

コンピュータシステムA - ハードウェアを中心に -

#8 bit, Byte, データ表現、互換性

bit : データの最小単位

- 1bit = 最小状態の単位 = 二進一桁
- コンピュータ内部は電気配線

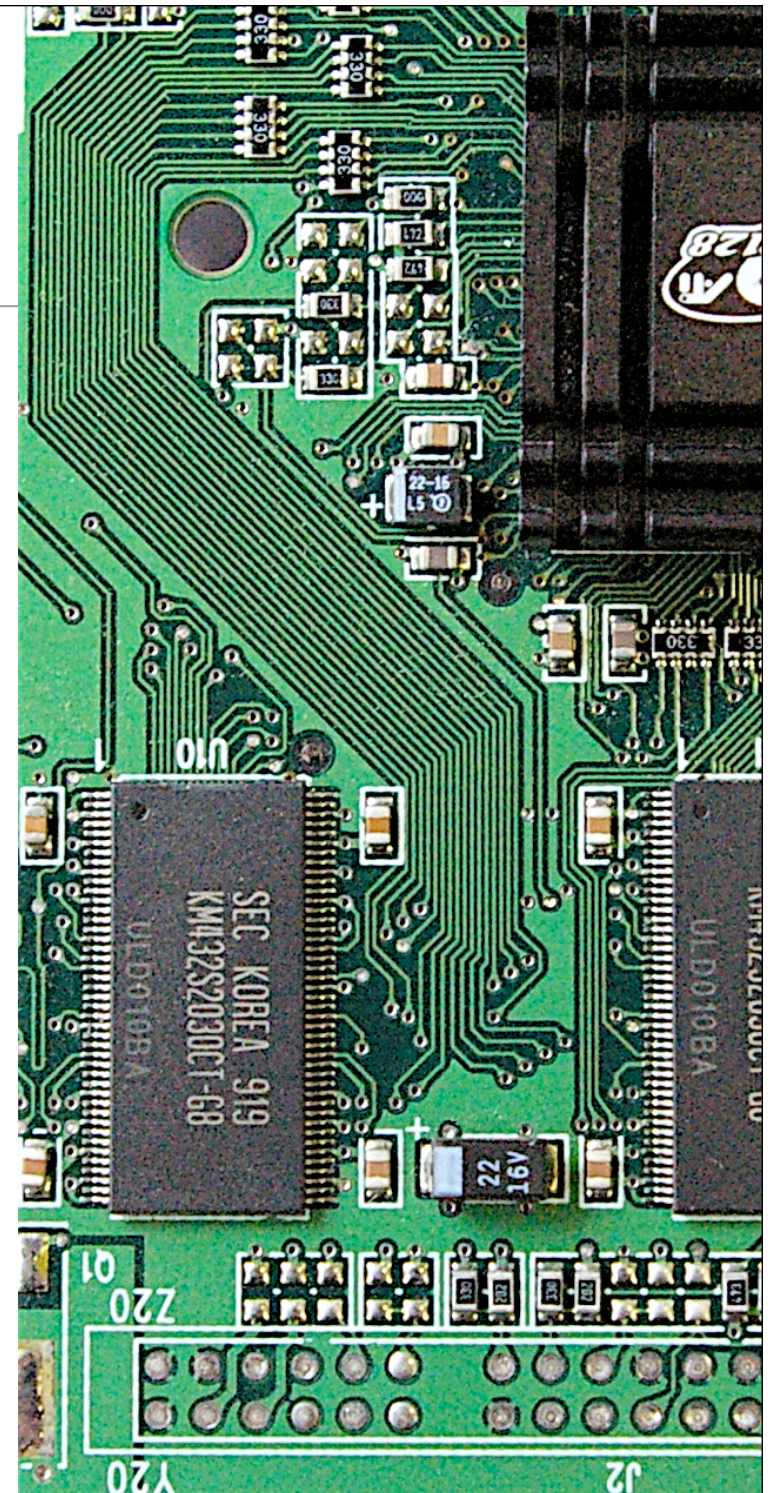
配線に電気が通っている、いない、だけで処理（状態は2種）

- 二値（二進）動作にうまく対応

二進一桁を配線一本で実現

「0と1（二進数）で動作」の実体

- 1bit = 二進一桁 = 配線一本



Byte : データの標準枠

- Byte (バイト)

コンピュータが扱うデータの基本単位 (歴史的経緯)

bit を 8 つまとめて 1 Byte とする

0-255までの256種類の値が入る

255を超える値は二桁 (2Bytes) 使う

- アルファベットは 1 バイトでおさまる
- 漢字は (普通は) 2 バイトを要する

「フロッピー1枚は新聞何枚に相当し、、」

確認

- デジタルは 0, 1 である（二値である）は間違い
- デジタル：数値で処理するところが重要
- コンピュータが二値である理由

組み合わせ数が少ないので回路が単純になる

中間的な電圧を利用するなど技術的に複雑になる
（結果的に速度を上げられない）

- 3値や多値ができて不思議はないが、今は無い
- フラッシュメモリなど部分的実用例はある

フォーマット（書式）

- データの解釈には解釈（復号）ルールが必要
- つまりデータにはフォーマット（書式）がある

フォーマットを間違えて解釈すると間違った結果が導き出される

異なるアプリケーションでデータが扱えない理由

（データにおける）「互換性」という概念の実体

いわゆる文字化けの原因

文字のデータ表現

- メモリ

数値を格納

- 文字に番号を振る

ASCII (右表)

文字

a
61

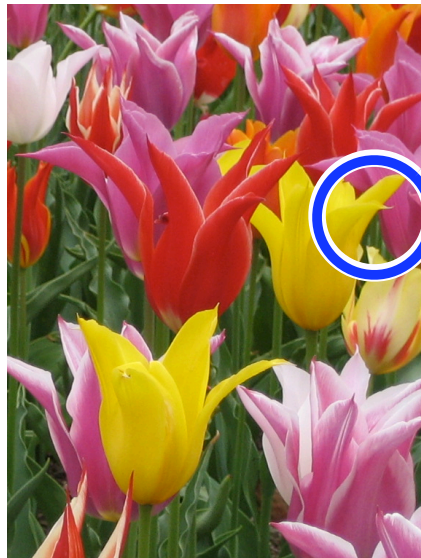
コード
(数値)

つまりメモリに格納されるのは文字ではなく文字に相当する値である

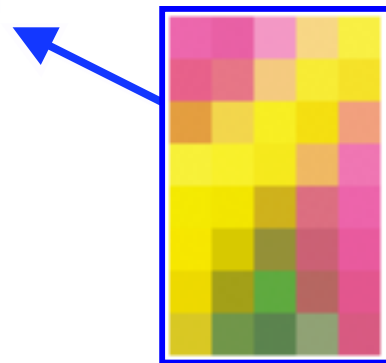
ASCII文字は 1 バイトで表現されているが、その実体は数値である。
つまり 'a' は番号 61 の文字。
61 は 16 進数表記なので、10 進数で表記すると 97 番文字となる。

\0		sp	0	@	P	`	p
00	10	20	30	40	50	60	70
		!	1	A	Q	a	q
01	11	21	31	41	51	61	71
		"	2	B	R	b	r
02	12	22	32	42	52	62	72
		#	3	C	S	c	s
03	13	23	33	43	53	63	73
		\$	4	D	T	d	t
04	14	24	34	44	54	64	74
		%	5	E	U	e	u
05	15	25	35	45	55	65	75
		&	6	F	V	f	v
06	16	26	36	46	56	66	76
		'	7	G	W	g	w
07	17	27	37	47	57	67	77
		(	8	H	X	h	x
08	18	28	38	48	58	68	78
\t		)	9	I	Y	i	y
09	19	29	39	49	59	69	79
\n		*	:	J	Z	j	z
0a	1a	2a	3a	4a	5a	6a	7a
		+	;	K	[	k	{
0b	1b	2b	3b	4b	5b	6b	7b
		,	<	L	\	l	
0c	1c	2c	3c	4c	5c	6c	7c
		-	=	M	]	m	}
0d	1d	2d	3d	4d	5d	6d	7d
		.	>	N	^	n	~
0e	1e	2e	3e	4e	5e	6e	7e
		/	?	O	_	o	
0f	1f	2f	3f	4f	5f	6f	7f

画像のデータ表現



絵は画素(Pixel : Picture Element)ごとに分解



一画素ごとに赤・緑・青 (RGB) に
色分解して各色256段階で記録
最大 16,777,216 色

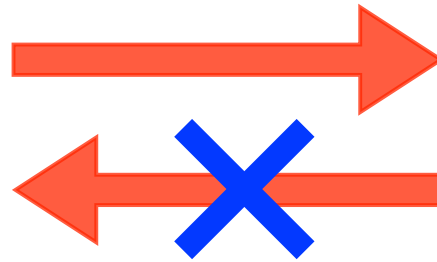
	赤	緑	青
	→ 229	83	158
	→ 242	231	0
	→ 80	155	46

動画も簡単にデータ化できますね？

可逆圧縮と非可逆（不可逆）圧縮



無圧縮：約 400KB



非可逆圧縮



JPEG : 40.9KB

JPEG
VS
GIF



無圧縮：約 34KB



可逆圧縮



GIF : 1.8KB

音声のデジタル表現

- サンプリング

標本化と量子化

CDは44KHz, 16bit

- MP3

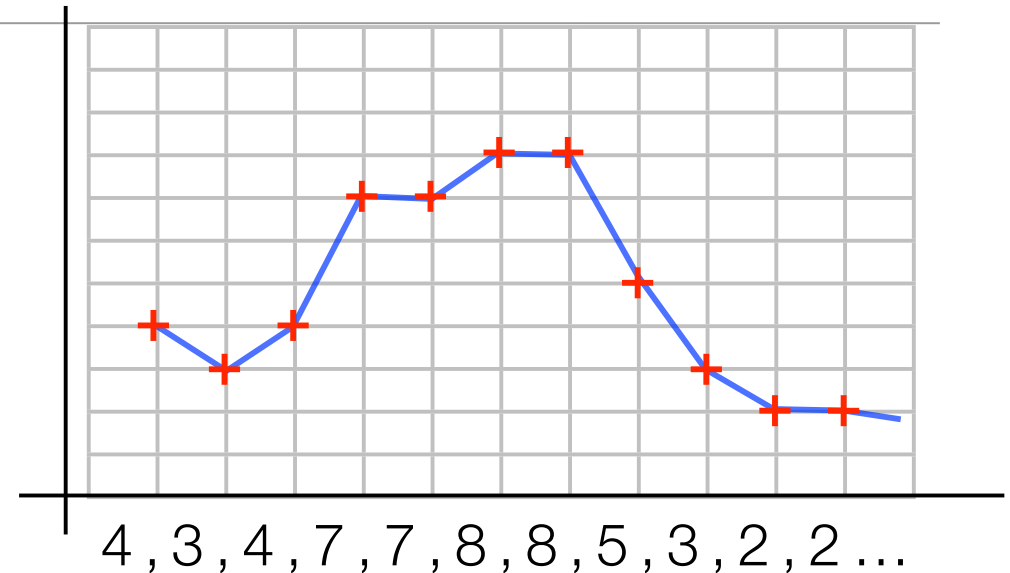
非可逆圧縮の一つ

人間が聞き取りにくい音の情報を削除する→音質劣化

CD音源を 1/8～1/15 程度に圧縮

- AAC, ATRAC, WMA などなど他多数

圧縮率と品質のよりよい両立を求めて



まとめ：デジタルデータとフォーマット

- その実体は数値（記号）の列

音声：111,121,122,89,80,82,75....

