情報処理学会論文誌 Vol.62 No.3 810–817 (Mar. 2021)

クラウドメールサービスのアドレス管理委譲補完方法

嶋吉 隆夫[1,a)] 笠原 義晃[1] 清家 史郎[2] 藤村 直美[1]

受付日 2020年6月22日, 採録日 2020年12月1日

概要：組織内で電子メールサービスを提供する際にクラウドメールサービスを利用することは，運用管理コストの削減に効果的だとされている．しかし，クラウドメールサービスでは一般的に，管理対象メールアドレスを限定した管理権限を設定する機能を持たないことから，大規模組織全体でクラウドメールサービスを利用する場合に，内部組織の管理者へ管理を部分的に委譲できず，組織全体のサービス管理者に負荷が集中するという課題がある．本研究は，この課題を解決するため，メールアドレス管理の内部組織への委譲を実現することを目的とする．本稿では，メールアドレス管理委譲の実現手法，および，クラウドメールサービスを補完するシステムについて述べる．提案手法では，メールサービスのアカウントの管理からメールアドレスの管理を分離し，内部組織別に設定されたインターネットドメイン名ごとに，メールアドレスの管理を委譲する．本稿で述べるメールアドレス管理システムはマイクロサービスアーキテクチャに基づくサーバレス構成の設計を採用する．マイクロサービスアーキテクチャとサーバレス構成により，システム運用管理コストの削減，および，保全性と拡張性が達成される．さらに，Identity as a Service を用いて，認証と認可の処理，および，管理権限情報の格納を行うことで，システムアカウントの機密性とセキュリティを確保できる．

キーワード：実用システム，システム運用管理，SaaS，マイクロサービス，サーバレス

A Supplementation Method for Delegation of Address Management on a Cloud Email Service

Takao Shimayoshi[1,a)] Yoshiaki Kasahara[1] Shiro Seike[2] Naomi Fujimura[1]

Received: June 22, 2020, Accepted: December 1, 2020

Abstract: In providing organizational email service, the use of cloud email services contributes to reducing operation and management costs, generally. In most cloud email services, the management of a certain part of email addresses is not delegatable to administrators of organizational units, but this causes operational load concentration on organizational service administrators in large-scale organizations. For solving this issue, this study aims to enable delegations of email address management to administrators of organizational units. This article introduces a method for delegation of email address management and a supplemental system onto a cloud email service. The proposed method divides the management of email addresses within internet subdomains for organizational units from account administration and delegates the email address management to subdomain administrators. The present system for email address management is designed based on microservice architecture and configured in serverless. This microservice architecture and serverless configuration reduces operation and management costs and contributes to the maintainability and extensibility of the present system. Furthermore, the use of an identity-as-a-service for authentification, authorization and storing privilege information ensures confidentiality and security of system accounts.

Keywords: practical system, system operation and administration, SaaS, microservices, serverless

1. はじめに

組織内で情報サービスを提供する際に，Software as a Service (SaaS) [1] を利用することが一般化している．SaaS を利用する場合，従来のようにオンプレミスシステムを運用する場合と異なり，サービスプロバイダにより共有の

[1] 九州大学情報基盤研究開発センター
Research Institute for Information Technology, Kyushu University, Fukuoka 819–0395, Japan
[2] 株式会社 Fusic 技術開発部門
Technology Development Department, Fusic Co., Ltd., Fukuoka 810–0001, Japan
[a)] simayosi@cc.kyushu-u.ac.jp

サービス基盤が提供され，独自にハードウェアやオペレーティングシステム（OS），システムソフトウェアを管理する必要がないことから，運用管理が効率化され，コストが低減できる [2], [3]．特に電子メールサービスについては，組織利用のために SaaS として提供されるクラウドメールサービス（以下，SaaS メールサービス）の利用は効果的である．近年は，フィッシングメールによるパスワードなどの詐取，メール添付ファイルによるマルウェアの配送，標的型攻撃，ビジネスメール詐欺など，電子メールがサイバー攻撃の主要な手段として用いられており [4]，電子メールサービスの運用にはセキュリティ対策が必須である．SaaS メールサービスでは一般的に，ウィルス，フィッシングメール，スパムメールの検査や，アカウント不正利用への対策が提供されており [5], [6]，独自にセキュリティ対策を行う場合に比べて運用管理負担が低減される．

一方で，SaaS メールサービスの利用には管理上の課題もある．ここで本稿では正確を期して，たとえば複数部局から構成される総合大学といった，複数の組織単位から構成される大規模組織のことを機関，大規模組織を構成する組織単位のことを組織と呼ぶ．SaaS メールサービスは一般的に，利用者である機関のサービス全体の管理者（以下，本稿ではサービス管理者と呼ぶ）だけがメールアドレスの作成や削除を実行できる方式である．しかし，この方式では，組織ごとに人員管理を行うような機関の全体でサービスを利用する場合に問題が生じる．機関内の組織では，たとえば総合大学の部局における客員教員などのように，機関外の人間に組織の役職とともにメールアドレスを付与したい場合や，組織内の一時的グループのためにメールアドレスを発行したい場合などがあるが，そのようなメールアドレスについても機関全体のサービス管理者が管理する必要があり，機関全体のサービス管理者に管理負荷が集中する．

