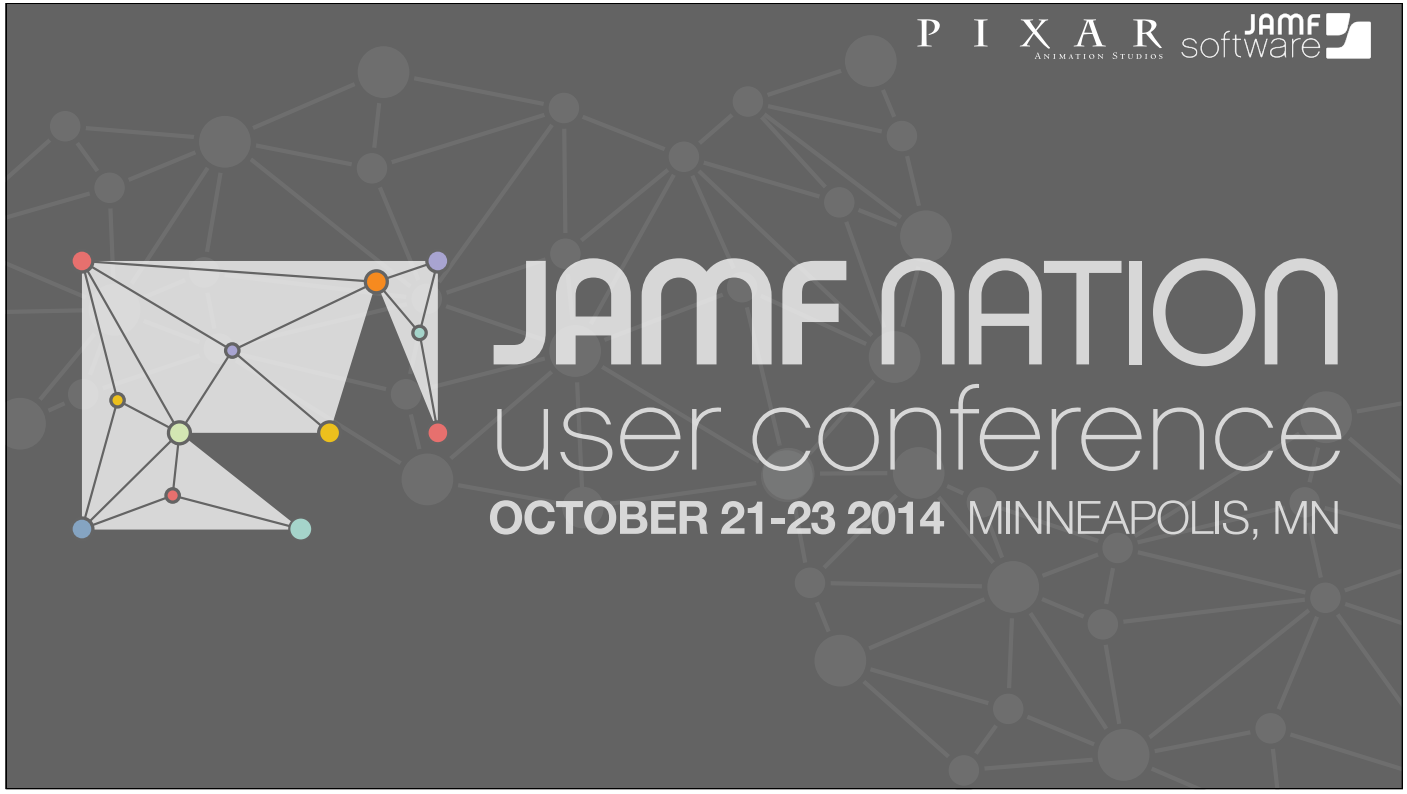

PIXAR
Animation Studios
JAMF software
JAMF NATION
user conference
OCTOBER 21-23 2014 MINNEAPOLIS, MN

