

# 腾讯IEG · 国际客户技术交付（育碧项目） — STAR法则深度拆解

岗位：腾讯IEG 基础技术产品部 服务拓展组 · AI技术产品实习生

时间：2026.02 - 至今

项目：育碧（Ubisoft）海外旗舰项目接入腾讯GCloud技术矩阵

## 一、项目全景（面试官问“你在腾讯做什么”时，先铺全景再展开）

### Why — 为什么要做这件事？

育碧是全球头部游戏发行商，其海外旗舰项目选择接入腾讯GCloud技术矩阵作为底层基础设施。GCloud技术矩阵为游戏提供**下载更新（Dolphin/Puffer）**、**区服导航（Maple）**、**语音通信（GVoice）**等核心组件。

但海外环境和国内有本质差异：

- 网络基建复杂**：海外CDN覆盖不均，需要整合Akamai、Zenlayer、Cloudflare、AWS、GCP等多家加速厂商
- 合规要求严格**：GDPR等数据隐私法规约束数据传输和存储
- 平台多样性高**：需适配PC、主机（PS4/PS5/Xbox/Switch）、移动端等全平台
- 组件耦合风险大**：Dolphin/Puffer/Maple/GVoice 四大组件各有客户端和服务端，任何一个组件出问题都会阻塞整体交付

一句话总结 Why：海外旗舰项目对底层技术组件的稳定性要求极高，但海外环境的复杂性使得每个组件都面临适配风险，必须逐项排查、逐项验收。

### How — 怎么做的？

我作为**技术接口人（Technical PM）**，主导了四大核心组件的全链路风险审查与联调交付流程：

#### 第一步：建立组件级风险审查体系（TDR - Technical Delivery Review）

我主导设计并落地了一套**组件级风险审查表（TDR-GCloud-ComponentRiskReview）**，覆盖4大组件、152+检查项，按以下维度逐项审查：

组件	审查范围	检查项数量	覆盖内容
Dolphin（下载更新）	客户端 + 服务端	~58项	APK更新策略、首包解压/修复、资源包大小限制、CDN预热、SNI支持、TDM数据上报、错误码展示
Maple（区服导航）	客户端 + 服务端	~15项	客户端基础逻辑、TreeID一致性、配置一致性、国内版/海外版服务端逻辑、状态上报

组件	审查范围	检查项数量	覆盖内容
Puffer (动态资源更新)	客户端 + 服务端	~38项	请求频率、正式域名、并发连接数、资源修复逻辑、批量下载、备用CDN源、SNI支持、DMP2.0协议、主机端存储预留
GVoice (语音通信)	客户端	~30项	权限校验、麦克风权限、iOS退后台机制、退房/断线重连、语音消息防截断、STT转文字、翻译并发、声纹/账号体系、谷歌翻译计费

每个检查项设置了：

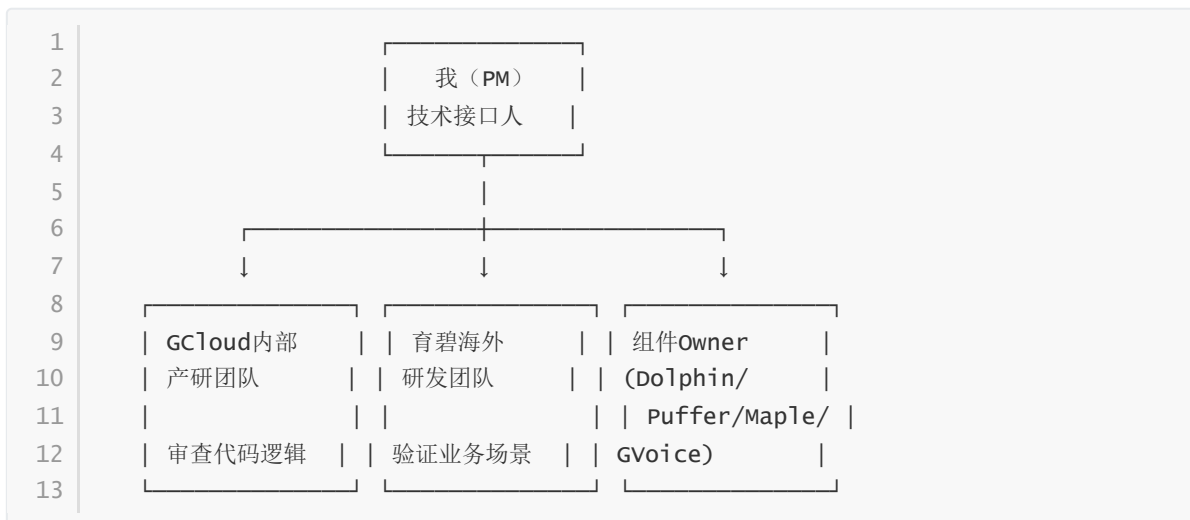
- **状态标签**：Pending / Passed / Failed / RiskPresent / NotInUse
- **GCloudTeam反馈列**：内部团队的审查结论
- **TODO列**：待确认/待验证的具体事项
- **闭环责任人**：明确每项的owner（内部产研 or 海外业务侧QA）

