

技术雷达之 平台崛起

徐昊 刘传湘 熊节
ThoughtWorks

下一代
软件开发
SOFTWARE
DEVELOPMENT

The logo graphic consists of three overlapping, semi-transparent, olive-green circular shapes that form a stylized, abstract shape resembling a radar or a signal. The shapes are positioned in the upper left quadrant of the page.

ThoughtWorks®

TECHNOLOGY
RADAR *VOL. 16*

26th Apr, 2017

5 NEW THEMES

CONVERSATIONAL UI AND NATURAL LANGUAGE PROCESSING

INTELLIGENCE AS A SERVICE

**DEVELOPER EXPERIENCE AS THE
NEW DIFFERENTIATOR**

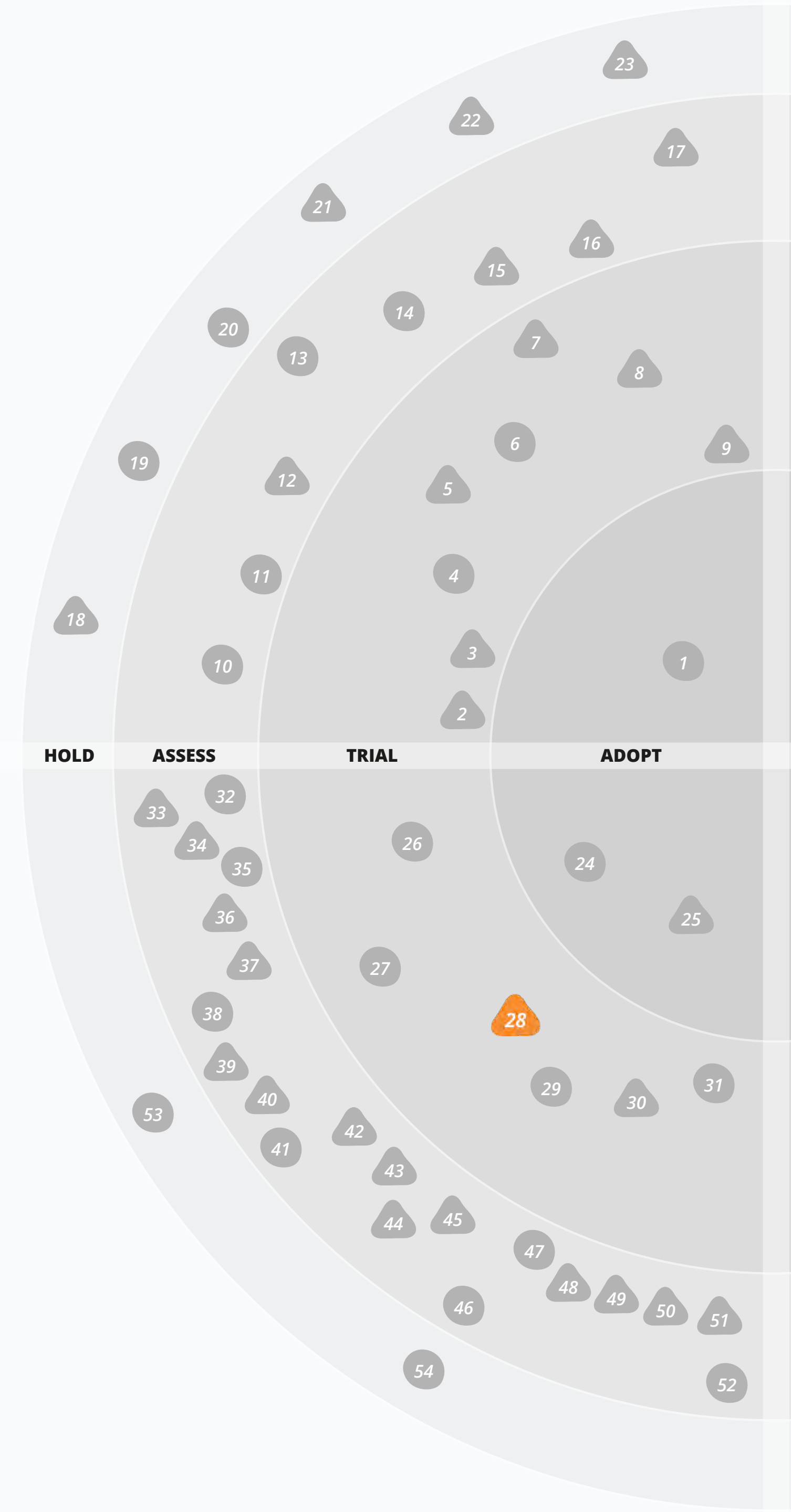
PERVASIVE PYTHON

THE RISE OF PLATFORMS

PLATFORMS

AWS Device Farm

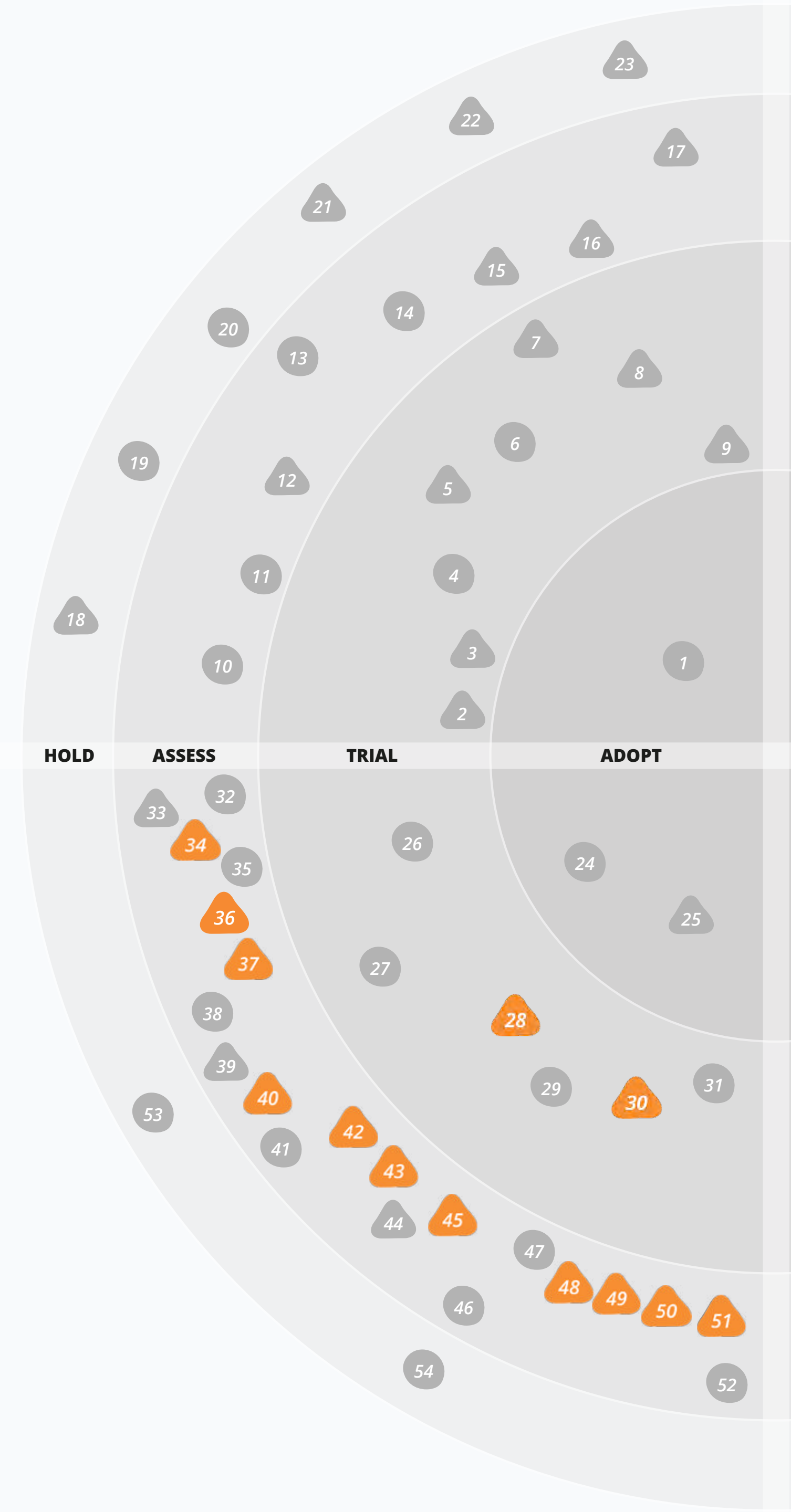
Trial

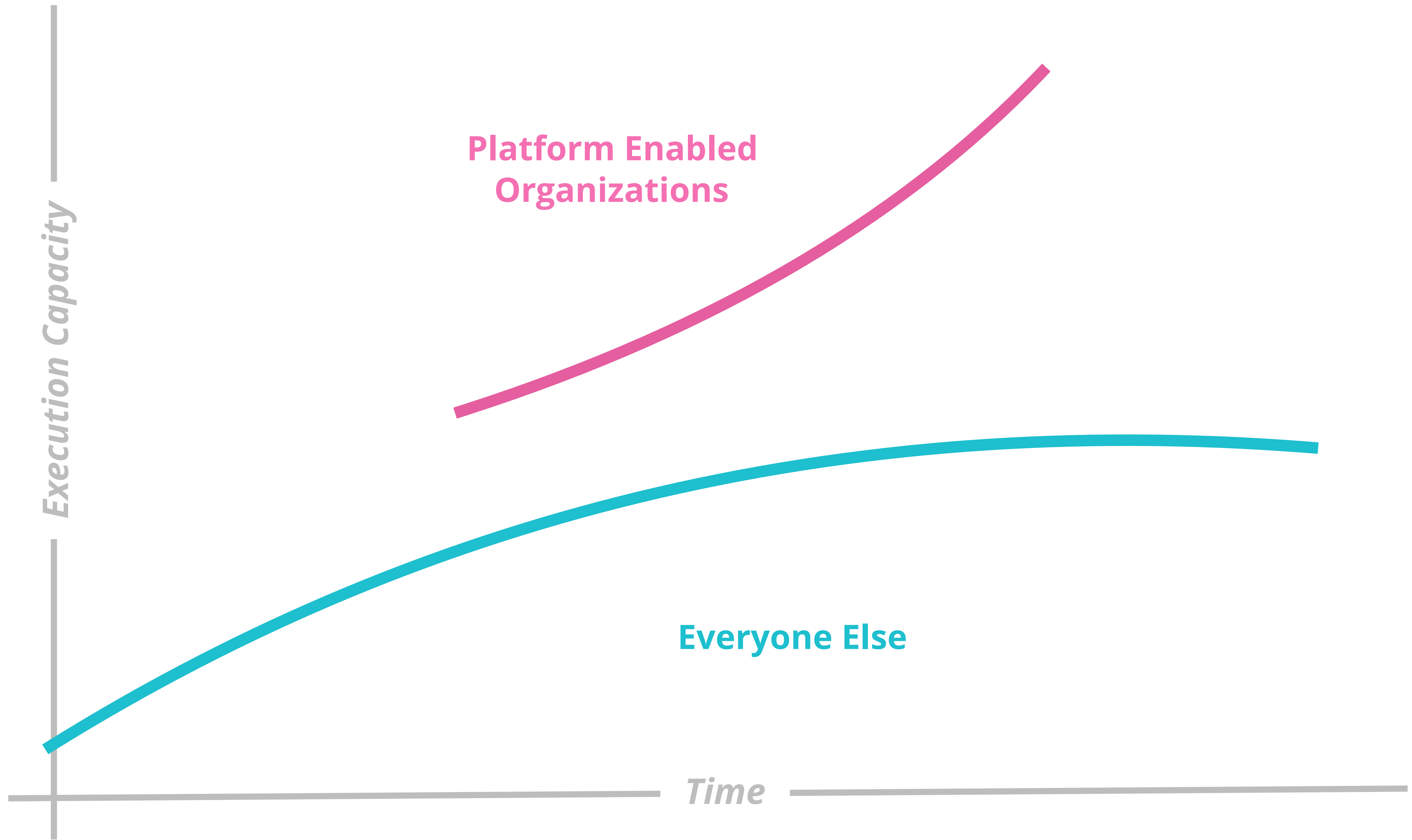


PLATFORMS

AWS Device Farm

Trial





```
var ApiBuilder = require('claudia-api-builder'),
    api = new ApiBuilder();

module.exports = api;

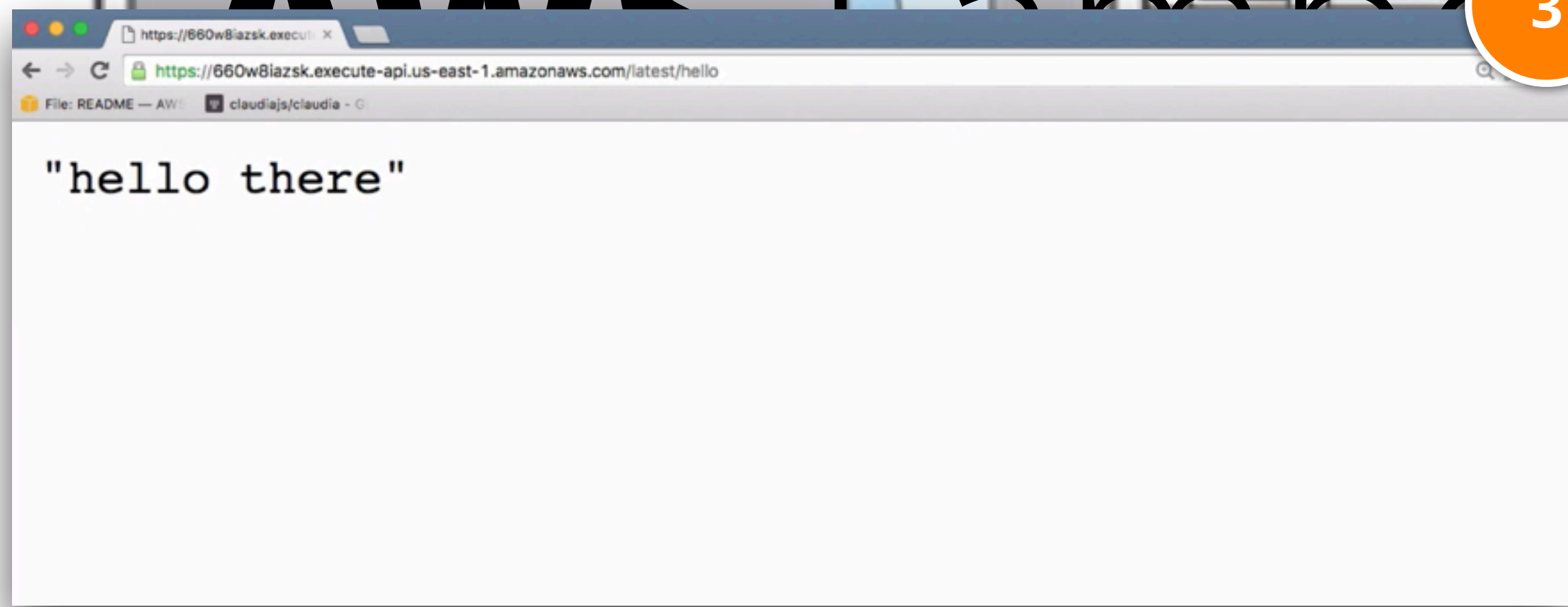
api.get('/hello', function () {
  return 'hello there';
});
```

1

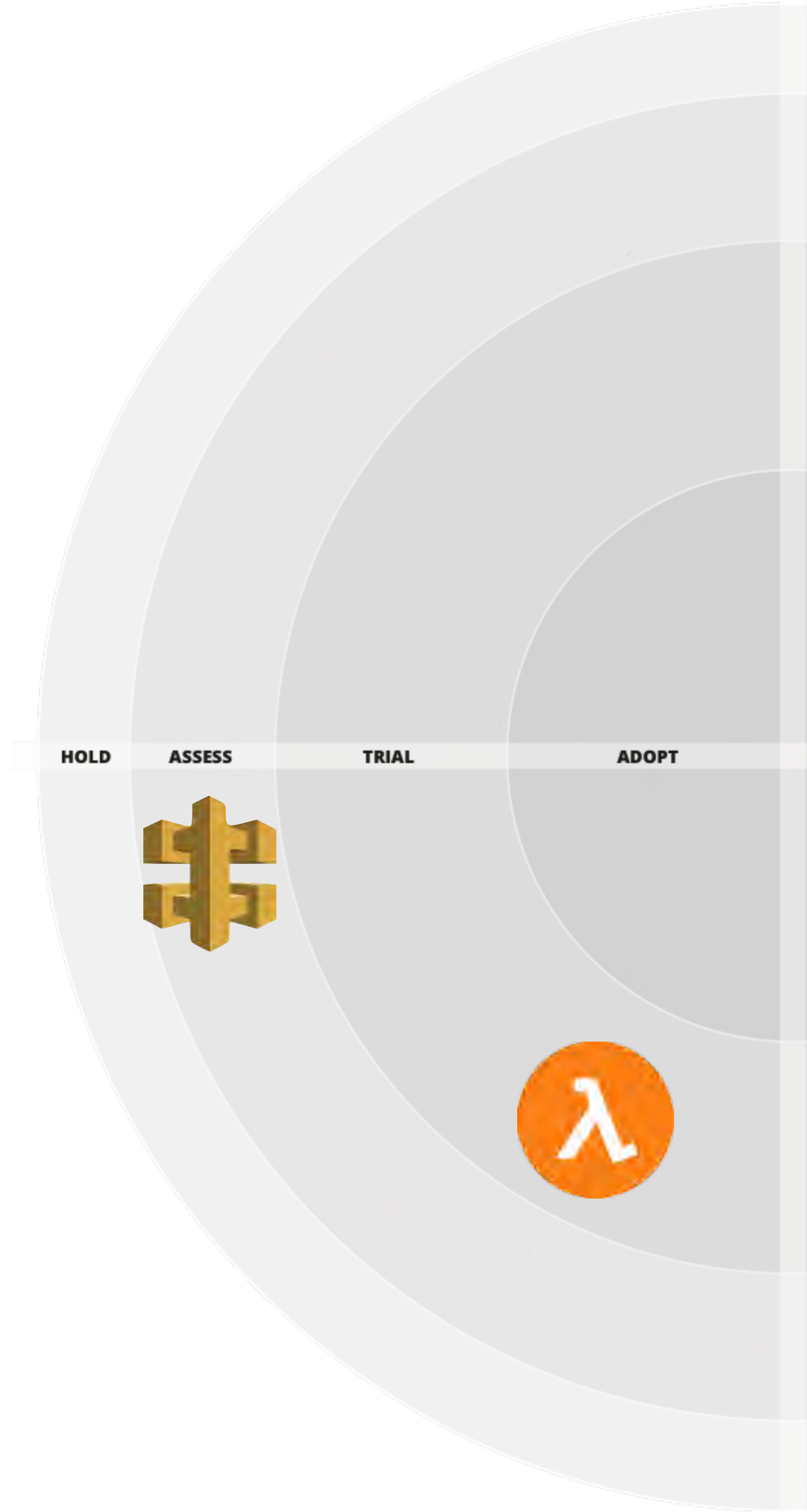


```
$ claudia create --region us-east-1 --api-module server
```

