ファイアウォールの基礎と構築

Internet Week 2000チュートリアル

T2

二木 真明（住友商事）

このコースの目標

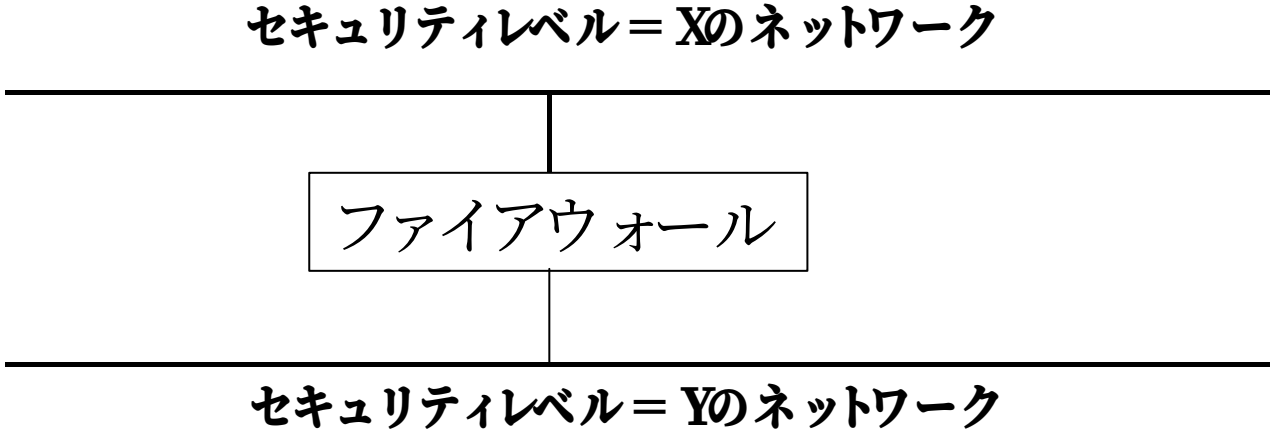
- ファイアウォールの「概念」を理解する。
- ファイアウォール構築のための手法、技術を理解する。
- ファイアウォールにできること、できないことを理解する。
- セキュリティ全般の中での位置づけを理解する。

ファイアウォールという概念について

- ファイアウォールは 概念」
  - 様々な製品を使って 構築」するもの
  - ルータ、コンピュータ・・様々な製品で構築が可能
  - ファイアウォール製品は、構築の 道具」
  - 形よりもポリシーが重要

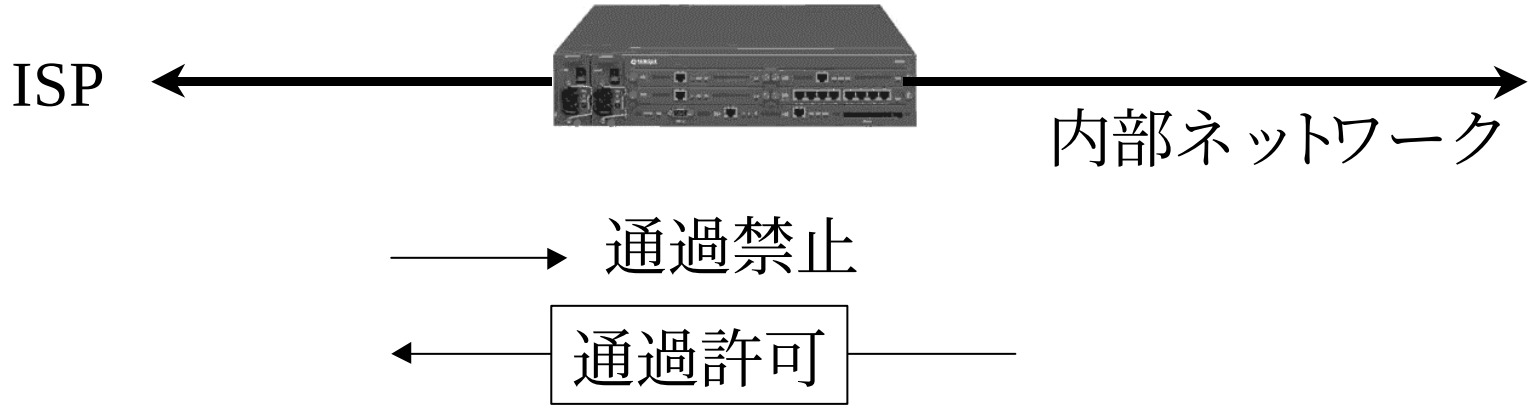
ファイアウォールはなにをするか

- ネットワークの関所
  - セキュリティレベルの異なるネットワークを接続する際、通信の通過を監視・制御することで、双方のレベルを保つ機能を提供する。



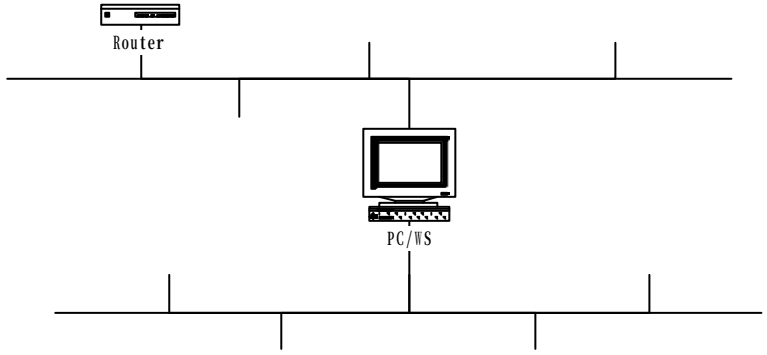
ファイアウォールの形

- ルータによる通過制御（パケットフィルタ）



ファイアウォールの形

- Dual（Multi）home host
  - 複数のネットワークに接続されたコンピュータ
  - パケットフィルタ、PROXYなどを実装
  - ファイアウォール専用ソフトウエア



ファイアウォールの仕事

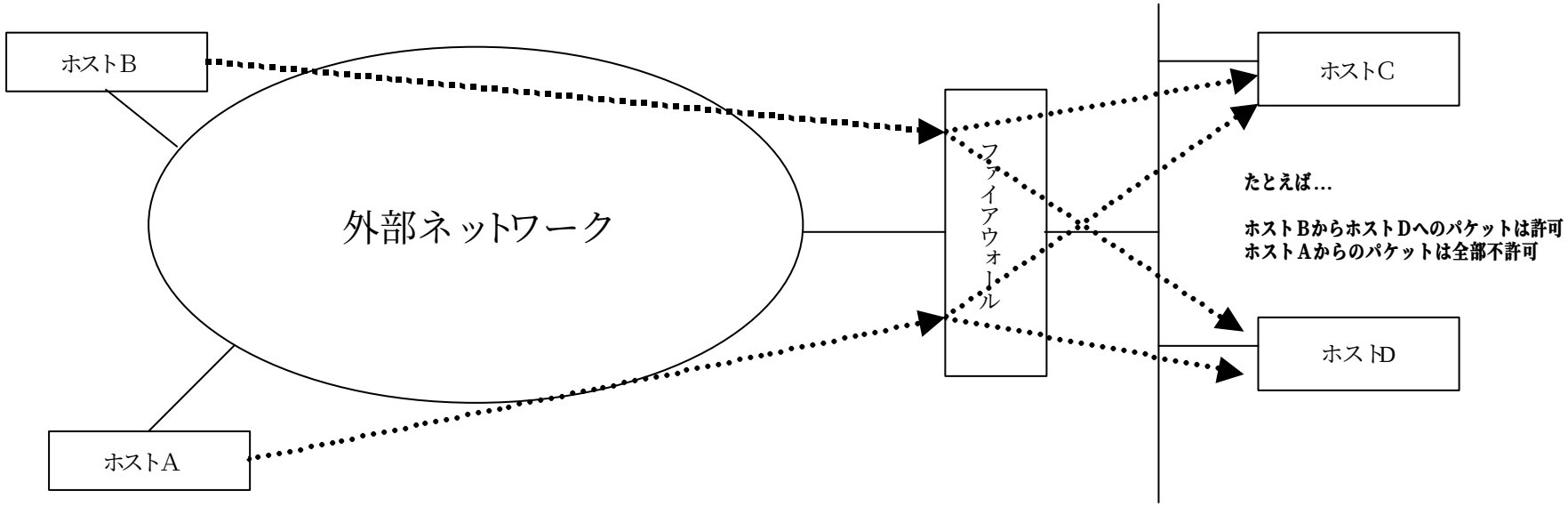
- 通信の種類、宛先、発信元による通過許可・禁止の制御
- 通信の相手先の確認・認証
- 通信の記録（時刻、宛先、発信元、サービスなど）
- 通信内容の監査（不正行為の検出、コンテンツの確認など）

ファイアウォールの基本方式

- パケットフィルタ方式
  - IPパケットのforwarding時に監査を行う方式
  - 宛先、発信元、サービスなどのヘッダ情報で制御
- アプリケーションゲートウエイ方式
  - IPパケットのforwardingは行わない
  - アプリケーション層でデータを中継 •監査
  - PROXY サービス

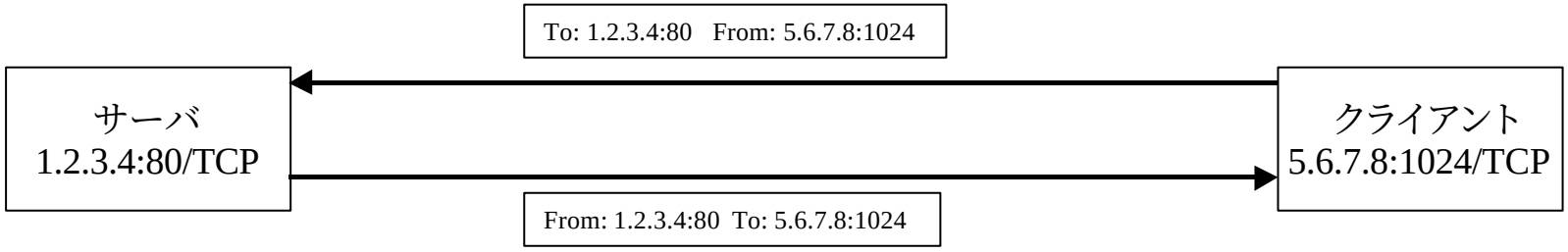
パケットフィルタ

- IP / TCP / UDP パケットヘッダ情報をもとに中継処理を制御
- 宛先・発信元のアドレス、ポート番号をもとに通過の可否を判断
- ルータ・ファイアウォールなどパケットの中継点（ゲートウエイ）で処理する。



