PyCon China 2020

PyConChina 2020
PYTHON 中国开发者大会 2020

Python For Good

Python Microservice Application Performance Monitoring

柯振旭

Engineer,
Tetrate.io

About Me

柯振旭

@kezhenxu94

tetrate

- Tetrate.io Engineer. Prev. Alibaba
- Open-source Enthusiast
  - Apache SkyWalking PMC Member
  - Apache Dubbo Committer
  - Apache Local Community Beijing (ALC) Member

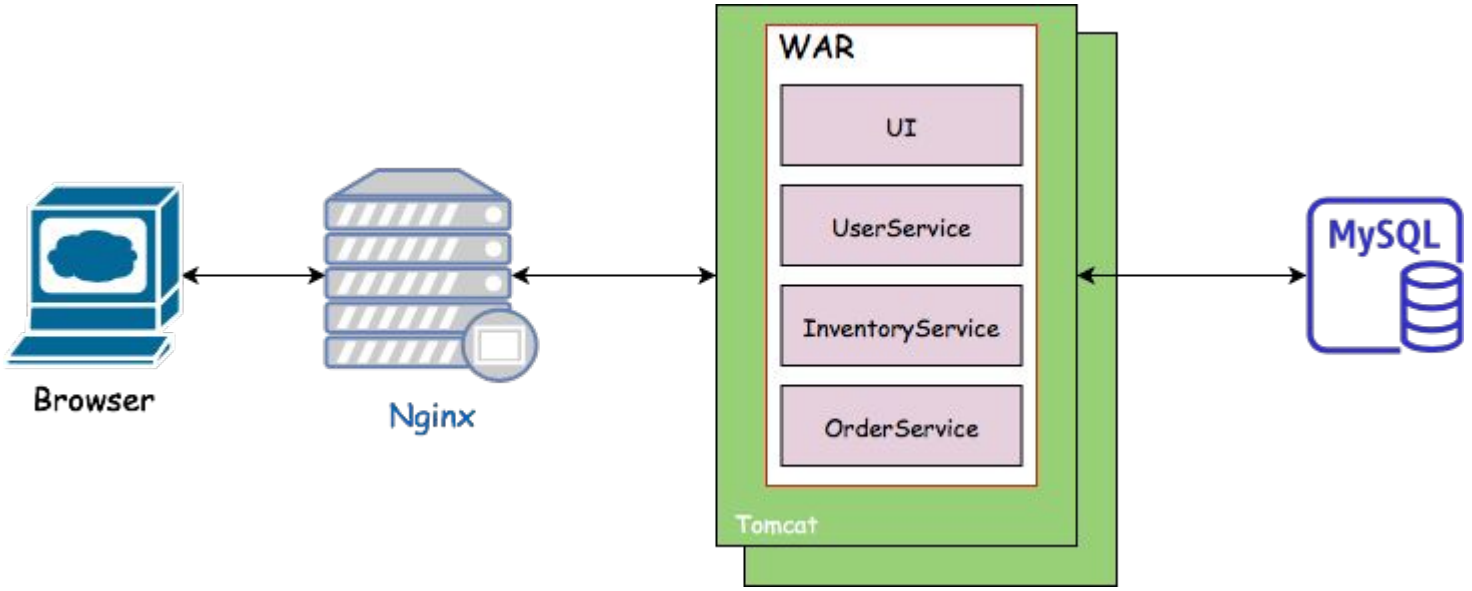
Agenda

