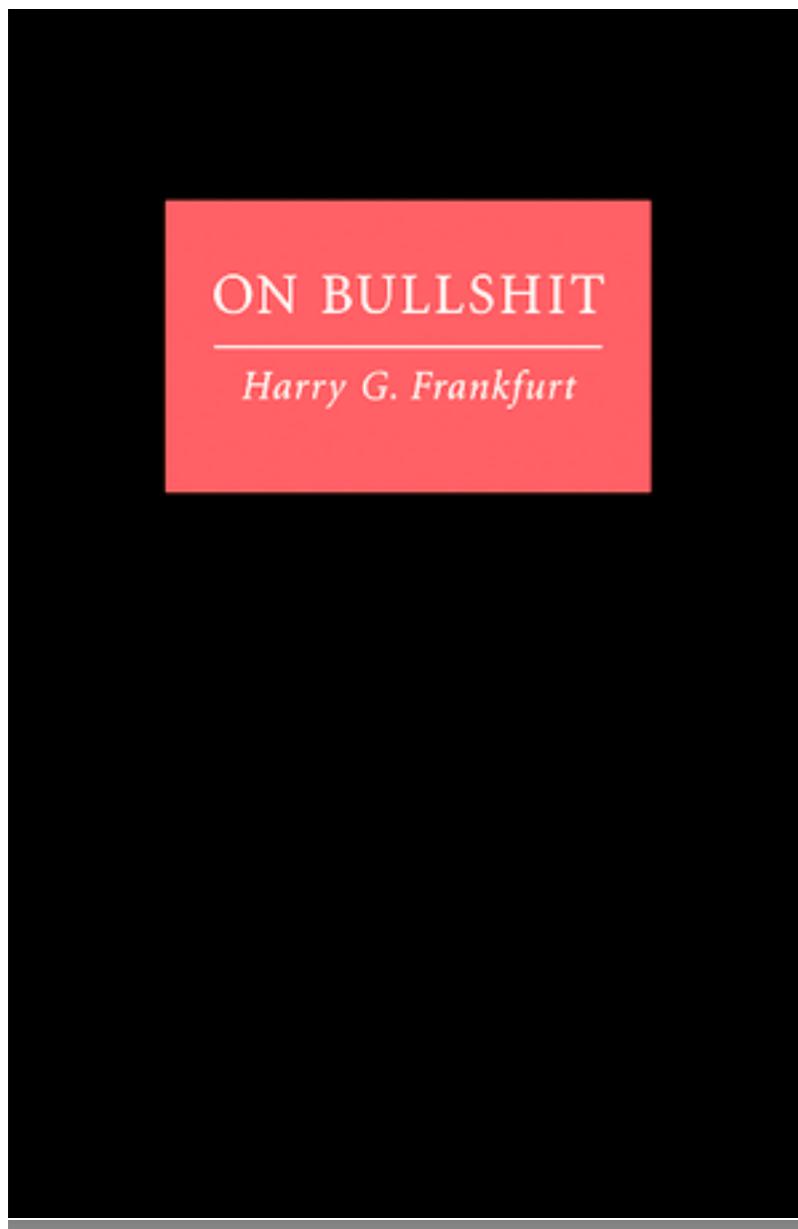


The Zero ~~Business~~ Architecture

Day Business & Tech Summit Chicago,
May 2008

Lars Trieloff, Product Manager

Before I start



“One of the most salient features of our culture is that there is so much bullshit.”

—Prof. Harry G. Frankfurt

Hello, my name is Lars Trieloff

- Product Manager Collaboration and Digital Asset Management
- With Day since November 2007
- Background:
 - Open Source Software Development
 - Collaboration Software
 - Technical Documentation (DocBook)
 - Blogging
 - IT Systems Engineering



Hello, my name is Lars Trieloff

- Product Manager Collaboration and Digital Asset Management
- With Dan Winkler 2007
- Background
 - Open Source Software Development
 - Collaboration Software
 - Technical Documentation (DocBook)
 - Blogging
 - IT Systems Engineering

**Product
Management:
Department of "not
yet"**



Goal Review

Goal Review

We want to build web
applications

Goal Review

We want to build web
applications to deliver value to our users

Goal Review

We want to build web applications to deliver value to our users

web applications that allow integration

Goal Review

We want to build web applications to deliver value to our users

web applications that allow integration and enable personalization

Goal Review

We want to build web applications to deliver value to our users

web applications that allow integration and enable personalization

that establish new communication channels

Goal Review

We want to build web applications to deliver value to our users

web applications that allow integration and enable personalization

that establish new communication channels, increase usability

Goal Review

We want to build web applications to deliver value to our users web applications that allow integration and enable personalization that establish new communication channels, increase usability and scale to large amounts of users and content

Goal Review

We want to build web applications to deliver value to our users web applications that allow integration and enable personalization that establish new communication channels, increase usability and scale to large amounts of users and content within limited time & budget.

Sounds easy, but



**May you have interesting
software projects**

Sounds easy, but

Why do so many software projects end up becoming
interesting?



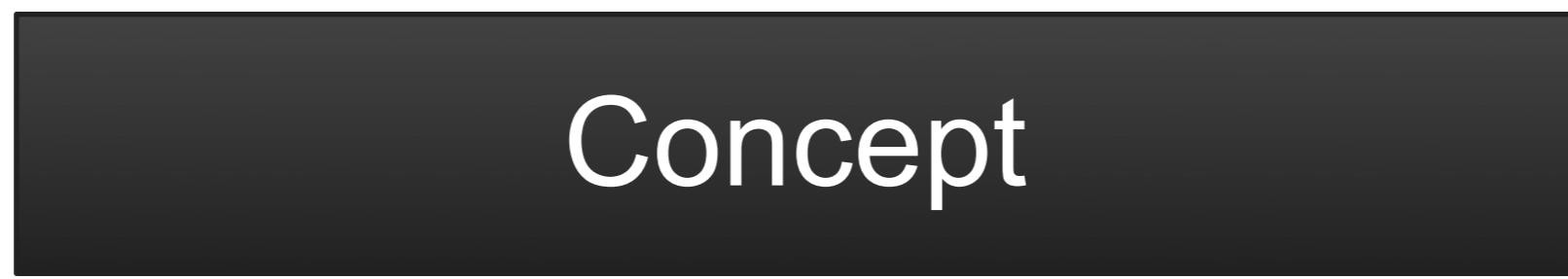
**May you have interesting
software projects**

Where do projects become interesting?

Where do projects become interesting?

Requirements

Where do projects become interesting?

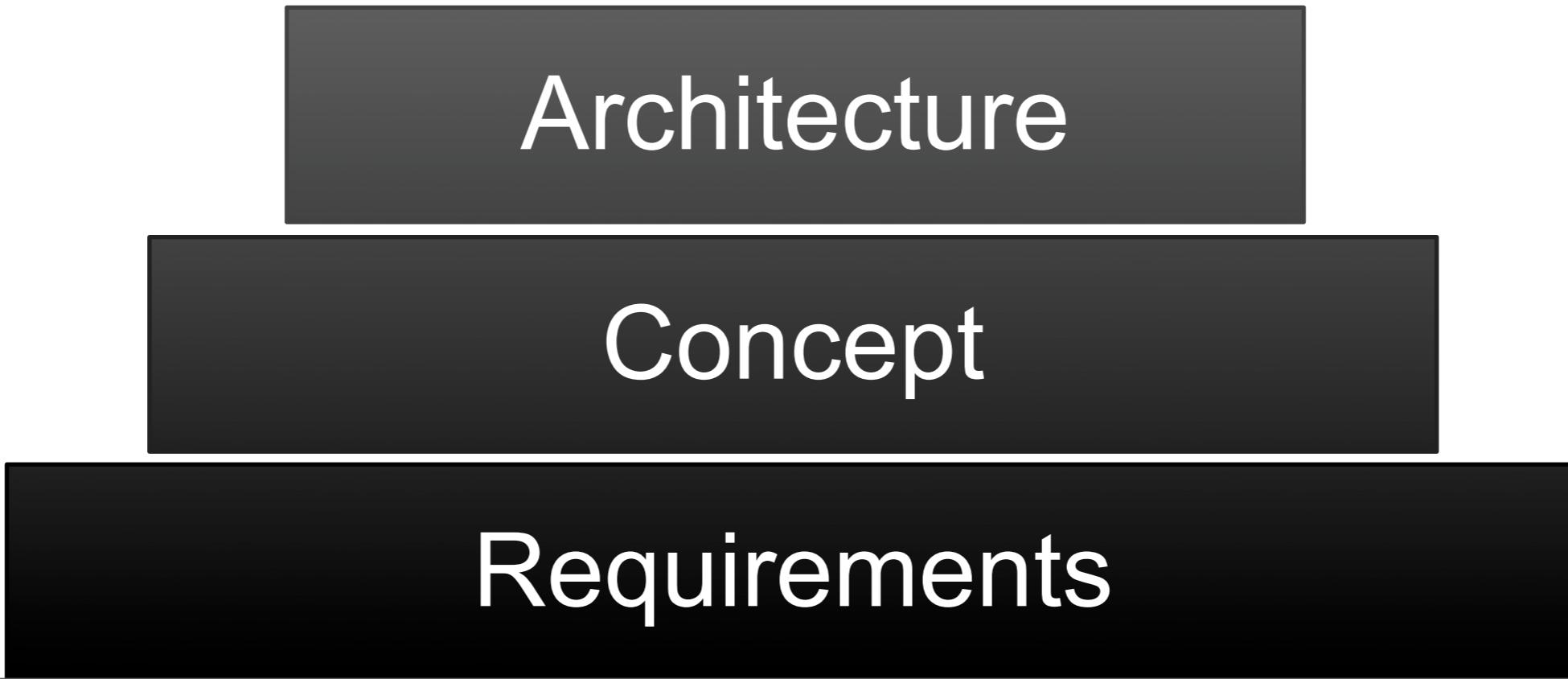


Concept



Requirements

Where do projects become interesting?

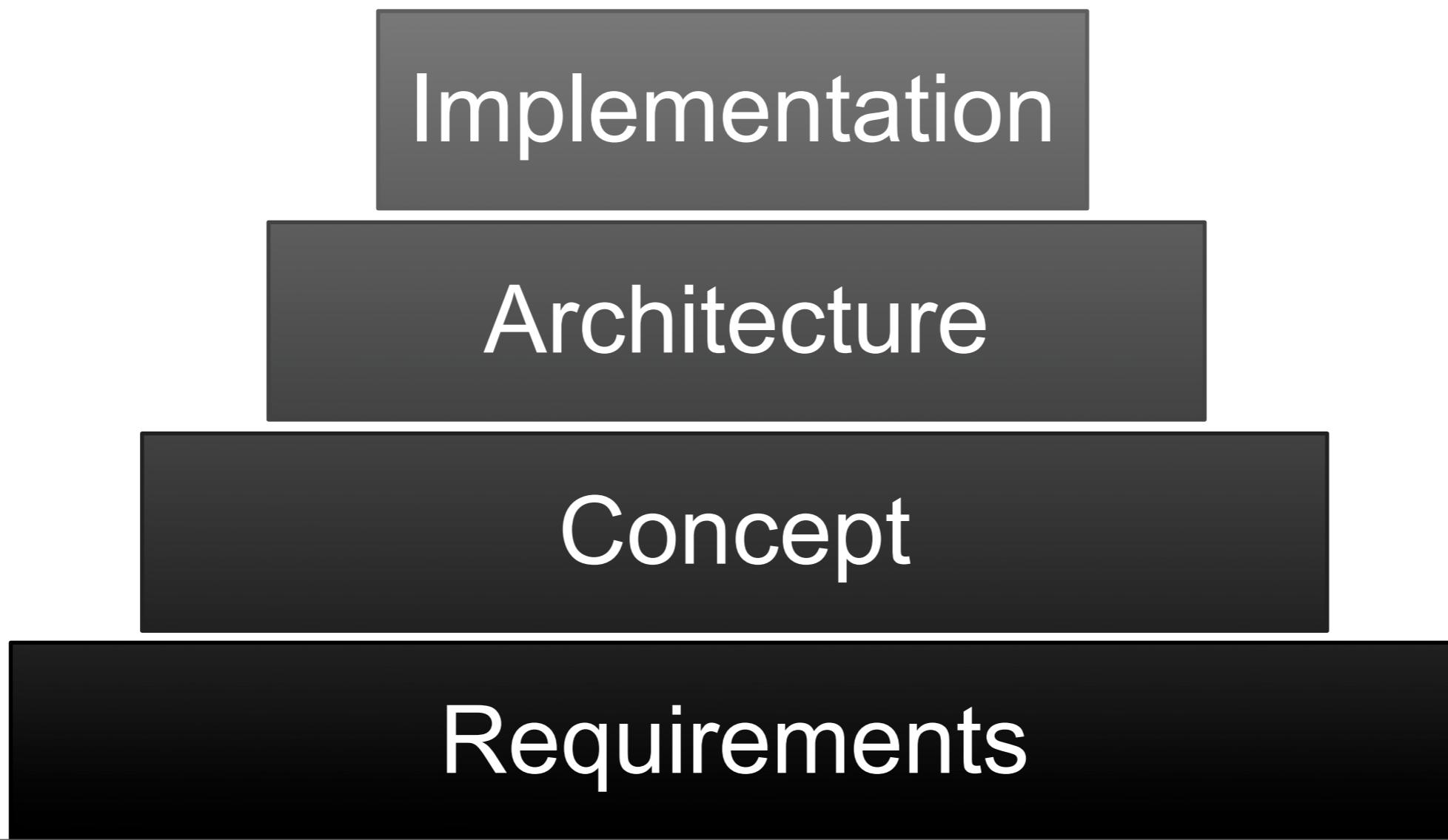


Architecture

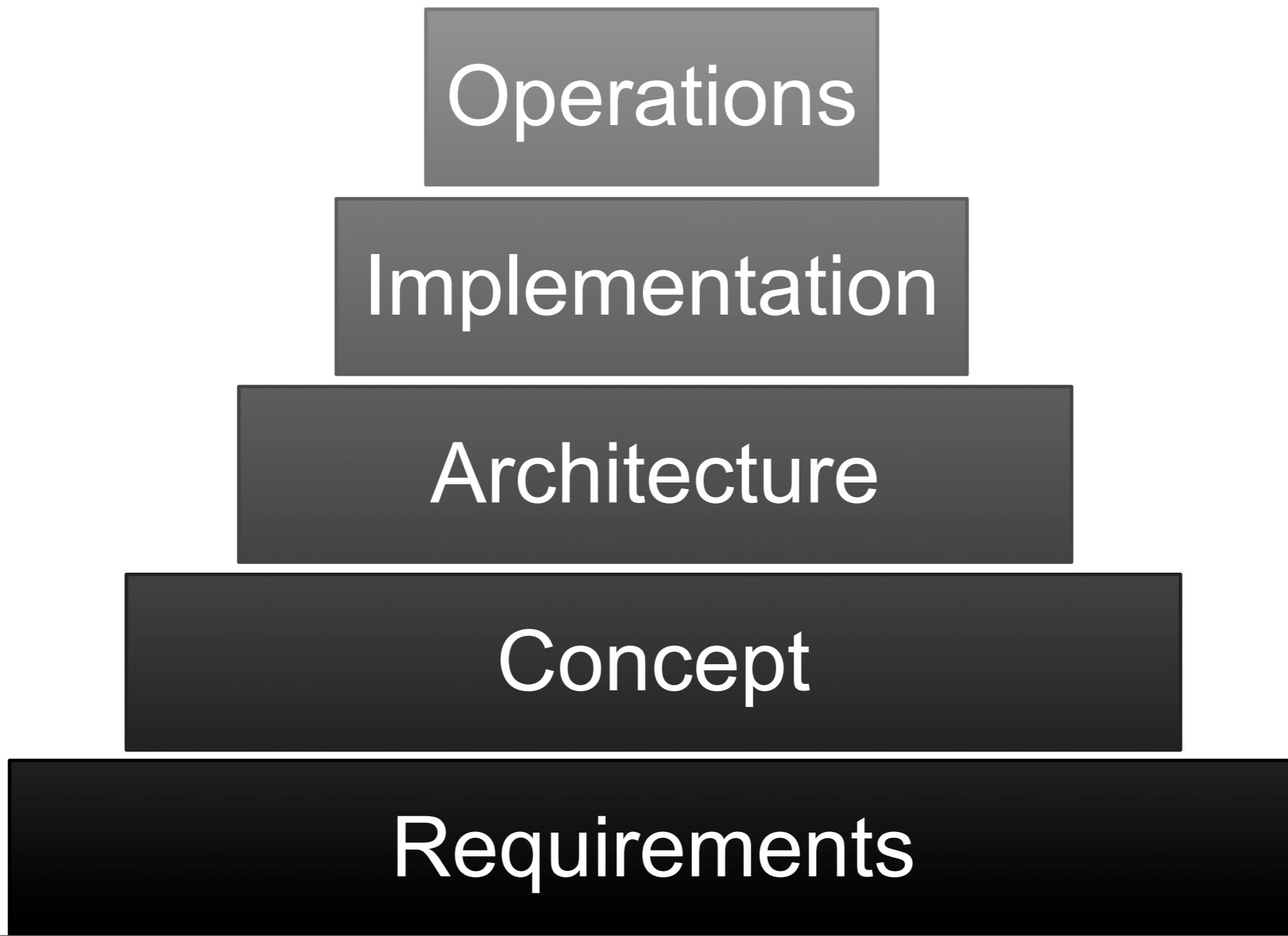
Concept

Requirements

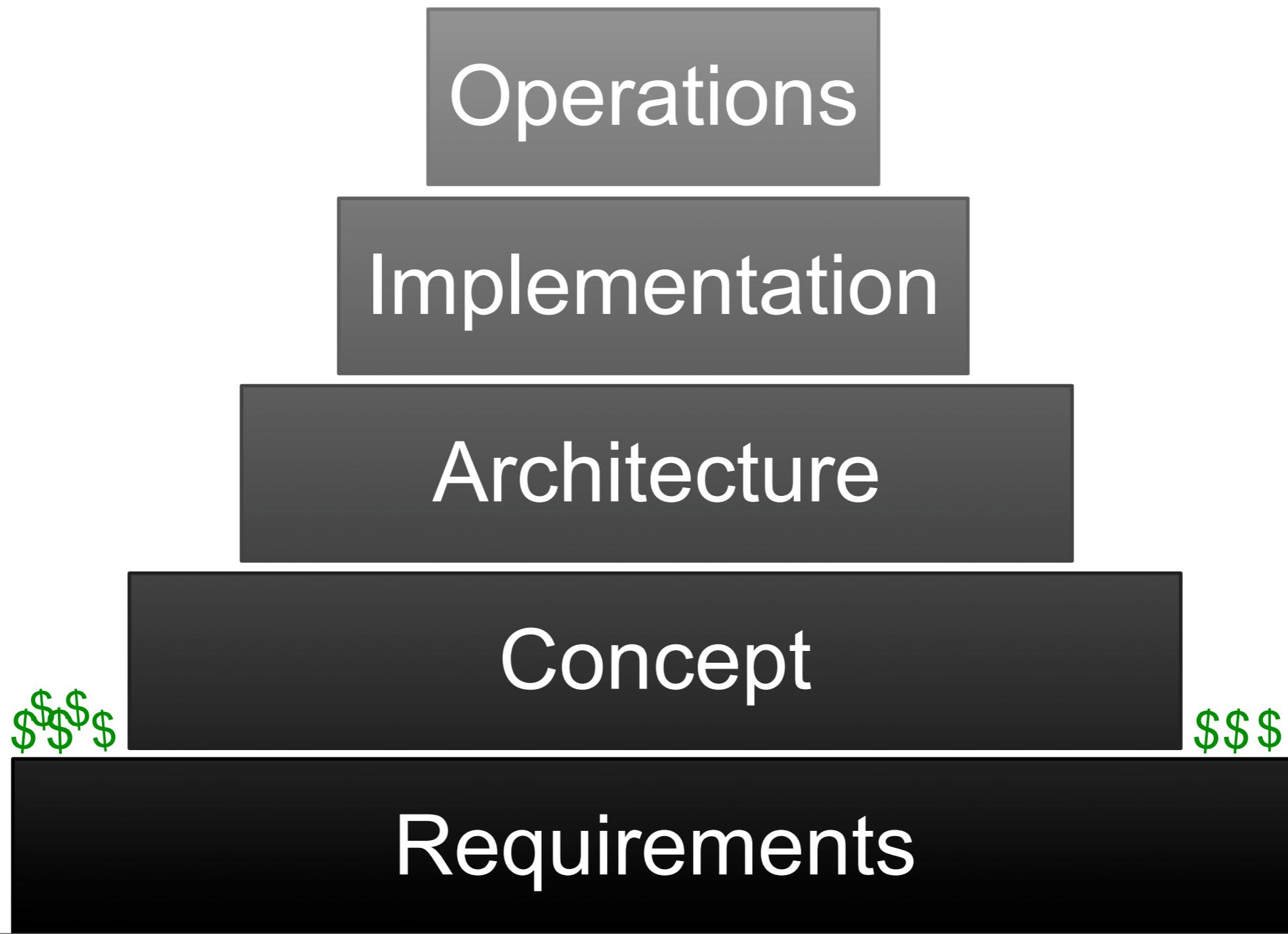
Where do projects become interesting?



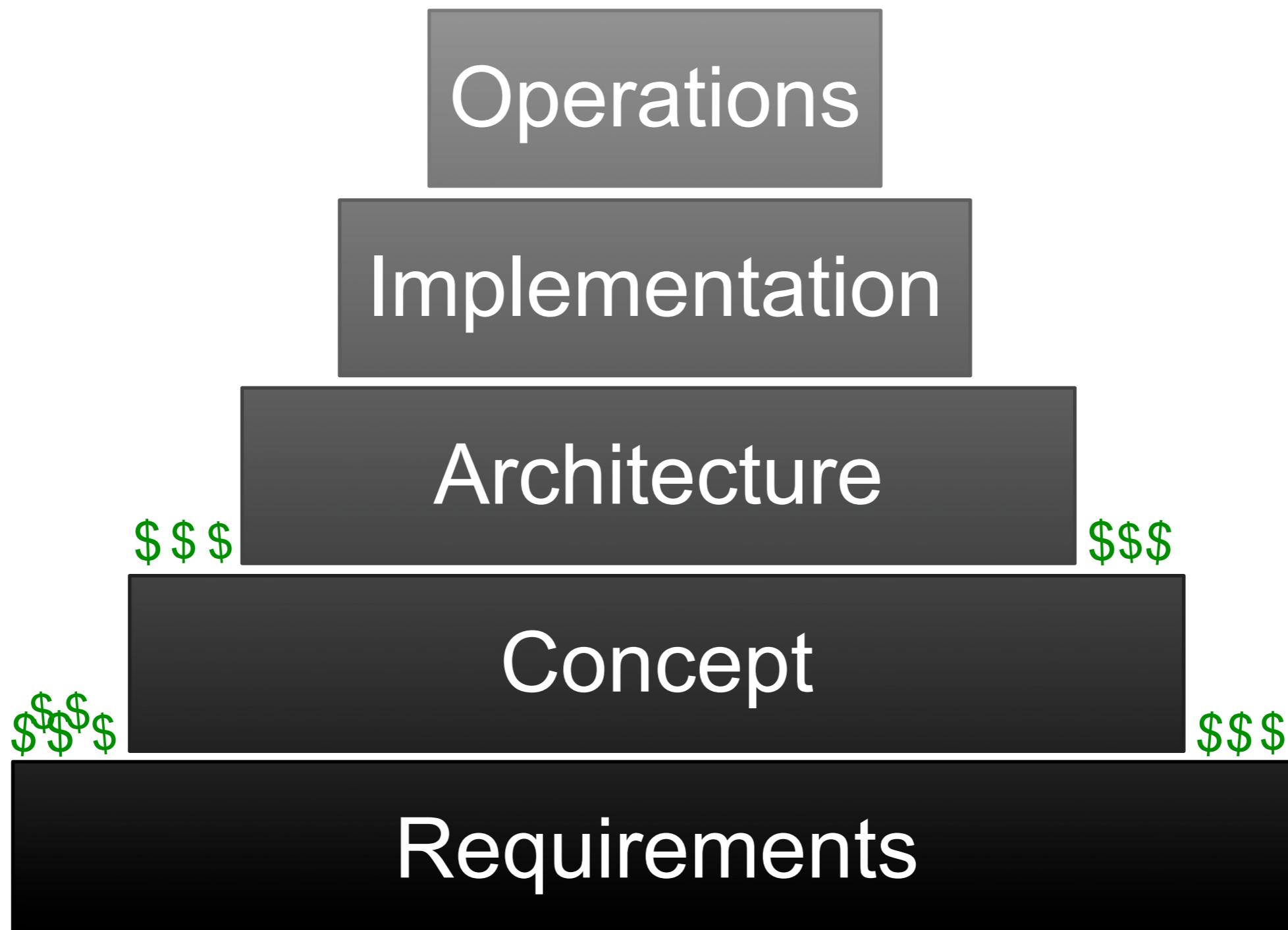
Where do projects become interesting?



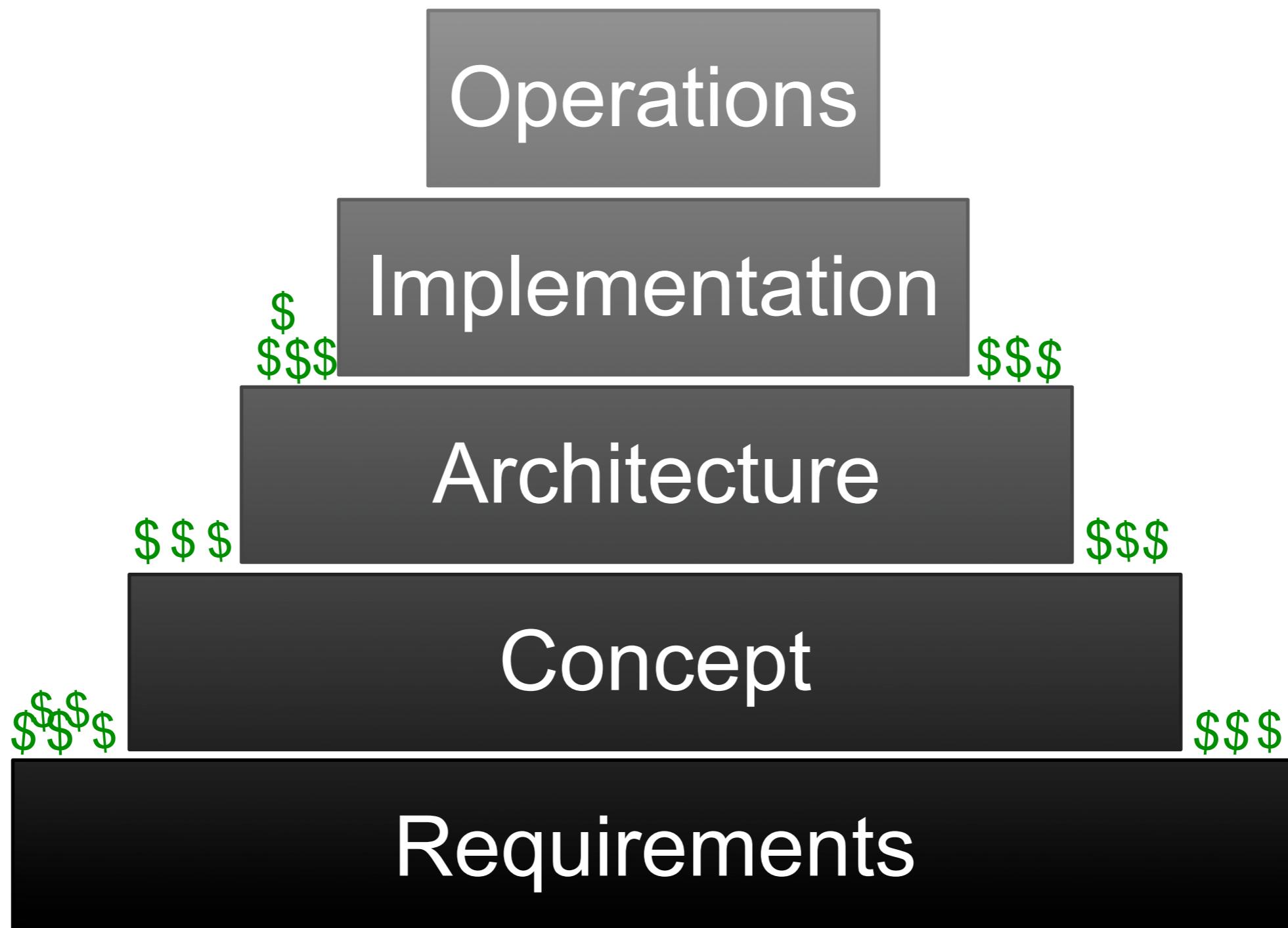
Where do projects become interesting?