静止画：10,240,22,30,34,80...

音声付き動画：12,33,45,1123,488...

文字：33,38,42,60,32,39,55,80...

- これだけでは利用できない（意味が汲み取れない）

符号化ルールとデータは常に一体

- このルールがフォーマット（書式）を生む

補足：数値の表現

補足：数値の表現

- 数値の表現

整数・浮動小数

- ハードウェア（主としてCPU）が直接処理できるか否か
- あり得る全ての数値を処理できるわけではない

処理速度・メモリ消費量（トレードオフ）

そのように作られた処理系もある

整数

- 整数

二進表記

1Byte : 8bit, 0~255

8bit

4Byte : 32bit, 0~4G (9桁、約43億)

32bit データ

8Byte : 64bit, 0~?? (19桁)

64bit データ

10進表記	4bit 2進表記
15	1111
14	1110
13	1101
12	1100
11	1011
10	1010
9	1001
8	1000
7	0111
6	0110
5	0101
4	0100
3	0011
2	0010
1	0001
0	0000

整数

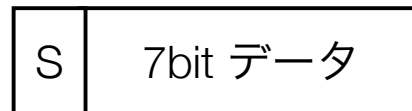
- 符号付き整数（負数の表現）

負数は 2 の補数
(1引いてビット反転)

符号なし 1Byte : 8bit, 0~255

符号あり 1Byte : 8bit, -128~127

Sは符号 (sign)



- なぜ？

負数を単純に加算することができる

桁あふれをどう処理するかはソフト
が判断すればよい

符号ビット

10進表記	4bit 2進表記	
7	0	111
6	0	110
5	0	101
4	0	100
3	0	011
2	0	010
1	0	001
0	0	000
-1	1	111
-2	1	110
-3	1	101
-4	1	100
-5	1	011
-6	1	010
-7	1	001
-8	1	000

負数

少数

- 浮動小数点表現

$$123.45 \rightarrow 1.2345 \times 10^2$$

指数部(^2)と仮数部(12345)に分けて表現

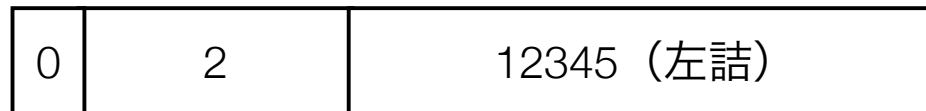
Sは仮数の符号 (sign)

単精度
(32bit)



$$10^{-38} \sim 10^{38}$$

10進表記
での例



ただし実際にはこれを 2 進で行う (後述)

倍精度
(64bit)



$$10^{-308} \sim 10^{308}$$

実際の浮動小数点表現（参考）

- 123.45 の浮動小数点表記は？

符号	指数部	仮数部
0	10000101	11101101110011001100110

$10000101 = 128 + 5$

1 111011 . 01110011001100110

略された 1 をつけて、6桁めに小数点

バイアスの127を

1111011 = 123（整数部）

引いた 6 桁めに少数点

続く 011100... が小数部

（つまり $0.25 + 0.125 + 0.0625...$ ）

表現できた数値は **123.4499969...**（有効桁数は24bitつまり7桁少し）

他の数値の例

$0.1 = 0, 01111011, 10011001100110011001101$ （誤差がある）

$0.5 = 0, 01111110, 000000000000000000000000$ （割り切れた）

誤差・有効桁数

- 有効桁数に注意

Microsoft Excel 2004 for Macintosh での数値の扱い

乗数 (Nとする)	2のN乗	2のN乗に +1 した結果
8	256	257
16	65,536	65,537
32	4,294,967,296	4,294,967,297
64	18,446,744,073,709,600,000	18,446,744,073,709,600,000

どこでおかしくなったのだろうか？

48	281,474,976,710,656	281,474,976,710,657
49	562,949,953,421,312	562,949,953,421,313
50	1,125,899,906,842,620	1,125,899,906,842,620
51	2,251,799,813,685,250	2,251,799,813,685,250

末尾はそもそも 4 であるはずだが、15桁位置で切り捨てられ、+1 も反映されない

デジタル化による利益

まとめ：デジタルデータとフォーマット（再掲）

- その実体は数値（記号）の列

音声：111,121,122,89,80,82,75....

静止画：10,240,22,30,34,80...

音声付き動画：12,33,45,1123,488...

文字：33,38,42,60,32,39,55,80...

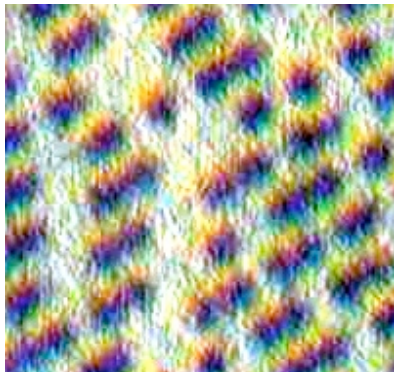
- これだけでは利用できない（意味が汲み取れない）

符号化ルールとデータは常に一体

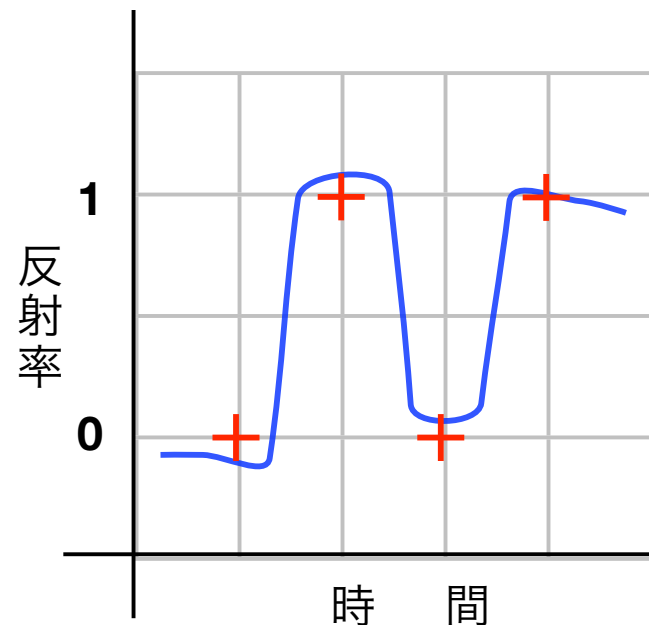
- このルールが**フォーマット（書式）**を生む

ノイズへの抵抗・完全な複製（復習）

- 1/2量子化単位以下の狂いであれば正しい値が得られる
二値化されている場合は 0/1 を間違えなければ良い
- 再複製の際に狂いが継承（蓄積）されない



CDのピット長は9種類

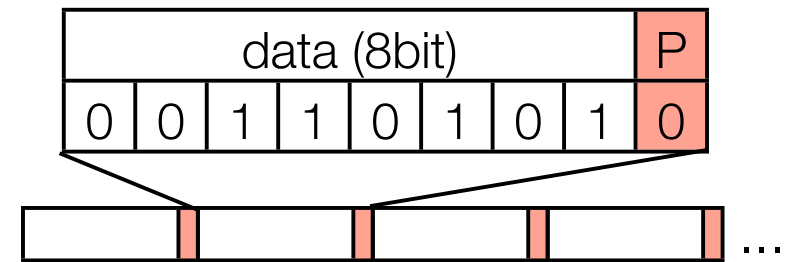


0 or 1 を間違えない程度に反射率の違いを検出できれば良い

最大ピット長の1/9 以下程度の誤差で長さ検出できれば良い

誤り検出・訂正

- 違った値が得られた場合の検出・修正が可能
 - 修正のための冗長な情報を付加
 - 誤り検出の例：
 - パリティ（奇偶性） - 1 bit 付加
-
- The diagram illustrates the addition of a parity bit. At the top, a box labeled 'da' contains three smaller boxes with the values 0, 0, and 1. A line connects this box to a larger box below, which represents a byte. This box is divided into two sections: the left section contains the original data (0, 0, 1) and the right section contains a single red box representing the parity bit, which has the value 1.



1 bit の誤りを検出可能 (2 bit の同時誤りは駄目)

- ## ● チェックサム

学生番号の合計は常に最下桁がゼロ（試してみよ）

- CRC (Cyclic Redundancy Check)

誤り検出・訂正

- 誤りを正せるような情報を加える
- 誤り訂正の例：
- 縦横チェックサム
- メモリにおける ECC (Error Correcting Code) (*)

