

RETHINKING
WHAT YOU THINK
YOU KNOW

Paul M. Jones

joind.in/talk/7fee8

The Addison-Wesley Signature Series

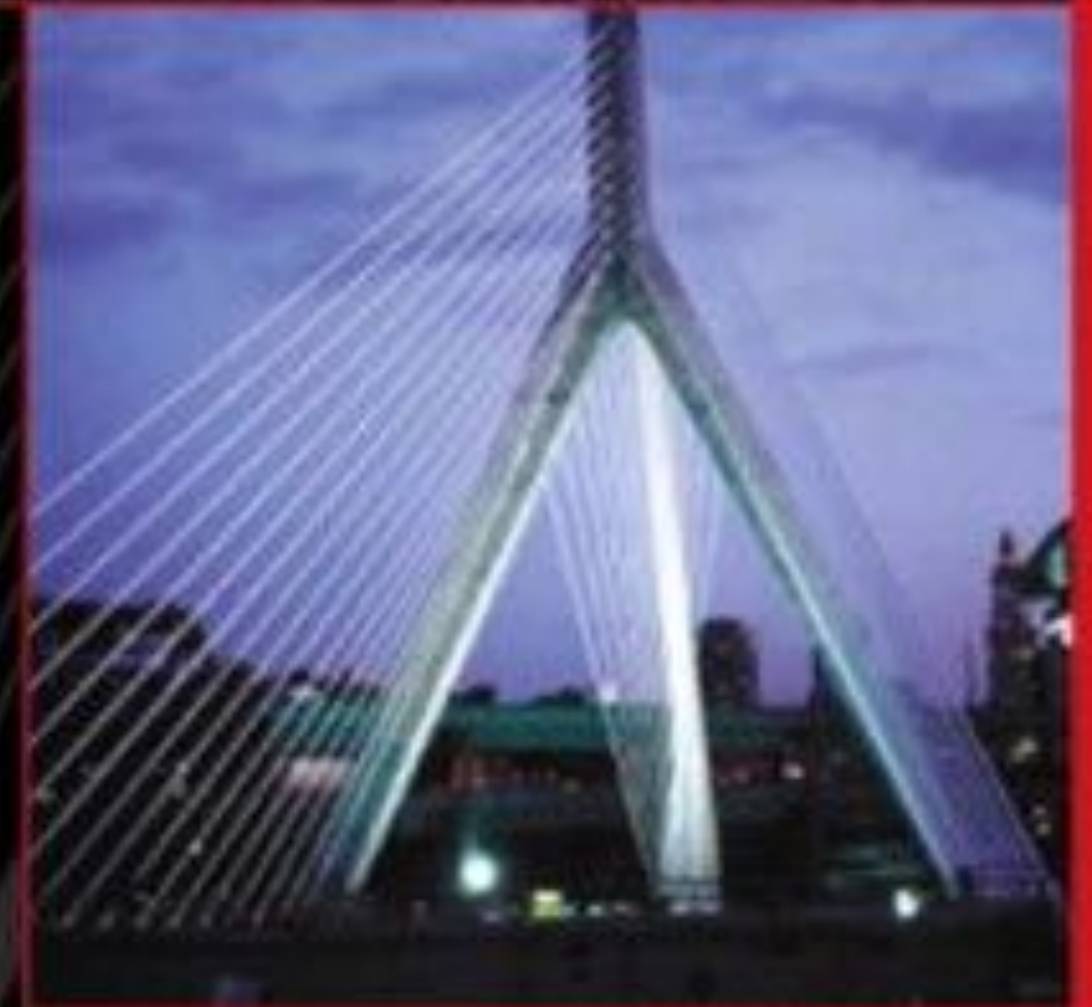


PATTERNS OF ENTERPRISE APPLICATION ARCHITECTURE

READ THIS

MARTIN FOWLER

WITH CONTRIBUTIONS BY
DAVID RICE,
MATTHEW FOEMMEL,
EDWARD HEATT,
ROBERT MEE, AND
RANDY STAFFORD



ABOUT

- BASIC in 1983, PHP since 1999
- Jr. Developer, VP Engineering
- Zend Framework 1 (DB and View)
- Founding member of PHP-FIG:
PSR-1, PSR-2, PSR-4
- Aura, Atlas, AutoRoute
- <https://leanpub.com/mlaphp>



Modernizing Legacy Applications in PHP

Paul M. Jones

OVERVIEW

- How we make decisions
- Problem: legacy code
- Server-side MVC
- Research and rethinking
- Action Domain Responder



VIA [THE ESSENTIAL DRUCKER](#), CH 17; ORIG. "THE EFFECTIVE EXECUTIVE", 1966.

HOW DO WE MAKE DECISIONS?

GET THE FACTS...

- Technical problem at work?
- Start with the facts!
- Facts will lead to the right solution.



