



Internet Week 2013 T2 アプリケーション・サービスのIPv6対応

# アプリケーションのIPv6対応概要

---

2013年11月26日

富士ソフト 株式会社  
技術本部 技術開発部

渡辺 露文

# About me

## ■ 渡辺 露文 (わたなべ つゆふみ)

◆ 富士ソフト株式会社 技術本部技術開発部 所属

### ◆ 業務経歴

- 1999年 富士ソフトABC株式会社 (現 富士ソフト株式会社) 入社
- 入社後、ISP、データセンター顧客向けシステムなどB2Cのシステム開発・インフラ構築・運用に従事
- 2008年～ 社内システムのインフラ企画・構築・運用に従事
- 2011年～ 技術調査および社内技術者教育に従事

### ◆ 主な社外活動

- 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
- 技術評論社 Software Design  
にて連載中 (共同執筆)



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  
 廣海 緑里 HIROMI Ruri 渡辺 露文 WATANABE Tsuyufumi 新 善文 ATARASHI Yoshifumi 藤崎 智宏 FUJISAKI Tomohiro



# Agenda

---

## 1. アプリケーションIPv6対応 その前に

## 2. アプリケーションIPv6対応概要

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



# Introduction

---

- 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 6原則 に沿って対応を！
  
- 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
  
- 6原則
  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
  4. IPv4/IPv6両対応のプログラミング言語と実行環境を使う
  5. 通信処理をIPv4/IPv6の両方に対応させる
  6. 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

# 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



---

# 1. アプリケーションIPv6対応 その前に

## 1. アプリケーションIPv6対応 その前に

# アプリケーションとインフラ

## ■ アプリケーションとインフラではライフサイクルが異なる

- ◆ 同一システムでも変更・改修の理由・時期は異なる

	変更・改修の理由
アプリケーション	<ul style="list-style-type: none"> <li>■ 業務要件の変更</li> <li>■ サービス内容の変更</li> <li>■ ユーザビリティ向上 …,etc.</li> </ul>
インフラ	<ul style="list-style-type: none"> <li>■ 資源管理 (<b>IPアドレス</b>、サーバラック…)</li> <li>■ 性能</li> </ul>

- ◆ 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アドレスに依存しないために…(1)

