

セキュリティのお話

～安全なゲームを作るために～

【 サーバ編 】

発表者
石森礼二

発表者の来歴

- 石森礼二

- ガラケーアプリの黎明期に就職
- アプリ大量に作っていたら、そのうちサーバー側もやるようになりました
- おかげで、連携部分で起きがちなミスを色々知ってます
- **SQUARE ENIX**でテクニカルディレクターをしています
- スクストサーバーフレームワークの基礎構築やりました

セキュリティの話

- 面白いゲームをちゃんとユーザーに届ける
 - コンシューマ畑の開発会社は、非常に面白いゲームを作ります
 - でも、サーバーセキュリティーに弱く、チーターに**ゲームを破壊**されます
 - 結果、面白いゲームがサービス停止に追い込まれます
- 防ぎたいですね

セキュリティって難しい？

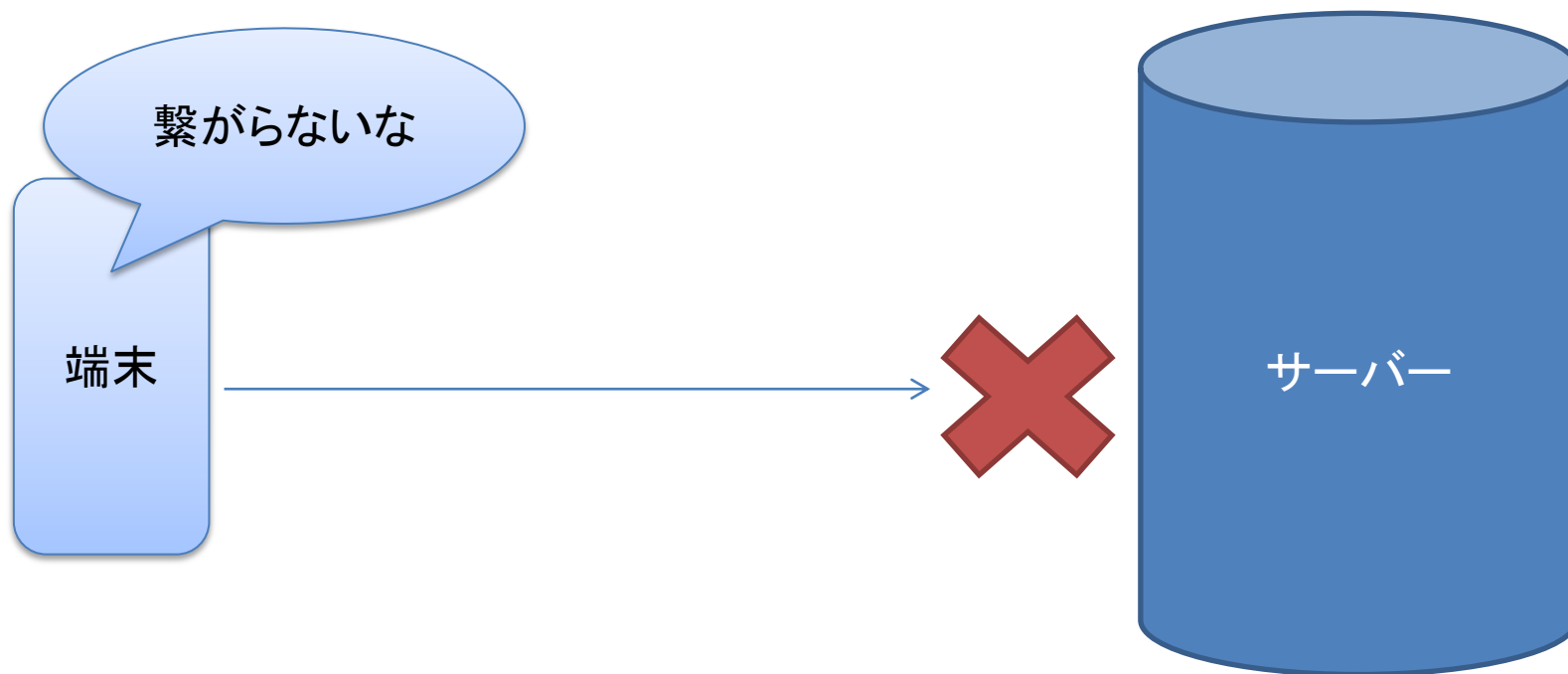
- セキュリティーは**基本の積み重ね**
だからこの発表でも、**超テクニク**は**出てこない！**
- セキュリティーって基本に忠実に作って
「**負けなければ勝ち**」なんです。
簡単（？）ですね

