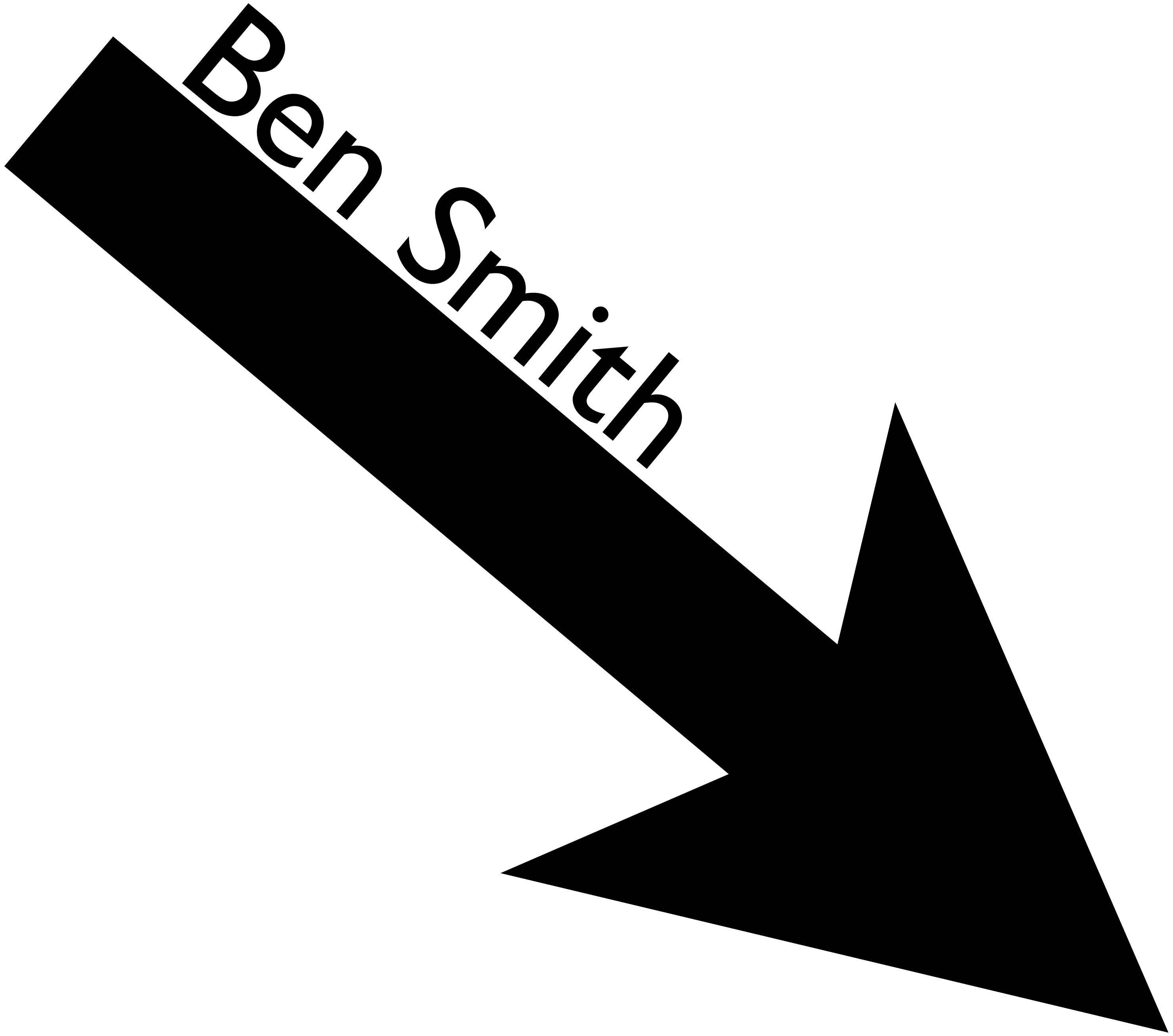
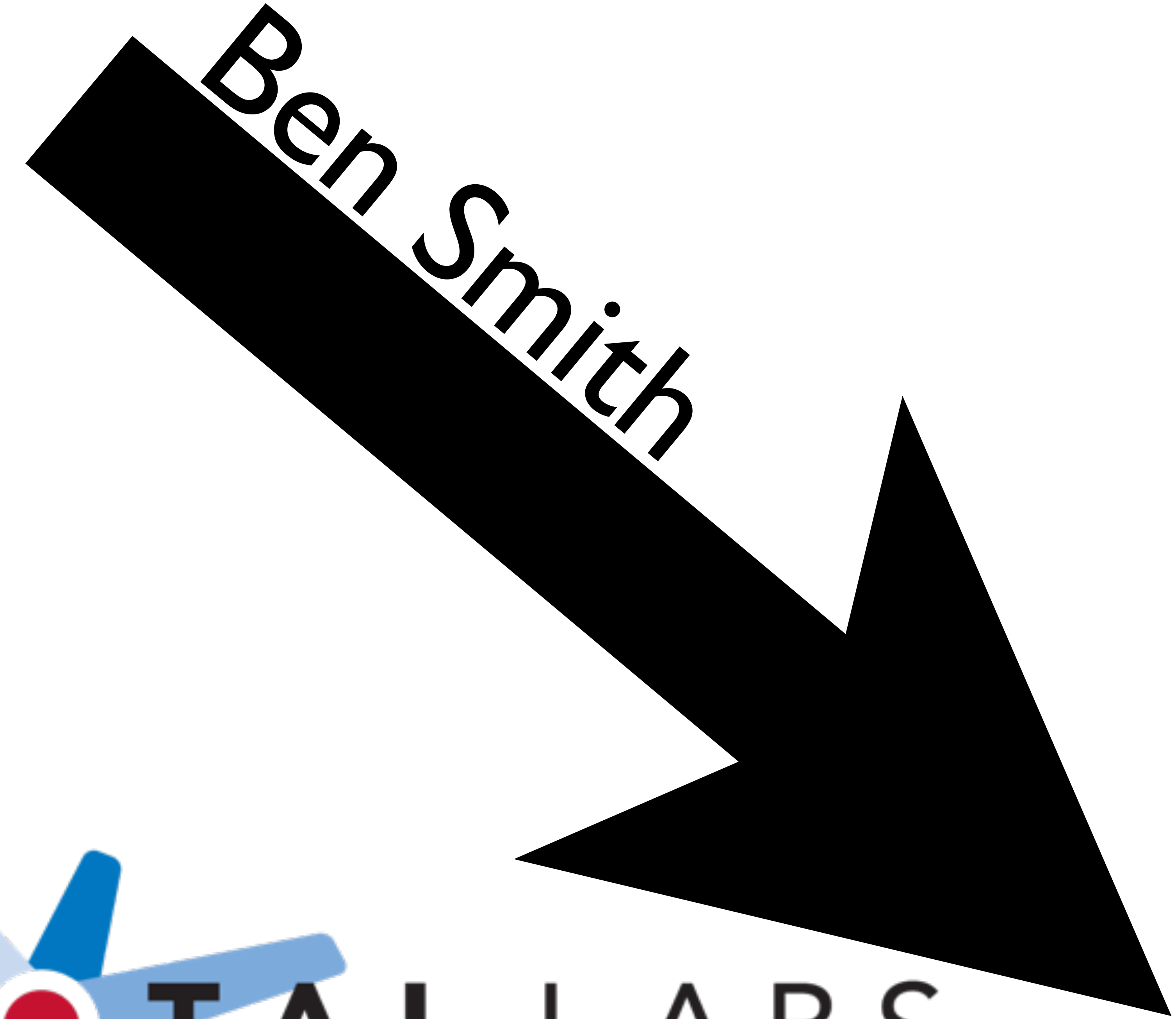
Architecting your Rails app for success!

Benjamin Smith



Ben Smith



Ben Smith

PIVOTAL LABS

Ben Smith

PIVOTAL LABS

Ben Smith

RAILS

PIVOTAL LABS

@benjamin_smith

Ben Smith

RAILS

PIVOTAL LABS

Architecting your Rails app for success!

Benjamin Smith

@benjamin_smith

PIVOTAL LABS

MVC

once upon a time

success

long project

NASA
MISSION CONTROL CENTER
esa
CAPCOM
PAYLOAD DEPLOY/
RETRIEVAL (PDRS)

big dev team

lots of codez

scalable

big bang release

success

- long project
- big dev team
- lots of code
- scalable
- big 1st release

a grain of salt

@benjamin_smith

PIVOTAL LABS

success

- long project
- big dev team
- lots of code
- scalable
- big 1st release

@benjamin_smith

engines

a little background

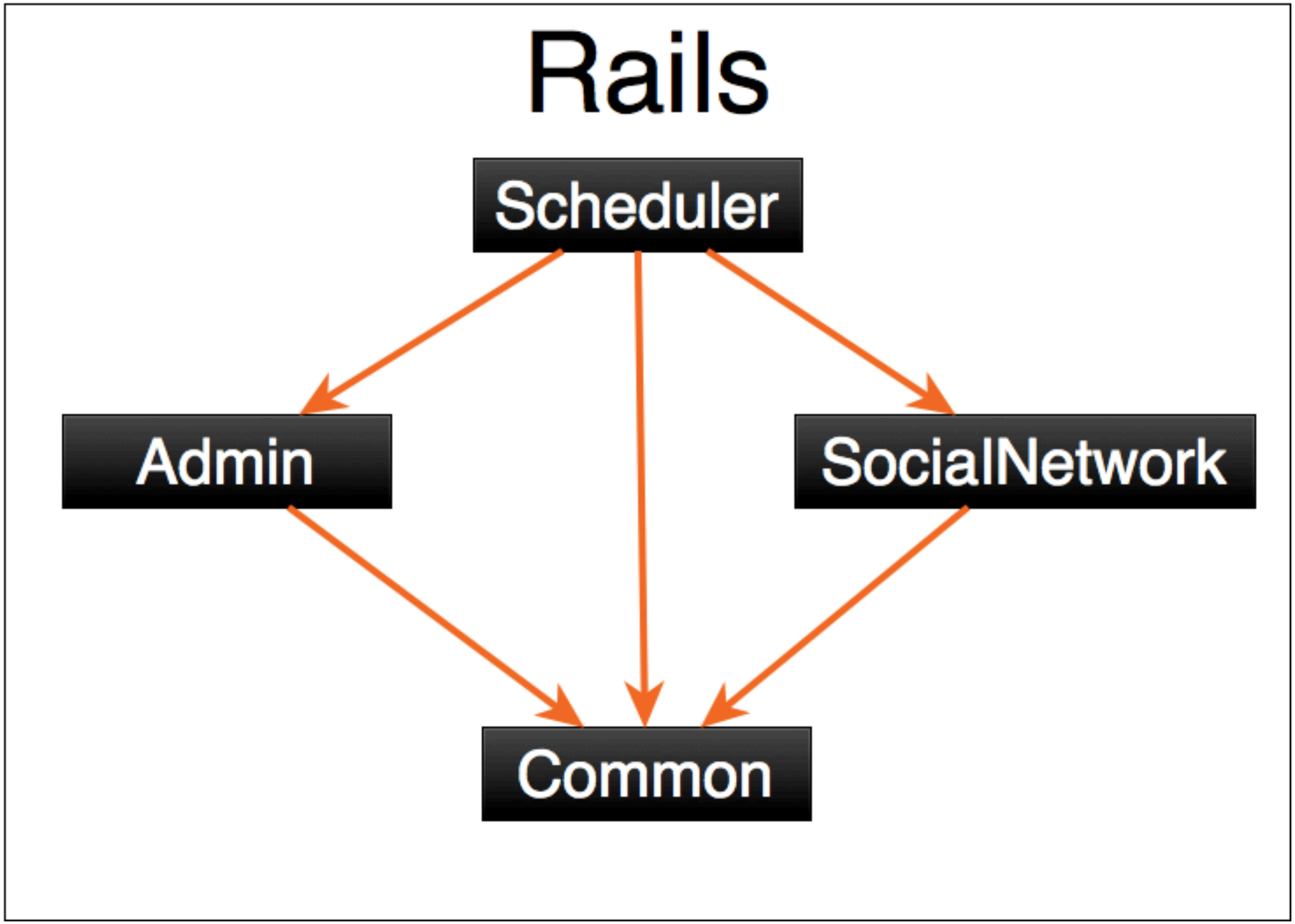
@benjamin_smith

PIVOTAL LABS

the product

- Multimedia content publishing
- A social network
- iOS app for consumers (using JSON API)
- CMS (Rails) for admins

