

Hot Topics: The DuraSpace Community Webinar Series

Series Three:

“Get a Head on Your Repository with
Hydra End-to-End Solutions”

Curated by Tom Cramer

Webinar 3: Hydra Technical Deep Dive

Presented by:
Matt Zumwalt,
Technical Lead, Hydra Project,
MediaShelf



Technical Framework

The Hydra has hatched & put on some muscle. What does it look like?

Matt Zumwalt
DuraSpace Webinar
30 October 2012





The Problem

long lived, complex content

the data we create extend beyond and
outlast the software used to create them

“the *open data* movement is more important
than the *open source* movement”

- Ian Davis
Code4Lib 2009

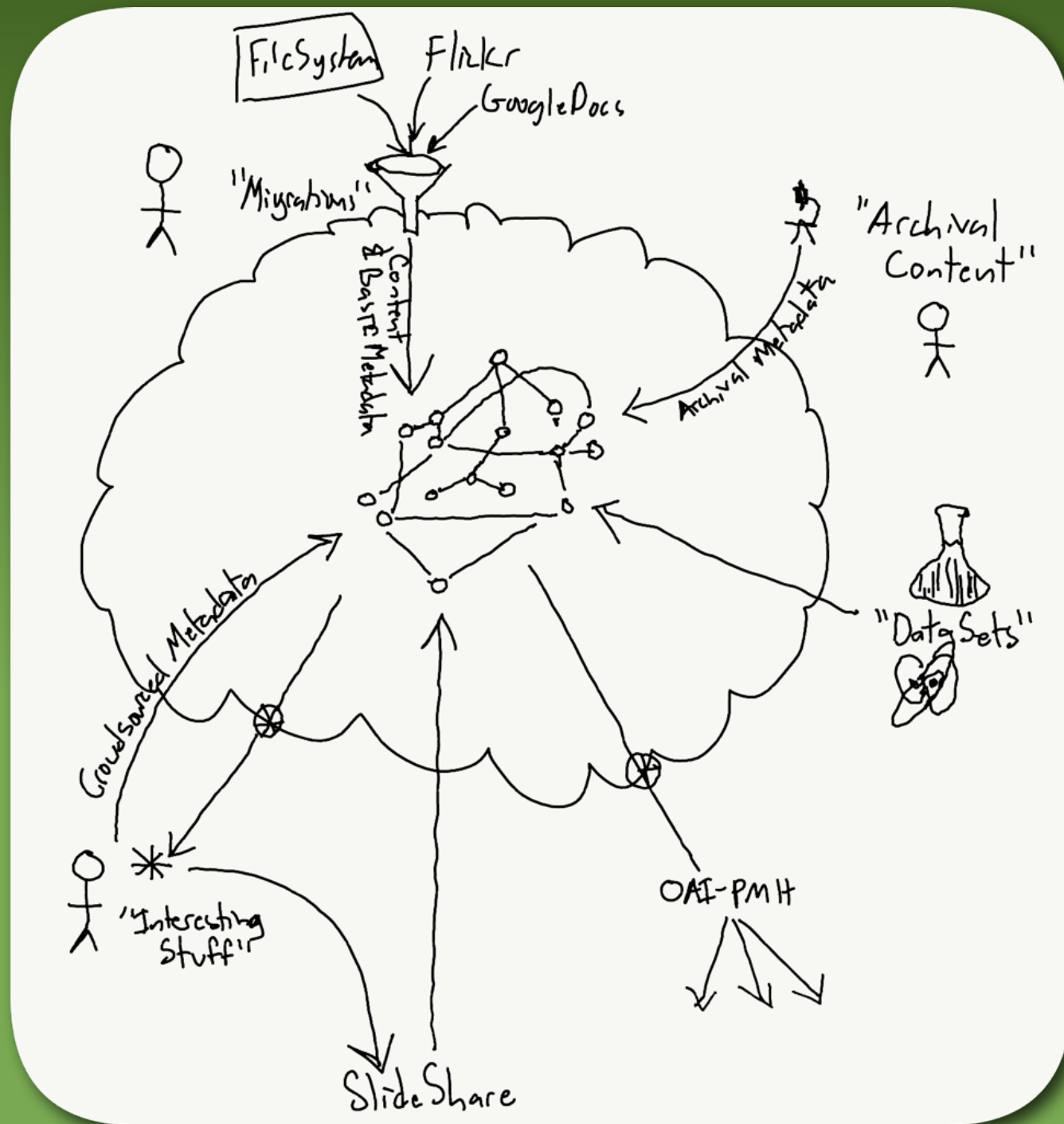
the funny problem

Our vision is to create complex, evolving, long-lived content. Meanwhile, each set of users wants to see & interact with that content in different ways relative to temporal and intellectual context.

content in open, complex,
long-lived context

users in task-specific,
time-specific context

many lightweight views onto complex content





The Solution: Framework & Community

software & community aimed at
supporting **teams** who are
creating & sustaining repository
solutions

