# Dubbo and Weidian's Practice on Microservice Architecture

Speaker：Xin Wang

CONTENTS

- Who's Weidian
- The stages of Weidian
- Challenges & Solutions
- About the future

# Who's Weidian ?

Weidian is a app of China to help sellers open their shop on mobile phones. Anyone can open their own shop through mobile phone numbers.

**Scale**
70 million

- 70 million stores, 1 billion goods; 9P data
- Payment, search, recommendation, risk control, IM, transaction, open platform, advertising, supply chain…

**Security**
600 million / day

- 5.6 million attacks / days;
- CC、SQLInjection、XSS、CSRF …
- 600 million + daily visit of crawlers…

**Cost**
Tens of millions / year

- In IDC, 2016/2017, our total expenditure is tens of millions of RMB.

**Efficiency**
600+

- 600+ RD work together
- 60 times a day on the line, 400+ release (including test / Pre deployment environment)

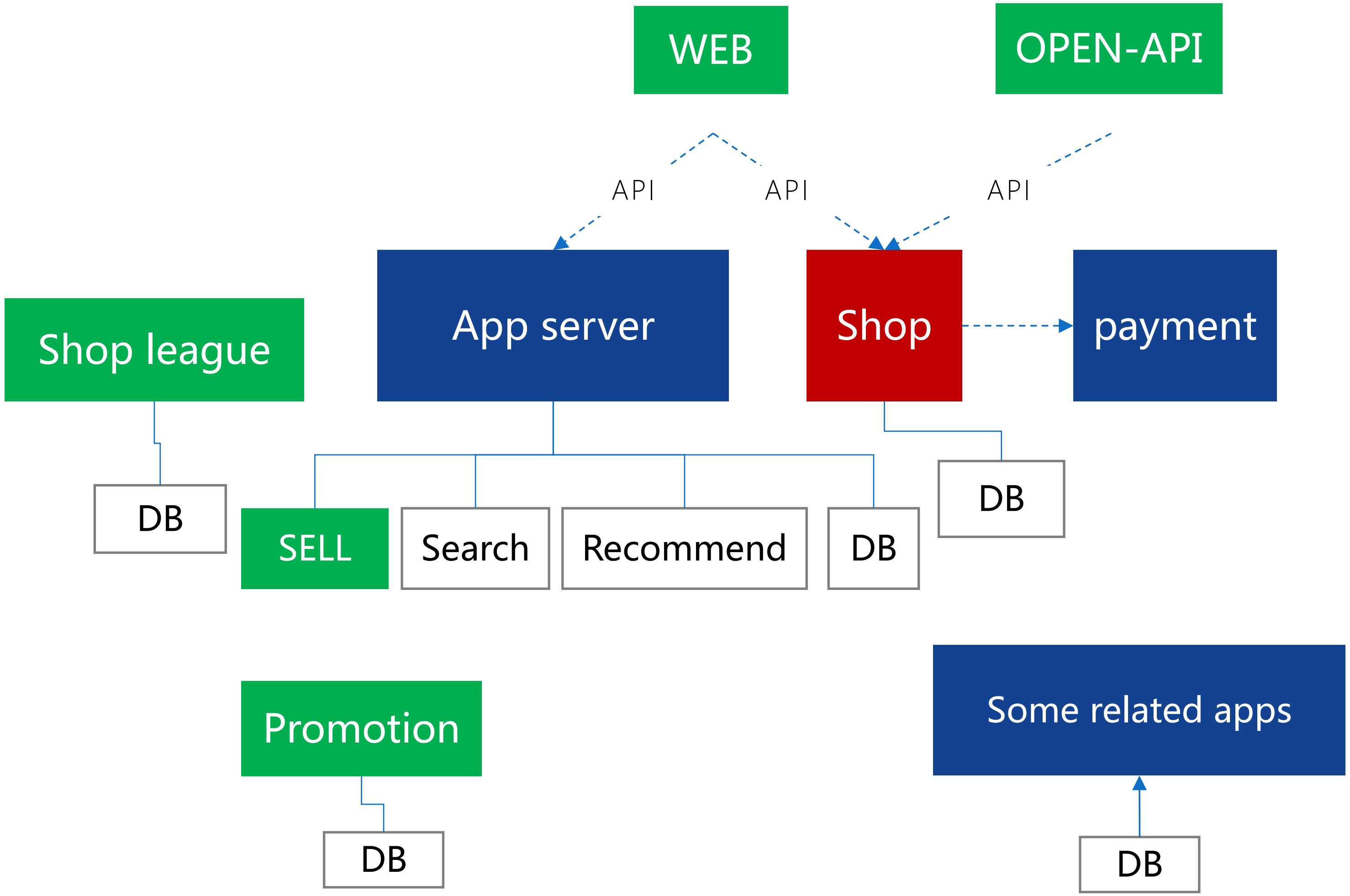
# Dubbo and Weidian's Practice on Microservice Architecture

Speaker：Xin Wang

CONTENTS

- Who's Weidian
- The stages of Weidian
- Challenges & Solutions
- About the future

# Monolithic application stage



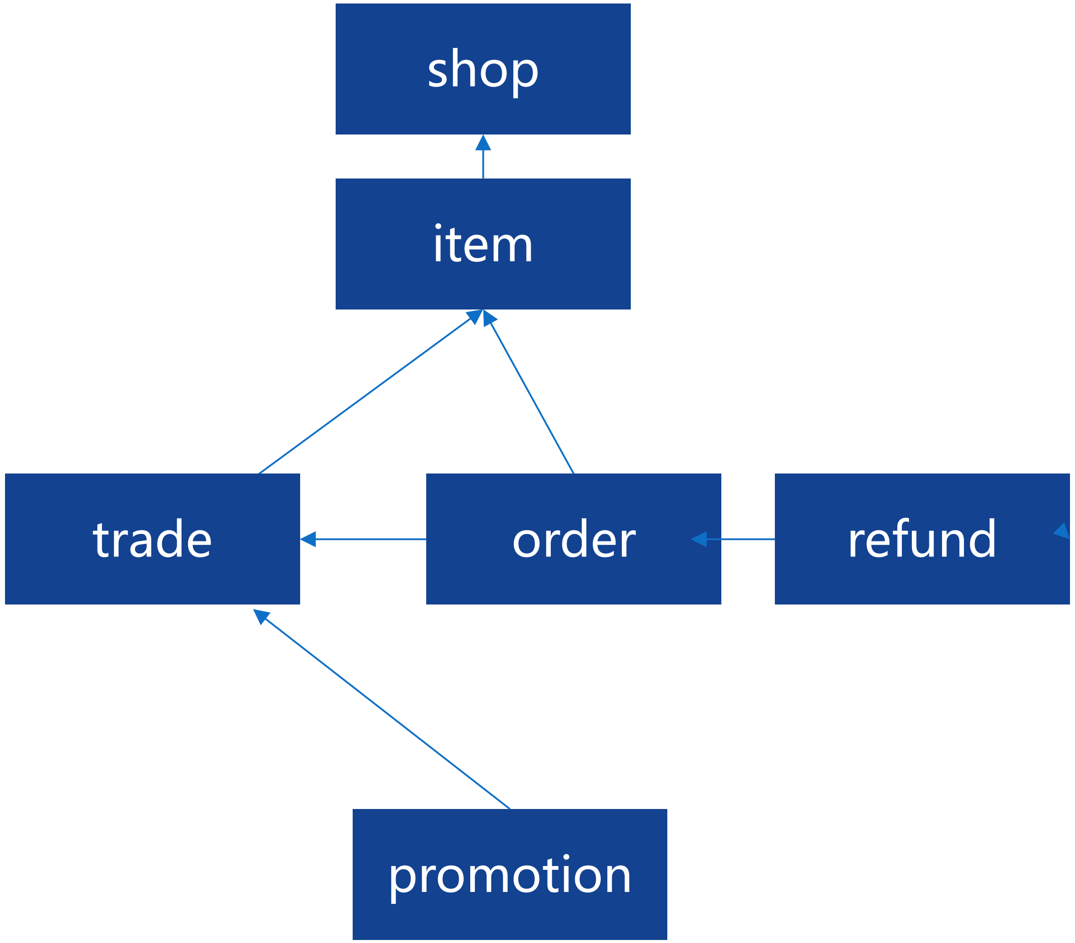
## Technology stack

- PHP
- Nginx
- Mysql
- Redis

## Technology stack

- Testing is difficult: requires all functional regression
- Scalability: weak
- Edge BUG, such as dead loop, OOM, and so on, drag down the entire site.

