

# サーバの復習

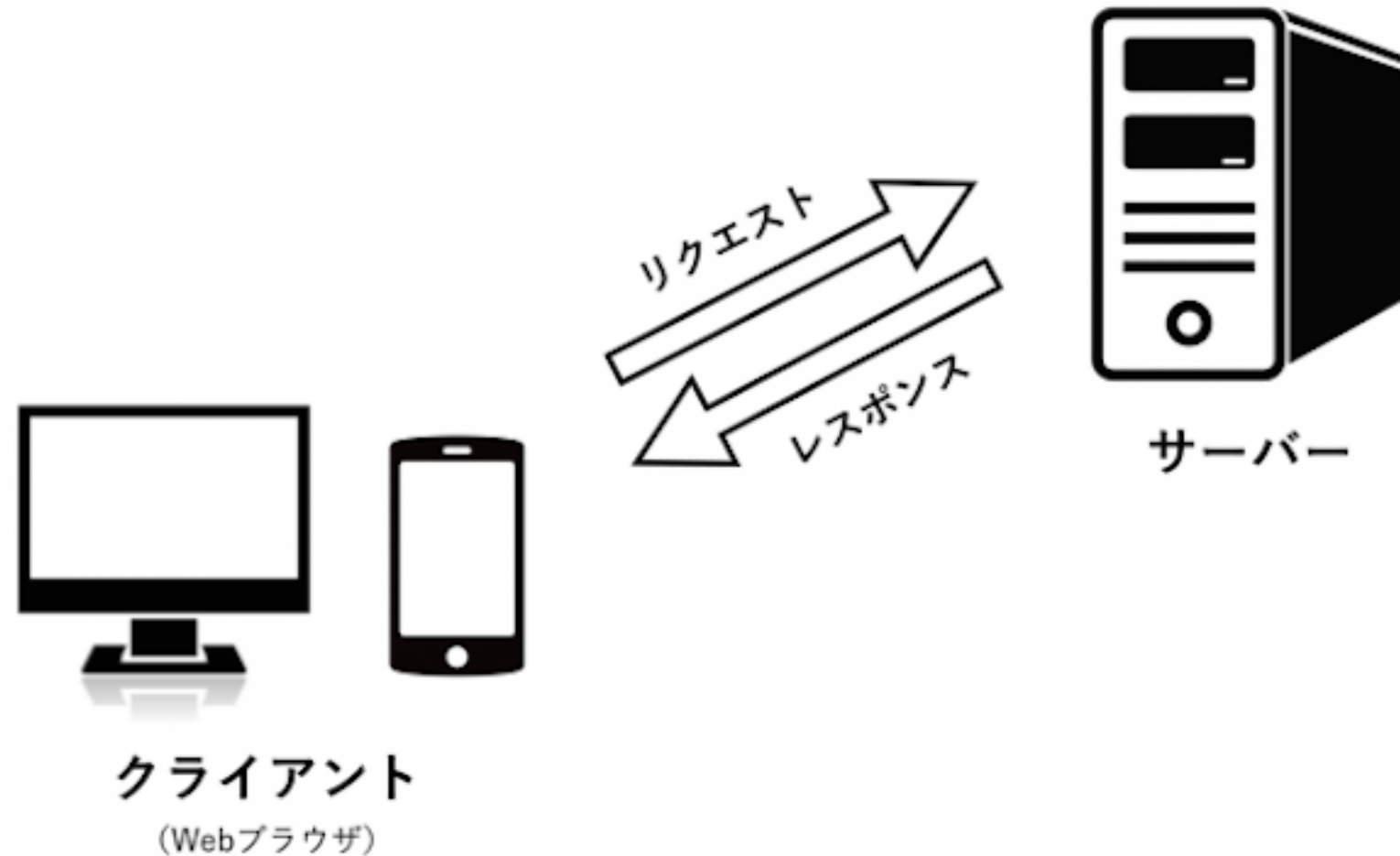
2015-06-07

講師: 猪俣 充央

# 改めて、サーバって？

- ・ サーバは、インターネット上など、クライアントから繋がる場所にあるアプリやソフトウェアのことです。
  - ・ サーバのソフトウェアはどこかのPCや端末上で動いているものなので、場所自体をサーバマシン、サーバ端末と呼びます。
  - ・ 今流行のクラウドやAWS、VPS、Google App Engine などもちっと違いますがサーバマシンのようなものだと思います。
- サーバソフトやサーバマシンをひっくるめて「サーバ」と呼びます。

# サーバとクライアント



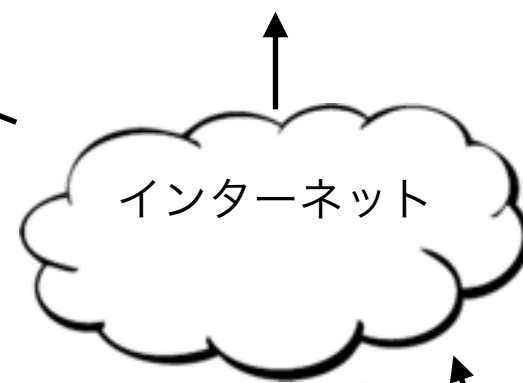
# サーバとクライアントは 世界中にたくさんいる

<http://google.co.jp>

<http://yahoo.co.jp>

<http://rakuten.co.jp>

サーバ



クライアント

# Webサイト・Webサービス にはサーバが必要

- ・ Yahoo, Google, 楽天, Amazon, Twitter, Facebook, アメーバブログなどなど、ブラウザから見られるすべてのWebサイト(Webページ)、Webサービスはサーバ上に置かれて、運営されています。

# アプリから見て、 どういう時にサーバが必要なのか

サーバにはいろいろな役目があります。

- ・ クライアント同士の通信を仲介する
- ・ 最新の情報、ニュースをクライアントに提供する
- ・ 端末が変わってもデータが消えないように保存しておく、複数環境でデータバックアップや同期をする。
- ・ などなど . . .

# サーバが必要な例(1)

## クライアント同士の通信を仲介する

- ・ 離れた相手ともチャット、SNS、ゲームのフレンド機能が使えるようにしようと思った時、どうすればいいでしょうか？
- ・ 例: LINE、Twitter、Facebook、パズドラ、モンスターなど。

# サーバが必要な例(1)

## クライアント同士の通信を仲介する

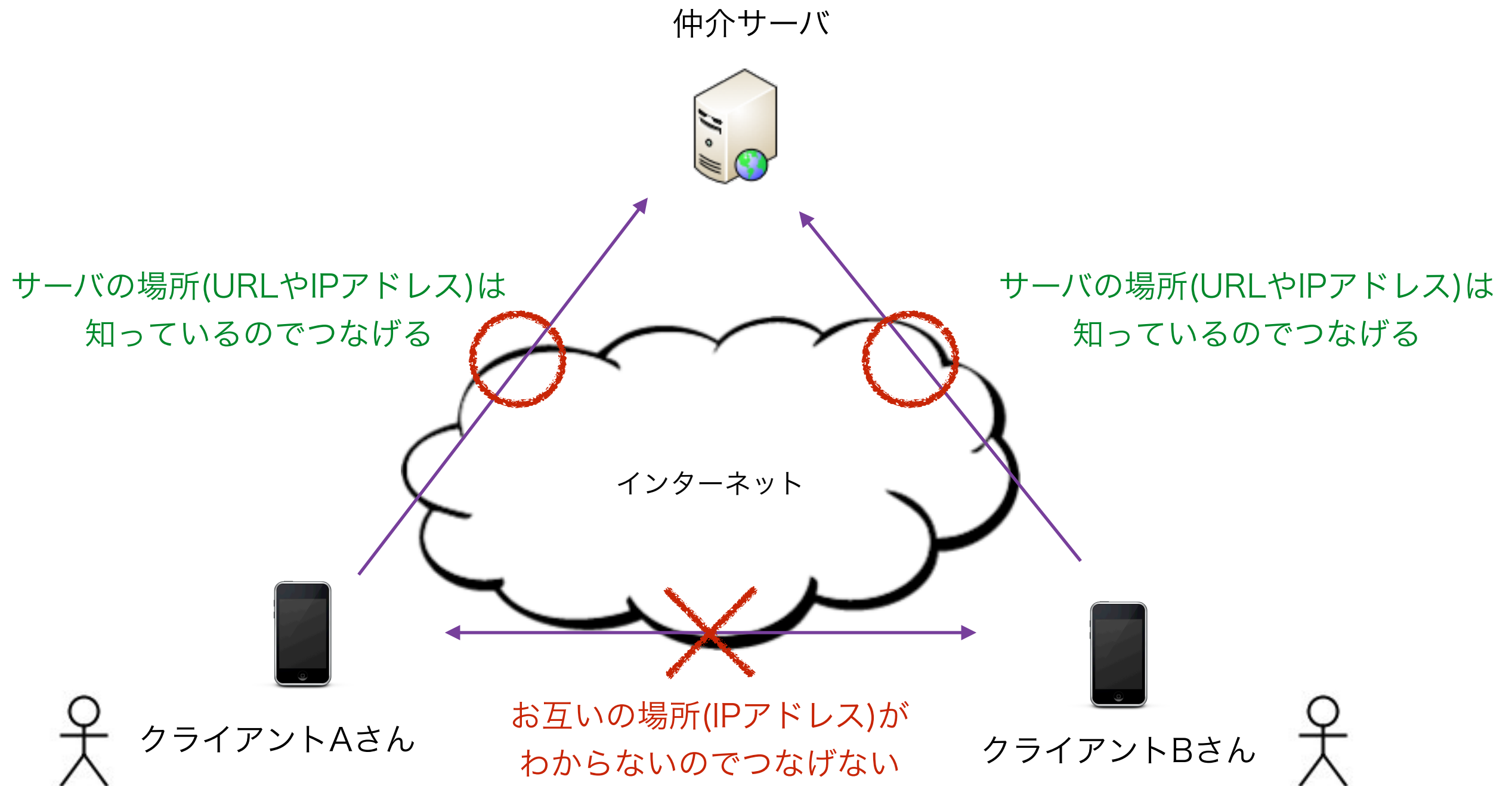
- ・ 相手との距離にかかわらず、いつでもどこでも相手と通信したいと思った時は、インターネットを使うのが現実的です。
- ・ 近くにいるときだけ使えればいいのなら、BluetoothやWi-Fi Directなどの別の選択肢もあります。
- ・ でも、家に帰ってから友達と連絡したい、大阪と沖縄で通信したい、日本とアメリカでも通信したい…となると、繋がっていれば場所を気にせず使えるインターネットが便利です。



