

Androidの次の飛躍を考える

--- Webアプリ / HTML5開発の新しい動向

@maruyama097

丸山不二夫

Agenda

- Androidの現在
 - 基本的なトレンド
 - Webアプリ / HTML5開発の新しい動向
 - AndroidとWebアプリ / HTML5開発
 - Androidの次の飛躍を考える
-



日本Androidの会 「災害時支援アプリ・マッシュアップ・ミーティング」

- 第一回 2011年3月25日 at 東大駒場
- 第二回 2011年4月3日 at 楽天
- 第三回 2011年4月9日 at 東証コンピュータ
Payforwardingプロジェクトとして継続



2012年10月20日 東北大で、ICT ERA+ABC開催

ICT Earthquake Reconstruction Aid
powered by Android Bazaar and Conference 2012 Tohoku



2012年 10月20日 (土) 10:00～ 場所：東北大学荻ホール
主催：日本Androidの会 共催：東北大学IIS研究センター

[TOP](#)[メッセージ](#)[イベント概要](#)[カンファレンス](#)[バザール](#)[懇親会](#)[会場](#)[FAQ](#)[お問い合わせ](#)[被災地視察](#)

ICT ERA =
ICT **E**arthquake **R**econstruction **A**id

10月21日 被災地視察



2012年の動き

Samsung Galaxy S3 2012/5/3



Windows Phone 8

Nokia Lumia 2012/09/05



Kindle Fire HD 2012/09/06



iPhone 5 2012/09/13



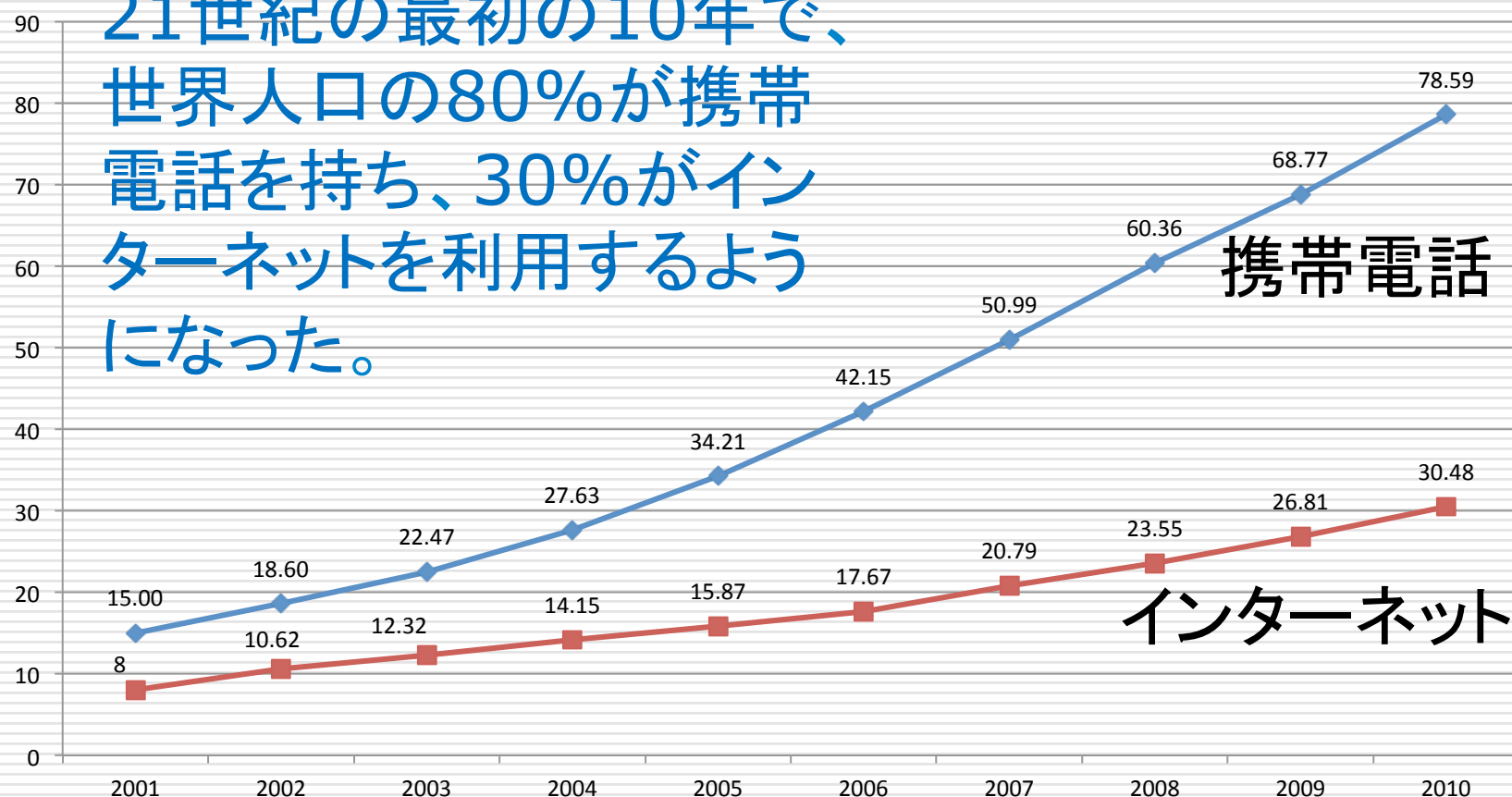
MS Surface 2012/10/26



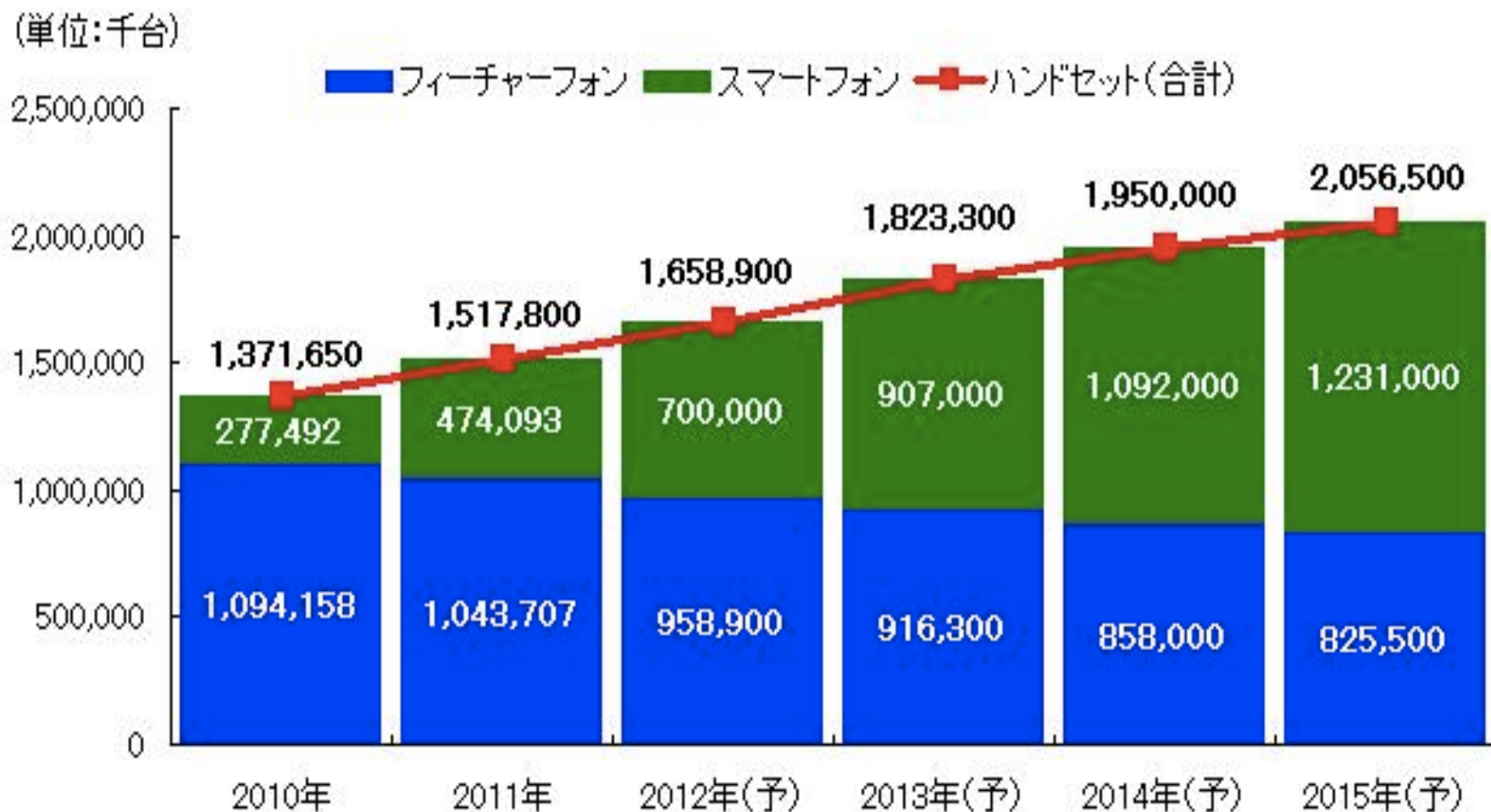
Androidの現在

21世紀、最初の10年 携帯電話の爆発的普及

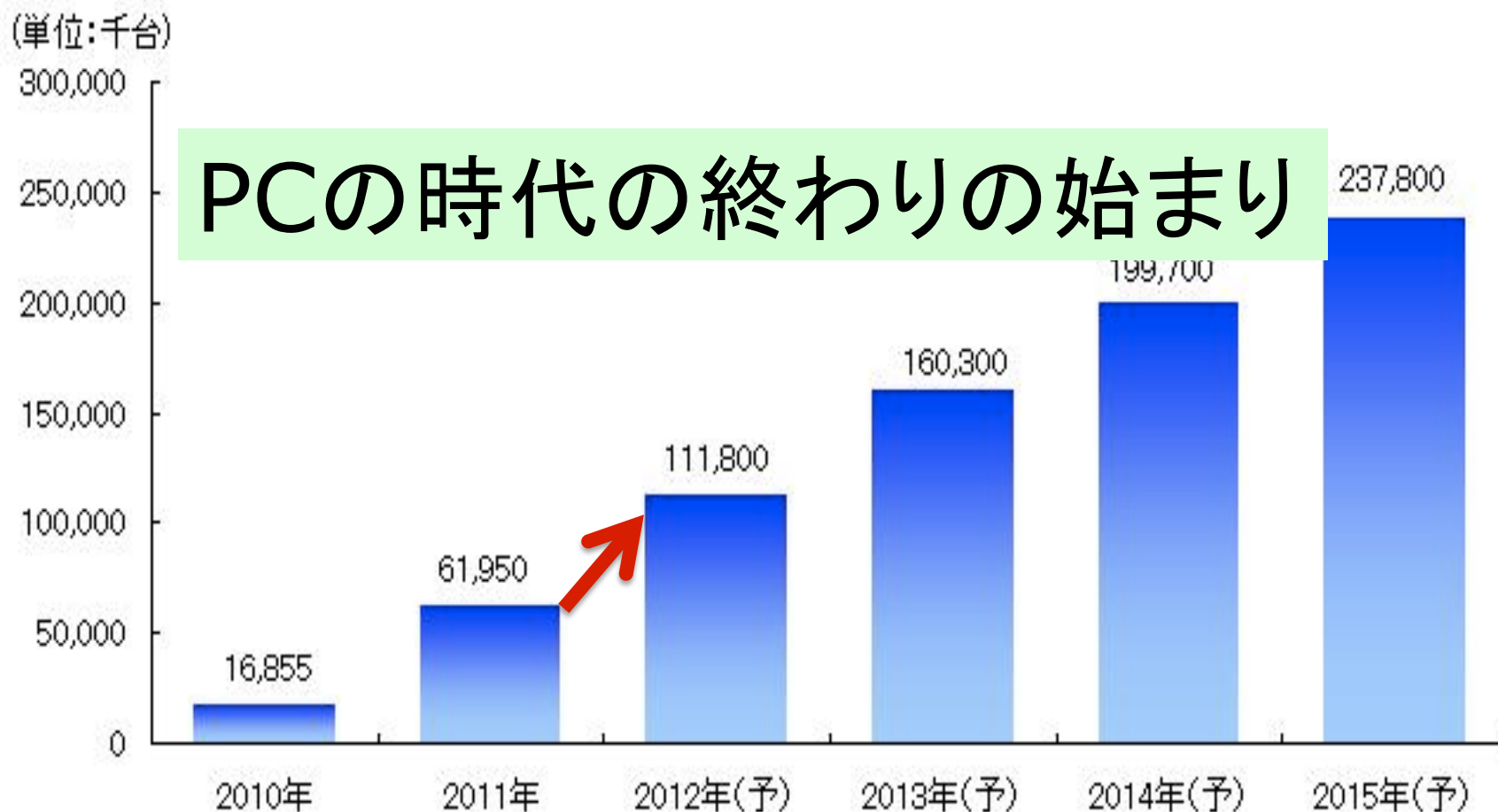
21世紀の最初の10年で、
世界人口の80%が携帯
電話を持ち、30%がイン
ターネットを利用するよう
になった。



2013-14年には、携帯電話の半分以上がスマートフォンに



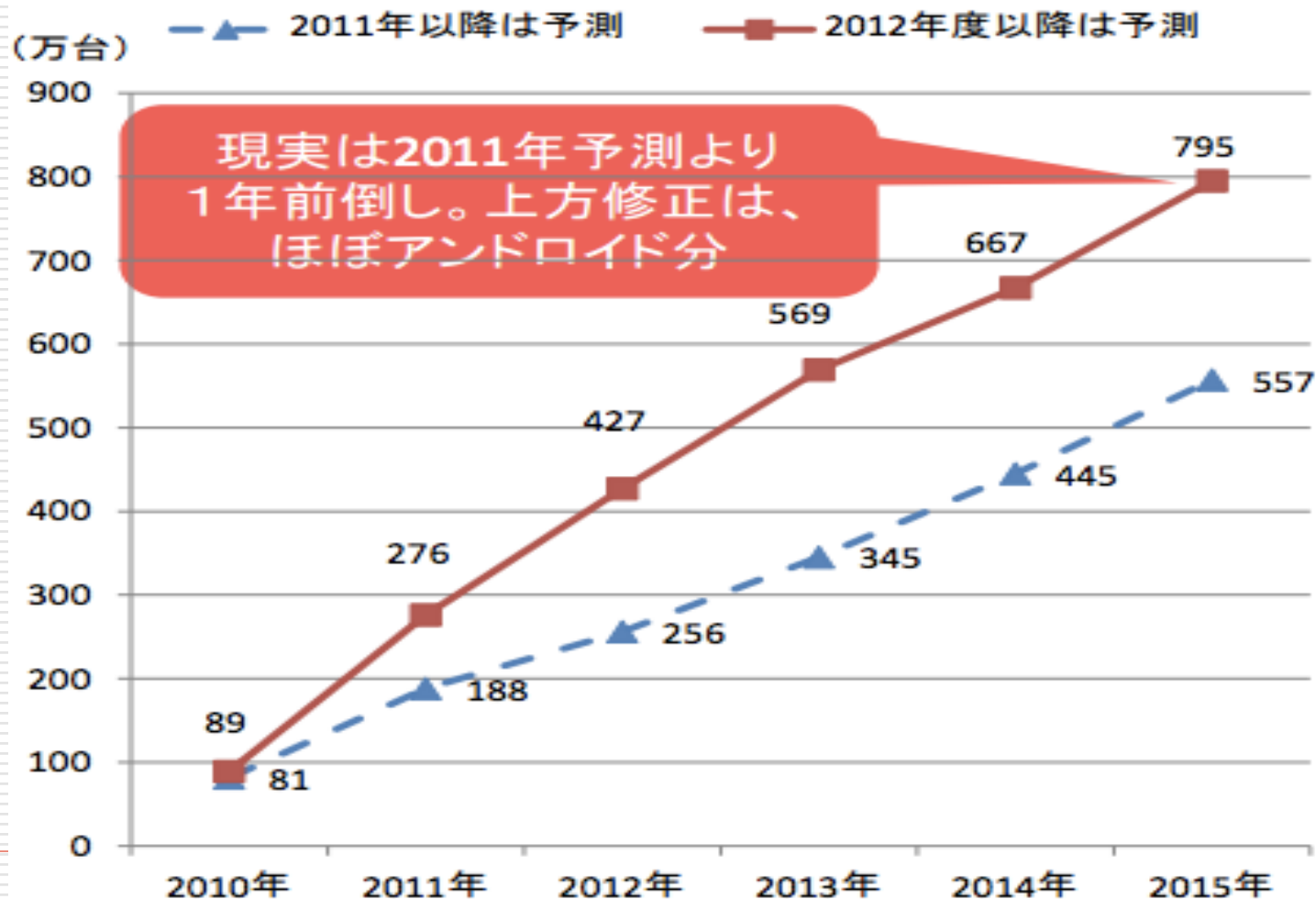
2012年 PCは、出荷台数 4.9%減 タブレットは、前年度比、約2倍



<http://www.yanoict.com/yzreport/253#a1>

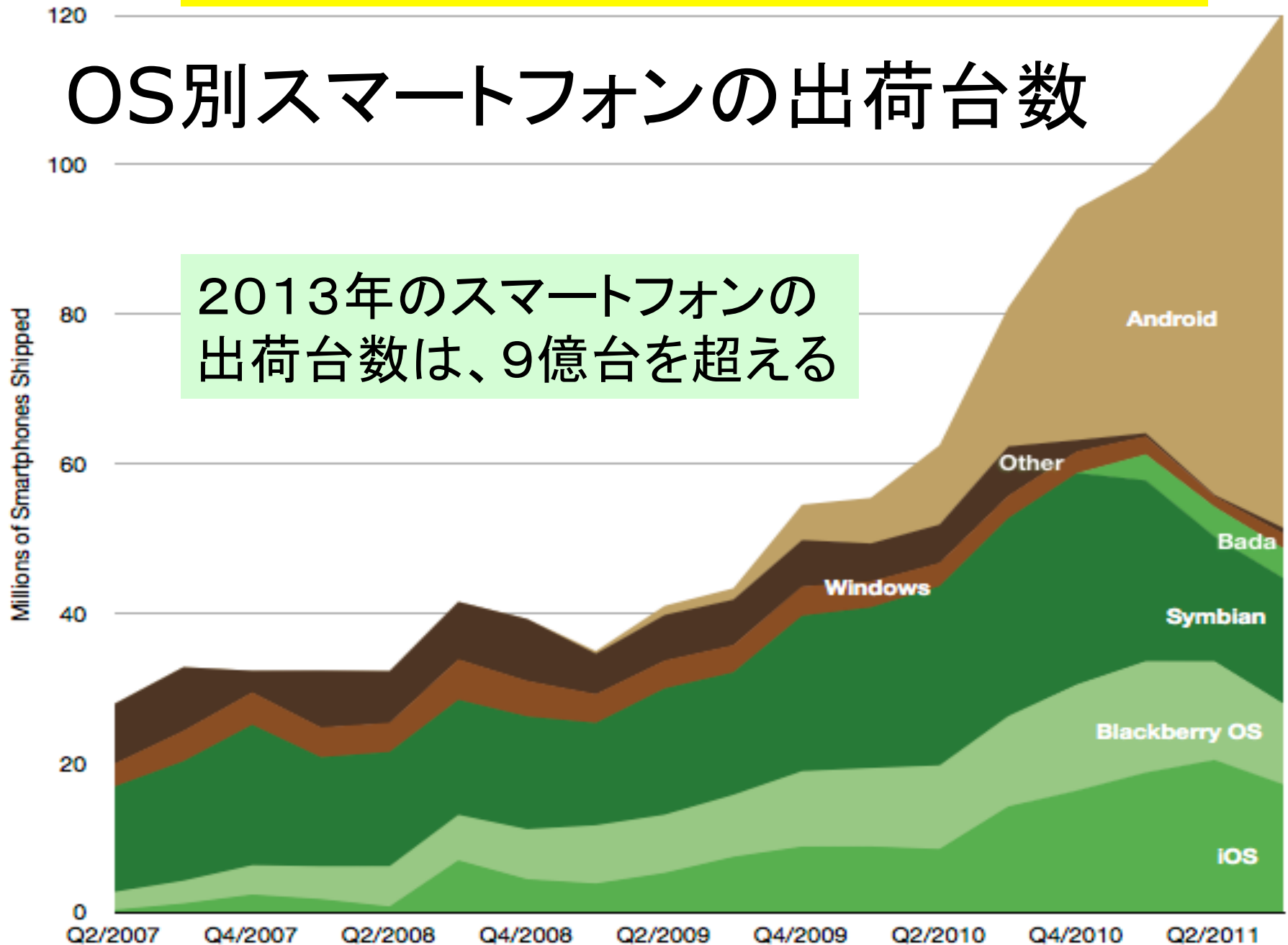
タブレット端末の国内出荷台数予想

タブレットでもAndroidの躍進



OS別スマートフォンの出荷台数

2013年のスマートフォンの
出荷台数は、9億台を超える



Androidのシェア 72.4%

iOS 13.9% の5.21倍

スマートフォン

Operating System	3Q12 Units	3Q12 Market Share (%)	3Q11 Units	3Q11 Market Share (%)
Android	122,480.0	72.4	60,490.4	52.5
iOS	23,550.3	13.9	17,295.3	15.0
Research In Motion	8,946.8	5.3	12,701.1	11.0
Bada	5,054.7	3.0	2,478.5	2.2
Symbian	4,404.9	2.6	19,500.1	16.9
Microsoft	4,058.2	2.4	1,701.9	1.5
Others	683.7	0.4	1,018.1	0.9
Total	169,178.6	100.0	115,185.4	100.0

