IoT Edge Working Group

エッジネイティブアプリケーションの原則についてのホワイトペーパー

著者

AMAR KAPADIA, AARNA NETWORKS; BRANDON WICK, AARNA NETWORKS; JOEL ROBERTS, CISCO; KATE GOLDENRING, FERMYON; DEJAN BOSANAC, RED HAT; TOMOYA FUJITA, SONY US LAB; RAVI CHUNDURU, VERIZON; NATALIE FISHER, VMWARE; STEVEN WONG, VMWARE.

レビュアー

FRÉDÉRIC DESBIENS, ECLIPSE FOUNDATION; PRAKASH RAMCHANDRAN, EOTF; MARK ABRAMS, SUSE.

CLOUD NATIVE COMPUTING FOUNDATION

出版： 2023年1月17日（第1版）, 2022年8月24日（草案）

目的

”エッジネイティブ”という用語は、Gartner、Macrometa、 FutureCIO などの業界ブログを含む多くの場所で言及されています。State of the Edge や Linux Foundation (LF) などの組織もエッジネイティブ アプリケーションについて議論していますが、エッジネイティブの原則には焦点が当てられていません。

このホワイトペーパーでは、エッジネイティブ アプリケーションとその原則の定義方法に焦点を当てています。

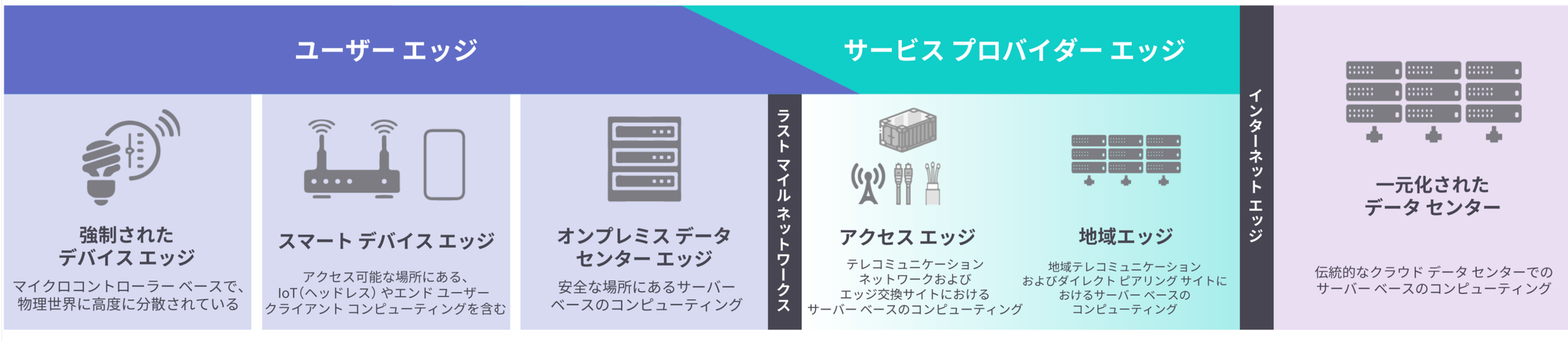
エッジとは何か？

エッジ コンピューティングは、データ処理をソースに近づけるパラダイムです。たとえば、工場でのロボット制御などです。

今後5年間で、業界は2022年から2030年にかけて38.9%成長すると予測されているため、エッジ コンピューティングはより至る所で目にするようになります。企業は、エッジにコンピューティング パワーを持つことのメリットを次のように考えています。

- レイテンシーの削減
- 帯域幅の管理
- 機密データのプライバシーの向上
- 信頼性の低いネットワークでの中断のない運用

エッジ コンピューティングにはさまざまな定義がありますが、このホワイトペーパーでは、データ処理リソースの地理的な位置に基づいて定義します。地理ベースのエッジは、ユーザーからの距離に応じて複数のカテゴリに分類されます。次の図は、Linux Foundation Edge Whitepaperで定義されているカテゴリを示しています。



エッジネイティブの原則は、クラウドネイティブの原則と多くの類似点を共有しています。
しかしながら、いくつかの重要な違いもあります。

クラウドネイティブ
対 エッジネイティブ

Cloud Native Computing Foundation (CNCF) によって定義されたクラウド ネイティブ テクノロジーは、次のとおりです。

> ”クラウドネイティブ テクノロジーは、パブリック クラウド、プライベート クラウド、ハイブリッド クラウドなどの最新の動的環境でスケーラブルなアプリケーションを構築し、実行できるようにします。コンテナ、サービス メッシュ、マイクロサービス、不変インフラストラクチャ、宣言型APIは、このアプローチの例です。
>
> これらのテクノロジーは、回復力があり、管理しやすく、観測可能な疎結合システムを可能にします。堅牢な自動化と組み合わせることで、エンジニアは最小限の労力で、影響の大きい変更を頻繁かつ予測可能に行うことができます。”