負け＝プログラマのミス

- 本来「全通信を受け付けないサーバー」なら**無敵**なはず。脆弱性があるのは機能を追加したプログラマが**ミスったから**
- スクストフレームワークの**根底設計**は「プログラマのミスを防ぐ」に特化し、**単純化**されています

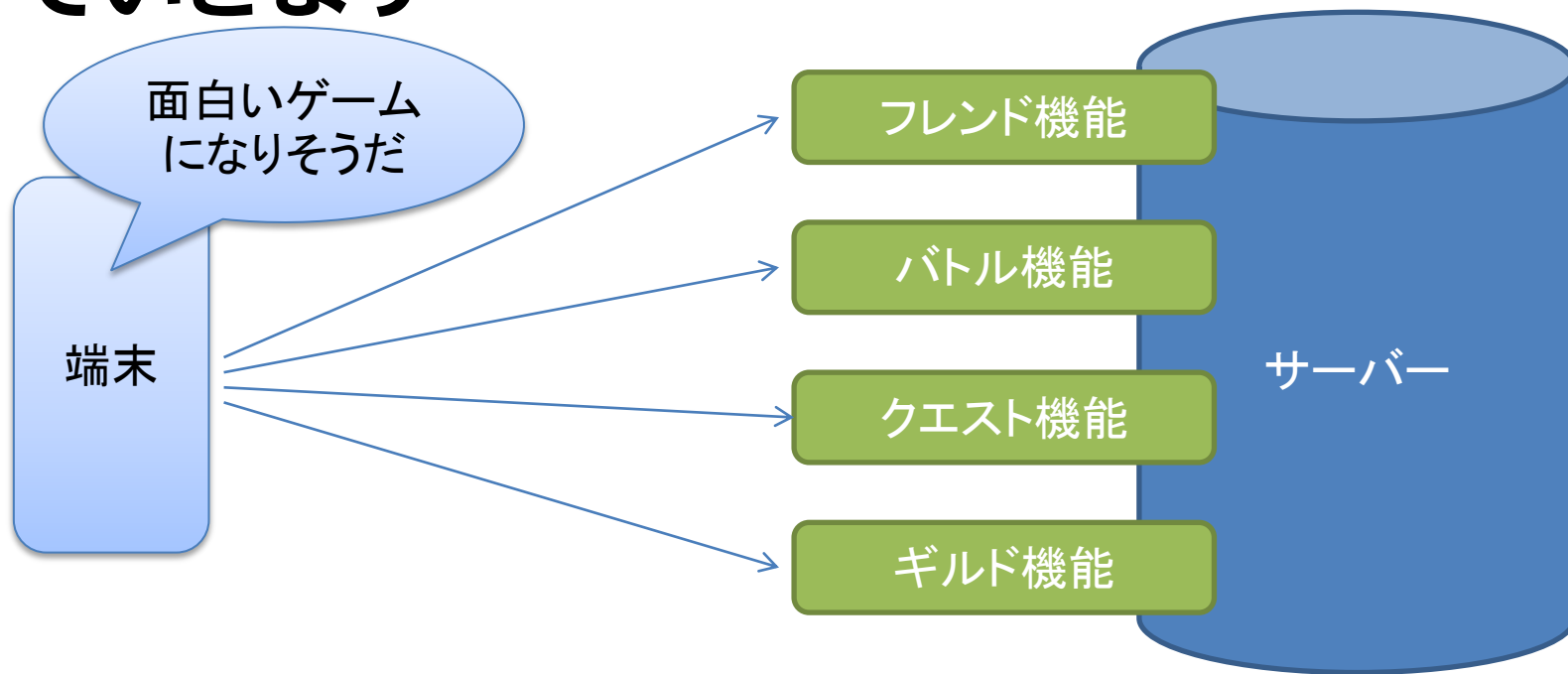
