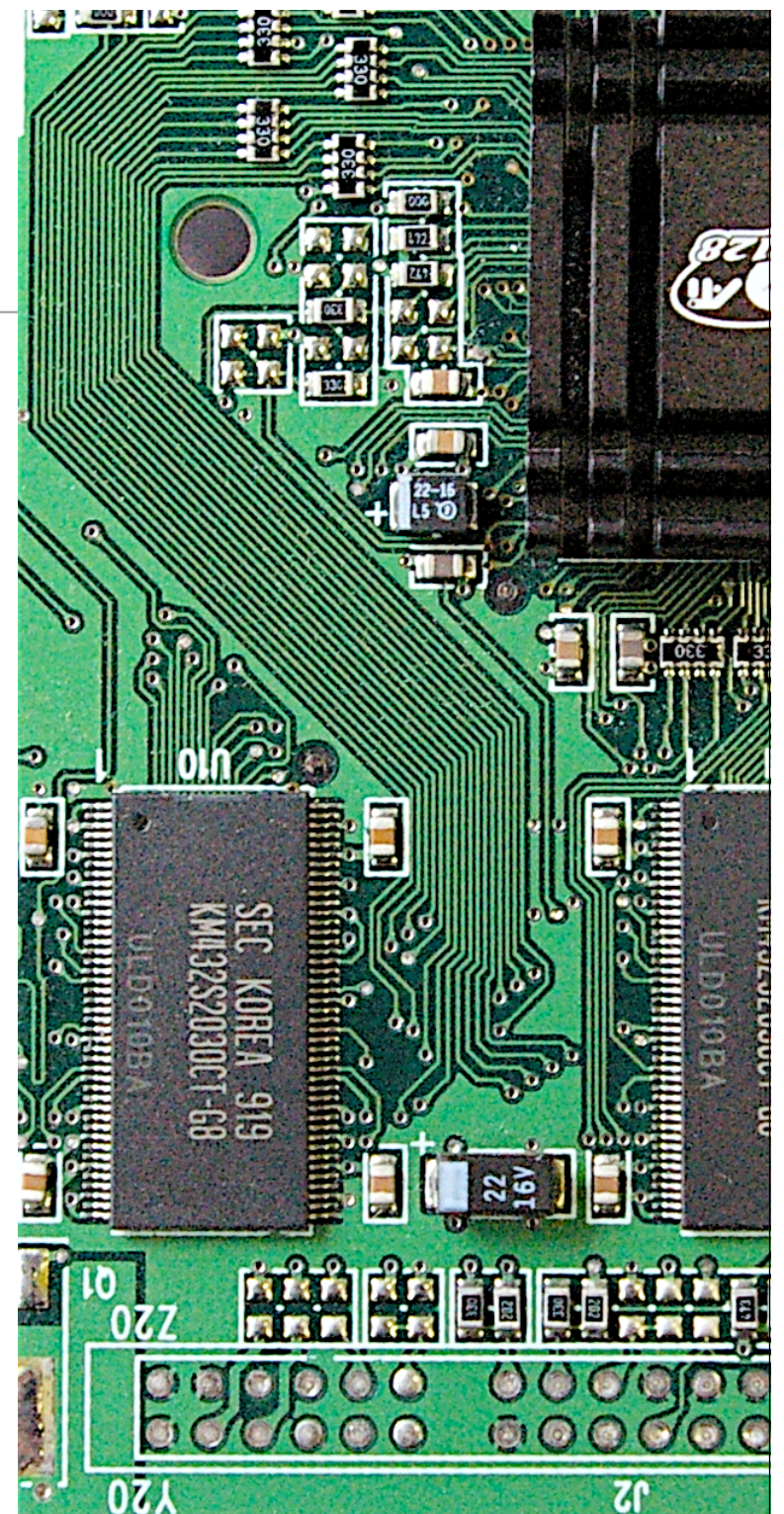


コンピュータシステムA - ハードウェアを中心に -

#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 = 配線一本
- Byte (バイト)
8bits (0-255まで)
- アルファベットは 1 バイト
256 文字以下
- 漢字は (普通は) 2 バイト
65536 文字以下



データ表現・フォーマット

- コンピュータはbitの集まりだけを処理できる

どのような情報でもbitに変えることができればコンピュータで処理できる

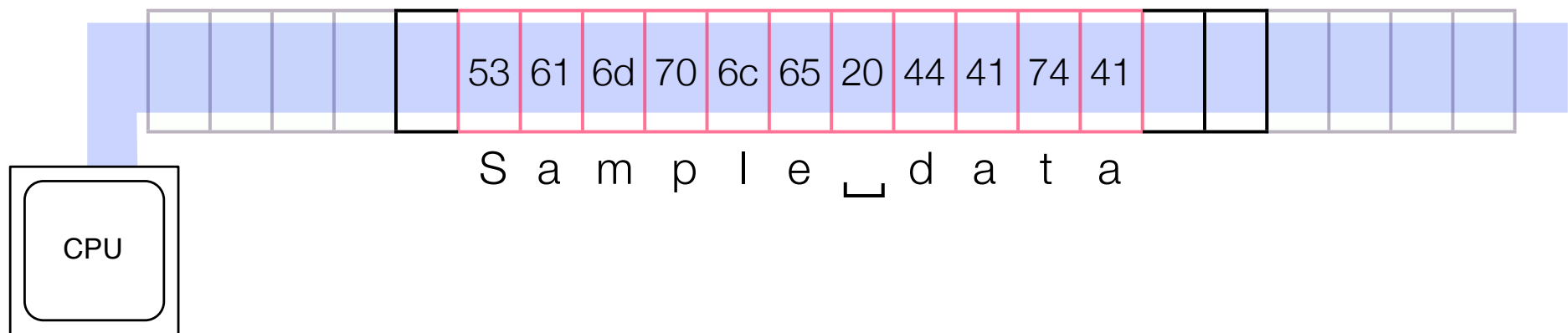
- データを bit に対応させる方法について知ろう
- フォーマット（書式）の存在

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

文字のデータ表現

1バイト文字 (ASCII encoding)



Unicode はUTF-16以外にも多くの種類がある

漢字 (Unicode encoding)

Unicode (UTF-16)

fe ff 6f 22 5b 57

BOM

漢 字

BOM : バイト順マーク (feff は正順)

音声・画像のデータ表現

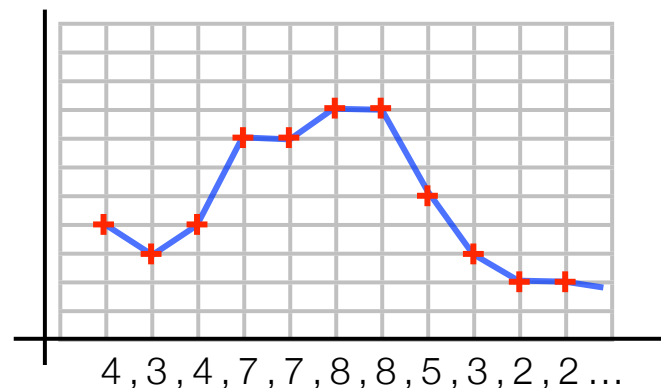
音声

サンプリング

標本化と量子化

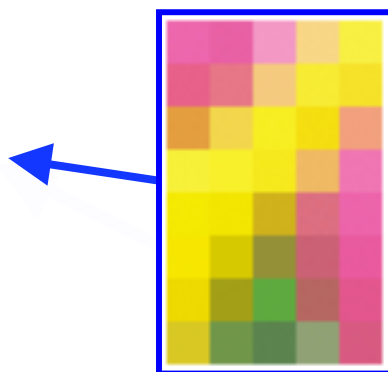
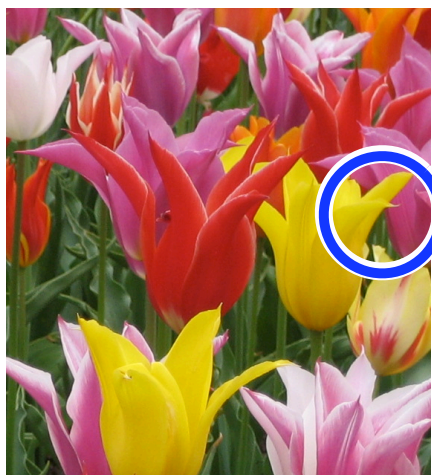
圧縮 (MP3, AAC, WMA etc. etc..)