started simple



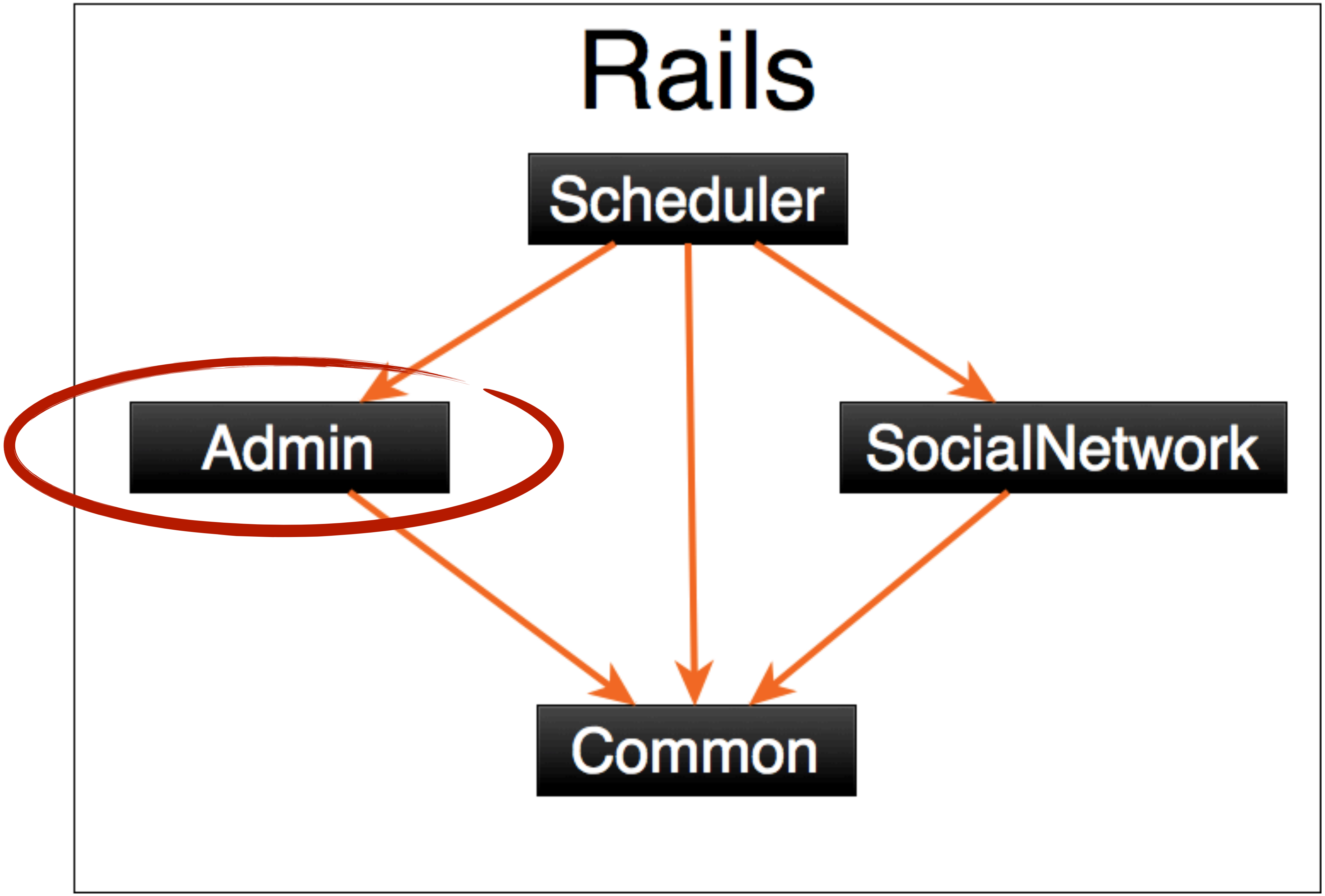
Rails

Scheduler

Admin

SocialNetwork

Common



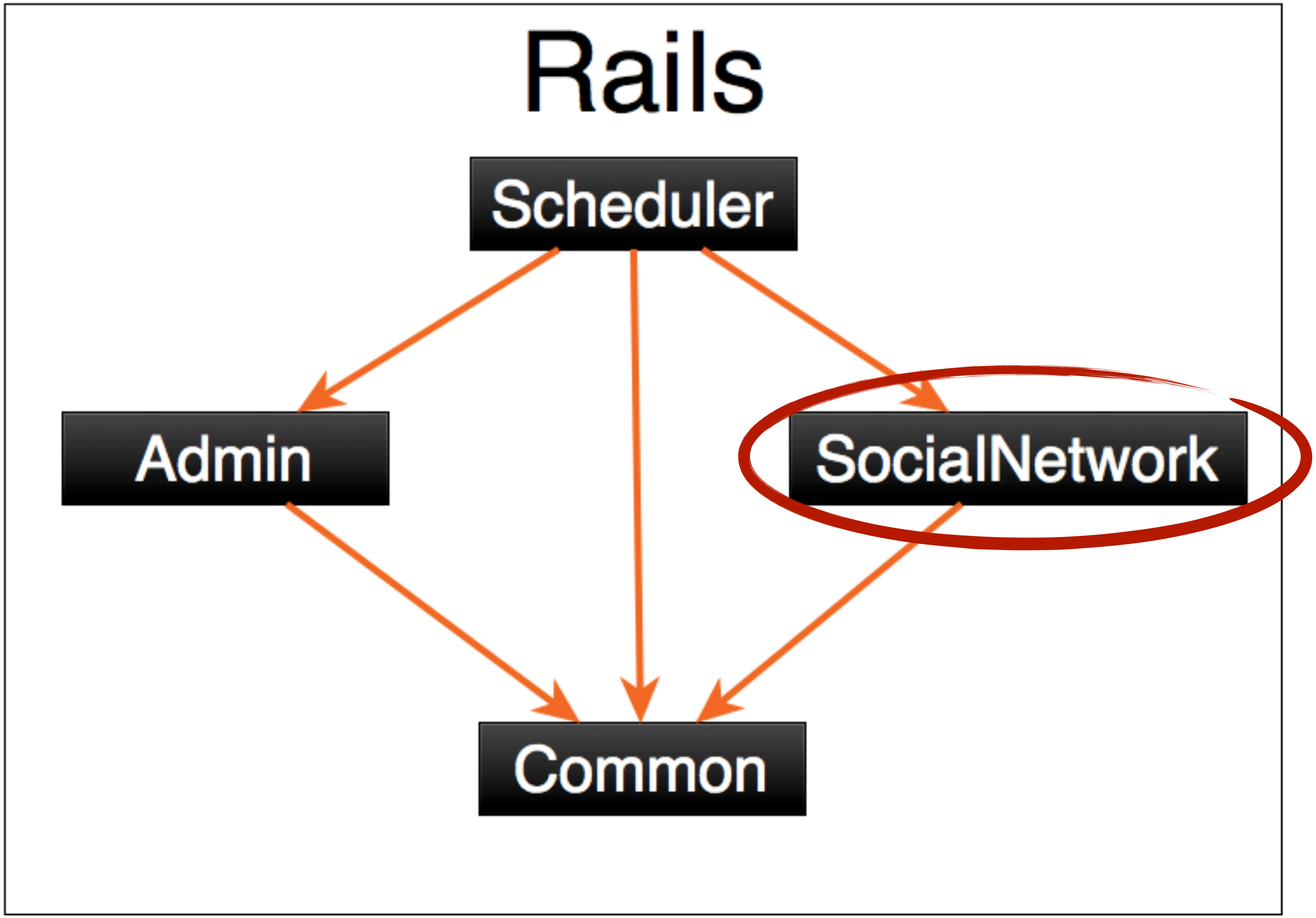
Rails

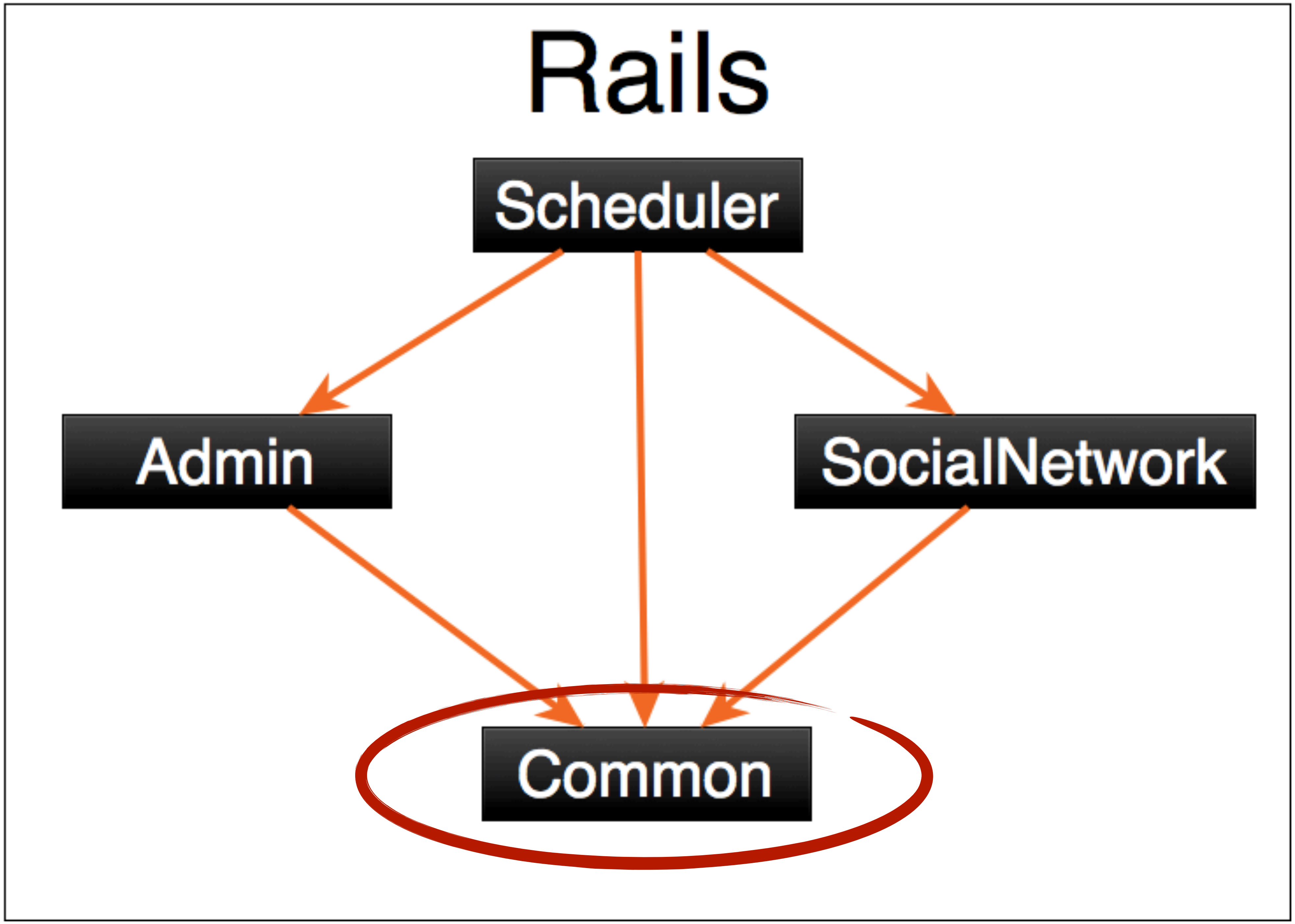
Scheduler

Admin

SocialNetwork

Common


Rails
Scheduler
Admin
SocialNetwork
Common



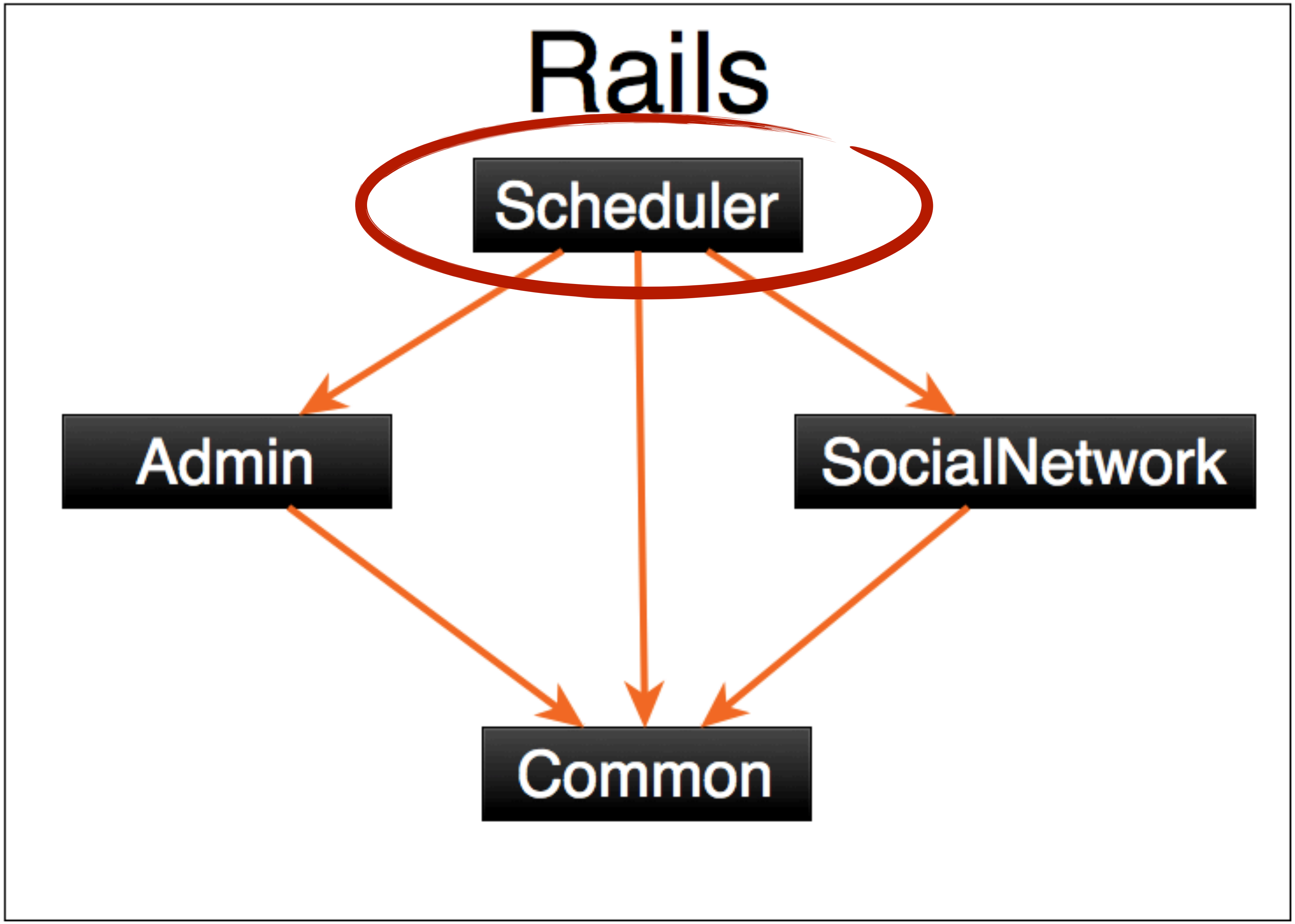
Rails

Scheduler

Admin

SocialNetwork

Common



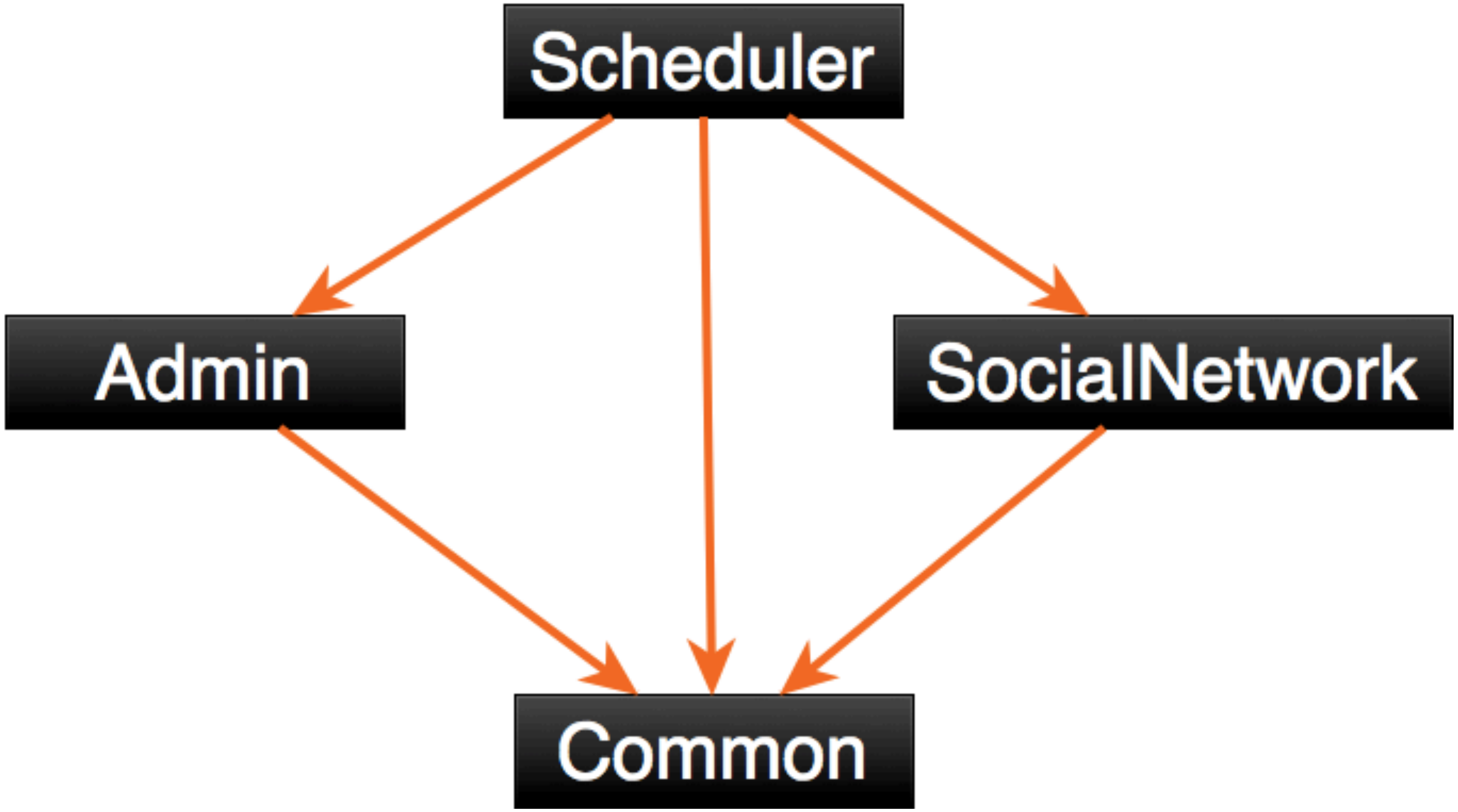
Rails

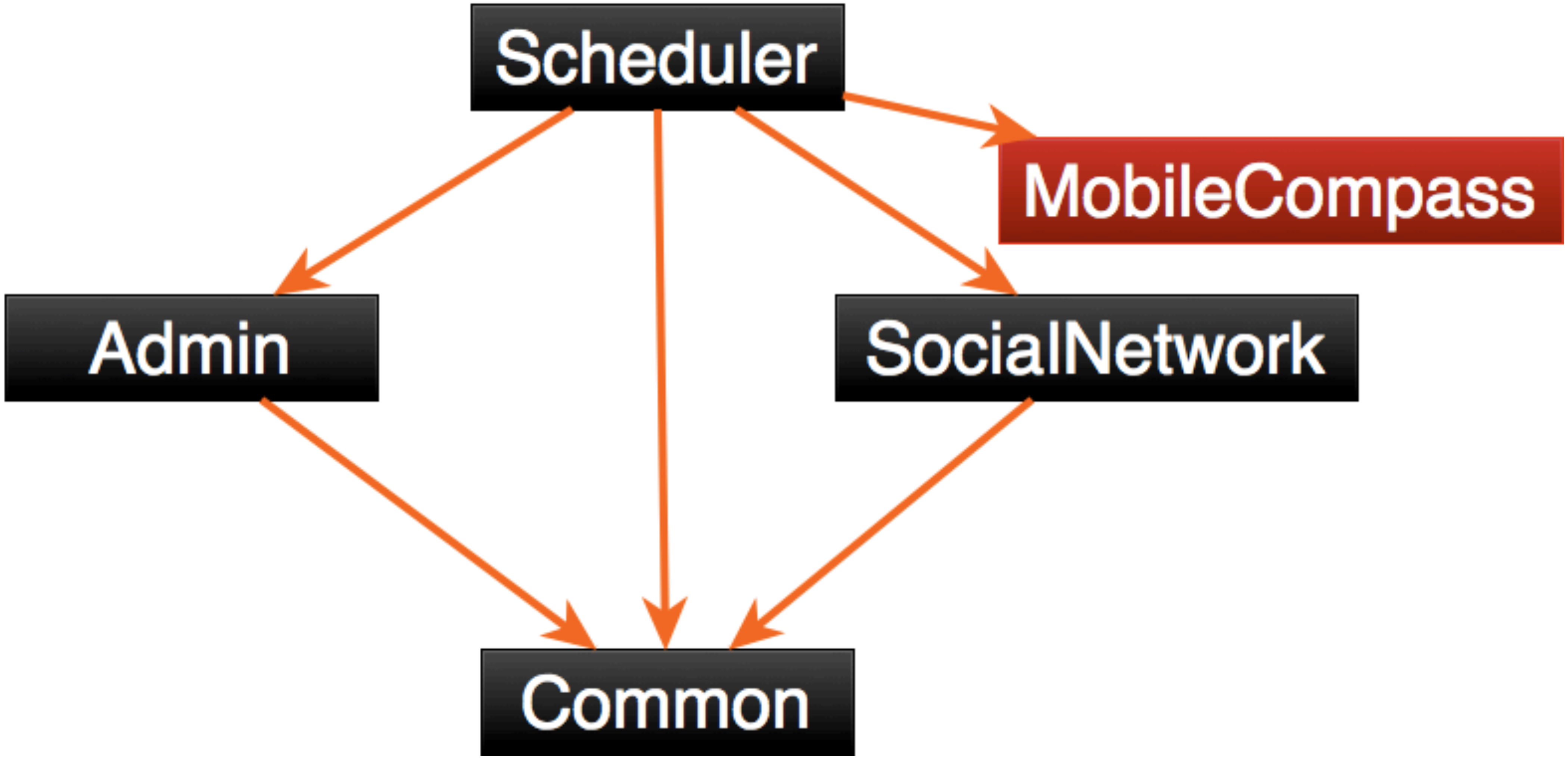
Scheduler

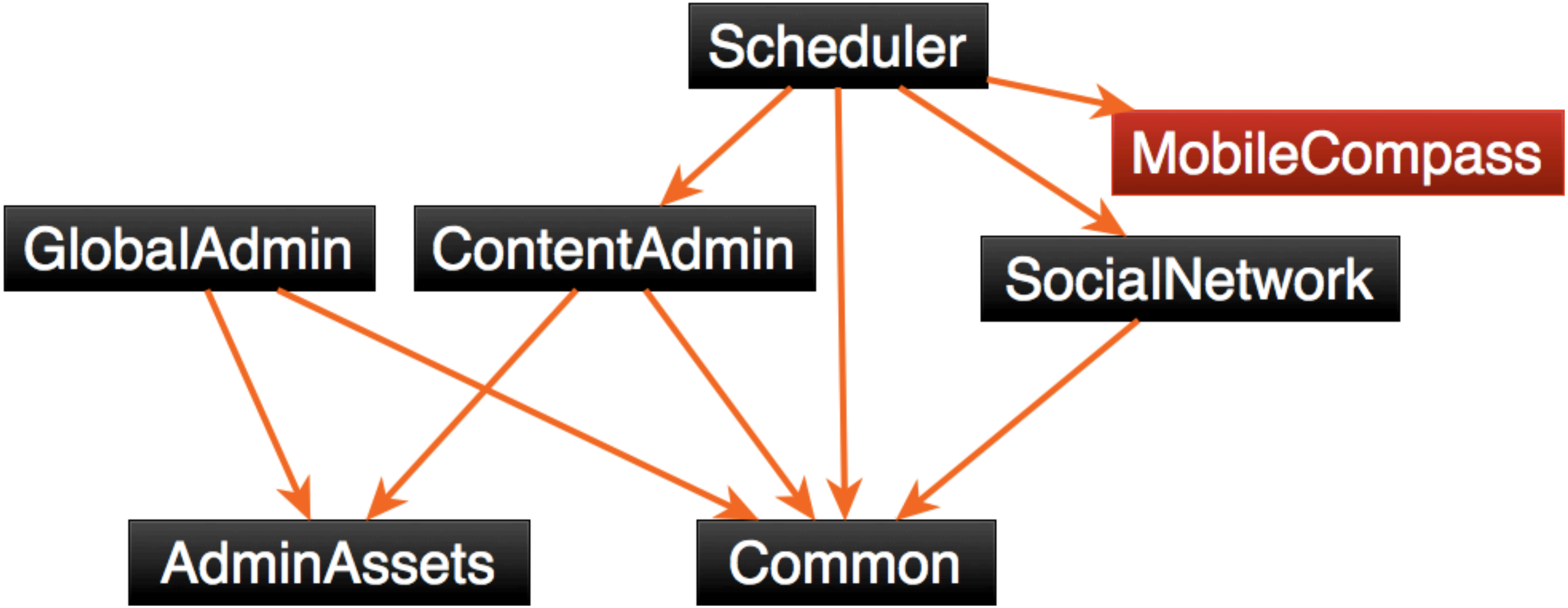
Admin

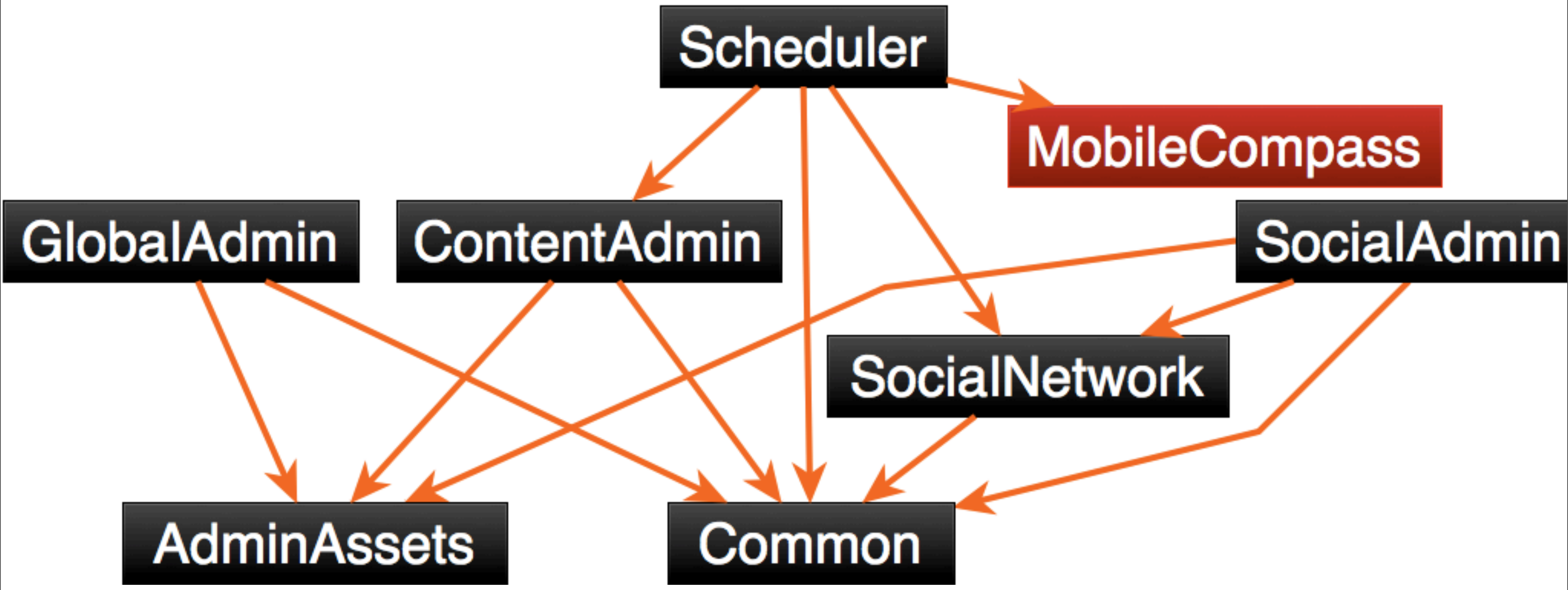
