

STATE OF CLOUD NATIVE DEVELOPMENT Q1 2026

クラウドネイティブ開発の現状
2026年 第1四半期



CLOUD NATIVE
COMPUTING FOUNDATION

2026年3月

このレポートのデータを共有できますか？

1. 使用許諾

このレポートは[Creative Commons Attribution-NoDerivatives Licence 4.0 \(International\)](#)の下でライセンスされています。簡単にいえば、本ライセンスの条件に従って、次のことが自由に行えます。

共有 — 営利・非営利を問わず、本レポートを複製したり、本レポートの一部を1つまたは複数の文書や出版物に組み込んだりすることができます。

ただし、以下の条件に従うものとします:

帰属 — 本レポートのスポンサーである SlashData™ および Continuous Delivery Foundation のクレジットを適切に表示し、変更が加えられた場合はその旨を表示する必要があります。その場合、合理的な方法であればどのような方法でも構いませんが、SlashData™ があなたやあなたの使用を支持していることを示唆するような方法で行うことはできません。

改変禁止 — レポートの内容を編集したり、作り替えたりすることはできません。変更されたコンテンツを配布することも禁止されています。

2. 責任の制限

SlashData™ は、本出版物に含まれる記述が信頼できると考えられる情報に基づいていると考えていますが、それが正確または完全であることを表明するものではなく、そのようなものとして依拠すべきではありません。表明された意見は、本出版物に掲載された日付現在の意見であり、ここに含まれる意見を含む情報は、予告なしに変更されることがあります。どのような目的であれ、第三者による本書の使用は、当該第三者が本書の内容を確認する上で十分な注意を払うことを免責するものではなく、また免責されるものでもありません。SlashData™ は、商品性または特定目的への適合性を含むがこれに限定されない、すべての黙示的保証を放棄します。

SlashData™、その関連会社、および代理人は、この出版物に基づいて行われた、または行われなかった決定、または行われた、または行われなかったアクションの結果として、第三者が被った直接的、付随的、特別、または結果的な損害または逸失利益について、もしあれば一切責任を負いません。

[The analyst of the developer economy | formerly known as VisionMobile](#)
[SlashData Ě Copyright 2026 | Some rights reserved](#)

この日本語文書は、State of Cloud Native Development Q1 2026の参考訳として、The Linux Foundation Japanが便宜上提供しているものです。

翻訳協力：阿久津恵太



レポートのハイライト	<u>4</u>	
		<u>5</u> はじめに
クラウドネイティブ開発の トレンド	<u>9</u>	
		<u>30</u> クラウドネイティブの 成熟度向上への道
付録	<u>46</u>	
		<u>49</u> 調査方法



レポートのハイライト

- 2026年第1四半期時点で、世界中に1,990万人のクラウドネイティブ開発者がいる。→
- バックエンド開発者の71%が、クラウドネイティブコンピューティングに関連する技術またはプラクティスを少なくとも1つ利用している。→
- ハイブリッドクラウドは過去最高を記録し、開発者の34%がハイブリッドクラウドへデプロイしている。これは、データ主権要件と規制対応(コンプライアンス)が背景にあると考えられる。→
- バックエンド開発者の88%が、何らかの形でDevOpsの標準化に取り組んでいる。→
- 開発者がクラウドネイティブの成熟度を高めていく過程において、フィーチャーフラグは、主流のインフラプラクティスとより高度なプラクティスとをつなぐ重要なゲートウェイ技術として機能している。→
- これらの高度なプラクティス(カオスエンジニアリング、サービスメッシュ、イミュータブルインフラストラクチャ、マルチクラスタ管理)は、互いに密接に結びついた「エリートクラスタ」——高度な技術同士がセットで採用されるグループ——を形成している。これらの併用率は、各プラクティスが独立に採用されていた場合に想定される水準を大きく上回っており、同じ層の開発者がまとめて導入していることを示している。→
- AI開発者は異なる成熟経路をたどっており、イミュータブルインフラストラクチャは到達点ではなく、より高度なプラクティスへ進むためのゲートウェイとなっている。→

はじめに

01

クラウドへのデプロイは、ソフトウェア開発に欠かせない要素となっており、今や開発者の大多数が何らかの形でクラウドインフラに依存している。クラウドへのデプロイがほぼ普遍的な存在となった一方で、すべてのクラウド利用が同じレベルで高度化しているわけではない。クラウドインフラの普及は広範に及んでいるものの、多くの開発者や組織は、クラウドの真価を引き出すクラウドネイティブ技術やアーキテクチャパターンを十分に活用できていないのが現状である。

本レポートは、Cloud Native Computing Foundation(CNCF)との協業により作成されたものであり、2026年第1四半期におけるクラウドネイティブ開発者の最新動向に関する知見を提示する。SlashDataのDeveloper Nation Surveyの第31版(2025年12月から2026年1月にかけて実施、世界100か国から12,500名を超える回答者を獲得)のデータに基づき、クラウドネイティブエコシステムを形作る主要なトレンドを分析している。

「クラウドネイティブ」という用語は開発者によって異なる意味を持つため、本レポートでは、以下の特定技術のうち2つ以上を利用している開発者を「クラウドネイティブ開発者」と定義する。

対象となる技術は、コンテナ、コンテナオーケストレーション・管理ツール、サービスメッシュ、Kubernetes、クラウドファンクション(サーバーレスコンピューティング)、イベント駆動型アーキテクチャ、オブザーバビリティツール、イミュータブルインフラストラクチャ、カオスエンジニアリング、マルチクラスタ管理、ストリーミング・メッセージングサービスである。

¹バックエンド開発に携わる回答者には、調査において、より多くのクラウドネイティブ技術が提示される。そのため、バックエンド開発者は少なくとも3つのクラウドネイティブ技術を選択する必要があり、その他の回答者は少なくとも2つを選択する必要がある。

本レポートは、従来のバックエンドサービス領域の外側に存在するコミュニティをカバーする「拡張された」クラウドネイティブ層を含むシリーズの第2弾である。

そのため本レポートでは、幅広い開発領域におけるクラウドネイティブ開発者の割合の推計と、これがクラウドネイティブコミュニティの将来や、こうした新たな開発者層に向けた技術開発にどのような影響をもたらさうのかについて、引き続き提示していく。また、多様な開発領域にまたがる各種クラウド技術の普及状況についても考察する。拡張されたクラウドネイティブコミュニティの分析全体を通じて、機械学習および人工知能(AI)コミュニティに対して特に重点を置いている。これは、クラウドネイティブ技術がこの潮流に応えようとするなかで、理解すべき重要な層だからである。

本レポートに新たに加わったセクションでは、クラウドネイティブ領域における技術採用の経路と成熟への道のりを考察する。アソシエーション分析(共起分析)を用いて、どのクラウドネイティブ技術が併用される傾向にあるかを特定し、基盤的なツールから主流のプラクティス、さらには高度な信頼性・スケール管理能力に至るまで、明確に区分される進展段階を明らかにする。

本分析は、技術提供者やプラットフォームチームに対して、自然な採用順序とアーキテクチャ上の親和性に関する知見を提供するものである。



地域別のスポットライト: 本レポート全体を通じて、アムステルダムで開催される KubeCon + CloudNativeCon Europe 2026との連携のもと、ヨーロッパにおけるクラウドネイティブの採用パターンとトレンドに焦点を当てる。

クラウドネイティブ開発のトレンド

Q2

2. クラウドネイティブ開発のトレンド

クラウドネイティブ開発者コミュニティは拡大を続けており、2025年第3四半期の1,560万人から、2026年第1四半期には約1,990万人に達した。

これは世界の開発者人口全体のおよそ39%に相当し、クラウドネイティブのプラクティスがバックエンドインフラという出自を超えて、領域を横断する主流のソフトウェア開発へと広がっていることを示している。



1,990万

2026年第1四半期におけるクラウドネイティブ開発者

特にバックエンド開発者(歴史的にクラウドネイティブの中核コミュニティを担ってきた層)に着目すると、2026年第1四半期時点でクラウドネイティブに分類される割合は52%であり、2025年第3四半期の58%から低下したものの、2025年第1四半期に記録された49%は依然として上回っている。

前四半期比での減少は注目に値するが、これは通常の変動の範囲内である可能性もあれば、2025年第3四半期の58%が一時的に高めの数値であった可能性もある。2026年第3四半期の計測結果によって、典型的なベースラインが52%に近いのか、58%に近いのか、あるいはその中間にあるのかが明らかになるだろう。前年同期比での成長は引き続きプラスを維持している。

また、こうした数値の変動は、開発者の雇用市場が圧力を受けている時期に生じている点も留意すべきである。

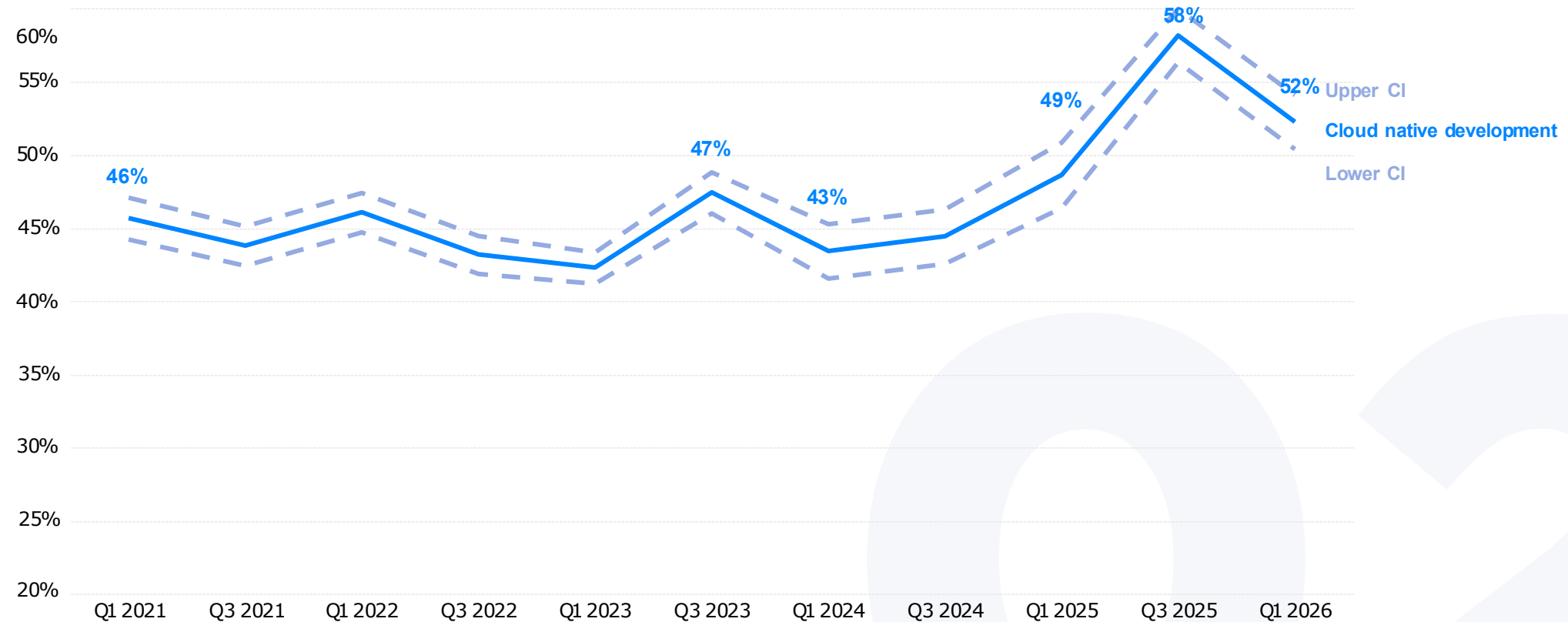
特に、2024年から2025年にかけて見られた業界再編(レイオフ・リストラクチャリング)の影響が顕著である。

市場の動向とクラウドネイティブの採用との直接的な因果関係を立証することはできないものの、クラウドネイティブ技術が当たり前のものとなった環境で教育を受けた開発者が、限定的あるいは抑制された規模でしか市場に参入できていないことが、この6か月間の低下の一因となっている可能性がある。

2. クラウドネイティブ開発のトレンド

クラウドネイティブバックエンド開発者の割合

バックエンド開発者の割合(%) (Q1 2021 (n=4,668) | Q3 2021 (n=5,260) | Q1 2022 (n=5,261) | Q3 2022 (n=5,619) | Q1 2023 (n=8,182) | Q3 2023 (n=4,847) | Q1 2024 (n=2,743) | Q3 2024 (n=2,754) | Q1 2025 (n=1,903) | Q3 2025 (n=2,791) | Q1 2026 (n=2,793))



設問文: 過去12か月間に、バックエンドサービス開発の一環として利用した技術は、以下のうちどれですか?