## 原則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…
- 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

## IPアドレスに依存しないために…(2)

### 原則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（LTE + 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

### 原則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、同時に張るセッ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

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。



---

## 2. アプリケーションIPv6対応概要

## 2. アプリケーションIPv6対応概要

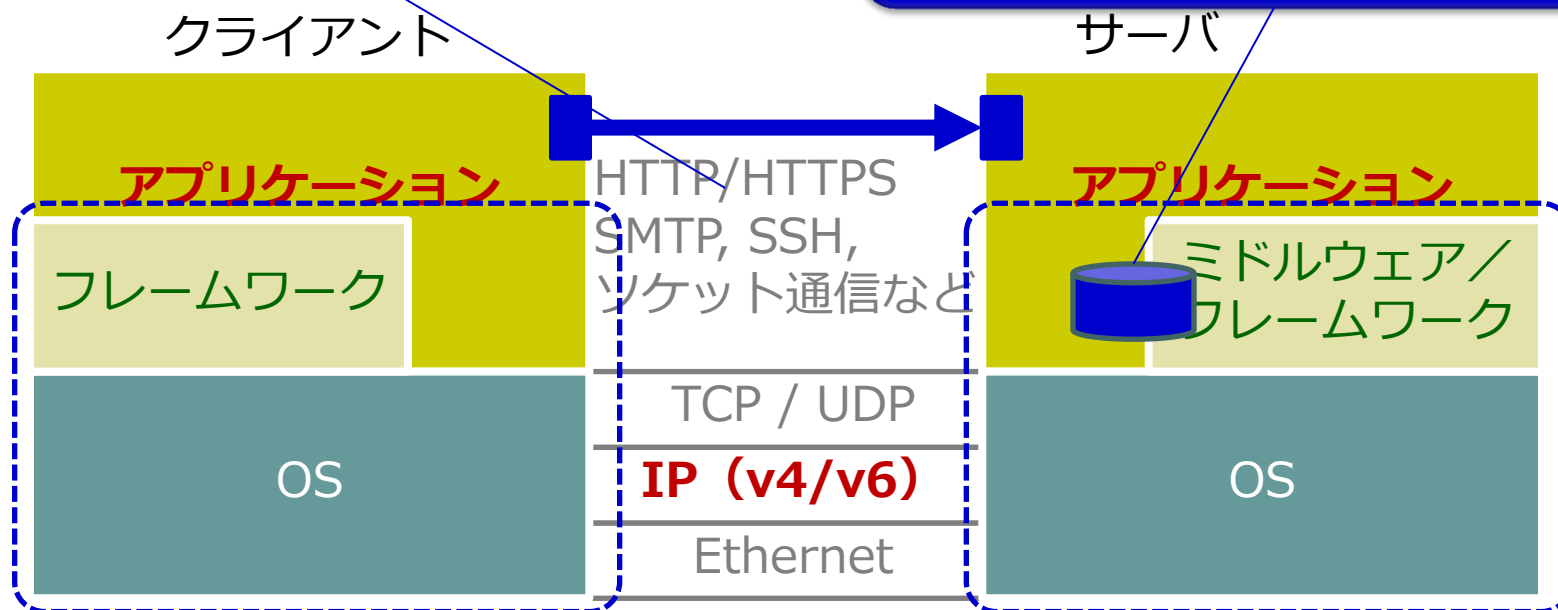
# アプリケーションIPv6対応のポイント

## 原則5

② 通信処理をIPv4/IPv6の両方に対応させる

## 原則6

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



## 原則4

① IPv4/IPv6両対応のプログラミング言語と実行環境を使う

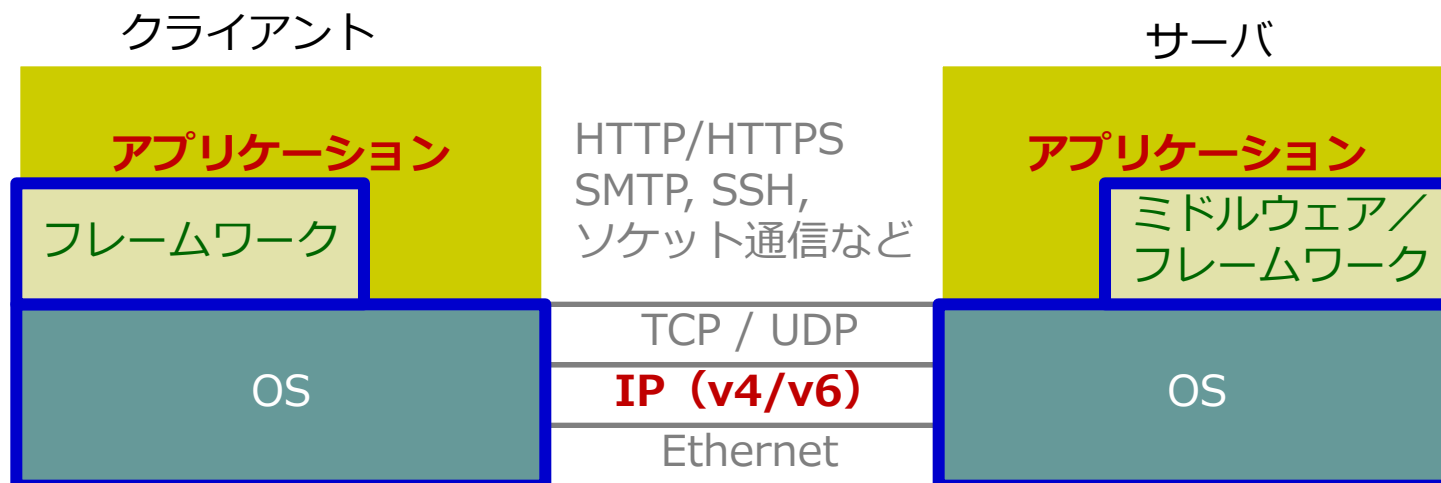
## 2-1. プログラミング言語と実行環境

### 原則4

### 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

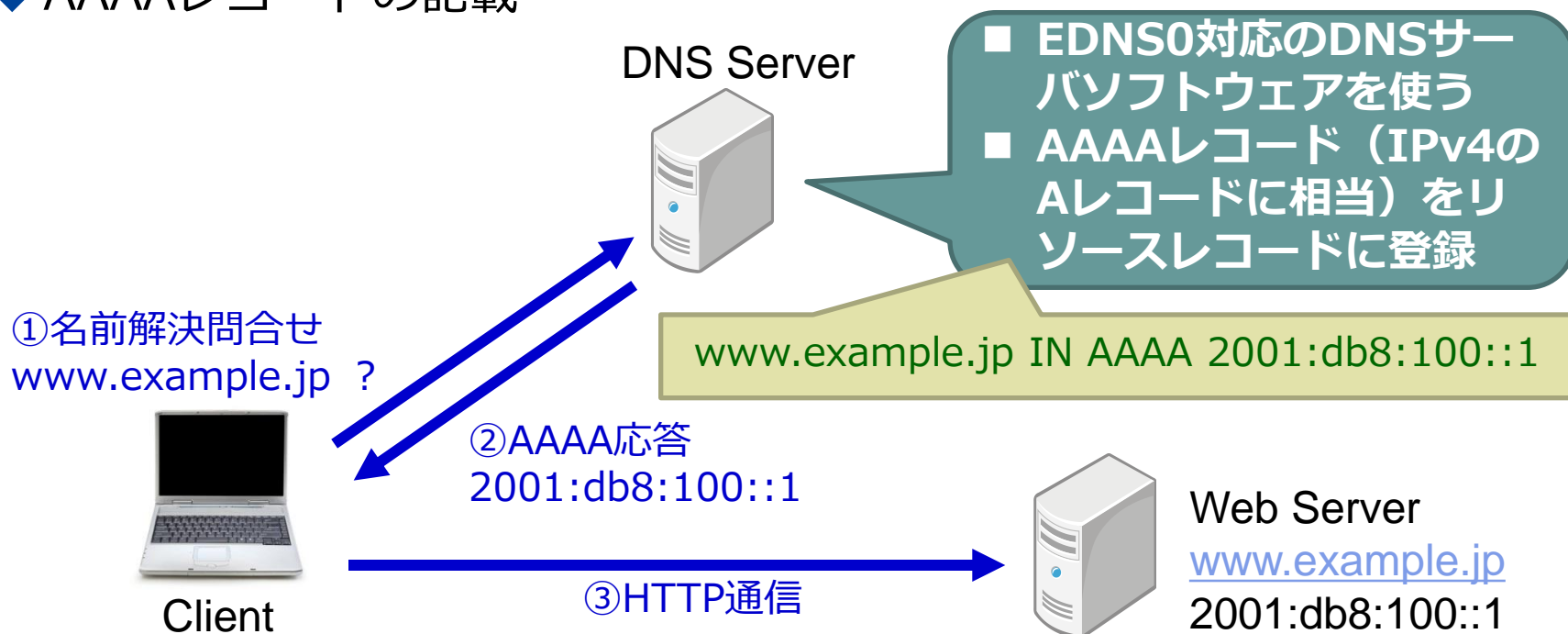
1. 名前解決でIPv4/IPv6両方のアドレスが扱える
2. IPv4/IPv6両方で接続できる



