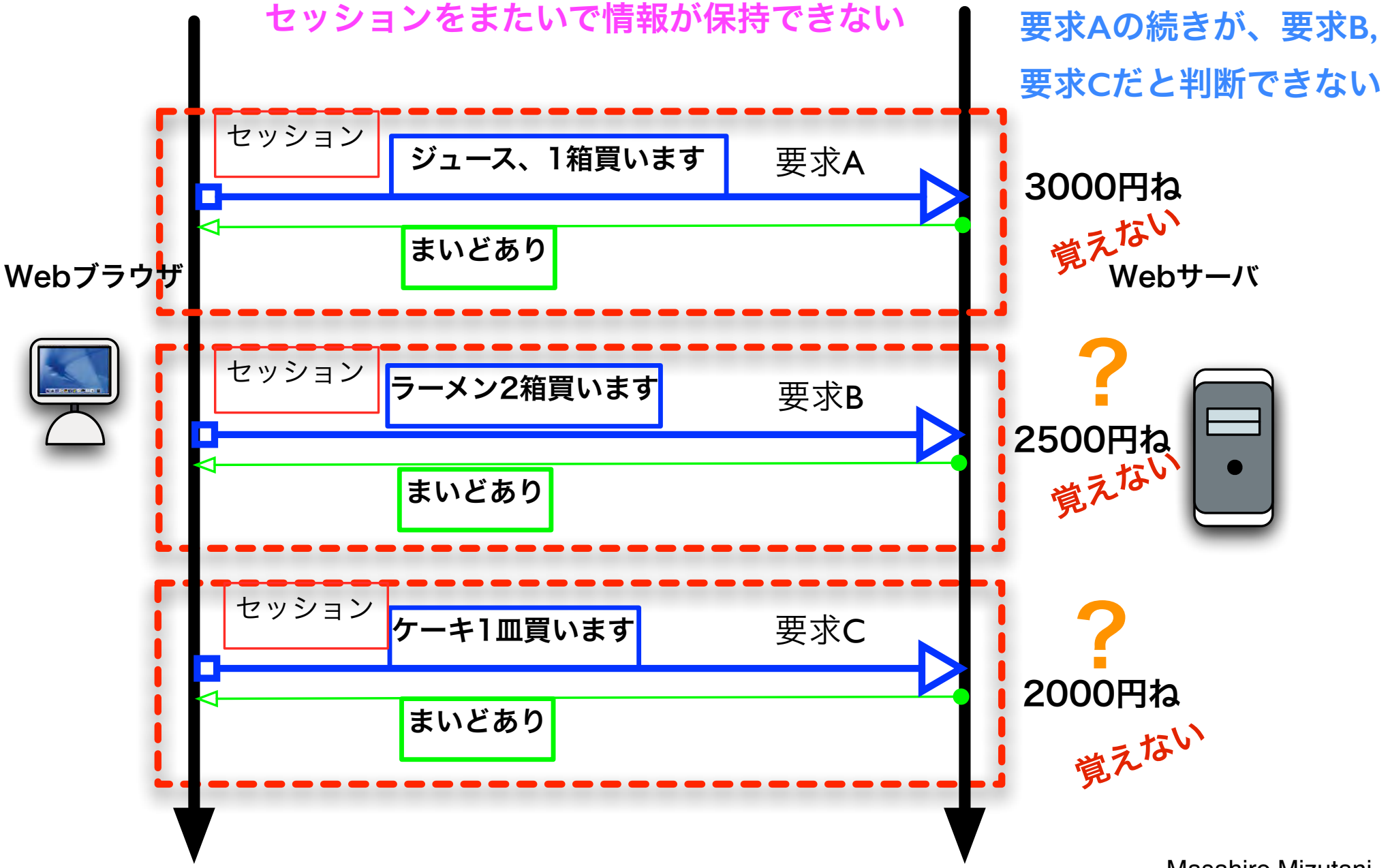
Cookieの周辺
と
Webアプリケーション

大東文化大学
水谷正大

HTTPの特徴と問題点

- 非常に単純
  - HTTP要求とその応答が１つのセッション
- ステートレス（stateless）
  - 次の通信は前のセッション結果と何ら関係を持たない
    - 処理結果は残さずに、その都度廃棄される
- トランザクション処理では工夫が必要
  - 状態を維持できないために、関連する複数の処理を一つの処理単位としてまとめることができず工夫が必要
    - ユーザ認証をした上でのページ移動
    - 複数選んだ商品の購入決算

