



CMDB的定位及价值



ITIL 先锋论坛

我们是谁?
WHO ARE WE?

国内最大的数字化时代IT服务管理交流社区，自2010年底成立以来，始终致力于以ITIL为代表的IT管理方法论在国内的推广与落地。

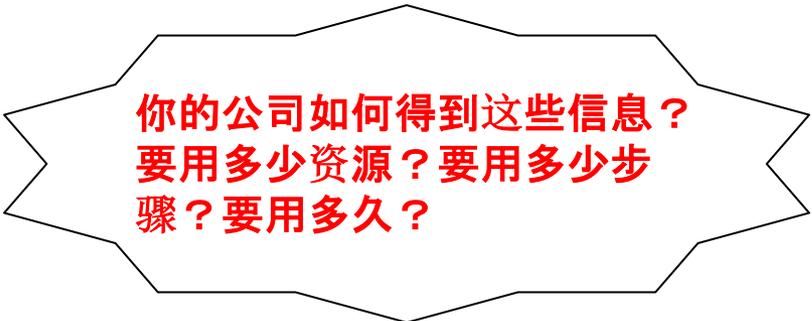
我们的服务
OUR SERVICES

数十个专业微信群、近千篇可一键下载的资料、视频号专家直播、全国一线城市巡回聚会、开源免费ITIL软件、国内最权威的ITIL知识库

- ▶ 常见的IT服务问题
- ▶ CMDB的定位
- ▶ CMDB的模型
- ▶ CMDB的价值
- ▶ CMDB的绩效



1. 2号核心机房的74号机柜中到底存放着多少个IT设备？ --位置
2. 网上银行的业务系统到底有多少台服务器？ --系统
3. 一个具体的系统管理员手中管理着多台小型机？ --人员
4. 6.5.21版本的ESP软件到底安装了多少个？ --版本
5. 刀片服务器有多少台？小型机有多少台？ --类型
6. DL380的服务器有多少台？有多少台库存备机？ --型号
7. 这台PC服务器使用了多长时间？
8. 整个公司有多少种Oracle数据库版本？分别有多少个？



**你的公司如何得到这些信息？
要用多少资源？要用多少步骤？要用多久？**

1. 这台序列号“99C9962”的X3850PC服务器，过去发生过哪一些事件？
2. 这款物流部门使用的VLS系统现在还有多少个BUG等待修正？
3. Internet区的网络下周都安排了什么变更？
4. 一个交换机接入了多少个对象？
5. CM20101105007这张变更单，当时到底动了哪几个设备？
6. 2号核心机房现在有多少个设备现在处于非正常状态？
7. 上个月事件最多的10台机器是哪几个？
8. 当一个用户做事件申报时，服务台怎样知道哪一些？
9. 对一台交换机做变更时，可能影响哪一些业务系统？
10. 哪一些软件与设备已经过保？或许可证不足？

**你的公司如何得到这些信息？
要用多少资源？要用多少步
骤？要用多久？**

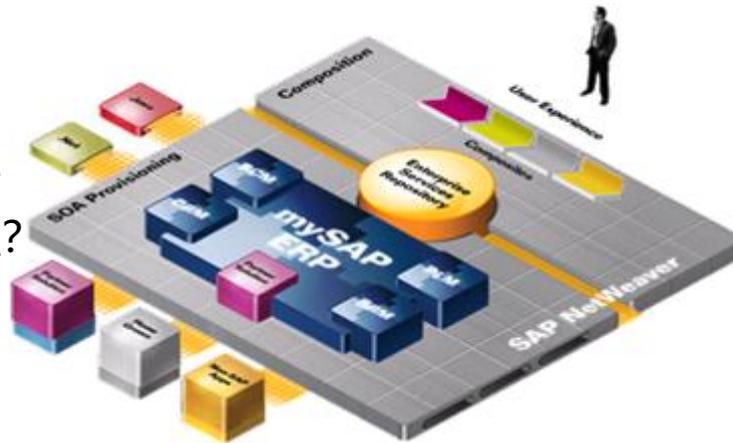
常见IT服务的问题（第三个层次）

每一个组件类中，事件有哪一些分布特性？比如中间件的哪一个版本事件率最高？

服务质量的下降到底是由哪一些组件决定的？反向的每一个组件可能与哪一些SLA相关？

哪一些组件是我们的架构的薄弱点？，明年的IT架构加固与改善的重点是什么？

终端机器的事件率与使用年限有没有函数关系？与品牌或厂商有没有相关性？



工程师的时间资源花在哪一些对象上或哪一类对象上？

哪一类组件的变更失败率或回退率最高？有怎样的分布特性？

服务目录与IT架构的关系是否有可能梳理出来？是否可能明确每个IT组件可能的操作动作是什么？

工程师的时间资源花费最多的是做哪一些事务（服务目录）？同样的动作，在不同的机型上花费的时间不同吗？明年的人员能力提升的重点是什么？

- ▶ IT服务的重心在于服务，IT服务管理的重心在于管理，回归管理的本质。
- ▶ 管理依赖信息，没有科学合理的统计分析，我们既不知现在，亦不知未来
- ▶ 统计分析又依赖于大量详实而基础的信息记录
- ▶ 记录是一切管理的基础，它是整个体系最基本的元素，同时又是最枯燥、繁重、琐碎的部份，也是最被人忽视的部份。
- ▶ 记录是我们的历史



