城轨·企铁

DOI：10.13879/j.issn.1000-7458.2022-06.21362

基于云平台的ATS系统微服务架构方案研究

宋　欣，张德明，徐　伟，王　超

摘　要：在当前智慧地铁的升级转型过程中，城市轨道交通ATS系统面临着稳定性差、运维方式繁琐和故障应急处理效率低等问题。为此，提出一种基于云平台的ATS系统微服务架构方案；研究了ATS系统在云平台上的微服务总体架构、ATS应用的微服务划分和设计、容器部署方式及通信交互模式，可实现系统松耦合和易扩展，达到节约计算资源、提高系统稳定性和运营效率的目的。

关键词：列车自动监控系统；云平台；微服务；容器；通信交互
中图分类号：U231+.7　　**文献标识码：**A

Research on Micro-service Architecture Scheme of ATS System Based on Cloud Platform

Song Xin, Zhang Deming, Xu Wei, Wang Chao

Abstract: In the current upgrading and transformation process of smart subway, the ATS system of urban rail transit is faced with such problems as poor stability, cumbersome operation and maintenance methods and low efficiency of fault emergency treatment. Therefore, a micro-service architecture scheme of the ATS system based on the cloud platform is proposed. The overall micro-service architecture of the ATS over the cloud platform, the micro-service division and design of ATS applications, container deployment mode and communication interaction mode are studied, which can facilitate loose coupling and easy expansion of the system, save computing resources, improve system stability and operation efficiency.

Key words: ATS; Cloud platform; Micro-service; Container; Communication interaction

当前，城市轨道交通行业云平台化已经进入了大规模的应用阶段，云平台有着低成本、高可用、高扩展等优势，能够更好地助力城市轨道交通行业实现服务业务创新[1-2]。

列车自动监控系统（Automatic Train Supervision，ATS）是城市轨道交通行车指挥和监控的核心系

宋　欣：北京华铁信息技术有限公司　工程师　100081　北京
张德明：中国铁道科学研究院集团有限公司通信信号研究所　副研究员　100081　北京
徐　伟：中国铁道科学研究院集团有限公司通信信号研究所　助理研究员　100081　北京
王　超：中国铁道科学研究院集团有限公司通信信号研究所　助理研究员　100081　北京
基金项目：中国铁道科学研究院集团有限公司科研课题（2019YJ072）
收稿日期：2021-09-22

统之一，对管辖区段内的列车进行统一的行车指挥和监控管理，是实现线网跨线运营统一调度的重要技术装备。ATS系统与云平台技术的结合，对于城市轨道交通行业数字化建设、智慧地铁转型升级具有重要的意义。

1　现状分析

传统ATS系统根据业务逻辑分别在中心和车站部署了实现不同功能的服务器和工作站，其架构见图1。该体系架构在当前运输调度指挥生产中遇到了一些问题和挑战，主要反映在以下3个方面。

1）随着城市轨道交通行业的发展，面对智慧地铁的升级转型，ATS系统新的功能需求和接口众多，功能变更升级涉及面广、影响大，应用系统的开发技术栈和架构都急需做出改变，但是这样的改变往往牵一发而动全身，极大地影响ATS系统的稳定性，风险性很高。

2）由于ATS系统的不断升级，单个应用程序所依赖的基础配置也越来越多。当程序发生故障重新启动时，需要从数据库或者配置文件中读取大量数据，程序内部初始化也需要一定的响应时间。另外，在对系统的软硬件问题进行应急处置时，通常需要人工介入，进行系统软硬件资源的再分配，例如导入备份、清理内存、增加硬盘，以及更换设备等，对于轨道交通这样实时性要求很高的行业需要做出相应的优化。

3）传统ATS系统应用程序在正式发布过程中，往往会出现开发环境、测试环境，以及现场环境不能完全一致的问题，导致开发或测试通过的程序在发布到现场后却不能正常使用，大大降低了工作效率，造成了工期的延误。