ステートレスなHTTP通信

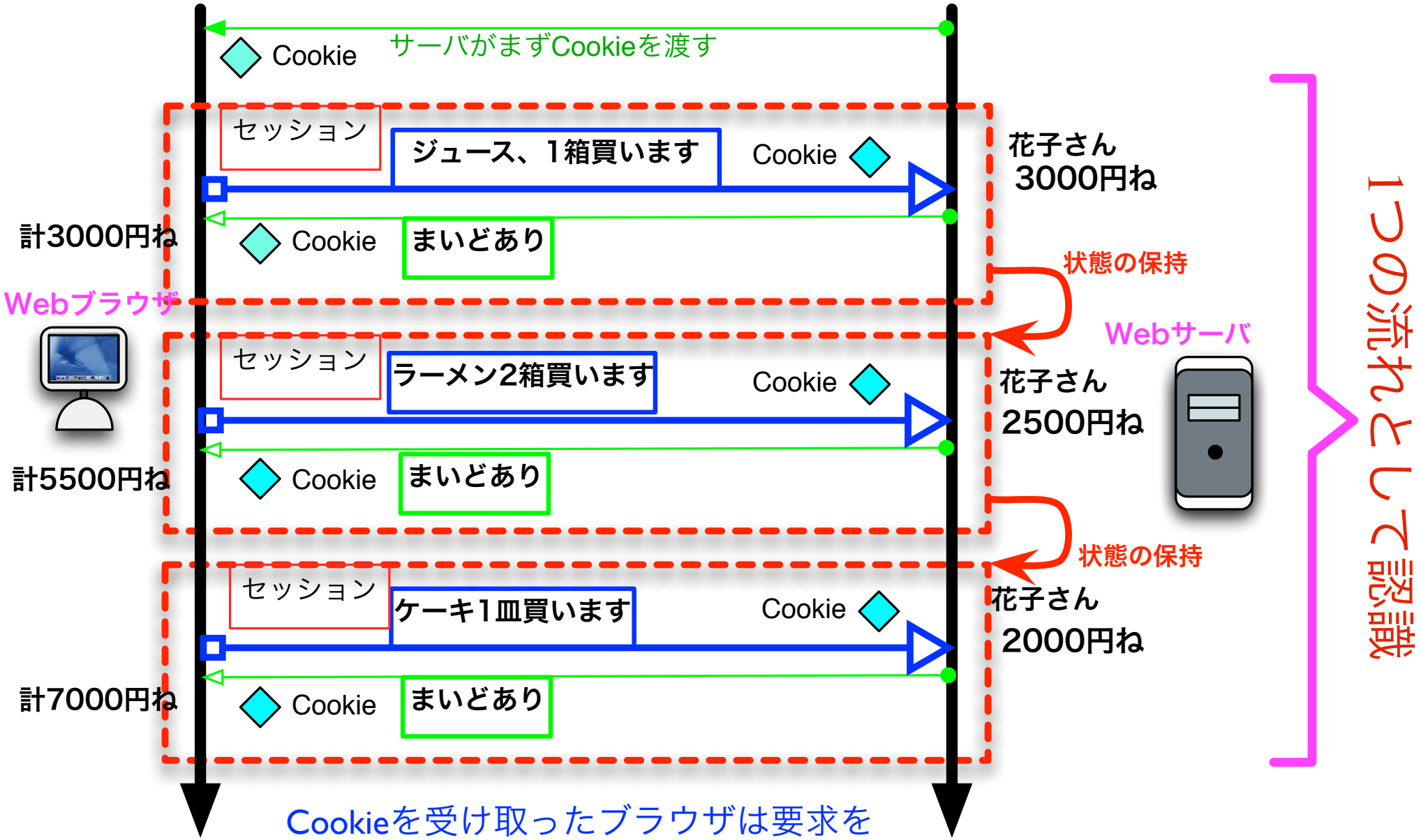


Masahiro Mizutani

クッキー（cookie）の利用

- トランザクション処理を可能
  - クッキーを使って状態を維持
- クッキーの仕組み (RFC2965/6265)
  - サーバからクッキーをブラウザに渡す
    - HTTPヘッダに埋め込むかJavascriptで利用
  - ユーザ情報をクッキーに書き込む
  - 同じURL（やサーバ）に接続するときはクッキー情報をサーバに送信
  - WebクライアントとWebサーバ間でお互いを認識し、継続的なデータ交換を可能にする

Cookieを使ったHTTP通信



Cookieを受け取ったブラウザは要求を
送信する際に、Cookieを毎回付加する

Masahiro Mizutani

Cookieを使う

1) Webサーバは応答ヘッダで指定したCookieをブラウザに渡す(Set-Cookie)

Set-Cookie: Customer=”Taro_Jirou”; domain=happy-shopping-town.com; path=/shopping/; expires=”25 Nov 2015 08:36:20 GMT”; secure

2) ブラウザが同じWeb サーバと通信する時に、Cookieをサーバに送信(Cookie)

Cookie: Customer=” Taro_Jirou”; PartItem=”Lemon_0987”,Shipping=”Post”;

3) サーバをCookieを受け取ると、サーバはDBに問い合わせてユーザを特定、ブラウザに渡す情報をカスタマイズして送信

以降は、2, 3（または1,2,3）を繰り返して通信する

Cookieの条件

- 有効期限（expires属性）内のCookieは保存
- Cookieの数は最大300個まで
- 1つのCookieは最大4KBまで
- 1つのサーバにつき、Cookieは最大20個まで

HTML を用いて Cookie の値を記録させることができる

```
<meta http-equiv="Set-Cookie" content="～">
```

～の部分に　*NAME*=値; expires=値; domain=値; path=値; secure

Cookieを表示する（1）

Internet Explorer

テキストファイルなのでエディタで開くだけ

Safari

(a) [環境設定]/[詳細]/メニューバーに”開発”メニューを表示

[開発]Webインスペクタを表示

[ストレージボタン]

該当ページのクッキーを表示

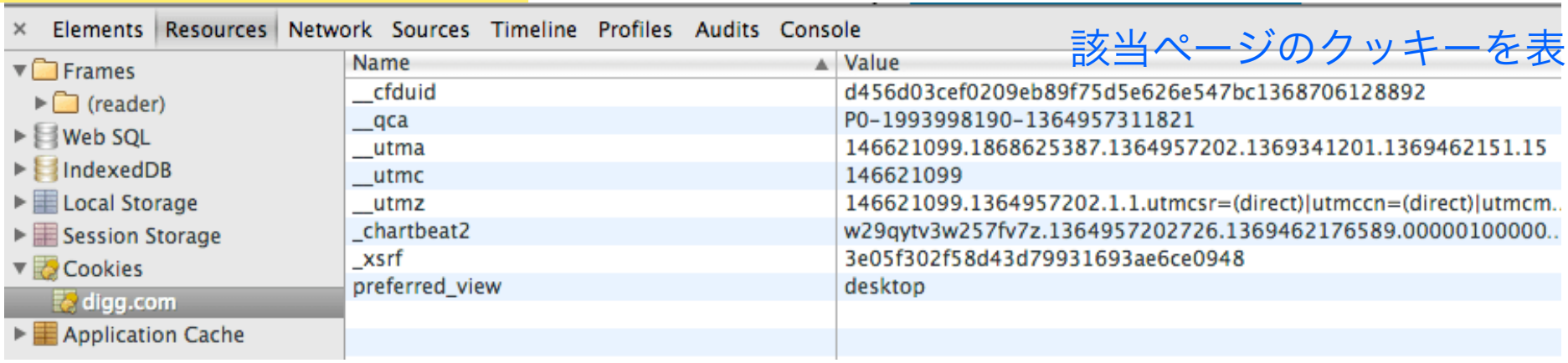
(b) Safari Cookieを使う(MacOS)　http://sweetpproductions.com/safaricookies/

