



团 体 标 准

T/CI 823—2024

# 工业软件云架构通用技术指南

General technical guide for industrial software cloud architecture

2024-12-24 发布

2024-12-24 实施

中国国际科技促进会 发 布  
中国标准出版社 出 版



## 目 次

前言 .....	III
1 范围 .....	1
2 规范性引用文件 .....	1
3 术语和定义 .....	1
4 缩略语 .....	2
5 工业软件云的架构设计 .....	2
5.1 概述 .....	2
5.2 工业软件云的架构设计原则 .....	2
5.3 工业软件云的通用架构 .....	3
5.4 信息安全防护体系 .....	5
6 基于微服务架构的工业应用开发与改造 .....	5
6.1 概述 .....	5
6.2 基于微服务架构的工业应用系统开发 .....	5
6.3 已有工业业务系统的微服务化改造 .....	5
附录 A（资料性）云环境下工业应用的持续集成与交付 .....	7
参考文献 .....	10



## 前 言

本文件按照 GB/T 1.1—2020《标准化工作导则 第 1 部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中核武汉核电运行技术股份有限公司提出。

本文件由中国国际科技促进会归口。

本文件起草单位：中核武汉核电运行技术股份有限公司、云南电网有限责任公司电力科学研究院、福建星瑞格软件有限公司、广州中望龙腾软件股份有限公司、浙江大学、上海积鼎信息科技有限公司、济南超级计算技术研究院、湖南麒麟信安科技股份有限公司、合肥九韶智能科技有限公司、云翼(嘉兴)软件科技有限公司、菏泽市智慧水务有限公司、北京适创科技有限公司、恒典信息科技(苏州)有限公司、中兴通讯股份有限公司、东风设备制造有限公司、武汉职业技术学院、兰州城市学院、哈尔滨工程大学。

本文件主要起草人：张大志、杨莉、胡岩洁、张道山、罗坤、沈卓、傅彦国、徐力敏、郭猛、仲红俊、杨鹏举、杨周旺、郑裕峰、廖祐明、张辉、张子珂、朱晖、戚晨、吴勉、吴艳霞、朱红、王振禄、吴艳霞。



# 工业软件云架构通用技术指南

## 1 范围

本文件提供了工业软件云架构的架构设计和基于微服务架构的工业应用开发与改造的建议。

本文件适用于流程工业、先进制造业等领域的基于云计算的工业应用架构的设计、开发、安装部署、持续集成与交付。

## 2 规范性引用文件

本文件没有规范性引用文件。

## 3 术语和定义

下列术语和定义适用于本文件。

### 3.1

**云架构 cloud computing architecture**

云计算的组织方式,包括前端和后端的技术组件、网络、软件和硬件资源。

### 3.2

**云原生 cloud native**

一组用于构建和运行在云环境中应用的技术,包括容器、服务网格、微服务、不可变基础设施和声明式 API。

### 3.3

**微服务 microservices**

一种使用一套小服务来开发单个应用的方式途径,每个服务运行在自己的进程中,并使用轻量级机制通信,通常是 HTTP API,这些服务基于业务能力构建,并能够通过自动化部署机制来独立部署,这些服务使用不同的编程语言实现,不同数据存储技术,并保持最低限度的集中式管理。

### 3.4

**数据管理 data management**

组织内部的数据收集、存储、保护、处理和分析的过程和策略。

### 3.5

**持续交付 continuous delivery**

以小的增量块生产与交付完整软件的方法。

### 3.6

**持续集成 continuous integration**

一种软件开发实践,团队中的成员频繁提交代码到代码仓库,且每次提交都能通过自动化测试进行验证,从而尽可能早地在开发周期中正确发现软件问题并确保整个平台的所有组成部分能够相互正确协作。