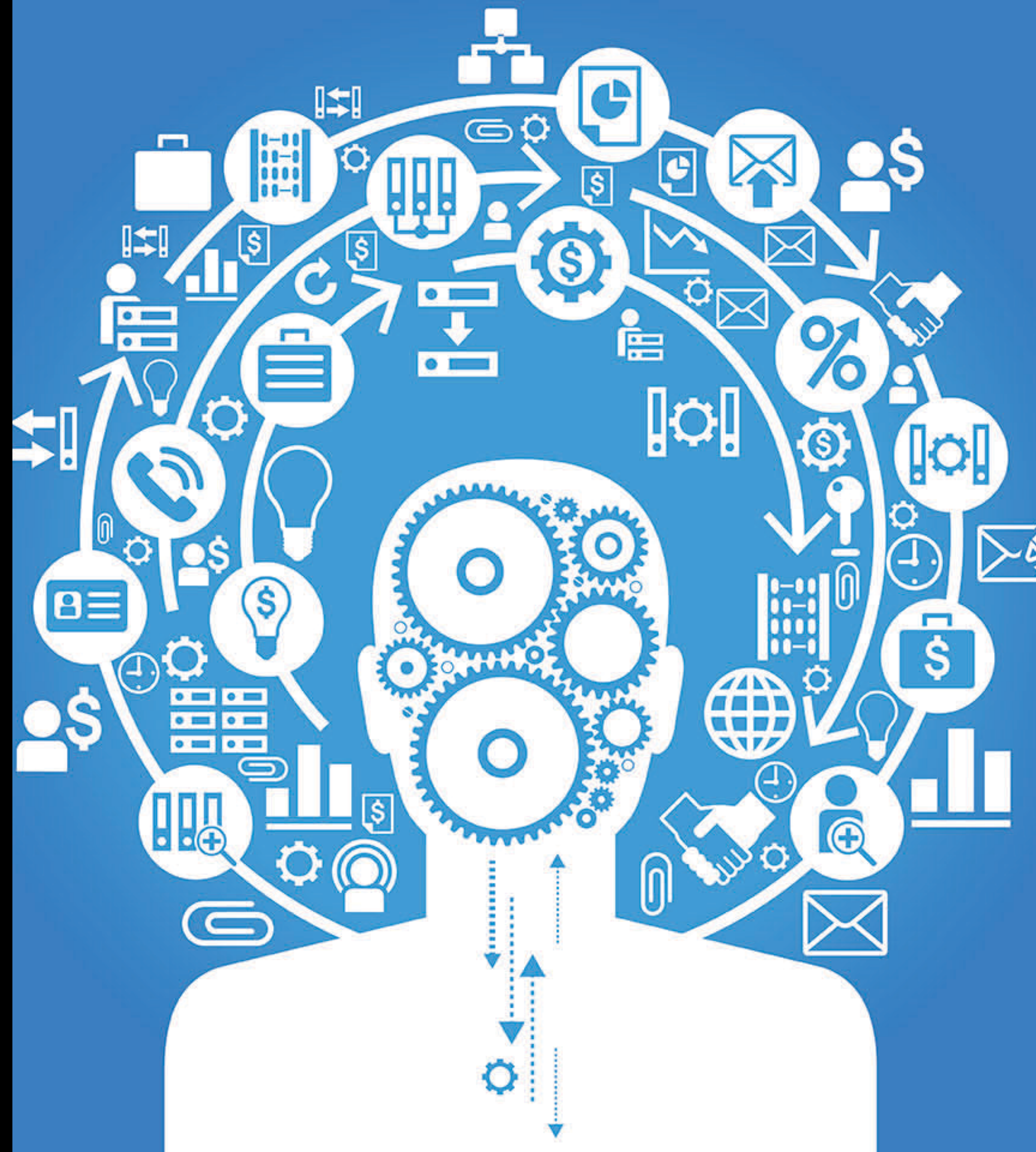
... WHICH FACTS?

- Water boils at 212° F (100° C).
- The Sahara Desert is in northern Africa.
- The Earth orbits ~93M miles (~150M km) from the Sun.



