美团配送数据治理实践

今天，数据资产日益成为企业的核心竞争力。但如果企业在走向数字化过程中遗忘了数据治理，可能再多的投入都会变成一种“徒劳”。

今天的文章来自美团配送数据治理团队，他们从数据治理的概念、达成的目标、何时启动数据治理以及如何开展数据治理等几个维度进行阐述，全面、系统地介绍了美团配送技术团队在数据治理过程中所进行的一些探索和实践。

背景

大数据时代的到来，让越来越多的企业看到了数据资产的价值。将数据视为企业的重要资产，已经成为业界的一种共识，企业也在快速探索应用场景和商业模式，并开始建设技术平台。

但这里要特别强调一下，如果在大数据“拼图”中遗忘了数据治理，可能再多的技术投入也是一种徒劳。因为没有数据治理这一环节，其带来后果往往是：随处可见的数据不统一，难以提升的数据质量，难以完成的模型梳理，难以保障的数据安全等等，源源不断的基础性数据问题会进一步产生，进而导致数据建设难以真正发挥其商业价值。

因此，消除数据的不一致性，建立规范的数据标准，提高数据治理能力，实现数据安全共享，并能够将数据作为企业的宝贵资产应用于业务、管理、战略决策中，发挥数据资产价值变得尤为迫切和重要，数据治理呼之欲出。本文将介绍美团配送技术团队在数据治理方面的一些探索和实践，希望能够对大家有所启发和帮助。

1. 如何理解数据治理

数据治理，从严格的定义来讲是对组织的大数据管理并利用其进行评估、指导和监督的体系框架。企业通过制定战略方针、建立组织架构、明确职责分工等，实现数据的风险可控、安全合规、绩效提升和价值创造，并提供创新的大数据服务。从个人实践的层面来讲，数据治理是对存量数据治理和增量数据管控的一个过程，对存量数据实现由乱到治、建章立制，对增量数据实现严格把控、行不逾矩的约束。

2. 要达成的目标

数据治理本身并不是目的，它只是实现组织战略目标的一个手段而已。从组织职能和体量大小方面来看，不同类型组织的数据治理目标大不相同，而基于目前美团配送数据团队所处的组织职能和发展阶段来说，我们希望通过数据治理解决数据生产、管理和使用过程中遇到的问题，完善已有的生产管理流程规范，保障数据安全和数据一致性，从而促进数据在组织内无障碍地进行共享。

3. 何时进行数据治理

找准数据治理的切入点，是关乎数据治理成败的关键。很多同学会问，如果将数仓建设分为数仓雏形阶段、数仓迭代阶段和能力沉淀阶段，数据治理应该在哪个阶段切入为宜呢？其实，我们不该把数据治理看作是一个阶段性的项目，它应该是一个贯彻数据建设各阶段的长期工程，只是在不同阶段根据业务特点和技术特点其覆盖的范围和关注的目标有所不同而已。

在数仓雏形阶段，也就是美团配送业务刚成立时，在该阶段中业务有两个特点：第一，重规模、快扩张；第二，业务变化快，数据需求多。为了快速响应业务的需求，并能够保障数据交付结果的准确性，我们主要进行技术规范和指标口径的治理，在规范治理方面，通过制定一系列研发规范来保障研发质量，并在实际建模过程中不断迭代和完善我们的研发质量。在指标治理方面，我们对存量指标口径进行梳理，从而确保指标口径对外输出一致。

在数仓迭代阶段，我们希望通过架构治理改变前期开发的“烟囱式”模型，消除冗余，提升数据一致性。并且随着数仓中管理的数据越多，数据安全和成本问题也变得越发重要。所以在该阶段，我们在产研层面逐步开展架构治理、资源治理和安全治理。在架构治理方面，我们明确了数仓中各层和各主题的职责和边界，构建一致的基础数据核心模型，并制定一系列的指标定义规范来确保指标的清晰定义，并基于业务迭代来不断完善和迭代相应的模型和规范。在资源治理方面，我们通过对不同层级的数据采用不同生命周期管理策略，确保用最少的存储成本来满足最大的业务需求。在安全治理方面，我们通过制定一系列的数据安全规范来确保数据的使用安全。

在能力沉淀阶段，我们基于前两个阶段所做的业务和技术沉淀，将前期一系列规范形成标准，从业务到产研，自上而下地推动数据治理，并通过建立相应的组织、流程和制度来保障标准在该阶段的全面落地实施，并通过建设数据治理平台来辅助更高质量地执行标准。

4. 如何开展数据治理

从大的阶段来看，数据治理主要分为存量数据“由乱到治”的阶段，以及增量数据严格按照规章制度实施确保“行不逾矩”的运营阶段。在“由乱到治”的过程中，我们需要沉淀出规章制度、标准规范，以及辅以规章制度标准规范实施的工具和组织。在增量数据的运营阶段，我们主要靠对应的组织确保规章制度的落实，通过审计定期考察实施效果，并在长期的运营中不断完善规章制度。在实现存量数据“由乱到治”的阶段，我们主要采取了“两步走”策略，具体执行策略如下所示。

4.1 定标准，提质量

第一步，主要围绕着业务标准、技术标准、数据安全标准和资源管理标准进行展开。通过业务标准，指导一线团队完成指标的规范定义，最终达成业务对指标认知一致性这一目标；然后通过技术标准来指导

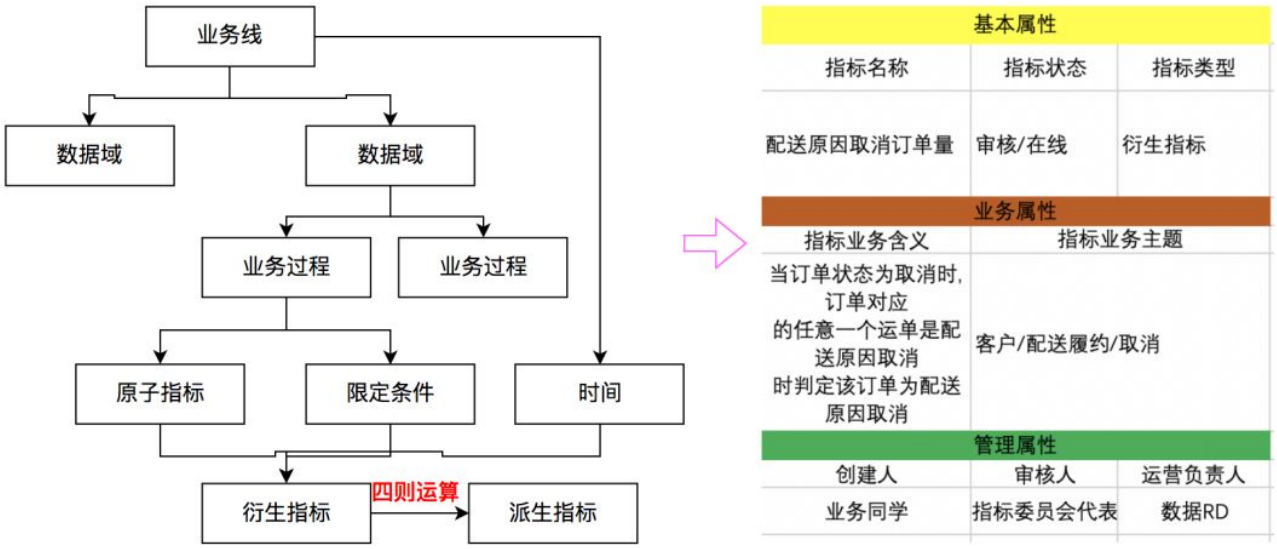
研发同学规范建模，从技术层面解决模型扩展性差、冗余多等问题并保障数据一致性；通过安全标准来指导我们加强数据的安全管控，确保数据拿不走、走不脱，针对敏感数据，用户看不懂；通过资源管理标准的制定，帮助我们在事前做好资源预算，在事中做好资源管理，在事后做好账单管理。

4.1.1 业务标准

业务标准主要是指标的管理和运营标准，我们主要解决三个问题：指标由谁来定义，指标该如何定义，指标该如何运营。基于这三个问题，我们同时提出了三条原则：

- 业务团队负责指标的定义。
- 产研商分负责给出指标定义标准和辅助工具，辅助业务团队完成指标的规范定义，达成指标认知一致性这一目标。
- 最后由指标管理委员会负责指标的管理与运营，保障指标从创建、审核、上线以及到最后消亡的整个生命周期的运营。