PIXAR Animation Studios JAMF software

Objects to Objects

The Casper Suite API through Ruby

PIXAR Animation Studios JAMF software

Chris Lasell

Apple Peeler
Pixar Animation Studios

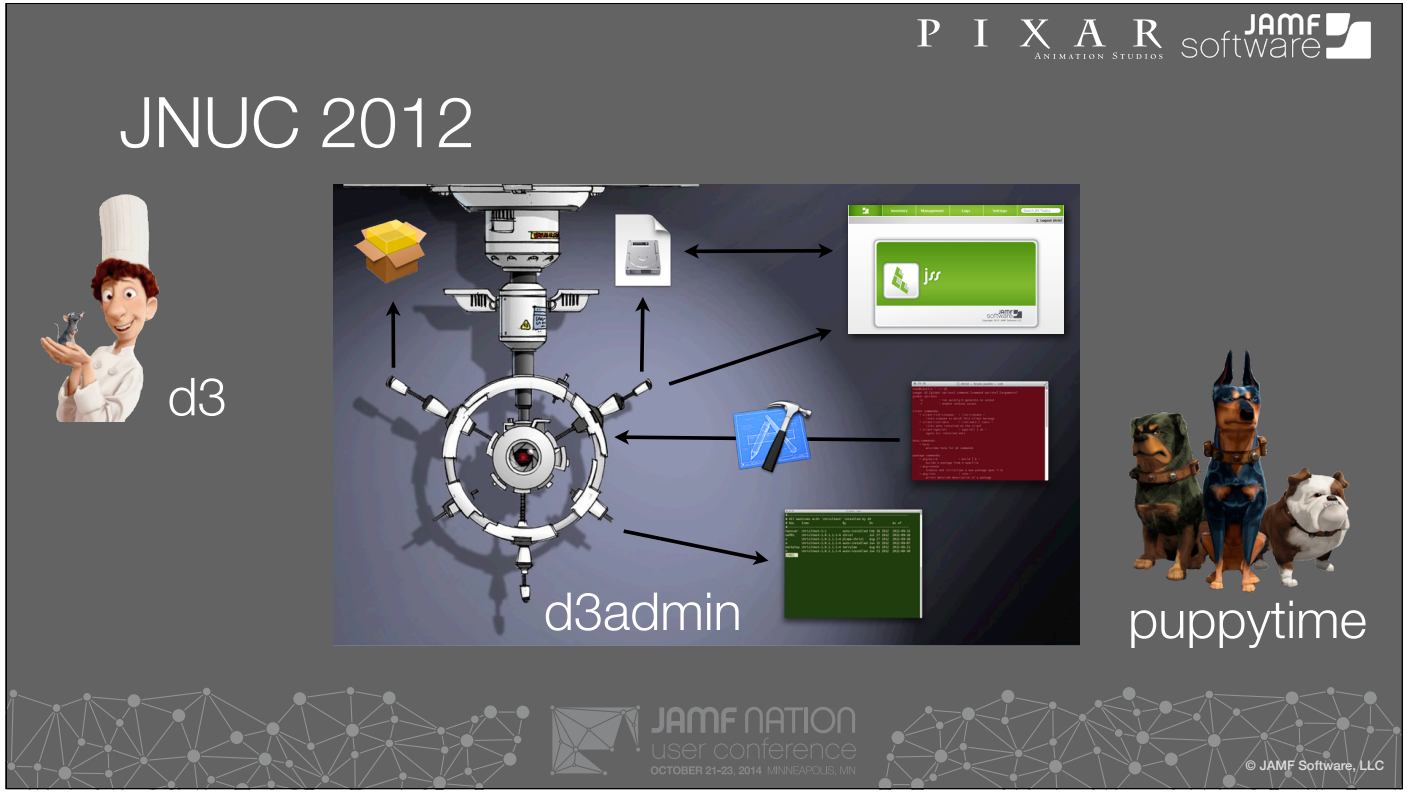
John Sutcliffe

VP, Software Engineering
JAMF Software

JAMF NATION user conference
OCTOBER 21-23, 2014 MINNEAPOLIS, MN

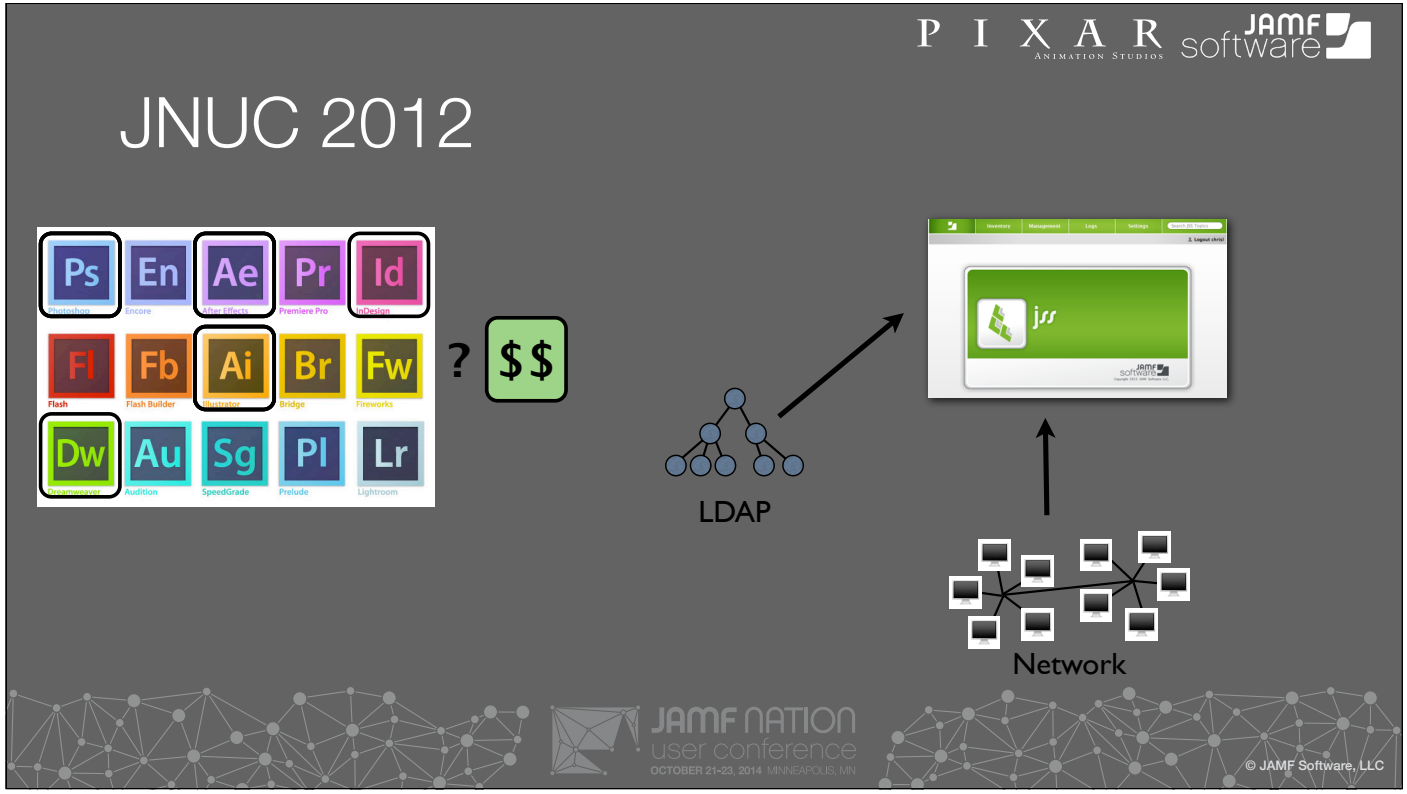
Objects to Objects

JAMF NATION user conference
OCTOBER 21-23, 2014 MINNEAPOLIS, MN



At JNUC 2012

I did a session about some command-line tools we wrote that extend the Casper Suite
Primarily about • d3, our package deployment & patch management system, which includes
•puppytime to entertain users during a logout-install,
and • d3admin which provides manual or scriptable deployment of packages, even directly from XCode



I also mentioned a few other things we do with the Casper API like
-• purging unused applications from user's machines
-• LDAP synchronization
-• automatic network segment updating

We do all that with a homegrown library of Ruby code.•

That library has been growing & by Casper 9, it needed a full re-write.
-For the past year plus, I've been working on that rewrite...

PIXAR ANIMATION STUDIOS JAMF software

JNUC 2012

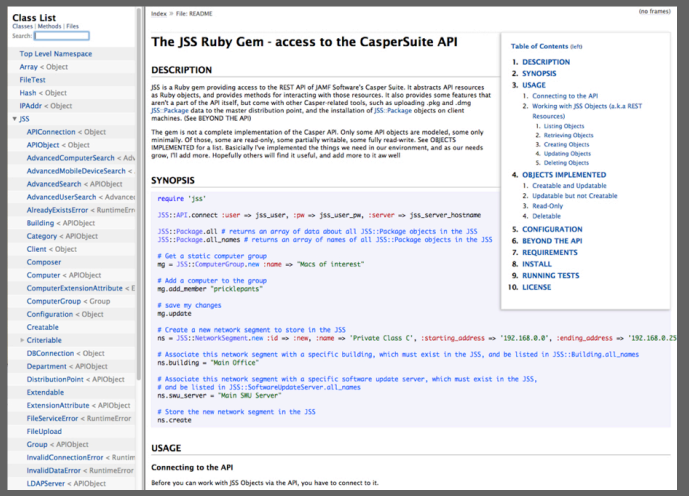
"Is it open sourced?"

-Rich Trouton

JNUC 2014

"Yes!"

-Pixar



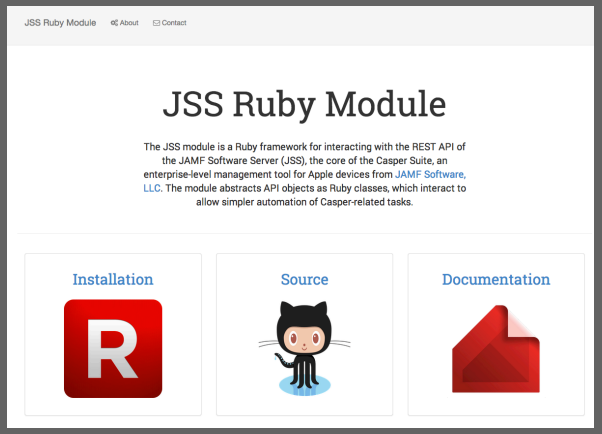
JAMF NATION user conference
OCTOBER 21-23, 2014 MINNEAPOLIS, MN

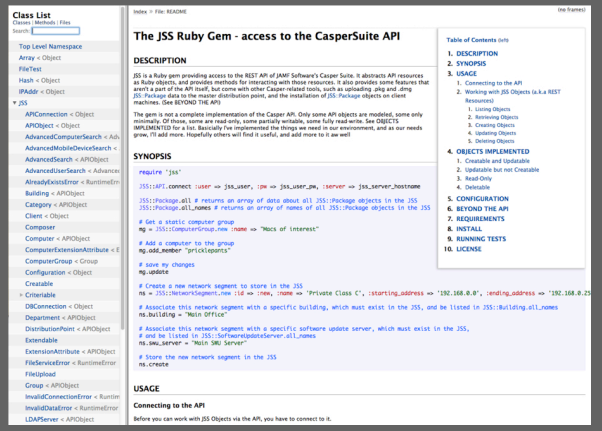
The biggest question after my 2012 talk •

Here we are in 2014 and, in response to Rich Trouton,
• I'm happy to say that the JSS Ruby Gem has been open sourced.

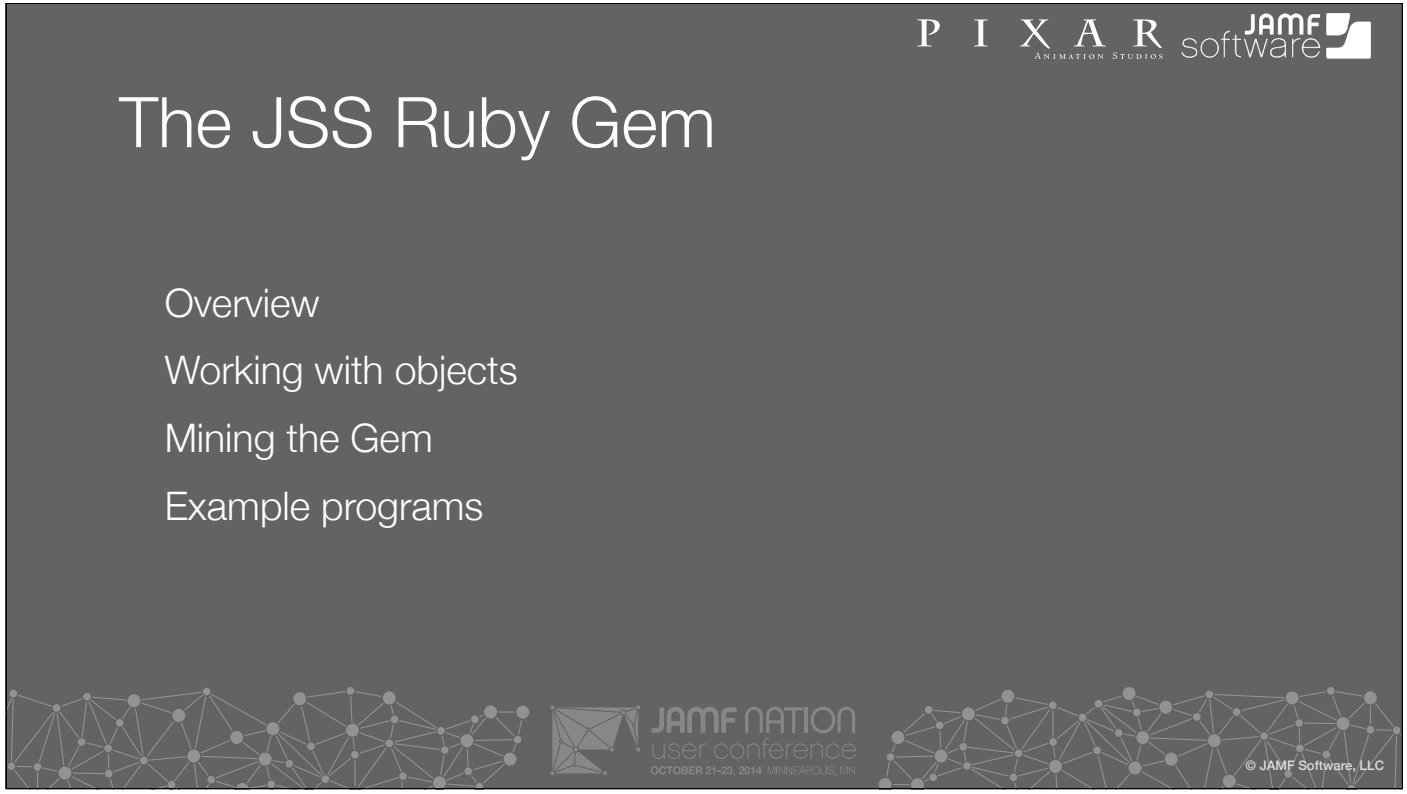
Where to get it

http://pixaranimationstudios.github.io/jss-api-gem





Here're the relevant details, they'll also be available at the end of the slideshow.

