软件开发生产线 CodeArts

代码托管产品介绍

天翼云科技有限公司

目　录

1 产品概述 3

2 产品优势 5

3 应用场景 7

4 功能特性 8

4.1 极致安全韧性 8

4.2 支持 Git 多种作业流 9

4.3 多形式代码检视 9

4.4 代码上库质量门禁 10

4.5 围绕代码研发资产追溯 11

4.6 内嵌仓库规范和模板 11

5 约束与限制 13

1 产品概述

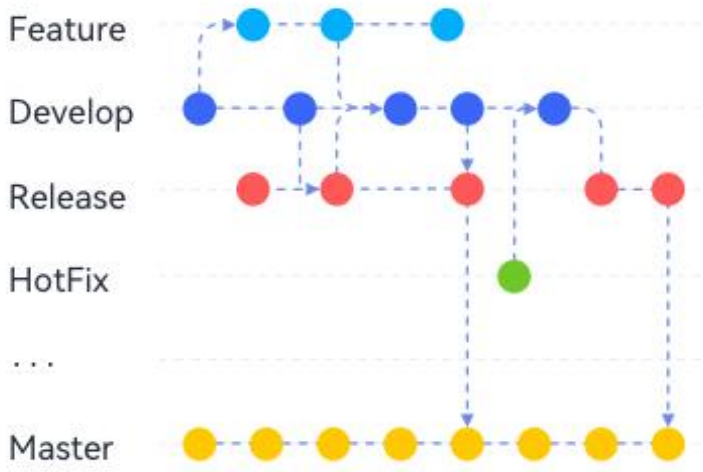
什么是代码托管？

代码托管（CodeArts Repo）是面向软件开发者的基于 **Git** 的在线代码托管服务，是具备安全管控、成员/权限管理、分支保护/合并、在线编辑、统计服务等功能的云端代码仓库，旨在解决软件开发者在跨地域协同、多分支并发、代码版本管理、安全性等方面的问题。

- 在线代码阅读、修改、提交，随时随地开发，不受地域限制。
- 在线分支管理，包含分支新建、切换、合并，实现多分支并行开发，效率高。
- 分支保护，可防止分支被其他人提交或误删。
- IP 白名单地域控制和支持 HTTPS 传输，拦截不合法的代码下载，确保数据传输安全性。
- 支持重置密码，解决用户忘记密码的问题。

代码托管的工作模式

- 代码托管（CodeArts Repo）采用 Git Flow 作为基础工作模式。
- Git-Flow 提供了一组建议，通过严格执行这些建议的规则，帮助中小型研发团队，能够更好的规范自己的开发工作。
  - **并行开发：**各个特性与修复 bug，可以并行。
  - **团队协作：**多人开发过程中，大家都能够理解其他人的当前工作。
  - **灵活调整：**通过 Hotfix 分支，支持各种紧急修复的情况。



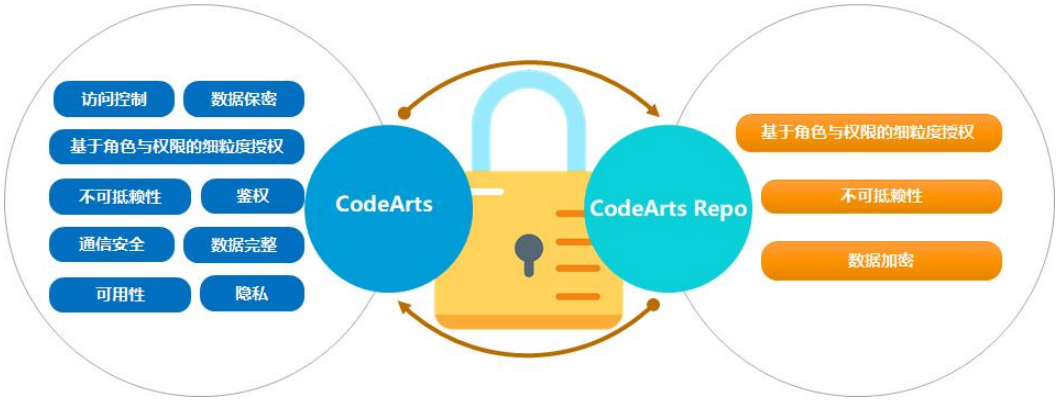
- **master 分支：**最为稳定，功能比较完整，随时可发布的代码。
- **develop 分支：**用于平时开发的主分支，并一直存在，永远是功能最新最全的分支，包含所有要发布到下一个 release 的代码，主要用于合并其他分支。
- **feature 分支：**用于开发新的功能的分支，一旦开发完成，通过测试，合并回 develop 分支进入下一个 release。
- **release 分支：**用于发布准备的专门分支。
- **hotfix 分支：**用于修复线上代码的 bug 。