2. クラウドネイティブ開発のトレンド

クラウドネイティブ層の拡大は、特定の領域に偏ることなく、幅広く進展した。

たとえば、プロフェッショナル向けゲーム開発者では、クラウドネイティブの割合が2025年第3四半期の30%から2026年第1四半期には39%に増加し、産業用IoT(IIoT)開発者も同期間に38%から42%へと上昇した。

こうした増加は、クラウドネイティブ技術が、専門的なインフラ業務に限らず、多様な開発領域における標準的なツールとなりつつあることを示している。



2026年第一四半期時点で、西ヨーロッパには450万人、東ヨーロッパには120万人のクラウドネイティブ開発者が存在すると推計される。

機械学習・AI開発者コミュニティも同様の成長を示したが、その伸びはより緩やかであり、クラウドネイティブのAI開発者は2025年第3四半期の710万人から、2026年第1四半期には約730万人に増加した。

クラウドネイティブ技術がスケーラブルなAIオペレーション、分散学習、本番環境におけるモデルデプロイをますます支える存在となるなか、このセグメントは戦略的に重要な位置を占め続けている。

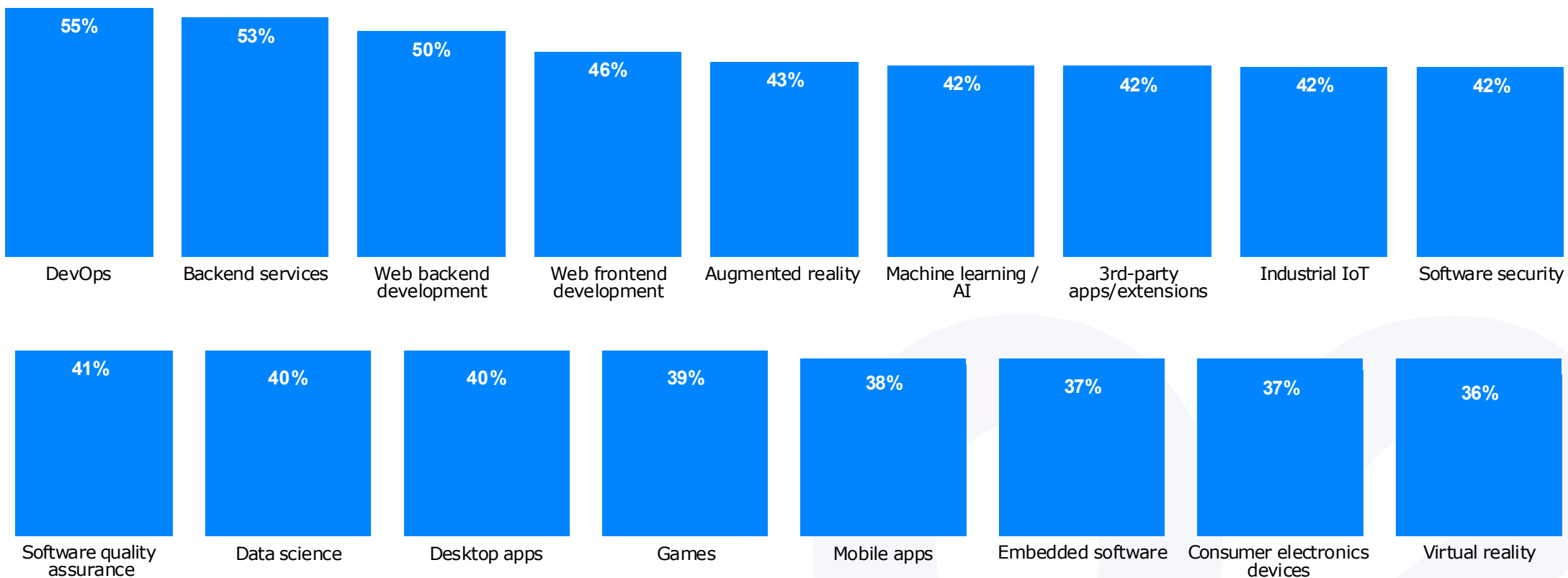


730万

クラウドネイティブを採用するAI開発者

各開発領域におけるプロフェッショナル開発者のうち、クラウドネイティブに該当する割合

各開発領域におけるプロフェッショナルなクラウドネイティブ開発者の割合(%) (Q1 2026 n=9,082)



設問文: 以下のような開発やプロジェクトに、どのように関与していますか? | 過去12か月間に、バックエンドまたはクラウドサービスの開発の一環として利用した技術は、以下のうちどれですか?

クラウドへのデプロイのトレンド

クラウドネイティブ開発者のトレンドに加えて、クラウドへのデプロイ選択が時間とともにどのように変化してきたかについても考察する。

開発者コミュニティ全体を見ると、過去5年間でハイブリッドクラウドの採用が大きく伸びていることがわかる。ハイブリッドクラウドへデプロイする開発者の割合は、2021年第1四半期の22%から、2026年第1四半期には34%に達した。

ハイブリッドクラウドは、中堅企業や、金融・ヘルスケアといった規制の厳しい業界において非常に人気の高い選択肢として浮上している。これは、多くの組織にとってますます魅力的、あるいは必要不可欠となっているセキュリティおよびコンプライアンス上の付加的なメリットをもたらすためである。

データ主権要件やプライバシー規制(欧州のGDPRや米国のCLOUD Actなど)が進化を続けるなか、クラウドのスケラビリティとオンプレミスでのデータレジデンシーの両立を求める組織にとって、ハイブリッド構成は魅力的な選択肢となっている。



2026年第一四半期において、全開発者のうち、ハイブリッドクラウドへデプロイした開発者の割合。過去最高の割合を記録した

バックエンド開発者の間でも、ハイブリッドクラウドの利用率は26%という新たな高水準に達した。

2023年第3四半期以降、23%前後で安定的に推移してきた状況からの上昇である。これまでこの安定的な推移は、バックエンド開発に携わる層がすでに数年前にハイブリッドクラウドへの移行を済ませており、その影響がようやくバックエンド層に依存する周辺の開発者コミュニティへと波及し始めた結果である可能性が指摘されてきた。

今回のバックエンド開発者における増加は、むしろ、ハイブリッド構成を未導入だった残りのバックエンドチームが、いまハイブリッドアプローチを採用し始めていることを示している可能性がある。これは、初期導入層を動機づけたものと同じ規制要因やセキュリティ要因によって後押しされている可能性がある。あるいは、これまで特定のワークロードに限定してハイブリッドインフラを導入してきた組織が、より多くのサービスへとハイブリッド採用を拡大している可能性も考えられる。



ヨーロッパの開発者の間では、プライベートクラウドが依然として最も多くの開発者(39%)に採用されているデプロイ方式であり、世界のその他の地域の平均(37%)をわずかに上回っている。

ハイブリッドクラウドは32%で2番目に多いデプロイアプローチとなっているが、その他の地域の平均(35%)を下回る水準にとどまっている。ハイブリッドクラウドがもたらすコンプライアンス上のメリットがあるにも関わらず、欧州の多くの組織は慎重な姿勢を保ち、プライベートクラウドおよびオンプレミスサーバーを引き続き利用している。

2. クラウドネイティブ開発のトレンド

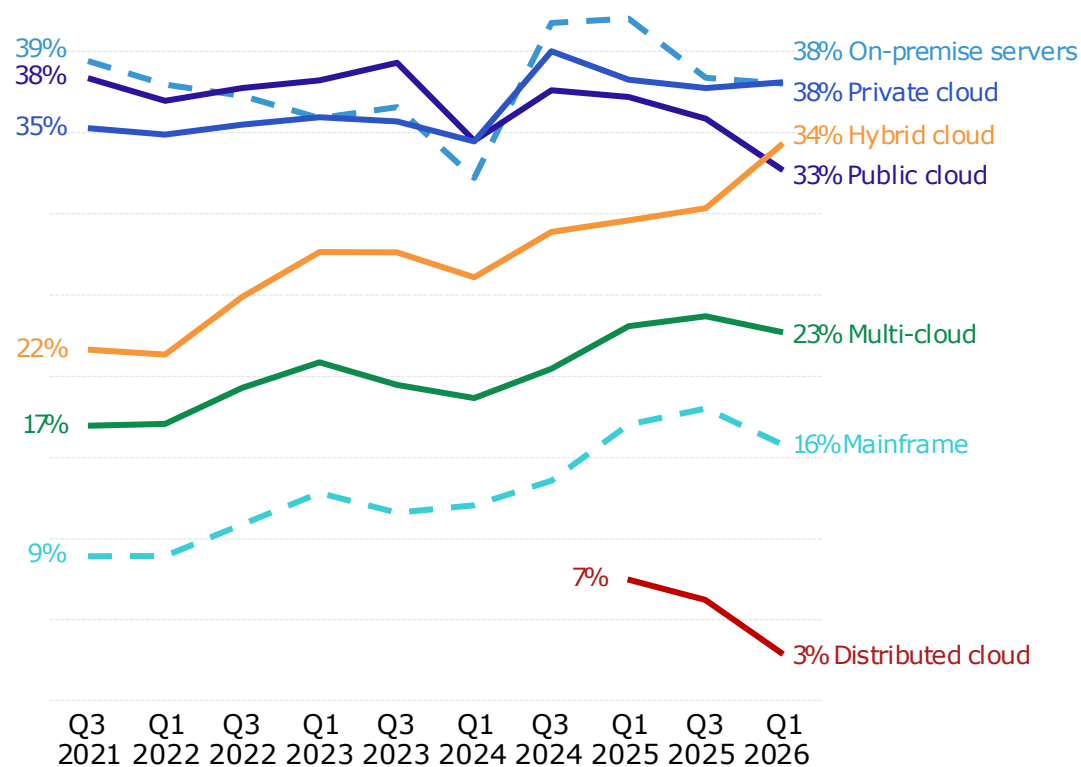
パブリッククラウドへのデプロイは、両方の集団において3パーセントポイント低下した(開発者全体では36%から33%へ、バックエンド開発者では33%から30%へ)。これは、ハイブリッドクラウドの成長と時期を同じくしている。横断的データから因果関係を断定することはできないものの、このパターンは、多くの組織がパブリッククラウドのみのデプロイを、ハイブリッドアーキテクチャによって補完あるいは部分的に置き換えていることを示唆している。その動機としては、コスト最適化、あるいは前述の規制およびセキュリティ上の懸念が考えられる。

