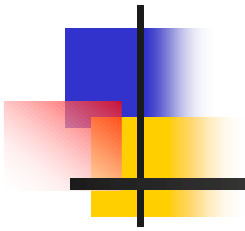
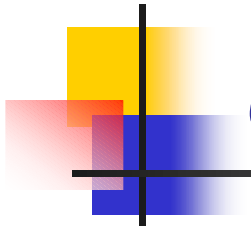


メールシステムの現状

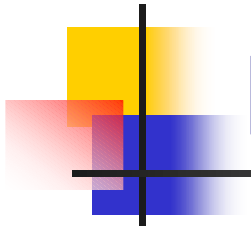
~迷惑メールを受けない投げないために~

安藤一憲
ando@ppml.tv



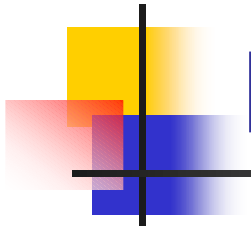
このチュートリアルの構成

- DNSの重要性
- メールの基礎知識
  - 普段目にするメールについての解説
- メール配送のモデル
  - MX配送、static配送の使い分け
  - 設定のまとめ
- spamの現状と対策
  - spam絶滅作戦



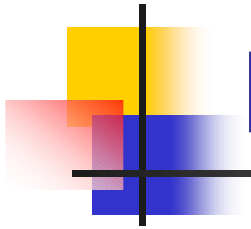
DNSの重要性

- あるドメインのリソース情報へ到達する鍵
  - 順次NSを手繰ってデータを引きに来る仕組み
    - 手繰れないと破綻
  - IPアドレス付け替え時、ドメイン変更時に注意
  - ほぼ全てのサービスに影響
    - FQDNを用いるもの全て
  - NSを死守すべし
    - 全ての基礎となるサービスのひとつという位置付け



DNSとメール

- user@example.gr.jp
  - ここから配送先をどう見つけるか?
  - 手がかりはドメイン名の部分
- example.gr.jpのMXレコードを調べる
  - 配送にはMXとサーバのAレコードが必要
    - 最も効率が良いのは、MXを聞いたらMXだけではなくAが同時に返ってくる場合
    - 答えるnamedがMXとAを両方知っているのがベスト



MXはCNAMEではいけない

- O DontExpandCnames=False
  - RFC822,1123的にはたぐるのが正しい
  - IETFはCNAMEをたぐらない方向に動いている
  - sendmailではオプションになった
- DNSのMXにはCNAMEを指定してはいけない
  - RHSにはAを書く
  - そのAはMXを答えるnamedが知っていると良い

メール本体とエンベロープ

Mail From: ando@ppml.tv
Rcpt To: motonori@media.kyoto-u.ac.jp

エンベロープ
SMTP的配送情報

```
From: Kazunori ANDO <ando@ppml.tv>
To: Motonori NAKAMURA <motonori@media.kyoto-u.ac.jp>
Subject: Re: smtpfeed-1.19
Message-Id: <ANDO.SB10224@ns0.ppml.tv>

（空行のあとが本文）
```

メール本体
ヘッダは基本的に配送とは関係ない

メール本体とエンベロープ（2）

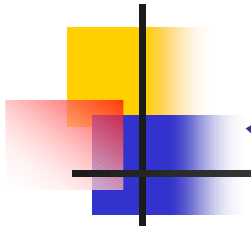
Mail From: motonori@media
Rcpt To: ando@ppml.tv

配送経路情報が記録される
（記述形式は任意）

```
Received: from query.media.kyoto-u.ac.jp
        by ns0.ppml.tv with ESMTP
        for <ando@ppml.tv>; 3 Dec 2003 10:00:01 +0900
Return-Path: motonori@media.kyoto-u.ac.jp
From: Motonori NAKAMURA <motonori@media.kyoto-u.ac.jp>
To: Kazunori ANDO <ando@p      tv>
Subject: Re: smtpfeed-1.19
Message-Id: <ANDO.SB10234
```

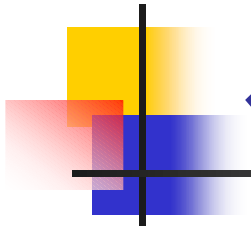
（空行のあとが本文）

Return-Path:にSMTPのMail From:が
保存される（設定による）



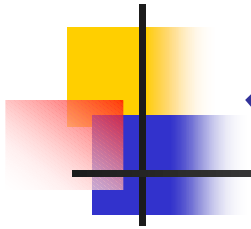
メール本体とエンベロープ（3）

- To: に書くとそこに送られるのは…
  - 実はMUAの仕業
  - To: ヘッダに書いたアドレスをSMTP的な配送先情報としてMTAに渡しているだけ
  - 例えば、To:ヘッダがメーリングリストのアドレスなのに自分にメールが届くのはこのため



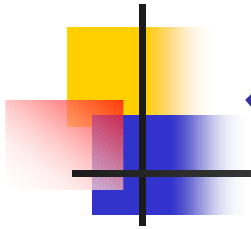
ヘッダの話（1）

- Field-name: Field-body（standard）
  - From:　差出人アドレス
  - Sender：差出人アドレスが不明確な場合に差出人を明示
  - To:　宛先アドレス
  - Cc:　カーボンコピー
  - Reply-To:　返信先アドレス
  - Message-Id: ５年間固有のID
  - Subject:　タイトル
  - Date:　差出時間
  - Return-Path: エラー返信先アドレス



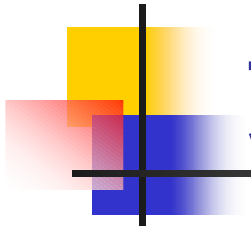
ヘッダの話（2）

- Field-name: Field-body（standard）
  - Received:　　配送経路
  - In-Reply-To:　どのメールに返信したかを示す
  - References:　どのメールに返信したかを示す
- Resent系（メールを再送信する場合の）
  - Resent-From:　　差出人アドレス
  - Resent-Sender:　差出人アドレスが不明確な場合に明示
  - Resent-Reply-To:　返信先アドレス
  - Resent-Message-Id: ５年間固有のID
  - Resent-Date:　　再送信日時



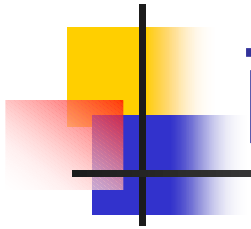
ヘッダの話（3）

- Field-name: Field-body
  - Precedence: 配送優先度
  - X-Authentication-Warning: アドレス詐称（？）
- （おまけ）MLドライバ等の付けるヘッダ
  - X-MLServer: fml
  - X-ML-System: ppml
  - X-Ml-Version: kkml
  - X-Distribute: distribute
  - Delivered-To: qmail等



文字の話

- 機種依存文字を使ってはいけない
  - ①②③④⑤⑥⑦⑧⑨⑩
  - Ⅰ Ⅱ Ⅲ Ⅳ Ⅴ Ⅵ Ⅶ Ⅷ Ⅸ Ⅹ
  - ﾄｳｾﾞﾝﾊﾝｶｸｶﾀｶﾅﾓﾀﾞﾒ
- 漢字コードはISO-2022-JPを使用する
  - SJISだめ、EUCだめ、UNICODEだめ
- さらなる制約もある
  - 例えばISO-8859-1な文字（ウムラウト付き文字等）とISO-2022-JPな漢字はメール上で混在できない



配送の実際(1)

- 宛先アドレスのMXを引いてみる

  MX　10　mail-g1.example.gr.jp

  MX　10　mail-g2.example.gr.jp

  - この場合はランダムでどちらかに配送

  MX　10　mail-g1.example.gr.jp

  MX　20　mail-g2.example.gr.jp

  - この場合は10の方に配送して駄目だったら20へ

配送の実際(2)

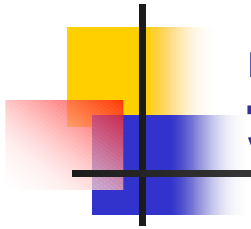
- MX RRの他にも答がいっぱい返ってくる

```
example.gr.jp   MX  10   mail-g1.example.gr.jp
example.gr.jp   MX  20   mail-g2.example.gr.jp
```

MX RR

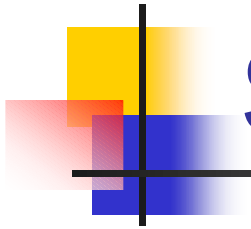
```
example.gr.jp   NS   ns1.example.gr.jp
example.gr.jp   NS   ns2.example.gr.jp
mail-g1.example.gr.jp   A   202.250.31.150
mail-g2.example.gr.jp   A   202.250.31.151
ns1.example.gr.jp    A  202.250.31.148
ns2.example.gr.jp    A  202.250.31.149
```

additional information



実際の配送(3)

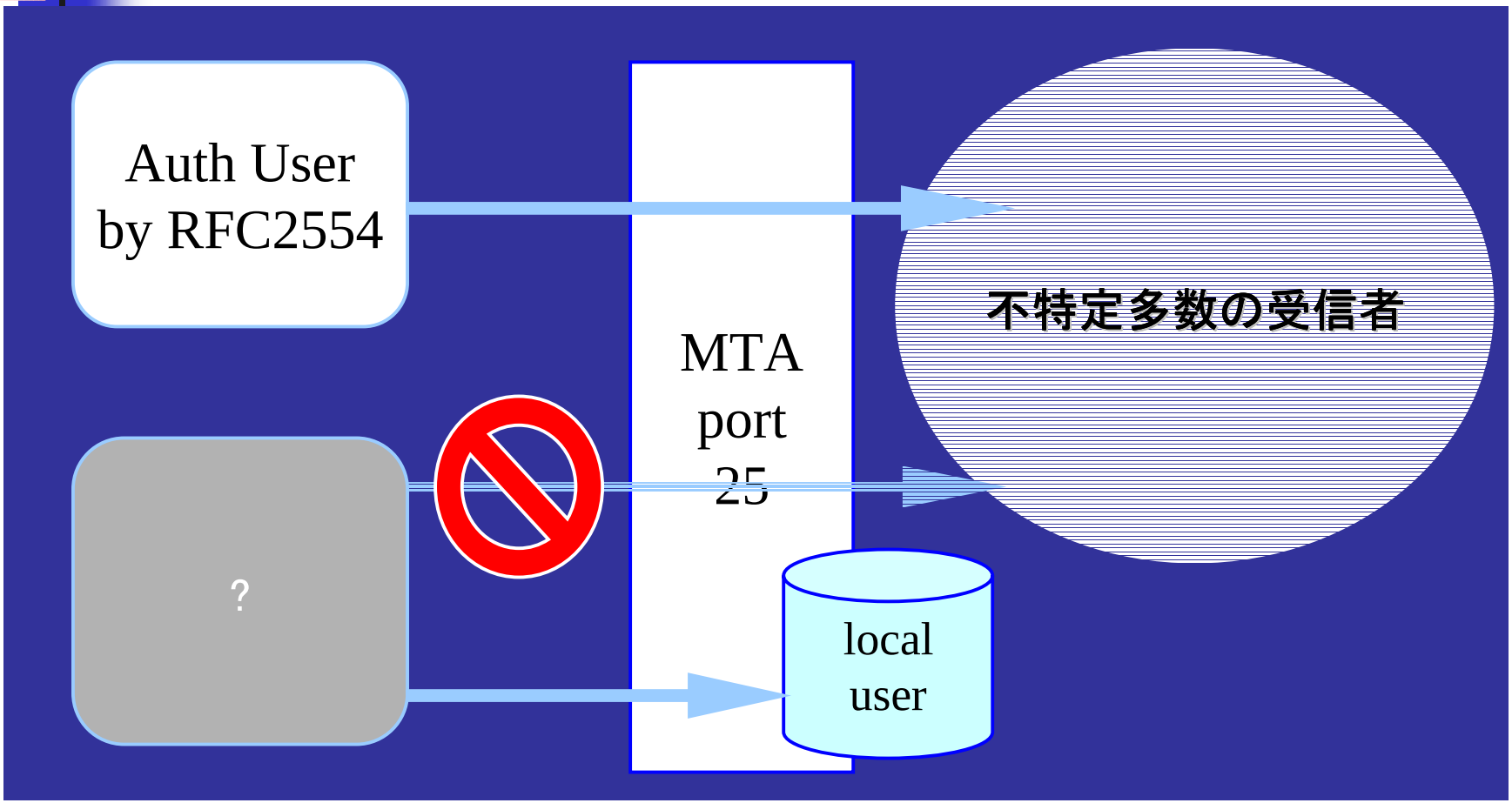
- Additional Information
  - MXを聞かれたNSがMXのAも知っている
    - MXとAがわかれば実際に接続しにいける
    - 1回のqueryで済むので効率が良い
    - MX RRを保持しているNSがAも保持することが重要
  - Additional InformationとMTA
    - 例えばSMTPfeedはAdditional Informationを利用
    - MTAが利用しなくても手元のnamedがcache
      - Aを聞きに行くとそこが答えるので速い

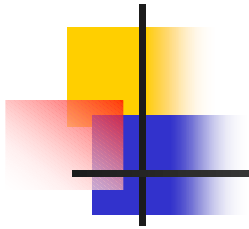


SMTP Authentication（RFC2554）

- SASL（RFC2222）を利用したRelay認証
  - sendmail-8.12では
    - 必要な作業
      - cyrus SASLライブラリをインストール
      - SASLを利用するようにsendmailをコンパイル
      - /usr/local/lib/sasl/Sendmail.confの準備（必要なら）
      - /etc/sasldb.dbの準備（saslpasswdコマンドでユーザ登録）
      - sendmail.cfの設定追加
    - 認証を通るとそのサーバ経由のRelay配送を許可

