～Webアプリケーション開発講座～

# Cookieとセッション

UZUZ COLLEGE ウズウズカレッジ

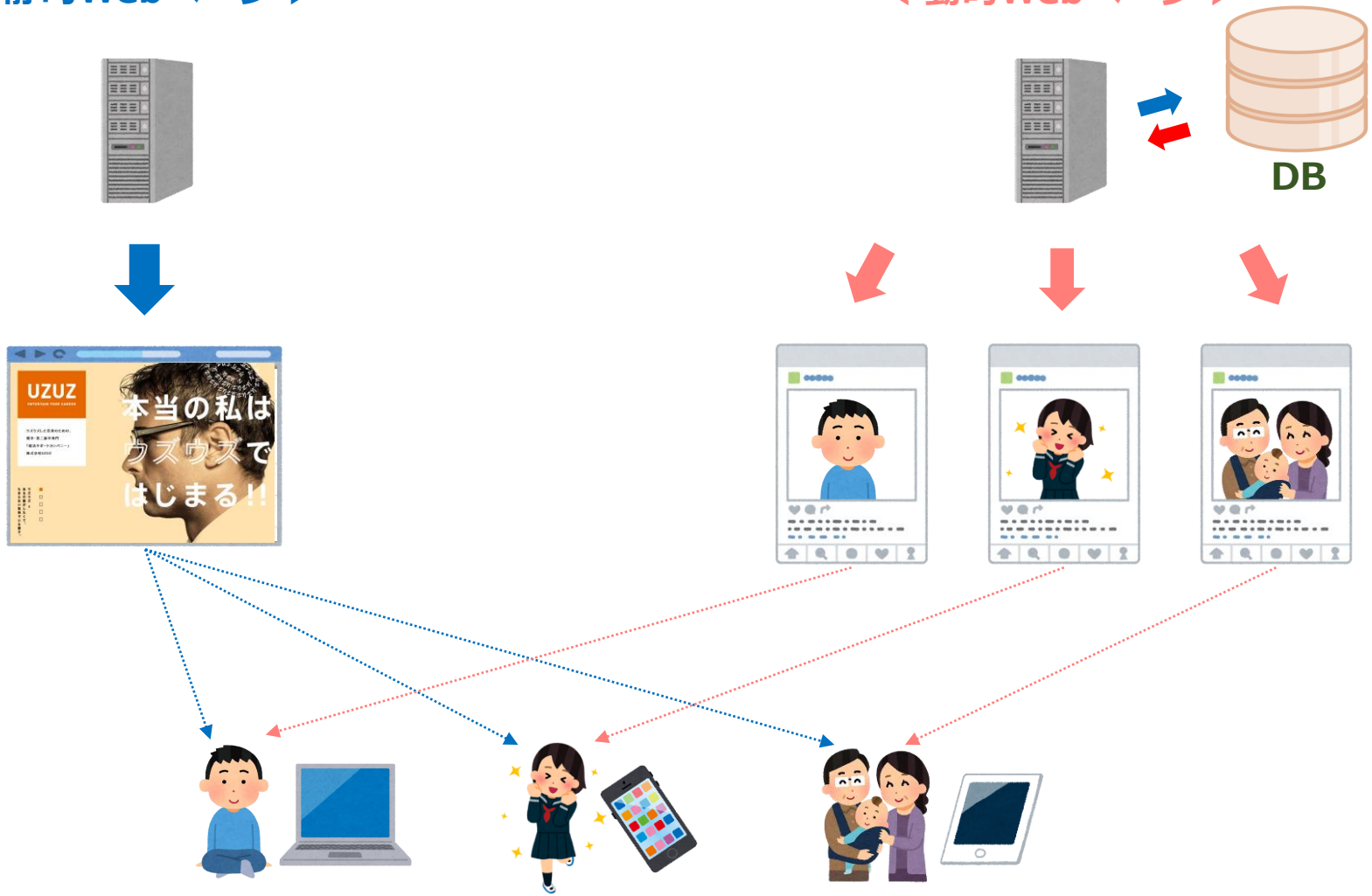
# ～データベースが必要なWebアプリケーション～

復習

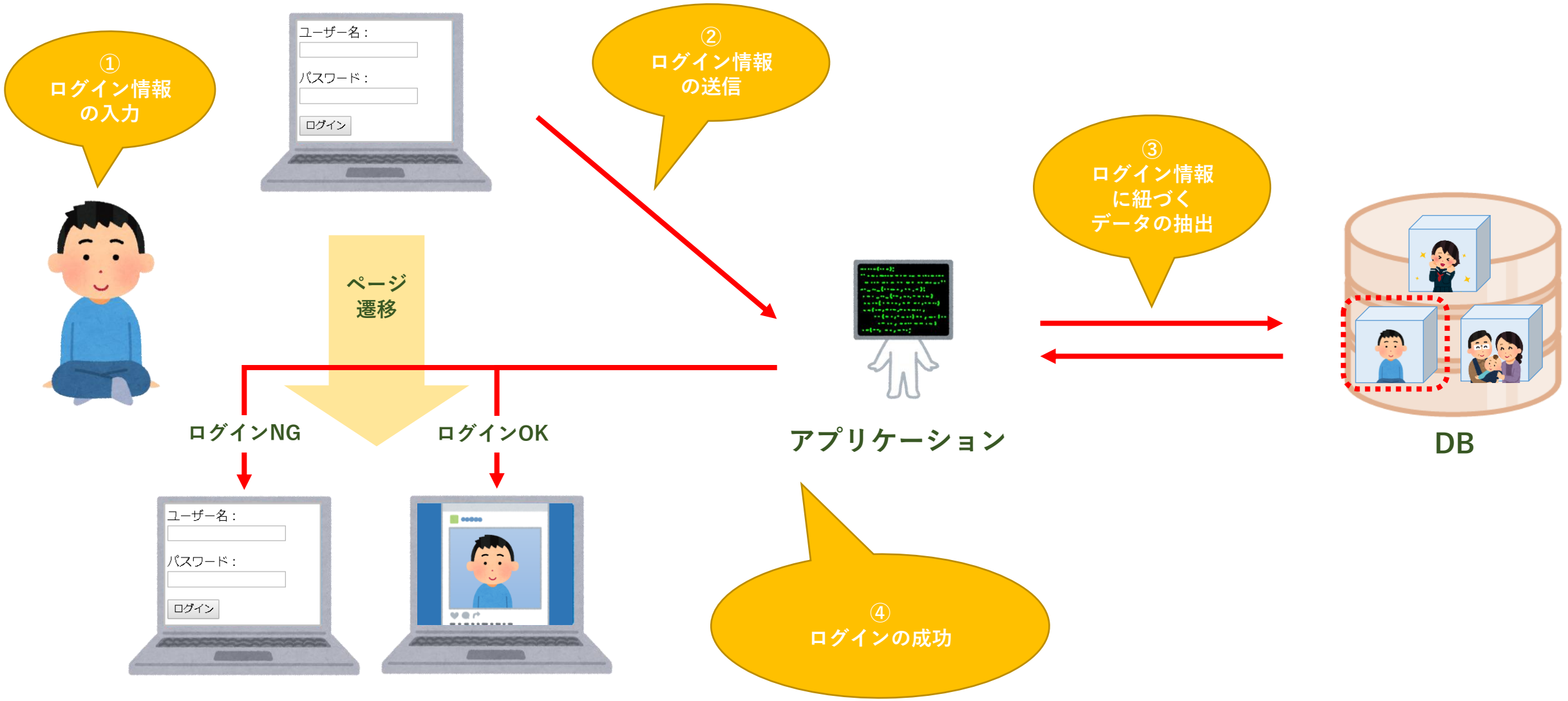
## 動的Webページ ／ 静的Webページ

＜ 静的Webページ ＞

＜ 動的Webページ ＞



# ～ ログイン管理を行う方法 ～



# ～ 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_06_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）配布のSample_5_06_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。

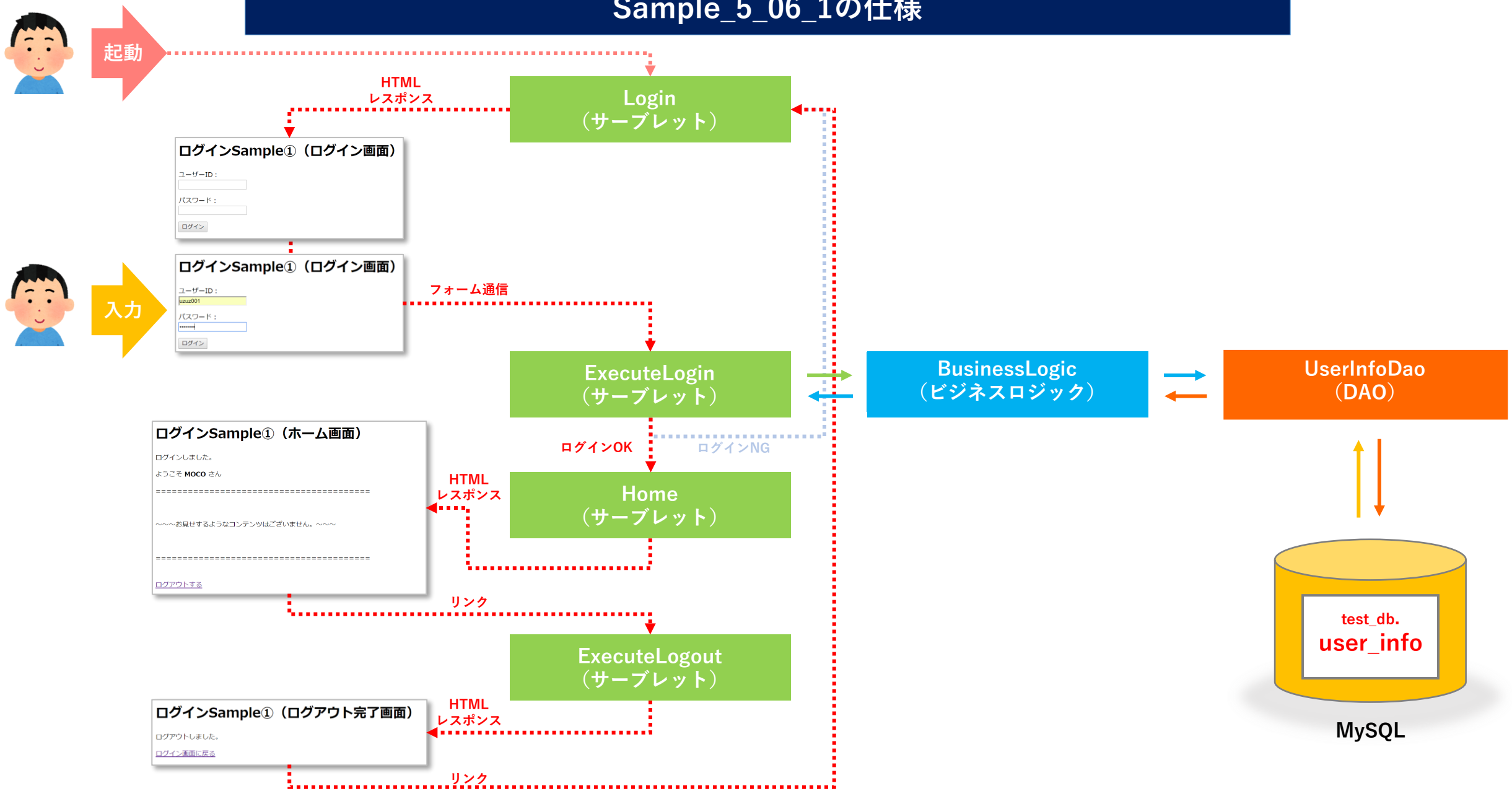
（４）同じくweb.xmlをドライブから取得して置き換える。
WebContent ＞ WEB-INF ＞ web.xml

