

SACC 第八届中国系统架构师大会
2016 SYSTEM ARCHITECT CONFERENCE CHINA 2016

架构创新之路

架构创新之路

董国兴

dongguoxing@tsinghua.org.cn

目录

□ 架构概览

□ 总体架构

□ 应用架构

□ 数据架构

□ 技术架构

架构是什么

架构在大家头脑是什么样子？千人千样



平台架构

规则？

约定？

软件架构

企业
组织

架构是业务与IT对应的工具，是业务人员与技术人员沟通和理解的桥梁。

架构是对实现组织业务及战略目标的整体梳理和规划。

架构规划带来的价值

提升IT投资回报比

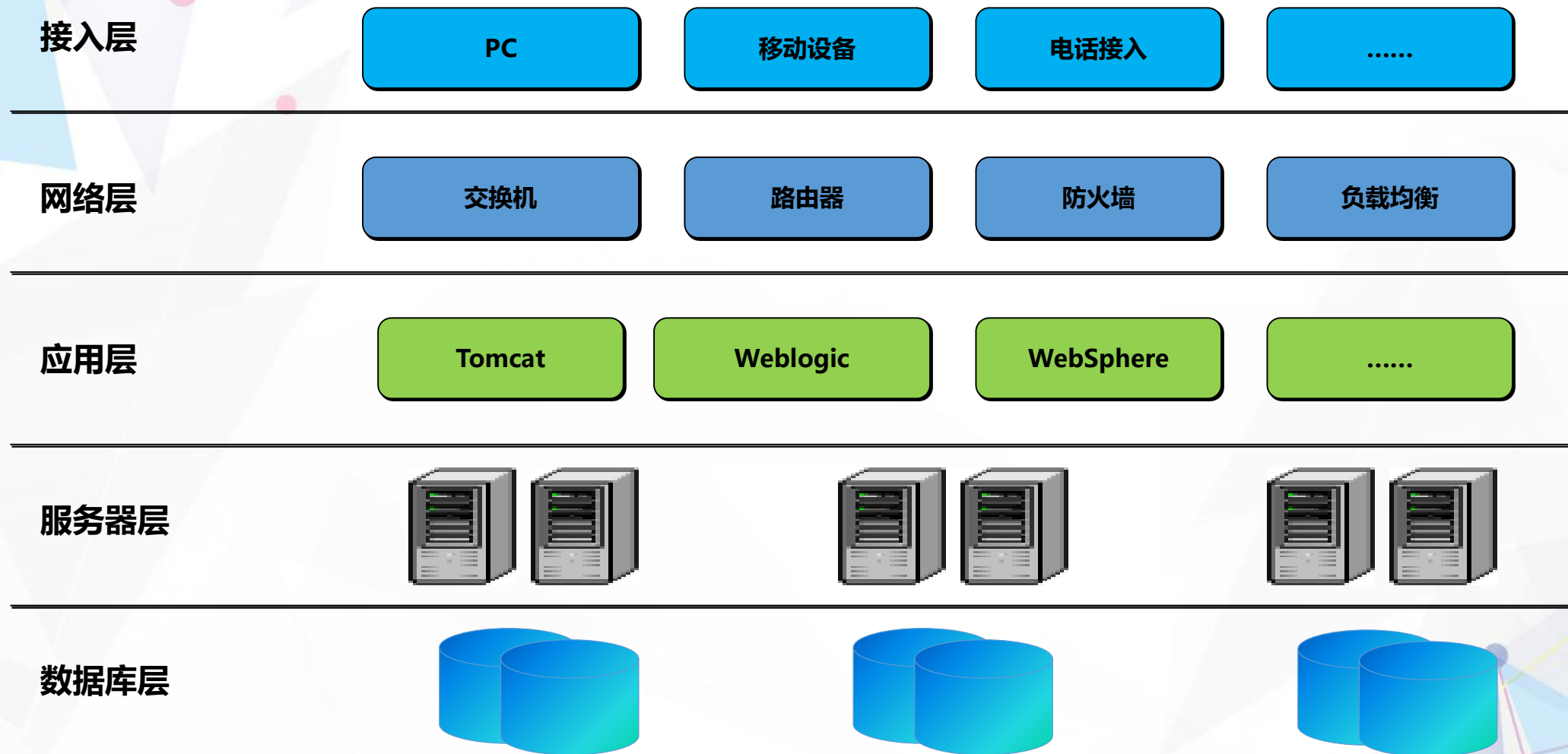
规避IT项目建设风险

具有整体的适用于业务的技术战略架构图

形成标准的流程和IT建设规范

实现整体信息和数据共享

传统架构 — 以IT为主，极少涉及业务



传统架构面临的新挑战



业务变化快

灵活度不高

无法快速部署

系统间耦合度高

IT标准缺乏

缺乏对业务整体支持的规划

出路在哪里？



架构方法论

TOGAF

- 国际标准权威组织 The Open Group 制定的架构框架。

FEAF

- 美国联邦政府制订的电子政务顶层设计。

Zachman

- Zachman 提出的独立于企业所使用的工具的平台框架。

