

PAS

面向未来架构的企业级应用服务器

PAS 产品白皮书

目录

1	PAS 产品定位	4
2	PAS 产品概述	5
3	PAS 产品客户价值	7
3.1	保障企业级应用的快速、平滑迁移	7
3.2	支持分布式微服务架构下的新应用	7
3.3	提供主流中间件层的完整解决方案	7
3.4	全国产信创生态的全适配与性能优化，保障业务系统的可靠运行 ..	8
4	PAS 产品特点	9
4.1	支持完整的 Jakarta EE8 特性	9
4.2	兼容国产生态	11
4.3	提供量身定制的多形态产品	11
4.4	性能卓越	12
4.5	完善的安全策略控制	12
4.6	开放统一的服务器扩展架构	13
4.7	应用性能监控和快照能力	13
4.8	充分保障应用的高可用、高可靠	13
5	PAS 产品功能	13
5.1	微内核	14
5.2	Web 容器	15
5.3	EJB	16
5.4	数据库	17
5.5	JCA	18
5.6	JNDI	18
5.7	JMS	19
5.8	并行资源	20
5.9	ORB	20
5.10	邮件	21
5.11	监控	22
5.12	批处理	22
5.13	集群	23
5.14	应用部署	24
5.15	安全	25
5.16	域管理	26
5.17	应用迁移工具	27
5.18	分布式缓存	27
5.19	微服务架构支撑能力	28
5.20	应用性能监控	28
5.21	应用支撑	29
6	PAS 产品环境配置	30
6.1	支持的 CPU	30

6.2	支持的操作系统	30
6.3	支持的数据库	31
6.4	支持的浏览器	31
6.5	支持的 JDK 版本	31
7	客户情况	32
8	联系我们	33
9	关于普元	35

1 PAS 产品定位

应用服务器中间件又称“应用服务器软件架构平台”，是当代软件的基础设施，也是最为重要的一类中间件。应用服务器把不同的应用软件作为构件整合到一个协同工作环境中，并为应用提供了消息，连接，事务，安全，数据访问等标准能力，稳定支撑应用的开发，部署，运行及管控。

近年来，随着企业自主掌控要求以及数字化转型的迅猛发展，应用服务器中间件的发展态势良好，国内主要中间件厂商纷纷在原有基础上进行产品升级，以适应当下的应用新标准、新架构。普元应用服务器产品紧跟发展趋势，不仅实现了 **Jakarta EE** 最新发布标准，还在新应用架构上（如微服务、容器化等）提供了友好支撑，帮助企业应用在双模的架构形态下稳定运行，稳步升级。

应用服务器在整个中间件平台层中，更像是一个桥接器或适配器，对下需要屏蔽各数据库、操作系统的差异，对上则要支撑各类业务基础软件平台的运行。在国产系统、数据库、工作流等中间件的市场占有率不断提升的背景下，其适配后的稳定性、性能要求变得尤为重要。**PAS** 默认提供对主流国产中间件的适配，通过标准化接口抽象，支持快速集成新品中间件，让客户中间件平台的选择更灵活。

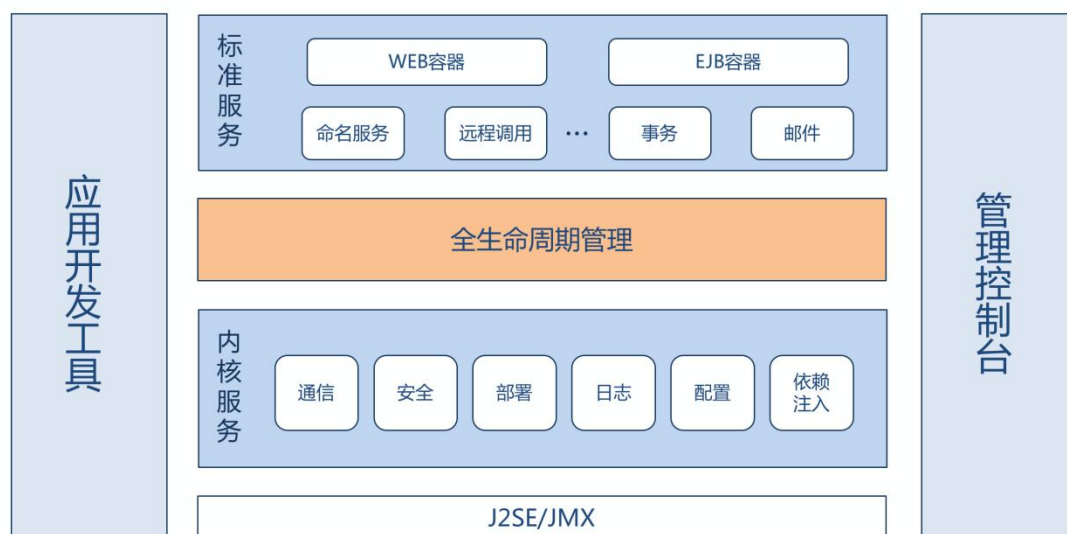


2 PAS 产品概述

应用服务器中间件是普元多年来主打的产品之一，公司于 2010 年启动第一个版本的研发，12 年正式推出 Java Platform Enterprise Edition 5 (Java EE 5)兼容的版本，近 10 年来，一直紧跟标准的发展不断升级优化，最新推出的 6.5 版本已完全兼容 Jakarta EE 8 规范。PAS 产品采用 hk2+osgi 的双核心架构，在企业级特性上，比如高可用性、可伸缩性、集群、综合监控、SOA、控制台易用性等方面具有不输于国外商用应用服务器的能力。

PAS 产品中，通过管理服务器对各服务器节点进行管理，各服务器节点中包括内核服务、生命周期管理、标准服务三部分：

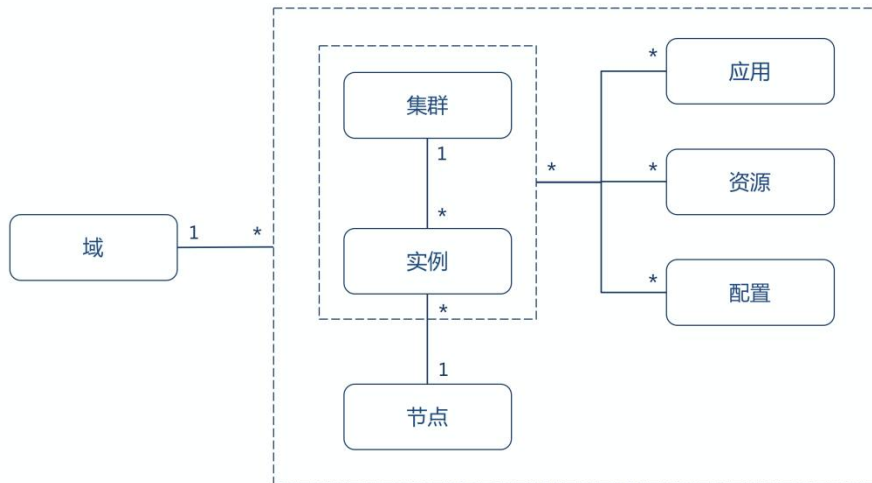
- 内核服务抽取了 Jakarta EE 各子项标准的公共能力，为上层的标准服务提供支撑能力；
- 全生命周期管理模块为应用提供生命周期事件管理，用户可基于标准接口扩展；
- 标准服务与 Jakarta EE 的标准一一对应，采用可插拔架构实现，包括 Web 容器、EJB 容器等，为应用运行、调用提供标准支撑；



PAS 6.5 提供企业版介质，结合当下应用架构形态和云化需要，可以进行量身定制，满足微服务、嵌入式及容器化的使用场景。

- PAS 6.5 企业版：支持完整 Jakarta EE 标准的版本，与 Jboss、Weblogic 类似，面向企业级应用架构；

通过对概念模型的抽象，普元应用服务器管理的一级对象包括：



- 域：是最外层的管理概念，管理员在控制台中可管理一个域，支持配置域的访问密码、日志存储等信息。
- 应用：对应标准部署物的部署后运行形态，比如 **war**、**ear**、**rar**、**jar**，这些部署后都称为应用。
- 实例：一个实例对应一个运行进程，类似启动一个 **tomcat** 实例，一个实例中可承载多个应用的运行。
- 集群：一个集群包含一个或多个实例，一个集群中的实例可统一配置管理，集群内实例支持会话同步。
- 节点：一个节点通常对应一个主机（也可以通过对主机的端口等切分，让一台实际主机作为多个节点来管理），实例绑定主机。
- 配置：配置管理的是实例运行的一些参数，比如 **JVM** 参数、会话存储方式、监控项、**HTTP** 服务，配置可与实例和集群关联（关联集群意味着最终作用到集群中的每个实例）。
- 资源：资源在 **Jakarta EE** 中是一个相对抽象的概念，包括线程池、数据库连接池、**JMS**、连接器等，资源注册后可与实例或集群关联，应用可通过 **JNDI** 查找使用。

3 PAS 产品客户价值

3.1 保障企业级应用的快速、平滑迁移

企业原有应用系统多数依赖于 Tomcat、Weblogic、Websphere、Oracle、DB2 等中间件，而随着自主掌控的要求，springboot、dubbo 等技术已经成了新应用建设的首选，但架构理念的区别导致应用很难短时间完成重构迁移。

