SAKURA internet

透過型SMTPプロキシによる送信メールの可観測性向上: Update Edition

Tomohisa Oda, **Researcher** and **Software Engineer** at **Sakura Internet Research Center**

Presented at **JPAAWG 7th General Meeting** on **Nov 11, 2024**

Tomohisa Oda / @linyows

https://tomohisaoda.com

- 福岡在住 🍜
- さくらインターネット研究所、COGNANOに所属
- Goのローカルコミュニティ Fukuoka.goを主催
- Weight Trainingとテニスが趣味

はじめに

さくらのナレッジに本発表記事があります

- 本発表は、今年初めに開催された「さくらの夕べ in 福岡」での発表をベースにお話しします。
- 「さくらの夕べ in 福岡」での発表は、「さくらのナレッジ」にて文字起こしされて記事になっていますので、現地参加できずにこの資料をご覧になる方は記事を読むとよいでしょう。

  https://knowledge.sakura.ad.jp/37111/

Outline

1. メールシステムの現状
2. メールホスティングの事情と課題
3. 透過SMTPプロキシによる送信メール可観測性向上の提案
4. 既存手法との定量比較
5. 今後の課題

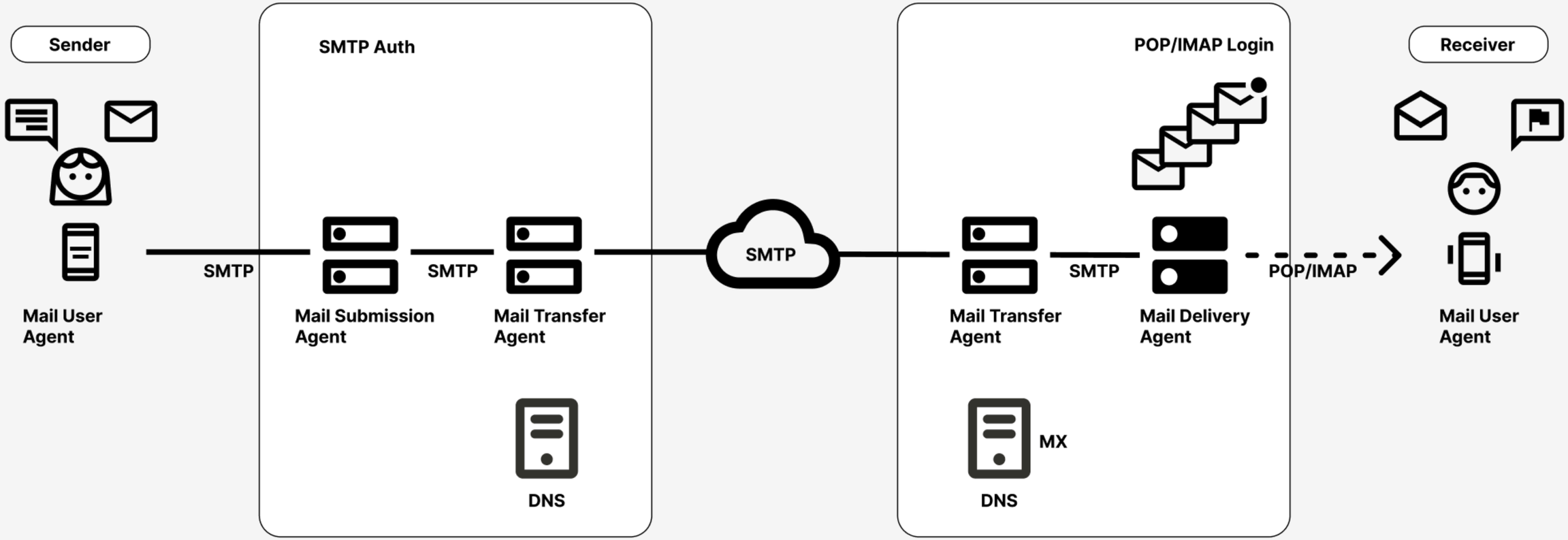
Outline

1. メールシステムの現状
2. メールホスティングの事情と課題
3. 透過SMTPプロキシによる送信メール可観測性向上の提案
4. 既存手法との定量比較（New!!! < IOTS 2024にて研究発表）
5. 今後の課題

メールシステムの現状

メールを送信して相手が受信するまでの全体像

複数のプロトコルと多くのサーバーコンポーネントが必要



メールにおけるセキュリティの問題

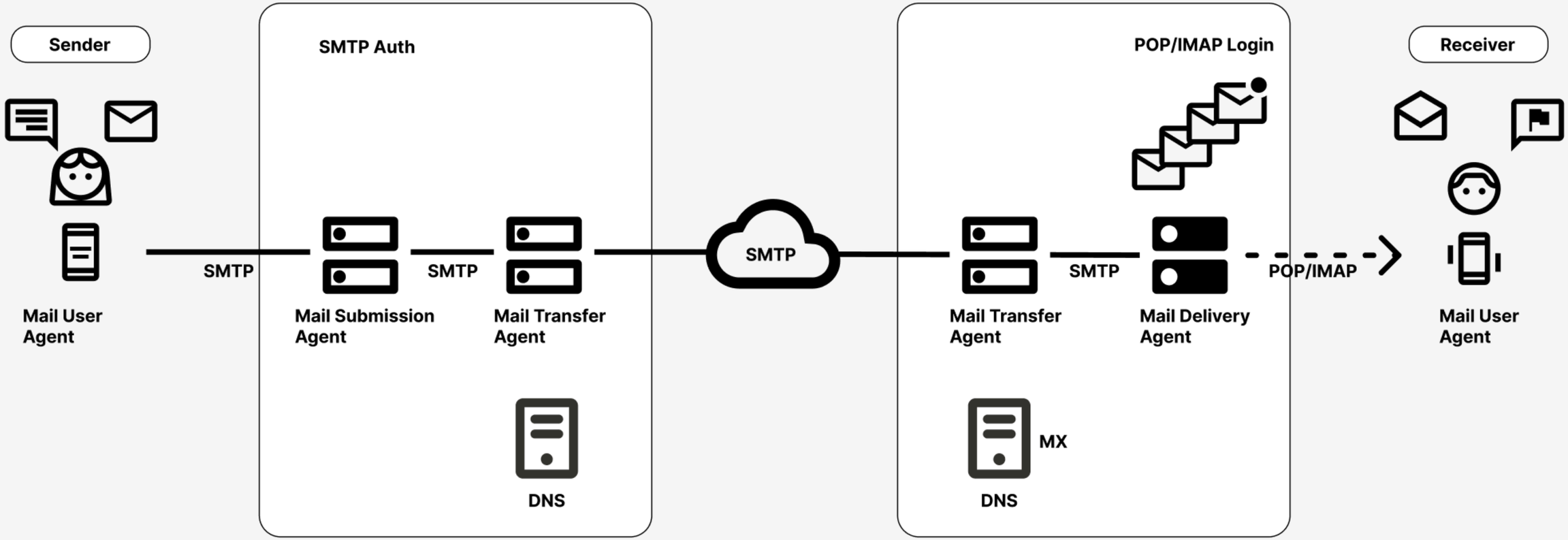
GmailやOutlook使っていれば何も考え(ry

- メールの送信者は正しいか？
- メールに盗聴や改ざんはされていないか？
- 送信者はなりすましをしていないか？
- 自分のフリをしてメールを送られていないか？

