

# SaaS 実現のための5つのキーステップ： オンプレミスソフトウェアをクラウドに移行する ためのテクニカルガイド

ホワイトペーパー

## 目次

はじめに.....	2
ソフトウェアをクラウドに移行させる際の課題.....	2
SaaS 実現のための5つのキーステップ.....	3
ビジネスとプロジェクトのプランニング.....	4
開発とサービスプロセスの再構築.....	4
要件の定義とインフラの移行.....	6
既存製品への対応.....	7
エンドユーザの事例： SaaSへの発展経路の技術的な例.....	7
SaaS投資を収益化するためのソリューション.....	9
まとめ.....	10
CorSource Technology Groupソリューションについて.....	10
SafeNetのソフトウェア収益化ソリューションについて.....	10

## ソフトウェア開発会社をクラウドへと駆り立てているものは何か？

- 新たな収益を求めて：75%
- 運用プロセスの簡素化：65%
- 新規・ニッチ市場の開拓：48%
- エンドユーザの操作性の向上：46%
- トラッキング機能やレポート機能の利用：33%
- 市場化までの時間の短縮：32%

~SafeNet and the Software & Information Industry Associationによる「ソフトウェアの収益化状況」に関する調査、2013年

自社のSaaSエンドユーザについて物理監査を行ったあるソフトウェア会社は、85%がサービス契約に違反していることを発見しました

## はじめに

Software as a Service (SaaS)は、ソフトウェアの提供方法を大幅に変化させました。従来のオンプレミスソフトウェアのメーカーも、製品をクラウドへと移行させようとするのが増える中、多くの企業がクラウド化を進める際の課題に直面しています。

本ペーパーでは、これらの課題について検討し、オンプレミスソフトウェアをクラウドに移行させる上で重要なプランニング、アーキテクチャデザイン、実装に関する検討事項について説明します。数多くの移行の成功例から得たベストプラクティスとケーススタディの例を利用して、ソフトウェア開発会社によるオンプレミスソフトウェアのクラウドへの移行と、SaaS製品への投資の完全な収益化を成功させるための段階的なプロセスを学ぶことができます。

## ソフトウェアをクラウドに移行させる際の課題

オンプレミスソフトウェアをオンデマンドモデルに移行させるには、技術の変化以上に多くのものが必要であり、また、最初に予想していたよりもずっと多くの時間がかかります。これには、事前の検討やプランニング、ビジネス戦略や製品に基づく多くの重要な判断が必要です。

現行のオンプレミス製品と、オンデマンド製品計画と実行化の間で、最適なバランスを見つけることは、非常に難しいと考えられています。まずは一般的な概要から検討を始め、そのあと詳細を埋めるような段階的なアプローチを用いることが重要です。最初にクラウドを実現するための方法について検討を始めると、進化的な開発アプローチを取るべきか、あるいは革新的な開発アプローチを取るべきか、判断しなければいけません。そして次に、技術的なアーキテクチャや要件、インフラ等について決定する必要があります。また、売るのが製品からサービスへ変わるということは、製品開発だけでなく、サービスカタログへの掲載方法や、プロビジョニングへの対応やユーザアクセスの承認方法など、ビジネスやサービスデリバリーにも検討が必要です。

サービス契約の順守は、クラウドに移行しようとするソフトウェア開発会社にとって、最も重要な課題の一つです。企業アカウント体制全体でエンタイトルメントを管理し、施行することは複雑です。ソフトウェアサービスを提供したら、エンドユーザに必ず、締結した契約の条件に従ってサービスを利用させるようにする必要があります。大半のソフトウェア開発会社では、エンドユーザが、契約した機能だけの利用や、あるいは、アクセス可能なユーザの数の制限を確保するための方法が整備されていません。例えば、あるソフトウェア開発会社は、アクセスを制限する方法を整備しないままソフトウェアサービスを開始しましたが、エンドユーザ企業にて実際に監査を行うと、85%がサービス契約を順守していない、つまり、ソフトウェアを無償で使わせていたことが分かりました。適切なコントロールが整備されていないと、潜在的な収益機会は失われてしまいます。

もう一つの課題は、進化を続ける新たなマーケットにサービスを迅速に適応させるために、オンプレミスソフトウェアと SaaS 提供の両方のためにバックオフィス機能を管理することが必要です。これはまだ始まりに過ぎません。ソフトウェアをクラウドに移行する際には、技術やビジネス面で検討すべき課題がその他にも数多くあります。

幸いなことに、参考になるものがあります。この後、何千回とは言わないまでも、何百回もの移行を成功させた他のソフトウェア開発会社やサービスプロバイダの体験とベストプラクティスについて説明していますので、ソフトウェアをクラウドに移行させ、継続的な開発と広範な製品ポートフォリオのサポートにうまく対応する上で参考にしてください。

