How to provide a reliable ridesharing service

DiDi Chuxing company service reliability assurance team

Ming Hua: huaming@didichuxing.com
Lin Tan: tanlintano@didichuxing.com



About Speakers

- **Ming Hua:**
  - Principal Architect, SRE team, DiDi Chuxing, China
  - Areas of focus:
    - Service reliability
    - Operation automation
  - Email: huaming@didichuxing.com

- **Lin Tan:**
  - Senior Engineer, SRE team, DiDi Chuxing, China
  - Areas of focus:
    - Service reliability
    - Container and cloud platform
  - Email: tanlintano@didichuxing.com

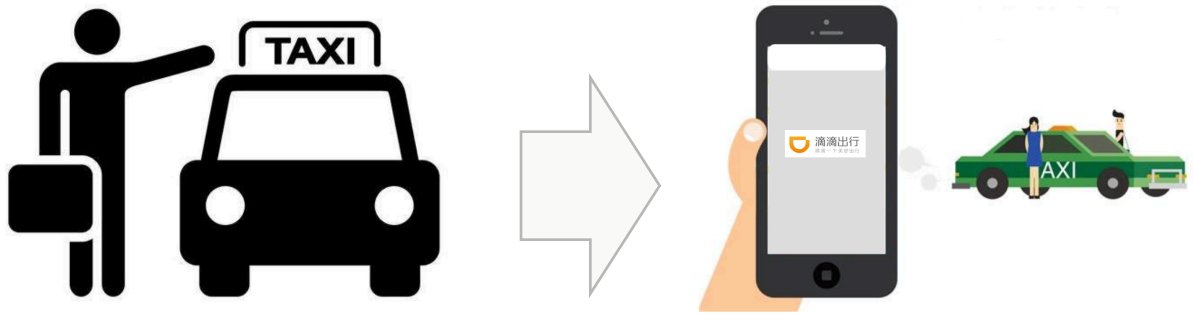
Agenda

- Introduction of DiDi Chuxing
- The challenges of reliability construction in rapidly growing service
- Our technical solutions on service reliability
- Our competition mechanism for reliability work
- What we learned

Introduction of DiDi Chuxing

滴滴
滴滴一下 美好出行

- **Founded in 2012 in China**
  - Experienced about 5 years rapid growth
  - One of the most fastest growing companies in the world.
- **The world's leading mobile transportation platform**
  - 400+ million users
  - 17+ million drivers
  - 400+ cities
- **DiDi Chuxing Services**
  - Taxi
  - Premier car
  - Express
  - Hitch
  - Chauffeur
  - Minibus
  - Bus
  - Test Drive
  - Enterprise
  - Car Rental



Challenges of service reliability in DiDi Chuxing



Challenges of service stability construction in the early stage

- **System overload**: 500% annual requests growth
- **Release risk**: more than 400 releases per day without standard process and environment
- **Vulnerable infrastructure**: Single-cluster architecture
- **Lack of stability improvement**: Missing a measurable mechanism

Our Solutions



Full-link stress test

A specific ridesharing service stress test for system capacity evaluation and bottleneck location.



Standardization

The standardization practices on configuration management、service monitor、program delivery.



Multi-cluster service

A high availability system which across different region zones, and a load balancing strategy bases on cities.



Competition Mechanism

A special mechanism for system reliability and all the works of the team.