- 送信ドメイン認証（SPF、DKIM、DMARC）
- 経路暗号化（TLS、MTA-STS、DANE）
- 転送対応（SRS、ARC）
- エンドツーエンド暗号化（S/MIME、PGP）

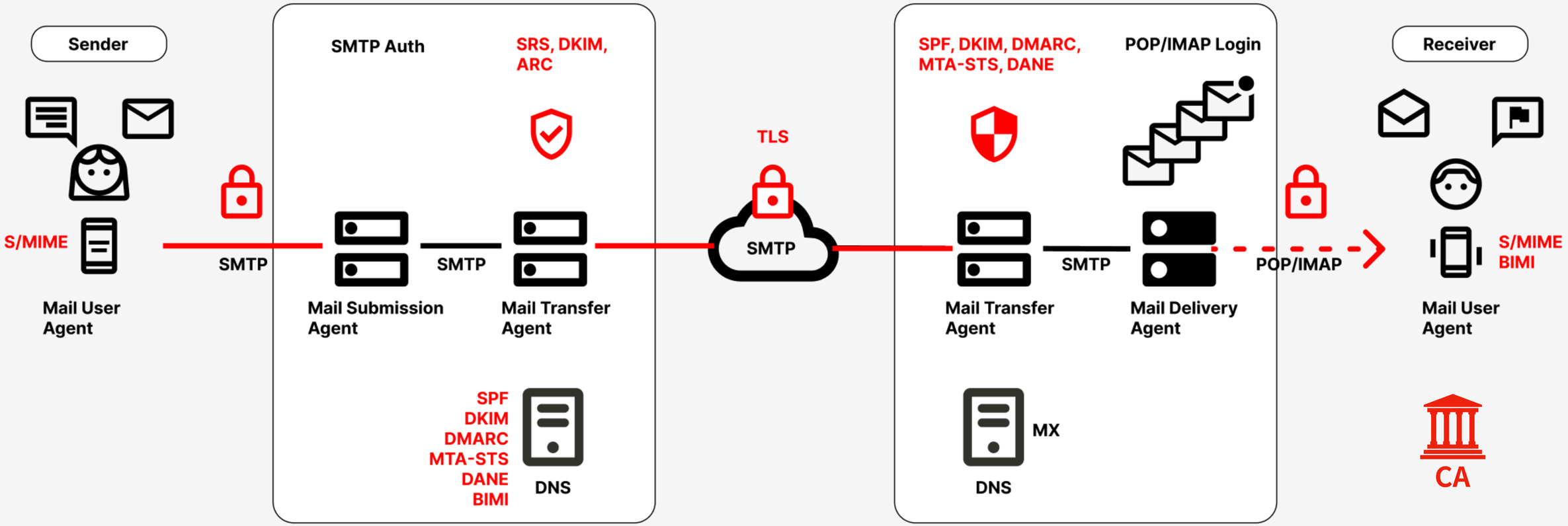
メールを送信して相手が受信するまでの全体像

複数のプロトコルと多くのサーバーコンポーネントが必要



長い年月をかけメールセキュリティを向上中

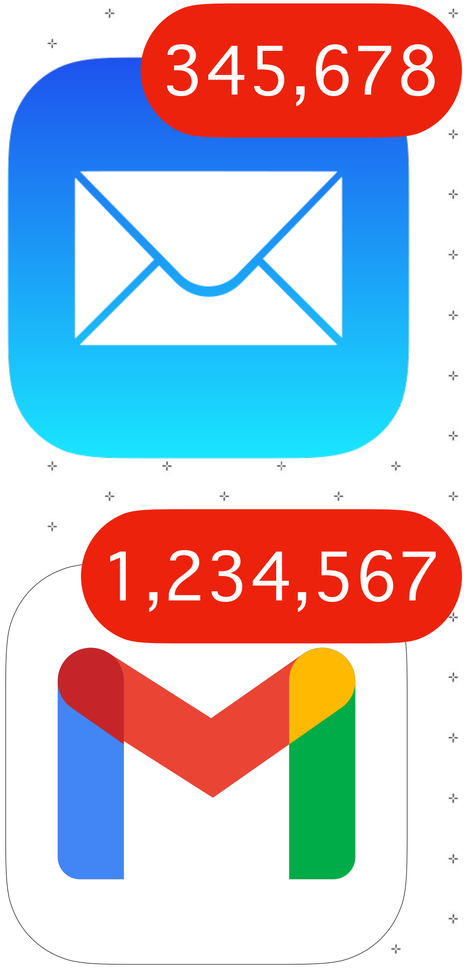
互換性維持を考慮する歴史的経緯とはいえ、複雑すぎますね…



メールに対する信頼性の低下

メールInboxは未読で溢れている

- メール配信が安価なマーケティングツールとなっていて、大量の広告メールがメールInboxを占有している
- 利用しているサービスのセキュリティインシデントにより、自分のメールアドレスが流出しており、大量のスパムメールを受信する
- そもそも、Instagram, X, TikTok, LINE, Discord, LINEといったSNSやメッセージプラットフォームで生活しているのでメールに問題があっても問題とは思わない



スパムメール対策

受信サーバにおける一般的なスパム対策の分類

- 解析・フィルターの活用（SpamAssassin, Amavis...）
- 送信元・受信者管理の活用（DNSBL, Reputation, Allowlist/Denylist）
- メール流量制御の活用（IP Throttling, Tarpit）
- 再送要求の活用（Greylisting）
- セッションフィンガープリントの活用（商用製品など）
- チャレンジ応答の活用（商用製品など）

