IPv6 Summit in TOKYO 2016

IPv6 アプリケーション開発入門

2016年11月28日

富士ソフト株式会社
技術本部 技術開発部
渡辺 露文

自己紹介

渡辺 露文（わたなべ つゆふみ）

- **所属**
  - 富士ソフト株式会社　技術本部 技術開発部
  - エキスパート（ネットワーク、セキュリティ）
- **経歴**
  - インフラエンジニアとしてDCでのシステム構築や、研究開発、社内インフラの構築・運用を経て、現在は、セキュリティ技術の調査・技術者教育、OSS利用管理などに従事
- **コミュニティでの活動**
  - IPv6普及高度化推進協議会
    - アプリケーションのIPv6対応検討SWG
    - IPv6導入に起因する問題検討SWG
  - 脅威分析研究会
  - 脆弱性診断研究会

Introduction

Do you know …

IPv6 ?

Internet Protocol version 6

インターネットの通信に関する規約（RFC791）
IPネットワークに接続するには1つ以上のIPアドレスが必要
皆さんが馴染んでいるのはIPv4（例：10.1.2.3）

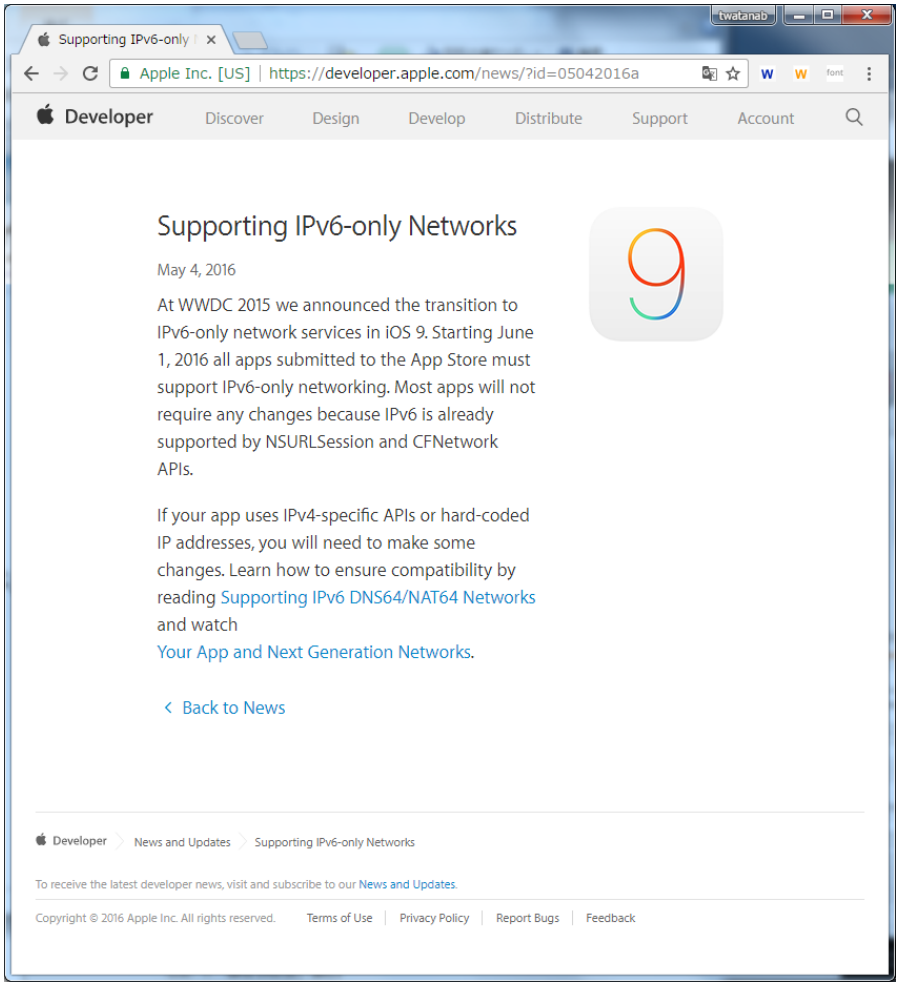
もっとも認識すべきこと

IPv4とIPv6は互換性がない

IPv4 同士、IPv6 同士は通信可能だが、
IPv4 と IPv6 は直接通信できない

今年5月 Apple のアナウンス

- **2016/6/1 から、App Store に載せるアプリは、IPv6-only ネットワークで動作しないといけない**
  - **ほとんどのアプリは何も変更しなくて大丈夫なはず**
  - もし、IPv4固有のAPIやIPアドレスをハードコードしてたら、（Networking Overview の）「Supporting IPv6 DNS64/NAT64 Networks」を読んで対応してね



https://developer.apple.com/news/?id=05042016a

今年9月 マイクロソフトのアナウンス

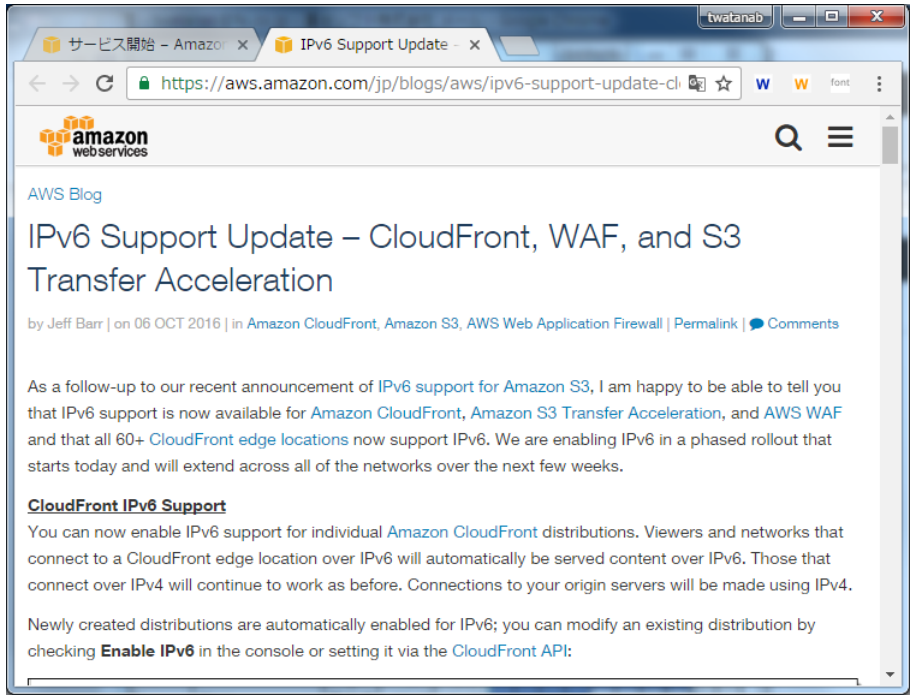
- **AzureがIPv6に対応！**
  - ネイティブでIPv6をサポート
  - 対象
    - ロードバランサー
    - Azure VM
      - デュアルスタック

