AP4R

Asynchronous Processing for Ruby
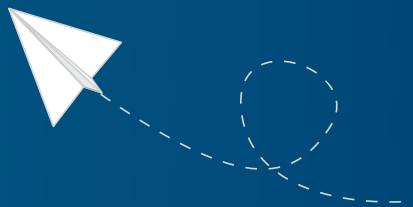
what is it?

- asynchronous job processor

- you mean like backgroundrb?

- yes! and no!

features

- async. message queueing and dispatching

- reliable messaging

- load balancing

the specials

- async. logic without extra workers

- an async. task is just another controller action

- supports http post, soap, xmlrpc

delivery

- immediately through drb

- store and forward
  - first store message (through activerecord)
  - then deliver
  - ensures guaranteed delivery

message storage

- file based

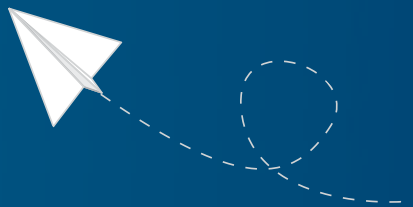- database (for now only mysql)

scenarios

- mass mailings

- cache sweeping

- search index rebuilding

- feed aggregation

setup

- ap4r is a gem

- and a plugin

- setup
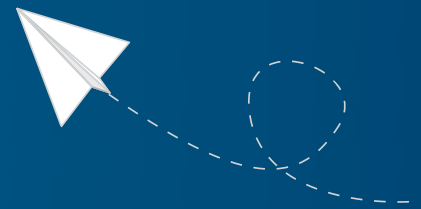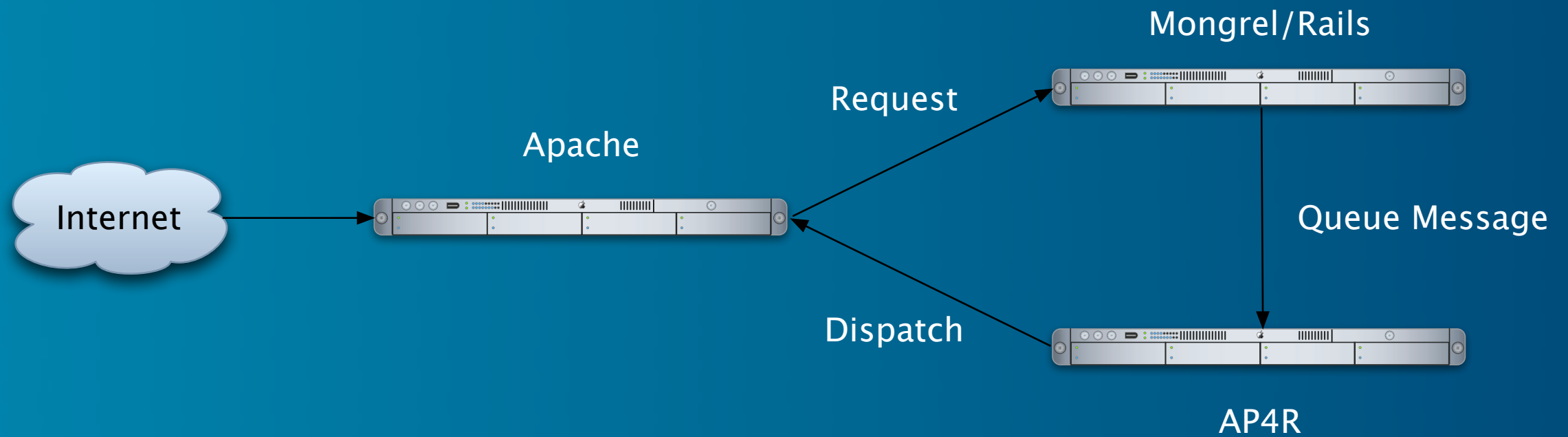 - ap4r_setup ap4r_server
  - install rails plugin

setup

- ap4r dispatcher runs with mongrel
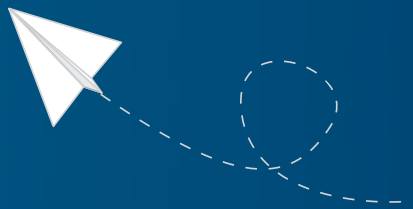
- checks for messages and dispatches

how it works

Mongrel/Rails

Apache

Internet

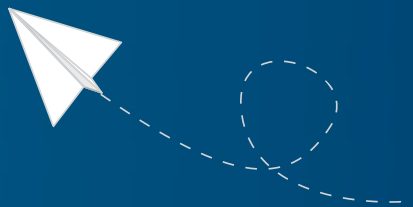Request

Queue Message

Dispatch

AP4R

but what about backgroundrb?

- ap4r is simpler

- integrates more seamlessly with rails

- it can async'ly execute http calls on any http server

but i love backgroundrb!

- what's in the background stays in the background

- no scheduling support

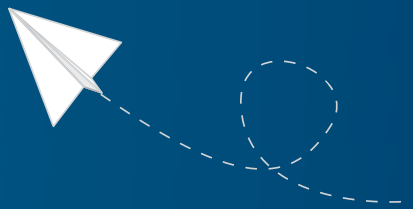- but backgroundrb doesn't have reliable scheduling anyway

code!

- immediate delivery

```
ap4r.async_to( {:controller => 'search', :action => 'reindex'},
               {:id => 12} )


ap4r.async_to( {:url => 'http://localhost:8080/someServlet'},
               {:id => 12} )
```
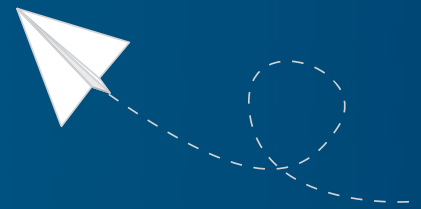
code!

- store and forward

```
ap4r.transaction do
  ap4r.async_to( {:controller...},
                 {:id => 12} )


end
```
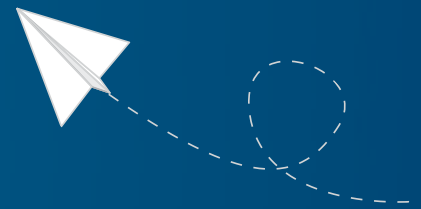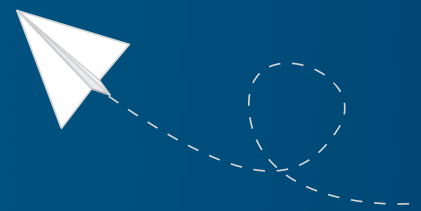
code!

- store and forward

```
ap4r.transaction do
  ap4r.async_to( {:controller...},
                 {:id => 12} )



end
```

Store in SAF

code!

- store and forward

```
ap4r.transaction do
  ap4r.async_to( {:controller...},
                 {:id => 12} )



end
```

Store in SAF

Commit Database, Deliver Message

further reading

- http://www.infoq.com/news/2007/06/messaging-with-ap4r

- http://ap4r.rubyforge.org/wiki/wiki.pl?GettingStarted

- http://rubyforge.org/docman/view.php/1765/1257/AP4R_on_RubyKaigi2007_EN.pdf

thanks!