## 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





## 2-1. プログラミング言語と実行環境

# 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

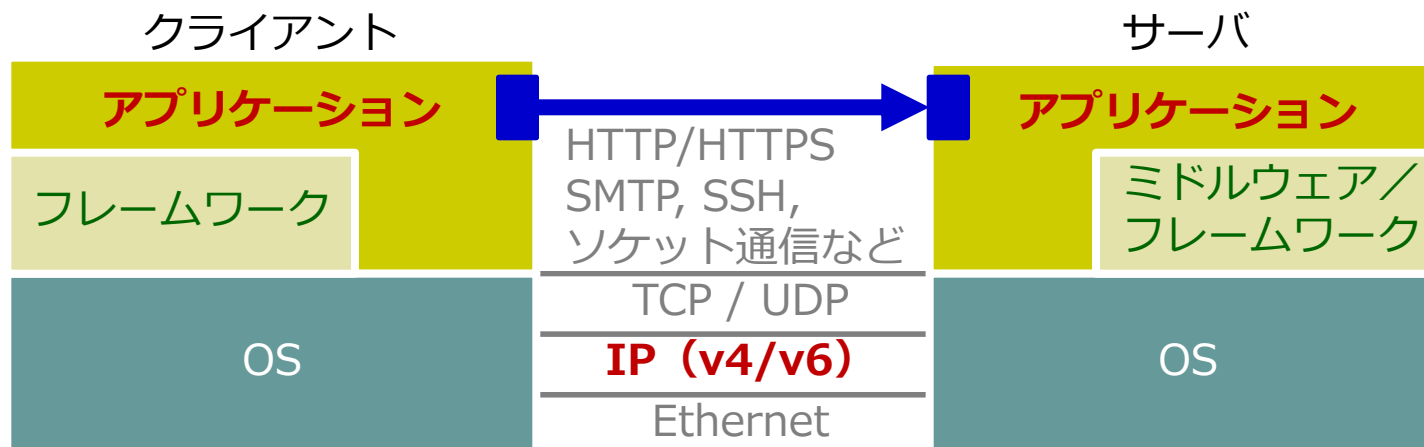