Framework for Creating

hydra heads for creation & management
of repository content

search interfaces

scripts / bots to operate on repository
contents

Pragmatic Development

becoming **better developers**

becoming **better teams**

sharing code

supporting software

Robust Collaboration

publicly visible code

continuous integration

test coverage

release cycle & release managers

developer documentation



The Basic Structure

CRUD

Create

Retrieve

Update

Delete

CRUD

Create/Submit

Modify

Delete

Index

CRUD

Search

Browse

View Details

CRUD in Rails

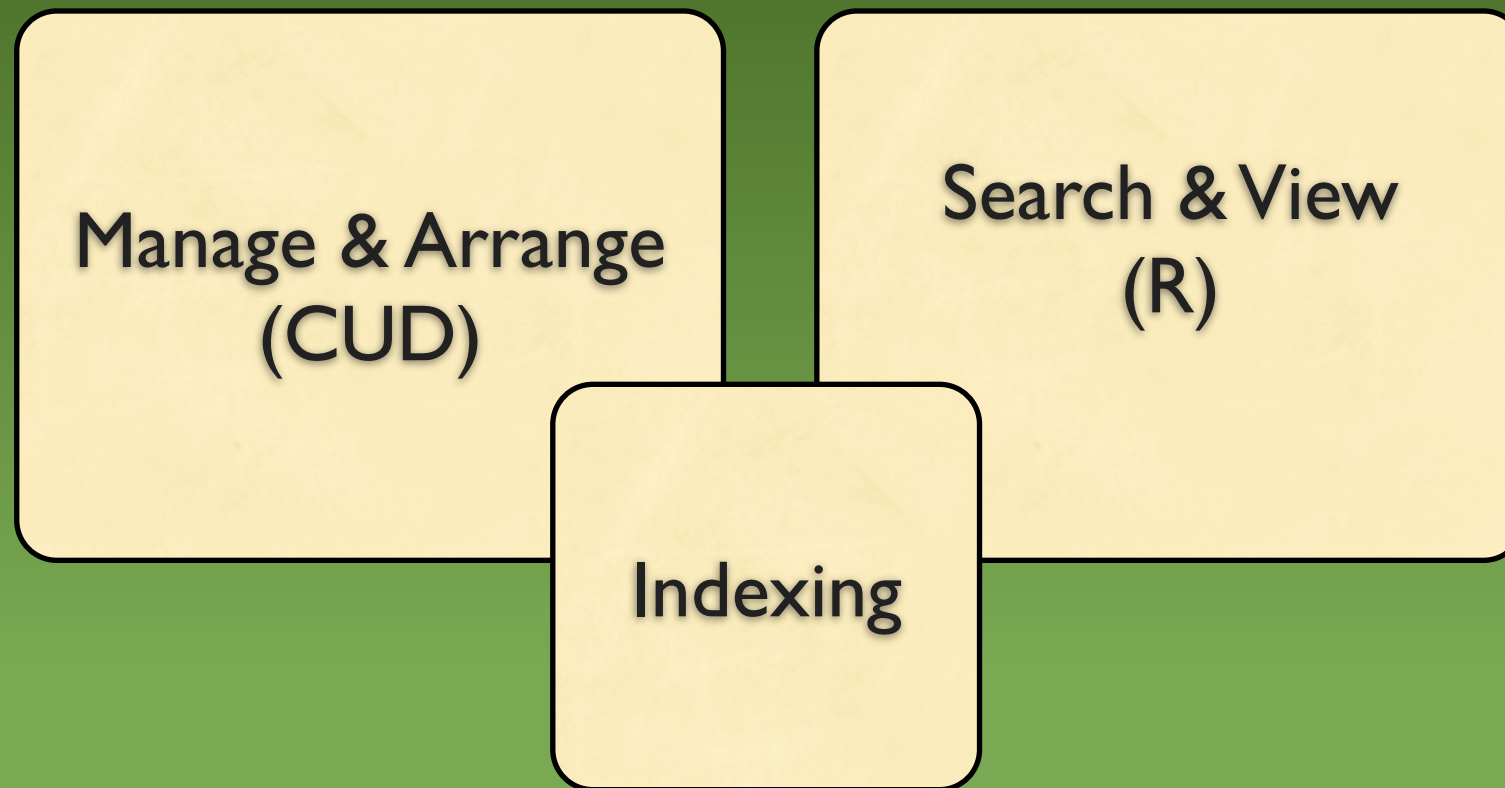


The diagram consists of two stacked rectangular boxes. The top box is light yellow with rounded corners and a thin black border, containing the text 'Rails MVC'. The bottom box is a smaller, solid light yellow rectangle containing the text 'RDBMS' in green. The boxes are centered horizontally and stacked vertically, indicating that the Rails MVC layer sits on top of the RDBMS layer.