機関内の組織別に設定された DNS サブドメイン [7] ごとにメールアドレスを管理するという伝統的なメールサービス構成の場合は，メールアドレスの管理権限を階層的に委譲することが可能である．しかし，この伝統的構成ではメールアドレスだけでなくメールサービスの管理も機関内で分散してしまい，アカウント漏洩のリスクが増大することや，複数組織に所属する構成員が複数アカウントを管理する必要があることなどの問題がある．なお，階層的に委譲された DNS サブドメインに対して個別に SaaS メールサービスを利用する場合でも，問題は同じである．

また，伝統的構成によるメールサービス運用から，機関全体で SaaS メールサービスを利用する形態に移行する場合には，メールアドレスの継続性が課題となる．メールアドレスは全世界で一意な一種の個人識別子であり，伝統的構成ではメールアドレスのドメインは組織への所属を表現する．また，メールアドレスは連絡先として印刷物に掲載される場合も多い．それゆえ，機関全体で SaaS メールサービスに移行したとしても，旧来のメールアドレスを使い続けたいという要望は大きい．

そこで，本研究の目的は，SaaS メールサービス利用における上記の課題を解決するため，集中管理方式の SaaS メールサービスにおけるメールアドレス管理権限の機関内組織への委譲を実現することである．本稿では，SaaS メールサービスである Exchange Online を対象として，権限委譲のための手法を提案し，その手法を実現するためのメールアドレスを管理するシステムの設計について述べる．まず，2 章で要件と制約について整理したのち，3，4 章で提案する実現手法とシステムについて説明し，5 章でその手法とシステムについて考察を加える．

2. 要件と制約

2.1 機能要件

本研究で実現を図る機能要件について以下に整理する（**図 1**）．ここで，組織ごとに割り当てられたインターネットドメイン名（以下，組織ドメイン）をドメインに持つメールアドレス（以下，組織アドレス）を権限委譲の対象とする．

メールの配送に関して，まず，ある組織に属する機関構成員が，その組織アドレスでメールを送受信できる必要がある．また，組織ドメインにおいて転送アドレスを設定可能とする．ここでの転送アドレスでは，Unix 系 OS において/etc/aliases に設定するようなメールエイリアス [8] と同等機能，つまり，組織アドレスから機関内および機関外のメールアドレスへの転送，および，組織アドレスから複数のメールアドレスへの転送を実現することを考える．

機関全体で利用する SaaS メールサービスにおける利用者アカウント（以下，機関アカウント）は，機関全体で一元的に管理する．組織ドメイン別にはアカウントを発行せず，機関全体のサービス管理者が集中管理することとする．また，機関の構成員は，機関アカウントを用いた認証により組織アドレスを利用可能とする．

組織アドレスの管理を，組織ドメインごとに定められた 1 人以上の管理者（以下，組織ドメイン管理者）へと委譲する．ここで，組織ドメイン管理者は，組織アドレスの発行

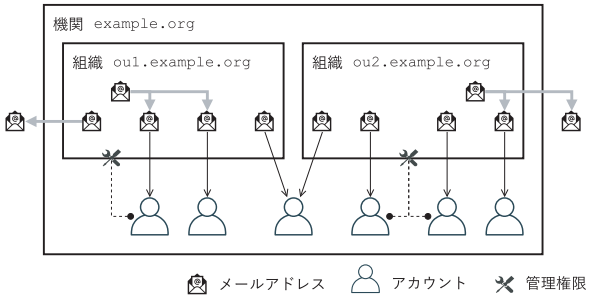

図 1 実現対象の概念図
Fig. 1 Objective configuration.

と廃止，発行済み組織アドレスの一覧の取得，個人用組織アドレスと機関アカウントとの対応の管理，組織ドメインの転送アドレスの転送先の管理を実行できる必要がある．

2.2 対象 SaaS の機能・制約

本稿での対象である，Microsoft が提供する Exchange Online の機能および制約 [9] について説明する．なお，ここで述べる説明は，本稿執筆時点のものである．Exchange Online は，複数のサービス利用組織で資源が共有されるマルチテナント型 [1] の SaaS メールサービスであり，テナント別に独立して構成および管理運用が可能である．電子メールの送受信機能はウェブアプリ，および，メールユーザエージェントから利用できる．受信メールに対する送信元レピュテーションや，ウィルスおよびスパム検査機能を標準で利用できる．

規定でテナントに割り当てられるドメイン名のほか，複数の独自ドメイン名を登録でき，登録ドメイン名はテナントのメールアドレスに利用できる．登録ドメイン名に対して，DNS の MX レコード [10] に設定するための，SMTP [8] のメール転送エージェント（MX ホスト）の完全修飾ドメイン名（Fully Qualified Domain Name）[11] が用意される．また，複数のメールアドレスをメンバとして登録しメール転送できる「配布グループ」というメールアドレスを作成する機能があり，この機能を用いることで，前述のメールエイリアス機能と同等な転送アドレスを作成できる．