## 2-2. 通信処理の対応

### 原則5

### 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**

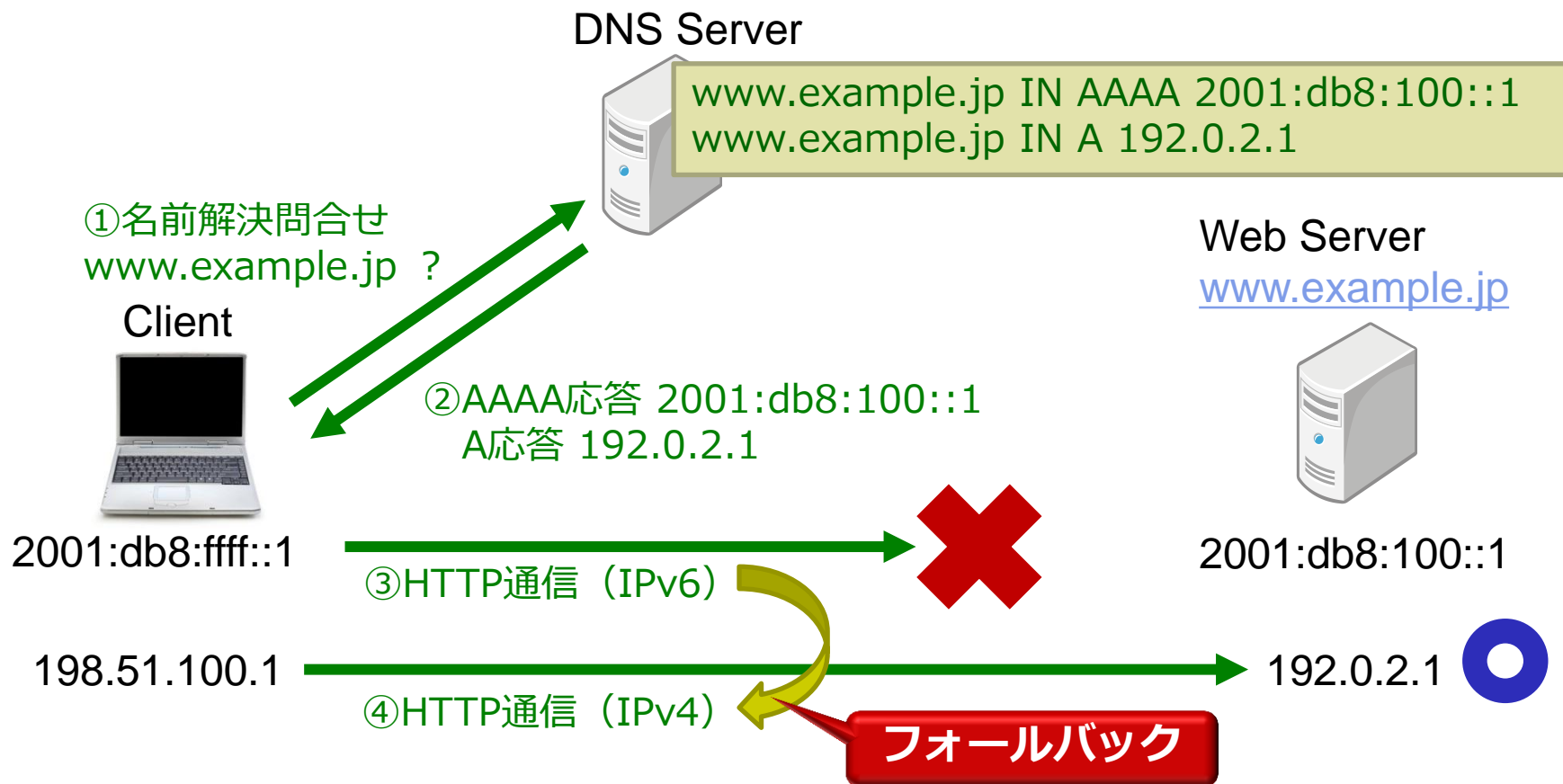
**いずれかが接続できない状況も想定に入れ、接続失敗時に接続先を迅速に切替えて接続する作りにする**

