～Webアプリケーション開発講座～

セキュリティ対策基礎

UZUZ COLLEGE ウズウズカレッジ

～ セキュリティの重要性 ～

▼ InputSurvey（プロジェクト：Ex_5_04※）

※『5_04_WebアプリからのDB操作』の演習で作成

わんちゃん暮らし改善アンケートフォーム

名前： 悪いネコ

年齢： 99

性別： ◉ オス ○ メス

満足度： とても不満 ▼

飼い主へのご意見・ご感想をご記入ください：

```
<script type="text/javascript">
while(true){ alert("ここはネコが占拠した！"); }
</script>
```

回答する(SaveSurveyを起動)

★ 悪意ある書き込み ★

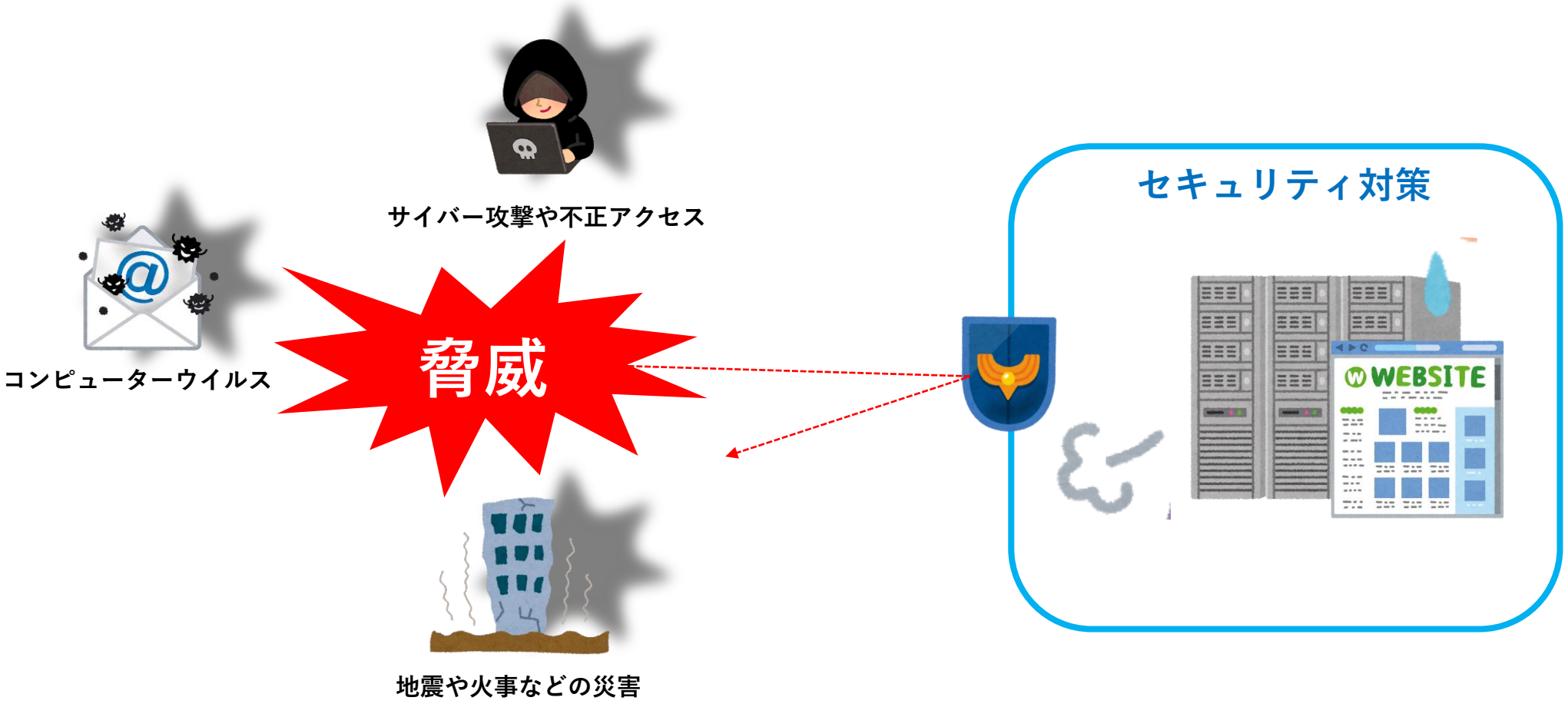
```
<script type="text/javascript">
while(true){
    alert("ここはネコが占拠した！");
}
</script>
```

