Atitit 2018 技术趋势与没落技术总结

目录

[1. 2018 技术雷达 1](#_Toc6638)

[1.1. HOSTED IDENTITY MANAGEMENT AS A SERVICE （SaaS）身份管理 1](#_Toc3781)

[1.2. POLYCLOUD策略，不再把所有业务全 “押在”一个服务供应商身上 2](#_Toc17341)

[1.3. GRAPHQL FOR SERVER-SIDE RESOURCE AGGREGATION。GraphQL可 以让客户端直接使用特定的查询语句去访问BFF以获取数 据 2](#_Toc31182)

[1.4. WEB BLUETOOTH能够直接从浏览器控制任意低功耗蓝 牙设备 3](#_Toc27231)

[1.5. Appium，它是最受欢迎 的移动端自动化测试框架之一。 3](#_Toc27466)

[1.6. KOTLIN的使用率得到了飞速增长， 3](#_Toc4064)

[1.7. 多云 账号（MULTI-ACCOUNT CLOUD SETUP） 4](#_Toc3725)

[1.8. 密钥即服务 （SECRETS AS A SERVICE） 4](#_Toc26089)

[1.9. Headless CMS （Content Management Systems, 内容管 理系统） 4](#_Toc23448)

[1.10. Cypress 端到端测试 5](#_Toc25973)

[1.11. Firefox 无头模式(HEADLESS FIREFOX) 5](#_Toc11406)

[1.12. MERMAID 使用类似 markdown 的标记语言来生成图表。 5](#_Toc30948)

[1.13. VISUAL STUDIO CODE 是微软推出的免费 IDE 编辑 器， 5](#_Toc32756)

[1.14. (Stanford CoreNLP) 越来越多的项目需要处理非结构化的数据，而从文本中 提取出有意义的业务信息是一项关键技术 6](#_Toc29631)

[1.15. Terraform 来实现代码化配置（as-code） 6](#_Toc21979)

[1.16. TRAEFIK 是一款开源的反向代理及负载均衡器。 6](#_Toc11711)

[1.17. MMKV 是微信开发的开源框架，为移动应用提供高速的键 值对存储。 7](#_Toc18271)

[2. 2018没落技术 7](#_Toc711)

[2.1. 数据饥饿软件包(DATA-HUNGRY PACKAGES)是一种解 决方案，它需要将数据吸附到自身才能运行 7](#_Toc13093)

[2.2. 低代码平台（LOW-CODE PLATFORMS）使用图形化用户 界面与图形化配置来创建应用程序。 7](#_Toc18420)

[2.3. MICROSERVICE ENVY特指那些盲目追赶微服务潮流的现象 8](#_Toc28349)

[2.4. 分层式微服 务架构（LAYERED MICROSERVICES ARCHITECTURE） 8](#_Toc1524)

[2.5. 机器人流程自动化（RPA） 8](#_Toc11276)

[3. Atitit 互联网2018风口与趋势大盘点 8](#_Toc2875)

[3.1. Atitit 未来 技术趋势 没落技术 attilax著 艾龙 总结 9](#_Toc31542)

# 2018 技术雷达

## HOSTED IDENTITY MANAGEMENT AS A SERVICE （SaaS）身份管理

身份管理是平台的关键组件之一。外部用户在使用移动 应用的时候，需要对其身份进行验证，开发人员需要被授 权才能访问基础设施组件，而微服务也需要向彼此证明自 己的身份。你应该考虑的是，身份管理是否真的有必要自 己来搭建和维护。根据我们的经验，HOSTED IDENTITY MANAGEMENT AS A SERVICE （SaaS）这种解决方案更 为可取。我们相信，像Auth0和Okta这样的顶级托管商可以 在“正常运行时间”和“安全”两方面提供更好的SLA。也就是 说，虽然有时候自行搭建和维护身份管理解决方案是一个 现实的选择，特别是对于那些有操作规范和资源的企业来 说，这个选择更为安全。但大型企业的身份解决方案通常包 含更广泛的功能，例如集中授权、治理报告和职责分离管理 等等。不过，这些担忧通常与员工身份更相关，特别是那些 受遗留系统限制的企业，尤其如此。

