NACOS.

New Features and Design Principles of Nacos 1.0.0

Pengfei Zhu (Github ID: nkorange) | 1st Meetup of Nacos at Hangzhou, Zhejiang | 13 Apr 2019

Contents

The New Nacos 1.0.0

The Design Principles of Nacos Naming

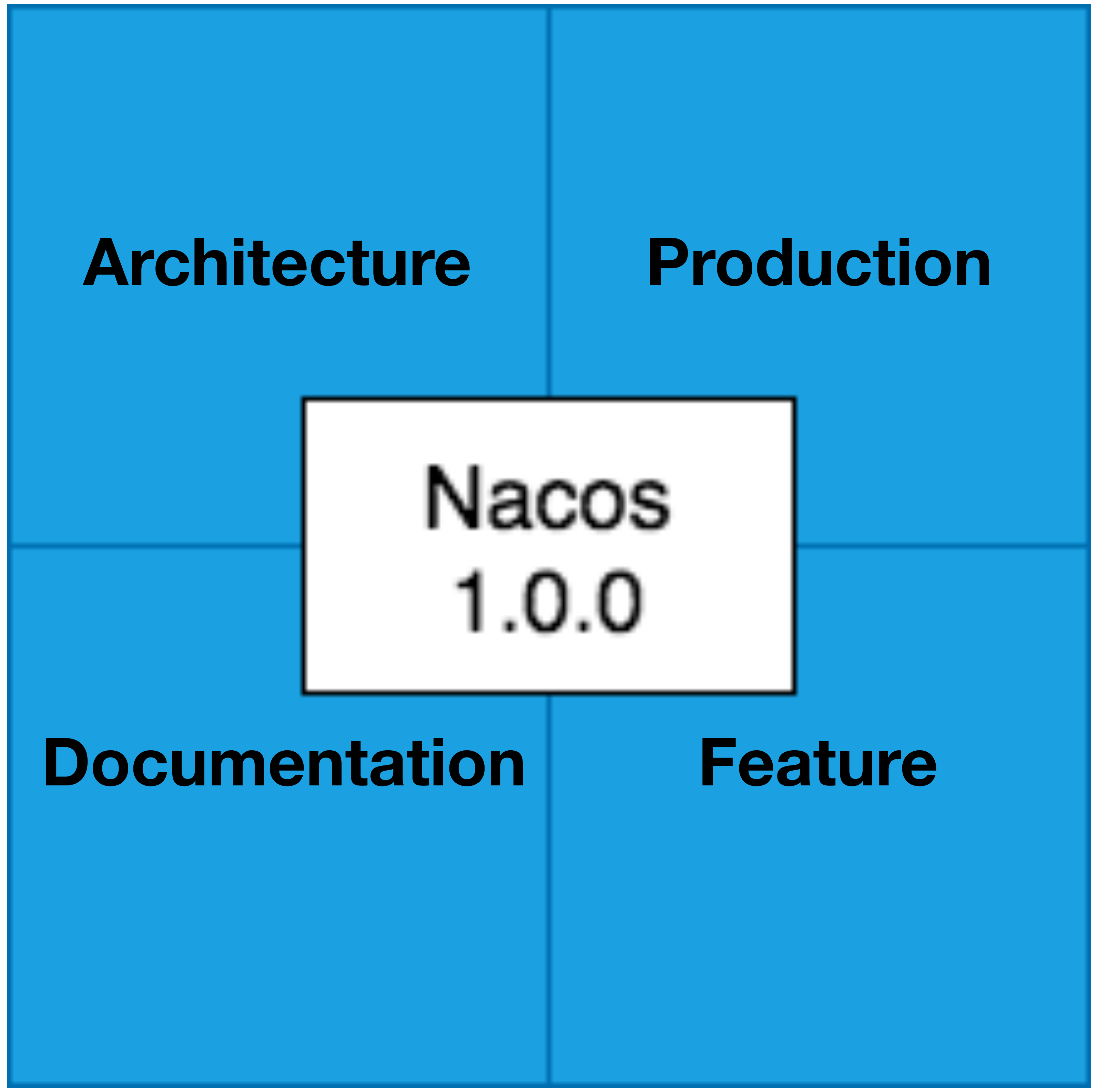
The Coming Features

NACOS.