（５）CREATE_USER_INFO.sql内のSQLを実行してテーブルを作成する
※是非INSERT文を書き換えて自身のユーザーも登録してみて下さい。

（６）サーバーを起動してLogin.javaを起動する。

# ～ サンプルを動かしてみよう ～

## Sample_5_06_1の仕様



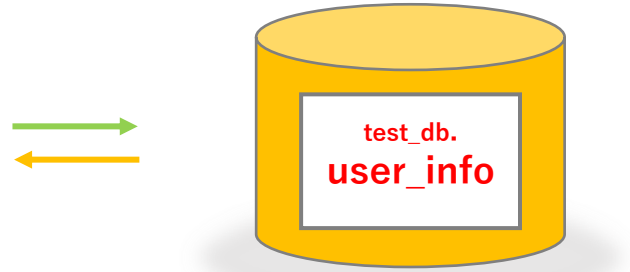
# ～ サンプルを動かしてみよう ～


リクエスト

ExecuteLogin

doPost

DBからのデータ取得
&
リダイレクト
（Home?user_name=MOCO）




MySQL



リダイレクト

Home

doGet

リクエストパラメータの取得
&
画面の出力


レスポンス

サーブレットからサーブレットに
データを受け渡すため、
URLにリクエストパラメータを埋め込んで
リダイレクトする！

# ～ Sample_5_06_1は不完全・・・ ～


リクエスト

リダイレクト


レスポンス

**ExecuteLogin**

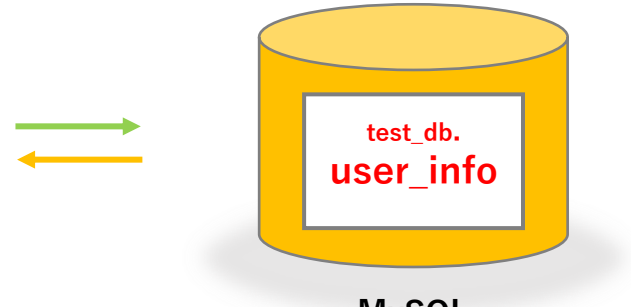
doPost

DBからのデータ取得
&
リダイレクト
（ Home?user_name=MOCO ）

**Home**

doGet

リクエストパラメータの取得
&
画面の出力

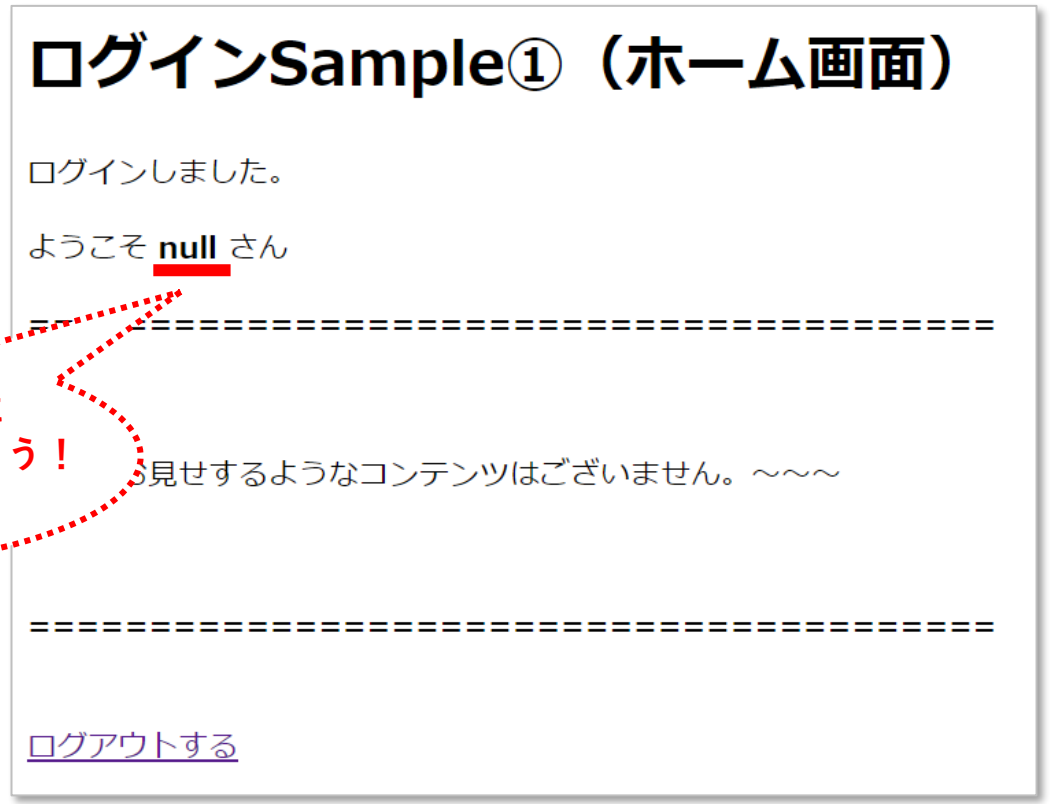

MySQL

サーブレットからサーブレットに
データを渡すため、
URLにリクエストパラメータを埋め込んで
リダイレクトする！

**サーバーの外は危険がいっぱい！**
**大事なデータを外に出したくない！**

# ～ Sample_5_06_1は不完全・・・ ～

ログインせずに次のURLにアクセスすると…？
http://localhost:8080/Sample_5_06_1/Home




ログインしていないのに
ホーム画面が表示できてしまう！
（ユーザー名はnull）

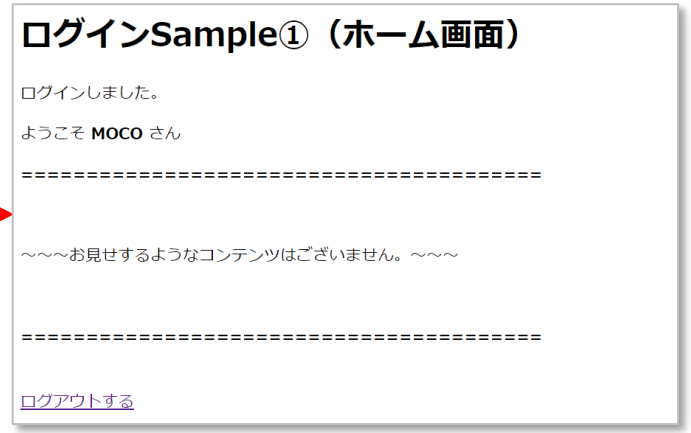
ログインしていないなら
TOP画面（ログイン画面）に
飛ばしたいところ…！

ログインしているなら
ちゃんと表示したい。

# ～ Sample_5_06_1は不完全・・・ ～

ログインせずに次のURLにアクセスすると…？
http://localhost:8080/Sample_5_06_1/Home

ログインしている
or
ログインしていない




ホーム画面
（ログイン済み）


ログイン画面

# ～ 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_06_2

（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）配布のSample_5_06_2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Login.javaを起動する。

# ～ サンプルを動かしてみよう ～

## ログイン機能の確認

**①ログイン前の状態で次のURLにアクセス。**

- http://localhost:8080/Sample_5_06_2/Login
- http://localhost:8080/Sample_5_06_2/ExecuteLogin
- http://localhost:8080/Sample_5_06_2/Home
- http://localhost:8080/Sample_5_06_2/ExecuteLogout

**→ ログイン画面が表示される。**

**②ログイン後の状態で次のURLにアクセス。**

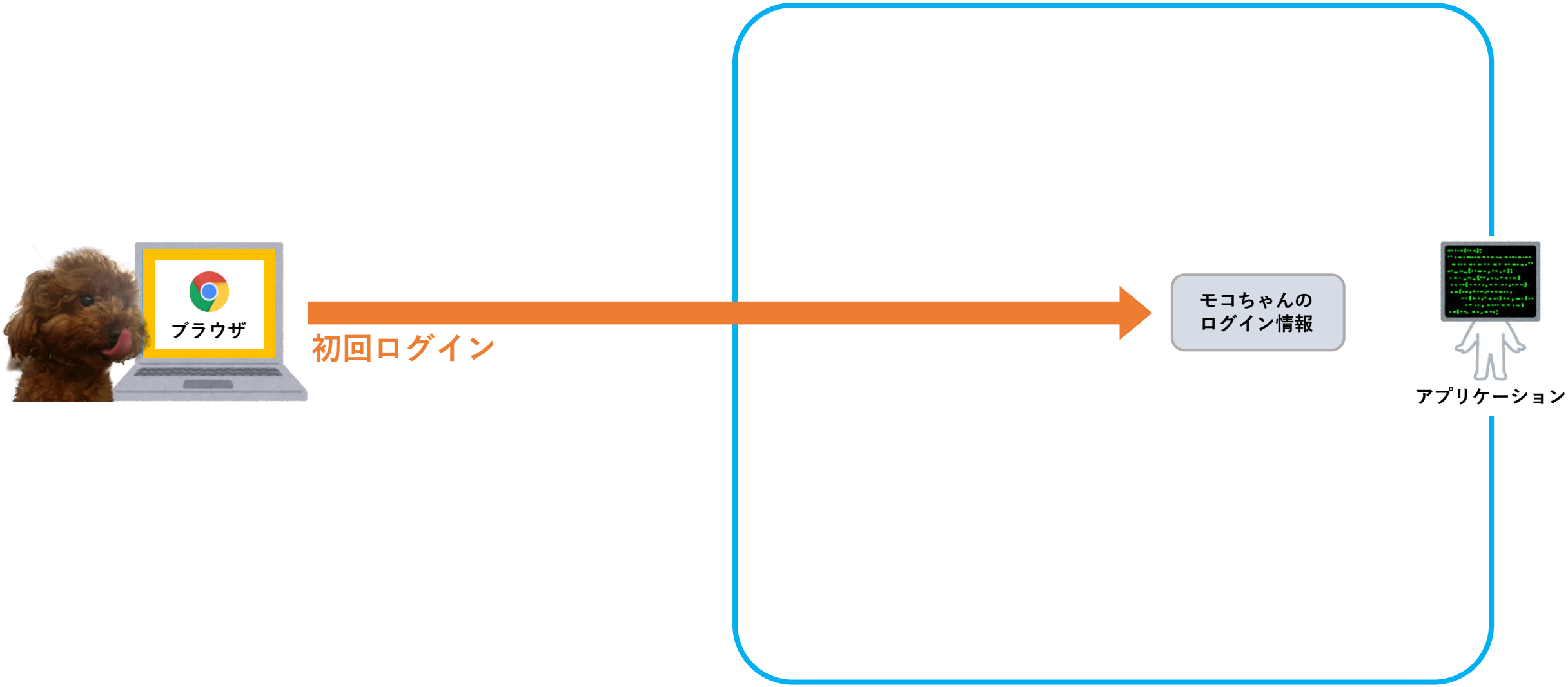
- http://localhost:8080/Sample_5_06_2/Login
- http://localhost:8080/Sample_5_06_2/ExecuteLogin
- http://localhost:8080/Sample_5_06_2/Home

