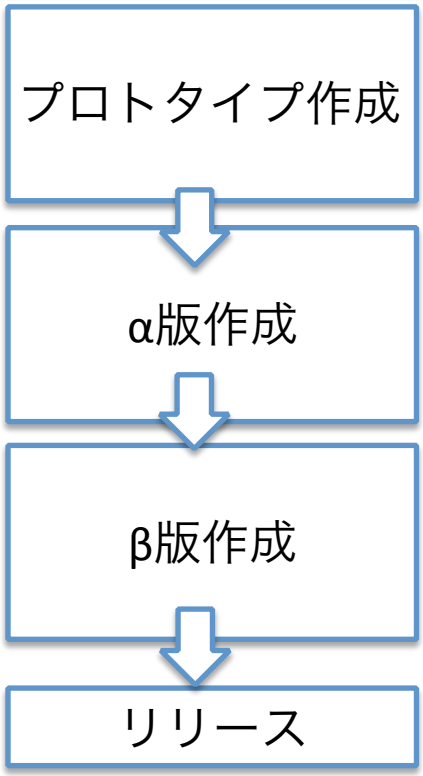
白猫プロジェクトの裏側！

~パフォーマンスチューニングとリアルタイム通信のすべて~

開発フロー

白猫開発フロー



開発メンバー：２～３名
期間：１ヶ月

開発メンバー：１０～１３名
期間：４ヶ月
アクションの基礎が楽しめる完成度

開発メンバー：１５～１８名
期間：４ヶ月
ゲームのサイクルの完成

開発期間：１年弱
開発：最大１８名

COLOPL, Inc. All Rights Reserved

開発の特徴

- 全社配布
  - 全社員にβ版の配布を行います
- 子供レビュー
  - 小学校高学年のお子様にレビューをしてもらいます

COLOPL, Inc. All Rights Reserved

メモリ・パフォーマンス

最適化事例

- 基本はキャッシュ
- フラグメントシェーダーを見直す
- 起動時間の高速化
- ロード時間の短縮化で出会った問題

COLOPL, Inc. All Rights Reserved

オブジェクトのキャッシュ

- ボトルネックはInstantiate
  - ほとんどのオブジェクトがキャッシュされていなかった

- 最も負荷が高かった箇所

```
if( action.hit ){
    // ダメージ表記の生成
    Instantiate(DamageNumber);
}
```


COLOPL, Inc. All Rights Reserved

オブジェクトのキャッシュ

```
void Start()
{
    // コード上で明示的に指定
    CachedObjectManager.Instance.PreCache(obj, num);
}

// 実際に使用するとき
if( action.hit ){
    CachedObjectManager.Instance.Borrow(obj);
}
```


COLOPL, Inc. All Rights Reserved

AudioClipのキャッシュ

- ボトルネックはLoad
  - 毎回ロードしてから再生していた

```
if(useAssetBundle(audio)){
   clip = assetbundle.Load(audio);
}
else{
   clip = Resources.Load(audio);
}
```


COLOPL, Inc. All Rights Reserved

AudioClipのキャッシュ

- 必要になったときにキャッシュするように

```
if(cachedClips.Contains(audio)){
    clip = cachedClips[audio];
}
else{
    ～略～
    cachedClips.Add(audio, clip);
}
```


COLOPL, Inc. All Rights Reserved

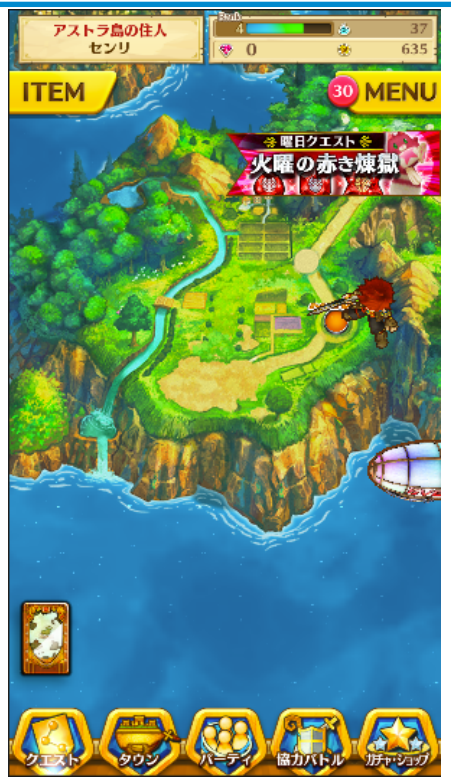
エフェクトのキャッシュ

- 今回は断念
  - Activeにしたときに別の座標に表示されることがある
  - ParticleSystems以外の制御もあった

COLOPL, Inc. All Rights Reserved

GPU側のボトルネック

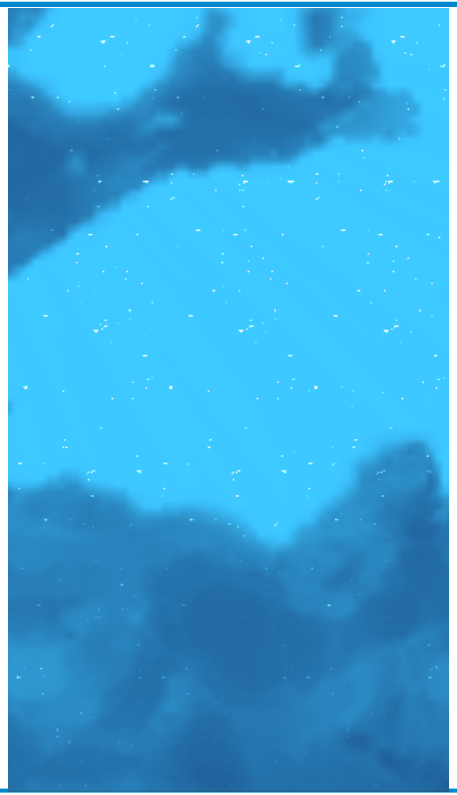
- マップ画面で処理落ち
  - 端末によってはバトルよりも重い



COLOPL, Inc. All Rights Reserved

シェーダーの問題

- 原因は海のシェーダー



COLOPL, Inc. All Rights Reserved

フラグメントをとにかく軽く

```
float4 frag (v2f i) : COLOR    {
    float2 coord1 = i.texcoord.zw*12;
    float2 coord2 = i.texcoord.zw*2*12;
    coord2.y -= _Timer / 16;
    coord1.x += sin( i.texcoord.w * 2 + radians( _Timer * 45 ) )/6;
    coord2.x -= sin( i.texcoord.w * 8 + i.ref.w )/3;
```

フラグメントシェーダーで波のアニメーション計算をしていた

COLOPL, Inc. All Rights Reserved

フラグメントをとにかく軽く

```
float4 frag (v2f i) : COLOR    {
    float2 coord1 = i.texcoord.zw*12;
    float2 coord2 = i.texcoord.zw*2*12;
    coord2.y -= _Timer / 16;
    coord1.x += sin( i.texcoord.w * 2 + radians( _Timer * 45 ) )/6;
    coord2.x -= sin( i.texcoord.w * 8 + i.ref.w )/3;
```

フラグメントで計算する必要がない

COLOPL, Inc. All Rights Reserved

フラグメントをとにかく軽く

```
v2f vert (appdata_img v) {
    ～略～
    o.coord.xy = o.texcoord.zw*12;
    o.coord.zw = o.coord.xy*2;
    o.coord.w -= _Timer * 0.0625;
    o.coord.x += sin( o.texcoord.w * 0.333 + radians( _Timer * 7.5 ) );
    o.coord.z -= sin( o.texcoord.w * 2.666 + o.ref.w * 0.3333);
```

