

# Pcard: 多様な情報端末やサービスを跨いだ 個人毎の情報流通支援フレームワーク

小田 朋宏  
中小路 久美代  
(株)SRA

ここに単純な  
マルチクラス文書分類器があります

# 分類器

## Bag-of-Words

$$\text{bow}_{\text{doc}} = \{w \rightarrow \text{count}(w, \text{doc}) / \sum_{w \in \text{doc}} \text{count}(w, \text{doc}) \mid w \in \text{doc}\}$$

$$\text{bow}_{\text{class}} = (\sum_{\text{doc} \in \text{class}} \text{bow}_{\text{doc}}) / |\sum_{\text{doc} \in \text{class}} \text{bow}_{\text{doc}}|$$

## マルチクラス分類器

$$\text{classify}(\text{Doc}) = \{\text{class} \mid \text{bow}_{\text{class}} \cdot \text{bow}_{\text{Doc}} \geq \text{Sim}_{\text{class}}\}$$

where

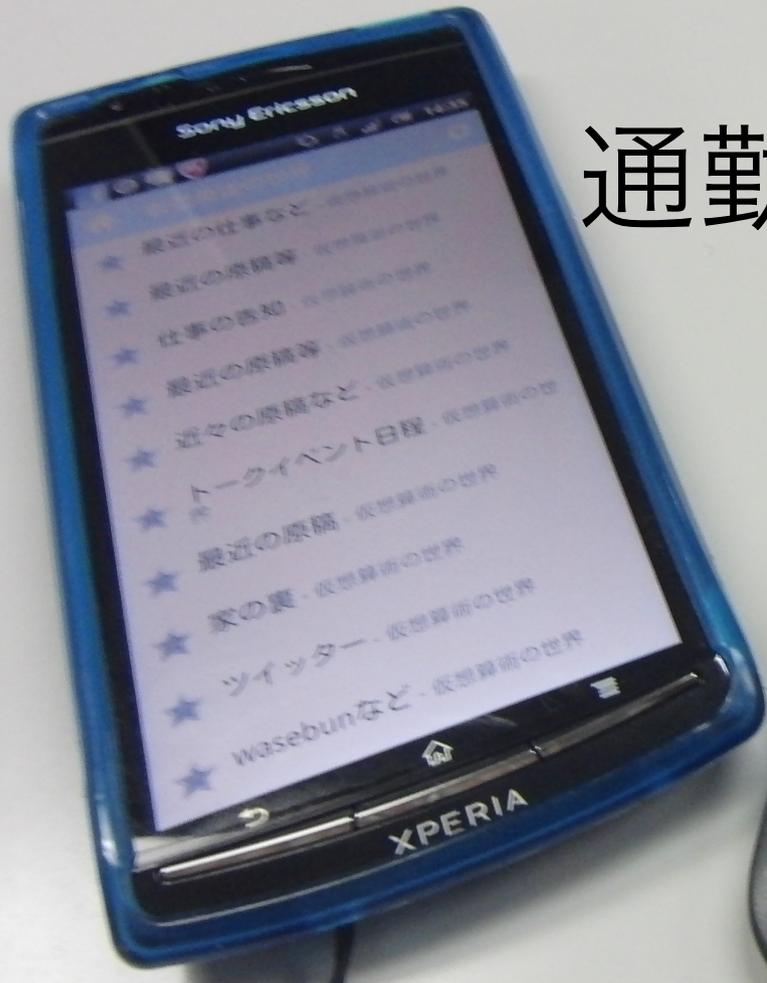
$$\text{Sim}_{\text{class}} = P_{100(1-\text{rate}_{\text{class}})}(\{\text{bow}_{\text{doc}} \cdot \text{bow}_{\text{class}} \mid \text{doc} \in \text{class}\})$$