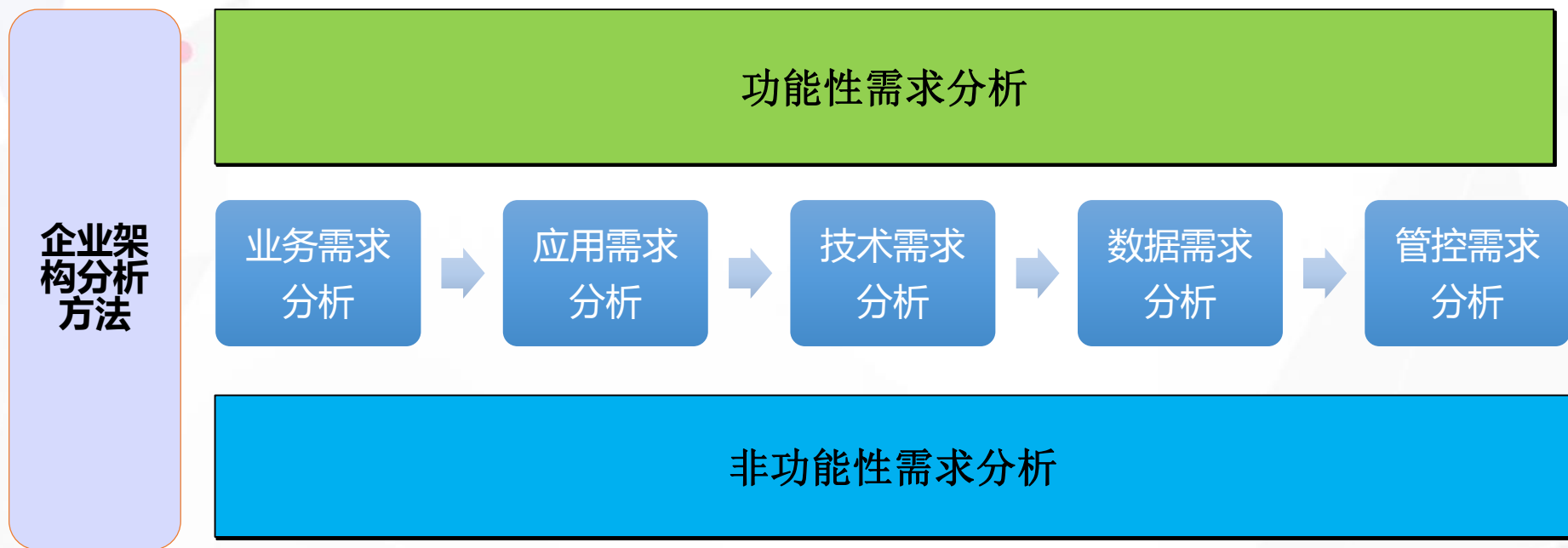
DoDAF

- 美国国防部提出的控制架构开发、维护和决策生成的组织机制。

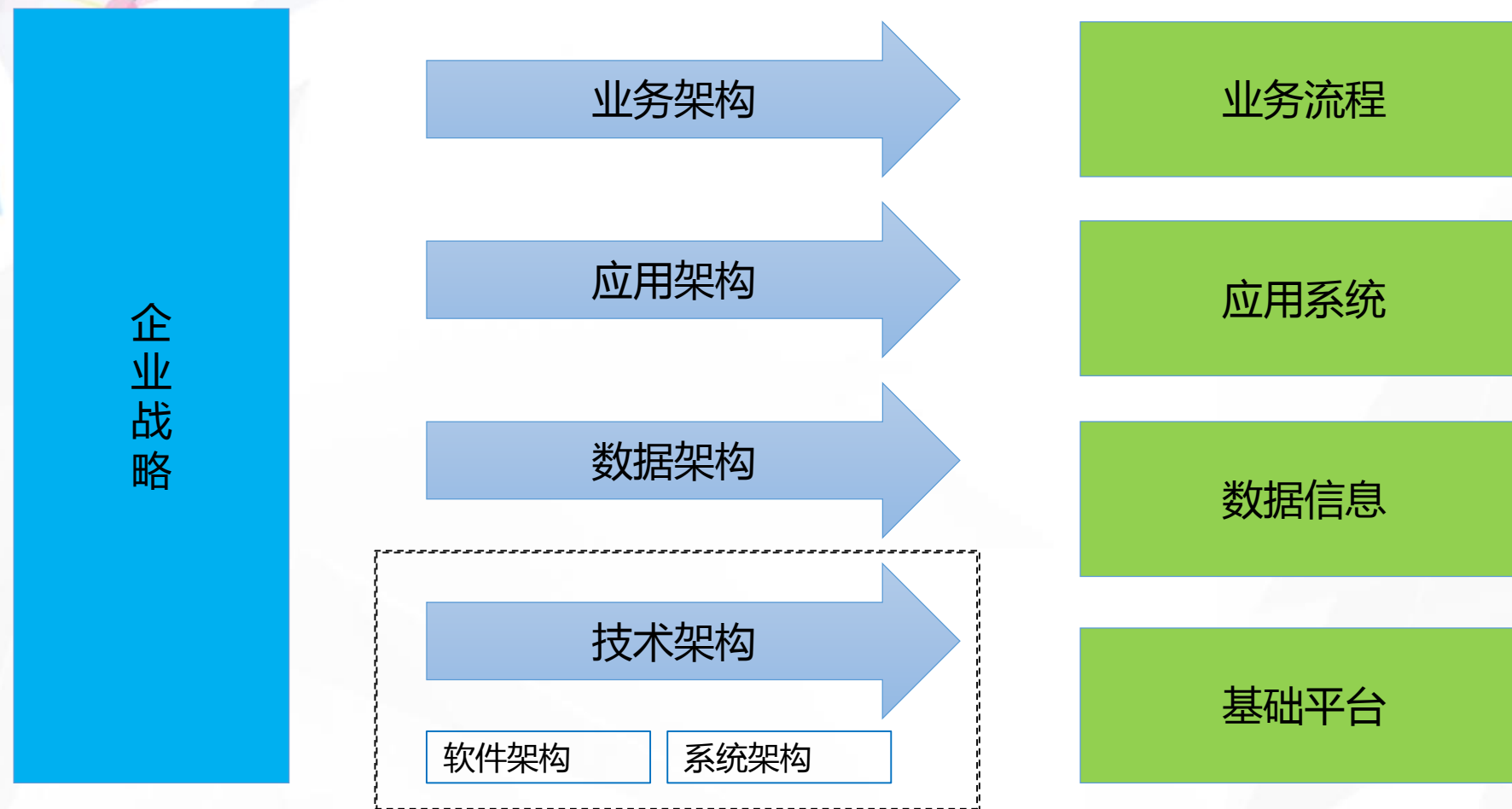
CBM

- IBM 开发的一套组件商业模式。

架构分析方法



架构概览



目录

□ 架构概览

□ 总体架构

□ 应用架构

□ 数据架构

□ 技术架构

应用架构设计

从业务需求出发，设计相应的功能模型，并在此基础上进行应用系统的划分，并进行场景验证和应用系统定义，形成应用蓝图。



功能模型（定义是什么）

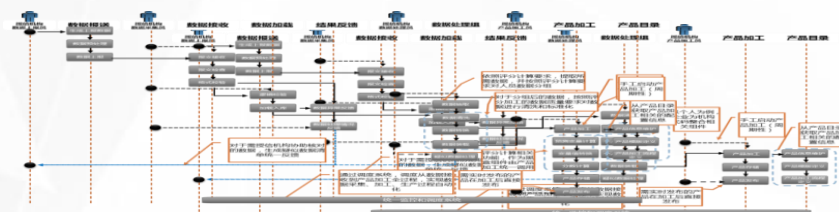


进一步抽象

分层模型（定义如何实现）



场景验证



应用系统定义



应用系统蓝图

数据架构设计

从数据特性出发，进行数据分类和模型化，根据操作要求设计数据目标架构，同时关注数据流转，以及管控要求。

数据特性

数据分类

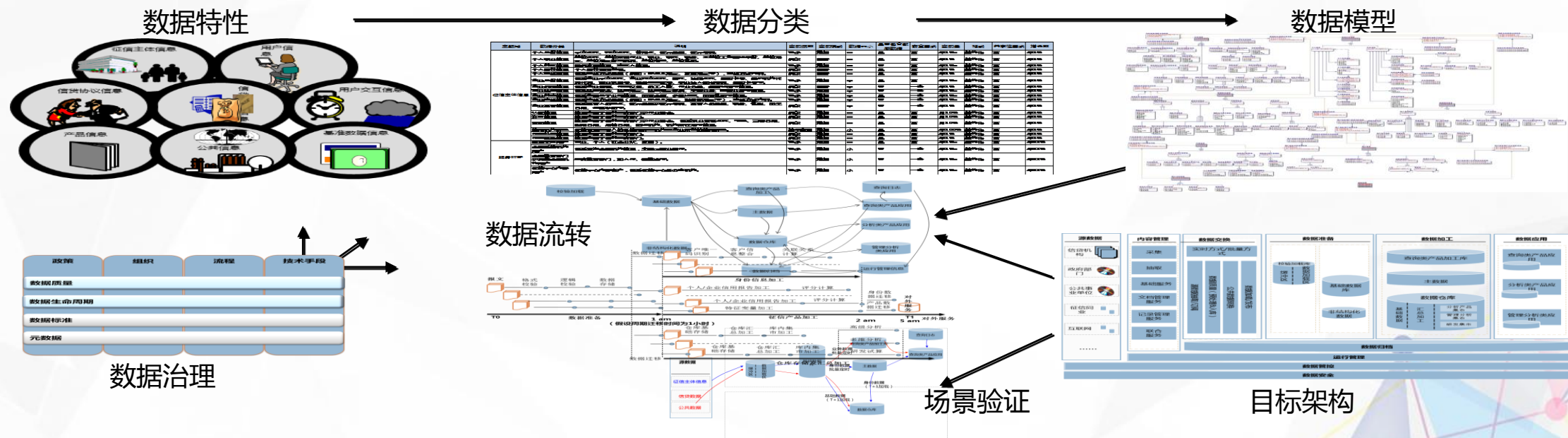
数据建模

形成目标架构

场景验证

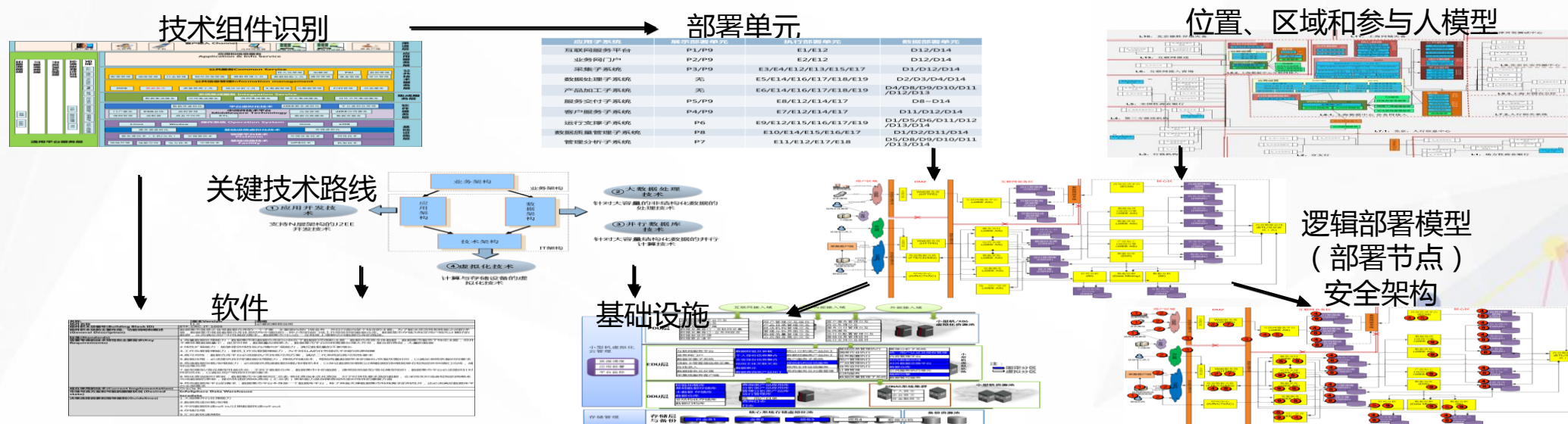
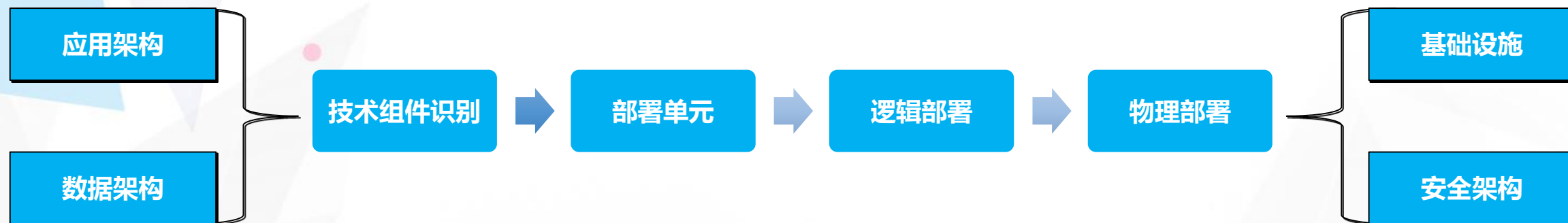
数据流转