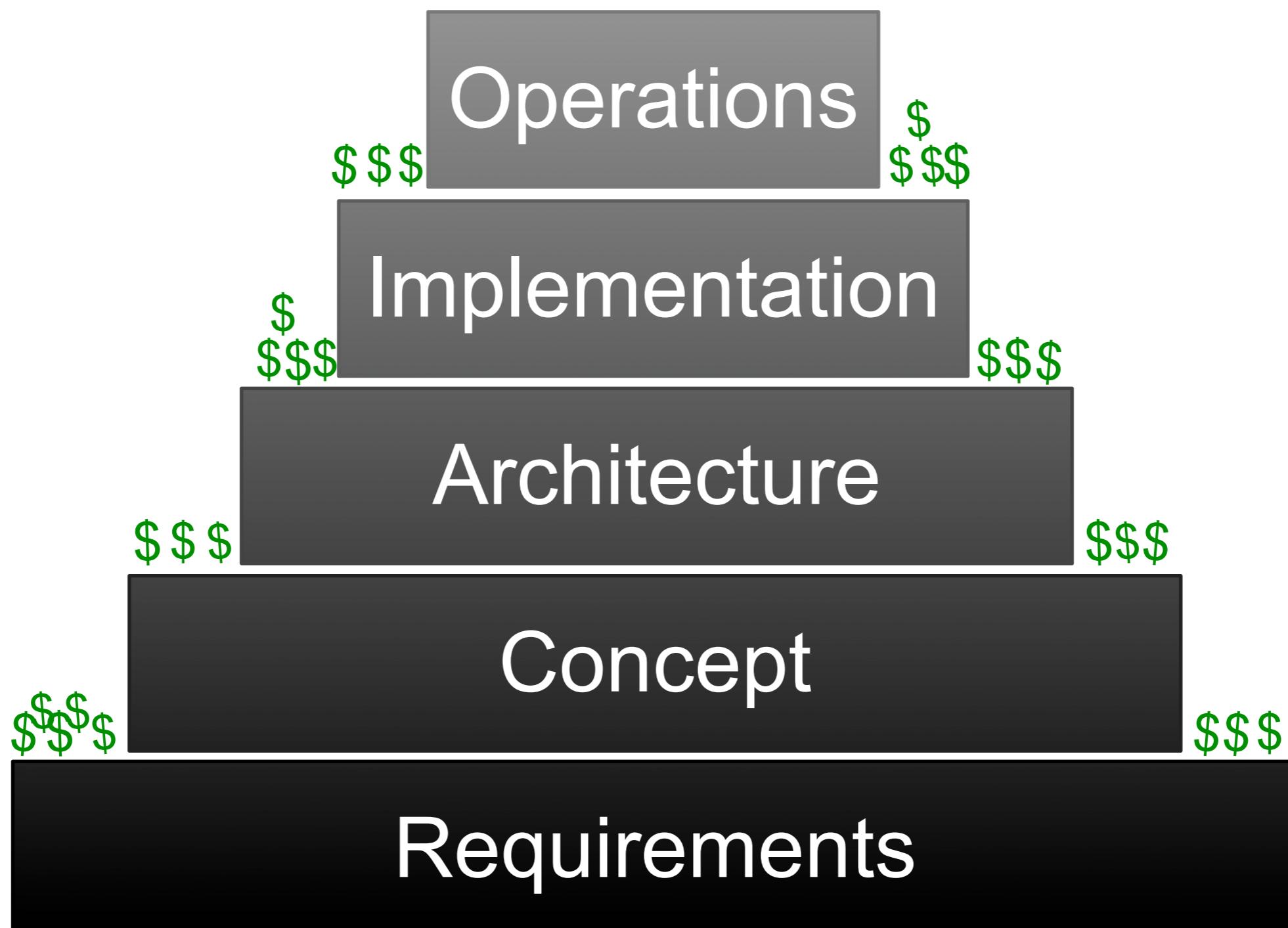
Where do projects become interesting?



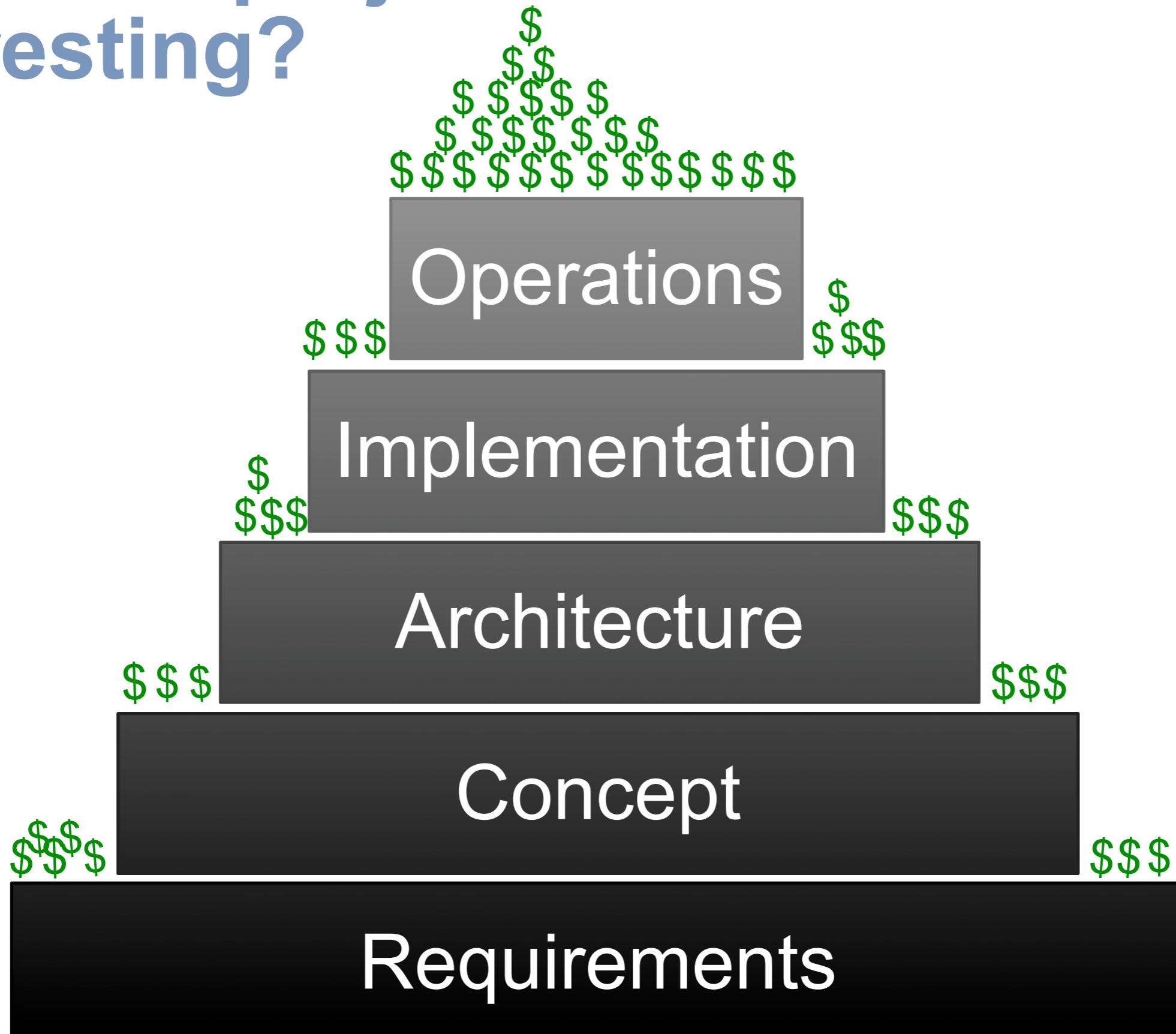
Where do projects become interesting?



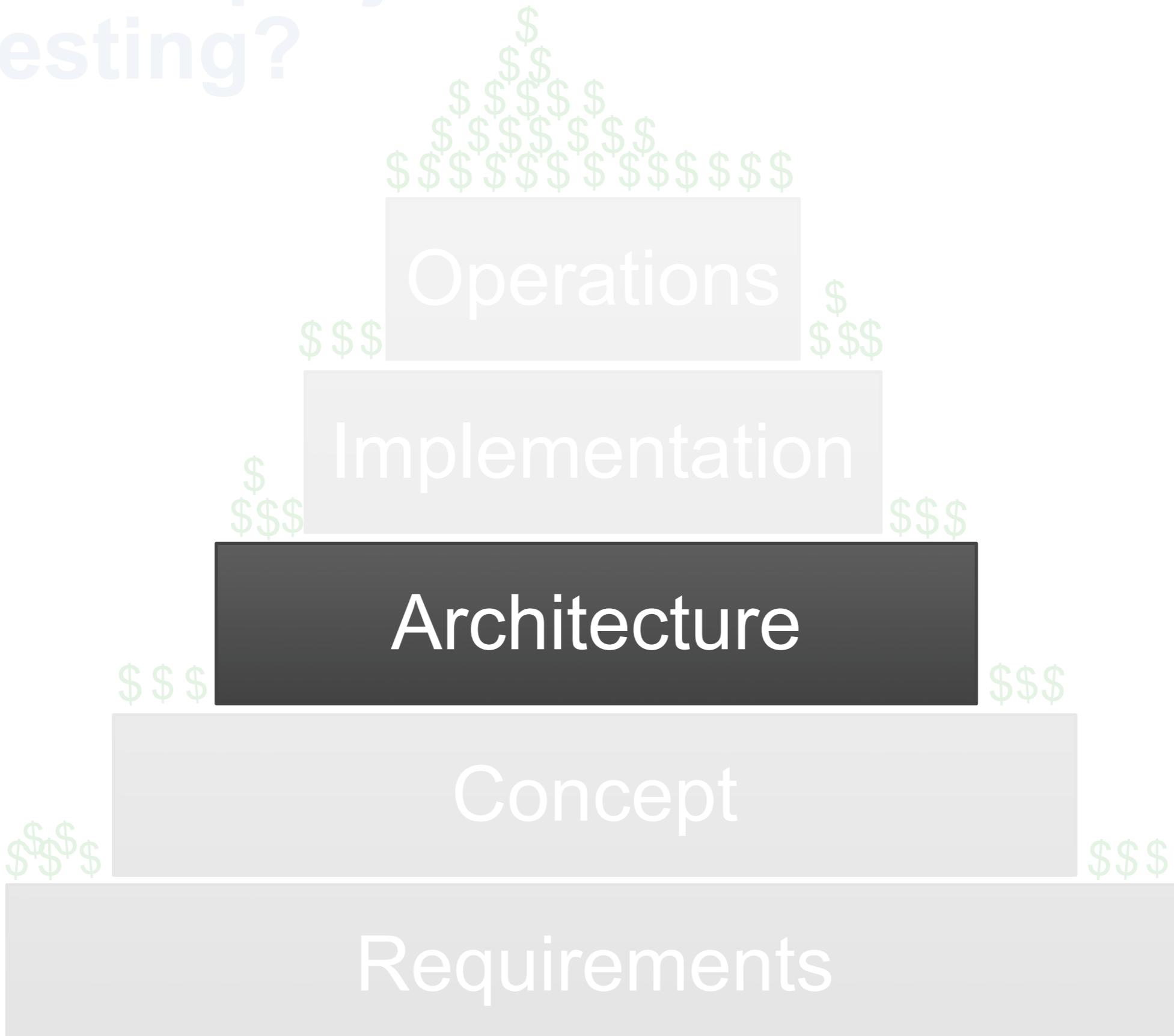
Where do projects become interesting?



Where do projects become interesting?



Where do projects become interesting?



Architectural Failure

■ Results of Architectural Failure

- Implementation and Operation too expensive
- Implementation too heavyweight, needs too much expensive resources
- Implementation not scalable, scaling becomes expensive
- Implementation not flexible, implementing new features becomes expensive
- System Components do not fit, integration becomes expensive

Architectural Failure

■ Results of Architectural Failure

- Implementation and Operation too expensive
- Implementation too heavyweight, needs to much expensive resources
- Implementation not scalable, scaling becomes expensive
- Implementation not flexible, implementing new features becomes expensive
- System Components do not fit, integration becomes expensive

You can fix everything later on, if you pay for it

Why does Architectural Failure happen?

Why does Architectural Failure happen?

“Nobody got fired for buying IBM”

Why does Architectural Failure happen?

**“Nobody ever
got fired for
buying
Microsoft”**

Why does Architectural Failure happen?

**“Nobody ever
got fired for
buying
Microsoft”**

**“We have
been doing it
this way
forever”**

Why does Architectural Failure happen?

“Nobody ever
got fired for
buying
Microsoft”

“We have
been doing it
this way
forever”

“Let’s use
Ruby on Rails,
everybody is
talking about
it.”

Why does Architectural Failure happen?

“Nobody ever
got fired for
buying
Microsoft”

“We have
been doing it
this way
forever”

“Let’s use
Ruby on Rails,
everybody is
talking about
it.”

“How will
EJB 3.0 look
on my
resume?”

Why does Architectural Failure happen?

“Nobody ever got fired for buying Microsoft”

“We have been doing it this way forever”

“Let’s use Ruby on Rails, everybody is talking about it.”

“How will EJB 3.0 look on my resume?”

“I do not need no capacity planning. I am using a scalable language.”

Why does Architectural Failure happen?

“Nobody ever
got fired for
buying
Microsoft”

“We have
been doing it
this way
forever”

“Let’s use
Ruby on Rails,
everybody is
talking about
it.”

“How will
EJB 3.0 look
on my
resume?”

“I do not need
no capacity
planning. I am
using a scalable
language.”

Why does Architectural Failure happen?

Result: Bullshit Architecture

“Let’s use Ruby on Rails, everybody is talking about it.”



“How will EJB 3.0 look on my resume?”



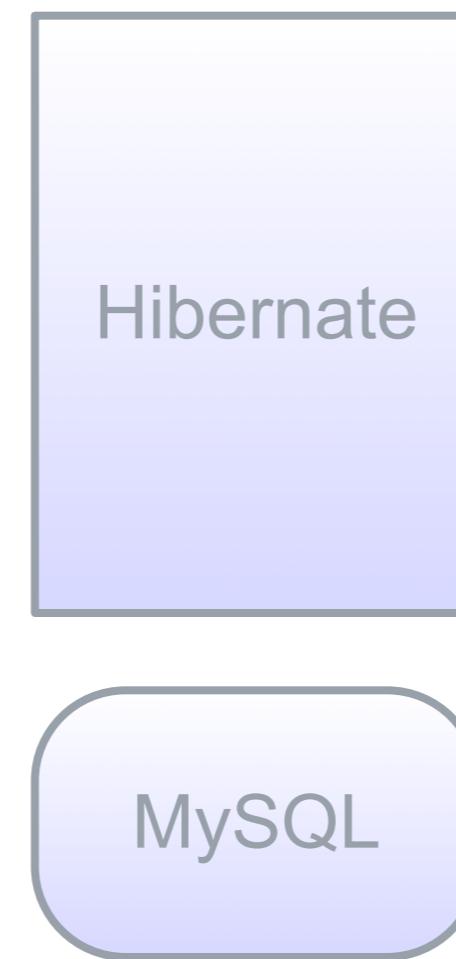
“I do not need no capacity planning. I am using a scalable language.”

Fighting Bullshit Architecture by Example

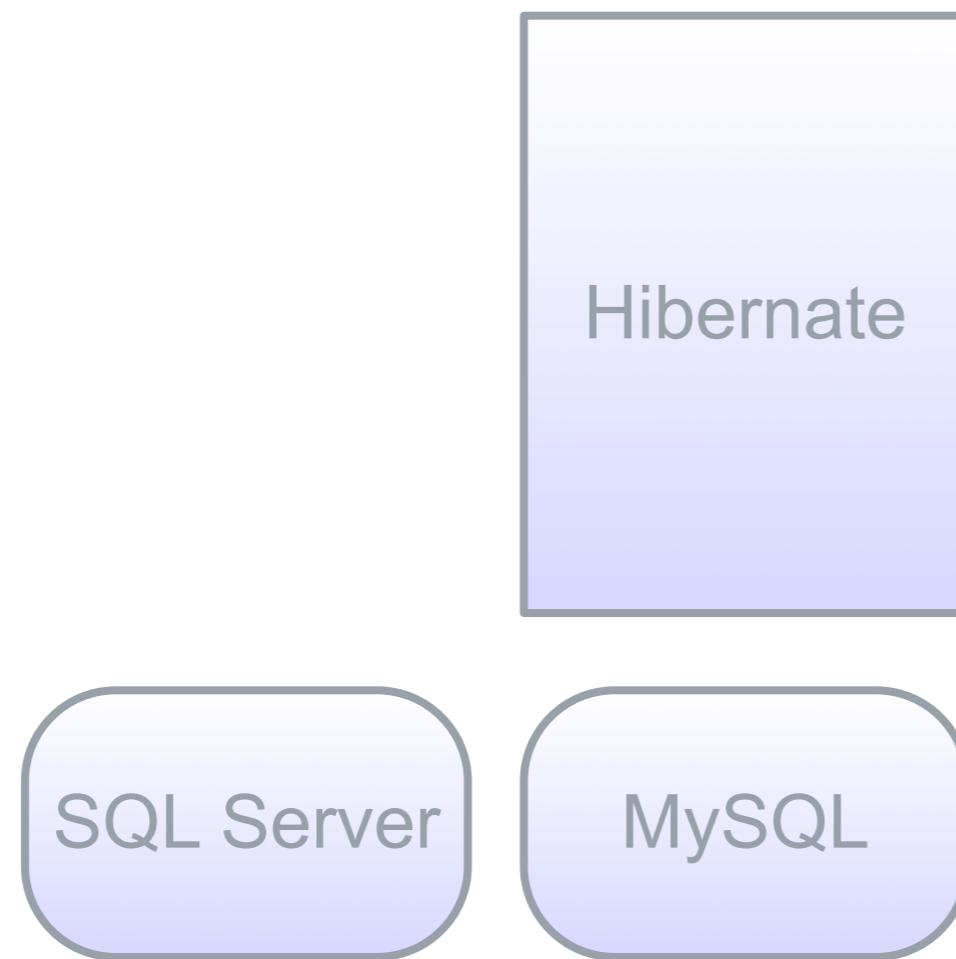
Fighting Bullshit Architecture by Example



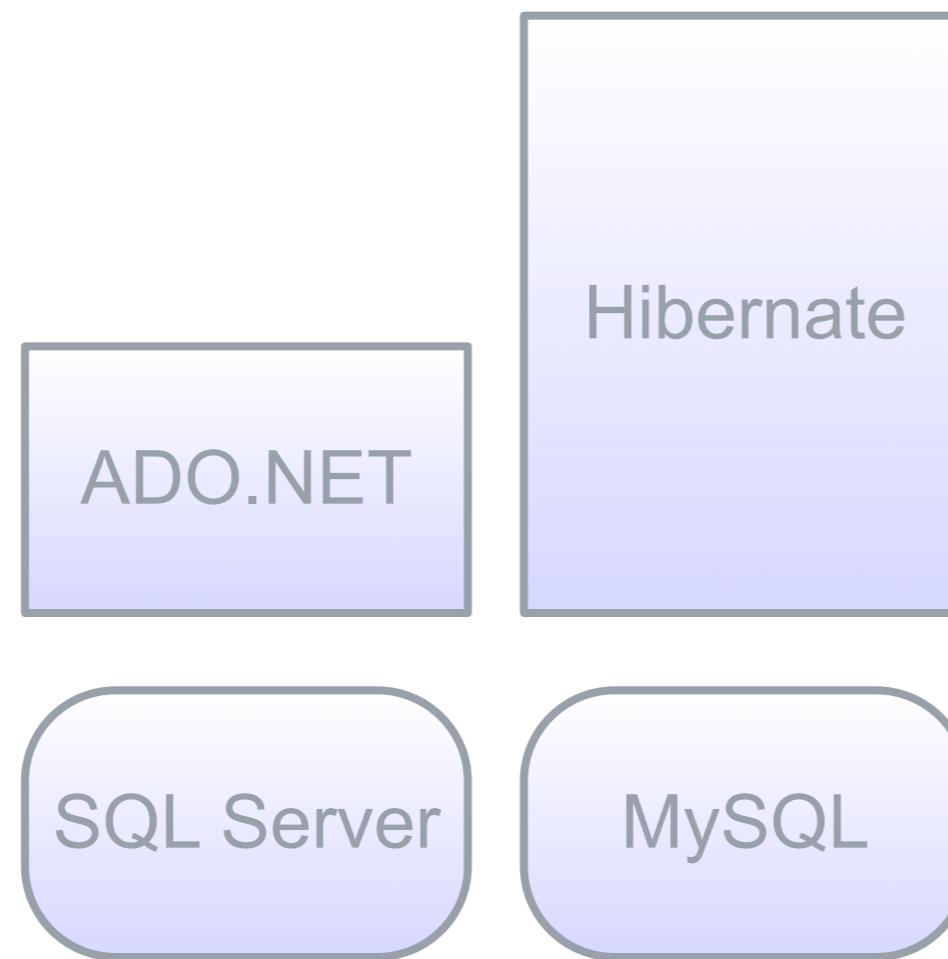
Fighting Bullshit Architecture by Example



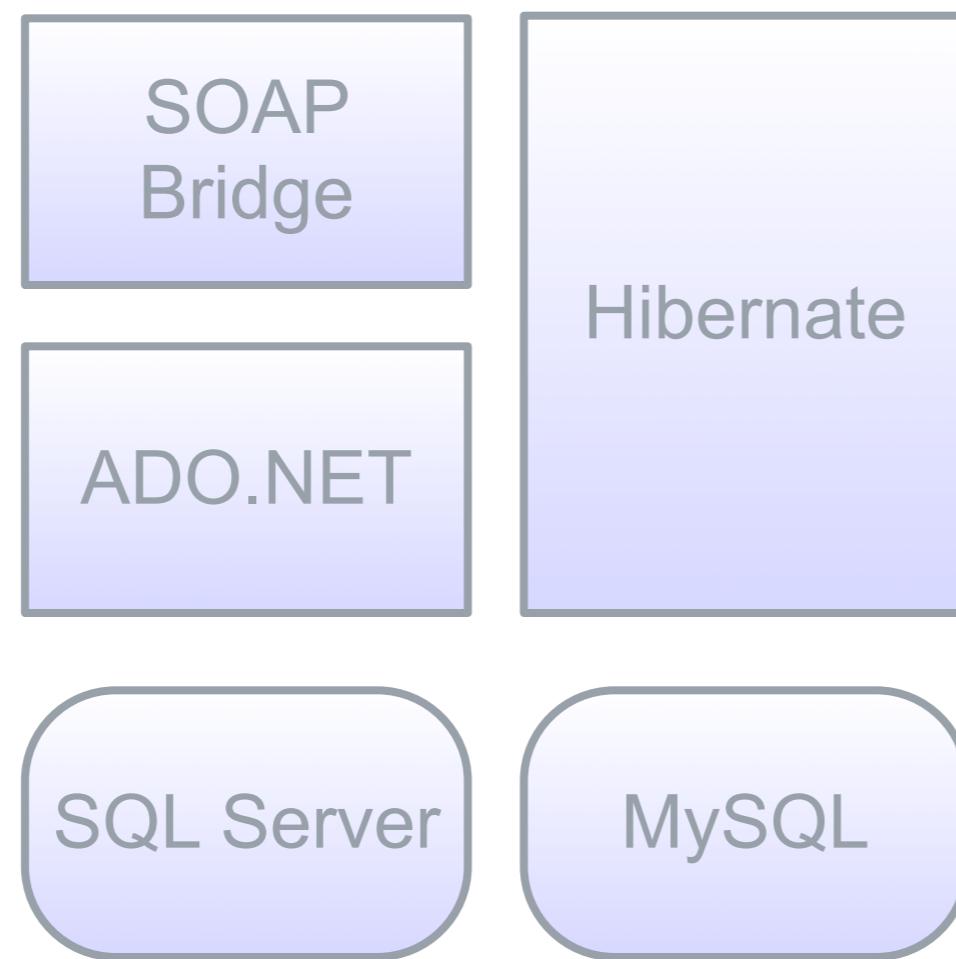
Fighting Bullshit Architecture by Example



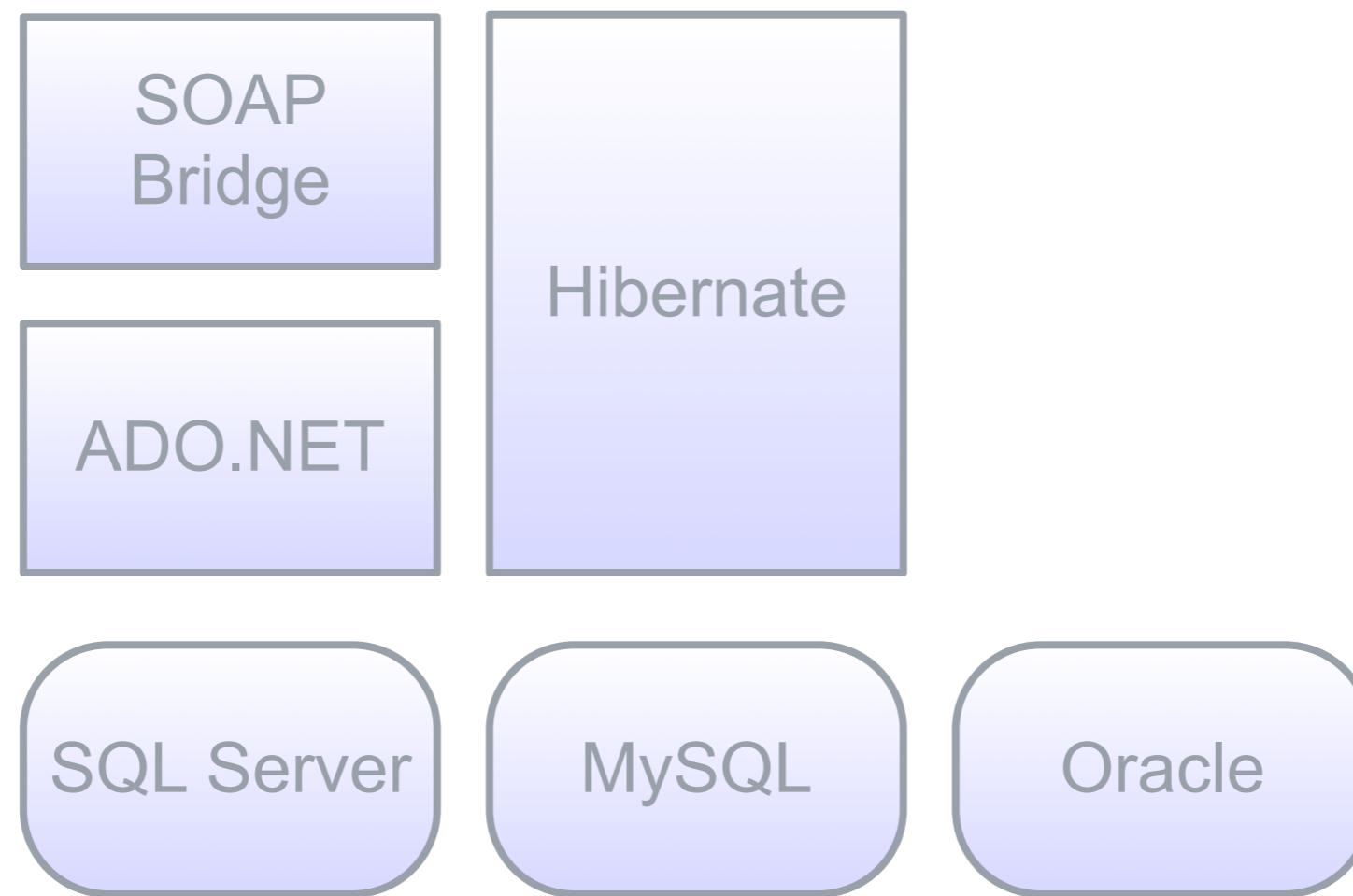
Fighting Bullshit Architecture by Example



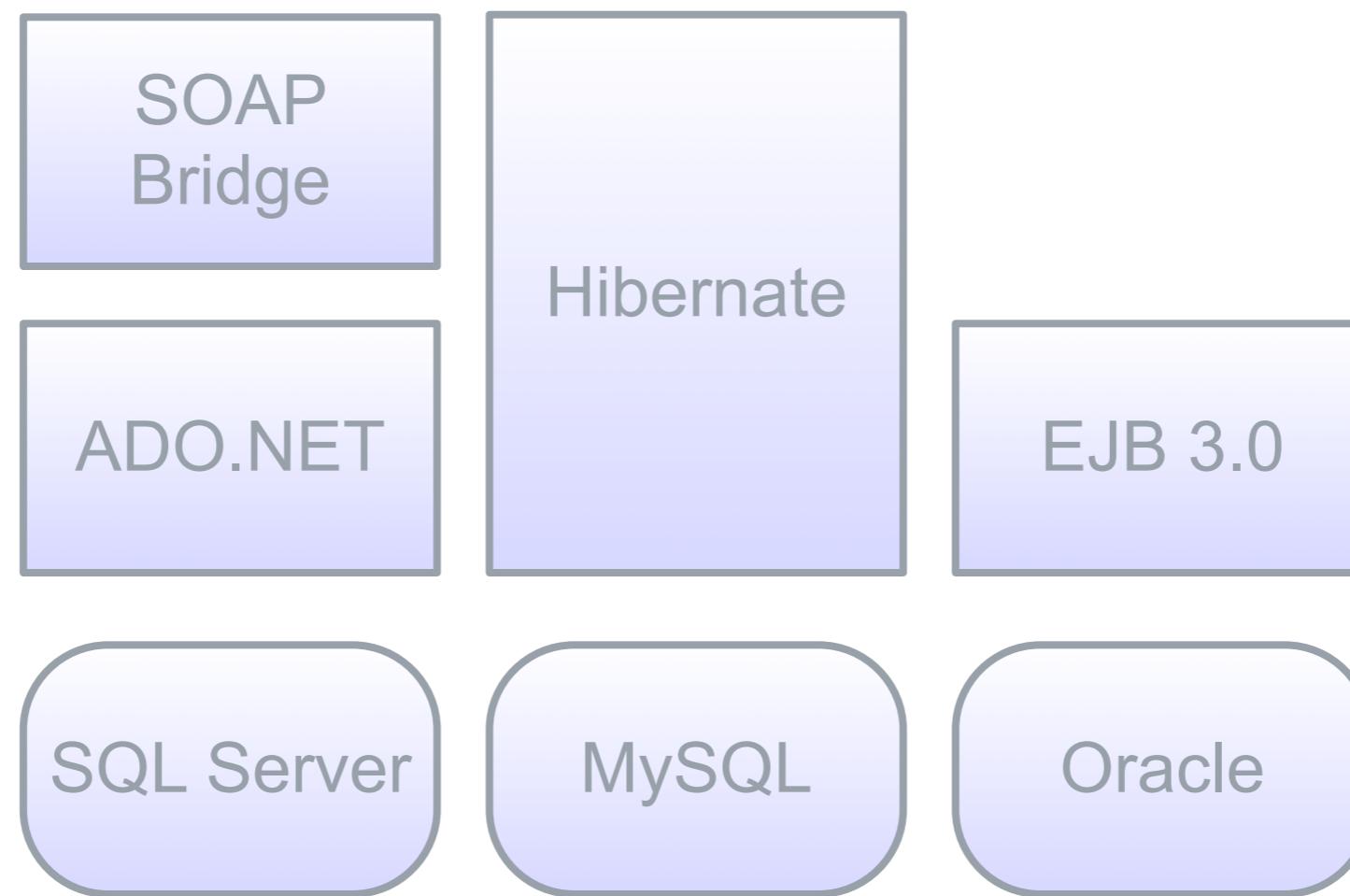
Fighting Bullshit Architecture by Example