The Open Glossary of Edge Computingは、”エッジネイティブ アプリケーション”はクラウドネイティブの原則を活用すると述べているので、この広範な使命はエッジ アプリケーションにも適用できます。

> ”エッジ コンピューティング機能を活用するためにネイティブに構築されたアプリケーションで、一元化されたデータ センターで運用するには非現実的または望ましくないものです。エッジネイティブ アプリケーションは、リソースの制約、セキュリティ、レイテンシー、自律性などの領域におけるエッジの独自の特性を考慮しながら、クラウドネイティブの原則を活用します。エッジネイティブ アプリケーションは、クラウドを活用し、アップストリーム リソースと連携して動作する方法で開発されています。一元化されたクラウド コンピューティング リソース、リモート管理、およびオーケストレーションを理解していない、またはCI/CDを活用していないエッジ アプリケーションは、真に”エッジネイティブ”ではなく、むしろ従来のオンプレミス アプリケーションによく似ています。”

クラウドネイティブのユースケースが従来のクラウドを離れてエッジ ロケーションにデータやイベントを組み込むにつれて、エッジの独自の特性を管理しながら、回復力があり、管理しやすく、観測可能な疎結合システムを提供するという目標に沿って、新しいツールや手法が進化しています。

類似点

多くのクラウドネイティブの原則がエッジネイティブに適用されます。
このセクションでは、これらの類似点について説明します。

項目	クラウドネイティブ＆エッジネイティブ
アプリケーションとサービスの移植性	アプリケーションとサービスは、インフラストラクチャからの結合を抽象化します。適切に作成されたアプリケーションは、どこで実行されているかを意識させず、プラットフォーム間での移植が期待できます。
観測可能性	このプラットフォームには、問題の検出とメトリクスの収集を可能にする、十分に文書化されたインターフェースとツール オプションの一式が付属しています。これにより、開発者は回復力があり、効率的に管理されたシステムを構築できます。
管理機能	アプリケーションとリソースを大規模に管理するためのインターフェイスとツール オプションが提供されます。プラットフォームには、ベースラインのネットワーク接続、サービス、管理機能を提供するプラグイン メカニズムもあります。
多様な言語とフレームワークのサポート	アプリケーションとサービスは、さまざまな一般的な言語とフレームワークを使用して実装およびホストできます。

エッジネイティブ アプリケーションの原則についてのホワイトペーパー

相違点

エッジネイティブとクラウドネイティブの広範なミッションは類似点を共有していますが、開発者はその相違点を認識する必要があります。

項目	クラウドネイティブ	エッジネイティブ
アプリケーション モデル	ロード バランスによる水平スケーリングのためにステートレスで構築されたマイクロサービス コンポーネントがほとんどです。	サービス プロバイダーのエッジ アプリは非常に似ていますが、ユーザー エッジ アプリはシングルトンの”モノリシック”であることがあります；どちらの場合も、ステートはアプリと同じに管理されることがあります。
データ モデル	ステートレスなコンポーネントをサポートする集中型モデルが一般的です。	キャッシング、ストリーミング、リアルタイム、および分散モデルがよく利用されます。
弾性	高速なスピン アップおよびスピン ダウン；通常、基盤となるリソースは無制限として扱われます。	エッジのハードウェア リソースに制約があるため、弾性に制限があります。利用可能な場合は、接続されたクラウドまで”垂直”に拡張できます。
耐障害性	障害ドメインに分散した冗長ノードを使用して、耐障害性はクラウド プロバイダーにアウトソーシングされます。	多くの場合、ハード化されたインフラストラクチャに依存しており、ステートフル コンポーネントのリカバリー アーキテクチャを備えています。多くの場合、耐障害性はクラウドよりも低い可能性があります。
スケール	通常は、少数の場所、インスタンスに限定されます。	多数の外部デバイス(最大100,000)をサポートし、多数のロケーション(最大10,000)にまたがる場合があります。
オーケストレーション	大規模なパブリック クラウドやオンプレミス クラウドでのオーケストレーションは、一元的にプールされたホスト上でワークロードを実行し、水平的にスケジュールすることで、効率性と可用性を実現するように設計されています。	エッジは分散化されており、ワークロードは分散的にデプロイされ、多くの場合、場所固有の方法でスケジュールされます。
管理	クラウドネイティブとエッジネイティブはどちらも管理可能ですが、そのメカニズムはさまざまで、クラウドネイティブは中央管理と自動化に依存しています。	エッジネイティブでは、リモートおよび一元管理と、ハードウェアおよびソフトウェアのゼロ タッチ プロビジョニングの組み合わせが必要です。エッジでのスタッフ配置は、トレーニングを受けていない、信頼されていない、最小限である、または存在しない場合があります。リカバリの課題があるため、”ブリック”耐性のある”アトミック”アップグレードが望ましいです。
ネットワーク	アプリケーションは、豊富な機能を備えた高速ネットワークに依存できます。	アプリケーションは、さまざまな速度（断続的で貧弱なものから優れたものまで）と機能を考慮する必要があります。非IPプロトコル ネットワークからのデータとイベントを統合する、モバイルおよび無線ベースのものが含まれます。
セキュリティ	セキュアな施設内の信頼された構造	セキュリティの低い環境では、信頼性がゼロになります。
ハードウェア対応	ほとんどのアプリケーションに対応するフレーバーを備えた計算リソースのデファクトの統一性のために、アプリケーションがハードウェアの配置に関心を持つ必要はほとんどありません。	アプリケーションには、ハードウェア プラットフォーム、ロケーション、および／またはセキュリティ対応を要件とする、より大きなリアルタイム要件がある場合があります。開発者は、より広範な種類のハードウェアおよびインターフェイスに対応する必要があります。
外部リソースとの相互作用	アプリケーションがローカルのハードウェア リソースと相互作用する必要はほとんどありません。	エッジにデプロイされたサービスは、多くの場合、カメラ、センサー、駆動装置、ユーザーなどのローカル環境と相互作用する必要があります。