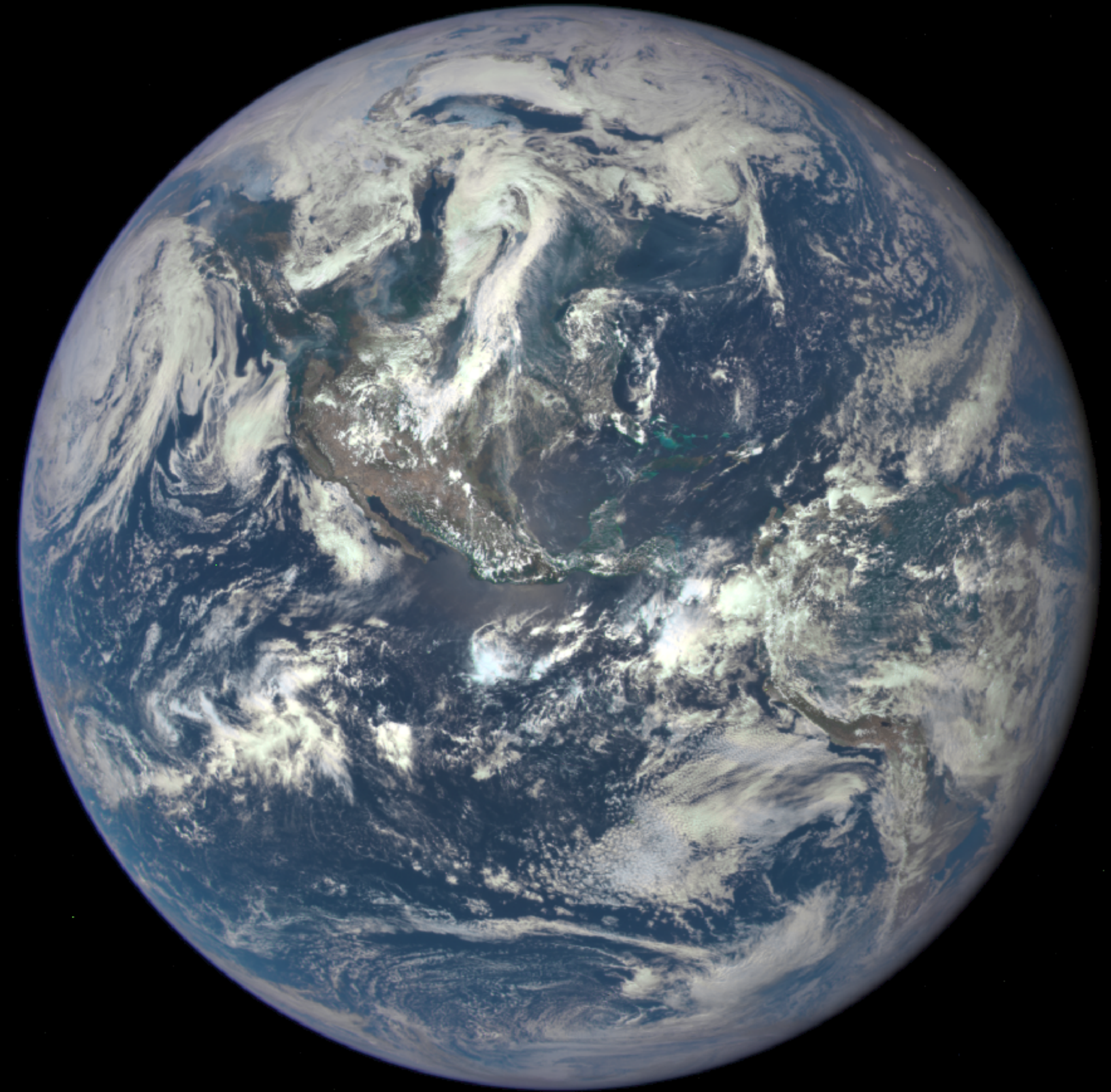
RELEVANCE

- Nobody starts with the facts.
- Everybody starts with "criteria of relevance."
- Filter against the infinity of facts.



PRE-EXISTING WORLD VIEW

- Observations and experience
- Theories and knowledge
- Opinions



CONFLICTING WORLD VIEWS

- Two sets of filters
- Different collections of facts
- Both collections are true
- Lead to very different proposed solutions



WHICH WORLD VIEW IS "RIGHT" ?

- Proposal is not based on "facts"
- It is based on "world view"
- Warning: not "everything is opinion"
- Proposal is a **hypothesis**
- Some more valid than others



TEST AND OBSERVE

- Don't argue about opinions
- Test hypotheses, per circumstances
- Observe results
- Pick the most-valid proposal
- Replicability is even better



EFFECT ON WORLD VIEW

- New observation is new knowledge
- Reinforce pre-existing world view
- Rationalize disconfirmation
- Accept new criteria of relevance
- Revise or replace world view



“There is only one science.
It is called physics.
Everything else is stamp collecting.”

–ERNEST RUTHERFORD

NOT SCIENCE

- Physics provides predictive capacity
- Collect observations
- Patterns of organization
- World views as ways of organizing knowledge
- Observably true



WHAT DECISION ARE WE TRYING TO MAKE?



HOW TO ORGANIZE SERVER-SIDE CODE

- Nuke it from orbit?
- No: reorganize and refactor
- Comprehensible
- Maintainable
- Testable



“When concerns are well-separated, individual sections can be reused, as well as developed and updated independently.

Of special value is the ability to later improve or modify one section of code without having to know the details of other sections, and without having to make corresponding changes to those sections.”