SocialNetwork

Common

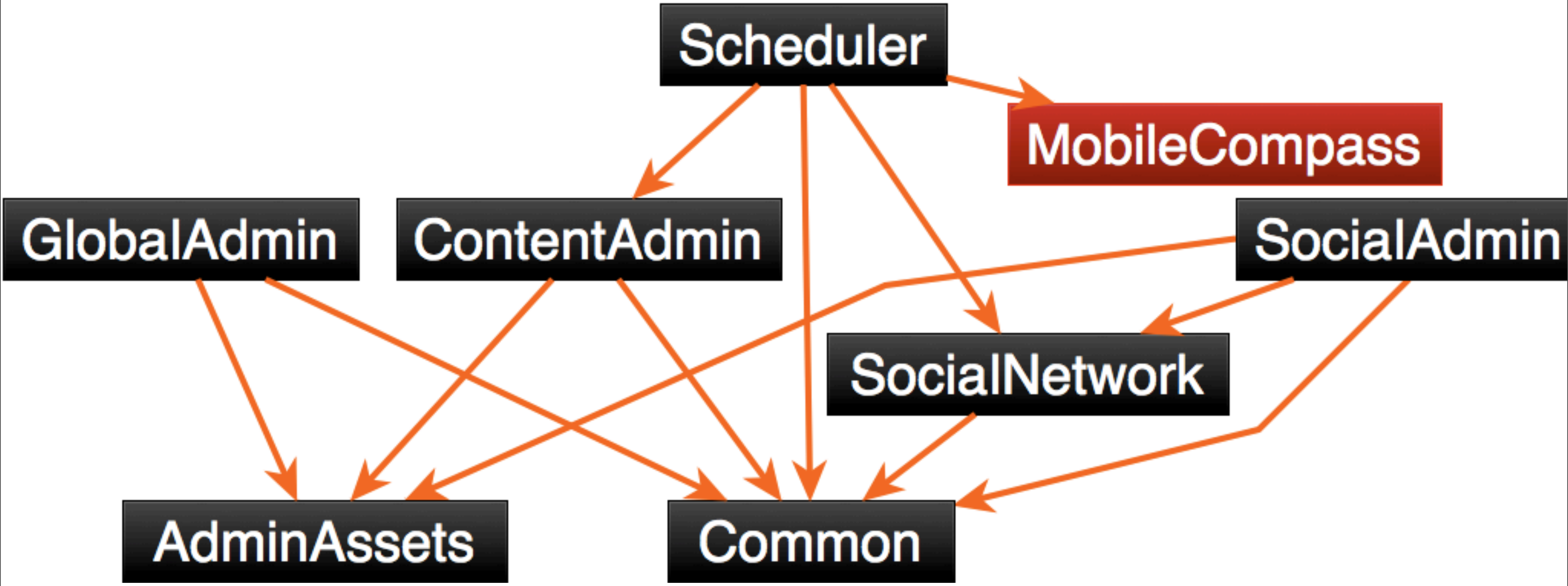

Scheduler
Admin
SocialNetwork
Common

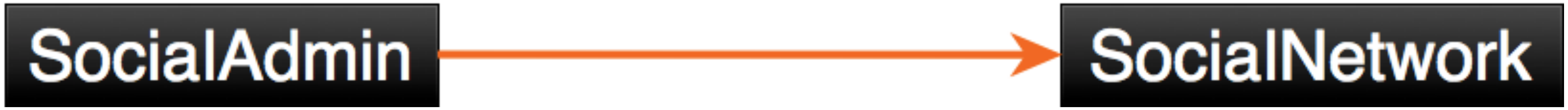

Scheduler
MobileCompass
Admin
SocialNetwork
Common


Scheduler
MobileCompass
GlobalAdmin
ContentAdmin
SocialNetwork
AdminAssets
Common


Scheduler
MobileCompass
GlobalAdmin
ContentAdmin
SocialAdmin
SocialNetwork
AdminAssets
Common

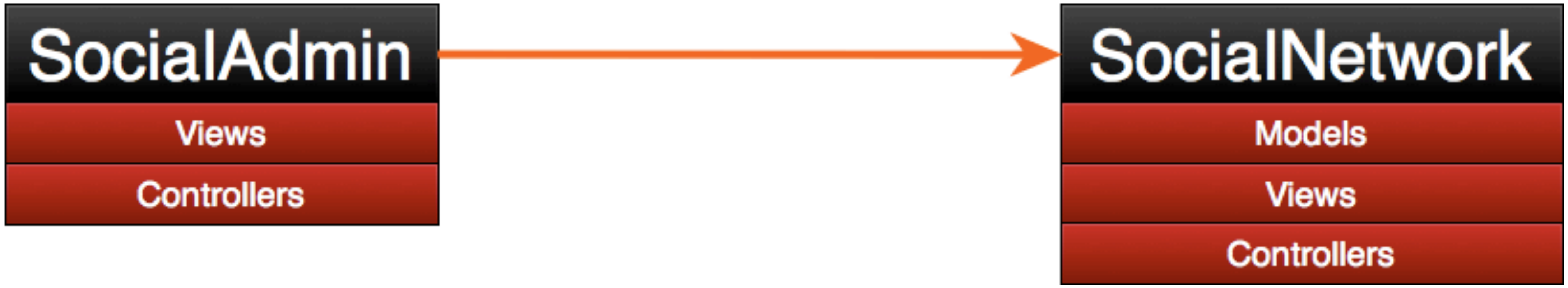
not too bad


Scheduler
MobileCompass
GlobalAdmin
ContentAdmin
SocialAdmin
SocialNetwork
AdminAssets
Common