“管理工作和记者的工作有一些相似之处，那就是你必须要有相关知识的储备，而且在工作过程中根据你掌握的信息进行调整，这样你才能成为一个优秀的分析者”。

---明茨伯格

事件记录		
概述	详细描述	解决方案
2010-06-07 08:58 集中监控 管理XX部 XX系统 服务器端口连接数异常	2010-06-07 08:58 集中监控 管理二部 XX系统 服务器端口连接数异常 生产主机:b2c-web-12:端口 [tcp_80_FIN_WAIT_2]的连接数当前为:50 08:58 联系系统XX部XXX, 通知查看 请XXX部工程师填写处理过程与恢复时间及业务影响, 并确认应用系统及事件类别项。	解决方案: 1、经运维人员联系patrol项目组后分析, 此报警信息属于新释放的报警, 报警阈值需要重新调整。 2、此报警不影响生产, 对业务无影响。
2010-06-16 12:48 监控报警:管理XX部:ZZ系统:Unix服务器文件系统空间占用率高。	2010-06-16 12:48监控报警:管理XXX部:XXX系统:生产主机:B2C7:tmp的Unix服务器文件系统空间占用率当前值为:100%, 告警阈值为:[95%-100%]。 2010-06-16 12:48通知二部工程师许金荣, 回复: 传输文件过大导致报警, 对业务无影响。 请系统工程师填写处理过程与恢复时间及具体报警原因。	运维回复: 在发现告警后立即删除此文件, 问题得以解决, 对业务无影响。

一个问题是: 我们应该如何记录? ?

变更记录		
应用系统	概要	变更内容描述
XXX系统	XXX系统部署后台服务器清理脚本	依照宁夏开发中心XXX项目组给出的清理策略编写了脚本, 计划于本周五部署到后台服务器上, 并增加到weblogic用户的crontab中定时执行.后台服务器共4台AP上。
YYY系统	YYY系统b2c、b2b、md新建应急库: 采用Oracle的DataGuard数据容灾备份技术, 通过让应急库延迟半小时应用归档日志防止影响数据库正常运行的大事物或者影响生产的错误操作同步到应急库, 关键时刻将生产切换到应急库已到达减少停业时间的目的。	1.在DB2上创建个人网银ebank的dataguard备库, 调整ebank的归档属性让应急库延迟半小时应用归档日志。 2.在EBSDB03上创建企业网银b2b的dataguard备库, 调整b2b的归档属性让应急库延迟半小时应用归档日志。 3.在DB2上创建md的dataguard备库, 调整md的归档属性让应急库延迟半小时应用归档日志。

- ▶ 记录不但意味着记录“事情”（事件、变更、发布），还意味着记录“物体”，只有“事物”相结合，记录才完整。
- ▶ 事情（工单）记录于IT服务流程之上。
- ▶ 物体（对象）记录于CMDB之上。
- ▶ 要实现真正面对服务的管理，必须有一个坚实的基础：CMDB
- ▶ 只有记录足够结构化与标准化，科学而全面的统计分析才有可能

