

HUAWEI CLOUD Microservice Tool Improves Development Efficiency

Department: Application Platform Service
Author: Wang Qijun
Date: 2019-09-20



Security Level:



Contents

1. Tool for Splitting Monolithic Applications into Microservices
2. Contract Management Tool

Advantages of the Microservice Architecture

Factor	Monolithic Architecture	Microservice Architecture
Delivery speed	Slow	Fast
Fault isolation scope	Thread-level	Process-level
Overall availability	Low	High
Continuous evolution	Difficult	Easy
Communication efficiency	Low	High
Technology stack selection	Restricted	Flexible
Scalable	Restricted	Flexible
Reusability	Low	High
Difficulty in breaking down business complexity	Difficult	Easy
Product innovation complexity	Difficult	Easy
Cost for achieving consistency	Low	High
Latency	Low	High
Resource cost	Low	High
Correlation query complexity	Easy	Complex
Remote calling	N/A	Involved
Service governance	N/A	Involved
Requirements for R&D personnel	Low	High
Dependency on tools	Low	Relatively high
O&M complexity	Low	High

Major Difficulty in Building the Microservice Architecture: How to Split an Application into Microservices?

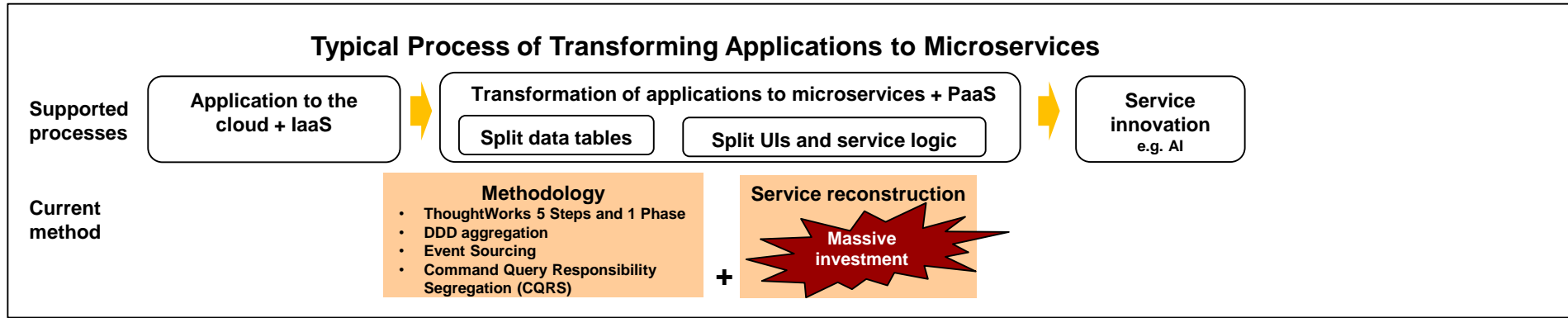
- Factors

- › Team size
- › Delivery cycle
- › Business direction
- › Fault scope
- › Data scale
- › Throughput
- › Consistency
- › ...

Possible Problems Caused by Improper Splitting

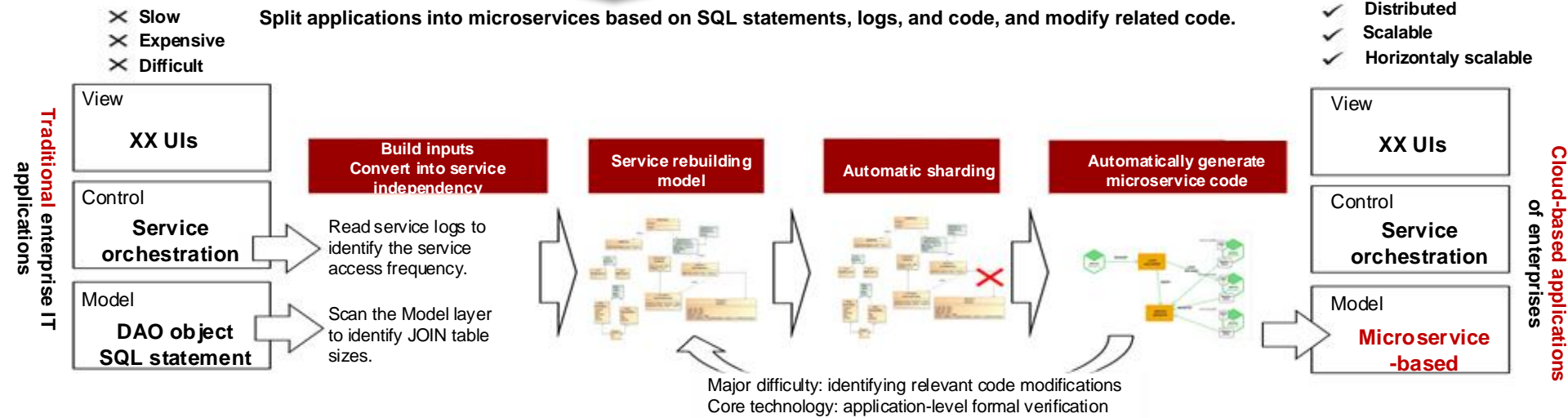
- Possible problems:
 - Services increase explosively, making service O&M more complex.
 - Too few services are available, which cannot be flexibly used.
 - One user story may affect multiple services.
 - APIs change frequently.
 - A massive number of association queries are performed.
 - The system architecture complexity increases.

