

SCORMからTin Canへ

—eラーニング標準規格の過去と未来—

千葉工業大学
仲林 清

あらまし

◆ TinCanの位置づけ・意味

- eラーニング標準化の流れ
- eラーニングを取り巻く技術・ビジネス環境
- eラーニング技術やビジネスに対する新たな付加価値

目次

◆ SCORMとは？ Tin Canとは？

- AICC⇒SCORM⇒Tin Can

◆ APIとは？

- いまどきのWebサービス
- eラーニングにおけるAPI

◆ 技術標準規格の価値？

- 技術アーキテクチャとビジネスアーキテクチャ

◆ 今後の展開

[Search](#)[Tin Can Explained ▼](#)[Tin Can Solutions ▼](#)[Developers ▼](#)[Blog ▼](#)

Tin Can API Developers

Prototypes & Docs

[Learn More](#)

- Download the spec
- Prototypes
- Docs
- Statement generator/viewer
- Forums

Tech Overview

[Learn More](#)

- Overview of the API
- Statements 101

Generating Statements

[Learn More](#)

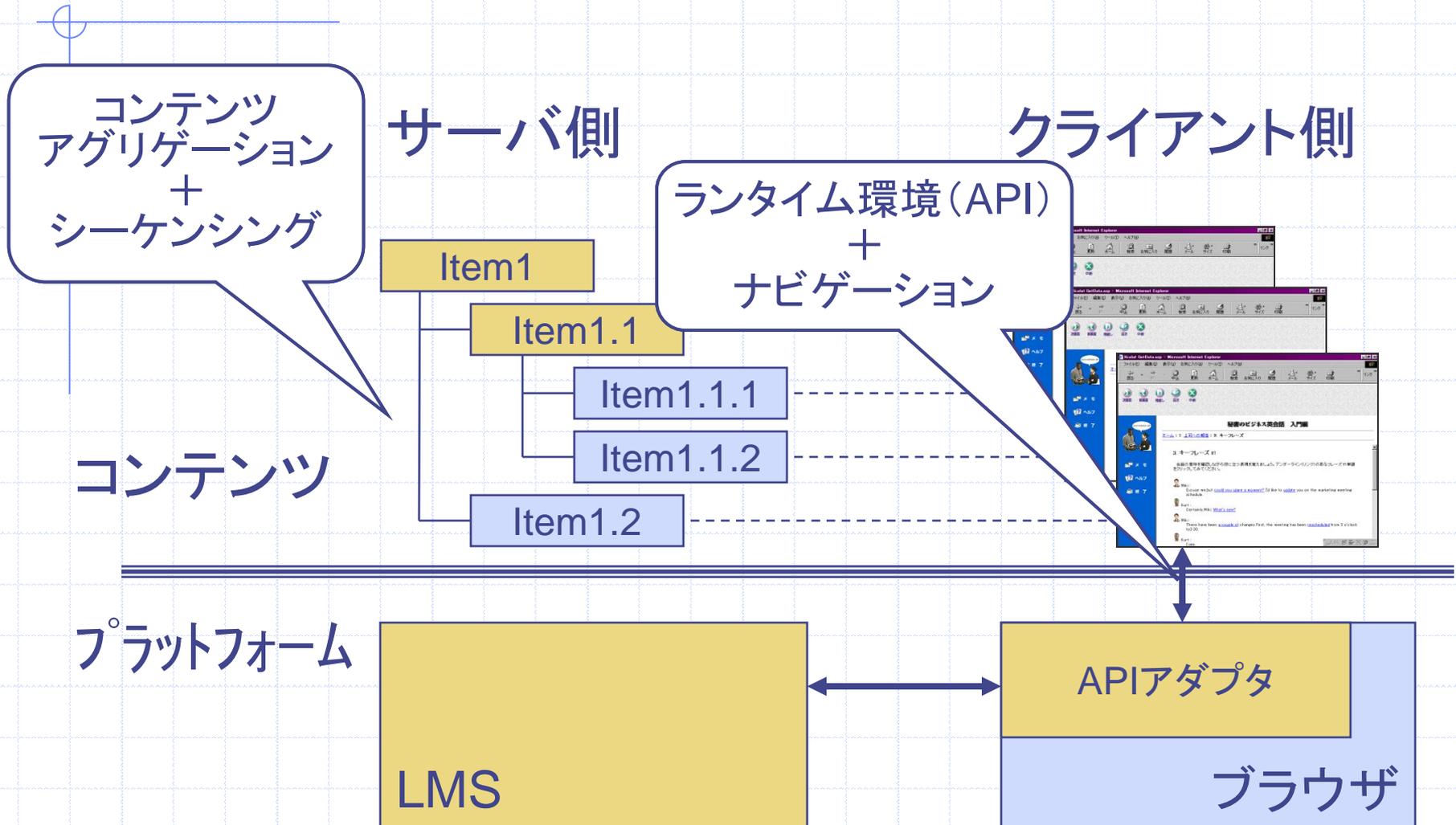
- How to generate statements
- Statement Generator
- Statement validator

Our Hosted Sandbox

[Learn More](#)

- Easiest way to experiment
- Sandbox for debugging
- Free to try

SCORM 1.2 ⇒ SCORM 2004



AICC ⇒ SCORM ⇒ Tin Can

時々の技術の影響を受ける

SCORMとの組み合わせが可能

スマホアプリ
タブレット
などもあり

クライアントはフルブラウザ

	AICC	SCORM 1.2	SCORM 2004	Tin Can
時期	1993~98	2000	2004~09	2013
コンテンツ構造記述	Ini/CSV	XML	XML	対象外
API	HTTPベース	JavaScript	JavaScript	REST/JSON/OAuth
シーケンシング	限定的	限定的	ルール記述	対象外

SCORM APIデータモデルの Tin Canでの実装

Appendix C: Example definitions for Activities of type "cmi.interaction"

true-false

```
"definition": {
  "description": {
    "en-US": "Does the xAPI include the concept of statements?"
  },
  "type": "http://adlnet.gov/expapi/activities/cmi.interaction",
  "interactionType": "true-false",
  "correctResponsesPattern": [
    "true"
  ]
}
```