2



3



AWS Device Farm

Improve the quality of your iOS, Android, and web applications by testing against real mobile devices in the AWS Cloud

Start Testing Today

*Your first 1000 minutes are free**

Device List

Pricing

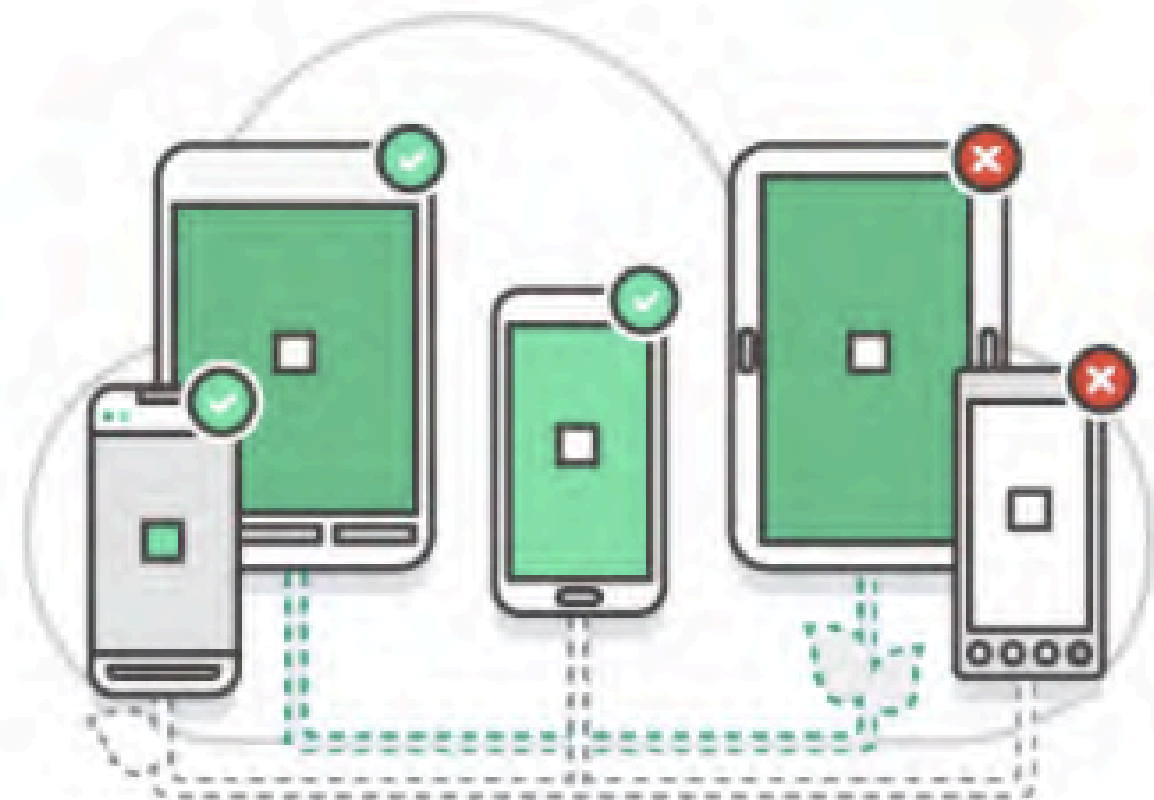
FAQs

Documentation

Blog

Plugins and Samples

AWS Device Farm is an app testing service that lets you test and interact with your Android, iOS, and web apps on many devices at once, or reproduce issues on a device in real time. View video, screenshots, logs, and performance data to pinpoint and fix issues before shipping your app.



Automated Testing

Test your app in parallel against a large collection of physical devices in the AWS Cloud



Remote Access

Gesture, swipe, and interact with devices in real time, directly from your web browser

HOLD

ASSESS

TRIAL

ADOPT



Wit.ai

Powered by Wit.ai 
Read our [blog post](#) explaining what Wit does

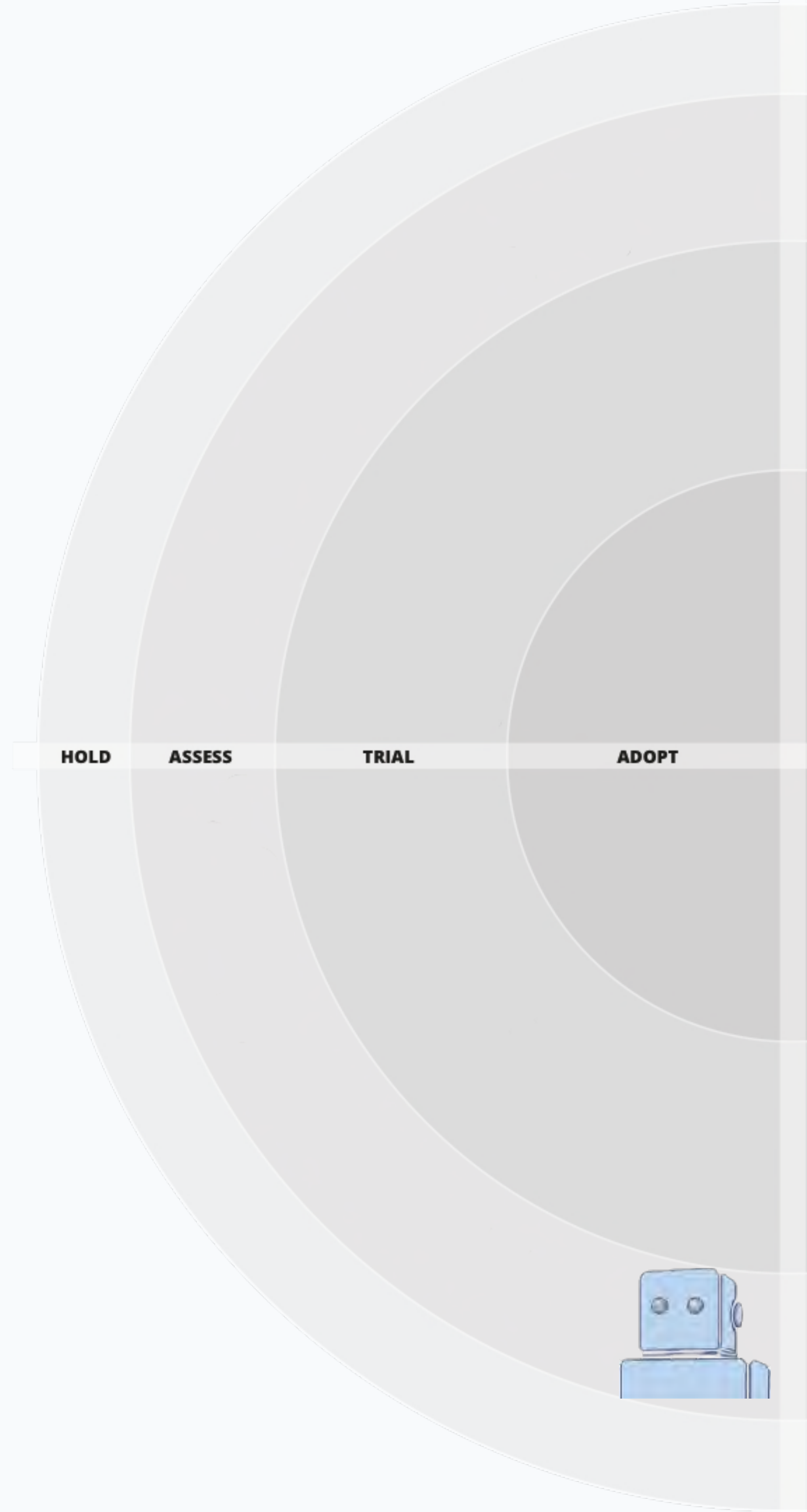


Click here to start (or press Space)

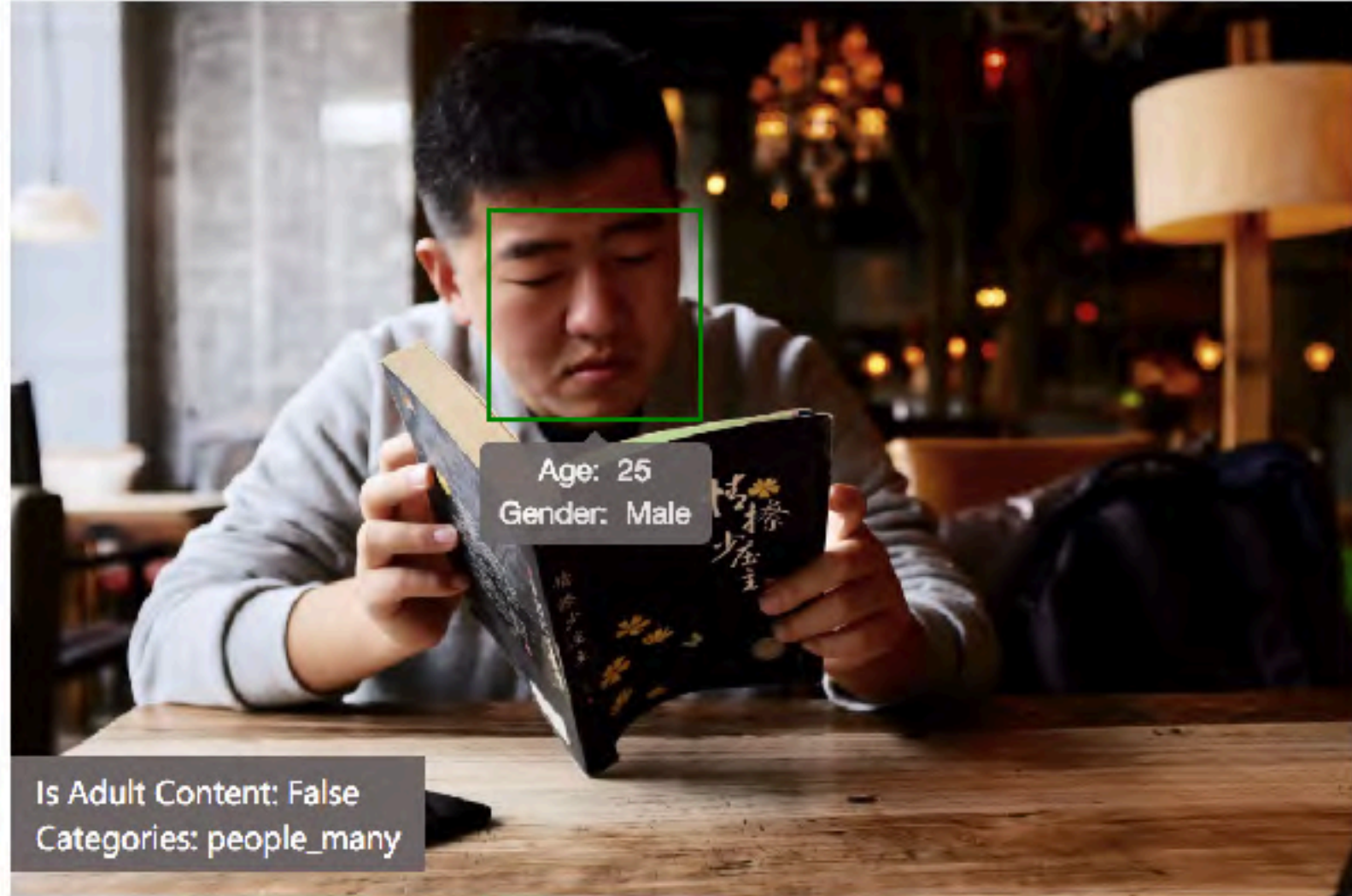
text = open the door
intent = doors
door_action = open

Did you try...

- o What time is it in Paris?
- o Set the temperature to 68 degrees
- o Tell me the weather in Las Vegas
- o Turn off the lights in the kitchen



Cloud-based image comprehension



Features:

Feature Name	Value
Description	{ "type": 0, "captions": [{ "text": "a man sitting at a wooden table using a laptop", "confidence": 0.1137671054172018 }] }
Tags	[{ "name": "person", "confidence": 0.9944087862968445 }, { "name": "table", "confidence": 0.9908931255340576 }, { "name": "indoor", "confidence": 0.9115577340126038 }, { "name": "dish", "confidence": 0.4516720175743103, "hint": "food" }]
Image Format	Jpeg
Image Dimensions	4896 x 3264
Clip Art Type	0 Non-clipart
Line Drawing Type	0 Non-LineDrawing



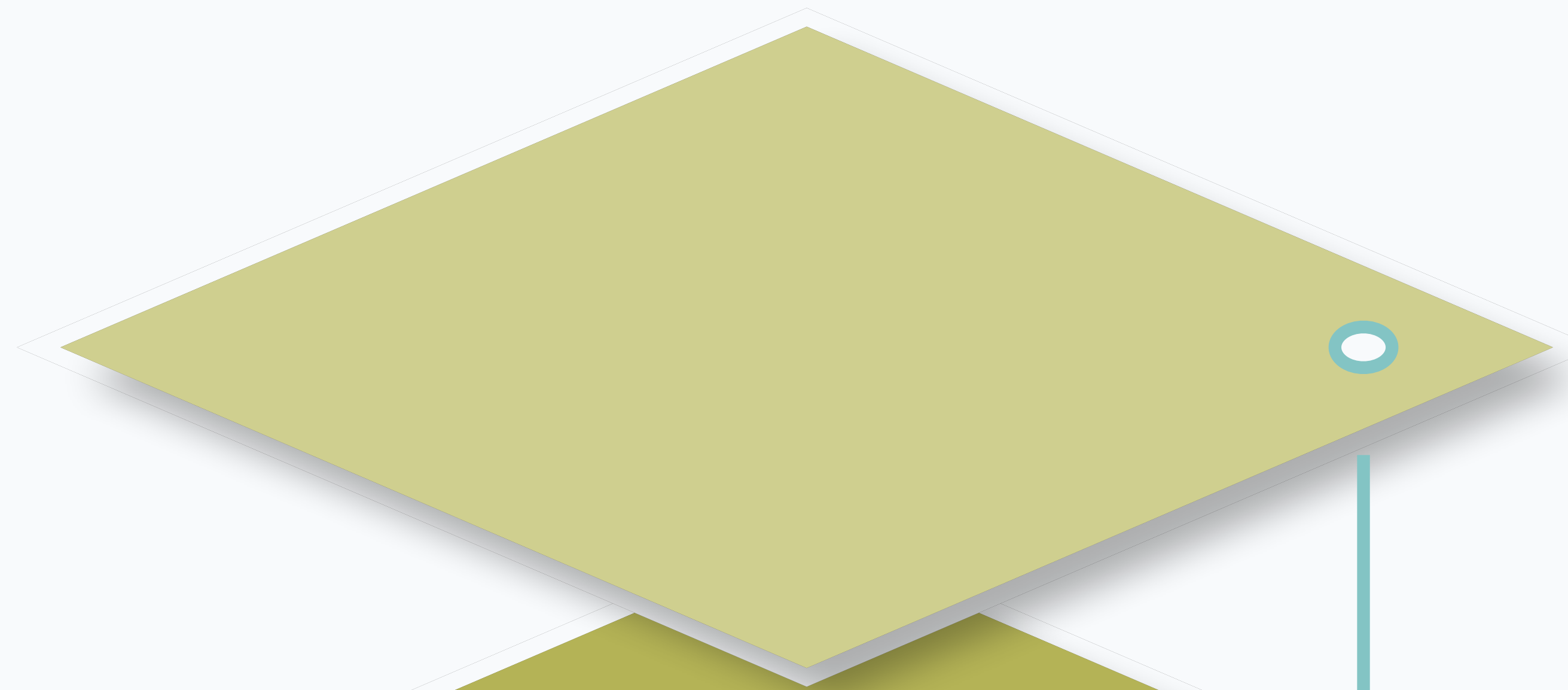
Features:

Feature Name	Value
Description	{ "type": 0, "captions": [{ "text": "a man sitting in front of a window with a laptop", "confidence": 0.13323378454508847 }] }
Tags	[{ "name": "indoor", "confidence": 0.9392566084861755 }, { "name": "person", "confidence": 0.8903388381004333 }, { "name": "looking", "confidence": 0.8601085543632507 }, { "name": "window", "confidence": 0.8437134027481079 }, { "name": "screen", "confidence": 0.7059780955314636 }, { "name": "watching", "confidence": 0.509236216545105 }, { "name": "image", "confidence": 0.3080187141895294 }, { "name": "staring", "confidence": 0.18520550429821014 }]
Image Format	Jpeg
Image Dimensions	1584 x 1018