利用者アカウントに複数のメールアドレスを登録できるが，それらのメールアドレス宛のメールは，利用者アカウントごとに準備される単一のメールボックスに配送される．さらに，送信メールのヘッダの From フィールド [12] に利用できるメールアドレスには制限がある．基本的に，利用者アカウントに「プライマリ SMTP アドレス」として登録された 1 個のメールアドレスしか，From フィールドに利用できない．利用者アカウントに登録したプライマリ SMTP アドレス以外のメールアドレスは「セカンダリ SMTP アドレス」として扱われ，セカンダリ SMTP アドレス宛のメールは受信できるが，送信メールの From フィールドには利用できない．この制限は，単一アカウントで複数アドレスを使い分けたい場合に問題となる．ただし，利用者アカウントに対して，特定配布グループについての「SendAs 権限」を付与することで，配布グループのメールアドレスを From フィールドに利用可能である．

利用者アカウントの管理および認証には，Identity as a Service（IDaaS）[13] である Azure Acitive Directory [14] が用いられる．利用者アカウントはテナント全体の管理権限を持つサービス管理者だけが作成，削除できる．管理権限の対象を一部のアカウントやドメインなどに制限することはできず，サービス管理者はすべての登録ドメインについて管理できる．配布グループの作成権限を持つアカウントは制限できるが，テナントに登録された任意のドメインに対して配布グループを作成できる．なお，個別の配布グループに所有者として登録されたアカウントは，所有する配布グループに対してメンバの登録，削除が可能である．

Exchange Online の管理は，ウェブブラウザから操作するウェブユーザインタフェースが提供されるほかに，PowerShell [15] を用いたリモート管理インタフェース [16] が提供されている．なお，Microsoft Graph API [17] と呼ばれる RESTful [18] ウェブ API では Exchange Online を管理するための API は提供されていない．

2.3 非機能指標

本研究目的の実現に必要となるシステムの設計において重要な非機能指標について以下に述べる．本研究の主目的は管理負荷軽減のための権限委譲にあり，また，運用管理の負担軽減という SaaS の利点を低減させないことは重要であるので，システムの運用や保守などに掛かる管理負担は最も重要な指標である．また，SaaS の仕様は利用者に断りなく変更される場合があり，それにともないシステムの機能に変更や追加が必要になる可能性があることから，システムの保全性や拡張性も重要である．さらに，前述のとおりメールサービスの運用においては，アカウント情報の機密性やシステムのセキュリティの保持は重要である．

3. 提案手法

3.1 権限委譲方法

2.1 節であげた必要な機能を SaaS メールサービスにおいて実現するために，以下に述べる方法を提案する．まず，権限委譲対象の組織ドメインは，機関の SaaS メールサービスのテナントに登録し，SaaS メールサービス上で利用可能にする．また，組織ドメイン宛のメールを SaaS メールサービスに配送するように，組織ドメインの MX レコードを設定する．

使用する組織アドレスそれぞれに対して，2.2 節で説明した配布グループを作成し，組織アドレスの転送先を配布グループのメンバとして登録する．ここで，組織アドレスが単一利用者のアドレスである場合は，その利用者に対応する機関アカウントのメールボックスへとメールが配送されるようにメンバ登録する．また，機関アカウントである配布グループのメンバには，配布グループに対する SendAs 権限を付与することで，組織アドレスを From フィールドに指定したメールを機関アカウントにより送信可能とする．

組織ドメイン管理者には，管理対象の組織ドメインの組織アドレスについてのみ，管理権限を委譲する必要があるが，これは以下の方法で実現する．まず，SaaS メールサービスのアカウントを管理する IDaaS 上に，組織ドメインごとに組織ドメイン管理者用のアカウントグループを作成し，この管理者グループに組織ドメイン管理者の機関アカウン

トを登録する．このとき，管理者グループは，対象とする組織ドメインから一意に定まるようにする．たとえば，組織ドメインと相互変換可能な名称を付けるなどの方法がある．そのうえで，組織アドレスに対応する配布グループの所有者として，その組織ドメインの管理者グループを指定する．機関アカウントに対してある組織ドメインの管理権限が委譲されているかは，その組織ドメインに対応する管理者グループに機関アカウントが所属しているかで確認できる．組織ドメイン管理者に権限委譲されている組織アドレスの一覧は，管理者グループを介して機関アカウントが所有者となっている配布グループの一覧により得られる．

そのうえで，組織ドメイン管理者が組織アドレスに対応する配布グループを管理するための，SaaS メールサービスを補完する組織アドレス管理システムを提供する．このようにして，機関アカウントと組織アドレスとの管理を分離することにより，機関アカウントは機関のサービス管理者が一元管理しつつ，組織アドレスを組織ドメイン管理者に委譲可能とする．