CMDB+流程=事+物=完整记录
完整记录→全面分析→科学管理

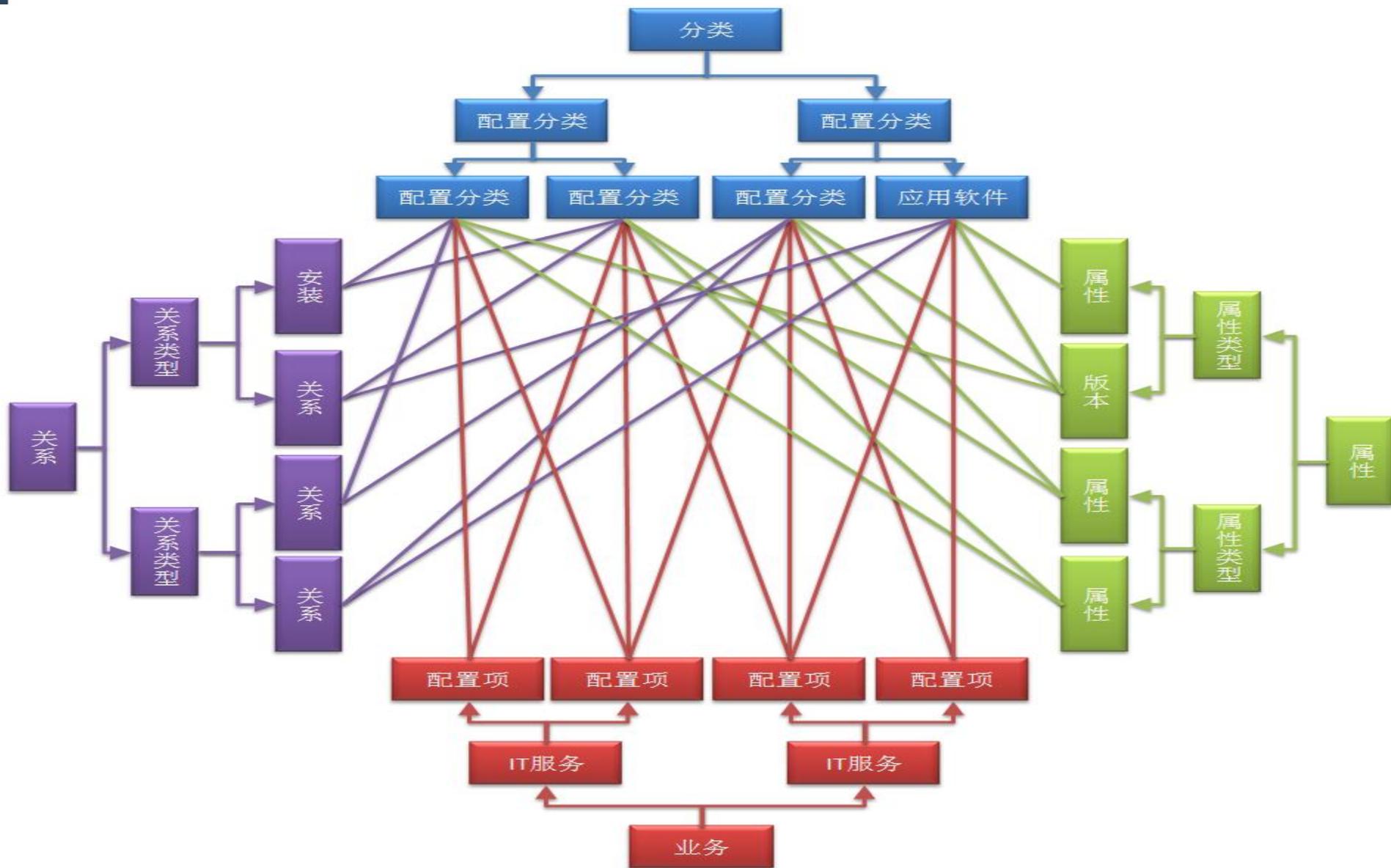
面向对象

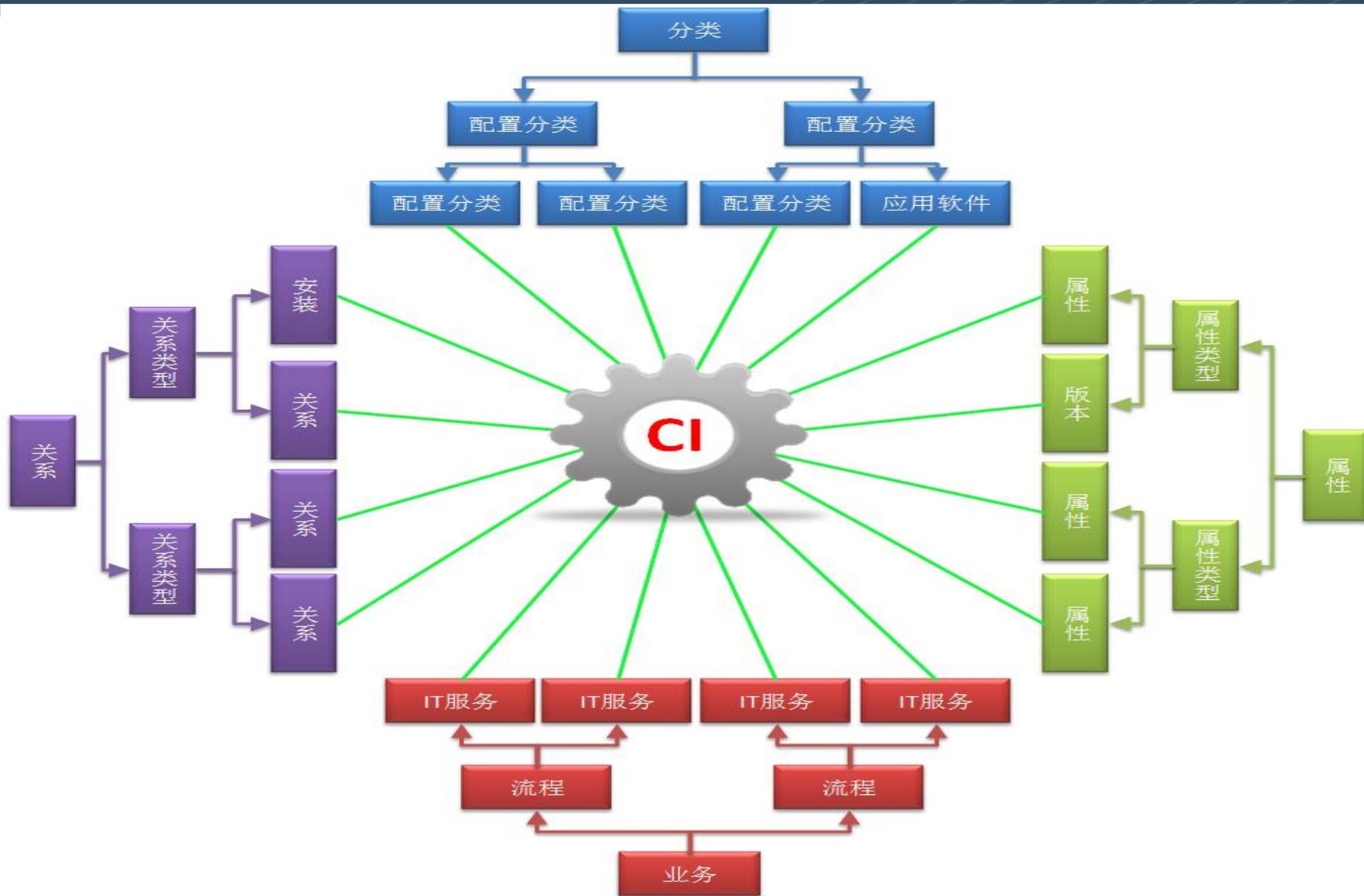
面向对象的思想
确定每一个在服务范围内的服务对象
每一个服务对象就是CMDB的CI
确定每一个CI的分类与属性
确定CI彼此之间的关系
投影出你的IT架构（即模型）