## POLYCLOUD策略，不再把所有业务全 “押在”一个服务供应商身上

组织们越来越习惯POLYCLOUD策略，不再把所有业务全 “押在”一个服务供应商身上，他们会根据自己的策略，把 不同种类的业务分配给不同的供应商。其中一些组织采用 了最佳的解决方案，比方说：把标准服务部署在AWS上，把 机器学习和面向数据的应用部署在Google，微软 Windows 应用则部署在Azure上。对于部分组织而言，这是一个关乎 文化和商业的抉择。比如，零售行业往往不愿意把数据放在 Amazon，他们会根据数据的不同分配给不同的供应商。“ 云不可知论”策略追求的是跨供应商的可移植性，这个代价 很大，并且会导致为迎合所有要求刻意而为的决策。与之不 同，Polycloud 策略更加注重选择每个供应商所提供的最 好服务。

## GRAPHQL FOR SERVER-SIDE RESOURCE AGGREGATION。GraphQL可 以让客户端直接使用特定的查询语句去访问BFF以获取数 据

在实践微服务的过程中，为了将后端资源进行聚合，我们 实践了一个又一个的模式。之前，我们经常使用类似于 Netflix Falcor这样的工具帮助我们实现BFF（Backend for Frontend ），现在很多项目已经开始使用GRAPHQL FOR SERVER-SIDE RESOURCE AGGREGATION。GraphQL可 以让客户端直接使用特定的查询语句去访问BFF以获取数 据。使用这项技术时，后端服务可以继续暴露 RESTful API， 而GraphQL可以轻易的将这些服务所提供的资源聚合在一 起，并且对客户端十分友好。我们推荐GraphQL是因为其简 化了BFF和其他聚合服务的实现。

## WEB BLUETOOTH能够直接从浏览器控制任意低功耗蓝 牙设备

。这样以前只能通过原生手机应用来处理的场景， 现在也可以适用了。该规范由 Web Bluetooth Community Group 发布，并且定义了一个 API，通过蓝牙 4.0 无线标准 发现设备并在设备间通信。 当前，Chrome 是唯一支持这个 规范的主流浏览器。借助Physical Web和 Web Bluetooth, 现在有了其他途径来让用户与设备进行交互, 且无需让他 们在手机上安装另一个应用。这是一个令人兴奋的领域, 值 得密切关注。

## Appium，它是最受欢迎 的移动端自动化测试框架之一。

在往期的技术雷达中，我们推荐过Appium，它是最受欢迎 的移动端自动化测试框架之一。 随着测试套件的扩大，在 多个设备上并行运行测试的能力成了缩短反馈循环的关 键。 APPIUM TEST DISTRIBUTION非常有效地解决了这 个问题，它能够并行运行测试，且能在多个设备上运行同 一套测试。它的出众之处在于在添加或移除测试设备时无 需手动配置，并且支持在远程设备上运行测试。过去几年 里我们在 ThoughtWorks 的一些项目中使用了这个工具， 效果很不错。 我们很开心使用 BACKSTOPJS 来做 web 应用的可视化 回归测试。作为可视化比较工具，它的可配置视窗和可调 节容错能力可以很容易定位到细微差别。它有很优秀的脚 本功能，并且可以选择在无界面Chrome、PhantomJS 和 SlimerJS 中运行。我们还发现，它在实时组件样式规范的 基础上运行时尤其有帮助。

## KOTLIN的使用率得到了飞速增长，

工具支持也突飞猛 进。其流行的背后原因包括语法简洁、空指针安全、易于从 Java迁移以及与其他JVM语言的互操作性。并且，它还是 非常不错的函数式编程入门语言。随着JetBrains 添加了 新功能，允许在多平台将 Kotlin 编译为原生二进制文件， 以及可以转译为JavaScript，我们相信，对于广大移动和原 生应用开发者来说，它具备进一步广泛使用的潜力。尽管 现在静态分析和代码覆盖率分析这样的工具还不成熟，但 基于我们在许多产品应用上使用Kotlin的经验，我们相信 Kotlin已经可以广泛采用。 自从第一次在雷达中提到GraphQL，我们就看到它在项目 中被稳定采用，特别是将它作为BFF的远程接口。随着使 用经验的提升，我们发现了一个GraphQL客户端 Apollo， 将其作为React应用程序访问GraphQL数据的首选。虽然 APOLLO项目也提供了服务端框架与 GraphQL 网关，但其 客户端简化了 UI 组件与 GraphQL 后端数据绑定的问题。 值得注意的是，Amazon AWS在最新启动的 AWS AppSync 服务中使用了Apollo。