## SaaS 実現のための5つのキーステップ



### 1. ビジネスとプロジェクトのプランニング

ビジネスとプロジェクトのプランニングは、SaaS実現のための最初のステップである、最も重要なステップの一つです。にもかかわらず、見過ごされることが最も多いステップでもあります。この製品のターゲットは誰か、どのように利益を上げるのか、どのライセンスモデルを選ぶのかなど、ビジネス目標やプロジェクトへのアプローチ方法について、事前に大いに検討することが必要となります。

まず、現在のオンプレミス ソフトウェア製品を調べ、SaaSバージョンに利用できるかどうか判断します。プロジェクトを進めるには、どのようなスキルが必要か？現在提供しているオンプレミス製品を進化させ、漸進的あるいは段階的なプロジェクト計画を採用するのか？それとも、新しい SaaS 製品開発プロジェクトをゼロから始めるか？このプランニング段階では、プロジェクト計画を策定する前に、ビジネス目標と企業の望むビジネスモデルを慎重に検討することが必要になります。

SaaSへの正しい道筋を選ぶことは、プロジェクトプランニング段階の重要な要素です。現在のオンプレミス製品の状態から SaaS に進化させるべきなのでしょうか、それとも、ゼロから新しいSaaS製品を開発すべきなのでしょうか。これを決定するには、下の表の質問に答えて参考にしてみてください。

SaaS への正しいステップを選ぶ		
御社のSaaS製品のターゲット市場は、現在のオンプレミス製品と同じですか？	はい	いいえ 新しいターゲット市場 (小企業など)
SaaS に移行するには、オンプレミス製品のビジネスモデルやワークフローを大幅に変更する必要がありますか？	いいえ	はい 主な機能の追加、機能の変更、ビジネスモデルやワークフローの変更
現在のオンプレミスソフトウェアは、マルチ階層アーキテクチャに適していますか？	はい	いいえ
<b>選択：</b>	<b>進化：</b> 現在のオンプレミスソフトウェアをSaaSに進化させる	<b>革新：</b> 新しいSaaS製品をゼロから開発する

多くのソフトウェア開発会社は、SaaSへ革新的な方法ではなく、段階的に進める進化的な方法を選択しています。現在のオンプレミスソフトウェア製品が、SaaS製品と同じシステムの顧客をターゲットにし、製品のビジネスモデルまたはワークフローに大幅な変更の必要がなく、アーキテクチャが適切であれば、段階的な方法を進める「進化」は正しい選択です。単純にインターフェースを新しくし、ターゲットとなる市場にソフトウェアのオンデマンドバージョンを提供するのであれば、進化的な経路が適切でしょう。

一方、新しい市場をターゲットにすることを決めるソフトウェア開発会社もあります。こうしたソフトウェア開発会社の大半は、現在の製品を買うことができない見込み客か、インターフェースやビジネスモデル、ワークフローの完全な書き換えや最新化を必要とする見込み客にソフトウェアのSaaSバージョンを提供することによって、市場を拡大したいと考えていることが、他の事例からも示されています。こうしたソフトウェア開発会社の多くは現在のアーキテクチャ、特に、一部の旧製品のアーキテクチャは、マルチ階層への移行には適しておらず、SaaSに進化させることはとても難しいということを認識しています。この場合、ソフトウェア開発会社はSaaSのために「革新」的な方法を選択し、製品開発をゼロから始めることを選択します。

オンプレミス ソフトウェア製品のポートフォリオを持つ多くの ソフトウェア開発会社 の場合には、オンプレミス ソフトウェアの SaaS モデルへの移行に対して進化的なアプローチを取る方が適切でしょう。このアプローチは、ステップを比較的小規模で扱いやすいものにし、ソフトウェア開発会社が、ビジネスを行うと同時に、現在の製品と顧客基盤に対応し続けることを可能にすることから、対応しやすいといえます。

## 2. アーキテクチャ プランニング

ビジネスとプロジェクトに関する計画が完了したら、アーキテクチャ プランニングに進みます。この時点で、アーキテクチャをどうマルチテナンシーに対応させるかを決定判断する必要があります。スケーラビリティや、管理容易性に関する数多くの検討事項についてはどうでしょうか？技術的にはどのアプローチを取りますか？既存のオンプレミス製品の中に、利用できるものはありますか？アーキテクチャ プランニング段階では、SaaS の成熟度の4つのレベルを理解することが重要です。

### 成熟度レベル1：

レベル1では、御社のソフトウェアはインターネットでアクセス可能で、各エンドユーザは、御社のアプリケーションと各自のデータの一意のインスタンスにアクセスできます。このレベルでは、設定で自由に変われることは最低限で、エンドユーザごとにカスタムコーディングが必要となることがあります。