choice

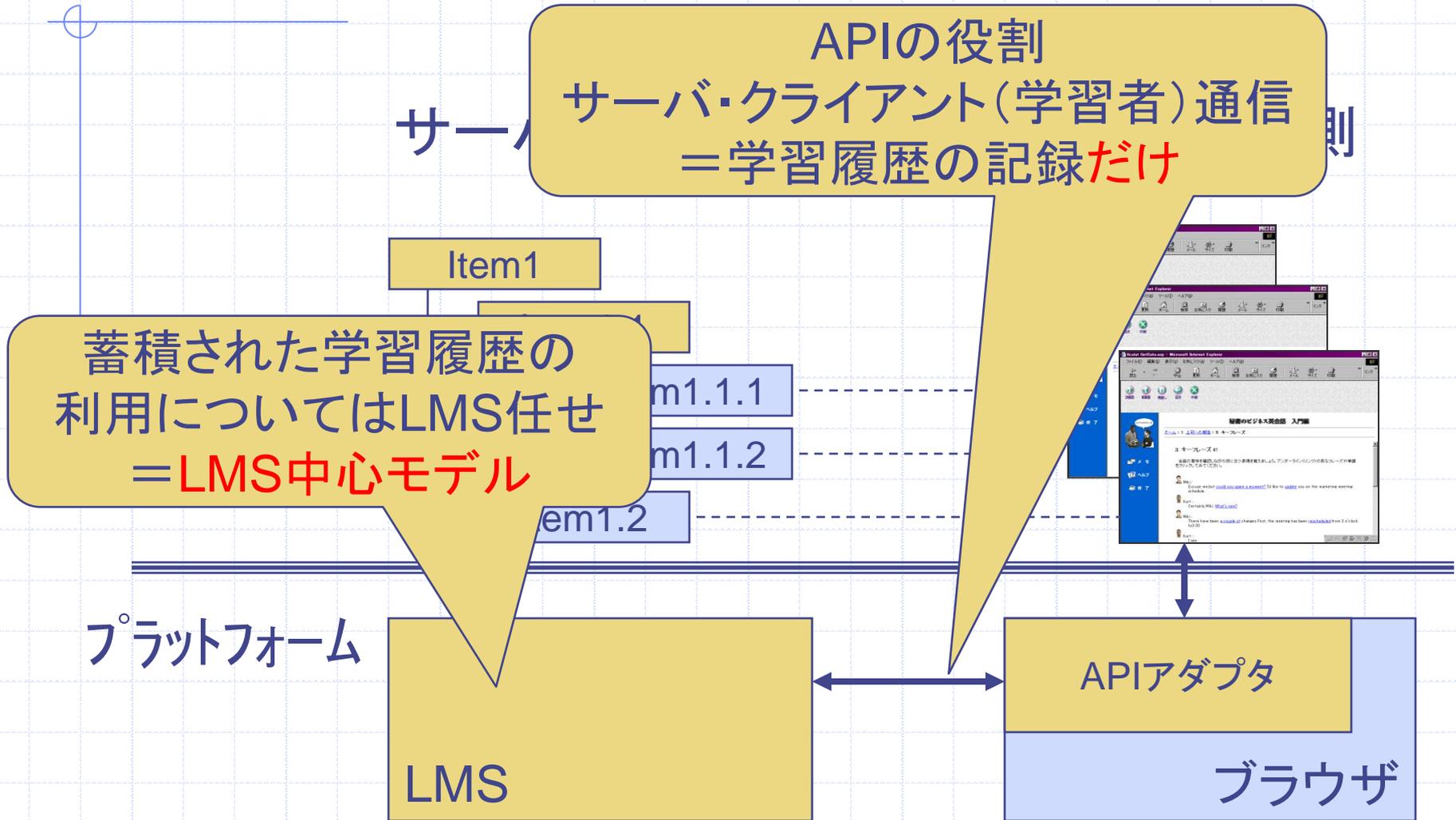
```
"definition": {
  "description": {
    "en-US": "Which of these prototypes are available at the beta site?"
  },
  "type": "http://adlnet.gov/expapi/activities/cmi.interaction",
  "interactionType": "choice",
  "correctResponsesPattern": [
    "golf[.].tetris"
```

choice

```
"definition": {
  "description": {
    "en-US": "Which of these prototypes are available at the beta site?"
  },
  "type": "http://adlnet.gov/expapi/activities/cmi.interaction",
  "interactionType": "choice",
  "correctResponsesPattern": [
    "golf[,]tetris"
  ],
  "choices": [
    {
      "id": "golf",
      "description": {
        "en-US": "Golf Example"
      }
    },
    {
      "id": "tetris",
      "description": {
        "en-US": "Tetris Example"
      }
    },
    {
      "id": "scrabble",
      "description": {
        "en-US": "Scrabble Example"
      }
    }
  ]
}
```

