**云原生时代的运维转型**

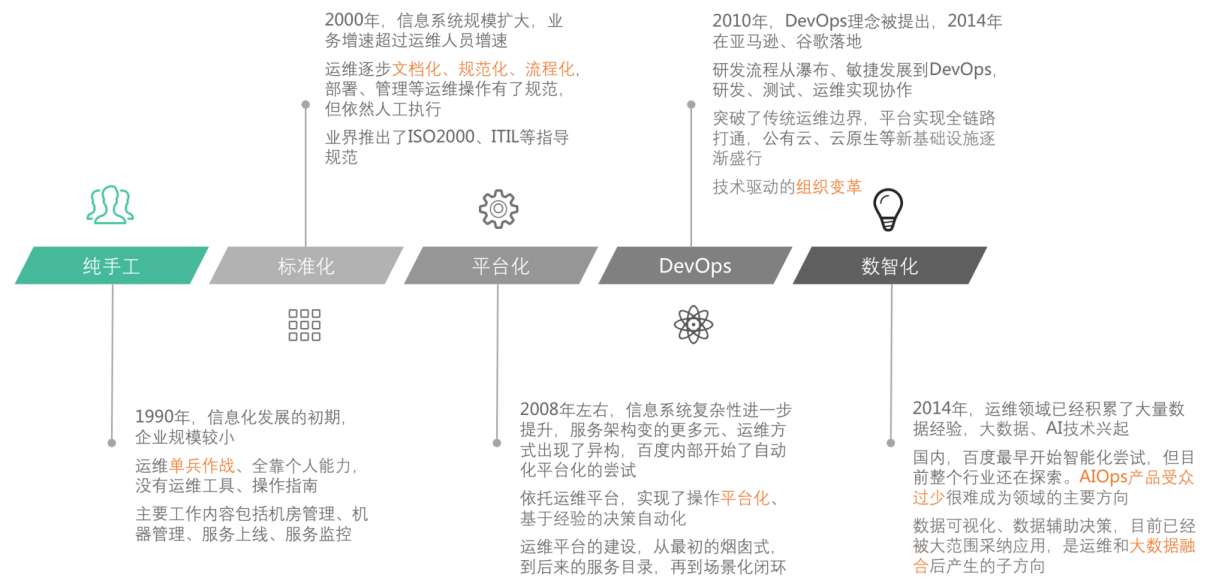
云原生时代来了，运维势必面临着转型的机遇和挑战。以下，是作业帮运维团队的转型探索实践。

**要点简述**

* 传统运维职责：将工业制成品组装成服务、交付给用户，并维持服务运转；特点是强依附于业务。
* 云原生时代，公有云大量使用、DevOps真实达成，传统运维的职责不断被外包、转移、替代，出现了领域危机。
* 运维转型，核心是改变角色认知。运维人，要把自己从依附于业务的运营角色，调整为独立的运维服务提供方。
* 运维转型，目标是运维即服务OPaS。运维转型为服务提供方，专注于打造OPaS能力，持续深化专业性；业务研发借助OPaS，自助完成日常的运营工作。
* 转型实践，作业帮做了一些转型实践，洋葱模型、超服务视角、管理面工单ToC这三条路，已经被我们证明可行。
* 平台是服务能力最有力的承接方式，但不是唯一方式。组织、规范、流程、平台，一样都不能少（除平台外，转移成本可能略高）

**运维发展史**

互联网运维整体上分为5个阶段：纯手工、标准化、平台化、DevOps、数智化。



20世纪90年代是信息化技术发展的初期，企业规模比较小，彼时的运维以单兵模式为主，全靠个人能力，主要工作内容是机房相关以及服务相关，这个阶段称为纯手工。

2000年左右，信息系统进一步扩大，业务的增速远远超过了运维人员的增速，此时不能再按照传统纯手工的模式了，因此逐渐发展出以文档化、规范化、流程化为代表的标准化规范。在这个阶段依然是需要人工执行部署和管理等操作，业界也相应地推出了ISO2000、ITIL等等指导规范。

时间来到2008年左右，这个时候信息系统的复杂度又有了进一步的提升，包括服务架构、语言变得更多元等，运维方式出现了异构（运维成本高）。在国内，百度最早开始平台化和自动化的尝试。这个阶段主要是依托运维平台来实现人工操作的平台化，以及部分基于经验的决策自动化。运营平台的建设也先后经历了早期的脚本化，之后是烟囱式的发展，再到服务目录的集成，最后是基于场景去闭环。