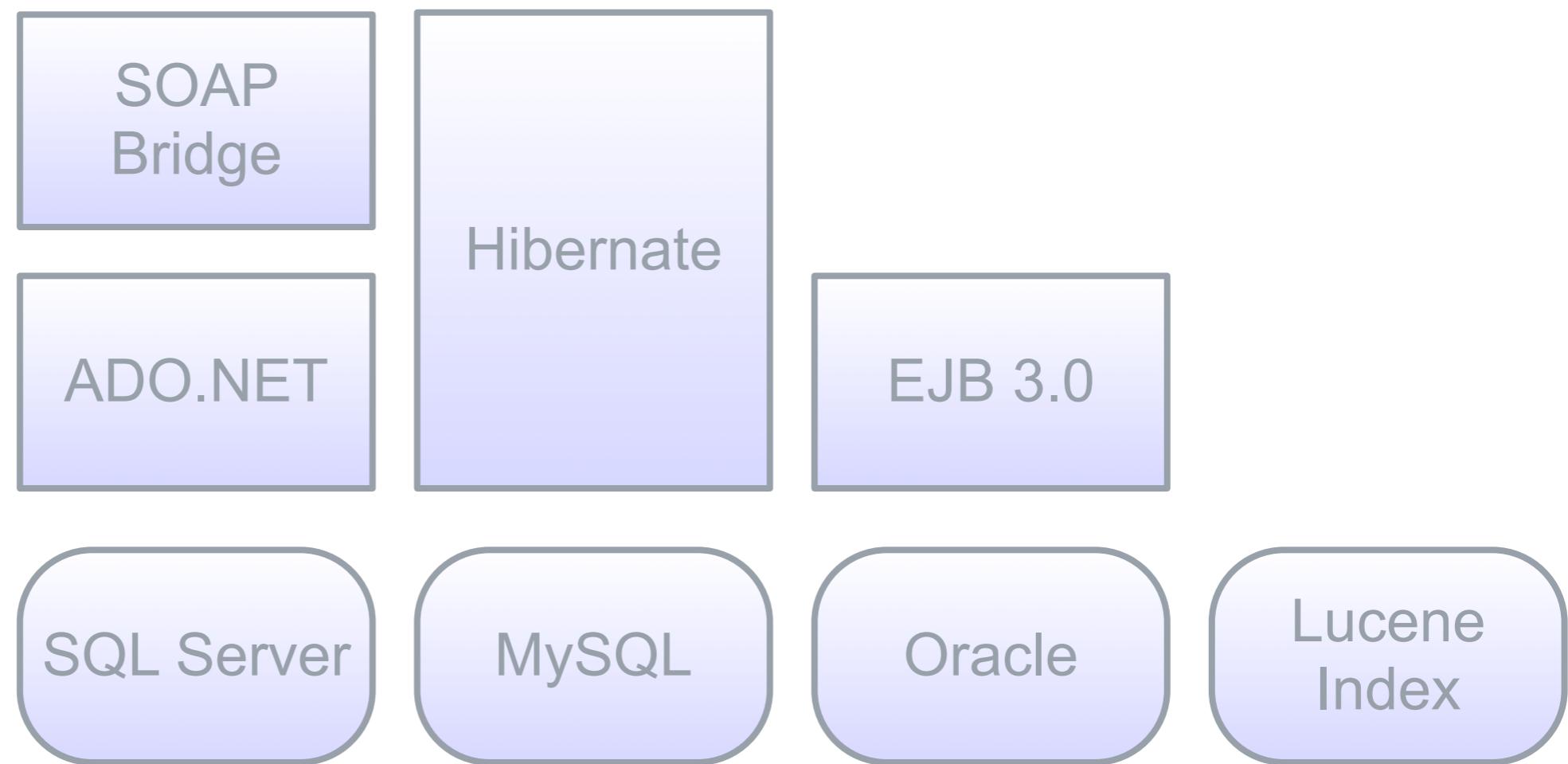
Fighting Bullshit Architecture by Example



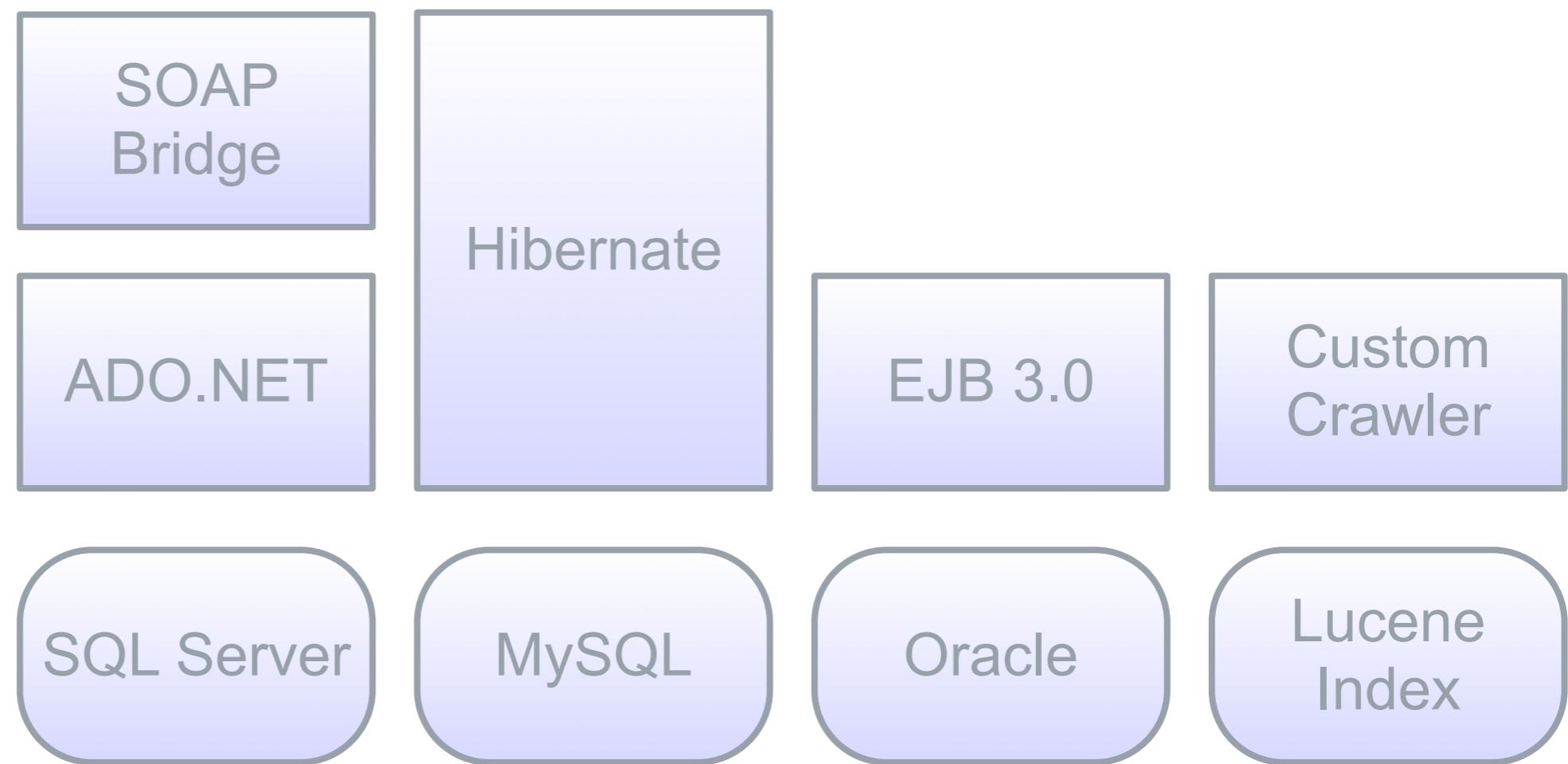
Fighting Bullshit Architecture by Example



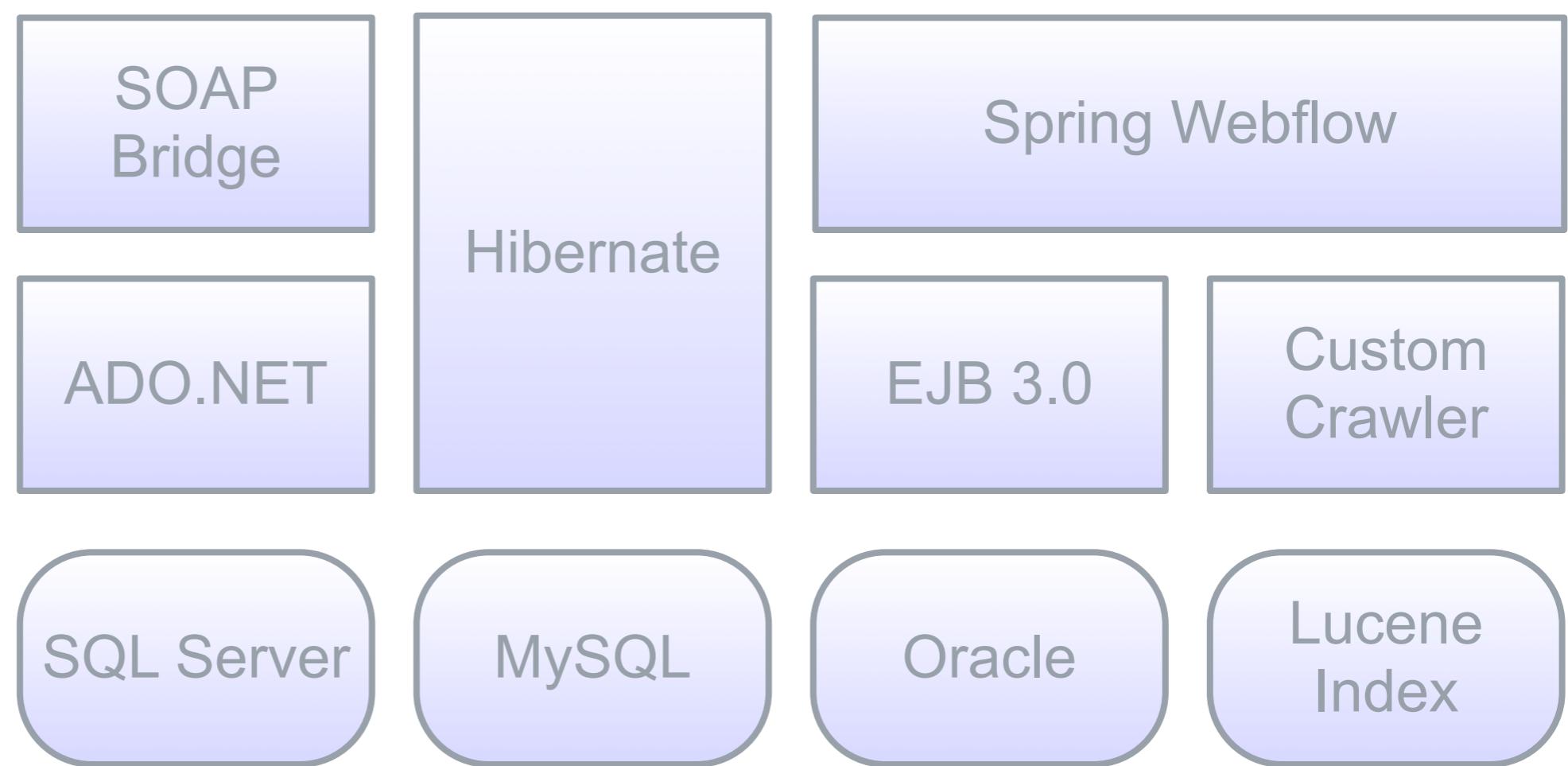
Fighting Bullshit Architecture by Example



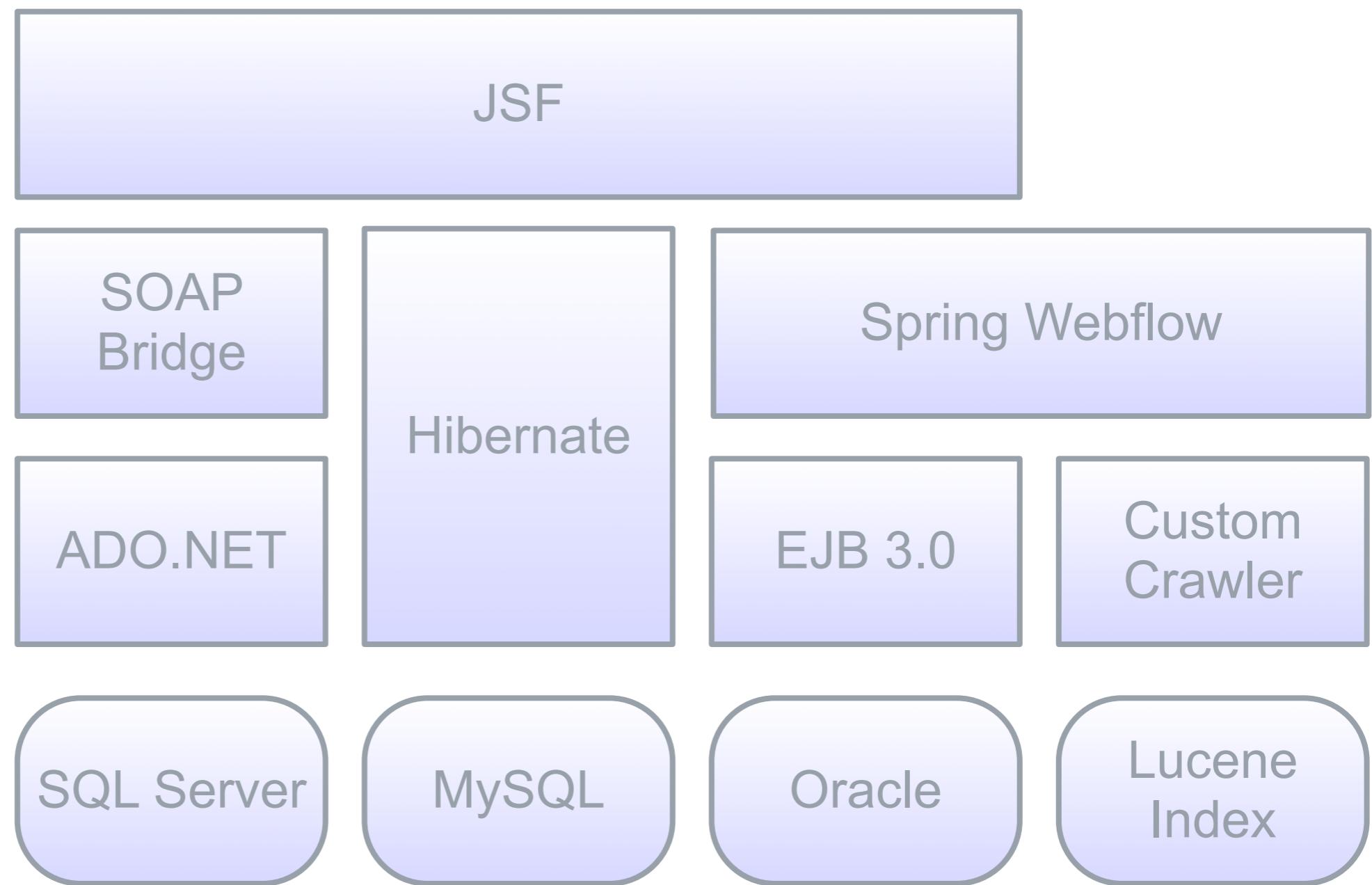
Fighting Bullshit Architecture by Example



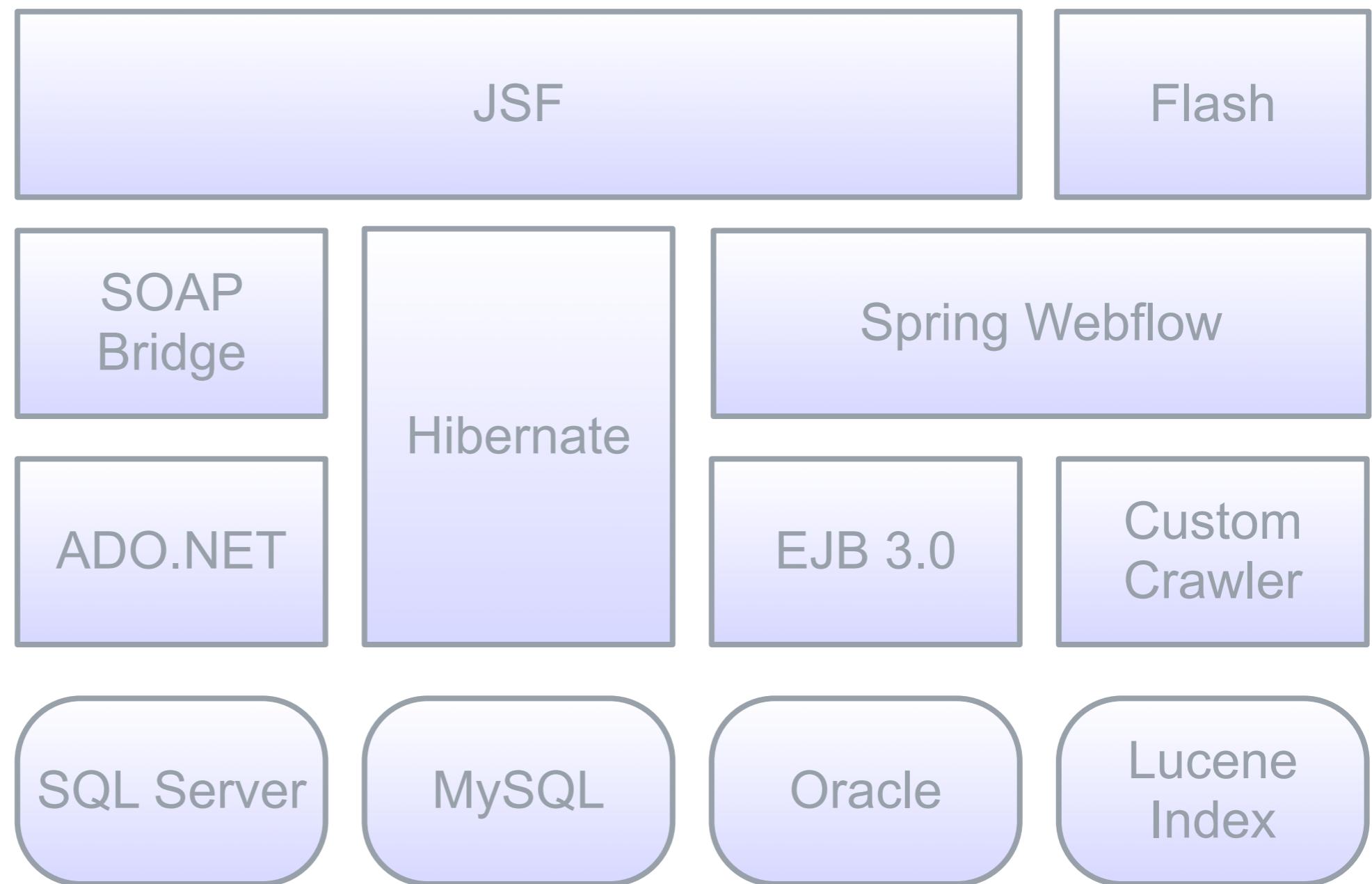
Fighting Bullshit Architecture by Example



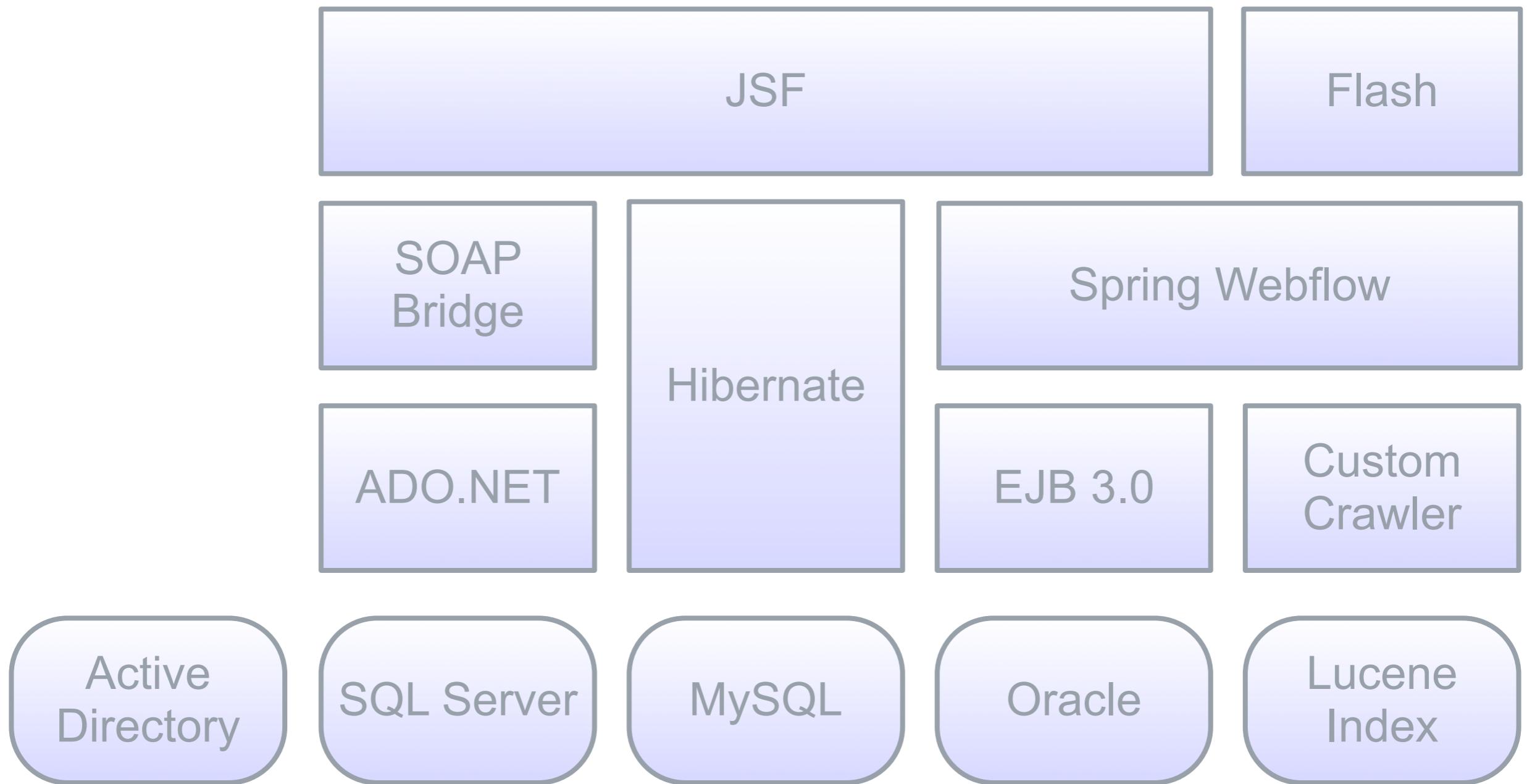
Fighting Bullshit Architecture by Example



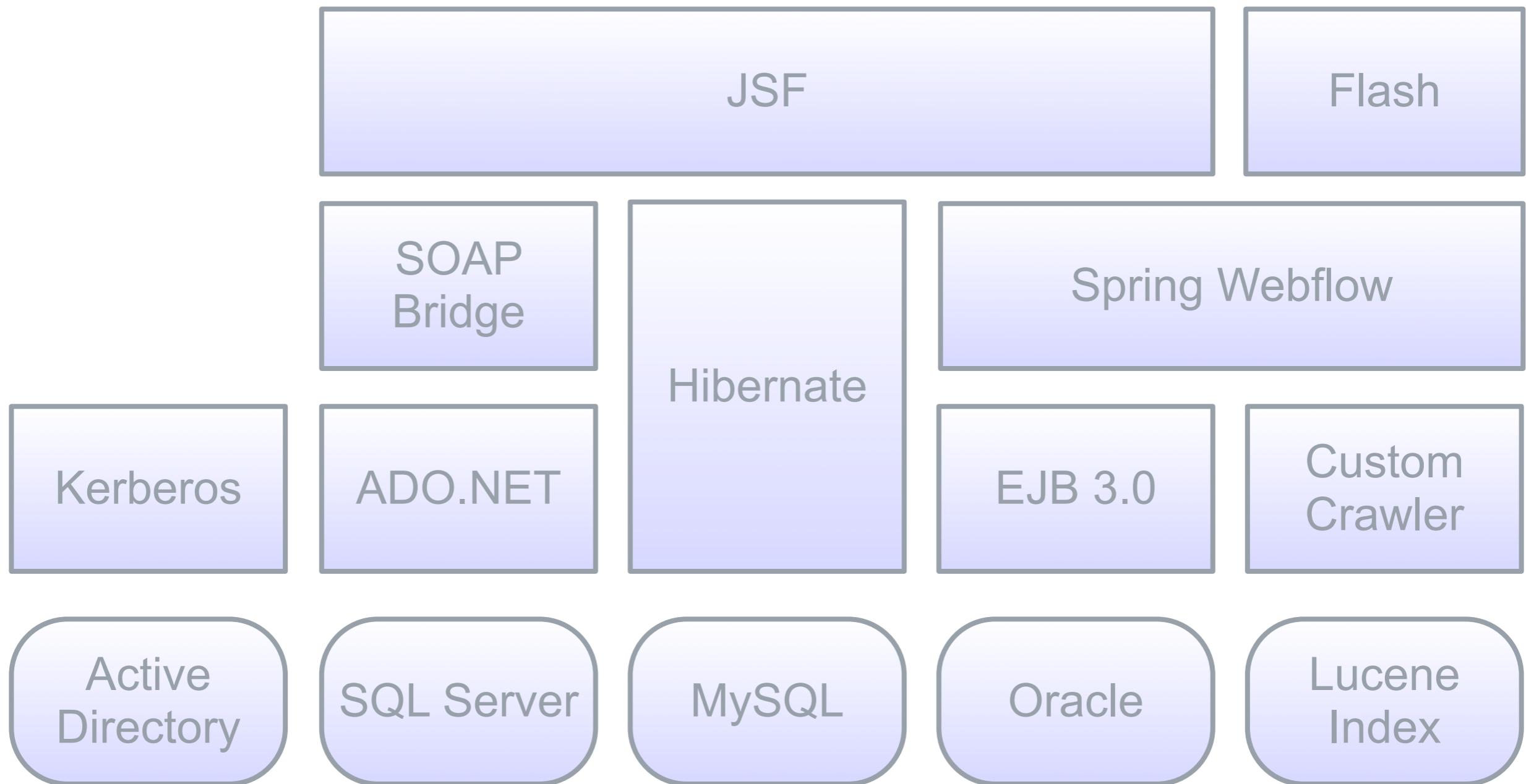
Fighting Bullshit Architecture by Example