Part I

The New Nacos 1.0.0

A global view of Nacos 1.0.0



Nacos 1.0.0

Architecture

- Symbiosis of AP and CP

Production

- Data warm up
- Server status control
- Stress test for large scale services

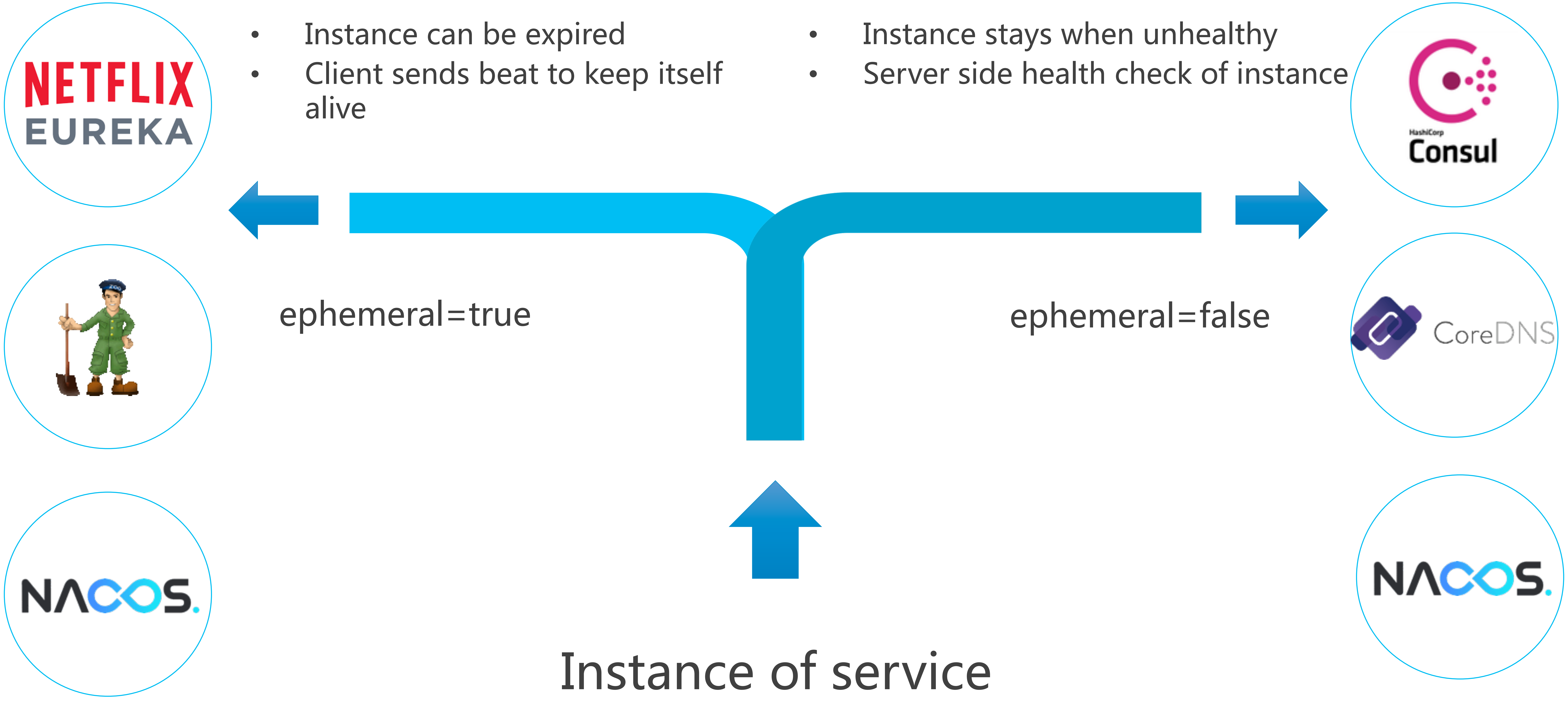
Documentation

- Complete list of API
- Insight of architecture
- Benchmark results

Feature

- Ephemeral instance
- Service group name
- MySQL 8.0 driver support

Support ephemeral property of instance



- Instance can be expired
- Client sends beat to keep itself alive

ephemeral=true

- Instance stays when unhealthy
- Server side health check of instance

ephemeral=false

Instance of service

Stress test results released

High Capacity

- 1000,000 instances
- 1000,000 configurations
- 1000,000 clients

High Scalability

- Extend to over 100 nodes

High Concurrency

- Write TPS over 10,000
- Read QPS over 10,000

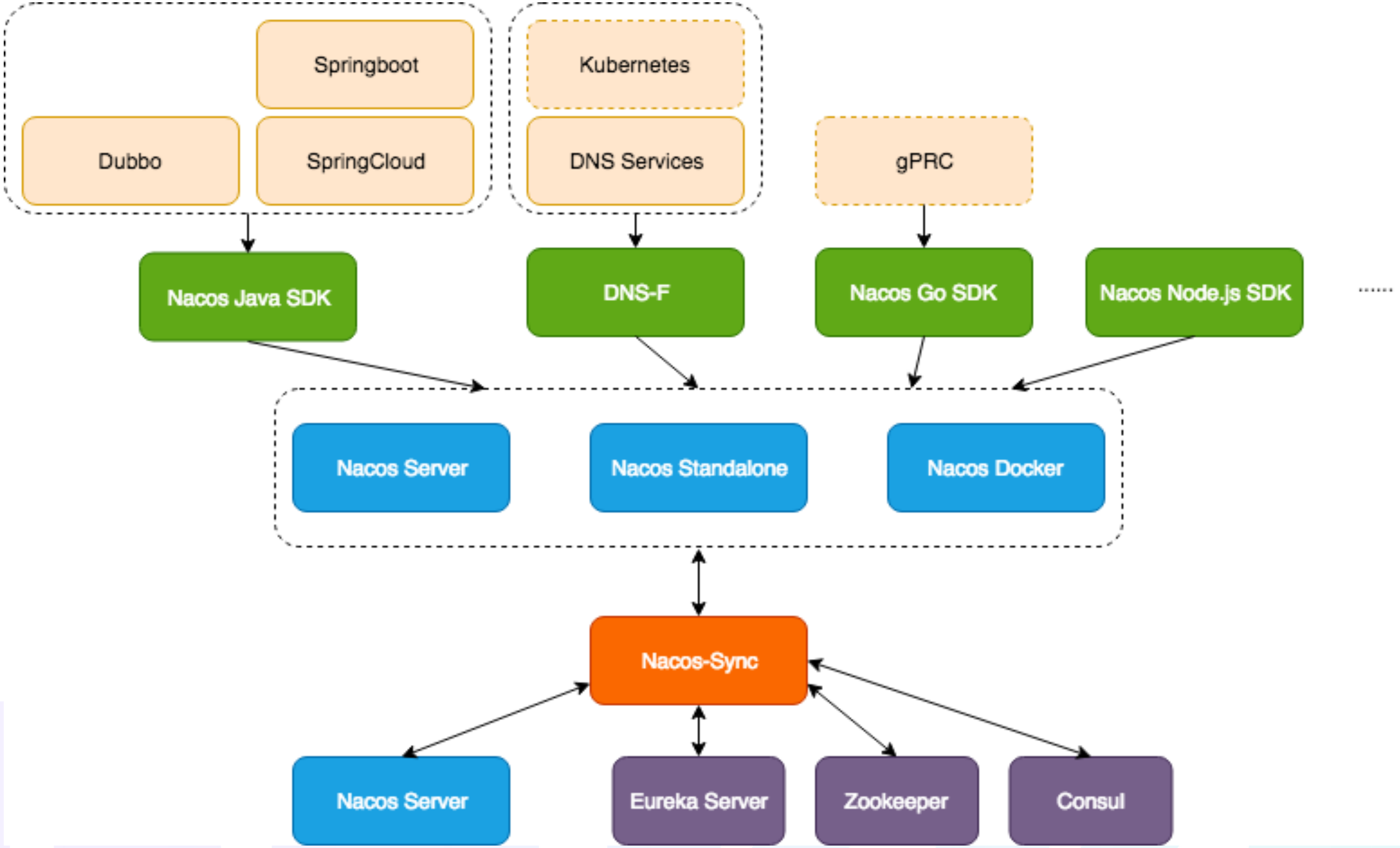
Low Latency

- 99.9% of 10,000 clients receive push in 3 seconds

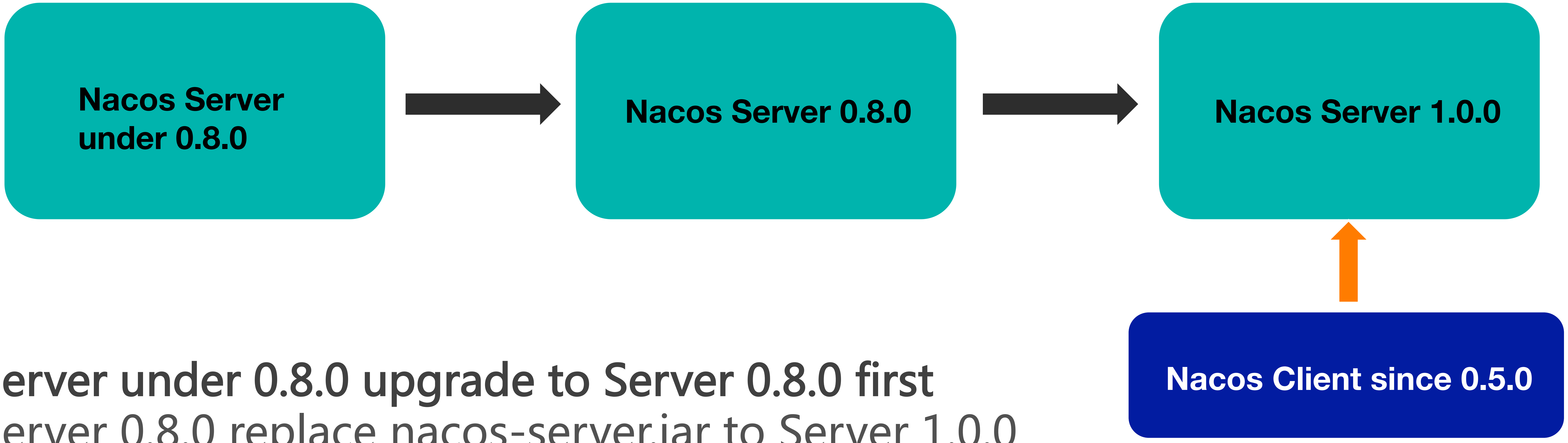
Complete reports available:

- https://nacos.io/en-us/docs/nacos-config-benchmark.html
- https://nacos.io/en-us/docs/nacos-naming-benchmark.html

One registry center for all



Upgrade to Nacos 1.0.0

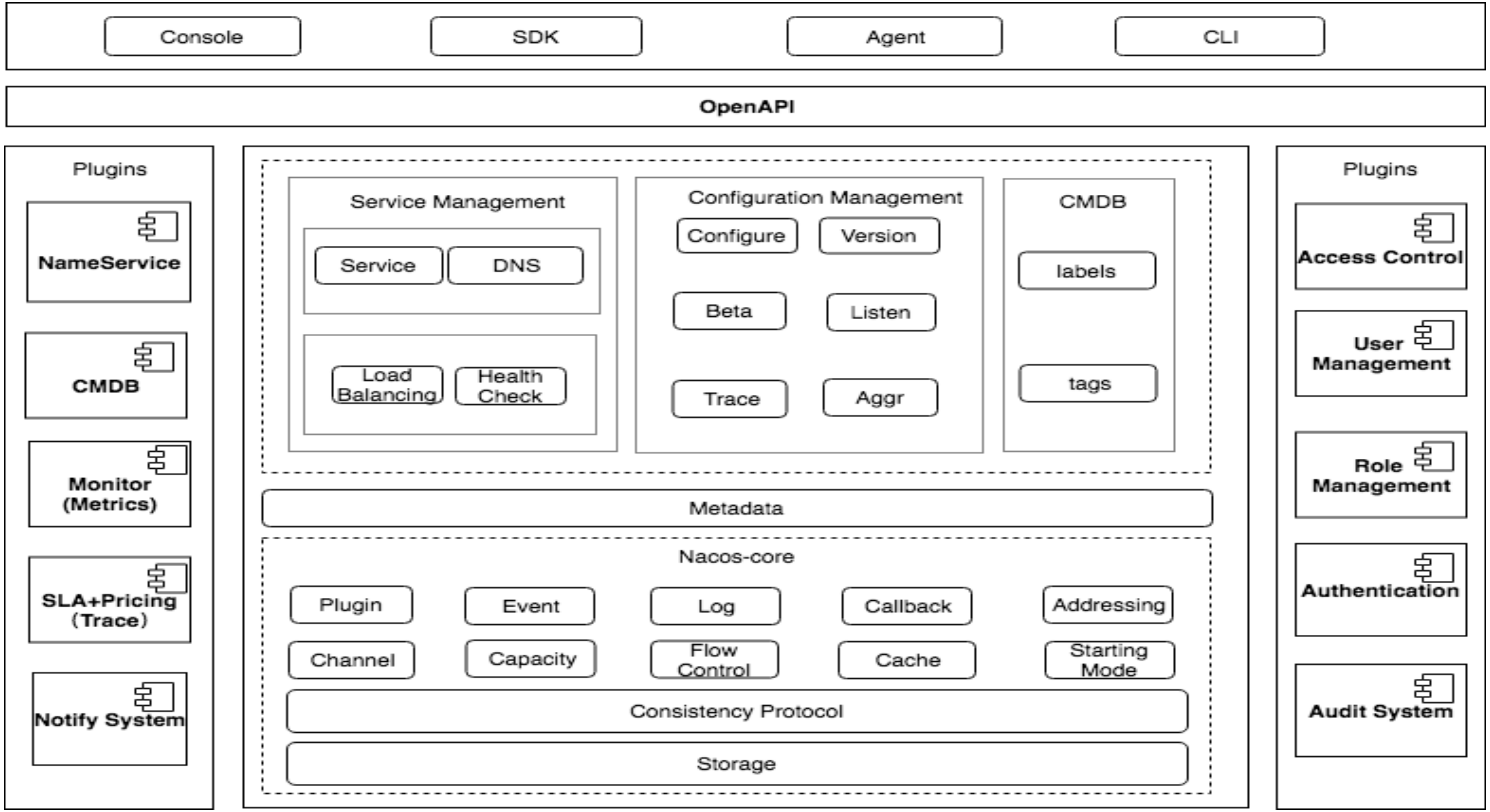


- **Server under 0.8.0 upgrade to Server 0.8.0 first**
- Server 0.8.0 replace nacos-server.jar to Server 1.0.0
- Client compatible to 0.5.0