※実行した場合はブラウザを閉じ、
該当の書き込みデータをDBからDELETE
してください。

～ セキュリティの重要性 ～

セキュリティ対策とは？

サービスの安定的な提供を脅かす**脅威**からWebサービスを守るための
仕組みを構築すること。



～ セキュリティ対策の三大原則 ～

セキュリティ対策の三大原則

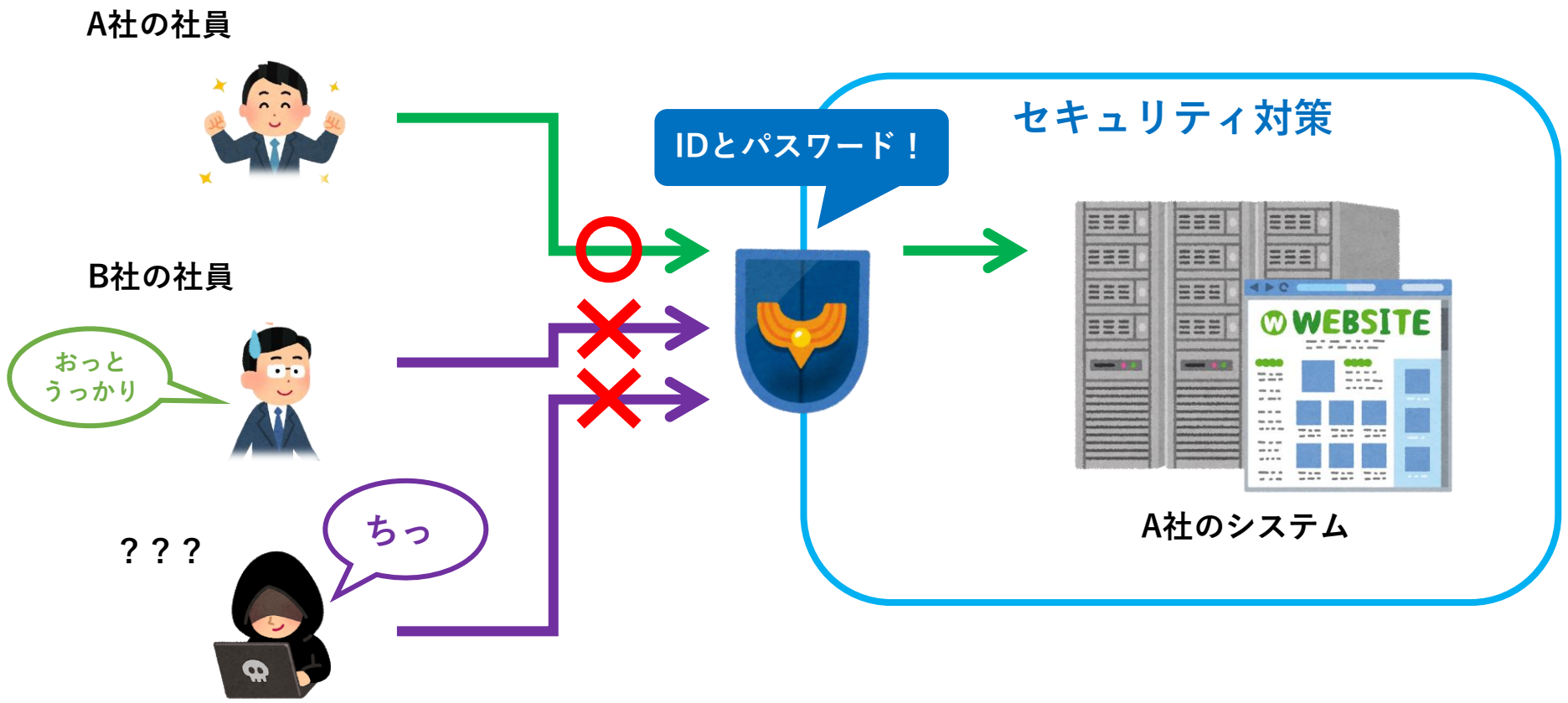
原則	詳細
機密性の確保 **（情報漏洩から守る）**	常に適切なユーザーだけがサービスを利用できる状態にしておくこと。 <対策例>ログイン管理、アクセス権の設定、セキュリティカード など
完全性の確保 **（データの改ざんから守る）**	不適切なユーザーによるデータの改ざんや不正な情報の混入を阻止する。 ＜対策例＞暗号化通信、不正なデータの検出・無効化 など
可用性の確保 **（サイバー攻撃や災害から守る）**	常にサービスが利用できる状態を保つ。 自然災害や停電によるサーバーの停止/ハードウェアの破損、 サイバー攻撃による障害や乗っ取りを事前に阻止する。 ＜対策例＞サイバー攻撃対策、システムの二重化 など