Source: Gartner (November 2012)

Gartner 2012年11月の調査

Androidのシェア 68.3%

iOS 18.8% の3.63倍

スマートフォン

Top Smartphone Operating Systems, Forecast Market Share and CAGR, 2012-2016

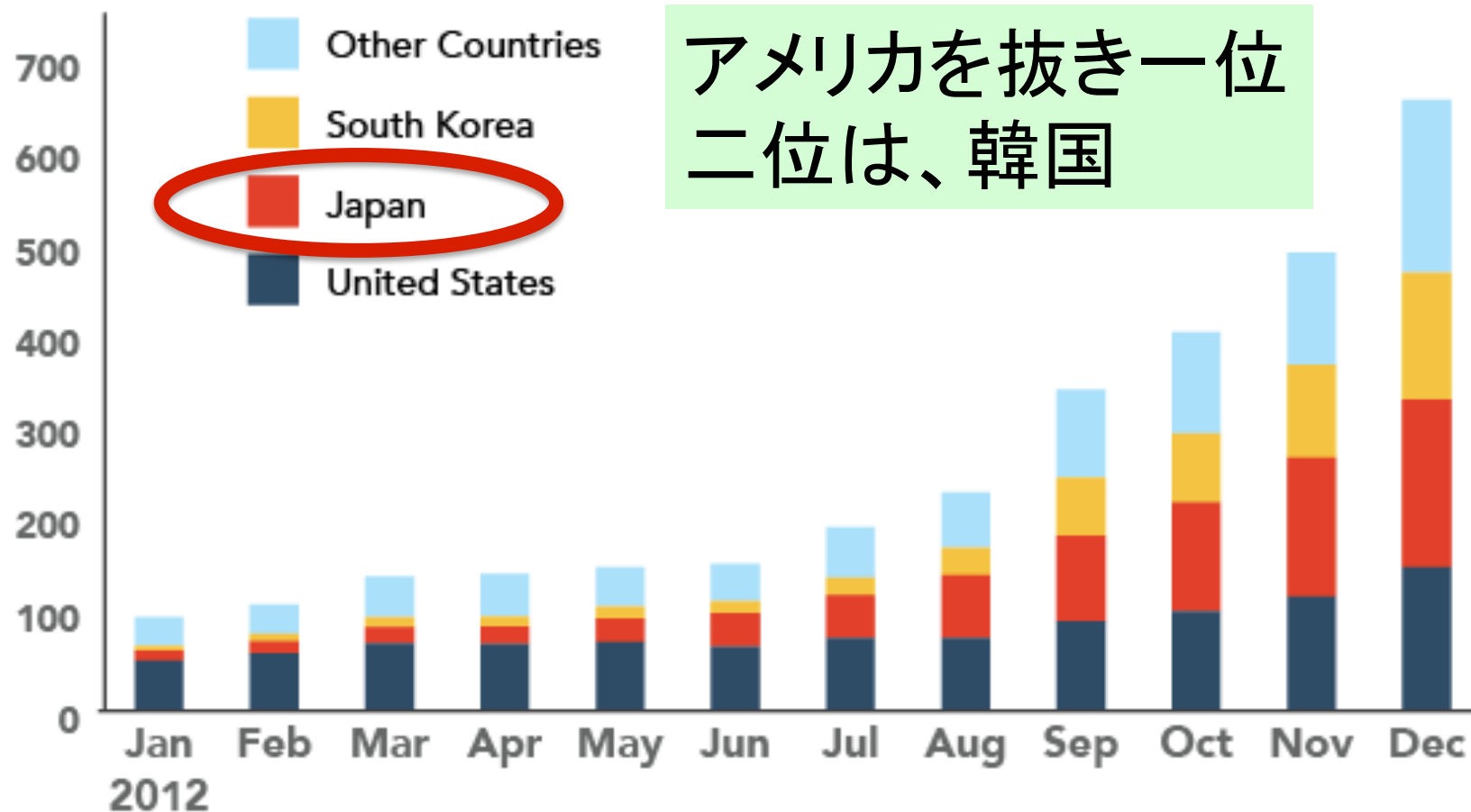
Smartphone OS	2012 Market Share	2016 Market Share	CAGR 2012 - 2016 (%)
Android	68.3%	63.8%	16.3%
iOS	18.8%	19.1%	18.8%
BlackBerry OS	4.7%	4.1%	14.6%
Windows Phone	2.6%	11.4%	71.3%
Linux	2.0%	1.5%	10.5%
Others	3.6%	0.1%	-100.0%
Total		%	18.3%

IDG 2012年12月の調査

Androidアプリ開発の可能性

- グローバルに目を向ければ、Androidアプリの潜在的なユーザーは、iOSアプリユーザーの3~5倍以上存在することを意味している。
- ハードの分野では、残念ながら、日本勢は、必ずしも強いとは言えない。日本Androidの会の会員の圧倒的多数は、アプリの開発者である。
- AppleのApp Storeとくらべると利益率が低いという重要な問題はあるながらも、Androidのアプリ開発には、大きな可能性が広がっている
- ポイントは、グローバルな市場に目を向けることだと思う。

2012年 Google Playの 売り上げで、日本躍進



アメリカを抜き一位
二位は、韓国

2012年12月のランキング

日本勢、上位独占

	Publisher	Rank Change	Category	Headquarters	# of Apps
1	NHN	▲ 1	Various	South Korea	182
2	DeNA	▲ 1	Games	Japan	180
3	GungHo Online	▲ 2	Games	Japan	15
4	GREE	▼ 3	Games	Japan	121
5	COLOPL	▼ 1	Games	Japan	42
6	Com2uS	-	Games	South Korea	84
7	GAMEVIL	▲ 2	Games	South Korea	82
8	Gameloft	▲ 3	Games	France	65
9	NextFloor	▲ 1	Games	South Korea	1
10	SUNDAYTOZ	▲ 7	Games	South Korea	1

<http://blog.appannie.com/app-annie-index-january-2013/>

ABC 2011 Winter での基調講演 で、会が呼びかけたこと





Androidで世界一に

- ❑ 重要なことは、国内市場を狙うのではなく、Androidでグローバルな市場で勝負すること。
 - ❑ まずは、Androidのアプリ開発で、日本が大きな地歩を築くこと。
 - ❑ 当面、この一年以内に、Androidマーケットへのupload数で、世界No1を目指そう。
-

Next Billionへ 次の10億人市場

今日のインターネット人口は20億人。2015年までに、さらに10億人がインターネットを使うようになる。その多くが、モバイルからインターネットにアクセスをするようになる。



http://researcher.ibm.com/researcher/view_project.php?id=1783

Web 3.0:

Social Software in the 3rd World

- ❑ African use of mobile phones has been far more innovative than in Europe, largely because meeting the specific needs of Africa requires innovation.
- ❑ Thinking of poor people as customers, instead of recipients of charity, radically changes the design process.
- ❑ In Cameroon, bankers complain of loan delinquency rates as high as 50 percent. But tontine payments are taken so seriously that borrowers faced with delinquency have been known to commit suicide. ([nyt](#))

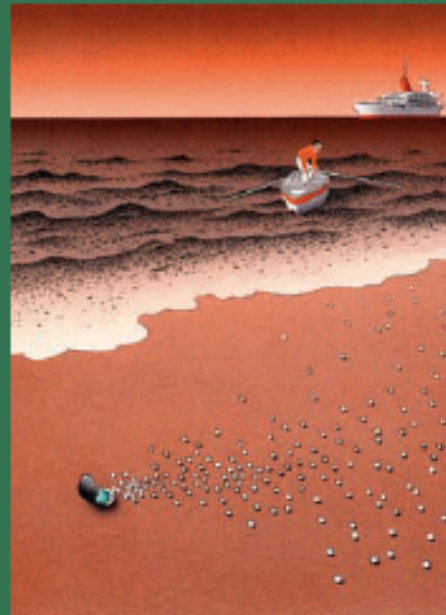
展 望

ボストン コンサルティング グループ

ネクスト・ビリオン
—世界最大の未開拓な
消費者市場

BCG

THE BOSTON CONSULTING GROUP



新市場は今でも、当時と
変わらず大きな魅力を持っ
ているようだ。
今や世界中の企業が現代
版「大航海時代」とばかり
に、ブラジル、中国、インド
さらにはアフリカやその他の
アジア諸国へ大胆に手を
伸ばし、新たな成長の機会
を探している。

Abu Dhabi	Cologne	Lisbon	New Delhi	Stockholm
Amsterdam	Copenhagen	London	New Jersey	Stuttgart
Athens	Dallas	Los Angeles	New York	Sydney
Atlanta	Detroit	Madrid	Oslo	Taipei
Auckland	Dubai	Melbourne	Paris	Tokyo
Bangkok	Düsseldorf	Mexico City	Philadelphia	Toronto
Barcelona	Frankfurt	Miami	Prague	Vienna
Beijing	Hamburg	Milan	Rome	Warsaw
Berlin	Helsinki	Minneapolis	San Francisco	Washington
Boston	Hong Kong	Monterey	Santiago	Zurich
Breusels	Houston	Moscow	São Paulo	
Budapest	Jakarta	Mumbai	Seoul	
Buenos Aires	Kiev	Munich	Shanghai	
Chicago	Kuala Lumpur	Nagoya	Singapore	www.bcg.co.jp

THE BOSTON CONSULTING GROUP





50ドル携帯で、Next Billionへ

- 50ドル製品登場、格安スマホ続々 インドネシア
<http://www.sankeibiz.jp/macro/news/130115/mcb1301150700024-n1.htm>
 - Nokiaの初Windows携帯、Lumina 710が50ドルの超低価格でアメリカ市場に登場
<http://jp.techcrunch.com/2011/12/15/20111214nokias-first-u-s-windows-phone-is-the-50-lumia-710/>
 - 適正価格は50ドル前後？スマホで「次の10億人」狙うグーグル
http://www.nikkei.com/article/DGXNASFK0702Z_X01C12A2000000/
-

基本的なトレンド

- 新しいプラットフォームと
新しいネットワーク・メディアの成立
- コマース市場の拡大と
新しいネットワーク・マーケットの成立
- コンシューマ化の進展と
それに対するオルタナティブ
- クラウド・デバイスの
Webの世界への統合

新しいプラットフォームと 新しいネットワーク・メディアの成立

新しいネットワーク・メディアへの期待

-- コミュニケーションと情報共有への希求

- 巨大なクラウドと無数のクラウド・デバイスの結合による、新しいネットワーク・メディアのプラットフォームの形成
 - 新しいネットワーク・メディアへの期待をドライブしているのは、次のような志向である。
 - 個人と個人の自由なコミュニケーション
 - 個人あるいはグループでの自由な情報共有
 - 個人の豊かな自由時間
-

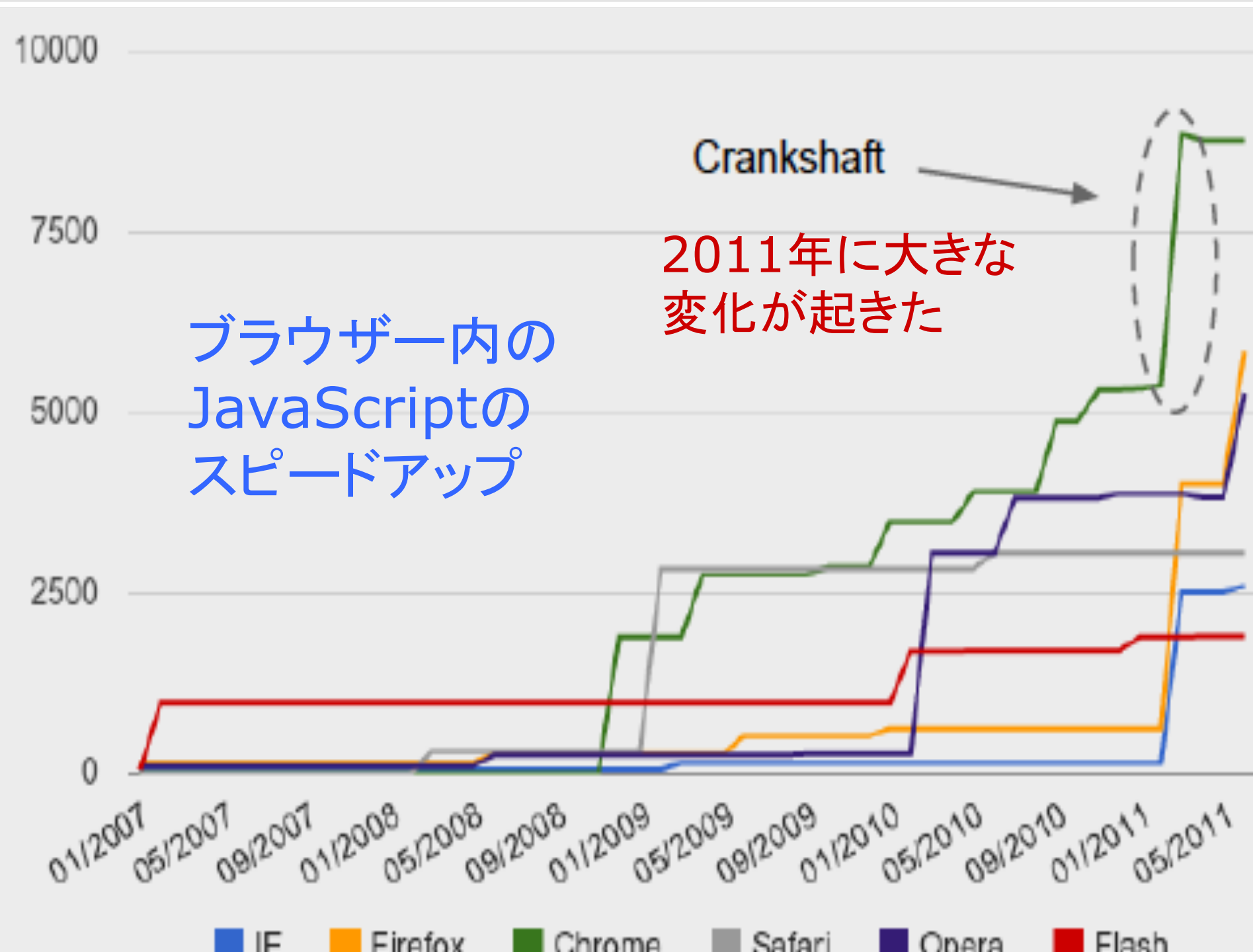
新しいプラットフォームと 新しいネットワーク・メディアの成立

- クラウド＋クラウド・デバイスのプラットフォームは**基本的にはWebの世界の上に**、新しいネットワーク・メディアの骨組みを構成する。
 - クラウドとクラウド・デバイスからなる新しいネットワーク・メディアの躍進は、現代のIT技術のコンシューマ化を牽引している。
 - **このWeb上のネットワークは、21世紀初頭の携帯の拡大のように、2010年代を通じて、世界人口の大部分を飲み込んで行くだろう。**
-

クラウド+クラウド・デバイスの 新しいプラットフォームの形成

	クラウド	クラウド・デバイス	マーケット
Apple	○	○	iTunes/ AppStore
Google	○	○	Android Market
Microsoft	○	○	Marketplace
Amazon	○	○	Amazon
Facebook	△	△	?

Gang of Four の覇権 **2011年**



コマース市場の拡大と 新しいネットワーク・マーケットの成立

コマース市場の拡大と 新しいネットワーク・マーケットの成立

- 日常の消費行動を含め、全ての経済行動が、ネットワーク上で行われるようになる。
 - 現在のエンタープライズ・システムの中核であるWebアプリのスタイルは、変形されつつ、踏襲される。
 - これを担うシステムは、これまでのエンタープライズ・システムの規模を大きく超えるものになる。システムの規模拡大は必須である。
 - 全ての処理がリアルタイムで行われる必要はないのだが、コンシューマ向けのクラウド・サービスとは異なって、そのトランザクション処理には、いくつかの厳しい要請が課せられることになる。
-

エンタープライズ内での デバイス利用の拡大

- 今後、大きな拡大が見込める分野
 - **BYOD** Bring Your Own Device
個人のマシンの会社への持ち込み
 - **BaaS** Backend as a Service
デバイス・アプリ向けのクラウド・サービス。ユーザー管理、データベース、ソーシャルとの連携、プッシュ通信等をサポート
 - **TSA** Thin Server Architecture
デバイス側のパワーを利用するアーキテクチャ（後述）
-

国際金融の世界は、 既にネットワーク化が進んでいる

- 相場は下落していても、買い向かう投資家がたくさんおられたということですし、大量の売り注文を消化する買い注文があったからこそ、**史上最高の出来高である3月15日の57億株**を記録するほどの売買が行われました。
- **史上最高の売買高や注文件数――3月16日は何と2208万件**、我々の最大キャパシティ――四千数百万株の半分くらいに至る、そこまで来主した

2011年3月22日 斉藤東証社長記者会見

GPS+NFC+個人へのヒモ付け

クラウド・デバイスは、最強のマーケット・デバイス

- 消費者の行動を収集・分析して新しいビジネスを起こす上で、クラウド・デバイスは、最強のマーケット・デバイスになろうとしている。
 - スマートフォン側のアプリを作るのは簡単。問題はその先にある。いずれも、ビジネス的にも技術的にも、チャレンジングな課題である。
 - プライバシーの保護
 - ビジネス・モデル
 - プラットホーム
 - クラウド・システム
 - Big Dataを扱う、データマイニングの手法
-

Amazonの狙い

- Amazonは、自社のコマース事業のポータル・デバイスとして、**Kindle Fire**を投入している。
- Gang of Fourで、「もの」を売れる手段をもっているのはAmazonだけである。
- 新しいネットワーク・マーケットの先頭を走っているのは、あるいは、Amazonかもしれない



クレジット決済の新しい流れ

Square / Paypal Here



クレジット決済の新しい流れ

楽天スマートペイ

Step
3

お客様のサイン



Step
4

レシート発行

Step
1

金額入力



Step
2

カード決済



Googleウォレット

2011年11月18日

- Google ウォレットは、客の支払い情報を安全に保存する仮想の財布(ウォレット)で、実店舗でもオンラインでも、すばやく支払いを行うことができる。従来のGoogle Checkout は Google ウォレットに統合
- **2014年までに50%のスマートフォンがNFCを内蔵するようになるだろう。**

コマース担当副社長
Stephanie Tilenius



Google Product Searchを有料化、Google Shoppingに改名

- ❑ 米Googleは現地時間2012年5月31日、米国の商品検索サービス「Google Product Search」を有料モデルに移行し、名称を「**Google Shopping**」に改めると発表した。
 - ❑ Googleは、「販売事業者と商業的関係を持つことは、販売事業者が商品情報を常に最新の状態に保とうとする要因になる。現在の価格、最新の特典、在庫状況などを正確に提供することが、優れたショッピング検索結果をユーザーにもたらす」と説明している。
-

“Empower Local Shopping Through Google Shopping”から

- Googleのミッションは、世界の情報を組織化して、そして、それを普遍的にアクセス可能で役に立つものにするものである。



- 私の近くのどのお店が、ある製品を売っているのか、お店には在庫があるのか、その値段はいくらするのか？
-

Google Shopping



ログイン

September 12, 2012

Fall Fashion

Indulge in designer pieces from Bluefly.com today with Google Offers

Pencils

The No. 2 pencil is No. 1 on our back to school list

Bathtubs

Splish, splash!

Eye Glasses

I can see clearly now...

Quilted Bags

Introducing the perfect finishing touch for any fall outfit

Most Wanted

What's on your fall must-have list?

\$25 for \$50 toward designer brands on Bluefly.com

[Buy Today with Google Offers »](#)



Google offers
beta



Amazon、Kindle Fireシリーズの標準検索エンジンをBingに変更

- New York Times紙が9月9日（現地時間）、AmazonとGoogleの間に現在起こっているある確執について報じている。記事によると、Googleは最近、同社の商品カタログ検索サービス「Google Shopping」での掲載有償化を各ECサイト会社に打診。Amazon側はこれを“Amazonつぶし”に向けた策略と察知し、同サービスから脱退したとのこと。

コンシューマ化の進展と それに対するオルタナティブ

- IT技術のコンシューマ・アイテム化
- ものづくりへの回帰

IT技術のコンシューマ・アイテム化

Apple i-watch





i
It's time to cha



iWatch

Free yourself from the wires.