Gmailのスパムメールの削減施策

2024年2月1日から以下の要件を満たす必要がある

- SPF, DKIMの認証
- IPの逆引き
- TLSによる接続
- メッセージフォーマット準拠 RFC5322

5,000/daily以上送信の場合

- 送信ドメイン認証
- ワンクリック配信解除 RFC8058

SAKURA internet

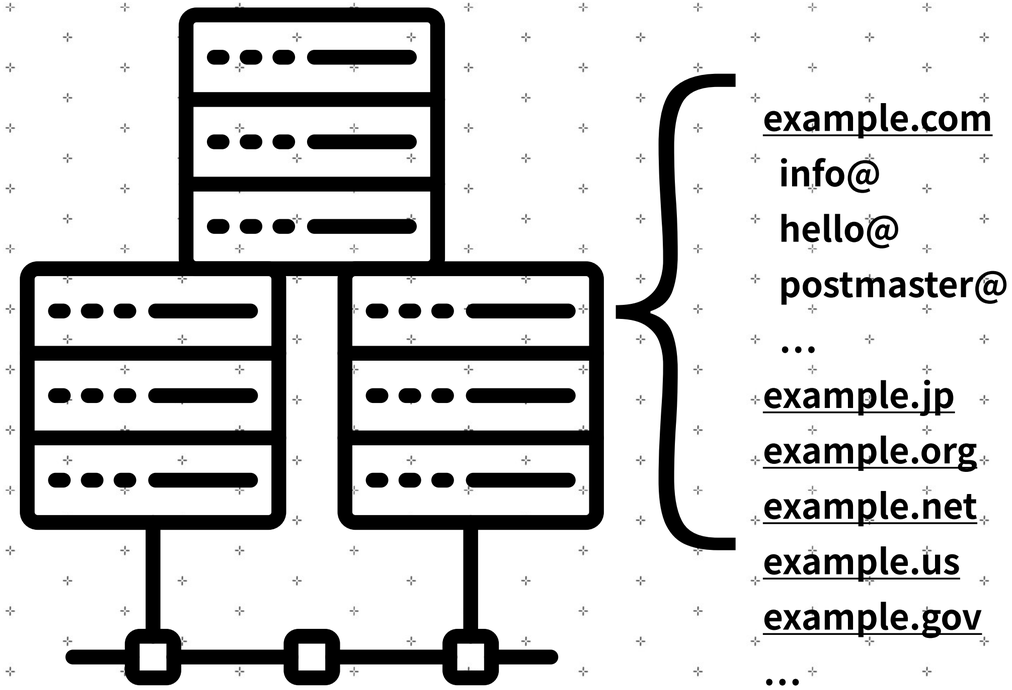
共有メールホスティングの事情と課題

SAKURA internet

共有メールホスティングのビジネスモデル

マルチテナントのためのリソース隔離と権限分離が重要

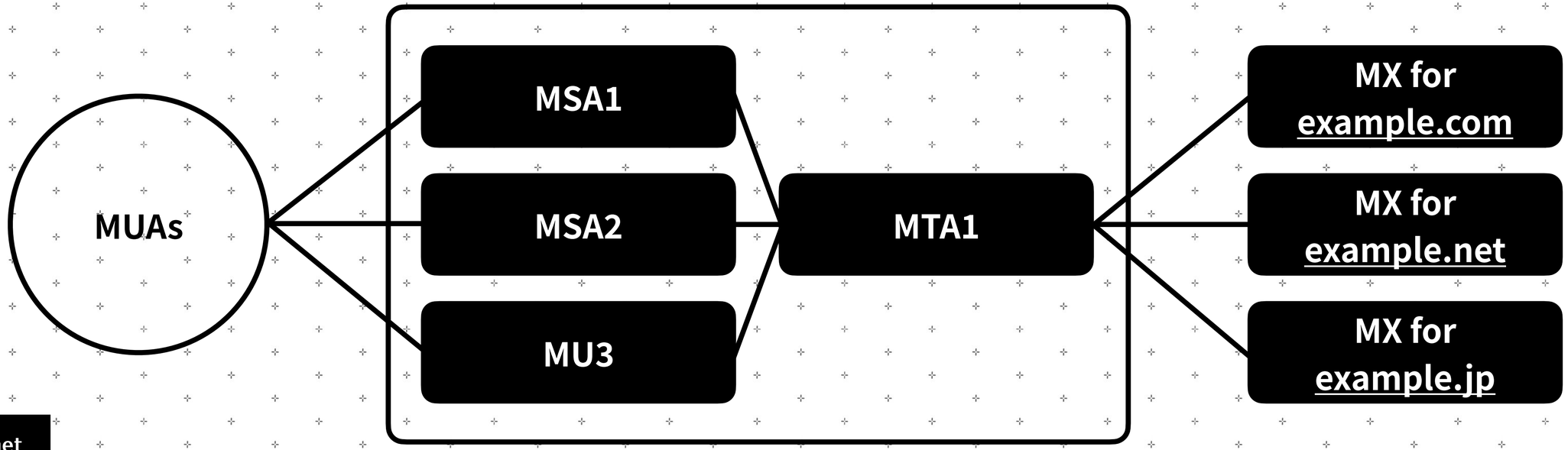
- メールホスティングは、テナントに対して仮想的に分割された物理サーバリソースを提供する
- テナントは複数のドメインを設定し、それに紐づくメールアカウントを使用する
- 1つの物理サーバに多くのテナントを収容することでサービス提供を安価にしている



共有メールホスティングの構成

メール送信にフォーカスした話

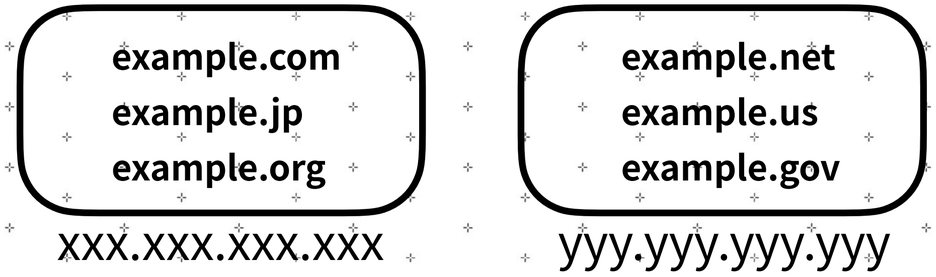
MUA（Mail user agent）からのメール送信を受け付けるMSA（Mail submission agent）が複数ある場合にIPv4アドレスの削減や運用の集約のためにリレーサーバを用いるケースを想定