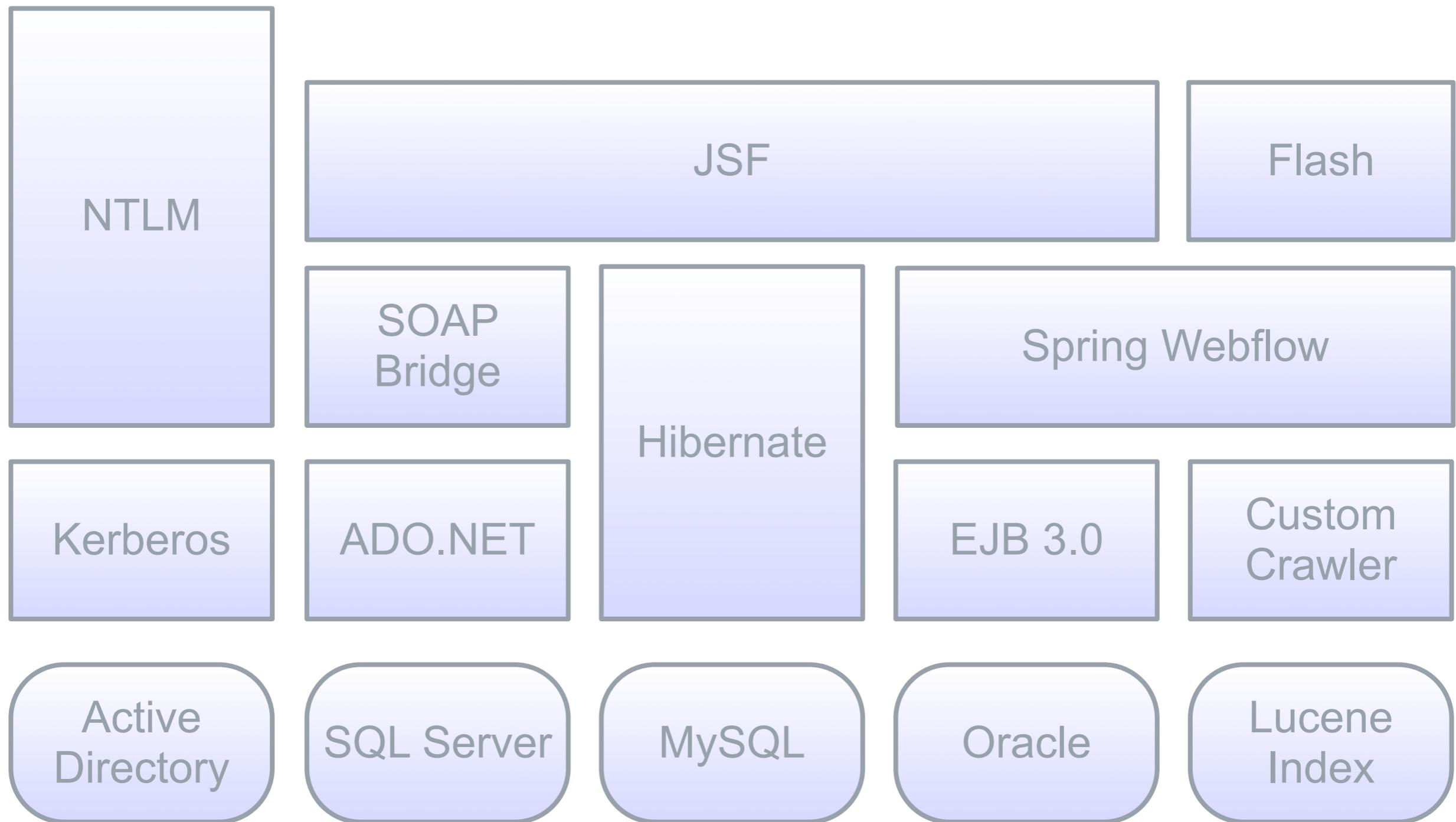
Fighting Bullshit Architecture by Example



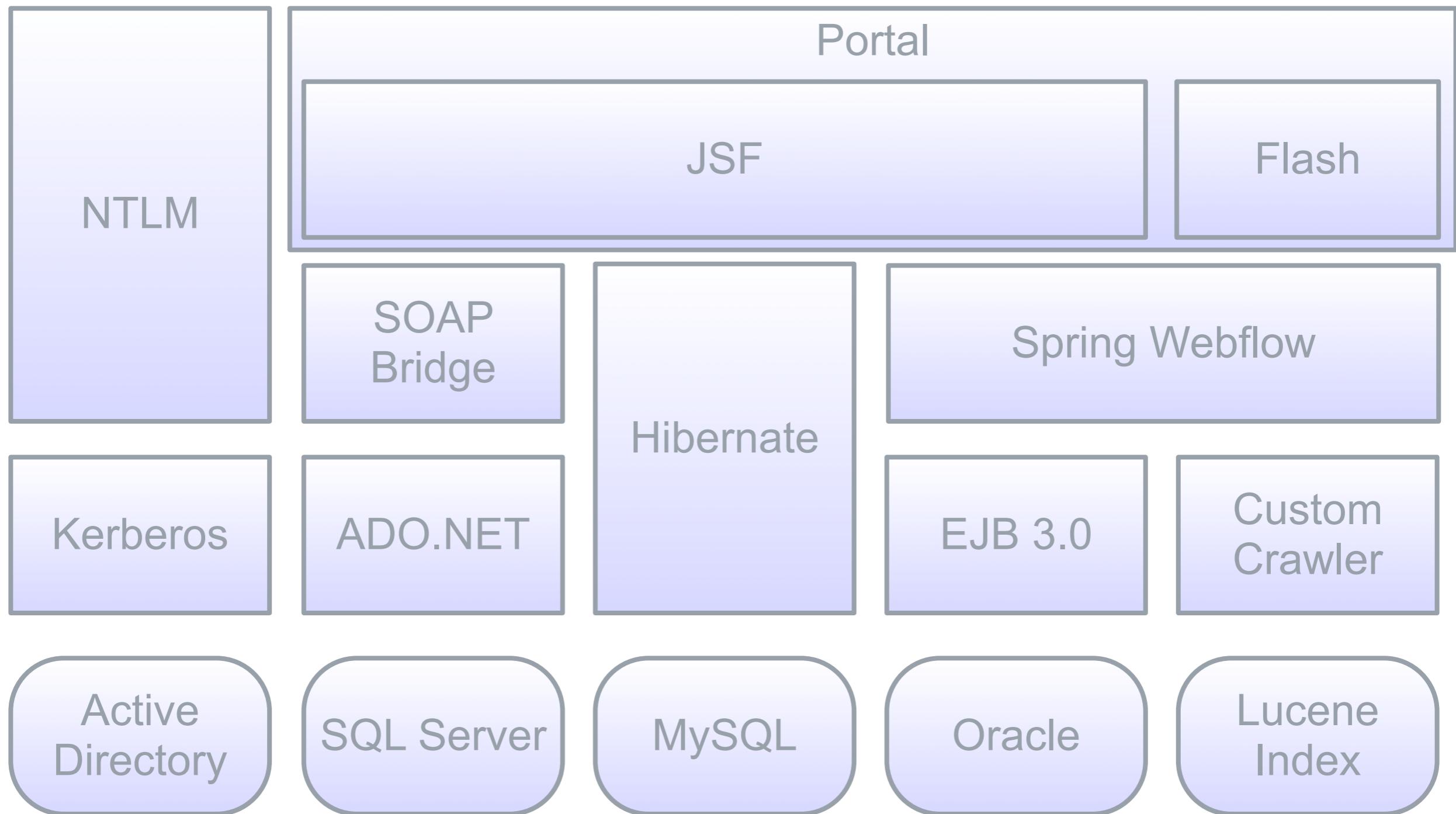
Fighting Bullshit Architecture by Example



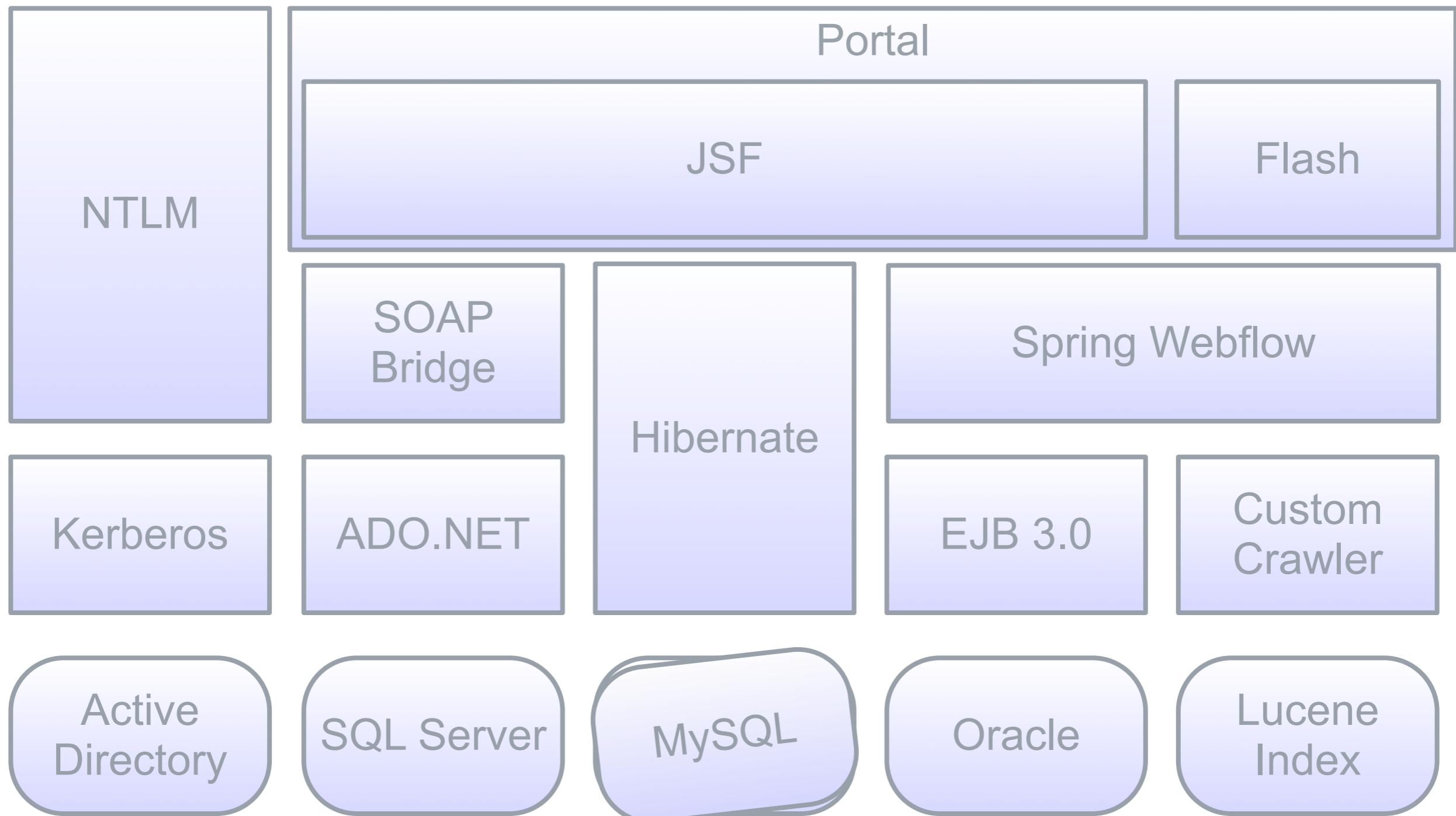
Fighting Bullshit Architecture by Example



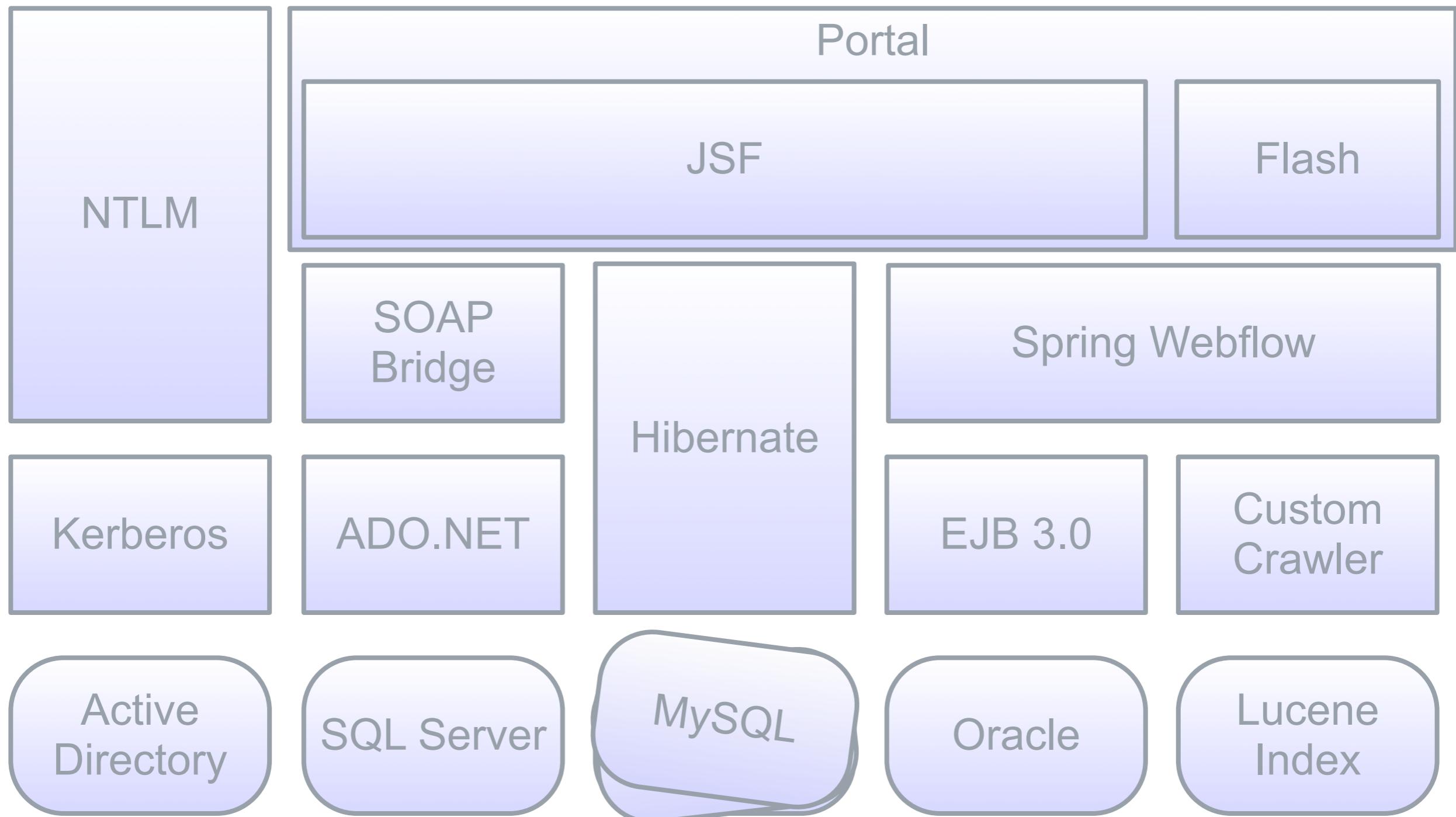
Fighting Bullshit Architecture by Example



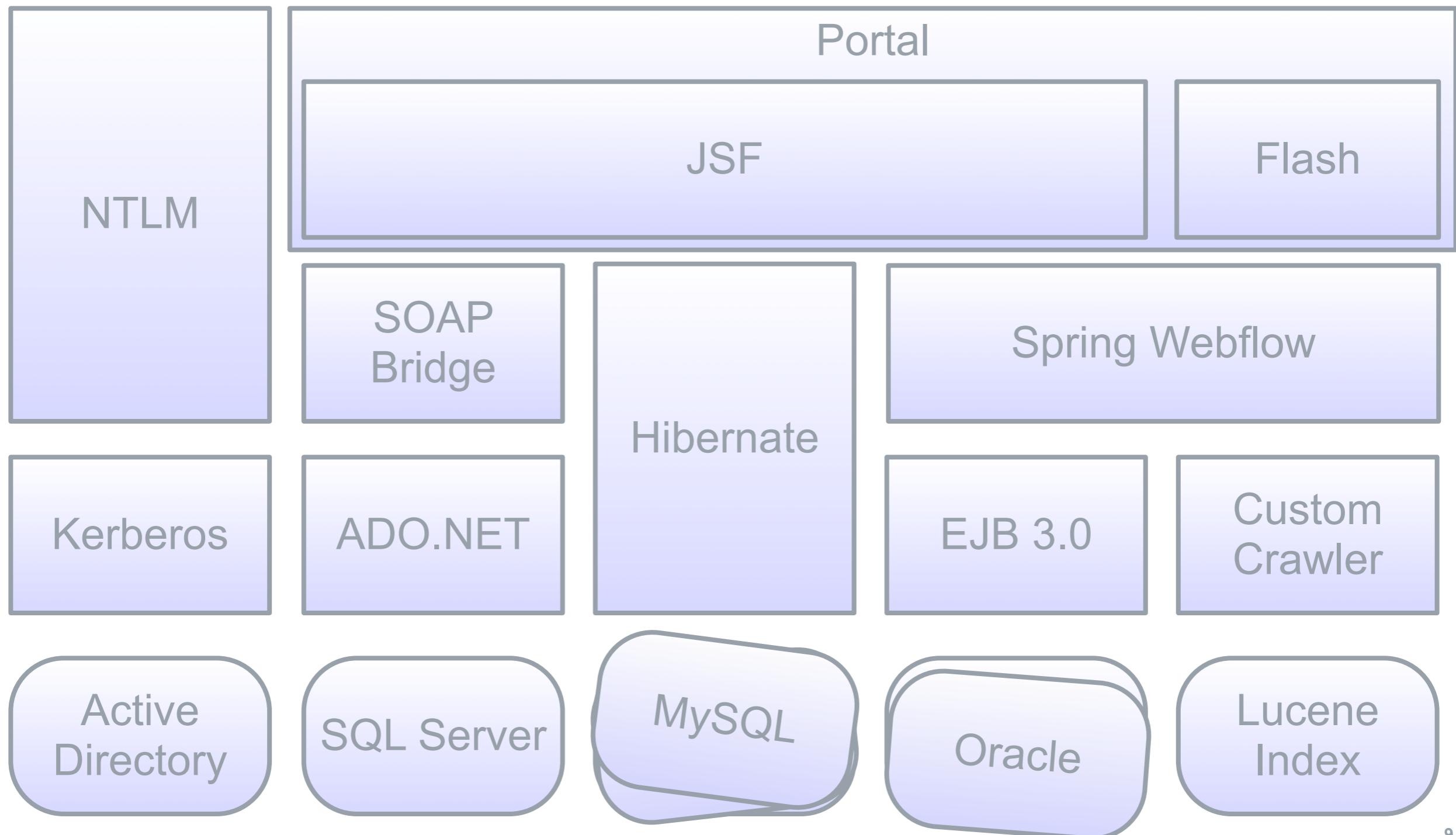
Fighting Bullshit Architecture by Example



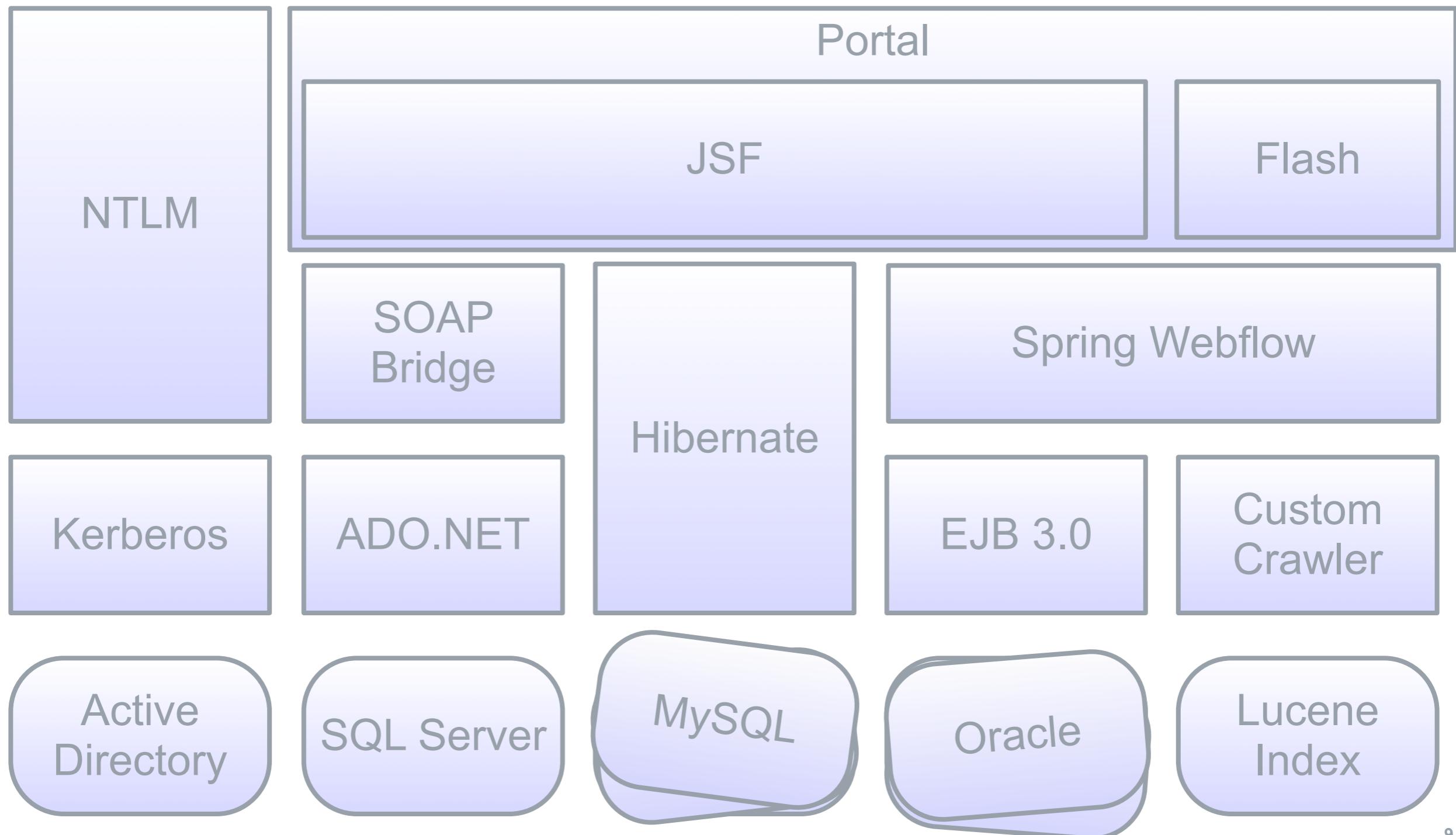
Fighting Bullshit Architecture by Example



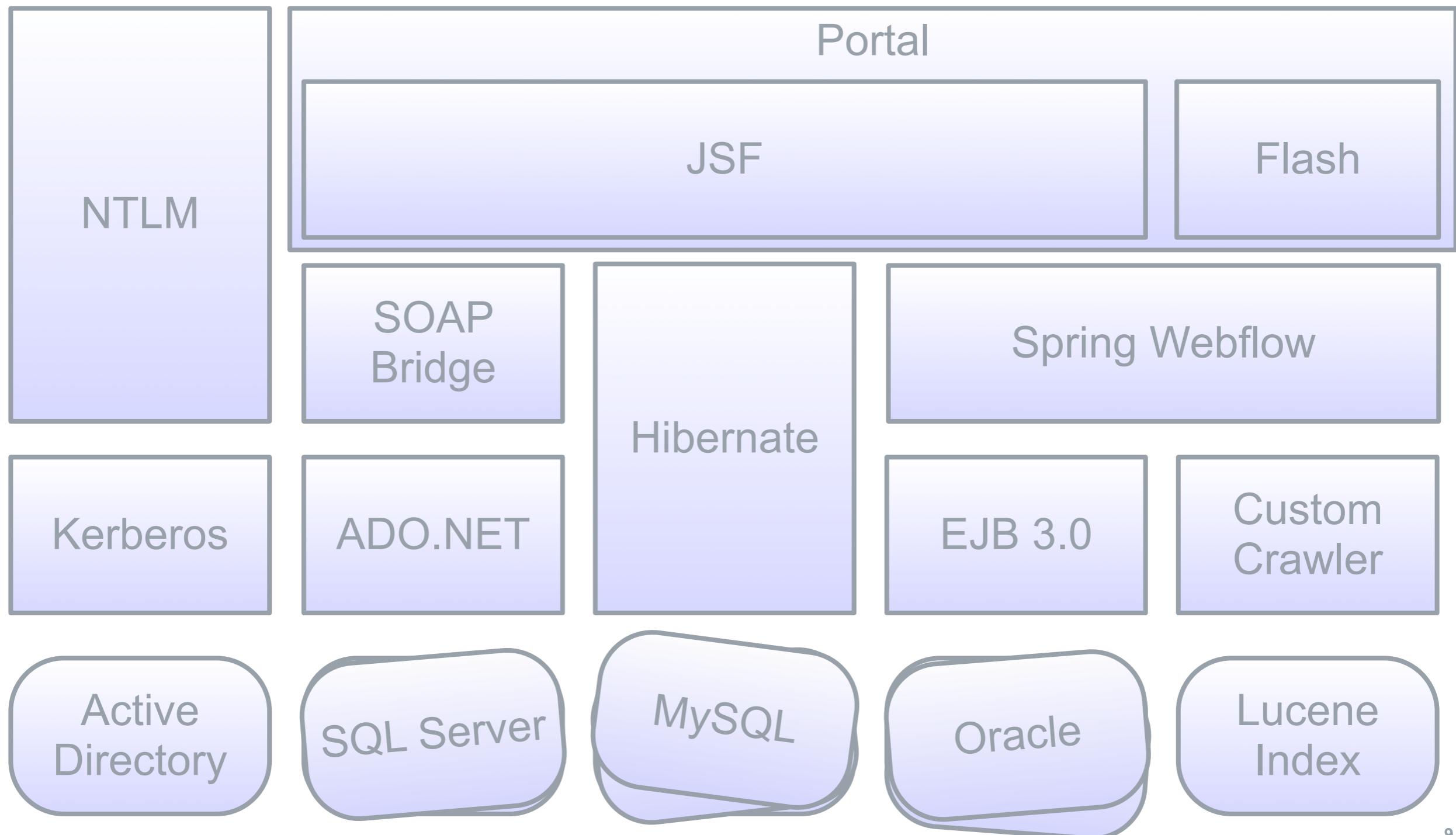
Fighting Bullshit Architecture by Example



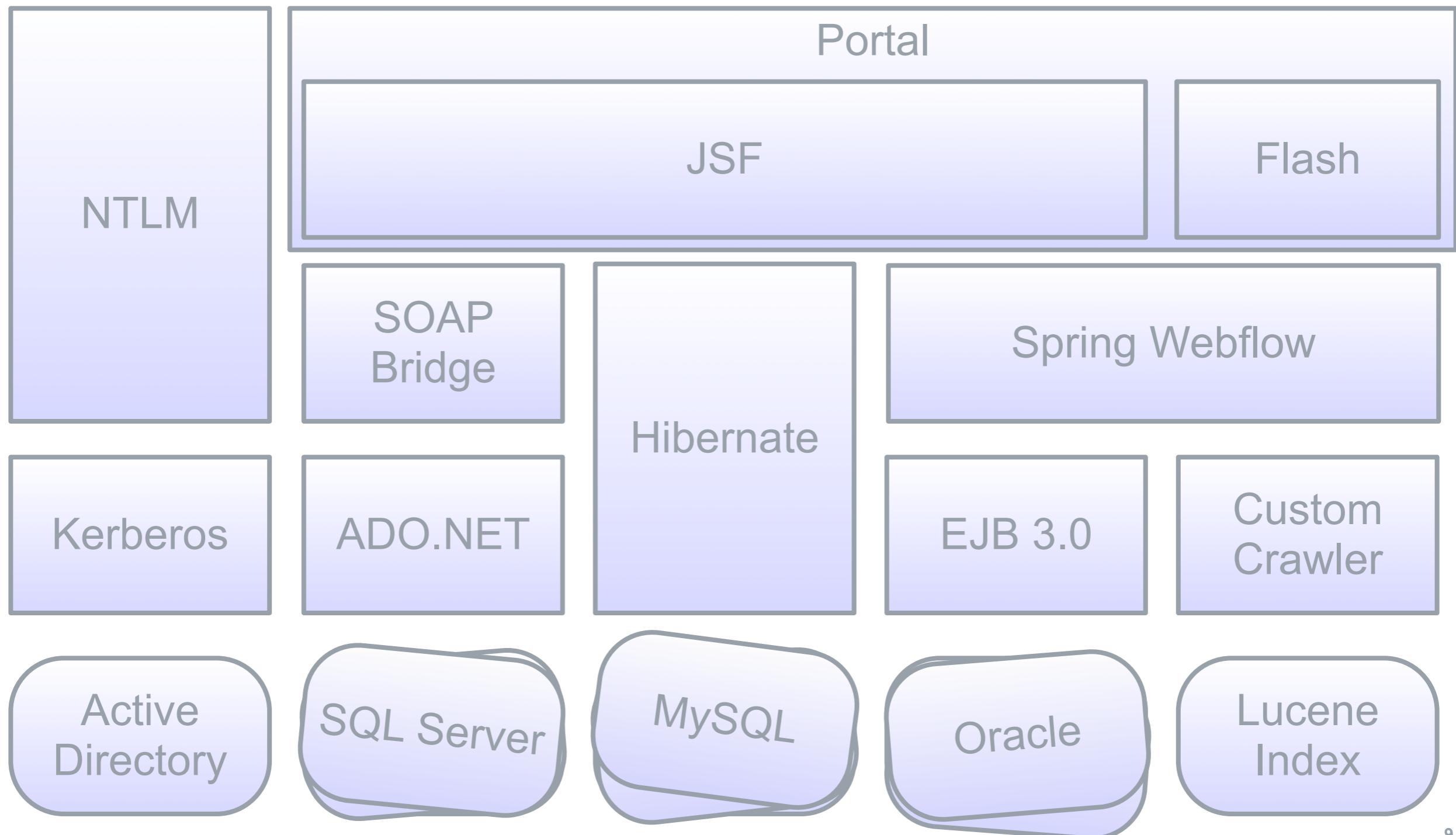
Fighting Bullshit Architecture by Example



