

eduroam をより導入しやすくするオープンソースツールの開発

リュウ・ジョシュア¹⁾, 下條 真司¹⁾

1) 青森大学 ソフトウェア情報学部

shinji.shimojo@aomori-u.ac.jp

Development of Open-Source Tools to Simplify eduroam Deployment

Joshua Liew¹⁾, Shinji Shimojo¹⁾

1) Faculty of Software and Information Technology, Aomori University

概要

青森大学は2024年1月から eduroam に参加しているが、導入において技術的な困難を経験し、現在も運用管理や学内の展開が課題である。本稿では、地方の小規模私立大学における、eduroam をはじめとする大学 ICT システム導入の困難を示すとともに、それを手助けする思考法や本学の取り組みについて共有する。具体的に、eduroam の SP と IdP 両方としての導入において、ワンストップで簡易に構築し、デプロイすることを可能にするオープンソースツールの開発について述べる。

1 はじめに

青森大学（以降、「本学」という）は、学生数約 1200 名、教員数約 100 名の地方の小規模私立大学である。また、青森・むつ・東京の3拠点にキャンパスを置き、4 学部で文理融合の教育を行っている。2024 年 1 月から eduroam（エデュローム）に参加しており、IdP（アカウントの認証等を提供する機関）と SP（ネットワーク接続サービスを提供する機関）の両方として機能している。

しかし、学内で eduroam による安全な無線 LAN を利用できるエリアがまだ限られており、展開が課題となっている。お金と時間の制約を受けてオープンソースソフトウェア（以降、「OSS」という）で必要なサーバを構築し、eduroam のコア機能（認証やネットワーク接続サービス）を実現しているが、導入および運用管理を簡易化するような OSS がほとんど存在しない。したがって、アカウントの削除や新規作成、インシデント対応などに手間がかかり、煩雑である。そのため、情報化推進センター（大学の ICT 環境の整備や運用管理を担当する部門）には現在 3 人しかいないことを加えて、労力的にも技術的にも学内やキャンパス間の展開が難しい現状にある [1]。

本学のように eduroam の導入や展開に困難を覚えている機関も少なくないと思われる。執筆時点（2025 年 9 月）では、eduroam が世界中に普及しており、国内約 460 機関で利用されている¹⁾。国内

の参加機関の所在地を地区単位ⁱⁱ⁾で単純に集計し比較したところ、近畿と関東（北関東と南関東）がそのうちの約 55% を占めていることが確認できたⁱⁱⁱ⁾。この地域差を説明する原因を見出すにはより精度の高い分析が必須であるが、地方における普及が特に難しいと示唆されている。小規模大学のニーズに合わせ、お金や時間、技術やマンパワーをあまり必要としない形で eduroam を導入、運用管理する方法が存在しないことが大きな要因だと考えられる。小規模大学における eduroam の普及を手助けする必要があるといえよう。

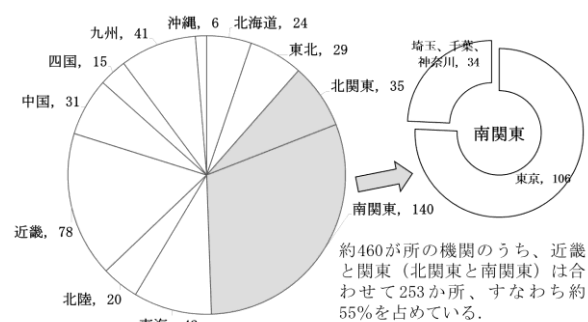


図1 eduroam JP 参加機関の分布

本稿では、eduroam を導入しやすくするツールの開発を構想する。初めに、eduroam に参加する過程を俯瞰して考えることで、導入や運用管理を容易にする必要性を探る。次に、大学 ICT を推進するようなツールを開発するにあたって重要となる視点を共有する。最後に、提案手法を示した上、実装の取り組みを紹介する。

2 eduroam の導入や運用管理の課題

文部科学省によると、地方の中小私立大学の経営状況は約4割が赤字傾向にあり [2], 内閣官房によると、若手の IT 人材を含め、地方から若手人材の流出が著しいことが分かる [3]. 本学もまた同様であり、技術者や ICT 環境の管理者が現在3人しかいないことで、DX への取り組みが難しい現状がある [1].