### 成熟度レベル2:

レベル2でも、各エンドユーザのアプリケーション インスタンスはまだ1つですが、一意のインスタンスではなくなります。各インスタンスには同じコードベースが含まれ、カスタムコードに頼らずに設定することが可能となります。

### 成熟度レベル3:

レベル3で、アプリケーションがマルチテナントで効率的になります。このレベルでは、御社の製品で多くのインスタンスが実行されてエンドユーザ基盤に対応するのではなく、御社のソフトウェアの1つのインスタンスのみで複数のテナントまたはエンドユーザに対応します。ですが、レベル3では、テナントデータは複数のデータベースには保管されません。このレベルでは、データベースはまだ1つしかありません。

### 成熟度レベル4:

レベル4では、ソフトウェアはマルチテナントかつ完全にスケーラブルになります。レベル4は、アプリケーションのロードバランスをサポートしながら、複数のテナントに対応し、可変数のサーバで実行されます。レベル4では、システムアーキテクチャを一切変更することなく、需要に合わせて処理能力をダイナミックに調整することができます。レベル4になると、アプリケーションは完全に、真のマルチ階層アーキテクチャによるサービススペースの提供内容に進化したといえます。

### SaaS の成熟度の4 レベル

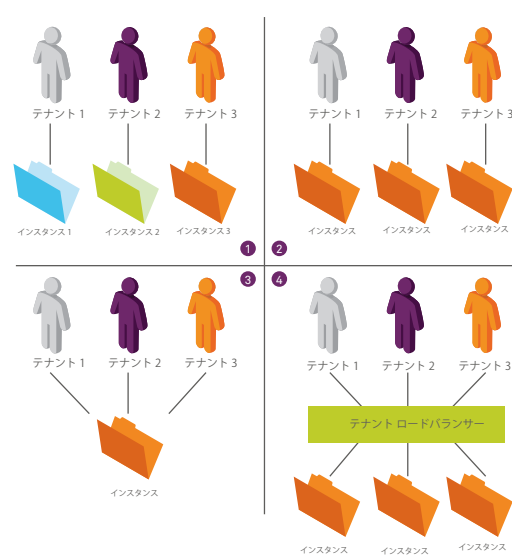


Diagram source: Microsoft

## 3. 開発とサービスプロセスの再構築

次に、開発とサービスプロセスの再構築を検討する必要があります。あらゆるソフトウェア開発同様、その先の問題を避けたいなら、柔軟性が重要です。開発プロセスは、要件、更新管理、テストという点においてフレキシブルである必要があります。また、カスタマーサービスモデルや、マルチテナント環境でのサポートの提供方法についても決定する必要があります。

Software as a Serviceに移行する際には、多くのソフトウェアレイヤを検討する必要があります。マルチ階層アーキテクチャの一番上のレイヤは、ユーザ インターフェースレイヤです。このレイヤは、ユーザがWebブラウザ、あるいはシンクライアントによって、システムと情報をやり取りする方法を表します。例えば、各エンドユーザ向けのソフトウェアサービスをエンドユーザのコーポレートカラーとロゴを使ってブランド化する場合、これはユーザ インターフェースレイヤで行われます。2番目のレイヤは、ビジネスロジックまたはサービスレイヤです。これは、エンドユーザにカスタム設定可能なワークフロー、ユーザ設定のセキュリティ アクセス、プロビジョニング、ライセンスング、レポートング、ビルディングなど、ソフトウェア投資を収益化することを可能にするビジネスサービスに相当するロジックです。3番目のレイヤは、データアクセスレイヤです。

これは、各エンドユーザのためのエンドユーザデータフィールドを実装できるレイヤです。これらはすべて単一のデータベースに属しますが、セキュアアクセスと適切なパーティショニングによって、データがエンドユーザ間に流出することを防ぎます。もう少し詳しくこのアーキテクチャの主な特徴について調べてみましょう。