### 3.7

#### 发布 **launch**

单一代码包发布代码到测试、预生产、生产环境。

### 3.8

#### 部署 **deployment**

推送版本到指定环境的行为。

### 3.9

#### 单体应用 **monolithic application**

应用程序的所有组件和功能都紧密集成到单个代码库中,作为单个单元执行,并且通常部署为单个应用程序。

## 4 缩略语

下列缩略语适用于本文件。

API:应用程序编程接口(Application Programming Interface)

BI:商业智能(Business Intelligence)

DaaS:数据即服务(Data as a Service)

DevOps:软件开发和系统运维相结合的实践(Development Operations)

IaaS:基础设施即服务(Infrastructure as a Service)

PaaS:平台即服务(Platform as a Service)

SaaS:软件即服务(Software as a Service)

URL:统一资源定位符(Uniform Resource Locator)

## 5 工业软件云的架构设计

### 5.1 概述

工业软件云的架构是指将传统的工业软件服务迁移到云计算平台上的技术框架和解决方案。以云原生架构为基础,充分利用了云计算的弹性和分布式协同能力,以云化、服务化为核心进行构建。工业软件云中工业应用通常构建为在容器中运行的一组微服务,经容器编排软件进行编排配置,并基于自动化、流程化的流水线进行更新和管理。云架构下工业软件有更快的迭代速度、持续可用的服务、弹性伸缩及其他一些非功能特性。

### 5.2 工业软件云的架构设计原则

工业软件云的架构设计中宜满足以下原则:

- a) 整体性和开放性的原则:在工业软件云的架构设计时宜充分考虑生产部门、管理部门和其他成员之间的关系,整体设计规划本系统,注重各种信息资源的有机整合;既考虑安全性,同时也考虑具有一定的开放性,平衡信息共享和信息安全之间的关系;
- b) 可扩展性和易维护性的原则:在工业软件云的架构设计时应具有一定的前瞻性,充分考虑系统升级、扩容、扩充和维护的可行性,以及将来其他子系统集成,或与第三方系统集成信息共享等;
- c) 经济性和实用性的原则:系统的设计实施尽最大可能节省系统投资,具有较好的性能价格比,设计面向实际,注重实效,坚持实用、经济的原则,充分合理利用现有设备和信息资源,帮助用户节省投资;

- d) 先进性和成熟性的原则:在系统设计时,将充分应用先进和成熟的技术,满足建设的要求,基于先进的技术手段,提出合理的业务流程;系统将使用先进成熟的技术手段和标准化产品,使系统具有较高性能,符合当今技术发展方向,确保系统具有较强的生命力,有长期的使用价值,符合未来的发展趋势;
- e) 可靠性和稳定性的原则:系统各环节具备故障分析与恢复和容错能力,并在安全体系建设、复杂环节解决方案和系统切换等各方面考虑周到、切实可行,建成的系统需安全可靠,稳定性强,把各种可能的风险降至最低;
- f) 安全性和保密性的原则:在系统设计把安全性放在首位,既考虑信息资源的充分共享,也要考虑信息的保护和隔离;系统在各个层次对访问都进行了控制,设置了严格的操作权限;并充分利用日志系统、健全的备份和恢复策略增强系统的安全性;
- g) 统一的数据模型,基于底层统一的数据模型,减少格式转换,减少物理文件存取,不再需要大数据量的文件下载。

### 5.3 工业软件云的通用架构

#### 5.3.1 通用架构

工业软件云的通用架构如图 1 所示。其技术部分依赖于在传统云计算的 4 层概念:基础设施即服务(IaaS)、数据即服务(DaaS)、平台即服务(PaaS)和软件即服务(SaaS),自下而上包括通用的基础云环境,工业软件平台和面向工业场景的各类应用服务等。



图 1 工业软件云的通用架构

#### 5.3.2 基础云环境

基础云环境提供大规模云存储和云计算技术服务能力,提供计算资源、存储资源、网络资源的虚拟化服务,支撑工业软件云平台安全稳定运行。又可分为 IaaS 层和 DaaS 层。