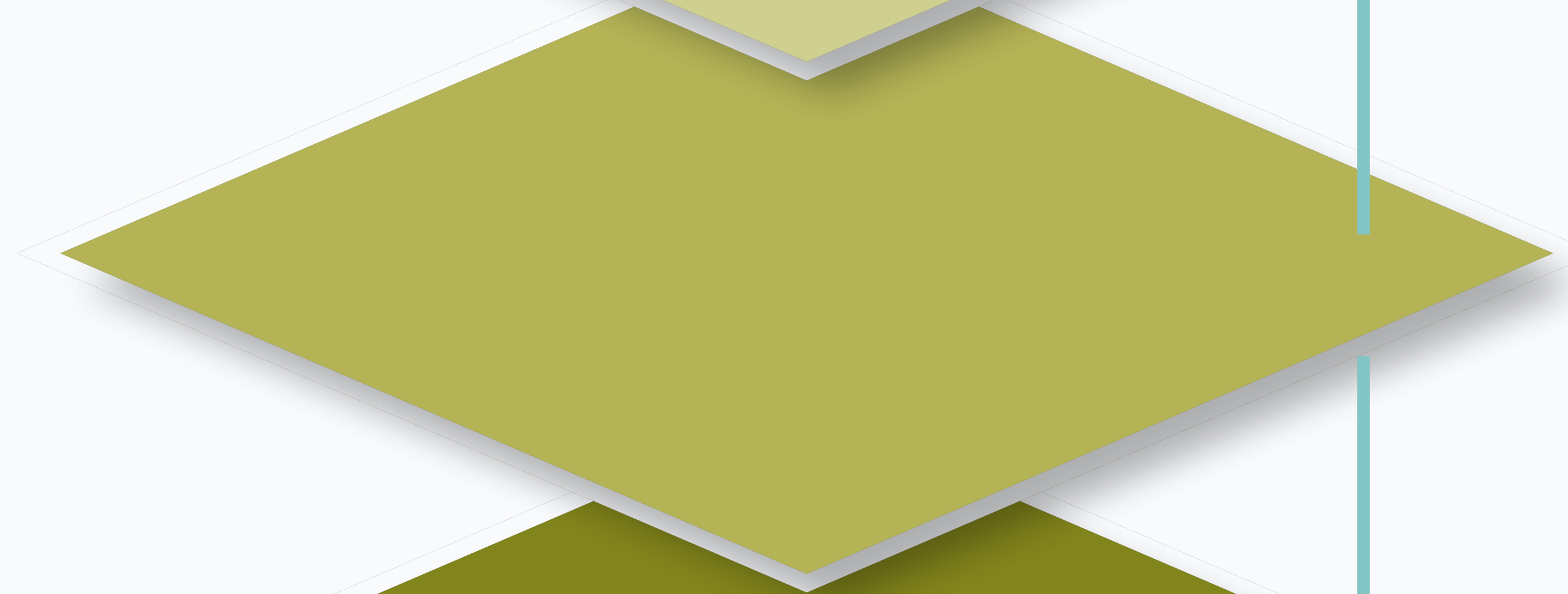
HOLD ASSESS TRIAL ADOPT



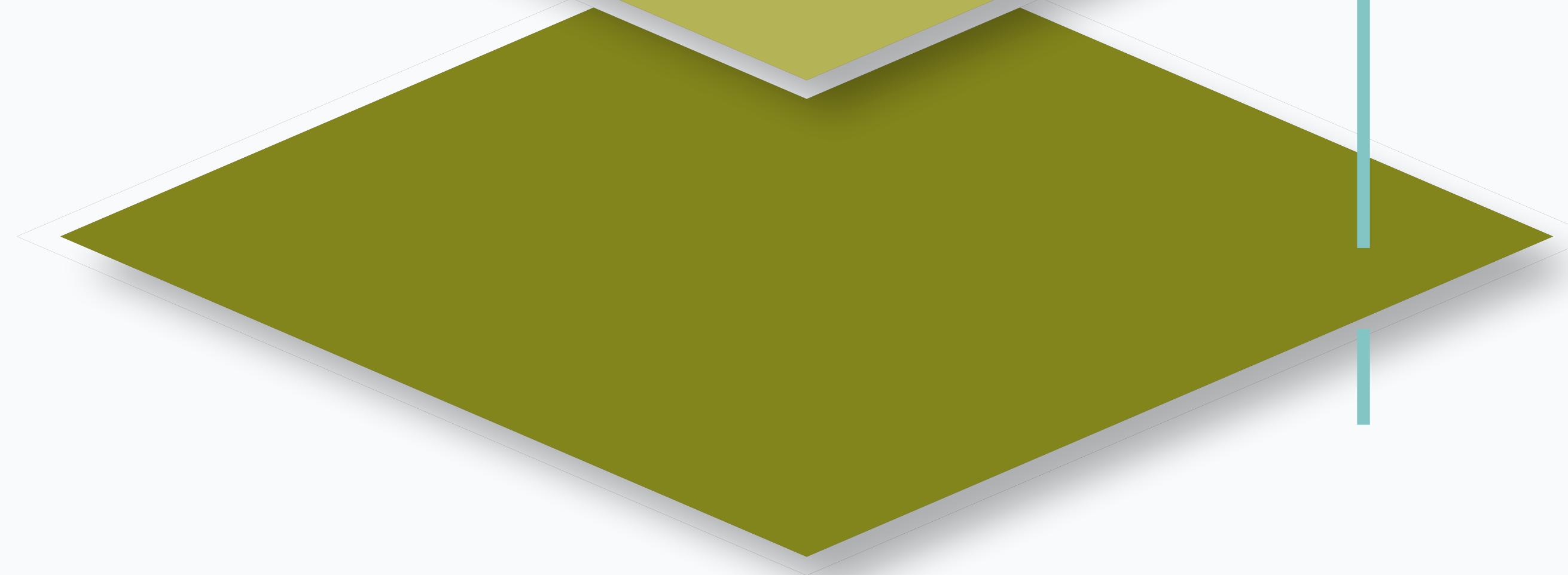
WHY NOW?



Business Platform



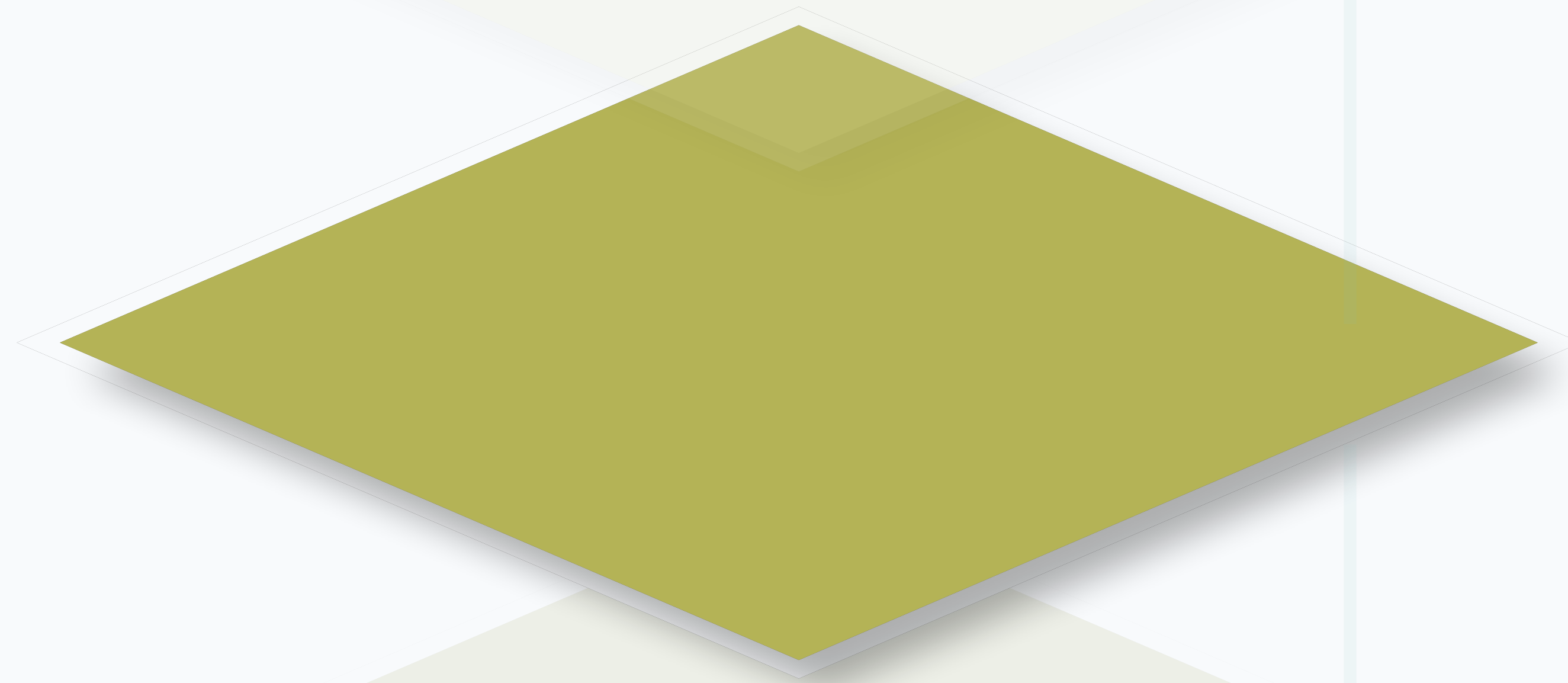
Application Platform



Technical Platform



Business Platform



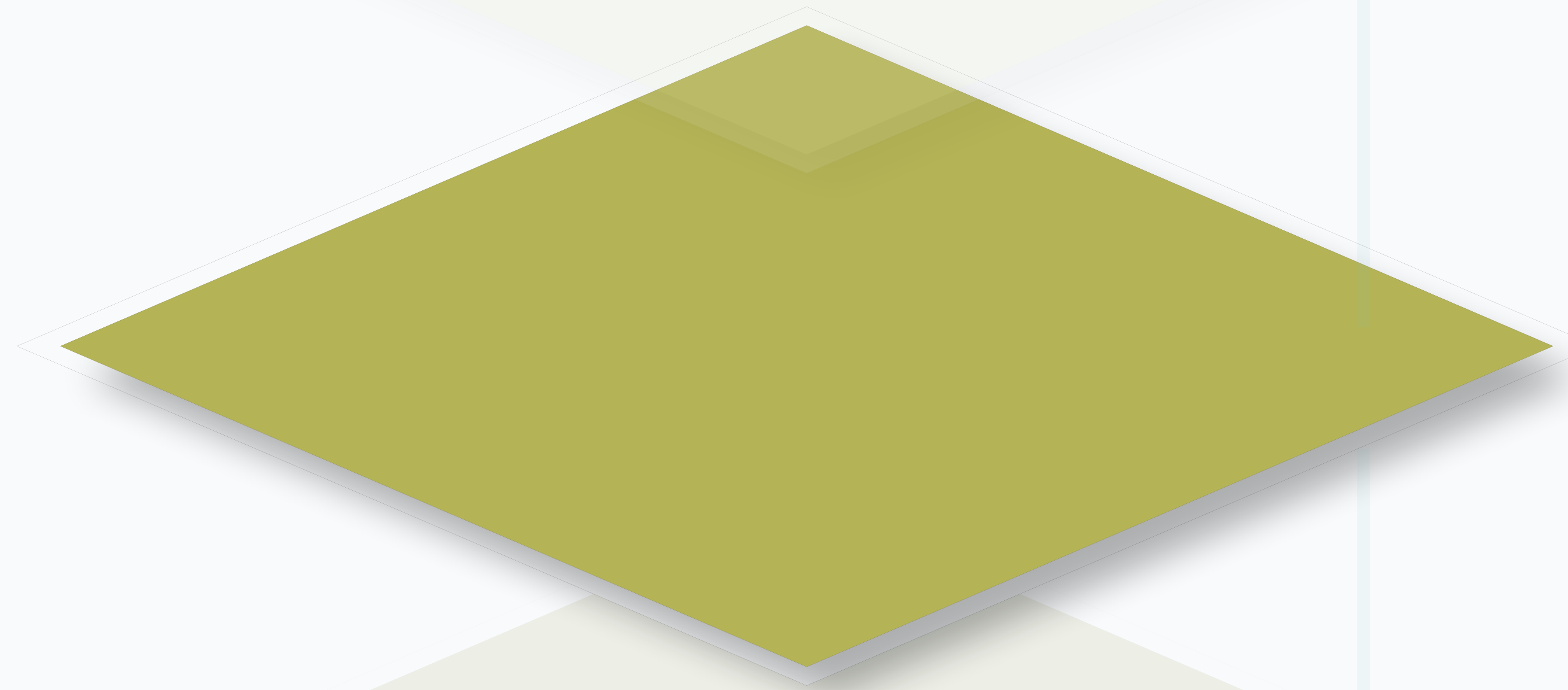
Application Platform



Technical Platform



Business Platform



Capability Platform



Technical Platform



Capability Platform



***Developers is
the customers***



Living instance



Self-Service

Log

```
package test;

import org.apache.log4j.Logger;

public class Log4JPropertiesTest {
    static Logger log = Logger.getLogger(Log4JPropertiesTest.class);

    public static void main(String[] args) {
        log.debug("This is a debug message");
        myMethod();
        log.info("This is an info message");
    }

    private static void myMethod() {
        try {
            throw new Exception("My Exception");
        } catch (Exception e) {
            log.error("This is an exception", e);
        }
    }
}
```

Type	Inputs	Actions
Files & directories <i>Upload a file, index a local file, or monitor an entire directory.</i>	5	Add new
TCP <i>Listen on a TCP port for incoming data, e.g. syslog.</i>	0	Add new
UDP <i>Listen on a UDP port for incoming data, e.g. syslog.</i>	0	Add new
Scripts <i>Run custom scripts to collect or generate more data.</i>	0	Add new
Database Configuration <i>Setup database connection.</i>	1	Add new
OPC DA Collector <i>Setup opc measure.</i>	5	Add new
Splunk RPC Startup <i>Start up RPC service server.</i>	1	Add new

Log4j

Splunk

Dashboards

```
1 class Dashing.Number extends Dashing.Widget
2   ready: ->
3     # This is fired when the widget is done being rendered
4
5   onData: (data) ->
6     # Fired when you receive data
7     # You could do something like have the widget flash each time data comes in by doing:
8     # $(@node).fadeOut().fadeIn()
9
10    # Any attribute that has the 'Dashing.AnimatedValue' will cause the number to animate when it changes.
11    @accessor 'current', Dashing.AnimatedValue
12
13    # Calculates the % difference between current & last values.
14    @accessor 'difference', ->
15      if @get('last')
16        last = parseInt(@get('last'))
17        current = parseInt(@get('current'))
18        if last != 0
19          diff = Math.abs(Math.round((current - last) / last * 100))
20          "#{diff}%"
21        else
22          ""
23
24    # Picks the direction of the arrow based on whether the current value is higher or lower than the last
```



Dashing

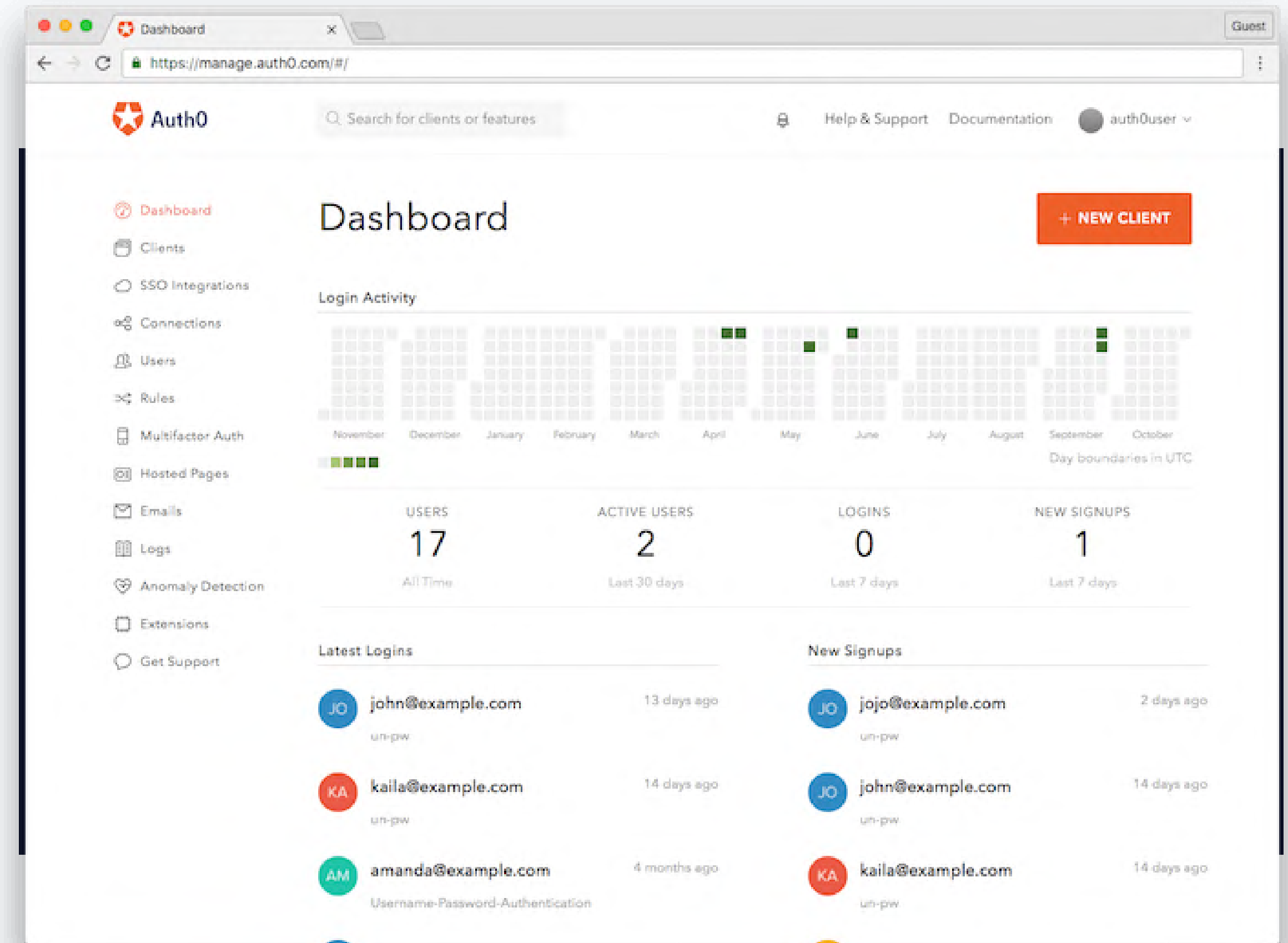
Grafana

Security

```
@Configuration
@EnableWebSecurity
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http
            .authorizeRequests()
                .antMatchers("/", "/home").permitAll()
                .anyRequest().authenticated()
                .and()
            .formLogin()
                .loginPage("/login")
                .permitAll()
                .and()
            .logout()
                .permitAll();
    }

    @Autowired
    public void configureGlobal(AuthenticationManagerBuilder auth) throws Excep
        auth
            .inMemoryAuthentication()
                .withUser("user").password("password").roles("USER");
    }
}
```

Spring Security



Auth0

NLP

```
DoccatModel model = null;
InputStream dataIn = null;
try {
    dataIn = new FileInputStream("en-sentiment.train");
    ObjectStream<String> lineStream =
        new PlainTextByLineStream(dataIn, "UTF-8");
    ObjectStream<DocumentSample> sampleStream = new DocumentSampleStream(lineStream);

    model = DocumentCategorizerME.train("en", sampleStream);
}
catch (IOException e) {
    // Failed to read or parse training data, training failed
    e.printStackTrace();
}
finally {
    if (dataIn != null) {
        try {
            dataIn.close();
        }
        catch (IOException e) {
            // Not an issue, training already finished.
            // The exception should be logged and investigated
            // if part of a production system.
            e.printStackTrace();
        }
    }
}
```

The screenshot shows the Wit.ai interface. At the top, a user message "Hi, my name is John" is displayed. Below it, a bot response "Hi, [contact]!" is shown, with a "wit/contact" entity highlighted. The interface includes a "Add a new entity" button and a "merge (context, entities)" button. The bottom of the interface has three tabs: "User says...", "Bot says...", and "Bot executes...".