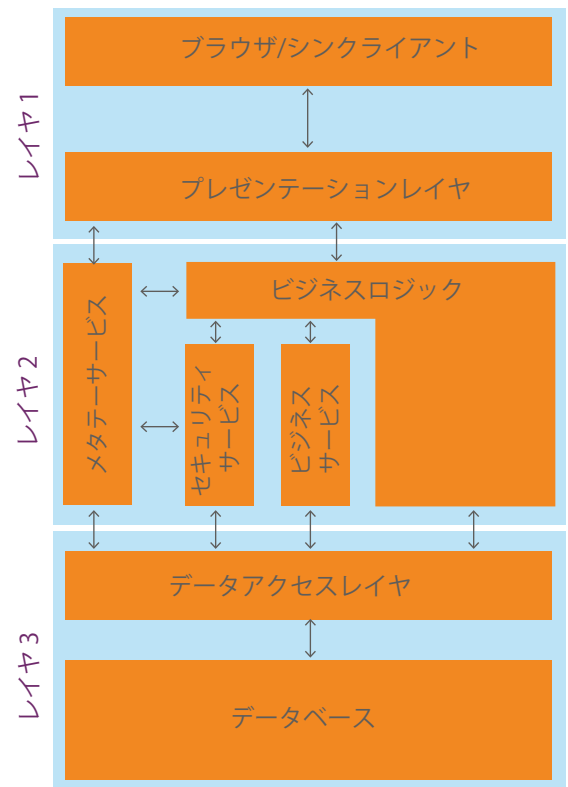
### SaaSのソフトウェアアーキテクチャの特徴

各エンドユーザのニーズに向けて最適化するため、プレゼンテーション、ビジネスロジック、セキュリティ、データベースなど、SaaSソフトウェアのすべてのレイヤをテナント設定可能な必要があります。通常はメタデータによってサポートされているので、ブランド化、インターナショナルライゼーションとローカリゼーション、ワークフロー、アプリケーションの様々な部分に対するユーザアクセスのコントロール、特定のデータ表やフィールドについて一意のテナント設定が可能で、また、クライアント別の拡張も可能です。

マルチテナント効率、テナント間でのアプリケーションアーキテクチャの主なレイヤの共有によって得られますが、テナント間の分離はそのまま維持されます。スレッドされプールされるビジネスロジックレイヤ、共有されるが個別のセキュリティレイヤ、共通ではあるが分離されたデータベースを使います。これには、現在の設計の適切性によって、予め設計が必要になるか、もしくは、既存のアプリケーションの完全な再設計が伴うこともあります。マルチテナント効率は、各クライアントのカスタムビジネスロジックを実現すると同時に、共通のコンポーネントを使って複数の目標を達成させます。

拡張性は、同じアプリケーションインフラを使って広範なクライアントにリーチするためのSaaSにおける要となる要素です。拡張性を考慮するには、ステートレスでプールされたビジネスロジックを使った設計が必要で、これには、キャッシュされ、プールされた非同期のデータアクセスが必要です。また、データベースへのアクセスをパーティションすることも必要になります。

マルチ階層のSaaSアーキテクチャ



SaaSでは、ソフトウェアの構造だけでなく、ビジネスの構造も異なってきます。あらゆる SaaS 製品の重要な特徴は、展開と継続的管理で運用可能なビジネスサービスです。これらのビジネスサービスには、ライセンスング、プロビジョニング、モニタリング等を実現するオペレーションサポート システム (Operational Support Systems:OSS) や、請求書発行やエンドユーザへの対応を実現するビルディングシステムサポート (Billing Systems Support :BSS) が含まれます。これらについては、要件の定義に関する次のセクションと、その後 SaaS 製品への投資の収益化について説明する際に、再び詳しく取り上げます。

#### 4. 要件の定義とインフラの移行

開発とサービスプロセスの再構築について検討した後は、要件を定義します。要件を明確にし、ビジネスやプロジェクトの目標と関連付けることが重要です。要件を決定する際には、現在のオンプレミス製品の機能を全部使うのか、それとも一部なのか、すべて終えてから、ライセンスングとプロビジョニングを利用して、異なったユーザセグメントをターゲットにする異なったバージョンを構築するのか、どのように管理するのか、オンプレミスソフトウェアにはない付加的な機能を追加するのか、ユーザインターフェースを最新化したいのか、といった質問に確実に答えられるようにしましょう。今日のオーディエンスに適切な操作性を提供したいのであれば、ユーザインターフェースを更新することは、重要な要件の一つです。

また、インフラの移行についても検討する必要があります。SaaS製品のためのインフラは、オンプレミスで使っているものとはまったく異なります。そのためオンデマンド製品向けに、より柔軟性の高い保守のしやすいインフラに移行する必要があります。

要件を定義する際のもう一つの検討事項は、製品の多用途性とビジネスアジリティです。ソフトウェア開発会社は、最初のSaaS製品を発売する際、自社の製品のなかでも非常に基本的なバージョンの販売から開始することがよくあります。その後、競争上の理由から、あるいは、エンドユーザからの依頼を受けて、バージョンや機能が異なる製品を提供する必要があるということをしすぐに認識します。それは小規模なビジネスユーザや中規模市場、あるいは、エンタプライズ向けのバージョンであることもあります。どんな理由であれ、ビジネス上の観点から、ソフトウェア開発会社は市場からのリクエストに対し機敏に対応する必要がありますが、多くのソフトウェア開発会社は、これらの調整を迅速に行うためのツールを整備していません。そして、この種の変更にはエンジニアリングが関与してプログラミングする必要があるため、ますます対応が遅くなります。

