セキュアで高速なアプリケーション開発と運用

Secure and Fast Application Development and Operation

合 原 忠 孝

要 約 日本ユニシスは，サービスビジネスを拡大し，サービス同士がつながり新たな価値を生み出すビジネスエコシステムの形成を目指しており，そのためのプラットフォームとサービス提供に向けてDevOpsセキュア環境を整備した．DevOpsセキュア環境は，「OWASP Top 10」を含む脆弱性を検出する機能として，動的セキュリティ検査，静的セキュリティ検査，構成・脆弱性管理という脆弱性診断を実装した．さらに，開発の初期段階からセキュリティチェックを実行するシフトレフトを取り入れ，脆弱性対策をビルドパイプラインに組み込むことで継続的かつ安定的なセキュリティ対策を実践している．また，セキュアなメガクラウド環境構築のため，CISが公開したAzure向けベンチマークプログラム「CIS Microsoft Azure Foundation」に準拠してIaaS型環境およびPaaS型環境を構築した．DevOpsセキュア環境が提供するIaaS環境は，クラウド推進ネットワーク基盤（CPNI）としてアクセス制御や監査ログ取得の機能をクラウド利用者に提供し，サービスオーナーが迅速かつセキュアにサービスを提供できる環境を実現している．

Abstract Nihon Unisys aims to expand the service business and to form a business ecosystem that links services to create new value, and adjusted the DevOps secure environment for providing platforms and services. DevOps secure environment implemented vulnerability diagnosis called dynamic security inspection, static security inspection, configuration and vulnerability management as a function to detect vulnerabilities including “OWASP Top 10”. Furthermore, we adopt shift left to execute security check from the early stage of development, and implement continuous and stable security measures by incorporating vulnerability measures into the build pipeline. In addition, IaaS type environment and PaaS type environment were built according to the benchmark program “CIS Microsoft Azure Foundation” for Azure released by CIS to build a secure mega cloud environment. IaaS environment provided by DevOps secure environment is by providing access control and audit log acquisition functions to cloud users as a cloud promotion network infrastructure (CPNI), we have realized an environment where service owners can provide services quickly and securely.

1. はじめに

近年，顧客へのソフトウェア提供形態は，製品の提供（Software as a Product，以下SaaP）だけでなく，サービス利用型の提供（Software as a Service，以下SaaS）が台頭している．市場の構成比は，カスタマイズ性の高さなどの利点により依然としてSaaPの比率が高いものの，企業内のクラウドコンピューティングの利用が一般化してきており，SaaSの市場は急速に拡大している．SaaSでは，MVP（Minimum Viable Product）[*1]の投入により仮説と検証を繰り返してサービスを成長させていくリーン・スタートアップ等のアプローチを適用し，サービス提供者は自身のサービスを早期に立ち上げ，継続的に改善して提供することがで

きる．

継続的な改善として新しい機能を安全かつ自動的にサービスへ統合していくため，継続的インテグレーション（Continuous Integration，以下 CI），継続的デリバリー（Continuous Delivery，以下 CD），メトリクス採取やフィードバックなど DevOps の開発手法の実践が求められる．サービスの早期立ち上げで課題となるセキュリティの要素を DevOps に追加する試みが DevSecOps であり，日本ユニシス株式会社（以下，日本ユニシス）では，DevSecOps によるサービス開発，運用を実践するための仕組みとして，DevOps セキュア環境を整備している．

本稿では，日本ユニシスが目指すビジネスエコシステムとその中核となるプラットフォーム上に構築するサービスに焦点を当て，DevOps セキュア環境について述べる．まず 2 章では，日本ユニシスが目指すビジネスエコシステム，およびビジネスエコシステムを形成するプラットフォームと DevSecOps の位置付けについて述べ，次に 3 章では，セキュアなサービスの早期立ち上げにおけるセキュリティ課題を整理して，4 章では，DevOps セキュア環境として実装する対策について論じる．

2. ビジネスエコシステムを支える DevSecOps

本章では，日本ユニシスが目指すビジネスエコシステム，およびビジネスエコシステムを形成するためのプラットフォームとサービスの提供に向けた取り組みにおける DevSecOps について述べる．