## 第二步：分类分级管理，推动逐项闭环

我将所有审查项按紧急程度分为四级：

级别	定义	处理策略	典型案例
<span style="color: red;">●</span> <b>Failed</b>	业务代码有明确错误	立即修复，阻塞交付	Puffer资源修复逻辑错误 (Row81-83)、Dolphin数据上报漏调接口 (Row17/20)、海外CDN SNI支持不通过 (Row92-94)
<span style="color: yellow;">●</span> <b>Pending</b>	状态待确认，缺少反馈	催办 + 拉通确认	Dolphin服务端全线6项 (Row53-58)、Maple服务端10项 (Row64-74)、GVoice账号体系与节点调度 (Row115-117)
<span style="color: green;">●</span> <b>Passed+TODO</b>	代码无误但有验证缺口	转交业务侧QA闭环	Dolphin资源修复需QA实际触发测试、Puffer iOS低端机并发需压测、GVoice谷歌翻译计费需商务确认
<span style="color: gray;">○</span> <b>NotInUse</b>	业务未接入	标记忽略	Switch主机端Puffer、QTS全系列组件 (业务未使用QTS框架)

### 第三步：拉通多方协作，建立问题升级机制



#### 我的协调动作：

1. **周级同步**：每周汇总各组件审查进度，更新状态表，同步内外部
2. **问题升级**：Pending超过1周末反馈的项，升级到组件负责人层面
3. **闭环定义**：为每个"Passed+TODO"项明确定义闭环标准——哪些由内部产研确认（代码逻辑层面），哪些转交业务侧QA（实际功能验证层面），哪些需要商务确认（如谷歌翻译计费归属）
4. **海外适配专项**：针对海外特有（SNI支持、CDN厂商切换、GDPR合规、多语种翻译）设立专项跟进

### What — 做出了什么结果？

指标	数据
覆盖组件	4大核心组件（Dolphin + Puffer + Maple + GVoice），客户端+服务端
总审查项	<b>152+ 检查项</b> ，逐项审查并分类管理
推动解决	<b>数十项技术卡点</b> ，包括代码逻辑错误、接口漏调、SNI不兼容、配置缺失等
交付成果	保障国际项目底层架构 <b>按期交付</b>
沉淀产出	TDR风险审查表 → 可复用模板，后续其他国际项目可直接套用

## 二、面试官深挖问题 & 标准回答

### Q1: 技术卡点具体有哪些？举几个例子？

答：举三个典型的不同类型的问题：

1. **代码逻辑错误 (Failed级)**：Puffer的资源修复逻辑存在Bug——当下载中断后重连时，资源修复流程没有正确回退到断点，导致需要从头重新下载。这直接影响玩家的更新体验，尤其在海外网络不稳定的情况下更为严重。
2. **海外网络适配问题 (Failed级)**：海外CDN的SNI (Server Name Indication) 支持不通过——部分海外CDN节点不支持SNI扩展，导致HTTPS连接握手失败。这需要和CDN厂商协调配置，同时在客户端增加SNI降级兜底逻辑。

3. **多方确认型问题 (Pending级)**：GVoice的谷歌翻译计费归属——游戏内语音同传功能调用了Google Translate API，但计费是走GCloud账号还是育碧自己的Google Cloud账号？这需要内部GCloud商务和育碧商务共同确认。

## Q2: 152+检查项是怎么设计出来的？

答：不是我从零发明的，而是基于三个来源的整合：

1. **GCloud各组件的标准接入文档**——每个组件 (Dolphin/Puffer/Maple/GVoice) 都有自己的接入checklist
2. **历史项目踩过的坑**——和团队中有经验的同事沟通，把之前国际项目中出过的问题都补进去
3. **海外环境特有的风险点**——CDN适配、SNI支持、多语种翻译、主机平台存储预留、GDPR合规等

然后我把它们按组件×客户端/服务端×风险等级三个维度结构化，形成了TDR表格。

## Q3: 遇到的最大困难是什么？怎么解决的？

答：最大的困难是**Pending项的推进**——很多审查项处于"等内部反馈"状态，但对方团队也有自己的优先级，不一定及时响应。

我的解决方法：

1. **可视化进度**：把整体进度做成看板，每周同步给上级和各组件owner，让所有人看到"哪些项卡住了、卡了多久"
2. **升级机制**：Pending超过1周的，我会直接拉对应组件的负责人进群对齐，而不是在底层继续催
3. **分类降本**：把"其实不需要内部确认、业务侧可以自行验证"的项提前拆出来，减少内部排队时间。比如很多"Passed+TODO"项，本质上代码逻辑已经pass了，只是需要业务侧QA做实际功能验证——这类我直接定义清楚闭环标准后转给育碧研发团队