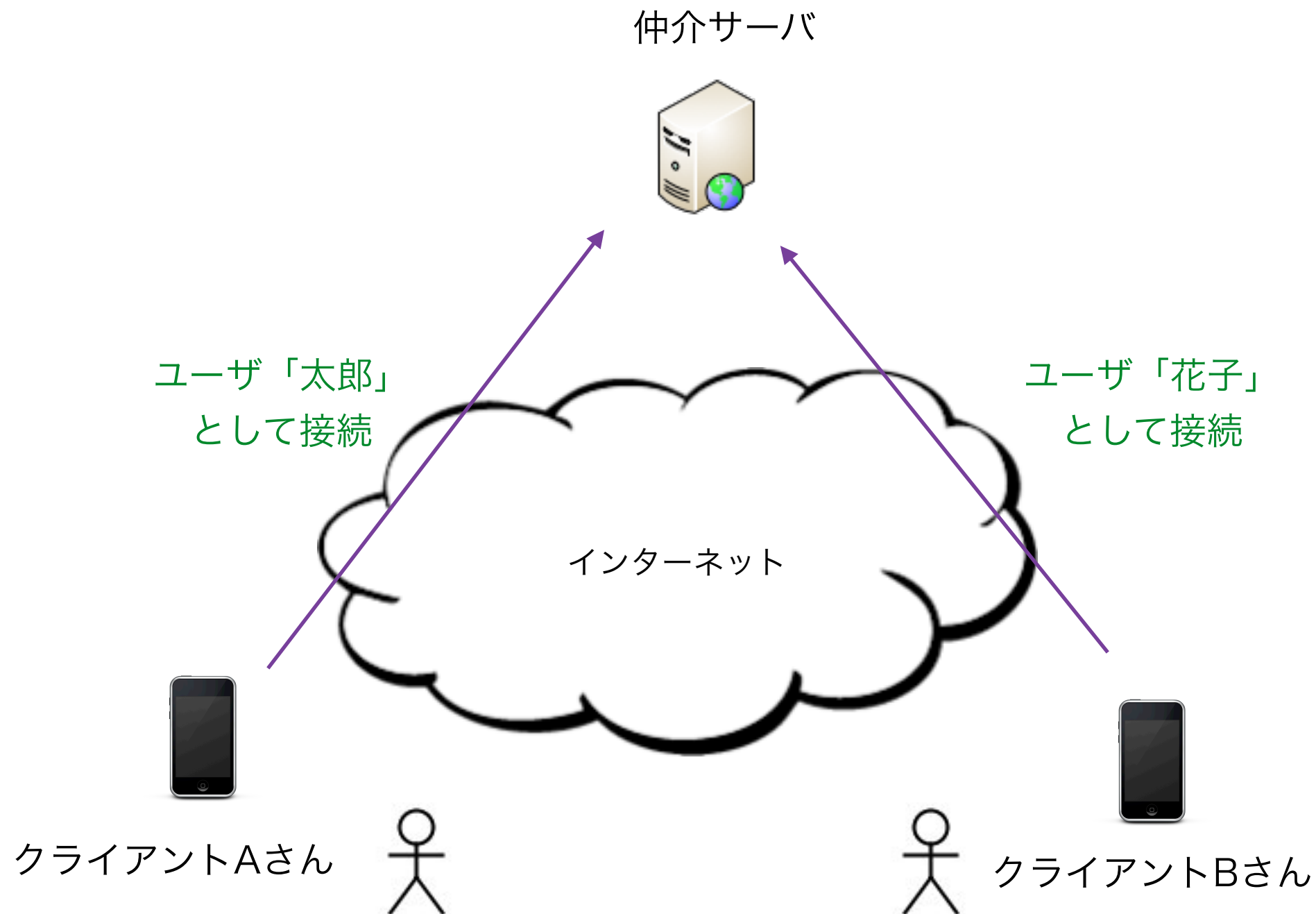
# クライアント同士で通信する のになぜサーバが必要？

- ・ 相手とクライアント同士でお互いのインターネット上の場所(IPアドレス)を知る必要があります。
- ・ しかし、携帯電話や一般的なコンピュータのIPアドレスは固定でなく動的に割り当てられるので、お互いに相手のIPアドレスを特定できません。
- ・ サーバのIPアドレスは通常固定されている(特定しやすくなっている)ので、クライアント同士はサーバに仲介してもらうことで相手の場所を特定して通信可能になります。

