



BluePotion

RedPotion for Android

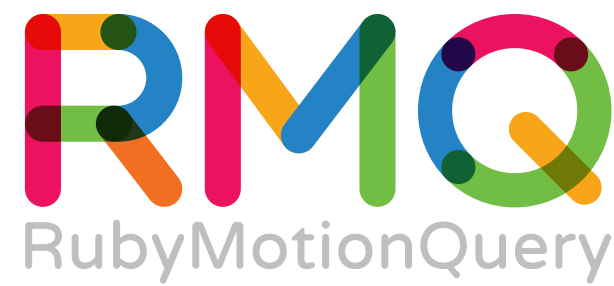


Presented by
Darin Wilson





Evolution of RedPotion



ProMotion



CORE DATA QUERY



AFMotion, motion-print,....



@darinwilson



infinitered

clearsight

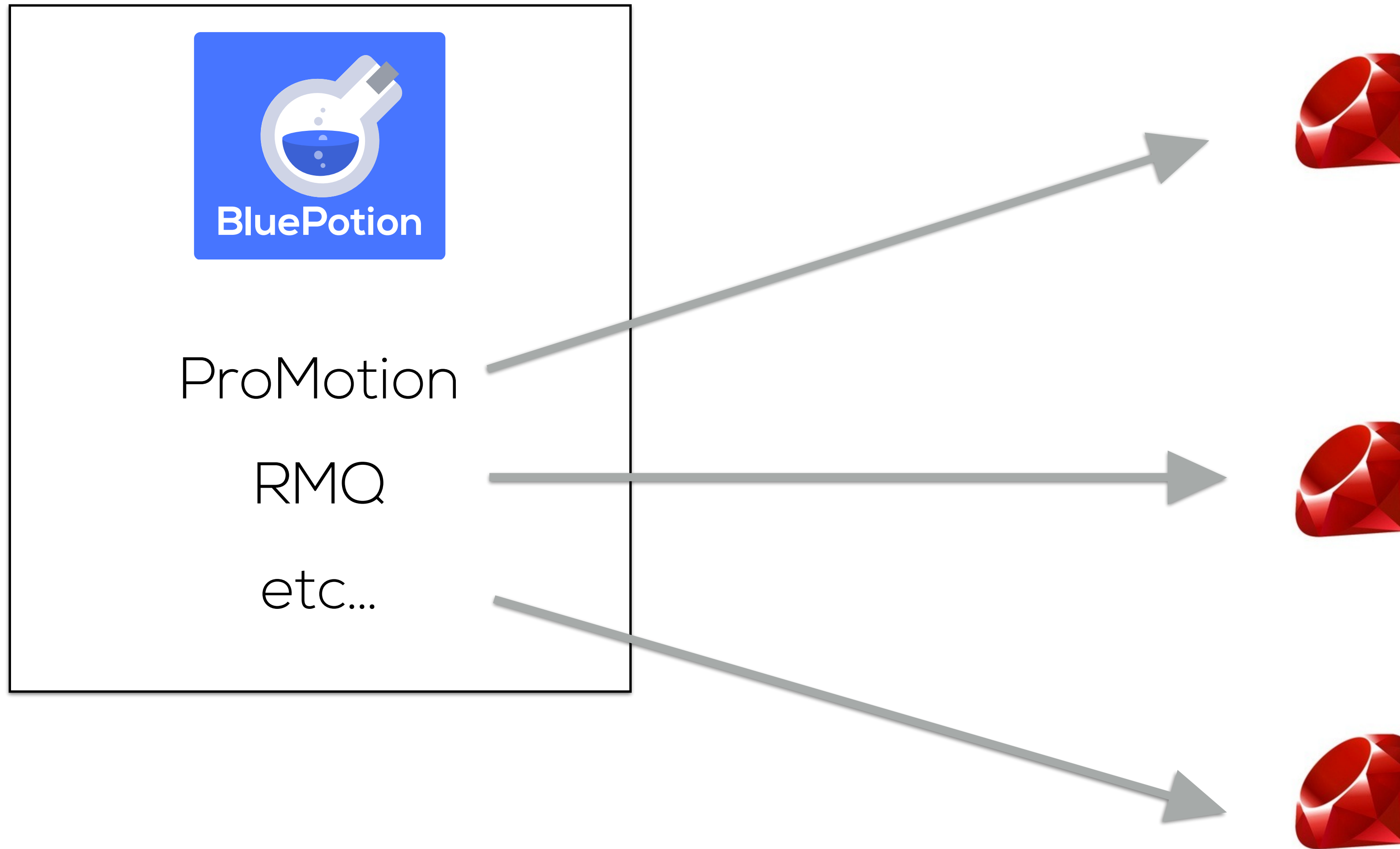
OMG!