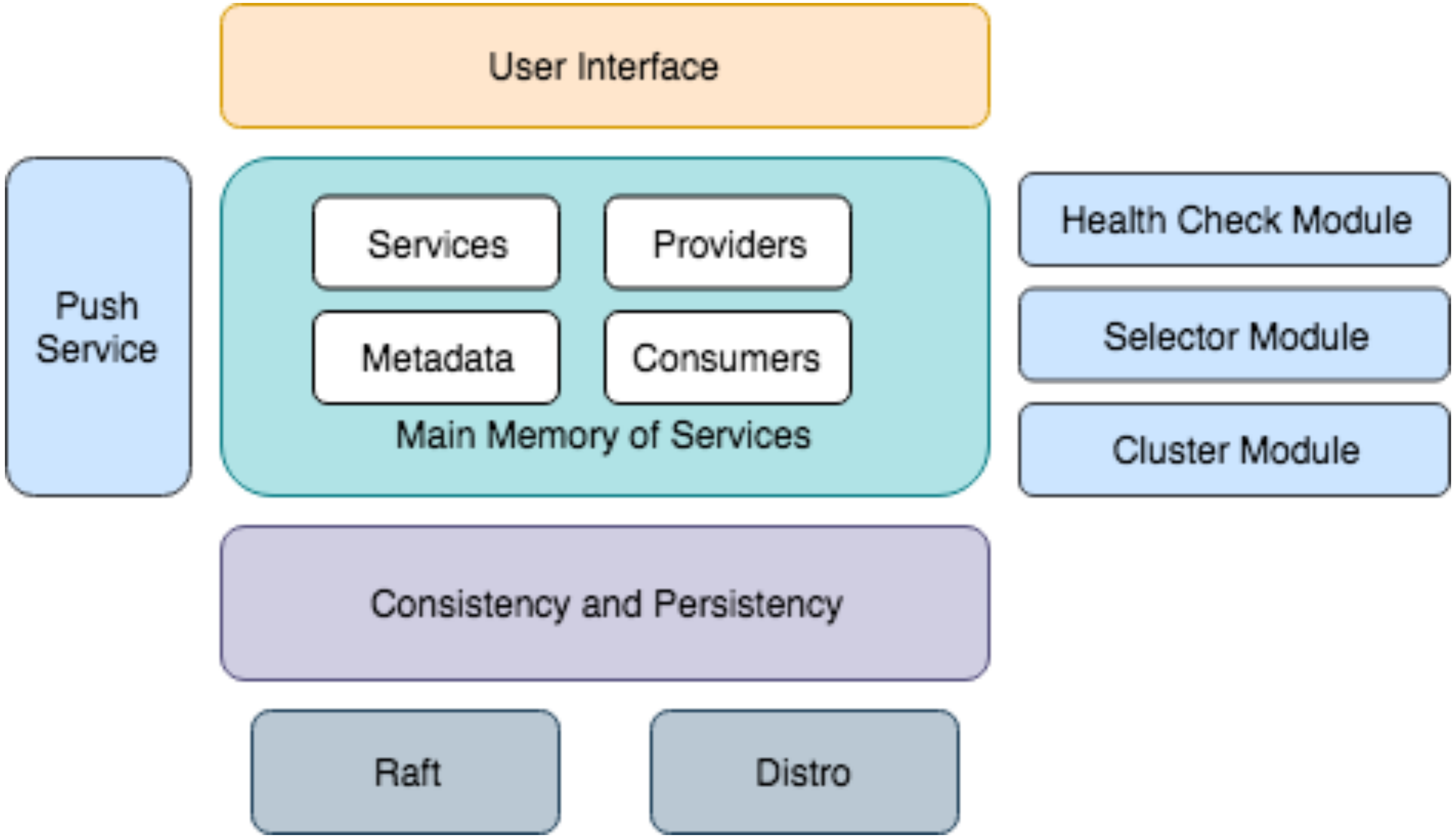
Part II

The Design Principles of Nacos Naming

Modules of Nacos



Architecture of Naming



User Interface

Push Service

Services

Providers

Metadata

Consumers

Main Memory of Services

Health Check Module

Selector Module

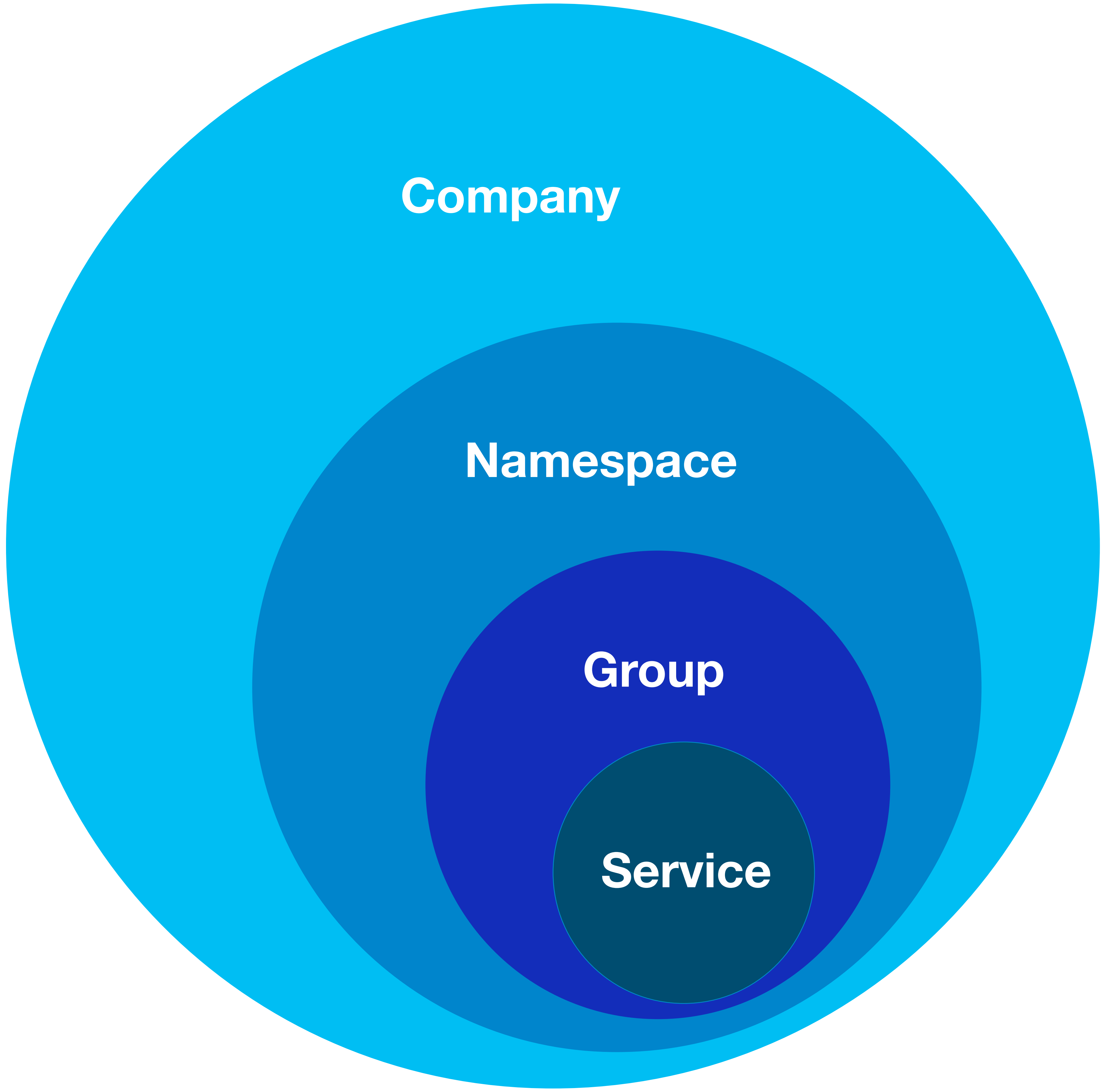
