

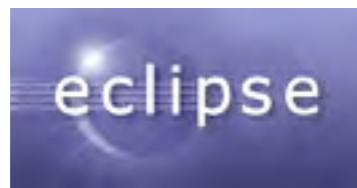
Aspect Orientated Programming in Ruby

Matthew Ford
matt@new-bamboo.co.uk





Matthew Ford



- Eclipse project AJDT @ IBM
(AspectJ Development Tools)
- New Bamboo
<http://new-bamboo.co.uk>



Outline

- Introduction to Aspect Orientated Programming (AOP)
- Why use AOP?
- AOP in Ruby
- Demo
- Q & A

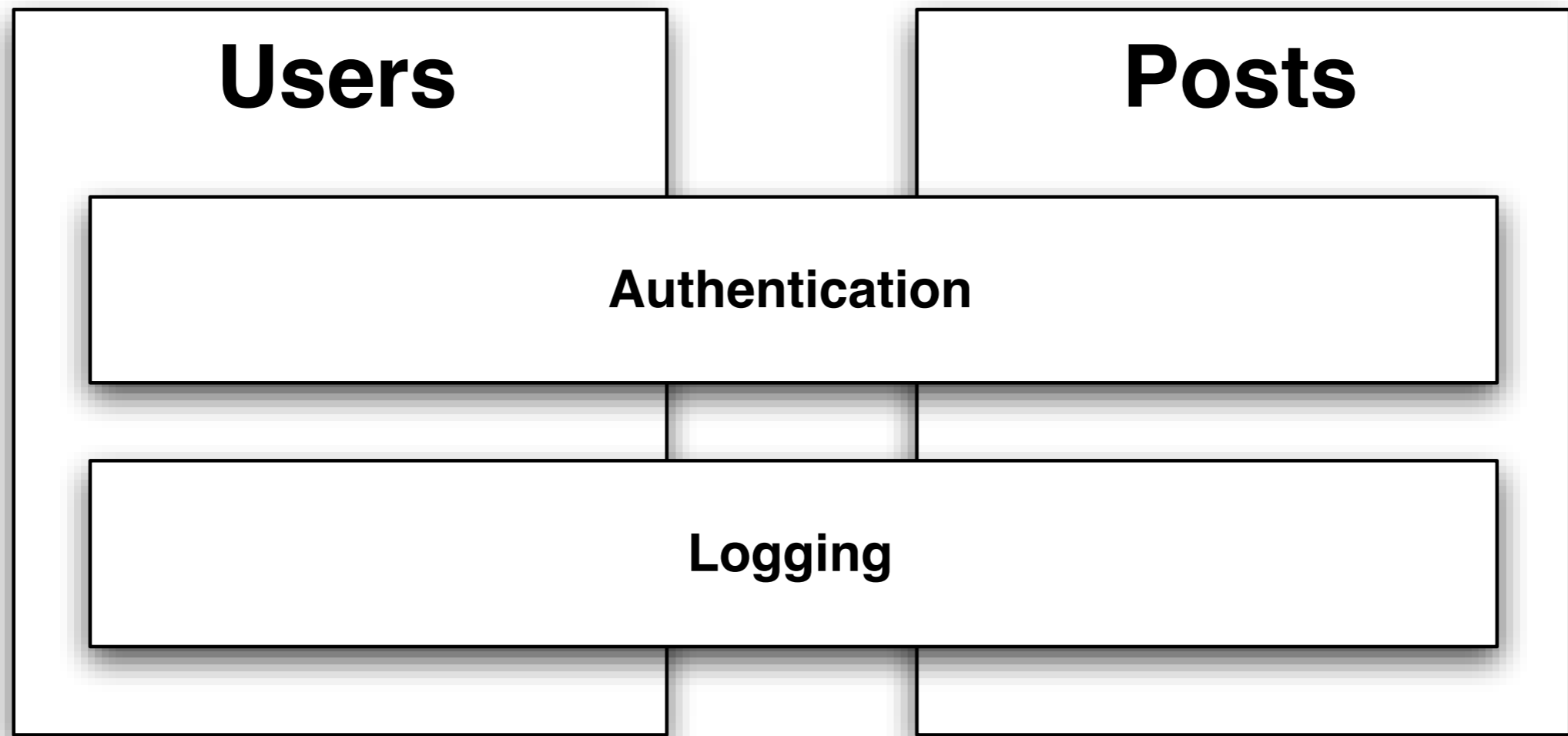
Intro to AOP

- AOP is a new way of thinking about software design.

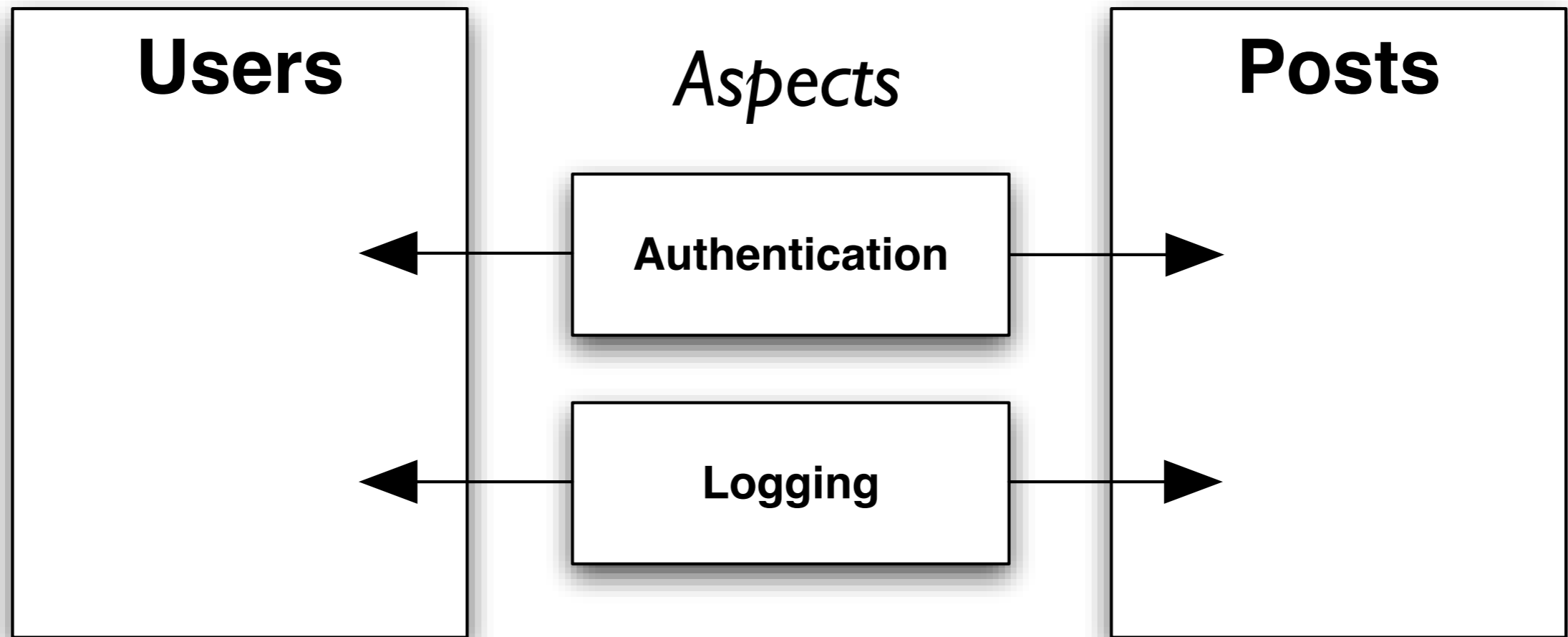
Software Design

- What is the ultimate goal when designing software?
- Aim to achieve a one-to-one mapping of design concepts to implementation code

Your app now



With AOP



What is AOP?

