云平台新架构技术方案

前言

随着公司的发展壮大，越来越多的客户使用我们的产品。产品现有的技术架构渐渐暴露了一些短板和不足，为了不断满足客户需求的增长和系统的稳定性、高性能、可伸缩、维护性、先进性，体现公司的技术实力的特点，决定将云平台从新规划和实现，技术更新，支持更高级别数据存储，提供数据挖掘、数据分析和数据展现工具，保障访问安全高速，提供完整的故障预警和处理机制。

第1章　概述

1.1　设计概述

通常，使用分布式架构设计能解决企业产品能够安全高速运行，支持更高级别的数据存储，可利用业界成熟的工具进行数据挖掘和分析以及故障预警和处理。利用商业云，我们可以做到可缩放、可复原且高度可用。我们的云平台将使用阿里云托管，围绕云托管设计，充分利用云服务，确保数据不丢失和可分析，具有故障预警和处理以及快速恢复能力。

1.2　上云概述

云计算中的虚拟化技术带来“弹性、灵活、安全、低成本”的特性，使“上云是常态，不上云是例外”成为共识。阿里云与其他主流商业云一样，具有国内和全球的云服务，利用这些云服务，即使是一个 IT 小

团队也能做到一个 IT 大团队所能做到的事情，如庞大的高速的安全的网络、大数据存储和分析、故障预警和处理。

1.3 分布式概述

分布式计算是计算机科学中一个研究方向，它研究如何把一个需要非常巨大的计算能力才能解决的问题分成许多小的部分，然后把这些部分分配给多个计算机进行处理，最后把这些计算结果综合起来得到最终的结果。分布式网络存储技术是将数据分散地存储于多台独立的机器设备上。

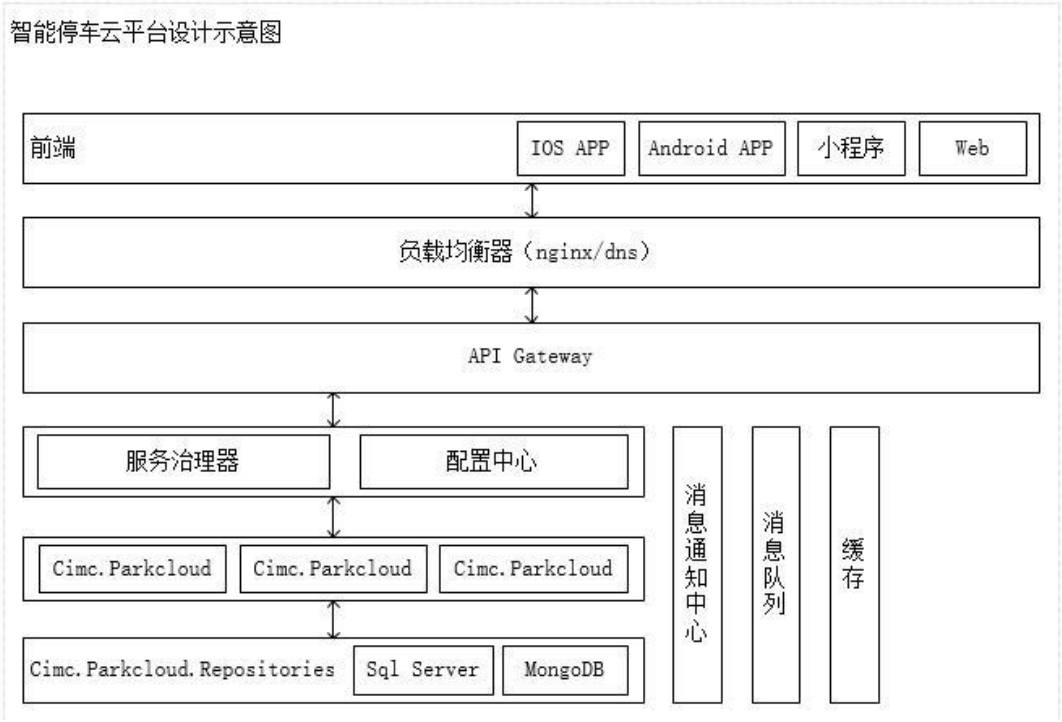
我们的云平台将使用阿里云完全托管，选择适合的云服务进行分布式部署和治理。

1.4 云平台概述

云平台使用微软最新.net core 框架开发设计，以分布式部署方式作为技术中台，以开放平台方式对接第三方业务，适应不断增长的业务和定制化需求。以编程语言无关性的 Restful API 承载各种前端，包括 web 前端，android、ios、小程序等，以及任何第三方合作方。

第2章 系统详细设计

2.1 云平台架构整体设计示意图



2.2　传输格式与约定

云平台以 Restful API 对外提供接口，以 json 作为数据传输方式。

2.3 核心逻辑设计

核心逻辑以标准库提供，适应多目标。核心逻辑包含实体、枚举、配置、扩展、接口处理逻辑，关系型数据库和 NoSQL 数据库支持等等。支持多种数据库类型的提供程序，使用统一接口实现，按需切换数据库类型。使用 Entity Framework Core 操作关系型数据库类型，如 SQL Server、MySQL 等。考虑到云平台全球部署时需正版验证，购买 SQL Server 费用较高，以及综合考虑业务场景，我们将选择 MongoDB 作为云平台的首选存储方式。云数据库具有弹性伸缩、高可用、容灾、备份回滚、性能优化等特点，这些传统需要一个 IT 大团队才做的事情，现在上云后小团队也能做到了。