Tool for Splitting Monolithic Applications into Microservices Improves Development Efficiency

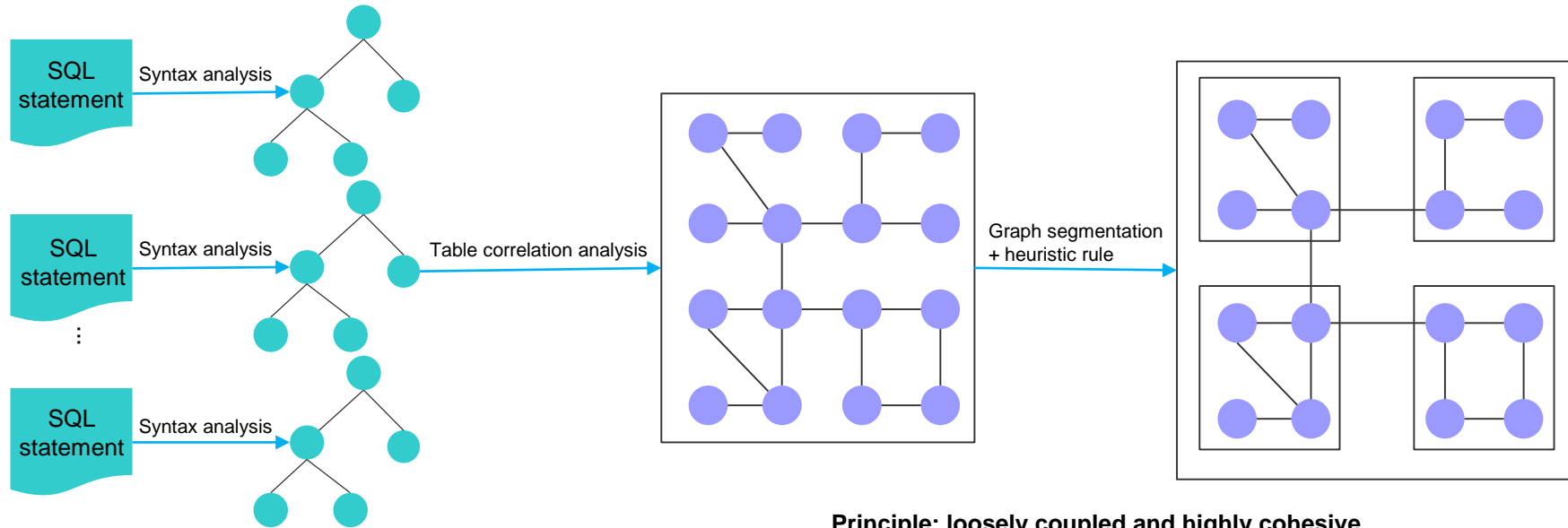


Challenges: The rebuilding process is highly business-related and time-consuming, and it requires participation of business and microservice experts.

Key technology: toolkit for transforming applications to microservices



Tool for Splitting Applications to Microservices — Automatic Sharding



Step 1: Extract all SQL statements in the system.

Step 2: Create syntax trees using SQL statements.

Step 3: Analyze the table correlations in each syntax tree and generate a weighted graph.

Step 4: Shard data tables to databases by means of the graph segmentation algorithm and heuristic rule.

