

# SOFAStack

服务网格  
产品简介

产品版本：AntStack Plus 1.11.0


文档版本：20221009

# 法律声明

蚂蚁集团版权所有©2022，并保留一切权利。

未经蚂蚁集团事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

## 商标声明

 蚂蚁集团  
ANT GROUP 及其他蚂蚁集团相关的商标均为蚂蚁集团所有。本文档涉及的第三方的注册商标，依法由权利人所有。

## 免责声明

由于产品版本升级、调整或其他原因，本文档内容有可能变更。蚂蚁集团保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在蚂蚁集团授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过蚂蚁集团授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

# 通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
<b>粗体</b>	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击 <b>确定</b> 。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[ ] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

# 目录

1.什么是服务网格	05
2.产品优势	07
3.产品架构	08
4.应用场景	10
5.使用限制	11

# 1.什么是服务网格

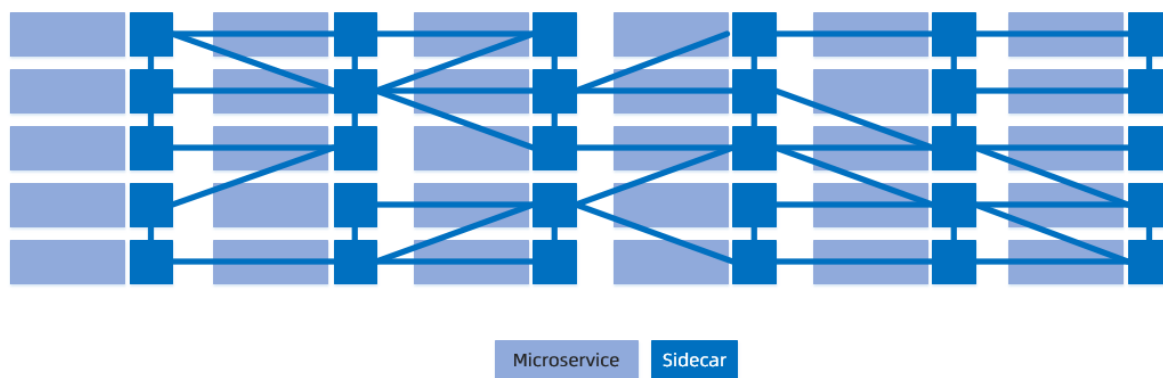
服务网格（SOFAStack Mesh）是蚂蚁集团自主研发的基于金融级生产实践的增强版服务网格平台，将传统微服务和 Service Mesh 技术进行了深度融合，其核心技术经过了蚂蚁集团的大规模生产实践验证。它深度、无缝对接了 SOFAStack 经典应用服务和容器应用服务，为客户提供了简单易用的 Service Mesh 架构的支撑平台。

## SOFAStack 服务网格概述

SOFAStack 服务网格（简称服务网格）= 传统微服务 + Service Mesh。

服务网格旨在提供与平台无关、语言无关、轻量无侵入的云原生分布式架构解决方案，在兼容 Kubernetes 和 Istio 生态的基础上还能支持虚拟机场景，同时支持 SOFA、Dubbo、SpringCloud 微服务框架的服务治理能力。在整个接入过程中，无需修改业务代码，帮助企业以最小的成本完成云原生落地，加速数字化转型。