～ セキュリティ対策の三大原則 ～

①機密性の確保

常に適切なユーザーだけがサービスを利用できる状態にしておくこと。

＜例＞ログイン管理

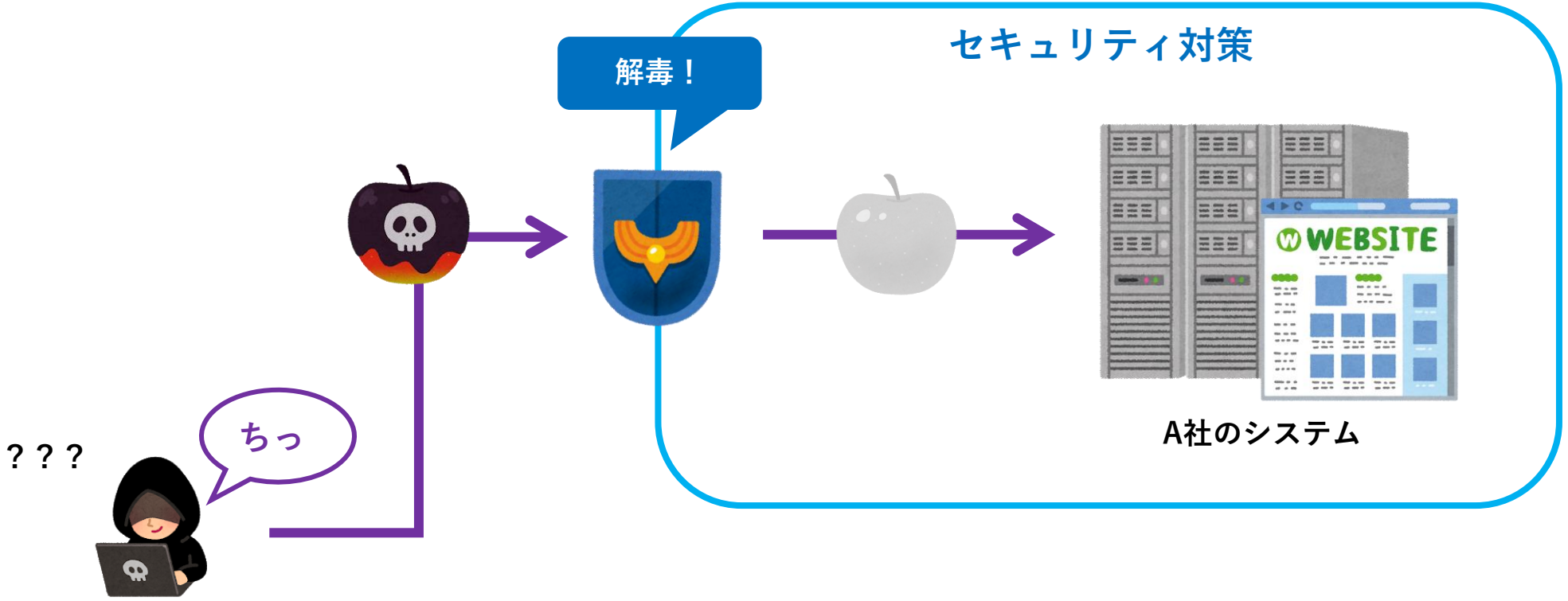


～ セキュリティ対策の三大原則 ～

②完全性の確保

不適切なユーザーによるデータの改ざんや不正な情報の混入を阻止する。

＜例＞不正データの検出・無効化

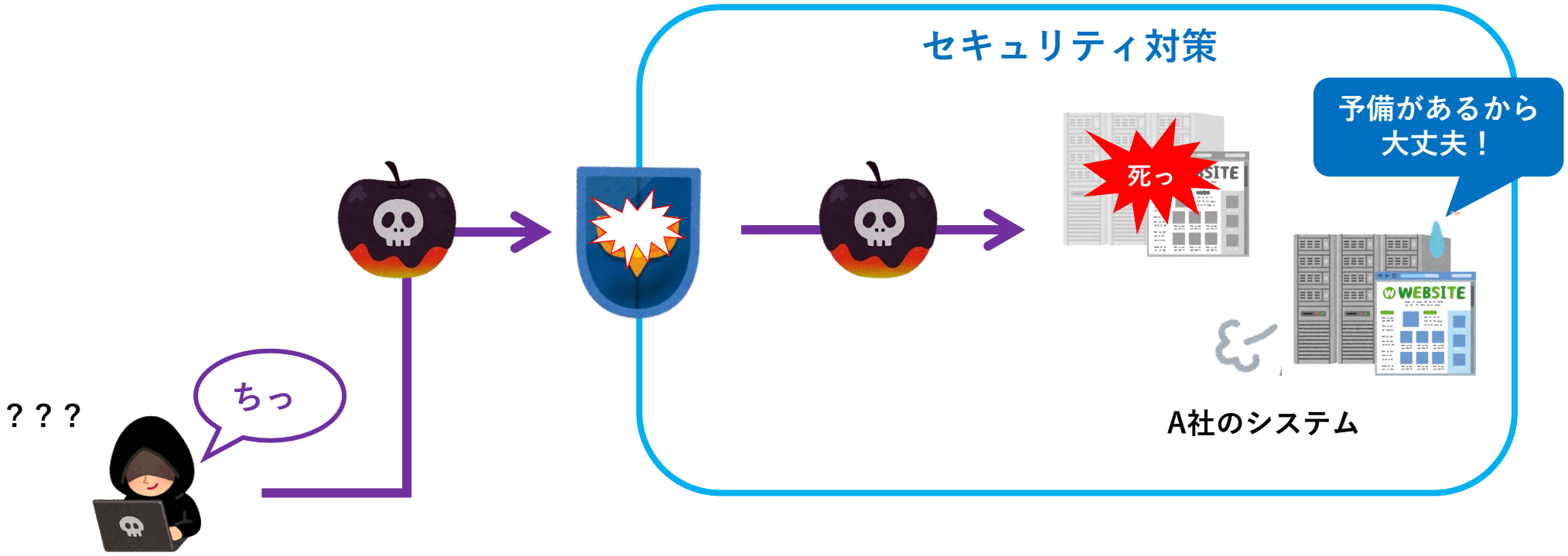


～ セキュリティ対策の三大原則 ～

③可用性の確保

常にサービスが利用できる状態を保つ。
自然災害や停電によるサーバーの停止/ハードウェアの破損、
サイバー攻撃による障害や乗っ取りを事前に阻止する。

＜例＞システムの二重化



～ 様々なサイバー攻撃とその対策 ～

サイバー攻撃の種類
・SQL インジェクション
・OS コマンド・インジェクション
・パス名パラメータの未チェック／ディレクトリ・トラバーサル
・セッション管理の不備
・クロスサイトスクリプティング
・CSRF（クロスサイト・リクエスト・フォージェリ）
・HTTP ヘッダ・インジェクション
・メールヘッダ・インジェクション
・クリックジャッキング
・バッファオーバーフロー
・アクセス制御や認可制御の欠落