数据管控



技术架构设计

从应用架构和数据架构出发，识别技术组件，在此基础上识别部署单元，设计逻辑部署与物理部署，同时关注基础设施规划与安全架构设计



目录

□ 架构概览

□ 总体架构

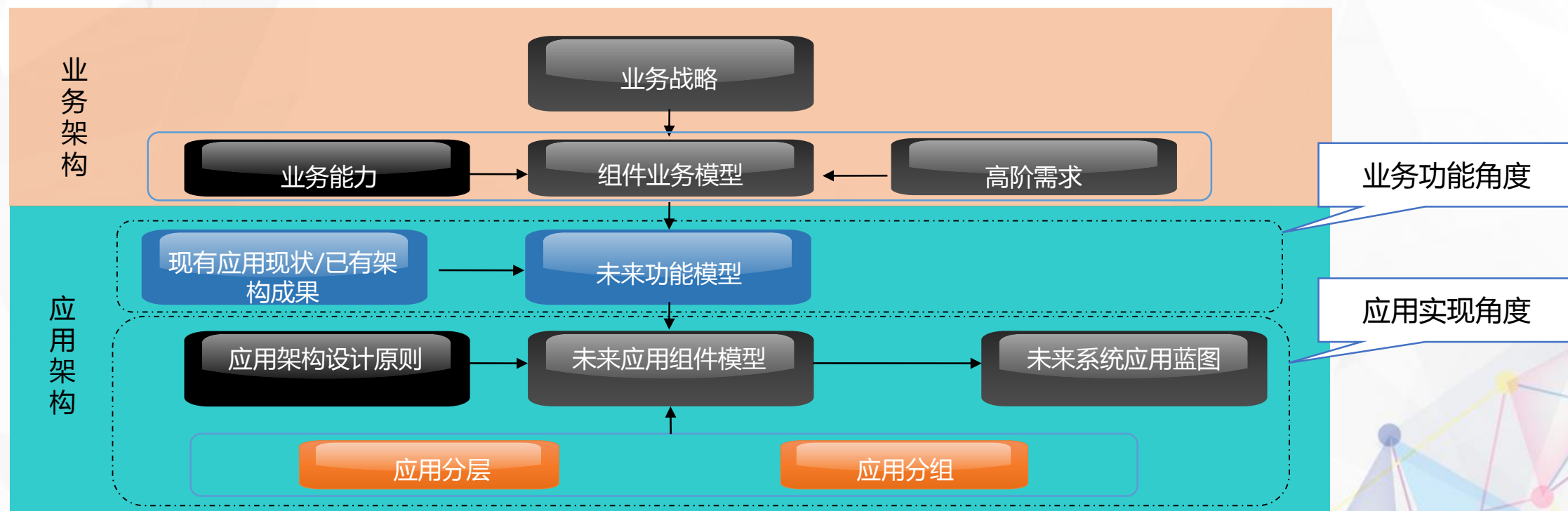
□ 应用架构

□ 数据架构

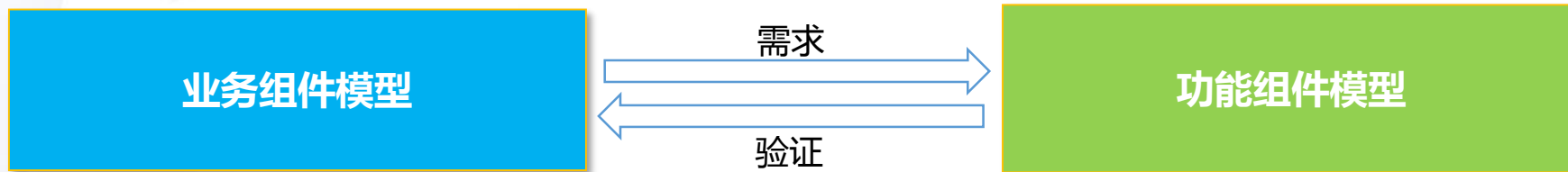
□ 技术架构

应用架构设计

目标应用架构来自业务架构的导入，实现对业务架构的支撑，从业务架构出发，梳理功能模型，形成未来的应用组件模型



组件识别，保证功能组件的完整性和唯一性



功能组件识别的重点

- ✓ 着重“是什么(what)”，而不是“怎么做(how)”
- ✓ 当两个组件功能重叠或近似时，要细分或者合并，确保没有重叠
- ✓ 每个功能要创造价值，并有重要的产出，去掉无价值的活动
- ✓ 从业务组件模型出发，检查功能组件覆盖度，确保功能组件覆盖所有热点业务组件

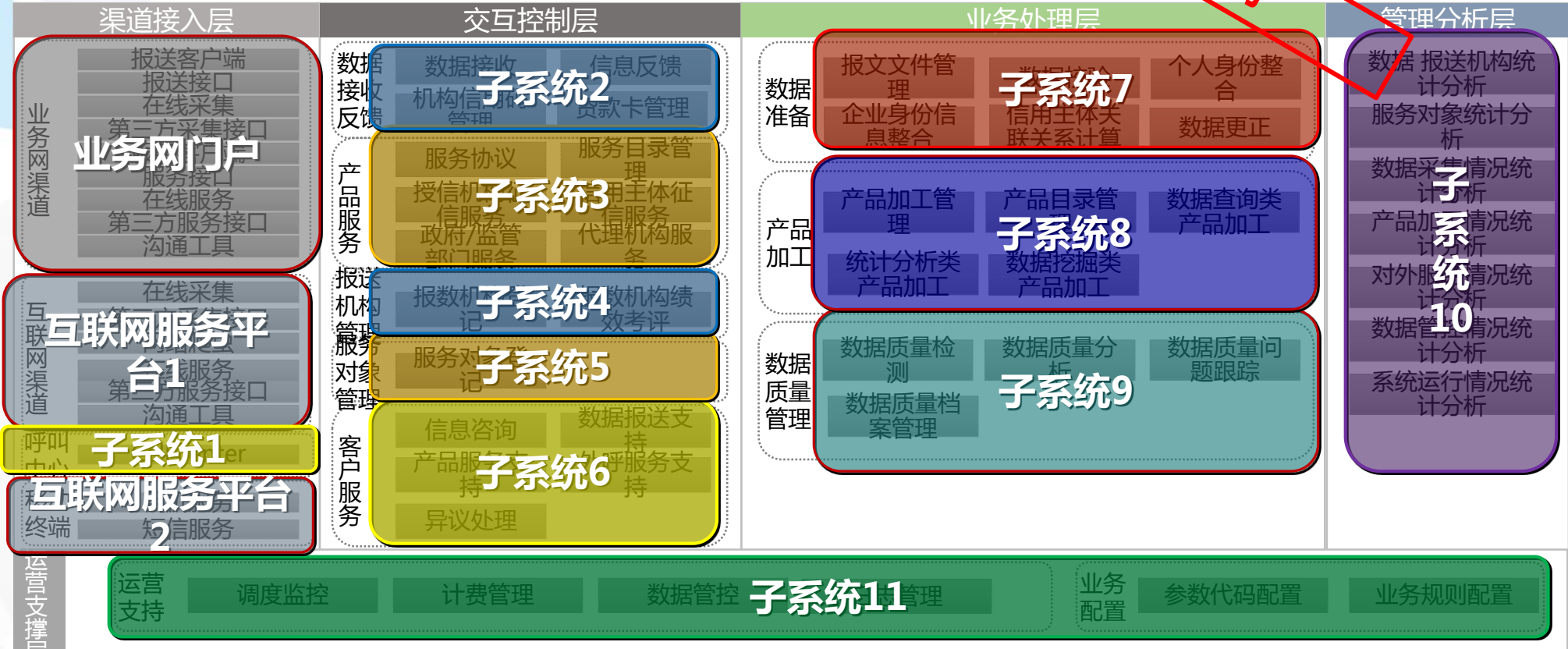
应用架构分层

良好的分层结构是实现**松耦合**的关键，以便系统易于维护和扩展

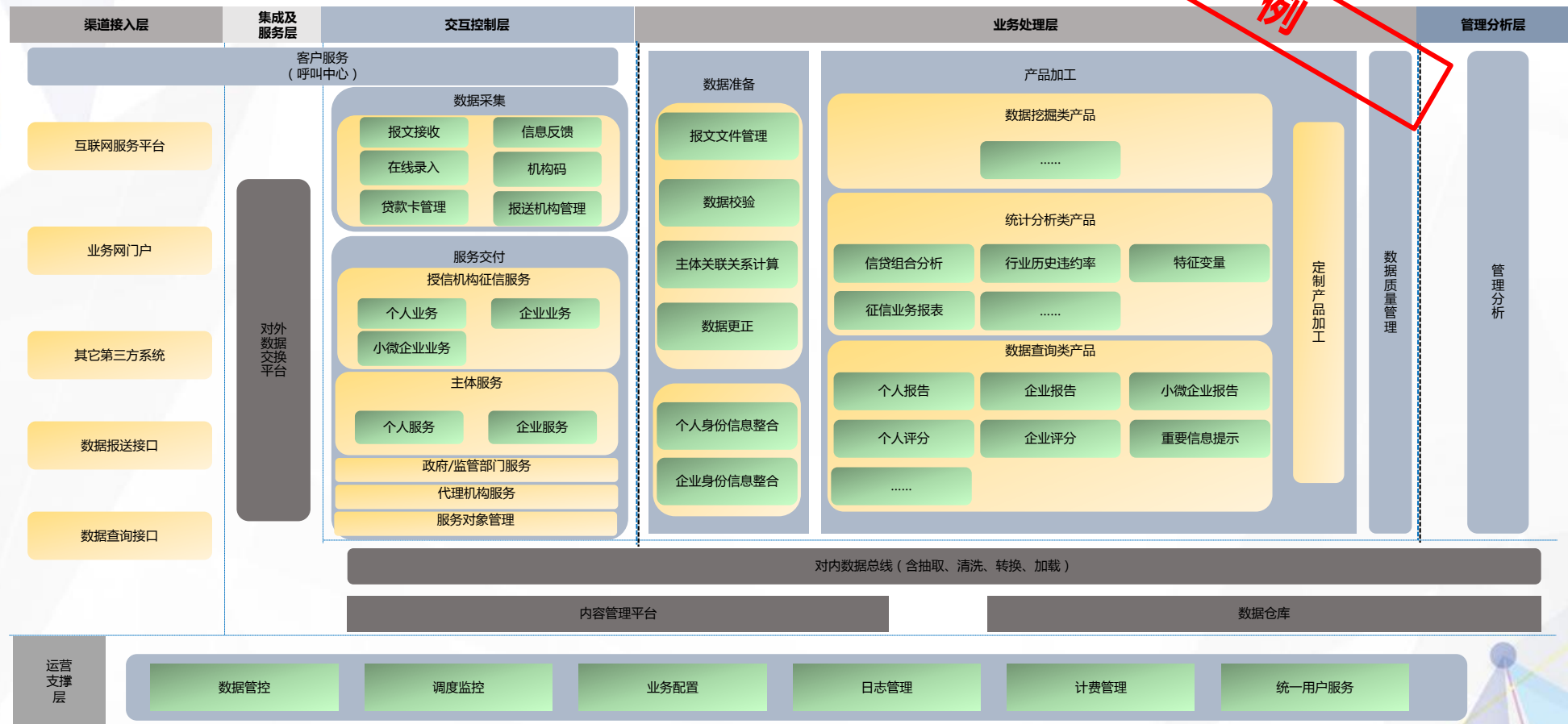
	术语	描述及举例
第0层 {	应用分层	应用架构的最高分类
第1层 {	应用组	有密切关连的一个或几个应用的逻辑组合
第2层 {	应用组件	用来搭建一个应用的业务功能组合

