

エンタープライズのプラットフォームとして 台頭するSaaSとWeb2.0のビジネスモデル

2007年5月14日

メタデータ(株) 野村 直之
リクルート(株) 八木 一平

アウトライン

- **背景:** 昨今の企業情報システムへの要求
- **1ビジョン:** 「オフィスの1個人のその日の仕事専用アプリが数分間で作成できること！」はもう実現?
- **SaaS:**
 - つくりがマッシュアップ and/or マッシュアップ可能なSaaS スイート
 - SaaSへの期待
 - SaaSはWeb2.0の本流の1つ ~ O'Reillyの7原則に照らして検証
 - ASP vs SaaS ~ 良いSaaS、悪いSaaS
 - 巧いSaaS運営のために
 - サービスの本質の研究
 - ITIL: サービスデリバリの応用
- **Web2.0のビジネスモデル**
 - エンタープライズ・マッシュアップのビジネスモデル (Top Down)
 - Webサービス(WebAPI) 提供者からみたビジネスモデルのバリエーション

背景:企業情報システムへの要求 ~エンタープライズが何故Web 2.0を必要とするか?

- Why “Web 2.0 for Enterprise”?

早い、安い、旨い

- 早い:超短納期
- 安い:チープ革命 e.g. はてなマップ~~5億~~ 13万円
- 旨い:高機能
 - 広範で深くホワイトカラー業務をサポート
 - 「1ビジョン」へ

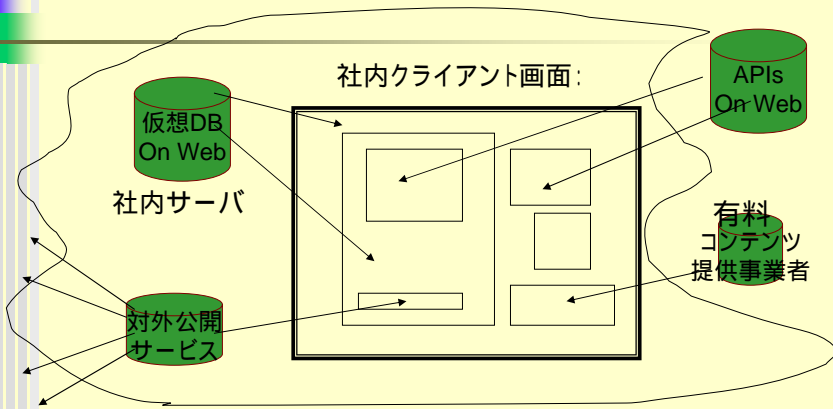
早い = 超短納期

- “チープ革命”の恩恵を最大限に享受しつつ急激に変化する市場ニーズ、業務ニーズに敏速に追随
- ↓
- Web, XMLベースの技術を採用すべし
 - Webベース・アジャイル開発の4手法
 - (1)「エンタープライズ・マッシュアップ」(クライアント)
 - (2)「XMLDBのスキーマ拡充やミドルウェア」(サーバ)
 - (3)「軽量言語 (Ruby on Rails等)の活用」(小規模サーバ)
 - (4)「“軽量”Webサービスやmicroformatsで連携」(必要な部分だけ作る)

1ビジョン:「オフィスの1個人のその日の仕事専用アプリが数分間で作成できること」



XML Consortium



基幹データ(含む個人mbox)を吸い出され過ぎないように注意
課題:セキュリティ、著作権管理&課金(DRM)

チープ革命が実現可能とする「機能のロングテール」

5

エンタープライズ基盤のビジネス化: 代表例: IBM × Google によるエンドユーザ・ マッシュアップ環境の提供



XML Consortium

Demo: Google Desktop x Gadget
「先の1ビジョンがもう実現した??」

行く付く先はWebOS??