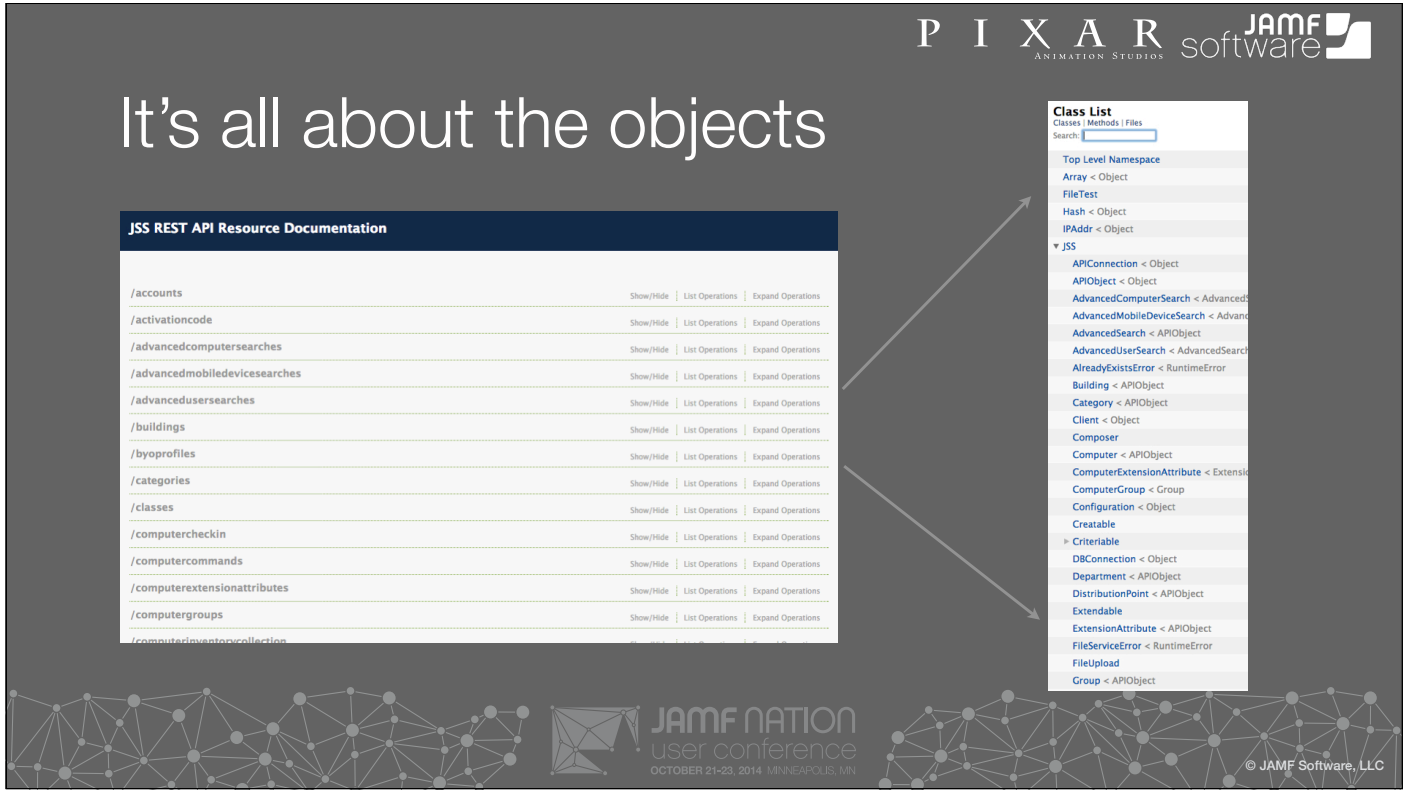


The rest of my talk is going to go over the JSS Gem...

- Overview - what it does, what it doesn't (yet) do
- Working with objects -The basics of using the gem to interact with the API
- Mining the Gem - Coding the Gem itself, how it's built
- Example Programs - A couple of shell commands built with the Gem.

There’s no way I can cover it all today, but
hopefully I’ll give you enough to whet your appetites, and plant the seeds of possibility.

Lets take a look at what it does.



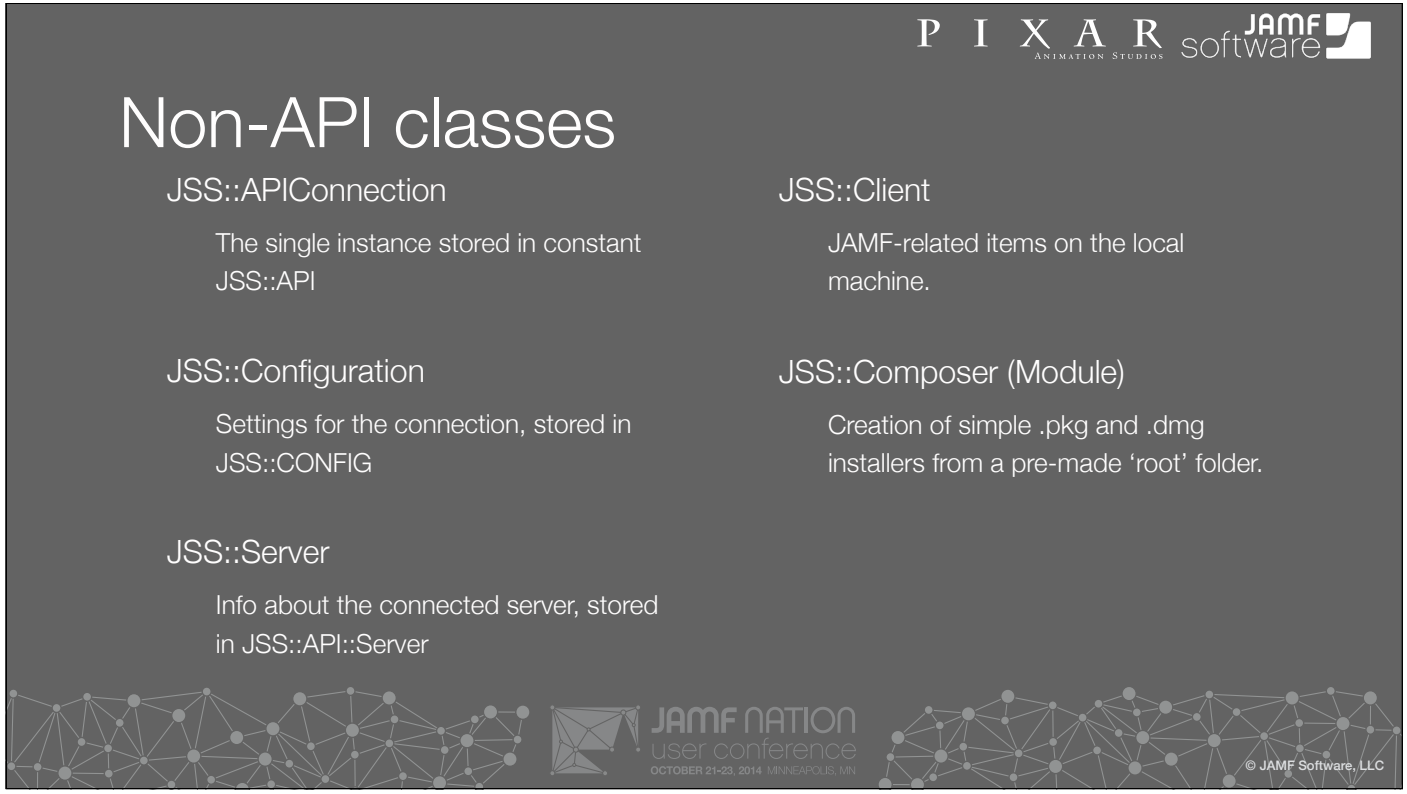
The API provides a way to interact with all kinds of objects stored in the JSS
The JSS Gem is a Ruby module which abstracts those objects as Ruby classes•

When you create an instance of a class, you get a Ruby object with many easy-to-use methods for manipulating it.

