

架构设计实践思路：什么是架构，怎么画架构图？

原创 胡斌 InfoQ 2019-09-20

作者 | 胡斌

策划 | 小智

本文是架构设计实践五部曲系列文章的第一篇，架构与架构图。本文将对架构作深入的阐释，并教你什么时候画架构图、怎么画架构图。

在日常系统开发过程中，作为技术人员想必大家都参与过架构设计的工作。做过一段系统架构工作之后，心里对于架构产生了越来越多的问题。

- 对于系统的架构，它的本质是什么，它对产品有何影响？
- 架构分为哪几类？
- 为什么要画架构图，可以不画架构图吗？
- 架构图该怎么画，怎么让画架构图不那么痛苦？

为了回答这些问题，我总结了这一系列的文章，沉淀自己对于架构的理解，总结架构设计的实践和思路。希望能帮助到在做架构设计过程中，同样有这些困惑的你。

什么是架构

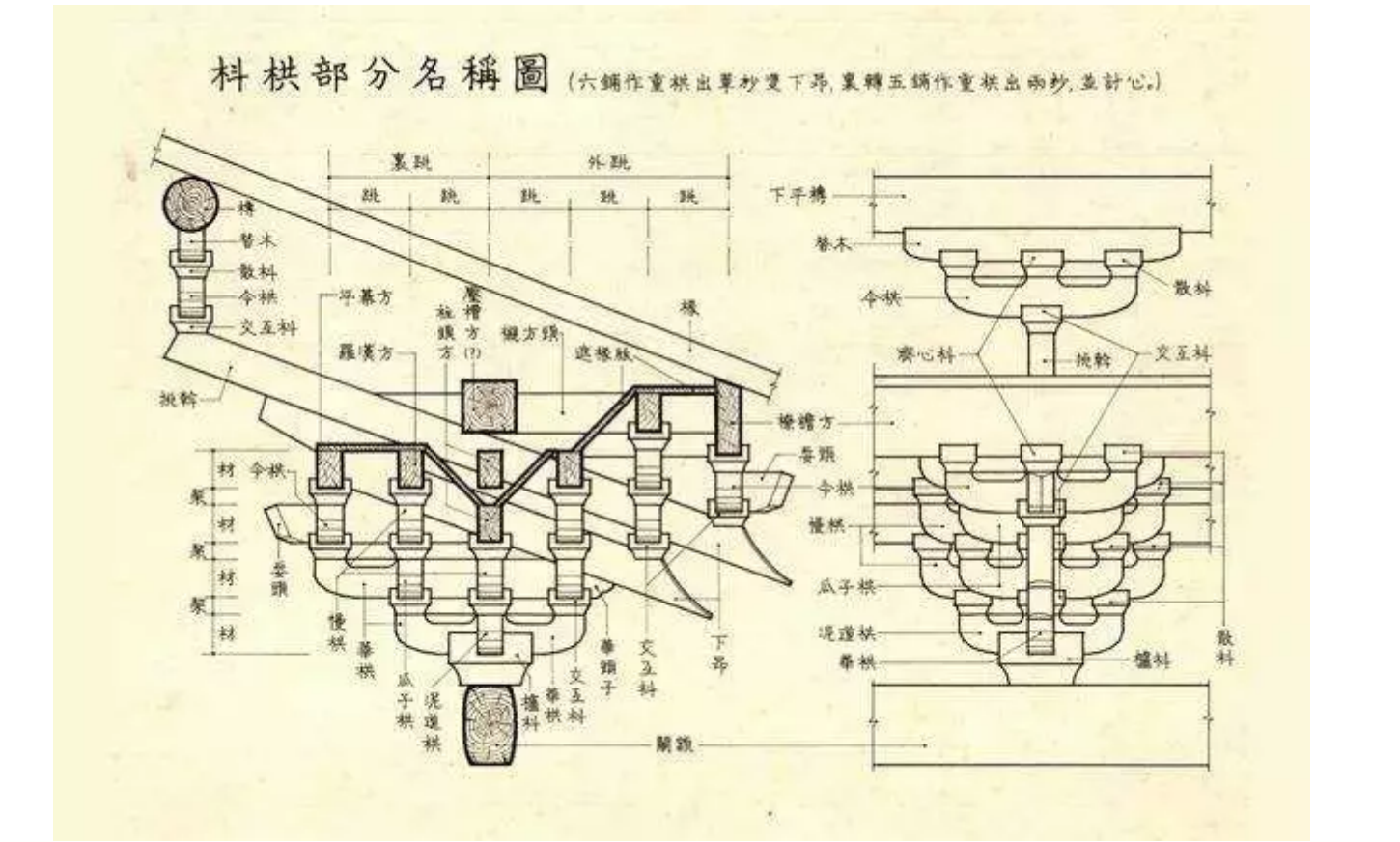
什么是架构？我们先看一下百科中是如何定义架构的。

在百度百科中的定义是：架构，又名软件架构，是有关软件整体结构与组件的抽象描述，用于指导大型软件系统各个方面的设计。

在维基百科中的定义是：软件体系结构是指软件系统的基本结构，创建此类结构的规则以及这些结构的文档。需要这些结构来推断软件系统。每个结构包括软件元素，它们之间的关系，元素和关系的属性，以及每个元素的引入和配置的基本原理。软件系统的体系结构是一种隐喻，类似于建筑物的体系结构。

从百科中我们提炼出一句话，“**架构是一种整体与局部关系的抽象描述**”。这句话还是稍显抽象，不太容易理解。换个角度，按照中文的字面理解，对“架构”两个字进行拆解，就会发现很有意思含义。架构从字面意思上，是源于古代的建筑术语。把架构拆分成两个字“架”和“构”。“架”就是“加”和“木”的结合，把木头加起来、连接起来就是架。“构”就是结构的意思。所以，“架构”就是把“木”按照一定的结构连接起来。

下图为古代的木质建筑的结构图：



对应到软件架构，这里面的“木”代表什么，软件架构中的“结构”是什么，这些软件系统的“木”又是如何连接的？