Cluster Module

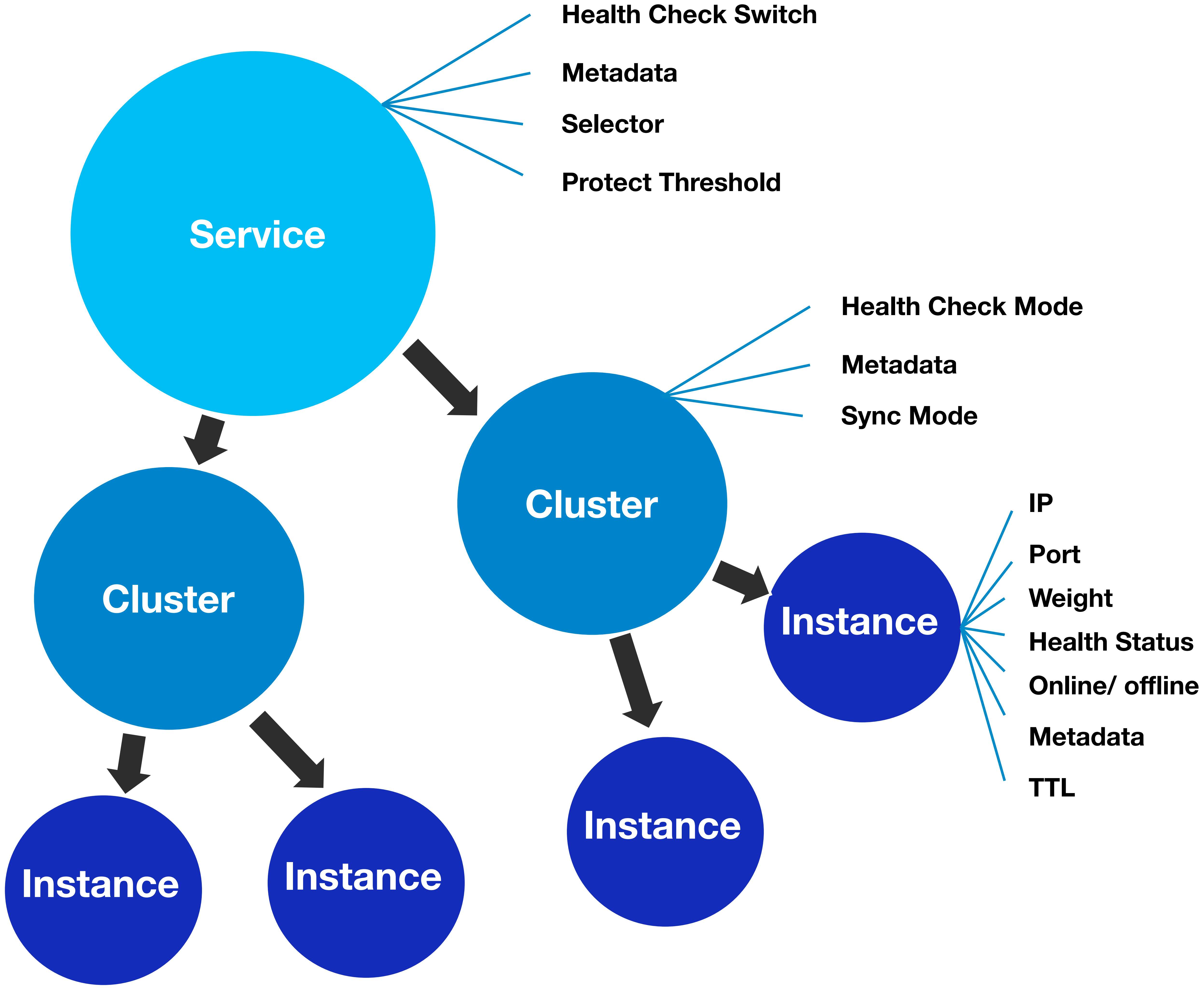
Consistency and Persistency

Raft

Distro

Data model of Naming

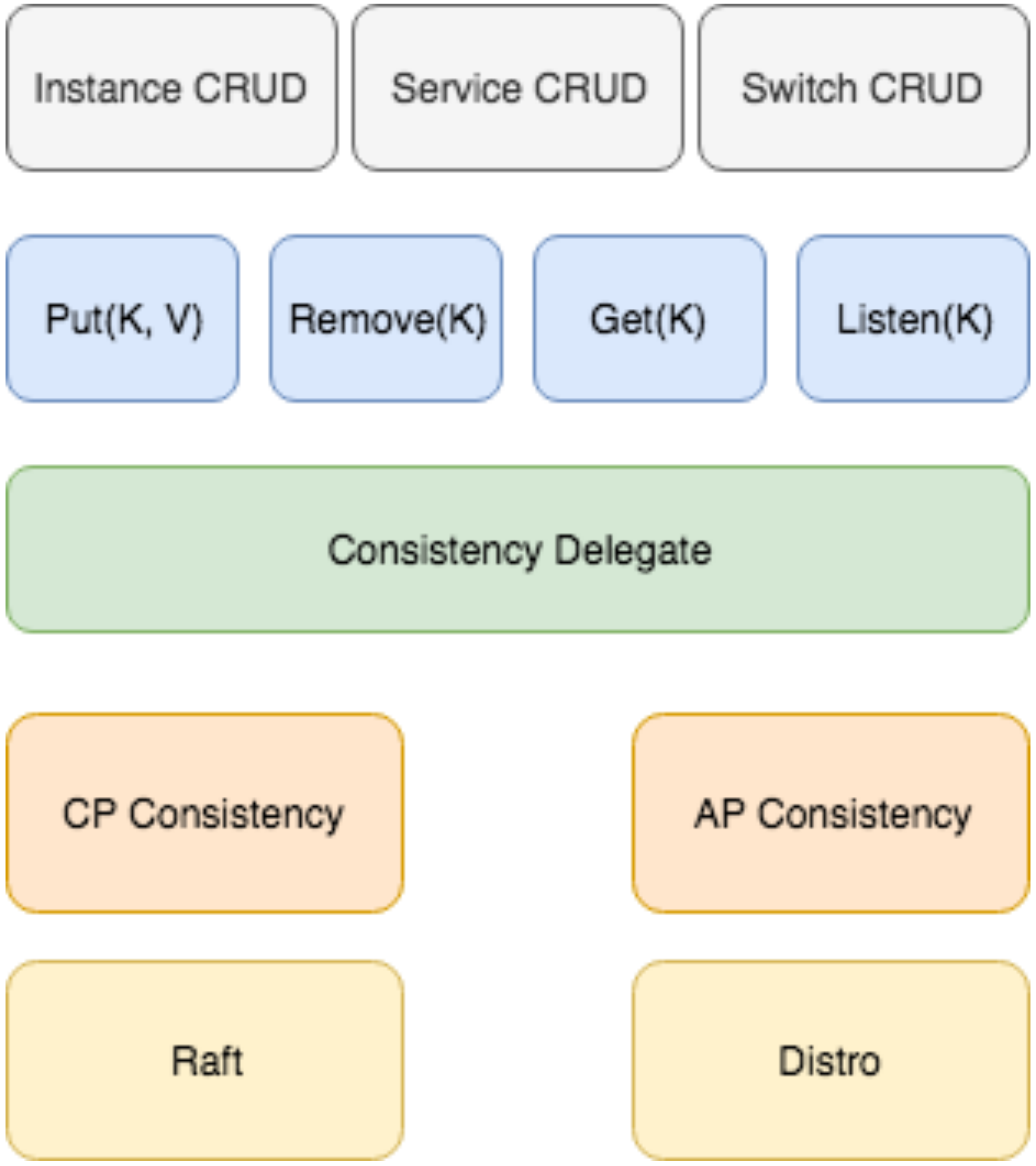




- Four levels of data isolation between services
- Three levels of data customization inside a service

NACOS.