**→ ホーム画面が表示される。**

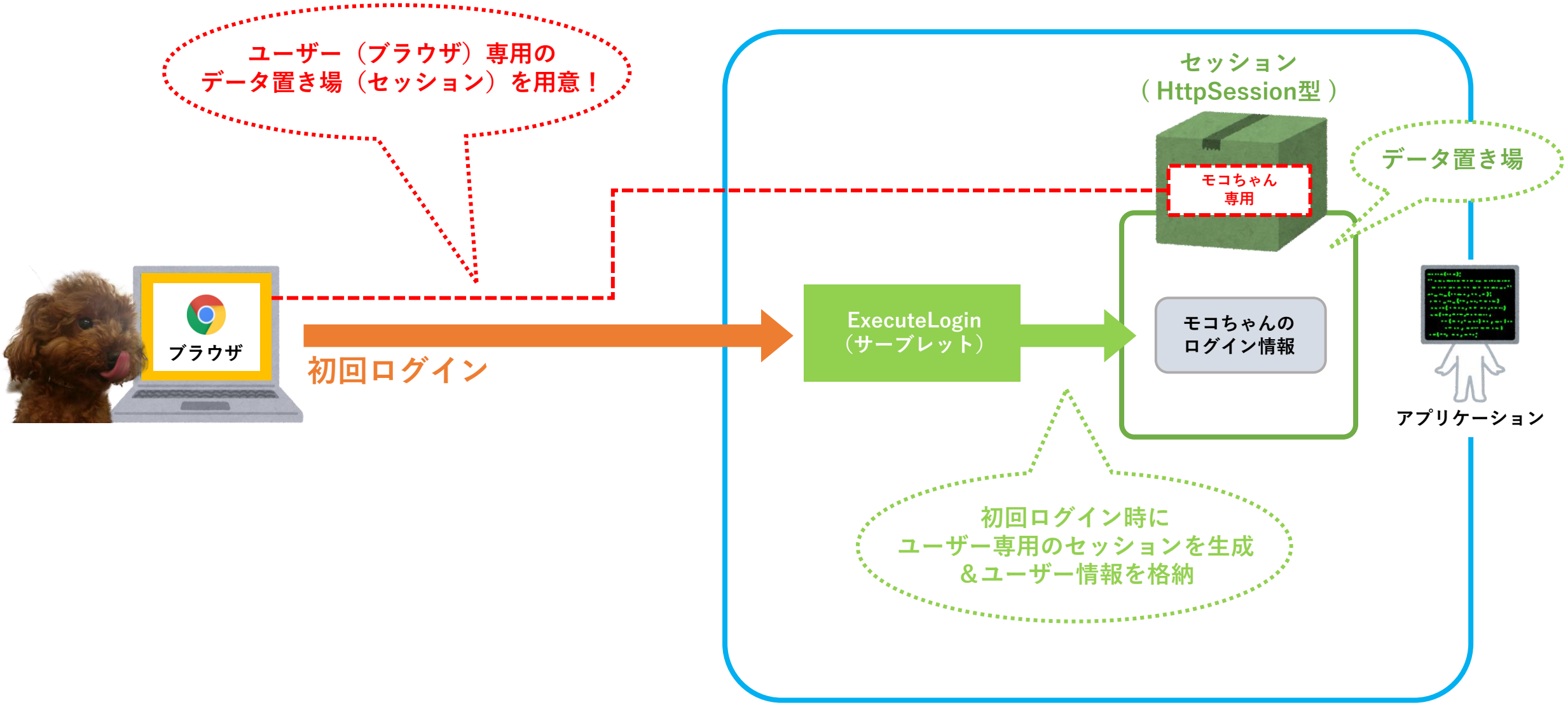
- http://localhost:8080/Sample_5_06_2/ExecuteLogout