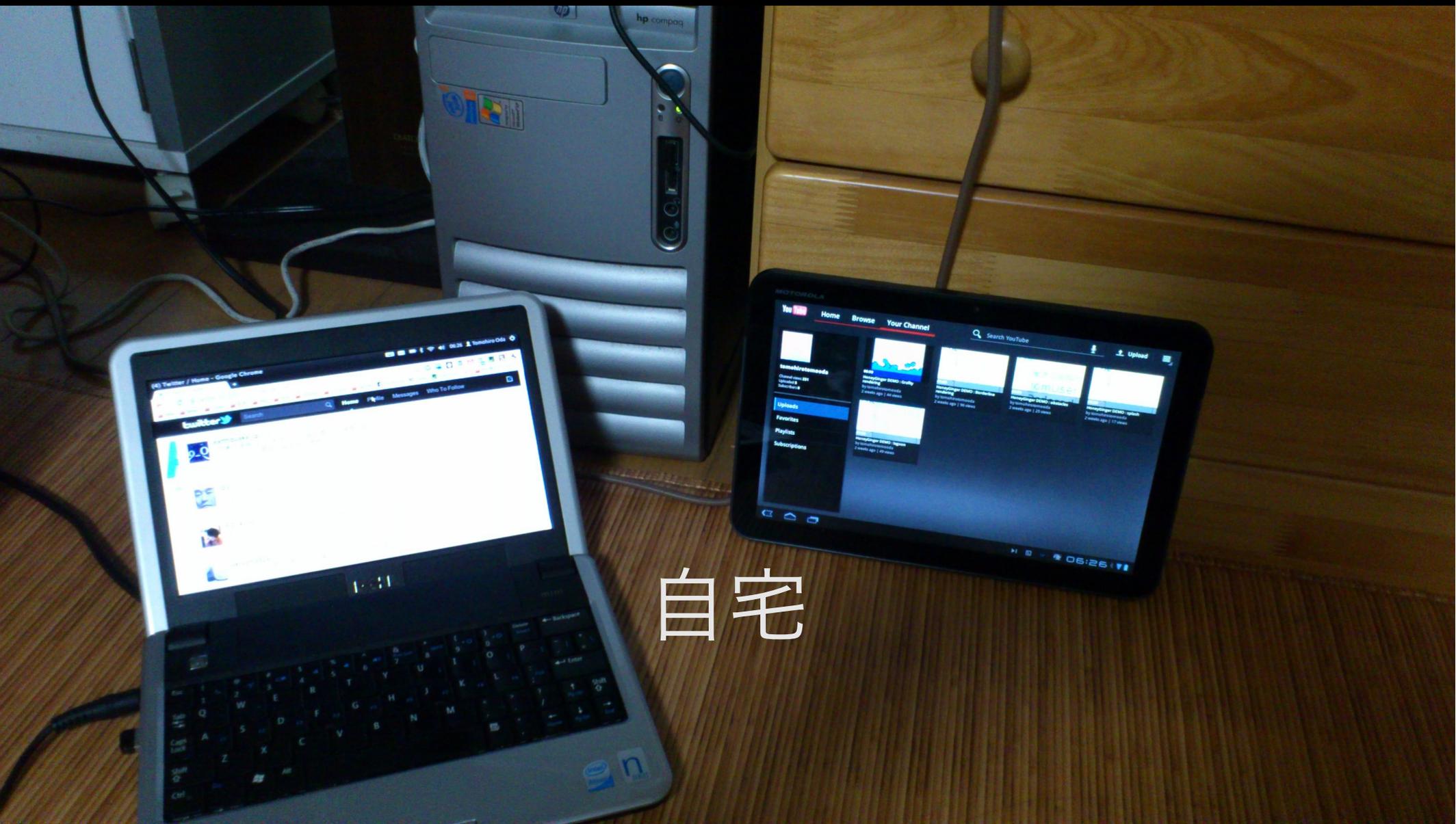
これを使って、情報システムのバラバラ感を  
なんとかしたい・・・

生活の様々なシーンに入ってきた  
情報デバイス

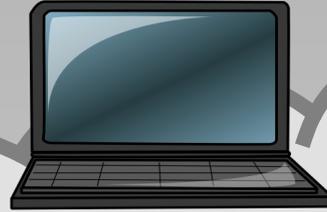


通勤中

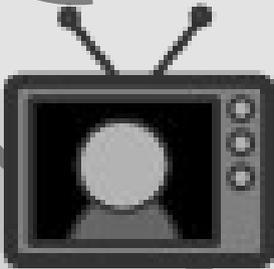




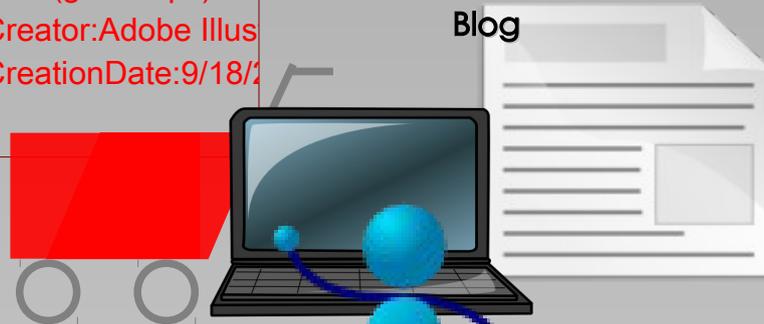
自宅



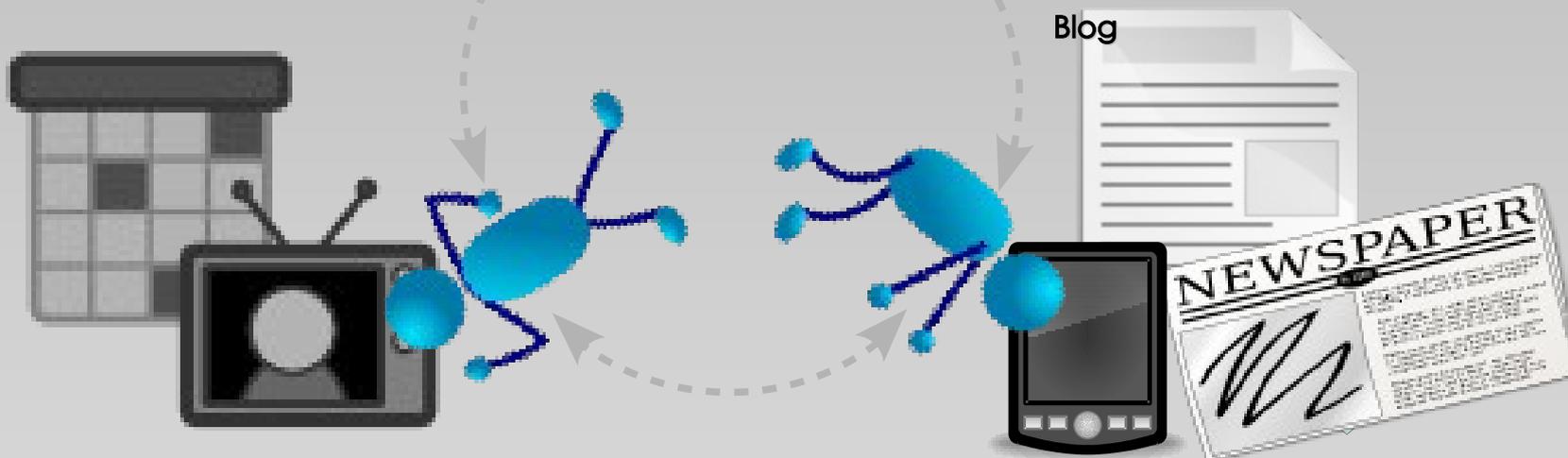
- どの情報デバイスもネットに繋がってるけど...



Title:(gift55.eps)  
Creator:Adobe Illus  
CreationDate:9/18/2



- イマイチ、情報の連携はバラバラなまま



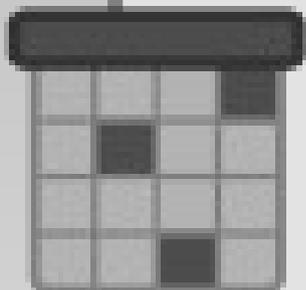
Title:(gift55.eps)  
Creator:Adobe Illus  
CreationDate:9/18/2

Blog



- クラウドサービスで情報を共有できるけど...

Blog



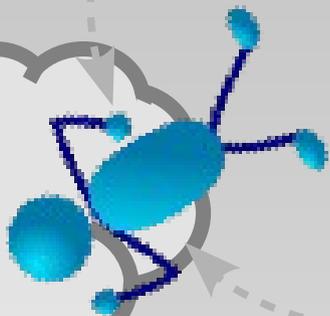
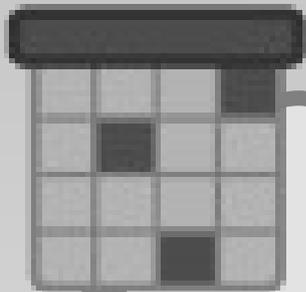
Title:(gift55.eps)  
Creator:Adobe Illus  
CreationDate:9/18/2



Blog

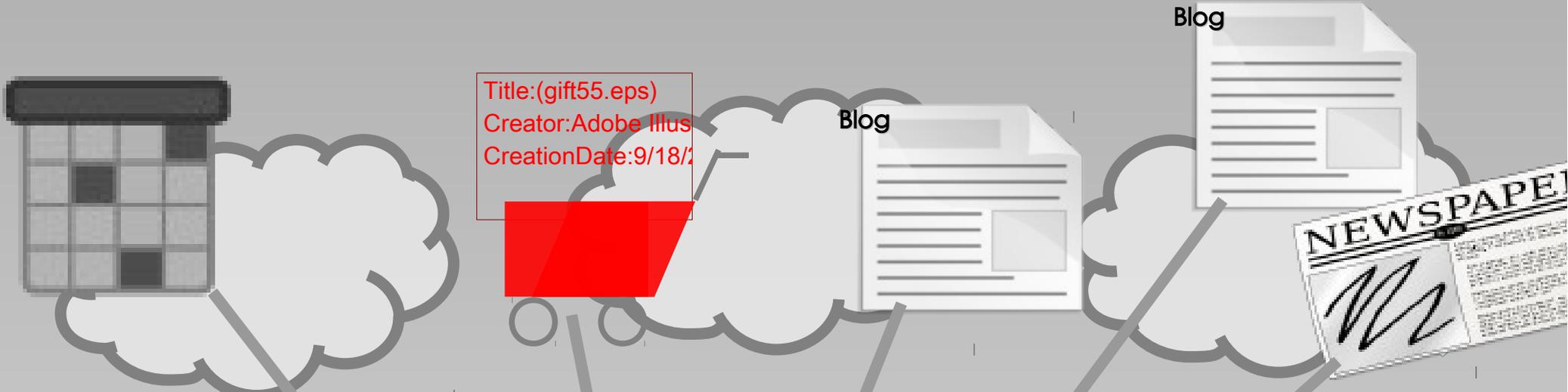


- でも、各クラウドサービスに閉じていることが多い



Blog





Title:(gift55.eps)  
Creator:Adobe Illus  
CreationDate:9/18/2008

- evernoteはクラウドサービスを越えて共有できるけど…

**EVERNOTE** サインアウト (tomooda)

新しいタブ

tomooda のノートブック

クリックしてタグを追加

T+ クリックしてコメントを追加

URL をクリップ

ノート

全てのノート      newtab

- クリップ操作をした時だけ共有できる



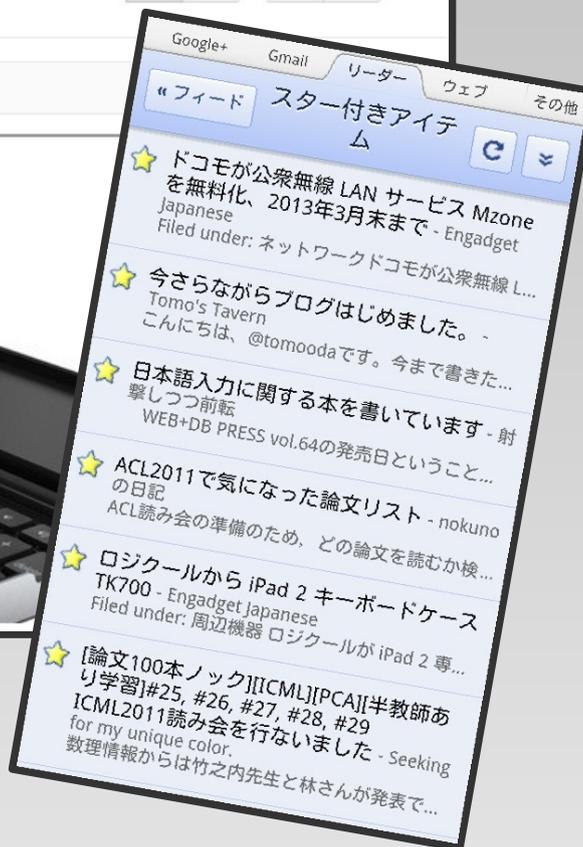
- 最初読んだ時は読み飛ばしてクリップせずについて、後になって

「あ、そういえば、前読んだような…」

と思っても…

- 後の祭り

- Google Readerで色々なデバイスからブログを読めるけど...