64bit のデータに 8bit のECC情報を付加
1bit の誤りを検出・修正
2bit の誤りは検出のみ（修正不可能）

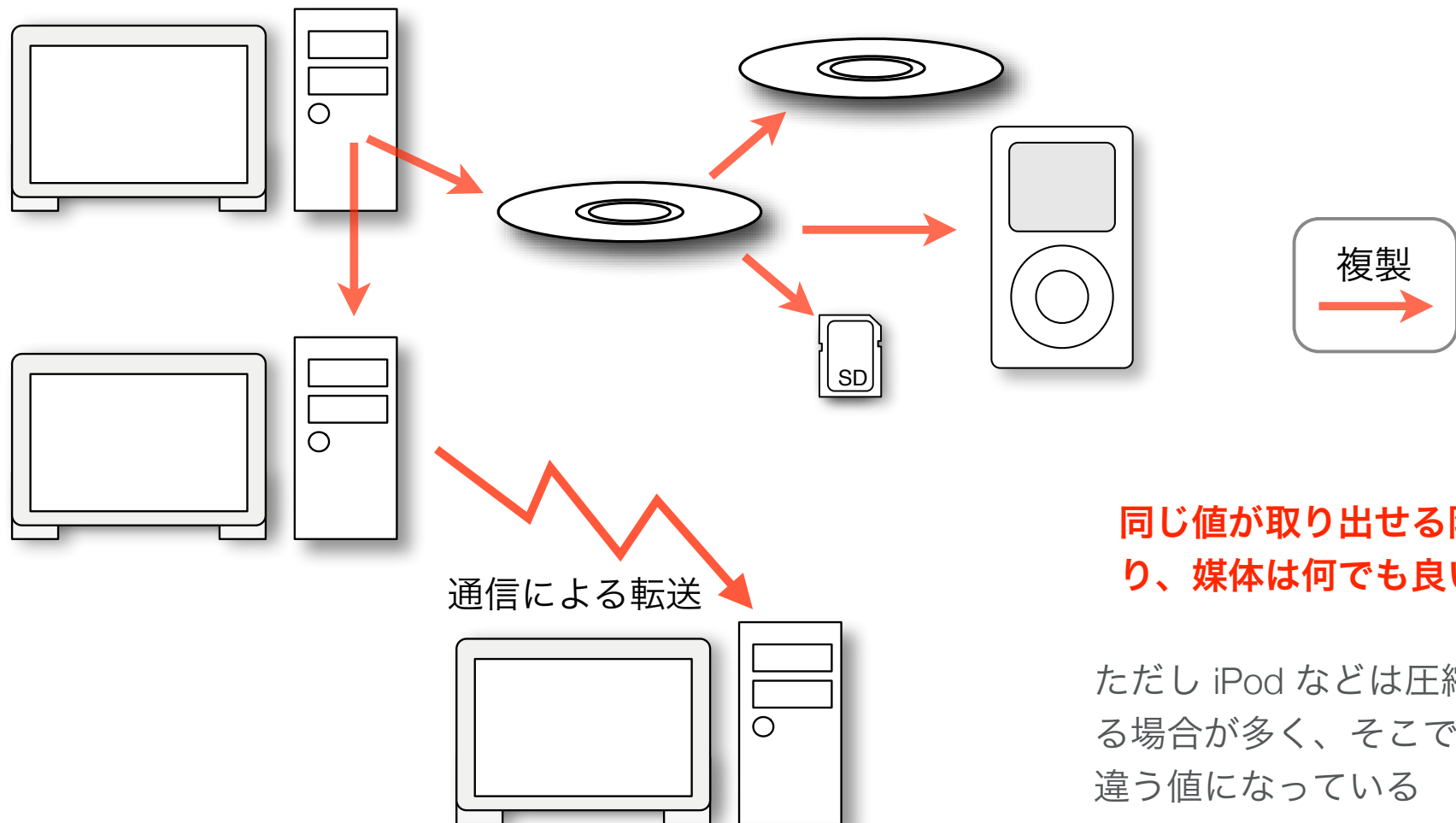
- CIRC : CD
- より多くの付加情報によってより広範囲な修正に対応

*ECC メモリは Error check and correct memory の略？

メディアの非依存性（復習）

CD/CD-R

パソコンを使って CD / CD-R / iPod / メモリカード 間の複製をとる



まとめ：デジタルデータの特徴

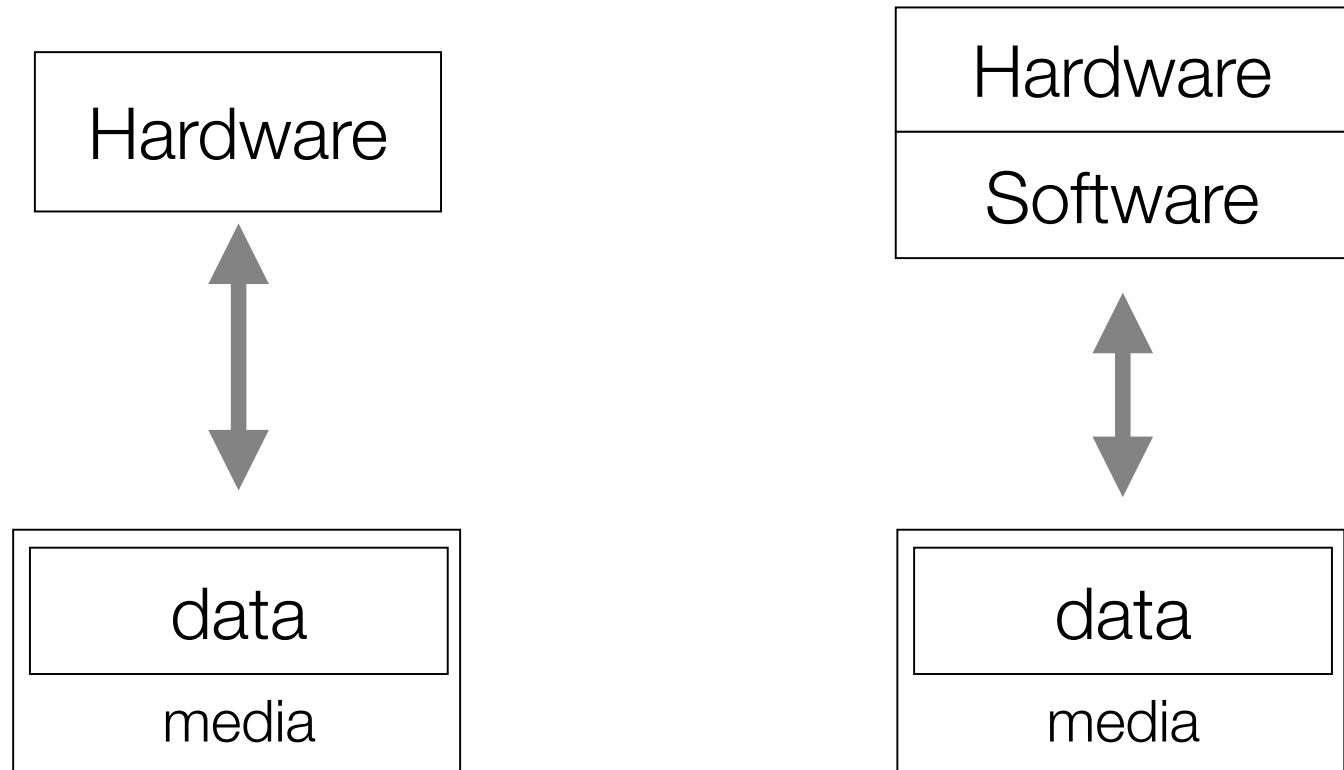
- 完全な複製
 - 複製・通信・保存に伴う劣化の回避
 - 完全さの検証も可能
- 不完全なデータ化
 - 初期ノイズの発生（近似でしかない）
- 考え方
 - 初めに精度を決めることでそれ以後の精度以内の変化をゼロにした
- 利益
 - 数学的なテクニックが多く適用可能に
 - コンピュータによる知的な自動処理が可能に

全体のまとめ

- デジタルデータのメリット
 - 完全な複製
 - デジタルコンピュータによる自動処理
- デジタルデータとフォーマットの関係
 - デジタル化で情報はメディア（物理的制約）からは自由になったがフォーマットが重要になった
 - 互換性という概念

デジタルシステムの柔軟性

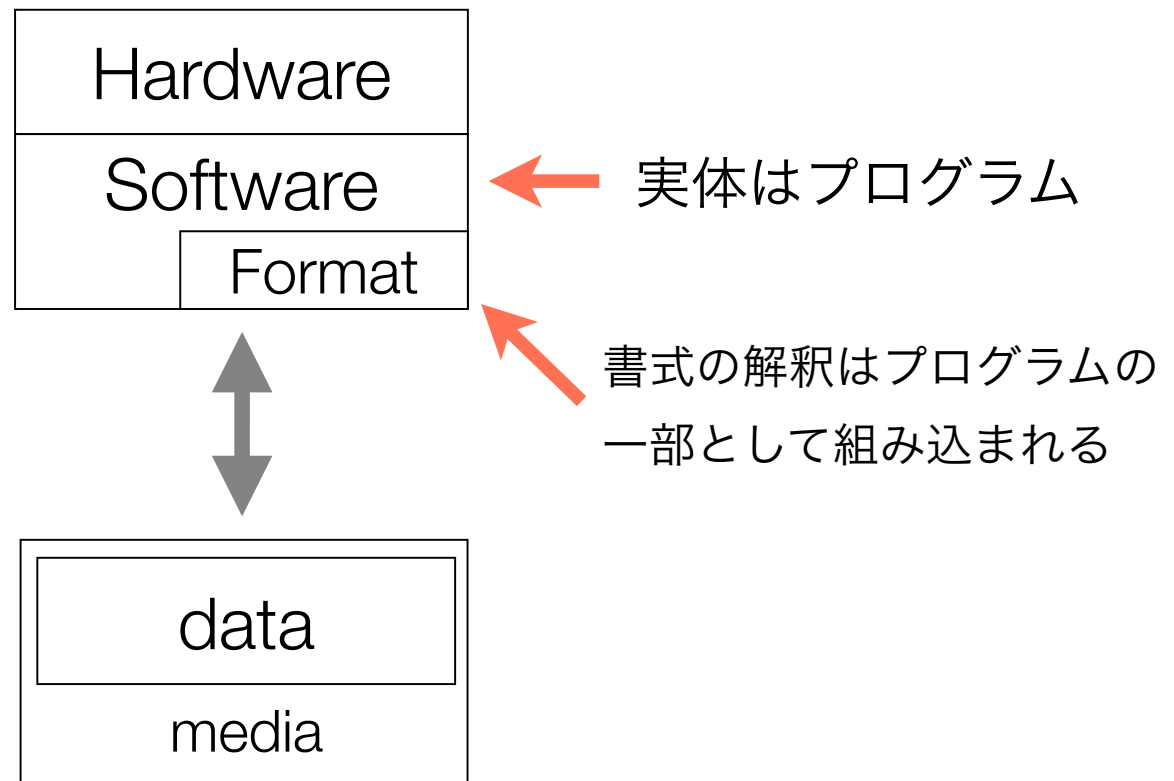
アナログシステムとデジタルシステム



典型的なアナログシステム
(レコードプレーヤーなど)

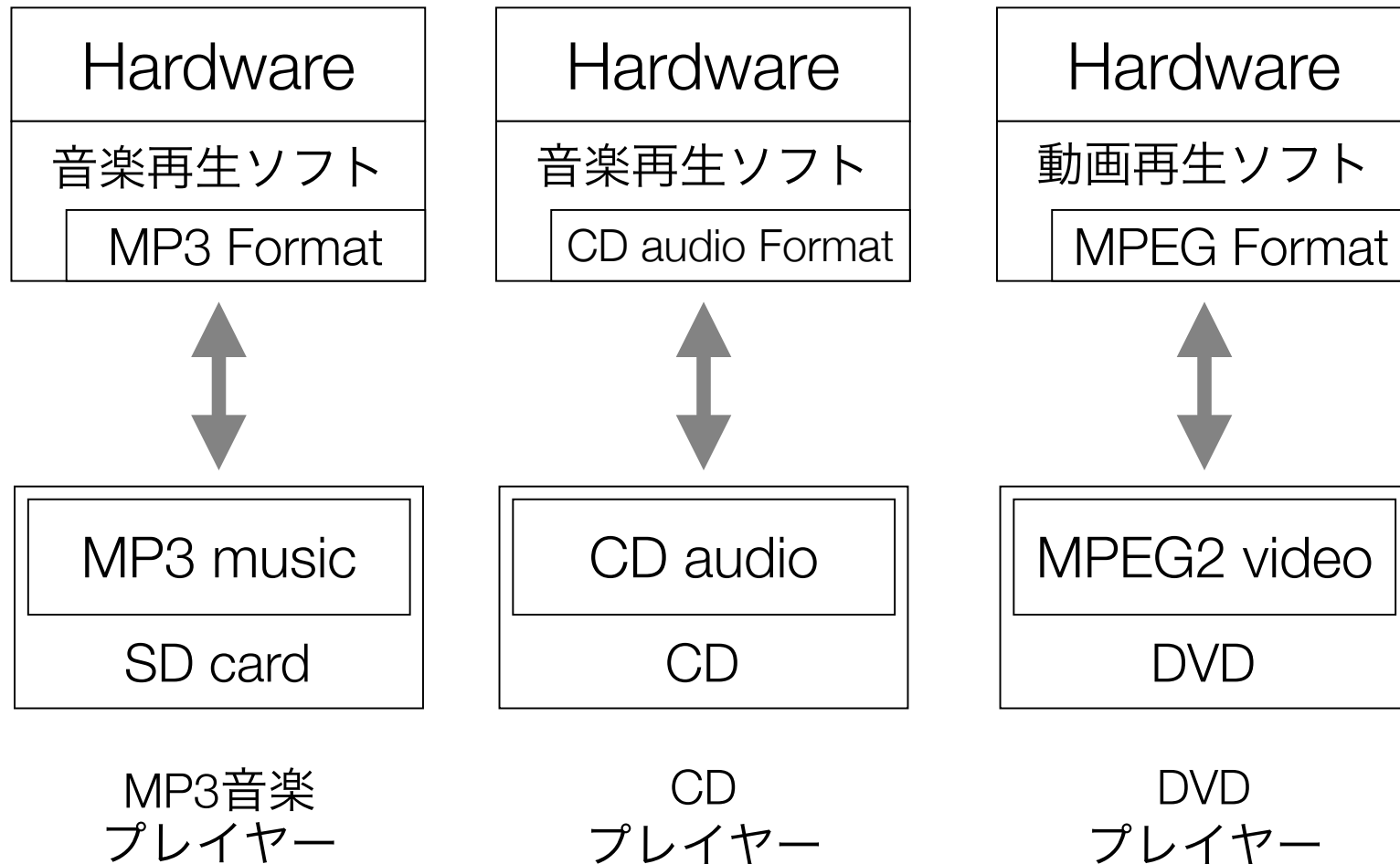
典型的なデジタルシステム
(コンピュータなど)