2.1 日本ユニシスが目指すビジネスエコシステム

ビジネスを取り巻く環境は構造の変化がますます激しくなり，企業単体での競争力強化が難しくなっている．ビジネスエコシステムは，参加者を増やしながら，プラットフォーム上に次々とサービスと利用者が増えるほど収益が拡大するモデルであり，サービスの早期立ち上げと迅速なビジネス連携を実現しなければならない．日本ユニシスは，コーポレートステートメントとして「Foresight in sight」を掲げ，サービスビジネスを拡大し，サービス同士がつながり新たな価値を生み出すビジネスエコシステムの形成を目指している．ビジネスエコシステムでは，利用者はプラットフォーム内外の各サービス間の連携を意識することなく，相互に乗り入れているサービス群を横断的に利用することで，単一のサービスだけでは提供できなかった価値を獲得できる．日本ユニシスがオーナーとなるサービスでは，日本ユニシス自身が要件を決め，迅速かつ小さくサービスを市場に投入し，ビジネス状況の変化に応じて修正とリリースを繰り返して成長させることにより，ビジネスエコシステムの形成に寄与している．

2.2 サービスをセキュアかつ高速に開発・提供する DevSecOps

サービスを早期に立ち上げ，継続的な改善を実施していくためには，小さな変更を頻繁にリリースするアジャイル型のサービス開発が不可欠であり，アジャイル型の開発を実践するには，DevOps の開発手法が有効である．より信頼性の高いアプリケーションをより早く頻繁にリリースする DevOps は，開発チーム（Development）と運用チーム（Operations）が連携して協力する仕組みである．また，DevOps にセキュリティ対策を付与した考え方が DevSecOps であり，DevSecOps によるサービスの開発・提供を実践することで，サービス開発・提

供の迅速さを犠牲にせずセキュリティ対策を講じることができる.

3. サービスの早期立ち上げにおけるセキュリティ課題

本章では，サービスを早期かつセキュアに立ち上げるための課題，および継続的な改善を迅速に実施していく際の課題について整理する.

3.1 メガクラウド環境の活用における課題

近年，海外・国内ともに，パブリッククラウド，特にメガクラウドと呼ばれる Microsoft Azure（以下，Azure），Amazon Web Services（以下，AWS）の利活用が進んでいる．メガクラウドの活用はコンピューティングリソースの調達負荷を軽減することでサービスの早期立ち上げに寄与する一方，メガクラウドサービス提供者のシステム基盤はインターネットに公開される環境のため，外部からのアクセスに対するセキュリティ対策が不可欠となり，構築されたメガクラウド環境へのセキュアな接続経路が求められる．また，メガクラウドでは仮想マシン（Virtual Machine）や仮想ストレージ，仮想ネットワークなど，オンプレミス環境でのコンピューティングリソースを代替するサービスにとどまらず，ミドルウェアのレイヤーまでを提供する PaaS（Platform as a Service）型のサービスや，必要な機能を必要な分だけサービスとして利用できるようにしたSaaS型のサービスも提供されている．SaaS型やPaaS型のサービスを利用する場合には，仮想マシンなどのコンピューティングリソースとは異なるセキュリティ対策が求められる.

3.2 セキュアなアプリケーションを開発するための課題

サービスを早期かつセキュアに立ち上げ，顧客要求・ビジネス要求に柔軟かつ迅速に，継続的な対応を行うには，従来の開発手法であるウォーターフォール型ではなくアジャイル型の開発手法を用いる．しかし，アジャイル型の開発手法では，セキュリティ対策がウォーターフォール型と異なる．本節では，両者の課題を整理してセキュアなアプリケーション開発のための対策を述べる.

3.2.1 ウォーターフォール型開発におけるセキュリティ課題

ウォーターフォール型開発は，あらかじめ工程を段階的に進捗させる開発計画を立案し，全体の機能設計を完了させてから機能を実装する開発手法である．ビジネス環境の変化が激しい近年の状況下では，サービスを企画した時点では有益だったアイデアでも，サービス提供時には価値が低減してしまうリスクがある。変化に応じた最適な形を目指して仕様を変更し，計画を適宜修正する俊敏な対応が求められるが，ウォーターフォール型開発では進行中の計画を俊敏に変更することは困難であり，仕様や計画の変更はプロジェクトに大きなインパクトを与える．また，セキュリティ観点のテストとしてセキュリティ専門家による脆弱性検査等が実施される時期は，サービスが提供する機能を一通り実装した後，すなわちサービス提供開始の直前になることが多い．サービス提供開始直前で検出された脆弱性の対応には膨大なコストがかかり，期日に間に合わない場合にはサービス提供の開始が遅れるリスクも存在する.