为统一指标的定义，我们将指标分为原子指标、衍生指标和派生指标，原子指标通过限定条件和时间的限定生成衍生指标。衍生指标间的“四则混合运算”构成了派生指标。我们不但制定了指标的标准定义，还对其做了准确的资产归属，一个指标出自一个具体的业务过程，一个业务过程归属于不同的数据域，多个数据域构成了美团配送业务线下的分析场景，如下图所示：



指标定义标准

4.1.2 技术标准

这里所说的技术标准，主要是针对数据RD提出的建模标准和数据生产规范，通过建模标准来明确数仓分层架构，并清晰定义每一层的边界与职责，采用维度建模的设计理念。我们的整个仓库架构分为四层：操作层、基础事实层、中间层和应用层，并在每一层同步制定对应的建模规范，如下图所示：

分层架构	层次职责	建模标准
中间层	存放公共指标汇总数据	1、细分为B2多维汇总数据和B1单维宽表模型 2、只做简单汇总，禁止任何逻辑封装
基础事实层	存放明细事实数据和维表数据	1、快照表建设大而全 2、操作层导快照层数据清洗过程中，做到过滤逻辑配置灵活 3、快照层实现敏感数据脱敏 4、通过衍生事实实现业务口径标签化，确保数据一致性
操作层	操作系统数据几乎无处理的同步到该层存储	1、与业务系统永久一致 2、基于业务场景分别建设全量表和增量表

数仓架构以及建模标准

除了建模标准外，我们还制定了涵盖从生产到运维环节的生产规范以保障模型的质量，主要包括上线前的模型评审、生产过程中的完成元数据配置、DQC、SLA和生命周期设置以及上线后的日常运维机制等等。尤其针对元数据管理和生命周期管理，我们分别制定了仓库每一层元数据维护规范和生命周期管理规范，其中元数据管理规范，是依据数仓各层级中各种类型表的建模标准来制定，需要做到规范命名，明确数据归属，并打通业务元数据和技术元数据之间的关系。而生命周期管理规范，是依据配送业务特点和数仓各层级现状来制定的，如下表所示：

数据仓库表分类	主题/实体	业务过程	主键	维度	维度属性	指标	是否核心模型
分析宽表	需要(跨主题)	X	需要	需要 指定对应的维度表	需要 有表的对应Code表，没有表的枚举值及说明，采用键值对方式	需要	是
汇总聚合表	需要(单一主题)	X		需要 指定对应的维度表		需要	
事实表(主题核心表)	需要(单一主题)	需要	需要	需要 指定对应的维度表	需要 有表的对应Code表，没有表的枚举值及说明，采用键值对方式	不需要	是
事实表(非核心表)	需要(单一主题)	需要	需要	需要 指定对应的维度表	需要 有表的对应Code表，没有表的枚举值及说明，采用键值对方式	不需要	否
维度表			需要	需要 指定对应的维度表	需要 有表的对应Code表，没有表的枚举值及说明，采用键值对方式	X	是
快照表	需要(单一主题)	需要	需要	X	X	X	
三方解耦表	需要(有可能跨主题)			需要 指定对应的维度表	需要	需要	

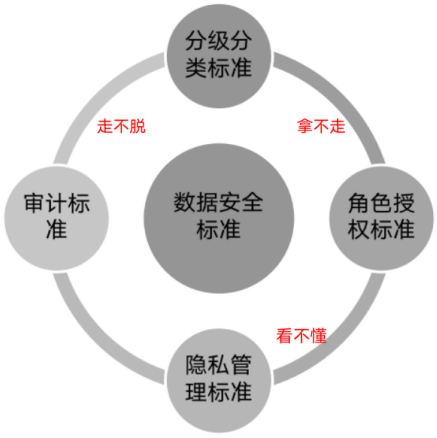
仓库各层元数据管理标准

仓库层级	数据类别	生命周期管理策略	存储方式
A	原始业务数据	永久保留策略	HDFS(ORC格式存储)
	原始LOG数据	周期性删除策略（保留1.5年）	RAID存储
B3/B1	基础明细数据/单维宽表模型	永久保留策略	HDFS(ORC格式存储)
B2/C	汇总表数据/应用表数据	周期性删除策略（2年）	HDFS(ORC格式存储)

仓库各层生命周期管理策略

4.1.3 安全标准

围绕数据安全标准，首先要有数据的分级、分类标准，确保数据在上线前有着准确的密级。第二，针对数据使用方，要有明确的角色授权标准，通过分级分类和角色授权，来保障重要数据拿不走。第三，针对敏感数据，要有隐私管理标准，保障敏感数据的安全存储，即使未授权用户绕过权限管理拿到敏感数据，也要确保其看不懂。第四，通过制定审计标准，为后续的审计提供审计依据，确保数据走不脱。

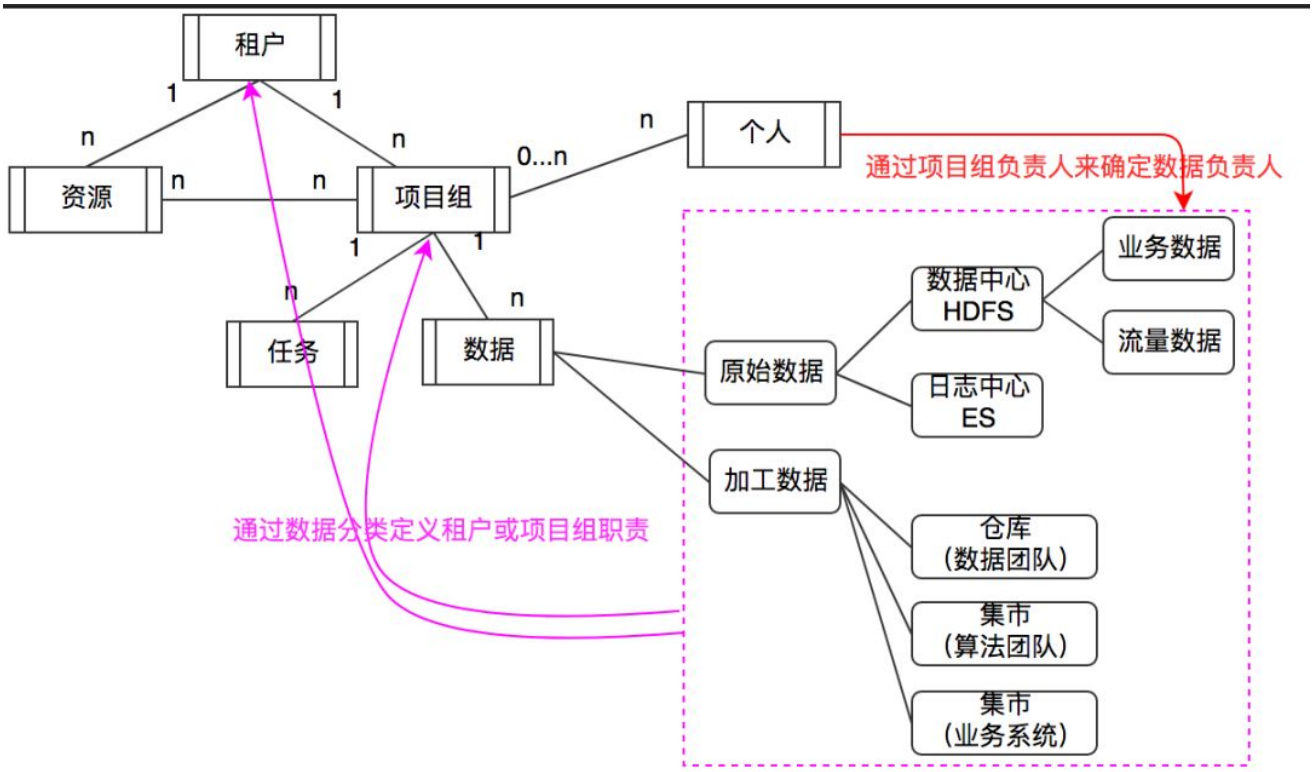


安全标准建设

4.1.4 资源管理标准

在资源管理方面，配送技术工程部已经对资源管理涉及的内容进行了合理抽象和准确定义，抽象出租户、资源和项目组等概念。不管是后续的资源预算还是资源管理，我们都需要基于租户和项目组来进行运营，因此，对于业务团队而言，我们只需要将租户和项目组特定职能划分清楚，然后根据不同的职能归属我们的资产，并分配生产该资产所需要的资源。为了方便后续的运营，我们对每个租户和项目组分配确定了责任人，由责任人对运营结果负责。

