CHORDS:
Cloud-Hosted Real-time Data Services for the Geosciences

Mike Daniels (NCAR), Branko Kerkez (UMich), V. Chandrasekar (CSU), Sara Graves (UAH), D. Sarah Stamps (VT), Aaron Botnick (NCAR), Charlie Martin (NCAR),
Matt Bartos (UMich), Ryan Gooch (CSU), Josh Jones (VT), Ken Keiser (UAH)

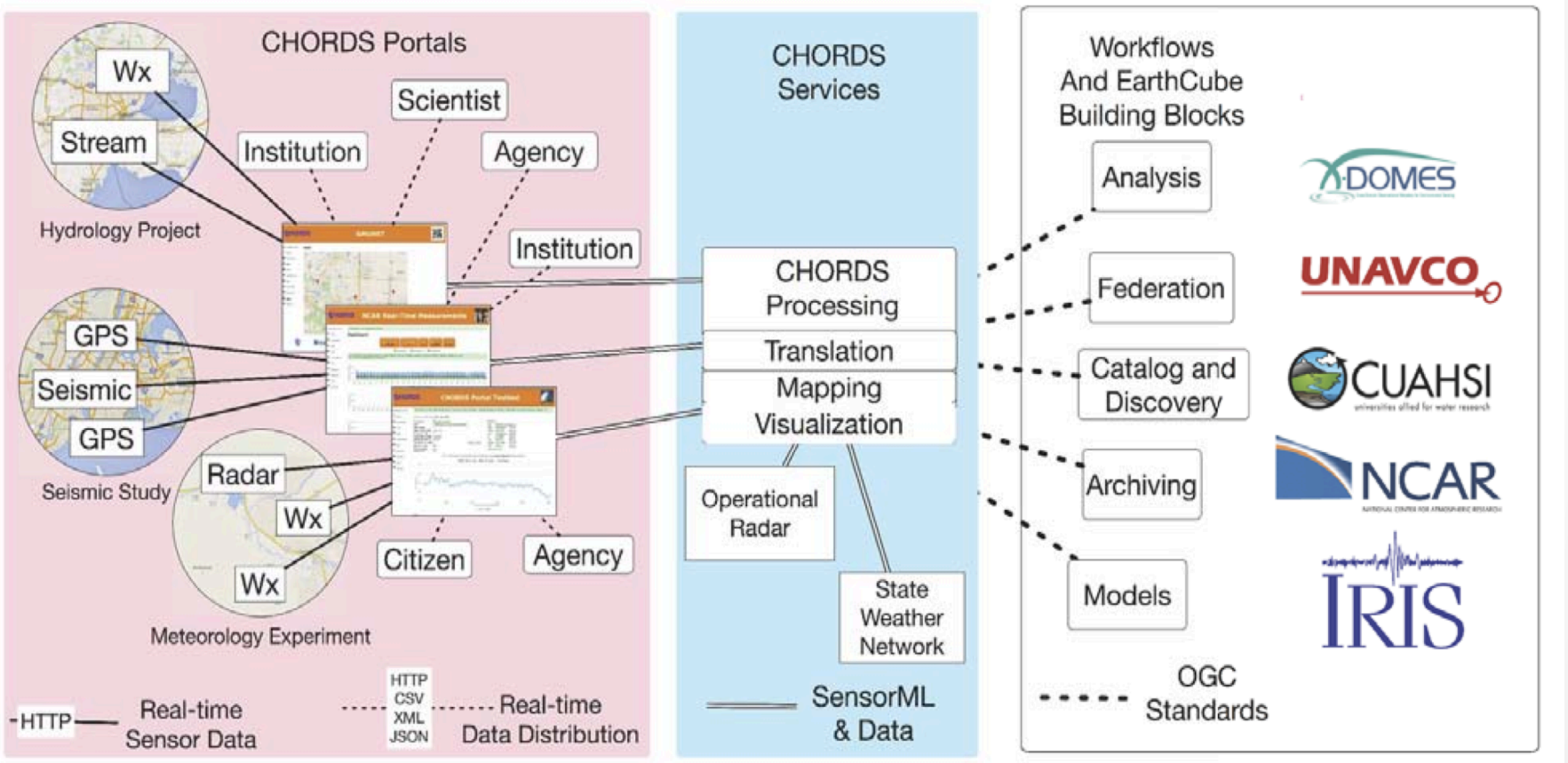
What Is CHORDS?