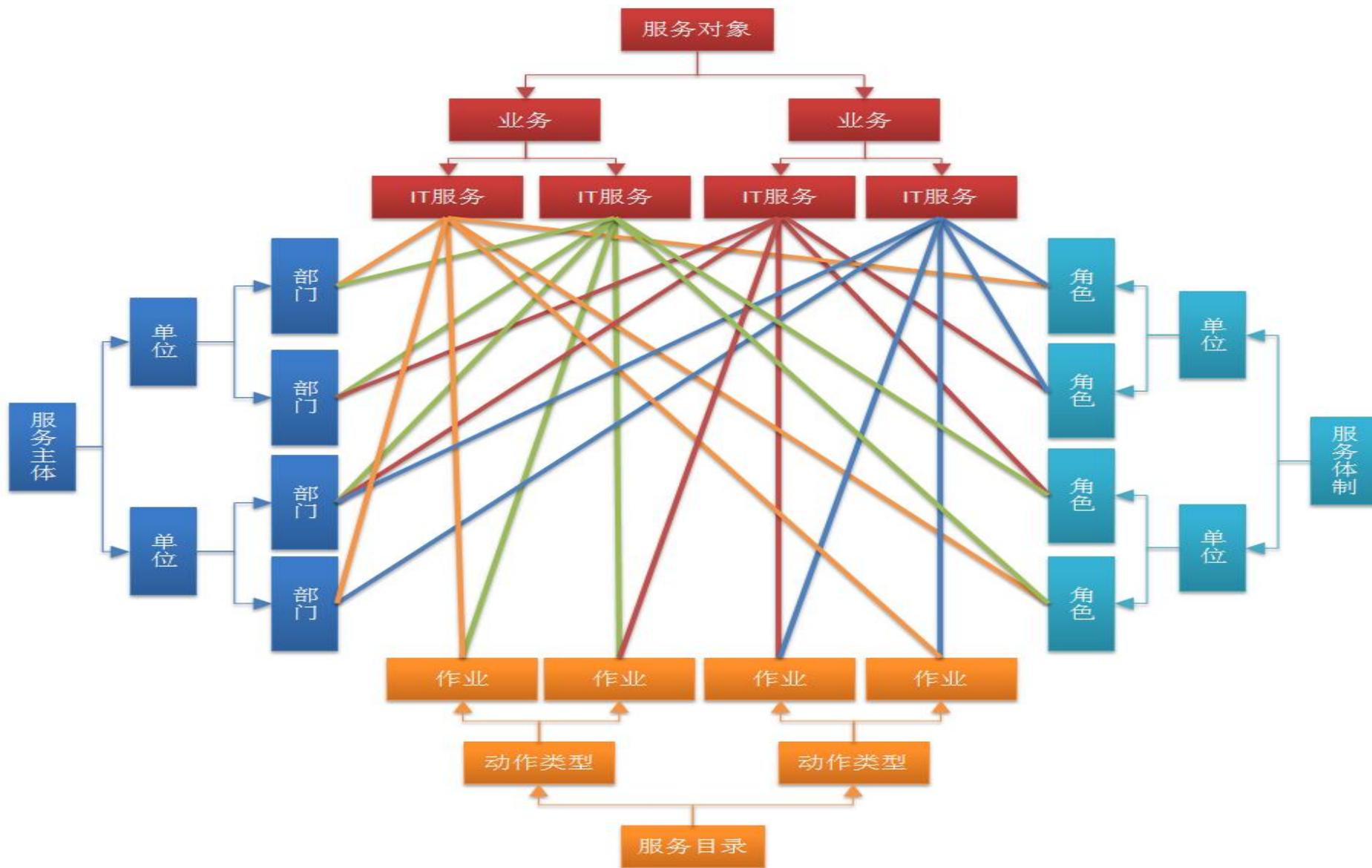
概念定义

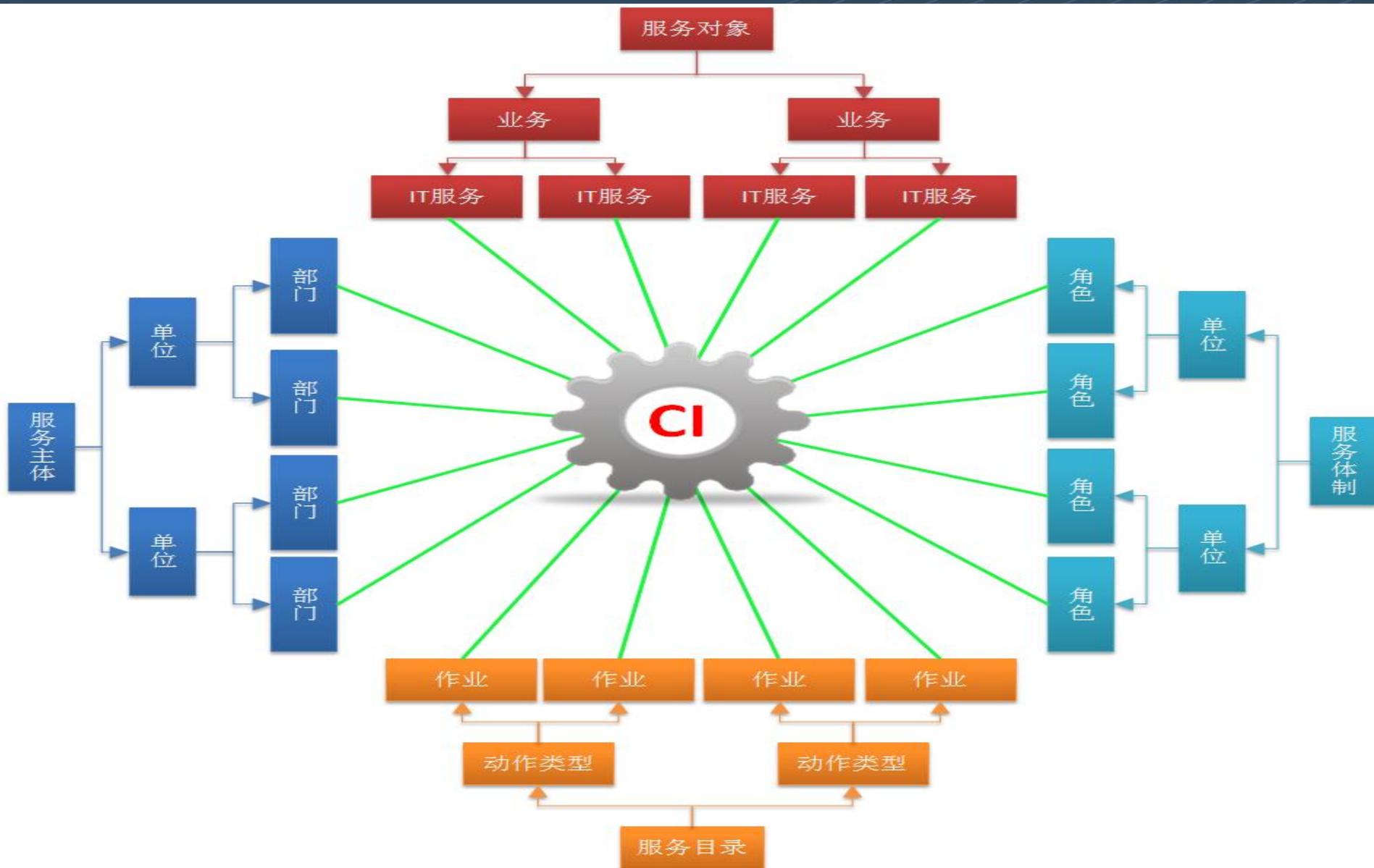
分类：有一些种类的对象需要被管理
属性：每一个种类的对象有一些特性需要被记录
状态：对象的生命周期中哪一些阶段需要被管理
关系：每一类对象跟哪一类对象存在哪些关系
CI：需服务的、唯一的、可变的、可管理的对象