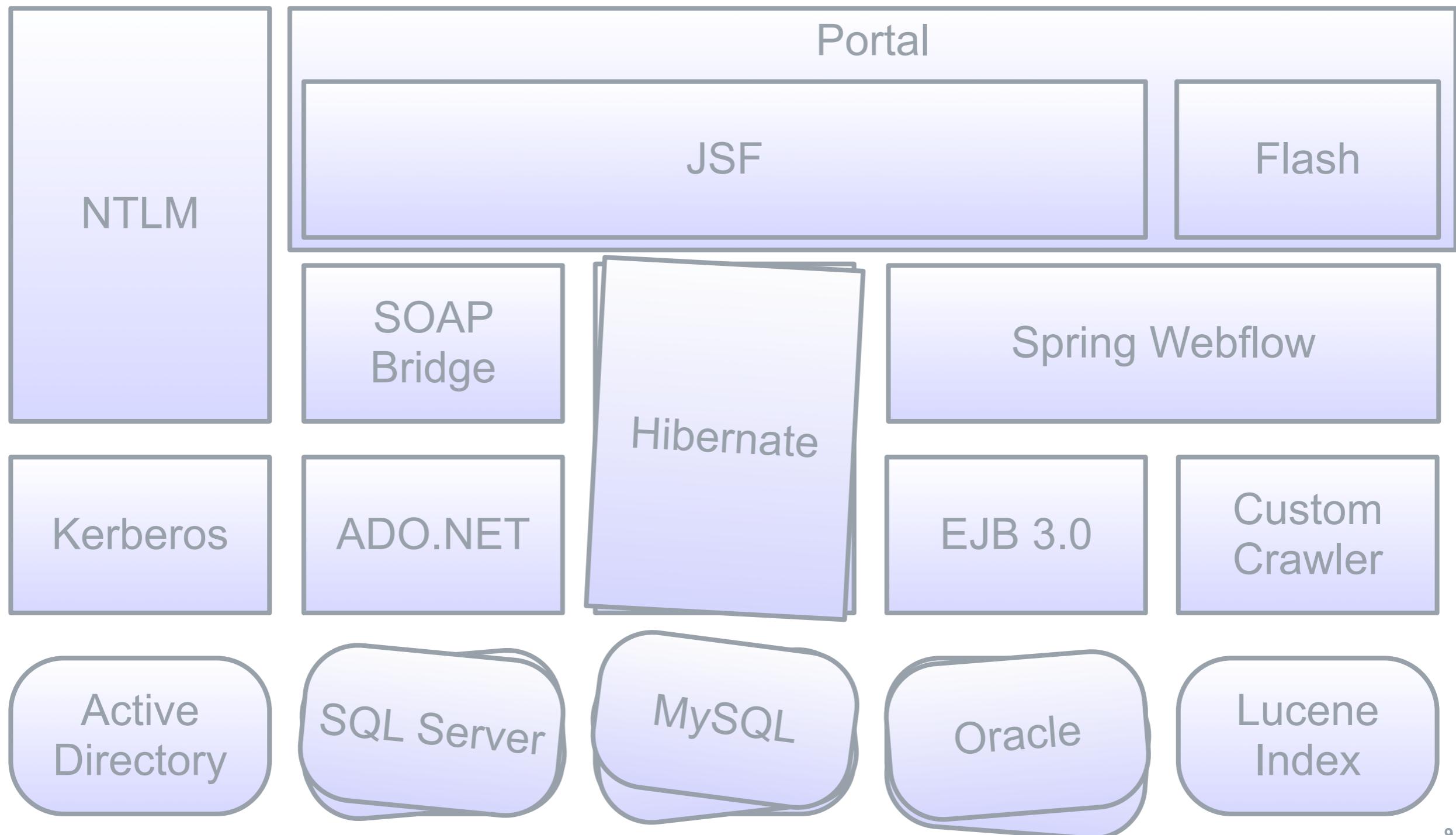
Fighting Bullshit Architecture by Example



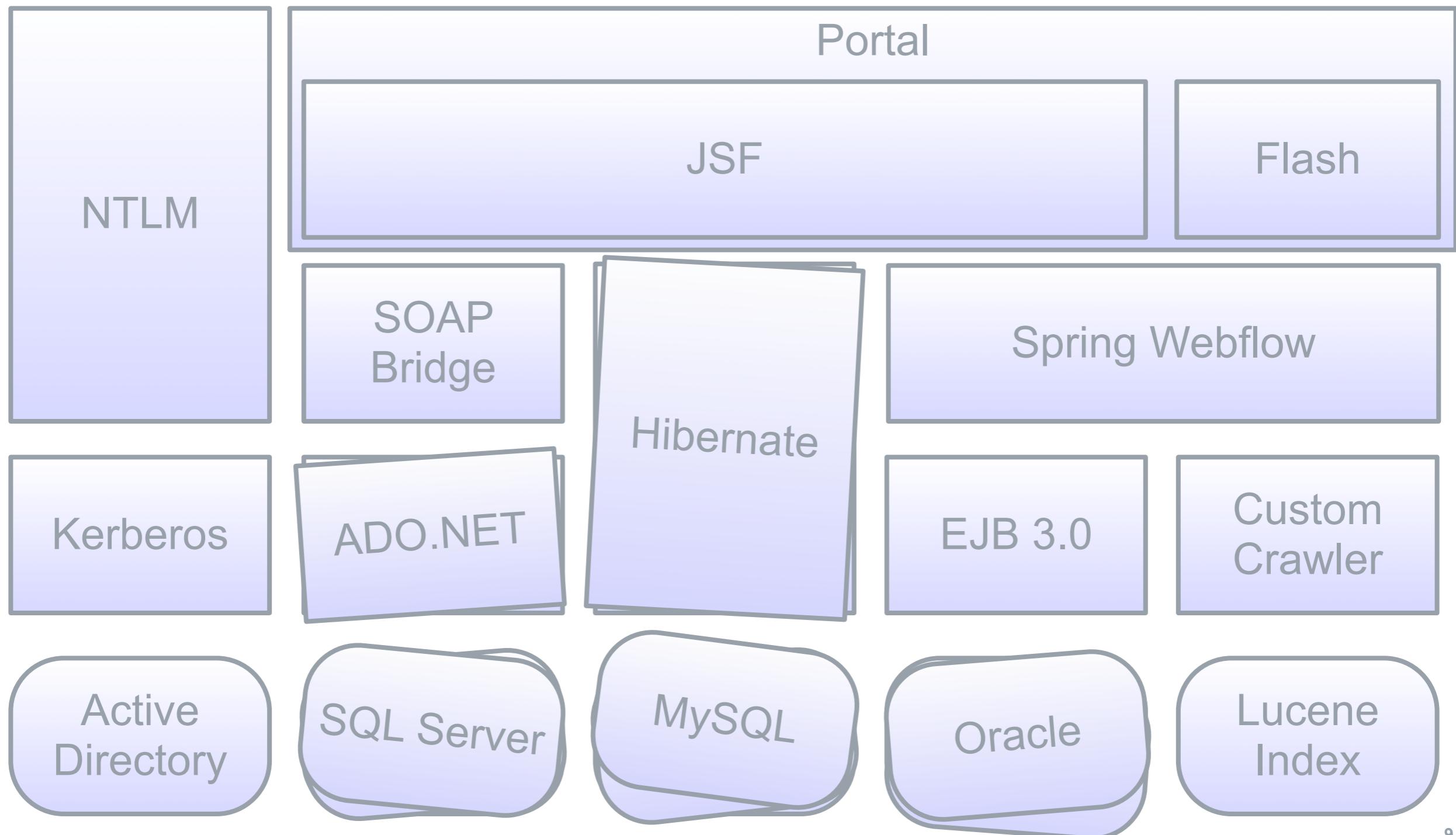
Fighting Bullshit Architecture by Example



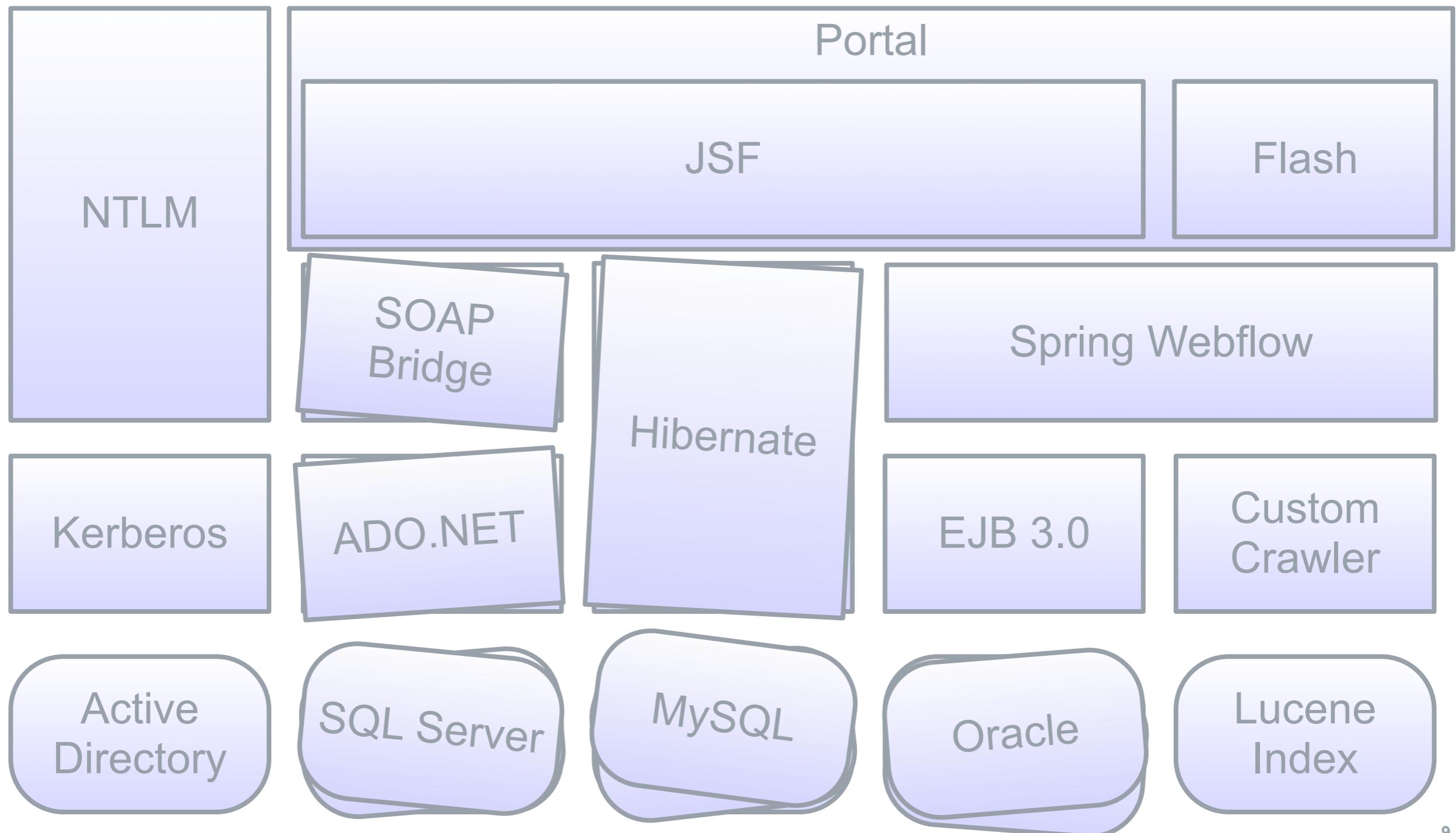
Fighting Bullshit Architecture by Example



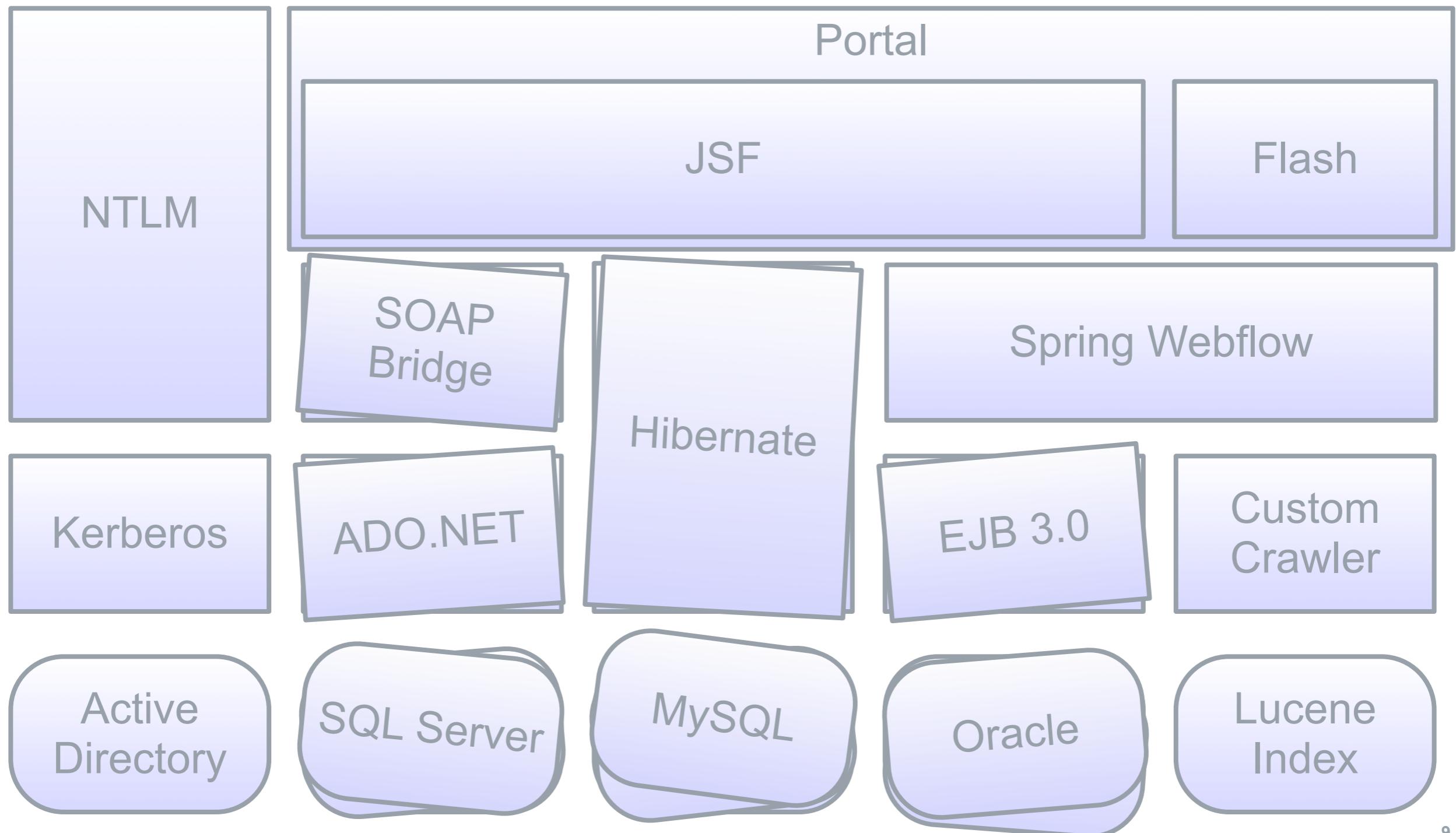
Fighting Bullshit Architecture by Example



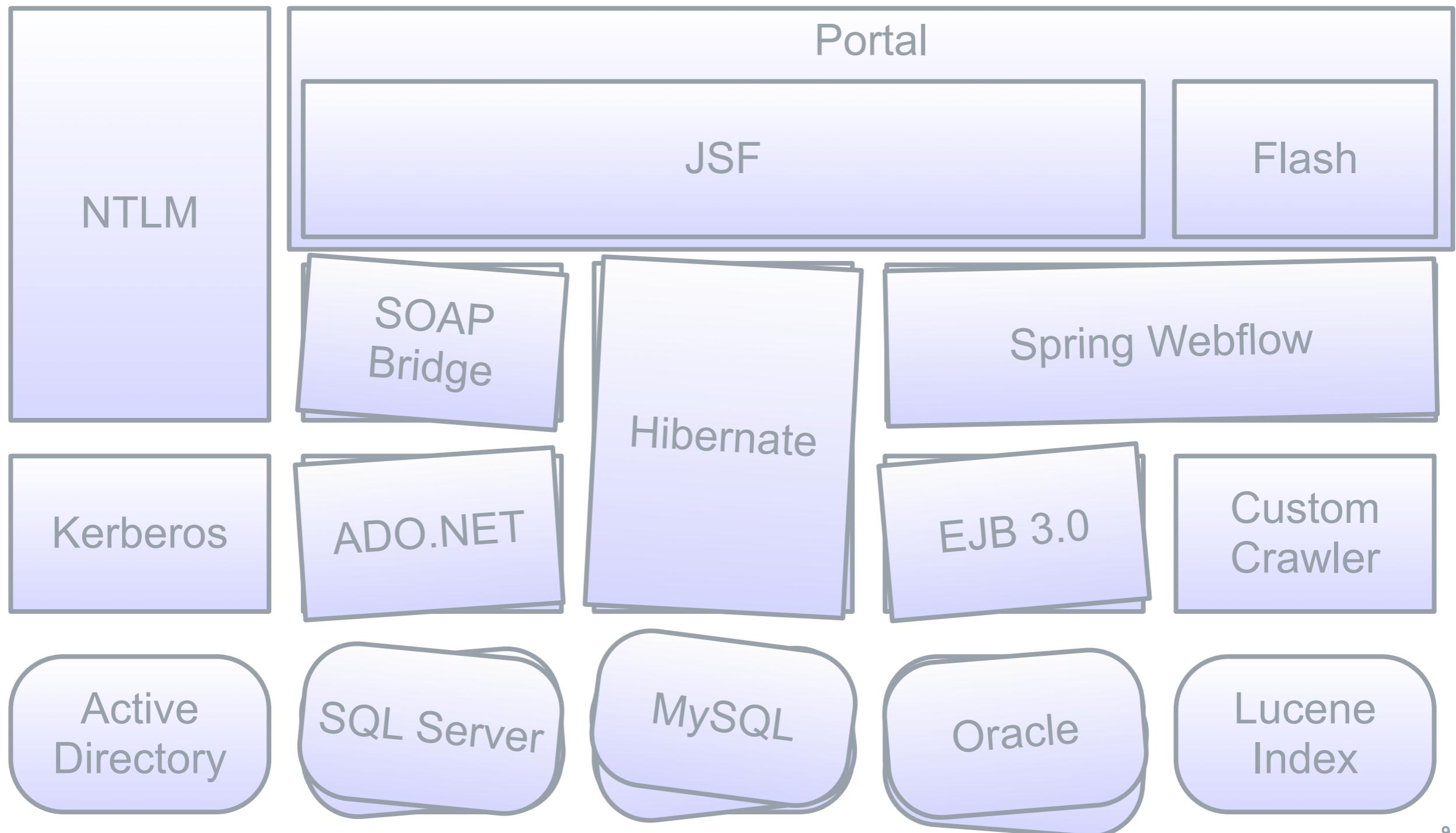
Fighting Bullshit Architecture by Example



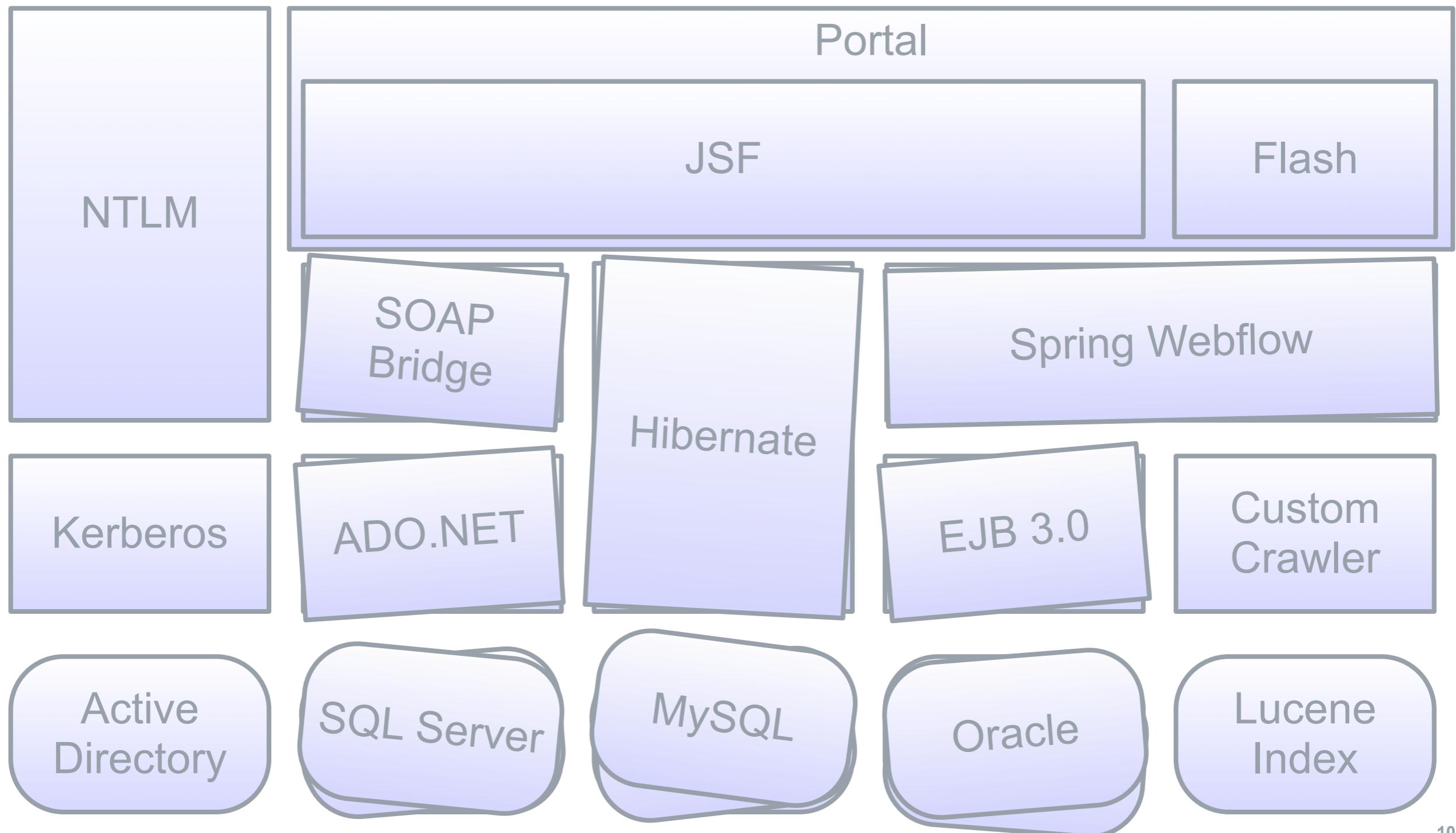
Fighting Bullshit Architecture by Example



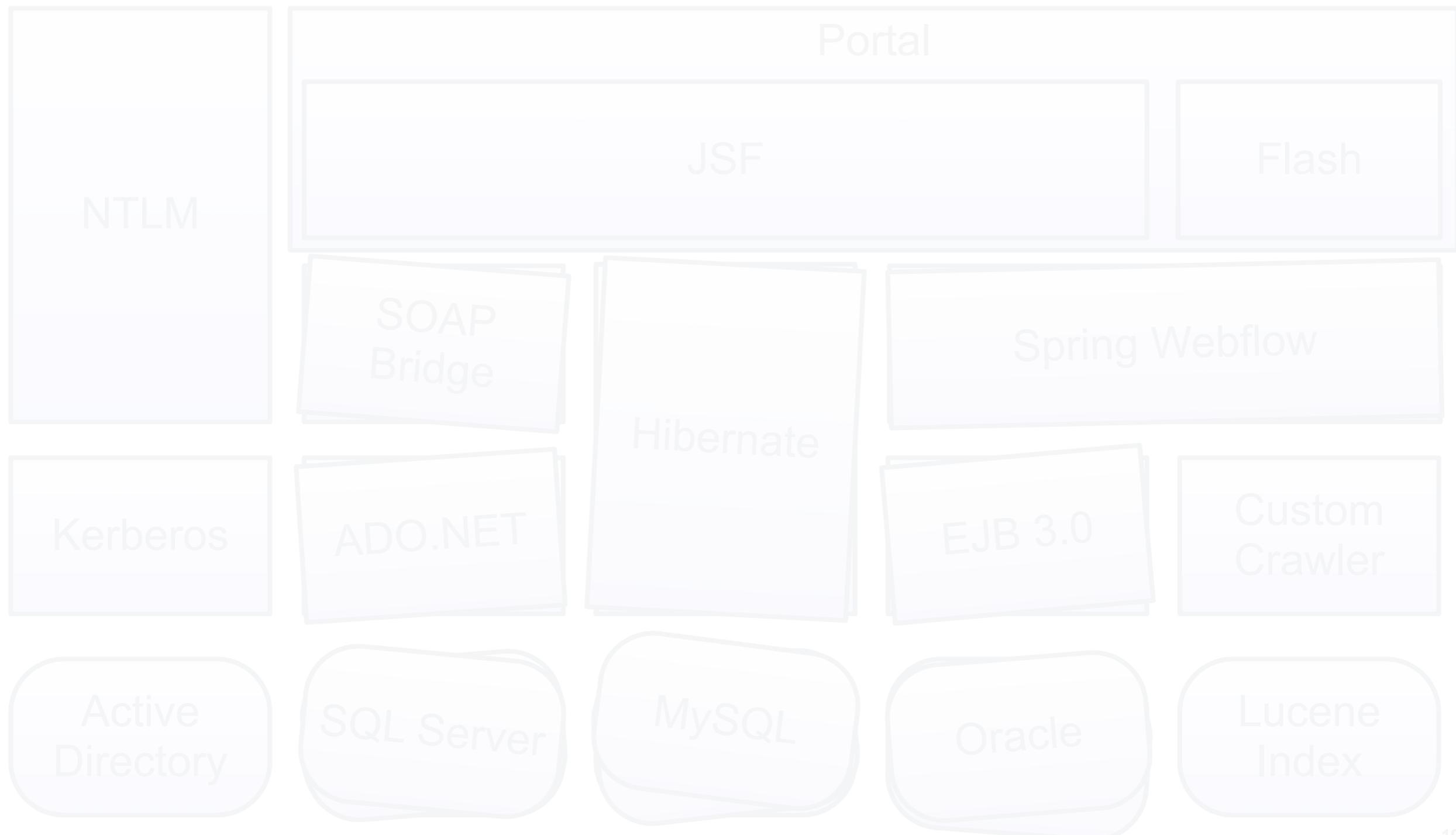
Fighting Bullshit Architecture by Example



Fighting Bullshit Architecture by Example

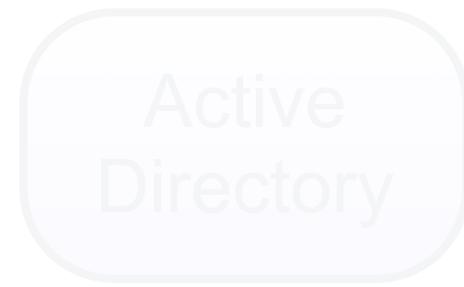
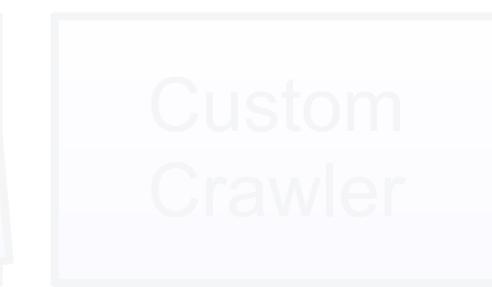
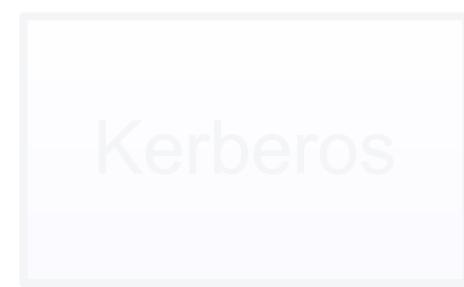


Fighting Bullshit Architecture by Example

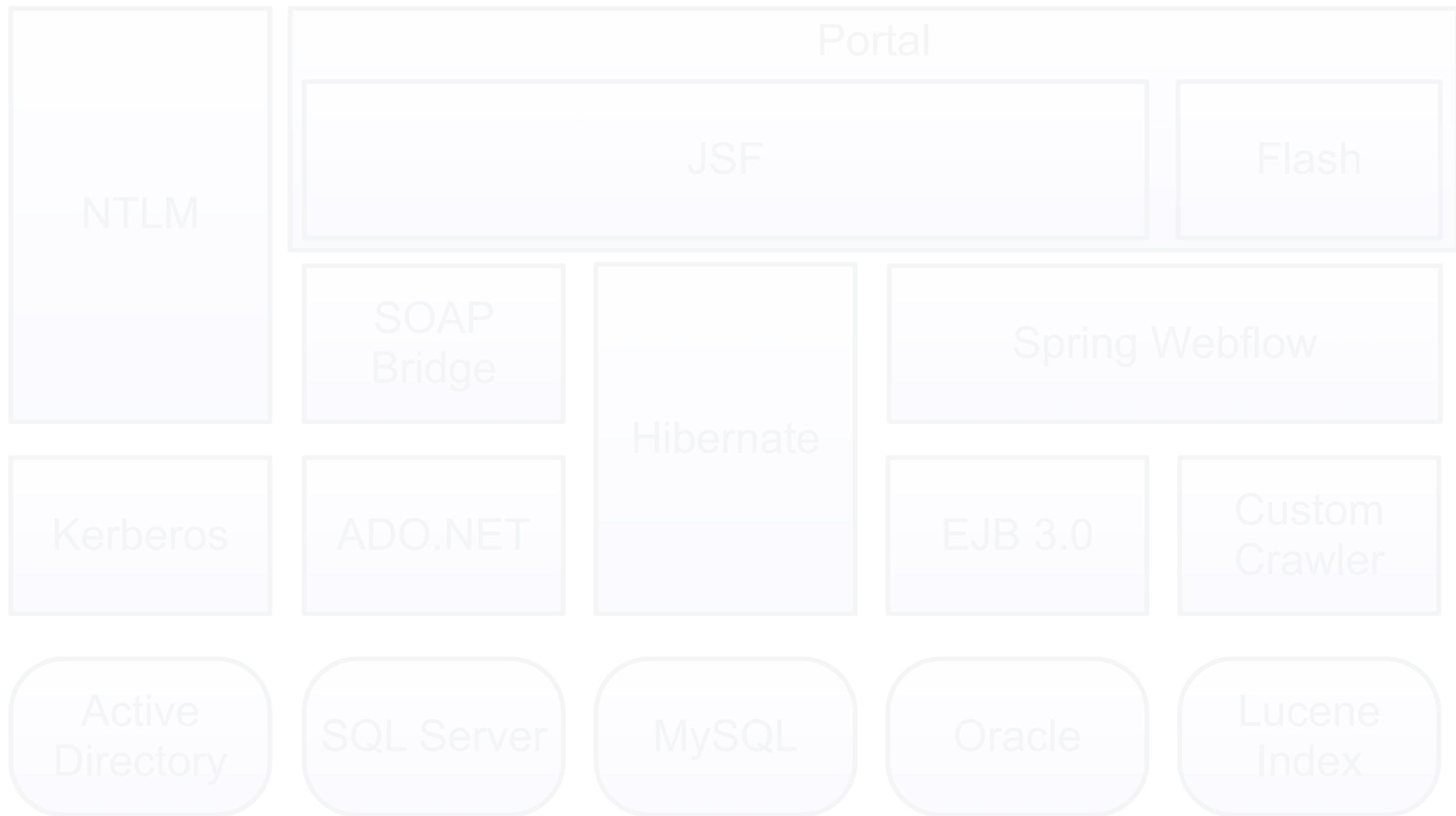


Fighting Bullshit Architecture by Example

Can we please
create an AIR app to
get it to the
desktop?