- System to allow anyone from citizen scientists to small or large research groups to ingest real-time sensor data
- Those data may eventually be used in model data assimilation or research in seemingly unrelated fields via their online availability
- CHORDS aims to minimize need for dedicated IT resources
- Free/open-source and deployable in different environments
- Provides for standards-compliant data/metadata interchange
  - GeoJSON, GeoCSV, SensorML
  - Coming Soon:
    - JSON-LD, TimeSeriesML, CF-Conventions
    - Community-specific ontologies/vocabularies

Scientists Want To Do Science!

SensorML
Web Mapping Service
GeoJSON
Climate Forecasting Conventions
TimeSeriesML
GCIS Ontology
Sensor Observation Service
OM-JSON
GeoCSV
…

CHORDS Architecture



NCAR EarthCube CHORDS

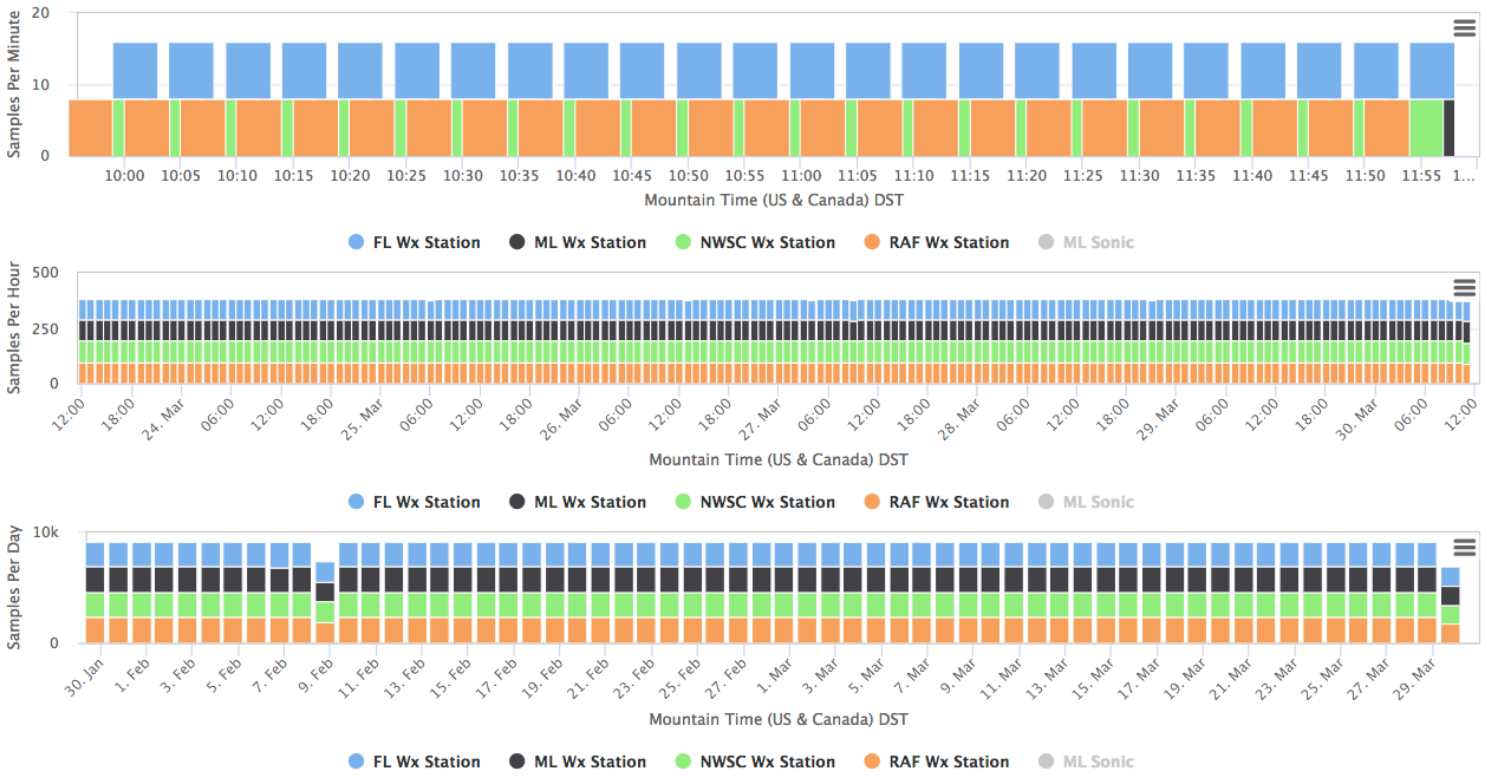
About Help Dashboard Data Sites Instruments Geo Visualization Sign In