- 会社からは「見えてくれない」 ブログも表示されちゃう

ホーム

▼ すべてのアイテム (1)

- ★ スター付きアイテム
- 📊 トレンド
- 📖 アイテムを参照

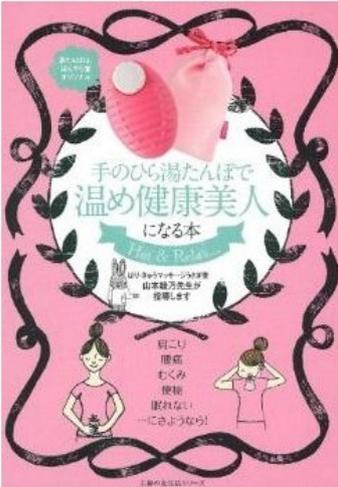
▶ ピックアップ

▶ 登録フィード

🔄 新着 1 件 ▼ すべて既読にする ▼ 設定を表示... ▼

すべてのアイテム

ミニ湯たんぼが付録についてムック登場!『手のひら湯たんぼで温め健康美人』📖 ☆  
ガジェット通信

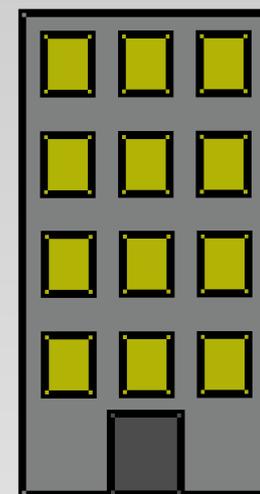


付録雑誌が大人気ですが、なんとこの秋、「湯たんぼ」山本綾乃さん監修『手のひら湯たんぼで 温め健康美人になる本』(主婦の友社/1100円税込み)。「体調不良の時は、まず温める!」。そんな情報がぎゅっと詰まった1冊となっています。

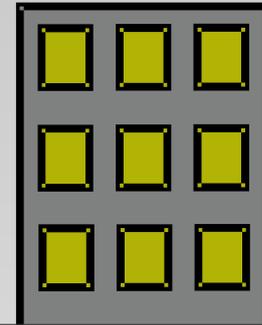
私自身、実は湯たんぼマニア。といってもソフトカバーとハードカバーの2種類を持つだけなんです。

■著者データ  
マイスピ

- 仕事中とか、通勤中とか、リビングとか、書斎とか、



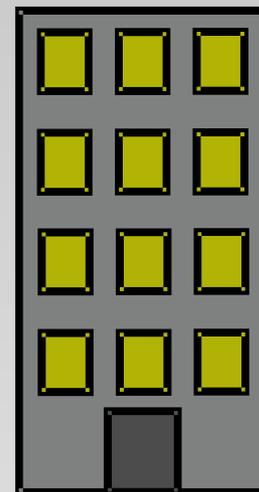
- そういった社会的コンテキストに応じて情報を配送しないと困る



欲しいのは

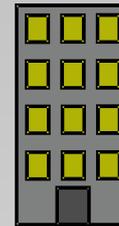
欲しいのは

- 色々な社会的コンテキストの中で



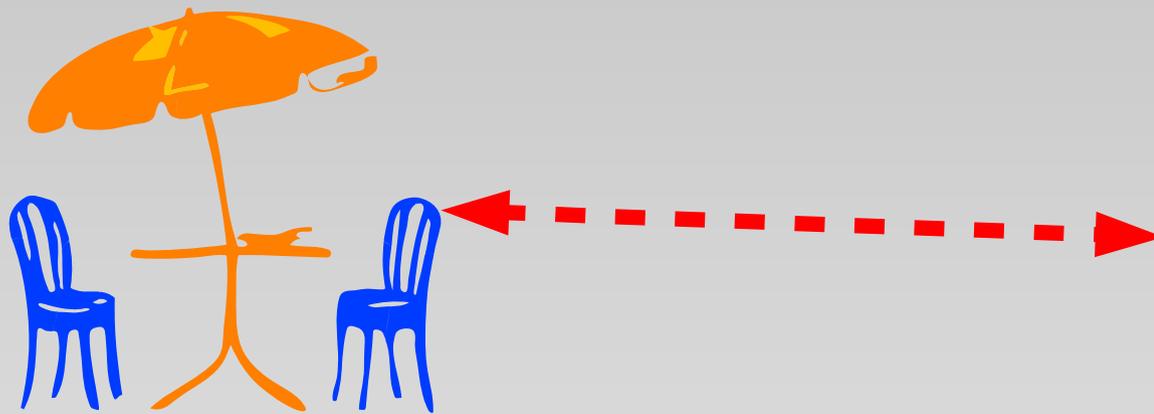
# 欲しいのは

- 色々な社会的コンテキストの中で
- 色々なトピックの情報を読んだものを



# 欲しいのは

- 色々な社会的コンテキストの中で
- 色々なトピックの情報を読んだものを
- 未来のコンテキストで使える場合に備えて



## 欲しいのは

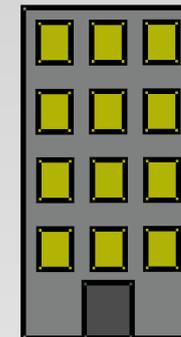
- 色々な社会的コンテキストの中で
- 色々なトピックの情報を読んだものを
- 未来のコンテキストで使える場合に備えて
- 「全てのコンテンツを」自動的に保存して



a u t o m a

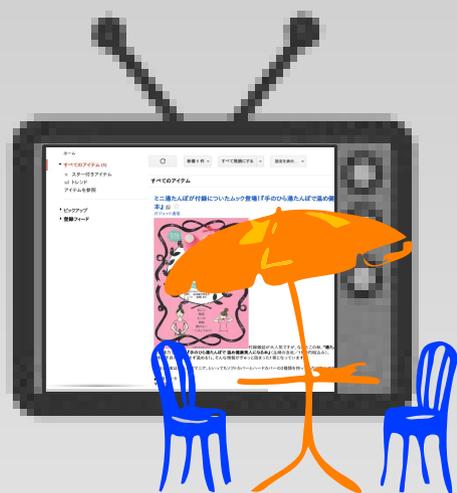
# 欲しいのは

- 色々な社会的コンテキストの中で
- 色々なトピックの情報を読んだものを
- 未来のコンテキストで使える場合に備えて
- 「全てのコンテンツを」自動的に保存して
- 色々な社会的コンテキストに合わせて



# 欲しいのは

- 色々な社会的コンテキストの中で
- 色々なトピックの情報を読んだものを
- 未来のコンテキストで使える場合に備えて
- 「全てのコンテンツを」自動的に保存して
- 色々な社会的コンテキストに合わせて
- 色々な情報デバイスで再表示する



欲しいのは

- 色々な社会的コンテキストの中で
- 色々なトピックの情報を読んだものを
- 未来のコンテキストで使える場合に備えて
- 「全てのコンテンツを」自動的に保存して
- 色々な社会的コンテキストに合わせて
- 色々な情報デバイスで再提示する

そんな情報流通を実現したい

多様な社会的コンテキストの中での  
情報の再提示

## 前提1

- 各情報は1つ以上のトピックを含んでいる
  - 例：情報推薦、統計モデル、機械学習、…

## 前提1

- 各情報は1つ以上のトピックを含んでいる
  - 例：情報推薦、統計モデル、機械学習、…
- ただし、最初に読んだ時に正確にタグ付けできるとは限らない
  - 後になって出てくる関連性

## 前提2

- 各情報デバイスはある程度決まった社会的コンテキストの中で使われる
  - 例：スマートフォンを電車の中で、デスクトップPCを会社で、ネットブックとタブレットを家庭

## 前提2

- 各情報デバイスはある程度決まった社会的コンテキストの中で使われる
  - 例：スマートフォンを電車の中で、デスクトップPCを会社で、ネットブックとタブレットを家庭
- ただし、情報デバイスと社会的コンテキストは1対1対応ではない
  - 1つの情報デバイスを複数の社会的コンテキストで利用
  - 1つの社会的コンテキストで複数の情報デバイスを利用