2. クラウドネイティブ開発のトレンド

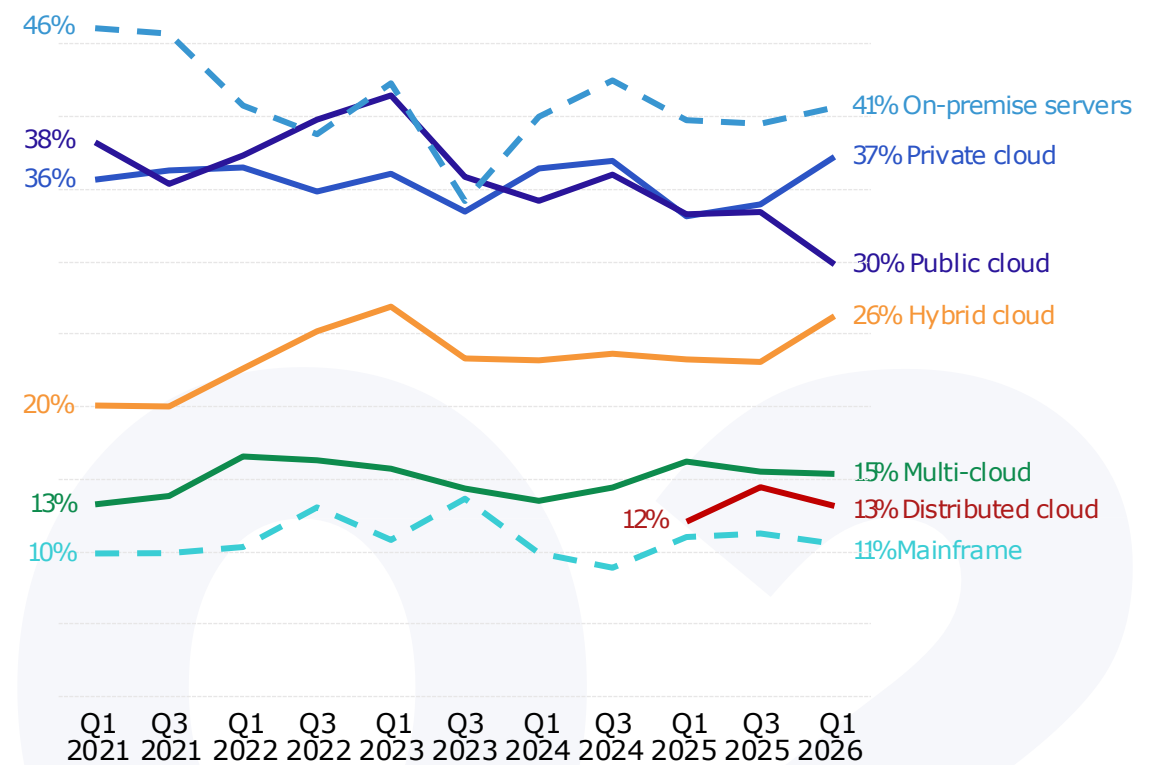
クラウドへのデプロイのトレンド

開発者の割合(%) | バックエンド開発者の割合(%) (21Q1 n=11,177 | 21Q3 n=12,132 | 22Q1 n=13,344 | 22Q3 n=17,820 | 23Q1 n=20,879 | 23Q3 n=10,521 | 24Q1 n=10,521 | 24Q3 n=9,886 | 25Q1 n=10,731 | 25Q3 n=12,021 | 26Q1 n=11,937)

開発者全体におけるクラウドデプロイのトレンド



バックエンドサービス開発者のデプロイメントトレンド



設問文: 過去12か月間に、バックエンドサービス開発の一環として利用した技術は、以下のうちどれですか?

クラウド関連の技術について

クラウドネイティブ開発は、開発者が利用する技術にとどまるものではない。とはいえ、クラウドネイティブ開発に関連する技術の利用状況を追跡することは、この領域がどのように進化しているかを把握するうえで重要な文脈を提供する。本セクションでは、バックエンド開発者におけるこれらの技術の採用状況に焦点を当てる。

最も顕著な変化は、コンテナ利用率の急激な低下(40%)である。2020年第3四半期以降、一貫して60%を上回って推移してきた水準からの落ち込みである。この減少は、コンテナ採用の劇的かつ予期せぬ後退に見えるかもしれないが、開発業界の動向を踏まえれば、そうではないことが明らかである。[本レポートの前版](#)で論じたとおり、この低下は、バックエンド開発者コミュニティをどのように定義するかという方法論上の変更を反映したものである。現在の定義では、従来のバックエンドインフラ開発者に加えて、Webバックエンド開発者——通常、基盤となるインフラを抽象化した高レベルのサービスを扱う層——が含まれている。この構成上の変化により、各サブグループ内での採用率自体は比較的安定したままであっても、報告されるコンテナ利用率は機械的に低下することになる。

¹ A modeling of the historic backend developers has a 58% adoption of containers.

その他の技術については、おおむね小幅な低下が見られる。ただし、サービスメッシュは例外であり、この6か月間で8%から11%へと増加した。とはいえ、母集団の変更が行われる前の過去最高値には依然として届いていない。他の技術が構成上の変化の影響を受けるなかでも、サービスメッシュの採用がこのように伸びていることは、採用が実質的に大きく拡大していることを示唆している。これは、サービスメッシュの導入コストに見合うだけのスケールと複雑性に到達した組織が増えていることが背景にある可能性がある。

技術採用に関する自己申告ベースの減少傾向は、バックエンド定義の拡張による構成上の変化が起こった後にも続いており、行動面の要因が影響している可能性を示唆している。プラットフォームエンジニアリングのプラクティスが広がるにつれて、開発者はコンテナ、Kubernetes、サービスメッシュといった技術を直接構成するのではなく、抽象化レイヤーを介してインフラと関わるが増えている。これにより、開発者はアプリケーションロジックやビジネス課題に集中できる一方で、基盤となるインフラはプラットフォームチームが管理する形となっている。



公式なプラットフォームやDevOpsプラクティスを導入していないバックエンド開発者の割合は20%から12%へと大幅に低下した。これはインフラの標準化と抽象化が進んでいることを示している。

個別のプラットフォームエンジニアリング手法はそれぞれ少数派の取り組みにとどまっているものの、それらを組み合わせて見れば、すでに広く普及する水準に達している。

現在、バックエンド開発者の88%が、何らかの形でのインフラ標準化に取り組んでおり、これは6か月前の80%から上昇している。具体的には、開発者体験(Developer Experience)チームを擁する開発者の割合は38%から42%へ、統合DevOpsシステムを利用している開発者の割合は36%から41%へと増加した。インフラ標準化を一切行っていない開発者の割合が8パーセントポイント低下したことは、コンテナ利用率の6パーセントポイントの低下、および他のクラウドネイティブ技術における同様の小幅な低下と時期を一にしている。

以前の「[DevOpsディバイド](#)」に関する調査でも論じたとおり、プラットフォームエンジニアリングが成功すると、インフラを運用する側(=技術を管理する者)と、アプリケーション開発者側(=簡素化されたインターフェース越しにそれらを利用する者)との間に分離が生まれる。ここで、計測上の問いが浮かび上がる。すなわち、開発者がDockerfileを書いたりレジストリを設定したりすることなく、IDP(内部開発者プラットフォーム)を介してコンテナ化されたアプリケーションをデプロイしている場合、その開発者は自らを「コンテナを利用している」と回答すべきだろうか？

標準化された環境の普及と、報告される技術利用率の低下が同時に進行していることは、たとえ基盤となるインフラ自体には変化がなくとも、インフラの抽象化が、開発者が自らの技術利用をどのように認識し、報告するかに影響を及ぼしていることを示唆している。

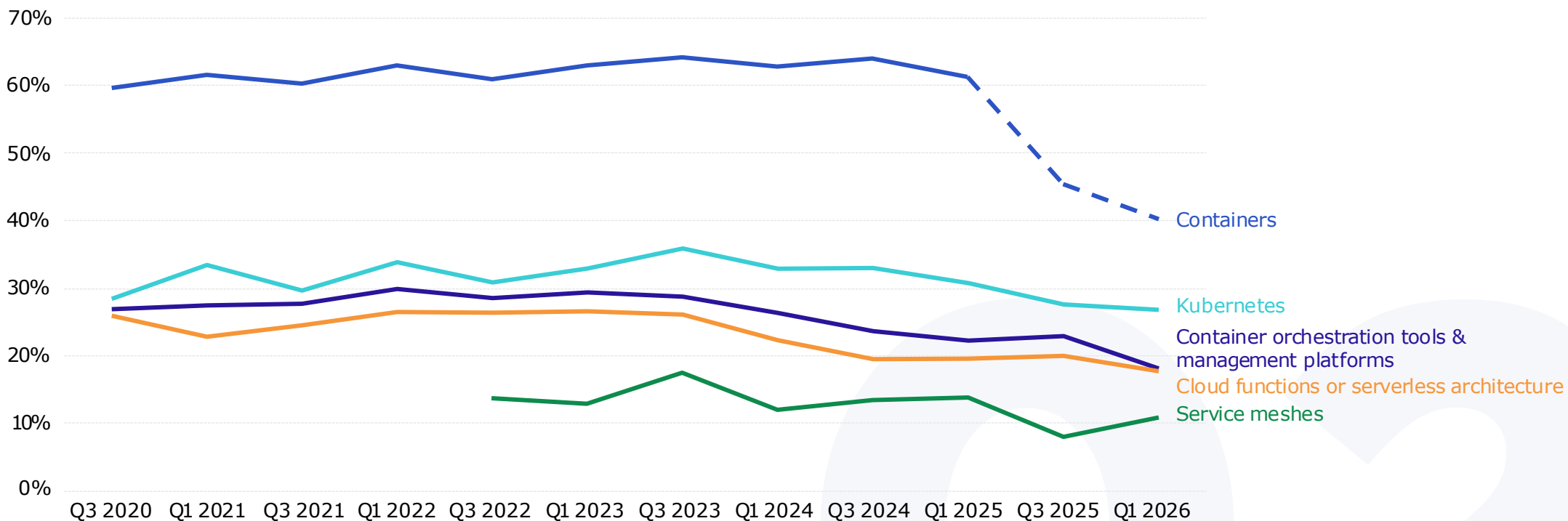


コンテナの利用率は、ヨーロッパのバックエンド開発者の間でより高く、西ヨーロッパでは45%、東ヨーロッパでは55%の開発者が、過去12ヶ月にコンテナを利用したと回答している。

東ヨーロッパで特にコンテナ利用率が高いのは、同地域がインフラ寄りの開発職やクラウドサービスプロバイダーにおいて存在感を示していることが反映されている可能性がある。これらの領域では、コンテナ化は標準的なプラクティスとなっている。

