2019-01-10 - VIVO Arch Fly-in Planning Meeting

Date

10 Jan 2019

Call-in Information

Time: 12:00 pm, Eastern Time (New York, GMT-05:00)

To join the online meeting:

- Go to: https://duraspace.zoom.us/j/823948749
- Or iPhone one-tap :
  - US: +14086380968,,823948749# or +16468769923,,823948749#
- Or Telephone:
  - Dial(for higher quality, dial a number based on your current location):
  - US: +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833
  - Meeting ID: 823 948 749
- International numbers available: https://duraspace.zoom.us/zoomconference?m=Qy8de-kt6W4fMMDQCAV_3qfH1W-lxAo5

Attendees

blocked URL Indicating note-taker

1. Justin Littman
2. Ralph O'Flinn
3. Alex Viggio
4. Benjamin Gross
5. Brian Lowe
6. Huda Khan
7. Jim Blake
8. Richard Outten
9. Andrew Woods

Agenda

1. Logistics

    a. 2019 Architectural Fly-in Travel Plans
2. Defining the outputs of the fly-in
    a. Top-level architectural diagrams
    b. Component diagrams
    c. Inputs/Outputs for each component
    d. Draft HTTP API
3. What are the salient features that will drive the architecture?
4. What are the current bottlenecks, pain-points?
    a. Product Direction for 2019
5. Working documents
    a. Architectural diagrams
        i. Top-level and Detailed, component-level
        ii. APIs and services, specified
    b. Features spreadsheet
        i. Questions to answer:
            1. What will the product will do/support?
            2. What additional business value-add can the re-architecture offer?
            3. What are core? and why?
            4. Which features are in/out/optional?
            5. What audiences are specific features for?
            6. What are the logical groupings for features into modules?
            7. Will we offer multiple product distributions, analogous to VIVO/Vitro?
                a. Perhaps an "enterprise VIVO" vs. a "small-shop VIVO" vs. Vitro
6. Meeting schedule leading up to Jan 29th
    a. Jan 10th @noon ET
    b. Jan 22nd @noon ET
    c. Jan 24th @noon ET

Notes

Audio recording

- [2018-01-10-vivo-fly-in-planning.mp4](#)

Questions / Observations for Straw-chitectures

vivo_arch_v1.png

1. VIVO is separate from read-only sites
2. Triplestore is central to VIVO
3. Difference between Source Systems and staging/loaders?
4. External Solr gets loaded by a loader?

vivo_arch_v3.png

1. Scalable ingest
2. No focus on refactoring VIVO
3. How are loaders triggered?
4. How are updates managed, from the UI?
5. jb - Why keep "VIVO Triple Application" with its triple-store and inferencing? To provide a SPARQL endpoint? To provide an editing environment?

VIVO Product Evolution Straw

1. Similar to above
2. No focus on refactoring VIVO
3. jb – At first glance, it appears to be straightforward to implement GraphQL as a layer over SPARQL. This might indeed insulate developers from SPARQL complexity, but it is unlikely to improve performance.

Screen Shot 2018-12-20

1. Focus on VIVO refactoring
2. How is performance and scale addressed?
3. Is the frontend decoupled from the APIs?
   - Server-side or client-side frontends?
4. jb – Would the internal front-ends access the JSON APIs, or the existing controllers?
5. jb – For the internal front-ends, Would there be any attempt to separate display from editing? Currently they are very much intertwined.

Future VIVO Example

1. Focus on VIVO refactoring
2. Maybe config does not need to be in a triplestore
3. jb – Is the "New REST API" read/write or read-only?
4. jb – Are the Legacy Freemarker pages just for admin functions, or are they also for searching, profile display, and profile editing?
5. jb – Why the second search index? Is it more efficient, more flexible to populate two indexes?

VIVO 2.0 Arch

1. Focus on VIVO refactoring
2. Similar to above
3. jb – This appears to be a draft of "Future VIVO Example"

arhitectura-vivo

1. Addresses updates from UI and other sources
2. Can service layer become the core VIVO app w/ content sources externalized?
3. jb – What about policy filtering? (i.e., data is restricted from display and/or editing based on user accounts.) Currently, language filtering is applied at the SPARQL query level, but policy filtering is implemented higher in the stack. Where would policy filtering be implemented in this architecture? Or would it be dispensed with?

VIVO arch brainstorming

1. jb – What is the benefit of JSON-LD? Isn't it just another RDF notation?
2. jb – This might avoid one of the pitfalls of a graph model: the graph is one continuous structure, with no boundaries (ref: the deletion problem).

VIVO arch brainstorming-2

1. No focus on refactoring VIVO
2. Similar to Duke?
3. Loader model
4. What is the interation between webapps and content stores? API?
5. What does deployment look like?

6. Where does the VIVO ontology come into the picture?
7. jb – What is the virtue of having both a triple-store and an RDBMS? What are the costs?

VIVO_Architecture_ideas.png

1. jb – In the short run, the Data Distribution API is a good way to rapidly develop a back-end for an Angular/React/etc front-end.

ACTIONS

Collection of existing architecture diagrams / resources

- Diagrams
  - Cornell (Huda)
    - VIVO (in general)
      - From the wiki: diagram from architecture overview page
        - component diagram from main architecture page
      - Can include more brainstorming/high level component diagrams/discussions from presentations and documents as well but not sure this is the bullet for it
    - Scholars at Cornell
      - Presentation: https://figshare.com/articles/Scholars_Cornell_Visualizing_the_Scholarship_Data/5277958/1 (Slide 19 shows a high level overview of the components)
      - Data Distribution API structure:
        - image which shows API-related components (these would connect to VIVO)

Discussion

Previous Actions

- All to complete priority row
- All to review produce diagrams
- Collect diagrams Opera (Brian)
- Collect diagrams Rialto (Justin)
- **ACTION:** ALL to help enumerate list of current features "Feature audit" (Google-doc)

  - What to carry forward
  - What to leave out
- Andrew to Potentially poll the community