

---

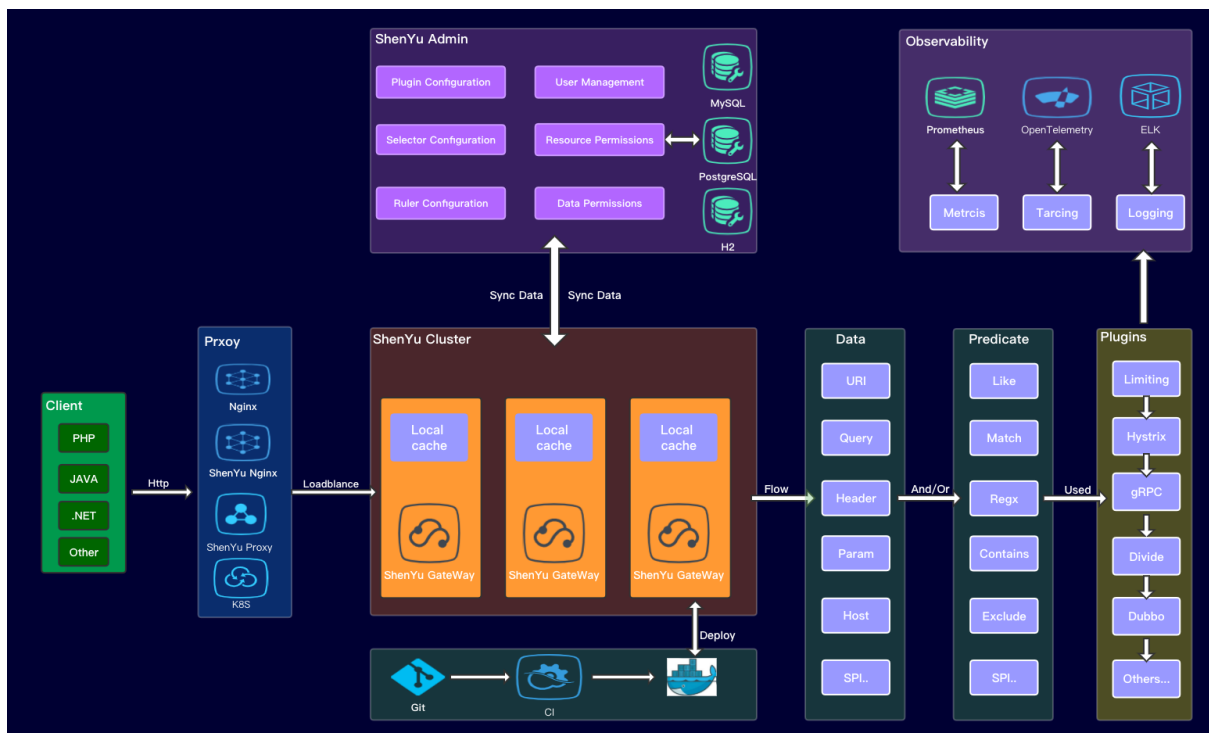
# **Apache ShenYu document**

**Apache ShenYu**

**Jul 22, 2022**



## Architecture



## Why named Apache ShenYu

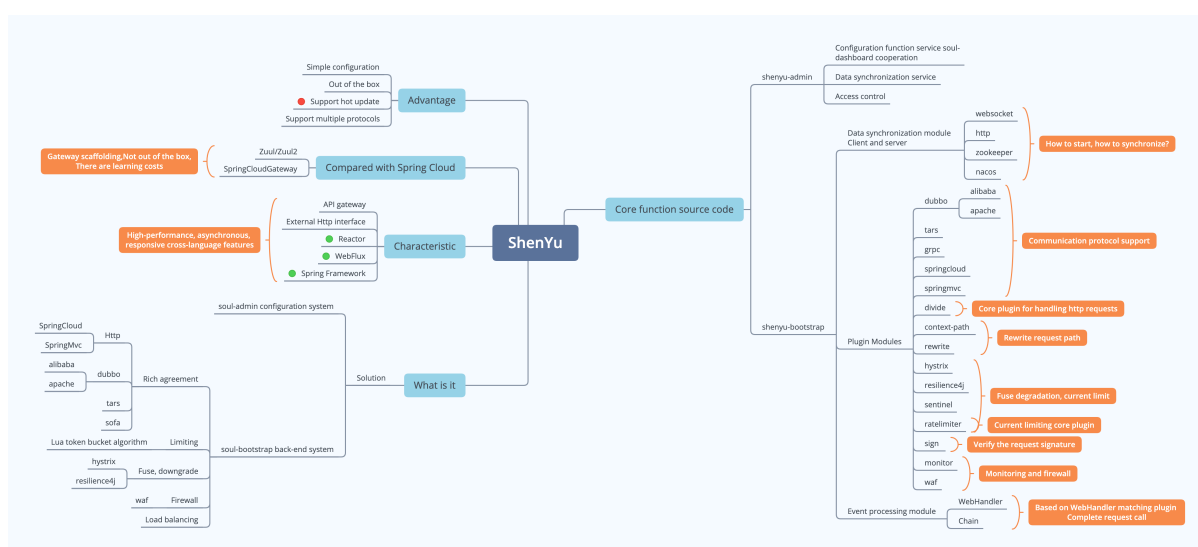
ShenYu (神禹) is the honorific name of Chinese ancient monarch Xia Yu (also known in later times as Da Yu), who left behind the touching story of the three times he crossed the Yellow River for the benefit of the people and successfully managed the flooding of the river. He is known as one of the three greatest kings of ancient China, along with Yao and Shun.

