

Canoe

通信量負担分散プロジェクト

3年 和田健吾

なぜ作ったのか

- スマホの7GB制限
- なぜスマホが流行ったのか？についての疑問

スマホの7GB制限

- 通信プランが月どれくらいデータ通信できるか決めている
- 使わない人も使う人も一律に7GBの料金を支払う
- それなら使う時、使わないときによって
必要な人に通信容量を渡したい

なぜスマホは流行ったのか

- 災害の安否確認にキー
- 基地局は電話が殺到→処理能力を超え利用困難に
- スマホは公衆WIFI→Twitter/Mixi/Facebookによる安否確認(可)

この違いがスマホの普及へ

2 テーマから浮かび上がる共通点

通信量・負担の分散

そこで生まれたCanoe

2:10

あなたのID

15CD6C1C-956B-4671-8E8D-55587F6C36CB

ダウンロードする対象のURL

<https://sample-videos.com/img/Sample-png-i...>

移譲先IP

<https://waterserver1.herokuapp.com>

ログ

40055B89-AE74-4DAE-9017-634EAFE30C84と接続されました
EE9E91B8-E0AA-4B94-9CFE-432EE98E3A1Cと接続されました

ダウンロードを要請する

2:12

あなたのID

15CD6C1C-956B-4671-8E8D-55587F6C36CB

ダウンロードする対象のURL

<https://sample-videos.com/img/Sample-png-i...>

移譲先IP

<https://waterserver1.herokuapp.com>

ログ

40055B89-AE74-4DAE-9017-634EAFE30C84と接続されました
EE9E91B8-E0AA-4B94-9CFE-432EE98E3A1Cと接続されました
5092141c-6db9-49c4-9e7a-b31d99cba355
最大:4
現在:1
残り3ダウンロード必要
最大:4
現在:2
残り2ダウンロード必要
最大:4
現在:3
残り1ダウンロード必要
最大:4
現在:4
残り0ダウンロード必要
ファイル名: Sample-png-image-30mb.pngのダウンロードが完了しました
40055B89-AE74-4DAE-9017-634EAFE30C84との接続が解除されました
EE9E91B8-E0AA-4B94-9CFE-432EE98E3A1Cとの接続が解除されました
40055B89-AE74-4DAE-9017-634EAFE30C84と接続されました

ダウンロードを要請する

特徴


- ダウンロードしたいコンテンツを自動的に小さく分割
- 同じアプリを起動する周囲の端末に自動的にダウンロード
- それらデータを自動的に結合し、一つのファイルへ

データの圧縮ではなく分割に着目


登場人物



クライアント
(iOS)