# Multi module stage



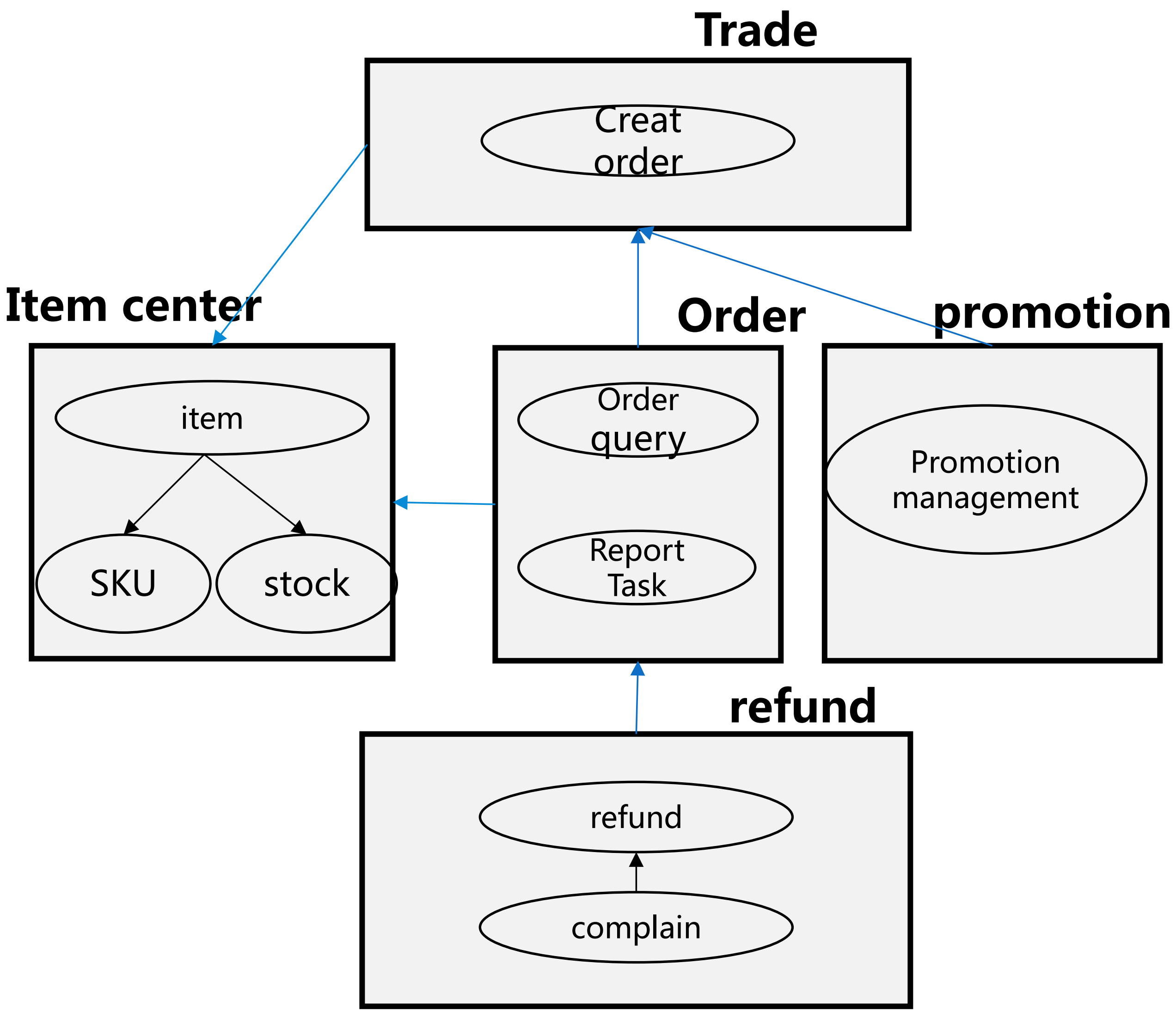
## Technology stack

- PHP
- Nginx
- Mysql
- Redis

## Problems encountered

- Code splitting, independent deployment, process isolation, technology stack has not changed much.
- Problems :
  - It is difficult to upgrade，because need to push the whole site
  - The pressure of the database connection pool is great

# Servitization stage



- **Service group appears, order center : item center, trading center, etc.**
- **Technology stack:**
  - Dubbo
  - VTrace
  - Vdds
- **Technology stack:**
  - Problem tracking is difficult
  - Configuration is troublesome, ops have heavy work
  - Difficulty in testing: interference with each other

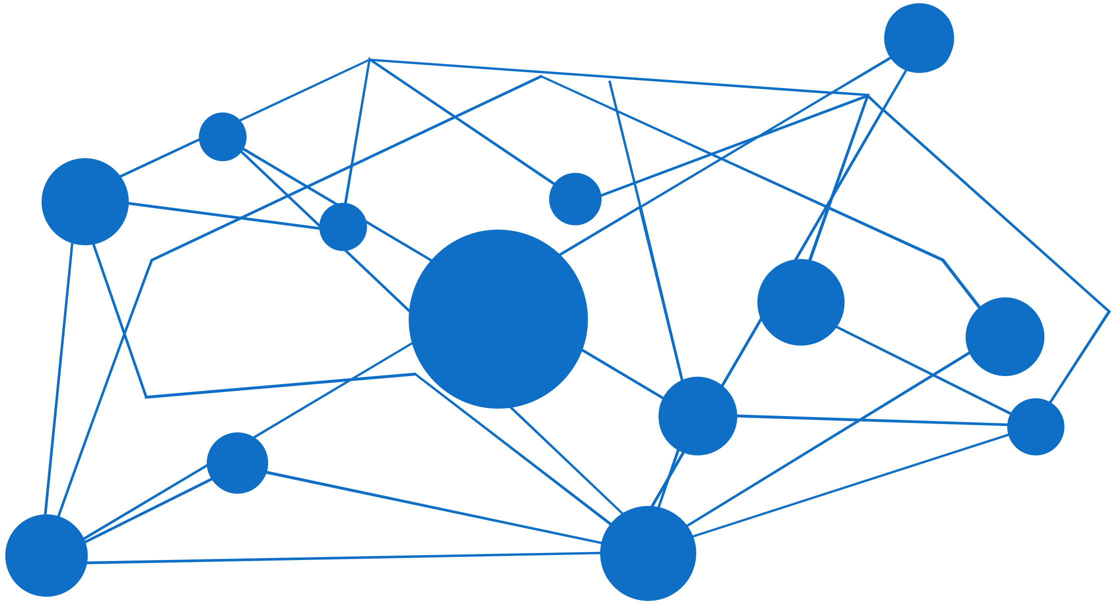
# Dubbo and Weidian's Practice on Microservice Architecture

Speaker：Xin Wang

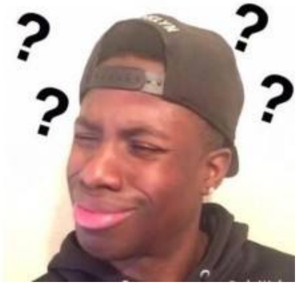
CONTENTS

- Who's Weidian
- The stages of Weidian
- Challenges & Solutions
- About the future

# Servitization-Module split



Complex business relations



**How to split the module?**

- Consider the business first VS Consider the data table first?
- Should the number of lines of code be used as the basis for subcontracting?
- How to balance the pursuit of design, engineering, and operation?