SocialAdmin

SocialNetwork



SocialAdmin

Views

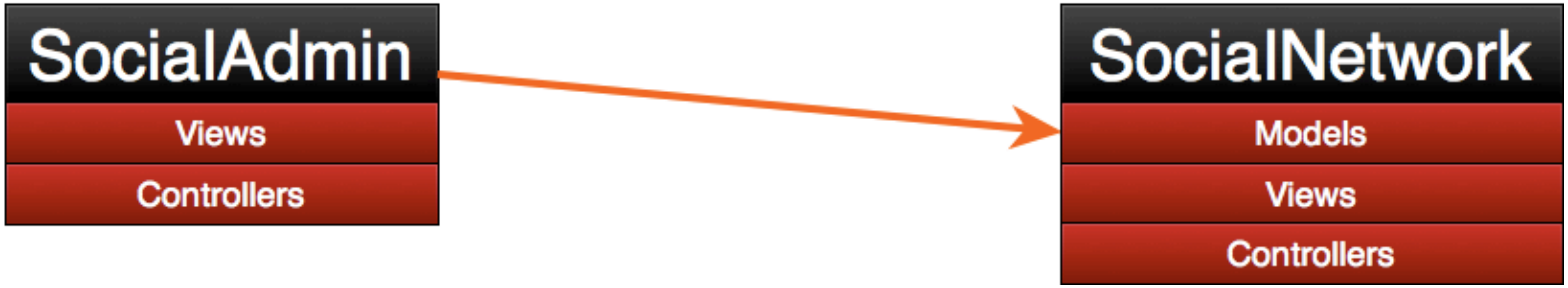
Controllers

SocialNetwork

Models

Views

Controllers



SocialAdmin

Views

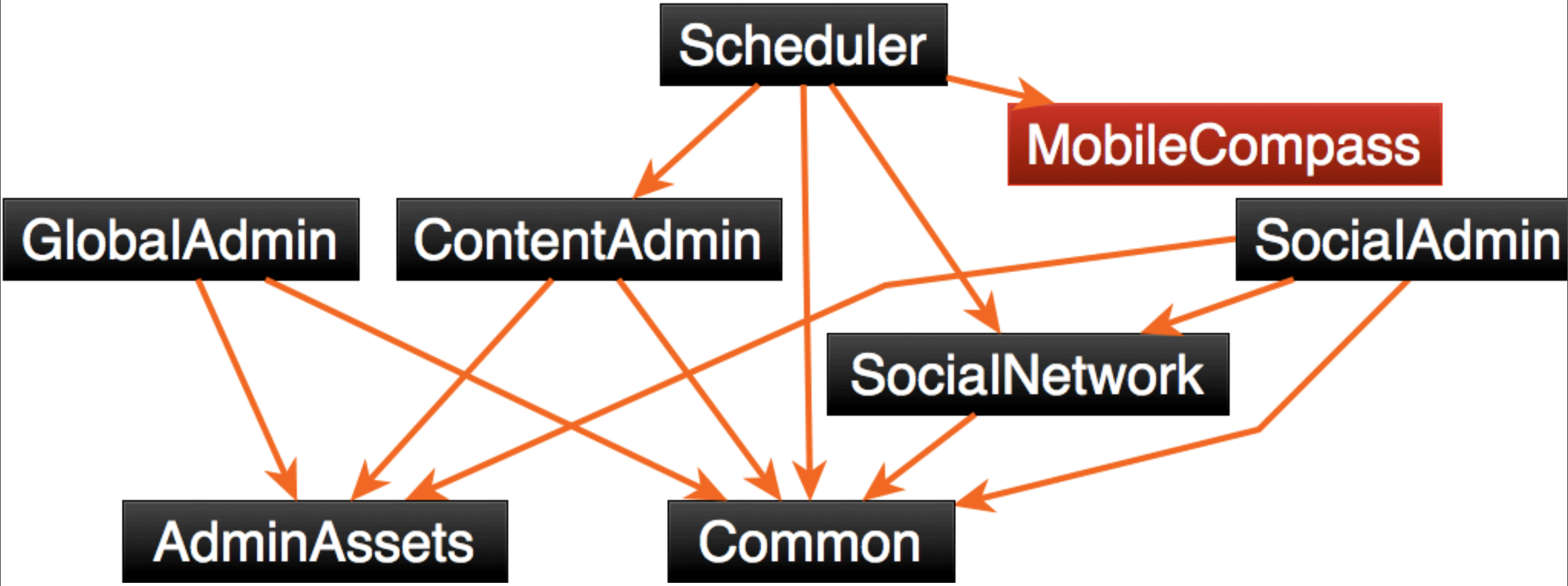
Controllers

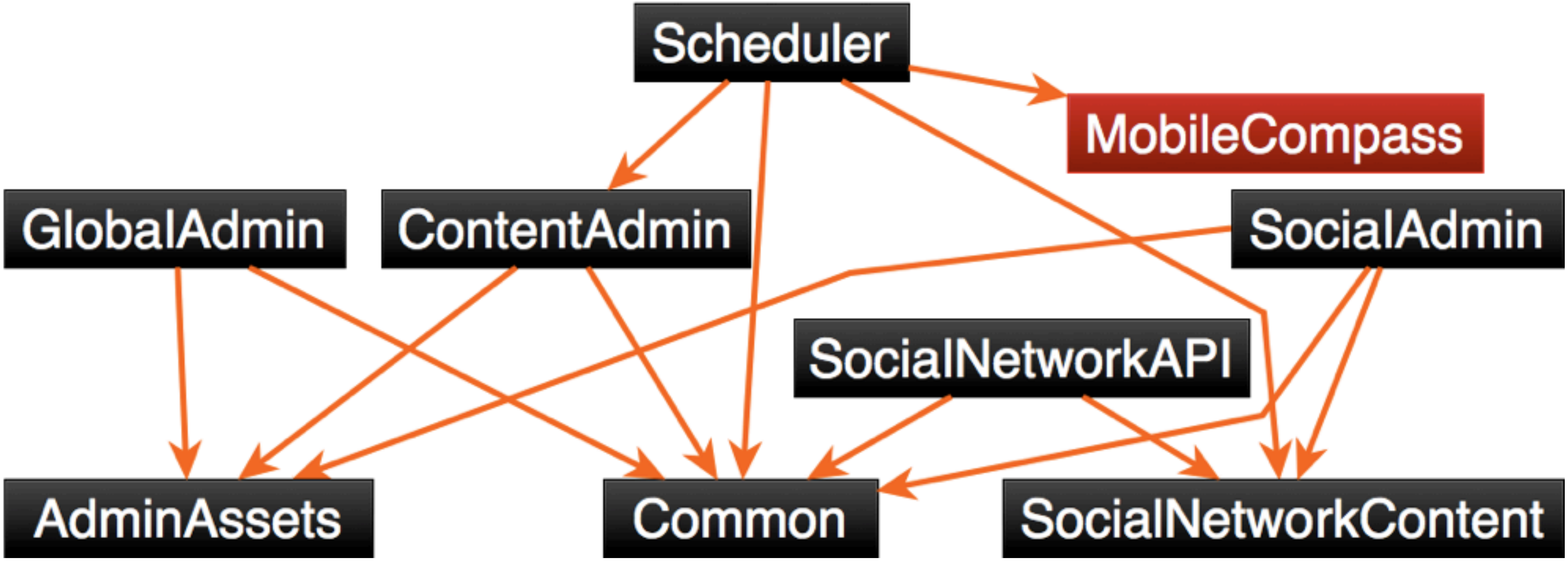
SocialNetwork

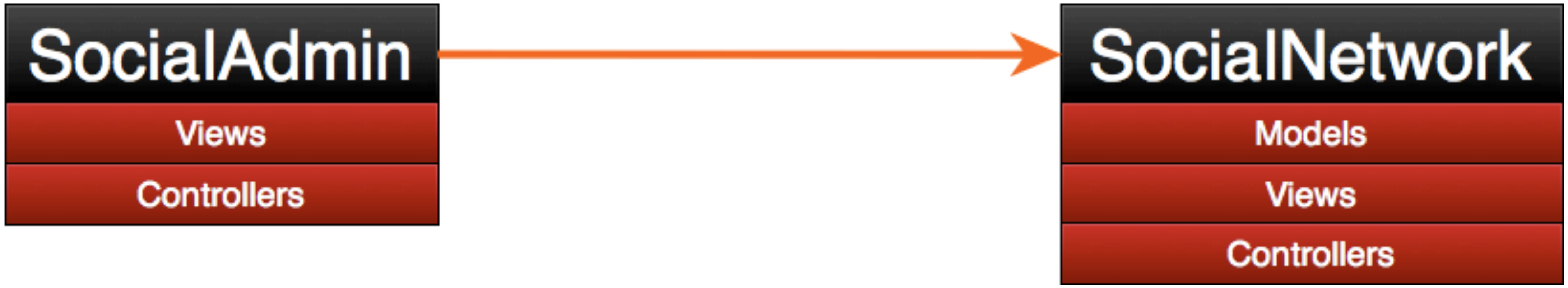
Models

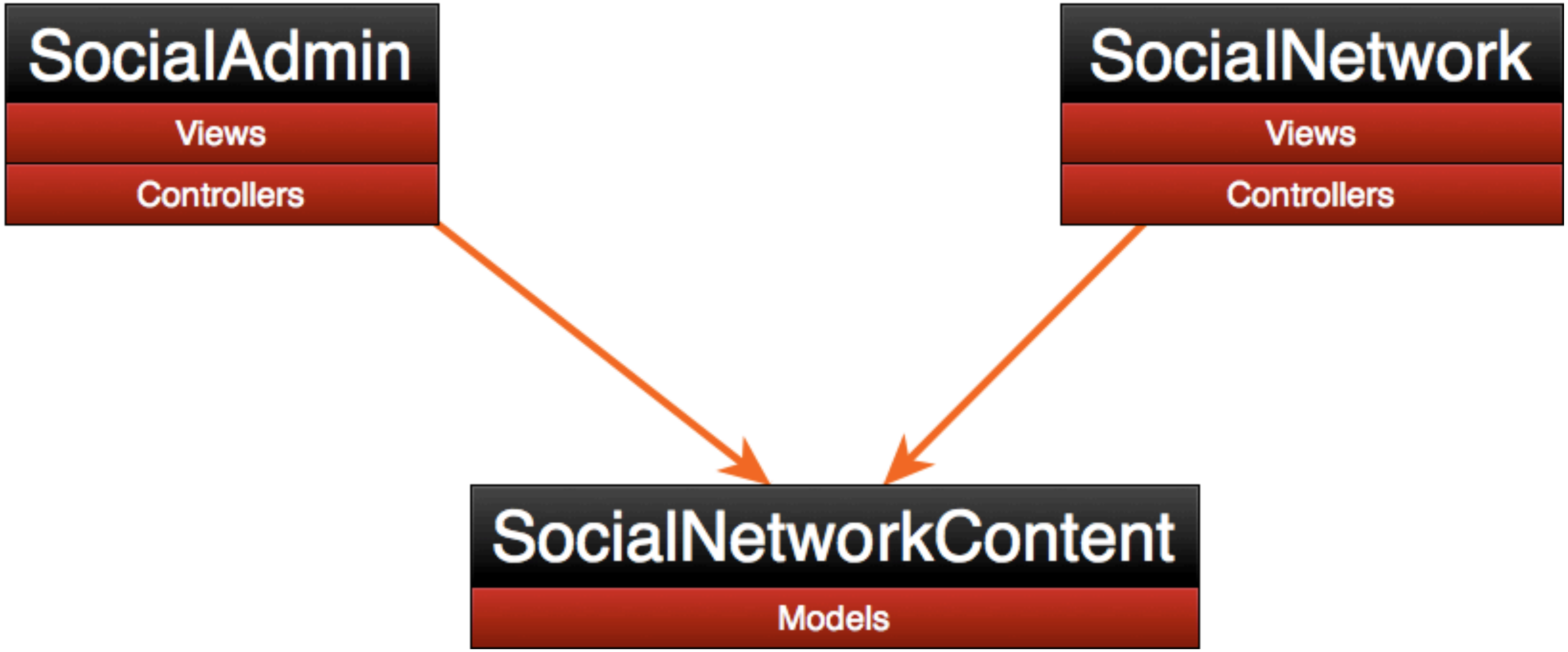
Views

Controllers

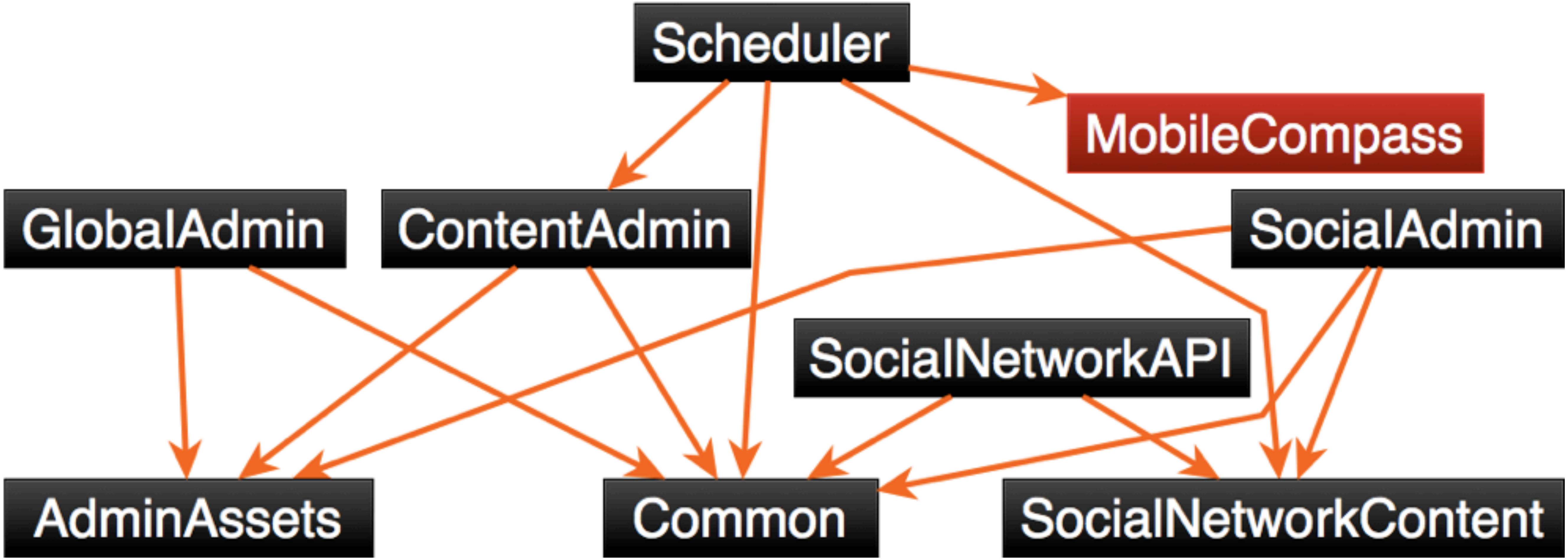

Scheduler
MobileCompass
GlobalAdmin
ContentAdmin
SocialAdmin
SocialNetwork
AdminAssets
Common

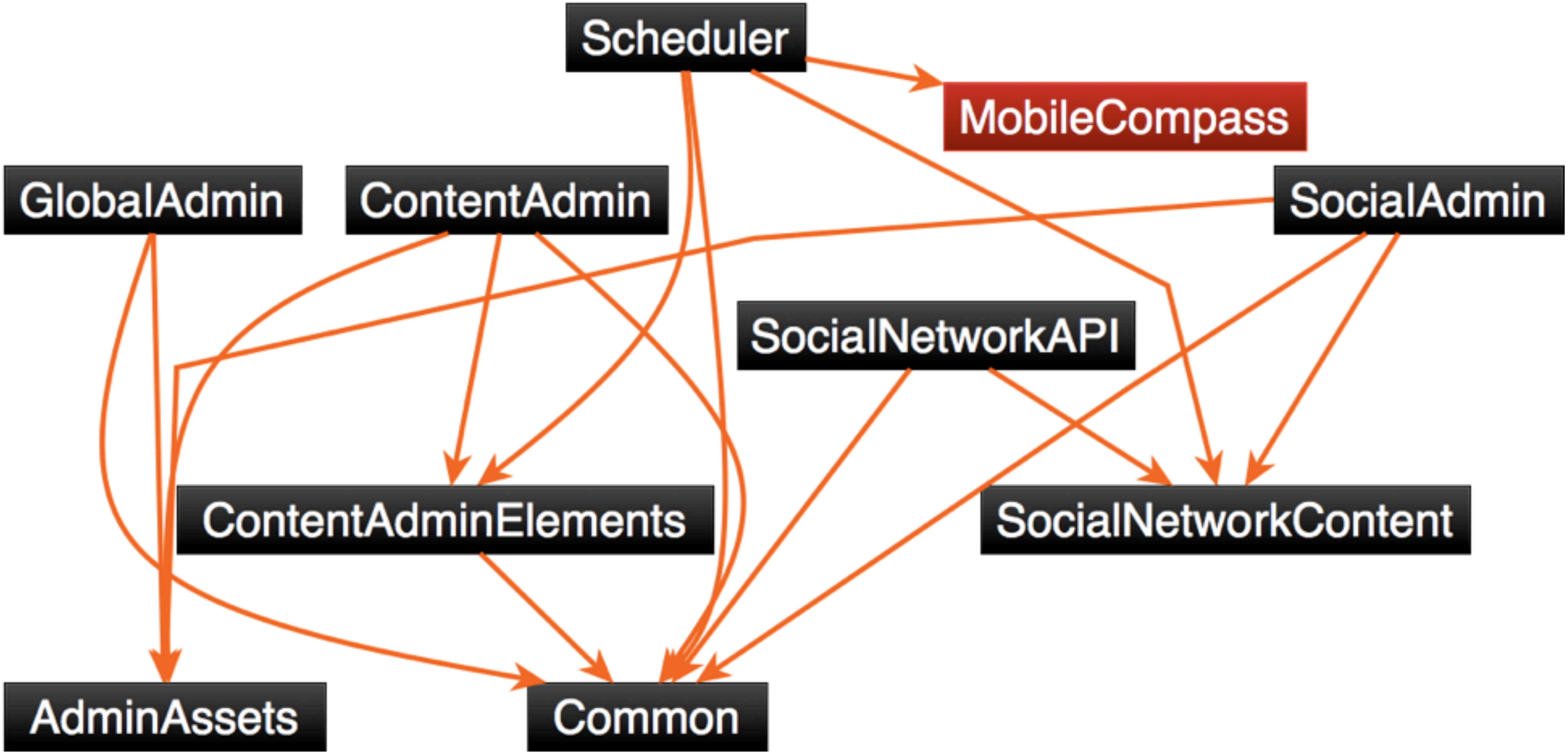

Scheduler
MobileCompass
GlobalAdmin
ContentAdmin
SocialAdmin
SocialNetworkAPI
AdminAssets
Common
SocialNetworkContent

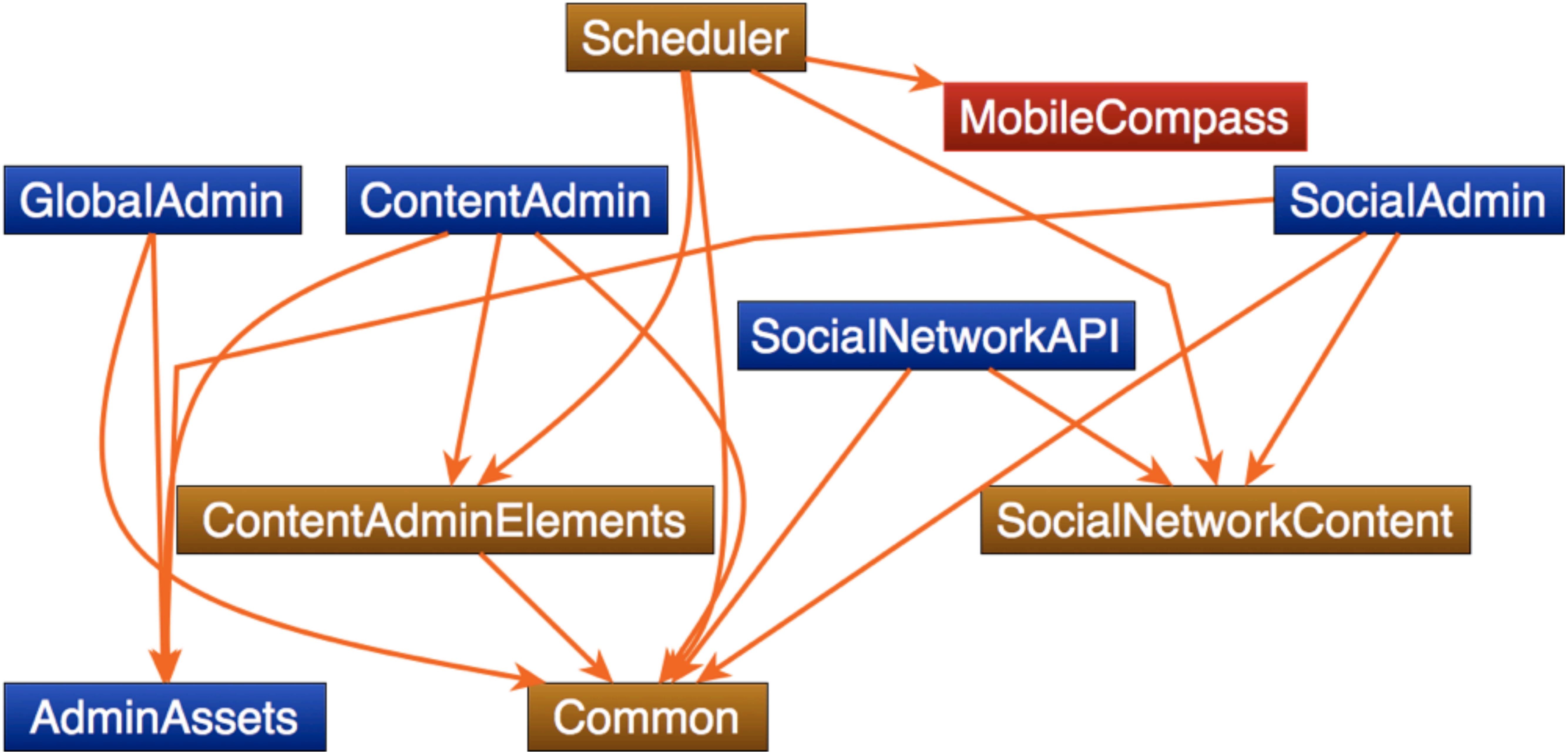

SocialAdmin
Views
Controllers
SocialNetwork
Models
Views
Controllers


SocialAdmin
Views
Controllers
SocialNetwork
Views
Controllers
SocialNetworkContent
Models

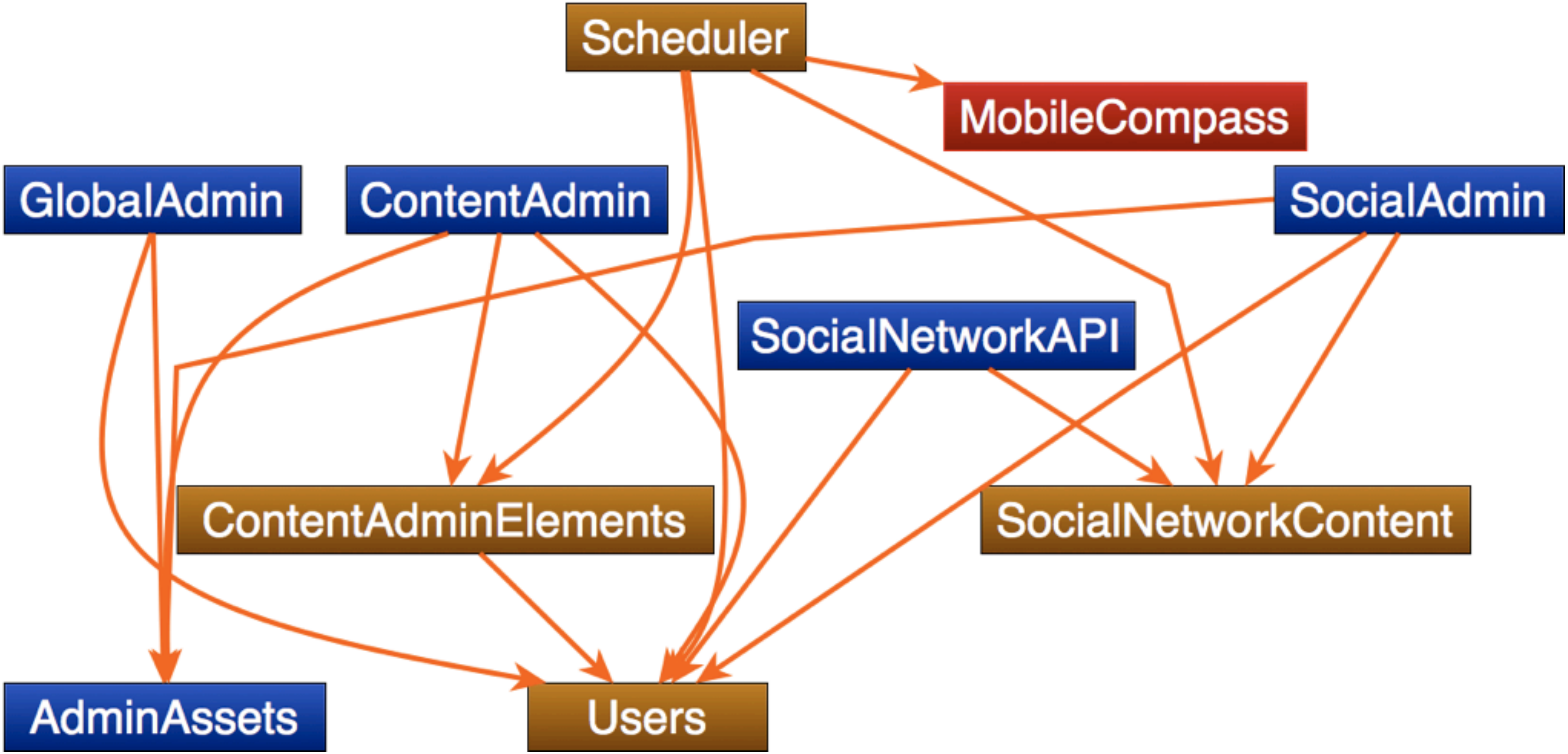
but why stop there?

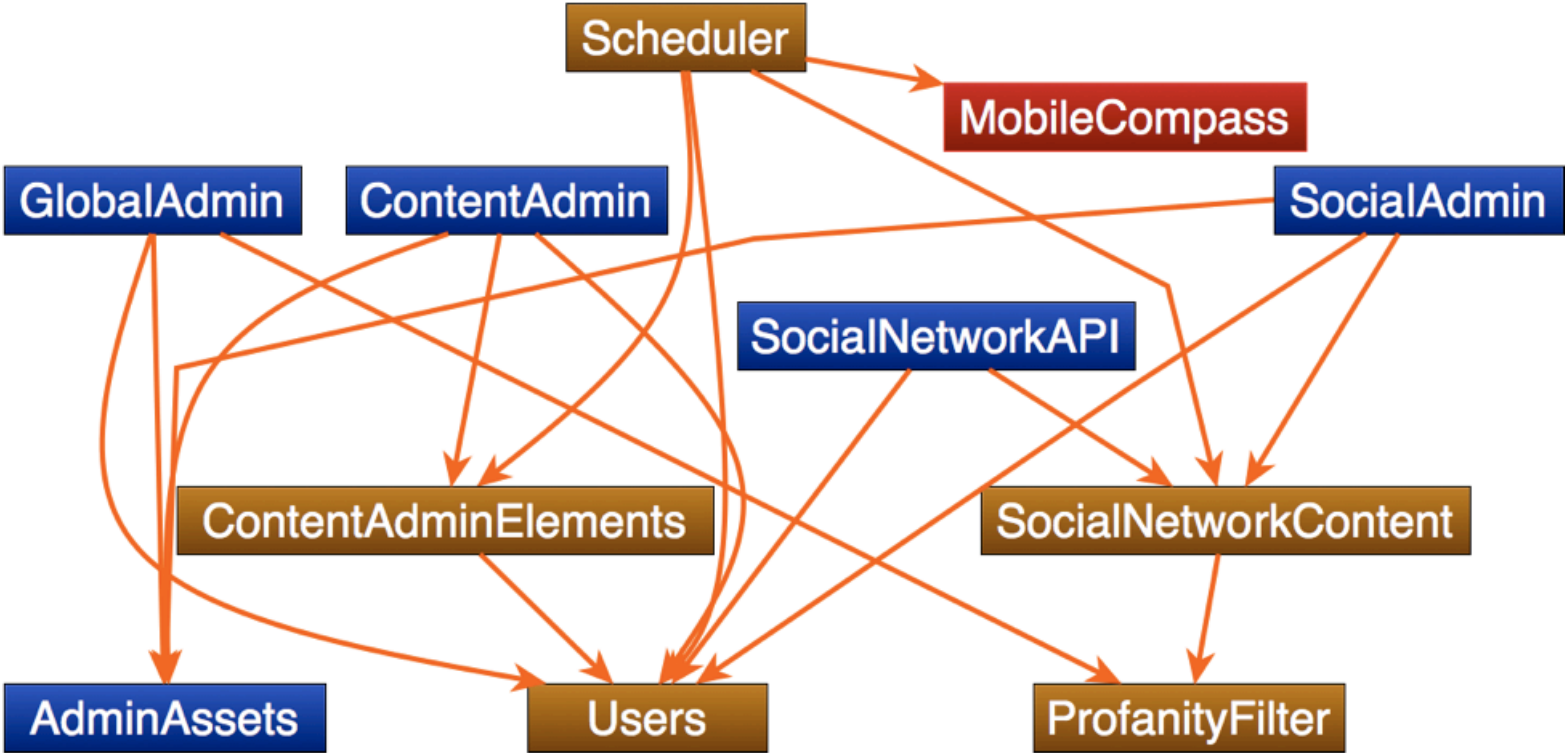

Scheduler
MobileCompass
GlobalAdmin
ContentAdmin
SocialAdmin
SocialNetworkAPI
AdminAssets
Common
SocialNetworkContent