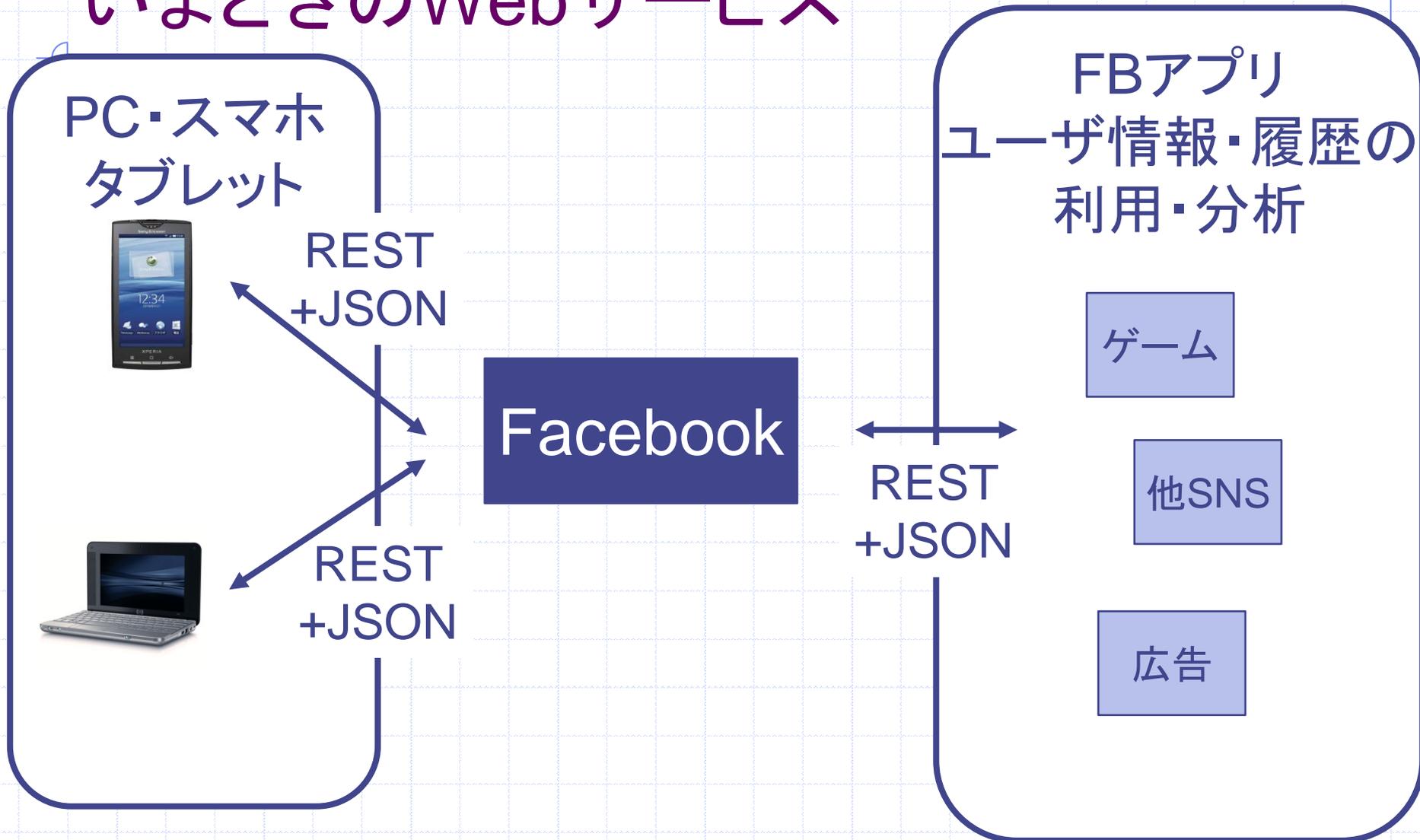
APIとは？

従来のAICC/SCORM



APIとは？

いまどきのWebサービス



APIとは？

Tin Can API

PC・スマホ
電子教科書



Tin Can
=REST
+JSON



Tin Can
=REST
+JSON

LRS
Learning
Record
Store

他のLMS
サーバ・アプリ

ユーザ情報・履歴の
利用・分析

学習履歴・成績
ポートフォリオ

Tin Can
=REST
+JSON

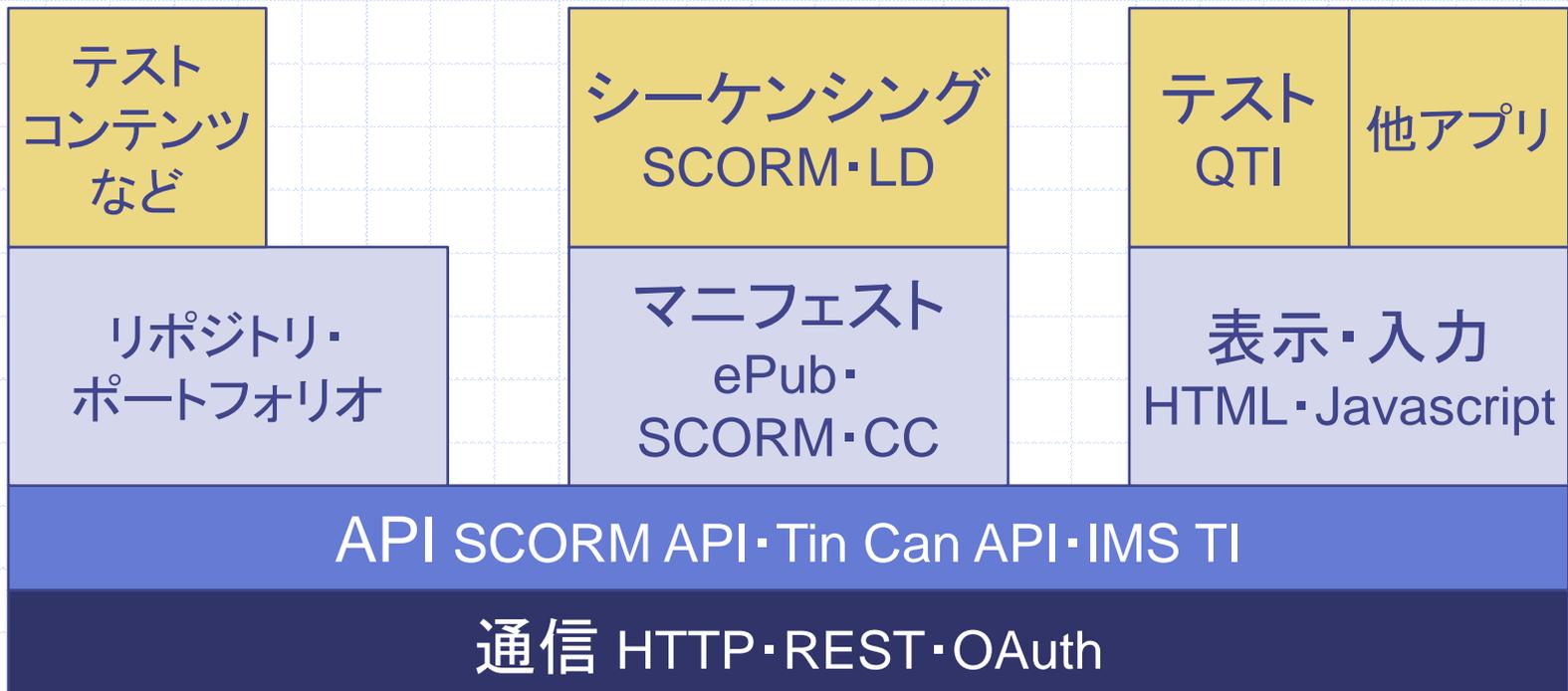
Data Mining
Learning Analytics

APIの位置づけ

他サーバ

サーバ

クライアント



標準規格の価値

◆規格の価値は何で決まるのか？

◆モジュール化の理論

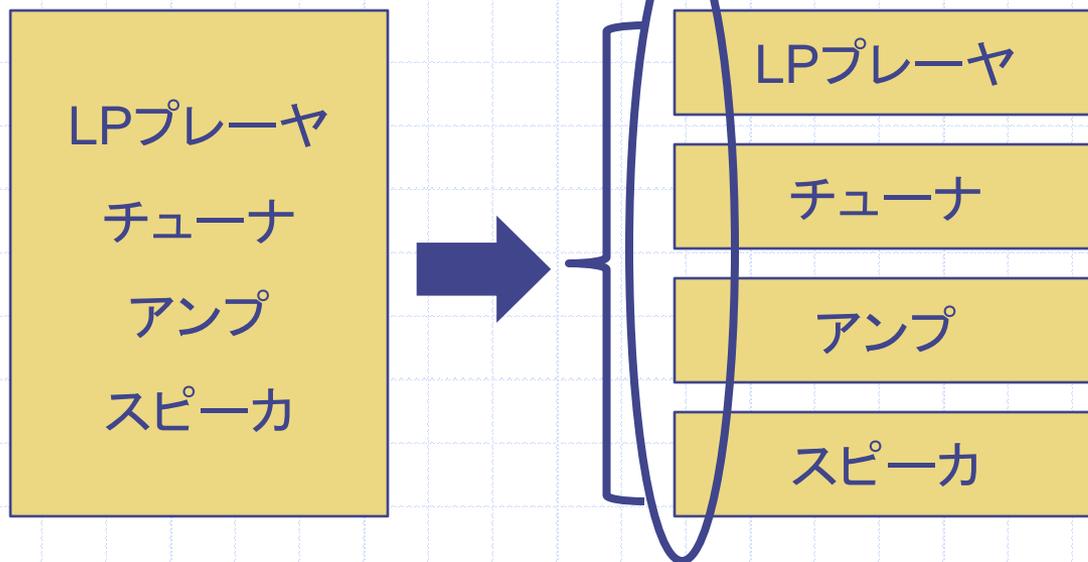
モジュール化オペレータ

- ◆ 分離
- ◆ 交換
- ◆ 追加
- ◆ 削除
- ◆ 抽出
- ◆ 転用

- ボールドウィン, クラーク. デザインルール. 東洋経済

分離

- ◆ モジュールをどのように分離するか決める＝相互依存性を排除する
- ◆ モジュール間のインターフェース(＝デザインルール)を決める
- ◆ 一体型ステレオ⇒コンポーネント・ステレオ