同様に、ソフトウェア開発会社は、発売時に採用したサブスクリプションベース（永久使用）のライセンスングは、十分ではないということも認識しつつあります。IDCのAmy.Konaryは、「ソフトウェア開発会社の多くは、ライセンスングポリシーがクラウドの登場よりも前の時代に作られたものであるため、問題に直面しています。ユーザごと、あるいは月ごとなど、オンプレミスソフトウェアにおいては有効だった基本的なライセンスングのコントロールやメトリックスは、SaaSクラウドサービスの規模には対応できません」と話します。

SaaS投資を収益化するための方法は他にも沢山あります。例えば、利用ベースのビジネスモデルを採用したいと考えるかもしれません。利用状況を追跡し、それに従って使用許諾するためのツールは整備されていますか？高度なライセンスモデルも数多く存在します。問題は、フリーミアム、サブスクリプションベース、機能ベース、利用ベース、従量制あるいはこれらの組み合わせなど、エンドユーザが希望する様々な購入方法でサービスを提供できるかどうかという点にあります。

エンドユーザの利用データを収集したら、そのデータを利用して、御社のビジネスと、エンドユーザの利用方法について、理解をより深めることが課題になります。これは、リソースをどこに投資するかあるいはどこから投資を引き上げるべきかを把握する上で役立つだけでなく、営業やマーケティングの取組みのための情報をも提供するビジネスインテリジェンスです。この分野において効果的であるためには、情報を適切に収集、集計、分析し、情報に従って行動するための適切なシステムを整備する必要があります。

「ソフトウェア開発会社の多くは、ライセンスングポリシーがクラウドの登場よりも前の時代に作られたものであるため、問題に直面しています。ユーザごと、あるいは月ごとなど、オンプレミスソフトウェアにおいては有効だった基本的なライセンスングのコントロールやメトリックスが、クラウドサービスには対応できません」

— IDC, Amy Konary

同様に、大半のソフトウェア開発会社はオンプレミス ビジネスを置き去りにしようとしているわけではなく、新しく提供する SaaS と平行して、オンプレミス ソフトウェアの開発、販売、サポートを継続して行うことも計画しています。ここから、バックオフィスのサポートという、次の問題が生まれます。業務面から、複数の製品とデリバリ プラットフォームにどのように対応すればいいのでしょうか？複数のバックオフィスシステムを利用しますか？両方のデリバリメカニズムを通じて、ライセンスとエンタイトルメントをどのように管理すればいいのでしょうか？

## 5. 既存製品への対応

最後に、この期間に、御社の様々な製品のそれぞれについて、継続的な開発をサポートするためにリソースのバランスをどのように取るかを決定します。経験から、これが、かつてはオンプレミスのみだったビジネスモデルに、SaaS への取り組みを追加する上で、ソフトウェア開発会社にとっては重要な分野であることが示されています。ソフトウェア開発会社は、既存のオンプレミス ソフトウェアはただ現状に留まるのではなく、SaaS への進化が、明らかに異なる2つの開発サイクルにより、何らかの形で前進し続ける必要があるということを認識するようになっていきます。既存のソフトウェア製品と新しいSaaS 製品の開発サイクルはそれぞれ異なる可能性があるため、既存製品と SaaS 開発との間で、限りある開発リソースのバランスをどう取るのか解決する必要があります。

### エンドユーザの事例：SaaSへの進化的な経路の技術的な事例

#### 企業

中小小売企業向けリテール POS システムとソフトウェア ソリューションの大手プロバイダ

#### 同社の課題

このリテール プロバイダは、これまで獲得したことの無い見込み客を開拓することによって、ミドルマーケット ビジネスを成長させたいと考えていた。同社では、より低価格の POS ソリューションのSaaSバージョンを開発することによってこれを行うことと、可能な限り速やかに発売することを決定した。同社は、リテールPOSソリューションの SaaS バージョンをどう構築したらいいのか分からない、自社のビジネスモデルに対するアーキテクチャの意味が分からない、また、現在のオンプレミス製品の開発にも忙しいので、利用可能なリソースがないと話し、CorSource Technology Group 社に協力を求めた。

#### 現在の技術、優先事項、プランニング

CorSource社によると、同社では、現在のオンプレミス ソフトウェア製品について、Delphi 開発環境、Crystal Report、SharePointなど、やや古い技術を使っていました。