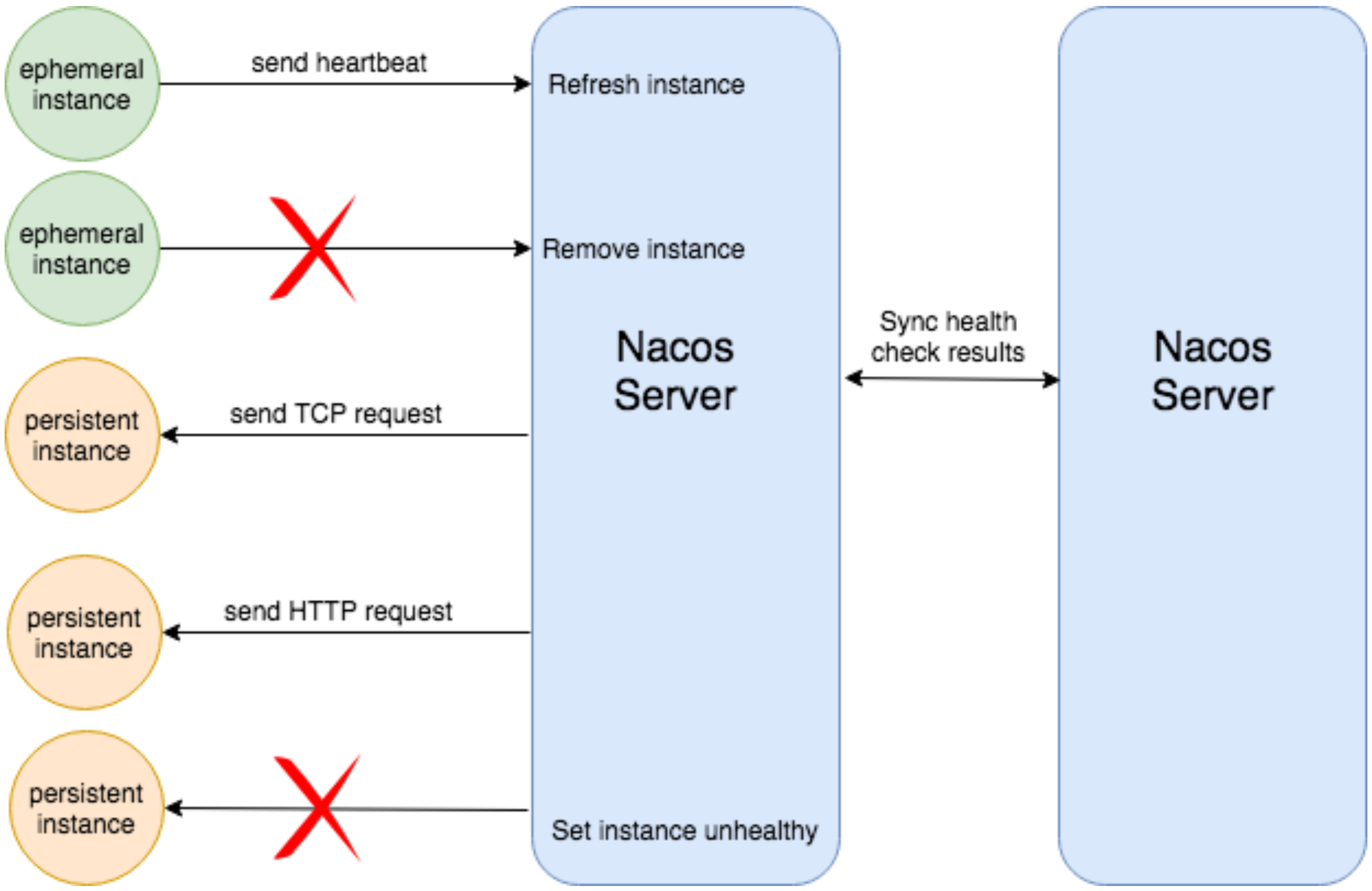
AP and CP work together



- Bussiness CRUD does not feel consistency type
- Abstracted datum CRUD
- Use a delegate to determine which way to go
- Two paths to choose: CP and AP
- Raft acting as CP, Distro acting as AP

Health check of Nacos Naming

- Ephemeral instance has to send hearbeat to keep itself alive
- Persistent instance can choose its health check request type: TCP, HTTP, MySQL or None
- One server for one instance
- Health check results are shared between Nacos servers



Load balancing strategies in Nacos

Basic elements for load balancing

- Health Status: unhealthy instances are in principle excluded for consumers
- Online Status: offline instances are excluded for consumers
- Weight: a way to differentiate the propotion of request flows (not used by Dubbo and Spring Cloud Alibaba now)

Advanced elements for load balancing

- Selector: select instances with customized strategies
  - Use label of instance to define the rule like: CONSUMER.label.site = PROVIDER.label.site
  - More selectors are coming.

NACOS.