- 关联到软件领域，木就是系统中的要素，我们将他们称之为架构要素。架构要素可以是子系统、模块、应用服务。
- 结构，是架构的产物。不同的软件系统会有不同的结构，这些结构是为解决不同场景而设计的。

- 连接，通过定义架构元素之间的接口和交互关系、集成机制，实现架构元素之间的连接。连接可以是分布式调用、进程间调用、组件之间的交互关系等。

总结一下架构的本质，即架构 = 要素 + 结构 + 连接，将系统要素按照特定结构进行连接交互。

架构域的分类

在软件设计中架构域是如何划分的，架构域包括：业务架构、数据架构、产品架构、应用架构、技术架构。首先需要熟悉业务，形成业务架构，根据业务架构，做出相应的数据架构和应用架构，最后通过技术架构落地实施。业务架构是战略，应用架构是承上启下，一方面承接业务架构的落地，另一方面影响技术架构的选型。如何针对当前需求，选择合适的架构，如何面向未来，保证架构平滑过渡，这个是软件开发人员，特别是架构师，都需要深入思考的问题。

业务架构

在需求初期，业务的需求描述往往比较模糊，可能只是一句话。他们可能来自老板、运营或者用户。直接把这句话作为核心产品功能是不恰当的，合理的做法是先把这个产品所有的问题域列清楚。

问题域，是指自己的产品能够解决的所有问题的空间集合。从核心需求出发，将所有当前需要解决、未来可能要解决的问题放入产品框架的范围。能够帮助我们的产品拥有更高的可拓展性，在后续具备迭代和优化的空间。

在经过问题域的罗列后，我们应该能够得到一个模糊的产品方向和功能范围。把这些问题域的答案抽象总结成一个确定的产品需求。根据核心需求和问题域的答案，梳理出业务流程。

业务架构就是在业务需求初期，将模糊的需求描述转变成清晰的问题域，梳理出清晰的业务流程。为产品架构提供输入。

业务架构包括业务规划、业务模块、业务流程，对整个系统的业务进行拆分，对领域模型进行设计，把现实的业务转化成抽象对象。没有最优的架构，只有最合适的架构，一切系统设计原则都要以解决业务问题为最终目标，脱离实际业务的技术情怀架构往往是空中楼阁。

