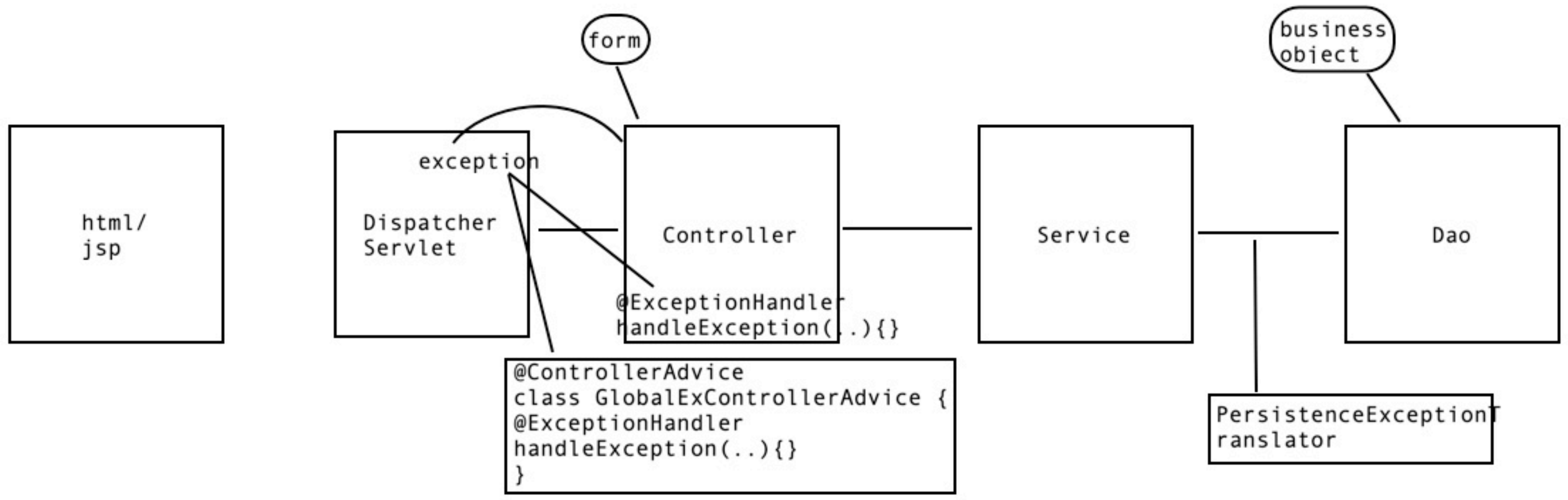

```
@Controller
@RequestMapping("/signup")
class FundsTransferController {

  @GetMapping
  public String setupForm(ModelMap model) {}

  @PostMapping
  public String transfer(@ModelAttribute("signUpForm") SignUpForm form,
BindingResult errors, ModelMap model) {}

  @ExcepionHandler
  public String handleLimitExceededException(LimitExceededException e,
ModelMap model) {
    model.addAttribute("errorMessage",
       "Funds not available in the source account");
    return "limit-exceeded-error";
  }
}
```

Per each exception we encounter across the handler methods of our controller we need to write one @ExceptionHandler method that takes care of translating the exception into friendly error page and returns to the DispatcherServlet asking to render.

FundsNotAvailableException
DormantAccountException
LimitExceededException

Its not that these exceptions creeps up in this Controller, there is a chance of hanving these exceptions reported across the handler methods of multiple controllers.

In such case we need to write the same exception handler methods in all the controllers of our application, so that again the exception handling logic is duplicated.
So to avoid duplicating this spring mvc has introduced @ControllerAdvice which is an global exception handler
@ControllerAdvice = the annotation itself makes us understand it works based on aop. The Advice class contains cross-cutting logic in handling the exceptions and this advice is applied across all the Controller classes which are annotated with @Controller in our application.

@ControllerAdvice is a short form of applying ThrowsAdvice across all the Controllers of our application

```
@ControllerAdvice
class GlobalExceptionControllerAdvice {
    @ExcepionHandler
  public String handleLimitExceededException(LimitExceededException e, ModelMap
model) {
    model.addAttribute("errorMessage",
       "Funds not available in the source account");
    return "limit-exceeded-error";
  }
}
```

Now this @ExceptionHandler would be called for any Controller class reporting LimitExceededException