コントロール・パネル機能までポータブルに
することでTCO削減にはなりそう。

6

つくりがマッシュアップ and/or マッシュアップ可能なSaaS スイート

「Zimbra Collaboration Suite」：

- ・国内向けプレスリリース（最近の注目記事）：
- ・Ajaxをデスクトップアプリ並に高めた、との定評有り：
<http://www.itmedia.co.jp/enterprise/articles/0511/30/news130.html>
http://www.ariel-networks.com/blogs/tokuriki/2006/03/ajax_zimbra.html
- ・グループウェアに「Web2.0」
<http://itpro.nikkeibp.co.jp/article/COLUMN/20070222/262613/>

「Zimbra Collaboration Suite (ZCS)」は、Webメール、スケジューラ、アドレス帳、文書作成といった機能を備えるグループウェア。開発元の米ジンブラが開示しているAPIを使って...

「すべてがデスクトップからブラウザに移動する」

<http://itpro.nikkeibp.co.jp/article/COLUMN/20070215/262094/>

...AjaxなどのWeb2.0技術を全面的に取り入れたグループウェア「Zimbra Collaboration Suite」を開発する、米ジンブラのサティッシュ・ガラマージ創設者兼CEOに、Web2.0の本質とそれが企業にもたらすものなどについて...

鍛冶克彦氏 於SaaS World 2007 (商務情報政策局、情報処理振興課長)

「ITフロンティア・イニシアティブ」by 甘利大臣
 日本企業にIT経営を定着させる (守りから攻めのIT投資へ)
 中小企業99%のIT化:特にサービス業、非製造業
 今まで以上に電子商取引、ICタグなどを活発化(製品安全、環境...)
 情報流通をメタデータで促進するのはきわめて重要
 IT産業、ユーザ含めての国際競争力の強化
 組込、金融分野を重点に
 1つ1つカスタマイズに金かけすぎて孤立していた

モジュール化して国際戦略商品つくろう
 日本のユーザ企業が競争力をそぐのに荷担していた

鍛冶克彦氏の発言(つづき) 於SaaS World 2007

- ユーザの利活用を考えると、SaaSは中小に向いていて素晴らしい。
- 国内の優秀なIT人材がベンチャーを立ち上げるが、なかなか巧く売れない。
- 共通のサービスpiで立ち上げれば広く売れるようになるのではないか。
- 企業、ユーザの皆様のビジネススピードにSaaSは向いているのでは？
- 政策的にお手伝いできるかはわかりません。
- でも、どのようにして相互運用性を確保するか(API公開とか)、スケーラビリティ、信頼性、セキュリティ、
- 技術開発の要素もあるし、それ以外の要素もあるだろう。

SaaSはWeb2.0の本流の1つ

~ O'Reillyの7原則に照らして検証

"<http://japan.cnet.com/column/web20/story/0,2000054679,20090039,00.htm>

1. プラットフォームとしてのWeb
2. 集合知の利用 e.g. Idea Exchange
3. いつでもどこでもデータは極めて重要
4. ソフトウェア・リリースサイクルの終焉
5. 軽量なプログラミングモデル
6. 単一デバイスの枠を超えたソフトウェア
7. リッチなユーザー経験

日経IT Pro “SaaS & Enterprise2.0” コラム

SaaSとWeb 2.0

XML Consortium

1. プラットフォームとしてのウェブ

SaaS

Netscape 対 Google : ソフトウェアの価値はそれが管理するデータの規模とダイナミズムに比例

DoubleClick 対 Overture / AdSense : Web 2.0の教訓 : ユーザのセルフサービスとアルゴリズムによるデータ管理を導入。中心部だけでなく周辺部、頭だけでなく長い尾の先にもサービスを提供。

Akamai 対 BitTorrent : 利用者が増えればサービスは自然に改善

2. 集合知の利用 教訓 : Web 2.0時代には、ユーザーの貢献がもたらすネットワーク効果が市場優位を獲得する鍵となる。

Salesforce.com
Idea Exchange

3. データは次世代の「インテル・インサイド」

Metadata Inc.

