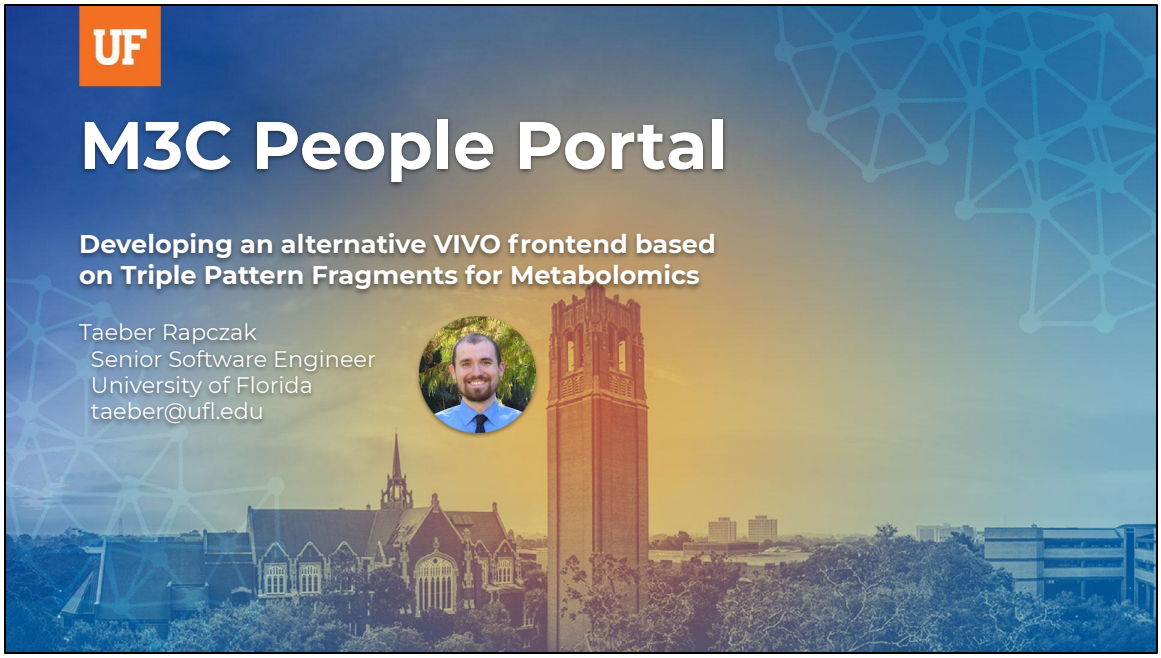
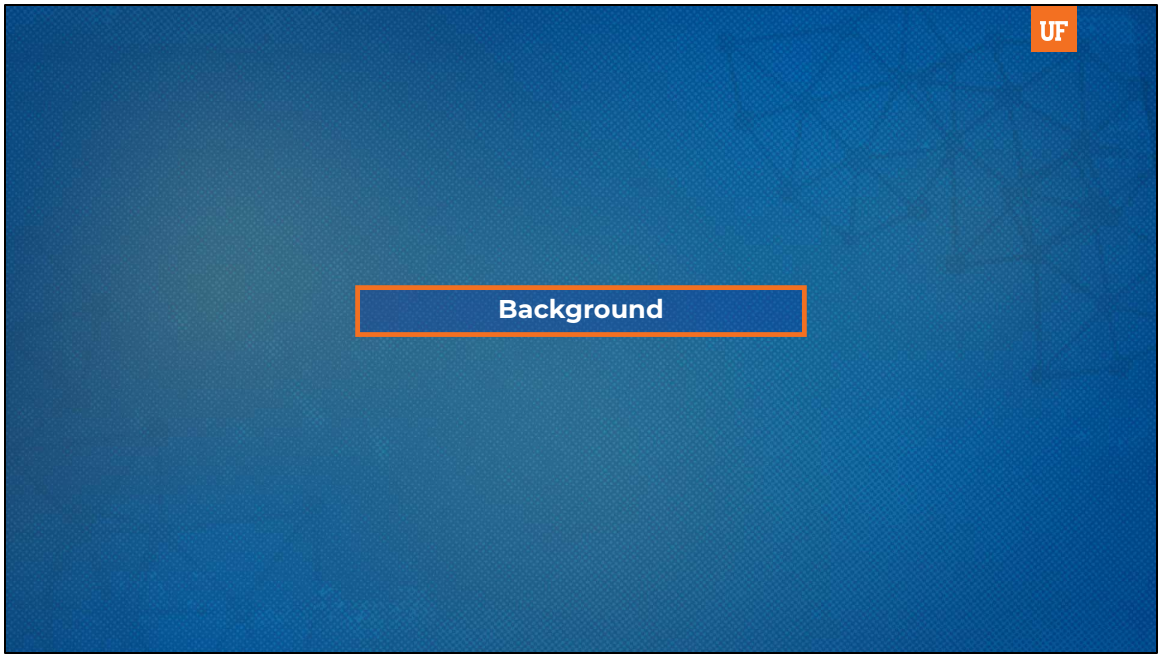


Hello. My name is Taeber Rapczak.