Dashboard

Measurements	Instruments	Sites	Storage	DB Expires	Uptime
18,394,755	5	4	132.6 MB	2016-04-01	18 days

FL Wx Station (1) ML Wx Station (2) NWSC Wx Station (3) RAF Wx Station (5) ML Sonic (6)

http://portal.chordsrt.com/measurements/url_create?instrument_id=6&wdir=058&wspd=003.12&status=60&key=hidden



Release: development (5aa6e1b) (2018-03-12 19:16:55 UTC)

NCAR EarthCube CHORDS

About Help Dashboard Data Sites Instruments Geo Visualization Sign In

http://portal.chordsrt.com/measurements/url_create?instrument_id=1&wdir=214&wspd=1.8&wmax=3.7&tdry=9.5&rh=42.7&pres=836.6&raintot=18.30&batv=17.9&key=hidden

FL Wx Station (id: 1) located at NCAR Foothills Lab

Description: Vaisala WXT510 weather station. See http://www.eol.ucar.edu/weather/weather_fl/station.html.
This instrument is designated as active: Yes
(If 'No', the instrument will not appear in the dashboard.)

Measurements

977602 measurements were reported.
This instrument is expected to report a measurement every 300 seconds.
The first measurement was measured at 2017-01-27 21:23:34 UTC.
The last measurement is 4 minutes old. It was measured at 2018-03-30 17:51:53 UTC.