「安全なウェブサイトの作り方（独立行政法人 情報処理推進機構）」より抜粋
URL： https://www.ipa.go.jp/files/000017316.pdf

～ 様々なサイバー攻撃とその対策 ～

＜ InputSurvey＞

※『5_04_WebアプリからのDB操作』の演習問題が終わっていること

わんちゃん暮らし改善アンケートフォーム

名前： 悪いネコ

年齢： 99

性別： ◉ オス ○ メス

満足度： とても不満 ▼

飼い主へのご意見・ご感想をご記入ください：

```
<script type="text/javascript">
while(true){ alert("ここはネコが占拠した！"); }
</script>
```

回答する(SaveSurveyを起動)

クロスサイトスクリプティング（XSS）

動的Webアプリケーションの脆弱性を利用したサイバー攻撃。
HTMLに悪意のあるスクリプトを埋め込むことによって内部のデータの漏洩や改ざん、管理権限の乗っ取り等を狙う。

～ 様々なサイバー攻撃とその対策 ～

クロスサイトスクリプティング（XSS）の対処法（エスケープ）

不正なスクリプトを埋め込むために利用されるHTML特殊文字を
特殊な意味を持たない文字列に変換することでスクリプトの動作を防ぐことを
エスケープと言います。

HTML特殊文字	エスケープ文字
<	&lt
>	&gt
&	&amp
"	&quot
'	&#039

不正なスクリプトで利用される代表的なHTML特殊文字とそのエスケープ文字

～ サンプルを動かしてみよう ～

手順

（１） eclipse上に新しい動的Webプロジェクトを作成する。
プロジェクト名： Sample_5_05_1

（２）プロジェクトの srcパッケージ直下に workパッケージを作成する。

（３）ドライブのSample_5_05_1フォルダからソースコードを取得して
workパッケージ直下にインポートする。

（４）同じくweb.xmlをドライブから取得して置き換える。