Works with your Bluetooth-enabled iPod as an interface to check your calendar, look up contacts, read notes, and see what song is playing. Play, pause, skip at the touch of a thumb: The circular, ergonomic controls make it simple.



iPod mini

Now with Bluetooth®.

KEEP CALM
www.LEGION.NET

LeGion

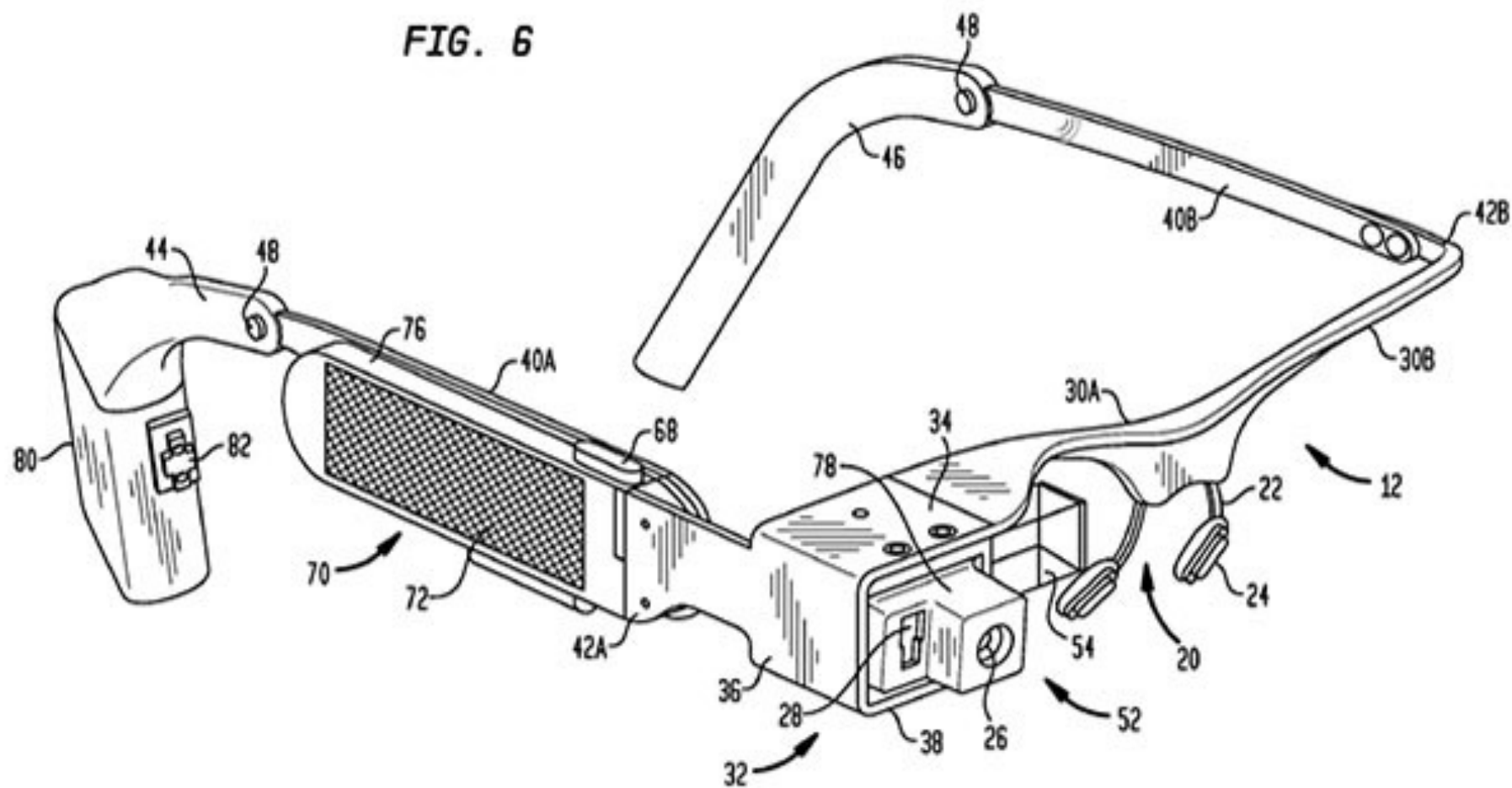
© Copyright 2005 Apple Computer www.apple.com

Google: Google Glass





FIG. 6

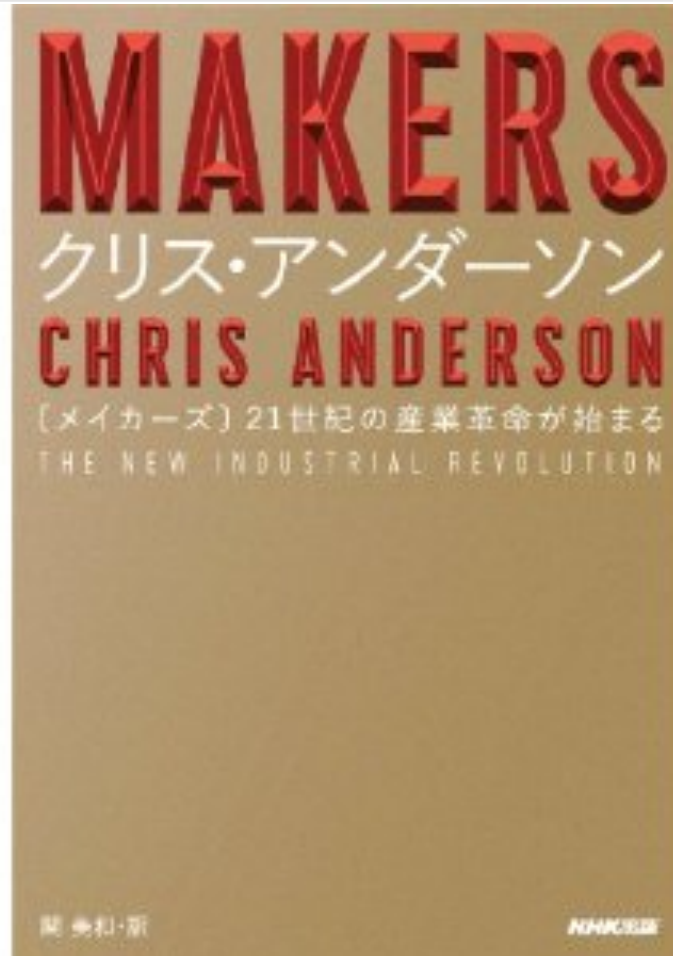




ものづくりへの回帰

MAKERS—21世紀の産業革命が始まる

クリス・アンダーソン



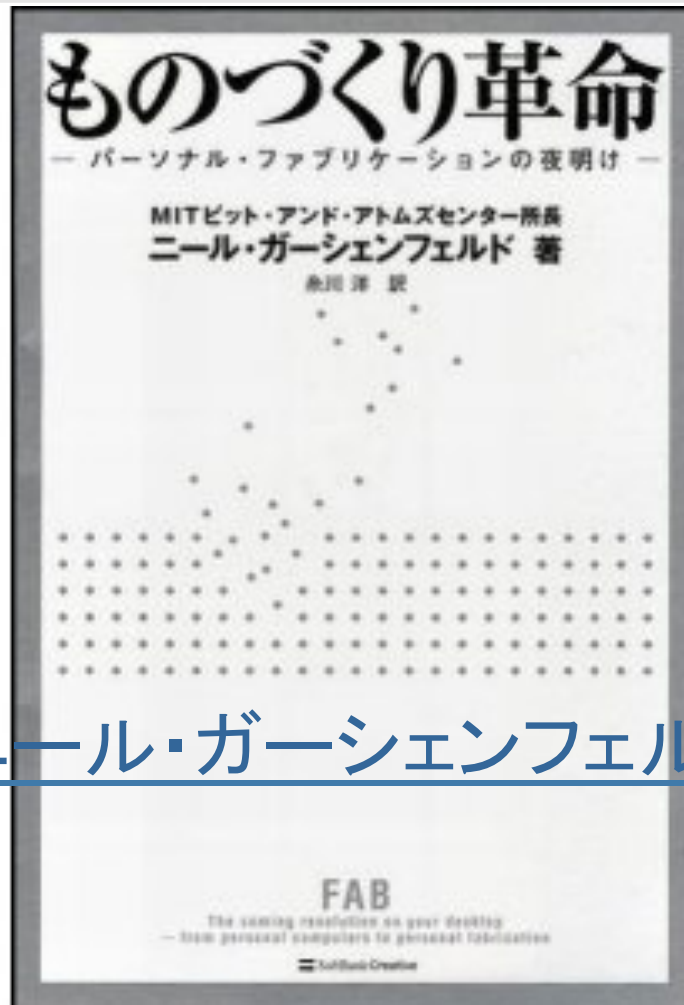
「デジタルによる革命は、 これからが本番だ——」

- 『フリー』以上の衝撃——ベストセラー『フリー』『ロングテール』のクリス・アンダーソンが描く次のパラダイムシフトは〈メイカームーブメント〉だ!
- 21世紀の製造業は、アイデアとラップトップさえあれば誰もが自宅で始められる。ウェブの世界で起こったツールの民主化が、もの作りの世界でも始まったのだ。メイカーズ(モノ作る人々)の革命が、世界の産業構造を再び変える!

「メイカーズ」世代

- 今日の起業家は、オープンソースのデザインと3Dプリンタを使って製造業をデスクトップ上で展開している。
- カスタム製造とDIYによる製品デザインや開発を武器に、ガレージでもの作りに励む何百万人という「メイカーズ」世代が、製造業の復活を後押しする。ウェブのイノベーション・モデルをリアルなもの作りに持ち込むことで、グローバル経済の次の大きな波を起こすのだ。

ものづくり革命 パーソナル・ファブリケーションの夜明け



ニール・ガーシェンフェルド

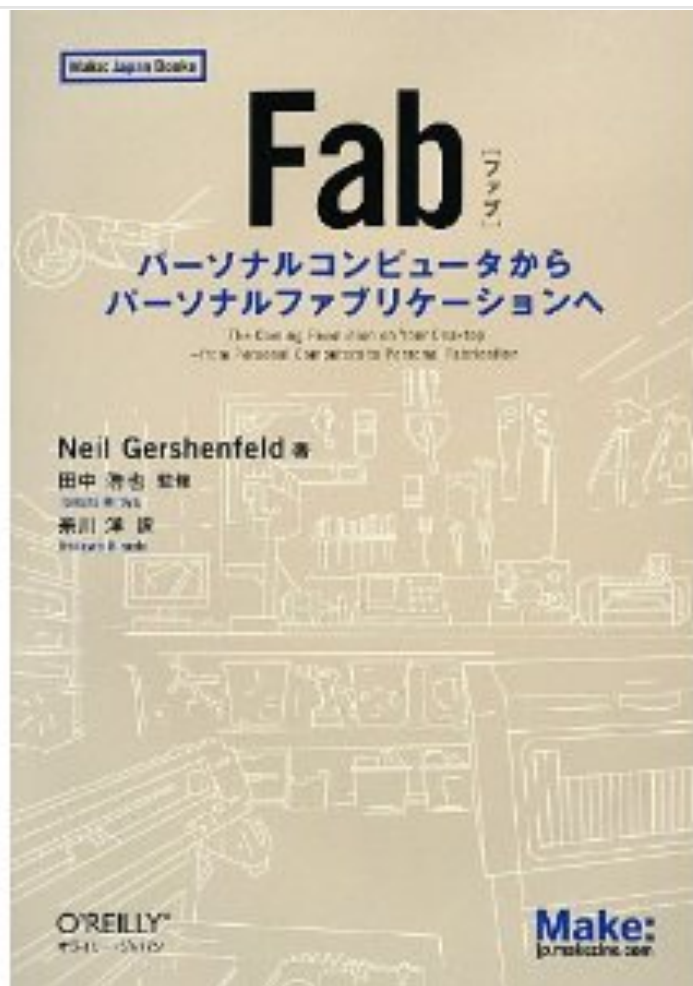
(ほぼ)あらゆるものを自分でつくる

- 大型計算機がデスクトップPCに変わってきたように、各種の工作機械もついにはPCレベルになる日が近い？！
- 過去、人類はまさに「欲しいものは自分でつくって」いました。現代は分業化が進み、誰もが歯車の一部のようになって、大量生産されたものを使うことを余儀なくされています。

「ものづくりの回帰」

- ガーシェンフェルド氏は、より高度な技術を使って、「ものづくりの回帰」を模索しています。人によって、「もの」の理想的な形はさまざまのはず。「欲しい形」を自分で設計し自分で制作できたら、、、
- そんな夢のような話が、すでに世界各地で実践されているのです。「ものづくり」に携わる人のみならず、すべての人に読んでいただきたい先端事例解説書です。

パーソナルコンピュータから パーソナルファブリケーションへ



-
- 本書は、MITビット・アンド・アトムズセンター所長の著者による「ものづくりのデジタル化」と「パーソナルアプリケーション」を解説する書籍です。
 - インド、ボストン、ノルウェー、ガーナなどにおいて、ものづくりとは縁のなかった人々が、自分だけが必要としているものをどんな方法で作ったのか、先進的な事例を紹介。
 - デジタルデータを物体に変換するための2つの手法(減産的技法、加算的技法)や、プログラミング、計測技術、通信技術など、新しいものづくりに必要な技術の解説と合わせて、パーソナルコンピュータの登場に匹敵するこのムーブメントの可能性を明らかにします。
-

3Dプリンター/3Dスキャナー





Making guns in your garage:

how 3D printers will revolutionise the manufacture of deadly weapons



3D Printer

機構部品などを組み立てた状態で出力可能な「EDEN」

大塚商会

1,200万円より



フルカラーでモデル出力が行える
「Zプリンタ」シリーズ
Zコーポレーション

1,180万円



3D Printer



低コストながら美しい仕上がりの
「**MODELA**」シリーズ
ローランドディー・ジー.

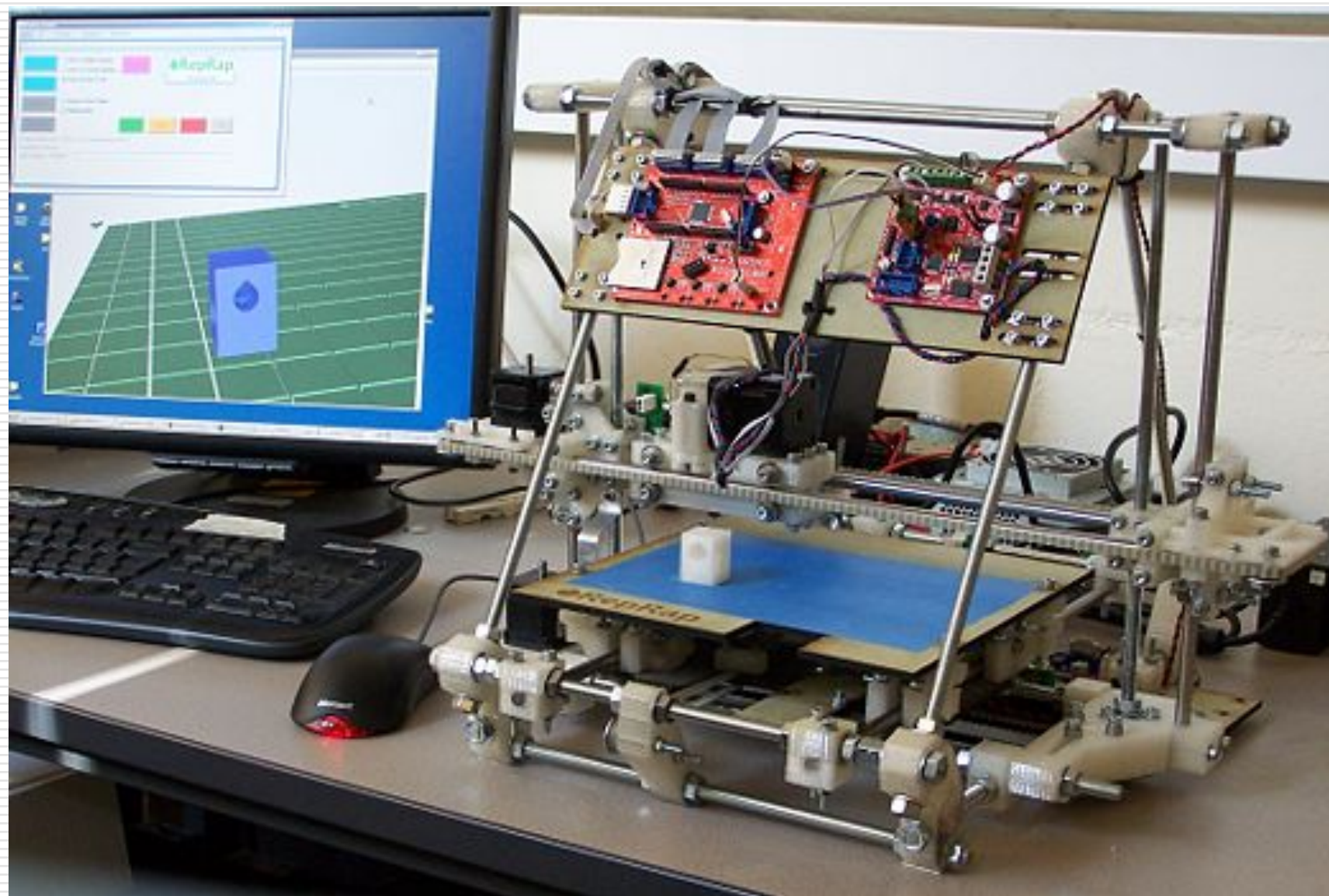
ハイエンド MDX-540SA
390万円

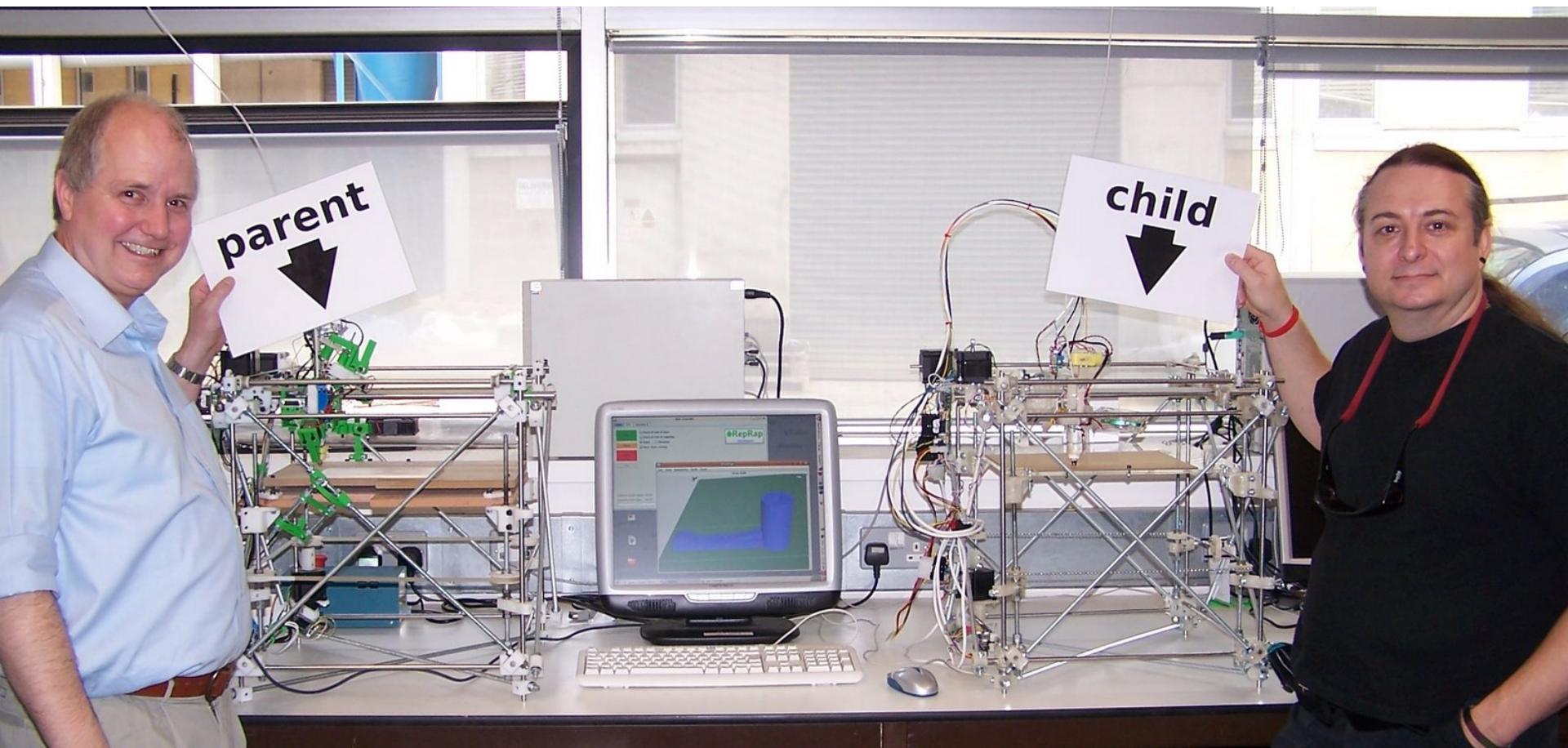
簡単な形状製作であれば、MDX-15
29万8千円

3Dプリンターのオープンソース化

Reprapプロジェクト

- **RepRap** は、プラスチックの物体を作成することができるフリー（自由）なデスクトップ3Dプリンタです。RepRap に使われている多くの部品はプラスチックで作られており、RepRap 自身で作成することができます。
 - 「**Reprap.org** はコミュニティ・プロジェクトである」それは、あなたはこのサイトのほとんどのページを編集し、また新規作成することができるということを意味します。
-





Hacking Conventional Computing Infrastructure

We started a project at Facebook almost two years ago with a pretty big goal: to build one of the most efficient computing infrastructures at the lowest possible cost. We decided to honor our hacker roots and challenge convention by custom designing and building our software, servers and data centers from the ground up – and then share these technologies as they evolve.