SMTP Authentication（RFC2554）

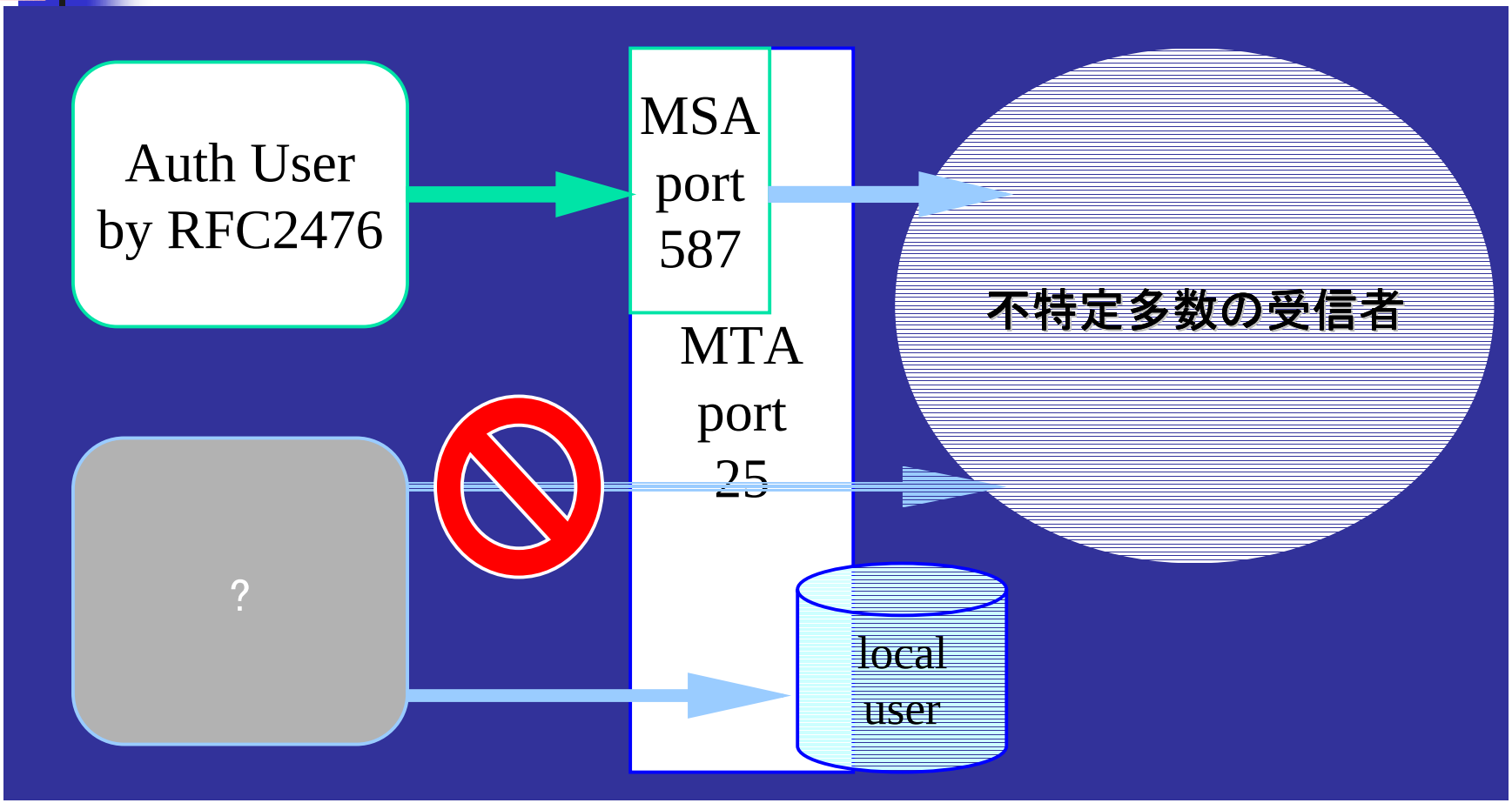


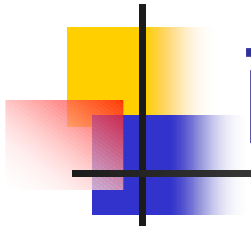


Message Submission（RFC2476）

- MSA（Message Submission Agent）
  - メールを「出す」新たな枠組み
    - Relayと区別することでSPAM不正中継を防止
      - SMTPではlocal宛のメールしか受けない
      - Submissionによる発信は自分のサイトからの接続だけを許可してさらに認証をかける
    - port 587
      - sendmail-8.11以降はdefaultでMSAになる
      - MSP（MessageSubmissionProgram/クライアント）からの接続を受け付ける

Message Submission（RFC2476）

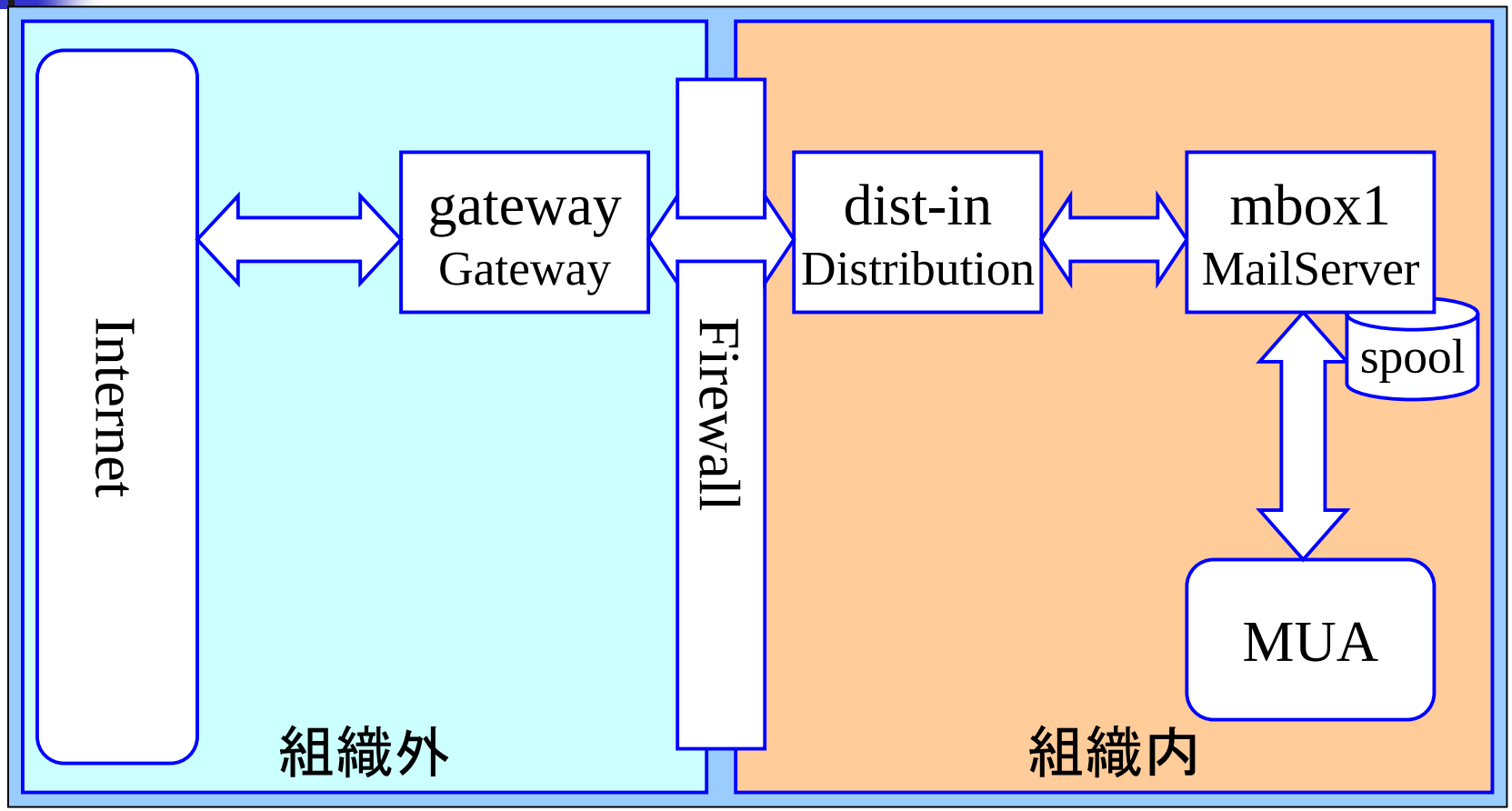




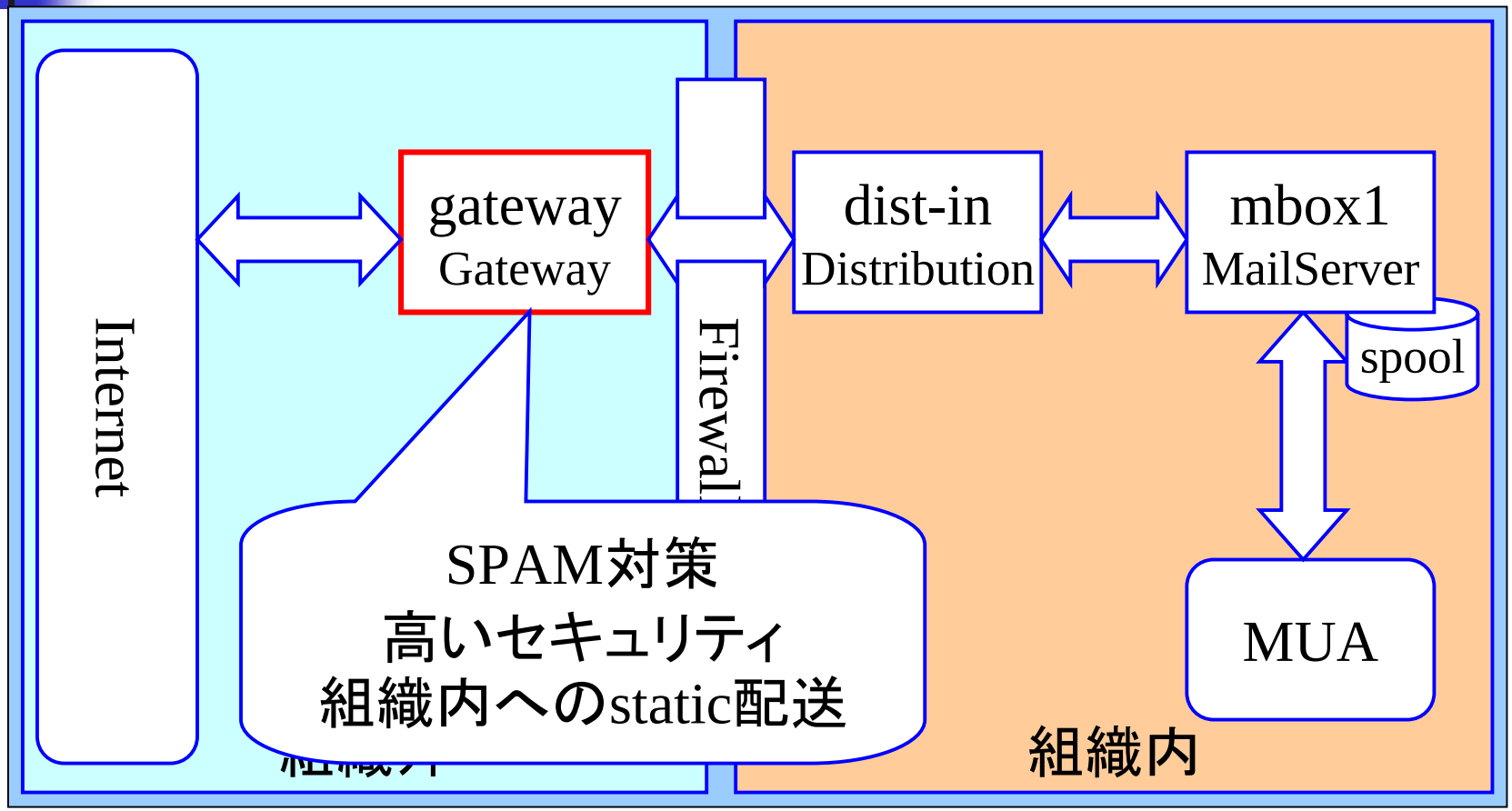
配送設定の基本要素

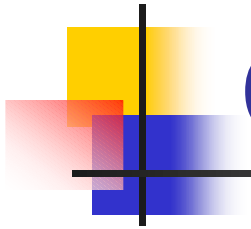
- MX配送かstatic(静的)配送か?
  - 対外配送はMX配送
  - 組織内部の配送はどちらか選択
    - 組織内部で独自のDNSの定義をしている場合
    - 集中サーバならstatic(mailertable)でもいける
  - resolv.confで参照するDNSサーバを指定

配送モデル



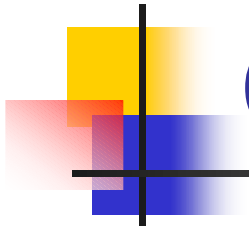
配送モデル:Gateway





Gateway

- 特別に必要な機能
  - 内側サーバへのstatic配送
    - FEATURE(`mailertable‘)
  - スパム不正中継の防止対策
    - FEATURE(`access_db’)
    - FEATURE(`blacklist_recipients')
    - Milter



Gateway: mailertable

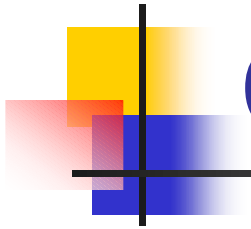
- static配送ルールを書く
- 設定ファイル名は/etc/mail/mailertable

/etc/mail/mailertable

```
.example.gr.jp      smtp:[dist-in.example.gr.jp]
.example.ad.jp      smtp:[non-mx.example.ad.jp]
.example.com        esmtp:mx.example.co.jp
```

```
# makemap hash /etc/mail/mailertable < /etc/mail/mailertable
```

このコマンドでmailertable.dbが生成され本設定が完了



Gateway: access_db

- 拡張されたspamlistの設定
- 設定ファイル名は/etc/mail/access

/etc/mail/access

```
spammers.net                    REJECT
spammer@ube.com                 ERROR:5.7.1:551 Relay denied
spam@uce.uce.com                DISCARD
example.gr.jp                   RELAY
localhost                       RELAY
127.0.0.1                       RELAY
```

```
# makemap hash /etc/mail/access < /etc/mail/access
```