## Q4: 验收标准怎么制定的？

答：验收分三层：

1. **代码逻辑层 (GCloudTeam负责)**：审查业务代码中对SDK接口的调用是否正确、参数是否合规、异常分支是否处理
2. **功能验证层 (业务侧QA负责)**：在真实设备和海外网络环境下，实际触发各功能场景，验证是否正常工作
3. **配置与合规层 (Ops+商务负责)**：确认控制台配置 (CDN节点、产品ID绑定等) 与实际一致，以及计费、数据合规等商务问题

## Q5: 你和组件开发者的分工是什么？你的价值在哪？

答：组件开发者负责"技术判断"——代码对不对、接口设计合不合理。我的角色是**全局协调和风险管理**：

1. **把散落在各组件各团队的信息汇聚成一张表**——之前每个组件team各自审查，没有统一视图，也没有统一标准
2. **定义什么算"闭环"**——很多项处于"Passed但留了TODO"的灰色地带，没人明确定义谁来收口。我把闭环责任拆清楚了

3. **推动跨团队对齐**——GVoice的问题可能需要Maple配合（比如区服切换后语音房间的重连），这种跨组件的依赖关系需要有人全局把控
4. **沉淀可复用资产**——TDR表格模板化后，下一个国际项目可以直接套用，不用从零开始

## Q6: 海外环境和国内环境的核心差异？

答：技术层面有三大差异：

1. **网络架构差异**：国内大部分走腾讯自有CDN，海外需要多厂商混合调度（Akamai/Cloudflare/AWS等），且部分地区SNI支持不完善
2. **平台差异**：海外主机（PS/Xbox/Switch）占比高，主机端的存储管理、安装包限制和移动端完全不同。比如Switch存储空间极其有限，需要精确控制资源包大小
3. **合规差异**：GDPR要求用户数据不能随意跨境传输，GVoice的语音数据处理和存储都需要满足合规要求；谷歌翻译API的调用也涉及计费和数据合规问题

## Q7: 这个项目对你的能力有什么提升？

答：三个核心提升：

1. **系统性思维**：从“看一个组件”到“看一个技术矩阵”——理解了Dolphin/Puffer/Maple/GVoice之间的依赖关系和整体架构
2. **跨团队协调能力**：同时对接GCloud内部多个组件团队 + 育碧海外研发团队 + 商务/运维，练就了多方拉通的能力
3. **风险管理方法论**：学会了怎么把“一堆零散的问题”结构化为“可追踪、可升级、可闭环”的体系

## 三、关键术语速查（面试被问到时不能卡壳）

术语	解释	在项目中的场景
<b>Dolphin</b>	GCloud启动更新组件，负责游戏安装包的下载和版本更新	审查APK更新策略、首包解压/修复、资源包4GB限制等
<b>Puffer</b>	GCloud动态资源更新组件，负责游戏内资源的按需下载和边玩边下	审查并发连接数、资源修复逻辑、批量下载、备用CDN源
<b>Maple</b>	GCloud区服导航组件，负责区服选择和路由	审查TreeID一致性、国内版/海外版逻辑、配置一致性
<b>GVoice</b>	GCloud语音通信组件，提供小队语音/范围语音/3D语音/语音转文字等	审查麦克风权限、退房/断线重连、语音消息防截断、翻译计费
<b>SNI</b>	Server Name Indication, TLS扩展，在HTTPS握手时告知服务器目标域名	海外部分CDN节点不支持SNI导致HTTPS连接失败
<b>DMP2.0</b>	GCloud的数据管理平台2.0	主机端需确认DMP2.0协议是否正确适配
<b>TDM</b>	GCloud的数据上报模块	Dolphin需要正确调用TDM接口上报下载数据

术语	解释	在项目中的场景
TDR	Technical Delivery Review, 我建立的组件级风险审查机制	覆盖152+检查项的系统性审查流程
QTS	自动化测试框架	育碧项目未使用QTS, 相关检查项标记为NotInUse

## 四、面试叙事模板（可直接背诵的一段话）

在腾讯IEG基础技术产品部，我的第一个核心项目是**育碧海外旗舰项目的GCloud技术交付**。

**为什么做：**育碧需要接入腾讯GCloud技术矩阵作为底层基础设施，但海外环境的网络复杂性、平台多样性和合规要求，使得每个组件都面临适配风险。

**怎么做：**我作为技术接口人，做了三件事。第一，我设计了一套TDR风险审查体系，把4大核心组件——Dolphin下载更新、Puffer动态资源更新、Maple区服导航、GVoice语音通信——总共152+检查项，按状态（Pending/Failed/Passed）和紧急程度分类管理。第二，我建立了问题分级和闭环机制——代码逻辑问题由内部产研修复，功能验证转交业务QA，配置和商务问题拉对应负责人确认。第三，我推动跨团队协作——每周同步进度看板，Pending超1周的升级处理，跨组件依赖关系我全局把控。

**做出什么：**推动解决了数十项技术卡点，包括Puffer资源修复逻辑Bug、海外CDN SNI兼容问题、GVoice翻译计费确认等，保障了国际项目底层架构按期交付。同时沉淀了TDR模板，供后续国际项目复用。

最后更新：2026-04-09 14:48 | 基于整体整理.docx + TDR-GCloud-ComponentRiskReview\_v3.xlsx