Principle: loosely coupled and highly cohesive

Coupling: sum of weights of the edges connecting microservices after an application is split into microservices

Cohesion: sum of weights of the edges between tables in a microservice after an application is split into microservices

Heuristic rule (customizable) for splitting microservices using the graph searching algorithm:

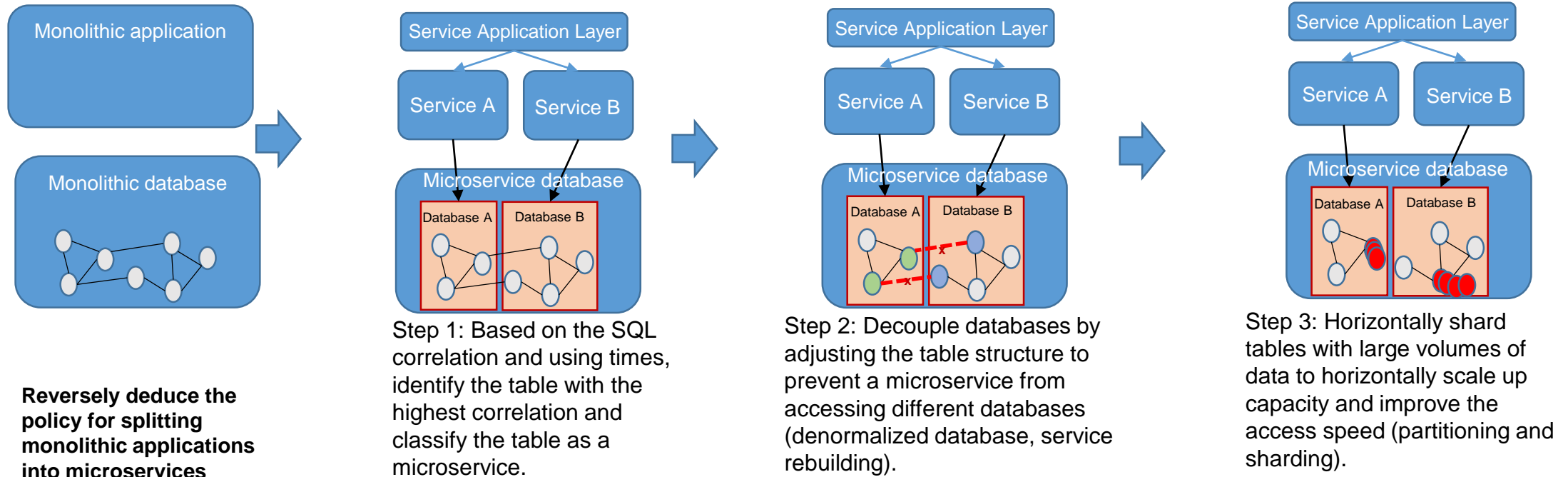
Rule 1: greedy algorithm

Rule 2: An independent table can belong to any microservice or be an independent microservice.

Rule 3: The number of tables belonging to the same microservice cannot be less than 5 (configurable).

Rule 4: Tables whose correlation is greater than 10 (configurable) belong to the same microservice.

Tool for Splitting Monolithic Applications into Microservices



Reversely deduce the policy for splitting monolithic applications into microservices based on the table correlation and usage.

Step 1: Based on the SQL correlation and using times, identify the table with the highest correlation and classify the table as a microservice.

Step 2: Decouple databases by adjusting the table structure to prevent a microservice from accessing different databases (denormalized database, service rebuilding).

Step 3: Horizontally shard tables with large volumes of data to horizontally scale up capacity and improve the access speed (partitioning and sharding).

Step 4: Automatically adjust SQL statements such as DDL/CRUD based on table structure changes to generate Java data access services.

Step 5: Execute data tables and migrate data.

