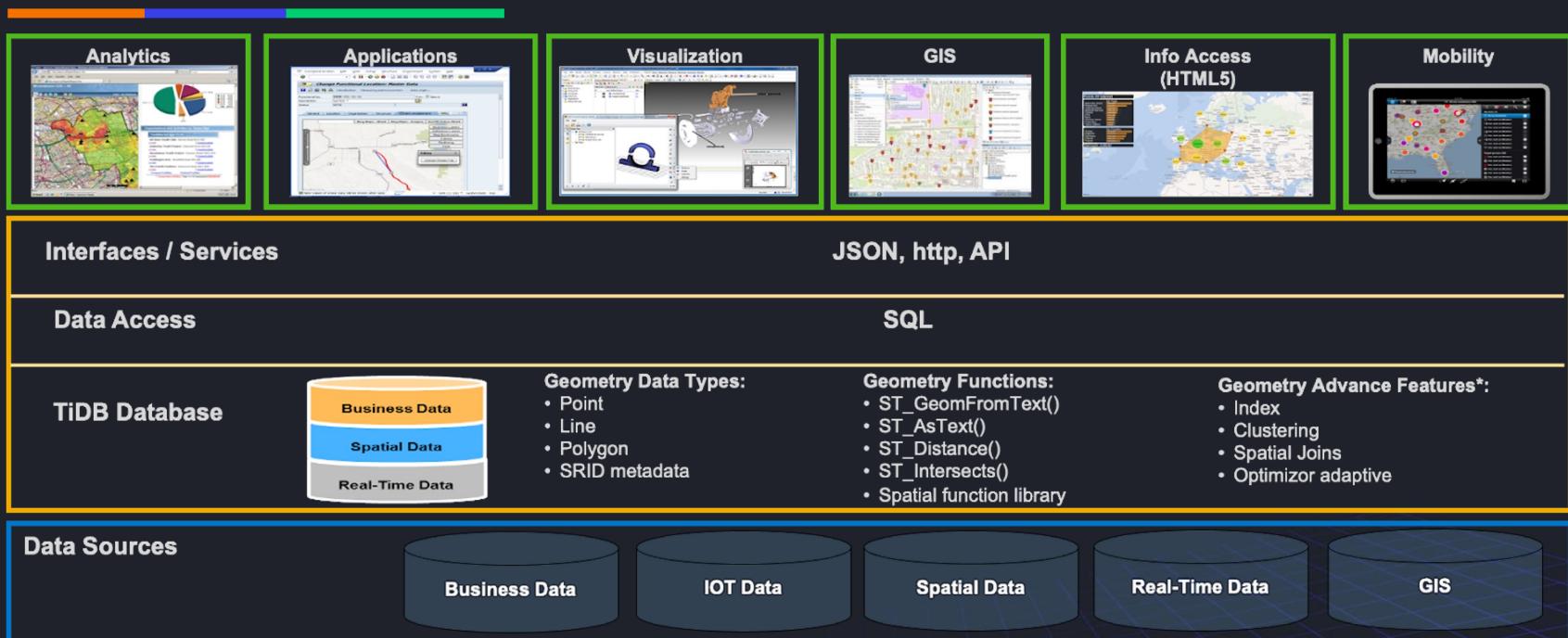


Fig. Whole picture of TiDB Spatial/GIS

Architecture for Demo - shared bike scenario

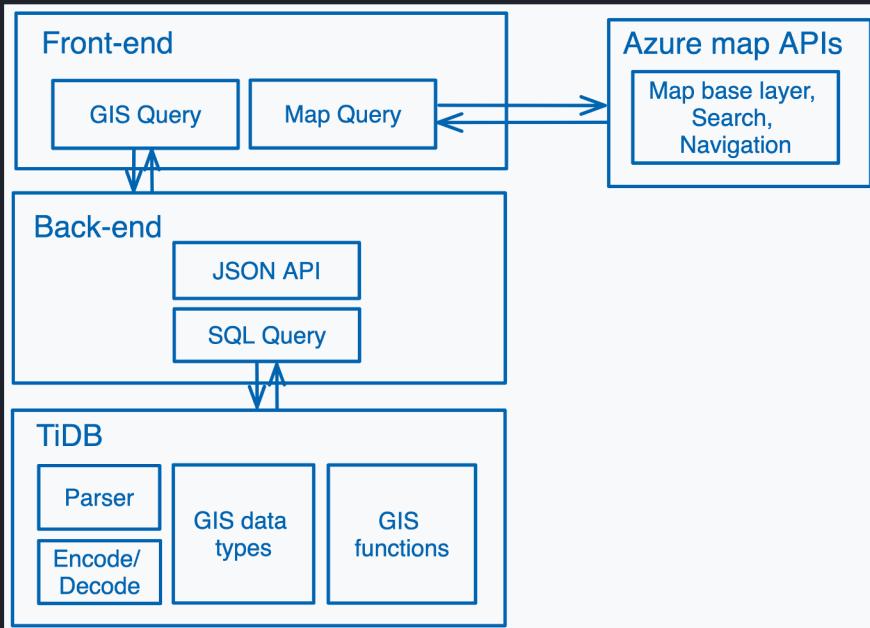


Fig. Architecture for shared bike scenario

T I D B H A C K A T H O N 2 0 2 2

Screenshot of the GitHub organization `Hackathon-2022-GIS` showing four repositories:

- webapi** (Public): Small web service for exposing new GIS feature in TiDB. Last updated 1 hour ago.
- way** (Private): Last updated 4 hours ago.
- tidb** (Public): Apache-2.0 license. Last updated yesterday.
- tools** (Public): Python language. Last updated 3 days ago.

<https://github.com/orgs/Hackathon-2022-GIS/repositories>

Support GIS Data Type

```
— insert and select gis data from tidb  
— show point, polygon in map
```

```
INSERT INTO poi(g) VALUES(ST_GeomFromText('POINT(121.48466 31.23530)'));  
INSERT INTO poi(g) VALUES(ST_GeomFromText('POLYGON ((4.83375 52.3311, 5.04079 52.37569, 4.83113 52.41397, 4.83375 52.3311))'));  
SELECT * FROM poi ORDER BY id DESC LIMIT 2;
```

The screenshot displays a database interface with a SQL editor and two maps.

SQL Editor:

```
SELECT * FROM poi ORDER BY id DESC LIMIT 2
```

Maps:

- Left Map:** A map of the Amsterdam area in the Netherlands. It shows the city's intricate canal network and surrounding regions. A blue polygon is drawn over the central part of the city, representing the geometry from the database query.
- Right Map:** A map of Shanghai, China, showing various districts like Jiading, Baoshan, Putuo, Yangpu, Pudong, Xuhui, Minhang, and Qingpu. Numerous blue points are scattered across the map, representing individual data entries from the database.



Support GIS Function

- ST_GeomFromText()
- ST_AsText()
- ST_Distance()
- ST_Intersects()

3.4.1. Parallel Execution Framework
3.4.2. Implementation of Vectorized Execution
3.4.3. Memory Management Mechanism
3.4.4. Implementation of Typical Operators
3.5. Transaction
3.5.1. Transaction on TiKV
3.5.2. Optimistic Transaction
3.5.3. Lock Resolver
3.5.4. Pessimistic Transaction
3.5.5. Async Commit
3.5.6. 1PC
3.5.7. MVCC garbage collection
3.6. Session
3.7. Privilege
3.8. Plugin
4. Project Management
4.1. Releases Train Model
4.2. TiDB Versioning
5. Extending TiDB
5.1. Add a function



TiDB Development Guide

Add a function

To add a builtin function to TiDB the best practice is to look at MySQL first and try to implement the function in such a way that it is compatible. Avoid adding functions that are already deprecated in MySQL or that might soon be deprecated.