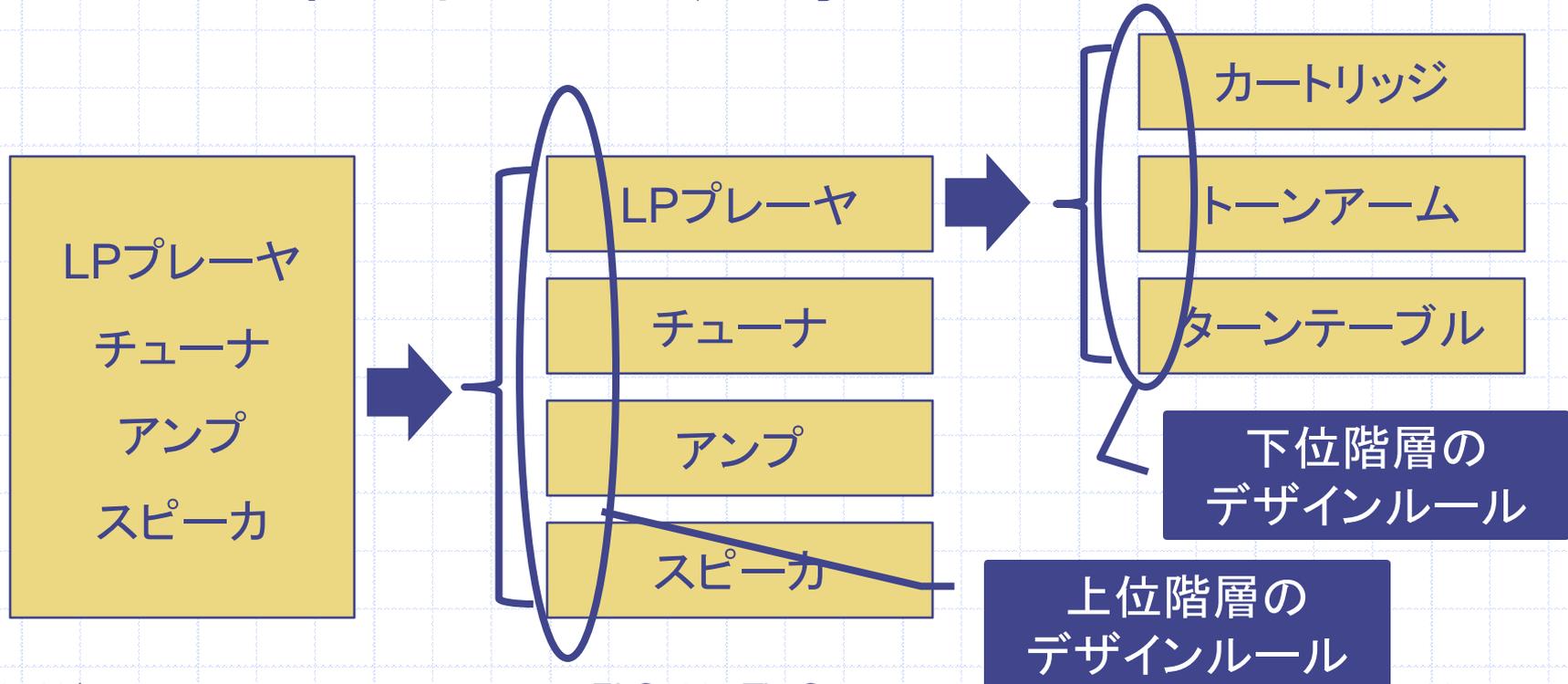


コンポーネント間の
接続ルール＝
デザインルール

分離

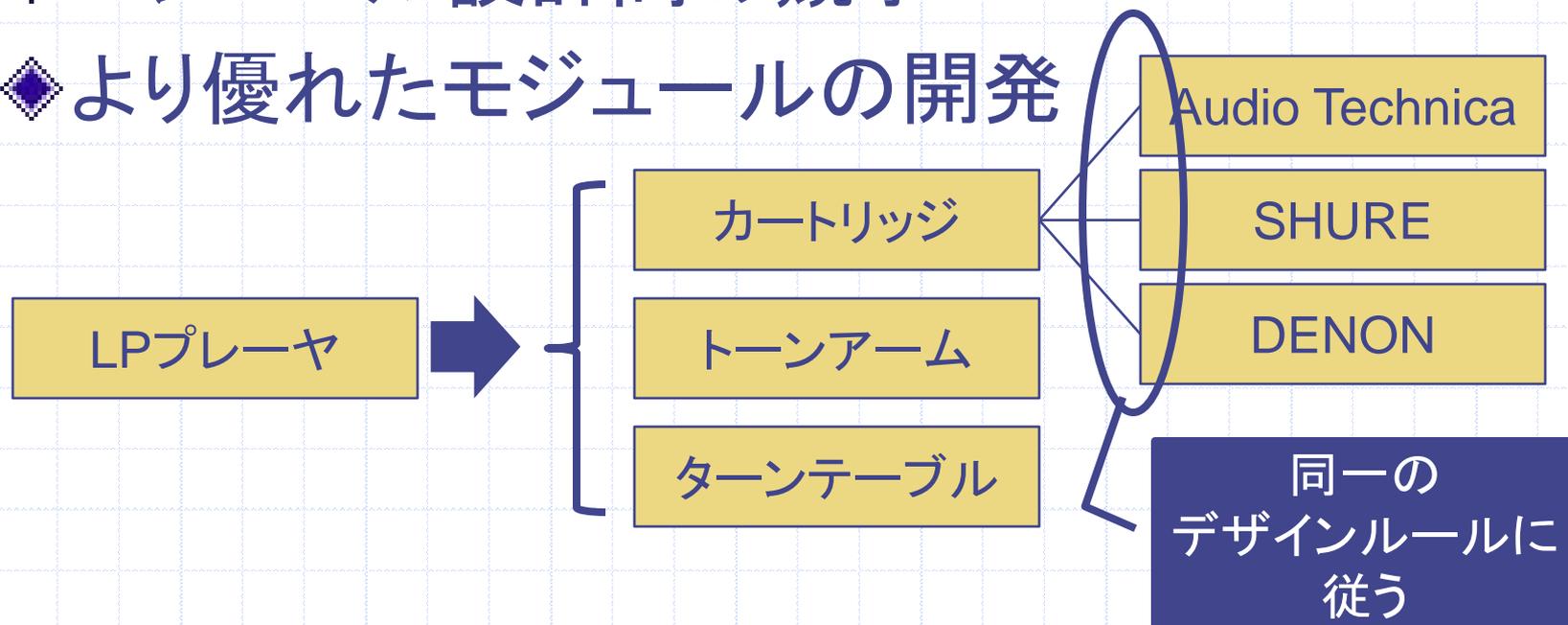
◆ 階層的に行われる場合がある

■ コンポーネント・ステレオ