WebContent ＞ WEB-INF ＞ web.xml

（５）サーバーを起動してWebページを開く。

～ サンプルを動かしてみよう ～

クロスサイトスクリプティング（XSS）対策の詳細

★悪意ある書き込み

```
<h1><font color="#ff0000">
好き勝手するニャ！<br>by悪いネコ
</font></h1>
```

XSS対策のサンプル（送信画面）

①XSS対策あり

<h1>好き勝手するニャ！
by悪いネコ</h1>

②XSS対策なし

<h1>好き勝手するニャ！
by悪いネコ</h1>

受信画面へ

XSS対策のサンプル（受信画面）

①XSS対策あり（入力値をそのまま表示）

<h1>好き勝手するニャ！
by悪いネコ</h1>

②XSS対策なし

好き勝手するニャ！
by悪いネコ

送信画面に戻る

～ サンプルを動かしてみよう ～

クロスサイトスクリプティング（XSS）対策の詳細

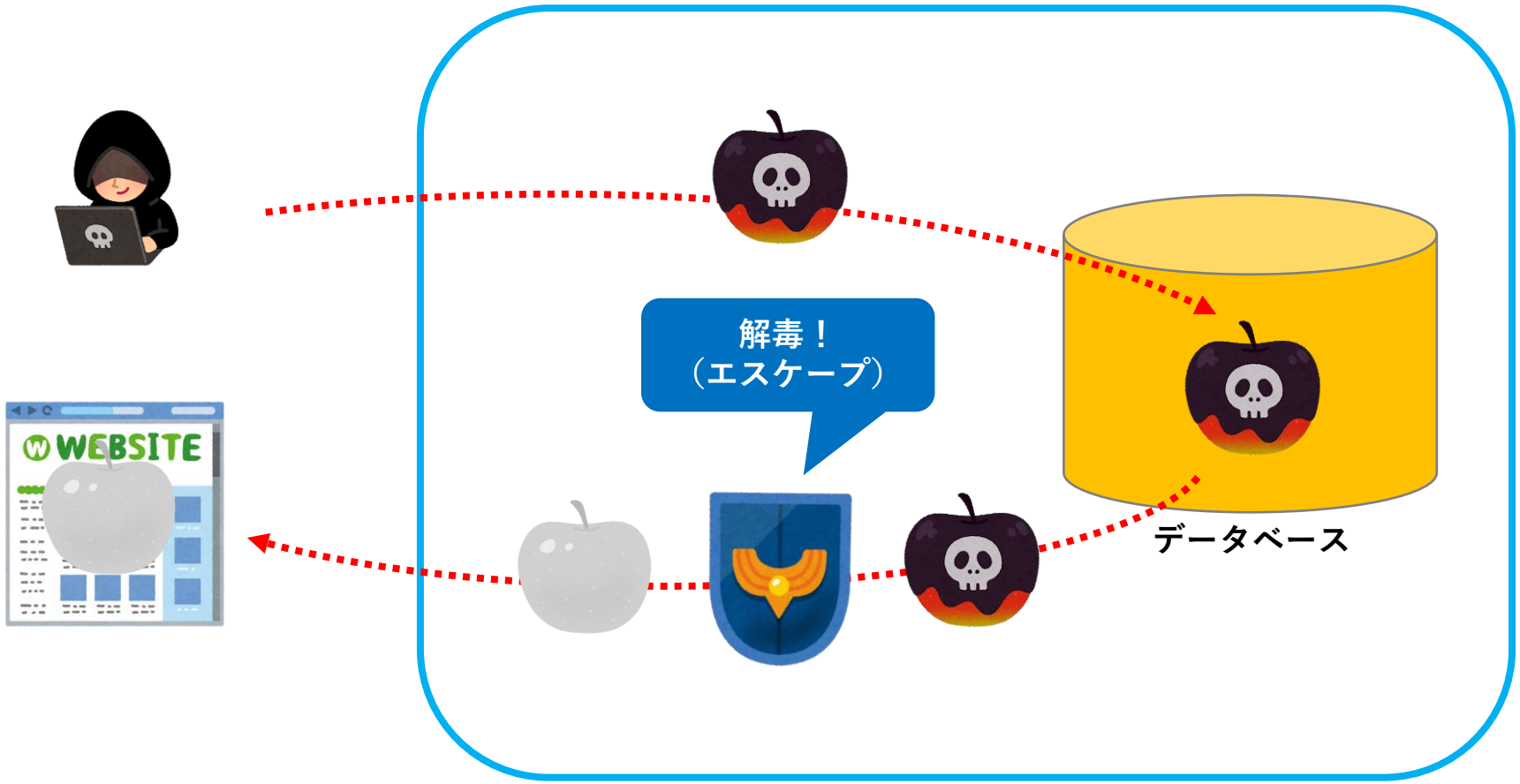
```
//XSSに使用される特殊文字をエスケープ処理する
private String replaceEscapeChar(String inputText) {

    // 「&」を変換して無害化
    inputText = inputText.replace("&", "&amp;");
    // 「<」を変換して無害化
    inputText = inputText.replace("<", "&lt;");
    // 「>」を変換して無害化
    inputText = inputText.replace(">", "&gt;");
    // 「"」を変換して無害化
    inputText = inputText.replace("\"", "&quot;");
    // 「'」を変換して無害化
    inputText = inputText.replace("'", "&#039;");
```