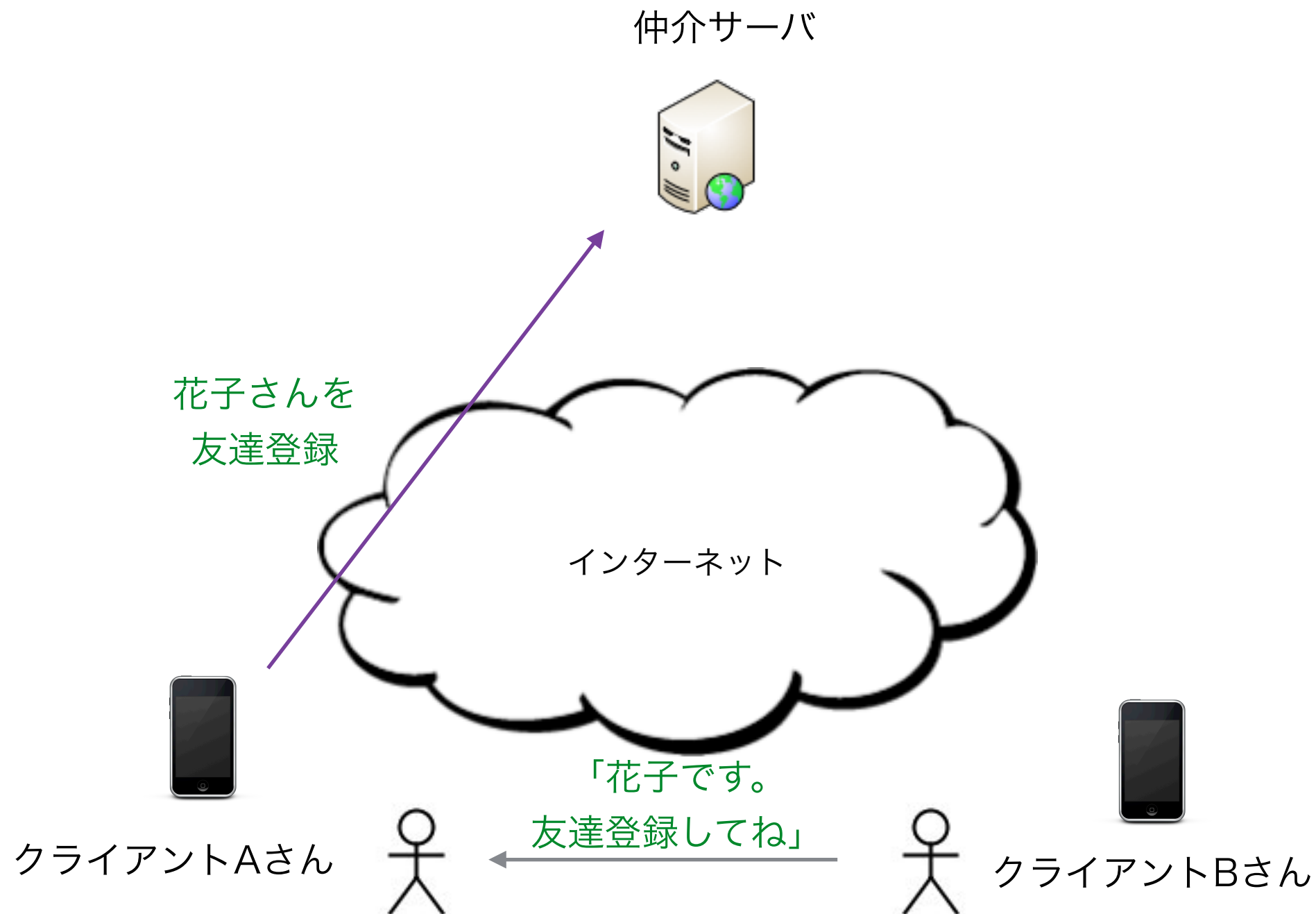
# サーバを経由した クライアント同士の通信



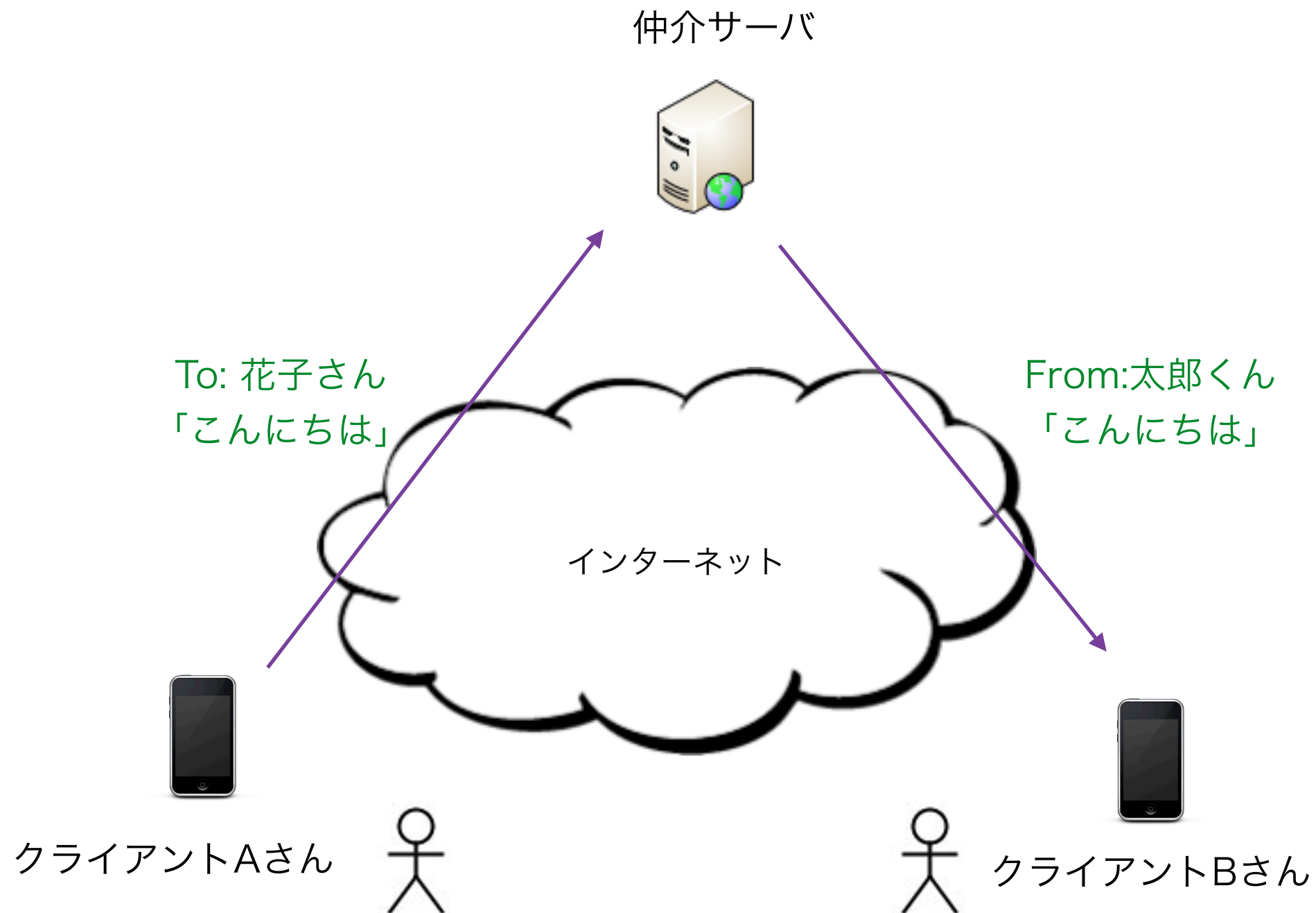
# サーバを経由した クライアント同士の通信



# サーバを経由した クライアント同士の通信



# サーバを経由した クライアント同士の通信



## サーバが必要な例(2)

### 最新の情報をクライアントに提供する

- ・ ゲームのキャンペーン・イベント更新をしたいとき、その他アプリでサービス運営側からのお知らせを出したいときなどにもサーバが使われている事が多いです。
- ・ でも、情報を更新する時に毎回Google Playで自分のアプリをアップデート(再配信)して、最新の情報に差し替えればサーバは必要ありません。

# サーバが必要な例(2)

## サーバがないと困るパターン

- ・ 1回、2回など、頻度が少ないならアプリのアップデートをすればいいのですが、たとえば週に1度イベントがあって、毎週お知らせの更新が必要だとするとどうでしょう？
- ・ 毎週更新が必要だと配信までの作業に手間がかかりますし、ユーザに負担をかけてしまいます。
- ・ ユーザによってはアプリが手動更新になっていて、いつまでたってもアップデートしてもらえないかもしれません。

# サーバが必要な例(2)

## サーバがあれば…

- ・ アプリ向けに情報提供をするサーバがあると、アプリ更新がなくてもアプリが起動するたびに通信して、常に最新の情報を表示することができます。
- ・ イベント情報、キャンペーンなど、頻繁に更新する情報がアプリ内にあるとわかっている場合、サーバから取得するようにしておくと便利です。



# サーバからの情報更新

2015年6月7日

お知らせ配信サーバ



頻繁に変わるお知らせ  
にもアプリ更新なしで  
対応できる。

明日から〇×キャンペーンです！

インターネット

新しいお知らせない？

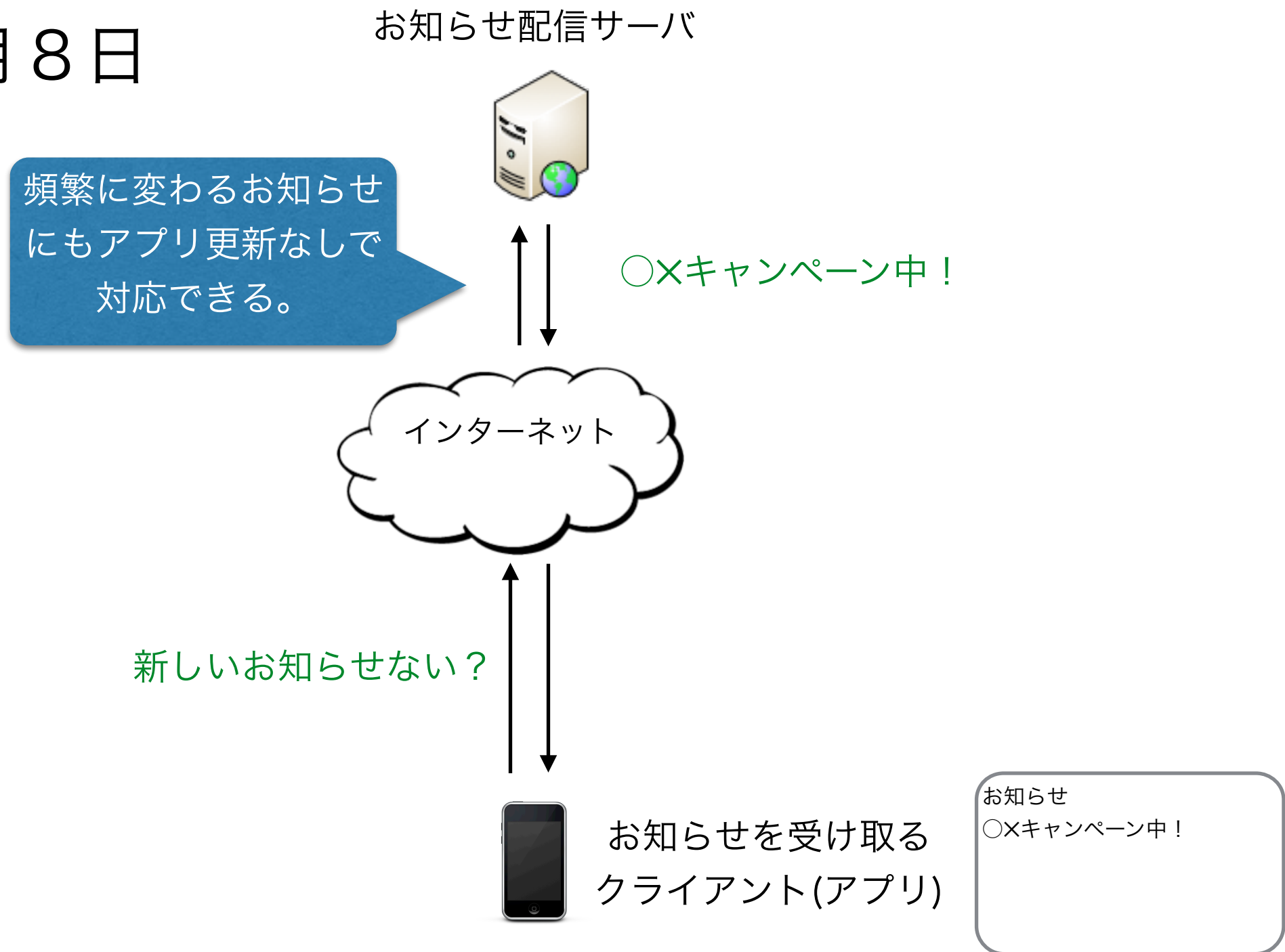


お知らせを受け取る  
クライアント(アプリ)