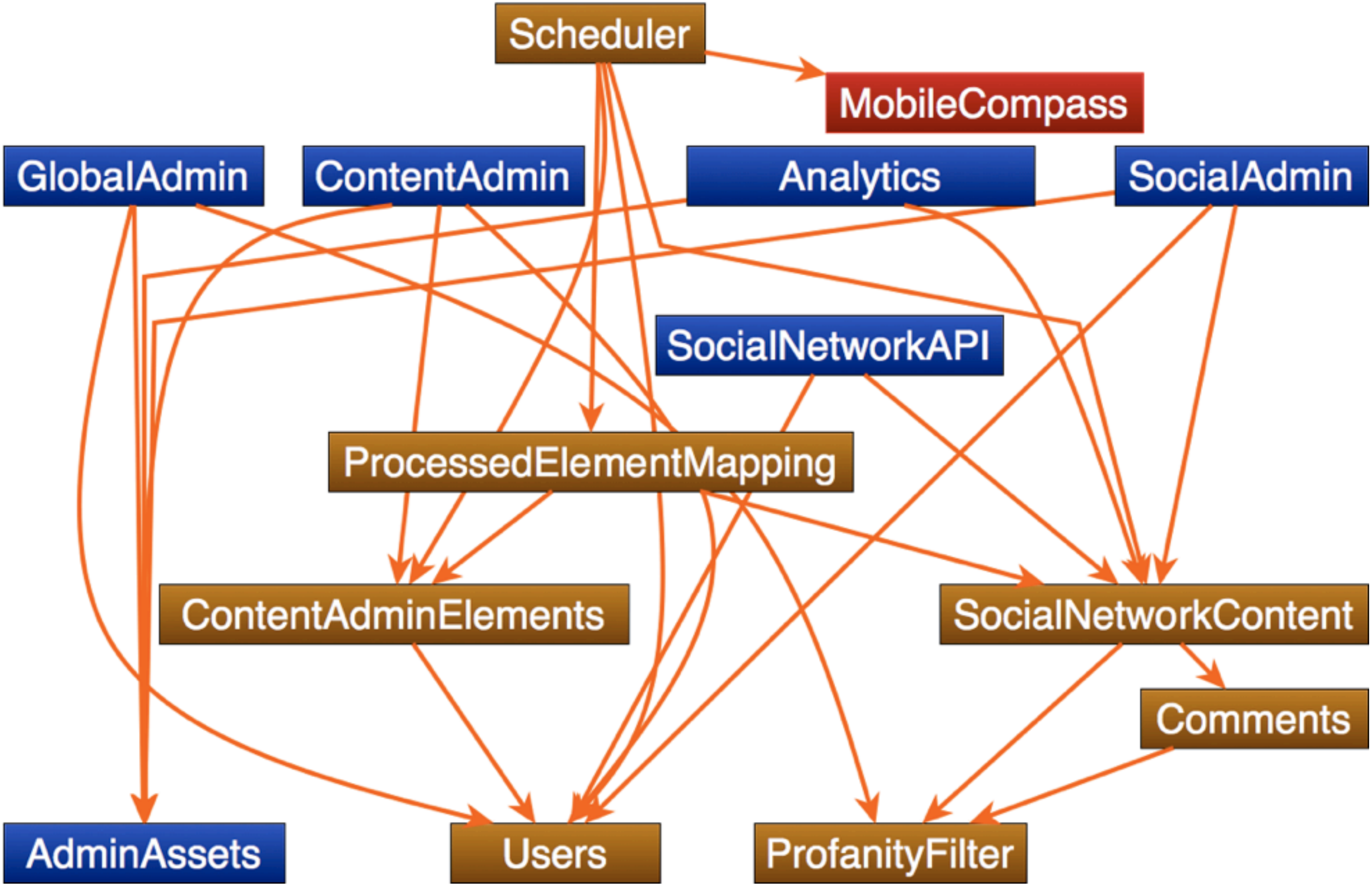

Scheduler
MobileCompass
GlobalAdmin
ContentAdmin
SocialAdmin
SocialNetworkAPI
ContentAdminElements
SocialNetworkContent
AdminAssets
Common

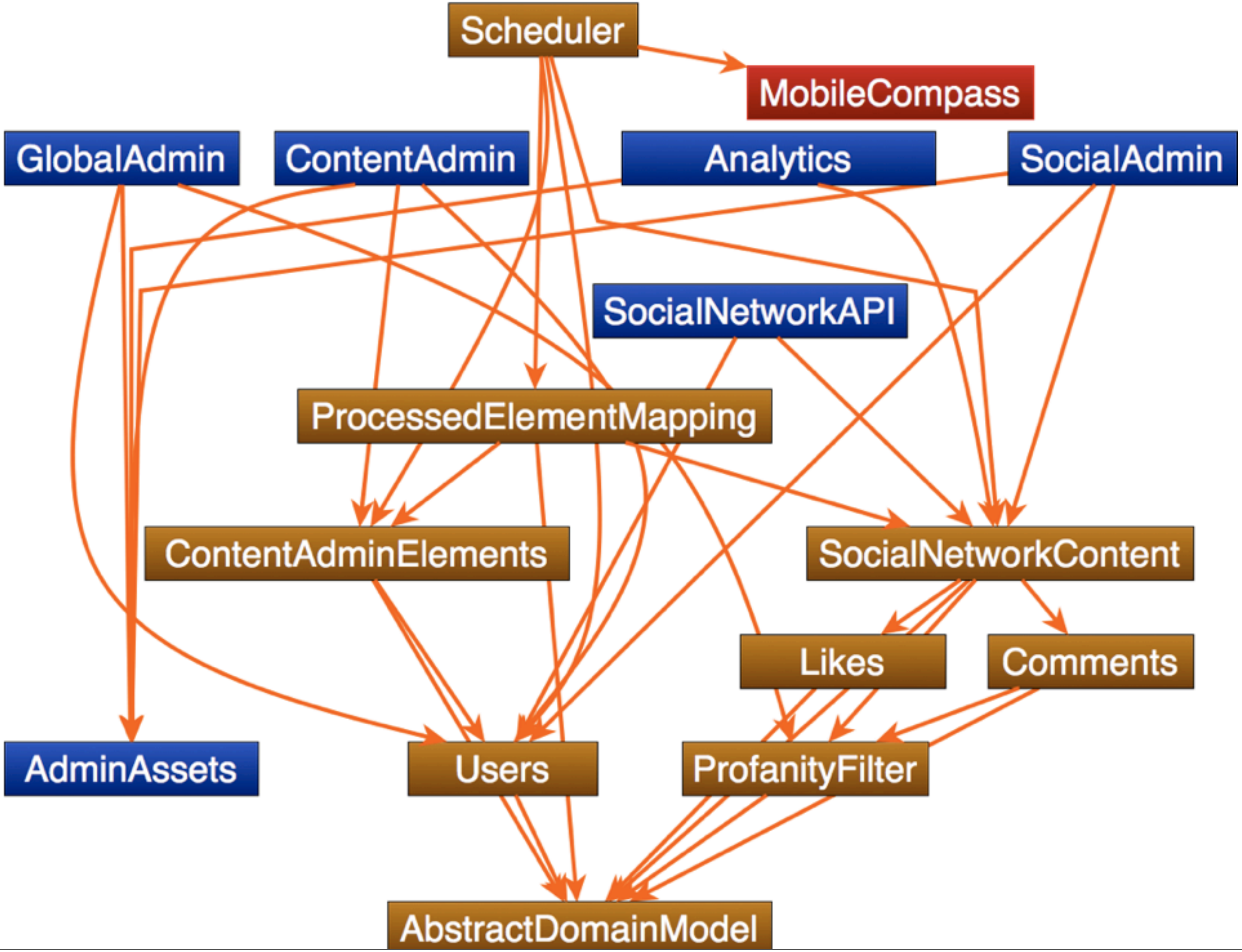


Scheduler
MobileCompass
GlobalAdmin
ContentAdmin
SocialAdmin
SocialNetworkAPI
ContentAdminElements
SocialNetworkContent
AdminAssets
Common


Scheduler
MobileCompass
GlobalAdmin
ContentAdmin
SocialAdmin
SocialNetworkAPI
ContentAdminElements
SocialNetworkContent
AdminAssets
Users


Scheduler
MobileCompass
GlobalAdmin
ContentAdmin
SocialAdmin
SocialNetworkAPI
ContentAdminElements
SocialNetworkContent
AdminAssets
Users
ProfanityFilter


Scheduler
MobileCompass
GlobalAdmin
ContentAdmin
Analytics
SocialAdmin
SocialNetworkAPI
ProcessedElementMapping
ContentAdminElements
SocialNetworkContent
Comments
AdminAssets
Users
ProfanityFilter


Scheduler
MobileCompass
GlobalAdmin
ContentAdmin
Analytics
SocialAdmin
SocialNetworkAPI
ProcessedElementMapping
ContentAdminElements
SocialNetworkContent
Likes
Comments
AdminAssets
Users
ProfanityFilter
AbstractDomainModel

how to get started

```
rails plugin new engine_name --mountable
```

```
gem "engine_name",
    path: "path/to/engine_name"
```

```
mount EngineName::Engine => "/engine_name",
:as => "engine_name"
```

```
rails plugin new engine_name --mountable
rails plugin new engine_template --mountable
```

```
s/engine_template/new_engine_name/g
s/EngineTemplate/NewEngineName/g

for file in $(find . -name "*engine_template*")
do
  mv $file `echo $file | sed s/engine_template/$1/`
done
```

```
bundle gem gem_name
```

```
gem "gem_name",
    path: "path/to/gem_name"
```

```
bundle gem gem_template

s/gem_template/new_gem_name/g
s/GemTemplate/NewGemName/g

for file in $(find . -name "*gem_template*")
do
  mv $file `echo $file | sed s/gem_template/$1/`
done
```

got it?