OpenNLP

Wit.ai



Library



Service

Techniques

Cloud

Adopt - Archived (2010)





Capability Platform



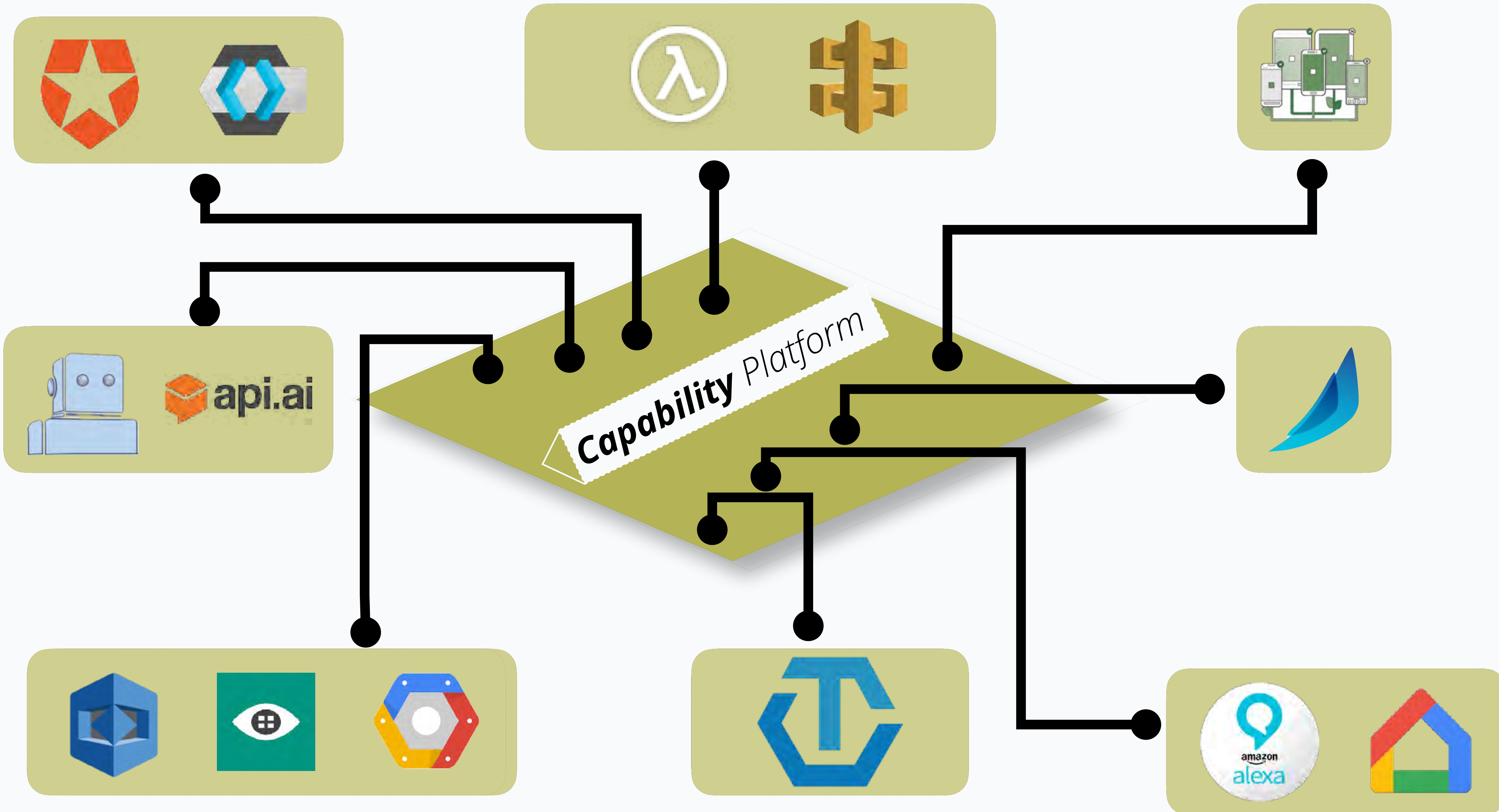
***Developers are
the customers***



Living instance



Self-Service



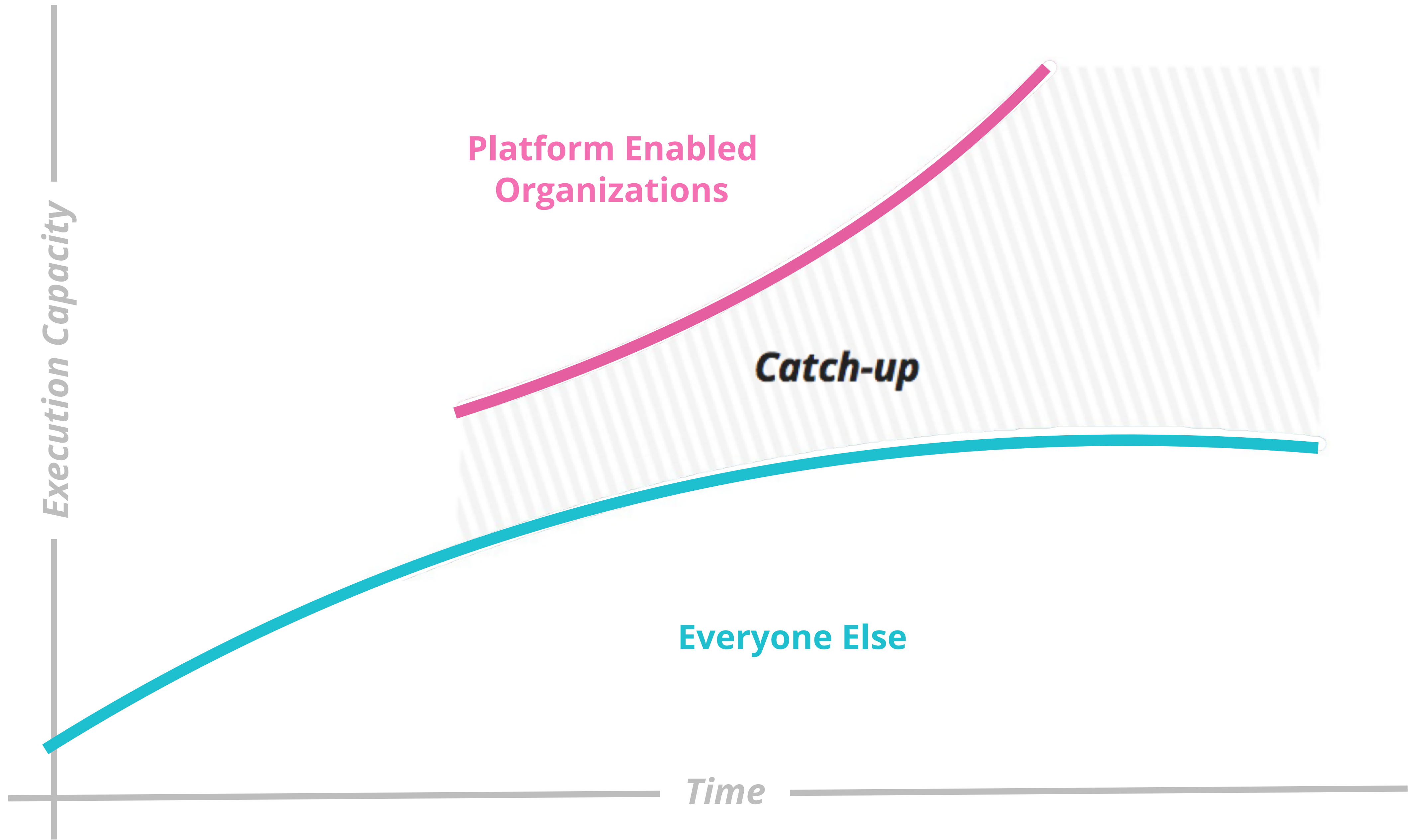
CAPABILITY AS PLATFORM

PLATFORM THINKING

GIVING GOVERNMENT FROM THE STORIES OF THE SONS

THAT GET IT RIGHT

BUILD YOUR PLATFORMS



A Blueprint that works

The ThoughtWorks Digital Platform Strategy offers a holistic approach that unleashes the power that an enterprise already possesses. By combining tailored platform solutions with practical strategic guidance, we collaborate with our partners to build and execute on a strategy to facilitate innovation, reduce time to market, and deliver unique competitive advantage.

Delivery Infrastructure

 **Deliver Faster**

- Elastic Infrastructure
- CD Pipelines
- Security
- Deployment Runtime
- Monitoring

API & Architecture Remediation

 **Build Ecosystem**

- Developer Experience (DX)
- Service Boundaries
- Event Driven Architecture
- Public Gateways
- Microservice SOA Topologies

Self Service Data

 **Gain Insights**

- Data Pipeline Design
- Realtime architectures (APIs)
- Data Lake Design
- Data as Product
- Granular Authorization

Experimental Infrastructure & Telemetry

 **Experiment Responsibly**

- Data Collection
- Canary Releasing Process
- Toggle Architecture
- Routing Technology
- Visualization & Instrumentation

Customer Touch Point Technology

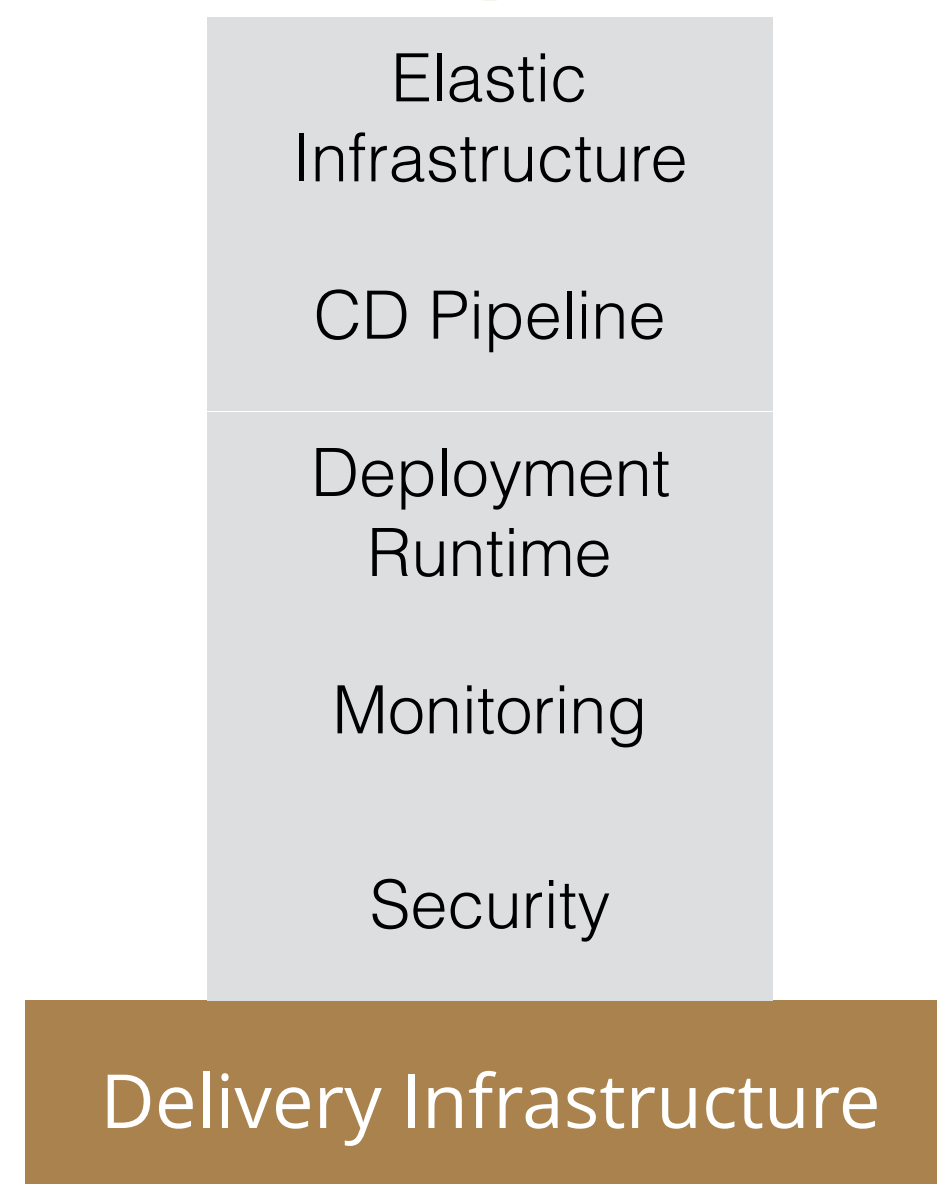
 **Consistent Experience**

- Single View of Customer
- Content Strategy
- Personalization
- CD for Mobile
- Channel Redirect

Engineering Organizational Design

DELIVER FASTER

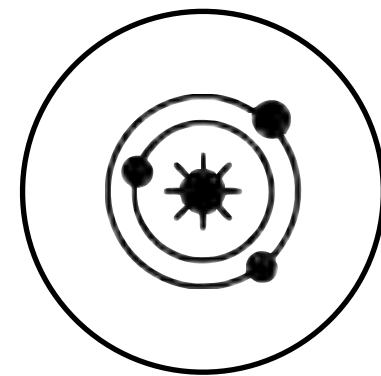
DELIVERY INFRASTRUCTURE



- Elastic infrastructure based on public/private cloud to enable environment consistency and on-demand provisioning.
- Continuous Delivery pipeline runs through application lifecycle, to ensure high quality and reliability.
- Deployment runtime environment should be highly standardized and automated.
- Comprehensive monitoring to IT operation and business operation should be in place to get information on system health.
- Security considerations should be built in system delivery process, instead of relying on security check at final stage.

BUILD ECOSYSTEM

API & ARCHITECTURE REMEDIATION



Developer
Experience

Service
Boundaries

Event Driven
Architecture

Public Gateways

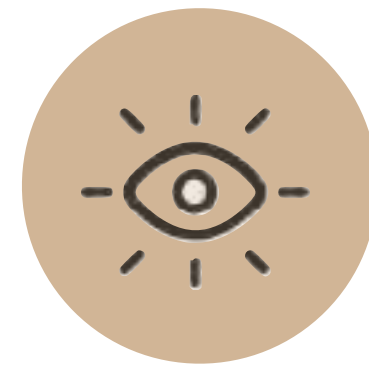
Microservice SOA
Topologies

API & Architecture
Remediation

- Developer experience should be considered in platform infrastructure and API design.
- Reasonable bounded context and service boundaries should be identified to ensure low coupling and high coherence of APIs.
- Asynchronous event mechanism should be employed to decouple domain models across service boundaries.
- API gateways should be provided as single facade of service consumers, and provide crosscutting features.
- API architecture should be designed in microservice style, and avoid overly complicated orchestration commonly seen in SOA architectures.

GAIN INSIGHTS

SELF-SERVICE DATA



Data Pipeline
Design

Realtime
Architectures

Data Lake Design

Data as Product

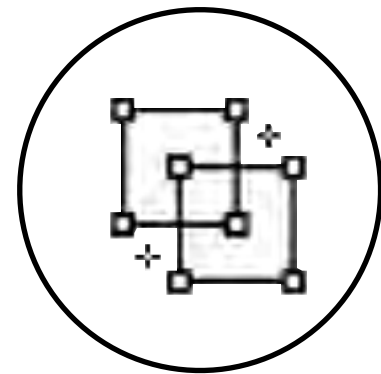
Granular
Authorization

Self-Service Data

- Data pipeline allows smooth data flow across collection, transformation, storage, exploration/prediction, visualization stages and to generate business insight.
- Realtime architecture and APIs handle large amount of non-structured data and influence decision in real time.
- Raw data is stored in data lake without ETL pre-process, so that teams can consume data later without relying on dedicated data warehouse teams.
- Encapsulate data to be products that 3rd parties can consume and generate further value.
- Granular authorization allows different data access based on user roles and permissions.

EXPERIMENT RESPONSIBLY

EXPERIMENTAL INFRASTRUCTURE & TELEMETRY



Data Collection

Canary Releasing
Process

Toggle
Architecture

Routing
Technology

Visualization &
Instrumentation

Experimental Infrastructure
& Telemetry

- Collect data from various sources to support innovation.
- Instrument the system according to business goals, and represent monitor results to business stakeholders.
- Release new version to a small portion of users as “canaries” to reduce risk of experiment.
- Feature toggles allow the team to shift system behavior without changing code. Working together with routing technologies such as blue-green deployment and A/B testing, it enables controlled experiments effectively.

CONSISTENT EXPERIENCE

CUSTOMER TOUCH POINT TECHNOLOGY



Single View of Customer

Content Strategy

Personalization

CD for Mobile

Channel Redirect

Customer Touch Point Technology

- Establish unique, consistent, accurate and holistic view to customers (and target customers).
- Consistent content strategy according to customer's profile and characteristics of digital channels.
- Direct customer journey across multiple channels (e.g. PC, mobile, physical shops) to create encounter with right time, right location, and right way.
- Personalized content, service and product recommendation.
- Continuous delivery for all types of digital channel applications, especially mobile applications, to ensure responsiveness.

A Blueprint that works

The ThoughtWorks Digital Platform Strategy offers a holistic approach that unleashes the power that an enterprise already possesses. By combining tailored platform solutions with practical strategic guidance, we collaborate with our partners to build and execute on a strategy to facilitate innovation, reduce time to market, and deliver unique competitive advantage.

Delivery Infrastructure

 **Deliver Faster**

- Elastic Infrastructure
- CD Pipelines
- Security
- Deployment Runtime
- Monitoring

API & Architecture Remediation

 **Build Ecosystem**

- Developer Experience (DX)
- Service Boundaries
- Event Driven Architecture
- Public Gateways
- Microservice SOA Topologies

Self Service Data

 **Gain Insights**

- Data Pipeline Design
- Realtime architectures (APIs)
- Data Lake Design
- Data as Product
- Granular Authorization

Experimental Infrastructure & Telemetry

 **Experiment Responsibly**

- Data Collection
- Canary Releasing Process
- Toggle Architecture
- Routing Technology
- Visualization & Instrumentation

Customer Touch Point Technology

 **Consistent Experience**

- Single View of Customer
- Content Strategy
- Personalization
- CD for Mobile
- Channel Redirect

Engineering Organizational Design

DIGITAL CAPABILITY MAP

CONSUMER GOODS CORPORATION

Key Focus

- Enable agile delivery, improve responsiveness to business need
- Generate business insight through data and controlled experiment
- Assure consistency and reusability in architecture and service design strategy across multiple projects

Delivery Infrastructure	API & Architecture Remediation	Self Service Data	Experimental Infrastructure	Customer Touch Point Technology
Elastic Infrastructure	Developer Experience	Data Pipeline Design	Data Collection	Single View of Customer
CD Pipelines	Service Boundaries	Realtime Architecture	Canary Releasing Process	Content Strategy
Security	Event Driven Architecture	Data Lake Design	Toggle Architecture	Personalization
Deployment Runtime	Public Gateways	Data as Product	Routing Technology	CD for Mobile
Monitoring	Microservices SOA Topologies	Granular Authorization	Visualization & Instrumentation	Channel Redirect





THANK YOU!