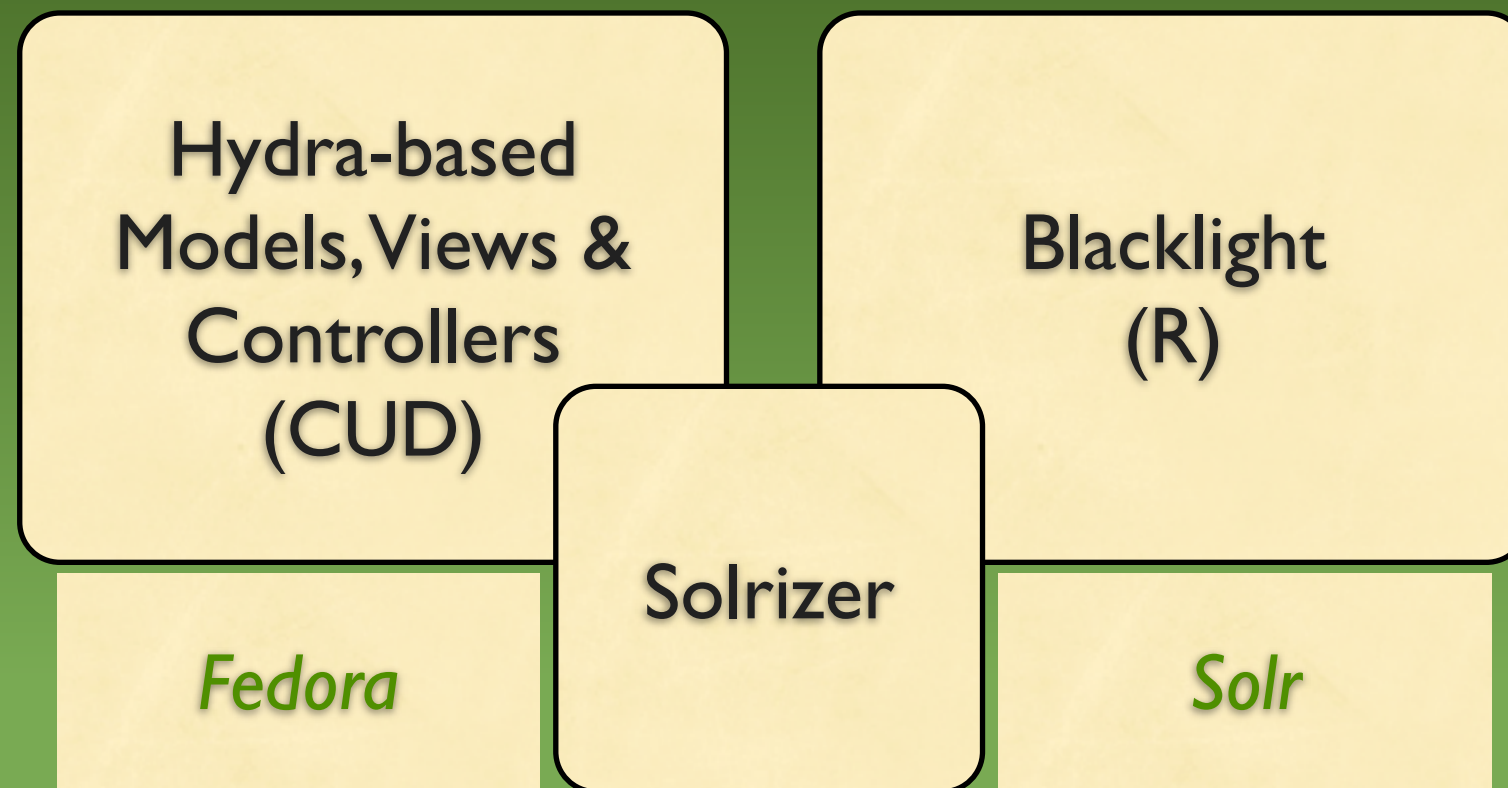
Rails MVC

RDBMS

CRUD in repositories



the Hydra framework

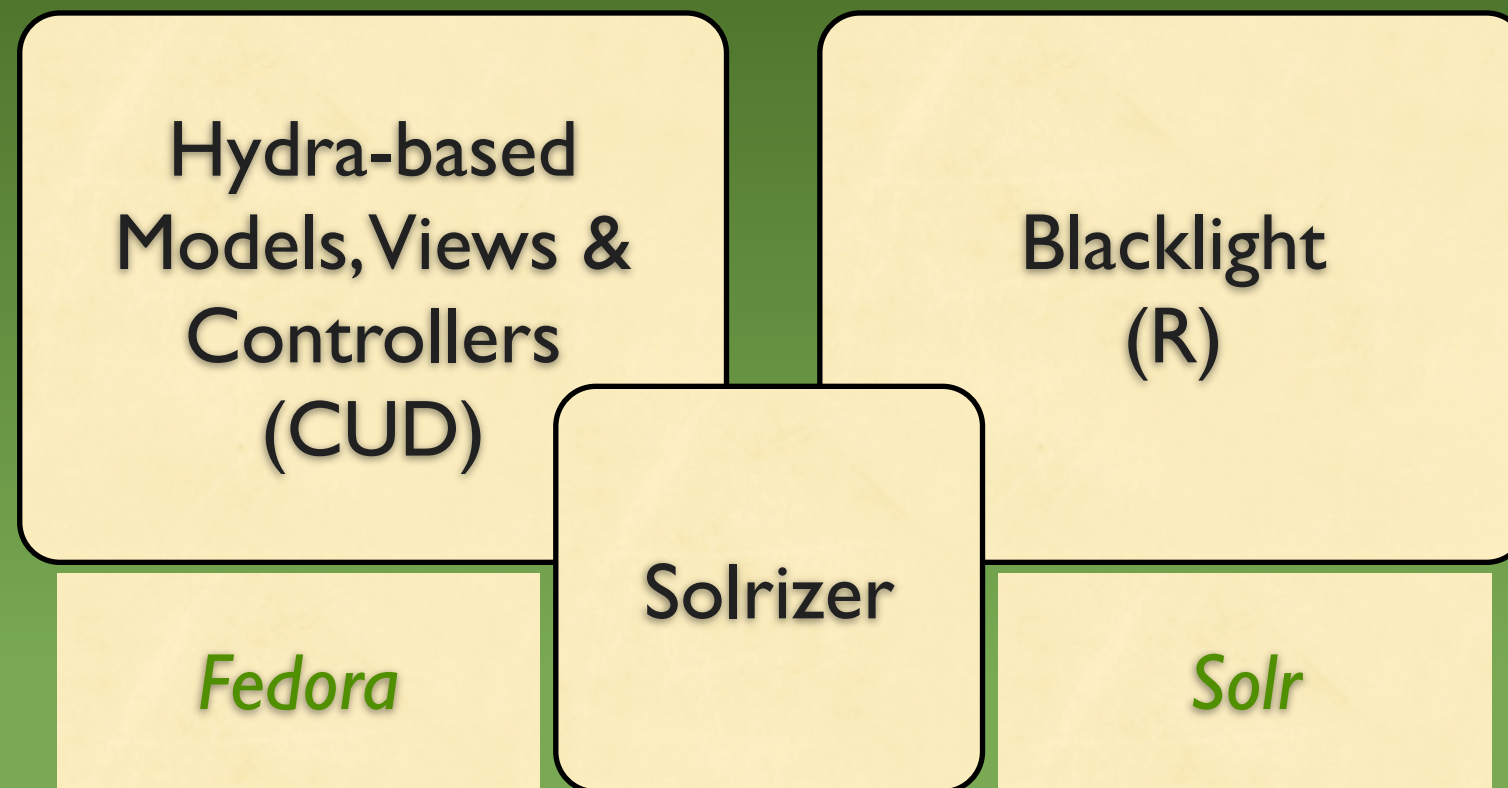


<http://github.com/projecthydra/hydrangea>
<https://wiki.duraspace.org/display/hydra/>



The Architecture

the Hydra framework



<http://github.com/projecthydra/hydrangea>
<https://wiki.duraspace.org/display/hydra/>

Browser/
Client
Side

HTML, CSS & JavaScript
or REST Client Applications

Hydra REST API

Blacklight REST API

Custom
Controllers

Custom
Views

Blacklight
Controllers

Blacklight
Views

Ruby on
Rails
(Server)