业务架构必须与其面向的实际应用场景相匹配，由于每个产品或项目的业务场景均有所不同，所以每次做新的软件开发前，必须设计软件架构。试图不经分析直接套用先前的架构方案，十有八九会让当前的系统在某个点上报出大问题导致推翻重建，更不要说直接拿别人的现成架构方案了。例如，A 业务中有套 A 系统，恰巧 B 业务需要解决类似 A 业务的场景。此时很多情况，B 业务的人员会考虑把 A 系统直接拿过来，以为做一些简单的修改，就能在 B 业务中落地。结果在系统落地的过程中，很多功能模块不能直接使用，都要重新按照业务进行修改。最终的结果是，A 系统经过不断的重写修改变成了 B 系统。

上述的案例正是由于业务架构没有做到位，没有做好软件架构的分析和设计，所以我们很难看出两个系统有多少差别，也无法确定用一个业务系统去覆盖另一个业务系统的可行性有多大。相反，对 A 和 B 业务领域进行业务架构梳理，我们就能清楚发现两者的一致与区别，就能有效的评估系统覆盖的可行性和合理性。

经过业务架构阶段之后，需要输出的产物包括：企业战略方向图、问题域列表、业务流程图。

数据架构

企业架构由业务架构驱动，从业务架构分析业务流程、定义数据架构，流程和数据结合定义产品架构。这中间，数据架构起着至关重要的作用。企业 IT 系统的价值并不在于选取的技术有多先进，使用的硬件有多强大。而是企业业务数据的处理和存储。一家公司最宝贵的资产无疑就是—数据。毫无疑问，在当今大数据的时代背景下，缺少数据资产的建设和使用，就失去与同行业争夺竞争的机会。

因此，数据架构主要解决三个问题：第一，系统需要什么样的数据；第二，如何存储这些数据；第三，如何进行数据架构设计。

系统需要什么样的数据

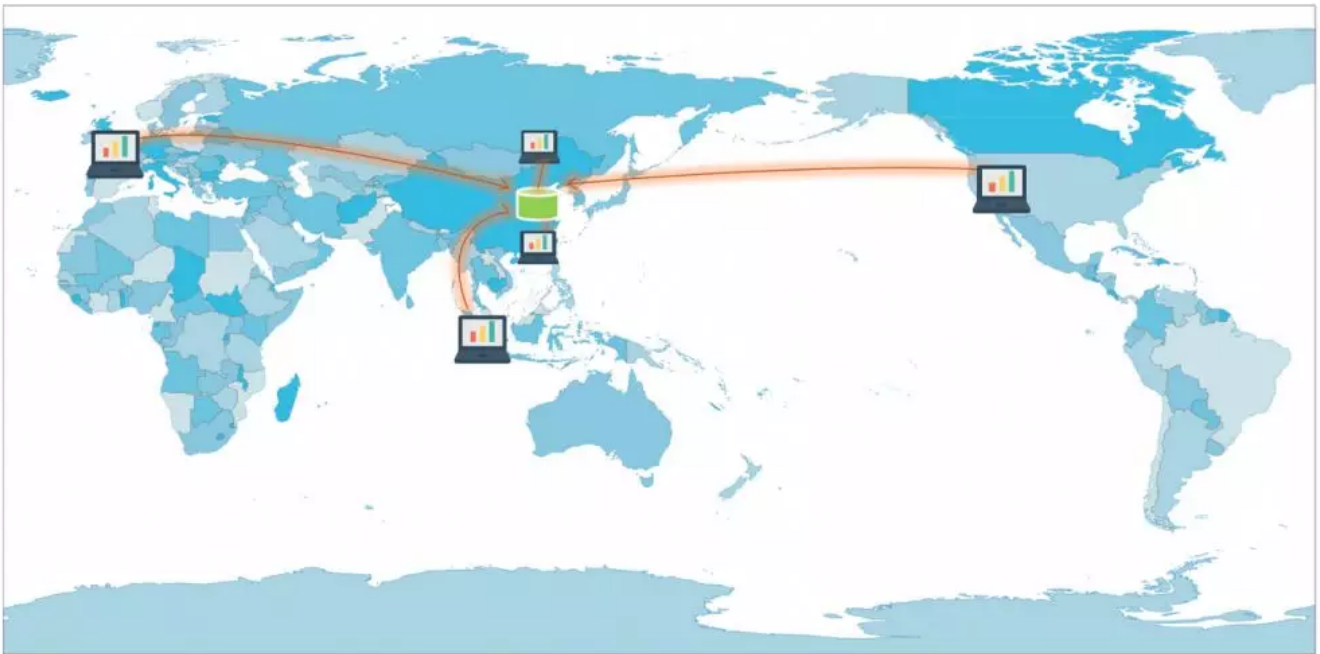
数据是对客观事物的真实表现，企业业务过程中的所有对象的状况都可以用数据记录下来。业务运营过程中有两条重要的线索：流程和数据。业务流程离不开数据流转，业务运营状况通过数据反映。基于业务架构的端到端的流程建模过程中，会衍生出对应的业务数据对象，需要与数据架构模型对接。流程模型和数据模型对接后落实到应用层面，就形成了产品架构。