2　设计方案

为了推进智慧城轨的建设，提高ATS系统的稳定性、运维管理能力和故障应急处置能力，可通过将ATS系统纳入云平台部署来实现。云平台能够为ATS系统提供一个良好的生产环境和管理环境，但是要充分利用云平台高可用和高扩展的优势，必须对ATS系统的架构进行重构，相应的部署方式和通信方式也需要做出改变。

2.1　总体架构

基于云平台的ATS系统的微服务架构可划分为应用层、网关层、业务逻辑层、数据访问层和存储层，其架构示意见图2。

网关层：API网关是一个服务器，是系统的唯一入口。它承载着ATS系统所有服务的组合路由转换等工作，除此之外，系统安全、系统保护等工作也需通过API网关来实现。例如，API网关会对外部应用请求进行权限判断，防止非法入侵；也会对外部应用的并发请求进行限流，通常是限速或限

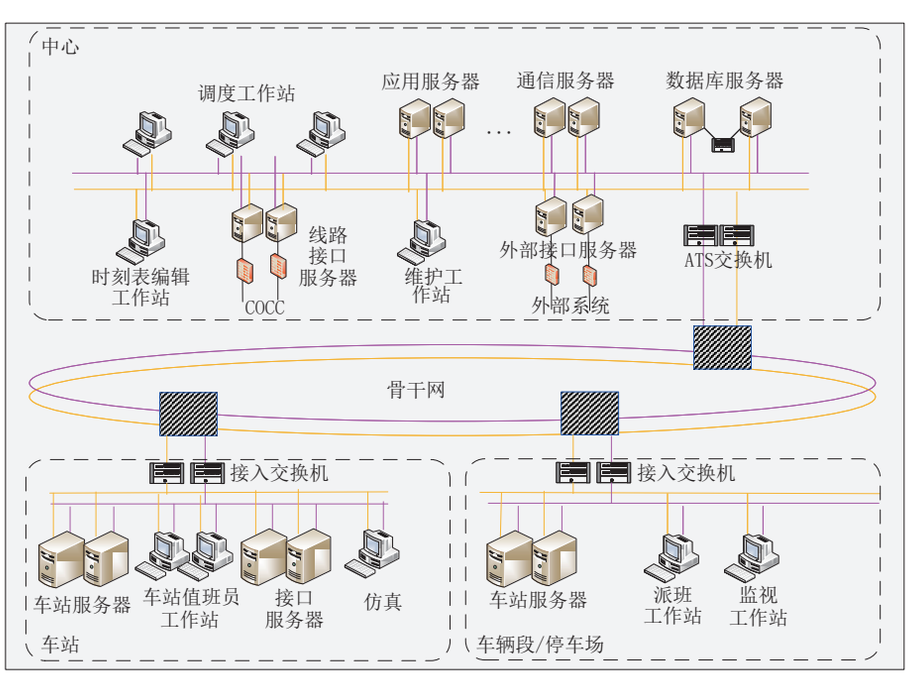

图1　传统ATS系统架构示意

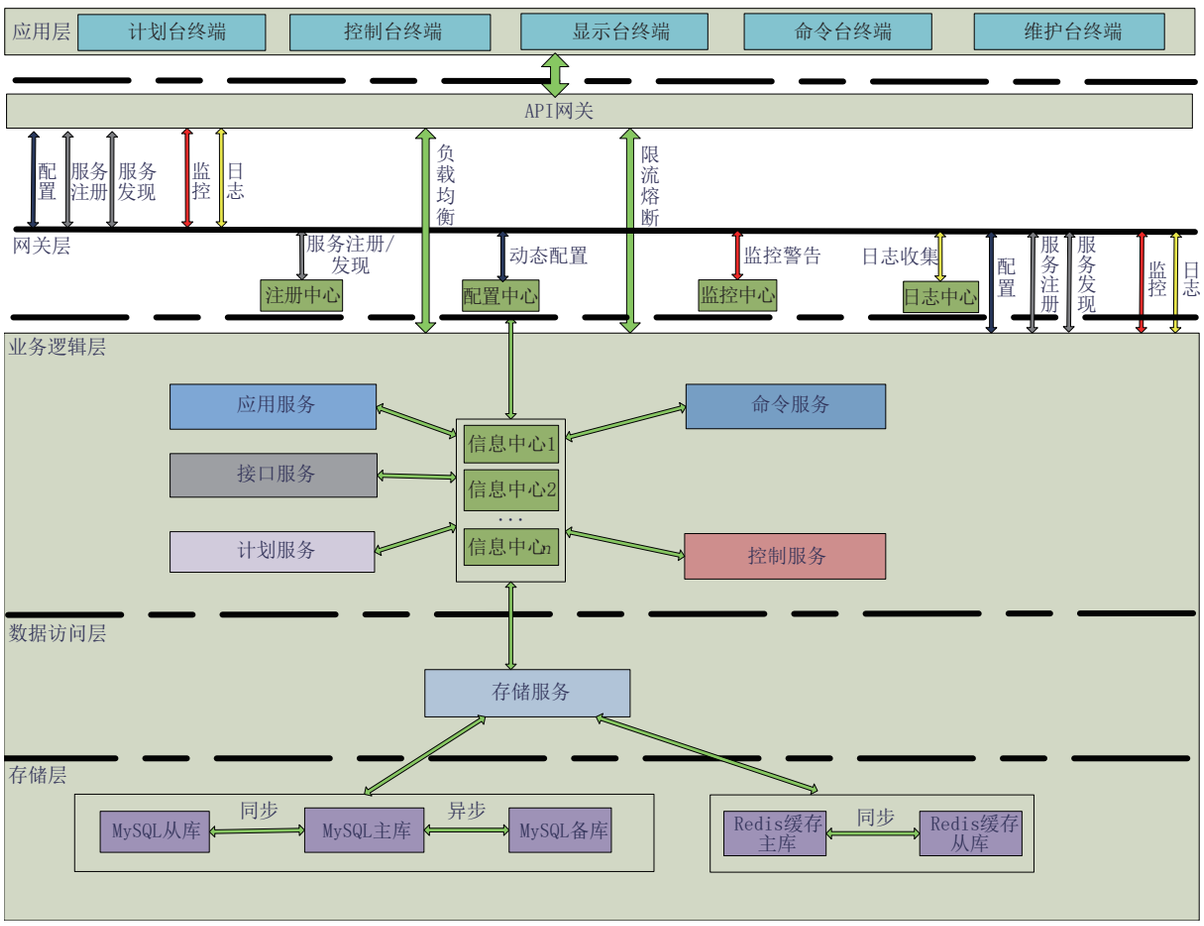

图2 ATS系统微服务架构示意

制一个时间窗口内请求的数量。API网关将接入逻辑和业务逻辑分开，只处理接入逻辑等非业务功能，实现对原有功能的解耦[3]。