Here we will implement a `HELLO()` function that has one argument that is a string. For this you need a clone of the [pingcap/tidb](#) repository

```
sql> SELECT HELLO("world");
ERROR: 1305 (42000): FUNCTION test.hello does not exist
```

The first step is to define the name of the function in `parser/ast/functions.go`:

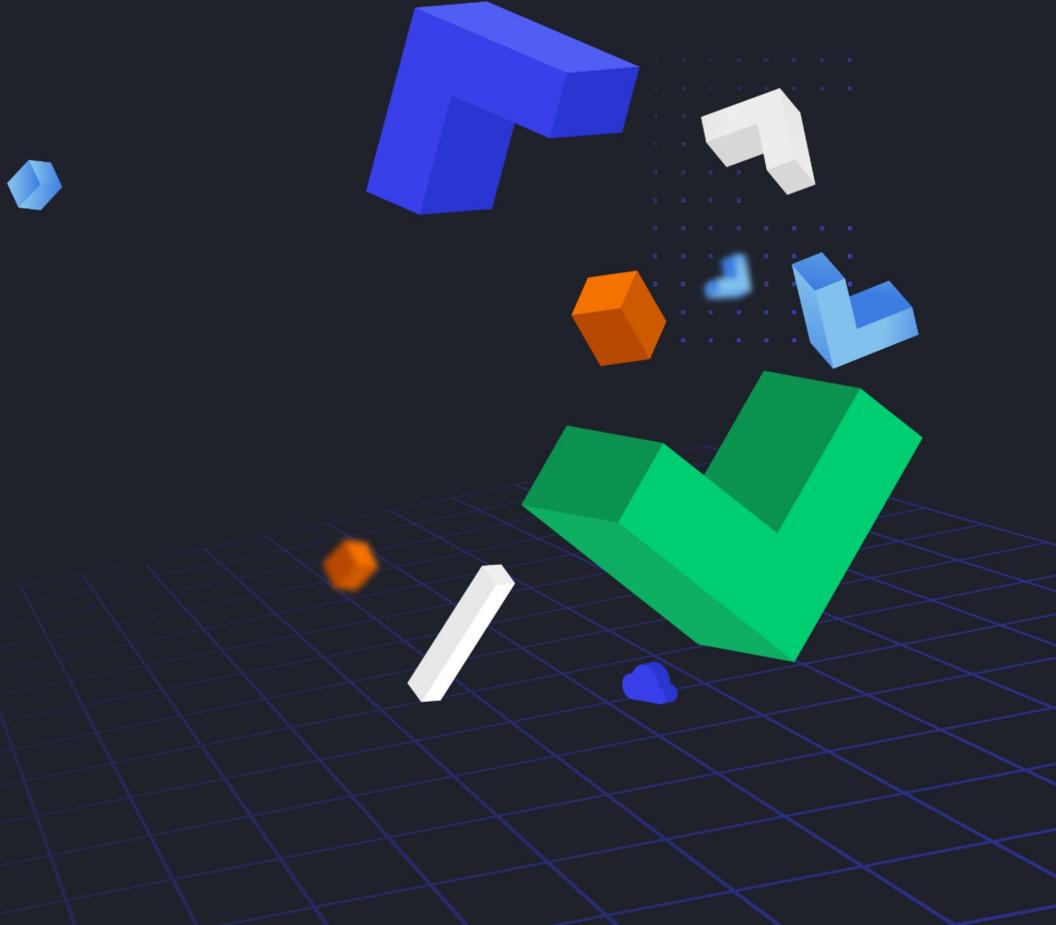
```
// List scalar function names.
const (
...
    Hello = "hello"
)
```

This links `ast.Hello` with "hello". Note that the lookup for the function is done with the lowercase name, so always use the lowercase name, otherwise it won't find the function.