Data Access and Downloads

Data ingest/fetch
Data URLs

Download current day as
XML, GeoJSON, GeoCSV

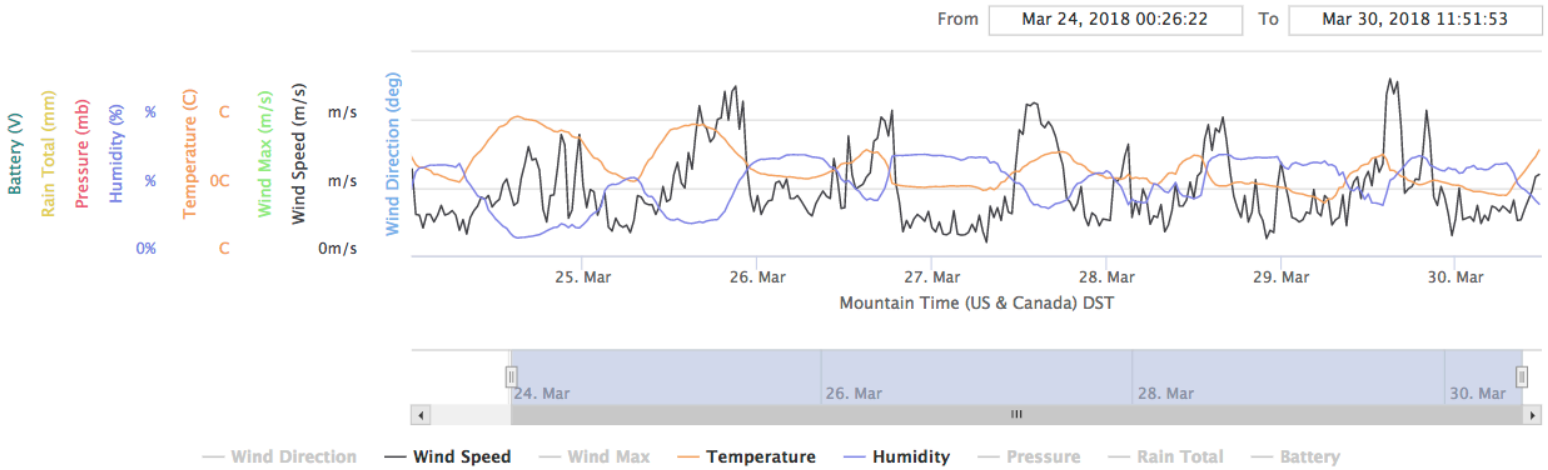
Instrument Description
SensorML

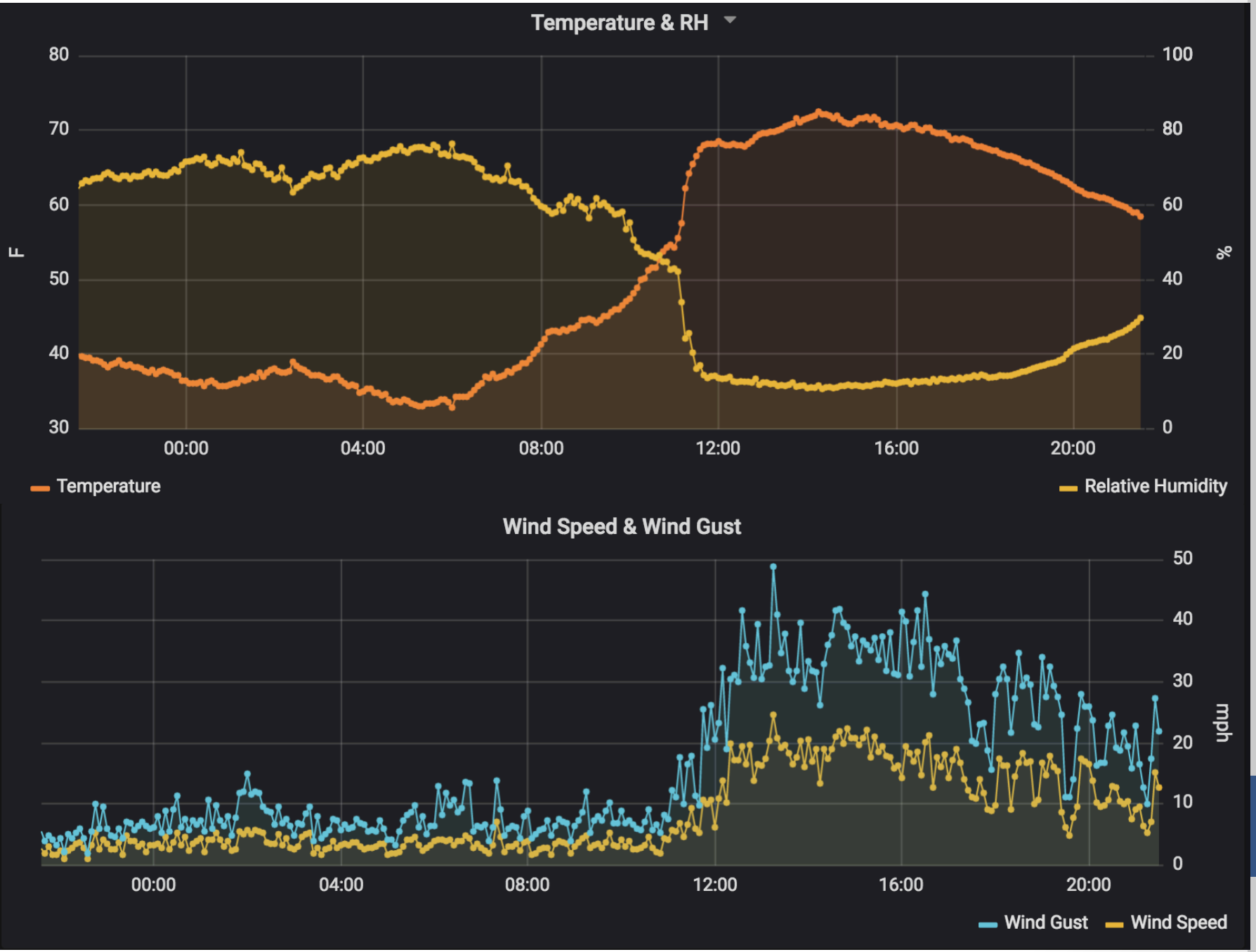
Instrument Metadata

InfluxDB Tags

Plot measurements for the last 1 weeks

FL Wx Station – Live Data




Temperature & RH
80
70
60
50
40
30
F
100
80
60
40
20
0
%
00:00
04:00
08:00
12:00
16:00
20:00
Temperature
Relative Humidity
Wind Speed & Wind Gust
50
40
30
20
10
0
mph
00:00
04:00
08:00
12:00
16:00
20:00
Wind Gust
Wind Speed

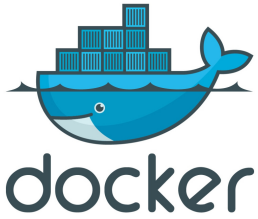
CHORDS and EarthCube

- EarthCube is a National Science Foundation initiative that seeks build and support software, interfaces and tools (e.g. "cyber-infrastructure") to transform research across the geosciences, including Polar, Atmosphere, Hydrology, Solid Earth and Ocean sciences

- CHORDS provides a means to integrate data from disparate sources and becomes a conduit of those data into other analysis and display software/tools, with an emphasis on those found within EarthCube

CHORDS Technologies

- Docker
- Amazon AWS EC2
- Grafana
- MySQL
- InfluxDB
- Ruby
- Ruby on Rails (RoR)



Ruby Language

- Created in the mid-90s by Yukihiro "Matz" Matsumoto
- Dynamically typed scripting language
- Pure Object Oriented
- Borrows concepts from Smalltalk, Perl, Eiffel, Ada, and Lisp
- Matz was quoted as wanting Ruby to be:
  "more powerful than Perl, and more object-oriented than Python"
- Various interpreters: MRI (CRuby), JRuby, IronRuby, etc
- RubyGems
  - Library of third party libraries/packages
  - Bundler – package used to handle install/update of packages via Gemfile and Gemfile.lock

Ruby on Rails (RoR)

- Created by David Heinemeier Hansson of 37Signals/Basecamp
- Initial release in 2005
- Server-side web application framework written in pure Ruby
- MVC / RESTful paradigm
- ActiveRecord pattern
- Influenced other web frameworks, such as:
  - Django (Python)
  - Laravel (PHP)
  - Phoenix (Elixir)
  - Sails.js (Node.js)

Who's Using RoR?