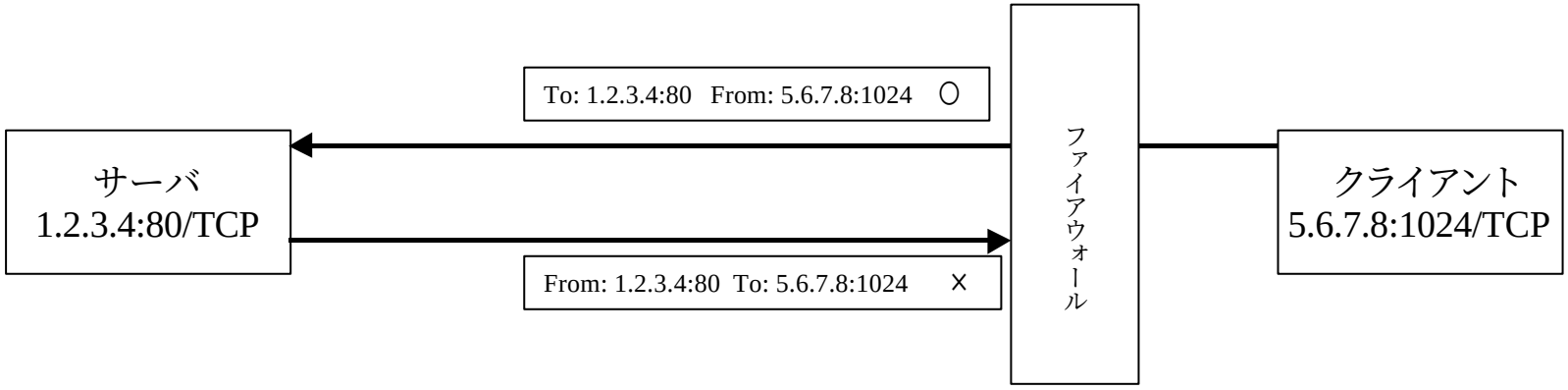
パケットフィルタの特性

たとえば WWW の通信は…



パケットフィルタの特性

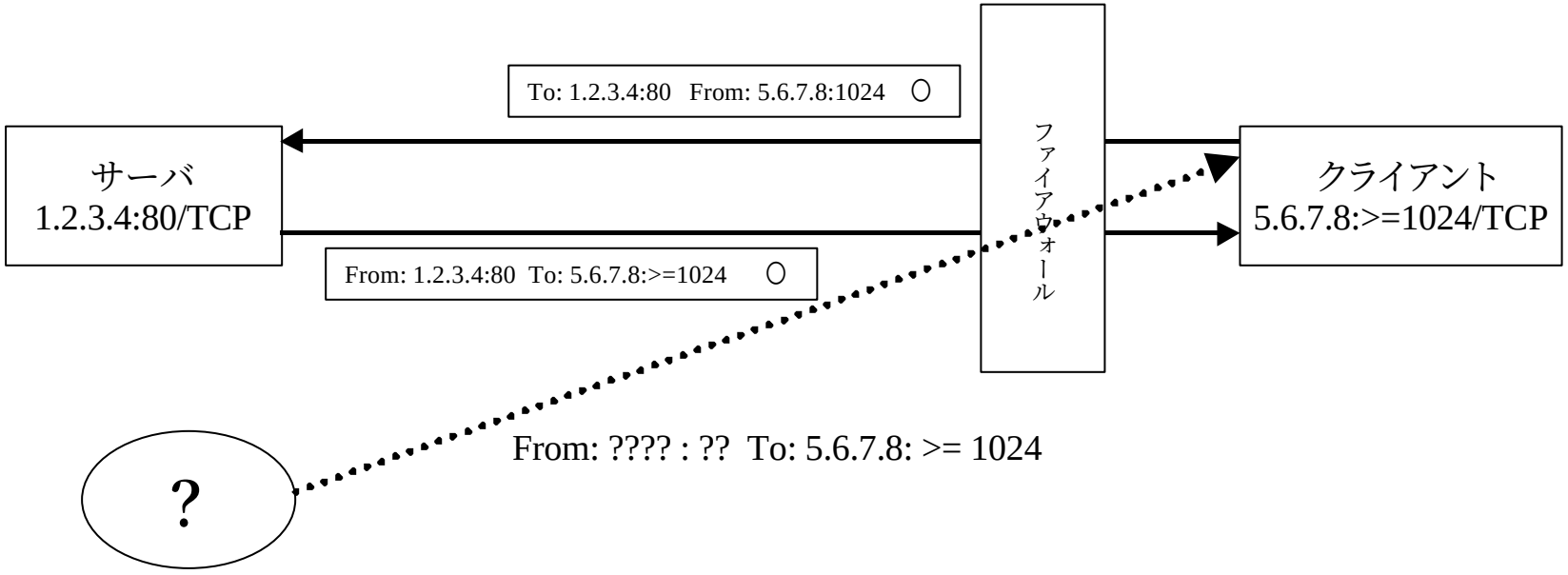
たとえばクライアントがファイアウォールの中にあって、外部への一方通行の通信が許可されているだけだとすると・・



1方向を許可しただけの単純なフィルタリングでは「行きはよいよい、帰りは…」で通信が成り立たない。

パケットフィルタの特性

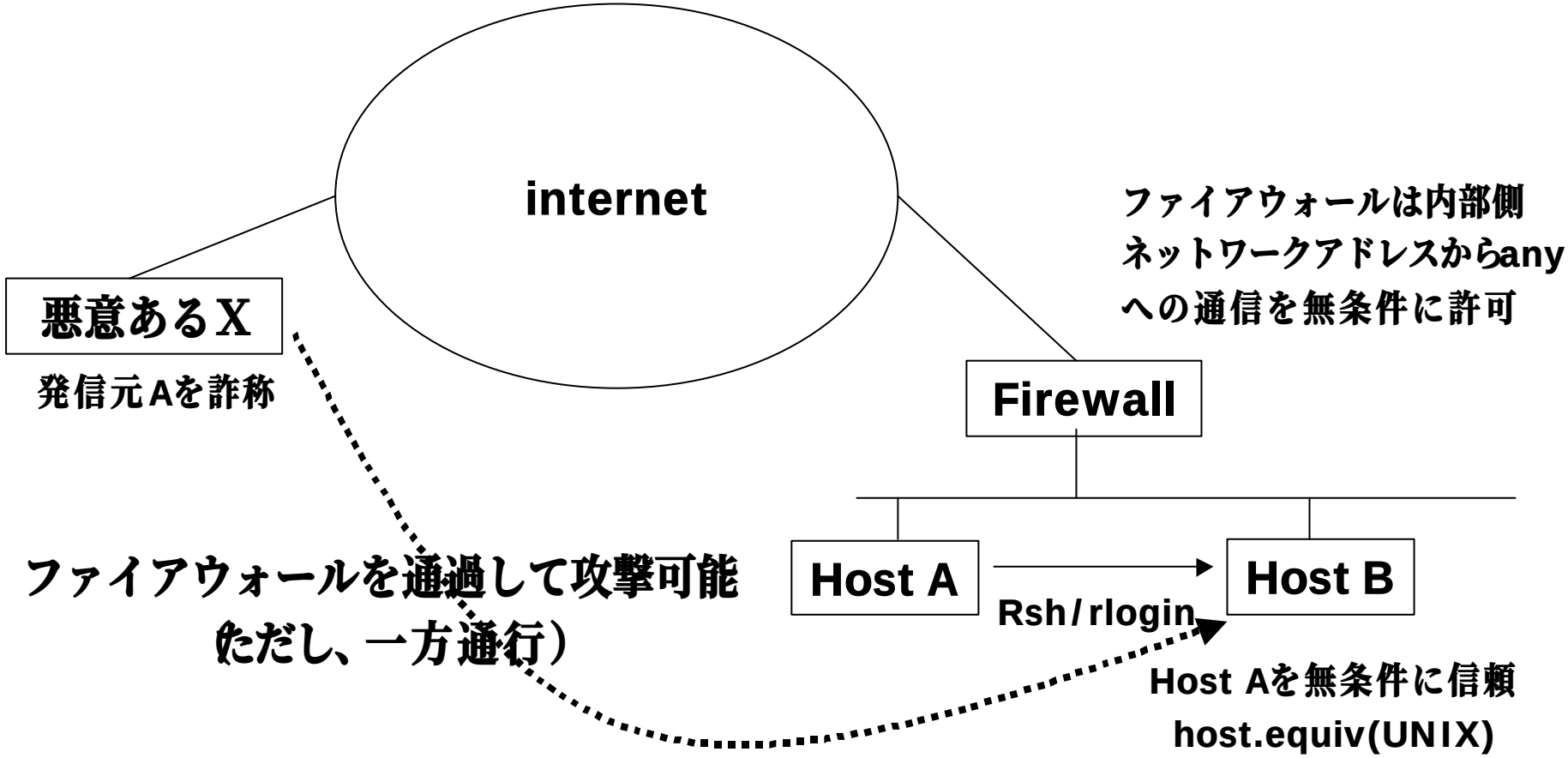
応答を通すためには外部から1024番以上のポートへの着信を許可しなければいけない。
もし、クライアント側の1024番以上のポートにサービスが上がっていると外部から攻撃されてしまいかねない。



パケットフィルタの特性

- IP詐称攻撃に対する本質的な弱点
  - 内部側ネットワークアドレスの詐称
  - 信頼しているネットワークアドレスの詐称
  - 発信元IPアドレスの詐称は容易
  - 一方通行でも有効な攻撃手法もある

IP詐称攻撃の例（内部側ネットワーク詐称）

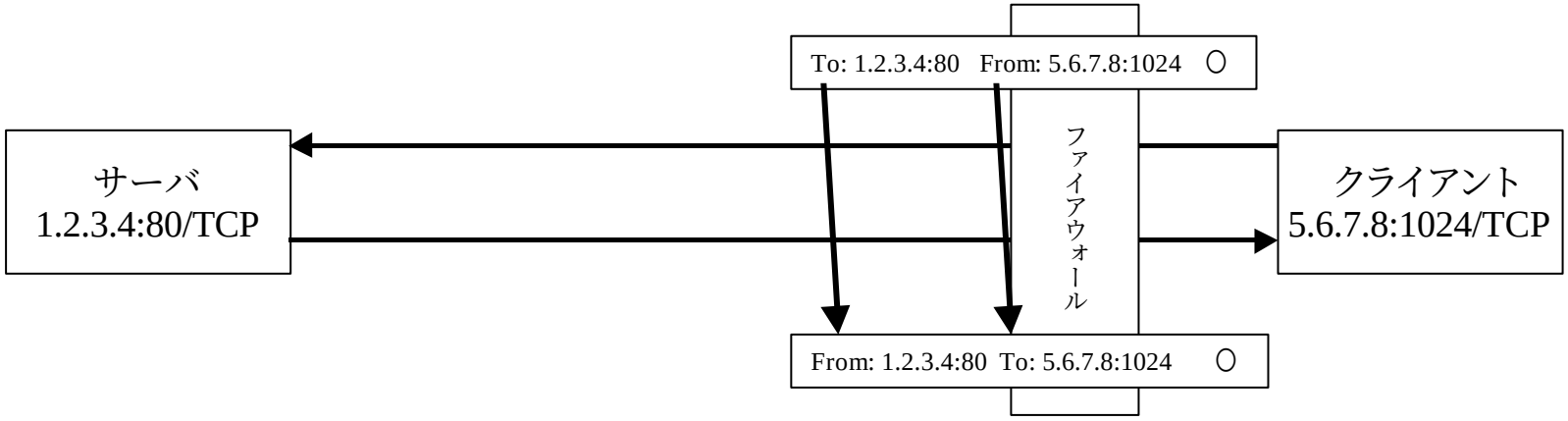


パケットフィルタの特性を補う機能

- ダイナミックパケットフィルタ
  - コネクション／セッションの認識と管理
  - 応答パケットを自動的に認識して通過させる
- 方向性フィルタ
  - パケットがどのネットワークインターフェイスから入ってきたか（出ていくか）を区別
  - 内部側アドレス詐称対策