3.2.2　アジャイル型開発におけるセキュリティ対策

アジャイル型開発は，顧客に価値を提供できる最小限の機能である MVP を作り，それを基に利用者からのフィードバックを集め，外部のビジネス環境の変化に対応しながら，仕様や機能の改善を継続することで最終成果物を目指すアプローチの開発手法である．アジャイル開発では，「イテレーション（反復）」という短い期間で開発・リリース・改善を繰り返すため，不具合や仕様変更が発生した場合でも，次のイテレーションに組み込むことで，手戻りの工数を最小限に抑えることができる．その結果，仕様変更や追加にも柔軟に対応でき，利用者にとってのサービスの価値を向上させる取り組みを俊敏に実施することができる．そのような高速かつ安定したリリースプロセスを継続して実践するためには，開発チームである「Development」と運用チームである「Operations」が協力関係を築き，DevOps の CI/CD の仕組みを構築しておくことが重要である．

CI/CD の仕組みによって高速かつ安定したリリースプロセスを実践することで，サービスのリリース回数を上げることができるが，ウォーターフォール型開発と同様，サービス提供機能がそろった後に脆弱性診断をリリースプロセスに組み込むことは，現実的に難しい．そのため，ウォーターフォール型開発と同様のセキュリティ対策のみでは，アジャイル型の開発の高速なリリースに対応できず，リリースが遅れる可能性がある．アジャイル型開発では，セキュリティ専門家による脆弱性診断だけに頼らず，高速なリリースを行うリリースプロセスの仕組みの中に，自動化された脆弱性対策の仕組みを取り込まなければならない．

4.　DevOps セキュア環境

前章にあげたサービスの早期立ち上げにおけるセキュリティ課題を解決するため，日本ユニシスでは，各課題への対策を実装した DevSecOps のプロトタイプ環境（以下，DevOps セキュア環境）を設計，構築し検証を行った（図 1）．本章では，DevOps セキュア環境で実装した対策について述べる．

4.1　セキュアなアプリケーション開発

DevOps セキュア環境では，アプリケーション開発のできるだけ早い段階から多角的な脆弱性検査を実施するための仕組みを，アプリケーション開発のための環境に組み込んだ．また，アジャイル型の開発で実践される高速かつ継続的なリリースに対応すべく，自動化された脆弱性検査を実施し，セキュアで高速なリリースを安定的に実践することを目指した．

4.1.1　脆弱性診断機能

アプリケーションの脆弱性は，アプリケーションを実装するための開発言語やアーキテクチャ等の変化に伴い，常に変化している．すべての脆弱性を完全かつタイムリーに根絶することは困難であるため，各脆弱性における攻撃者やその攻撃手法，セキュリティ上の弱点となる箇所などに関する可能性を評価したうえで，技術面やビジネス面で組織への影響を考慮し，対応を判断しなければならない．このようなアプリケーションの脆弱性に対して，ウェブアプリケーションセキュリティをとりまく課題の解決を目的とする国際的でオープンなコミュニティである OWASP（Open Web Application Security Project）[*2] は，最も重大な Web アプリケーションセキュリティリスクを特定することに焦点を当てて整理した結果を「OWASP Top

