



100% Performance Improving, 90% Cost Saving

— How Tencent Scales to Millions with Serverless Next.js

Tinafang
Senior Product Manager
Tencent Cloud

Who Am I?



Serverless Advocate

Product Manager @  Tencent Cloud

github.com/tinafangkunding

[@tinafangkd](https://twitter.com/tinafangkd)

tinafang.com

Agenda

01

**Why Using
Next.js**

02

**Architecture of
Tencent SSR**

03

**Combine Next.js
with Serverless**

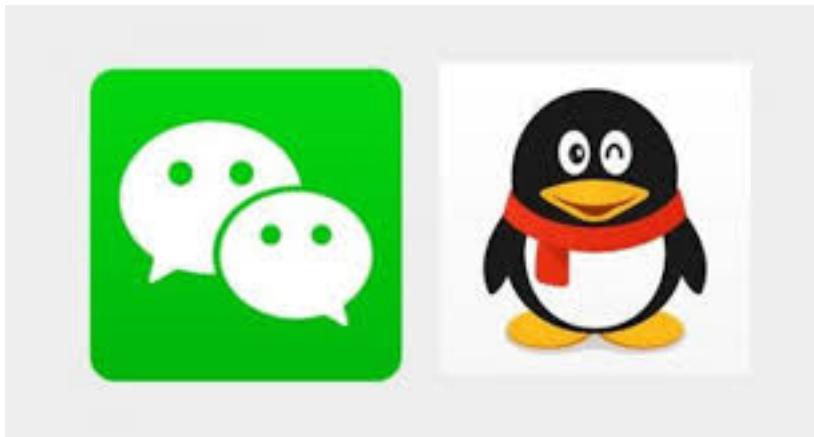
04

Summary

About Tencent

Tencent 腾讯

- One of the world's top technology companies
- Services include social network, music, e-commerce, mobile games, etc.
- Known as instant messengers like Tencent QQ and WeChat



1. Why Using Next.js?



Story Begins From...



ABCmouse

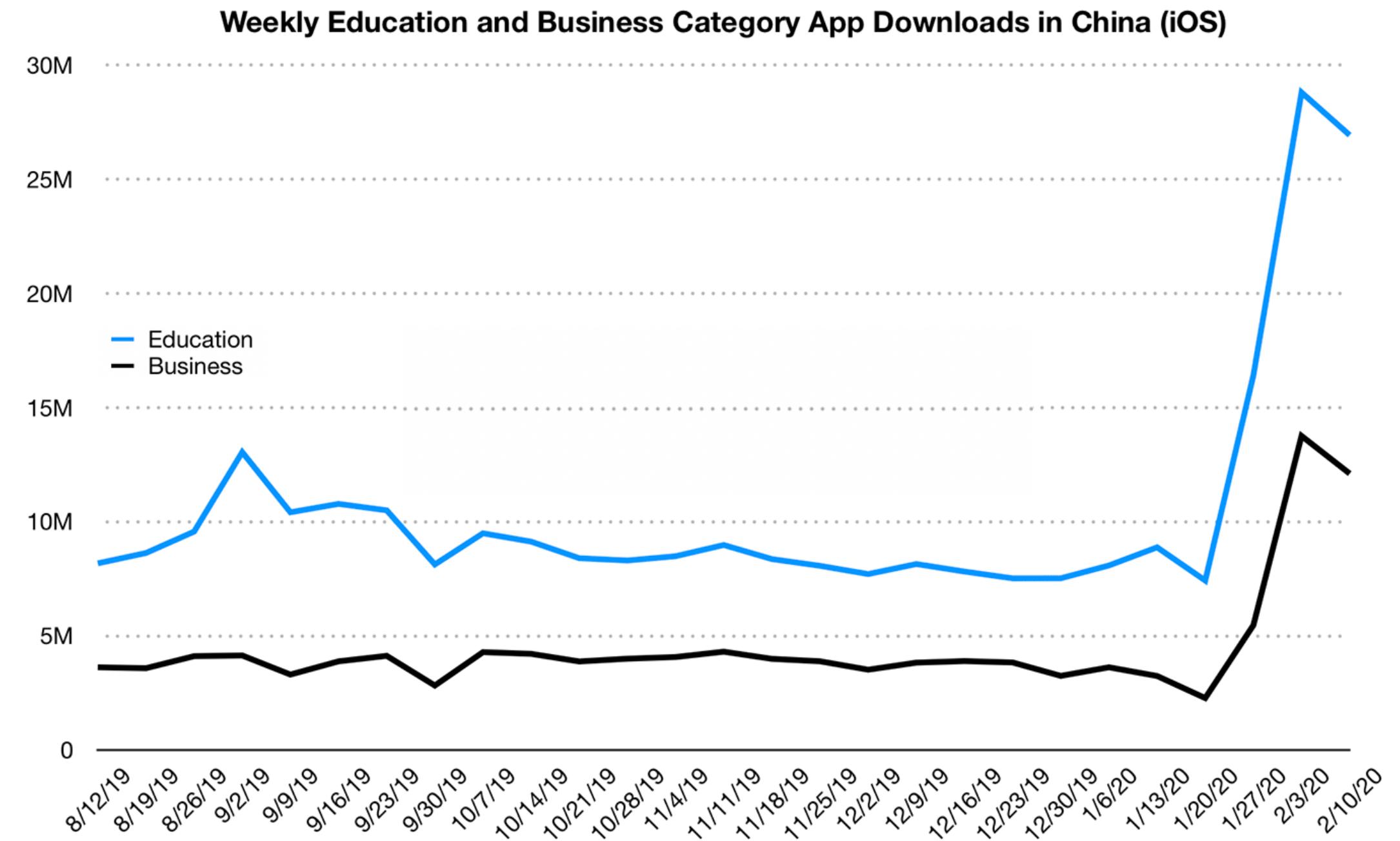


企鹅辅导



腾讯课堂

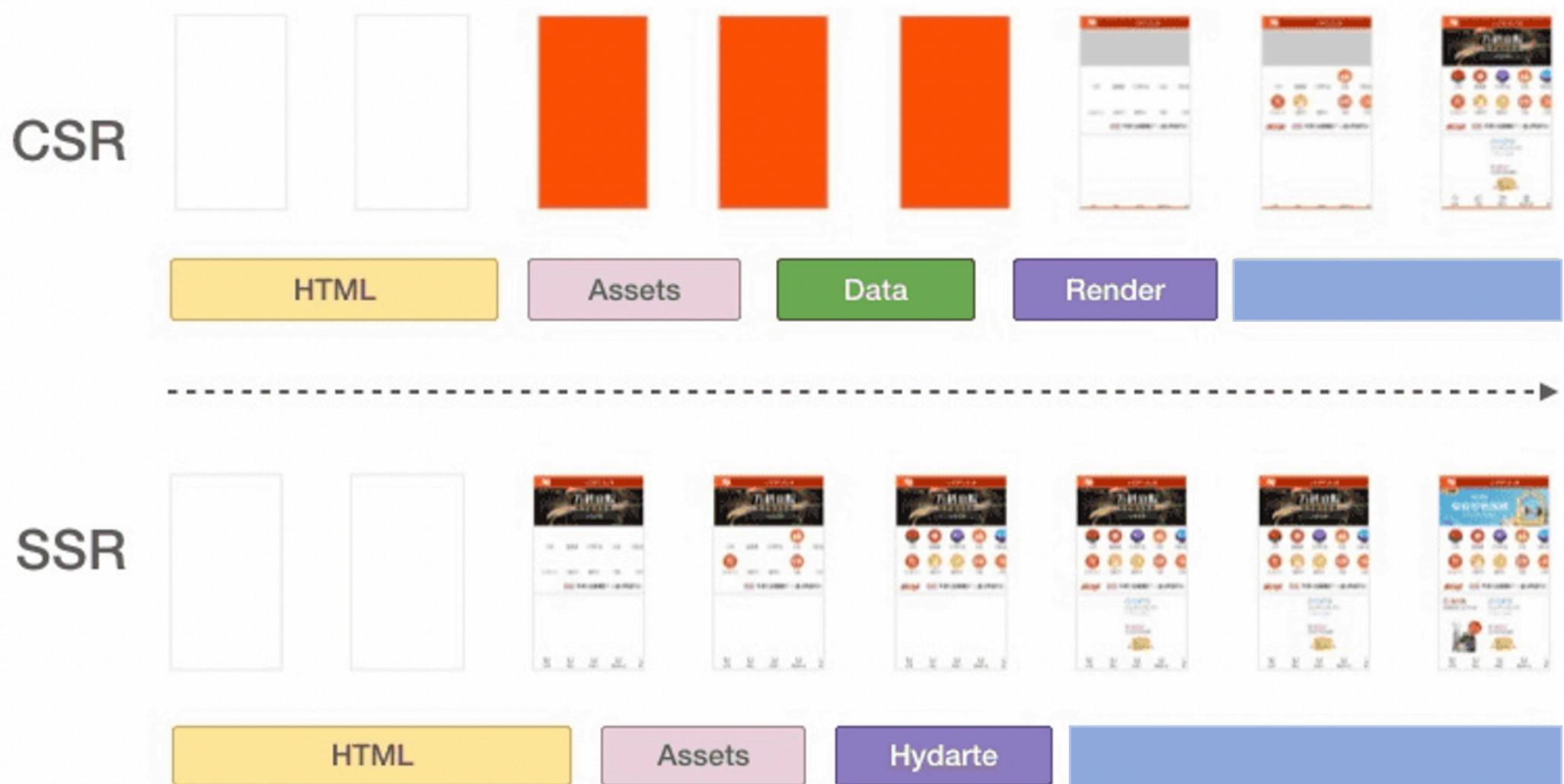
| | |
|--------------|------------------|
| Active Users | 40,000 → 300,000 |
| Courses | 100,000 |
| Teachers | 50,000 |