ダイナミックパケットフィルタ

- **通過する通信のセッションを監視することで、通過を許可したパケットへの応答のみを選択的に通過させる。**
- **TCP はコネクション開設、切断を監視して許可とその取り消しを制御できる。（シーケンスNoなどもチェック可能）**
- **UDPやその他の特殊なものについては、行きのパケットの通過で応答通過も許可し、一定時間通信がなければ取り消す。**

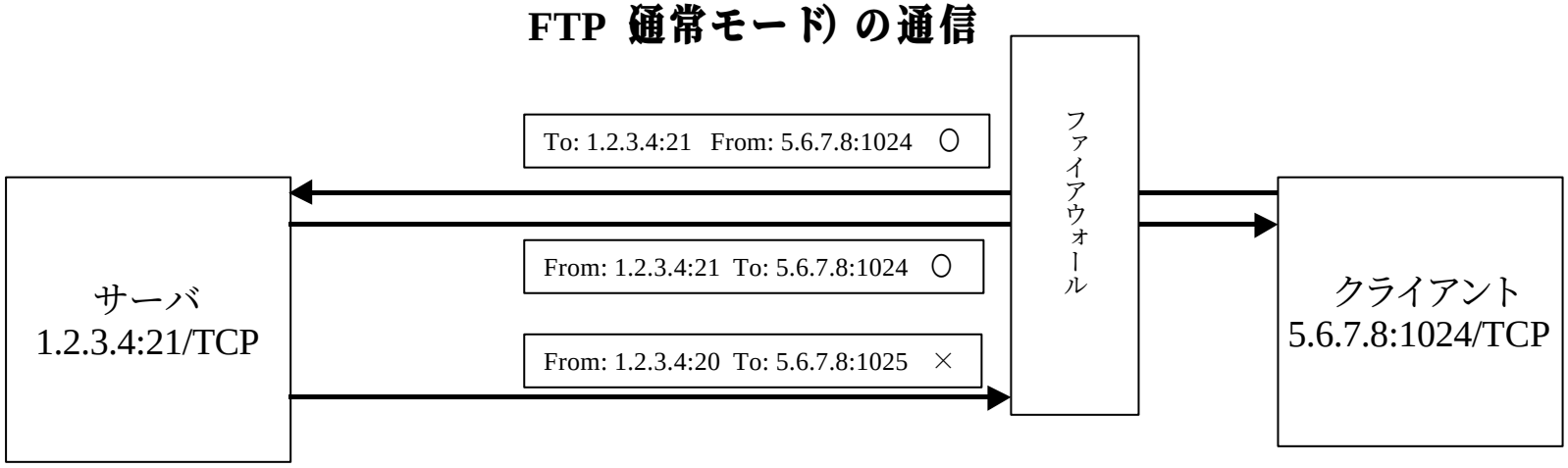


ダイナミックパケットフィルタの課題

- **コネクションレスタイプのプロトコル（UDPなど）の許可取り消しタイミングが微妙。**
  - **一定時間の無通信で許可を取り消すため、その時間、ポートが開きっぱなしになる。**
  - **でも、全部開きっぱなしよりは格段に安全ではある。**
- **複数セッションを使用する通信への対応が困難。**
  - **古典的には ftp。サーバ 21 番へのコネクションのみではなく、サーバ側からクライアントのダイナミックなポートへの張り返しがある。**
  - **Streaming 系アプリ (Real**, Stream Works, VDO Live, NetMeeting, CU-SeeMe etc.) 多くは複数の TCP/UDP セッションを複合した通信を行う。**
  - **上位プロトコルの通信をモニタして対応するしかない。（新しい技術の登場はファイアウォールメーカ泣かせ）**

Ftpとダイナミックパケットフィルタ

データコネクションに使用するポート番号はコマンドコネクションの通信内でネゴシエーションが行われ動的に決定される。自動的に通過許可を行うためには、通信内容の監視が必要。



FTP の場合は PASV モードで回避できるのだが…。
市販製品のほとんどは、通常モード対応可

プライベートアドレス使用の普及

- IPアドレス枯渇の危機とプライベートアドレス
  - RFC1918による定義と使用の推奨
  - 広大な空間(10.0.0.0/8, 172.16.0.0/12,192.168.0.0/16)を自由に使用可能
  - 一方でインターネットとの通信がそのままでは不可能

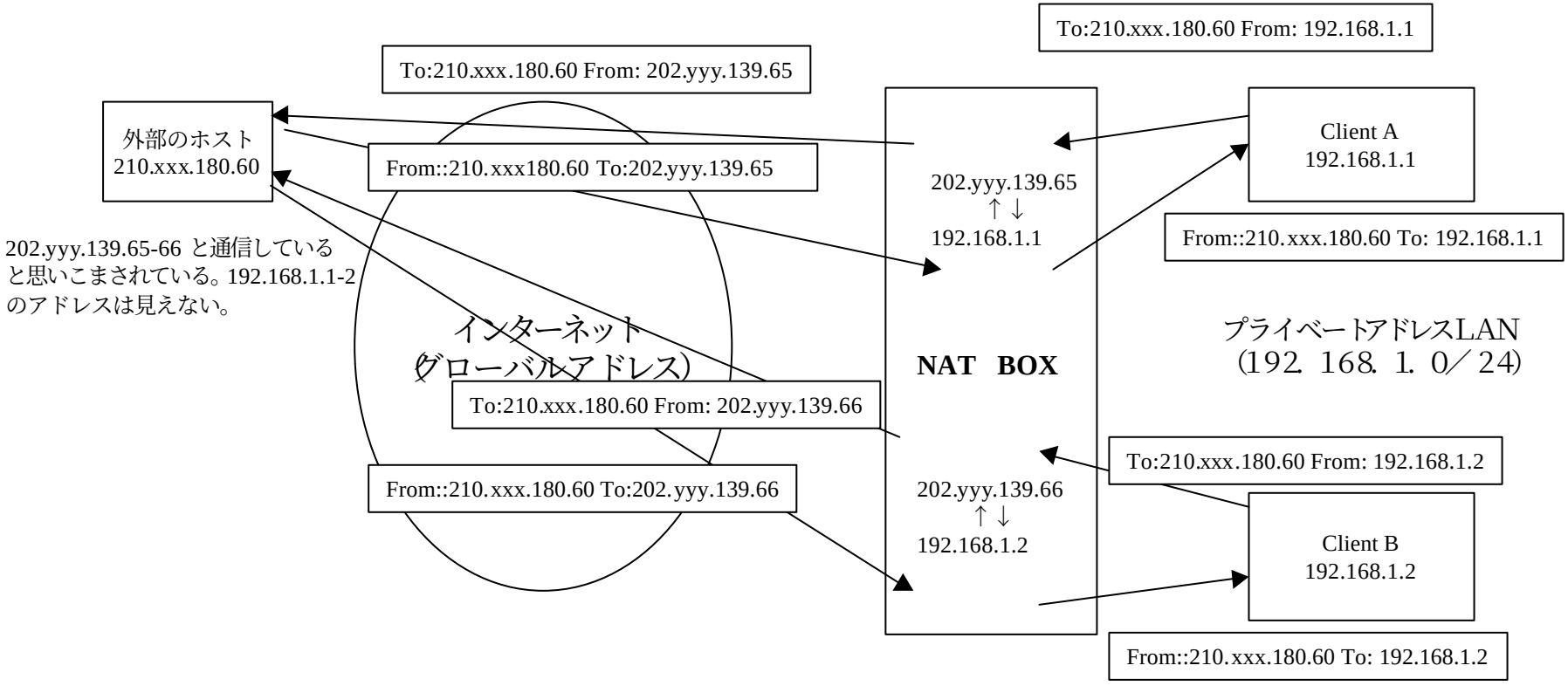
NAT (Network Address Translation)とは

- 通過するパケットのヘッダにあるアドレスを書き換えて、サーバやクライアントを騙す手法。
- プライベートアドレス（RFC1918）を使用したネットワークをインターネットなどに接続する技術として使われる。
- NAT の基本は RFC1631 で規定される。しかし、一般にはさらに広い意味で解釈されており、多少、用語的な混乱もみられる。
- NAT の考え方を基本にして様々な応用型が可能で、現実にファイアウォール製品、ルータなどへの実装が行われている。

プライベートアドレスを隠すNAT(RFC1631)

- RFC1631はNATの基本形
  - 内部側クライアント用に一定数のグローバルアドレスをプールしておく。
  - 内部側クライアントが外部と通信しようとする時に、１クライアントについて1個のアドレスを割り当てる。
  - 内部側から外部へのパケットは発信元アドレスを割り当てたアドレスと交換する。
  - 外部から割り当てたアドレスへのパケットは宛先を内部側クライアントのアドレスと交換する。
- １クライアント対１グローバルアドレス（1対1）のアドレス変換
- インターネットへの同時接続数はプールされているグローバルアドレスの個数に依存する。

NAT(RFC1631)の様子



NAT(RFC1631)の問題点

- 同時接続数分のグローバルアドレスを消費する。
  - 多数のクライアントの同時接続性を求めると、あまりアドレスの節約にはつながらない。
- プールされたアドレスの割り当てと解放のタイミングが難しい。
  - 割り当てたアドレスの使用を終わってから、他のクライアントに割り当てを行う時間をどれくらいに設定するか。（DHCPなどとよく似た問題）
  - 短すぎると同一クライアントでアドレスがころころ変わる。
  - 長すぎると、解放されるまで待たされるクライアントが増える。