数据架构中的数据包含静态数据和动态数据。相对静态部分如元数据、业务对象数据模型、主数据、共享数据。相对动态部分如数据流转、ETL、数据全生命周期管控治理。

如何存储这些数据

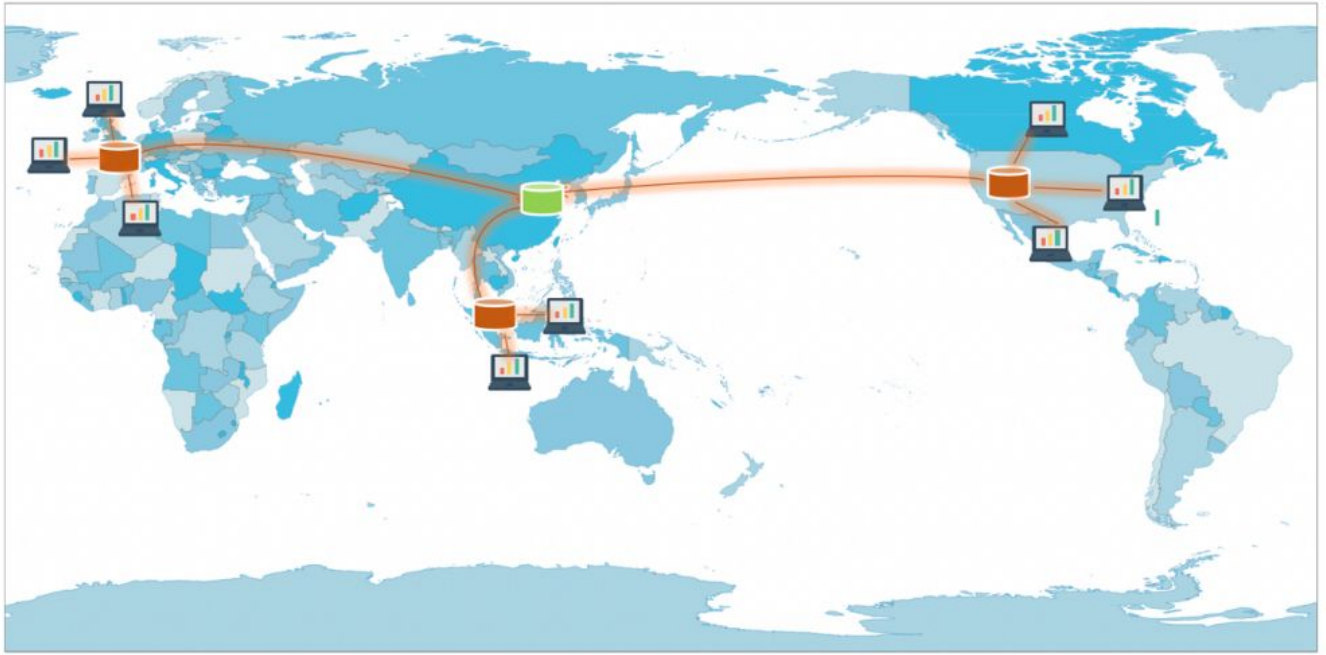
数据架构是为了建立一个共享、通用、一致的数据基础平台，解决企业信息孤岛。如何存储业务数据，需要结合自身需求，采取合适的数据分布策略。通常，数据存储的分布策略有两种：一种是集中式存储，一种是分布式存储。