“オープンデータ” (/知識マスタ) の戦略も強力 メタデータ社のSaaS戦略

4. ソフトウェア・リリースサイクルの終焉

SaaS

オペレーションそのものがコアコンピタンス
ユーザーを共同開発者として扱う

O'ReillyのWeb 2.0原則との対比 続き

XML Consortium

5. 軽量なプログラミングモデル

SaaS

システムをゆるやかに統合可能に
調整 (coordination) ではなく、連携 (syndication)
ハッキング可能でリミックス可能なデザイン
組み合わせによる革新
再利用を徹底した軽量なビジネスモデルへ

6. 単一デバイスの枠を超えたソフトウェア

iTunesとTiVo : シームレスにインフラと一体化
データ管理、メタデータ管理をWeb/localで管理

SaaSケータイ
PDAフォン

7. リッチなユーザー経験

強いSaaS

Ajax, [RSS/Atom] Feed, ソーシャル (共有),
SNS/仮想体験空間

小川浩さんによる 最も
シンプルなWeb 2.0の定義

そもそもSaaSは“使いやすくなったWeb”の1種

O'Reilly:「Web 2.0企業のコアコンピタンス」:

1. パッケージソフトウェアではなく、費用効率が高く、拡張性のあるサービスを提供 SaaSをやるだけでもWeb 2.0的な企業に近づく
2. 独自性があり、同じものを作ることが難しいデータソースをコントロール。
このデータソースは利用が増えるほど、充実していく。
3. ユーザーを信頼し、共同開発者として扱う。
4. 集合知を利用する。
5. カスタマーセルフサービスを通して、ロングテールを取り込む。
6. 単一デバイスの枠を超えたソフトウェアを提供。
7. 軽量のユーザーインターフェース、軽量の開発モデル、そして軽量のビジネスモデルを採用。

-ASP vs SaaS ~ 良いSaaS、悪いSaaS

ITR 内山悟志氏 (IT Research代表;
CIOマガジン責任編集) の論点引用(抄)

北米企業に見るSaaSモデルの浸透状況

2005.11 Forrester Res. 583社 → 日本の現状?

CRM, Security, ERP, backup, messaging, SCM, etc.

28% 23% 22% 20% 18% 17% 39%

運用管理系のソフトがSaaSの対象になっている

・ITR, CIO magazine IT投資動向調査2007 調査月:2006.09
実施状況:未検討50.3% 実施済23% 検討後見送り11%
重要度:高11% 中37% 52% 低い

従来型ASPの失速原因

- ネットやDCへの過剰投資 : アプリに資金確保できず
 - 熱心だったのは製品ベンダでなくアウトソーシングベンダ
 - 資源やデータの外部以降への抵抗感を過小評価
- パッケージ製品比の低競争力: 業務アプリはコモディティ化していなかった
 - サービスとして提供される内容・範囲が偏ってしまった
 - 技術的にカスタマイズと連携機能に制約あり拡張性が犠牲に
- ターゲット市場(中堅・中小)の選択ミス: マスマーケット対象ビジネスの経験がなく顧客ベース獲得に失敗
 - 規模の経済性に働かず、ユーザへの低価格サービス提供に失敗

料金モデルが魅力的でなかった
「でも、しか」ASP、は駄目

- 給与計算は10社に売れたら10通りの考え方に基づいてカスタマイズが必要
 - 運用するとどんどん不利益: 運用の手間がかかりすぎ
- 成功しているモデルは、「顧客が満足し得る範囲内でのセルフ・カスタマイズが Wizardなどのできるようになっていた」!
- ネット前提のアプリ設計: マルチテナント;
 - 対象領域 アプリ、ツール、ミドルウェアなど
 - メタデータの活用がカスタマイズ性向上の鍵
 - 連携: API準備のコストが低いこと

「ライセンスへの不満」

- 体系が複雑でわかりにくい
- 外資のは交渉余地ない
- 変更を強制される
- 既存体系は利用度から納得感がない
- 技術と価値のバランスが納得できない
- 保守料が高く料率に対して納得できる説明がない
- 高額な保守料に見合ったサポートを受けていない
 - バグ直しに料金支払い？
 - R&D投資に付き合わされている？