Full-link stress test

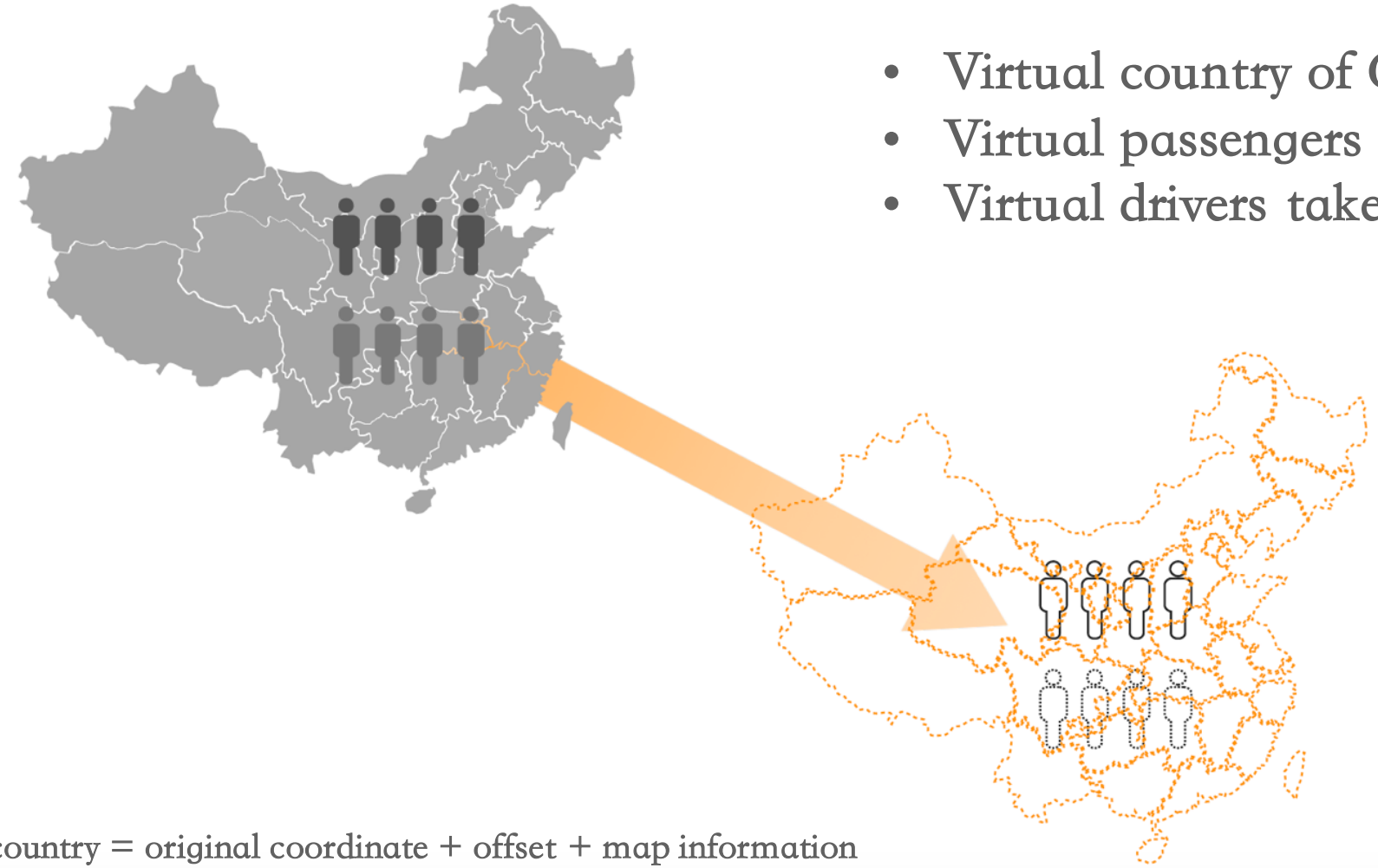
Full-link stress test

- **Purpose:**
  - Service capacity evaluation
  - Bottleneck location

- **Challenges:**
  - Data separation
  - Data accuracy

- **Strategy:**
  - Simulation
  - online test

Stress test - strategies for simulation



- Virtual country of China in the Pacific Ocean
- Virtual passengers send virtual orders
- Virtual drivers take virtual orders

Virtual country = original coordinate + offset + map information
Virtual passenger/driver = original id + offset + passengers/drivers information(some attributions were also made offset)

Stress test - virtual data logic isolation

Reasonable offset and range make sure don't overlap with any real data

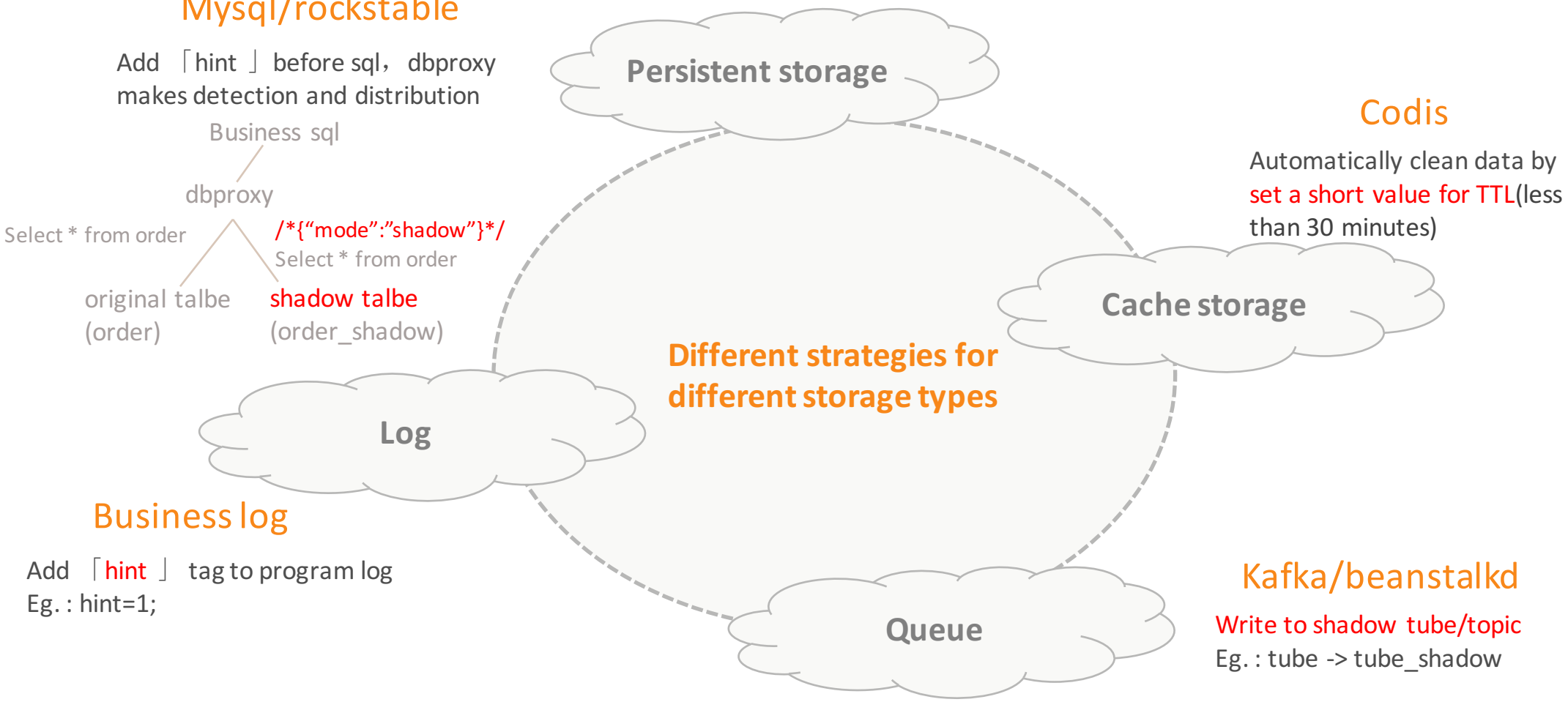
Virtual data	construction	comment
City ID mapping	ID + 10000	Not overlap with any normal city ID
Coordinate mapping	Lng - 230.80078 Lat - 59.63827	Locate in Pacific Ocean Not overlap with any normal country
Passenger and driver ID	ID + 140737488355328	|<- uid: 64 bits ->| |role id: 16bits| p or d id: 48bits |
Virtual passenger phone number	11100020000-11169999999	About 68 million
Virtual driver phone number	11170000000-11199999999	About 30 million
Virtual order ID	High 8 bits is non 0	Normal id range is enough to use for million years