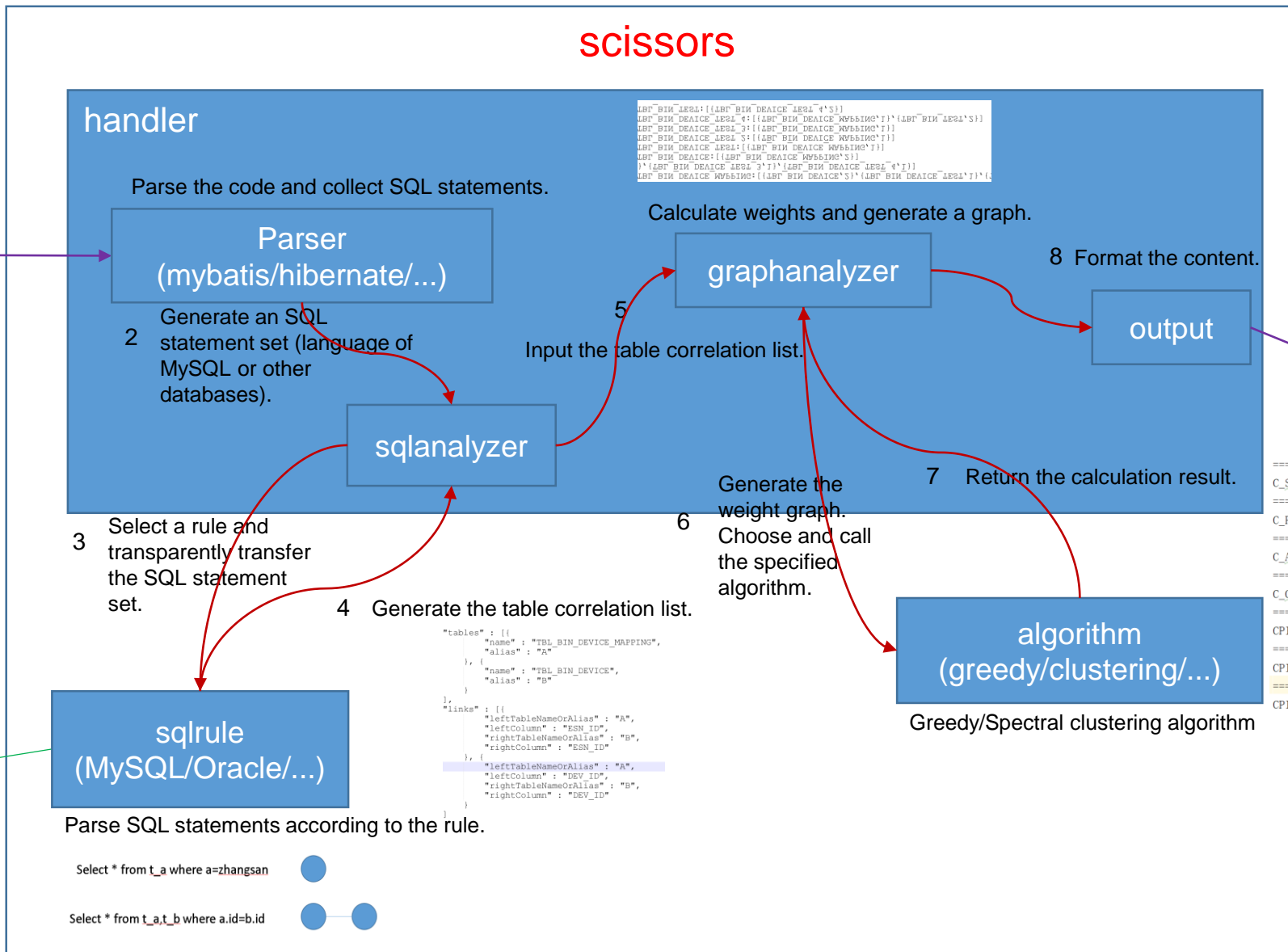
Principle for Splitting Monolithic Applications into Microservices