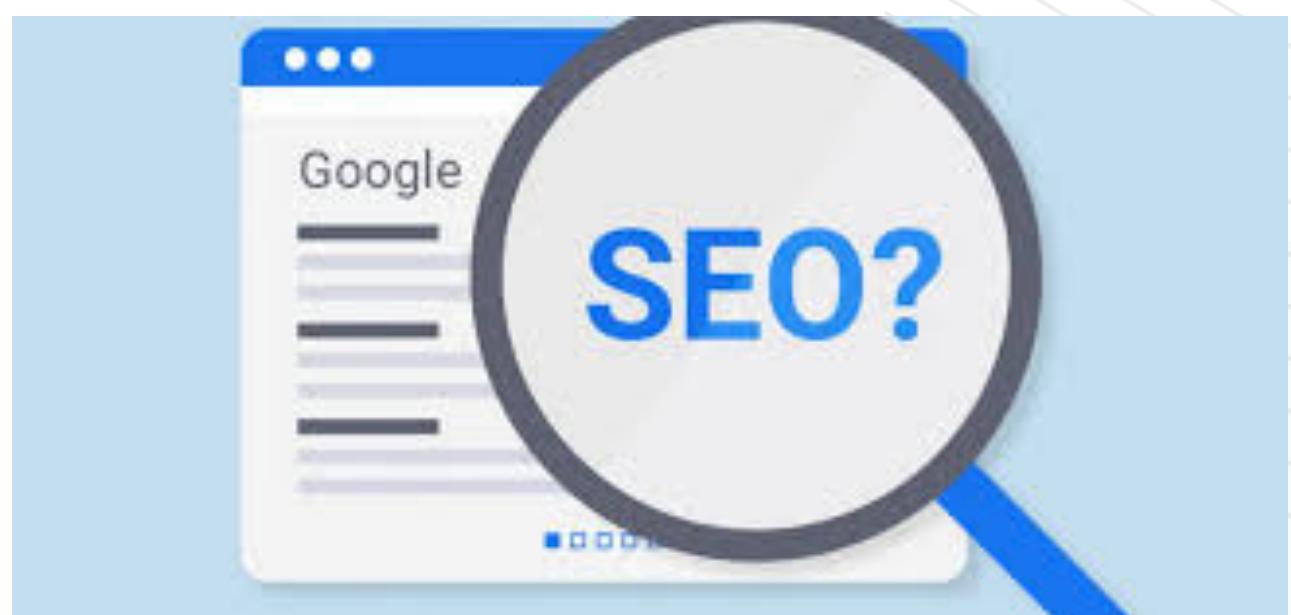
<https://techcrunch.com/2020/02/23/china-roundup-tech-companies-coronavirus/>

Why Using SSR?

1. Performance (FCP or FMP)



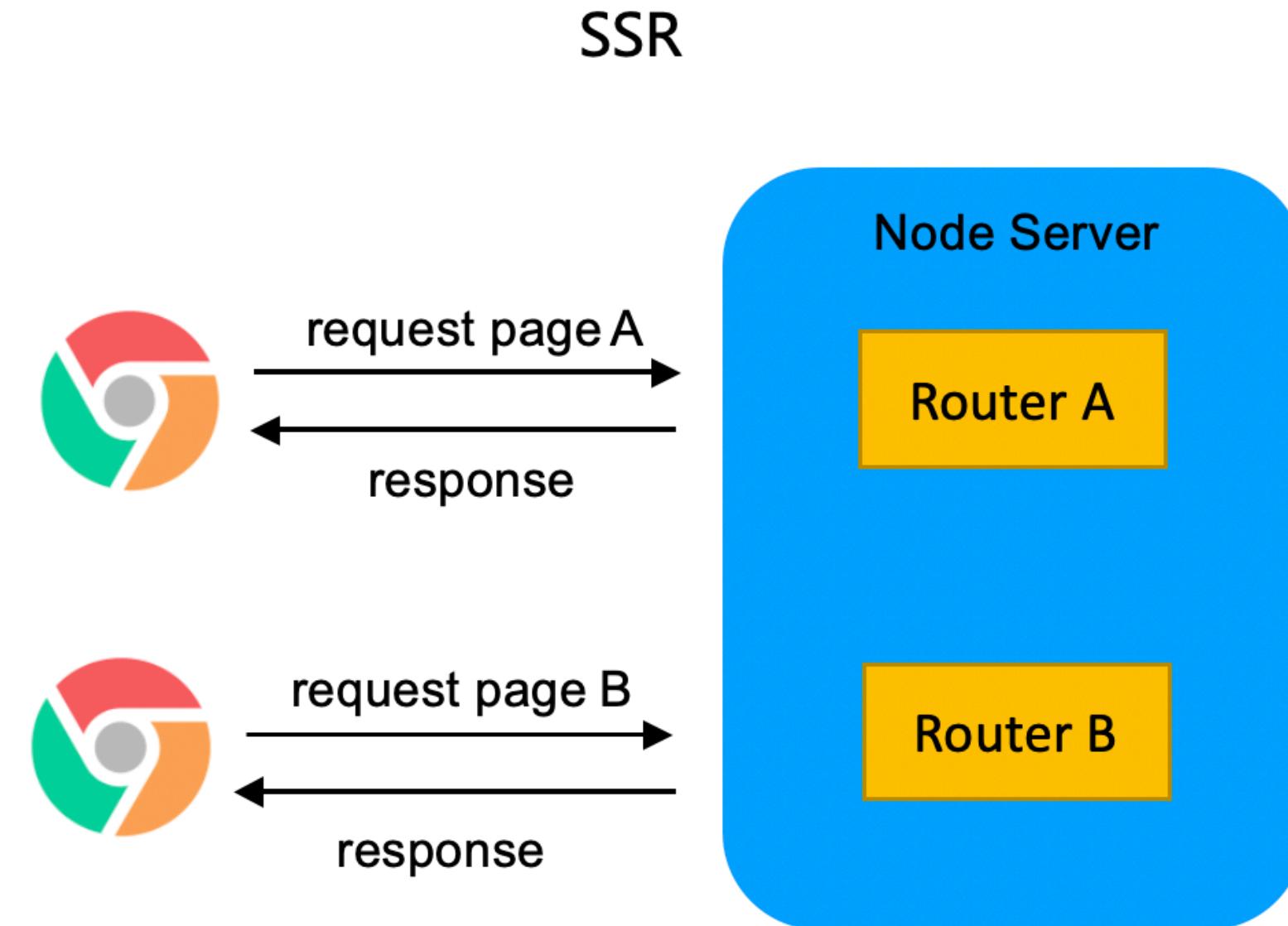
2. SEO



<https://www.oschina.net/news/113671/rax-ssr-news>

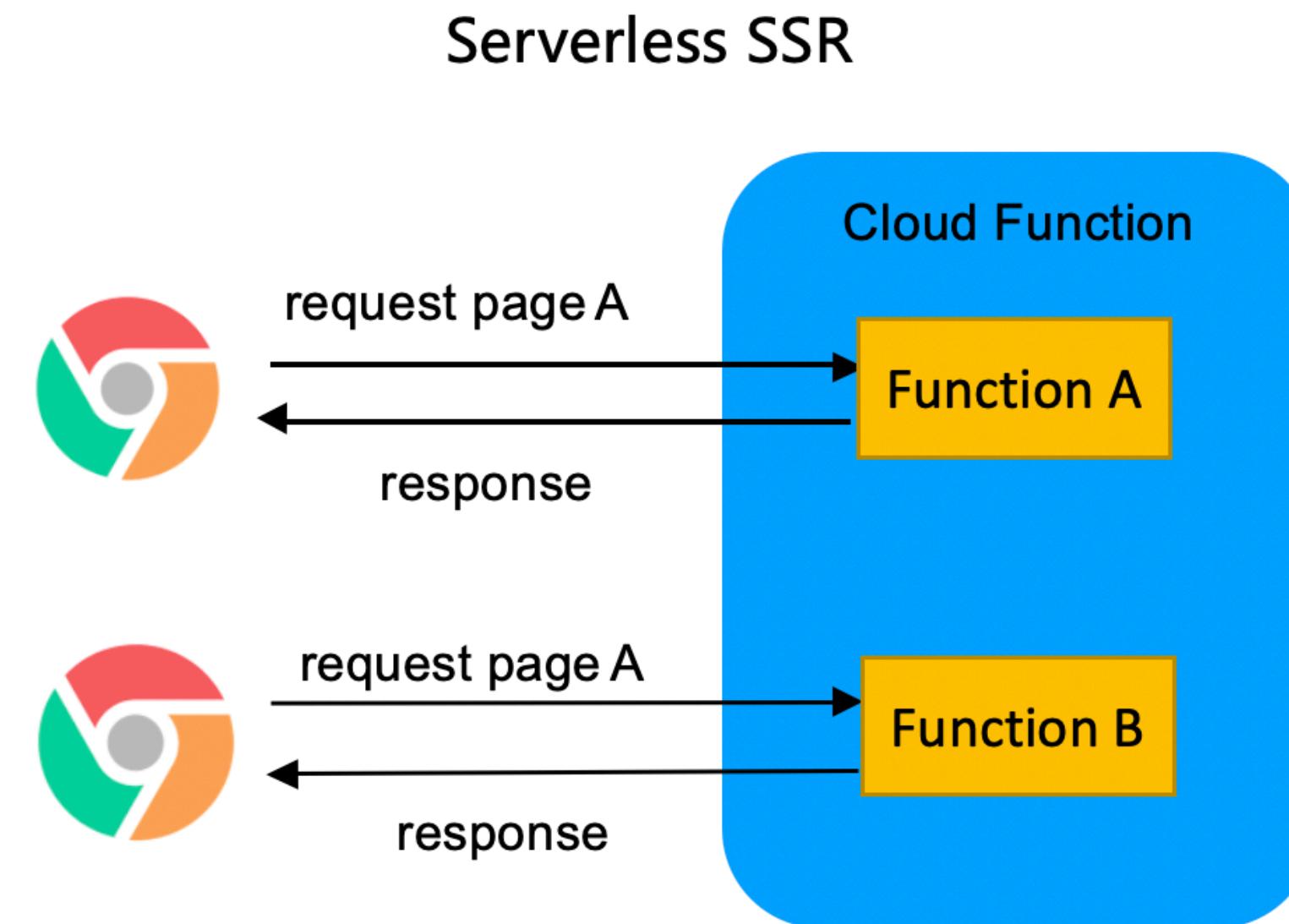
Challenges in SSR

- Maintain Node Server
- Auto Scaling
- Cost of Servers
- Nginx Configurations
- ...