同社の優先事項は、重要度順に、できるだけ早く SaaS POS ソリューションを開発して市場シェアを獲得すること、より最新の技術に移行すること、提携先に対するソフトウェアの拡張性を向上させること、ユーザ インターフェースと機能性を改善すること、モジュラー展開のための製品のセグメンテーションに対応することでした。

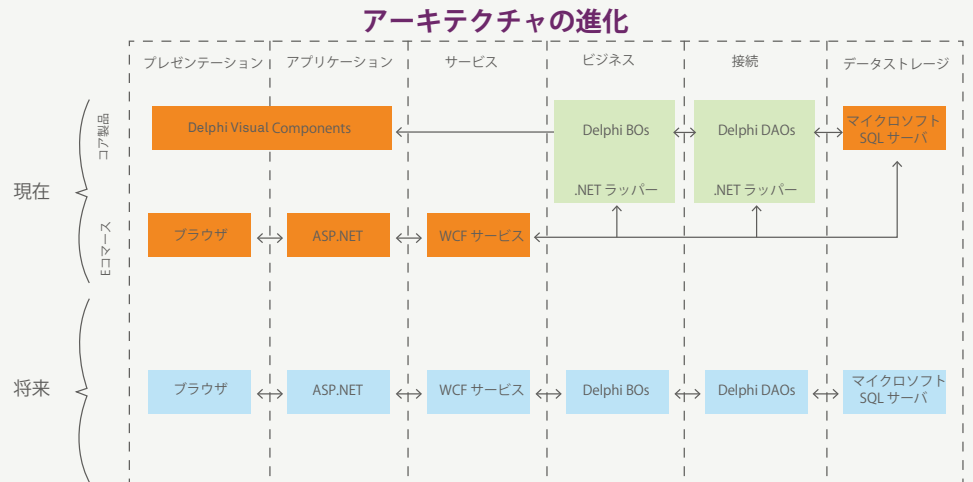
計画では、ソフトウェア製品の設計の大部分を変えず、既存の製品を活用することに決めました。また、当初はコードを変えず、ステートレスアーキテクチャ(クライアントのアプリケーション状態をサーバで管理しない)方式で、既に搭載していたものを再利用する予定でした。現行のプログラムコードとスキルを活用し、従業員へもキャリアパスを提供するため、マイクロソフトの技術を採用し、比較的シンプルなビジネスを目指しました。新規エンドユーザの展開にはすべて、プロフェッショナル サービスの利用を継続することにしました。また、ユーザへよりよい使い勝手を提供するためSaaS製品では現在の GUI を変更することにしました。

## SaaS開発のための主な技術選択

同社では、現在のスキルと、基本的なアーキテクチャの一部が マイクロソフト SQLサーバにあったので、アプリケーションアーキテクチャは、Microsoft.NETにすることを決めました。また、.NETとのアラインメントに最も優れていることと、PHP、Ruby on Rails等のその他のスクリプティングや迅速な開発のための選択肢では、企業向けソフトウェアに必要な堅牢性とフレキシビリティが得られなかったことから、GUIプラットフォームにはMicrosoft ASP .NET を選択しました。データベース管理システムには Microsoft SQL サーバを選択しました。

## アーキテクチャの進化

下のこの図は、同社のオンプレミス POS ソフトウェア ソリューション を SaaS に進化させるためのアーキテクチャのロードマップと、必要なステップを示したものです。



既にいくらかモジュール化のされた現在の製品アーキテクチャを見ると、同社のコア製品が、データベースにマイクロソフト社のSQL サーバを使って Delphi に構築されていることが分かります。次に、現在の製品を更新する際に、.NET ラッパーで Delphi ビジネスオブジェクトとデータオブジェクトと接続する通信サービスにより、ブラウザベースで、Microsoft ASP .NET で書かれているeコマース コンポーネントを追加しました。ビジネスオブジェクトとデータレイヤ全体を同じまま保ったという点で進化的ですが、.NET により、新しい eコマース製品が既存の Delphi アーキテクチャをすべて使えるようになりました。

次に、SaaS のための将来的な製品アーキテクチャに備え、マイクロソフトを使ったフロントエンドと通信を書き換えます。同様に、SaaS 製品に新しい機能を引き続き追加し、.NET ラッパー技術を利用する際に、すべてをゼロから書き換える必要がなく、一度に一つずつ進化させ、展開します。

最終的には、進化した SaaS アーキテクチャ全体が マイクロソフト技術を使って書かれるように、将来の製品にも、マイクロソフトを使ったバックエンドの書き換えが含まれます。さらに、同社のアーキテクチャは実際にリリース計画に沿って策定されます。