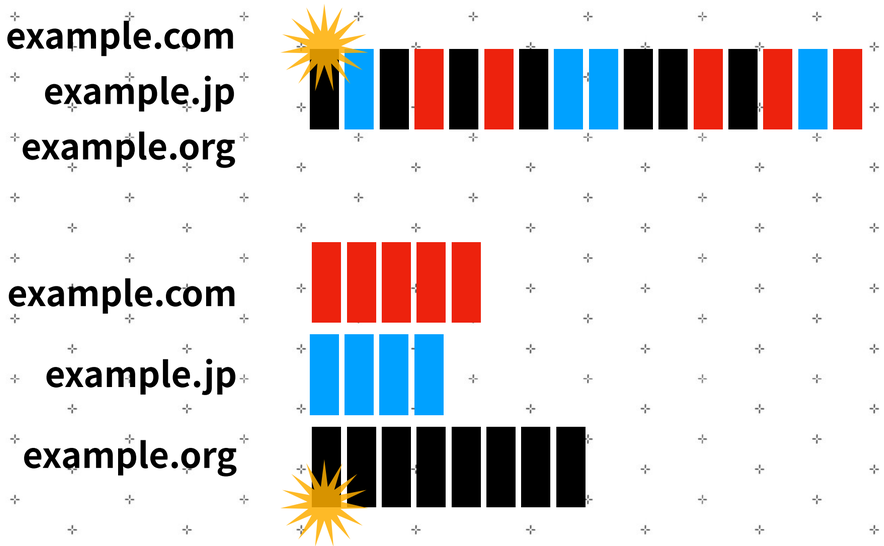


共有メールホスティングの事情

メール送信にフォーカスした話

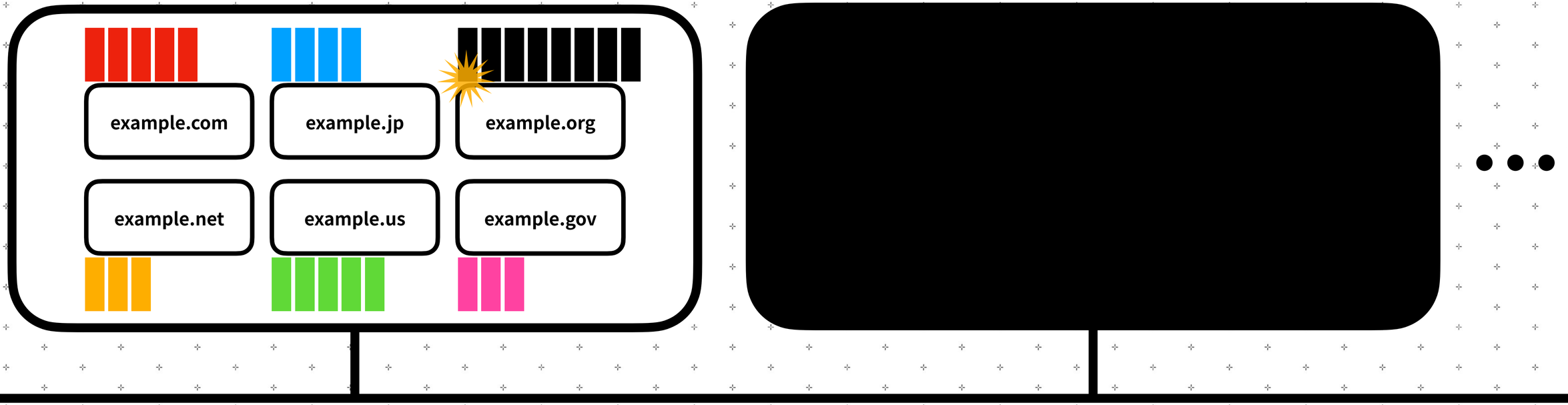
- ドメイン単位にIPv4を割り振ることはできないため、メール送信に使用するIPは集約したい
- 集約しすぎるとIPに起因するスロットリングなどの影響範囲が大きくなる
- 送信メールキューも同時に集約するとキュー滞留時に影響範囲が大きくなり運用が大変になる
- つまり、メールキューは分散しIPは適切な分量に集約するのが良い





コンテナ化でIP集約とメールキュー分散を実現する

「精緻に制御可能な恒常性のある高集積マルチアカウント型のメール基盤（DICOMO2018）」



各MTAの可観測性が低いという課題

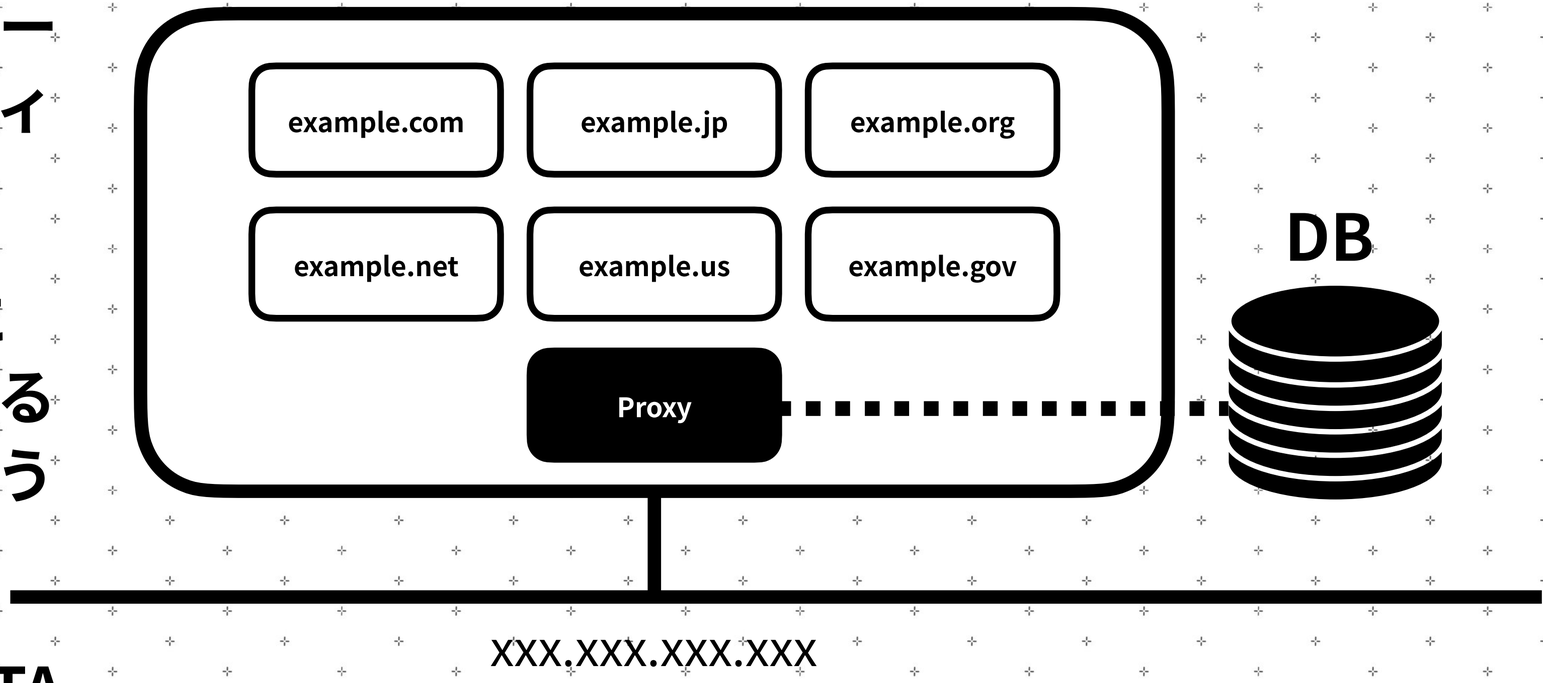
- MTAが分散することで出力するログを集約しパースすることによってメールシステム全体で何が起きているか把握出来るようにはなる
- 一方で受信サーバによるTarpitやThrottling のようなSMTPコマンドレベルの変化を検知することはできない
- そして、過去に何が起きていたかを確認することはできても、リアルタイムに問題を検知して修復することは不可能
- 例えばThrottlingが発生していることを検知し、発生の引き金となったドメインの送信をブロックし、他のドメインの送信に使うIPを付け替えるといった運用の自動化などを行える環境にしたい