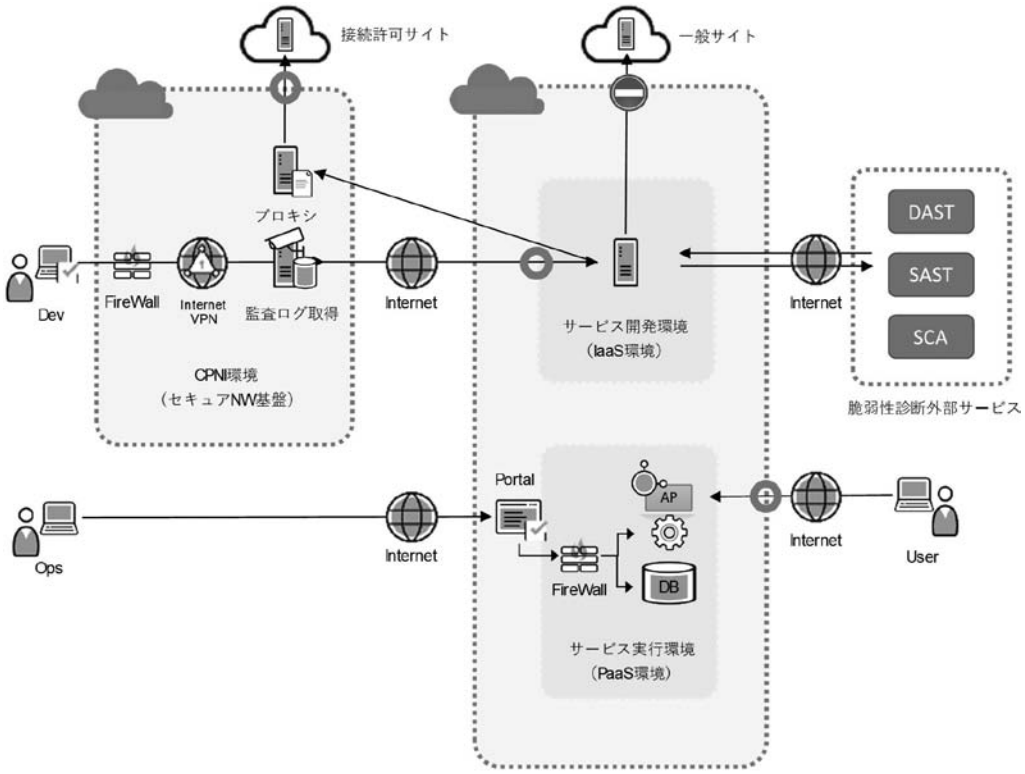

図 1　DevOps セキュア環境　概要図

表 1　OWASP Top 10 アプリケーションセキュリティリスク-2017

ID	脆　弱　性
A1:2017	インジェクション
A2:2017	認証の不備
A3:2017	機微な情報の露出
A4:2017	XML 外部エンティティ参照（XXE）
A5:2017	アクセス制御の不備
A6:2017	不適切なセキュリティ設定
A7:2017	クロスサイトスクリプティング（XSS）
A8:2017	安全でないデシリアライゼーション
A9:2017	既知の脆弱性のあるコンポーネントの使用
A10:2017	不十分なロギングとモニタリング

10」[*3] として公開している（表 1）．アプリケーションの脆弱性への対策では，まずは最新版の「OWASP Top 10」にリストアップされている脆弱性に対応することが費用対効果の高い対策となる．

また，脆弱性は刻々と変化するため，「OWASP Top 10」としてリストアップされた脆弱性

以外にも検出できるよう，多面的な脆弱性診断を実施できる仕組みを検討しておくべきである．DevOps セキュア環境では，最新版の OWASP Top 10 を含む脆弱性を検出する機能について，外部の SaaS 型のサービスを利用する．外部の脆弱性診断サービスは，提供される脆弱性検査で検出される項目が日々アップデートされることを選定時の指標とし，多面的な脆弱性診断を行うため，以下の脆弱性診断を実装する方針とした．

1）動的セキュリティ検査（Dynamic Application Security Testing：DAST）
外部から Web アプリケーションにさまざまな入力を与え，その出力結果を基に脆弱性の有無を検出する検査機能である．実際に悪用され得る脆弱性を効果的に検出することができるが，ソースコードの脆弱性の存在箇所までは判断できない．また，アプリケーション全体を検査する際には，クローリングのためのシナリオの設計，設定を実施するため，検出結果の精度に加え利用時の設定容易性も考慮しなければならない．