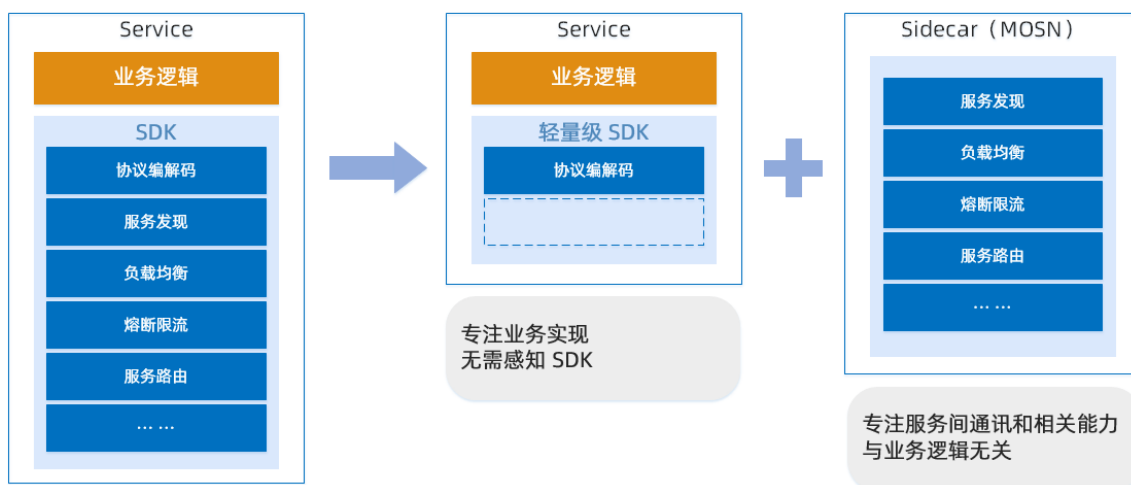
## Service Mesh 图解



从一个全局视角来看，Sidecar 连接成网，组成 Service Mesh。

TCP 协议催生了分布式系统，分布式系统催生了微服务，Service Mesh 就是下一代微服务技术的代名词，是微服务时代的 TCP 协议。Service Mesh 以 Sidecar 形式，将服务治理从业务逻辑中剥离，并拆解为独立进程，实现异构系统的统一治理和网络安全。

## Sidecar 模式图解



Service Mesh 将 SDK 拆解为单独进程，并以 Sidecar 模式进行部署，使得业务进程专注于业务逻辑。

Sidecar 是一个轻量级的网络代理，它们与应用程序部署在一起，对所有流入与流出的网络请求进行拦截，实现各种网络策略，例如：服务发现与负载均衡、流量拆分、故障注入、熔断器以及分阶段发布等功能。

## 功能一览

服务网格包含以下子产品和功能，可以进行各种服务管理和监控操作。

- 服务管控

支持异构服务的统一注册与服务详情查看，包括其基本信息、服务提供者以及消费者信息等。

- 服务治理

支持服务限流、服务路由、服务熔断降级、服务鉴权，提供了精细化调配的服务路由与服务限流功能，保证应用高可用，保障业务持续运行。

- Sidecar 管理

支持查看及管理当前工作空间中的 Sidecar 实例，提供了 Sidecar 状态、Pod 状态、Sidecar 注入时间等信息。

## 2. 产品优势

在传统单体式架构向微服务架构迁移的过程中，随着应用微服务数量的增加，微服务间的通信、监控以及安全性的管理成为新的挑战。服务网格作为应用与基础设施的桥梁，突破传统的 SDK 接入方式，以应用透明的方式处理服务之间、服务与基础设施间的通信，实现应用研发和基础设施最大程度地解耦，让应用研发只需专注于业务开发。

业界最大规模落地实践	客户案例多	生产级优化
<ul style="list-style-type: none"><li>蚂蚁内部1500+应用，40万+容器的规模</li><li>业界最大规模的 Mesh 生产集群</li><li>支持内部聚石塔、IoT 等业务场景</li></ul>	<ul style="list-style-type: none"><li>工商银行、江西农信、中信银行、华夏银行、浙商证券、华瑞银行、网商银行</li></ul>	<p>解决开源产品无法生产落地的问题</p> <ul style="list-style-type: none"><li>Mixer 性能问题</li><li>突破 k8s 服务注册模型</li><li>服务注册增量下发机制</li><li>解决 xds 全量下发引起的性能问题</li></ul>
产品特性完善	产品矩阵丰富	开源开放
<ul style="list-style-type: none"><li><b>多协议</b>：适配业界通用同步、异步通信协议，具备可扩展性和协议转换能力</li><li><b>多平台</b>：容器、虚拟机</li><li><b>多注册中心</b>：适配业界通用的注册中心</li><li><b>安全</b>：mtls、国密、rbac 等安全能力</li></ul>	<ul style="list-style-type: none"><li>RPC能力下沉的 Service Mesh</li><li>异步通讯的消息 Mesh</li><li>数据代理能力的 DB Mesh</li><li>Mesh 化网关</li></ul>	<ul style="list-style-type: none"><li>中国最大的社区 Service Mesher</li><li>底层技术开源开放</li><li>先驱布道者和引领者</li></ul>

### 技术业务分离

在之前，微服务体系都是由中间件团队提供一个 SDK 给业务应用使用，在 SDK 中会集成各种服务治理的能力，如：服务发现、负载均衡、熔断限流、服务路由等。有了服务网格之后，我们就可以把 SDK 中的大部分能力从应用中剥离出来，拆解为独立进程，以 Sidecar 的模式部署。通过将服务治理能力下沉到基础设施，可以让业务更加专注于业务逻辑，中间件团队则更加专注于各种通用能力，真正实现独立演进，透明升级，提升整体效率。