I'm a Senior Software Engineer at the University of Florida.

Today I'm presenting the M3C People Portal: Developing an alternative VIVO frontend based on Triple Pattern Fragments for Metabolomics.



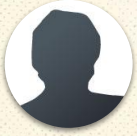
Just a little background before we get started.

Collaborators

UF | UNIVERSITY of FLORIDA

Christopher P. Barnes

Kevin S. Hanson


Samantha G. Emerson

Michael Conlon

Alyssa Kelly

Hunter Jarrell

I work with a team that includes all of these fine people at the University of Florida.

M3C

Metabolomics Consortium Coordinating Center

Our team works with the M3C, or Metabolomics Consortium Coordinating Center.

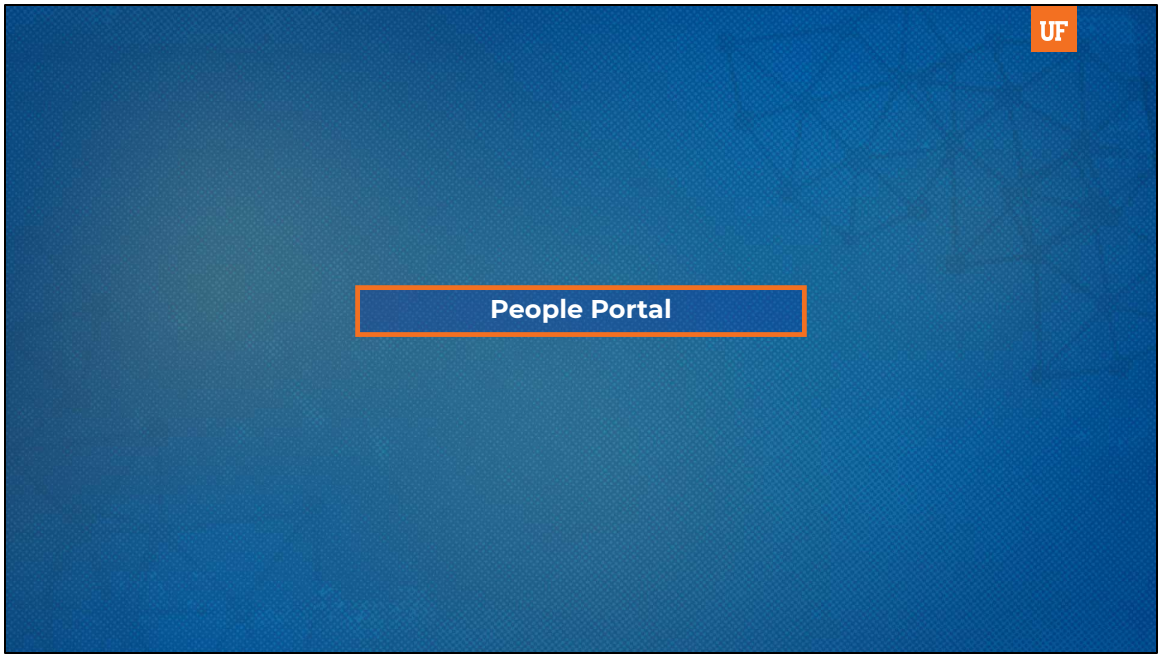
The M3C was established with a grant from the NIH Common Fund Metabolomics program.