クッキーを管理できて便利

Cookieを表示する（2）

Google Chrome

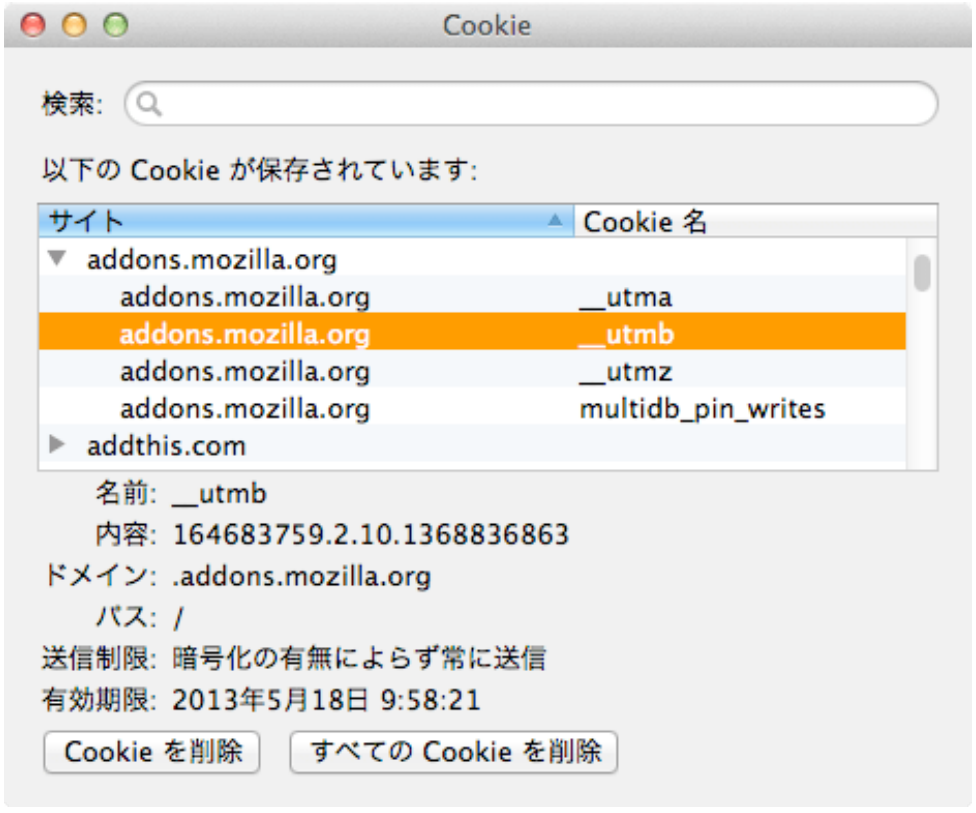
[表示]/[開発/管理]/デベロッパーツール



該当ページのクッキーを表示

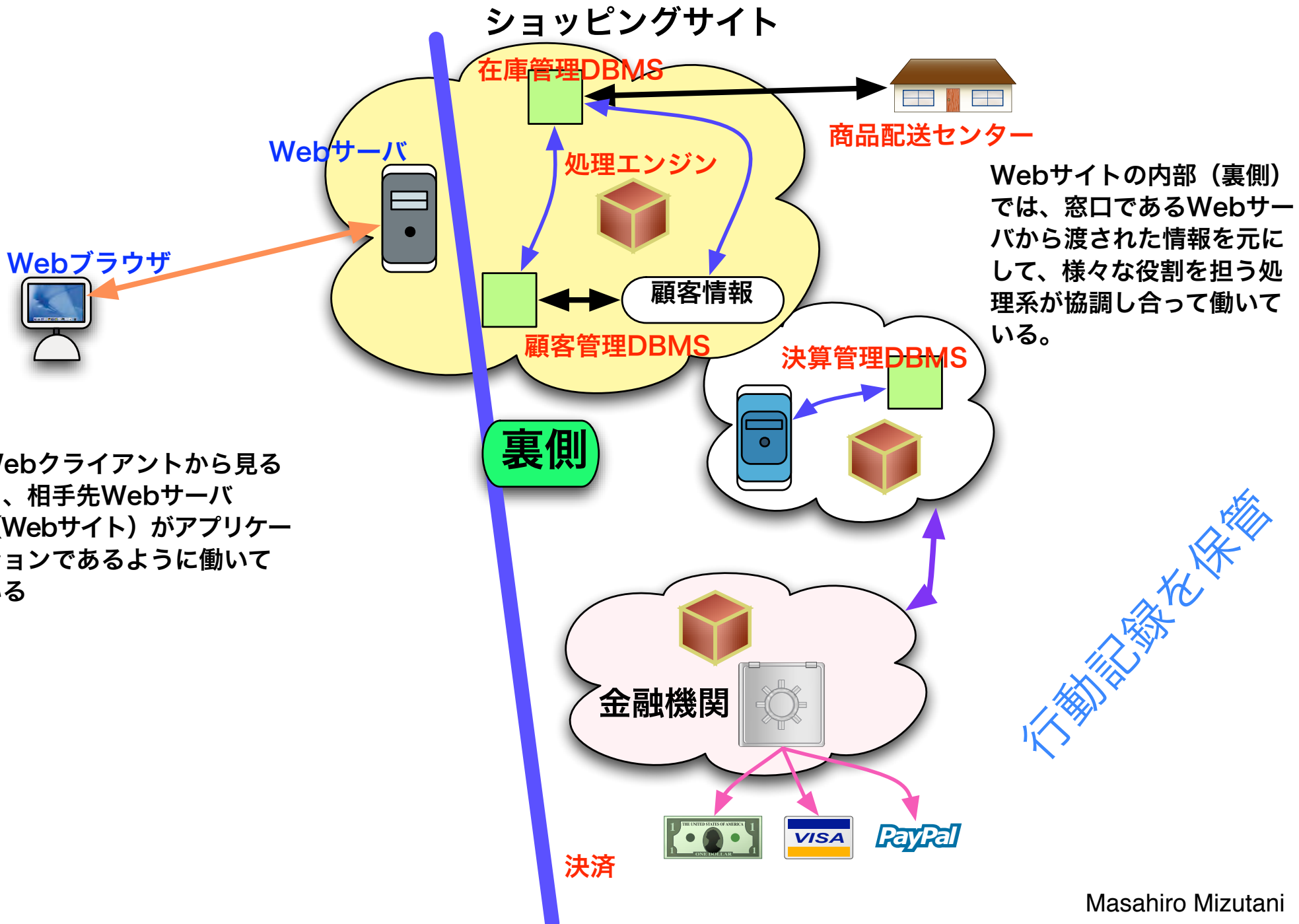
Firefox

(a) [環境設定]/[プライバシー]Cookieを個別に削除



(b) アドオン Firebug を使って編集

Webアプリケーション



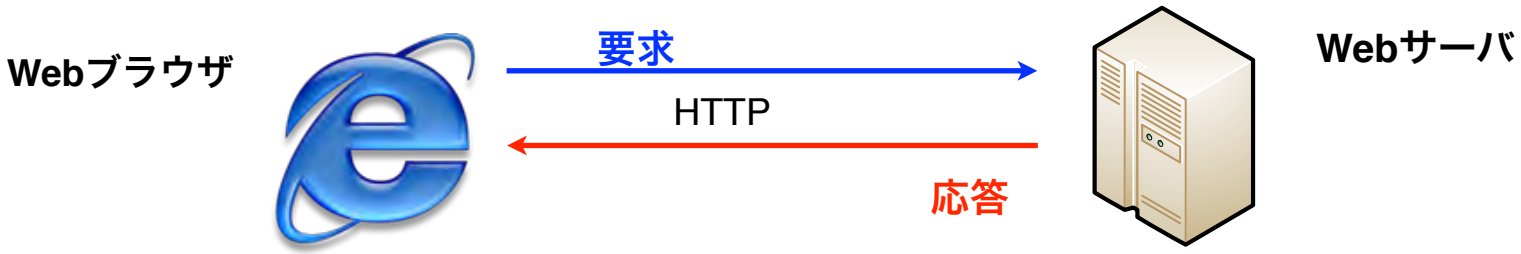
Webサイトの内部（裏側）では、窓口であるWebサーバから渡された情報を元にして、様々な役割を担う処理系が協調し合って働いている。