- A technique for modularising cross cutting concerns

Cross Cutting Concerns

- Profiling
- Persistence
- Transactions
- Access Control Lists
- Exception Handling

Scattered

- Code where a design concept is spread over a number of models
- one-to-n
- DRY?

Tangled

- Code which deals with multiple design concepts
- n-to-one mapping
- Hard to maintain?

The Real World

- Some requirements don't always map nicely in OOP
- n-to-n
- scattered and tangled



Spaghetti Code

AOP to the rescue

- Aspects
- AOP is essentially a DSL built on top of the language



Modular

Wait! Ruby has Metaprogramming

So why use AOP in Ruby?

- AOP provides another level of abstraction
- Clear syntax for describing points in your application where to inject code
- Modularises behaviour

AOP in Ruby

- Aquarium is a Ruby AOP framework
- DSL for describing cross cutting behaviour with aspects

Terminology

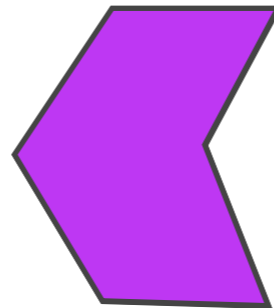
- Aspect:
 - Join Point - point in the program
 - Pointcut - matches a number of join points
 - Advice - behaviour evoked at pointcut

Join point

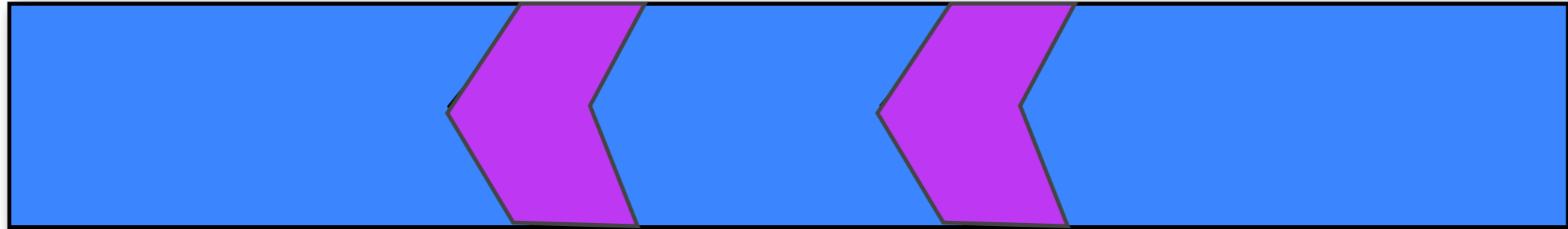
Join point



Pointcut



Aspect



Result of matched aspect

Demo Time!



Ruby Advising Java (and visa versa)

- Latest release of Aquarium can run on JRuby
- Ruby aspects can advise Java
- Java aspects can advise Ruby code

Here Be Dragons



 new bamboo

I'm going to use aspects
for everything!



Don't

HUMANDESCENT

Summary

- Aspects are cool
- Abstract and modularise cross cutting concerns
- Know when to use them!

Links

- Demo App Merb-meet-AOP:
<https://github.com/deimos1986/merb-meet-aop/>
- Aquarium <http://aquarium.rubyforge.org/>
- New Bamboo blog <http://blog.new-bamboo.co.uk>
- Dean Wampler (creator of Aquarium)
<http://blog.aspectprogramming.com/>
- The AspectJ Programming Guide <http://www.eclipse.org/aspectj/doc/released/progguide/index.html>

Attribution

- Spaghetti pic - <http://www.flickr.com/photos/cstein96/2344488576/in/set-72157604157601577/>
- Dragon pic - Flight of the Conchords- Albi
- Squicken pic - humandescent.com

Q & A

Thanks!