Serverless + SSR

- No need to maintain
- Auto Scaling
- Pay as you Go
- ...

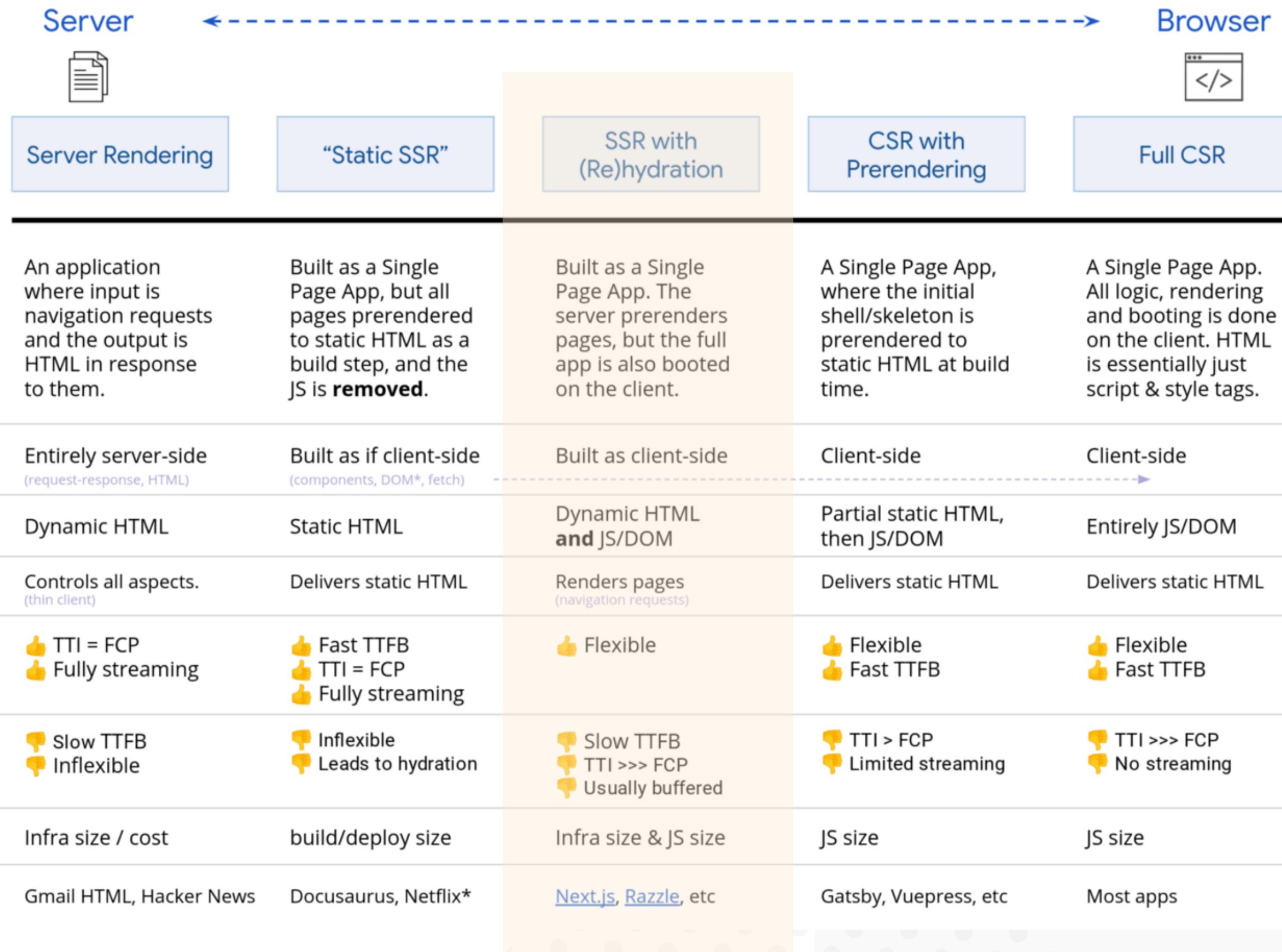


Next.js

 Unstar

54,396

- Static Site Generation (SSG)
- Server Side Rendering (SSR)
- CDN+Static Storage
- Dev Toolkit/Platform
- Open Source
- ...

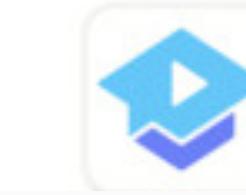
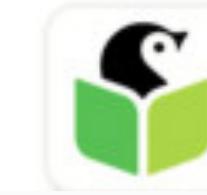


<https://github.com/vercel/next.js>

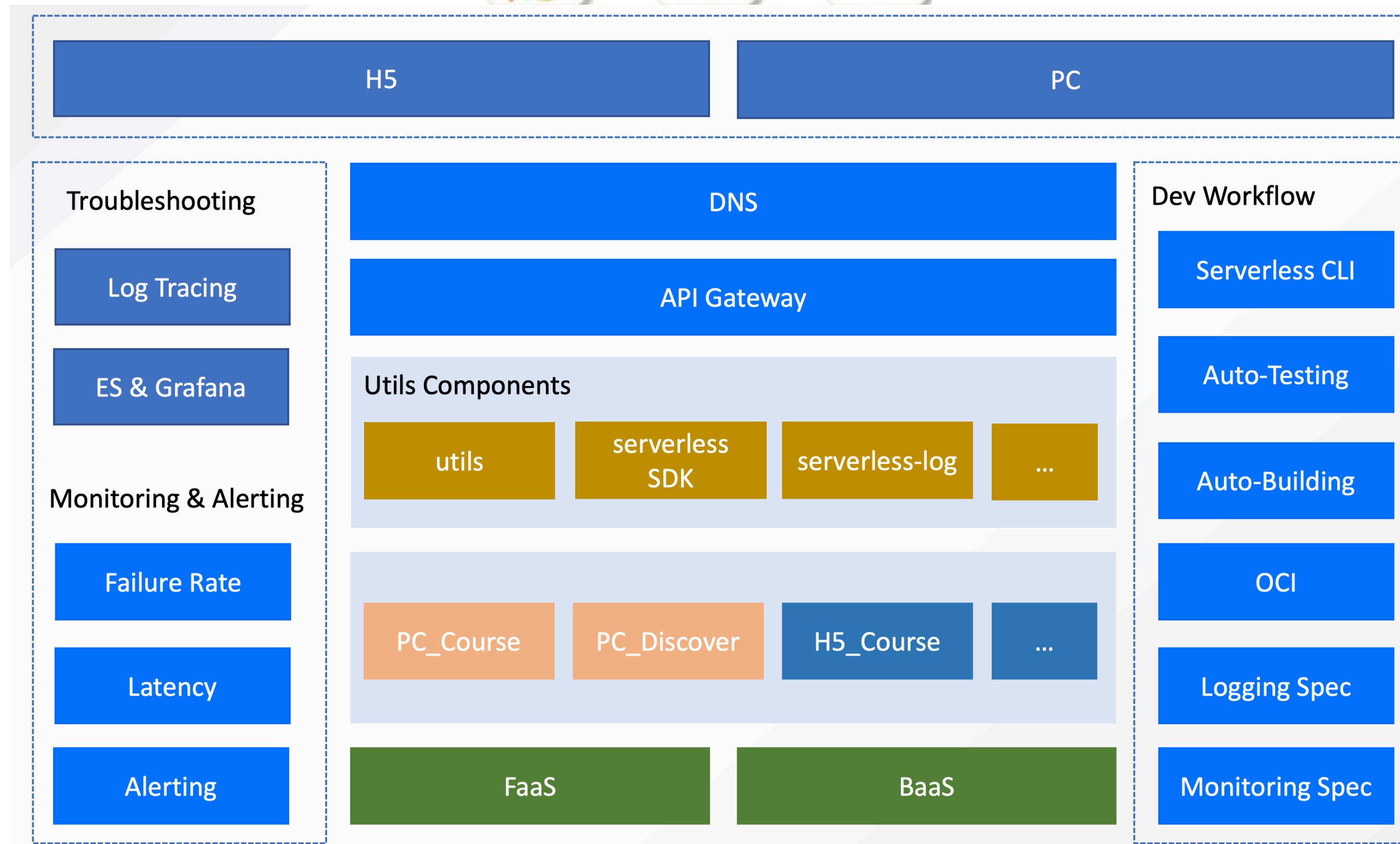
2. Architecture of Tencent SSR



Architecture in Tencent Next.js App



<https://fudao.qq.com/>



Next.js Functionalities

LRU Cache

```
async function cacheRender(req, res) {
  const key = getCacheKey(req);
  // reder /list to /
  // const reqPath = req.path === '/list' ? '/' : req.path;
  const reqPath = req.path;
  if (Cache.has(key)) {
    res.setHeader('X-Cache', 'HIT');
    res.send(Cache.get(key));
    return;
  }

  try {
    const html = await app.renderToHTML(req, res, reqPath, req.query);
    if (res.statusCode !== 200) {
      res.send(html);
      return;
    } else {
      Cache.set(key, html);
      res.setHeader('X-Cache', 'MISS');
      res.send(html);
    }
  } catch (err) {
    res.statusCode = 500;
    app.renderError(err, req, res, reqPath, req.query);
  }
}
```