SaaSモデルのメリットとデメリット

Pros:

- 短期導入可能
- 初期費用の負担軽減
- 小規模開発・保守での対応作業やコスト低減 (install, ver.up)
- 稼働pfを問わない

Cons:

- 希望するカスタマイズのすべてが可能ではない
- アプリの所有権を放棄
- SLAを厳密に定義しなければならない

SLAの例: 可用性の桁が違えば サービス料金の桁も違うかも

明治政大
H25E

ISO/IEC17799 の可用性ガイドライン

XML Consortium

可用性 %	非稼働の"分"/年	グレード
90	52560	管理不十分
99	5256	一応管理
99.9	526	良い管理
99.99	53	耐エラー性十分 (Fault tolerant)
99.999	5	高稼働
99.9999	0.5	超高稼働
99.99999	0.05	超々高稼働

19

双方の意識改革がまだまだ必要そう..

XML Consortium

- モノよりサービス; 所有から利用へ
 - 固定費も削減できる; ITILのマネジメント・モデルが使える
 - 脱・原価積み上げ方式; 御利益に応じたプライシング
- データは「あちら側」に置いた方が安くて安心 cf. 銀行預金
- 自前で作るのは1%。残りはWebAPI, SOAPから Wiki, Blog..
 - チープ革命で、高機能、高性能、高品質の部品が只でふんだんに使える
 - 可用性など、SLAに注意 (中には「只ほど高いモノはない」場合も)
- 要求仕様を固定したシステムの寿命は3日~3ヶ月
- アプリの御利益を評価・測定し、活用する方が大事
 - 新ビジネスモデルの実装はもちろん発想に結びつくIT
 - アジャイル開発で大事な「要求開発」とその道具 (e.g. RoR)
- セキュリティはバランスと、対策のスピード
 - 最新の対策にいち早く追従するプラットフォームが優秀

20

April Fool, 2007.04.01

「SaaSは日本古来の概念であることが判明」<http://d.hatena.ne.jp/nomuran7/20070331>

- ハードウェアで売り上げ、ソフトウェアは「サービスしてしまおう」古き良き時代をご記憶であろうか。
- 某長老顧問がSaaSについてコンサルタントの説明を受けた後インタビューに答えて曰く：
- **「サービスとしてのソフトウェア」、実に懐かしい、素晴らしいものではないか。そう、ソフトウェアのような形の無い、触れない、目方の無いあやしげなものなんかにお金を払ってはいけないのだ。昨年あたりから若い連中もようやくそれに気づいて、「ソフトウェアはサービスです」と言うようになったとな。結構、結構。**

21

サービス[商品]の基本特性

- その場で(最終)生産され、消費される
(リアルタイム性)
 - 生産行為と消費行為を分離できない
 - “触れない” “目に見えない” ことが多い
適正価値(価格)の評価(市場で)が困難
 - 運搬できない
 - cf. 羽田 福岡の輸送サービスは? 「羽田 福岡に顧客1人を運ぶ」という商品をドイツや中国にもっていくことはできない
 - ストック(在庫)が効かない
 - 原則、4W3H (5W – why + “how long” ”how much”) を自在に取り替えられない
 - Business Process自体が商品
 - 業務知識管理 = 企業ナレッジマネジメント
 - カスタマイズ性に富む cf. commodity性が薄くなる

22

サービス・プライシング

XML Consortium

- ソフトウェアの買い取りから利用量[時間/期間] 等に基づく従量制へ
 - 事例: ASP化した歯科レセコン by 日立ソフト
 - 19800円 / 月・クライアント
 - 医療事務担当者1名が処理できるデータ量はたかが知れているので定額制で十分
 - 「食べ放題」のレストランの方が利益率高いのと同様
- サービスの評価が、期待値との差分でなされるため (ServQual法)、原価の積み上げによるプライシングはナンセンス。反発くらいやすい。
 - ソフトウェアも、原価=人件費ではなく効果、ご利益でプライシングされるべき