业务逻辑层：与传统ATS单体应用程序的架构模式不同，ATS系统的主业务都以微服务的形式部署在业务逻辑层的容器中，实现ATS业务逻辑处理功能。

数据访问层：主要对ATS数据进行存储前的格式转换，分别转换成规范数据后存储到MySQL数据库或者Redis缓存中。

存储层：MySQL数据库设置主库、从库和备库，Redis缓存数据库设置主库和从库，以保证数据的完整性和实时性，满足ATS系统对数据有效性和容灾的需求。

传统ATS系统分为中心和车站两层业务，本方案采用系统整体上云的解决方案，由中心级云平台和站段级云节点构成[4]，逻辑架构示意见图3。

2.2 微服务划分

微服务的划分粒度越大，系统发生故障时产生的影响也越大；而微服务划分粒度越小，拆分的模块越多，系统结构越复杂，开发成本和风险也越高。微服务高内聚、低耦合的特点对开发设计提出了更高的要求，划分不当反而会影响系统的开发进度，增加开发成本，成为阻碍业务发展的因素。因此，微服务的划分必须基于系统业务本身，并权衡可用性、可靠性和成本等多方面因素，来决定划分微服务粒度的大小。本方案中ATS系统微服务划分见图4。根据ATS系统功能拆分全线业务，实现数用分离。微服务主要划分为通用服务和专用服务两大类。