- Firstly, the name ShenYu is to promote the traditional virtues of our Chinese civilisation.
  - Secondly, the most important thing about the gateway is the governance of the traffic.
  - Finally, the community will do things in a fair, just, open and meritocratic way, paying tribute to ShenYu while also conforming to the Apache Way.
-

## Features

- Proxy: Support for Apache® Dubbo™, Spring Cloud, gRPC, Motan, SOFA, TARS, WebSocket, MQTT
  - Security: Sign, OAuth 2.0, JSON Web Tokens, WAF plugin
  - API governance: Request, response, parameter mapping, Hystrix, RateLimiter plugin
  - Observability: Tracing, metrics, logging plugin
  - Dashboard: Dynamic traffic control, visual backend for user menu permissions
  - Extensions: Plugin hot-swapping, dynamic loading
  - Cluster: NGINX, Docker, Kubernetes
  - Language: provides .NET, Python, Go, Java client for API register
-

## Mind maps



## Quick Start (docker)

### 5.1 Run Apache ShenYu Admin

```
> docker pull apache/shenyu-admin
> docker network create shenyu
> docker run -d -p 9095:9095 --net shenyu apache/shenyu-admin
```

### 5.2 Run Apache ShenYu Bootstrap

```
> docker network create shenyu
> docker pull apache/shenyu-bootstrap
> docker run -d -p 9195:9195 --net shenyu apache/shenyu-bootstrap
```

### 5.3 Set router

- Real requests : <http://127.0.0.1:8080/helloworld>,

```
{
  "name" : "Shenyu",
  "data" : "hello world"
}
```

- Set routing rules (Standalone)

Add `localKey: 123456` to Headers. If you need to customize the `localKey`, you can use the `sha512` tool to generate the key based on plaintext and update the `shenyu.local.sha512Key` property.

```
curl --location --request POST 'http://localhost:9195/shenyu/plugin/
selectorAndRules' \
--header 'Content-Type: application/json' \
```

```
--header 'localKey: 123456' \  
--data-row '{  
  "pluginName": "divide",  
  "selectorHandler": "[{\"upstreamUrl\":\"127.0.0.1:8080\"}]",  
  "conditionDataList": [{  
    "paramType": "uri",  
    "operator": "match",  
    "paramValue": "/**"   
  }],  
  "ruleDataList": [{  
    "ruleHandler": "{\"loadBalance\":\"random\"}",  
    "conditionDataList": [{  
      "paramType": "uri",  
      "operator": "match",  
      "paramValue": "/**"   
    }]  
  }]  
}'
```

- Proxy request : <http://localhost:9195/helloworld>

```
{  
  "name" : "Shenyu",  
  "data" : "hello world"  
}
```

---



Whenever a request comes in, Apache ShenYu will execute it by all enabled plugins through the chain of responsibility.

As the heart of Apache ShenYu, plugins are extensible and hot-pluggable.

Different plugins do different things.

Of course, users can also customize plugins to meet their own needs.

If you want to customize, see [custom-plugin](#) .

---

## Selector & Rule

According to your HTTP request headers, selectors and rules are used to route your requests.

Selector is your first route, It is coarser grained, for example, at the module level.

Rule is your second route and what do you think your request should do. For example a method level in a module.

