SPDYによるトラフィック変化

ミクシィ 吉野純平

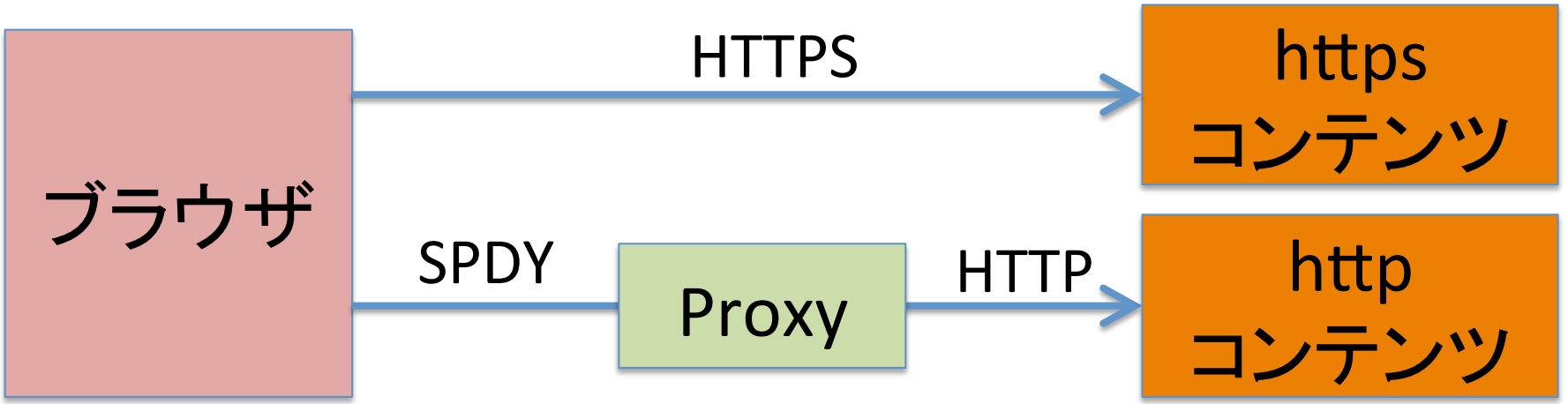
過去の議論のおさらい（1）

- **「Outbound Port 80 blockingのご提案」**
  - JANOG 31
  - Firesheepという無線LANからsessionを盗むアプリ
  - 大手サイトのHTTPS化
    - wildcard証明書の登場
    - SPDYやHTTP/2.0の登場でTLS通信が重要に
  - httpsの利用をするためのレスポンスヘッダ紹介

過去の議論のおさらい（2）

- 「**HTTP 2.0のインパクト**」
  - JANOG 32
  - プロトコルの解説や標準化動向
  - CGNからの視点で
    - 使えるポート数が少なくても幸せ
  - Akamai様から見たSPDYの対応状況
    - 30%程度がSPDY対応ブラウザ