透過型SMTPプロキシによる送信メールの可観測性向上を提案

SAKURA internet

コンテナホストごとに透過型SMTPプロキシを導入する

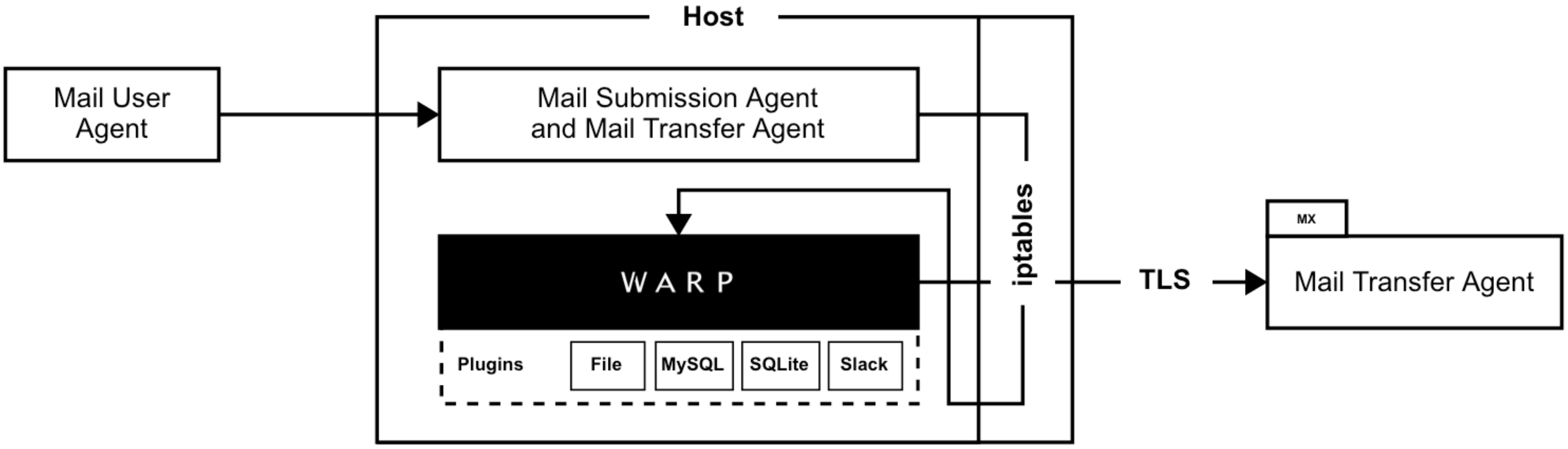
- 透過プロキシなのでメールキューはコンテナ側で管理し、ルーティングだけを変える
- SMTPプロトコルレベルでDBに通信情報蓄積し何が発生しているかリアルタイムで検知できるようにする
- 同一ホストのためコンテナのMTAとプロキシ間はTLSを使わない



透過型SMTP送信プロキシ: WARP

https://github.com/linyows/warp

- 一般的なSMTPプロキシで送信用に使えるものはないためGoで自作
- MTAプロセスからの25番宛先のパケットをiptablesでDNATし、本当の宛先をシステムコールから取り出して使っている
- 相手MXからのEHLOレスポンスに含まれるSTARTTLSを削除してMTAに返却しプロキシは相手MXとTLS接続する
- フックイベントを作っているのでプラグインから処理追加が可能
- カナリアリリースで本番導入検証済み


Host
Mail User Agent
Mail Submission Agent and Mail Transfer Agent
WARP
Plugins
File
MySQL
SQLite
Slack
iptables
TLS
MX
Mail Transfer Agent

取得できるデータ例

MySQLとSQLiteプラグインでは、Connectionsテーブルと Communicationsテーブルがあり、SMTPのやりとりが後者にレコードとして保存される

```
# sudo mysql -uroot -D warp
mysql> select * from connections;
+----------------------------+-------------+---------------+----------------------------+
| id                         | mail_from   | mail_to       | occurred_at                |
+----------------------------+-------------+---------------+----------------------------+
| 01FR74VW574PVQ5WGYE5RQATTG | root@sender | root@receiver | 2021-12-31 02:24:56.009302 |
| 01FR755XZKA594WA8SACQB4HC3 | root@sender | root@receiver | 2021-12-31 02:30:25.557302 |
+----------------------------+-------------+---------------+----------------------------+
2 rows in set (0.00 sec)

mysql> select communications.occurred_at, direction as d, substring(data, 1, 40) as data from
+----------------------------+----+------------------------------------------+
| occurred_at                | d  | data                                     |
+----------------------------+----+------------------------------------------+
| 2021-12-31 02:30:25.523678 | -- | connected to 192.168.30.50:25            |
| 2021-12-31 02:30:25.534128 | <- | 220 receiver ESMTP Postfix (Ubuntu)\r\n  |
| 2021-12-31 02:30:25.534692 | -> | EHLO sender\r\n                          |
| 2021-12-31 02:30:25.535251 | <- | 250-receiver\r\n250-PIPELINING\r\n250-SI |
| 2021-12-31 02:30:25.535399 | |< | 250-receiver\r\n250-PIPELINING\r\n250-SI |
| 2021-12-31 02:30:25.538790 | -- | pipe locked for tls connection           |
| 2021-12-31 02:30:25.538791 | |> | STARTTLS\r\n                            |
| 2021-12-31 02:30:25.538820 | >| | MAIL FROM:<root@sender> SIZE=327\r\nRCPT |
| 2021-12-31 02:30:25.539568 | |< | 220 2.0.0 Ready to start TLS\r\n         |
| 2021-12-31 02:30:25.539701 | |> | EHLO sender\r\n                          |
| 2021-12-31 02:30:25.547124 | |< | 250-receiver\r\n250-PIPELINING\r\n250-SI |
| 2021-12-31 02:30:25.547459 | -- | tls connected, to pipe unlocked          |
| 2021-12-31 02:30:25.547811 | -> | MAIL FROM:<root@sender> SIZE=327\r\nRCPT |
| 2021-12-31 02:30:25.554912 | <- | 250 2.1.0 Ok\r\n250 2.1.5 Ok\r\n354 End  |
| 2021-12-31 02:30:25.555126 | -> | Received: from sender (localhost [127.0. |
| 2021-12-31 02:30:25.556812 | <- | 250 2.0.0 Ok: queued as 1EA19412C8\r\n22 |
| 2021-12-31 02:30:25.559877 | -- | connections closed                       |
+----------------------------+----+------------------------------------------+
17 rows in set (0.00 sec)
```

検証環境での事例

SMTPセッション数10で100通を大手メールサービス宛に送る実験

1. 同時最大接続数制限に引っかかった

   451 4.7.652 The mail server [xxx.xxx.xxx.xxx] has exceeded the maximum number of connections.

2. レピュテーションの低さから受信拒否の判断をされた

   550-5.7.1 [xxx.xxx.xxx.xxx] Our system has detected that this message is likely suspicious due to the very low reputation of the sending domain. To best protect our users from spam, the message has been blocked.