## 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

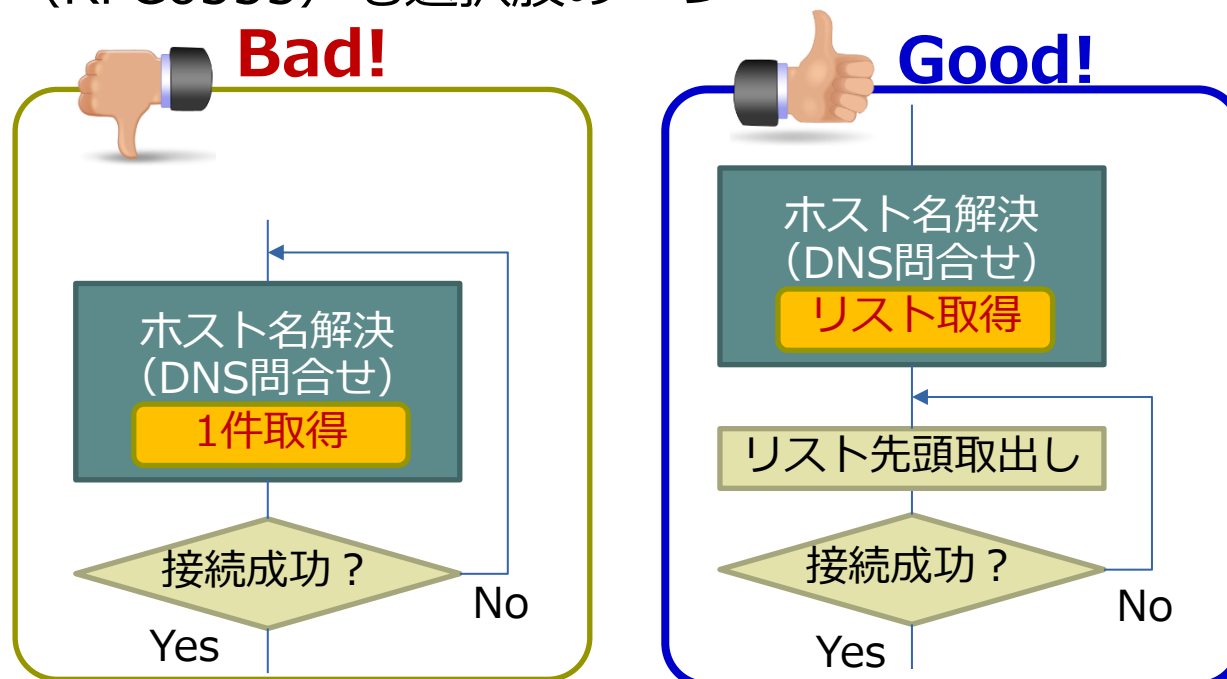
- ◆ 設定の不備を修正する
  - サービスを提供していないIPアドレスをDNSに登録しない
  - サービスを適切に提供する
- ◆ ネットワークの接続性を健全に保つ
- ◆ ポリシーテーブルをカスタマイズする
  - ポリシーテーブル：アドレス選択の優先順位を制御するテーブル