エッジネイティブ アプリケーション

エッジネイティブ アプリケーションは、エッジ用に作成されたアプリケーションおよびサービスです。これらは、上記の類似点と相違点を考慮した方法で作成されています。これらのアプリケーションの中核となる原則を次に示します。

エッジネイティブ アプリケーションの原則についてのホワイトペーパー

エッジネイティブの原則

エッジネイティブ アプリケーションは、前述のエッジネイティブの使命を果たすために、次の原則に従う必要があります。

	原則	解説
リソース＆ デバイス対応	ハードウェア対応	開発者は、同種のハードウェア プラットフォームを持つのではなく、より広範な種類のハードウェアとインターフェイスに対応する必要があります。
	外部デバイスへの 接続性	アプリケーションは、環境内のデバイスへの接続方法に対応し、実行時の機能の変更に対応する必要があります。たとえば、初期設定または新しいデバイスがネットワークに入った後に、センサーをエッジ サーバーに接続解除/接続することに応答する必要があります。機能は静的ではなく、環境を含むため、オーケストレーターは機能の変更をアプリケーションの状態に調整できる必要があります。
	可変の接続性対応	アプリケーションは、非同期通信、キューイング、キャッシュなどのメカニズムを使用して、信頼性が低い、または使用できない(エア ギャップがある)ネットワーク接続に適応する必要があります。エッジが中央サイトから設定をプルする場合、スケール、ネットワーク接続、およびセキュリティの問題を克服するには、”プル”メカニズムが必要になる場合があります。
大規模管理	一元的な監視可能性	エッジネイティブ　アプリケーションとクラウド ネイティブアプリケーションの両方が一元的に監視可能である必要がある一方で、エッジネイティブ アプリケーションには独自の考慮事項があります。エッジネイティブ アプリケーション インスタンスは、大規模なインスタンスにデプロイされたり、制約のあるスタッフやフィールド サポートのみであったりする可能性があります。これらの理由から、分散データ収集や集中集約、オープン ループ(人間が観測可能/実行可能)とクローズド ループ自動化(マシン実行可能)の組み合わせなどの技術が必要です。監視可能性は、メトリック、ログ、デジタル ツイン、アラート(イベントとアラーム)、ヘルス モニタリングをカバーします。
	インフラストラクチャと プラットフォームの 大規模な管理	エッジ アプリケーションでは、インフラストラクチャとプラットフォームの大規模な管理が重要であるため、宣言的な目的ベースのメカニズムが必要になります。さらに、デバイスのオンボーディング、水平スケールの制限、ベア メタル環境の管理など、独自の要件が存在する可能性があります。プラットフォーム レベルでは、Kubernetesや仮想化レイヤー、さまざまなプラグインのデプロイや管理も懸念されます;アプリケーションの移植性を可能にするために、プラットフォーム レイヤーを可能な限りベンダー中立に保つ一方で。
	アプリケーションの 大規模な管理	アプリケーションの数とこれらのアプリケーションのインスタンスの数は、エッジ上で非常に多くなる可能性があり、宣言的な配置と構成の意図、クローズド ループの自動化による自動化されたサービス保証、複数のアプリケーション インスタンスにわたる集約された管理ビューなど、さまざまな自動化が必要になります。アプリケーションにはリアルタイムのニーズもあるため、アプリケーションとインフラストラクチャ/プラットフォーム(GPU、DPU、FPGA、CPUアーキテクチャ、カーネル最適化、Kubernetesプラグインの使用など)との連携は、クラウド アプリケーションよりも緊密になる可能性があります。つまり、アプリケーション オーケストレーションは、インフラストラクチャ/プラットフォーム オーケストレーションをトリガーする可能性があります。
スパニング		アプリケーションは1つの場所に存在するのではなく、ティアーは地理的、遅延、および障害ドメインの境界をまたぐことができます。実際、エッジ アプリケーションは、パブリック クラウドまたは集中型プライベート クラウドにまたがる場合もあります。
リソース利用率の 最適化		エッジ コンピューティングのリソースには制約があるため、アプリケーションはリソースの使用状況を継続的に最適化する必要があります。これは、オンデマンドでアプリケーションを起動および停止することを意味する場合があります。これは、配置や可用性の意図に基づくマイグレーションやレプリケーションを意味する場合があります。これは、1日の異なる時間に異なるワークロードを実行することを意味する場合があります。
アプリケーションの移植性と 再利用性(制限あり)		抽象レイヤーは、ベンダー ニュートラルなPaaSを通じて、インフラストラクチャとプラットフォームに依存しない移植性を提供しようとします。ただし、ローカル リソース、ハードウェア プラットフォーム、セキュリティ、モバイル ネットワーク、およびその他の対応のために、ローカルの違いに適応するための設定オプションが必要です。