集中式存储就是讲数据集中存放于总部数据中心，所有的下属机构或子公司不放置和维护数据，都想总部数据中心进行访问。



集中式存储

分布式存储就是数据分布存放于总部、分支机构或者子公司，每个分布节点需要维护和管理自己的数据。分布式的数据存储架构中，还需要考虑每个分布式节点的数据与总部节点数据进行同步、备份，做到数据资产的安全、可靠。



分布式存储

如何进行数据架构设计

数据源自于企业的业务流程，从业务流程中我们可以找出领域对象，基于领域对象进行分析，就能得到对象的属性。根据业务关系进而抽取领域对象之间的关系。因此，领域建模是一种对数据架构很有帮助的建模思想。通过领域建模，我们不仅能清晰的反映企业的业务域，还能清晰的描绘出一幅企业的数据模型。

数据模型最常用的视图就是 ER 图，它主要描述企业数据实体、属性和关系。

- 实体 (Entity): 企业领域对象
- 属性 (Attribute): 领域对象的属性
- 联系 (Relationship): 两个领域对象之间的关系 (1:1, 1:n 或者 m:n)

产品架构

基础的产品框架脱胎于业务流程，但相比业务流程，更加注重产品功能的枚举、功能模块之间的分界。

当我们打开一个系统，我们会看到一个精美的页面，一些丰富的信息、导航。这些东西会引导我们去使用这个系统。这些东西就是这个系统的组成部分，就是这个系统的功能模块。产

品架构，就是将这些不同用途的功能模块围绕特定的业务目标进行分类整合。

功能模块是用户能够完成一个操作的最小粒度的完整功能。比如一个展示可购买商品的列表页、一个修改用户密码的功能。在功能模块设计过程中，需要确保用户能通过一个功能模块完整的完成一项工作，而不是半个工作。

产品架构中，功能模块是根据其相互之间的关系来组织的。一个产品中不同的功能模块之间的关系分直接关系和间接关系。只有直接关系的功能模块才会被组织到一起，形成一个子系统。那些存在间接关系的模块，会在不同的层级通过直接关系的模块产生联系。

当具有直接关系的功能模块组合成一个子系统后，解决相同问题域的子系统就形成一个功能层级。功能层级按照接近用户实操的距离程度进行从上到下，或者从左至右的划分，这就形成了产品架构的分层。

应用架构

应用架构是要说明产品架构分哪些应用系统，应用系统间是如何集成的。这就是应用架构和应用集成架构。产品架构在业务架构的基础上，按照解决的业务问题域，划分出不同的功能模块，再根据功能模块间的关系，组合成子系统。应用架构在产品架构的基础上考虑两件事情：第一、考虑的是子系统间的关系。第二、考虑将可复用的组件或模块进行下沉，沉淀到平台层，为业务组件提供统一的支撑。

应用架构在规划时，需要遵循以下几个原则：

简单性

体现在应用架构是否有清晰、明确的层次划分，各应用系统之间的连接关系是否简单明确，系统之间的耦合程度低。

灵活性

体现在应用架构适应业务的快速变化，不仅要求在快速增加新应用时保持现有应用架构的稳定性，还要在适应业务变化的同时主动促进业务变革。

整合性

通过应用系统之间的解耦和组合，以统一的方式对外提供一致的服务接口，从而实现应用系统之间的共享和协作。

技术架构

应用架构本身只关心需要哪些应用系统，哪些平台来满足业务目标的需求，而不会关心在整个构建过程中你需要使用哪些技术。技术架构是承接应用架构的技术需求，并根据识别的技术需求，进行技术选型，把各个关键技术和技术之间的关系描述清楚。