Stress test - tag for test traffic

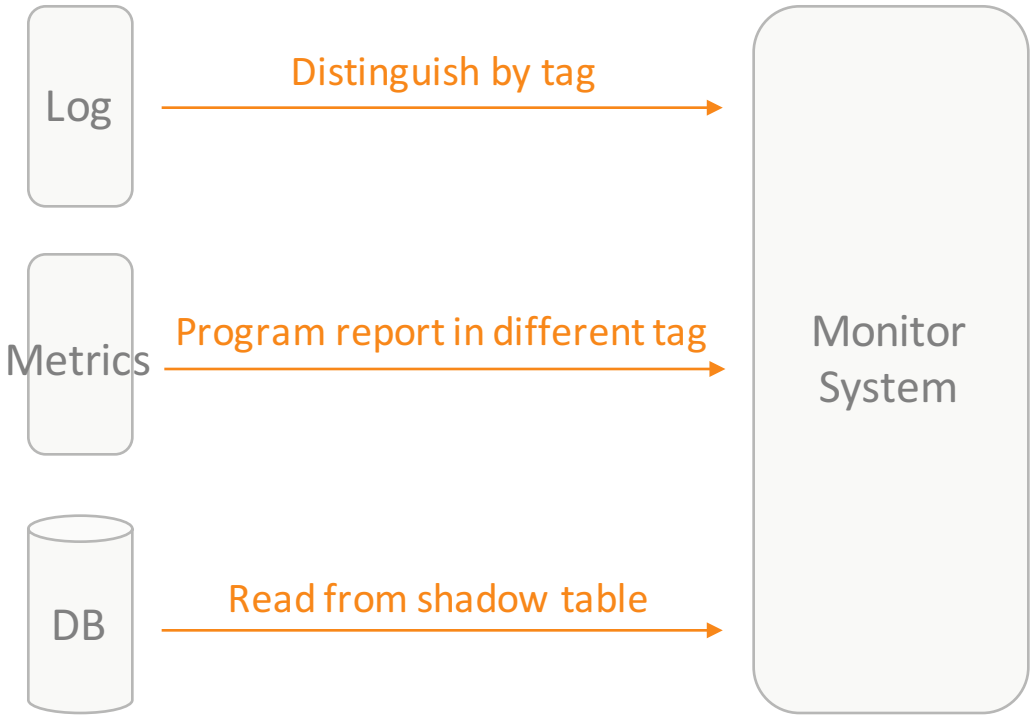
HTTP communication	Thrift communication	Database request	Business traffic

Add hint-code=1 to http header
 	
didi-header-rid:$traceid;
didi-header-spanid:$spanid;
didi-header-hint-code:$hintCode;
didi-header-hint-content:$hintContent;
	Add hintCode to message structure
	
	struct trace {
	1: required string logid;
	2: required string caller;
	3: optional string spanid;
	4: optional string srcMethod;
	5: optional i64 hintCode;
	6: optional string hintContent;
	}
		Add hint tag before sql
		
		/*{"mode":"shadow"}*/ $sql
			Distinguish by specific range
			
			cityid;
			passengerid;
			driverid;
			phone;
			orderid;

Stress test - data storage and cleaning



Stress test - monitor support





Stress test - traffic generation

Online stress test and implement during low peak period of the business



virtual customer orders:
tens of thousands per minute



simultaneous online drivers:
several millions



Standardization

Standardization

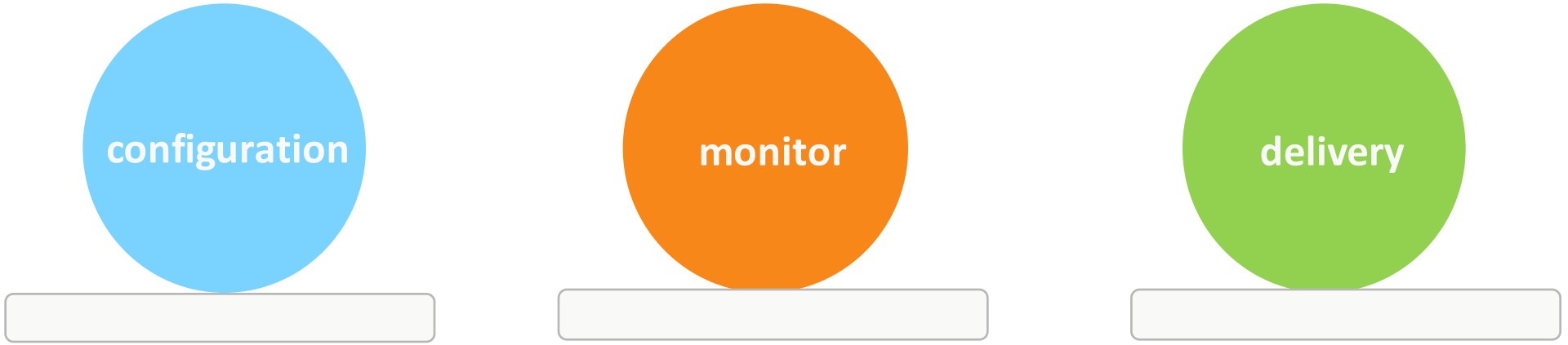
- The benefits of a standardization environment



- The necessary of standardization in DiDi （3H）
  - Hard to handle service relationship
  - Hard to do trouble-shooting
  - High risks of misoperation and delivery accident