- a) IaaS层基于标准虚拟化技术,将物理服务器的计算和存储能力以及网络设备虚拟化成虚拟计算、分布式存储和软件定义网络。其中,计算包含消息队列、网络控制、卷管理、运算工作组件等,存储包含对象存储、块存储、文件存储组件,网络管理包含虚拟网络、租户网络组件,访问控制等,并在此基础上提供云数据库、大数据处理、分布式中间件服务,为用户的应用系统提供 IT 基础服务的支撑能力。
- b) DaaS层提供大数据交换、存储、管理、计算和分析等公共服务。数据交换实现对数据进行交换

和清理处理;支持分布式文件存储数据库、关系型存储数据库,列式存储数据库、文档存储数据库等实现工业领域结构化数据、半结构化数据、非结构化数据的存储;提供大数据管理工具和集成的第三方大数据管理引擎,解决大数据存储与分析工具的集群化部署、服务配置管理、服务状态监控展示的问题,降低大数据平台运维和使用的难度。

### 5.3.3 平台服务层

#### 5.3.3.1 通用服务构建

##### 5.3.3.1.1 通用服务的 IT 实现

通用服务基于容器+微服务+DevOps 的研发思路,提供一站式研发、管理、测试、运维服务。平台服务一般基于业界主流的混合容器技术、开放灵活的微服务技术,具备弹性伸缩和服务编排功能。提供应用发布、持续集成、代码托管、服务绑定等能力。支持开发者将应用快速发布到云端运行环境,能够提供面向行业的服务治理能力及自动化运维支持,并提供多语言应用引擎环境(Java、Go、PHP、Ruby 等)。基于容器化技术为应用提供高效、稳定和弹性的服务运行环境。附录 A 介绍了云环境下工业应用服务持续集成与交付的流程与原则。

##### 5.3.3.1.2 通用服务的功能

通用服务基于全生命周期理念工业应用在云环境下的运维,提供启停、伸缩、版本切换等控制和管理功能,支持用户可在部署界面上实现组合式创建应用及一键部署,从而实现对应用准备、应用部署、应用运维、应用营销全生命周期的监控管理。有下述六个功能:

- a) 资源概况:包括应用概览,用量概览,开发者可以通过界面直观看到应用情况和资源用量;
- b) 应用管理:使用者可以通过应用管理实现应用的部署、更新、停止、查看等一系列对应用的管理操作;
- c) 代码托管:提供使用者代码托管和持续集成;
- d) 平台管理:提供平台的组织、空间、域名、配额、账号等平台基础服务管理;
- e) 监控管理:提供组件与服务详情、当前使用状况等监控;
- f) 平台中间件管理:提供了应用全生命周期管理的日志服务。

#### 5.3.3.2 工业领域专用服务构建

根据工业领域具体的业务功能,提供多种服务,包括但不限于通用服务组件、数据库服务和领域专用服务组件、工业微服务、第三方微服务组件等,全面支持各类工业应用的快速开发与迭代,具体内容应符合以下要求。

- a) 基本功能和智能化服务。宜包含消息中间件服务、对象存储服务、工作流引擎、工业数据管理、计算引擎、BI报表工具、多维分析、数据可视化、事件发布与订阅、数据云网关服务等功能服务和大数据分析算法库、人工智能算法库、知识图谱、视频监控、图像检索等智能应用服务。
- b) 领域专用服务组件。针对具体业务场景有大修组件、规程组件、仿真建模部件、事故推演、工程组态、几何组件、设备维修、设备诊断、安全评估、工业信息模型服务、工业时序储存服务、工业事件与告警等领域专用服务。

### 5.3.4 工业应用服务层

平台应用服务层宜提供满足工业行业全产业链中包含设计、生产、管理、服务等一系列创新性业务应用,如生产运行管理,设备管理、维修管理等业务领域应用,目的是提高工厂硬件、服务、数据的使用效率