バーテックスシェーダーへ波の計算を移動
使っていない変数に計算後の値をセット

COLOPL, Inc. All Rights Reserved

シェーダーの一括変換

- 必要のないカラー計算をしているシェーダーを一括変換
- PC用のシェーダーを使ってしまっているものも変換

- 白猫で実際に変換した際の条件

①Colorプロパティを持つ
②Colorをフラグメントシェーダーで乗算
③Colorの値が初期値のままである
④アニメーションをしない

COLOPL, Inc. All Rights Reserved

起動時間の高速化

- ある程度プレイすると起動時間が遅くなることが判明
  - Androidで25秒
  - iOSで１分

COLOPL, Inc. All Rights Reserved

起動時間の高速化

- ある程度プレイすると起動時間が遅くなることが判明
  - Androidで25秒
  - iOSで１分

白猫プロジェクトのAssetBundleは現在**約３万種類**

Caching.readyが遅かった。

COLOPL, Inc. All Rights Reserved

独自のキャッシュシステムの作成

colopl

目標

- Caching.readyを待たずに起動速度を早く
- www.LoadFromCacheOrDownloadを使わない
- AssetBundleのバージョン管理が出来る必要がある

COLOPL, Inc. All Rights Reserved

独自のキャッシュシステムの作成

①起動時にAssetVersionの入ったリストをダウンロード
②PlayerPrefsに保存されているVersionと比較
③更新されていたらダウンロードする
④ストレージへの書き込み
⑤PlayerPrefsに新しいバージョンを入れる

COLOPL, Inc. All Rights Reserved

独自のキャッシュシステムの作成

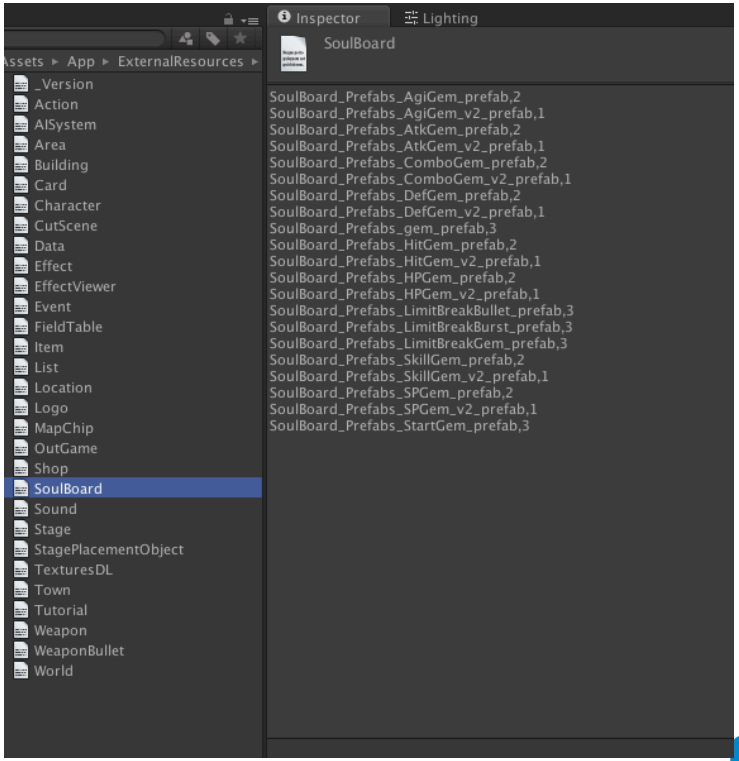
①起動時にAssetVersionの入ったリストをダウンロード
②PlayerPrefsに保存されているVersionと比較
③更新されていたらダウンロードする
④ストレージへの書き込み
⑤PlayerPrefsに新しいバージョンを入れる