The next step is to modify `expression/builtin.go`

```
var funcs = map[string]functionClass{
...
    ast.Hello: &helloFunctionClass{baseFunctionClass{ast.Hello, 1, 1}},
}
```

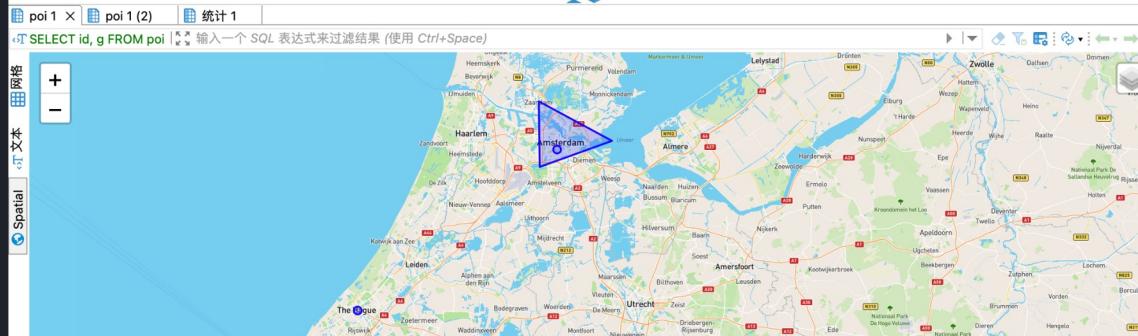
Demo



Function ST_Intersects()

```
bikeshare | *<bikeshare> Script X | *<bikeshare> Script-2 | <bikeshare> Script-1
-- ST_GeomFromText for insert and select point and polygon from tidb
-- ST_Intersects for calculate if points within a polygon or not
DROP TABLE IF EXISTS poi;
CREATE TABLE poi (
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    g GEOMETRY NOT NULL
);
SET @area := ST_GeomFromText('POLYGON((4.83375 52.33110, 5.04079 52.37569, 4.83113 52.44397, 4.83375 52.33110))'); -- Amsterdam City
INSERT INTO poi VALUES
(1, ST_GeomFromText('POINT(4.8830301 52.3599976)'), -- Rijksmuseum
(2, ST_GeomFromText('POINT(4.3158167 52.0788683)'), -- Eiffel Tower
(3, ST_GeomFromText('POINT(2.2922926 48.8583701)'), -- Eiffel Tower
(999, @area);
SELECT id, g FROM poi;
SELECT id, g, ST_Intersects(g, @area) FROM poi WHERE id < 999; -- If points IN area
```

123	id	010 011 110	g	123	ST_Intersects(g, @area)
1	1		POINT (4.8830301 52.3599976)	1	1
2	2		POINT (4.3158167 52.0788683)	0	0
3	3		POINT (2.2922926 48.8583701)	0	0



Function ST_Distance()

```
-- Example data is coming from https://account.capitalbikeshare.com/map
-- Data loaded with https://github.com/Hackathon-2022-GIS/tools/blob/master/bikeshare.py
-- 
-- Geo function: ST_Distance() with filters

SELECT b.bike_id, b.battery_pct, b.status,
       s.station_location
  FROM stations s INNER JOIN bikes b ON s.station_id = b.station_id
 WHERE ST_Distance(`station_location`,ST_GeomFromText('POINT (-77.0367641 38.8996581)')) < 0.004; -- find bike distance near 0.005

SELECT b.bike_id, b.battery_pct, b.status,
       s.station_location
  FROM stations s INNER JOIN bikes b ON s.station_id = b.station_id
 WHERE ST_Distance(`station_location`,ST_GeomFromText('POINT (-77.0367641 38.8996581)')) < 0.004
   AND b.battery_pct > 60 AND status IN('docked'); -- one SQL to query result from relational filter condition, IOT data and GIS data
```

bikes(+ 1) bikes(+ 1 (2)) 统计 1

输入一个 SQL 表达式来过滤结果 (使用 Ctrl+Space)