说明

- 所有开发分支从 develop 分支拉取。
- 所有 hotfix 分支从 master 分支拉取。
- 所有在 master 分支上的提交都必须要有 tag ，方便回滚。
- 只要有合并到 master 分支的操作，都需要和 develop 分支合并下，保证同步。
- master 分支和 develop 分支是主要分支，主要分支每种类型只能有一个，派生分支每个类型可以同时存在多个。

2 产品优势

CodeArts 与 CodeArts Repo 的安全优势



CodeArts 层面的安全特性：

- **访问控制**：公有云采用“租户+用户+用户组+角色”统一模型对权限进行控制。
- **鉴权**：CodeArts 基于公有云统一的认证服务 IAM 来进行认证。用户通过 HTTPS/SSH 访问代码仓库，将使用 SSH Key 或者仓库用户名及密码进行访问鉴权。
- **基于角色与权限的细粒度授权**：不同的角色，在不同的服务中，根据不同的资源，可以有不同的操作权限。还可以做自定义的权限设置。
- **不可抵赖性**：CodeArts 基于公有云 IAM Token 机制，所有操作都必须带有 Token，对所有关键操作进行审计记录。审计日志被持久化，可保留足够长时间，并可进行精确的回溯。
- **数据保密性**：对于敏感信息，CodeArts 会进行加密等进行存储。
- **通讯安全**：CodeArts 对外提供的服务均使用 HTTPS、SSH 等安全协议，保证了通讯的安全性。
- **数据完整性**：CodeArts 的关键信息都保存在内部数据库中，通过事务等各种机制保障了数据的一致性。
- **可用性**：CodeArts 的各个服务都是集群方式，通过保证了服务的高可用性。
- **隐私**：CodeArts 不涉及到租户及用户的隐私。

代码托管（CodeArts Repo）层面的安全特性：

- **基于角色与权限的细粒度授权**：在 CodeArts Repo 层面，提供针对代码访问的，更加细粒度的授权模型。
- **不可抵赖性**：我们提供代码仓库的完整访问日志，供用户审计。
- **数据加密**：用户的代码在 CodeArts Repo 中，是以加密方式存储的。

跨地域协同开发

- 在线代码阅读、修改和提交，随时随地，不受限制。
- 在线分支创建、切换、合并，多分支并行开发，效率高。
- 支持 Git-LFS ，大文件存储无忧；
- 支持在线 Code Review ，团队协作利器。

基于代码的统计分析

- 代码仓库提交信息统计。
- 代码仓库贡献者统计。
- 代码语言统计。

3 应用场景

异地协同开发

- 应用：面向中小企业、孵化中心，协同合作。
- 场景特点：用户群体对开发工作的推进效率，敏捷度要求更高，需要高效的协作管理方式和更低开发成本。面临异地开发协同效率低、代码合并冲突频繁的难题。
- 适用场景：云端代码托管服务，实现协同开发。多分支管理功能和合并请求功能，彻底解决代码合并冲突的难题。

高校教学

- 应用：高校教师与学生，学习与授课。
- 场景特点：目前缺少功能完备的研发工具链，搭建研发工具环境耗费大量时间，环境维护耗费精力，现有的研发工具上手慢，学习成本高，不利于教学。
- 适用场景：代码托管服务提供完整的代码托管服务，以及丰富的代码仓库模板，使学生可以迅速上手。

