



ThoughtWorks®

现代软件开发与 设计师的协作

王汝佳

ThoughtWorks 客户体验与产品设计业务线负责人



01

设计的重要性

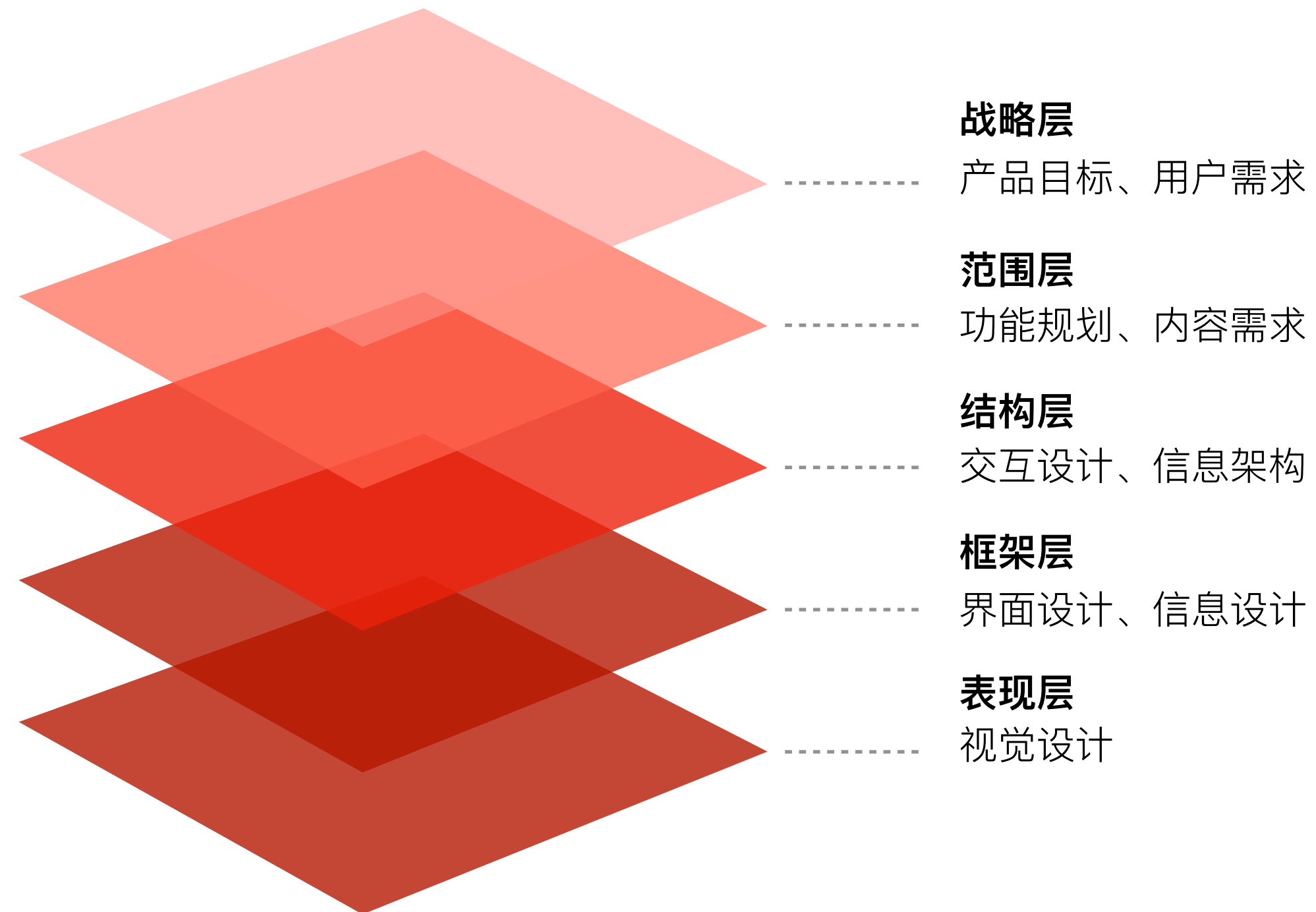
ThoughtWorks®

用户对体验的要求越来越高

Higher expectations of customer experience

ThoughtWorks®

也是用户觉得一个产品好坏的重要因素
Key contributor of satisfaction with product



“这个APP有什么用呢，下载浪费流量，浪费空间”

“为什么不能线上操作？” “为什么不能用我的积分”

“太多步骤了” “花了那么多时间突然误操作了一下全部没了”

“找不到按钮” “找不到信息”

“太丑了”

市场做了积极的调整
The market responds

ThoughtWorks®



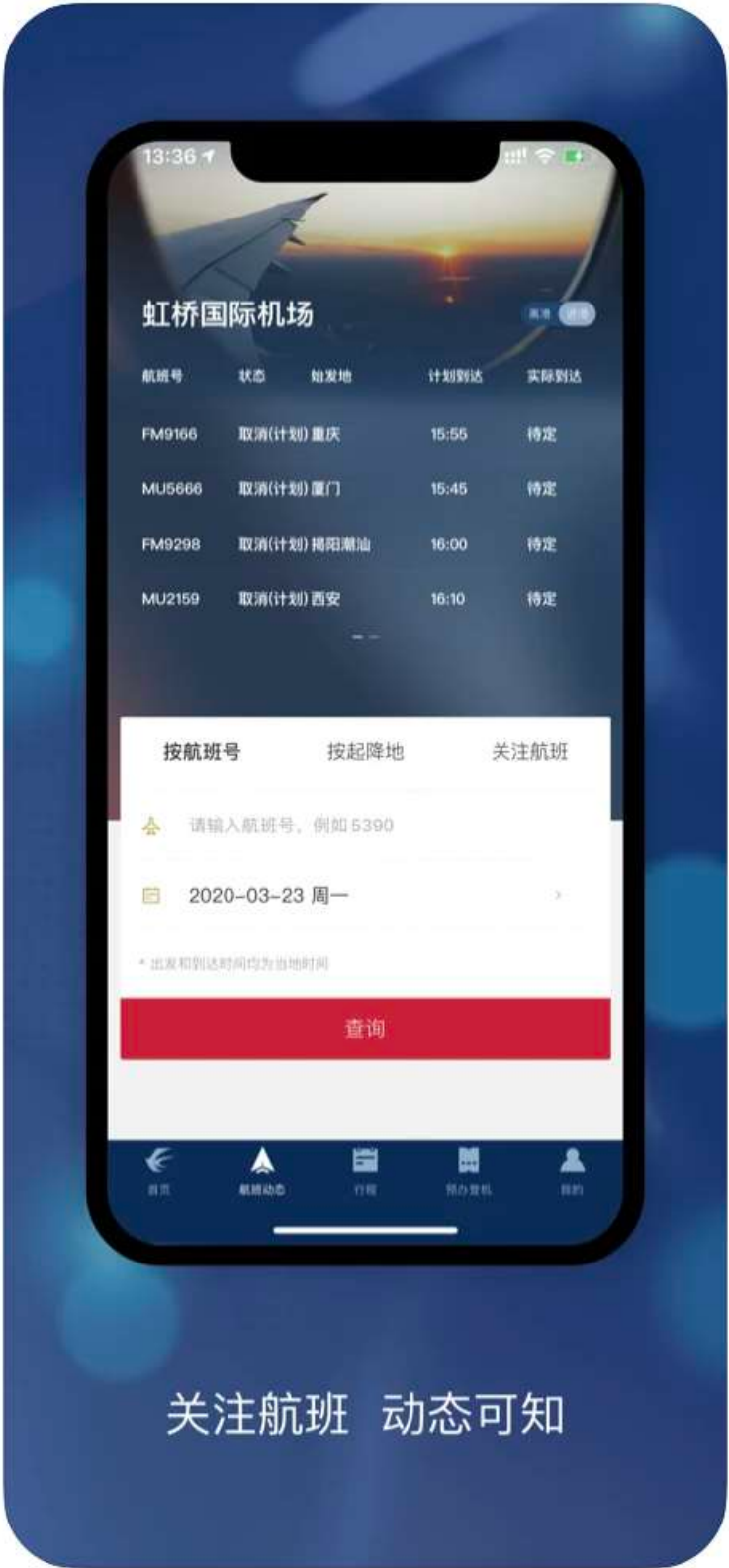
<https://giphy.com/gifs/movie-film-cute-QHwf11y2ZfFSM>



东方航空 iOS App Store 截图



品质出行 触手可及



关注航班 动态可知



安全积分 高端服务

市场做了积极的调整 The market responds

ThoughtWorks®



<https://giphy.com/gifs/movie-film-cute-QHwf11y2ZfFSM>



宜家家居 iOS App Store 截图

市场做了积极的调整 The market responds

ThoughtWorks®

100 : 1

ThoughtWorks 中国 2011年
工程师：设计师 比例

17 : 1

ThoughtWorks 中国 2021年
工程师：设计师 比例



02

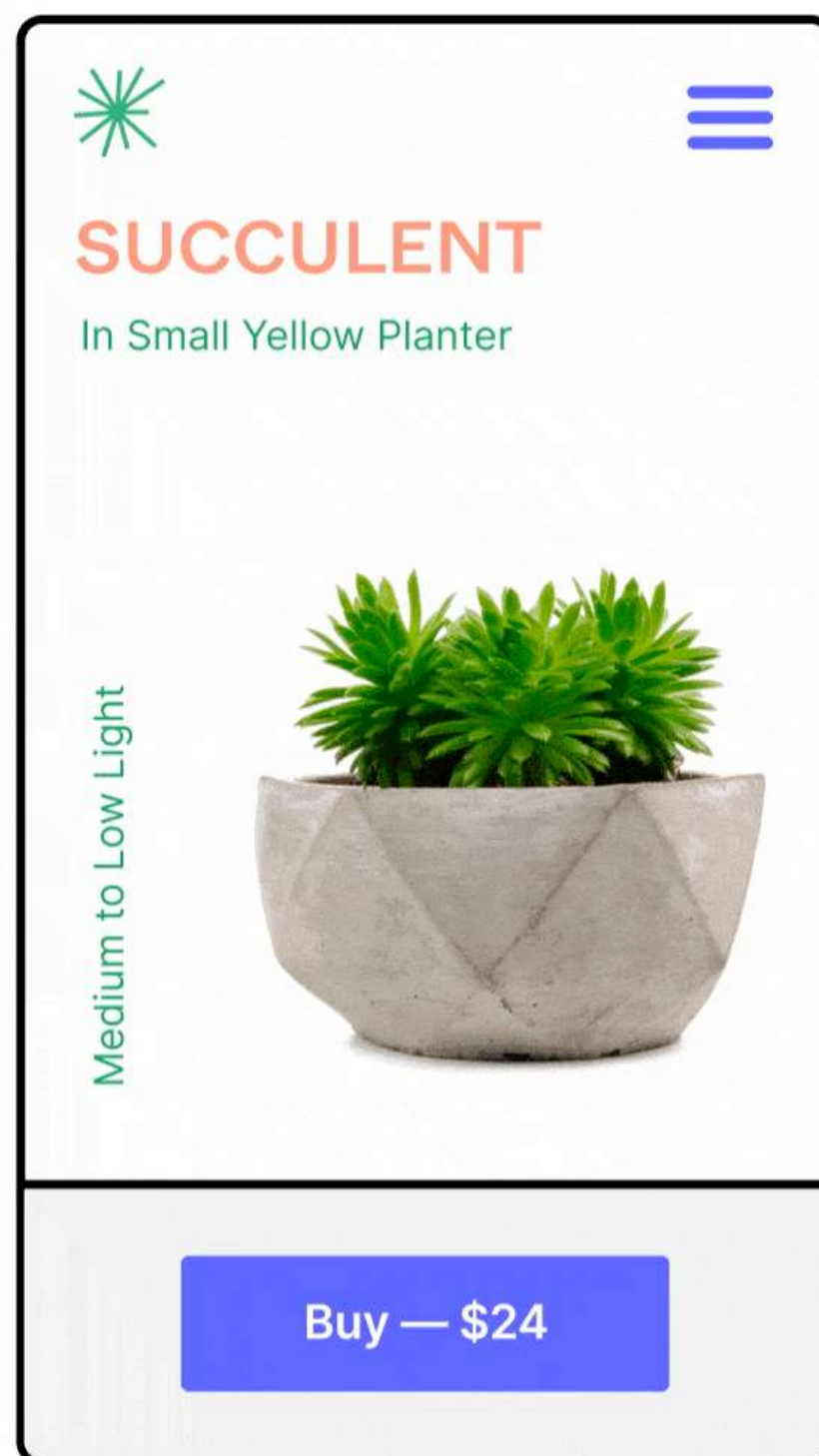
与设计相关的 Blips

ThoughtWorks®

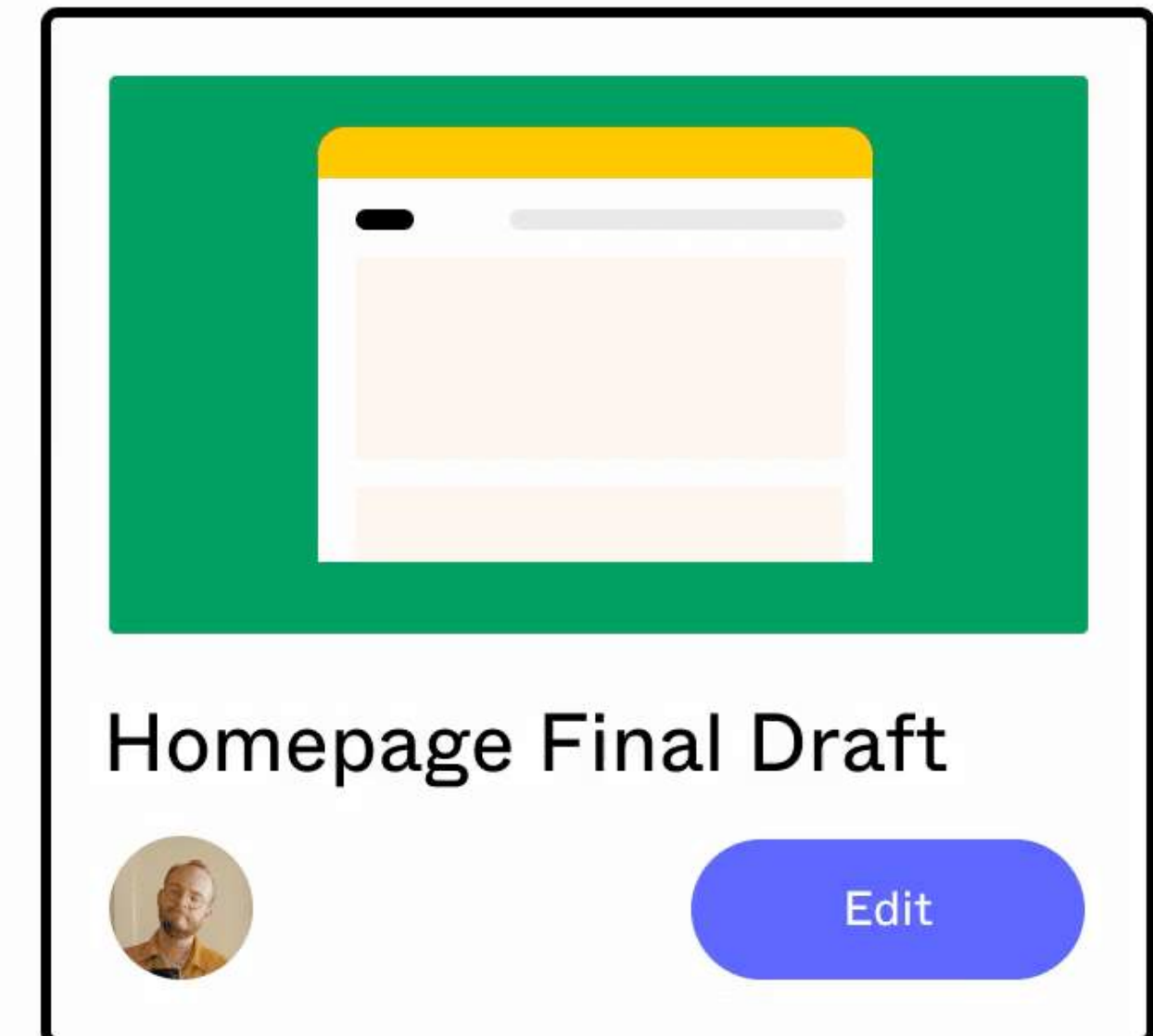
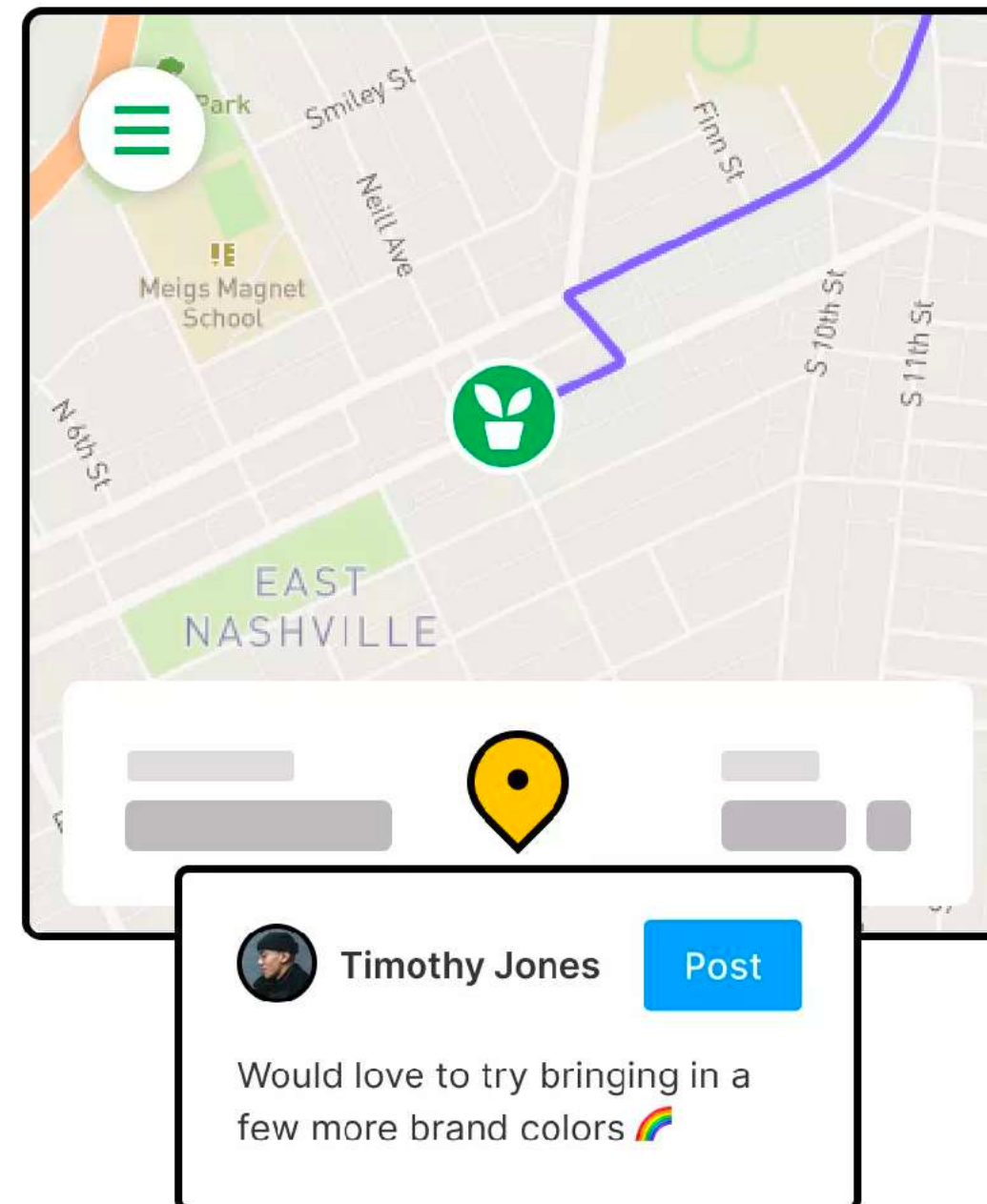
解决不断的改稿和切图：Figma

The final, final, final, FINAL version resolved with Figma

ThoughtWorks®



<https://www.figma.com/>



让动画变得更简单: Lottie

Making animations easier for developers with Lottie

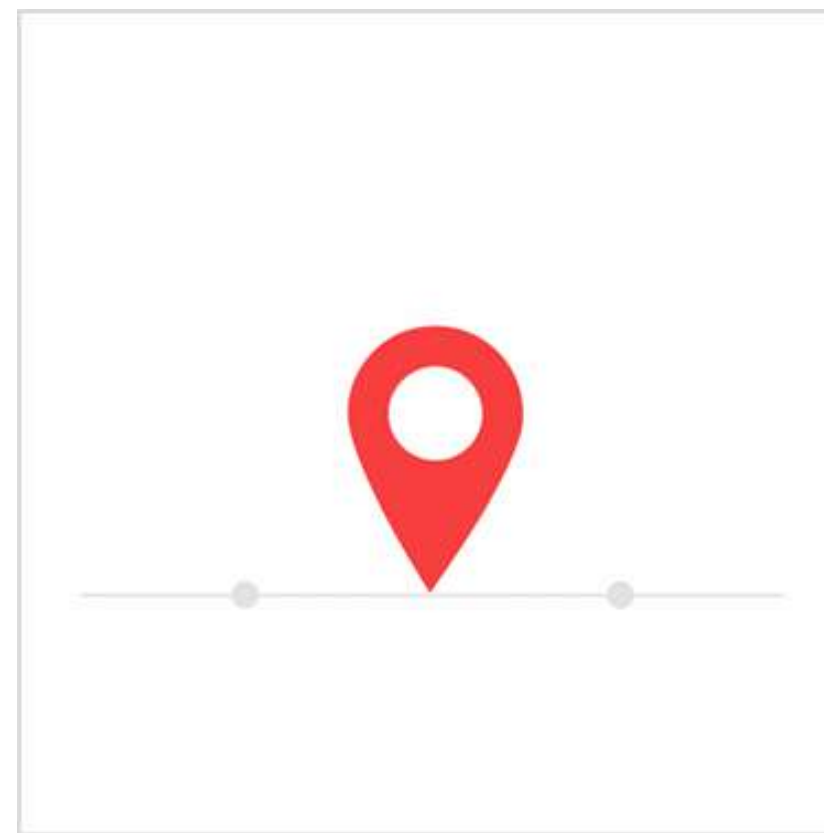
ThoughtWorks®



还记得以前实现这些有多么复杂?



<https://www.deastudios.com/post/breath>



<https://airbnb.design/introducing-lottie/>

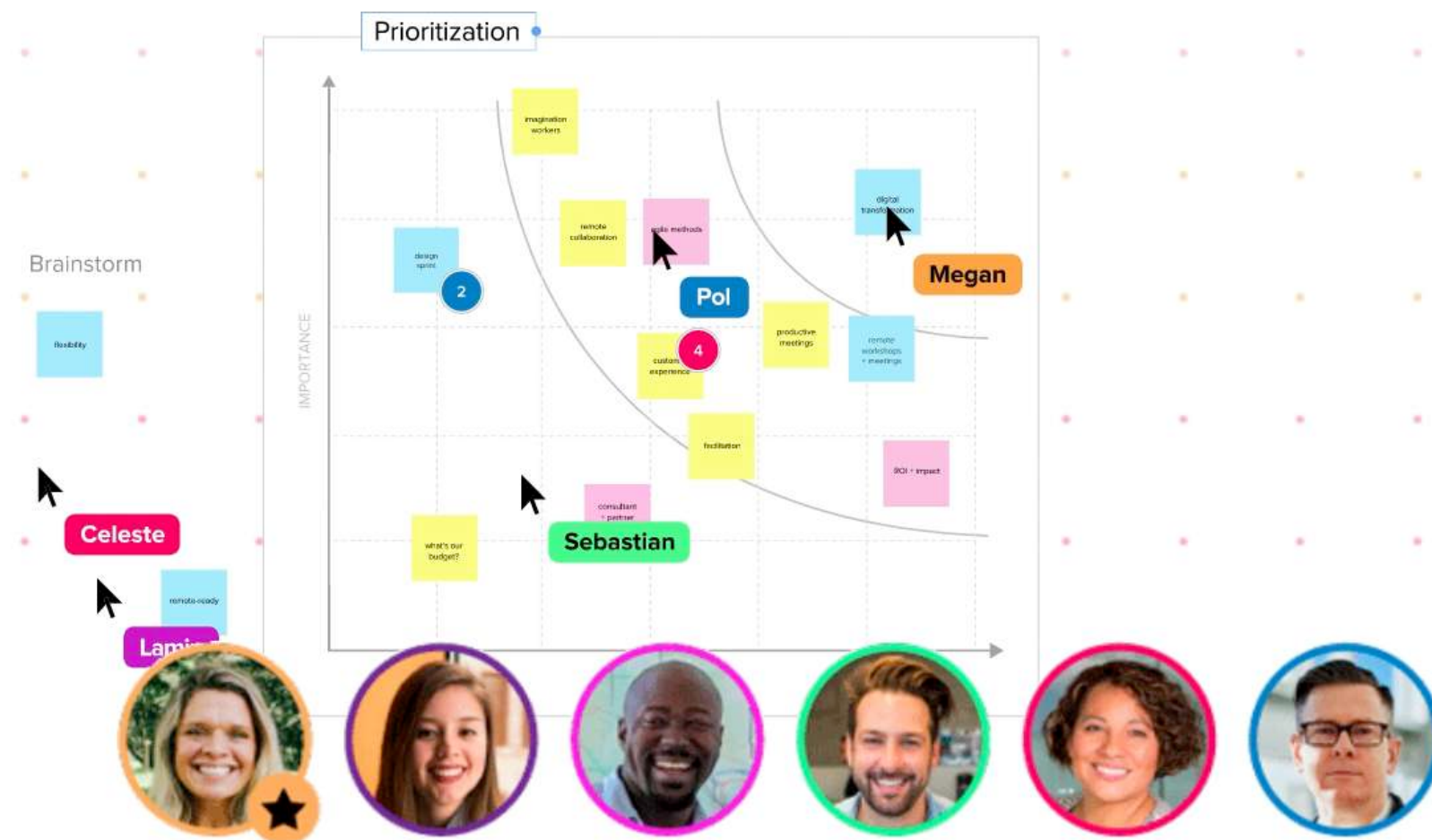


让开发与设计师协作，共创变得更美好高效

Design together when we can't be together: Mural/BeeArt

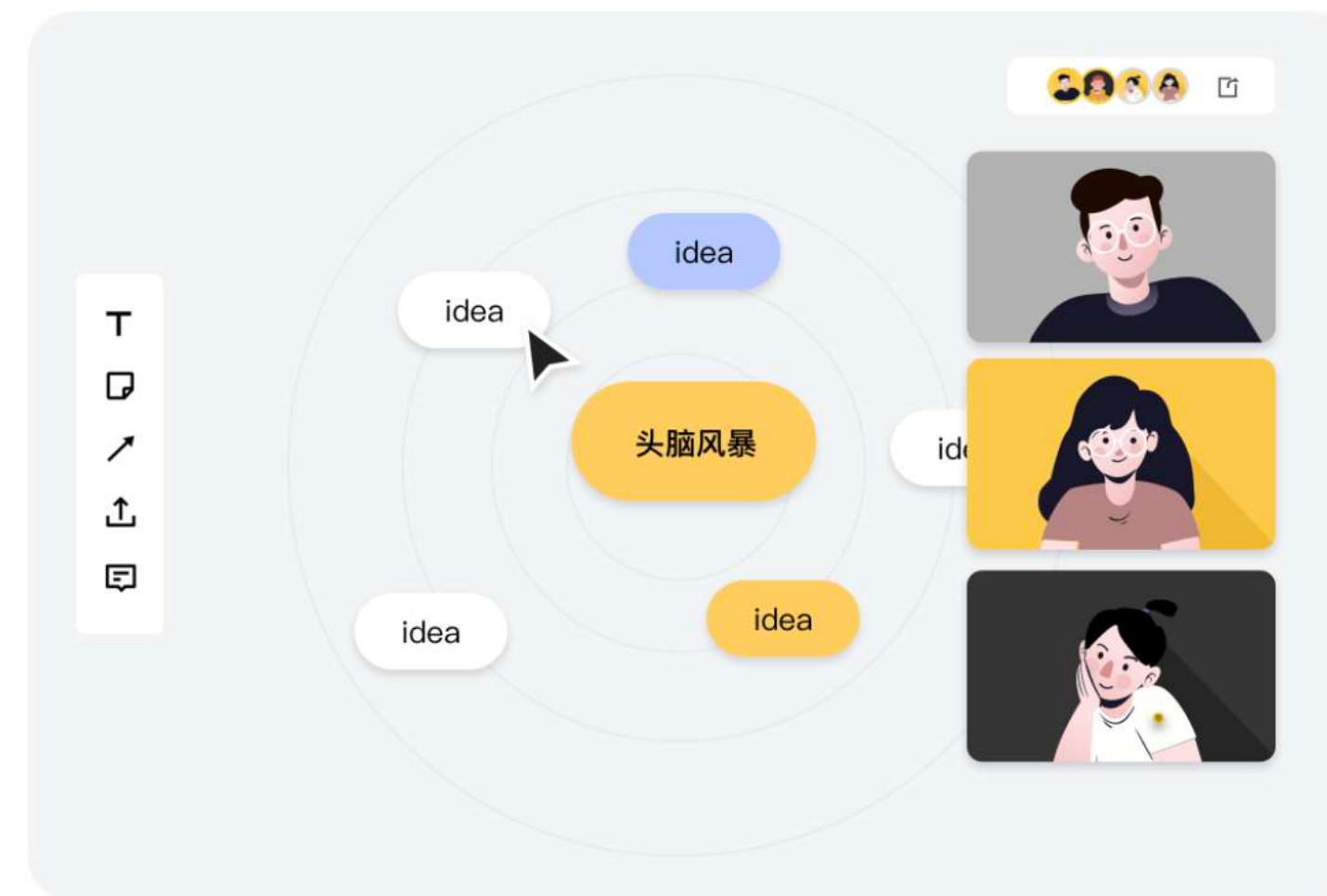
ThoughtWorks®

MURAL



<https://www.mural.co/>

BeeArt



<https://beeart.hello-bees.com/introduce>

协作，是涌现价值的过程

线上会议 & 工作坊

足不出户即可开始一场高效的在线会议 & 工作坊，多人协作，让想法可视并可追踪

◀ **共情共创 & 头脑风暴**

随时随地和你的分布式团队身处一间会议室中，一起激发灵感，共创思考

流程图 & 架构绘制

将复杂的流程与系统架构可视化，让团队中每个人都可以看清价值并为之共同努力

保证高质量，减少设计与开发成本: 设计系统

Ensuring digital product consistency: Design systems

ThoughtWorks®

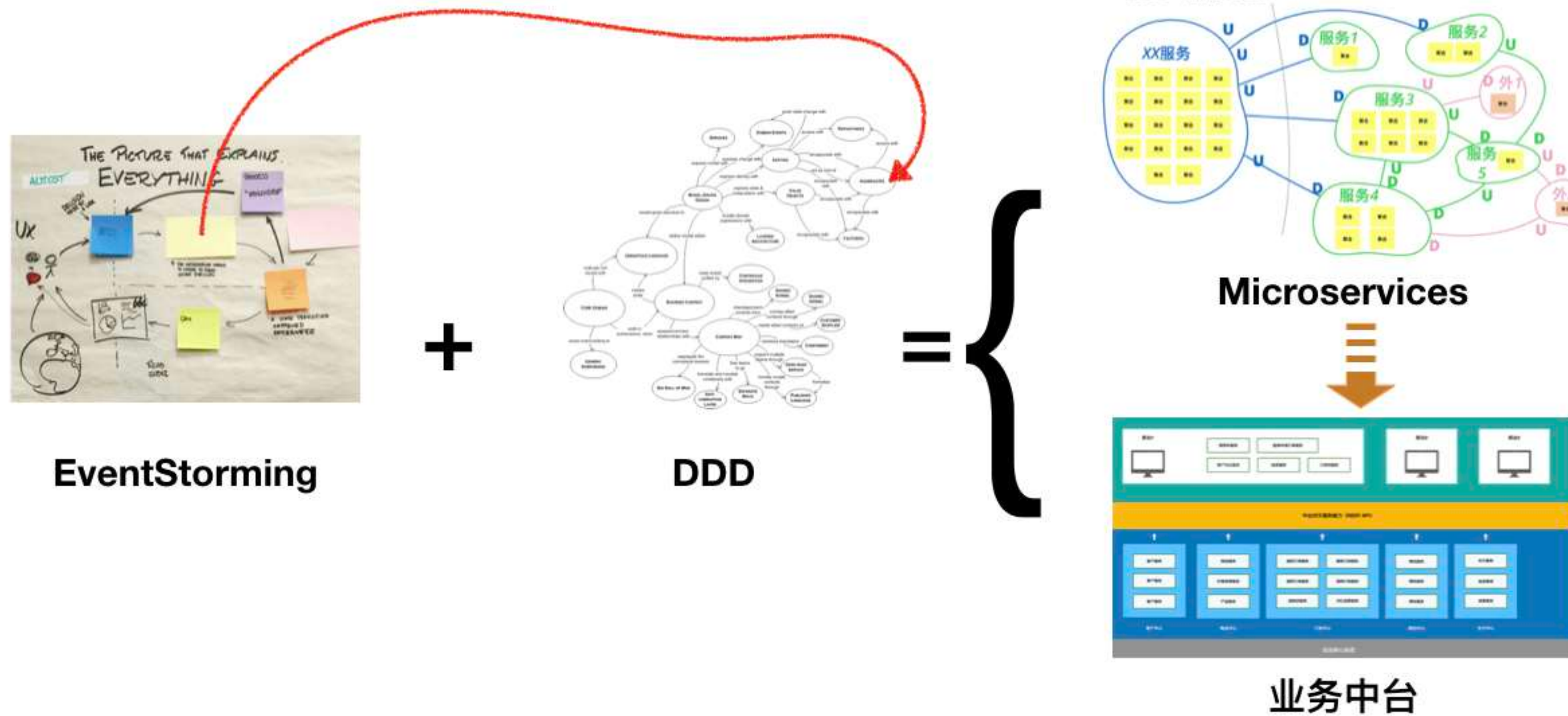


“设计系统包括完整的设计价值观和原则、设计标准、场景定义，以及一套工具包（UI 模式库和代码），用于搭建和组合产品与服务。”

提升技术的业务影响力

Increasing business influence of technologists

ThoughtWorks®



03

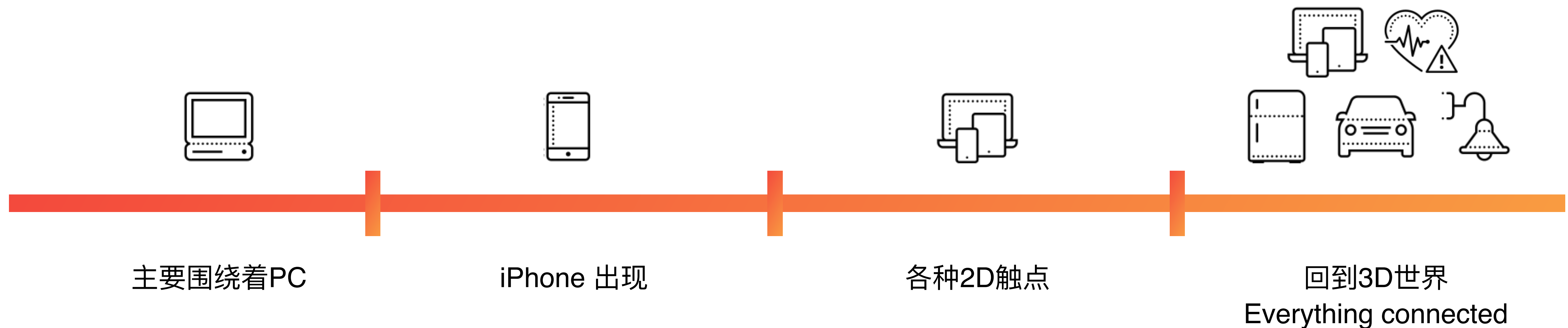
未来的可能性

ThoughtWorks®

不断演进的交互体验 and 用户诉求会带来新的技术重点

Evolving interactions could drive next wave of radar blips

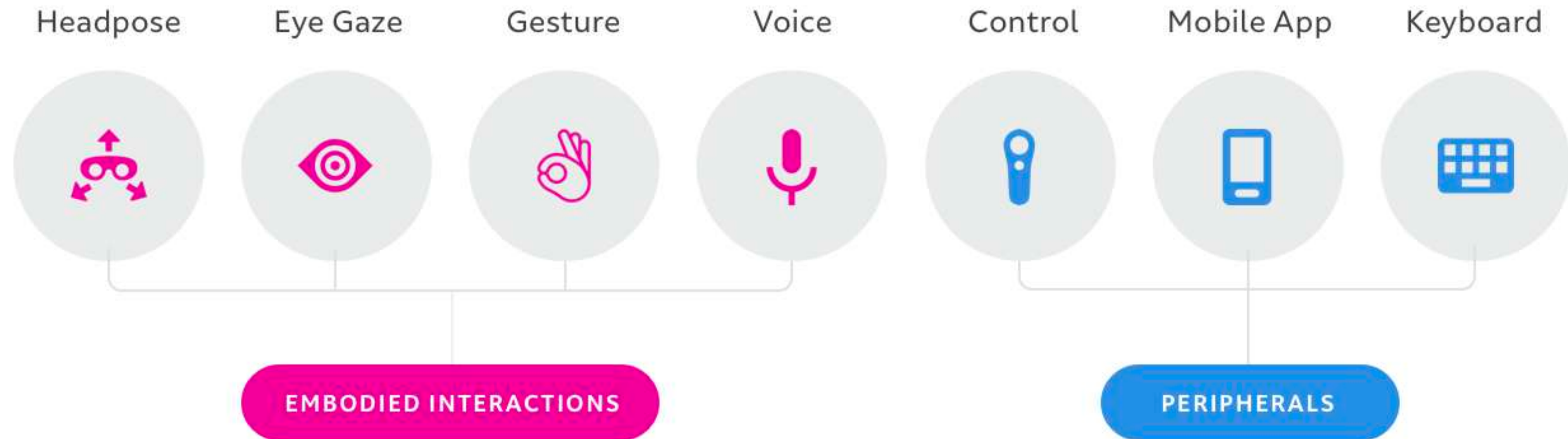
ThoughtWorks®



不断演进的交互体验 and 用户诉求会带来新的技术重点

Evolving interactions could drive next wave of radar blips

ThoughtWorks®



设计师是否会被代码与机器人替代？

Could code or robots replace designers?

ThoughtWorks®

TEXT PROMPT an armchair in the shape of an avocado. an armchair imitating an avocado.

AI-GENERATED
IMAGES



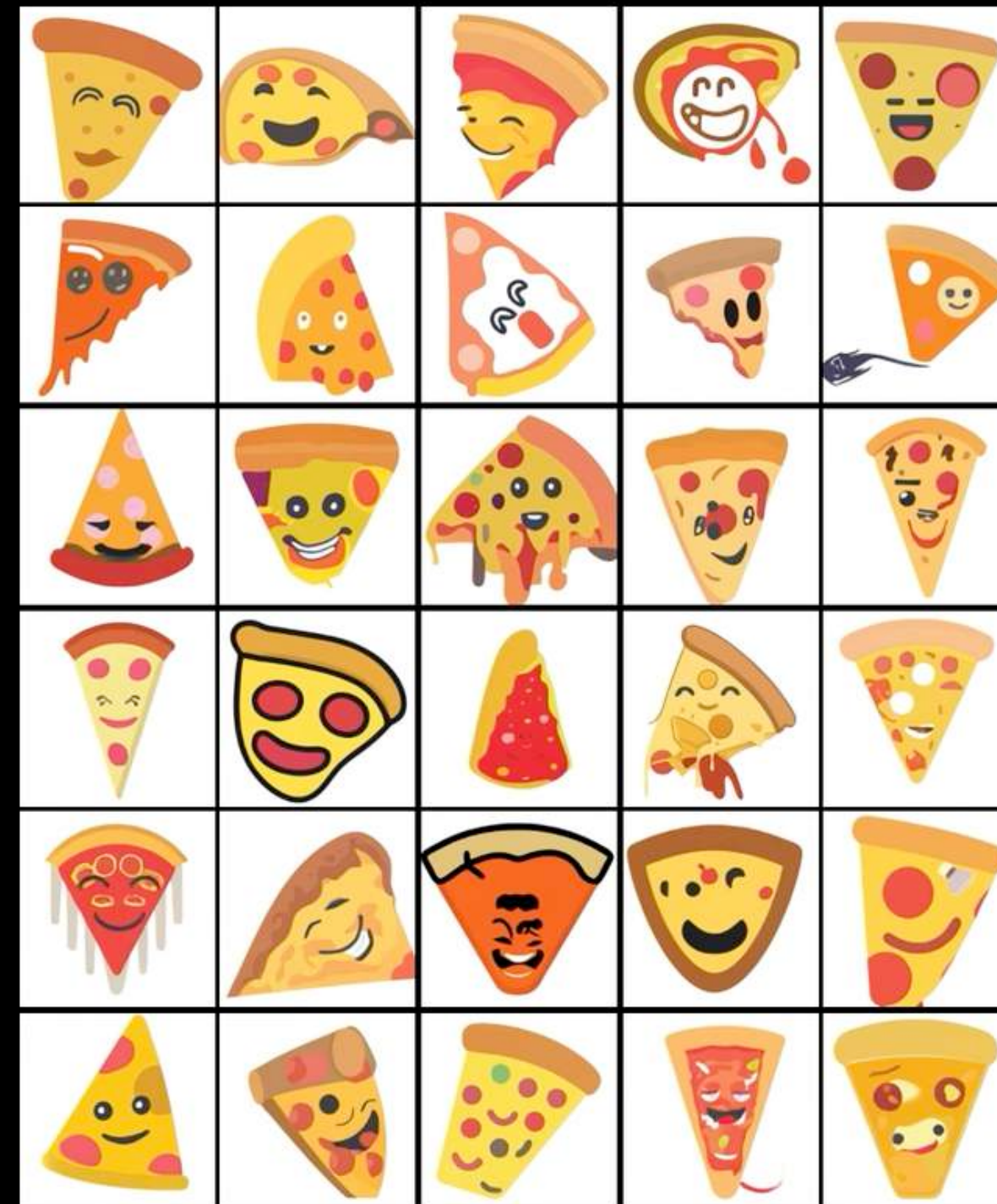
设计师是否会被代码与机器人替代？

Could code or robots replace designers?

ThoughtWorks®

TEXT PROMPT a professional high quality emoji of a happy slice of pepperoni pizza

AI-GENERATED
IMAGES



We find that DALL-E is sometimes able to transfer some emojis to animals and inanimate objects, such as food items. As in the preceding visual, we find that inserting the phrase "professional high quality" before "emoji" sometimes improves the quality and consistency of the results.