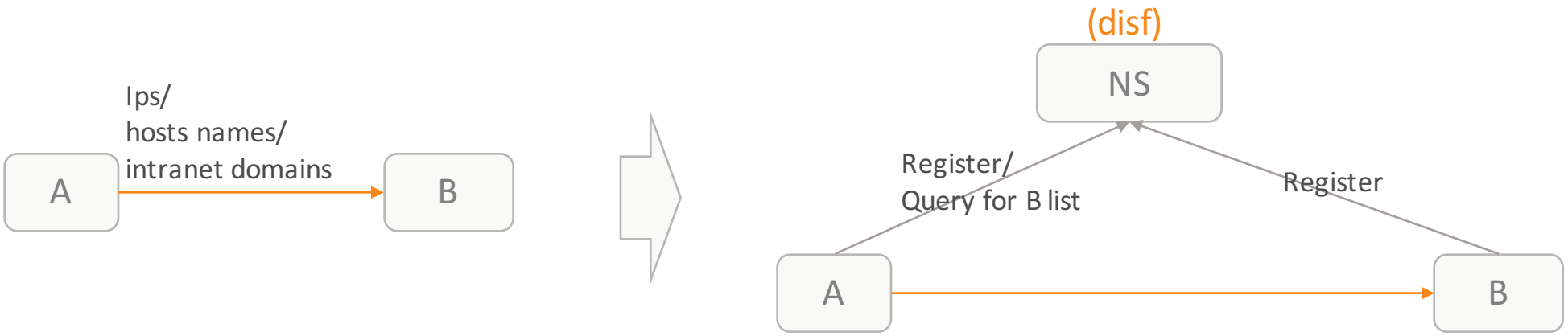
Our Standardization works

Focus on three fields :



Standardization for Configuration

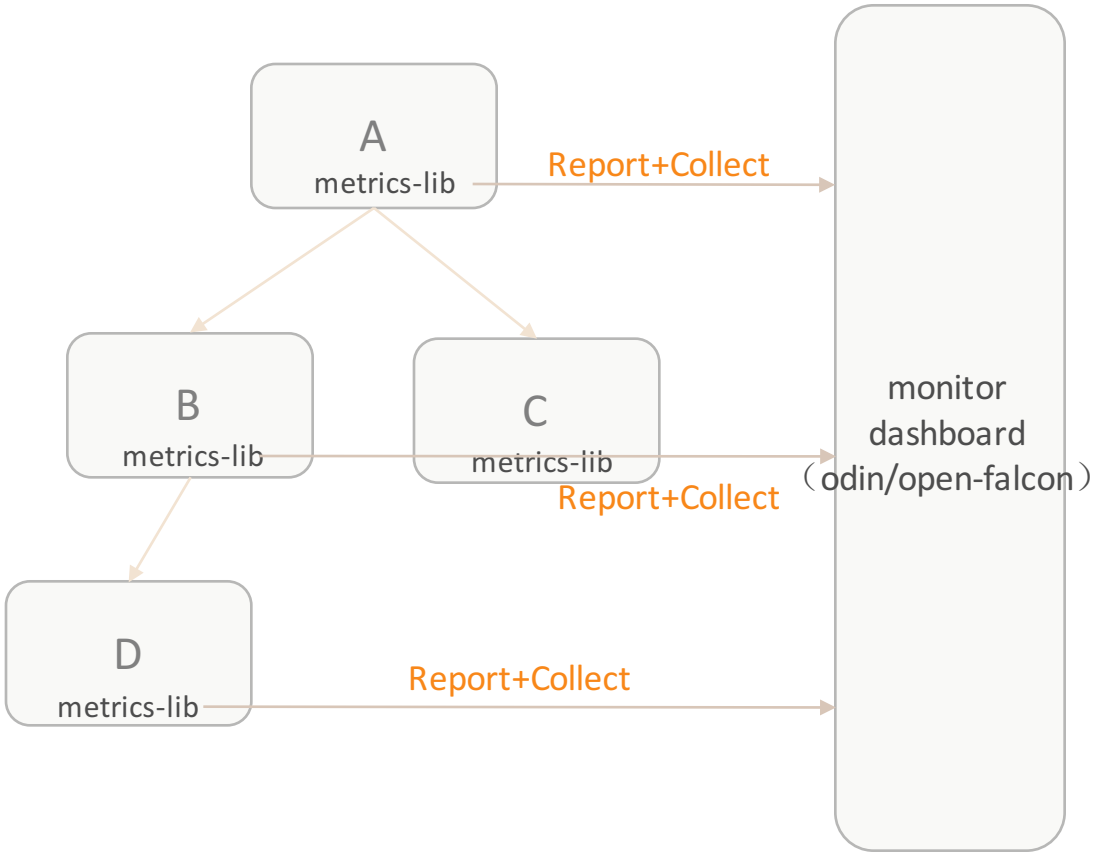
Connection relationship management standardization - disf



- Relay on IP： hard to manage and change
- Relay on NS： centralized management and easy to change

Standardization for Monitor

Principle： all programs involve metrics-lib and report their own status



Every business have a health dashboard



Standardization for Delivery

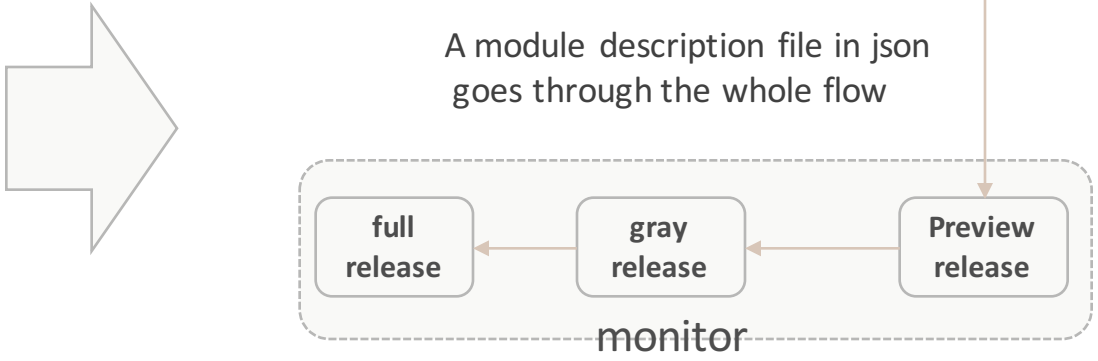
Man dependent => fully-automatic workflow:

- blue/green deployment
- necessary pause in every step
- standard deployment path and backup
- standard program control interface: control.sh start|stop|reload

Semi-automatic



Fully-automatic workflow



Result

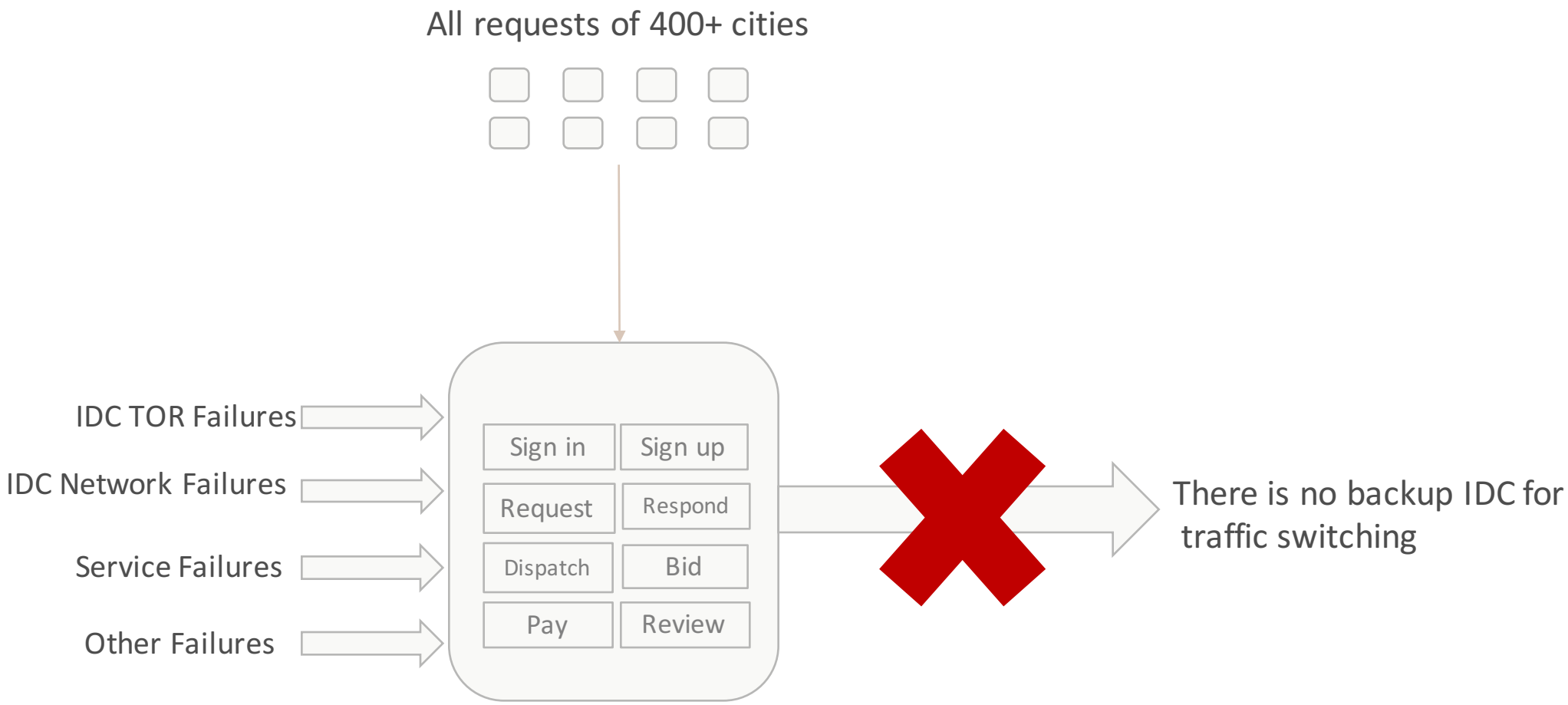
- Frequent service crash caused by misoperation and delivery have been well controled.
- Accident alarm and fault location takes less time.