検証環境での事例

SMTPセッション数10で100通を大手メールサービス宛に送る実験

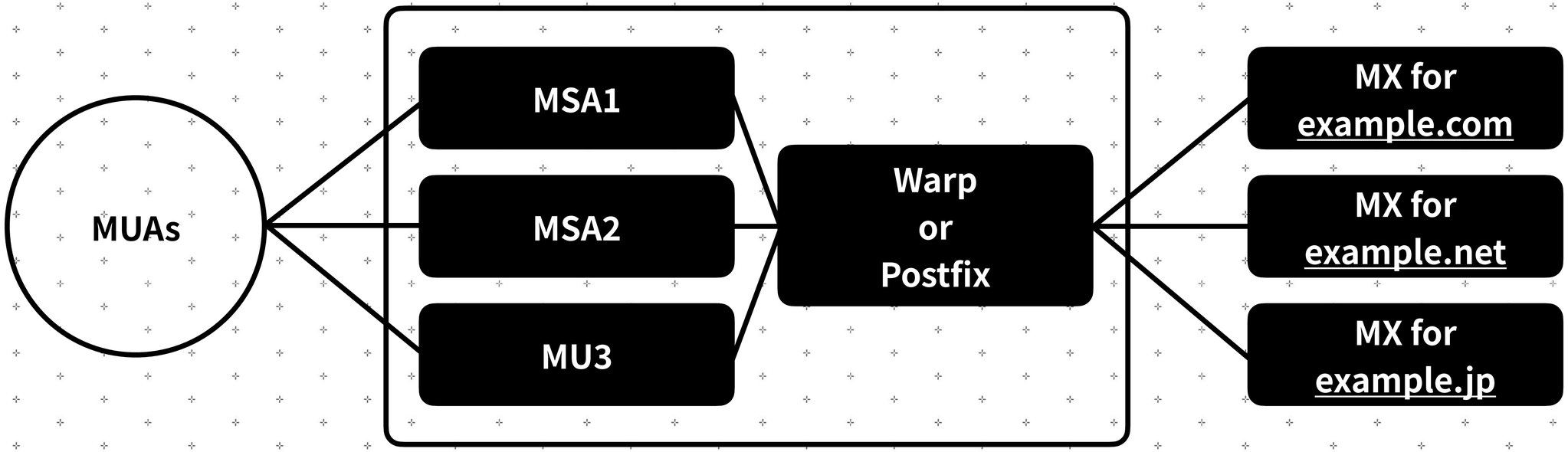
3. 実験途中からMAIL FROMとRCPT TOの間で20秒ほどの応答遅延が観測された

```
[2023-08-13T23:09:41.307892] -> MAIL FROM:<alice@mydomain>
                                RCPT TO:<outgoing@mac.com>
                                DATA
[2023-08-13T23:09:41.463738] <- 250 2.1.0 Ok
[2023-08-13T23:10:02.328554] <- 250 2.1.5 Ok
[2023-08-13T23:10:02.449577] <- 354 End data with <CR><LF>.<CR><LF>
```

既存手法との定量比較

既存手法のリレー方式と透過型プロキシで性能比較

集約サーバにWarpを使うパターンとPostfixを使うパターンで比較した



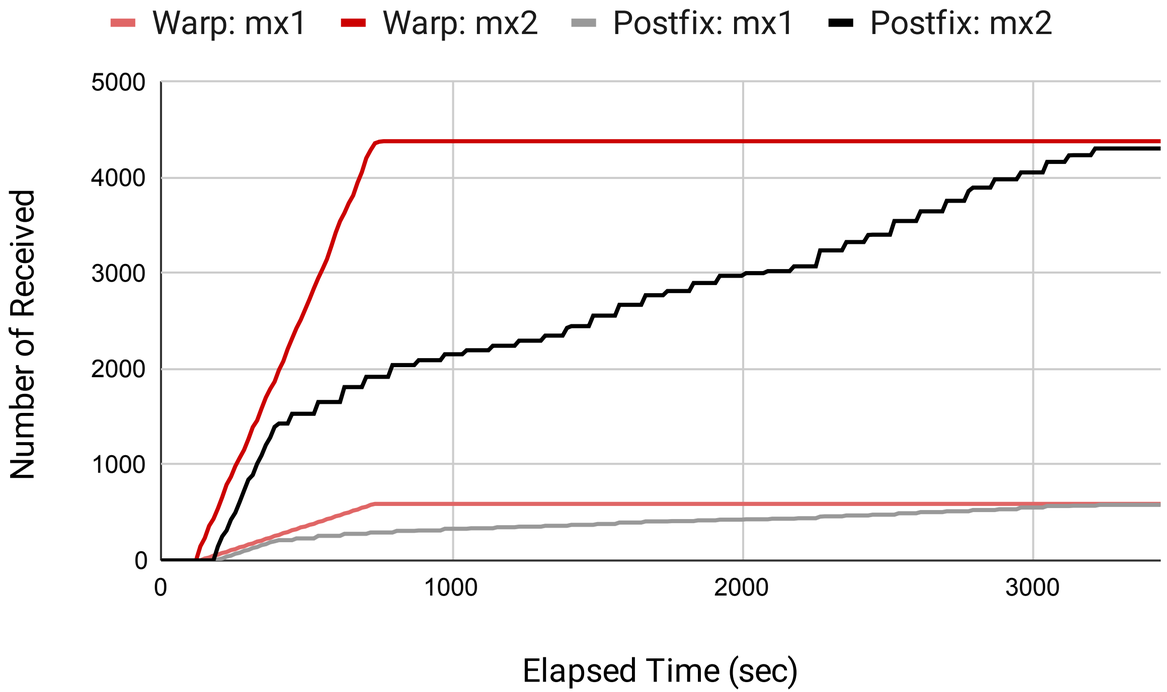
実験した内容

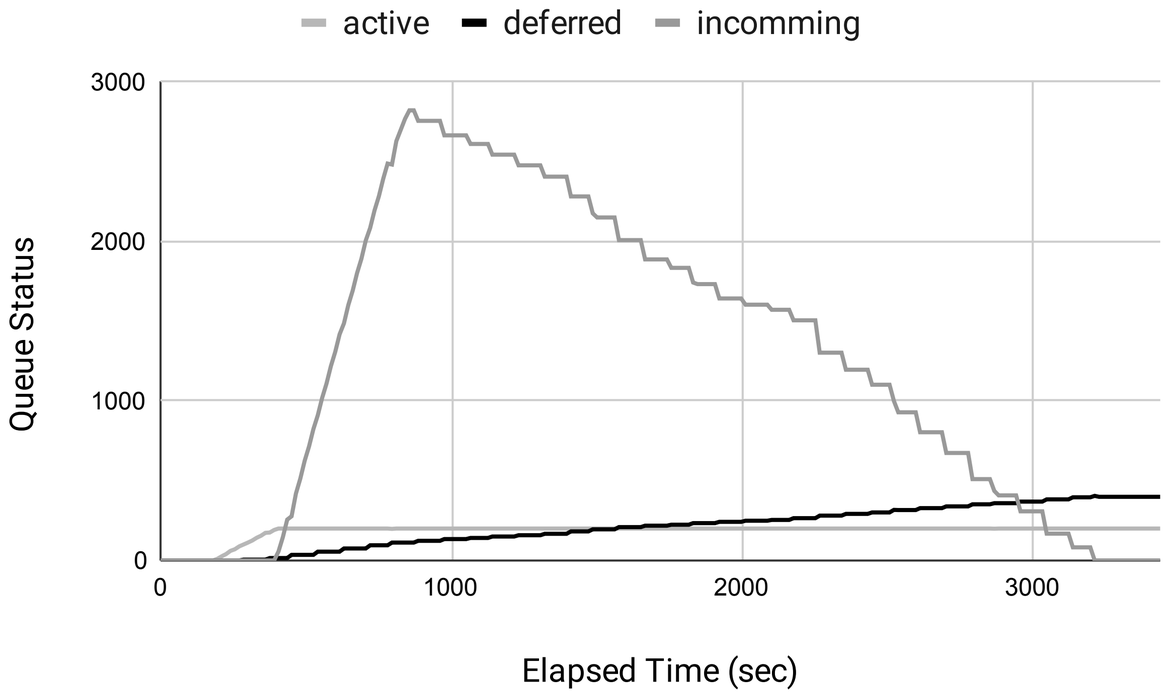
以下の3種類の実験を行った

- 送信メールキューが滞留した状態を再現しメールレイテンシーの変化を比較
- メールを大量に送信して集約サーバにおけるリソース消費の比較
- 受信サーバが行う同時接続数の制限状態を再現し、複数のMSA（テナント）から同一ドメインに対して送信し、送信量の変化を比較