As of Casper 9.4, the API provides 58 kinds of objects

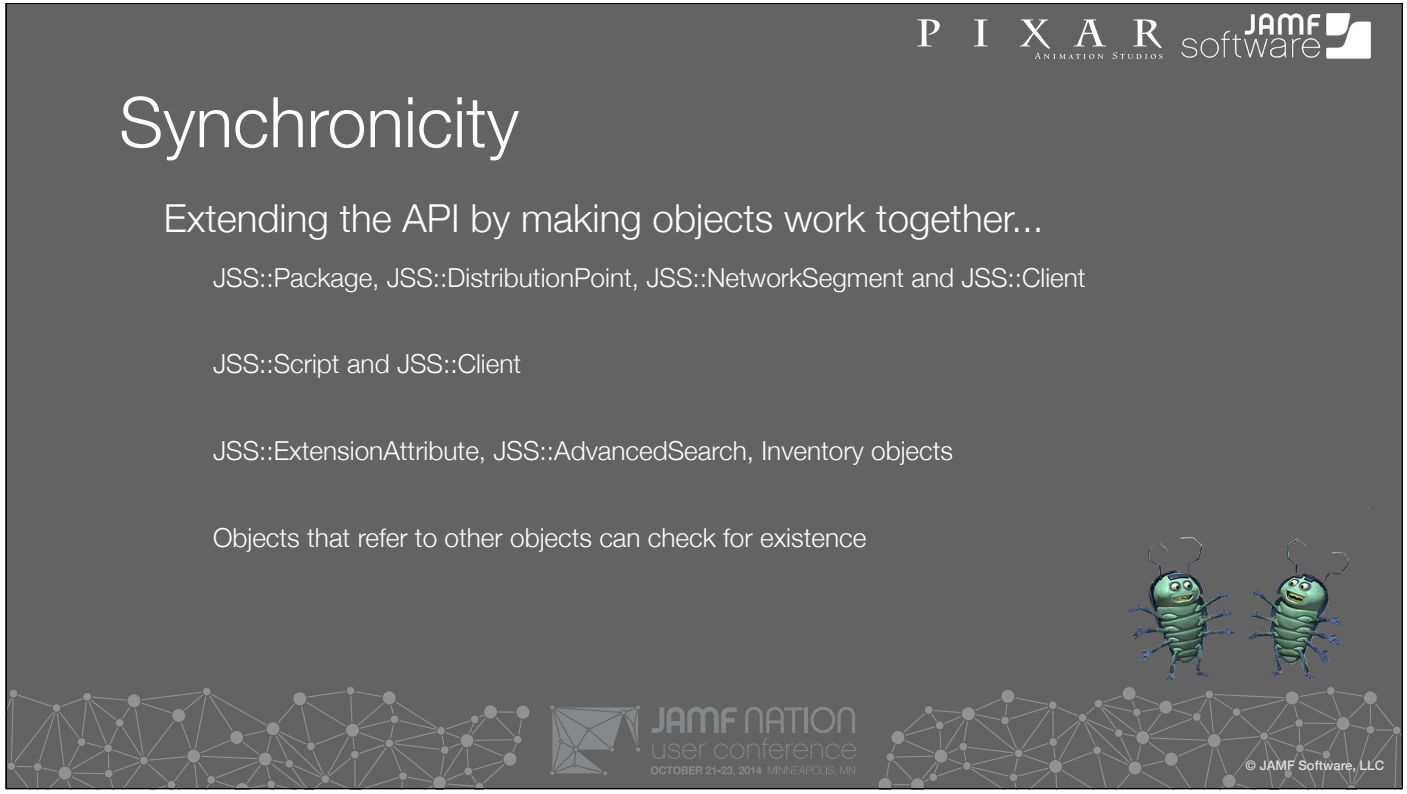
So far the Gem implements 26, to varying degrees, as Ruby classes

- Create, Read, Update & Delete
- Read, Update & Delete
- Read & Delete



There are a few classes and submodules within the Gem that don't reflect API objects directly, but are useful nonetheless.

- APIConnection
  - Only a single instance
  - #connect method takes all connection data, used at any time to change credentials,
  - timeouts can be changed mid-stream
- Config
  - System wide and user-specific config files, settable via methods
- Server
- Client
- Composer



The API provides access to JSS objects, but once we get them into a unified programming environment, those objects can work together to make something even more powerful.

- - Install packages using the jamf binary from the correct dist.point. for the net seg.
- - Run scripts via jamf binary
- - report on ExtAttr values, enforce data consistency for ExtAttrs
- - Better data consistency before getting API errors

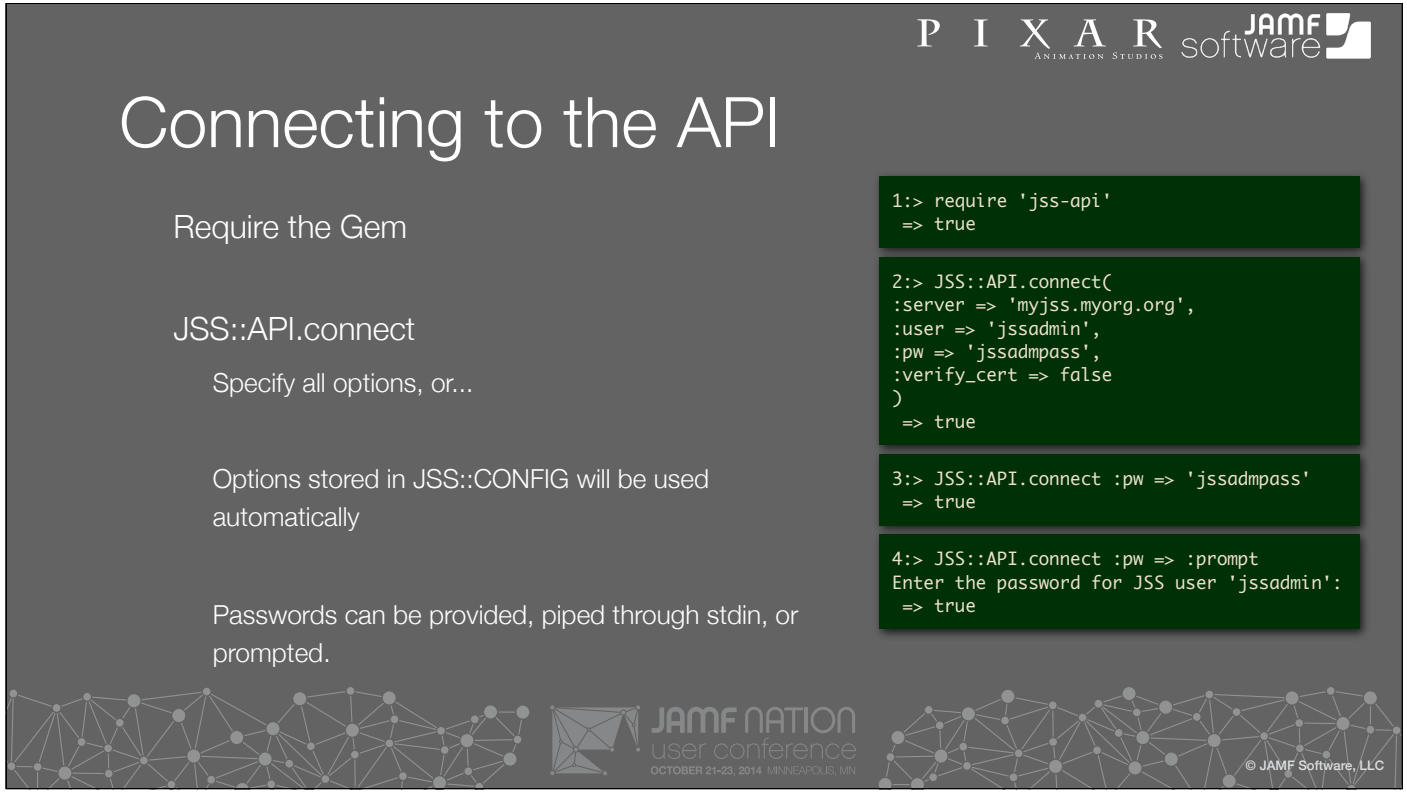
So lets take a look at how to use this thing.

The examples I'll show you are captured from an "IRB" session, which
is like a shell where you can type ruby code just like you can type bash code in a bash shell.

Like all gems, JSS is easy to install..