PAS 产品可帮助客户在极少量代码变更的前提下完成迁移，同时也可为应用的新老架构并存提供兼容的运行环境。

3.2 支持分布式微服务架构下的新应用

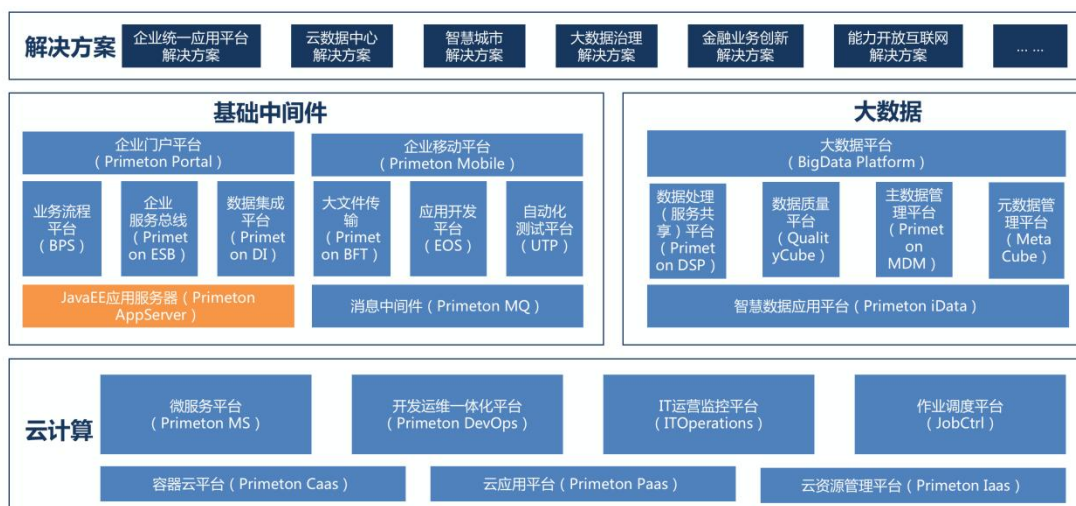
企业 IT 建设的一大挑战是新技术变革带来的架构冲击，应用服务器作为传统中间件，无论其本身技术，还是可支撑的应用技术，都需要与时俱进。

运行于 PAS 产品上的应用，既可以采用 Jakarta EE 标准开发，也可以采用 springcloud 微服务架构开发，应用服务器在运行期可根据需要自行加载对应组件和配置，让客户在业务发展的同时，亦能保持技术竞争力。

3.3 提供主流中间件层的完整解决方案

中间件层的品目众多，不同厂商之间的中间件集成易产生瑕疵，很多中间件或平台厂商，习惯于将这类问题放在客户大平台建设中来解决，往往会带来一些未知风险。

普元中间件产品家族里，除应用服务器外，还提供包括业务流程、文件传输、数据集成、消息中间件、服务总线等其他标准产品 PAS 支持与家族中其他中间件的无缝集成，同时与多套国产中间件完成了测试验证，可为客户提供一体化的平台解决方案，让客户更聚焦在于业务创新。



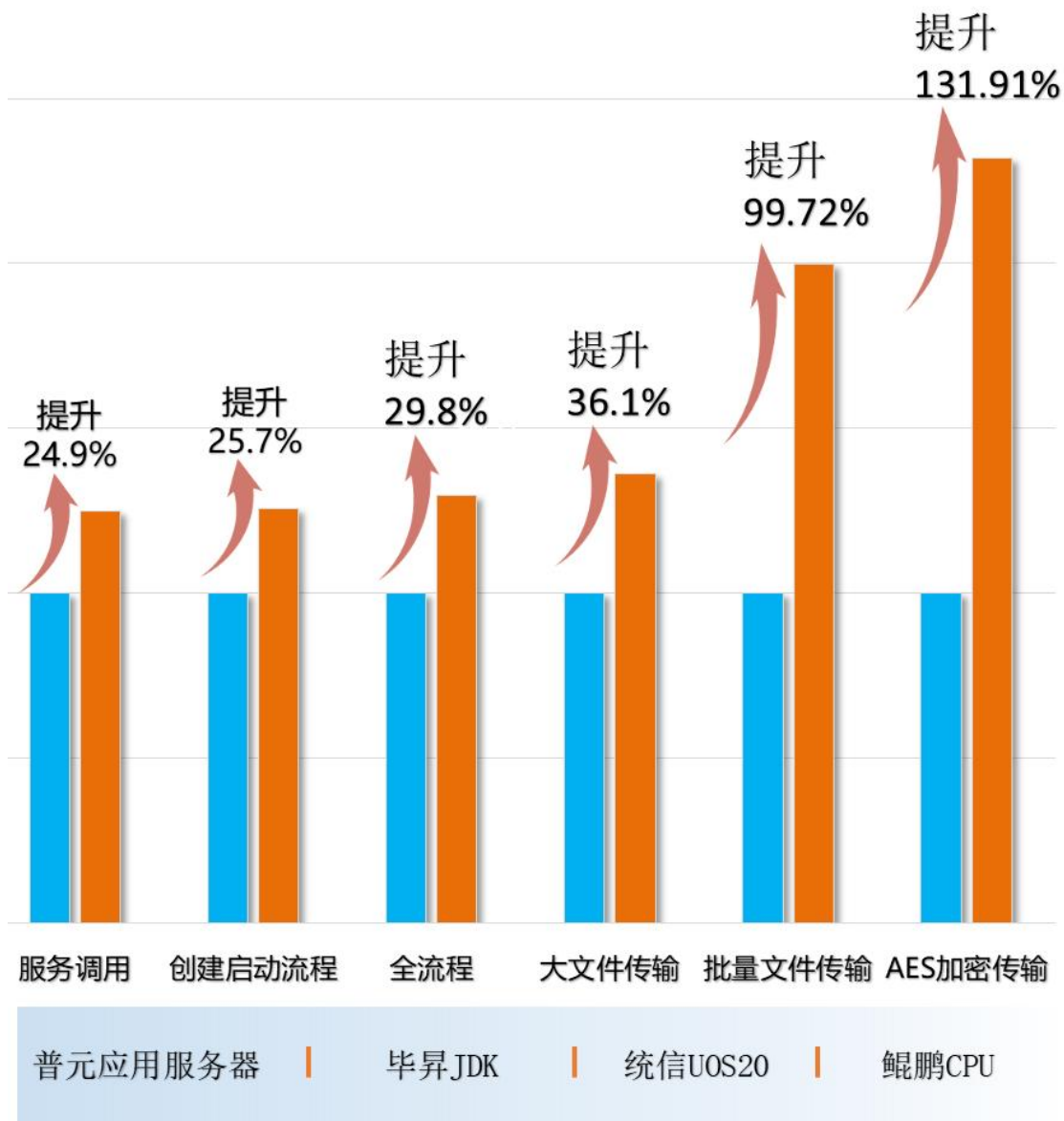
3.4 全国产信创生态的全适配与性能优化，保障业务系统的可靠运行

信创生态产业链涵盖上游、中游和下游几大环节。上游主要是以 CPU、服务器为代表的底层硬件和基础设施；中游包括中间件、数据库、操作系统这些基础软件和平台服务；下游包括各种应用软件、企业应用和解决方案。普元信息技术股份有限公司产品响应国产化趋势，积极打造国产化生态环境，确保客户业务系统在信创环境中高质量，高可靠，平稳的运行。



普元技术平台与中间件产品专门针对国产信创环境进行深度性能优化,在与华为鲲鹏实验室深度合作中,普元应用服务器性能针对鲲鹏体系进行了深度性能优化,性能较主流方案性能提升 **25% 以上**。

普元中间件产品,可以根据客户选择的其它信创环境,针对大并发的场景,进行有针对性的优化提升。



4 PAS 产品特点

4.1 支持完整的 Jakarta EE8 与 9.1 特性

PAS 支持 Jakarta EE 9.1 与 Jakarta EE8 完整特性，普元公司亦通过 Eclipse jakarta 的官方认证，是目前国内唯一一家 jakarta ee 的企业级成员。

(<https://jakarta.ee/membership/members/>)。

PAS 最新版本重点对变更较大的 Servlet5.0、Bean validation3.0、CDI3.0 规范进行了对应升级，并重点实现了包括 JSON-B、Security API 等全新规范。