技术架构解决的问题包括：如何进行纯技术层面的分层、开发框架的选择、开发语言的选择、涉及非功能性需求的技术选择。由于应用架构体系是分层的，那么对于的技术架构体系自然也是分层的。大的分层有微服务架构分层模型，小的分层则是单个应用的技术分层框架。大的技术体系考虑清楚后，剩下的问题就是根据实际业务场景来选择具体的技术点。各个技术点的分析、方案选择，最终形成关键技术清单，关键技术清单考虑应用架构本身的分层逻辑，最终形成一个完成的技术架构图。

总之，技术架构是将产品需求转变为技术实现的过程。

为什么要画架构图

梳理自己对产品和技术方向的判断

我们是不是都会遇到这样的情况，每次画图前，脑海中有一张看似清晰的图。但是一到画图那一刻，这张图确又是那么无比的模糊。思考这张图如何设计的过程，也是帮助你梳理“未来产品该朝什么方向发展、功能迭代如何分期和落地、和其他产品的依赖关系是什么、技术方案该如何演进”等问题的过程。

提供产品 & 技术 & 运营的输出

当产品架构图被设计出来后，按照产品架构图的结构和路径，项目的里程碑（RoadMap）就可以被清晰的拆解出来，同时项目成员也可以根据这张架构图产出运营计划、技术架构等强依赖产品方向的方案。

当技术架构图被设计出来后，技术方案、技术开发进度就可以被清晰的制定，技术选型就会明确的选定。

让他人可视化的理解你的架构

能清晰简单的呈现自己的思路、明确自己产品的边界和技术边界、指明发展的方向，帮助不了解你产品的人快速的建立对你产品结构、功能、技术的认知。

提供面向不同人员的视图语言

不同角色对于架构视图的需求有所不同，一张图走天下是不太现实。往往需要针对不同的用户，提供不同视角的架构视图，便于不同角色用户对于产品的理解。

来看一个生活中视图的例子。就拿中国地图为例，不同的人员会关心不同的问题，例如图 1（中国气候类型）是气象学家关心的，而图 2（商品粮基地分布）是粮食局人员关心的。如果气象学家拿到一张商品粮基地分布地图，对他们来说根本就没有什么研究价值。看地图的人希望一幅地图能针对他的需求来展现。

同理，软件架构视图也需要按照不同的视角提供不同的视图。面向业务或者产品，需要以产品架构图去展示。面向技术同学，需要以技术架构图去沟通。不同的视图有助于大家理解同一件事。



