

# Ruby Metaclasses

Singleton, Eigen and Ghost Classes



How to do your head in.



# Singleton Methods

- ✦ Defined on an instance of a class
- ✦ Held in a virtual class



# Singleton Methods

```
foo = Foo.new
```

```
def foo.bar  
  puts 'bar'  
end
```

```
foo.bar # => bar
```



# Class Methods

- ✧ Defined on a class
- ✧ Held in a virtual class



# Class Methods

```
class Cat  
  
  def self.says  
    'Meow'  
  end  
  
end
```



# Class Methods

- ✦ Why is it held in a virtual class?
- ✦ Cat is an instance of Class



# Instance Methods

- ✧ Defined on a class
- ✧ Held in the class



# Class Methods

```
class Cat  
  
  def meow  
    'meow meow meow !!!'  
  end  
  
end
```



# instance\_eval

- ✧ Used on an instance of a class to create a singleton method
- ✧ Used on a class to create a class method



# class\_eval

- ✦ Used on a class to create an instance method



# Metaclass Diagram



Cat



# Metaclass Diagram



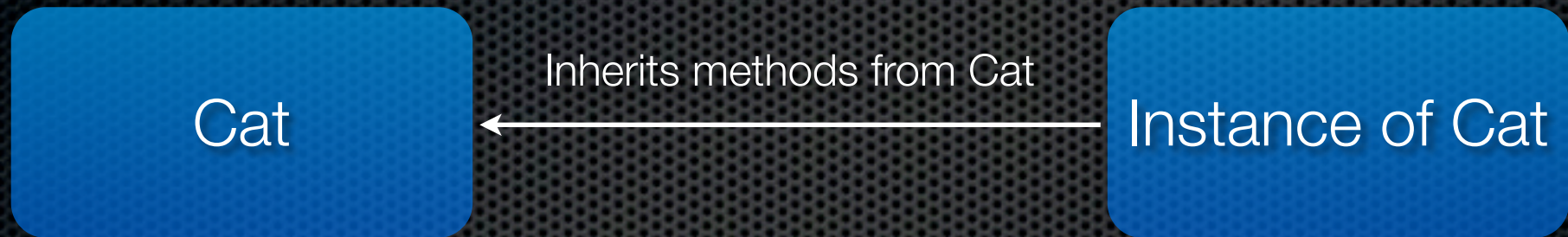
```
graph LR; Cat[Cat]; Instance[Instance of Cat];
```