和共享范围,实现业务与资源的智能管理及运营优化,体现工业软件云平台的最终价值。

#### 5.4 信息安全防护体系

平台的安全建设参考国内外发布的安全标准,通常按照国家信息安全等级保护 3 级(安全标记保护级)要求,从物理资源与接入安全、基础设施安全、网络安全、控制安全、数据安全、应用安全等多个维度提供全方位的安全防护,提供工具库、病毒库、漏洞库等安全防护模块,打造安全可靠、自主可控的工业云平台。

### 6 基于微服务架构的工业应用开发与改造

#### 6.1 概述

传统工业应用通常为单体软件。工业软件云架构基于微服务架构实现。微服务架构是以单一功能组件为基础,通过模块化组合方式实现“松耦合”应用开发的软件架构。一个微服务就是一个面向单一功能、能够独立部署的小型应用,将多个不同功能、相互隔离的微服务按需组合在一起并通过 API 集实现相互通信,就构成了一个功能完整的大型应用系统。工业软件云架构以工业微服务为基本的功能模块,对原有生产体系进行解构,从而在平台中构建起富含各类功能与服务的微服务组件池,通过根据实际需求调用相应的微服务组件,进行高效率 and 个性化的面向用户的工业 App 研发。与单体软件架构相比较,微服务架构恰恰弥补了单体架构的不足,通过有效的拆分应用,实现敏捷开发和部署,有如下特点:

- a) 由多个独立的微服务共同组成系统;
- b) 微服务单独部署,运行在自己的进程中;
- c) 每个微服务为独立的业务开发;
- d) 分布式管理;
- e) 非常强调隔离性。

#### 6.2 基于微服务架构的工业应用系统开发

通过微服务技术可以在平台上快速开发、部署、发布工业应用系统。其步骤如下。

- a) 抽象或归纳企业的业务流程,抽取各个业务相关应用系统的功能模块,按照企业自己的业务流程走向进行的集成。建议按照现代企业的流程走向进行调整,从现有的信息系统中抽取相关业务功能模块,使各个应用系统按照业务流程来运作。
- b) 抽取的功能模块通过集成机理模型与算法实现工业微服务开发。不同应用系统的不同功能模块通过微服务的配置来实现触发和调用,通过容器化应用运行环境,为微服务提供路由服务、用户认证授权、日志监控、服务代理等整套功能体系。
- c) 应用流程管理通常以浏览器方式实现信息传递、数据同步、业务监控和企业业务流程的持续升级优化。各个业务系统的集成是紧耦合的,这样对于集成后的外在展示门户而言,各个应用系统是一个整体,业务流就在这个整体中流动。

#### 6.3 已有工业业务系统的微服务化改造

已有工业软件和业务系统拆分为微服务组件的过程中宜遵循以下原则。

- a) 隔离:
  - 1) 服务宜设计为相互隔离工作,当将一个整体系统分解成一组服务时,这些服务宜彼此解耦,这样才能更加连贯和独立;
  - 2) 每个服务宜能处理其自己的故障,而不会影响或破坏整个应用程序或系统;

- 3) 隔离和解耦特性使服务能非常快速地从故障状态中恢复,服务的隔离特性能支持持续交付和有效监控,具有更好的扩展性和可测试性。
- b) 自治性:
  - 1) 服务应设计为自主自治的,应具有凝聚力,能独立地实现其职能;
  - 2) 每项服务可使用良好定义的 API(URI)独立调用,API以某种方式标识服务功能;
  - 3) 自主服务应处理自己的数据,每个服务都有自己的持久存储;
  - 4) 自主服务支持服务编排和协调,更好扩展,通过良好定义的 API进行通信,实现更快速和可控的部署。
- c) 单一责任:
  - 1) 服务应设计为高度凝聚;
  - 2) 单一的职责(责任)原则是服务只执行一个重要的功能;
  - 3) 单一责任功能保障服务的可重用性,可扩展性和可维护性。
- d) 异步通信:
  - 1) 设计离散边界和使用其自己的有界上下文设计服务时,跨边界的服务通信应是异步的;
  - 2) 异步通信模式自然导致服务之间的松耦合,并允许更好的缩放;
  - 3) 异步通信还可带来实现良好定义的集成或通信模式的概念,以实现涉及不同服务的逻辑工作流。
- e) 位置独立:
  - 1) 根据设计,微服务是在虚拟化环境或容器中部署;
  - 2) 应能以位置独立的方式来寻址或定位服务;
  - 3) 服务的客户端或消费者不必考虑部署或配置特定服务的位置,仅使用某种逻辑或虚拟地址来定位服务。