## 多云 账号（MULTI-ACCOUNT CLOUD SETUP）

自助使用、按需付费是当代云计算重要的特性（和优势）。但 是，当使用单个账号去部署大规模的服务时，事情就变得非 常复杂。我们必须为该账户设置使用规则和流程，往往这些 流程包含大量的审批步骤，从而降低我们的效率。设置多云 账号（MULTI-ACCOUNT CLOUD SETUP）是一个不错的 方案，我们可以申请多个账号，并且为每个团队指定一个 账号。当然这个方案增加了其他方面的复杂度，例如共享计 费、不同 VPC 之间的通信、与云供应的关系管理等。但是， 也会提高我们的开发效率与安全性，审计只用于单服务的 账号相对容易，在发生数据泄漏时，我们所受到的影响也相 对较小。设置多云账号可以降低我们对云供应商的粘性，因 为单服务账号能提供清晰的边界，所以便于整体迁移到另 一个云提供商。 可观测性是运转分布式系统与微服务架构 必不可少的一部分。我们强烈建议使用代码 来配置可观测性生态系统，称为可观测性即 代码，并且采取基础设施即代码的方式搭建 其基础设施。 (Observability as code) 可观测性是运转分布式系统与微服务架

## 密钥即服务 （SECRETS AS A SERVICE）

我们长期以来都警示人们，要抵制在代码仓库中保存密 钥信息的诱惑。在往期雷达中，我们曾推荐了将代码和密 钥管理解耦，但如今我们发现了一系列提供密钥即服务 （SECRETS AS A SERVICE）的好工具。通过这些工具， 系统能够从外部服务中获取秘密信息，而不是使用硬编码 或将其配置在运行环境中的方式。例如来自HashiCorp的 Vault这样的工具帮助你将密钥与应用程序分开管理，同时 有助于你强制实施一些安全政策，例如周期性密钥轮换。

## Headless CMS （Content Management Systems, 内容管 理系统）

正在成为数字化平台的常见组件。CONTENTFUL 是一个现代化的 headless CMS。我们的团队已经成功把它 集成到开发工作流中。我们特别喜欢其“API 优先”的特点， 及其CMS as Code的实现。它支持强大的内容建模原语代 码和内容模型演化脚本，并允许将其视为其他数据存储的 schema，并将演进式数据库设计实践应用到 CMS 开发中。 我们所喜欢的其他特性包括：默认包含两个 CDN以提供多 媒体资源和 JSON文档，本地化的良好支持和与Auth0集成 的能力（尽管需要做出一些努力）。

## Cypress 端到端测试

运行端到端测试时经常会遇到一些棘手的问题，比如运行 时间过长，测试过于零碎，还需要修复无头模式下运行的 测试所导致的 CI 失败。我们的团队借助 CYPRESS 很好 地解决了性能差、响应时间长、资源加载慢等常见问题。 Cypress 是一款很有用的工具，可以帮助开发者构建端到 端测试，还可以将所有测试步骤保存为 MP4 视频，便于检 查错误。

## Firefox 无头模式(HEADLESS FIREFOX)

与用于前端测 试的 Chrome 无头模式一样成熟。 与 Chrome 无头模式类 似，在 Firefox 无头模式下运行浏览器测试时无需渲染 UI 组件，因此大大加快了 UI 测试的速度。