2.1 導入する際の制約条件と検討事項

本学における eduroam の導入を検討していた際、それにかかるコストまたは技術力や労力が特に問題となった。

検討事項として特に重要となってきたのが、eduroam IdP としての責任をどこまで負うのか、ということであった。eduroam を導入するのであれば、当然 SP として機能しなければならない。学認（学術認証フェデレーション）による eduroam JP 認証連携 ID サービスを利用したり、SecureW2 による Radius-as-a-Service の「Cloud Radius」を利用したりすることで IdP としての責任と負担を大きく軽減し、あるいは完全に無くすこともできる。

しかし、コストの側面から考えると、学認への加入や外部の IdP 利用が現実的ではないと判断した。そのため、SP と IdP 両方として機能できるよう、低コストかつオンプレミスで稼働するシンプルなアーキテクチャを採用した。OSS で無償であるという理由で RADIUS 認証サーバ・ソフトウェアとして FreeRADIUS、比較的導入しやすいという点で認証方式としてパスワードベースである PEAP-MSCHAPv2、認証情報を保管するデータベースとして馴染みのある SQL データベースを採用し、ミニ PC 上に構築することにした。

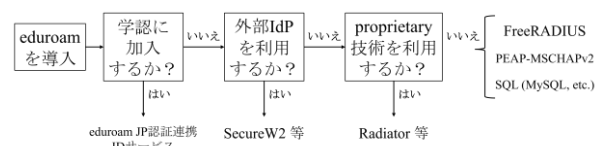


図2 eduroam 導入の検討事項

2.2 導入や運用管理のワークフロー

著者らは、2023年6月ごろから eduroam の導入に向けてサーバを構築したり、ネットワークの見直しを行ったりしていた。上記のアーキテクチャや技術によって各作業の流れが定まっていた。

以下にそのワークフローを簡略的にまとめる。まず、Linux マシンを用意した上で FreeRADIUS

による RADIUS 認証サーバを構築する。次に、PEAP-MSCHAPv2 でパスワードベースの認証が行えるように SQL データベースを合わせて構築する。実は eduroam のコア機能（アカウント認証とネットワーク接続サービス）をこれだけで実現できる。但し、証明書ベースの認証方式である EAP-TLS を使用するために PKI（公開鍵基盤）の構築が求められる。そして、PKI と FreeRADIUS の間に連携が必要となる。

次に、eduroam コンプライアンス・ステートメントを満たすには、FreeRADIUS の標準ログ・モジュールだけでは不十分なため、ログ・モジュールを新たに作成する必要がある。また、アカウント管理やインシデント対応がより簡単にできるよう、管理者向けのインターフェースなどを一般的に導入する。ユーザ（学生や教員）が自身の認証情報が確認できるよう、ユーザ向けのインターフェースも合わせて導入することが多い。

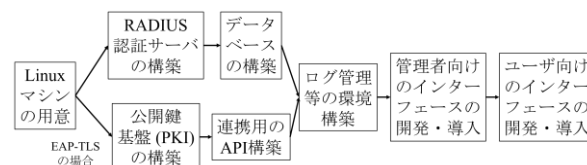


図3 eduroam 導入のサーバ構築ワークフロー

このようにサーバの構築ができるが、eduroam に参加するためにセカンダリ・サーバの構築も求められる。よって、上記の一連の作業を繰り返し行い、プライマリとセカンダリのリソースを複製させたり同期させたりしなければならない。

同様に、技術やアーキテクチャが同じような場合、運用管理にあたっての、アカウント管理やログ管理、インシデント対応といった業務は機関を問わず似たような流れで行われるであろう。多くの機関が検討時に同じような技術的な決定を行うとすれば、同じようなワークフローで導入および運用管理を試みると思われる。

すなわち、導入や運用管理の技術的なワークフローの簡略化や効率化が必要である。

2.3 導入や運用管理の課題

前述のように、eduroam の導入を簡易化した上で運用管理を容易化する必要がある。

しかし、導入や運用管理のプロセスを自動化するようなツールがほとんど存在しないことが現状の課題である。また、使い勝手が良く低コストでオープンに提供されるものはない。そもそも eduroam の導入や運用管理に特化された OSS のエ

コシステムが乏しい。get-eduroam や eduroam CAT は確かにオープンに提供されているものの、こういったツールは eduroam を利用する際の設定を容易化するものである。