<https://github.com/vercel/next.js/issues/14737>

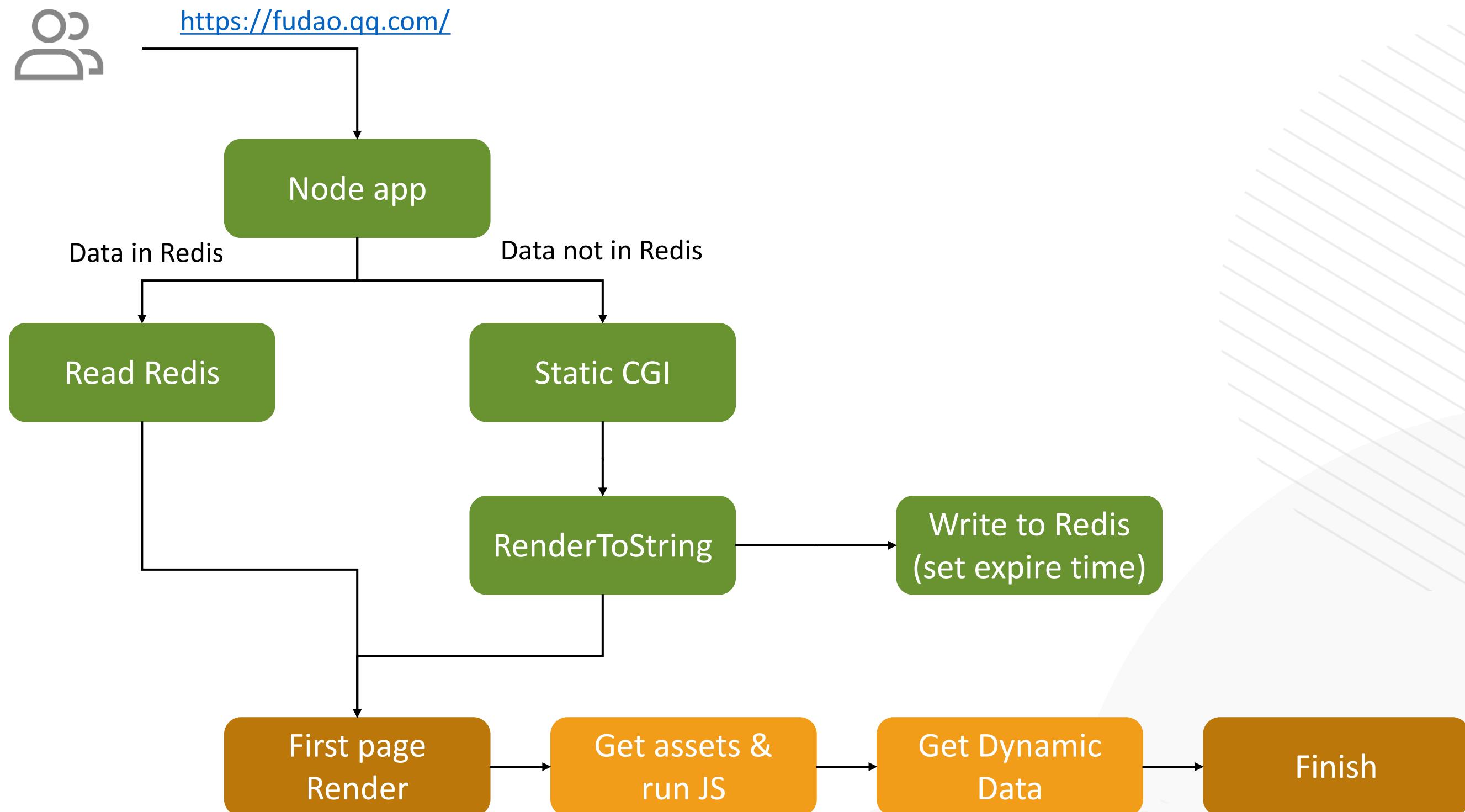
Separate Dynamic/Static Data

Static Data

- Course Name
- Course Description

Dynamic Data

- Login Related data
- Discount of courses
- Status of payment

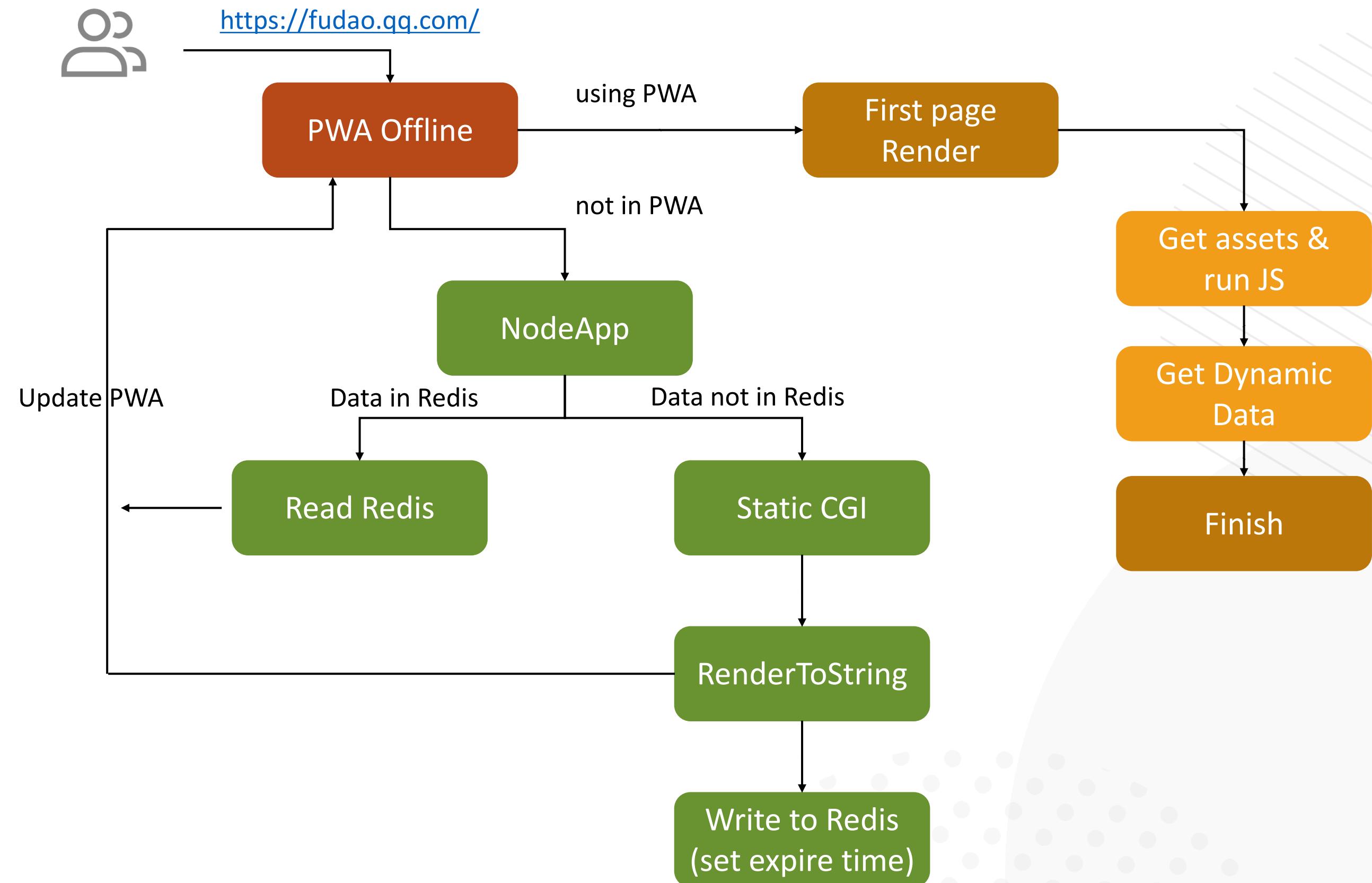


Use PWA to Work Offline

Further Step:

Client Side

PWA to Cache HTML Page



Next.js Functionalities

Custom Server: Express/Koa

```
async function startServer() {
  await app.prepare();

  const server = Express();
  server.use(Express.static(join(__dirname, '../public/static')));

  server.get('/', async (req, res) => {
    return ssrCache({ req, res });
  });

  server.get('/about', async (req, res) => {
    return ssrCache({ req, res });
  });

  server.get('/topic', async (req, res) => {
    return ssrCache({ req, res });
  });
}
```

```
const express = require('express')
const next = require('next')

async function createServer() {
  const app = next({ dev: false })
  const handle = app.getRequestHandler()

  await app.prepare()
  const server = express()

  server.all('*', (req, res) => {
    return handle(req, res)
  })

  // define binary type for response
  // if includes, will return base64 encoded, very useful for images
  server.binaryTypes = ['/*']

  return server
}

// comment out `listen`
// exports app server

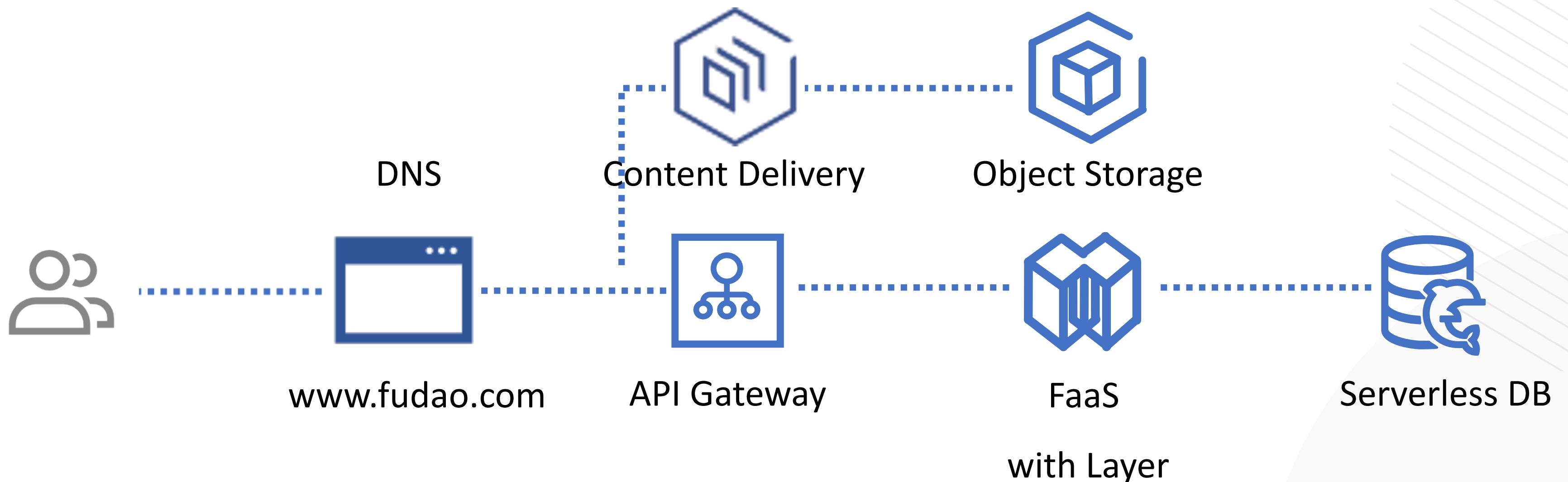
// createServer().listen(3000);

module.exports = createServer
```

3. Next.js + Serverless



Services we use in Next.js Applications



Performance Improving 100%

Using Object Storage to Separate Static Assets, which Includes:

- `next/static` folder
- Files inside `public` folder

Update `next.config.js`

```
const isProd = process.env.NODE_ENV === "production";
const STATIC_URL =
  "https://serverless-nextjs-xxx.cos.ap-guangzhou.myqcloud.com";
module.exports = {
  env: {
    // for local development
    STATIC_URL: isProd ? STATIC_URL : "http://localhost:3000",
  },
  assetPrefix: isProd ? STATIC_URL : "",
};
```

Update Path in `public` folder

```
<!-- before -->
<head>
  <title>Create Next App</title>
  <link rel="icon" href="/favicon.ico" />
</head>

<!-- after -->
<head>
  <title>Create Next App</title>
  <link rel="icon" href={`${process.env.STATIC_URL}/favicon.ico`} />
</head>
```