このコマンドでaccess.dbが生成され本設定が完了

Gateway: blacklist_recipient

- 自ドメインのあるアドレスが狙われた場合の措置手段
- /etc/mail/accessに設定を付加できるようになる

/etc/mail/access

```
bogus_user@                 REJECT
bogus.example.gr.jp         ERROR:550 Bogus host
junk@other.example.gr.jp    ERROR:550 Mailbox unavailable
```

```
# makemap hash /etc/mail/access < /etc/mail/access
```

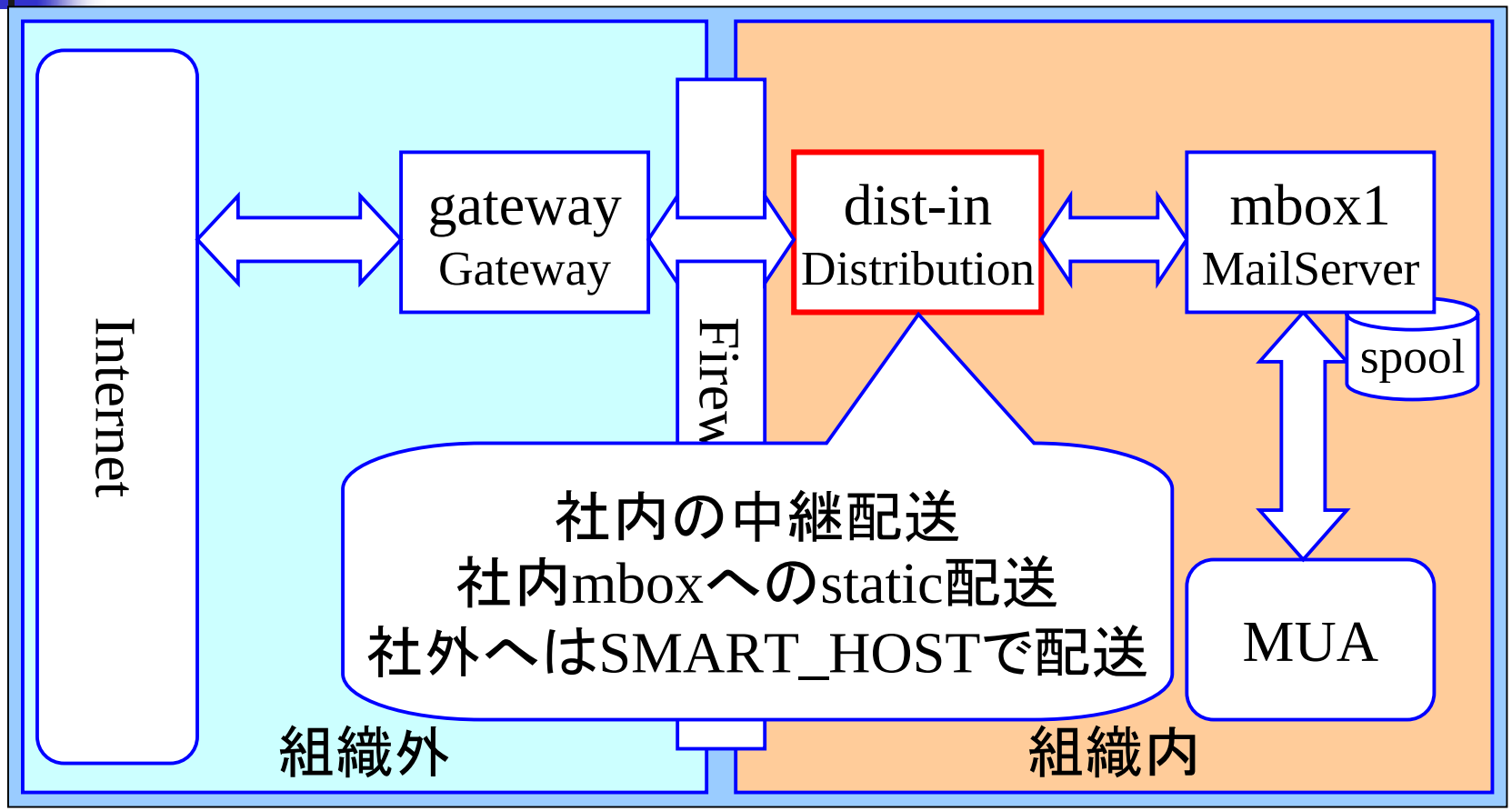
このコマンドでaccess.dbが生成され本設定が完了

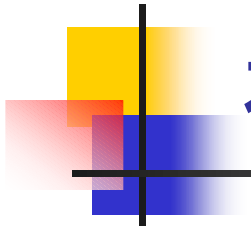
Gateway: config.mcファイル

```
divert(0)dnl
VERSIONID(`$Id: config.mc,v 1.1 2001/12/4 13:48:05 ando Exp $')
OSTYPE(bsd4.4)dnl
DOMAIN(generic)dnl
FEATURE(`nocanonify')dnl
FEATURE(`mailertable')dnl
FEATURE(`access_db')dnl
INPUT_MAIL_FILTER(`myfilter', `S=local:/var/run/perl.sock')dnl
MAILER(local)dnl
MAILER(smtp)dnl
define(`confDOMAIN_NAME',`$w.$m')dnl
```

```
cd ${SENDMAIL_SRC}/cf/cf
make config.cf
make install-cf CF=config
```

配送モデル:社内中継サーバ





社内中継サーバ

- 特別に必要な機能
  - 社内メールサーバへのstatic配送
    - FEATURE(`mailertable‘)の利用
  - 自ドメイン以外へのメールをGatewayへ
    - クラス SMART_HOST にGatewayを設定

社内中継サーバ: mailertable

- 社内でのstatic配送ルールを書く
- 設定ファイル名は/etc/mail/mailertable

/etc/mail/mailertable

```
sub1.example.gr.jp      smtp:[mbox1.example.gr.jp]
sub2.example.gr.jp      smtp:[mbox2.example.ad.jp]
sub1.example.ad.jp      smtp:[192.168.10.25]
```

```
# makemap hash /etc/mail/mailertable < /etc/mail/mailertable
```

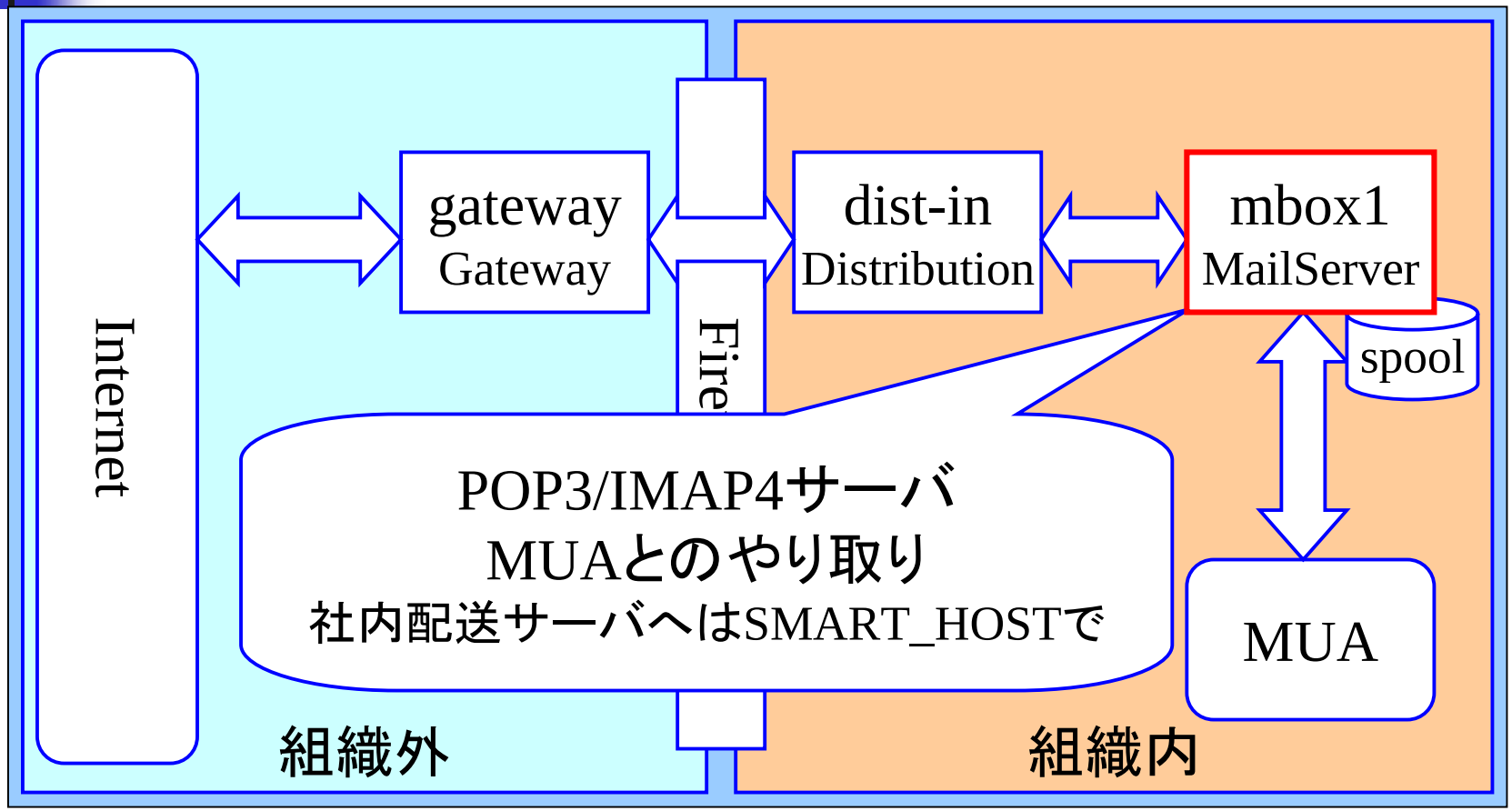
このコマンドでmailertable.dbが生成され本設定が完了

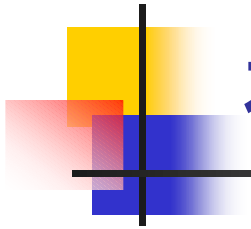
社内中継サーバ: config.mcファイル

```
divert(0)dnl
VERSIONID(`$Id: config.mc,v 1.1 2000/12/17 22:48:05 ando Exp $')
OSTYPE(linux)dnl
DOMAIN(generic)dnl
FEATURE(`mailertable')dnl
MAILER(local)dnl
MAILER(smtp)dnl
define(`SMART_HOST',`gateway.example.gr.jp')dnl
```

```
cd ${SENDMAIL_SRC}/cf/cf
make config.cf
make install-cf CF=config
```

配送モデル：社内メールサーバ





社内メールサーバ

- 特別に必要な機能
  - 社内中継サーバへのstatic配送
    - クラス SMART_HOST の利用
  - 知らないドメインでもそのまま中継に渡す
    - 自分のドメインを付加しない
  - ドメイン名のマスカレード
    - マシン名だけ含まないアドレスでメールを出したい

社内メールサーバ: config.mcファイル

```
divert(0)dnl
VERSIONID(`$Id: config.mc,v 1.1 2000/12/17 22:48:05 ando Exp $')
OSTYPE(linux)dnl
DOMAIN(generic)dnl
FEATURE(`nocanonify')dnl
MASQUERADE_AS(`example.gr.jp')dnl
MASQUERADE_DOMAIN(`myhost.example.gr.jp')dnl
FEATURE(`limited_masquerade')dnl
FEATURE(`masquerade_envelope')
FEATURE(always_add_domain)dnl
MAILER(local)dnl
MAILER(smtp)dnl
Dmexample.gr.jp
define(`SMART_HOST',`dist-in.example.gr.jp')dnl
```

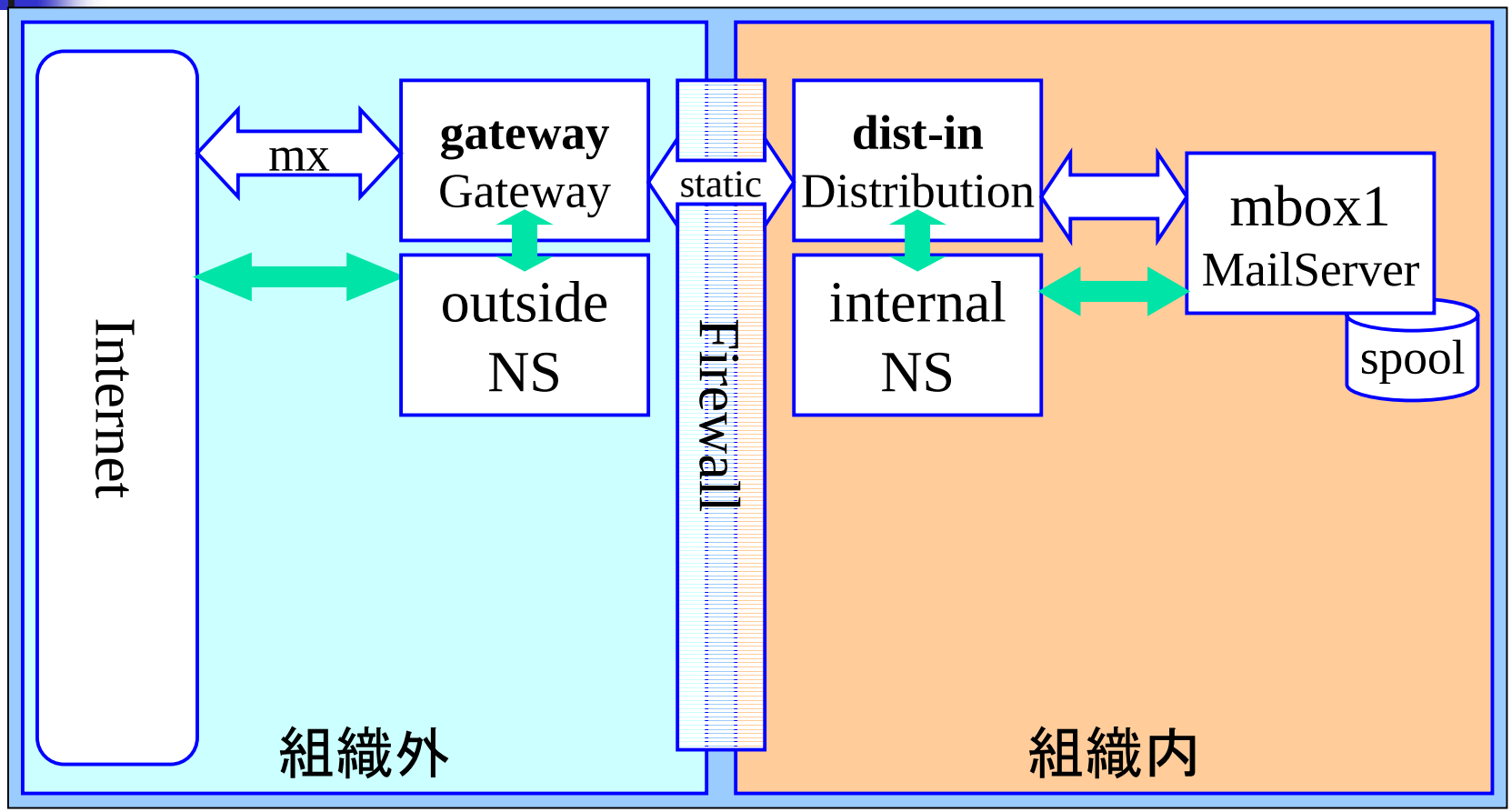
マスカレードするドメインの範囲を指定。

```
MASQUERADE_DOMAIN(`example.gr.jp')dnl
FEATURE(masquerade_entire_domain)dnl
```