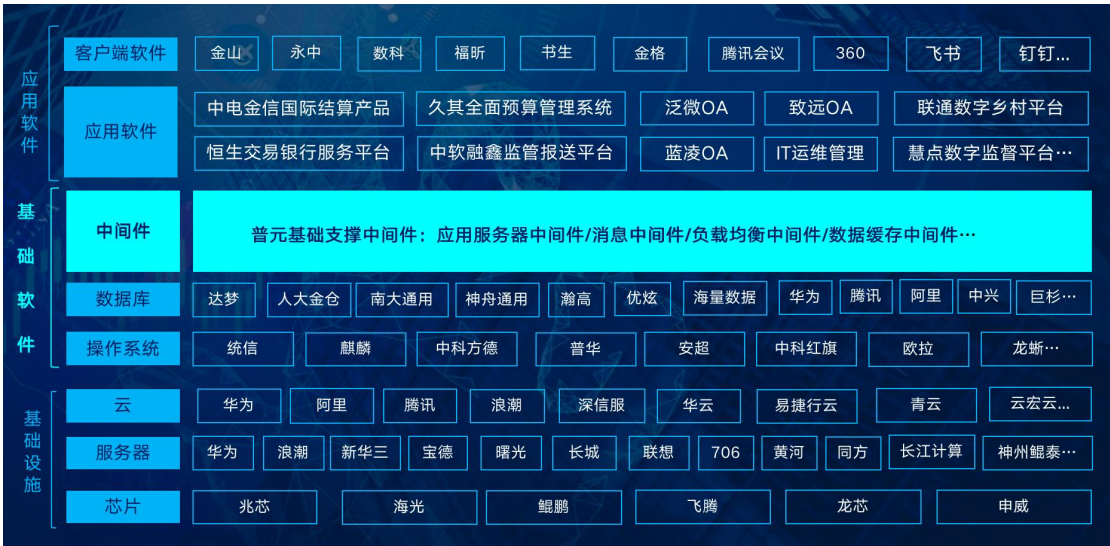
附：Jakarta EE8 相对 Java EE8 的主要命名与版本变更：

Jakarta EE 8	Java EE 8
Jakarta™ EE Platform 8	Java™ Platform, Enterprise Edition 8
Jakarta™ Enterprise Beans 3.2	Enterprise JavaBeans™ 3.2
Jakarta™ Annotations 1.3	Common Annotations for the Java Platform 1.3
Jakarta™ Servlet 4.0	Java™ Servlet 4.0
Jakarta™ WebSocket 1.1	Java API for WebSocket 1.1
Jakarta™ Server Faces 2.3	JavaServer Faces 2.3
Jakarta™ Server Pages 2.3	JavaServer Pages™ 2.3
Jakarta™ Standard Tag Library 1.2	Standard Tag Library for JavaServer Pages 1.2
Jakarta™ Expression Language 3.0	Expression Language 3.0
Jakarta™ Debugging Support for Other Languages 1.0	Debugging Support for Other Languages 1.0
Jakarta™ Messaging 2.0	Java™ Message Service 2.0
Jakarta™ Transaction 1.3	Java™ Transaction API 1.2
Jakarta™ Mail 1.6	JavaMail™ API 1.6
Jakarta™ Connectors 1.7	Java EE™ Connector Architecture 1.7
Jakarta™ Enterprise Web Services 1.4	Web Services for Java EE 1.4
Jakarta™ XML RPC 1.1	Java™ API for XML-based RPC 1.1
Jakarta™ XML Registries 1.0	Java™ API for XML Registries 1.0
Jakarta™ RESTful Web Services 2.1	Java™ API for RESTful Web Services 2.1
Jakarta™ JSON Processing 1.1	Java API for JSON Processing 1.1
Jakarta™ JSON Binding 1.0	Java API for JSON Binding 1.0
Jakarta™ Management 1.1	Java™ Platform, Enterprise Edition Management 1.1
Jakarta™ Deployment 1.7	Java™ Platform, Enterprise Edition Deployment 1.2
Jakarta™ Authorization 1.5	Java™ Authorization Service Provider Contract for Containers 1.5
Jakarta™ Authentication 1.1	Java™ Authentication Service Provider Interface for Containers 1.1
Jakarta™ Security 1.0	Java™ EE Security API 1.0
Jakarta™ Persistence 2.2	Java Persistence 2.2
Jakarta™ Bean Validation 2.0	Bean Validation 2.0
Jakarta™ Managed Beans 1.0	Managed Beans 1.0
Jakarta™ Interceptors 1.2	Interceptors 1.2 rev A
Jakarta™ Contexts and Dependency Injection 2.0	Contexts and Dependency Injection for the Java EE Platform 2.0
Jakarta™ Dependency Injection 1.0	Dependency Injection for Java 1.0
Jakarta™ Concurrency 1.1	Concurrency Utilities for Java EE 1.0
Jakarta™ Batch 1.0	Batch Applications for the Java Platform 1.0 rev A

4.2 兼容国产生态

随着信息技术领域逐步走出开放自研、安全可信的道路，在关键技术领域的自主能力变得非常重要（具体包括：主板处理器、整机、存储、BIOS、操作系统、数据库、中间件、办公软件、应用软件）。

PAS 产品隶属中间件领域，在国产生态的适配方面再次增强，提供对各种中文字符编码问题的容错支持，兼容国内外主流数据库、操作系统，具体兼容环境可参阅产品环境章节。

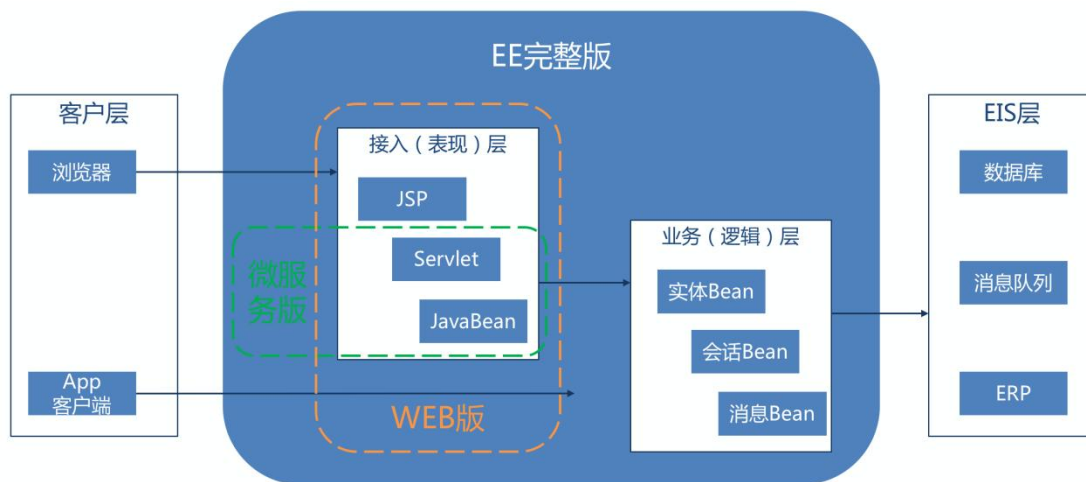


4.3 提供量身定制的多形态产品

作为 Jakarta EE 标准产品，一般包括四层结构：

- 客户层，运行在客户计算机上的组件。
- Web 层，运行在 Jakarta EE 服务器上的组件。
- 业务层，同样是运行在 Jakarta EE 服务器上的组件。
- 企业信息系统层（EIS），是指运行在 EIS 服务器上的软件系统。

结合当下流行的服务架构、云架构，上述四层会有一些取舍，且对外提供形态也会存在多样性，PAS 自身是一个插拔式架构，可根据用户需求量身定制。



4.4 性能卓越

PAS 6.5 版本进行了大量性能优化工作，包括线程池调用优化、改进的 Session 持久化、主机系统管理优化、jsp 编译器优化等。

PAS 6.5 将丰富的性能调优经验直接落地到产品中，支持性能参数配置化：

- 支持部署时设置 JSP 是否预编译，大幅提升首次访问体验；
- 引入 EJB 高速缓存，增强对状态的检测；
- 提供文件缓存设置，提高包含静态文件的应用程序的性能
- 修改 Session 存储算法，提高 Session 存储效率；
- 默认支持了 HTTP 和 RMI-IIOP 负载均衡；

4.5 完善的安全策略控制

PAS 6.5 支持高可用环境中部署应用。产品从协议、证书、会话、审计、预警、三员管理等维度，保护对应用的访问，防止恶意攻击，最终保障企业数据安全。

提供负载均衡与备份能力，支持动态扩展的功能实现集群管理。

通过监控管理能力对服务运行情况进行实时跟踪，提供便捷的、脱敏的日志管理功能以供审计。PAS 6.5 同时与普元开发运维类产品无缝集成，确保应用从开发到上线整个生命周期的安全可控。

4.6 开放统一的服务器扩展架构

伴随应用服务器功能特性日渐庞大，产品微内核化是一个必然趋势，通过微内核减少内部接口的复杂性，提供公共的基础能力供其他标准服务使用。PAS 产品基于 HK2+OSGI 双内核的设计，既统一了内部各标准服务的实现架构，也统一了外部三方能力的集成方式，从而提供更好的扩展性和更有效的应用环境。

4.7 应用性能监控和快照能力

提供应用性能监控软件，破解应用黑盒难题，提供无侵入式的、细粒度的应用服务监控能力，帮助用户掌握系统运行情况、定位性能瓶颈，确保系统可靠运行。

应用服务器提供导出服务器快照能力，可以实时生成快照文件，方便用户搭建问题验证环境，对内存 dump 进行分析定位，同时可以设定内存使用阈值，一旦应用内存使用超过设定的阈值，平台自动 dump 内存快照，供相关人员下载分析。

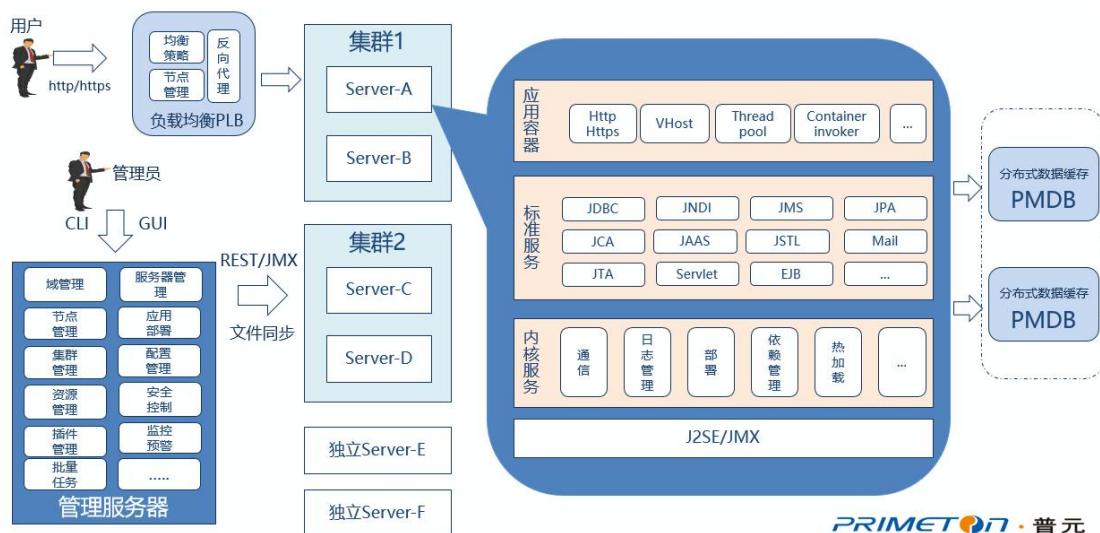
4.8 充分保障应用的高可用、高可靠

应用服务器作为应用的载体，需要在可用性、可靠性上有一定的保障，尤其是很多企业的核心系统都部署在应用服务器上，如果经常因为各种原因导致应用的不可访问，势必给企业带来很多麻烦，普元应用服务器在高可用、高可靠方面从多个维度进行保障：

- 应用的集群部署模式，解决应用单点问题
- 前端架设普元负载均衡组件（PLB），利用健康检查机制保障后端应用的可用性
- 通过实时监控实例状态，在实例状态异常停止服务后，自动重新启动
- 为应用提供滚动升级模式，让应用真正做到不停机更新