2.2.1 通用服务

通用服务主要包括以下内容。

1）注册服务。主要包括服务注册和服务发现。其中，服务注册是将为ATS系统提供某个服务的模块信息（例如IP地址和端口）注册到注册中心公共的组件上，由注册中心统一管理；服务发现是ATS系统新注册的服务模块能够被其他调用服务及时发现。新增和删减的服务都能够被注册中心自

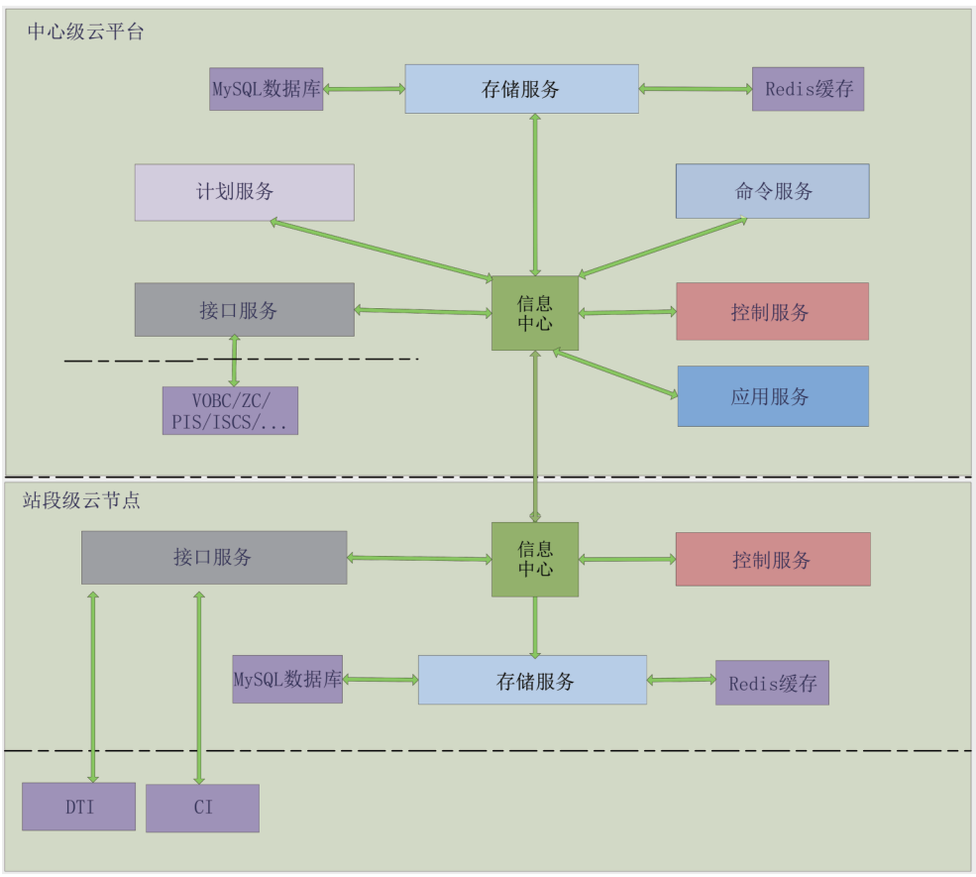

图3　ATS系统微服务业务逻辑架构示意

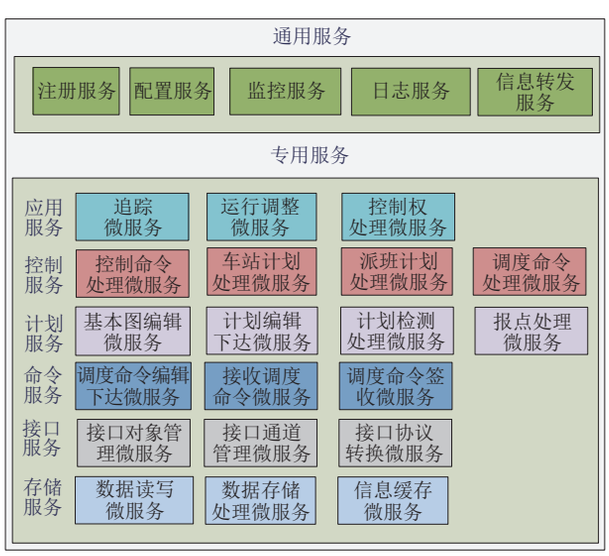

图4　ATS系统微服务划分示意

动发现和管理[5]。

2）配置服务。ATS系统的各个应用都有大量的配置文件，通过配置服务可以将配置文件从各应用中剥离出来，对配置进行统一管理，ATS系统应用自身无需管理配置。用户可在配置中心更新配置信息，及时得到配置更新通知，并从配置中心获取配置。

3）监控服务。可以实时了解和分析ATS系统各个微服务的运行状况和性能数据，同时实时收集所有微服务实例的运行性能数据，进行分析和统计。

4）日志服务。负责收集ATS系统各个微服务的运行日志。当微服务系统调用过程中出现问题时，定位故障日志比较困难，日志中心将所有节点上的日志统一收集，进行管理和访问，可按需求提供给大数据平台做数据筛查和数据分析，极大地提高了效率。

5）信息转发服务。ATS系统各个微服务通过信息转发微服务（以下称为“信息中心”）进行交互。当应用集群足够大，单信息中心很难满足通信需求时，信息中心和其他微服务都可启动多个容器实例，通过负载均衡来解决问题[6]。

2.2.2　专用服务

专用服务从逻辑功能上主要分为应用服务、控制服务、计划服务、命令服务、接口服务，以及存