23

その他、サービスの基本特性

～ サービス生産のためのITとビジネスモデル考察のために

XML Consortium



<http://itpro.nikkeibp.co.jp/article/Watcher/20060714/243390/> “長続き”するWeb 2.0のビジネス・モデルを考える

法政大学大学院イノベーション・マネジメント研究科での
 拙講義資料: 1988年以降のサービス評価モデル等

<http://pc.nikkeibp.co.jp/article/COLUMN/20070219/262412/?ST=saas>

24

ITサービス継続性管理

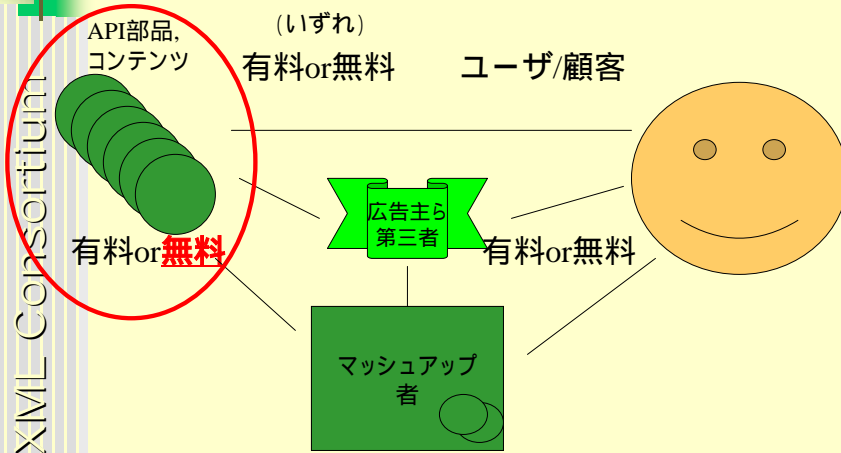
- 目的
 - インシデント、問題が与える影響度の最小化
 - インシデントの再発防止
 - リソース(人、スキル、金など)の効率利用の促進
- 責任
 - 問題コントロール(根本原因究明まで:RFCを作るまで)
 - エラーコントロール(RFCを実施し、根本原因を除去するまで)
 - 重大障害のバックアップ
 - 障害の事前予防
 - 管理情報作成元の提供
 - 重大障害の原因レビュー
- 特記
 - 原則Proactive(& Reactive)

可用性管理

- 費用対効果が高く持続的な可用性レベルの提供
- 可用性とは
 - 可用性(SLAの一部)
 - 信頼性、保守性(OLAの一部)
 - サービス性(UCの一部)
 - セキュリティ(全側面に必要)
- 責任
 - 可用性の監視、報告、最適化
 - 事業の視点で可用性の要件を決定
 - 可用性計画の作成
 - 継続的な可用性のレビューと改善
- 留意点
 - 可用性計画では技術のみならず、プロセス、ツール、技法にも焦点をあてる

Eマッシュアップのビジネスモデル

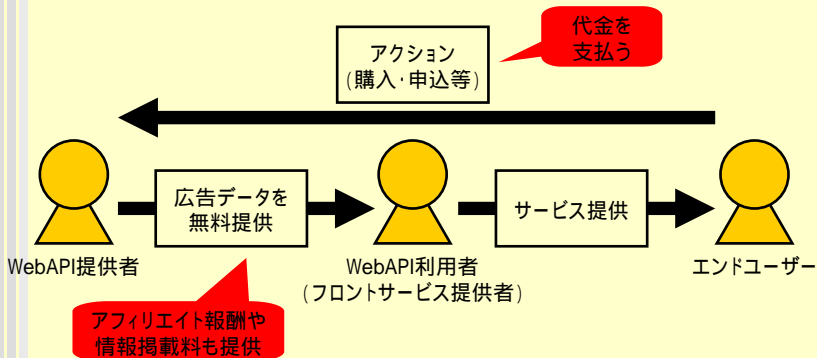
• どれが生き残るでしょうか？ 手順、イニシアチブ、等等
(例: 逆オークション) でバリエーション