Metabolomics

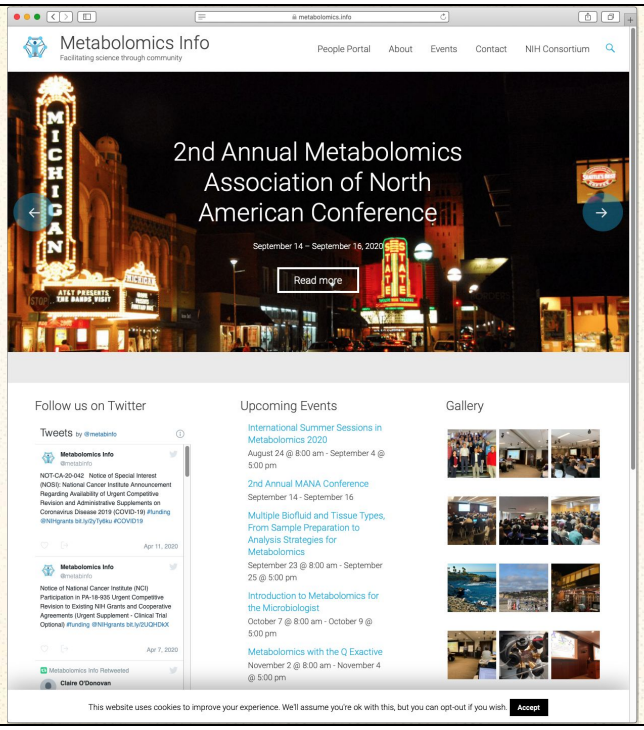
Since this is a technical talk, it's not vital that you know about Metabolomics, so let me give the quickest overview of what it is.

Metabolomics is the scientific study of metabolites present within an organism, cell, or tissue.

Human metabolites are small molecules found in tissue as a result of our natural metabolism, medicines we take, food we eat, and the environment we live in.

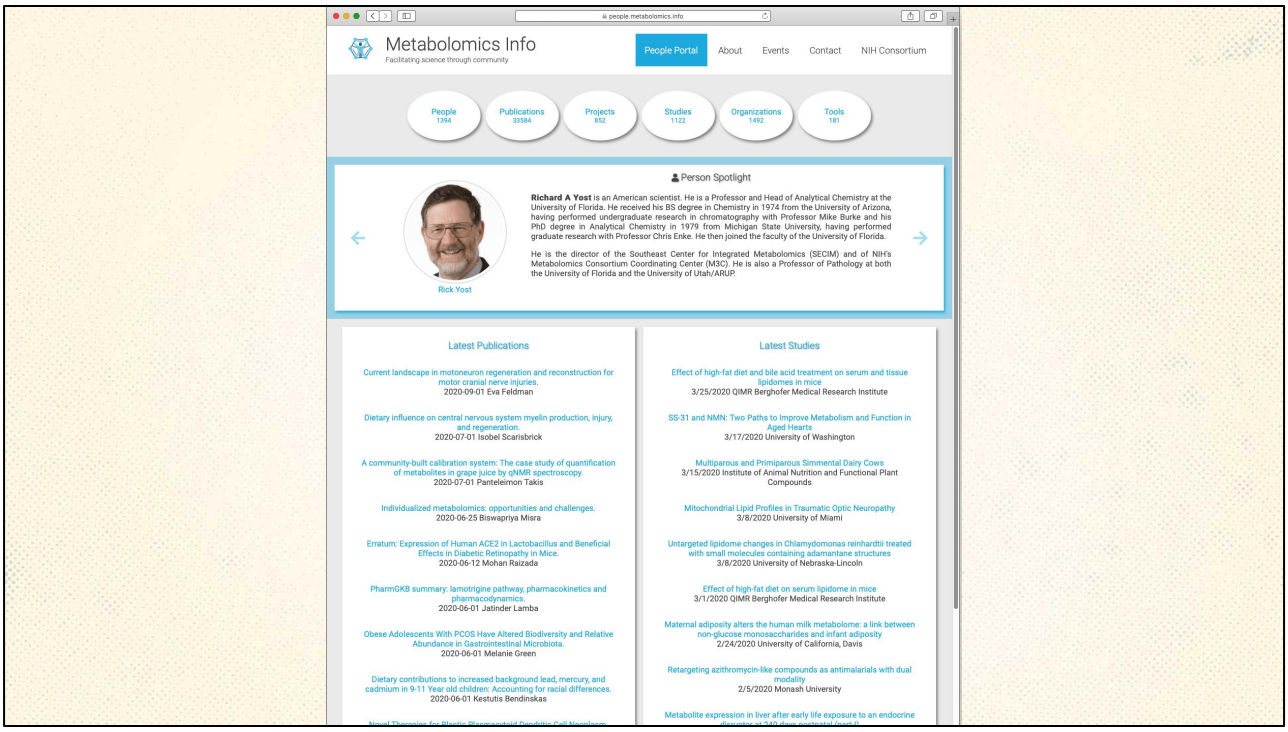


The People Portal



Well, let's start off with screenshots. People like screenshots.

Part of what we do for the M3C is maintain its informational website at metabolomics.info.



