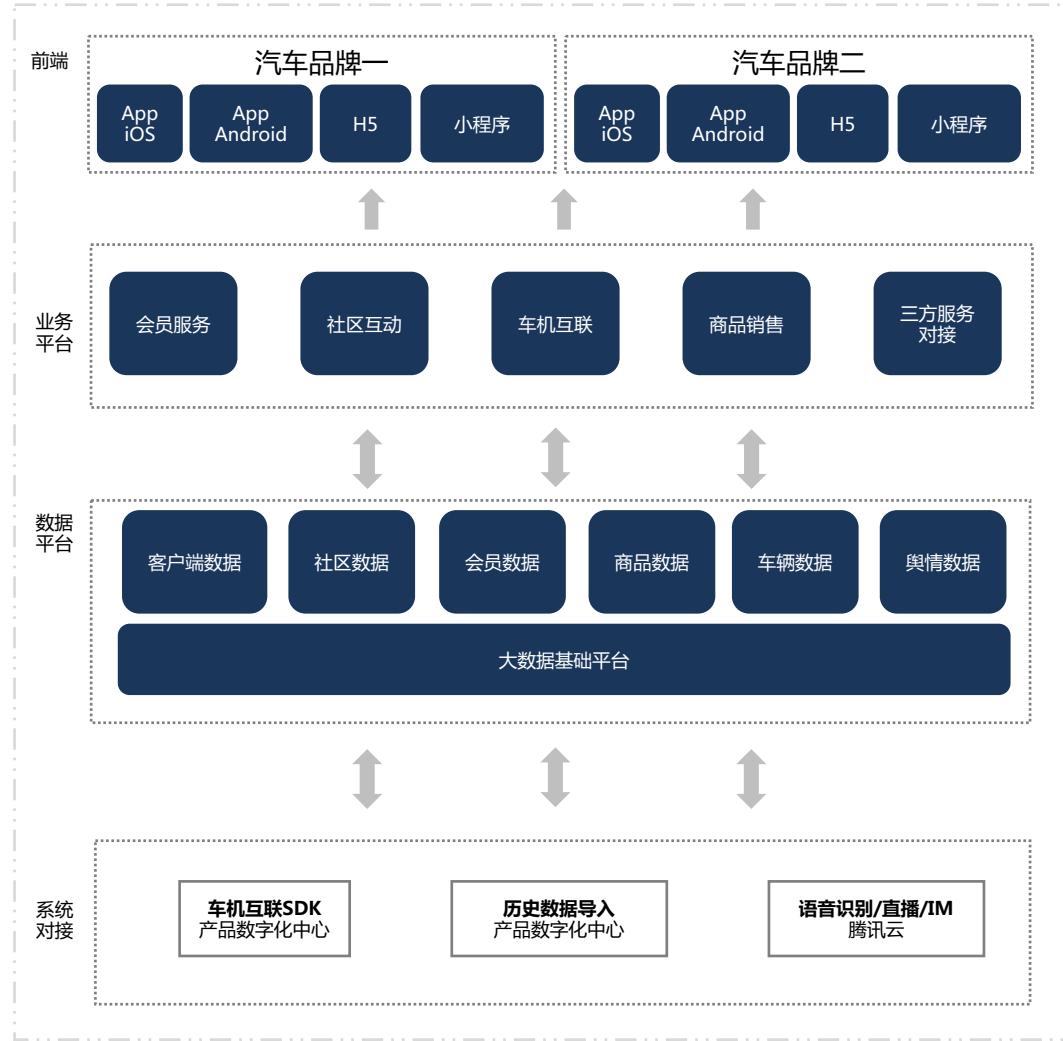


[目录 CONTENTS]

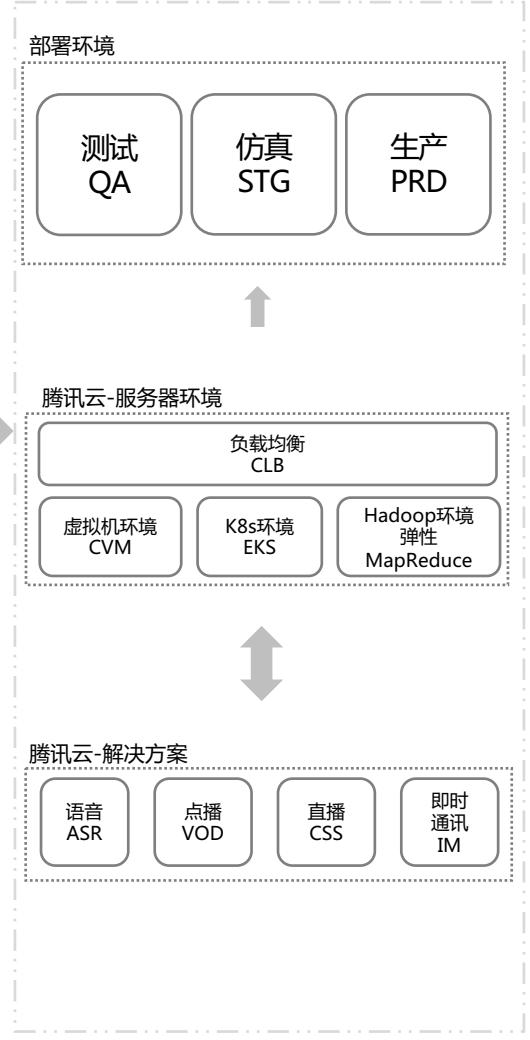
- 整体方案
- 业务平台
- 数据平台
- 前端技术架构
- 后端技术架构
- 服务器部署