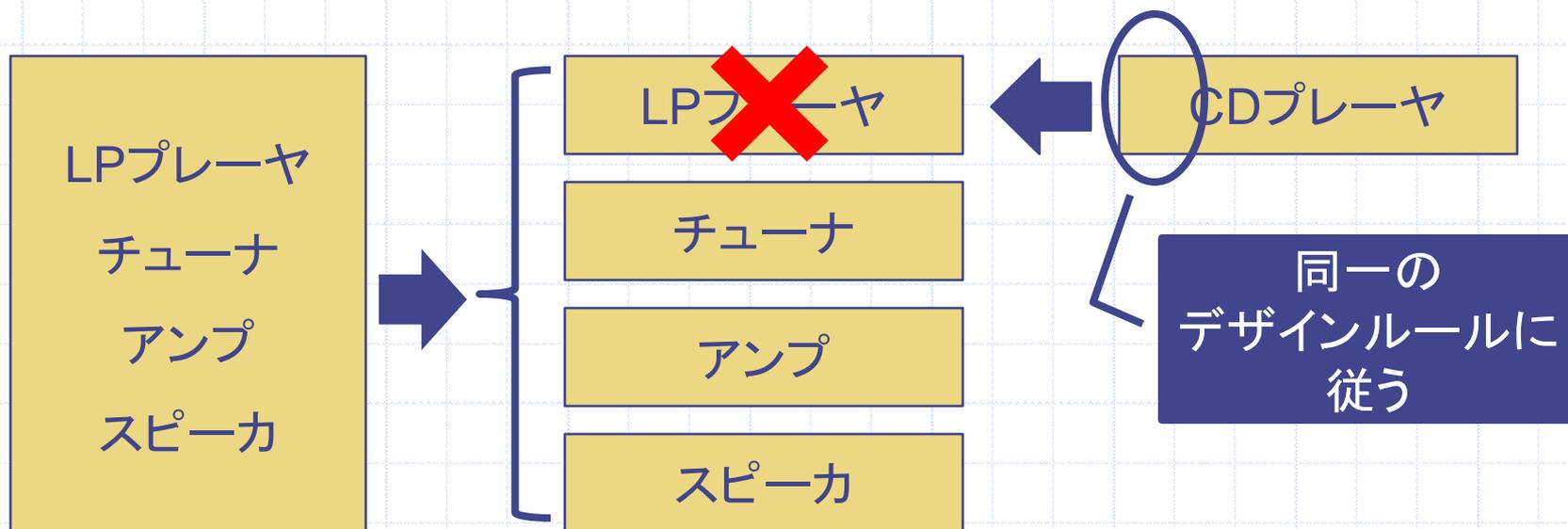
交換

- ◆ 「分離」されたモジュールの「交換」
- ◆ モジュール設計間の競争
- ◆ より優れたモジュールの開発



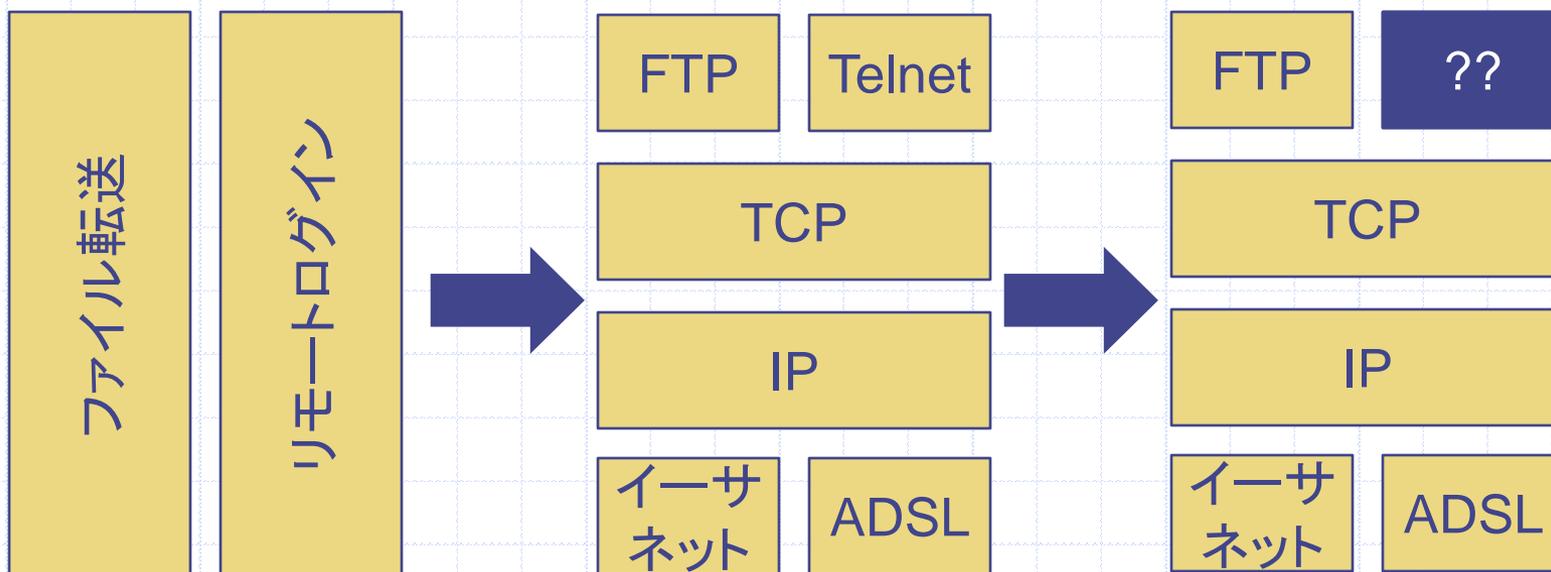
追加・削除

- ◆ ユーザによる好み・ニーズに合った構成の選択



追加の例??