engines the right way

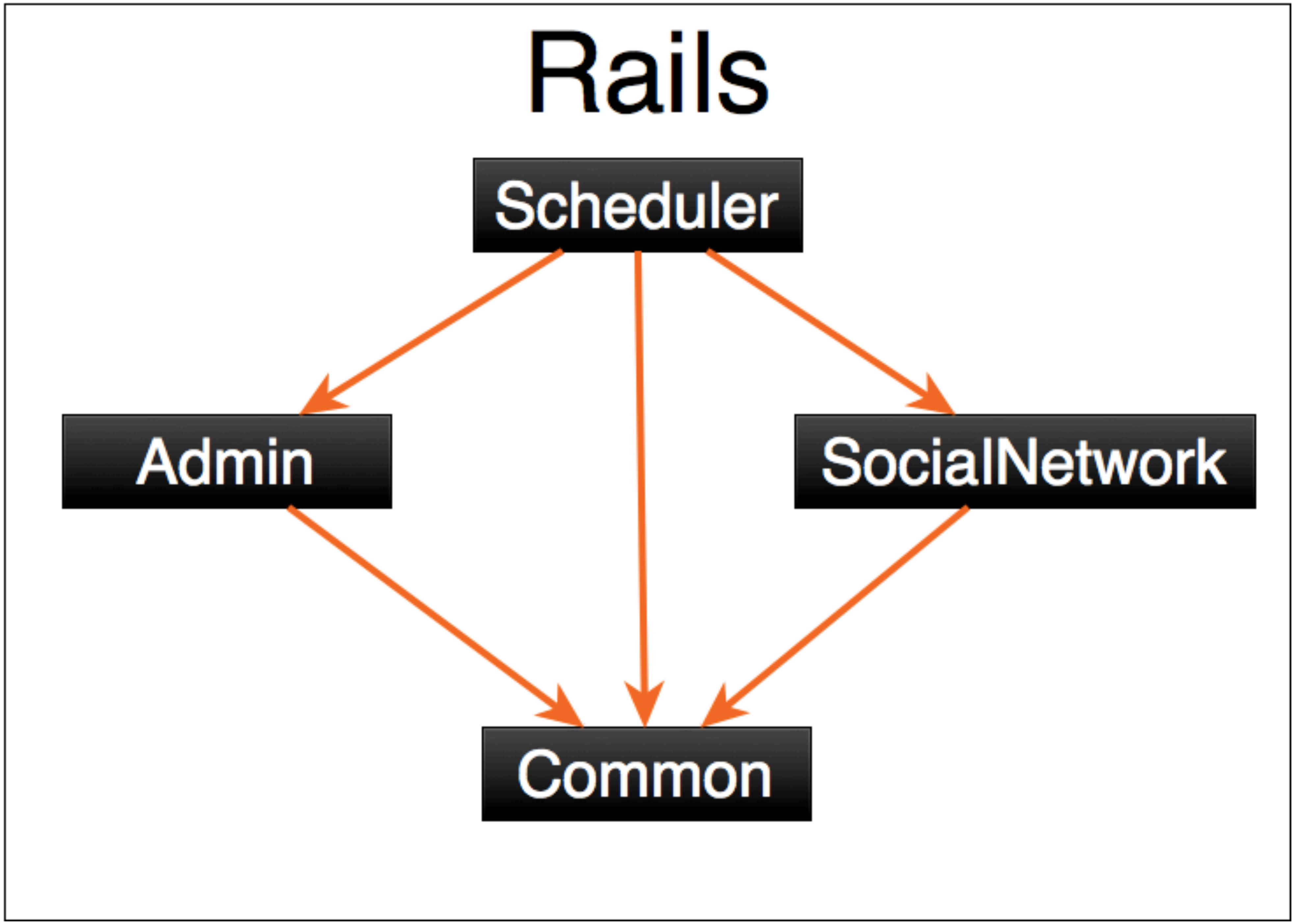
fully contained engines

- all tables namespaced
- migrations
- tests

table namespacing

@benjamin_smith

PIVOTAL LABS



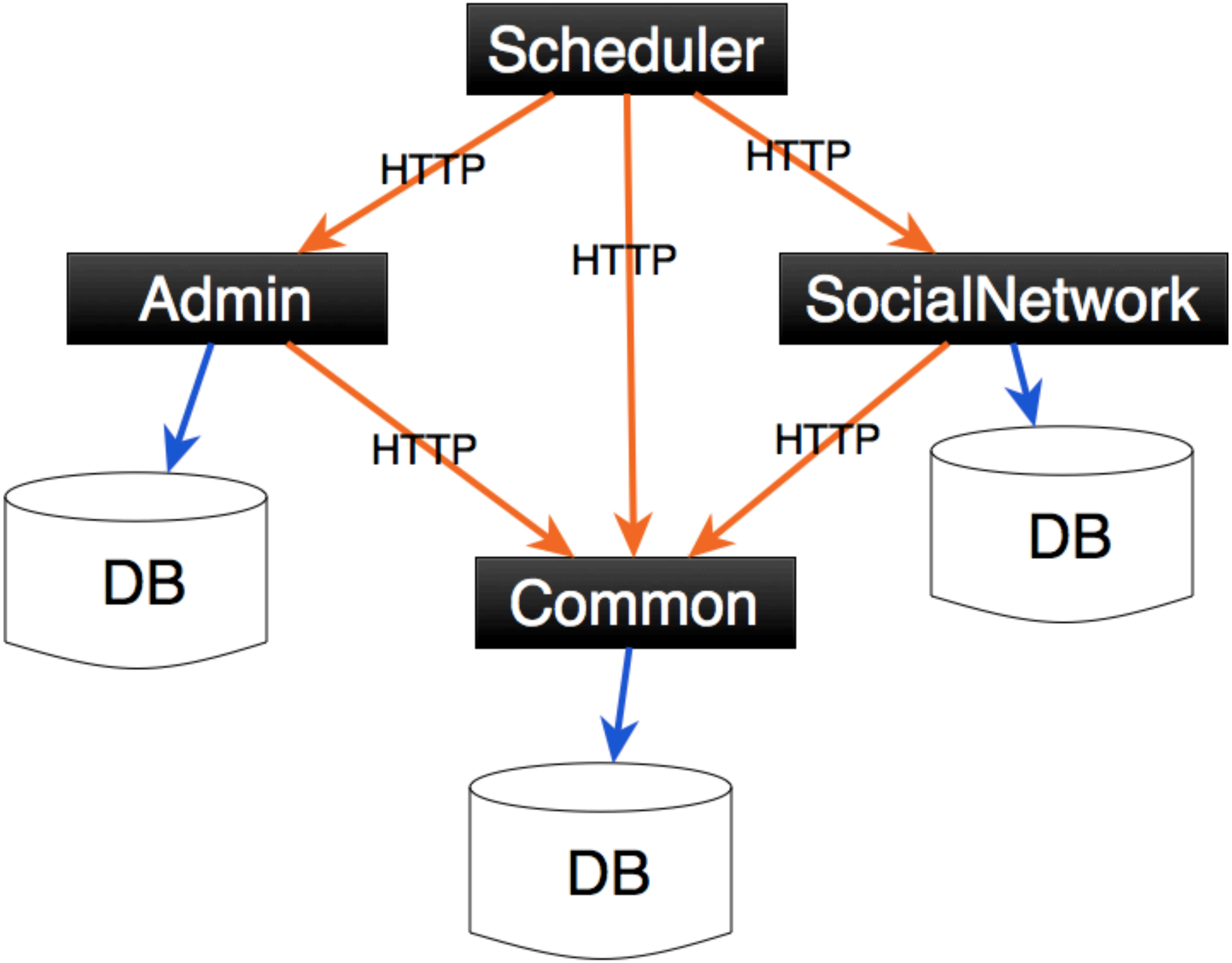
Rails

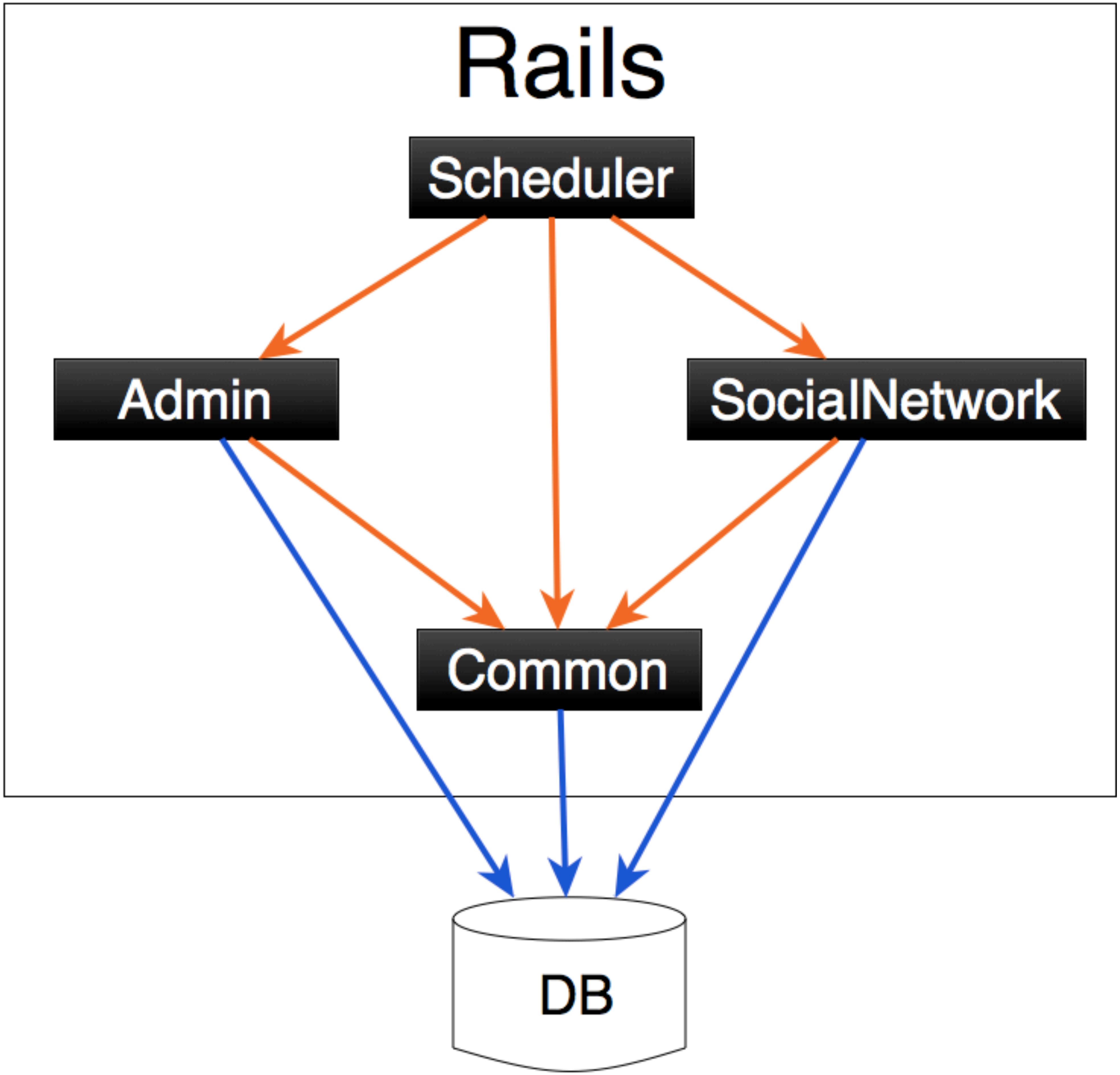
Scheduler

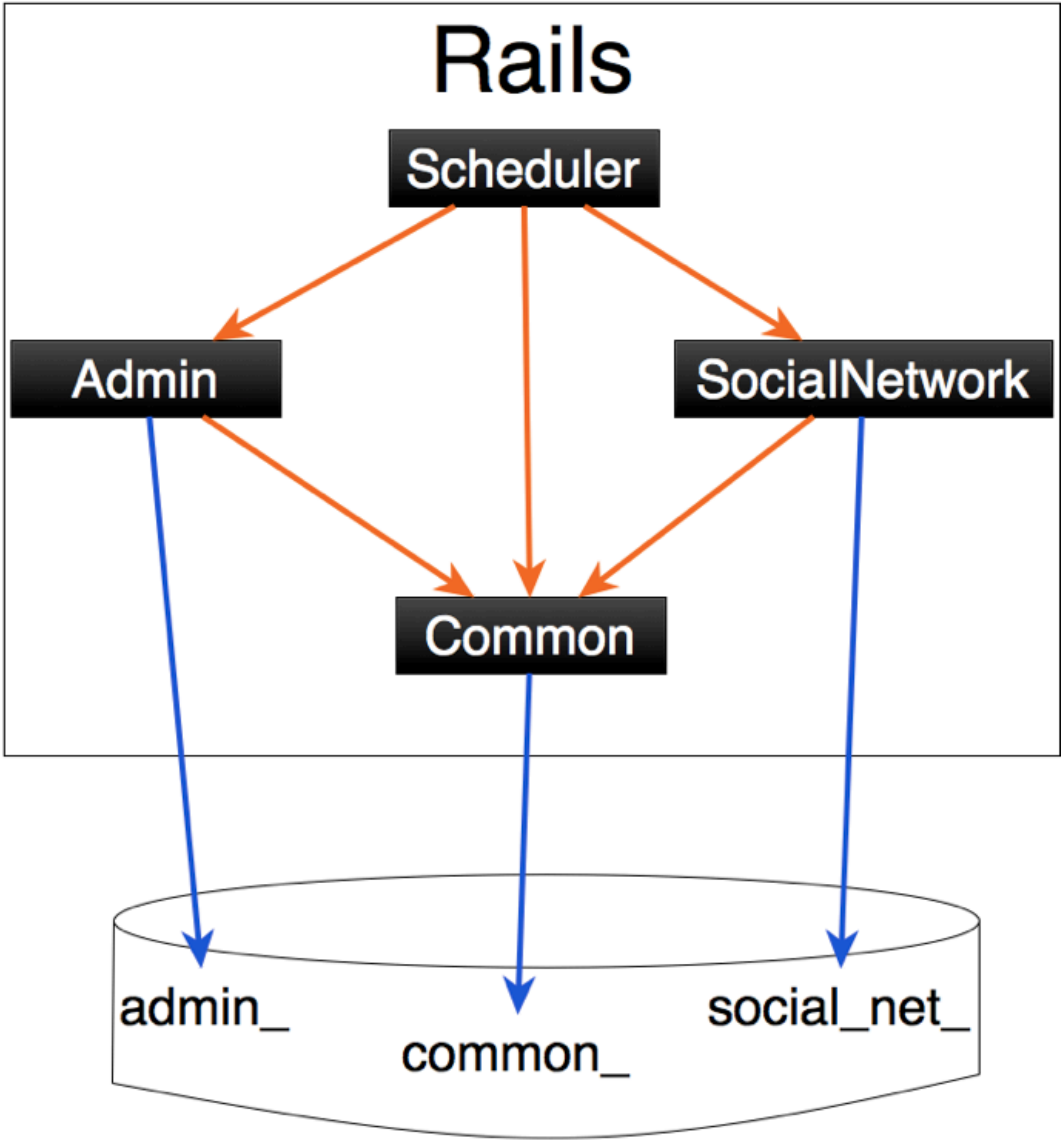
Admin

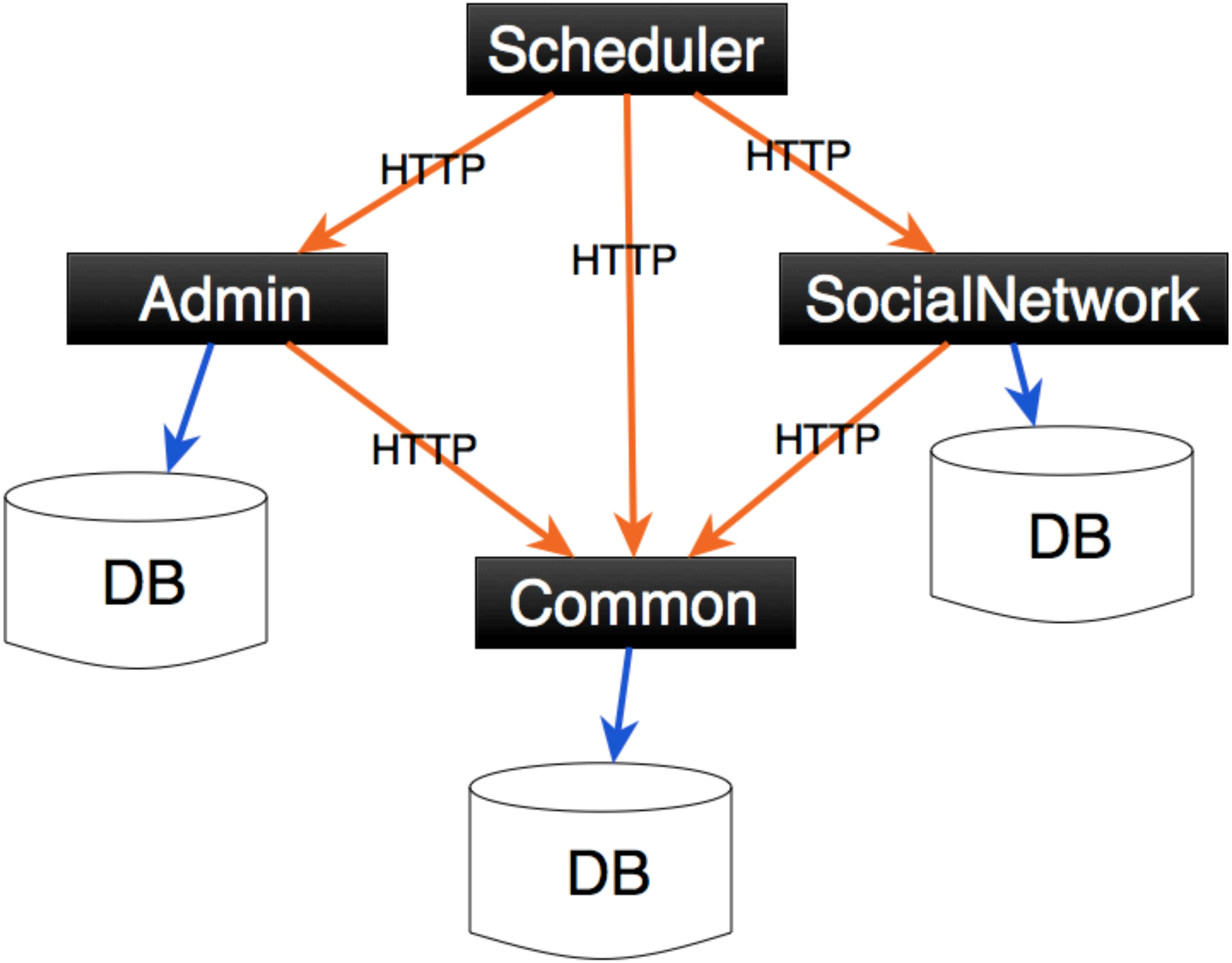
SocialNetwork

Common


Scheduler
HTTP
HTTP
HTTP
Admin
SocialNetwork
HTTP
HTTP
DB
DB
Common
DB


Rails
Scheduler
Admin
SocialNetwork
Common
DB


Rails
Scheduler
Admin
SocialNetwork
Common
admin_
common_
social_net_


Scheduler
HTTP
HTTP
HTTP
Admin
SocialNetwork
HTTP
HTTP
DB
DB
Common
DB

migrations inside engines

@benjamin_smith

PIVOTAL LABS

```
rake your_engine_name:install:migrations
```

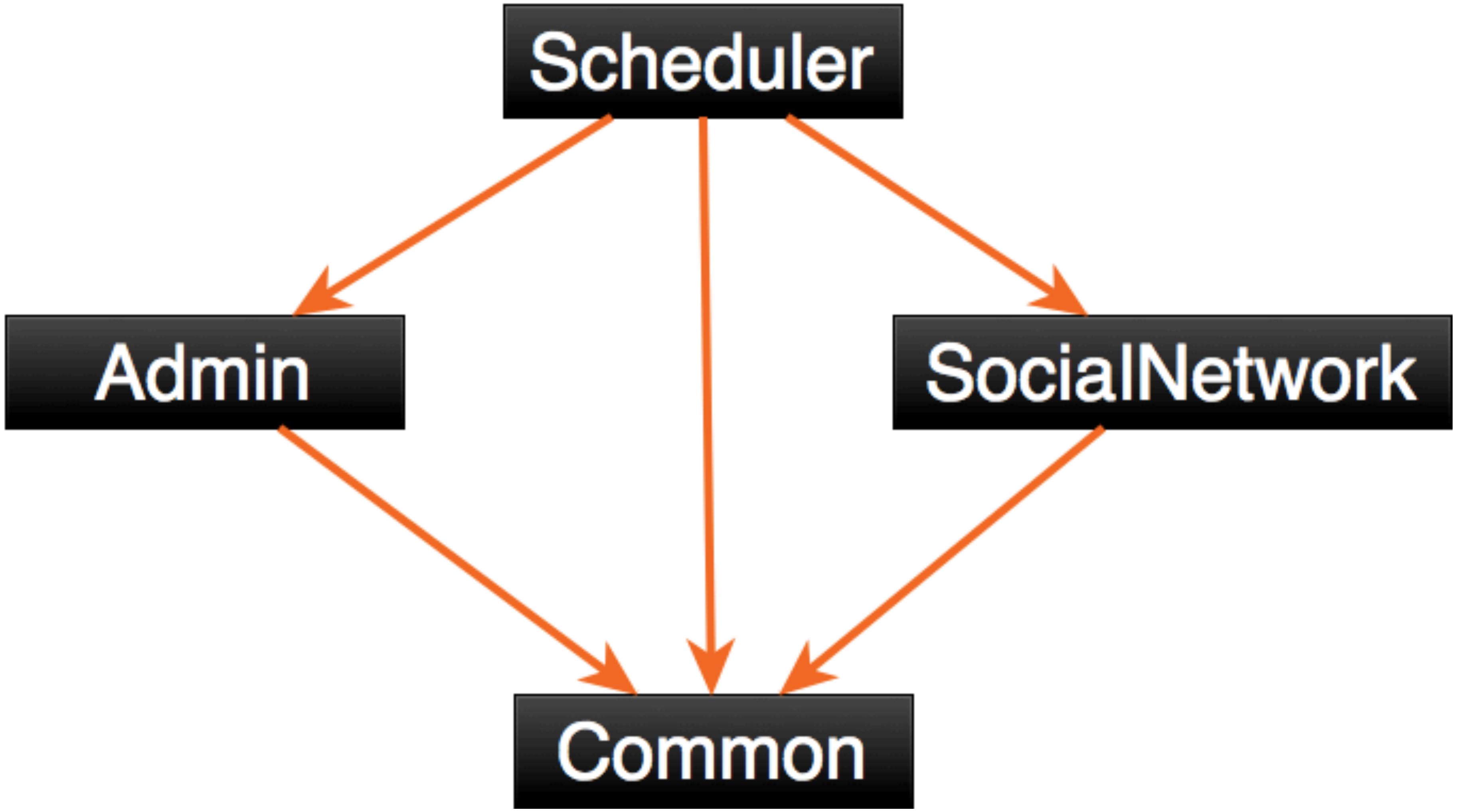
```
module EngineWithMigrations
  class Engine < ::Rails::Engine
    isolate_namespace EngineWithMigrations

    initializer :append_migrations do |app|
      unless app.root.to_s.match root.to_s
        app.config.paths["db/migrate"] +=
                  config.paths["db/migrate"].expanded
      end
    end
  end
end
```

http://pivotallabs.com/leave-your-migrations-in-your-rails-engines/

one table per migration

http://pivotallabs.com/moving-db-tables-between-rails-engines/

testing the engines in isolation



Scheduler

Admin

SocialNetwork

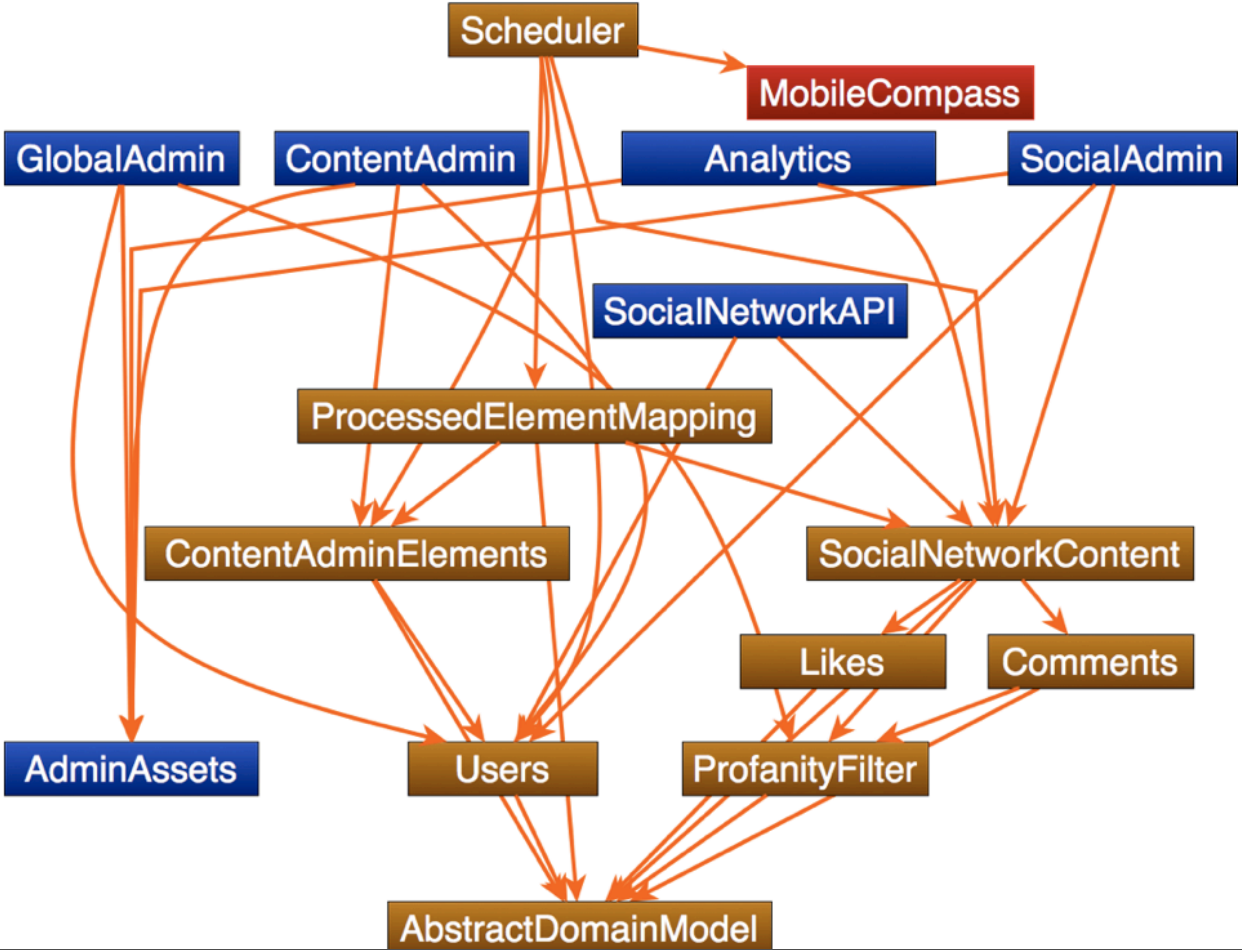
Common

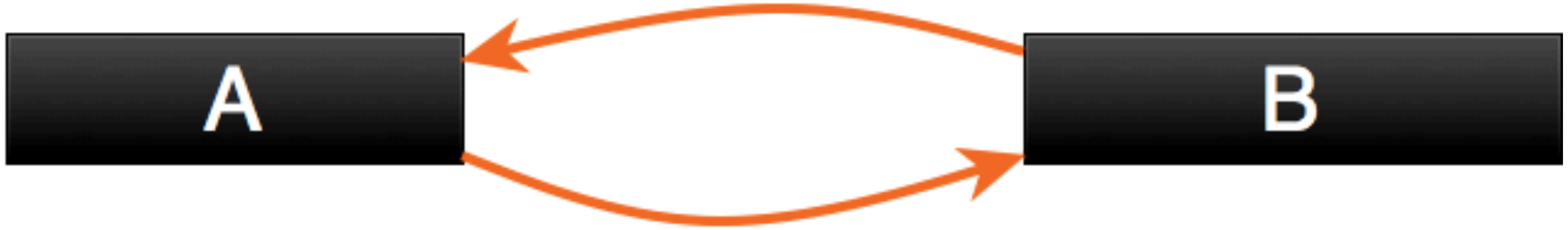
engines the right way

fully contained engines

- all tables namespaced
- migrations
- tests

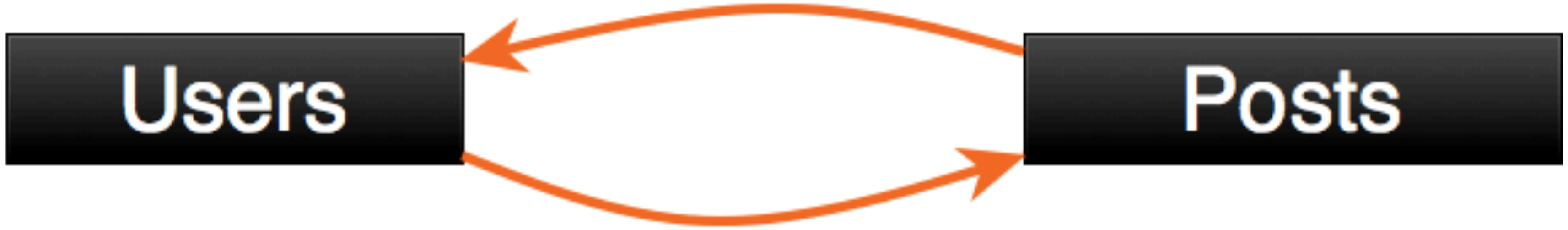
no circular dependencies


Scheduler
MobileCompass
GlobalAdmin
ContentAdmin
Analytics
SocialAdmin
SocialNetworkAPI
ProcessedElementMapping
ContentAdminElements
SocialNetworkContent
Likes
Comments
AdminAssets
Users
ProfanityFilter
AbstractDomainModel