2）静的セキュリティ検査（Static Application Security Testing：SAST）
アプリケーションのソースコードを解析し，脆弱性を検出する検査機能である．ソースコードの脆弱性の存在有無や箇所を判断できるが，悪用の可能性まで判断することは難しく，過剰な脆弱性の検知が大量に報告されることがある．そのため，検出結果については，過検知および誤検知の精度も選定時の指標となる．

3）構成・脆弱性管理（Software Composition Analysis：SCA）
脆弱性が報告されているサードパーティー製のライブラリを使用していないかを検査する機能である．オープンソースソフトウェア（以下，OSS）のライセンスポリシーにおける違約も検出する機能を提供しているサービスもあり，タイムリーな脆弱性情報の更新に加え，検出する内容が選定時の指標になる．

4. 1. 2　シフトレフトによる脆弱性の検出

ウォーターフォール型開発では，脆弱性がサービスインの直前で検出される傾向にあり，致命的な脆弱性への対応はプロジェクトの大きな負担となる．この問題を解決するために，アプリケーション開発におけるセキュリティ対策として「Shift Left」（以下，シフトレフト）という概念が提唱されている（図 2）．シフトレフトでは，開発の初期段階から脆弱性診断などのセキュリティチェックを実行する仕組みを実装し，後工程での影響度を軽減することを推奨しており，結果的にアプリケーションの開発コストの低減に寄与する．

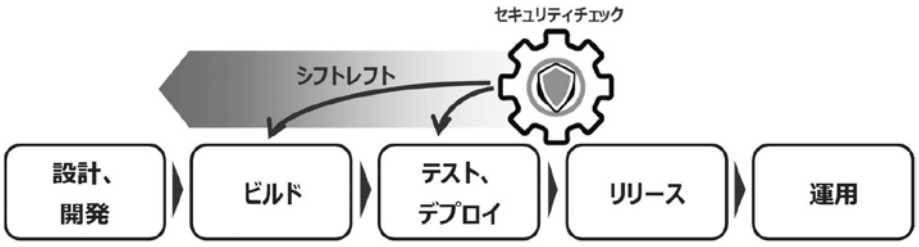

図 2　シフトレフト概念図

4. 1. 3　ビルドパイプライン組み込みによる自動実行

アジャイル型開発では，DevOps の CI/CD を構築し，サービスをリリースするためのリリー

スプロセスを実践する．自動化された脆弱性対策をビルドパイプラインに組み込むことで，高速リリースの実践と併せてセキュリティ課題への対策を継続的かつ安定的に実践することができる．

DevOps セキュア環境で実践する脆弱性診断の各機能は，ビルドパイプラインを形成するコンポーネントの一つと位置付け，代替できる外部の SaaS 型のサービスを利用している．これにより，利用中の脆弱性診断サービスが提供を停止した場合でも，他の外部サービスをビルドパイプラインに組み込みなおして，アプリケーション開発を継続することができる．

4.2　セキュアなメガクラウド環境構築

メガクラウドは，サービスを提供するプロバイダーがシステム基盤をコントロールしており，プロバイダーと契約することで利用者は自らハードウェアなどを調達しなくても即座にコンピューティングリソースを利用できるメリットがある．しかし，メガクラウドのシステム基盤はインターネットに公開される環境のため，プロバイダーが講じるセキュリティ対策だけでは不十分であり，利用者がメガクラウド環境をセキュアな形で利用できるようセキュリティ対策を実施しなければならない．

DevOps セキュア環境は，セキュリティの対策基準において，国際インターネット・セキュリティ組織（Center for Internet Security，以下，CIS）[*4] が公開するベンチマークプログラムに準拠して構築した．CIS は，セキュリティの促進を目的とした米国の非営利団体で，セキュリティの専門家により精査されたセキュリティ基準をセキュリティベンチマークプログラムとして公開している．CIS のセキュリティベンチマークプログラムは，セキュリティを評価して強化できるよう明確に定義された，公平でコンセンサスベースの業界のベストプラクティスとして提供されている（表 2）．CIS が公開している各分類におけるシステムの設定項目において，推奨される設定を行うことにより，セキュアな環境を作ることができる．

DevOps セキュア環境では，コンピューティングリソースを Microsoft 社が提供する Azure を利用して構築している．CIS は Azure 向けのベンチマークプログラムとして「CIS Microsoft Azure Foundations」[*5] を公開しており，DevOps セキュア環境でも，このベンチマークを環境構築時の指針としている．