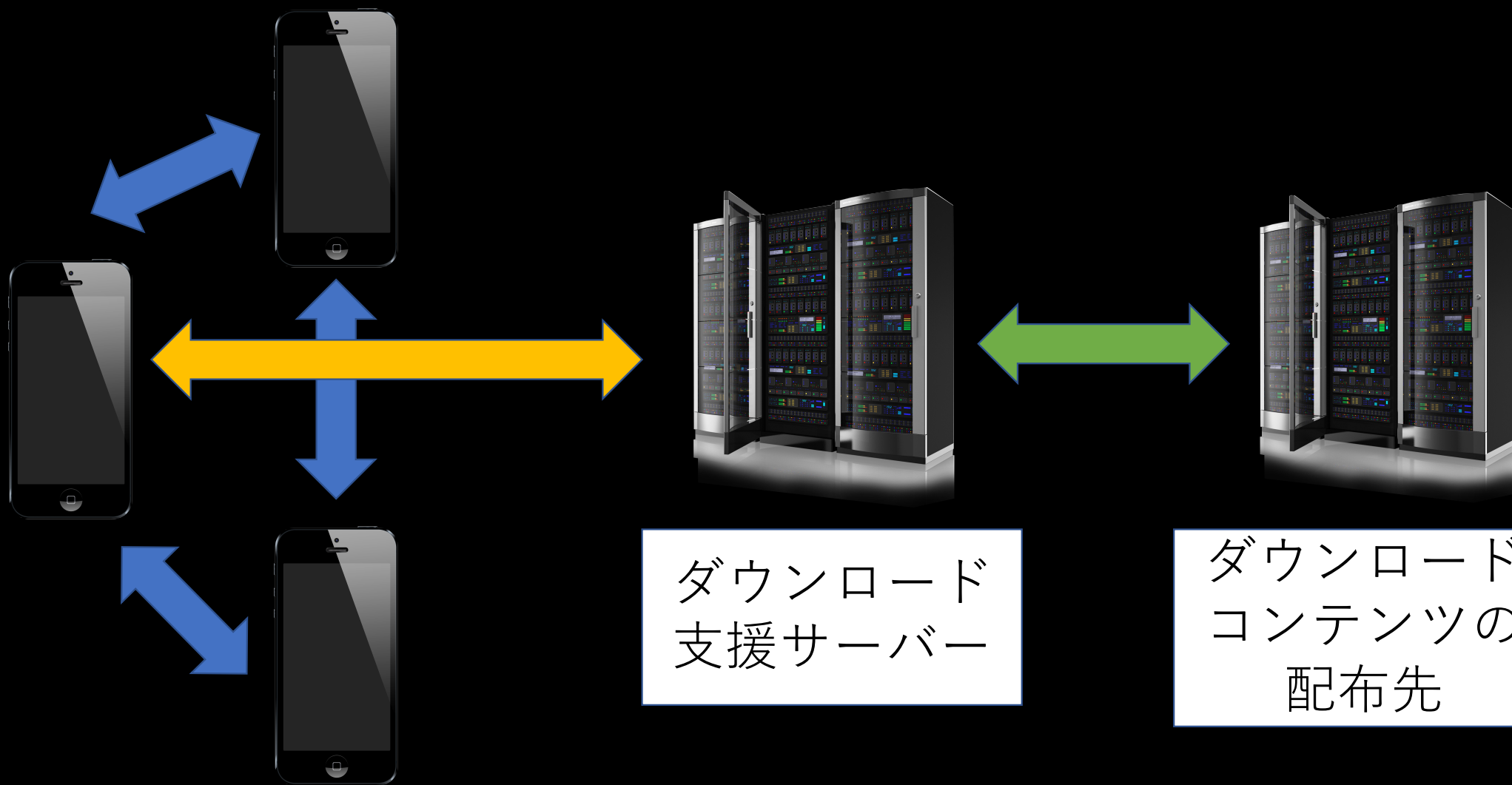


ダウンロード
支援サーバー

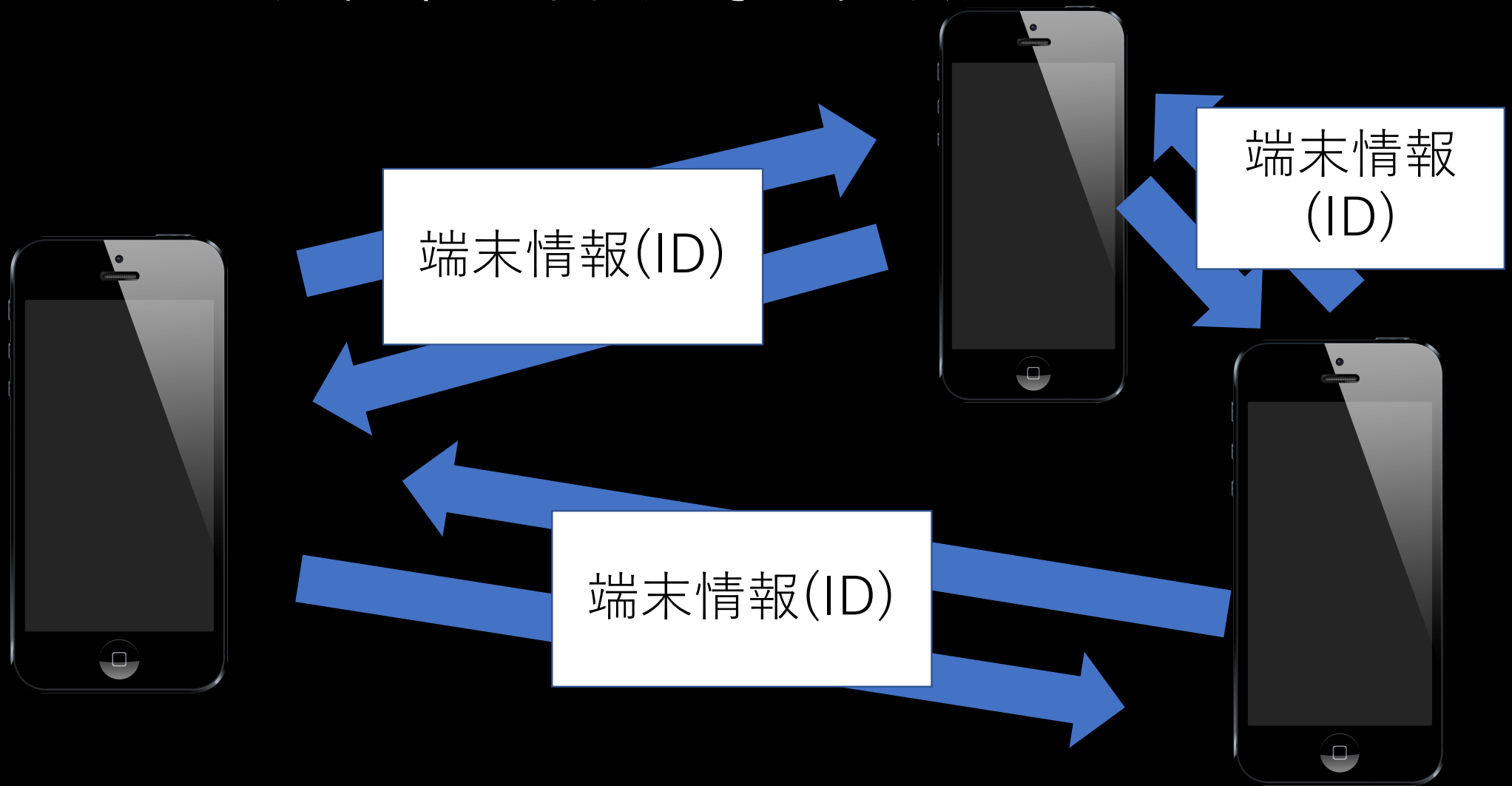


ダウンロード
コンテンツの
配布先

大雑把な構図



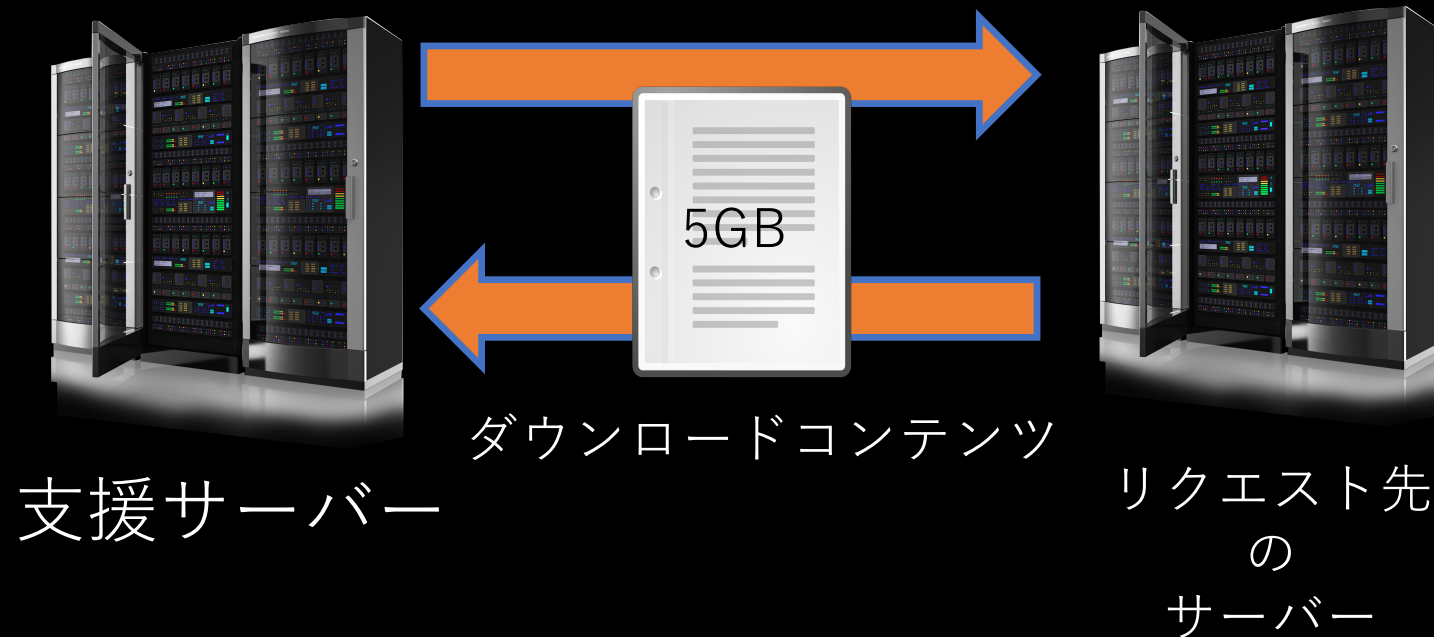
アプリ起動端末を自動的に認識



ダウンロードリクエストを 支援サーバーに送信



支援サーバーからリクエスト先の コンテンツをダウンロード



ダウンロードしたデータを
10MBずつに分割しアップローダーにアップロード

支援サーバー



この作業の後ろでは…

支援サーバー



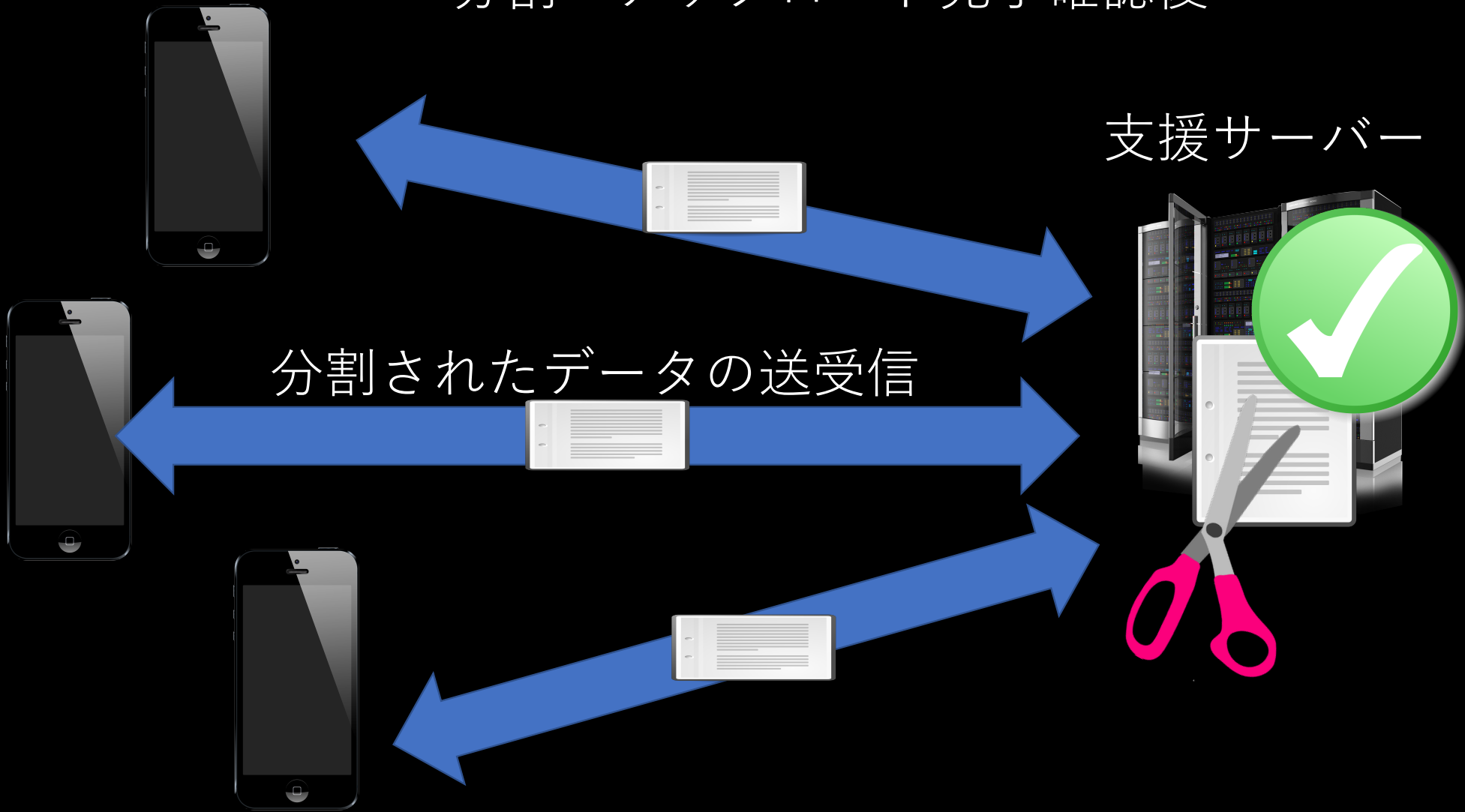
分割作業完了リスト
の常時確認



分割・アップロード完了確認後