The result is a data center full of vanity free servers which is 38% more efficient and 24% less expensive to build and run than other state-of-the-art data centers.

Why Open Hardware?

By releasing Open Compute Project technologies as open hardware, our goal is to develop servers and data centers following the model traditionally associated with open source software projects. That's where you come in.

SERVER designing vanity free, low cost compute nodes

STORAGE designing low cost building block storage

DATA CENTER designing energy efficient data centers

VIRTUAL IO designing open, scalable compute resources

HARDWARE MANAGEMENT designing remote mgmt tools

CERTIFICATION designing standards for solution providers



[Click here for talk and video archives from the May 2012 summit in San Antonio, TX](#)

 [Join a Mailing List](#)

 [Join us on Facebook](#)

 [Follow us on Twitter](#)

 [Follow our Feed](#)

Activity

FEBRUARY 6 VIA OPEN UPDATES

The Open Compute Project is hosting its first international event in Hong...

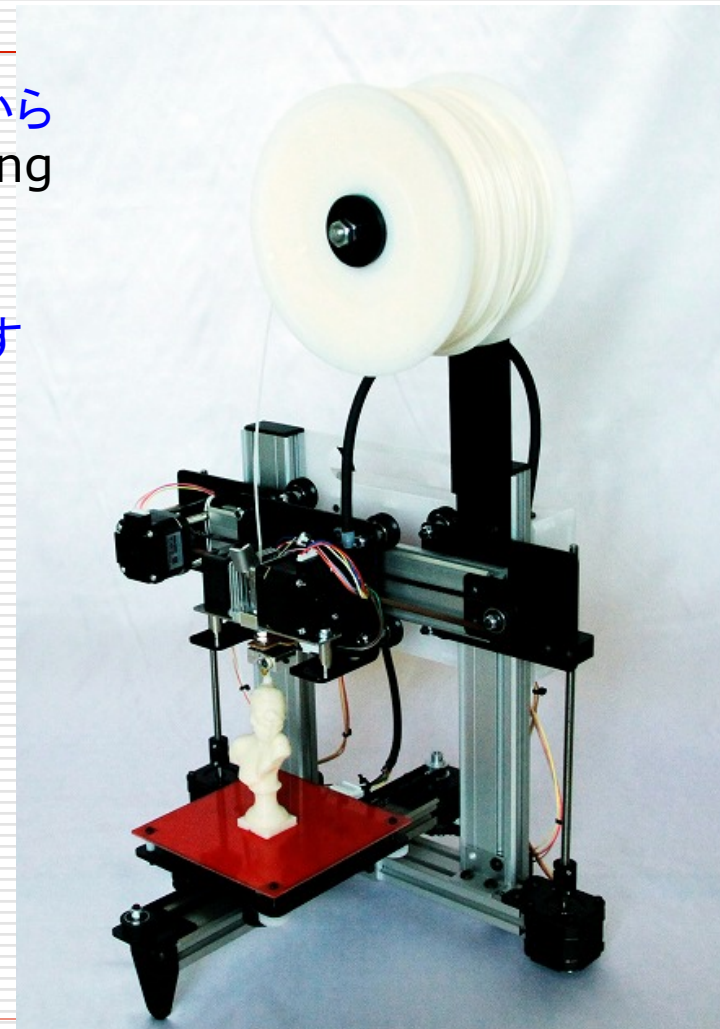
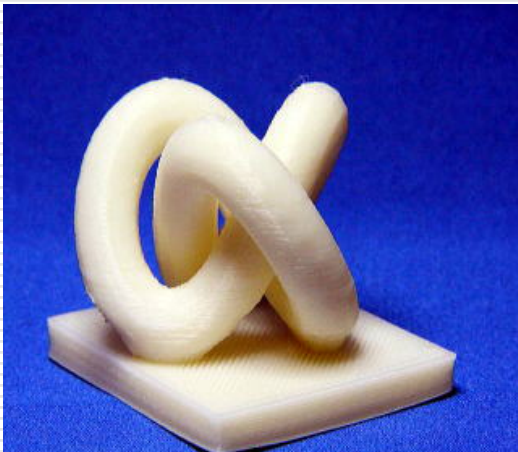
JANUARY 24 VIA TWITTER

3D Printer

「Blade-1」はオープンソースの「RepRapプロジェクト」から派生した3Dプリンターで、アメリカ・シカゴのBarton Dringさんが開発したフレーム一体型レール「MakerSlide」で構成された強固な筐体となっています。

3Dデータをオープンソースのソフト「Printun」内で展開すると、半自動的に立体を造形してくれます。

なお「Blade-1」の標準価格は**136,500円**(税込)です。初回5台限定だった特別価格(税込**99,800円**)分はすでに完売したそうです(4/17修正)



<http://www.ohmsha.co.jp/robocon/a/2012/04/3dblade-1.html>

3D Printer

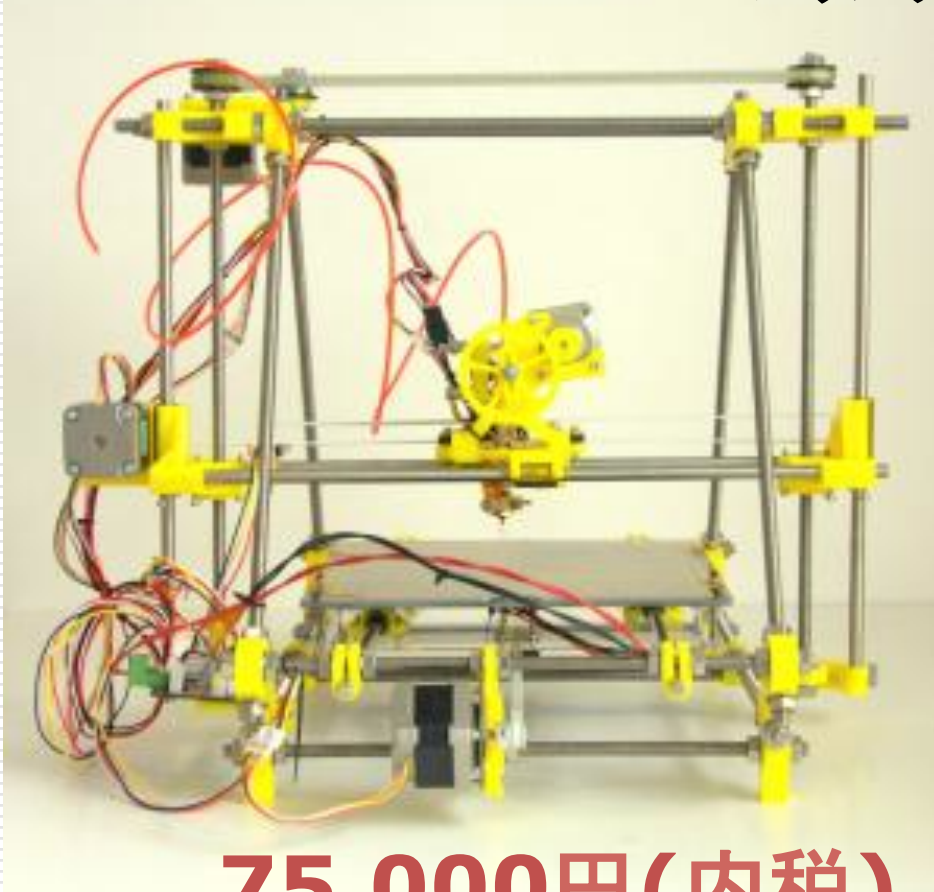
Cubify社の
「Cube 3D Printer」



1,299ドル
12万円

3D Printer

Lunavast PrusaV2 3Dプリンターキット



75,000円(内税)

Android Tablet

- touch screen
- 4 GB memory
- 600 Mhz processor
- accelerometer
- inexpensive

Arduino Mega

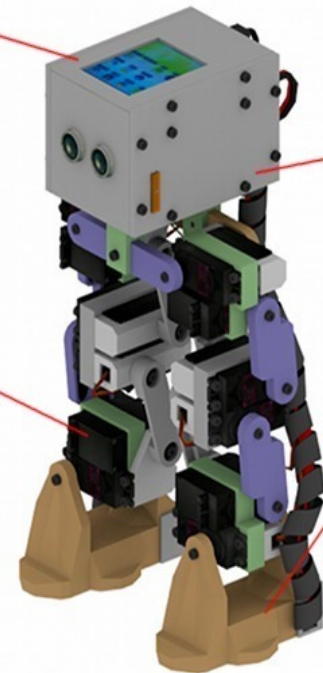
- drives 48 servos
- open source
- easy availability

Servos

- high torque
- inexpensive
- metal gearing

Lipo Batteries

- 1 hour of power
- 2 cells
- easy connector



3D Scanner

3D Scannerに関連した広告 ⓘ

10万円の3Dレーザースキャナ - opt-techno.com

www.opt-techno.com/

全国40以上の大学・高専で利用、mm精度 その他、3Dプリンタ、ハンディ3Dスキャナ

人体、歯、部品等を3D計測する

室内計測、住宅展示のプレゼに利用

ロボットの部品を作りたい（学校）

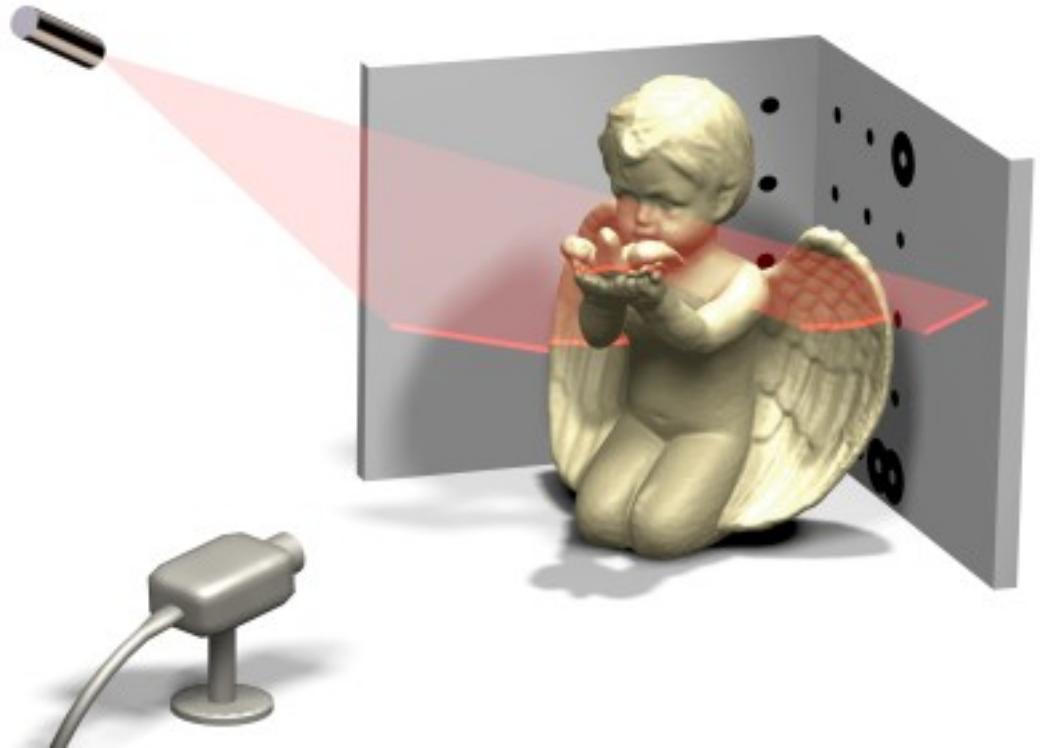
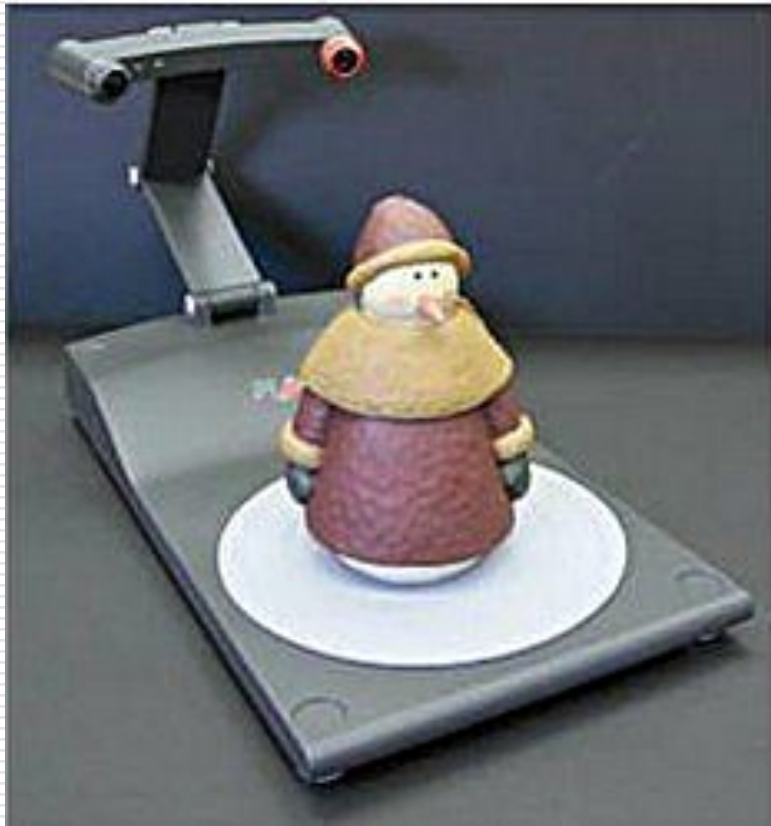
溶接ビード検査システム

レーザースキャナーはFARO - faroasia.com

www.faroasia.com/

498万円～の低価格 重量5kg パソコンいらずで楽々スキャン

3D Scanner



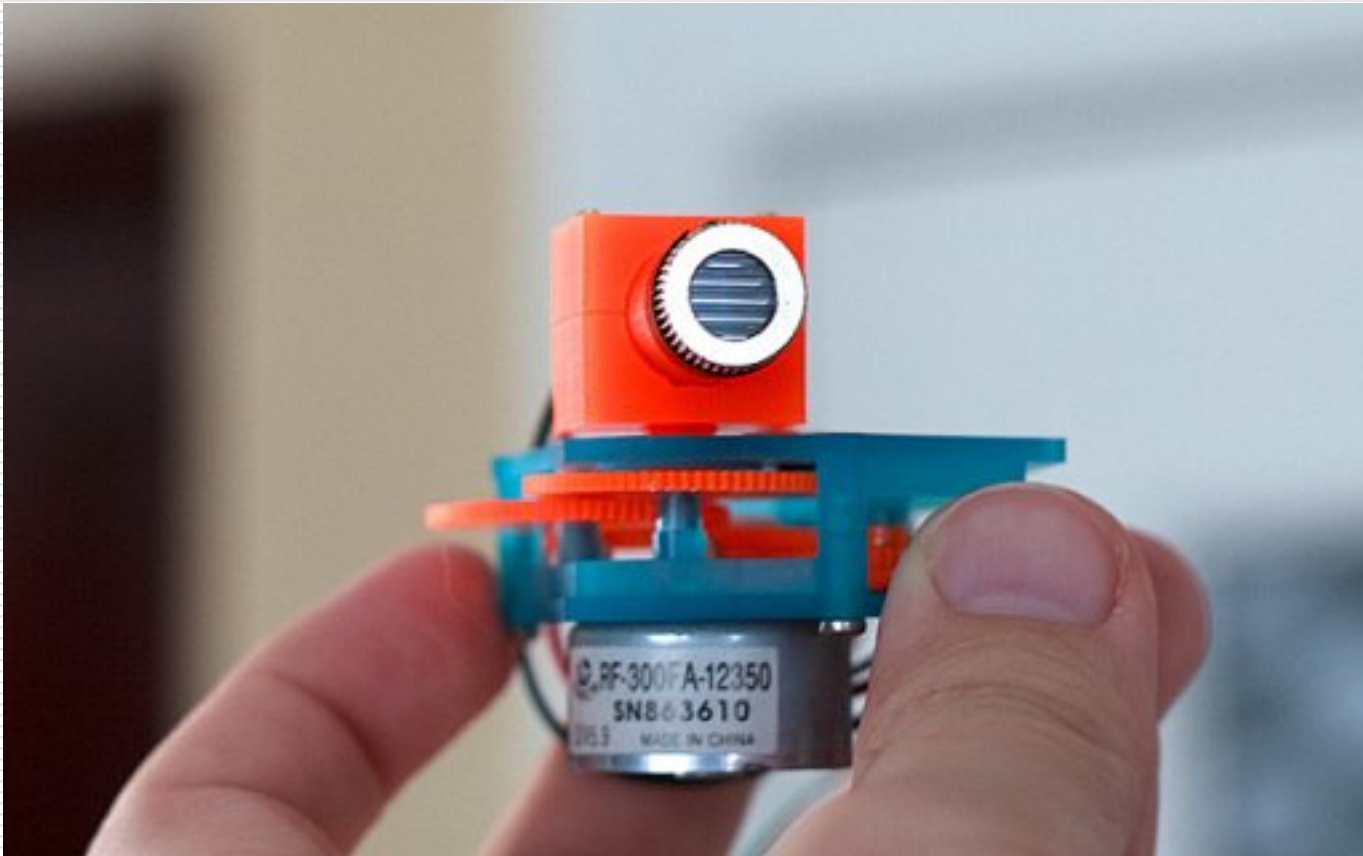
3D Scanner



3D Scanner



3D Scanner



FabLab



FabLab

- ファブラボ(ふぁぶらぼ、英: Fab Lab、**fabrication laboratory**)は、「ほぼあらゆるもの("almost anything")」をつくることを目標とした、3Dプリンタやカッティングマシンなど多様な工作機械を備えたワークショップ。
- 世界中に存在し、市民が自由に利用できる事が特徴。「ほぼあらゆるもの」の中には、大量生産・規模の経済といった市場原理に制約され、いままで作り出されなかったものが含まれる。

FabLab

- ファブラボは、個人が、自らの必要性や欲求に応じて、そうした「もの」を自分(たち)自身で作り出せるようになるような社会をビジョンとして掲げており、それを「ものづくり革命 (Industrial (Re)volution: 第2次産業革命)」とも呼んでいる。「ファブ」には、「Fabrication」(ものづくり)と「Fabulous」(楽しい・愉快的な)の2つの単語がかけられている。