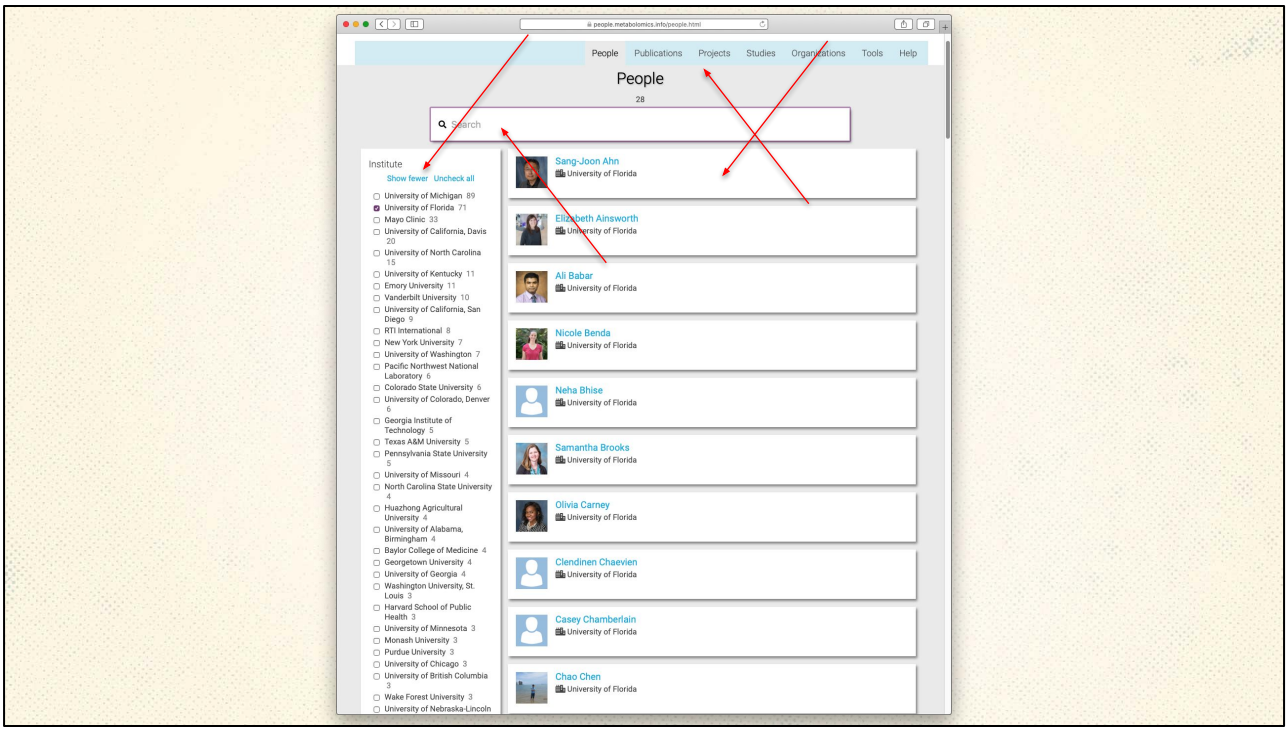
This is the “People Portal”, which is where VIVO gets involved.

Our group also runs the University of Florida’s main VIVO site, so we knew it would be an appropriate tool to use for the M3C People Portal.

In the backend, we’re running version 1.10, but we are only using the ontology, Jena inferencer and SDB tools, and the Triple Pattern Fragments server.

The frontend you see is plain HTML, CSS, and JavaScript.

No frameworks; just a few shims to support Internet Explorer 11 and Font Awesome for some iconography.



Here's another screenshot that shows our People Listing Page.

Animation Click
Notice that the navigation bar along the top lists our main concepts:

- People, Publications, Projects, Studies, Organizations, and Tools.