書式とデータの関係

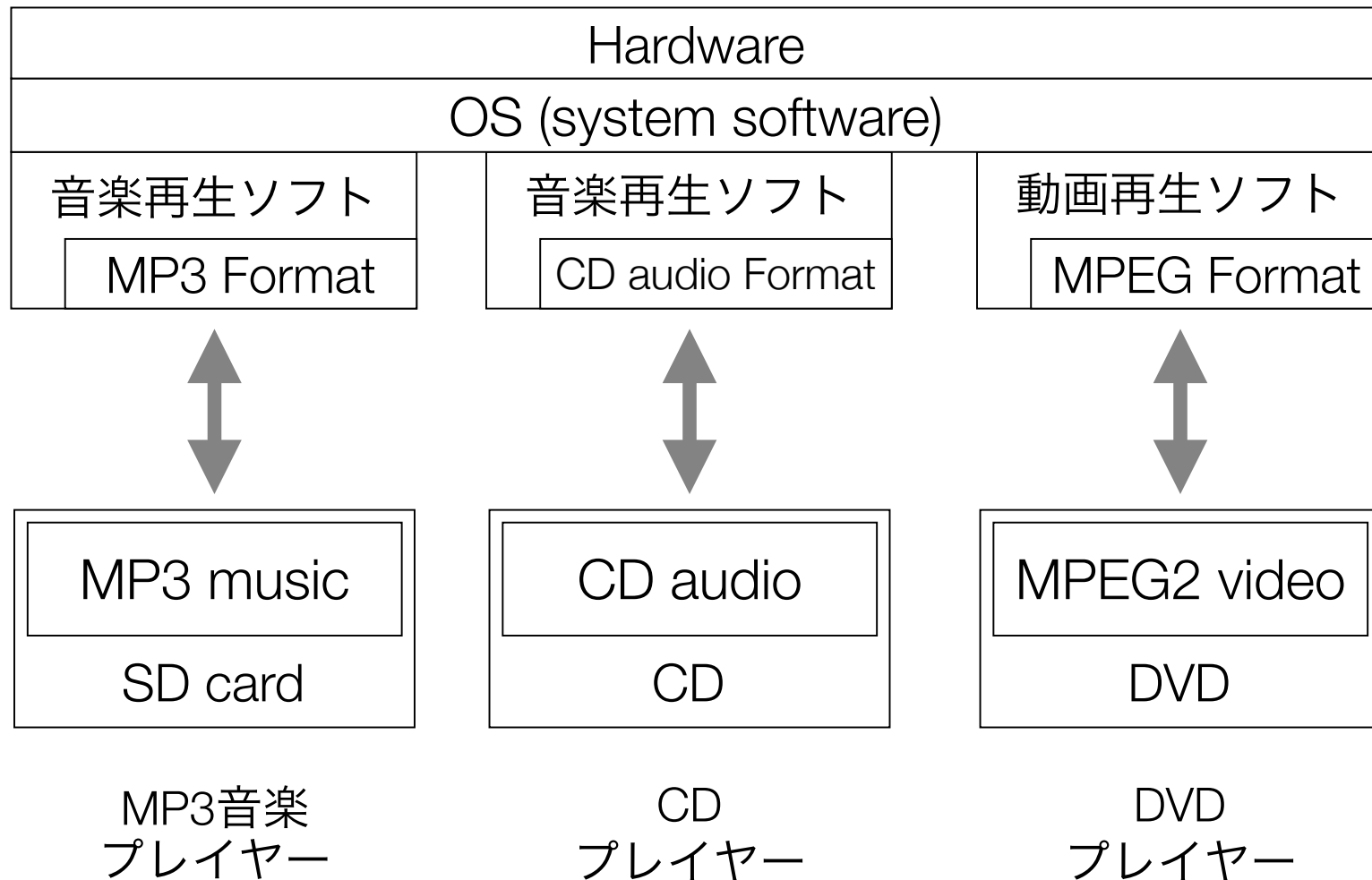


典型的なデジタルシステム
(コンピュータなど)