3.2 権限委譲手順

本提案手法により組織ドメインの組織アドレス管理権限を委譲する際に必要な手順を以下にまとめる．

(1) 機関全体のサービス管理者が，IDaaS において，委譲対象の組織ドメインに対応する管理者グループを，組織ドメイン管理者の機関アカウントをメンバとして作成する．
(2) 機関全体のサービス管理者が，SaaS メールサービスに対して，対象の組織ドメインを登録する．
(3) 組織ドメイン管理者が，DNS における組織ドメインの MX レコードに，SaaS メールサーバが提供する MX ホスト名を登録する．

4. 提案システム設計

4.1 全体設計

3 章で述べた提案手法の実現に必要となる，組織ドメイン管理者へ提供する組織アドレス管理システムについて，2.3 節で述べた非機能指標を考慮した設計を以下に提案する（**図 2**）．システムは，ウェブアプリケーションとして設計する．保全性や拡張性を考慮し，このウェブアプリケーションにはマイクロサービスアーキテクチャ [19] を採用する．各サービスはクラウドサービスの Function as a Service（FaaS）[20], [21]，Platform as a Service（PaaS）[1]，IDaaS を用いて実現し，計算機や仮想機械，基盤ソフトウェアなどの構成，保守や管理を要しないサーバレス [22] 構成とする．さらに，システムのユーザインタフェースは，処理をクライアント側で実行するシンサーバ構成とする．

本システムは主に以下の要素で構成される．

組織アドレス管理ウェブサービス RESTful ウェブサービスとして，SaaS メールサービス上の組織アドレスの管理機能を，REST API により提供する．

管理者用ウェブアプリケーション クライアント側のウェブブラウザ上で実行され，組織ドメイン管理者に管理用ユーザインタフェースを提供する．

静的ウェブサイト PaaS ストレージ上でホスティングし，前記ウェブアプリケーションを静的コンテンツとして配信する．

4.2 組織アドレス管理ウェブサービス

組織アドレス管理ウェブサービスは，FaaS 上で処理を行う RESTFul ウェブサービスである．組織ドメイン管理者からの処理要求に応じて，サービス管理者権限で SaaS メールサービスへの設定処理を実行する．

HTTP [23] を用いた REST API として，以下の機能を提供する（**表 1**）．

一覧取得 組織ドメイン管理者が管理権限を委譲されている組織ドメインについて，組織アドレスの一覧を応答

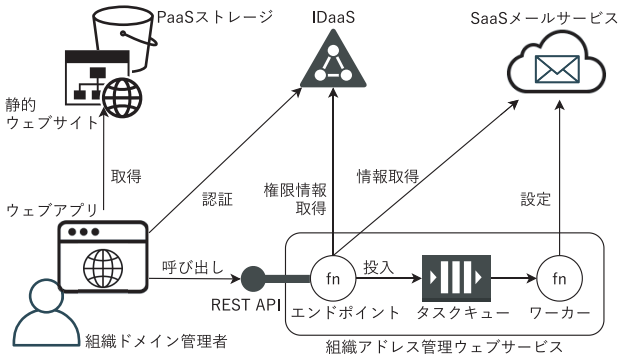

図 2 提案システム概略図
Fig. 2 System framework diagram.

表 1 組織アドレス管理 REST API
Table 1 REST API for management of organizational addresses.

機能	HTTP メソッド	パラメタ	応答
一覧取得	GET	–	組織アドレス一覧
作成	POST	組織アドレス，転送先一覧	処理受付成否
転送先一覧取得	GET	組織アドレス	転送先一覧
転送先更新	POST	組織アドレス，転送先一覧	処理受付成否
削除	POST	組織アドレス	処理受付成否

として返す．

作成 組織アドレスと転送先一覧を指定して組織アドレスを作成する．

転送先一覧取得 指定された組織アドレスの転送先一覧を応答として返す．

転送先更新 指定された組織アドレスの転送先を，指定された転送先一覧で上書き更新する．

削除 指定された組織アドレスを削除する．

今回，組織アドレスとその転送先は，配布グループとそのメンバに対応し，Exchange Online に対してリモート PowerShell を用いて，配布グループの作成，読み取り，変更，削除の処理を実行する．

HTTP の GET メソッドを用いる取得系エンドポイント（表 1）では，実時間処理で SaaS メールサービスから必要な情報を取得し，GET メソッドの応答として結果を返す．一方，その他のエンドポイントでは，後述の方法で処理権限を確認した後，非同期メッセージキューであるタスクキューに処理要求を投入し，即座に応答を返す．実際の処理は，ワーカースレッドによるバックグラウンド処理として，SaaS メールサービスの設定処理を実行する．これは，SaaS メールサービスにおける作成処理や設定変更処理などは時間が掛かる場合があること，また，Exchange Online のリモート PowerShell 接続では最大接続数が数個に制限されており，同時に多数の REST API 呼び出しがあっても並行処理ができないことなどが理由である．一方，取得系の機能については，呼び出し元で処理を継続する必要があることから，実時間で結果を返すこととする．ただし，呼び出し元では処理がタイムアウトとなった際の再試行を考慮する必要がある．今回，エンドポイント処理およびワーカースレッドには FaaS として Azure Functions [24] を利用し，タスクキューには PaaS である Azure Queue Storage [25] を用いる．バックグラウンド処理の結果は，組織ドメイン管理者宛に電子メールで送信することとする．