送信メールキューの滞留再現し送信への影響を確認

Postfixのqmgr_message_active_limitを20000から200にして実験





送信メールキューの滞留再現しレイテンシーを比較

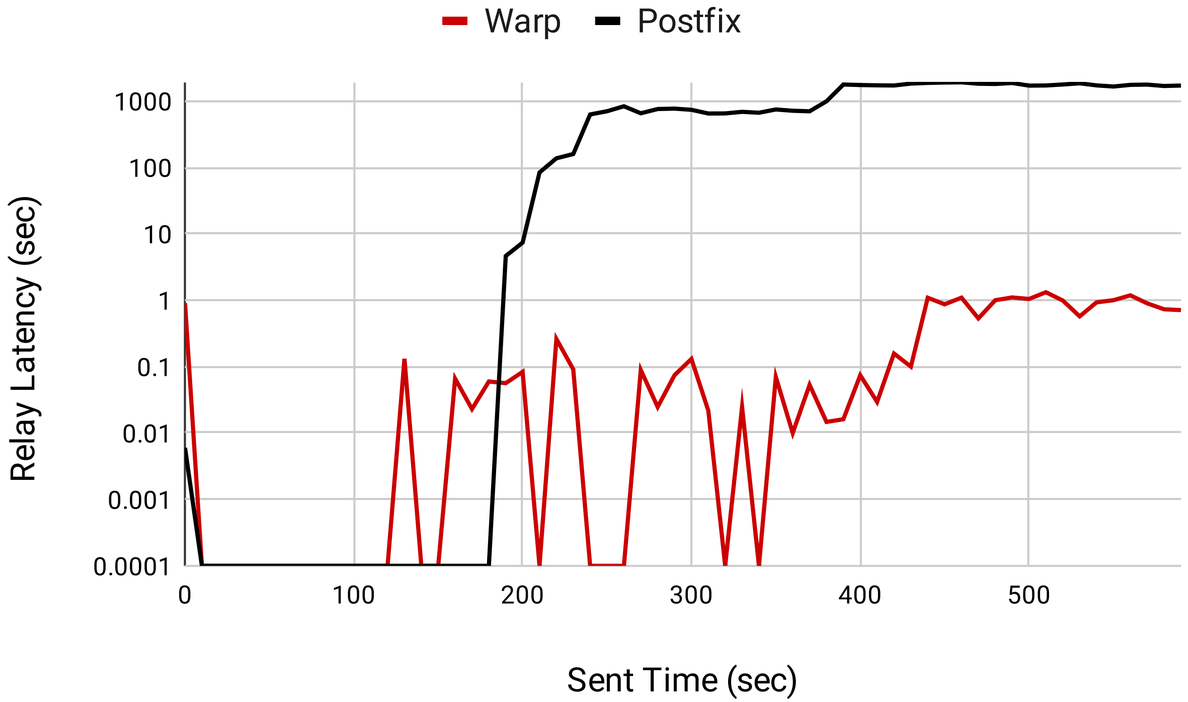
Postfixに発生したキューの滞留の影響がWarpにはない

End-to-End Latency

Relay Latency

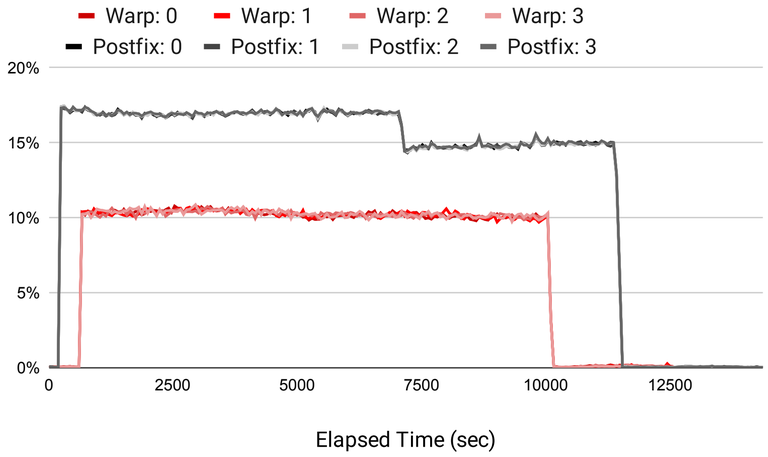
```
Return-Path: <alice@a.example.com>
X-Original-To: bob@a.example.jp
Delivered-To: bob@a.example.jp
Received: from mail1.example.jp (1-2-0-192.example.jp [192.0.2.1])
        by mx1 (Postfix) with UTF8SMTPS id D791E825DA6
        for <bob@a.example.jp>; Sun, 25 Aug 2024 08:38:42 +0000 (UTC)
Received: from msa2.example.com (2-2-168-192.example.com [192.168.2.2])
        (using TLSv1.3 with cipher TLS_AES_256_GCM_SHA384 (256/256 bits)
         key-exchange X25519 server-signature RSA-PSS (2048 bits))
        (No client certificate requested)
        by mail1.example.jp (Postfix) with UTF8SMTPS id EF56C1B00B7E
        for <bob@a.example.jp>; Sun, 25 Aug 2024 16:59:53 +0900 (JST)
Received: from localhost (unknown [172.18.0.1])
        by msa2.example.com (Postfix) with UTF8SMTP id 36D07825EAC
        for <bob@a.example.jp>; Sun, 25 Aug 2024 07:59:12 +0000 (UTC)
From: alice@a.example.com
To: bob@a.example.jp
Date: Sun, 25 Aug 2024 16:59:11 +0900
Subject: Experiment: Case 1 - 361d42cbeac2b6b2aafb937f9247f25d2282283f

This is a test mail.
```