圧縮率と品質のよりよい両立を求めて



静止画

(bitmap)



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

音声のデジタル表現

画像の圧縮 (JPEG)

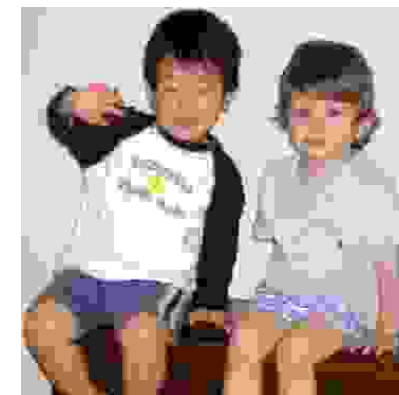


40.9KB (1/10)

品質 = 高い
データ量 = 多い



10.7KB (1/40)

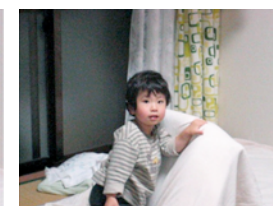
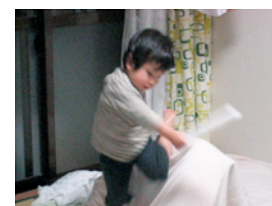
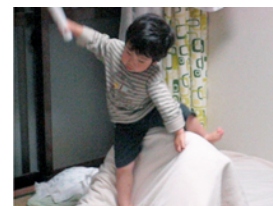


8.4KB (1/50)

品質 = 低い
データ量 = 少ない

動画

キーフレーム以降の
変化分だけ送る
= データの節約
= 圧縮



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

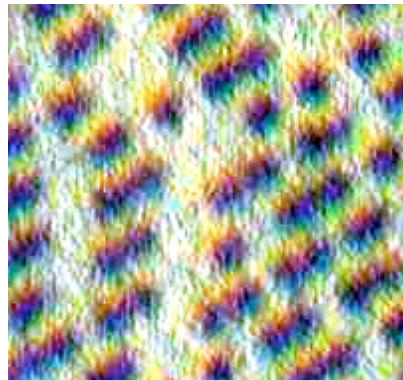
デジタル化による利益

デジタル化による利益

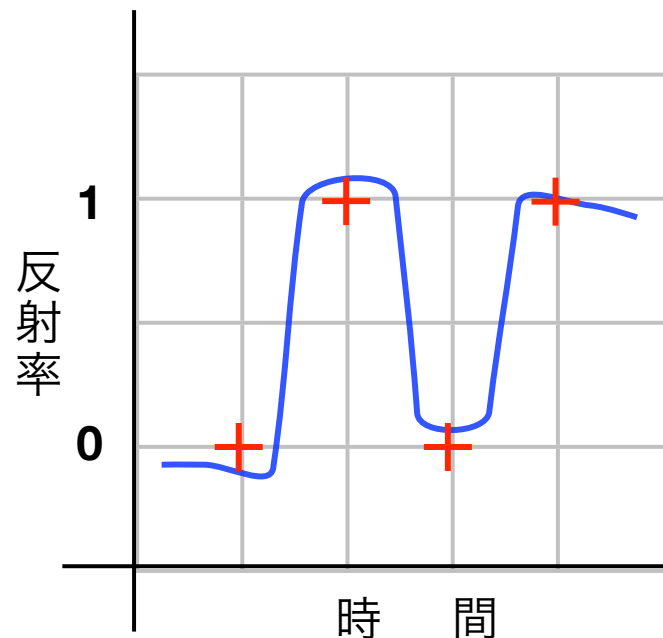
- 数値による表現
 - 文字・映像・音声
 - 連続的な値の変化ではなく、離散的な数値として表現
- そうすることの利益は？