- Why APM System
- Apache SkyWalking
- Apache SkyWalking-Python Uncovered

Python For Good
PyCon China 2020 | PYTHON 中国开发者大会 2020

Traditional Monolithic Architecture

- Easy to
  - Develop (Test)
  - Deploy
  - Scale



Why APM System

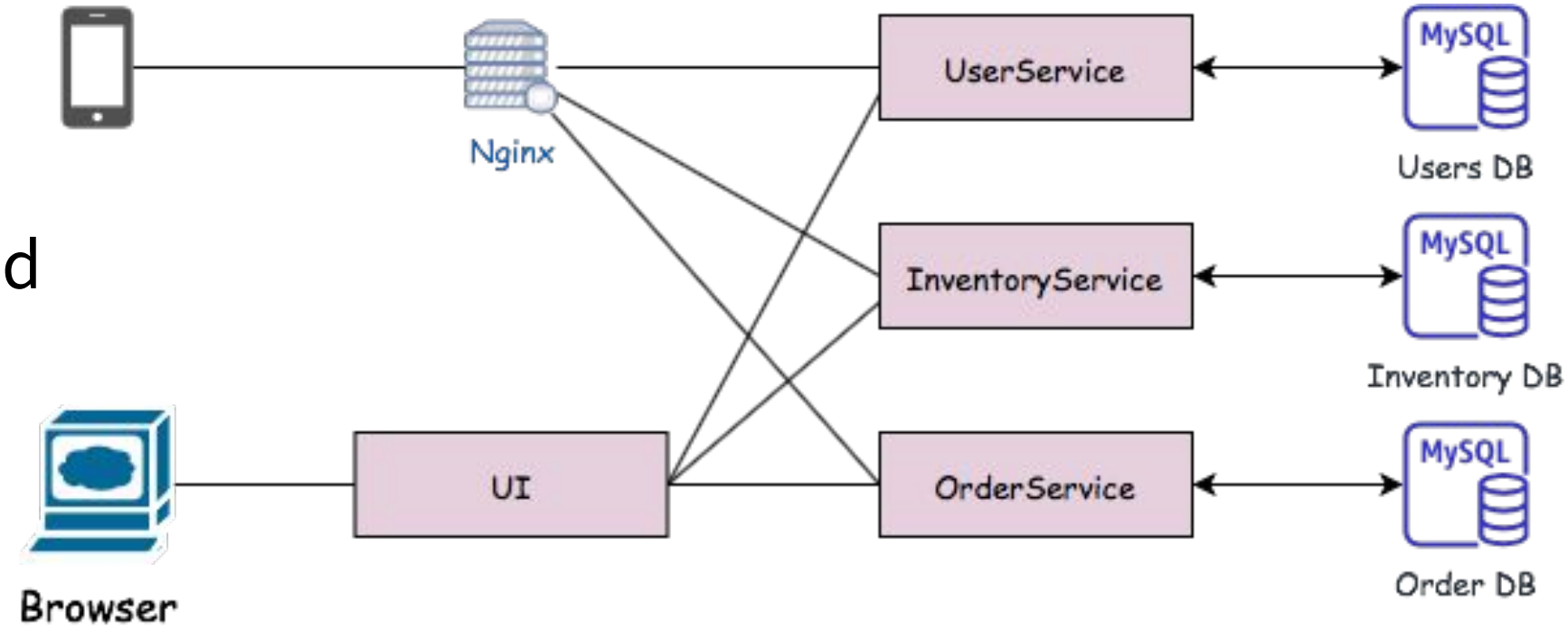
Microservice Architecture

Difficult to

Debug

Comprehend

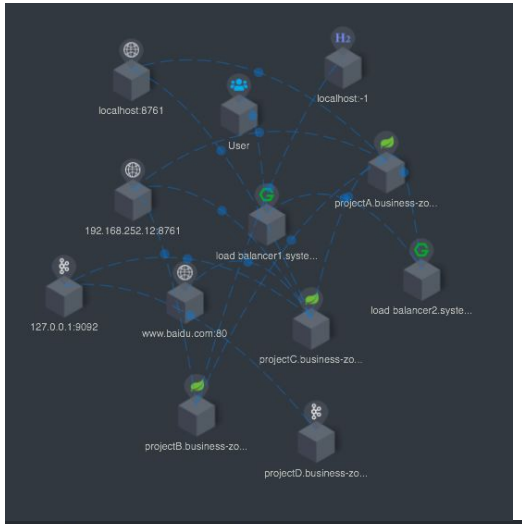
Deploy

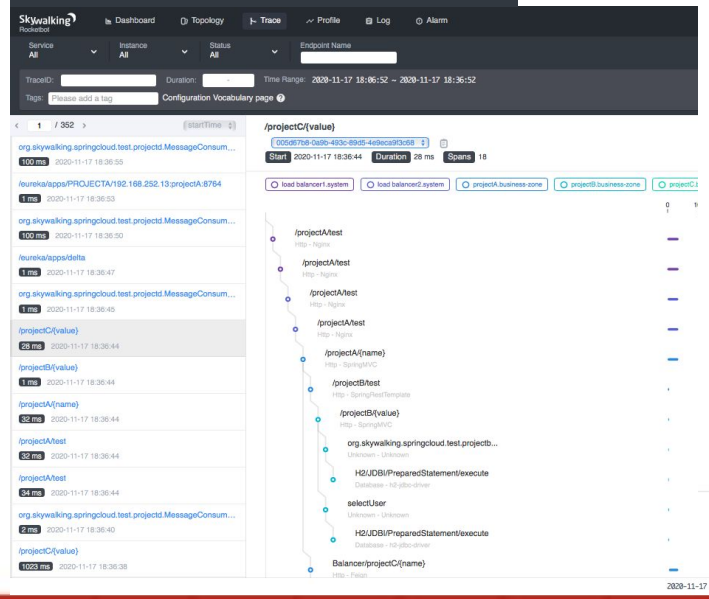