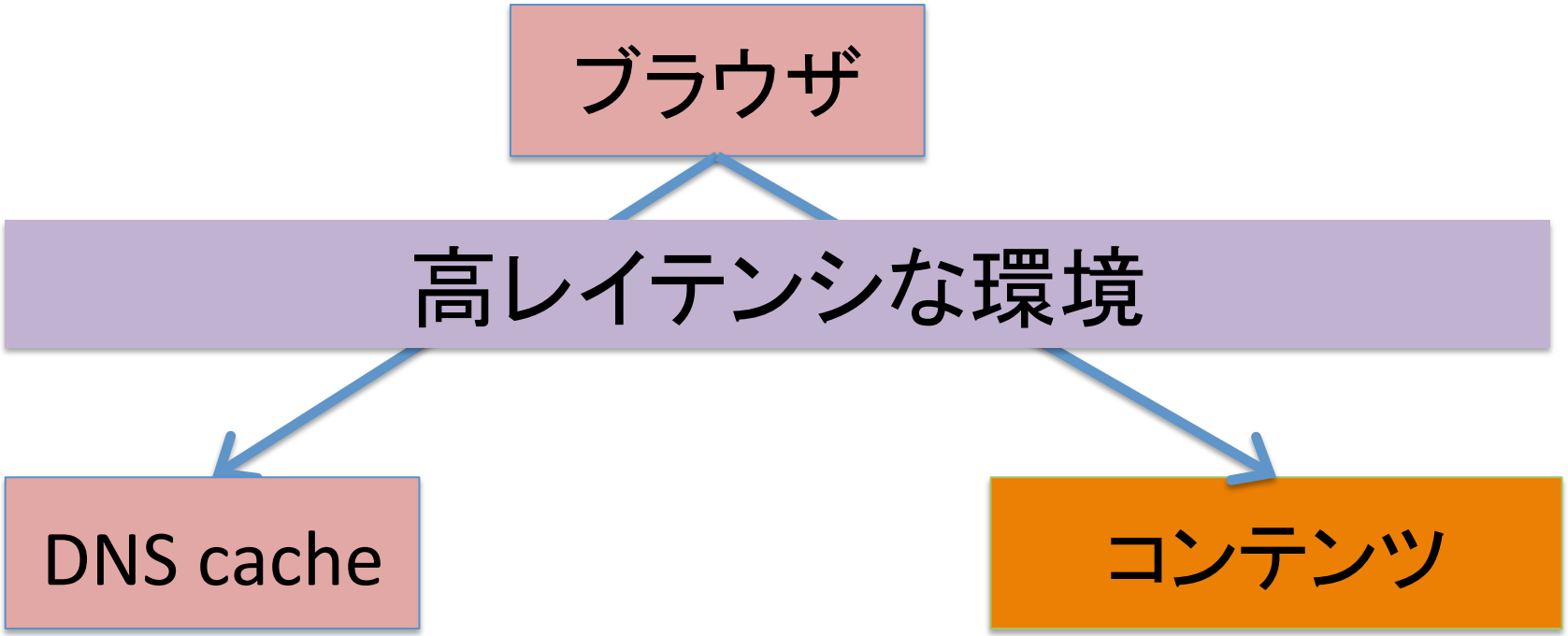
SPDY Forward Proxy



- SPDY、すばらしいと思います
- 今日は、この図の「ドキドキ」な話をします
- モバイル版chromeのデータ圧縮オプション

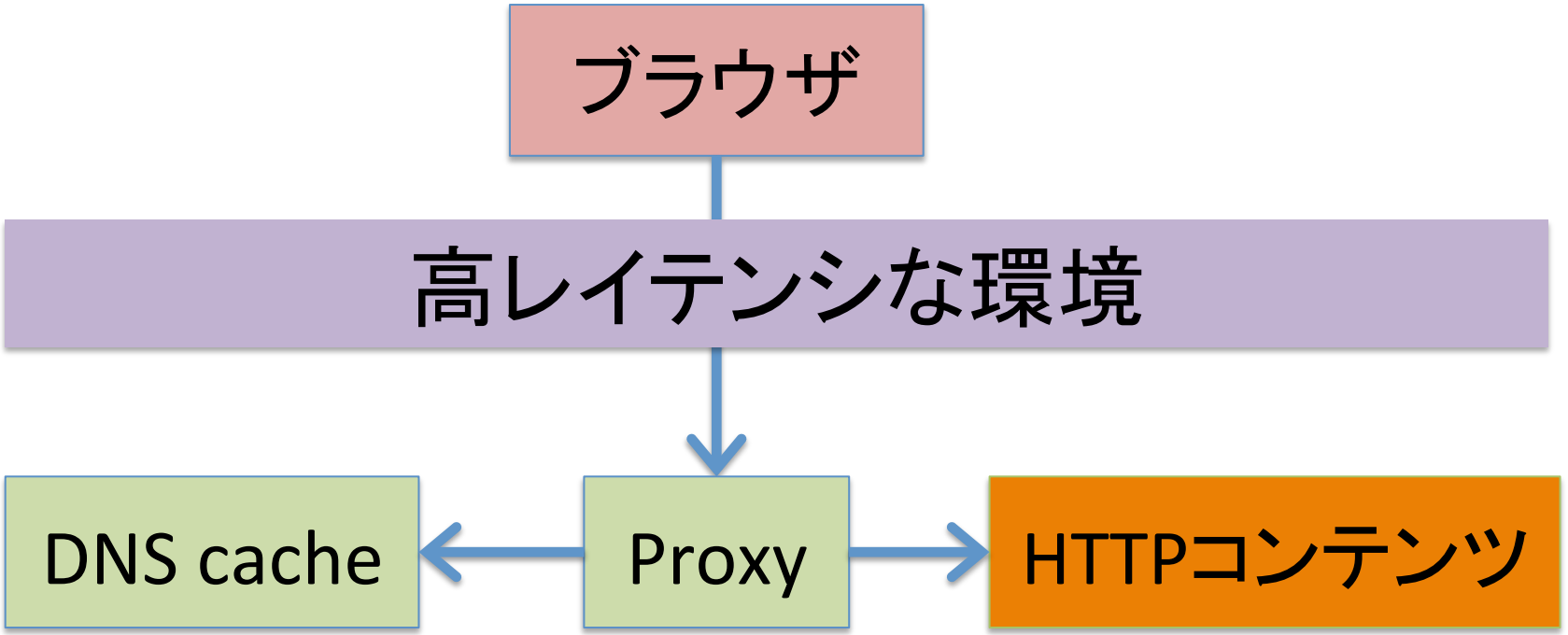
オリジナルはこちら
https://developers.google.com/chrome/mobile/docs/data-compression