不会，直到AI在同理心和理解人类方面能够等同于设计师

Not until they beat us at empathy & understanding humans

ThoughtWorks®

“这些解脱了的 UX设计师会有时间离开办公室，来到使用场景与用户面对面的聊天，体验他们的生活，洞察隐藏的需求；

会有时间踏踏实实地针对每个产品实施可用性实验，而不是仅仅依靠小规模访谈或者专家走查；

会有时间重新审视品牌资产，考虑如何在体验流程的各个触点与消费者建立更好的联系，培养忠诚度；

这些工作已被证明并不是没有价值，只是在现阶段不得不让位于“赶紧画完稿子让产品上线赚钱”，所以价值没有被充分认识到。”

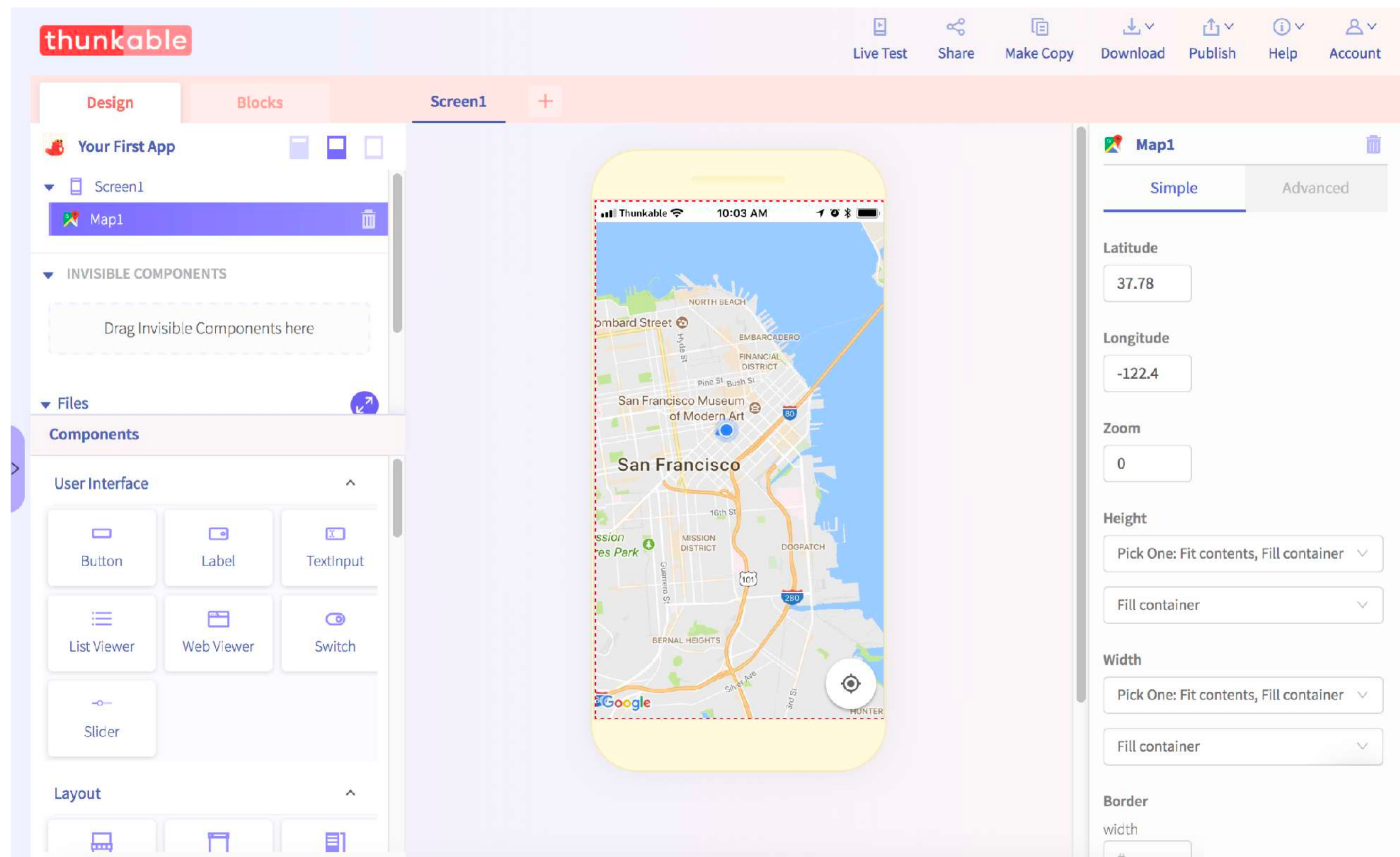
— 孙浩伟，ThoughtWorks洞见



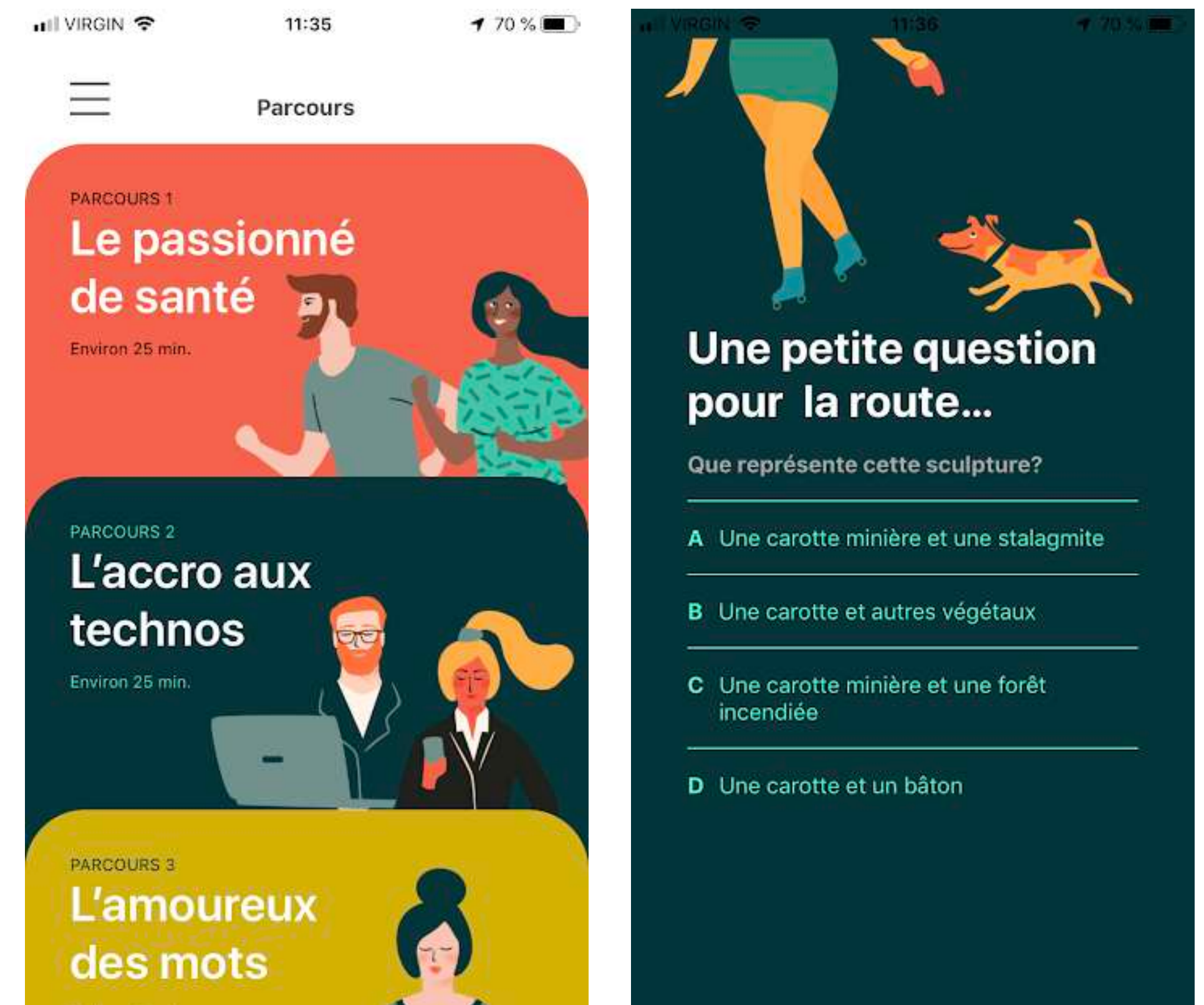
设计师是否会替代前端开？

Could designers replace front-end developers?

ThoughtWorks®



<https://blog.thunkable.com/on-thunkable-anyone-can-build-their-own-mobile-apps-for-android-and-ios-f61abb17be11>



<https://play.google.com/store/apps/details?id=com.gmail.alainchnrd.dambulcopy>

THANK YOU

欲了解详情，请发送邮件至：
rujiaw@thoughtworks.com

ThoughtWorks®



ThoughtWorks®

云原生时代 如何重塑业务

刘冠军

腾讯专家工程师

李铸

ThoughtWorks 中国区生态合作负责人

01 后疫情时代数字化挑战

02 云原生

03 业务重塑

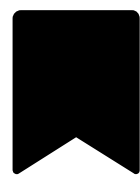
目录

ThoughtWorks®

01

后疫情时代数字化面临的挑战

ThoughtWorks®



工作方式

- 远程协作
- 灵活办公



业务流程

- 线上线下整合
- 虚拟场景、数字孪生



运营平台

- 敏捷创新
- 业务适应性



信息安全

- 基础设施安全可靠
- 隐私信息得以保障

企业服务

消费升级

加速上云

02

云原生

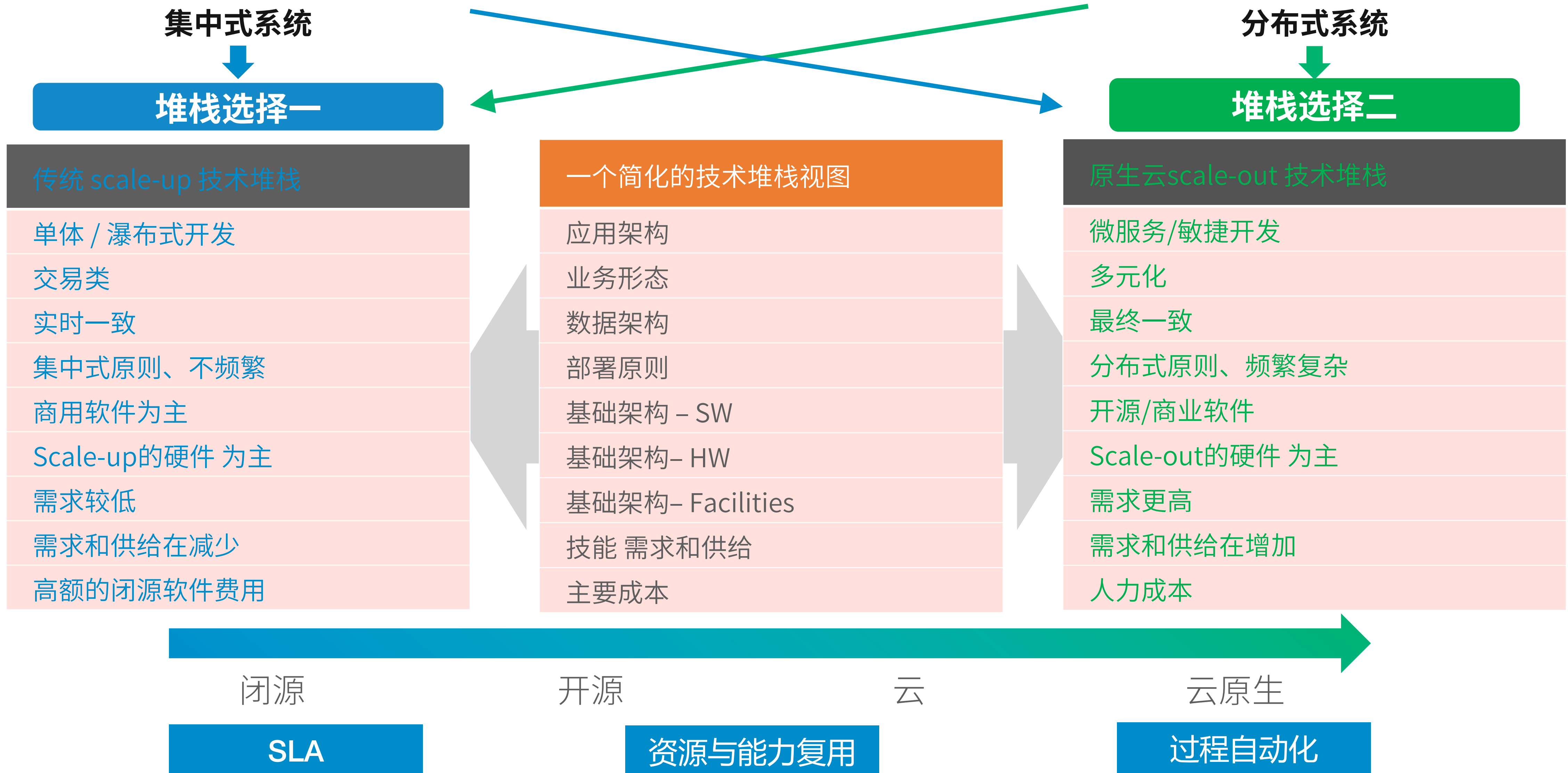
ThoughtWorks®

企业架构演进趋势 – 云原生缘何而起

单机时代	单体架构	IT系统	- 紧耦合，系统复杂、错综交互，动一发而牵全身 - 重复制造各种轮子：OS、DB、Middleware - 完全封闭的架构
PC互联网时代	SOA架构	接口层 逻辑层 数据层	- 通常通过ESB进行系统集成，松耦合 - 需要集中式、计划内停机扩容或更新 - 团队庞大交互负载
移动互联网时代	分布式云计算架构	资源弹性 微服务	- IaaS平台急剧扩张 - 分布式应用支撑高并发 - 开源软件
产业互联网时代	云原生架构	微应用 容器化 自动化	- DevOps: CI, CD, 全自动化 - 可扩展性：自动弹性伸缩 - 高可用：升级、扩容不中断业务 - 可移植性：容器化部署，跨平台

传统集中式 VS 分布式

ThoughtWorks®



什么是云原生？

1

参考CNCf定义

What the CNCf says

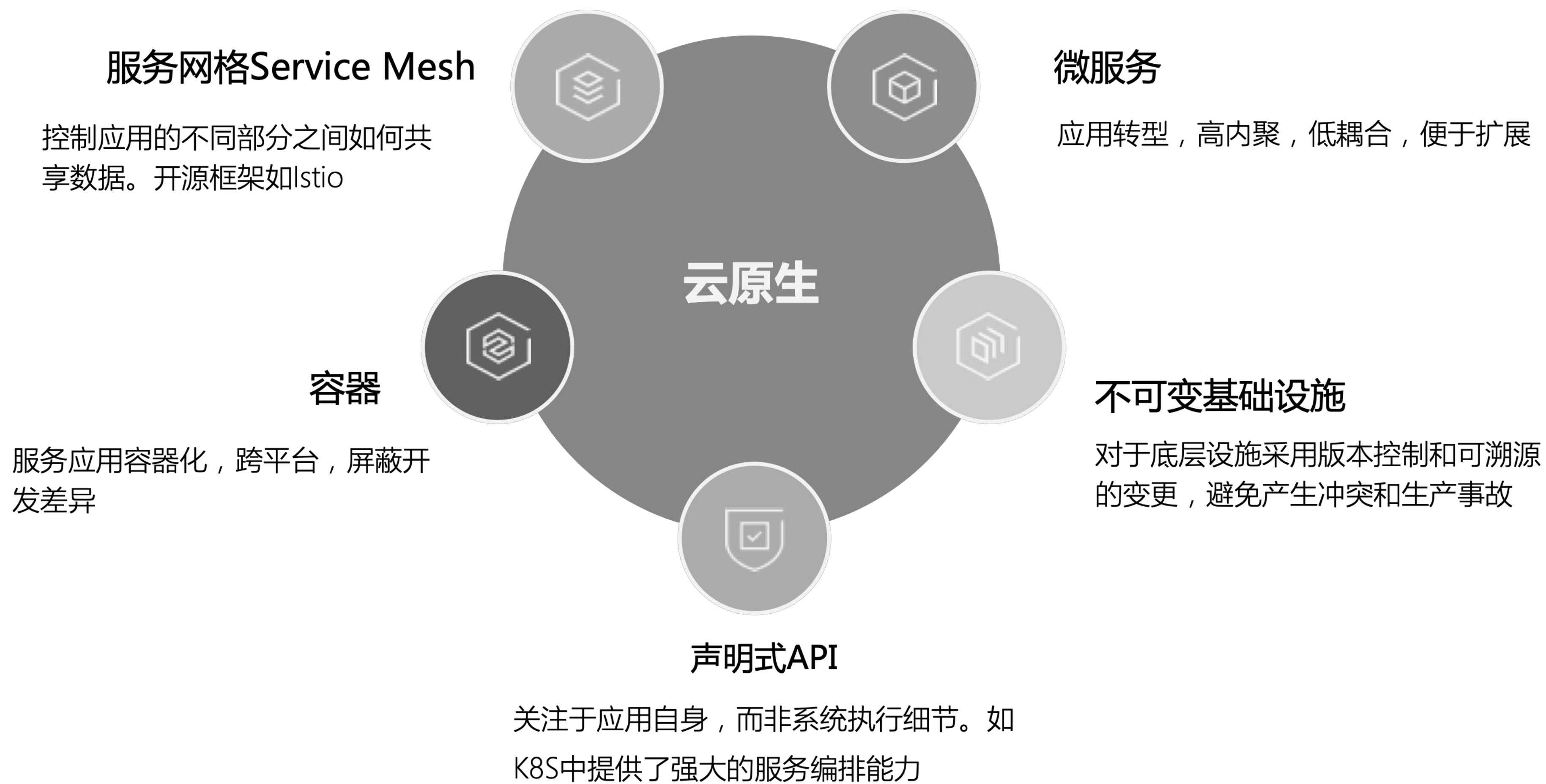
云原生包括容器化封装、自动化管理、面向微服务、服务网格、声明式API

How it looks like

采用开源堆栈（K8s+Docker）进行容器化，基于微服务架构提高灵活性和可维护性，借助敏捷方法、DevOps支持持续迭代和运维自动化，利用云平台设施实现弹性伸缩、动态调度、优化资源利用率。

什么是云原生？

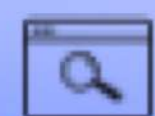
ThoughtWorks®



随着云原生架构的普及，市场上对于容器技术的需求越发旺盛，相对IaaS云，容器云能够提供更敏捷的基础设施，也打通了开发和运维之间的界限。



什么时候需要微服务：需求爆炸式增长，快速迭代，弹性扩缩的业务场景



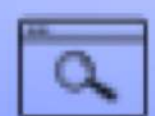
日日可发布

及时响应业务需求，对于特定用户进行版本发布
小版本灰度迭代，金丝雀发布



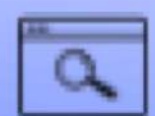
实时可扩缩

基于访问量和请求耗时等业务参数，
进行实时弹性扩缩容



平滑过渡

基于微服务的spring cloud或服务 mesh，
可以进行跨语言，跨系统的和已有业务对话



中台战略

开发的代码或者接口，都可以进行复用，
达到越来越快的效果



1、微服务架构-代码层演化

演进方式：自上而下
蜂窝化代码结构

2、DevOps流水线

自动化开发运维体系

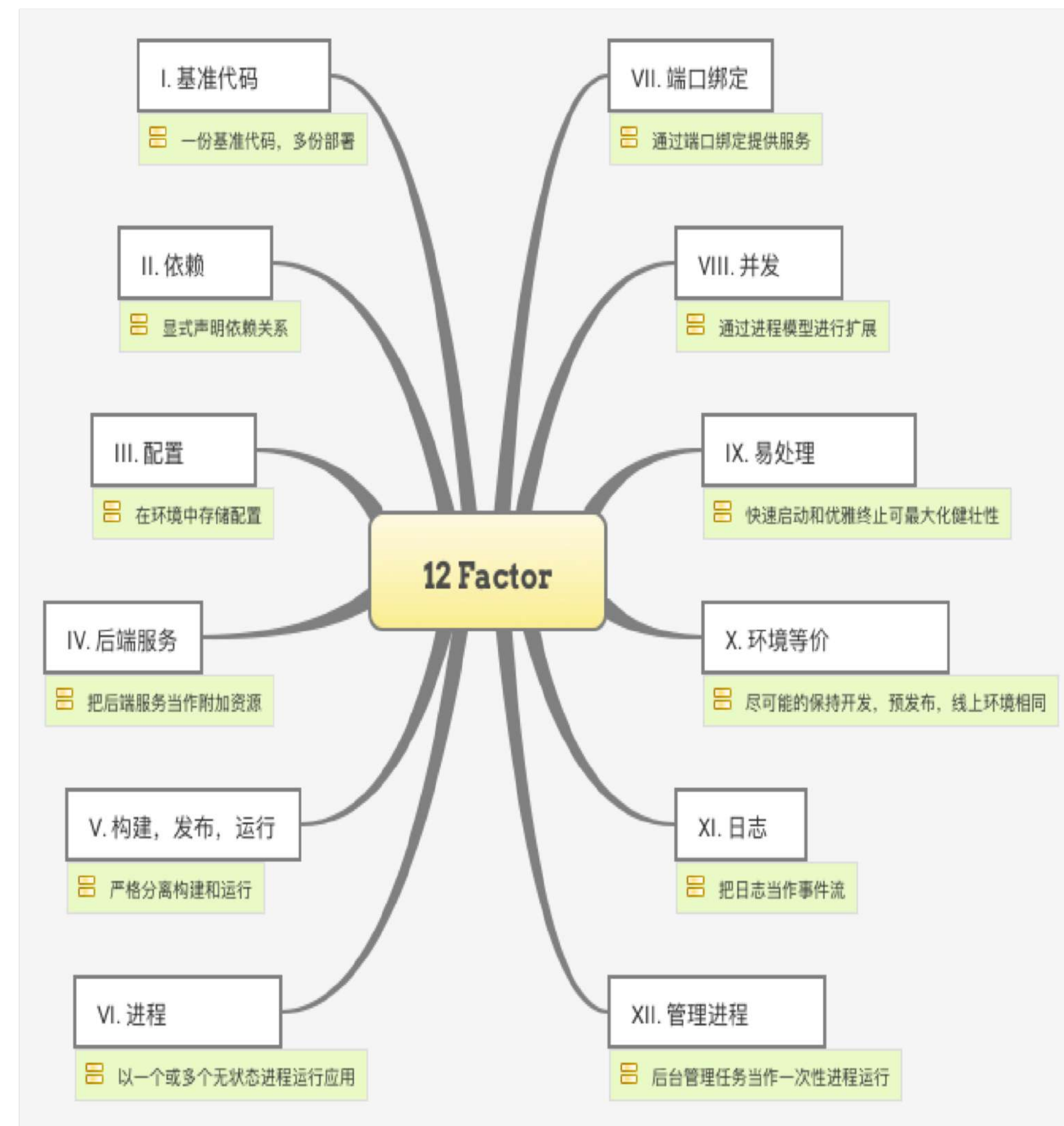
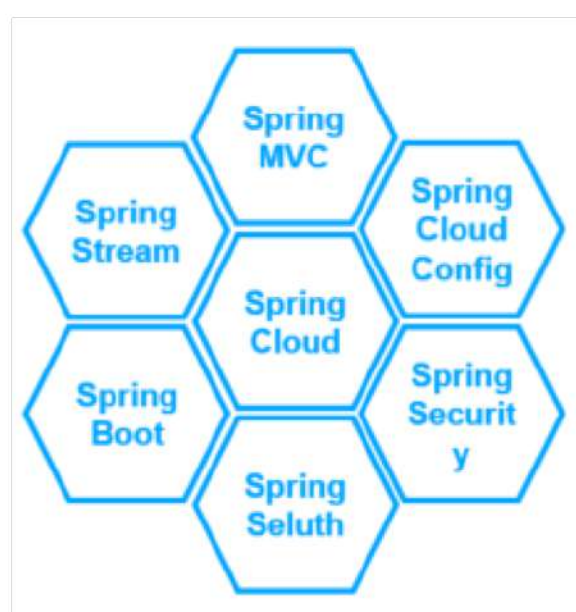
3、容器-资源层演化

演进方式：自下而上
蜂窝化计算资源

云原生应用特征

ThoughtWorks®

- 符合云原生的12个规范
- 简洁、灵活、高效的迭代方式 - 面向微服务
- 可以快速编码、构建、部署和管理 - 面向敏捷开发
- 灵活迁移，不依赖平台和底层设施 - 面向容器技术
- 健壮而规范 - 开源框架



云原生转型两大痛点之一 - 平台建设

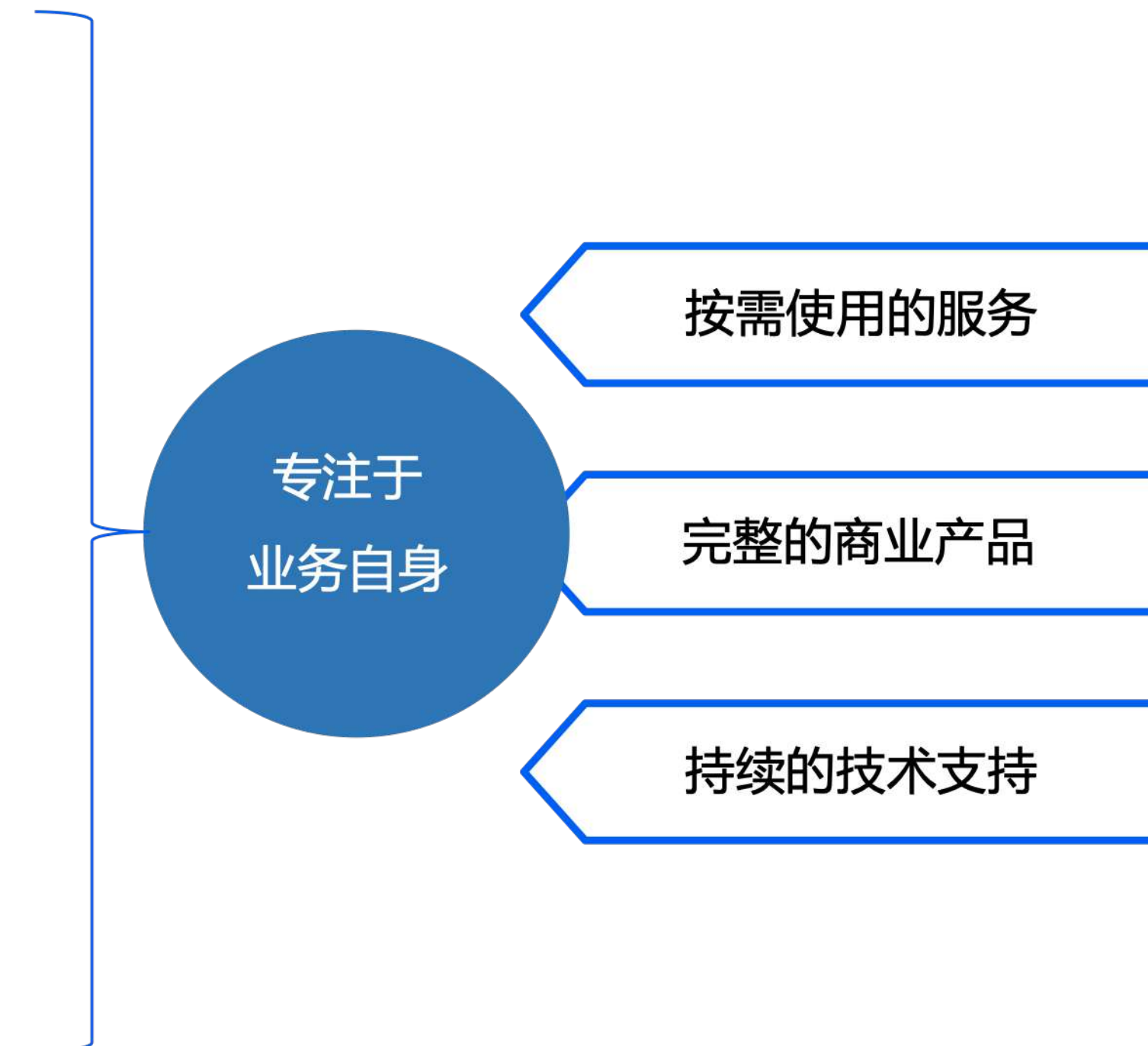
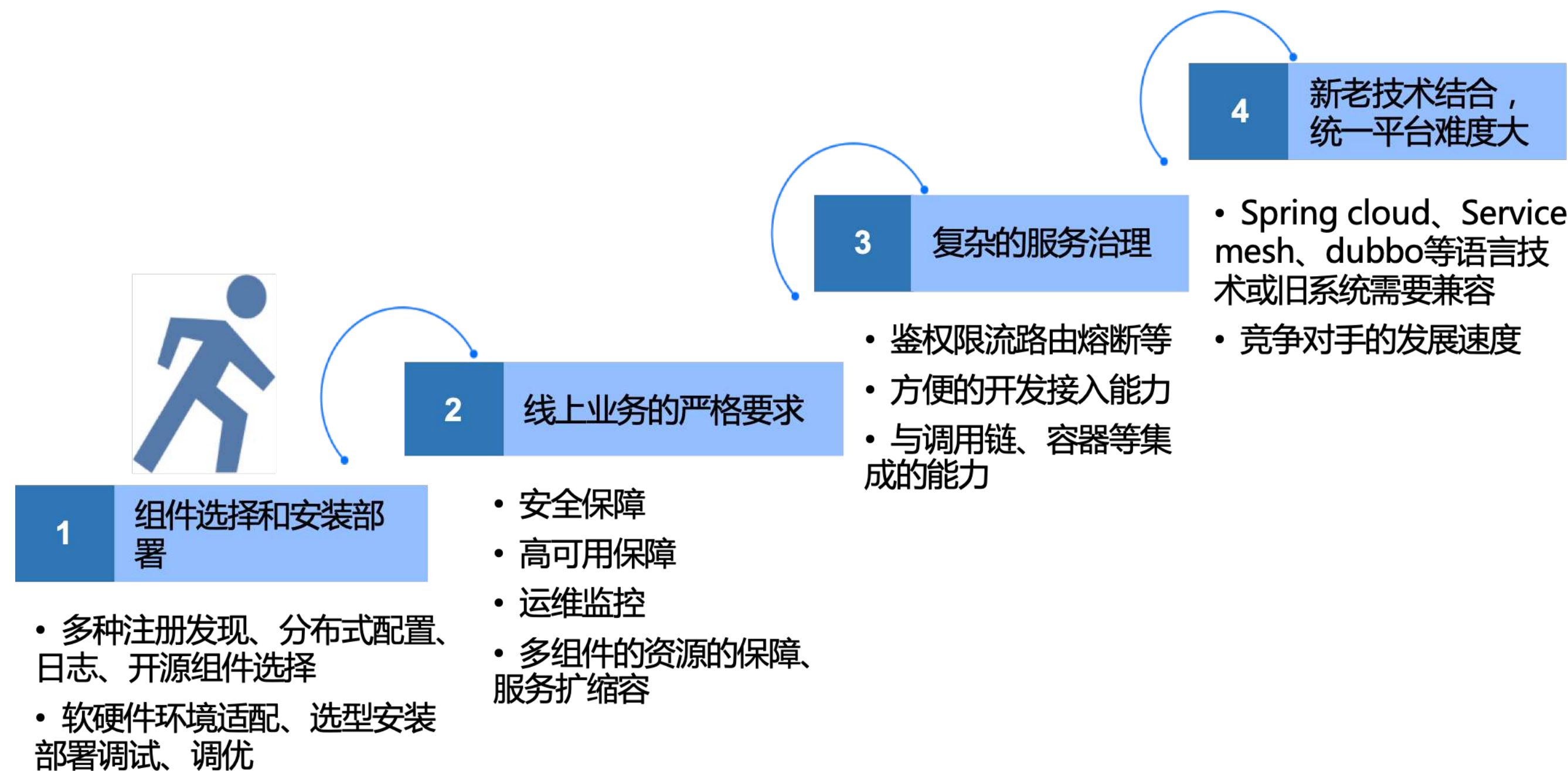
ThoughtWorks®

自建微服务的平台挑战

需要长期的大量研发投入、时间、资源很大，学习成本高，影响业务的发展时机。

云上之旅

将复杂的组件运维托管给云服务
享受微服务平台的便利





PaaS



注册中心



服务治理



数据运营



应用迁移

全面覆盖分布式微服务、分布式消息、云开发等热点技术。提供企业技术中台解决方案。



微服务平台

提供微服务开发框架，服务注册发现，服务治理，应用生命周期管理等整体微服务解决方案



消息队列

丰富的消息队列，金融级TDMQ 大数据Ckafka.



云开发

CloudBase，云端一体化的后端云服务，支持快速构建小程序/小游戏、Web、移动 App 等应用



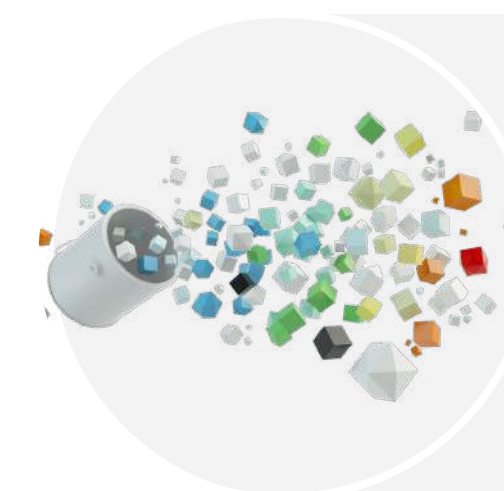
应用性能监控

提供应用性能指标监控、全链路追踪，实时日志分析



分布式调度

负责批量任务的分布式调度，定时任务，任务编排。



分布式事务

提供基于TCC、FMT的事务方案，来解决跨服务的一致性问题



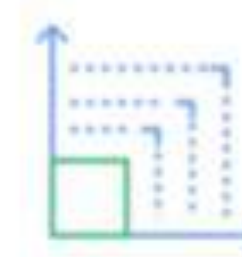
高可用

99.99%+ 的可用性确保整个系统始终连续运行不中断



一致性

在任何异常情况下数据最终一致，确保资金安全



可扩展

支持应用级，数据库级，机房级、地域级的快速扩展

云原生架构两大痛点之二 – 如何拆分？

ThoughtWorks®



03

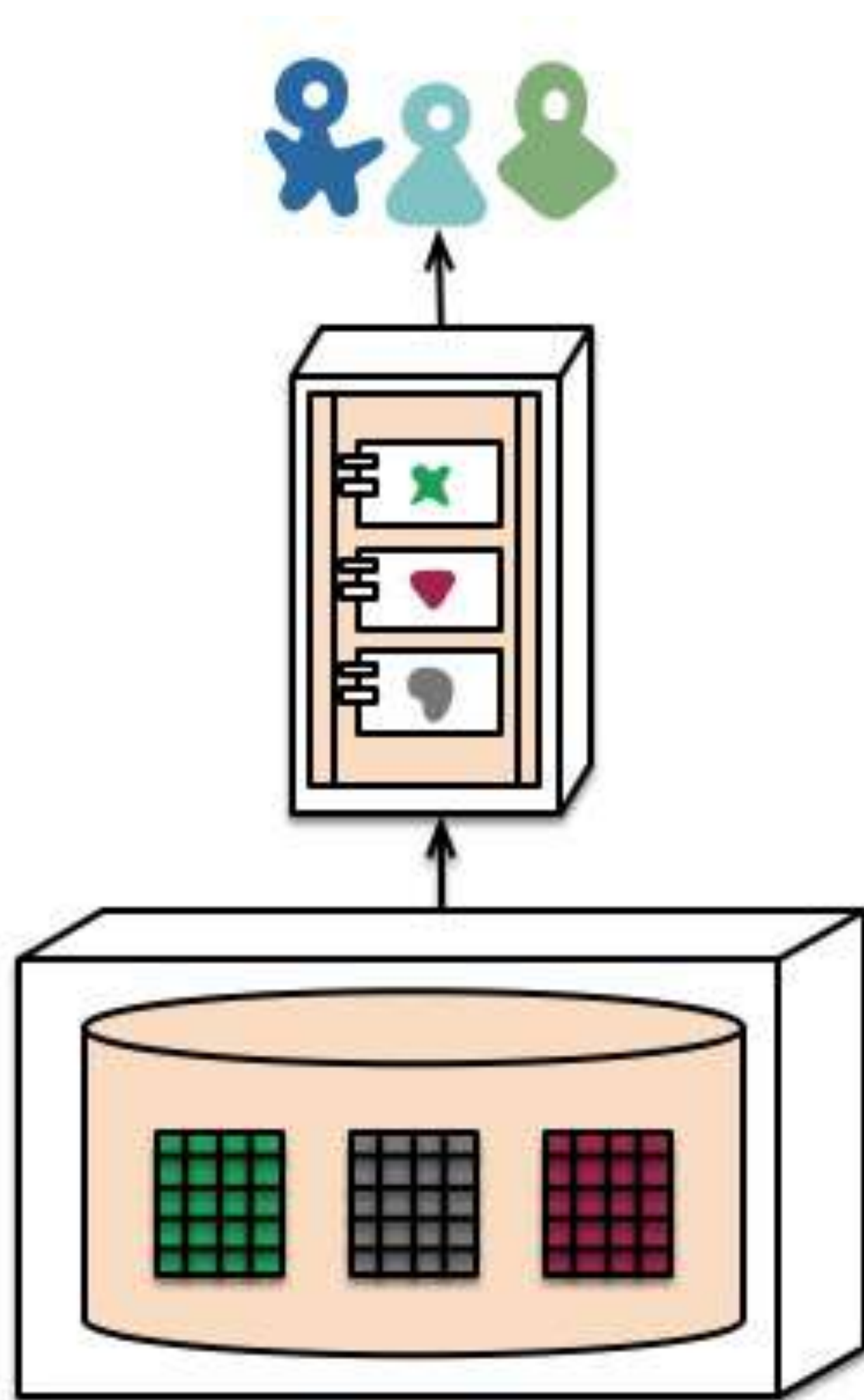
业务重塑

ThoughtWorks®

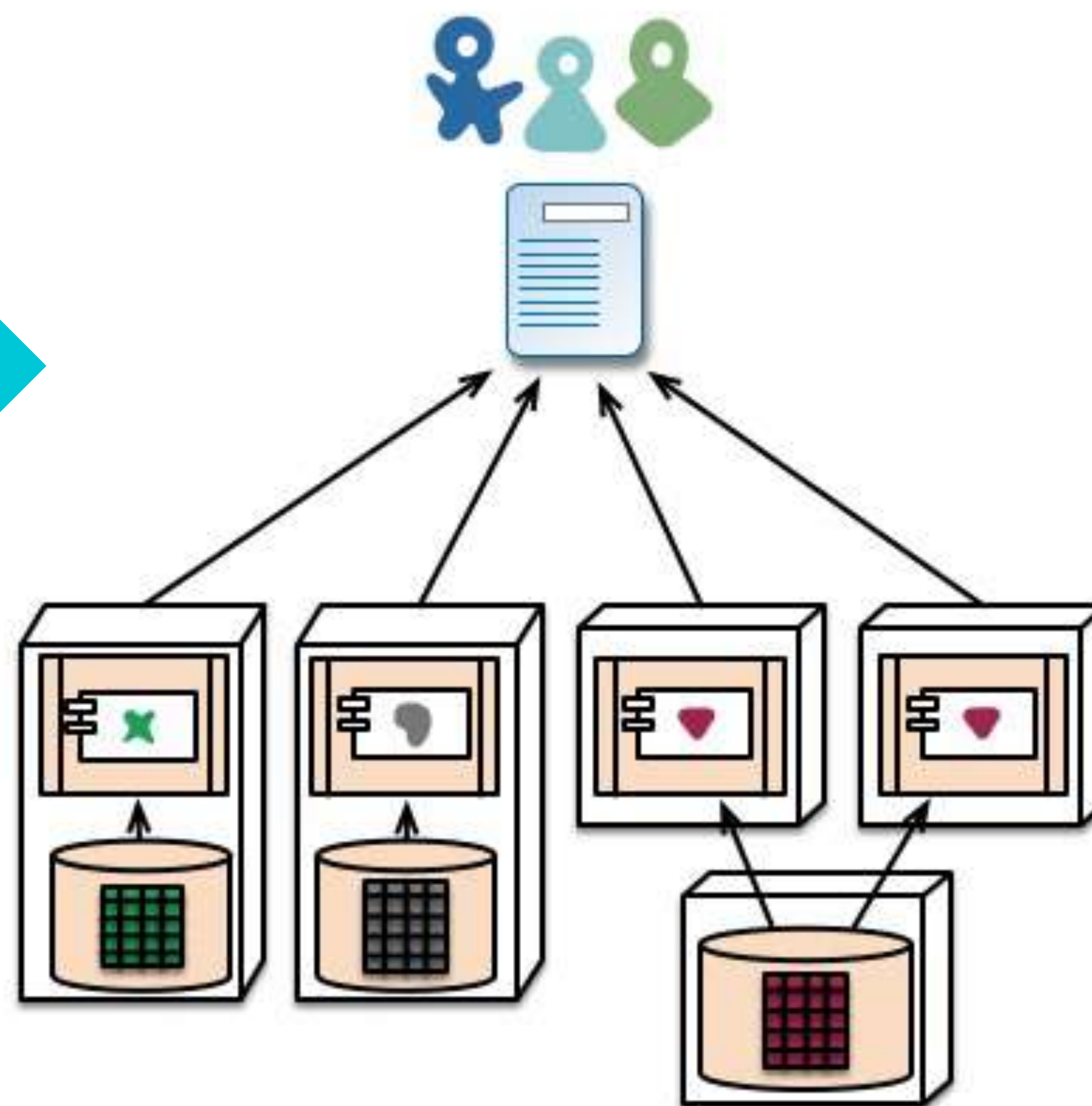


传统架构到微服务架构

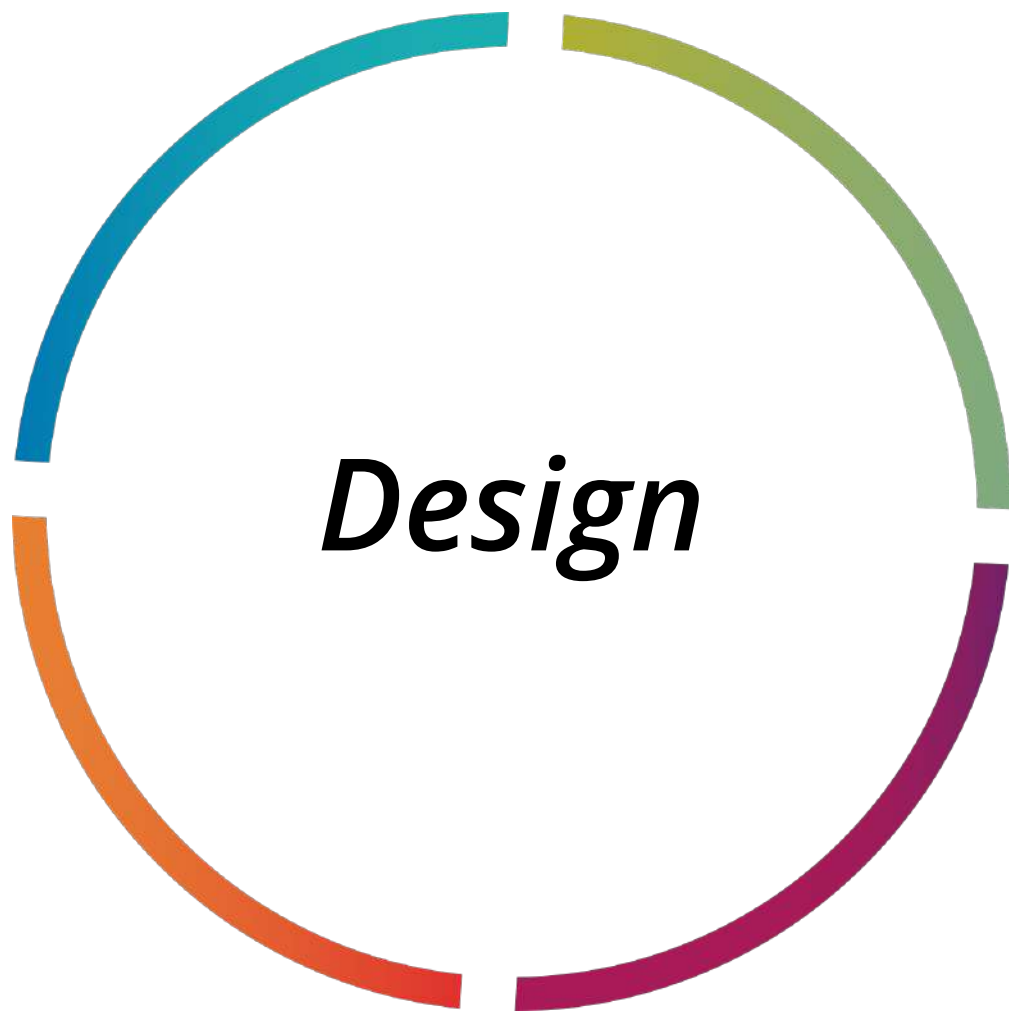
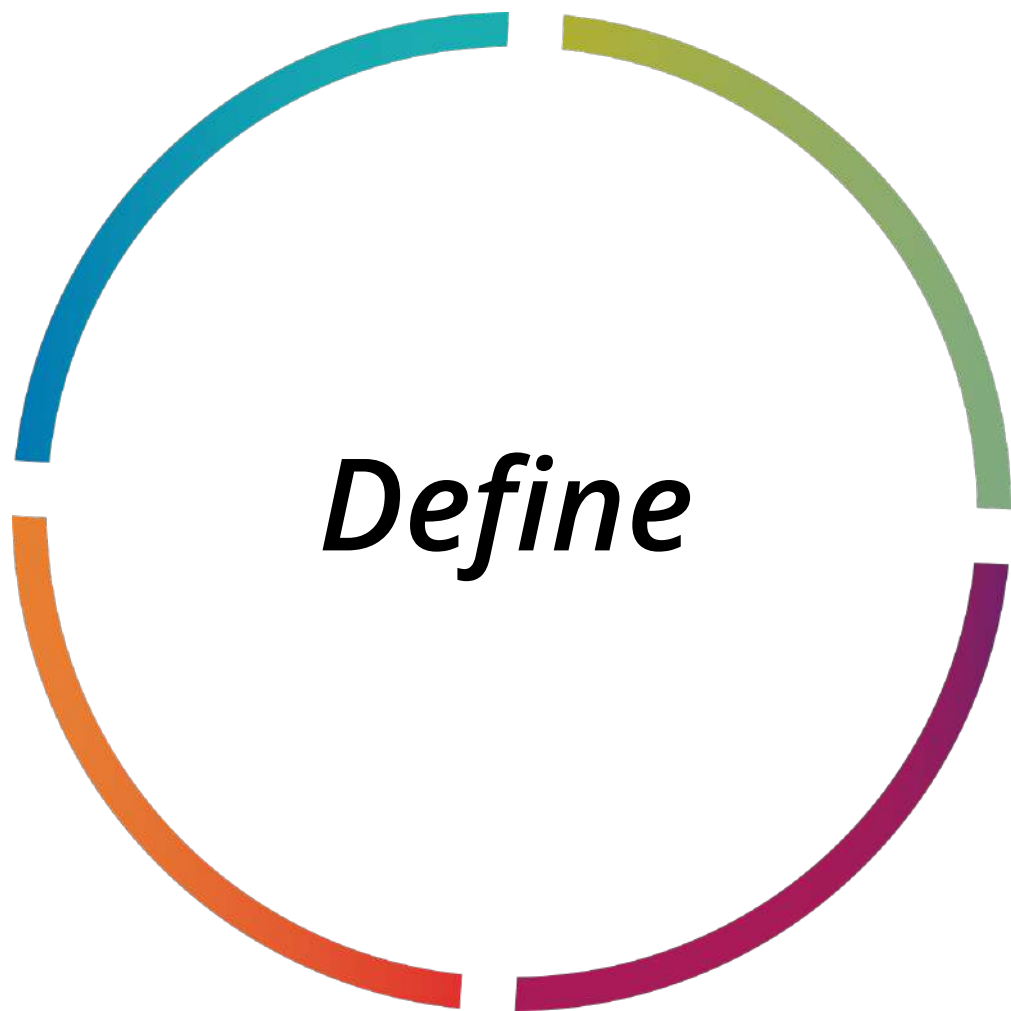
ThoughtWorks®

