Perl Design Patterns

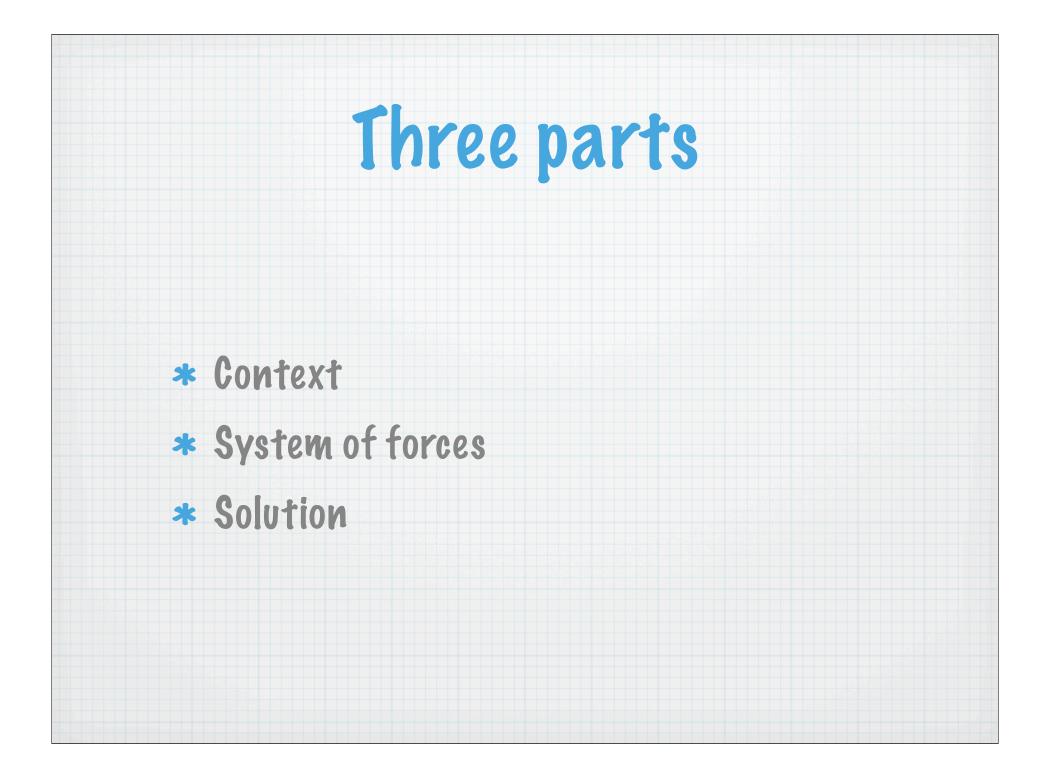
brian d foy Stonehenge Consulting Services February 24, 2006 Grand Rapids Perl Mongers

Sponsored by PriorityHealth



Patterns are a ...

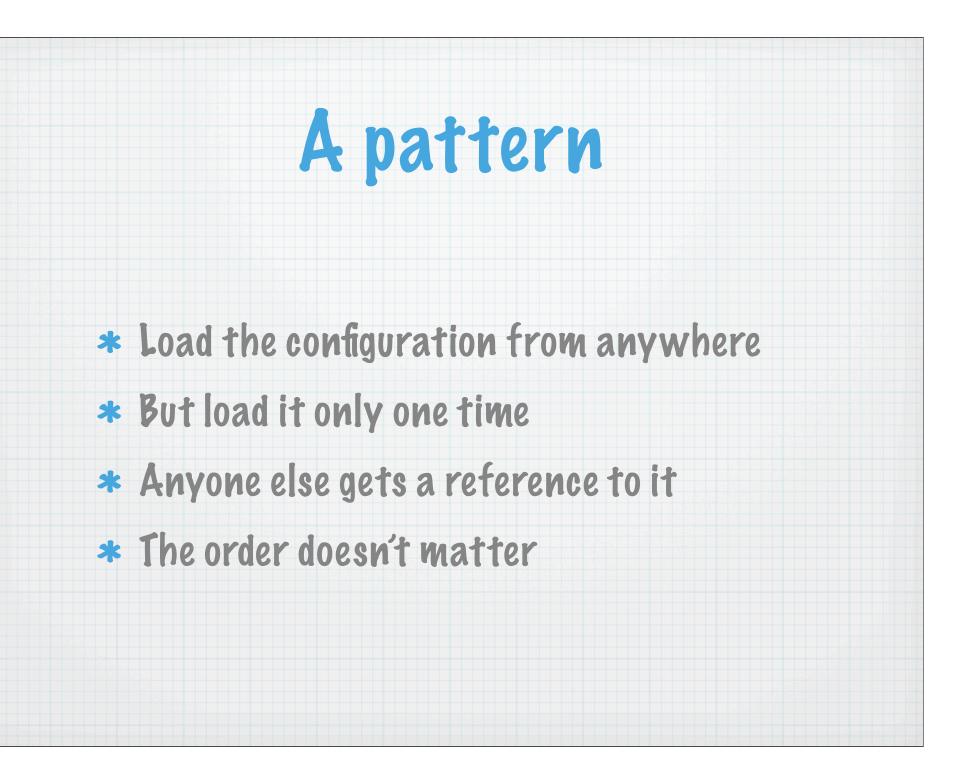
- * abstraction for things we need to do
- * solutions for common problems
- * name for a design element
- * common language for design discussions
- * way to apply a common design to code
- * relationships