ノイズへの抵抗（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



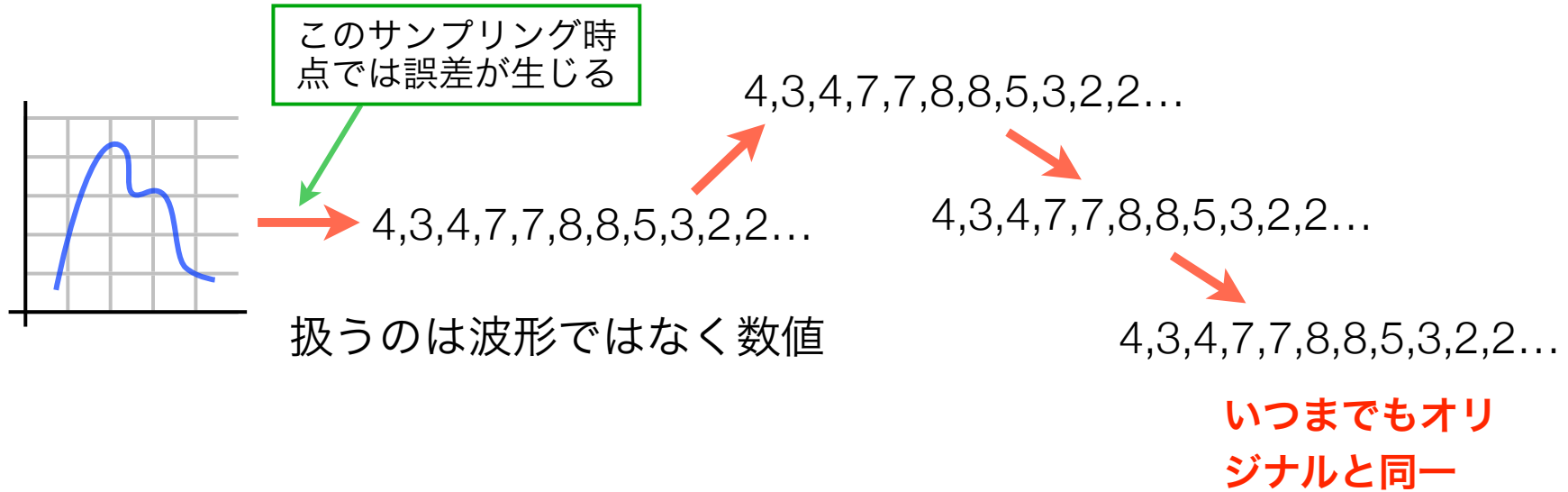
CDのピット長は9種類



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

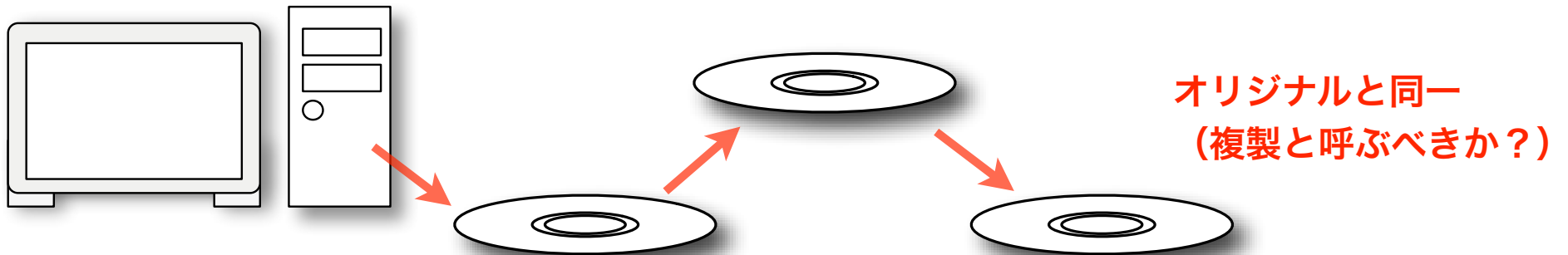
最大ピット長の1/9以下程度の誤差で長さ検出できれば良い

完全な複製（復習）



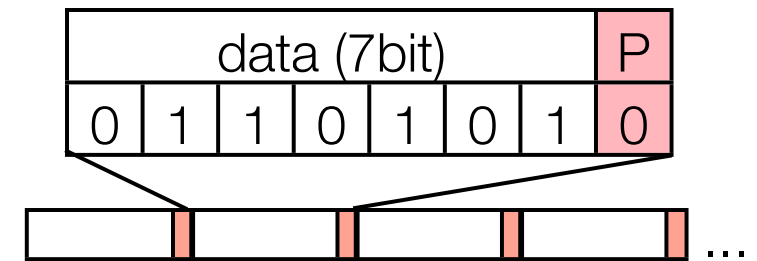
CD/CD-R

パソコンを使って CD / CD-R 間の複製をとる



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（偶奇性） - 1 bit 付加



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)