https://docs.microsoft.com/ja-jp/azure/load-balancer/load-balancer-ipv6-overview

今年8月および10月の AWS のアナウンス

- **IPv6サポート開始**
  - Amazon S3, S3 Transfer Acceleration
  - CloudFront
  - WAF

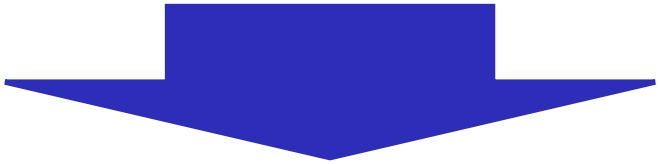
https://aws.amazon.com/jp/blogs/news/now-available-ipv6-support-for-amazon-s3/



https://aws.amazon.com/jp/blogs/aws/ipv6-support-update-cloudfront-waf-and-s3-transfer-acceleration/

最近の流れ

- **今年、大御所の IPv6 対応／ IPv6 対応義務化が目立った**
  - Apple： iOS App Store 登録アプリのIPv6対応義務化
  - Microsoft： Azure VM およびロードバランサの IPv6 サポート
  - AWS：S3, CloudFront のIPv6サポート



アプリケーションも IPv6 対応で作る時代が到来

Summary

- **このセッションで伝えたいこと：**
  基本方針とアプリケーションIPv6対応 3原則 に沿って対応を！

- IPv6対応の前提
  1. IPアドレス直書き禁止！FQDNを使用する
  2. IPアドレスでユーザを識別しない
  3. 同時に多数のセッションを張らない

- **基本方針**
  - **IPv6対応＝IPv6/IPv4の両方で動作させること**
  - **シングルソースコードで対応する**

- **3原則**
  1. **IPv4/IPv6両対応のプログラミング言語と実行環境を使う**
  2. **通信処理をIPv4/IPv6の両方に対応させる**
  3. **データとしてIPアドレスを扱う箇所をIPv4/IPv6の両方に対応させる**

Agenda

1. **アプリケーションIPv6対応 その前に**

2. **IPv6対応アプリケーション開発入門**
   1. プログラミング言語と実行環境
   2. 通信処理の対応
   3. データとして扱う箇所の対応

1. アプリケーションIPv6対応 その前に

とあるソースコード

- このコード、イケてない…

```
use IO::Socket::IP;
$host = “198.51.100.1”;
:
my $sock = IO::Socket::IP->new(
 PeerAddr => $host,
 PeerPort => $port,
 Proto => 'tcp'
) or die “Error: $!¥n”;
:
```

とあるソースコード

- このコード、イケてない…

```
use IO::Socket::IP;
$host = “198.51.100.1”;
:
my $sock = IO::Socket::IP->new(
 PeerAddr => $host,
 PeerPort => $port,
 Proto => 'tcp'
) or die “Error: $!¥n”;
:
```

えっ、IPアドレス直書き？

IPアドレス直書きすると、アドレス変更時に

- 修正が必要
- 再テストも必要

他にもイケてないコードがある

- とある Androidプログラミング書籍におけるソケット通信のサンプルコード

```
public class SocketEx…
…
…
        private final static String IP=“192.168.11.12”;//★変更必須
```

良い子はマネしちゃダメ

1. アプリケーションIPv6対応 その前に

IPアドレスの誤用を防ぐ(1)

前提1
IPアドレス直書き禁止！FQDNを使用する

- **IPアドレス直書きの場合、IPv6対応に関わらず、IPアドレス変更時に修正が必要となる**
- **ハードコーディング厳禁！**
  - ハードコーディングの場合は要コンパイル、再テスト、再配布…

ダメ。ゼッタイ。

- **IPアドレスの管理はインフラに任せるべき**
  - 最大限OS環境を利用
  - 独自実装は不具合を生みやすい

1. アプリケーションIPv6対応 その前に

なぜIPアドレス直書きがダメなのか?

	目的	変更・改修の理由
アプリケーション	**機能の提供**	■ 業務要件の変更 ■ サービス内容の変更 ■ ユーザビリティ向上　…,etc.
インフラ	**資源の提供**	■ 資源管理（**IPアドレス**、サーバリソース…） ■ 性能

同一システムでも変更・改修の理由・時期は異なる

互いに変更の影響を受けるべきではない

アプリケーションは、IPアドレスに依存すべきではない

1. アプリケーションIPv6対応 その前に

IPアドレスの誤用を防ぐ…(2)

前提2
IPアドレスでユーザを識別しない

- **IPアドレスはユーザに割当てるべきものではない**
- **IPアドレスから端末を一意に特定できない**
  - IPv6に限らず、1台の端末で複数のアドレスを使用することが当たり前になっている（4G + WiFiなど）
  - 大規模NATに限らず、NATにより複数の端末に同一のIPアドレスが割当てられることが当たり前になっている
- **端末からユーザを一意に特定できない**

1. アプリケーションIPv6対応 その前に

IPv6へ移行しない環境への配慮

前提3
同時に多数のセッションを張らない

- **今後、IPv6化が行われないアクセス網で用いられる大規模NAT（CGN）では、一人あたりのセッション数が少なくなるため、同時に張るセションは最小限に留めるべき**
  - 特にモバイル通信網が顕著になることが予想される
- **むしろセッション/コネクションを使い回すのが最近の流れ**
  - HTTP/2, Web Socket

CGN（Carrier-Grade NAT）：
インターネットサービスプロバイダ(ISP)などの電気通信事業者が、自社内のネットワークと他社のネットワークの分界点付近でネットワークアドレス変換(NAT)を行うこと。

各プラットフォームの公式ドキュメントを参照

- **Microsoft環境**
  - MSDN
    - https://msdn.microsoft.com/library
- **Apple**
  - 日本語ドキュメント – Apple Developer
    - https://developer.apple.com/jp/documentation/
  - **ネットワーク周りは「Networking Overview」**
    - **IPv6（DNS64/NAT64）への対応方法も丁寧に書かれている**
    - https://developer.apple.com/jp/documentation/NetworkingInternetWeb/Conceptual/NetworkingOverview/Introduction/Introduction.html
- **Android**
  - Android デベロッパー
    - https://developer.android.com/index.html

