

# Mocha BSM 技术架构

优秀的技术架构是产品的灵魂和核心，也是产品具有良好的可扩展性的基础，摩卡软件重视架构设计，邀请了一批具有国际设计水平的架构师，他们曾在美国硅谷等任职，具有丰富的产品架构设计经验。

## 产品架构概述

Mocha BSM 的技术架构拥有以下几个特性：

- n 基于 **Java 2 Enterprise Edition (J2EE)** 开发，符合现在受欢迎的主流技术。**Mocha BSM** 可以在不同的 **Java** 容器上运行，包括 **Tomcat**，**Weblogic** 或者 **WebSphere**。
- n 由于 **Java** 能轻易支持跨平台，**Mocha BSM** 可以在很多不同平台运行，这包括：
  - n Windows 2003 Server
  - n AIX 5.2、5.3
  - n Solaris 9、10
  - n Red Hat Enterprise Linux AS 4、CentOS 4
  - n 采用标准 **Java Database Connectivity (JDBC)**，所以支持广大的数据库：
    - γ MySQL 5
    - γ Oracle 9i、10g
    - γ MS SQL 2005
- n 对标准协议的支持：
  - n HTTP、HTTPS
  - n JMX
  - n SNMP
  - n JDBC
  - n Telnet、SSH、WMI 架构设计原则



这“3P”即是

### 1.1 Personal 人性化

产品的界面设计必须人性化，达到“傻瓜式”用法，用户可以不经过培训就能会用。在这方面我们花了很多心思与精力，关键界面的设计都历经了数月才能最终定稿，并在后续版本不停地改进。我们在人性化方面具有以下特点：

- n 提供可视化管理，“化繁为简”，使 **IT 管理员快速掌握主机和应用的管理工作**



- n ‘一键式’基于经验值的策略配置

摩卡产品的设计初衷都是绕着 **3P** 的原则。



- n 自动关联事件，避免海底捞针式的查找
- n 图与表结合，清晰的指标展现
- n 使用上表（指标）下图方式



## 1.2 Product 产品化

一个好的 IT 运维管理产品必须满足以下条件：

- n 产品架构可扩展，能支持更多的资源
- n 产品架构健壮，不轻易宕机
- n 产品可配置性强，不需要开发
- n 产品模块化，模块之间存在松耦合的关系
- n 产品重视安全
- n 产品容易安装、部署

除了满足了以上对产品严格的要求以外，Mocha

BSM 特别重视以下的‘三高’



- n 高可扩展

Mocha BSM 产品的初衷设计就是支持几千台至

上万台的被管理资源。为了达到高可扩展的目的，我们从几方面入手：

- 可扩展模块

当一个 Mocha BSM Server 在数量上遇到瓶颈后，能够通过增加一个 Distributed Monitor Server (DMS) 来突破数量和网络方面的限制。

- Portal 响应时间

当监控数量大量增加以后，必须处理和展现的数据也大量增加，对 Portal 的压力也更加大。所以我们 Portal 的应用代码都必须经过**压力测试**，在数千个监控资源，我们给 Portal 定的响应时间是**5-8 秒**。

- 数据采集与处理

当监控数量大量增加以后，需要采集的数据也大量增加。往往在这种情况下，会出现以下情况：

- γ 图表展现断点，数据采集不上来
- γ 数据延迟，该在某时间点采集的数据被延迟了
- γ 由于采集的数据量很大，导致内存溢出

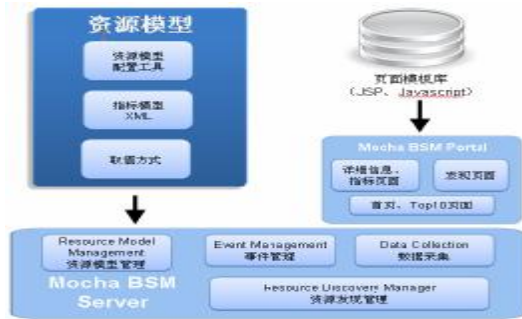
在这方面，我们在架构设计以及优化上做了很多工作，包括以下：

- γ 把数据采集的引擎分为数个，而不是单靠一个，所以一旦发生某些特殊问题，一个数据引擎的宕机不会影响到其他引擎，导致服务器不可用。
- γ 代码进行大量优化，提高采集数据的性能。
- γ 进行数个月的持久性测试，确保产品即使在大的采集数据底下，也能无误运行数月