Webクライアントから見ると、相手先Webサーバ（Webサイト）がアプリケーションであるように働いている

行動記録を保管

Masahiro Mizutani

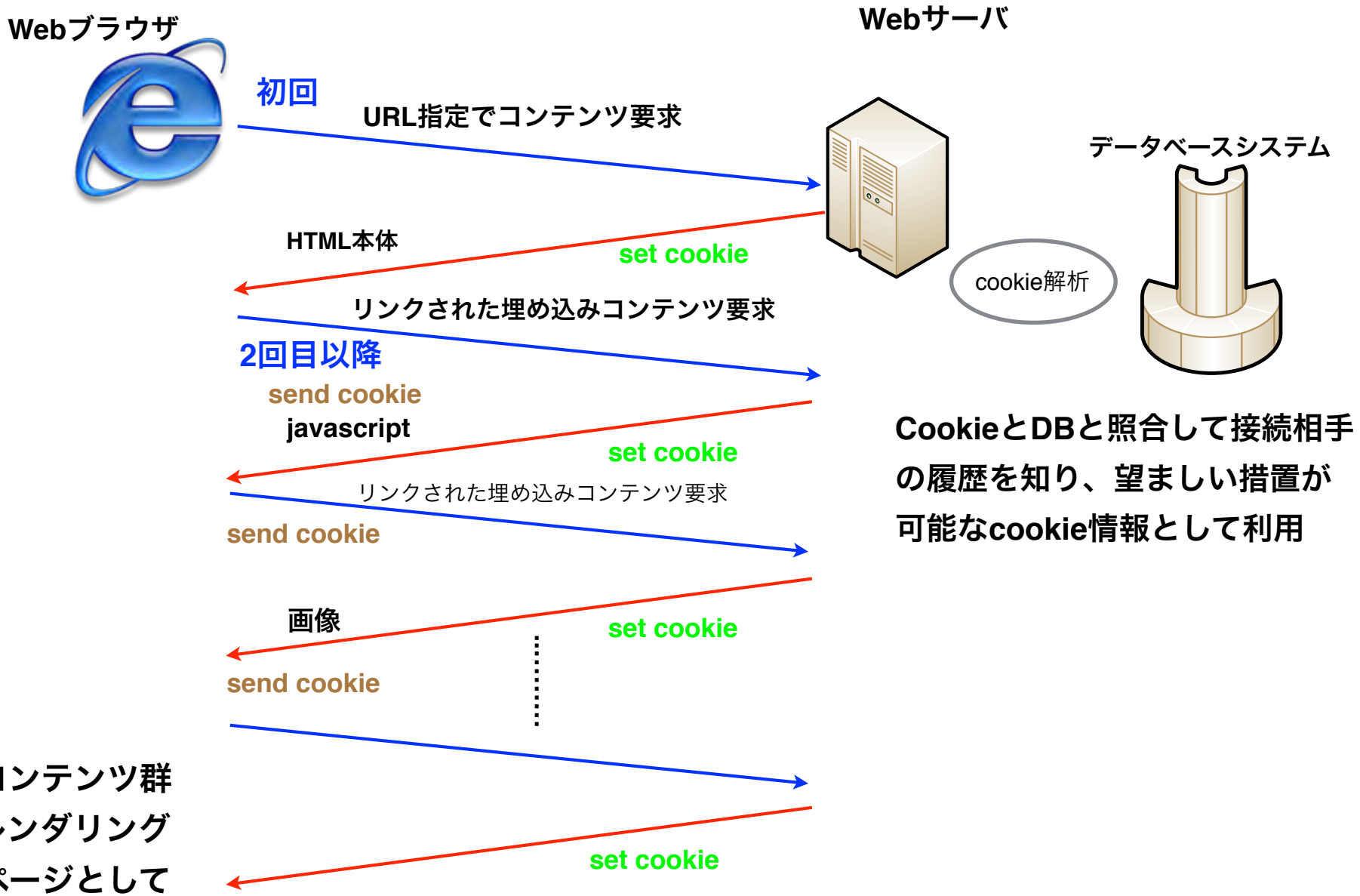
HTTPの実際



WebブラウザとWebサーバ間の通信なんだけど。。。

Webページが表示されるまで

Webサーバ間では複数のHTTP通信が発生



Cookieの送受信の発生

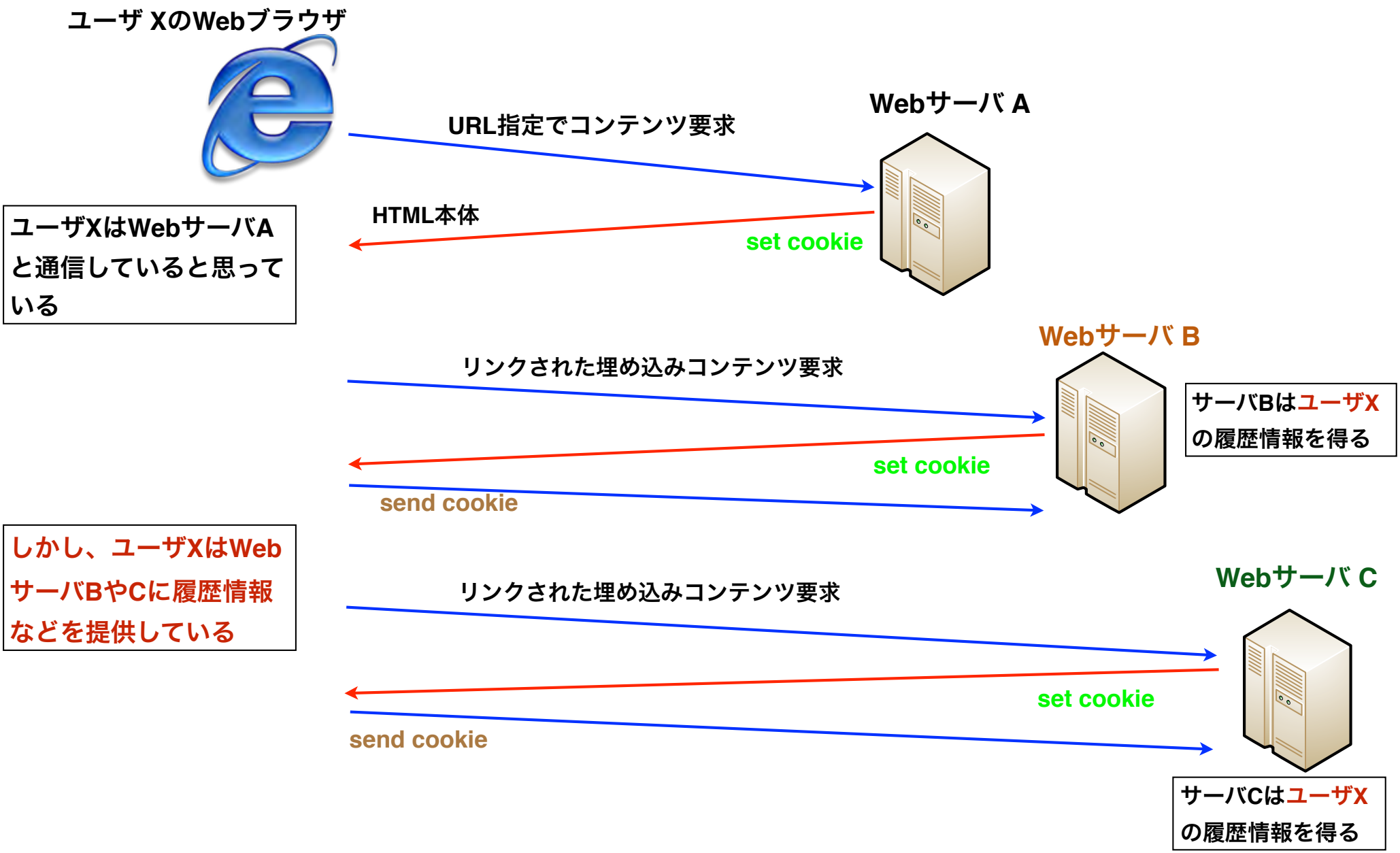
1. Cookieの受け入れ処理は
   Webブラウザ側の責任

Webブラウザ Safariの[プライバシー設定]

2. HTTP要求があればWebサーバはブラウザへの**set cookie権利**を持つ

3. ブラウザ内のCookie内容は以降のHTTP要求時に当該Webサーバに送信

こんなHTTP通信も可能（１）



Weサイト側の不手際による危険性

Cookieが盗まれたら

成りすまし

サーバーはクッキーの内容を見てクライアントを判別するだけで、送信されてきたクッキーが盗まれたものであるか否かは判断できない

別サイトで発行されたCookieは受け取らない

セッションフィクセーション

取得したセッションIDをセットして新規ユーザー登録画面へ誘導する罠をしかける

クロスサイト・リクエスト・フォージェリ（CSRF）

カートへの追加リクエストを対象としたCSRF攻撃を受けると、セッション変数が上書きされてしまい、結果としてカートの中身が書き換えられて意図しない商品を購入させられてしまう