对业务部门来说，资源管理的关键是对数据资产做清晰的分类，基于数据的分类划分不同的租户和项目组，将数据和租户、项目组实现一一映射。由于租户和项目组都有特定的责任人对其负责，因此，我们通过这种映射关系，不仅实现了资产的隔离，还实现了资产确权（项目组负责人同时对资产负责和运营）。我们整体将数据分为两大类，一是原始数据，包括流到数据中心的数据和日志中心的数据，针对流入数据中心的数据，根据其产生的方式不同，又进一步分为业务数据和流量数据。二是加工数据，对应着数据团队的仓库建设和其他团队的集市建设。基于上述的描述，针对资源管理，我们做了如下划分和确权：



资源划分与管理

4.2 重实施，保落实

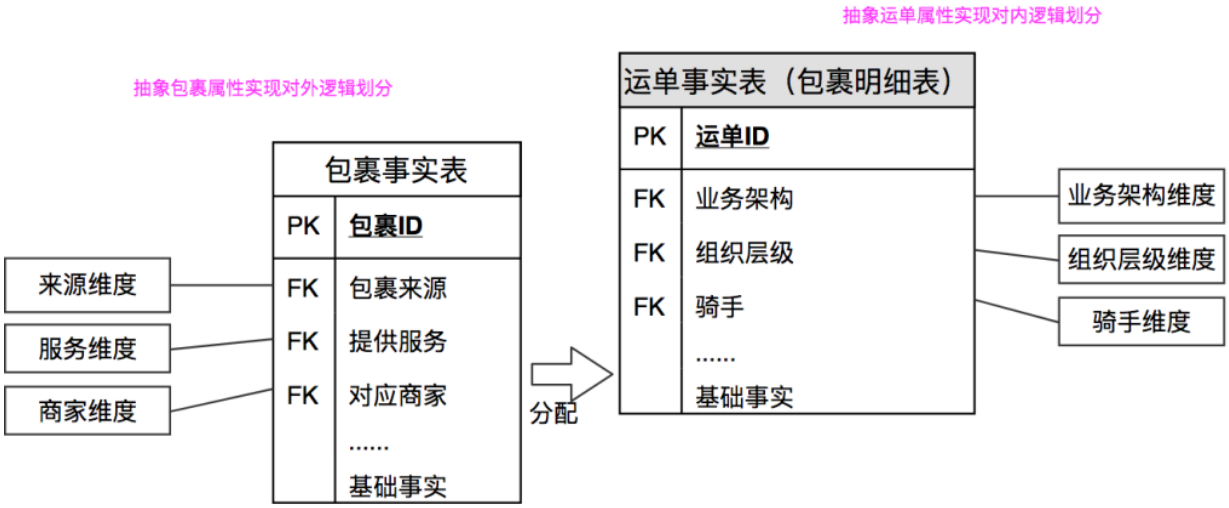
第二步，落实第一步的标准，完成数据治理第一阶段的目标，实现存量数据“由乱到治”，并完成相应组织和工具的建设，为实现第二阶段“行不逾矩”这一目标提供工具和组织能力。在此过程中，主要分成三个方面的治理工作：第一，架构模型“由乱到治”的治理，消除模型冗余、跨层引用和链路过长等问题，在架构上保证模型的稳定性和数据一致性；第二，元数据“由乱到治”的治理，实现指标的标准定义、技术元数据的完整采集并建立指标与表、字段的映射关系，彻底解决指标认知一致性，以及用户在使用数据过程中的“找数难”等问题；第三，围绕着隐私安全和共享安全加强数据的安全管控来实现数据走不脱、拿不走，以及隐私数据看不懂这一目标。

4.2.1 架构治理

总结起来，架构方面的治理主要是解决两个问题：第一，模型的灵活性，避免需求变更和业务迭代对核心模型带来的冲击，让RD深陷无休止的需求迭代中；第二，数据一致性，消除因模型冗余、跨层引用等问题带来的数据一致性问题。

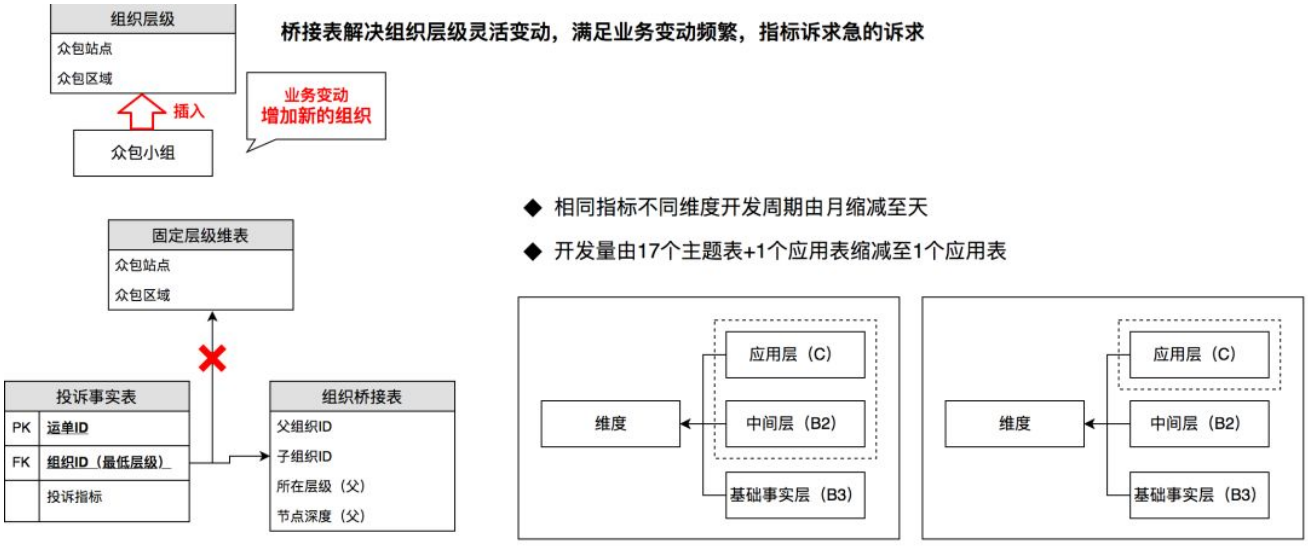
模型灵活性

配送解决的是效率、成本和体验三者之间的平衡问题，即在满足一定用户体验的条件下，如何提升骑手配送效率，服务更多的商家，以及如何管控骑手，降低配送成本。抽象到数据层面，基本上反映为上游包裹来源的变化、配送对外提供服务的变化以及对内业务管控的变化。为屏蔽业务迭代给核心模型带来的冲击，我们通过对外封装包裹属性和对内封装运单属性，抽象出包裹来源、提供服务、业务架构等一致性维度，任何业务迭代在数据层面只涉及维度的调整，大大降低了对核心模型冲击和“烟囱式”数据建设问题（新来一个业务，就拉起一个分支进行建设）。



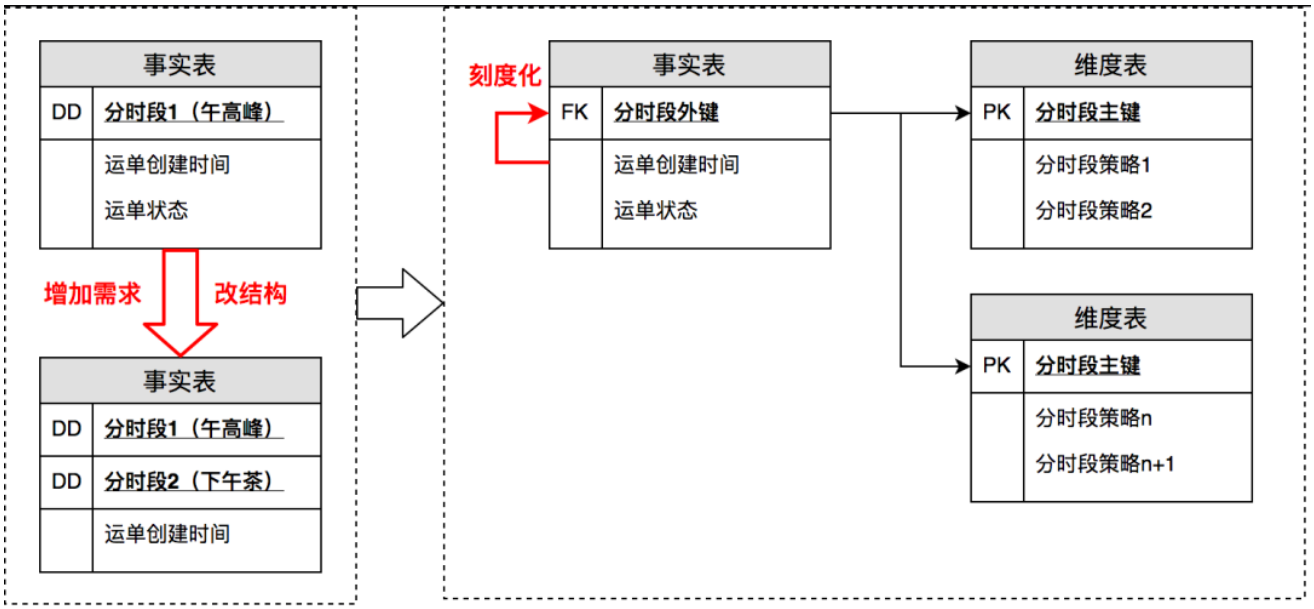
包裹事实分配到运单明细构造单一运单模型

配送指标体系建设的一个重点就是要输出各组织层级的规模、体验和效率指标，实现对运力的有效管控，运力所属组织的层级关系会随业务的迭代而不断变化。为了适应这种变化，避免仅仅因增加维度带来中间层数据的重复建设，我们将组织层级维表由固定层级建模方式调整为桥接表的方式来自适配组织层级变化，从而实现了中间层模型可以自动适配组织层级的变化，能自动产生新维度的指标。如下图所示：