Hydra Access Controls

Custom Models

OM

ActiveFedora

Rubydora

Solrizer

Blacklight Search
& Facet Logic

User Model

Fedora REST API

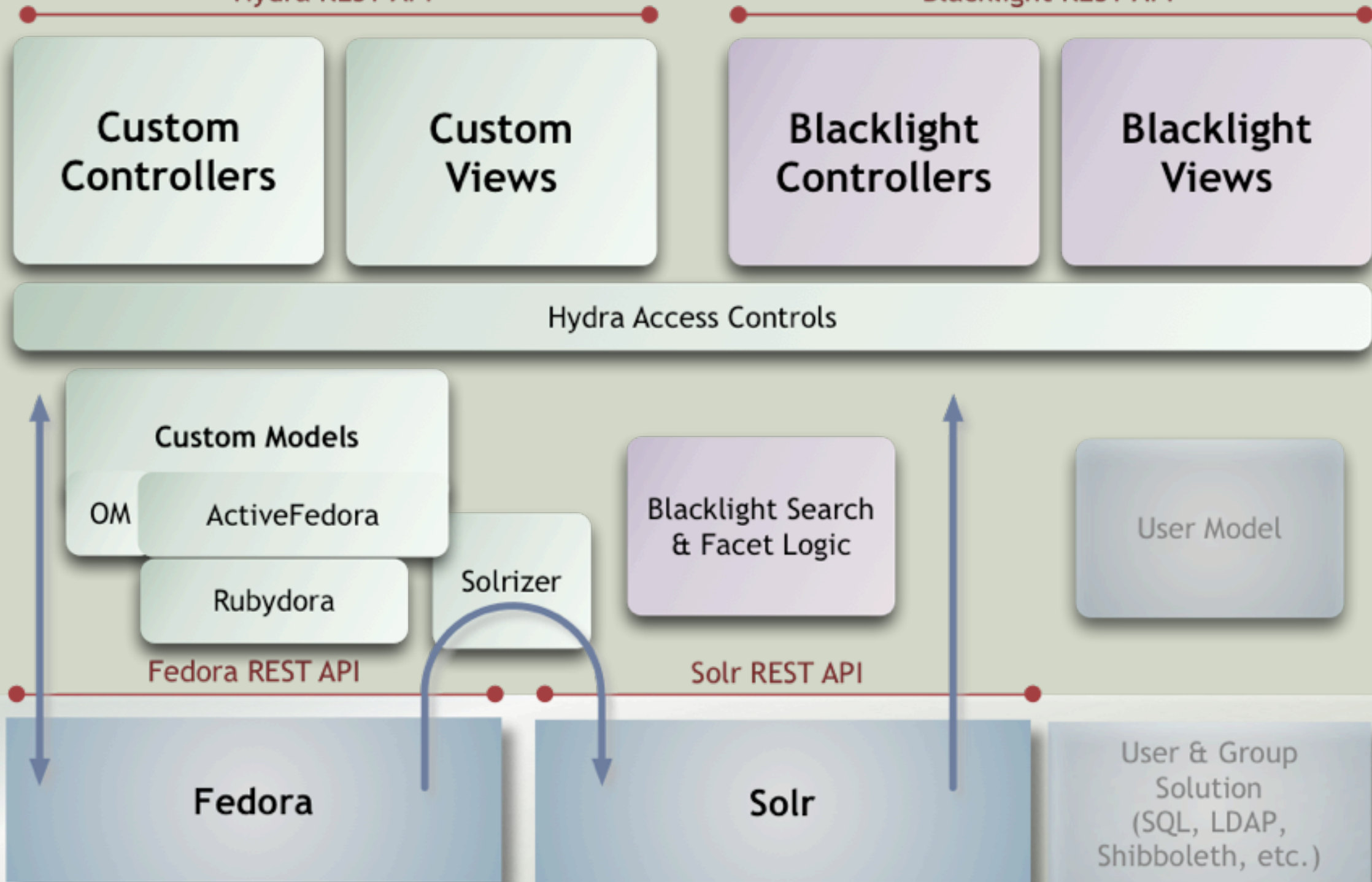
Solr REST API

Java & SQL
(System
Services)

Fedora

Solr

User & Group
Solution
(SQL, LDAP,
Shibboleth, etc.)



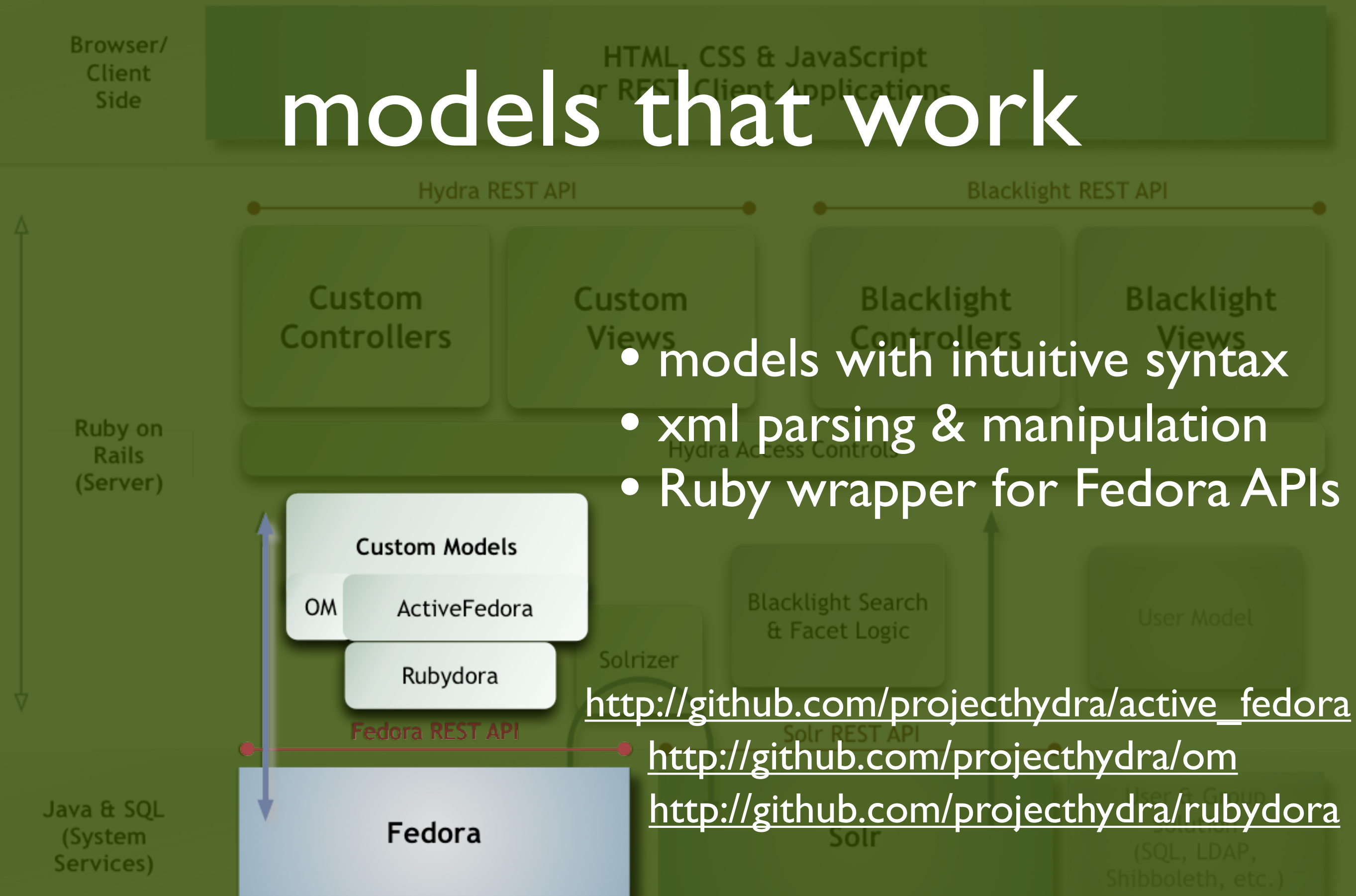
models that work

- models with intuitive syntax
- xml parsing & manipulation
- Ruby wrapper for Fedora APIs