- n 高可配置

监控软件经常需要对新的或者新版本的被监控应用提供支持，所以支持的响应速度很重要。产品的设计初衷是通过配置来达到快速开发的要求(新应用支持、二次开发)：

- 通过 XML 配置来描述需要采集的**性能、可用性、配置指标**
- 配置取值方式
- 配置需要的页面和 Javascript
- 自动生成页面和部署资源模型



以前 3 个月的新应用支持开发变成今天的 3 天。

n 高度模块化

产品分为 4 层， 每一个层次的模块都是分开，而且之间的关系都是松耦合。层次与层次之间的沟通都是通过 API 接口。每一个不同的层次也分为数个关键模块，模块与模块之间都是松耦合的关系，修改一个模块不会影响其他模块。

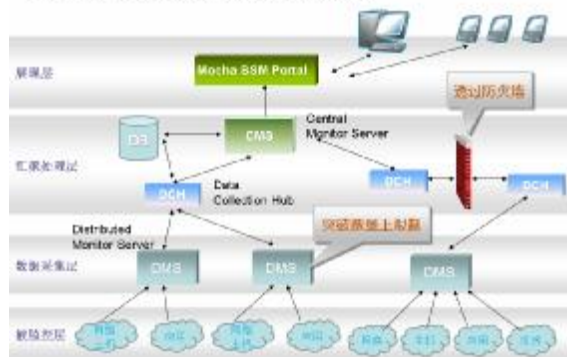
1.3 Pervasive 产品无所不在

通过 Portal， 产品提供了“3A”的无所不在：

- n **Anywhere** 在哪里客户都能访问到 Mocha BSM IT 运维平台。
- n **Anytime** 不分任何时间，只要有相对应的权限，都可以访问 Portal
- n **Anyhow** 支持多种接入方式，包括浏览器，手机等

产品部署架构

Mocha BSM 分布式部署架构图



产品被划分为 4 个层次，层次之间都是通过 API 沟通。这 4 层分为：

1.4 展现层

展现层主要用于图形化展现采集来的监控数据；与用户互动，响应用户的操作与设定；以及集成第三方监控产品等。展现层主要组件及其特点是：

n Mocha BSM Portal

- n 统一资源管理
  - 集成企业现有 LDAP
  - 提供用户管理以及权限信息
- n 统一认证管理与登入
  - 提供 LDAP 认证方式
  - 采用单点登录 (SSO)，无需用户频繁登录/退出各种 IT 营运系统。用户无需记得多套帐号和密码。
- n 统一页面展现
  - 采用了统一的界面设计、功能设计。用户只要掌握了一种资源在系统中的监控方法，即可触类旁通。
  - 同时，Mocha BSM Portal 可集成第三方监控产品，现在用户不需要不同系统页面切换了。

1.5 汇聚处理层

汇聚处理层用于处理数据采集层采集到的监控数据（例如，根据监控数据生成报警），并写入数据库，以供展现层从数据库调用监控数据；同时也用于处理展现层传递的用户操作与设定（例如，将用户定义的监控策略下发到数据采集层）。汇聚处理层主要组件及其特点是：

- n **CMS:** 全称为 Central Monitor Server（中央监控服务器），负责生成报警、策略的下发、处理 Portal 中用户的输入等。
- n **DCH:** 全称为 Data Collection Hub（数据采集集线器），负责将采集的数据写入数据库，或突破防火墙上配置。DCH 的上级 CMS 可在配置文件配置，并且允许配置多个备用 DCH 节点，以实现高可用性（High Availability）。

## 1.6 数据采集层

数据采集层用于采集被监控资源各项指标的数据，并上传至汇聚处理层。数据采集层的主要组件及其特点是：

- n **DMS:** 全称为 Distributed Monitor Server（分布式监控服务器），负责发现被监控资源并采集各项指标。
  - n 可扩展性：支持横向扩展（即一个 DCH 下可配置多个 DMS），以突破监控数量的限制；
  - n 可配置性：DMS 上级 DCH 可在配置文件灵活配置；
  - n 高可用性：可配置多个备用 DMS 节点，实现高可用性（High Availability）；

## 1.7 被监控层

被监控层主要包括 Mocha BSM 通过 Agent 或 Agentless 方式监控的各种资源，例如，主机、网络设备、应用等。

### 关键模块介绍



Mocha BSM 主要模块包括：

## 1.8 Portal

- n **可定义 Portlets:** 提供丰富的 Portlets（8 种类别，30 个配置 Portlets）供展示之用，可直接在系统管理设置，无需二次开发。可满足各类角色的需求，如 IT 部门领导、系统管理员、应用管理员，业务分析员等。
- n **Dashboard:** 以业务的角度展现监控内容，让业务部门和 IT 部门领导了解提供给他们业务服务状况而不是技术细节，通过仪表盘展现业务服务状况以及 KPI。