桥接表自适配组织层级灵活变动

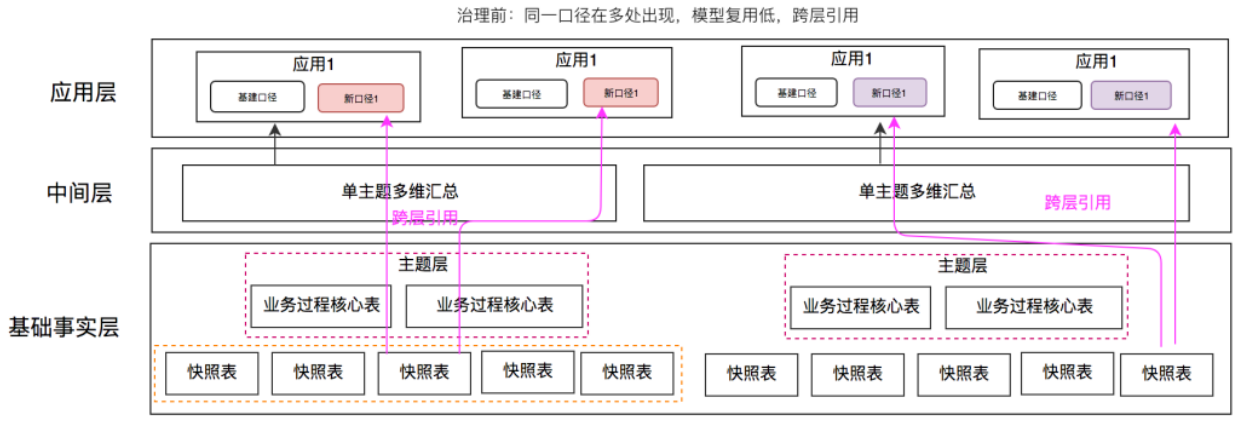
在精细化分析的场景下，业务会有分时段、分距离段以及分价格段的数据分析诉求。我们以分时段为例，有晚高峰、午高峰、下午茶等不同的分时段，不同的业务方对同一个时段的定义口径不同，即不同的业务方会有不同的分时段策略。为解决该场景下的分析诉求，我们在事实表中消除退化维度，将原来封装到事实表的时段逻辑迁移到维度表中，并将事实表中的时间进行按特定的间隔进行刻度化作为维表中的主键，将该主键作为事实表的外键。这样，针对业务不同的时间策略需要，我们就可以在维表中进行配置，避免了重复调整事实表和反复刷数的问题。即通过将时间、价格、距离事实刻度化，实现灵活维度分析。如下图所示：



通过将时间刻度化，实现灵活分析

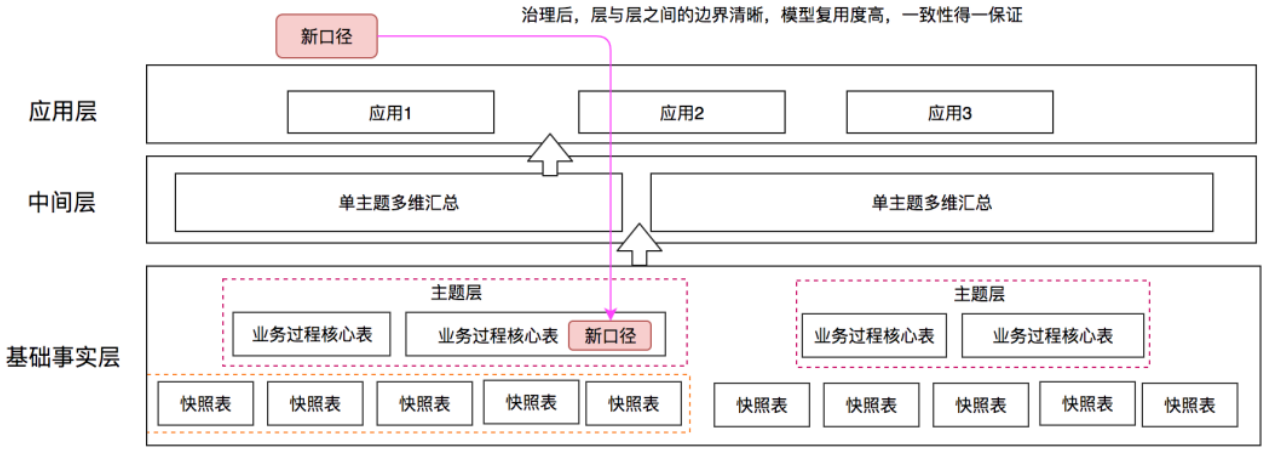
数据一致性

数据一致性得不到保障的一个根本原因，是在建模的过程中没有实现业务口径标签化，并将业务口径下沉到主题层。很多同学在基于需求进行开发时，为实现方便，将新指标口径通过“Case When”的方式在应用层和中间层进行封装开发，主题层建设不能随着业务的迭代不断完善，RD在开发过程中会直接引用仓库的快照表在中间层或应用层完成需求开发。久而久之，就会造成数据复用性低下，相同指标的口径封装在不同的应用表来满足不同报表的需求，但随着应用的增多，很难保障相同指标在不用应用表封装逻辑的一致性，数据一致性难以得到保障，同时这种方式还带来两个严重后果：第一，跨层引用增多，数据复用性低下，造成计算和存储成本的浪费；第二，一旦指标口径发生变化，将是一个“灾难”，不仅影响评估是一个问题，而且涉及该指标的应用层逻辑调整对RD来说也是一个巨大的挑战。



治理前模型架构

因此，我们在“由乱到治”的治理过程中，以衍生事实的方式实现业务口径标签化，将业务逻辑下沉到主题层，消除跨层引用和模型冗余等问题，从技术层面保障数据一致性是该阶段架构治理的重点。我们在业务上，已经划分了严格的数据域和业务过程，在主题建设层面，将业务划分的数据域作为我们的主题，并基于业务过程进行维度建模，将属于该业务过程的指标口径封装在对应业务过程下的衍生事实中。