2. アプリケーションIPv6対応概要

もっとも認識すべきこと

IPv4とIPv6は互換性がない

IPv4 同士、IPv6 同士は通信可能だが、
IPv4 と IPv6 は直接通信できない

- **送信側、受信側、中継者のそれぞれが対応する必要がある**
- **システム、サービスとして対応するためには、アプリケーションも対応しないといけない**

アプリケーションIPv6対応の基本方針

IPv6対応　＝
IPv4とIPv6の両方で動作する

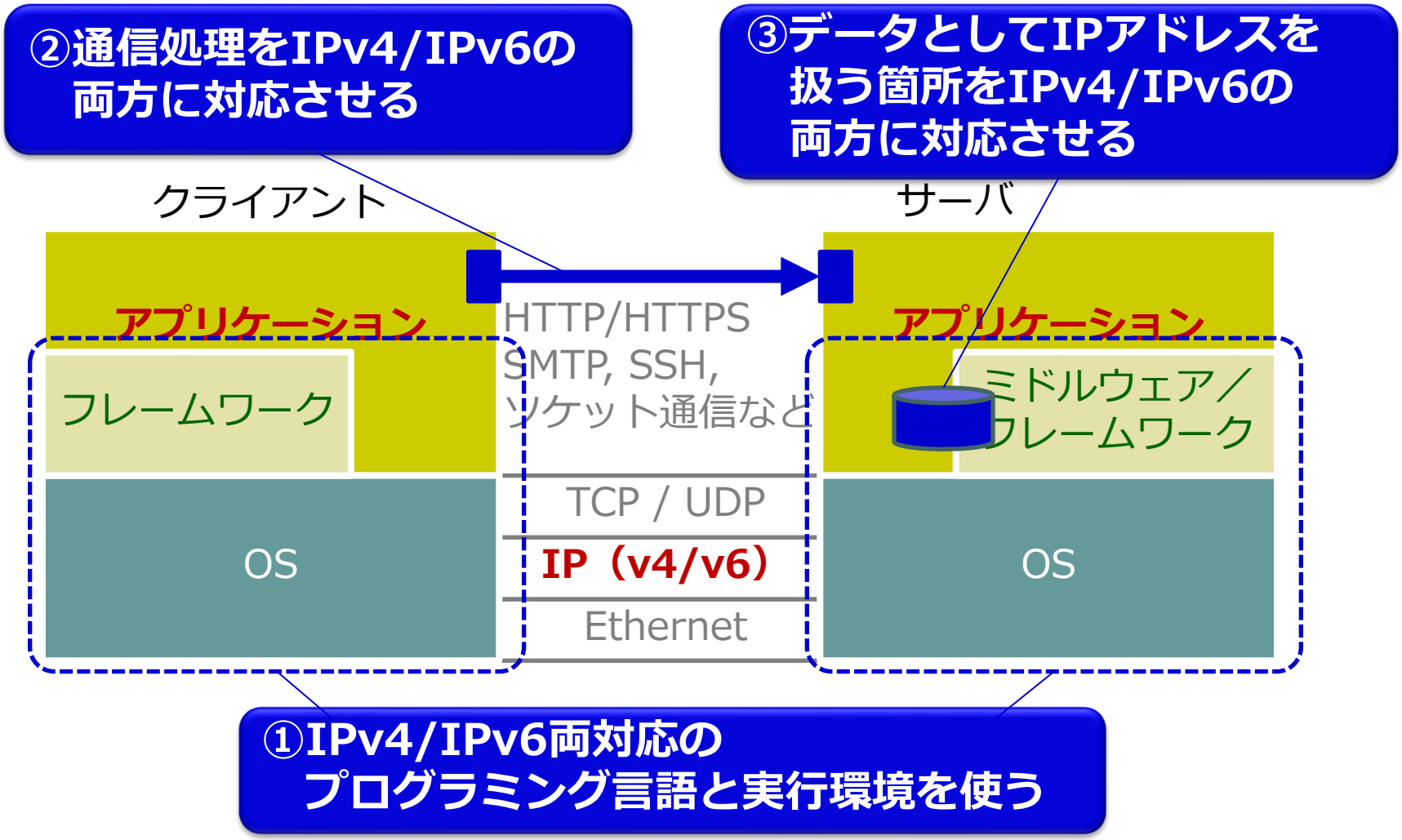
- IPv6とIPv4の共存期間が長く続く
- これまでIPv4で提供してきたサービスは、今後も継続してIPv4でも動作する必要あり

シングルソースコードで対応

- 各開発言語が概ねIPv6に対応しており、プロトコルによって開発言語を分ける必要がなくなった
- アプリケーションのメンテナンス性を重視し、プロトコルによって機能差異が生じることを未然に防ぐ