5 PAS 产品功能

如产品概述章节所述，PAS 6.5 产品包括管理控制台+开发工具+运行引擎三部分。管理控制台负责多个运行引擎的统一管理，开发工具则是可选组件，用于应用的标准化开发与快速调试。



运行引擎中，内核服务层提供了底层的通信、日志、配置、依赖管理等核心功能，标准服务则是遵循 Jakarta EE 8 标准实现的各类服务，应用容器则为具体应用提供了运行框架支撑。此章节对产品的主要功能特性进行阐述。

5.1 微内核

PAS 6.5 产品微内核提供了基础的通讯、资源、事务、日志等服务，其他扩展服务可以通过插拔的方式纳管进来。微内核具有如下特性：

- 支持服务的全生命周期管理；
- 支持服务的依赖注入；
- 支持 JMX 访问；
- 支持服务定义的配置化和持久化；



PAS 6.5 产品微内核默认提供如下服务实现：

- 日志服务：提供基础日志框架，为所有上层服务提供统一的日志配置与持久化能力，日志框架采用 JDK 原生能力，支持国际化、异常处理；
- 控制台服务：提供管理控制的基础框架，以 **ConsoleProvider** 为扩展点，进行各类菜单、页签的扩展机制；
- 配置服务：提供 xml 的配置加载、持久化、变更监听、同步等机制，便于 **server** 的信息管理以及 **sever** 间的信息同步；
- 通讯服务：提供接入接出能力的基础支撑，默认可支持 HTTP、HTTPS、RMI / IIOP 等通信协议；
- 资源服务：提供对 JDBC Resources、JCA Resources、JMS、Concurrent Resources 的管理和使用能力；
- 事务服务：提供全局事务管理能力，支持 **XA** 管理器管理分布式的事务资源，管理目标包括数据库连接、JCA 资源等；

5.2 Web 容器

PAS 6.5 的 Web 容器提供一个 JSP 编译和 Servlet 运行的统一环境，Web 容器为每个 Web 应用提供一个类加载器，当应用产生变更时，可自动重新加载该应用。PAS 6.5 版本支持 Servlet4.0 和 JSP2.3 规范，主要增强的特性包括：

- 支持 Application 全生命周期管理，可监听 ServletContext 对象创建、销毁事件，属性 (attribute) 添加、变更或移除事件，HttpSession 创建、失效、超时事件等；
- 支持 JSP 预编译设置，支持禁用 JSP 重新加载，可大幅提升性能；
- 支持重定向 URL，指定将对旧 URL 的请求视为对新 URL 的请求，为虚拟服务器指定重定向的 URL，请使用 **redirect_n** 属性，其中 **n** 是允许指定多个的正整数，这些 **redirect_n** 属性中的每个属性均由虚拟服务器上部署的所有 Web 应用程序继承；
- 支持虚拟服务器设置，可向虚拟服务器分配 Web 模块，分类设置和管控；
- 支持一个 Servlet 通过不同协议，连接到多个 web 服务器上，增强 Servlet 的复用性；
- 支持配置会话信息的多种存储，包括文件存储、redis 存储，在集群环境中，支持会话信息的内存复制；

5.3 EJB

EJB 组件技术为应用提供了一种方便的组件模型，客户端使用 EJB API 调用，服务端使用 EJB API 实现业务逻辑。PAS 6.5 支持三种类型的 EJB：会话 Bean、实体 Bean 和消息驱动 Bean。会话 Bean 主要用于实现业务逻辑，实体 Bean 代表了数据库的数据抽象，消息驱动 Bean 则是负责处理消息请求的组件：

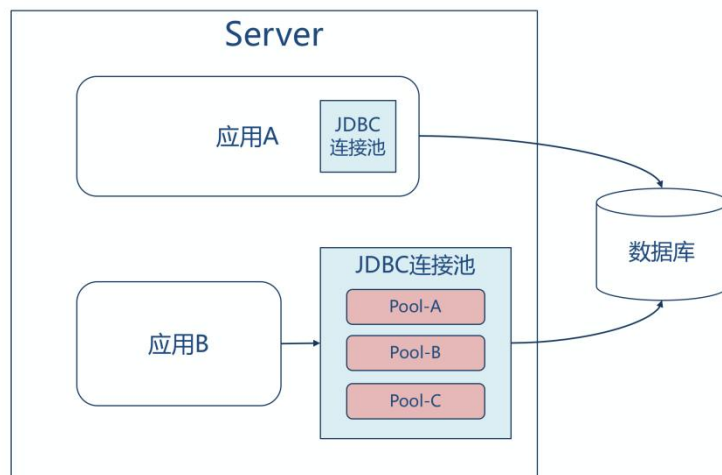
- 会话 Bean(Session Bean)：是客户程序的一部分，之所以叫做会话 Bean，是因为其生命周期和客户的每一次会话相同。Session Bean 一般实现业务逻辑、业务规则和工作流。会话 Bean 分为有状态和无状态两类，有状态 Bean 用于建模需要维护会话状态的任务，EJB 容器会为我们保持会话的状态，比如购物车，也正是因为要保持状态，其开销会比较大；无状态 Bean 是没有特定用户状态的，不保存数据，是线程安全的。
- 实体 Bean(Entity Bean)：与会话 Bean 相比，实体 Bean 一般不用于实现业务逻辑，而是代表了真实存在的数据实体，一般情况下一个实体 Bean 对应着数据库中的一张表，而一个实体类的实例对应着这张表中的一条记录。
- 消息驱动 Bean(MDB)：实体 Bean 和会话 Bean 都是同步的，以阻塞方式进行，调用者在收到返回后才能进行下一步处理，而消息驱动 Bean 则是异步的实现，通常与 JMS 协同使用，可以将其理解为一个异步的无状态 Session Bean，客户端调用 MDB 后无需等待，MDB 将异步处理客户请求，一个 MDB 类实现 MessageListener 接口，当容器检测到 bean 守候的队列一条消息时，就调用 onMessage()方法，将消息作为参数传入。当一个业务执行的时间很长，而执行结果无需实时向用户反馈时，很适合使用消息驱动 Bean，如订单成功后给用户发送短信。

PAS 6.5 符合 EJB3.2 规范，主要增强的特性包括：

- 支持 EJB-JAR 重新部署时，通过在描述符中设置 “<keep-state>true</keep-state>”（也可在线配置），保留 Web 会话, SFSB 实例，以及带持久化能力的 EJB Timer；
- 提供了底层的实例池管理器，实现 EJB 的池化管理，EJB 容器对 EJB 组件进行缓存和池化以提高性能，调整缓存和池属性可以为 EJB 容器带来显著的性能优势。池设置对无状态会话和实体 Bean 有效，而缓存设置对有状态会话和实体 Bean 有效；
- 支持针对容器管理持久性查询方法的查询语言（EJB QL）；
- 支持对有状态会话 Bean 的生命周期回调拦截，支持在一个事务中执行；

5.4 数据库

数据库连接管理提供对分布式交易和数据库连接池的支持，每一个数据源到一个逻辑数据库，逻辑数据库隐藏了不同 JDBC 和数据库实现的细微差别，产品提供数据库连接池管理，降低建立和关闭数据库连接的开销。PAS 6.5 产品中，应用既可以使用自身的连接池管理，也可以使用平台提供的连接池管理，建议采用平台提供的连接池，可支持更多的动态特性管理。



PAS 6.5 版本在数据库管理上主要增强的特性包括:

- 默认支持 Datasource、XADatasource、ConnectionPoolDatasource、Driver 四种资源类型；
- 支持该池在创建或重新配置期间执行 ping 命令以识别并警告其属性的任何错误值；
- 支持对连接池的初始大小、最大值、单步调整数、空闲超时、等待时间的设置；
- 支持事务处理，默认支持 read-committed、read-uncommitted、repeatable-read、serializable 四种事务处理隔离；
- 支持初始 SQL 设置、支持 SQL 跟踪，用户可自定义实现；
- 支持数据库连接设置，包括验证次数、泄露超时、泄露回收、尝试次数、延迟连接等；
- 对编码忘记关闭数据库连接导致连接数据溢出的情况，提供了一定的容错强制处理功能，最大化的保障运行可靠性；
- 支持 Oracle、Derby、Embedded-Derby、JavaDB、DB2、Microsoft SqlServer、Sybase、Pointbase、Cloudscape、Informix、MySQL、Postgresql，达梦、Kingbase 等数据库，产品默认携带内置数据库 derby；

5.5 JCA

企业信息系统（EIS）是指保存组织数据的任何系统。它可以是大型机，消息传递系统，数据库系统或应用程序。JCA 全称 **Jakarta EE Connector Architecture**，使平台能够连接到异构 EIS，解决企业应用集成问题。PAS 6.5 实现了 JCA1.7 规范，主要包括以下几个重要概念：

- 连接器连接池：连接器连接池用于管理连接企业信息系统（EIS）的具体连接，进行连接池配置时需要先部署 **rar** 程序，连接池需选择连接工厂定义，设置连接池各参数以及连接验证信息；
- 连接器资源：连接器资源是一个程序对象，可以为应用程序提供到企业信息系统（EIS）的连接，连接器资源需要定义其 JNDI 名称，EIS 的连接器资源的 JNDI 名称通常在 `java :comp/env/eis` 特定子上下文中；
- 资源适配器：一个资源适配器是一个系统级的 **Driver**，用于应用服务器或应用客户连接到 EIS，资源适配器可针对已部署的连接器进行连接属性的具体值设置；

PAS 6.5 在实际运行时，当应用程序连接到 EIS 时，将进行如下交互过程：

- 通过 JNDI API 进行调用，该应用程序获取与 EIS 关联的连接器资源；
- 使用连接器资源，应用程序将获得 EIS 连接。PAS 6.5 从连接池中检索与 EIS 资源相对应的物理连接；
- 建立 EIS 连接后，应用程序可以读取，修改数据并将数据添加到 EIS。应用程序通过调用 JMS API 来访问 EIS 信息。
- 当应用程序完成对 EIS 的访问后，应用程序将关闭连接并将连接返回到连接池。