-SEPARATION OF CONCERNS (INFOGALACTIC)

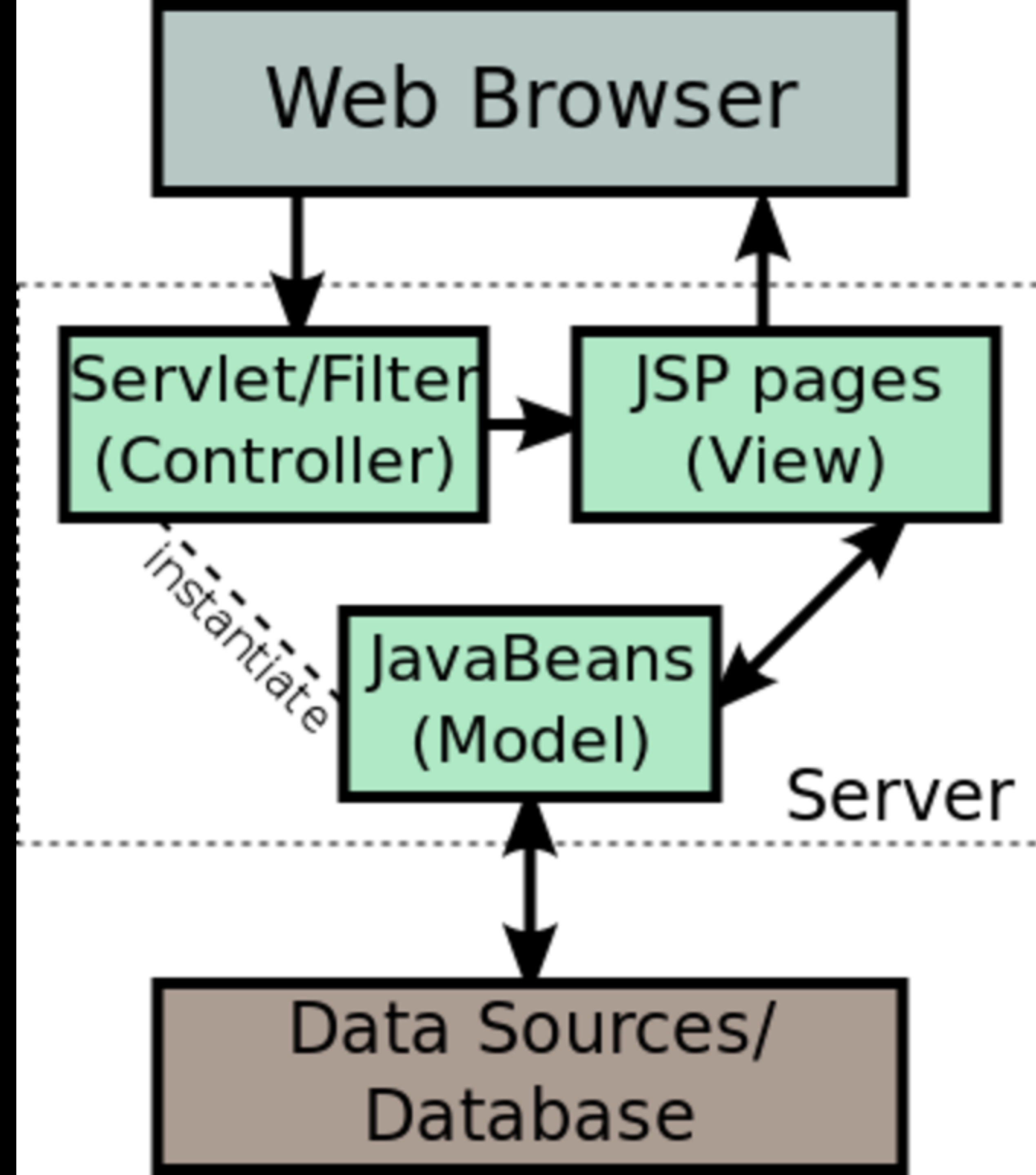
MODEL VIEW CONTROLLER

- Invented in 1979 for GUI desktop apps
- Model: data source and business logic
- View: what you see on the screen
- Controller: receives user input, coordinates model and view



SERVER-SIDE MVC

- Applied to server-side apps by Sun in 1999 ("Model 2")
- Controller handles HTTP request
- Model performs business logic
- View generates HTML
- Organizing principle / world view



MODERNIZATION PROCESS

- Extract data source logic
- Extract domain logic
- Extract presentation logic



Modernizing Legacy Applications in PHP

Paul M. Jones

WHERE DOES HEADER() GO?

- Model? Not data or business logic.
- View? Can't see it on the screen.
- Controller! Last place left in our application architecture.

```
19     $id = $article_transactions->submitNewA
20     $result = 'created';
21 }
22
23 $failure = $article_transactions->getFailure
24 $input = $article_transactions->getInput();
25
26 if (! $failure && $result == 'created') {
27     http_response_code(201);
28     header('Location', "/articles/{$id}/edit");
29     exit();
30 }
31
32 if ($failure) {
33     http_response_code(422);
34 }
35
36 $template = new Template('/path/to/app/view');
37 $template->setVars([
38     'id' => $id,
39     'result' => $result,
40     'failure' => $article_transactions->getFailure(),
41     'input' => $article_transactions->getInput(),
42     'action' => $_SERVER['PHP_SELF'],
43 ]);
44 return $template->render('article.html.php');
45
```

COGNITIVE DISSONANCE

- More comprehensible
- More maintainable
- Model is testable
- View is testable
- Controller ... not testable?