2. アプリケーションIPv6対応概要

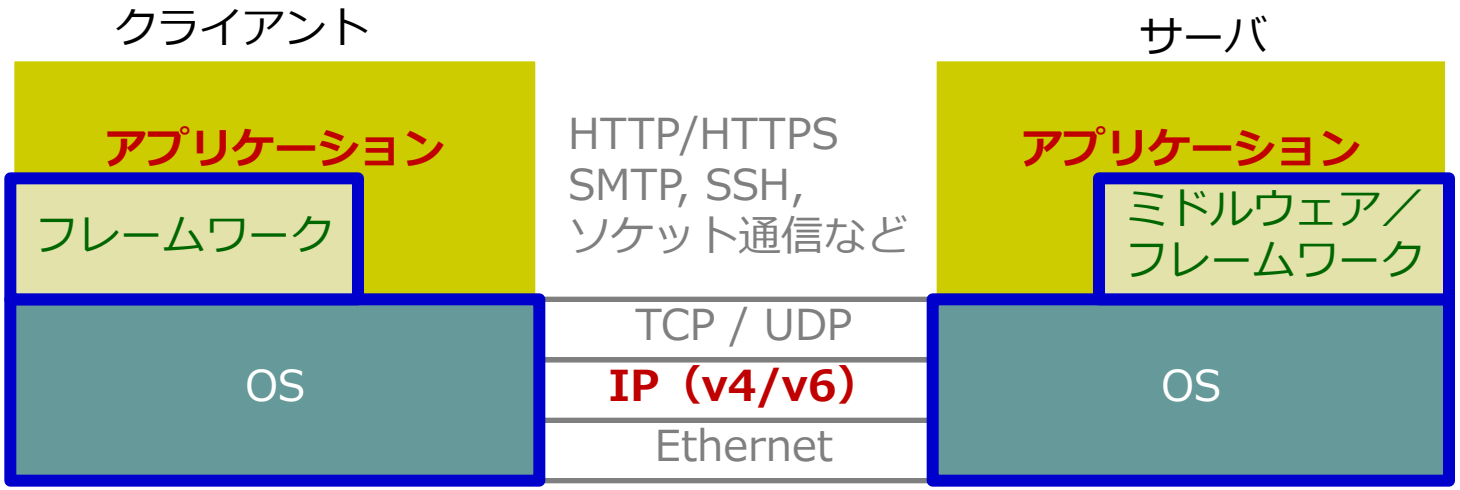
アプリケーションIPv6対応のポイント



2-1. プログラミング言語と実行環境

原則1
IPv4/IPv6両対応のプログラミング言語と実行環境を使う

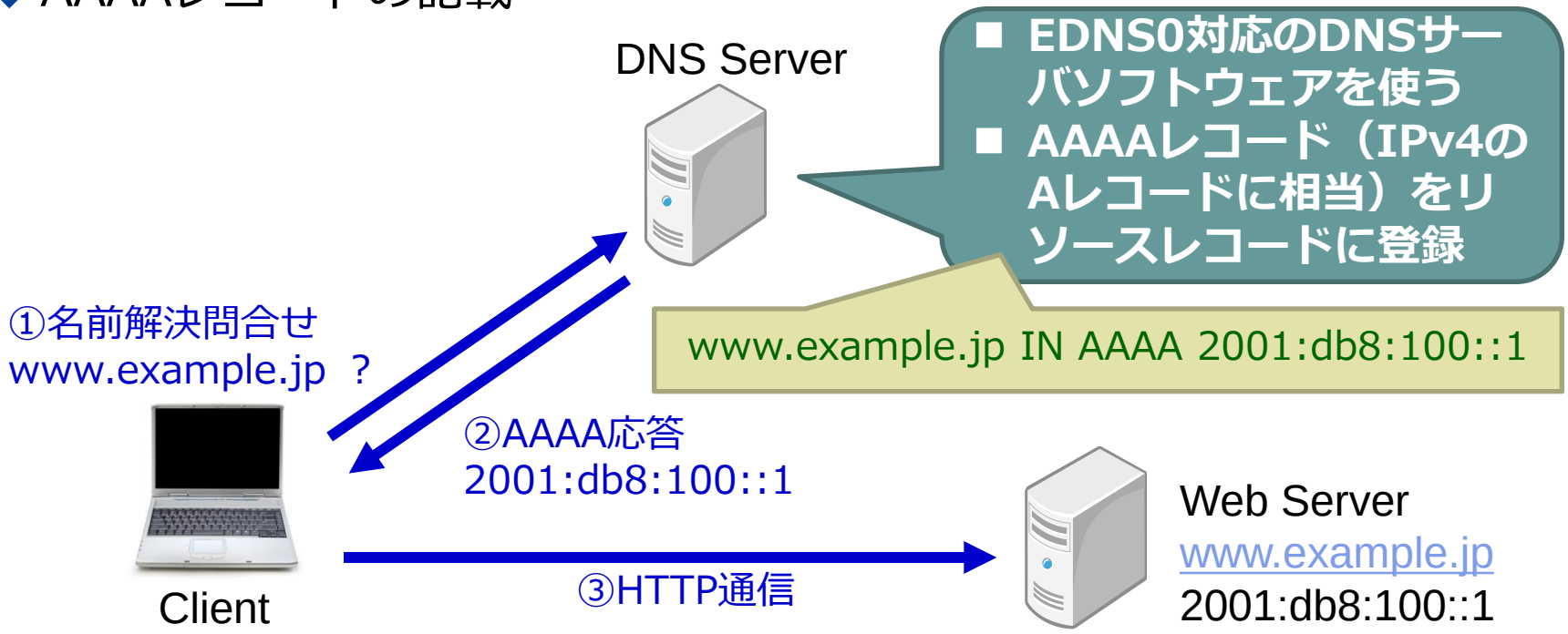
- **プログラミング言語と実行環境に求められること：**
  IPv4/IPv6両方で通信できる
  1. 名前解決でIPv4/IPv6両方のアドレスが扱える
  2. IPv4/IPv6両方で接続できる



2-1. プログラミング言語と実行環境

IPv6とDNS

- **アプリケーションのIPv6環境での動作には、DNSのIPv6対応が不可欠**
- **IPv6アドレスの名前解決に必要なこと**
  - EDNS0対応のDNSサーバ／クライアント
  - AAAAレコードの記載



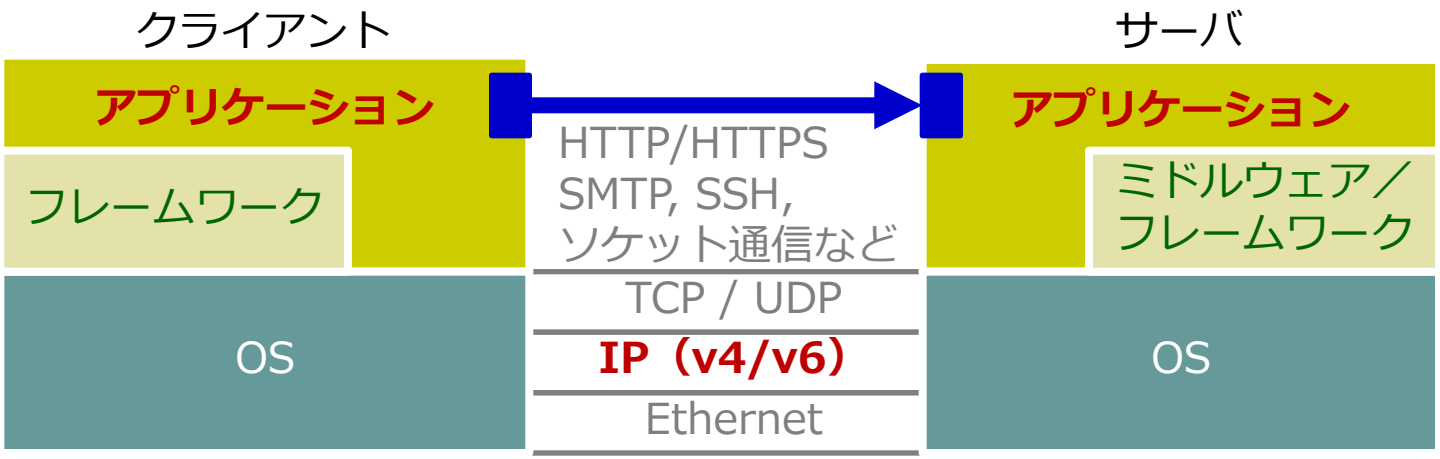
プログラミングにおける留意点

- **IPv4/IPv6の双方に対応するライブラリ、オブジェクト、関数、データ型を使う**
  - 従来（IPv4のみ）のものとは別に用意されていることがある
    - C addrinfo構造体、getaddrinfo()
    - Java InetAddressクラス
    - Perl IO::Socket::IP　　など
- **アドレス検証、変換などはライブラリを有効活用**