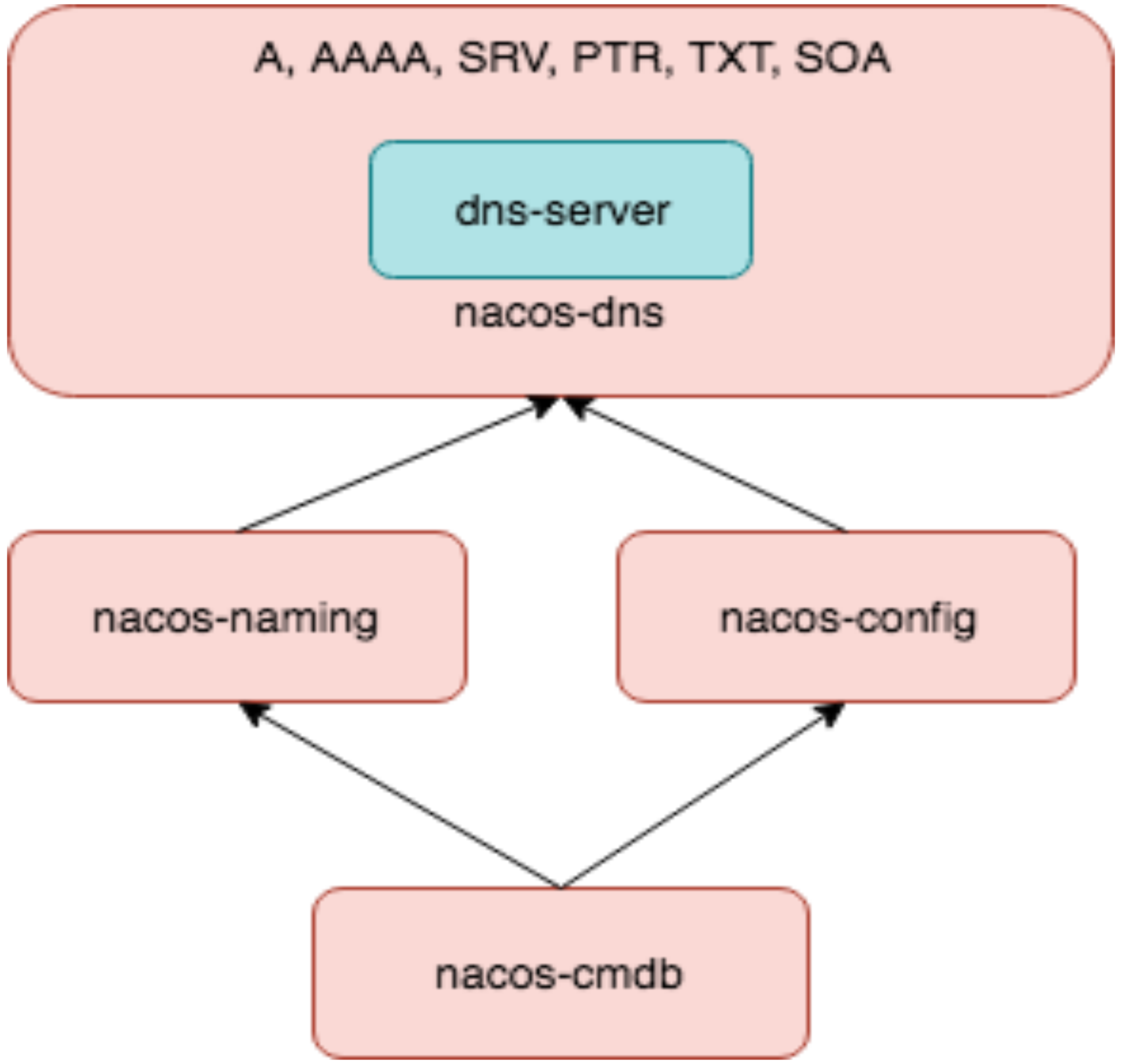
Part III

The Coming Features

Server side DNS resolution

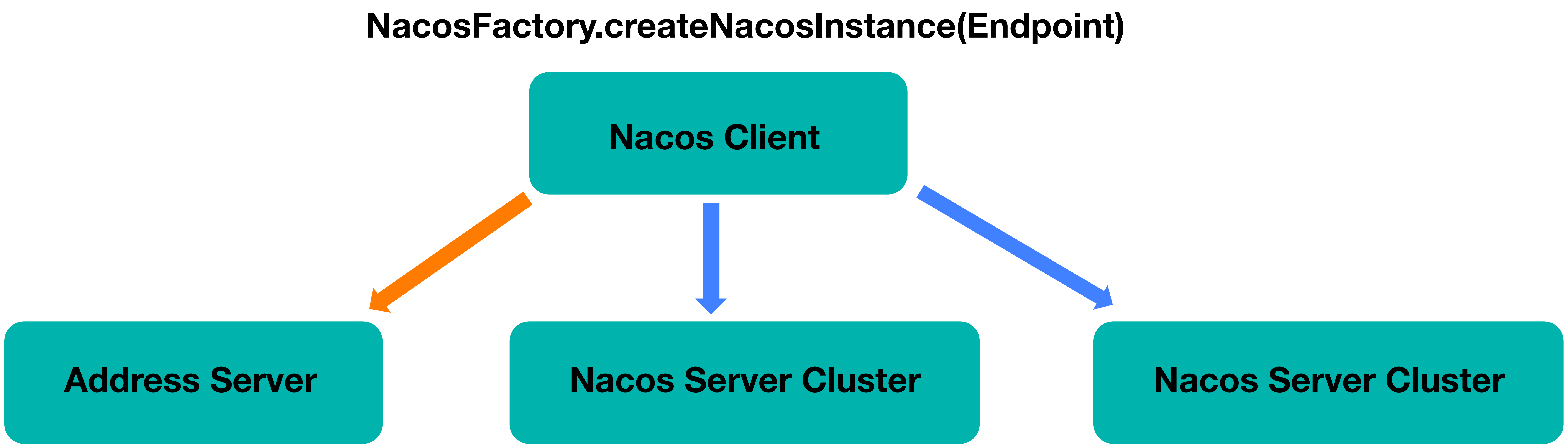
Why support DNS?

- Popular and easy to understand
- Libraries supported by every operation system and programming language
- DNS provides extensive types of data for service discovery



- Server side DNS for high availability and easier upgradation
- At least supports A, SRV, TXT, PTR
- Service data from nacos-naming
- System configuration from nacos-config
- Labels and metadata from nacos-cmdb