Cookieは危険なのか？（1）

決められたサーバーや指定のディレクトリにあるHTMLファイルを要求したときにのみ、Cookieをサーバーへ送信

ユーザが管理方法を間違えない限り（パソコンを盗まれるなど）、Cookieは外部へは出ない

Cookieファイルはテキストファイルなので、すぐさまプログラムとして動作させることは不可能（ウイルスでない）

Webサーバーのソフトの欠陥 または管理者によるCookieの設定ミスによるクロスサイトスクリプティングなどの脆弱性があり得る

Cookieは危険なのか？（2）

Cookieを利用した個人の行動監視　トラッキング

元々、Web サーバはどのIPアドレスからどのページが何回呼び出されたか把握・記録している（Webの仕組み）。誰がどのページを見たのかを知るためにCookieを使って識別番号を誰がどう閲覧したかは分かるようになる。

さらにユーザー登録という形で付加サービスを利用するときに、実名や住所を入力すると、利用者が完全に特定できる

トラッキングを利用したターゲット広告

広告にCookieを埋め込むことで、広告企業は誰がどのサイトでどの広告を見たのか把握することができる。広告マーケティングにCookieが利用されている（この是非は議論すべきだ）。

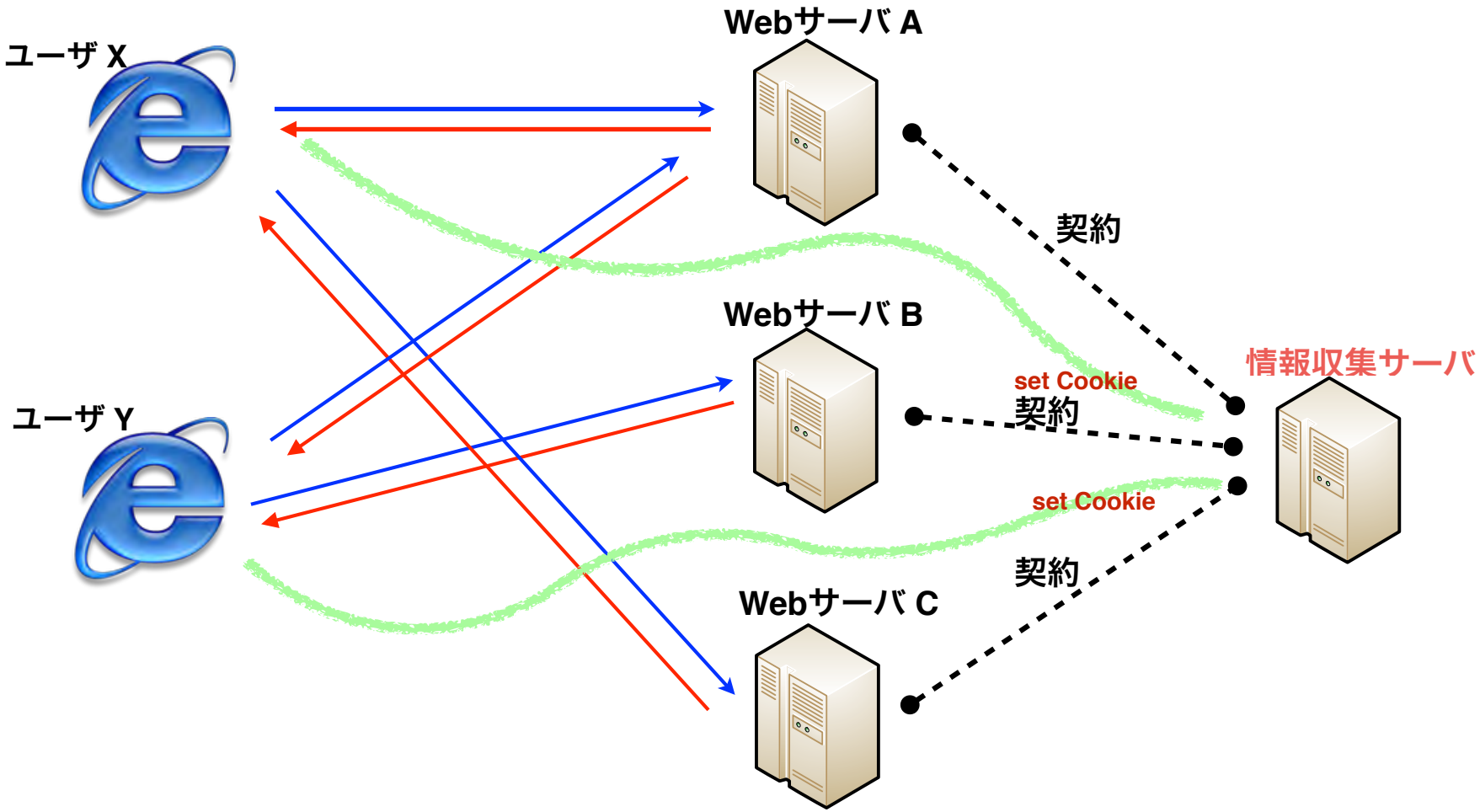
ただし、現在閲覧しているページ以外の第三者サーバーがCookieを要求しても送信しないようにできるWebブラウザ機能がある

こんなHTTP通信も可能（2）

1) サーバA, B, C…での各コンテンツには情報収集サーバへのHTTP通信を発生させるリンクを埋め込んであり、情報収集サーバはset Cookieの権利を得る。

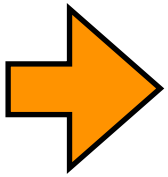
2) 情報収集サーバーはユーザのサーバA, B, C…へのアクセス情報を集めることができる。

3) 情報収集サーバーは膨大なユーザ履歴を分析し、その結果を契約サーバに開示できる



Cookieの課題

- セッション・ハイジャック
  - 第三者にセッションIDを読み取られる
- トラッキング・クッキー
  - ユーザのアクセス履歴の追跡
  - Web広告業者、マーケティング
    - 第三者への情報開示