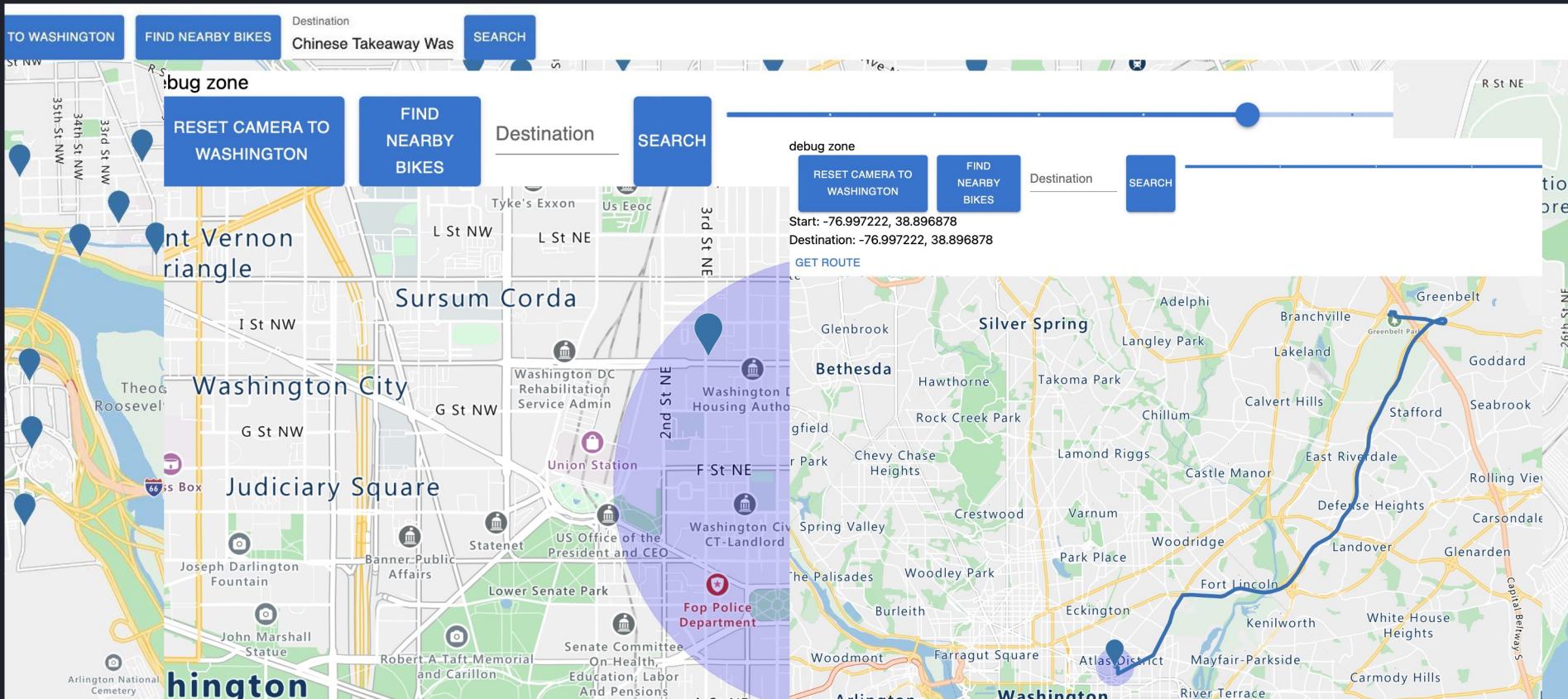
121 bike_id	123 battery_pct	RBC status	010 011 station_location
1 3,458,764,513,820,541,204	86	docked	POINT (-77.03445106744766 38.8996581)

数值查看器 X

Map Text

Leaflet | Map data © OpenStreetMap contributors CC-BY-SA, Imagery Mapbox

Shared bike



Summary



Parse GIS SQL

Support GIS data types

- Point
- Line
- Polygon



Encode/Decode GIS data

Support GIS functions

- ST_GeomFromText()
- ST_AsText()
- ST_Distance()
- ST_Intersects()



Integrated with Map API

Easy SQLs can build MVP

- Bing map integration
- Few SQLs can get core business logic



Beyond Hackathon!