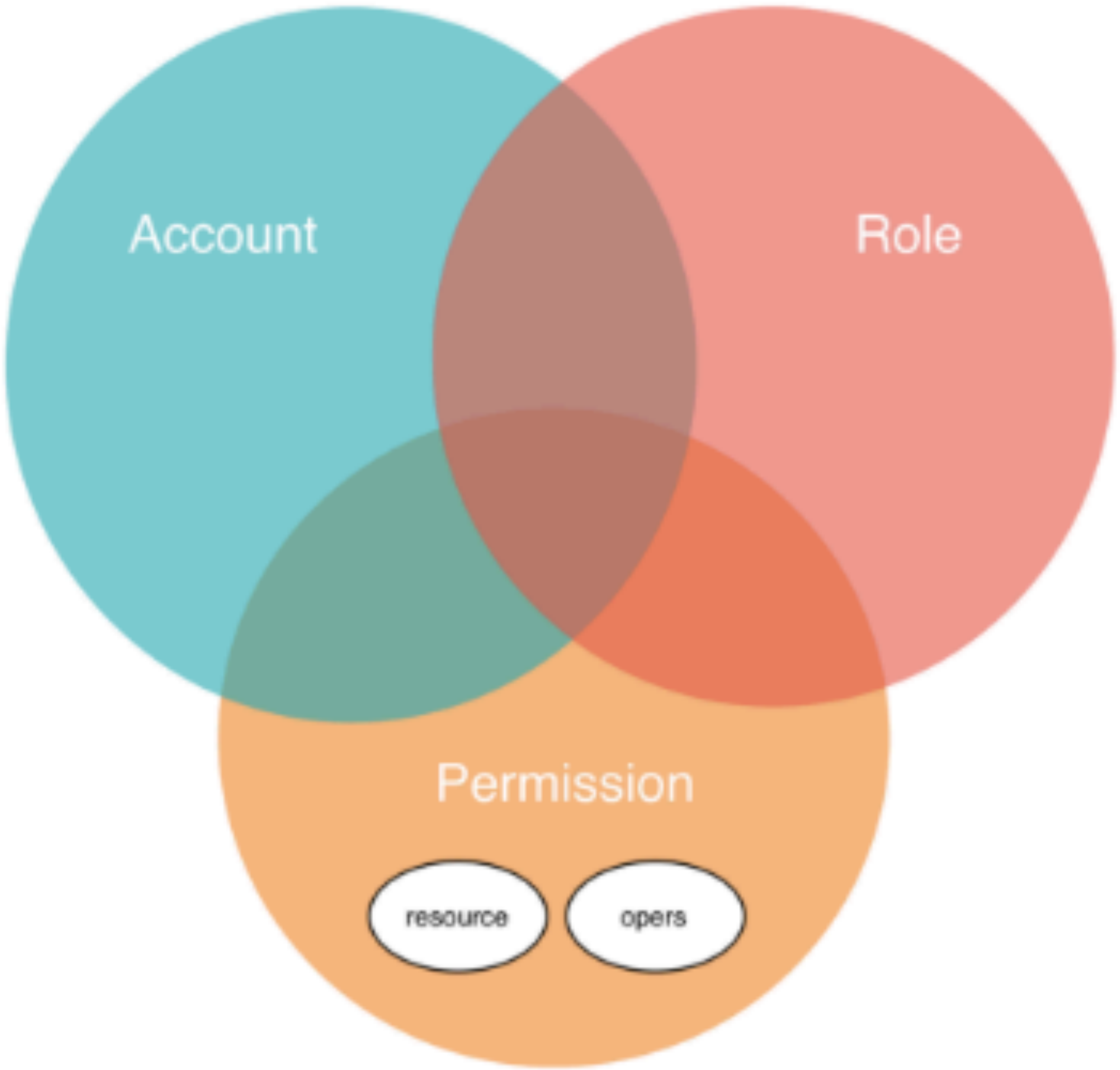
Address server mode



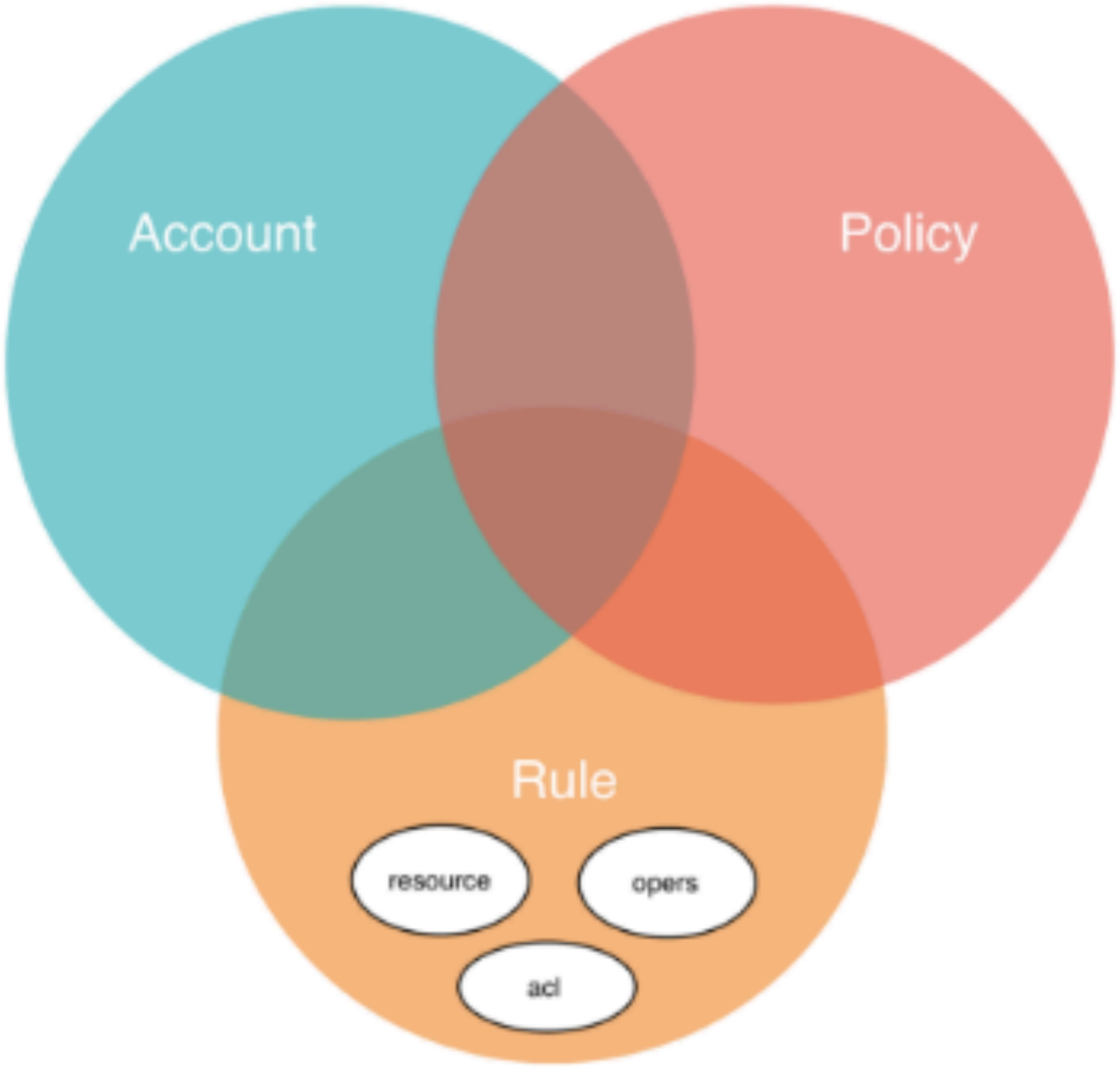
- Request address server to get Nacos server list
- Enable dynamic switch of Nacos server cluster by internally refreshing the server list from address server
- Different server list for different clients

Full access control of resource

RBAC



ABAC



- Enable secure access of resource
- Contain a built-in light-weighted implementation
- Open integration for different auth systems

Extensive plugin integration

- What we provide are never enough
- A better way to extend and develop new features

Expecting plugin support for:

- Health check
- Load balancing
- Notify system
- Storage system
- Message channel
- ……

- Plugin is optional
- Plugin can be chained

Test framework of Nacos

A more comprehensive test engine for Nacos

- Automatically deploys Nacos cluster
- Continously writes to and read from Nacos cluster, records and reports errors
- Introduces cluster failure to test the high availability of Nacos
- Open source for user running and contribution

Open for good ideas!

Thank You !