Fighting Bullshit Architecture by Example



Fighting Bullshit Architecture by Example

Wait, I have some questions

NTLM

SOAP
Bridge

Kerberos

ADO.NET

Hibernate

Spring Webflow

EJB 3.0

Custom
Crawler

Active
Directory

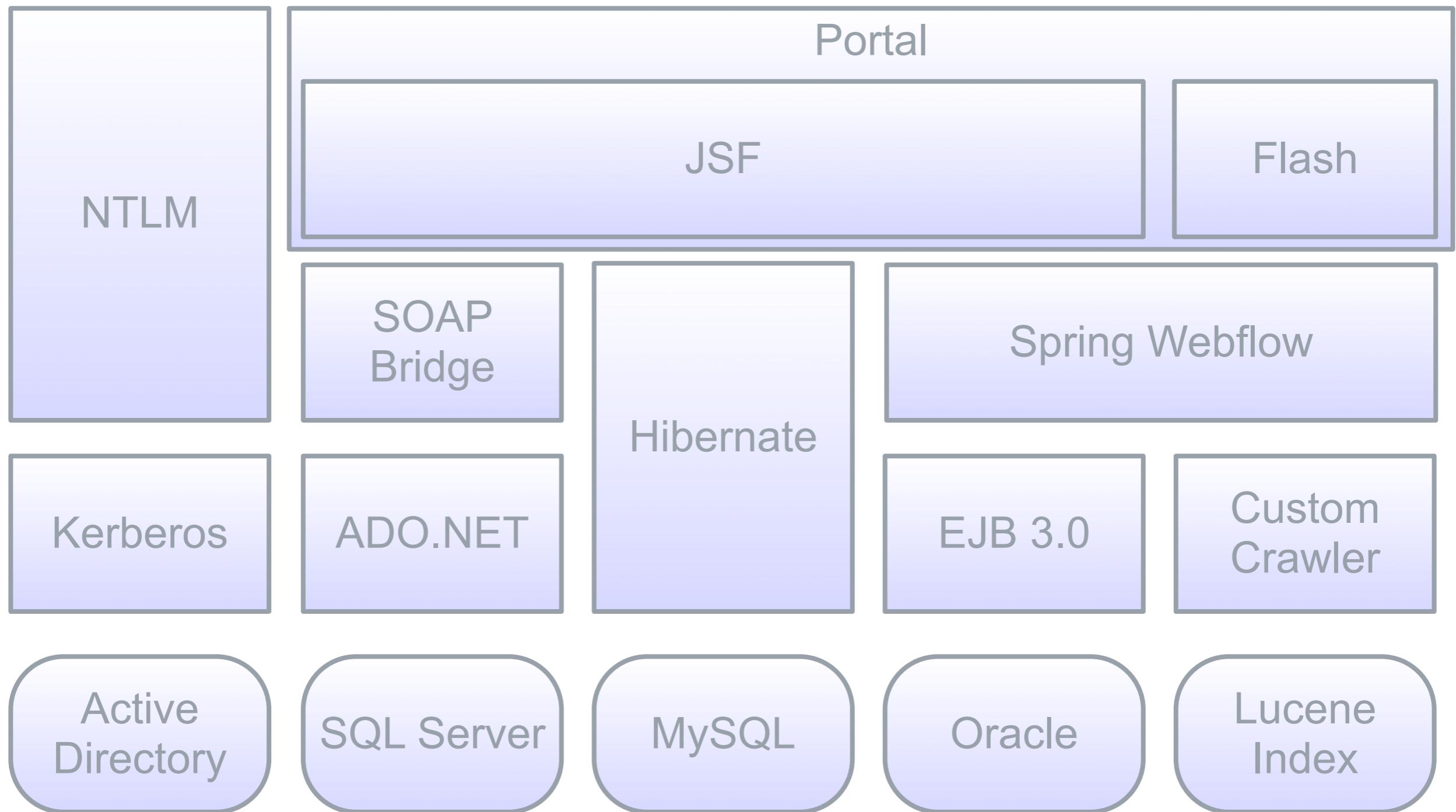
SQL Server

MySQL

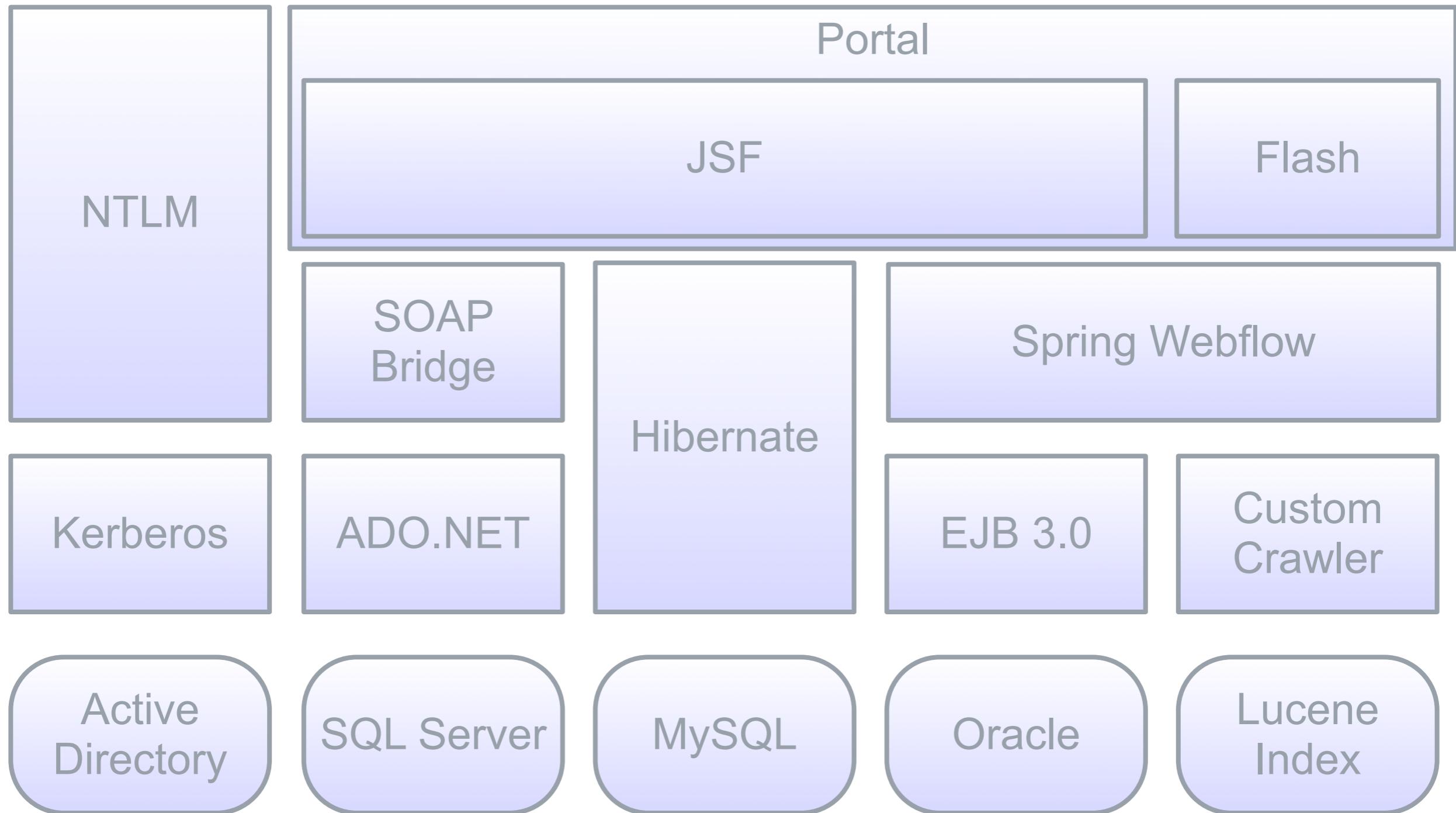
Oracle

Lucene
Index

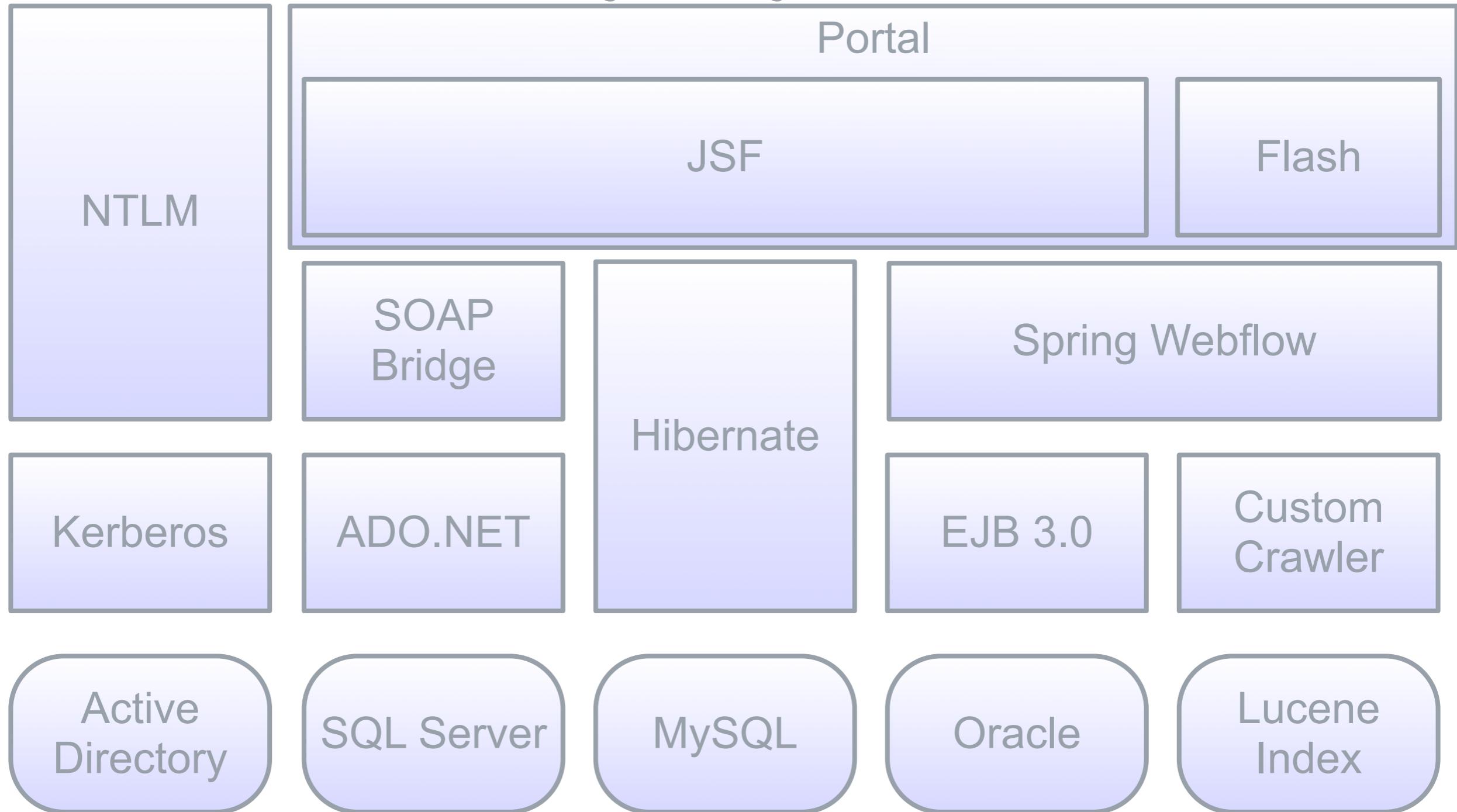
Where is our data stored?



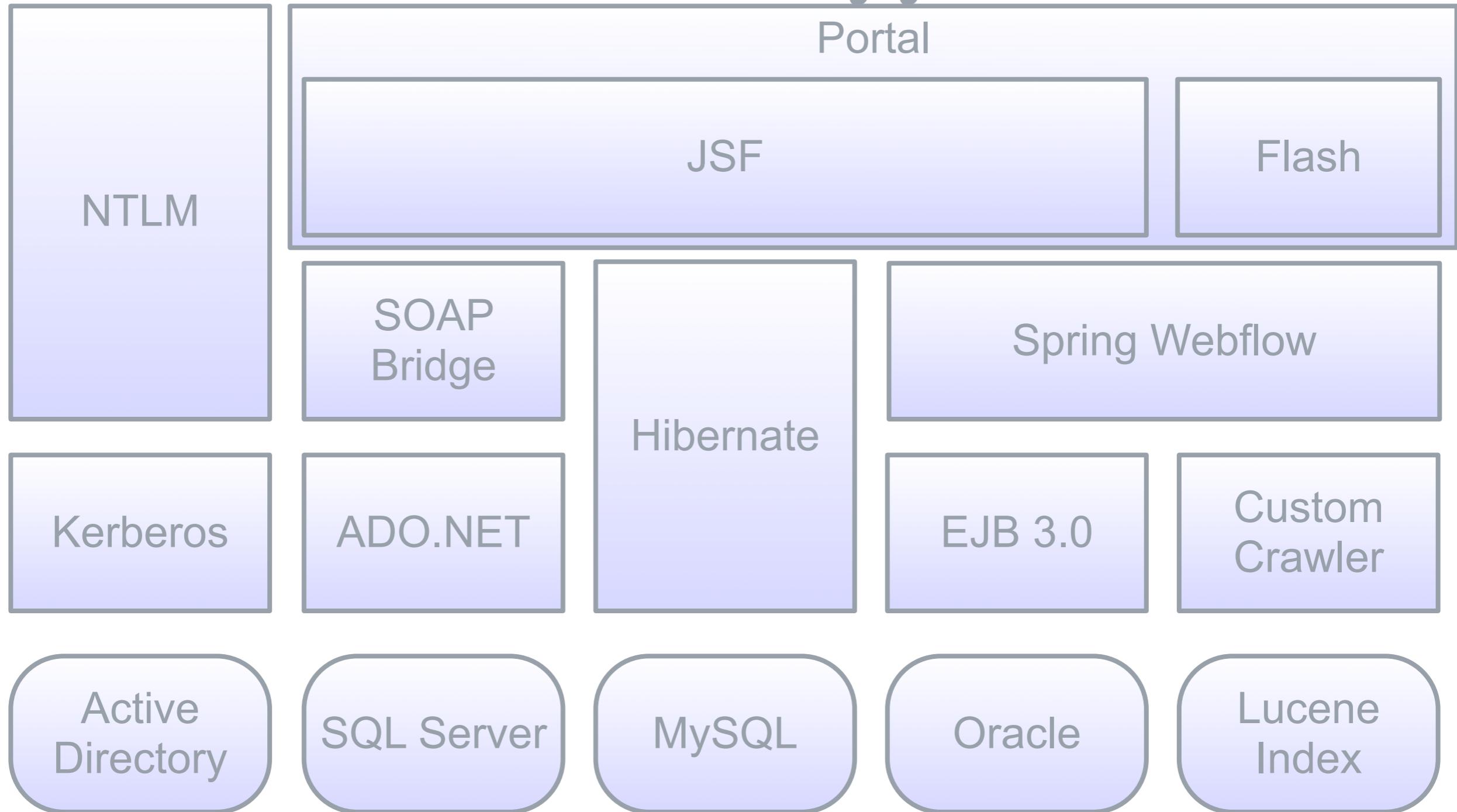
And where is our content? I cannot see it.



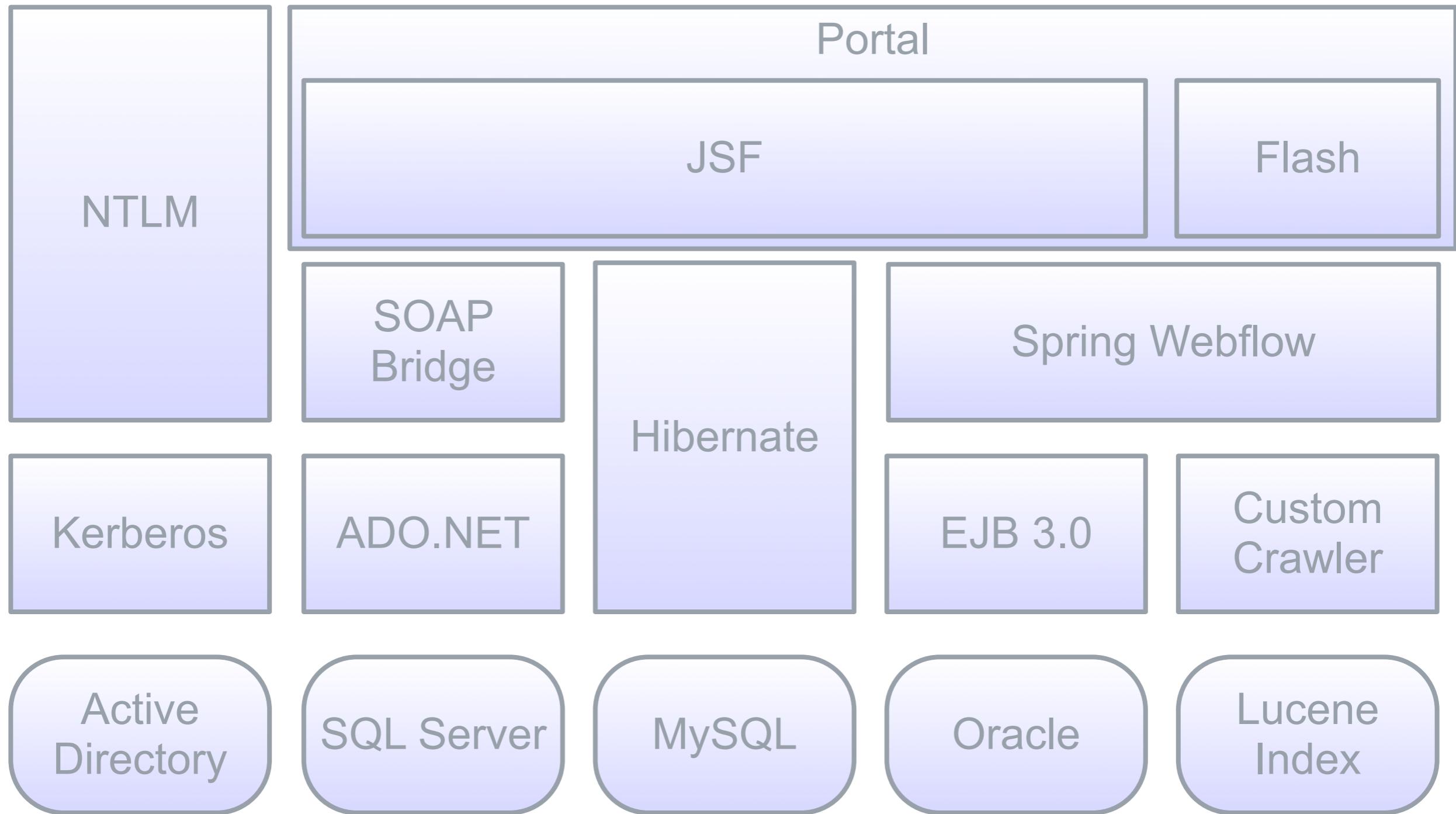
This is a n-tier architecture. But how many layers are there?



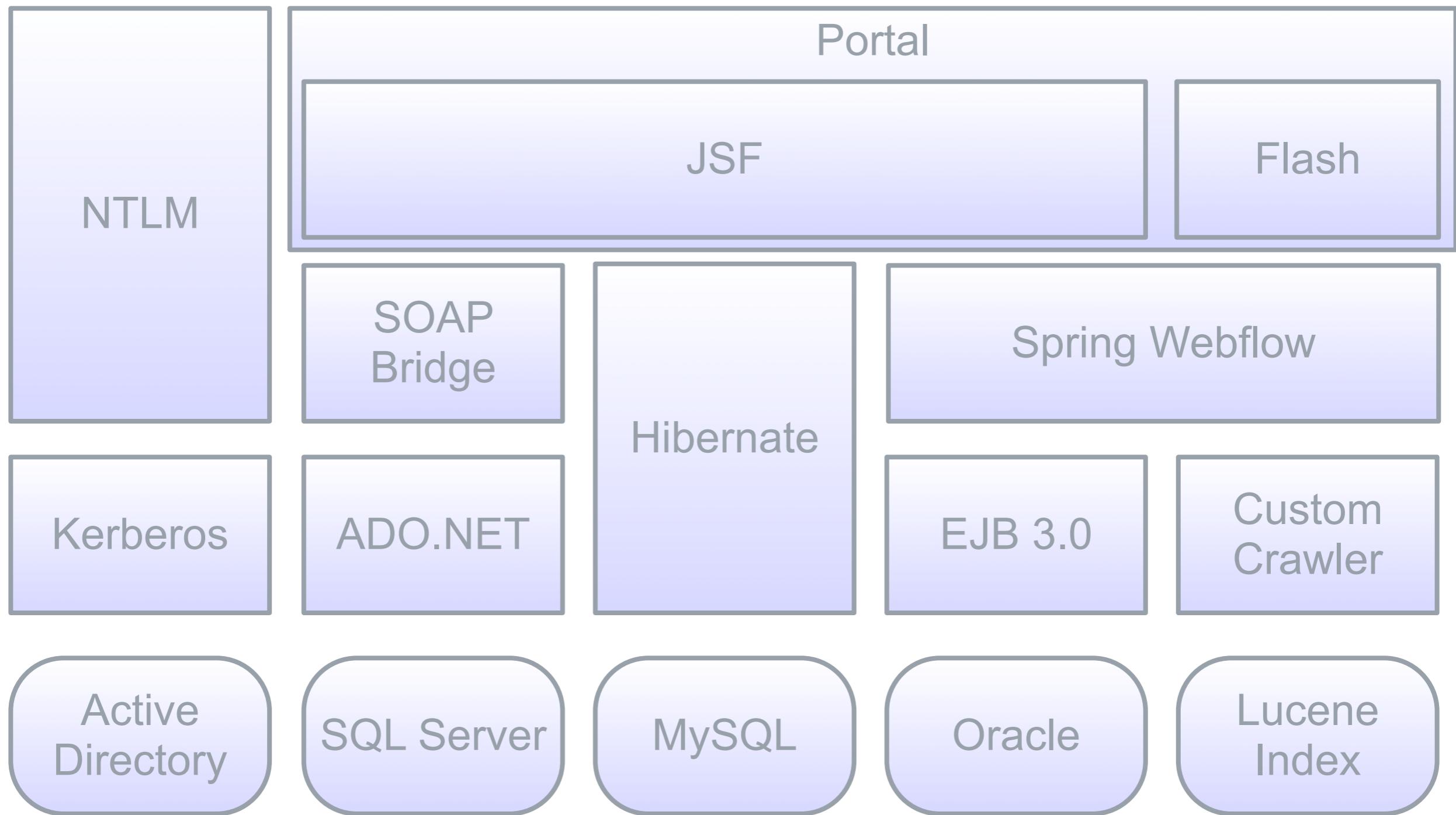
Where is the frontend? And who controls the application?



How does this scale? And why should it scale?



Let me extend this: DB migration, EJB annotation, POJO definition, Spring XML, ...

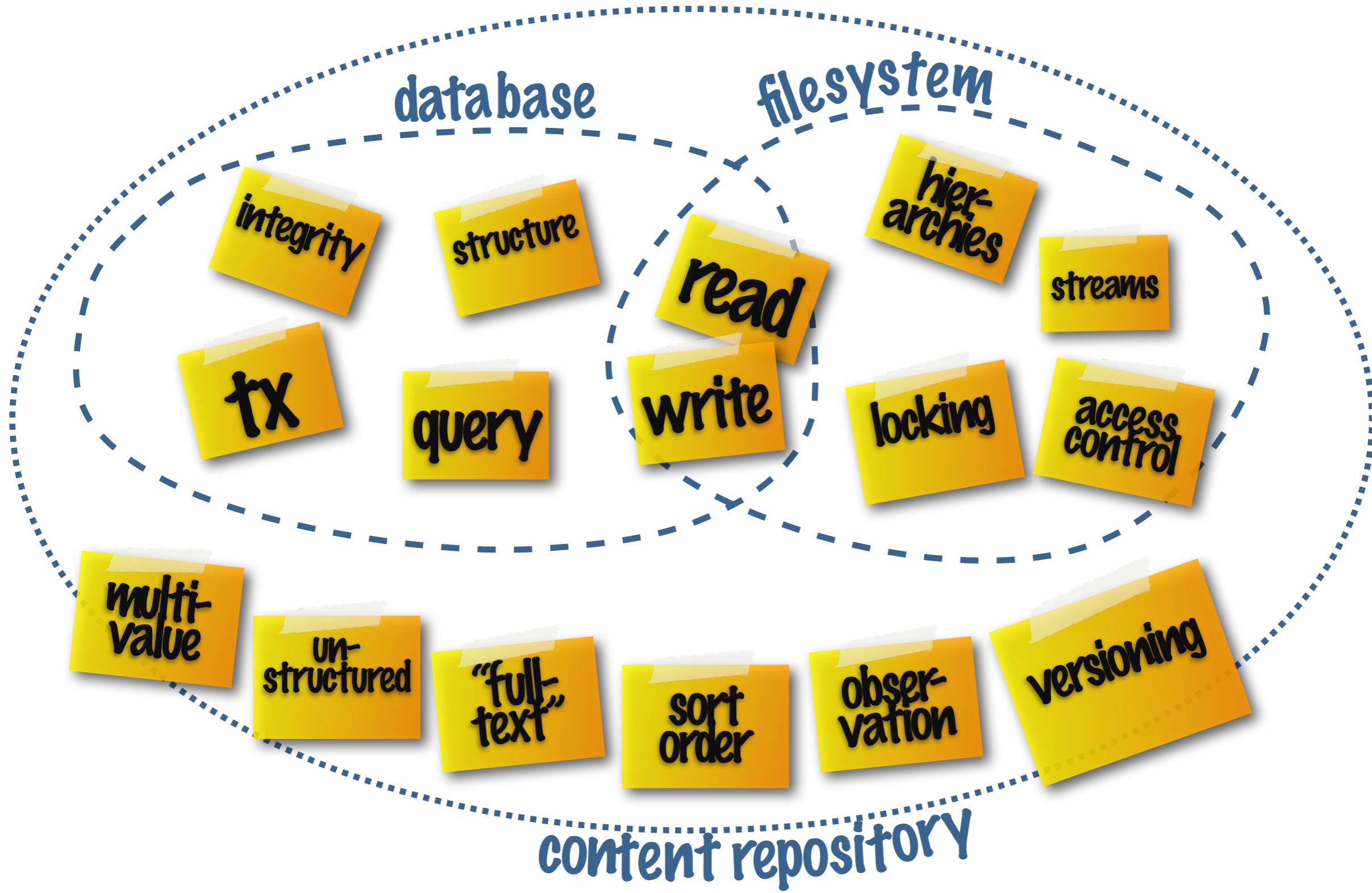


Building an alternative

Building an alternative

CRX (+Connectors)