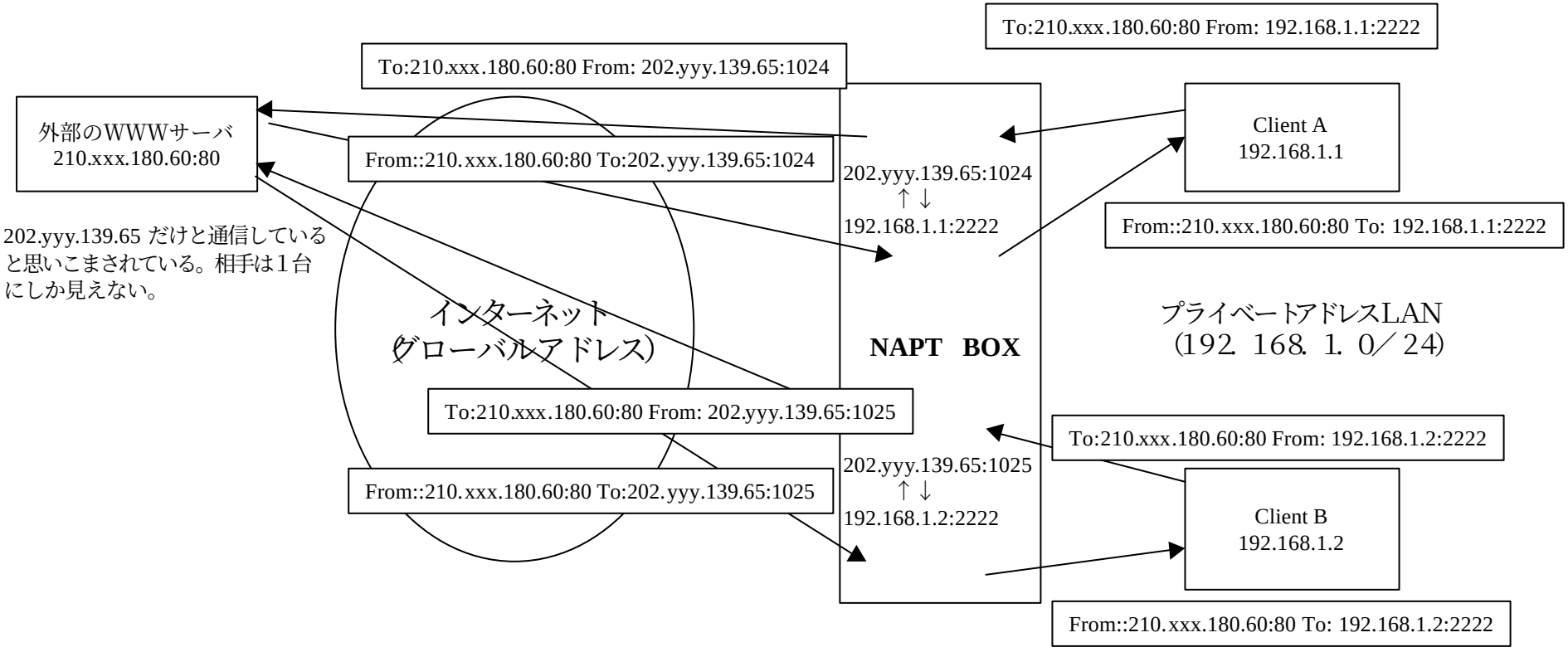
NAT(RFC1631)の問題点の回避策

- 1アドレスに複数のクライアントを割り当てる方法(NAPT)
  - Network Address Port Translation (NAPT)
  - RFC2391(LSNAT:後述)内に記述されている手法
  - ちまたの IP Masquerade 、NAT Plus、NATe、Advanced NAT…etc. と呼ばれる手法はほぼ同一のもの。
  - アドレスのみでなく、ポート番号も変換することで、同じアドレスを異なるクライアントの複数のセッションで共用する。
- 大量のクライアント(セッション)を1アドレスでさばくことが可能。
  - 1024 - 65535 のポート番号をダイナミックに割り当てれば、計算上は最大64511同時セッションに対応できる。

NAPT の考え方

- なぜ NAT は 1対 1でないといけないか。
  - 各クライアントは発信元ポートを1024以上の任意の値に割り当ててくる。
  - 当然、同じ番号を使う可能性があり、その場合、同じアドレスに変換すれば区別がつかなくなってしまう。
- ポート番号の変換の考え方
  - NAT のアドレスプールの考え方と同様にポート番号のプールを考える。
  - アドレスを1個にするかわりに、クライアントのポート番号をプールから割り当てたポート番号と入れ替える。
  - ポート番号が重ならなくなるので、セッション単位に識別が可能になる。
  - プールが大きいので解放タイミング問題も比較的ルーズな解決が可能である。（順次割り当て：一周するころには再割り当て可能になる）
- ダイナミックパケットフィルタとの親和性がよい
  - パケットフィルタ型ファイアウォールへの実装

NAPT の様子


To:210.xxx.180.60:80 From: 192.168.1.1:2222
To:210.xxx.180.60:80 From: 202.yyy.139.65:1024
外部のWWWサーバ
210.xxx.180.60:80
From::210.xxx.180.60:80 To:202.yyy.139.65:1024
202.yyy.139.65:1024
↑↓
192.168.1.1:2222
Client A
192.168.1.1
From::210.xxx.180.60:80 To: 192.168.1.1:2222
202.yyy.139.65 だけと通信していると思いこまされている。相手は1台にしか見えない。
インターネット
(グローバルアドレス)
NAPT BOX
プライベートアドレスLAN
(192. 168. 1. 0／24)
To:210.xxx.180.60:80 From: 202.yyy.139.65:1025
To:210.xxx.180.60:80 From: 192.168.1.2:2222
From::210.xxx.180.60:80 To:202.yyy.139.65:1025
202.yyy.139.65:1025
↑↓
192.168.1.2:2222
Client B
192.168.1.2
From::210.xxx.180.60:80 To: 192.168.1.2:2222

NAPT の問題点

- 発信元に固定のポート番号が必要なサービスへの対応が困難。（発信元ポートを変えると動作しないものもある）
  - NAPTとNATをうまく併用する。固定割り当てと動的割り当ての併用。

- 複数の動的なセッションを使ったり、サーバからの逆向きセッションが付随するようなアプリケーションへの対応が困難。
  - ダイナミックパケットフィルタと同種の問題が生じる。

- NAT , NAPT とも、アプリケーション層で IP アドレスを受け渡すような種類のアプリケーションへの対応が困難。

NAT,NAPT の応用型

- 仮想ホスト (Static NAT)
  - プライベートアドレスにあるサービスを一部だけグローバルアドレスに見せる手法。
  - 複数サーバの異なるサービスをひとつのアドレスに見せたりすることが原理的に可能。(固定ポート変換との組み合わせ)
  - ひとつのサーバの異なるサービスを複数のアドレスの同一のサービスに見せることも原理的に可能。(固定ポート変換との組み合わせ)

- アクセス負荷分散への応用
  - ひとつのアドレスのサービスへのアクセスをセッション単位で複数のサーバに分散させるような応用も可能。(LSNAT: RFC2391)

アプリケーションゲートウエイ

- デュアルホームホストだが、原則としてパケット中継はしない。
- ゲートウエイ上で動作するPROXYサーバによるアプリケーション層でのデータ中継
- 「遅いが安全」と言う神話

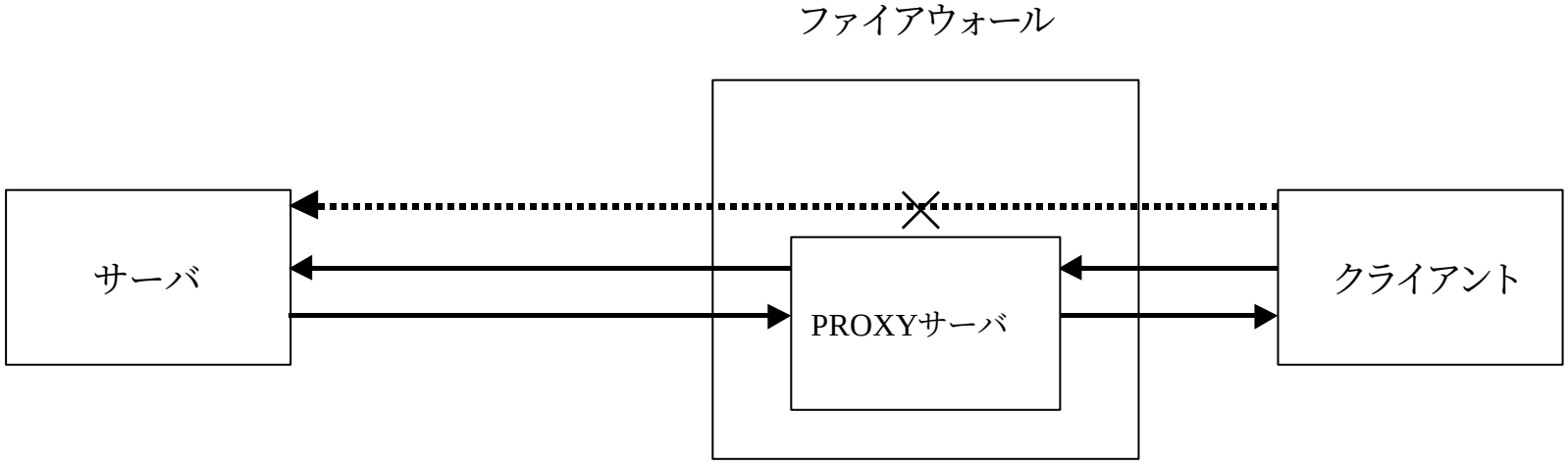
PROXY と言う名の代理人

- ファイアウォールとして見たPROXY（サーバ）
  - パケットフィルタ＝パケットレベルの通信の中継制限によるセキュリティー（TCP/UDP/IP 層）
  - PROXY（プロキシ、プロクシ）＝アプリケーションレベルの中継（代行サービス）

- パケット中継は禁止して、外部へのアクセスを代行させる。
  - PROXY も内外の 2つのネットワークに接したホスト（デュアルホームホスト）で動作する。
  - クライアントは、PROXY サーバにアクセスするための手順をサポートする必要がある。（最近の製品は透過モードをサポート）
  - PROXY サーバは基本的には対象となるアプリケーション（サービス）ごとに設置する必要がある。（汎用 PROXYサーバも存在するが、これはトランスポート層 (TCP/UDP 層）でのデータ中継を行うのみ）

PROXYサーバの構成

ファイアウォールは内外の２つのネットワークに接しているが、その間のパケット中継は一切行わないため、内部ネットワークは保護される。クライアントは直接サーバと通信できず、ファイアウォール上の PROXY サーバに中継を依頼する。



PROXYの弱点

- パケットフィルタに比べて使い勝手は制限される。
  - たとえば、Web ブラウザにPROXY を使う設定をする必要があるなど、PROXY を使うために特別な設定が必要な場合もある。
  - PROXY サーバが用意されているサービスしか利用できない。
  - アクセスにタイムラグが発生する。（一旦、データを蓄積して転送）
  - 一般にUDPサービスの中継が不得手

何故PROXYを使うのか

- では、利点は何？
  - 遅い」と 遅くていい」の違いとは？
    - アプリケーション層ではリアルタイム性の要求が低く、重い処理でも組込可能
  - パケットフィルタには困難なアプリケーションレベルのチェックを組み込むことが可能。（ウイルスチェック、URL のチェック、特定のキーワードやデータパターンによる排除など）
  - キャッシュ機能の組み込みなどでアクセスの効率化が可能。
  - 詳細なログを取ることができる。（どこのサーバに接続したか、だけではなくどのファイルをアクセスしたか…など）
  - NAT が不要。内部側がプライベートアドレスでも問題ない。

高速化と安全性の追求

- 両方式それぞれの弱点の強化
- Packet filter →Stateful Inspection
  - CheckPoint社
- Proxy → Adaptive Proxy
  - Network Associates 社

Stateful Inspection

- CheckPoint社が開発、Firewall-1 に実装
- パケット中継のみでなく、アプリケーション層を流れるデータの監査が可能。
- 検査パターンを定義可能（専用定義言語のサポート）
- 速度を多少犠牲にして安全性を強化

Adaptive Proxy

- Network Associates 社が開発、Gauntletに実装
- Proxy サーバとダイナミックパケットフィルタを組み合わせたハイブリッド方式。
- 通信の最初の部分を厳密に監査、残りはパケットフィルタで通過処理
- 安全性を多少犠牲にして高速化

適材適所が重要

- 高速性の要求
  - Packet Filter ベースのファイアウォール
- 安全性の要求
  - アプリケーションゲートウエイ方式のファイアウォール
- ハイブリッド化の進行
  - 1個の製品内での適材適所