2010年左右，DevOps理念被正式提出来，2014年左右在亚马逊和谷歌完成了落地，研发流程也从原来的瀑布、敏捷开发模式发展到了研发、测试、运维一体化协同。这基本突破了运维的原有边界，工具体系或者平台体系也实现了全面弱化。这一阶段可以认为是技术驱动的组织结构变革。

同样在2014年左右，运维领域已经积累了大量的数据，又恰逢大数据和AI技术兴起，国内也是百度最早开始尝试智能化。但是从目前来看，AIOps还是处在探索阶段，产品的受众比较少，很难成为一个领域的主要方向。而数据化和数据辅助决策目前已经被大范围地采用，发挥了不错的效果。这可以认为是运维和大数据结合之后产生的一个子方向。

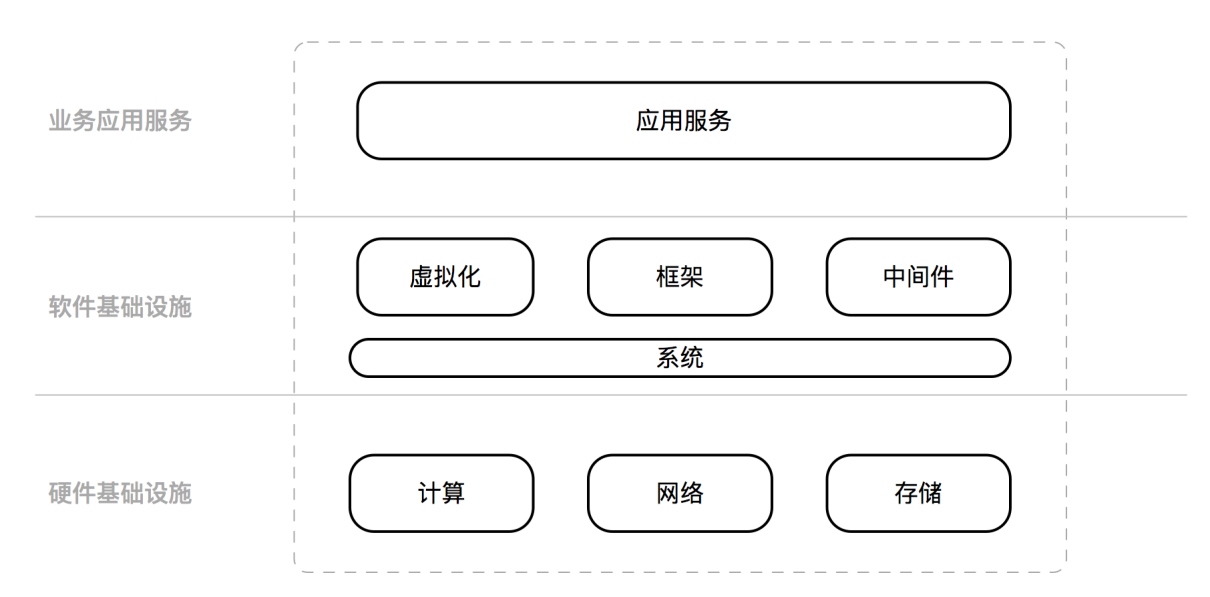
从上述的运维发展过程中，我们可以看到两个特点。

* 第一，继承性。新阶段往往继承、发扬老阶段的优秀经验，又会在理念、技术、组织上有所创新。比如，平台化继承、强化了标准化阶段的成果，数据化继承了平台化的成果，同时引入大数据技术，DevOps也是一个左右能够衔接起来的组织变革。
* 第二，职能的转移，特别是运维职能的转移。DevOps是运维管理模式的一个分水岭。
* 一方面，沿着本领域专业化的方向继续推进，不断强化工具体系建设的重要性，包括不断完善自动化、数据化机制，甚至是现在还不成气候的智能化机制。
* 另一方面，则强调运维研发的模式融合，运维的职责开始逐步转移到业务研发。

随着后来公有云和云原生的大范围使用，运维职责被压缩的趋势更加明显，这也引发了对运维转型的相关思考。

**传统运维**

在传统的运维模式下，服务对象基本上可以划分为三层。最底层的是基础设施层，包括了计算、网络、存储；中间层是软件基础设施层，包括操作系统、虚拟化技术、代码框架、中间件等；最上层对应的主要是业务交付的应用服务。