## 1.9 Mocha BSM Server

- n **事件管理**
  - n 根本原因分析：准确评估事故所造成的影响，快速确定处理事故的范围；以及准确定位事故的根本原因，对症下药的快速解决事故；
  - n 自动关联事件：将引起当前事件的可用性、性能或配置变更指标自动关联，不必大海捞针般去查找，提高了工作效率；
  - n 自定义事件：用户可将系统预定义事件灵活组合成用户所需事件，这也为事件触发流程提供了更丰富的触发条件；
- n **资源模型管理**
  - 快速开发：使用 XML 配置被监控资源的指标、事件、状态，并提供资源模型开发工具，使每个资源模型的开发周期仅为 3 天左右；
  - 即插即用：开发完新的资源模型后，上传至 Mocha BSM 服务器即可实行监控，无需改动代码；
- n **资源发现管理**
  - 多种发现方式：可单个发现资源，也可批量导入要发现的资源；
  - 自动发现网络拓扑：可自动发现网络设备并生成网络拓扑图；

n 数据采集

- 多种采集方式：支持 Agent 与 Agentless  
两种采集方式；
- 高安全性：支持 SSH 加密，使用 256RSA  
SSH 公钥私钥协议,提供认证，完整性验证  
和数据压缩功能；
- 支持多种脚本：Perl,C,JavaScript,Windows  
Script,VB Script,Python,Ruby；
- 支持多种协议：Performance Lib, WMI,  
JDBC, JMX, SMARTS 等；
- 支持二次开发：支持开发多种指标插件，指  
标开发语言包括 Window Host Script, Shell  
Script, PERL, C/C++, Python, Rhino,  
VB, Java 等；

### 1.10 插件容器

- n 指标取值插件分类管理：MRAM, JDBC, JMX, 应用等插件服务器互不影响，一旦某插件服务器不可用，不会影响 Mocha BSM Server 采集其他类型指标；

### 1.11 摩卡 IT 运维管理 (Mocha IT Operations Management)

- n ITIL 流程模板：提供丰富的 ITIL 流程模板，例如：故障上报流程、关键指标超标流程、宕机管理流程、发布流程等。无需用户开发，可直接使用。
- n 流程、表单定义：
- n 工单管理
- n 灵活的表单和流程配置功能：提供工单创建功能，实现工单与表单和流程的绑定。

## Mocha BSM 4+1 介绍

### n 三位一体的产品定位

摩卡软件是亚太区率先推出三位一体产品定位的软件提供商之一，三个定位包括了：

#### n 网络管理

( Network Management System )

— 传统意义上的网络、系统、应用监控，满足了成长中企业的需要；

#### n IT 运维管理

( IT Operations Management )

— 把监控上升至管理的层面，帮助企业规划、运维和改进 IT 系统。通过端到端的监控，帮助中大型企业管理 IT 系统；

#### n IT 服务管理

( IT Service Management )

— 基于 ITIL 流程框架，带领企业进入流程化，规范化和自动化的时代。



三位一体的解决方案

### n Mocha BSM 4+1 做得更多

为了满足三位一体的定位，摩卡软件推出了 Mocha BSM 4+1 产品套装。

Mocha BSM 4+1 涵盖了以

下几方面：

#### 🔗：基础架构管理

— 网络拓扑、主机、流量分析、IT 资产；

#### 🔗：应用管理

— 应用服务器、数据库、Web 服务器等；

#### 🔗：端到端响应时间管理

— 应用响应时间管理，端到端监控；

#### 🔗：业务服务管理

— 以业务视角看待 IT；

#### 🔗：IT 运维管理

— 基于 ITIL 流程框架，满足对事故管理、问题管理、性能管理、变更管理、配置管理发布管理、知识库等需求。



Mocha BSM 解决企业 4+1 方面的问题

### n Mocha BSM 4+1 套装包括

n 以服务的视角看待 IT，提供以服务导向的监控

— 摩卡业务服务管理

Mocha BSM ( Business Service Management)

n 完整的 IT 资产生命周期

— 摩卡 IT 资产管理

Mocha ITAM ( IT Asset Management)

n 帮助企业找出网络带宽的瓶颈

— 摩卡流量分析

Mocha NTA (Network Traffic Analyzer)

n 基于 ITIL 流程框架，以服务台为中心，提供流程化管理

— 摩卡 IT 运维管理

Mocha ITOM (IT Operations Management )

n 提供端到端监控

— 摩卡端到端监控管理 Mocha E2E (End To End ) Monitoring)