4 功能特性

极致安全韧性

支持 Git 多种作业流

多形式代码检视

代码上库质量门禁

围绕代码研发资产追溯

内嵌仓库规范和模板

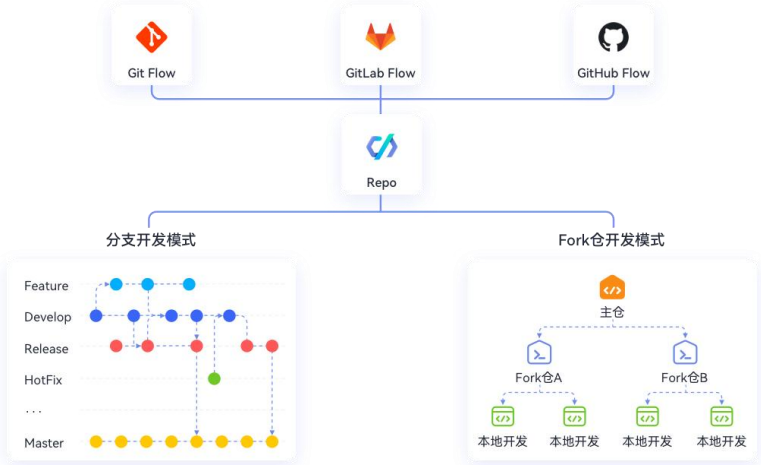
4.1 极致安全韧性

基于云原生架构全栈自研，提供极致韧性和安全的代码托管能力，源于多年实践成果，覆盖云、管、端、车、IT 等超大产品协同开发，10 亿级代码管理，万人团队并发在线协同作业、高并发代码下载，超大存储容量。

4.2 支持 Git 多种作业流

多种开发作业协同方式

提供基于 Git 的多种开发协作模式，既适合中小企业灵活开发模式，也支持中大型企业的复杂开发协作模式。



4.3 多形式代码检视

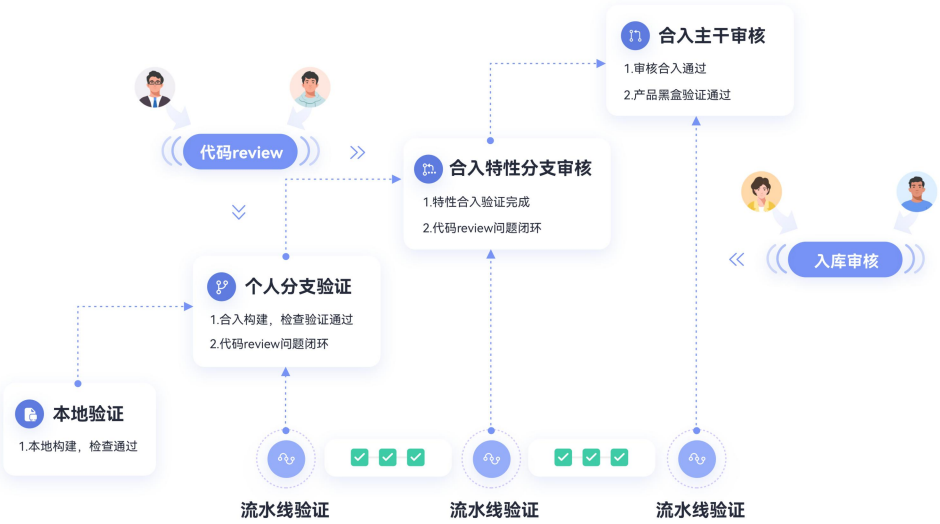
多形式的代码检视活动

支持基于文件的随心检视、合并请求代码检视能力，让团队集中检视或者分散式协同检视，支持检视模板、检视人自动分配、检视任务通知设置，检视意见可跟踪，可闭环，详见《用户指南》的“使用代码托管仓库 > 管理合并请求”章节。