上記の課題を解消すれば、地方の中小規模大学における eduroam の導入や運用管理がより容易にできるようになる。このように参加するハードルを下げることによって、地方の中小規模大学からはじまり、全国的な普及が促進されると考えられる。これを核心的な課題として捉え、導入におけるプロセスの自動化のツールを開発し、eduroam をより導入しやすくすることに取り組んでいく。

3 ツールの開発方針について

本章では、前節で述べた課題解決に向けたツール（以降、「本ツール」という）の開発の方向性について説明する。端的に、従来のアーキテクチャに準じて、デベロッパー・エクスペリエンスを意識し、エンド・ツー・エンドの解決策を提供することを開発方針とする。

3.1 eduroam アーキテクチャ

eduroam は、RFC7593 によってアーキテクチャを規定されている [4]。セクション 1.3 では、従来の eduroam アーキテクチャを設計する上で重要視する指針が述べられている。本ツールの開発において、従来のアーキテクチャに準じて、次の三つを指針として取り入れる。

1. Scalable
本ツールで提供するインフラは、多くのユーザに効率良く対応することを目標とする。
2. Easy to install and use
本ツール自体と、本ツールで提供するインフラは、導入しやすく、且つ利用しやすいものでなければならぬ。特に、必要な初期設定を少なくすることを目標とする。
3. Secure
本ツール自体と、本ツールで提供するインフラがセキュアであることを目標とする。

3.2 デベロッパーエクスペリエンス (DevEx)

“Developer Experience”（以降，“DevEx”と略す）は、主にソフトウェア開発における概念として、近年アカデミックな文脈においても取り上げられつつある。

Fagerholm らは、DevEx には “cognition (認知)”, “affect (感情・情緒)”, “conation (意志・意欲)” の三つの要素が含まれている、と主張した [5]。

DevEx とは、ソフトウェア開発において、開発環境に対して、知的価値があると思われるか、満足度が高いと思われるか、目標達成への貢献度が高いと思われるか、を総じて表現する用語であるとされる。それをもとに、本稿でいう DevEx とは、eduroam の導入において、本ツールは、技術者や管理者にとって、技術的な好奇心を刺激するものであるか、使いたくなると感じさせるものであるか、導入に実際に役に立つと思わせるものであるか、を総じて表現する言葉である。

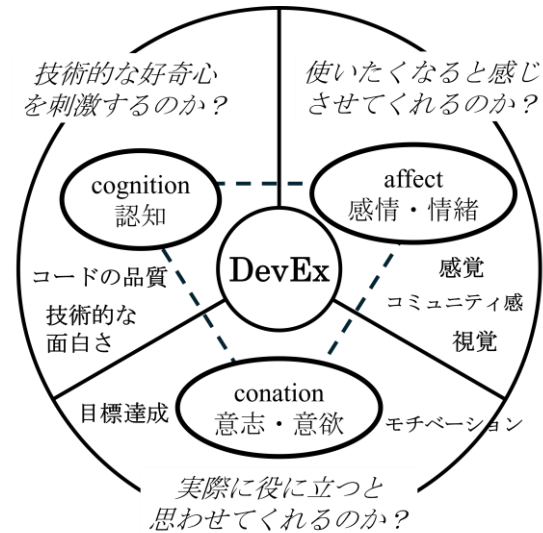


図4 eduroam 導入のツールに関する DevEx

よく似た概念の“Developer Productivity”, つまり「開発者・エンジニアの生産性」に関連する研究が多く行われてきた。例えば、Google の研究によると、ツールやインフラが “not innovative (革新性・創造性がない)” と思われた場合、それを用いたエンジニアの生産性が低くなるようである [6]。eduroam の普及を考える際にもこれらを意識すべきである。導入および運用管理という目標の達成に貢献する質の高いツールが存在すれば、技術者や管理者の間に eduroam への意識が高まり、最終的に普及につながっていくであろう。

以上の観点より、本ツールの開発において、DevEx を特に重要視するものとする。

3.3 OSS のエンド・ツー・エンドソリューション

“End-to-End”（以降、「E2E」と略す）とは、「端から端まで」を意味する英単語である。37signals 社 (Basecamp や HEY) による「Writebook」¹⁾は、Web 上に本を簡単に執筆し公開できるようにするツールであり、E2E ソリューションの事例の一つである。本ツールを OSS の E2E ソリューションとして位置付けるのは、eduroam の導入のワークフ