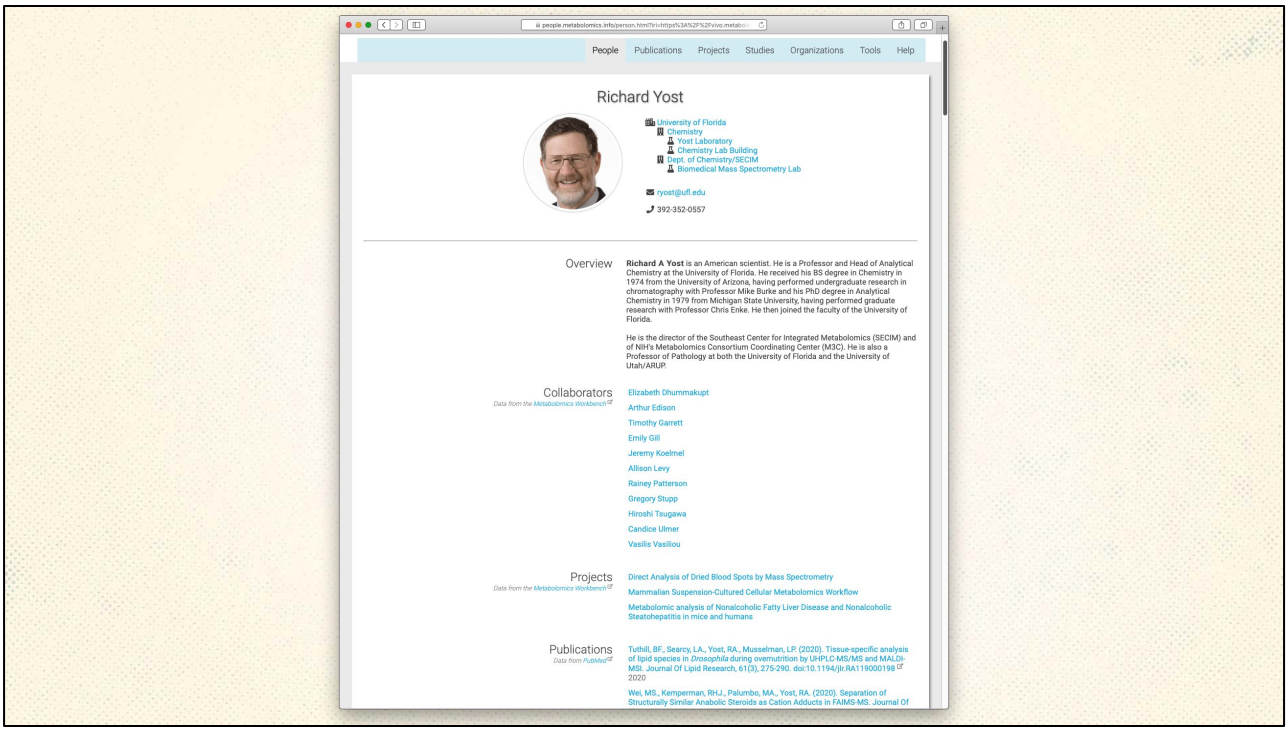
Each concept has a Listing Page like this.

There's also Profile Page with details for a specific instance of a concept.

Animation Click
The Listing pages have a search bar at the top that can perform client-side filtering.

Animation Click
On the left, there are also facets which can also narrow down

Animation Click
Finally, the Items—or Results—are on the right.



The last screenshot shows our Person Profile Page, specifically the one for Dr Richard Yost.

Again, all of this is rendered using HTML, CSS, and JavaScript pointed at a TPF server.

Triple Pattern Fragments

So what is TPF? Triple Pattern Fragments.

Subject Predicate Object

Think back to when you were introduced to VIVO and the concept of a graph database.

You were probably told that a graph consists of triples comprising a Subject, Predicate, and Object.