http://github.com/projecthydra/active_fedora

<http://github.com/projecthydra/om>

<http://github.com/projecthydra/rubydora>



Browser/
Client
Side

MVC on Fedora

HTML, CSS & JavaScript
or REST Client Applications

Hydra REST API

Blacklight REST API

Custom
Controllers

Custom
Views

Blacklight
Controllers

Blacklight
Views

Ruby on
Rails
(Server)

Hydra Access Controls

Custom Models

OM

ActiveFedora

Rubydora

- models: active-fedora & om
- views: (regular rails)
- controllers: (regular rails)

Blacklight Search
& Facet Logic

User Model

Solrizer

Fedora REST API

Solr REST API

Java & SQL
(System
Services)

Fedora

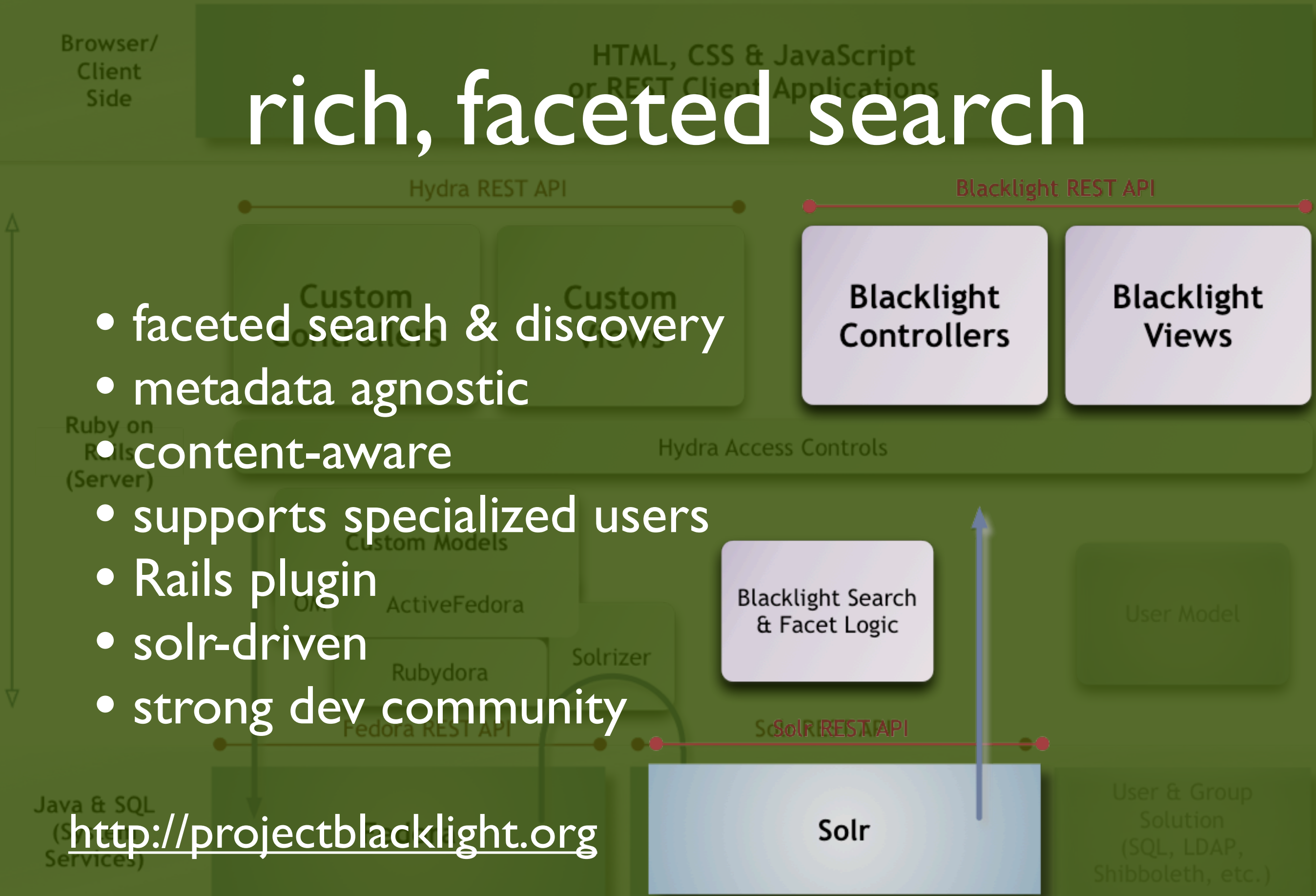
Solr

User & Group
Solution
(SQL, LDAP,
Shibboleth, etc.)