2-2. 通信処理の対応

原則2
通信処理をIPv4/IPv6の両方に対応させる

- **通信処理のIPv6対応で行うべきこと**
  - 接続の試行順序を理解する
  - IPv4/IPv6いずれかが通信できない状況を想定する
  - アルゴリズムを見直す
    - クライアントプログラム：フォールバック時の迅速な切り替え
    - サーバプログラム：IPv4/IPv6 両プロトコルでの接続を処理



2-2. 通信処理の対応

接続の試行順序

■ IPv4/IPv6の両方で接続可能ということは…

◆ 接続の試行順序がある

接続の優先順位： IPv6 ＞ IPv4

- RFC3484およびRFC6724で定義されるアドレス選択の優先順位に従う
- ポリシーテーブルをカスタマイズすることにより変更可能

◆ 接続失敗時には別の宛先アドレスに切替えて接続する
（＝**フォールバック**）

- アプリケーションの作りが悪いと…
  - 切り換えに時間がかかる
  - 正常に切り替わらないこともある

ユーザの利便性を損なう

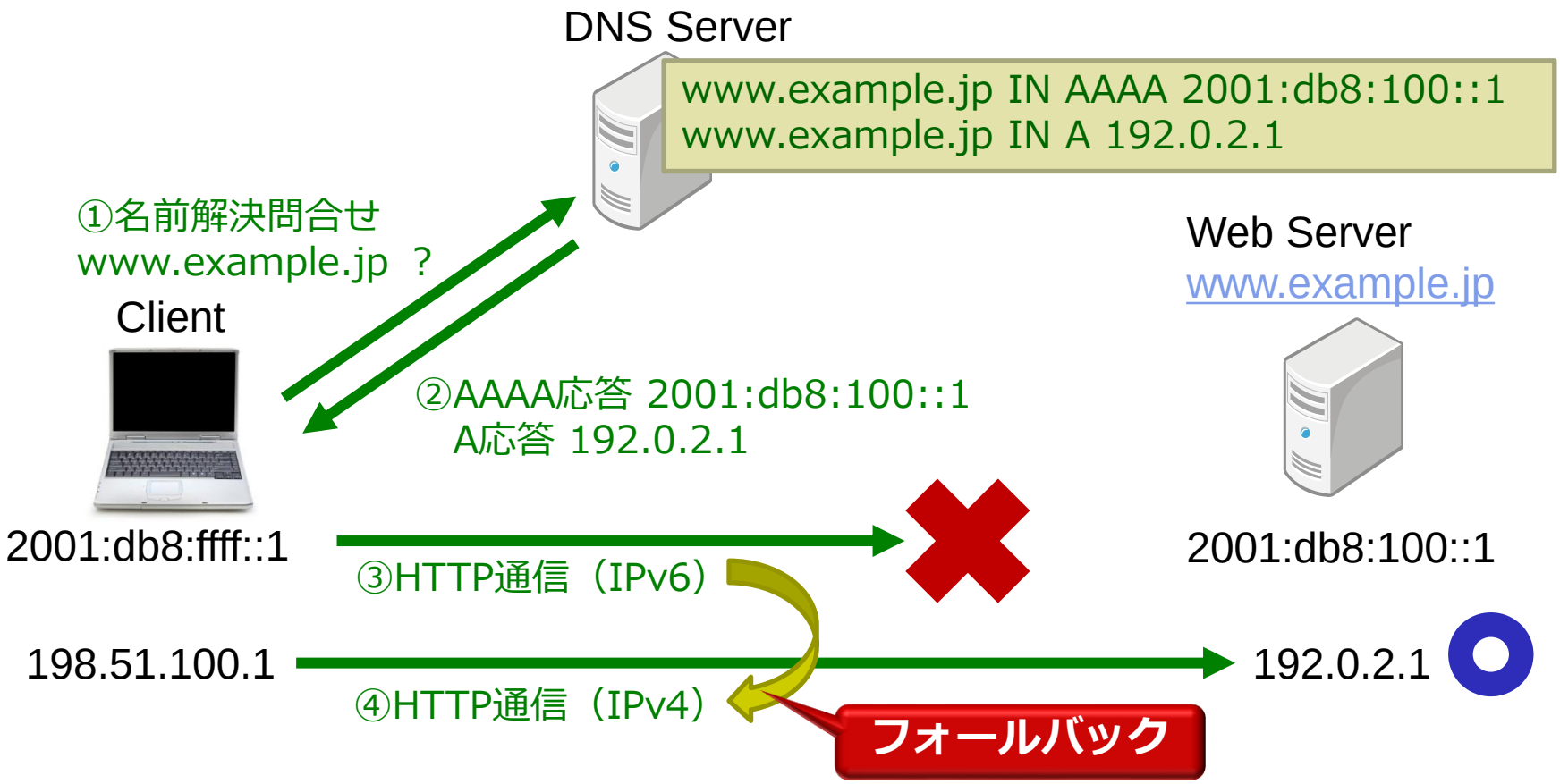
フォールバック問題

いずれかが接続できない状況も想定に入れ、接続失敗時に接続先を迅速に切替えて接続する作りにする

2-2. 通信処理の対応

フォールバック(1)

- **接続できない場合に別の接続先への接続に切替える動作**
  - IPv6⇒IPv4 ／ IPv6⇒IPv6 ／ IPv4⇒IPv4 ／ IPv4⇒IPv6



2-2. 通信処理の対応

フォールバック(2)

想定される主な原因

- サーバが当該のサービスを提供していない【サーバ側の問題】
  - DNS誤登録、障害等
- ネットワークの接続性が失われている【経路の問題】
  - グローバルアドレスを利用している閉域網等
- サーバへの到達性がないアドレスで通信を行おうとしている【クライアント側の問題】