**→ ログアウト完了画面が表示される。**

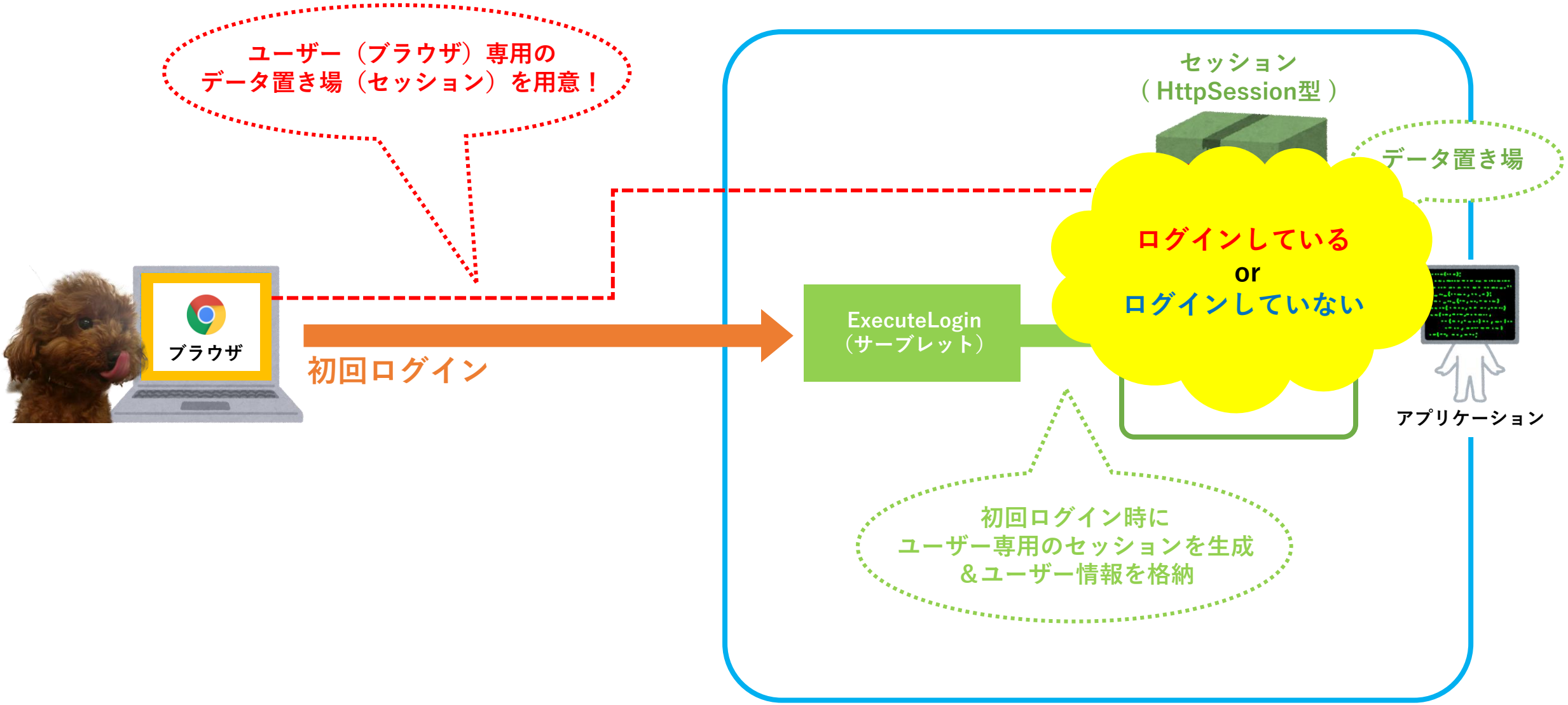
# ～ Cookie／セッションの必要性 ～



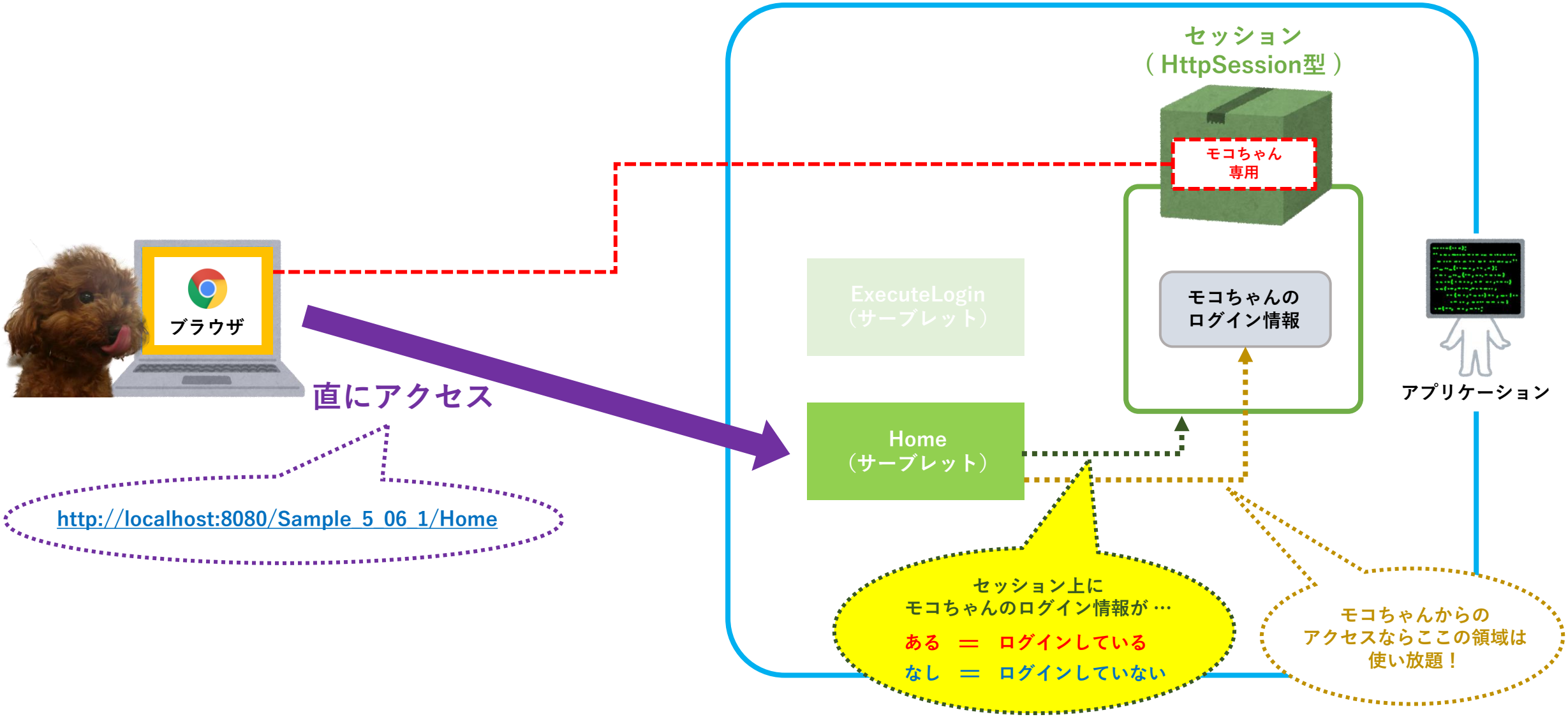
# ～ Cookie／セッションの必要性 ～



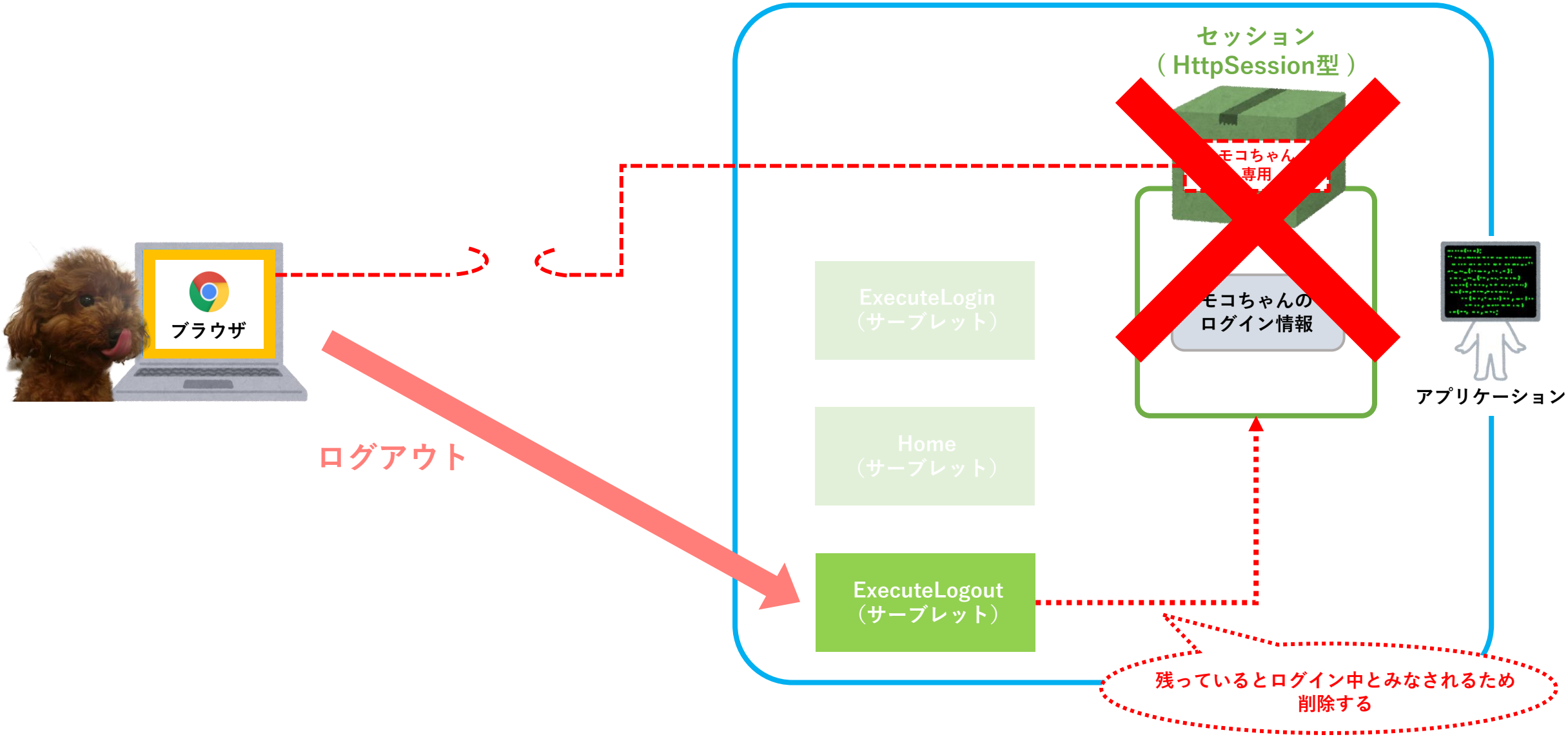
# ～ Cookie／セッションの必要性 ～



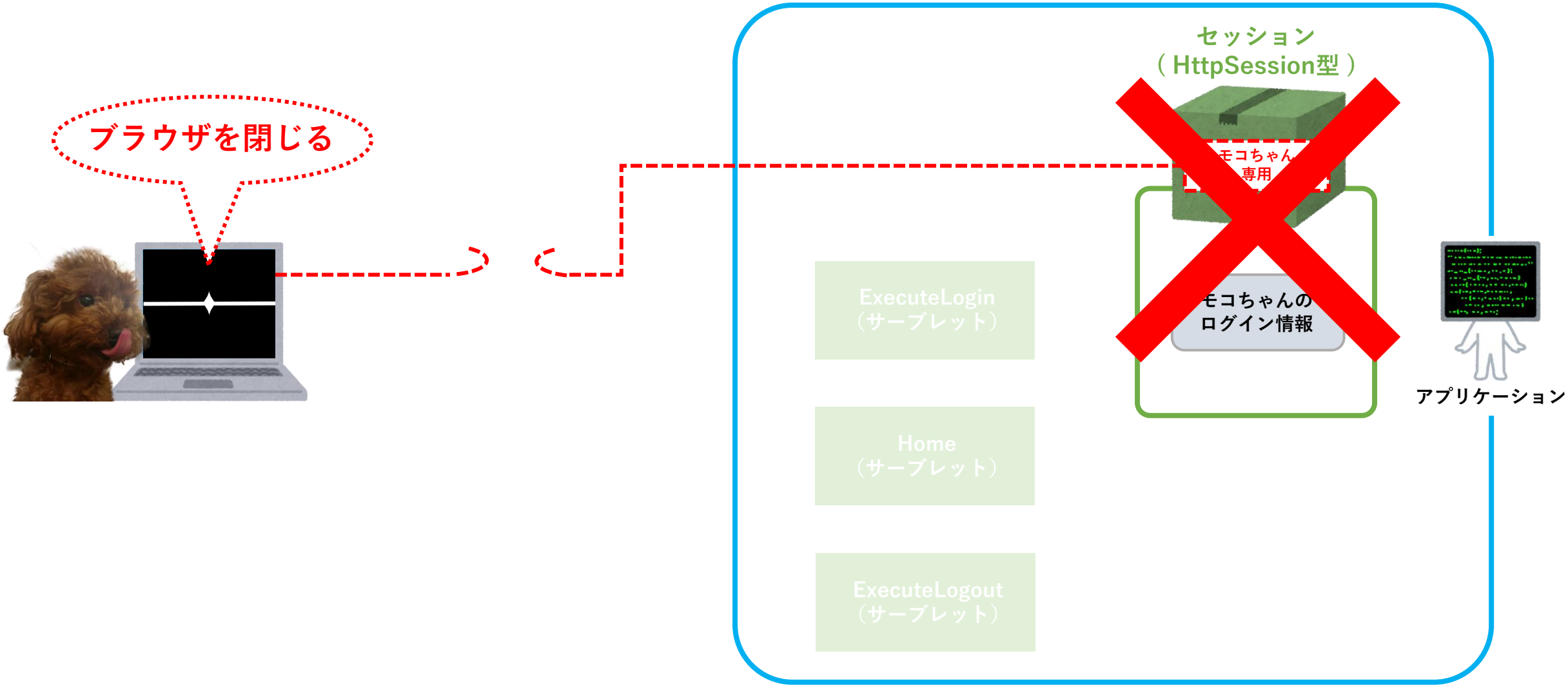
# ～ Cookie／セッションの必要性 ～



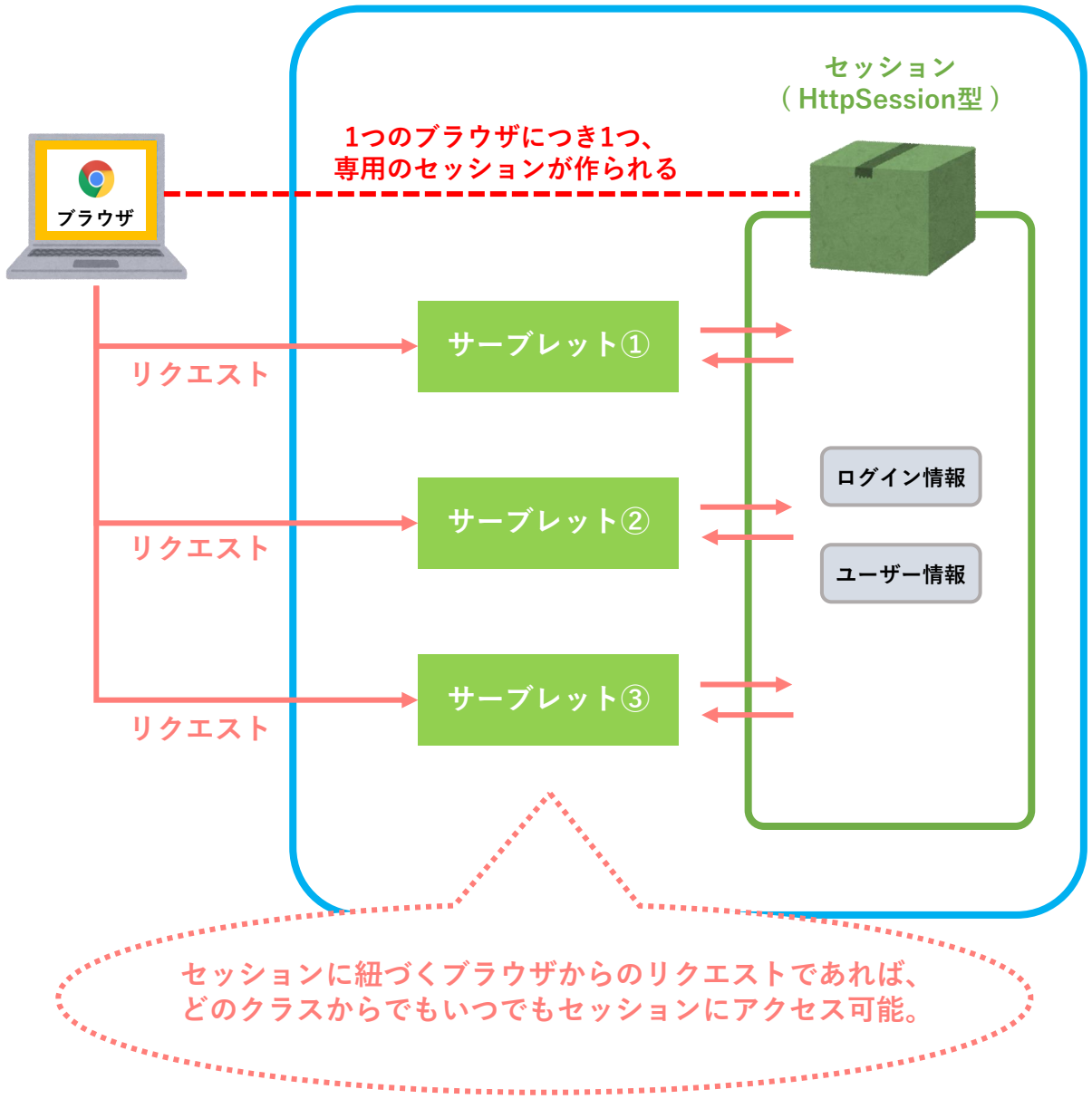
# ～ Cookie／セッションの必要性 ～



# ～ Cookie／セッションの必要性 ～



# ～ Cookie／セッションの必要性 ～




## ≪セッション≫

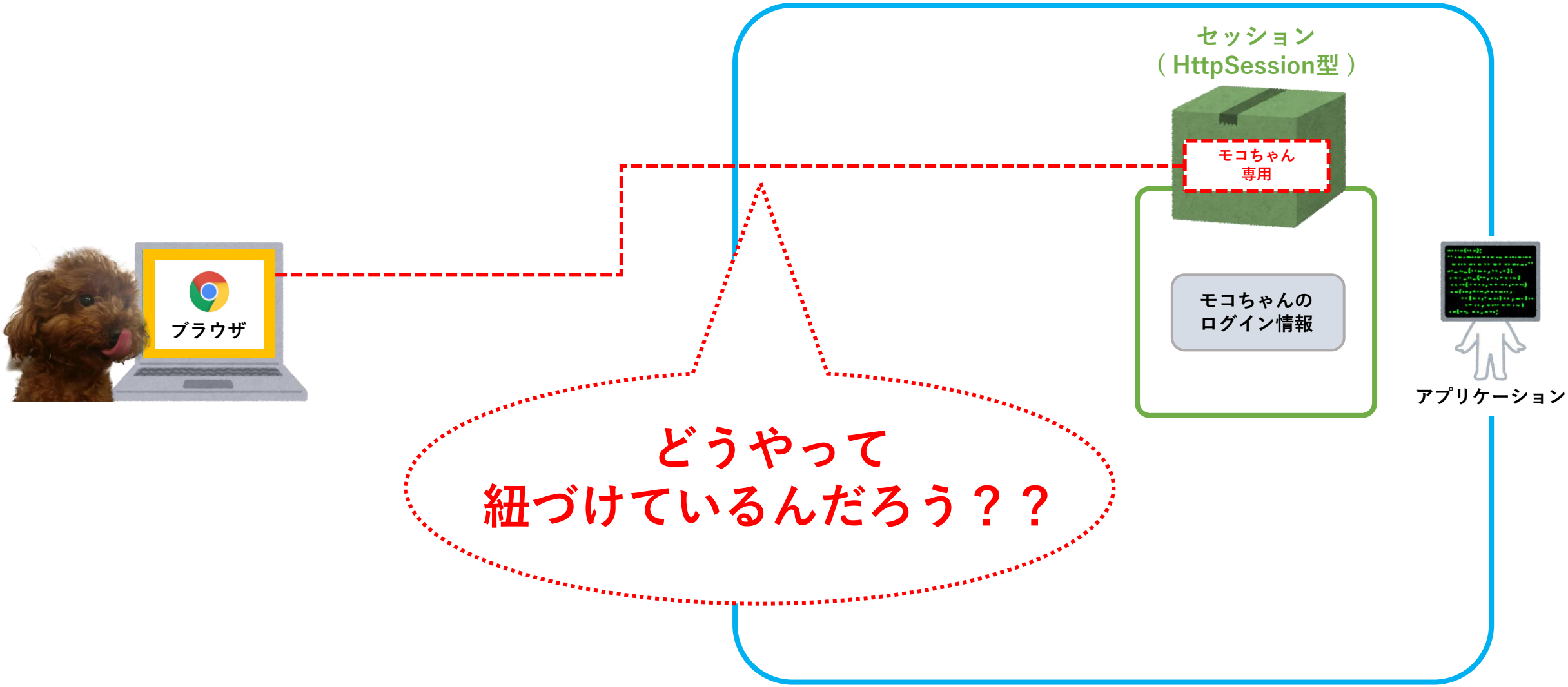
□同じブラウザからのアクセスに限り、リクエストをまたいで使用することのできるインスタンスのことを**セッション**と言います。
ログイン情報やユーザー情報など、リクエストをまたいで保持しておきたいユーザー固有の情報を格納しておくと便利です。