运维的传统职责其实是将上述工业制成品组装成服务，交付给用户，并维持服务的正常运转。通常会要求在稳定、成本、安全、效率之间达到多快好省的平衡。在这个职责过程中，运维必须依赖于业务，才能体现出个人的价值。

云计算和云原生兴起之后，传统的运营模式遇到了很多挑战，比如：

* 随着企业大量使用公有云，IaaS/PaaS甚至SaaS都被服务化了，通过API就可以简单的获取；大量的运维建设工作由云厂商帮忙完成了，比如硬件、系统、网络、数据库、大数据等，原厂只需要保留少量的专业选型和集成能力（外包）。
* 随着云原生技术的普及，DevOps大范围达成，之前只能由专业的运维人员来做的管理操作，可以交给业务研发自助完成，比如服务的交付、变更、观察、容量等，运维职责被大量转移到业务研发团队。
* 公有云的专业聚集效应，以及云原生的开源体系，提供了持续向好的工具前景。工具化提高效率，就意味着做同一件事需要的人变少了。同时工具化沉淀了很多专业的经验，意味着对运维从业者的技术门槛要求降低了。同时工具如果进化到了自动化、智能化以后，机器就可以完成对人工的替代。

从上面讲到的这几个方向，可以看到基础设施外包给公有云、云原生之后，运维职责从运维团队转移到业务研发团队，云平台对人工本身有替代性。面对这样的事实和趋势，运维从业者需要考虑做出一些转型。

**运维转型思路**

云原生时代组织结构是一个什么样子？从长期来看，云原生时代的组织形态大概由几部分构成。



最上面的终端用户，是企业的甲方用户或者最终用户，通常提供了潜在的盈利目标人群。第二个是业务团队，为终端用户服务，角色包括了产品、商务、营销、市场等等。业务研发，直接为业务团队服务，主要是提供SaaS化的应用或服务。橙色底色的平台研发，则是为了业务研发去服务的，它提供各式各样内部的PaaS化的平台，方便业务研发更高效地完成自己的工作。对下封装了云厂商的各种能力。同时我们还有一些横向的跨功能组织，如负责成本运营的FinOps团队，负责稳定性运营的SRE团队，负责工程效率的EP团队还有行政团队等。

在组织结构中，大家的目标其实都是为了集中力量把终端用户服务好。整体来说就是，越向上的团队，越关注业务的价值；而越往底层的团队，越关注对上层团队的服务质量。随着技术的进步，跨功能的组织也会慢慢地细化为平台研发团队，最终还是以服务化的方式来呈现价值。