整体技术方案

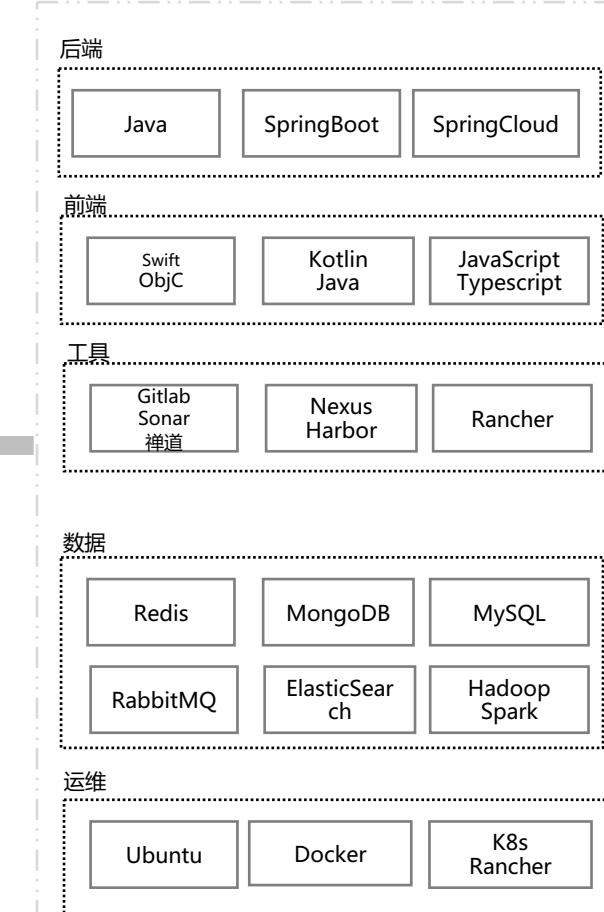
系统概览



系统部署



技术栈



业务平台架构

前台



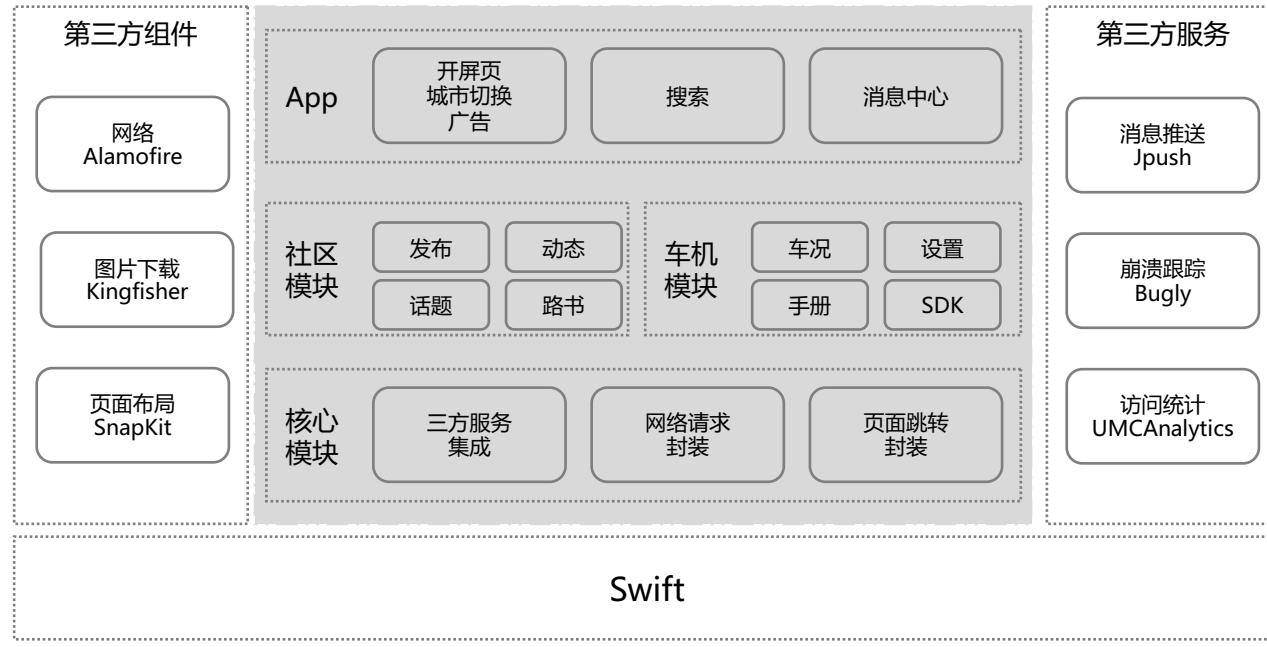
中台



后台



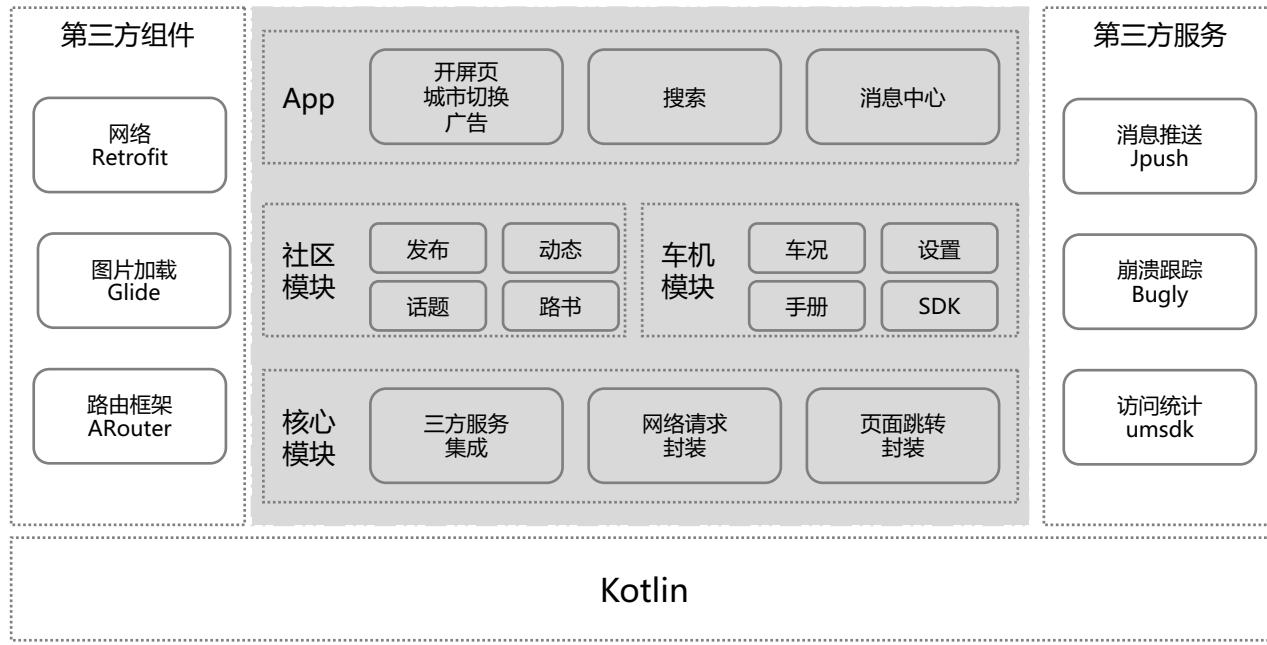
前端技术架构 - iOS



The screenshot shows the Xcode interface with the project structure and some code snippets:

- App-Core:** Shows files like App.swift, Device.swift, BaseViewModel.swift, etc.
- App-Ticket:** Shows files like App-Ticket, Grade, View, Model, ViewModel, App-Ticket.entitlements, RefundOrder, Order, Base, Confirm, ViewModel, TicketOrderCo...ViewModel.swift, CountdownTimer.swift, Model, View, Payment, SelectSeat, Supporting Files, Products, App-Ticket.app, Pods, Pods-App-Ticket.debug.xcconfig, Pods-App.release.xcconfig, Frameworks, and Pods_WandaFilm_Ticket.framework.
- App-Swift:** Shows files like App-Swift, Common, Misc, UserRemind, Splash, City, Model, CityLocation.swift, View, Cells, CityChooseTableHeaderView.swift, CityChooseTableHeaderView.xib, CityChooseViewController.swift, CityChooseViewController.xib, TitleDownArrowView.swift, ViewModel, CityManager.swift, CityChooseViewModel.swift, LocationManager.swift, Cinema, Advertise, View, PopAdvertiseView.swift, FullScreenAdvertiseViewController.swift, and ViewModel.
- AdvertiseViewModel:** A class definition for AdvertiseViewModel, which extends BaseViewModel. It includes methods for requestFullScreenAdvertiseDidSuccess and requestPopAdvertiseDidSuccess.