### 异构服务治理

既能借助蚂蚁集团久经考验的微服务框架 SOFA 在云上构建微服务应用，也可以支持原生 Dubbo 和 Spring Cloud 上云，无需构建 ZooKeeper、Eureka、Consul 等微服务依赖的自建服务，极大降低运维成本。

### 轻量无侵入接入

业务应用系统通过 Service Mesh 技术架构轻量级接入，实现对应用无侵入的服务注册与服务治理方案，减少改造成本。

### 跨平台支持

该方案支持容器平台、虚拟机平台，能够满足企业用户未容器化的场景对 Service Mesh 架构转型的需求。

### 网络安全

通过网络安全，可以更方便地实现应用的身份标识和访问控制，辅之以数据加密，就能实现全链路可信，从而使得服务可以运行于零信任网络中，提升整体安全水位。

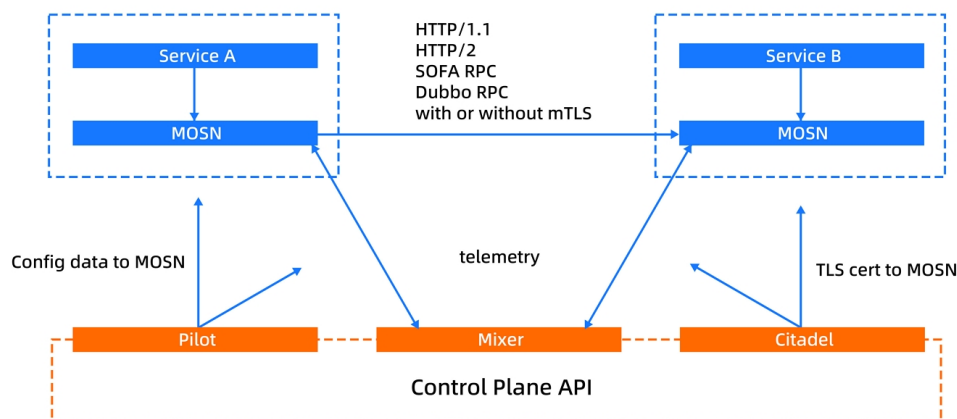
### 简单易用易维护

SOFAStack 服务网格提供集中式图形化易操作的管理平台，支持多租户管理能力，满足企业级高级特性需求，简化分布式应用的服务管理、服务治理、可观察性等能力，让用户便捷的对应用服务统一管理和治理。

## 3. 产品架构

服务网格的产品架构逻辑上分为控制层面和数据层面。

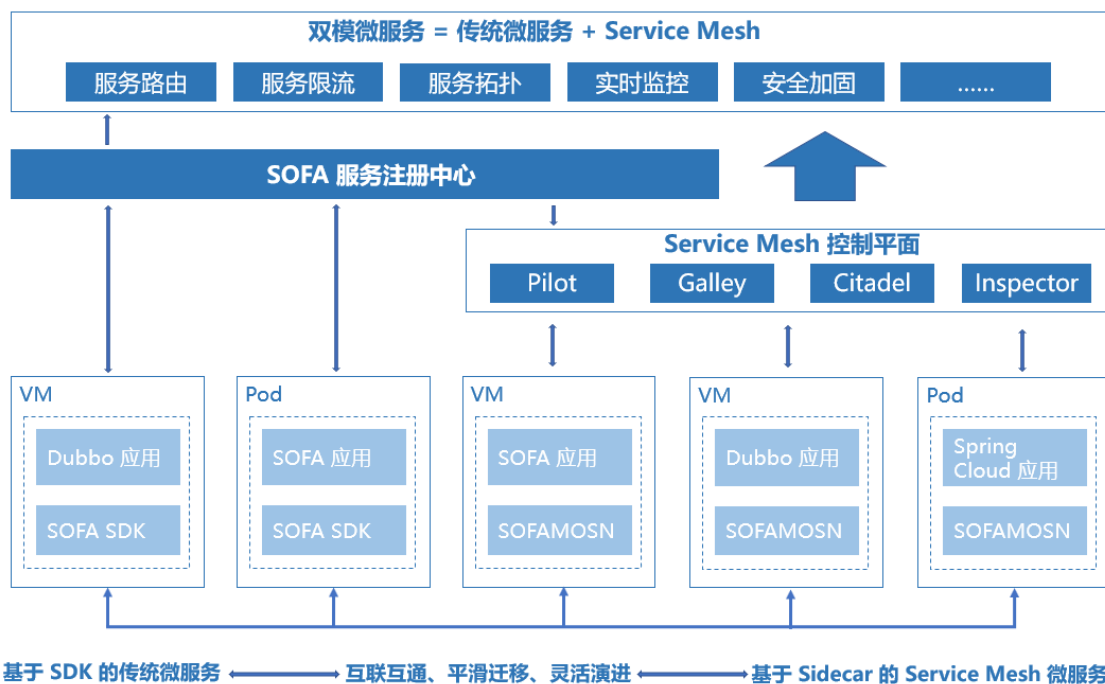
- 控制层面：管理代理（蚂蚁自研 MOSN），用于管理流量路由、运行时策略执行等。
- 数据层面：由一系列代理（蚂蚁自研 MOSN）组成，用于管理和控制服务之间的网络通信。