予防策

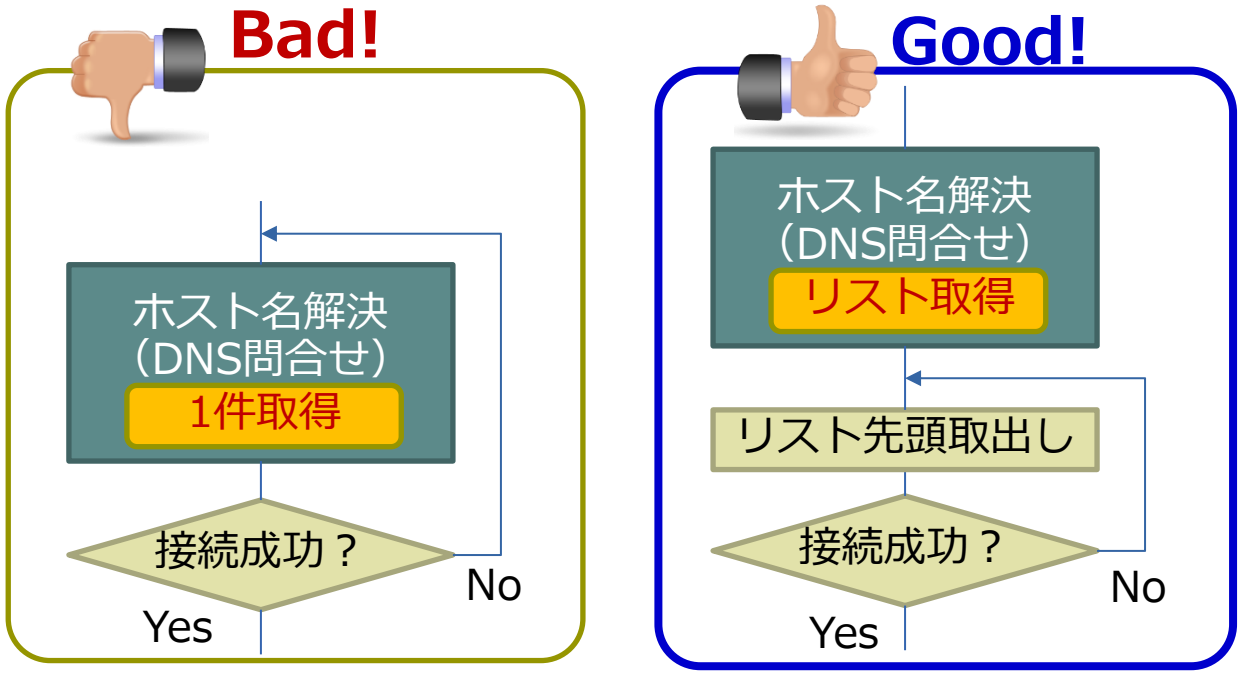
- 設定の不備を修正する
  - サービスを提供していないIPアドレスをDNSに登録しない
  - サービスを適切に提供する
- ネットワークの接続性を健全に保つ
- ポリシーテーブルをカスタマイズする
  - ポリシーテーブル：アドレス選択の優先順位を制御するテーブル

2-2. 通信処理の対応

クライアントプログラム

■ フォールバックを迅速に行う

◆ ホスト名の名前解決結果をリスト形式で取得し、アドレスリストの順に接続を試み、接続が確立したものと送受信を行う（RFC6724に準拠させる）



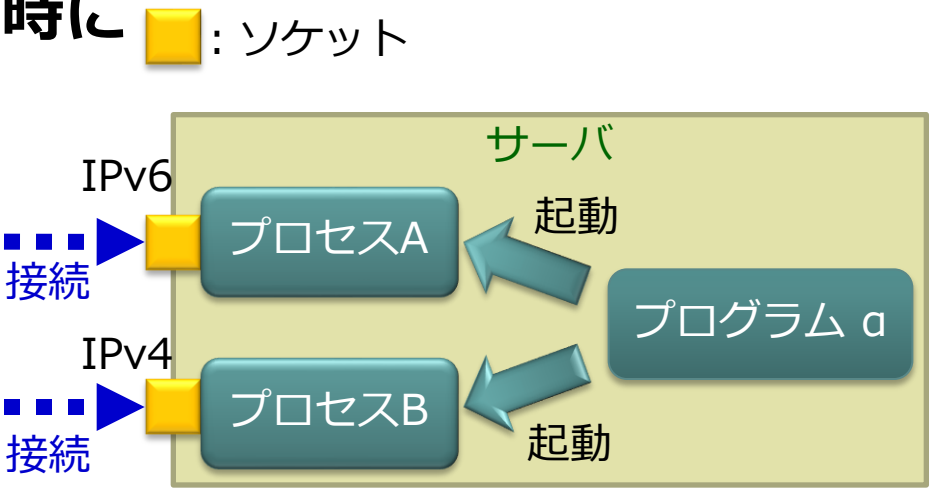
◆ 更に迅速にフォールバックを行うために、Happy Eyeballs（RFC6555）も選択肢の一つ

2-2. 通信処理の対応

サーバプログラム

IPv4/IPv6両方の接続を同時に受付ける2種類の手法

1. 2プロセス型
   - 同一プログラムをIPv4用、IPv6用それぞれ起動
   - 既存プログラムのアルゴリズムは変更せず起動時にIPv4/IPv6を選択
   - 例：vs-ftpd



2. 1プロセス型
   - 単一のプロセスでIPv4/IPv6両方の接続を処理
   - 接続待受を並列化
   - 例：Apache HTTP Server, Postfixなど

2-2. 通信処理の対応

iOSアプリ、Androidアプリでは…

- **プラットフォームが提供するAPIでこの辺を対応**
  ⇒ 開発者に、この辺を意識させない

- **iOSアプリ**
  - 高レベルネットワークフレームワークの使用を推奨
    - 「Networking Overview」の中で明記
    - WebKit：Webページを読込む複雑なプロセスに対応
    - Cocoa URL：アプリケーションでURLと参照先のリソースを操作
    - CFNetwork.Core　Services：さまざまなネットワークタスク