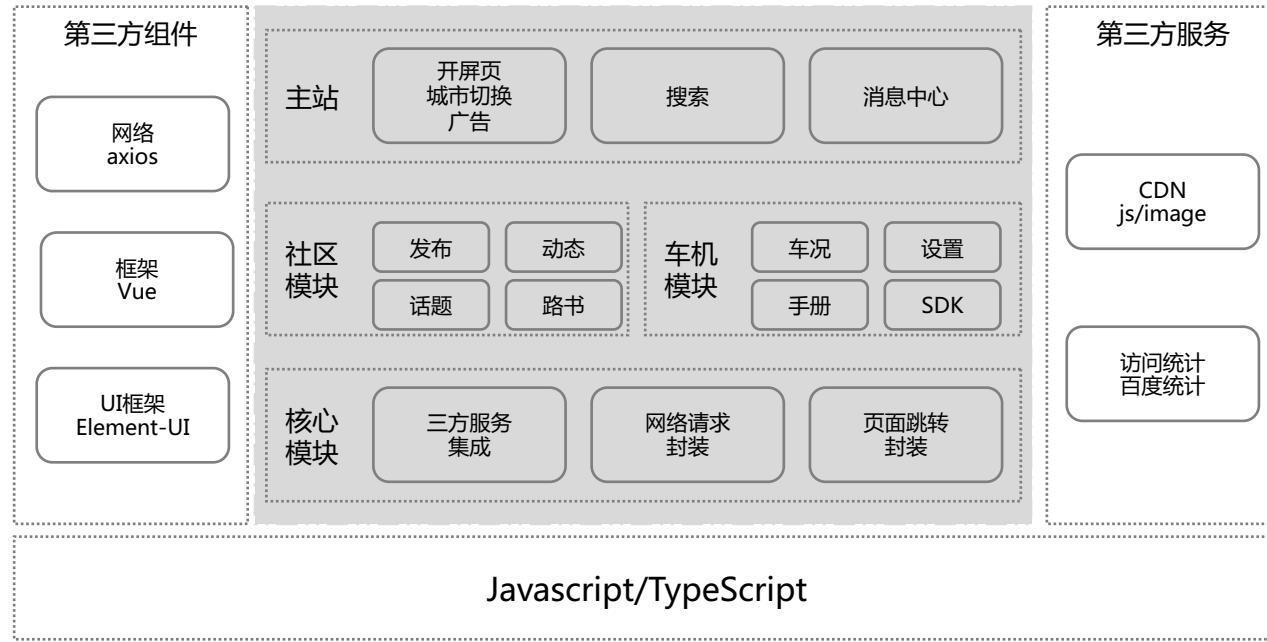
前端技术架构 - Android



The screenshot shows the Android project structure and some code snippets:

- Project Structure:** The project includes modules for MainApp, MallModule, PayModule, PersonModule, PresentModule, and ResourceModule. It uses JitPack for dependencies and includes build scripts like build.gradle and proguard-rules.pro.
- Code Snippets:**
 - MainApplication.java**: The application entry point, extending FrameApplication. It initializes skinService and handles onCreate().
 - GoodsListActivity.kt**: A Kotlin Activity for listing goods. It initializes variables like titleOfNormal, cinemaId, and saleCategoryId, and overrides onCreate() to handle intent data and initialize location manager.