TPF is a *queryable*
Linked Data interface

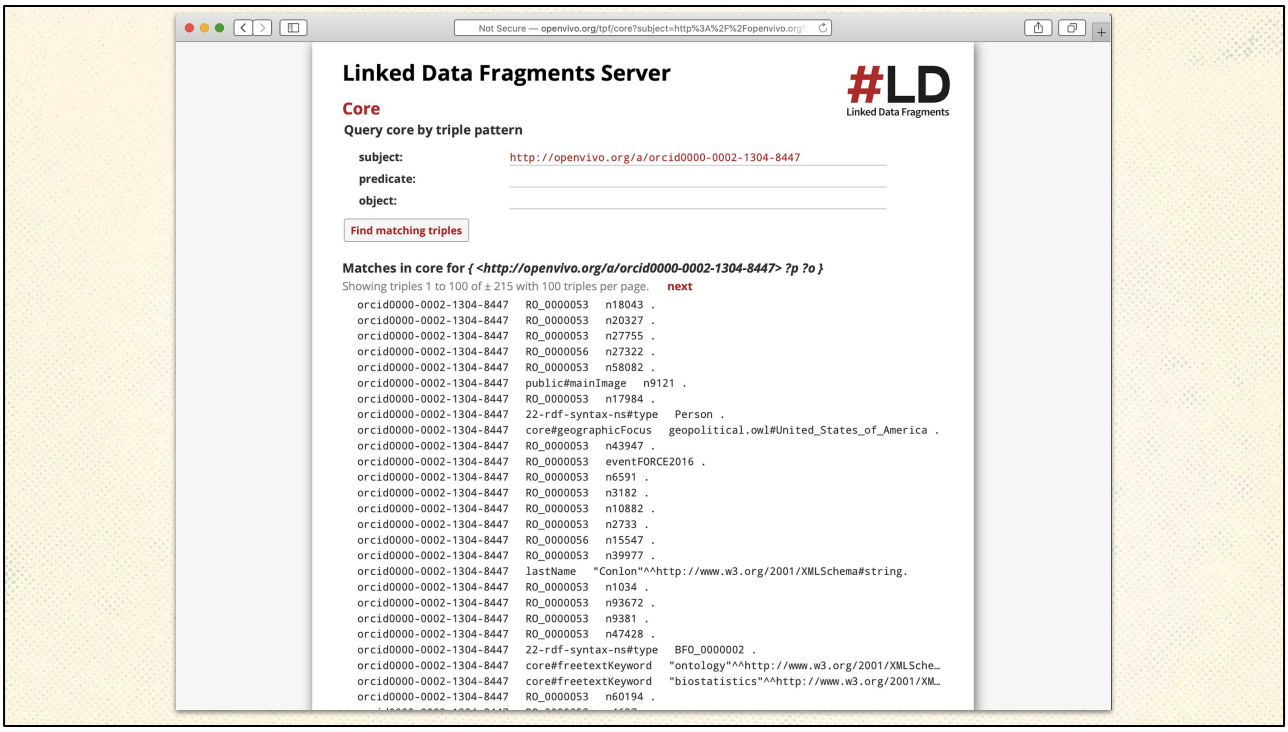
The TPF specification uses that level of abstraction in representing Linked Data and defines a queryable interface, including one for HTTP.

```
http://openvivo.org/tpf/core
    ?subject=
       http://openvivo.org/a/orcid0000-0002-1304-8447
    &predicate=
    &object=
```

To query an HTTP-TPF server, you specify the subject, predicate, and object you want using regular HTTP query parameters.

So, let’s see an example using OpenVIVO.

You can see the OpenVIVO TPF endpoint along with Mike Conlon’s URI for the subject and empty values for predicate and object.



Here’s a screenshot of that URL in a browser.

This is the HTML view.

For the purposes of developing our own frontend, however, we don’t want the HTML.

Instead we want the raw triples, so to speak.

```
...
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000053> <http://openvivo.org/a/n18043> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000053> <http://openvivo.org/a/n20327> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000053> <http://openvivo.org/a/n27755> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000056> <http://openvivo.org/a/n27322> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000053> <http://openvivo.org/a/n58082> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://vitro.mannlib.cornell.edu/ns/vitro/public#mainImage>
<http://openvivo.org/a/n9121> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000053> <http://openvivo.org/a/n17984> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://xmlns.com/foaf/0.1/Person> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://vivoweb.org/ontology/core#geographicFocus>
<http://aims.fao.org/aos/geopolitical.owl#United_States_of_America> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000053> <http://openvivo.org/a/n43947> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000053>
<http://openvivo.org/a/orcid0000-0002-1304-8447-http://openvivo.org/a/eventFORCE2016> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000053> <http://openvivo.org/a/n6591> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000053> <http://openvivo.org/a/n3182> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000053> <http://openvivo.org/a/n10882> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000053> <http://openvivo.org/a/n2733> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000056> <http://openvivo.org/a/n15547> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000053> <http://openvivo.org/a/n39977> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://xmlns.com/foaf/0.1/lastName>
"Conlon"^^<http://www.w3.org/2001/XMLSchema#string> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000053> <http://openvivo.org/a/n1034> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000053> <http://openvivo.org/a/n93672> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000053> <http://openvivo.org/a/n9381> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://purl.obolibrary.org/obo/RO_0000053> <http://openvivo.org/a/n47428> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://purl.obolibrary.org/obo/BFO_0000002> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://vivoweb.org/ontology/core#freetextKeyword>
"ontology"^^<http://www.w3.org/2001/XMLSchema#string> .
<http://openvivo.org/a/orcid0000-0002-1304-8447> <http://vivoweb.org/ontology/core#freetextKeyword>
"biostatistics"^^<http://www.w3.org/2001/XMLSchema#string> .
...
```

Fortunately, by setting the result type we want, we can get the same triples in N-Triples format.