- * Our application needs config information
- * But we have several modules
- * Who loads the data?
- * How do the other modules get it?







An implementation

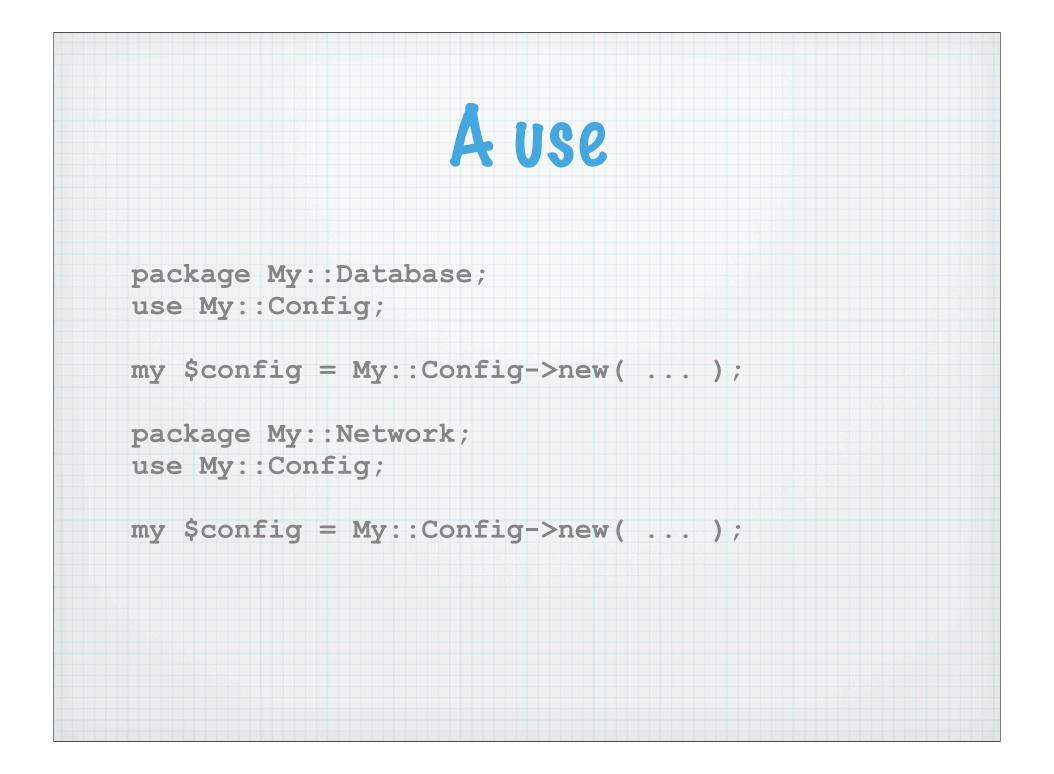
```
package My::Config;
```

```
my $singleton = undef;
```