前端技术架构 – H5



```

OPEN EDITORS
src > client > page > home > TS index.tsx > Home > render
1 import * as React from "react";
2 // import Icon from "antd/lib/icon";
3
4 import Page from "../../../../component/page/index";
5 // import Page from "../../../../component/page/index";
6 import Swiper from "../../../../component/carousel/index";
7 // import Footer from "../../../../component/footer/index";
8 import HotActivityItem from "../../../../component/hotActivityItem";
9 import ExcellentCreator from "../../../../component/excellentCreator";
10 import EducationList from "../../../../component/educationList/index";
11 import Address from "../../../../component/address/index";
12 import SendMessage from "../../../../component/sendMessage/index";
13 // import BackTop from "../../../../component/backTop/index";
14 import ActivityApi from "../../../../api/activity";
15 import CreatorApi from "../../../../api/creator";
16 import WorkApi from "../../../../api/work";
17 // import Spin from "antd/lib/spin";
18 import {activityStatus, activityType} from "../../../../constants/activity";
19 // import message from "antd/lib/message";
20 import {WORKMANAGER} from "../../../../utils/tn";
21 const WORKMANAGER = 30;
22 export interface IState {
23   activityData: any;
24   creatorData: any;
25   workData: any;
26   isShowMessage: boolean;
27   callCreator: any;
28   addFlag: any;
29   workParam: any;
30 }
31 }

class Home extends React.Component<{}, IState> {
  private css = require("./home.css");
  ...
}

constructor(props) {
  super(props);
  this.state = {
    activityData: [],
    creatorData: [],
    workData: [],
    isShowMessage: false,
    callCreator: {},
  };
}

OPEN EDITORS
src > client > page > login > render
1 import * as React from "react";
2 // import Icon from "antd/lib/icon";
3
4 import Page from "../../../../component/page/index";
5 // import Page from "../../../../component/page/index";
6 import Swiper from "../../../../component/carousel/index";
7 // import Footer from "../../../../component/footer/index";
8 import HotActivityItem from "../../../../component/hotActivityItem";
9 import ExcellentCreator from "../../../../component/excellentCreator";
10 import EducationList from "../../../../component/educationList/index";
11 import Address from "../../../../component/address/index";
12 import SendMessage from "../../../../component/sendMessage/index";
13 // import BackTop from "../../../../component/backTop/index";
14 import ActivityApi from "../../../../api/activity";
15 import CreatorApi from "../../../../api/creator";
16 import WorkApi from "../../../../api/work";
17 // import Spin from "antd/lib/spin";
18 import {activityStatus, activityType} from "../../../../constants/activity";
19 // import message from "antd/lib/message";
20 import {WORKMANAGER} from "../../../../utils/tn";
21 const WORKMANAGER = 30;
22 export interface IState {
23   activityData: any;
24   creatorData: any;
25   workData: any;
26   isShowMessage: boolean;
27   callCreator: any;
28   addFlag: any;
29   workParam: any;
30 }
31 }

class Login extends React.Component<{}, IState> {
  private css = require("./login.css");
  ...
}

constructor(props) {
  super(props);
  this.state = {
    activityData: [],
    creatorData: [],
    workData: [],
    isShowMessage: false,
    callCreator: {},
  };
}

OPEN EDITORS
src > client > router > TS router.tsx > ...
15 import personalInformation from "../../../../page/personalHome/index";
16 import userHome from "../../../../page/userHome/index";
17 import contact from "../../../../page/contact/index";
18 import help from "../../../../page/help/index";
19 import policies from "../../../../page/policies/index";
20 import userAgreement from "../../../../page/userAgreement/index";
21 import copyright from "../../../../page/copyright/index";
22 import college from "../../../../page/college/index";
23
24 function RouterConfig(history, app: {history: any, app: any}) {
25   const ConnectedRouter = routerRedux.ConnectedRouter;
26   return (
27     <ConnectedRouter history={history}>
28       <Switch>
29         <Route path="/" component={Home}/>
30         <Route path="/login" component={Login}/>
31         <Route path="/stories" component={worksChannel}/>
32         <Route path="/activity" component={activity}/>
33         <Route path="/copyright" component={college}/>
34         <Route path="/messageSystem" component={messageSystem}/>
35         <Route path="/search" component={search}/>
36         <Route path="/createRank" component={createRank}/>
37         <Route path="/follow" component={follow}/>
38         <Route path="/myActivity" component={myActivity}/>
39         <Route path="/myProduction" component={production}/>
40         <Route path="/collect" component={collect}/>
41         <Route path="/personalInformation" component={personalInformation}/>
42         <Route path="/useHome/:id" component={userHome}/>
43         <Route path="/contact" component={contact}/>
44         <Route path="/help" component={help}/>
45         <Route path="/policies" component={policies}/>
46         <Route path="/userAgreement" component={userAgreement}/>
47         <Route path="/userCopyright" component={copyright}/>
48         <Route path="/" component={Home}/>
49       </Switch>
50     </ConnectedRouter>
51   );
}

OPEN EDITORS
src > client > router > TS router.tsx > ...
15 import personalInformation from "../../../../page/personalHome/index";
16 import userHome from "../../../../page/userHome/index";
17 import contact from "../../../../page/contact/index";
18 import help from "../../../../page/help/index";
19 import policies from "../../../../page/policies/index";
20 import userAgreement from "../../../../page/userAgreement/index";
21 import copyright from "../../../../page/copyright/index";
22 import college from "../../../../page/college/index";
23
24 function RouterConfig(history, app: {history: any, app: any}) {
25   const ConnectedRouter = routerRedux.ConnectedRouter;
26   return (
27     <ConnectedRouter history={history}>
28       <Switch>
29         <Route path="/" component={Home}/>
30         <Route path="/login" component={Login}/>
31         <Route path="/stories" component={worksChannel}/>
32         <Route path="/activity" component={activity}/>
33         <Route path="/copyright" component={college}/>
34         <Route path="/messageSystem" component={messageSystem}/>
35         <Route path="/search" component={search}/>
36         <Route path="/createRank" component={createRank}/>
37         <Route path="/follow" component={follow}/>
38         <Route path="/myActivity" component={myActivity}/>
39         <Route path="/myProduction" component={production}/>
40         <Route path="/collect" component={collect}/>
41         <Route path="/personalInformation" component={personalInformation}/>
42         <Route path="/useHome/:id" component={userHome}/>
43         <Route path="/contact" component={contact}/>
44         <Route path="/help" component={help}/>
45         <Route path="/policies" component={policies}/>
46         <Route path="/userAgreement" component={userAgreement}/>
47         <Route path="/userCopyright" component={copyright}/>
48         <Route path="/" component={Home}/>
49       </Switch>
50     </ConnectedRouter>
51   );
}