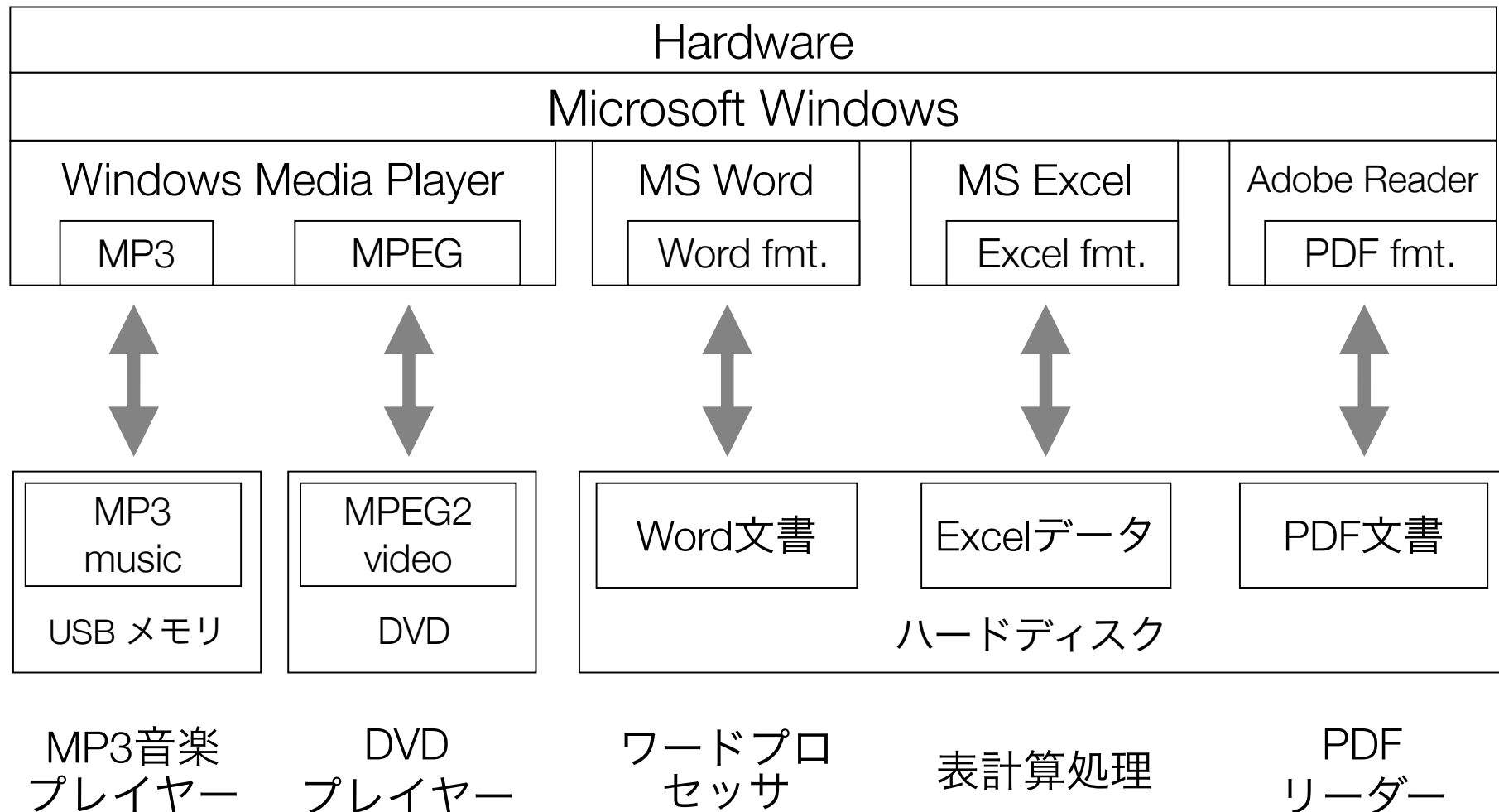
デジタルシステムの柔軟性



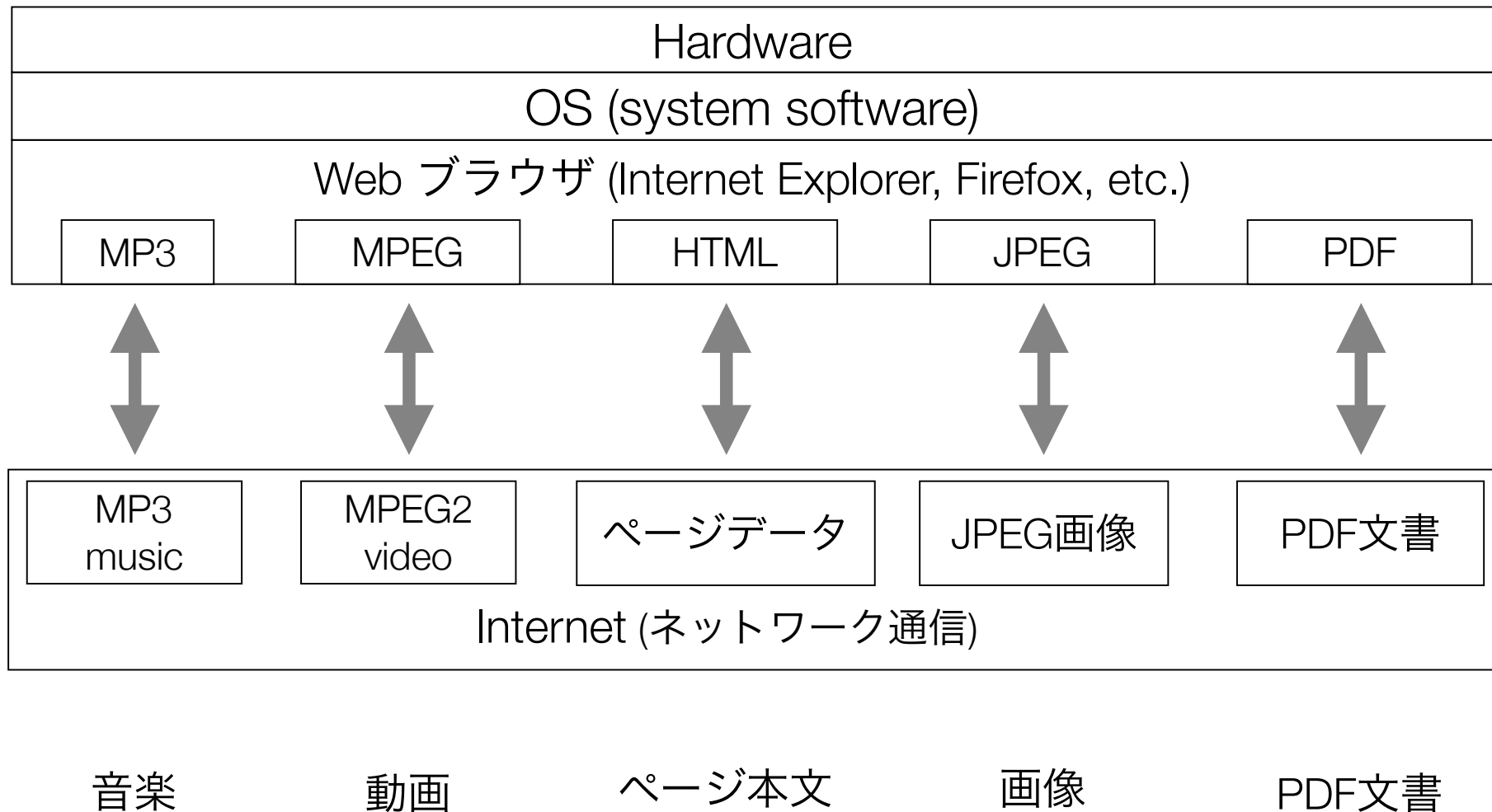
PC : 汎用デジタル処理システム



いつも使っている Windows パソコン



Web ページ閲覧におけるデータ処理



デジタル化のインパクト

- 汎用性
 - 情報はフォーマットと値で表現される
- 汎用(generic)のものに特定(specific)の機能を載せる
 - 汎用データ通信網に特定用途サービスを載せる
 - このサービスを汎用コンピュータに特定用途アプリケーション・ソフトウェアを載せて実現
 - ソフトウェアを入れ替えて新しい機能を実現可能
 - ソフトウェアで対応することの柔軟性

全体のまとめ

- デジタルデータのメリット
 - 完全な複製
 - デジタルコンピュータによる自動処理
- デジタルデータとフォーマットの関係
 - デジタル化で情報はメディア（物理的制約）からは自由になったがフォーマットが重要になった
 - 互換性という概念
- デジタル化のインパクト
 - ソフトウェアによる柔軟性

Case Study

- HTML5 の動画フォーマット
 - Google / WebM の VP8 （元 On2 の Ogg Theora）が意味するものは何か？
 - Apple は H.264 を推している
 - 何故か？



Google launches open WebM web video format based on VP8 (2010/5/19, Engadget)
<http://www.engadget.com/2010/05/19/google-launches-open-webm-web-video-format-based-on-vp8/>

Case Study

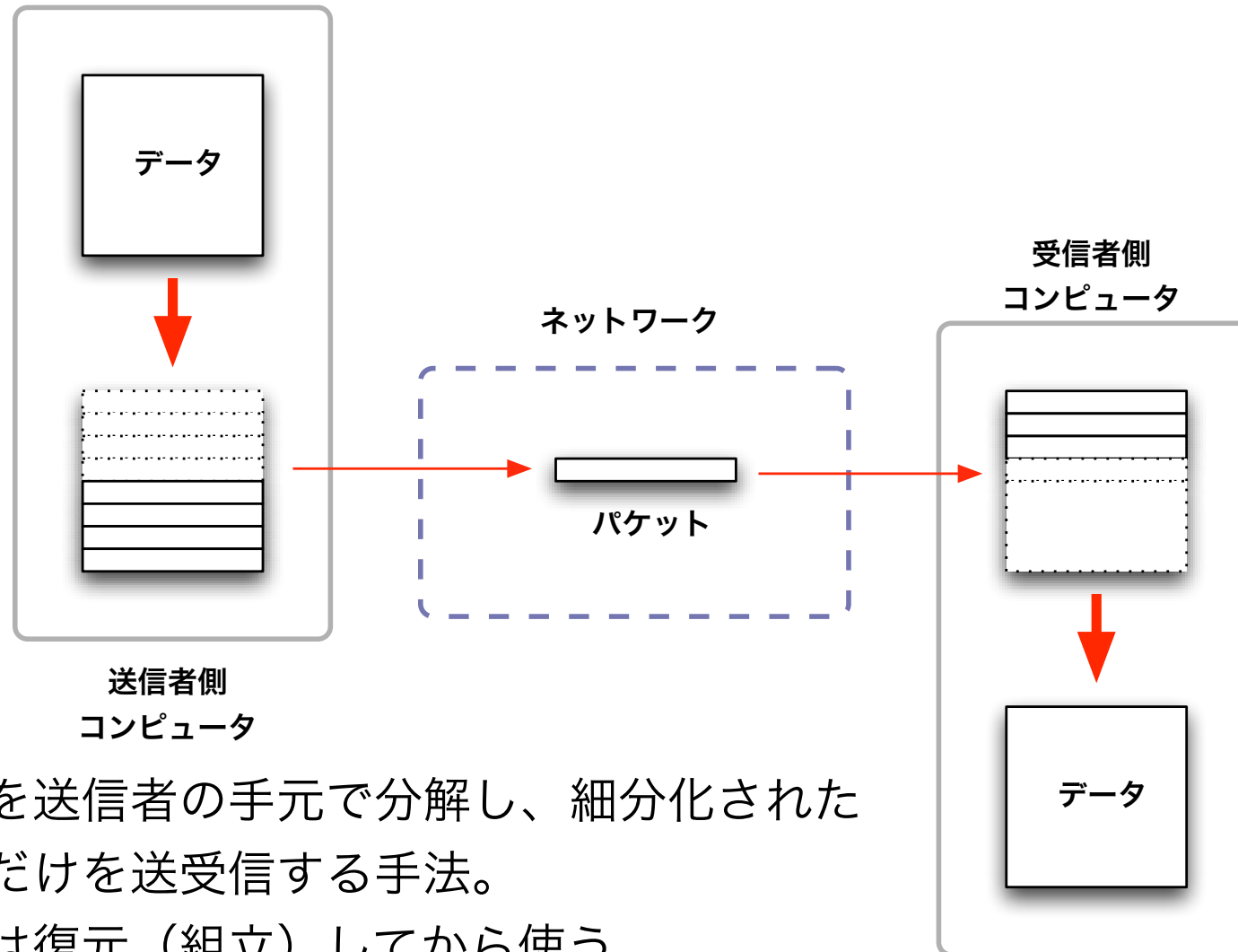
- HTML5 の動画フォーマット
 - Google / WebM の VP8 （元 On2 の Ogg Theora）が意味するものは何か？
 - Apple は H.264 を推している
 - 何故か？
- 互換性
 - 全ての環境で使えるフォーマットが我々には必要だ

コンピュータシステムA - ハードウェアを中心に -

#11 Internet の構造と歴史

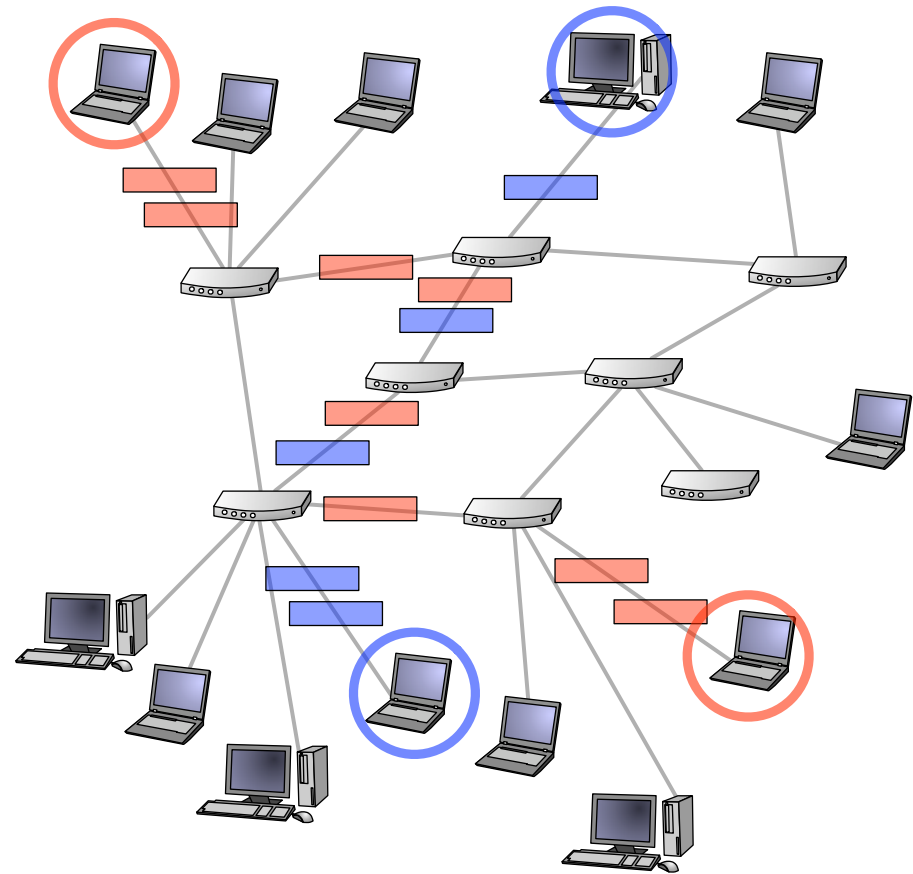
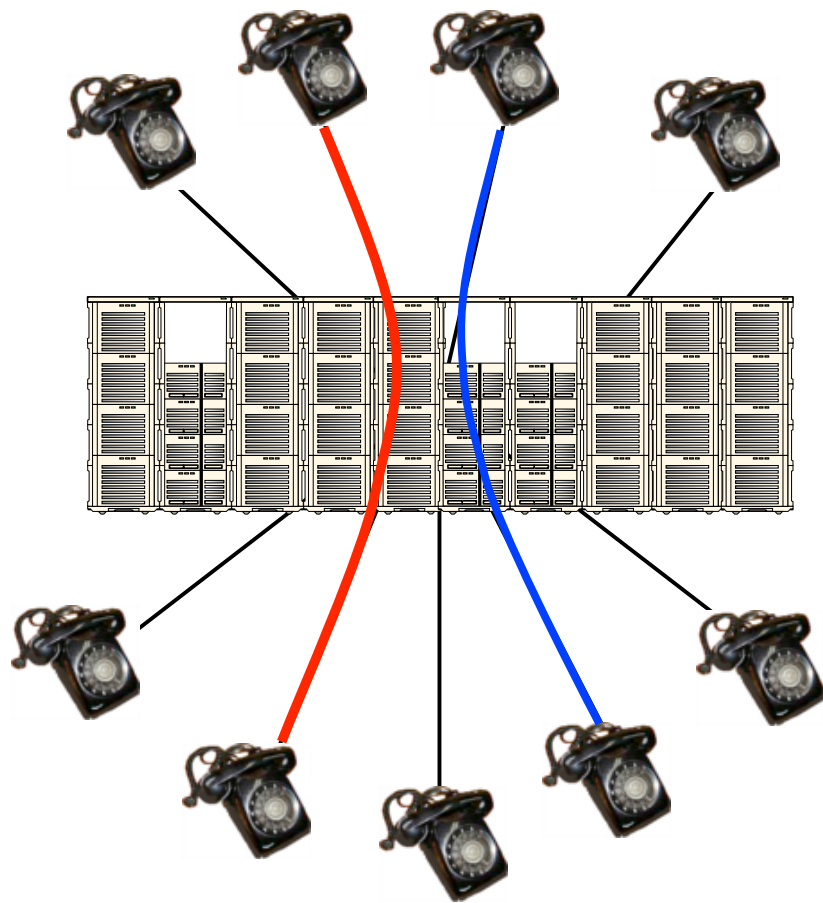
Yutaka Yasuda