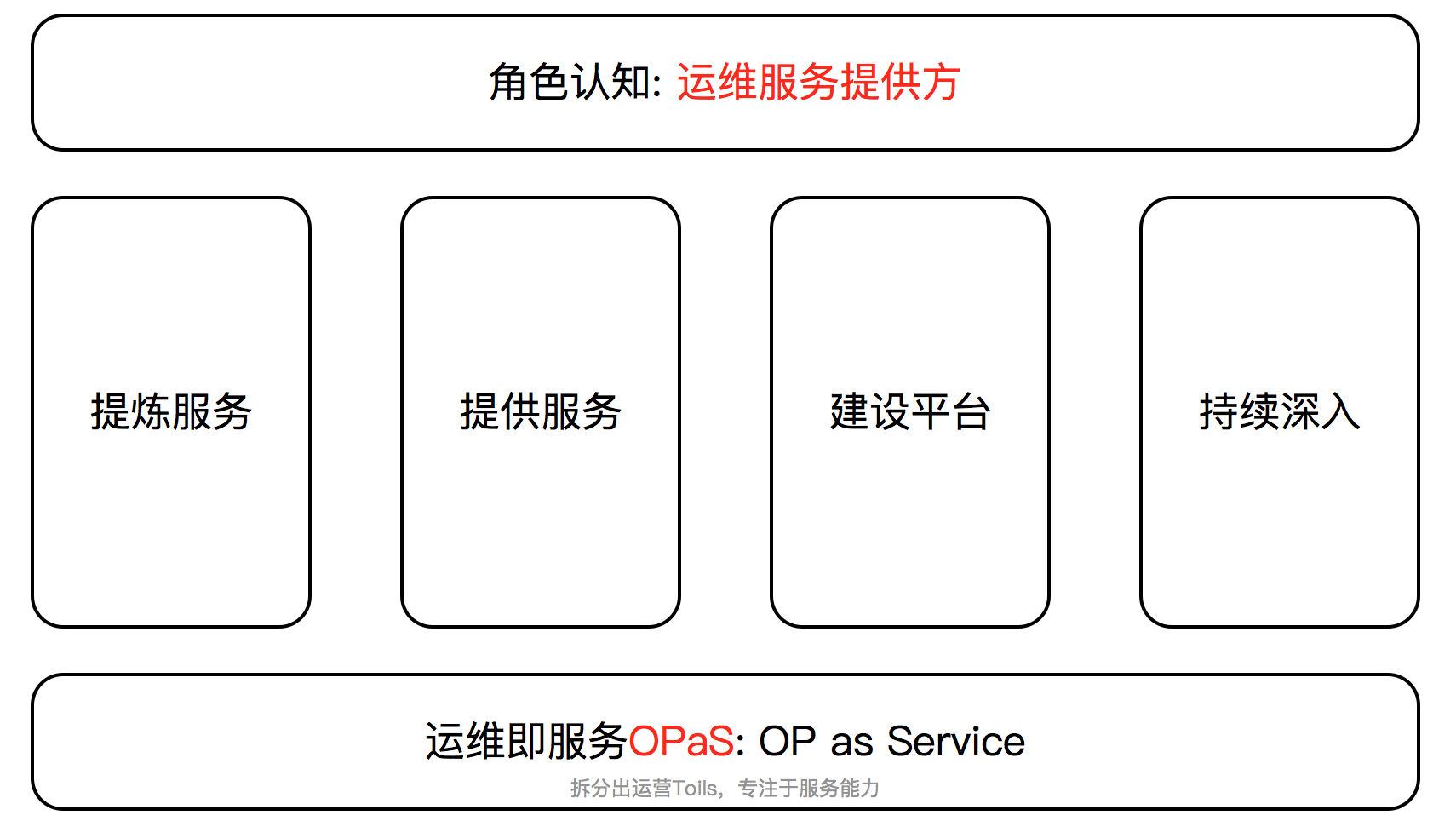
这个组织结构其实就是对于传统运维转型思考的组织背景。

对于运维团队来说，除了全局类的规范制定、标准实施、全局管理等跨团队的职能外，还需要向着用平台来提供服务的方向去转型。

首先要改变角色的认知。运维人要把自己从原来依附于业务才能产生价值的这种运营工作，调整为具备独立价值的运维服务提供方。这是整个转型过程中最关键的部分。有了这个认知之后，接下来就要实施转型，基本上可以按照四步来做：

* 提炼出自己的服务是什么。
* 提供服务的能力，先把服务提供出去。
* 有了提供服务能力之后，要通过建设平台的方式把这个能力固化，让服务提供的更加标准、持久。
* 围绕着提供服务这个场景，把它变成一个专业的领域，持续深入下去，形成一个纵向深入的、比较完整的闭环。

这四步运维转型，最终的目标都是要达成运维即服务OPaS。运维转型为服务提供方，专注于打造运维即服务OPaS的能力、持续深化专业性；业务研发则借助OPaS，自助完成日常的运营工作。