## 附录 A

(资料性)

## 云环境下工业应用的持续集成与交付

## A.1 基于 DevOps 的应用的持续集成

A.1.1 DevOps 提倡通过一系列的技术和工具减少开发和运维之间的隔阂,实现从开发、构建到最终部署的全流程自动化,从而达到开发运维一体化。

A.1.2 持续集成指在软件开发过程中,软件开发人员持续不断地将开发出来的代码和其他开发人员的代码进行合并,每次合并后自动进行编译、构建,并运行自动化测试进行验证,而不是等到最后各自开发完成后才合并在一起。

A.1.3 持续集成能从根本上提高一个团队的软件开发效率。

A.1.4 持续集成的实施要求具有良好的团队纪律,团队成员可执行持续集成过程中的各项规则,基于良好的配置管理,借助自动化工具创建并维护一个自动化构建和测试流程。

A.1.5 一个 DevOps 开发环境需要满足以下要求:

- a) 环境一致性:在本地开发出来的功能,无论在什么环境下进行部署,都应能得到一致的结果;
- b) 代码自动检查:每次代码提交后,系统都应自动对代码进行检查,以便及早发现潜在的问题,并运行自动化测试;
- c) 持续集成:每次代码提交后,系统可自动进行代码的编译和打包,无须运维人员手动进行;
- d) 持续部署:代码集成完毕后,系统可自动将运行环境中的旧版本应用更新成新版本应用,并且整个过程中不会让系统不可用;
- e) 持续反馈:在代码自动检查、持续集成、持续部署的过程中,一旦出现问题,要能及时将问题反馈给开发人员和运维人员,开发人员和运维人员收到反馈后对问题及时进行修复;
- f) 快速回滚:当发现本次部署的版本出现问题时,系统应能快速回退到上一个可用版本;
- g) 弹性伸缩:当某个服务访问量增大时,系统应对这个服务快速进行扩容,保证用户的访问;当访问量回归正常时,系统能将扩容的资源释放回去,从而实现根据访问情况对系统进行弹性伸缩;
- h) 可视化运维:提供可视化的页面,可实时监控应用、集群、硬件的各种状态。

A.1.6 云环境下工业应用持续集成流程如图 A.1 所示,应包括以下步骤:

- a) 开发人员在本地开发并验证完功能后,将代码提交到代码仓库;
- b) 通过事先配置好的触发器通知方式,当开发人员提交完代码后,部署在云端的持续集成工具会实时感知,并从代码仓库中获取最新的代码;
- c) 获取到最新代码后,集成工具会启动测试平台对最新的代码进行检查以及执行单元测试,执行完成后在测试平台上生成测试报告;如果测试未通过,则以发送电子邮件的方式通知研发人员进行修改,终止整个流程;如果测试通过,则将结果反馈给持续集成工具并执行下一步;
- d) 代码检查以及单元测试通过后,持续集成工具会将代码发送到持续集成服务器中,在服务器上对代码进行编译、构建,然后打包成能在容器环境中运行的镜像文件;如果中间有步骤出现问题,则通过发送电子邮件的方式通知开发人员和运维人员进行处理,并终止整个流程;
- e) 将镜像文件上传到私有镜像仓库中保存;
- f) 镜像上传完成后,持续集成工具会启动持续交付服务器,对云环境中运行的应用进行版本更新,