A
B

A&B

```
class User < ActiveRecord::Base
  has_many :posts
end


class Post < ActiveRecord::Base
  belongs_to :user
end
```



Users

Posts

domain api

domain api

- simple classes to wrap ActiveRecord calls
  - take as input: params or ids
  - output PORO

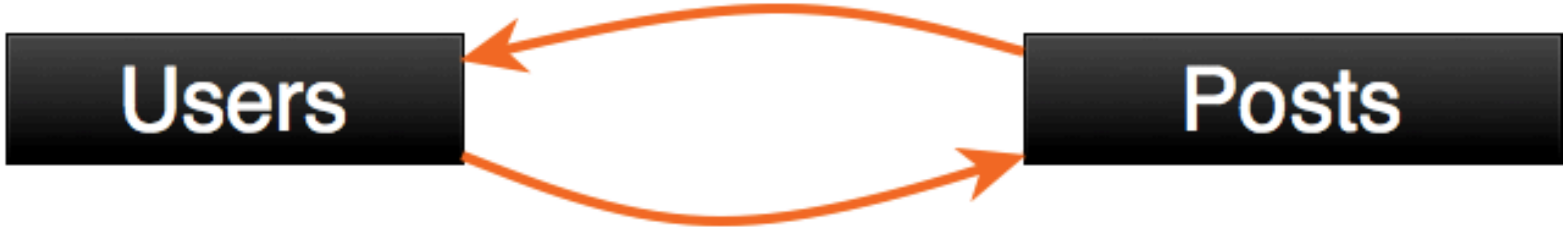
```
class UserManager
  def find_by_id(id)
    User.new(UserRecord.find(id))
  end

  private
  class UserRecord < ActiveRecord::Base
    validates :email, presence: true
  end
end
```

```
class UserManager
  def find_by_id(id)
    User.new(UserRecord.find(id))
  end

  private
  class UserRecord < ActiveRecord::Base
    validates :email, presence: true
  end
end
```

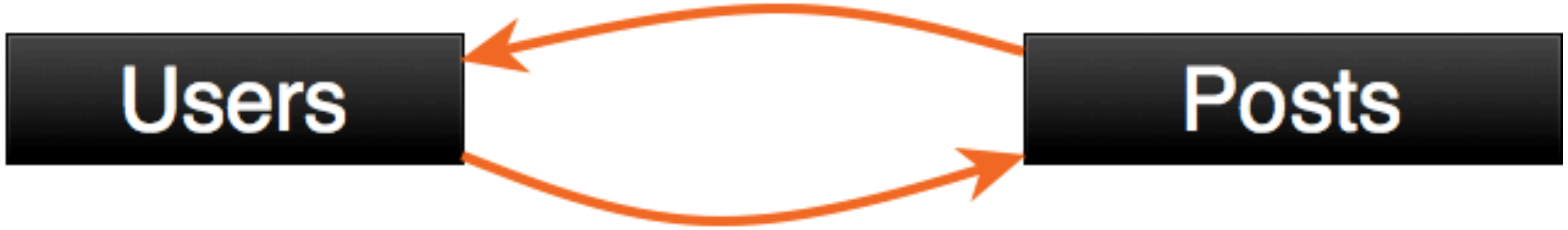
```
class UserManager
  def find_by_id(id)
    User.new(UserRecord.find(id))
  end

  private
  class UserRecord < ActiveRecord::Base
    validates :email, presence: true
  end
end
```

```
class UserManager
  def find_by_id(id)
    User.new(UserRecord.find(id))
  end

  private
  class UserRecord < ActiveRecord::Base
    validates :email, presence: true
  end
end
```

```
class User
  def initialize(user_record)
    self.email = user_record.email
  end
end
```



Users
Posts



Users

Posts

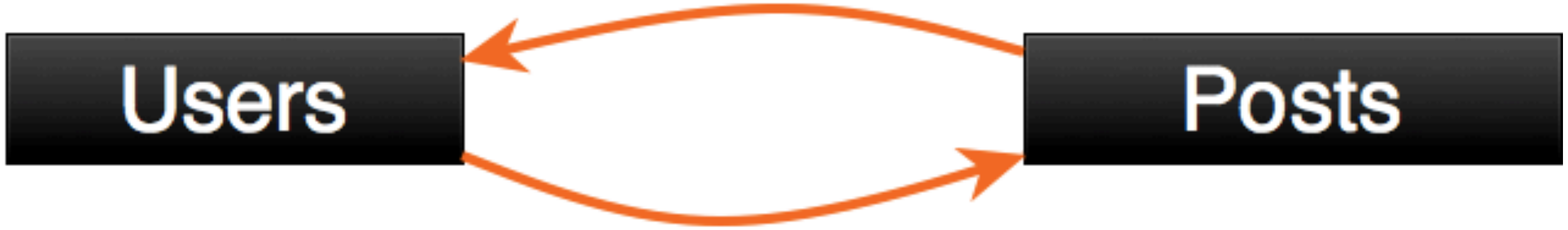
@user.posts

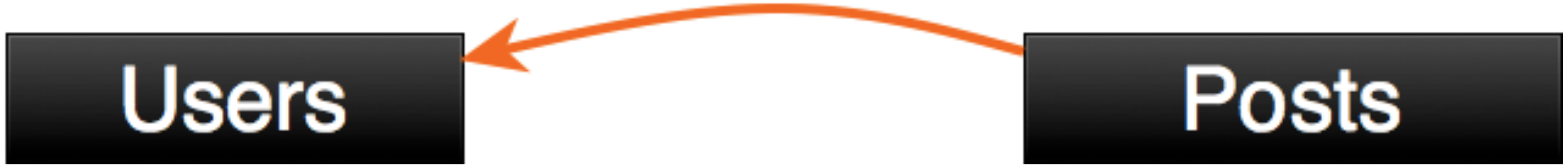
@post.user

```
class PostManager
  def find_all_by_user_id(user_id)
    PostRecord.where(user_id: user_id)

    # ... convert to PORO ...
  end
end
```

```
class PostManager
  def find_all_by_user_id(user_id)
    PostRecord.where(user_id: user_id)

    # ... convert to PORO ...
  end
end
```

```
post_manager = PostManager.new
@posts = post_manager.find_all_by_user_id(@user.id)
```


Users
Posts



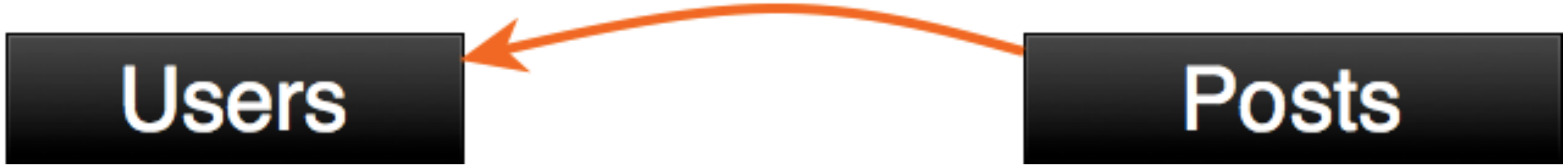
Users

Posts

```
class UserManager
  def find_by_id(id)
    User.new(UserRecord.find(id))
  end
end
```

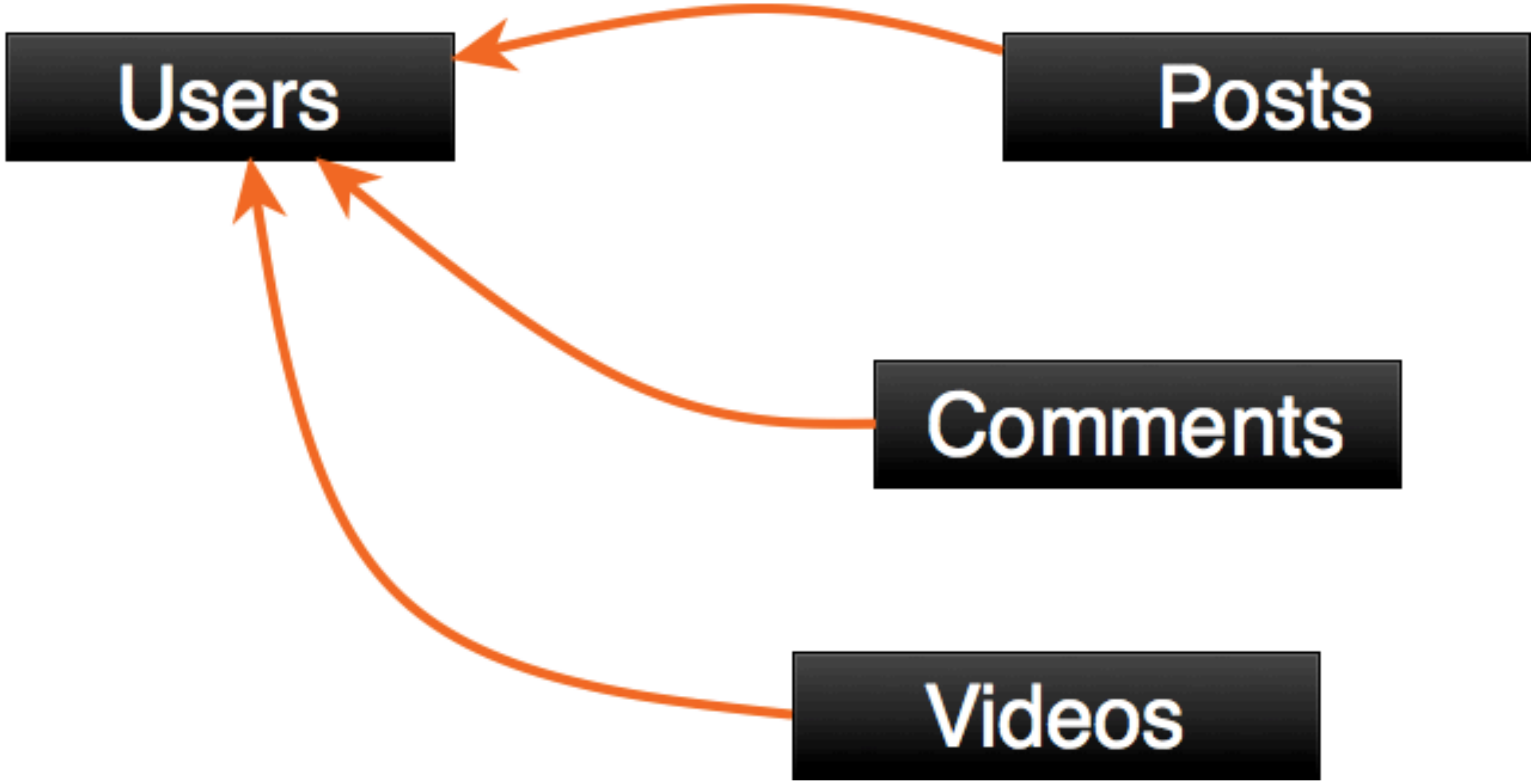
```
class UserManager
  def find_by_id(id)
    User.new(UserRecord.find(id))
  end
end
```

```
user_manager = UserManager.new
author = user_manager.find_by_id(@post.user_id)
```