起動時間が１分から８秒に！

COLOPL, Inc. All Rights Reserved

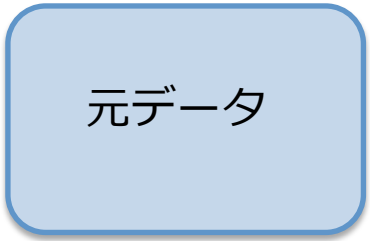
AssetVersionListの作成

Path,VersionNumberの形式で各AssetBundleのバージョンを管理



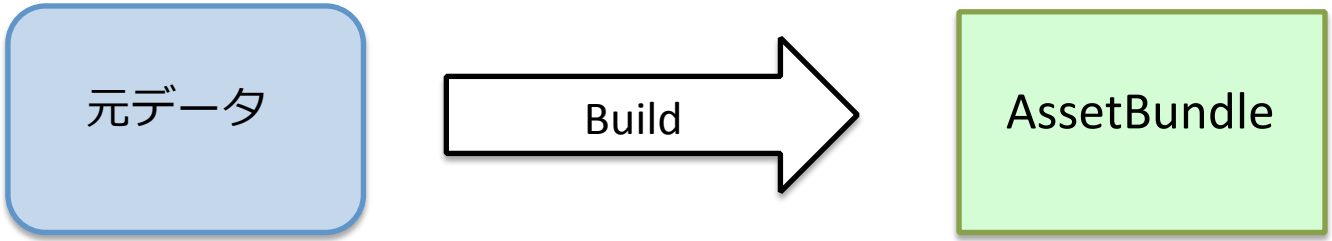
COLOPL, Inc. All Rights Reserved

AssetVersionListの作成



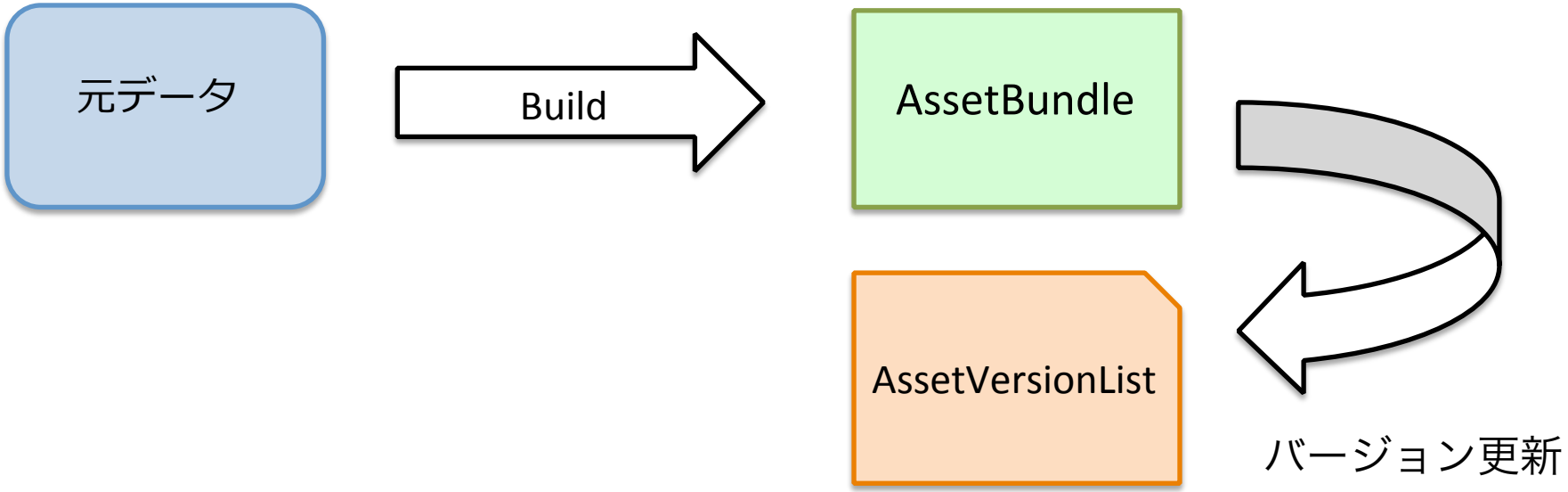
COLOPL, Inc. All Rights Reserved

AssetVersionListの作成



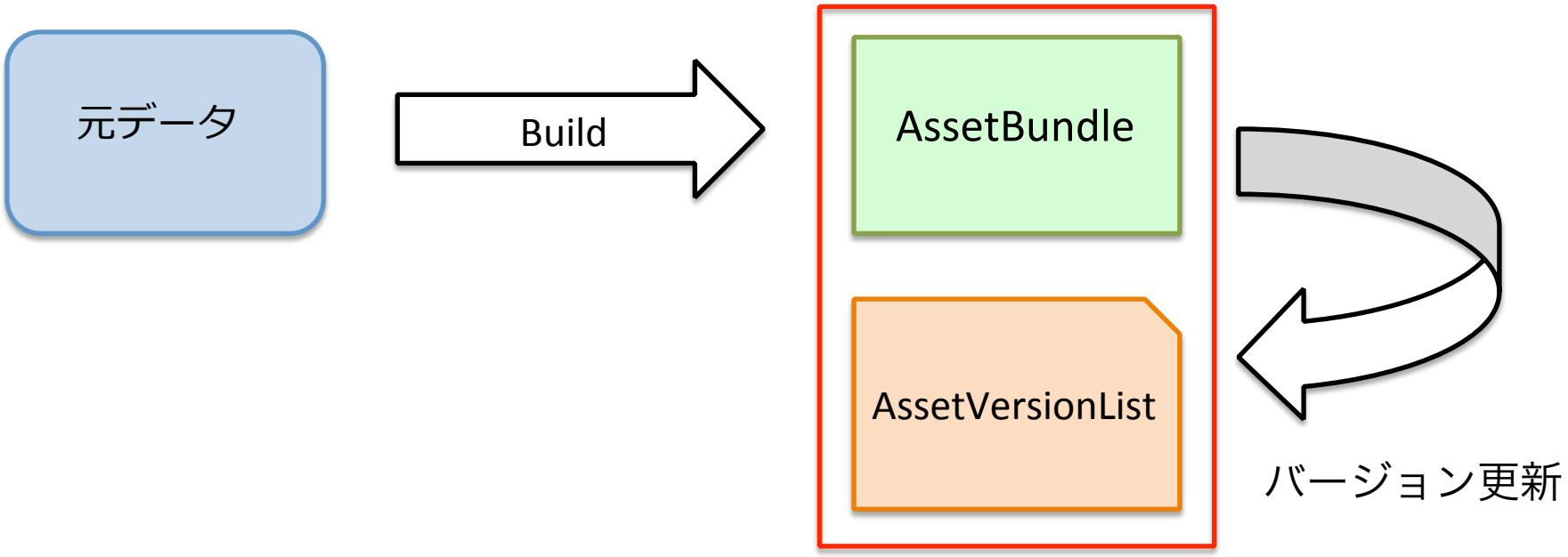
COLOPL, Inc. All Rights Reserved

AssetVersionListの作成



COLOPL, Inc. All Rights Reserved

AssetVersionListの作成



この２つをサーバーにアップロードする

COLOPL, Inc. All Rights Reserved

ファイル読み込み数の問題

- iOSでは同時に開けるファイルの制限が256個
- カットシーンで大量のAssetBundleをダウンロードしていた

COLOPL, Inc. All Rights Reserved

ファイル読み込み数の問題

- 複数ファイルのダウンロード処理の呼び出し

```
RequestAssetsCoroutine(
    List<string> paths,
    Action<Dictionary<string, AssetBundle>> onComplete
);
```

paths分ダウンロードをしてDictionaryにつめて返す

COLOPL, Inc. All Rights Reserved

ファイル読み込み数の問題

- 複数ファイルのダウンロード処理の呼び出し

```
RequestAssetsCoroutine(
    List<string> paths,
    Action<Dictionary<string, AssetBundle>> onComplete
);
```

← 256個以上指定されると進行不能に

paths分ダウンロードをしてDictionaryにつめて返す

COLOPL, Inc. All Rights Reserved

ファイル読み込み数の問題

- 発生した原因
  - 体感ダウンロード時間を少しでも減らすためにバックグラウンドでのダウンロードを増やしていった
- 問題点
  - 上限数を超えたときに発生するため原因が特定しにくい
- 解決方法
  - カットシーンでのロードは１ファイルずつCloseしOnCompleteで何も行わないようにした

COLOPL, Inc. All Rights Reserved

メモリ、プログラムサイズ最適化

- 不必要なDLL、コードの削除
- StrippingLevelの変更
- AssetBundleのメモリリークの修正

COLOPL, Inc. All Rights Reserved

不必要なdll, コードの削除

- AssetStoreで購入した物のサンプルなどは完全に削除するかEditorフォルダの中に入れる
- 同様の機能を行うものがすでに入っているのにdllをリンクしてしまっているケースがあった
- 1つでも.jsがあるとBoo.Lang.dllがリンクされるので必要ないのであれば削除しておく

COLOPL, Inc. All Rights Reserved

StrippingLevelの変更

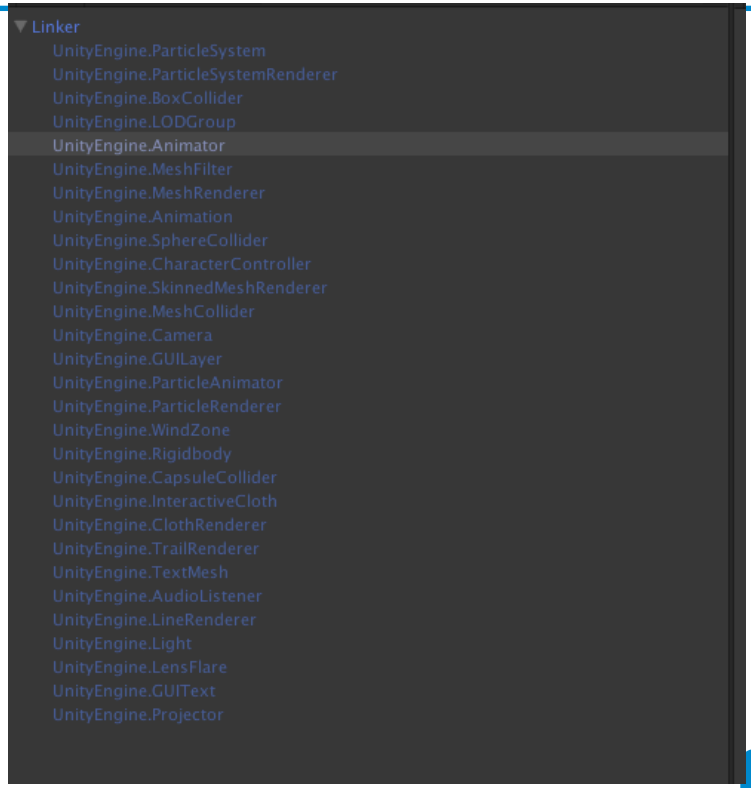
- UnityEngine以下の必要なコンポーネントまでStripされてしまうことがあった