クラウドネイティブ開発に関連する技術の採用トレンド

バックエンド開発者の割合(%) (Q1 2021 n=4,668 | Q3 2021 n=5,260 | Q1 2022 n=5,261 | Q3 2022 n=5,619 | Q1 2023 n=8,182 | Q3 2023 n=4,847 | Q1 2024 n=2,743 | Q3 2024 n=2,754 | Q1 2025 n=1,903 | Q3 2025 n=2,666 | Q1 2026 n=2,792)



注: バックエンドサービス開発の性質が変化していること、および本調査における方法論上の捕捉対象が変更されたことに伴い、コンテナ利用率の大幅な減少は、バックエンド開発者の構成変化を反映したものである。従来の集計対象に基づく動向の推計値については、『State of Cloud Native』2025年第3四半期版を参照ください。

設問文: 過去12か月間に、バックエンドまたはクラウドサービスの開発の一環として利用した技術は、以下のうちどれですか？

バックエンド技術とそのアプローチ

本セクションの最後のパートでは、バックエンド開発に焦点を当て、インフラに使用される技術およびアプローチについて、より粒度の高い分析を行う。

このデータにより、本レポートにおいて初めて、それぞれの相対的な普及度を明確に把握することが可能となる。さらに、クラウドネイティブに分類される閾値を満たさない開発者——彼らの多くも個別のクラウドネイティブ技術を依然として利用している——も含めて、クラウドネイティブ技術やアプローチがどの程度広く採用されているかを把握することができる。

バックエンド開発者の71%が少なくとも1つのクラウドネイティブ技術を利用しているが、クラウドネイティブとして分類される基準(3つ以上の技術)を満たしているのは52%にとどまる。この19パーセントポイントの差は、多くの開発者がクラウドネイティブ採用の初期段階にあることを示している。

すなわち、1つまたは2つの技術を利用しているものの、成熟したクラウドネイティブ実践の特徴である複数技術スタックの活用には至っていない開発者層である。



39%がマイクロサービスを利用している一方で、カオスエンジニアリングやイミュータブルインフラストラクチャといった高度なプラクティスを採用しているのはわずか6-7%にとどまる

最も普及している技術はAPIゲートウェイで、バックエンド開発者の47%が利用している。次いで、最も多く採用されているアプローチはマイクロサービスであり、バックエンド開発者の39%が採用している。これは、APIゲートウェイとマイクロサービスがいずれも基盤的な性質を持っていることを反映している。

両者は、他の技術を利用する前提として必要とされることが多いほか、現代のサービススタックの機能を支える要素として不可欠だからである。

注目すべきは、APIゲートウェイとマイクロサービスが、通常はKubernetesインフラ上で稼働するアーキテクチャパターンであるにもかかわらず、Kubernetesを利用していると明示的に回答した開発者は27%にとどまる点である。

これは、開発者がマイクロサービスをKubernetes以外の環境にデプロイしている可能性、あるいは前述のとおり、Kubernetesがデプロイの基盤として動作していたとしても、プラットフォームの抽象化によって開発者からその存在が見えなくなっている可能性を示唆している。



東ヨーロッパの開発者は、プロジェクトにおけるマイクロサービスの採用率が57%と最も高く、フィーチャーフラグの利用率も最も高い水準にあると報告されている。このパターンは、同地域がインフラよりのエンジニアリング職に集中していること、およびクラウドサービスプロバイダのチームにおいて大きな存在感を持っていることを反映している可能性がある。これらの領域では、こうしたプラクティスが標準となる

前述のとおり、IDP(内部開発者プラットフォーム)の普及とDevOpsプロセスの中央集約化は、技術利用の自己申告を減少させる一因となっている可能性が高い。

この場合、頻繁にKubernetesに依存する技術を利用しているにもかかわらず、自らをKubernetesユーザーとは認識していない開発者が存在することになる。IDPの利用拡大とDevOpsプロセスの中央集約化は、組織のプラットフォームエンジニアリングが大きく顕著な成熟段階へと進んでいることを反映しているとはいえ、組織は依然として、開発者の生産性を支える基盤プラットフォームを管理・トラブルシューティング・最適化するためのインフラ専門知識を、チーム内に確保しておくべきである。

Kubernetes(27%)に次いで、利用が多い技術およびアプローチは、フロントエンドゲートウェイ(21%)、オブザーバビリティツール(21%)、イベント駆動型アーキテクチャ(21%)、ストリーミング・メッセージングサービス(20%)である。

これらはAPIゲートウェイやマイクロサービスと比較すると採用率は低いものの、より高度なアーキテクチャアプローチに位置づけられる。この統計は、バックエンドコミュニティの相当な割合が、より成熟したインフラアプローチに取り組んでいることを示している。

Kubernetesおよびコンテナは、デフォルトでイミュータブルインフラストラクチャを提供するが、本調査では開発者がイミュータブルインフラストラクチャの理念に基づくプラクティスや行動様式に取り組んでいるかを尋ねている。

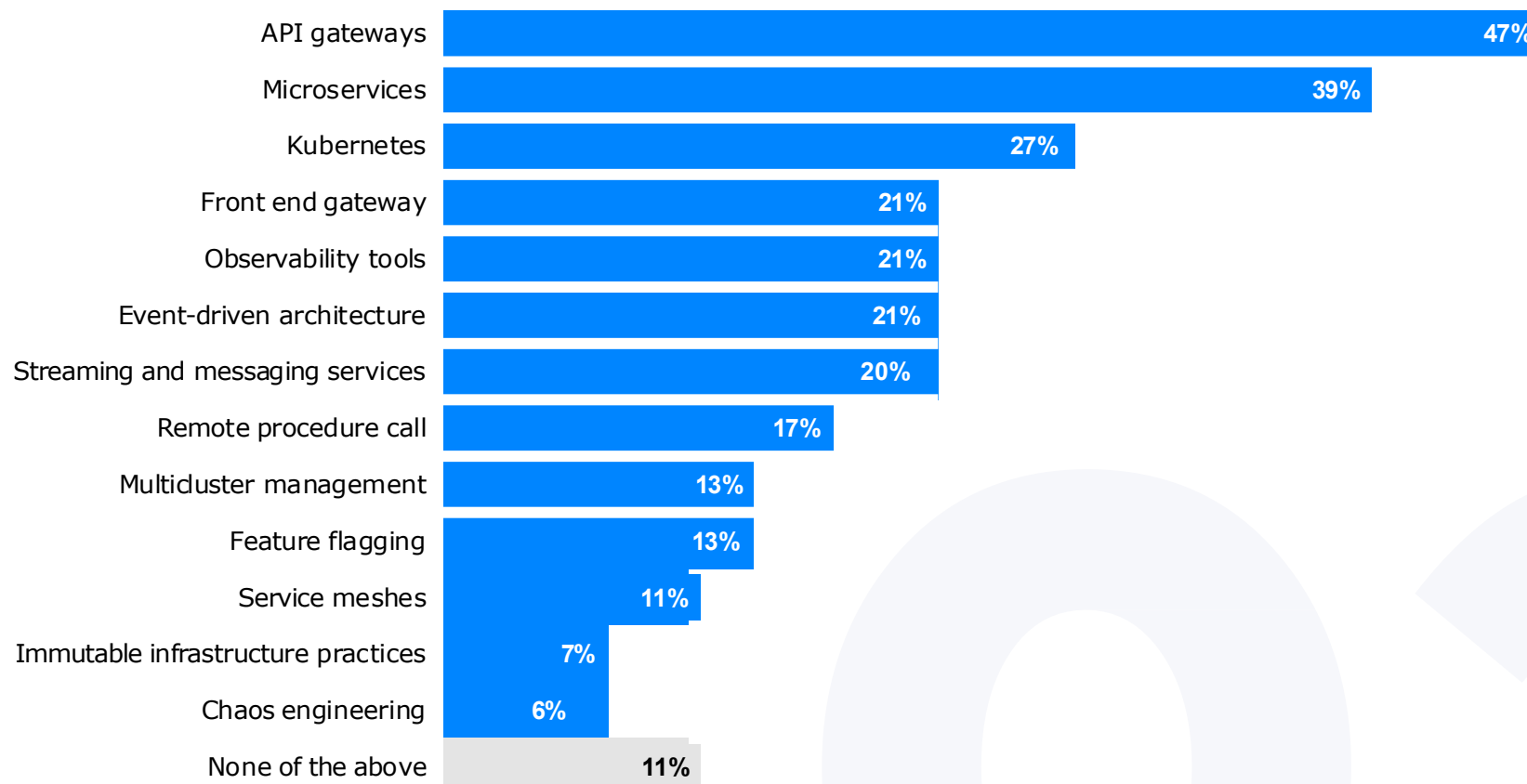
その結果、自らをイミュータブルインフラストラクチャの実践者と認識している開発者の割合は7%、カオスエンジニアリングについては6%にとどまり、いずれも低い採用率となっている。

この背景には、複数の要因があると考えられる。すなわち、多くの組織がこれらのプラクティスから明確なROI(投資対効果)を得られるだけの規模や複雑性に到達していないこと、技術的能力にとどまらず、組織的・文化的な成熟が導入には不可欠であること、そして、現状のツールが、典型的なチーム規模にとっての導入の手間に対して、得られるメリットとのバランス上、依然として摩擦を生じさせている可能性があることが挙げられる。

プラットフォームエンジニアリングがさらに成熟し、ツールが改善されるにつれて、これらのプラクティスは主流の開発チームにとっても、より身近なものとなっていく可能性がある。

バックエンド技術およびインフラアプローチの採用状況

バックエンド開発者の割合(%) (Q1 2026 n=2,792)



設問文: 過去12か月間に、バックエンドサービスにおいて、以下のインフラアプローチや技術のいずれかを利用しましたか？ 利用した場合、どれを利用しましたか？

クラウドネイティブの 成熟度向上への道

03

クラウドネイティブ技術は開発者コミュニティ全体へと拡大を続けており、クラウドネイティブ開発者人口は2026年初頭に約2,000万人に達した。

しかしながら、クラウドネイティブの普及状況を理解するには、2つの次元から検証する必要がある。すなわち、広がり(どれだけ多くの開発者がクラウドネイティブのプラクティスを採用しているか)と、深さ(その実装がどれほど高度であるか)である。広がりについては比較的計測しやすく、クラウドネイティブ技術を利用している開発者を数えればよい。

一方、深さの計測はより複雑である。なぜなら、より高度な技術の利用状況だけでなく、洗練されたアーキテクチャパターンを示す兆候にも着目する必要があるからだ。

すべての開発者が高度なクラウドネイティブのプラクティスを必要とするわけではない。基盤的なツールで十分なプロジェクトに取り組んでいる開発者も存在する。しかしながら、複雑で大規模なシステムを管理する開発者にとっては、クラウドネイティブの成熟への道のりを理解しておくことが、戦略的な技術選定の判断材料となりうる。

この課題を掘り下げるため、本レポートの今版では、開発者が技術採用において辿る経路をマッピングする試みを行う。

具体的には、Kubernetesやマイクロサービスといった主流のクラウドネイティブのプラクティスから、イミュータブルインフラストラクチャやカオスエンジニアリングといった高度なプラクティスへと進んでいく道筋を捉えることを目指す。

この分析にあたっては、技術同士の相対的な併用状況、そしてとりわけ重要な指標として、技術間の「リフト(lift)」を検証する。

リフトとは、2つの技術が偶然に期待される水準を超えて頻繁に併用されているかを測る指標である([「方法論」](#)セクション参照)。

言い換えると、リフトが1を上回る場合は正の関連性を、1を下回る場合は負の関連性を示す。

3. クラウドネイティブの成熟度向上への道

この分析からは、相互に補完しあう2つのアウトプットが得られる:

リフトマトリクス:すべての技術ペア間の関連性の強度を示す包括的な表であり、どの組み合わせが併用される傾向にあるかを明らかにする

開発者ジャーニーマップ:成熟への道のりを視覚的に表現したもので、組織が基盤的なクラウドネイティブのプラクティスから高度な運用パターンへと進んでいく過程で、技術がどのような順序で採用されていくかを示す。

こうした採用経路を理解することは、複数の関係者にとって価値をもたらす。

すなわち、**技術提供者**は、自社のツールへの入り口となる技術を特定できるようになる。プラットフォームチームは、自然な進展ルートに沿ったオンボーディング戦略を設計できる。そして組織は、自らの成熟度をベンチマークし、場当たりの技術採用を試みるのではなく、論理的な次の一手を見出すことが可能となる。

¹ The underlying network graph visualization is available in the [appendix](#).