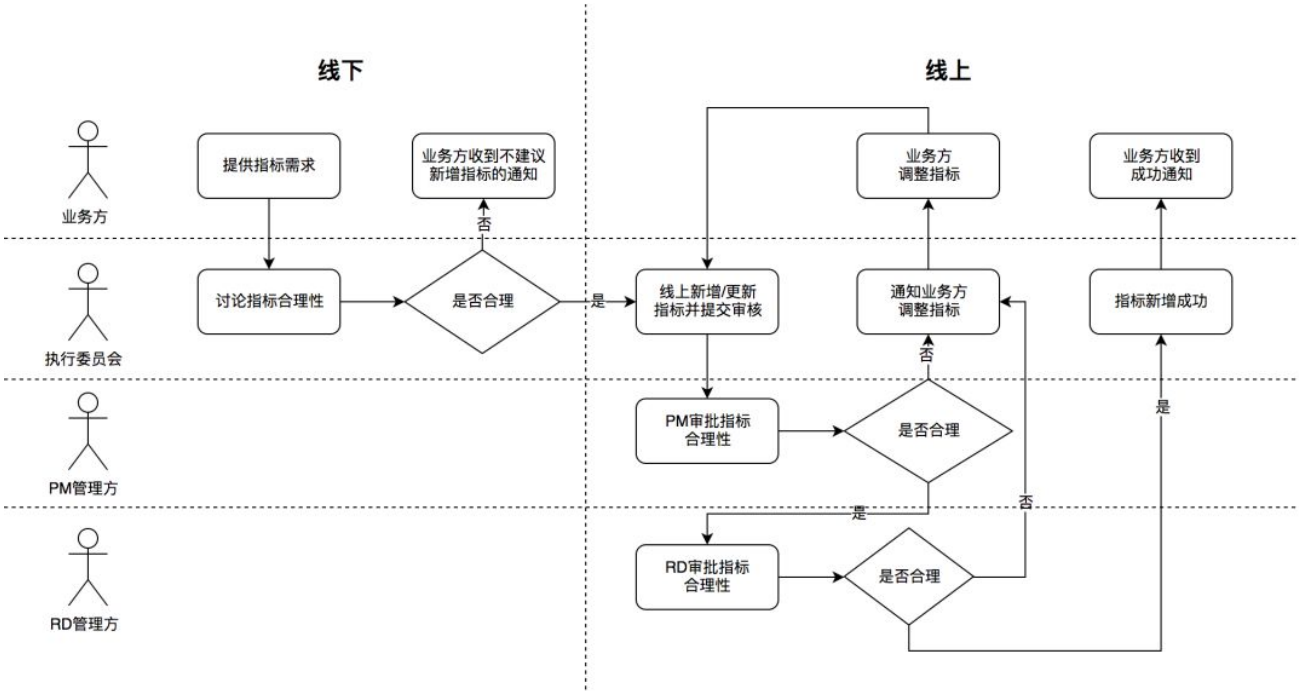


治理后模型架构

4.2.2 元数据治理

元数据治理主要解决三个问题：首先，通过建立相应的组织、流程和工具，推动业务标准的落地实施，实现指标的规范定义，消除指标认知的歧义；其次，基于业务现状和未来的演进方式，对业务模型进行抽象，制定清晰的主题、业务过程和分析方向，构建完备的技术元数据，对物理模型进行准确完善的描述，并打通技术元数据与业务元数据的关系，对物理模型进行完备的刻画；第三，通过元数据建设，为使用数据提效，解决“找数、理解数、评估”难题以及“取数、数据可视化”等难题。

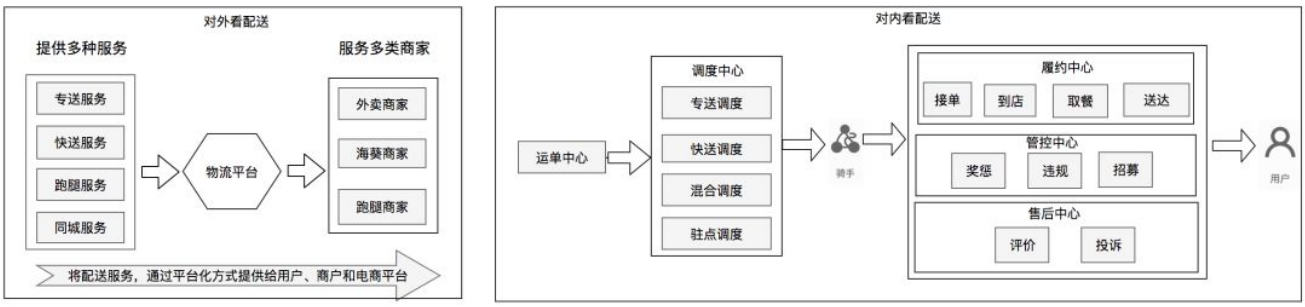
首先，为保障业务标准的顺利实施，实现业务对指标认知一致性这一目标。我们协同产研、商分、业务部门推动成立了度量衡委员会，并建立起指标运营机制，通过组织保障来实现指标运营按照规范的标准和流程实施。如下图所示：



指标注册流程

其次，基于配送业务的现状和未来演进方式，我们进行了高度的业务抽象，完成了主题、业务过程和分析方向等元数据内容的建设。配送即物流，通过线上系统和线下运营，我们将用户的配送需求和美团的

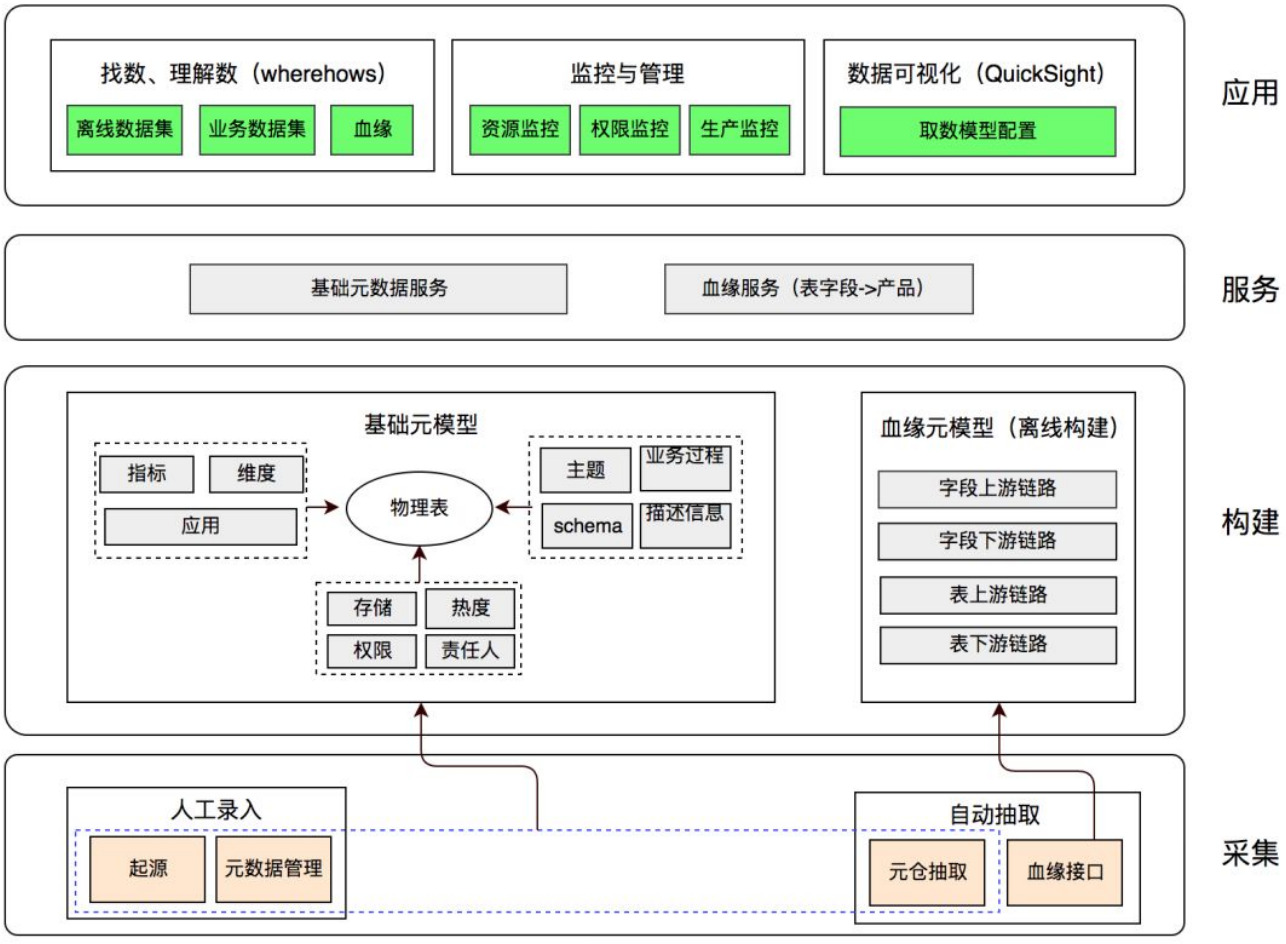
运力进行有效的资源配置，实现高服务体验、低成本的配送服务。对外，我们将配送服务通过平台化的方式，提供给用户、商户和电商平台，以满足不同用户在不同业务场景下的配送需求。对内，我们通过不同的调度模式将运单池中的运单调度给合适的骑手来完成履约，平衡规模、成本和体验之间的关系。如下图所示：



配送业务模式抽象

基于以上的业务模式，我们划分了运单主题（对履约数据域下的数据进行构建，支撑规模和体验的数据分析需求）、调度主题（调度数据域下产生的数据，用于支撑调度策略的分析）、结算、评价、投诉、取消主题（用于支撑体验、成本数据分析需求）和管控主题（用于支撑运力奖惩、违规和招募分析需求）等各种主题，并在每个主题下划分对应的业务过程，在应用层制定分析方向的分析标签，通过对元数据内容的建设完成对业务的抽象，为物理模型的刻画准备了基础数据。