□セッションはブラウザ固有の存在なので、**ブラウザが閉じられるとセッションも消去されます。**セッションが不要になった際はinvalidateメソッドを使用して明示的に消去することが可能です。
また、一定時間経過で消去する（**セッションタイムアウト**）設定も施すことが可能です。

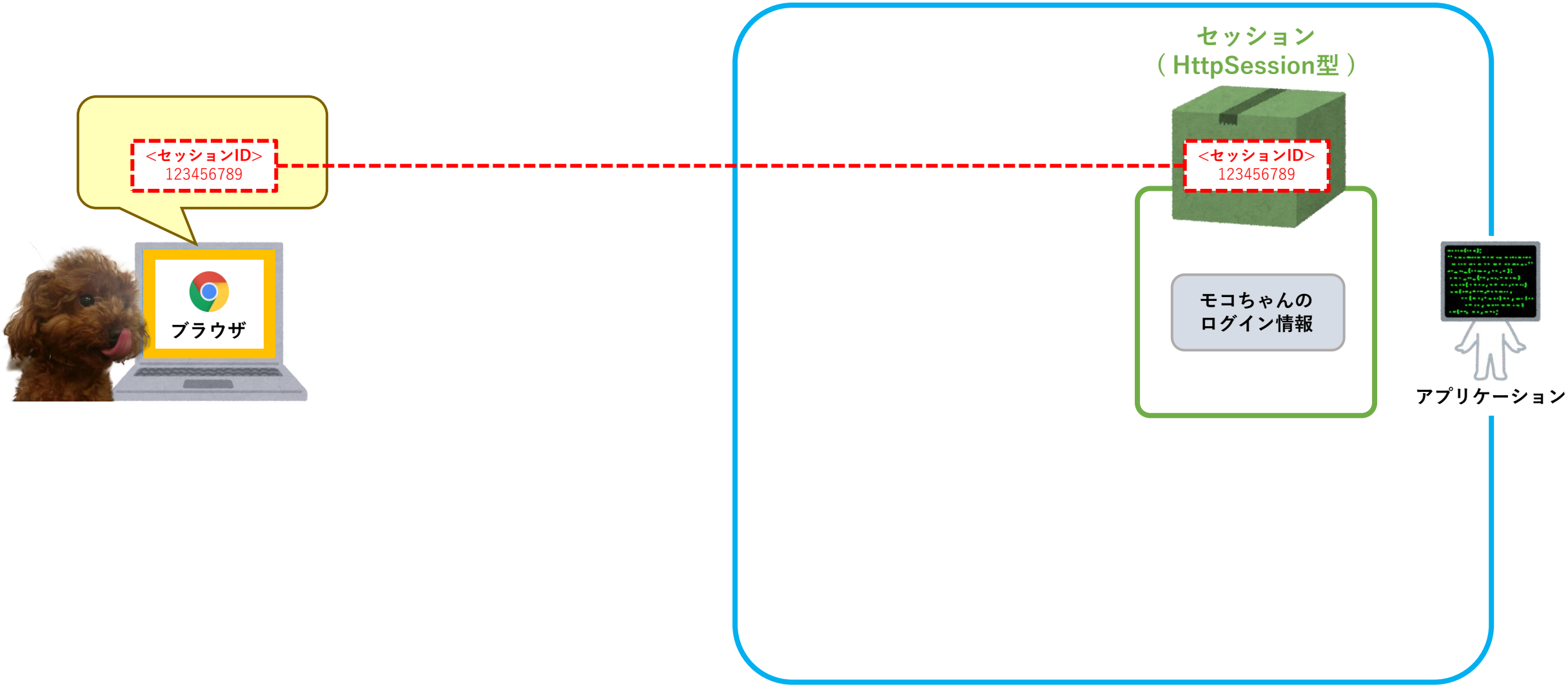
□以下の要領で使用します。

- セッションの生成
  HttpSession session ＝ request.getSession() ;
- セッションへのデータの保存
  session.setAttribute(“属性名“ , 保存したいインスタンス) ;
  ※保存できるのはインスタンスオブジェクトのみです。プリミティブ型変数は保存できないのでラッパークラスに変換するなどの工夫が必要です。
- セッション上のデータの取得
  (取得するインスタンスの型)session.getAttribute(“属性名“)
  ※getAttributeの戻り値は Object型 であるため、必ず取得するインスタンスの型でキャストする必要があります。
- セッションの消去
  session.invalidate() ;