開発者ジャーニー

一般的な開発者については、技術の成熟度を高めていく道筋として、2つの重要なプロセスおよび技術——フィーチャーフラグと、オブザーバビリティツール——を経由するパターンが浮かび上がる。

APIゲートウェイは最も多く利用されている技術であるものの、多くの技術との間でリフト値が約1.0となっており、これは特定のアーキテクチャパターンと結びついているというよりも、成熟度のレベルを問わず広く利用されていることを示している。

注目すべき例外として、APIゲートウェイはイミュータブルインフラストラクチャ(リフト値:0.79)およびカオスエンジニアリング(リフト値:0.53)との間で負の関連性を示している。

エリート実践クラスタに属する組織は、より高度な代替的トラフィック管理アプローチを選好している可能性があり、これは、エリート実践者がより主流なクラウドネイティブ採用層とは、アーキテクチャ面で異なる存在であることを示唆している。

代わって、開発者の間で最初に見られる関連性は、Kubernetesとマイクロサービスの併用である。

この組み合わせは、Kubernetesが本来設計された目的——コンテナ化されたマイクロサービスアーキテクチャをオーケストレーションすること——を反映している。マイクロサービスを採用する組織は、コンテナオーケストレーション、サービスディスカバリ、ロードバランシングといった機能を求めて、自然とKubernetesへと引き寄せられていく。これらの機能は、マイクロサービスの数が手動管理の限界を超えて拡大するにつれ、不可欠なものとなる。

3. クラウドネイティブの成熟度向上への道

Kubernetes利用者の間では、オブザーバビリティツール(リフト値:1.41)とストリーミング・メッセージングサービス(リフト値:1.33)の両方で、採用率が高まっている様子が観察される。このパターンは運用上の必然性を反映している。すなわち、Kubernetesデプロイメントが複雑性を増し—複数のサービス、ネームスペース、クラスタを管理するようになる—につれて、分散システムを監視するためにオブザーバビリティが不可欠となる。一方、ストリーミング・メッセージングサービスは、マイクロサービスアーキテクチャが疎結合性と回復性(レジリエンス)を実現するために必要とする、非同期通信のインフラを提供する。

また、ストリーミング・メッセージングサービスは、イベント駆動型アーキテクチャに対してもリフト値が上昇しており(リフト値:1.45)、これは想定どおりの結果である。ストリーミングプラットフォームは、イベント駆動型のパターンを大規模に実用化するためのインフラ基盤を提供する。イベント駆動型アーキテクチャがアーキテクチャ上のアプローチを表すのに対し、ストリーミング・メッセージングサービスはその技術的な実装を担うという関係にある。

次のつながりは最も重要なものである。

すなわち、フィーチャーフラグが、オブザーバビリティツール(リフト値:1.53)、ストリーミング・メッセージングサービス(リフト値:1.31)、そしてイベント駆動型アーキテクチャ(リフト値:1.45)との間で示すリフトである。

さらに、フィーチャーフラグは、イミュータブルインフラストラクチャ(リフト値:1.33)、マルチクラスタ管理(リフト値:1.40)、カオスエンジニアリング(リフト値:1.44)とも、利用上のリフトを示している。したがって、フィーチャーフラグは、クラウドネイティブサービスの成熟度を高めようとする開発者にとって、鍵となりうる重要なゲートウェイ技術アプローチとして機能している。

フィーチャーフラグが主流のプラクティスと高度なプラクティスをつなぐ橋渡しとなるのは、運用上の重要な能力を2つもたらすからである。

すなわち、コードの変更をデプロイしても、ユーザーには即座に公開せずに済むようにする能力(デプロイとリリースの分離)と、変更を一部のトラフィックに対して段階的にテストする能力である。これらの能力は、組織にプログレッシブデリバリーのパターンと、制御された実験(コントロールド・エクスペリメンテーション)というメンタルモデルを身につけさせる。そしてこれらのメンタルモデルは、カオスエンジニアリング(制御された障害注入)やイミュータブルインフラストラクチャ(デプロイメントを使い捨ての実験として扱う考え方)といった高度なプラクティスに直接転用できるものである。

フィーチャーフラグは、本来であれば導入を躊躇するような洗練された運用技術への、低リスクな入り口として機能している。

高度なプラクティスへ至るもう1つの経路は、オブザーバビリティツールを経由するルートであり、これはイミュータブルインフラストラクチャとの関連性(リフト値:1.33)を示している。このルートはおそらく、包括的なオブザーバビリティに投資し、その結果として、イミュータブルインフラストラクチャを自信を持って採用するために必要な可視性を獲得した組織を反映している。オブザーバビリティは、「使い捨てて再デプロイする」戦略を実現可能にするための、セーフティネットを提供するのである。

開発者がいったん高度なプラクティスを採用すると、技術間に著しく高いリフト値が観察される。すなわち、イミュータブルインフラストラクチャとカオスエンジニアリング(2.22)、サービスメッシュとカオスエンジニアリング(1.76)、イミュータブルインフラストラクチャとサービスメッシュ(1.75)である。これらの値は、基盤的な技術間の関連性と比べて大幅に高く、組織が複数の高度なプラクティスを段階的にではなく同時並行的に採用する「エリートクラス」の存在を示している。この際立って高い関連性(2.22—データセット中で最も強い値)は、技術的な依存関係(イミュータブルインフラストラクチャがあれば、失敗した実験を迅速に置き換えられるため、カオスエンジニアリングをより安全に実施できる)と、理念的な親和性(両方のプラクティスとも、インフラを使い捨て可能なものとして扱い、失敗を学習の手段として受け入れる)の双方を反映していると考えられる。一方を採用した組織は、もう一方にも自然と引き寄せられていく可能性がある。

3. クラウドネイティブの成熟度向上への道

一般的な開発者については、データから3層の成熟度モデルが浮かび上がる:

基礎: Kubernetes + マイクロサービス (学習のエントリーポイント)

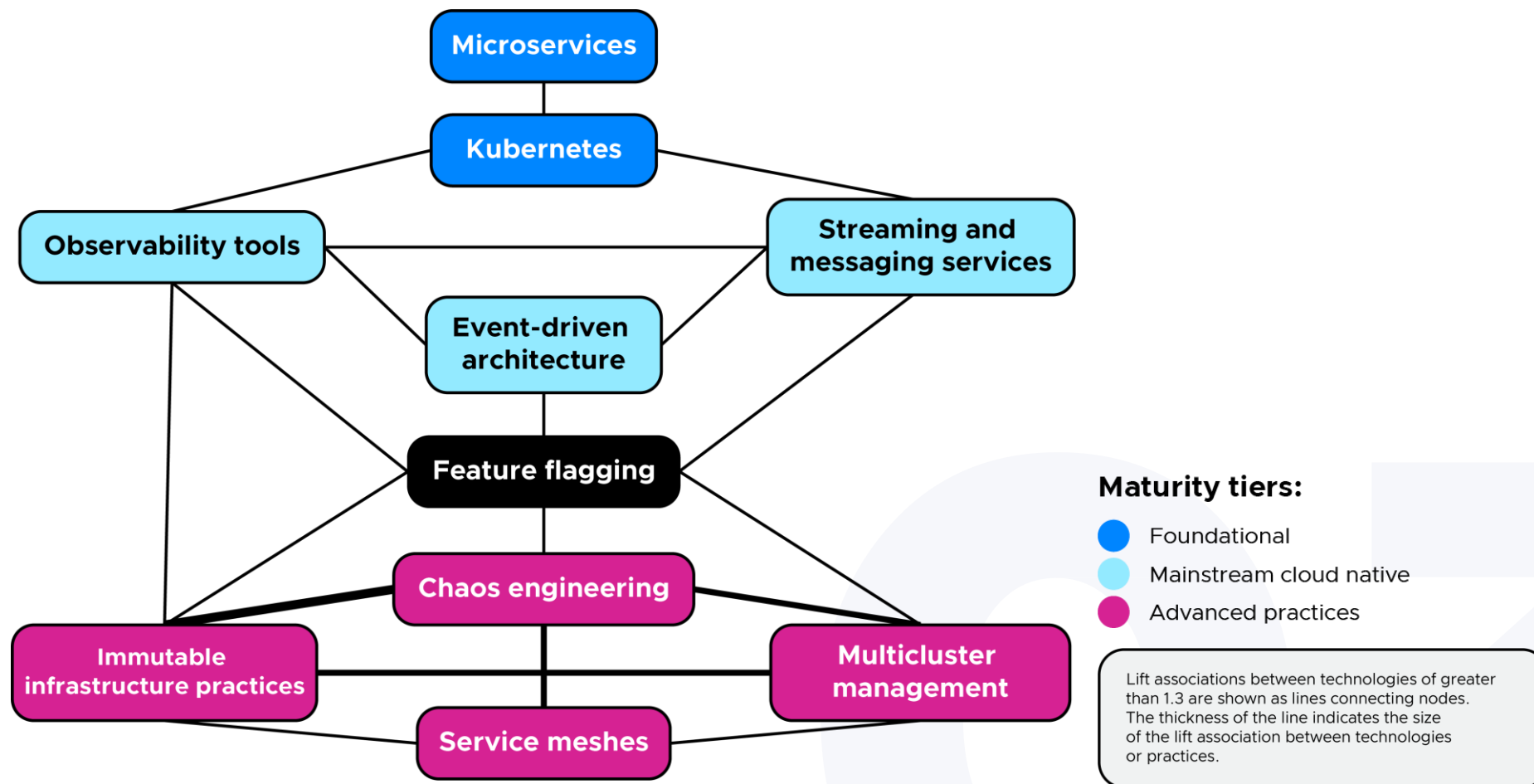
主流のクラウドネイティブ: オブザーバビリティツール、ストリーミング/メッセージング、イベント駆動型アーキテクチャ(運用上の高度化)

高度なプラクティス: イミュータブルインフラストラクチャ、カオスエンジニアリング、サービスメッシュ(エリートレベルの運用成熟度)