What Can APM System Do?

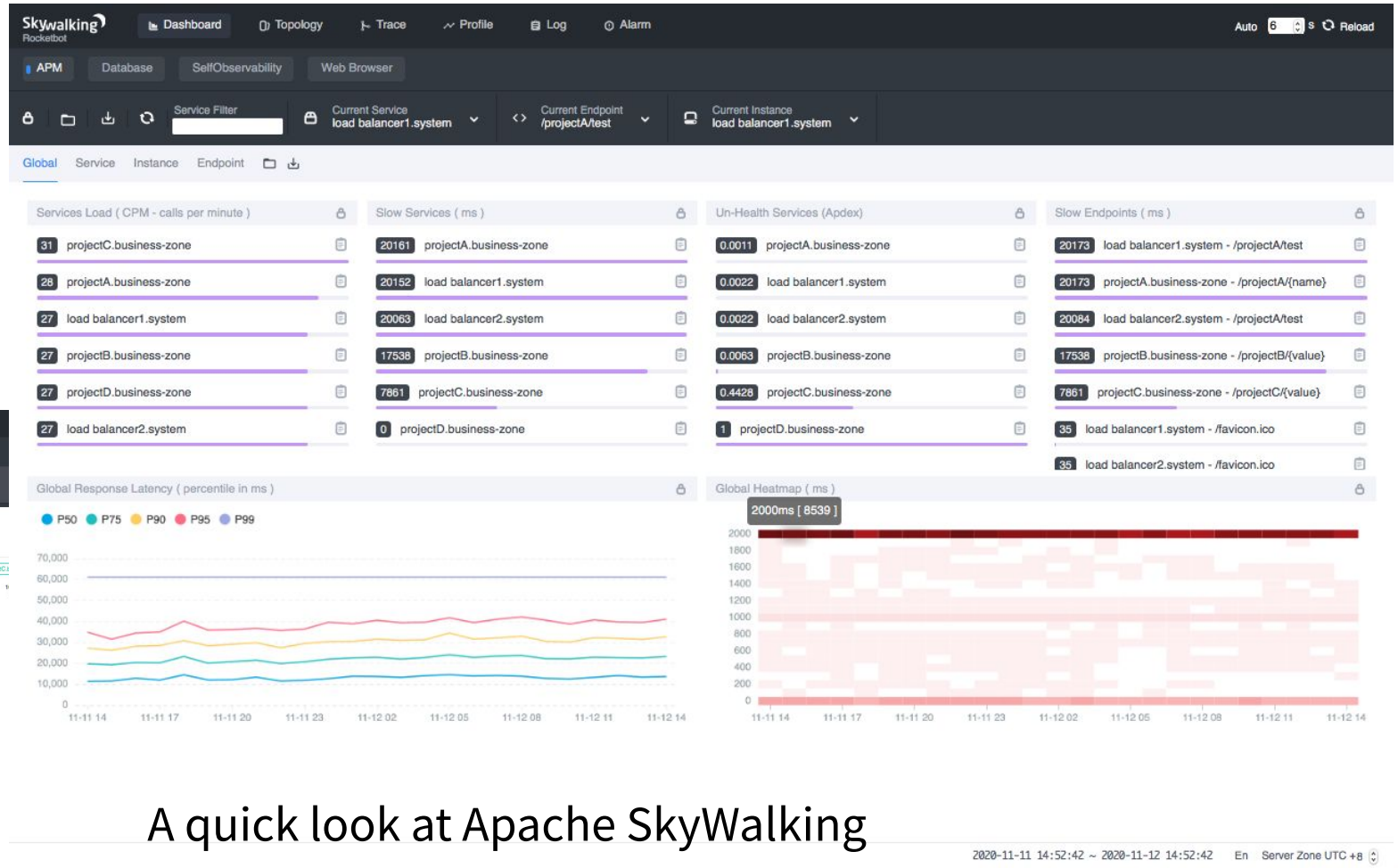
- **Distributed tracing and context propagation**
- **Service topology map analysis**
- **Service, service instance, endpoint metrics analysis**
- Root cause analysis. Profile the code at runtime
- Service, service instance and endpoint dependency analysis
- Slow services and endpoints detected
- Performance optimization