## 前提3

- 社会的コンテキストとトピックの適合性により、情報デバイスごとのトピックの偏りが出る

## 前提3

- 社会的コンテキストとトピックの適合性により、情報デバイスごとのトピックの偏りが出る
- ただし、偏りはあるが、結構オーバーラップしている

したがって

- 各情報デバイスで読み書きしている情報の偏りには社会的コンテキストがある程度反映されている

そこで

- 情報デバイスをクラスとするマルチクラスな文書自動分類器を使って、全ての文書を社会的コンテキストが適合する全ての情報デバイスへ自動転送する

これまで試作した  
情報再提示のプロトタイプ

# cSorter

The screenshot shows a web browser window with the URL `http://localhost:8080/bookmarks?folder=Recommender`. The page title is "Workspace: Recomme...". The main content area is titled "Workspace" and contains a navigation bar with "Classifier", "Parser", and "<- Recommender ->". There are buttons for "Update bookmarks", a search input field, and a "new" button. Below the navigation bar, there is a section for the "Recommender" classifier, showing "(documents: 3, recommended documents: 10)" and "keywords: wiki, user, data, music, recommendation". To the right of this section are buttons for "delete", "Recommender" (input field), "rename", and "Auto-Category" (dropdown menu). The main area is divided into two columns of document cards. Each card has a close button (X), a title, and keywords. The left column contains three cards: "Naive Bayes classifier - Wikipedia, the free encyclopedia" (keywords: spam, wiki, independence), "Recommender system - Wikipedia, the free encyclopedia" (keywords: wiki, user, data), and "Collaborative filtering - Wikipedia, the free encyclopedia" (keywords: music, user, recommendation). The right column contains five cards: "The Python Standard Library \* Python v2.6.4 docume ..." (keywords: data, user), "Supervised learning - Wikipedia, the free encyclopedia" (keywords: data, spam, wiki), "L3D Philosophy" (keywords: user), "Bayesian network - Wikipedia, the free encyclopedia" (keywords: data, wiki), and "Artificial intelligence - Wikipedia, the free encyclopedia" (keywords: data, music, wiki).

Workspace: Recomme... x

← → ↻ 🏠 ☆ `http://localhost:8080/bookmarks?folder=Recommender` ▶ 📄 🔧

## Workspace

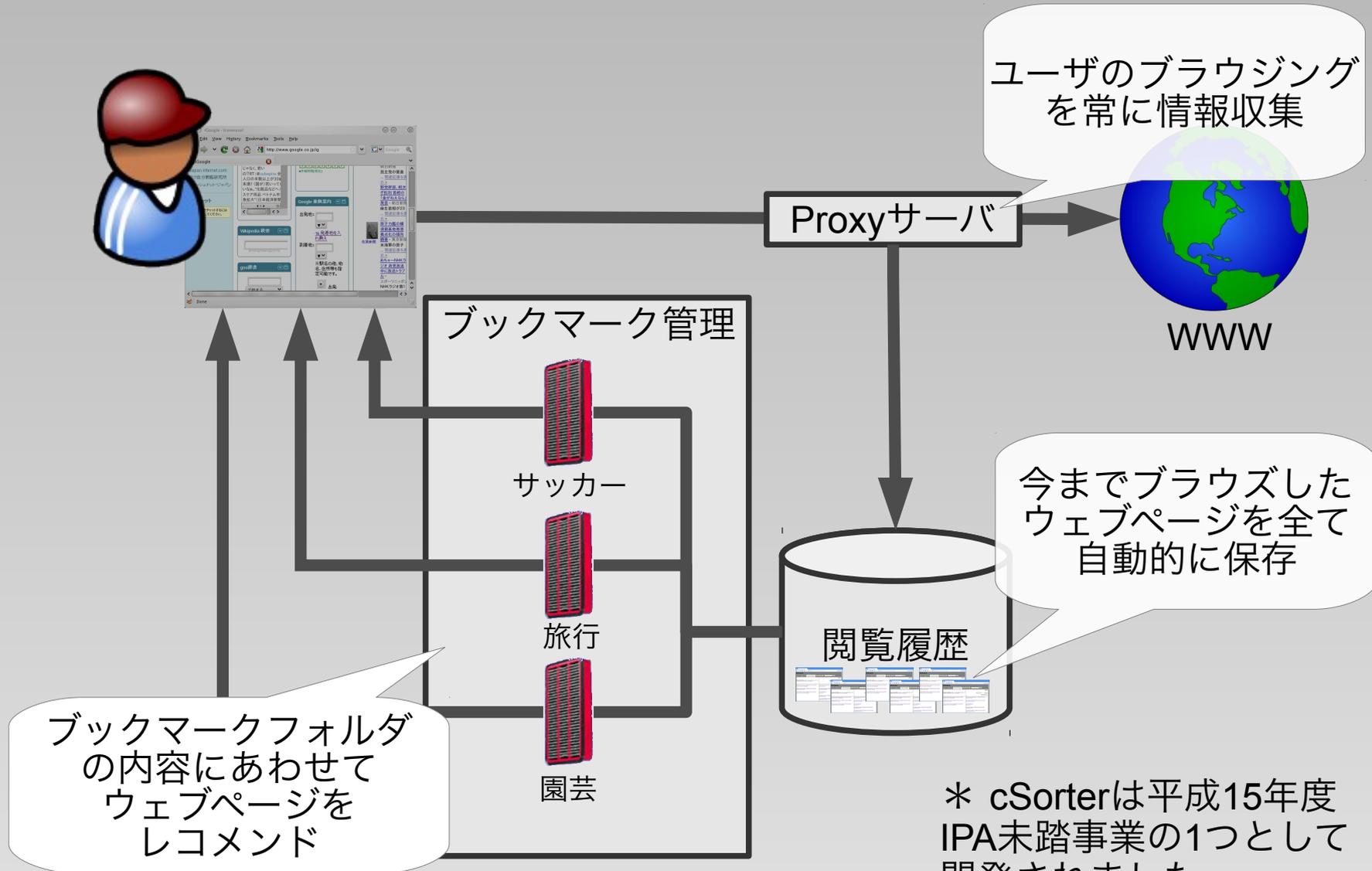
Classifier Parser <- Recommender -> Update bookmarks [input] new

Recommender (documents: 3, recommended documents: 10)  
keywords: wiki, user, data, music, recommendation

[delete] [Recommender] [rename] [Auto-Category ▼]

- X [Naive Bayes classifier - Wikipedia, the free encyclopedia](#)  
keywords: spam, wiki, independence
- X [Recommender system - Wikipedia, the free encyclopedia](#)  
keywords: wiki, user, data
- X [Collaborative filtering - Wikipedia, the free encyclopedia](#)  
keywords: music, user, recommendation
- <- [The Python Standard Library \\* Python v2.6.4 docume ...](#)  
keywords: data, user
- <- [Supervised learning - Wikipedia, the free encyclopedia](#)  
keywords: data, spam, wiki
- <- [L3D Philosophy](#)  
keywords: user
- <- [Bayesian network - Wikipedia, the free encyclopedia](#)  
keywords: data, wiki
- <- [Artificial intelligence - Wikipedia, the free encyclopedia](#)  
keywords: data, music, wiki

