

# 2019 Architectural Fly-in

The VIVO architectural fly-in will be focused on bringing an architecturally-minded team together who individually represent distinct VIVO stakeholder constituencies for the purpose of developing architectural approaches required to address the direction of the project. The primary goal of the two-day face-to-face meeting will be to assess and document a plan for improving the VIVO application architecture towards enabling and realizing the technical efforts defined in the "[Statement on VIVO's Product Direction for 2019](#)".

The success of this effort relies on balancing the perspectives of:

- Research Intelligence
- Modern web interface
- Semantics and ontology
- Modern development tools and practices
- Performance / Scalability, and
- Existing VIVO architecture
  
- Logistics
  - When
  - Where
    - Parking
    - Recommended hotel
  - Who
- Agenda
- Requirements
  - Considerations
- Thoughts / Notes / Resources
  - Components to decouple
  - Interfaces for decoupled components
    - Core
    - Search Index
    - Triplestore
  - Mixed Components
  - Component packaging
  - Community contributions
  - Related discussions/presentations

## Logistics

### When

- January 29th - 30th, 2019
- Travel on 28th and 31st

### Where

Orlando - [UF Research and Academic Center - Lake Nona](#)

University of Florida Research and Academic Center  
Room 131  
6550 Sanger Rd.  
Orlando, FL 32827

### Parking

Please park in the area marked as visitor parking ([map](#))

Andrew will be renting a van for transport to/from hotel and meeting location



### Recommended hotel

[LaQuinta Airport North](#)

7160 N Frontage Rd, Orlando, FL 32812

Reasonable rates, AAA discount, newly renovated, very close to the hotel, upgraded wifi, breakfast included.

This is prime season to for Orlando (Dec-Mar) please book early to get a good rate.

## Who

- [Justin Littman](#)
- [Ralph O'Flinn](#)
- [Alex Viggio](#)
- [Benjamin Gross](#)
- [Brian Lowe](#)
- [Huda Khan](#)
- [Jim Blake](#)
- [Richard Outten](#)
- [Andrew Woods](#)

## Agenda

[2019-01-29 to 30 - VIVO Arch Fly-in f2f Meeting](#)

## Requirements

1. VIVO must support ingest rates of x-number of triples per hour
2. VIVO must support total scale of y-number of triples
3. VIVO must support the removal of triples, regardless of the number of triples or size of graph
4. VIVO must support response times of < 2sec for profiles, regardless of number of relationships
5. VIVO must service any frontend user interface via an API
6. VIVO must support API-driven integration tests
7. VIVO must support the swappability of the following components
  - a. Search index
  - b. Triplestore
  - c. Asset store
8. VIVO must support ingest without stopping other components (indexer)
9. VIVO must support configuration update without requiring application restart

## Considerations

1. VIVO should be designed towards being "cloud-ready"

## Thoughts / Notes / Resources

### Components to decouple

- Core
- Search index
  - Searching and indexing can be separate components
- Triplestore(s)
  - Reasoners - ABox and TBox
- UI
- Authentication
- Authorization

### Interfaces for decoupled components

#### Core

- What endpoints / services are required by UI?
- Starting point: Classes with @WebServlet annotations

#### Search Index

#### Triplestore

- Starting point: [RDFService.java](#)
- Can inferencing be split out? [TBoxReasoner.java](#)

## Mixed Components

- Not all components are easily compartmentalized, like APIs or TripleStores
- Consider Authentication, which includes:
  - back-end business logic,
  - data model storage,
  - UI elements
- If we wanted to support a new authentication scheme (say, two-factor authentication), we would need changes in all of these areas.

## Component packaging

- Back-end code can be delivered in a JAR file
- How is front-end code delivered?
- What about mixed components?

## Community contributions

Scenario:

- University X develops a new Triplestore component, called newTriple
- University Y wants to use newTriple
- One possibility is to wait for VIVO central to adopt newTriple into the core distribution. Then University Y can upgrade their system to the latest release, to get newTriple
- What are other possibilities?
- This leads back to packaging.

## Related discussions/presentations

- [VIVO conference presentation discussing community alignment](#)
- [Brainstorming document reviewing possibilities for API directions](#)