◆ インターネット



一体型アプリ

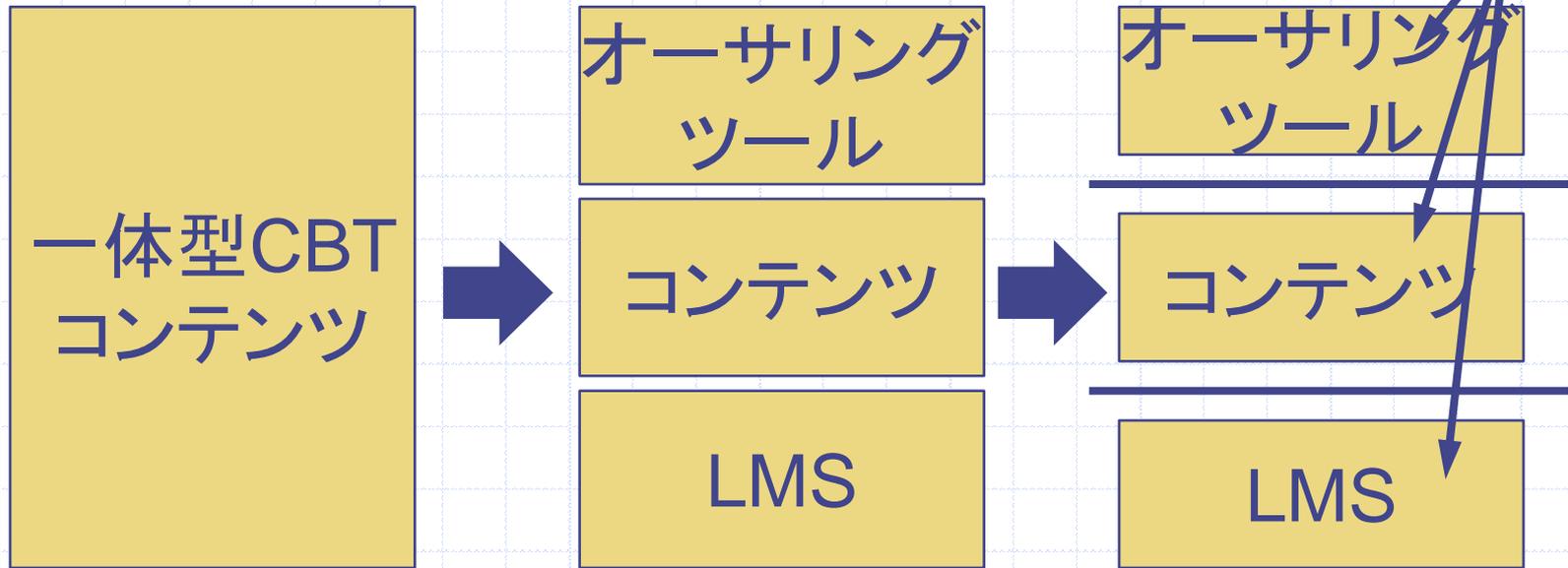
モジュール化

画期的機能の追加!!

追加

- ◆ 事前に全く想定しなかった画期的機能の
容易な追加＝予見不能!!
 - 新たなモジュールの追加のために、システム
本体の設計者との調整は不要
 - システム本体の設計者は、設計時に「追加」
の可能性を知らなくてよい

eラーニングシステム



一体型
~1990

分離(独自仕様)
~2000

標準化
2000~

現在のLMS(Moodleなど)の
プラグインアーキテクチャは
この段階

利用LMS (2011年のSCORM利用状況調査)

◆幅広いLMSが活用されている

(n=45, 複数回答)	人数
市販LMS(自社開発・国内)	10
市販LMS(他社開発・国内)	20
市販LMS(自社開発・海外)	5
市販LMS(他社開発・海外)	12
オープンソースLMS(国内)	12
オープンソースLMS(海外)	17
自主開発LMS(自社内でのみ利用)	9
ASP型LMS	12
LMSは利用していない	1

利用LMS種別数

- ◆ 複数種別利用が大半
- ◆ 10種以上という回答も
- ◆ 標準規格のメリットを活用

1	~3	~5	~9	10~	不明
17	17	5	2	2	2

利用オーサリングツール

- ◆幅広いツールが活用されている
- ◆独自ツールがかなり多い

(n=45, 複数回答)	人数
SCORM専用オーサリングツール(市販品)	21
SCORM専用オーサリングツール(フリーソフト)	8
LMS付属オーサリングツール	14
Flashなど汎用コンテンツ作成ツール	31
自社独自オーサリングツール (変換ツール・テンプレートを含む)	22
コンテンツは作成していない	2

交換の価値

- ◆ 分離されたモジュールをより良い設計に「交換」する
- ◆ k 個の設計から最良のものをを用いる
- ◆ k が増えれば良い結果が得られる確率が高まる

- ◆ j 個のモジュールの複雑さが同一，各モジュールの設計回数 k が同一とすると，価値の向上は
 - $V(j,k) = \sigma(Nj)^{1/2} Q(k)$