お知らせ  
明日から〇×キャンペーンです

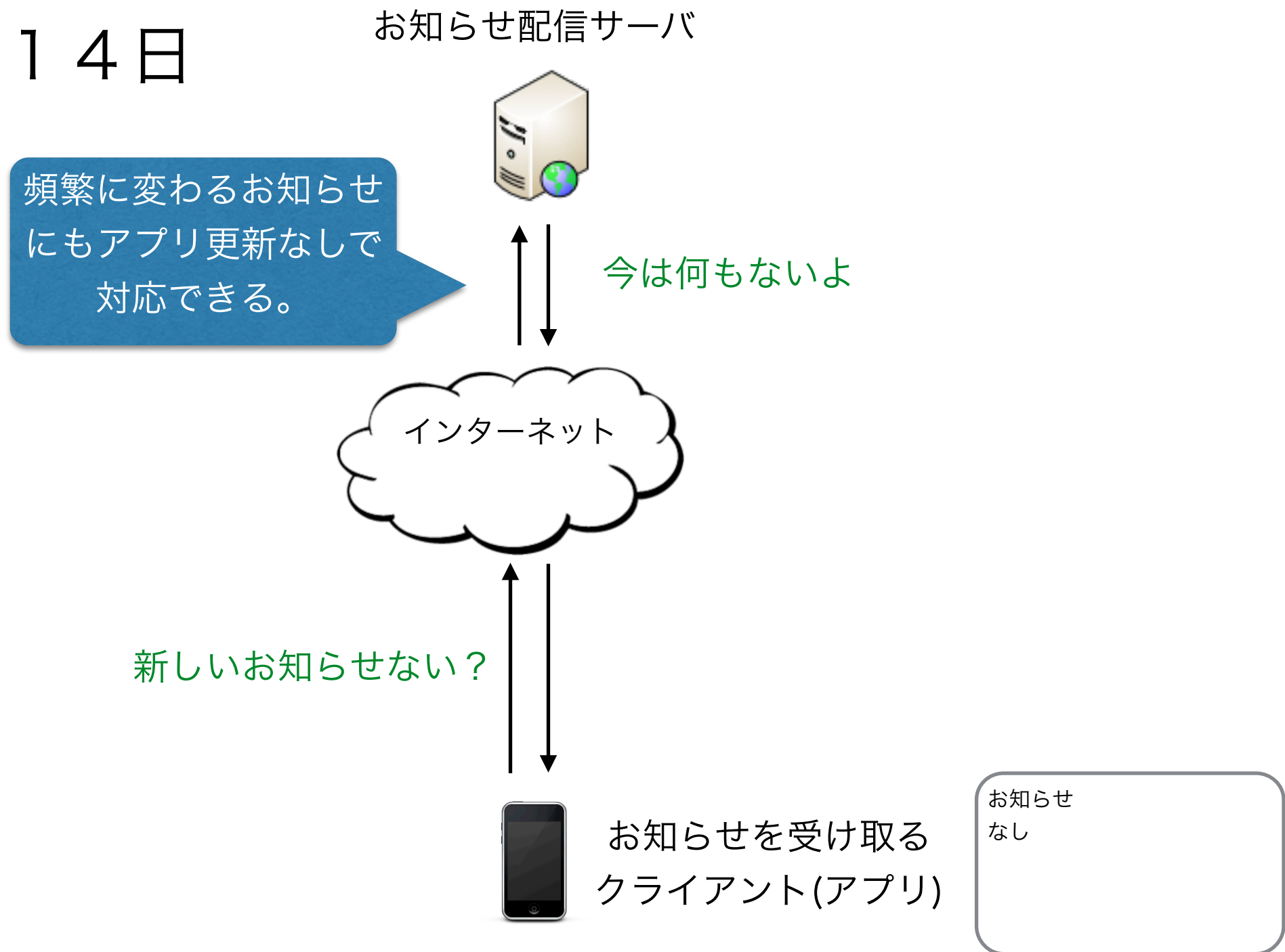
# サーバからの情報更新

2015年6月8日



# サーバからの情報更新

2015年6月14日

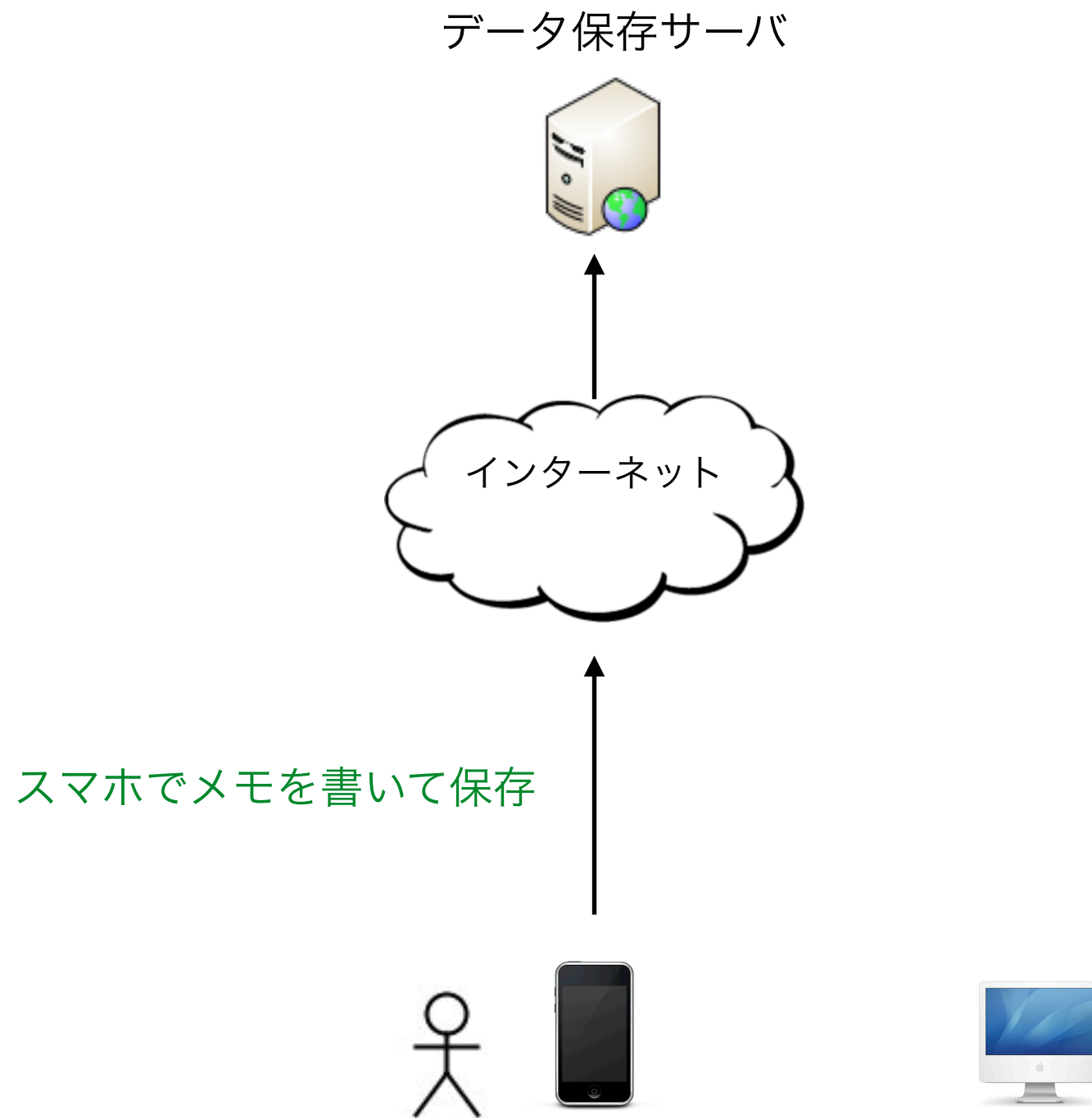


# サーバが必要な例(3)

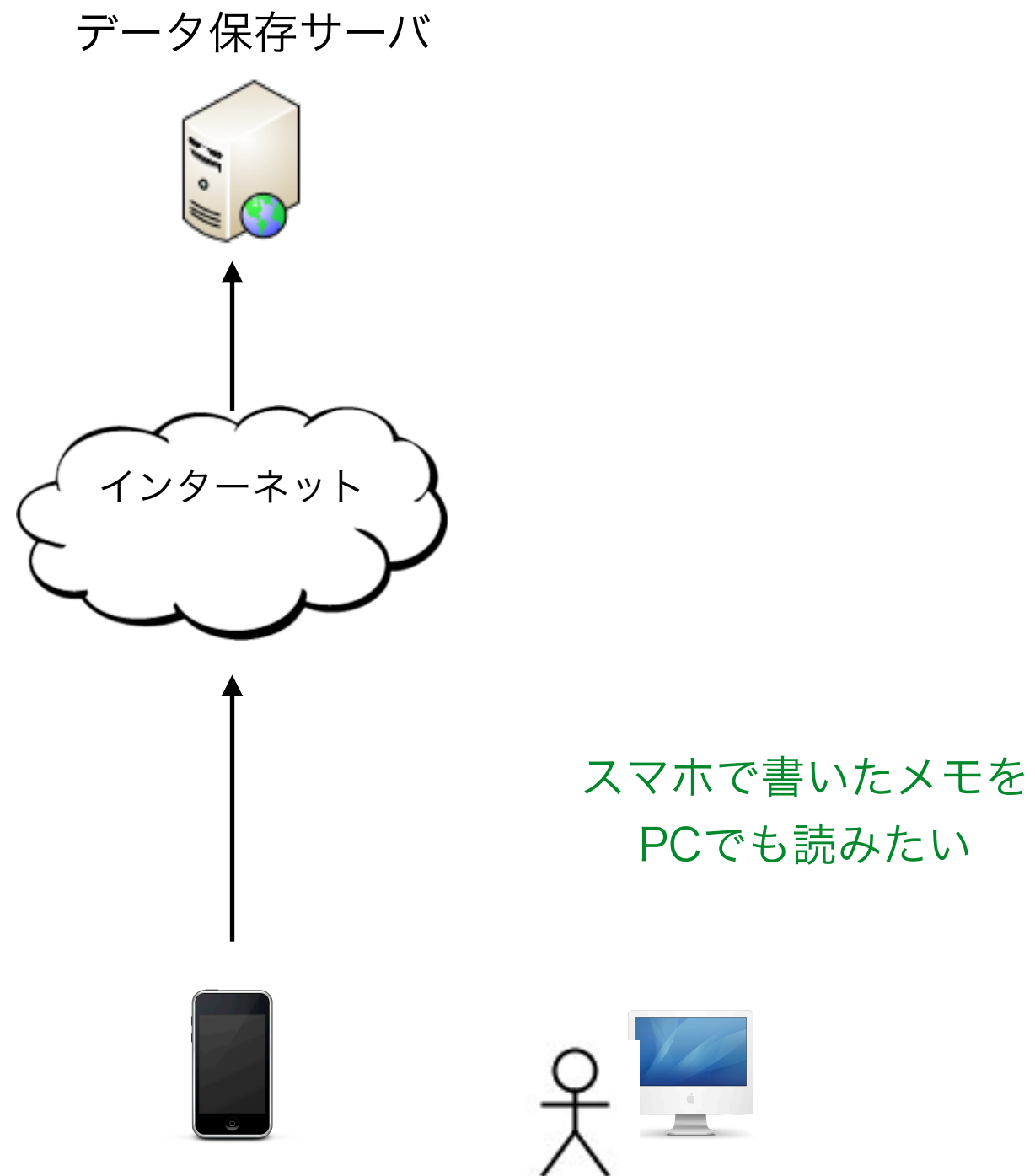
## 端末同士のデータ同期

- ・ 自分の持っているPCやスマートフォン、タブレットでデータを共有(同期)したい時もサーバが使われることが多いです。
- ・ また、機種変更などで端末が変わった場合に新しい環境にデータを引き継ぐ機能があるアプリでは、サーバに保存しているデータからアプリの情報を復元します。
- ・ 例: DropBox, Google Photos(Picasa), ゲームなどの機種変更機能