## 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

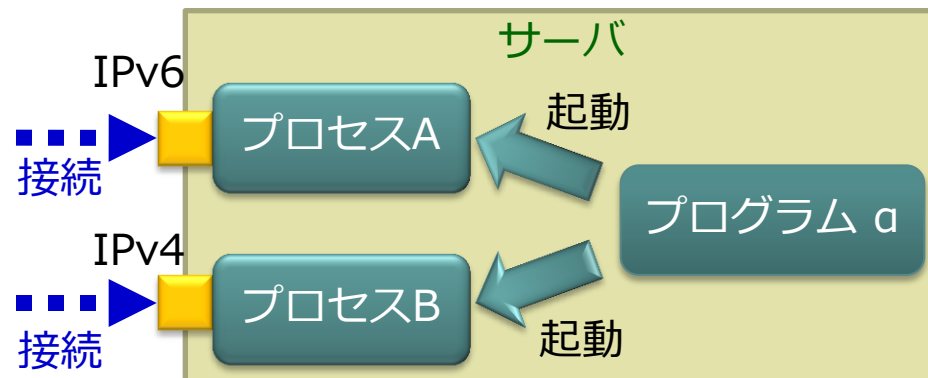
# サーバプログラム

## ■ IPv4/IPv6両方の接続を同時に受付ける2種類の手法

■ : ソケット

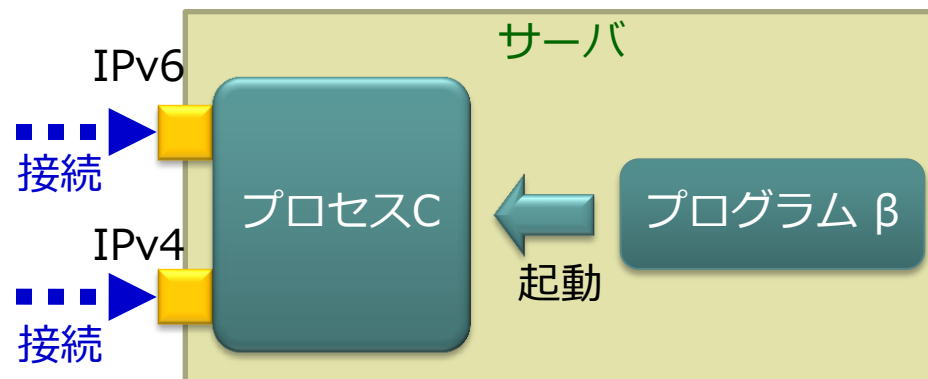
### 1. 2プロセス型

- 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-ftp



### 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



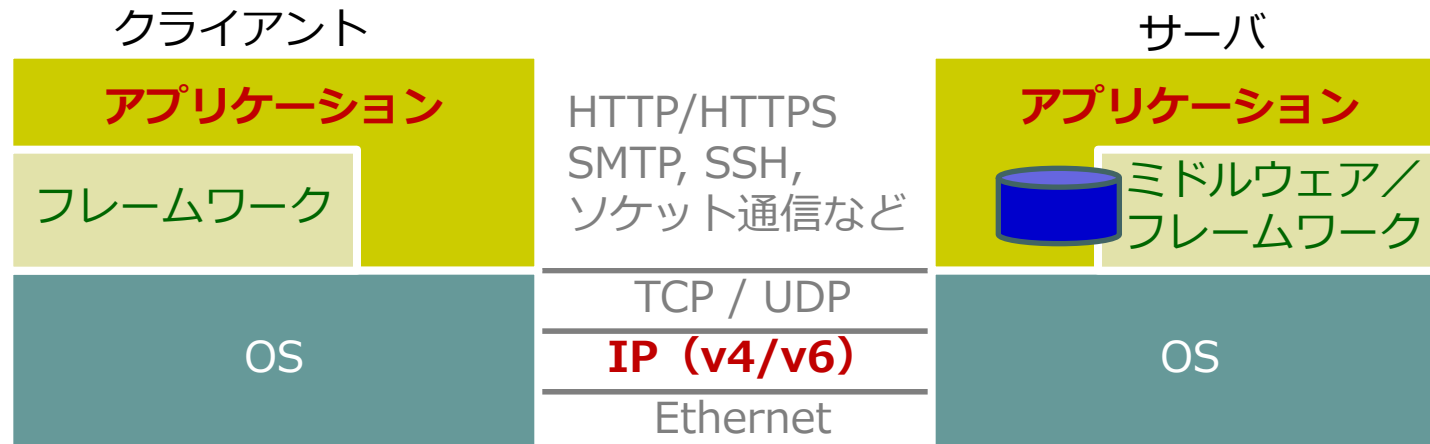
## 2-3. データとして扱う箇所の対応

### 原則6

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	<b>128bit</b>
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、 <b>16進数</b> で表記
	区切り文字	. (ドット)	: (コロン)
	文字列長	15文字以内	<b>39文字以内</b>

## ■ 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

IPアドレス

**Bad!**

IPv4/IPv6両対応

39文字以内の文字列 [VARCHAR(39)]