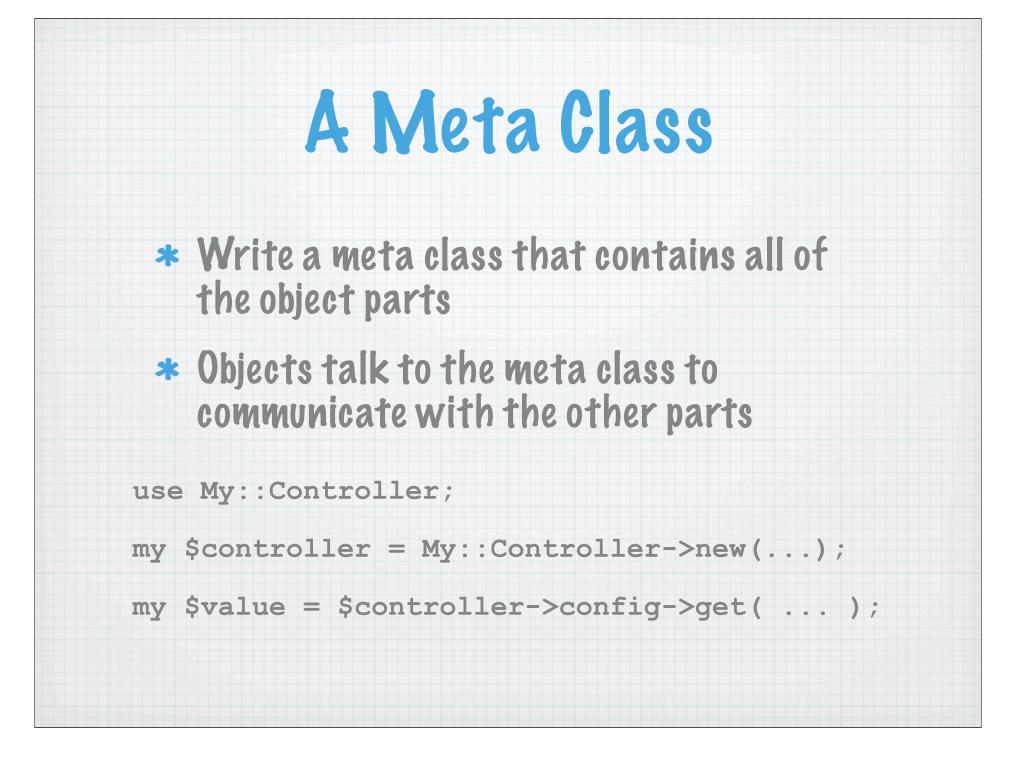
```
sub new {
  my $class = shift;
```

return \$singleton if defined \$singleton;