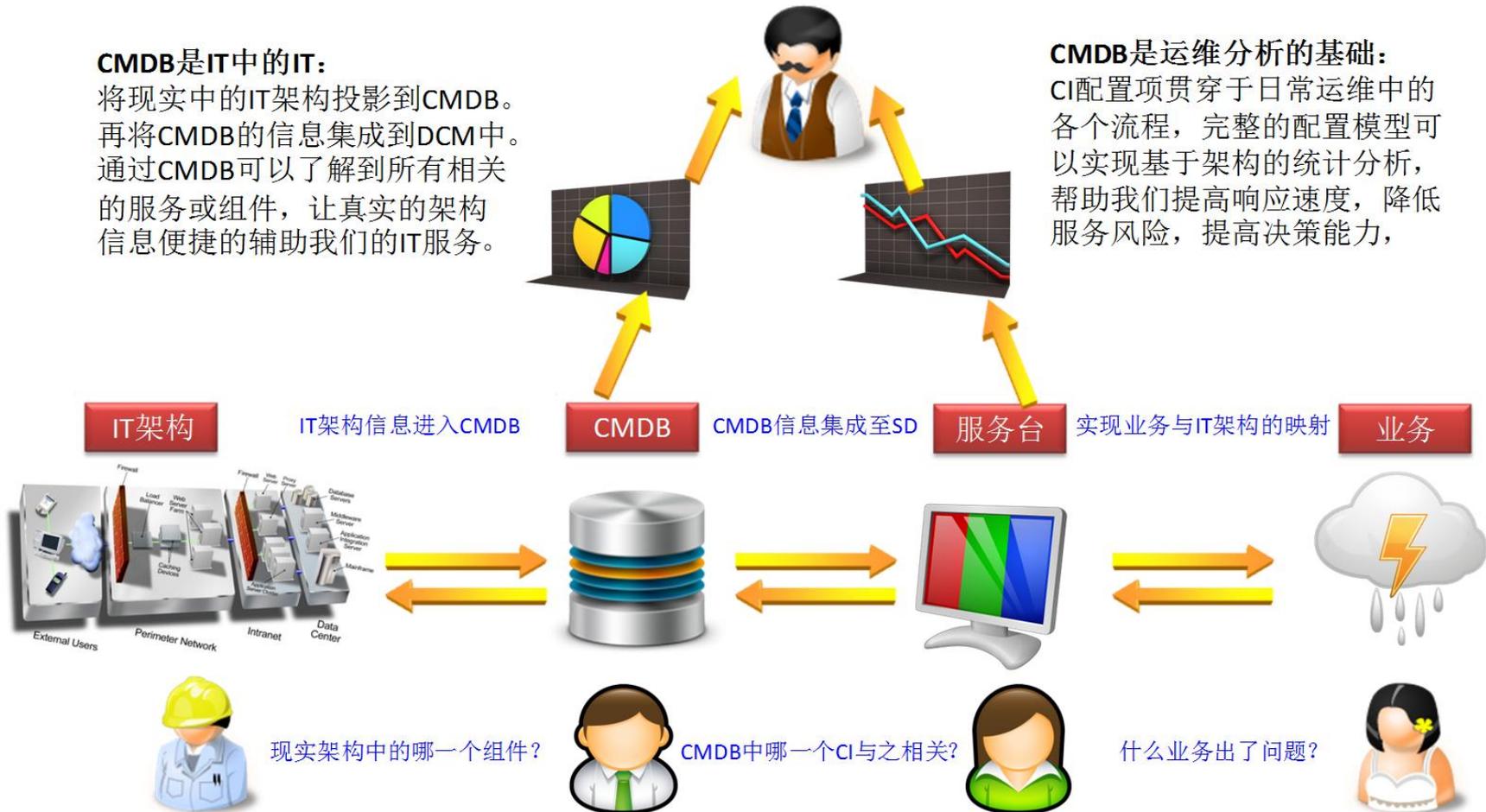


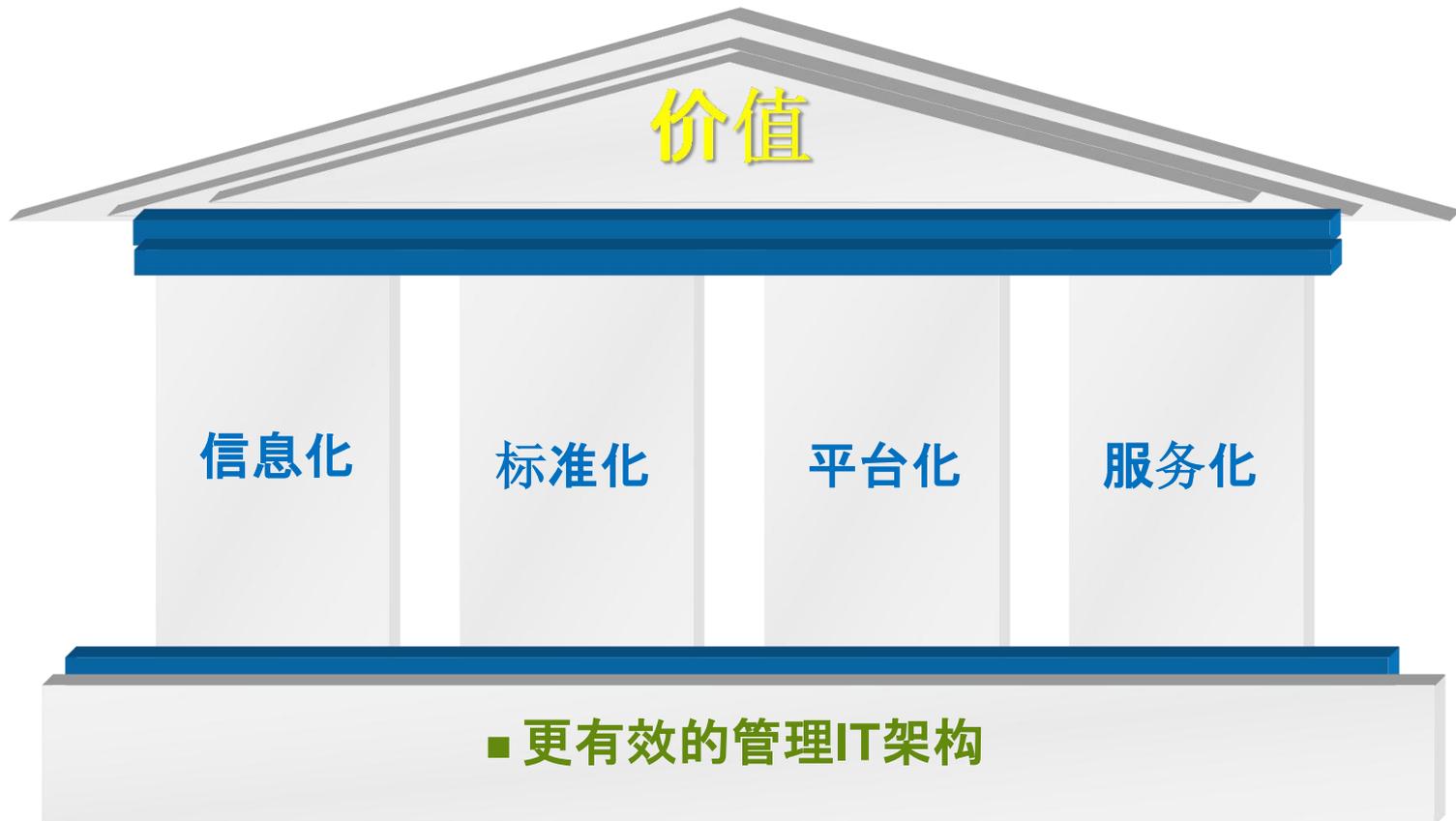
CMDB是IT中的IT:

将现实中的IT架构投影到CMDB。
再将CMDB的信息集成到DCM中。
通过CMDB可以了解到所有相关的
服务或组件，让真实的架构
信息便捷的辅助我们的IT服务。

CMDB是运维分析的基础:

CI配置项贯穿于日常运维中的
各个流程，完整的配置模型可以
实现基于架构的统计分析，
帮助我们提高响应速度，降低
服务风险，提高决策能力，





1. 实施CMDB的过程是一个重新理解、审视你的IT架构的过程
2. 将每个服务的对象进行识别的过程，会有助于我们理清运维权责，理清灰色的运维地带。
3. 当我们识别出IT架构的所有组件，以及这些组件之间的关系，我们才真正“描述了这个IT架构，通过这些数据的描述，投影出相应的模型，这是我们对IT架构本身的”理解”。
4. 真正理解定义了哪一些是“配置项”，变更的控制范围与深度清晰可见，我们可以清楚定义出，调整一些对象（CI）是变更，调整对象的哪一些部份（属性）是变更，甚至可以定义基于对象承载的动作（服务目录）是变更。
5. 管理一个大而封闭的IT架构是困难的，但当我们将它瓦解成一个个小的组件，根据组件的不同特性与类型，分散给不同的人员进行维护，它将变得相对容易，更重要的是，我们还掌握了还原组装这些IT组件的知识，即关系，并且通过可视化的方法让大家理解彼此如何成一个整体，即模型。