# Servitization-Module split

Microservices - the new architectural style

*Martin Fowler, Mar 2014*

## ◆ Microservice:

- System consisting of distributed services
- Organize organizations by business rather than technology
- Automated operation and maintenance
- Rapid evolution of service iteration

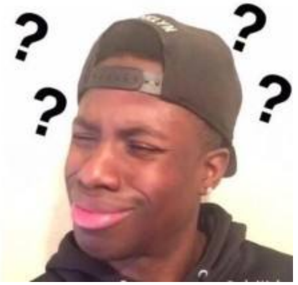
## ◆ DDD Domain-driven Design

- Solve core issues in the domain by building domain models
- Implementation of code driven by domain models
- Technical architecture design or data storage, etc. are on the periphery of the domain model
- DDD vs SSI ( spring+hibernate ) : congestion model / blood loss model

## Split principle:

- DDD Business-driven
- Balance of design, engineering, operations, etc.
- Progressive evolution

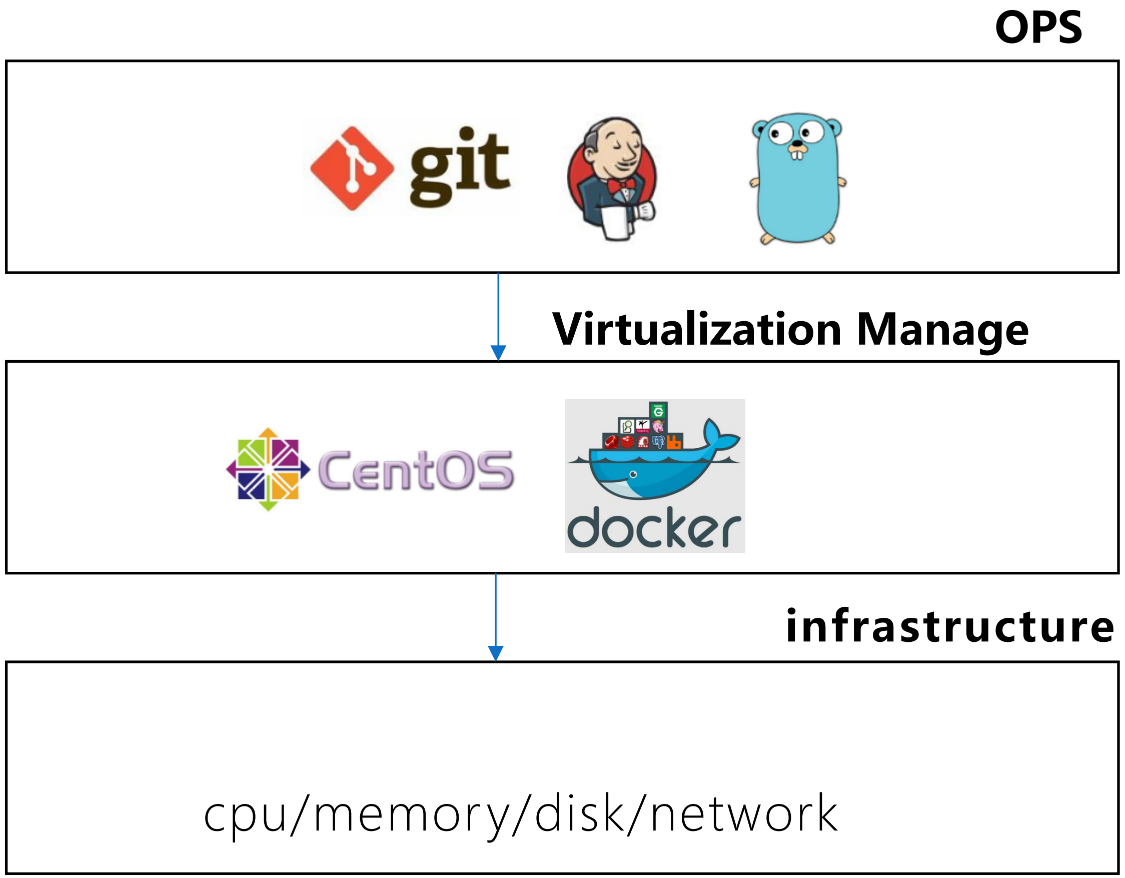
# Servitization-Automated deployment



**Ops and deployment problem**

- Services, the configuration workload is getting bigger, how to ensure that no mistakes are made?
- How to ensure that the test system is independent according to business needs?
- How to implement the CI/CD pipeline system?
- How do you fully squeeze the usage of the infrastructure (CPU/memory/disk/network)?