THE CENTER FOR BITS AND ATOMS PRESENTS: THE FAB WORLD

Fab Lab Boston
Mr. Mel King, Director
setcboston@yahoo.com

Fab Lab Chicago
Steven Willis
steven.willis@msichicago.org

Fab Lab Minnesota
Scott Simerson
scott.simerson@century.edu

Fab Lab Ohio
Scott Zitek
rzitek@lorainccc.edu

Fab Lab Rhode Island
(Coming Soon)
David Ortiz
david@as220.org

Fab Lab San Diego
Xavier Leonard
X.L@Headsonfire.org

Fab Lab South Bronx
John Santiago
jsantiago@ssbz.org

Fab Lab Wisconsin
James J. Janisse
jjanisse@fvic.edu

Fab Lab Costa Rica
Milton Villegas Lemus
mvilem@usa.net

Fab Lab Amsterdam, NL, EU
Bas van Abel
bas@waaq.org

Fab Lab Belgium, EU
Nele De Schouwer
nele@technopolis.be

Fab Lab Barcelona, ES, EU
Tomás Díez
tdiez@aacat

Fab Lab Bilbao, ES, EU
(Coming Soon)
Gregorio Rivera
gregorio2020@gmail.com

Fab Lab Iceland
Smari McCarthy
smarim@nmi.is

Fab Lab Utrecht, NL, EU
Joris van Tuerben
joris@protospace.nl



Fab Lab Norway, EU
Haakon Karlson
haakon@fablab.no

Fab Lab Takoradi, Ghana
Emmanuel Azasoo
Azagbey@yahoo.com

Fab Lab Kisumu, Kenya
(Coming Soon)
Vigdis Achola
vachola@yahoo.com

Fab Lab Nairobi, Kenya
Kamau Gachigi
kamaugachigi@yahoo.com

Fab Lab Innovation Hub, ZA
Lindi Mophuti
lmophuti@csir.co.za

Fab Lab Capetown, ZA
Deepuck Chumali
ChumaliD@cpur.ac.za

Fab Lab Free State, ZA
Deon duPreez
ddupreez@out.ac.za

Fab Lab Kimberley, ZA
Thato Riet
fabkimberley@gmail.com

Fab Lab Potchefstroom, ZA
Pieter Tolmay
12126853@rwu.ac.za

Fab Lab Soshanguve, ZA
Khutso Maruma
fabssosh@gmail.com

Fab Lab Bithoor, IN
Prof. Debabrata Goswami
dgoswami@itk.ac.in

Fab Lab Delhi, IN
Dhanaraj V. Gadre
dhanaraj.gadre@gmail.com

Fab Lab Vignyan Ashram, IN
Yogesh Kulkarni
vashram@vsnl.com

Fab Lab Jalalabad, AF
Tim Lynch
Tim.lyndh@vss.af

1.3 設立年度からの傾向

設立年度が2010年以降にみられる主な施設間の共通結果

1) ものづくりのジャンルとして電子工作が多い

2) ものづくり以外の施設利用（無線Lan利用、読書など）がある

3) Twitterを用いた利用者同士の交流がある

4) 設立の動機に運営者の個人的な興味や関心を含む

施設分類	施設名	設立年度	施設の種別	延べ床面積(m ²)	定員数	利用可能時間	標準機材	主な性別・年齢	物・制作ジャンル	利用者にみられる主な利用目的	施設外のコミュニティ	運営者数	設立の契機、目的	益の確保	スタッフからの技術力提供
FabLab β	技術高校 門田ロボテック	1886	教室	十工場:250	16	不定期	なし	代	金属加工	卒業研究のための制作	なし	1	学校に必要な施設として		教員から
その他	某県立大学 創造工房	2009.11	工房	100	10	9:00~17:00	なし	男性、20代	電子工作	趣味、研究	なし	5	美術館に含める新しい用途の検討により		有り
その他	タミヤプラモデルファクトリー レッザ横浜店	2008.3	プラモデル工房		50	20	10:00~21:00	なし	男性、30~40代、親子の利用が多い	プラモデル	プラモデルの作成の他、塗装など家ではできない作業を主に。	5	社長の感心から	商品の販売	店長、プラモデラーから
その他	Y-GSA パワーブランドスタジオ	2010.4	大学製図室	300	50	11:00~17:00	なし	男女、20代	建築	模型制作、設計。	なし	11	空間の拡大のため		有り
FabLab β	多摩美ハッカースペース	2010.4	大学工作室	70	10	9:00~21:00	一部有り	男女、20代	美術作品	機材の試用、作品への活用	なし	1	先生のものづくりへの関心から		有り
FabLab リンク施設	はんだづけカフェ	2010.5	電子工作オープンスペース	20	10	平日18:00~20:30、土日は家族連れも13:00~18:00	一部有り	男女、10~40代	電子工作、プラモデル	電子工作の利用が主。無線Lan利用、集合場所、イベント会場などにも。	Twitter	5	電子工作への関心のもと、電子工作を広めるため	他の事業にて	店員により有り
FabLab リンク施設	ガジェットカフェ	2010.9	ものづくりコミュニケーションスペース	70	最大40	不定期	一部有り	男女、20~40代	電子工作	イベント中心	Twitter	1	店主のものづくりへの関心から	イベントの会費	有り
その他	創造空間 ナノラボ	2011.3	オープンスペース	30	14	火~日 13:00~20:00	一部有り	男性、20~40代	電子工作、機械工作	パソコンの制作、ゲーム制作、ネット利用など	Twitter	4	秋葉原のニーズを想定して	他の事業にて	有り
FabLab	FPGA-CAFE / FabLab Tsukuba	2011.5	コミュニケーションスペース	70	10	日曜のみ。13:00~20:00	完備	男性、20~40代	電子工作	機材の利用。その他技術の相談やコミュニケーション	Twitter	2	カフェ運営への興味、大学の研究の一環	商品の販売	店主から
その他	石巻工房	2011.7	復興支援工房	40	8	9:00~17:00	なし	今後の運営による	木工・手芸	ベンチ、テーブルの制作。ワークショップへの場所貸し	なし	5	復興の促進	商品の販売	有り

Webアプリ / HTML5開発の 新しい動向

「Webアプリ」とは何か？

Webアプリの特徴

- クライアントとしてのWebブラウザの利用
 - データ表現に、HTMLを利用。
 - 通信プロトコルに、HTTPを利用。
 - サーバー・サイド・プログラミング。
-
- 人間がブラウザの前にいることを想定したインタラクティブなアプリ。
-

Webの世代について

- 第一世代 Static Web
 - 第二世代 Dynamic Web
 - 第三世代 Structured Web
 - 第四世代 Real-Time Responsive Web
-

Webを取り巻く環境の変化

- クラウド・デバイスの、飛躍的な拡大。
 - マシンのハードウェア性能の向上。
 - ネットワークの高速化と、それを上回るスピードでのトラフィックの増大。
 - 最新のWeb技術に接する人の増大。
-

変化の三つの方向

- WebプロトコルとHTTPの見直し
 - サーバーとクライアントの役割の見直し
 - 開発言語の見直し
-

Web^oプロトコルとHTTPの見直し

SPDY: HTTP自体の高速化の取り組み

WebSocket: HTTPの置き換え

サーバーとクライアントの 役割の見直し

Rich Client

Thin Server Architecture

サーバーとクライアントの 役割の見直しの一般的な背景

1. ハードの性能が、特にクライアント側で飛躍的に向上したこと。クラウド・デバイスは、PCより、はるかにリッチなクライアントである。
 2. サーバーの負荷の増大
 3. ネットワーク・トラフィックの増大
 4. プログラムとViewの分離の難しさ
 - 全てがサーバー側でコントロールされ、特に、両者が密に絡み合っているようなWebアプリでは、どちらの変更も、開発の工程に大きなインパクトを与える可能性がある。
 - プログラマはデザインの変更を嫌い、デザイナーは、プログラムの変更を嫌う。
-

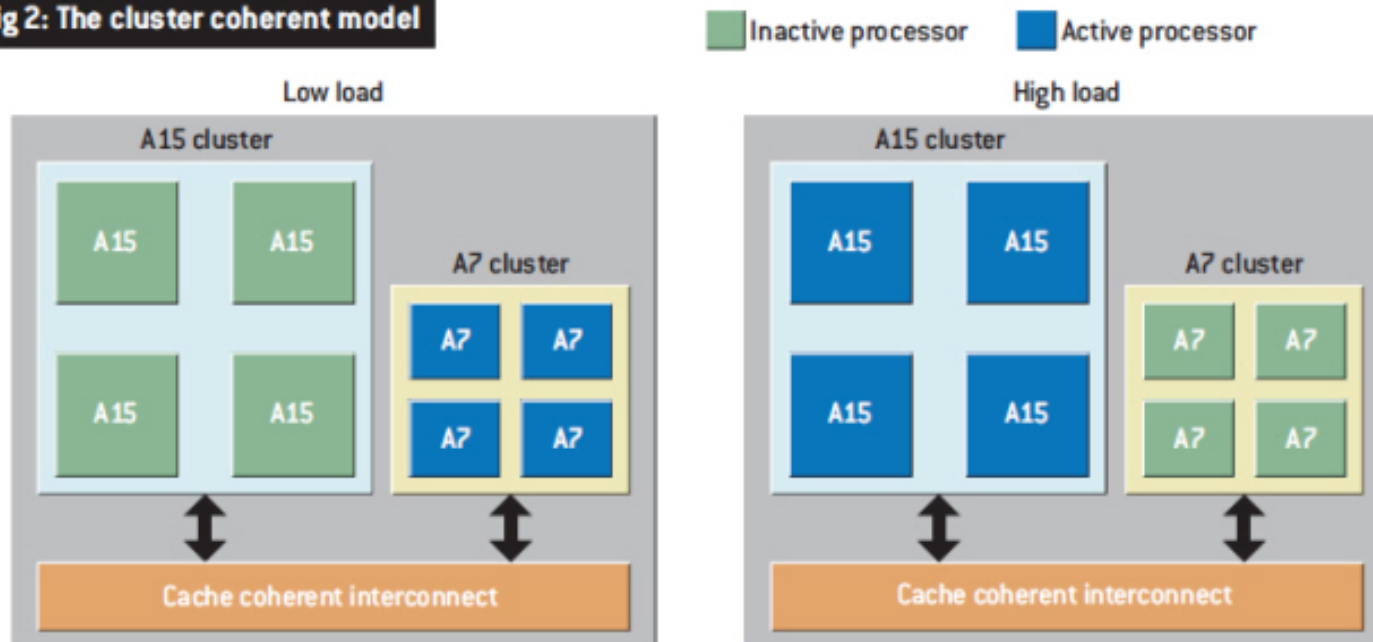
Intel Xeon Phi 60 cores



8 core Android時代の始まり

- ❑ Samsung announces the **8-core Exynos 5** Octa mobile processor
- ❑ **ZTE Apache 8-core** flagship smartphone coming next year(2013)

Fig 2: The cluster coherent model



開発言語の見直し

“Write Once, Run Everywhere”というのは、かつてのJavaのスローガンであった。
Webアプリは、一面ではこれを可能にすると同時に、他面では開発言語の多様化をもたらした。

JavaScriptの役割の拡大

- Webアプリの開発言語の多様化の一方で、重要な変化が進行する。かつては、HTMLのscriptタグの内部に、ひっそりと存在していただけのJavaScriptが、第三世代のStructure Webの時代には、AJAXの手法の主要な担い手として、スポットライトを浴びるようになる。
 - jQueryを初めとして、数十ものJavaScriptライブラリーが登場し、JavaScriptは、事実上のWebアプリの標準言語になった。どのような開発言語を選んだとしても、多くのWebアプリの開発では、JavaScriptのコーディングが必要になってきている。ResponsiveなWebアプリを作ろうと思ったら、なおさら、JavaScriptの知識は、必須になる。
-

開発者の負担

- いまや、PCに代わってクライアントの標準になろうとしている、スマートフォンやタブレットといったクラウド・デバイスの普及は、クラウド・デバイス上のWebアプリ開発のニーズを、今まで以上に高めている。
 - ただ、クラウド・デバイスのベンダー毎の多様化の進行は、アプリの開発者を、ベンダーの特定の開発言語に縛り付けている。
 - 全く同じ機能のアプリを開発するのに、クラウド・デバイスのベンダーと機種に応じて、開発環境を変えないといけないというのは、合理的でも経済的でもない。
-

ユーザーのニーズ

- 一方で、ユーザーは、PCだけではなく、スマートフォンやタブレットといった複数のクラウド・デバイスを所有するようになってきている。この流れは、テレビやゲーム機がクラウド・デバイス化することによって、さらに大きなものになるだろう。
 - こうしたマルチ・デバイス、マルチ・スクリーン環境では、ユーザーは、同じ機能のアプリが、いつでも、どこでも、どんなデバイスの上でも動くことを望むようになる。
 - そうしたニーズは、ベンダー毎に分断された開発環境のもとでは、開発者側の負担を増大させるだけである。
-

Lingua Francaへの期待

- こうした中で、本当の意味での“Write Once, Run Everywhere”を実現する言語への期待が高まっている。
 - そのもっとも有力な候補は、JavaScriptに他ならない。おそらく、今後の言語のイノベーションの中心は、JavaScriptを舞台に展開されることになると思われる。ここでも、様々な模索が行われている。
-

クラウド・デバイス側の状況

1. クラウド・デバイスの普及に伴って、どんなデバイスでも動くアプリへのニーズが、市場に生まれてきている
 2. クラウド・デバイスのアプリでも、ネットワークに接続するアプリが確実に増え始めている
 3. こうして、クラウド・デバイスのアプリ側では、「アプリをWebアプリ化する」ことが、重要な選択肢の一つとして浮かび上がってきている。
- 基本的には、それは、HTML5への期待として、「アプリを、HTML5を利用して、Webアプリ化する」という展望として定式化出来ると思う。
-

奇妙な状況・混乱

- サーバー側では、HTTPを含んだWebアプリの見直しの動きが起きている、その時に、クライアント側では、Webアプリへの期待が高まっている。
 - アプリへのHTML5の導入に熱心だったFacebookが、一転して、それを「最大の誤り」として、ネイティブの開発に転換したのも、現在の混沌とした状況を表している。
-

「Webアプリ」概念の変化

WebアプリとAndroid Nativeアプリ

J2EE, Spring
.ASP, .NET
等のサーバー
サイドアプリの
クライアント

WebアプリとAndroid Nativeアプリは、
現在は、ほとんど独立に存在している。
それは、なぜか？

**Android
Native**

WebアプリとAndroid Nativeアプリ

J2EE, Spring
.ASP, .NET
等のサーバー
サイドアプリの
クライアント

Androidのアプリの大部分が、ゲームを中心とした個人向けのAndroid単体で動くアプリであったこと。

Androidアプリで、クラウドのサービスを利用するアプリが、これまでは少数だったこと。

Androidが、Webアプリ中心のエンタープライズ業務の領域で十分に利用されてこなかったこと。

Androidの標準のWebブラウザーのHTML5対応が遅れていたこと。

**Android
Native**

Webアプリ+HTML5への期待

Webアプリ
+
HTML5の新機能

こうした現状の中で、少なくない開発者は、エンタープライズの世界でこの10年の間に歴史的な成功を収めてきたWebアプリとHTML5の新しい機能が結合することで、新しいアプリの世界が大きく拡大すると考えてきた。

**Android
Native**

Webアプリ+HTML5への期待

Webアプリ
+
HTML5の新機能

残念ながら、こうした展望は、そのままの形では実現されてはいない。
昨年 of Facebook の HTML5 から Native の突然の変更は、多くの人に驚きを持って受け止められた。ただ、それには理由がある。

**Android
Native**

Webアプリ+HTML5への期待

Webアプリ
+
HTML5の新機能

従来のサーバー・サイドのWebアプリのスタイルに、サーバーの負担・ネットワーク帯域の消費等、いろいろな困難がうまれていて、それらの見直しが進行中であること。

その一方で、クライアントのデバイスの処理能力が飛躍的に向上して、単なるサーバーの生成するHTMLをブラウザでレンダリングする以上の能力を持ち始めたこと。

**Android
Native**

アプリケーションの多様化

J2EE, Spring
.ASP, .NET
等のサーバー
サイドアプリの
クライアント

**Application
Cache**を利用
したアプリ

Widget

**Packaged
Web App**
Tizen
Firefox OS

**JavaScript
MVC
Frameworks**

**Android
Native**

現実に行進しつつあるのは、「Web
アプリ+HTML5」の世界の一方的
拡大ではなかった。

アプリケーションの多様化

J2EE, Spring
.ASP, .NET
等のサーバー
サイドアプリの
クライアント

**Application
Cache**を利用
したアプリ

Widget

**JavaScript
MVC
Frameworks**

**Packaged
Web App**
Tizen
Firefox OS

**Android
Native**

サーバーサードのWebアプリと、クライアント中心のNativeアプリの中間に、様々なアプリケーションの形態が生まれつつある。

Web Application on Mobile

- ❑ Application Cache
- ❑ Widget
- ❑ Packaged Web App

Application Cache

- 基本的には、サーバー・サイドのWebアプリのスタイル。サーバーとの接続が途絶えた時などに、あらかじめmanifestファイルで宣言されたキャッシュ上のリソースを読み込む。
 - HTML5の仕様書
<http://www.whatwg.org/specs/web-apps/current-work/#applicationcache>
の6.7節 **Offline Web applications** 以下に詳細に定義されている。
-

Cache Manifestの例

```
<!-- clock.html -->
<!DOCTYPE HTML>
<html manifest="clock.appcache">
  <head>
    <title>Clock</title>
    <script src="clock.js"></script>
    <link rel="stylesheet" href="clock.css">
  </head>
  <body>
    <p>The time is: <output id="clock"></output></p>
  </body>
</html>
```



clock.appcache

CACHE MANIFEST
clock.html
clock.css
clock.js

キャッシュ・マニフェスト・ファイル
*.appcache のMIME Typeは、
text/cache-manifest

```
/* clock.css */
output { font: 2em sans-serif; }
```

```
/* clock.js */
setInterval(function () {
  document.getElementById('clock').value = new Date();
}, 1000);
```

Application Cache API

```
interface ApplicationCache : EventTarget {
```

```
    // update status
```

```
    const unsigned short UNCACHED = 0;
```

```
    const unsigned short IDLE = 1;
```

```
    const unsigned short CHECKING = 2;
```

```
    const unsigned short DOWNLOADING = 3;
```

```
    const unsigned short UPDATEREADY = 4;
```

```
    const unsigned short OBSOLETE = 5;
```

```
    readonly attribute unsigned short status;
```

```
    // updates
```

```
    void update();
```

```
    void abort();
```

```
    void swapCache();
```

Application Cache API

```
// events
    attribute EventHandler onchecking;
    attribute EventHandler onerror;
    attribute EventHandler onnoupdate;
    attribute EventHandler ondownloading;
    attribute EventHandler onprogress;
    attribute EventHandler onupdateready;
    attribute EventHandler oncached;
    attribute EventHandler onobsolete;

};
```

Widget

- ❑ Widgetは、WebアプリをZIPファイルの形で配布しようと言うもの。ZIPファイルには、XML形式の構成ファイルが含まれていて、それによって、アプリの署名、アクセス・コントロール、ネットワーク利用の制限等を行う。
- ❑ デフォルトのsecurity policyは、全ネットワーク・アクセスのブロックである。
- ❑ 仕様は、Packaged Web Apps (Widgets) - Packaging and XML Configuration (Second Edition)

<http://www.w3.org/TR/widgets/>

ZIPファイルの取得

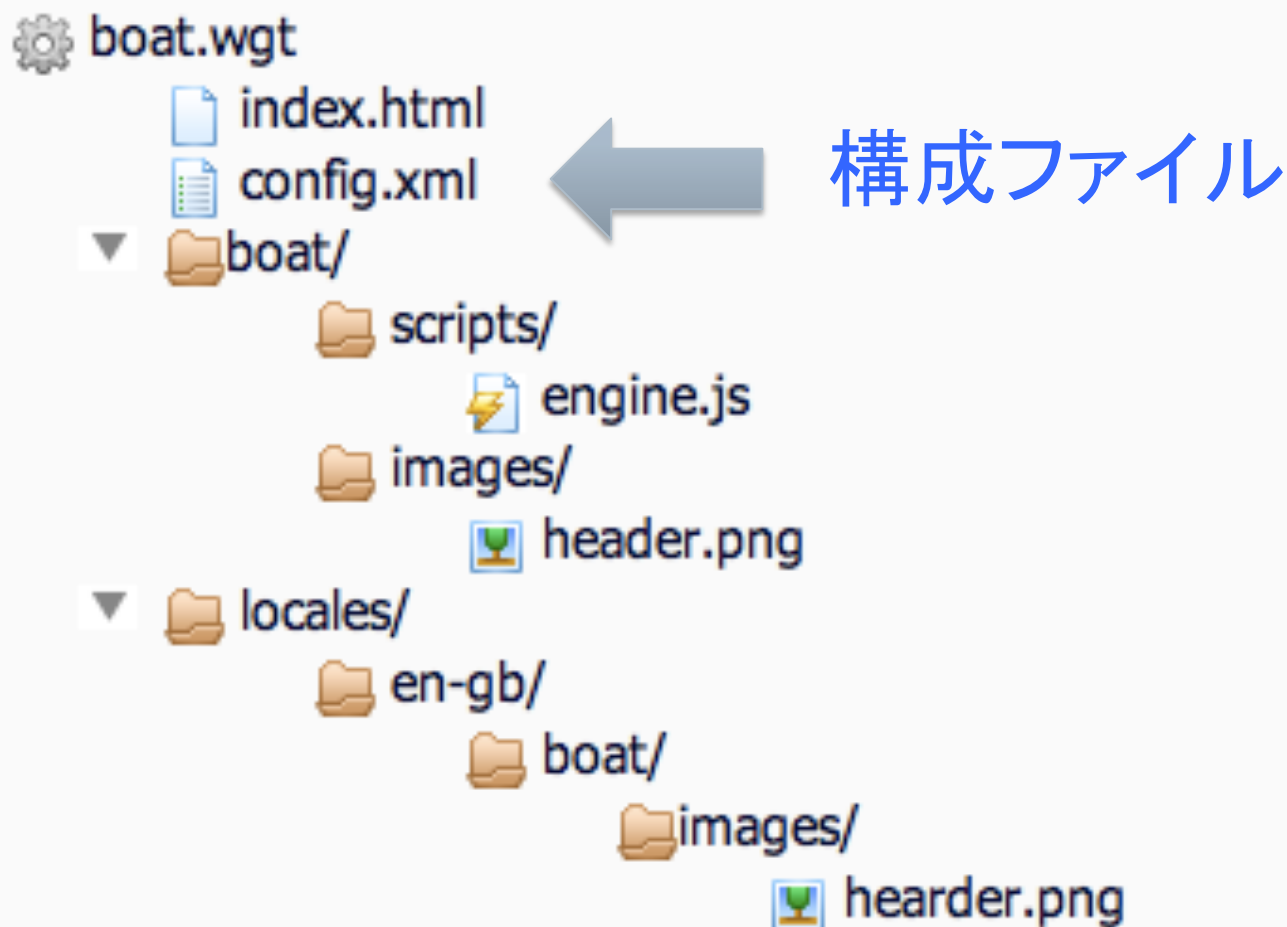
Request

GET /foo.wgt HTTP/1.1
Host: www.example.com
Accept: application/widget

Response

HTTP/1.1 200 OK
Date: Tue, 04 Sep 2007 00:00:38 GMT
Last-Modified: Mon, 03 Sep 2007 06:47:19 GMT
Content-Length: 1337
Content-Type: application/x-gadget ←-- application/widget