Apache SkyWalking



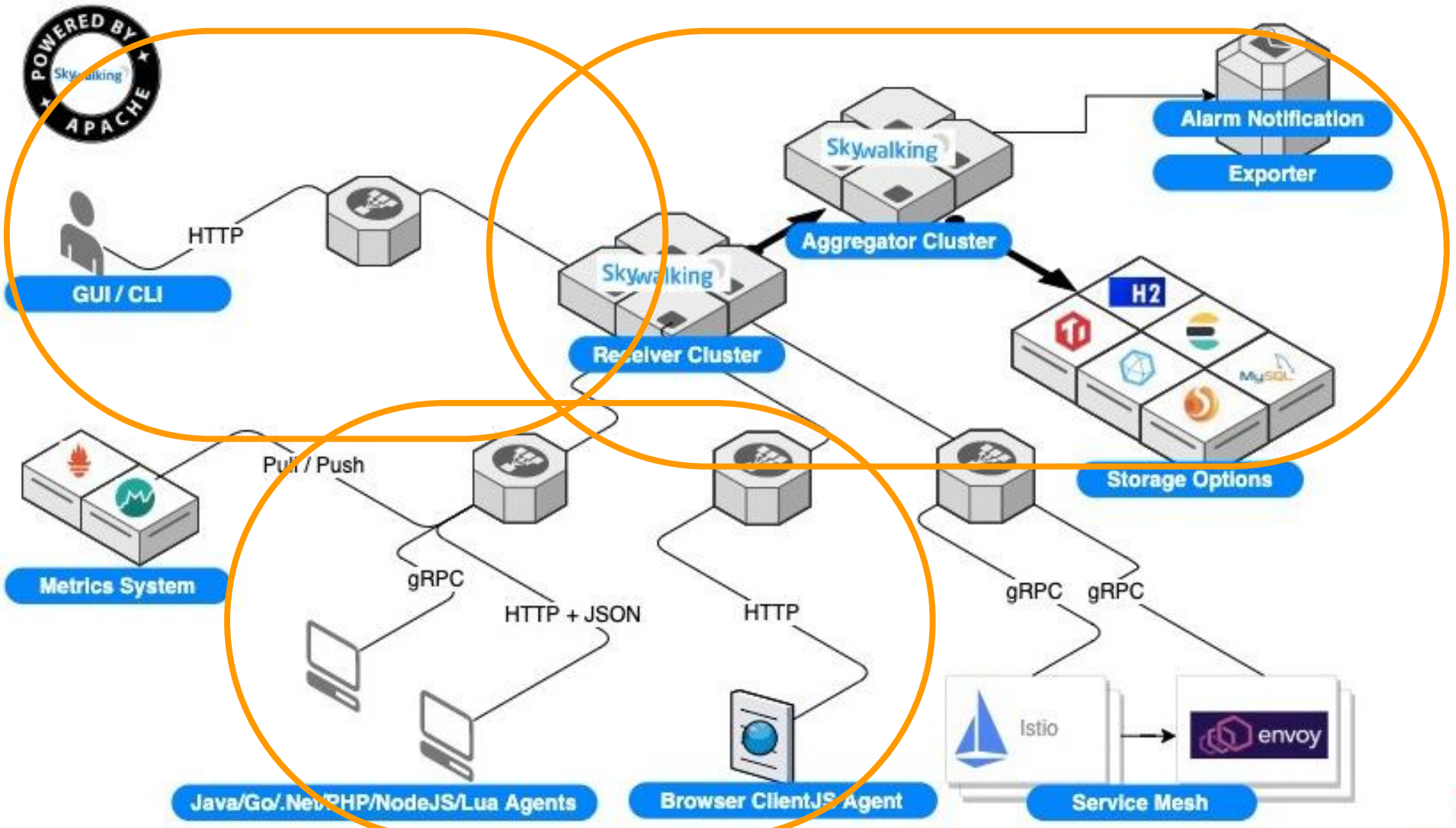


http://122.112.182.72:8080



A quick look at Apache SkyWalking

Apache SkyWalking



Apache SkyWalking Python

The Python Agent for Apache SkyWalking, which provides the native tracing abilities for Python applications.

Apache SkyWalking Python

Quick Start

```
$ pip install apache-skywalking
```

```
from skywalking import agent, config

if __name__ == '__main__':
    config.init(collector='127.0.0.1:11800', service='your service')
    agent.start()
    # your codes ...
```

Apache SkyWalking Python

Quick Start

```
$ pip install apache-skywalking
```

```
from skywalking.decorators import trace
```

```
@trace()  # the operation name is the method name('my_method') by default
def my_method():
    print('Hello World')
```

Apache SkyWalking Python

Quick Start

```
$ pip install apache-skywalking
```

```
context: SpanContext = get_context()  # get a tracing context

with context.new_entry_span(op='https://github.com/apache') as span:
    span.component = Component.Flask
    span.tag(Tag(key='Org', val='Apache'))

# context.new_exit_span()/context.new_local_span()
```

Agent Core Uncovered

```
# Client Side, Exit Span
context = get_context()
carrier = Carrier()