Cf. Google: まずアプリの形で出して市場の反応みて決める

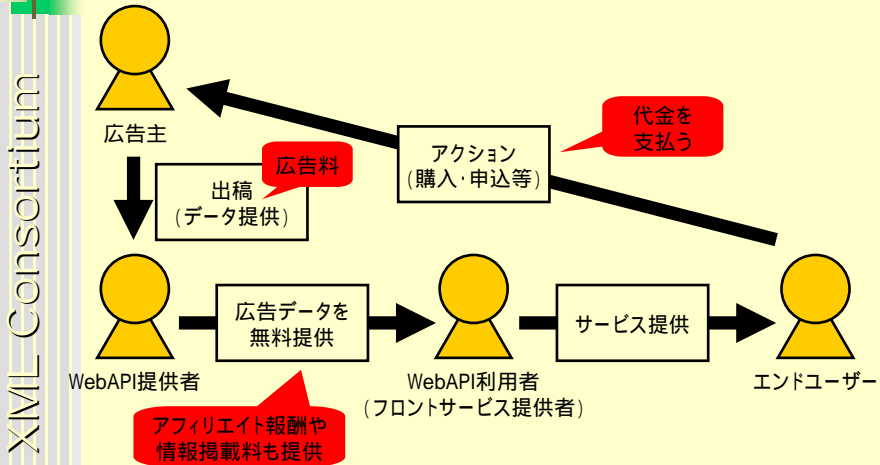
1 - 1

自社プロモーション型



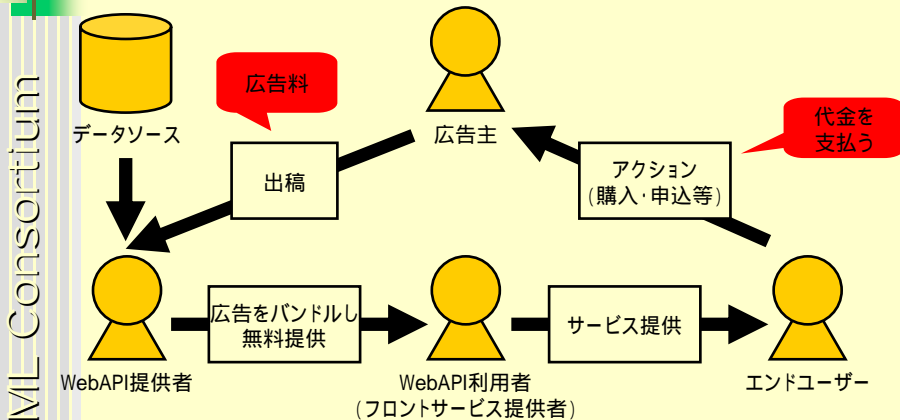
1 - 2

広告代理店プロモーション型

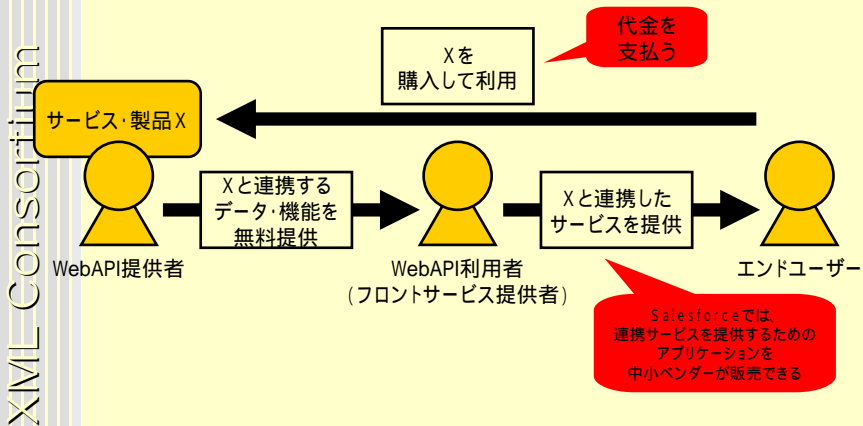


1 - 3

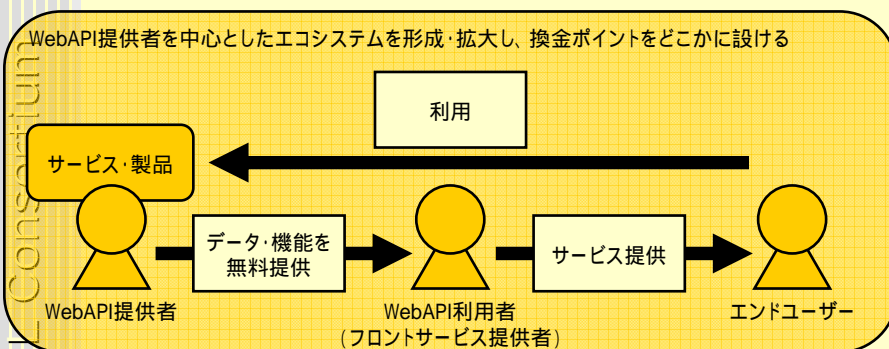
広告バンドル型



1 - 4 ロックイン型

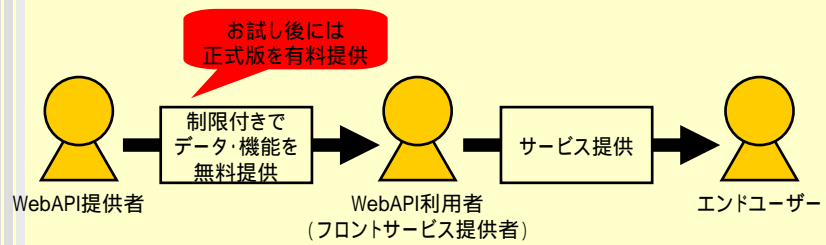


1 - 5 エコシステム強化型



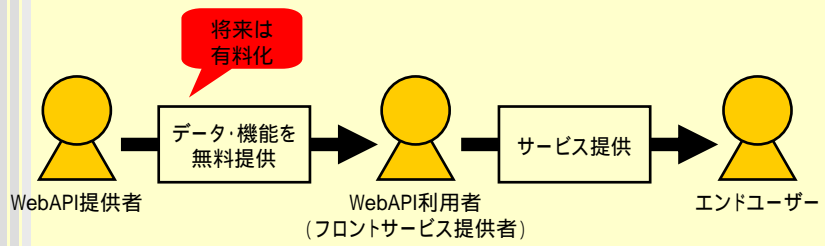
1 - 6 お試し提供型

XML Consortium

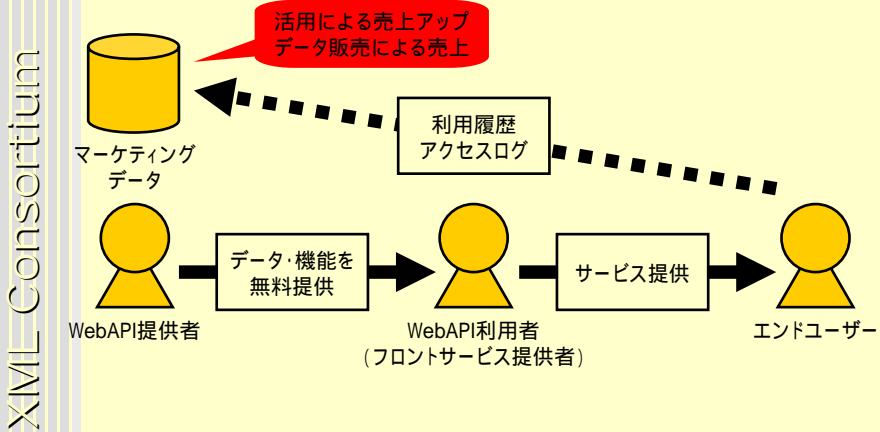