# cSorter の仕組み



\* cSorterは平成15年度IPA未踏事業の1つとして開発されました。

## cSorterは

- トピックをクラスとする  
マルチクラスな文書自動分類器を使って、

過去に閲覧した全てのウェブページから

トピック的コンテキストが適合する  
全てのブックマークフォルダへ

推薦する

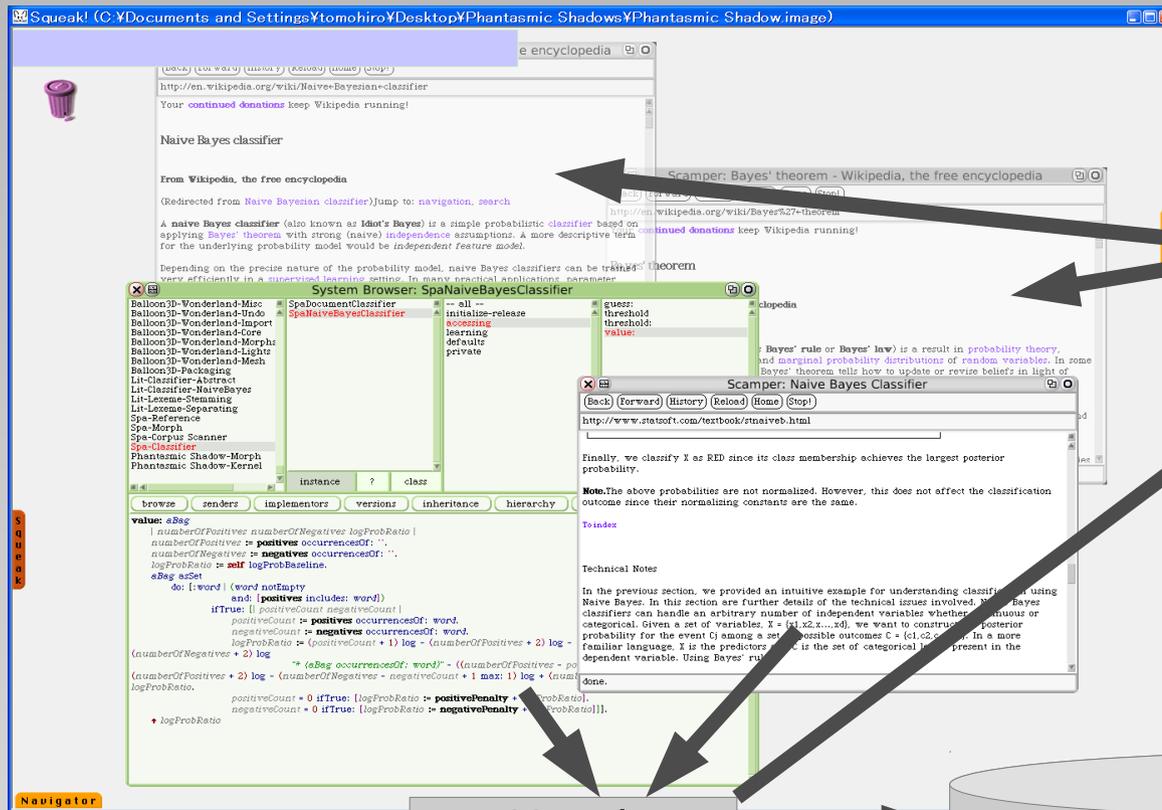
# Phantasmic Shadows

The screenshot displays the Squeak! environment with the following components:

- System Browser: SpaNaiveBayesClassifier**
  - Classes: Balloon3D-Wonderland-Misc, Balloon3D-Wonderland-Undo, Balloon3D-Wonderland-Import, Balloon3D-Wonderland-Core, Balloon3D-Wonderland-Morphs, Balloon3D-Wonderland-Lights, Balloon3D-Wonderland-Mech, Balloon3D-Packaging, Lit-Classifier-Abstract, Lit-Classifier-NaiveBayes, Lit-Lexeme-Stemming, Lit-Lexeme-Separating, Spa-Reference, Spa-Morph, Spa-Corpus Scanner, **Spa-Classifer**, Phantasmic Shadow-Morph, Phantasmic Shadow-Kernel.
  - Methods: initialize-release, accessing, learning, defaults, private.
  - Value: `aBag`
  - Code:

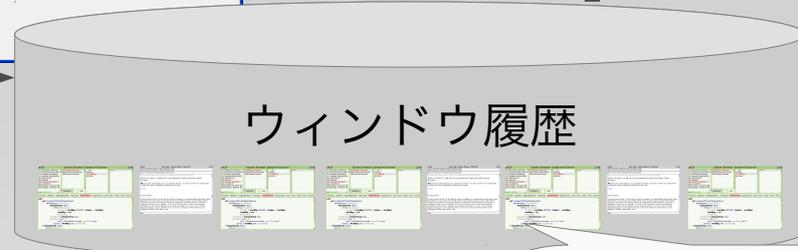
```
numberOfPositives numberOfNegatives logProbRatio |
numberOfPositives := positives occurrencesOf: ''.
numberOfNegatives := negatives occurrencesOf: ''.
logProbRatio := self logProbBaseline.
aBag asSet
do: [:word | (word notEmpty
and: [positives includes: word])
ifTrue: [| positiveCount negativeCount |
positiveCount := positives occurrencesOf: word.
negativeCount := negatives occurrencesOf: word.
logProbRatio := (positiveCount + 1) log - (numberOfPositives + 2) log -
(numberOfNegatives + 2) log
* (aBag occurrencesOf: word) - ((numberOfPositives - po
(numberOfPositives + 2) log - (numberOfNegatives - negativeCount + 1 max: 1) log + (num
logProbRatio.
positiveCount = 0 ifTrue: [logProbRatio := positivePenalty + logProbRatio].
negativeCount = 0 ifTrue: [logProbRatio := negativePenalty + logProbRatio]].
↑ logProbRatio
```
- Scamper: Bayes' theorem - Wikipedia, the free encyclopedia**
  - URL: <http://en.wikipedia.org/wiki/Bayes%27-theorem>
  - Text: "A **naive Bayes classifier** (also known as **Idiot's Bayes**) is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be *independent feature model*. Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation of a naive Bayes classifier is done using the *expectation-maximization algorithm*. The Bayes' theorem or Bayes' law is a result in probability theory, and marginal probability distributions of random variables. In some cases, Bayes' theorem tells how to update or revise beliefs in light of new evidence."
- Scamper: Naive Bayes Classifier**
  - URL: <http://www.statsoft.com/textbook/stnaiveb.html>
  - Text: "Finally, we classify X as RED since its class membership achieves the largest posterior probability. Note. The above probabilities are not normalized. However, this does not affect the classification outcome since their normalizing constants are the same. To index Technical Notes In the previous section, we provided an intuitive example for understanding classification using Naive Bayes. In this section are further details of the technical issues involved. Naive Bayes classifiers can handle an arbitrary number of independent variables whether continuous or categorical. Given a set of variables,  $X = \{x_1, x_2, \dots, x_d\}$ , we want to construct the posterior probability for the event  $C_j$  among a set of possible outcomes  $C = \{c_1, c_2, \dots, c_d\}$ . In a more familiar language,  $X$  is the predictors and  $C$  is the set of categorical levels present in the dependent variable. Using Bayes' rule: done."