Users

Posts

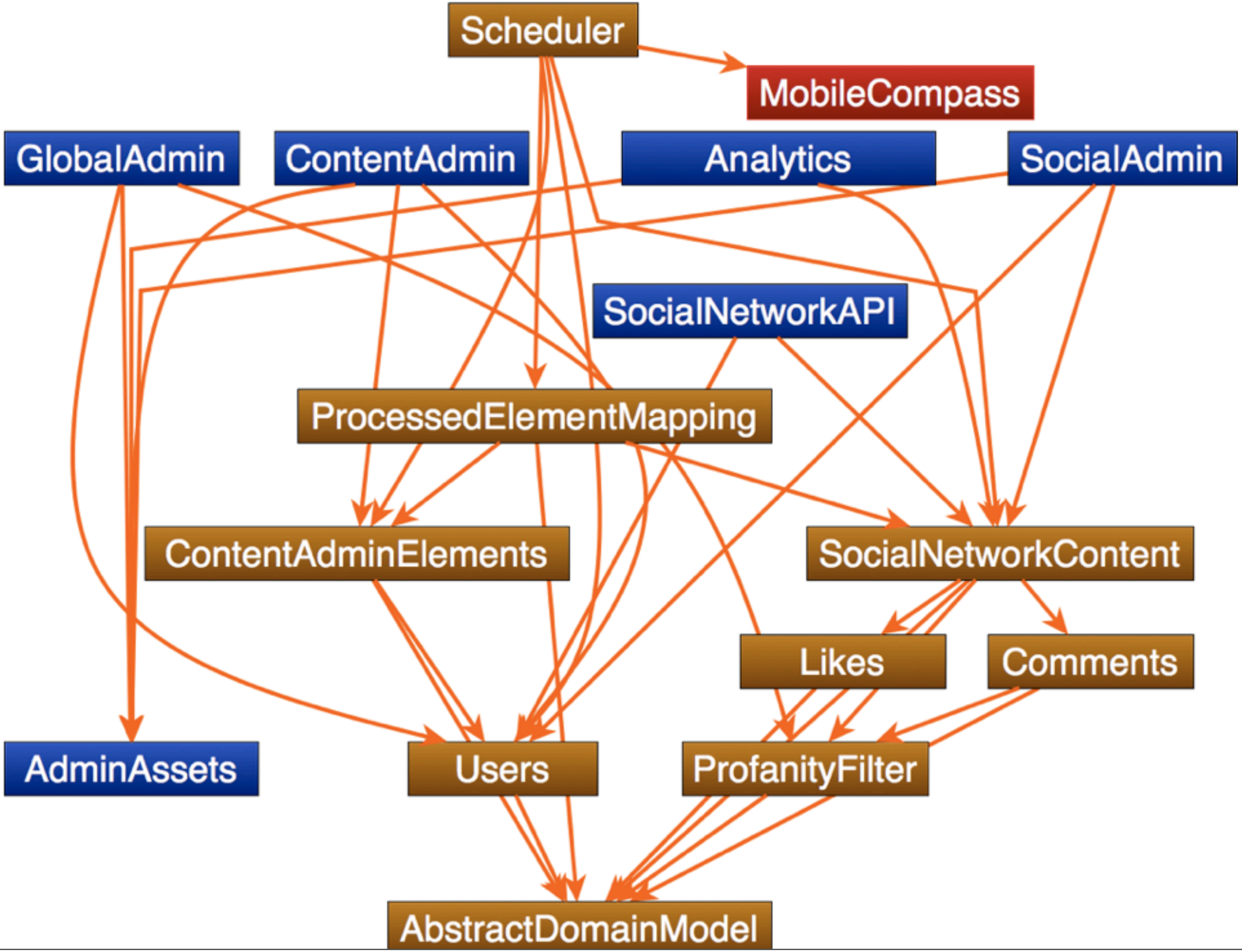


Users
Posts
Comments
Videos

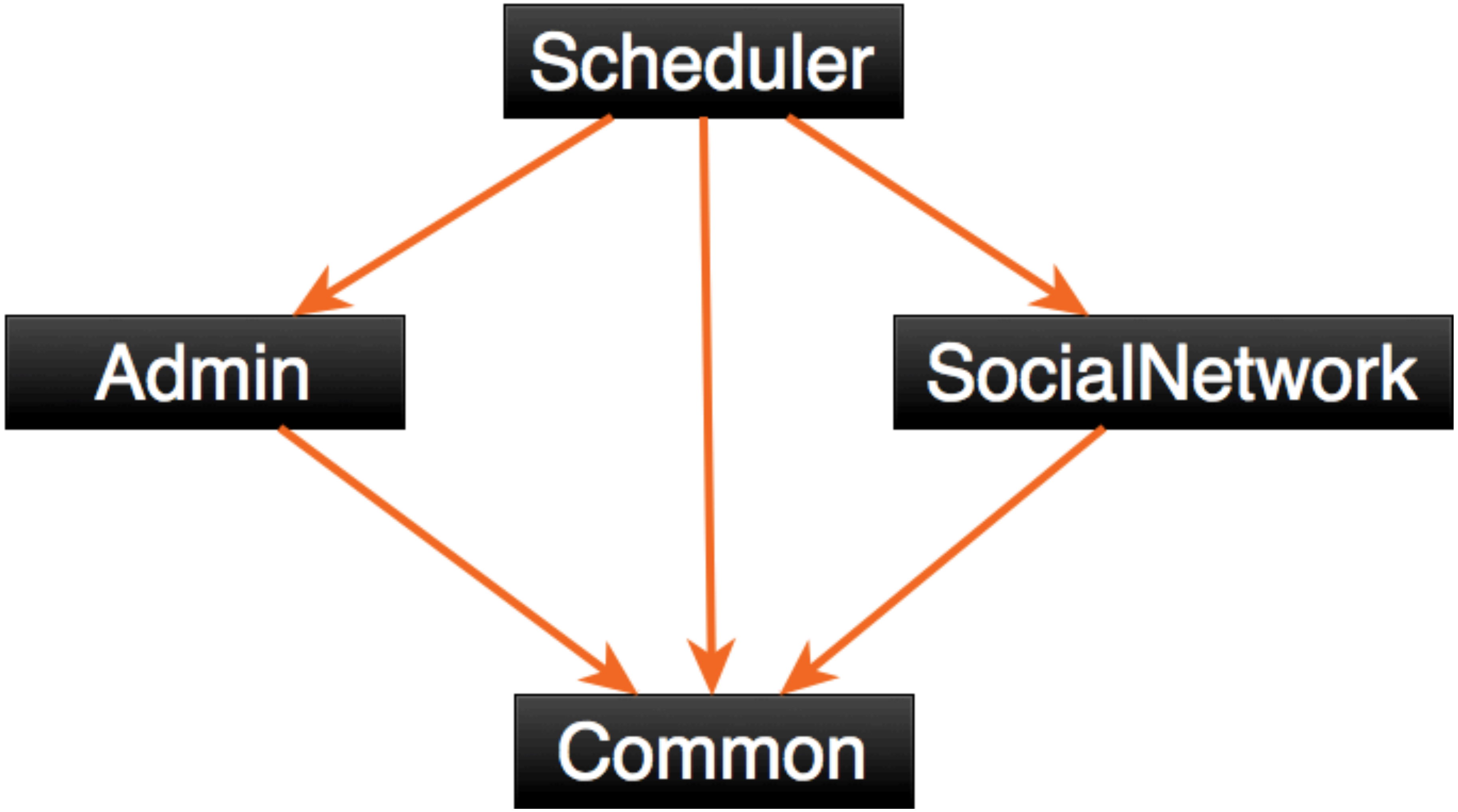
engines the right way

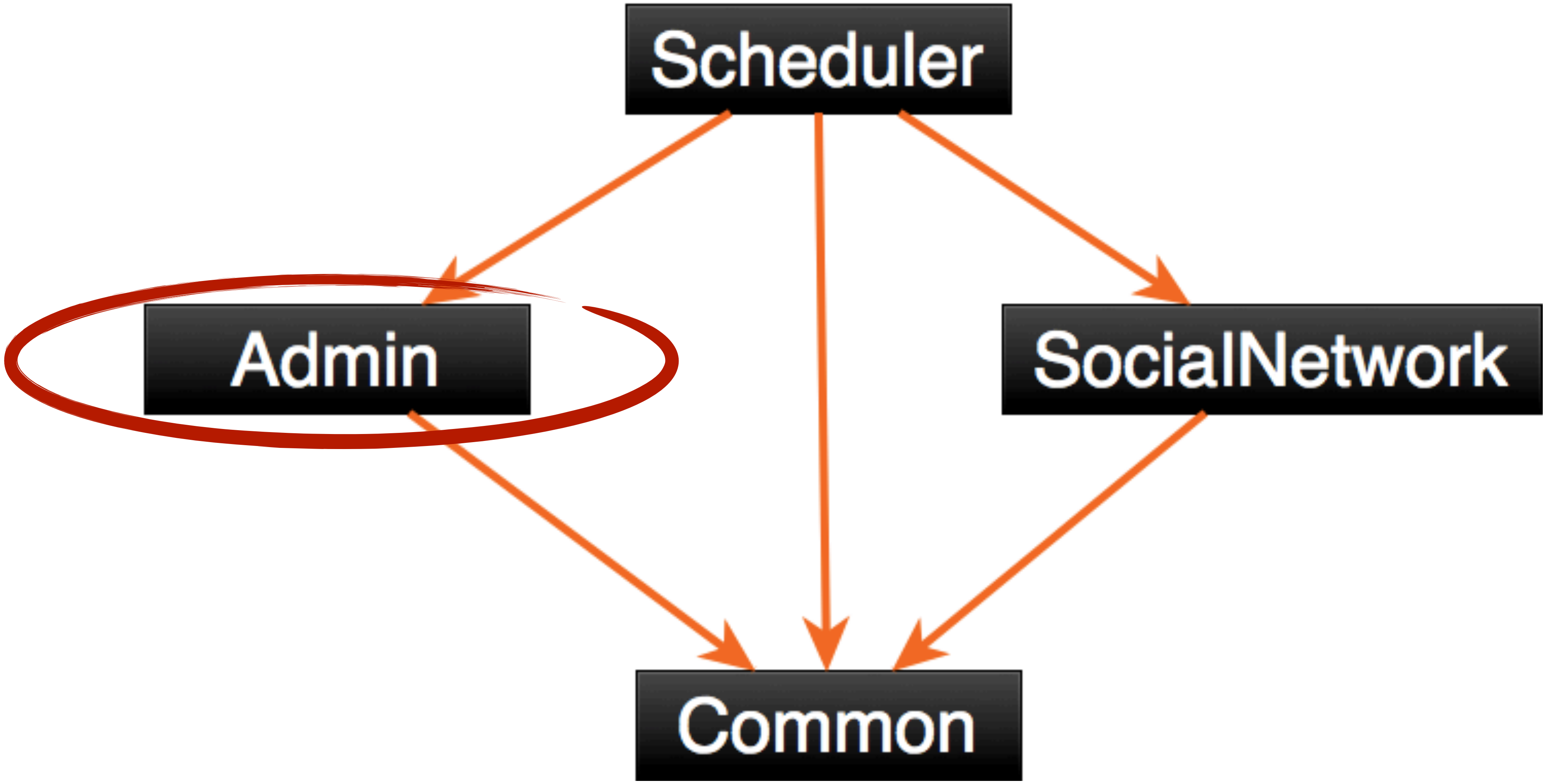
no circular dependencies

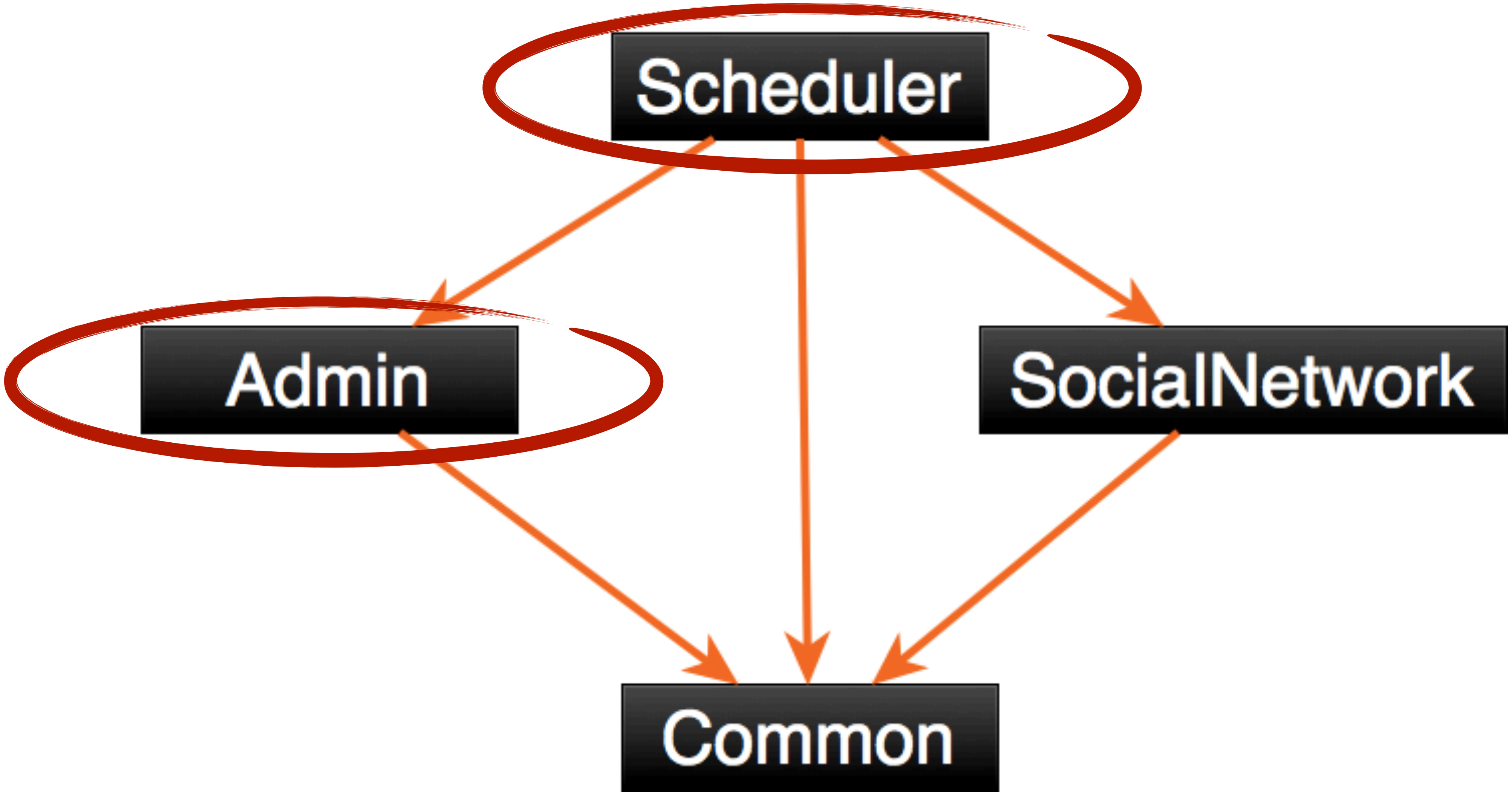
started simple


Scheduler
MobileCompass
GlobalAdmin
ContentAdmin
Analytics
SocialAdmin
SocialNetworkAPI
ProcessedElementMapping
ContentAdminElements
SocialNetworkContent
Likes
Comments
AdminAssets
Users
ProfanityFilter
AbstractDomainModel

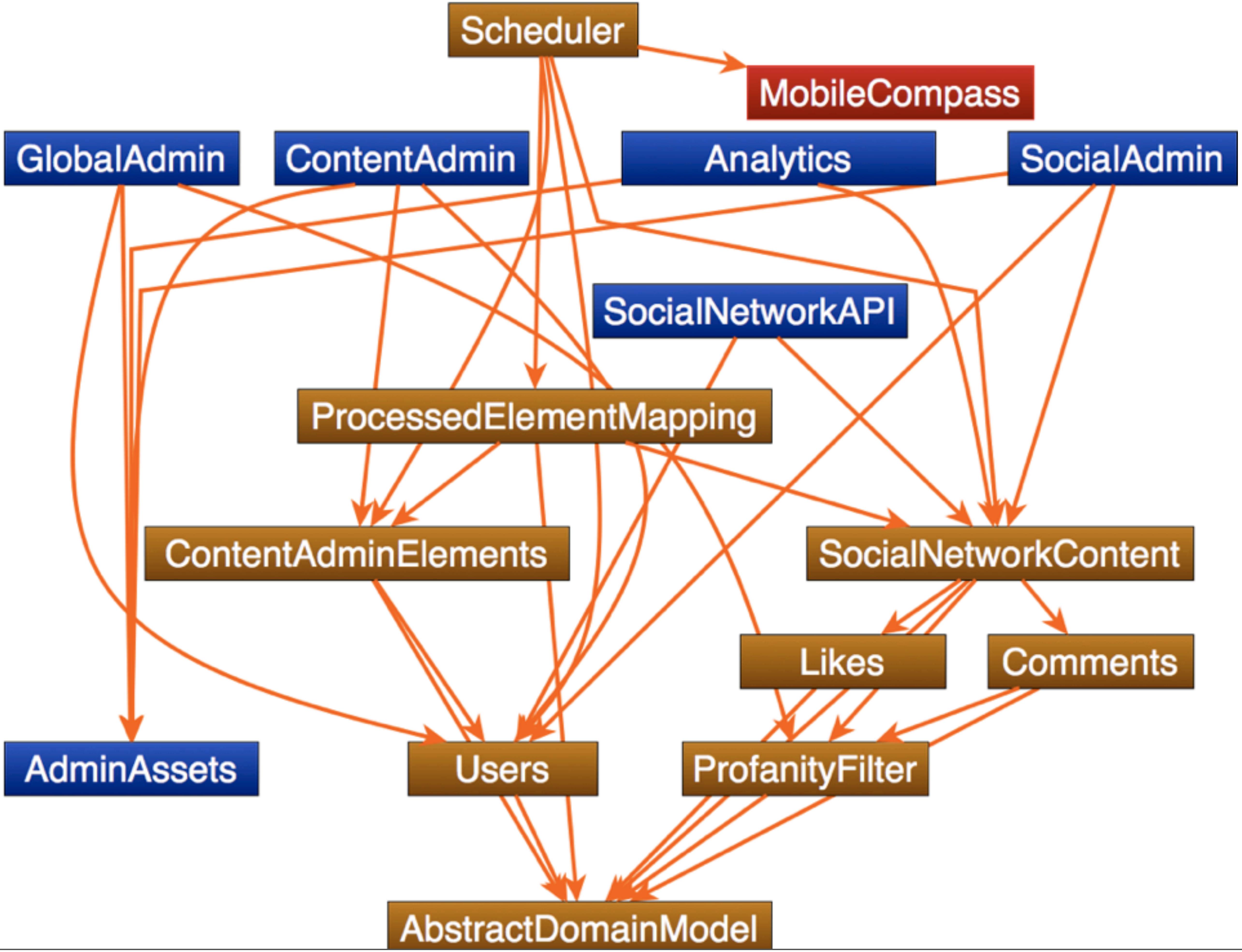
I live my life one green build at a time


Scheduler
Admin
SocialNetwork
Common


Scheduler
Admin
SocialNetwork
Common


Scheduler
Admin
SocialNetwork
Common

git whatchanged


Scheduler
MobileCompass
GlobalAdmin
ContentAdmin
Analytics
SocialAdmin
SocialNetworkAPI
ProcessedElementMapping
ContentAdminElements
SocialNetworkContent
Likes
Comments
AdminAssets
Users
ProfanityFilter
AbstractDomainModel

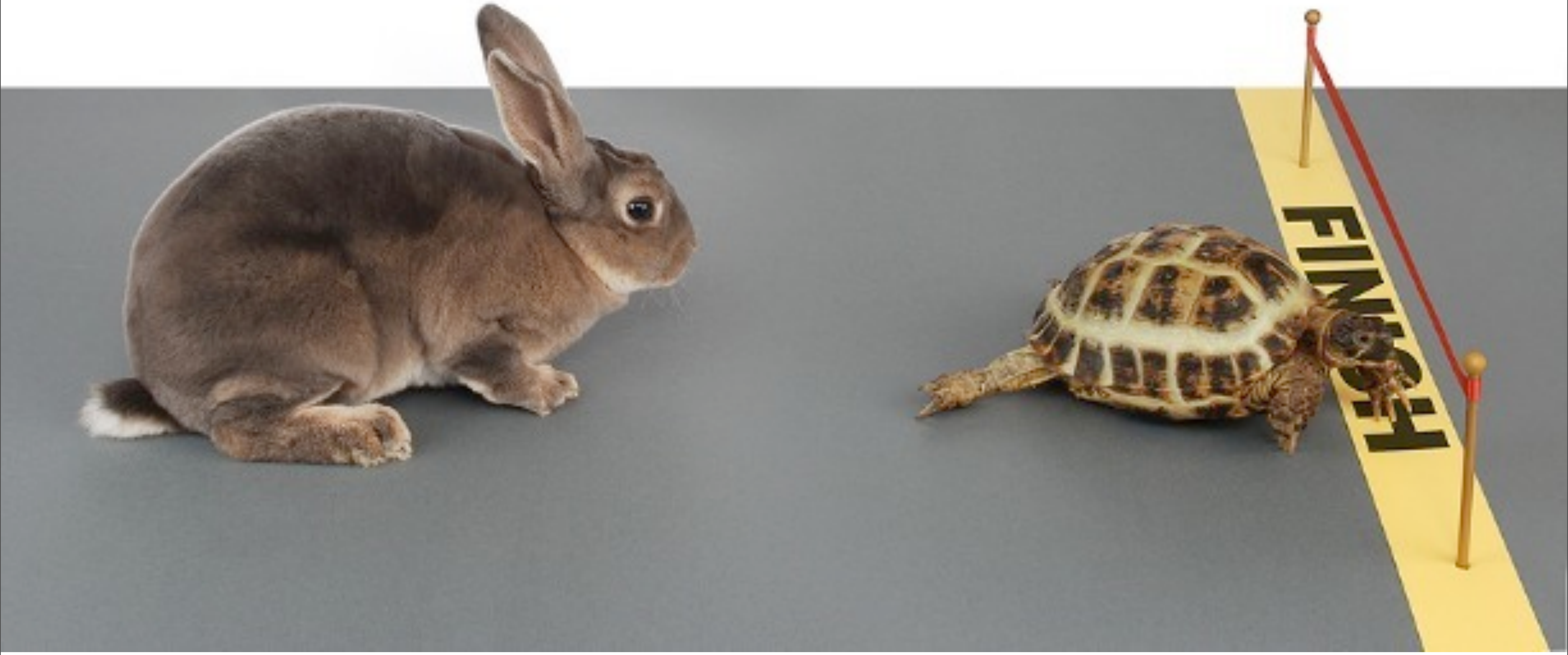
how did it work out?

engines are a pain...

- LOW velocity for weeks (4-6)
    - client worries
    - opposite of Rails
    - one pair to start is a MUST
- some technical hurtles
- monkey patching Rails

@benjamin_smith

PIVOTAL LABS

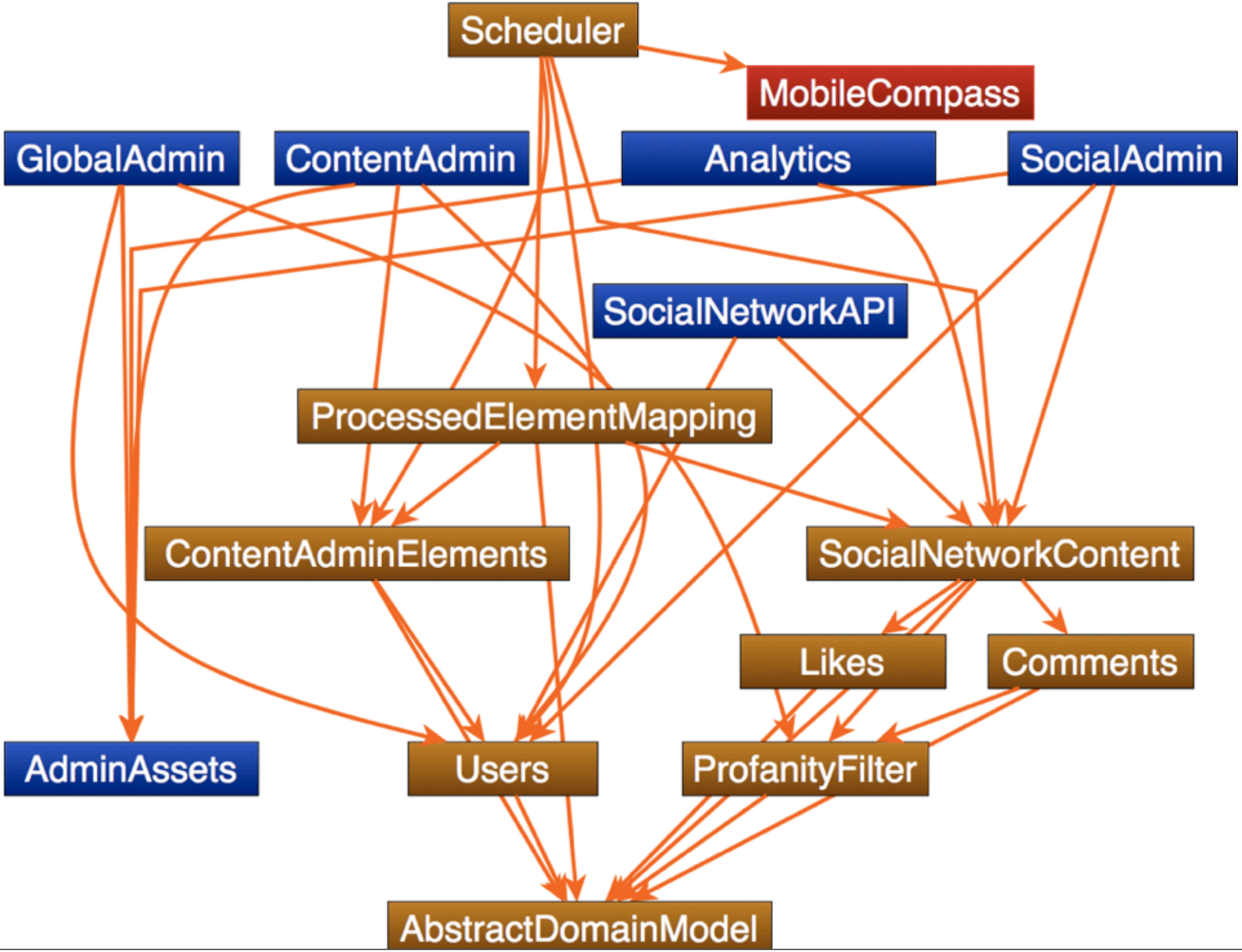

FINISH

... but they get better

- no slow down in development time
- they age well
- easier parallel development
- potential for scaling
- smart and fast build scripts

@benjamin_smith

thumbs up



Scheduler
MobileCompass
GlobalAdmin
ContentAdmin
Analytics
SocialAdmin
SocialNetworkAPI
ProcessedElementMapping
ContentAdminElements
SocialNetworkContent
Likes
Comments
AdminAssets
Users
ProfanityFilter
AbstractDomainModel

```ruby
class UserManager
  def find_by_id(id)
    User.new(UserRecord.find(id))
  end

  private
  class UserRecord < ActiveRecord::Base
    validates :email, presence: true
  end
end
```

thanks

questions?

http://pivotallabs.com/tag/rails-application-suites/

@benjamin_smith