应用架构与子系统对应关系

示例



应用架构蓝图



目录

□ 架构概览

□ 总体架构

□ 应用架构

□ 数据架构

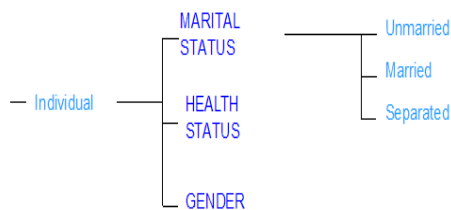
□ 技术架构

数据模型

- ✓数据模型是根据一定的业务规则，对业务概念或数据进行逻辑化表现。
- ✓数据模型从大的层级可以分为概念模型、逻辑模型和物理模型。

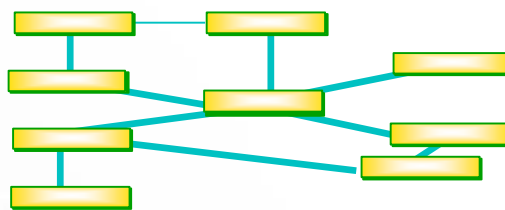
概念模型

- 概念模型是从业务角度对业务的抽象描述



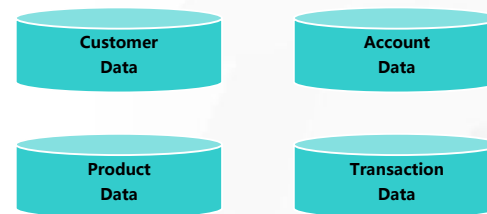
逻辑模型

- 逻辑模型是对概念模型的逻辑化，遵循第三范式



物理模型

- 物理模型是对逻辑模型的物理化，可以不遵循第三范式



良好的数据分类可以促进沟通

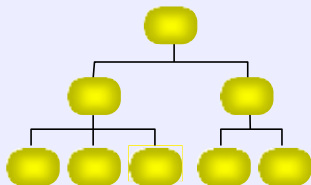
数据需求

- 业务人员描述所需的数据内容



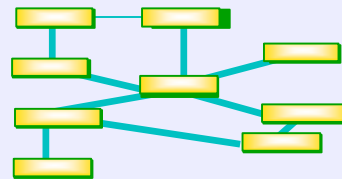
数据分类

- 根据数据内容抽象出数据分类，确定数据分类属性内容



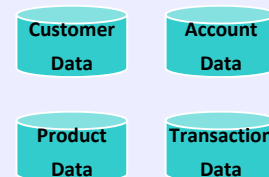
数据实体

- 根据分类，抽取出数据实体，用ER图描述其关系



数据库表

- 将数据实体物化为数据库表，供系统使用

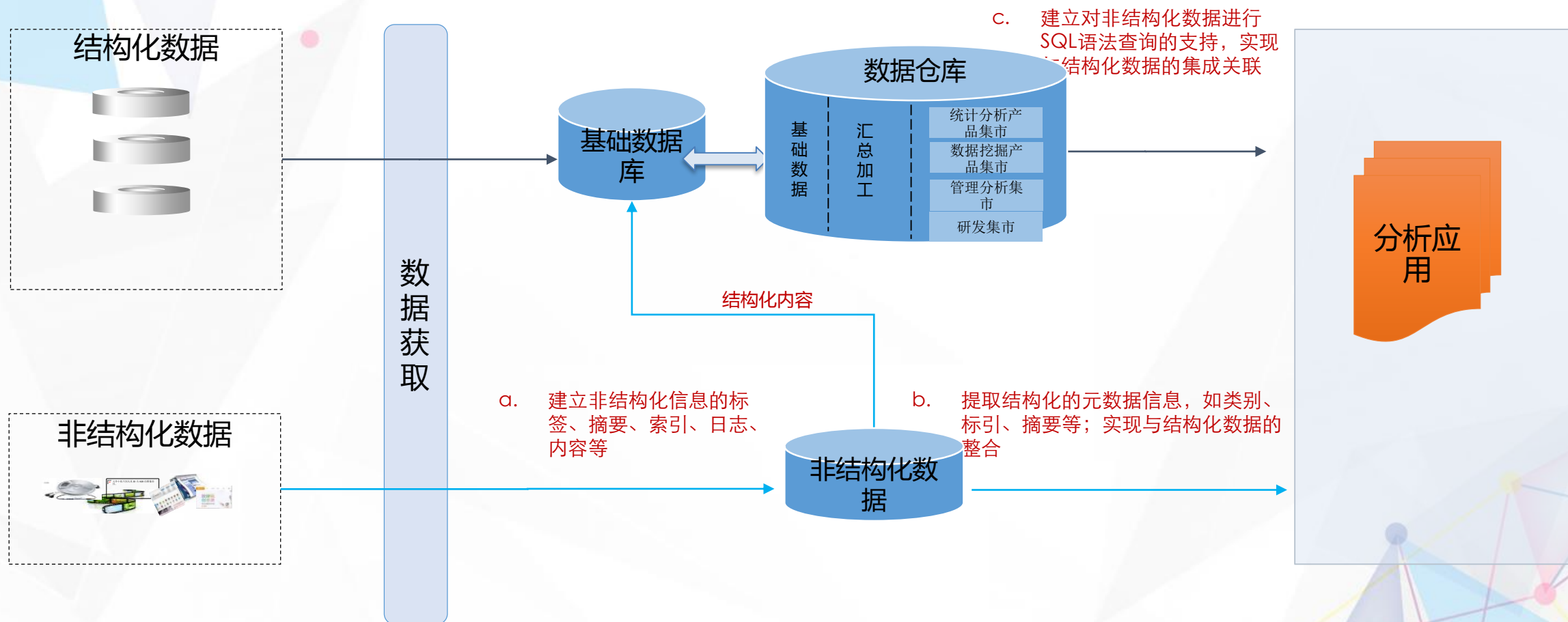


业务人员

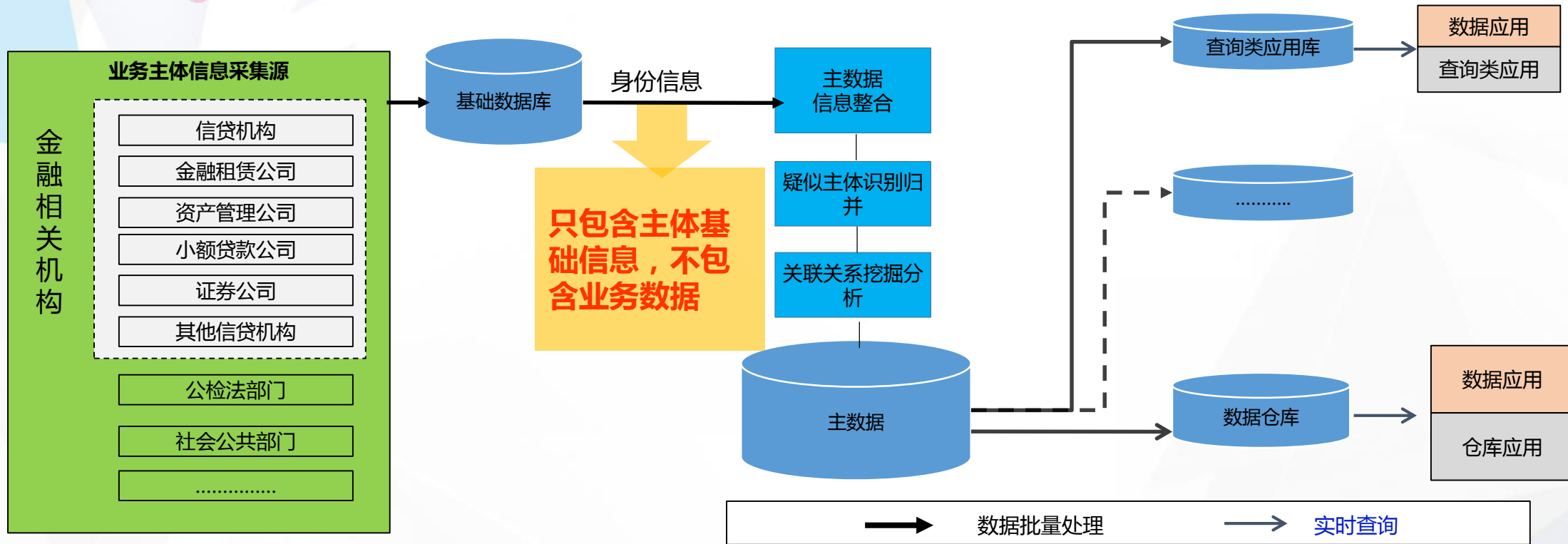
技术人员

结构化与非结构化数据整合

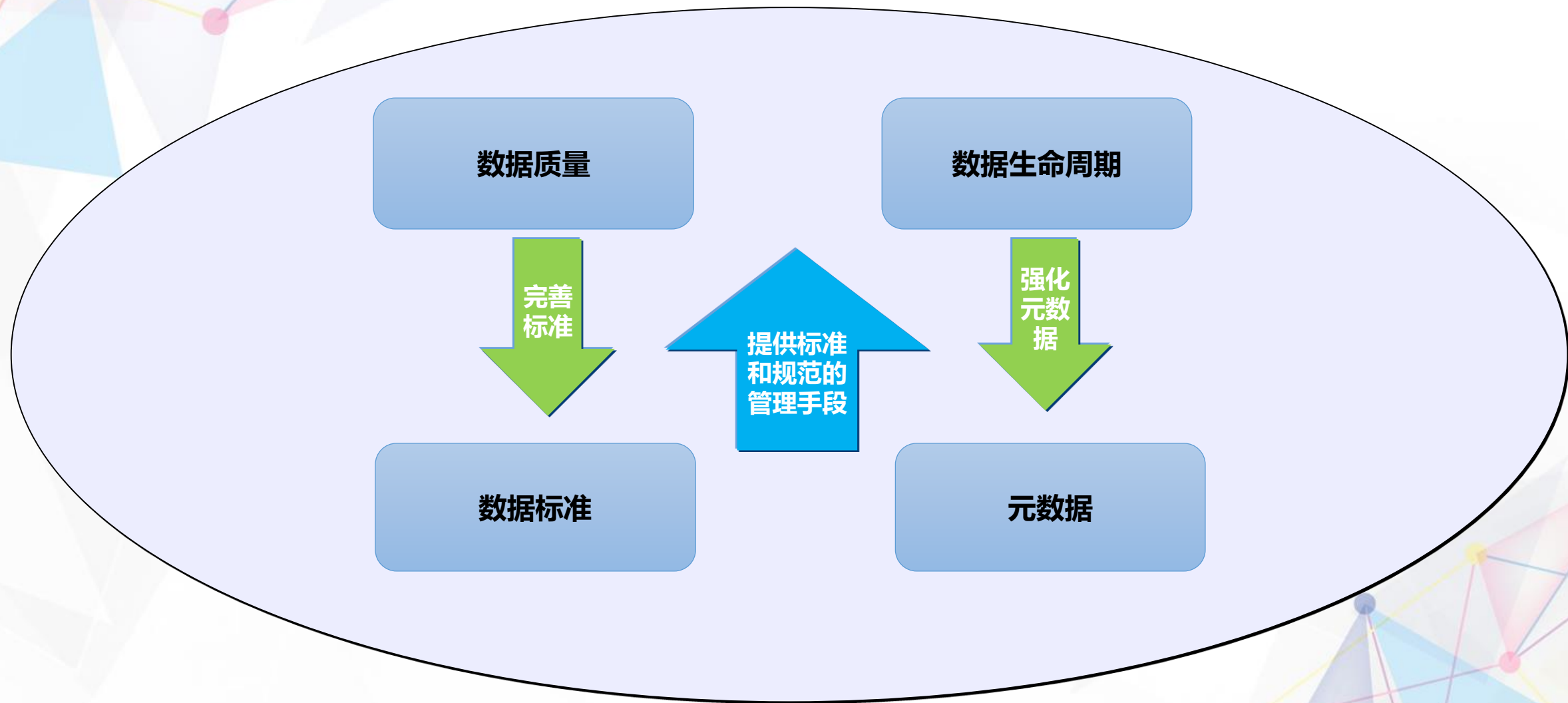
通过标注、文本挖掘等技术，建立非结构化数据的元数据，并与结构化数据整合，供分析使用