- Global Cooperation ☺
- Community Feedback ☺

Future work



- More spatial predicates
- More spatial functions
- Compatibility

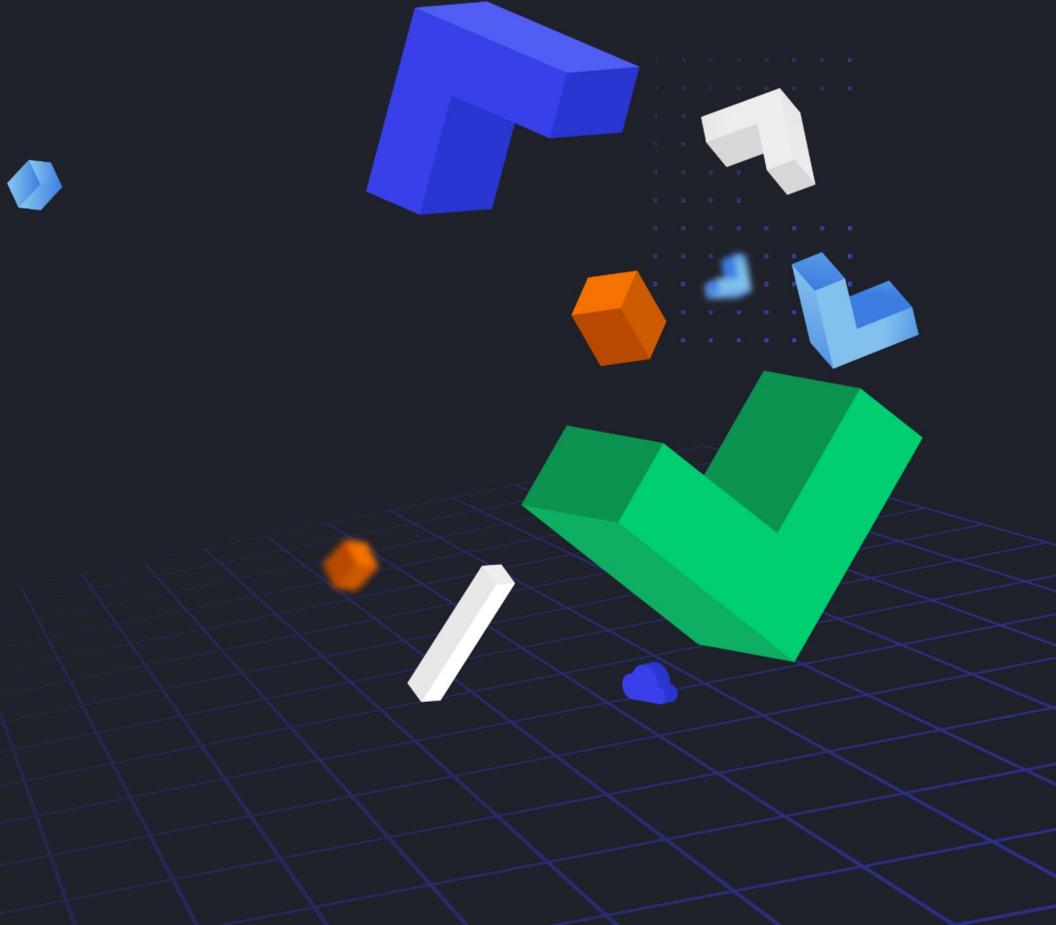


- Spatial index(Quad Tree, Rectangle Tree)
- Optimizer adaptive, accelerate search, calculation, prediction performance



- SRID and WGS84
- Integration tools
- Community & Ecosystem

THANK YOU!



References