第三，元数据服务建设，我们打通了元数据从采集到构建再到应用的整条链路，为使用数据提效，解决“找数、理解数、评估”难题以及“取数、数据可视化”难题。在整个建设过程中，我们围绕着元数据采集、元模型构建、元数据服务以及最后的产品应用进行展开，整体架构如下图所示：



元数据采集

元数据采集分为人工录入和自动抽取，通过人工录入的方式实现物理表的准确归属（包括该表属于仓库哪一层、对应的主题、业务过程、星型模型关系等）以及指标的采集，从而完成技术元数据和业务元数据的采集，通过自动抽取的方式完成生产元数据的采集和使用元数据的采集，主要包括：物理模型的依赖关系、存储占用、热度、等信息。

元模型构建

分为以物理表为核心的基础元模型构建，以及以血缘为中心的血缘元模型。基础元模型构建以物理表为中心，打通其与技术元数据（主题、业务过程、Schema）的关系，实现了物理表的清晰归属，打通其与生产元数据的关系，为其加上了物理表查询热度、资源消耗、查询密级等生产使用信息，打通其与指标、维度和应用的对应关系，为上层的取数应用建立了完备的元数据。血缘元模型以血缘为中心，不仅构建了从上游业务表到仓库离线表的物理血缘，而且打通了仓库离线表到下游对应报表的血缘，为后续的影响评估构建了完备的元数据基础。

元数据服务

统一元数据服务（OneService），主要提供两类元数据服务，提供查询表、指标、维度基本信息的基础元数据服务以及查询表级血缘、字段级血缘的血缘服务。

元数据应用

主要孵化出了三个产品，以“找数、理解数、影响评估”为应用场景的数据地图（Wherehows），以“取数、数据可视化”为应用场景的数据可视化（QuickSight），以及以管理审计为目的的管理审计报表。

4.2.3 安全治理

安全治理主要加强了敏感数据的安全治理和数据共享环节的安全治理。通过对隐私数据的安全治理，不仅要保证其在存储环节的不可见性，而且还要保证在其使用环节对用户进行双重鉴权，字段的密级鉴权和解密的密钥鉴权；通过对数据共享环节的安全治理，我们在数据分级分类的基础上，使数据的权限控制从表级权限控制扩展到行级权限控制。

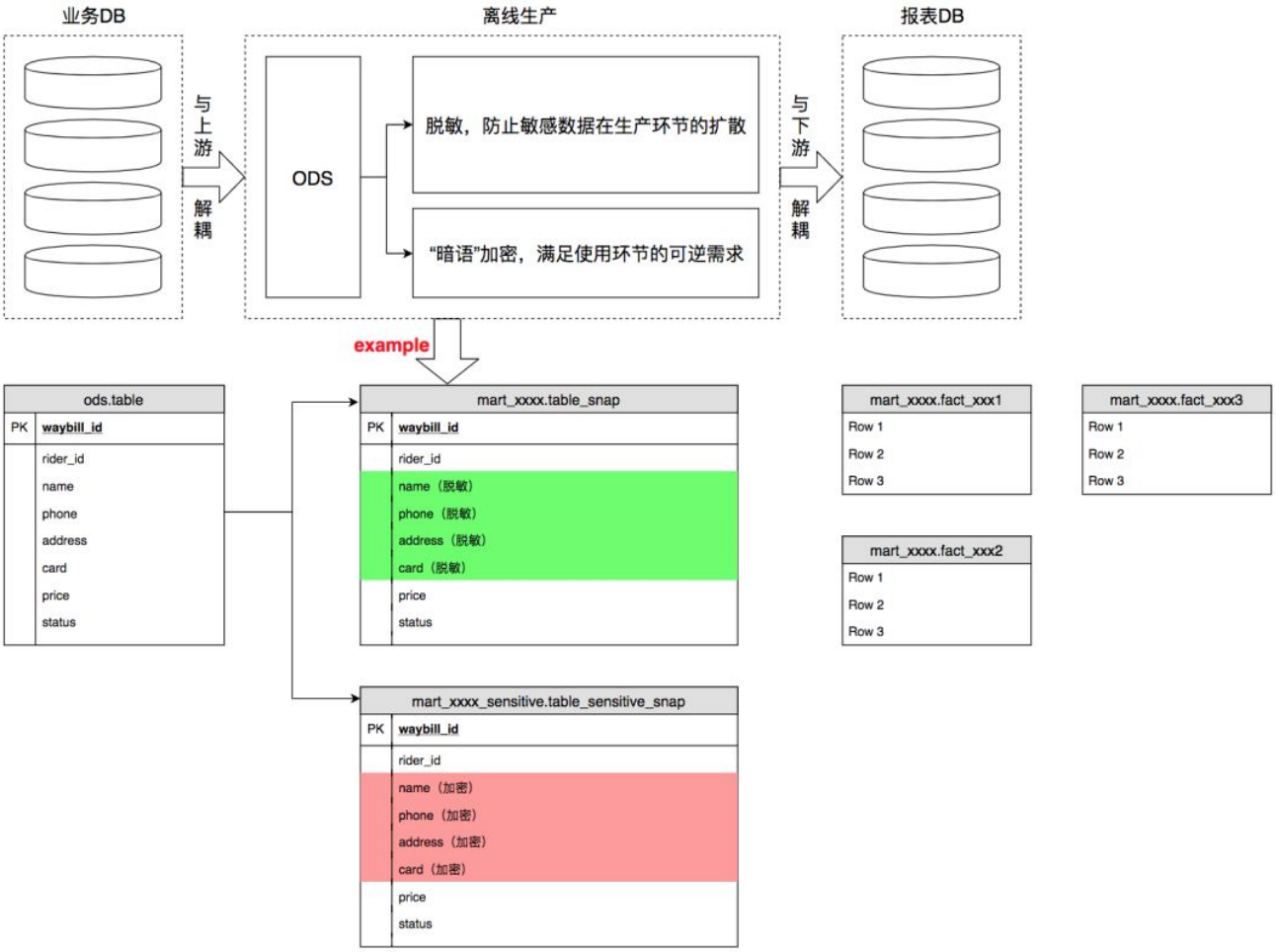
敏感数据安全治理

敏感数据的安全治理，主要是解决敏感数据的存储安全和使用安全。离线场景下，敏感数据存储安全要解决两大挑战：

- 确保仓库侧处理方案既要屏蔽上游业务系统变动带来的影响，又要屏蔽自身策略对下游BI系统的影响。
- 要避免敏感数据在整个加工链路中的扩散。

因此，为解决仓库处理方案与上游业务系统和下游BI系统的解耦问题，我们在上游敏感数据落到ODS环节，确保落到ODS层的敏感数据必须是明文，为保障其安全，对ODS层的所有数据进行文件加密，但是在使用层面，对下游链路透明保障下游链路的正常生产，并限制ODS层数据权限的开放。

ODS层数据只用于安全生产，通过此方案既屏蔽了上游处理方案对仓库的影响，又解决了敏感数据的安全问题。当数据从离开仓库时，在传输环节对敏感数据进行可逆操作，将敏感数据以明文的形式推入BI库，实现与下游BI系统的解耦。为解决敏感数据在整个生产链路的扩散，我们在快照层对敏感数据进行脱敏处理，从快照层开始消除敏感数据，为保障敏感数据的可逆性，将ODS层的敏感数据抽取到安全库中并进行加密存储，实现安全独立管理。具体执行如下图所示：



针对敏感数据的使用安全，我们通过对敏感字段的权限控制和对解密密钥的权限控制，来实现敏感数据使用安全这一目标。针对单独抽取的敏感数据，我们除了针对敏感数据设置其相应的密级确保敏感数据的权限管控外，还基于"暗语"的加密方式为每个项目组分配一个相同的密钥，并且将该密钥存放到与Hadoop集群集成的KMS进行管理（确保支撑离线计算的高并发），确保解密时实现密钥的权限管控。

共享环节安全治理

针对共享环节的安全治理，我们主要是在数据生产环节完成数据的分级分类和数据确权，在数据的使用环节完成数据的表级权限控制和行级权限控制。确保数据在使用环节规范的审批流转，权限开放以后的安全审计，保证数据走不脱。

首先，我们在生产环节B3、B2、B1层数据按照主题或实体C层数据按照应用方向进行逻辑划分，并设定

资源的密级和权限负责人。特别地为实现B3层数据在查询环节可按照业务线进行权限管控这一目标（即行级鉴权），针对B3层数据，我们标记该数据需要在查询环节进行行级权限管控，标记使用行级鉴权所需的字段和该字段对应的枚举值。