ZIPファイルの構成例



Widget Configurationファイルの例

```
<?xml version="1.0" encoding="UTF-8"?>
<widget xmlns      = "http://www.w3.org/ns/widgets"
      id          = "http://example.org/exampleWidget"
      version     = "2.0 Beta"
      height      = "200"
      width       = "200"
      viewmodes  = "fullscreen">

  <name short="Example 2.0">
    The example Widget!
  </name>

  <feature name="http://example.com/camera">
    <param name="autofocus" value="true"/>
  </feature>
```

Widget Configurationファイルの例

```
<preference name    = "apikey"  
          value    = "ea31ad3a23fd2f"  
          readonly = "true" />
```

```
<description>
```

A sample widget to demonstrate some of the possibilities.

```
</description>
```

```
<author href = "http://foo-bar.example.org/"  
      email = "foo-bar@example.org">Foo Bar Corp</author>
```

```
<icon src="icons/example.png"/>
```

```
<icon src="icons/boo.png"/>
```

```
<content src="myWidget.html"/>
```

Widget Configurationファイルの例

<license>

Example license (based on MIT License)

Copyright (c) 2008 The Foo Bar Corp.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, INSULT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

</license>

</widget>

Widgetを実装したブラウザ

- Widgetは、Web Application Working GroupのRecommendationになっているがその仕様を完全に実装したブラウザは多くはない。
 - Aplix Web Runtime
 - Opera 11.5 Beta
 - OBIGO X10
 - Apache Wookie 0.9
 - Borqs WRT
-

Packaged Web App

- 第三の方法が、Packaged Web Appである。System Applications Working GroupとWeb Applications Working Groupで仕様の検討が始まったばかり。
- 非公式のドキュメントは、“Runtime and Security Model for Web Applications”
<http://sysapps.github.com/sysapps/proposals/RunTime-Security/Overview.html#packaged-applications>
これは、話題の**Firefox OSの仕様**を踏襲している。

Packaged Web App

- ❑ Packaged Web Appの特徴は、**Packaged Application**と**Hosted Application**という二つのタイプのWebアプリに対応していること。
- ❑ Hostedアプリは、サーバーが必要な従来型のWebアプリと考えていいが、Packagedアプリは、アプリの配布にもサーバーの設置を必要としない。
- ❑ 問題は、こうしたアプリ群も、「Webアプリ」と呼ばれていること。(On Mobileだが)ただ、それは従来のサーバーサイドのWebアプリとは違うものだ。

Application Manifest

```
{
  "name": "MozillaBall",
  "description": "Exciting Open Web development action!",
  "launch_path": "/",
  "version": "1.0",
  "icons": {
    "16": "/img/icon_16.png",
    "48": "/img/icon_48.png",
    "128": "/img/icon_128.png"
  },
  "developer": {
    "name": "Mozilla",
    "url": "https://mozilla.org/en-US"
  },
  "appcache_path": "/cache.manifest",
```

Application Manifest

```
"locales": [  
  "es": {  
    "description": "¡Acción abierta emocionante del desarrollo del  
Web!",  
    "developer": {  
      "url": "https://mozilla.org/es-ES"  
    }  
  }  
],  
"default_locale": "en",  
"screen_size": {  
  "min_width": "600",  
  "min_height": "300"  
},  
"required_features": [  
  "touch", "geolocation", "webgl"  
],
```

Application Manifest

```
"permissions": {  
  "contacts": {  
    "description": "Required for autocompletion in the share screen",  
    "access": "read"  
  }  
},  
"fullscreen": "true",  
"relNotes": {  
  "1.0": "Bugs fixed. New exciting effects. Ready for an official  
release!"  
  "0.5": "First alpha version with still some bugs but already fun!"  
}  
}
```

Application Interface

```
interface Application {
    readonly attribute DOMString    origin;
    readonly attribute Object       manifest;
    readonly attribute DOMString    installOrigin;
    readonly attribute unsigned long installTime;
    readonly attribute Object       parameters;
    DOMRequest    launch ();
    DOMRequest    uninstall ();
    readonly attribute DOMString    updateState;
    readonly attribute unsigned long downloadSize;
    DownloadRequest downloadUpdate ();
    readonly attribute DOMString    state;;
    void                hide ();
    void                exit ();
        attribute EventHandler onlaunch;
        attribute EventHandler onpause;
        attribute EventHandler onresume;
        attribute EventHandler onterminate;
};
```


Packaged Applicationの manifest.webappの例

□ demopaid

- index.html

- **manifest.webapp**



```
{
  "name": "Demo Paid",
  "developer": {
    "name": "Mozilla Labs Demo",
    "url": "http://mozillalabs.com"
  },
  "description":
    "Testing the paid app functions",
  "icons": {
    "128": "/icon.png"
  },
  "launch_path":
    "/examples/demopaid/index.html",
  "installs_allowed_from": ["*"]
}
```

Packaged Applicationの manifest.webappの例

□ mozillahead

- icon.png
- index.html
- jquery-1.4.2-mini.js
- **manifest.webapp**



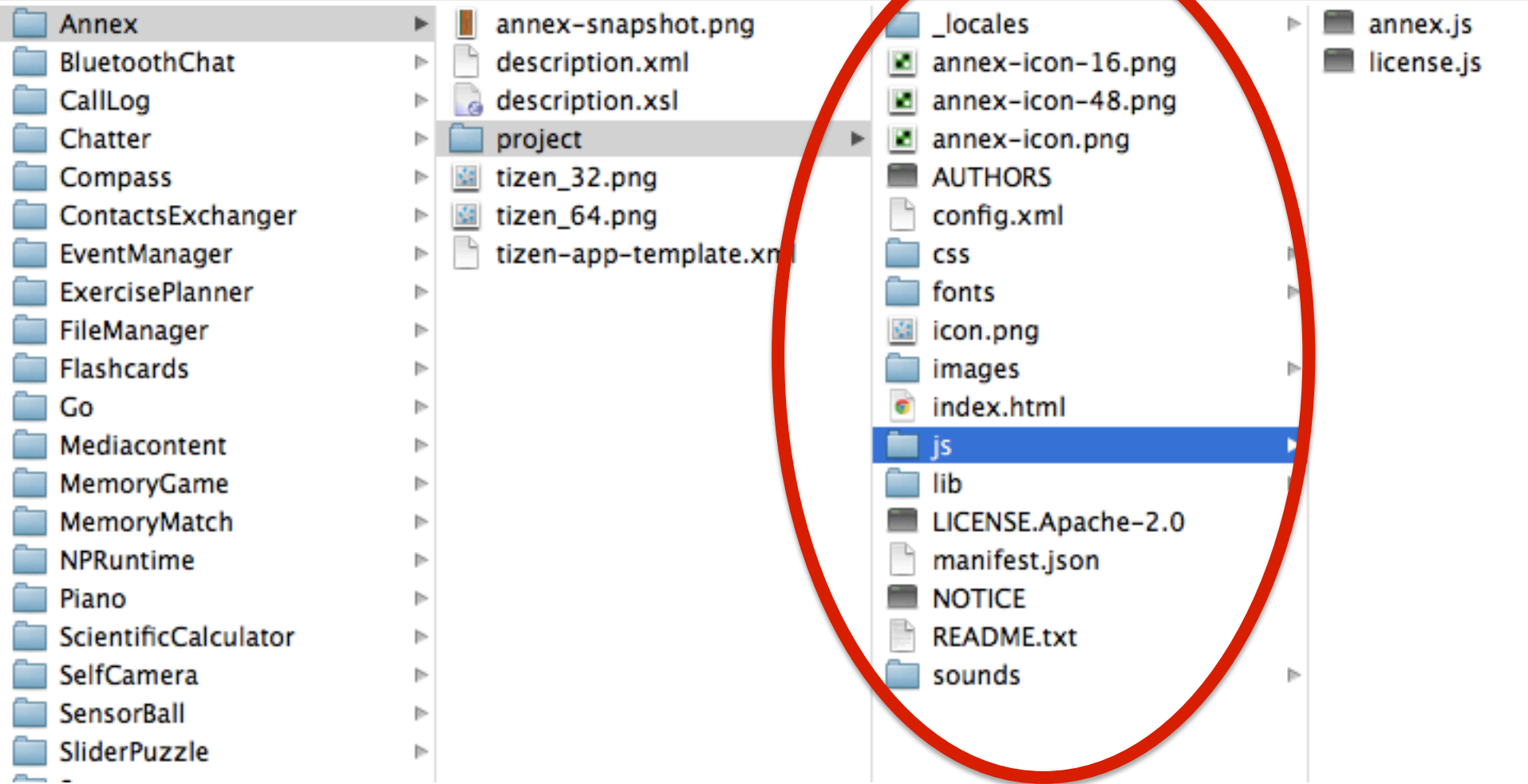
```
{  
  "name": "MozillaHead Danger Muffin  
    Attack",  
  "developer": {  
    "name": "Mozilla Labs Demo",  
    "url": "http://mozillalabs.com"  
  },  
  "description": "Got Mozilla on the  
    Brain!?",  
  "icons": {  
    "128": "/icon.png"  
  },  
  "widget":  
    { "width": 250, "height": 100 },  
  "default_locale": "en"  
}
```

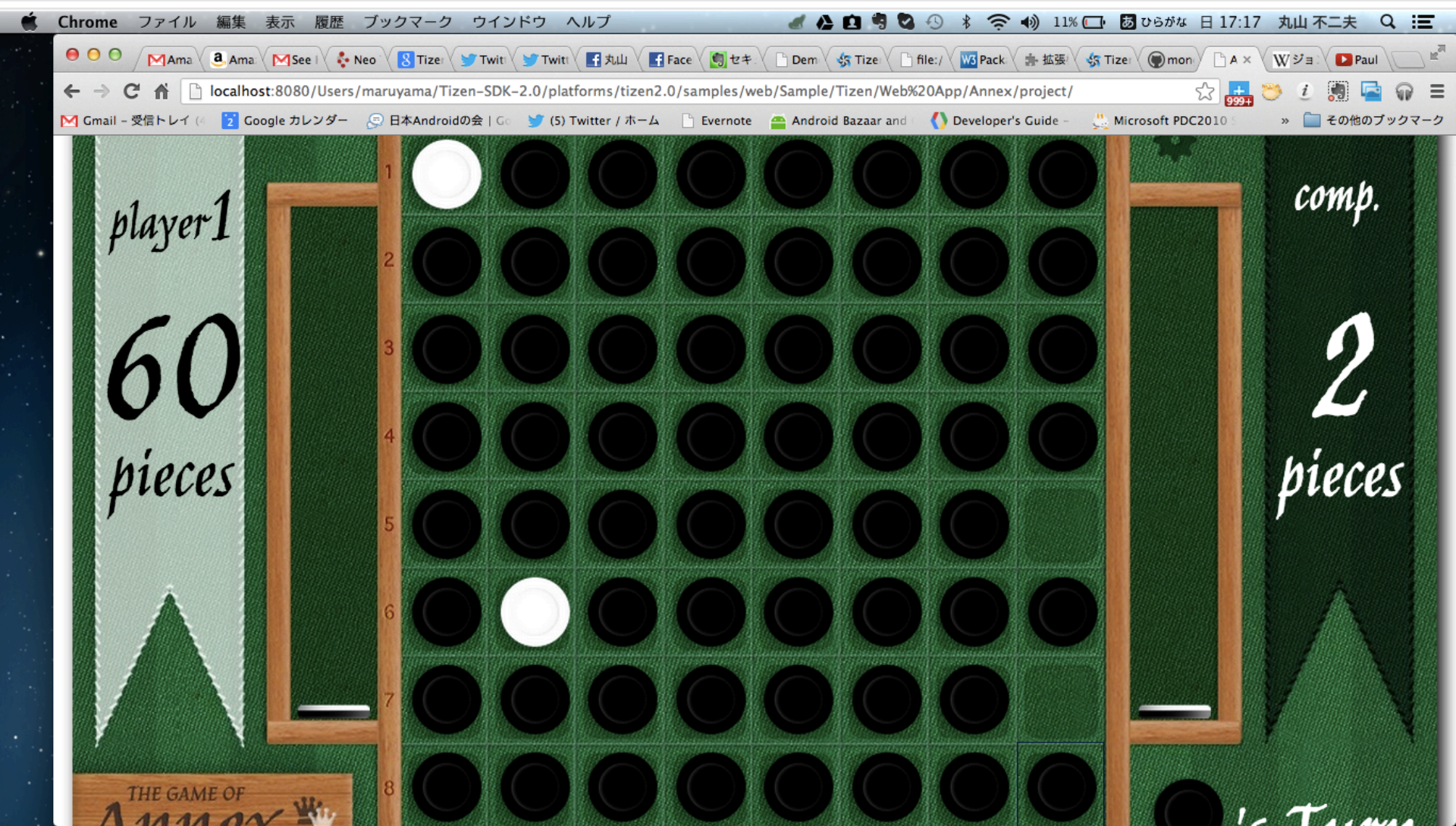
This is MozillaHead!



This page exists as an example of how a normal website can be wrapped up as an Open Web App, targeted at people with mozillas on their head!

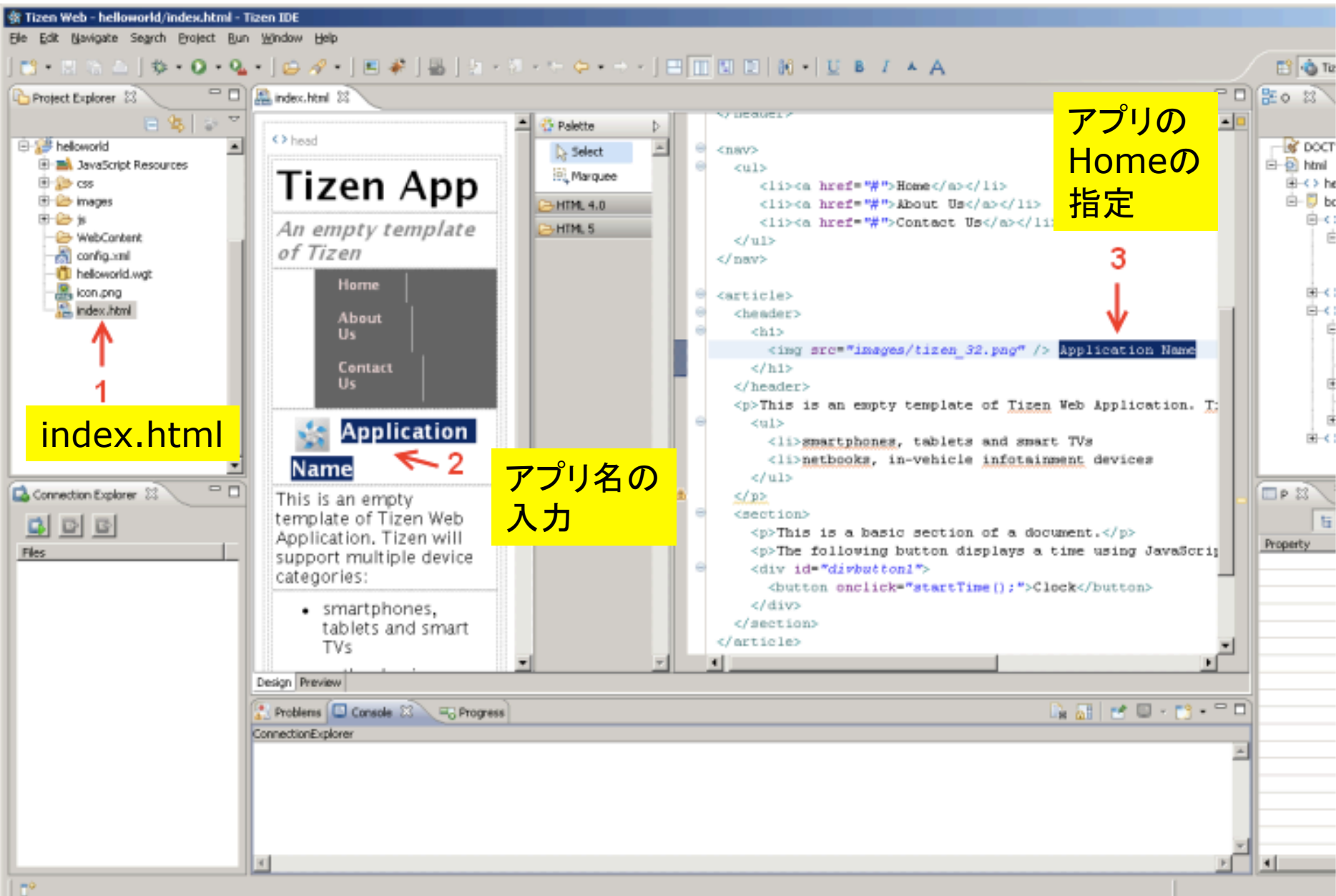
Tizenのパッケージング





ローカルにWebサーバーを立てることが必要だった。

TizenのIDE環境



JavaScript MVC Frameworks

- ますます沢山のロジックがブラウザーの内部で実行されるようになるにつれて、フロントエンドのJavaScriptのコードベースは、ますます大きくなり、維持・管理するのが難しいものになってゆく。