支援サーバー

分割されたデータの送受信



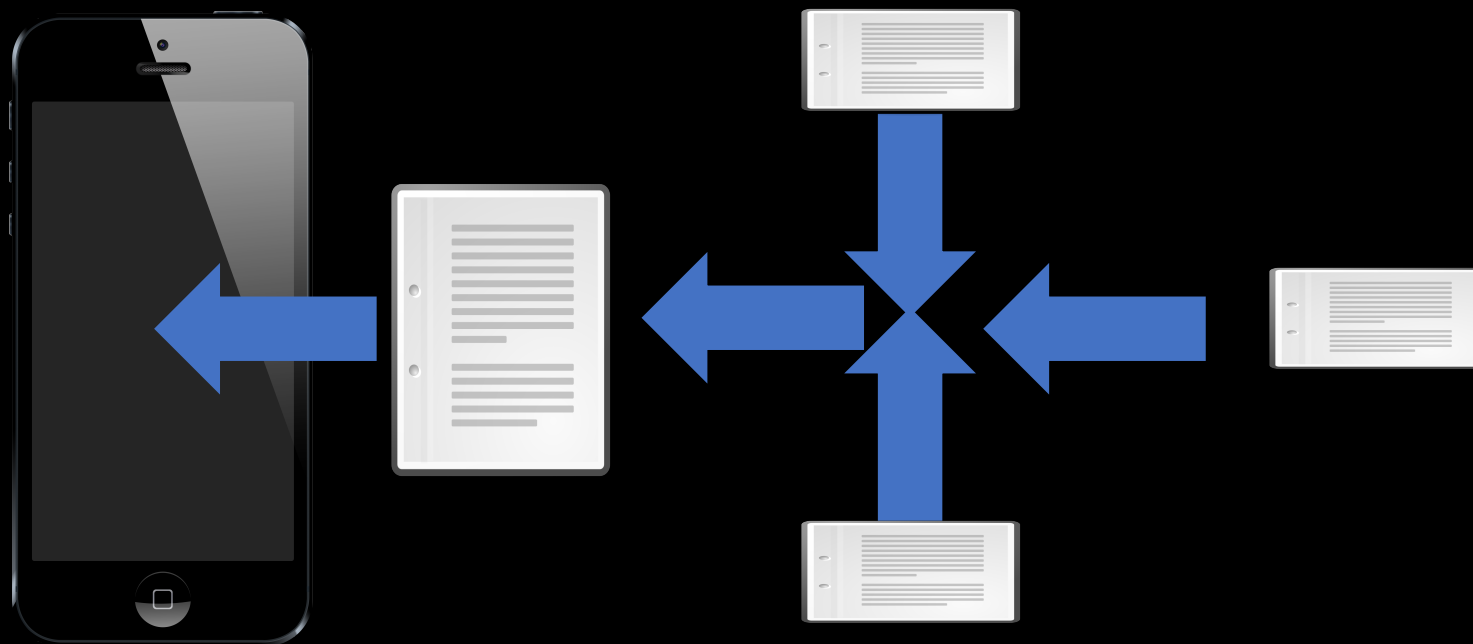


WIFI かBluetoothを介して

The diagram illustrates a data distribution system. A central server, represented by a white rectangular icon with horizontal lines, is positioned at the top center. Two thick blue arrows originate from the server and point downwards and outwards towards two mobile devices, which are depicted as black