[#TWTechRadar](#)

[@thoughtworks](#)



ThoughtWorks®

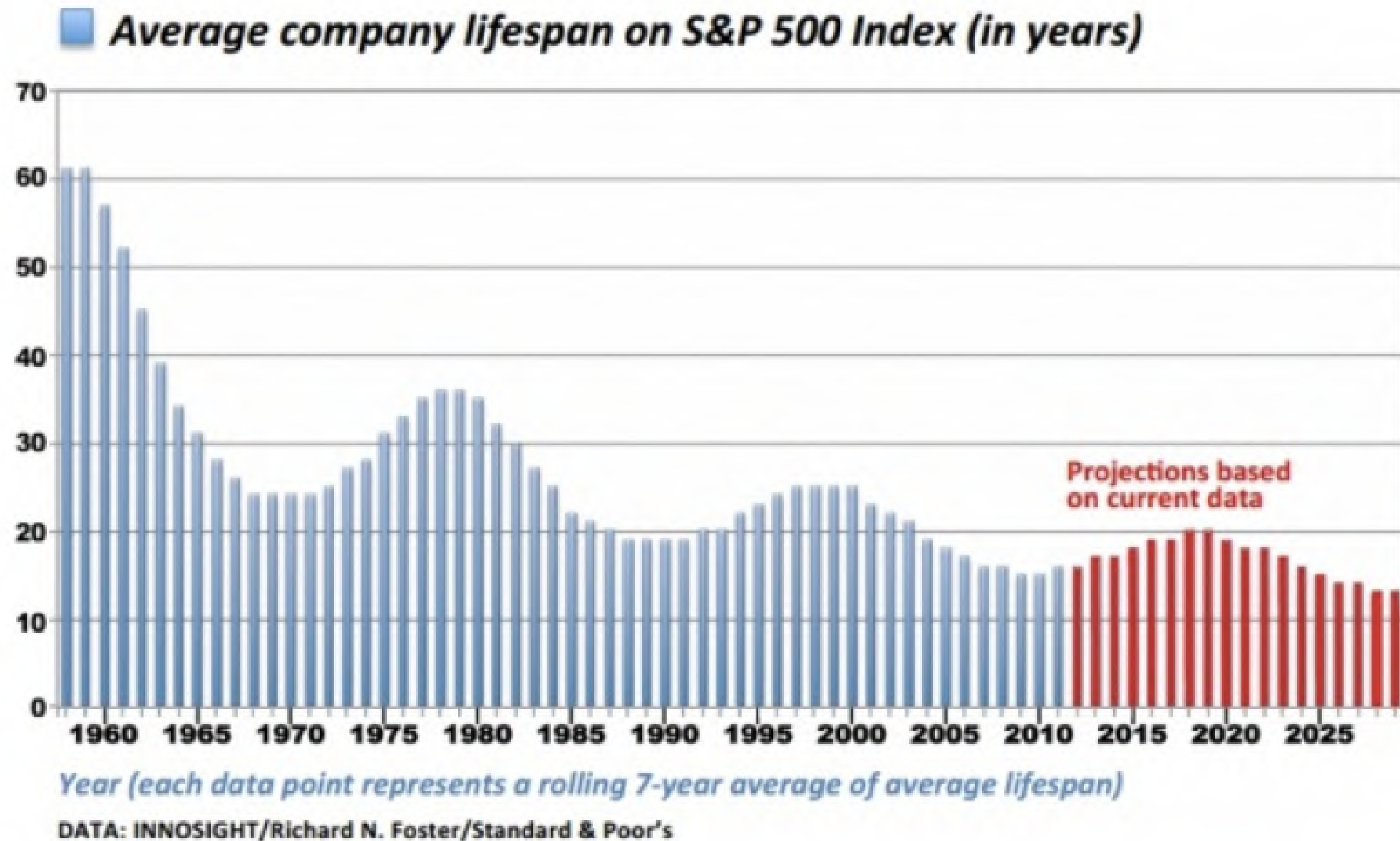
构建数字平台 助力企业创新

DIGITAL PLATFORM STRATEGY

ThoughtWorks®

数字化时代的挑战

数字化时代企业面临的挑战



和用户的距离?
对用户喜好的
变化是否敏
感?

数字化思维?

员工满意度降
低, 难以维持
活力和仓断力?

巨额投资难以
产生成效?

产值达到100亿

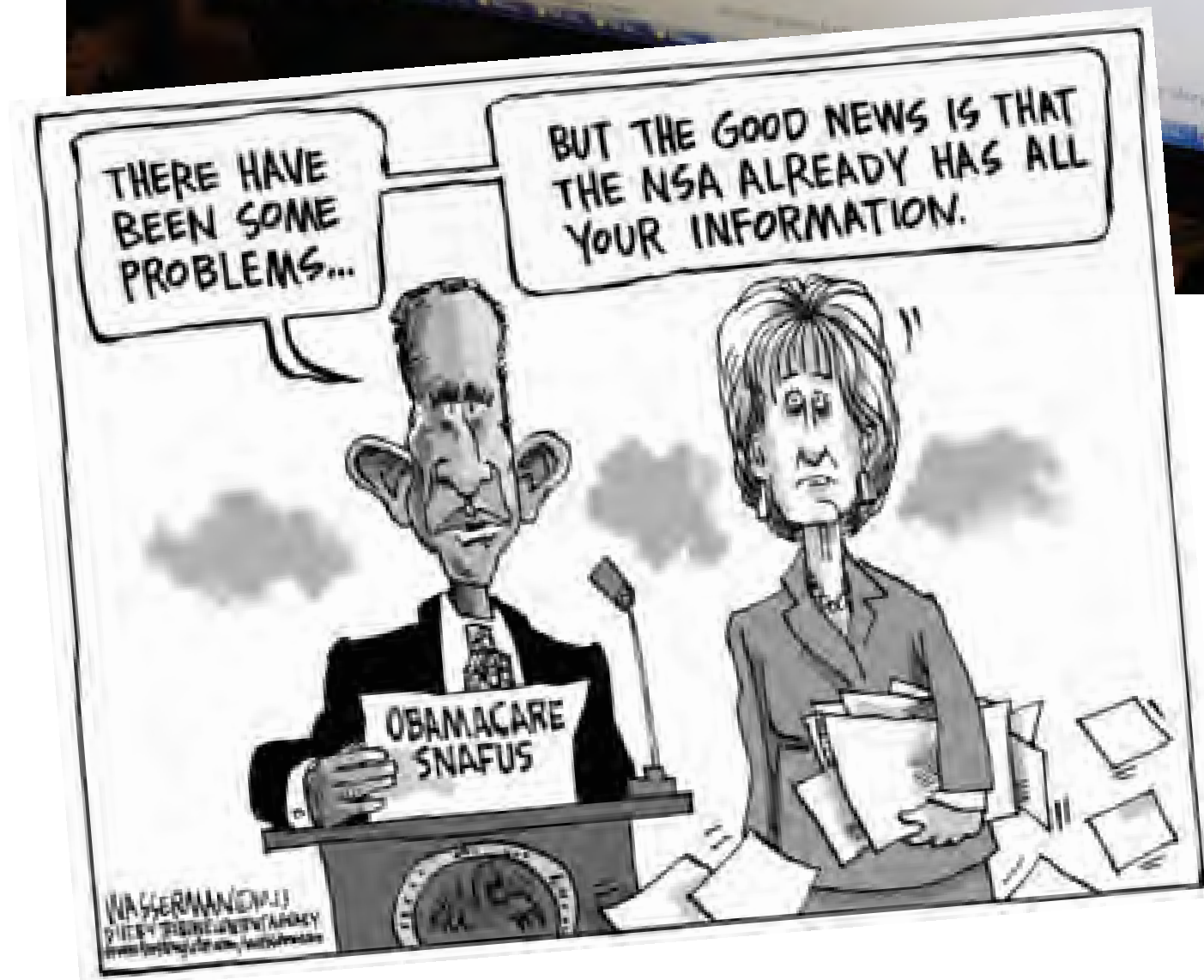
首钢用了71年

联想用了13年

小米用了3年

数字化时代企业面临的挑战

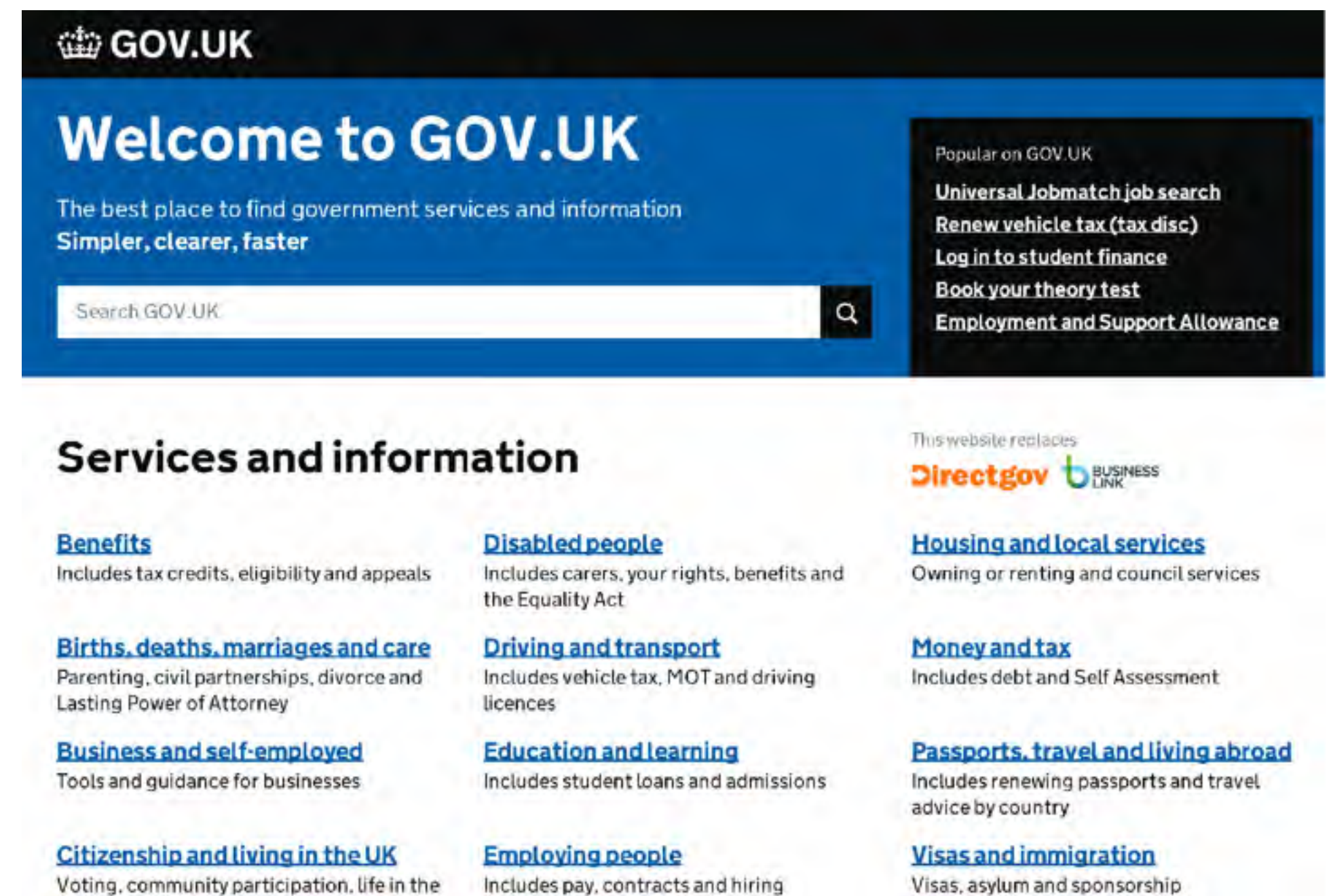
- 支撑奥巴马医疗法案的美国HealthCare.gov网站
 - 随着奥巴马医疗改革方案遭遇技术危机，承包商的支付额飞涨
 - 三年以后，网站仍然市场处于不稳定和不可用阶段，
- 英国国家IT发展计划
 - 英国国家卫生署交付一套全新的IT基础设施和一个数字化的病历系统，在2002年启动时，预计将花费23亿英镑。
 - 尽管该计划在2011年被取消了，但人们预计它如果到结束时的成本会超过100亿英镑。



精益 - 敏捷有助于解决问题

• GOV.UK

- Alpha版本只用了12个星期上线，花了£261,000. (14个人的团队)
- 在2012-2013财年，Government Digital Service (GDS) 通过以GOV.UK替换掉Directgov和BusinessLink网站节省了 4千万英镑
- 在2013-2014财年，预计GDC将通过GOV.UK关闭更多的政府网站并将其纳入单一域名节省超过5千万英镑
- 该项目采用了和传统IT不同的模式进行开发，也就是精益敏捷的管理、工程能力和相适应的架构
- 交付团队由政府公务员为主，在政府大楼里的偏僻会议室内完成了Alpha, Beta版