表 2　CIS Microsoft Azure Foundations Benchmark におけるチェック項目

ID	Recommendations
1	Identity and Access Management
2	Security Center
3	Storage Accounts
4	SQL Services
5	Logging and Monitoring
6	Networking
7	Virtual Machines
8	Other Security Considerations
9	AppService

DevOps セキュア環境では，アプリケーション開発のための CI/CD パイプラインや DevOps における開発に必要なサービス群を，コンピュータシステムを構築および稼働させるための基盤（Infrastructure as a Service，以下 IaaS）を利用して構築している．さらに，サービスを提供する実行環境として，メガクラウドが提供する PaaS 型のサービスを組み合わせた環境を構築している．前者は IaaS を利用した環境であることから IaaS 型環境，後者は PaaS を利用した環境であることから PaaS 型環境となり，いずれも CIS ベンチマークに準拠する環境として設計，構築している．今後も CIS ベンチマークが更新された折には，DevOps セキュア環境も追随し，環境をアップデートしていく．以下に，IaaS 型環境，PaaS 型環境，それぞれのセキュリティ対策を示す．

1）IaaS 型環境

メガクラウド環境上に仮想マシンを構築し，アプリケーションの実行およびシステム運用保守に必要な機能を構築した環境である．仮想マシンから構築するためサービスの要件に応じて柔軟に対応できるが，OS やミドルウェアの更新プログラム適用やエンドポイントセキュリティ対策の導入およびパターンファイルの更新など，仮想マシンの運用保守作業が発生する．

2）PaaS 型環境

メガクラウドが提供する PaaS で構成した環境である．段階的なリソースの増加や機能拡張にもメガクラウドのポータル画面から容易にスケールアウト，機能追加できる．メガクラウドの PaaS が提供する基本的なシステム運用の機能を利用することができ，仮想マシンの保守は不要のため，保守コストを低減できる．また，仮想マシンに侵入されるセキュリティリスクはメガクラウド提供者側で担保しており，セキュリティ対策コストを低減できる．

両環境を比較した場合，PaaS 環境はメガクラウドが提供するサービスとしての機能制限があるものの，運用保守やセキュリティ対策の負荷，および利用料金などのコスト面から有利と判断し，DevOps セキュア環境では，PaaS 環境を積極的に利用し，機能的に要件を満たさない場合にのみ IaaS 環境を検討する方針とした．

4. 3　セキュアな共同利用型接続ネットワーク基盤

PaaS型環境として構築する場合，メガクラウドが提供するPaaSのリソースへのアクセスは，インターネット回線を経由してメガクラウドの管理ポータルあるいはコマンドラインから実施することになるが，リソースへのアクセス時にはユーザーアカウントの認証を求められ，アクセス権のないユーザーがリソースにアクセスすることはできない．また，実施した操作はサービスプロバイダー側で操作履歴が残されており，有事の際の監査ログとして利用できる．

一方，IaaS型環境の場合は，インターネット回線を経由してアクセスする仮想マシンのため，適切なアクセス制御を利用者側で実施しなければならない．また，実施した操作の履歴はメガクラウド側で情報を取得できないため，その仕組みも利用者側で検討しなければならない．メガクラウドを活用してコンピューティングリソースを安価かつ迅速に調達できても，上記のアクセス制御や証跡取得の仕組みを構築するための機器を調達し，設置や実装がボトルネックになっては，サービスの早期立ち上げの価値を低減させる．また，個々のサービスを開発するプロジェクトが，上記の仕組みを個別に構築することで，ネットワーク環境のセキュリティ強度

の品質が低下するリスクもある.

このような背景から, DevOpsセキュア環境では, 様々なサービス開発プロジェクトがセキュアにメガクラウド接続できる共同利用可能なネットワーク基盤（クラウド推進ネットワーク基盤：Cloud Promoting Network Infrastructure, 以下 CPNI）を構築した. IaaS 型環境の仮想マシンに日本ユニシスとグループ会社（以下, 日本ユニシスグループ）のイントラからセキュアに接続を行うための共有利用型ネットワーク基盤として, DevOps セキュア環境以外の開発案件も含め 2018 年度から利用を開始している（図 3）.