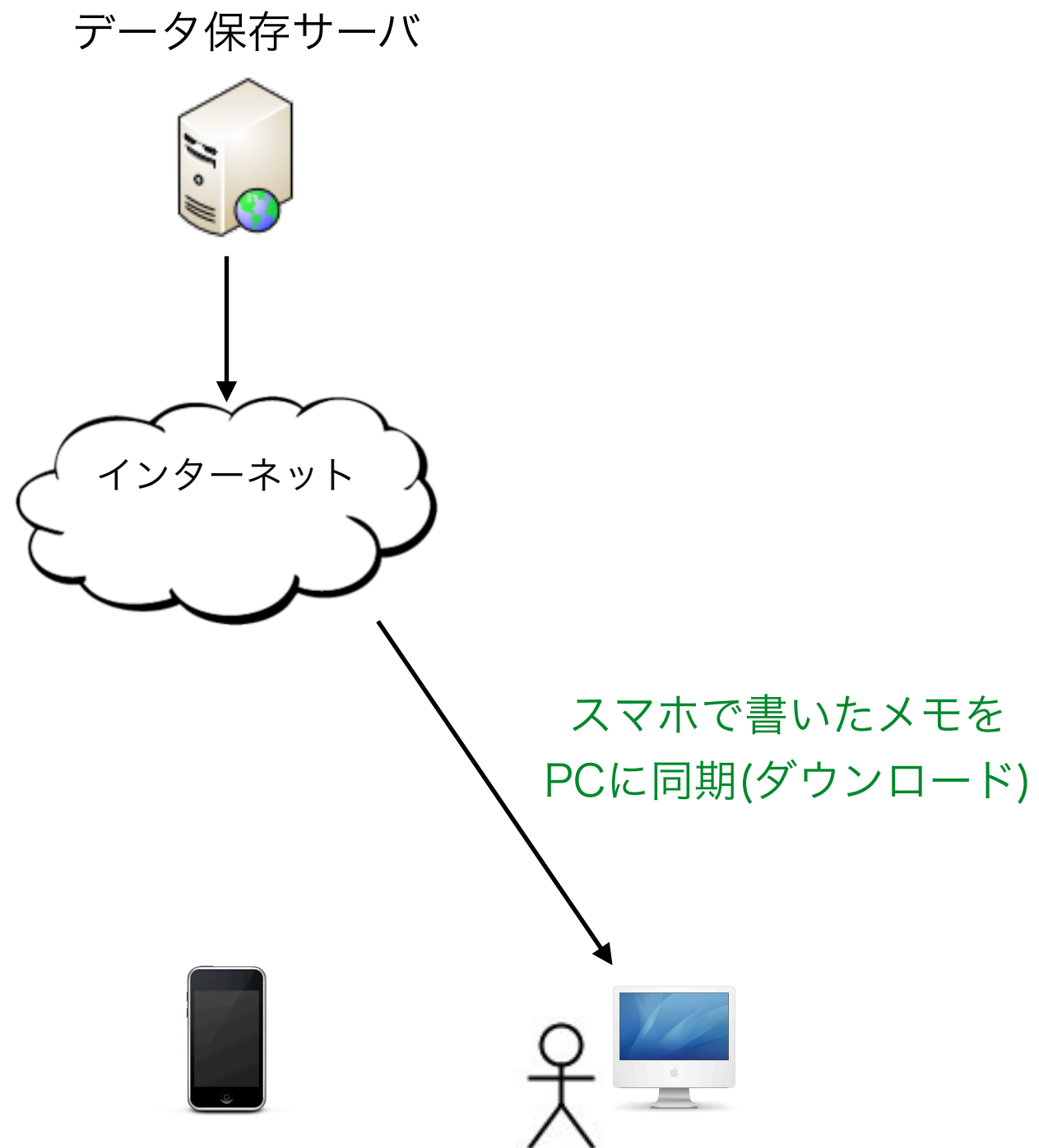
# データ同期



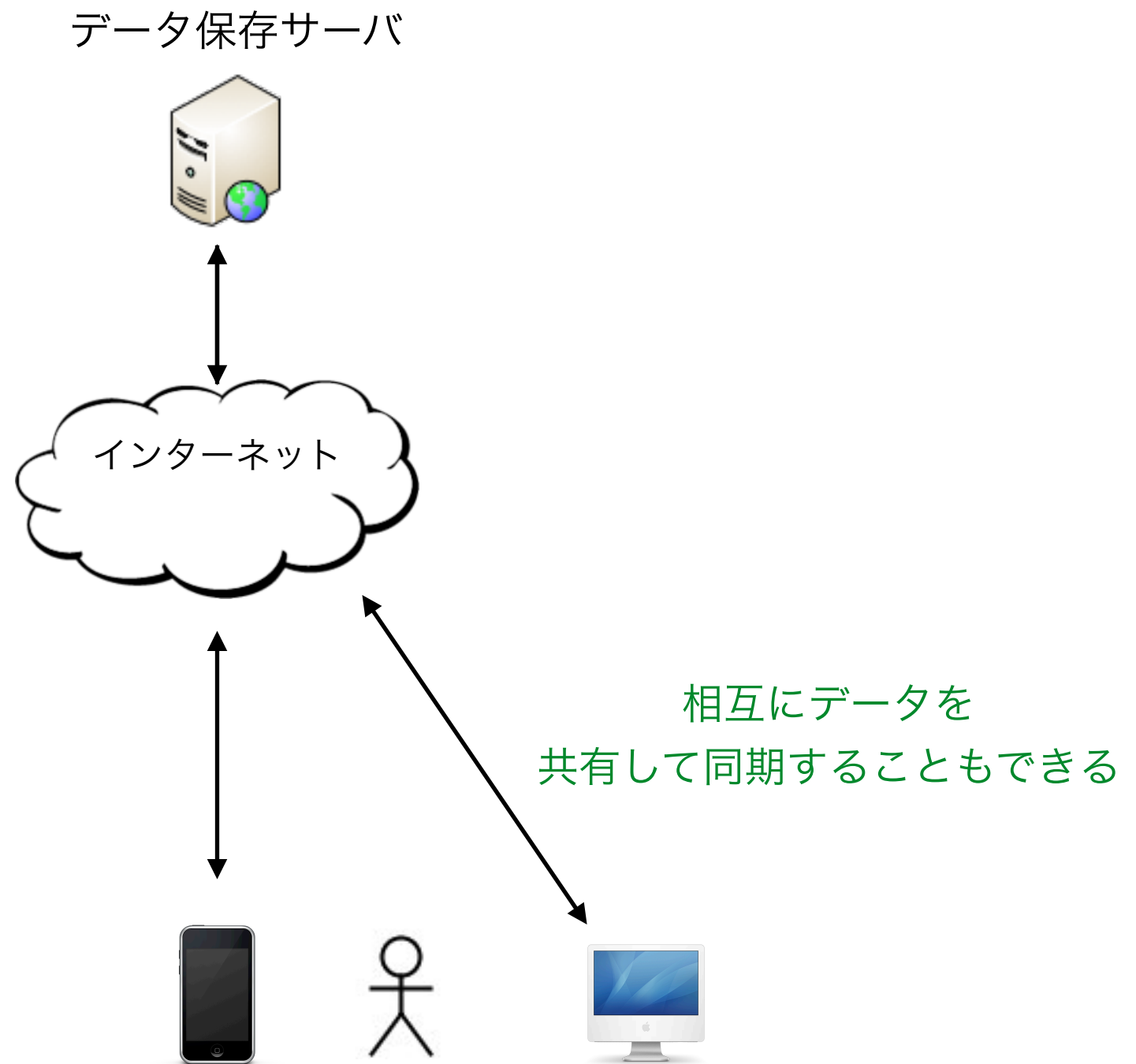
# データ同期



# データ同期

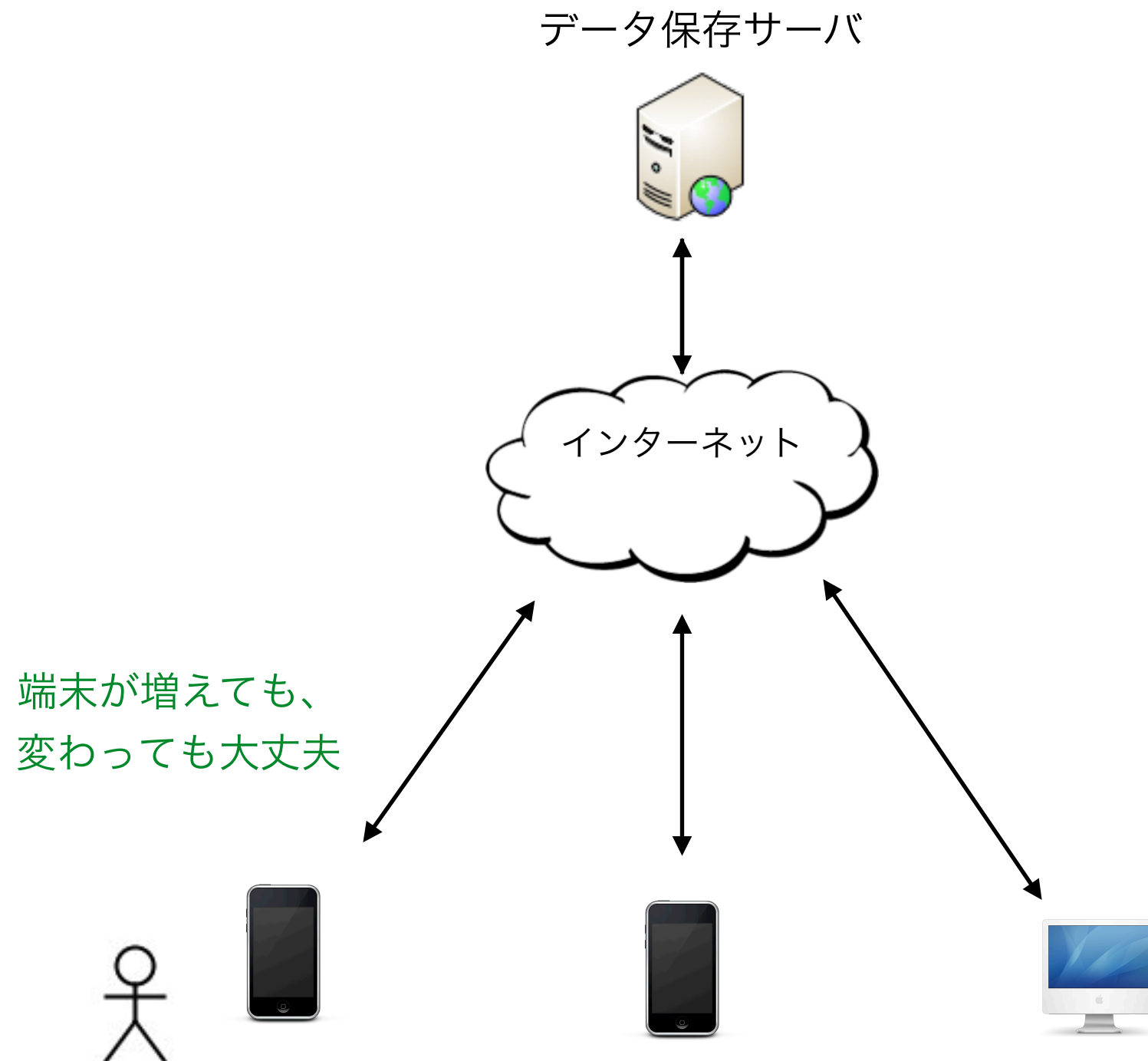


# データ同期





# データ同期



# サーバにデータを保存する？

- ・ 今まで挙げた例のように、サーバ上には様々なデータを保存する場所が必要になります。
- ・ ユーザのアカウント情報  
(メールアドレス、パスワードなど)
- ・ チャット、メッセージなどの履歴、タイムライン
- ・ アプリから送られた写真などのファイル

# どうやって保存するか

- ・ サーバ上でデータを保存するには大きく分けて、2種類の方法があります。
- ・ データベースへの保存
- ・ ストレージ(ハードディスク等)へのファイル保存

# 1. データベースへの保存

- ・ データベース(DB)とは、何百万件ものデータでも一度に取り扱えるように作られたシステム(ソフトウェア・ライブラリ)のことです。
- ・ データベースを取り扱うためのサーバソフトウェア(またはマシン)を、データベースサーバと呼びます。

# データベースとSQL

- ・ データベースでは、**SQL**(エスキューエル)というデータ処理用の言語がよく使われます。(SQLを使えるデータベースを**リレーショナルデータベース(RDB)**とも呼びます)