ファイアウォールをどう使うか

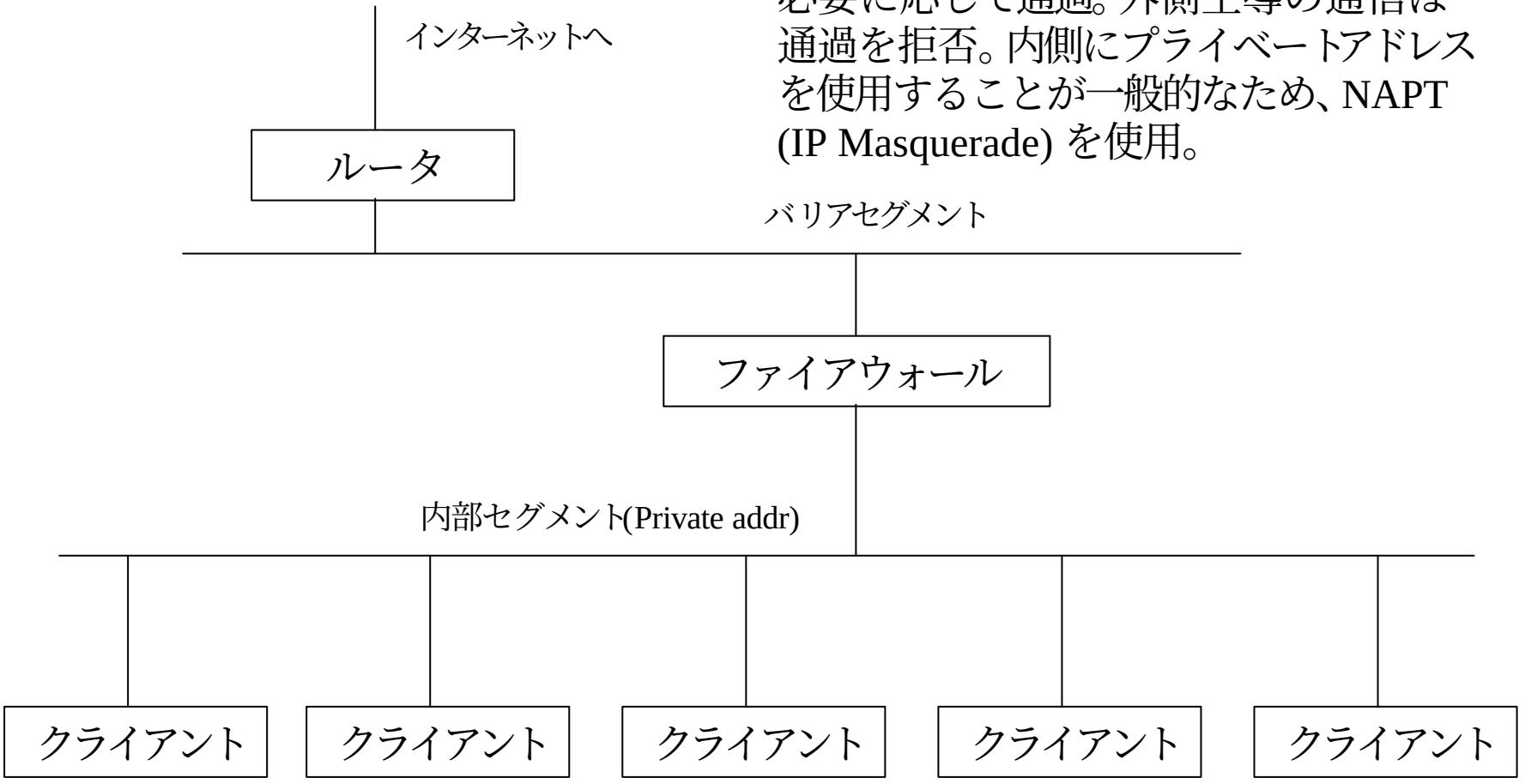
- インターネットから内部ネットワークを保護
- 公開サーバの保護
- 複数の異なるネットワーク間のセキュリティ確保
- VPNゲートウエイ

内部ネットワークの保護（単純モデル）

- 原則として、外部側から内部あての着信を禁止。
  - 内部側から開始したセッションに対する応答パケットのみを通過。（パケットフィルタ）
  - 必要なサービスについてProxyサーバを起動
- 内部側がプライベートアドレスの場合、パケットフィルタでは、NAT(NAPT / IP Masquerade)を併用。

単純モデル

基本的なポリシーは、内側主導の通信は必要に応じて通過。外側主導の通信は通過を拒否。内側にプライベートアドレスを使用することが一般的なため、NAPT (IP Masquerade) を使用。



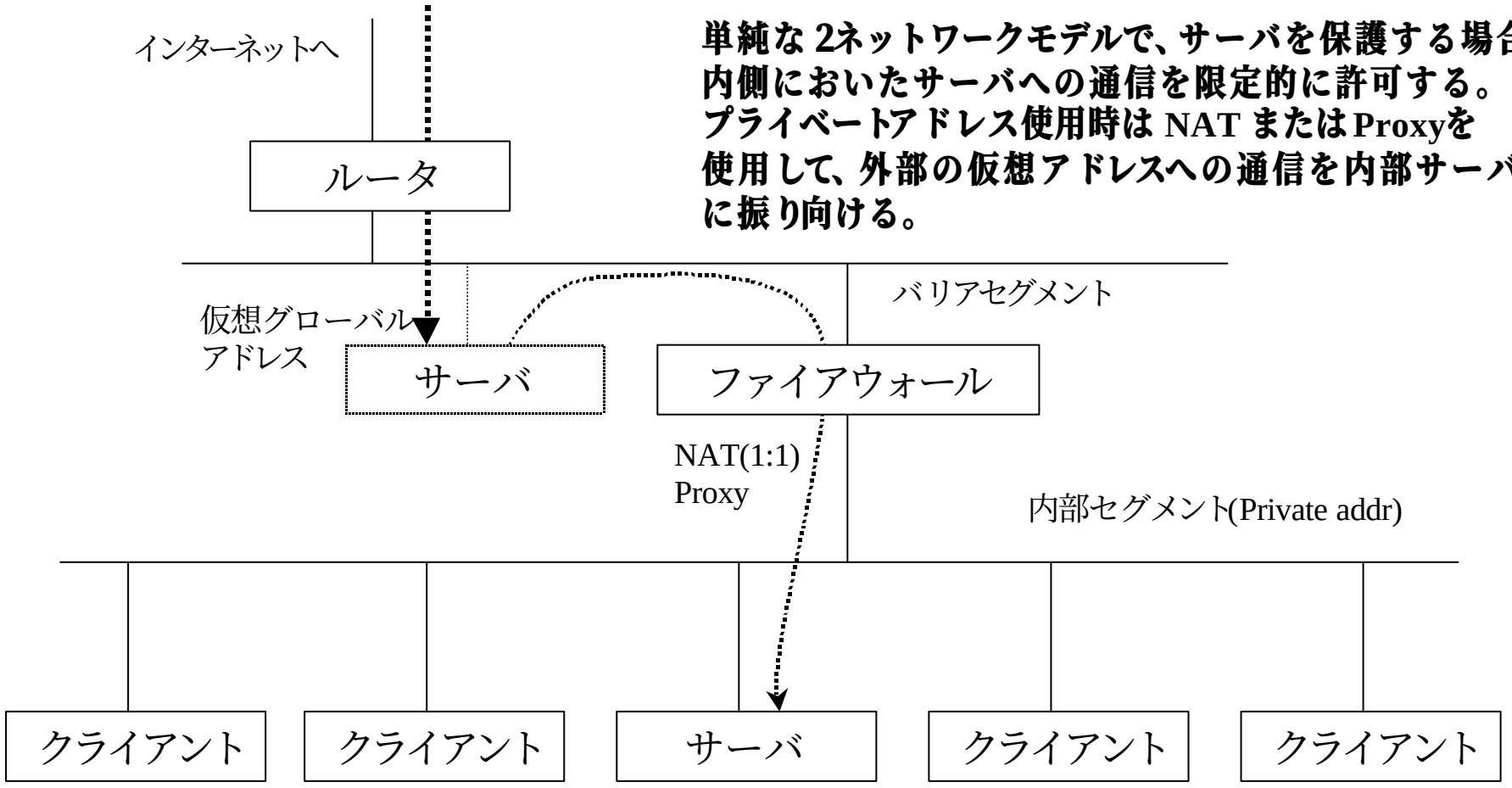
ファイアウォールによるサーバ保護

- 公開以外のサービスへのアクセスを阻止
- サーバ自身のセキュリティ対策を簡素化
  - 公開サービスに対策を集中できる
- 公開サービスは、サーバ側でセキュリティ対策を

サーバ保護（単純モデル）

- ファイアウォールの内側にサーバを配置
- 内部側がプライベートアドレスの場合
  - Static NAT を使用してグローバルアドレスに特定サービスをマッピングする。
  - Proxy を外部側の公開アドレスにバインドさせ、中継させる。
- 万一、侵入を受けた場合のリスクに配慮が必要。

サーバを保護する場合（単純モデル）

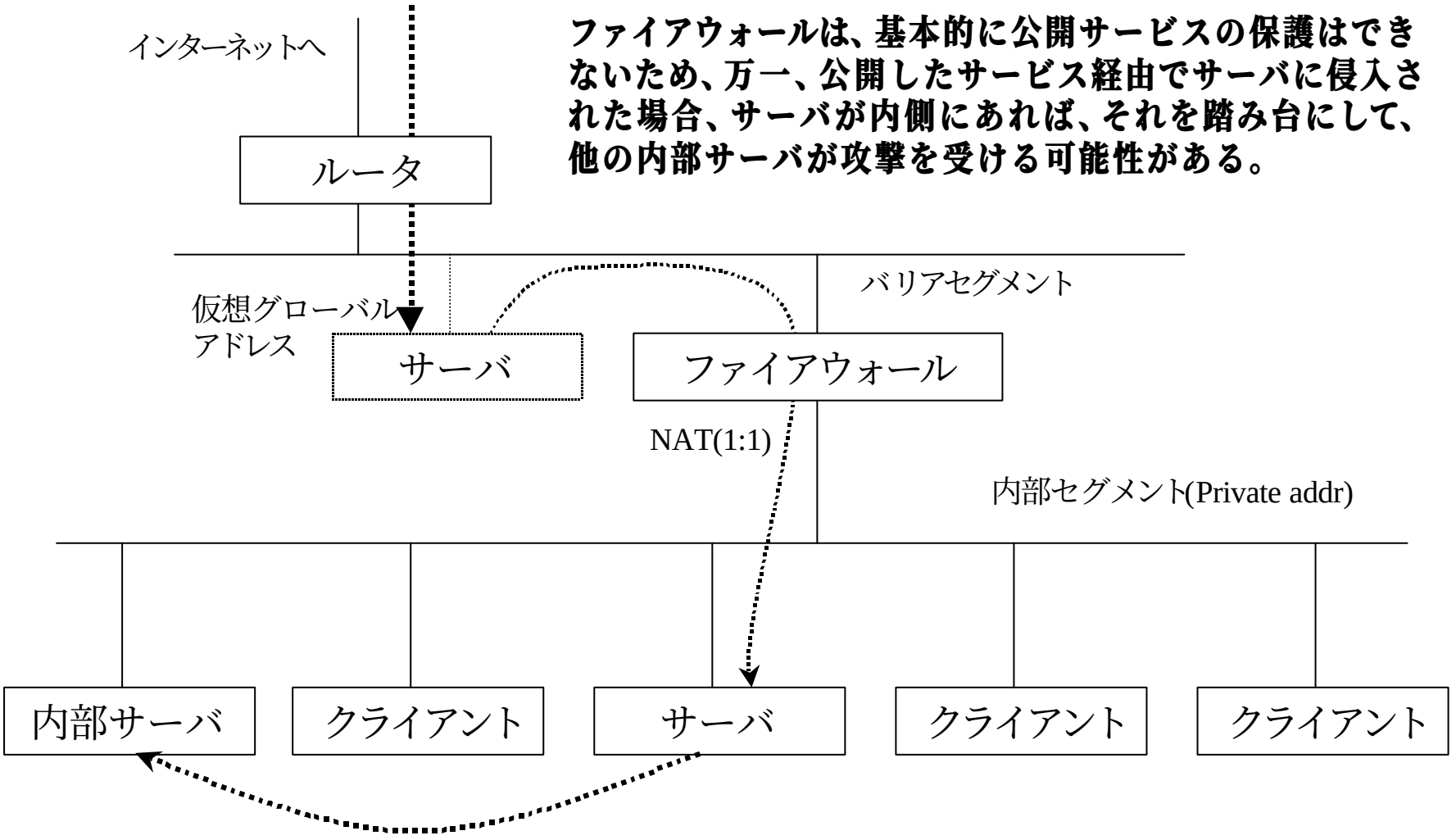


単純な2ネットワークモデルで、サーバを保護する場合内側においたサーバへの通信を限定的に許可する。プライベートアドレス使用時はNATまたはProxyを使用して、外部の仮想アドレスへの通信を内部サーバに振り向ける。