パケット通信



データを送信者の手元で分解し、細分化されたデータだけを送受信する手法。
受信側は復元（組立）してから使う。

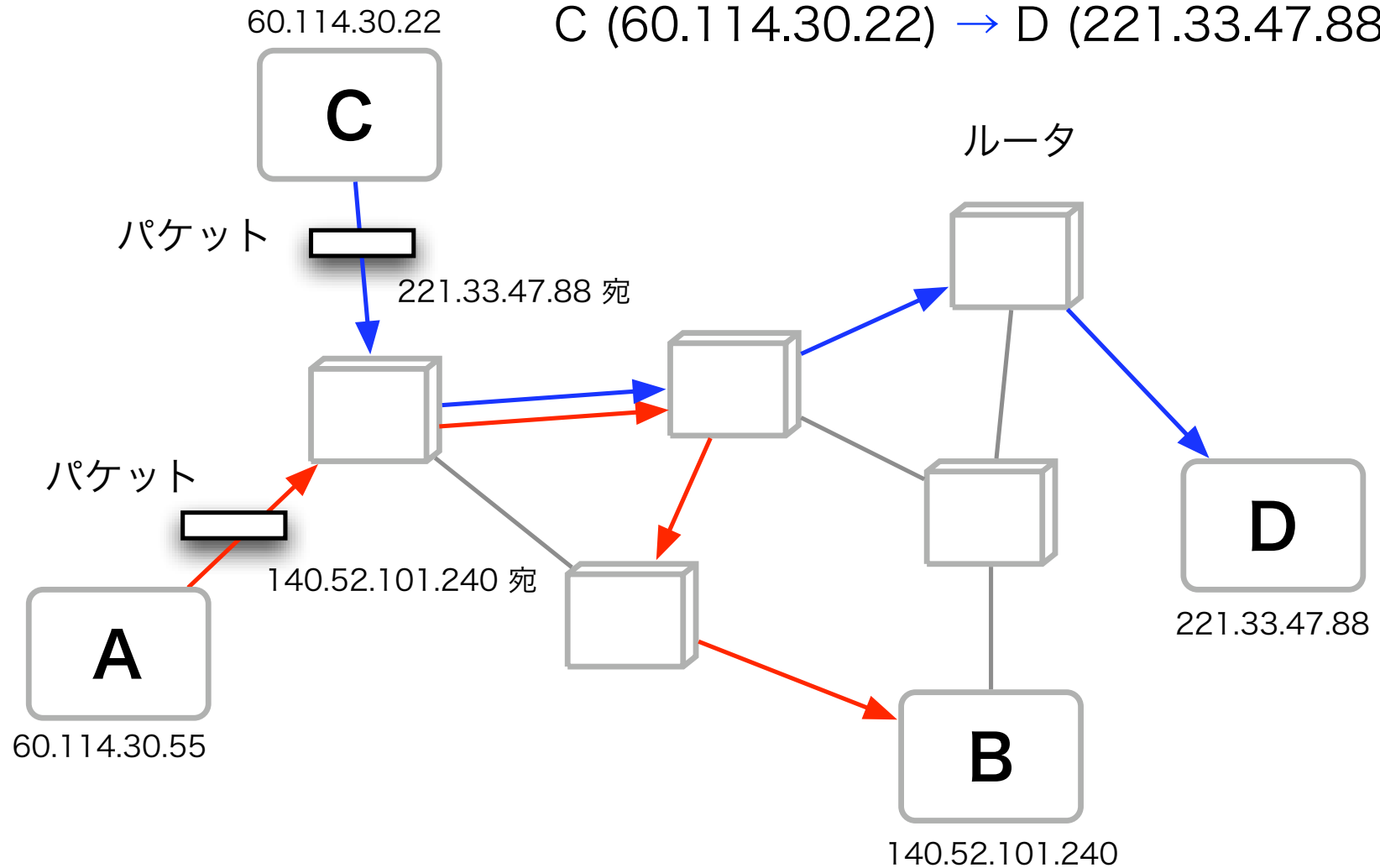
回線交換とパケット交換



ルーティング (再掲)

A (60.114.30.55) → B (140.52.101.240)

C (60.114.30.22) → D (221.33.47.88)



プロトコル

- 通信のための決められた一連の手続き

IP (Internet Protocol)

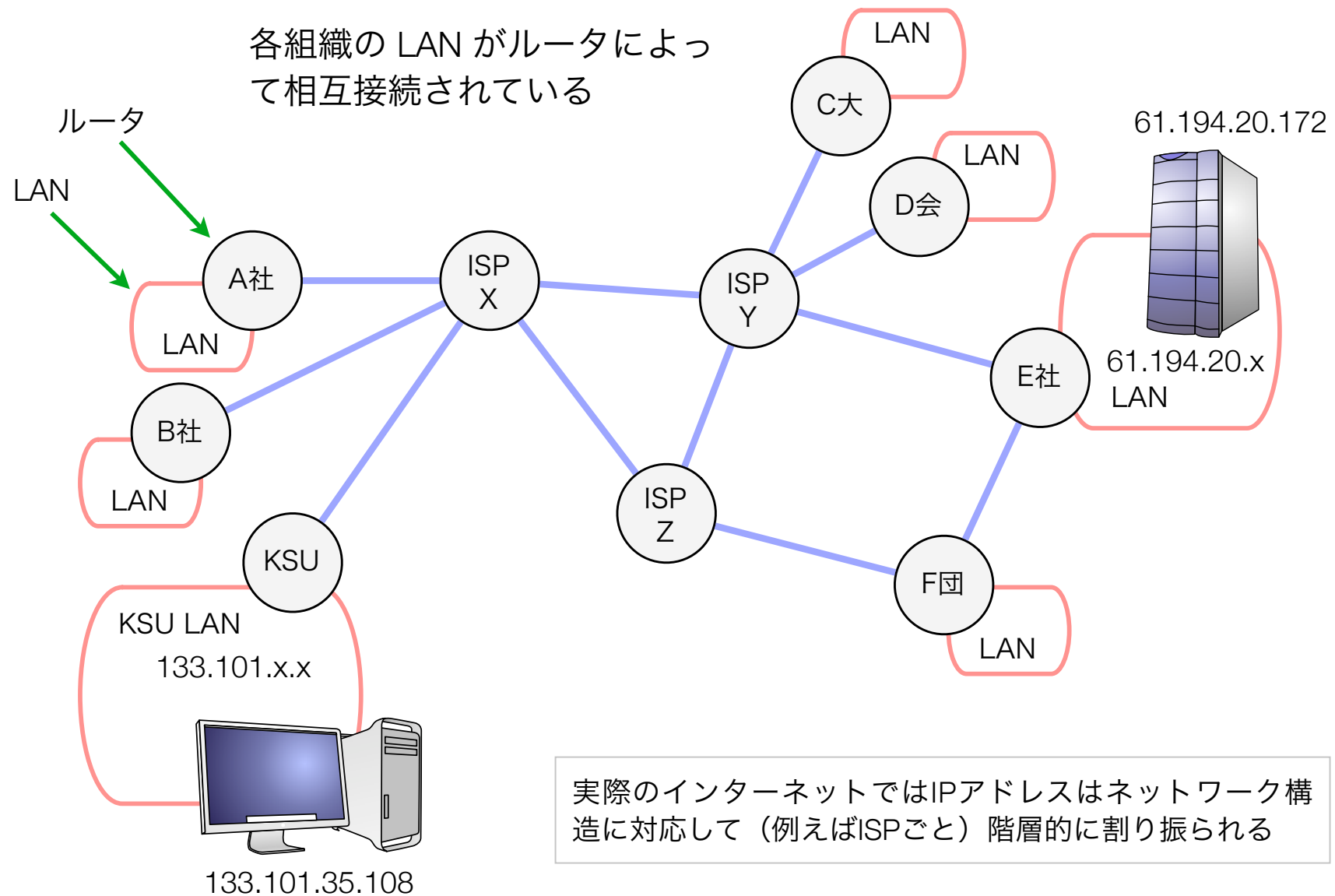
IPアドレスを用い、インターネットの中でデータを交換するための手順

- TCP

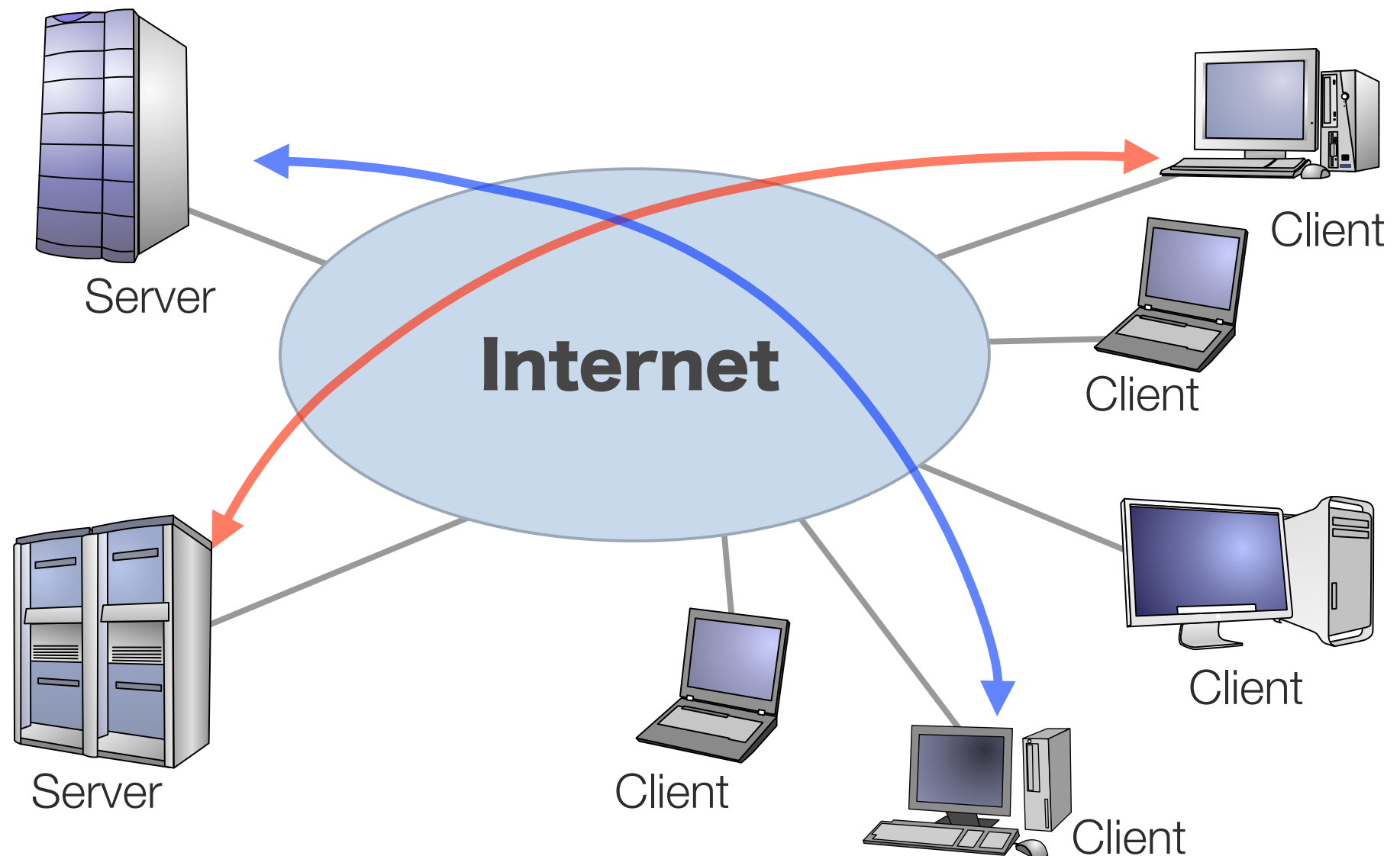
IPを利用した上で、エラー訂正などの手順を加えた手順

- SMTP (Simple Mail Transfer Protocol)
- HTTP (Hyper Text Transport Protocol)