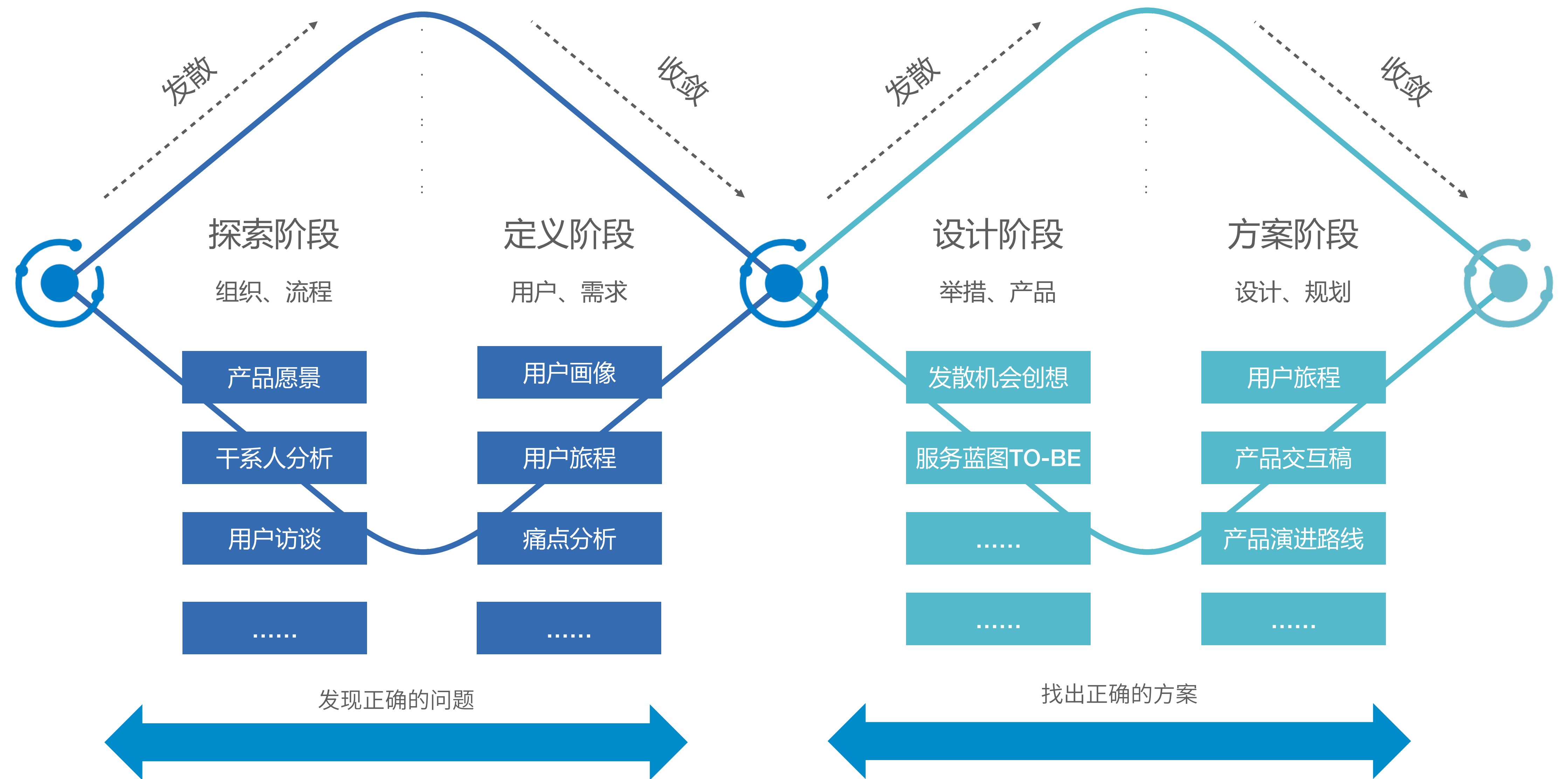


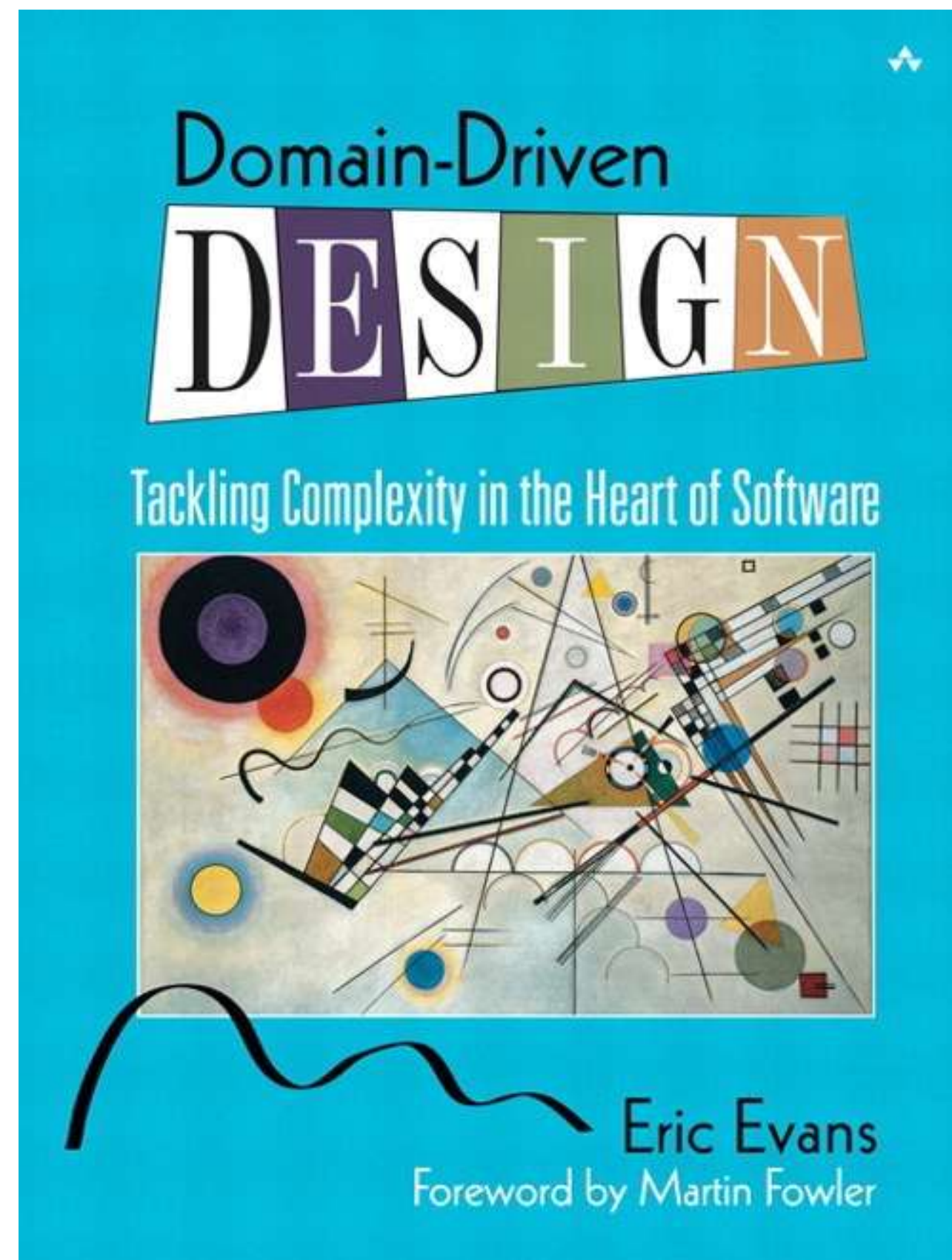
monolith - single database



microservices - application databases







领域驱动设计是一种处理高度复杂域的设计思想，试图分离技术实现的复杂性，围绕**业务概念**构建**领域模型**来控制业务的复杂性，以解决软件难以理解，难以演化等问题。团队应用它可以成功地开发复杂业务软件系统，使系统在增大时仍然保持敏捷。领域驱动设计强调业务与技术深度融合，业务人员与技术人员协同设计。

领域驱动设计实践 ——事件风暴工作坊

ThoughtWorks®

简介

Powerful
Efficient

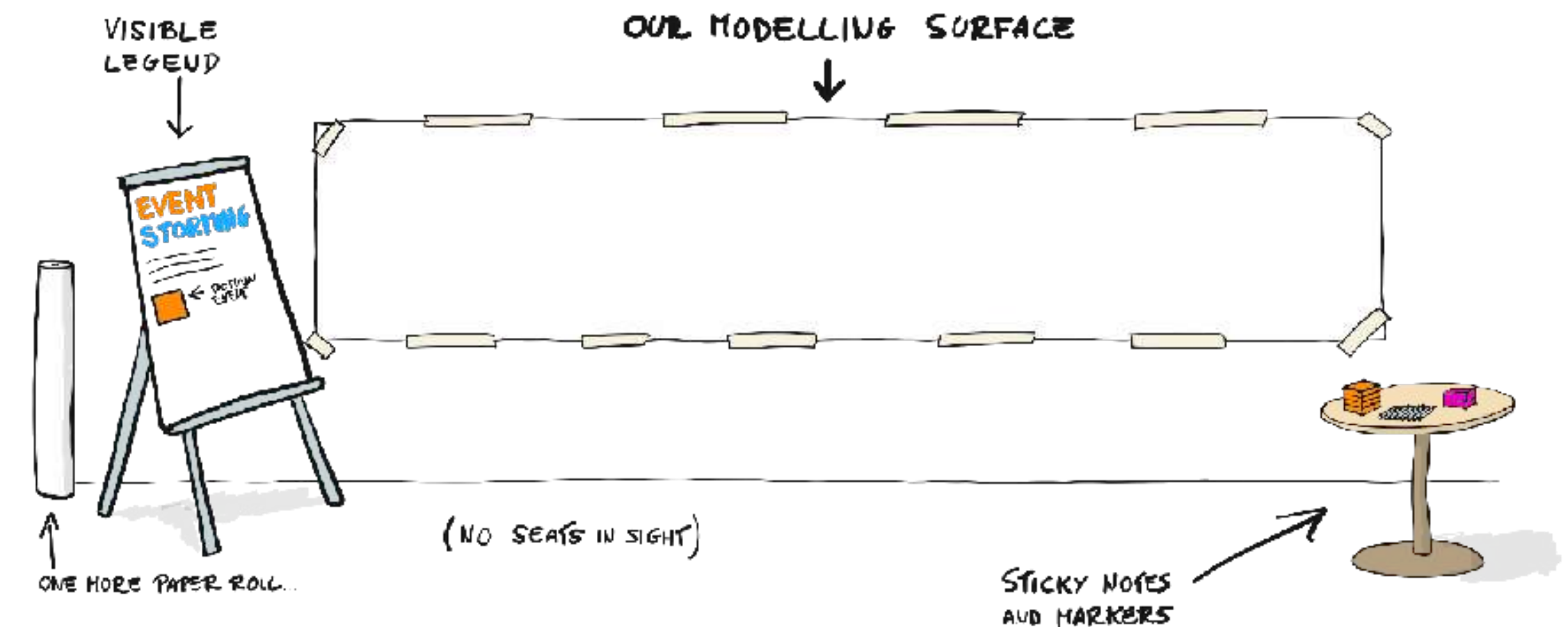
Engaging
Easy

- 事件风暴工作坊（Event Storming）是一种领域驱动设计的实践，是一种快速探索复杂业务领域的方法：

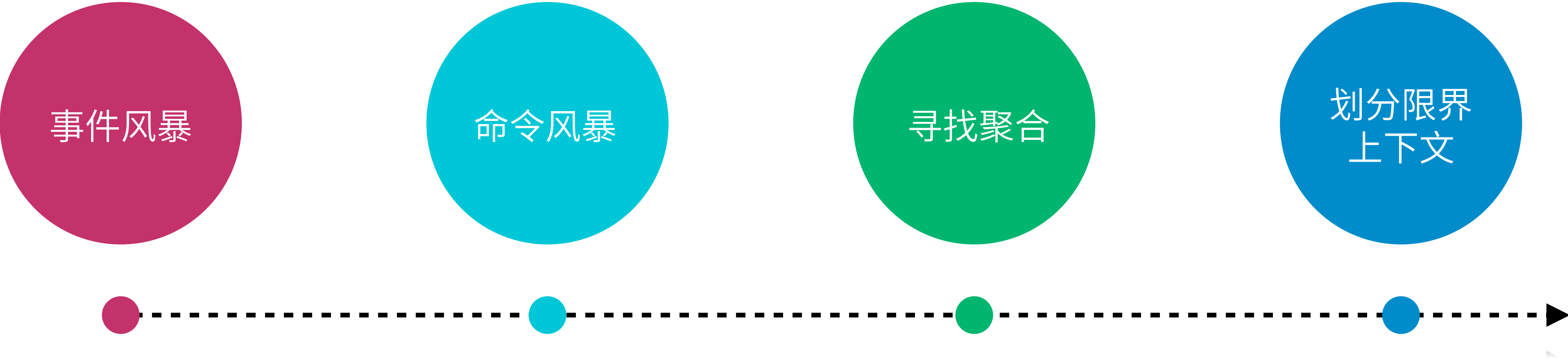
- 最初由Alberto Brandolini 开发，经过DDD社区和TW的很多团队实践验证后。

<http://eventstorming.com/>

活动准备



- 正确的人：业务人员，领域专家，技术人员，架构师，测试人员等关键角色都要参与其中
- 开放空间：有足够的空间可以将事件流可视化，让人们可以交互讨论
- 彩色即时贴：至少三种颜色



业务重塑方案实践总览

ThoughtWorks®

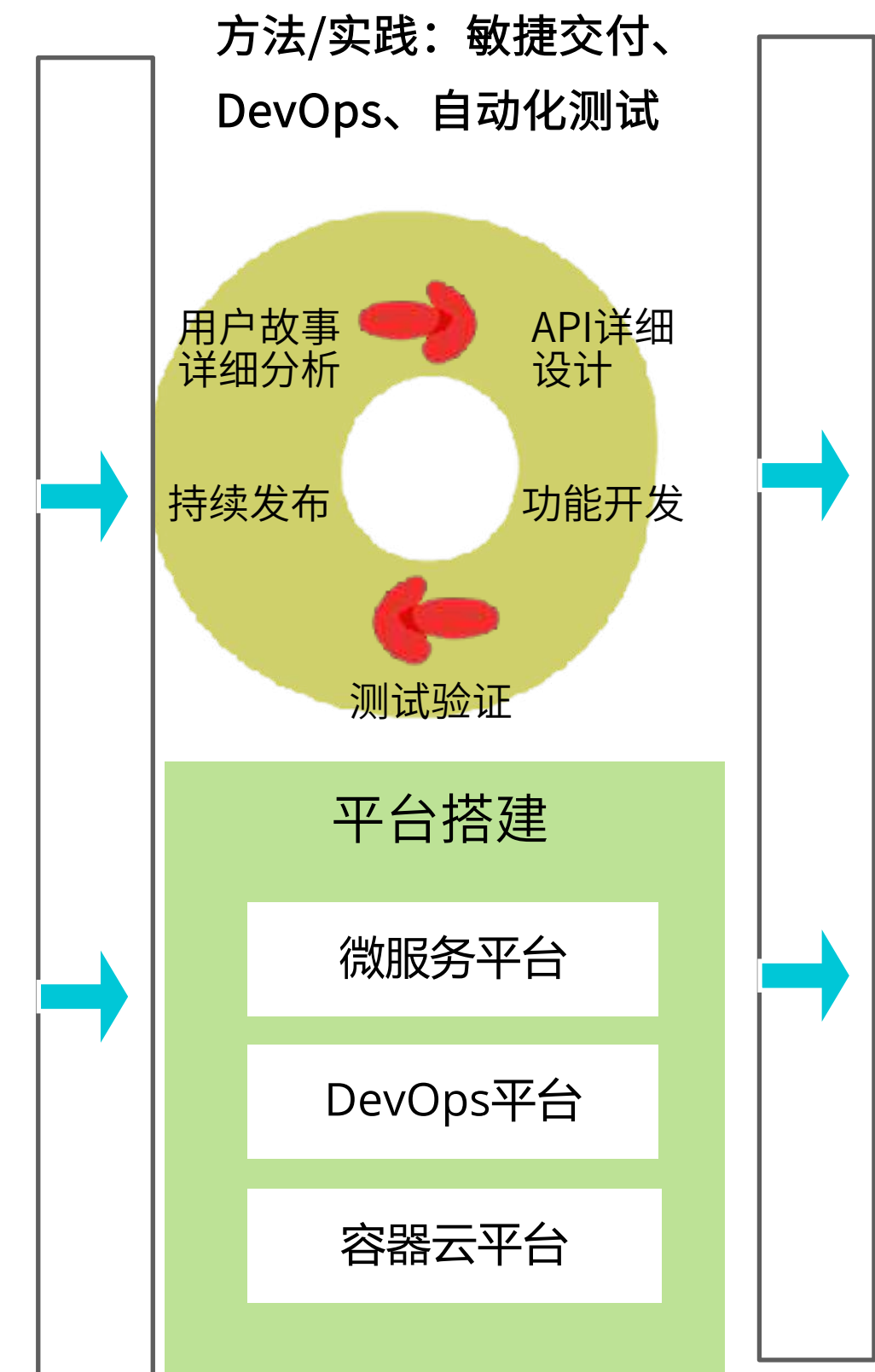
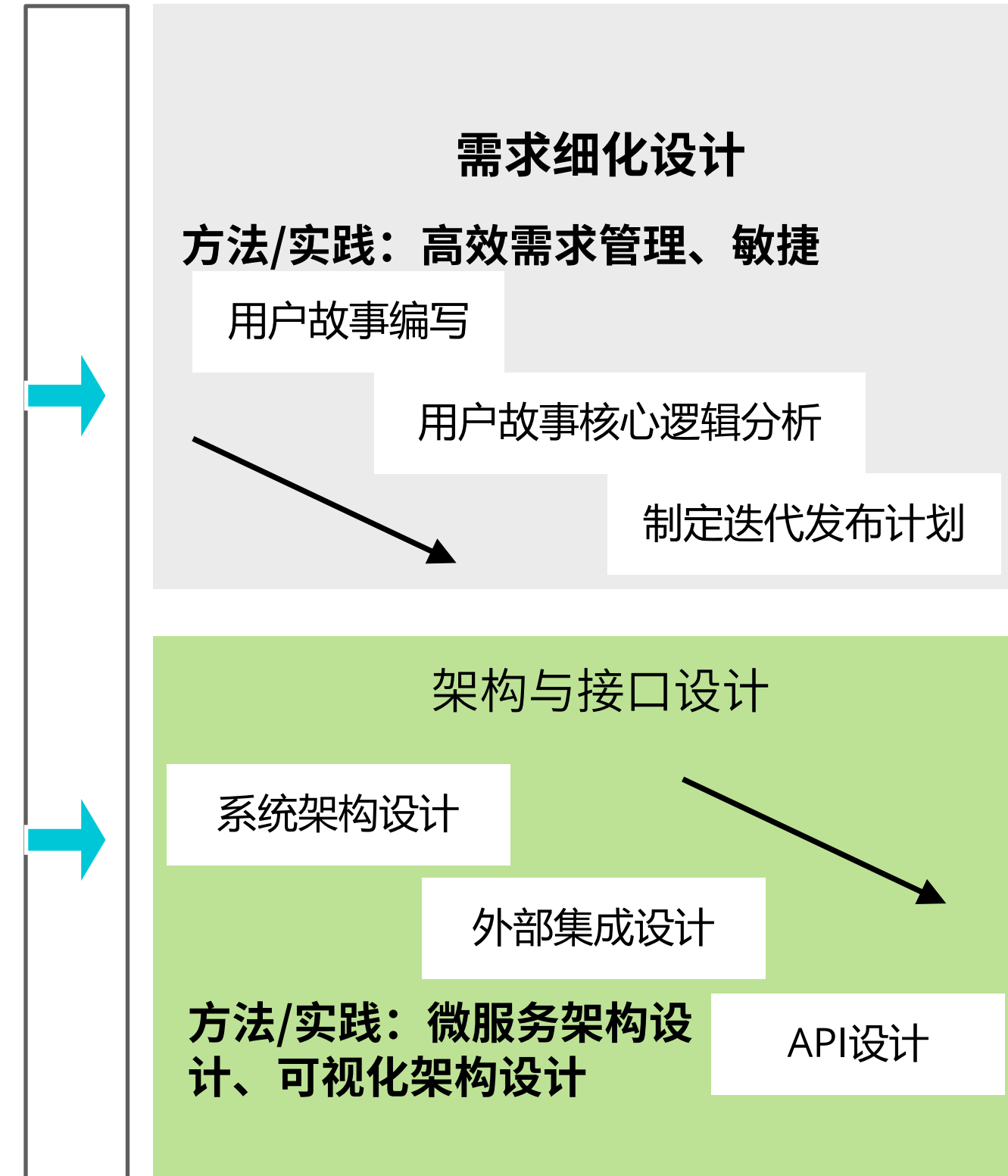
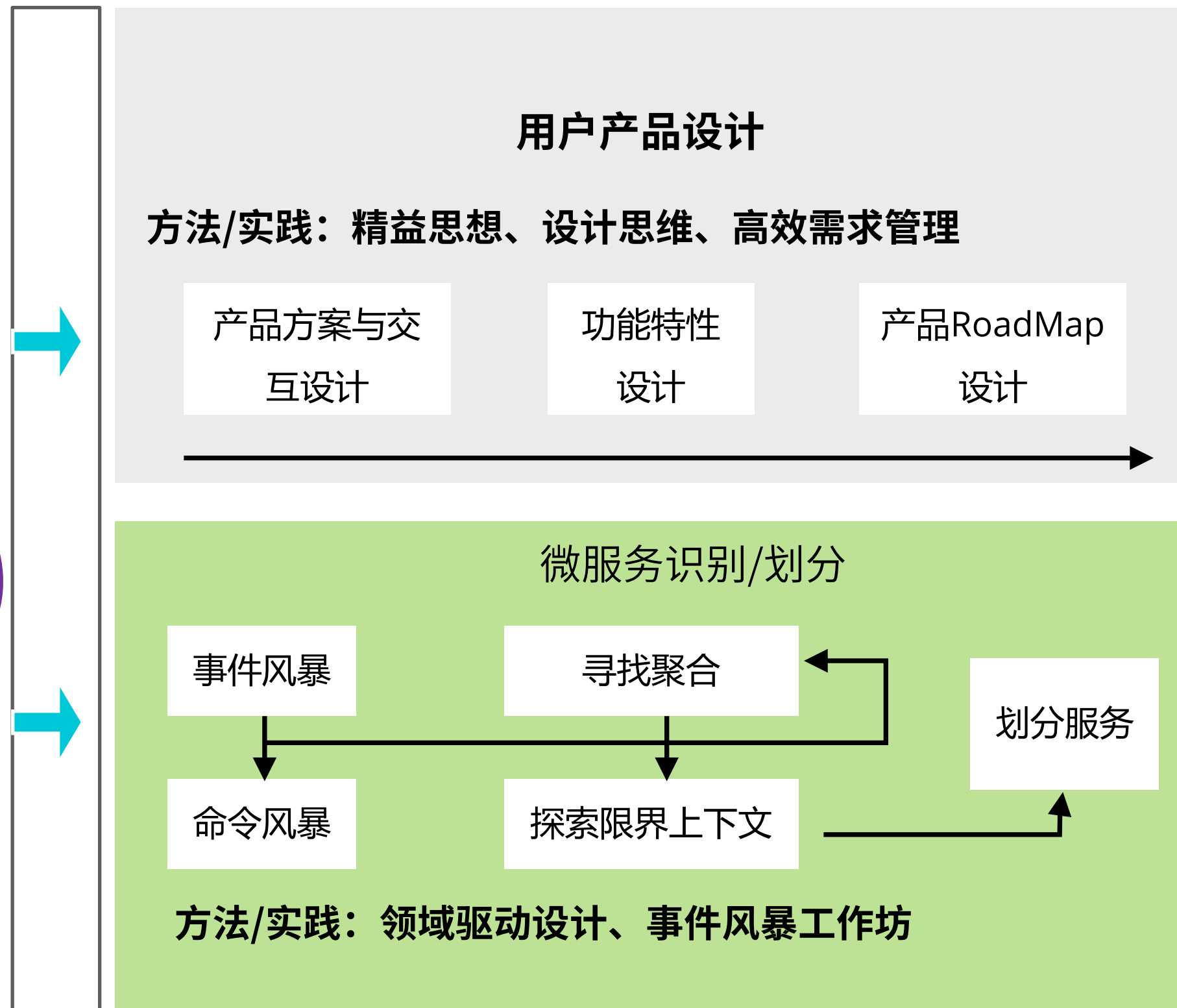
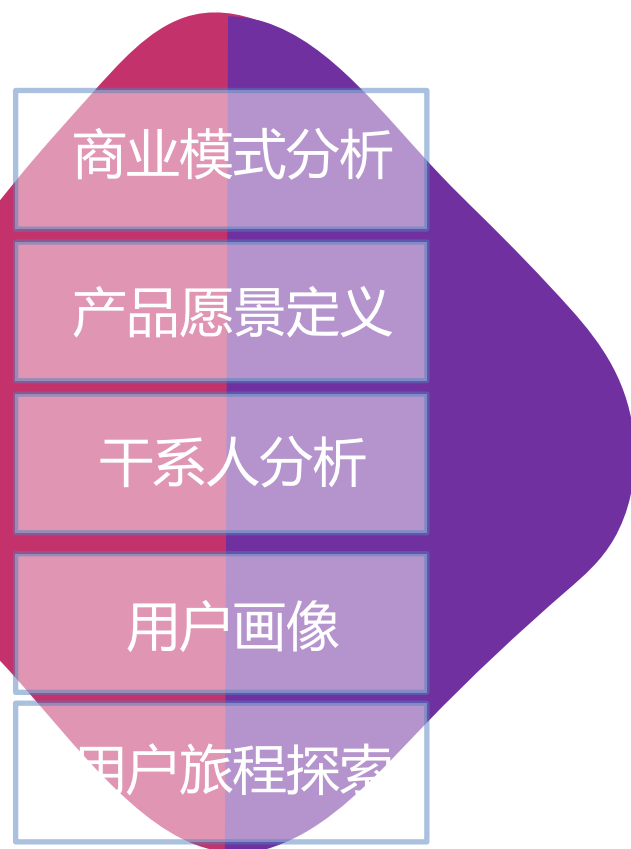
业务方案探索
Solution Discovery

产品+服务定义
Product and Service Define

交付设计
Delivery Design

迭代交付
Iterative Delivery

方法/实践：
精益思想、设计思维、
产品战略规划





THANK YOU

欲了解详情，请发送邮件至：
championliu@tencent.com
zhuli@thoughtworks.com

ThoughtWorks®



ThoughtWorks®

企业技术投资 可以从构建雷达做起

王威

ThoughtWorks

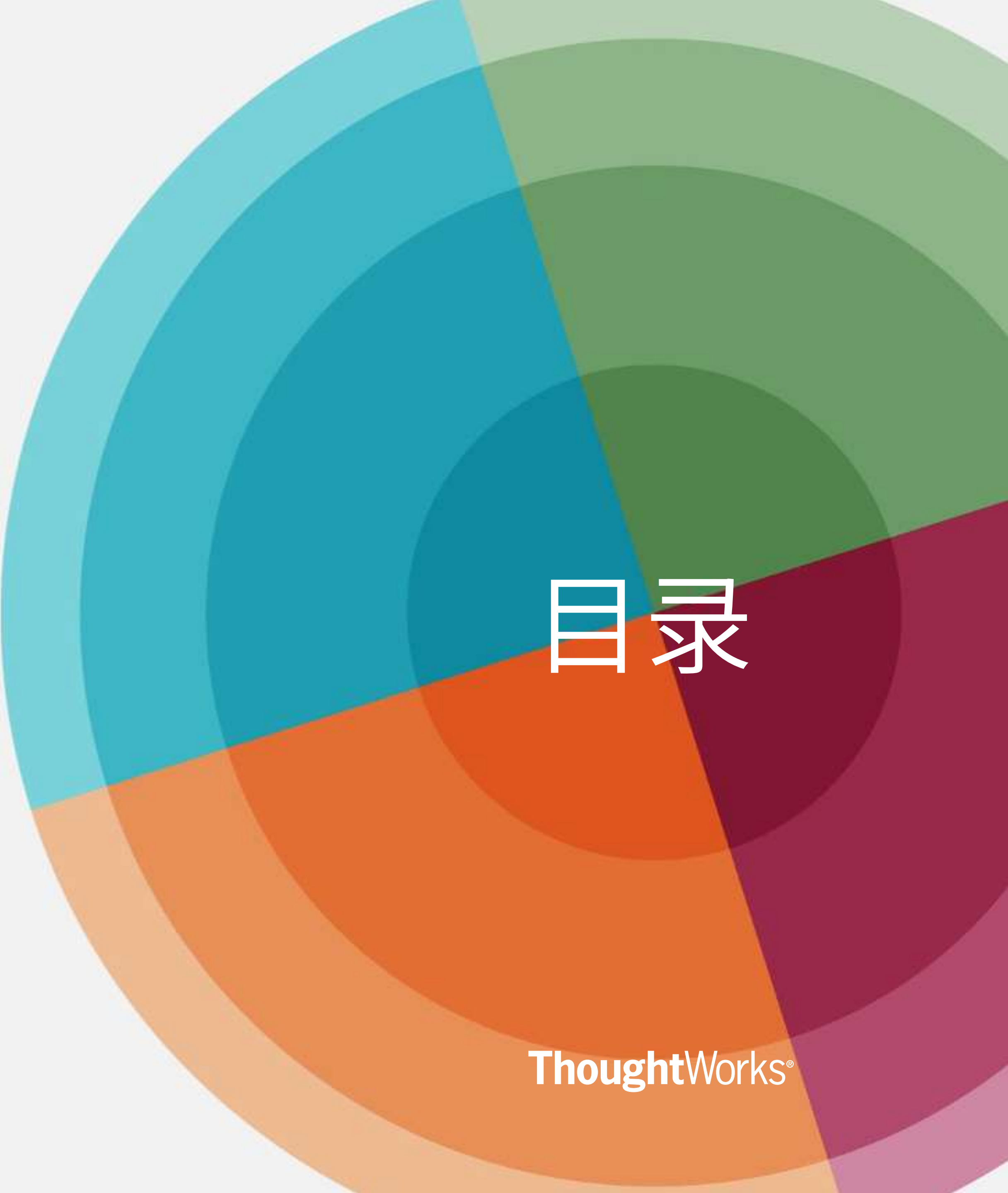
中国区技术战略咨询负责人

杨雨静

ThoughtWorks

高级咨询顾问

- 01** 技术投资的重要性
- 02** 场景+技术，双向驱动的投资策略
- 03** 管理约束，实现场景和技术投资落地
- 04** 总结、Q&A



目录

01

技术投资的重要性

ThoughtWorks®

TECHNOLOGY RADAR VOL. 24

有态度的前沿技术解析



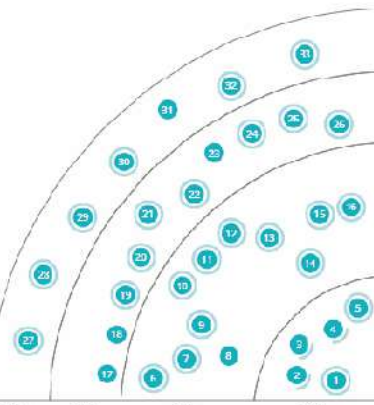
技术

- 采纳**
1. API 扩张 - 收缩
 2. 机器学习的持续交付 (CD4ML)
 3. 设计体系
 4. 平台工程产品团队
 5. 服务帐户轮换方法

- 试验**
6. 云沙箱
 7. Contextual bandits
 8. Distroless Docker images
 9. Ethical Explorer
 10. 假设驱动的遗留系统改造
 11. 轻量级的 RFCs 方法
 12. 最简的机器学习
 13. 单页应用 (SPA) 注入
 14. 团队的认知负担
 15. 使用工具管理 Xcodeproj
 16. UI/BFF 共享类型

- 评估**
17. 限界低代码平台
 18. 去中心化身份
 19. 部署漂移提示器
 20. 同态加密
 21. Hotwire
 22. 微前端中的模块映射
 23. 开放应用程序模型
 24. 关注隐私的网络分析

- 暂缓**
25. Remote mob programming
 26. 安全多方计算



工具

- 采纳**
52. Sentry

- 试验**
53. axe-core
 54. dbt
 55. esbuild
 56. Flipper
 57. Great Expectations
 58. k6
 59. MLflow
 60. OR-Tools
 61. Playwright

- 评估**
62. Prowler
 63. Pyright
 64. Redash
 65. Terratest
 66. Tuple
 67. Why Did You Render

- 暂缓**
80. AWS CodePipeline

- 试验**
83. Angular Testing Library
 84. AWS Data Wrangler
 85. Elazor
 86. FastAPI
 87. io-ts
 88. Kotlin Flow
 89. LitElement
 90. Next.js
 91. 按需分发模块
 92. Streamlit
 93. SWR
 94. TrustKit

- 评估**
95. .NET 5
 96. bUnit
 97. Dagster
 98. Flutter for Web
 99. Jotai 和 Zustand
 100. Kotlin Multiplatform Mobile
 101. LVGL
 102. React Hook Form
 103. River
 104. Webpack 5 模块联邦

暂缓

平台

- 采纳**
34. AWS 云开发工具包
 35. Backstage
 36. Delta Lake
 37. Materialize
 38. Snowflake
 39. 可变字体

- 评估**
40. Apache Pinot
 41. Bit.dev
 42. DataHub
 43. Feature Store
 44. JuiceFS
 45. 用 Kafka API 而非 Kafka

- 暂缓**
46. NATS
 47. Opstrace
 48. Pulumi
 49. Redpanda

- 试验**
50. Azure Machine Learning
 51. 自研基础设施即代码 (IaC) 产品

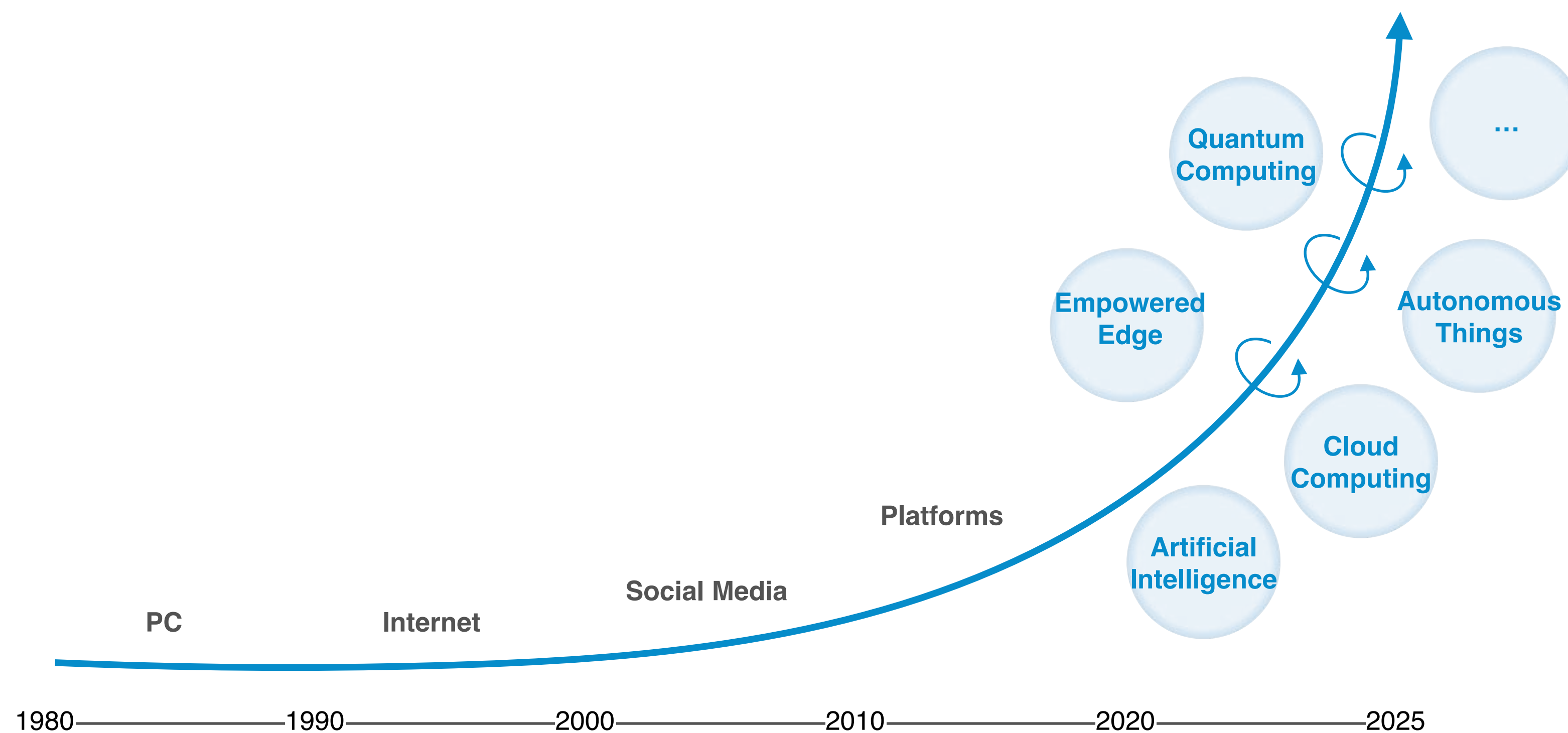
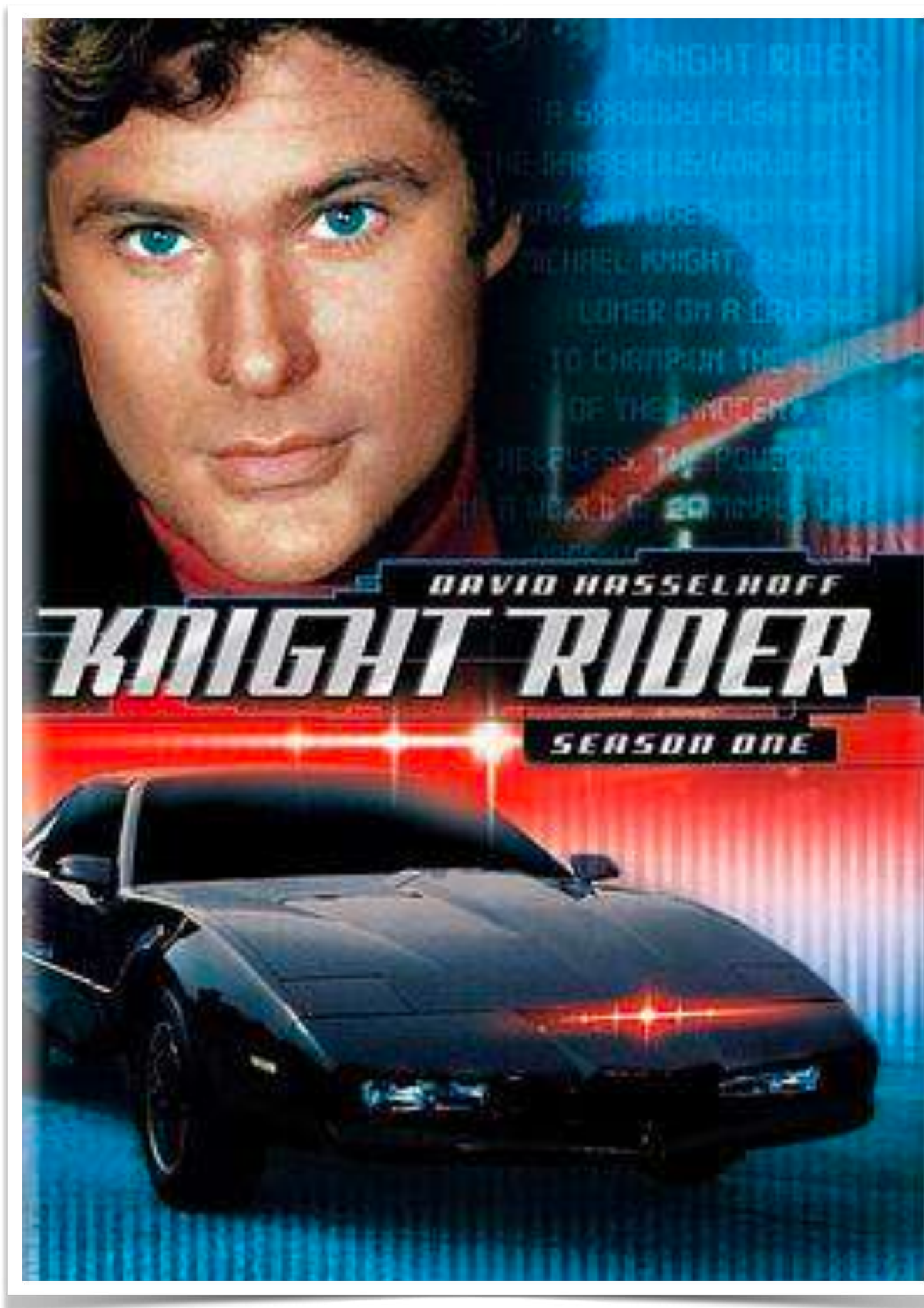
语言&框架

- 采纳**
81. Combine
 82. LeakCanary

“五年后市场上最流行的语言和框架是什么？”