实现IT架构的信息化：

- ▶ 简化信息调查的工作量，让信息调查的发起者可以做基于CMDB的信息采集，而不需要面对每一个调查部门采集每一份信息的难题，被调查部门可大大减少配合信息时间，同时减少信息在不同人员之间传递、编辑、汇总的错误。
- ▶ 改变当前手工维护分散表单的管理难题，软件化管理各种配置信息，信息的正确性、及时性将得到大大的提高。
- ▶ 同一个配置信息被重复维护的低效状况将被解决，实现一处维护，四处调用的基本软件管理便利。
- ▶ 配置信息的检索可以随时进行，不再依赖部门或人员的配合，减少协调成本。
- ▶ 提供实时的统计分析报表，各类统计分析活动将更加便利。
- ▶ 维护一份正确的配置信息的作业成本将降少，不论是从系统管理员、系统经理，还是领导层或其它部门，其信息获取的成本及难度都将下降。

建立配置信息的标准：

- ▶ 解决了当前各部门、各系统对同一个配置信息的命名、记录、填写规范不一致的问题，这会以后信息集中分析与统计带来诸多便利，同时减少沟通成本
- ▶ 解决了当前各部门、各系统的配置信息表单多式多样的问题，帮助建立一个标准化的配置视图，将各部的信息模式统一标准化，信息以最简洁的方式被维护及统一。
- ▶ 每一个对象（CI）将成为形成独有的档案，它可以告诉过去与现在正在发生什么（事件、问题、变更）或者是什么样子（属性），它影响着什么，又被什么影响（关系），改变那种只有配置信息没有配置历史的记录模式，这将从根本上改变配置信息的使用范围及质量状况。