- AssetBundleに該当するコンポーネントを含んでしまっていた場合に正常に動作しなかった

COLOPL, Inc. All Rights Reserved

Linkerシーンの作成

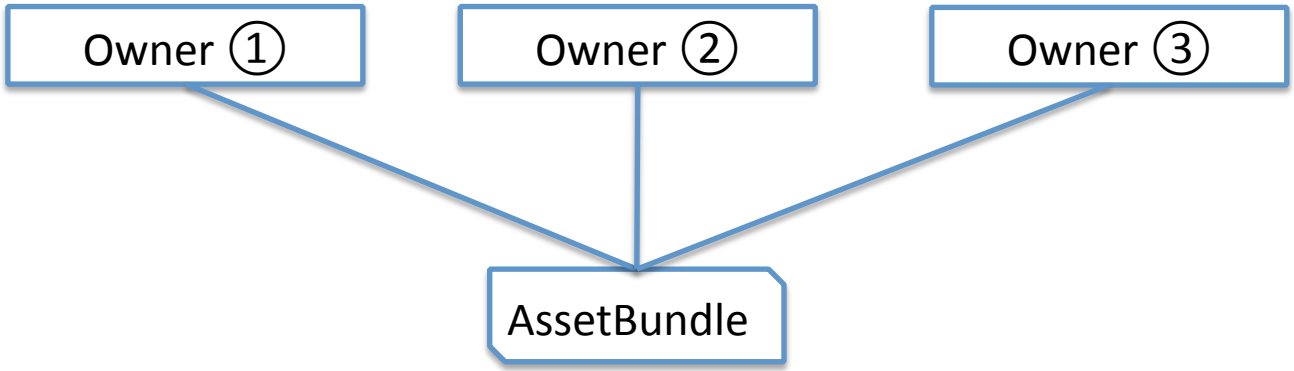
- 必要なコンポーネントをもつオブジェクトをシーンに配置
- シーンの作成には自動でコンポーネントを抽出してGameObject生成
- AwakeでルートをDestroy