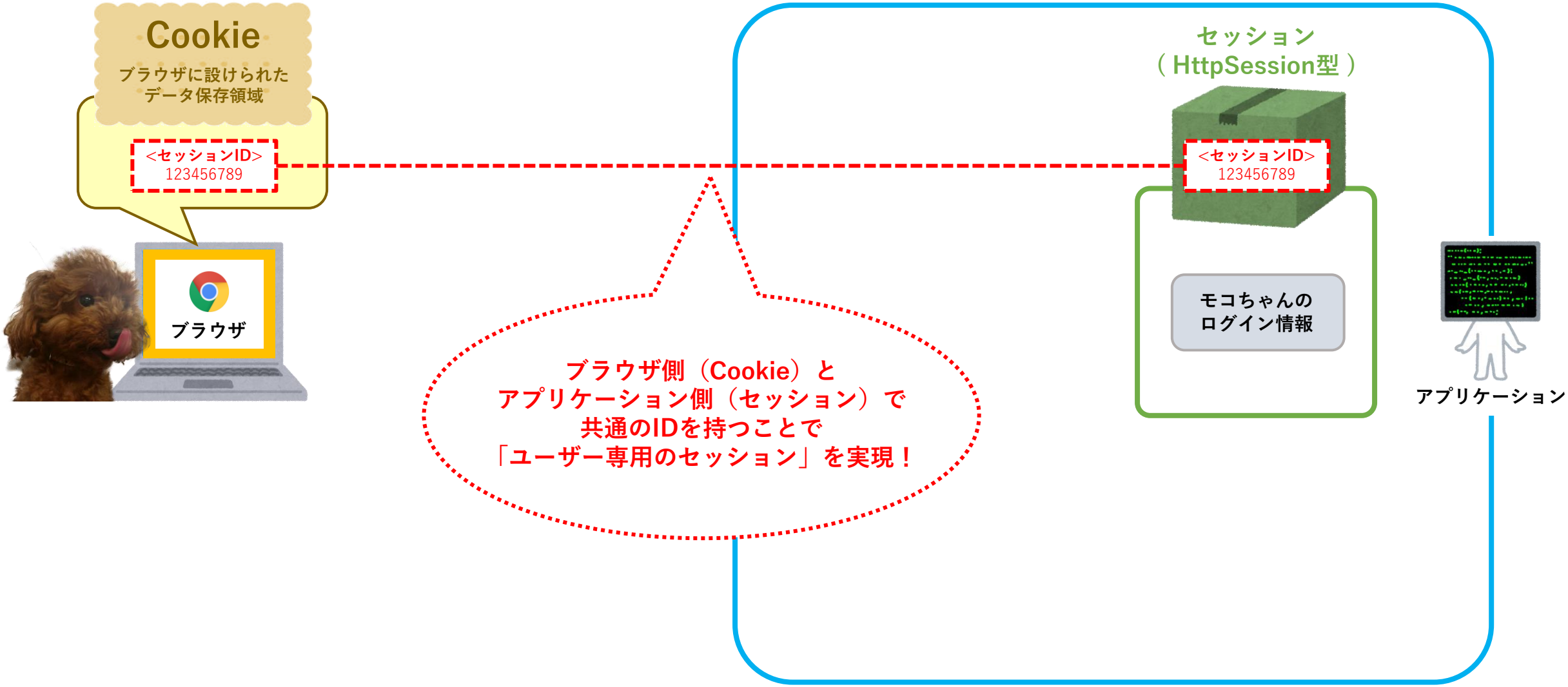
# ～ Cookie／セッションの必要性 ～



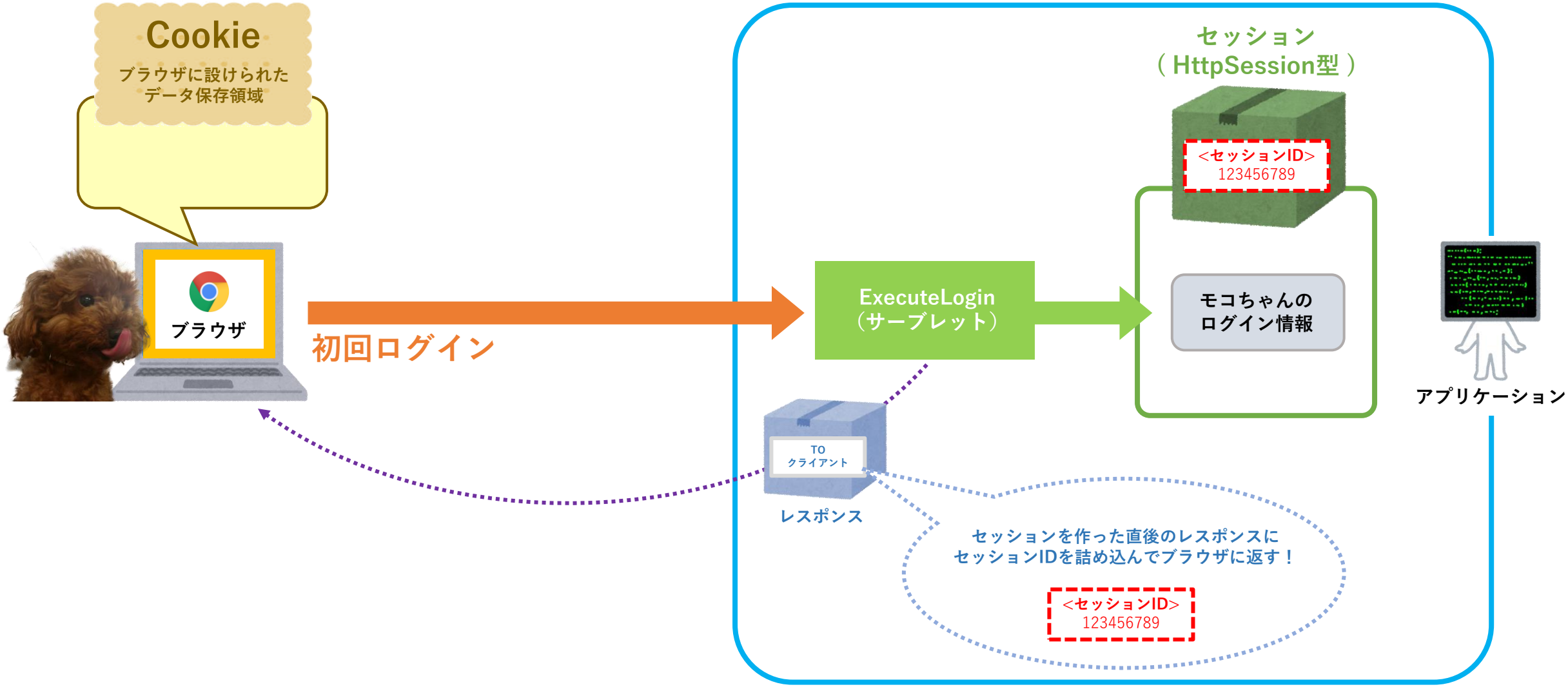
# ～ Cookie／セッションの必要性 ～



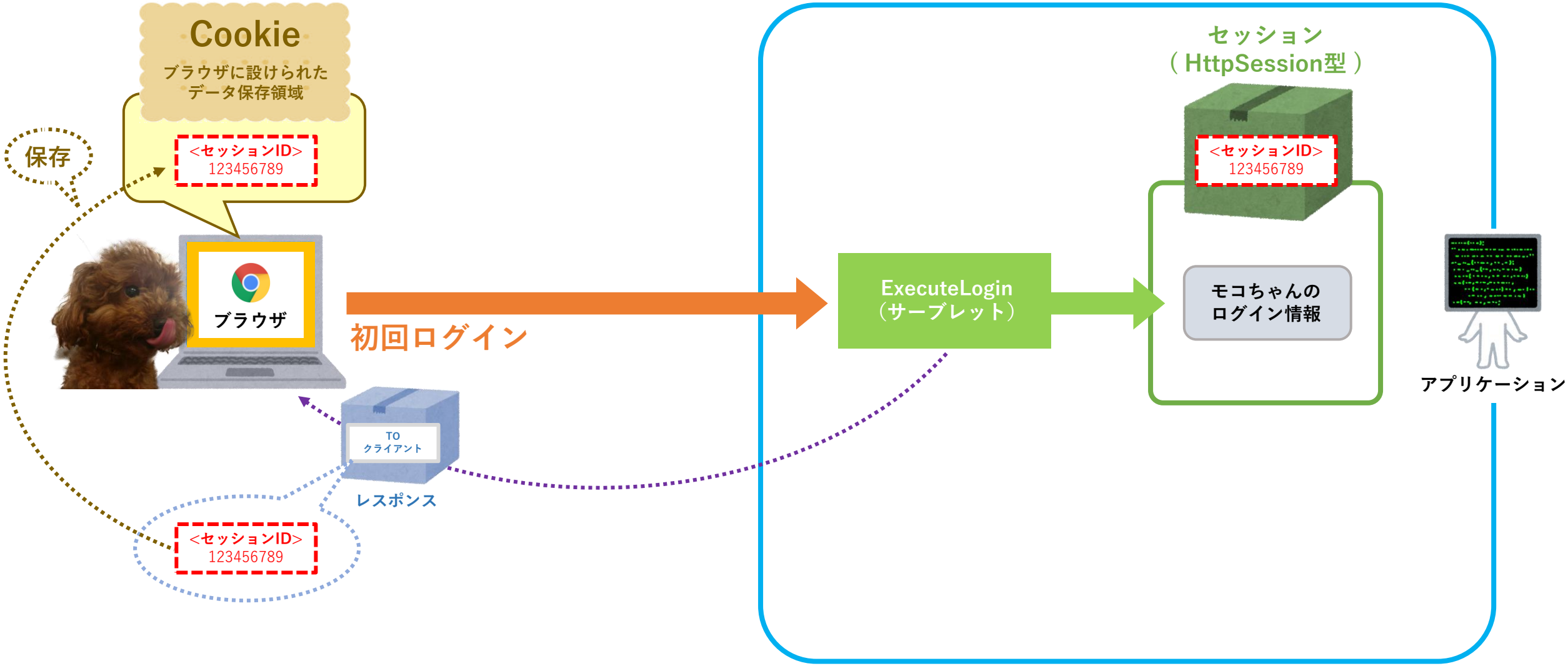
# ～ Cookie／セッションの必要性 ～



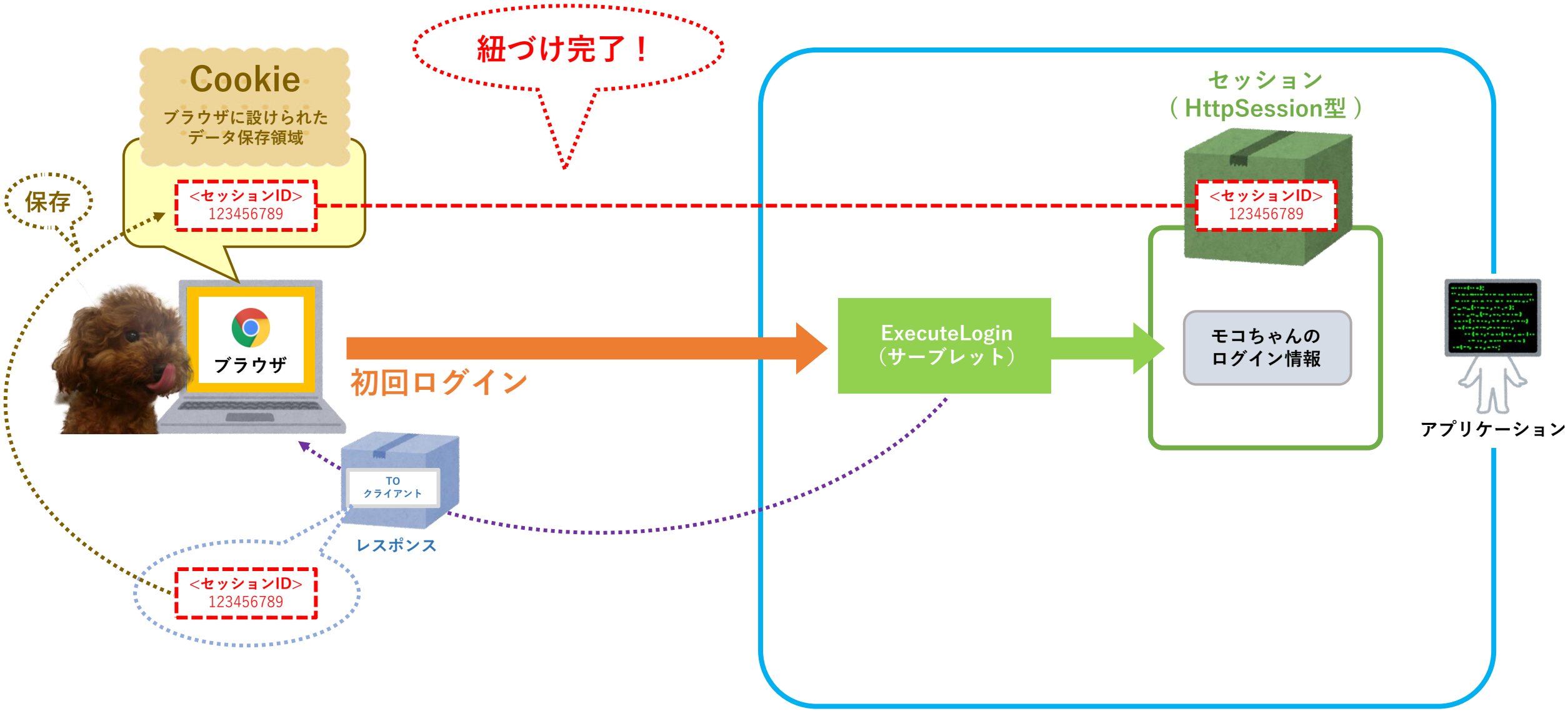
# ～ Cookie／セッションの必要性 ～



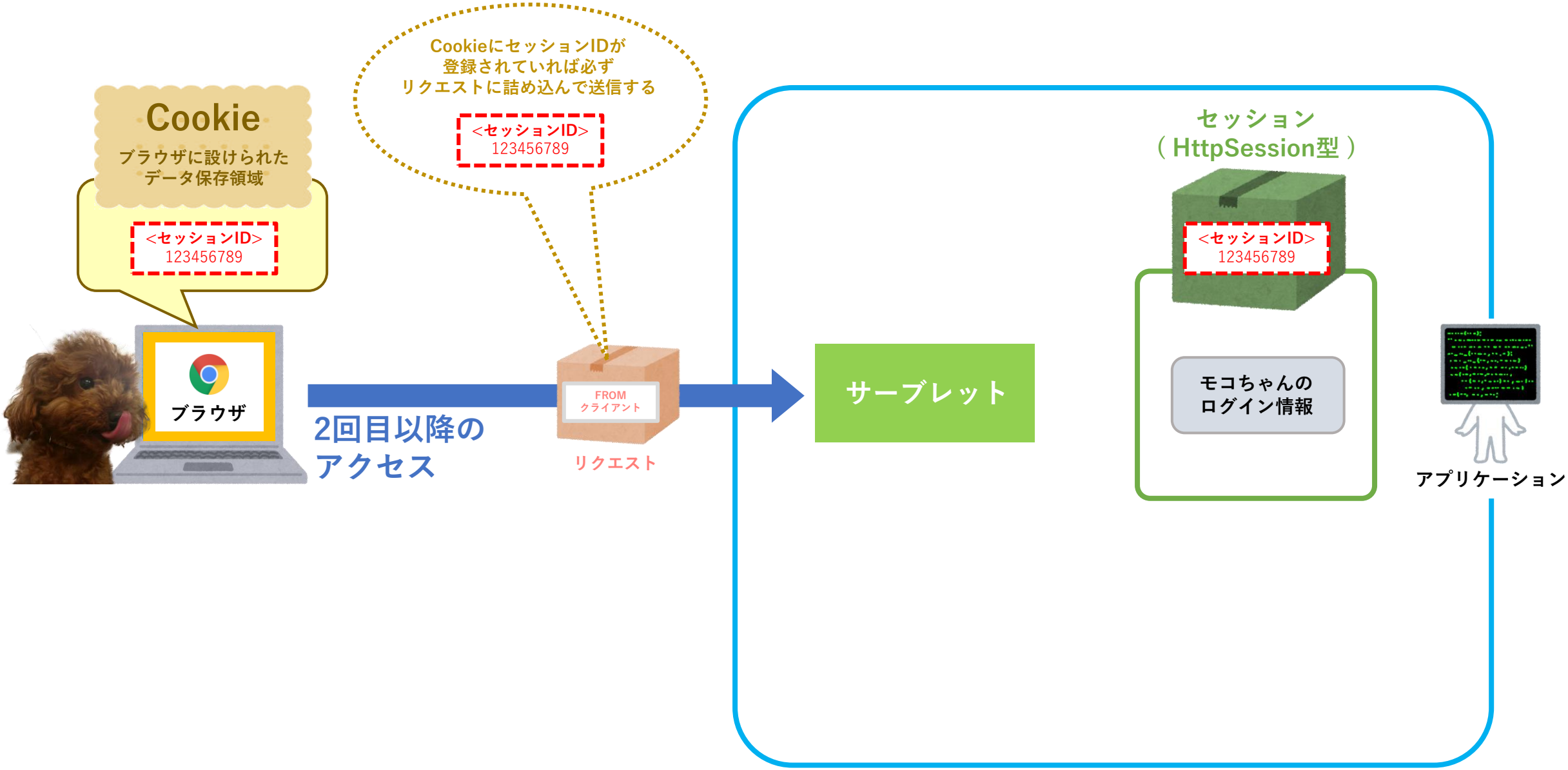
# ～ Cookie／セッションの必要性 ～



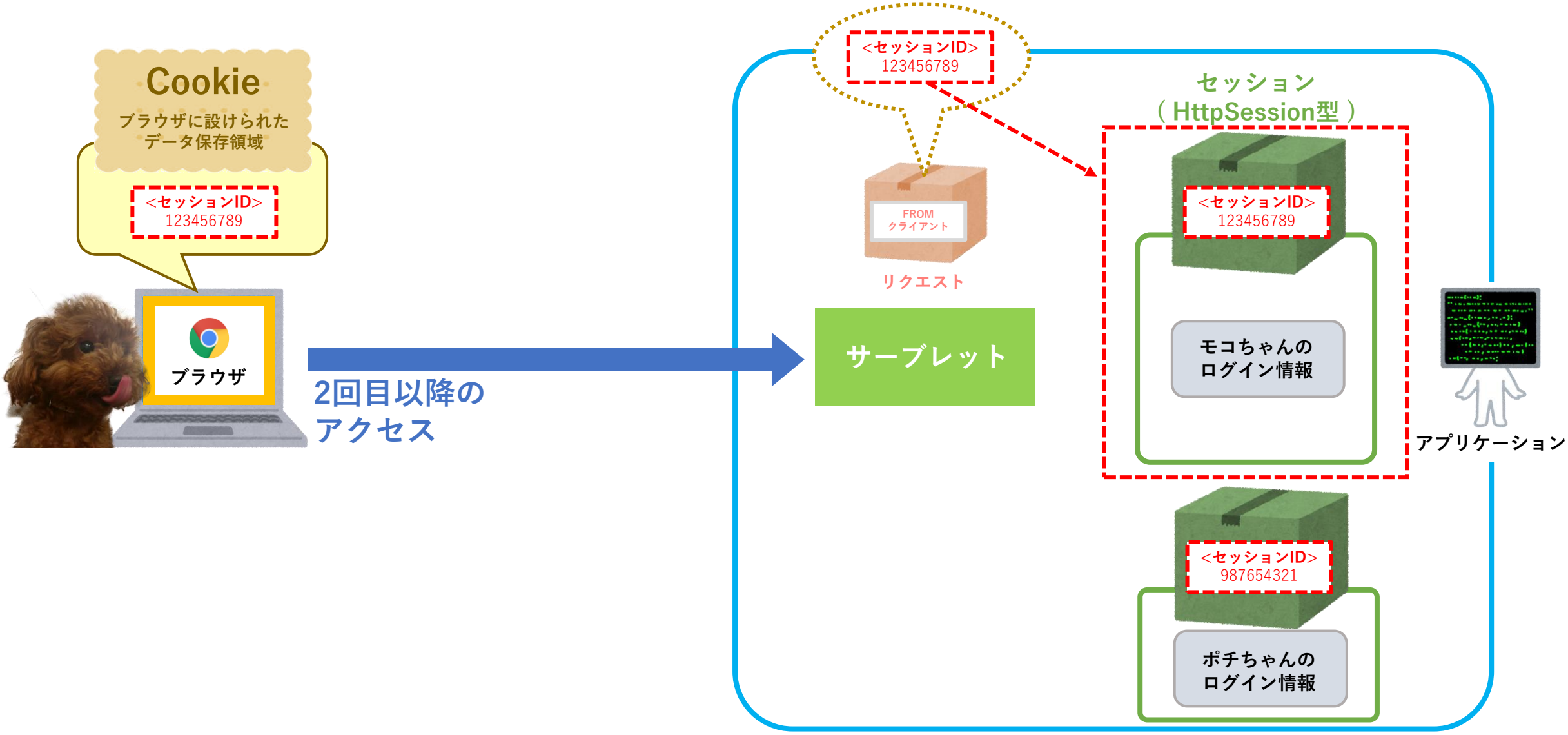
# ～ Cookie／セッションの必要性 ～



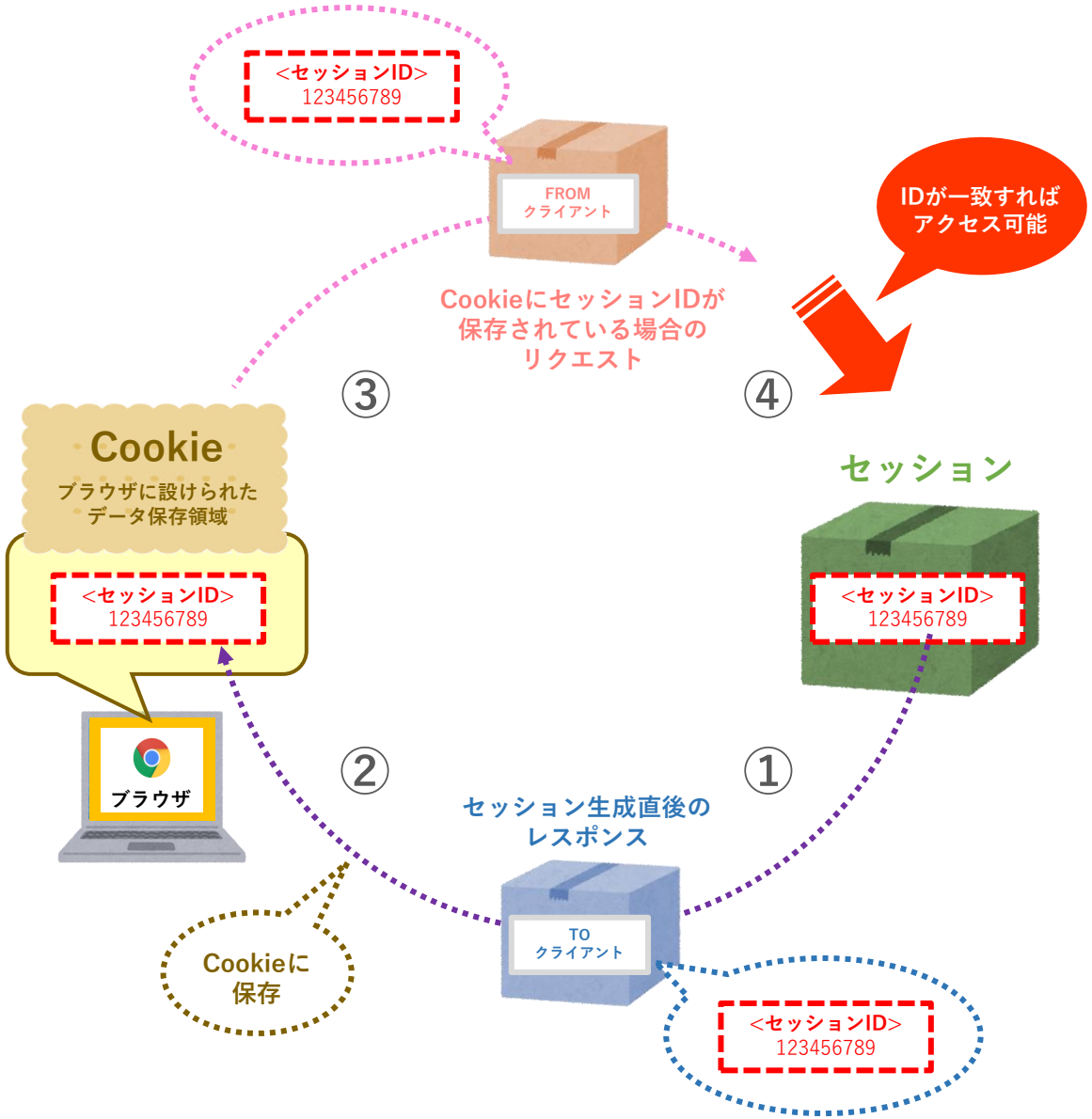
# ～ Cookie／セッションの必要性 ～



# ～ Cookie／セッションの必要性 ～



# ～ Cookie／セッションの必要性 ～




## ≪Cookie≫

□ブラウザに設けられたデータ保存領域のことを**Cookie**と言います。
ブラウザに持っておいてほしい情報を保存するために使用されます。