with context.new_exit_span(op=url_param.path or "/", peer=url_param.netloc, carrier=carrier) as span:
    for item in carrier:
        headers[item.key] = item.val

# Server Side, Entry Span
context = get_context()
carrier = Carrier()
for item in carrier:
    sw_http_header_key = ...
    if sw_http_header_key in request.META:
        item.val = request.META[sw_http_header_key]
```

Agent Plugin Uncovered

Everything in Python is an object

Nearly all objects are mutable in Python

Agent Plugin Uncovered

```python
from django.core.handlers.base import BaseHandler


_get_response = BaseHandler.get_response

def _sw_get_response(this, request):
    context = get_context()
    carrier = Carrier()
    # ...
    with context.new_entry_span(op=request.path, carrier=carrier) as span:
        resp = _get_response(this, request)
        # ...
        return resp

BaseHandler.get_response = _sw_get_response
```

Apache SkyWalking Python

Plugins

- http.server
- urllib.request
- requests
- Flask
- PyMySQL
- Django
- redis-py
- kafka-python
- tornado
- pika
- pymongo
- elasticsearch
- urllib3

PyCon China 2020

Python For Good

PyConChina 2020 | PYTHON 中国开发者大会 2020

THANK YOU

@kezhenxu94

kezhenxu94

kezhenxu94

@ASFSkyWalking