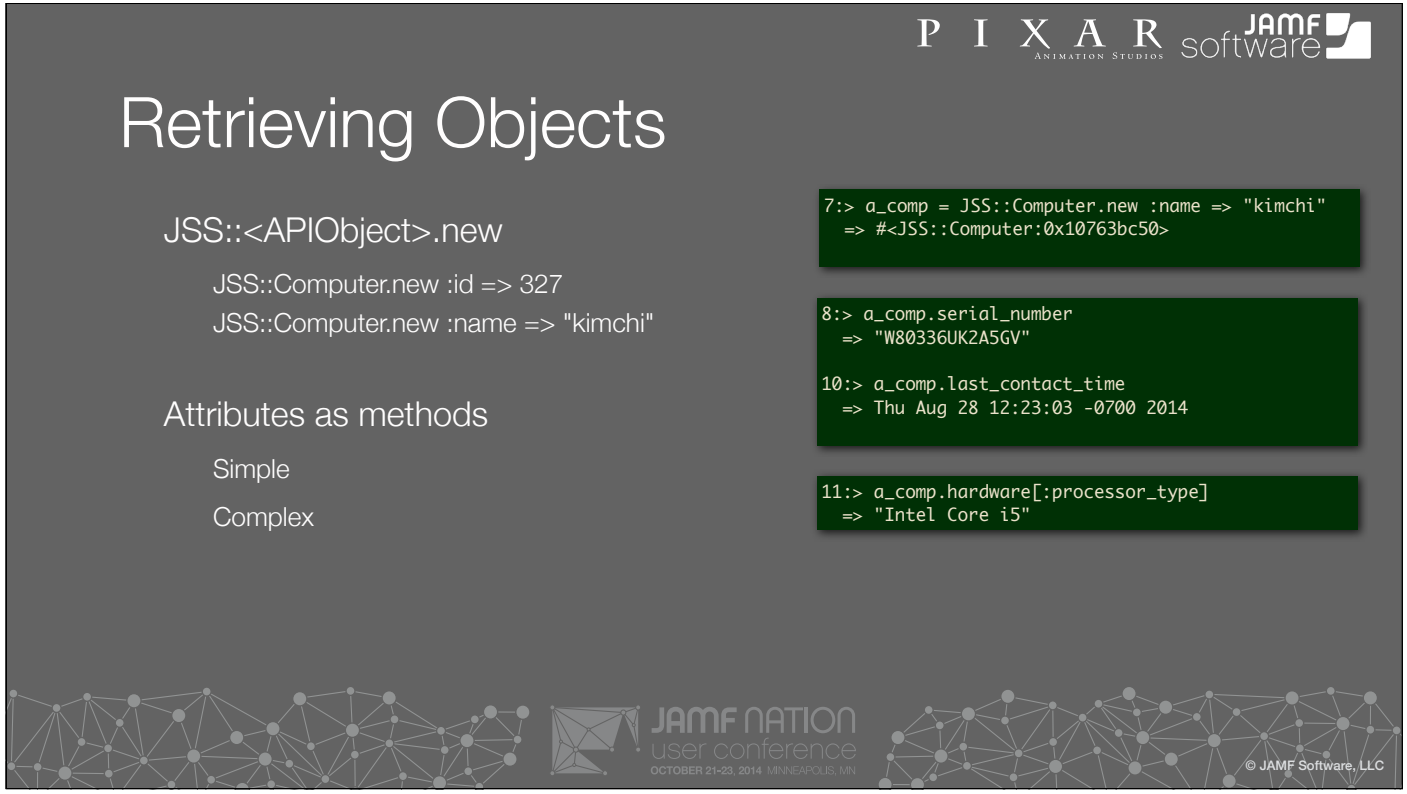
- gem install jss

- installs dependencies as needed



- - require the gem
- - JSS::API.connect, specify all options
- - or use Config and just the passwd
- - or :prompt or :stdin

• APIObject.all
- An Array of Hashes for each object
- The JSON response of the API list resource
• APIObject.all_identifiers - all_names, all_ids, all_macaddresses, etc
• APIObject.map_all_ids_to :name, :serialnumber, etc...



• - Objects are retrived by creating a Ruby instance with the :id or :name of a JSS object.
Sometimes by other identifiers, eg Computers by :id, :name, :serialnumber, :macaddress, :udid
• - Attributes from the API are available via method calls, usually with the same name.
All objects have #id and #name methods. Other vary depending on the object.

Updating Objects

Simple Attributes

More Complex Data

Send Changes to the JSS

```
11:> a_comp.barcode_1 = "123abc890"
  => "123abc890"

12:> a_comp.po_number = "po-98765"
  => "po-98765"

13:> a_comp.po_date = 'Sep 13, 2014'
  => "Sep 13, 2014"
```

```
14:> a_policy = JSS::Policy.new :name => "sw-update-iphoto"
 => #<JSS::Policy:0x101d3d978>

15:> a_policy.scope.add_limitation :network_segments, 'backups'

>>>>> JSS::NoSuchItemError: No existing network_segments with name 'backups'

16.^ JSS::NetworkSegment.all_names
  => ["backup-lan", "av-theater", "main-3-l1"...]

17:> a_policy.scope.add_limitation :network_segments, 'backup-lan'
  => true
```

```
18:> a_comp.update
  => 3313

19:> a_policy.save
  => 735
```


- - Change the value of a simple attribute with =
- - Some attributes are more complex, like scope, which is a Scope object. - errors can be raised
- - Save is an alias of update

Creating Objects

JSS::<APIObject>.new :id => :new, :name => '<a_name>'

JSS::Package.new :id => :new, :name => "filemaker13.pkg"

Add attribute data

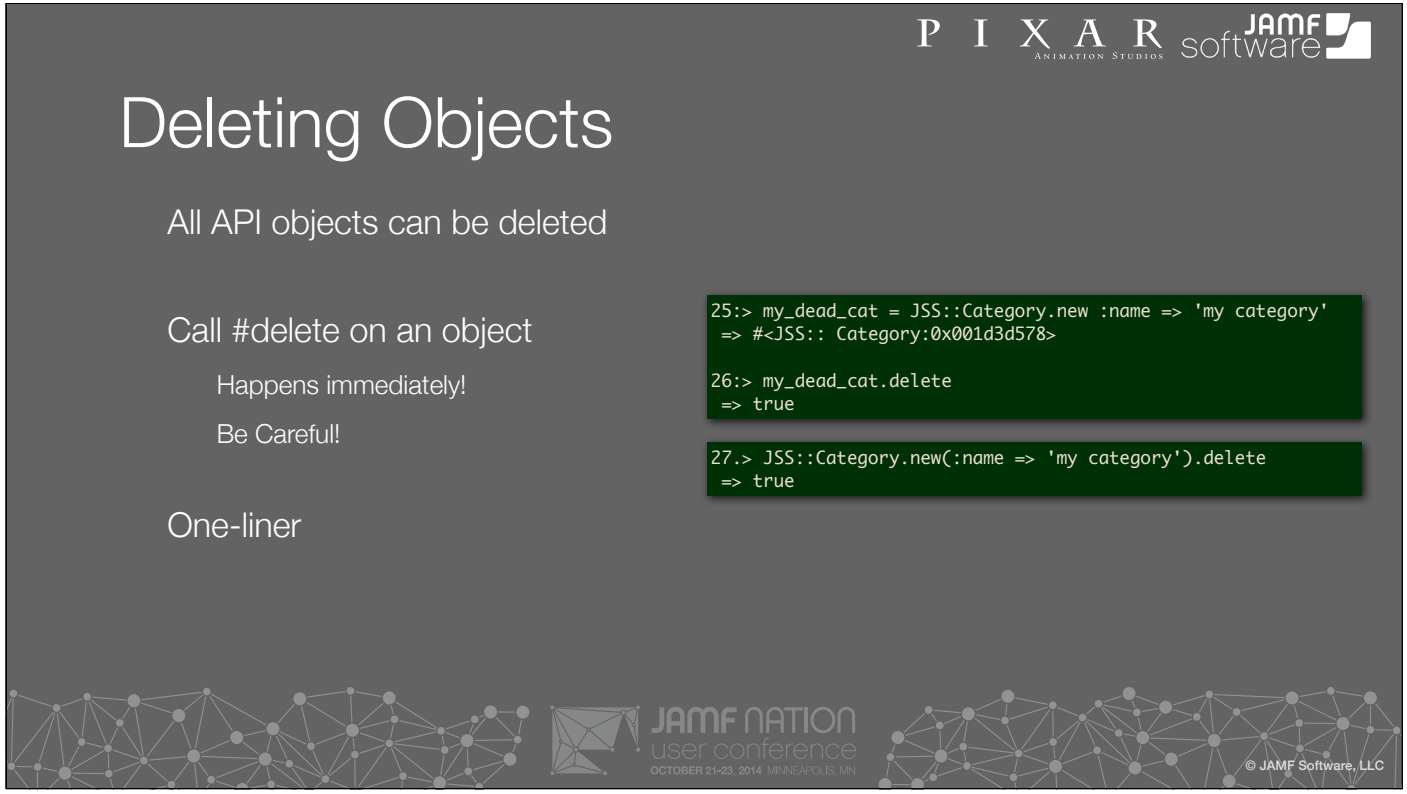
Create it in the JSS

```
20:> new_pkg = JSS::Package.new :id => :new,:name =>
"filemaker13.pkg"
 => #<JSS::Package:0x101d3d978>

21:> new_pkg.in_jss?
 => false
```

```
22.> new_pkg.category = 'Useful Software'
  => true

23:> new_pkg.os_limitations = '>=10.6.x'
  => true
```

```
24.> new_pkg.create  # Also new_pkg.save
  => 573
```


• - :id => :new, need at least a :name.
Some classes require more than just :name e.g. Groups require :type as either :smart or :static
• - This doesn't create anything in the JSS, just gives you a Ruby object to work with..
• - Save is also an alias for create