フィーチャーフラグは、左記の第2層と第3層をつなぐ重要な橋渡しとして機能しており、2つの経路が並行して存在する。すなわち、組織は、実験における成熟(フィーチャーフラグ)と、運用上の可視性(オブザーバビリティツール)のいずれかを通じて、高度なプラクティスに到達できる。組織がいったん第3層に達すると、複数の高度なプラクティスを段階的にではなく同時並行的に採用する傾向があり、これは、これらのプラクティスが個別の機能ではなく、一貫した運用哲学を形成していることを示唆している。

開発者のクラウドネイティブ・ジャーニー

バックエンド技術間のリフト関連性



設問文: 過去12か月間に、バックエンドサービスにおいて、以下のインフラアプローチや技術のいずれかを利用しましたか？ 利用した場合、どれを利用しましたか？

3. クラウドネイティブの成熟度向上への道

バックエンド技術間のリフト関連性

バックエンド技術間のリフト関連性 (Q1 2026 n=2,792)

Microservices	1.39											
Service meshes	1.04	0.87										
API gateways	1.08	1.13	0.77									
Event-driven architecture	1.27	1.17	1.24	1.08								
Observability tools	1.41	1.11	1.28	0.98	1.27							
Immutable infrastructure practices	1.12	0.83	1.75	0.79	1.15	1.33						
Chaos engineering	0.66	0.68	1.76	0.53	0.77	0.96	2.22					
Feature flagging	1.19	1.01	1.27	0.97	1.45	1.53	1.33	1.44				
Remote procedure call	0.90	0.87	0.96	1.00	0.95	0.89	0.98	0.88	1.16			
Front end gateway	1.04	0.92	1.09	1.17	0.89	0.97	1.14	0.82	1.19	1.03		
Multicloud management	0.96	0.84	1.42	0.86	1.08	0.91	1.68	1.64	1.40	1.28	1.04	
Streaming and messaging services	1.33	1.15	1.26	0.97	1.45	1.40	1.11	0.95	1.31	1.15	1.06	1.08
	Kubernetes	Microservices	Service meshes	API gateways	Event-driven architecture	Observability tools	Immutable infrastructure practices	Chaos engineering	Feature flagging	Remote procedure call	Front end gateway	Multicloud management

設問文: 過去12か月間に、バックエンドサービスにおいて、以下のインフラアプローチや技術のいずれかを利用しましたか? 利用した場合、どれを利用しましたか?

クラウドネイティブにおける機械学習ジャーニー

開発者ジャーニーの主たる焦点は、より広範な開発者エコシステムがクラウドネイティブの成熟度をどのように高めていけるかを理解することにあるが、本レポートでは機械学習開発者にも焦点を当てる時間を設ける。

機械学習および人工知能に対する現在の強い注目、とりわけクラウドネイティブがこうしたワークロードを追求する組織をいかに支えうるかという観点を踏まえると、組織がクラウドネイティブの能力拡張に取り組むなかで、この経路に焦点を当てることは重要である。

全体として、経路と成熟度は概ね類似しているものの、明確かつ注目すべき相違点が存在し、それらは大きく異なる思想的フレーミングと技術的動機を浮き彫りにしている。

AI開発者にとって、成熟への道のりは、純粋なクラウドネイティブの高度化というよりも、AIサービスのライフサイクル段階を表すものとなっている。一般的なバックエンド開発者が、基盤層から主流層、そしてエリートクラウドネイティブ層へと進んでいくのに対し、AIチームは、コンテナ化からデータパイプライン運用、続いて学習・実験フェーズの運用へと進み、最終的に本番サービング運用へとスキルアップを図っていく。

学習・実験段階で採用されるフィーチャーフラグとイミュータブルインフラストラクチャは、本番サービングの成熟度に到達するための2つの必須要件となる。これは、一般的な開発において、イミュータブルインフラストラクチャ自体がエリート層の採用を表すパターンとは、明確に異なる構図である。

最も顕著な違いは2つある。すなわち、イミュータブルインフラストラクチャが橋渡し技術へと位置づけを変えていることと、AI開発者の間でリモートプロシージャコール(RPC)の重要性が高まっていることである。

AI開発者がクラウドネイティブスタックを成熟させていく過程において、イミュータブルインフラストラクチャがより重要な要件として位置づけ直されているのは、機械学習における再現性の必須性を反映している。

一般的な開発においては、イミュータブルインフラストラクチャはカオスエンジニアリングとの関連性が最も強く(リフト値:2.22)、これはエリートレベルの信頼性プラクティスを示している。一方、AIにおいては、イミュータブルインフラストラクチャはオブザーバビリティツールとの間でより強い関連性を示し(1.58、一般開発の1.33に対して)、カオスエンジニアリングとの関連性は弱まる(1.28、一般開発の2.22に対して)。

AI開発者の間では、イミュータブルインフラストラクチャはおそらく二重の役割を果たしている。すなわち、再現可能な学習環境の確保と、本番サービング向けの一貫したデプロイメント成果物の提供である。

AIチームは、クラウドネイティブのプラクティスを模索する過程で、一般的な開発者よりも早い段階でイミュータブルインフラストラクチャを採用している可能性があり、これが、カオスエンジニアリングとの関連性が弱い理由を説明しうる。

多くのAIチームは依然として実験段階にあり、その段階では再現性は重要であるものの、カオステストは重要ではないからである。

リモートプロシージャコール(RPC)のパターンは、一般的なバックエンド開発ではほぼ目立たない存在であるものの、AIワークフローにおいては独自に重要な要素として浮上している。RPCは2番目に強い関連性を示している。すなわち、RPCとフィーチャーフラグの間のリフト値は2.07(一般開発の1.16に対して)である。これは、モデル推論API、分散学習、そして特徴量ストアへのアクセスにおいて、RPCが中心的な役割を担っていることを反映していると考えられる。

フィーチャーフラグとの並外れた関連性は、AIチームがモデルバージョンのルーティングといった動作において、フィーチャーフラグへの依存度をより高めている可能性を示している。RPCが数あるコミュニケーションパターンの1つにすぎない一般的なバックエンド開発とは異なり、AIモデルをサービングする場面では、同期型のリクエスト・レスポンスがより強く求められる傾向があり、RPCの採用が本番AIチームの技術スタックにおける重要な構成要素となっている。

これらを総合すると、AI開発者の成熟への道のりを別の観点から捉え直すことができる。

クラウドネイティブの領域に参入すると(すなわち、Kubernetesとマイクロサービスを利用し始めると)、AI開発者はデータパイプライン運用の改善方法を模索する。これは、イベント駆動型アーキテクチャ(Kubernetesに対するリフト値:1.31)、ストリーミングサービス(イベント駆動型アーキテクチャに対するリフト値:1.50)、そしてオブザーバビリティツール(リフト値:1.25および1.34)に現れている。

こうした傾向は、AI特有のニーズから生まれている。すなわち、イベント駆動型の仕組みが再学習パイプラインをトリガーし、ストリーミングが継続的な特徴量エンジニアリングを担い、オブザーバビリティがデータ品質とモデル性能を監視するためである。注目すべきは、オブザーバビリティがカオスエンジニアリングとの間で負の関連性(リフト値:0.50)を示している点であり、これは、AIにおけるオブザーバビリティが、インフラの回復性(レジリエンス)ではなく、データやモデルのメトリクスに焦点を当てていることを裏付けている。

次の段階では、学習・実験フェーズの運用を改善することに焦点が当てられる。

この段階では、フィーチャーフラグ、RPC、そしてイミュータブルインフラストラクチャが、多くのAIチームにとって自然な到達点となりうる。これらの技術およびアプローチは、より高度な本番運用の洗練度を必要とすることなく、成熟したAIワークフロー——再現可能な実験、バージョン管理されたモデル、基本的なサービング——を支えるからである。

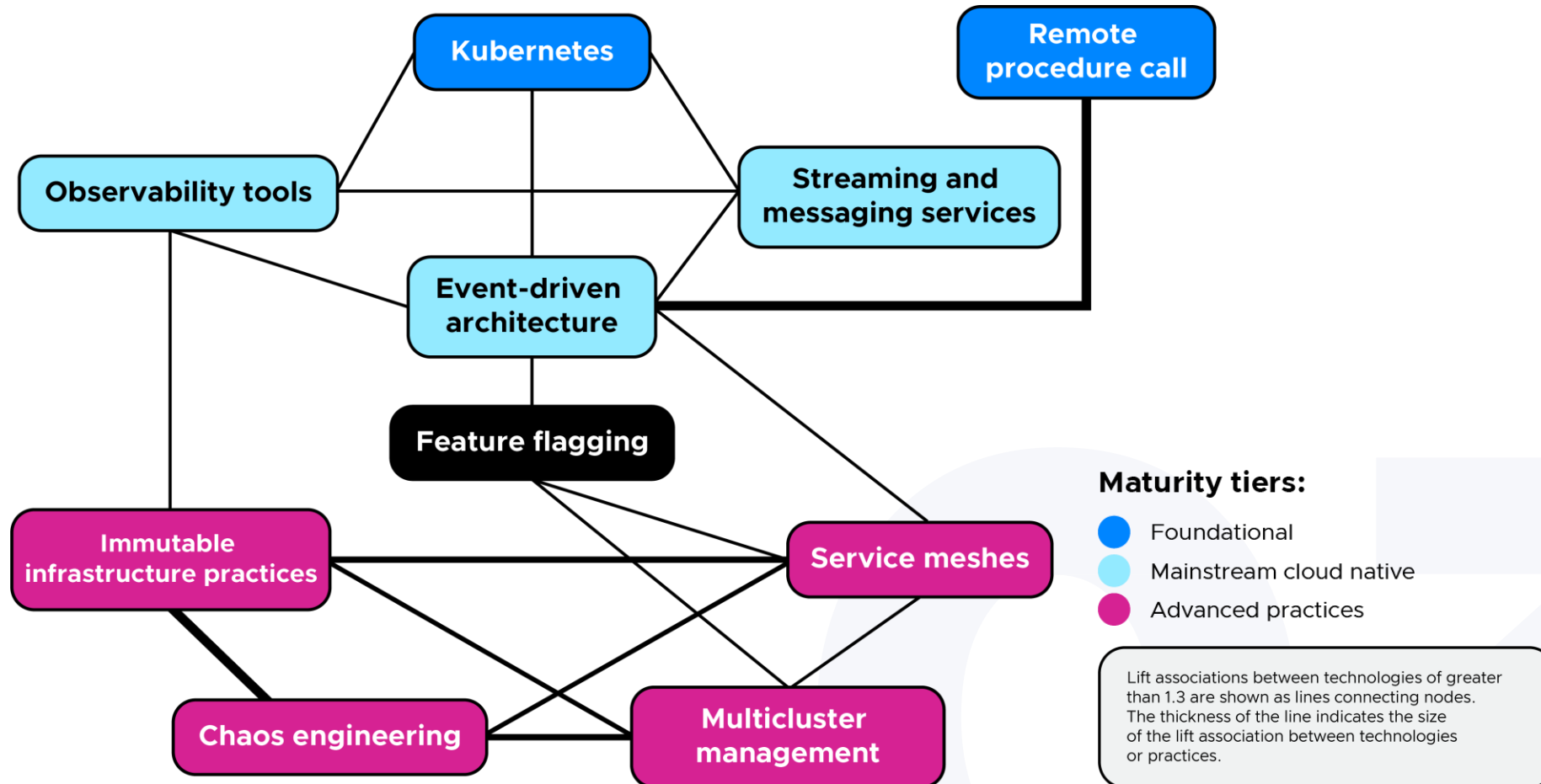
拡張された道のりは、サービスメッシュ、カオスエンジニアリング、マルチクラスタ管理を中心に形成される、密接に結びついたエリートクラスタで結実する。

サービスメッシュはトラフィック分割を提供し、カオスエンジニアリングはモデル劣化のテストを担い、マルチクラスタは分散推論を可能にする。この高度な本番運用クラスタに到達するには、前段階で挙げたフィーチャーフラグとイミュータブルインフラストラクチャの両方が必要であり、これが両者の間の相対的なリフト値の高さを示唆している。

3. クラウドネイティブの成熟度向上への道

AI開発者のクラウドネイティブ・ジャーニー

AI開発者におけるバックエンド技術間のリフト関連性(Q1 2026 n=1,270)



設問文: 過去12か月間に、バックエンドサービスにおいて、以下のインフラアプローチや技術のいずれかを利用しましたか? 利用した場合、どれを利用しましたか?