4.3 管理者用アプリケーション

組織ドメイン管理者には，組織アドレスを管理するためのユーザインタフェースをウェブアプリケーションとして提供する．ウェブアプリケーションは静的ウェブコンテンツとして PaaS ストレージ上に配置してホスティングし，ウェブサーバ側では HTTP 応答以外の処理は行わない．クライアント側の組織ドメイン管理者が使うウェブブラウザ上で，静的ウェブコンテンツを取得し，前節で述べた REST API の呼び出し，応答の処理，入出力などの処理を実行する．今回，PaaS ストレージである Azure Blob Storage [26] でホスティングし，ウェブアプリケーションの処理系として HTML5 [27] と ECMAScript 2015（ES6）[28] を用いる．

4.4 認証と認可

本システムの認証および認可には，SaaS メールサービスのアカウントを管理する IDaaS を用いる．認証と認可に関する情報は IDaaS のみに登録し，システムではいっさいのアカウント情報を独自に保持せず，認証と認可は IDaaS を介して処理する．今回は，Exchange Online が利用する Azure Active Directory を IDaaS として用いる．

4.2 節で述べた組織アドレス管理ウェブサービスの REST API では，機関アカウントに権限委譲された組織ドメインに対する処理のみを受理するが，これには OAuth 2.0 [29] を用いて以下の方法で行う．OAuth 2.0 のクライアントに該当する管理者用ウェブアプリケーションは，認可サーバである IDaaS へとリダイレクトして組織ドメイン管理者の認証を行い，IDaaS から認証コード，次いで，アクセストークンを取得する．その後，管理者用ウェブアプリケーションが，リソースサーバに該当する組織アドレス管理ウェブサービスに対して処理要求する際には，取得したアクセストークンを附して REST API エンドポイントにアクセスする．組織アドレス管理ウェブサービスは，渡されたアクセストークンにより組織ドメイン管理者を認証する．また，アクセストークンに含まれる機関アカウントの識別名を用いて，要求された組織ドメインに対する処理権限を上述の方法により確認し，権限のある処理要求のみを受理して実行する．

5. 考察

本稿で述べた手法およびシステムにより，特定ドメインのメールアドレスに限定した管理権限をアカウントに与える機能を持たない SaaS メールサービスにおいて，組織ドメインごとに組織アドレスの管理権限を委譲することが実現できる．これにより，メールアドレス管理の負荷集中を回避し，サービス管理者による SaaS メールサービスの運用管理負担を軽減できる．また，本手法では，権限委譲の管理には IDaaS 上の管理者グループとそのメンバ管理が必要とされるだけであり，権限委譲に必要な作業は 3.2 節に示したとおり軽微である．それゆえ，SaaS メールサービス運用管理の総合的な負担はほとんど増加しない．なお，管理者グループのメンバ管理を組織ドメイン管理者自身に委ねる運用も可能である．さらに，組織ドメイン管理者は組織アドレスの転送先を管理する作業だけが求められ，伝統的構成による権限委譲と比べて，組織ごとのメールサービスの運用管理，アカウント管理やセキュリティ保守が不要であることから，運用管理負担は大幅に削減される．

なお，単に組織ドメインについて SaaS メールサービスを利用することが目的であれば，組織ドメイン別に SaaS メールサービスのテナントを用意したうえで機関内で共通のアカウント管理基盤から個別テナントへとアカウントを連携する構成も考えられるが，機関内で複数のテナントを

管理する必要があることから，1 章で問題としてあげた，アカウント漏洩リスクや，単一利用者による複数アカウント管理の必要性を解決できず，この方法では本稿の目的は達成できない．また，単に組織アドレスを機関全体のテナントで扱うことが目的であれば，組織アドレスごとに別のアカウントを作成する方法も考えられるが，やはり単一利用者による複数アカウント管理の必要性を解決できず，本稿の目的は達成できない．

本システムは全体をサーバレス構成として設計しており，サーバレス構成は，オンプレミスや Infrastracture as a Service（IaaS）[1] 上でシステムを構築する方法と比較して，ハードウェアや仮想マシン，基盤ソフトウェアの運用管理を必要としないことから，運用管理負担は大幅に削減される [30] とともに，必要な計算資源がサービスプロバイダにより自動的に提供されることから，計算資源の管理も必要ない [30], [31]．また，FaaS は，コンテナなどの OS レベル仮想化を用いて構築する方法と比較しても，ソフトウェア実行環境を管理する必要がないことから構成や運用の負担を軽減する [32]．さらに，管理者用アプリケーションは静的コンテンツとして PaaS 上から配信する構成であるので，ウェブサイトの運用管理負担はほとんど生じない．これらのことから，本システムにより実現される管理負担の軽減に比して，本システムの管理負担は十分に小さい．