Awesome



Evolution of BluePotion



@darinwilson





BluePotion

**How does it
compare?**



“Write once, run everywhere”



@darinwilson



infinitered

clearsight



~~“Write once, run everywhere”~~

Learn once, write anywhere

@darinwilson



infinitered

clearsight



Using [Red|Blue]Potion

- A common architectural framework
- Best practices
- A lot less typing

@darinwilson





BluePotion

ProMotion



Screens

```
class HomeScreen < PMScreen
  title "Home"

  def on_load
    # . . .
  end
end
```

@darinwilson





What's an Android screen?

Activity?

Fragment?

```
class PMScreen < Android::App::Fragment
```

@darinwilson



infinitered

clearsight



What's an Android screen?

“ You should implement an Activity when you want to allow others to launch specific parts of your app, such as for executing a share operation or opening some content contained in the app.

<https://medium.com/google-developers/developing-for-android-vii-the-rules-framework-concerns-d0210e52eee3>

@darinwilson





Screens

open DetailsScreen

open DetailsScreen, activity: MyActivity

@darinwilson



infinitered

clearsight



Screens

open `DetailsScreen`

by default, works like `UINavigationController`

@darinwilson





Screens with Arguments

```
open DetailsScreen, foo: "bar"
```

```
class DetailsScreen
```

```
  def foo=(foo)  
    @foo = foo  
  end
```

```
end
```

@darinwilson



infinitered

clearsight



Screen Lifecycle Events

```
class HomeScreen < PMScreen

  def on_load
    # Android: onActivityCreated
  end

end
```

@darinwilson





Screen Lifecycle Events

```
class HomeScreen < PMScreen  
  
    def on_start; end  
  
    def on_resume; end  
  
    def on_pause; end  
  
end
```

@darinwilson



infinitered

clearsight



Coming Soon

```
class MyListScreen < PMListScreen  
  
end
```

@darinwilson





BluePotion

RMQ



The Basics

```
append(Potion::TextView, :text_view)  
  .data("Show Details")  
  .on(:tap) do  
    open DetailsScreen  
  end
```

@darinwilson





Stylesheets

```
class HomeScreen < PMScreen  
  title "Home"  
  stylesheet HomeStyleSheet
```

```
  def on_load  
    append(Potion::TextView, :text_view)  
  end
```

```
end
```

@darinwilson



infinitered

clearsight



Stylesheets

```
class HomeStyleSheet
```

```
  def text_view(st)
```

```
    st.layout_width = :wrap_content
```

```
    st.layout_height = :wrap_content
```

```
    st.text_color = "#000000"
```

```
    st.text_size = 14
```

```
    st.padding_left = 8
```

```
  end
```

@darinwilson



infinitered

clearsight



Selecting

```
find(Potion::TextView)
```

```
find(:my_id)
```

```
find(:a_style)
```

@darinwilson



infinitered

clearsight



Selecting with Actions

```
find(Potion::TextView).hide
```

```
find(:my_id).disable
```

```
find(:a_style).apply(:another_style)
```

@darinwilson



infinitered

clearsight



Traversing

```
find(Potion::TextView).first
```

```
find(:my_id).parent.children
```

```
find(:a_style).siblings
```

@darinwilson



infinitered

clearsight



XML Support

```
class HomeScreen < PMScreen
```

```
  xml_layout :home_screen
```

```
  def on_load
```

```
    text_view = find(:text_view)
```

```
  end
```

```
end
```

@darinwilson





The Application Object

`app`

`app.identifier` => "com.infinitered.bluepotion"

`app.release?`

`app.test?`

`app.development?`

@darinwilson



infinitered

clearsight



`app.r`

`R::Layout::My_screen`

or...

`app.r.layout(my_screen)`

@darinwilson



infinitered

clearsight



app.net

```
url = "http://openweathermap.org/data/2.1/find/name"

app.net.get_json(url, q: "paris") do |response|
  if response.success?
    temp = response.object["list"]
      .first["main"]["temp"]
    app.toast("The weather is #{temp} degrees")
  end
end
```

@darinwilson





app.async

```
app.async do
```

```
  # run something in the background
```

```
end
```

@darinwilson



infinitered

 clearsight



app.async

```
app.async do
  # fetch some data
end.on(:completion) do |data|
  update_views(data)
end
```

@darinwilson





device

`device.os_version` `#=>` 5.0

`device.sdk_version` `#=>` 21

`device.sdk_at_least?(18)` `#=>` true

`device.width` `#=>` 1080

`device.height` `#=>` 1776

@darinwilson



infinitered





And more...

<https://github.com/infinitered/bluepotion>

@darinwilson





BluePotion

Thank you

bluepotion.org