### 系统架构

SOFAStack 服务网格结合了 SOFAStack 微服务的优势，将 SOFAStack 微服务和 Service Mesh 双剑合璧，即基于 SDK 的传统微服务可以和基于 Sidecar 的 Service Mesh 微服务实现下列目标：

- 互联互通：两个体系中的应用可以相互访问。
- 平滑迁移：应用可以在两个体系中迁移，对于调用该应用的其他应用，做到透明无感知。
- 灵活演进：在互联互通和平滑迁移实现之后，我们就可以根据实际情况进行灵活的应用改造和架构演进。



在控制面上，引入了 Pilot 实现配置的下发（如服务路由规则），在服务发现上保留了独立的 SOFA 服务注册中心。



在数据面上，使用了自研的 SOFA MOSN，不仅支持 SOFA 应用，同时也支持 Dubbo 和 Spring Cloud 应用。

在部署模式上，我们不仅支持容器/k8s，同时也支持虚拟机场景。

## 4. 应用场景

依据服务网格特点和优势，服务网格适用于以下应用场景。

### 服务治理与业务逻辑解耦

在运行时，SDK 和业务应用的代码是混合在一个进程中运行的，耦合度非常高，会产生如下问题：

- 升级成本高：每次升级都需要业务应用修改 SDK 版本号，重新发布。在业务飞速发展时，更倾向于专注业务，不希望分散精力做升级等事情。
- 版本碎片化严重：由于升级成本高，但中间件还是会向前发展，久而久之，就会导致线上 SDK 版本各不统一、能力参差不齐，造成很难统一治理。
- 中间件演进困难：由于版本碎片化严重，导致中间件向前演进过程中需要兼容各种老版本的逻辑，无法实现快速迭代。

使用 Service Mesh 后，您可以将 SDK 中的大部分能力从应用中剥离拆解为独立进程，以 Sidecar 的模式部署。通过将服务治理能力下沉到基础设施，可以让业务更加专注于业务逻辑，中间件团队则更加专注于各种通用能力，真正实现独立演进、透明升级，提升整体效率。

### 多语言、多协议支持

随着新技术的发展和人员更替，在同一家公司中往往会出现使用各种不同语言、不同框架的应用和服务，为了能够统一管控这些服务，以往的做法是为每种语言、每种框架都重新开发一套完整的 SDK，维护成本非常高，而且对中间件团队的人员结构也带来了很大的挑战。

有了服务网格之后，通过将主体的服务治理能力下沉到基础设施，多语言的支持就轻松很多了，只需要提供一个非常轻量的 SDK，甚至很多情况都不需要一个单独的 SDK，就可以方便地实现多语言、多协议的统一流量管控、监控等治理需求。

### 云原生架构转型助力

从单体应用到微服务架构改造以及全面容器化的云原生架构基础往往带来很高的改造成本。SOFAStack 服务网格可以满足未容器化的虚拟机部署方案，也可以兼容过渡阶段的部分容器化和虚拟化混合部署的场景，加速企业云原生架构转型。

### 金融场景网络安全

当前很多公司的微服务体系建设都建立在内网可信的假设之上，然而这个原则在当前大规模上云的背景下可能显得有点不合时宜，尤其是涉及到一些金融场景的时候。

通过服务网格可以更方便地实现应用的身份标识和访问控制，辅之以数据加密，就能实现全链路可信，从而使得服务可以运行于零信任网络中，提升整体安全水位。

## 5.使用限制

服务网格在使用时有一些限制，比如 JDK 版本、微服务类型等。

详细的限制信息如下：

限制项	限制范围	限制说明
微服务开发框架	Dubbo、SpringCloud、SOFA	服务注册中心 SDK 仅支持这三种框架的兼容。
SOFA SDK 语言限制	Java	SOFA SDK 仅支持 Java 语言开发。
JDK 版本	JDK 1.8 及以上	仅支持 JDK 1.8 及以上的版本。
虚拟机操作系统版本（虚拟机部署）	虚拟机操作系统版本（虚拟机部署）	虚拟机操作系统版本（虚拟机部署）。