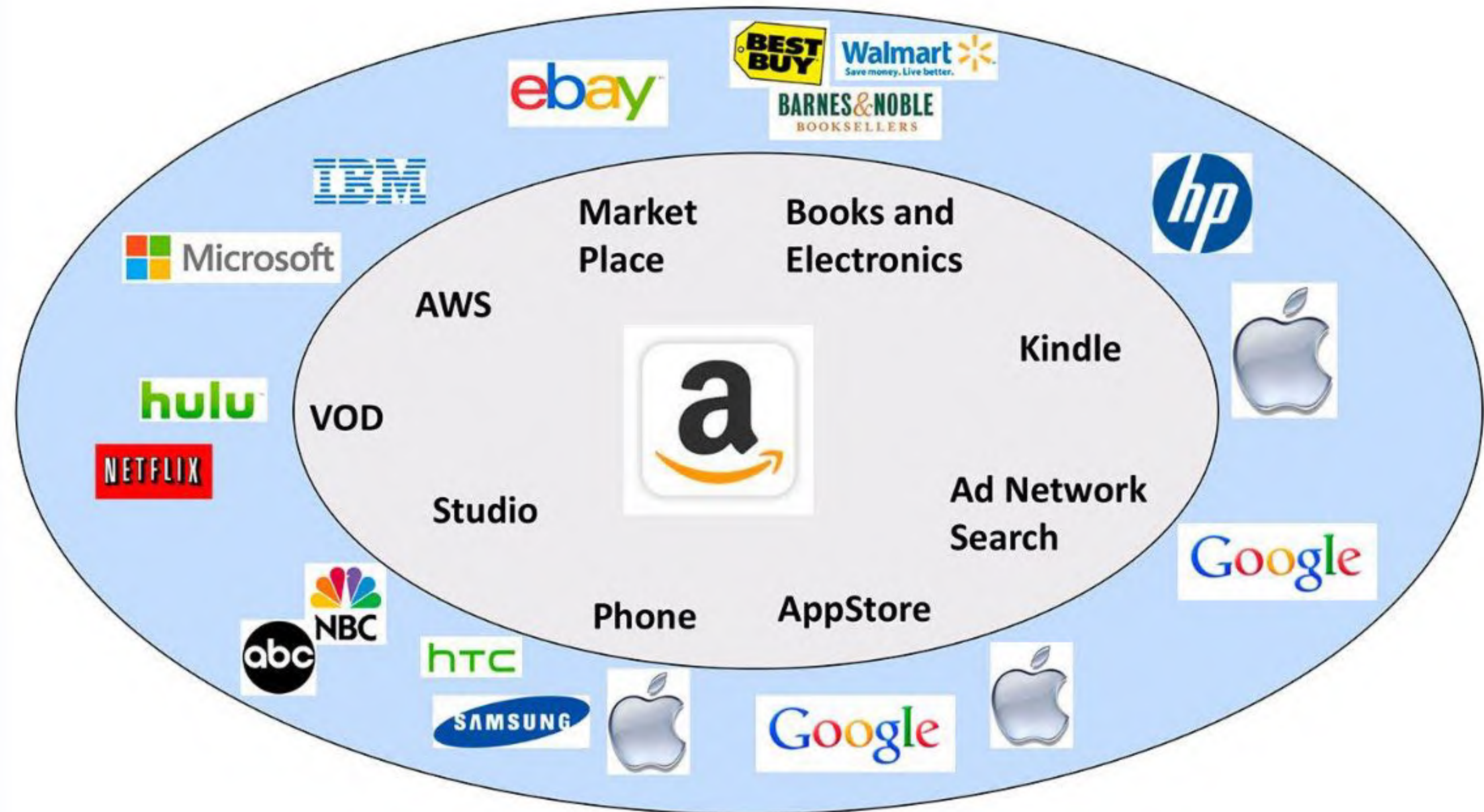


快速变化是数字化企业的核心竞争力

数字化企业的代表亚马逊，在零售、广告、消费电子终端、应用商店、云服务等多个领域与各领域的领先企业竞争。

亚马逊还有Dash Button、Echo、Prime Air、AWS等大量创新。据AWS的CEO说，还有更多创新项目失败了——而亚马逊认为完全OK。

数字化企业的核心能力是快速实验、快速学习、快速调整。单靠模仿一款产品或一种业务模式，无法跟上数字化企业的步伐。

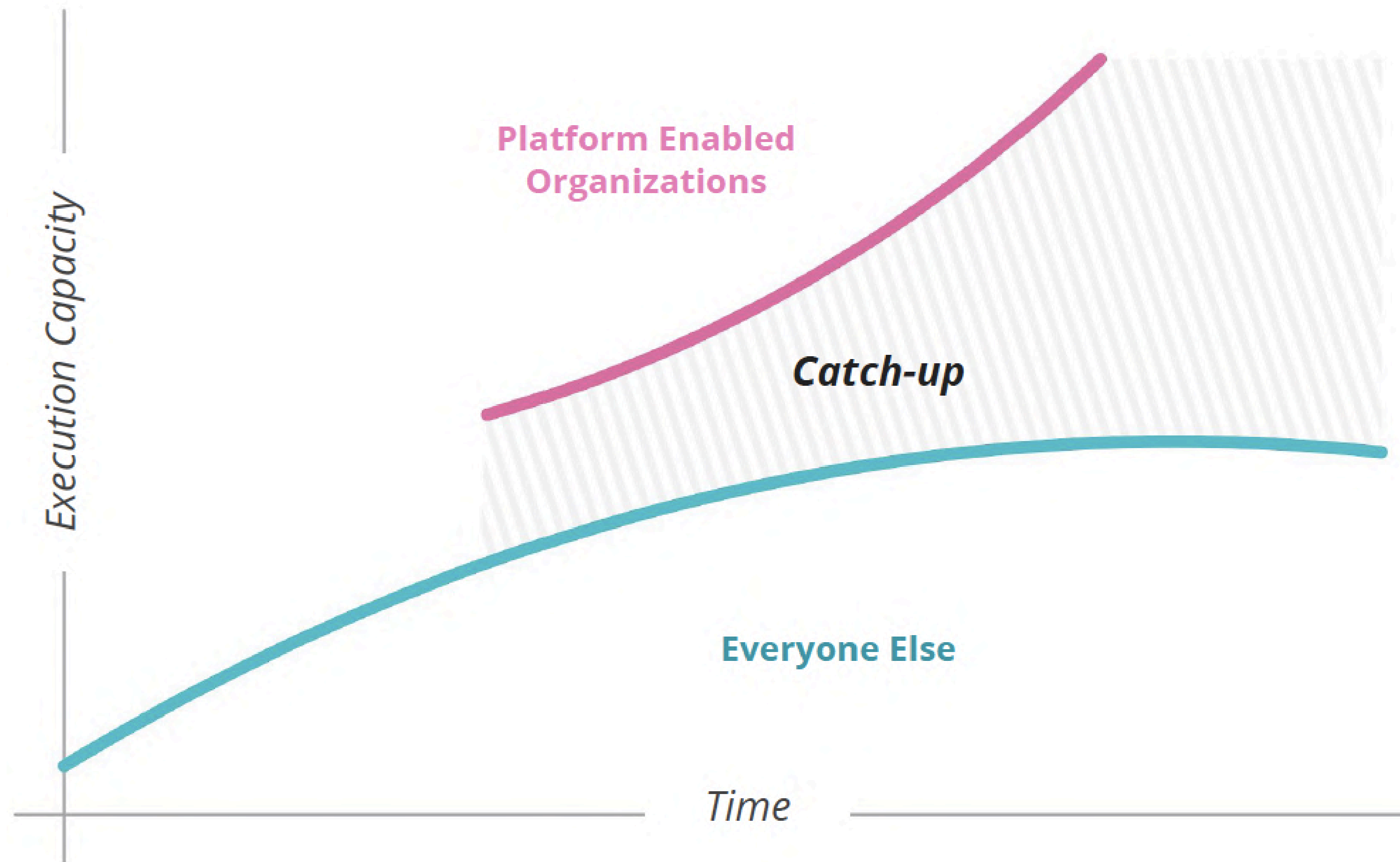


传统企业如何应对数字化挑战

经过多年的技术投资、建设、并购、整合，很多传统企业积累了大量与信息技术相关的系统、人员、流程和文化。他们感到这些遗产阻碍了探索和创新。

与此同时，新进入行业的数字化企业能够更快地响应变化、尝试各种可能性、并从失败中快速学习。

不断建设割裂的信息系统无助于企业追赶数字化浪潮。企业需要用平台思维看待数字化能力，方能激活核心资产，构建独特竞争优势。

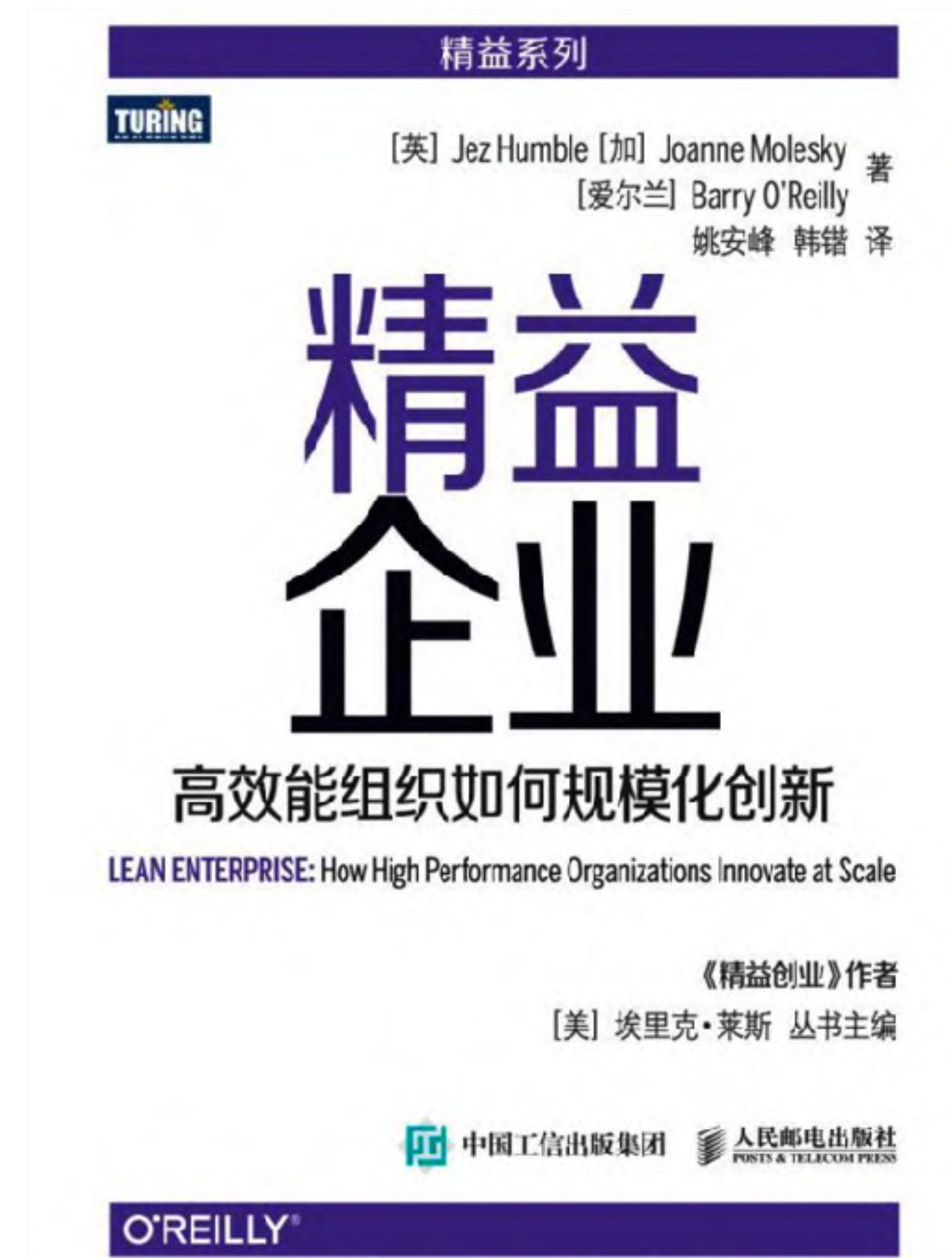


下一代软件研发

高效能

创新

规模化



ThoughtWorks®

数字平台战略

DIGITIZATION GOALS

企业的数字化转型目标

扫除技术障碍，充分利用企业多年积累的宝贵资产，建立面向未来的数字化平台。

快速交付

更高的发布频率
更短的响应周期
更快的故障恢复

洞察业务

发现新的商业机会
识别经营短板
催生行业变革

快速创新

快速实现创新想法
复用已有资源
创新效果一目了然

提供用户渠道

建立完备多样的用户渠道
推动新业务快速触达用户
提供渠道分发能力

支撑用户体验

品牌形象一致性
不同渠道展现统一优质的体验
同时充分利用平台优势

构建生态系统

提供开发协作的技术基础
与合作伙伴对接整合
共建行业生态

快速交付

	获取环境	配置环境	管理数据库	测试	QA环境管理	UAT管理	预发布	生产	认证授权	数据内建	系统监控	CyleTime
	申请流程(3天)	手工配置(2天)	手动	没有测试 手工测试	手动(1天)	手动(1天)	手动(1-3天)	手动管理 手动扩容 (2-10天)	定制开发 (3-10天)	无	无	大于30天
	一键完成 (10分钟)	无	自动完成 无须干预	TDD 自动化测试	自动完成 无须干预	自动完成 无须干预	一键完成 容器化部署	一键完成 容器化部署	基于SSO 重用OAuth	安全扫描 安全测试 恶意用户防 御	监控视图 系统预警 快速定位 熔断	小于1天
差距	3天	2天	每周1-10人 天	有 / 无	1天	1天	3天	10天	10天	有 / 无	有 / 无	大于29天
弹性基础架构		持续交付流水线 部署运行时						安全架构 监控设施				



交付基础架构 Delivery Infrastructure

构建生态系统

	技术栈管理	开发者体验	内部集成环境	解耦的架构设计	团队内聚性	合作伙伴集成
	手动配置 定期检查 (每周1-3天)	手动管理环境 手动配置技术栈 (每周1天)	手动管理API文档 复杂的团队依赖 接口管理靠开会 (每周2天)	无	频繁开会 定期架构重构 (每周1-3天)	定制开发 (1-2个月)
	自动化管理 一键生成 (30分钟)	统一开发环境 统一技术栈 一键式配置环境 (30分钟)	契约式开发 自助式的API	DDD 系统解耦 EventStorming 四色建模	去中心化的组织 去中心化的系统 (无需集中管理)	统一API管理 统一权限控制 统一安全策略 微服务架构 (10天)
差距	3天	每周1天	每周2天	有 / 无 服务划分	每周3天	50天
	技术栈管理 开发者体验			微服务SOA 事件驱动的架构		公共网关



API与架构治理 API & Architecture

洞察业务

	业务可视	系统状态可视	业务数据归集	业务改进	业务风控	营销推送	定制化数据
	BI引擎 手动报表 (延迟1-30天)	手动报表 系统日志 (延迟1-3天)	数据库 ETL (延迟3-30天)	BI引擎 手动报表 (延迟3-30天)	BI引擎 手动报表 (延迟1-30天)	定期手动发送 (每隔7天)	无
	业务全流程动态可视 可随时查看	实时可视	数据湖	业务数据回溯 预测性数据 验证性数据	实时监控 持续审计 风险预警	根据用户行为自动发生	根据用户定制数据
差距	30天	3天	30天	30天	30天	7天	有 / 无
	数据管道		数据湖	实时性架构		数据即产品	细粒度鉴权

自服务的数据 Self Service Data



快速创新

	创新业务实现	创新效果测量	用户测试	版本与流量管理
	<p>新作系统 打通渠道 (4-10个月)</p>	<p>数据埋点 构建数据处理与展示系统 (5-20天)</p>	<p>手动设置用户路由 手动细分目标用户 (10-20天)</p>	<p>手动分批次部署 手动回滚 手动处理版本差异 (2-20天)</p>
	<p>重用IT资产 快速触及用户 (1-3个月)</p>	<p>系统数据采集 重用现有数据平台 (3-5天)</p>	<p>系统控制流量路由 系统自动细分用户用户 (1-3天)</p>	<p>系统管理分批次部署 统一的分支管理 无需回滚 (3-5天)</p>
差距	270天	17天	19天	17天
	持续交付	统一数据采集 可视化与仪表盘	流量路由	Toggle架构 金丝雀部署

实验性架构

Experimental Infrastructure

用户体验与用户渠道

	内容管理	全渠道用户体验	内容个性化	用户触达	全渠道管理
	各个渠道分别管理 频繁沟通对齐体验设计 (2-3个月)	各渠道分别管理 各业务分别管理 (2-3个月)	无	修改系统 重新发布 App审核 (2-4周)	无
	重用内容管理服务 (5-10天)	重用内容管理服务 (5-10天)	精细的用户细分	App热部署 (1-2天)	多渠道分发 内容个性化 统一数据
差距	60天	60天	有 / 无	19天	有 / 无
统一用户视图 内容策略 内容个性化			App持续交付 渠道分发		

用户触点技术

Customer Touch Point Technology

DPS

数字平台战略



弹性基础设施

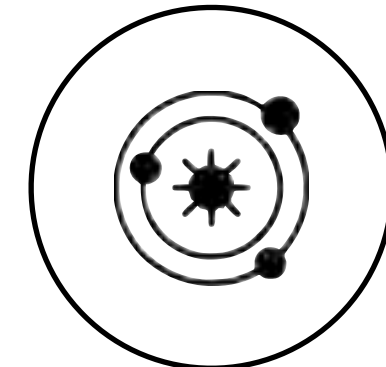
持续交付流水线

部署运行时

监控

安全

交付基础设施



开发者体验

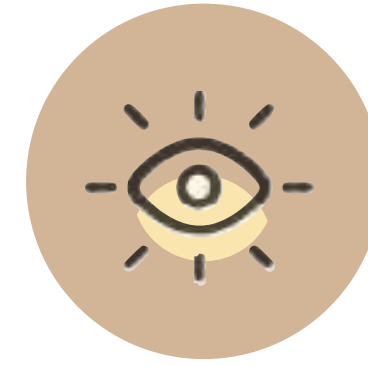
服务边界

事件驱动架构

公共网关

微服务SOA拓扑

API 和架构治理



数据流水线设计

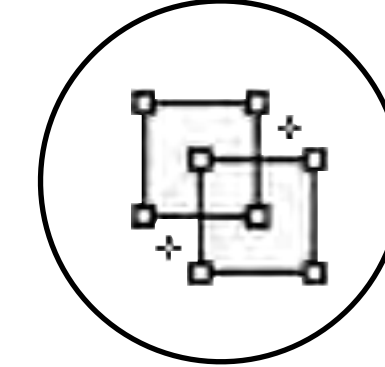
实时架构和API

数据湖设计

数据即产品

细粒度授权

数据自服务



数据采集

金丝雀发布

Toggle架构

路由技术

埋点和可视化

创新实验基础设施和监控体系



统一顾客视角

内容战略

个性化

移动应用持续交付

跨渠道引流

客户触点技术

ThoughtWorks®

落地实施方式

按需渐进式的实施

通信

- 科技公司强调快速交付
- 通信行业供应链复杂，需要开放协作
- 产品迭代迅速，需要快速触达客户

快速交付	构建生态	获取洞见	快速创新	体验一致性
弹性基础架构	开发者体验	数据管道设计	数据采集	一致性视图
持续交付流水线	服务边界	实时性架构	金丝雀发布	内容战略
安全	事件驱动架构	数据湖	Toggle架构	个性化
部署运行时	公共网关	数据即产品	流量路由	App持续交付
监控设施	微服务SOA	细粒度鉴权	可视化与仪表盘	渠道分发

银行

- 银行业强调安全和高可用
- 银行业相对封闭，不注重构建生态
- 银行业强调渠道，关注线上线下的全渠道分发

快速交付	构建生态	获取洞见	快速创新	体验一致性
弹性基础架构	开发者体验	数据管道设计	数据采集	一致性视图
持续交付流水线	服务边界	实时性架构	金丝雀发布	内容战略
安全	事件驱动架构	数据湖	Toggle架构	个性化
部署运行时	公共网关	数据即产品	流量路由	App持续交付
监控设施	微服务SOA	细粒度鉴权	可视化与仪表盘	渠道分发

按需渐进式的实施

汽车

- 该公司强调快速交付
- 行业相对封闭，不看重构建生态
- 车企以往不能直销，因此对渠道投入不足，新政之后会有改变

快速交付	构建生态	获取洞见	快速创新	体验一致性
弹性基础架构	开发者体验	数据管道设计	数据采集	一致性视图
持续交付流水线	服务边界	实时性架构	金丝雀发布	内容战略
安全	事件驱动架构	数据湖	Toggle架构	个性化
部署运行时	公共网关	数据即产品	流量路由	App持续交付
监控设施	微服务SOA	细粒度鉴权	可视化与仪表盘	渠道分发

零售

- 零售对IT快速交付的关注有限
- 零售产业链复杂，因此注重构建外部生态
- 零售对用户触点技术非常关注

快速交付	构建生态	获取洞见	快速创新	体验一致性
弹性基础架构	开发者体验	数据管道设计	数据采集	一致性视图
持续交付流水线	服务边界	实时性架构	金丝雀发布	内容战略
安全	事件驱动架构	数据湖	Toggle架构	个性化
部署运行时	公共网关	数据即产品	流量路由	App持续交付
监控设施	微服务SOA	细粒度鉴权	可视化与仪表盘	渠道分发

THANK YOU

期待为您的创新服务

ThoughtWorks®



平台化能力建设

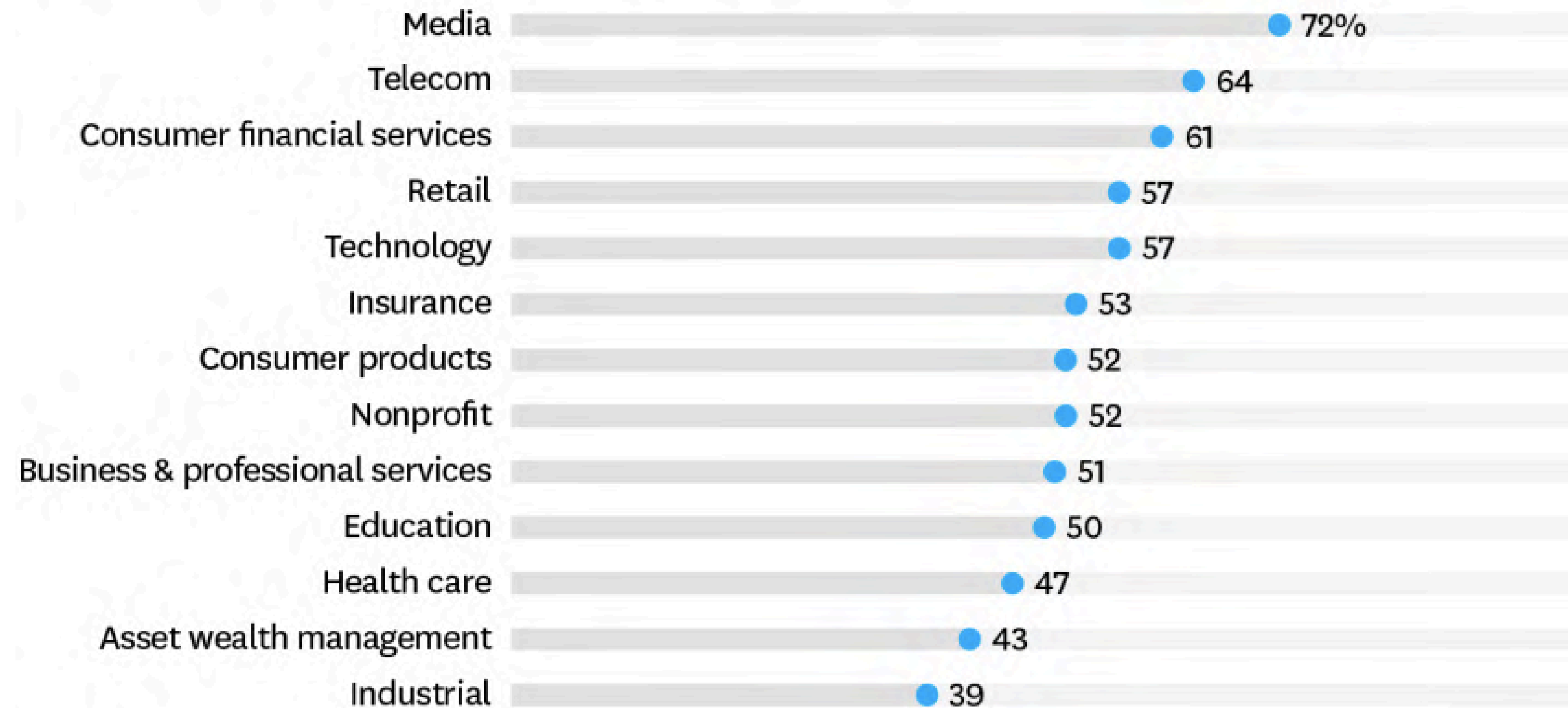
技术领导者即服务

ThoughtWorks®

数字化浪潮冲击传统行业

据《哈佛商业评论》的调查，媒体、通信、消费者金融服务、零售、科技、保险、消费者产品、专业服务和教育等10个行业受到数字化浪潮的影响最大，半数以上的企业高管认为自己的业务将在12个月内受到较大幅度的冲击。