# Servitization-Automated deployment



OPS

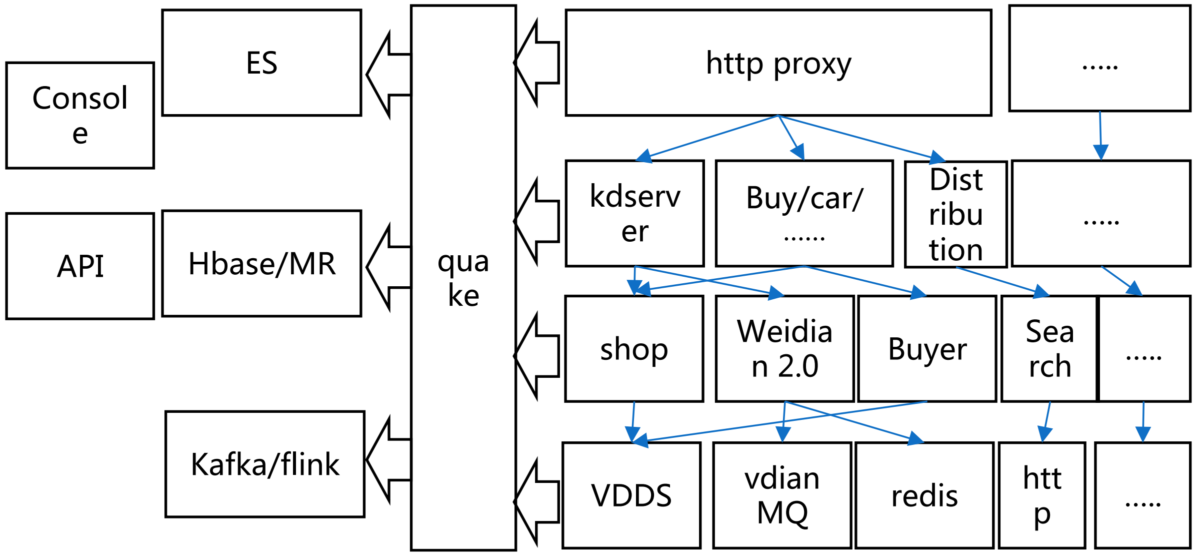
Virtualization Manage

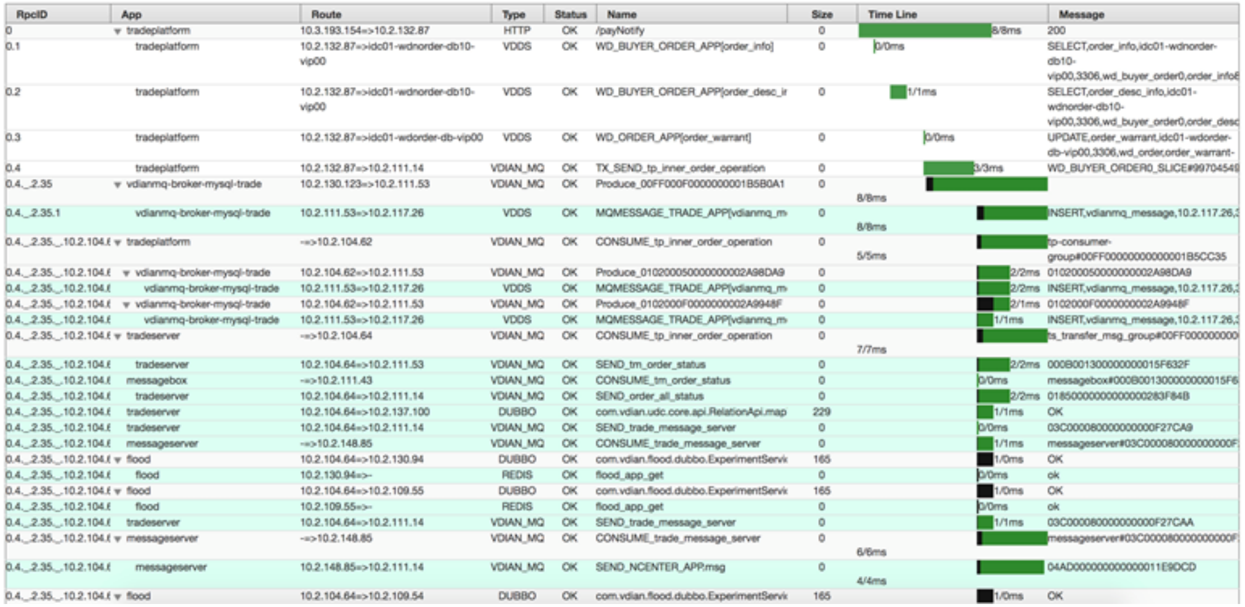
infrastructure

cpu/memory/disk/network

- **OS version**
  - Centos7.2
- **Virtualization**
  - KVM
  - Docker
- Network mode
  - Bridge
    - ✓ Pipework
    - ✓ Libvirt
  - Configuring unique IP, and interworking all over the network