PAS 6.5 在交互安全性上做了一定增强，支持将 EIS 主用户或 EIS 用户组提交的调用方身份映射到 PAS 安全域中的适当主用户或用户组。

5.6 JNDI

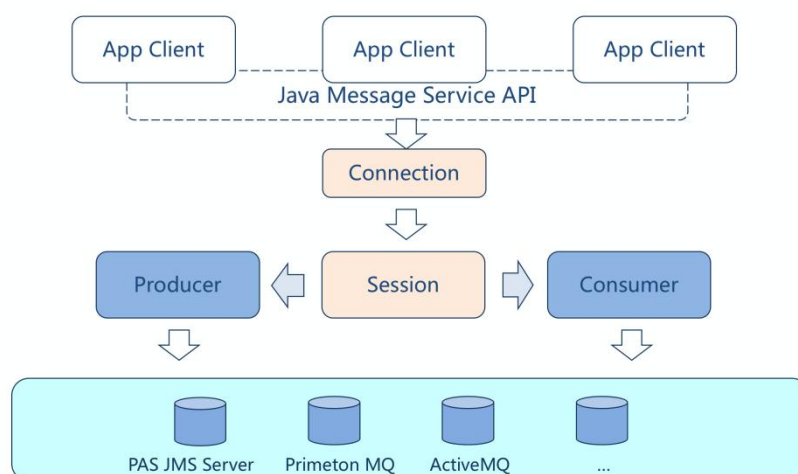
JNDI 是指命名目录服务。使用 JNDI 一般分使用者和注册者，多类服务(如 LDAP、DNS、Mail 等服务)可快速注册到平台的 JNDI 结构中，而 Java 应用程序能够通过 JNDI API 快速访问这些服务。JNDI 通过 Jakarta EE 服务器提供的命名和目录服务绑定到其对象，例如，Derby 数据库的 JNDI 名称为 `jdbc/_default`，在启动时，PAS 6.5 从配置文件中读取信息，

并自动将 JNDI 数据库名称添加到名称空间。

PAS 6.5 提供 JNDI 定制资源和外部资源两种，定制资源信息中，需要定义工厂类及资源类型，外部资源则使得应用程序能够获取对存储在外部资料档案库中的资源的访问权限（如 LDAP）。

5.7 JMS

Java 消息服务（JMS）API 是一种消息传递标准，它允许 Jakarta EE 应用程序和组件（包括消息驱动 Bean（MDB））创建，发送，接收和读取消息。它支持松散耦合，可靠和异步的分布式通信。PAS 6.5 上的应用程序除了使用内置的 JMS Server 支持外，还可以使用三方 JMS Server，具有可移植性和灵活性。



PAS 6.5 上的 JMS 服务配置信息包括：

- **JMS 主机：**JMS 主机是消息服务器，用于承载目标，存储消息并与应用程序交互以通过连接发送和接收消息。在 Message Queue 中，JMS 主机称为代理。JMS 服务支持以下类型的 JMS 主机：
 - 嵌入式类型，其中 JMS 主机在与 Sever 实例相同的 JVM 中运行，其配置和生命周期由 JMS 服务管理；
 - 本地类型，其中 JMS 主机与 Sever 实例在同一主机上单独运行，其配置和生命周期由 JMS 服务管理；
 - 远程类型，其中 JMS 主机表示 JMS 服务外部的 Message Queue 代理或代理群集，使用 Message Queue 管理工具管理其操作；
- **JMS 连接工厂资源：**JMS 连接工厂资源包含应用程序用来连接到 JMS 提供程序的信

息。对于每个 JMS 连接工厂，JMS 服务将自动维护连接器资源和连接器连接池，以支持连接池和故障转移。

- **JMS 目标资源：**JMS 目标资源是指生产的消息的目的地、或消费的消息的来源地。
- **JMS 物理目标：**JMS 物理目标提供了一种创建和管理 JMS 目标的方法，而不是在应用程序需要时动态创建它们。尽管在应用程序运行过程中可动态创建目标位置，但通过管理方式创建的物理目标位置更适合生产环境。

5.8 并行资源

并发资源为应用程序提供并发功能的托管对象，PAS 6.5 中可以配置并发资源，最终提供给应用程序的组件（如 Servlet、EJB 等）使用，创建并发资源时，可以为资源指定唯一的 JNDI 名称。应用程序使用此名称访问资源。PAS 6.5 引入了逻辑 JNDI 的概念，逻辑 JNDI 可将标准 JNDI 映射到特定的资源。

PAS 6.5 中支持 4 种类型的并发资源：

- **上下文服务：**上下文服务用于创建动态代理对象，这些对象可获取容器的上下文，允许应用程序在该上下文中运行，容器的上下文会传播到具体执行任务的线程。
- **托管线程工厂：**应用程序使用托管线程工厂来按需创建托管线程，线程由容器启动和管理，容器的上下文传播到执行任务的线程，支持指定创建线程的优先级；
- **托管执行器服务（ExecutorService）：**应用程序使用托管执行器服务来异步执行提交的任务，在由容器启动和管理的线程上执行任务，容器的上下文传播到执行任务的线程，支持指定设置托管任务执行池的大小、有效期等参数；
- **托管的调度执行器服务（ScheduledExecutorService）：**应用程序使用托管的调度执行器服务在特定时间异步执行提交的任务，在由容器启动和管理的线程上执行任务，容器的上下文传播到执行任务的线程。

5.9 ORB

理解 ORB 前先要理解 CORBA 模型，CORBA 模型基于客户端通过一个明确定义的接口，以远程方法请求的形式向远程对象发出请求。远程方法请求包含有关需要执行的操作的信息，包括服务提供者的对象名称和所调用方法的参数。CORBA 可自动处理任务，例如对象注册，对象位置，对象激活，请求多路分解，错误处理，编组和操作调度。

ORB (**Object Request Broker**) 是 CORBA 的核心组件, ORB 提供了识别和定位对象、处理连接管理、传送数据和请求通信所需的框架结构。CORBA 对象之间从不直接进行通信, 对象通过远程桩对运行在本地计算机上的 ORB 发出请求。本地 ORB 使用 Internet Inter-Orb Protocol (IIOP) 将该请求传递给其他计算机上的 ORB, 然后远程 ORB 定位相应的对象、处理该请求并返回结果。使用 RMI-IIOP, 应用程序或对象可将 IIOP 用作远程方法调用 (RMI) 协议, 比如 EJB 模块的远程客户机通过 RMI-IIOP 与 Application Server 进行通信。

PAS 6.5 产品中, 内置了 ORB 的安全实现, 支持为 ORB 创建多个 IIOP 侦听器, 每个侦听器可单独指定监听端口, 网络地址, 安全属性等。

5.10 邮件

JavaMail, 顾名思义, 提供给开发者处理电子邮件相关的编程接口, 是用来处理 email 的 API, 它可以方便地执行一些常用的邮件传输。在 PAS 6.5 中, 允许应用程序组件通过 Internet 发送电子邮件通知以及从 IMAP 和 POP3 邮件服务器读取电子邮件。

PAS 6.5 产品在创建邮件会话 (mail session) 时, 会使用为服务器端组件和应用程序分配的会话属性, 使其能够通过 JNDI 访问 JavaMail 服务。邮件会话中可以指定邮件主机, 传输和存储协议以及默认邮件用户, 以便使用 JavaMail 的组件不必一一设置这些属性。

PAS 6.5 产品实现了 Java Mail 1.6 规范, 可以指定如下设置:

- JNDI 名称: 邮件会话的唯一名称一般为 JavaMail 资源使用命名上下文前缀 mail/;
- 邮件主机: 默认邮件服务器的主机名, 如果未提供特定于协议的主机属性, 则存储对象和传输对象的 connect 方法将使用此值, 该名称必须可在服务器上解析为实际的主机名;
- 默认用户: 连接到邮件服务器时提供的默认用户名, 如果未提供特定于协议的用户名属性, 则存储对象和传输对象的 connect 方法将使用此值;
- 默认寄信人地址: 默认用户的电子邮件地址, 通常与默认用户相同;
- 存储协议: IMAP 或 POP3;
- 传输协议: SMTP, 支持 SSL 通道;
- 其他高级设置: PAS 6.5 还支持为邮件会话启用附加调试输出, 方便用户进行协议跟

踪等；

5.11 监控

对应用服务器的监控，最终是为了检查系统运行情况、提高性能或定位问题。PAS 6.5 平台自带的监控服务，可以跟踪和显示操作统计信息，如每秒的请求数，平均响应时间和吞吐量。通过监视部署的各种组件和服务的状态，您可以确定性能瓶颈，预测故障，执行根本原因分析并确保一切正常，通过监视收集的数据在性能调整和容量规划中也很有用。

PAS 6.5 监控能力以模块化的方式公开，以便许多客户端模块可以访问和显示监视统计信息。考虑到大量实施监控对性能的影响，产品对监控指标进行了详细拆分和独立配置，按需选择，平台默认支持以下指标：JVM、事务处理服务、连接器服务、Jms 服务、安全性、Web 容器、Restful Web 服务、Java Persistent、Jdbc 连接池、线程池、EJB 容器、ORB、连接器连接池、部署、Http 服务，能够按照不同层级，指标分类进行细粒度监控数据展示，平台可先找到其虚拟主机列表，再分虚拟主机上的请求响应，再将请求详细做更细粒度的分类展示。

Server1 ----virtual-server1 ----request ----virtual-server2 Server2 ...	count200	状态码等于200的响应数
	count2xx	状态码在2xx范围内的响应数
	count302	状态码等于302的响应数
	count3xx	状态码在3xx范围内的响应数
	count400	状态码等于400的响应数
	count401	状态码等于401的响应数
	count403	状态码等于403的响应数
	count404	状态码等于404的响应数
	errorcount	错误计数的累积值，响应代码大于等于400的总数
	hosts	虚拟服务器的主机名
	maxtime	请求的最长响应时间
	processingtime	处理每个请求所用时间的累积值
	requestcount	到目前为止已处理的请求总数
	state	虚拟服务器的状态
		