よくある負けパターン

- ・ サーバーは最初、全ての通信を受け付けません



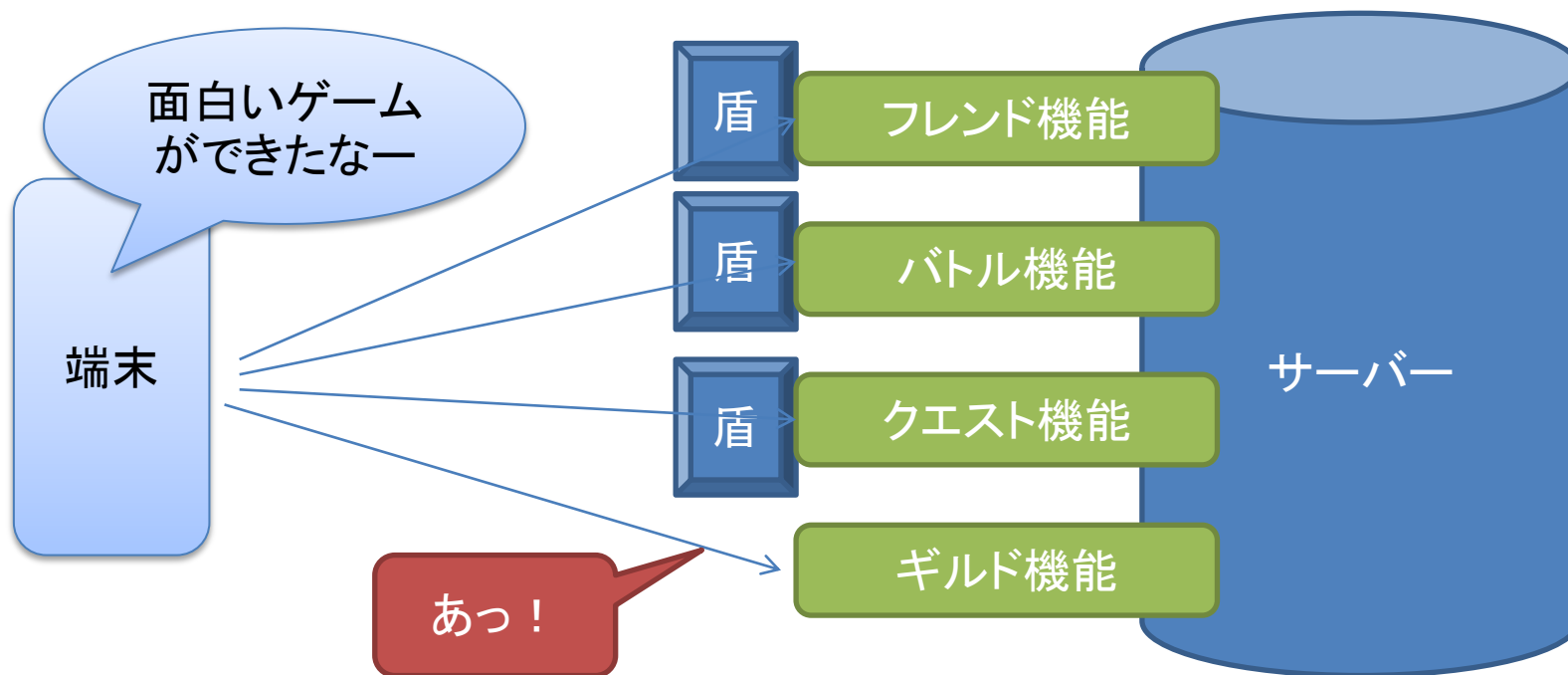
よくある負けパターン

- とりあえずゲームを動かしたいので、サーバーに通信を受け付ける口を増やしていきます