1 - 7 いずれは有料提供型

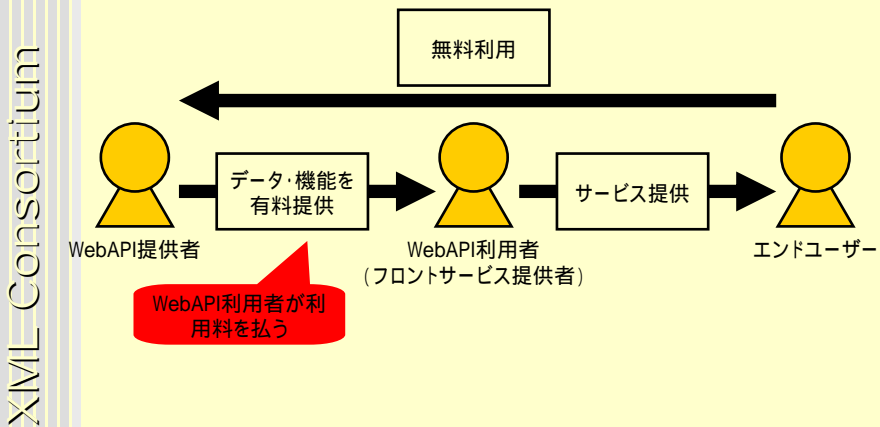
XML Consortium



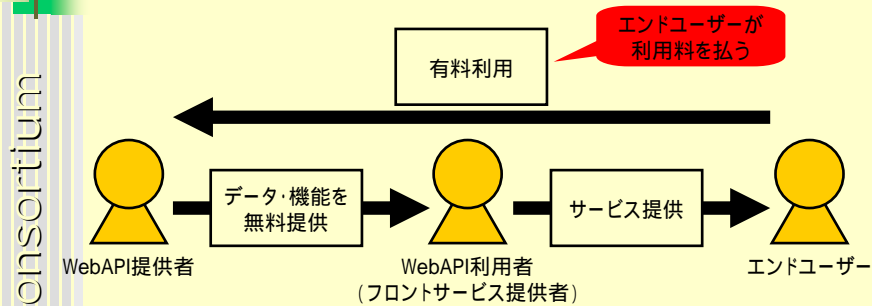
1 - 8 マーケティング型



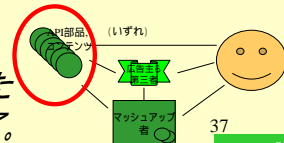
2 - 1 WebAPI利用者課金型



2 - 2 エンドユーザー課金型



以上で、WebAPI提供者から見た
マッシュアップの収益モデル、了。



そもそもビジネスモデルとは？ ~どのような事業活動をするかのモデル

1. 戦略モデル:

どんな顧客(市場)に何をどのように付加価値(魅力)付けて提供するか表現

2. 業務モデル(BPM):

戦略モデルを実現するための業務プロセス、具体的なオペレーション設計・構築手法とその結果としての構造表現モデル(BPM)

3. 収益モデル:

事業活動の利益を確保する仕組み(収入モデルとコストモデル)

結語: Web2.0 的なビジネスモデル, スタートアップ

- 課金モデル (このモデルの変更にユーザが参加!)
 - いろんな種類の従量制へ
 - でも「使い倒した方が得」という階段関数的な特徴は残す
 - シンプルだけど細粒度へ
 - 原価積み上げでなく御利益や相場で料金決定
- バリュー・チェーン、Stakeholder間フロー (3者以上が必須)
 - ユーザ参加 (CGM, 従来コスト高かった売り手や仲介役, サービス生産への参加)
 - ガチガチの業界規格でなくオープンで無料のREST API等
 - 実質的な連携は**メタデータ**により遂行
- コストモデル ~ 原価積み上げは終焉
 - 徹底した低コスト追求 & 同じ仕組みで高度なスケーラビリティの遺伝子を最初から組み込むこと e.g. youtube; 参入障壁を徹底的に低く: 「まず無料」は当然
 - 広く浅い大量のトランザクションを激安コストでさばく