COLOPL, Inc. All Rights Reserved

AssetBundleのメモリリークの修正

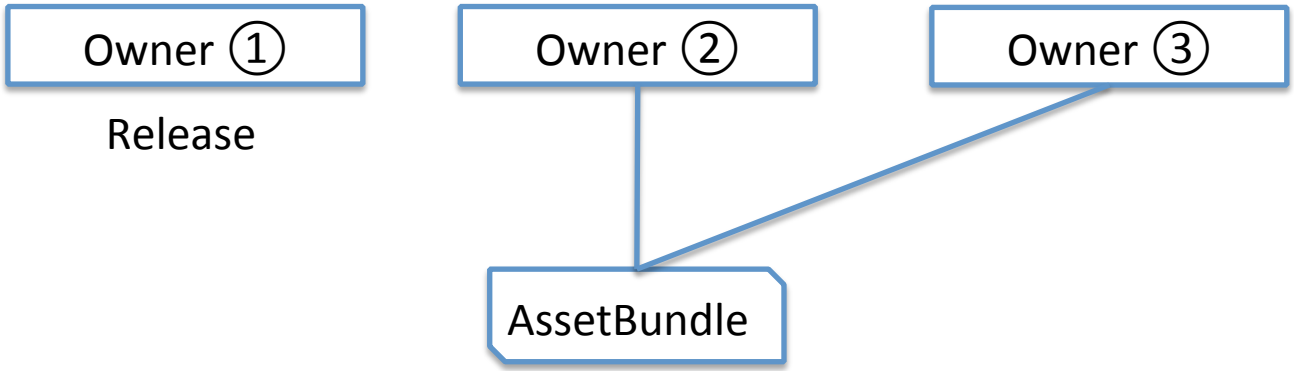
- 白猫プロジェクトでは明示的にReleseする必要がある管理形式



COLOPL, Inc. All Rights Reserved

AssetBundleのメモリリークの修正

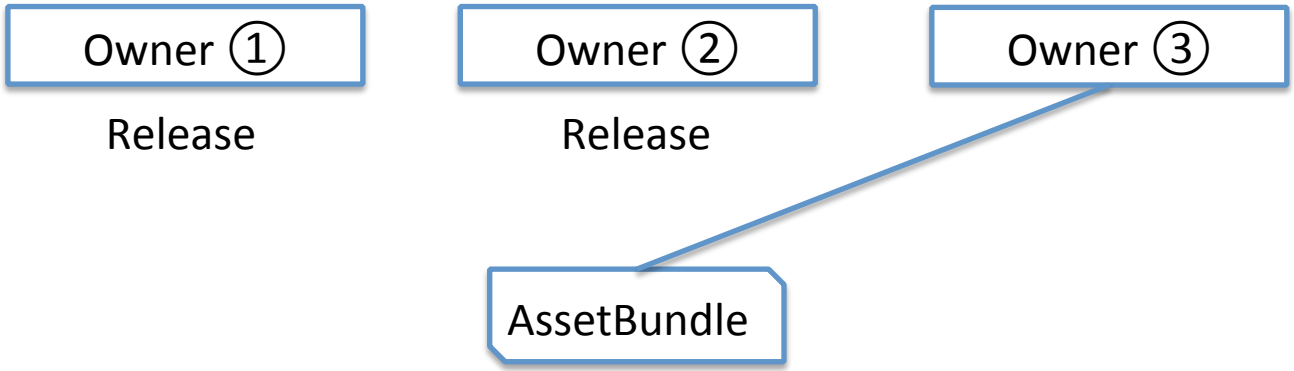
- 白猫プロジェクトでは明示的にReleseする必要がある管理形式



COLOPL, Inc. All Rights Reserved

AssetBundleのメモリリークの修正

- 白猫プロジェクトでは明示的にReleseする必要がある管理形式



COLOPL, Inc. All Rights Reserved

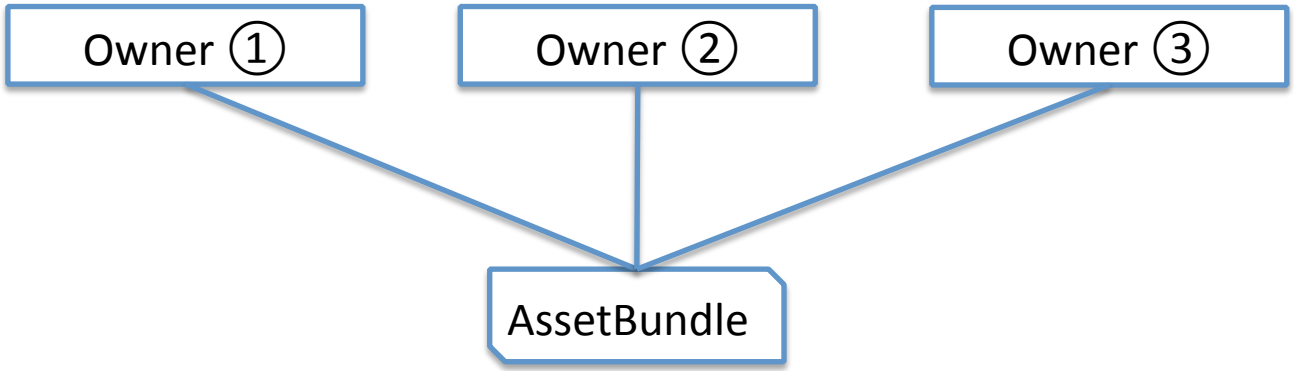
AssetBundleのメモリリークの修正

- 白猫プロジェクトでは明示的にReleseする必要がある管理形式

COLOPL, Inc. All Rights Reserved

AssetBundleのメモリリークの修正

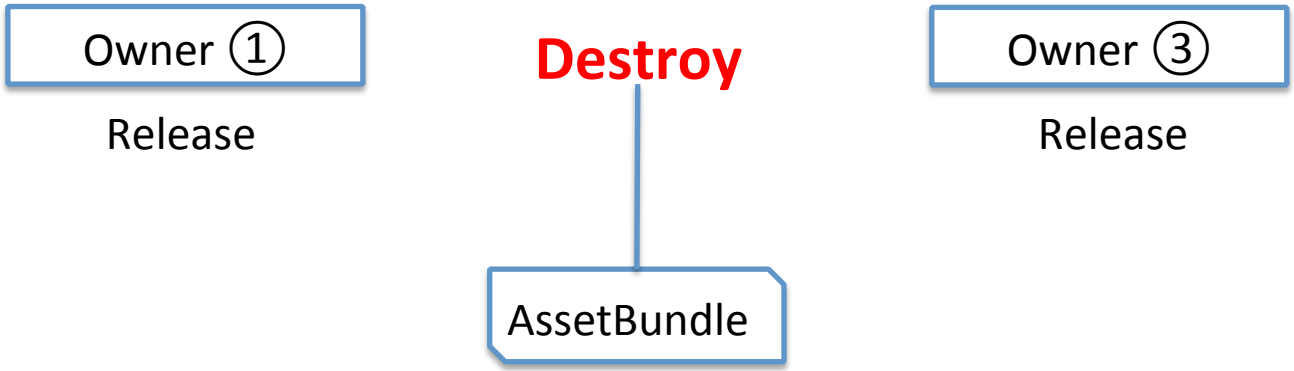
- 実際にあった事例



COLOPL, Inc. All Rights Reserved

AssetBundleのメモリリークの修正

- 実際にあった事例

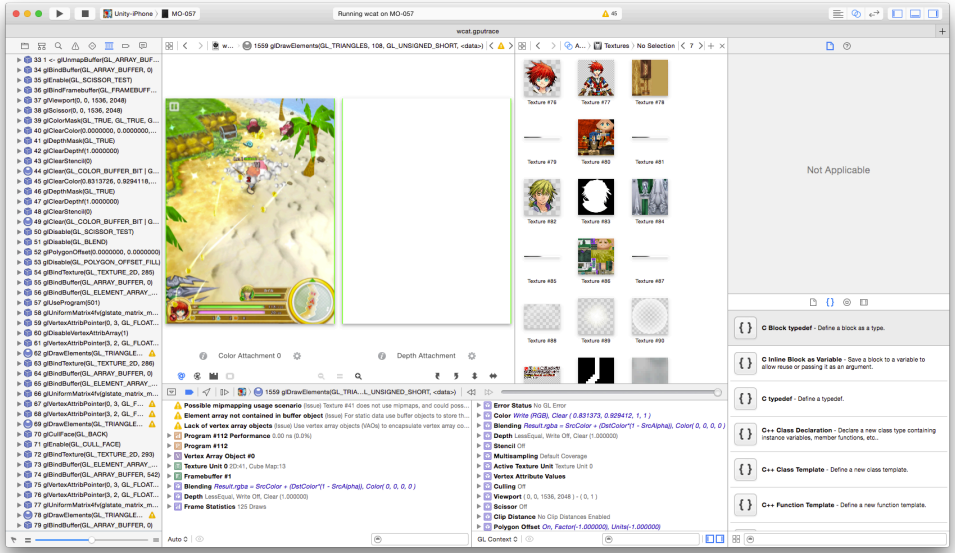


Releaseの前にDestroyされてしまい
AssetBundleが解放されなかった

COLOPL, Inc. All Rights Reserved

Xcodeを使って調査

- FrameDebuggerでVRAMの中身をチェックして破棄されてなければならないテクスチャを調査



COLOPL, Inc. All Rights Reserved

今後の課題

- CharacterControllerをやめる
- UIの構築用ではない軽量な2Dアセットを探す（もしくは作る）
- キャッシュできて大量に表示できるエフェクトシステム
- ダメージパケットなど大量に送信する際のバッチング