Cookieの信頼性=Cookieの管理者の信頼

サイト内容が信頼できないとき

はCookieも信頼すべきでない

Cookieファイルは定期的にチェック

ユーザの知らない間にクッキーが使われている

- Webサイト上の**行動履歴**
  - プライバシの問題も生じ得る
  - 信頼できないサイトによるCookieの悪用
- Webブラウザは最新版を使う
- 定期的にCookieファイルのチェックと削除
  - 各種ブラウザを使い分けて、その設定や特長を把握
  - Windows：IEの[ツール][インターネットオプション][全般]から「Cookieの削除」

Cookieにさえ気をつければ
行動履歴の追跡から安全か？

残念ながらNo

高度な手段が開発されている

Webトラッカー情報を知る

Webブラウザ拡張機能

https://www.ghostery.com

閲覧しているページのトラッカー情報を提供
ユーザー自身が判断を下す

アプリケーションの変遷

- デスクトップアプリケーション
  - ローカル端末内にインストール
- Webアプリケーション
  - 同等な処理をWebを入り口にして実現
  - ストレージサービス
    - Dropbox
  - 文書サービス
    - Google Drive/OneDrive
  - バンキング

Webアプリの利点

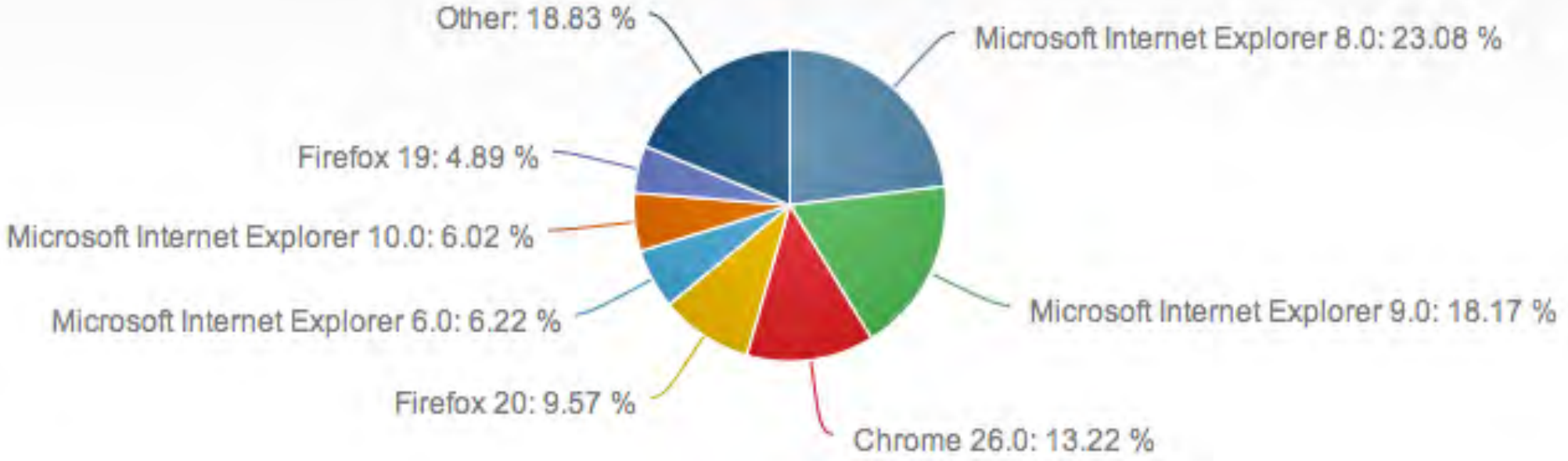
- 端末アプリの管理から開放される
  - 端末側にはWebブラウザだけがあればよい
  - Webサイトの内側でなされている複雑な処理はマスクされてユーザからは見えない
- 端末側の負担を小さくできる
  - 要求ハードウエア仕様が低い
- Webサービスの充実
  - サービスを提供するWebサイトにアクセスできれば多様なサービスを享受できる

クラウドの問題点

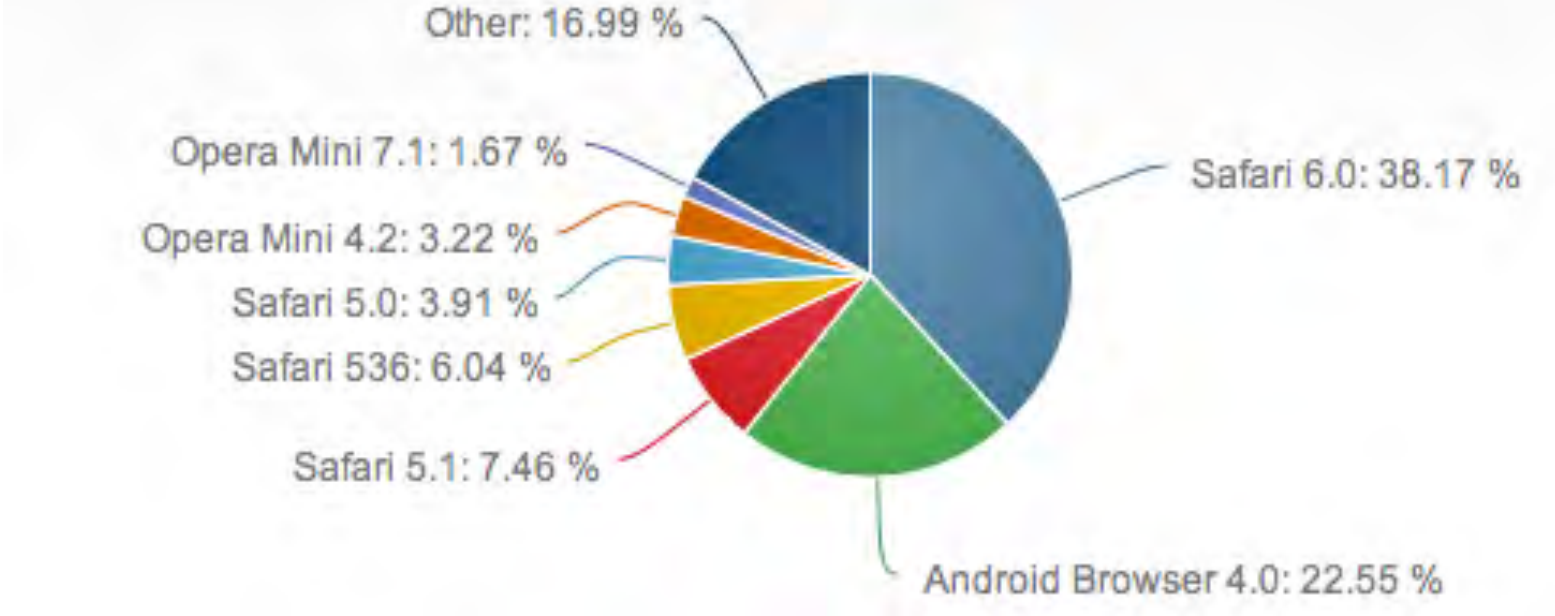
- ユーザの都合でサービス内容を変更できない
- クラウド提供側の機器障害、倒産などの事由によりサービスが停止する
- クラウド側に全データが集中している情報管理上の問題
  - 情報流出のリスク
    - 攻撃対象になりやすい
    - クラウドの破壊や政治的利用の影響が甚大