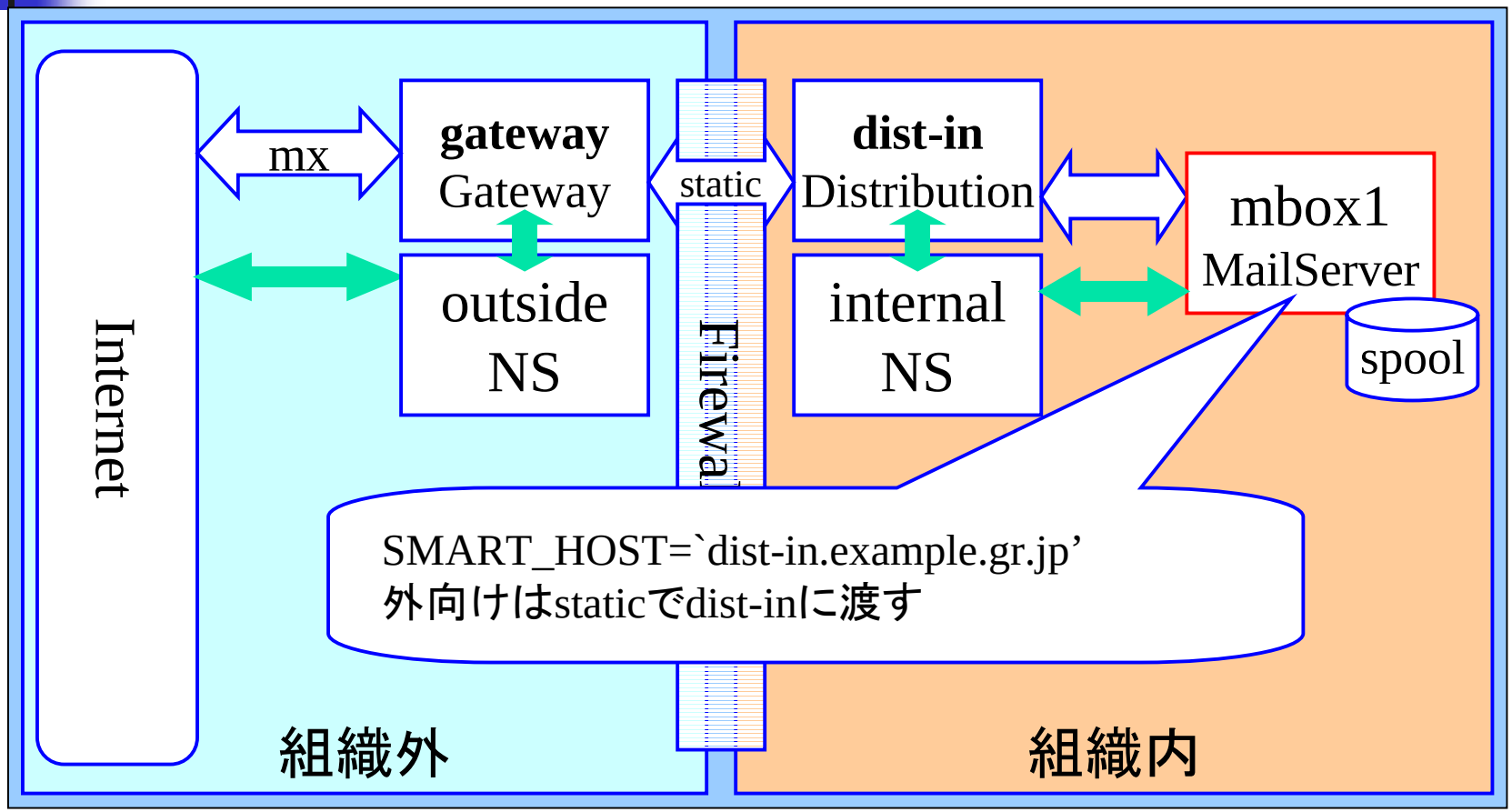
とするとそのドメイン以下全部のマシン名をマスカレード

```
cd ${SENDMAIL_SRC}/cf/cf
make config.cf
make install-cf CF=config
```