# Servitization-Call link tracking

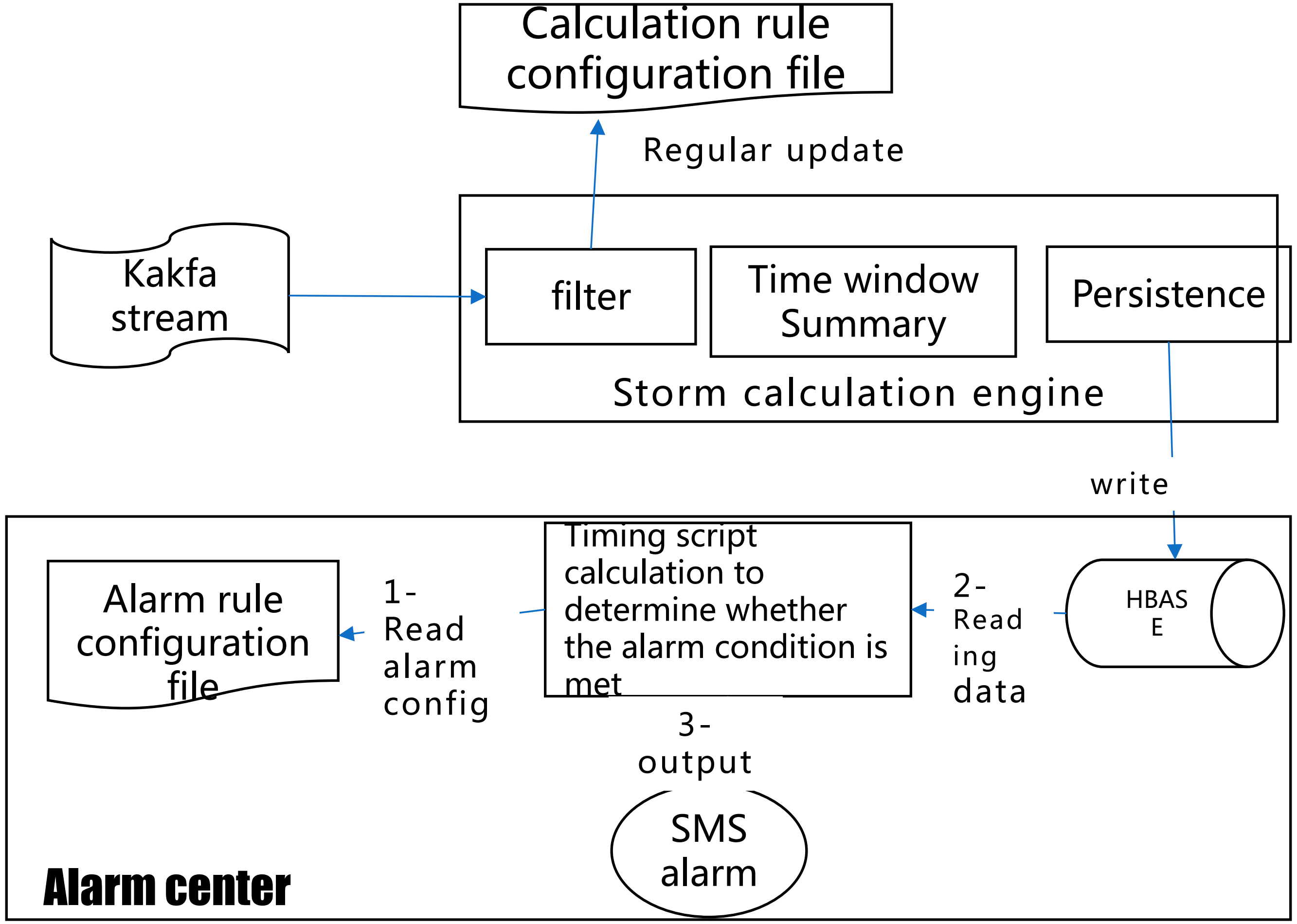




| RpcID | App | Route | Type | Status | Name | Size | Time Line | Message |
|---|---|---|---|---|---|---|---|---|
| 0 | ▾ tradeplatform | 10.3.193.154=>10.2.132.87 | HTTP | OK | /payNotify | 0 | 8/8ms | 200 |
| 0.1 | tradeplatform | 10.2.132.87=>idc01-wdnorder-db10-vip00 | VDDS | OK | WD_BUYER_ORDER_APP[order_info] | 0 | 0/0ms | SELECT,order_info,idc01-wdnorder-db10-vip00,3306,wd_buyer_order0,order_info8 |
| 0.2 | tradeplatform | 10.2.132.87=>idc01-wdnorder-db10-vip00 | VDDS | OK | WD_BUYER_ORDER_APP[order_desc_ir | 0 | 1/1ms | SELECT,order_desc_info,idc01-wdnorder-db10-vip00,3306,wd_buyer_order0,order_desc |
| 0.3 | tradeplatform | 10.2.132.87=>idc01-wdorder-db-vip00 | VDDS | OK | WD_ORDER_APP[order_warrant] | 0 | 0/0ms | UPDATE,order_warrant,idc01-wdorder-db-vip00,3306,wd_order,order_warrant- |
| 0.4 | tradeplatform | 10.2.132.87=>10.2.111.14 | VDIAN_MQ | OK | TX_SEND_tp_inner_order_operation | 0 | 3/3ms | WD_BUYER_ORDER0_SLICE#99704549 |
| 0.4._2.35 | ▾ vdianmq-broker-mysql-trade | 10.2.130.123=>10.2.111.53 | VDIAN_MQ | OK | Produce_00FF000F000000001B5B0A1 | 0 | 8/8ms | |
| 0.4._2.35.1 | vdianmq-broker-mysql-trade | 10.2.111.53=>10.2.117.26 | VDDS | OK | MQMESSAGE_TRADE_APP[vdianmq_m | 0 | 8/8ms | INSERT,vdianmq_message,10.2.117.26,3 |
| 0.4._2.35._10.2.104.6 | ▾ tradeplatform | =>10.2.104.62 | VDIAN_MQ | OK | CONSUME_tp_inner_order_operation | 0 | 5/5ms | tp-consumer-group#00FF0000000000001B5CC35 |
| 0.4._2.35._10.2.104.6 | ▾ vdianmq-broker-mysql-trade | 10.2.104.62=>10.2.111.53 | VDIAN_MQ | OK | Produce_01020005000000002A98DA9 | 0 | 2/2ms | 01020005000000002A98DA9 |
| 0.4._2.35._10.2.104.6 | vdianmq-broker-mysql-trade | 10.2.111.53=>10.2.117.26 | VDDS | OK | MQMESSAGE_TRADE_APP[vdianmq_m | 0 | 2/2ms | INSERT,vdianmq_message,10.2.117.26,3 |
| 0.4._2.35._10.2.104.6 | ▾ vdianmq-broker-mysql-trade | 10.2.104.62=>10.2.111.53 | VDIAN_MQ | OK | Produce_0102000F000000002A9948F | 0 | 2/1ms | 0102000F000000002A9948F |
| 0.4._2.35._10.2.104.6 | vdianmq-broker-mysql-trade | 10.2.111.53=>10.2.117.26 | VDDS | OK | MQMESSAGE_TRADE_APP[vdianmq_m | 0 | 1/1ms | INSERT,vdianmq_message,10.2.117.26,3 |
| 0.4._2.35._10.2.104.6 | ▾ tradeserver | =>10.2.104.64 | VDIAN_MQ | OK | CONSUME_tp_inner_order_operation | 0 | 7/7ms | ts_transfer_msg_group#00FF000000000 |
| 0.4._2.35._10.2.104.6 | tradeserver | 10.2.104.64=>10.2.111.53 | VDIAN_MQ | OK | SEND_tm_order_status | 0 | 2/2ms | 000B00130000000015F632F |
| 0.4._2.35._10.2.104.6 | messagebox | =>10.2.111.43 | VDIAN_MQ | OK | CONSUME_tm_order_status | 0 | 0/0ms | messagebox#000B00130000000015F6 |
| 0.4._2.35._10.2.104.6 | tradeserver | 10.2.104.64=>10.2.111.14 | VDIAN_MQ | OK | SEND_order_all_status | 0 | 2/2ms | 0185000000000000283F84B |
| 0.4._2.35._10.2.104.6 | tradeserver | 10.2.104.64=>10.2.137.100 | DUBBO | OK | com.vdian.udc.core.api.RelationApi.map | 229 | 1/1ms | OK |
| 0.4._2.35._10.2.104.6 | tradeserver | 10.2.104.64=>10.2.111.14 | VDIAN_MQ | OK | SEND_trade_message_server | 0 | 0/0ms | 03C000080000000000F27CA9 |
| 0.4._2.35._10.2.104.6 | messageserver | =>10.2.148.85 | VDIAN_MQ | OK | CONSUME_trade_message_server | 0 | 1/1ms | messageserver#03C00008000000000F |
| 0.4._2.35._10.2.104.6 | ▾ flood | 10.2.104.64=>10.2.130.94 | DUBBO | OK | com.vdian.flood.dubbo.ExperimentServi | 165 | 1/0ms | OK |
| 0.4._2.35._10.2.104.6 | flood | 10.2.130.94=>- | REDIS | OK | flood_app_get | 0 | 0/0ms | ok |
| 0.4._2.35._10.2.104.6 | ▾ flood | 10.2.104.64=>10.2.109.55 | DUBBO | OK | com.vdian.flood.dubbo.ExperimentServi | 165 | 1/0ms | OK |
| 0.4._2.35._10.2.104.6 | flood | 10.2.109.55=>- | REDIS | OK | flood_app_get | 0 | 0/0ms | ok |
| 0.4._2.35._10.2.104.6 | tradeserver | 10.2.104.64=>10.2.111.14 | VDIAN_MQ | OK | SEND_trade_message_server | 0 | 1/1ms | 03C000080000000000F27CAA |
| 0.4._2.35._10.2.104.6 | ▾ messageserver | =>10.2.148.85 | VDIAN_MQ | OK | CONSUME_trade_message_server | 0 | 6/6ms | messageserver#03C00008000000000F |
| 0.4._2.35._10.2.104.6 | messageserver | 10.2.148.85=>10.2.111.14 | VDIAN_MQ | OK | SEND_NCENTER_APP.msg | 0 | 4/4ms | 04AD000000000000011E9DCD |
| 0.4._2.35._10.2.104.6 | ▾ flood | 10.2.104.64=>10.2.109.54 | DUBBO | OK | com.vdian.flood.dubbo.ExperimentServi | 165 | 1/0ms | OK |

- Dapper: rpc tracks the basis of the paper
- Rpc framework generation, no business intrusion
- Trace Id: initial call to generate uuid string
- Each rpc call generates a span while recording the parent span
- Full support for other middleware：tomcat/dubbo/vdianMQ/vdds/redisClient

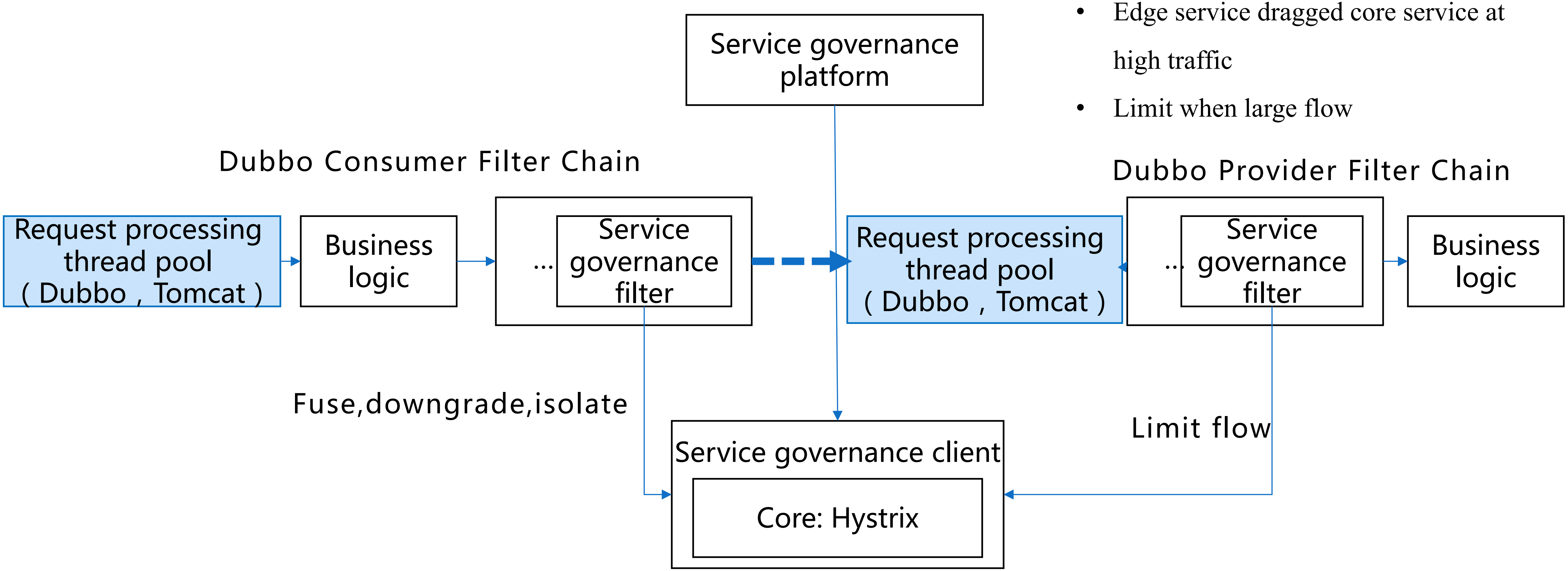
# Servitization-Application service monitoring, alarm