5.12 批处理

PAS 6.5 为批处理作业的调度和执行提供了批处理运行时环境，批处理作业通常是长期运行的，无需进行用户交互。如计费，统计报告等。批处理应用程序将作业提交到平台，提供有关时间、执行条件、执行步骤的说明，平台按照与应用程序打包在一起 XML 处理步骤，将有关任务的信息存储在作业存储库中。

PAS 6.5 平台上，支持用户查看批处理作业的运行详情，包括执行情况，执行步骤等，平台上的批处理任务编程需遵守 JSR 352，此规范为定义，实现和运行批处理作业提供支持。

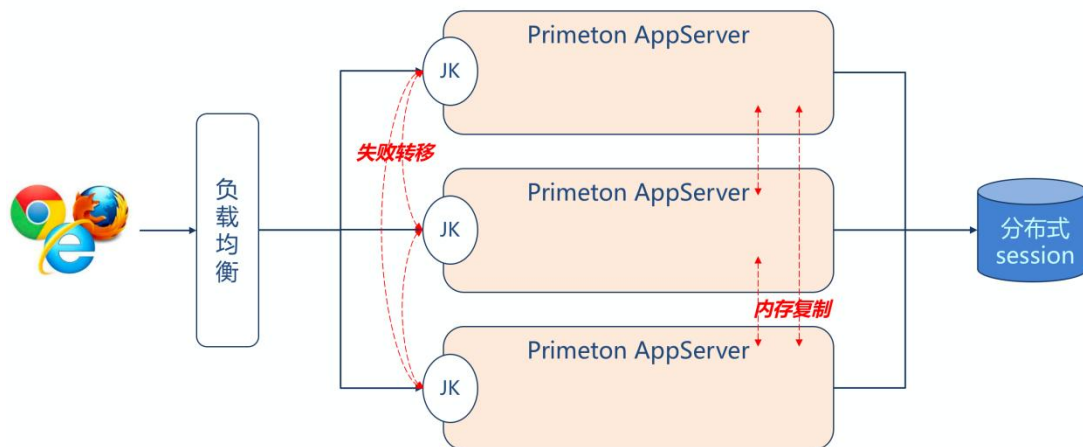
5.13 集群

在三层应用体系结构中，应用服务器位于前置负载分发服务器（apache、nginx、f5 等）的后面，多个服务器可以通过集群技术来支持大规模业务。当访问一个 URL 的时候，前置根据多种策略将请求转发给后端的某个应用服务器实例，从而实现多应用服务器之间的负载均衡。PAS 6.5 中，集群中的各实例可共享相同应用程序，资源和配置信息，

PAS 6.5 的集群实现了故障保护，可伸缩性和负载平衡，进而保障运行程序高可用性：

- 故障保护：如果集群中的实例或主机发生故障，PAS 6.5 会检测到故障并恢复用户会话状态。如果为集群配置了负载均衡器，则负载均衡器会将请求从失败的实例重定向到集群中的其他实例。因为相同的应用程序和资源位于集群中的所有实例上，所以一个实例可以故障转移到集群中的任何其他实例。为了使用户会话状态能够恢复，集群中的每个实例将内存中状态数据发送到另一个实例，由于在任何情况下都更新状态数据，因此将复制数据；
- 可扩展性：如果需要增加容量，则可以在不中断服务的情况下将实例添加到集群。在平台上添加或删除实例后，相关配置等更新将自动处理；
- 负载均衡：如果集群中的实例分布在不同的主机之间，则工作负载可以分布在主机之间，以提高整体系统吞吐量；

在通过上述集群机制保持平台高可靠性的前提下，为了改善高并发请求环境中 Session 持久化造成的性能瓶颈，同时避免大集群情况下 Session 同步带来的网络风暴风险，PAS 6.5 引入了“分布式 Session 存储”的技术，使用 redis 持久化 Session 数据，大大提高了应用服务器在高并发、大集群情形下的性能表现。



PAS 6.5 支持的集群包括 Web 集群、JNDI 集群、EJB 集群和 JMS 集群，集群节点可支持高度异构性。

5.14 应用部署

国内 IT 系统升级更新较快，使得对部署能力要求很高，无论是常规的全量包、增量包部署，还是灰度、蓝绿的发布模式，都与部署能力息息相关。PAS 6.5 提供了单机及集群部署能力，支持一键部署、嵌入式部署、目录级部署、热部署等多种部署方式。PAS 6.5 支持以下类型的模块：

- **Web 模块：**Web 模块，也称为 Web 应用程序，是 Servlet，EJB，HTML 页面，类和其他资源的集合，可以将其打包并部署到多个应用程序服务器。Web 应用程序存档（WAR）文件是组装 Web 应用程序的标准格式；
- **EJB 模块：**EJB 模块是一个可部署的软件单元，由一个或多个企业 Bean 以及 EJB 部署描述符组成。Java 归档（JAR）文件是组装企业 bean 的标准格式。EJB JAR 文件包含 Bean 类（本地，远程，本地和实现），所有实用程序类以及注释或部署描述符。如果 EJB 组件是具有容器管理的持久性（CMP）的 2.1 版或更早版本的实体 Bean，则还可以包括一个 .dbschema 文件和一个 CMP 映射描述符（sun-cmp-mapping.xml）；
- **RAR 模块：**即连接器模块，也称为资源适配器模块，是可部署的软件单元，它为 EJB 组件提供了一种可移植的方式来访问 EIS 数据。连接器模块由用于实现资源模块的所有 Java 接口，类和本机库以及资源部署描述符组成。RAR 是组装连接器模块的标准格式。每个连接器均具有注释或部署描述符文件（ra.xml），PAS 6.5 也支持在程序中使用注解申明 ra.xml 里的信息；

- 应用程序客户端模块。应用程序客户端模块是一种可部署的软件单元，由一个或多个类以及应用程序客户端部署描述符组成，应用程序客户端 JAR 文件适用于客户端，PAS 6.5 支持标准的 Jakarta EE 应用程序客户端规范；
- 生命周期模块：生命周期模块提供了一种基于服务器事件触发任务的方法，服务器触发事件包括初始化、启动、请求准备、关闭等，生命周期模块不是 Jakarta EE 标准模块，是 PAS 6.5 特有的增强能力；
- 其他：用于扩展，比如 OSGi bundle、EBA、CBA 等部署；

PAS 6.5 支持部署的模块化和程序多版本管理，最多可以在服务器实例上启用一个模块或应用程序的一个版本，其他版本则被禁用（当启用一个版本会自动禁用所有其他版本）。从安全性与扩展性考虑，PAS 6.5 对模块或应用程序版本变更进行了以下控制：

- 模块或应用程序不应使用不同的版本标识符进行两次部署，要沿用重新部署模块或应用程序，必须使用 **force** 参数；
- 在部署和取消部署过程中创建或删除的数据库表是全局资源，无法通过应用程序版本进行限定；
- 在重新部署新版本期间将保留 Web 会话；
- 旧版本的资源不会自动引用模块或应用程序的较新版本。因此，您必须为模块或应用程序的较新版本显式创建资源；
- OSGi 已经拥有自己的版本控制系统，因此部署 OSGi bundle 时，平台会忽略该名称提供的任何版本信息。

5.15 安全

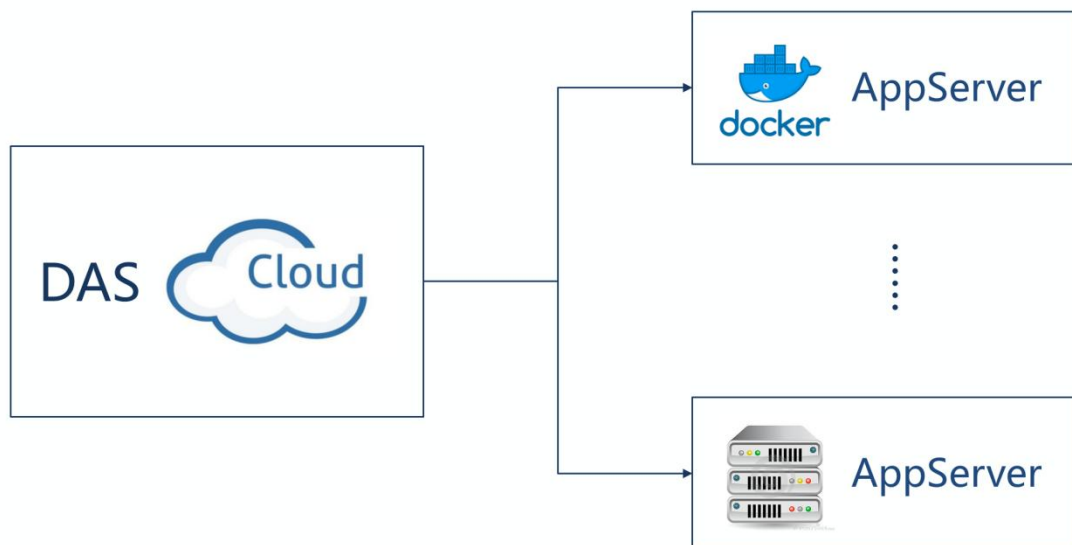
Jakarta EE 安全性的标准主要是用于如何防止未经授权访问、如何保障传输完整等方面。PAS 6.5 基于 Java 安全模型构建，使用沙箱承载应用程序，保障应用程序本身不会给系统或用户带来潜在风险。PAS 6.5 实现了用于身份验证和身份存储的可移植插件接口，以及新的可注入类型 **SecurityContext** 接口，该接口提供了程序安全性访问。平台主要从以下方面进行增强：

- 认证方式：使用安全凭证进行身份验证，可以是用户名和密码，数字证书或其他信息。通常服务器或应用程序会对客户端进行身份验证，也有客户端反向来认证服务器。PAS 6.5 支持双向认证。