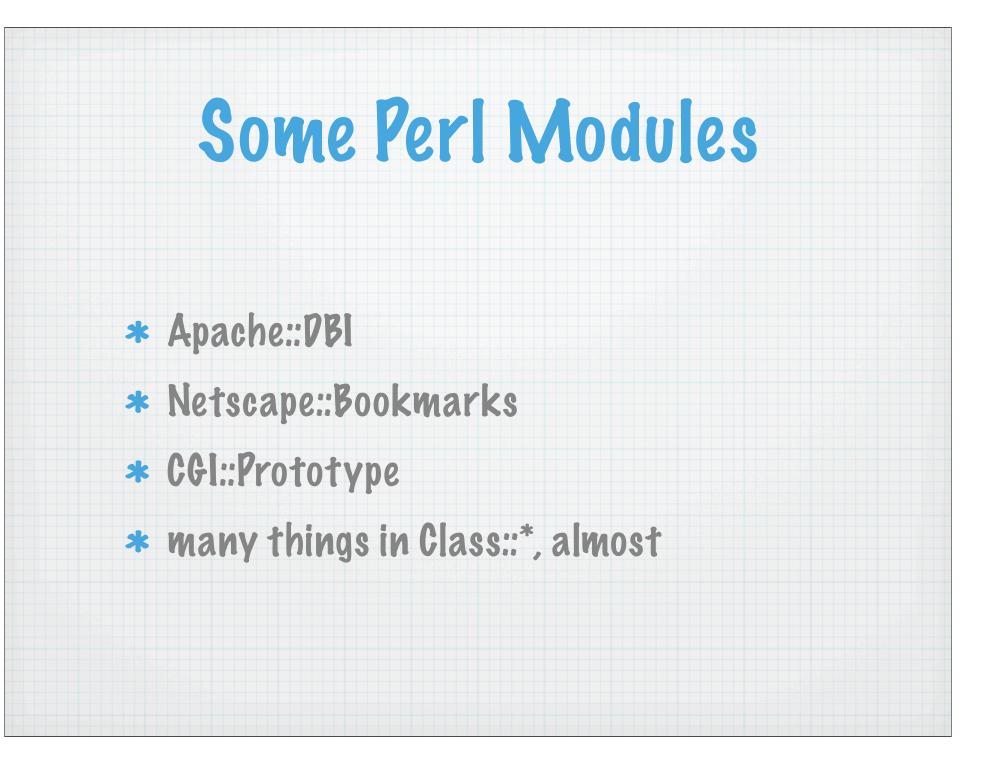
```
$singleton = bless {}, $class;
```

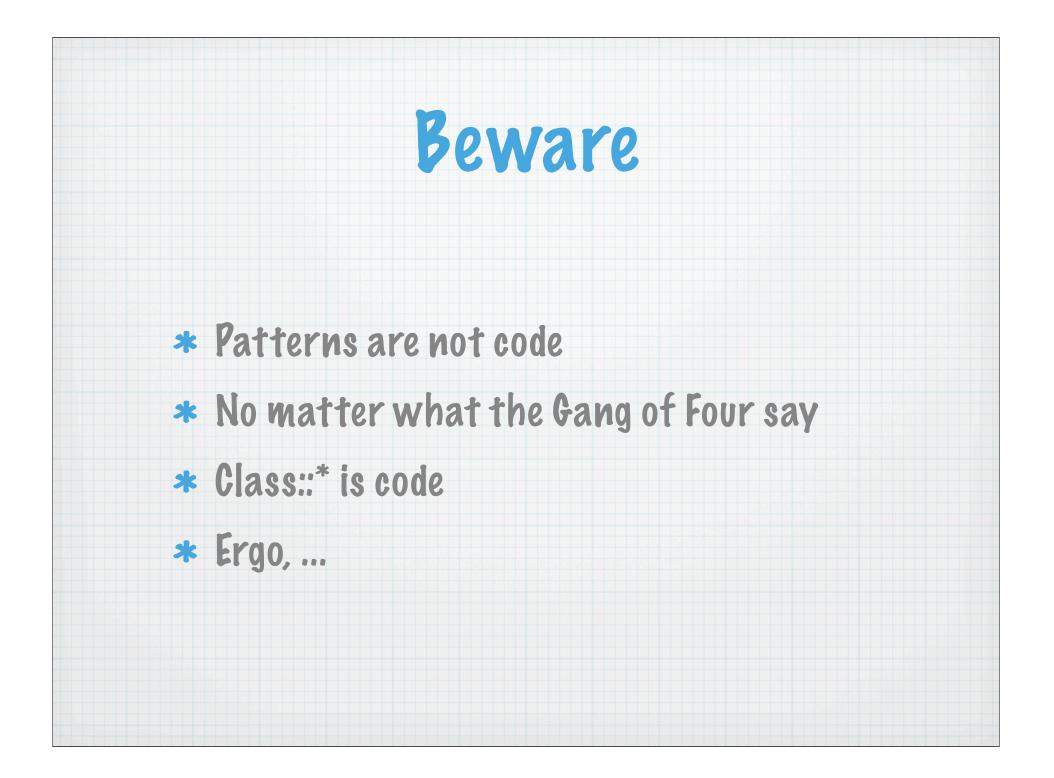






```
Delegates
package My::Controller;
sub new {
   my $class = shift;
   my $self = bless {}, $class;
   # weaken some of these circular refs
   @$self{ qw( config database network ) } = (
      My::Config->new( controller => $self ),
      DBI->new( ... ),
      My::Socket->new( controller => $self ),
      );
   $self;
sub config { $ [0] { config} }
```







Further Reading

- * The Perl Review (lots of articles (by me))
- "Design Patterns Aren't" by Mark Jason
 Dominus
 - http://perl.plover.com/yak/design/
- Pesign Patterns Erich Gamma, et al. (Gang of Four)
- * Perl Design Patterns Wiki
 - http://perIdesignpatterns.com/