技术发展的速度呈几何级数增长

ThoughtWorks®

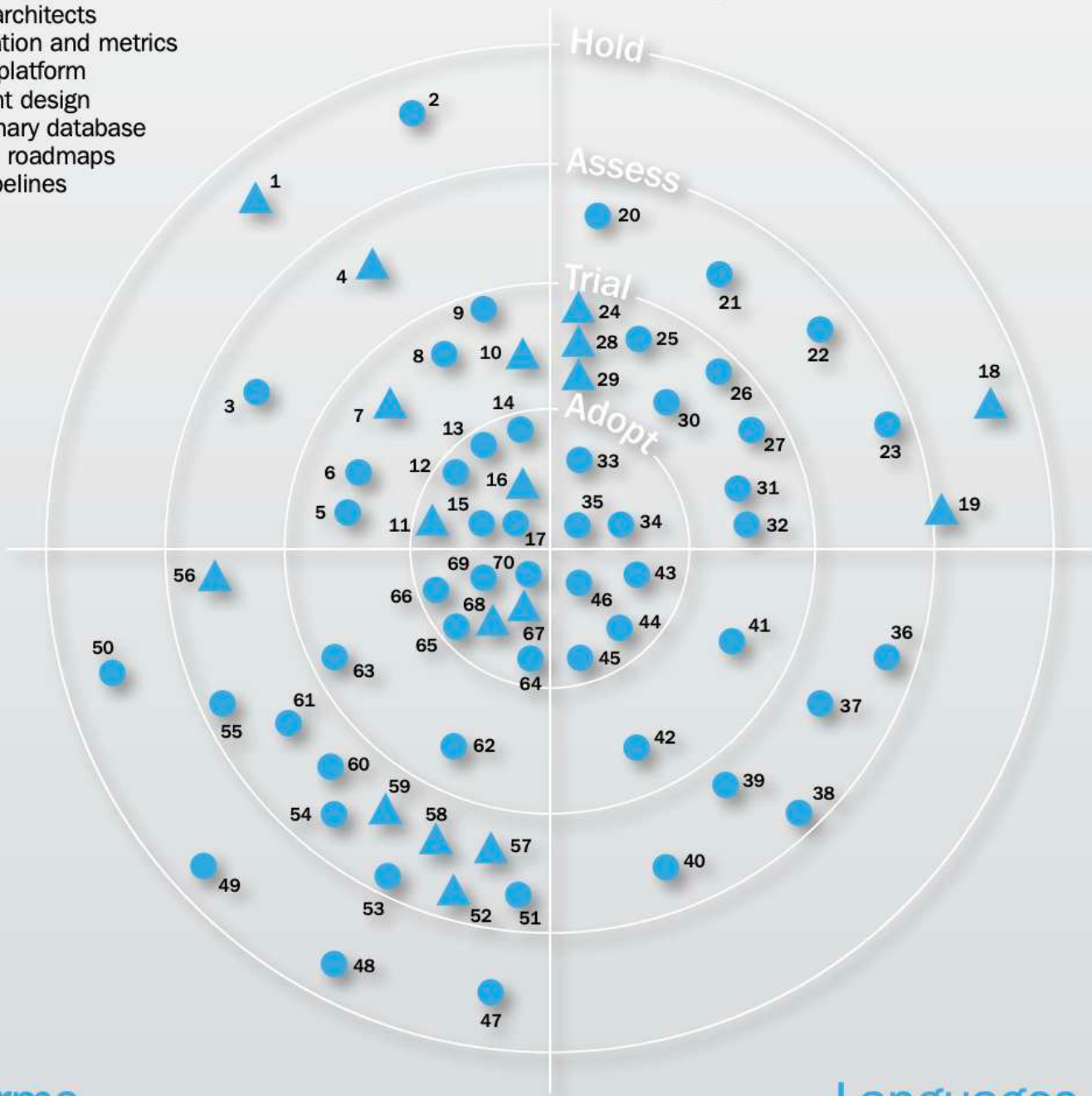


Techniques

- 1. Database based integration
- 2. Scrum certification
- 3. Incremental data warehousing
- 4. DevOps
- 5. Polyglot programming
- 6. Automation of technical tests
- 7. Capability modeling
- 8. Service choreography
- 9. Continuous deployment
- 10. Evolutionary architecture
- 11. Coding architects
- 12. Visualization and metrics
- 13. Web as platform
- 14. Emergent design
- 15. Evolutionary database
- 16. Platform roadmaps
- 17. Build pipelines

Tools

- 18. ESB
- 19. Intentional
- 20. Cross mobile platforms
- 21. Github
- 22. Restfulie
- 23. RDF triple stores
- 24. Apache camel
- 25. Next gen test tools
- 26. NoSQL
- 27. Neo4j
- 28. Message buses without smarts
- 29. Puppet
- 30. mongoDB
- 31. Mercurial
- 32. Git
- 33. Squid
- 34. ASP.NET MVC
- 35. Subversion



Platforms

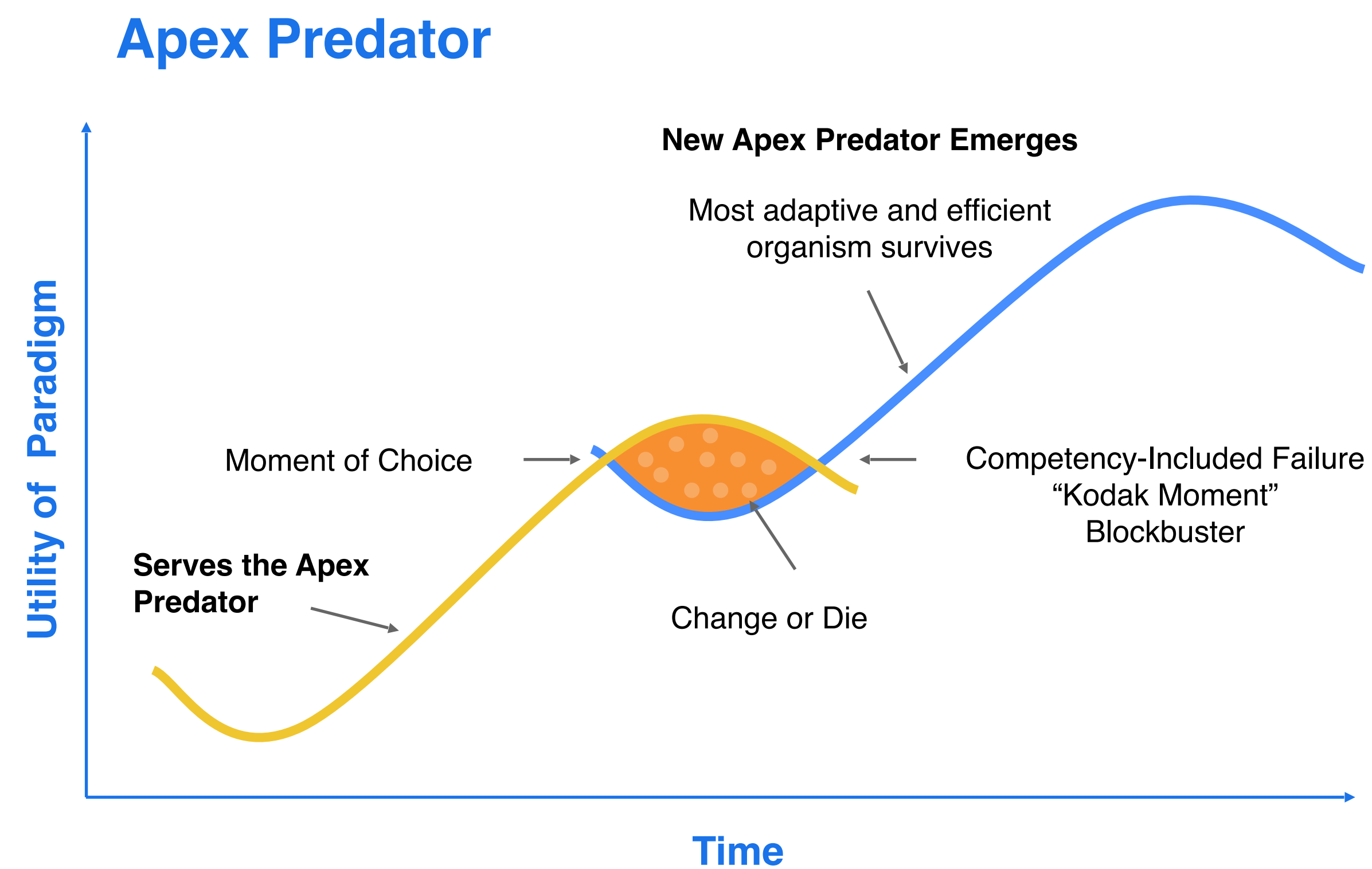
- 47. Rich internet applications
- 48. GWT
- 49. IE8
- 50. WS-* beyond basic profile
- 51. Azure
- 52. Mobile Web
- 53. Google App Engine
- 54. Application appliances
- 55. Google as corporate platform
- 56. GPGPU
- 57. App containers
- 58. OAuth
- 59. RDFa

- 60. Location based services
- 61. iPad
- 62. EC2 & S3
- 63. Facebook as business platform
- 64. JVM as platform
- 65. iPhone
- 66. Android
- 67. KVM
- 68. Atom
- 69. ALT.NET
- 70. IE6 End of Life

Languages

- 36. Java language end of life
- 37. F#
- 38. Scala
- 39. Clojure
- 40. HTML 5
- 41. DSL's
- 42. Groovy
- 43. C# 4.0
- 44. JRuby
- 45. JavaScript as a first class language
- 46. Ruby

4. DevOps



Concepts adapted from Bangor University and/or Cognitive Edge Pie Ltd.



2021年1月



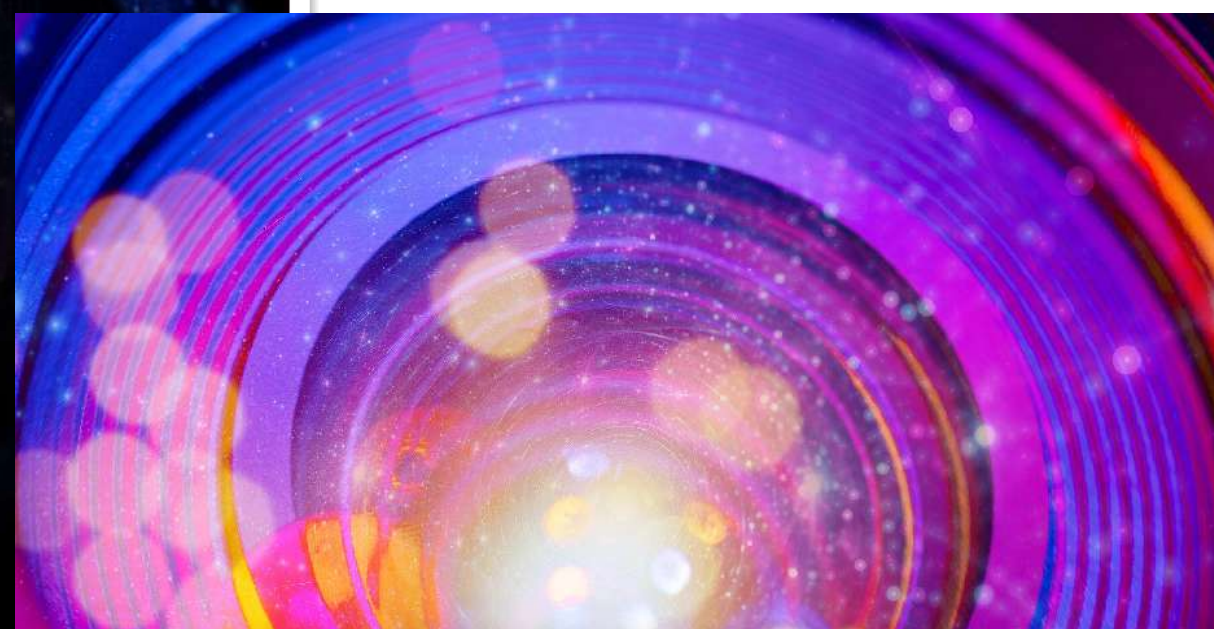
#1
增强人类



#2
加速迈向可持续发展



#3
不断演进的交互体验



#4
计算结构的变革



#5
竞合促使平台走向生态系统



#6
技术作恶的影响不断扩大

	正在发生	开始出现	即将出现
增强人类	自然语言处理	可解释的AI	渐进式神经网络
加速迈向可持续发展	分布式能源	数字化碳管理	暗数据中心
不断演进的交互方式	人机协作	无接触交互	光场显示技术
计算结构的变革	多云	量子计算	DNA数据存储
平台构筑生态	开放API平台	雾计算	协作生态
技术作恶的影响不断扩大	生物识别技术	差异化隐私	密码的消失

02

场景+技术，双向驱动的投资策略

ThoughtWorks®



技术有了，那我应用到哪里呢

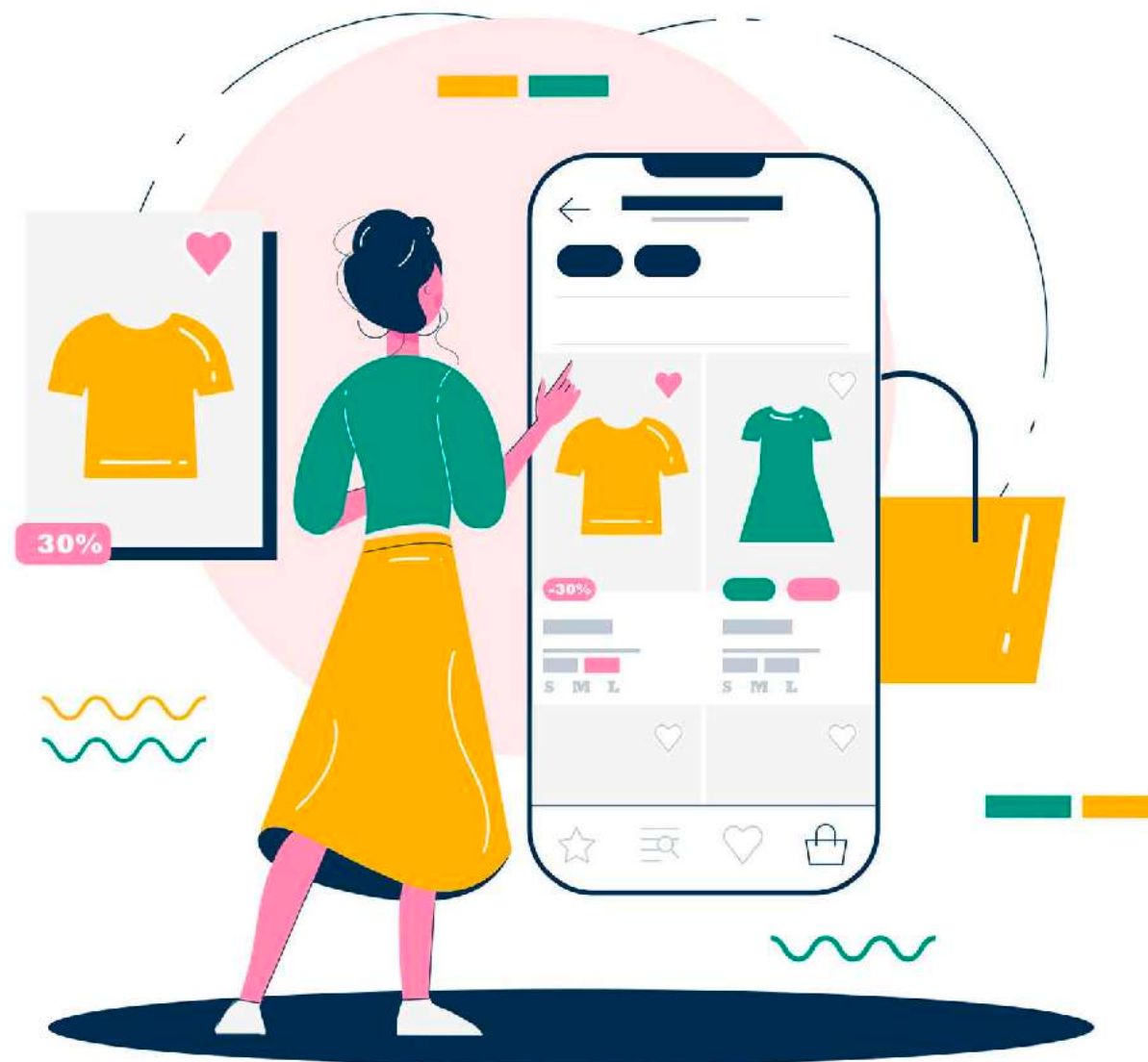
“如果技术是解决问题的答案，那问题是什么呢？”

不谈及问题的解决方案是“中性”的

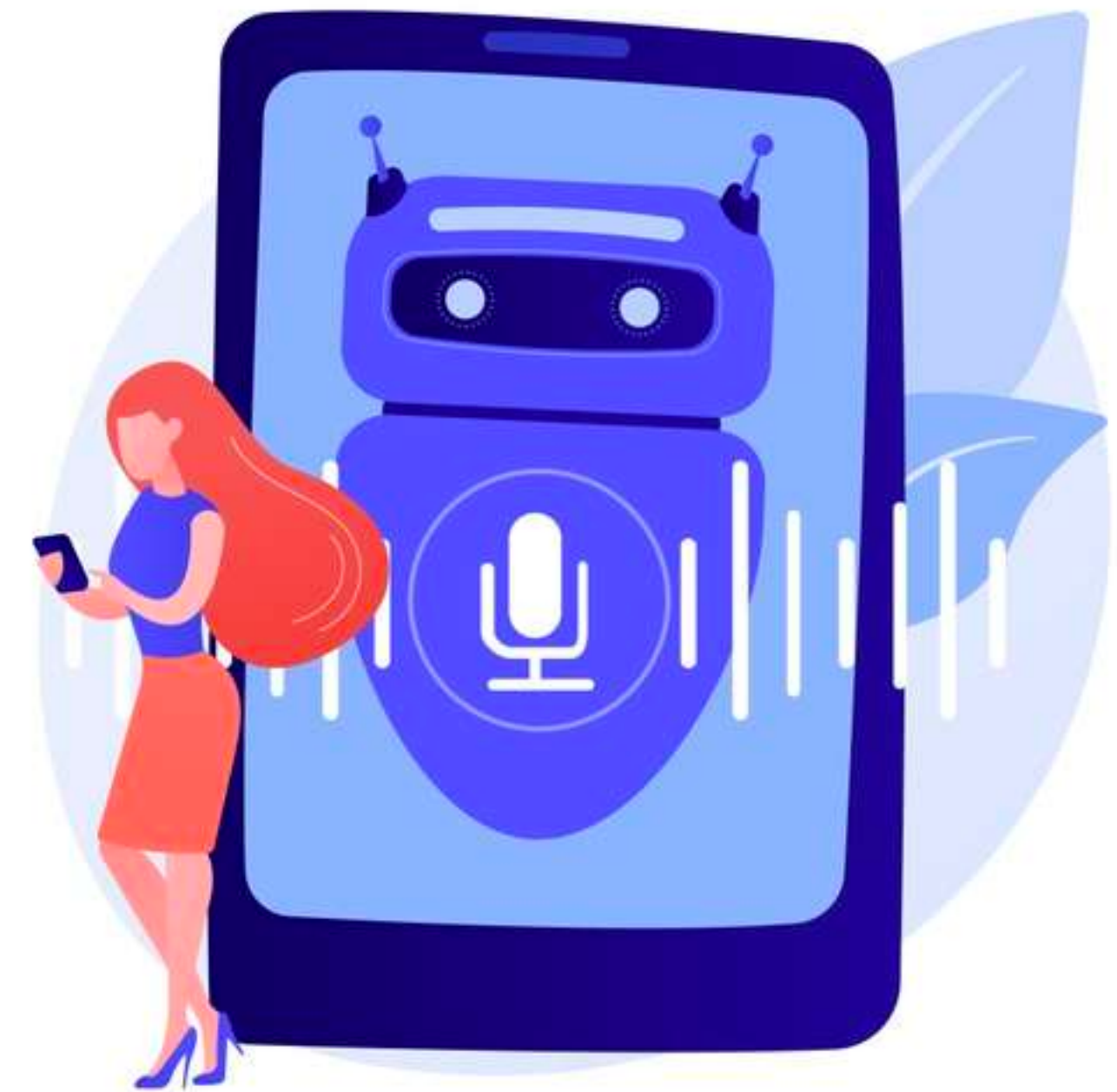
ThoughtWorks®



自动驾驶的人工智能



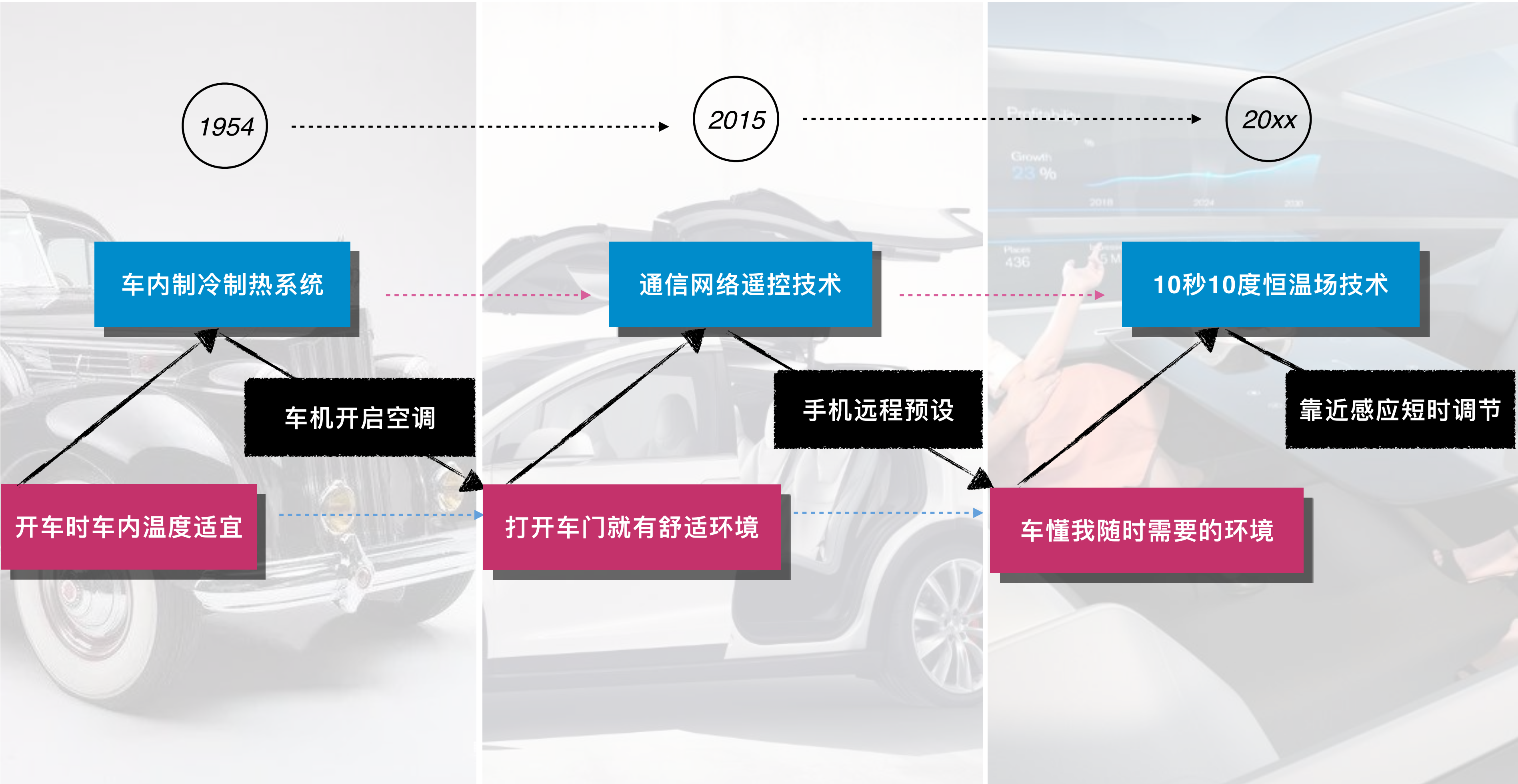
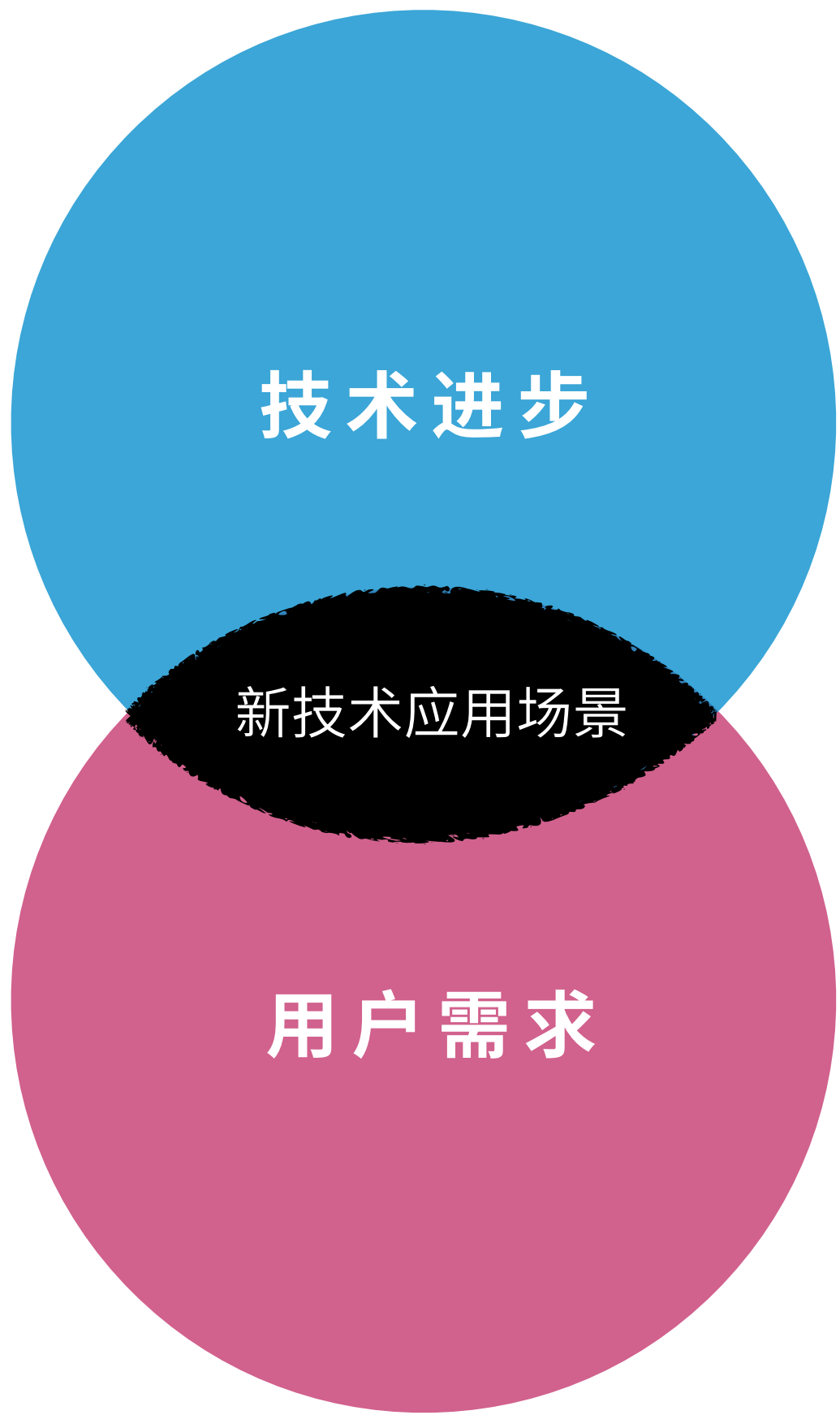
线上购物的人工智能



智慧语音的人工智能

技术的进步与用户需求本身，是协同演进的

ThoughtWorks®



关注解决方案的同时，关注问题本身。

关于用户需求的“谎言”

ThoughtWorks®



关于用户需求的“谎言”

ThoughtWorks®

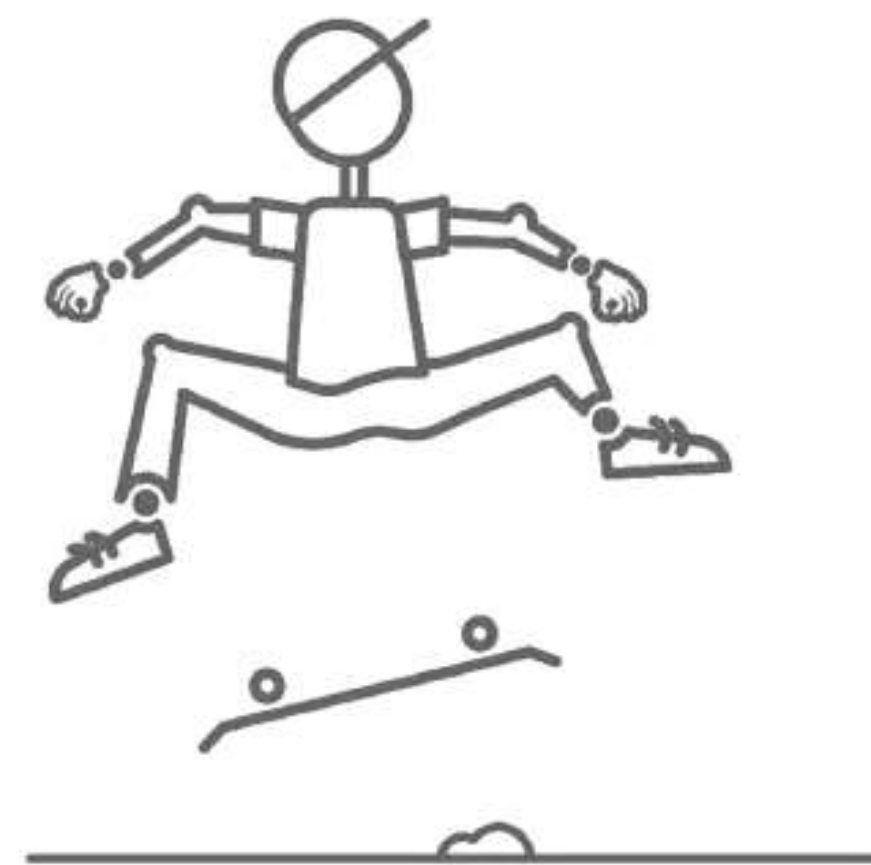
“你要问我想要什么的话，
那我想要一匹更快的马。”



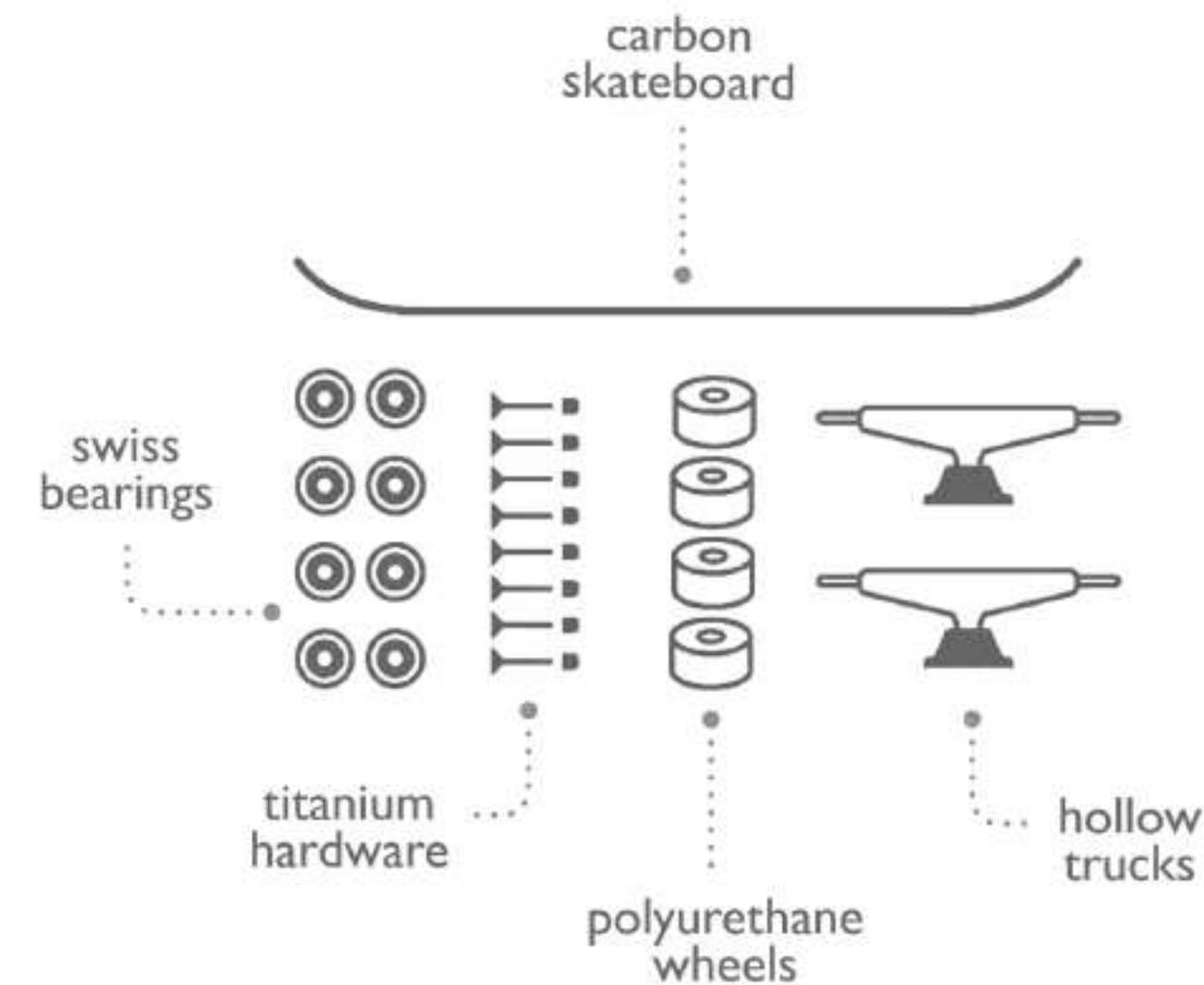
关注用户场景，而不只是用户说的话

人们购买/雇佣一个产品，去完成他/她的“Jobs”

This is most important to customers...



...not this.



Jobs-to-be-Done Needs Framework (Clayton Christensen)

WHO
谁

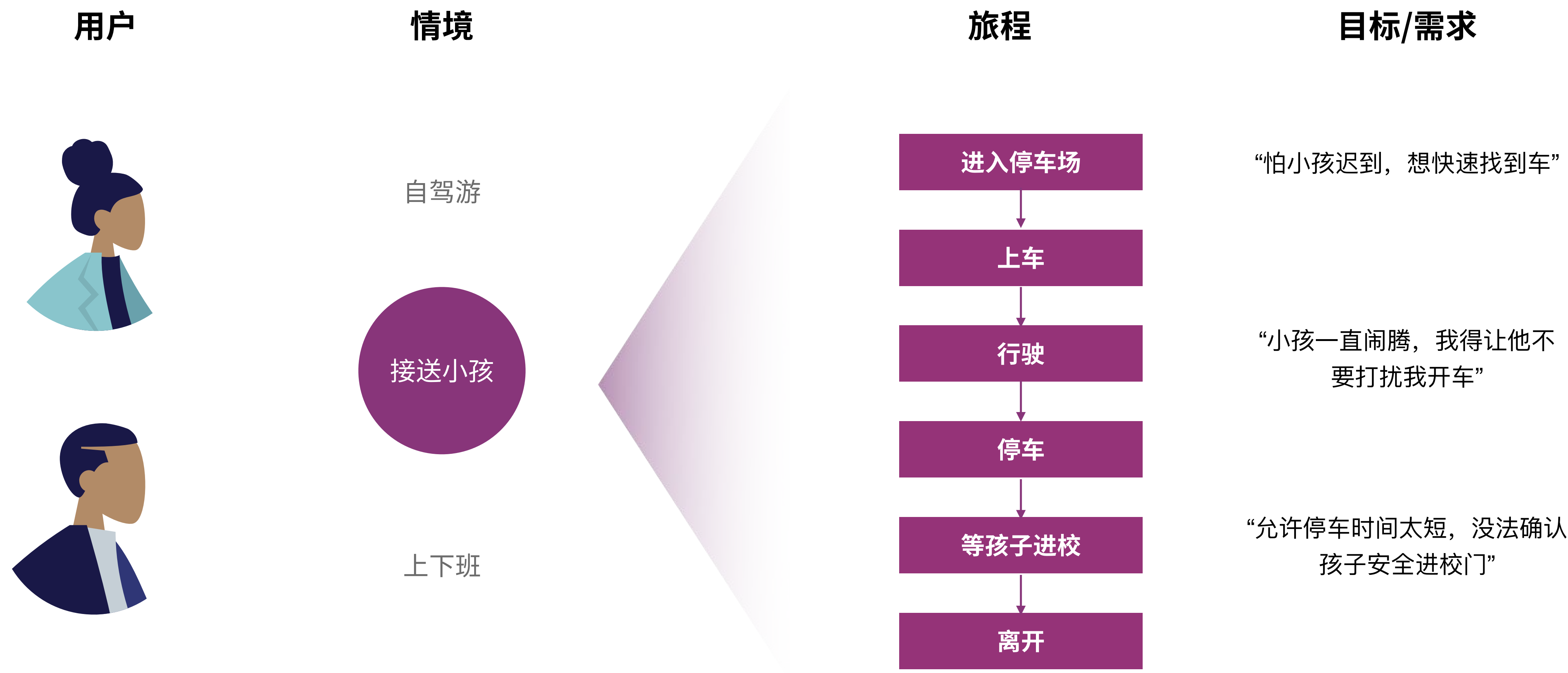
WHEN
什么时候

WHERE
在哪

WHAT
遇到什么情况

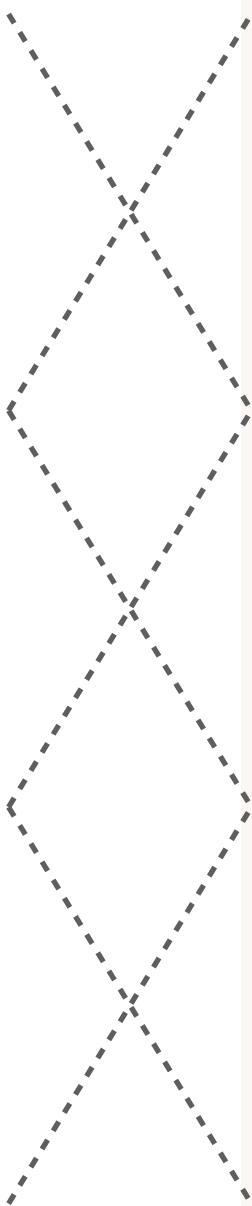
HOW
周围环境如何





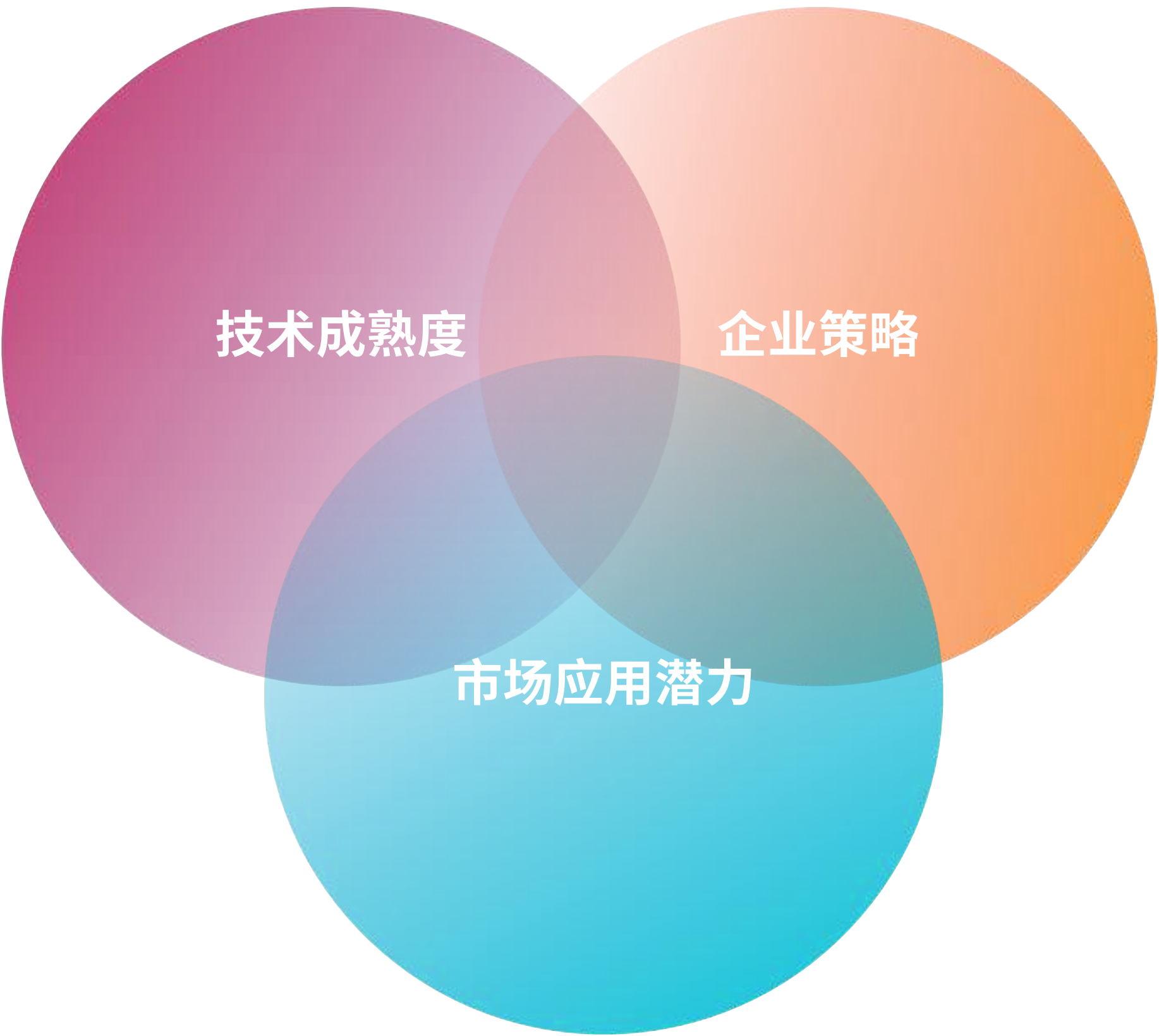
技术趋势池

	正在发生	开始出现	即将出现
增强人类	自然语言处理	可解释的AI	渐进式神经网络
加速迈向可持续发展	分布式能源	数字化碳管理	暗数据中心
不断演进的交互方式	人机协作	无接触交互	光场显示技术
计算结构的变革	多云	量子计算	DNA数据存储
平台构筑生态	开放API平台	雾计算	协作生态
技术作恶的影响不断扩大	生物识别技术	差异化隐私	密码的消失



场景需求池

			
忘记提前开空调	停车场找不到车	晚高峰交通拥堵	通勤
小孩吵闹	太堵，耽误小孩写作业		接送小孩
	路途太长，驾驶疲劳	封路无法提前知道	自驾游
高速路上大货车多，危险	机场没有停车位		出差
周末路上行人太多		周末不知道去哪	市内出行



技术成熟度

- **商业成熟度**
 - 该技术在市场上是否有足够的供应商或者合作伙伴？
- **技术成熟度**
 - 该技术是否尚未形成技术壁垒和专利壁垒
- **监管合规成熟度**
 - 监管机构对该技术持有开放性的态度？

企业策略

- **企业 /部门发展战略与目标**
 - 该技术是否能够带来符合企业发展方向的机会？
- **核心业务领域**
 - 该技术在企业哪些核心业务领域发挥作用？
- **技术应用现状**
 - 该技术在企业当前的应用情况何如？

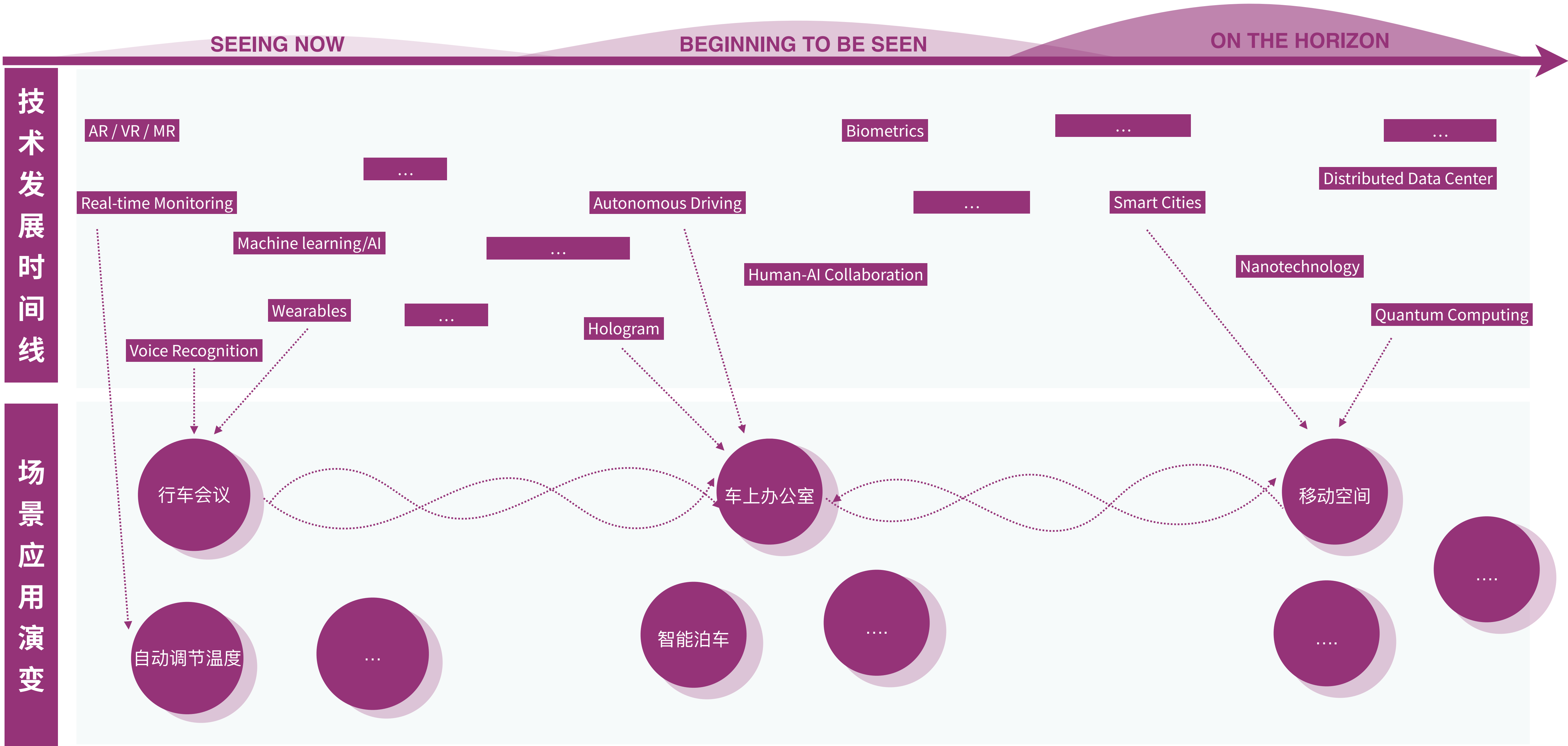
市场应用潜力

- **市场定位**
 - 该技术带来的创新机会时都能够帮助企业加强市场定位或者创造新的市场？
- **用户价值**
 - 该技术带来的创新机会带来多大的用户价值？
- **创新程度**
 - 该技术带来的创新机会与现有解决方案相比，有多创新？

技术趋势	技术能力	场景应用	场景需求
Immersive Experience	Wearables Voice Recognition	行车会议	“作为一个商业人士，我在下班回家过程中，想要继续完成我的会议...”
	Weather Monitoring	自动调节温度	“作为一个母亲，我在接送孩子前，想要将车内温度调至适合孩子...”
	Machine Learning Real-time Monitoring	疲劳驾驶提醒	“作为一个DiDi司机，我在行车过程中，想要及时识别疲劳驾驶状态...”
Fully Automatic Technology	Autonomous Driving Autonomous Parking IOT	智能泊车 车上办公室 移动空间	
....	...	...	...
	...	...	...
	...	...	...