smartphones. The top arrow points towards a device on the left, and the bottom arrow points towards a device on the right. The background is a solid black color.

別個にダウンロードされたデータを1つの端末に送信

分割されたデータを結合する



端末に保存する

使用言語・フレームワーク

クライアント側

Swift

Alamofire

Multipeer Connectivity Framework

サーバー側

Java

Spark Java (Like Ruby - Sinatra)

問題点

- ・ 通信が不安定

端末同士のやりとりに使うメッセージシステムの不安定さ

現状→複数回送信することによって対応

問題→どのくらい送信すれば完了なのか？

- ・ 分割データの順序が保証できていない

現状→未解決

Javaのサーバーロジックが複雑化を整理

同時に問題の解決へつなげる

将来これをどう活かすか

- ・ 増え続ける端末

増加する端末に対して、単にサーバーを増強していくのでは
いたちごっこになる → 端末の数が増えていくことをデメリット
からメリットに！

- ・ 災害

東日本大震災 TwitterやフリーWIFIを活用 スマホが流行る要因
端末が2011より明らかに多い！ →サーバーにかかる負担を減らし、
高速な情報共有が可能に

将来これをどう活かすか

- ・ 増え続ける端末

増加す

いたな

からメ

Androidにも同種のアプリを
作りたい

ット

- ・ 災害

東日本

端末が2011より明らかに多い！→サーバーにかかる負担を減らし、
高速な情報共有が可能に