ローの一部だけでなく、OS や環境のセットアップからサーバのデプロイまで、一連の流れをすべて自動化することを目的としているためである。

4 eduroam インフラ構築自動化ツール

上記の開発方針の下、eduroam の導入のワークフローを自動化するツールを開発する。具体的に、eduroam を稼働するインフラ (RADIUS 認証サーバ等) の構築や初期設定を自動化することを主な目的とする。なお、本ツールは、eduroam CAT や get-eduroam のような、ユーザ (学生や教員) 向けのネットワーク利用設定を簡単にするツールではなく、基盤の構築からサービスの提供までの一連の作業を簡単にするものである、ということを改めて強調しておく。

4.1 提案手法

中心的な課題は、現在の導入のワークフローが自動化できていないため、再現性 (すなわち、同じ操作を誰でもいつでも繰り返すことの可能性) が低いことである。自動化のメリットとして、再現性を向上し導入をフラットに促進できること、技術の統一により透明性を向上しバグやエラーの対応を一元的に行えることが挙げられる。

理想は、一つの技術者が一台のマシンで一つのスクリプトあるいはツールで必要なものすべてを構築・デプロイすることである。ただし、なるべく多くのハードウェアに対応できるように、本ツールを扱える OS を制限する必要があると思われる。さらに、自動化に伴って技術が統一されるため、本稿で述べられる技術以外は議論の対象外となる。

以上を踏まえ、提案手法は、Linux をベースに eduroam に必要なインフラの構築を一つのスクリプトで可能にし、必要な初期設定や操作を一つの CLI ツールで可能にすることである。例の CLI ツールは、最初のスクリプトでインストールすることを想定する。いわゆる“One Person Framework”である^{vii}。

4.2 テックスタック (技術スタック) と実装

実装にあたり、次の技術を利用するものとする。

- FreeRADIUS : eduroam の認証やそのプロキシは RADIUS 認証サーバによって実現する。FreeRADIUS は、OSS の RADIUS 認証サーバソフトウェアであり、世界中に最も多く使用される RADIUS 認証サーバである。ソースコードは C 言語で書かれているが、“unlang”というスクリプト言語でサーバの設定を行う。
- HashiCorp Vault : EAP-TLS 認証を行うために公開鍵基盤 (PKI) が必要である。Vault は、PKI-as-a-service であり、証明書管理 (発行・更新・失効 等) を自動化する OSS である。ソースコードが Go 言語で書かれ、HCL で設定が可能で、充実した API を提供している。
- Golang (Go 言語) : スクリプトと CLI ツールの開発において Go 言語を使用する。標準ライブラリでスクリプトを作成し、Bubble Tea フレームワークで CLI ツールを開発する。API の構築にも適していると言われ、必要に応じて API の構築を行うものとする。
- Ruby-on-Rails+Kamal : いずれ何等かの Web アプリケーション (例えば、ドキュメンテーション