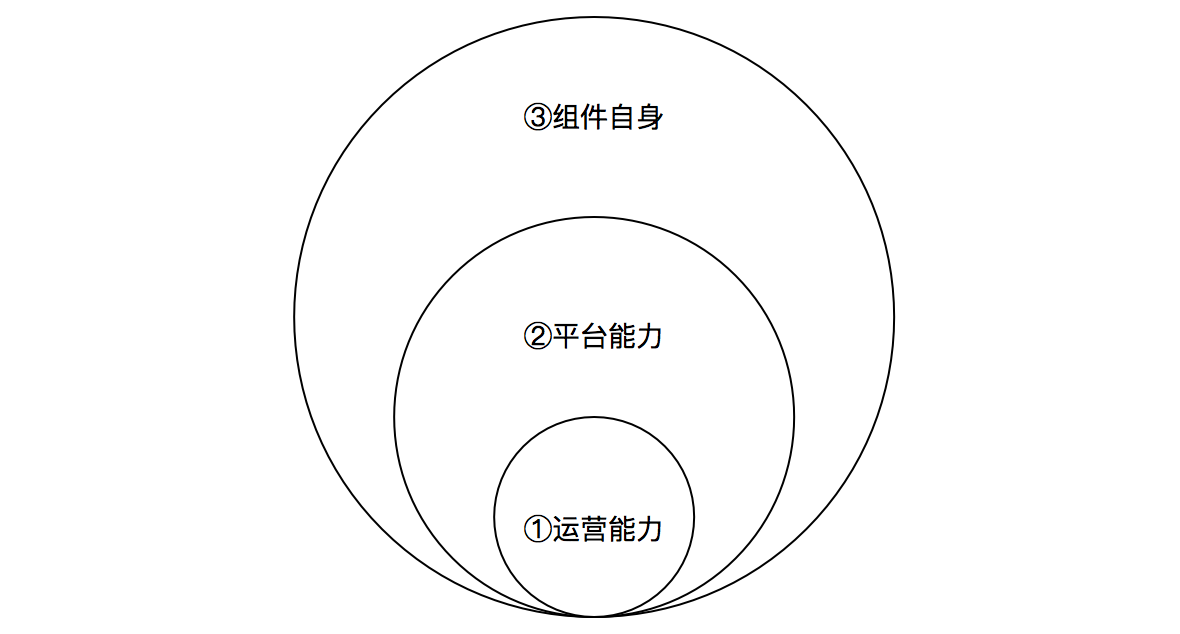


如上图所示，运维转型的三明治模型，符合我们对云原生组织形态的判断。这里也有一个额外的好处，运营Toils交给业务研发自助，解放了运维、提效了研发，DevOps!

**转型实践**

洋葱模型

对于云服务、中间件、数据库、大数据这样的运维角色来说，提炼运维服务是一个相对简单的过程。在转型过程中，他们基本上可以按照下面洋葱模型这三步来走。



* 第一步，立足于现在已有的运营优势，把原来的运维对象直接变成服务的实体，然后对外提供出去，初步建立一个价值底线。
* 第二步，抽出很大一部分精力来建设服务实体相关的管理平台。把服务实体的生命周期管理给完整地做下来，平台最终的目标其实是要借鉴DevOps的思想，最终能ToC化，方便降低团队的人力负担。
* 第三步，深入到组件自身的专业领域中，从代码、架构、运维、性能等多个方面来提升专业性，建立专业的壁垒。做到第三步的时候，运维其实已经转换成了平台研发的角色。

整个模型称为洋葱模型，从下层最熟悉的运营领域逐步过渡到要求相对较高的平台领域，再到一个要求非常高的专业领域，难度是从低到高的渐变过程。

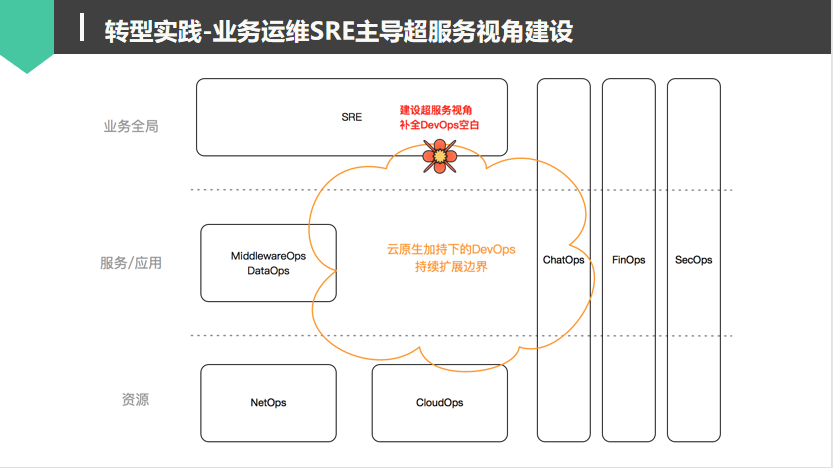
这个模型在之前是数据库、大数据、中间件等运维角色通用的成长模型，现在可以直接拿过来，应用到运营转型中。比如作业帮的云服务运维CloudOps团队，就是借鉴这个模型来实施转型的，具体如下：