其次，在使用环节，我们按照资产密级和使用人角色完成数据的审批流转，实现数据的安全共享。

第三，针对B3层数据，审计是否设置了行级权限管控。在数据开放时是否存在越权使用的情况，以及针对即将离职员工加强数据的使用审计，保证数据走不脱。

在数据“由乱到治”的治理过程中，我们不仅实现了存量数据的“由乱到治”，并且在此过程中沉淀出了一系列的建模方法论、工具，并建立了相应的安全小组和指标运营组织。同时，我们为后续增量数据治理确保数据建设“行不逾矩”，提供了强有力的组织保障、稳定的辅助工具和严格的执行标准。在数据治理的第二阶段实现增量数据的“行不逾矩”的过程中，我们主要围绕大数据架构审计、大数据安全与隐私管理审计、大数据质量管理审计和大数据生命周期管理审计这四方面的工作展开，保障治理工作的持续进行，不断提高了组织的治理水平。

5. 工具简介

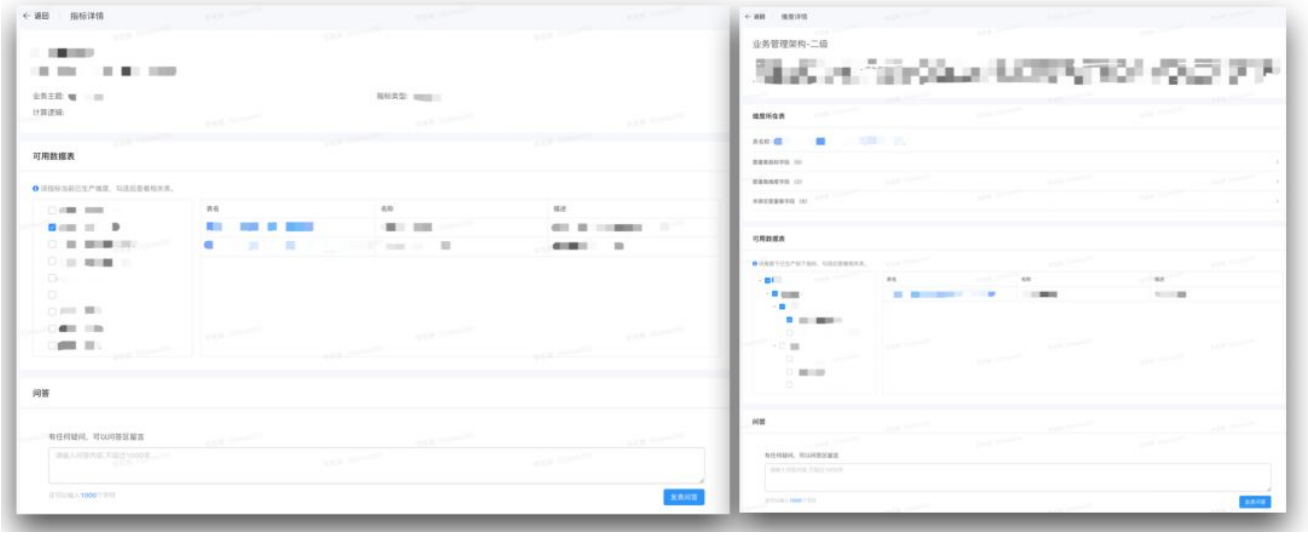
5.1 数据地图（Wherehows）

数据地图作为元数据应用的一个产品，聚焦于数据使用者的“找数”场景，实现检索数据和理解数据的“找数”诉求。我们通过对离线数据集和在线数据集的元数据刻画，满足了用户找数和理解数的诉求，通过血缘图谱，完成物理表到产品的血缘建设，消除用户人肉评估的痛苦。

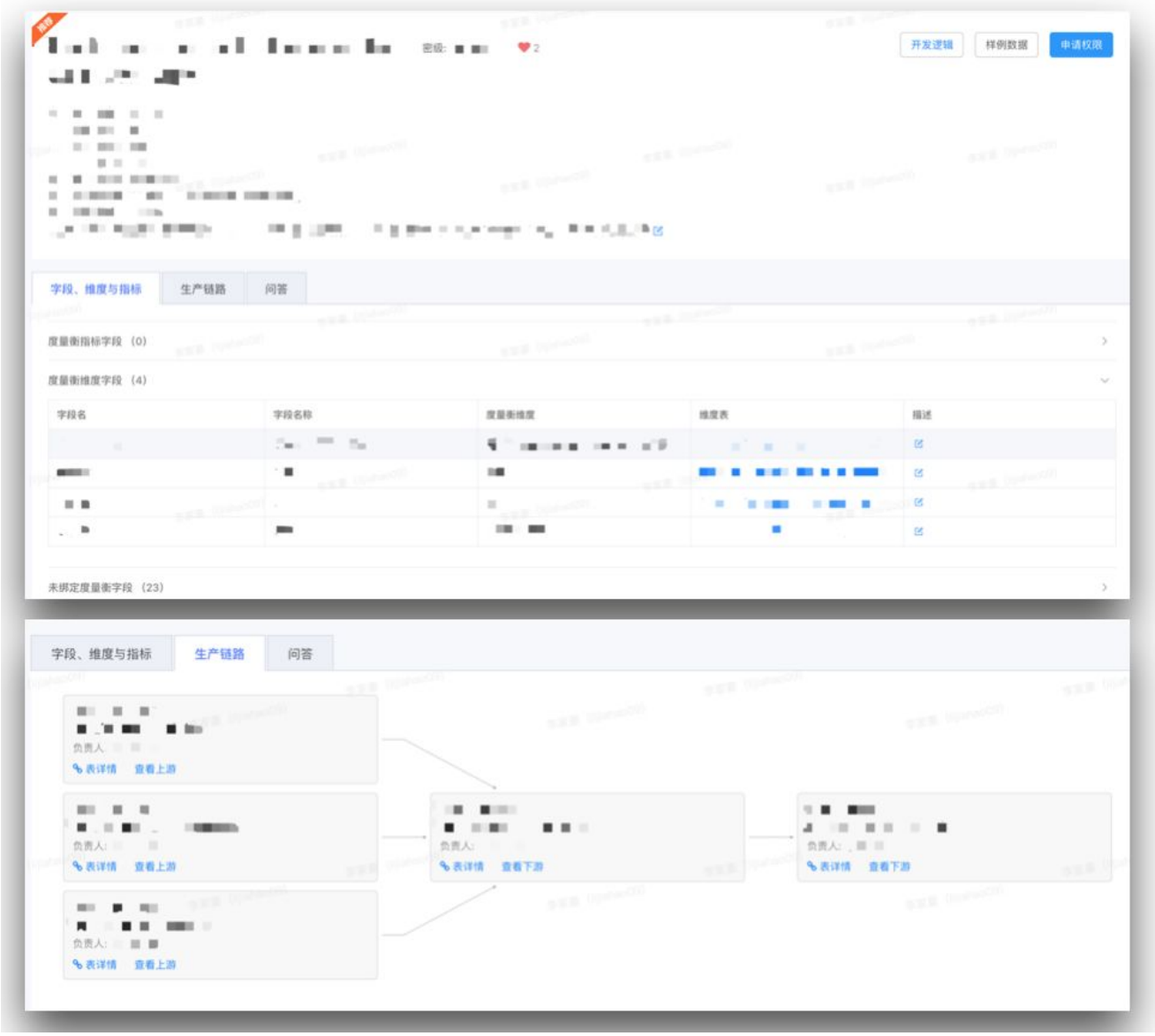
离线数据场景

1. 关键字检索和向导查询共同解决了“找数据”的问题：大部分的检索数据场景下，数据使用者都可以通过关键字检索来得到匹配结果。剩下的一小部分场景，例如，对于新人入职后如何了解整个数仓和指标的体系（数仓分几层，每层解决什么问题，都孵化出什么模型；整个指标、维度体系都是怎么分类，有哪些指标和维度），这部分场景可以使用向导查询功能。向导查询相当于分类查询，将表和指标按照业务过程进行分类，用户可以按照分类逐步找到想要的表或指标。

2. 我们打通了业务元数据和技术元数据之间的关系，提高了“找数据”的能力：通过“Wherehows”查找到指标后，不仅不可查看指标的业务定义，还能查看指标的技术实现逻辑，指标在哪些维度或维度组合中已经实现，并且能够在哪张表里找到这些维度，或维度组合的指标数据。反之，也可以知道在某个维度下已经实现了哪些指标，对应的指标在哪些表里。这些功能能让用户更加方便地找到想要的数据。



3. 我们提供了较为完善的数据信息，帮助用户更好理解数据：对于表的信息，“Wherehows”除了提供表和字段的中英文名称、描述信息等基础信息外，为了帮助用户更好地理解表的建设思路，我们还提供了表的星型模型（可以关联的一致性维度及对应的维度表）、表的血缘关系等信息。