技术投资蓝图

ThoughtWorks®

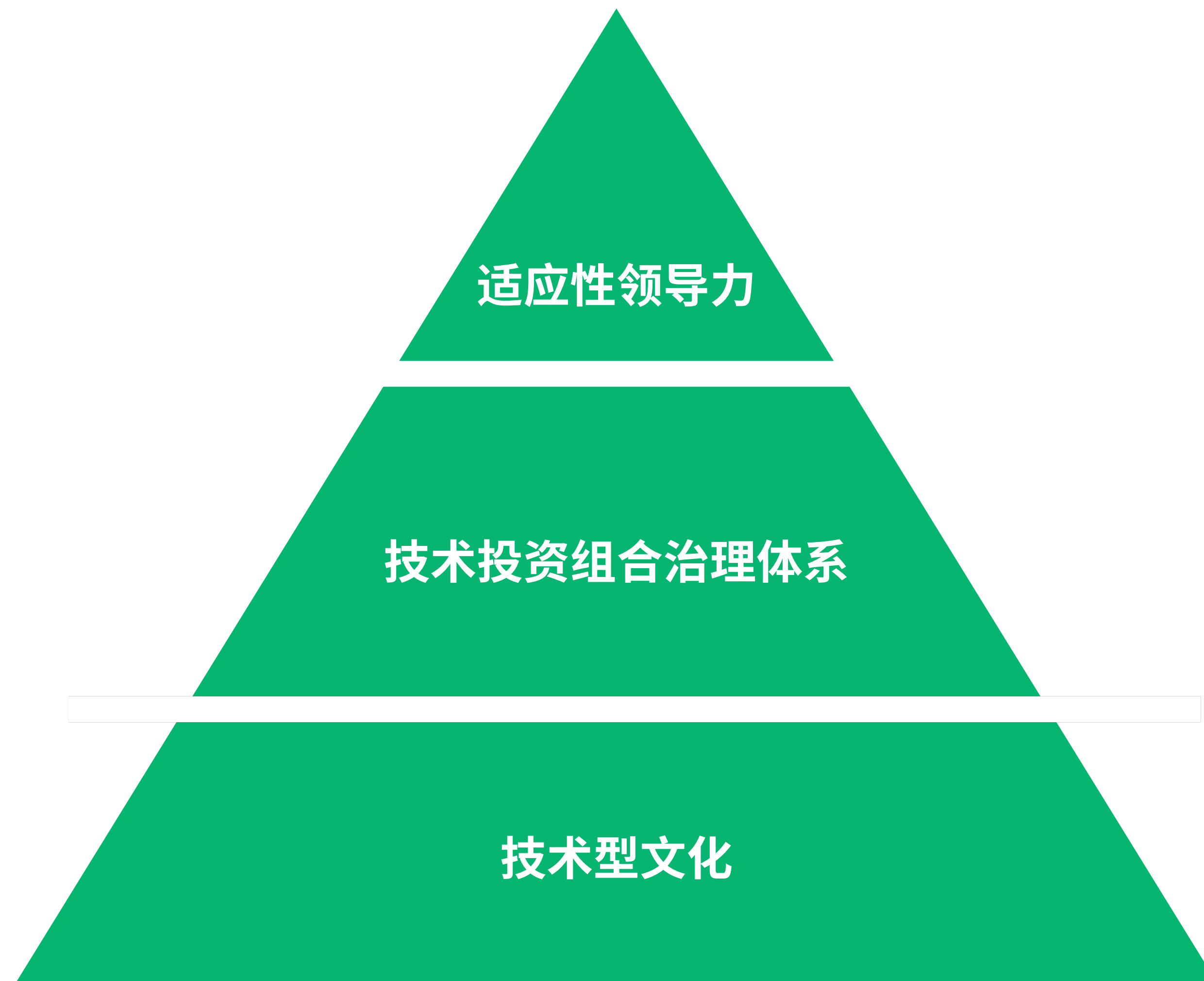


03

管理约束，实现场景和技术投资落地



ThoughtWorks®

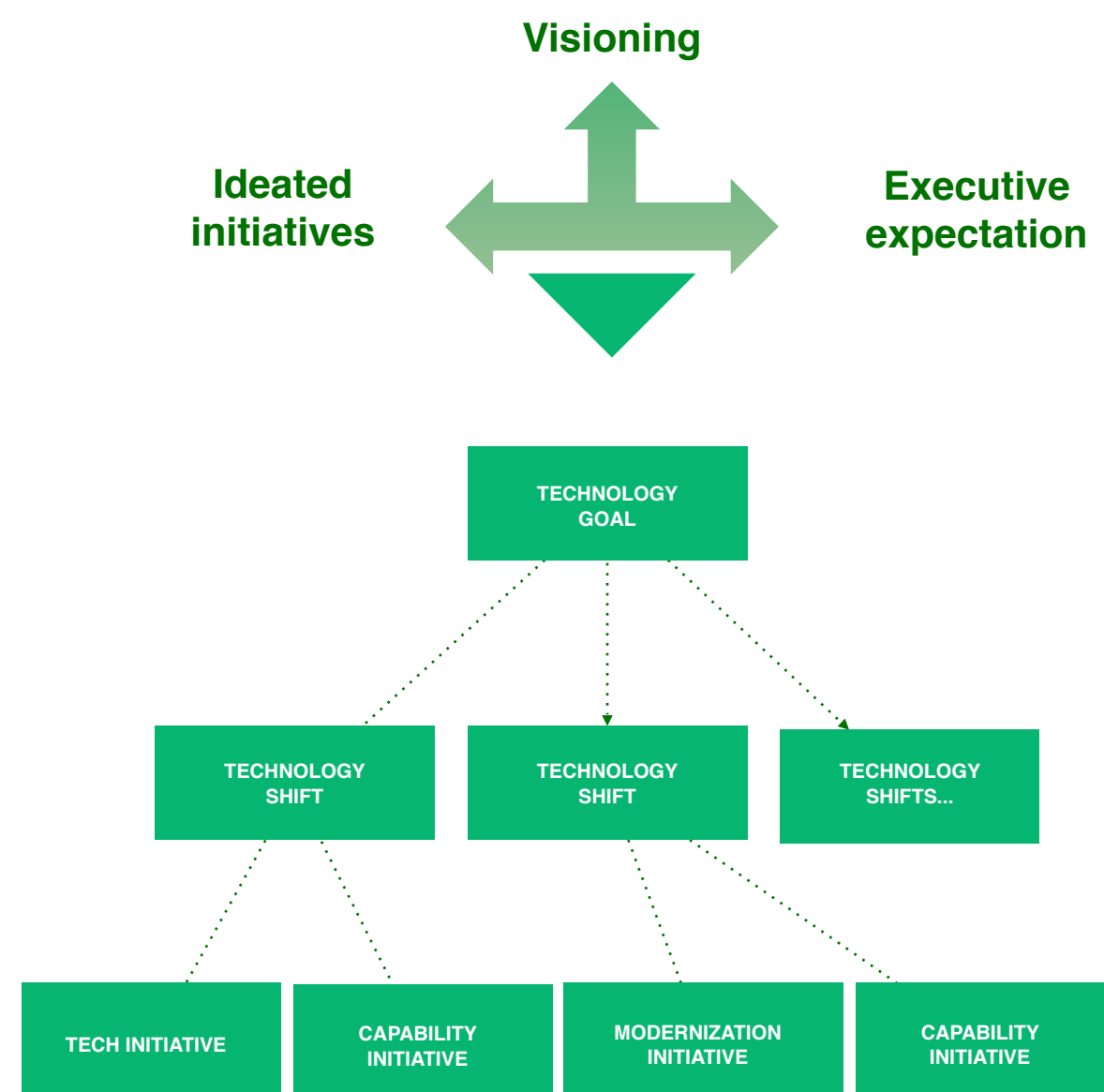




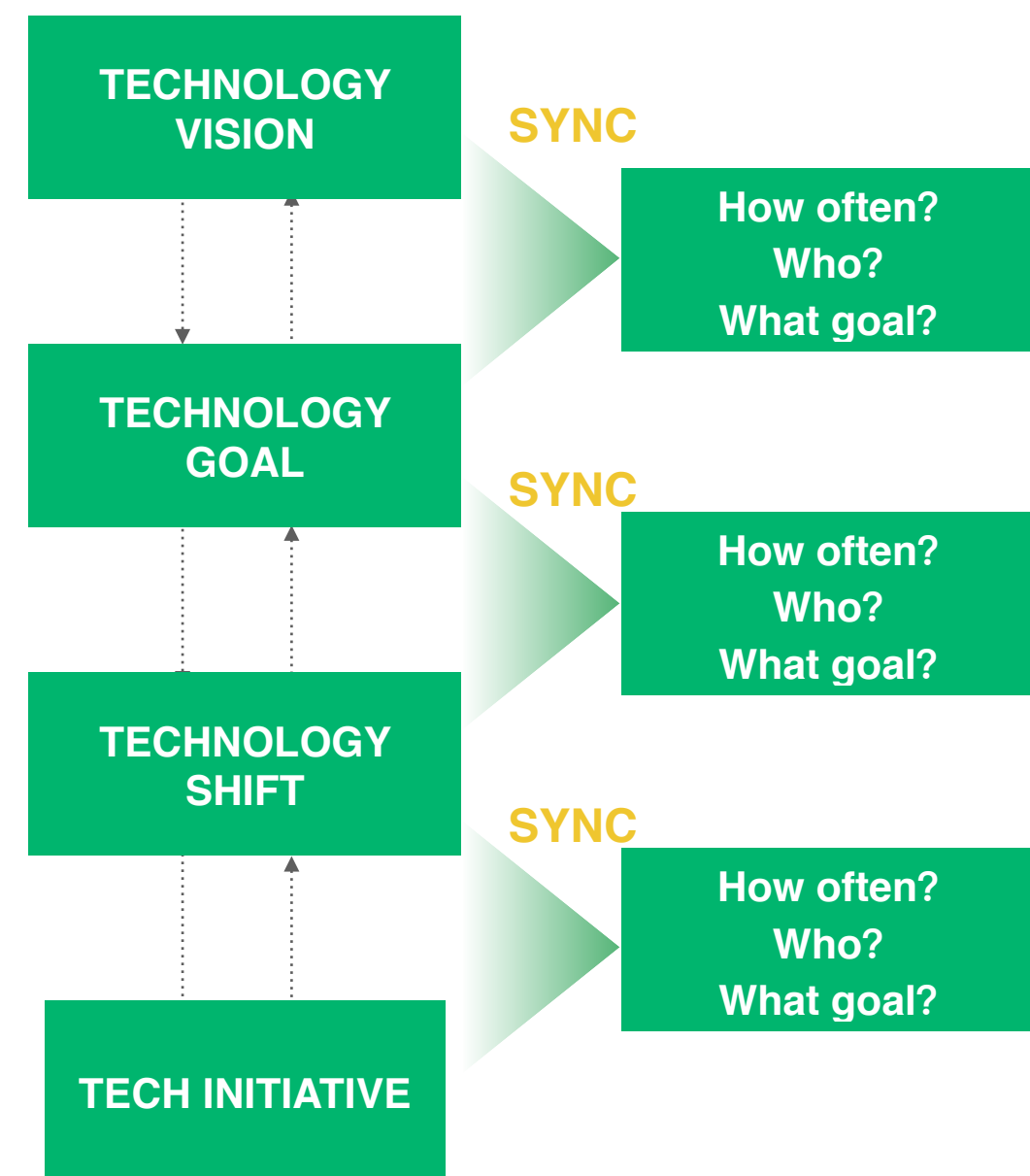
技术投资治理体系

ThoughtWorks®

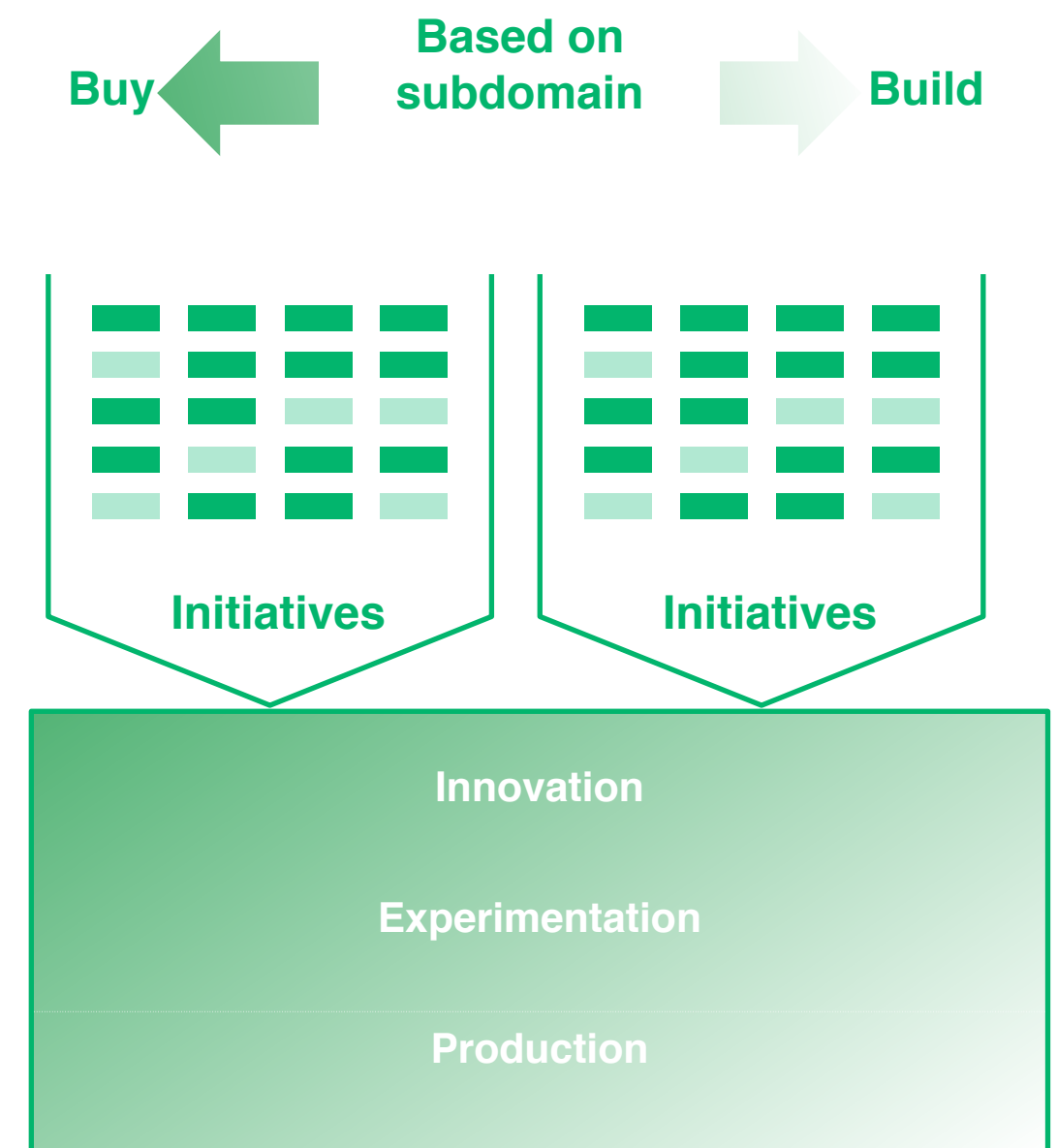
结合技术愿景，技术举措，与
企业期望，得到技术投资组合



根据业务领域划分决定投资计划，衡
量并形成每类技术举措的投资比例



根据业务领域划分决定投资计划，衡
量并形成每类技术举措的投资比例



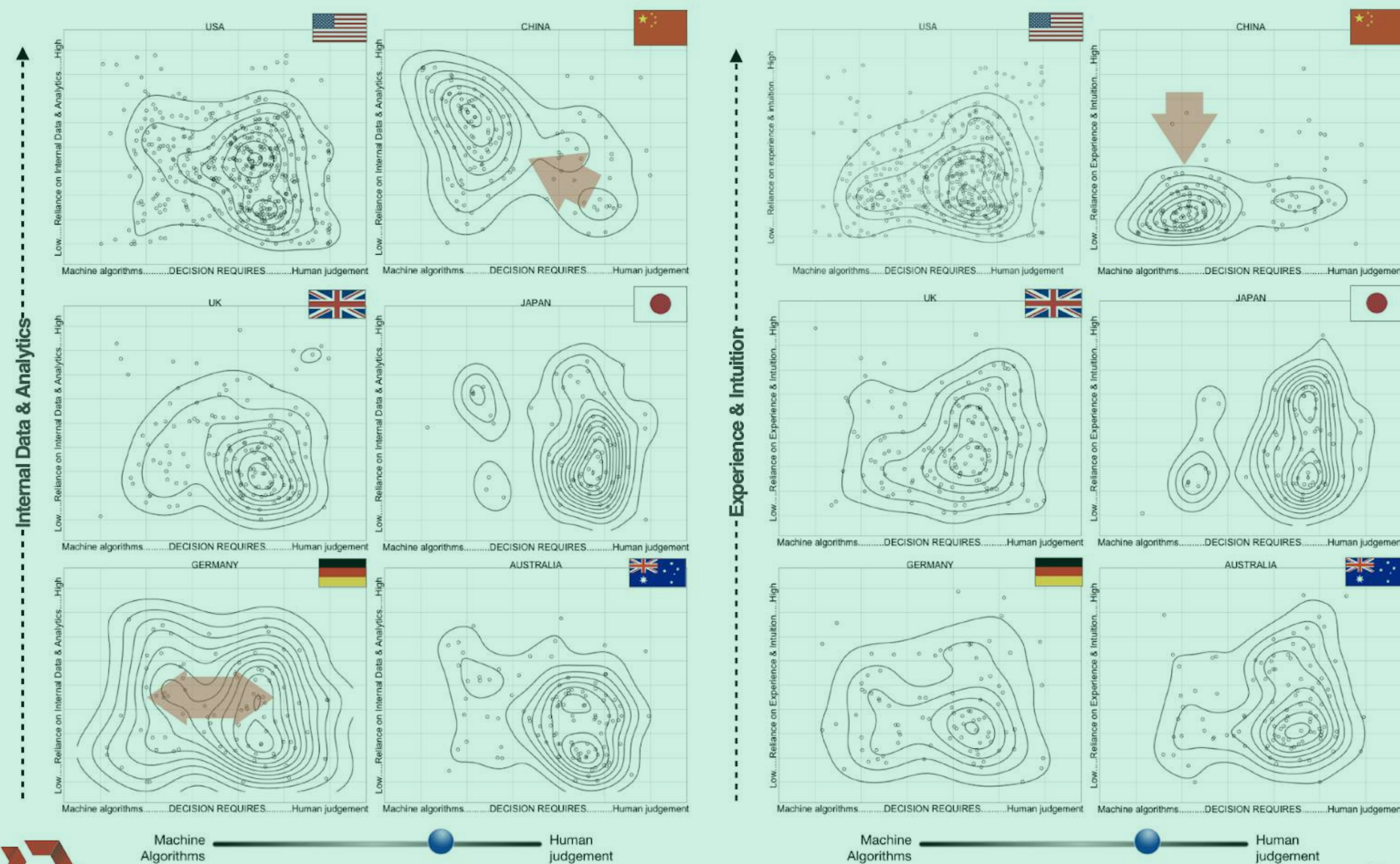
ThoughtWorks®



技术型文化构建

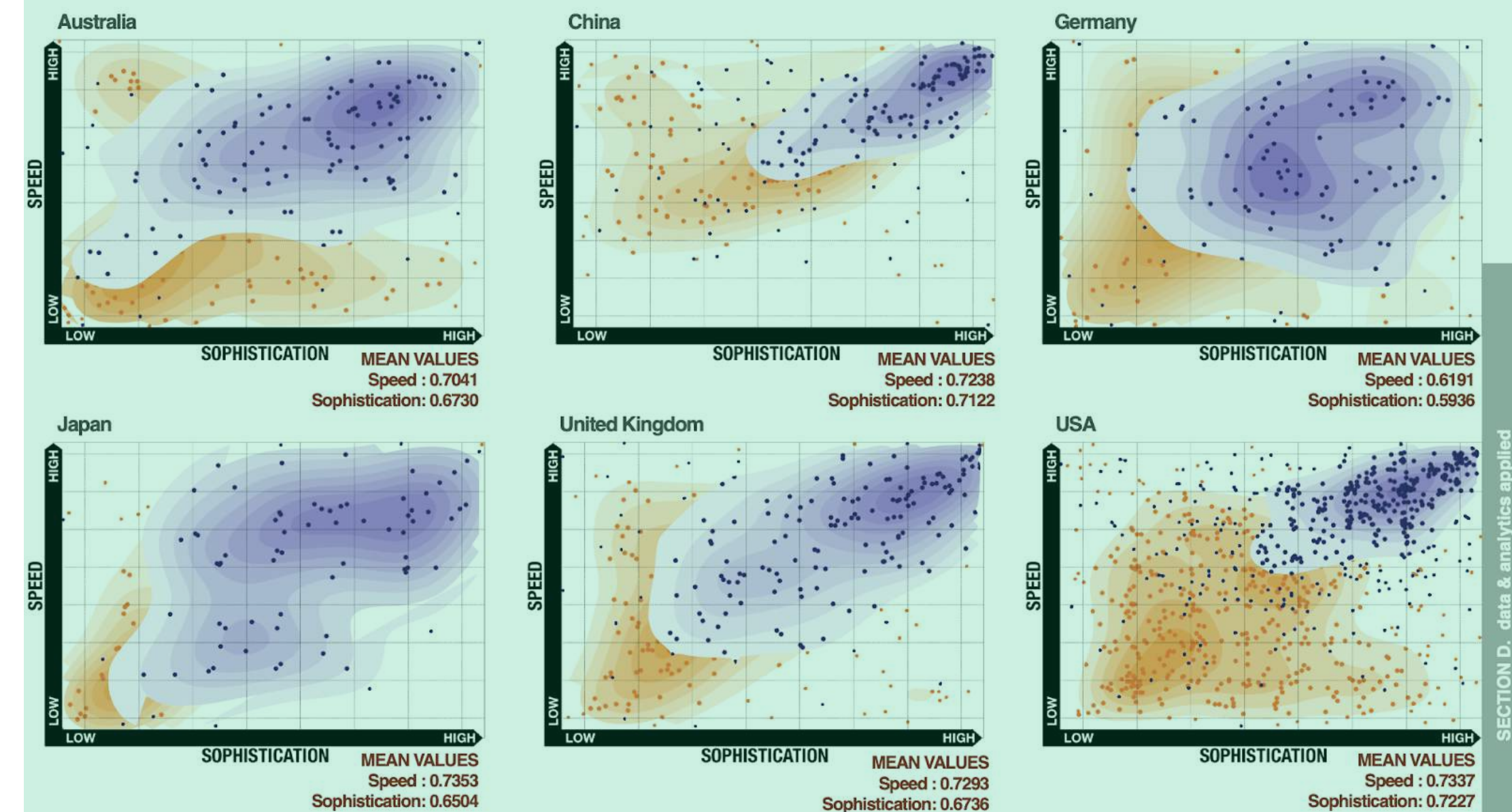
LANDSCAPE CONTRASTS BY COUNTRY

China displays the highest reliance on Internal data and analytics AND Machine algorithms



WHERE CAPABILITIES NEED TO BE IN 2020 BY COUNTRY

Across the countries, there appears to be a stronger desire for higher speed than sophistication



通过技术雷达，在团队级把握技术方向



技术趋势	技术能力	场景应用	场景需求
Immersive Experience	Wearables Voice Recognition	行车会议	“作为一个商业人士，我在下班回家过程中，想要继续完成我的会议...”
	Weather Monitoring	自动调节温度	“作为一个母亲，我在接送孩子前，想要将车内温度调至适合孩子...”
	Machine Learning Real-time Monitoring	疲劳驾驶提醒	“作为一个DiDi司机，我在行车过程中，想要及时识别疲劳驾驶状态...”
Fully Automatic Technology	Autonomous Driving Autonomous Parking IOT	智能泊车 车上办公室 移动空间	
....	...	...	...
	...	...	...
	...	...	...



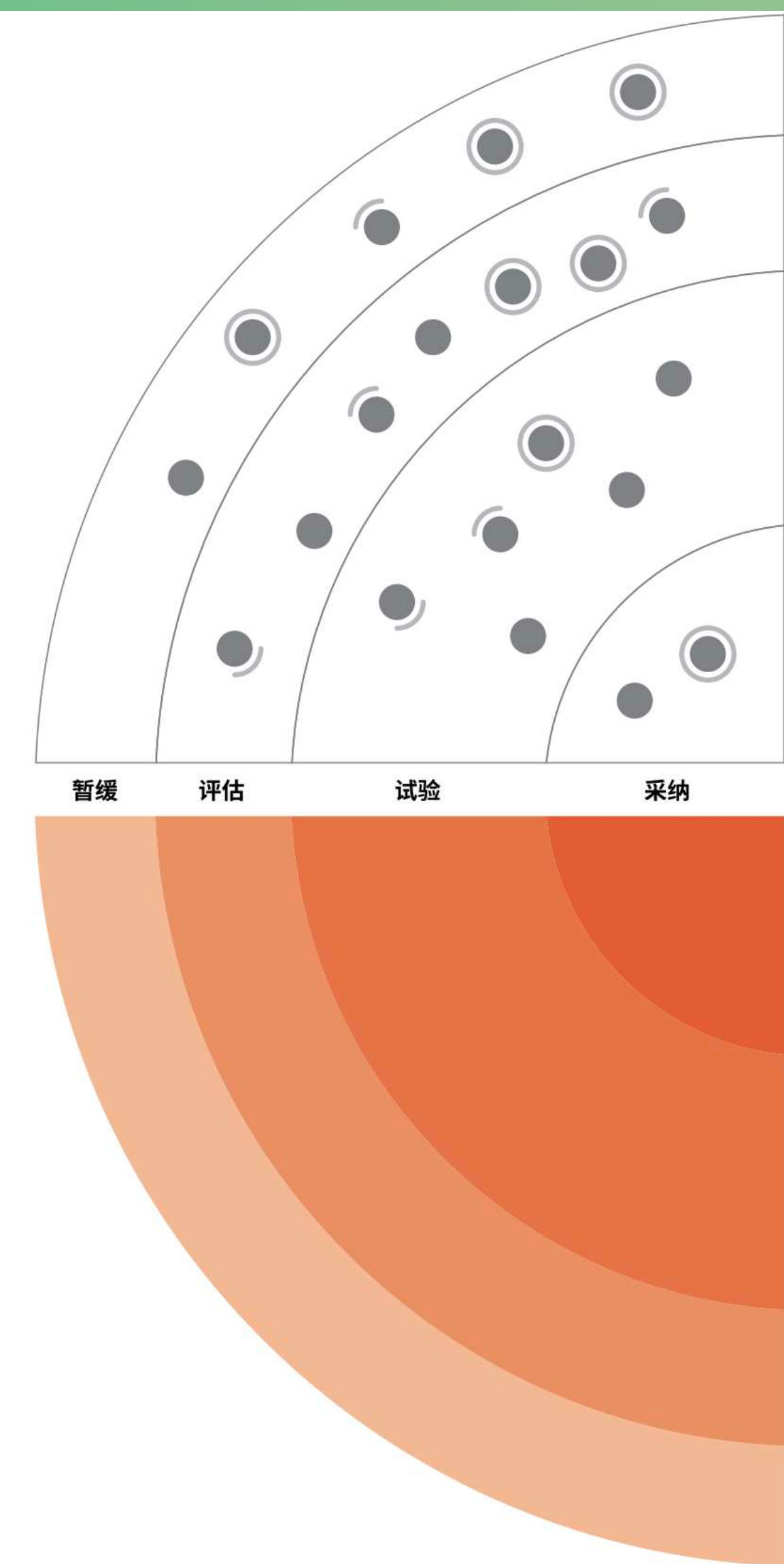
ThoughtWorks®

BUILD YOUR **OWN RADAR**

Set your tech priorities

通过技术雷达，提供技术投资战略的底层逻辑

ThoughtWorks®



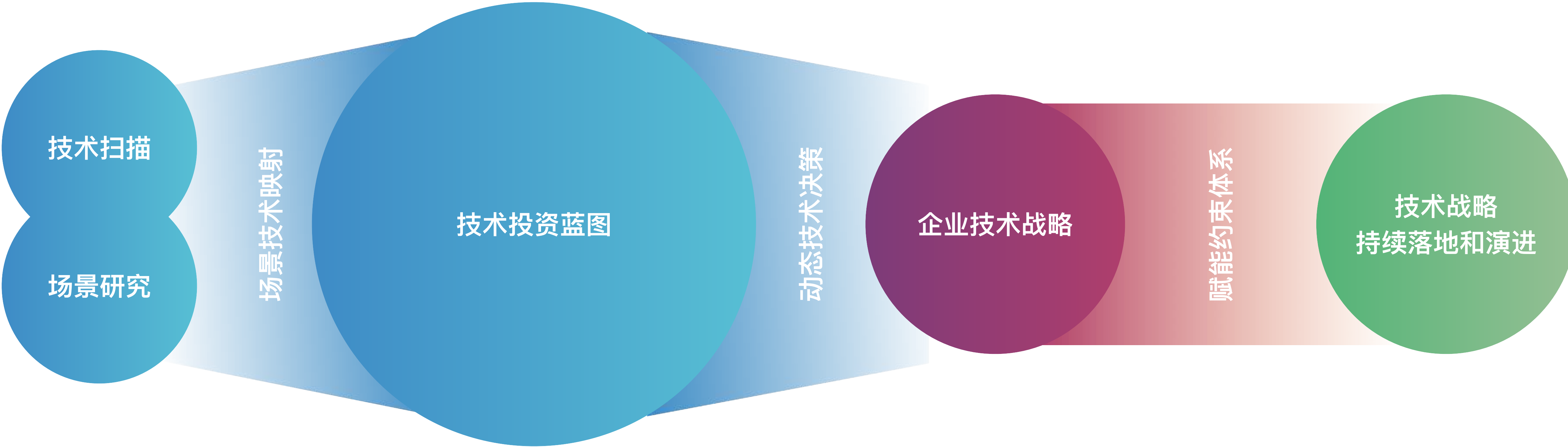
04

总结

ThoughtWorks®

技术场景双驱的技术投资策略

ThoughtWorks®



KEY TAKEAWAYS

ThoughtWorks®



THANK YOU

欲了解详情，请发送邮件至：

yujing.yang@thoughtworks.com

davidw@thoughtworks.com

ThoughtWorks®



ThoughtWorks®

用产品化的思维构建企业内部平台

王健 ThoughtWork 首席咨询师

01 平台

02 企业内部平台

03 挑战

04 破局

目录

ThoughtWorks®

01

平台

ThoughtWorks®

TECHNOLOGY RADAR *SUMMIT*

2017技术雷达峰会

洞察构建未来的技术和趋势

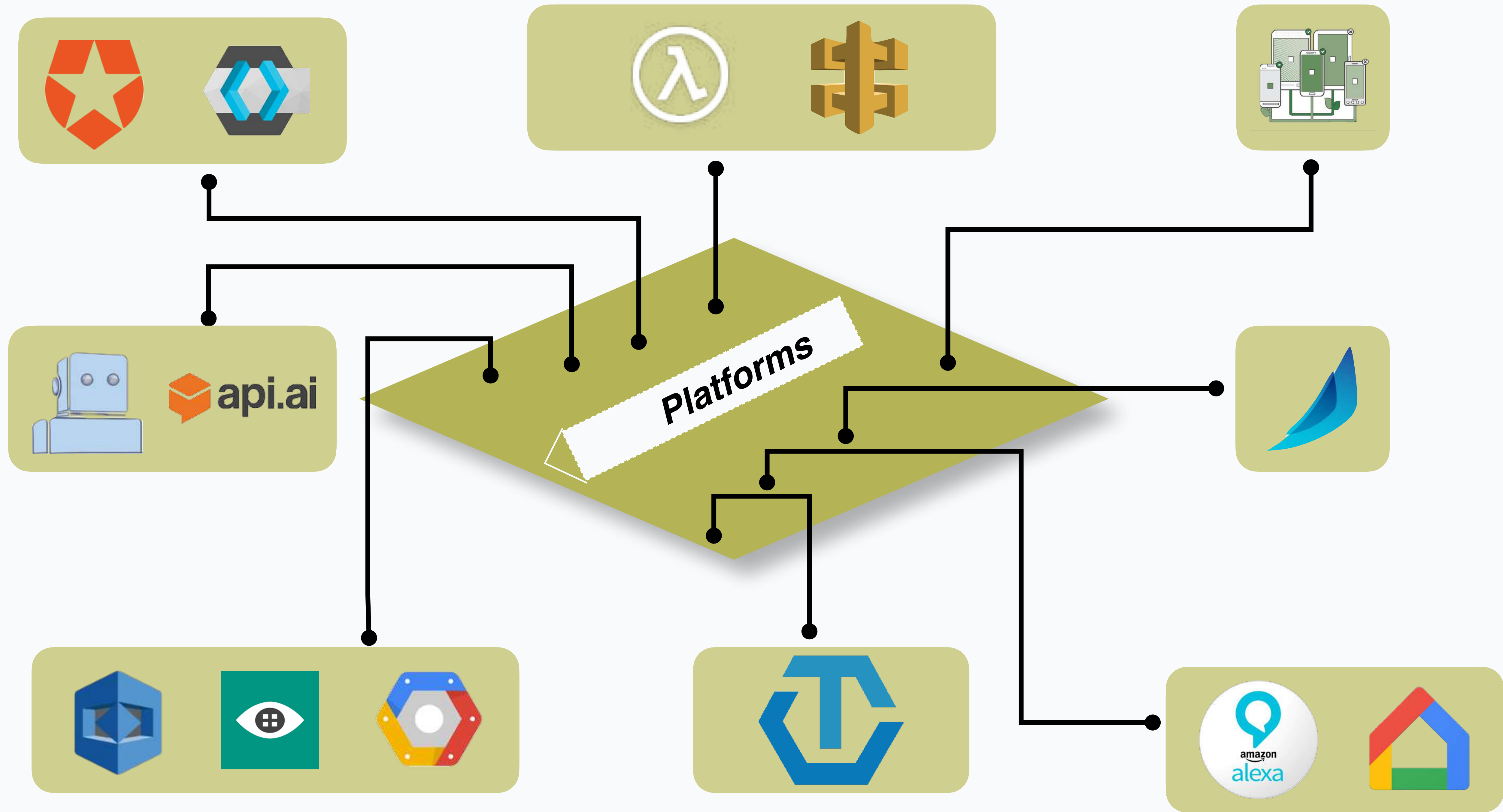


THE RISE THROUGH WORK

THE RISE OF PLATFORMS

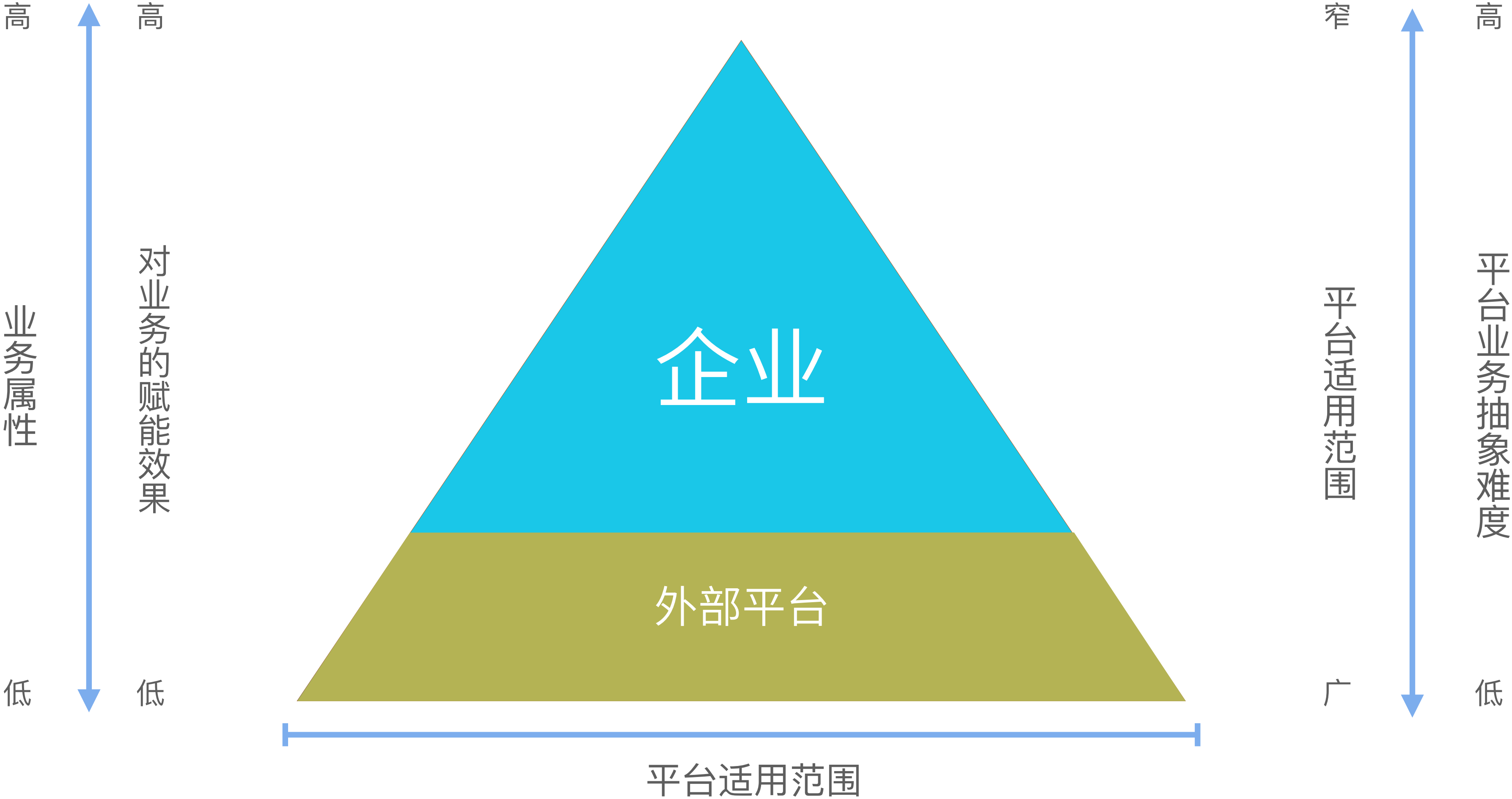


THE RISE OF PLATFORMS



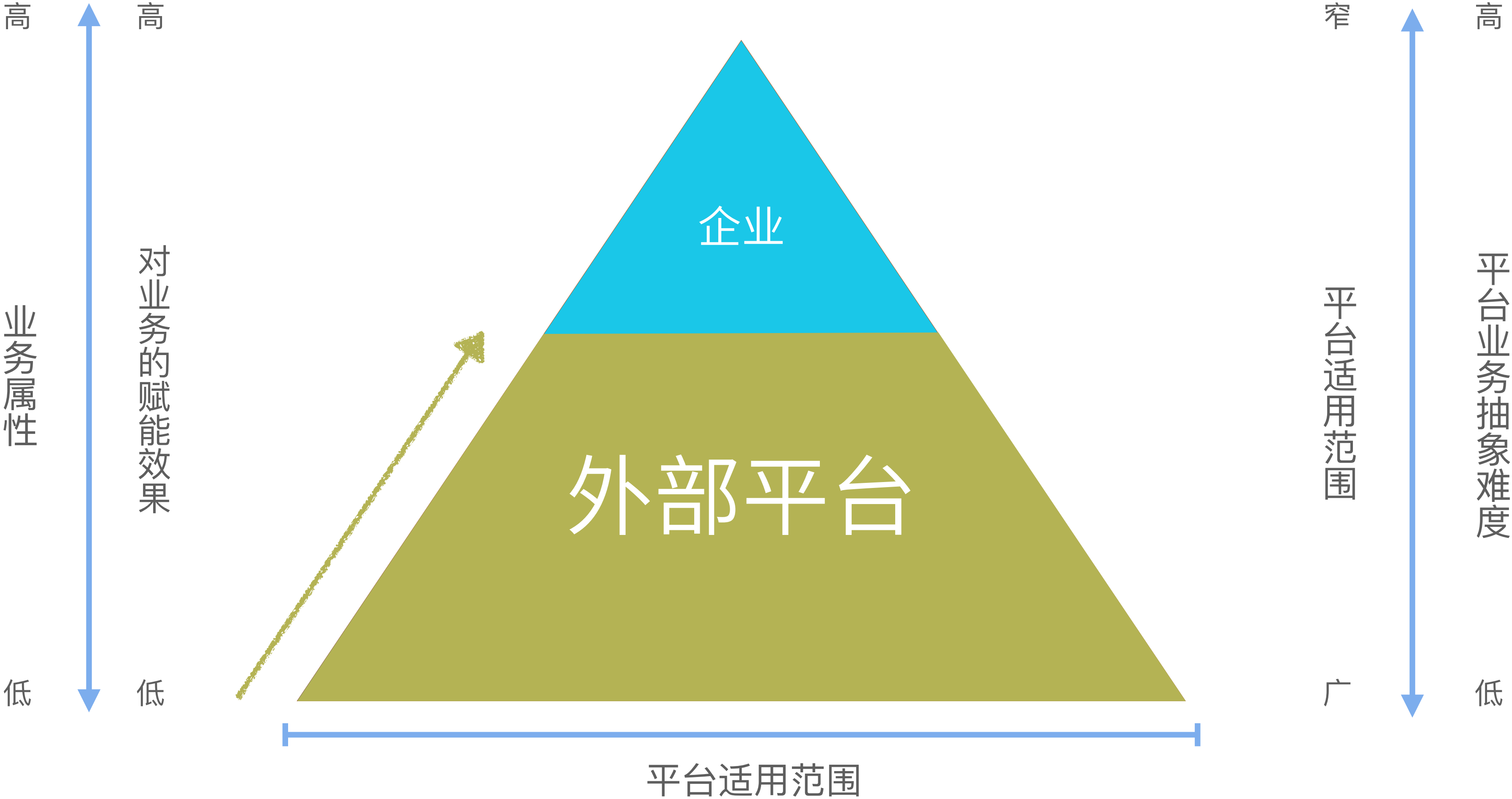
平台

为什么需要平台?



平台

为什么需要平台?