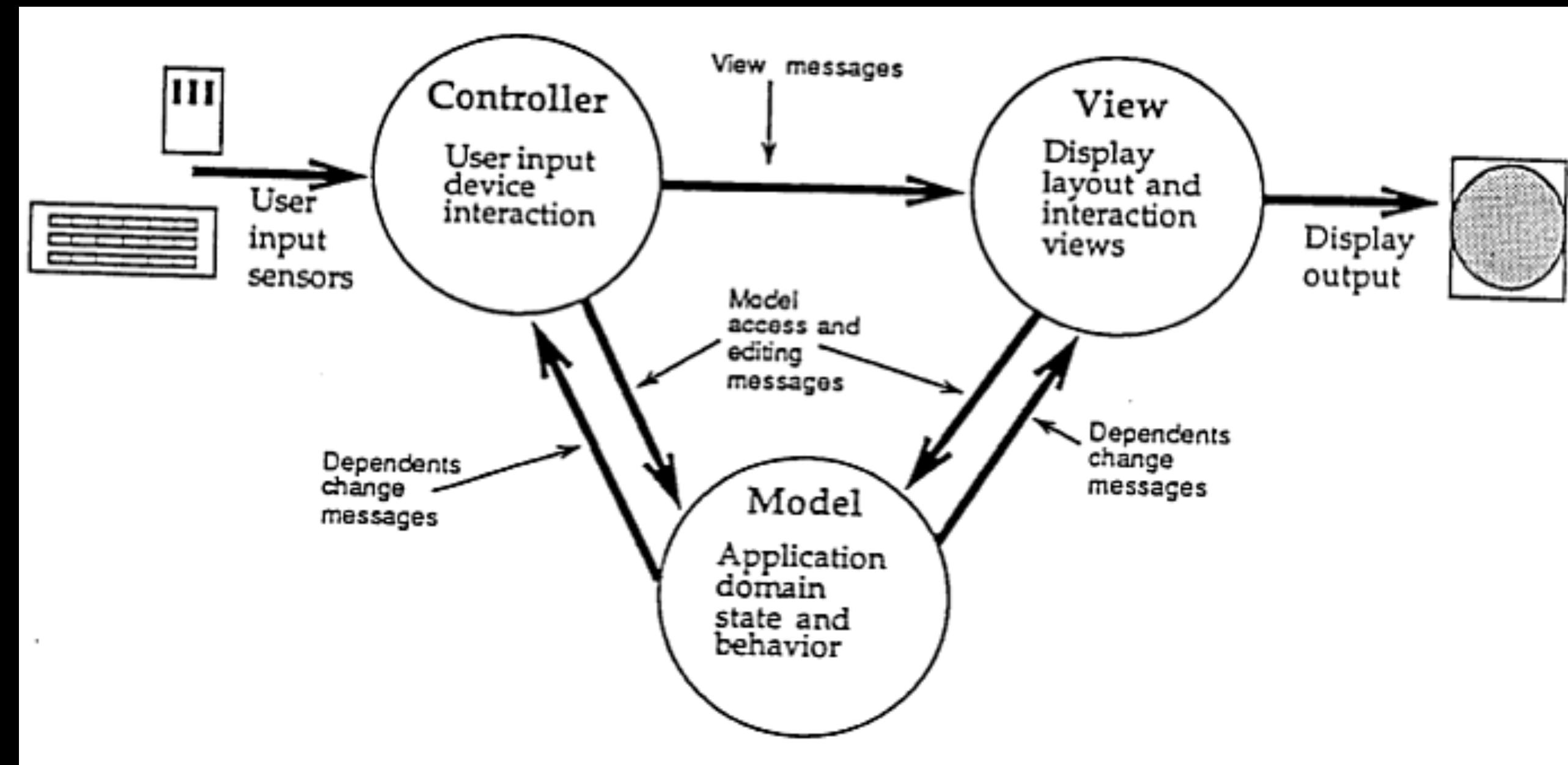
USE A RESPONSE OBJECT?