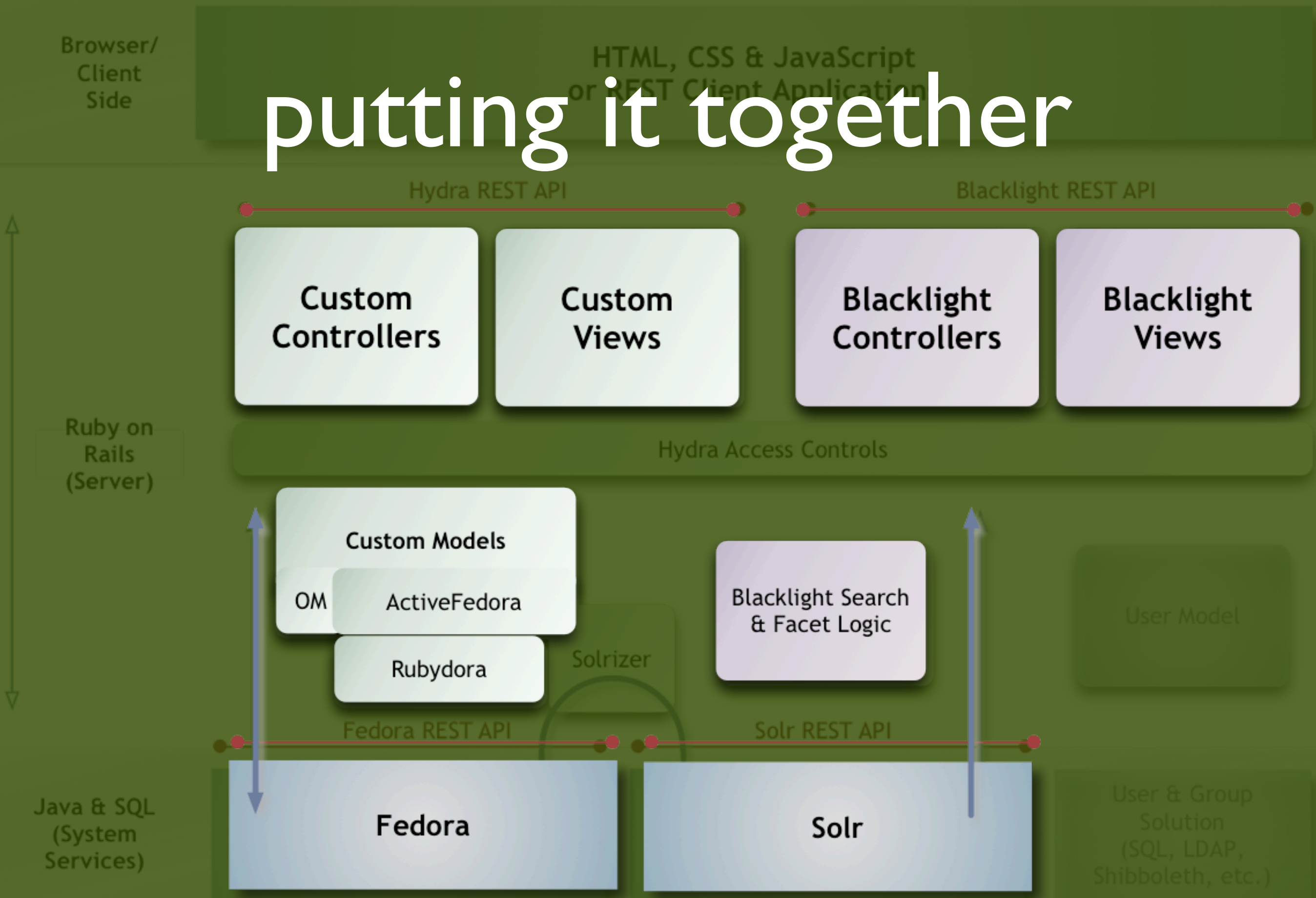
rich, faceted search

- faceted search & discovery
- metadata agnostic
- content-aware
- supports specialized users
- Rails plugin
- solr-driven
- strong dev community