AI開発者におけるバックエンド技術間のリフト関連性

AI開発者におけるバックエンド技術間のリフト関連性(Q1 2026 n=1,270)

Microservices	1.22											
Service meshes	0.84	0.79										
API gateways	1.09	1.11	0.92									
Event-driven architecture	1.31	1.25	1.38	1.15								
Observability tools	1.25	1.06	1.05	0.99	1.25							
Immutable infrastructure practices	1.15	0.89	2.07	0.92	0.81	1.58						
Chaos engineering	0.78	0.81	2.36	0.50	0.84	0.50	1.28					
Feature flagging	1.26	0.99	1.35	1.02	1.45	1.41	1.10	1.10				
Remote procedure call	1.11	1.13	0.69	1.17	1.01	0.94	1.19	0.59	2.07			
Front end gateway	0.87	1.02	0.83	1.16	0.80	0.79	1.02	0.82	0.87	0.93		
Multicluster management	0.90	0.89	1.62	0.82	1.04	0.81	1.05	2.03	1.48	0.85	0.79	
Streaming and messaging services	1.21	1.23	1.18	0.97	1.50	1.34	1.13	0.97	1.53	1.19	1.07	0.99
	Kubernetes	Microservices	Service meshes	API gateways	Event-driven architecture	Observability tools	Immutable infrastructure practices	Chaos engineering	Feature flagging	Remote procedure call	Front end gateway	Multicluster management

設問文: 過去12か月間に、バックエンドサービスにおいて、以下のインフラアプローチや技術のいずれかを利用しましたか? 利用した場合、どれを利用しましたか?

AIチームの支援

成熟への道のりに関するこのモデルを踏まえると、クラウドネイティブ領域におけるAI開発者を支援するうえで、AIのクラウドネイティブ採用を後押しするために、以下の戦略的優先事項が浮かび上がる::

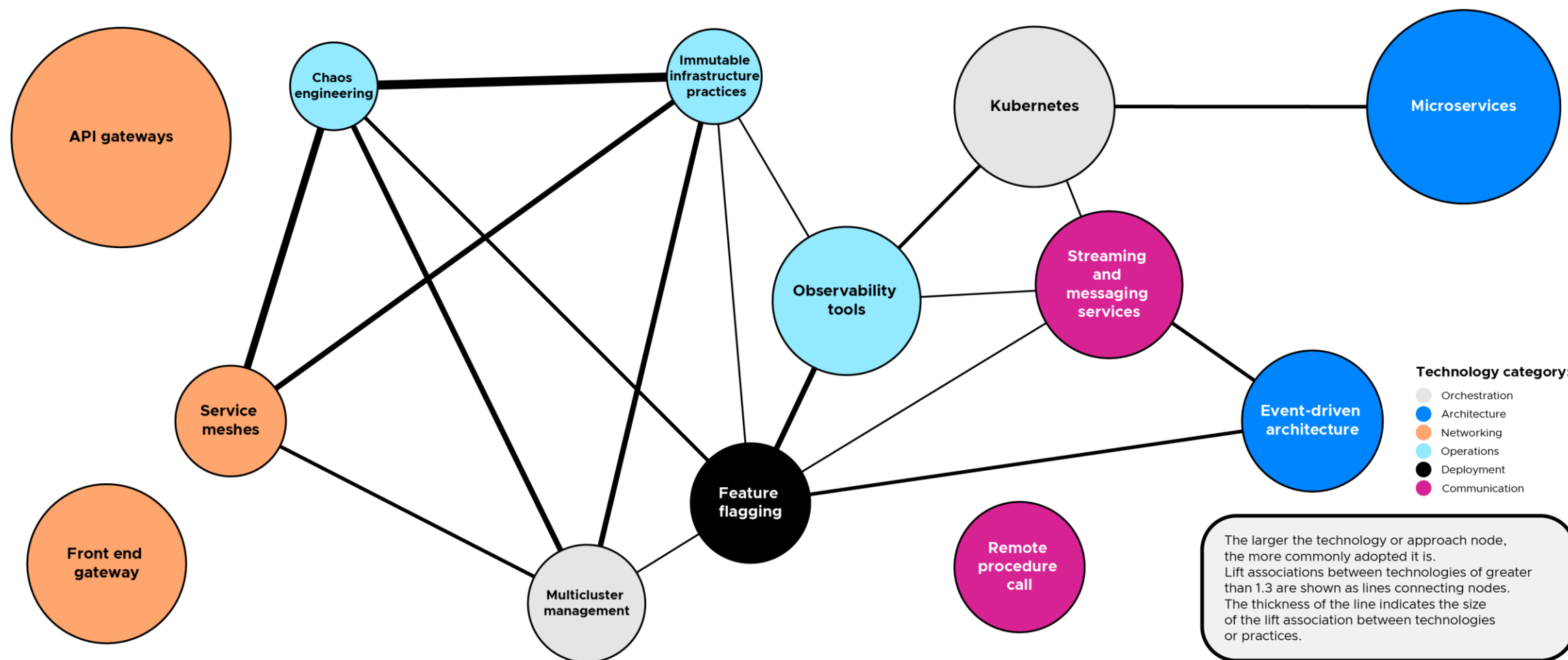
- すべてのAIチームが、本番運用レベルの洗練度を必要とするわけではない。
多くのチームは、実験・学習フェーズにとどまることに正当な理由があり、高度なサービングインフラよりも、データパイプライン関連のツールの改善と再現性プラクティスから最も大きな恩恵を受けている。
- フィーチャーフラグとイミュータブルインフラストラクチャは、本番運用AIに到達するための2つの必須要件である。これらの中間段階の技術に関する教育を優先することは、AIチームが本番運用レベルの成熟度に到達することを妨げている主要な障壁に対処することにつながる。
- サービスメッシュ、カオスエンジニアリング、そしてマルチクラスタ管理は、AI特有のフレーミングを必要とする。すなわち、モデルのA/Bテストのためのトラフィック分割、モデル劣化に対する回復性テスト、そして地理分散型推論である。これらは、汎用的なインフラ上のメリットとは異なる文脈で位置づけられるべきものである。
- AIモデルの本番運用におけるRPCの中心的な役割は、AIのクラウドネイティブ教育において重点的に取り上げる価値がある。モデル推論APIは、本番運用レベルの成熟度を支える基盤的な要素だからである。



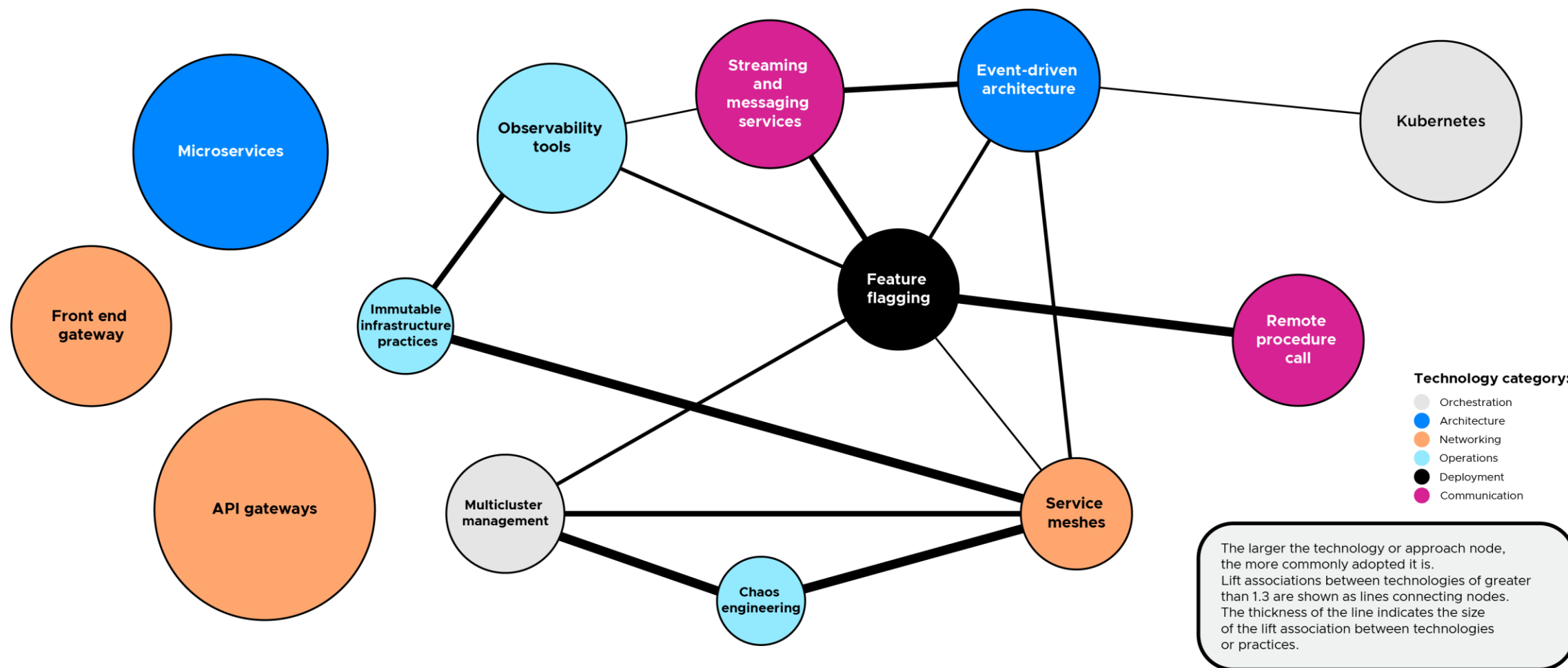
付録



バックエンド技術を利用する開発者間の技術リフトのネットワークグラフ



バックエンド技術を利用するAI開発者間の技術リフトのネットワークグラフ





調査方法





調査方法

リフト値の算出

概要

リフト(リフト比または関心度ファクターとも呼ばれる)は、2つの変数それぞれの単独での出現頻度を考慮したうえで、両者の関係の強さを測定するアソシエーションルール(関連性ルール)の指標である。技術採用の文脈においては、リフトは、両方の技術が単に人気があるという理由で同時に現れているのか、それとも一貫したアーキテクチャパターンの一部として真に併用されているのかを明らかにする。