储服务六大类。本方案中将这六大类服务拆分为21类专用微服务，更加细化，粒度更小。相对于传统ATS系统的应用程序，有的专用微服务，如追踪微服务，只是应用服务器程序中一个单独的功能模块，有着独立的算法和处理逻辑，所以单独划分为1个微服务；再如接口对象管理微服务，是中心接口服务器、外部接口服务器和TCC接口服务器等多个程序的共有功能模块，扩展频率更高，因此也单独划分为一个微服务，更易于开发。传统的ATS系统应用程序在物理概念上被多个专用微服务所分解取代，能够解决传统的ATS系统单体式应用程序维护升级困难、设计开发成本高、软件耦合度高等问题[7]，其优势如下。

1）整个ATS系统松耦合，能够更加高效地利用计算资源。开发人员只需要为额外的组件部署计算资源，开发和维护单个微服务相对简单，这样整个ATS应用业务都可以被维持在一个可控的状态。传统ATS系统单个应用程序需要修改时，需要重新部署整个应用程序，而拆分成微服务后只需要对相关的微服务进行修改，再重新部署该微服务即可，在系统升级维护过程中更加高效，降低了维护成本和风险。

2）可以结合ATS自身业务的特点，合理选用技术栈。技术栈不再受限制，更易扩展。例如，计划服务和命令服务可以调用存储服务，使用关系型数据库MySQL对数据进行长期存储；应用服务、控制服务、接口服务可以使用非关系型数据库Redis进行数据缓存。根据功能需求，不同的微服务可以利用差异化的开发语言组织开发；还可根据实际业务需求，实现细粒度的扩展。例如，ATS系统中的信息转发微服务需要提高并发处理能力时，可以通过增加内存、增加实例节点或者升级CPU来解决。

2.3 微服务部署

ATS系统划分的每个微服务均独立部署在容器中，可独立编译。容器技术是一种以应用程序为中心的虚拟化技术，直接使用物理资源，所以几乎没有性能损耗，比起虚拟机更加轻量和易于快速部署，能大幅度降低云平台的资源使用总量，并提高应用分发的部署效率[8]，为自动化测试和持续集成/部署、微服务提供支持[9]。K8S是目前主流的容器集群管理平台，具备完整的资源调度、业务部署、状态监控、服务配置、安全管理等功能。云平台可根据系统需求和资源方案提供计算、存储、网络、安全资源，以及K8S组件，完成镜像及容器部署[10]。