□セッションとブラウザの紐づけを行う際には必ずCookieの存在が必要になります。紐づけされる仕組みは以下のとおり。

①新しくセッションが生成された場合、その直後のレスポンスにセッション固有の番号（セッションID）が格納される。

②レスポンスにセッションIDが格納されている場合、ブラウザはそれをCookieへと保存する。

③CookieにセッションIDが保存されている場合、ブラウザは必ずこれをリクエストに格納して送信する。

④サーブレット内でセッションの生成「request.getSession()」を行う際、リクエストに格納されたセッションIDと合致するセッションを取得する。

□Cookieは少し知識があれば簡単に閲覧が可能な情報であるため、**悪意ある攻撃を受けやすい**場所です。
IDやパスワードといった重要な情報は保存しない、保存するのは一見何を表しているのかわからない情報に限るといったことを肝に銘じておきましょう。

# 【演習】

プロジェクト「Ex_5_05」に
ログイン/ログアウト機能を追加して
ユーザー別に動作するサイトへと作り替えましょう！

ここがしょうねんば！！

# 【演習】

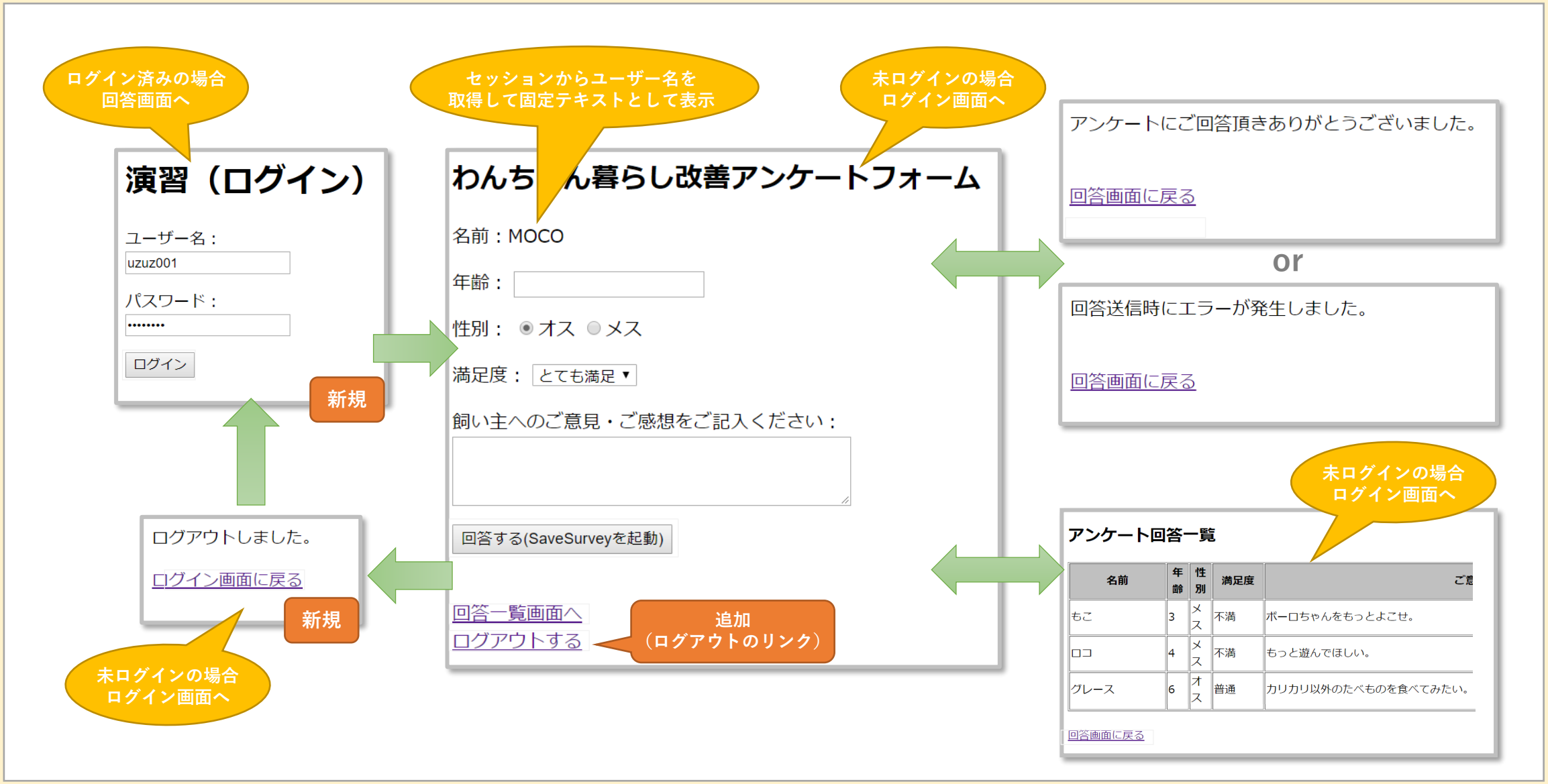
## 手順

（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_06

（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_05 」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

# 【演習】


ログイン済みの場合
回答画面へ
セッションからユーザー名を
取得して固定テキストとして表示
未ログインの場合
ログイン画面へ
演習（ログイン）
ユーザー名：
uzuz001
パスワード：
ログイン
新規
わんちゃん暮らし改善アンケートフォーム
名前：MOCO
年齢：
性別： オス メス
満足度： とても満足
飼い主へのご意見・ご感想をご記入ください：
回答する(SaveSurveyを起動)
回答一覧画面へ
ログアウトする
追加
（ログアウトのリンク）
アンケートにご回答頂きありがとうございました。
回答画面に戻る
or
回答送信時にエラーが発生しました。
回答画面に戻る
ログアウトしました。
ログイン画面に戻る
新規
未ログインの場合
ログイン画面へ
未ログインの場合
ログイン画面へ
アンケート回答一覧
名前
年齢
性別
満足度
もこ
3
メス
不満
ボーロちゃんをもっとよこせ。
ロコ
4
メス
不満
もっと遊んでほしい。
グレース
6
オス
普通
カリカリ以外のたべものを食べてみたい。
回答画面に戻る

# ～ inputタイプ『hidden』 ～