1. Twitter (originally, but now using Java)
2. Basecamp
3. Yellow Pages
4. Hulu
5. Slideshare
6. GitHub
7. Shopify
8. Groupon
9. Urban Dictionary
10. AirBnB

https://prograils.com/posts/top-10-famous-sites-built-with-ruby-on-rails

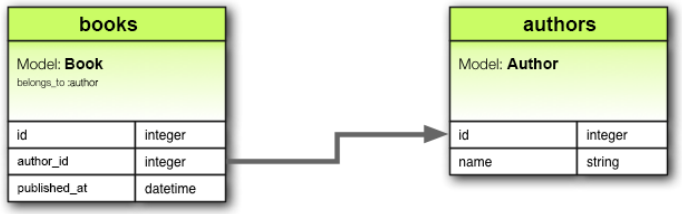
The Rails Doctrine

1. Optimize for programmer happiness
2. Convention over Configuration
3. The menu is omakase
4. No one paradigm
5. Exalt beautiful code
6. Provide sharp knives
7. Value integrated systems
8. Progress over stability
9. Push up a big tent

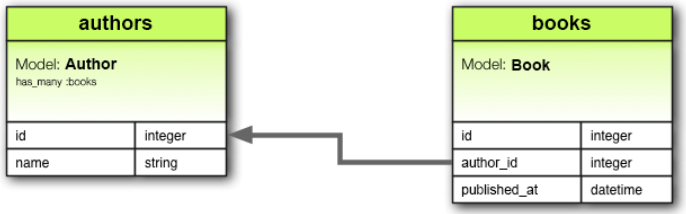
http://rubyonrails.org/doctrine/

Rails Conventions

- Don't Repeat Yourself (DRY)
- Scaffolding / Templates
- ActiveRecord and Database Structure
  - id => default primary key
  - books (table) => Book (class)
  - associations
    - authors (table) => Author (class)
      - Author => has_many :books
      - author2.books => returns array of Book objects
    - Book => belongs_to :author
      - book2.author => returns Author object (one to one association)



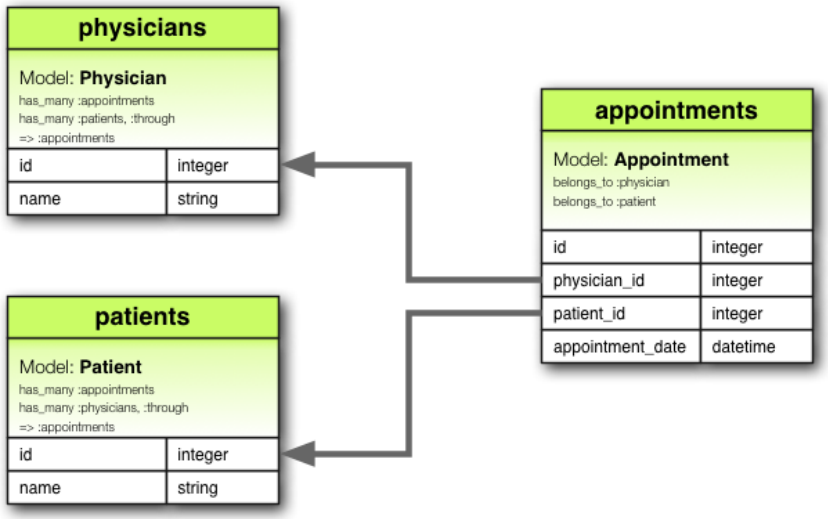

```
class Book < ActiveRecord: :Base
  belongs_to :author
end
```




```
class Author < ActiveRecord: :Base
  has_many :books
end
```

Association Types

```
belongs_to
has_one
has_many
has_many :through
has_one :through
has_and_belongs_to_many
```




```
class Physician < ActiveRecord::Base
  has_many :appointments
  has_many :patients, :through => :appointments
end

class Appointment < ActiveRecord::Base
  belongs_to :physician
  belongs_to :patient
end

class Patient < ActiveRecord::Base
  has_many :appointments
  has_many :physicians, :through => :appointments
end
```