城市的本质
nature of city

公共服务
Public goods

政府
Government is a firm

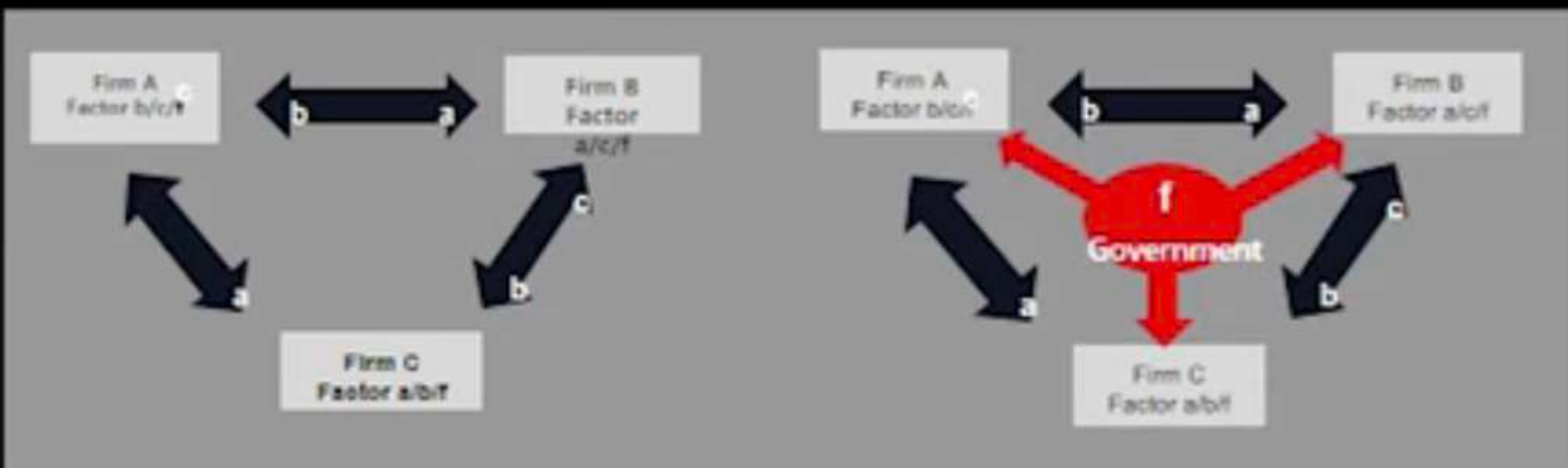
平台=公共产品

市场的结构

平台公司+社会公司

平台公司=政府

社会公司=厂商+家庭



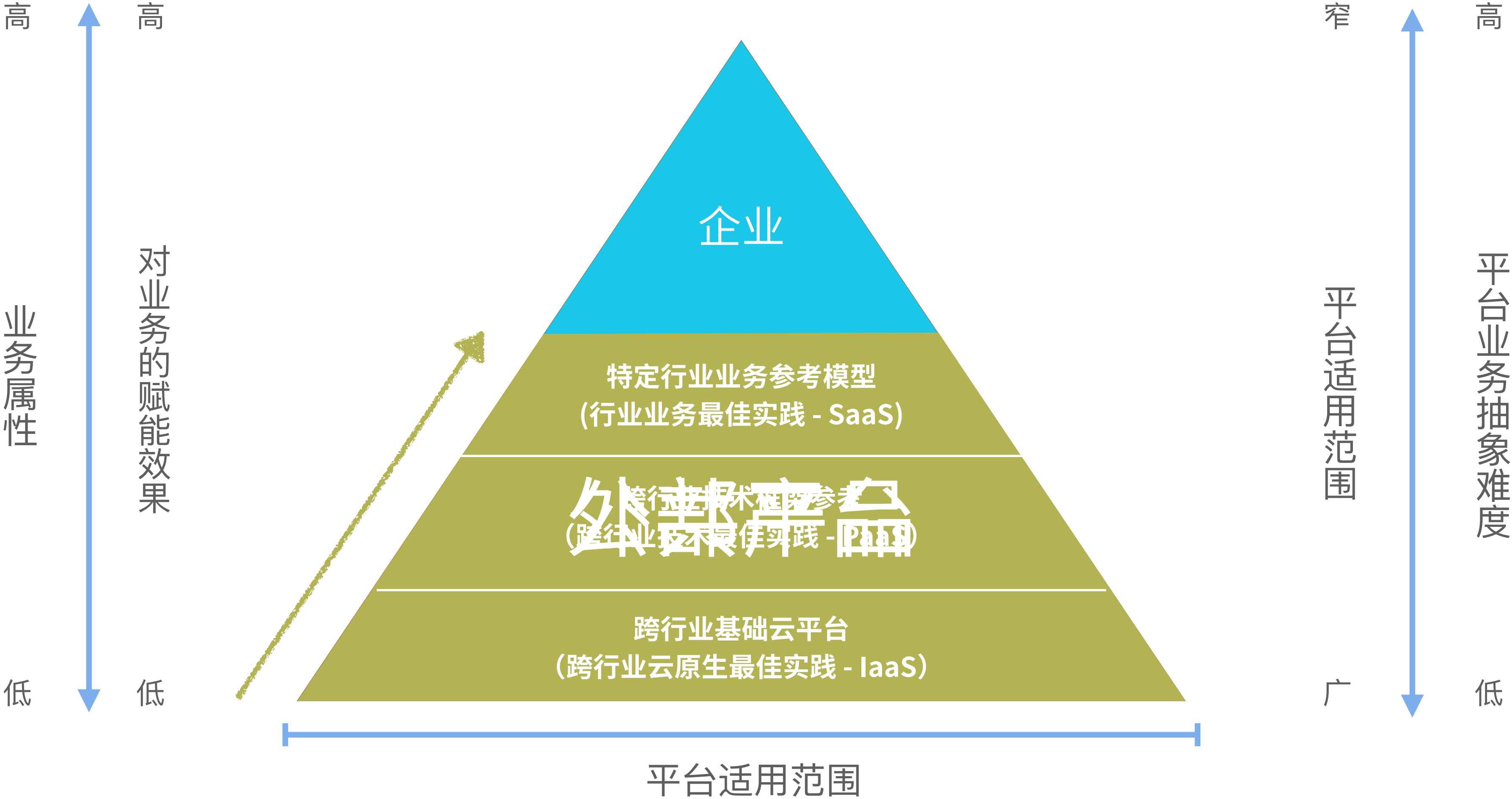
- 现实中最典型的平台企业就是传统经济学中公共产品的提供者——政府。对于每个厂商和家庭而言，基础设施（道路、电网、港口等）和服务（教育、医疗、安全、法律等）都是极其昂贵的。
- 由平台公司（政府）统一提供可以极大降低单一厂商和家庭的成本。而平台公司可以通过收费，获得投资回报。

平台经济一个特点就是它是一个公共产品



平台

为什么需要平台?



01

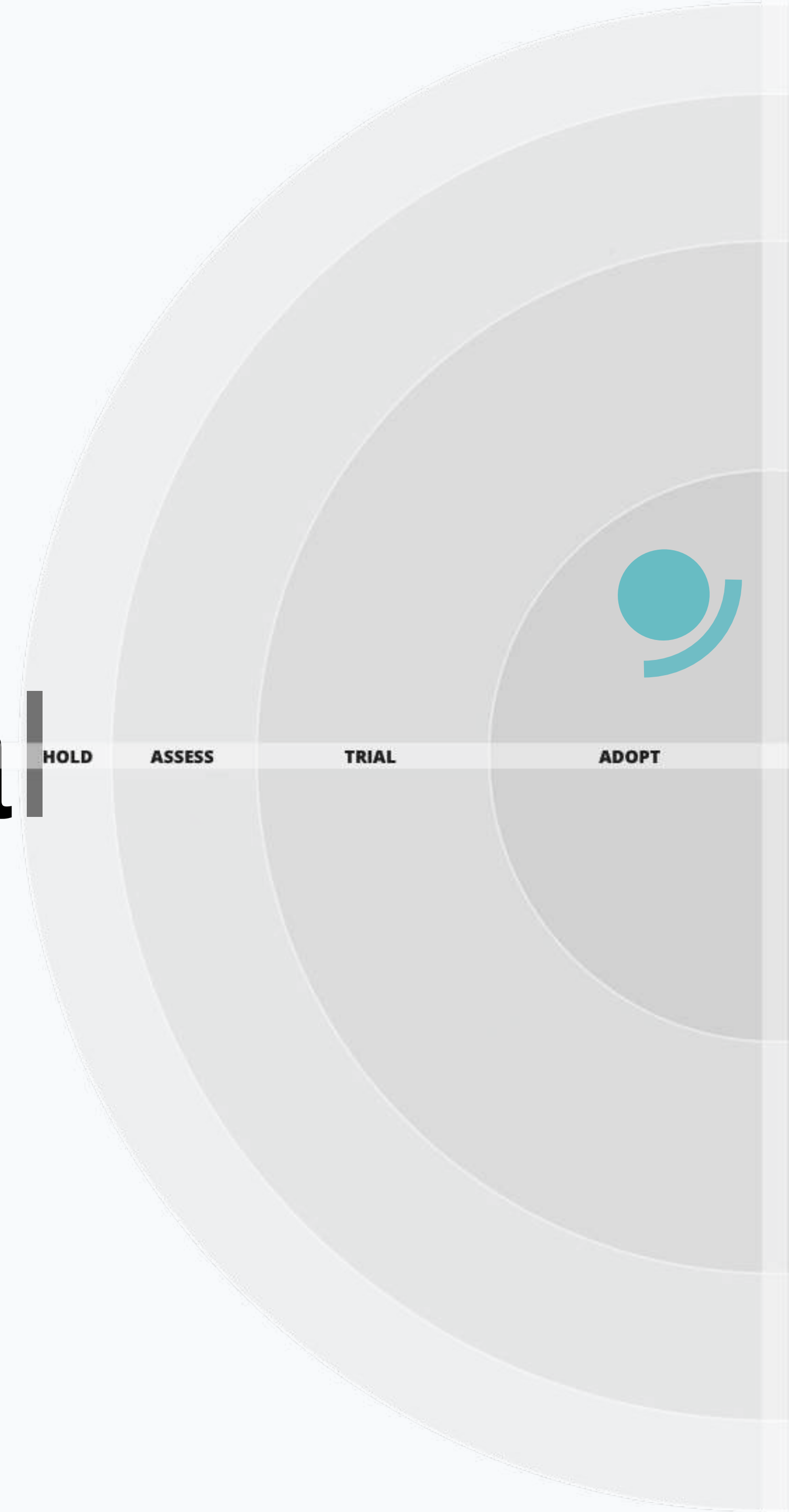
企业内部平台

ThoughtWorks®

Techniques

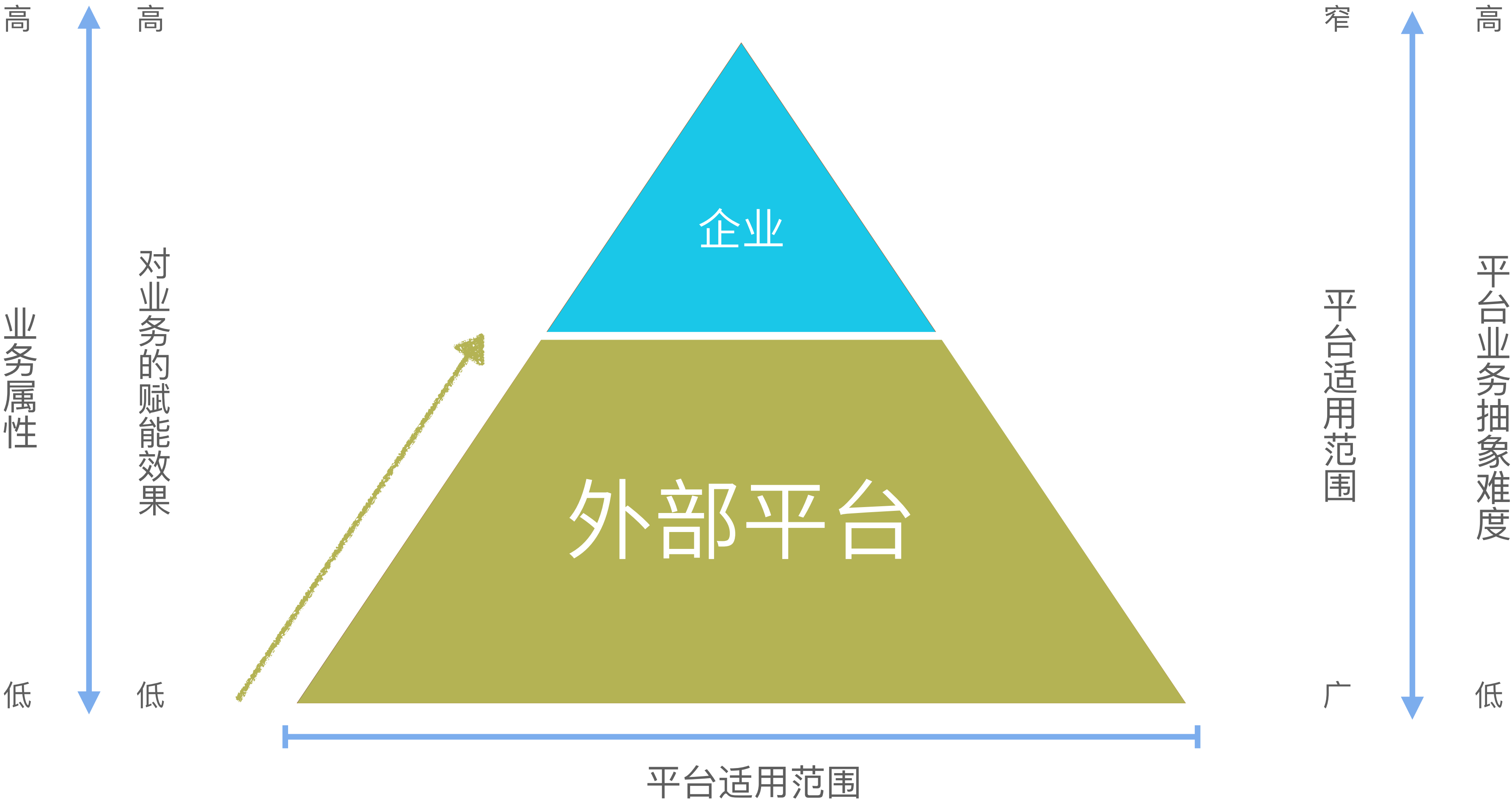
Applying product management to internal platforms

Adopt - Archived (2021)



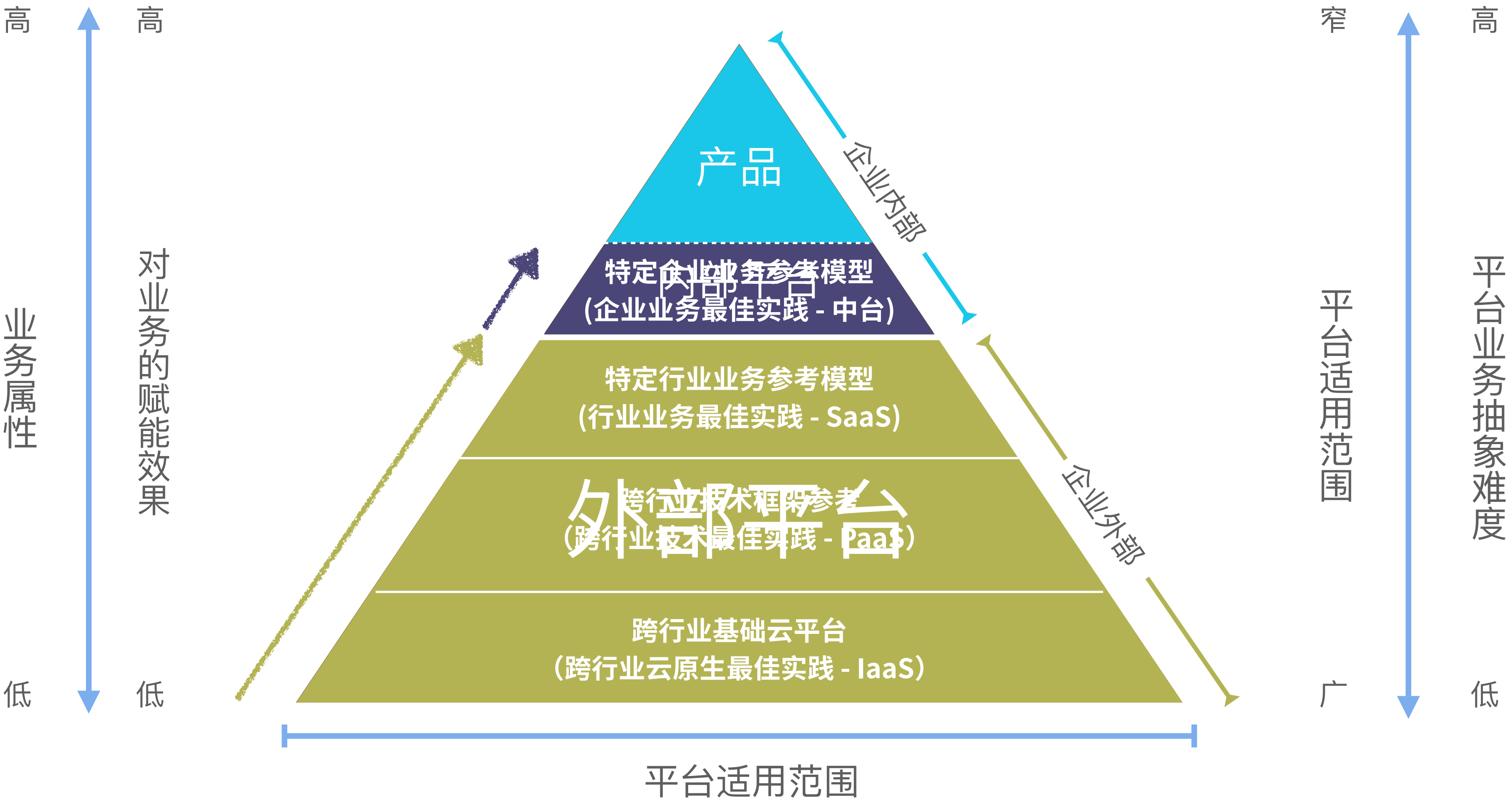
企业内部平台

什么是企业内部平台？



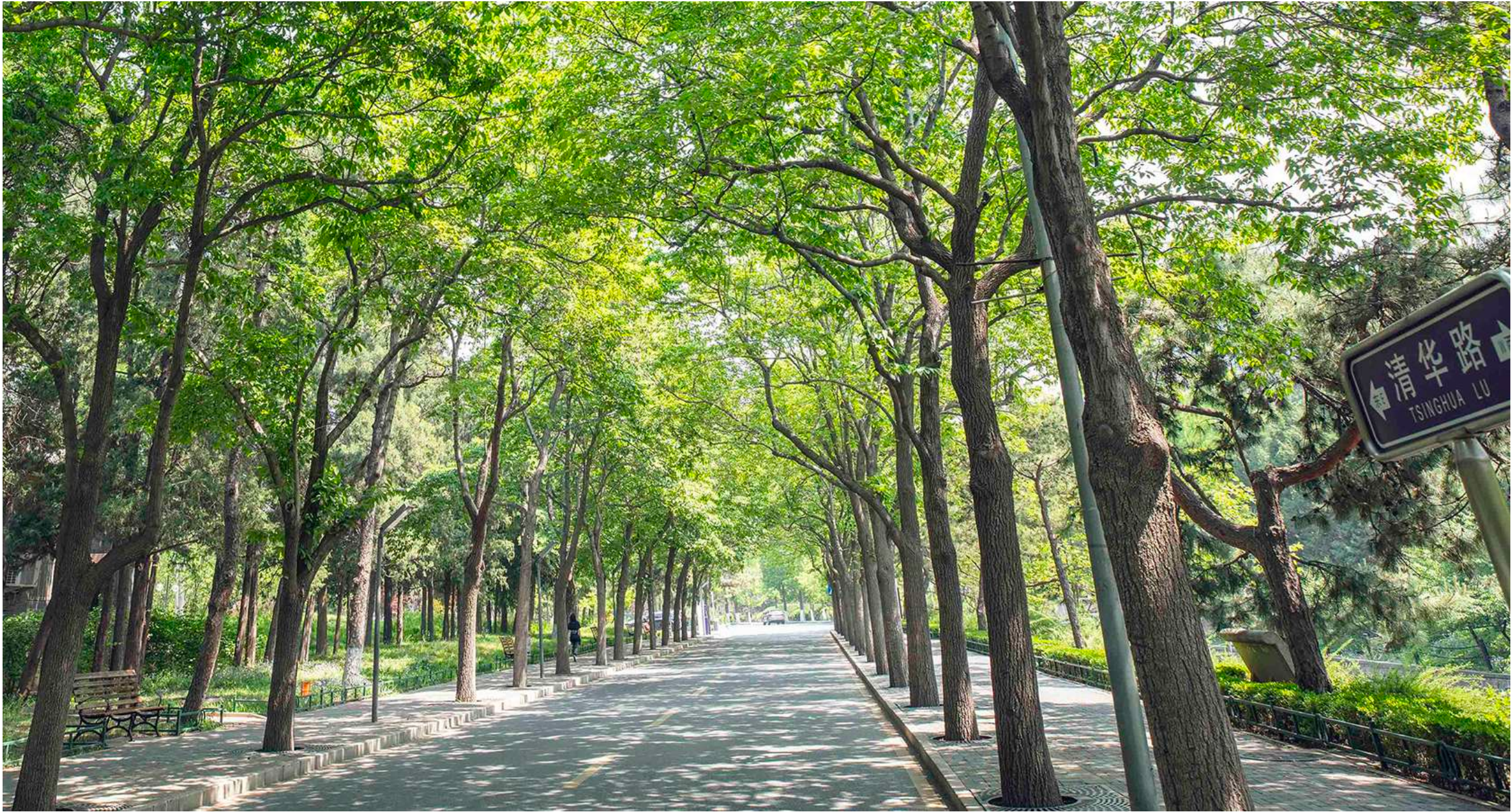
企业内部平台

什么是企业内部平台？



企业内部平台

什么是企业内部平台？



02

挑战

ThoughtWorks®

挑战

企业内部平台的最大挑战

区分平台企业与社会企业

观学院官方 bilibili

社会向右：充分竞争+自由+轻资产

- 私有化/反垄断/自由
- 集体化损失效率

平台向左：充分垄断+民主+重资产

- 公有化/垄断/民主
- 拆分会损失效率

最大难度：区分平台+社会+所有制

- 平台企业与社会企业的共生

由于缺少能将平台企业和社会企业分开的制度设计，人们只能在全部公有或全部私有之间进行制度选择。

《赵燕菁：平台反垄断，最好的办法既不是公有也不是私有》<https://www.bilibili.com/video/BV1uV411n7qM>

平台与产品的碰撞背后是两种制度的碰撞
(自由 vs 民主 | 反垄断 vs 垄断 | 效率 vs 经济 | 私有化 vs 公有制)

挑战

企业内部平台的最大挑战

区分平台企业与社会企业

观学院官方 bilibili

社会向右：充分竞争+自由+轻资产

- 私有化/反垄断/自由
- 集体化损失效率

平台向左：充分垄断+民主+重资产

- 公有化/垄断/民主
- 拆分会损失效率

最大难度：区分平台+社会+所有制

- 平台企业与社会企业的共生

由于缺少能将平台企业和社会企业分开的制度设计，人们只能在全部公有或全部私有之间进行制度选择。

《赵燕菁：平台反垄断，最好的办法既不是公有也不是私有》<https://www.bilibili.com/video/BV1uV411n7qM>



平台与产品的碰撞背后是两种制度的碰撞

(自由 vs 民主 | 反垄断 vs 垄断 | 效率 vs 经济 | 私有化 vs 公有制)

挑战

企业内部平台的最大挑战

区分平台企业与社会企业

观学院官方 bilibili

社会向右：充分竞争+自由+轻资产

- 私有化/反垄断/自由
- 集体化损失效率

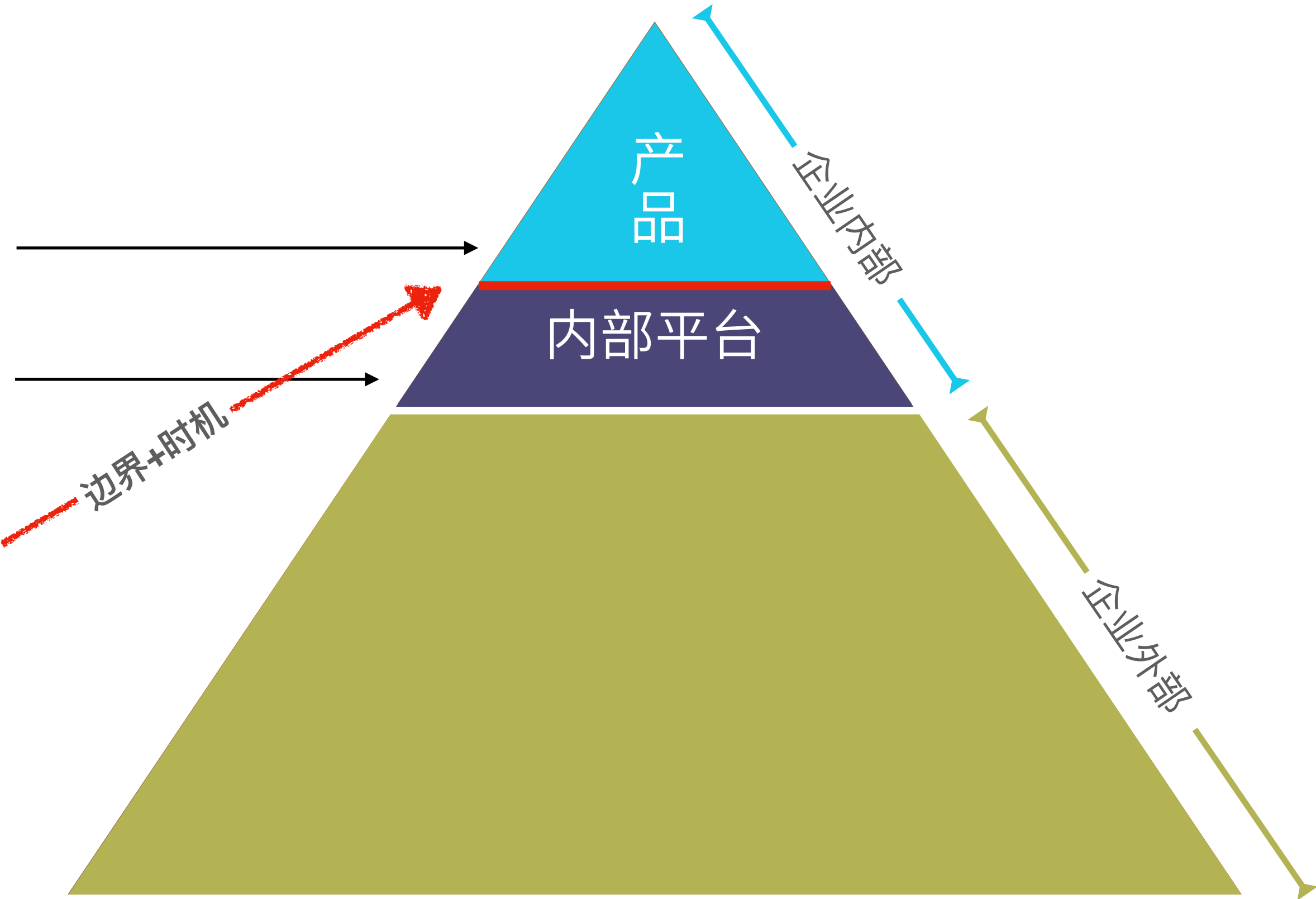
平台向左：充分垄断+民主+重资产

- 公有化/垄断/民主
- 拆分会损失效率

最大难度：区分平台+社会+所有制

- 平台企业与社会企业的共生

由于缺少能将平台企业和社会企业分开的制度设计，人们只能在全部公有或全部私有之间进行制度选择。



最大难度： 怎么区分？
(平台越接近业务越难，纠缠在一起)

挑战
企业内部平台建设的前置条件

平台如何定位

愿景?

组织?

组织如何适配

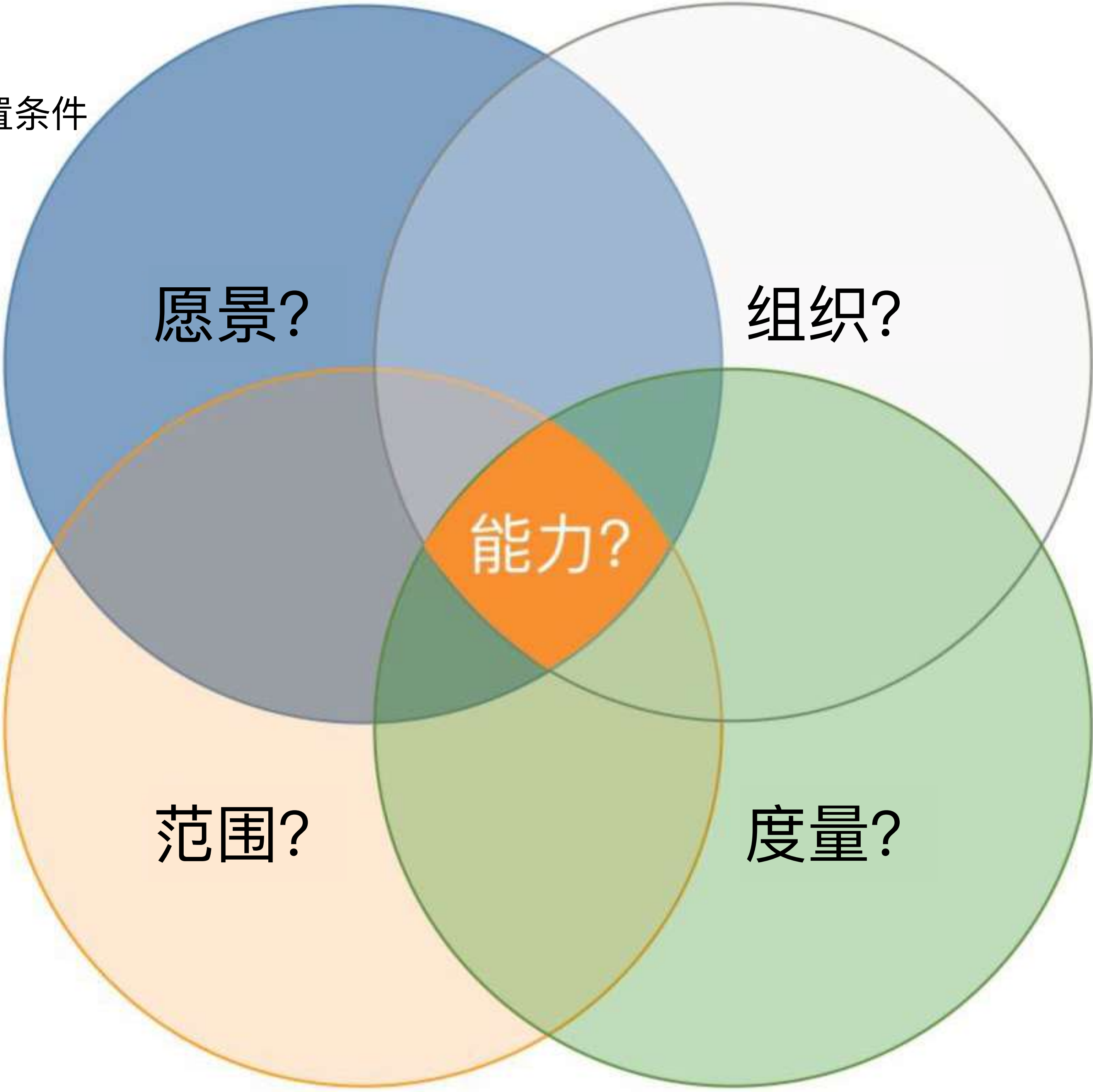
能力?

边界如何界定

范围?

度量?

结果如何度量



03

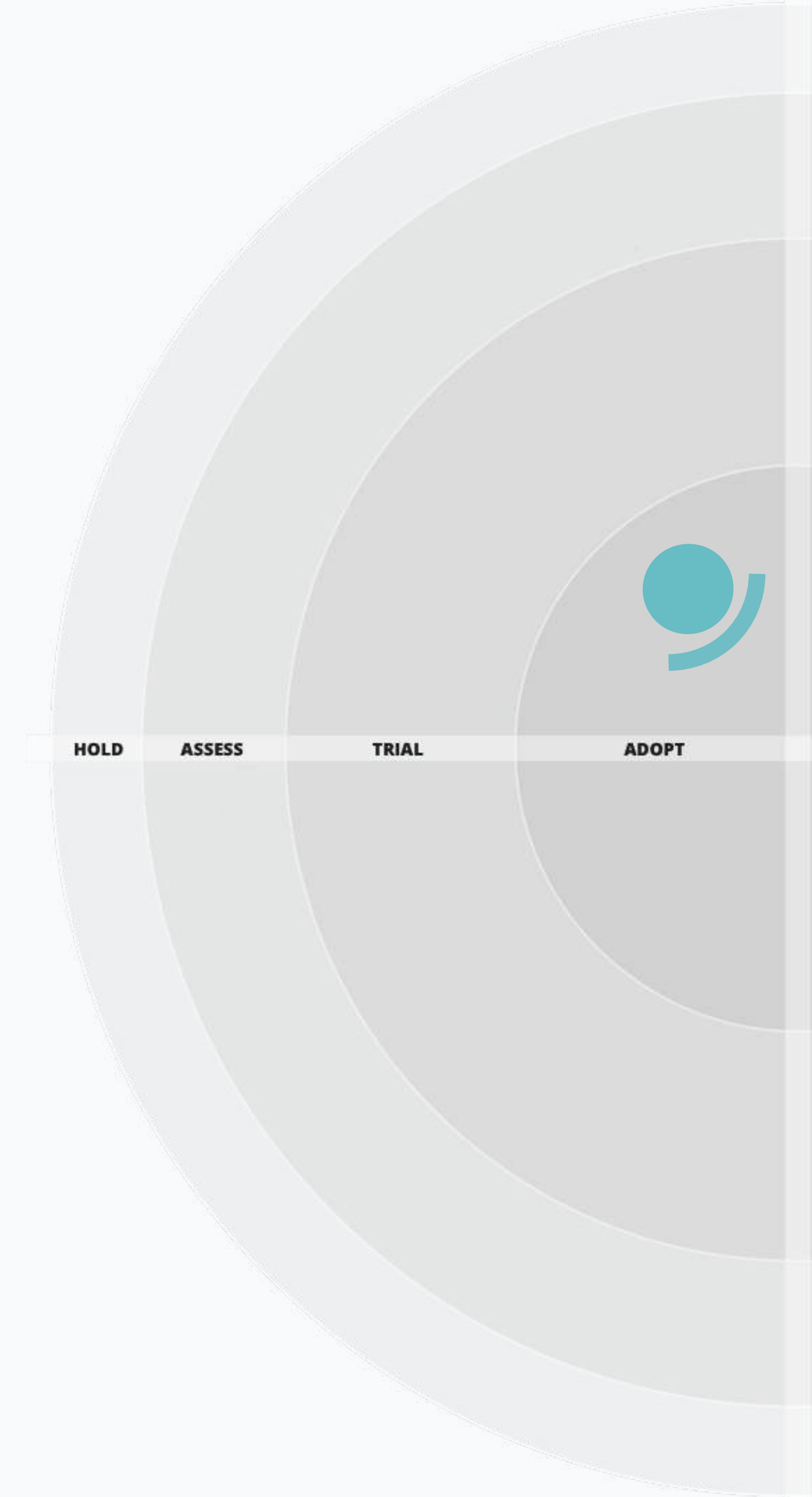
应对

ThoughtWorks®

Techniques

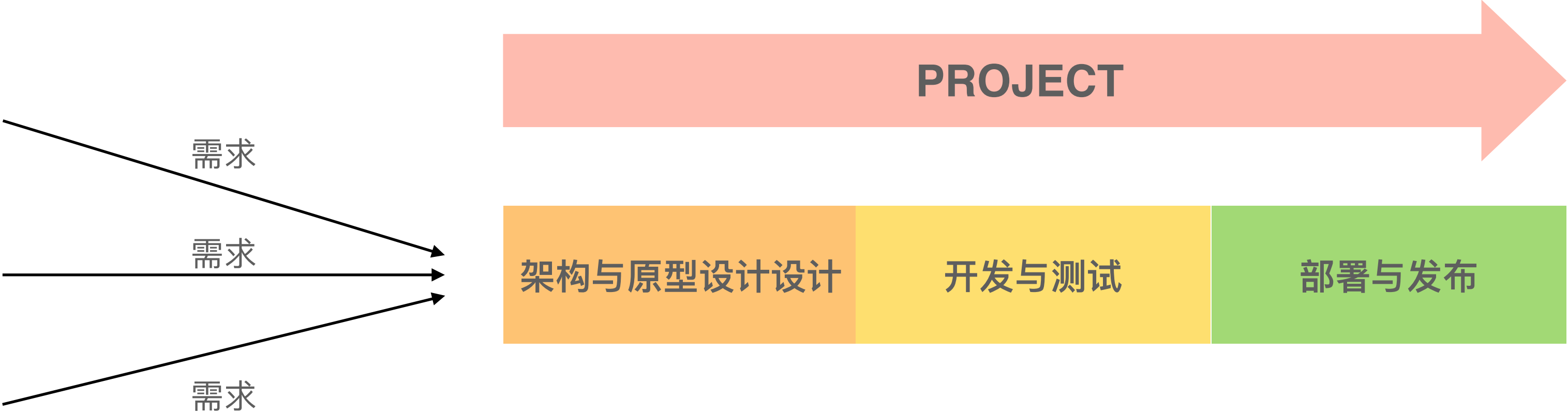
products over projects

Adopt - Archived (2016)



应对

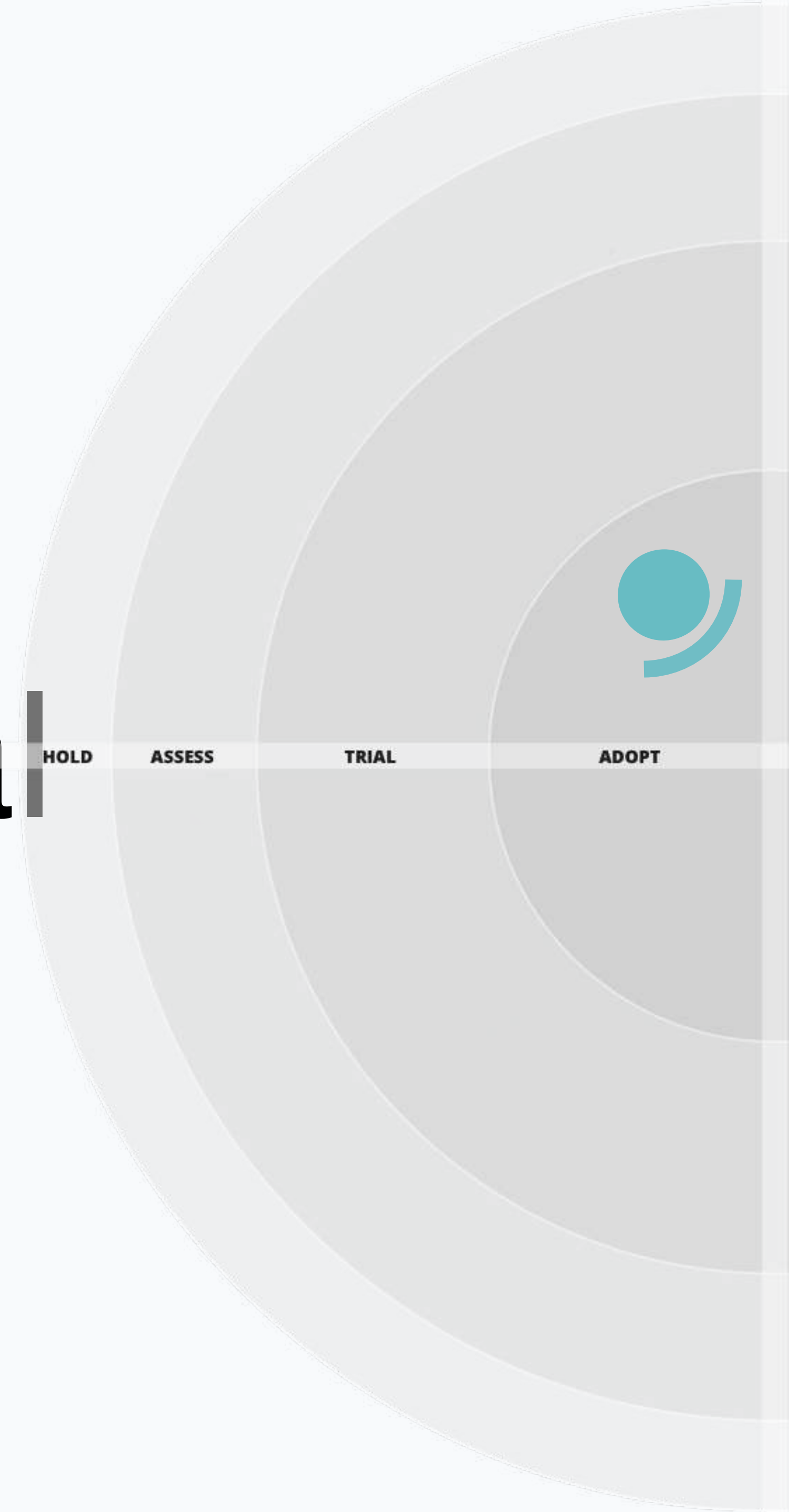
采用项目制构建企业内部平台的问题： 只能处理确定性问题



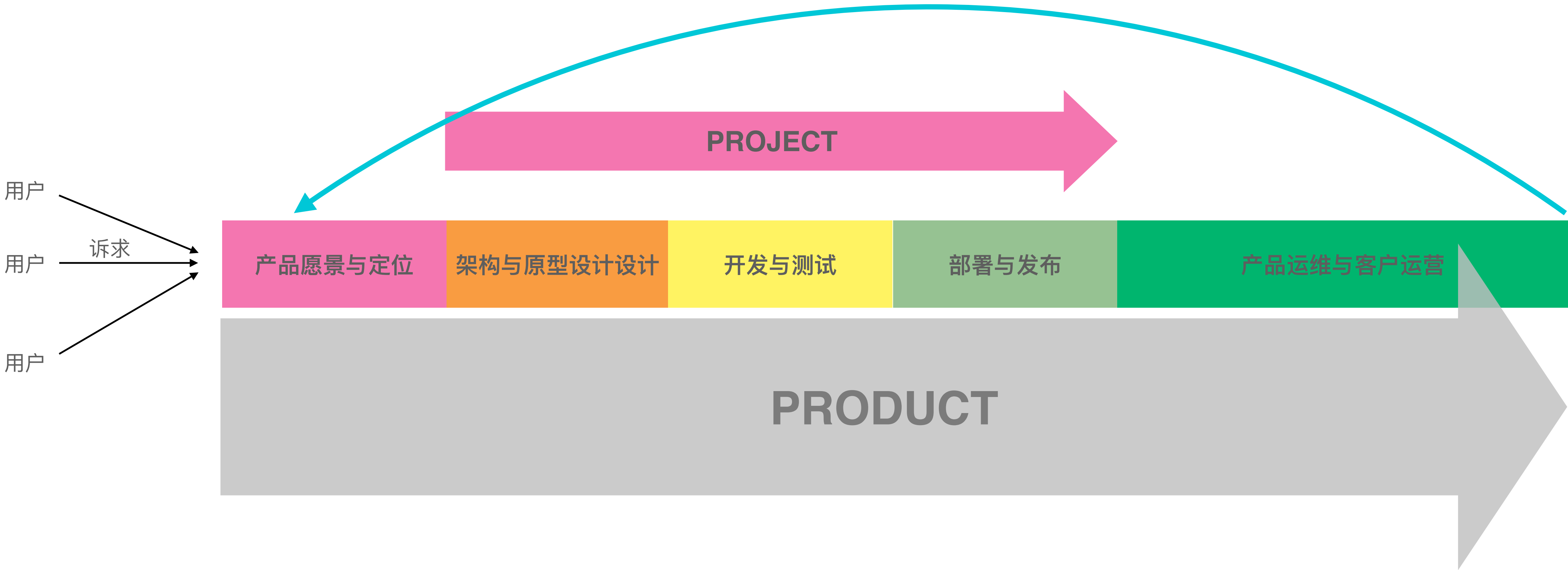
Techniques

Applying product management to internal platforms

Adopt - Archived (2020)