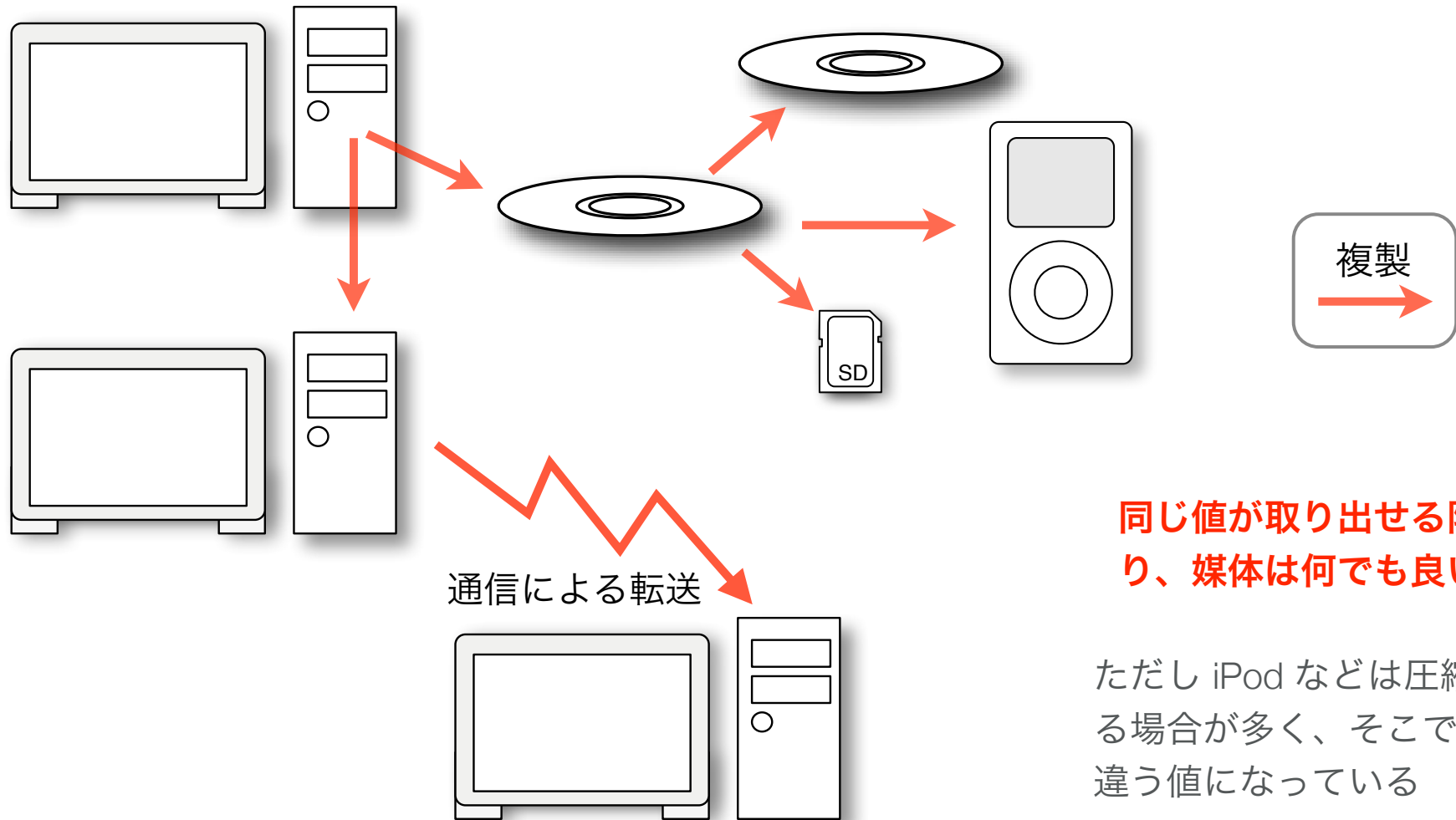
- CIRC : CD
- より多くの付加情報によってより広範囲な修正に対応

*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

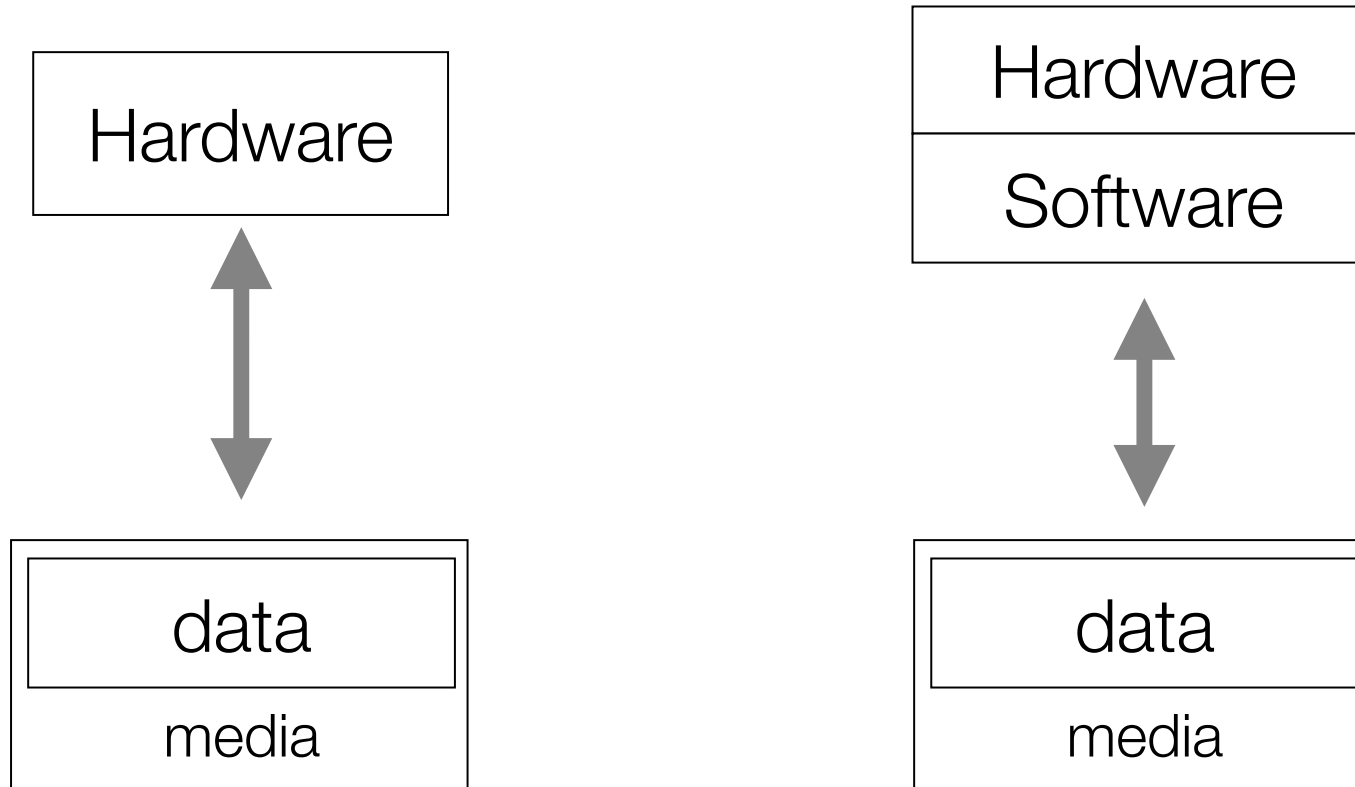


まとめ：デジタルデータの特徴

- 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適用可能に：圧縮・エラー訂正
 - コンピュータによる知的な自動処理が可能に

デジタルシステムの柔軟性

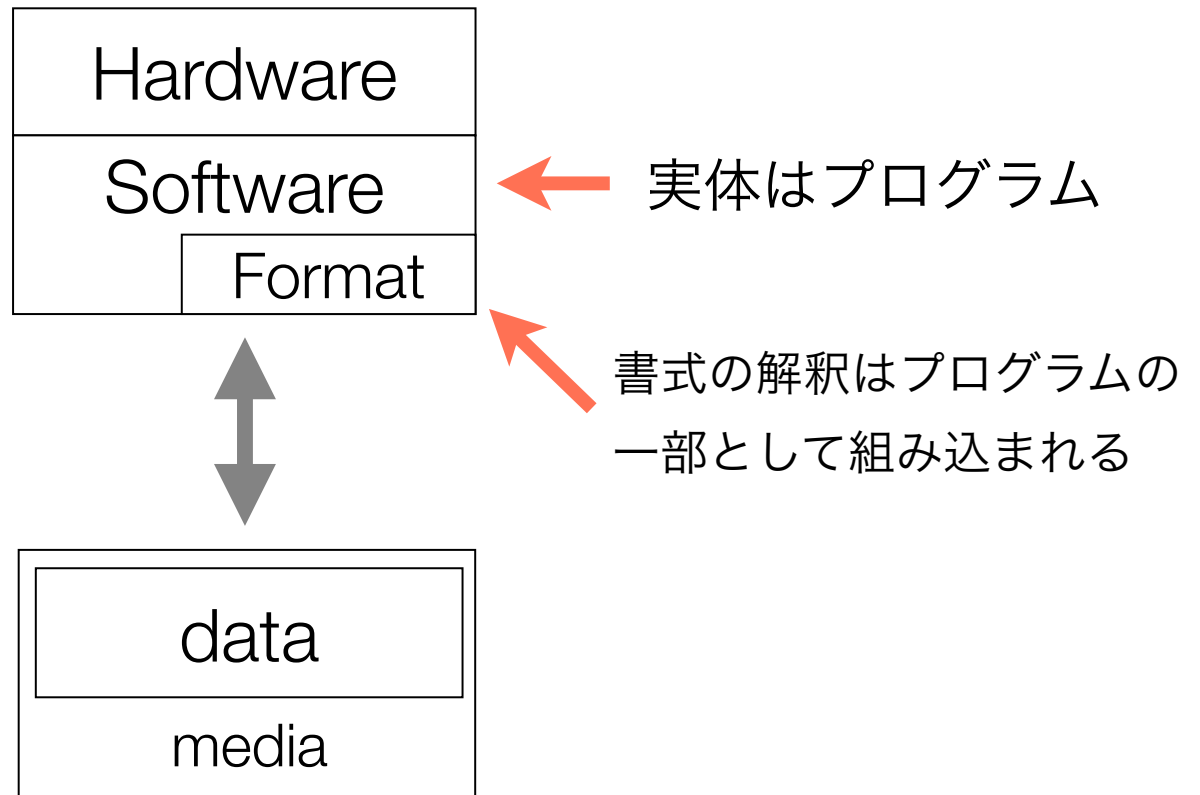
アナログシステムとデジタルシステム



典型的なアナログシステム
(レコードプレーヤーなど)

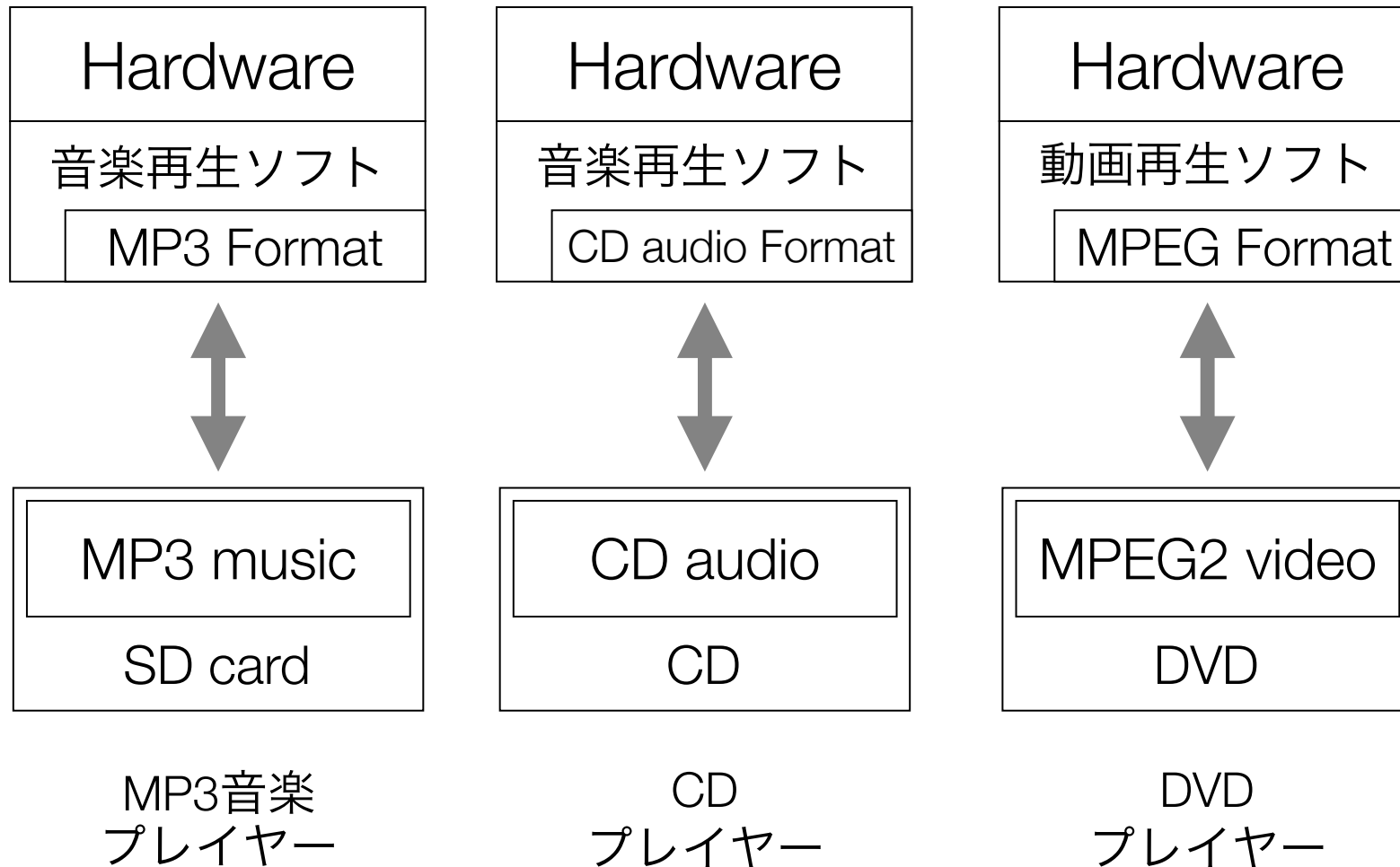
典型的なデジタルシステム
(コンピュータなど)