## 同社の「進化的な」リリース計画

### リリース1 – Web対応

本質的に成熟度レベル1であるWeb 対応段階では、このPOS プロバイダは、ソフトウェアを Web アクセス製品と言えるものになりたいと考えていました。この最初の段階では、軽いサービスレイヤによって、新しい Web ユーザ インターフェースと、インターフェースの拡張を構築しようとしていました。現在の Delphi ビジネスオブジェクトとデータアクセスオブジェクトを使い、フォーム デザイナーまたはプラグインによってカスタマイザビリティを組込み、オフライン POS 等にデータ インテグレーションを行います。Crystal Reports から Web ベースのレポートに移行しますが、データにはまだ、SharePoint サービスが含まれます。ソフトウェアの新たなインストールはすべて、計画通り、プロフェッショナルなサービス組織が設定します。



## リリース2 –SaaSの発売

SaaSの発売時点では、アーキテクチャに関してまだいくつかの検討事項が含まれていることから、成熟度レベル2のソフトウェアをリリースします。アプリケーションはマルチテナントを認識し、いくつかの機能拡張やエンドユーザの違いによってパーティション可能なインターフェースやエンドユーザごとに分離はされた（ただし同じ）データベースを備え、すべてが順調に動いていることを確保するためのロードテストも含まれることになっていました。ソフトウェアの新たなインストールはすべて、プロフェッショナルなサービス組織が設定を行います。

## リリース3 –完全な SaaS 製品

完全な SaaS 製品のリリースには、マルチテナントデータベース、テナント管理ツール、移行ツール、追加的な機能拡張が含まれます。ソフトウェアの新たなインストールはすべて、プロフェッショナルなサービス組織が設定を行います。

## リリース4 –SaaSへの完全な技術の移行

このリリースは、すべての技術をマイクロソフトに移行し、マイクロソフト方式の単一のSaaSソリューションを持ちます。データアクセスはすべて .NET によって行われ、完全なサービスセットと Active Directory が含まれます。この時点で、真の成熟度レベル4に達したと見なされます。

ソフトウェアのリリース4は最大の開発部分で、最初の3回のリリースと平行して開発を行うことによってのみ、3年という目標以内に達成することが可能です。

## オプションによるリリース5 – 高度なスケーラビリティ

このリリースは、オプションによりデータ アクセスオブジェクトのリファクタリングを行うことによって、高度なスケーラビリティを提供するものです。

このエンドユーザ事例は、あるソフトウェア開発会社が、SaaS への移行において必要とした技術的な経路について示すものです。

同社の約3年に及ぶリリース計画は、レガシー開発プラットフォームの上に構築されたオンプレミス POS ソフトウェア製品から完全にスタッフの揃った SaaS POS ソリューションへの進化が実際に成功した例と言えます。

技術的移行は、必要なことの半分に過ぎません。ソフトウェアへの移行投資の収益化における様々な側面について検討することも重要です。新しい SaaS 製品は、製品自体や、製品のマーケティングにおけるビジネスや業務面を管理すると同時に、簡単にパッケージ化して、エンドユーザに提供することが可能でなくてはなりません。

「Sentinel クラウドを立ち上げて運用するのに、開発者1人が2日費やすだけで済みました。2日間の内訳は、機能させるのに1日、実装を完成させるのに1日です」。

~Centercode

## SaaS 投資を収益化するためのソリューション

Sentinelクラウドサービスは、SaaS プロバイダが多用途に対応するサービスカタログを構築したり、ユーザアクセスのプロビジョニングや許可を行ったり、サービスの利用を測定したり、提供サービスの内容を新たな、また進化する市場機会にすぐに適応することを迅速かつ容易にします。セーフネットのSentinelのみがソフトウェア開発会社に、クラウドベースのアプリケーションのパッケージ化やデリバリ、またPC、ノートパソコン、携帯機器などへのデリバリ後の管理を成功させることができます。ソフトウェア収益化ライフサイクルに完全に一致した Sentinel クラウドサービスは、ソフトウェア開発会社が次のことを行うことを可能にします。