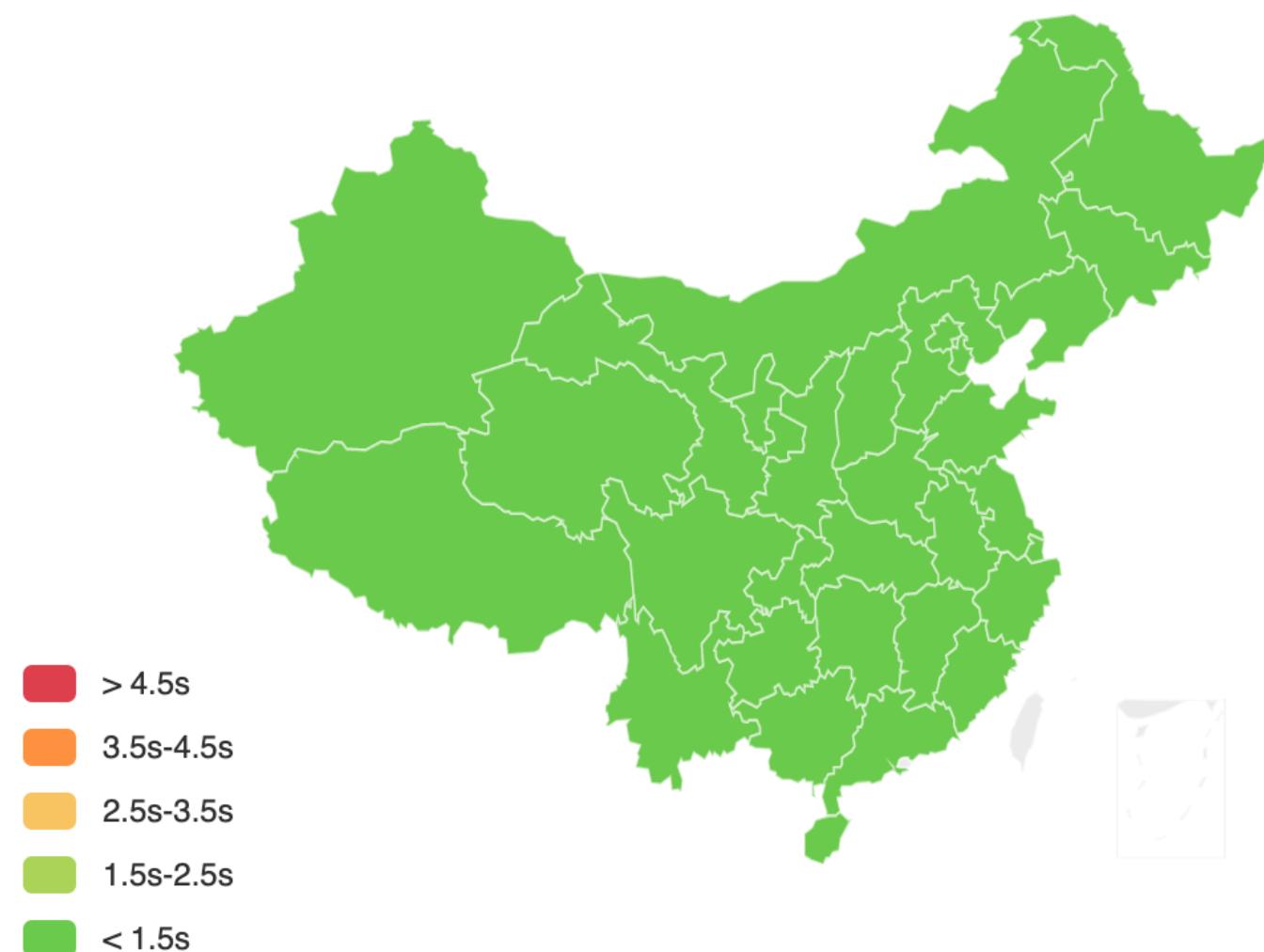
Performance Improving 100%

CDN Acceleration
with World-Wide Nodes
1000+ Nodes
50+ Regions/Countries



Performance Improving 100%

| 检测域名 <https://fudao.qq.com/> 2020-10-19 18:39:13



运营商: 全部 电信 移动 联通

地区: 全部 华北 东北 华东 华中 华南 西南 西北

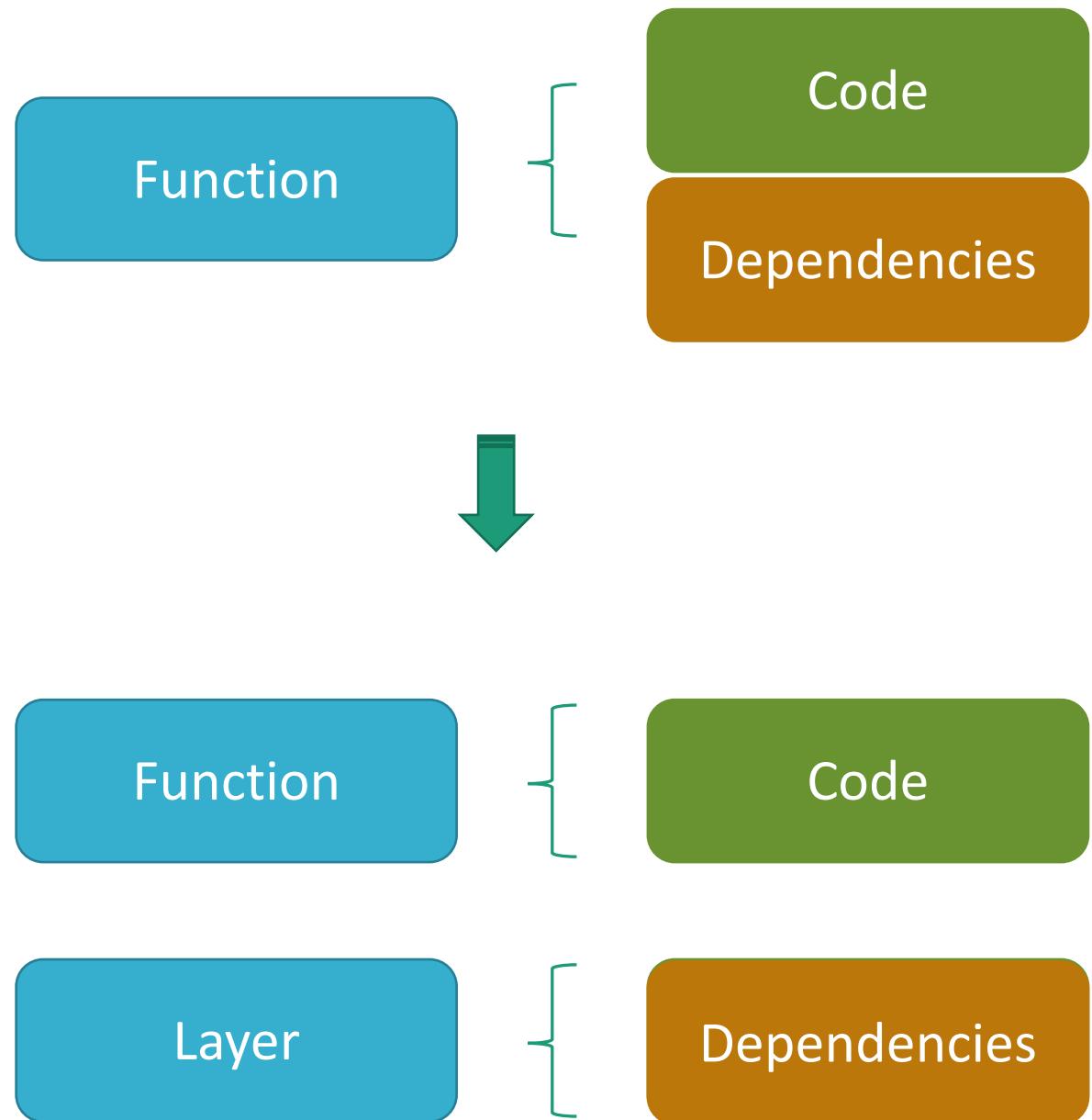
| | | | |
|-------|----------|------------|--------|
| 93 | 0.31s | 2.111s | 0.612s |
| 检测节点数 | 上海电信(最快) | 广东韶关联通(最慢) | 平均响应 |
| nodes | min | max | avg |

| 线路 | 最快节点 | 最慢节点 | 平均响应 |
|----|----------------|---------------------|--------|
| 电信 | 上海电信 0.31s | 新疆克孜勒苏柯尔克... 0.856s | 0.574s |
| 移动 | 上海移动 0.328s | 河北保定移动 1.341s | 0.623s |
| 联通 | 河北张家口联通 0.381s | 广东韶关联通 2.111s | 0.639s |

Performance Improving 100%

Layer in Functions

to accelerate deployment

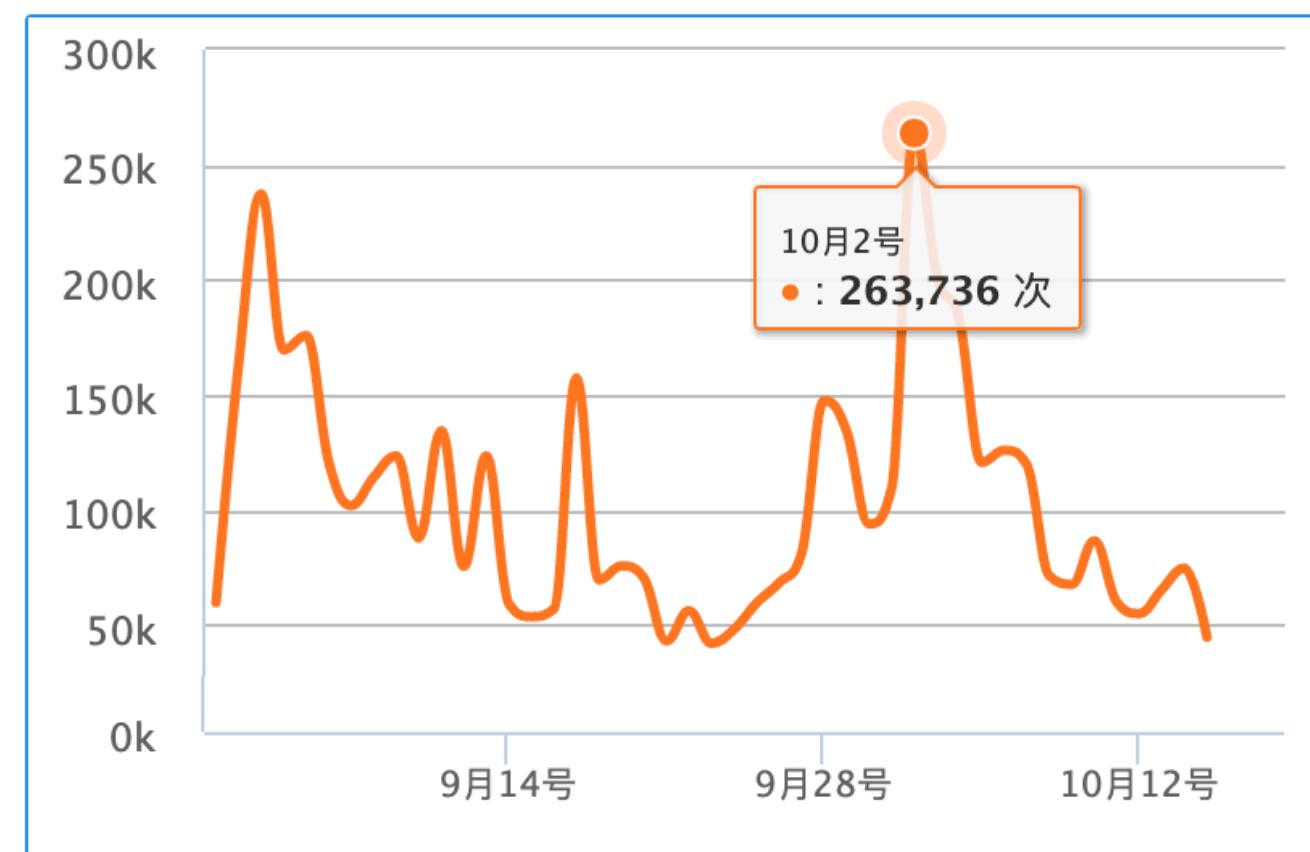


Cost Saving 90%

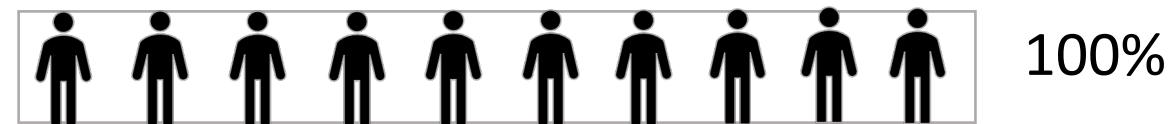
Using “Serverless” Services instead of prevision resources

- SCF – Cloud Function
- API Gateway
- COS & CDN
- ...

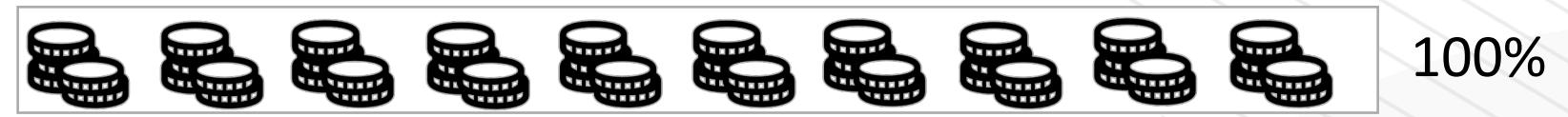
调用次数 (单位: 次) **Invocations**



Labor Costs



Resource Cost



Time to Market



CI/CD & Monitoring

CI/CD Integration

- Orange CI (Tencent Internal CI)
- Github Actions
- Coding CI/CD
- ...

The image displays a composite view of various CI/CD tools. At the top, there are three logos: GitHub Actions (blue play button icon), Jenkins (cartoon character holding a coffee cup), and Coding Cloud Development (monkey head icon). Below these are two screenshots. The left screenshot shows the Oragne CI (Tencent Internal CI) dashboard with a dark blue sidebar and a main area showing merge requests, issues, and other project management features. The right screenshot shows a merge request in the git.code.oo.com system, specifically for the 'orange-ci / nodejs-demo' repository. It shows a failed test named 'test: test fail' and a detailed log of the build process, including a 'Shell Script' step that failed with a 'sudo serverless deploy' command. The log output includes several deployment logs from Tencent Cloud services.

```
1488 74s > dashboard > Deploying ...
1489 aNA DEBUG - Website deployed successfully to URL: http://1251971143.cos-website.ap-guangzhou.myqcloud.com.
1490 74s > dashboard > Deploying ...
1491 aNA
1492 74s > dashboard > Deploying ...
1493 aNA
1494 74s > dashboard > Deploying ...
1495 74s > dashboard > Deploying ...
1496 aNA
1497 74s > dashboard > Deploying ...
1498 aNA
1499 dashboard:
1500 url: http://cgedsqk-47hdbq-1251971143.cos-website.ap-guangzhou.myqcloud.com
1501 env:
1502 apiUrl: https://service-pd0h8n3i-1251971143.gz.apigw.tencentcs.com/release
1503 api:
1504 region: ap-guangzhou
1505 functionName: tencent-fullstack-vue-api
1506 apiGatewayServiceId: service-pd0h8n3i
1507 url: https://service-pd0h8n3i-1251971143.gz.apigw.tencentcs.com/release/
1508 74s > dashboard > done
1509
1510
```

CI/CD & Monitoring

CI/CD Integration

- Orange CI (Tencent Internal CI)
- Github Actions
- Coding CI/CD
- ...