- Backbone.js

- AngularJS

- Knockout

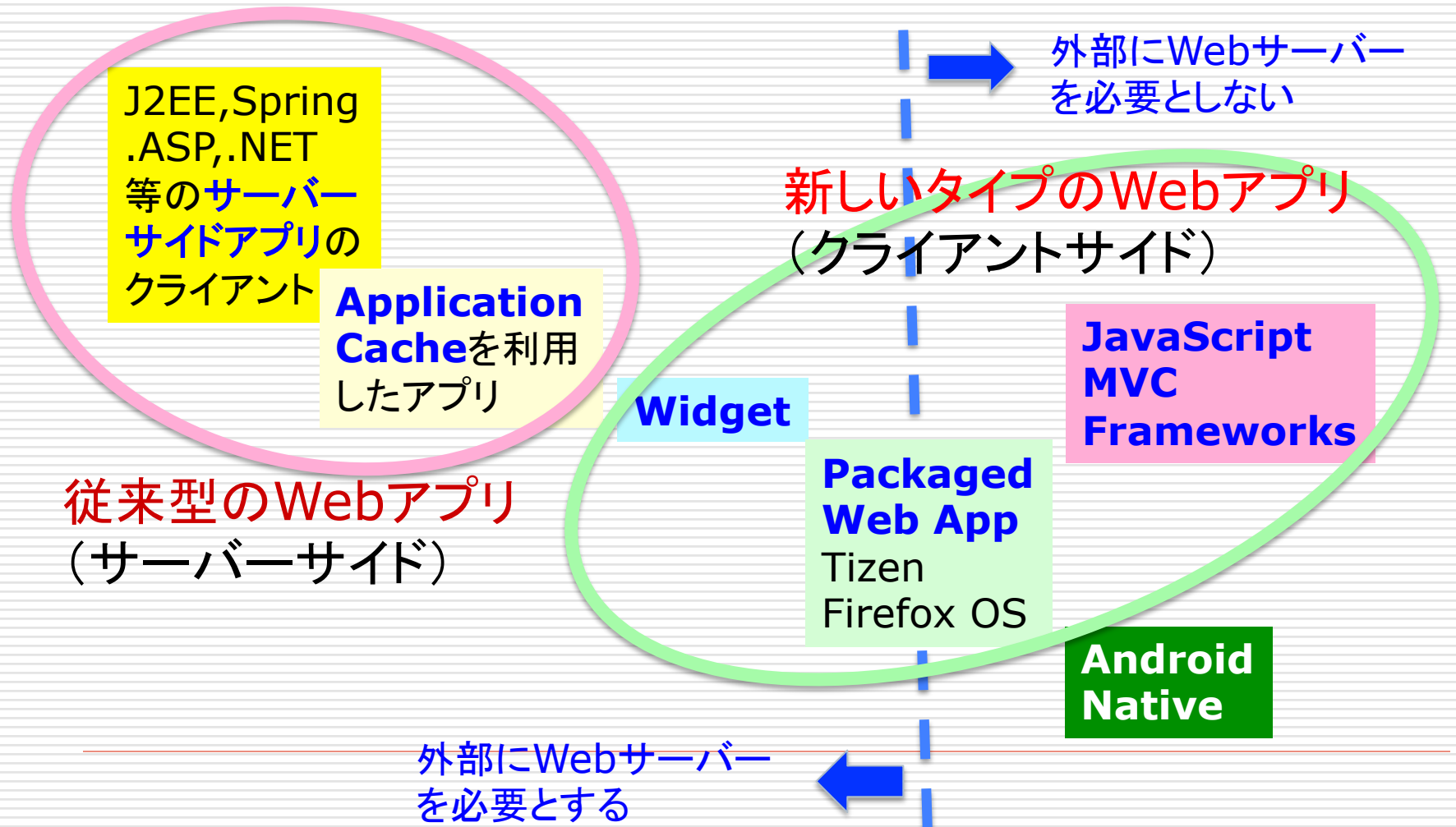
- Agility.js

- CanJS

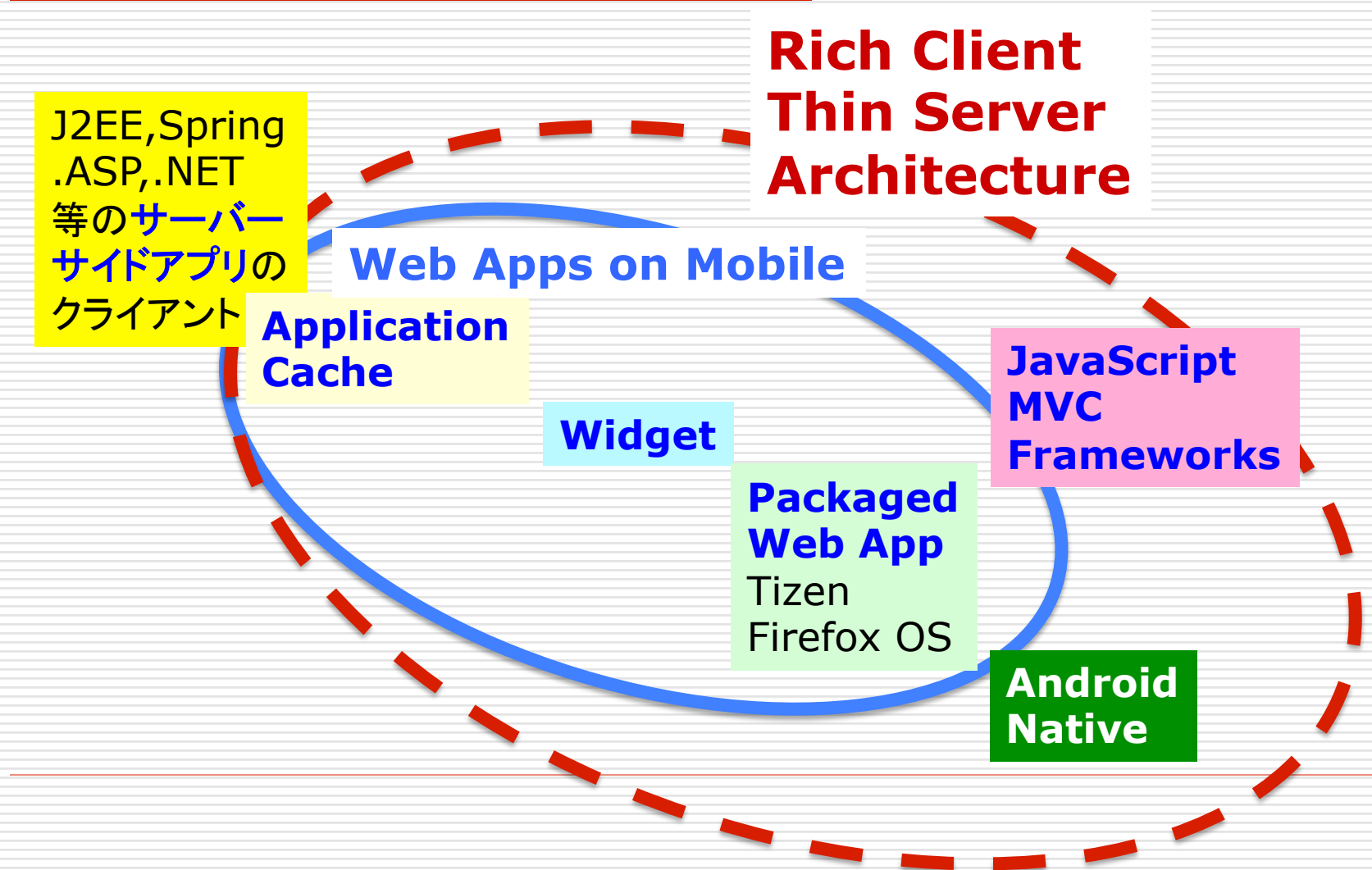
- Spine

- ...

新旧二つのタイプのWebアプリ



Web Apps on Mobile と Rich Client / Thin Server Architecture



Thin Server Architecture

- Thin Server Architecture というのは、2008年の初め頃に、Mario Valenteを中心とするグループが考えだしたコンセプト。
- 残念ながら、発表当時は、大きな反響を起こすこともなく、忘れられていた。今また、こうした数年前のコンセプトに照明があたりと言うのも、興味深いことである。

<https://sites.google.com/a/thinserverarchitecture.com/home/Home>

Thin Server Architectureの定義

Thin Server Architecture(TSA)は、今日のWebアプリケーション・フレームワークの、慢心と複雑さに対する、反応である。

TSAは、全てのプレゼンテーション層のロジックを、サーバーから、クライアント(Webブラウザ)上の、JavaScript(あるいは他)のロジックに移し変えることを提案する。

これによって、次の三つのポジティブなことが得られる。

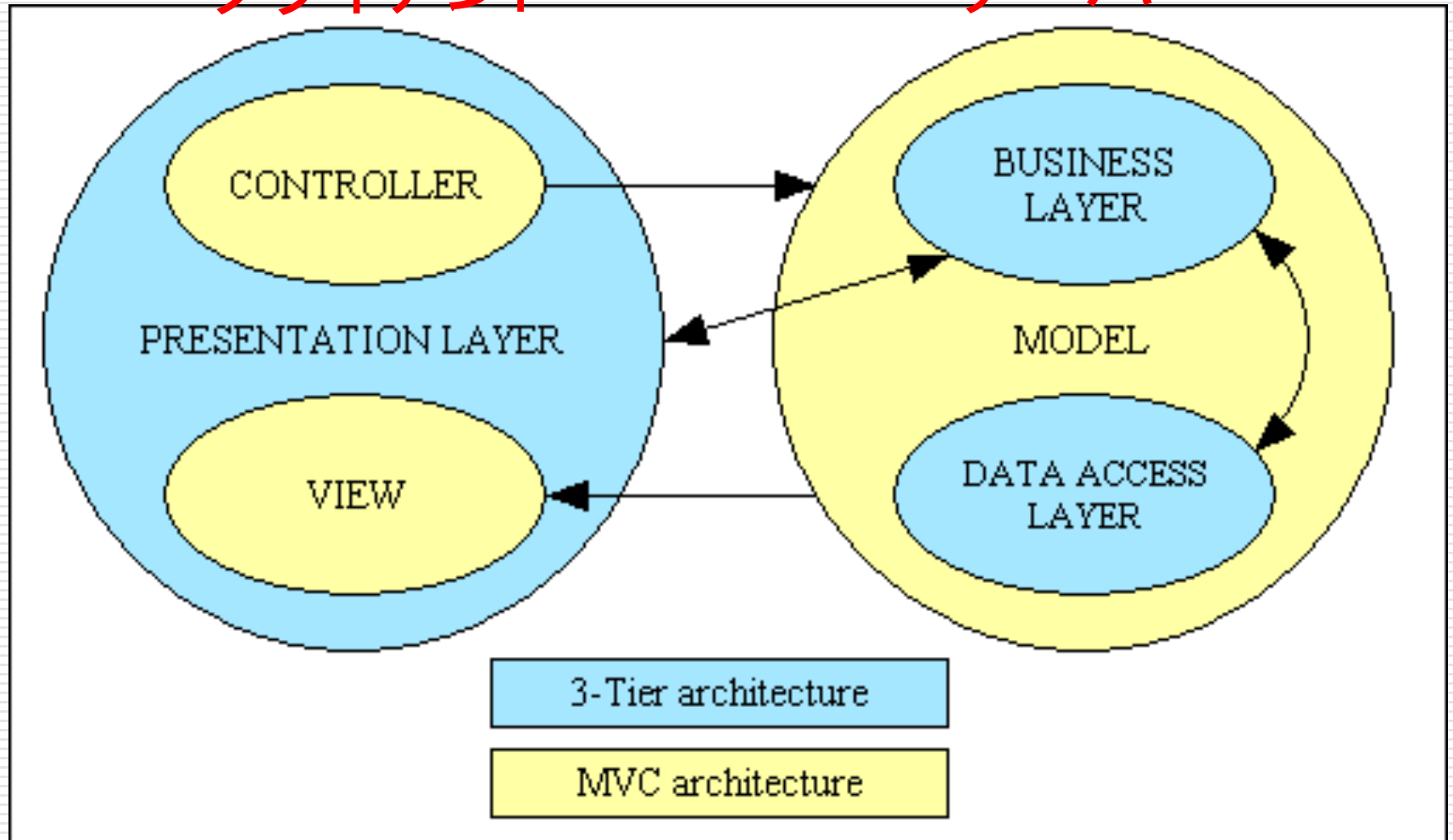
Thin Server Architectureの定義

1. サーバー側の開発者は、ビジネス・ロジックに集中出来る。
2. クライアントが分離して開発されるので、アプリケーションは、より複雑でなくなる。
3. サーバーとクライアントの通信は、データを他方の、あるいは将来のシステムとの間のデータのインポート、エクスポート、あるいは表現に利用可能な、プロトコルを利用する。

Thin Server Architectureの図式

クライアント

サーバー



Androidと Webアプリ / HTML5開発

AndroidでのWebアプリ開発の遅れ

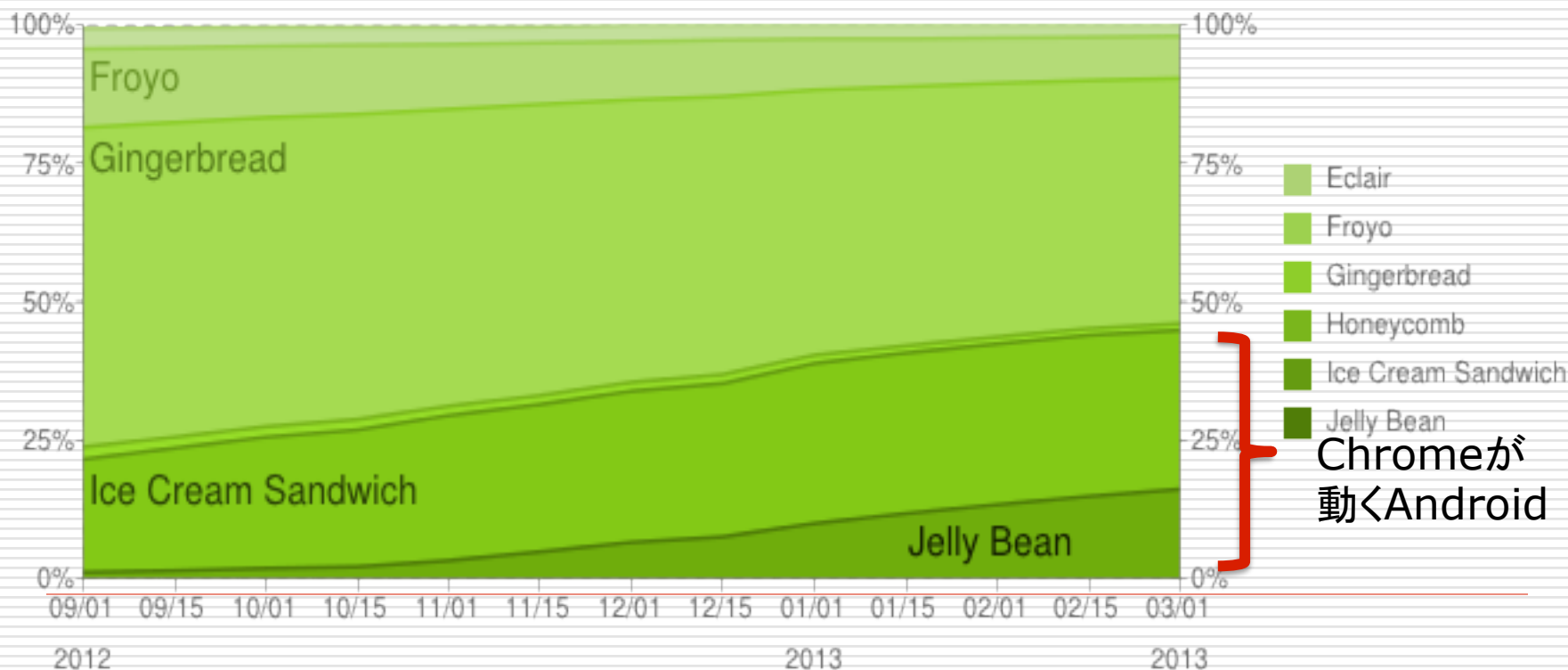
- ❑ 残念ながら、AndroidでのWebアプリ / HTML5開発は、Android標準搭載のWebブラウザのHTML5対応の遅れにより、遅れていた。
 - ❑ この状況は、昨年来のAndroidへのChrome搭載開始によって、大きく変わりつつある。
 - ❑ Googleは、AndroidへのChrome搭載に、全力で取り組み始めたように見える。
 - ❑ NEXUS 7では、初めてChromeがAndroidの標準ブラウザになった
-

AndroidへのChrome搭載

- 2012年6月27日 Google IOでのアナウンス
同日 Chrome 18 for Android ダウンロード開始
 - 2013年1月12日 Chrome beta for Android
 - 2013年3月1日 Chrome 25 for Android
 - 現在のPC版Chromeの最新バージョンは、26である。Googleは、PC版・Android版のChromeの同一バージョン同時リリースを、今年度中に実現するとしている。
-

Chromeが動くAndroidの割合

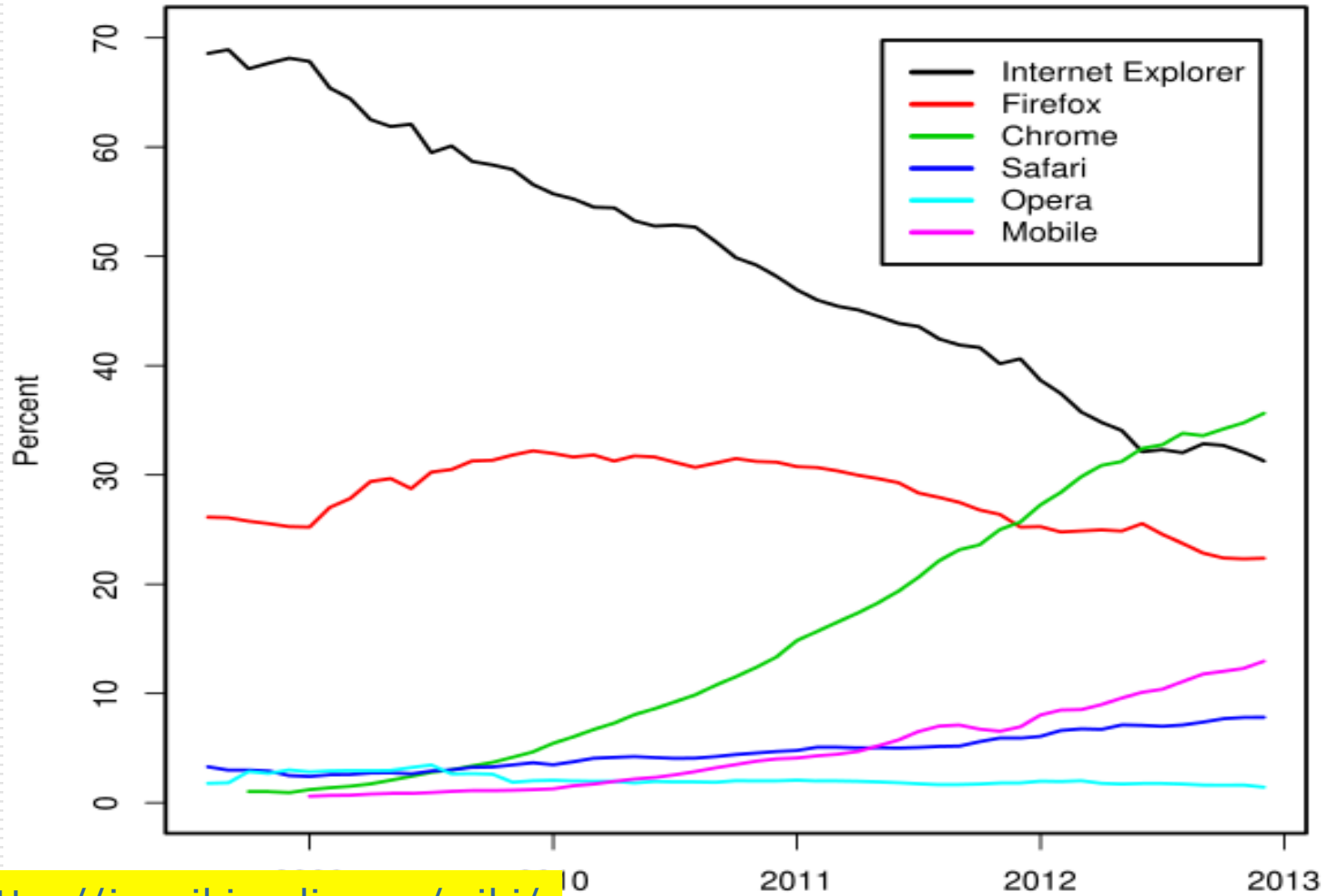
残念ながら、Chrome for Androidは、Androidのバージョン4.0以上でしか動かない。ただ、Chromeが動くAndroidの割合は5割に近づき、この一年半で倍増している。



Googleの Webアプリ / HTML5の取り組み

- AndroidでのWebアプリ / HTML5の遅れは、Googleがこの分野で立ち後れていることを、少しも意味しない。
- その中心は、Chrome+Chrome OSの開発チームである。Chromeは、短期間でシェアを大きく拡大し、HTML5対応では最先端を走り、そのJavaScriptエンジンであるV8は、もっとも高速なエンジンの一つである。
- Chrome OSは、商業的には成功したとは言えないが、そのコンセプトは先駆的で、現在の実装は、Firefox OSよりはるかに進んでいる。