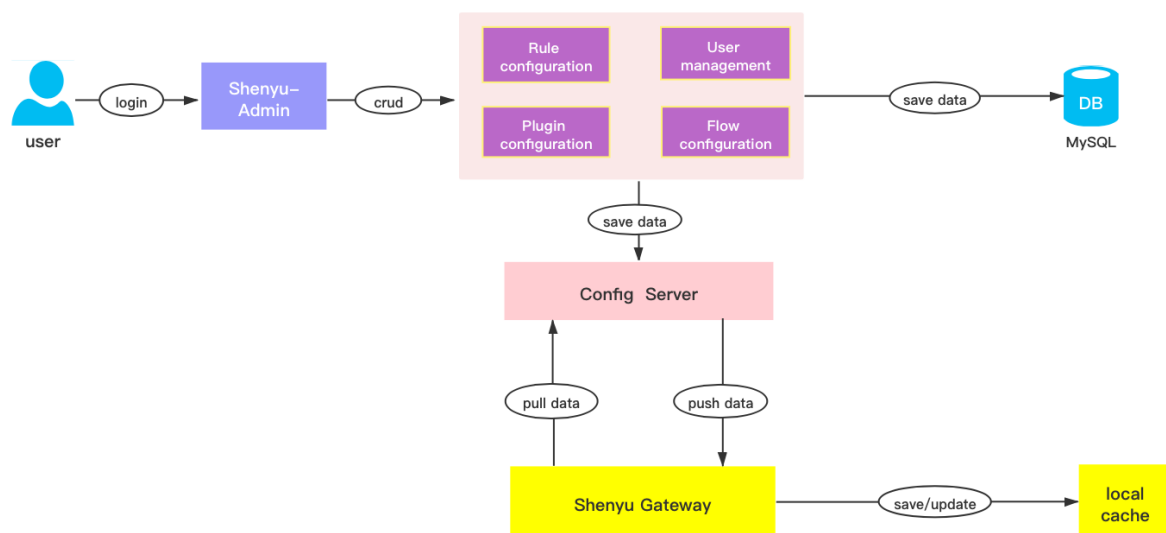
The selector and the rule match only once, and the match is returned. So the coarsest granularity should be sorted last.

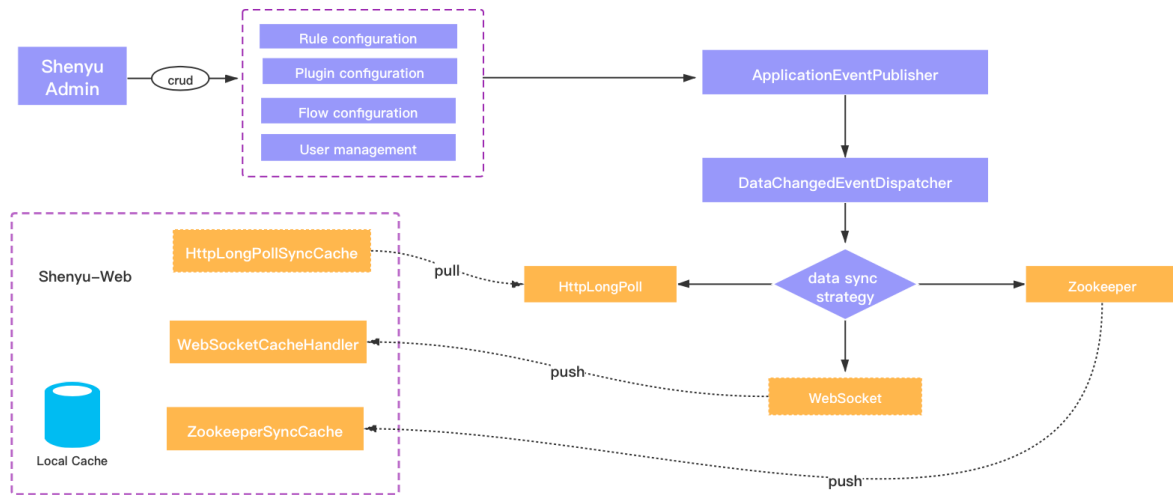
---

## Data Caching & Data Sync

Since all data have been cached using ConcurrentHashMap in the JVM, it's very fast.

Apache ShenYu dynamically updates the cache by listening to the ZooKeeper node (or WebSocket push, HTTP long polling) when the user changes configuration information in the background management.





**Prerequisite**

- JDK 1.8+
-