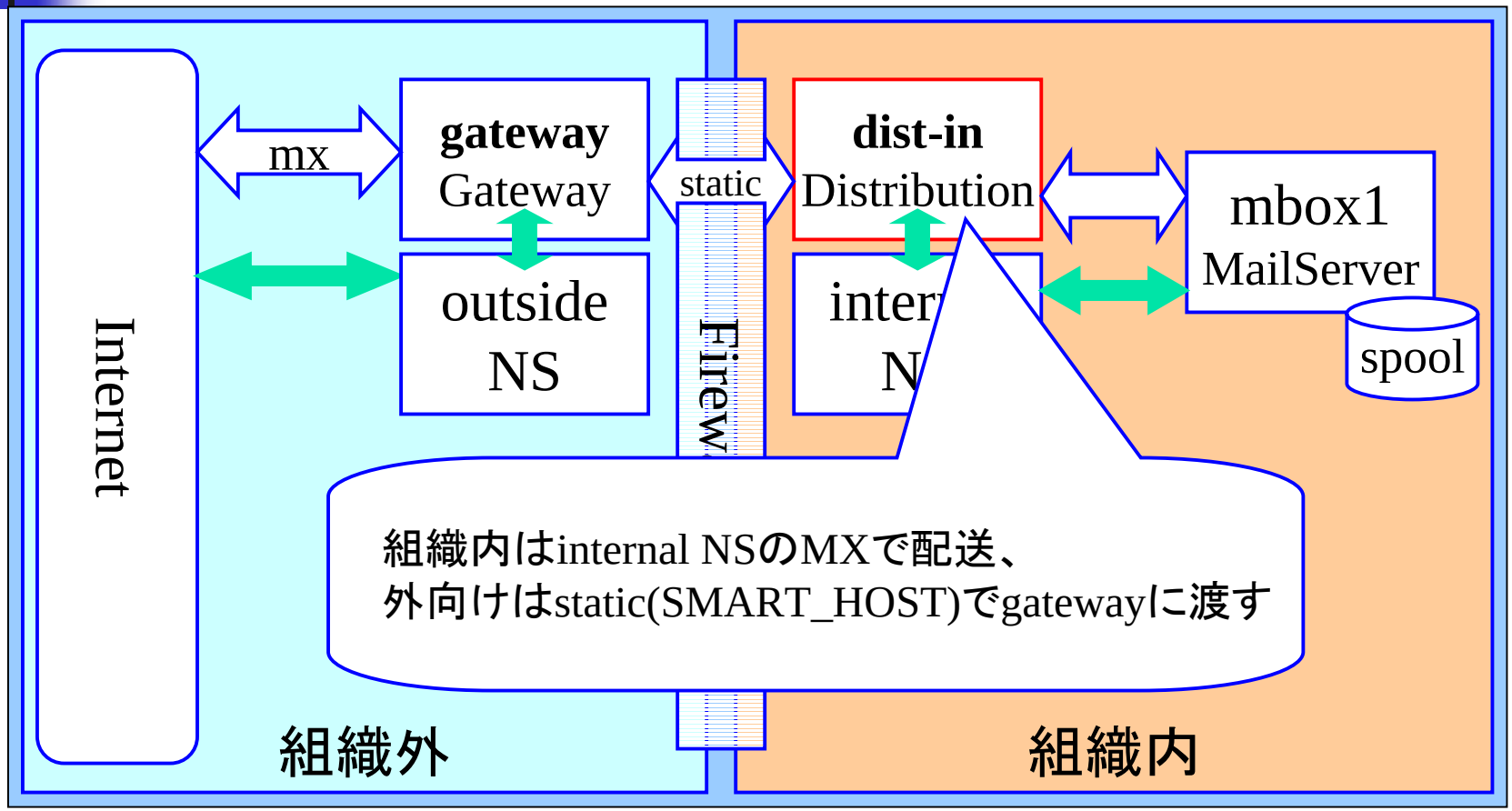
配送モデル（社内DNS利用）



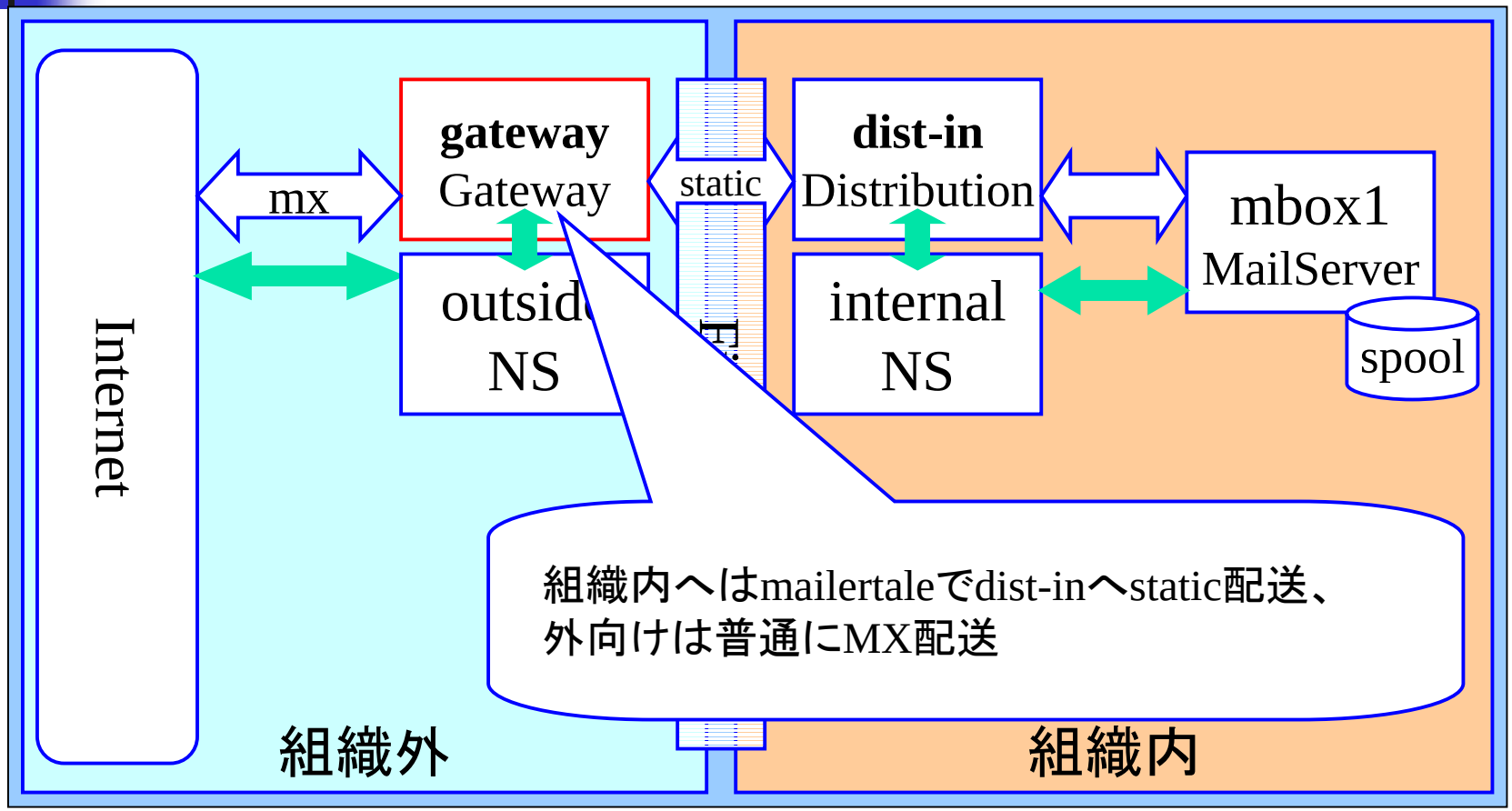
内部MSの設定

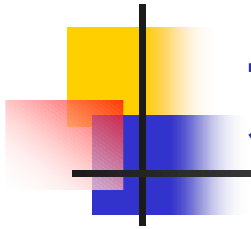


内側distributionサーバの設定



外側サーバの設定

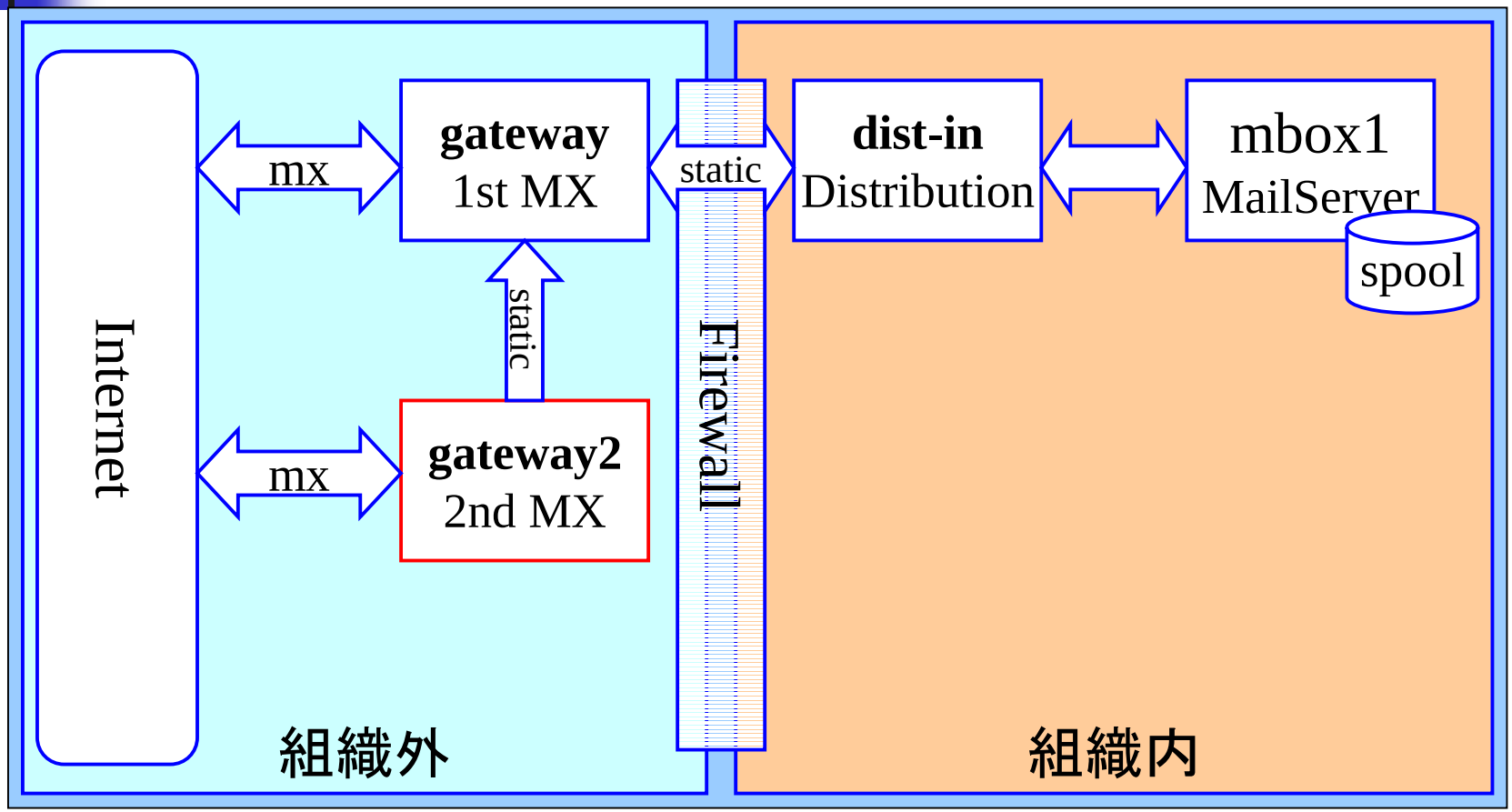


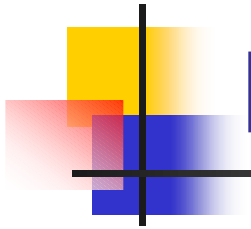


対外受信ホストの多重化

- MXを複数にする理由
  - メールが集中して負荷が高い場合
    - 一時的にため込む
    - 可能なら2nd MXは1st MXとは独立に配信
  - メンテナンス用
    - 片方が停止しても受け取りに支障を出さない

2nd MXのある配送モデル

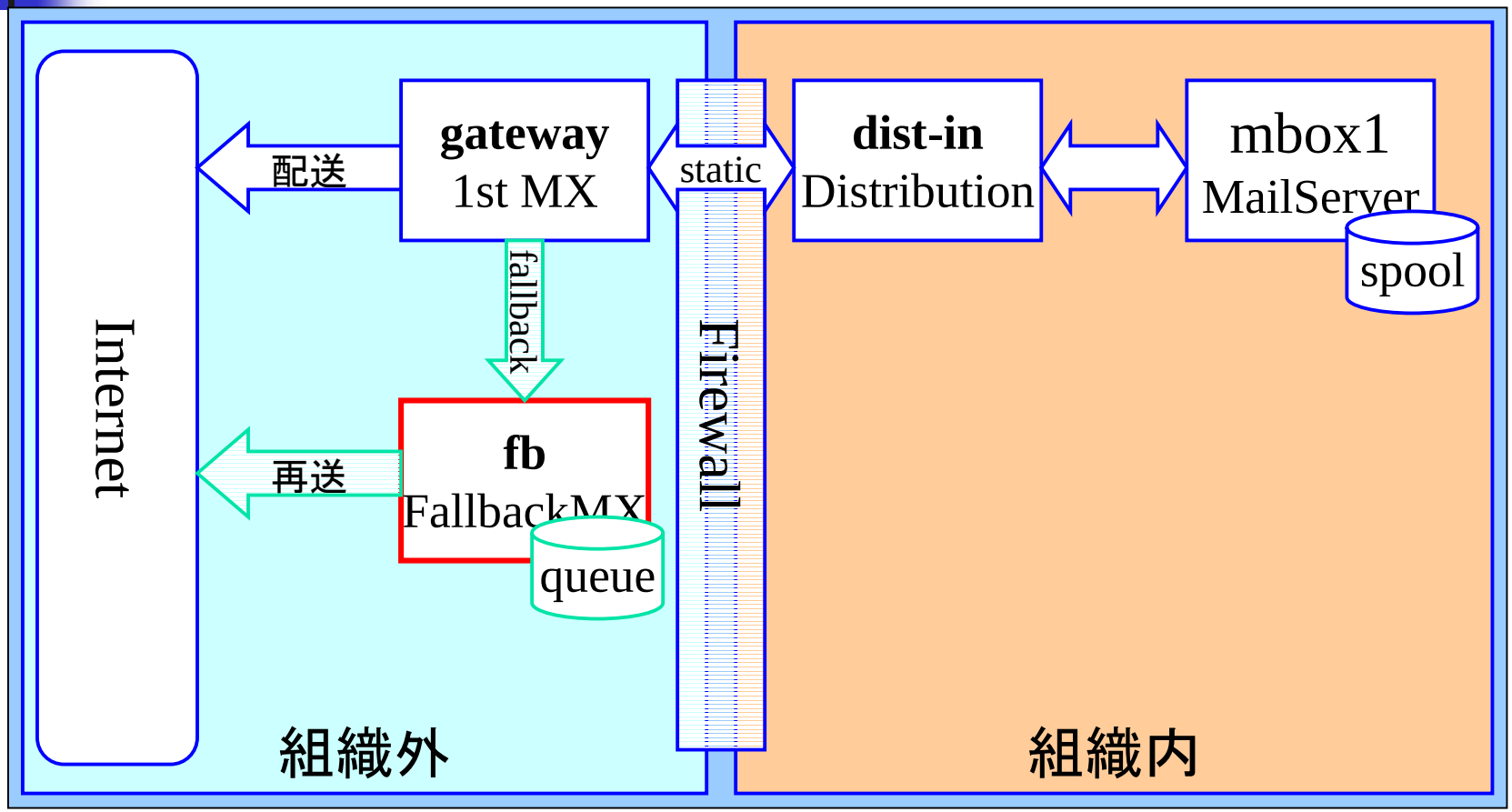


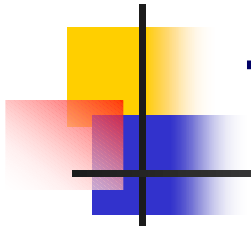


FallbackMX

- 再送専用ホスト
  - 再送queueを特定のホストに集める
    - DNSが引けなかった場合
    - 全MXに対してメールが送れなかった場合
  - ネットワーク的なトラブルがすぐわかる
  - 再送を試みる期間の調整
  - `define(`confFALLBACK_MX',`fb.example.gr.jp')dnl`

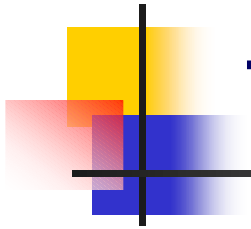
FallbackMXのある配送モデル





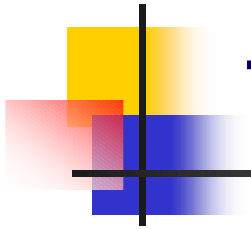
設定の勘所（1）

- まずはstatic配送＝mailertable
  - sendmailは配送先判定で真っ先にmailertableを見る
  - 全丸投げstatic＝LOCALRELAY
    - spool（localuser）がいない→LOCALRELAY
- ドラえもんstatic＝SMART_HOST
  - localとstaticで配送先がわからなかったら
    →SMART_HOST
- SmartHost設定がない＝MX配送
  - MXもAも引けない→エラー



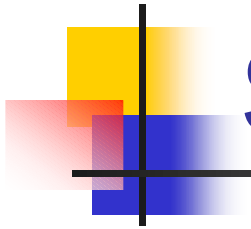
設定の勘所(2)

- spam対策はaccess_dbに集約
  - 身内のrelay配送の可否
  - spamlist的設定
  - blacklist-recipient設定
- DNSBLももちろん使える
  - デフォルトはMAPS RBL
  - もちろん他も指定可能



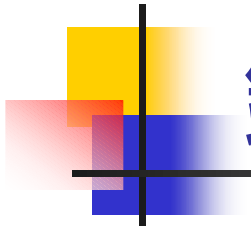
設定の勘所(3)

- ドメイン名の書き換え
  - MASQUERADE_ASで書き換え後のドメイン指定
- 範囲指定
  - FEATURE(`limited_masquerade‘)だと
    - MASQUERADE_DOMAINに指定したドメインだけ書き換え
  - FEATURE(`masquerade_entire_domain’)だと
    - MASQUERADE_DOMAINに指定したドメイン以下の全ドメインを書き換え



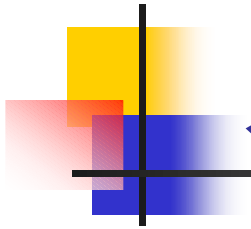
SMTP/TLSの利用

- TLS（Transport Layer Security）
  - 乱暴に言うと、SSL接続への移行を視野に入れた接続の枠組みのこと
  - サーバ間SMTPを経路暗号化
  - sendmailでもこのTLSの枠組みを用いてSMTPの接続をSSLへ移行することが可能
    - OpenSSLの利用が前提
    - 商用版では使えるようになっている製品もある



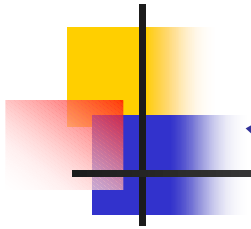
鍵の準備

- TLS(SSL)には認証用の鍵が必要
  - CA(認証局)から購入
    - 他社からの接続でもTLS利用が可能に
  - 自前で準備
    - 鍵の配布範囲にTLSの利用が限定される



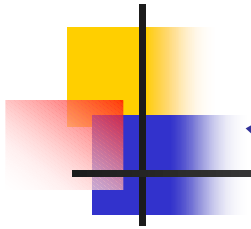
メール経由のウイルス（1）

- 添付ファイルが感染源であることが多い
  - マクロウイルス（Excel、Word、PowerPoint）
    - 中に忍ばせてあるOfficeオブジェクトが曲者
  - 実行形式ファイル
    - 不用意に実行してはいけない
    - Happy99、SirCam、Nimda、Aliz、Badtrans、Bugbear、Sobig.F、Swen



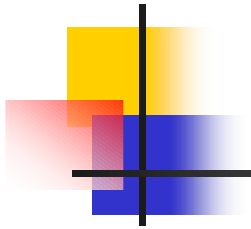
メール経由のウイルス（２）

- 自動的に実行されてしまう添付ファイル
  - .wav (nimda) とか.pif(Sircam)とか.scr(bugbear)とか
- 感染スピードの爆発的上昇
  - メール、HTTP、JavaScript、ファイル共有など複数経路で感染するワームの登場
  - 市販のウイルス対策プログラムのupdateが追いつかず、防ぎきれない例も多発
  - ウイルス除去プログラムが影響を除去し切れない例もある模様。
  - こまめにWindows updateを!



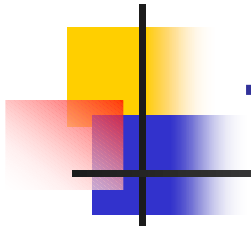
メール経由のウイルス（3）

- 添付ファイル
  - 元凶はMIME-multipart（便利さの代償?）
    - 入れ子構造でファイルを添付できる
      - 2段目にファイルを添付した後の1段目にウイルス添付（nimda）
      - たまにデリミタの使い方を間違っているワームもある（Sircam）
      - 使われるContent-Typeも多様化している
  - 無限段まで入れ子をチェック
    - DoS対象になってしまうかも....