インターネットの構造

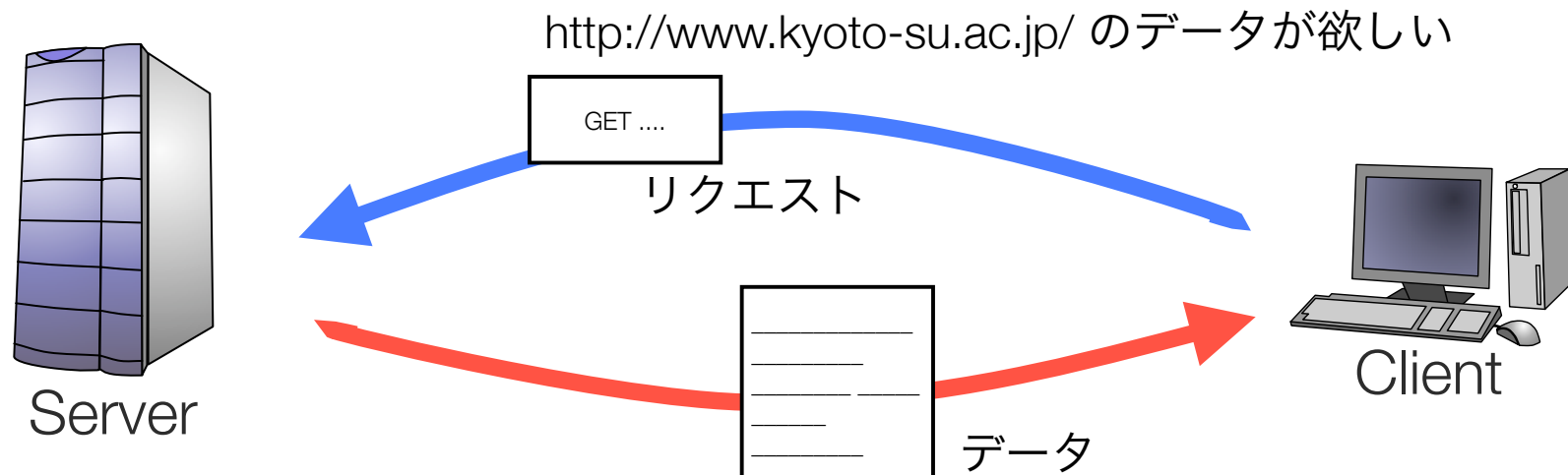


インターネットのサービスモデル



Web

- Web サーバと Web ブラウザの共同作業
- サーバ：データの蓄積と提供を担当
- ブラウザ：データの取得と表示を担当
- この種の役割分担モデルをサーバ・クライアント型と呼ぶ



ドメイン名

- IPアドレスによるアクセス

不便＋ネットワーク構成変更

- 階層的に管理されたインターネット上の名前空間

例：www.kyoto-su.ac.jp

- DNS (Domain Name System)

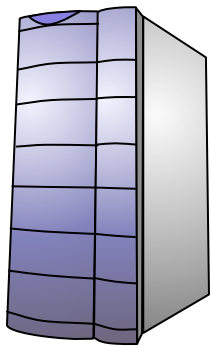
ドメイン名からIPアドレスを検索

Web サーバへのアクセス

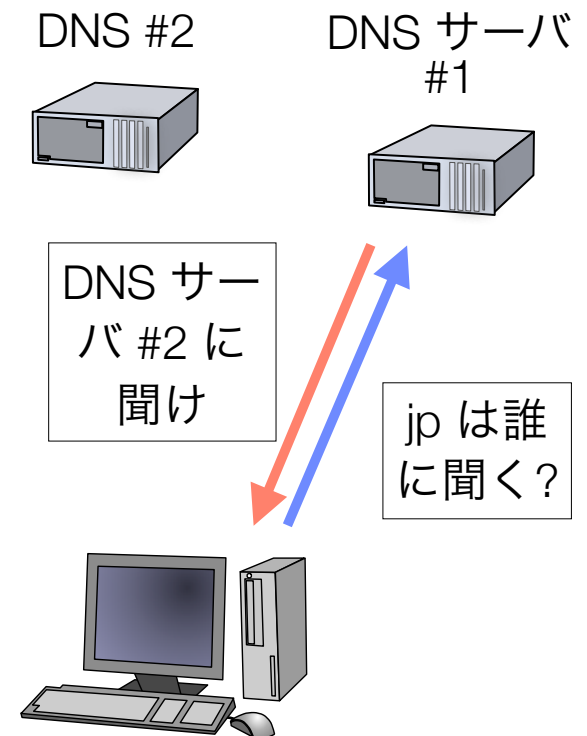
- DNS サーバへの問い合わせ
- 通信の確立
(IPアドレスとポート番号を指定)
- データの送受信
- 通信の終了

大学のWebサーバにアクセスする

1. <http://www.kyoto-su.ac.jp/> の IP アドレスは何か？
2. DNS サーバに .jp は誰に聞けば良いかを聞く
(最初の訊ね先は PC が知っている)
3. DNS サーバは訊ね先の DNS サーバの IP アドレスを答える