- Solves the testing problem by buffering header() calls
- But where do the \$response->header() calls go?
- Why is this the job of the Controller?

```
20     $result = 'created';
21 }
22
23 $failure = $article_transactions->getFailure();
24 $input = $article_transactions->getInput();
25
26 $response = new Response();
27
28 if (! $failure && $result == 'created') {
29     $response->setCode(201);
30     $response->setHeader('Location', "/articles/{$id}");
31     return $response;
32 }
33
34 if ($failure) {
35     $response->setCode(422);
36 }
37
38 $template = new Template('/path/to/app/views');
39 $template->setVars([
40     'id' => $id,
41     'result' => $result,
42     'failure' => $article_transactions->getFailure(),
43     'input' => $article_transactions->getInput(),
44     'action' => $_SERVER['PHP_SELF'],
45 ]);
46
47 $html = $template->render('article.html.php');
48 $response->setBody($html);
49
50 return $response;
51
```

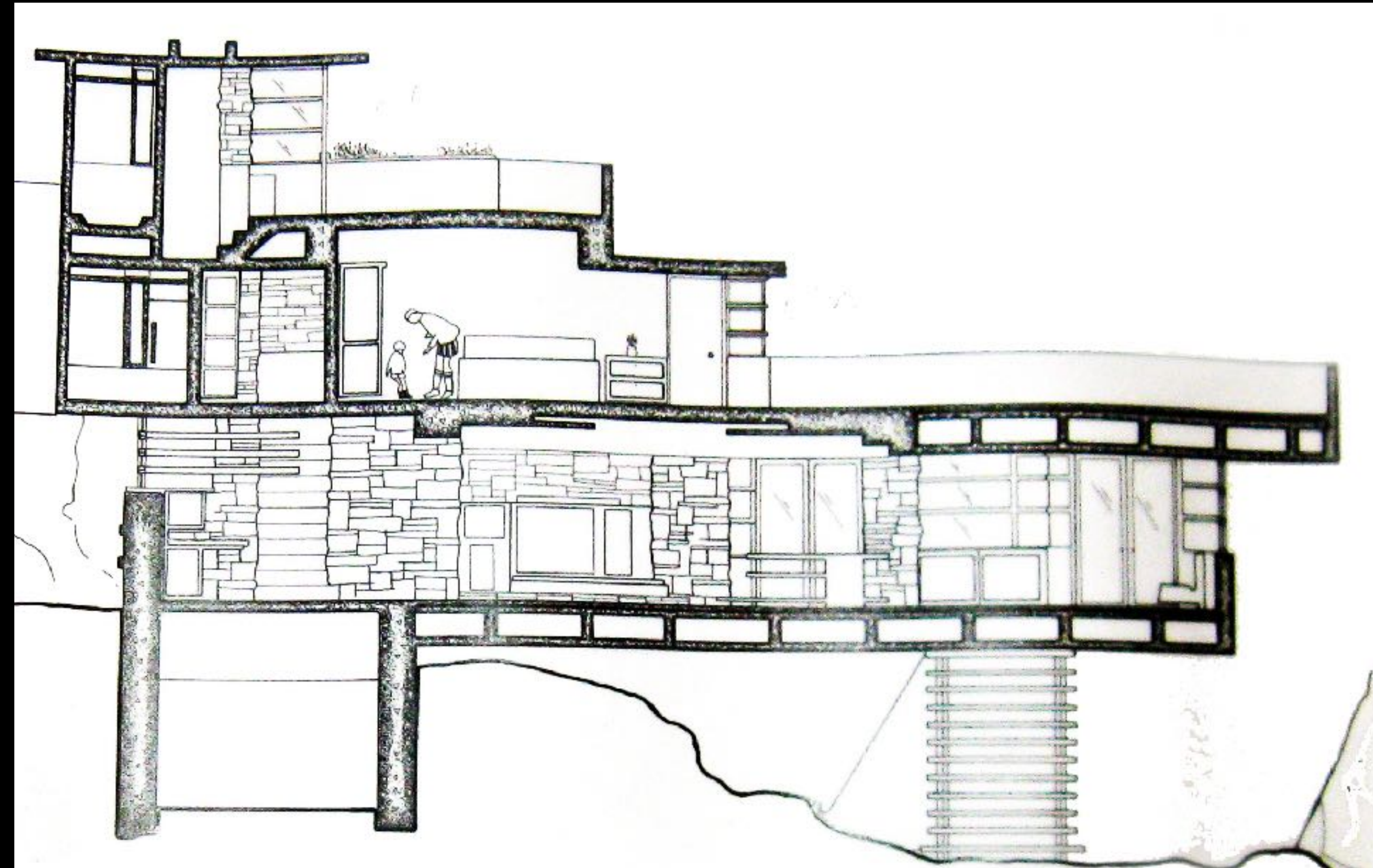

ENVIRONMENTS AND RELATIONSHIPS

- Client-side in-memory GUI
 - One MVC triad per screen element
 - Notifier/observer system
 - Continuous invocation event loop
- Server-side over-the-network shared-nothing request/response



NOT AN APPLICATION ARCHITECTURE

- MVC is a **user interface pattern**
- Fowler: "Web Presentation"
- Our concept of MVC is a world view
- Outcomes are unsatisfactory
- New knowledge: new world view