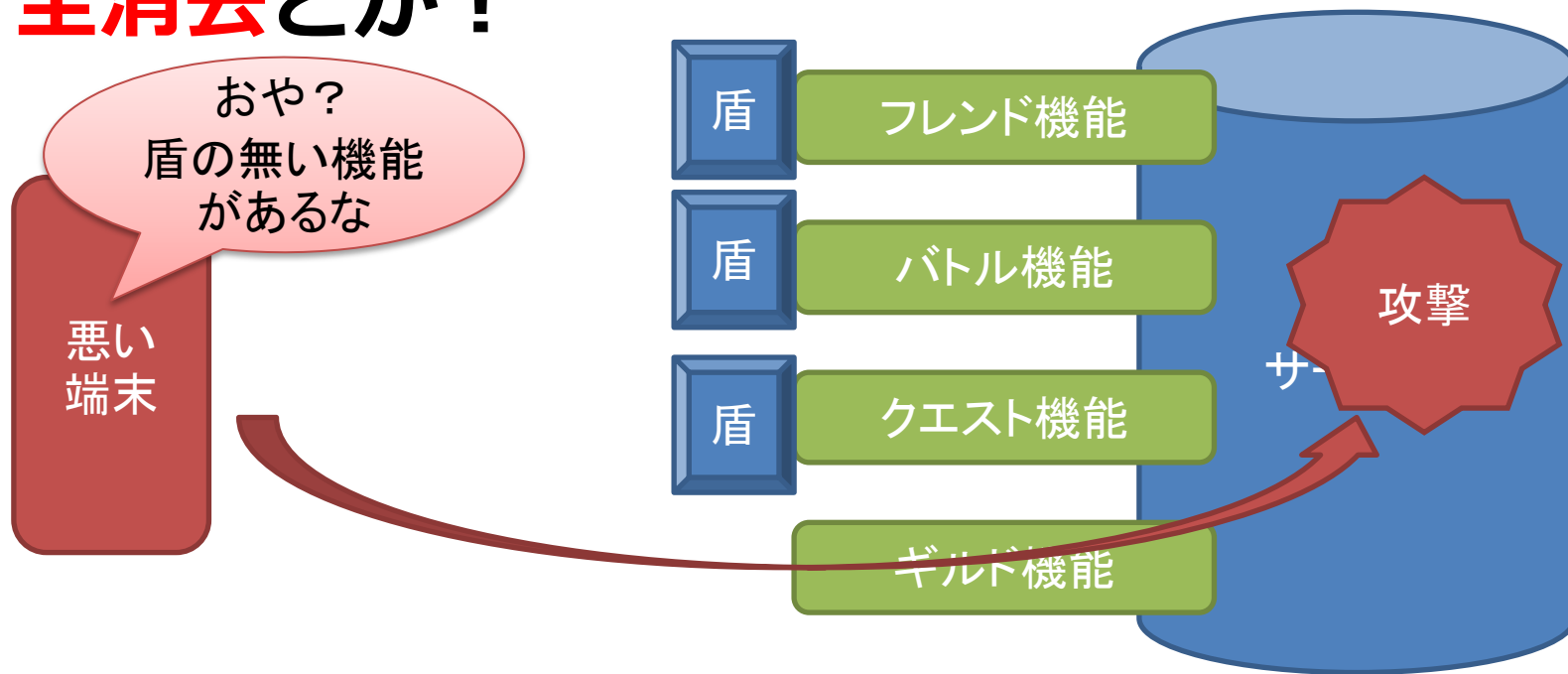
よくある負けパターン

- 完成してからセキュリティを足す



よくある負けパターン

- 足し忘れてゲームを破壊される
- **SQLインジェクションでユーザーデータ全消去とか！**

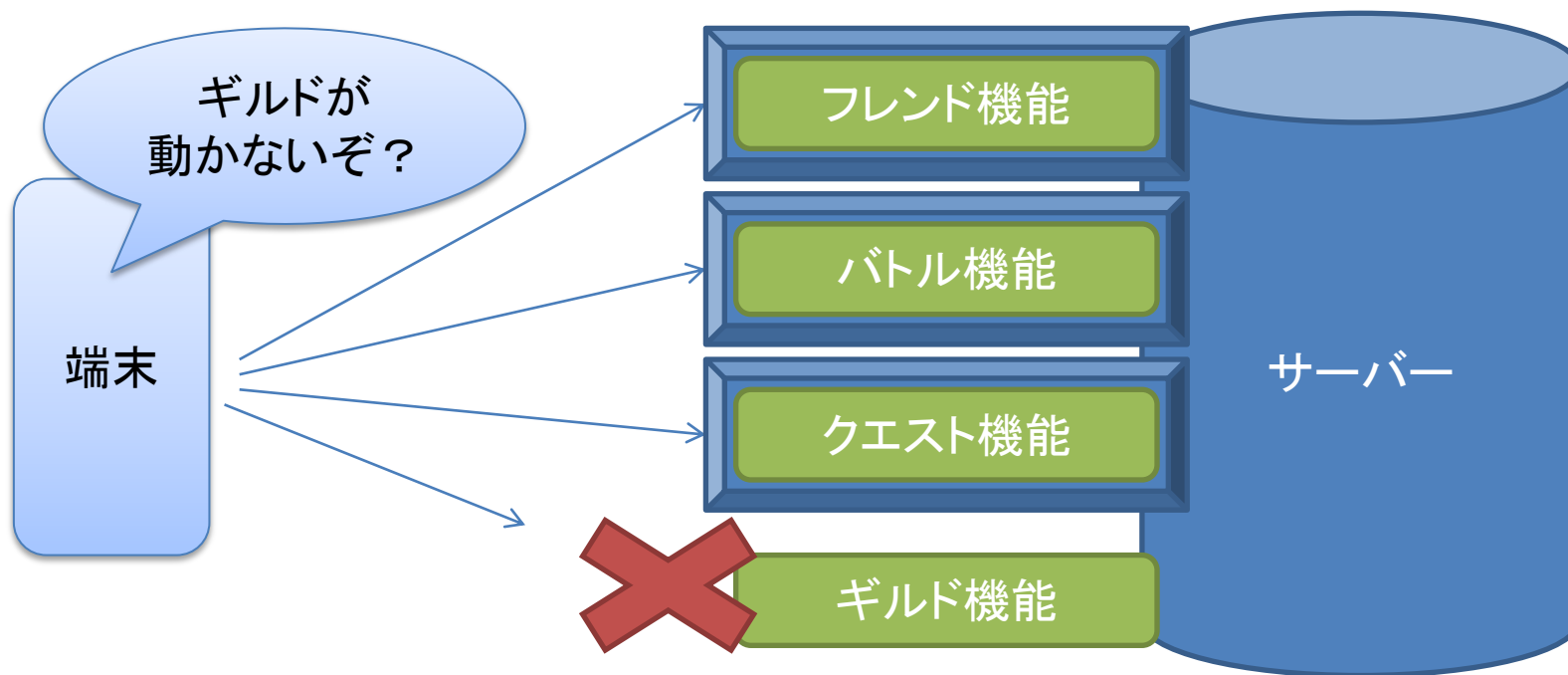


問題にいつ気づくか

- コンパイルタイムエラーが望ましいが、PHPやPerlでは難しい
- 攻撃されてみないと分からないのでは、絶対にまずい
- いかに**問題に気づきやすいか、簡単に直せるか**が重要
- フレームワークで**気づきやすく、直しやすく**してあげています