書式とデータの関係

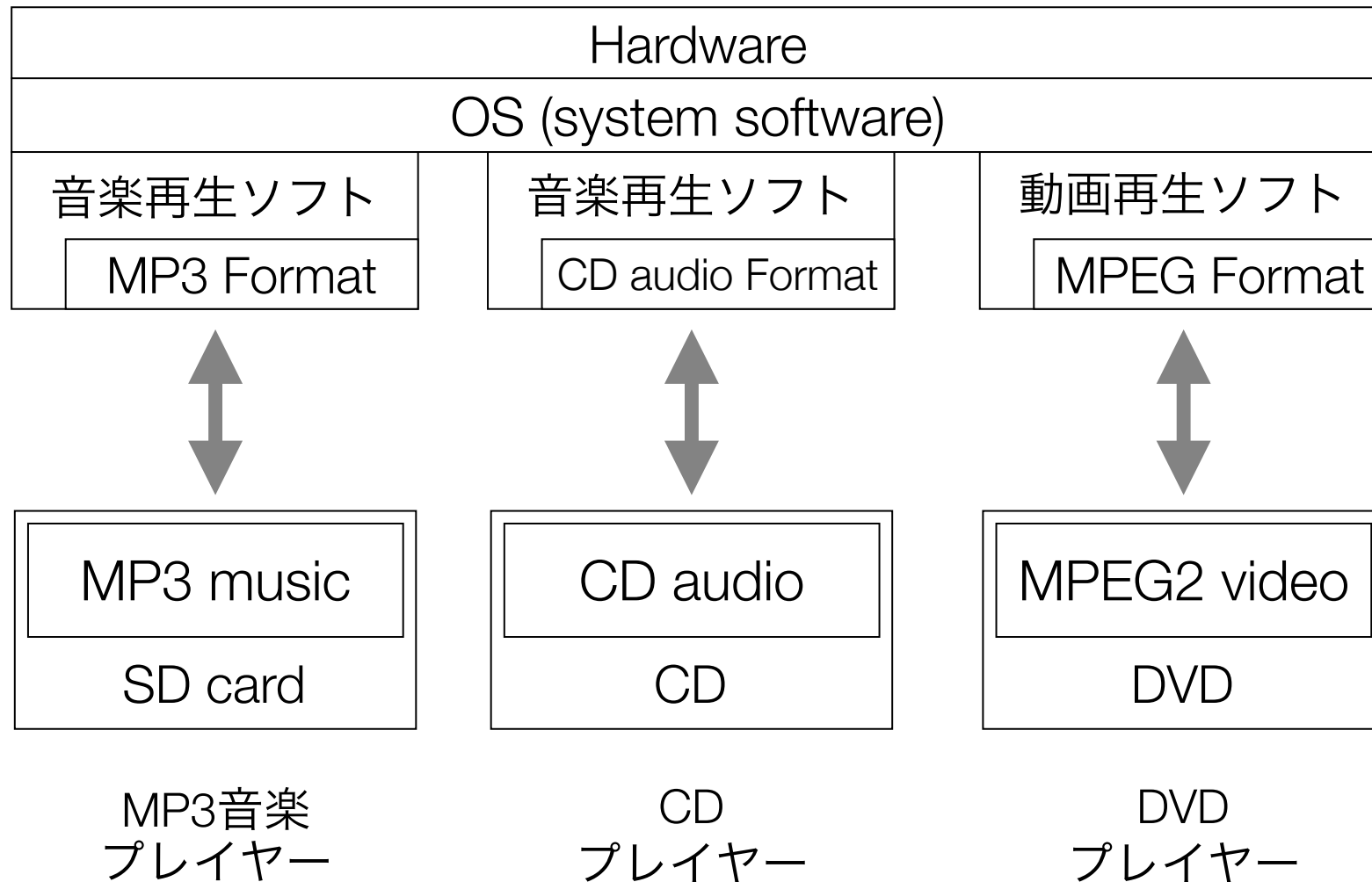


典型的なデジタルシステム
(コンピュータなど)