- ・ SQLが使えるデータベースサーバソフトウェアには以下のようなものがあります。

**MySQL**、**Google Cloud SQL**、PostgreSQL、Oracle、Microsoft SQL Server

また、サーバソフトウェアの不要なローカルファイル式のシンプルなデータベースとして、**SQLite**があります。

(Androidから利用できます)

# SQLの例

— user テーブル作成

```
CREATE TABLE user  
(id INTEGER NOT NULL,  
 name VARCHAR(255),  
 email TEXT,  
 updated TIMESTAMP);
```

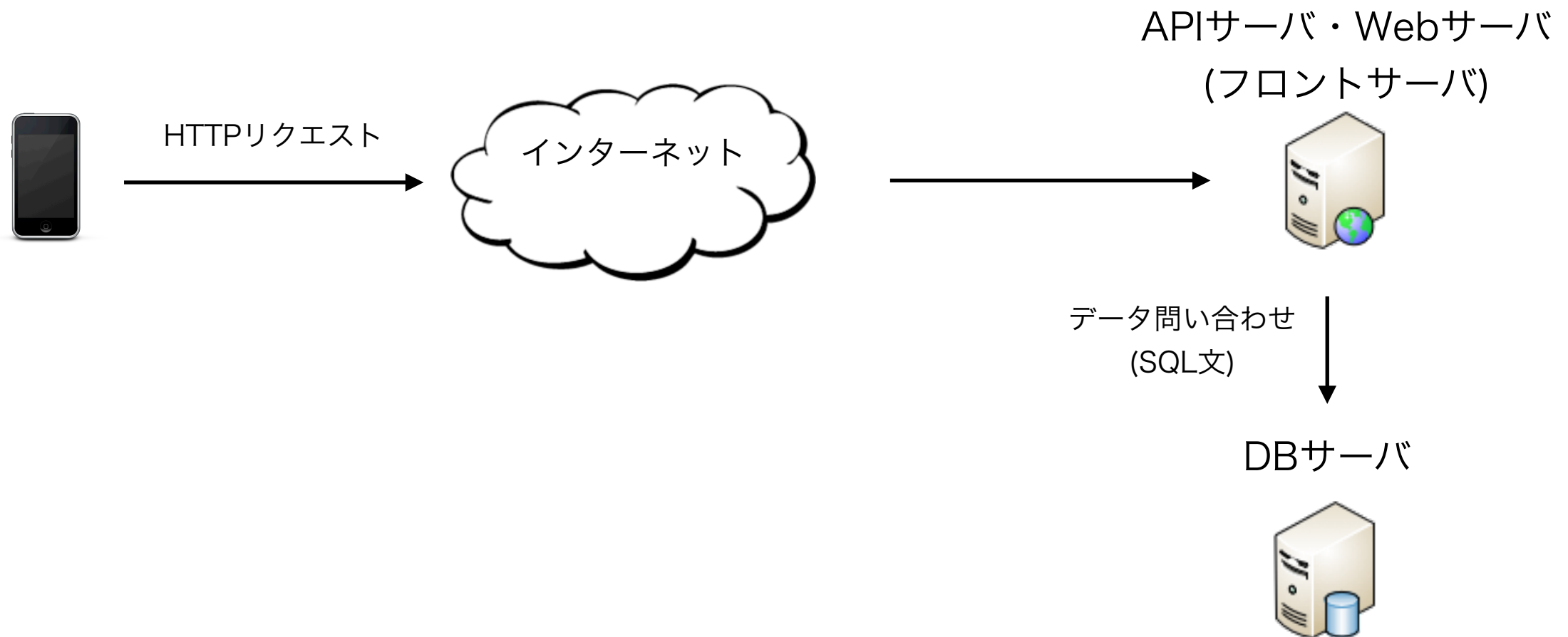
— user テーブルへ2レコードのデータを投入

```
INSERT INTO user (id, name, email) VALUES  
(1, '山田太郎', 'yamada@example.com'),  
(2, '田中一郎', 'tanaka@example.com');
```

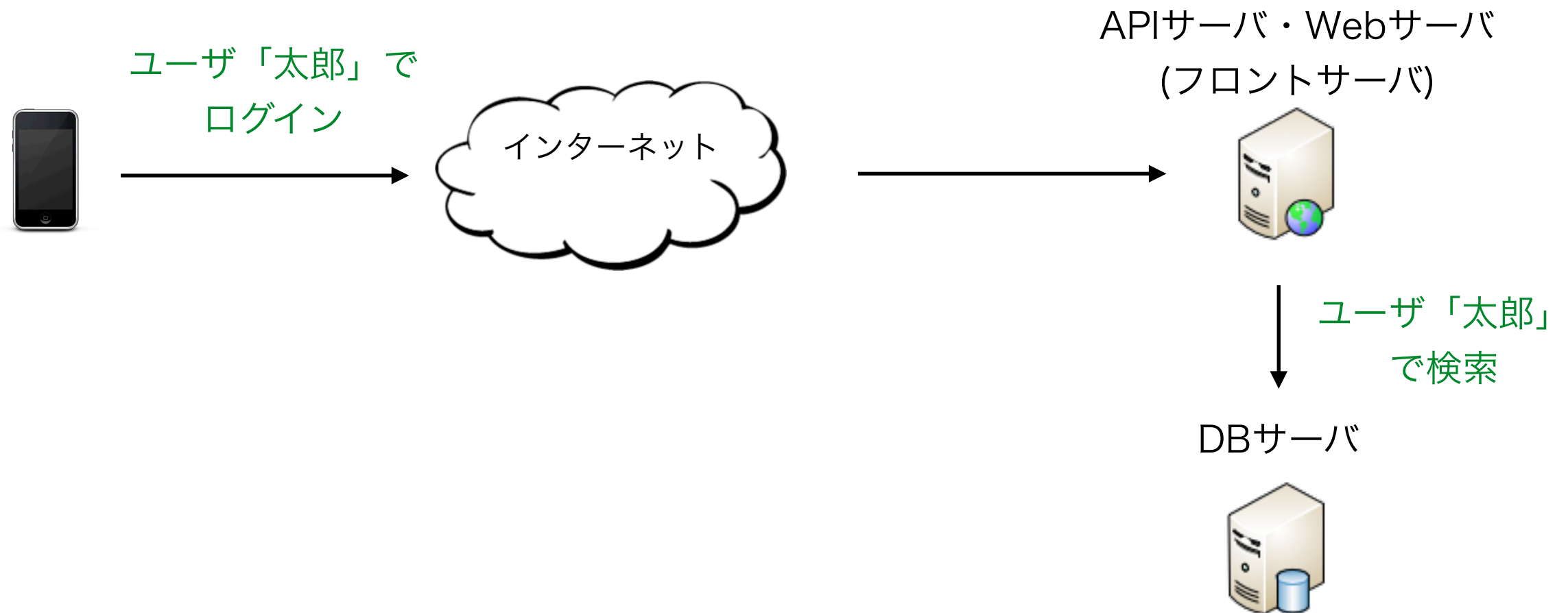
— userからid=2のユーザ情報(田中一郎)を取り出す。

```
SELECT * FROM user WHERE id = '2';
```