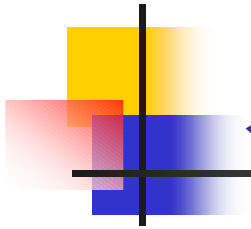
ウイルス・ワーム対策体制の例

- ウイルス対策プログラムを過信しない
  - ウイルスの感染の方が速い場合がある
- できるだけ速い情報の収集
  - ワームによるアクセスを監視（WWWサーバやIDSで）
  - 感染経路情報を示して警戒呼びかけ
    - なにもやらないのと比較して格段の防御になる
- 大量感染源になり得る部分での対策
  - メーリングリスト・ドライバで添付ファイルの拡張子チェック＋削除（メーリングリストでの添付ファイル使用の禁止）
  - Windowsのsecurity-updateに常に注意を払う



チェインメール

- 善意の協力依頼を装う(あるいは本物)
  - 「このメールを転載して下さい」が曲者
    - 無制限の転載を意図している場合には無視
    - 本来の目的を達成するには、期間や範囲を限定して一定数しか転載されない工夫を
- 不幸・幸福のメール
  - 「このメールを5人に転送しないと．．．」
    - 初心者の多い環境で流行りやすい

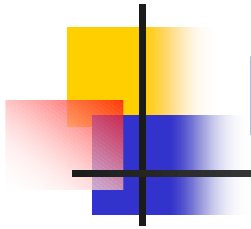


メール爆撃（Bombing）

- ２種類ある
  - 巨大なサイズのメールを送付
  - 膨大な数のメールを送付
    - どちらもspoolを膨らませる結果になる
    - loopと見分けがつきにくい場合がある
- サイズ制限、通数制限等の防御
  - メーリングリストではさらに深刻な問題に
  - O MaxMessageSize＝500000

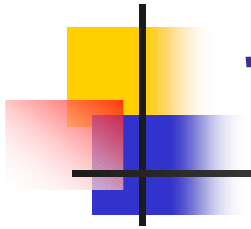
知っておくべきメールアドレス

- MAILBOX NAMES FOR COMMON SERVICES, ROLES AND FUNCTIONS(RFC2142)で挙げられているもの
- 例えば、
  - abuse@example.gr.jp
    - いざという場合の問い合わせ先
  - postmaster@example.gr.jp
    - メール配送についての問い合わせ先
  - hostmaster@example.gr.jp
    - DNSについての問い合わせ先



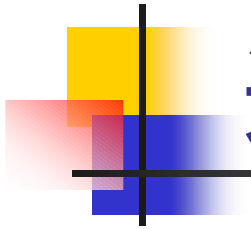
MLの周辺アドレス

- 周辺アドレスの例
  - owner-hoe@example.gr.jp
    - sendmail的にちょっと考慮されたMLの発信者アドレス
  - hoe-admin@example.gr.jp
    - 管理者のaliasとして使われることがある
  - hoe-request@example.gr.jp
    - RFC2142的管理者アドレス
  - hoe-errorsto@example.gr.jp
    - エラーメールの専用受信アドレスを用意している場合



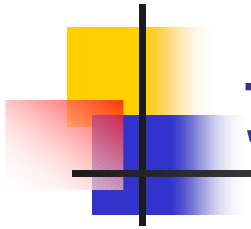
アドレス詐称・隠蔽問題

- bombing等では発信者アドレスが偽装される
  - SPAM発信者を偽装して発信者をbombing
- MLに他人のアドレスを登録する
  - 自動登録でConfirmなしだとアウト
- 無料メールアドレスの転送機能
  - 誰に届くかわからないという意味で曲者



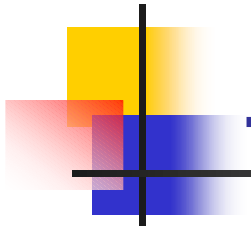
運用上の留意点

- spam対策
  - 「来たときの対策」と「出させない対策」
    - SMTP Authentication（RFC2554）
    - Message Submission（RFC2476）
    - SMTP over TLS（RFC2487）
    - RBL/SBL
    - Bayesian filter
    - URL filter
  - メーリングリストではアドレス一覧を出さないこと
    - 例えばPPMLは一般参加者のwhoコマンドに対してGECOSの一覧を出す



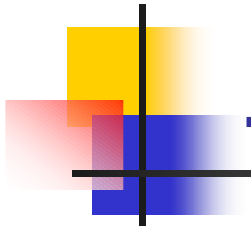
最近の傾向

- 大規模化に伴う相対的な管理レベルの低下
  - ISP等では大規模化する一方
    - ユーザ管理の省力化を目的にディレクトリサーバを利用するケースも珍しくなくなってきている
  - 携帯電話メールのトラフィックの増加
    - 容量は小さいが通数はものすごい
  - MIME-multipartによる添付文書
    - 容量が大きいのでspool容量の再考が必要なケースも



エラーメールの基礎

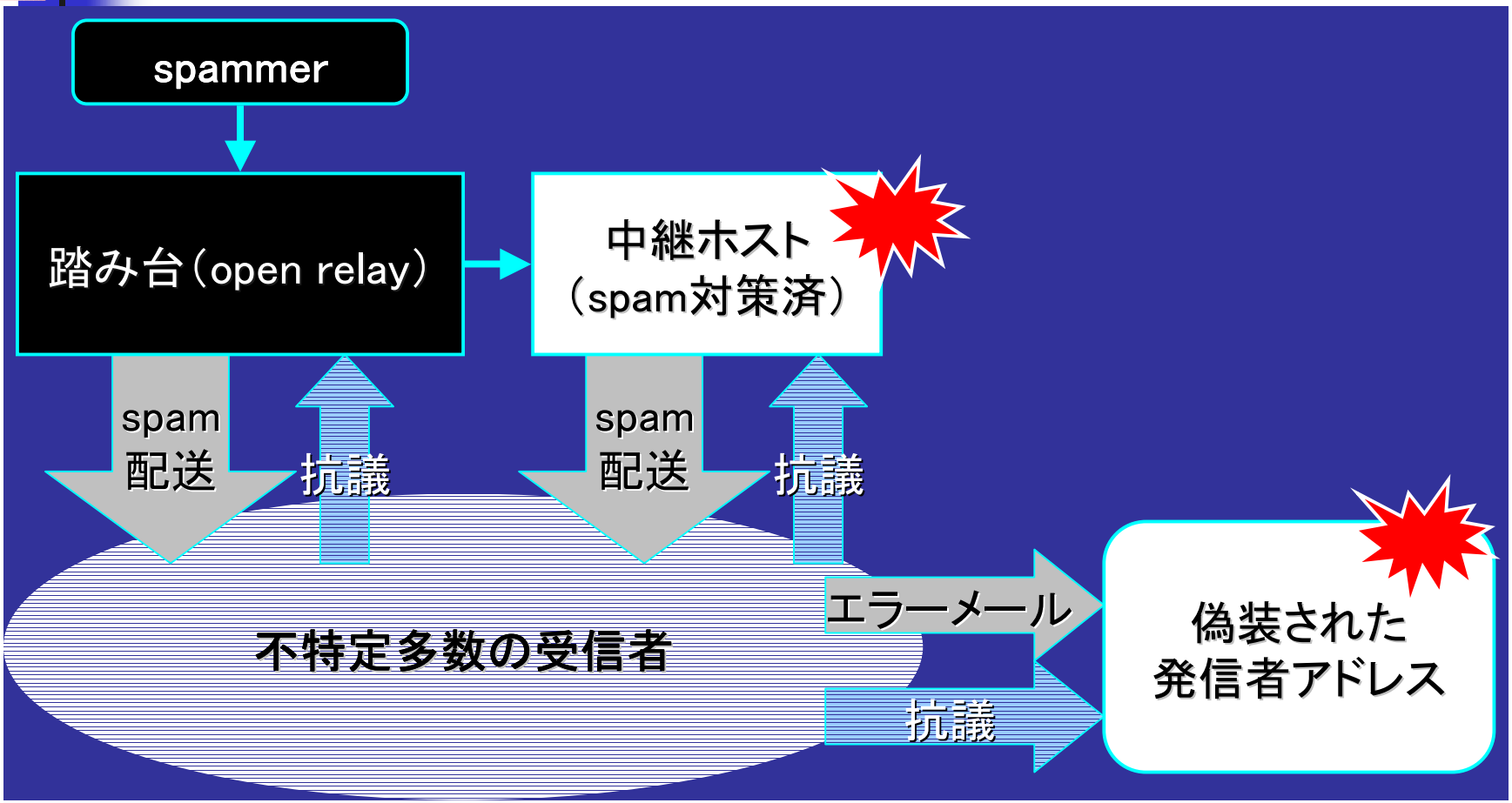
- エラーメール配信の枠組み
  - DSN（Delivery Status Notification)
    - Envelope From は null address（ <> ）
      - エラーメールに返信アドレスはない
- トラブルの種類を判定する手段
  - RFC1893（Status Code）
    - Status: 5.1.1
      - 5.X.X　Permanent Failure
      - X.1.1　Bad destination mailbox address

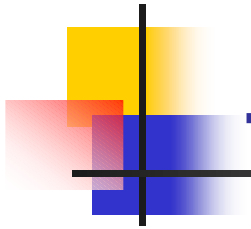


エラーハンドリング問題

- 配送エラーコード（status code）の実装
  - 実際にRFCを守っているか？
    - sendmailやPostfix、SIMS等守っているものも多い
    - その他の対応はいまいち
      - MTAの数だけエラーハンドリングのプログラムが必要
      - 標準を守ろうとしないMTAは大迷惑なんだけど…
    - 大量にメールを配るところでは頭痛のタネ
    - 最近はウイルス通知メールの嵐
      - エラーメールの通知形式に準拠してくれないかなぁ….

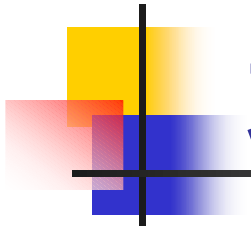
spam中継の被害の構図





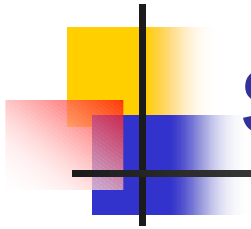
エラーメールによるRDDoS

- envelope-fromを詐称されてしまった場合
- 非常に多数のサイトから大量のエラーメール
- メインの1stMXが潰れそうになったら、1stMXをDNSから削除、TTLの短い2ndMXのみにする
  - RDDoSのエラーメールはDNSを新たに引いて2ndMXへ
  - 普段から良くメールの来る相手はDNSのcacheがあるので1stMXにメールが来る
  - DNSのcacheの生存時間を利用したエラーメールの振り分け
    - 岡山大学の山井先生の考えられた手法です（JANOG12）
      - RDDoS＝Reflected Distributed Denial of Service



大量のDoubleBounce

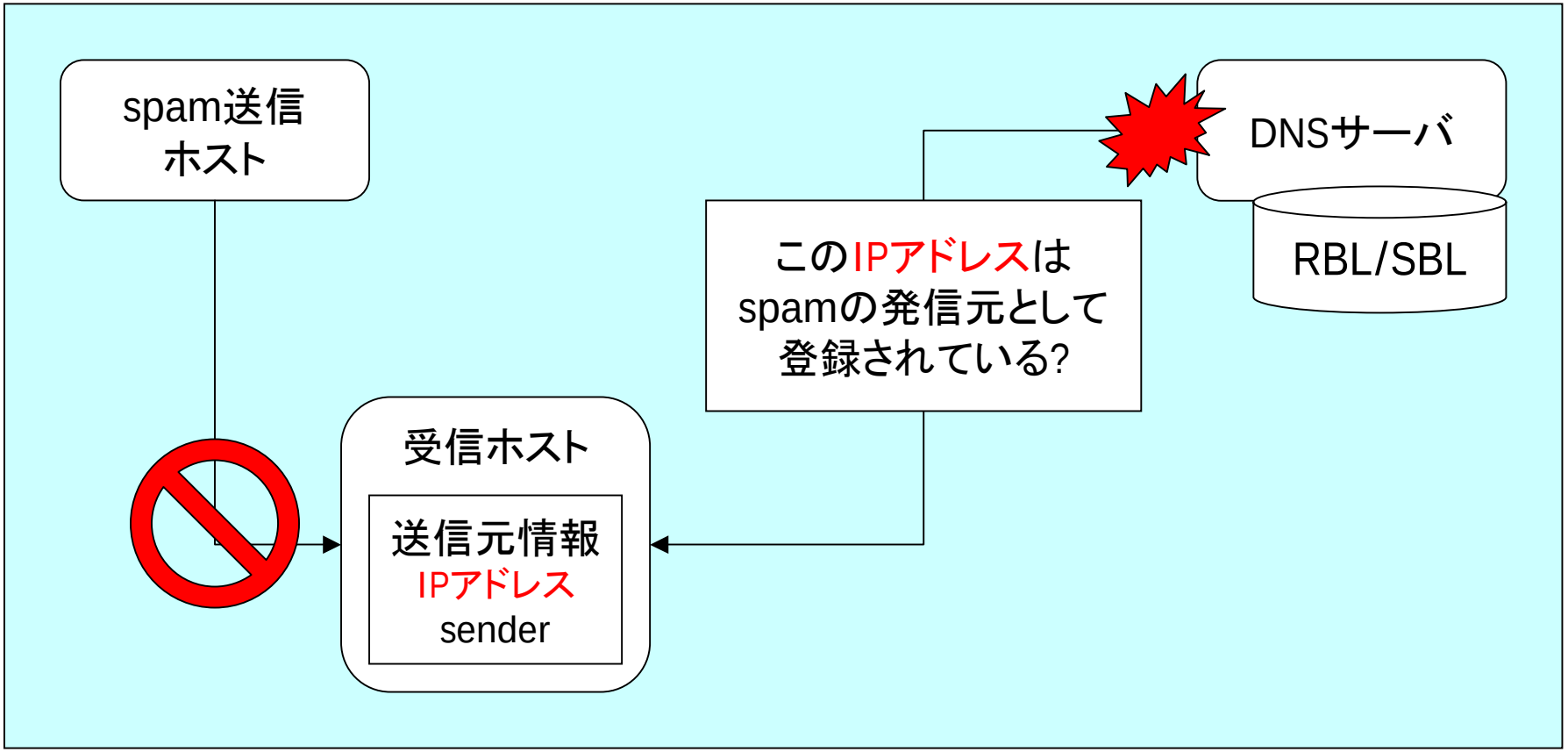
- エラーメールの配送エラー
  - 通常、エラーメールのエラーは消失する
    - エラーメールのsenderはnull-address
    - 例外がDoubleBounceの機能
    - DefaultではPostmaster宛になっている
  - envelope-fromの詐称による絨毯爆撃型spamの副作用として発生
  - DoubleBounceをOFFにする
    - ログをチェックすることが条件