主数据定位为基础信息整合，为其他数据库供数



数据管控



数据质量管理



数据生命周期管理

数据创建

数据使用

数据归档

数据销毁

数据生命周期重点关注

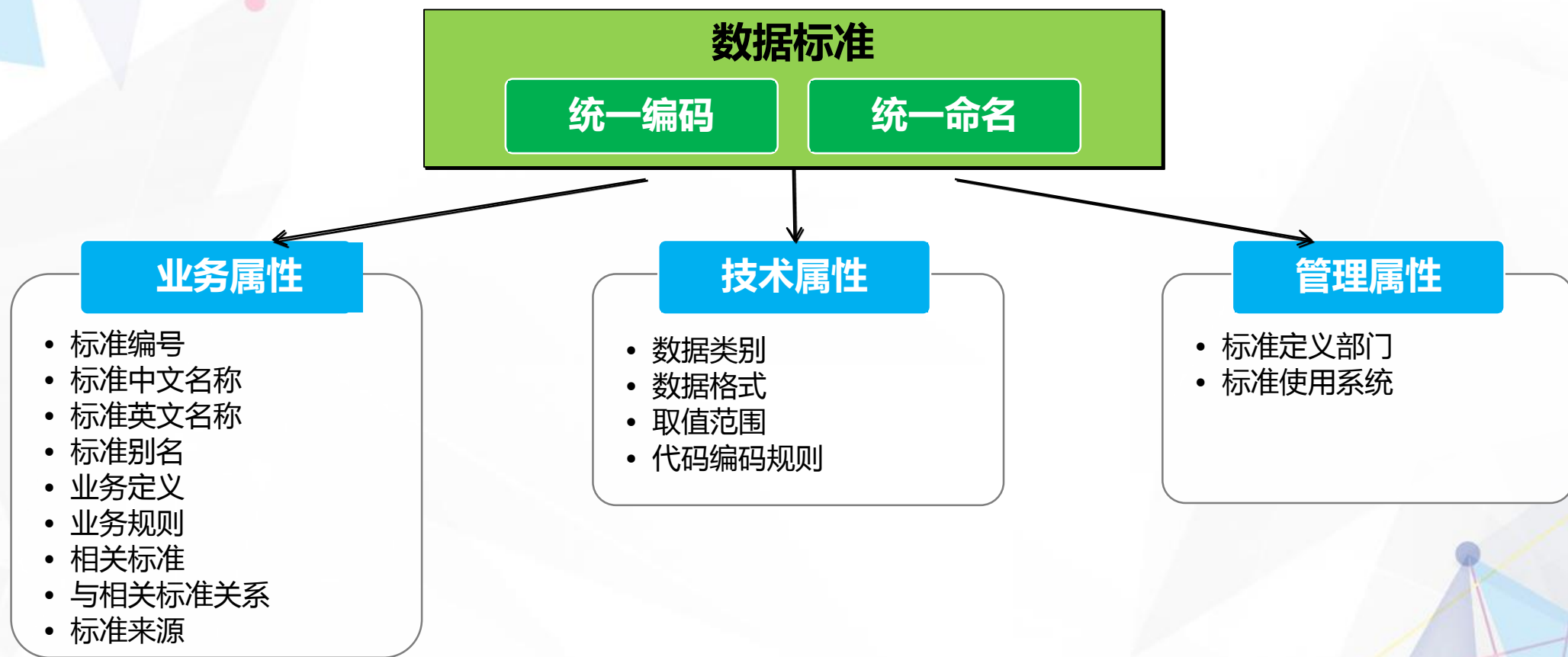
- 满足业务操作和管理分析
- 满足对历史数据查询的管理制度
- 满足审计管理

- 减少数据冗余，提高数据一致性
- 减少存储、硬件、运维等方面基础设施投入
- 提升应用系统性能，提高响应速度

数据标准管理

✓数据标准是企业所有系统都应该遵守的数据定义，实现对每个数据项的统一、规范性描述。

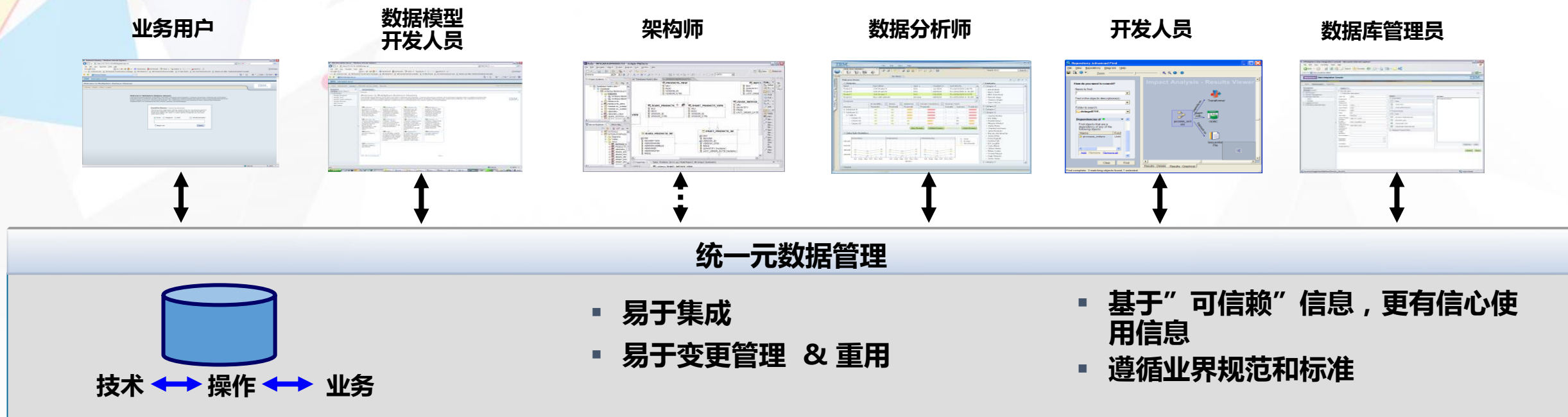
✓数据标准包括每个数据项的业务属性、技术属性和管控属性。



元数据管理

元数据是描述数据的数据

加强部门协作，让IT向业务看齐

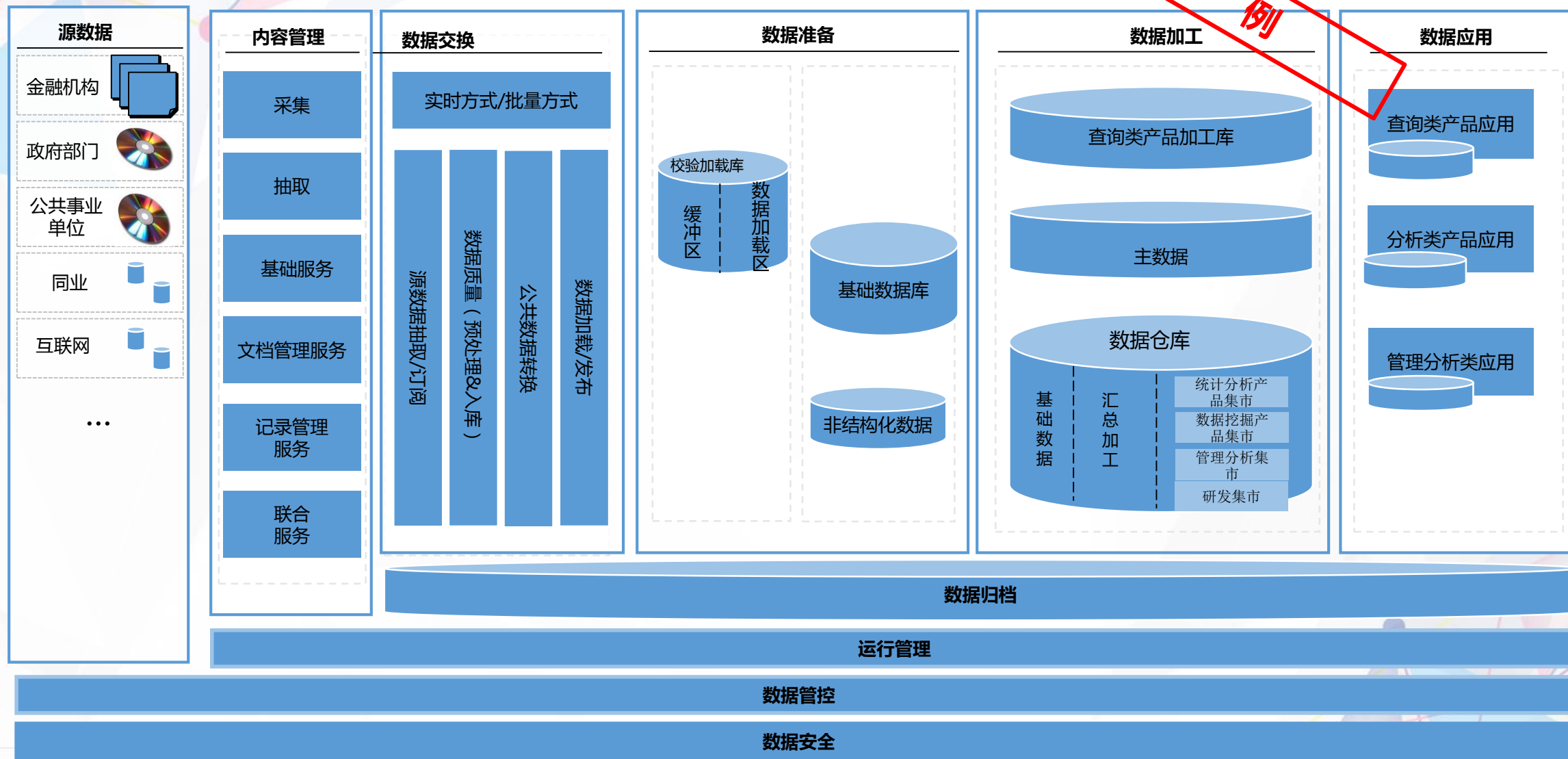


业务元数据：描述业务领域相关概念、关系和规则的数据。

技术元数据：描述系统中技术领域相关概念、关系和规则的数据。

管理元数据：管理元数据是描述管理领域相关概念、关系和规则的数据。

数据架构蓝图



示例

目录

□ 架构概览

□ 总体架构

□ 应用架构

□ 数据架构

□ 技术架构

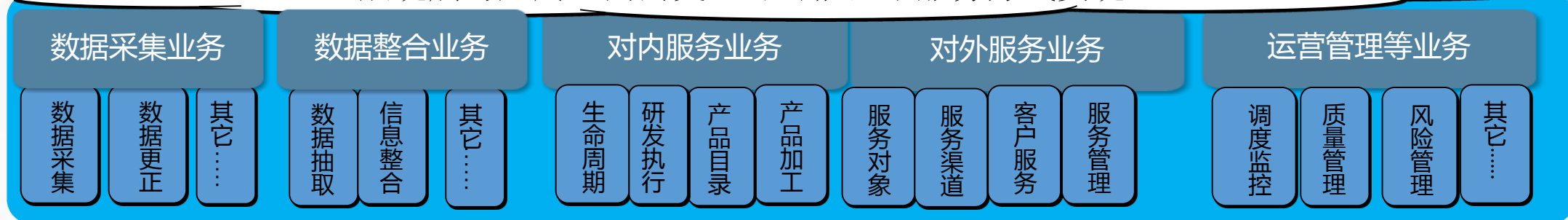
应用分层设计

界面
展现层

展现与接入
(门户、移动终端、统一认证管理)