**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

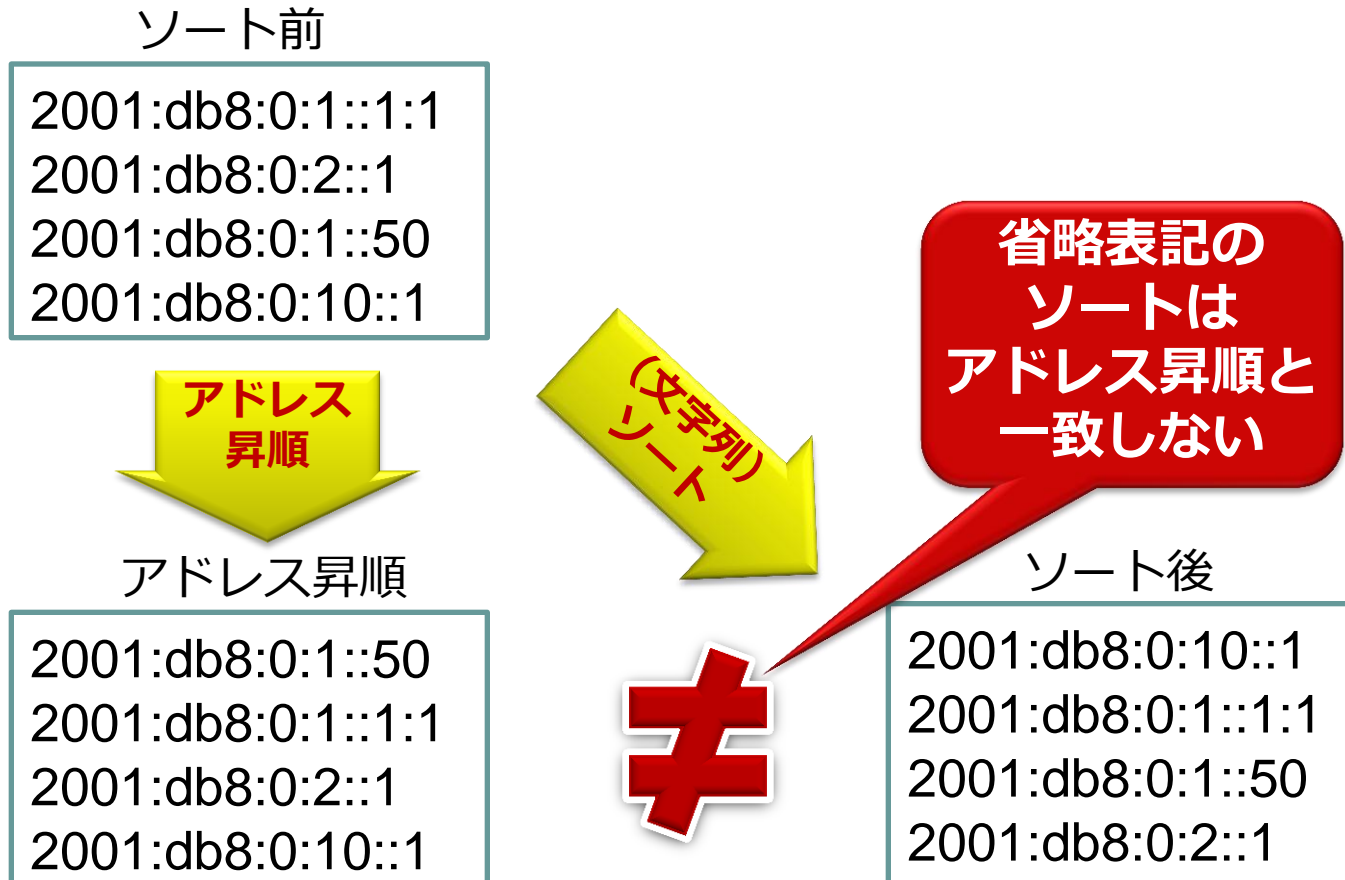
## ■ 検索・パターンマッチング

- ◆ 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

ソート前

```
2001:db8:0:1::1:1
2001:db8:0:2::1
2001:db8:0:1::50
2001:db8:0:10::1
```

完全表記

```
2001:0db8:0000:0001:0000:0000:0001:0001
2001:0db8:0000:0002:0000:0000:0000:0001
2001:0db8:0000:0001:0000:0000:0000:0050
2001:0db8:0000:0010:0000:0000:0000:0001
```

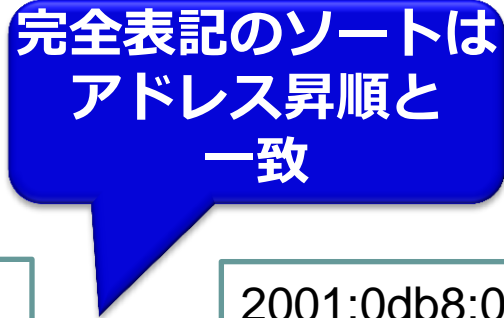
アドレス  
昇順



アドレス昇順

```
2001:db8:0:1::50
2001:db8:0:1::1:1
2001:db8:0:2::1
2001:db8:0:10::1
```

完全表記のソートは  
アドレス昇順と  
一致



(文字列)  
ソート



ソート後

```
2001:0db8:0000:0001:0000:0000:0000:0050
2001:0db8:0000:0001:0000:0000:0001:0001
2001:0db8:0000:0002:0000:0000:0000:0001
2001:0db8:0000:0010:0000:0000:0000:0001
```



---

# まとめ



# まとめ

- **基本方針とアプリケーションIPv6対応 6原則 に沿って対応を！**
  
- **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6原則**
  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
  4. IPv4/IPv6両対応のプログラミング言語と実行環境を使う
  5. 通信処理をIPv4/IPv6の両方に対応させる
  6. 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

## 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>
  
- 技術評論社 Software Design 2013年8月号 p.162-167  
（「IPv6化の道も一歩から」連載第8回）