### 定義

フィーチャレベルでサービスカタログと価格決定モデルを定義し、製品の多用途性とビジネス アジリティを向上

### プロビジョニング

サービス契約をすぐに提供し、業務効率を向上させ、手作業によるエラーを削減

### コントロール

ユーザ承認をコントロールし、サービス契約の順守を実現

### 測定

エンドユーザの利用を測定しビジネスインテリジェンスとピリングのサポートにより業務を簡素化し戦略的意思決定能力を向上

### 適合

提供サービスと価格決定モデルをエンジニアリングの必要なく動的に適合させ、市場需要の変化にすぐに対応

「SafeNet はクラウド ライセンシングのユースケースとビジネスモデルに、最新かつ最も包括的に対応しています」

~Forrester Research

## まとめ

SaaS によってソフトウェアが提供される方法は大きく変化し、従来のオンプレミス ソフトウェアのメーカーがソフトウェア製品をクラウドに移行させるケースがますます増えています。メタデータによるマルチテナント効率、高度なスケーラビリティとコンフィギュラビリティという原則に基づいて運営している ソフトウェア開発会社も、現在のエンドユーザや潜在的なエンドユーザに、より安価な方法でソフトウェアを提供することができるようになります。本資料で示すように、進化的か、革新的かのいずれかの移行モデルを使い、フィーチャベースのユーザプロビジョニング、許可、計測、管理を行うことによって、ソフトウェア開発会社は移行や、クラウド サービスデリバリの課題に適切に対応することができるでしょう。

これらの原則を採用することは、オンプレミス製品を理解してクラウドに移行させるための概念的なフレームワークの策定だけでなく、エンドユーザを獲得し、サービスを提供するための方法を変え、最終的にはソフトウェア開発会社のソフトウェアビジネスを成長させる地歩を固める上で役立つでしょう。

「当社のプロジェクトのまとめと計画の検証における御社チームの力には、私たち全員が感銘を受けました。現在、セーフネットの協力のお陰で順調に進行しており、アーキテクチャについて適切な判断ができたと確信しています」

—CorSource 社クライアントPittsburgh Glass Works ITサービス担当取締役  
Darrin Olson

## CorSource Technology Groupについて

CorSource Technology Groupは、動きが早く、競争の激しいソフトウェア開発やITといった世界で、成功するために企業が必要とする、戦略的コンサルティング、開発サービス、技術要員を提供する有数の企業です。

CorSourceのソフトウェア開発の専門家は、フロントエンドのビジネス分析とプランニングに協力すると同時に、プロセスと要件を明確に周知徹底させます。インフラの移行への対応でも、レガシー製品のサポートでも、経験豊かなコンサルタントのチームがクラウド内でのアプリケーションのポジショニングを行います。

CorSourceはオレゴン州ポートランドの本社のほかに、カリフォルニア州サンマテオにオフィス置き、全国のSMBとソフトウェア開発会社に対応しています。詳しくは、[www.corsource.com](http://www.corsource.com) をご覧ください。

## SafeNetのソフトウェア収益化ソリューションについて

Sentinelは、安全でフレキシブルな、将来も使い続けることが可能なソフトウェア収益化ソリューションとして、業界で最も信頼されているブランドです。製品やサービスの堅固なポートフォリオにより、複製や知的財産 (IP) の保護から、製品カタログの管理やエンドユーザの操作性の継続的な改善に至るまで、ソフトウェア収益化のライフサイクルのあらゆる側面に対応しています。SafeNetは、エンドユーザ デバイスに至るあらゆるセールスチャネルを介して、ビジネスモデルのあらゆる組み合わせを使って、パッケージソフトウェアや組込み系ソフトウェア、クラウドサービスなど、あらゆる種類のソフトウェアの収益化を実現する、ソフトウェアライセンスとエンタイトルメント管理のためのソリューションの完全なポートフォリオを提供する、最初で唯一のベンダです。

Sentinelクラウドサービスは、クラウドのためのクラウドで提供される、業界初かつ唯一のソフトウェアライセンスとエンタイトルメント管理のためのソリューションです。

## ソフトウェア収益化のリソース：

無料トライアルを申し込む

[Sentinel クラウドサービス](#) について詳しく知る

「ソフトウェア収益化の状況」に関する調査、[Sentinelクラウドの収益化に関する調査](#) を今見る

## 会話に加わる

→ Sentinel Online  
[www.safenet-inc.com/sentinel](http://www.safenet-inc.com/sentinel)

→ Twitter  
[twitter.com/LicensingLive](https://twitter.com/LicensingLive)

→ LinkedIn  
<http://bit.ly/LinkedInLicensingLive>

→ Sentinel Video Cloud  
<http://sentinelvideos.safenet-inc.com/>

→ LicensingLive  
<http://licensinglive.com/>

→ BrightTalk  
<http://www.brighttalk.com/channel/5572>



お問い合わせ先：全オフィスの所在地と連絡先については、[www.safenet-inc.com](http://www.safenet-inc.com) をご覧ください。  
最新情報：[www.safenet-inc.com/connected](http://www.safenet-inc.com/connected)

© SafeNet, Inc. 禁無断転載 SafeNetとSafeNetのロゴは、SafeNetの登録商標です。その他の製品名は各所有者の登録商標です。WP(EN)-05.14.13

SaaS 実現のための5つのキーステップ：  
オンプレミスソフトウェアをクラウドに移行するためのテクニカルガイドホワイトペーパー