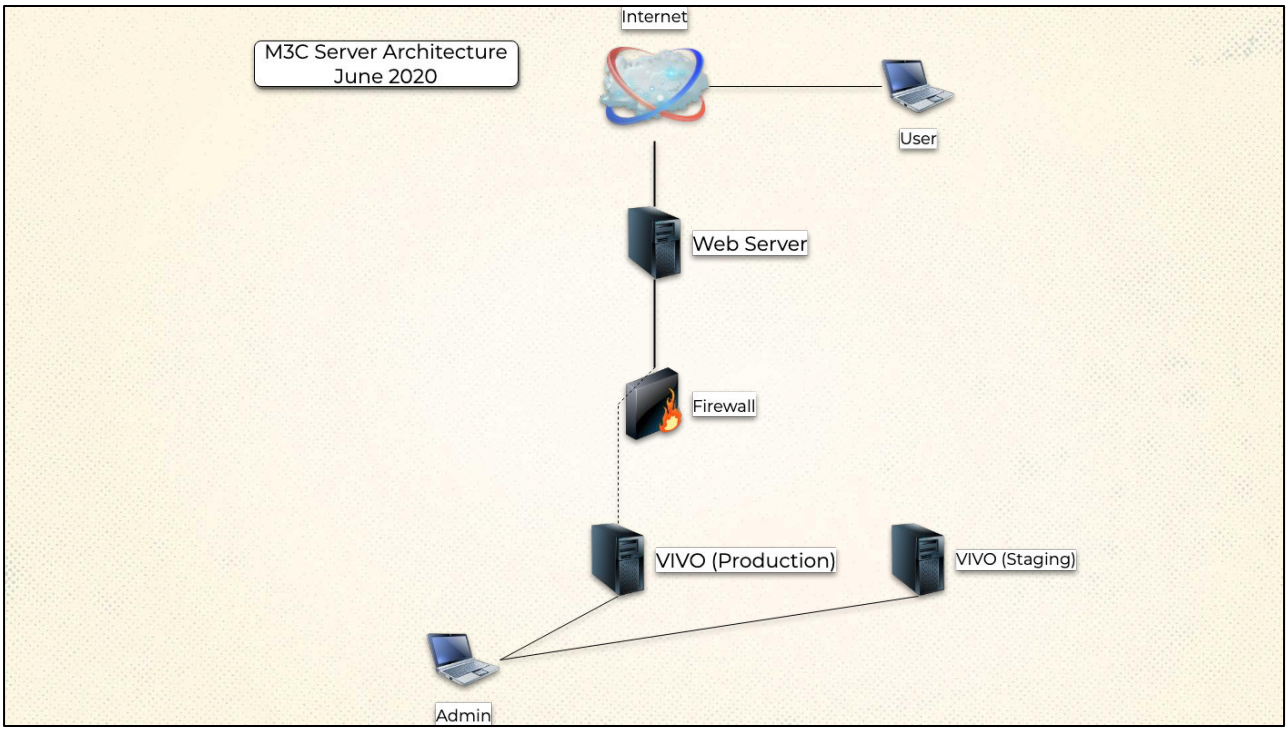
To recap, TPF allows us to operate at the Subject-Predicate-Object level of abstraction for our data.

That means we don't have to worry about how the data is actually stored (such as in MariaDB or TDB).

The downside is that we don't get the benefits of using a higher-level of abstraction, such as SPARQL's joins and optionals.

Joins are left up to the client to deal with.

OK, so I'm going to give an overview of our server architecture.



So you can get an idea of how everything fits together. Here the diagram.

The main takeaways is that we have a public-facing **Web Server** in front of a Firewall.

Behind that, there is both a Production and Staging instance of our VIVO along with our custom programs.

Web Server

Apache HTTP Server

ProxyPass /tpf

/photos

m3c-web

tpf.js

m3c-proxy

Our Web Server runs Apache configured with ProxyPass to pass TPF requests through to our Production VIVO server, which is safely tucked away behind that Firewall.

It hosts the m3c-web frontend along with a caching proxy we created that eliminates unnecessary calls to the Production TPF server.

m3c-web

Plain JavaScript, HTML, CSS

tpf.js - TPF JavaScript library

Caches requests by URI

m3c-web is our plain JavaScript, HTML, and CSS frontend.

The JavaScript TPF client we developed should work with any TPF endpoint as it doesn't have anything M3C-specific.

m3c-proxy

TypeScript

Server-side caching of TPF requests

This simply NodeJS program dramatically improves the end-user's experience in about 100 lines of TypeScript.

Repeated TPF requests are handled by it eliminating the need to query the actual TPF server.

VIVO (x2)

VIVO

- Apache HTTP Server
- MariaDB
- (TPF) Server.java
- Jena sdb{load,dump}

m3c-tools

- PostgreSQL
- Python Flask

The VIVO installation is fairly straightforward.