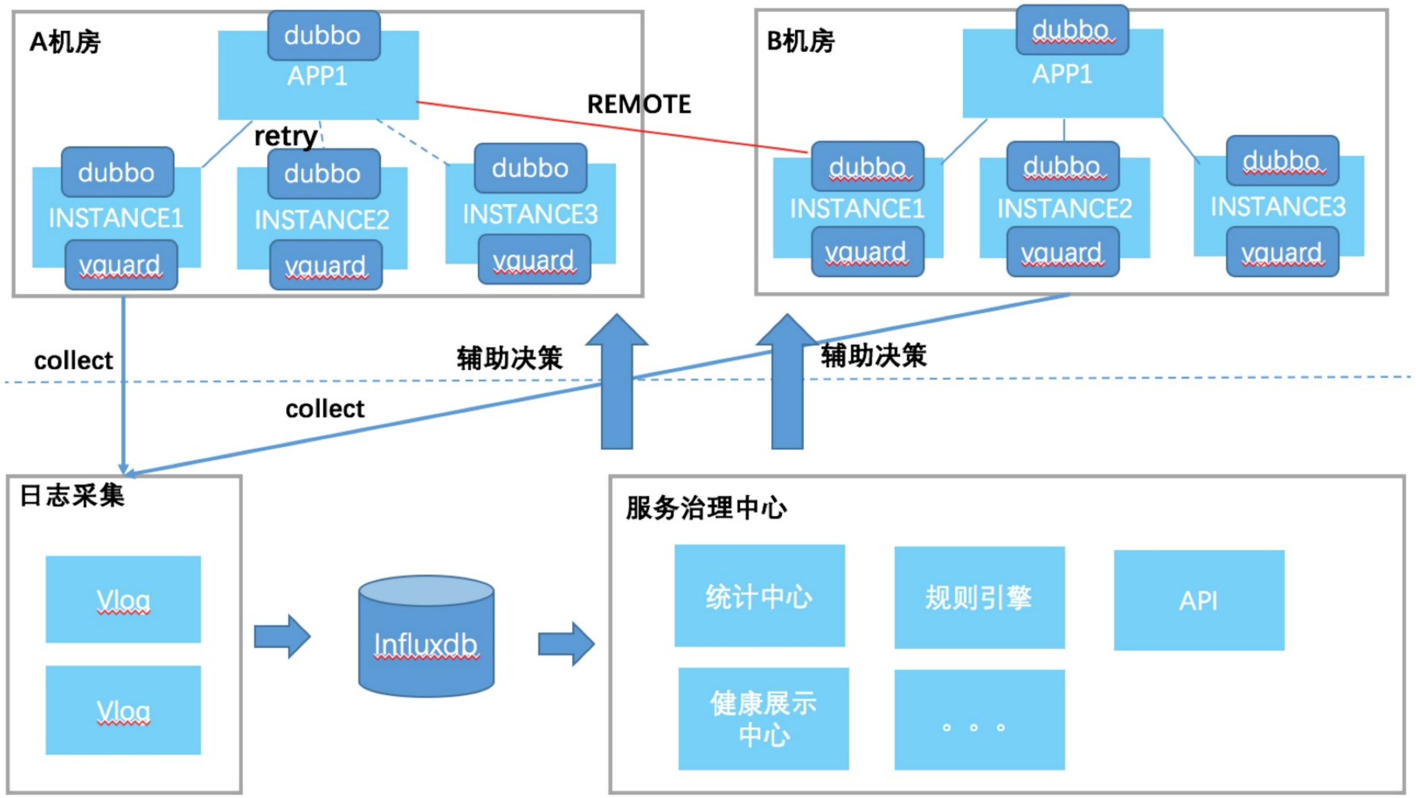
- **Application service monitoring, alarm**
  - Vmonitor agent collects rpc trace logs
  - Alarm rule configuration (traffic exception/service exception super threshold...)
- **Solved problems:**
  - Abnormal traffic warning
  - Error warning

# Servitization-Application fuse, limit flow

- **Solved problems:**
  - Edge service dragged core service at high traffic
  - Limit when large flow


Service governance platform
Dubbo Consumer Filter Chain
Request processing thread pool ( Dubbo , Tomcat )
Business logic
… Service governance filter
Fuse,downgrade,isolate
Service governance client
Core: Hystrix
Dubbo Provider Filter Chain
Request processing thread pool ( Dubbo , Tomcat )
… Service governance filter
Business logic
Limit flow

# Servitization-Business challenge: double alive



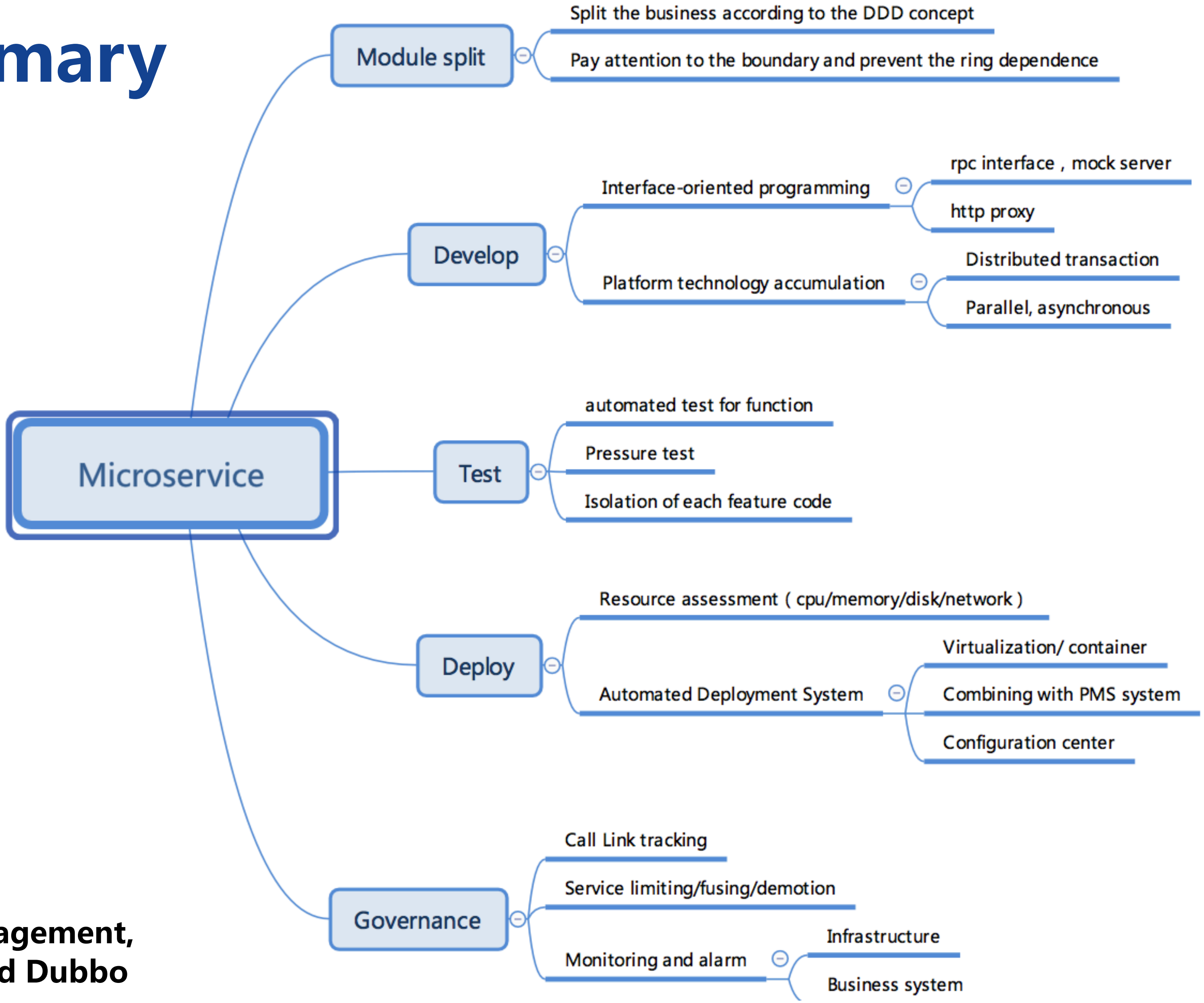
- **Requirement: Computer room migration(Double alive in the same city)**