# Phantasmic Shadows の仕組み



TFIDF類似度による  
フィルタ

UIリスナー



開いているウィンドウ  
から常に情報収集

今まで開いた  
ウィンドウを全て  
自動的に保存

# Phantasmic Shadowsは

- 文書自動分類器を使って、

過去に開いた全てのソースコード、ウェブページ、  
テキスト文書から

タスク的コンテキストが適合するものを

ユーザのタスクを邪魔しないように  
周辺領域や穏やかな視覚効果によって

推薦する

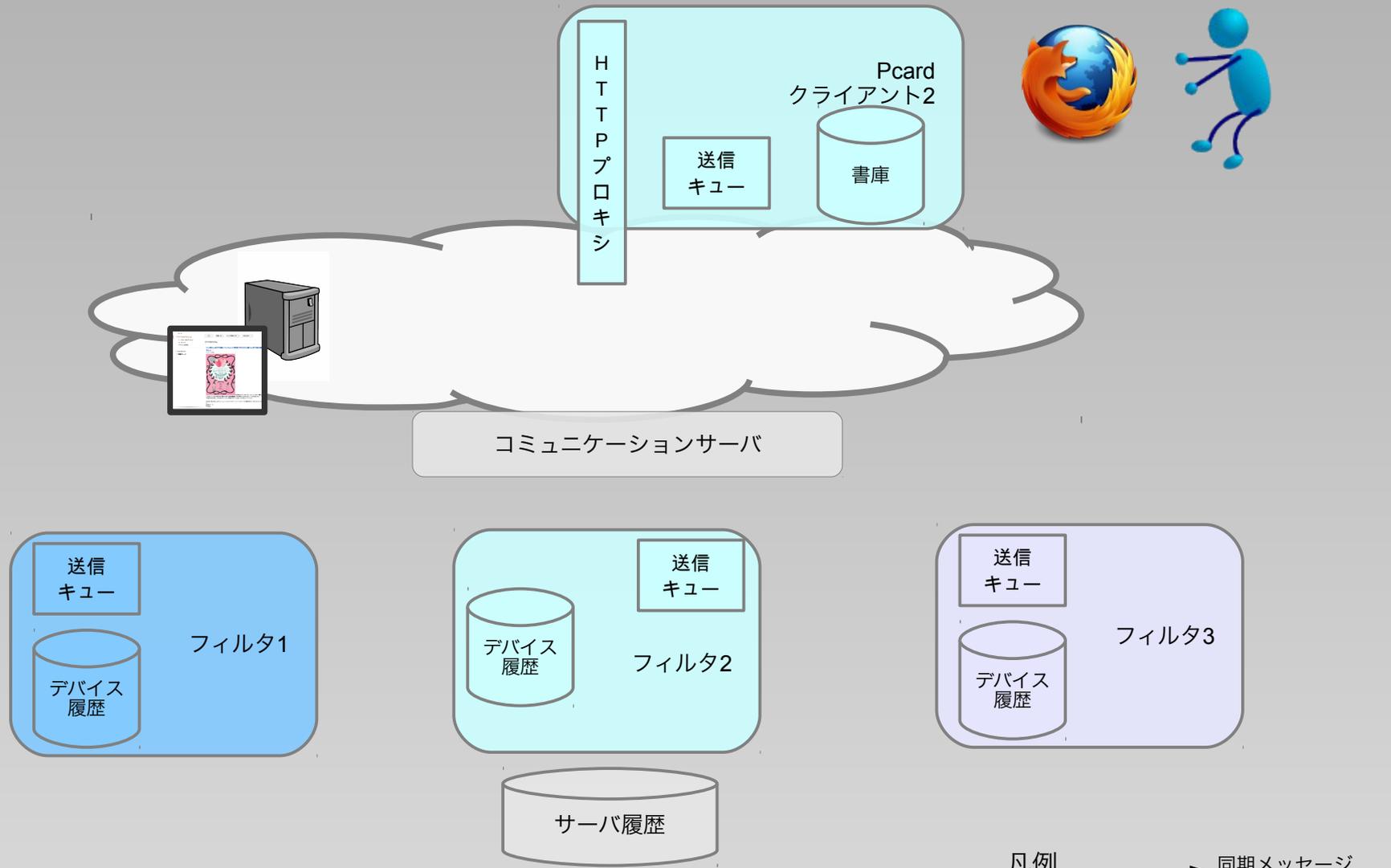
## これまで試作したプロトタイプの課題

- 複数の情報デバイスにまたがった情報再提示
- 情報デバイスと社会的コンテキスト、トピックの相関  
を利用しての社会的コンテキストによる情報推薦

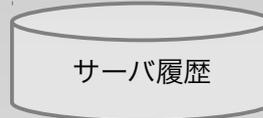
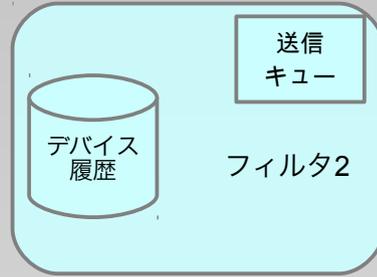
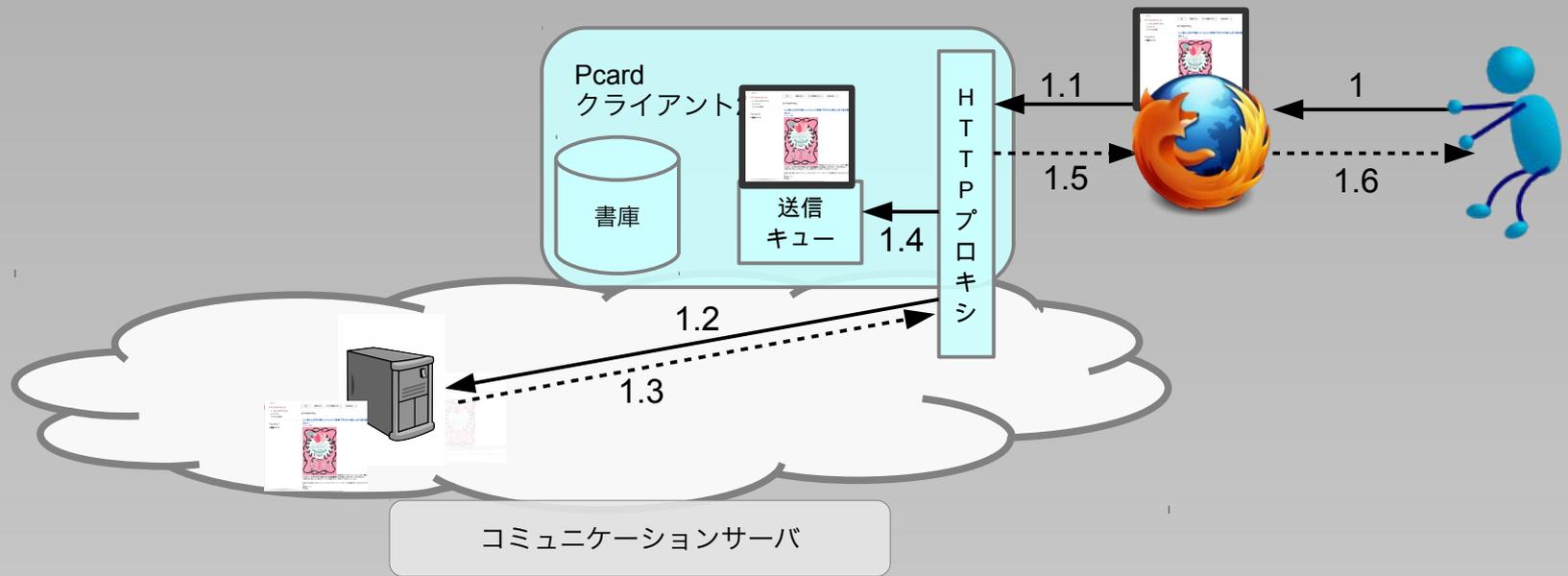
Pcard



# Pcardの構成

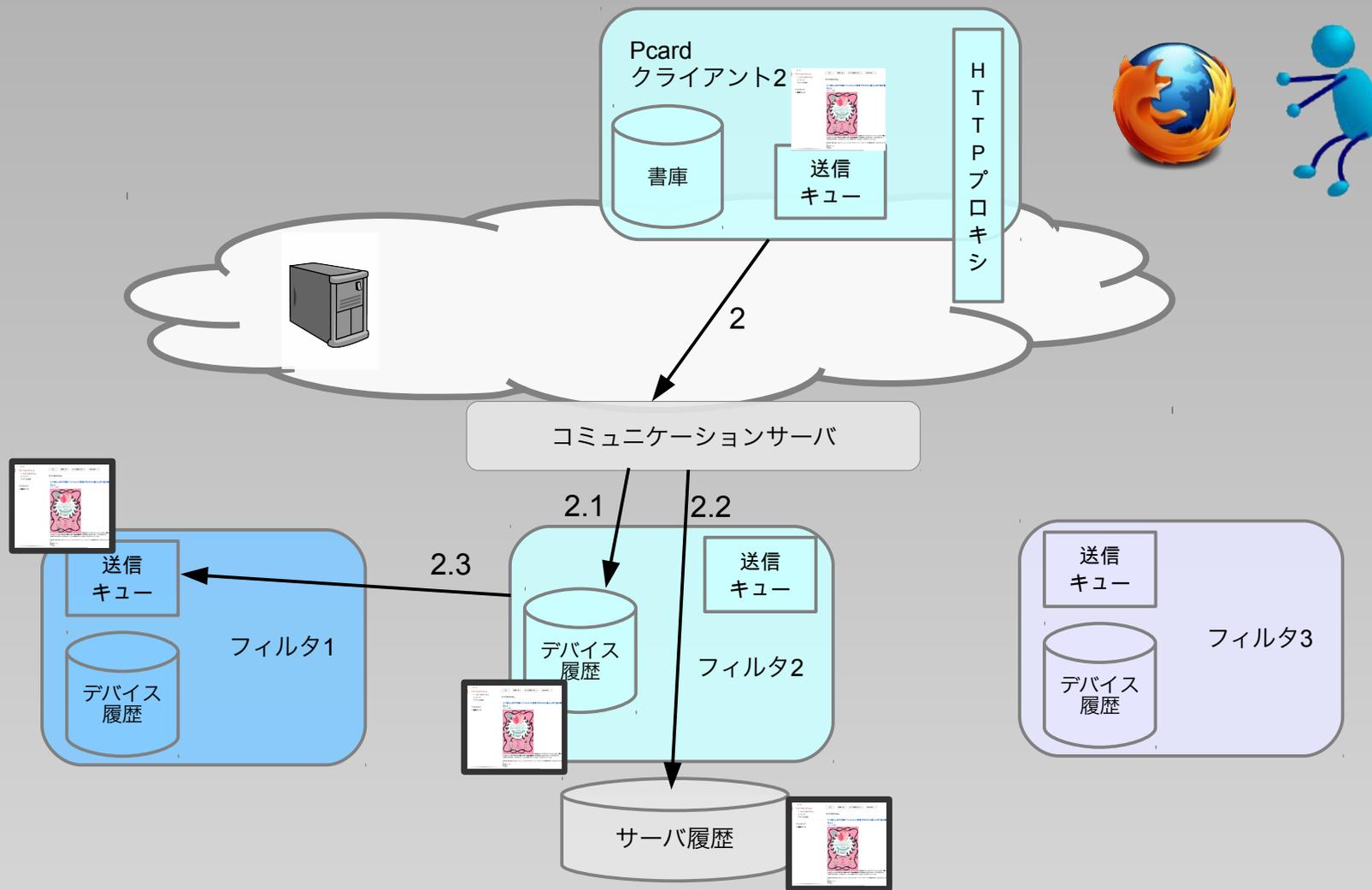


# Pcardの動作



# Pcardの動作 (HTTPの 捕捉)





# Pcardの動作 (マルチクラス分類)

凡例

- 同期メッセージ
- 非同期メッセージ
- レスポンス

# 分類器

## Bag-of-Words

$$\text{bow}_{\text{doc}} = \{w \rightarrow \text{count}(w, \text{doc}) / \sum_{w \in \text{doc}} \text{count}(w, \text{doc}) \mid w \in \text{doc}\}$$