Executives Who Anticipate Moderate or Massive Digital Disruption in the Next 12 Months, by Industry



SOURCE "DIGITAL PULSE 2015," BY RUSSELL REYNOLDS ASSOCIATES

<https://hbr.org/2016/03/the-industries-that-are-being-disrupted-the-most-by-digital>

© HBR.ORG



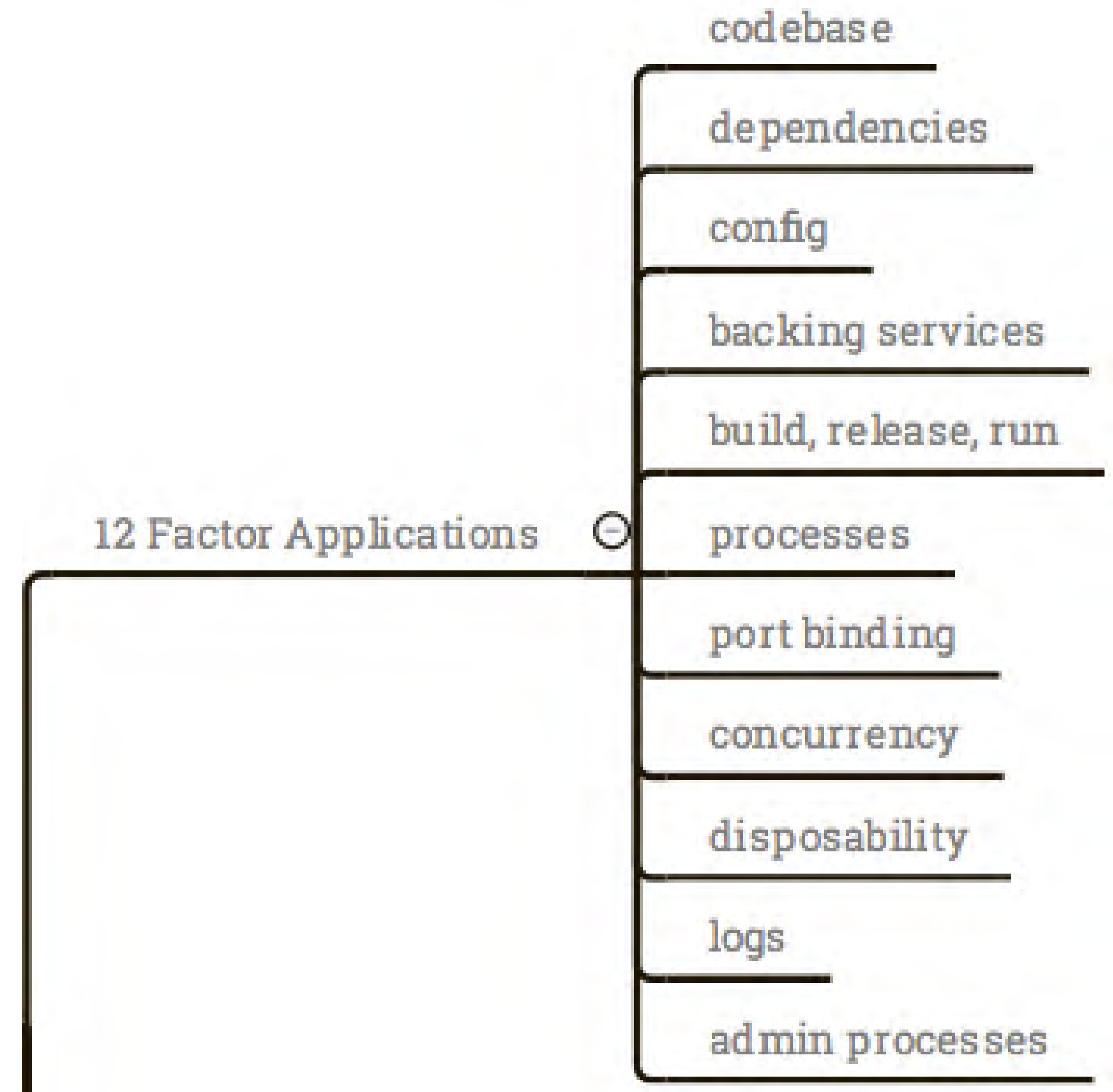
21世纪什么最贵？

人才！

如何保障架构设计先进性

“12要素应用（12-factor application）是Heroku最初发展的一组云原生（cloud native）应用的架构模式。这12个要素描绘了一个应用程序的原型框架，能够充分发挥云计算的优势。”

然而我不知道 该怎么做.....



如何保障交付质量

“持续集成是一种软件开发实践。采用持续集成时，开发人员会定期将代码变更合并到一个中央代码库中，之后系统会自动运行构建和测试操作。持续集成的主要目标是更快发现并解决缺陷，提高软件质量，并减少验证和发布新软件更新所需的时间。”

然而我的项目里 并没有……



如何建设**团队能力**

“Spring Boot的一大优点是提高开发者的生产力，因为它已经提供了许多通用的功能，很容易进行装配及使用。它允许开发者选择在整个框架中他们所需的那部分，而无需使用庞大的、或是不必要的运行时依赖。”

然而培训并没有 什么x用.....



IT技术能力三大难题



制订适合项目要求、紧跟技术潮流的技术方案，需要知识渊博经验丰富的架构师



保障交付顺利开展，用持续交付流水线拉动敏捷技术实践，需要能力出色的团队技术领导者



结合项目情况制订有针对性的能力评估和发展办法，需要理论实践紧密结合的教练

当然，这些人都很贵……

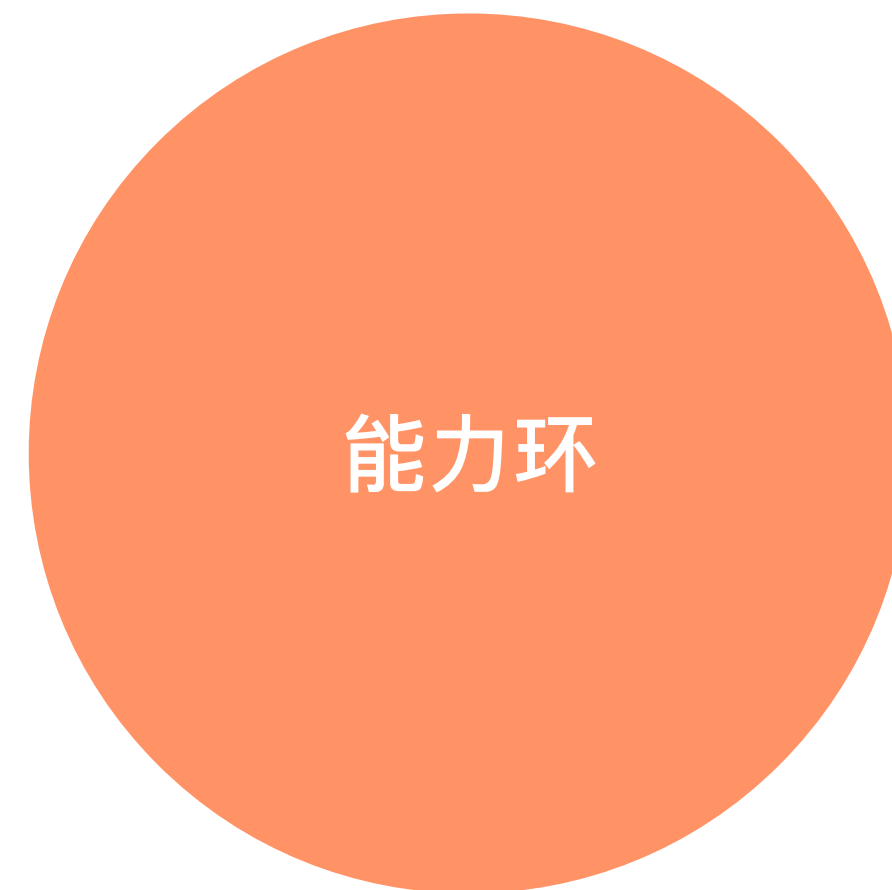
方案-交付-能力三环脱节



- 如何选择最适合业务需求的技术方案?
- 如何确保跨项目技术方案统一性?
- 如何不断演进技术方案并推广到各个团队?



- 如何避免开发/测试环境管理的麻烦?
- 如何建立流畅的自动化交付流水线?
- 如何保证交付达到基本质量标准?



- 如何评估团队技术能力?
- 如何识别项目能力风险?
- 如何有针对性地开展能力建设?



我们怎么办

云计算能帮忙吗？

用平台承载技术能力

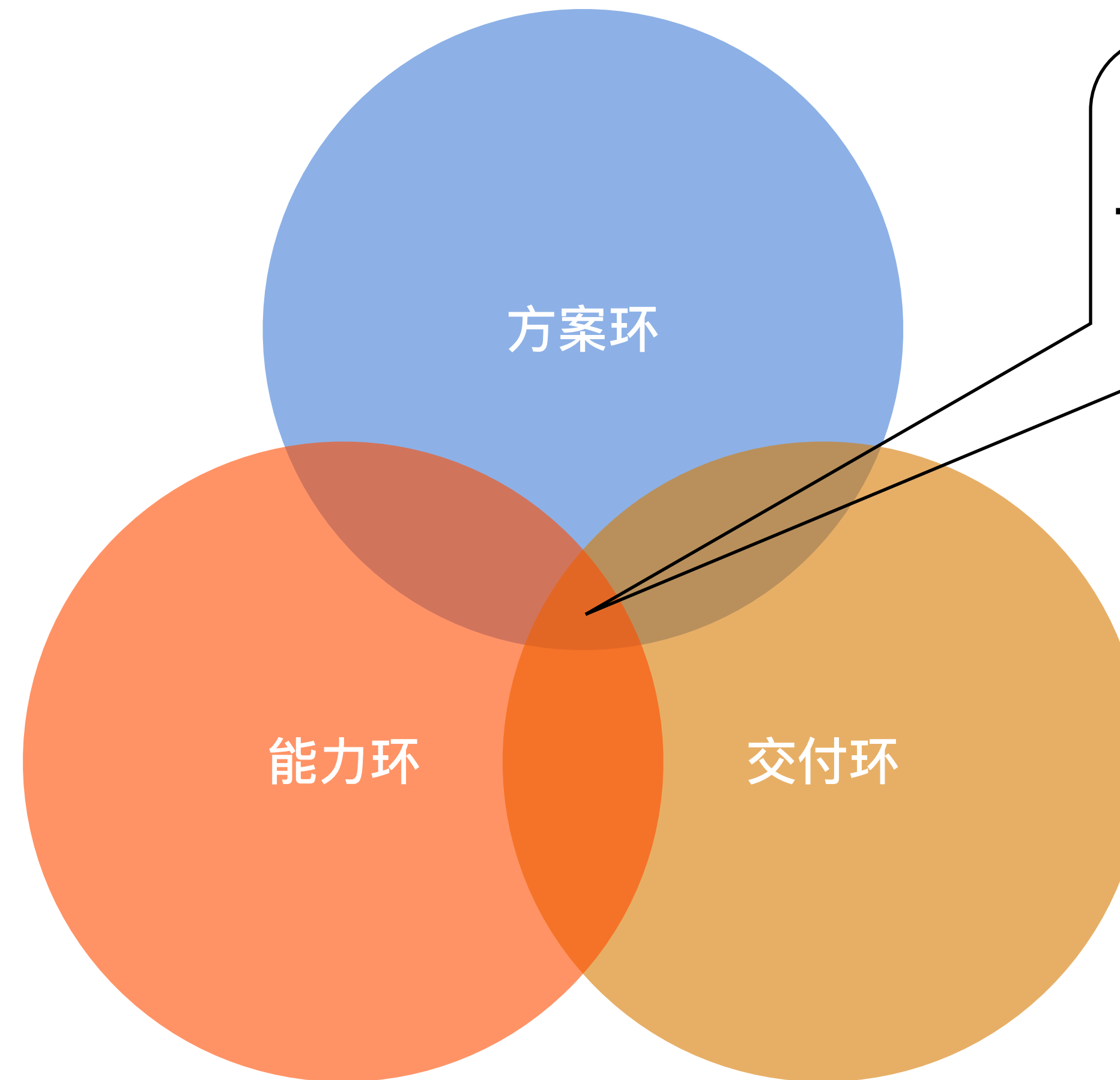
软件开发者作为直接用户

将能力/资源封装成在线服务

自助式按需使用服务

方案-交付-能力三环联动

- 用技术栈的形式，记录用一组特定技术、解决一类特定问题的最佳方式。
- 借助云平台的自动化能力和弹性能力，将原本依赖架构师/技术领导者的能力转化为云服务：
 - 制订适合项目要求的技术方案
 - 保障交付顺利开展
 - 管理和提升团队的能力