これらの9つの原則は、5つの原則のより小さなセットにグループ化できます。ハードウェア対応、外部デバイス接続、および可変ネットワーク可用性の対応はすべて、リソースおよびデバイス対応のより広い原則の下で考慮できます。同様に、エッジ アプリケーションに必要とされる、大規模な管理可能性、一元的な監視可能性、管理可能なインフラストラクチャとプラットフォームはすべて、大規模な管理の原則の下で分類できます。拡大された5つの原則は、スパニング、リソース使用率の最適化、制限付きの移植性と再利用性、リソースおよびデバイス対応、および大規模管理です。

エッジネイティブ アプリケーションの原則についてのホワイトペーパー

結論＆
次のステップ

この論文は初版であり、改訂される可能性があります。
この論文のサブセクションに関連する今後の報告が予想されます。

参加方法

The CNCF IoT Edge Working Groupには、定期的なミーティング、メーリングリスト、Slackがあります。最新の情報については、ワーキンググループのGitHubページのCommunicationセクションを参照してください。エッジ関連のプロジェクトを発表したり、グループの活動領域のアイデアを提供したり、このホワイトペーパーの改訂や続く報告の草稿を支援したりすることなどの、読者の参加を歓迎します。

プロジェクトやイニシアチブ

このホワイトペーパーの一環として、the CNCF IoT Edge Working Groupは、アプリケーション開発者がこのホワイトペーパーで概説されているエッジネイティブ アプリケーションの原則を達成するのに役立つオープンソース プロジェクトのワーキング リストを収集しています。

リストは、このスプレッドシートまたはQRコードで確認できます。プロジェクトを追加するための編集アクセス権を取得するには、IoT Edge Working Group Google groupに参加してください。

この文書は、以下のホワイトペーパーの参考訳です。
Edge Native Application Principles Whitepaper

翻訳協力：橋本修太

参考資料

Linux Foundation Edge Whitepaper:
https://www.lfedge.org/wp-content/uploads/2020/07/LFedge_Whitepaper.pdf

Open Glossary of Edge Computing [v2.1.0] State of the Edge:
https://github.com/State-of-the-Edge/glossary/blob/master/edge-glossary.md#edge-native-application

Cloud Native Computing Foundation (CNCF) Charter:
https://github.com/cncf/foundation/blob/main/charter.md

Gartner ‘Cloud Native Isn’t Edge Native’:
https://blogs.gartner.com/thomas_bittman/2020/04/17/cloud-native-isnt-edge-native/

Macrometa ‘Edge Native is not Cloud Native’:
https://www.macrometa.com/blog/edge-native-is-not-cloud-native

Future CIO ‘Cloud-Native versus Edge-Native: know the difference’:
https://futurecio.tech/cloud-native-versus-edge-native-know-the-difference/

Edge Computing Market Size, Share & Trends Analysis Report By Component (Hardware, Software, Services, Edge-managed Platforms), By Application, By Industry Vertical, By Region, And Segment Forecasts, 2022 - 2030
https://www.grandviewresearch.com/industry-analysis/edge-computing-market