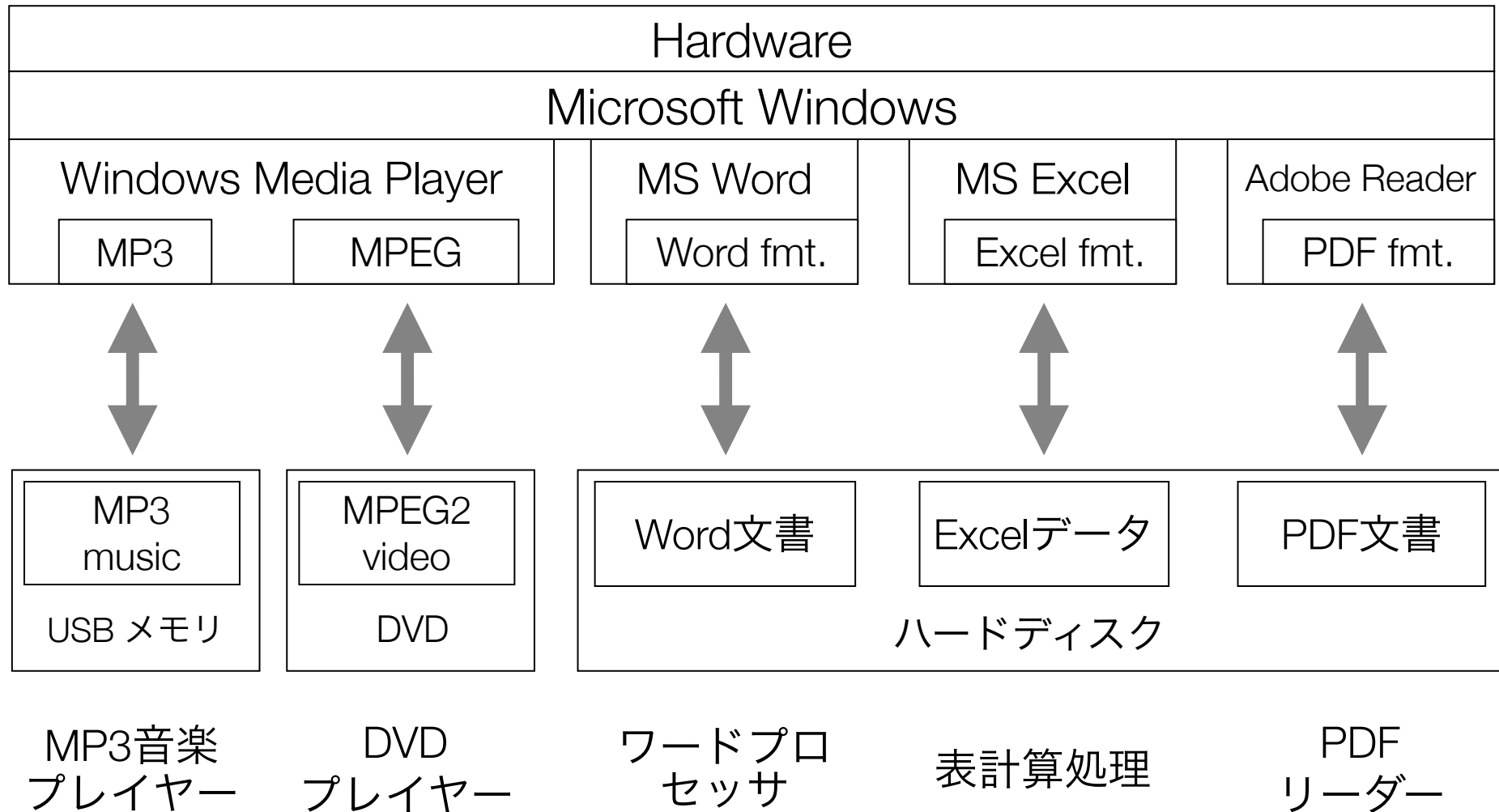
デジタルシステムの柔軟性



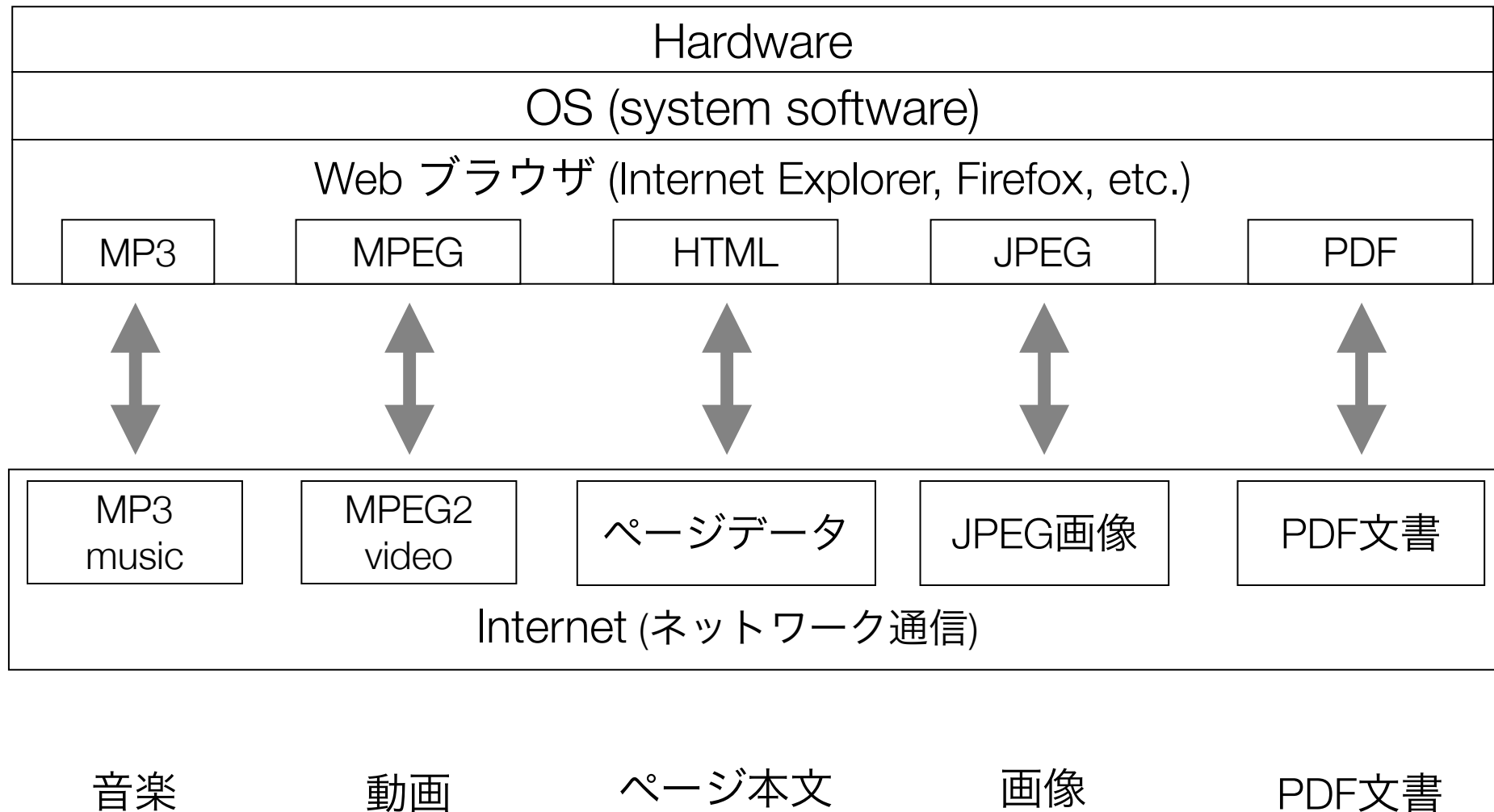
PC : 汎用デジタル処理システム



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



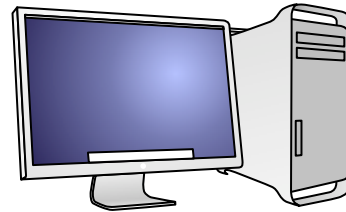
デジタル化のインパクト

- 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

事例：Web アプリケーション

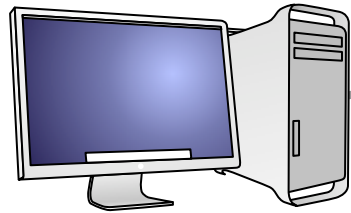
アプリケーションの実行形態

Stand alone



アプリケーションの実行形態

online



Internet



Local application(s)