本システムでは，アカウント管理および認証認可処理には IDaaS，静的ウェブサイトのホスティングには PaaS ストレージサービスを用い，さらに，組織アドレス管理ウェブサービス内部では，非同期メッセージキューに PaaS キューサービス，処理実行に FaaS を用いているように，各機能を分割して個別のサービスとして実現するマイクロサービスアーキテクチャを用いている．マイクロサービスは適切に構成することで機能の追加や変更のコストを削減し，保全性に寄与する [33]．本システムの設計ではサービス間の依存関係は限定されている．SaaS メールサービスの仕様に変更があった場合も，4.2 節で述べた REST API を変更しない限りは，本システムの影響範囲は組織アドレス管理ウェブサービスの FaaS 部分だけに限定される．本システムでは，REST API として組織アドレス管理ウェブサービスのインタフェースを提供していることから，機関内の組織が独自で運用するシステムと連携することも可能である．さらに，REST API の追加などによりシステム機能を容易に拡張でき，管理者用ユーザインタフェースの機能追加や改修は管理者用ウェブアプリケーションの変更で対応できる．また，マイクロサービスアーキテクチャおよび FaaS では，システムで実装を必要とする部分が限定されることでコード量が削減される [30], [33] とともに，FaaS はコード配備を簡易化し [30]，マイクロサービスアーキテクチャではシステム改修時の段階的移行が可能であることから保全性が向上する [33]．これらのことから，本システムの設計は保全性および拡張性に優れているといえる．

一方，マイクロサービスアーキテクチャを用いた場合，システム全体の可用性は構成要素の可用性の影響を受ける [33]．本システムでは，利用する FaaS，PaaS，IDaaS のいずれか，または，それらの間の接続が利用不可になると，全体が利用できなくなる．ただし，今回採用した構成では，SaaS メールサービスと同じプロバイダが提供するサービスをすべての構成要素で用いることで，サービス間の接続性やサービス全体の可用性に配慮している．

マイクロサービスアーキテクチャや FaaS では，サービス間でネットワークを介して情報が受け渡されることから，一般的に機密性やセキュリティが課題とされる [31], [33]．しかし，本システムでは，認証と認可に関する情報を除けば，サービス間で受け渡される情報に機密性の高いものはない．認証と認可には，アカウント情報を IDaaS のみに格納し，現時点で広く普及している OAuth 2.0 を用いており，標準的なセキュリティは満たしているといえる．ほかに，マイクロサービスアーキテクチャやサーバレス構成では，攻撃対象領域が増大するという課題もある [21], [33]．しかし，本システムにおけるウェブサーバは，静的コンテンツを配信するだけの機能を PaaS ストレージを用いて実現しており，通常のウェブアプリケーションサーバよりもセキュリティリスクは低いと考えられる．それゆえ，本システムにおける実質的な攻撃対象領域は組織アドレス管理ウェブサービスだけであり，マイクロサービスアーキテクチャによる攻撃対象領域の増大は生じていない．また，組織アドレス管理ウェブサービスに対しては，REST API だけが外部から通常の手段でアクセス可能であり，各アカウントにより可能な処理は管理対象ドメインのアドレス管理に限定されることから，組織ドメイン別にメールサービスを運用する場合などと比べれば，大幅にセキュリティリスクは低減しているといえる．ただし，組織アドレス管理ウェブサービスは，SaaS メールサービスの管理者権限を持つアカウントを利用することからセキュリティ侵害の影響は大きいが，SaaS 側でアカウントへの適切なアクセス制限を施すことでアカウント侵害のリスクを軽減できる．これらのことから，本システムの設計は，求められる機密性とセキュリティを満たしていると考えられる．

本稿では，SaaS メールサービスとして Exchange Online を対象としたが，本手法およびシステム設計は，メールエイリアス機能と同等の機能，および，各アカウントで送信に利用可能なメールアドレスを追加する機能を備え，ネットワーク経由でプログラムから呼び出し可能な管理インタフェースを備える，G Suite の Gmail [34] などの他の SaaS メールサービスにも適用可能である．また，メールサービス以外にも一部対象に限定した管理権限の付与ができない SaaS は多いが，それらの SaaS に対して権限委譲を補完する用途にも，本稿で述べた手法や設計の基本的な考え方は