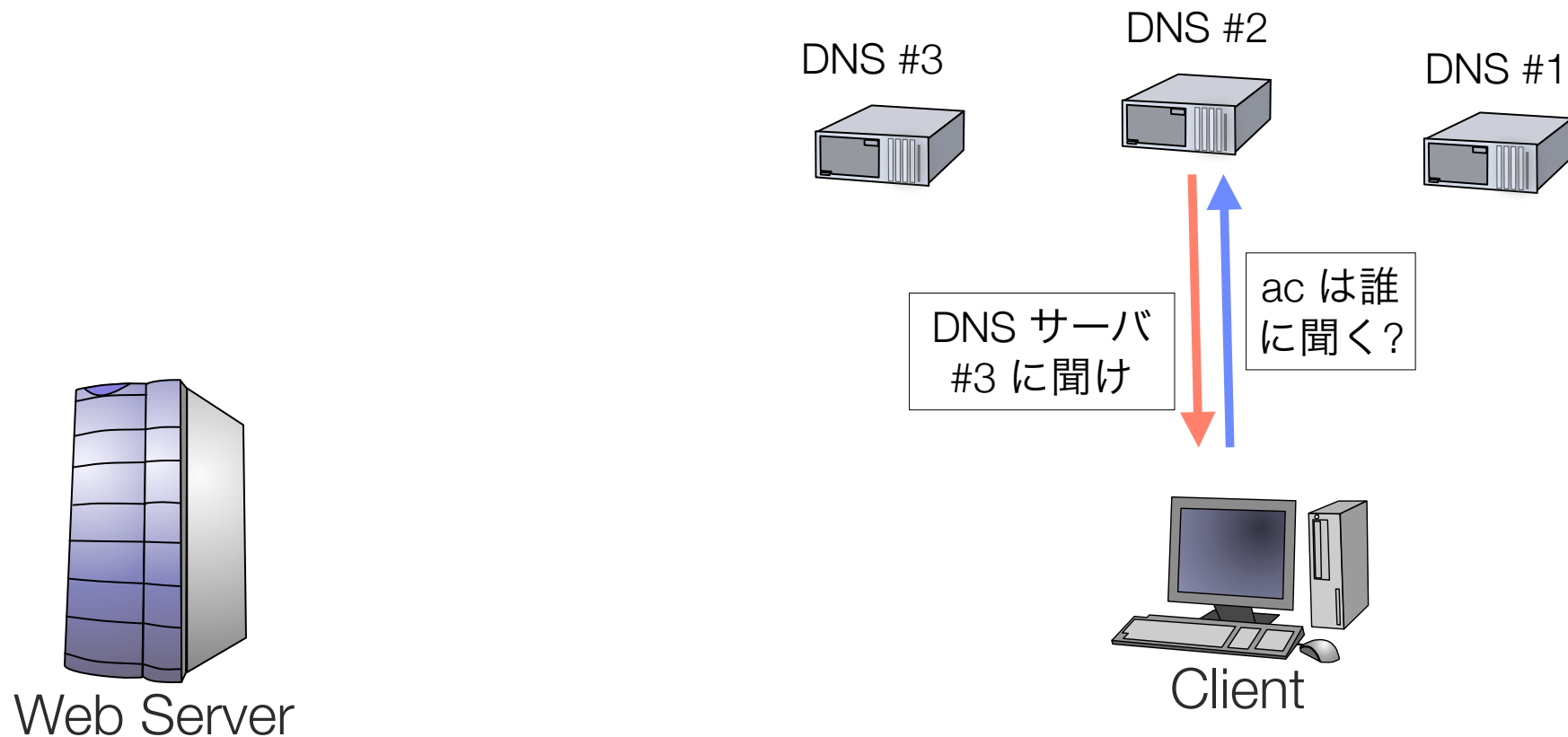
Web Server



大学のWebサーバにアクセスする

4. DNS サーバ#2 に ac は誰に聞けば良いかを聞く

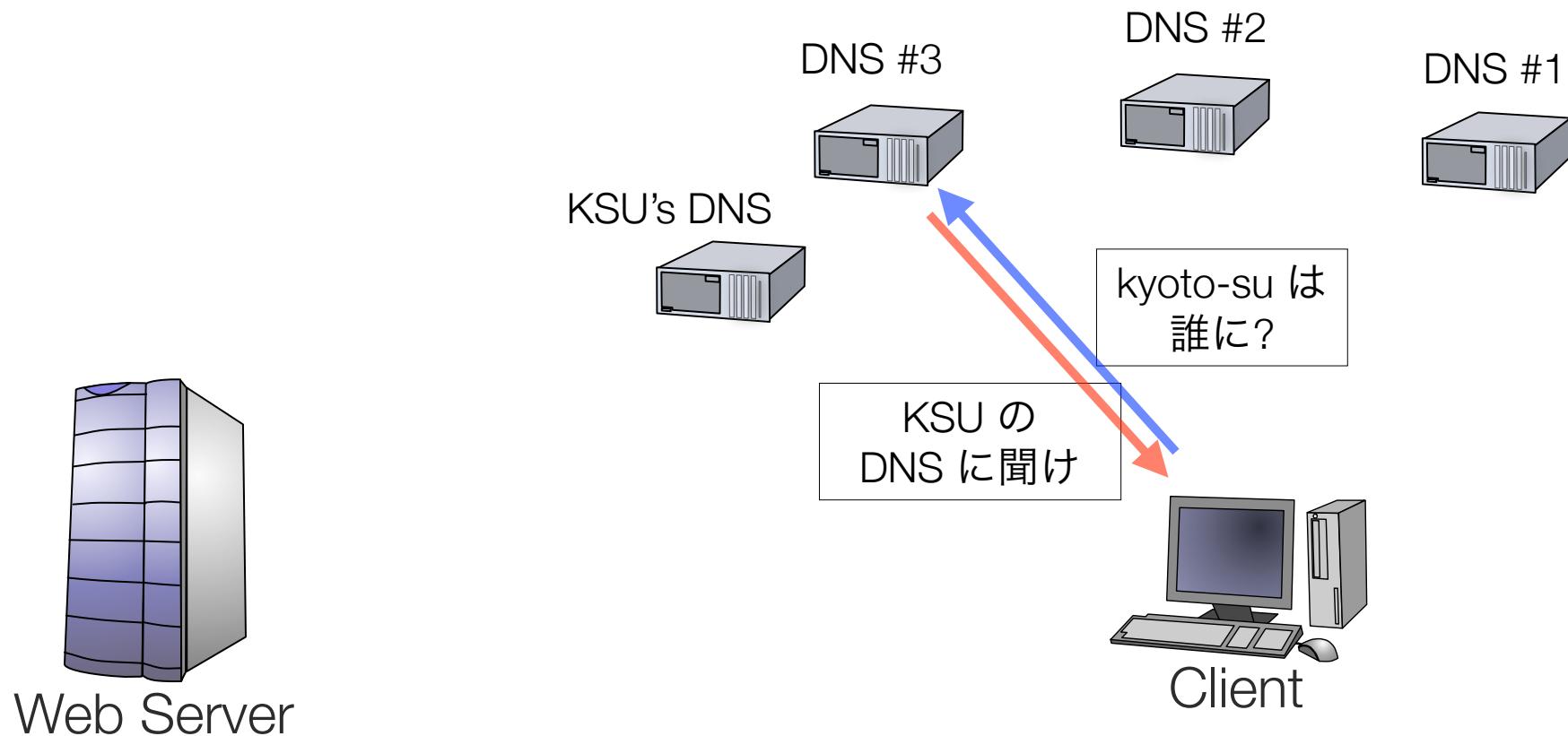
5. #2 は #3 を答えた



大学のWebサーバにアクセスする

6. DNS サーバ#3 に kyoto-su は誰に聞けば良いかを聞く

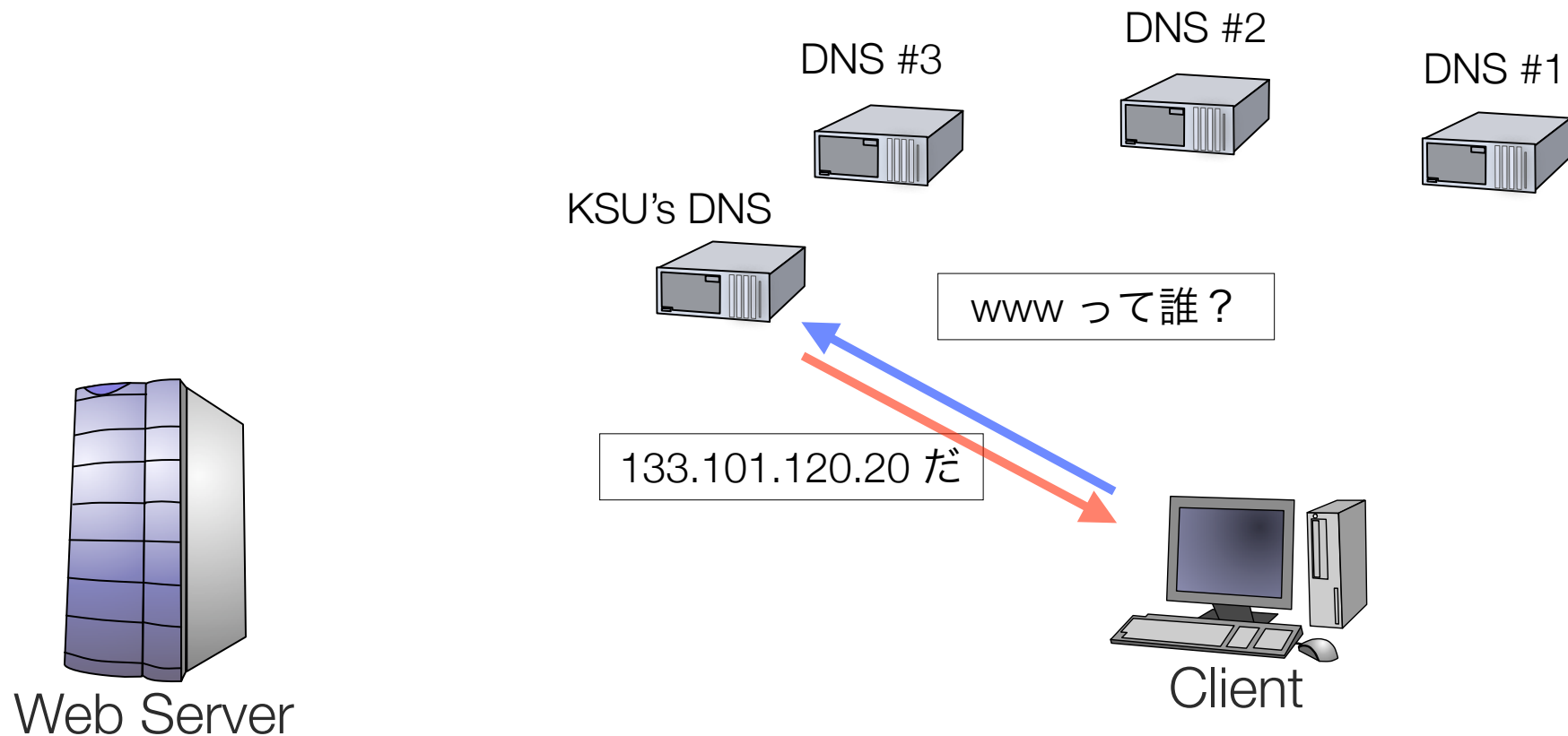
7. #3 は KSU の DNS を答えた



大学のWebサーバにアクセスする

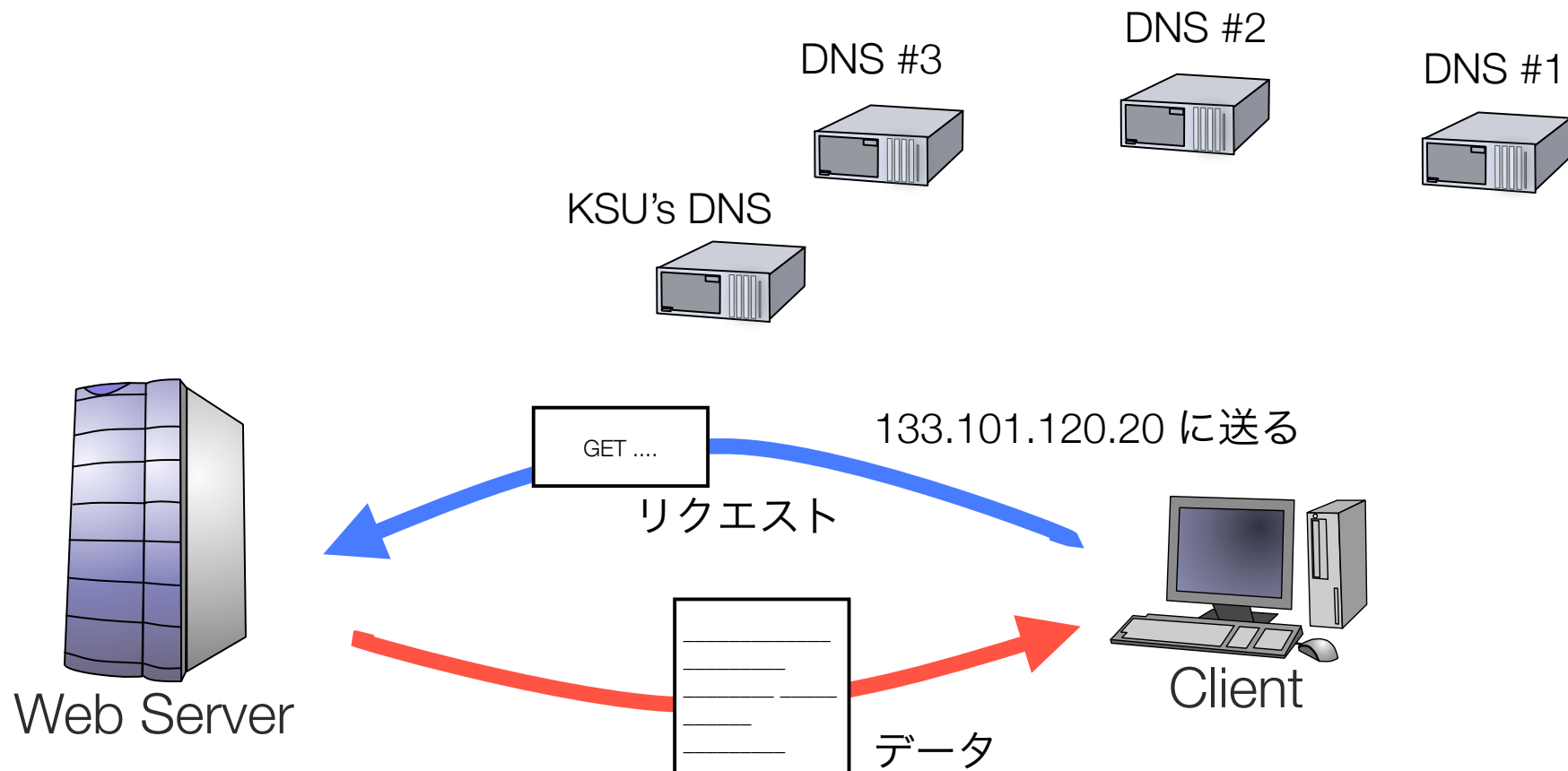
8. KSU の DNS に www の IP アドレスを聞いた

9. アドレスは 133.101.120.20 だと答えた



大学のWebサーバにアクセスする

10. (ようやく) アドレス 133.101.120.20 に Web ページ閲覧の依頼を送る
11. 依頼したページのデータが返ってきた



まとめ

- インターネットの成立

パケット交換・ARPANET

- コンピュータ・ネットワークの構成

クライアント・サーバ

- プロトコル

TCP/IP・IP アドレス

- 実際のアクセス

DNS と分散処理