Cat

Instance of Cat

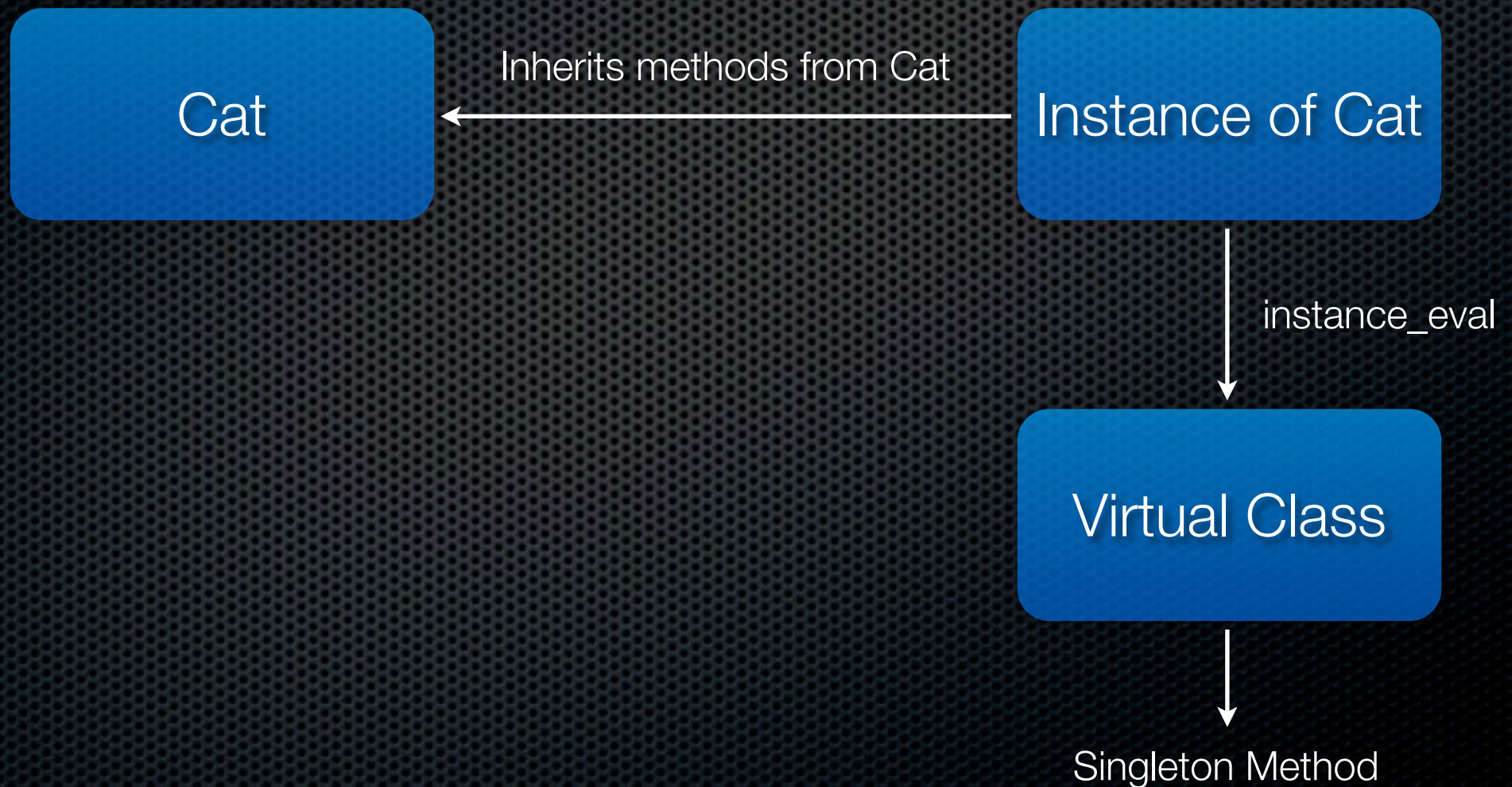


# Metaclass Diagram



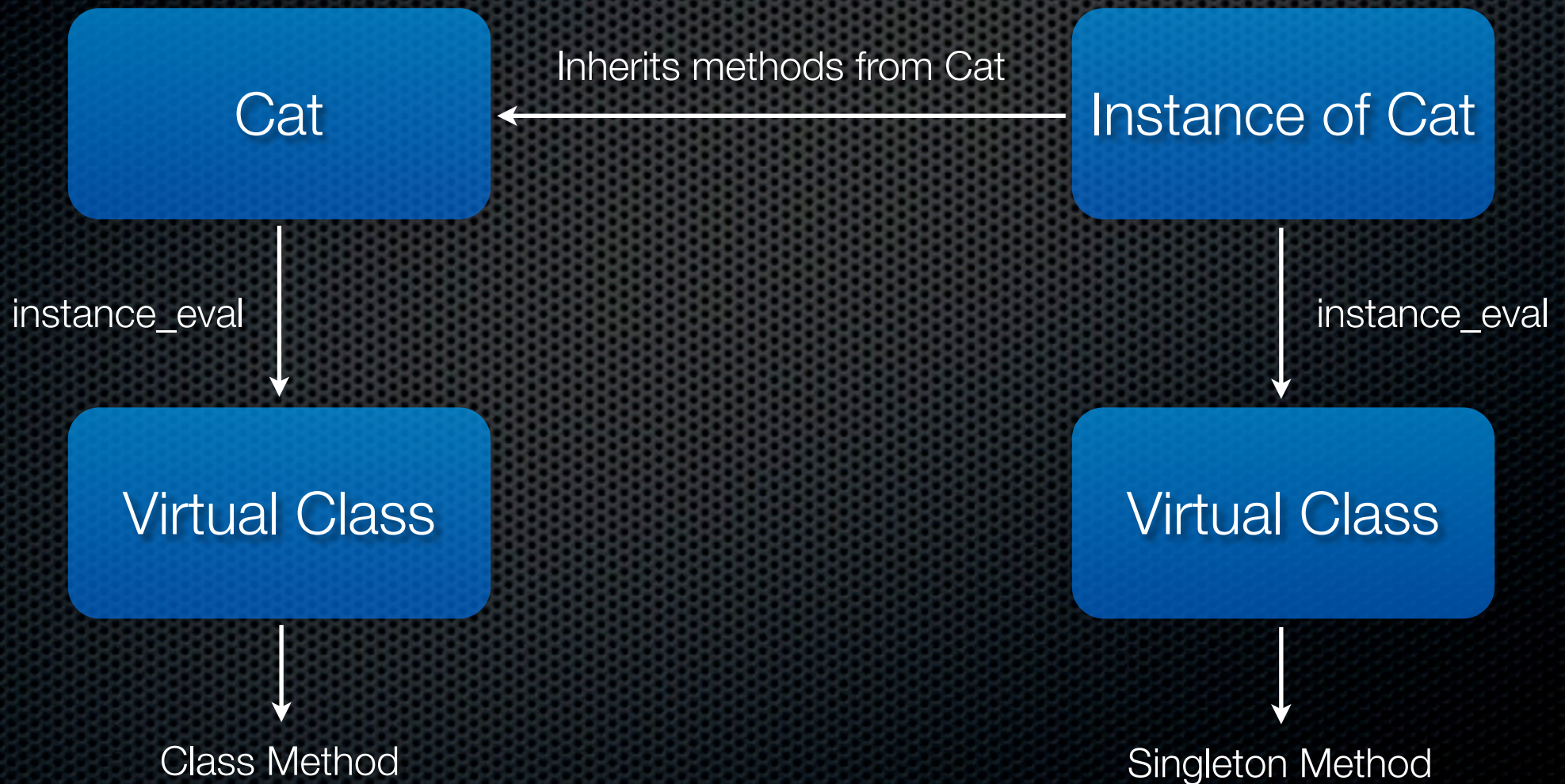


# Metaclass Diagram





# Metaclass Diagram





# Metaclass Diagram