2.3 缓存提供程序

分布式缓存是由多个应用服务器共享的缓存，通常作为外部服务在访问它的应用服务器上维护。 分布式缓存可以提高 ASP.NET Core 应用程序的性能和可伸缩性，尤其是在应用程序由云服务或服务器场托管时。云平台使用 redis 作为分布式缓存。通常，Redis 缓存提供比 SQL Server 缓存更高的吞吐量和更低的延迟。 但是，通常需要进行基准测试来确定缓存策略的性能特征。

2.4 日志提供程序

通常，人们喜欢使用 ElasticSearch 作为海量的日志存储，但是 ElasticSearch 安装部署麻烦，且消耗内存很大，实践证明，数据量大时 ElasticSearch 查询特别缓慢，影响工作效率。我们的云平台可能将选择 mongodb 作为日志数据的存储作为暂定方案。Microsoft.Extensions.Logging 扩展，支持适用于各种内置和第三方日志记录提供程序的日志记录 API。

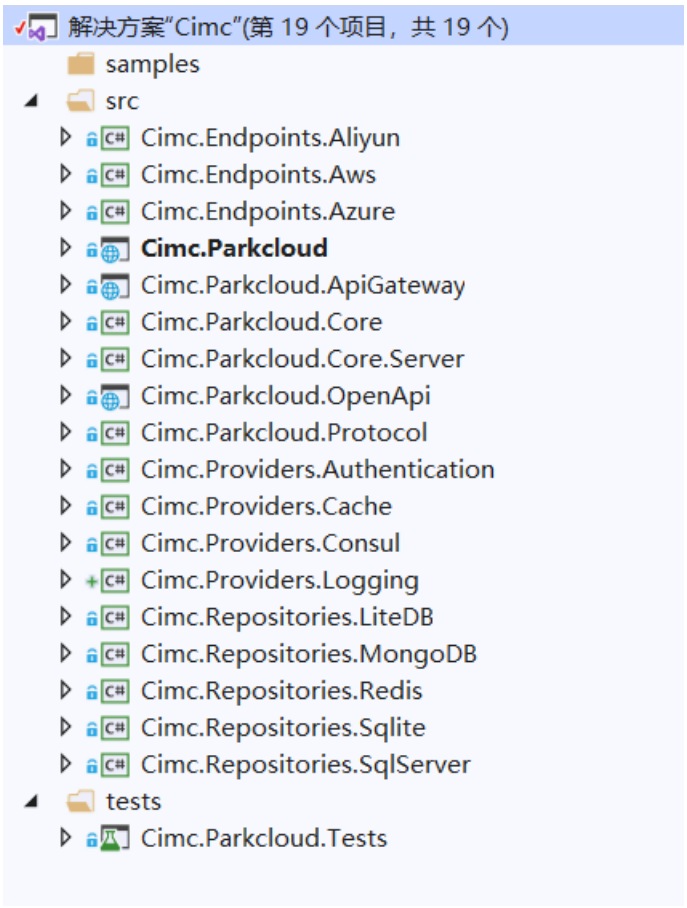
2.4 负载均衡

负载均衡是将访问流量根据转发策略分发到后端多台云服务器的流量分发控制服务。负载均衡扩展了应用的服务能力，增强了应用的可用性。使用云服务提供的负载均衡将是我们云平台的首选，因为云服务具有很大优势，如高可用、可扩展、低成本 、安全 、高并发

等等。

2.5 解决方案结构

解决方案模板生成一个多项目解决方案。对于名为 Cimc 的解决方案，将创建以下文件夹结构:



src 中的项目是按一定职能命名的，看其名即可猜到它的实现以及用途。tests 中的项目是单元测试，一般用在需要测试其功能和验证结果时。

2.5.1 核心逻辑项目（Cimc.Parkcloud.Core）

核心层不应依赖于数据访问和其他基础结构层，因此这些依赖关系可

以反转。这是通过在 Core 中添加由 Core 外部各层实现的接口或抽象来实现的。例如，如果要实现存储库模式，可以通过在 Core 中添加接口并在基础射设施层中添加实现来实现。

2.5.2 核心逻辑服务器项目（Cimc.Parkcloud.Core.Server）

依赖于核心逻辑项目，给终结点提供接口服务。可在此项目设置路由或访问特性，提供外部 API 等等。

2.5.2 存储层项目

Cimc.Repositories.SqlServer 、 Cimc.Repositories.MongoDB 、 Cimc.Repositories.Sqlite 这些以 Cimc.Repositories 开头的项目说明是存储层，这些是 Core 中的接口实现，支持各个流行数据库，如对关系型数据库 Sql Server、MySQL，PostgreSql 的支持；NoSQL 数据库：MongoDB 的支持。

2.5.3 商业云实现项目

Cimc.Endpoints.Aliyun（阿里云）、Cimc.Endpoints.Aws（亚马逊云）、Cimc.Endpoints.Azure（微软云），这些以 Cimc.Endpoints 开头的项目是上云后做相应的实现以及对接其相应的 API。

2.5.4 日志项目

日志记录可提供便捷的系统跟踪和调试，分布式大规模部署的机

器和数据都成倍增长，如何通过日志记录排查问题，发现问题能快速定位并解决，无疑是给我们带来了巨大的挑战。Cimc.Providers.Logging 项目是云平台的日志记录实现。使用 Microsoft.Extensions.Logging 扩展实现，以分布式数据存储和 web 查看。

2.5.5 分布式缓存项目

缓存对于提升性能和吞吐量极其重要，特别是对于分布式集群大规模部署来说，分布式缓存带来了一些技术的挑战，数据量也会成倍的增长。Cimc.Providers.Cache 是云平台分布式缓存实现。，