適用できると考えられる．

6. おわりに

本稿では，一部メールアドレスに限って管理権限を付与できない集中管理方式の SaaS メールサービスにおいて，組織ドメインごとの管理者に組織アドレスの管理権限を委譲する手法，および，その手法を実現するために SaaS メールサービスを補完する組織アドレス管理システムについて述べた．本手法および本システムでは，組織ドメインの管理者が作成，変更，削除できる組織アドレスについて，組織に所属する構成員が，一元管理された各自の機関アカウントを用いてメールを送受信でき，また，機関内外のメールアドレスへと転送できる．それを実現する組織アドレス管理システムは，マイクロサービスアーキテクチャに基づくサーバレス構成のウェブアプリケーションとして設計したことにより，システム管理負担が低く，かつ，保全性と拡張性に優れており，IDaaS と標準的な認証認可手法の利用により必要な機密性とセキュリティを満たしている．

本稿で述べた手法と設計に基づき，九州大学情報統括本部においてシステムを構築し，2020 年 4 月より運用を開始している [35]．今後は，構築したシステムを活用することで，九州大学内の各組織で独自に運用されているメールサーバについて，情報統括本部が管理する全学 SaaS メールサービスへの集約を進めていく計画である．また，現時点で構築したシステムでは，管理者用アプリケーションのユーザインタフェース機能が非常に限られているので，今後，クライアント側実装の改修により機能拡張を行っていく予定である．組織アドレス管理ウェブサービスの REST API については，必要があれば，学内組織が運用するシステムに公開することも考えられる．

謝辞 機能要件およびシステム設計の検討において，九州大学情報統括本部メールサーバ集約タスクフォースのメンバに意見をいただいた．ここに感謝に意を表する．

本研究は JSPS 科研費 JP20K11791 の助成を受けた．

参考文献

[1] Mell, P. and Grance, T.: The NIST Definition of Cloud Computing, Special publication 800-145, National Institute of Standards and Technology (2011).
[2] Badger, M.L., Grance, T., Patt-Corner, R. and Voas, J.M.: Cloud Computing Synopsis and Recommendations, Special publication 800-146, National Institute of Standards and Technology (2012).
[3] 河合輝欣，児西清義，米村征洋：ASP・SaaS の動向と普及促進の状況（前編），情報処理，Vol.49, No.11, pp.1325–1333 (2008).
[4] Verizon: 2020 Data Breach Investigations Report, Technical report, Verizon Communications Inc. (2020).
[5] Google Cloud: Google Cloud Security and Compliance Whitepaper, Google, LLC (online), available from ⟨https://storage.googleapis.com/gfw-touched-accounts-pdfs/google-cloud-security-and-compliance-whitepaper.pdf⟩ (accessed 2020-06-22).
[6] Microsoft: Protect against threats, Microsoft Corporation (online), available from ⟨https://docs.microsoft.com/en-us/microsoft-365/security/office-365-security/protect-against-threats⟩ (accessed 2020-06-22).
[7] Mockapetris, P.V.: Domain names - concepts and facilities, STD 13, Internet Engineering Task Force (2008).
[8] Klensin, J.: Simple Mail Transfer Protocol, RFC 5321, Internet Engineering Task Force (2008).
[9] Microsoft: Exchange Online, Microsoft Corporation (online), available from ⟨https://docs.microsoft.com/en-us/exchange/exchange-online⟩ (accessed 2020-06-22).
[10] Mockapetris, P.: Domain names - implementation and specification, RFC 1035, Internet Engineering Task Force (1987).
[11] Malkin, G.: Internet Users' Glossary, RFC 1983, Internet Engineering Task Force (1996).
[12] Resnick, P.: Internet Message Format, RFC 5322, Internet Engineering Task Force (2008).
[13] Habiba, U., Masood, R., Shibli, M.A. and Niazi, M.A.: Cloud identity management security issues & solutions: A taxonomy, *Complex Adaptive Systems Modeling*, Vol.2, No.1, p.5 (2014).
[14] Microsoft: What is Azure Active Directory?, Microsoft Corporation (online), available from ⟨https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-whatis⟩ (accessed 2020-06-22).
[15] Microsoft: PowerShell Documentation, Microsoft Corporation (online), available from ⟨https://docs.microsoft.com/en-us/powershell/⟩ (accessed 2020-06-22).
[16] Microsoft: Exchange Online PowerShell, Microsoft Corporation (online), available from ⟨https://docs.microsoft.com/en-us/powershell/exchange/exchange-online/exchange-online-powershell⟩ (accessed 2020-06-22).
[17] Microsoft: Use the Microsoft Graph API, Microsoft Corporation (online), available from ⟨https://docs.microsoft.com/en-us/graph/use-the-api⟩ (accessed 2020-06-22).
[18] Fielding, R.T.: Architectural styles and the design of network-based software architectures, PhD Thesis, University of California, Irvine (2000).
[19] Lewis, J. and Fowler, M.: Microservices, *Martin-Fowler.com* (2014) (online), available from ⟨https://martinfowler.com/articles/microservices.html⟩.
[20] Fox, G.C., Ishakian, V., Muthusamy, V. and Slominski, A.: Status of Serverless Computing and Function-as-a-Service (FaaS) in Industry and Research, arXiv, No.1708.08028 (online), DOI: 10.13140/RG.2.2.15007.87206 (2017).
[21] Roberts, M.: Serverless Architectures, *Martin-Fowler.com* (2018) (online), available from ⟨https://martinfowler.com/articles/serverless.html⟩.
[22] Roberts, M. and Chapin, J.: Differentiating Serverless, *What Is Serverless?*, O'Reilly Media, Incorporated, chapter 5 (2017).
[23] Fielding, R. and Reschke, J.: Hypertext transfer protocol (HTTP/1.1): Message syntax and routing, RFC 7230, Internet Engineering Task Force (2014).
[24] Microsoft: An introduction to Azure Functions, Microsoft Corporation (online), available from ⟨https://docs.microsoft.com/en-us/azure/azure-functions/functions-overview⟩ (accessed 2020-06-22).
[25] Microsoft: What are Azure queues?, Microsoft Corpora-