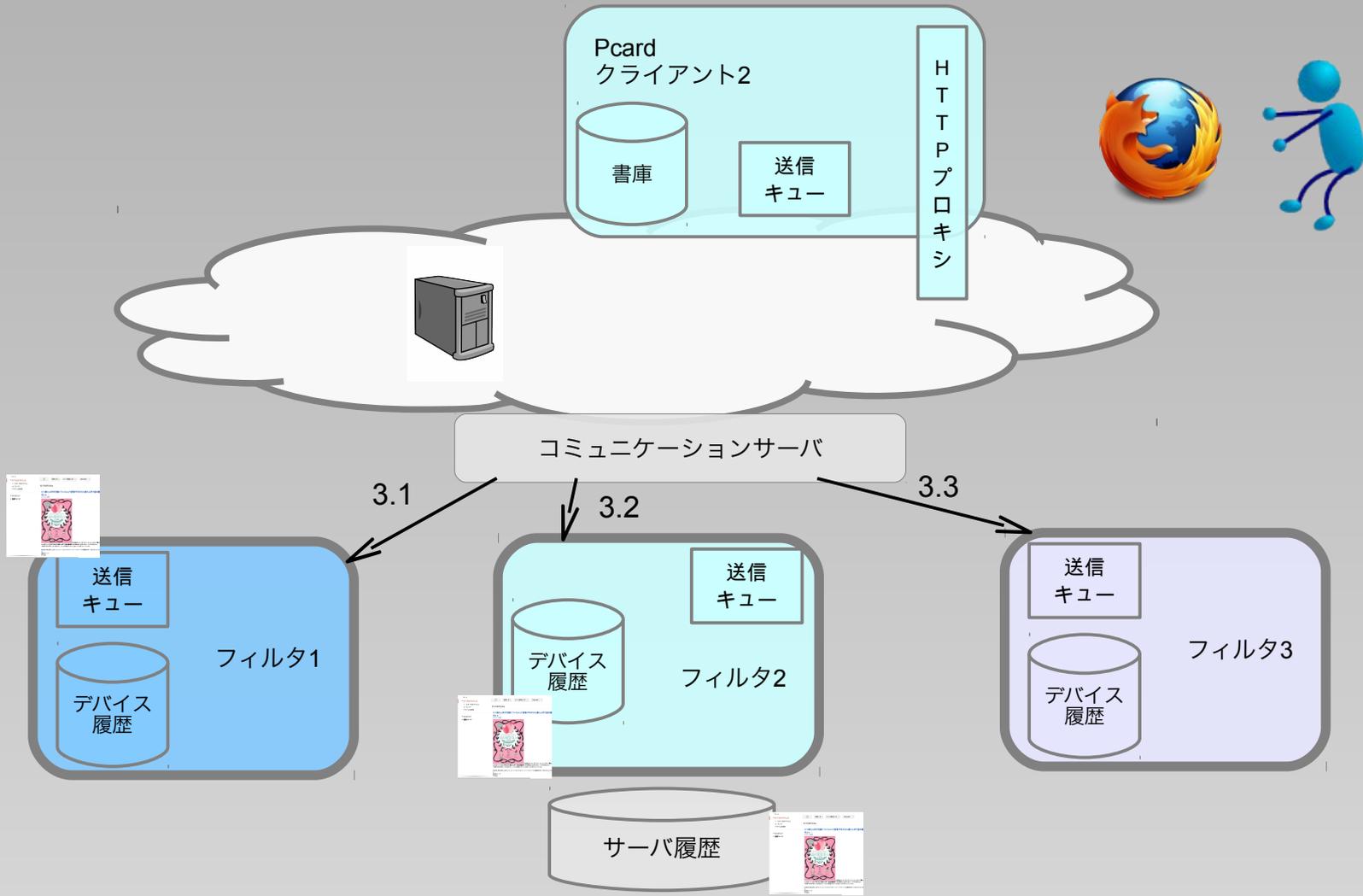
$$\text{bow}_{\text{dev}} = (\sum_{\text{doc} \in \text{dev}} \text{bow}_{\text{doc}}) / |\sum_{\text{doc} \in \text{dev}} \text{bow}_{\text{doc}}|$$

## マルチクラス分類器

$$\text{classify}(\text{Doc}) = \{ \text{dev} \mid \text{bow}_{\text{dev}} \cdot \text{bow}_{\text{Doc}} \geq \text{Sim}_{\text{dev}} \}$$

where

$$\text{Sim}_{\text{dev}} = P_{100(1-\text{rate}_{\text{dev}})}(\{\text{bow}_{\text{doc}} \cdot \text{bow}_{\text{dev}} \mid \text{doc} \in \text{dev}\})$$



# Pcardの動作 (フィルタ更新)

- 凡例
- 同期メッセージ
  - - - - -> 非同期メッセージ
  - .....> レスポンス

# 分類器

## Bag-of-Words

$$\text{bow}_{\text{doc}} = \{w \rightarrow \text{count}(w, \text{doc}) / \sum_{w \in \text{doc}} \text{count}(w, \text{doc}) \mid w \in \text{doc}\}$$

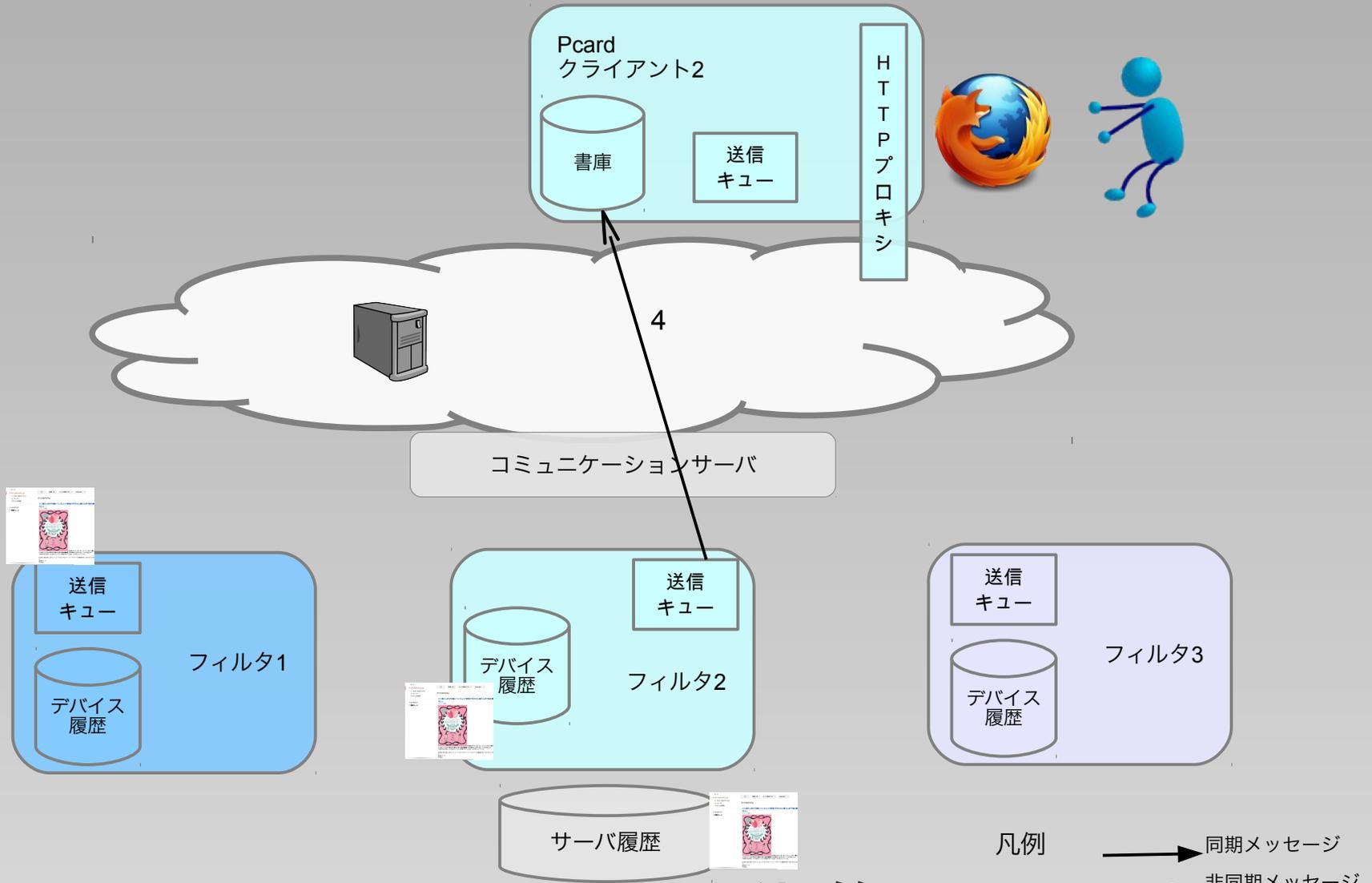
$$\text{bow}_{\text{dev}} = \left( \sum_{\text{doc} \in \text{dev}} \text{bow}_{\text{doc}} \right) / \left| \sum_{\text{doc} \in \text{dev}} \text{bow}_{\text{doc}} \right|$$