The screenshot shows the Orange CI interface for a deployment. At the top, there's a navigation bar with links for '发布部署' (Deployment), '需求列表' (Requirement List), and '发布详情' (Deployment Details). Below this, deployment details are listed:

- 单号: 5f3f4ffd4dbe0fe37c6585a8
- 分支: afext_thanos_test
- 责任人: aflextyang (highlighted by a blue arrow labeled 'Feature owner')
- TAPD: 辅导题库云函数升级serverless component&重新打TAG发布 (highlighted by a blue arrow labeled 'project url')
- 项目地址: k12_proj/scf_fudao_tiku imweb_serverless/imf-scf-demo (highlighted by a blue arrow labeled 'project url')
- 评审单: (empty)
- 环境占用: 已占用 (highlighted by a blue arrow labeled 'Feature owner')

Below the details is a '发布进度' (Deployment Progress) section with three stages: 'prepare', 'pre-release', and 'release'. The 'prepare' stage is completed, indicated by a blue dot. The 'pre-release' stage is in progress, indicated by a grey dot. The 'release' stage is not yet started, indicated by a white dot.

On the left, there's a legend for build types:

- Orange-CI CI构建
- SCF SCF 构建

On the right, there are two tables showing function details:

| k12_proj/scf_fudao_tiku | imweb_serverless/imf-scf-demo |
|----------------------------------|-------------------------------|
| Function name | version |
| IMFS_fudao-robot-test_wechattest | 125 2020-08-21 12:31:20 |

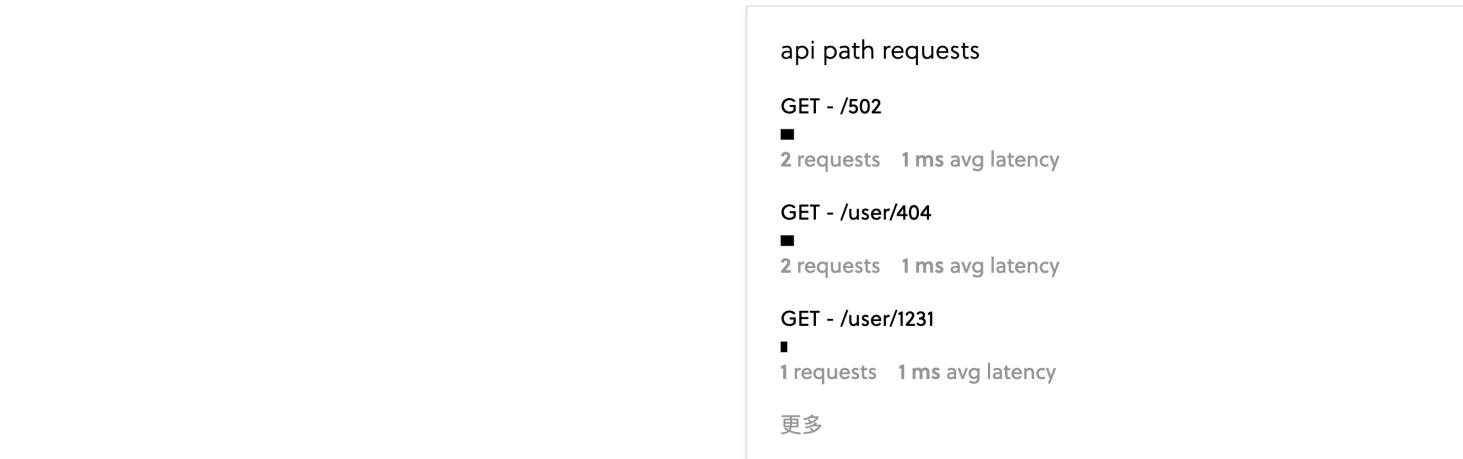
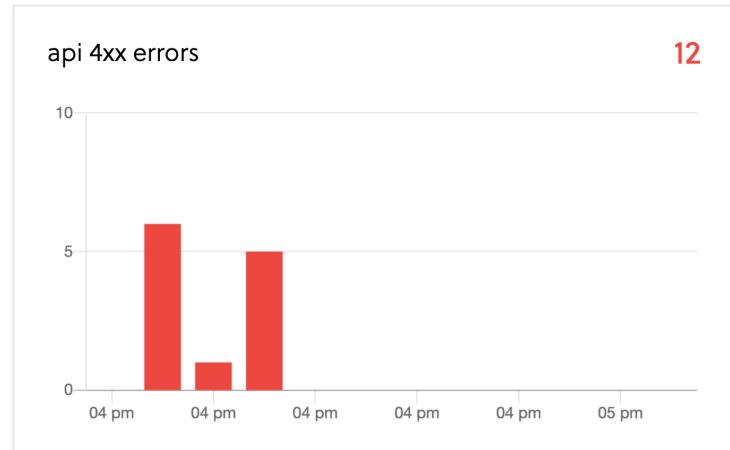
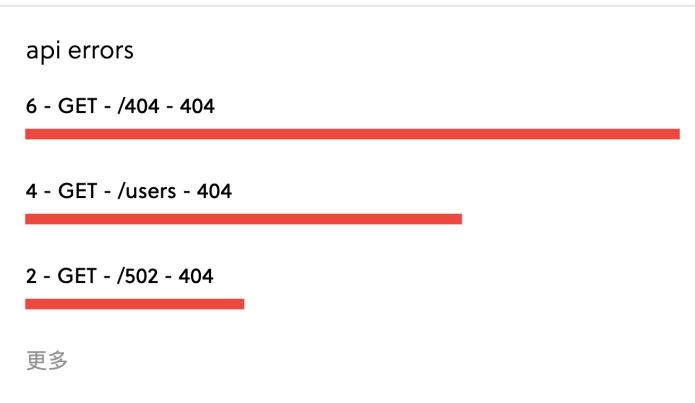
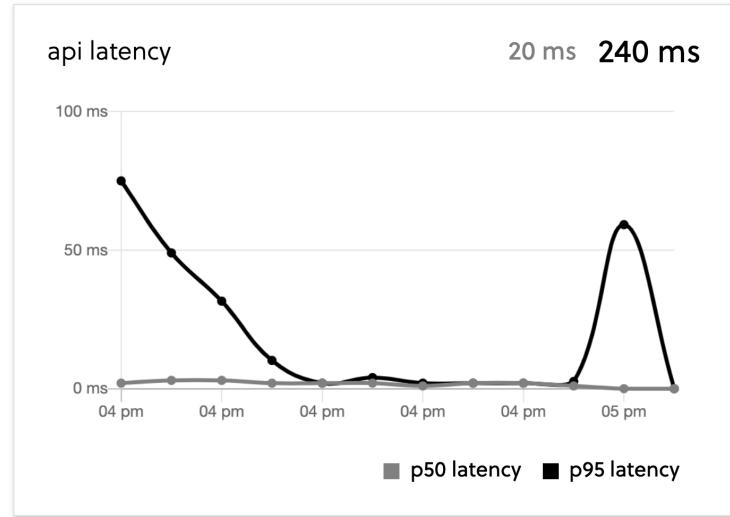
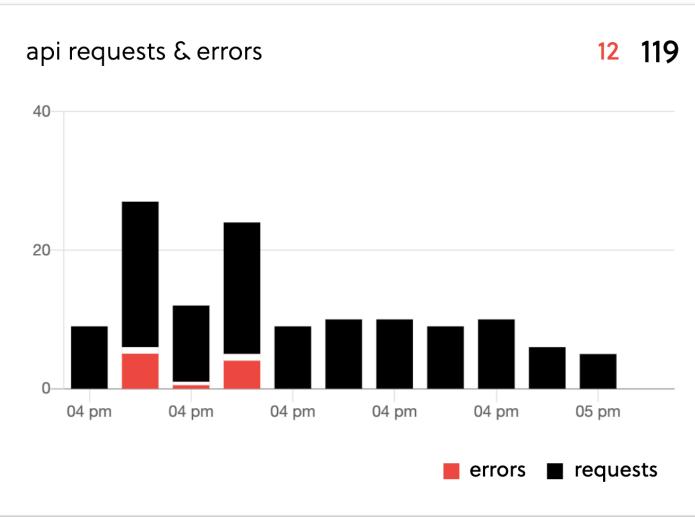
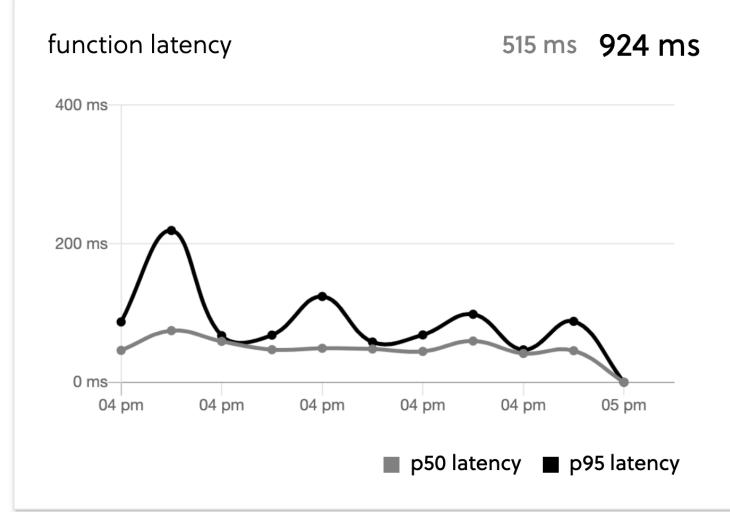
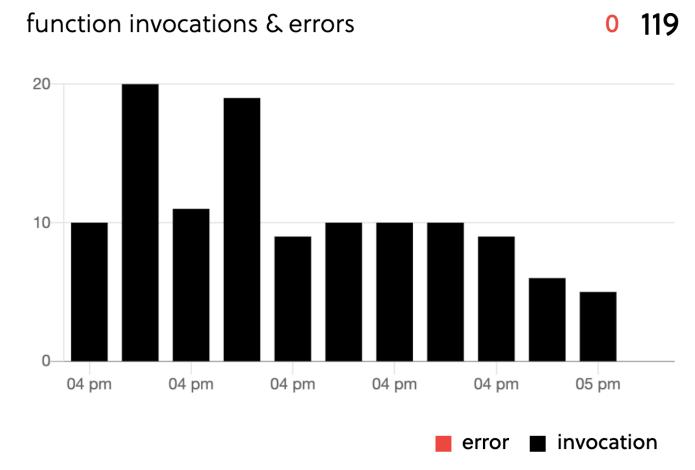
| imweb_serverless/imf-scf-demo | |
|-------------------------------|------|
| 函数名称 | 当前版本 |