## MERMAID 使用类似 markdown 的标记语言来生成图表。

Mermaid 为简化文档编写而生，并且发展迅速。目前已经 为许多工具提供了插件支持，比如 Confluence 、 Visual Studio Code 和 Jekyll 。 可以试用一下 GitHub 上的在线编 辑器。此外， Mermaid 还提供了很好用的命令行，可以使用 图表定义文件生成 SVG/PNG/PDF 。我们已经在许多项目 中使用了 Mermaid ，尤其欣赏的是，它可以用 markdown 简洁地描述图形和流程图，同时也可以将图表定义文件存 储在代码仓库中。

## VISUAL STUDIO CODE 是微软推出的免费 IDE 编辑 器，

可以跨平台使用。我们曾用它同时进行前端 React、 TypeScript 和后端 GoLang 的开发，而无需在不同的编辑 器之间切换，体验很好。 Visual Studio Code 中的工具、语 言支持和扩展插件数量都在迅猛增长，也越来越好用。我 们要特别推荐在实时协作及远程结对编程时使用 VS Live Share 。固然微软或 Jetbrains 成熟的 IDE 对使用静态类型 语言（如 Java 、 .NET 或 C++ ）的复杂项目支持得更好，但 我们也发现 Visual Studio Code 正逐渐成为基础设施开发 组和前端开发组的首选工具。

## (Stanford CoreNLP) 越来越多的项目需要处理非结构化的数据，而从文本中 提取出有意义的业务信息是一项关键技术

越来越多的项目需要处理非结构化的数据， 而从文本中提取出有意义的业务信息是一 项关键技术。 Stanford Corenlp 是一组基于 Java 的自然语言处理（NLP）工具集。 (Stanford CoreNLP) 越来越多的项目需要处理非结构化的数据，而从文本中 提取出有意义的业务信息是一项关键技术。 STANFORD CORENLP 是一组基于 Java 的自然语言处理（NLP）工具 集，支持英语、汉语和阿拉伯语等多种语言的命名实体识 别、关系抽取、情感分析与文本分类，也提供了用于标记语 料库和训练模型的工具。 Stanford CoreNLP 协助我们使用 NLP 领域的最新研究成果来解决各种业务问题

## Terraform 来实现代码化配置（as-code）

。 我们广泛的应用 Terraform 来实现代码化配置（as-code） 云基础设施。TERRAGRUNT 是 Terraform 的一个轻量 级的封装，用来落地《 Terraform: Up and Running 》 书 中主张的实践。我们发现 Terragrunt 很有帮助，因为它通 过一些便利的特性来促进版本化模块和不同云环境的复 用性，这些功能包括递归执行子目录下的代码。我们希望 Terragrunt 能再进化一些，能够原生地支持持续交付的实 践，我们希望在持续交付的流水线上所有基础设施的代码 都能被打包、版本化并能在不同环境下复用（我们团队已经 通过替代方案实现了这些功能）。

## TRAEFIK 是一款开源的反向代理及负载均衡器。

如果 只是需要具有简单路由功能的边缘代理，而非 NGINX 、 HAProxy 这样的重量级产品，就可以考虑 Traefik 。它提供 了微服务所必不可少的免重载更新配置、度量、监控与断路 器等功能，同时也很好地集成了 Let’s Encrypt 以支持 SSL 。 相比于 Traefik ，为了扩展、添加或移除微服务， NGINX 、 HAProxy 等工具可能需要额外的工具进行模板化配置；有 时还需要重启，给生产环境造成很大麻

## MMKV 是微信开发的开源框架，为移动应用提供高速的键 值对存储。

它利用 iOS 的内存映射功能来避免直接保存修 改，因此性能极高。 MMKV 也能够在应用程序异常崩溃时 保存并快速恢复数据。

# 2018没落技术

## 数据饥饿软件包(DATA-HUNGRY PACKAGES)是一种解 决方案，它需要将数据吸附到自身才能运行

。在某些情况 下，它们甚至可能需要成为这些数据的“主人”。一旦这种软 件包拥有了数据，它就成为更新、修改或访问这些数据的唯 一方法。这种软件包可能会解决诸如ERP这样特定的业务 问题。但是，组织的有关库存或财务的“数据需求”，通常需 要复杂的集成和对位于原始范围之外的系统的更改。