COLOPL, Inc. All Rights Reserved

リアルタイム・サーバー

リアルタイム・サーバー

サーバーシステム構成

リアルタイム通信

クライアント・サーバー注意点

COLOPL, Inc. All Rights Reserved

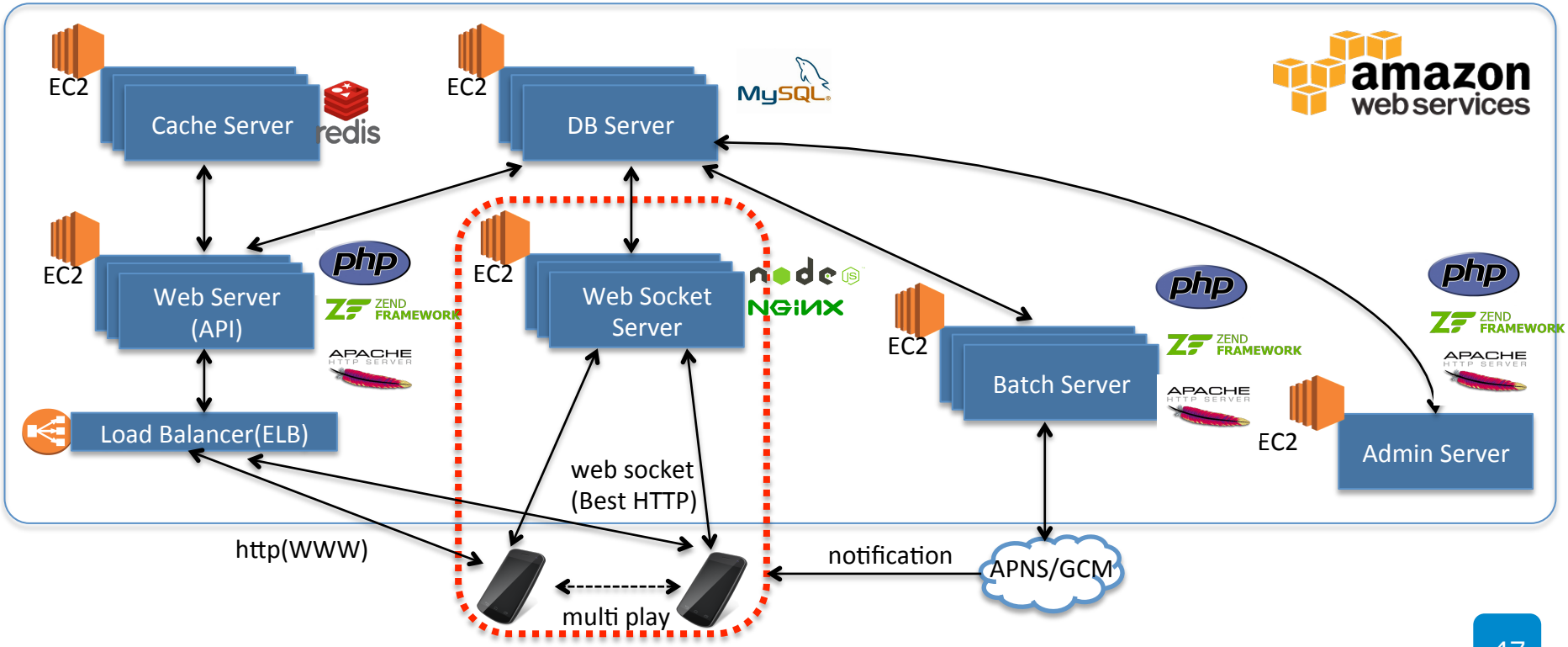
サーバーシステム構成

リアルタイム通信

クライアント・サーバー注意点

COLOPL, Inc. All Rights Reserved

白猫プロジェクト システム構成



COLOPL, Inc. All Rights Reserved

リアルタイム・サーバー

サーバーシステム構成

リアルタイム通信

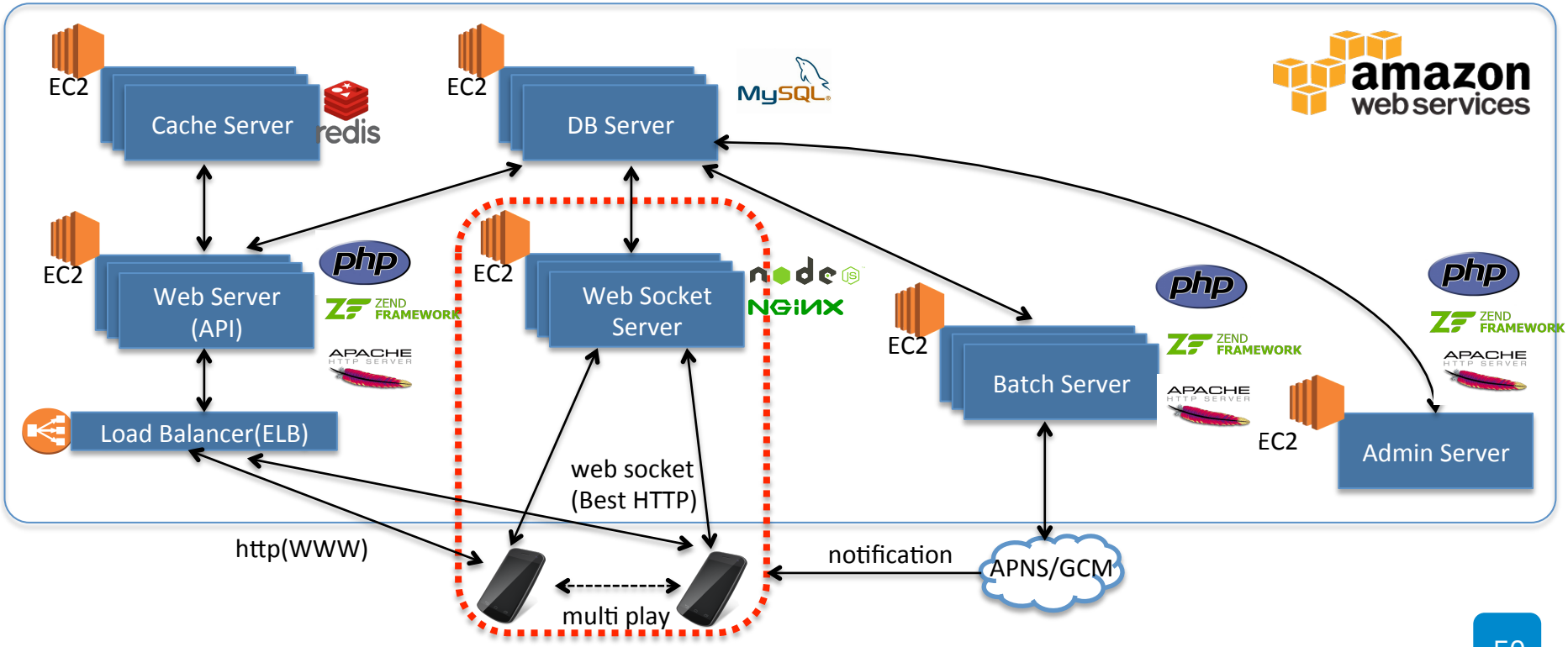
クライアント・サーバー注意点

COLOPL, Inc. All Rights Reserved

白猫マルチプレイ動画