CI/CD & Monitoring

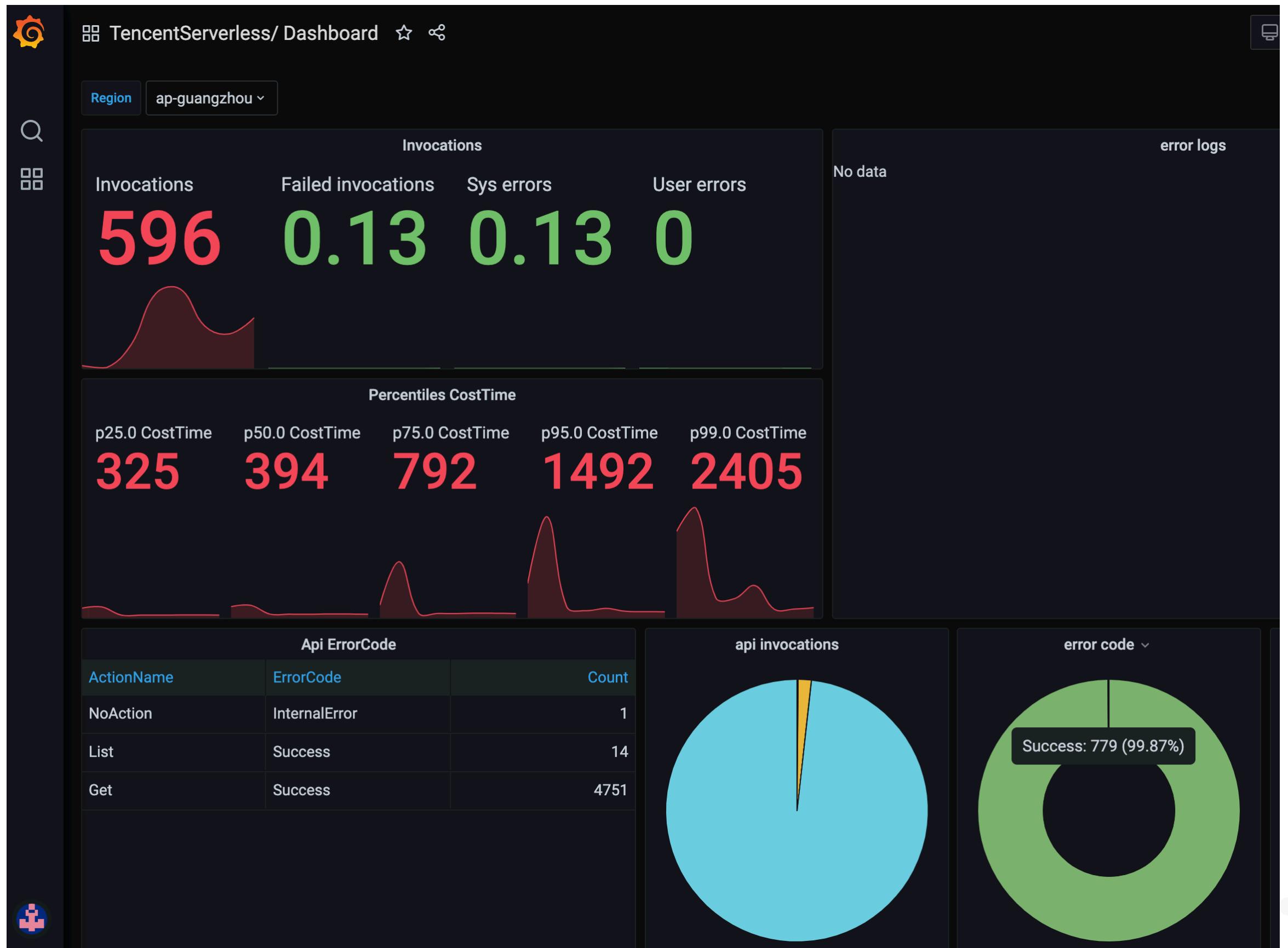
Application Monitoring for Next.js

- API Latency
- 4XX 5XX Status Codes
- API Requests & Errors
- API Path Requests & Errors
- ...

Data reported by an agent : ServerlessSDK



CI/CD & Monitoring



| | |
|------|----------|
| 持续时间 | 4分钟 |
| 告警渠道 | 邮件、微信 |
| 告警策略 | SF后台_云函数 |
| 所属项目 | 默认项目 |

监控数据 函数错误次数



告警记录

| 序列号 | 触发时间 | 恢复时间 |
|-----|---------------------|---------------------|
| 1 | 2020-10-14 09:04:00 | 2020-10-14 09:08:00 |

4. Recap & Summary



Recap & Summary

Next.js Functionalities

- ✓ Custom Express Server
- ✓ LRU Render Cache
- ✓ Redis & PWA for Caching

In the Future

- ✓ End to End Tracing
- ✓ Vendor Agnostic (AWS & Azure)
- ✓ Automatic Configuration

Serverless Combination

- ✓ FaaS and API Gateway to deploy Next.js
- ✓ COS to Separate Dynamic/Static Assets
- ✓ CDN Acceleration
- ✓ Layer to Deploy node_modules

Operations

- ✓ Monitoring & Alerting
- ✓ CI/CD Combination

Demo of Serverless Next.js

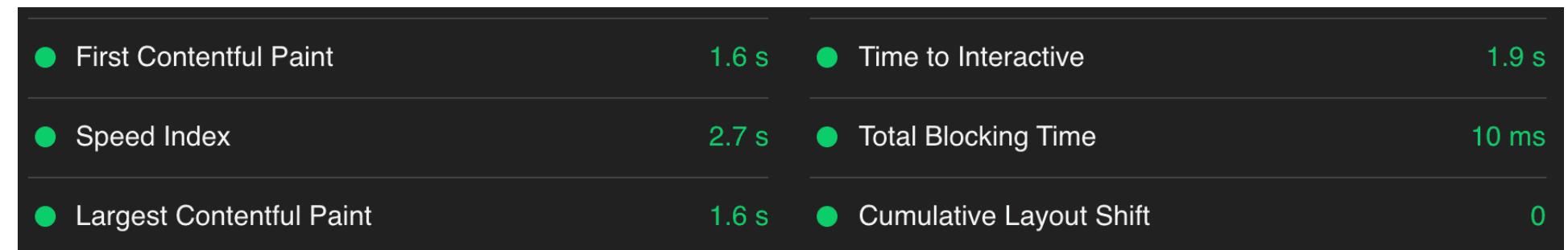
<https://github.com/serverless-plus/serverless-v2ex>

Lighthouse Performance

Before



After



The screenshot shows the homepage of v2ex.yuga.chat. The header includes navigation icons and the URL. Below the header is a blue navigation bar with the text "首页 - Serverless V2EX". The main content area displays a list of posts from various users. Each post includes a small profile picture, the title, and the author's name and timestamp.

- 罗技 MX Master 3 对手小的人来说真是噩梦
计算机 • 166953天前 • 0
- 请问有能实现这种聚合效果的 chrome/edge 插件吗?
Chrome • 166953天前 • 0
- access_log 日志中存在大量 AHC/1.0
程序员 • 166953天前 • 0
- 各大厂商路由器系统更新支持时间?
问与答 • 166953天前 • 0
- 腾讯云 11·11 续费活动低至 2 折操作
云计算 • 166953天前 • 0
- [日经求推荐手机] 原生 (至少流畅使用 google 服务) | 双卡 | 128G 或以上
Android • 166953天前 • 3
- 哪里有 Golang 的 websocket 框架对比
Go • 166953天前 • 1
- iperf 内网测速不理想, 求支招
问与答 • 166953天前 • 0
- 良心云又开始瞎搞活动了, 双 11 提前上线, s4 的 2-4-3 配置 3 年打完折
500 多块, 企业用户特殊活动数据库高防证书等产品白送
云计算 • 166953天前 • 0
- 请教大佬, Win10 有没有办法快速的把一个窗口移动到下一个虚拟桌面
上?
Windows • 166953天前 • 2



THANK YOU!

Q&A