```

后端技术架构



服务器部署架构

接入层



服务层



运维层



环境	腾讯云服务	配置	数量	备注
研发测试 (QA)	CVM	4C/8G/50G	8	服务组件部署
研发测试 (QA)	EKS	256C/512G/2T	1	服务部署
研发测试 (QA)	TencentDB for MySQL	高可用版(8C/32G/200G)	8	数据库
研发测试 (QA)	TencentDB for Redis	内存版·集群架构(4GB/2节点)	8	缓存
研发测试 (QA)	CMQ			仅内网使用
研发测试 (QA)	Elasticsearch Service	8C/32G/2T/3节点	3	前台搜索/后台搜索/日志
研发测试 (QA)	弹性MapReduce	Master : 4C/8G/云盘100G Core : 4C/8G/云盘200G Common : 2C/4G/云盘100G	1	数据平台
研发测试 (QA)	COS	标准存储(多AZ, 200G)	1	
仿真 (STG)	CVM	4C/8G/50G	16	服务组件部署
仿真 (STG)	EKS	256C/512G/2T	2	服务部署
仿真 (STG)	TencentDB for MySQL	高可用版(8C/32G/200G)	16	数据库
仿真 (STG)	TencentDB for Redis	内存版·集群架构(4GB/2节点)	16	缓存
仿真 (STG)	CMQ			仅内网使用
仿真 (STG)	Elasticsearch Service	8C/32G/2T/3节点	3	前台搜索/后台搜索/日志
仿真 (STG)	弹性MapReduce	Master : 8C/32G/云盘100G Core : 8C/32G/云盘2000G Common : 2C/4G/云盘100G	1	数据平台
仿真 (STG)	COS	标准存储(多AZ, 200G)	1	
生产 (STG)	CVM	4C/8G/50G	16	服务组件部署
生产 (STG)	EKS	256C/512G/2T	2	服务部署
生产 (STG)	TencentDB for MySQL	高可用版(8C/32G/200G)	16	数据库
生产 (STG)	TencentDB for Redis	内存版·集群架构(4GB/2节点)	16	缓存
生产 (STG)	CMQ			仅内网使用
生产 (STG)	Elasticsearch Service	8C/32G/2T/3节点	3	前台搜索/后台搜索/日志
生产 (STG)	弹性MapReduce	Master : 8C/32G/云盘100G Core : 8C/32G/云盘2000G Common : 2C/4G/云盘100G	1	数据平台
生产 (STG)	COS	标准存储(多AZ, 200G)	1	
生产 (STG)	CDN	100M	1	
生产 (STG)	CLB	100M	1	
生产 (STG)	ASR		1	语音
生产 (STG)	VOD		1	点播
生产 (STG)	CSS		1	直播
生产 (STG)	IM		1	即时通讯
运维 (OP)	CVM	4C/8G/50G	12	Gitlab/Maven/Habor/DNS/禅道/Nexus/监控/日志/CI/CD/跳板机
运维 (OP)	EKS	64C/128G	1	Gitlab/Maven/Habor/DNS/禅道/Nexus/监控/日志/CI/CD/跳板机
运维 (OP)	TencentDB for MySQL	高可用版(8C/32G/200G)	2	Gitlab/Maven/Habor/DNS/禅道/Nexus/监控/日志/CI/CD/跳板机
运维 (OP)	CMQ			仅内网使用
运维 (OP)	Elasticsearch Service	8C/32G/2T/3节点	1	

附录

性能问题

基本原则

- 动静分离
- 读写分离
- 数据分片
- 按需获取
- 批量操作
- 减少I/O

缓存

- 多级缓存
- 本地缓存
- 主动缓存

服务调用

- 方法幂等
- 异步调用
- 结果缓存

稳定问题

冗余

不允许单点
可快速扩容

限流熔断

限制服务连接数
直接丢弃

失败重试

超时时间
重试次数
重试节点

安全问题

数据安全

- 敏感数据不可逆加密存储，关键数据混淆（遮蔽/变形/收缩）存储
- 后台页面操作行为，生产环境登录、操作行为记录并审计
- API接口请求响应数据加密传输，图片、视频数据添加盲水印

账户安全

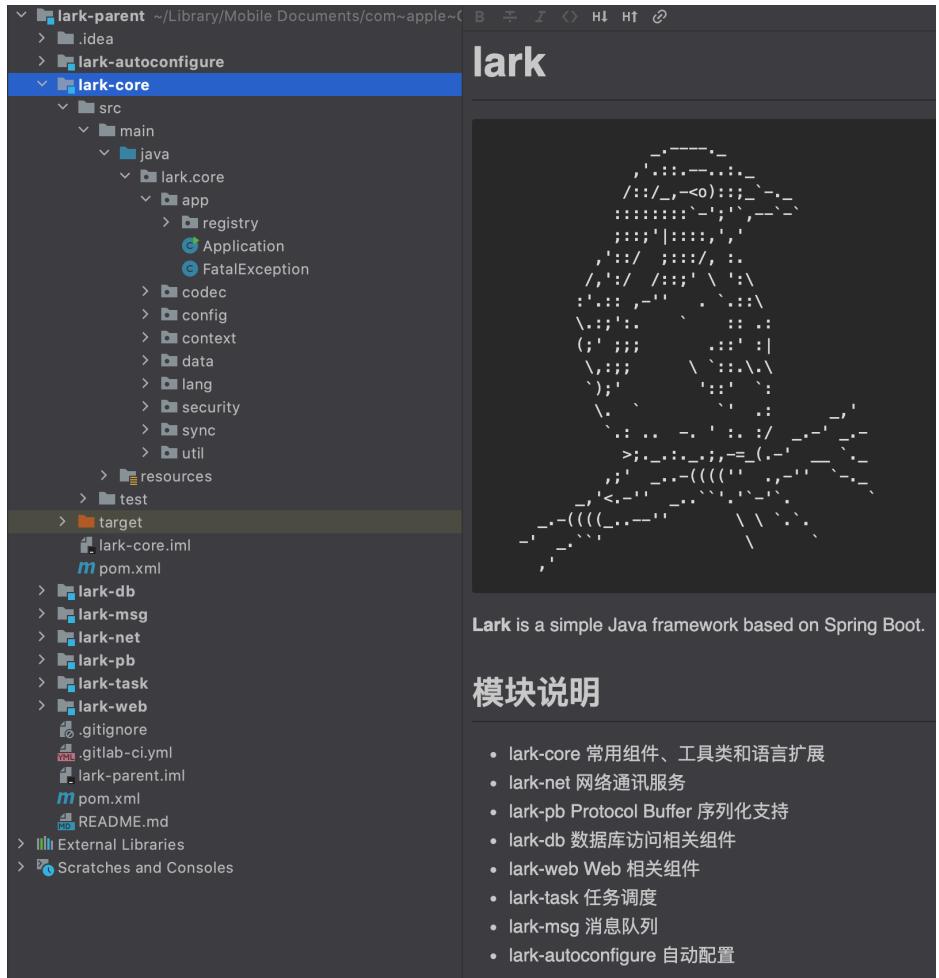
- 登录管理：异地登录提醒，更换手机号、城市后强制重新登录
- 资产变动：支付密码验证
- 异常监控：频繁登录（IP/手机号）报警，频繁交易报警（IP/手机号/商品）