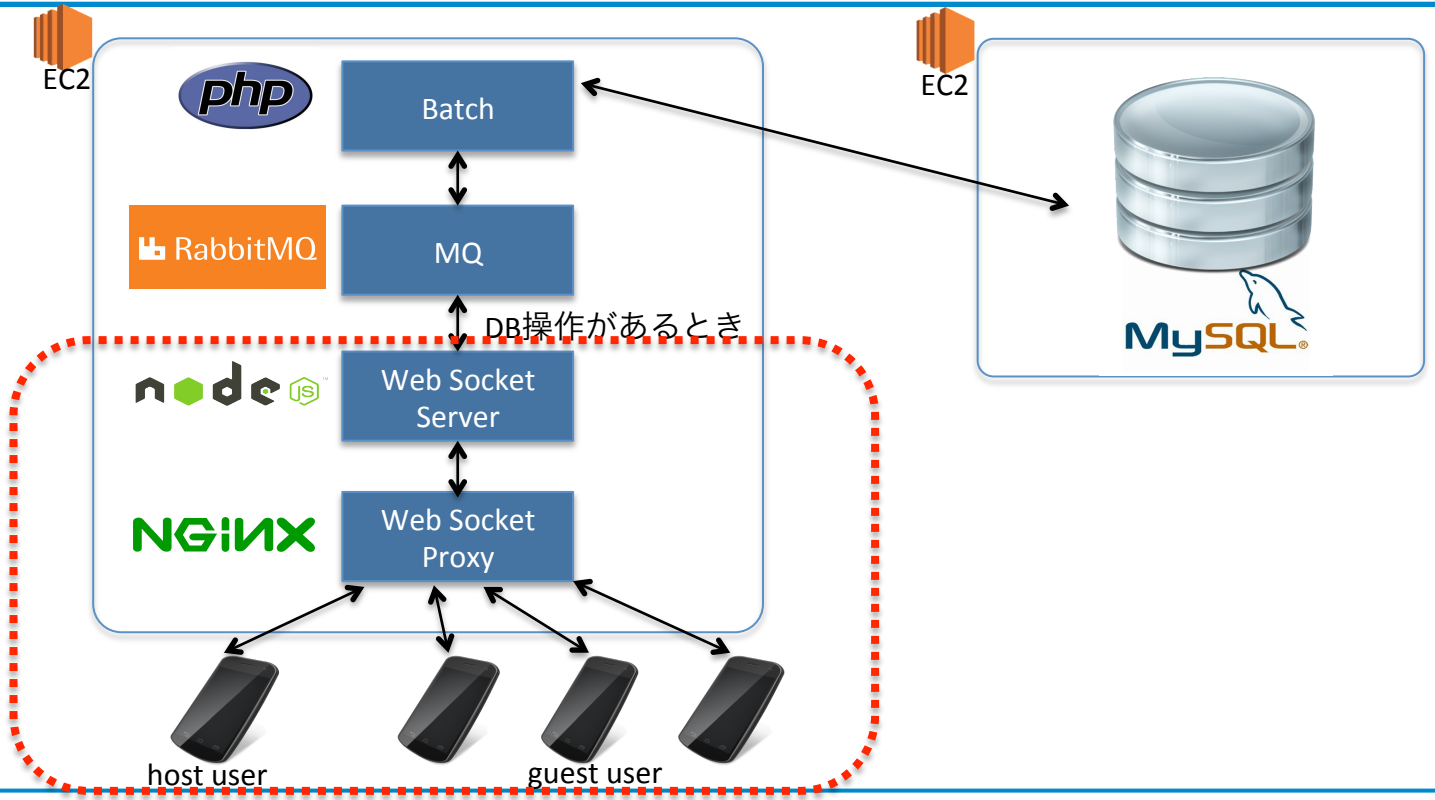
COLOPL, Inc. All Rights Reserved

白猫プロジェクト システム構成



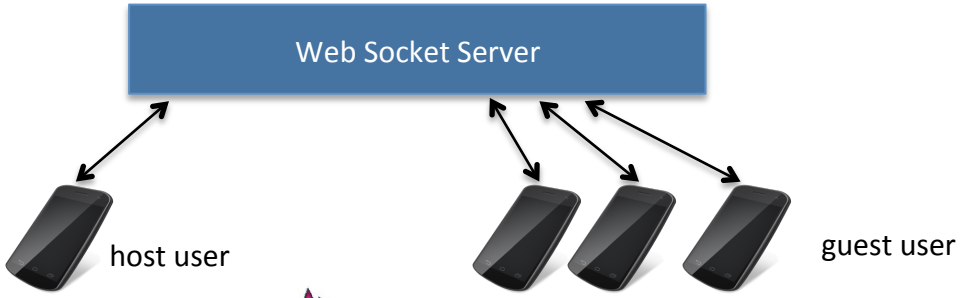
COLOPL, Inc. All Rights Reserved

Web Socket Server



COLOPL, Inc. All Rights Reserved

マルチプレイ同期データ



host(ルーム作成ユーザ)
・Bossデータ(AIステート遷移 etc)
・自キャラ
- 位置
- アクション(攻撃,スキル)
- アクション結果(ダメージ,ギミック)
・ステージギミック

共闘で必要になる同期データの
マスターデータを持つ

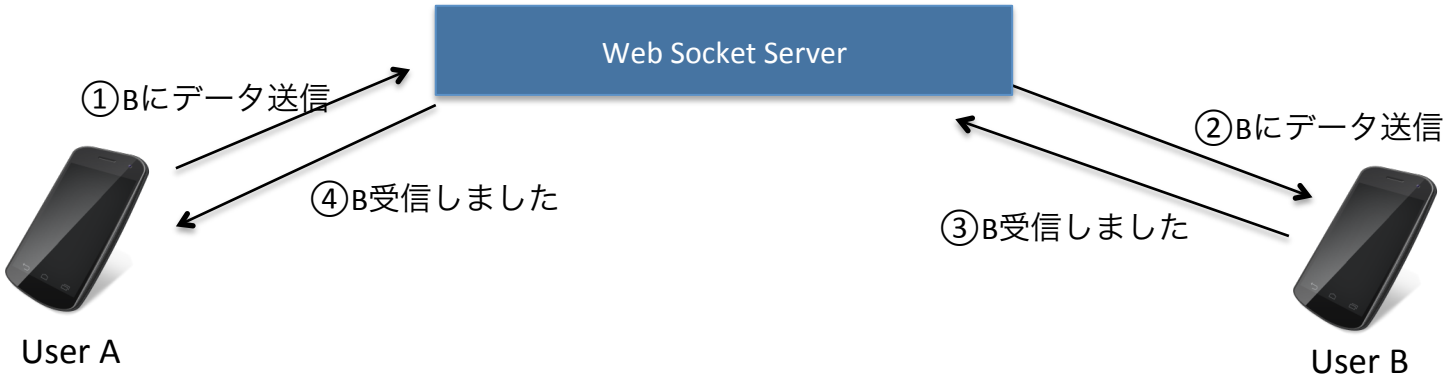
guest(ルーム参加ユーザ)
・自キャラ
- 位置
- アクション(攻撃,スキル)
- アクション結果(ダメージ,ギミック)