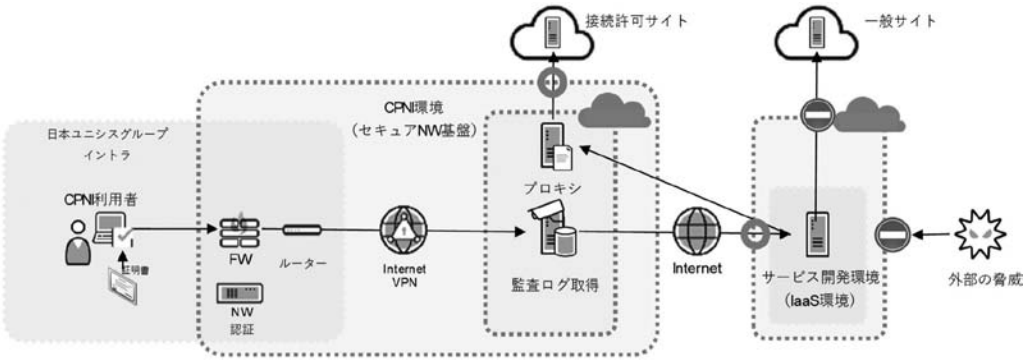

図 3　CPNI 環境概念図

DevOps セキュア環境では, アプリケーション開発を行う環境として, OSS を活用した CI/CD パイプラインを IaaS 型環境上に構築しているため, CPNI 環境を利用して開発拠点からの接続を行い, 仮想マシンに対して実施した操作履歴を取得している. また, プロキシサーバーを用いたアクセス制御も行っている. 以下の 2 項で解説する.

4. 3. 1　監査ログの取得

DevOps セキュア環境の IaaS 型環境は, サービス利用者に対してサービスを提供する以外の通信経路からの接続を閉じており, サービス利用者は CPNI 環境を経由して IaaS 環境に接続する. ユーザーアカウントと端末に紐づくクライアント証明書を接続端末に導入して, 承認済みのアカウントが承認済みの接続端末を利用する場合にだけ接続が許可される. さらに, CPNI 環境を経由してメガクラウド上の IaaS 環境へ接続する場合, IaaS 環境の仮想マシンに直接接続することは許可されておらず, CPNI 環境に構築されたサーバーを踏み台として経由し, メガクラウド上の接続先となる仮想マシンの OS 種別に応じた方法（Linux サーバーの場合は ssh 接続/Windows サーバーの場合はリモートデスクトップ接続）で接続する. 踏み台のサーバーには, 接続時に実施した操作の記録を取得・保持しているため, 有事の際にはこの操作記録を監査証跡として利用できる.

4. 3. 2　インターネットへのアクセス制限

CPNI 環境では, 各案件の開発環境からセキュアにインターネットへのアクセスを行うためのプロキシサーバーの機能を提供する. プロキシサーバーはメガクラウド上に構築した IaaS 型環境からインターネットに接続可能な接続先リスト（以下, ホワイトリスト）を保持してお

り，ホワイトリストを一元的に管理することで，各サービス開発案件のIaaS型環境が外部すなわちインターネットへセキュアに接続できる環境を提供している．また，プロキシサーバーは，ユーザー認証により，CPNI環境の利用者として登録されているユーザーだけが利用可能となっている．

4.4　今後の取り組み

サービスを運営していく上で，市場や顧客ニーズの変化に追随し，サービスの価値を継続的に向上させていくだけではなく，高いセキュリティを維持し，ユーザーの情報や資産を保護することは，サービス運営者にとって重要な責務である．そのためのセキュリティ確保プロセスの標準化やツールの積極的な活用による自動化が重要となる．日本ユニシスは，日本ユニシスグループの標準ビジネスプロセスを基に，サービス提供型のビジネスに対応するプロセスを定義し，セキュリティ対策ガイド，システム開発セキュリティプロセスを実践するための取り組みを行っている．サービス提供型のビジネスにおいて，アジャイル型の開発でサービスのリリースを高速化していく場合に懸念されるセキュリティのリスクに対し，DevOpsセキュア環境では，脆弱性を検出するための仕組みの自動化やサービスを開発・提供するためのセキュアな環境を構築する仕組みについて検討してきた．