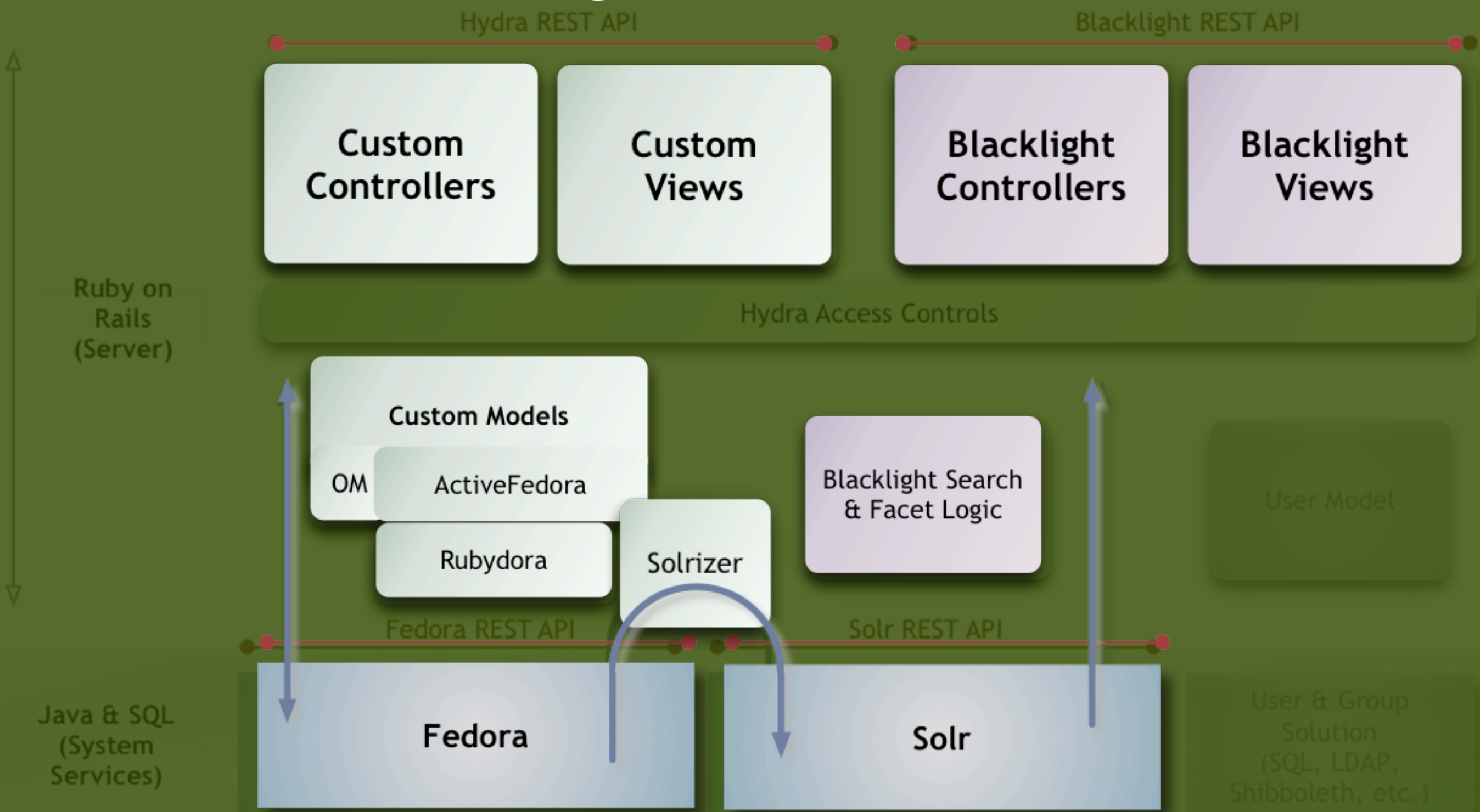
<http://projectblacklight.org>



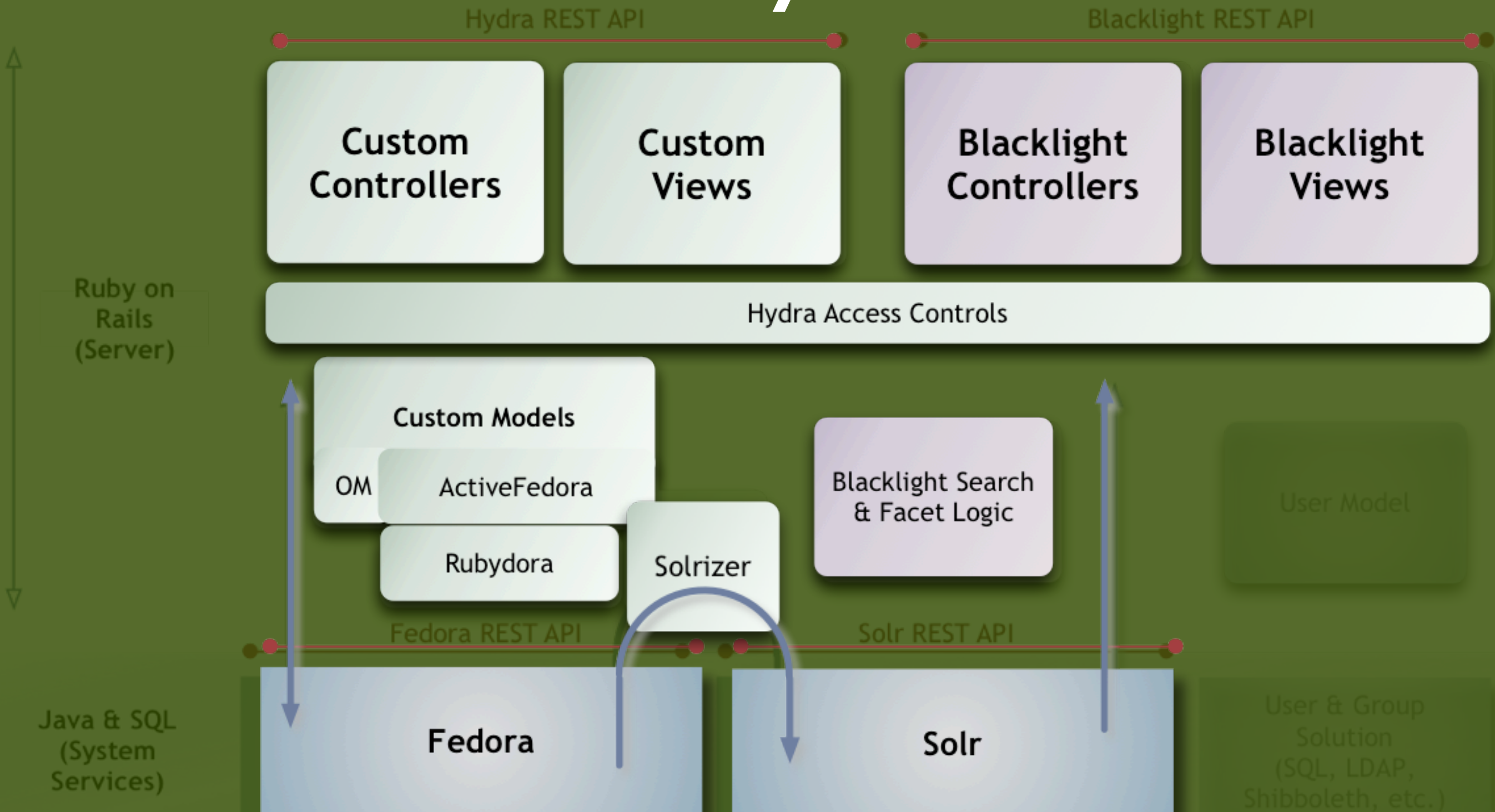
putting it together



indexing with solrizer



a whole hydra head





Key Technical Parts

the technology stack presents a **hierarchy of promises** where each piece relies on a set of dependencies, and provides its own set of promises

<http://is.gd/hydrastack>

pragmatic modeling syntax
makes it easy for you to use Fedora
without Fedora using you

abstraction layer on top of
repository opens possibility for
putting Hydra on something other
than Fedora

simple, configurable **solrization** &
faceted search with **gated**
discovery

access controls that use
Hydra rightsMetadata, **not XACML**

Browser/
Client
Side

HTML, CSS & JavaScript
or REST Client Applications

Hydra REST API

Blacklight REST API

Custom
Controllers

Custom
Views

Blacklight
Controllers

Blacklight
Views

Ruby on
Rails
(Server)