hostユーザからデータをもらう

※雑魚敵の動きは同期しておらず
各ユーザで自由に動き、ダメージなどは同期

COLOPL, Inc. All Rights Reserved

通信保証データ



データが受け取れないと進行不能になるような重要度の高いデータは
通信保証データとして受け取り済みであることを送信者に送信する

④が返ってこなかったら再度①から実施
※接続されているかは定期的に各クライアントがpingを送信し
　サーバ側でpingがないユーザは切断したと判断する

COLOPL, Inc. All Rights Reserved

リアルタイム・サーバー

サーバーシステム構成

リアルタイム通信

クライアント・サーバー注意点

COLOPL, Inc. All Rights Reserved

クライアントとサーバーのデータ不一致

クライアントがデータを保持するようなゲームアプリの場合データの不一致が発生する

- リクエスト成功、レスポンス受け取れないケース
- モバイルでは移動などにより接続が切断されやすい

データが不一致だとサーバー側でエラーが発生

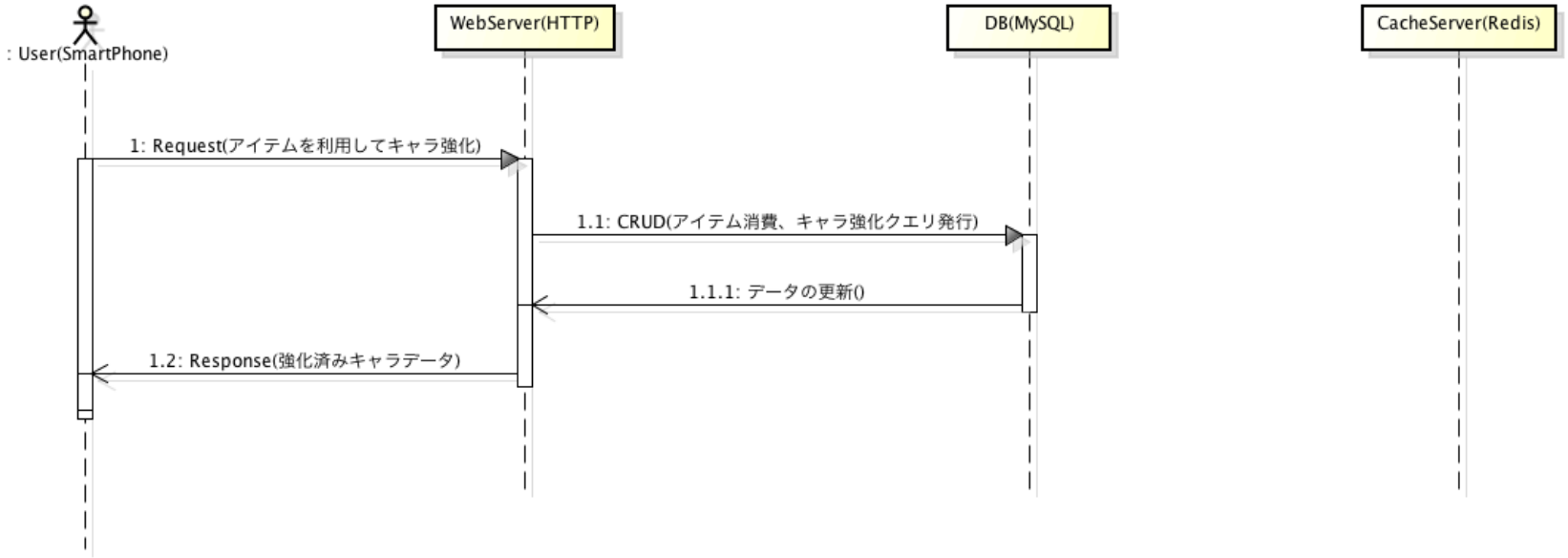
User A
Item X 5個

消費したリクエストが成功したかわからないため
クライアントのアイテム個数は元のまま

user_id	item_id	Num
A	X	5->3

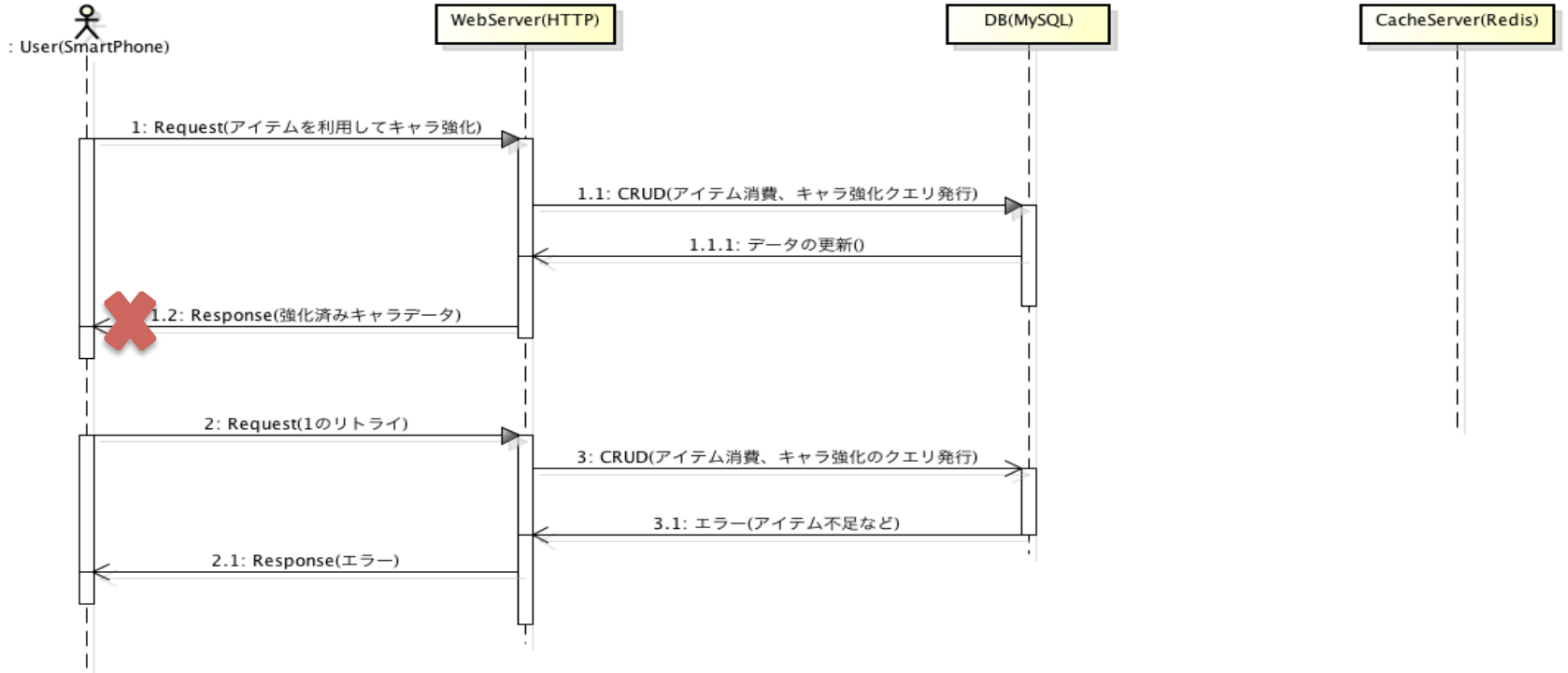
COLOPL, Inc. All Rights Reserved

クライアントとサーバーのデータ不一致



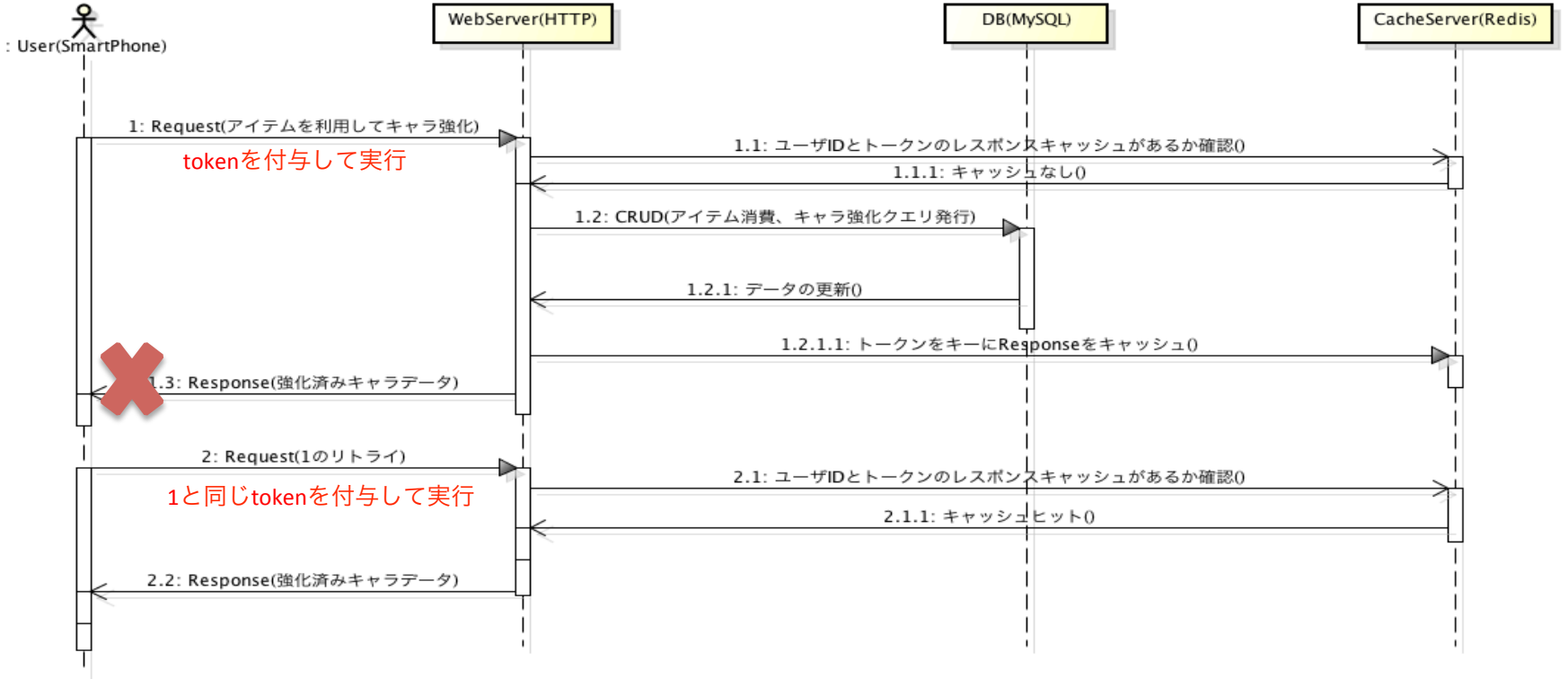
COLOPL, Inc. All Rights Reserved

クライアントとサーバーのデータ不一致



COLOPL, Inc. All Rights Reserved

クライアントとサーバーのデータ不一致



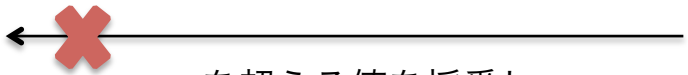
COLOPL, Inc. All Rights Reserved

クライアントの数値データの上限値

DBの最大値を入れてテストする

-クライアントの修正を伴う不具合になるかも

int(System.Int32)　21億



Intを超える値を採番し
クライアントへ送りエラー

Bigint(最大値 922京)

id(bigint)	Foo	bar
2,147,483,647	…	…

COLOPL, Inc. All Rights Reserved

白猫プロジェクトの裏側

ご清聴ありがとうございました。

COLOPL, Inc. All Rights Reserved