4.4 代码上库质量门禁

多层级、细粒度代码上库质量门禁

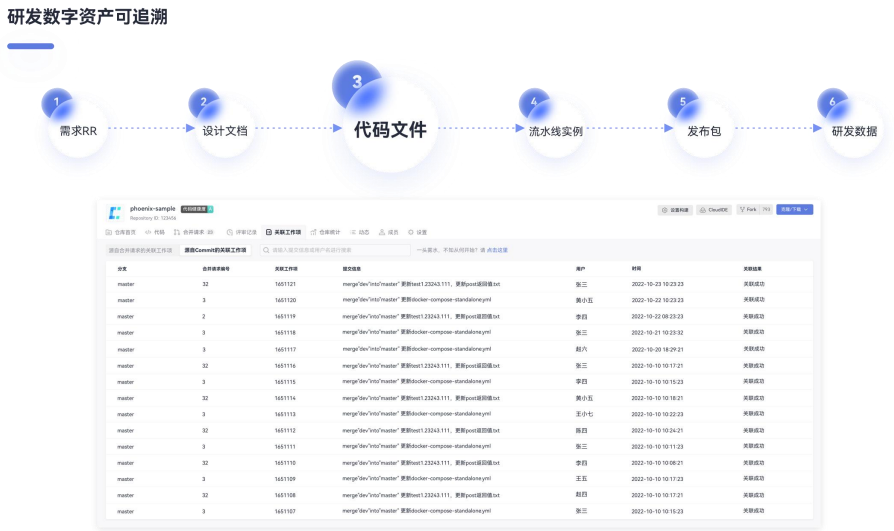
支持人工审核、自动化流水线集成对上库代码进行质量管控，不符合质量指标的代码不允许入库，人工审核支持权责分离原则（SOD），自动化检查，支持分支级管控，详见《用户指南》的“使用代码托管仓库 > 管理合并请求”章节。



4.5 围绕代码研发资产追溯

围绕代码的研发资产追溯

提供从需求、任务、设计、缺陷、代码、版本的记录追溯，掌握每一行代码的来龙去脉，方便网上问题定位和审计，详见《用户指南》的“配置代码托管仓库 > 服务集成 >E2E 设置”章节。



4.6 内嵌仓库规范和模板

丰富的仓库模板，标准化的团队开发活动

提供仓库模板，代码检视模板，合并请求模板，可配置必填和选填字段，确保团队开发行为统一，更加方便的基于研发数据做效能分析和改进，详见《用户指南》的“配置代码托管仓库 > 模板管理”章节。

代码提交信息
代码合入方式
门禁规则
分支、tag命名
代码检视意见格式
合并请求格式
代码推送规则
仓库权限
统一开发架构
统一开发语言
统一目录结构
统一仓库结构
（仓库模版）

5 约束与限制

本节介绍了代码托管中的限制，如下表所示。

表 5-1 使用限制说明

指标类型	指标项	限制说明
浏览器	类型	目前适配的主流浏览器类型包括： • Chrome • IE10 以上 • Microsoft Edge • Firefox • Safari 推荐使用 Chrome、Microsoft Edge 浏览器，效果会更好。
分辨率	分辨率大小	推荐使用 1920*1080 及以上。
单个仓库规格	单文件上传大小限制（页面）	<=10MB。
	单文件推送大小限制（本地）	<=200MB。
	在线修改代码，单次保存行数限制	<=5000 行。
	仓库容量（超出容量限制会导致仓库部分功能无法使用，如代码无法上传）	<=2GB。

说明

当您处于以下情况时，会受到的影响：

- **仓库容量超出限制**：超出容量限制会导致仓库部分功能无法使用，如新建/编辑文件、新建目录、新建子模块、新建分支/tag、上传文件、解决代码冲突、合入合并请求、Git 客户端推送等。仓库容量正常后，系统启动定时任务恢复仓库状态。
- **关闭代码托管服务：**您不可进入仓库，界面并提示需开通服务，开启代码托管服务后，恢复仓库状态。关闭代码托管服务超过 30 天，系统自动删除仓库，不可恢复。