展现层与应用之间的交互尽可能基于服务方式实现

应用层
服务和组件



集成层
流程

数据总线

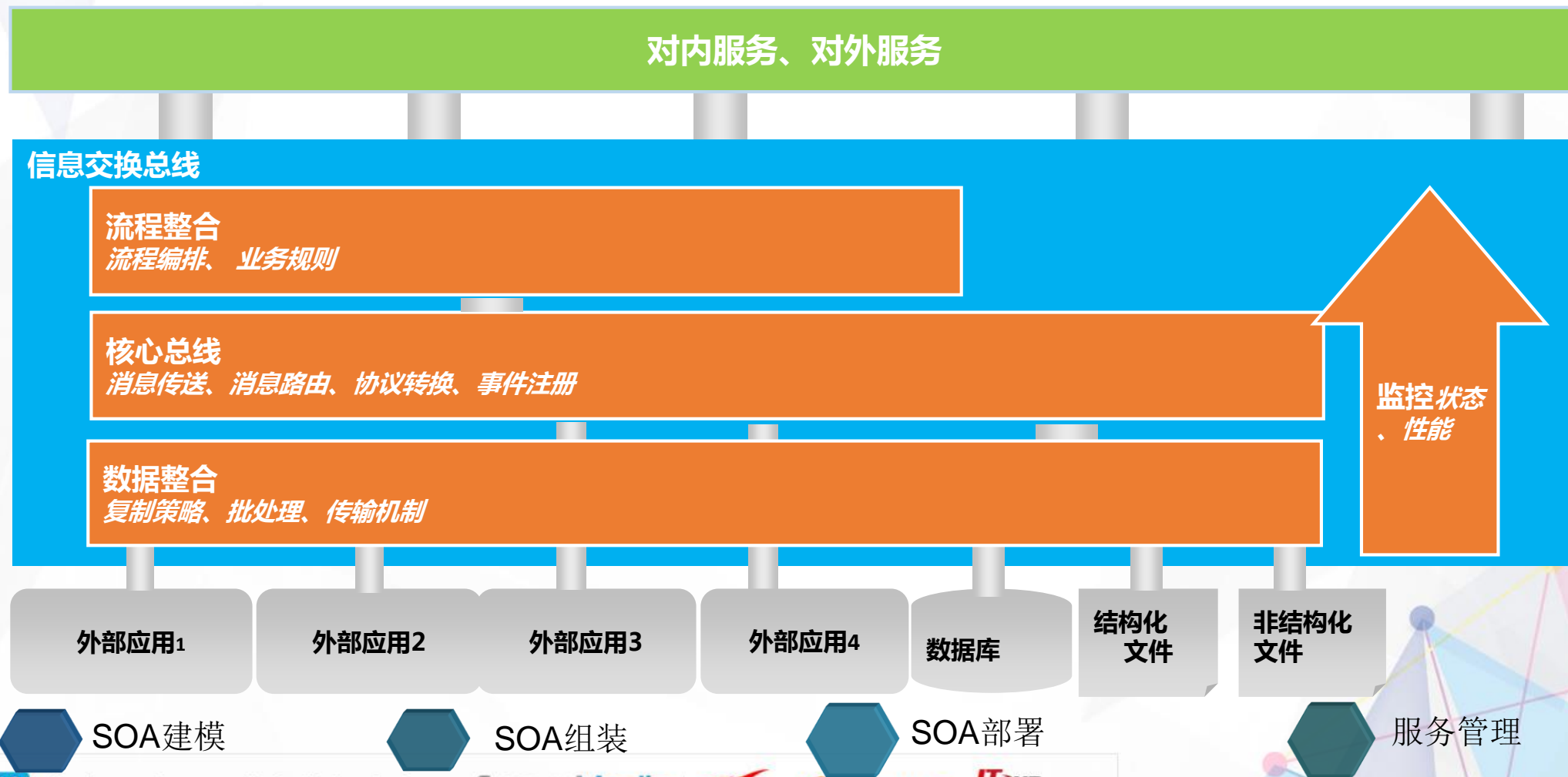
数据层



应用集成设计

将不同的功能模块定义为服务，并定义接口，采用统一的方式进行交互。

对外集成层



部署单元设计

- ✓ 部署单元是逻辑部署架构的最小构成单元；
- ✓ 组件按照展现、接口逻辑、应用逻辑、数据存放的区分部署单元

展现层

展现部署单元

关注接入渠道、用户数、访问量、访问时段、用户体验等需求

应用层

执行部署单元

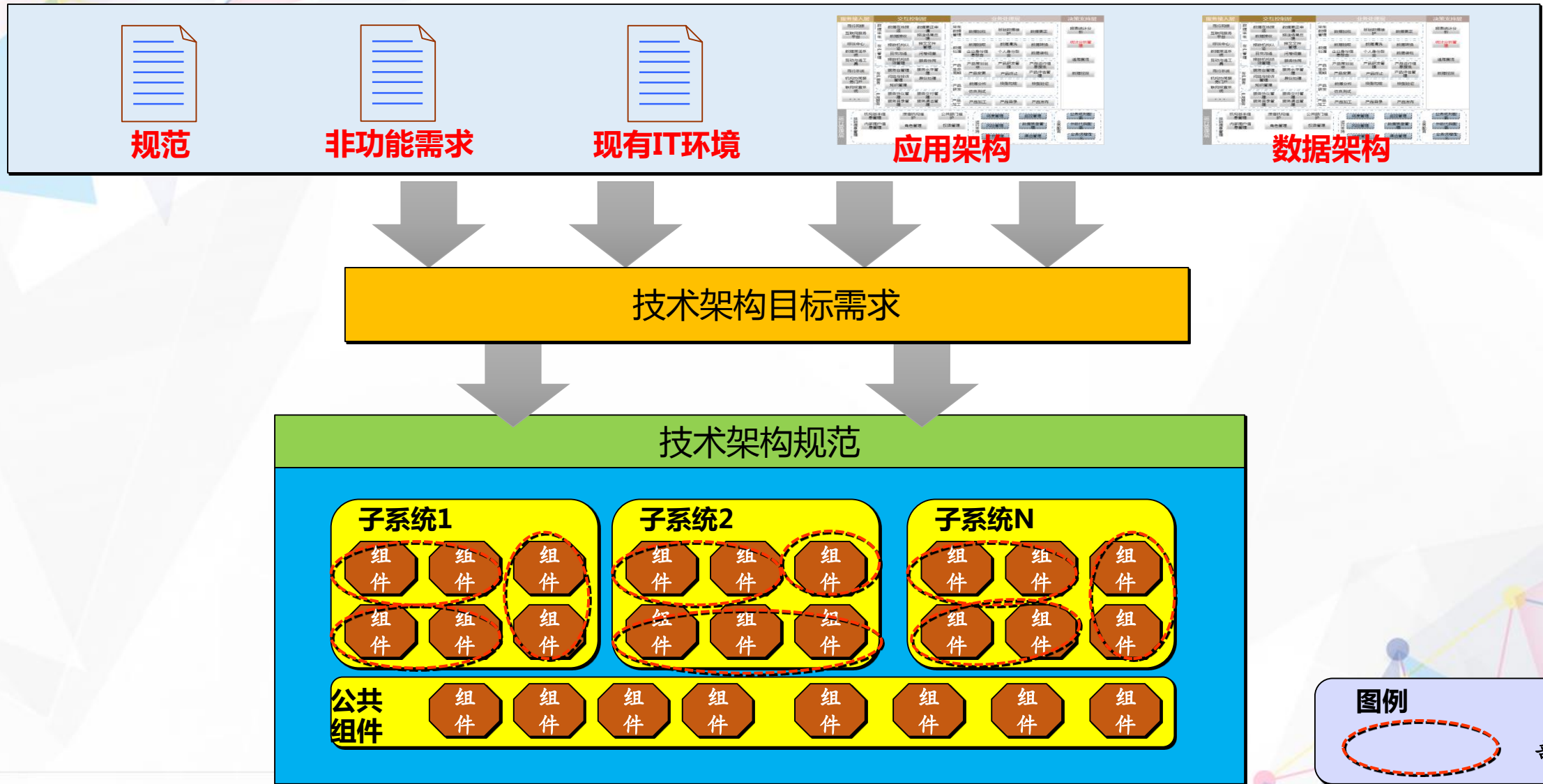
关注接入运行特征、响应时间、部署位置、可处理窗口等需求

数据层

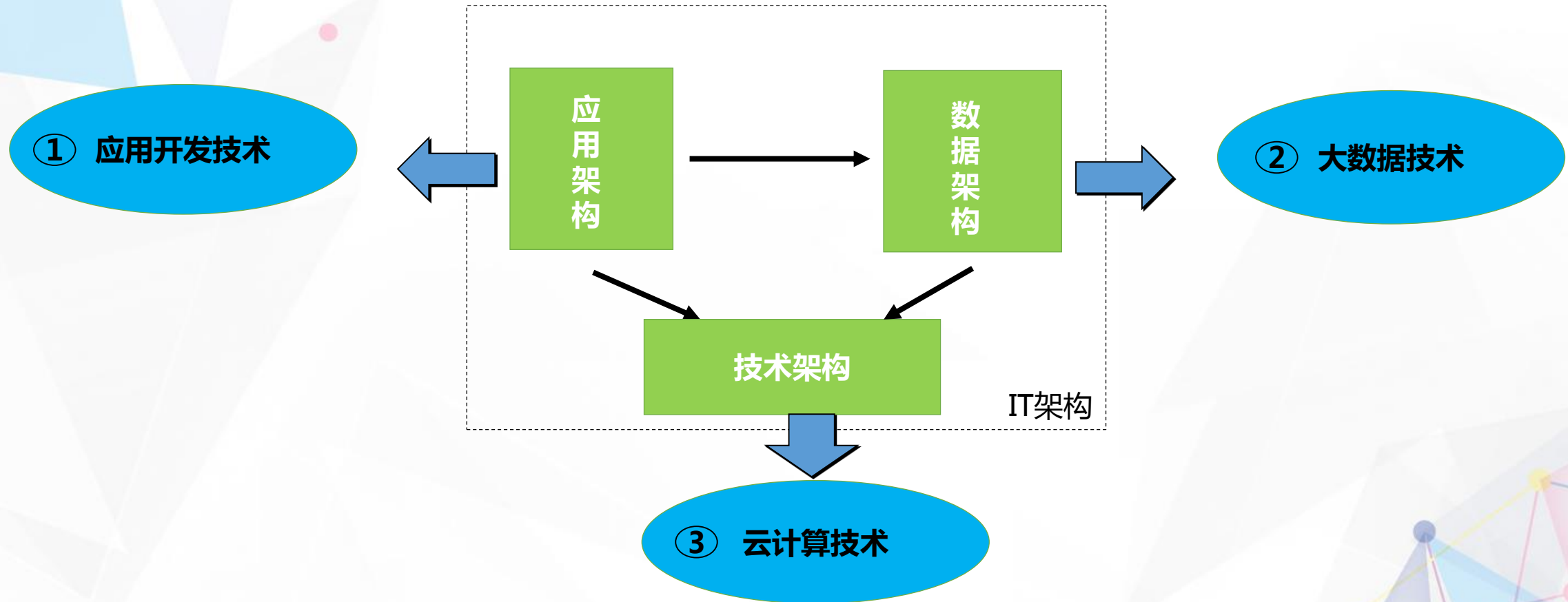
数据部署单元

关注数据类型、生命周期、规模、IO类型、处理方式、重要程度、安全要求、数据活跃度等需求

子系统划分



关键技术路线选择



开发技术选择

中后端开发
技术

J2EE, .NET, C++

SQL, Procedure

前端开发技
术

HTML5、JavaScript、
.NET、PHP、Python

业务逻辑的实现



应用层

数据库层



系统压力

系统扩展性



数据库技术选择

数据库技术	适用场景	主要技术	代表产品