* 这个团队的服务实体是各种云服务。在资源层可以看到腾讯云、阿里云以及K8s等各式各样的公有云、混合云的资产，是一个多云的架构。
* 两年前，团队通过各种手工的方式，对外提供了机器、存储、公网带宽等资源，来支撑当时业务的快速发展。
* 之后我们开始建设多云管理平台。多云管理平台主要结构就是上图里的中间部分，分为资源层、引进层和业务层，管理各种云服务的生命周期，多云管理平台对内提供了机器、带宽、对象存储、CDN等多种能力，CloudOps也成功转型为公司内部的二级云服务提供商。这处在洋葱模型的第二阶段。
* 接下来我们还会不断强化对公有云的深入学习、选型以及认知的提升，并且会发挥甲方的优势，推动它去做一些有利于我们的演化。争取在公共有或者云服务这个领域建立更多的专业性，达成洋葱模型的第三个阶段。

超服务视角

话题回到业务运维领域，当前在云原生的加持下，DevOps持续扩展边界。主要是围绕着应用和服务视角，将底层的资源交付细节屏蔽；在中间这层又让资源调度、应用变更基本实现了自动化或者自助化；再往上向左和向上的服务治理环节，包括服务注册发现、服务通信、服务观察、流量控制等服务治理的关键领域治理手段也得到了非常大的提升。



原来属于运维团队的职责，都可以通过成熟的工具体系，比较安全地转移给业务研发同学，那么这个时候危机就来了，大部分的职责都转移之后，我们的业务运维同学到底做什么？这个过程中不可避免地会发生一些团队规模缩小问题，这是不可避免的事情。

但是我们也能看到云原生下的DevOps领域拼图并不完整，在应用或者服务视角之外，还有大量的能力空白。上图中橙色的圈以外，其实都属于DevOps当前没有覆盖的领域，特别是向上的业务视角、部门视角、公司视角等，姑且称为超服务视角。

对于超服务视角，传统的业务研发同学通常是没有能力，也没有动力去抓这个事情，而部门的主管或者部门的架构师，通常可以照顾到本部门的需求，但一旦涉及到跨多个部门，也会受限于岗位职责，很难去做到相对全局的视野。反观，超服务视角是传统业务运维的老战场，他们具备无与伦比的体验、理解和认知优势。

这样，业务运维主导超服务视角建设，提供运维及服务OPaS的能力。这既能发挥我们传统业务运维的专业优势，同时又能填补云原生工具领域空白，运维还能借势转型，这是一个双赢的选择。

基于这一思考，作业帮的SRE团队，瞄准了超服务视角这个方向。按照洋葱模型来看，首先SRE团队提炼出了全局视角的各个服务方向，包括感知方向、预案方向、多云度量方向等。在经过了大半年的运营实践之后，我们将上述能力逐渐平台化。如下图所示，是全局感知下面的报警大盘上的信息，这个大盘是用来做全局的辅助故障定位的，再往下走还会有在预案场景的多云切流以及各个业务的入口级的多云切流，甚至是跨业务的入口切流，这是我们全局的预案场景。



上述提到的场景，本身对于传统运维或传统业务运维来说，并没有太多的难度，但如果把这个方向交给其他人来做，那么他们对产品设计场景理解是有非常大的困难的，也会有一些领域的准入瓶颈。

管理面通过工单ToC

无论是公有云，还是内部的K8s管理平台，都存在着大量的运维管理操作。这些To Manager的操作往往缺少一些必要约束，只能资深人士开放。为了优化分工、提升组织效率，可以通过工单的方式，将运维管理面安全ToC。工单本身的设计，大量融入了运维变更管理的最佳实践，目前也是运维能力服务化的一个子方向。

**教训**

在转型过程中，作业帮也遇到了很多问题：

* 转型和保守的折中。传统运维转型到服务提供方，既不能一蹴而就，也不能全员迁徙，总要有人留下来殿后。因为资源更集中，殿后人员会获得更多的价值回报。
* 研发能力要区分梯度。从运维转型到开发的同学，开发能力参差不齐，要从简单的业务需求做起；团队要配备足够精致的运维中台开发能力，保障底层架构的干净。在中台这一层把所有研发能力参差不齐的劣势隔离在中台之上，提供一个比较干净的基础底层架构。
* 平台不是唯一选择。平台是服务能力最有力的承接方式，但绝对不是唯一方式。组织、规范、流程、平台，一样都不能少。
* 警惕纯粹项目思维。运维还是要参与一些项目，短期内能爆发价值、揽获成就感，但也很容易人走茶凉、价值归零；需要有意识的设计目标，在项目过程中沉淀服务能力。