以技术栈为核心，研发云平台为支撑，方案-交付-能力三环联动

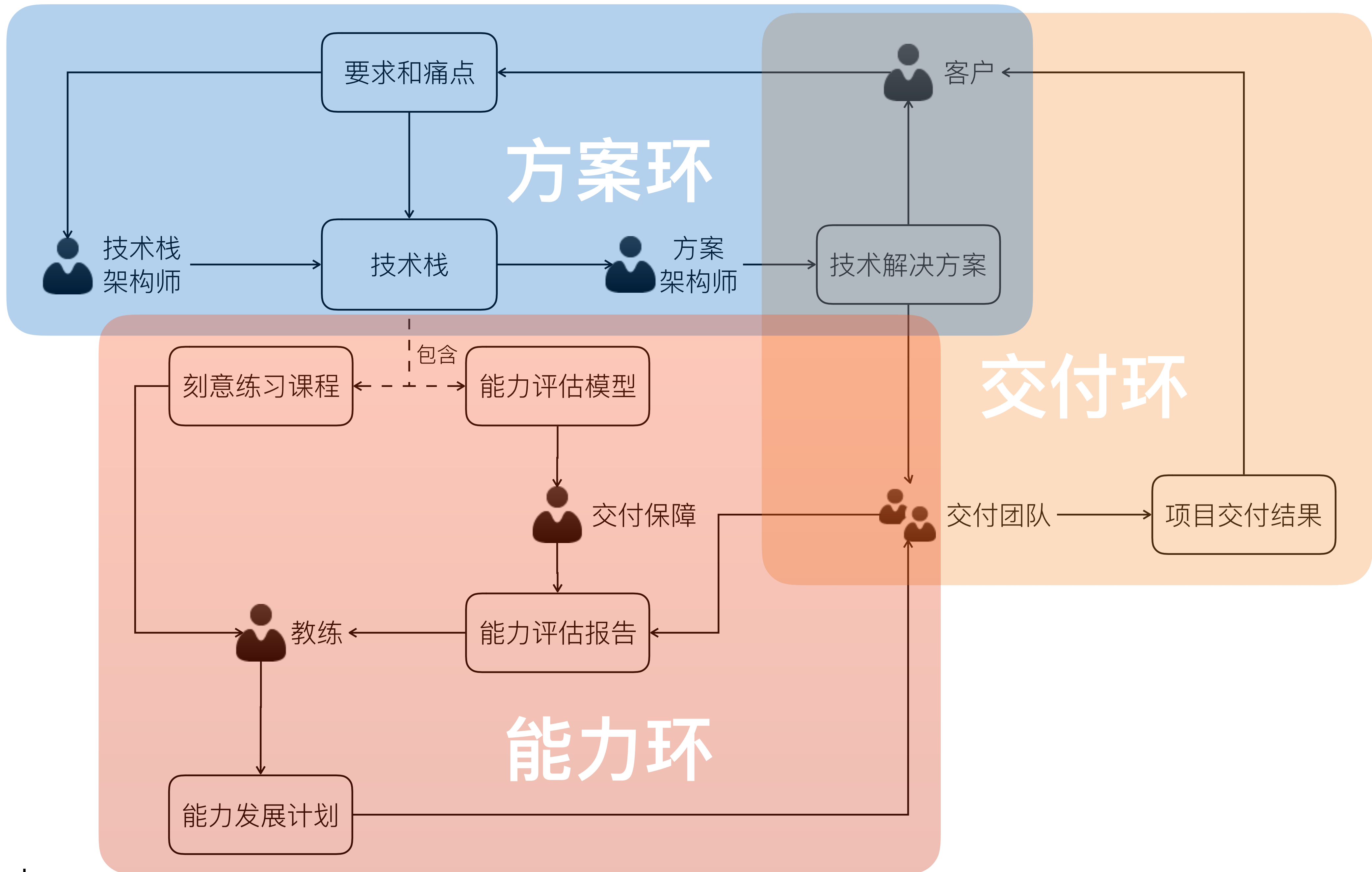
什么是技术栈

类似“5年Java经验”这样的描述无法帮助我们了解一个程序员的能力，我们还是不知道他/她会做什么、能做多快/多好。

技术栈应该包含三方面具体信息：

- 用什么工具；
- 解决什么问题；
- 如何度量解决问题的效果。





三环联动模型中的关键角色

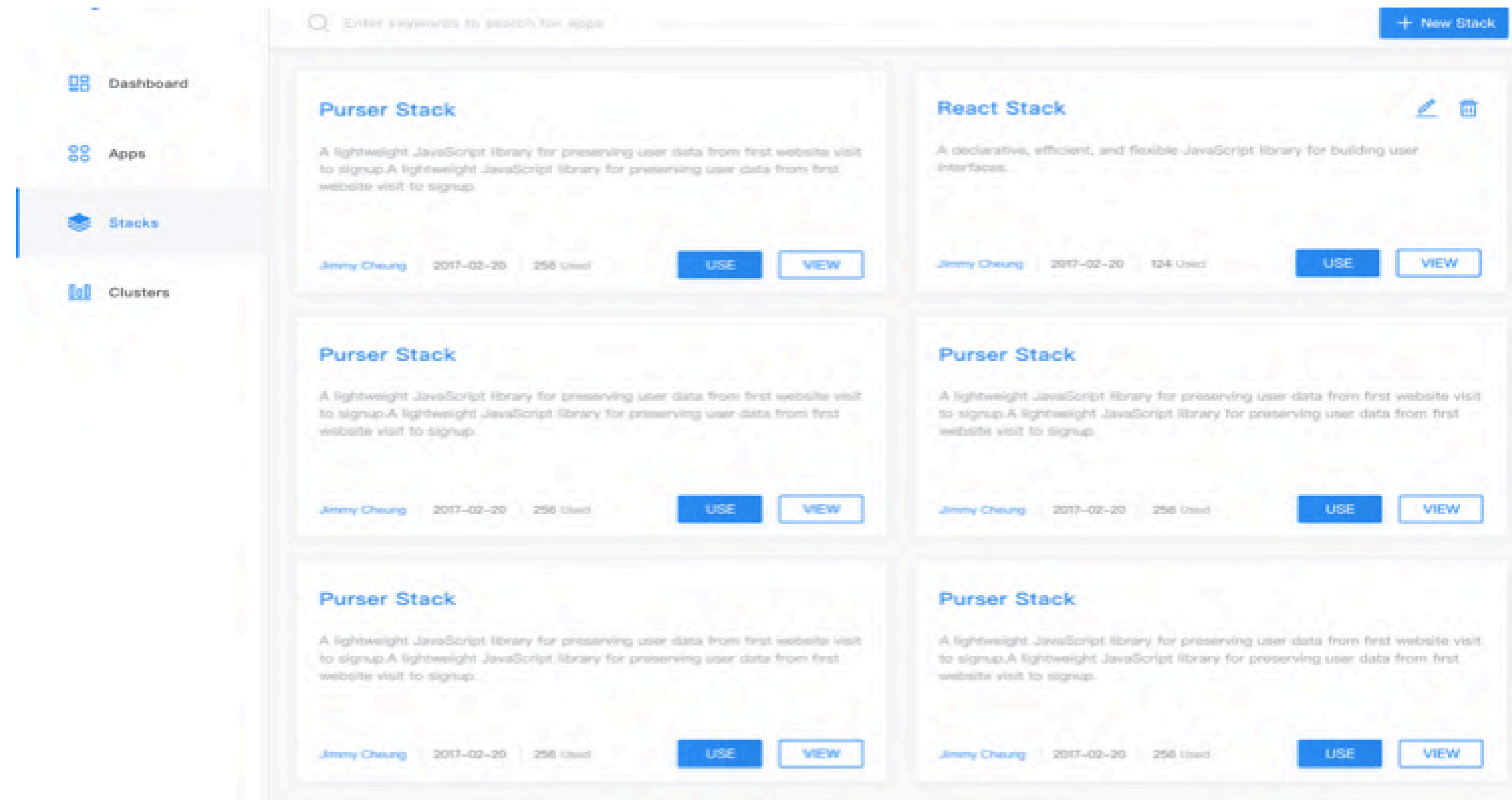
角色	职责
技术栈架构师	根据市场和技术发展趋势，定义技术栈，包括能力评估模型和刻意练习课程
方案架构师	结合客户、项目需求和技术栈能力，创建技术解决方案
交付保障	根据能力评估模型，评估团队/个人能力水平，识别能力差距和交付风险
教练	参与创建刻意练习课程，结合团队能力差距，定制适合团队的能力发展计划
协调者	推动和保障三环流程正常运转，收集反馈，改进流程



支撑系统

方案环： 技术栈管理平台

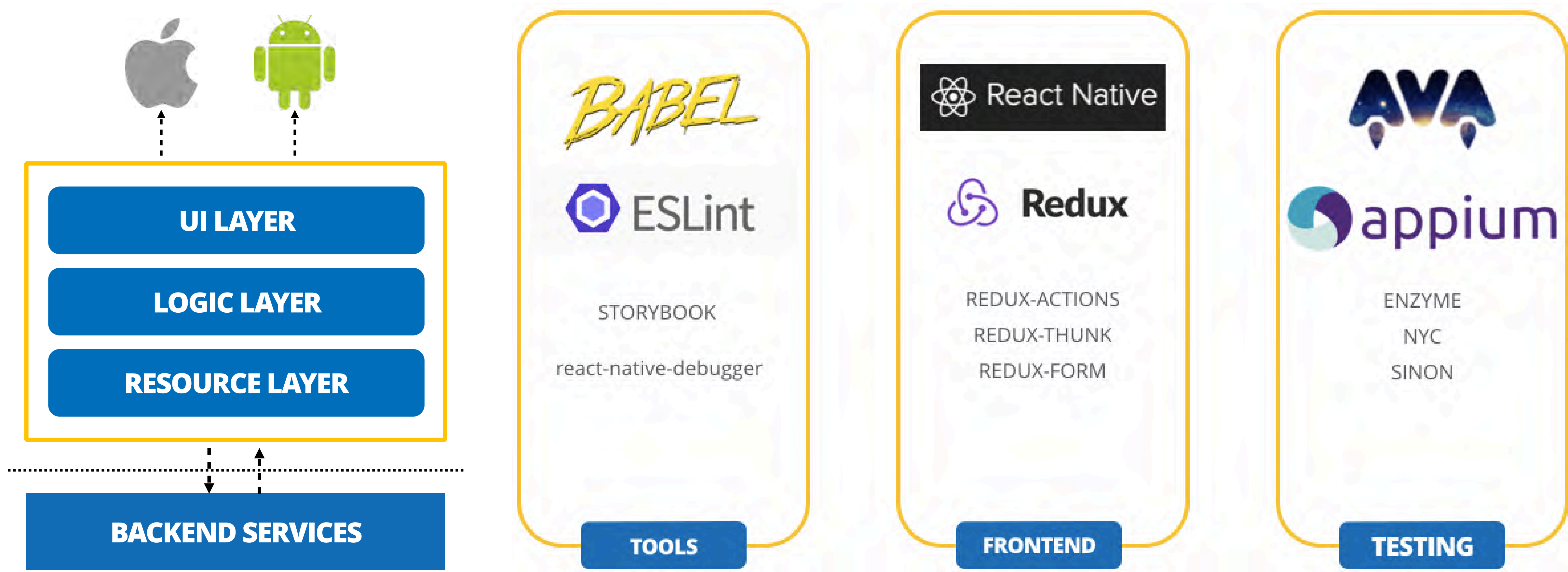
- 集中管理常用技术栈，全组织复用最佳经验与实践
- 基于PaaS平台10分钟启动新项目，节省项目启动前期选型和搭建框架工作量，规避技术选型风险
- 支撑技术栈架构师维护升级现有技术栈，持续引入优秀工具和实践，迅速推广复制到所有项目团队



例如：微服务基础架构

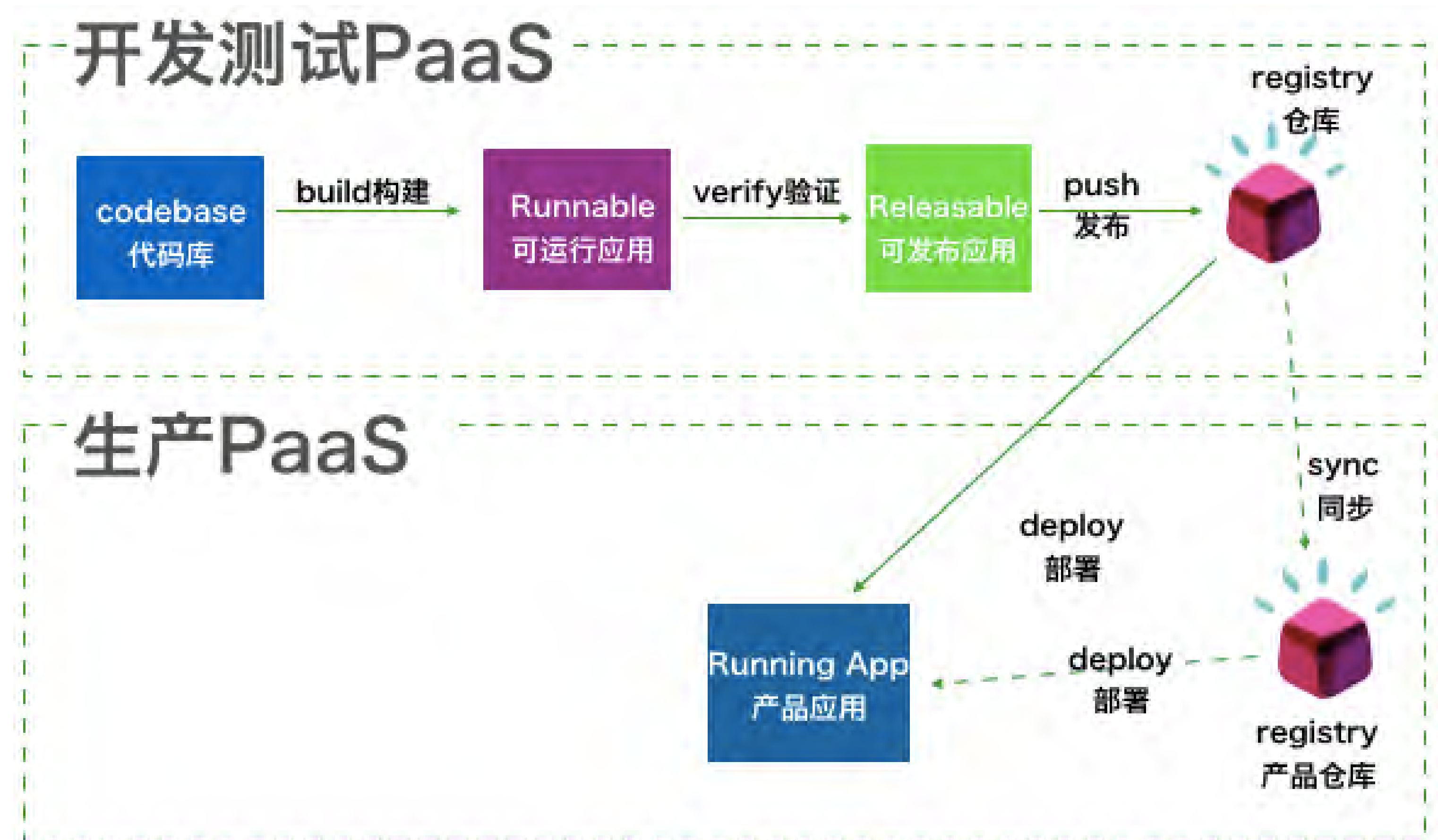


例如：移动应用基础架构



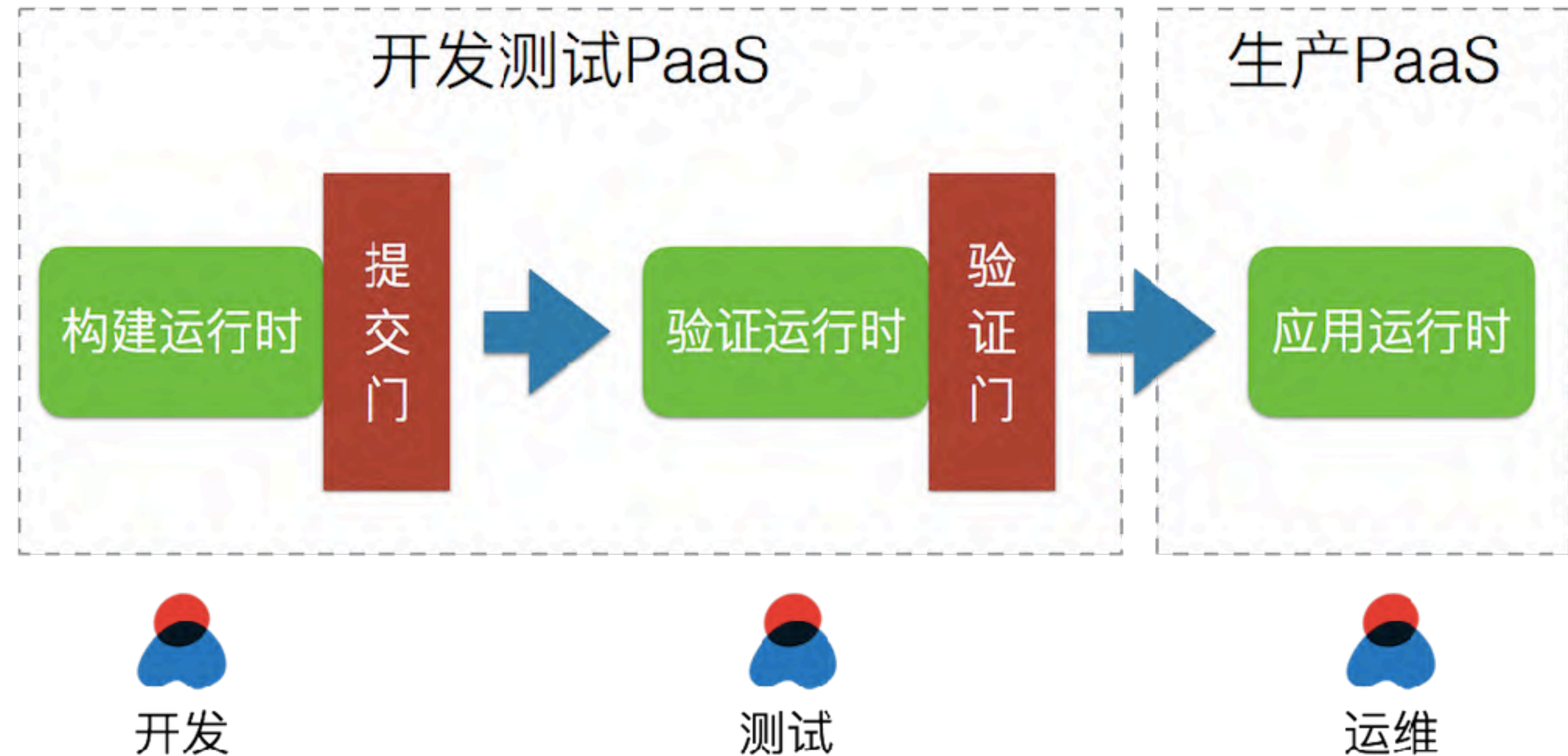
交付环：云原生研发环境

- 通过开发测试PaaS，一键式开通标准开发/测试环境，降低新人启动难度
- 统一管理构建、验证、应用三种运行时环境，消除环境不一致的问题，提高研发效率
- 标准化持续交付流程，强制质量检查，在开发过程中植入质量保障



云原生持续交付流程

- 集中式的持续集成服务器是计算资源短缺和版本控制工具受限的时代背景的产物
- 云原生环境下可以制订一套标准的构建流水线，并要求计算环境向这套流水线对齐
- 这时，持续集成就可以不必是每个团队的技术领导者都掌握的“技能”和“手艺”，可以针对一类项目定制和快速复制



云原生交付10要素

脚手架

项目初始代码基于脚手架生成，内嵌推荐架构和技术选型

自包含应用

应用程序自包含，不依赖应用服务器等中间件，最小化环境依赖

个人构建

开发者个人构建发生在容器内和/或云上，与团队构建完全一致

镜像化

环境依赖和构建产物都以虚拟机/容器镜像形式保存和版本管理

基础设施即代码

运行环境以代码形式配置，通过命令行和编程接口管理

流水线即代码

持续集成流水线以代码形式配置，随代码库同步更新

构建产物仓库

构建产物（包括容器镜像）纳入仓库统一管理，支持跨团队复用

团队共享工具

项目管理、代码仓、持续集成、知识库等团队共享工具开箱即用

服务质量透明

生产环境依赖以服务形式提供，隐藏服务质量水平（SLA）内部细节

弹性部署

各级环境的部署都可以动态伸缩，按需支持各种测试和发布策略

能力环：能力评估和发展平台

- 以技术栈为核心，明确定义团队能力基线：
 - 用什么技术栈
 - 解决什么问题
 - 长时间完成
 - 达到哪些基本质量要求
- 真题编程考试，准确度量团队成员能力水平
- 配套刻意练习计划，支撑有针对性的能力建设活动

Capabilities of Rotation Engineering Program

Capability Matrix Stacks

	OpenAPI4/jersey-mysql	OpenAPI3/jersey-mysql	OpenAPI2/jersey-mysql	Open Api/jersey-mysql
hxu	evaluating...	Evaluation	Evaluation	Evaluation
jksun	Evaluation	Evaluation	Evaluation	evaluating...
scxu	Evaluation	Evaluation	Evaluation	Evaluation
zzshen	✓	✓	✓	✓
pzzheng	Evaluation	✓	✓	✓
syzhang	Evaluation	✓	✓	✓
hpan	evaluating...	✓	✓	✓
ybzhou	Evaluation	✓	✓	Evaluation
zyongliu	Evaluation	✓	✓	✓

Add Capability

Stack History

- May 22nd 17 Create stack jersey-mysql for OpenAPI4
- May 21st 17 Create stack jersey-mysql for Open Api
- May 21st 17 Create stack jersey-mysql for OpenAPI2
- May 21st 17 Remove stack jersey-mysql for Open Api

研发能力建设从**科举**开始

- 根据项目/产品需求，统一设置和管理技术栈
- 针对技术栈定义测评题目和标准
- 自动生成测评代码库
- 自动统计测评结果

所有测评

Single Page App / webpack-nightwatch

Created at 6/21/2017, 9:07:50 AM
已经用时 17327 分钟

未通过

你需要在你的电脑上安装 `cde` 和 `git`

```
curl -sSL http://ketsu-api-web.cnpw.io/users/jxiong/evaluations/eval-es-6-todo-mvc-jxiong-149800727000
```

然后你可以通过 `git push cde master` 提交代码，提交代码后会自动执行相应的测试，你在本地的命令行上可以看得到执行的结果

不是替代，而是扩展技术领导者的影响力

越多对技术领导者的技术支撑，就会越少“人品驱动开发”的项目

方案环

- 如何选择最适合业务需求的技术方案？
- 如何确保跨项目技术方案统一性？
- 如何不断演进技术方案并推广到各个团队？

- 如何评估团队技术能力？
- 如何识别项目能力风险？
- 如何有针对性地开展能力建设？

能力环

交付环

- 如何避免开发/测试环境管理的麻烦？
- 如何建立流畅的自动化交付流水线？
- 如何保证交付达到基本质量标准？



熊节 | 总监咨询师
ThoughtWorks®

2017中国企业 敏捷实施情况调查

中国敏捷实施现状如何？
企业敏捷实施过程中会遇到哪些困难？
哪些实践是比较普遍的？



如果你已经开始实践敏捷，请**扫码参与调查**。同ThoughtWorks一起，为中国敏捷行业打造这份权威报告。

您需要一份权威报告！

深入了解敏捷趋势、最佳实践和经验教训，帮助你成功实现敏捷转型。