# データベースの流れ

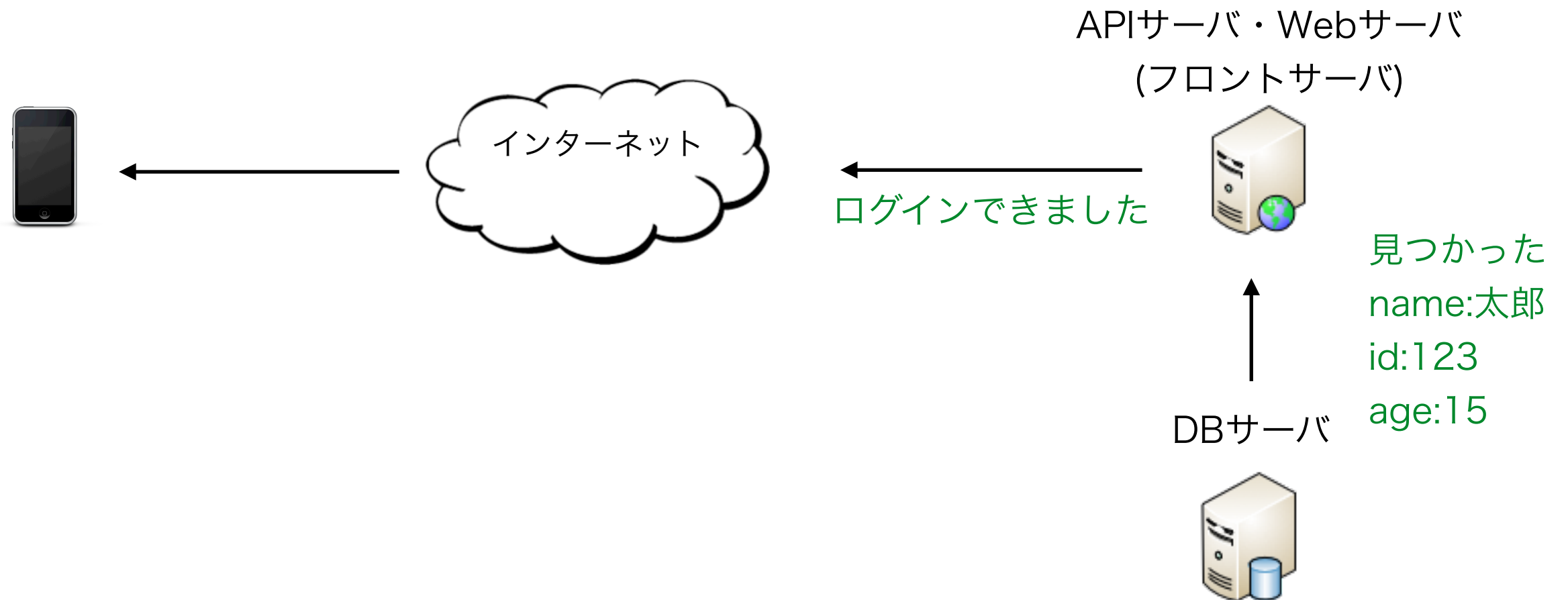


# データベースの流れ

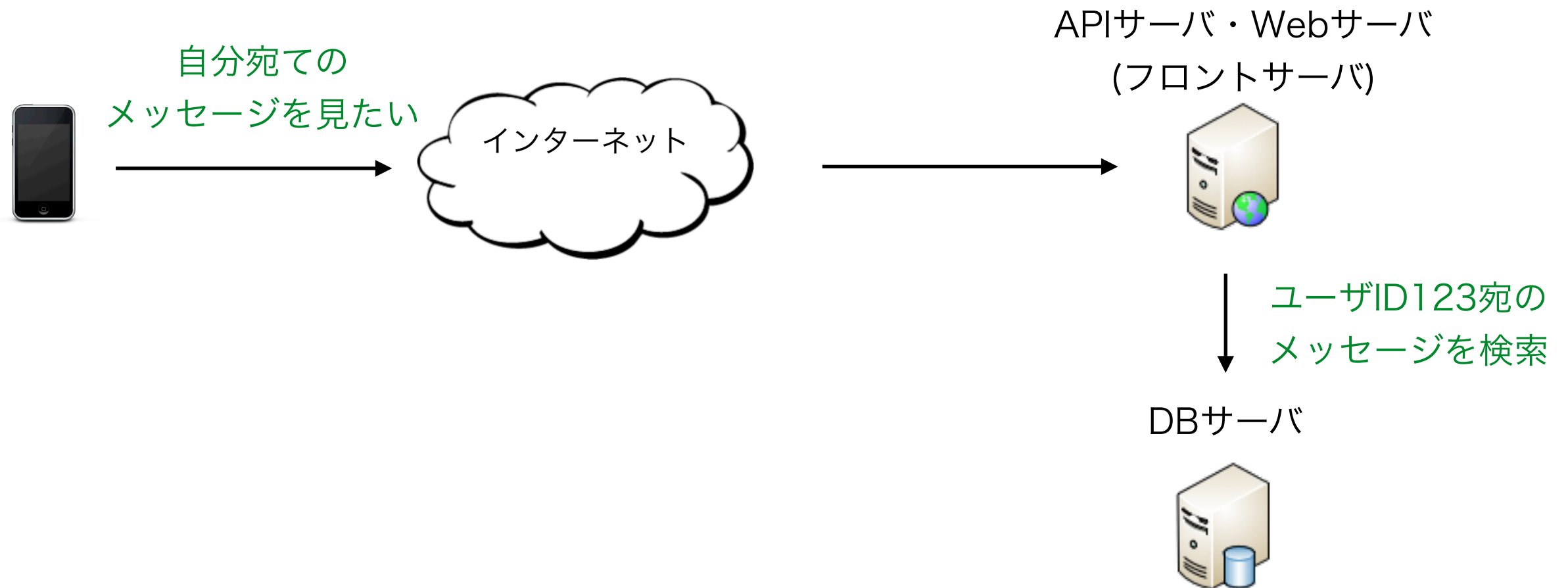




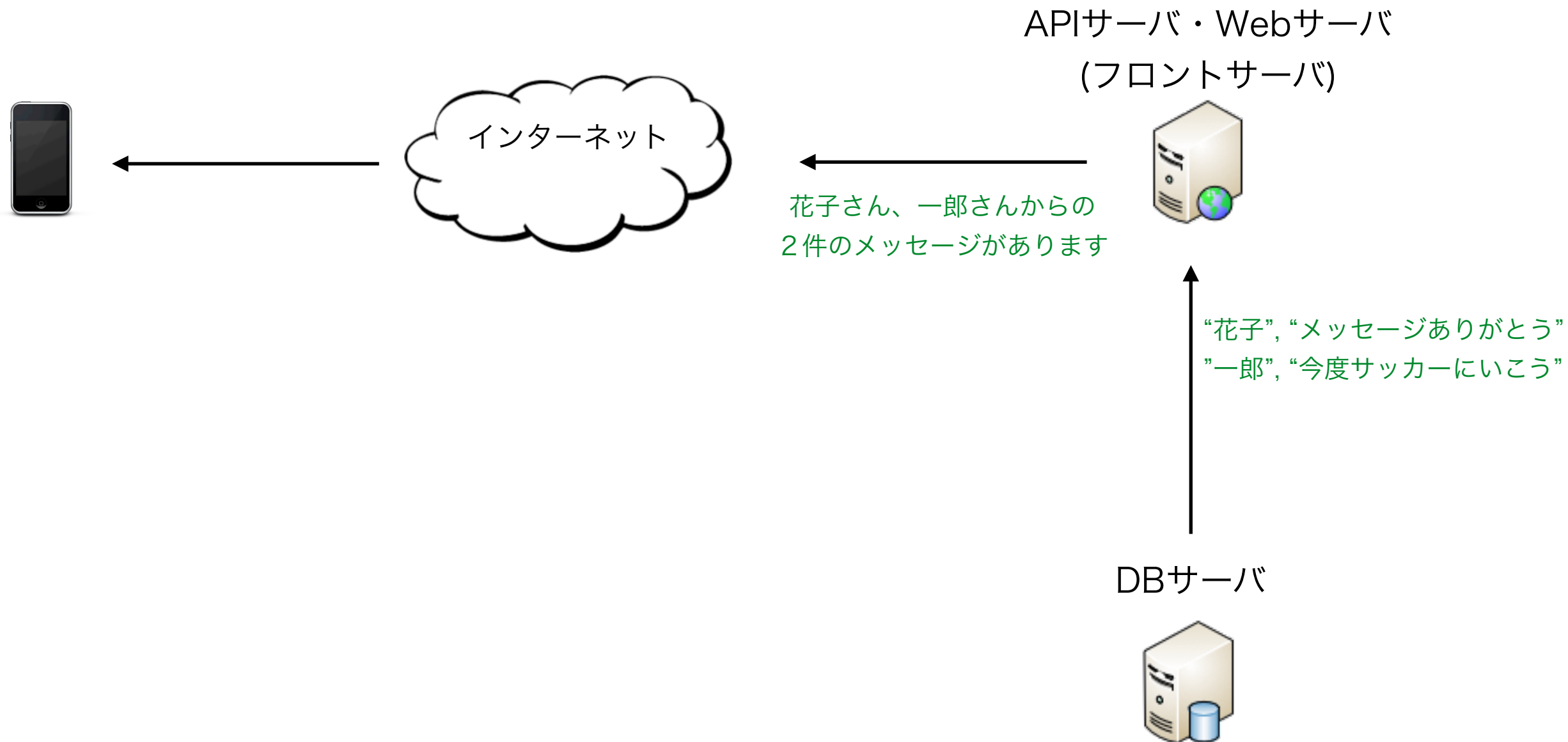
# データベースの流れ



# データベースの流れ



# データベースの流れ



# データベースでできること

- ・ テキストで処理できるデータ(文字列、日付、数値など)を効率よく保存して、素早く検索、集計ができます。
- ・ 大量のデータを保存、更新する事ができます。
- ・ 何百万件もデータがあっても一瞬で検索できたりします。  
(例: 100万人ユーザの居るサービスで、1ユーザあたり平均1000通のメッセージを誰かに送っていても、ユーザの全部のメッセージをDBに記録することができます)
- ・ SQL文を書くことで、複雑な条件でも簡単に検索することができます。  
(例: 2014/10/01～2014/10/31までの間にアプリを使い始めて、今もアプリを使っているユーザの数とメールアドレス一覧を表示したい)

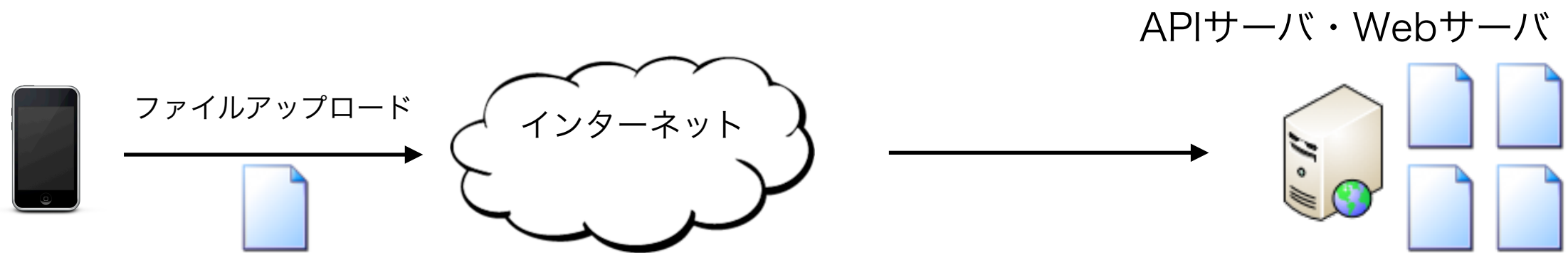
# ストレージ

- ・ ストレージとは、ファイルなどを保存できる領域のことです。  
※セキュリティの話で出てきた拡張ストレージ、専用フォルダなどもストレージです。
- ・ サーバ上で画像ファイルなどを保存する必要がある場合に使います。基本的に、サーバマシンにあるハードディスク(クラウド・VPSの場合は仮想的なハードディスク)をストレージとして使用します。
- ・ Google App Engine、Heroku、PagodaBox など、PaaSと呼ばれるサーバ環境にはサーバ自体にストレージがないため、別のストレージサービスが必要になることもあります。

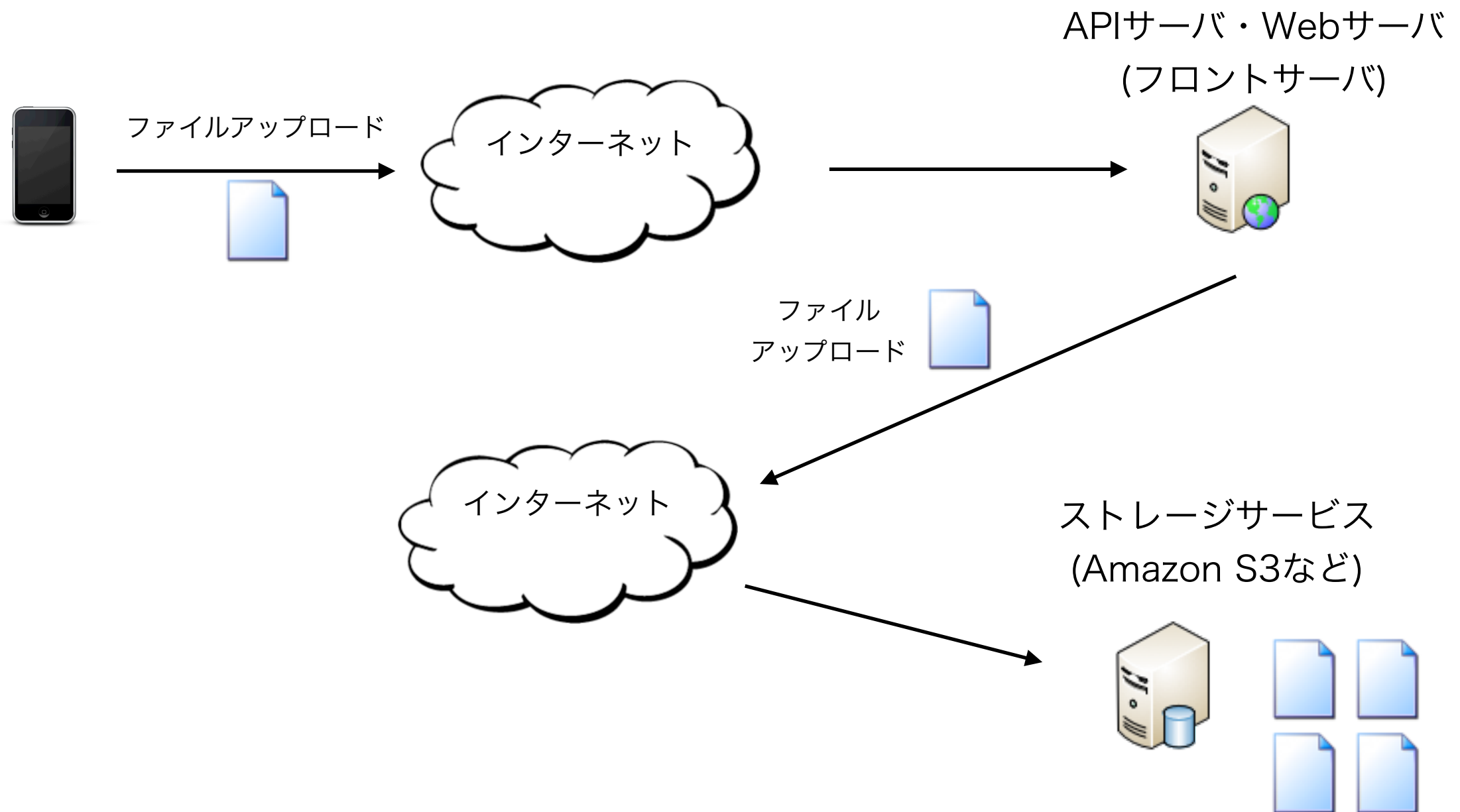
# ストレージサービスの例

- ・ Google App Engine
  - ・ Blobstore、Google Cloud Storage
- ・ その他のPaaS(Herokuなど)
  - ・ Amazon S3