整个更新过程中会确保对服务的访问不中断;持续交付服务器会将最新的镜像文件拉取到容器集群中,并采用逐步替换容器的方式对应用进行更新,在服务不中断的前提下完成更新。

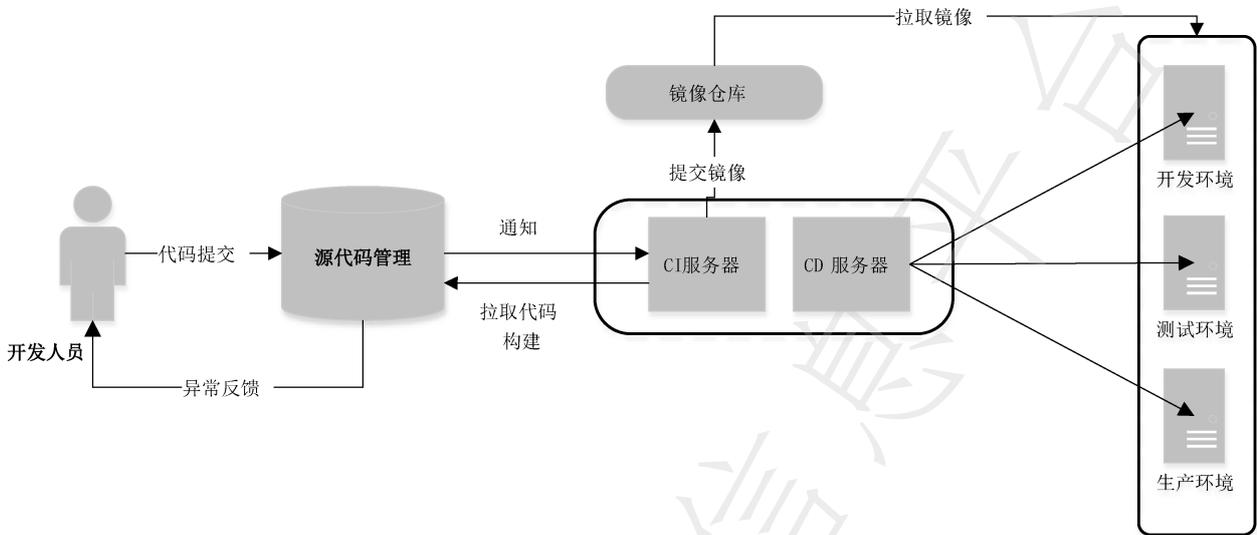


图 A.1 云环境下工业应用持续集成流程

## A.2 基于容器的持续交付

A.2.1 容器化技术也同样可应用到持续交付中。最显著的好处是在打包过程中,如果在一个传统的非容器化项目里打包,包括代码、配置文件、环境变量配置等工作都需要人工设计、规划和操作;但如果以容器的镜像文件为对象来进行操作,只需要考虑注入参数,打包是更完整的、更彻底的,包括操作系统参数的打包,在打包时不仅进行了代码的编译,甚至完成了应用安装的过程,将应用程序和运行环境所需要的都进行了打包,最后整合成一个镜像文件。容器为持续交付带来了标准规范,非常有利于应用程序在不同环境间的流转和发布,提升了交付效率,这充分体现了DevOps里面所谓的配置即代码、基础设施即代码的核心理念。

A.2.2 如图 A.2 所示,容器化之后的部署对象不是 App 或者是 Java 文件,而应是容器镜像文件;且也不应关注运行在什么操作系统下,操作系统如何配置。在搬运时移动的是镜像文件,运行时是容器镜像;只需要关注外面注入的环境参数即可,而且在运行过程中基于容器编排软件来做统一的管理。基于容器化的环境再做 DevOps,代码搬运、打包、运行监控等操作都已经把对象从 App 转到容器上来了。容器化带来的是整个持续集成效率的提升,这是一个组件化到整体化的质变。