今後は，日本ユニシスグループの標準ビジネスプロセスとして定義されたサービス提供型のビジネスに対応するプロセスとの連携を深めることで，日本ユニシスグループ内にDevOpsセキュア環境を普及させる取り組みを実践していく．アジャイル型の開発を実践するには，組織としての標準化プロセスに準拠するだけでなく，開発チームと運用チームが協調するDevOpsの概念を実践する必要がある．また，セキュリティのリスクは日々変化しており，標準的なプロセスや環境の利用だけでは対応が十分ではない場合もある．DevOpsでは，単なる自動化による効率化だけでなく，プロジェクトチームのメンバーが協働できることを目指す文化がチームに醸成されていることが重要である．さらにDevOpsにセキュリティの観点を付与したDevSecOpsでは，プロジェクトチーム内に，そのサービスのセキュリティに対して責任を持ち，セキュリティ対応におけるプロジェクトチーム内の協働を推進する役割を持つメンバーが必要となる．今後のDevOpsセキュア環境を活用したプロジェクトを通じて，プロジェクトチーム内の協働を推進するメンバーを育成していく．

5. お　わ　り　に

2015年に国連が批准したSDGs（Sustainable Development Goals）の影響や社会的な構造の変化などにより，企業は社会的な課題を解決する動きを加速させる必要に迫られている．企業単体での競争力強化ではなく，ビジネスエコシステムの形成や発展によって，顧客価値の向上を迅速かつ継続的に目指すことが重要である．日本ユニシスグループは，様々な業種・業態の企業をICTで支え続けてきた経験を活かしてプラットフォームを提供することにより，ビジネスエコシステムの形成と発展に貢献していきたい．

最後に，本稿の執筆にあたり，多くの方々にご助言とご指導を頂いた．この場を借りて深く御礼申し上げる．

＊1　MVP（Minimum Viable Product）：構築-計測-学習のループを回せるレベルの製品で，最小限の労力と時間で開発できるものを言う．
＊2　OWASP（Open Web Application Security Project）：The OWASP Foundation を運営母体とする国際的でオープンなコミュニティ．2001 年に設立され，2004 年 4 月 21 日よりアメリカ合衆国にて政府認定された NPO．現在，全世界に 120 以上の個別テーマのプロジェクトがあり，日本では，2011 年に OWASP Japan が発足．
＊3　OWASP Top 10：OWASP で 2014 年に発足した OWASP Internet of Things プロジェクトの成果物の一つ．ほぼ 3 年毎にリリースされており，本稿執筆時点の最新版は「OWASP Top 10 - 2017」．
＊4　CIS（Center for Internet Security）：セキュリティ構築の不具合によるビジネスやイーコマースにおける組織のリスク軽減を支援する米国の非営利組織．
＊5　CIS Microsoft Azure Foundations：本稿執筆時点の最新版は「v1.1.0 - 02-15-2019」

参考文献

［1］ エリック・リース著，井口耕二訳，「リーン・スタートアップ」，日経 BP 社，2012 年 4 月
［2］ 河村聖悟/北野太郎/中川貴尋/日下部貴章/株式会社リクルートテクノロジーズ著，「DevOps 導入指南」，株式会社翔泳社，2016 年 10 月
［3］ ベッツィ・ベイヤー/クリス・ジョーンズ/ジェニファー・ペトフ/ナイル・リチャード・マーフィー著，澤田武夫/関根達夫/細川一茂/矢吹大輔訳，「SRE サイトリライアビリティエンジニアリング」，株式会社オライリー・ジャパン，2017 年 8 月
［4］ Havard Myrbakken/Ricardo Colomo-Palacios，「DevSecOps: A Multivocal Literature Review」，Conference Paper，2017.9

執筆者紹介

合　原　忠　孝（Tadataka Gobaru）
2003 年日本ユニシス(株)入社．地銀システムなどの大規模開発案件の基盤・運用設計および，日本ユニシスグループのマネージドクラウドサービス U-Cloud に従事した後，2017 年よりプラットフォームサービスにおける企画・設計に従事．