Best of both worlds



Building an alternative

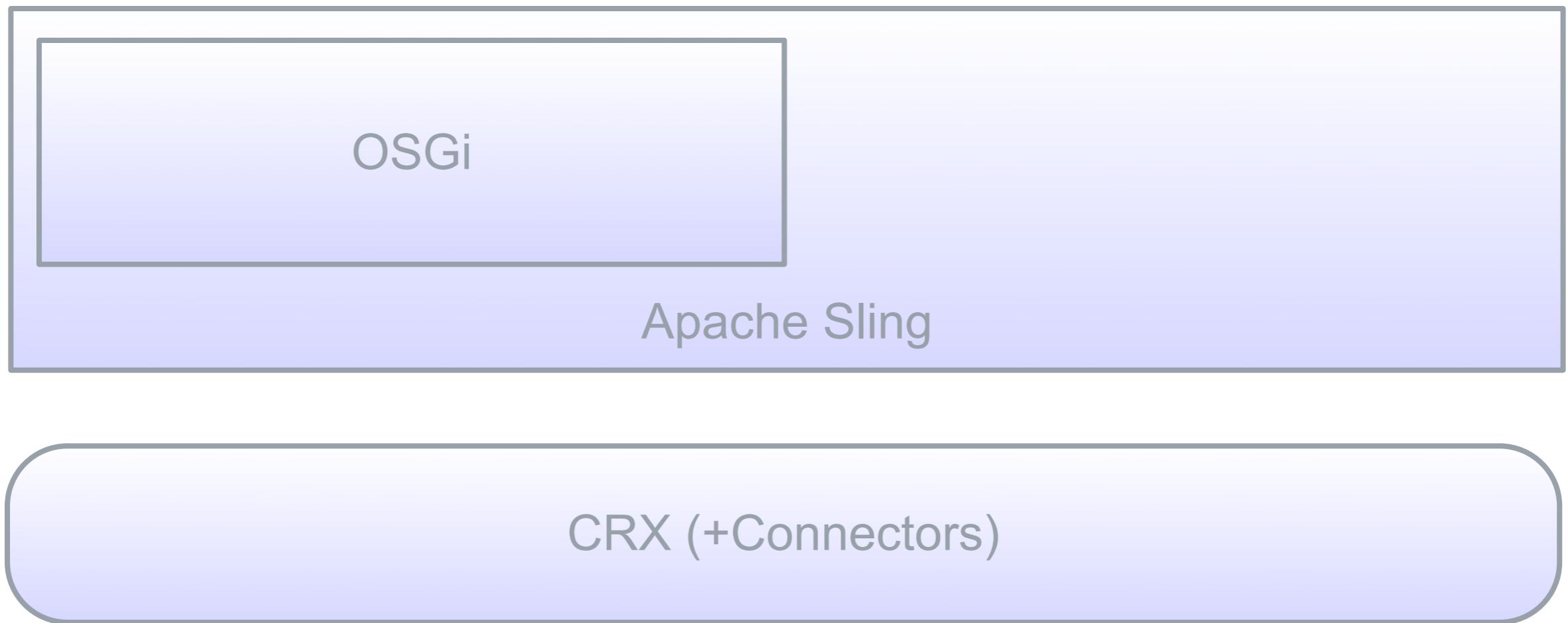
CRX (+Connectors)

Building an alternative

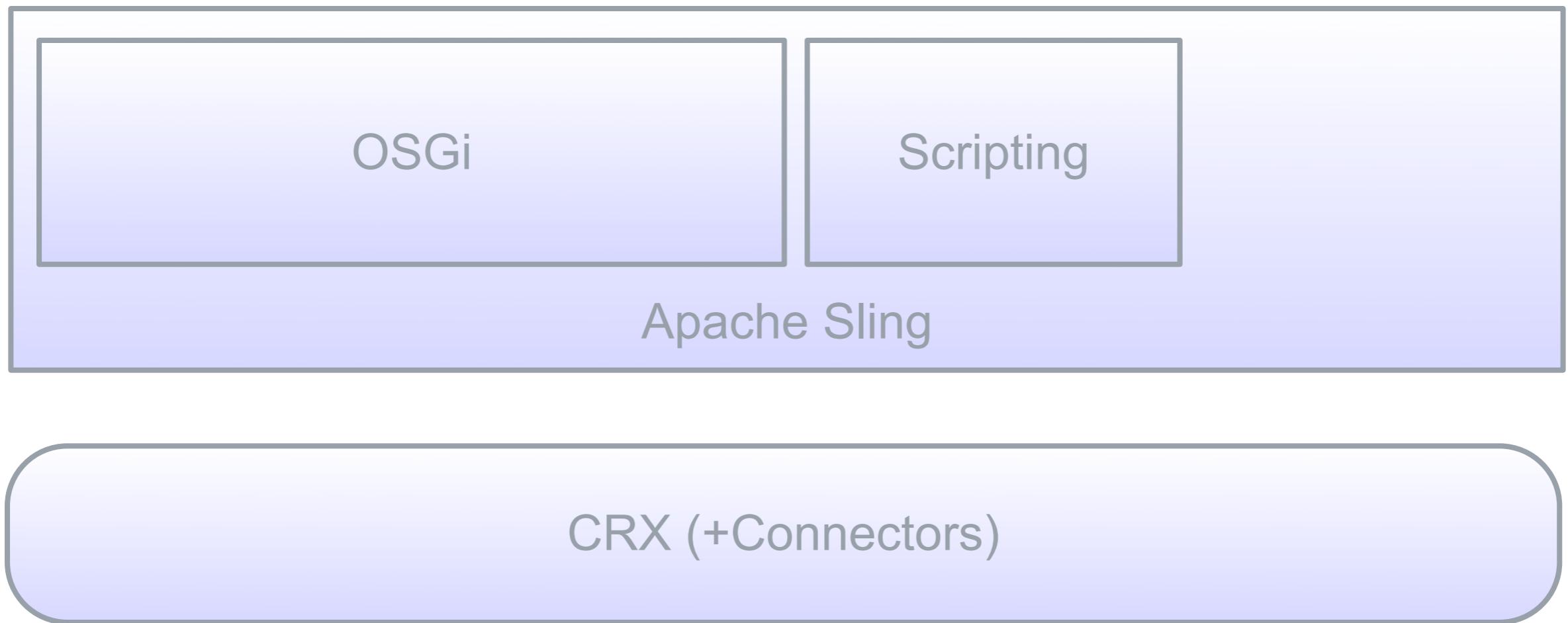
Apache Sling

CRX (+Connectors)

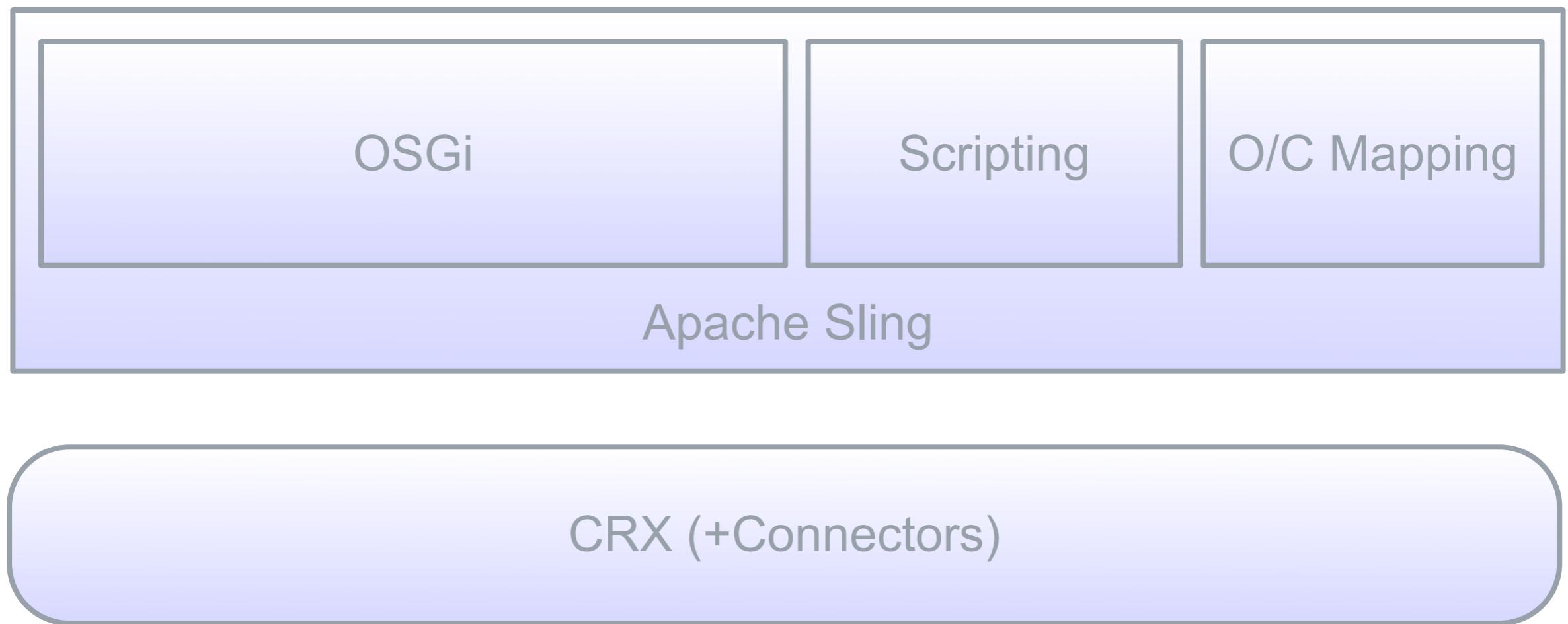
Building an alternative



Building an alternative



Building an alternative



Building an alternative

Web Browser (Frontend Runtime)

OSGi

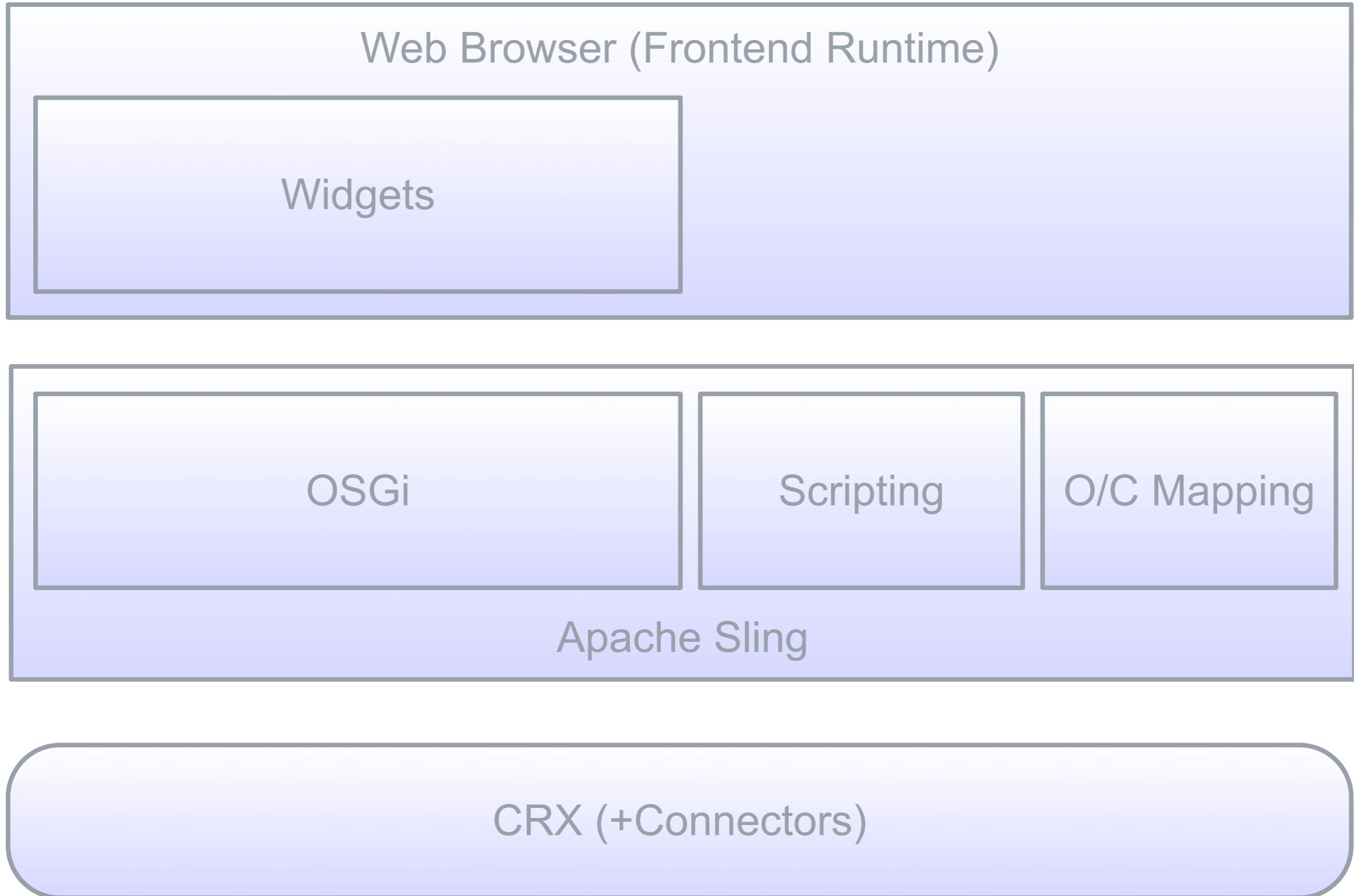
Scripting

O/C Mapping

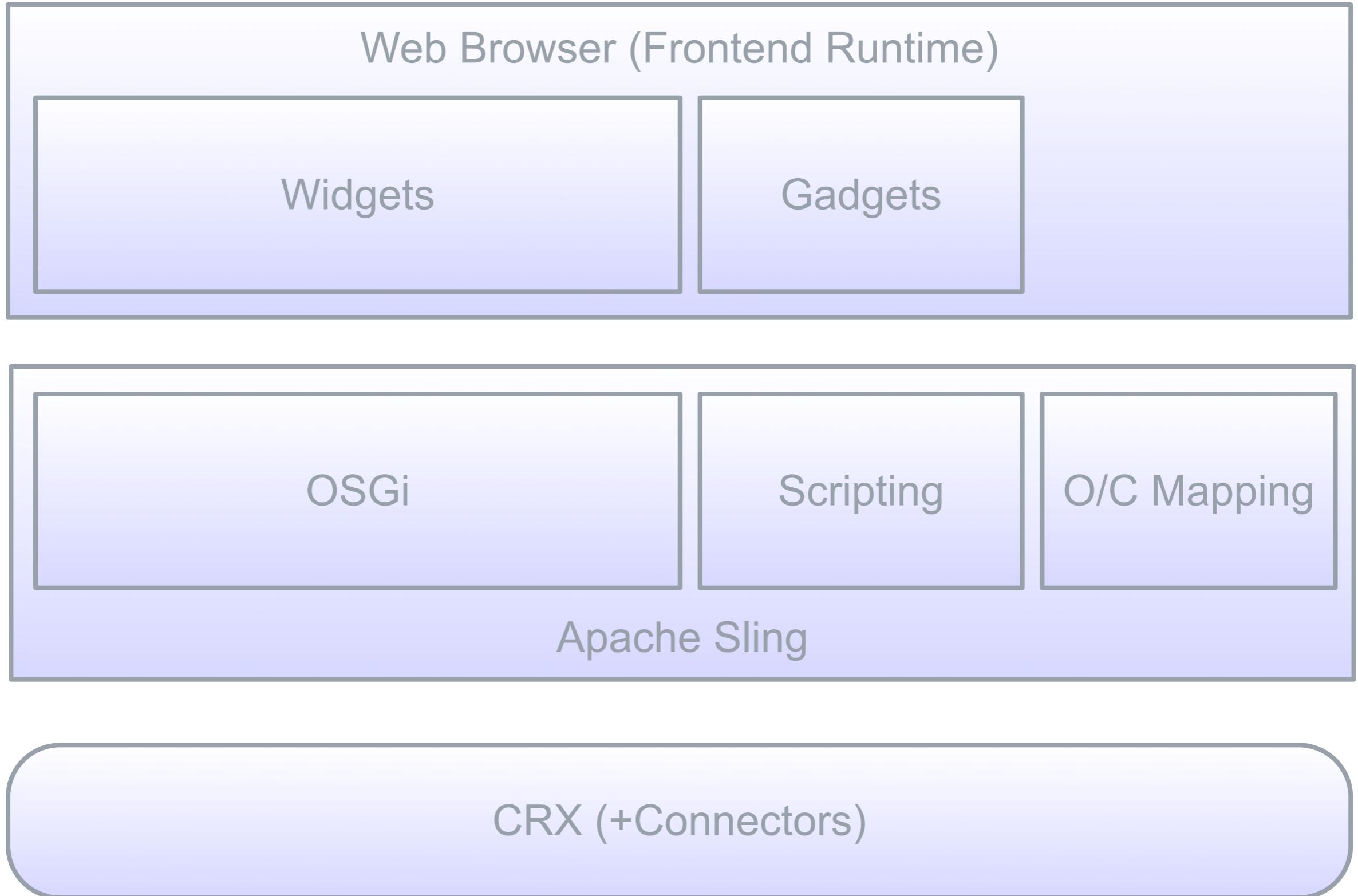
Apache Sling

CRX (+Connectors)