## マルチクラス分類器

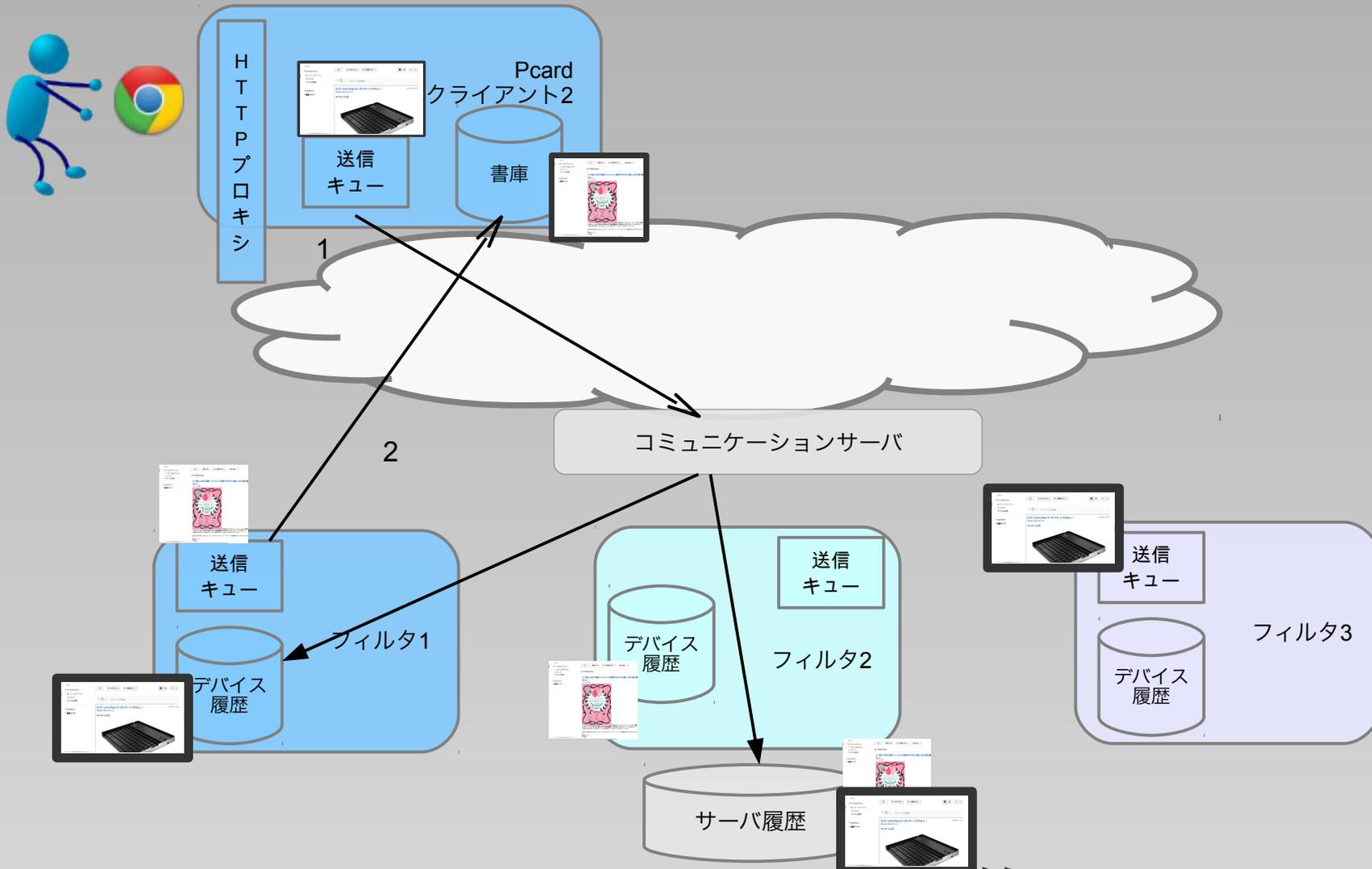
$$\text{classify}(\text{Doc}) = \{ \text{dev} \mid \text{bow}_{\text{dev}} \cdot \text{bow}_{\text{Doc}} \geq \text{Sim}_{\text{dev}} \}$$

where

$$\text{Sim}_{\text{dev}} = P_{100(1-\text{rate}_{\text{dev}})}(\{ \text{bow}_{\text{doc}} \cdot \text{bow}_{\text{dev}} \mid \text{doc} \in \text{dev} \})$$



# Pcardの動作 (他デバイスからの情報の配送)



# Pcardの動作 (他デバイス)

- 凡例
- 同期メッセージ
  - 非同期メッセージ
  - > レスポンス

- 利用者がすること
  - 各文書
    - タグ付け
    - 「共有」メニューを開く
  - 各情報デバイス
    - フィルタルールの記述
    - キーワード登録
    - 透過率の設定
      - 全体の情報流量のうちデバイスに再配送する文書の割合
    - 最大容量の設定

## 実装済

- BoWマルチクラス分類器
  - 設定された透過率から類似度の閾値を推定
  - 各Bag-of-Words類似度分類器にMapする
- 送信キュー
- サーバ/クライアント間プロトコル
  - シリアライザ/デシリアライザ

## これから

- HTTPプロキシ
  - cSorterの実装を流用
- クライアント側ストレージ設計
- 以上を貼り合わせ

# まとめ

- 社会的コンテキストを考慮した情報再提示
  - 多中心
    - 様々なデバイス(CPUパワー、ストレージ容量)に対応
    - 様々な接続状態(帯域、可用性、安定性)に対応
  - 非同期
    - 内部的に多中心な構成をしたサーバがデバイス間配送を中継
    - HTTPサーバで文書を捕捉し、ネットワーク帯域に余裕がある時にサーバと同期する
  - マルチクラス分類器
    - 情報デバイスと社会的コンテキストの相関に依拠
    - 最小限の操作、最小限の設定でそれなりの分類
    - 色々なサイズの文書を色々な文書サイズのクラスにそれなりに分類

# Pcard

(Poly-Centric Asynchronous Re-distribution of Data)

- Poly-Centric
  - 各情報デバイスが中心
- Asynchronous Re-distribution of Data
  - デバイスAの情報をデバイスBに配送する時、両方が同時にオンラインである必要がない

## 今後の課題

- 実装、評価
- 各情報デバイス内でのコンテキストへの分類
  - 社会的コンテキスト、トピック的コンテキスト、…
- 再提示の方法
  - 様々な物理的/機能的制約のあるデバイスでの、タスクの妨げにならない情報再提示
  - 正しい時に、正しい方法で、正しい情報を再提示