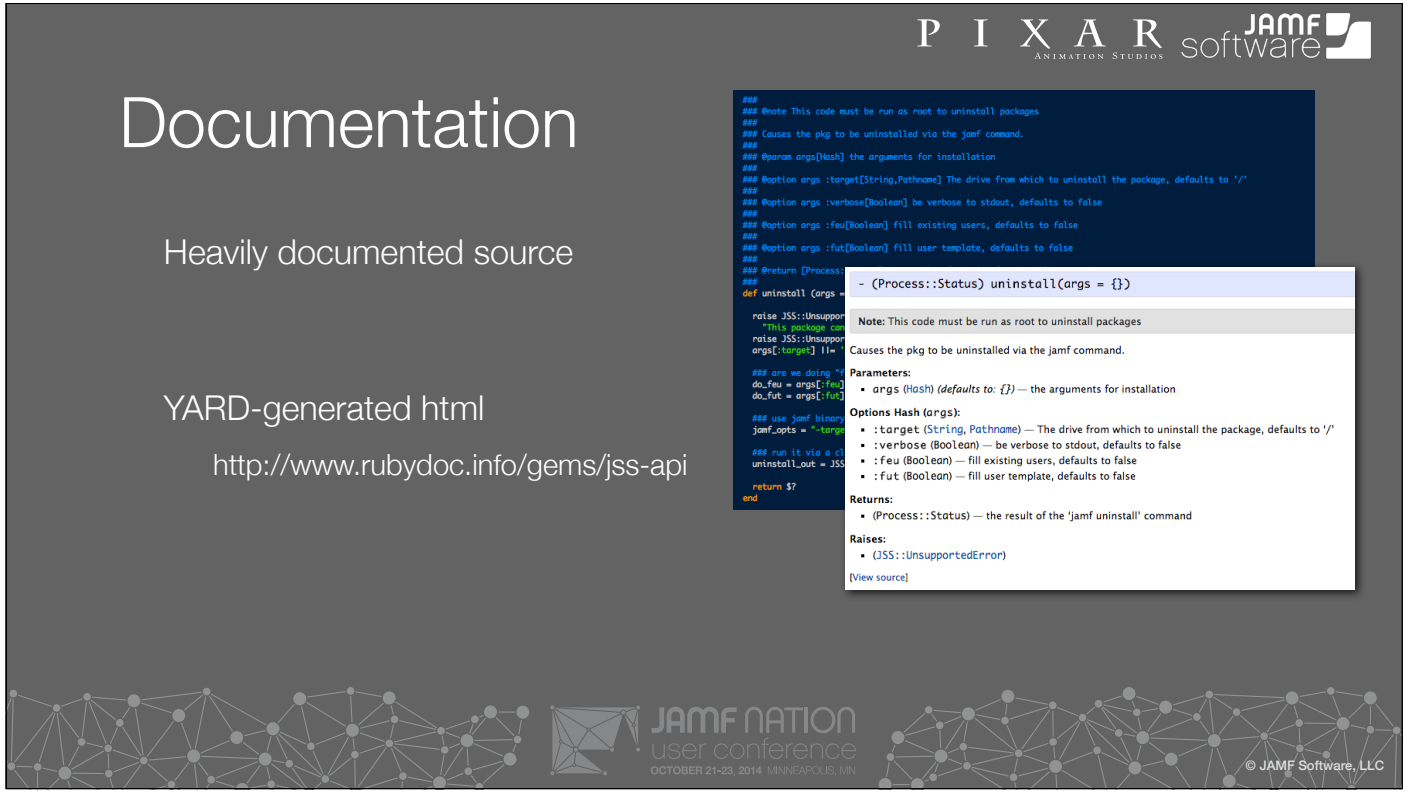


- - All API objects can be deleted
- - retrieve any object, and you can call its delete method
- - Ruby lets you retrieve and delete in one step.

The past few slides have shown the basics of using the Gem.

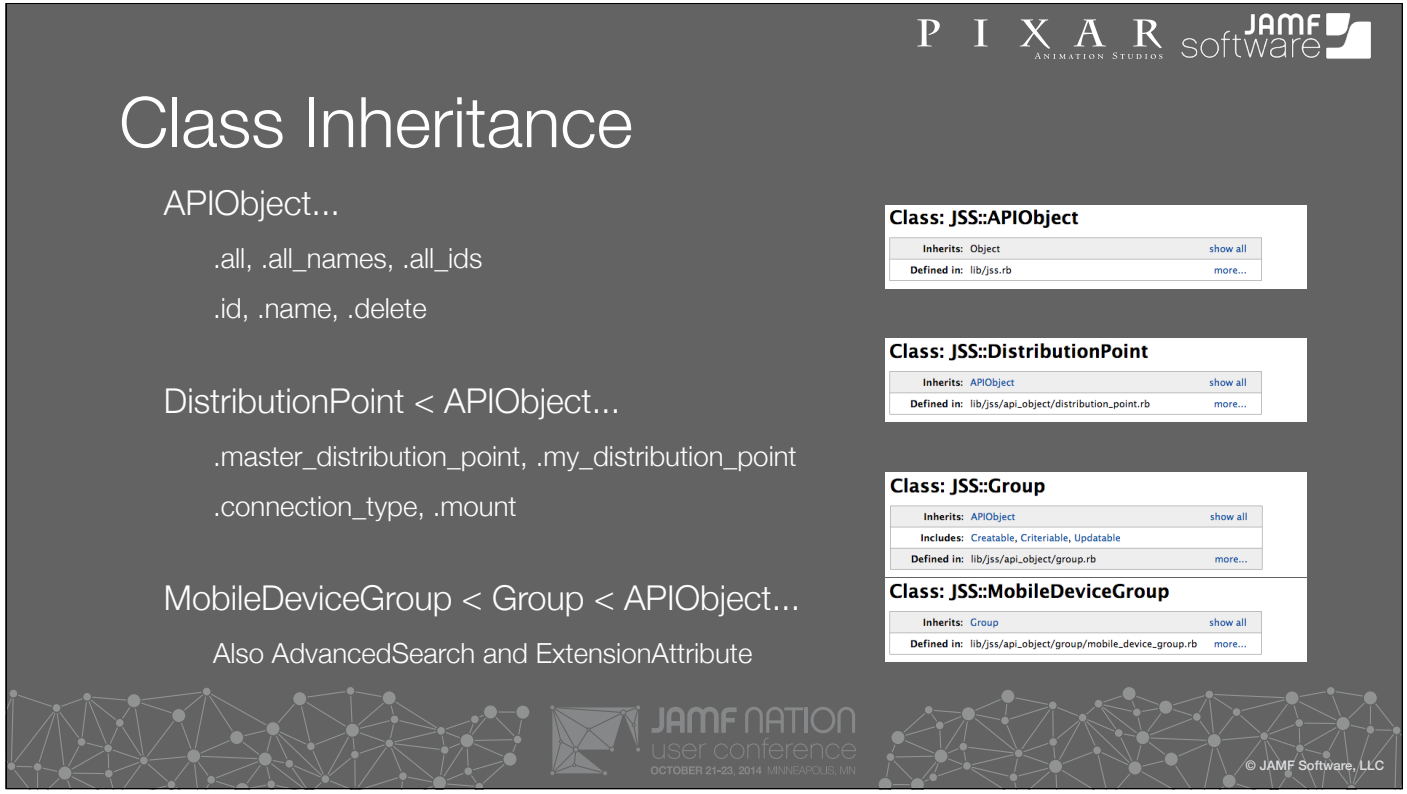
Now we'll look a little deeper, not only at using it, but also programming the Gem itself.

Of course, I've written it with our own environment and JSS in mind, but
since it's available in github, I'm hoping others will improve it by adding the functionality they need.

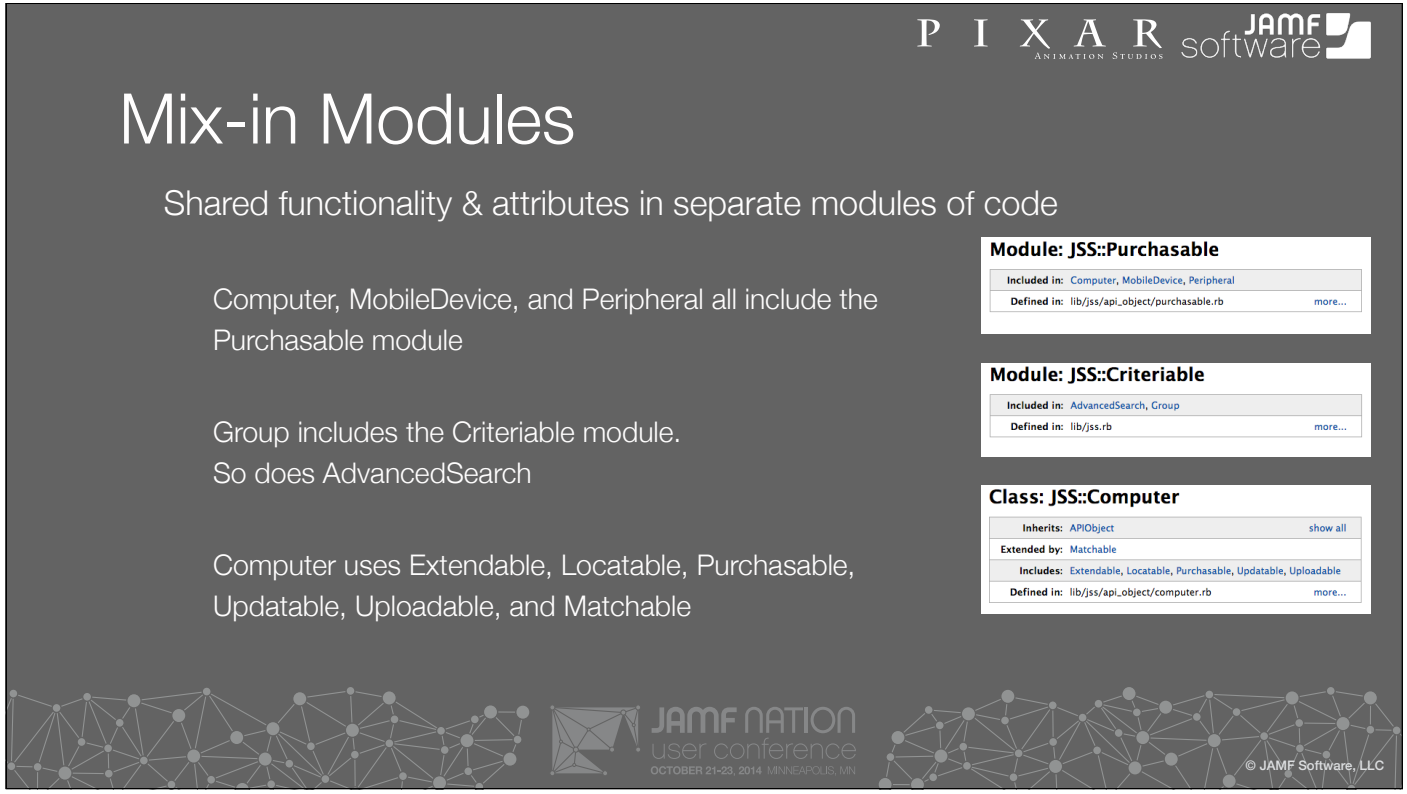


The methods I've shown so far are just the tip of the iceberg. To really use the Gem you'll have to start looking at the documentation.