本方案中采用Docker镜像将ATS应用程序本身，以及ATS应用程序依赖的所有软件，包括操作系统环境全部打包，由K8S统一进行容器编排，解决了微服务运行环境不能在本地环境、测试环境以及现场环境保持一致的难题。

通过微服务部署，可以打通整个开发、测试和发布的流程，即将在本地环境中运行测试通过的代码，以及依赖的软件和操作系统本身打包成一个镜像，然后自动部署在测试环境中进行测试，测试通过后可以自动发布到现场环境中。另外，在云平台上只需通过一条指令就可以为一个应用创建多个容器实例，还可以自动实现资源扩展、应用故障后自动重启等功能，相当于一个24 h不间断服务的运维工程师[11]。同时，单个微服务代码量小，配置文件也相应拆分运行在单独的容器中，应用程序更加轻量化，可以实现秒级启动。

2.4 通信交互

传统ATS系统多采用客户端/服务器模式，信息实时交互频繁，通过TCP/UDP协议进行通信。而在当前主流的云平台微服务架构中，系统多采用浏览器/服务器模式，通过HTTP协议进行前/后台的通信。对于ATS系统来说，由于HTTP协议短连接状态不稳定，因此不能满足系统通信需求。

首先，将传统ATS系统的前/后台模式重构为浏览器/服务器模式，前台采用WEB网页形式；其次，前台和后台之间增加API网关，通过API网关进行前台和后台的信息交互；最后，后台容器中的微服务之间可相互调用通信，处理系统内部逻辑业务。在前/后台通信和后台内部通信的设计方案中，均采用基于TCP协议的长连接，这样能够建立起稳定的通信通道，使得信息能够持续实时交互，既保留了ATS系统既有的TCP通信方式，又减少了开发难度。

在系统前/后台通信交互过程中，采用了WebSocket的通信方式。WebSocket是独立的、创建在TCP上的协议，浏览器和后台服务只需要完成一次握手，二者之间就可以直接创建持久性的连接，并进行双向数据传输。由于协议是全双工的，所以后台服务可以随时主动给客户端下发数据。相对于HTTP需要等待客户端发起请求，服务端才

能响应来说，响应延迟明显更短。

在系统后台内部通信中，采用了远程过程调用（Remote Procedure Call，RPC）通信方式，在内网中的通信速度快、效率高。基于TCP协议的RPC调用，由服务的调用方与服务的提供方建立Socket连接，屏蔽了跨进程调用函数（服务）的各类复杂细节，使调用方调用远端函数就像调用本体函数一样，使服务提供方实现服务就像实现一个本地函数一样。

由上述分析可知，本系统采用了“对外WebSocket，对内RPC”的通信解决方案，因此需要在API网关注册和管理服务中增加WebSocket-RPC的协议转换，对内向信息中心提供RPC接口调用，对外向客户端提供WebSocket接口调用。客户端在调用服务端WebSocket接口时，传送IP地址、端口号和设备类型等信息，API网关负责对客户端进行卡控，并将有效信息通过RPC调用接口发送给内部信息中心。

信息中心与其他专用微服务的通信交互逻辑见图5。信息中心是本方案中负责内、外部信息交互的重要组成部分，其主要作用如下。

1）对所有交互的信息类型做卡控，非法的信息不做转发处理，可以保证信息的可靠性和系统的安全性[12]。

2）对某些信息类型做屏蔽，暂时对某些微服务不转发该信息类型，可以提高微服务的信息处理效率和灵活性。

3）复制信息转发。ATS系统的信息交互比较繁杂，而且常会有新扩展的设备或者新开发的功能，很多信息往往需要增加或减少微服务之间的调用，这就需要信息中心通过灵活配置来控制信息转发。信息中心启动后，会从配置中心调用配置，内外交互的信息类型将配置成一个转发配置组，例如：Infol=［65 535，255，255，0x83，0，17，255，0，18，255］。其中，65 535表示站码广播，255表示设备类型广播，0x83表示信息类型，0表示中心，17、18表示设备类型；该配置表示为之前所有发送到中心设备类型17的0x83信息要新生成一份转发到中心设备类型为18的设备中。当信息中心收到0x83信息后，会自动重新生成一包新数据发往中心类型为18的设备，这样对于发送方的微服务只需要发送一份信息即可，减轻了发送方微服务的负担。