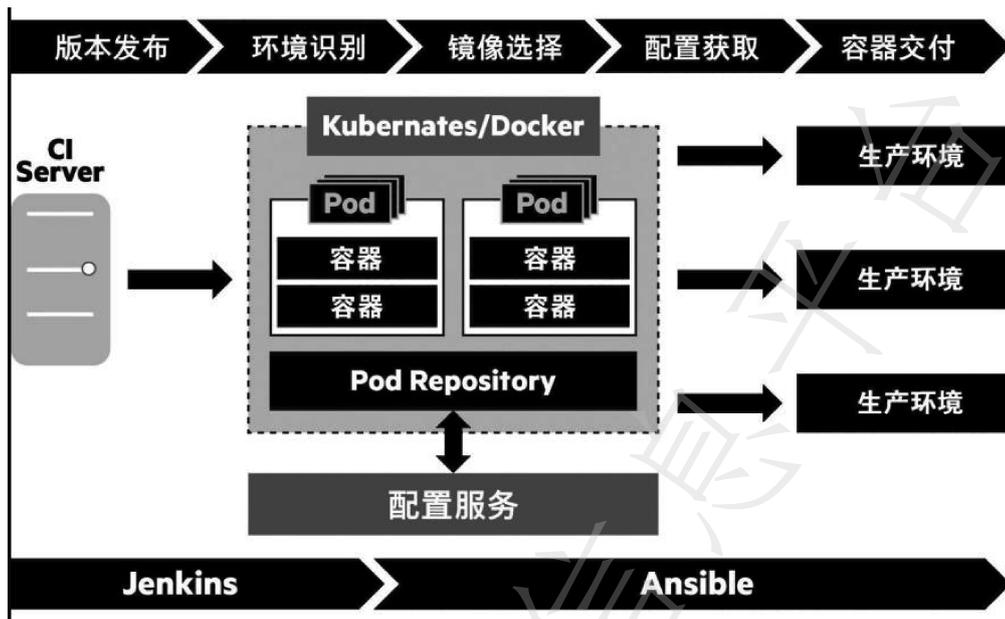


图 A.2 容器化持续交付

参 考 文 献

- [1] YD/T 4398—2023 电信行业云原生平台架构与技术要求
  - [2] YD/T 4409.1—2023 云原生能力成熟度模型 第1部分:技术架构
  - [3] ISO/IEC 20000-1:2018 Information technology—Service management—Part 1: Service management system requirements
  - [4] ISO/IEC 22123-1:2023 Information technology—Cloud computing—Part 1: Vocabulary
  - [5] ISO/IEC 25002:2024 Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—Quality model overview and usage
  - [6] ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection—Information security management systems—Requirements
  - [7] ISO/IEC 27017:2015 Information technology—Security techniques—Code of practice for information security controls based on ISO/IEC 27002 for cloud services
  - [8] ISO/IEC 29119 (all parts) Software and systems engineering—Software testing
  - [9] ISO/IEC 38500:2024 Information technology—Governance of IT for the organization
  - [10] ANSI/ISA-95 Enterprise—Control System Integration
  - [11] IEEE 1471—2000 IEEE Recommended Practice for Architectural Description for Software—Intensive Systems
-







中国国际科技促进会  
团体标准  
工业软件云架构通用技术指南  
T/CI 823—2024

\*

中国标准出版社出版发行  
北京市朝阳区和平里西街甲2号(100029)  
网址 www.spc.net.cn  
总编室:(010)68533533 发行中心:(010)51780238  
读者服务部:(010)68523946  
中国标准出版社秦皇岛印刷厂印刷  
各地新华书店经销

\*

开本 880×1230 1/16 印张 1.25 字数 21 千字  
2025年3月第1版 2025年3月第1次印刷

\*

书号:155066·5-11065 定价 43.00 元

如有印装差错 由本社发行中心调换  
版权所有 侵权必究  
举报电话:(010)68510107



T/CI 823-2024