Other Rails Features

- Test Frameworks: TestUnit and RSpec
- Multiple environments: development, test, production, etc
- JavaScript: jQuery, Unobtrusive JS, CoffeeScript
  - Common helpers for forms and links, remote: true
  - Confirmation boxes in html templates
- Embedded Ruby (ERB) HTML templates and HAML
  - Form helpers
  - Partials

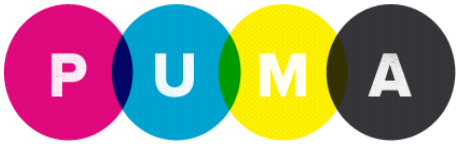
```
%h1 Editing Instrument
= render 'form'
= link_to 'Show', @instrument
|
= link_to 'Back', instruments_path
```


Other Rails Features (cont'd)

- Fat models, lightweight controllers
  - Actions include basic CRUD operations
  - Custom actions may be used but must be routed manually
- Model validations => validate presence, values, etc
- Database migrations
  - Allow easy changes to database schema and versioning
  - Uses a DSL, but supports Ruby syntax
  - Can be reversible, if written correctly

Web Application Servers

- Typical usage is alongside Apache Web Server or Nginx
- Unicorn
  - One of the oldest and most stable, used by GitHub
  - Forked processes to handle multiple requests
  - Does not handle slow clients on its own, Nginx can be used to buffer requests
  - Forking uses more memory than threading
  - Zero downtime deploys
- Puma
  - Current Rails 5.x default
  - Threads for multiple requests, but only works if code is thread-safe
  - Can use worker processes if not thread-safe
  - Handles slow clients
  - Zero downtime deploys
- Apache Phusion Passenger
  - Tied directly into Apache Web Server
  - Handles slow clients
  - Configuration is more difficult
  - Best when you are serving web apps in written in different languages
  - Many features only available with enterprise license



Rails/CHORDS and APIs

- Can serve many data types: HTML, JSON, XML, CSV, etc
- While DRY concepts are better, versioned APIs offer better maintainability with separate controllers for views and the API
  - POST /api/v2/instruments
- JBuilder and others allow JSON API templates like ERB partials
- When using Rails naming conventions, most of the boilerplate associated with parsing requests is removed

```ruby
def create
  authorize! :manage, Instrument

  @instrument = Instrument.new(instrument_params)

  respond_to do |format|
    if @instrument.save
      format.html { redirect_to @instrument, notice: 'Instrument was successfully created.' }
      format.json { render :show, status: :created, location: @instrument }
    else
      format.html { render :new }
      format.json { render json: @instrument.errors, status: :unprocessable_entity }
    end
  end
end
```

JSON Templates Using JBuilder

```
json.array!(@instruments) do |instrument|
  json.extract! instrument, :id, :name, :site_id
  json.url instrument_url(instrument, format: :json)
end
```

```
json.extract! @instrument, :id, :name, :site_id, :created_at, :updated_at
```

```
[
    {
        "id": 1,
        "name": "Weather Station",
        "site_id": 1,
        "url": "http://localhost:3000/instruments/1.json"
    },
    {
        "id": 2,
        "name": "WxStation2",
        "site_id": 1,
        "url": "http://localhost:3000/instruments/2.json"
    }
]
```


CHORDS Future Work

- Refactor and bring code in line with Rails conventions
- Revamp user authorization and roles
- Implement a full API
- Improve portal configuration management
- Complete SensorML and GeoJSON implementation
- Plan for adding real-time, 2D image data (e.g. radar, satellite)
- Improve UI
- Implement EarthCube P418 Linked-Data (JSON-LD)

Questions?

botnick@ucar.edu

chordsrt.com

CHORDS

CHORDS is funded by the US National Science Foundation's EarthCube program, grants 1639750, 1639720, 1639640, 1639570 and 1639554.