- **Strategy：**
  - New and old machine room share a Dubbo root root node
  - The service of the new computer room is registered under the same service directory. In the URL, the belongTo attribute needs to be added.
  - If all the local services are down, the router will be routed to the remote execution.

- **Changes：**
  - The Dubbo router policy needs to be re implementeddubbo
  - The timeout retry strategy needs to be reimplemented

# Servitization-Summary

Thanks to Dubbo's support for the business



- Microservice
  - Module split
    - Split the business according to the DDD concept
    - Pay attention to the boundary and prevent the ring dependence
  - Develop
    - Interface-oriented programming
      - rpc interface , mock server
      - http proxy
    - Platform technology accumulation
      - Distributed transaction
      - Parallel, asynchronous
  - Test
    - automated test for function
    - Pressure test
    - Isolation of each feature code
  - Deploy
    - Resource assessment ( cpu/memory/disk/network )
    - Automated Deployment System
      - Virtualization/ container
      - Combining with PMS system
      - Configuration center
  - Governance
    - Call Link tracking
    - Service limiting/fusing/demotion
    - Monitoring and alarm
      - Infrastructure
      - Business system

**Summary**: **using Dubbo to do service management, we should make efforts in all around Dubbo**

# Dubbo and Weidian's Practice on Microservice Architecture

**Speaker : Xin Wang**

**CONTENTS**

- **Who's Weidian**
- **The stages of Weidian**
- **Challenges & Solutions**
- **About the future**

# About the future-Giving back to the community

## How to replace the dubbo version of the private maintenance with the latest dubbo version?

- Serialization compatibility (in the rolling upgrade, the business is guaranteed to be ordered)
- After graduating from the apache community, the package will be changed from com.alibaba.dubbo to org.apache.dubbo and will provide an extension.

## Private dubbo version, how to integrate with the official version of dubbo?

- Please submit your private extension to the dubbo communitydubbo
- Dubbo official will make the best practices in the following areas: rpc tracking, current-limiting fuse, multi-language support, etc.

# **About the future-**Multi-language support

- **Method：**

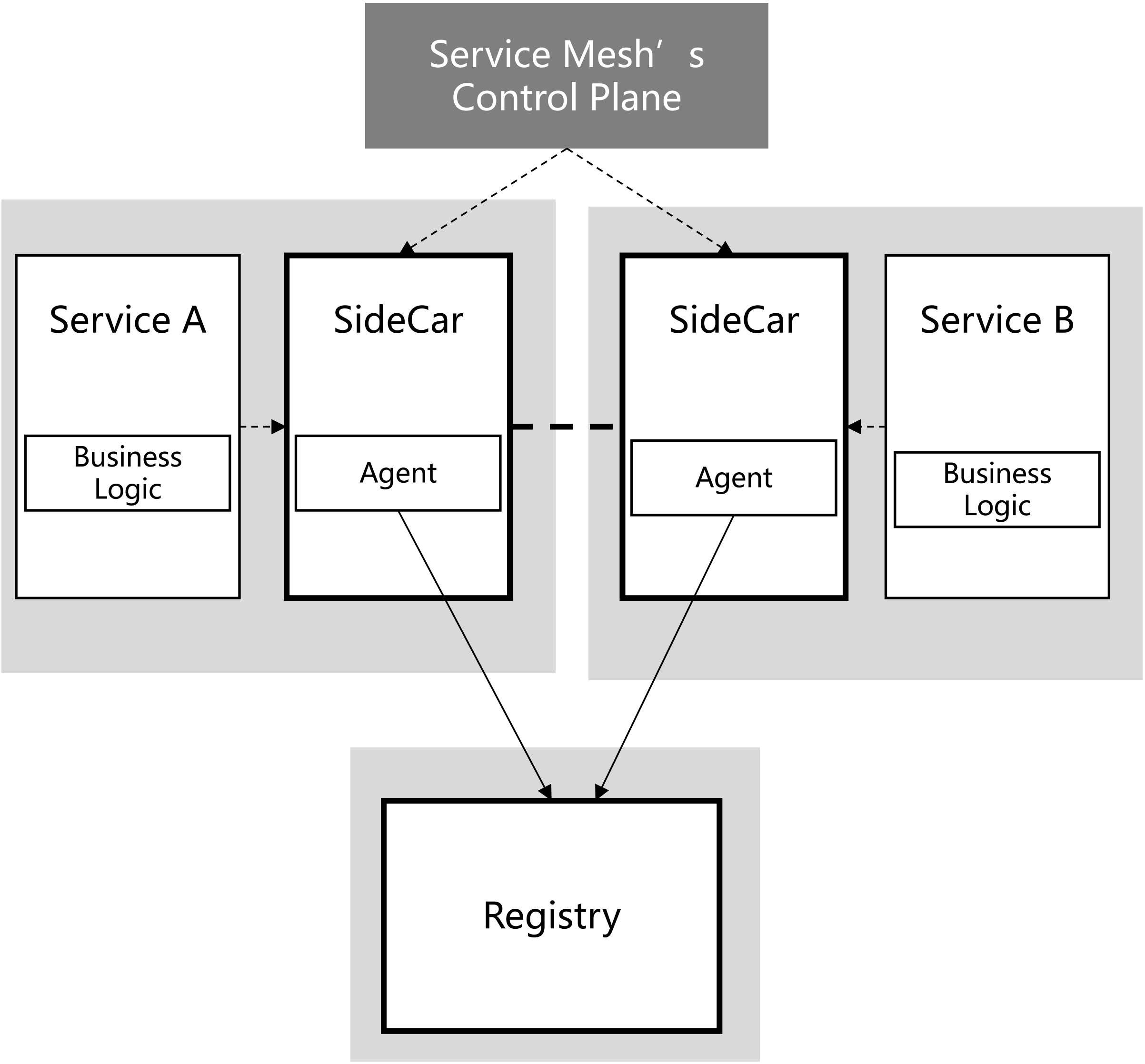
Use a effective  language to develop a agent, other than JAVA language can interact through the agent process and registry. (Service mesh sidecar mode)

## Benefit ：

No need to implement a dubbo client in every language (in multi-language communication, thrift, gRPC and other serialization frameworks based on tcp protocol will perform better than http1+json) serialization