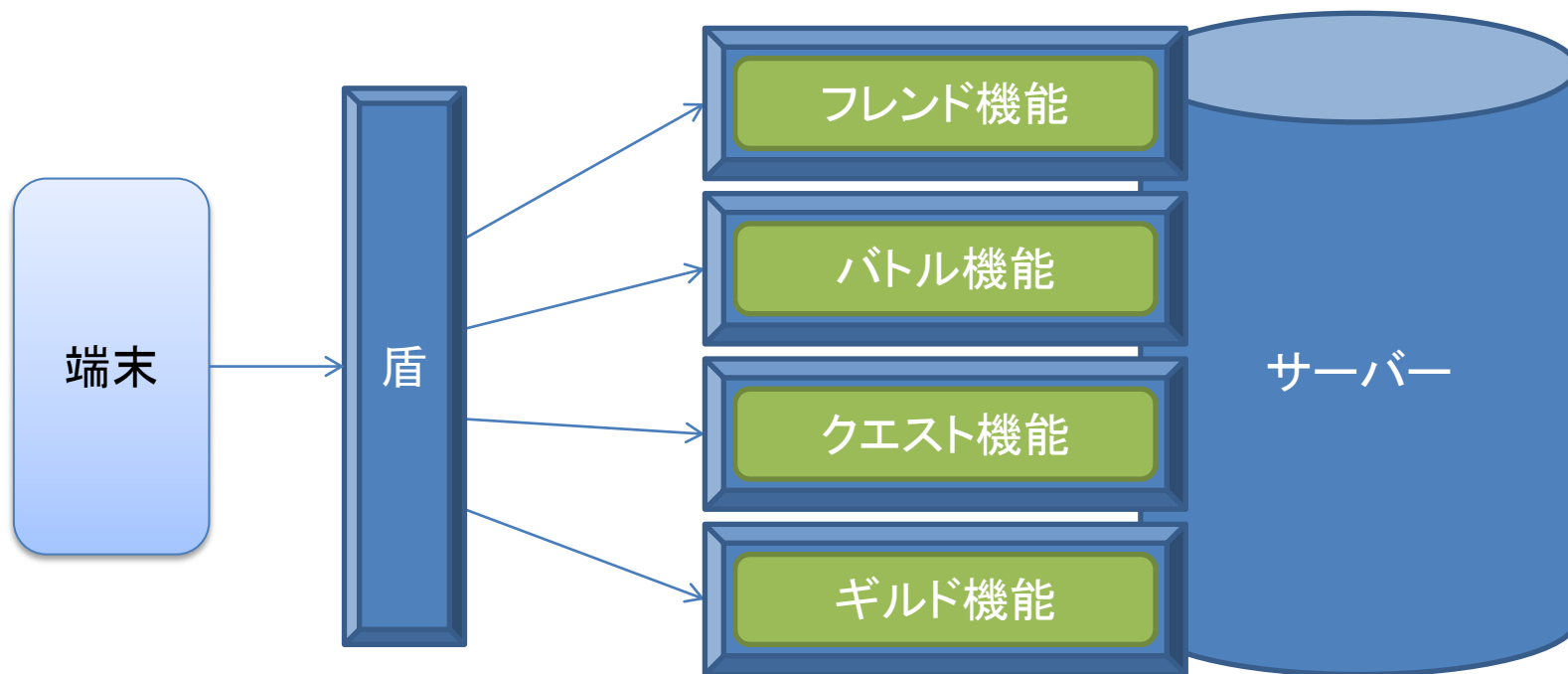
フレームワークでサポート

- そもそも、盾なしだと機能が動かないようにする（テストしてたら**必ず気づく**）



フレームワークでサポート

- 玄関を1つにして、全通信を監視する機能を提供。対策を**一括追加**できます



攻撃は入り口で防ごう

- 端末側ができる攻撃は「異常な通信内容の送信」のみです
- つまり、通信内容を入り口で精密検査すれば、大抵の攻撃は無効化できます
- ゲームロジックの手前で安全性を担保してあげれば、ロジック作りに集中できます

PHPの弱点をついた攻撃

- **整数**が送られてくるはずなのに、**小数**が送られてきた
- さらに**文字列**なことまで
- しかも**文字コードの壊れた**文字列だったり
- そもそもサーバーは**厳密さが重要**なのに、**型が無い**という**弱点**を攻撃されまくる！

MySQL + PHPの弱点ネタ

- MySQLは、整数型のカラムに小数を入れようとすると**四捨五入**します。しかも Warning とか出しません
- `(0 <= $n && $n < 3)` でチェックして、0 or 1 or 2 で制限をかけたつもりが…。**2.8** を送りつけられて、DB 上に **3** が代入される

自前で厳密にやるしかない

- 整数を受け取るときはGetInt()
それなのに小数や文字列を受信したら
チーター認定！ 通信シャットアウト！
- 文字列を受け取るときはGetString()
文字コードが壊れてたら**通信シャットアウト！**
- **疑わしきは罰せよ**

大切な2つの思想

- 対策は最初から。後からは無理
 - 機能全てに **1つもミスなく** 盾を足せる？
 - 全通信が玄関を通るように **作り直せる？**
 - 入力値の取得方法を **作り直せる？**
- 怪しい場合は安全側に倒そう
 - 実装が怪しかったら即停止して、プログラマに警告
 - ユーザーの通信内容が怪しかったら即打ち切り

ご清聴ありがとうございました