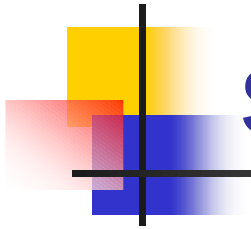


spam対策（1）

- RBL（Realtime Blackhole List）
- SBL（Spam Blocking List）
  - spamの発信元を登録する闇魔帳
    - DNSと同じ枠組みで作られている
      - MTAがメール送信元のIPアドレスを照会
    - 残念ながら訴訟対策のためかどんどん有料化
      - ORDBでも寄付を募っている
    - 自分のサーバが登録された場合
      - メールを受け取らない所が出てくる

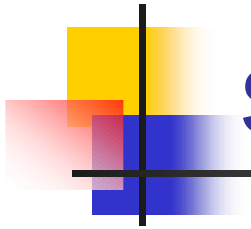
spam対策(1)





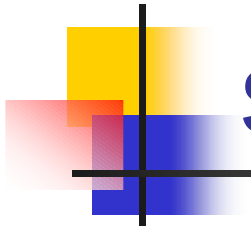
spam対策(2)

- SPAMLIST(access_db)
  - 発信元についていずれかを指定して排除
    - メールアドレス(envelope from)
    - ドメイン
    - IPアドレス
- POP before SMTP
  - ISPで取り入れられている手法
    - POPアクセスの発信元に対してSMTP接続を許可する
    - 例えばqpopperにパッチを当てて実現する



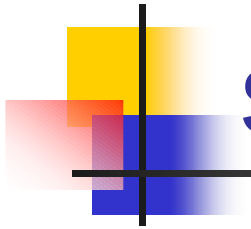
spam対策(3)

- ベイズ推定を用いたフィルタ
  - 狙いはspamに登場する語句の出現傾向
    - 語句の出現傾向からspamかどうかを判定する
    - 辞書が比較的大きくなる
    - 言語依存(現状で英語、日本語くらいならOK)
  - 弱い相手
    - 画像1枚、リンク1つだけのspam
    - 大量の一般的な文書に埋め込まれた広告
    - あの手この手の偽装手段



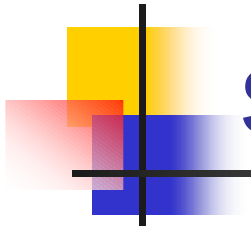
spam対策(4)

- パターンマッチ
  - 例えば正規表現でパターンを指定
    - 個人で使ってもあまり効果はない
    - サーバで使用すると効果的
    - 誤判定リスクはパターン次第
    - 言語への依存性は実装次第



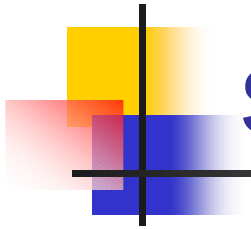
spam対策(5)

- ヒューリスティック・フィルタ
  - 各部のパターンを抽出して確率で引っ掛ける
    - Fromヘッダの特徴
    - Subjectの特徴
    - Toの特徴
    - Receivedの特徴
    - Content-Typeの特徴
    - ．．．と積み上げて判定する手法



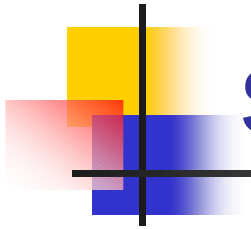
spam対策(6)

- URLをベースにしたspam排除
  - URLのパターンマッチ的な手法はよくある
    - 誤判定リスクは排除すべきURLの確認に依存
    - userinfoとquery部分を宛先ごとに改変している例
    - 言語依存性なし



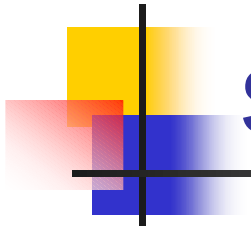
spam対策(7)

- デジタルシグネチャ(d-sig)のDB化
  - spamの各パートのd-sigを検知する
    - MIME multipart解析
    - d-sigが一致する(同一の内容の)partがあればspamと判定する
    - spamの内容も(ランダム文字列等で)その都度改変されるので、データの共有と更新が効果を上げる鍵になろう



spamの現状（1）

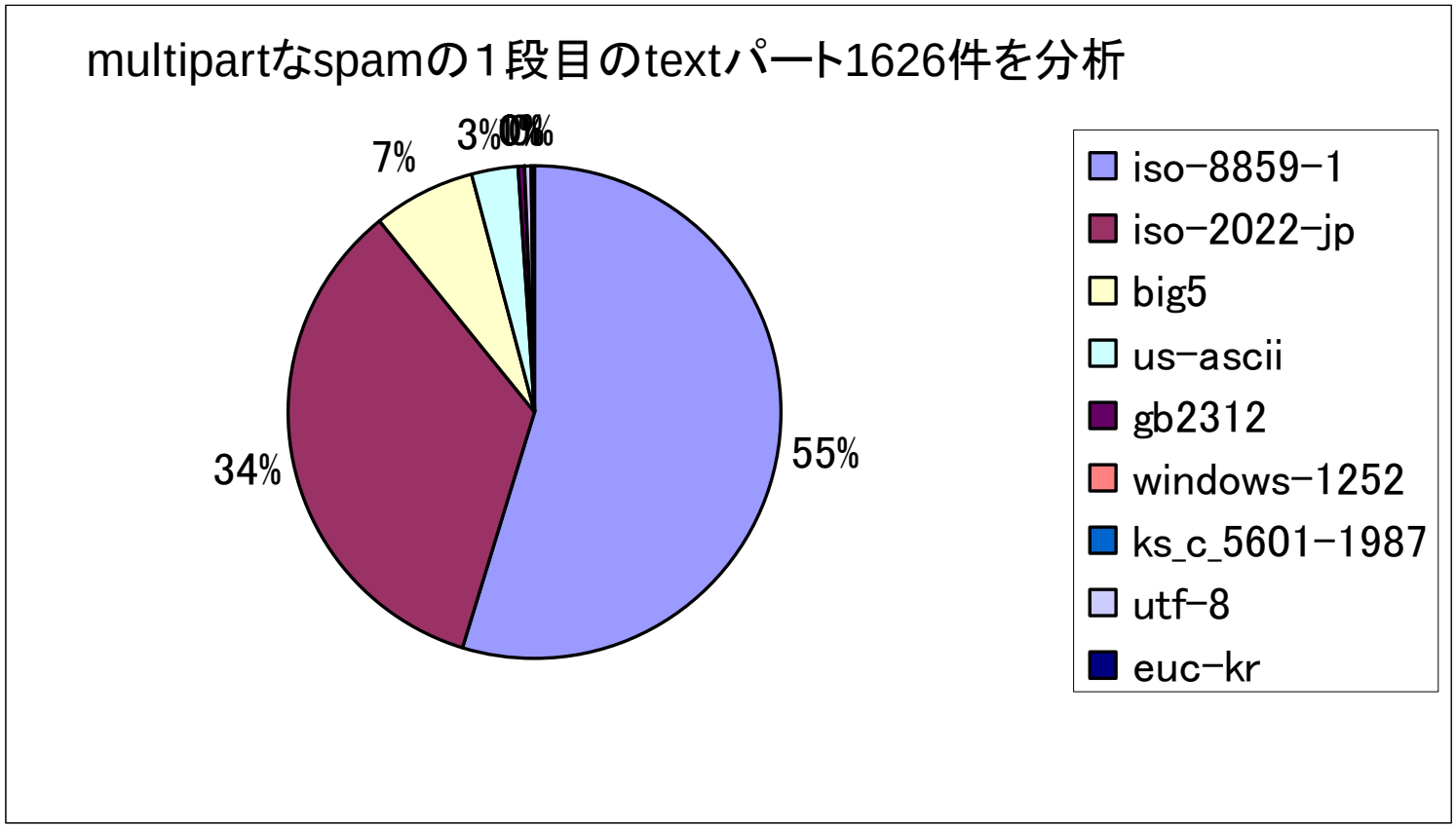
- RBLに存在する壁
  - Dial-upアカウントからのゲリラ的spam発信
    - DHCPで変わるアドレスをRBL登録?
      - 無実の人間がメールを受け取ってもらえない
    - アドレスブロックごとRBL登録?
      - メールを受け取ってもらえない人が大量発生
  - RBLがDoSアタックされる事件も発生
    - RBLへの到達性をいかに保障するか？

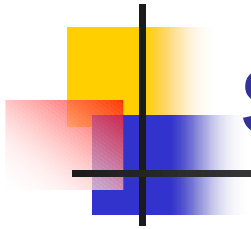


spamの現状（2）

- spamlistに存在する壁
  - 発信者のメールアドレス（envelope from）
    - 詐称可能
  - ドメイン
    - DNS逆引きチェックはできる．．．が
    - DNS逆引きも詐称する例がある
  - IPアドレス
    - Dial-upアドレス複数から一斉に送信

spamに使用されるcharset

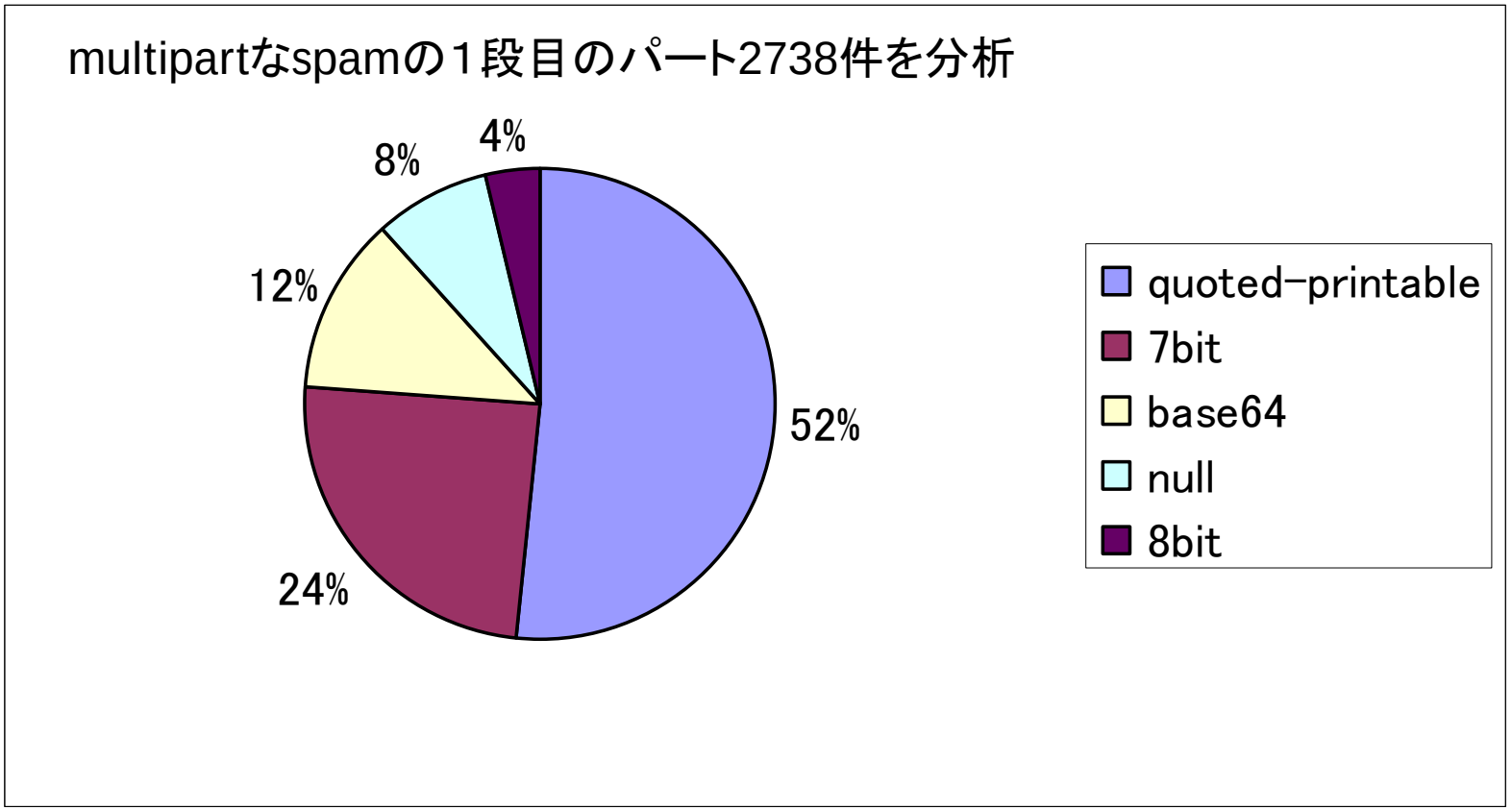


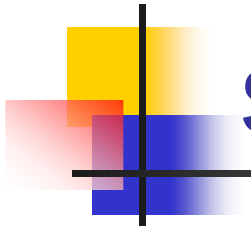


spamに使用されるcharset

- ISO-8859-1が多い
  - 8bitなのでquoted-printableとの組み合わせは自然
  - ISO-2022-JPとUS-ASCIIでは対応が不十分?
  - big5やgb2312（中文）も無視できない
    - 個人宛のspamの集計なので比率に偏りはあるかも
  - 頑張れベイジアンフィルター
    - WWWブラウザ以上に言語と文字への対応が必要
    - 英語への対応だけで95%とかいう認識率になるのはある意味特殊なspam環境下での話か?