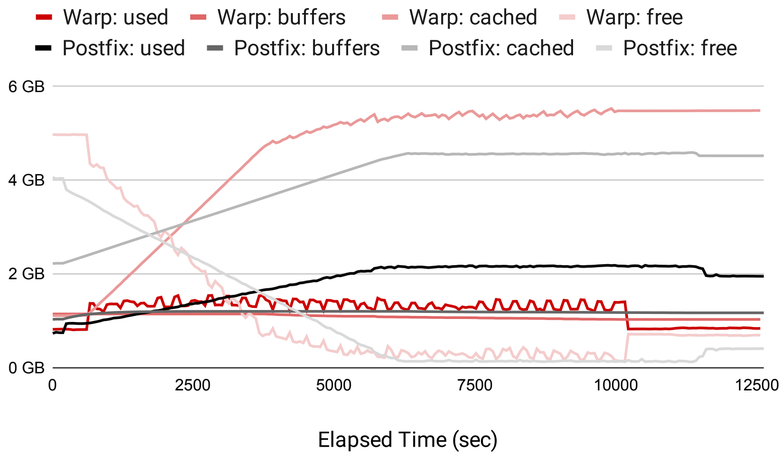


サーバリソース消費の比較

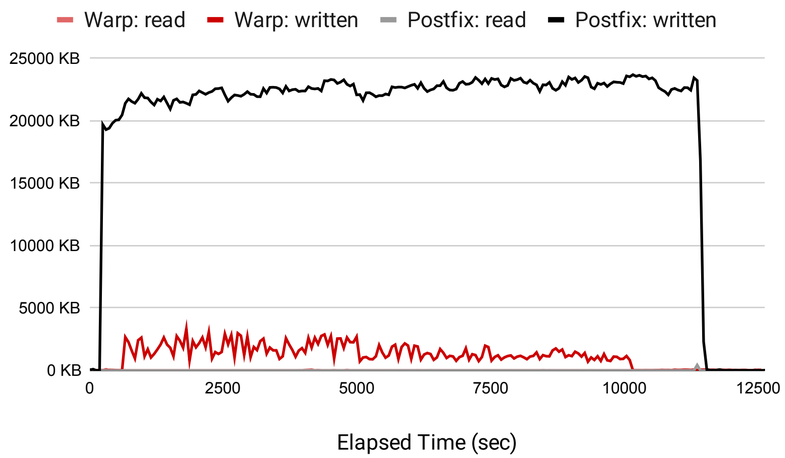
メールを転送するのとパケットを転送する違いからDisk I/Oに顕著な差が出た



CPU usage



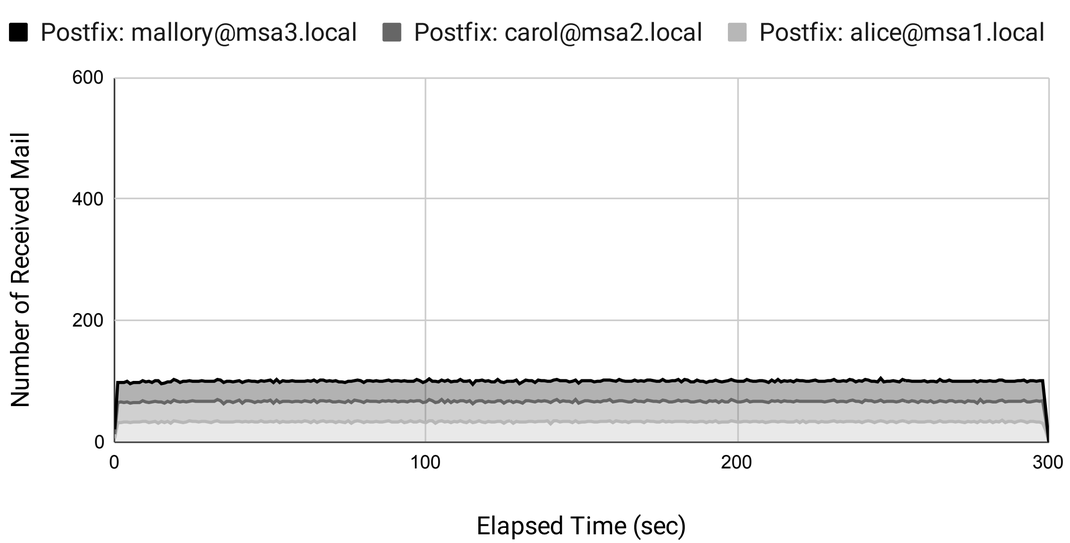
Memory usage

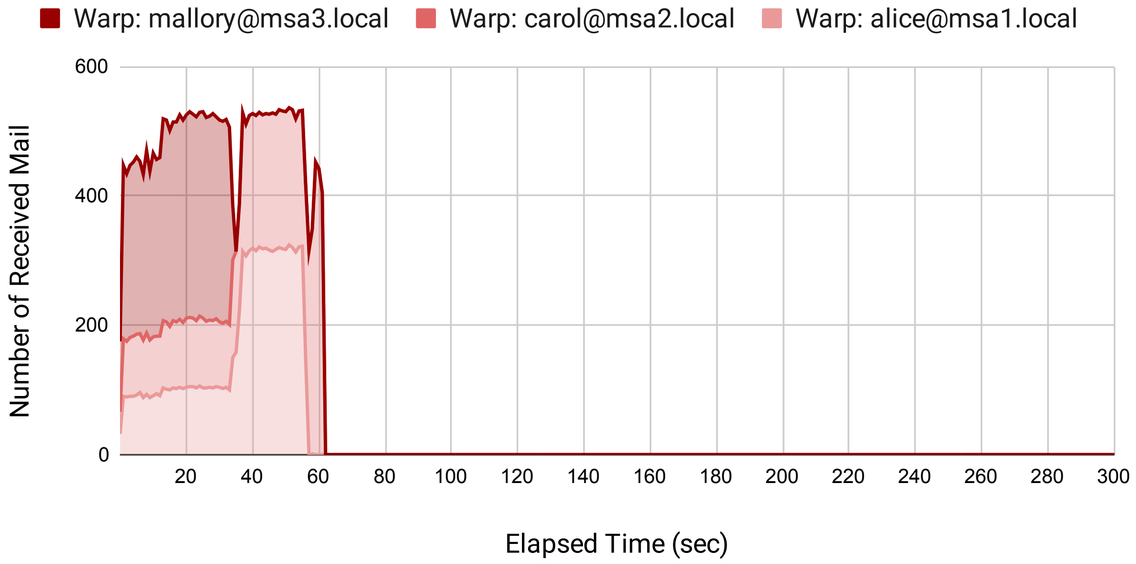


Disk I/O usage

同時接続数制限を再現しテナントごとの送信量の比較

Warpの送信完了は短いが送信流量のコントロールをしないことによる送信量の偏りが発生





透過型SMTPプロキシの定量評価については来月の「インターネットと運用技術シンポジウム (IOTS 2024)」で研究報告として論文掲載予定

今後の課題

実運用に向けての課題

さまざまな機能が不足しているためたくさん実装が必要

- MTA-STSやDANEなどへの対応：送信先である外部MXとTLS通信するのは透過型SMTPプロキシになるため、経路暗号化強制のための実装が必要である
- 3つ目の実験により分かったSMTPコネクションの公平性への対応：同一宛先への送信は複数IPでラウンドロビンし同時接続数の制限を受けないようにすることや、同じMSAがSMTPセッションを使い続けないようにあえて切断するなどを検討する
- 複数IP対応：内部レピュテーションによるものや受信サーバレスポンス変化によるものによって複数IPの使い分け機能を追加する
- メールフォーマットの緩和：現状RFCに準拠していないメールは送信できないため、ある程度緩和する
- メトリクス対応：Observability製品と連携できる機能を実装する

謝辞：

本研究は JSPS科研費20K11791「軽量コンテナによる大規模高集積メールホスティング基盤における送信機能の高機能化」の助成を受けたものです。