単純モデルによるサーバ公開の危険性

- 侵入を許してしまった場合、直ちに内部ネットワークに危険が生じる。
- 直接的にログインを許すようなサービスは避けるほうが賢明。
- サーバ（サービス）のセキュリティホールに注意
  - バッファオーバランなどを利用したサービス乗っ取りも可能

単純モデルでのサーバ保護の問題点



ファイアウォールは、基本的に公開サービスの保護はできないため、万一、公開したサービス経由でサーバに侵入された場合、サーバが内側にあれば、それを踏み台にして、他の内部サーバが攻撃を受ける可能性がある。

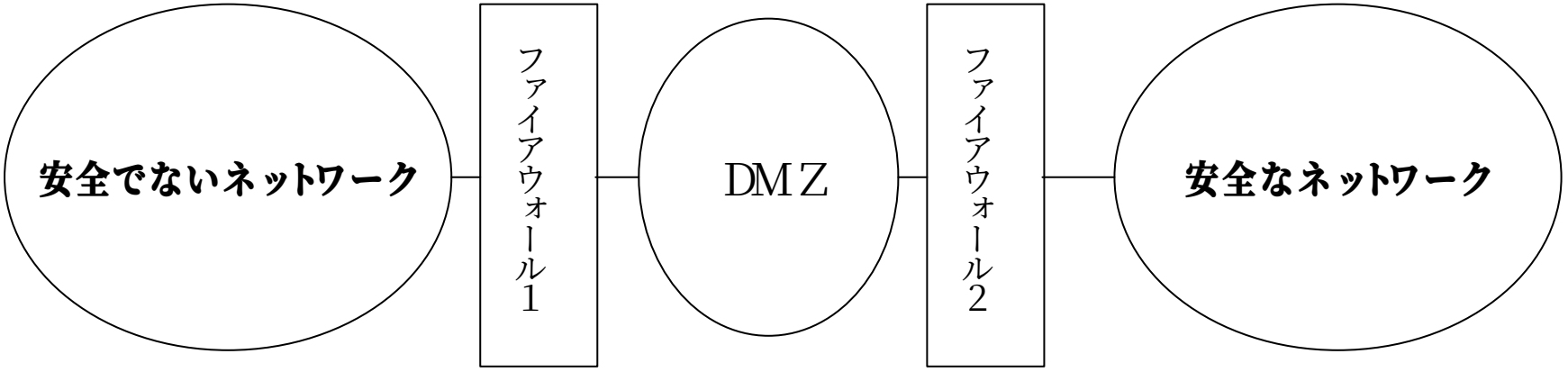
DMZ（緩衝地帯）モデル

- DMZは中間的な保護層
  - 公開サーバ群をファイアウォールで保護し、必要以外のアクセスを排除する。
  - 万一、公開サーバが不正アクセスにより侵入されるなどの事態が生じても、そこから内部に直接入れないようにすることで、安全性の向上をはかる。（不正アクセスに対応する時間をかせぐ）
  - さらに、外部へのアクセスも制限することで、侵入されたサーバを踏み台にして外部を攻撃することも困難にする。（かごの鳥作戦）
  - 不正アクセスによって深刻な事態に陥るような重要なホストは置かない。

教科書的DMZの形式

外部→DMZ
必要なサービスへの
アクセスのみ通過

DMZ→内部
基本的に通過できな
いように設定。



外部←DMZ
基本的に通過できな
いように設定。どうし
ても必要なもののみ
通過。

DMZ←内部
必要なサービスへの
アクセスのみ通過

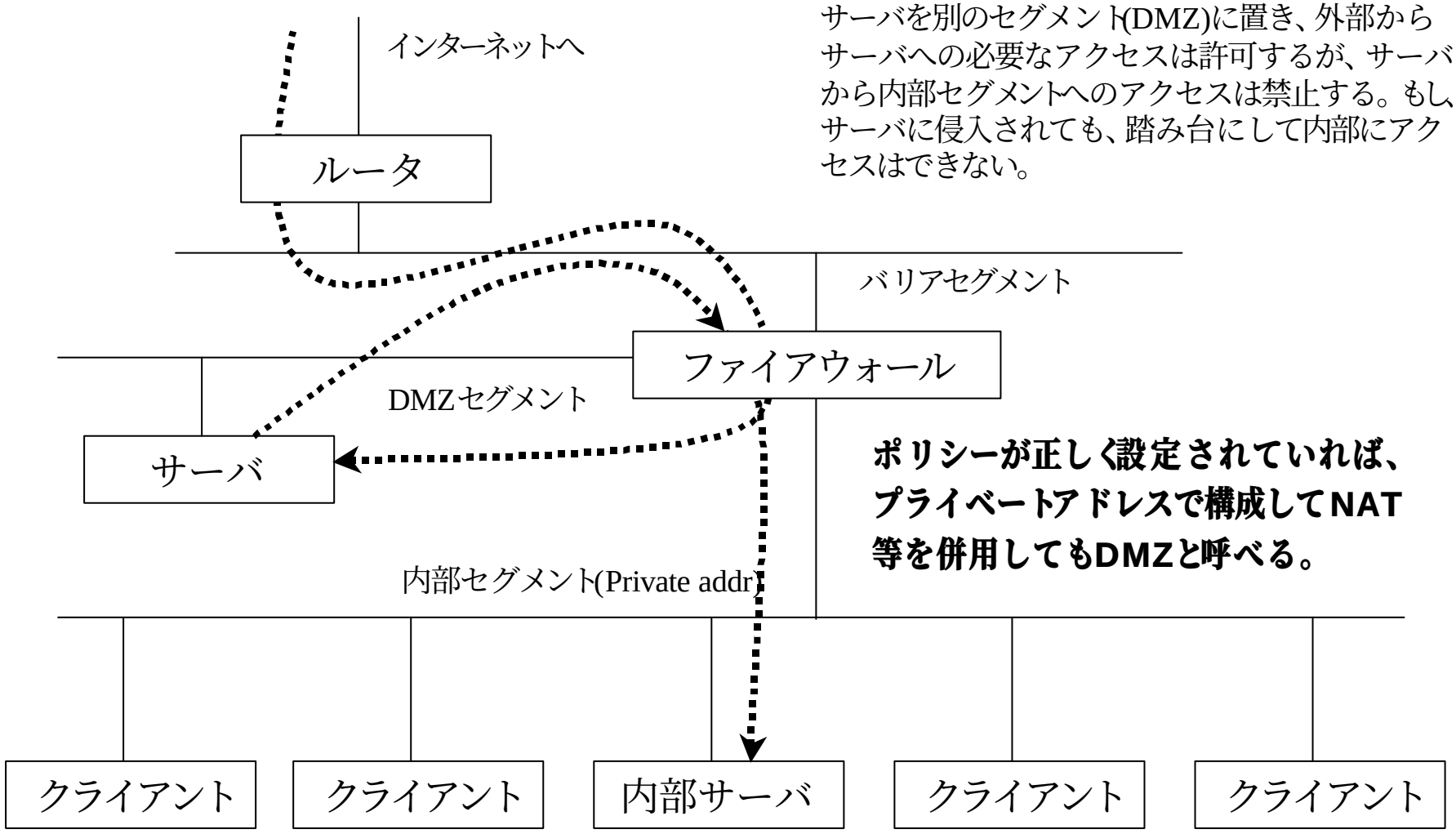
非武装地帯」の罠… DMZ

- DMZ (De-Militarized Zone) ＝ 非武装地帯」は軍事用語の直訳
  - 非武装」＝ 無防備」という言葉上の誤解を生む。
  - 実際の意味は不意の侵略に対応する時間を稼ぐための 緩衝地帯」に近い。
  - そういう意味ではまさに 最後の防衛線

- ファイアウォールの 3個目のネットワークカードは DMZ?
  - 本来は 2個のファイアウォールに挟まれた中間層。
  - 1台のファイアウォールでエミュレートしているだけ。
  - きちんと設定しないとDMZ とは呼べない。
    - DMZから内部へは入れない 原則としてすべて禁止、どうしても必要な場合でも、可能な限り対象を限定し、セキュリティ対策を講じる）
    - DMZから外部へも出られない 必要なサービスのみしか許可しない）

DMZ（緩衝地帯）モデル

サーバを別のセグメント(DMZ)に置き、外部からサーバへの必要なアクセスは許可するが、サーバから内部セグメントへのアクセスは禁止する。もし、サーバに侵入されても、踏み台にして内部にアクセスはできない。