4. 我们通过评论问答功能，帮助用户可以快速得到问题反馈：如果用户看了信息后还是感到有问题，“Wherehows”提供评论问答的功能，用户通过这个功能可以进行提问，会有相应的负责人进行回复。对于重复问反复问的问题，用户通过查看其它人的提问和回复就能找到答案。并且负责人还会定期的将问答信息沉淀到对应的元数据里，不断地对元数据进行补充和完善。

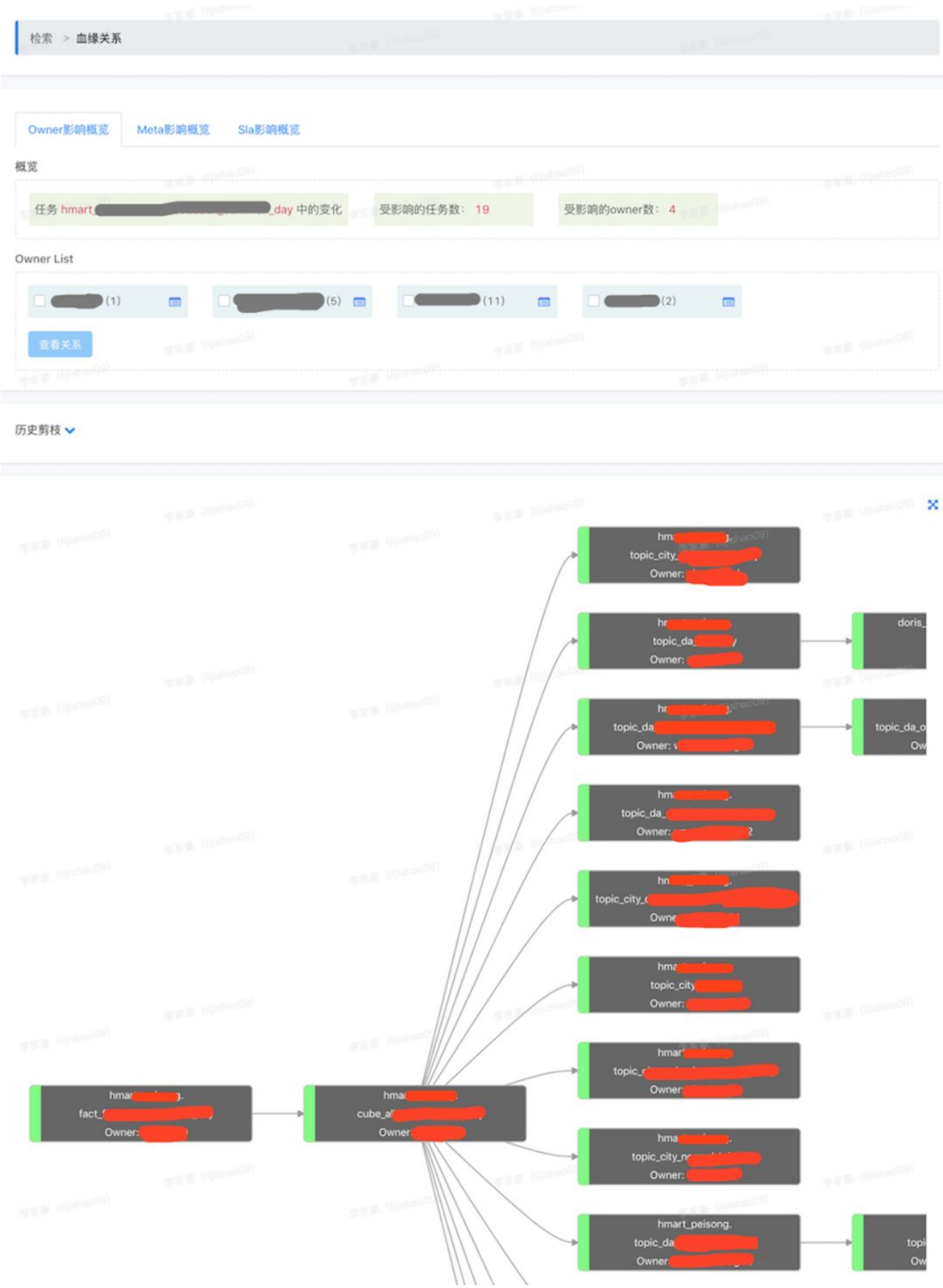
业务数据场景

业务数据场景主要想解决的一个问题是，如何知道一个业务表（MySQL表）有没有同步到数仓。如果没

有同步，能够找谁进行同步。因为已经打通“业务表 -> 数仓表 -> 产品”三者之间的血缘关系，我们能够轻松解决业务数据场景的问题。

生产评估场景

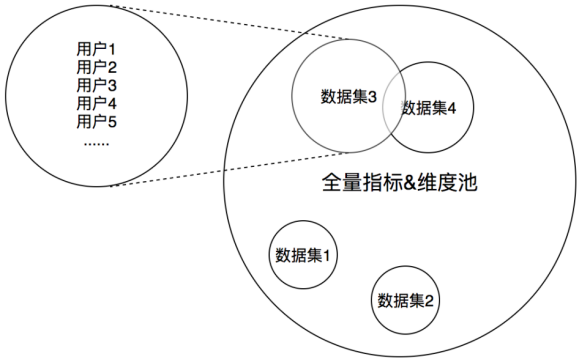
在日常数据生产工作中，我们经常需要对表进行影响评估、故障排查、链路分析等工作，这些工作如果靠纯人工去做，费时费力。但现在我们已经打通了“业务表/字段 -> 数仓表/字段 -> 产品”三者之间的血缘关系，就能够在10分钟内完成评估工作。对于不同的场景，血缘链路提供了两个便捷的功能：过滤和剪枝。例如，某个表逻辑需要修改，需要看影响哪些下游表或产品？应该要通知哪些RD和PM？这种情况下，血缘工具直观地显示影响了哪些负责人和产品，以及这个表的下游链路。



有些表的链路很长，整个血缘关系图很大，这样会导致用户定位信息或问题。所以血缘工具提供了剪枝的功能，对于没用的、不想看到的分支可以剪掉，从而让整个链路变得更加直观。

5.2 数据可视化（QuickSight）

聚焦于数据使用者“取数”场景，使用QuickSight，用户可以不再关心数据的来源，不再担心数据的一致性，不再依赖RD的排期开发。通过所选即所得的方式，满足了用户对业务核心指标的二次加工、报表和取数诉求。首先，我们通过指标池、数据集等概念对离线生产的指标进行逻辑隔离，针对不同用户开发不同的数据集以达到权限控制的目的，如下图所示：



用户、指标池与数据集间的关系

其次，我们为用户提供一系列的组件，帮助用户基于为其开放的数据集实现指标的二次加工和数据可视化功能，满足其在不同业务场景下的取数和可视化应用。如下图所示：

指标加工组件

6. 总结与展望

经过三个阶段的治理工作，我们在各个方面都取得了较好的效果：

- 在数据标准方面，我们制定了业务标准、技术标准、安全标准、资源管理标准，从而保障了数据生产、管理、使用合规。
- 在数据架构方面，我们通过桥接表、时间刻度化、业务口径下沉等手段提升模型灵活性，并保障数据一致性，消除跨层引用和模型冗余等问题。
- 在数据安全方面，我们加强了对敏感数据和数据共享环节的安全治理，保证数据拿不走、走不脱，隐私数据看不懂。
- 在元数据建设方面，我们打通了从采集到构建再到应用的整条链路，并为数据使用人员提供数据地图、数据可视化等元数据应用产品，帮助他们解决了“找数”、“取数”、“影响评估”等难题。

未来，我们还会继续通过组织、规范、流程等手段持续对数据安全、资源利用、数据质量等各方面进行治理，并在数据易用性上下功夫，持续降低用户的数据使用成本。

- 在数据架构方面，随着数据库技术的飞速进步，现在已经有很多数据库能够支持千万级乃至亿级数据的现算先用，我们也在尝试使用这些数据库帮助提升数据开发效率，改善数仓分层管理和应用支撑效率。
- 在数据产品方面，我们将持续完善数据地图、数据可视化等数据应用产品，帮助用户快速探查、高效分析，真正发挥数据的业务价值。

作者简介

王鹏，2016年加入美团点评，目前在配送事业部数据团队负责众包业务数据建设、数据治理和系统化相关工作。

家豪，2018年加入美团点评，目前在配送事业部数据团队负责众包业务数据建设、数据治理和系统化相

关工作。