RETHINKING OUR OPINIONS

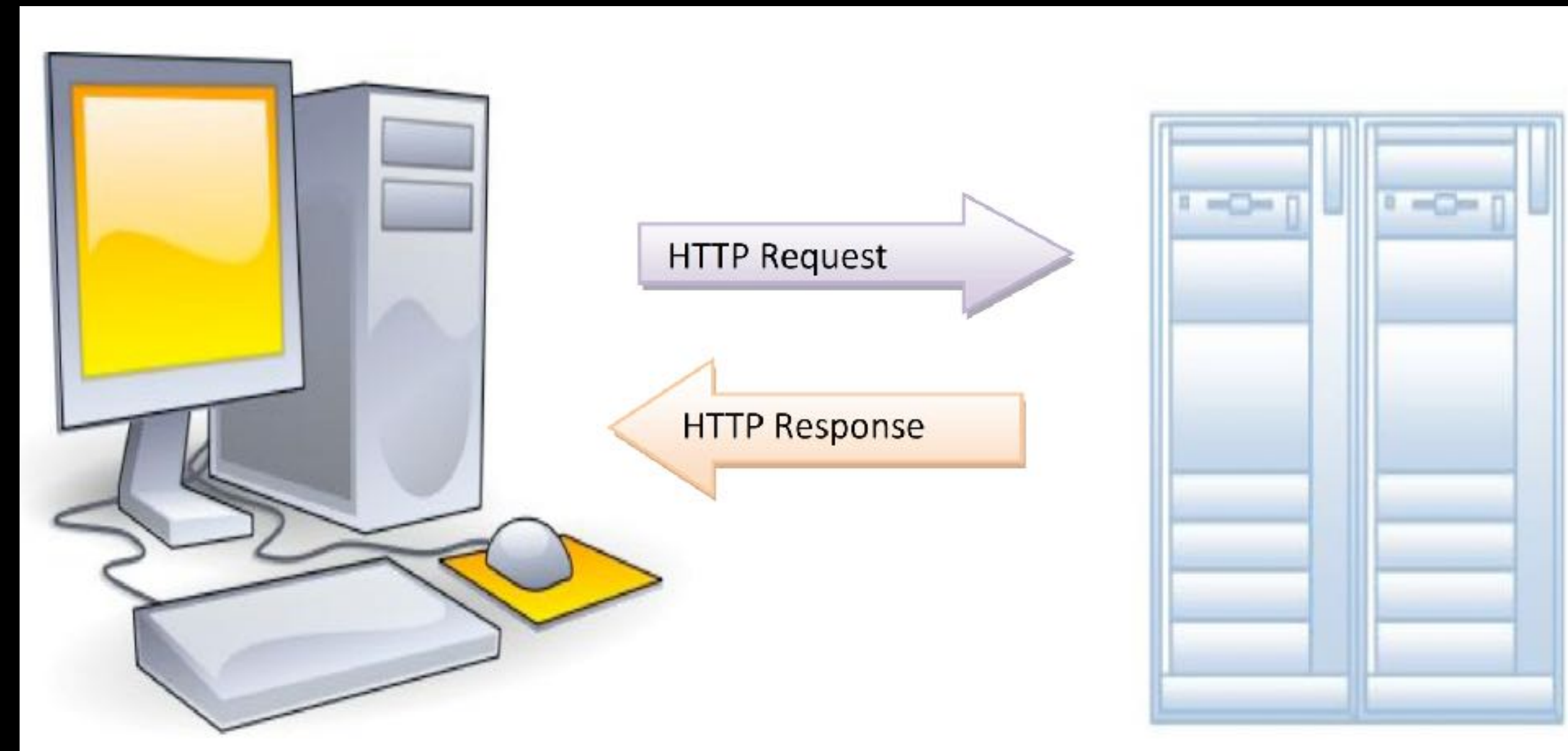
WHAT IS THE VIEW?

- Old: "The part the user sees."
- New: "The output from the application."
- Subtle distinction
- Fits previous facts
- Expands relevance filter



WHAT IS THE SERVER OUTPUT?

- Input is HTTP Request
- Returns HTTP Response
- This is our user interface
- The Template is not the View
- The Response is the View
- Both body and headers



RESPONDER

- View: "What the user sees" -> "What the server presents"
- Presentation layer is the response-building subsystem
- header() etc. go in the Responder
- "Unseen" bodies: JSON, XML, etc.
- Responder builds the Response

MORE TESTABLE

- Responder isolated from Model and Controller
- Controller feeds data to Responder
- Responder sets headers, uses template system, etc.
- Returns a Response object
- More satisfactory outcome

“Ensure that any code that manipulates presentation only manipulates presentation, pushing all domain and data source logic into clearly separated areas of the program.

... GUI widgets and structures in a rich client application,
HTTP headers and HTML in a web application,
or command line arguments and print statements
in a command line application.”