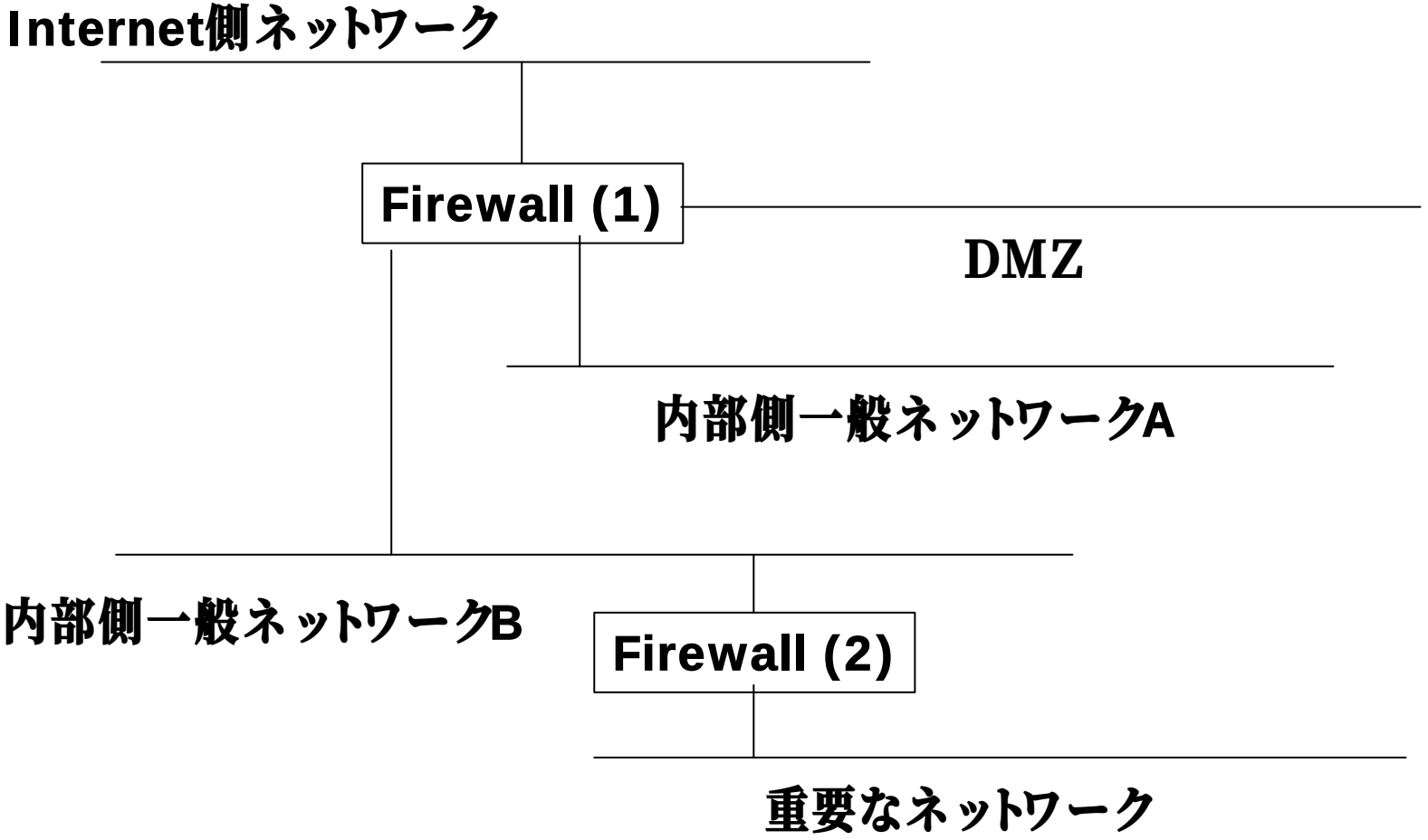
複数のネットワークのポリシー管理

- 内側 V.S. 外側という構図
  - 内側＝安全、外側＝危険という単純な発想
- セキュリティは相対的なもの
  - 扱う情報、リソースの種類によって保護ポリシーが異なる。
- 当然、内部側にもポリシーの異なる複数のネットワークがあっていい。（ないといけない）

複数のネットワークのポリシー管理

- 接続しないことも重要な管理手段
- 他のネットワークとの接続時のアクセス制御
  - アドレスによるアクセス制御（フィルタ）
  - ユーザによるアクセス制御（認証）

内部側複数ネットワークモデルの例



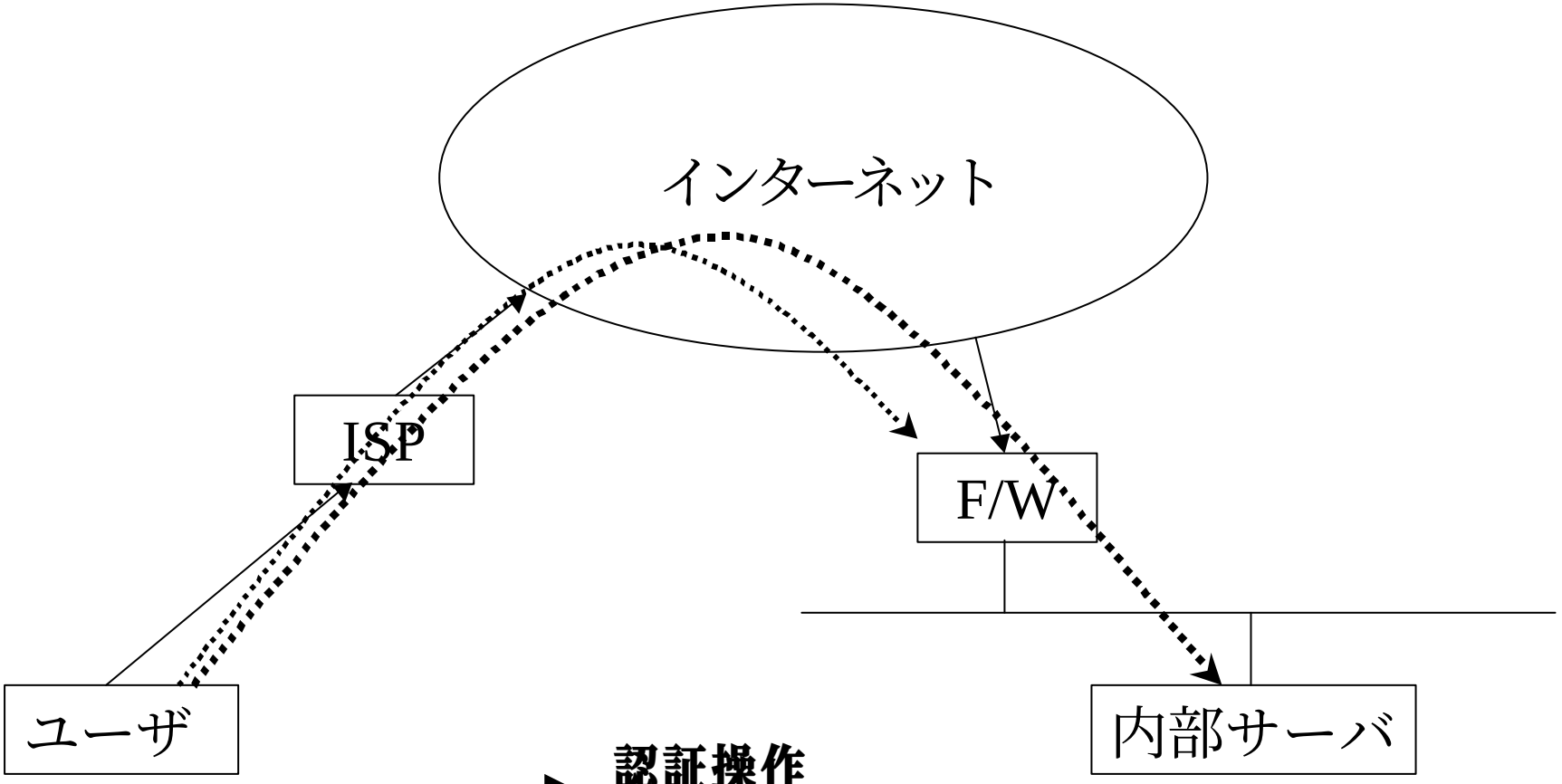
認証によるアクセス許可

- フィルタ設定では、アドレス単位のアクセス許可しかできない。
  - たとえば、ダイヤルアップユーザが外部のプロバイダから内部にアクセスした場合、毎回アドレスが変わり、フィルタでの処理が不可能。
- 一旦、ファイアウォールに接続して認証を受けることで、その通信に限ってアクセスが許可される枠組みが必要。
- 認証の弱いサービスに対し、強力な認証機能を提供
  - ワンタイムパスワード認証（S/Key, SecurIDなど）
  - RADIUS等の認証サーバとの連携

認証の方式

- サービス単位の認証
  - Proxy サーバ等で接続中継時に認証を行う
  - サービス（アプリケーション）ごとに認証操作
- IPアドレス認証
  - 一回の認証操作でそのIPアドレスを使っているユーザを認証
  - 複数のサービスに対するアクセス権を一括して与える

認証操作



ファイアウォールとVPN

- ファイアウォール製品の多くがVPN接続に対応
- IPSecへの対応による相互接続性
- ファイアウォール、ルータなどとの相互接続による仮想ネットワーキング
- モバイルクライアントへの安全なアクセス手段の提供

VPNの方式と互換性

独自方式の VPN

- 初期のファイアウォールなどに搭載されたメーカ独自の方式
- 基本的に他社製機器との接続性は保証されない
- 高いシェアのメーカによる顧客の囲い込みに有利
- 互換性がないことが利点である場合もないではないが…

標準方式のVPN

- IPsec の標準化の進行（RFC24xx）
- 標準に準拠した異なるメーカの機器を相互接続
- 暗号鍵を自動的に交換（IKE）
- 企業連合によるエクストラネットの構成に有利