Multi-cluster service

Multi-clusters service

It often takes a long time to recover even a slight fault occurs

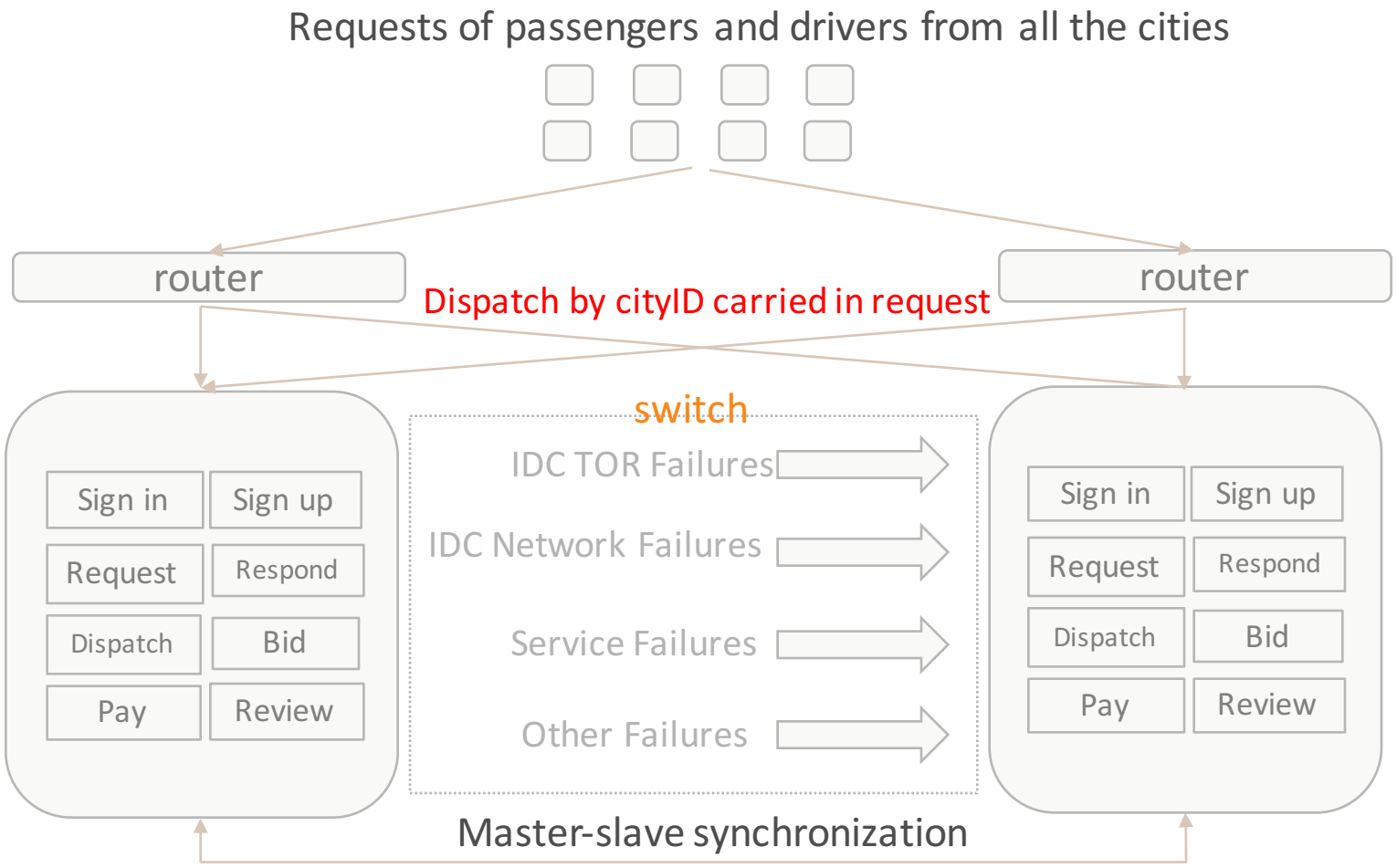


Multi-clusters challenges

- The difficulties of building multi-clusters for ridesharing service
  - Most of modules are stateful;
  - Consistency requirement for data;
  - Both driver-client and passenger-client are location-aware;

Multi-cluster implementation

All factors related to a region are closed to one unit/cluster



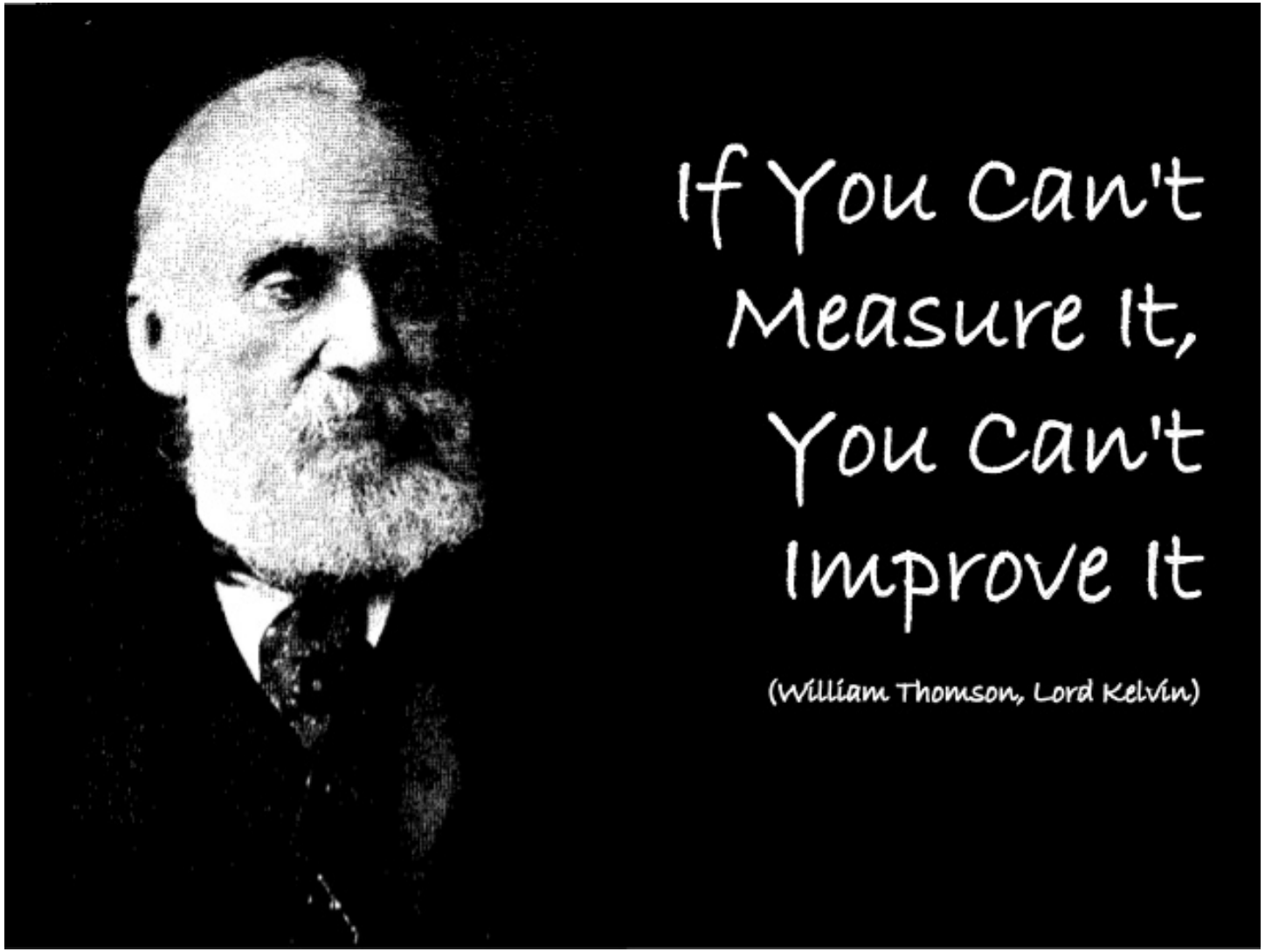
Key solutions:

- All data of drivers and passengers and others was distinguished by citiy, and was split to more than one copies.
- Each cluster has partial data and a copy of other cluster.
- Dispatching requests by cityID in the request entrance.



Competition Mechanism

The Principal Rule



Our competition mechanism for reliability work

A competition named 『**starflower**』 :
A mechanism for ensuring all teams to invest in reliability work



Starflower - Competition Rules

Judge: Organizing Committee (OC)

Participants: All the reliability related teams (across nine business units and the technical departments)

Goal:

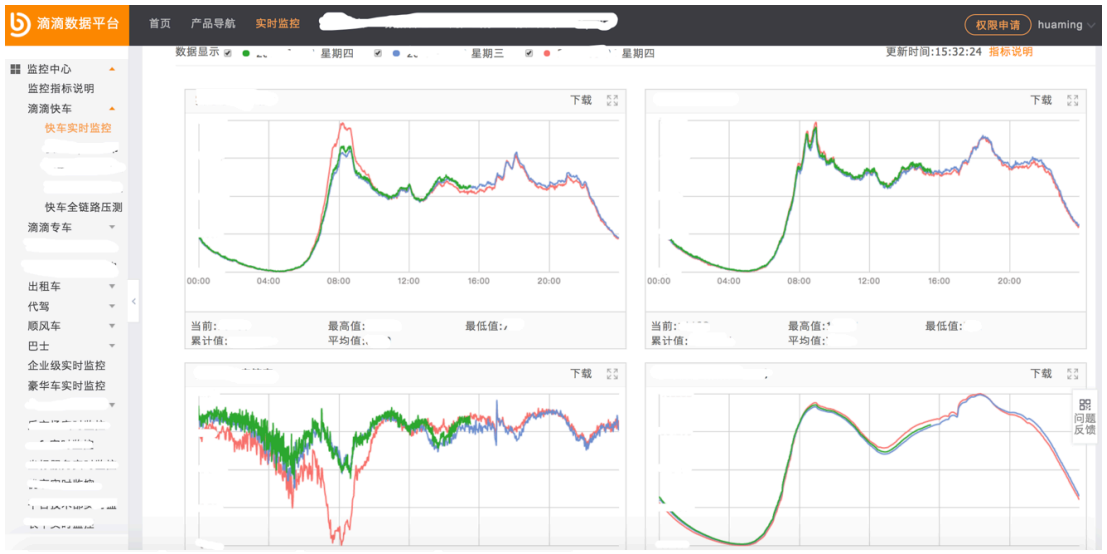
- Server reliability :
  - 2016 : 99.9% downtime per month less than 40 minutes.
  - 2017: 99.95 downtime per month less than 22 minutes.

Rule:

- Downtime:
  - The time when the core performance indicators (request, respond, pay) was decreased by 10% * the affect orders / the total number of weekly orders.
- Quota:
  - The downtime available was assigned to each participants as their reliability quota.

Starflower - exception handling

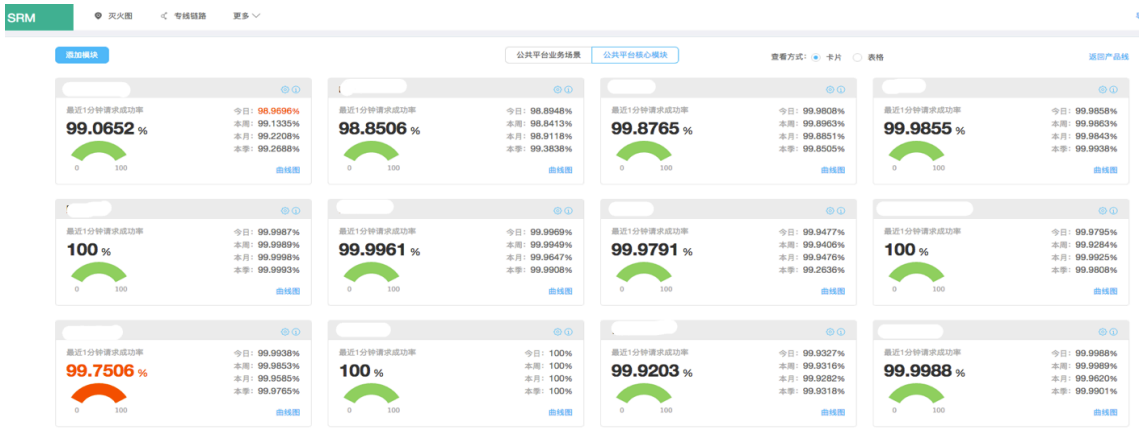
Core Indicators monitoring: Phone call in urgent case