# サーバ上に ストレージがあるケース



# サーバ上に ストレージがないケース





# サーバを自分で作らないといけないもの？

- ・ わかった、自分のアプリにはサーバが必要だ。
- ・ サーバが必要だったら、自分で絶対にサーバ側でプログラミングしないといけない？  
→そんなことはありません。
- ・ 調べてみると、欲しい機能を提供しているWebサービスがあったり、すでに便利なWeb APIが提供されていることがあります。
- ・ 既存のサービスやWeb APIを賢く使うのが、アプリ実現への近道です。

# 既存サービスが使えないか 調べよう

- ・ Google Play Services  
(ゲーム関係、課金システムなど)
- ・ <http://dev.classmethod.jp/smartphone/android/android-google-play-services-1/>
- ・ Twitter, Facebook, Googleアカウント (Google+)  
(ログイン、ソーシャル共有、友達との連携)  
それぞれのサービスにSDK(ライブラリ)があります。
- ・ その他にもSDKが提供されていたり、サービスを活用するAPIがあるサービスがたくさんあります。

# APIが使えないか調べよう

- ・ マッシュアップアワード API一覧ページ
  - ・ <http://mashupaward.jp/apis>
- ・ Web制作に役立つAPI一覧(2015年度)
  - ・ <http://webadviser.org/web-tool/web-api-2015/>

# APIの注意(復習)

- ・ 無料で使えるものと有料で使えるものがあります。
- ・ 接続回数の制限があることもあります。
- ・ 利用規約で商用禁止になっていることもあります。
- ・ 制限事項、利用規約をちゃんと読んで、自分のサービスに使えるか、検討しましょう。

# まとめ

- ・ アプリにサーバが必要になるケースは結構あります
- ・ ただし、必ずしも自分で作る必要がないこともあります。
- ・ みなさんが作りたいアプリはサーバが必要か、考えてみましょう。
- ・ サーバが必要な場合、既存のサービス、Web API を使って楽ができないか調べてみましょう。