何时需要画架构图

在复杂项目开始前画

当你要开始设计一个系统性、完整性的系统时，如果一开始就跳过设计产品架构图、技术架构图，直接开始写文档、画 demo，就很容易发生改了又改，做了又推翻的情况。

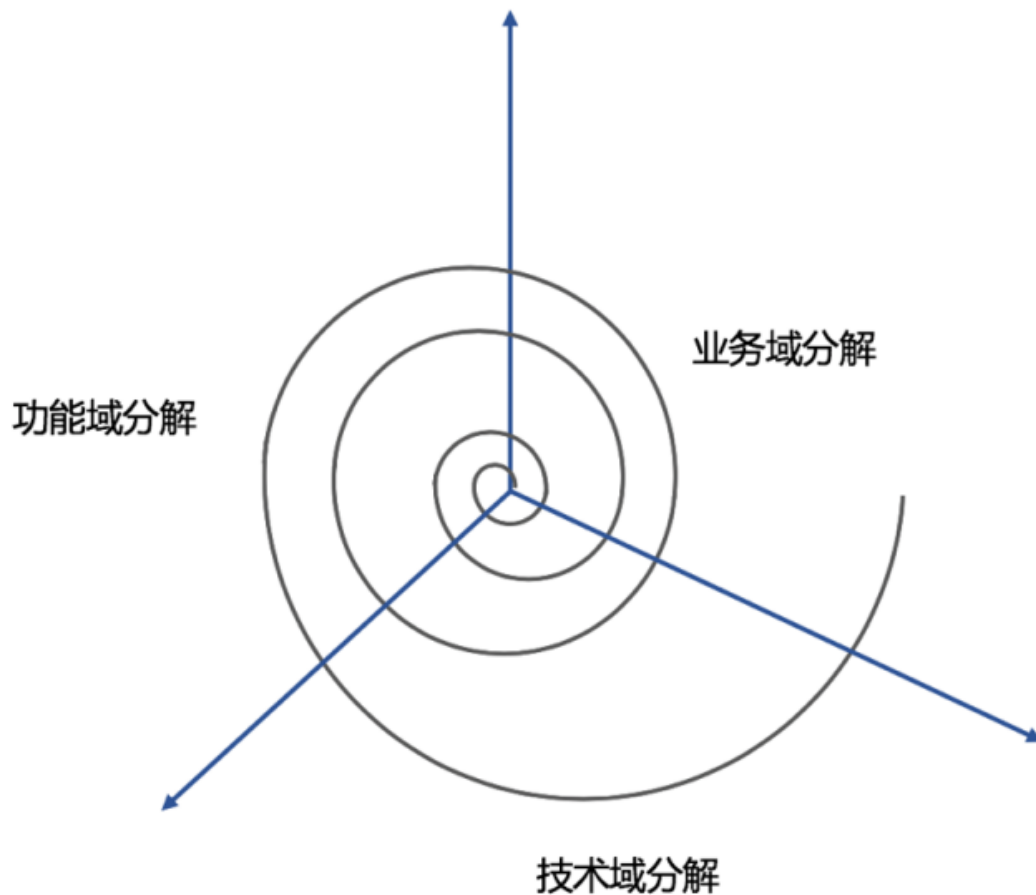
当你觉得是该画的时候 (do it)

如果项目已经进行了一半，或者项目都已经结束了，但自己却从未画过架构图。那么从此刻开始，动手开始画吧。有一句话“种一棵树最好的时间是十年前，其次是现在”。

如何画架构图对复杂的系统，特别是前人没有做过的新系统，通常难以一下子设计出合适的架构。在架构设计的初期，通常都要经历一个不断探索的阶段。

在架构设计过程中，架构分解是必不可少的关键步骤。如何进行架构分解，从哪里入手开始进行分解？这些需要一套架构分解的过程模型和过程方法来指导分解。

从架构域的分类：业务架构、数据架构、产品架构、应用架构、技术架构这 5 个域，依次需要进行架构分解。每个结构域的分解过程，都是一个迭代过程。从无到有、从粗到细、从模糊到清晰，一步步精细化、丰富架构。迭代的过程就是一个否定之否定的过程，随着分解的逐步推进或系统的架构演化，后面的分解除了会识别出新的架构元素，也可能对先前识别出的架构作出调整。整个架构分解的迭代过程，通过画架构图的方式是种非常直观的表现形式。



根据这些架构的分类，依次需要产出业务架构图、数据架构图、产品架构图、应用架构图、技术架构图。每个架构域如何进行设计和架构图的绘制，将通过后续 4 篇文章来阐述。

作者介绍

胡斌，菜鸟网络技术专家，目前负责菜鸟风控系统的建设。曾在淘宝技术部先后负责卖家平台、商家运营等领域。在大规模分布式应用、大数据、架构领域有多年的开发和管理经验。

ArchSummit 全球架构师峰会，精选 100+ 国内外专家技术实践落地案例，包含但不限于 AIOps、微服务架构、数据中台、大前端趋势等火热专题。**8 折**抢票开始了，限时立减 **1760 元**！更多详情请点击【[阅读原文](#)】或者【[扫码](#)】，也可直接联系票务经理灰灰：15600537884（同微信）

 全球架构师峰会 | 北京

让架构不再是难题

☒ 微服务架构

☒ 运筹优化

☒ 短视频架构算法

☒ 数据中台建设



🕒 12月06-07日 📍 北京 · 国际会议中心



前沿技术的传播者
顶尖科技的观察者



点个在看少个 bug 

阅读原文

喜欢此内容的人还喜欢

新产品销售策略研讨会材料分享
埃皮蒂管理咨询

骂人讲究三大技巧, 当领导一定要会骂人！（老板必修）

中层领导力

会计指标应该反映出经营活动 - 读《经营与会计》

唐巧