And if you'd like to modify (and hopefully contribute to) the gem, the docs are *VERY* important
The source code is heavily documented. • Here's the code for the uninstall method of the Package class
The reason for the extensively formatted comments ...auto-generated • YARD HTML docs

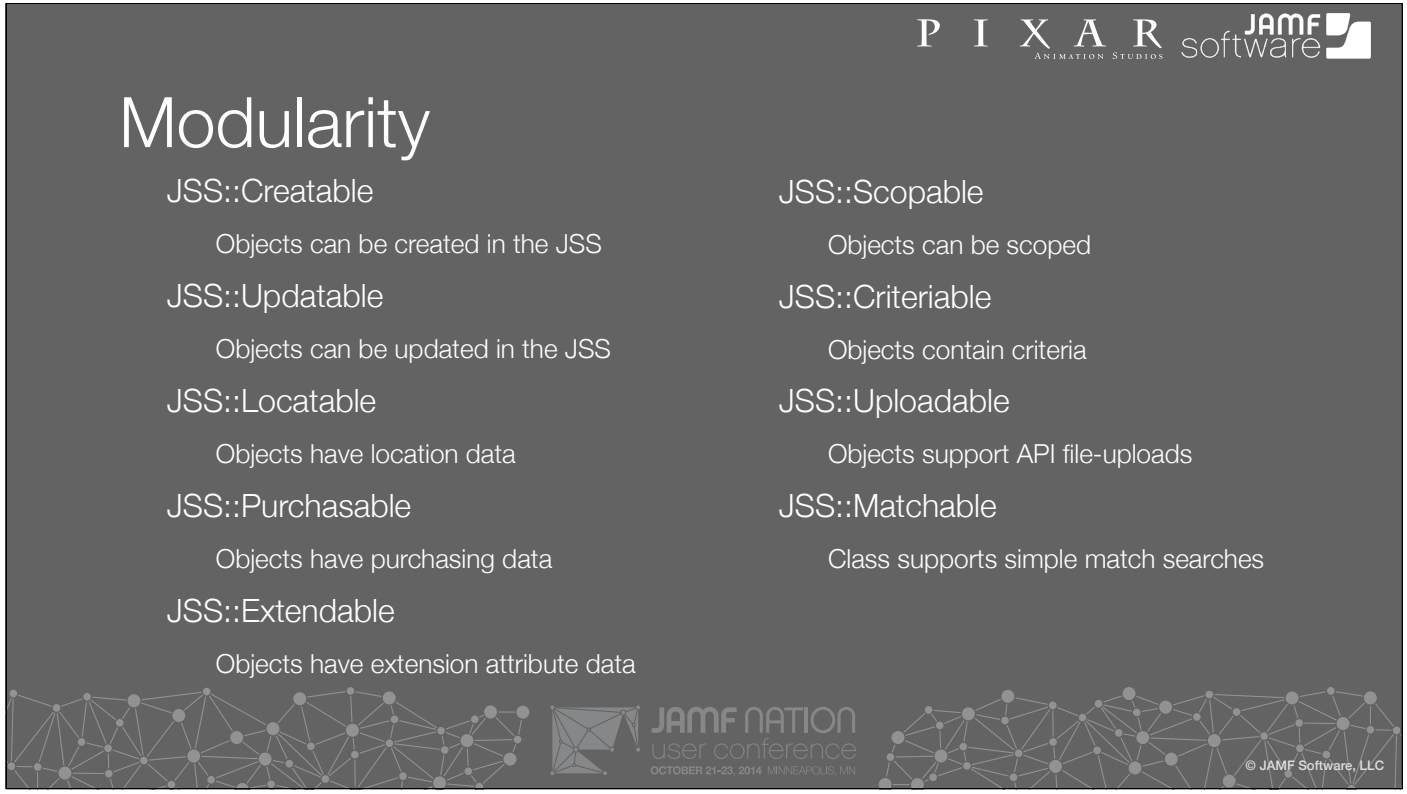


Lets talk how the Gem itself is programmed.
• JSS::APIObject is the parent class of all objects from the API, defines things they all share, like ID's and names
• All Gem objects that come from the API are children of APIObject
- #delete is defined in APIObject, which is why all of them can be deleted.
• Some objects have two levels of parents



Ruby uses mixins
- Computers, MobileDevices and Peripherals all have purchasing data
- Smart groups have criteria, so do Advanced Searches
- Computers use six mixin modules

The documentation for module discusses the requirements for mixing in.



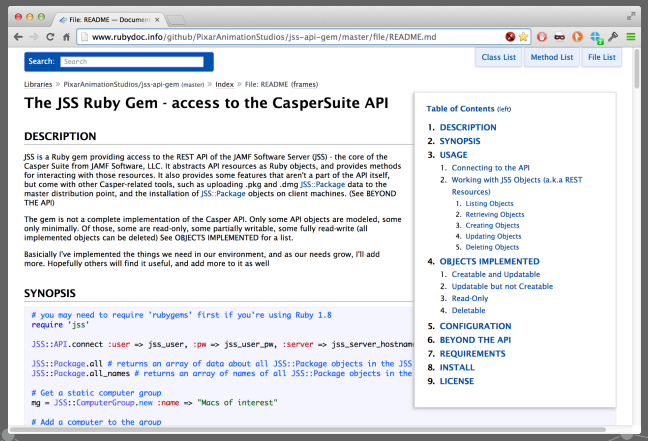
Here are the mix-in modules in the JSS at the moment.

The documentation for each gives details of how to use them.

- 9 clicks till Matchable

Documentation - again

http://www.rubydoc.info/gems/jss-api

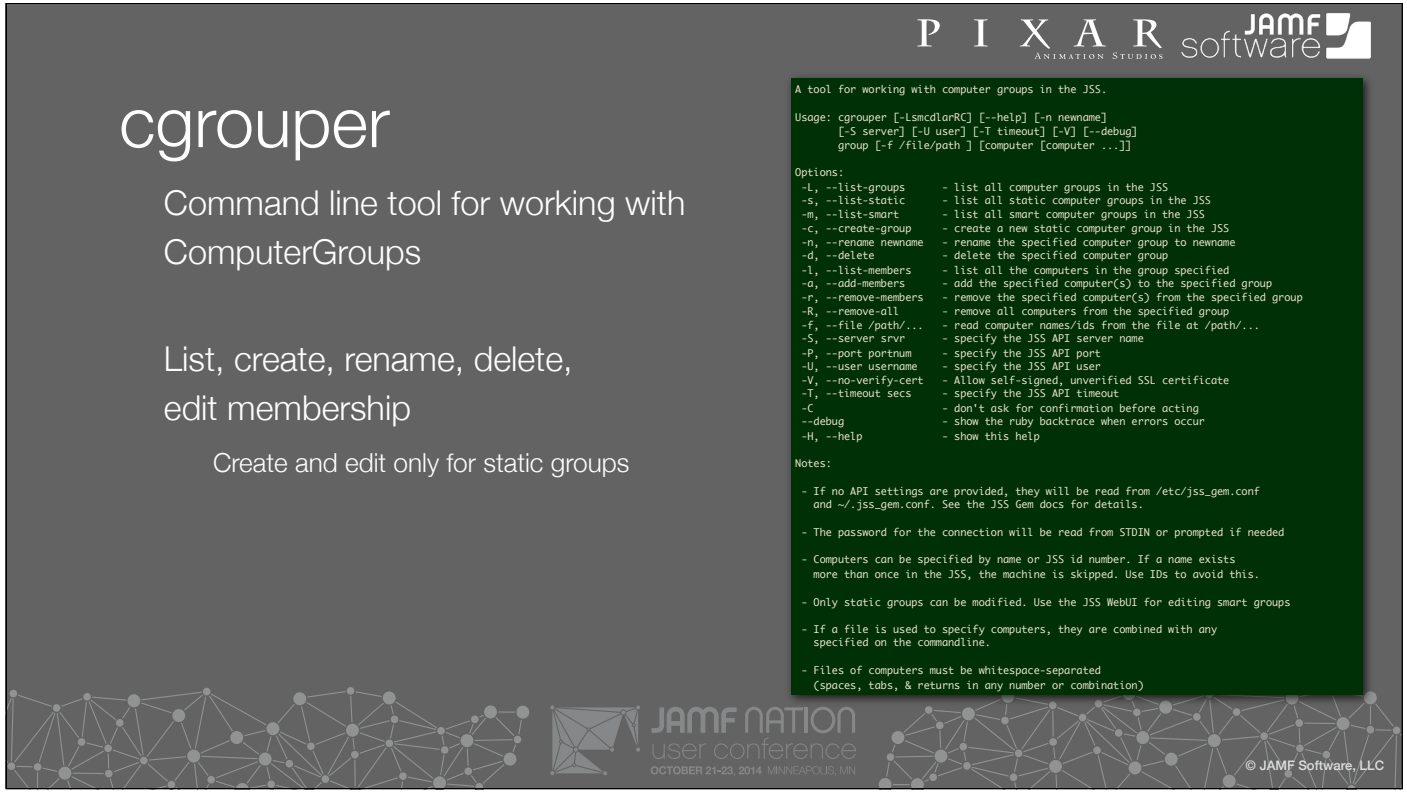


Just like other languages and frameworks, when you create subclasses and use mix-in modules

there are certain requirements your code must meet. They are all detailed in the documentation.