- 授权：也称为访问控制，是指授予用户访问数据或执行操作的权限。用户通过身份验证后，用户的授权级别将决定其可以执行的操作，在 **PAS 6.5** 中，用户的授权基于角色进行的，而角色定义了可以访问哪些应用程序以及每个应用程序的哪些部分，以及这些用户或组可以对应用程序执行什么操作，平台默认支持三员管理。
- 审计：审计是指通过捕获与安全相关的事件，目的是评估安全措施的有效性。**PAS 6.5** 使用审计模块来捕获所有身份验证和授权操作等记录，除了提供了默认审核模块，还支持插入自定义审核模块，审核模块的作用范围是整个服务器，意味着服务器上的所有应用程序使用同一审核模块。
- 管理 JACC 提供商：Java 容器授权协议（Java Authorization Contract for Containers，简称 JACC）可将基于 Java 权限的安全模型扩展到 EJB 和 Servlet，JACC 提供者使用 Java 验证和授权服务（JAAS）API。**PAS 6.5** 默认提供两个 JACC 实现，分别名为 default 和 simple，也支持用户自行扩展。
- 此外，平台在邮件安全、密码安全、LDAP 等方面也进行了扩展。

5.16 域管理

PAS 6.5 以域为一级管理概念，域包含一组需一起管理的应用服务器实例，每个域都有一个用于管理应用程序的域管理服务器（DAS）。不仅仅在传统基础设施上，产品还提供了在云设施上的管理兼容性。



域管理控制台的能力主要包括：

- 服务器（Server）管理：管理集群或独立服务器实例，查看服务器运行状态、JVM 内

存、类加载、线程、HTTP 端口、IIOP 端口、内部资源、批处理等信息；

- 节点管理：管理实际物理资源节点，可通过 SSH 远程管理；
- 应用管理：发布各类型的应用，对应用进行启停、重新加载等；
- 资源管理：管理并行资源、连接器、jdbc、jms 等资源，此类资源可供域内所有 server 使用；
- 配置管理：支持集群或独立服务器的配置管理，可共享。配置信息包括 Http 服务、线程池、虚拟主机、Web 容器、EJB 容器等；
- License 管理，license 对域的可管理范围进行控制，包括 DAS 服务有效期、服务器数量、应用数量等；

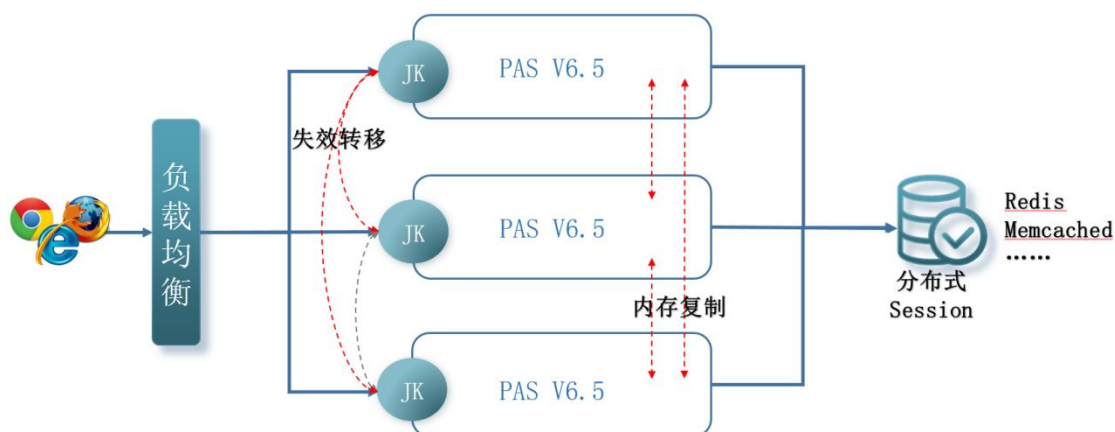
5.17 应用迁移工具

PAS 产品提供多种应用迁移工具，方便用户快速将 Tomcat、WebLogic、WebSphere 上应用迁移到 PAS 平台中。

- 源码扫描工具：基于业务系统源码及配置文件的兼容验证工具，查找非 JakartaEE 标准的实现、接口及配置，方便用户进行信创迁移。
- 资源冲突查找工具：针对应用依赖的资源进行冲突检测，生成多种格式报告，排查冲突问题，减少业务部署、运行错误。
- 配置快速迁移工具：通过读取 Tomcat、WebLogic、WebSphere 中间件配置文件，将应用系统需要的数据源、连接池、线程池等配置进行解析，生成 PAS 服务器配置项，助力应用快速迁移部署设置。

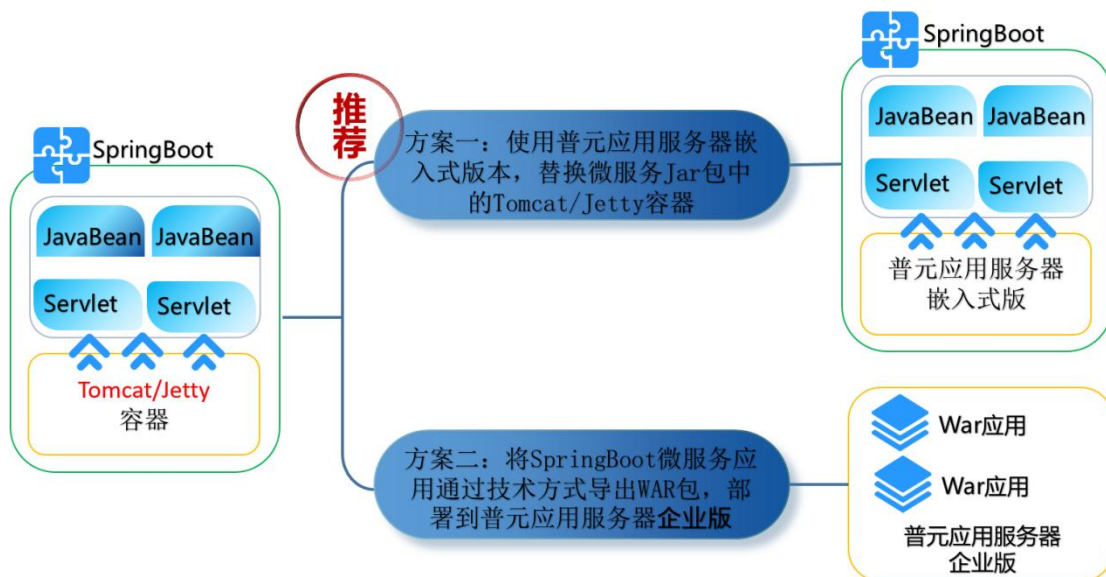
5.18 分布式缓存

业务系统采用集群机制保持平台高可靠性的运行。为了改善高并发请求环境中 Session 持久化造成的性能瓶颈，同时避免大集群情况下 Session 同步带来的网络风暴风险，PAS 产品引入了“分布式 Session 存储”的技术，使用 Redis/Memcached 持久化 Session 数据，大大提高了应用服务器在高并发、大集群情形下的性能表现。



5.19 微服务架构支撑能力

当下流行的服务架构、云架构主要采用 springcloud 微服务架构进行开发。Spring Boot 使用 Tomcat/Jetty 作为内置容器支撑 JavaBean、Servlet 等运行，以此来对外提供服务。普通 PAS 产品提供微服务嵌入式容器，用来替代微服务中之使用的 Tomcat/Jetty 容器。



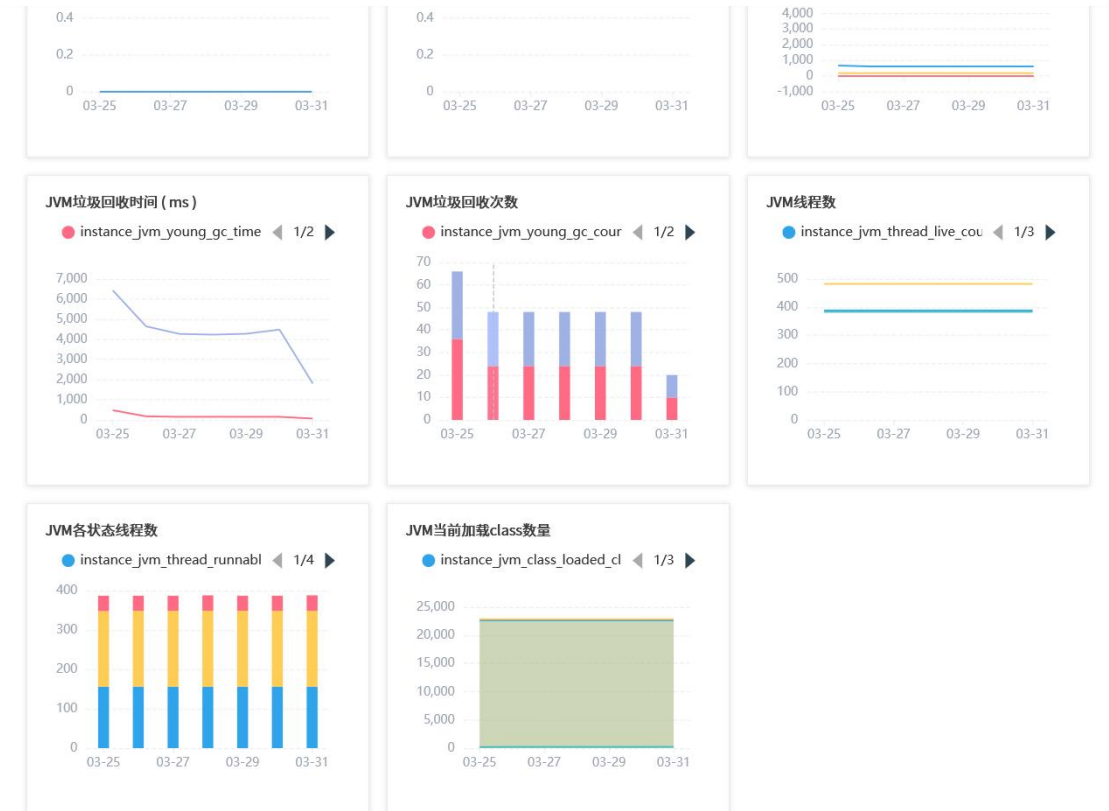
5.20 应用性能监控

普元 PAS 产品可以通过 APM（Application Performance Management 应用性能监控）组件对 PAS 服务器、部署在 PAS 上运行的业务应用、业务系统使用的数据库、分布式缓存工具进行监控。通过 APM 获取精准的监控信息，帮助用户迅速定位应用的性能瓶颈，协助开发/运维人员对服务器，业务系统及相关环境进行优化。