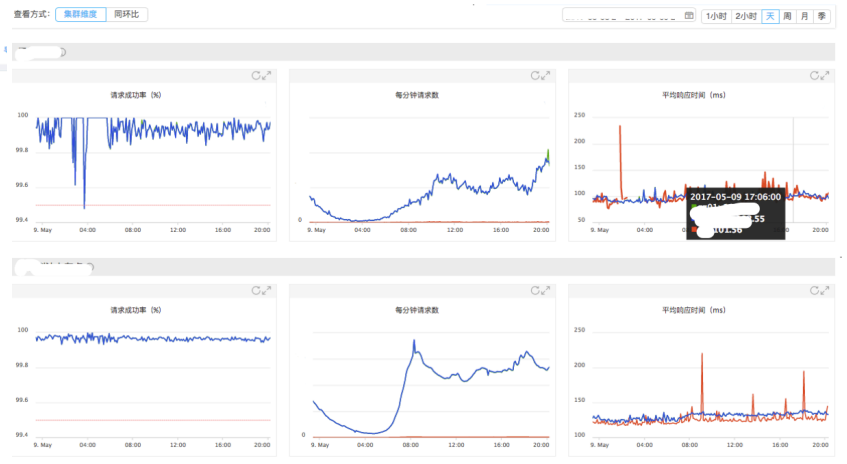


Firemen Group: Communication channel



Fire Map

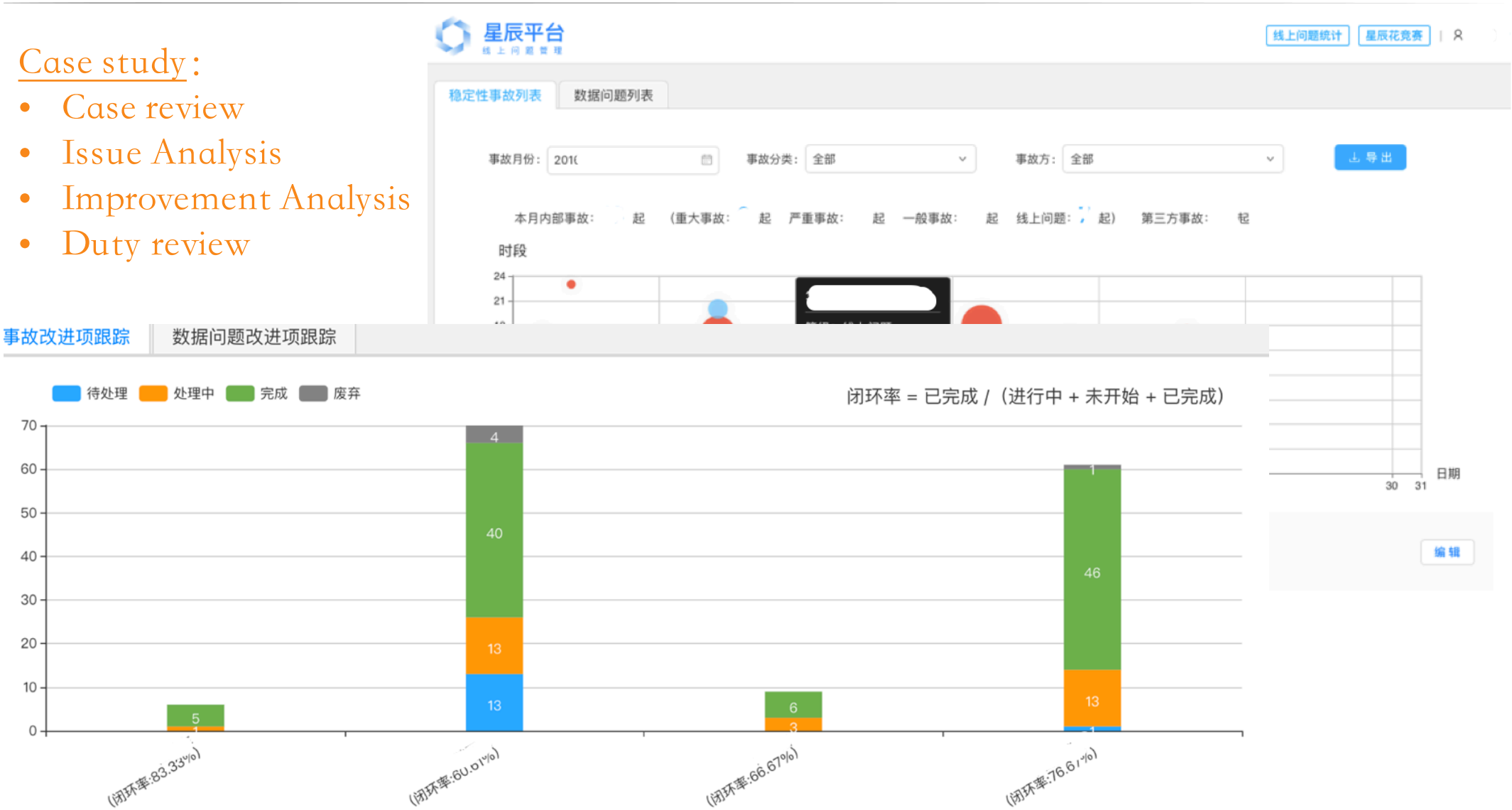




Starflower - case follow

Case study：

- Case review
- Issue Analysis
- Improvement Analysis
- Duty review



Starflower - Rewards and punishments

Failure level:

- p1-p5 according to the severity of the failure

Punishment:

- 1,0000 RMB (about $1400) penalty for P1
- Summary conference every month
  - The teams who held their quota and done well jobs in reliability will be rewarded in money and honor titles;
  - The teams who broke their quota will be punished by money and negative titles;
  - CEO、CTO or other VPS will be the presenter;





Result and the key points

All the teams and the whole technical department regard stability as the most important work

Results:

- ✓ 99.9% 2016
- ✓ 99.95% 2017(up to now)

Key points:

- ✓ The judge should not be the competitor
- ✓ Measurable
- ✓ Clear rules
- ✓ Punishment and reward
- ✓ Continuous follow-up



What we learned

What we learned



The early the better



Collaborative working



Long term investment

Thanks!