tion (online), available from ⟨https://docs.microsoft.com/en-us/azure/storage/queues/storage-queues-introduction⟩ (accessed 2020-06-22).
[26] Microsoft: What is Azure Blob storage?, Microsoft Corporation (online), available from ⟨https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-overview⟩ (accessed 2020-06-22).
[27] WHATWG: HTML Living Standard, WHATWG (Apple, Google, Mozilla, Microsoft) (online), available from ⟨https://html.spec.whatwg.org⟩ (accessed 2020-06-22).
[28] Ecma International: ECMAScript 2015 Language Specification, Standard ECMA-262 6th Edition, Ecma International (2015).
[29] Hardt, D. et al.: The OAuth 2.0 Authorization Framework, RFC 6749, Internet Engineering Task Force (2012).
[30] Roberts, M. and Chapin, J.: Benefits of Serverless, *What Is Serverless?*, O'Reilly Media, Incorporated, chapter 1 (2017).
[31] Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., Mitchell, N., Muthusamy, V., Rabbah, R., Slominski, A. and Suter, P.: Serverless Computing: Current Trends and Open Problems, *Research Advances in Cloud Computing*, Chaudhary, S., Somani, G. and Buyya, R. (Eds.), Springer Singapore, Singapore, pp.1–20 (online), DOI: 10.1007/978-981-10-5026-8_1 (2017).
[32] Lynn, T., Rosati, P., Lejeune, A. and Emeakaroha, V.: A Preliminary Review of Enterprise Serverless Cloud Computing (Function-as-a-Service) Platforms, *2017 IEEE International Conference on Cloud Computing Technology and Science* (*CloudCom*), pp.162–169 (2017).
[33] Dragoni, N., Giallorenzo, S., Lafuente, A.L., Mazzara, M., Montesi, F., Mustafin, R. and Safina, L.: Microservices: Yesterday, Today, and Tomorrow, *Present and Ulterior Software Engineering*, Mazzara, M. and Meyer, B. (Eds.), Springer International Publishing, Cham, pp.195–216 (online), DOI: 10.1007/978-3-319-67425-4_12 (2017).
[34] Google: Gmail - G Suite Admin Help, Google, LLC (online), available from ⟨https://support.google.com/a/topic/9202⟩ (accessed 2020-09-21).
[35] 嶋吉隆夫，笠原義晃，清家史郎，藤村直美：九州大学における独自運用メールサービス集約のためのシステム開発，情報処理学会研究報告インターネットと運用技術（IOT），Vol.2020-IOT-50, No.9, pp.1–8 (2020).

嶋吉 隆夫 （正会員）

1999 年京都大学大学院工学研究科修士課程修了．同年三菱電機（株）．2004 年（財）京都高度技術研究所．2008 年京都大学大学院情報学研究科博士後期課程修了．博士（情報学）．2013 年同研究科助教．2015 年九州大学情報基盤研究開発センター准教授．情報システム運用，生理学シミュレーションの研究に従事．電子情報通信学会，生体医工学会，ACM，IEEE-EMBS 各会員．

笠原 義晃 （正会員）

1993 年九州大学大学院工学研究科修士課程修了．1996 年同博士後期課程修了．博士（工学）．同年同大学大型計算機センター助手．2000 年同大学情報基盤センター助手．2007 年同大学情報基盤研究開発センター助教．情報システム，情報セキュリティに関する研究に従事．電子情報通信学会，ACM 各会員．

清家 史郎

2009 年琉球大学工学部電気電子工学科卒業．同年コスミックビジネス（株）．2011 年（株）ミックスネットワーク．2016 年（株）Fusic．Web アプリケーションの開発に従事．

藤村 直美 （正会員）

1978 年九州大学大学院工学研究科博士課程単位取得退学．1978 年同大学助手（工学部）．1980 年同助教授．1988 年九州芸術工科大学助教授．1995 年同大教授．2003 年九州大学教授（芸術工学部）．2016 年同大学名誉教授，特任教授．2020 年同大学情報基盤研究開発センター訪問研究員．工学博士．ICT を活用した教育，共同利用センターの管理運営に関する研究に従事．ACM，IEEE 各会員終身会員．ACM SIGUCCS Hall of Fame．本会フェロー．