提供统一的配置信息管理平台：

- ▶ 解决了配置信息大量分散的各种文件的问题；
- ▶ 解决了配置文件存放各处，副本丛生的问题；
- ▶ 解决了配置信息存放于多个系统缺乏唯一合法源的问题；
- ▶ 解决了一份配置文件需要多个人维护，多个部门维护管理的问题；
- ▶ CMDB的应用会转变大家对信息的所有权及归属意识，信息将不再受限于每一个工程师，或每一个IT经理，或每一个部门，一切的配置信息属于企业，信息将在企业的级别被查阅与审查，每个系统或系统管理员手中的信息维护结果，不再封装在个人或部门内部，而需要呈现出来做检阅，
- ▶ 当CMDB统一将所有用户引导到平台之上进行维护调阅信息后，CMDB将为成唯一合法的配置信息平台，从而将所以一切对信息有更精细的需要的驱动力引导到对CMDB的数据改进之上，最后整个配置信息的精度将不断提高，这又导致更加深入的配置信息应用，整个企业级的信息PDCA将得到建立。

为面向服务的运维提供信息基础：

- ▶ IT架构信息将以服务的形式提供任何流程、任何部门、任何人员，信息更加真实、及时、方便、科学、规范，以往获取信息所耗费的时间与成本将降低，因此受益的可以是任何流程与任何部门或人员。
- ▶ 改变以往配置信息是一个平面数据的现状，采用模式图例的方式让用户理解IT架构，数据将被加工后以更加直观的方式表达信息，对宏观与细节将能更方便表达与获取，CMDB将为用户者们提供基于事实的信息服务。
- ▶ 统一、标准化的配置信息，将为日后的IT架构标准化、更新换代、新技术应用提供决策分析需要的事实依据，比如现在的中间件、数据库版本的标准化，或虚拟化应用等等。
- ▶ 从IT服务管理的角度，人 - 事 - 物的记录结合与关系建立，将帮助我们更加合理的分析现在的运维状况，也帮助我们对未来的状况做一定程度的推理，这将会影响我们管理服务成本的方式，甚至我们的IT运维资源的部署方式，
- ▶ 以面向对象的方式解析整个IT架构，加上自动化采集的手段，以往没有发现的对象或信息，或灰色地带将被清理出来，运维风险、安全风险、审计风险、法律风险也随之会降低。

不同时期，权重不同，通过权重调配作业重心



主项	权重	次项	权重	绩效点	权重	计算
应用	20%	事件	30%	事件CI定位率	100%	事件CI定位的总数与事件总数之比
		问题	20%	根源CI定位率	100%	根源CI定位的问题总数与问题总数之比
		变更	30%	变更影响关联率	100%	CI影响关联的变更单数与变更单总数之比
		报表	20%	CI分类引用率	100%	调用CI数据的报表数与报表总数之比
信息	30%	配置范围	20%	CMDB覆盖率	100%	CMDB中CI数与现实CI分类的对象总数之比
			10%	分类覆盖率	50%	CMDB中的CI分类数与现实对象分类数之比
		属性信息	20%	分类正确率	50%	CMDB中的CI分类与属性维护正确的CI分类数与CI分类总数之比
			20%	属性空置率	40%	该属性值为空的CI总数与有该属性的CI总数之比
				属性正确率	40%	该属性值为空或错误的CI总数与有该属性的CI总数之比
		CI信息	50%	属性增长率	20%	CI分类的属性总数增长比例
				信息精确率	50%	关系精确数/关系总数+属性精确数/属性总数（关系或属性值为空则计错）
CI正确率	50%	CI信息正确的CI总数与CI总数之比（关系或属性为空视为正确）				
作业	50%	CI信息维护	50%	基础数据维护执行率	20%	维护次数与应维护次数之比
			50%	CI数据维护执行率	80%	维护次数与应维护次数之比
		审计作业	30%	平均CI审计率	20%	一年内被审计CI总数与CI总数之比
				每周内部随机审计执行率	30%	随机审计执行次数与周数之比
				每月外部审计执行率	30%	审计执行次数与月数之比
				每年外部审计执行率	20%	审计执行次数与年数之比
		管理会议	20%	内部月配置管理总结会议	40%	会议次数与月数之比（专门时间，材料，会议）
				外部季度配置管理总结会议	30%	会议次数与季度数之比（跨部门级联合配置管理会议）
				年度配置管理总结会议	30%	会议执行次数与年数之比

- ▶ 我们回顾了IT服务过程中的常见的问题或信息需求。
- ▶ 我们了解了这些问题背后的深层次原因
- ▶ 我们知道了CMDB对最基础的工作的改变
- ▶ 我们分享了一个CMDB模型
- ▶ 我们分析了CMDB的价值
- ▶ 我们探讨了CMDB的绩效