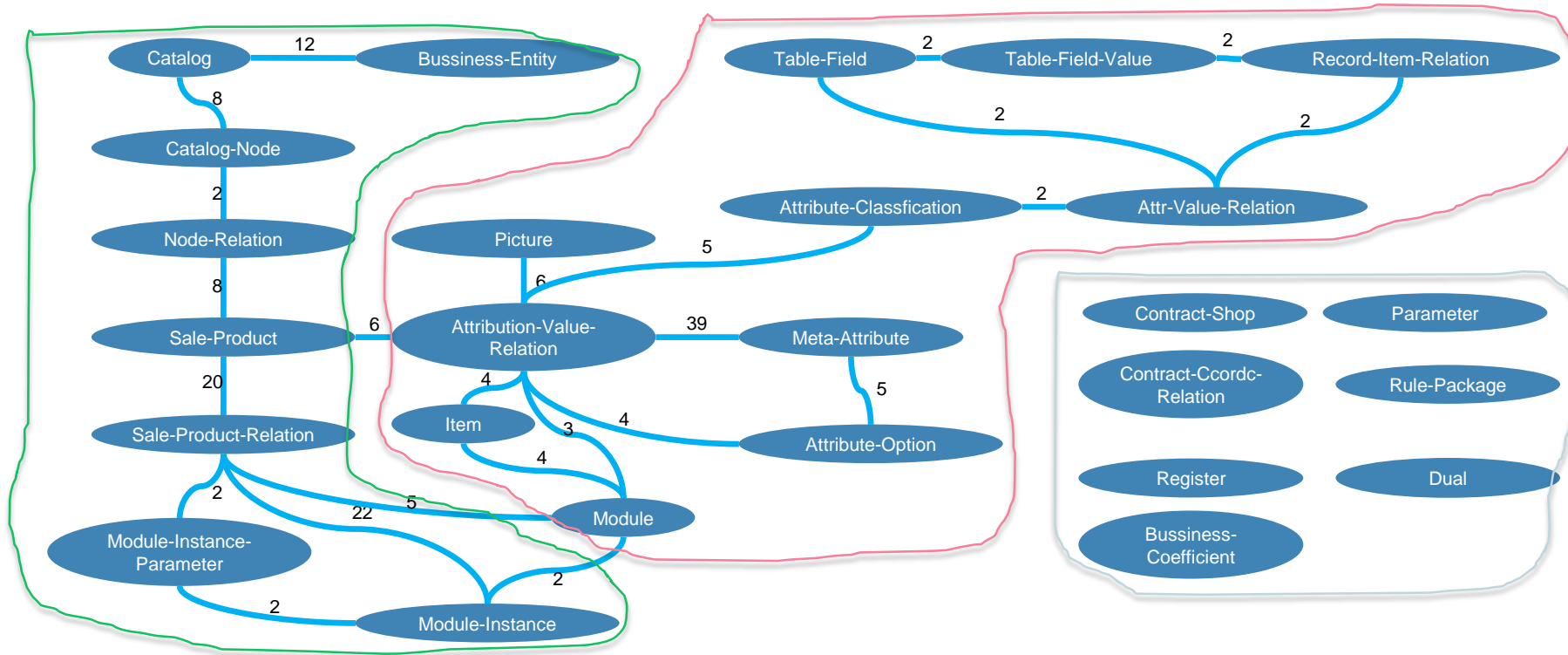
1. Specify the code path. /home/xxx/project
2. Specify the algorithm (greedy algorithm by default).
3. Specify the SQL rule. (MySQL is used by default, and intelligent analysis will be used in the future.)

Preprocessing during development: Use the Antlr4 to generate objects of parsing rules that are recognizable to Java, and embed the objects into the source code.