ブラウザのシェア

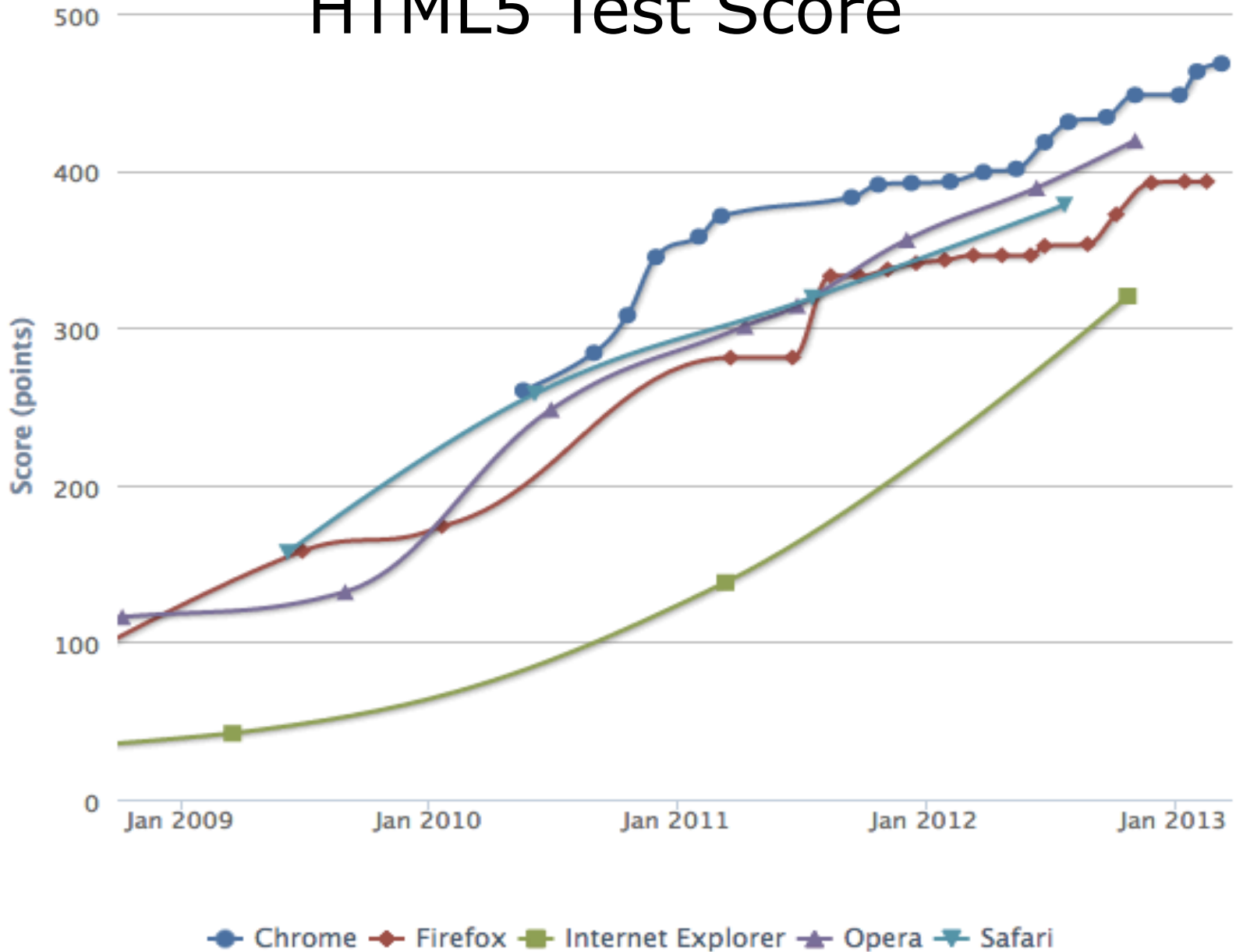


<http://ja.wikipedia.org/wiki/ウェブブラウザの利用シェア>

Year
Source: StatCounter

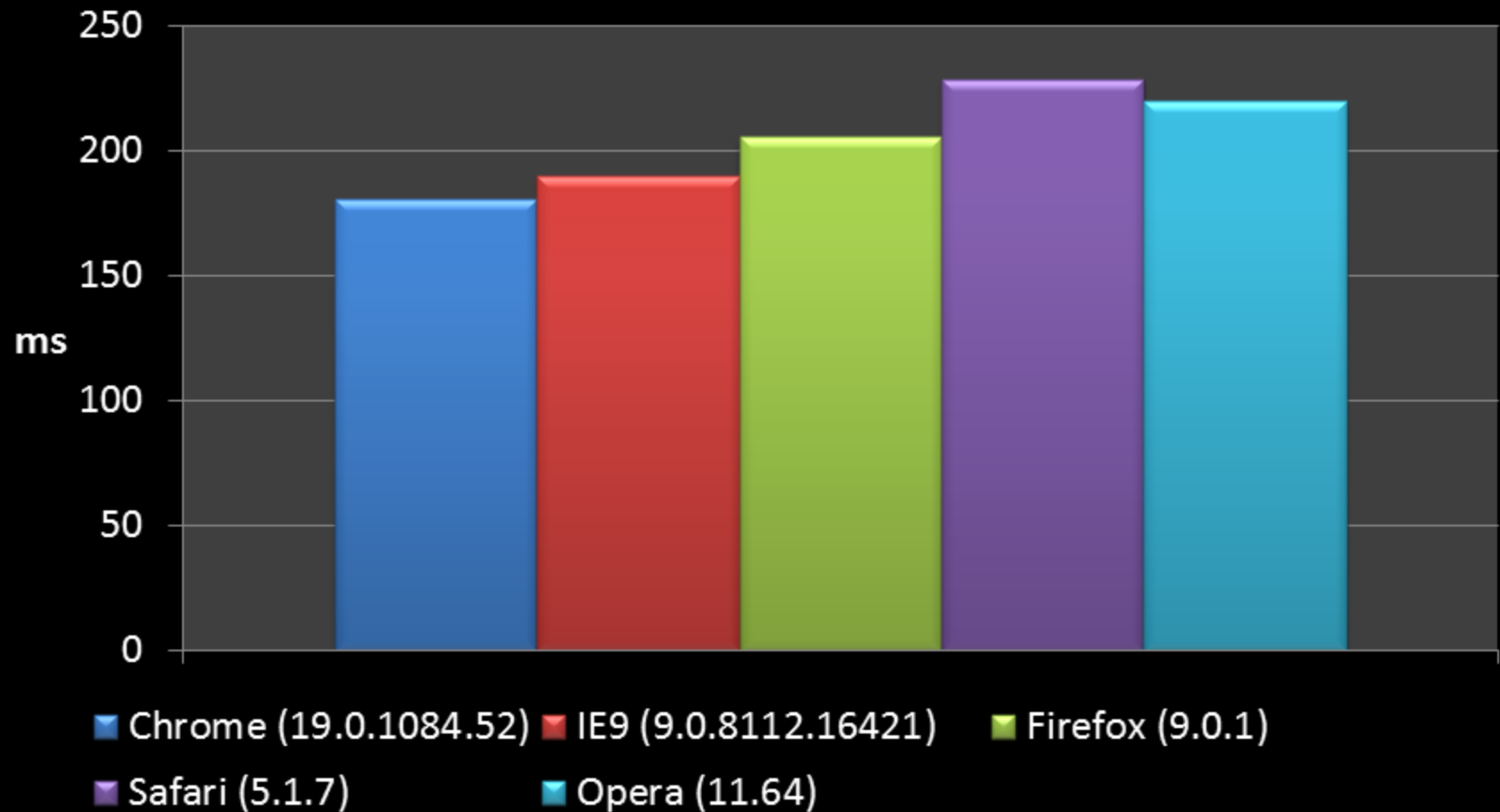
調査によって大幅に異なる

HTML5 Test Score



<http://html5test.com/results/desktop.html>

SunSpider JavaScript Benchmark 0.9.1 (Win7 Ult SP1)



<http://codehenge.net/blog/2012/06/javascript-engine-benchmarks-2012-edition/>

Chrome.* APIs

alarms
bookmarks
browserAction
browsingData
contentSettings
contextMenus
cookies
debugger
declarativeWebRequest
downloads
events
extension
fileBrowserHandler
fontSettings
history
l18n
idle
input.ime
management
omnibox
pageAction
pageCapture

permissions
privacy
proxy
runtime
scriptBadge
storage
tabs
topSites
tts
ttsEngine
types
webNavigation
webRequest
webstore
windows

Chrome external APIs

experimental.app
experimental.bluetooth
experimental.devtools.audits
experimental.devtools.console
experimental.discovery
experimental.identity
experimental.infobars
experimental.keybinding
experimental.mediaGalleries
experimental.offscreenTabs
experimental.processes
experimental.serial
experimental.socket
experimental.speechInput
experimental.usb

HTML5 API以外の
Chrome OSでのAPI拡張

Firefox OS WebAPIs

Alarm
AudioPolicy
Browser
Contacts
DesktopNotification
DeviceStorage
FM Radio
GeoLocation
Storage
SystemXHR
TCP Socket
mozCameras
mozMobileConnection
mozMobileNetworkInfo

HTML5 API以外の
Firefox OSでのAPI拡張

Firefox Marketplace APIs

Submission
Payment
Marketplace

Google **Dart**

– 新しいWeb言語の提案

Dartは、Googleによる、JavaScriptに代わる新しいWeb言語の提唱である。

Dart brings structure to web app engineering with a new language, libraries, and tools.

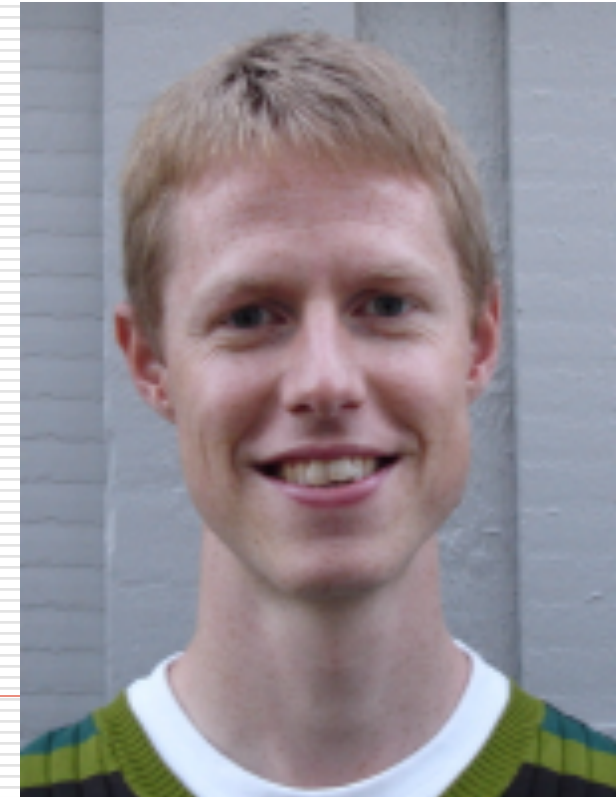
<http://www.dartlang.org>

Dart

□ **Lars Bak**
Kasper Lund
Seth Ladd
Eli Brandt

...

□ 2011年10月
GOTO Conferenceで発表





DART

Dart brings structure to web app engineering with a new language, libraries, and tools. [Read about our new M1 release!](#)

 Download now

 Circle +Dart

 Subscribe

 Follow @dart_lang

Introducing Dart

共有  詳細情報



✓ FAMILIAR

Dart is a **class-based, object-oriented language** with lexical scoping, closures, and optional static typing. Dart helps you build

⚙️ PRODUCTIVE

Dart Editor and the SDK provide an **integrated development and debugging** experience. The editor supports refactoring,

🌐 UBIQUITOUS

Dart can be **compiled to JavaScript**, so you can use it for web apps in all modern desktop and mobile browsers. Our

Dart Presentations

Watch videos and review slides from presentations about Dart. You can also watch episodes of [Dartisans](#), our Dart videocast.

Together at last!



Productivity or Performance. Pick Two.

by Seth Ladd

from Devovx 2012, 2012-11-14

 Slides



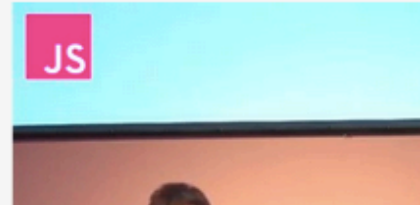
Breaking Open: Dart

by Seth Ladd

from Breaking Open, 2012-11-07

 Watch

A Mixin is the Delta between
a Class and its Superclass



Dart-to-JavaScript compiler goals

- Support Dart apps on all modern browsers
 - Tested on Chrome, Firefox, IE, and Safari
 - Ensures that the use of the Dart VM is optional
- Generate efficient and compact JavaScript

<http://www.dartlang.org/slides/>

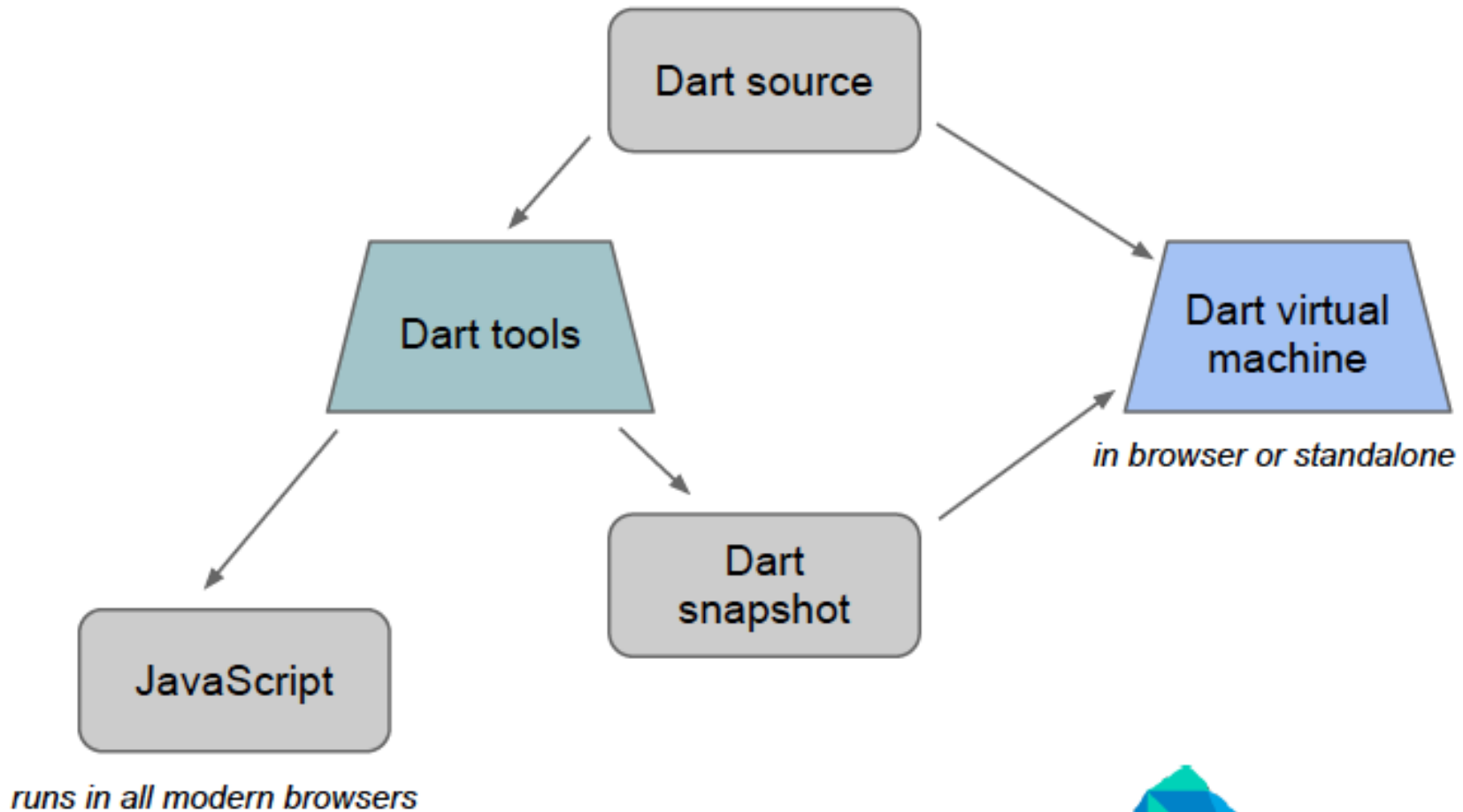
Dartの問題提起

-- JavaScriptはScaleしない

Dartは、Googleの提案する、**新しいWeb言語**である。ブラウザを中心として対象とする領域は、JavaScriptと重なっている。基本的には独自のVM上で動くのだが、JavaScriptに変換するコンパイラーを持ち、JavaScriptを強く意識している。

なぜ、GoogleはJavaScriptに代わるWeb言語を提案しているのだろうか？

Deployment and execution



DART

何故、Webはプログラムが難しいのか

- ❑ 大規模の、きちんと動くアプリを書くのは、大変難しい。
- ❑ プログラムの構造を推察するのが難しい
- ❑ 立ち上がりのパフォーマンスが、しばしば、非常に悪い。
- ❑ (型がないことによる)意図を文書化することの困難
- ❑ modules, packages, librariesがサポートされていない。

Dart: Modern Web Language
Kasper Lund

<http://www.infoq.com/presentations/Dart-A-Modern-Web-Language>

それでは、何故、 我々は、Dartを作ったのか？

- 我々は、Webプラットフォームを改善したい
 - 大規模なプログラミングの、より良いサポート
 - とりわけ、モバイルでの迅速なアプリの立ち上げ
 - より予測可能で、より良い実行時のパフォーマンス
 - JavaScriptは強力なツールではあるが、危険なところも持っている
 - イノベーションの勢いを維持する
 - Webは、ファンタスティックなペースで進化し続けている
 - 開発者のツールは、水準を上げなければならない
-

JavaScriptは、「驚き」に満ちている

- 沢山の沢山の、暗黙の型変換
- 間違った、あるいは、初期化しないままで値を渡したとき、ほとんどの関数は、理解可能な仕方
方で失敗するのではなく、とんでもない結果を返してくる



Keep on truckin'

Googleの選択

JavaScriptとDart

- Googleは、JavaScriptのコミュニティにも依然として積極的なコミットを続けながら、Dartの開発・普及を強力に押し進めている。
- MicrosoftのTypeScriptは、ECMA6互換という旗印を掲げて、JavaScript陣営の中にとどまり、JavaScriptの発展を目指すというスタンスを堅持している。
- それに対してDartは、JavaScriptのコードも生成出来るのだが、基本は、独自のVM上で動く独自言語で、JavaScriptの置き換えを目指しているし、それを隠さない。

JavaScriptとDart

- ❑ こうしたこともあって、JavaScript開発者の間では、Dartは必ずしも好意的には受け止められていないようにも見える。
 - ❑ ただ、もしそこに、現在のJavaScriptで十分だと言う意識があるのなら、Webアプリの大規模化が進む中でそれは正しくないと感じている。
 - ❑ JavaScriptでは、大規模なWebアプリの開発は困難であるというGoogleの認識は正しいように思う。TypeScriptを開発したMicrosoftも基本的には同じ認識だと思う。
-

DartとChrome

- Web言語であるDartにとって、ブラウザーは本質的に重要である。すでに、DartのVMを搭載したChromeであるDartiumが公開されている。
 - ただし、ブラウザーの競合各社に、Dartサポートの動きは全くない。
 - Googleは、標準のChromeブラウザーに、デフォルトでDart VMを搭載するだろう。
-

AndroidとChrome

- Googleの選択の中で、もっとも重要なのは、Android OSとChrome OSの関係を、どう付けていくのかという選択である。
- この一年、Googleは、AndroidへのChromeブラウザーの導入を中心に、懸命にAndroidでのHTML5対応の遅れをカバーしようとしてきた。
- Android, iOSのあとを追う「第三勢力」が、MicrosoftのWindows 8を含めて、JavaScript + HTML5の新しいWebアプリの世界に注力しているのは、そこに数少ない「勝機」があると判断しているからに他ならない。

AndroidとChrome

- Googleが進むべき道は、明確なように思う。
AndroidのChrome標準ブラウザー化を第一歩として、Chrome OSで培った、あたらしいWebアプリの世界でも第一級のものとして通用するJavaScript+HTML5のノウハウをAndroidに導入することである。
 - Googleが行うであろう次の一步は、Googleの標準ブラウザーへのDart VMの搭載になるであろう。
-

AndroidとChrome

- こうして、これまで蓄積されたAndroidのNativeのリソースを生かしながら、JavaScript+HTML5を中心とした新しいWebアプリへの対応も可能なHybridなシステムに、Androidは生まれ変わっていきだろう。
 - それは、Firefox OSやTizenの挑戦を迎え撃つに、十分のパワーを持ちうるものになると思う。
 - 要は、AndroidとChromeは統合されていくだろうという展望を僕は持っている。
-

Android Chief Andy Rubin Switching Roles At Google In Surprise Management Move

+ Comment Now + Follow Comments

Updates to add analyst comment.

Andy Rubin, head of Android at [Google](#), is leaving that job to take on another unspecified role at the company in a surprise management switch up, Google CEO Larry Page announced in a blog post today. Sundar Pichai, head of Chrome and Apps, is now in charge of Android.



Androidの次の飛躍を考える

Next Billion市場での飛躍

- Next BillionでのAndroidの拡大は、新たに、数億人規模の人々が、インターネットの世界に参加することを意味する。
 - いまだ、世界人口の半数以上の数十億人が、インターネットに接続していない中で、スマートフォンの普及・拡大は、そうした現実を変えていく最大の力になる。
 - それは、20世紀に進行したPCの普及とそれに続いたインターネットの爆発に匹敵する、歴史的な意味を持つものになる。
 - 低価格Androidが、そうした流れを牽引する。
-

新しいWebアプリの世界の拡大

- Androidとchromeとの統合は、新しいWebアプリの世界の拡大に、大きなインパクトを与えるであろう。
Androidは、新しいWebアプリの潮流の最大のプラットフォームになるのだから。
 - 新旧二つのWebアプリの世界の拡大は、オープンなWeb標準技術が、ほとんど全てのIT技術を包み込んでいく過程の進行である。
 - それは、おそらくこうした動きに続くであろう、全てのモノ、全てのヒトがネットワークにつながる世界を準備する。
ただし、重要なのは、モノのネットワークではなく、人間のネットワークである。
-

新しいWebアプリの世界の拡大

- Webアプリは、携帯と同様に、常にマシンの前に人間がいることを想定している。Webアプリは、そうした特徴を持つソフトウェアの形式である。ただし、全てのソフトウェアが人間を必要としている訳ではない。無人の世界に電話はいらないように、人間との相互作用のないところにWebアプリの必要はない。
-

日本のAndroidアプリ開発者の課題

- JavaScript+HTML5を中心とした新しいWebアプリの世界への準備をただちに始めること。Android上でもそうした環境整備は、今後、急速に進んでいく。この分野でも、Nativeアプリ同様、世界一を目指そう。
 - Stand Aloneのアプリだけではなく、クラウドのサービスと一体となったアプリ/サービスを構想すること。「コミュニケーションと情報共有」の領域では、まだまだ大きな可能性が残されている。
 - エンタープライズの分野でのデバイス利用の推進。
-

日本のAndroidアプリ開発者の課題

- 日本でのニーズが飽和しているように見えたら、グローバルな市場に目を向けること。そこでの成功は決して簡単ではないが、アプリの市場が爆発的に拡大しようとしているいま、チャンスはある。
-