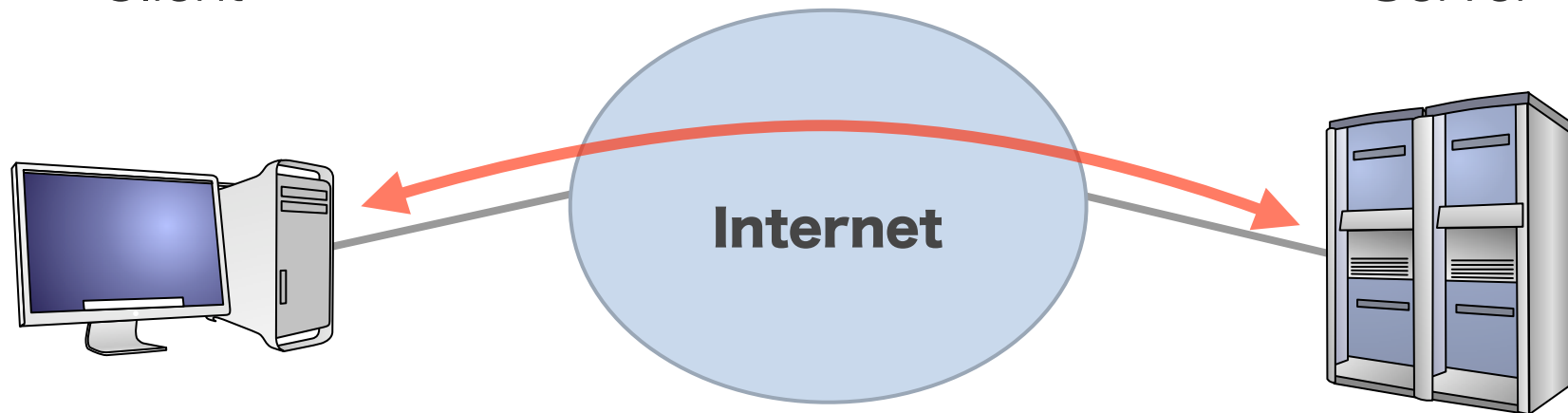


Web
browser

Web アプリケーション

Client

Server



Web browser



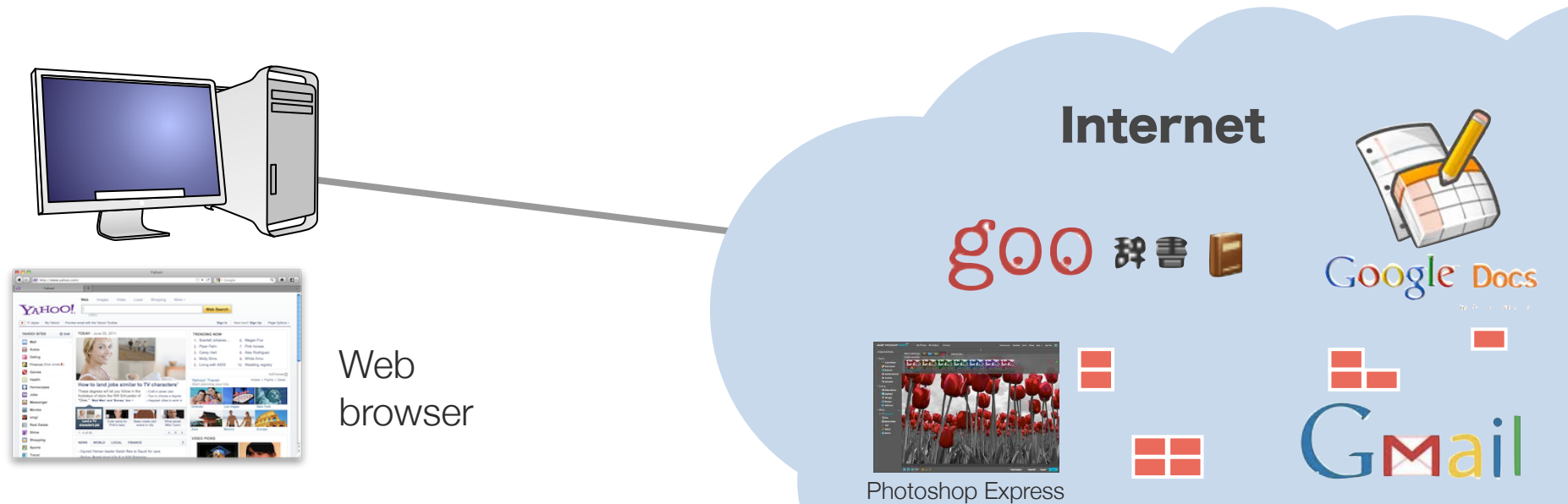
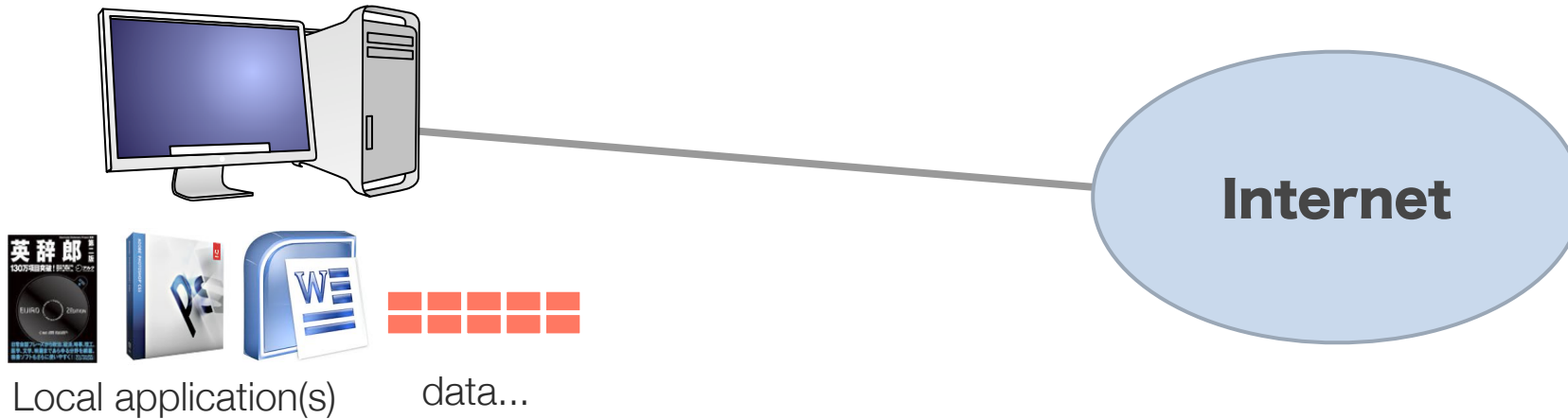
Google Docs
<http://www.goo.com>