以ATS系统车站现地终端下达控制命令为例，

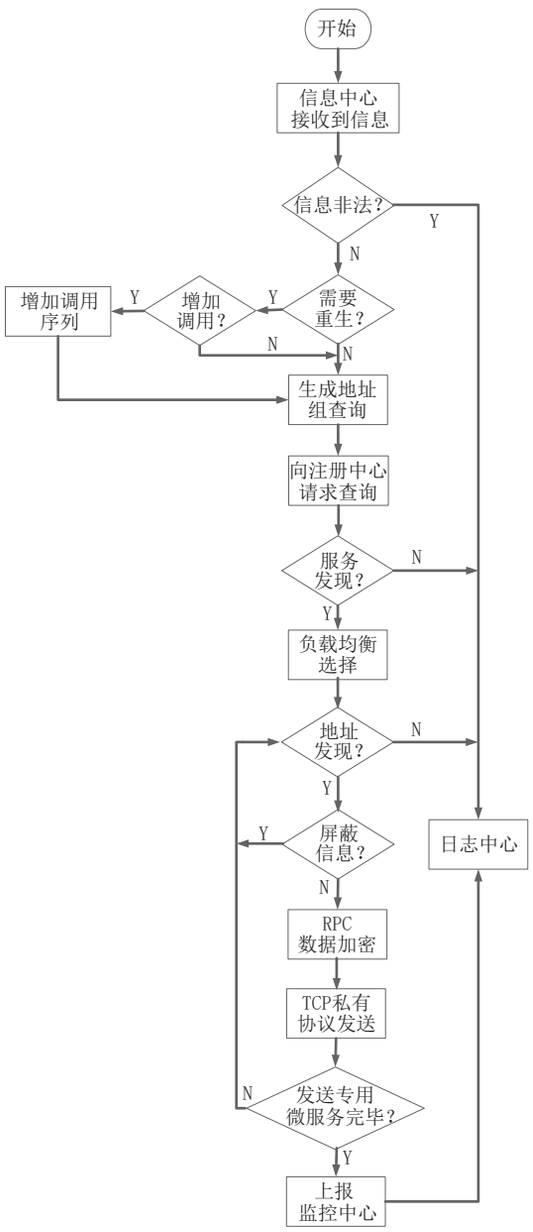

图5　信息中心与其他专用微服务的通信交互逻辑

描述本方案中的信息流见图6。ATS系统现地终端下达控制命令到API网关，API网关进行WebSocket-RPC协议转换后，通过RPC调用信息中心微服务；信息中心根据控制命令的信息类型和目的地（联锁），从注册中心获取对应的微服务地址组，再按地址组序列将地址组信息和控制命令信息转发给控制命令微服务；控制命令微服务收到后对命令进行逻辑处理，处理完毕后按地址组序列依次发送给其他微服务；最后，命令信息通过接口通道管理微服务下达给联锁系统；联锁系统收到命令后，返回命令回执/表示信息，并将其显示在现地终端上。