Stringクラスのreplaceメソッドで特殊文字をエスケープ文字に変換！

～ エスケープのタイミングについて ～

エスケープのタイミングは “出力” の時に！

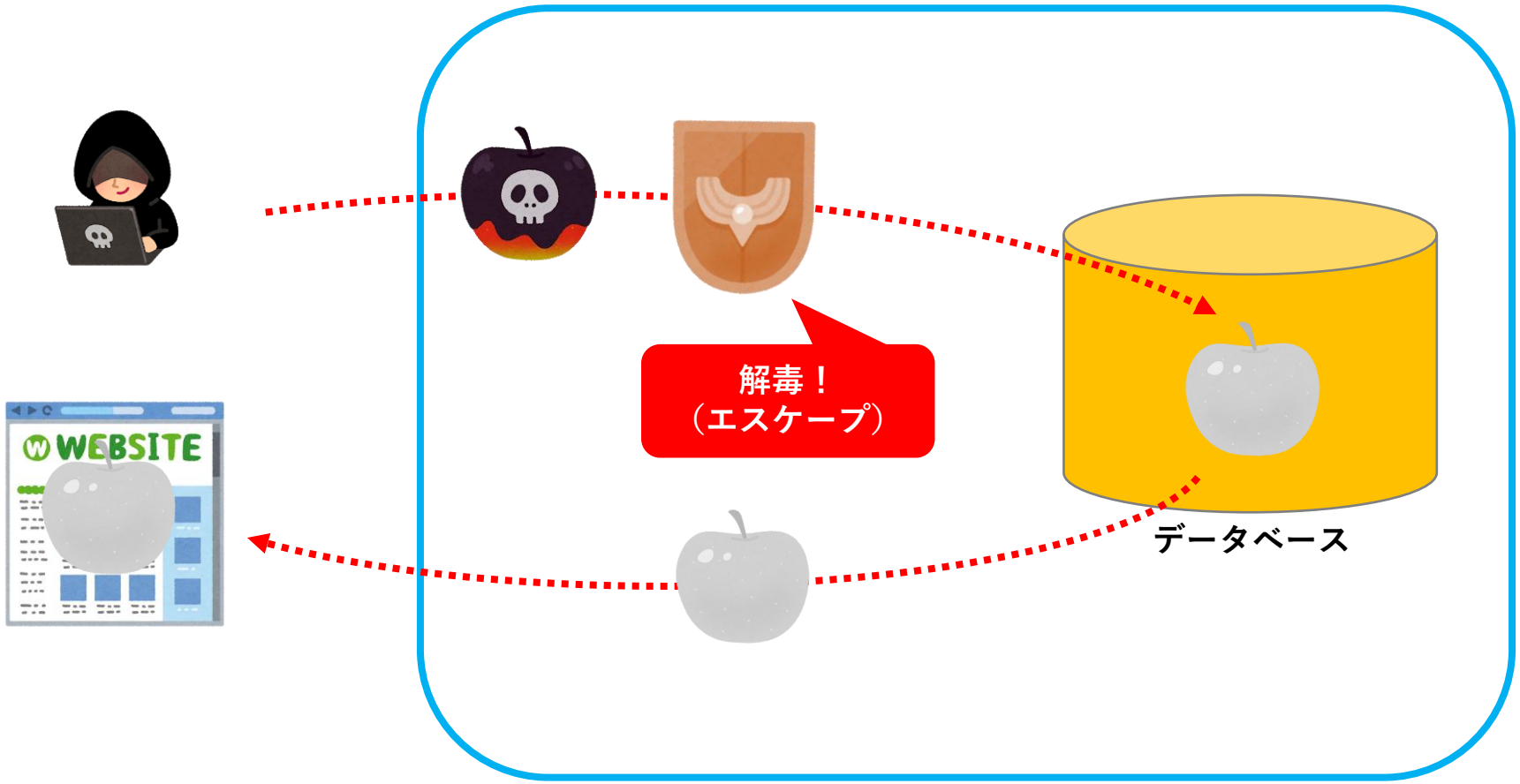


ユーザーが入力したデータを
無暗に加工しちゃダメ！
必要な時だけ必要な処理を！

OKなパターン

～ エスケープのタイミングについて ～

エスケープのタイミングは “出力” の時に！



ダメなパターン

【演習】

プロジェクト「Ex_5_04」にXSS対策を施して
セキュリティの高いページに作り替えましょう！

以下のポイントに気をつけてね!

- XSS対策を行うタイミングはどこか適切かな？
- どの項目に対してXSS対策を行えばいいかな？

【演習】

手順

（１） eclipse上に新しい動的Webプロジェクトを作成する。
プロジェクト名：Ex_5_05

（２）プロジェクトの srcパッケージ直下に workパッケージを作成する。

（３）プロジェクト「 Ex_5_04 」を参考に演習に取り組む。
・Javaリソース＞src＞work 直下にjavaソースコードを配備
・WebContent＞WEB-INF 直下にweb.xmlを配備