We are using MariaDB to store our 1.2 million triples. And, of course, we're using the embedded TPF server.

We wrote our own set of ingest and administrative tools in Python, which we also run on each server.

m3c-tools

Python 3

Imports data and publications

Generates triples

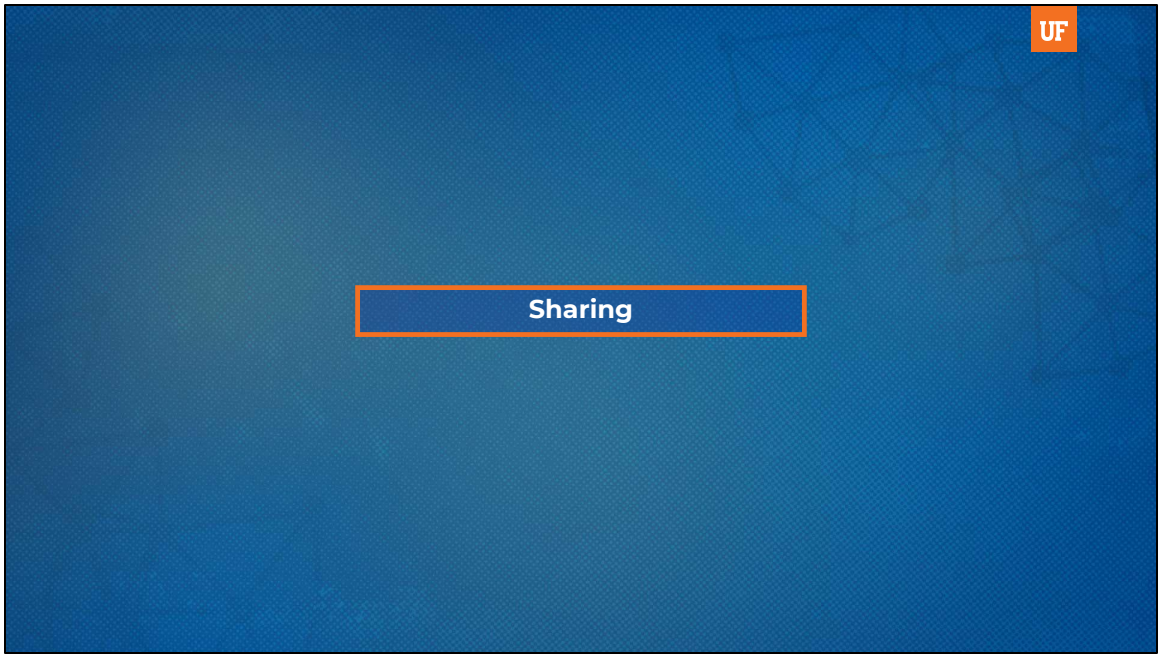
Serves administrative website

m3c-tools is our suite of Python programs.

It imports data and publications.

Generates triples for a monthly ingest.

It also serves our administrative website, where we can add, edit, and withhold data.



The last point I want to make is that we are big on sharing.

Open-source software

CC-0 licensed data

All of our code is open source under either MIT or Apache 2 licenses.

All of our data is licensed under the Creative Commons Zero (or Public Domain) license.

Links

People Portal	https://people.metabolomics.info/
TPF endpoint	https://people.metabolomics.info/tpf/core/
Data exports	https://people.metabolomics.info/data/
TPF client	https://github.com/ctsit/tpf/
m3c-web	https://github.com/ctsit/m3c-web/
m3c-tools	https://github.com/ctsit/m3c-tools/
m3c-proxy	https://github.com/ctsit/m3c-proxy/
Ontology	https://github.com/ctsit/metabolomics_ontology/

Here are all the links to our sites, software, and data.

Appendices

Finally

Acknowledgements

The M3C People Portal and metabolomics.info is a service of the Metabolomics Consortium Coordinating Center (M3C) (grant 1U2CDK119889-01) of the NIH Common Fund Metabolomics Program.

Our data is sourced from:

- The Metabolomics Workbench (https://www.metabolomicsworkbench.org/)
- PubMed (https://pubmed.gov)
- Metabolomics Tools Wiki (https://raspicer.github.io/MetabolomicsTools/)

Here are some important acknowledgements...

More Links!

TPF Specification
https://linkeddatafragments.org/specification/triple-pattern-fragments/

N-Triples Specification
https://www.w3.org/TR/n-triples/

W3C's Linked Data Description
https://www.w3.org/standards/semanticweb/data

and some more links that.

Thank you and God bless.