分離・交換の価値

◆ 実際にはオーバーヘッドがある

- デザインルール設計コスト: C_j
- モジュール設計コスト: $C_k k$
- 設計の検証コスト: $T(j, k)$

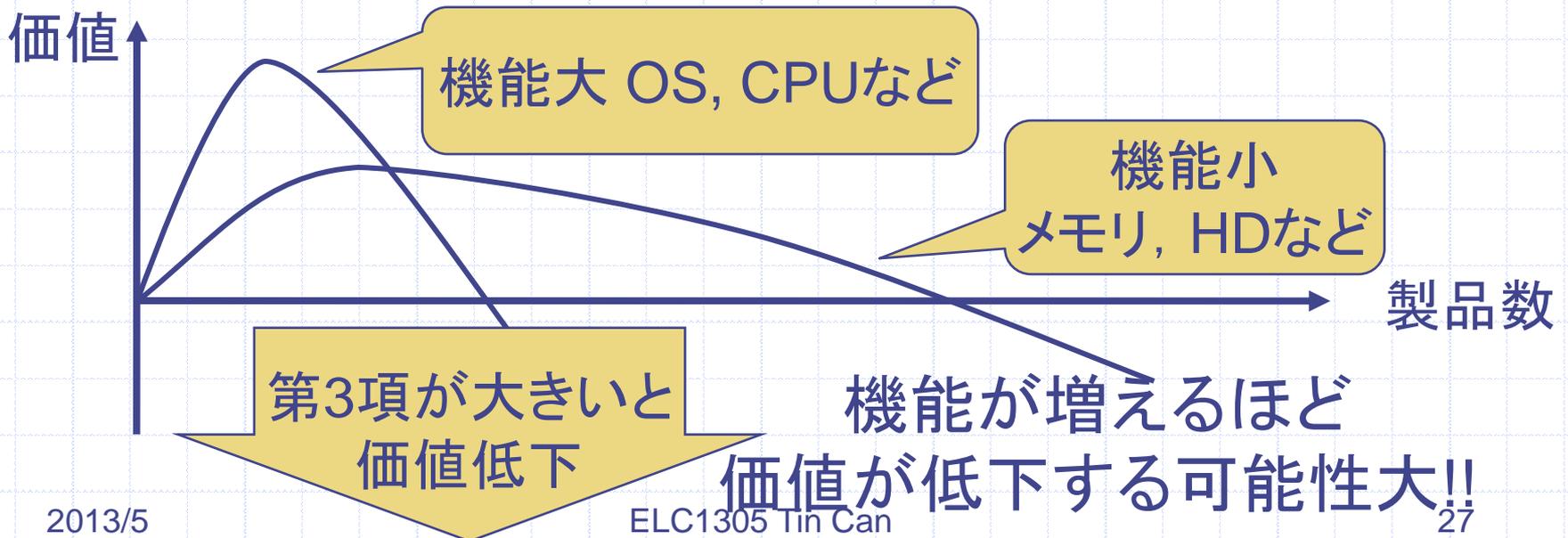
標準化の価値の定量化

◆ モジュールの価値 \propto

(機能・製品数)^{1/2}に比例する価値

— 機能・製品数に比例する開発コスト

— 機能変更が他に及ぼす影響



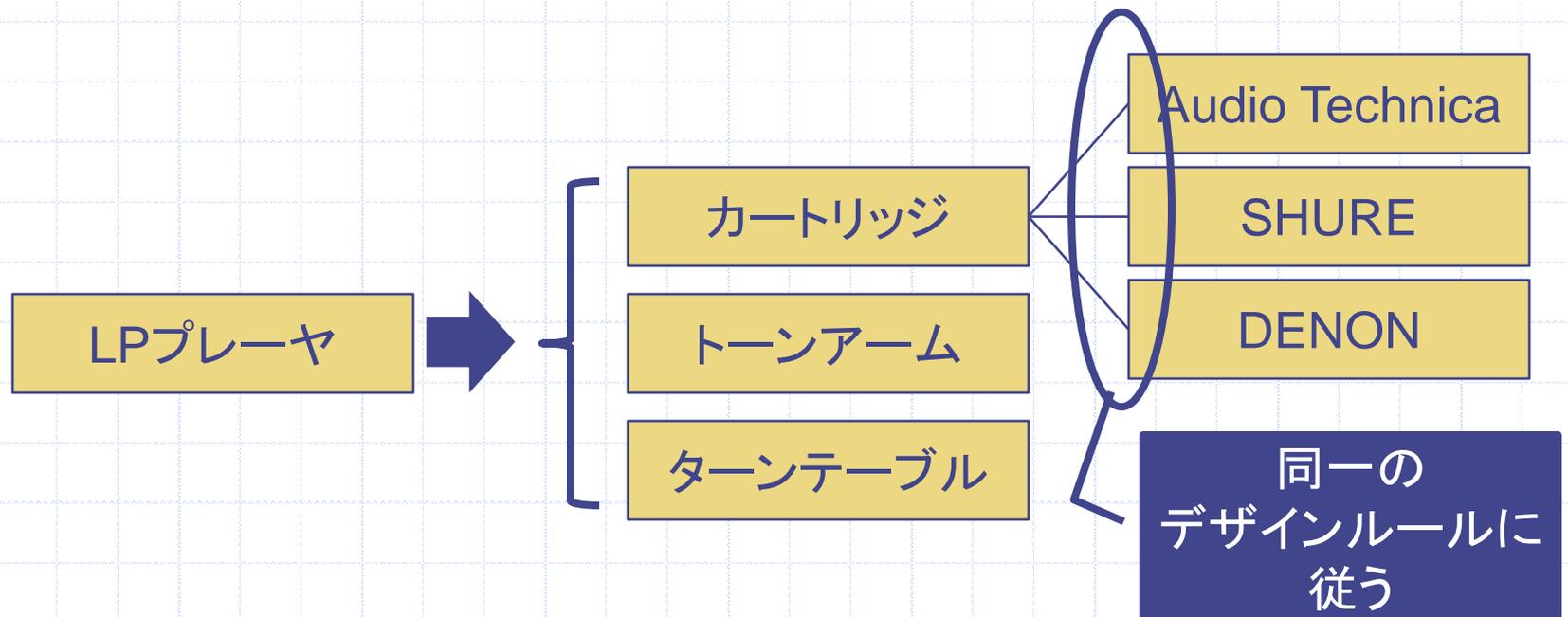
標準規格の価値

- ◆ 小規模・軽量な標準規格が価値を生む!!
 - 大規模で高機能な標準規格は価値向上をもたらしにくい

- ◆ 非常に逆説的・誤解する人多し
 - わかったふうな人から必ず出る反応
⇒「あれはオモチャだ」
 - SCORMがなぜそこそこ普及したか？

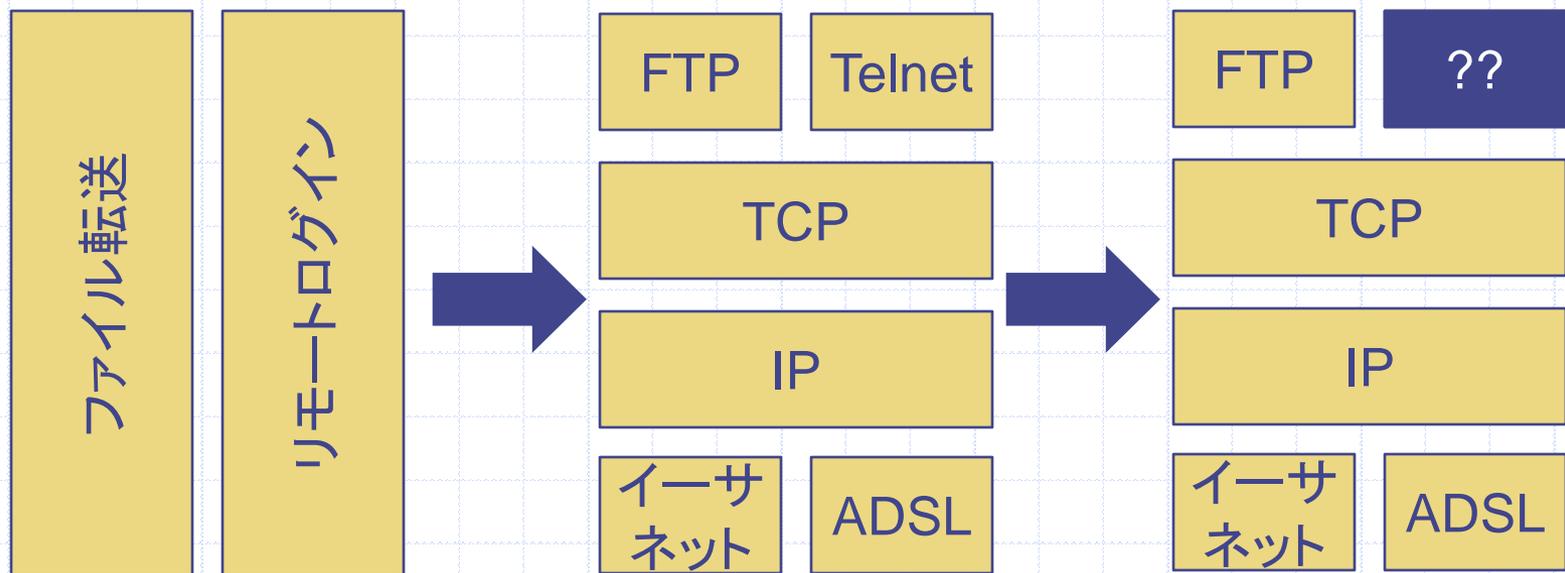
技術アーキテクチャと ビジネスアーキテクチャは相似する