SPDY Forward Proxyが無い場合



- 高レイテンシな転送路を複数回往復
- 暗号化されない
- コンテンツが圧縮されてないケースもある

SPDY Forward　Proxyがある場合



名前解決はProxyで行う
Proxyでコンテンツ圧縮を行う

ユーザーが嬉しいこと

- wifi区間でのセッションハイジャック対策
- データ転送の量が減る
- 高レイテンシへの対策がいろいろ
  - セッション数が減る
  - 名前解決が無線区間を通らない

現状の状況

- 機種割合
  - Android 53.33%
  - ios 41.43%
- スマートフォンにおけるブラウザ
  - chrome 3.76%
  - chrome以外 96.24%

UserAgentから判定（ただし、エンドユーザで詐称可能）
PCからスマートフォンになりすますことも可能なので、正確ではありません

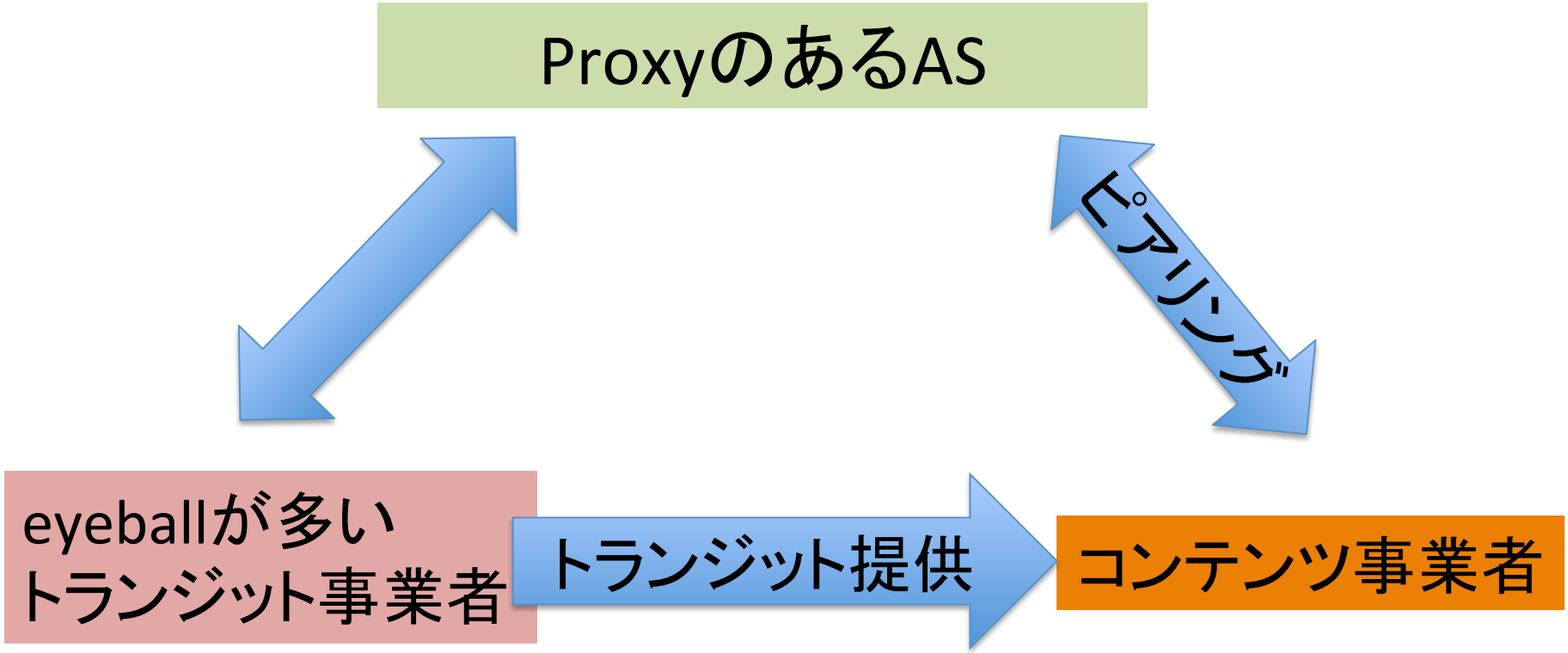
影響度は拡大する？

- Android4.4のOEMからブラウザが消える
  - webviewで独自実装ブラウザを作る？
  - chromeをライセンス契約してプレインストール？
- ブラウザのシェアが大きく変わる可能性も
  - SPDY Forward Proxyが爆発的に伸びると・・・

今回の切り口

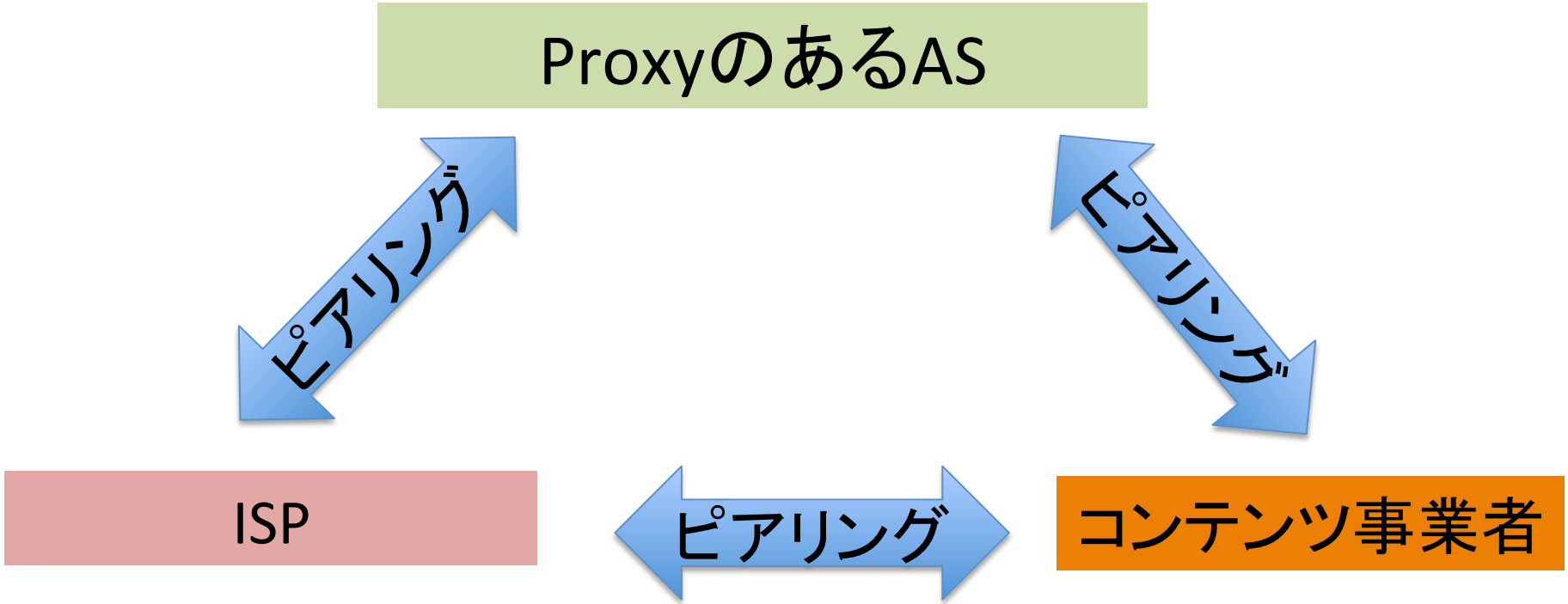
1. Proxyによるトラフィック中継
2. 外部からくるX-forwarded-forの処理問題