APM 组件，提供无侵入性，细粒度的监控，可以从各个维度进行监控、服务统计、性能创

析。

在 PAS 服务器监控中，用户可以了解服务器整体每分钟请求数、吞吐量、JVM 情况等信息。



针对业务服务除了能对应用请求时长、成功率，每分钟请求数等进行监控统计，还能深入监控应用的请求链路进行详细监控，排查业务服务处理过程中的性能卡点。

GET:/samples/servlet/bookServlet
ed193306046d4018bda8f93f0ad477ce.200.16485422418910001
开始时间 2022-03-29 16:24:01 持续时间 2172 ms 跨度 29

方法名	开始时间	总执行时间(ms)	执行成功率(%)	执行时间(ms)	所属方法类型	所属服务
GET:/samples/servlet/bookServlet	2022-03-29 16:24:01	2172		267	AppServer	pas-162:server/samples
Log4jServletFilter.doFilter	2022-03-29 16:24:02	1905		4	Filter	pas-162:server/samples
SetCharacterEncodingFilter.doFilter	2022-03-29 16:24:02	1901		3	Filter	pas-162:server/samples
DM/JDBC/Statement/executeUpdate	2022-03-29 16:24:02	9		9	dm-jdbc-driver	pas-162:server/samples
DM/JDBC/Statement/close	2022-03-29 16:24:02	0		0	dm-jdbc-driver	pas-162:server/samples
DM/JDBC/Statement/executeQuery	2022-03-29 16:24:02	18		18	dm-jdbc-driver	pas-162:server/samples
DM/JDBC/Statement/close	2022-03-29 16:24:02	0		0	dm-jdbc-driver	pas-162:server/samples
Log4jServletFilter.doFilter	2022-03-29 16:24:02	1178		1178	Filter	pas-162:server/samples

针对目前广泛使用的国产信创服务器、分布式存储工具，APM 也提供监控能力，可以方便的查找出响应时间长、成功率低的数据库操作。

5.21 应用支撑

普元应用服务器在做好作为应用载体的本质工作前提下，还从应用扩展角度出发，考虑为应

用提供共通能力：

➤ 序列号生成器

在应用中需要标识记录唯一性的序列号不重复，用户需要在应用中自己实现序列号生成器，如果考虑到分布式场景下，在生成器中还需要通过雪花算法实现，给应用开发带来一定的工作量，普元 PAS 部署后自动了基于雪花算法的序列号生成器，客户部署在上面的应用可以直接调用接口获取序列号。

➤ 应用防重

很多客户应用部署到 web 服务器中，都没有考虑前端请求重复提交了，如何处理，在运行过程中很多都会重复处理，造成数据的不一致，影响正常业务。

普元应用服务器默认自带了防重处理能力，部署在上面的应用，如果前端有重复提交能够自动拦截处理。

➤ 全局日志流水号

在应用部署到 PAS 后，用户可以选择是否需要启用全局流水号，如果开启此功能，则所有访问应用的请求会在请求头中加入全局日志流水号，并且在平台的日志记录中会一起记录流水号，同时在应用开发中也可以读取此流水号进行日志记录。

6 PAS 产品环境配置

6.1 支持的 CPU

- 飞腾 CPU：FT-1500A+ 、FT-2000+、S2500
- 鲲鹏 CPU：920A
- 龙芯 CPU：3A3000/3B3000、3A4000/3B4000、3A5000/3B5000
- 海光 CPU：3000 系列、5000 系列、7000 系列
- 兆芯 CPU：开胜® KH-30000 系列、开胜® KH-20000 系列、开胜® ZX-C+系列
- 申威 CPU：申威 SW

6.2 支持的操作系统

- 银河麒麟高级服务器操作系统 V10
- 统信服务器操作系统 V20
- 银河麒麟服务器操作系统 V4

- 中标麒麟高级服务器操作系统软件 V7.0
- Windows 10
- macOS High Sierra+
- CentOS 7.2+

6.3 支持的数据库

- 达梦数据库 7.6、8.0
- 南大通用
- 人大金仓
- 神通数据库
- 优炫数据库
- 瀚高数据库
- 华为高斯数据库
- 海量数据库
- MySQL 5.7+
- Derby 10.14.2
- Oracle 10、11
- PostgreSQL 9.0
- OceanBase
- Informix
- SQL Server
- DB2

6.4 支持的浏览器

- 360 安全浏览器
- 奇安信浏览器
- Firefox 74.0
- Chrome 80.0

6.5 支持的 JDK 版本

- SUN JDK 1.8.0_191+
- OpenJDK 1.8.0_191+
- 毕昇 JDK 8(bishengjdk-8)
- 腾讯 Kona8
- Alibaba Dragonwell8

7 客户情况

PAS 已成功应用于政府、金融、教育、军工、能源、交通、制造、运营商、医疗等行业。

政府机关	卫计委、中国科协、浙江省政府、江苏省政府、湖南省政府、上海市政府、重庆市政府、海南省政府、长三角示范区、贵州省政府、贵阳市政府、新疆维吾尔自治区工信厅、新疆维吾尔自治区政协、新疆维吾尔自治区商务厅、河南省交通运输厅、广西交通运输厅、安徽省住房和城乡建设厅、安徽省药品监督管理局、安徽合肥市党政机关、安徽省淮南市数据资源管理局、安徽省林业局、江苏省基础地理信息中心、淮安市自然资源与规划局、海南省生态环境厅、海口市交通运输和港航管理局、湖南省机关事务管理局、江苏省测绘局、上海高院、上海 GA、上海市浦东新区党政机关、上海市普陀区政府、上海普陀区大数据中心、上海市浦东新区档案局、上海市监狱管理局、上海市城市交通运输管理处、上海市卫生人才交流服务中心、上海市青浦区科学技术委员会、上海市浦东新区废弃物管理中心、苏州工业园区、南通信创产业园、渝北档案馆
银行保险	新疆农信、长安银行、南粤银行、天津银行、陕西农信、广东农信、陕西信合、柳州银行、上海银行、交通银行、兴业银行、九江银行、贵州农信、建信金融、中国投资有限责任公司、中国出口信用保险公司、中央国债登记结算有限责任公司
教育研究院	中国科学院、中国科协、苏州大学、中共襄阳市委党校、中共六安市委党校、中共大同市委党校、苏州职大、桂林市职业学校、常州工程职业技术学院、北京劳动保障职业学院
央企	中国铁建、中车集团、中核集团、安徽中烟、首钢集团、东方电气集团、北京化工集团、中国北方车辆研究所、中国核动力研究设计院、GA 部第三研究所、北方信息控制研究院集团、南京南瑞继保电气有限公司、中国钢研科技集团有限公司
军工	中船重工、航天三江、中国航发中传机械
能源	神华神东煤炭集团、国家开发投资集团
交通	山东高速集团、中国铁路设计集团

运营商	中国移动（深圳）
医疗	威高集团、江苏万邦生化医药集团
制造业	青岛海信、索菲亚家居

8 联系我们

了解关于普元相关产品的更多内容，请致电免费电话：400-820-5821。

或联络普元各地办公室

普元信息技术股份有限公司(公司总部)

上海市张江高科技园区碧波路 456 号 4 楼 邮编： 201203

Tel: +86 021-5080 5188

Fax: +86 021-5080 1900

普元北京办公室

北京市海淀区彩和坊路 8 号天创科技大厦东门 1301 室 邮编： 100080

Tel: +86 010-6269 8005

Fax: +86 010-6269 8015

普元广州办公室

广州市天河区体育西路 191 号中石化大厦 B 塔 1712 室 邮编： 510630

Tel: +86 020-3821 8599, 3821 8499, 3821 8455

Fax: +86 020-3821 7272

普元西安办公室

陕西省西安市高新区唐延路 25 号银河新坐标 A 座 1404 室 邮编： 710075

Tel: +86 029-8882 3459

Fax: +86 029-8889 7491

普元成都办公室

普元信息技术股份有限公司

成都市高新区天益街 38 号理想中心 1 栋 605 室 邮编：610041

Tel: +86 028-8519 0741, 8519 0743

Fax: +86 028-8519 0745

普元武汉办公室

湖北省武汉市江汉区马场路 菱角湖万达广场 A3 写字楼 2008 号 邮编：430015

Tel: +86 027-8561 9935

Fax: +86 027-8561 9937

9 关于普元

普元信息技术股份有限公司是专业的软件基础平台产品与解决方案提供商，股票代码：**688118**。公司主要面向金融、电信、政务等行业的大中型企事业单位，为其提供创新可靠的软件基础平台产品及相关技术服务。公司软件基础平台产品和技术服务涵盖云应用平台软件、大数据中台软件和 **SOA** 集成平台软件三大技术领域，目前拥有国家级企业技术中心、博士后科研工作站，是国家规划布局内重点软件企业。

普元长期致力于软件基础平台核心技术自主研发与创新能力的构建，打造了基于组件化的技术平台和核心技术组合，形成了体系化的技术研发能力、平台化的产品开发能力及完善的项目实施方法论。目前公司取得授权发明专利 **27** 项，软件著作权 **172** 项，参与 **11** 项相关领域国家标准制定，产品和技术方案的技术性能、成熟度及安全可靠性已在金融、政务、能源、电信、制造业等国民经济重点领域的上千个大型、关键项目中得到验证。

普元信息在国产软件生态领域不断深化合作并取得了诸多突破性合作成果，未来将继续发挥自主创新技术优势，为国产生态环境建设贡献力量，并努力发展成为具有国际竞争力的基础软件厂商。

了解普元，请访问 www.primeton.com