## 低代码平台（LOW-CODE PLATFORMS）使用图形化用户 界面与图形化配置来创建应用程序。

然而不幸的是，该平台 的推广理念却是软件开发不再需要有经验的开发团队。这 种理念忽视了这样的事实，即在创建高质量软件所需要的 所有实践中，编写代码仅仅是其中的一小部分，而诸如控制 源代码、测试和精心设计解决方案这些实践，也同样重要。 尽管这种平台有时也是有用的，但我们仍建议要小心对待， 特别是当他们肆无忌惮地宣称其能实现更低成本和更高产 出的时候。

## MICROSERVICE ENVY特指那些盲目追赶微服务潮流的现象

，很多团队在实 践微服务时并没有简化其系统架构，大多数的实践方案只 是将一些简单的服务聚合在一起而已。目前，Kubernetes等 平台简化了复杂的微服务系统的部署问题，其他服务提供 商们也正在推进他们的微服务治理方案，这些强大的工具 都可能裹挟团队走上微服务之路。 但请千万谨记，微服务是 通过开发复杂度来换取运维复杂度，并需要自动化测试、持 续交付和DevOps文化提供坚实的支撑。

## 分层式微服 务架构（LAYERED MICROSERVICES ARCHITECTURE）

的组织，他们在某些方面存在着明显的矛盾。这些组织都 陷入了以技术角色为主来划分服务的误区，比如，用户体验 API、进程 API 或系统 API等。我们会很自然地将技术团队按 层划分，这会导致想要交付任何有价值的业务变更，都需要 缓慢而昂贵的多团队合作。请务必小心这种分层方式带来 的影响，我们更推荐根据业务能力来划分服务和团队。

## 机器人流程自动化（RPA）

是许多数字化转型计划的关键部 分，因为它有望在不必对底层架构和系统进行现代化改造 的情况下节省成本。 这种仅关注自动化业务流程而不解决 底层软件系统或功能的方法的问题在于，引入额外的耦合 会使底层系统更改起来更加困难。 这也会让未来任何解决 遗留IT环境的尝试都变得更加困难。 很少有系统能够忽视 变化，因此RPA的进展需要与适当的遗留系统现代化战略 相结合。

# Atitit 互联网2018风口与趋势大盘点

目录

1.1. 短视频 1

1.2. 互联网金融 1

1.3. 区块链 1

1.4. 新零售 1

2. 2018热度榜丨你能抓住的风口有几个？.html 1

## Atitit 未来 技术趋势 没落技术 attilax著 艾龙 总结

1. 2018技术趋势 2

1.1. 人工智能与区块链 2

1.2. 2、 PWA 或将大热 2

1.3. 5、 广义的 XaaS 2

1.4. 6、 容器技术越来越受欢迎 2

1.5. other 3

趋势1：正在日益增长的“数据化”（datafication）生活

趋势2：物联网（IoT），以及更加聪明的日常设备

趋势3：计算能力的指数增长正在推动技术的大发展

趋势4：人工智能（AI）的惊人崛起

趋势五：自动化，一辆不可阻挡的列车

趋势6：3D打印为制造商（和其他人）打开新世界的大门

趋势7：我们以不同的方式与技术进行互动

趋势8：区块链：一个可以改变我们的世界的发明

趋势9：平台是企业前进的方向

这十大科技分别是人工智慧基础、智慧应用程式与分析技术、智慧型物件、数位分身、从云端到边缘运算、对话式平台、沉浸式体验、区块链、事件驱动、持续性的适应风险与信任。为

1.智能助手之间展开对决

2.可以帮你在房里走动的机器人

3.能够陪伴你的机器人

4.健身技术可以让你无需去健身房

5.具有创新功能的巨型电视

6.无处不在的无线耳机

1.深度学习（DL）。机器

2.数字货币。比特币，以

3.区块链。比特币的使用

4.工业物联网。工业物联

5.机器人。尽管机器人研

6.辅助交通。尽管由于众

7.辅助现实和虚拟现实（

8.隐私，安全和责任的道

9.加速器和3D。随着功率

10.网络安全和人工智能