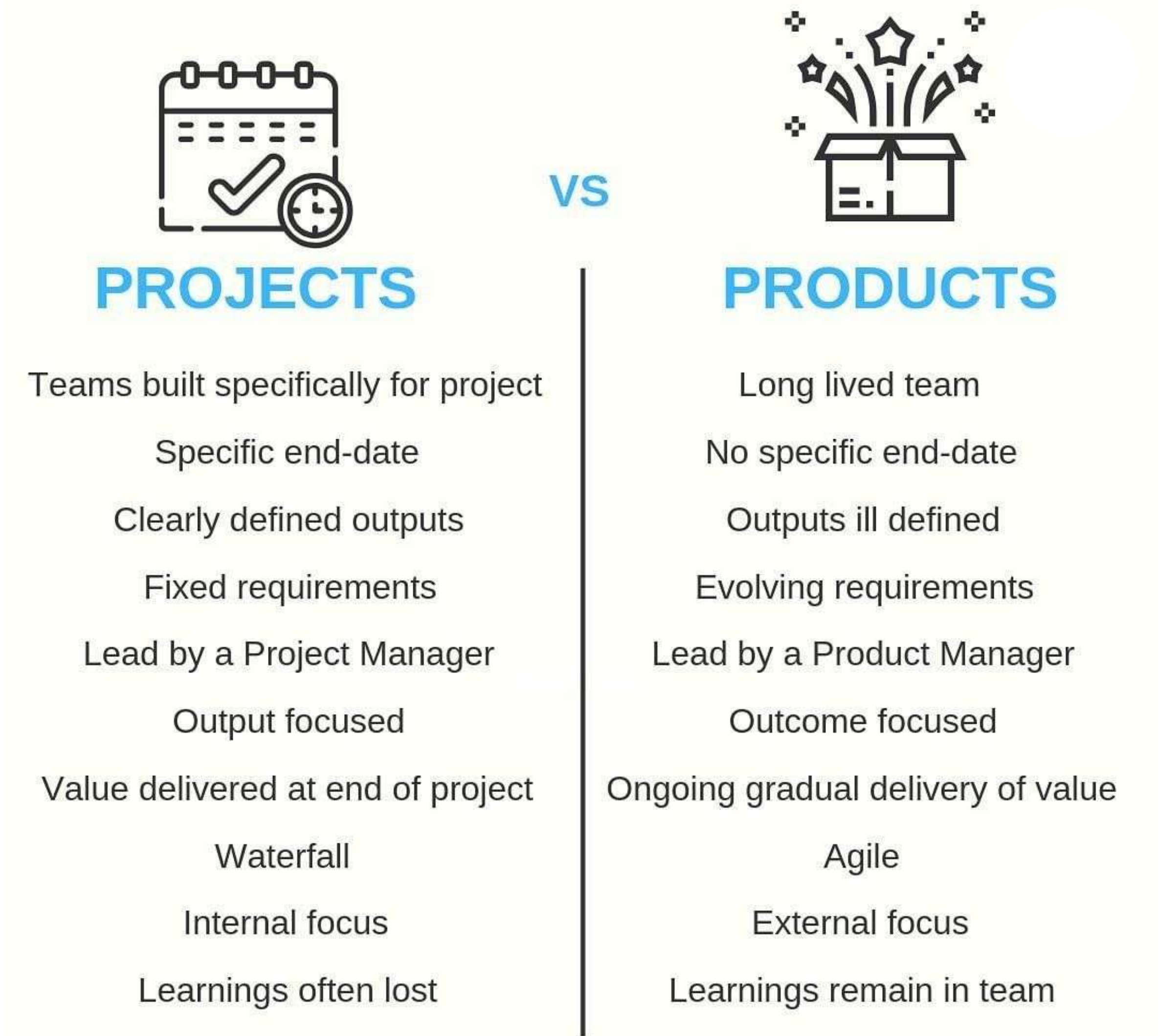


应对
产品制over项目制：企业内部产品是个高度不确定性的问题



应对
产品制over项目制：企业内部产品是个高度不确定性的问题

产品制的思维和项目制思维最本质的区别是，项目制是以确定性（IT要解决的问题是确定的）为基础的，因而注重计划、预算、执行效率（人力利用率）；产品制的基本观则是，IT所要解决的问题都是高度变化的，因此应该不断挖掘客（用）户的真正需要，持续迭代更新解决方案，一切以提供给客（用）户的价值为目标。



应对

产品制over项目制：企业内部产品是个高度不确定性的问题

产品化思维-企业内部平台



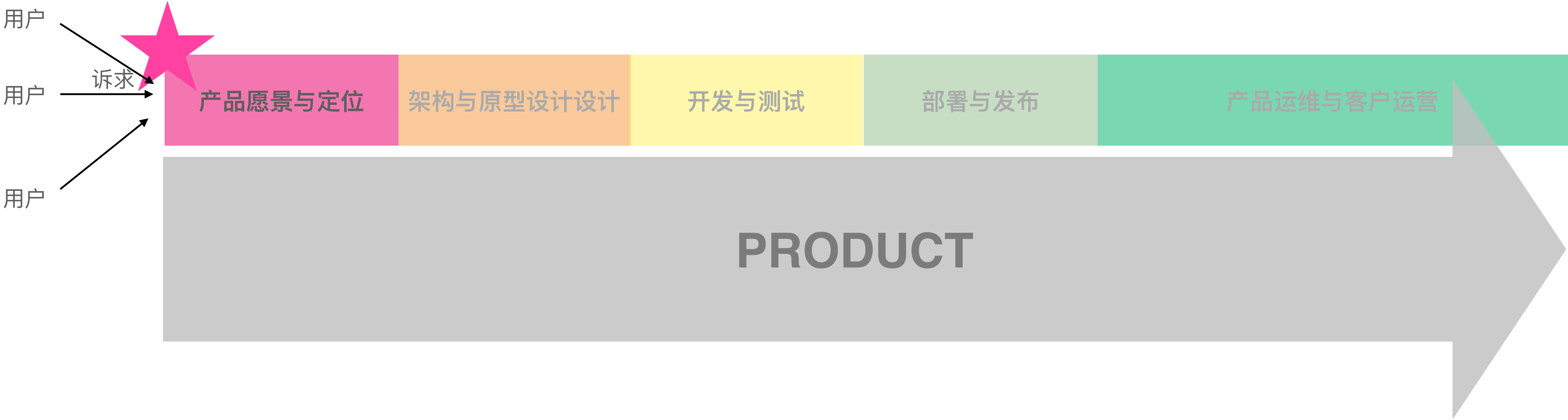
1. 愿景是什么？
2. 用户是谁？用户划分？
3. 为用户解决什么问题？
4. 竞争环境是什么？团队构成？
5. 如何获取用户？营销？售前？
6. 如何向用户提供服务？自助？定制？
7. 如何验证价值？成本核算？定价机制？谁来买单？
8. 如何保证服务质量？SLA？售后？
9. 如何升级演进？产品运营？客户渠道关系维护？

04

方法

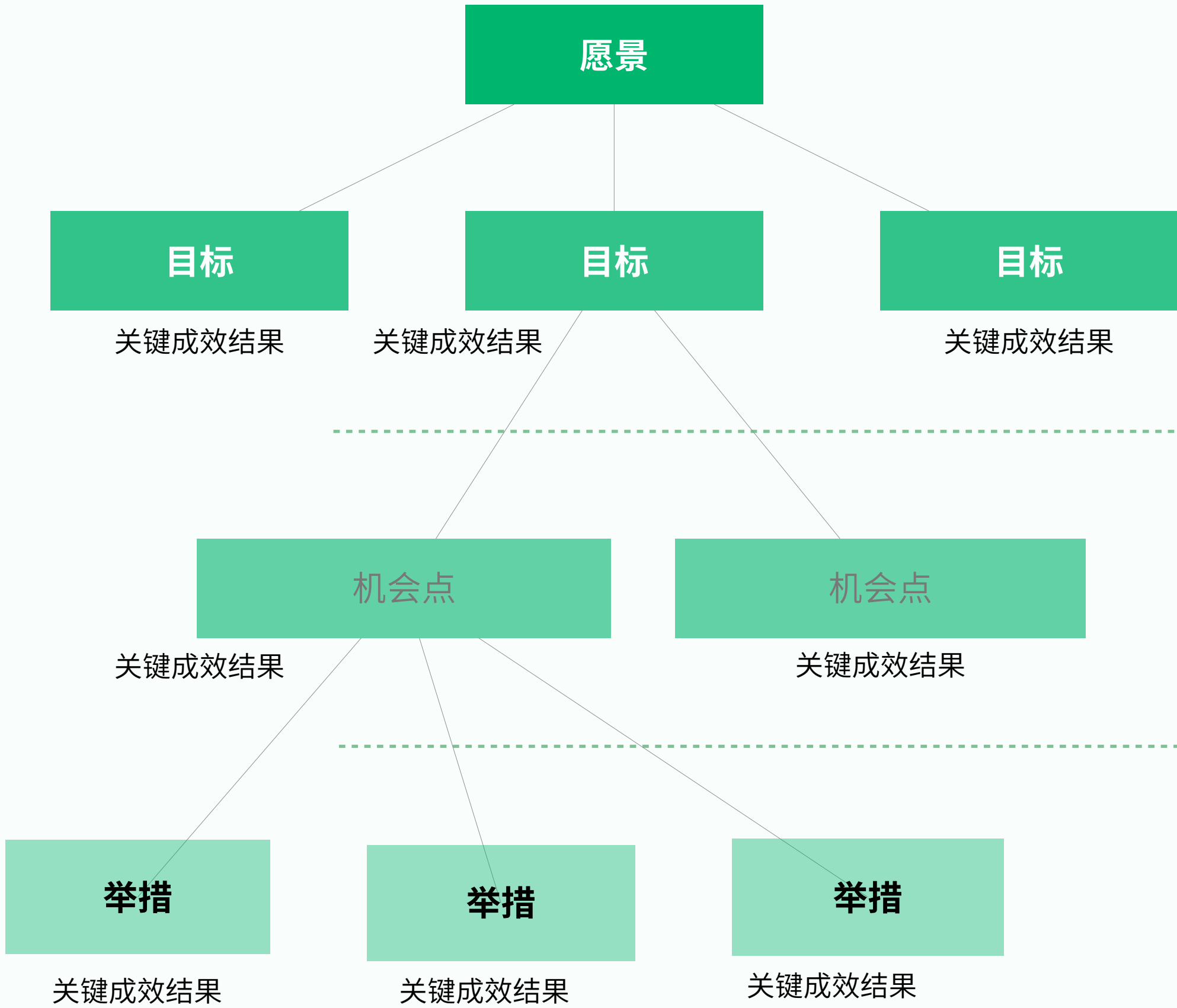
ThoughtWorks®

应对
产品制over项目制



应对

产品制 - 愿景与定位



精益价值树来捕捉和共享组织级愿景和战略

愿景与目标

- 由组织管理层制定愿景和目标。“**愿景**”设置了一个总体指导方向，所有投资都应该为其做出贡献；
- 围绕业务愿景，每年进行一次整体规划，形成业务的发展目标，体现该业务的发展策略与竞争策略；
- 每个目标会设置基于业务成效的关键结果，并设置负责人。

机会点

- 由业务领域规划为实现领域目标所需的投资点假设；
- 每个机会点需要有明确的关键成效结果衡量指标；
- 一般由最相关的业务领域作为负责人。

举措

- 体现有形价值的行动。在整体规划中会形成一系列面向业务成效的专题，并在产品迭代中也会持续浮现新的机会，这些专题和新的机会会转化为具体的产品特性、项目需求等，即举措。
- 每个举措也需制定明确的关键成效结果衡量指标；由领域或特性团队作为负责人

应对

产品制 - 愿景与定位

产品
愿景

- 目标客户/用户是谁？
- 聚焦要解决的问题？
- 产品的外观和形态如何？
- 产品成功的关键属性？
- 相关竞品或已有解决方案
- 独特价值和优势是什么？



产品
愿景

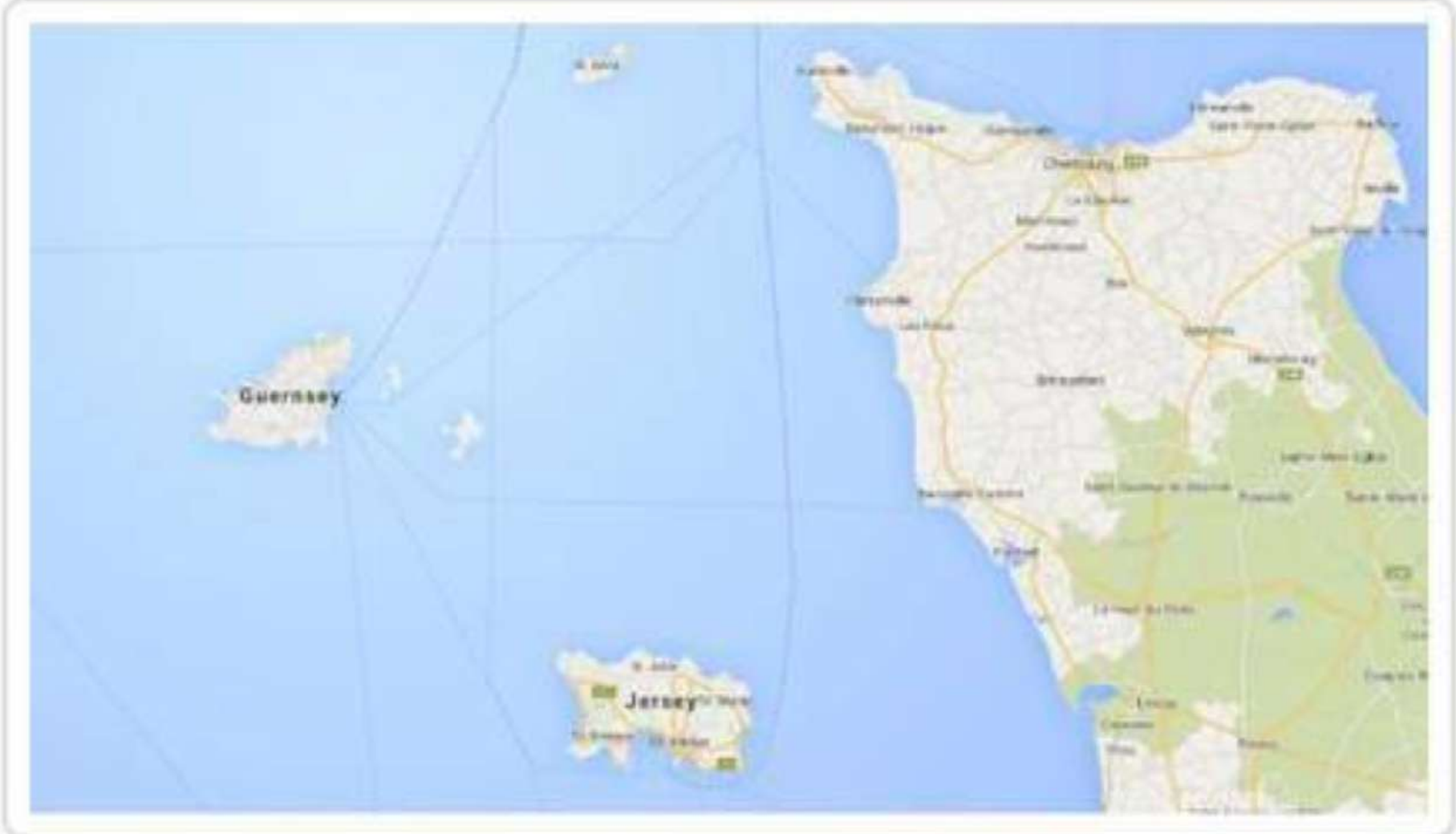
- 目标客户/用户是谁？
- 聚焦要解决的问题？
- 产品的外观和形态如何？
- 产品成功的关键属性？
- 相关竞品或已有解决方案
- 独特价值和优势是什么？

前台产研团队？业务大老板？
创新业务？移除孤岛？构建生态？
低代码/No代码？H5？API？MQ？
统一账户？稳定？架构前瞻性？
当前的解决方案是什么？
新体验 > 旧体验 + 迁移成本？

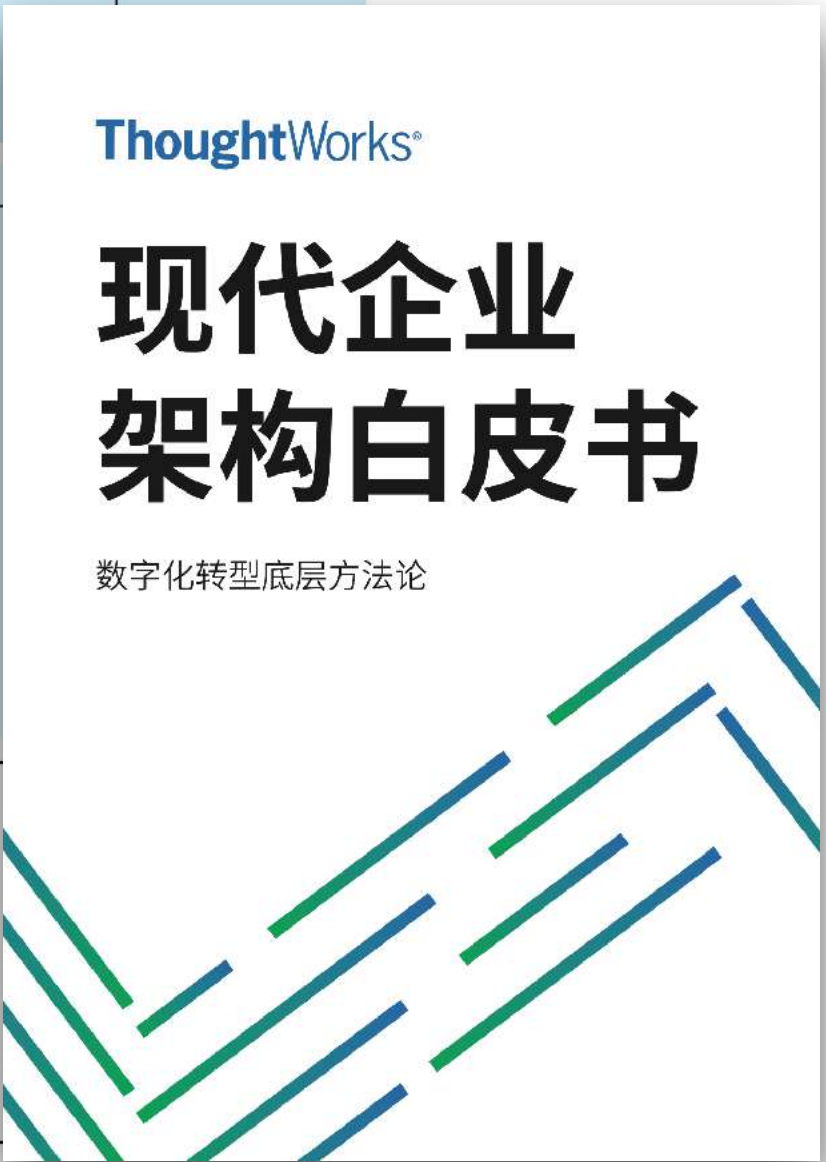
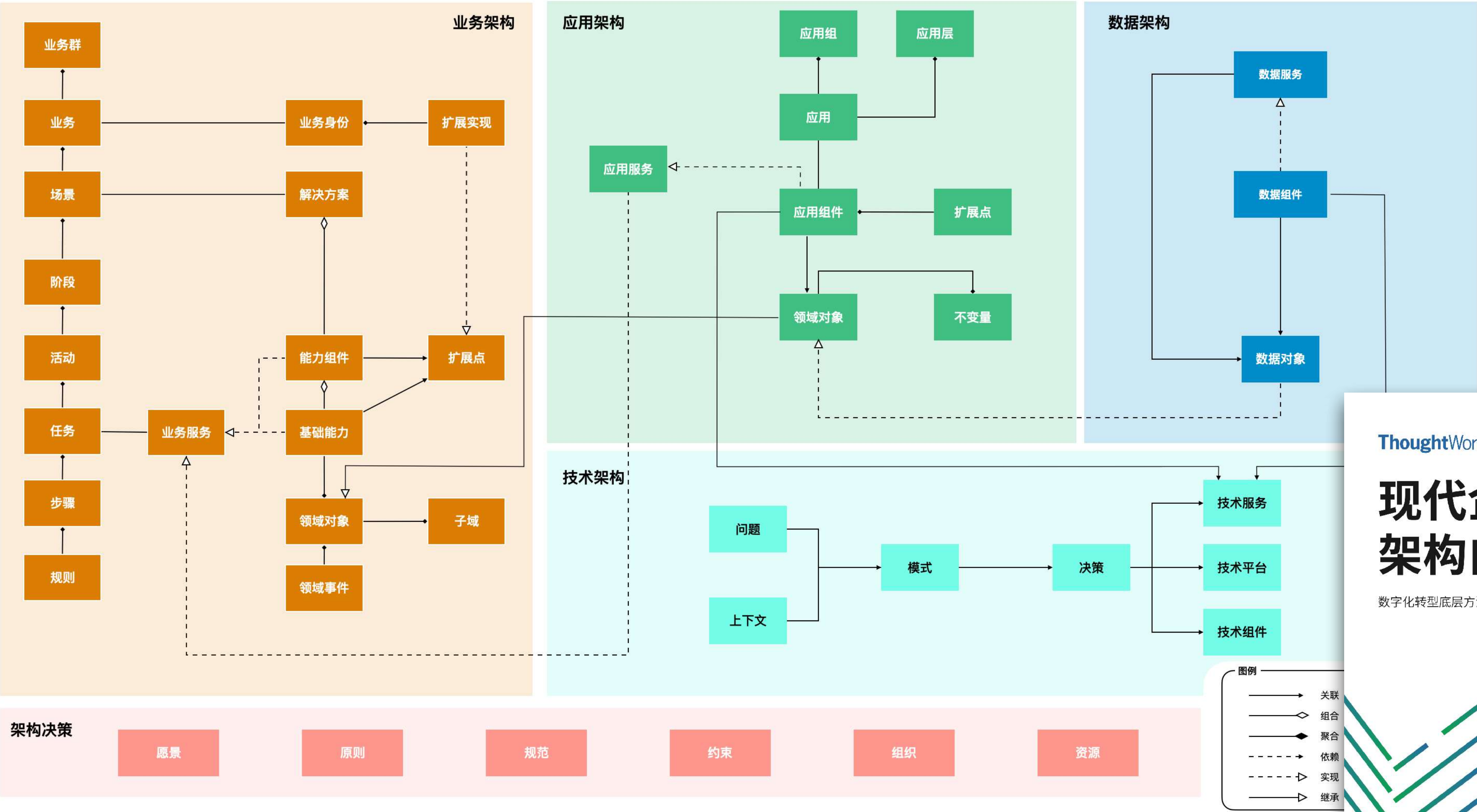
应对
产品制over项目制



应对
产品制 - 产品与架构设计



应对
产品制 - 产品与架构设计



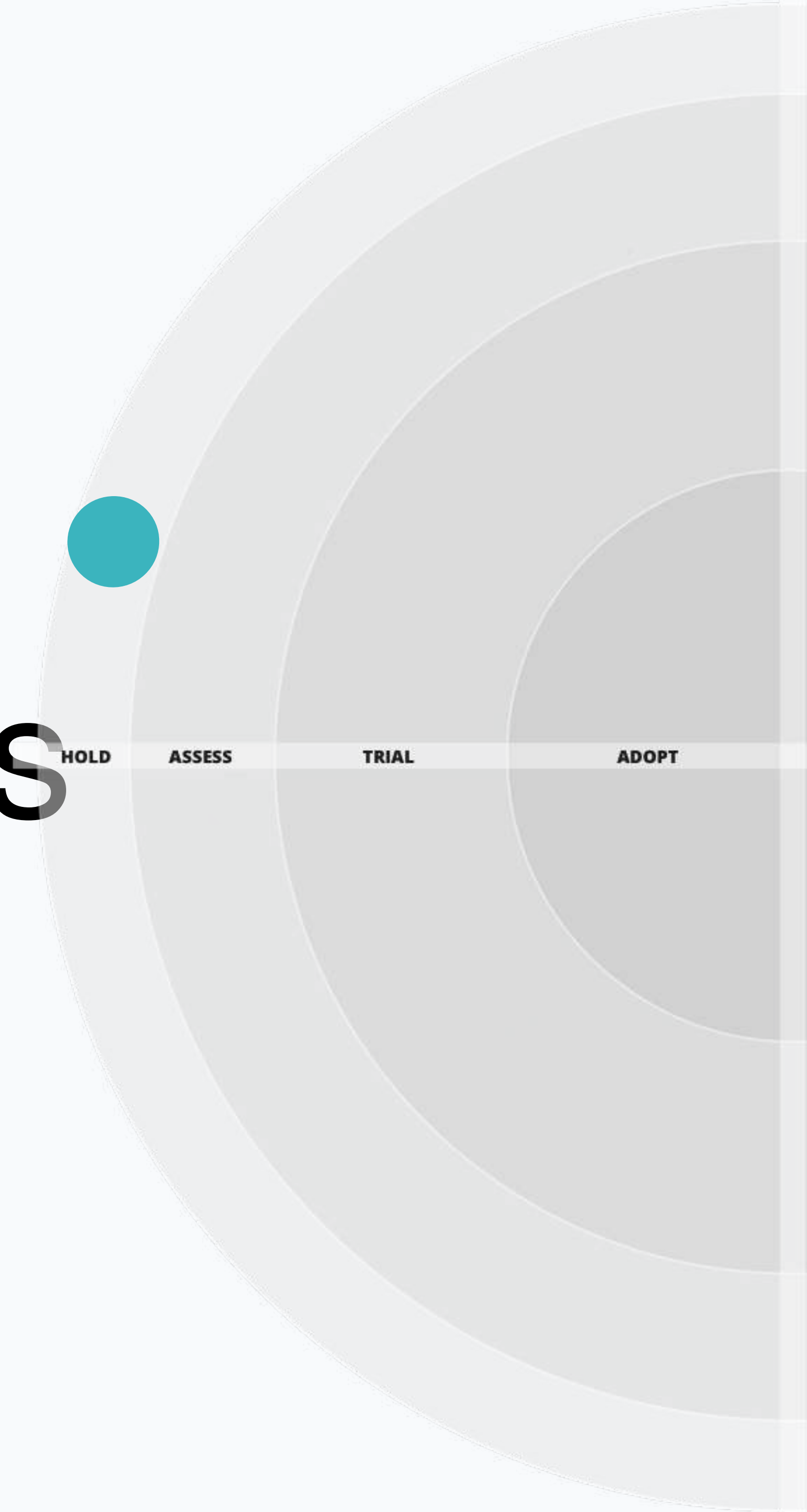
应对
产品制over项目制



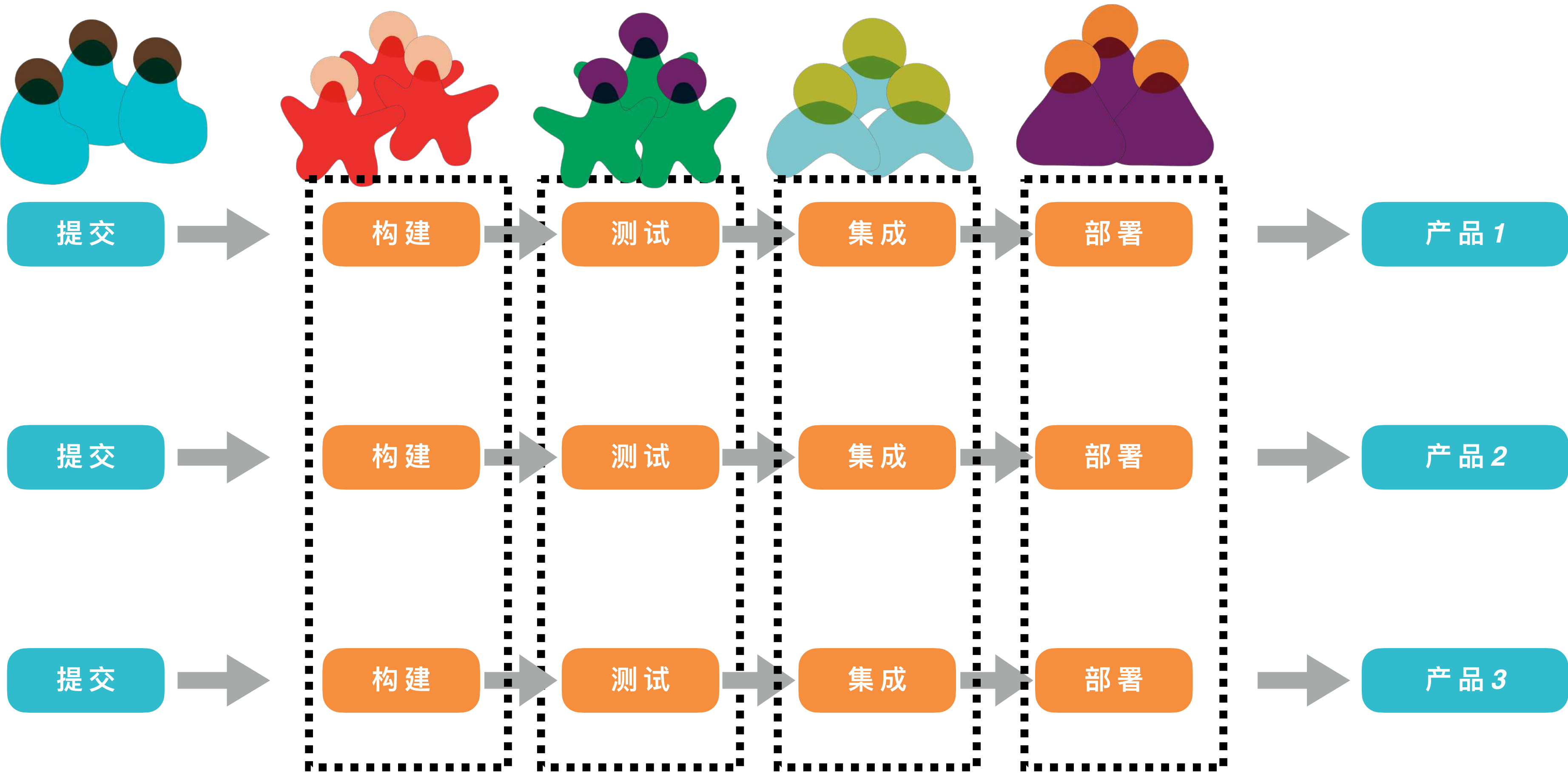
Techniques

Layered platform teams

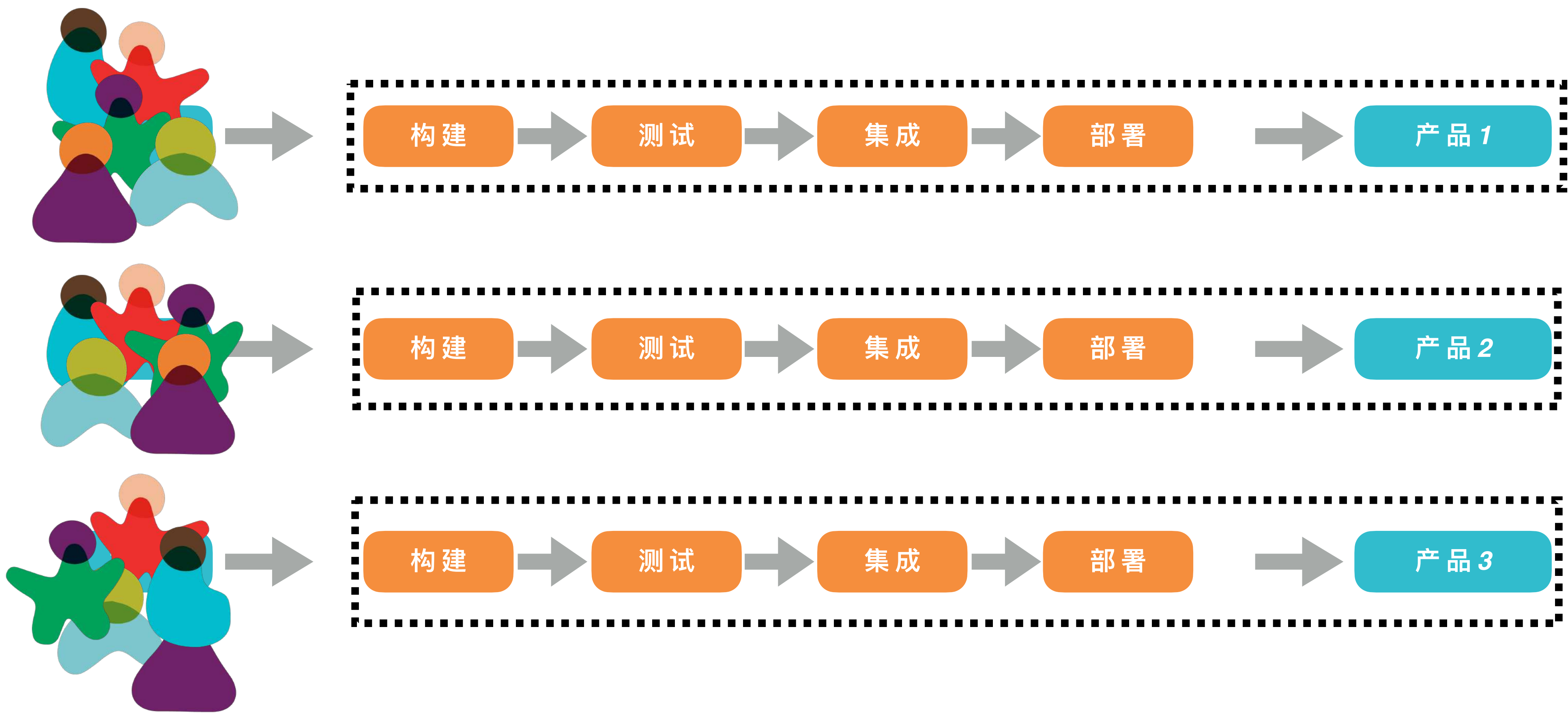
Adopt (2021)



应对
产品制 - 产品团队over分层团队



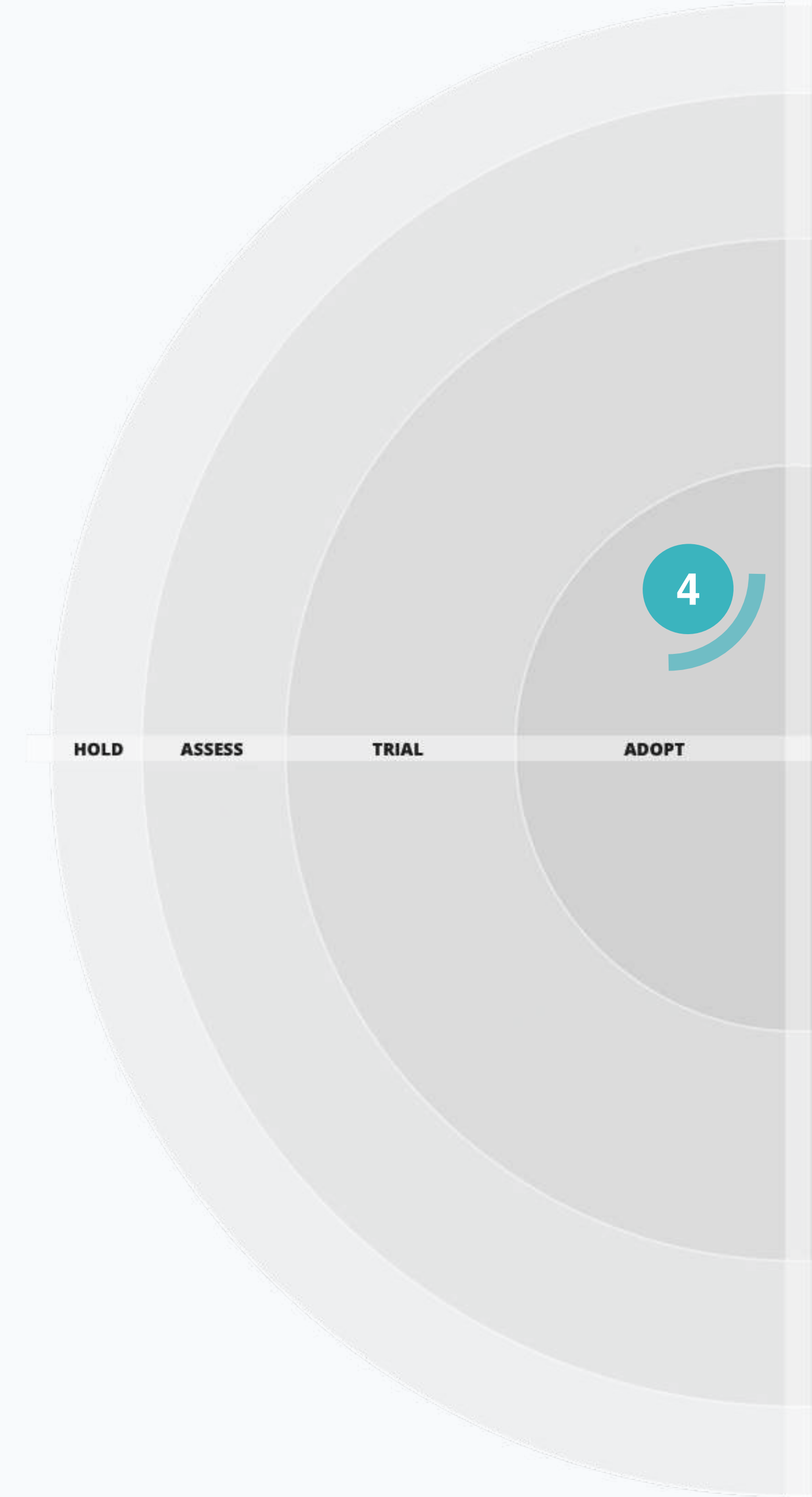
应对
产品制 - 产品团队over分层团队



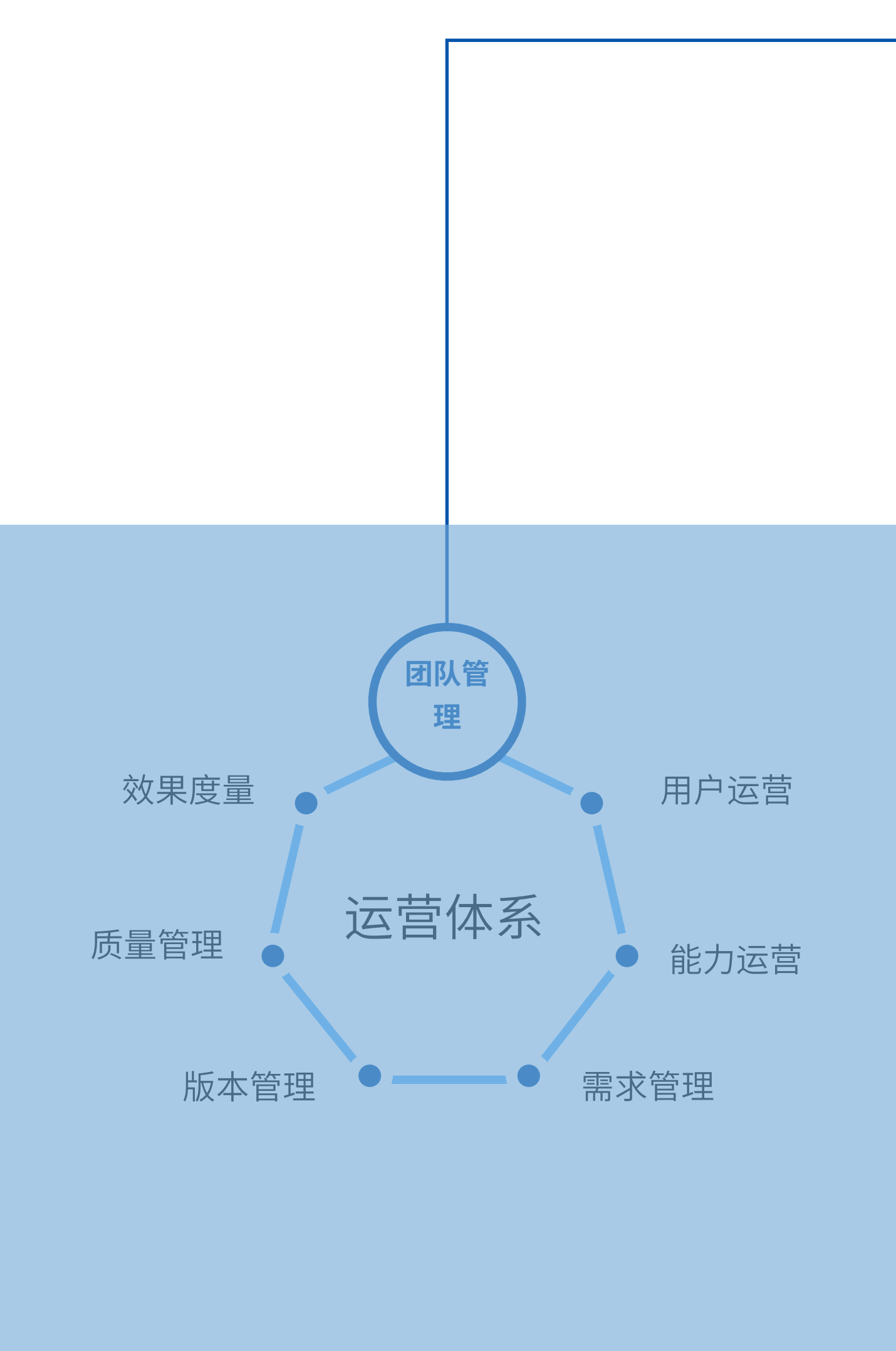
Techniques

Platform engineering product teams

Adopt (2021)