リフトの数学的定義

技術Aと技術Bの間のリフトは、以下の式で算出される:

$$Lift(A,B) = P(A \cap B) / [P(A) \times P(B)]$$

ここで:

- $P(A \cap B)$: 結合確率——技術Aと技術Bの両方を利用している開発者の割合
- $P(A)$: 周辺確率——技術Aを利用している開発者の割合
- $P(B)$: 周辺確率——技術Bを利用している開発者の割合

調査の標本設計を考慮し、結果がより広範な開発者集団を代表するものとなるよう、すべての確率計算に調査ウェイトを適用している:

実装

リフト指標は、以下のように解釈される:

- リフト値 = 1.0:
独立。技術Aと技術Bは、両者の採用判断が独立であった場合に期待される確率とまったく同じ確率で併用されている。これは、両技術の間に関連性がないことを示す。
- リフト値 > 1.0:
正の関連性。両技術が、期待される確率よりも高い頻度で併用されている。その大きさが関連性の強さを示す
- $1.0 < \text{リフト値} < 1.3$:
弱い正の関連性
- $1.3 \leq \text{リフト値} < 1.7$:
中程度の正の関連性
- リフト値 ≥ 1.7 :
強い正の関連性
- リフト値 < 1.0:
負の関連性。
両技術が、偶然に期待される水準よりも低い頻度で併用されている。



調査方法

The Developer Nation Survey

世界的な広がり

第31回グローバルDeveloper Nation Surveyの回答者の地理的分布
(2025年12月～2026年1月)

31th

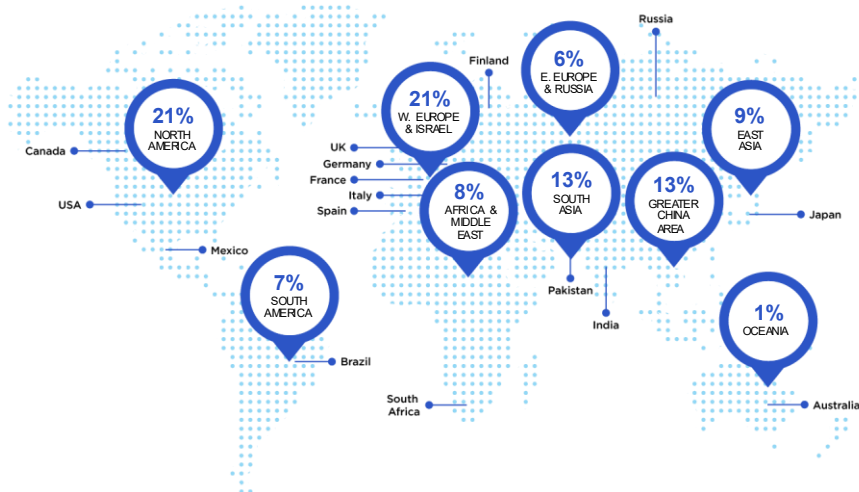
GLOBAL WAVE

12,800+

DEVELOPERS

100+

COUNTRIES REACHED



Developer Nationの第31版は、世界中の12,800名を超える回答者にリーチした。

これにより、Developer Nationシリーズは、モバイル、デスクトップ、産業用IoT、コンシューマーエレクトロニクス、サードパーティアプリエコシステム、バックエンド、Web、ゲーム、AR/VR、そして人工知能・機械学習に携わる開発者およびデータサイエンティストを総合的に対象とした、これまでに実施された最も包括的な独立系調査であり続けている。

本レポートは、SlashDataが設計・制作・実施した、2025年12月から2026年1月にかけての7週間にわたる大規模オンライン開発者調査に基づいている。

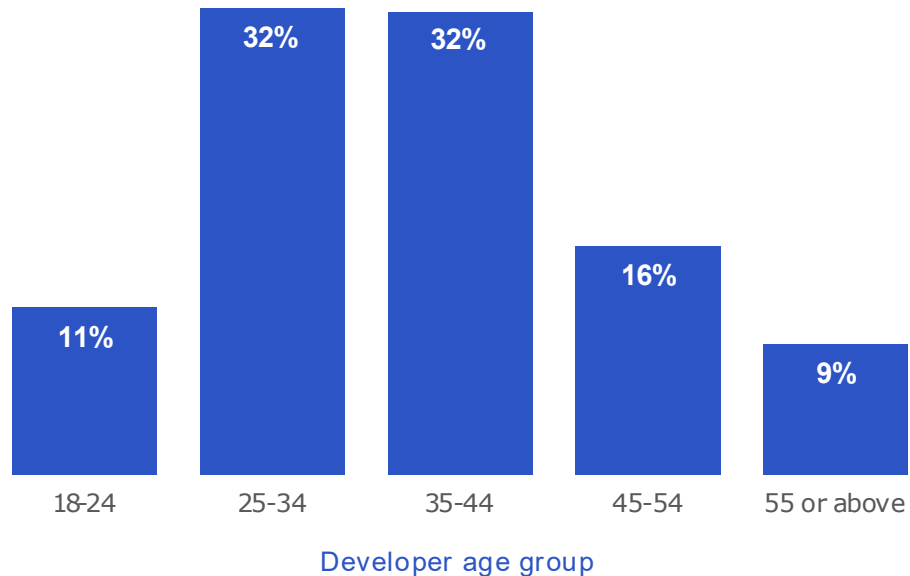
オンライン調査の回答者は、米国、中国、インド、日本、ブラジル、英国をはじめとする100か国以上から集まった。

この調査の地理的な広がり、開発者経済のグローバルな規模を反映している。オンライン調査は、英語に加えて9言語(簡体字中国語、繁体字中国語、フランス語、スペイン語、ポルトガル語、ベトナム語、ロシア語、日本語、韓国語)に翻訳された。また、ソフトウェア開発業界における14の主要パートナーおよびDeveloper Nationパネルを通じて告知された。

回答者は、18歳未満の若いコーダー・クリエイターから、55歳を超えるベテランまで、幅広い年齢層にわたっている。年齢について「回答したくない」とした人を除いた、回答者の年齢プロフィールは以下のとおりである。

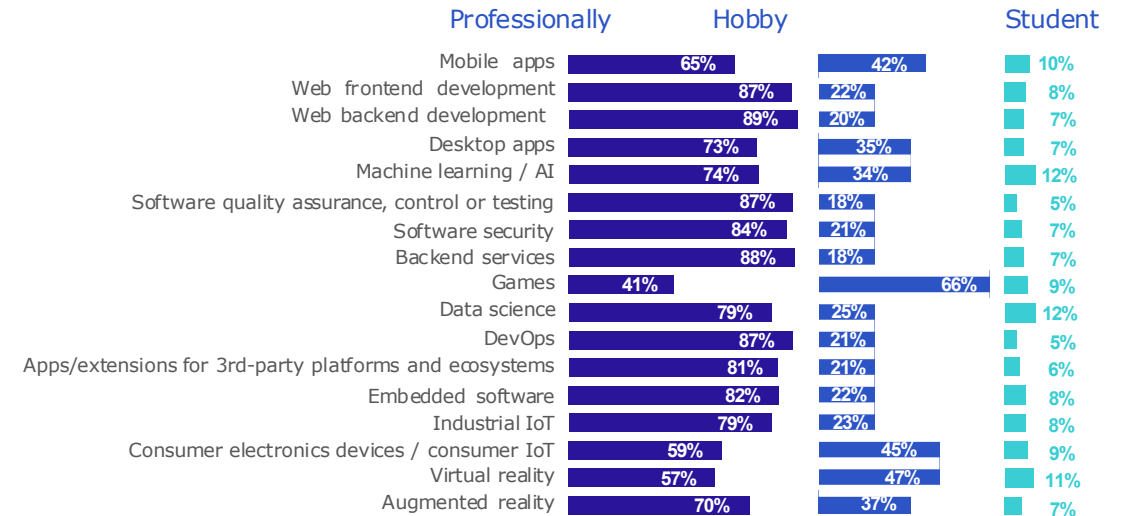
調査回答者の年齢分布

調査回答者の割合(%) (Q1 2026 n=12,818)



各ソフトウェア領域への回答者の関与状況

各領域における回答者の割合(%) (Q3 2025 n=12,021)



回答者は、調査対象となった13の分野——Webアプリ/SaaS、モバイルアプリ、デスクトップアプリ、バックエンドサービス、拡張現実(AR)、仮想現実(VR)、ゲーム、データサイエンス、機械学習・人工知能、産業用IoT、コンシューマーエレクトロニクスデバイス、組み込みソフトウェア、そしてサードパーティアプリエコシステム向けのアプリ・拡張機能——のうち、どの種類のプロジェクトに携わっているかを尋ねられた。さらに、それぞれの関与領域について、プロフェッショナル、趣味、学生、あるいはこれらの組み合わせのいずれの立場で関わっているかも回答した。加えて、各領域における経験年数についても申告した。

地域別のサンプリングバイアスの影響を排除するため、9つの地域にわたる地域分布に対して、Developer Nationの調査で特定された地域分布および成長トレンドに基づく係数を用いて、ウェイト付けを行った。

アウトリーチチャンネル全体にまたがるその他の重要なサンプリングバイアスを最小化するため、回答に対しても重み付けを行い、利用されている技術および開発者セグメントについて代表性のある分布を導き出した。

アンサンブルモデリングの手法を用いて、独立した代表性のあるチャンネルから得られたデータをもとに重み付き分布を導出した。その際、当社の調査パートナーのチャンネルは除外し、これらのチャンネル経由で募集された回答者に起因するサンプリングバイアスを排除している。

各個別ブランチ(産業用IoT、コンシューマーエレクトロニクス、サードパーティアプリエコシステム、バックエンド、組込み、拡張現実および仮想現実)は、それぞれ独立にウェイト付けを行ったうえで、最終的に統合された。

方法論の詳細については、以下をご参照ください

<https://www.slashdata.co/methodology>.



調査方法



Liam Bollmann - Dodd

Principal Market Research Consultant

Liamは、かつて反物質の実験物理学者であり、CERN(欧州原子核研究機構)に在籍しながら物理学の博士号を取得した。現在は、変化を続けるクラウド開発のランドスケープ、サイバーセキュリティ、そして技術の進展とそれが社会に与える影響との関係に関心を寄せている。

✉ liam.dodd@slashdata.co



Álvaro Ruiz Cubero

Research Manager

Álvaroは、戦略コンサルティングおよびオペレーションコンサルティングの経験を持つ市場調査アナリストである。経営学修士(MBA)を取得しており、データに基づく意思決定の力を信奉している。Álvaroは、企業が複雑な戦略的経営課題に取り組み、徹底した調査・分析に裏付けられた戦略的判断を下せるよう支援することに情熱を注いでいる。

✉ alvaro.ruiz@slashdata.co

著者について

Navigate **AI** **technology decisions** **with confidence & clarity**

SlashData is an AI analyst firm which has been working with the top Tech brands to provide clarity and confidence in their decision-making.

For 20 years, we have been tracking software technology trends and helping technology brands make product and marketing investment decisions, challenging assumptions and reframing market trends to empower industry leaders to drive the world towards the future.

Find us at slashdata.co





/DATA



CLOUD NATIVE
COMPUTING FOUNDATION