键值型数据库	数据以字符串的形式存在，适用于数据的高并发读写，尤其适合实时数据分析场景	KVP	Riak, Redis
分布式数据库	为应对海量结构或半结构化数据访问I/O问题，数据跨列存储，方便数据压缩，适合批量数据处理、实时信息交换查询和分析	DFS/MapReduce	Hbase , Cassandra , Spark , Redis
图数据库	基于节点、关系、属性的图模型，适用于使用点、线、面描述图形关系的设计及存储	Node/Relationship /property	Neo4J,FlockDB
文档型数据库	海量数据高效率存储和访问，适用于生成大量的报告，报告随着变化能够动态进行内容整合	JavaScript/JSON	MongoDB、CouchDB
对象存储数据库	适用于通过对象方式存储数据，提供面向对象语言语法操作数据库	Hibernate/JPA/EJB	DB4o
XML数据库	适合半结构化数据以XML格式存储的复杂的数据结构数据集，支持XML查询语法如Xpath , XQuery	对象或表格映射	Berkeley DB XML BaseX
传统数据库	以结构化形式存在	共享存储存在	Oracle、DB2、SQL Server等

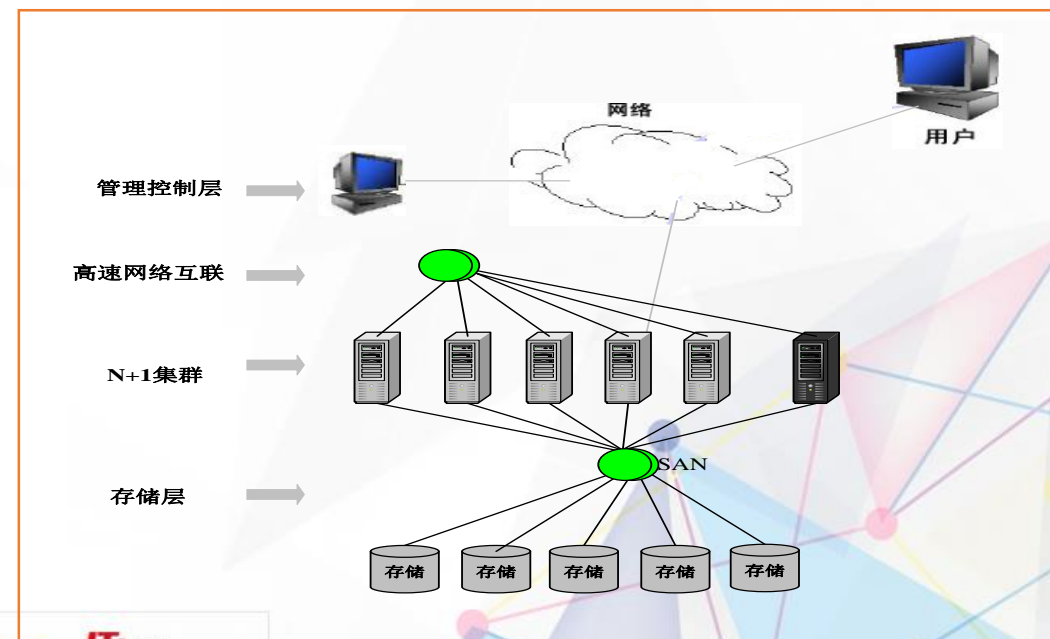
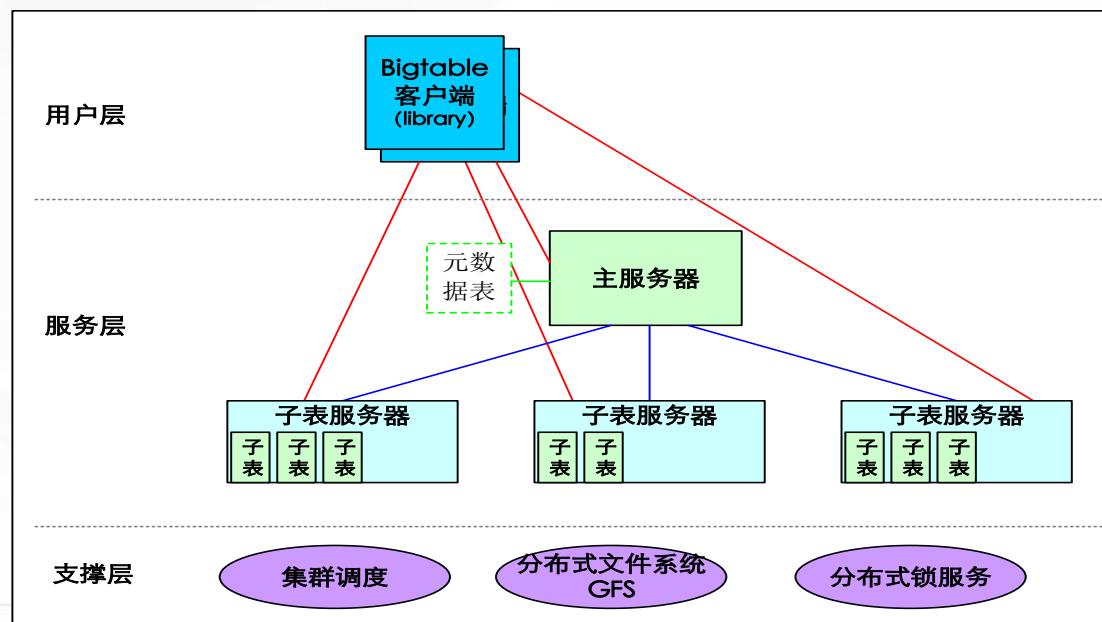
关系型与Nosql数据库整合使用