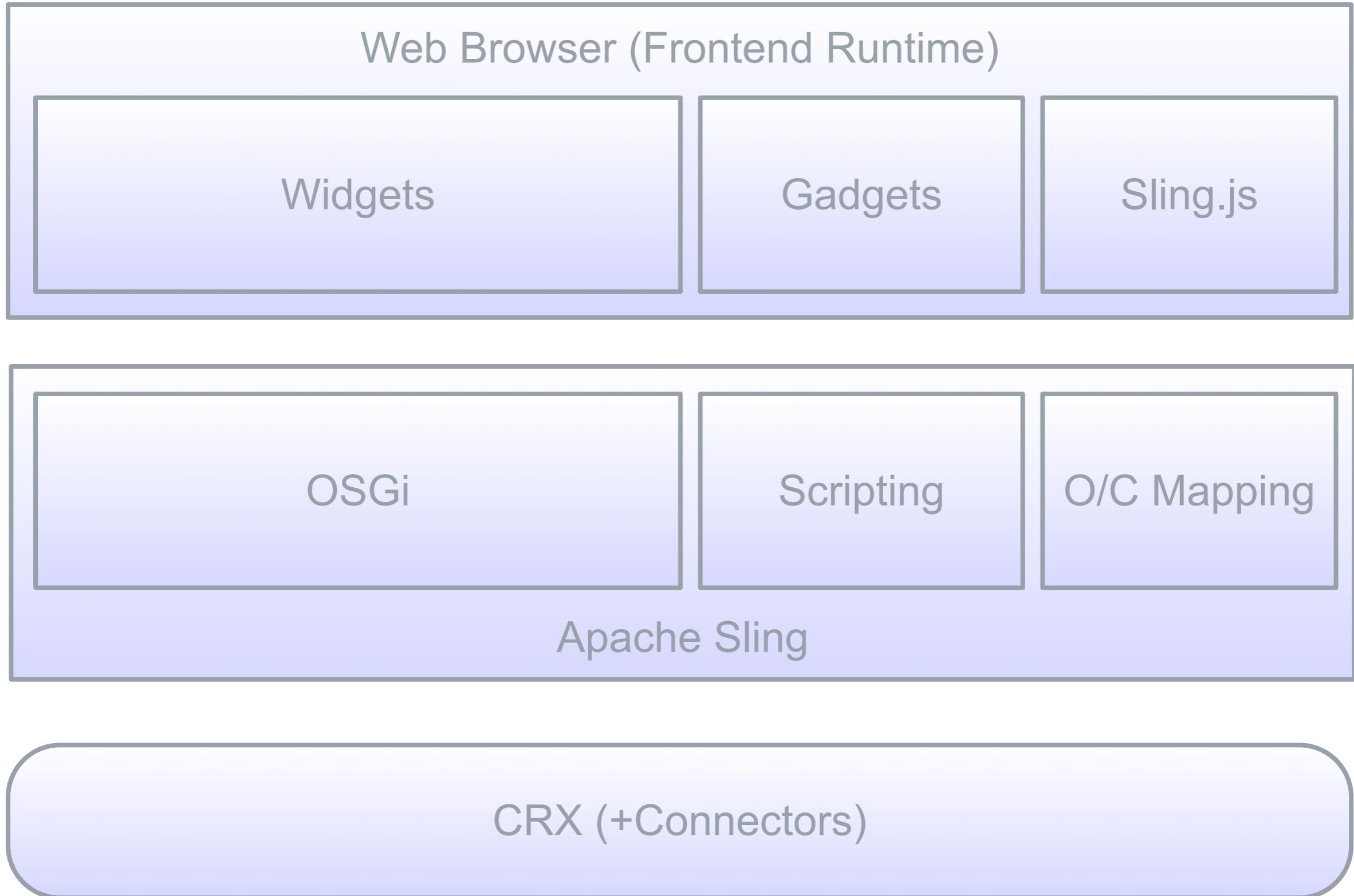
Building an alternative



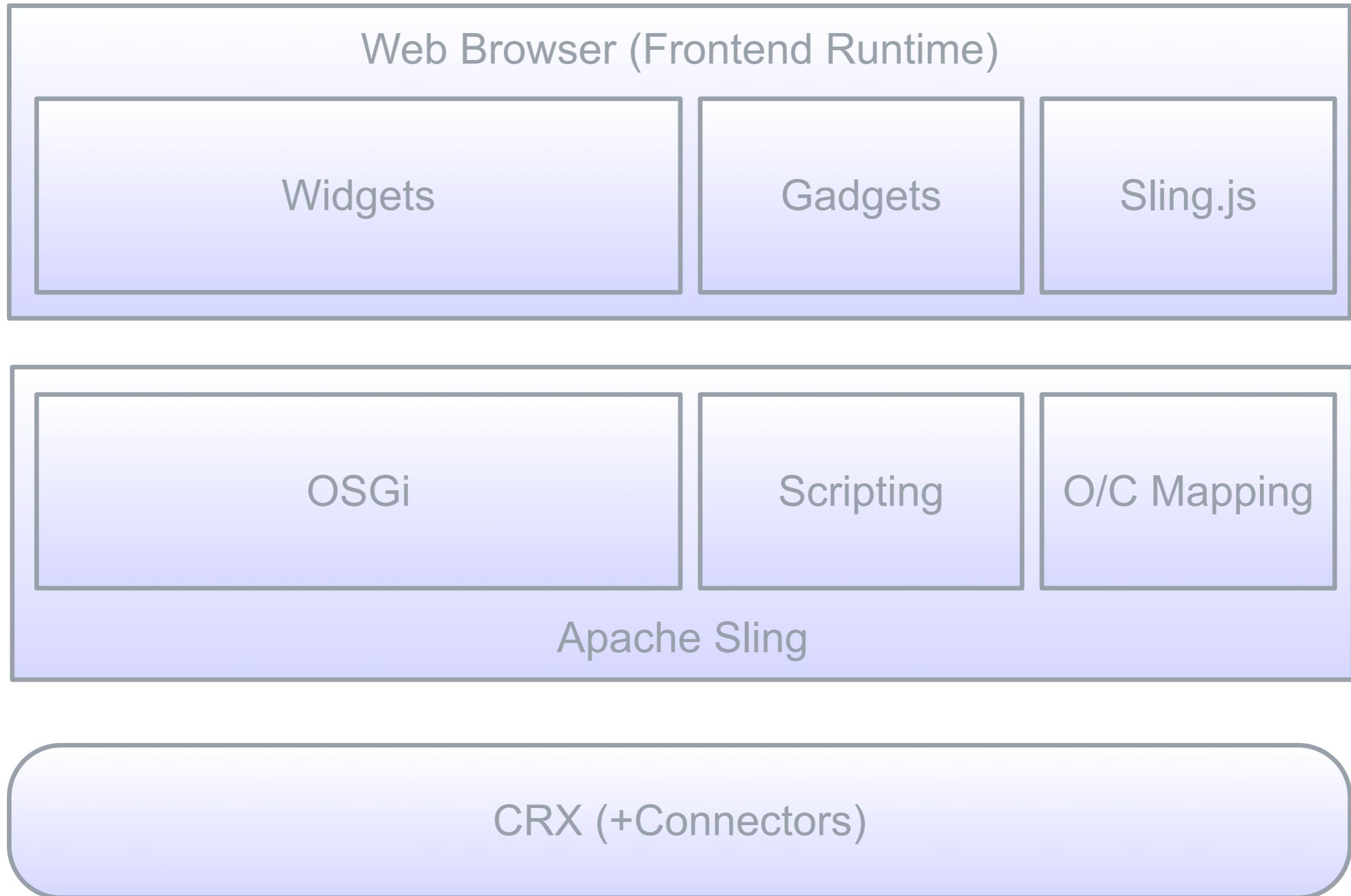
Building an alternative



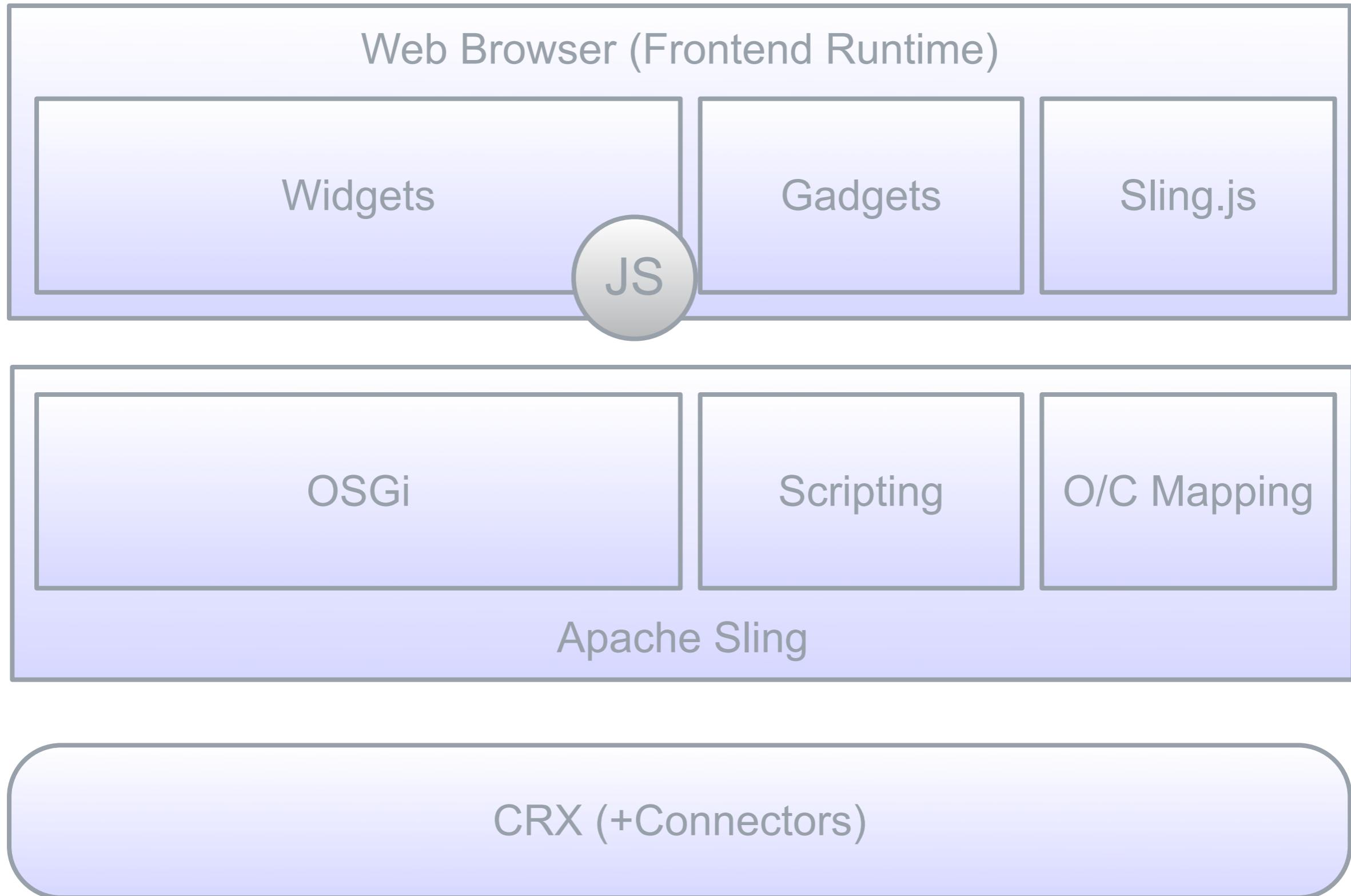
Building an alternative



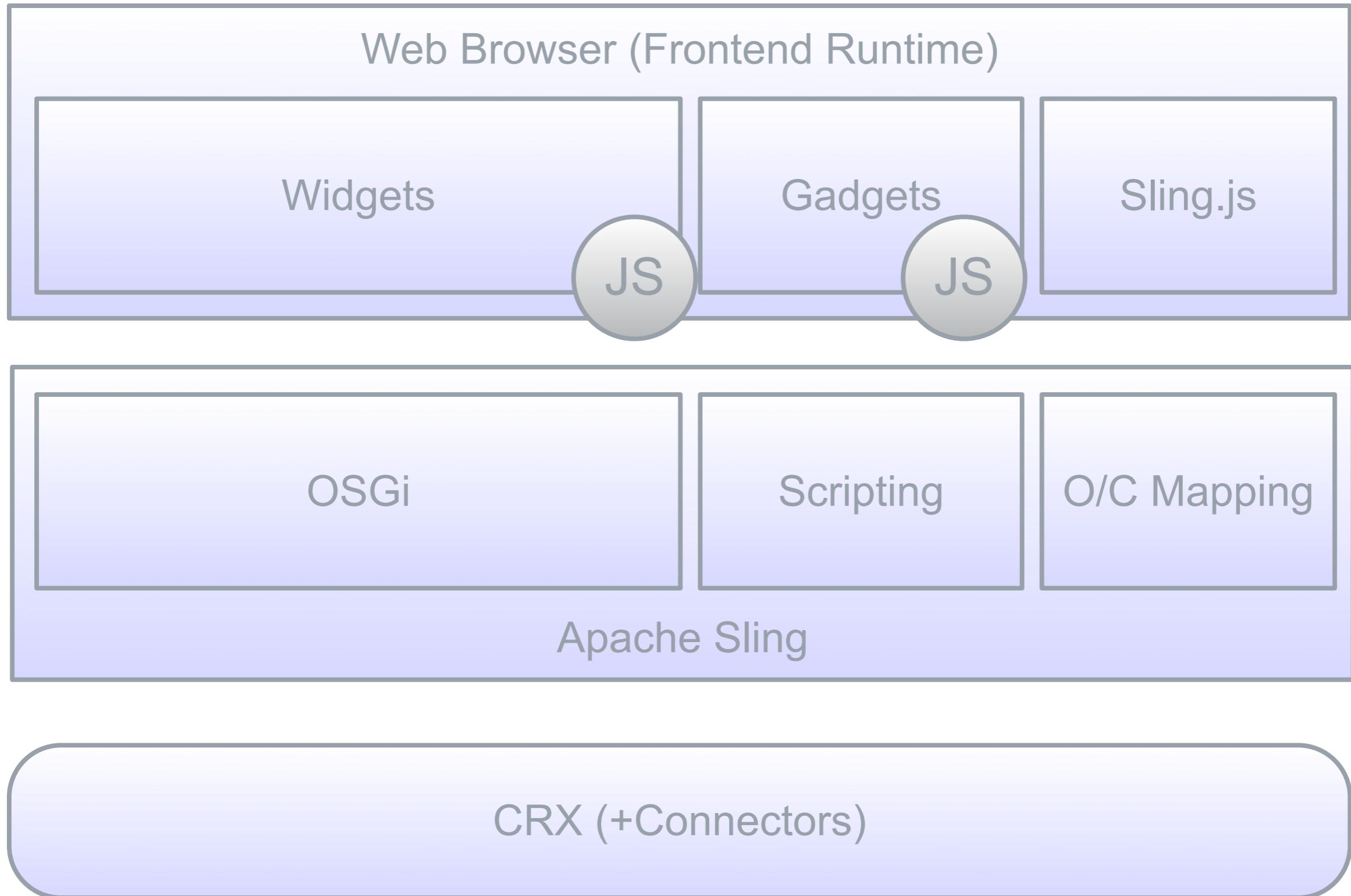
Comparing approaches - Languages



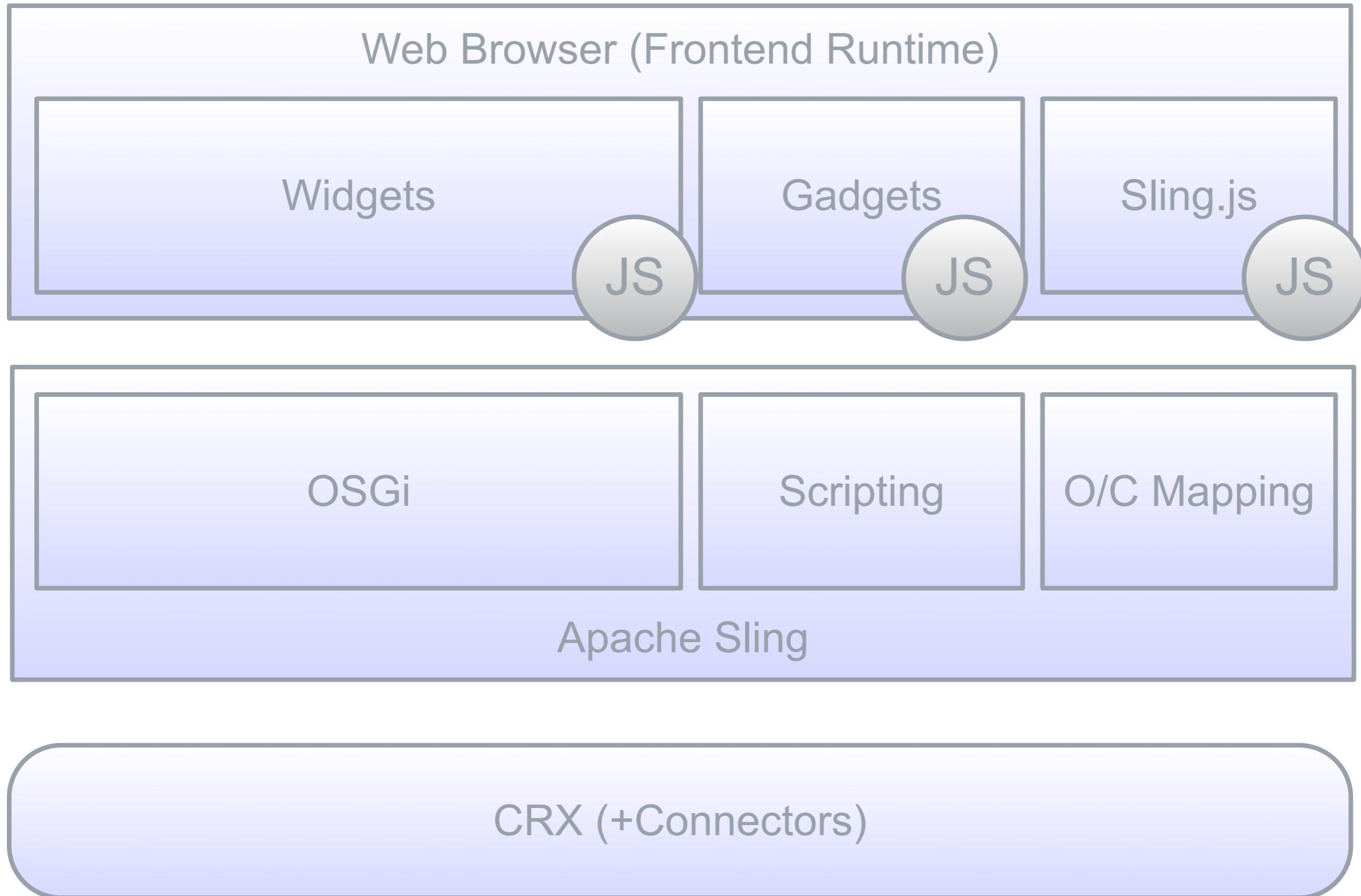
Comparing approaches - Languages



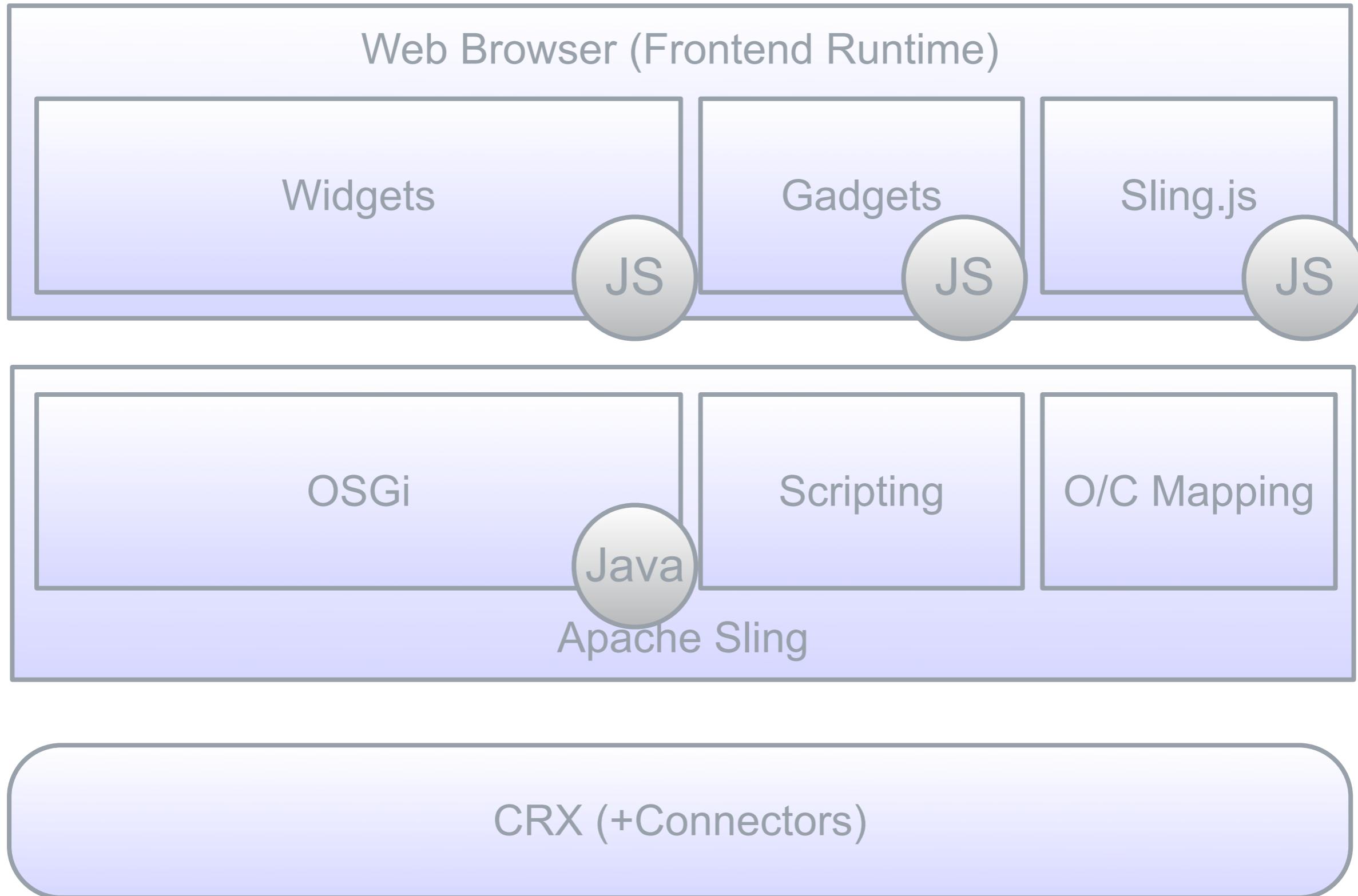
Comparing approaches - Languages



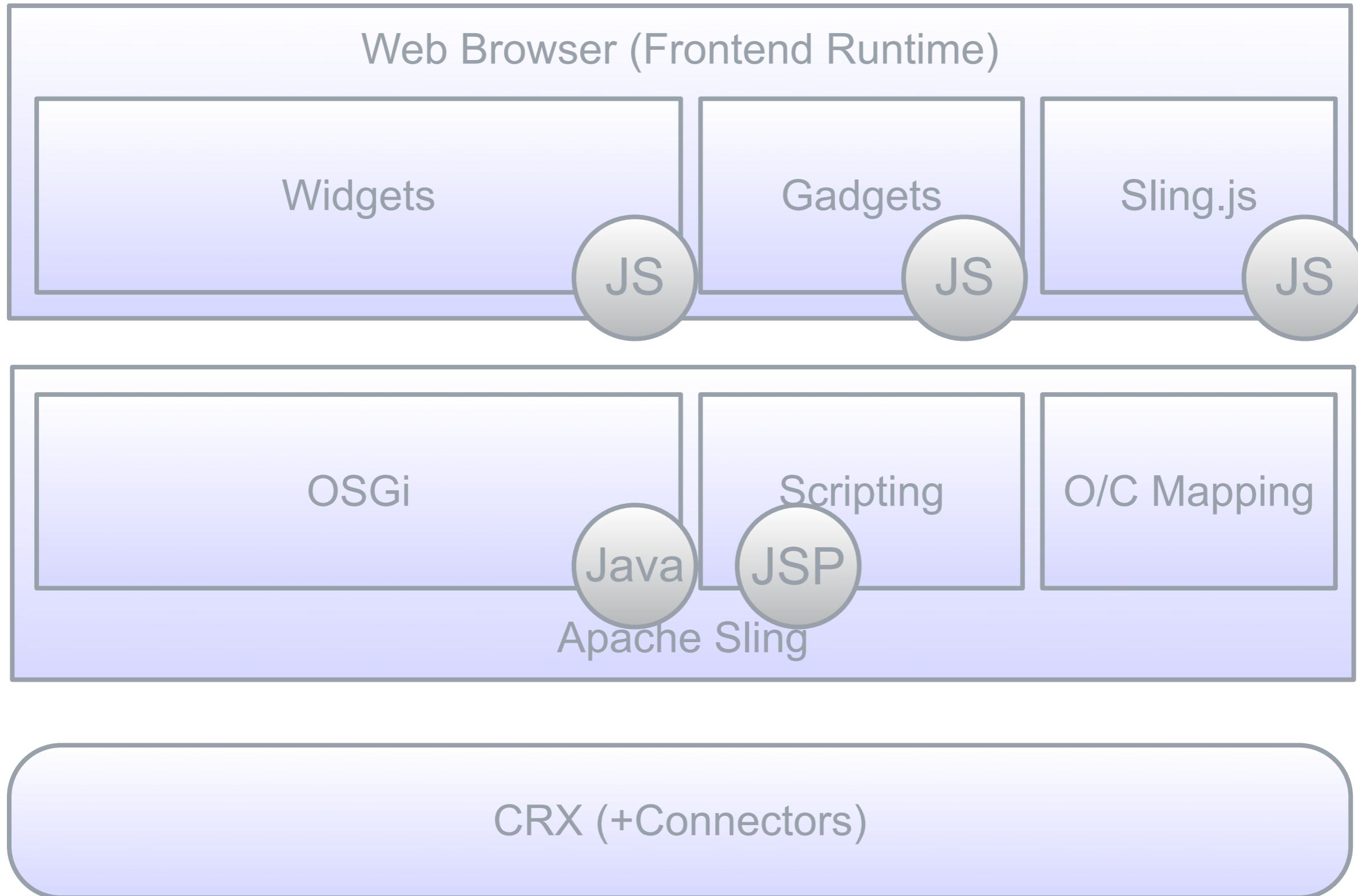
Comparing approaches - Languages



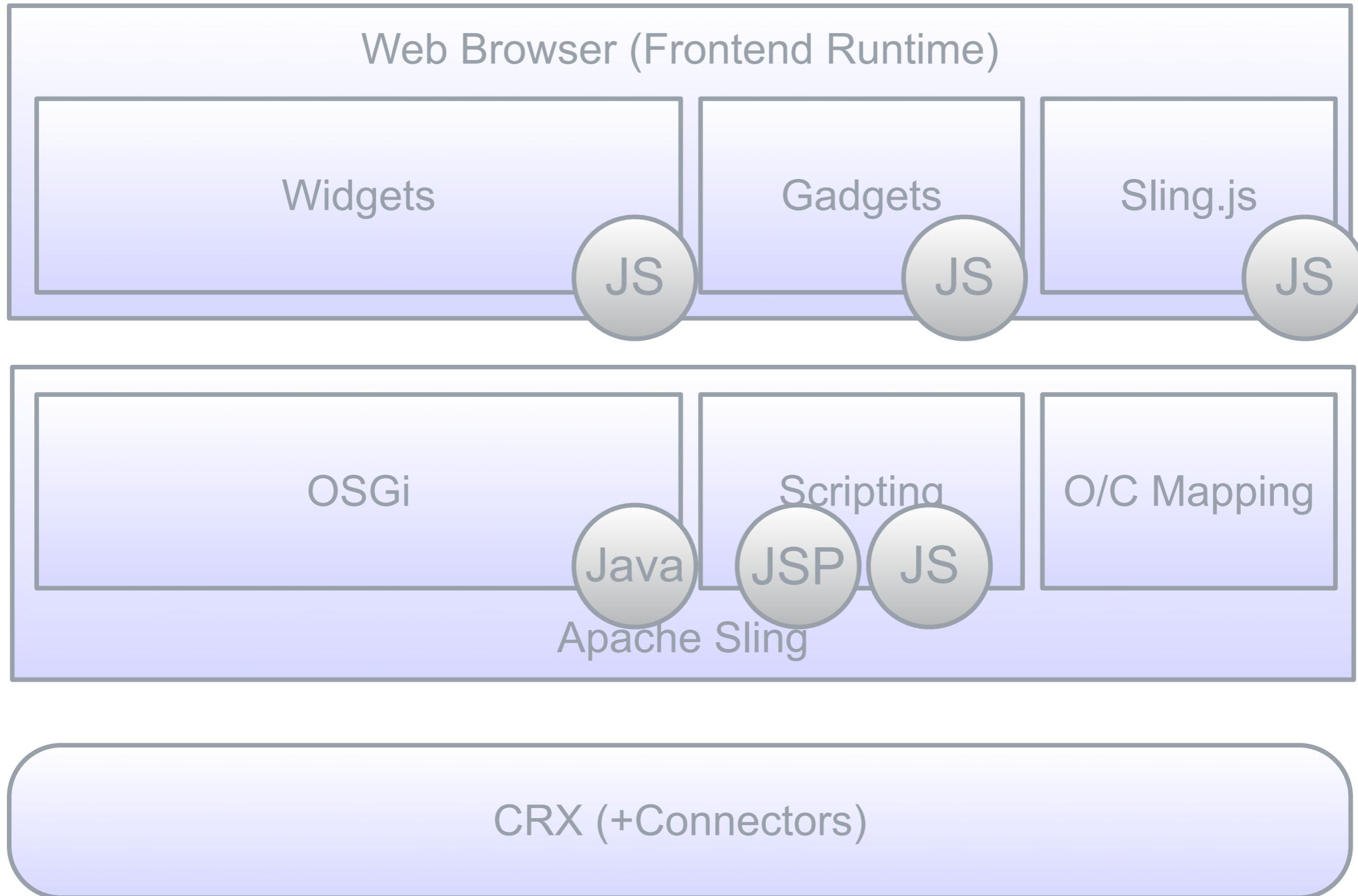
Comparing approaches - Languages



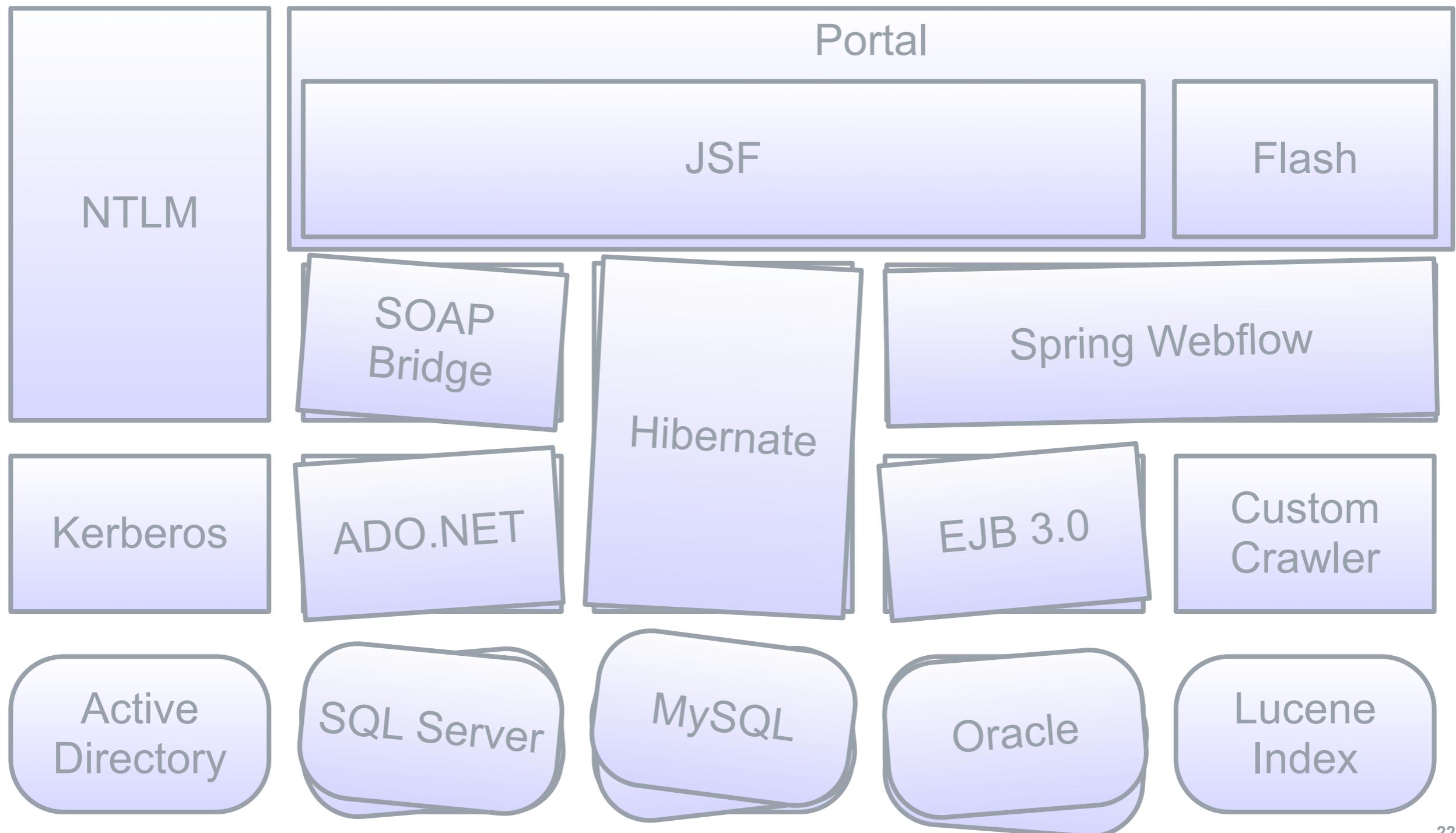
Comparing approaches - Languages



Comparing approaches - Languages



Comparing approaches - Languages



Comparing approaches - Interfaces

Web Browser (Frontend Runtime)

Widgets

Gadgets

Sling.js

OSGi

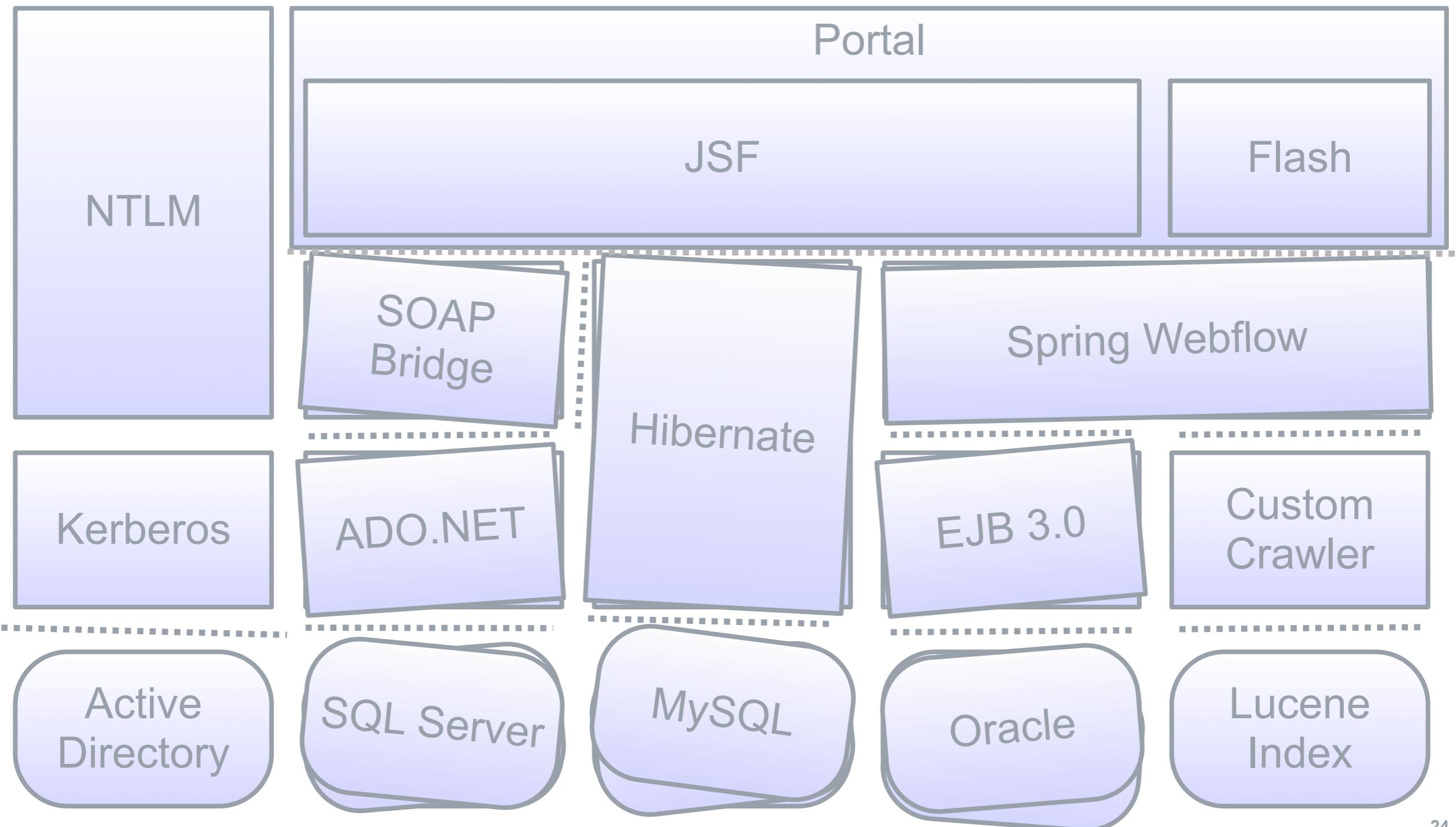
Scripting

O/C Mapping

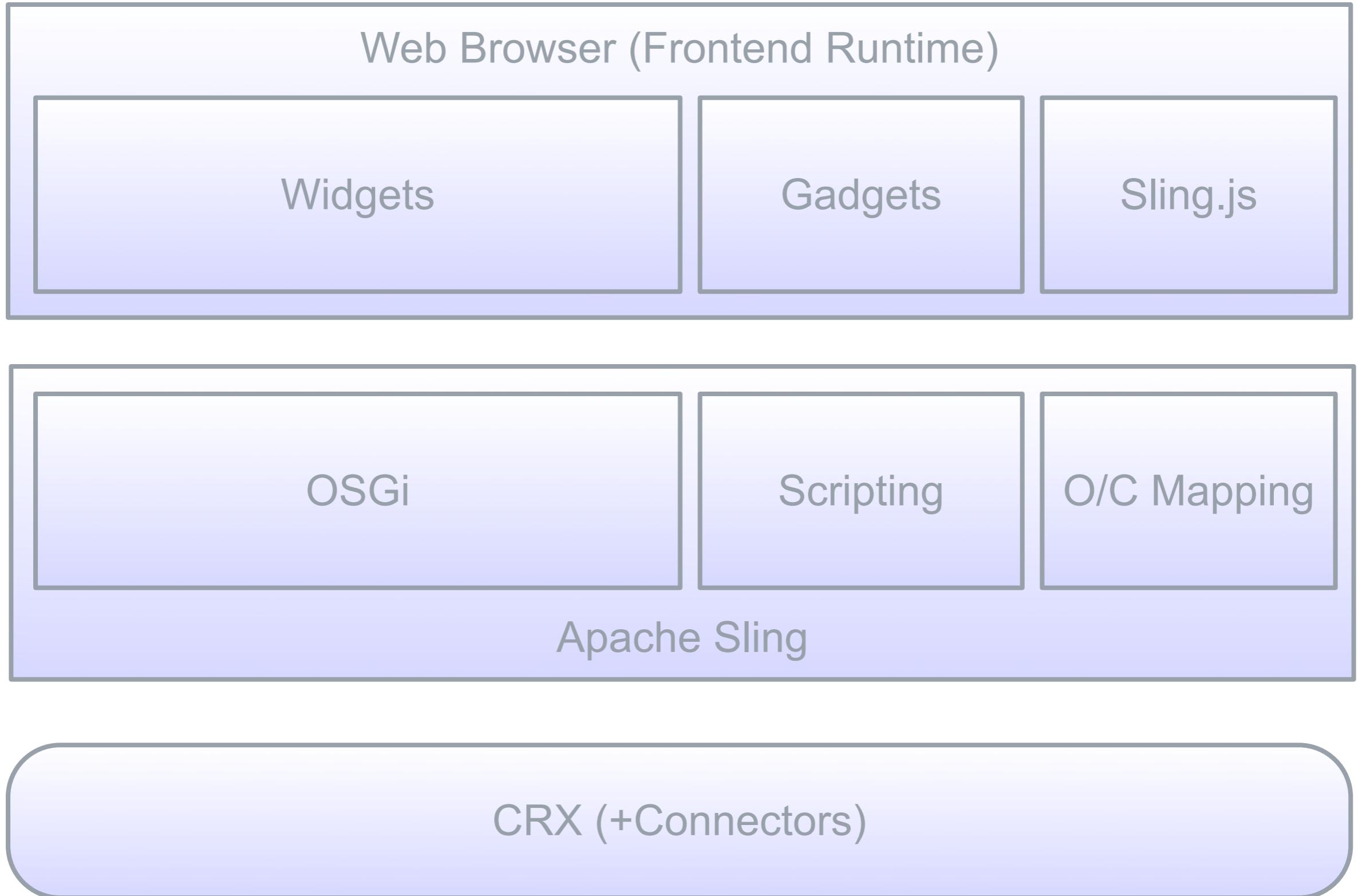
Apache Sling

CRX (+Connectors)

Comparing approaches - Interfaces



Building an alternative



Building an alternative

If you need a
desktop client: try
Prism & Fluid

CRX (+Connectors)

I

Don't bullshit

1

Don't bullshit

2

think before you act

1

Don't bullshit

2

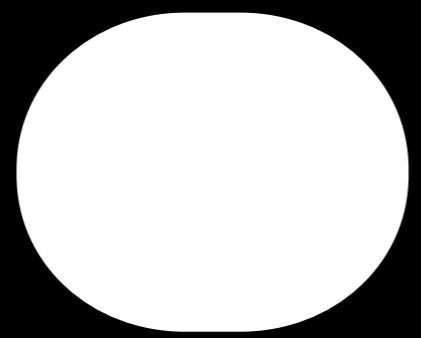
think before you act

3

consider CRX

Thank you

Learn more at www.day.com
or ask lars.trieloff@day.com



Day

**JOURNAL
CUP '08**