Photoshop Express



ローカルアプリ vs Web アプリ

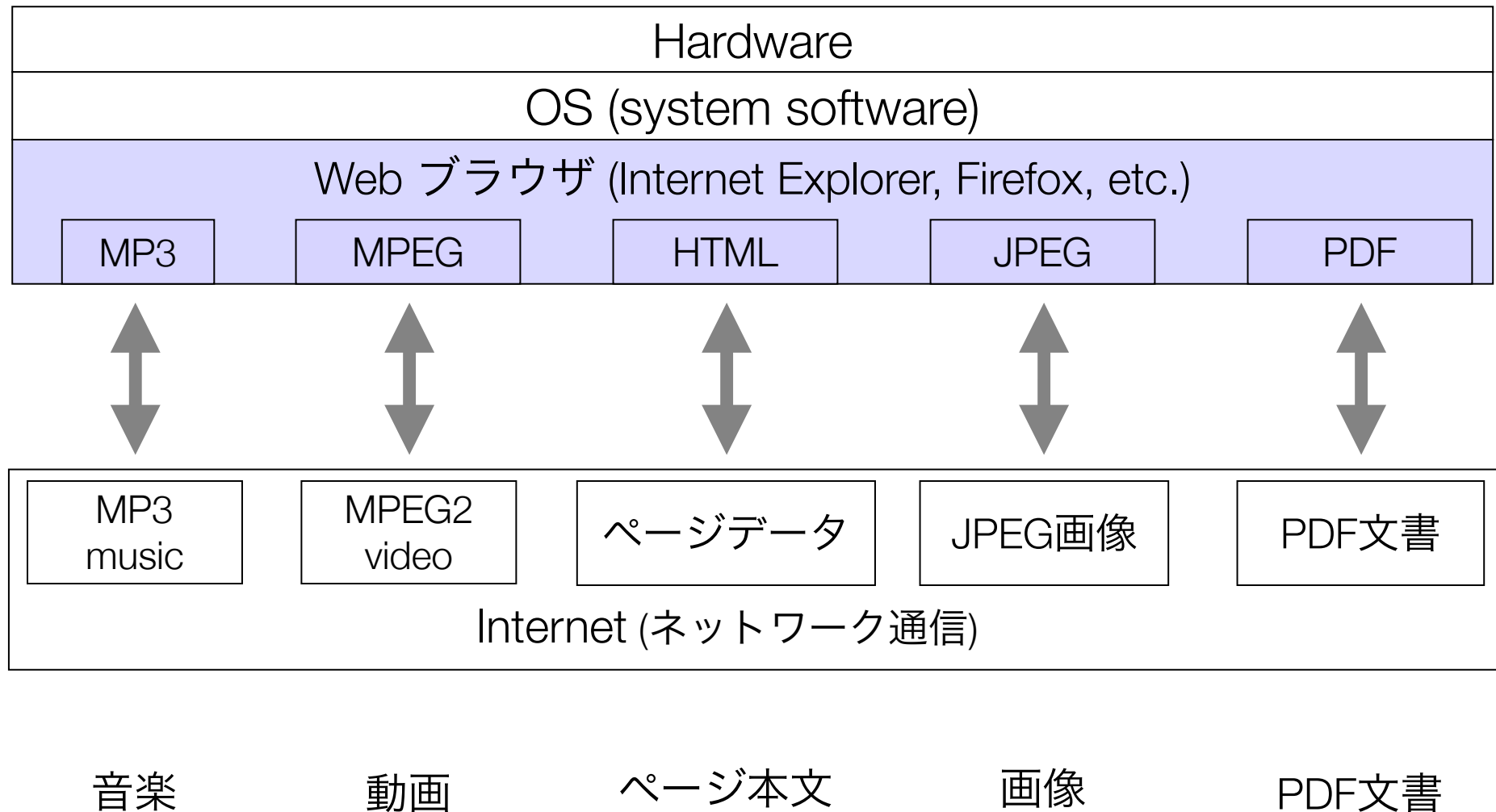


ローカルアプリ vs Web アプリ

- ローカルアプリケーション
 - 手元のコンピュータにインストールして利用
 - パッケージ販売ビジネスモデル
- Web アプリケーション
 - ネットワークアクセス必須
 - インストール不要 (Web ブラウザさえあれば良い)
 - 広告や有料会員による収入
 - 現状では機能・性能でローカルアプリより劣りがち(※)
- 徐々に Web アプリが増えつつある

(※) ブラウザ上でJavaScriptで実行すること、データアクセスの通信遅延などが原因

Webブラウザ only (専用 OS) の可能性



Web アプリケーション

Client

Server

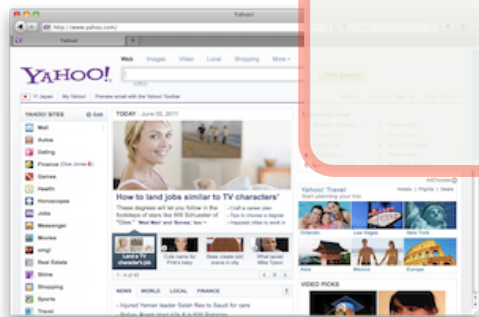
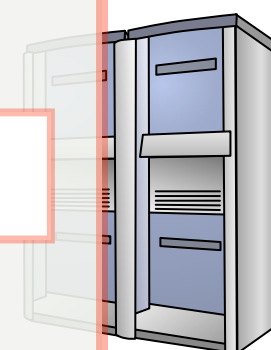
HTML

GIF, JPEG, PNG, MPEG, H.264, AIFF, WAV

JavaScript + DOM, etc. etc..

HTML5

Web
browser



Google Docs
<http://www.google.com>



Photoshop Express



Firefox



Safari



Chrome



Internet Explorer

事例：HTML5 の動画フォーマット

HTML5 の動画フォーマット

- Brendan Eich
 - JavaScript の開発者
 - Mozilla で長くブラウザ開発
 - 現在も Firefox を開発
- 2008 取材：
 - データを棺桶に入れるな
 - フォーマットはまだいい
 - 問題はロイヤリティだ



HTML5 の動画フォーマット

- 共通の動画フォーマットが欲しい
- フリーの動画フォーマット Ogg/Theora でどうか
 - Mozilla (Firefox) : Ogg/Theora に対応、ロイヤリティが必要な H.264 には反対
 - Apple : Ogg/Theora は特許の懸念、ハード対応無しのため反対、H.264 に対応
 - Microsoft (IE) : no comment → H.264 対応へ
 - Google (Chrome) : Ogg/Theora と H.264 の両方に対応
- 結局 HTML5 仕様では標準フォーマットは決まらず
- 2010/2 Google による On2 Technologies の買収
 - On2 は Ogg が使っていた VP3 の開発元
 - この VP8 をロイヤリティ・フリーにして公開か？

WebM プロジェクト

- On2 Technologies を Google が買収して実現

検索を基幹技術とする Google にはフォーマットを共通化し「棺桶に入れない」ことがとても重要

- 2010年5月の Google I/O イベントで発表

動画 VP8 + 音声 Vorbis の組み合わせ

仕様は公開、コードはオープン化

利用に関するロイヤリティなし

- これを（デファクト）標準となるよう広めたい

パテントの壁

- MPEG LA による H.264 のライセンス管理

パテント・プール：関連特許を集めてライセンスを一括管理

所定のロイヤリティを払えばそれ以上の問題が起きない

- Ogg Theora の問題

現在は使用料なし

将来特許が顕在化し権利者から提訴されるかも知れない

- VP8 は本当にパテント・フリーか？

事例：Forgent による JPEG 特許

- JPEG 画像フォーマット

利用無料のために長期間使われ普及

標準化目的の委員会で作られ、特許の存在は想定外

しかし委員会メンバーだった企業が1987年に出願

- Forgent Networks によるライセンス料の要求

2002年にForgentがライセンス料を要求し、SONY 1500
万ドルなど数社が支払った

2004年には31社を提訴

- 2006年に再審査請求が認められ、一部却下になる

同年、Forgent は訴訟を取り下げ

事例：Unisys による GIF (LZW) 特許

- GIF 画像フォーマット

利用無料のために1987年から長期間使われ普及

- Unisys によるライセンス料の要求

1985年、LZW (Lempel Ziv Welch)圧縮の特許が成立

1996年、GIF が LZW を利用し、その特許に抵触すると
して利用料をエンドユーザやソフト開発者に要求

2003年に米国での特許の有効期限が切れた

2004年に日本での期限も切れた

Google の「特許破壊的」行動

- YouTube の動画を WebM に変換作業中

そのうちにH.264で出てこなくなるかもしれない

Chrome ブラウザは対応しているので彼らの環境では問題なく再生できる

- ライセンス条項

「VP8が特許を侵害しているという訴訟を起こしたら、VP8の利用に必要な特許の使用権を剥奪する」

MPEG LA の必須特許募集に対抗できるか？

Brendan Eich の Blog

Can we win the long war? I don't know if we'll see a final victory, but we must fight on. Patents expire (remember the LZW patent?). They can be invalidated. (Netscape paid to do this to certain obnoxious patents, based on prior art.)
不愉快な 先行技術

They can be worked around. And patent law can be reformed.

長期戦
Mozilla is here for the long haul.
We will never give up, never surrender.



Video, Mobile, and the Open Web, 2012 March 18. Brendan Eich.
<http://hacks.mozilla.org/2012/03/video-mobile-and-the-open-web/>

HTML5 動画フォーマットのまとめ

- 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？

パテントの問題

- 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

- たかがフォーマット

しかしそれが影響を及ぼす範囲はとて大きい