- ◆ モジュール設計間の競争 =
モジュールに特化した専門企業間の競争



技術アーキテクチャと ビジネスアーキテクチャは相似する

◆ モジュールに特化した専門企業間の競争



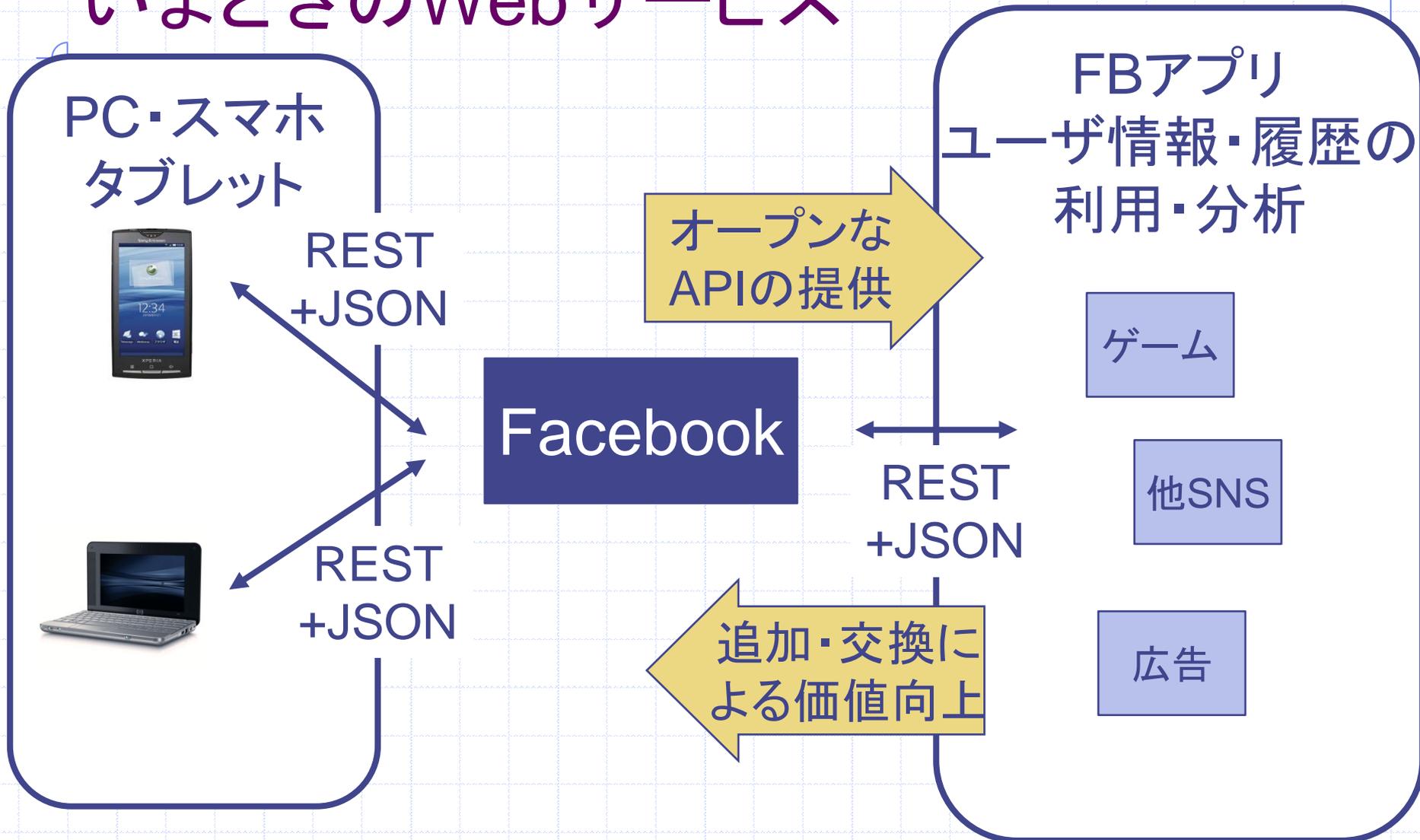
一体型アプリ

モジュール化

画期的機能の追加!!

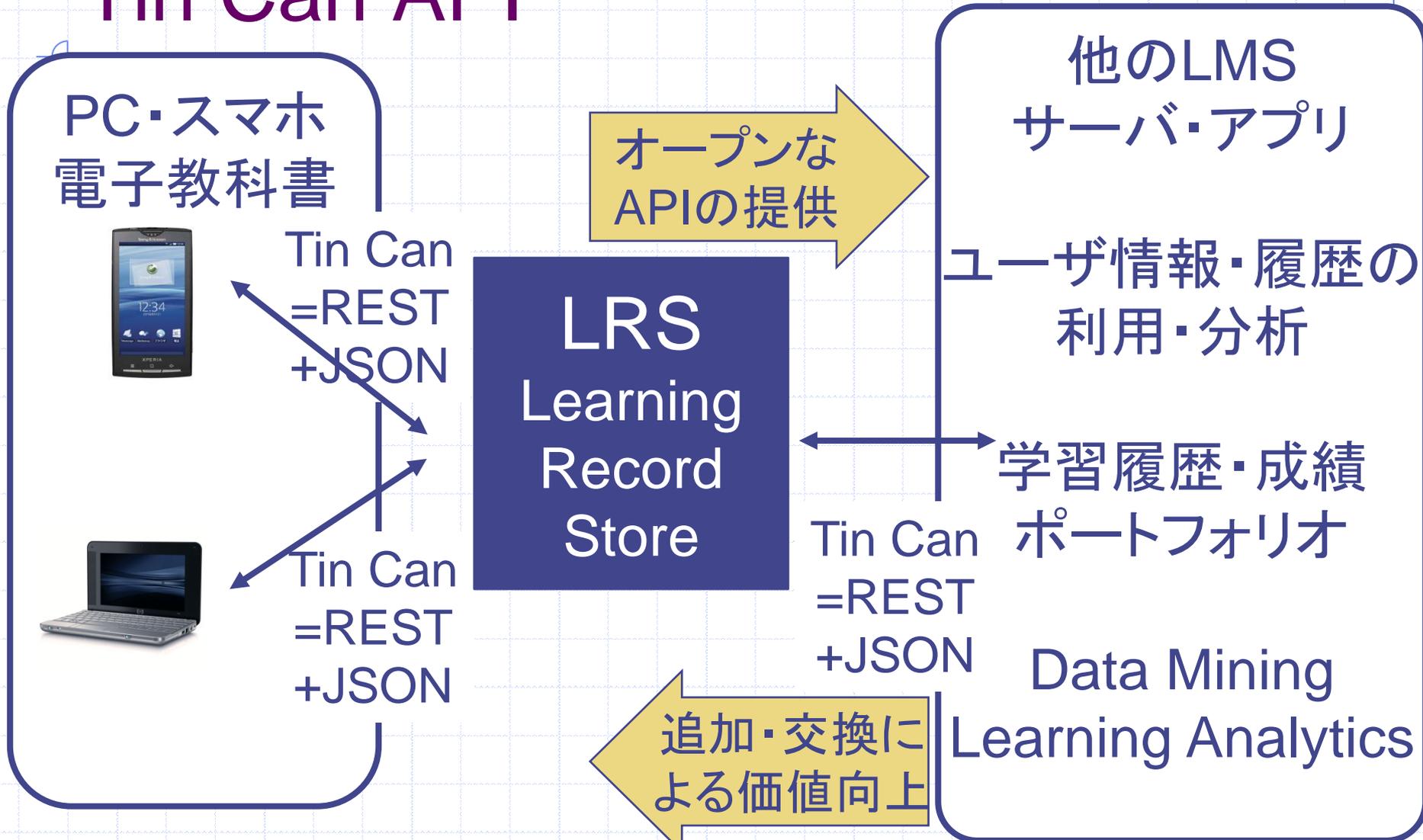
APIとは？

いまどきのWebサービス



APIとは？

Tin Can API



Tin Canへの今後の展開

◆ モバイル・電子教科書

◆ あらゆるコンテンツ

◆ シミュレーション・実習・教室

◆ 学習履歴データからの価値創造

- あらゆる学習履歴データの蓄積
- Learning Analytics

まとめ

◆ TinCanの位置づけ・意味

- eラーニング標準化の流れ
- eラーニングを取り巻く技術・ビジネス環境
- eラーニング技術やビジネスに対する新たな付加価値