-SEPARATED PRESENTATION (FOWLER, 2006)

REPLACING SERVER-SIDE MVC

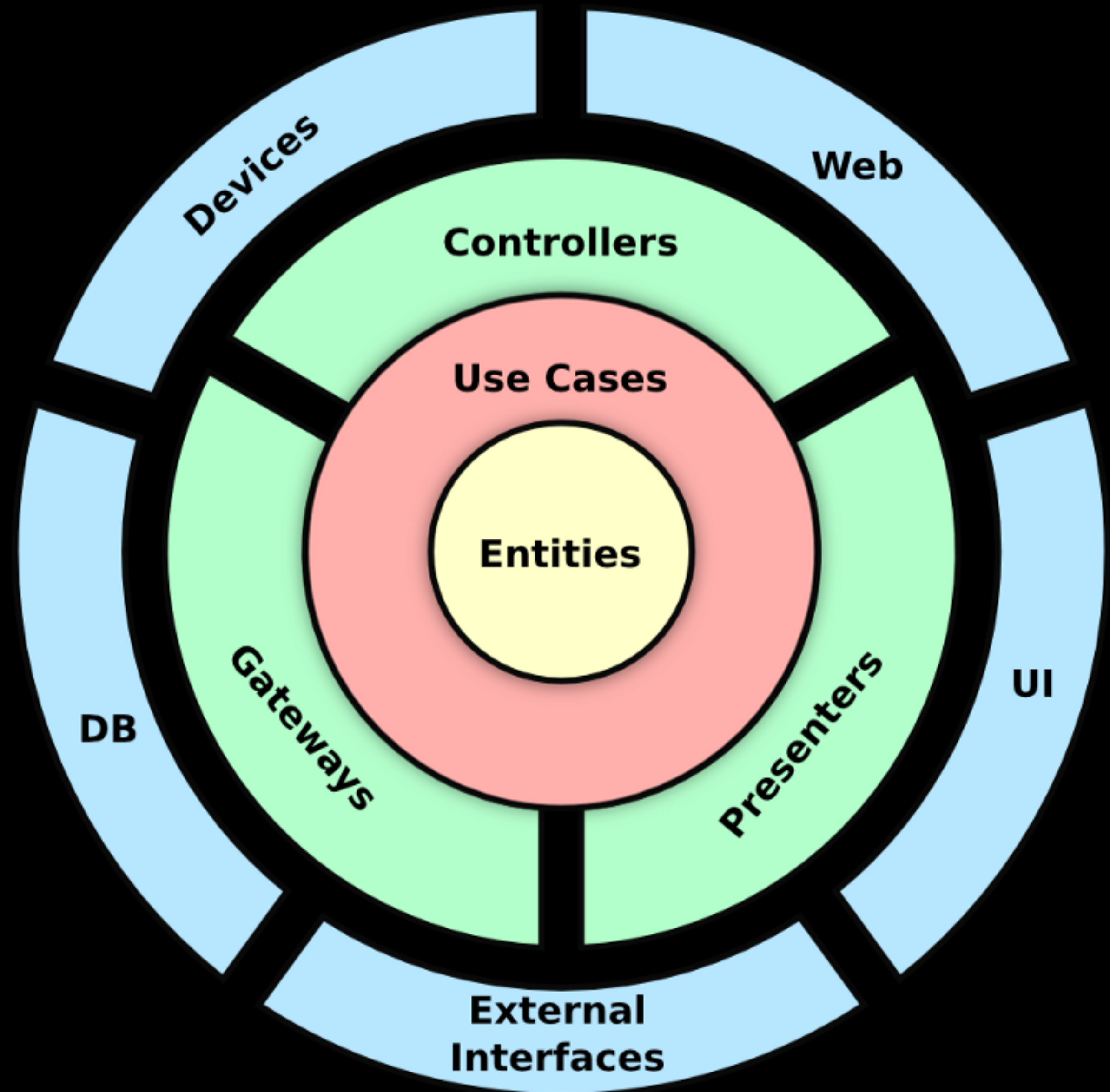
CONSEQUENCES OF WORLD VIEW

- Responders are yet another thing to inject: one per Controller action
- Leads to single-action Controllers
- Leads to shorter constructors
- Leads to other domain logic patterns



DOMAIN LOGIC

- Transaction Script
- Service Layer
- Application Service
- Use Case



ACTION DOMAIN RESPONDER

- **Action** is the logic to connect the Domain and Responder. It invokes the Domain with inputs collected from the HTTP Request, then invokes the Responder with the data it needs to build an HTTP Response.
- **Domain** is an entry point to the domain logic forming the core of the application, modifying state and persistence as needed.
- **Responder** is the presentation logic to build an HTTP Response from the data it receives from the Action.

CONCLUSION

FROM ONE WORLD VIEW TO ANOTHER

- How we make decisions: facts filtered through theory for hypotheses
- Code organization according to MVC principles
- From "View" as Template + Response, to Responder
- Replace MVC world view, with Action Domain Responder

THANKS!

- paul-m-jones.com/adr
- leanpub.com/mlaphp
- joind.in/talk/7fee8