MySQL/Oracle rule file

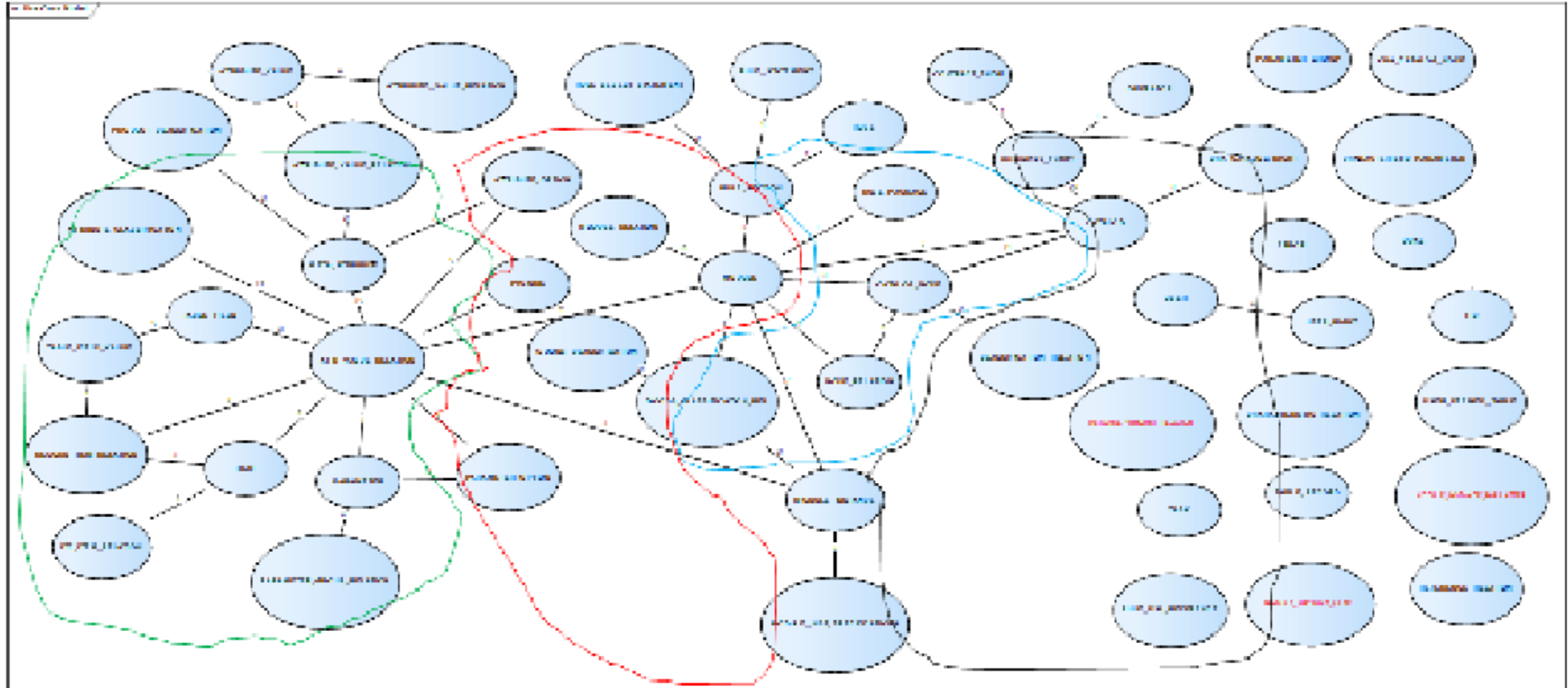


Case 1: Analysis on Splitting Legacy System 1 into Microservices (Using the Graph Searching Algorithm)



The picture shows offering library, template library, and discrete tables from left to right. The offering library and template library come from the rebuilt legacy system. The discrete tables are introduced by the system to associate with external systems and are stored in a separate library.

Case 2: Analysis on Splitting Legacy System 2 into Microservices (Using the Graph Searching Algorithm)



The picture shows extended attribute services, model/model instance services, catalog services, and discrete table services from left to right. The discrete table services in red should be classified into the model/model instance services, but are automatically identified as independent services. This is because the SQL does not present the correlation between these discrete table services and model/model instances.

API Management and Control Challenges in Microservice Architecture


- Explosive growth of APIs in the microservice architecture
- Mobile Internet, and IoT
- An API is equivalent to a contract.
- API First, a developer for decoupling services

ServiceStage Contract Management Tool — Separation of Management and Control

*** 应用设计名称** model-sc8345

*** 版本** 1.0.1

描述 This is a sample server Petstore server.

*** 契约地址**  GitHub

Github是一家源代码托管网站, 提供商业计划和免费帐户

授权信息

用户名/组织 仓库名称

*** 契约文件**

契约文件	契约地址
petstore	petstore.yaml

新增契约

```
1
2 swagger: "2.0"
3 info:
4   description: "This is a sample server Petstore server. You can find out
5   version: "1.0.0"
6   title: "Swagger Petstore"
7   termsOfService: "http://swagger.io/terms/"
8   contact:
9     email: "apiteam@swagger.io"
10  license:
11    name: "Apache 2.0"
12    url: "http://www.apache.org/licenses/LICENSE-2.0.html"
13 host: "petstore.swagger.io"
14 basePath: "/v2"
15 tags:
16 - name: "pet"
17   description: "Everything about your Pets"
18   externalDocs:
19     description: "Find out more"
20     url: "http://swagger.io"
21 - name: "store"
22   description: "Access to Petstore orders"
23 - name: "user"
24   description: "Operations about user"
25   externalDocs:
26     description: "Find out more about our store"
27     url: "http://swagger.io"
28 schemes:
29 - "https"
30 - "http"
31 paths:
32 /pet:
33   post:
34     tags:
35     - "pet"
36     summary: "Add a new pet to the store"
37     description: ""
```

*** 契约文件**

*** 契约地址**

Swagger Petstore 1.0.0

[Base URL: petstore.swagger.io/v2]

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on [#swagger](irc://irc.freenode.net). For this sample, you can use the api key `special-key` to test the authorization filters.

[Terms of service](#)
[Contact the developer](#)
[Apache 2.0](#)
[Find out more about Swagger](#)

pet Everything about your Pets Find out more: <http://swagger.io>

- POST** /pet Add a new pet to the store
- PUT** /pet Update an existing pet
- GET** /pet/findByStatus Finds Pets by status

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home and
organization for a fully connected,
intelligent world.

**Copyright©2019 Huawei Technologies Co., Ltd.
All Rights Reserved.**

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