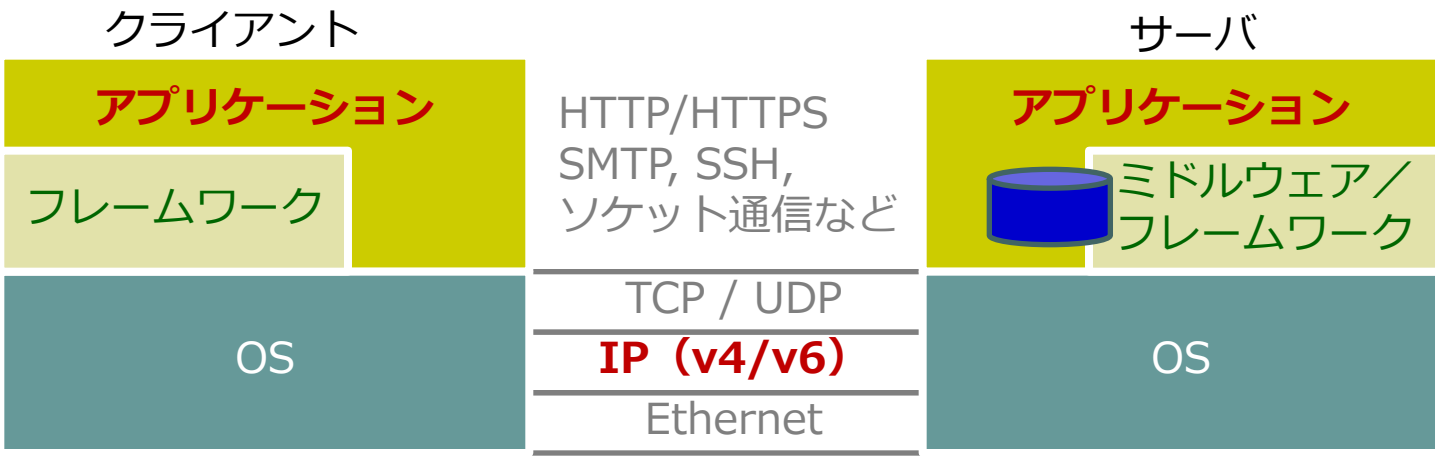
- **Androidアプリ**
  - Web：WebView（Android.webkit.WebView）、
    HttpURLConnection（java.net.HttpURLConnection）
  - Web以外：Socket（java.net.Socket）

2-3. データとして扱う箇所の対応

原則3
データとしてIPアドレスを扱う箇所をIPv4/IPv6の両方に対応させる

◆データとして扱う箇所

- 入力
- 格納
- 出力
- 検索
- ソート



2-3. データとして扱う箇所の対応

IPv4/IPv6アドレス比較

RFC5952に従い表記する

		IPv4アドレス	IPv6アドレス
アドレス長		32bit	**128bit**
文字列表記	表記法	8bitずつ区切り、10進数で表記	16bitずつ区切り、**16進数**で表記
	区切り文字	.（ドット）	**:**（コロン）
	文字列長	15文字以内	**39文字以内**

サブネットマスク／プレフィックス長を考慮すると、上記＋”/”＋数字3文字

例外あり（ゾーンID・スコープID、射影アドレス）

■ **例**

IPv4）192.0.2.1

IPv6完全表記）2001:db8:0000:0000:0001:0000:0000:0001

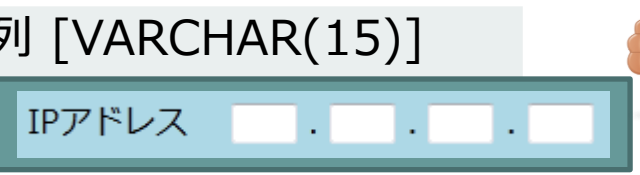
IPv6省略表記）2001:db8::1:0:0:1

2-3. データとして扱う箇所の対応

IPアドレスの入力・格納

文字列で入力・格納

IPv4のみ	15文字以内の文字列 [VARCHAR(15)] もしくは整数×4
IPv4/IPv6両対応	39文字以内の文字列 [VARCHAR(39)]



Bad!

Good!