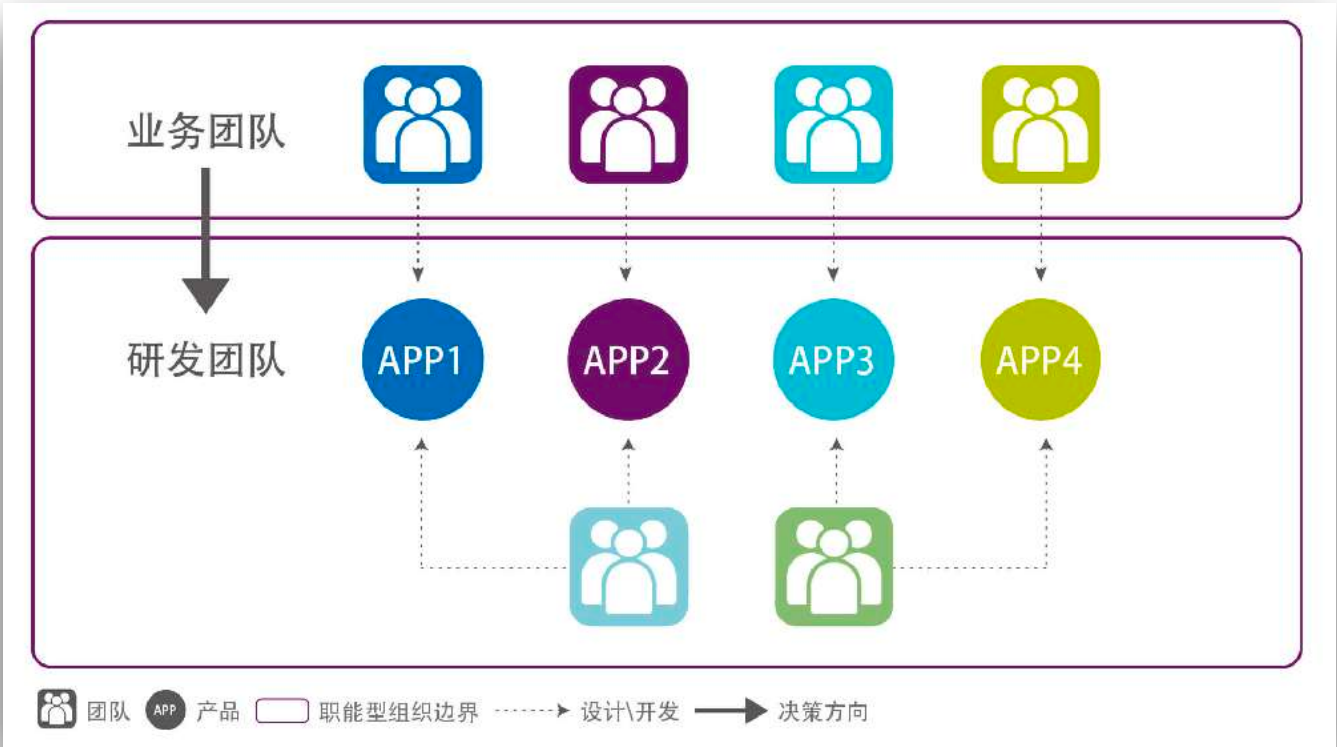


运营方案 - 团队如何有效组织与管理？ - 产品化团队

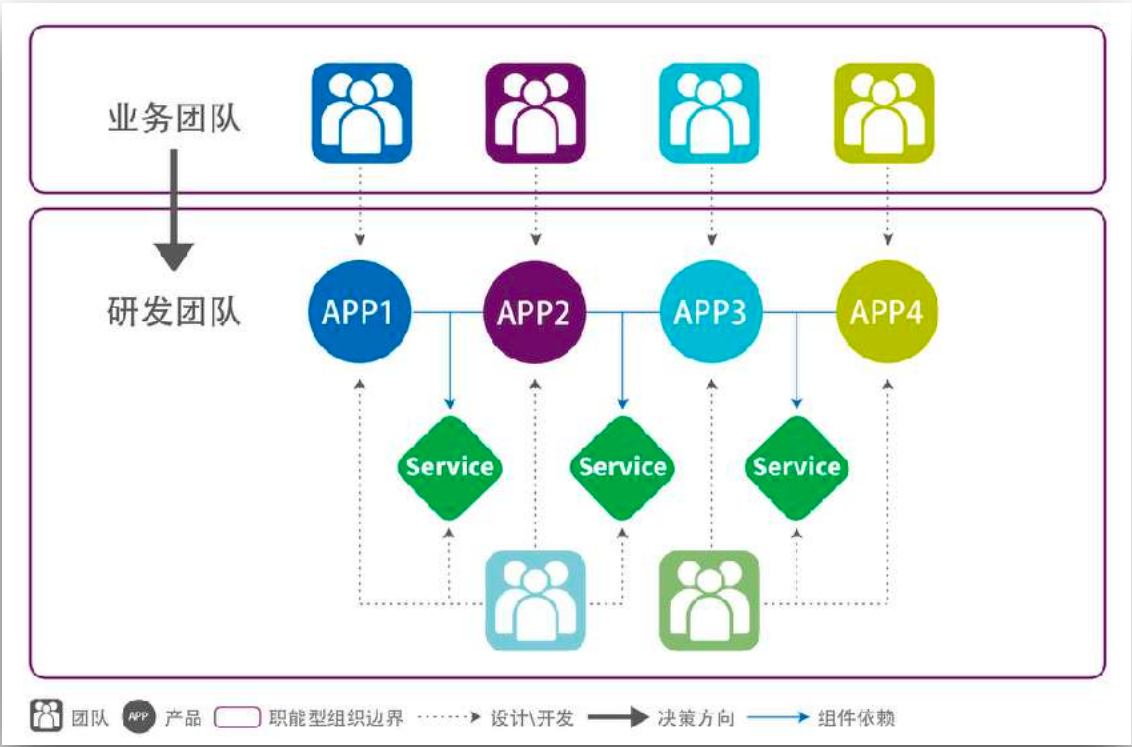


打造中台产品化团队

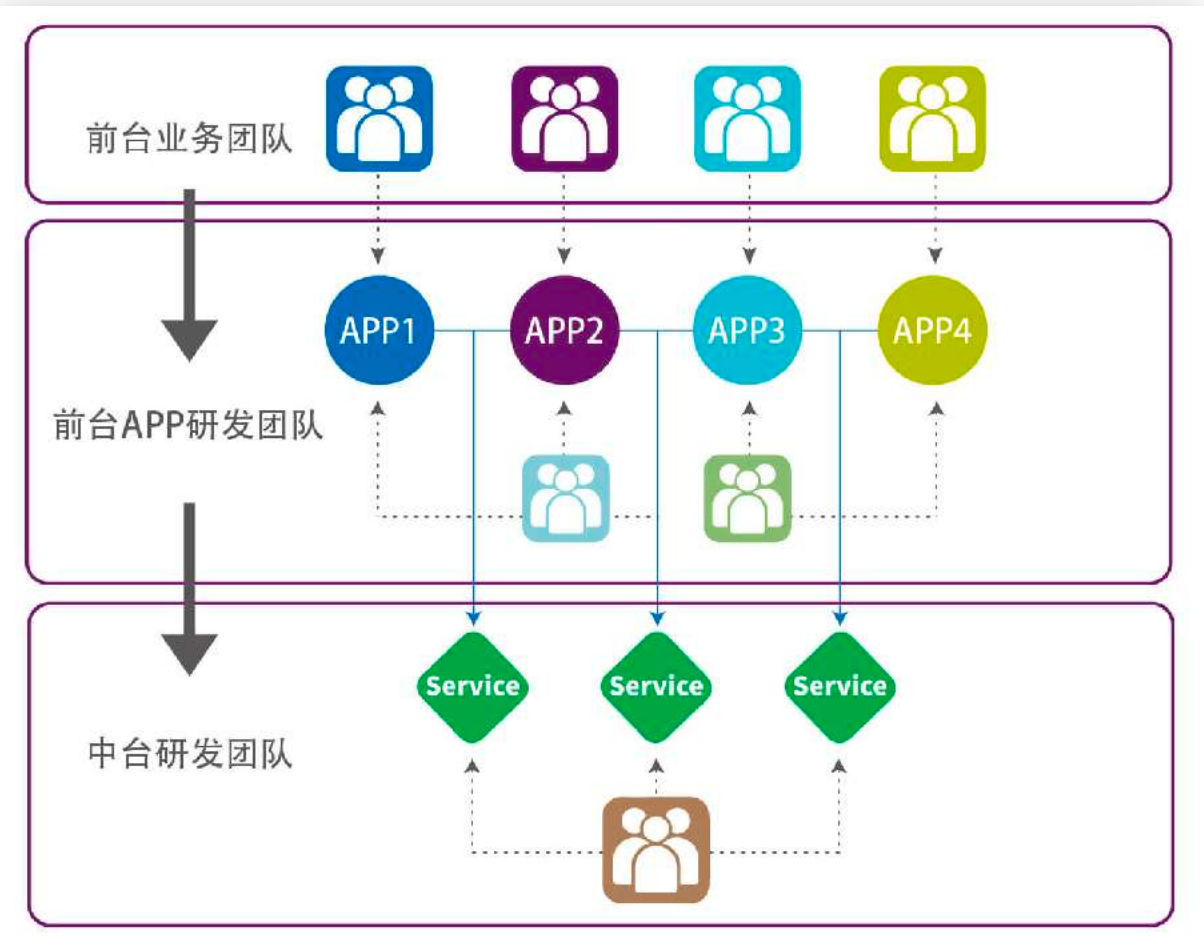
① 经典职能型组织



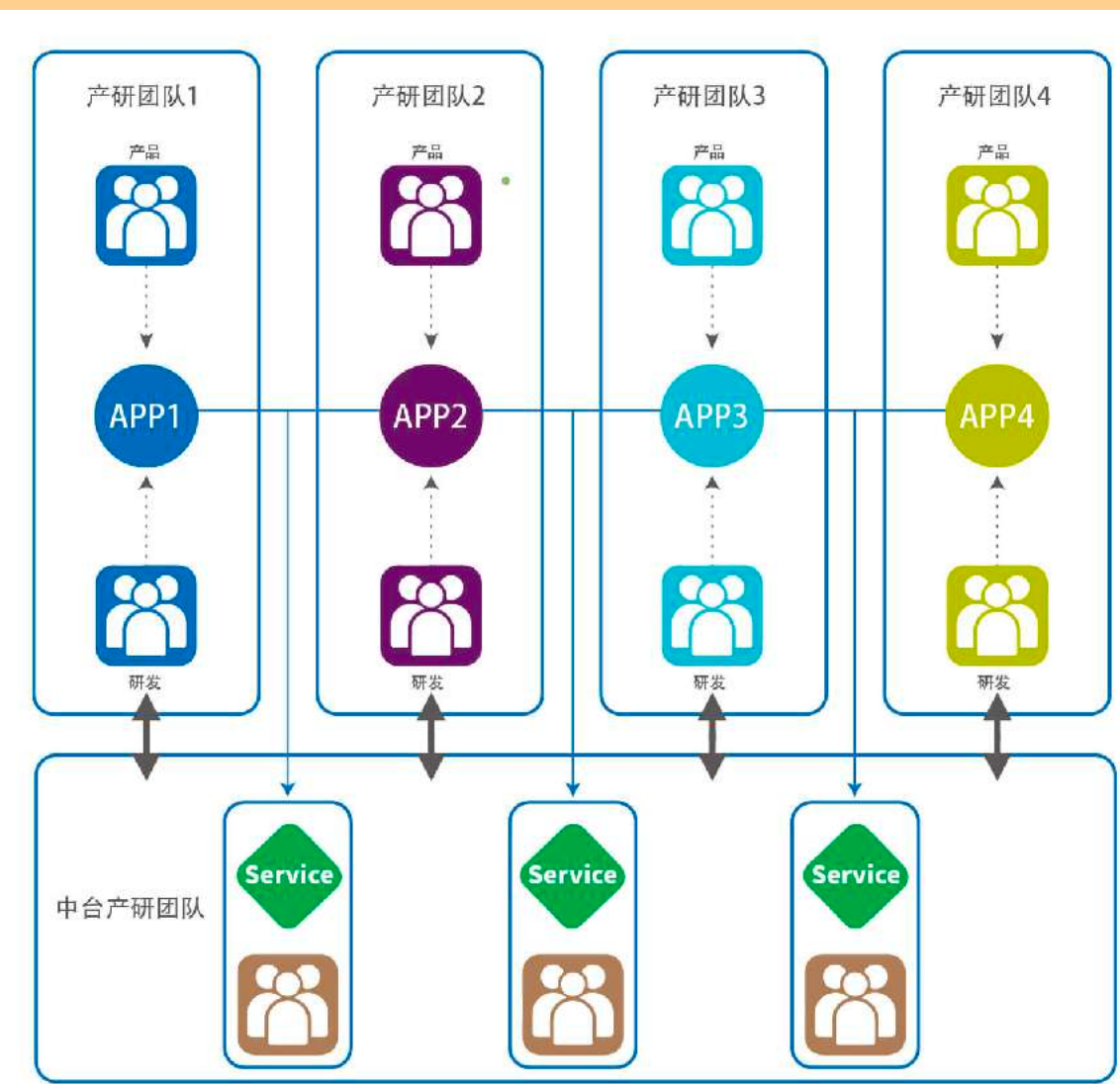
② 服务化



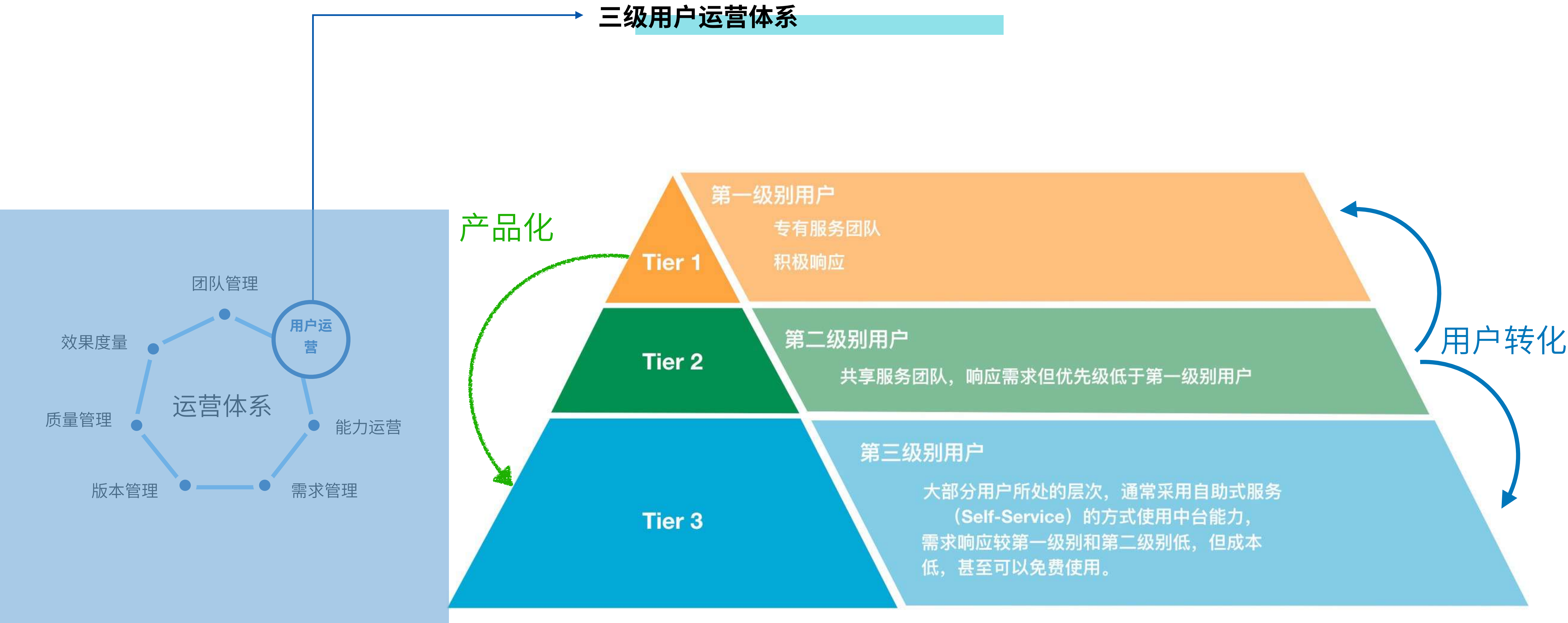
③ 分离前中台团队（内部外包模式）



④ 企业内部平台团队产品化运作



运营方案 - 能力如何有效运营推广？ - 三级客户运营推广体系



Techniques

API expand-contract

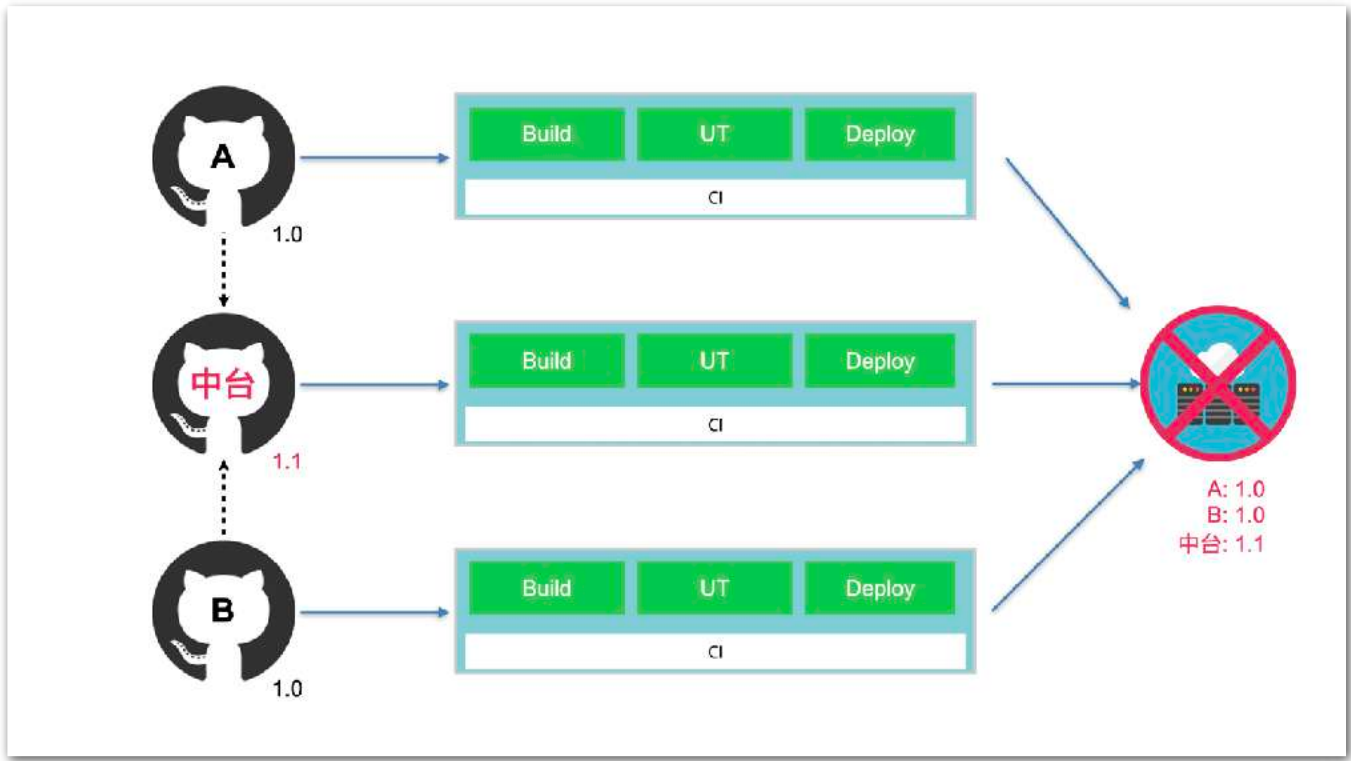
Adopt (2021)



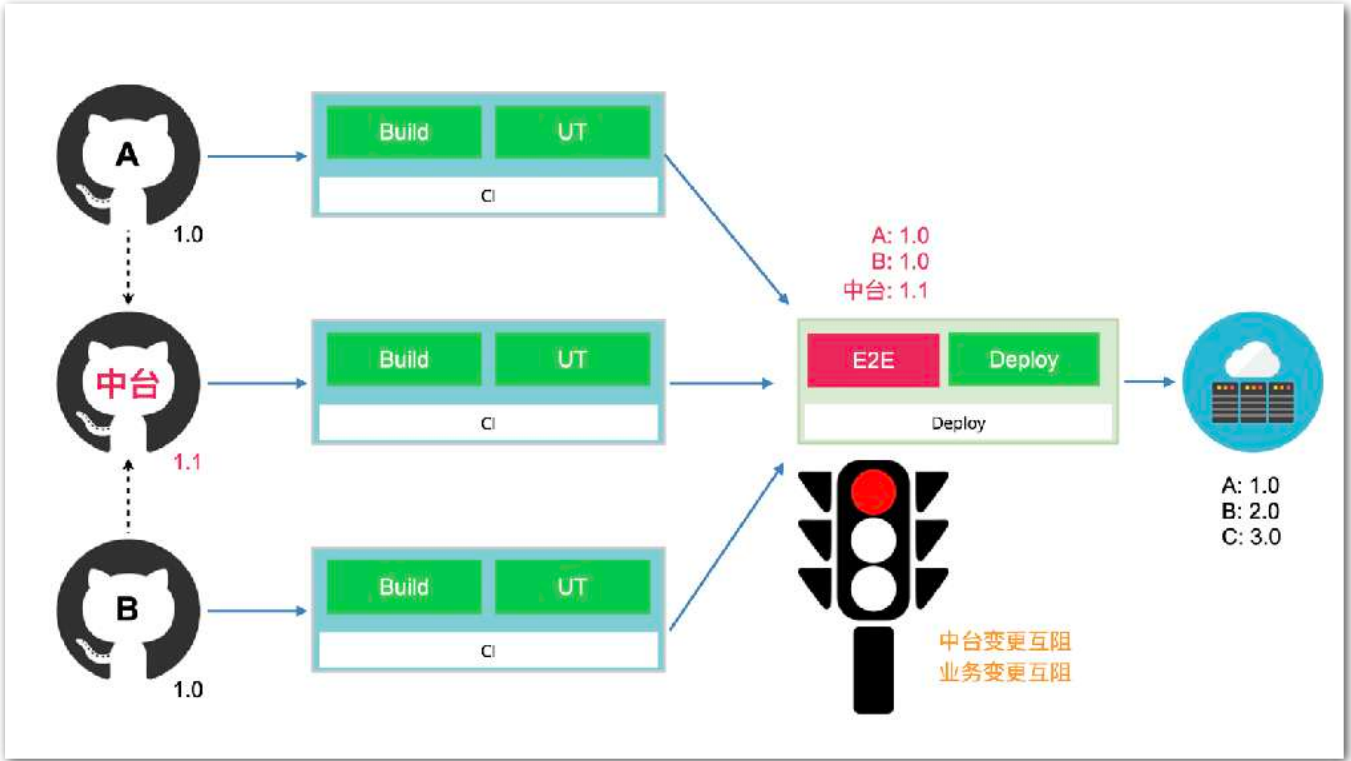
运营方案 - 版本策略与质量如何管理？ - 契约测试+适度版本机制

版本管理/版本策略/质量保障

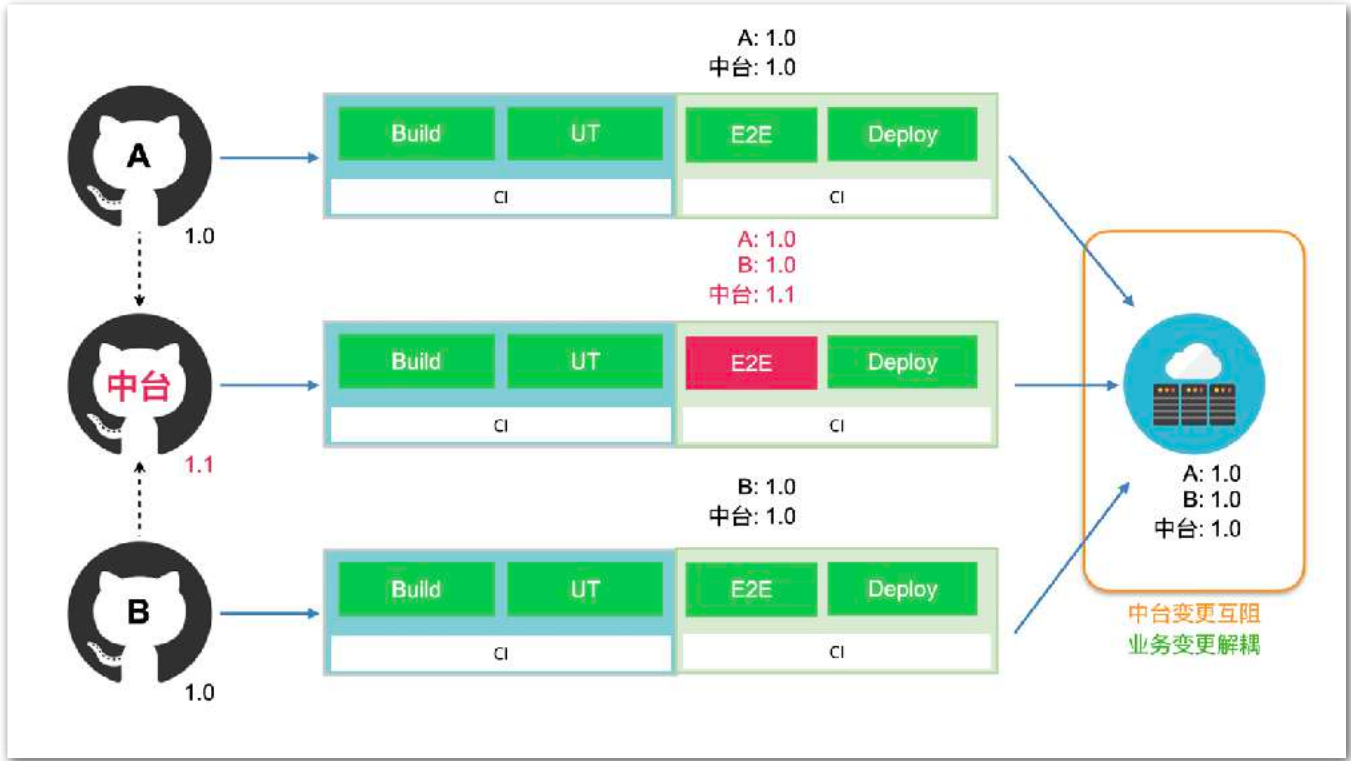
① 无测试（手动回归）易导致生产事故



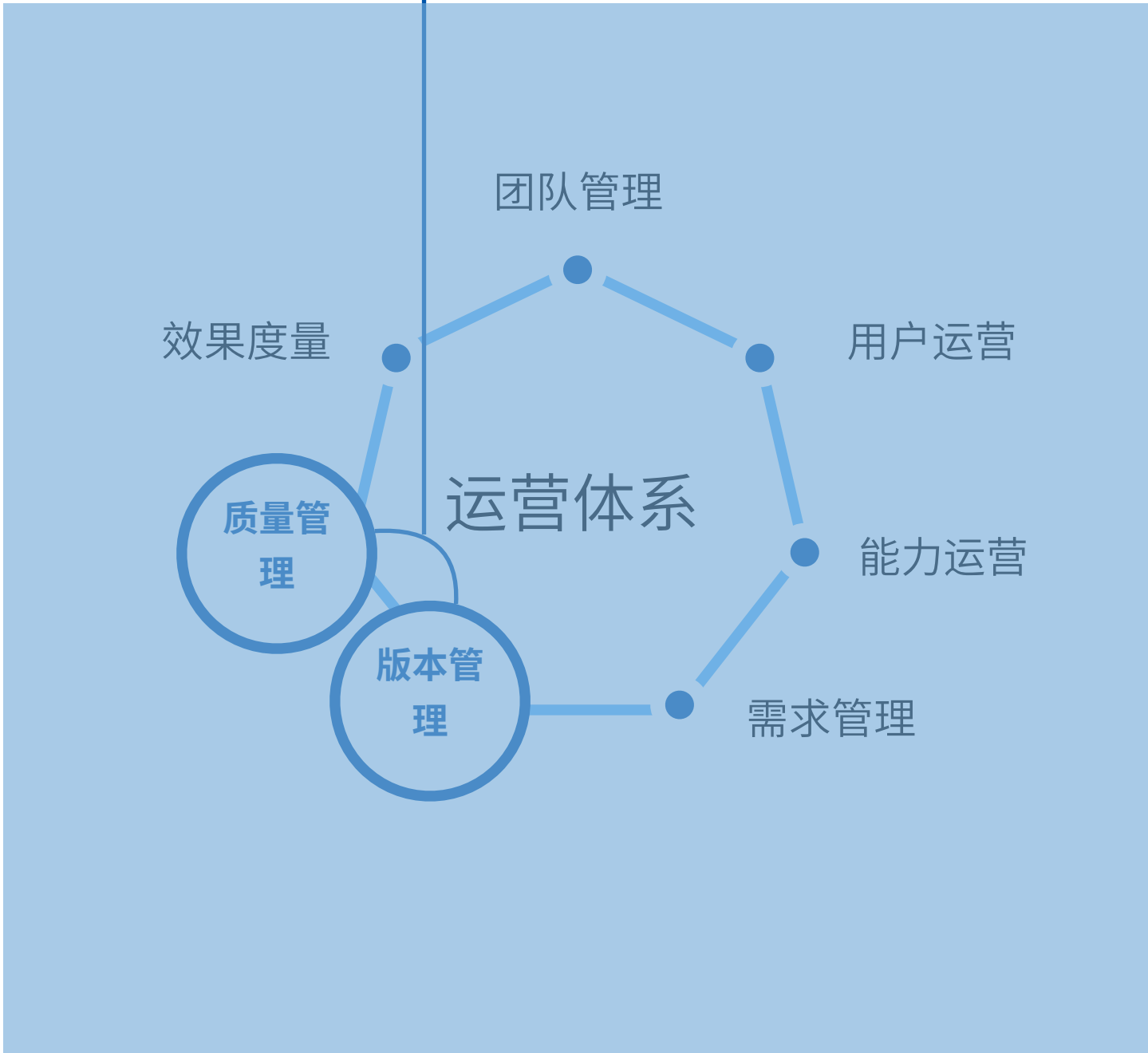
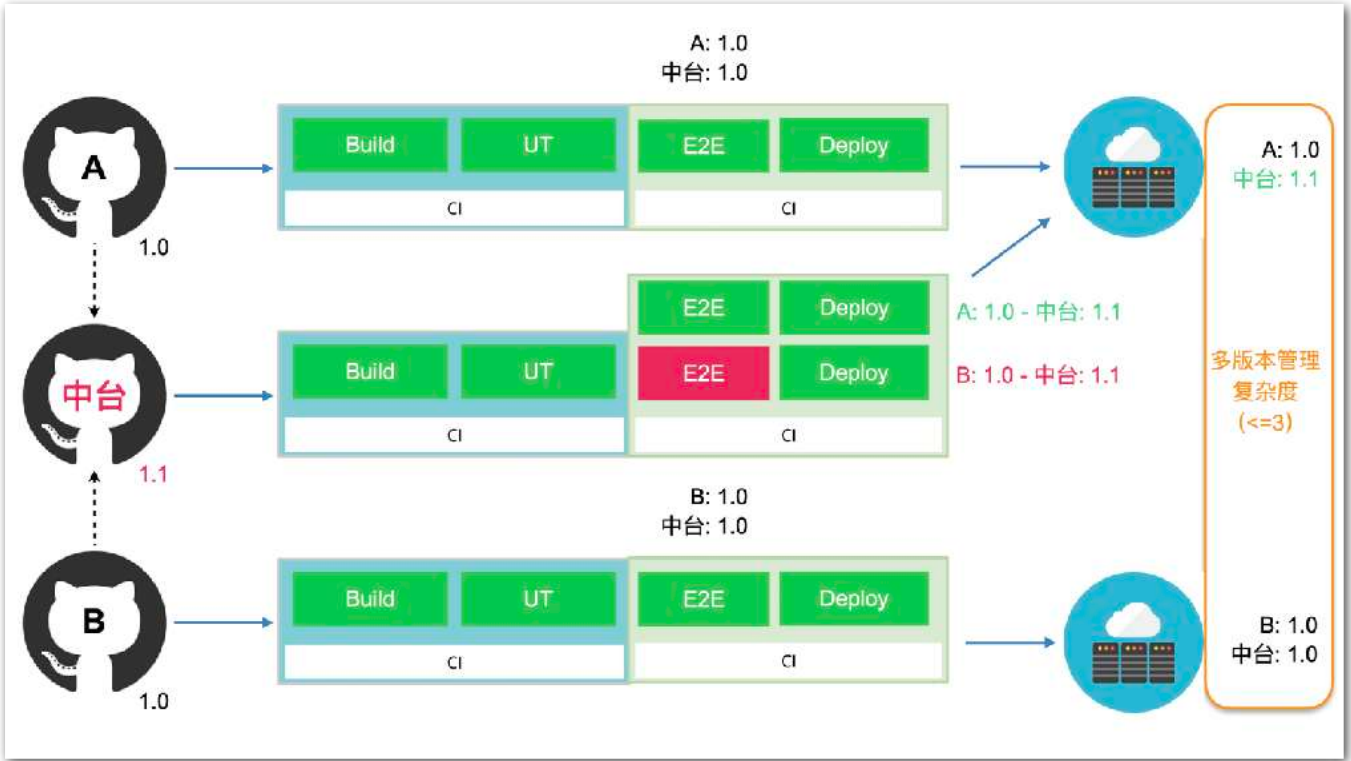
② 加入集成测试导致强耦合



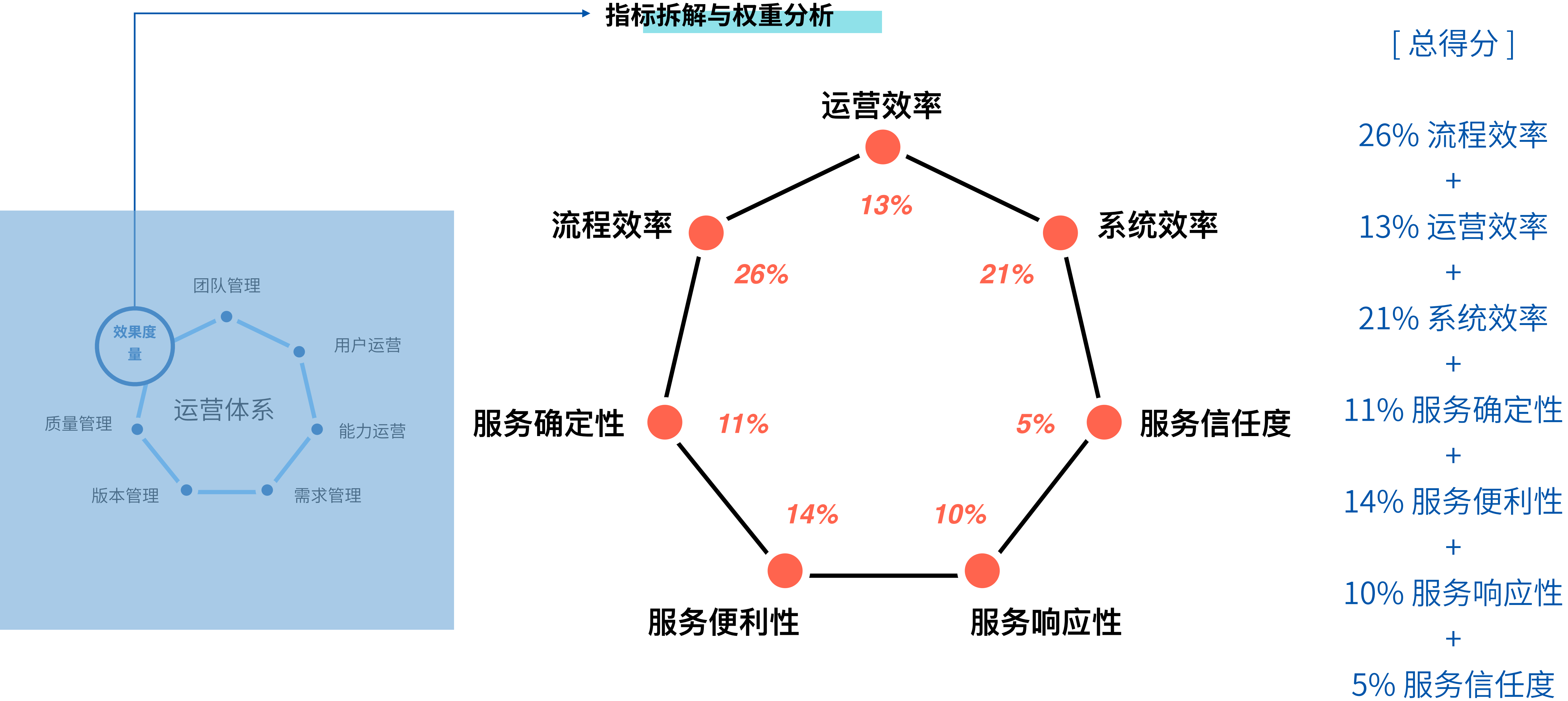
③ 引入契约测试解耦业务互阻



④ 引入多租户实现多版本管理



运营方案 - 效果如何评估与度量？ - 步骤2： 多维度指标拆解与权重分析



技术

采用

- 1. Consumer-driven contract testing
- 2. Decoupling deployment from release new
- 3. Generated infrastructure diagrams
- 4. NoPSD
- 5. Products over projects
- 6. Threat Modelling

试验

- 7. BEM new
- 8. BFF - Backend for frontends new
- 9. Docker for builds new
- 10. Event Storming new
- 11. Flux
- 12. Idempotency filter new
- 13. iFrames for sandboxing new
- 14. NPM for all the things new
- 15. Offline first web applications
- 16. Phoenix Environments
- 17. QA in production new

评估

- 18. Accumulate-only data
- 19. Bug bounties new
- 20. Data Lake
- 21. Hosted IDE's new
- 22. Monitoring of invariants new
- 23. Reactive Architectures

暂缓

- 24. Gitflow new
- 25. High performance envy/web scale envy new
- 26. Microservice envy
- 27. Pace-layered Application Strategy
- 28. Programming in your CI/CD tool
- 29. SAFe™
- 30. Separate DevOps team

平台

采用

- 31. TOTP Two-Factor Authentication

试验

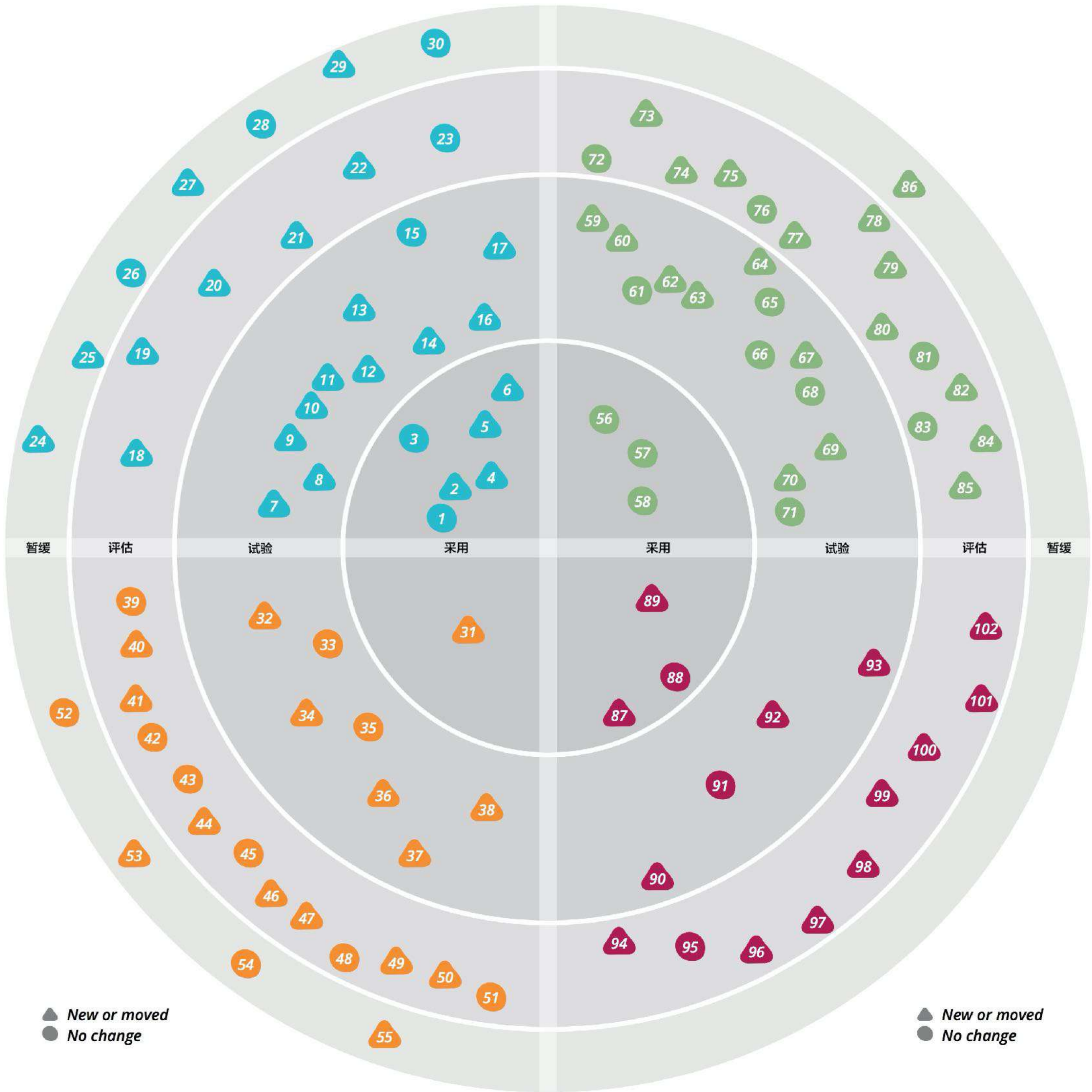
- 32. Apache Mesos
- 33. Apache Spark
- 34. AWS Lambda new
- 35. Cloudera Impala
- 36. Fastly new
- 37. H2O
- 38. HSTS new

评估

- 39. Apache Kylin
- 40. AWS ECS new
- 41. Ceph new
- 42. CoreCLR and CoreFX
- 43. Deis
- 44. Kubernetes new
- 45. Linux security modules
- 46. Mesosphere DCOS new
- 47. Microsoft Nano Server new
- 48. Particle Photon/Particle Electron
- 49. Presto new
- 50. Rancher new
- 51. Time series databases

暂缓

- 52. Application Servers
- 53. Over-ambitious API Gateways new
- 54. SPDY
- 55. Superficial private cloud new



工具

采用

- 56. Composer
- 57. Mountebank
- 58. Postman

试验

- 59. Browsersync new
- 60. Carthage new
- 61. Consul
- 62. Docker Toolbox new
- 63. Gitrob new
- 64. GitUp new
- 65. Hamms
- 66. IndexedDB
- 67. Polly
- 68. REST-assured
- 69. Sensu
- 70. SysDig new
- 71. ZAP

评估

- 72. Apache Kafka
- 73. Concourse CI new
- 74. Espresso new
- 75. Gauge new
- 76. Gor
- 77. ievms new
- 78. Let's Encrypt new
- 79. Pageify new
- 80. Prometheus
- 81. Quick
- 82. RAML new
- 83. Security Monkey
- 84. Sleepy Puppy new
- 85. Visual Studio Code new

暂缓

- 86. Citrix for development

语言和框架

采用

- 87. ECMAScript 6 new
- 88. Nancy
- 89. Swift

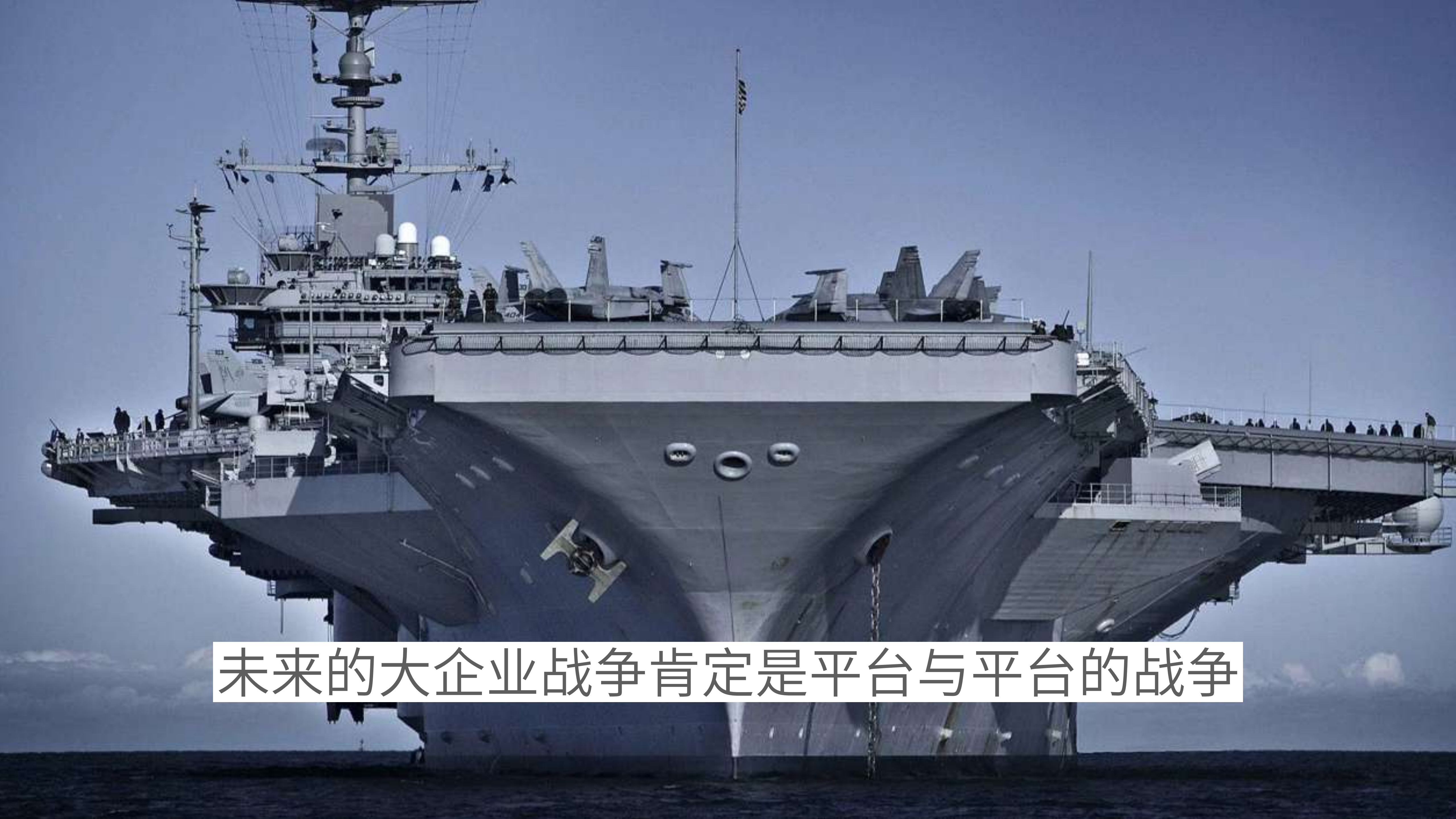
试验

- 90. Enlive new
- 91. React.js
- 92. SignalR new
- 93. Spring Boot

评估

- 94. Axon new
- 95. Ember.js
- 96. Frege new
- 97. HyperResource new
- 98. Material UI new
- 99. OkHttp new
- 100. React Native new
- 101. TLA+ new
- 102. Traveling Ruby new

暂缓



未来的大企业战争肯定是平台与平台的战争



<https://www.thoughtworks.com/radar>

THANK YOU

欲了解详情，请发送邮件至：
jianwang@thoughtworks.com

ThoughtWorks®