トラフィック中継による変化　1



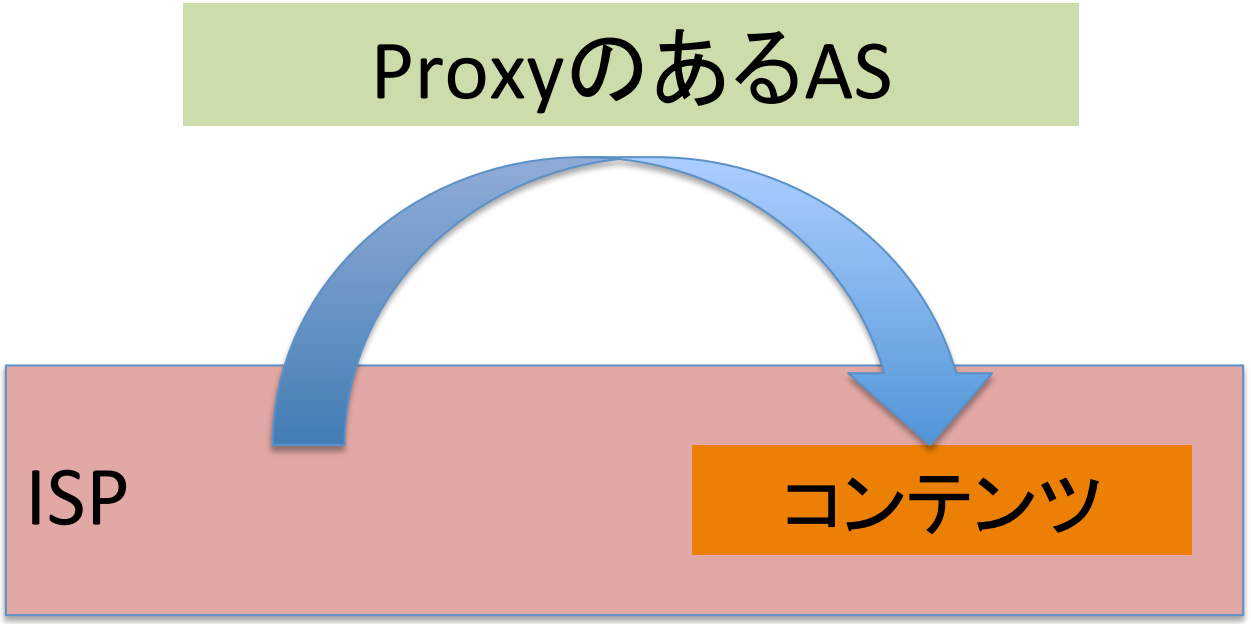
トランジット経由しなくなって嬉しい側面もあるかもしれない

トラフィック中継による変化　２



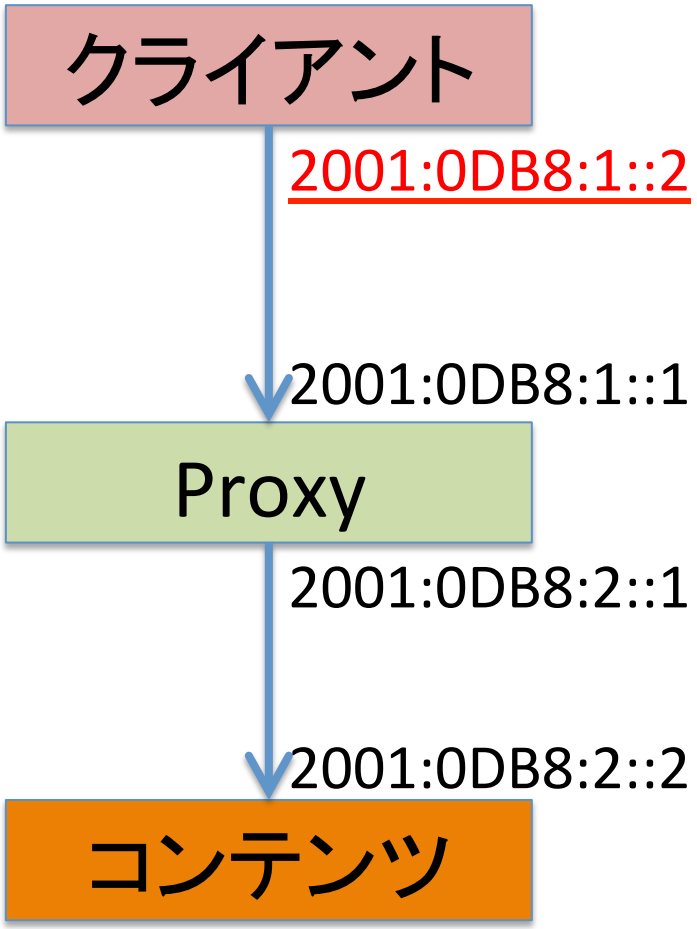
ビジネス上の要件に合わせてチューニングしたピアリングをしたとしても効果が得られない？