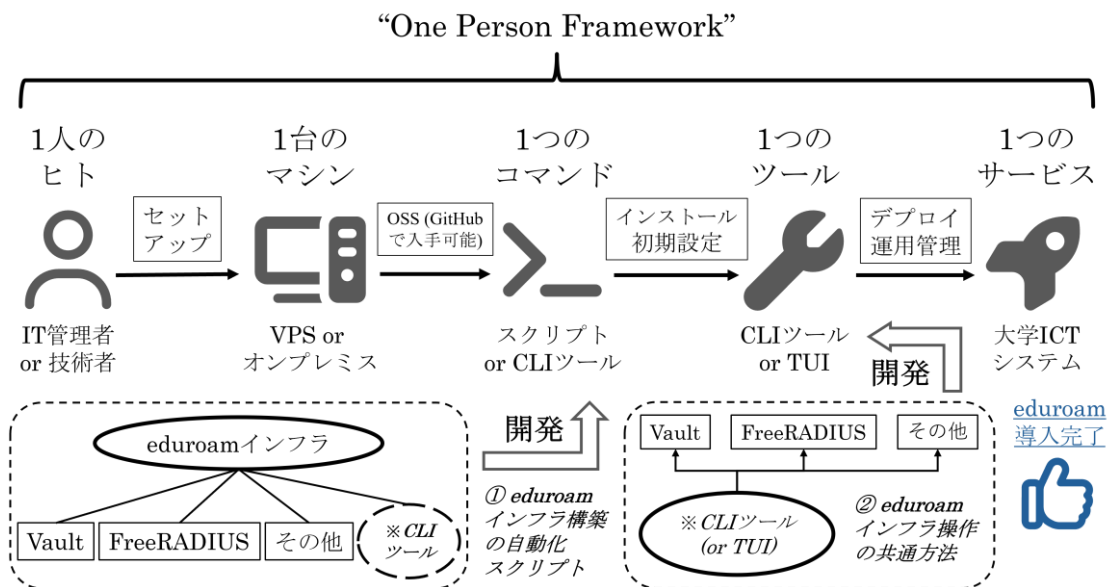


図5 eduroam 導入の One Person Framework

ョンサイト など) を立ち上げることを想定し、その場合に Ruby-on-Rails と Kamal を用いて開発やデプロイを簡素化する。

4.3 発展や評価方法

上記の提案にくわえて、eduroam の運用管理に特化された OSS ツールも必要だと考えられる。また、自機関でホストできるような、eduroam の利用設定を簡単にする、get-eduroam 並びの OSS ツールも必要であろう。これらのニーズを考慮しつつ、他ツールの開発や提供を視野に入れる。

評価方法として、本ツールを用いて、青森県内の他教育機関に eduroam を導入することを目標とする。このように、実用した上で、①本ツールの有用性や操作性、②本ツールで提供するインフラの信頼性や拡張性、③全国的な普及の可能性の三つを確かめるものとする。

5 まとめ

本稿では、eduroam の導入における課題を取り上げ、その課題解決のツールの開発について構想した。この取り組みを通じて、地方の中小規模大学のニーズに合わせた ICT 推進の必要性を掲げつつ、eduroam の全国的な普及の促進に貢献することを目指している。

参考文献

- [1] 工藤 貴裕, 小野 敬明, 角田 均, 下條 真司, “小規模大学でもできる? 青森大学の DX への挑戦!”, 大学ICT推進協議会年次大会論文集, 2023 年度年次大会論文集, p. 151-154, 2023
- [2] 文部科学省, “私立大学の経営状況について”, https://www.mext.go.jp/a_menu/koutou/shinkou/main5_a3_00003.htm (2025 年 9 月 23 日参照)
- [3] 内閣官房, “都道府県別の 15 歳~39 歳 (総数) の転入・転出の状況”, https://www.cas.go.jp/jp/seisaku/chiikihatarakikat/a/contents/20250425/11_sankou03.pdf (2025 年 9 月 23 日参照)
- [4] K. Wierenga, S. Winter, T. Wolniewicz, “The eduroam Architecture for Network Roaming”, <https://datatracker.ietf.org/doc/html/rfc7593> (2025 年 9 月 23 日参照)
- [5] F. Fagerholm, J. Muench, “Developer Experience: Concept and Definition”, IEEE Press, In Proceedings of the International Conference on Software and System Process (ICSSP '12), p. 73-77, 2012
- [6] L. Cheng, E. Murphy-Hill, M. Canning, C. Jaspán, C. Green, A. Knight, N. Zhang, and E. Kammer, “What Improves Developer Productivity at Google? Code Quality”,

Association for Computing Machinery, In Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2022), p. 1302-1313, 2022

- ⁱ “eduroam JP の概要” <https://www.eduroam.jp/about>
“eduroam 参加機関 / Participants”
<https://www.eduroam.jp/participants/siteinfo.html>
- ⁱⁱ 内閣府による地域区分に関するレポートを参考に、本稿では地域区分 C を採用する。
https://www5.cao.go.jp/j-j/cr/cr16/chr16_04.html
- ⁱⁱⁱ 本稿で使用された、参加機関に関するデータや処理方法は次のリポジトリから取得できる。
<https://github.com/joshua-liew/fyp.eduroam-institution-s-count>
- ^{iv} CHAP を規定する RFC1994 の中に “secret” という単語が使われるが、本稿では「パスワード」という。 <https://www.ietf.org/rfc/rfc1994.txt>
- ^v “eduroam Compliance Statement”
https://eduroam.org/wp-content/uploads/2025/07/eduroam_Compliance_Statement_v2-FINAL.pdf
“eduroam コンプライアンス・ステートメント”
<https://eduroam.jp/sites/default/files/2024-03/eduroamコンプライアンス・ステートメントv2.pdf>
- ^{vi} “ONCE — Writebook” by 37signals
<https://once.com/writebook>
- ^{vii} “The One Person Framework”
<https://world.hey.com/dhh/the-one-person-framework-711e6318>