网络安全

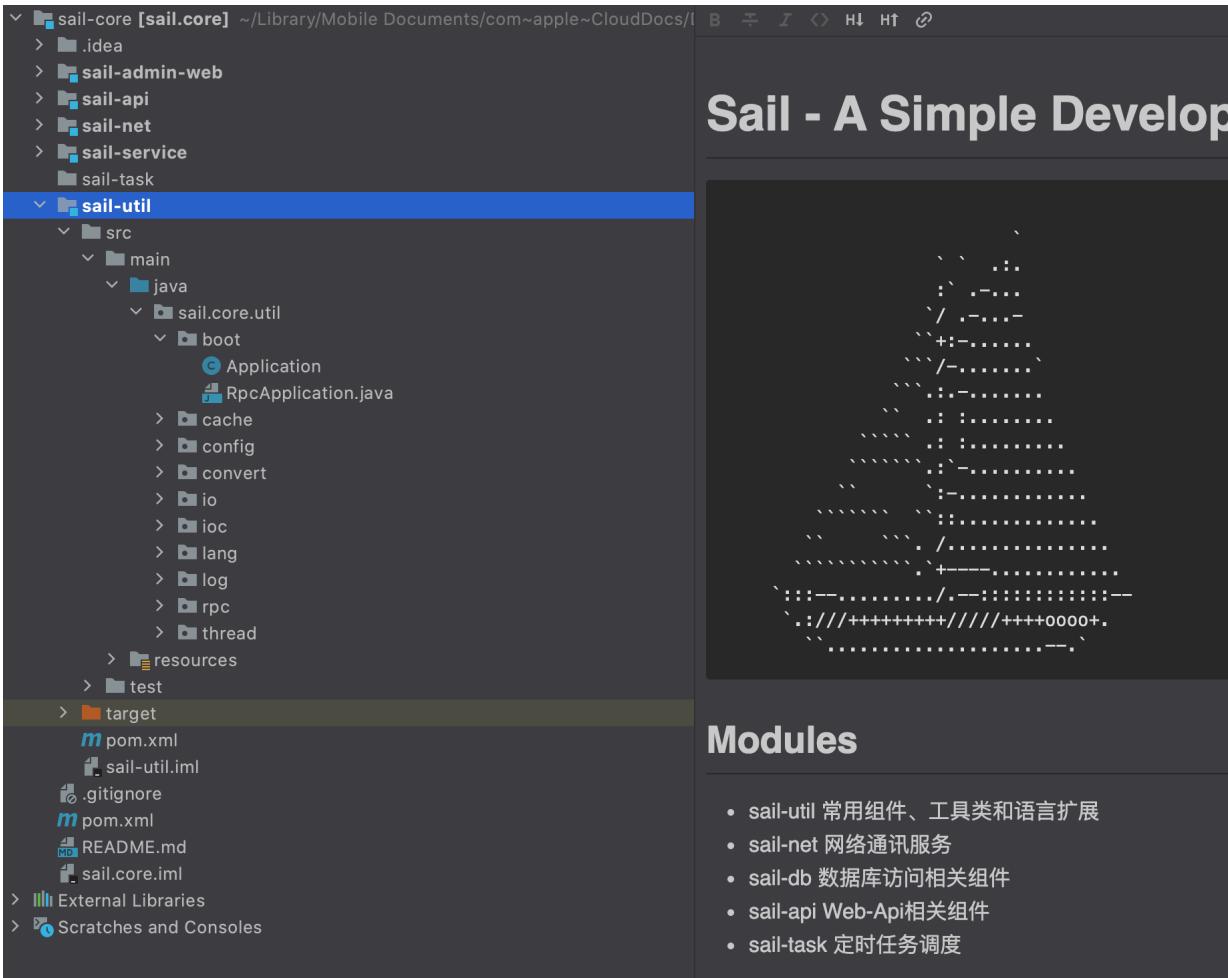
- 生产环境管控：入口服务仅开放必要端口（80/443），所有服务禁止访问外网
- 办公环境管控：禁止直连生产环境，禁止通过堡垒机上传、下载文件
- DDoS/CC攻击：采购高防IP服务