トラフィック中継による変化　３



eyeballユーザのトラフィックが同一ISP内にあるコンテンツを見るのにProxyを経由する

切り口２：接続元情報の扱い



Proxyが
リクエストヘッダ
X-Forwarded-Forを
2001:0DB8:1::2の値で
挿入する

X-Forward-Forのつらさ

- X-forwarded-forは改ざん可能
- 外から付いてきた同ヘッダーを使うのは危険
  - tcpセッションを確立した相手のアドレスを使う
  - 信用するアドレスリストを持つ手もある
- 外部からのX-forwarded-forは記録もされない
  - 各種調査の時に困ることになりそう

この先の見通し

- HTTP/2.0は暗号化なしも検討されている
- 無線区間を守るために暗号化は望ましい

- HTTP/2.0の非暗号化版が標準化されても、「http://」でブラウザに入力することになるので、SPDY Forward Proxyを通るのではないか

コンテンツを持つ者の選択肢

- 依存して生きて行く
  - うまくピアリングできるなら幸せかもしれない
  - デフォルトSSL化するコストを払わなくていい
  - 依存させてもらうことへの恐怖と戦う
- 依存しないで生きて行く
  - SSL対応して、SPDY等の対応まで行う

某サービスの場合

- httpsでもブラウザで閲覧可能にするための工数
  - 梅プラン
    - 全ドメインでエラーが無い
    - 外部リソース対応は、表示なし等で対応する
    - ページ遷移時にプロトコル維持を気にしない
  - 竹プラン
    - プロトコルを維持する
    - 難易度高めのサブドメインも対応する
  - 松プラン
    - すべてのリンクでプロトコルを引き継ぐ
    - 外部リソースをproxyする形で対応する

ざっくり工数

- 梅

  秘密 人日
- 竹

  秘密 人日
- 松

  秘密 人日

ただし、別途QA工数も必要

※firesheepの際に工数をざっくり計算したもの

具体的にどんな作業？

- js での http:// 固定記述への対応
- テンプレート での http:// 記述撲滅
- コード内 での http:// 記述撲滅
- DB 内に URL が入っている場合の対応
- ページ遷移時の http/https 状態保持ルール
- 外部リソース対策

現時点で目指したいところ

- 社外にリソースを提供部分のHTTPS化
  - 提供先がHTTPS化するための障壁にならない

依存したときの恐怖

- ピアリング関連でポリシーの変更
- ある日突然の終了
  - ISPさんにはどこ宛にトラフィックあるかわからない
- 障害に巻き込まれる
- X-forwarded-forの位置づけの変化

恐怖のシナリオと議論

- ISPさん
  - 売っていたトランジットが大きく減少するかも
  - 特定ASへの偏りが加速するかも
- CDN事業者さん
  - 「Proxyの近くにサーバが有ります」と売る？
  - SPDY等の付加サービス売りやすくなる？
- DC事業者さん
  - お客さんのSSL化動向で原価が変わるかも
- コンテンツ屋
  - firesheepのリスクがケアされた状況で、進化が止まる
  - 情報を抜かれうる状況

まとめ

- エンジニアとして
  - 自ら望んで進化しなければ明日は無い
  - だが、コストを無視するのは違う
  - さて、どう考えて説明して進みましょうか？