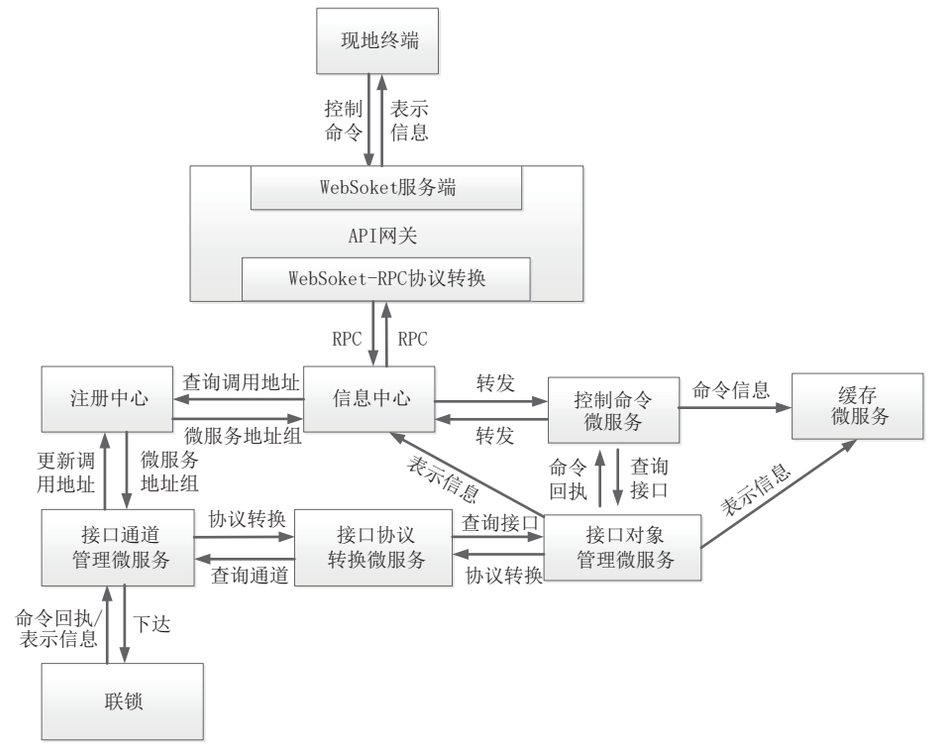

图6 ATS系统微服务信息流示意

3 结论

1）在城市轨道交通行业中，ATS系统作为行车指挥的核心系统之一，功能需求多，业务复杂，接口众多，应用粒度细。基于云平台的ATS系统采用松耦合的微服务架构体系，既满足了与既有系统的对接需求，节约了硬件资源，又提高了其扩展性和运营效率。

2）由于微服务架构具有可迭代特点，ATS系统无论在前期的开发还是后期需求的变更升级，都可以将实施影响控制在较小的范围内，这种对系统升级逐步验证的方式，既降低了投入的风险，也保证了系统的稳定性，对于城市轨道交通数字化建设有着良好的应用前景，为智慧地铁转型升级奠定了坚实的基础。

参 考 文 献

[1] 黄龙，张博．城市轨道交通云平台建设方案分析[J]．铁道通信信号，2020，56(9)：76-80.

[2] 刘海建，刘占英，樊艳，等．城市轨道交通线网云平台的研究与应用[J]．铁道通信信号，2020，56(4)：78-81，84.

[3] 孙杰，山金孝，张亮，等．企业私有云建设指南[M]．北京：机械工业出版社，2019.

[4] 中国城市轨道交通协会．T/CAMET 11002—2020 城市轨道交通云平台构建技术规范[S]．2020.

[5] 向逸尘，林浩宇，徐源．微服务架构在智慧城市平台建设中的应用初探[J]．计算机技术与自动化，2019，38(3)：136-140.

[6] 黄永华．应用Socket的微服务之间的通讯[J]．福建电脑，2020，36(2)：68-71.

[7] 唐稳，刘艳辉．轻量级C++微服务框架的设计[J]．计算机工程与设计，2019，40(8)：2396-2400.

[8] 袁忠良．容器云计算平台关键技术研究[D]．南京：南京大学，2017.

[9] 李昱见，许利沙，秦义展．智慧轨道交通中的容器云[J]．信息技术与标准化，2019(5)：72-75.

[10] 马博超．容器技术在城轨云平台中的应用研究[J]．电气化轨道，2020(S1)：230-232.

[11] 佟业新，曲新奎．民航微服务架构设计[J]．中国科技信息，2019(21)：27-29.

[12] 中国城市轨道交通协会．T/CAMET 11005—2020 城市轨道交通云平台网络安全技术规范[S]．2020.

（责任编辑：王 菲）