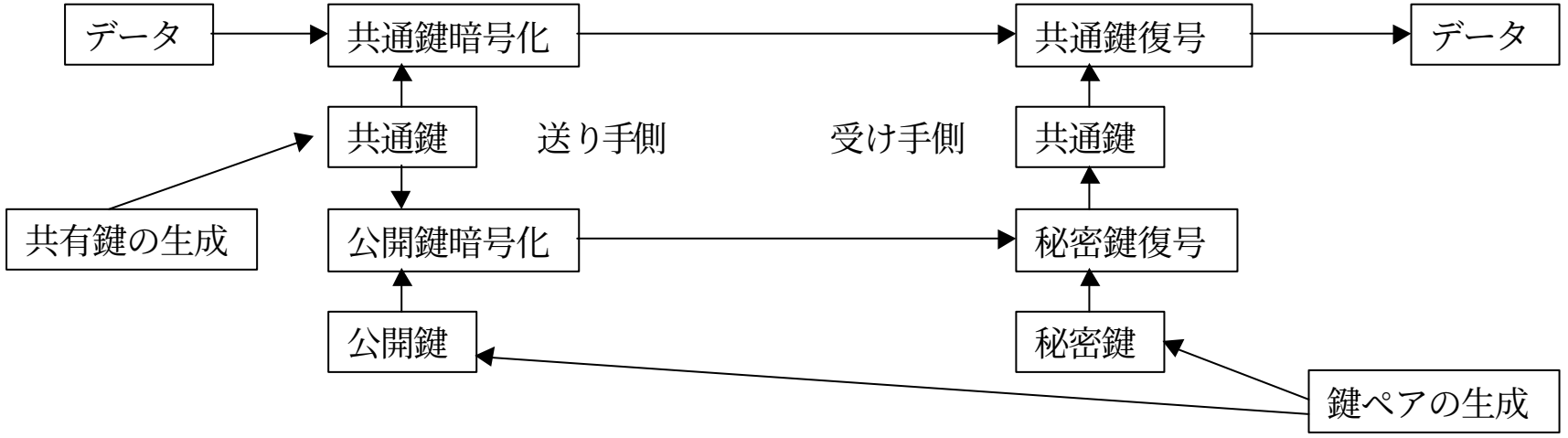
VPNの基礎技術（暗号技術）

公開鍵暗号系と共通鍵暗号系の併用

本体の暗号化は共通鍵暗号を使用（高速に大量のデータを処理）

共通鍵の交換のために、公開鍵暗号系を使用（少量のデータ）

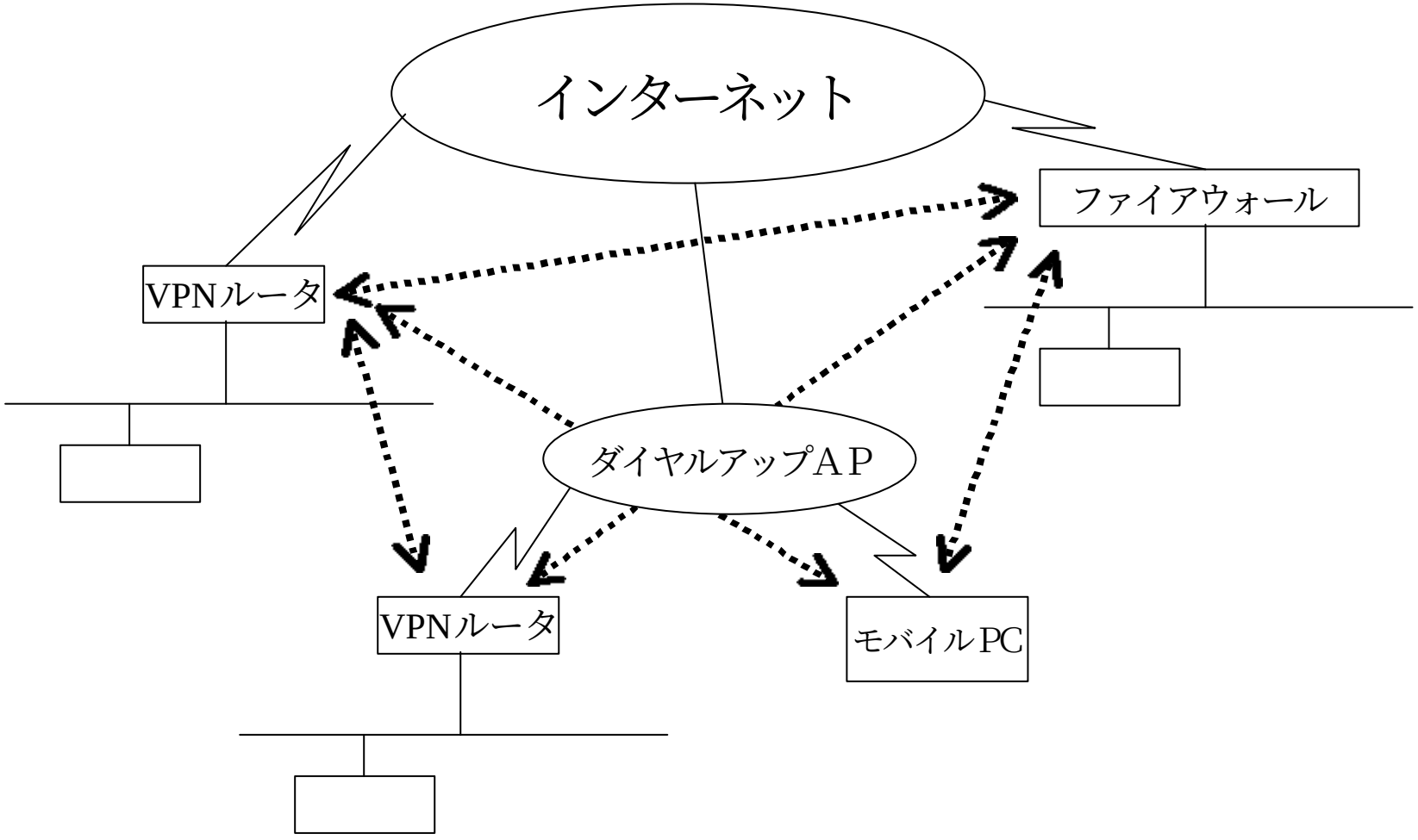
共通鍵は安全のため定期的に更新（自動的に処理可能）



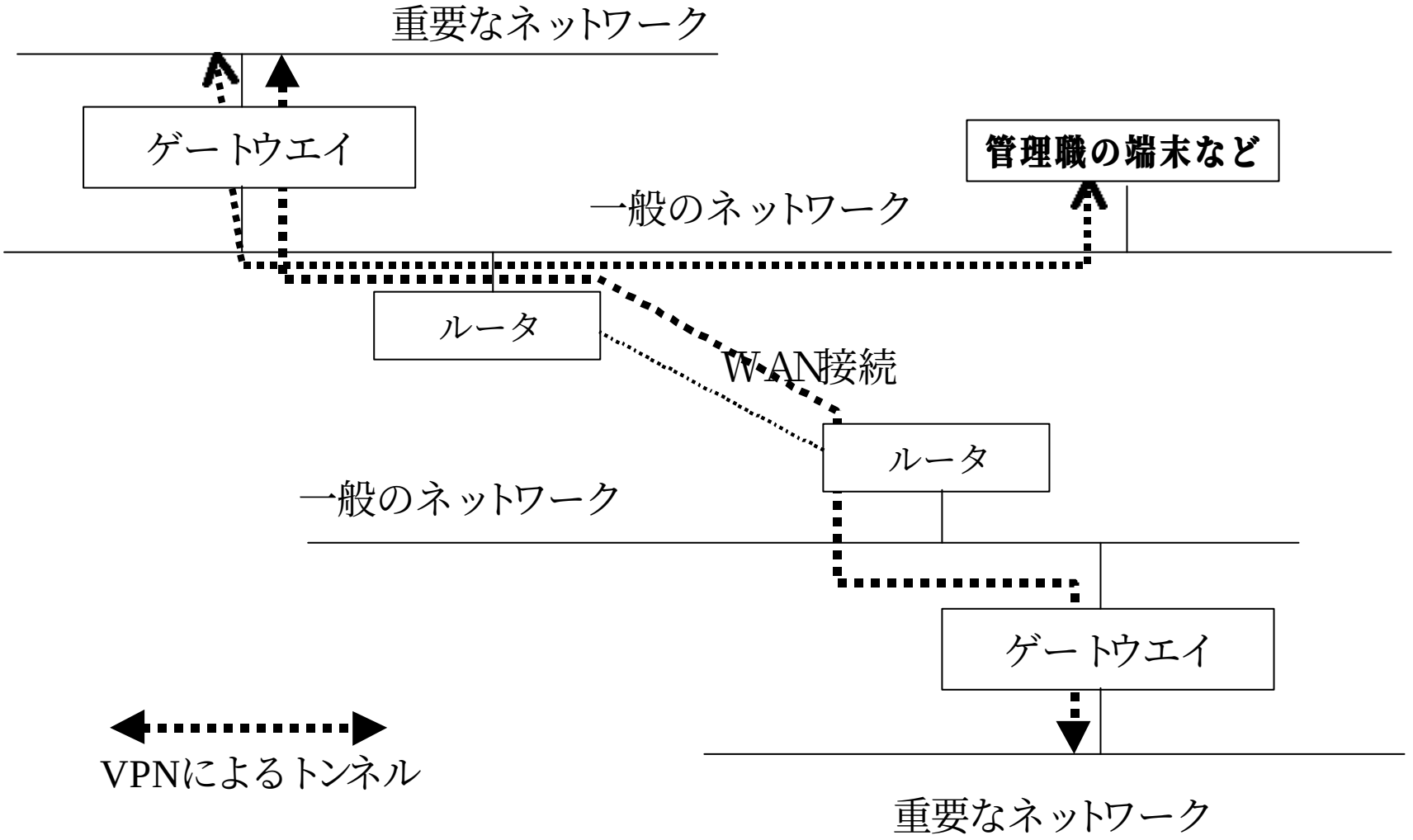
VPNの利用目的

- 同一組織のブランチ間のインターネット経由接続
  - 回線費用の節約
- モバイルアクセスのコスト削減と安全性の確保
- 複数組織の協同ネットワーク（エクストラネット）構築
  - 回線費用の節約
  - インターネットの利用による柔軟性の確保
- 組織内のネットワークセキュリティーの階層的強化
  - 組織内 LAN のセキュリティー階層化
  - セキュリティーの低いネットワークを使って重要なネットワークを接続

インターネットとVPN



組織内でのVPN利用例



ファイアウォール導入までのプロセス

- リスクアセスメント
  - どのようなリソースにどのような危険が存在するか
  - 危険度の評価（被害の想定と、対応コストの検討）
- セキュリティポリシー策定
  - どのような危険を想定し、どのように対処するかという枠組みの決定
- セキュリティシステムの設計
  - セキュリティ機器の配置と個々の設定ポリシーの定義
  - セキュリティ規則や運用方法の規定

ファイアウォールの機能的位置づけ

- 予防対策的位置づけ
  - アクセス・通過の制限
  - 通信の監視（一部のパターンのみ）と警告
  - いわば、検問所の役割
- 事後対応的位置づけ
  - 通信の記録（不正アクセスの記録）

ファイアウォールでは難しい仕事

- 通過許可した通信に対するチェックは限定的
  - アプリケーションレベルの攻撃の検知が困難
    - 攻撃シグネチャ検出は理論的には可能だが性能面で問題
- DoS（サービス妨害）的攻撃への対応
  - 大量のトラフィック負荷を伴うような妨害攻撃への有効な対策手段は少ない
- ファイアウォールに頼り切らないことが必要
  - 保護下のサーバの公開サービスの確実な管理
  - 侵入検知システム（IDS）などとの併用

ファイアウォール製品を選ぶ際に

- 操作性と安全性は別
  - Windows だから設定が簡単という 誤解」
  - UNIX だから安全という 誤解」
- プラットホームは問題ではない
  - プラットホーム(Windows/UNIX etc.)が持つGUIなどの特性よりも、目的とするポリシーをいかに設定しやすく設計してあるかがポイント
- アプライアンスかソフトウエアか
  - 導入の容易さを取るか、自由度をとるか

ファイアウォールの日常管理

- 物理的管理（物理的セキュリティ）
  - ファイアウォール設置場所の入室管理
- ファイアウォールのシステム管理
  - ログイン、設定変更記録の管理
- アラートへの対応
  - 電子メールなどによる警告への緊急対応
- ログの管理・監視
  - 定期的なログ解析やログのバックアップ

ファイアウォール自身の管理

- ファイアウォールそのものを物理的に守る
  - ファイアウォール設置場所はなるべく入室管理可能な場所にする。
  - ファイアウォール装置のログインパスワードは厳重に管理し、定期的に変更を。
  - ファイアウォールへのログインは、できればアラート対象にし、記録を確実にとる。

アラートへの対応

- アラートを受けたら必ず原因の確認を
  - 誤りも多いが、こまめに原因確認を
  - 詳細ログから警告原因を特定して対処
- アラートのチューニングも大切
  - 発生原因がわかっていて頻繁に出るアラートは出ないようにできればベター
  - 「オオカミが…」の結末にならないように

ログの管理・監視

- 最終的には、やはりログを読むしかない。
  - 市販の解析ツールの利用も有効
  - 問題発生時に原因を特定するための足ががりに
- ログの安全確保
  - root はログ改竄も可能（ファイアウォールに侵入されたら）
  - 定期的なバックアップまたは、別マシンに転送しておくほうが安全（cron による別ディレクトリへの自動コピー、syslog転送などが有効）

そのほかの付加機能

- コンテンツスキャニング
  - URLフィルタ、コンテンツフィルタ機能
  - ウイルス検出、排除機能
- 障害対策
  - ファイアウォールの二重化（Hot Stand-by/fail over）
  - 負荷（トラフィック）分散（Load balancing）
    - トラフィックの分散が必要な局面では、設定内容も複雑になりがちなことに留意する必要がある。すべてのポリシーを入り口のファイアウォールで管理することは本当に妥当だろうか。

ファイアウォールの今後（望まれるもの）

- 集中管理とポリシーベース設定機能
  - 大規模ネットワークでの複数ファイアウォールの集中管理、管理のアウトソーシングなどへの対応
- より細部の監視（アプリケーションレイヤ）
  - すべてファイアウォールでやるべきかどうかは？
- 他のツールとの連携
  - 侵入検知システム等との連動によるポート制御
- ログ解析機能、レポートの充実

まとめ

- 「ファイアウォールがあれば安心」は×
- なにができるか、できないかを正しく把握する
- 機能比較表に惑わされないこと
  - 機能の多さよりも、必要な機能があるかどうか
- 「導入すること」よりも「正しく設定・運用すること」

作成：2000/12 二木 真明（住友商事）