后端技术架构 – 微服务框架

Lark – 整体概览



Sail – 整体概览



后端技术架构 – 微服务框架

Lark – 服务定义与调用

```
/**  
 * 测试服务  
**/  
  
public interface TestService {  
    /**  
     * 测试  
    **/  
    HelloResponse hello(HelloRequest request);  
}  
  
@Component  
@RpcService(name = "TestService", description = "测试服务")  
public class TestServiceImpl implements TestService {  
    @Autowired  
    private TestBiz testBiz;  
  
    @Override  
    public TestDto.HelloResponse hello(TestDto.HelloRequest request) {  
        TestObject object = testBiz.getObject(request.getId());  
        TestDto.HelloResponse response = new TestDto.HelloResponse();  
        response.setResult(object.getName());  
        return response;  
    }  
}  
  
@Autowired  
TestService testService;  
public void testHello() throws Exception {  
    TestDto.HelloRequest request = new TestDto.HelloRequest();  
    request.setId(123);  
    request.setType(TestType.GOOD);  
    request.setTime(LocalDateTime.now());  
  
    TestDto.HelloResponse response = testService.hello(request);  
}
```

Sail – 服务定义与调用

```
@RestController  
public interface TokenService {  
    /**  
     * 获取用户Token  
     * @param id 用户ID  
    **/  
    @PostMapping(value = "/token.srv")  
    @ResponseBody  
    TokenServiceDto.  
    TokenResponse getAuthToken( @RequestParam("id")  
                                String id );  
}  
  
@Service  
public class TokenServiceImpl implements TokenService {  
    Logger log = LogManager.getLogger( TokenServiceImpl.class );  
    @Autowired  
    private TokenBiz tokenBiz;  
  
    /**  
     * 获取用户Token  
     *  
     * @param id  
     *      用户ID  
    **/  
    @Override  
    public TokenServiceDto.TokenResponse getAuthToken(String id) {  
        log.info( msg: ">>> sail.demo.service --> TestServiceImpl -> getAuthToken:begin >>>" );  
        TokenInfo tokenInfo = tokenBiz.getToken( id );  
        if ( tokenInfo != null ) {  
            return tokenBiz.toTokenResponse( tokenInfo );  
        }  
        return null;  
    }  
}  
  
@Service  
public class AuthBiz {  
    @Autowired  
    private TokenService tokenSrv;  
    /**  
     * @param id  
     * @return  
    **/  
    public TokenInfo getAuthToken(String id) {  
        TokenServiceDto.TokenResponse tokenResponse = tokenSrv.getAuthToken(id);  
        if ( tokenResponse != null ) {  
            return new TokenInfo(tokenResponse getToken(), DateTimeKit.getUtcDate());  
        }  
    }  
}
```

后端技术架构 – 部署示意

CI 持续集成

All 193 Finished Branches Tags

Run Pipeline Clear Runner Caches CI Lint

Status Pipeline Triggerer Commit Stages

#8026 latest 调整用户订单列表 00:03:34 2 weeks ago

Job #20554 triggered 2 weeks ago by 高爽

Duration: 3 minutes 17 seconds
Timeout: 1h (from project)
Runner: docker-runner (#5)

Job artifacts

Commit f8f5e20b 调整用户订单列表

Pipeline #8026 for 2.0

build

→ build

CD 持续交付

Job #20555 triggered 2 weeks ago by 高爽

This job is deployed to dev.cotaiticketing.xtech.fun.

Duration: 16 seconds
Timeout: 1h (from project)
Runner: docker-runner (#5)
Tags: docker

Job artifacts

Commit f8f5e20b 调整用户订单列表

Pipeline #8026 for 2.0

deploy-dev

→ deploy-dev

SWIRL 首页 单机 集群 系统

SWIRL v0.8.5(git.12.4)
一个Docker管理工具，专注于Swarm集群。

GitHub

节点 10 网络 7 服务 16 待办 0

SWIRL 首页 单机 集群 系统

10 节点

名称	Role	Version	CPU	Memory	地址	状态	操作
VM-255-10-ubuntu	manager	19.03.5	2	3.70 GB	172.27.255.10	ready	<button>编辑</button> <button>删除</button>
VM-255-14-ubuntu	worker	19.03.5	8	31.26 GB	172.27.255.14	ready	<button>编辑</button> <button>删除</button>
VM-255-15-ubuntu	manager	19.03.5	2	3.70 GB	172.27.255.15	ready	<button>编辑</button> <button>删除</button>
VM-255-23-ubuntu	worker	19.03.5	8	31.26 GB	172.27.255.23	ready	<button>编辑</button> <button>删除</button>