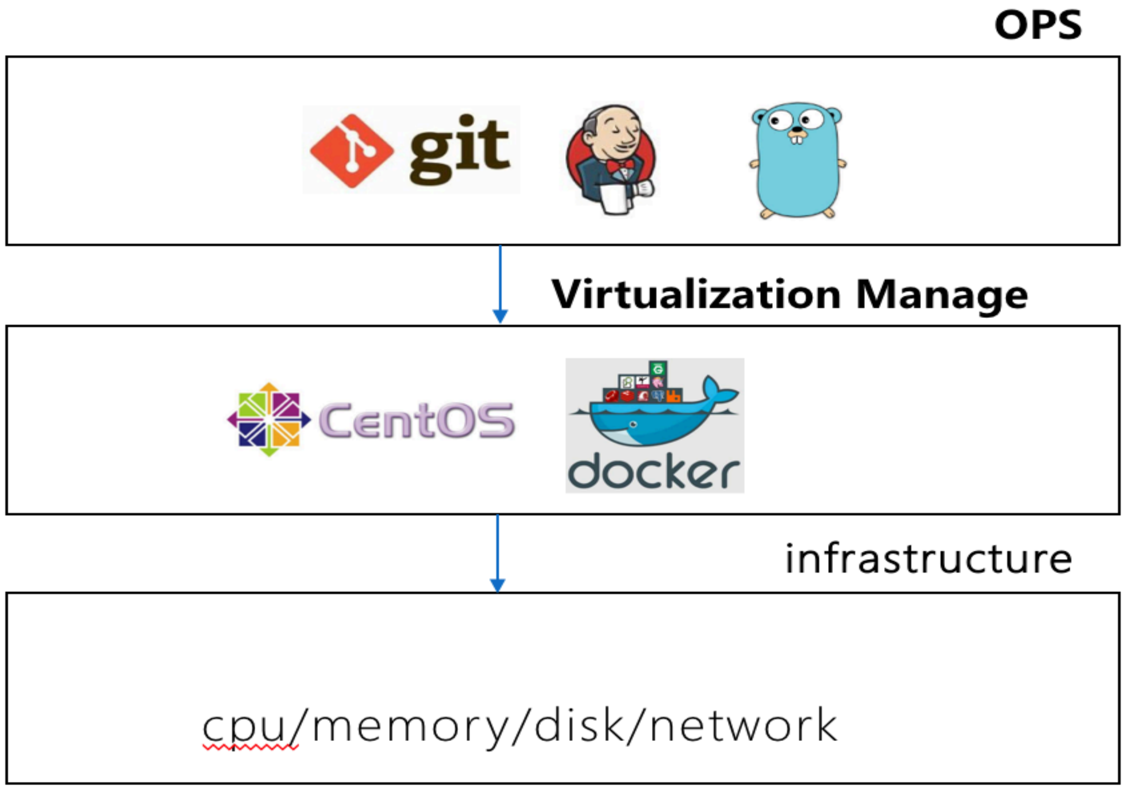
## Problem：

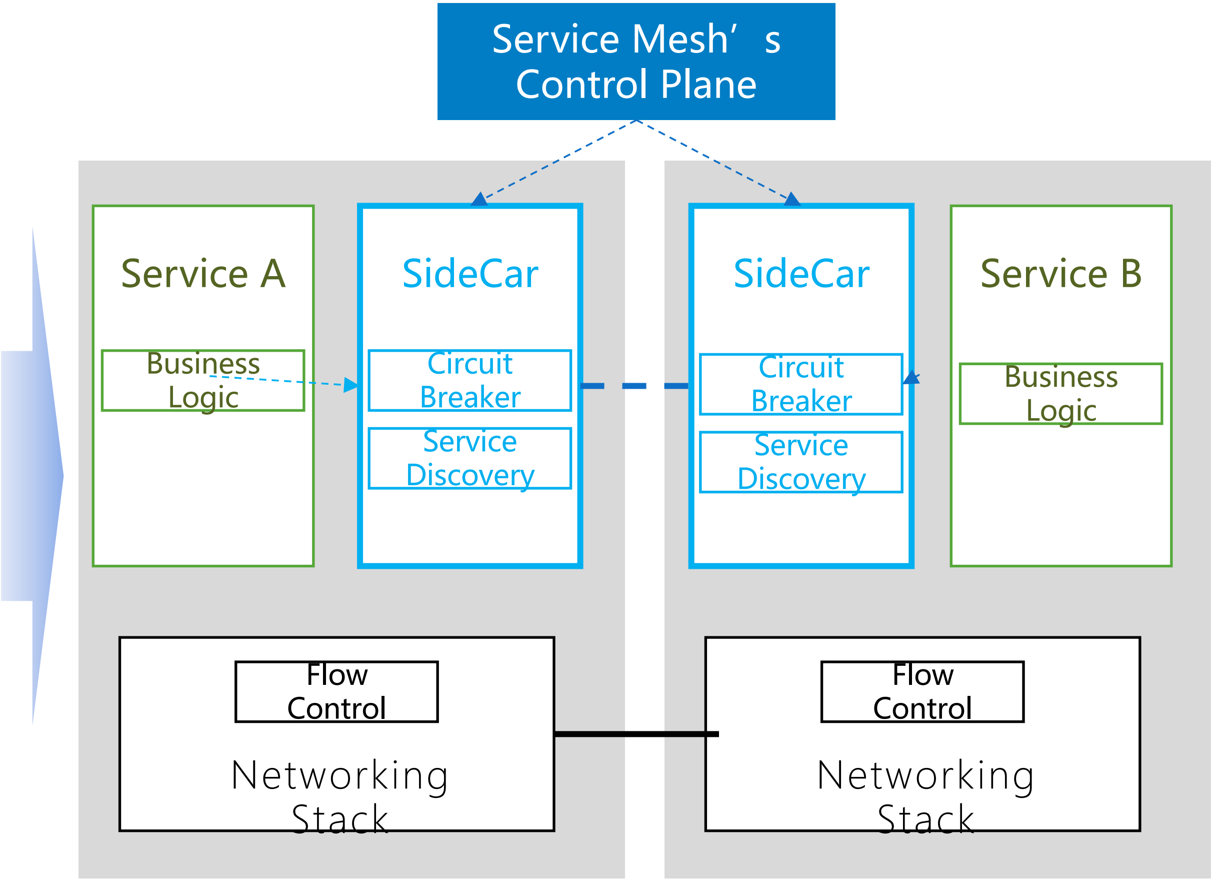
A node has been added: when the agent process is abnormal or hangs, the business process of the agent service has to remove the traffic.



# About the future- Silver Bullet or not，When dubbo meet Service mesh

**View:** 1. For the use of technology, it depends on the stage of business development. 2. For the new technology ecology: Please use some of his ideas, such as sidecar, such as this multi-language client practice





# About the future- Dubbo+Reactive programming

## Principle of Finagle:

1, Service as a function;

2, function is in line with the functional programming paradigm: transparent reference, no side effects, can be combined, can be transformed, etc.;

3, the return value of the function is unified Fututre [Xxx] / Try [Xxx] / Option [Xxx]. Try and so is a Monad Tip: Refer to the basic concept of Monad in functional programming;

4. Based on the above three points, the whole system can be constructed like building blocks;

## Reactive programming :

Microservices need to consider their responsiveness, ie

1. Resilience (in response to failure);

2, scalability (response under different load conditions)

3, message-driven (loosely coupled, through asynchronous message isolation) and responsive programming requirements: message-driven, designed for failure, asynchronous. Therefore, responsive programming is required;

# Contact us

**About author：**

- Github: https://github.com/lovepoem
- twitter: wangxinvictor

**Community：**

- dubbo.apache.org
- dubbo.io

**Repos：**

- github.com/apache/incubator-dubbo
- github.com/dubbo

**Email list**

- dev@dubbo.apache.org