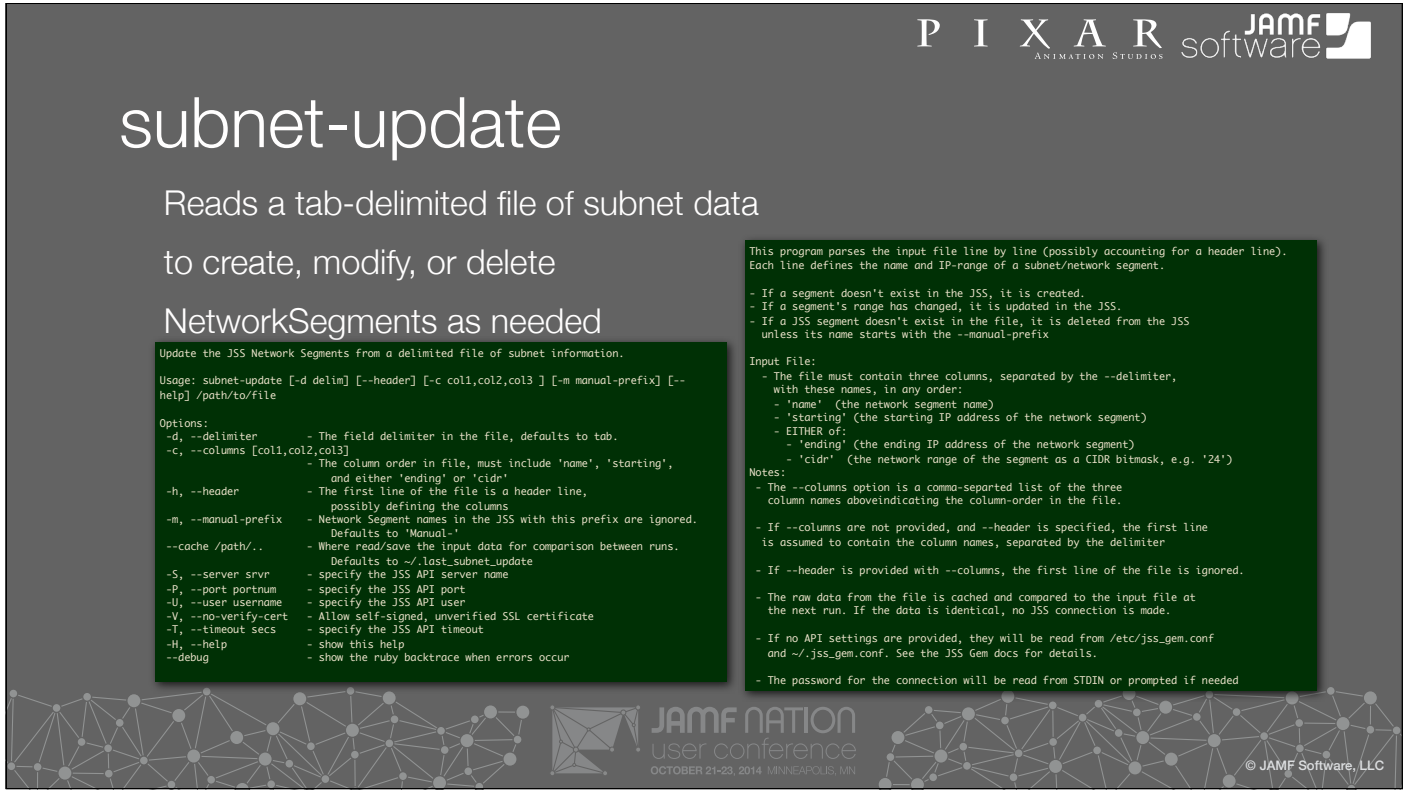
Included with the gem are a couple small example programs....

I'm just showing screenshots at the moment, but might be able to show live action during the QnA



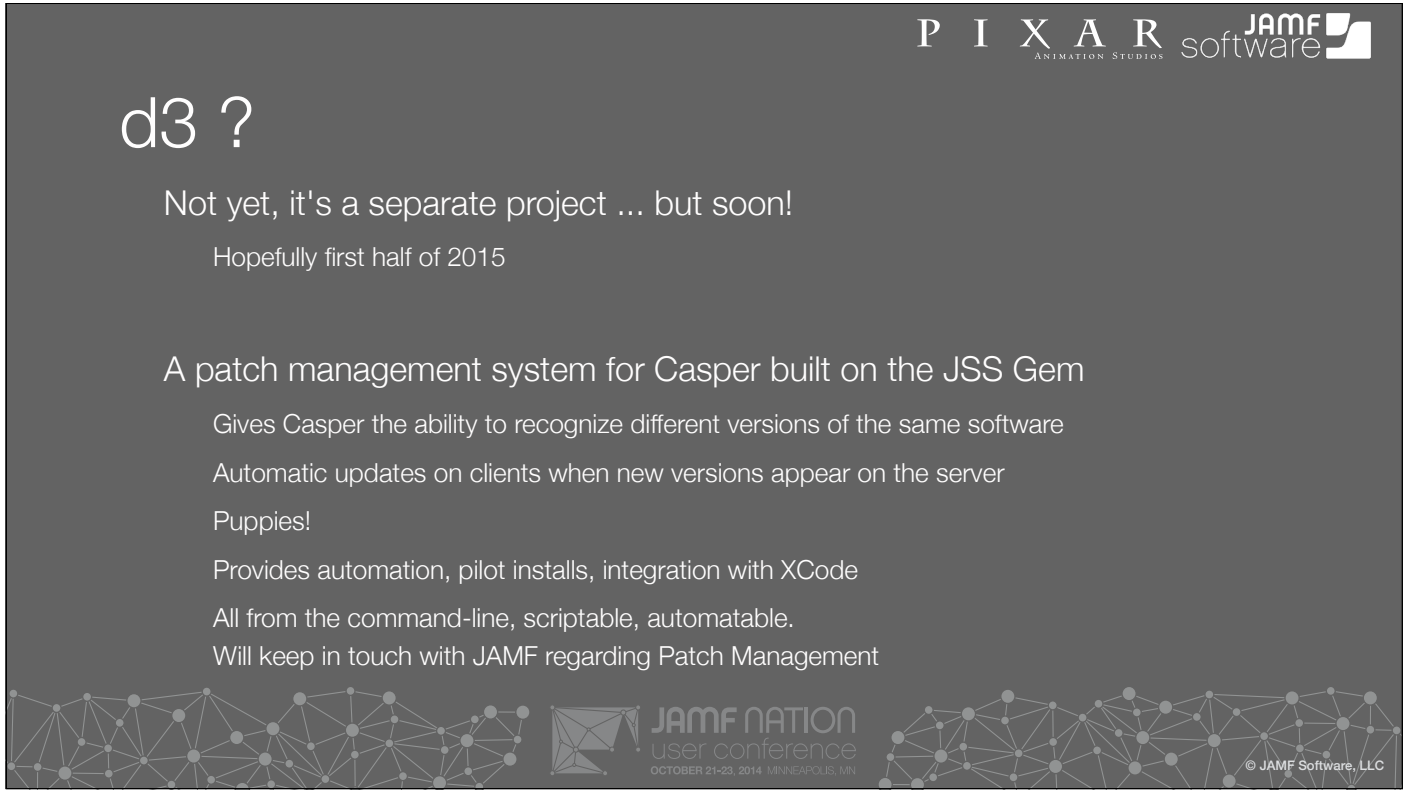
cgrouper - commandline tool for working with computer groups

-Why? We had lots of times where lists of machines came from other sources, but were needed in casper.



subnet-update: maintain NetworkSegments from an external source.

-Why? Our Network Admins maintain canonical subnet data in a git-repo, frequently updated.



d3? - •not yet.

separate, very large, project, dependent on this gem
what is it? •

PIXAR Animation Studios JAMF software

Open Source

Where to get it

Home page

http://pixaranimationstudios.github.io/jss-api-gem

Email

jss-api-gem@pixar.com

JAMF Nation

ChrisL

PIXAR Animation Studios JAMF software

Thank you!

http://pixaranimationstudios.github.io/jss-api-gem

jss-api-gem@pixar.com