http://www.netmarketshare.com/browser-market-share.aspx

パソコンWebブラウザシェア 2013年4月

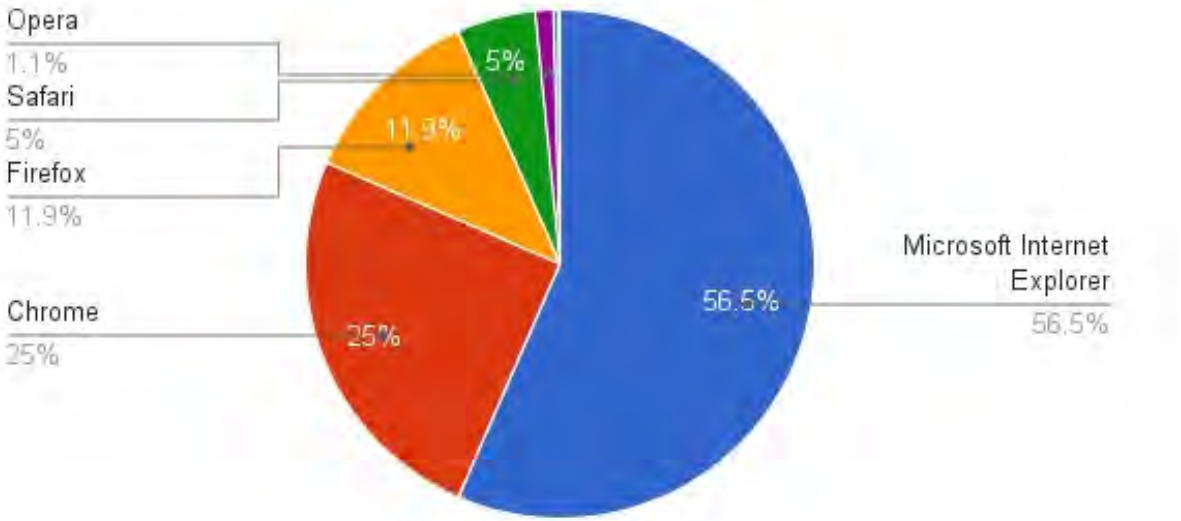


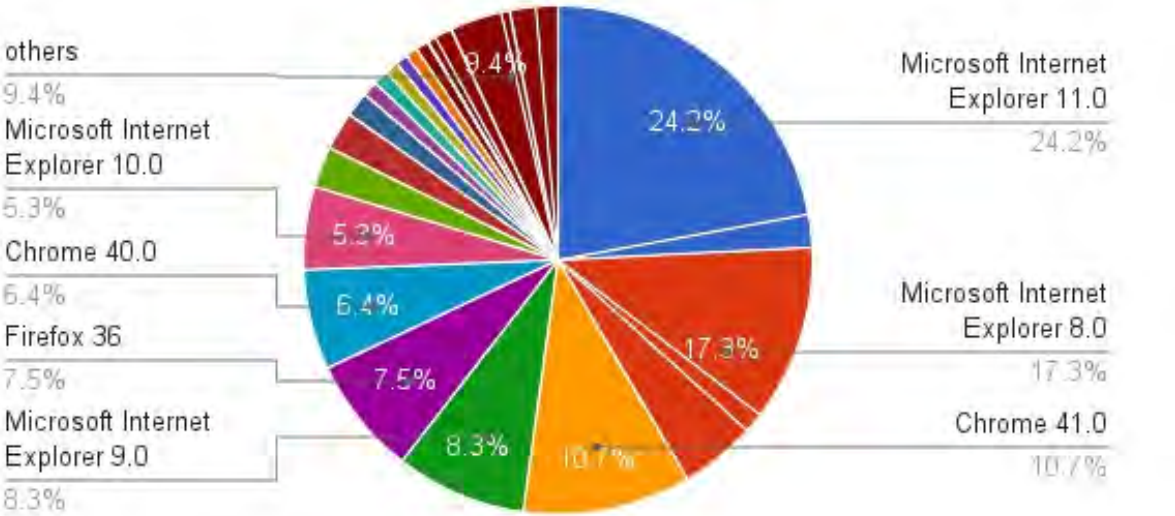
Mobile/Tablet Webブラウザシェア 2013年4月



http://marketshare.hitslink.com

Net Applicationsから2015年3月のデスクトップ・ブラウザのシェア





Microsoft Internet Explorerがシェアを落とし、それ以外の主要ブラウザ(Chrome、Firefox、Safari、Opera)すべてがシェアを増やした