非关系型数据库

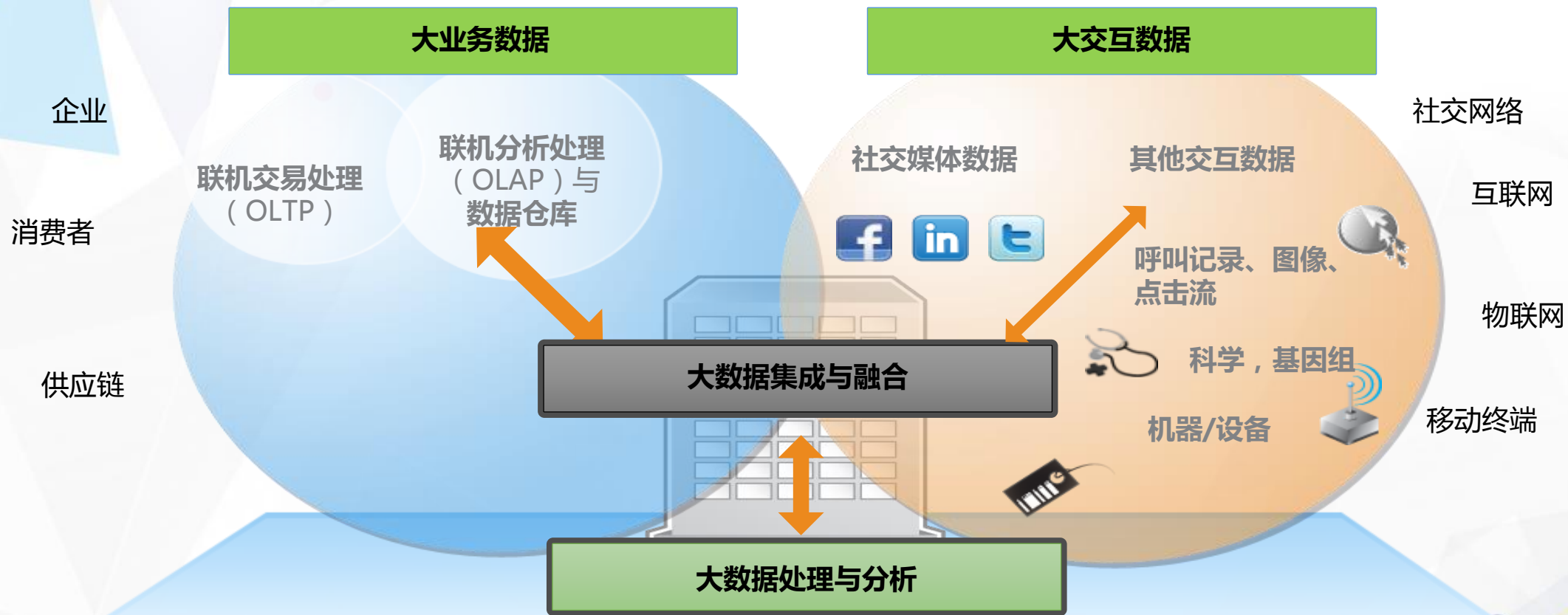
1. 适合批量追加写简单查询操作
2. 具有很强的弹性扩展能力
3. 极强的分布式处理能力
4. 数据弱一致性设计
5. 非标准化的接口
6. 适合大数据量的弱关联查询

关系型数据库

1. 通用设计，但也带来性能的限制
2. 弹性扩展能力一般
3. 分布式能力一般
4. 数据强一致性设计
5. 标准的数据访问接口
6. 适合关联操作性较强的查询

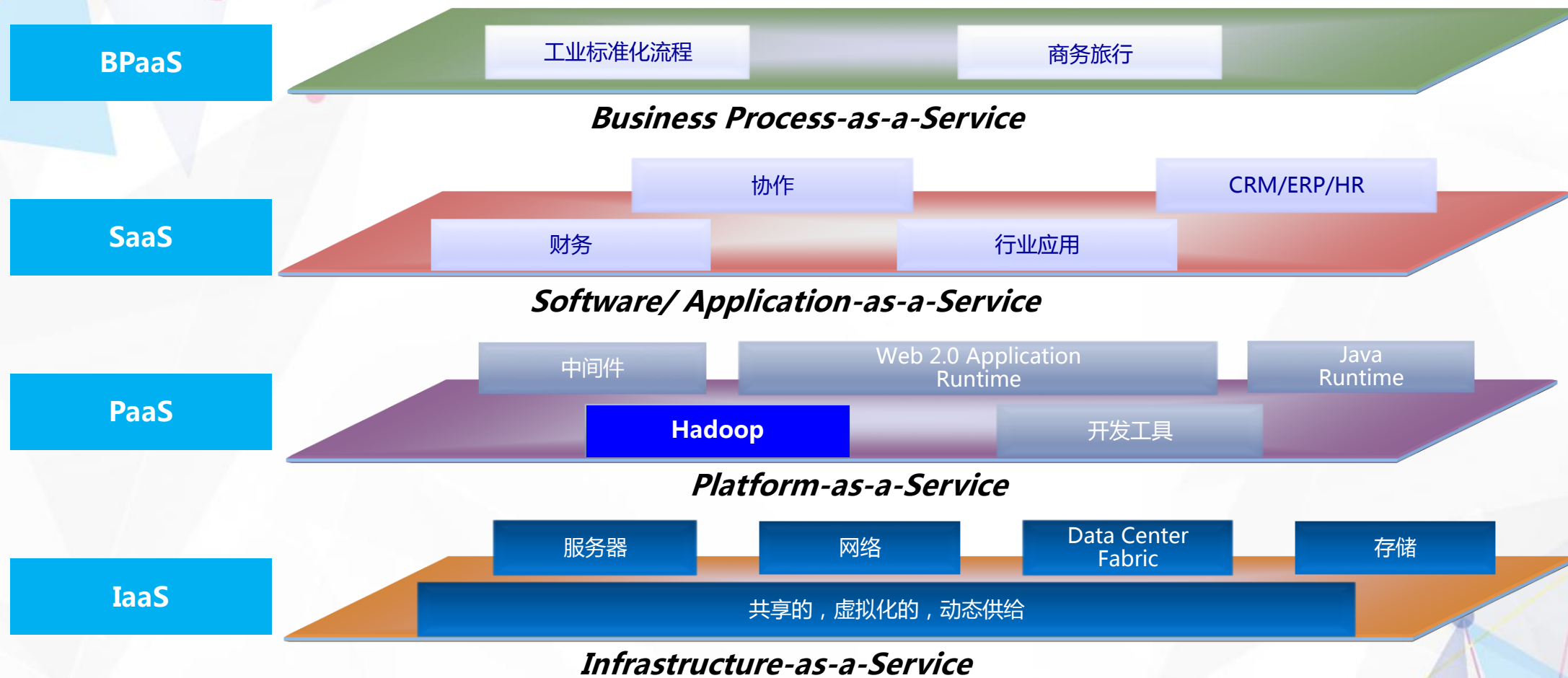


大数据技术



- 大交易数据：各种交易数据，包括来自电商的数据，包括B2B、B2C、C2C、团购等
- 大交互数据：来自社交网络的数据，SNS、微博等

云计算技术



DevOPS

- ✓ DevOPS是一个跨全生命周期的理念，强调自动化，需要以虚拟化为基础。
- ✓ 通过持续集成、交付、反馈与优化进行端到端整合，进行跨团队、跨系统协作。



更加适用于子系统的研发与交付

总结

□ 架构概览

□ 总体架构

□ 应用架构

□ 数据架构

□ 技术架构

总结

何谓架构创新：

- ✓ 不断的变化和扩展的业务形态，在一段时期内要有与之适应的可落地架构存在，业务与技术高度融合。
- ✓ 新技术的更迭在一段时期内对整体架构影响甚微。
- ✓ 架构是一项整体工程，而非仅仅是IT部门的工作，是一项长期工作，能够带来长远利益工作。

THANKS

SequeMedia
盛拓传媒

IT168.com
专注导购16年

ChinaUnix.net

ITPUB
www.itpub.net