Hydra Access Controls

Custom Models

OM

ActiveFedora

Rubydora

Solrizer

Blacklight Search
& Facet Logic

User Model

Fedora REST API

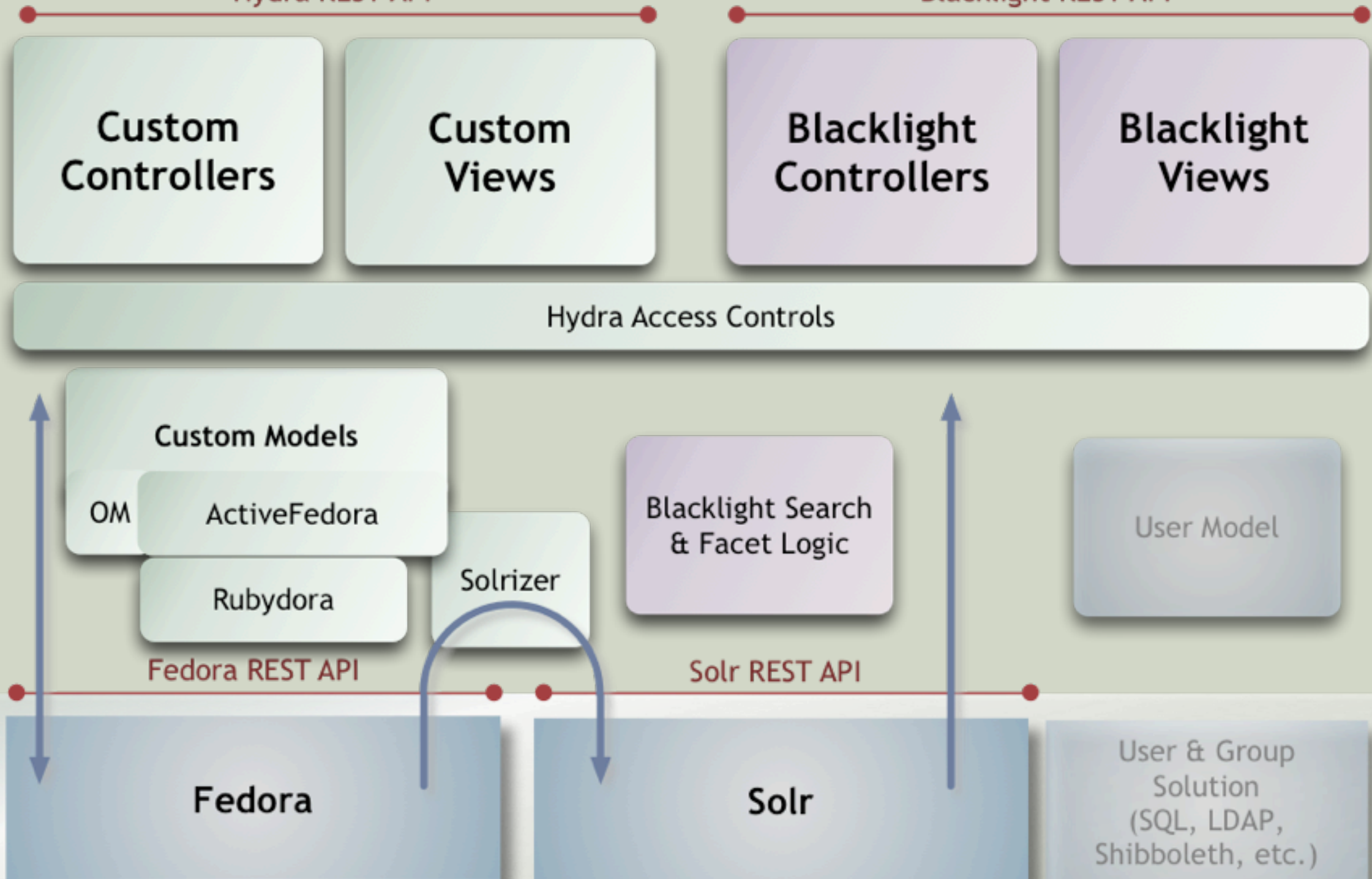
Solr REST API

Java & SQL
(System
Services)

Fedora

Solr

User & Group
Solution
(SQL, LDAP,
Shibboleth, etc.)



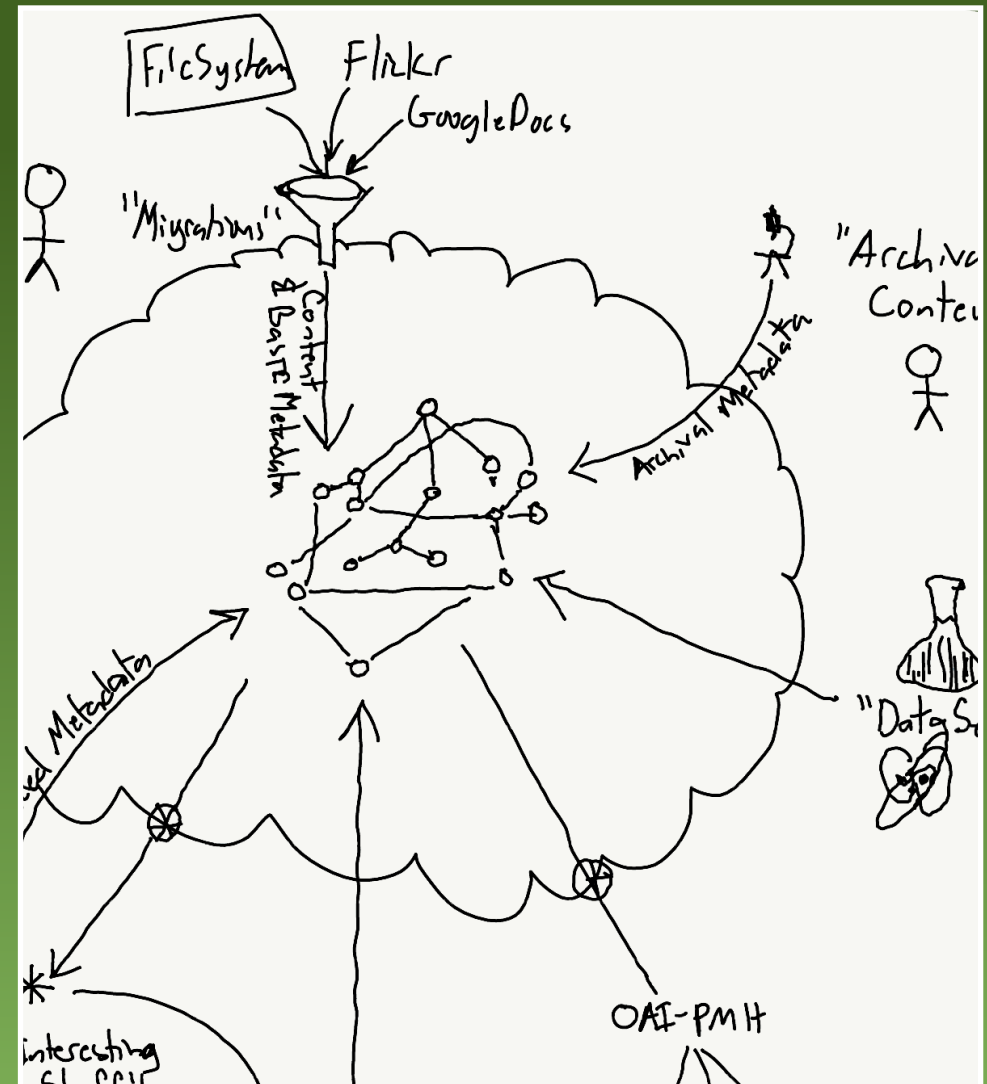
now the fun
begins...

<http://is.gd/hydrastack>

<http://projecthydra.org>

<http://yourmediashelf.com>

<http://curationexperts.com>



MediaShelf