入力値の検証はライブラリの関数・フィルタを利用

- 例）PHP Net_IPv6::checkIPv6();
  （PEARにて提供されるNet_IPv6パッケージに含まれる）

ネットワークアドレス用のデータ型があれば、それを利用

- データ型とセットで関数が用意されていると更に便利
- 例) PostgreSQLのネットワークアドレス型

2-3. データとして扱う箇所の対応

出力への影響と検索・パターンマッチング

出力への影響

- 体裁を整える出力やデータ加工では注意が必要
  - アドレス部分の文字列長が長くなる
  - アドレスの区切り文字が変わる

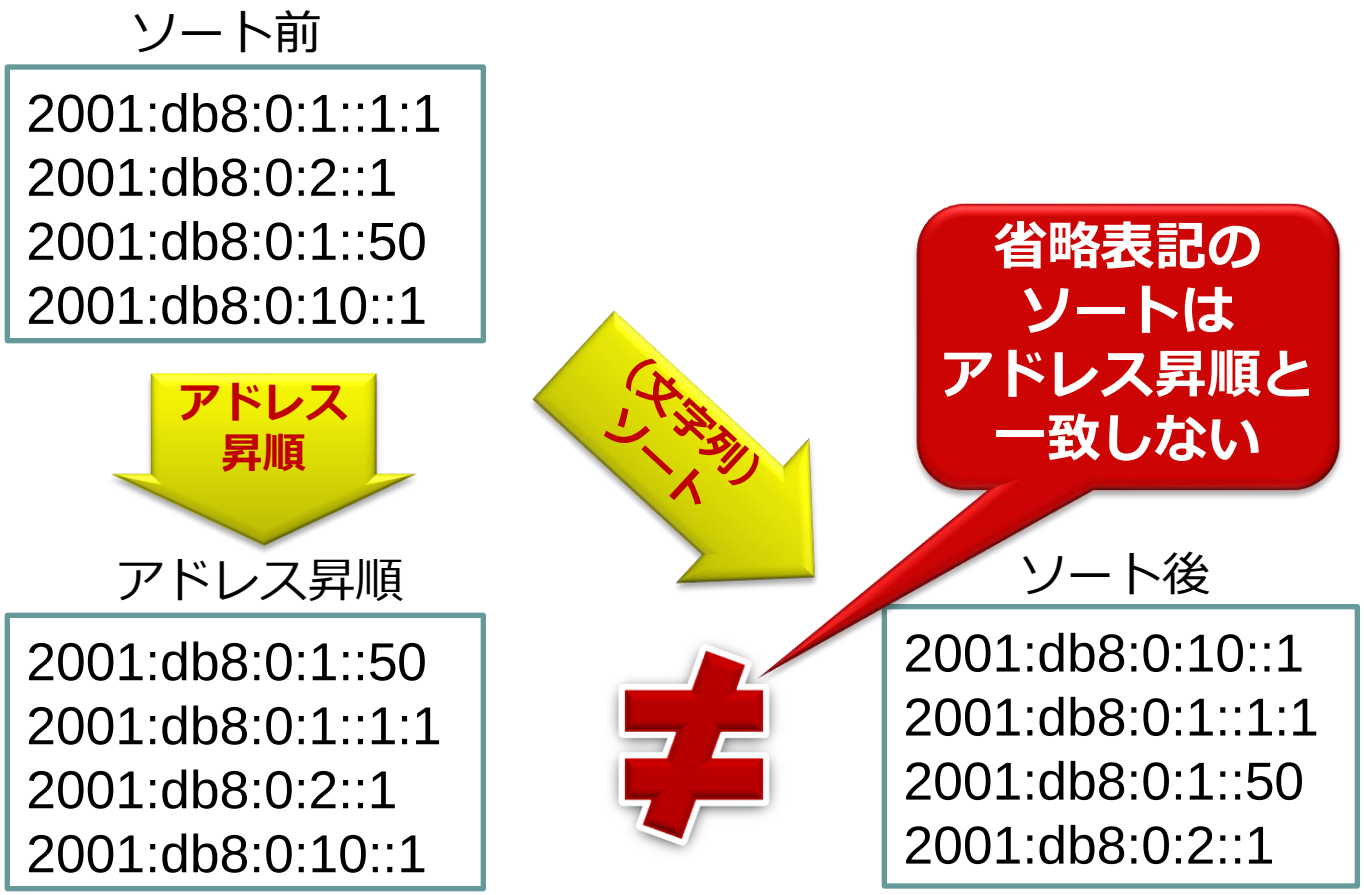
検索・パターンマッチング

- 検索文字列と検索対象を、省略表記もしくは完全表記で統一する
  - 省略表記⇔完全表記の変換はライブラリを有効活用する
- 過去に開発されたシステム・ツールでは、RFC5952に準拠しない省略表記が存在しうるので要注意

2-3. データとして扱う箇所の対応

ソート(1)

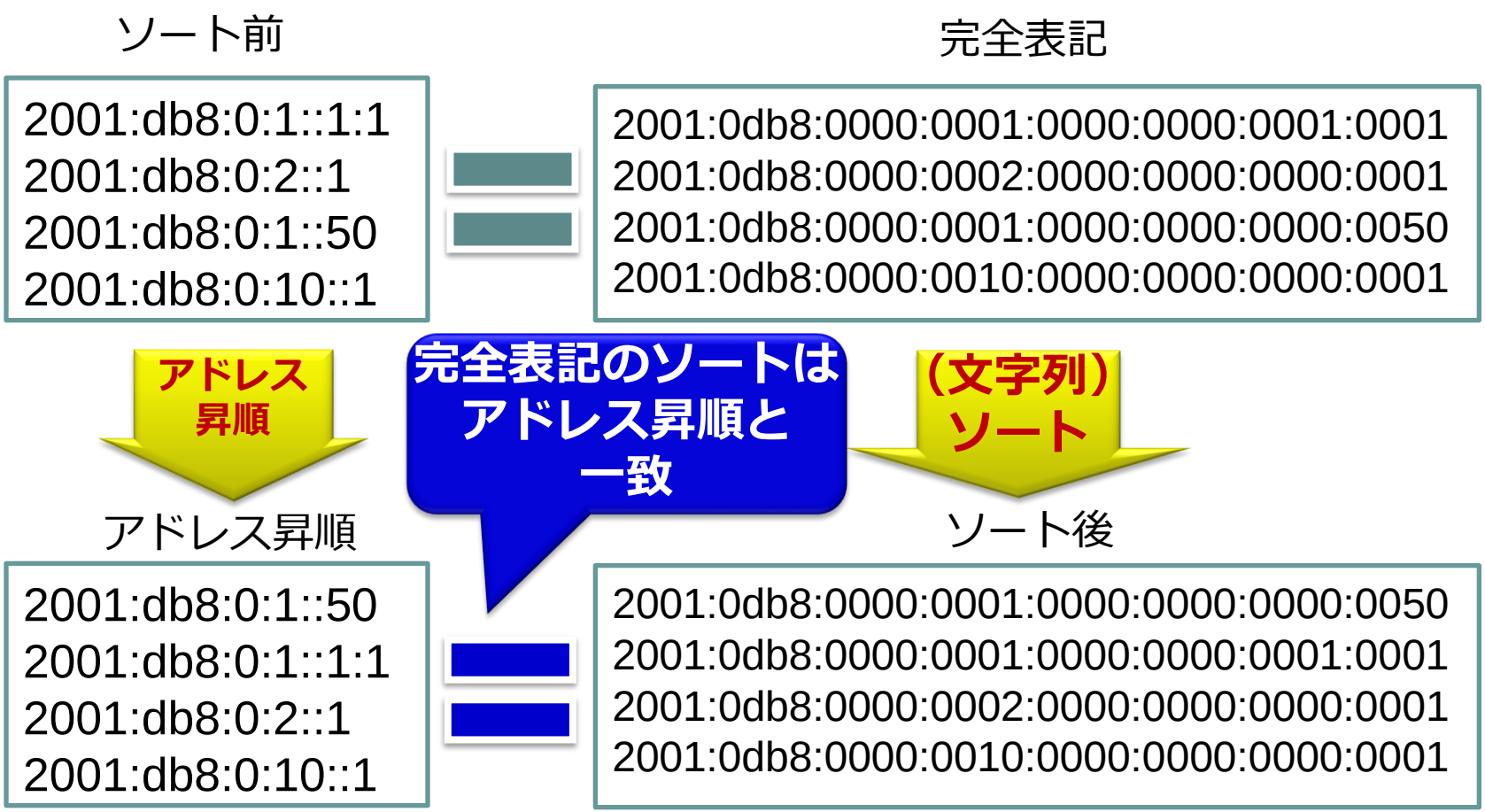
■ **推奨表記のままソートしてもアドレス昇順にはならない**



2-3. データとして扱う箇所の対応

ソート(2)

■ ソートは完全表記で行う



まとめ

まとめ

- 基本方針とアプリケーションIPv6対応 3原則 に沿って対応を！

- IPv6対応の前提
  1. IPアドレス直書き禁止！FQDNを使用する
  2. IPアドレスでユーザを識別しない
  3. 同時に多数のセッションを張らない

- IPv6対応の基本方針
  - ◆ IPv6対応＝IPv6/IPv4の両方で動作させること
  - ◆ シングルソースコードで対応する

- IPv6対応の3原則
  1. IPv4/IPv6両対応のプログラミング言語と実行環境を使う
  2. 通信処理をIPv4/IPv6の両方に対応させる
  3. データとしてIPアドレスを扱う箇所をIPv4/IPv6の両方に対応させる

つづきはWebで（参考文献）

- 「アプリケーションのIPv6対応ガイドライン 基礎編」／IPv6普及・高度化推進協議会 IPv4/IPv6共存WG アプリケーションのIPv6対応検討SWG
  - http://www.v6pc.jp/jp/entry/wg/2012/12/ipv610.phtml

- 「アプリケーションのIPv6対応ガイドライン Webアプリ編（案）」／IPv6普及・高度化推進協議会 IPv4/IPv6共存WG アプリケーションのIPv6対応検討SWG
  - http://www.v6pc.jp/jp/entry/wg/2014/06/ipv6web.phtml