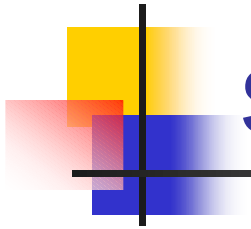
spamに使用されるencode





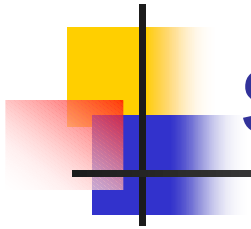
spamに使用されるencode

- quoted-printableが多い
  - Soft-line-breakを利用した単語の分断
    - bayesian filter回避？
  - URL中の「＝」はquoted-printableでは「＝3D」になる
    - URL filterの回避？
- 割合としてbase64も無視できないレベル
  - 何が書いてあるかdecodeしてみるまでわからない



spamの現状（3）

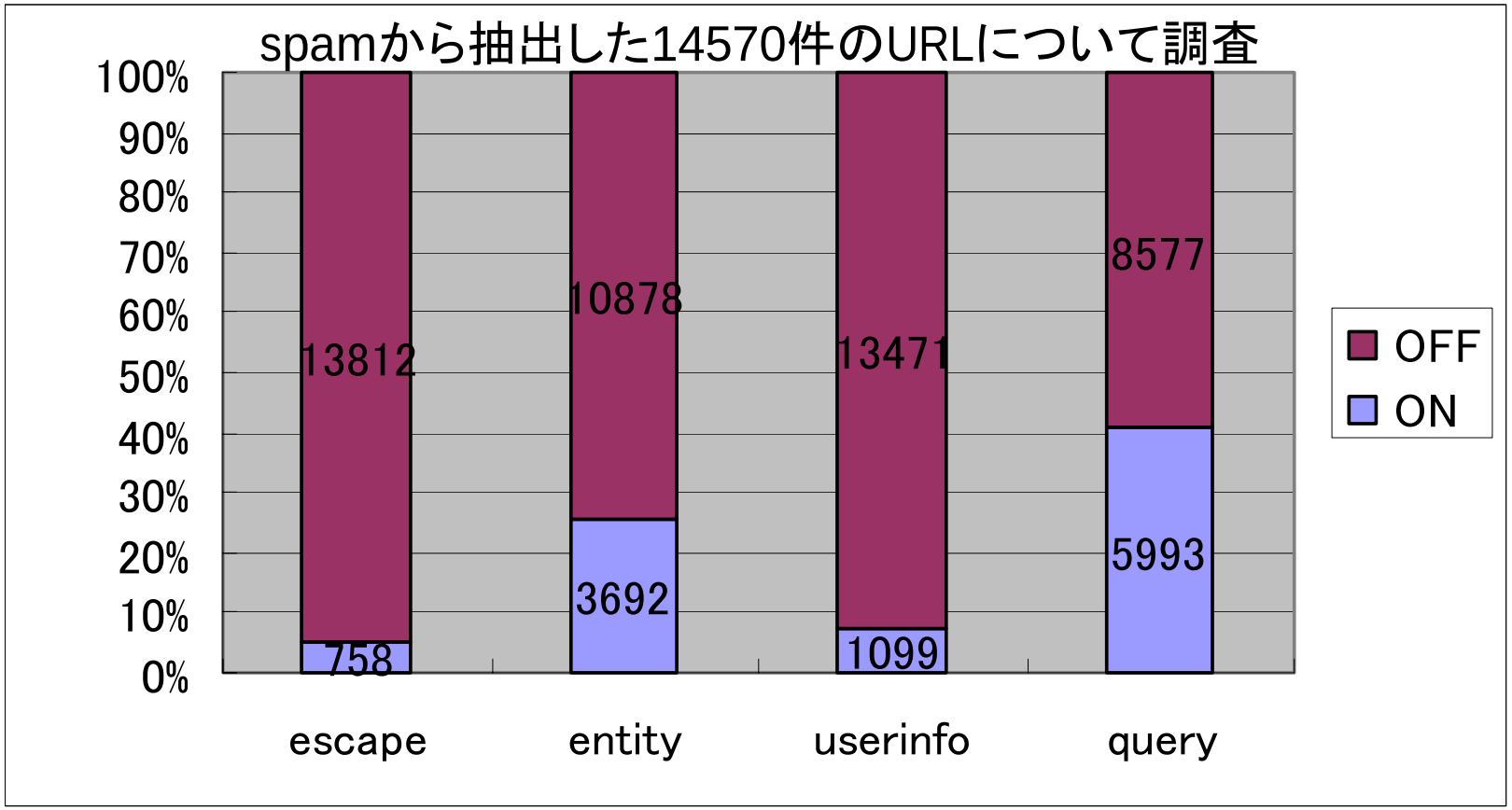
- 本文部分へのフィルタに対する壁
  - <!-- -->
    - HTMLのコメントを用いた語句の分断
  - w
    - 数値実体参照を用いた文字の隠蔽
  - =（改行）
    - quoted-printableのSoft-line-breakを用いた語句の分断
  - base64を用いたパート全体の隠蔽
    - フィルタを使う前の処理が肝心
  - さらに言語依存の問題がある
    - 例えば、POPfileは英語と日本語は大丈夫のようだ

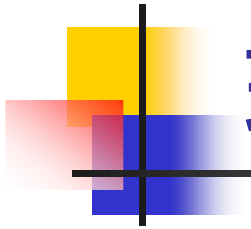


spamの現状（4）

- URLベースのフィルタに対する壁
  - 「%77」とか「&#１１９;」とか
    - エスケープを利用した文字の隠蔽
    - 数値実体参照（entity）を用いた文字の隠蔽
  - http://user@host:port/path?query
    - user部分とquery部分が可変でやってくる

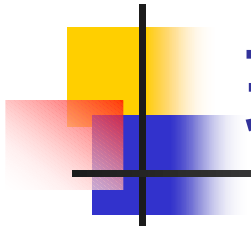
URLの改変可能要素





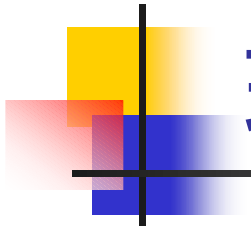
現状への対処(1)

- 発信元情報によるspam排除
  - RBLまでの到達信頼性の確保
    - 例えば複数のRBLを併用するなど
- メールの内容によるspam排除
  - 最低でもMIME-multipart解析が必要
    - partごとにContent-Transfer-Encoding:が指定可
      - quoted-printableとbase64への対応
    - HTML特有の事情への対応
      - <!-- -->　とか実体参照とか



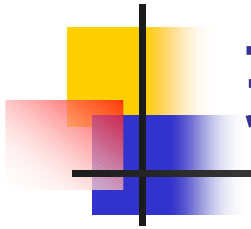
現状への対処(2)

- SpamAssassinに見る手法
  - PerlのModuleとして実装されている
  - ヘッダ/テキスト解析
    - rule-baseを使用
    - MIMEDefang等を併用することでMIME-multipartに対応
    - Bayesian filter
  - RBL/SBL
    - 併用可能
  - spam signature
    - 先行技術～Vupil's Razor(2001年5月公開)
    - パートごとに2種類のd-sigを計算、ネットワークで共有する



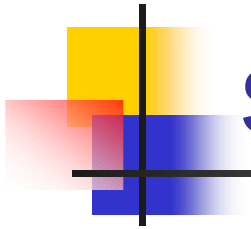
現状への対処(3)

- POPFileに見る手法
  - メール振り分けツール(ベイズ推定)
    - Perlで書かれている
    - POPのproxyとして動作する
    - MIME-multipart解析もやっている
    - 実体参照の復号もやっている
    - 日本語にも対応した(Kakasi)



現状への対処(4)

- bsfilterに見る手法
  - spamフィルタ(ベイズ推定)
    - Rubyで書かれている
    - POPやIMAPのproxyとしても動作する
    - MIME-multipart解析もやっている
    - 実体参照の復号はやってないかも
    - 日本語にも対応(Kakasi)

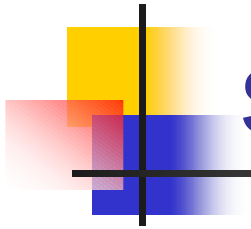


spam絶滅作戦(1)

- フィルタは何種類あっても良い
  - ベイジアンフィルタはユーザに近い所で普及
  - サーバ側にあって欲しいフィルタは?
    - 他のフィルタの弱点を補完するようなフィルタ
      - 軽いこと
      - 言語依存性がないこと
      - 巻き添えでメール送信不能になったりしないこと
    - 原理的に誤認識がないこと(重要)

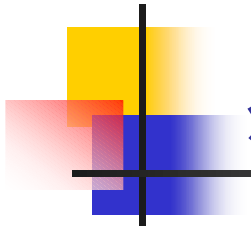
spam絶滅作戦（2）

	bogofilter	SpamAsassin	POPFile	bsfilter	BrightMail	SPAMBlock	PICKY	Outlook2003	Eudora 6	Netscape7.1
種別	フィルタ	フィルタ	フィルタ	フィルタ	フィルタ	フィルタ	フィルタ	MUA	MUA	MUA
ライセンス	GPL	GPL/商用	GPL	GPL	商用	商用	未定	商用	商用	無料/商用
spamlist	×	×	×	×	×	○	×	×	×	×
RBL/SBL	×	△	×	×	◎	×	×	×	×	×
pattern	×	○	×	×	○	◎	×	○	○	○
MIME解析	×	×	○	○	?	?	○	×	×	×
Signature	×	○	×	×	○	?	○	×	×	×
bayesian	◎	◎	◎	◎	×	?	×	◎（学習済）	◎	◎
→日本語	△	△	○	○	×	?	×	?	?	?
→その他の言語	?	?	○	?	×	?	×	?	?	?
URL	×	○	×	×	○	○	◎	×	×	×
→偽装解除	×	×	×	×	○	?	○	×	×	×
開発言語	C	Perl	Perl	Ruby	N/A	N/A	C	N/A	N/A	C
リソース	中	高	高	高	低	低	低	中	中	中
誤認識	あり	あり	あり	あり	あり	なし	なし	あり	あり	あり



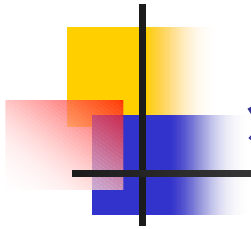
spam絶滅作戦(3)

- 複数の種類のフィルタで防護
  - MUAにbayesian-filterが装備されつつある
    - 最終的な振り分けはここに任せるとして
  - bayesian-filterが複数段あっても無意味
    - 複数あっても同じspamを見逃す可能性がある
  - 各段階で対策を
    - メールサーバ、POP/IMAPサーバ、MUA
  - 認識率95%のフィルタが2段、3段あると. . .
    - (1-(0.05)^2)*100 = 99.75%
    - (1-(0.05)^3)*100 = 99.9875%



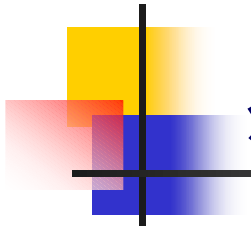
懸案事項(1)

- メールが媒介するウイルスの多様化
  - 次から次へ新種
  - ちょっと昔のやつも根強く繁殖
  - ウイルス対策ソフトが売れる理由
    - 大量繁殖を防ぐにはMTAかMLドライバのレベルで添付ファイルのチェックを
- MailとWWW、死守すべきはどっち?



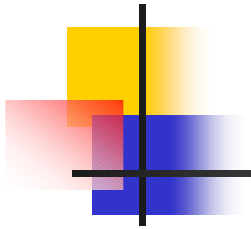
懸案事項(2)

- 大規模サイトのサーバの受信能力不足
  - 再送が再送を呼んで昼間は常に輻輳しているとしか思えないサイトもある
  - ある程度のメール流量のあるメールゲートウェイやFallbackMXサーバの残存queueの観察でいらないことがいろいろわかってしまう
    - いやでもわかってしまう



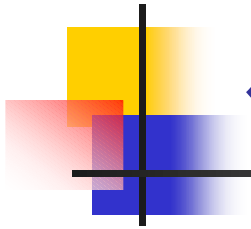
懸案事項(3)

- 管理者不足
  - MTAを設定できる人間が不足?
    - 真っ先にspam対策で困るケースが多発
    - Postfixなら、qmailなら、sendmailなら．．．
      - 混乱するだけなのでどれか1つ完璧に把握してから他のMTAに言及して欲しいなぁ．．．
  - 付属のcfでも設定する項目はそれほど多くない
    - READMEが英語なのが障壁?
      - 日本語訳したcf/READMEくらいならネットワーク上にはある
    - 凝ったことをしようと思ったらコウモリ本もある



匿名掲示板に学ぶこと

- 個人メールをWWW上に公開する?
  - プライバシーの侵害に注意すること
    - 誹謗中傷の場合、伏せ字でも個人が特定できるとアウトです
  - 著作人格権の侵害にも注意すること
    - 誹謗中傷を付けて公開したりするとアウトです
  - メール内容の証拠能力は補助的なもの
    - いくらでも書き換え・なりすましが可能
    - 電子署名の利用で少し改善されるかも
  - 公開は気軽にできるが、法的な意味は意外に重大
    - 注意(訴訟対策?)が必要です



付録（社内ホスト設定例）

```
VERSIONID(`$Id: config.mc,v 1.5 2003/12/03 11:00:11 ando Exp ando $')
OSTYPE(bsd4.4)dnl
DOMAIN(generic)dnl
MASQUERADE_AS(`example.gr.jp')dnl
MASQUERADE_DOMAIN(`iw2003.example.gr.jp')dnl
FEATURE(`limited_masquerade')dnl
FEATURE(`masquerade_envelope')dnl
EXPOSED_USER(`root postmaster')dnl
FEATURE(`mailertable')dnl
FEATURE(`nocanonify')dnl
FEATURE(`access_db')dnl
FEATURE(`blacklist_recipients')dnl
FEATURE(`accept_unresolvable_domains')dnl
MAILER(local)dnl
MAILER(smtp)dnl
Dmexample.gr.jp
Dwiw2002
define(`confDOMAIN_NAME',`$w.$m')dnl
define(`SMART_HOST', `esmtp:[192.168.0.3]')dnl
define(`confCF_VERSION', `IW2003')dnl
define(`confTO_IDENT',`0s')dnl
```