openEHR Conformance

Conformance for HiGHmed

* we needed to test openEHR conformance on the modifications to EtherCIS

* separation of test case spec from technical implementation

  - focused on SM not REST

  - specified detailed cases for each service, for normal and problematic cases

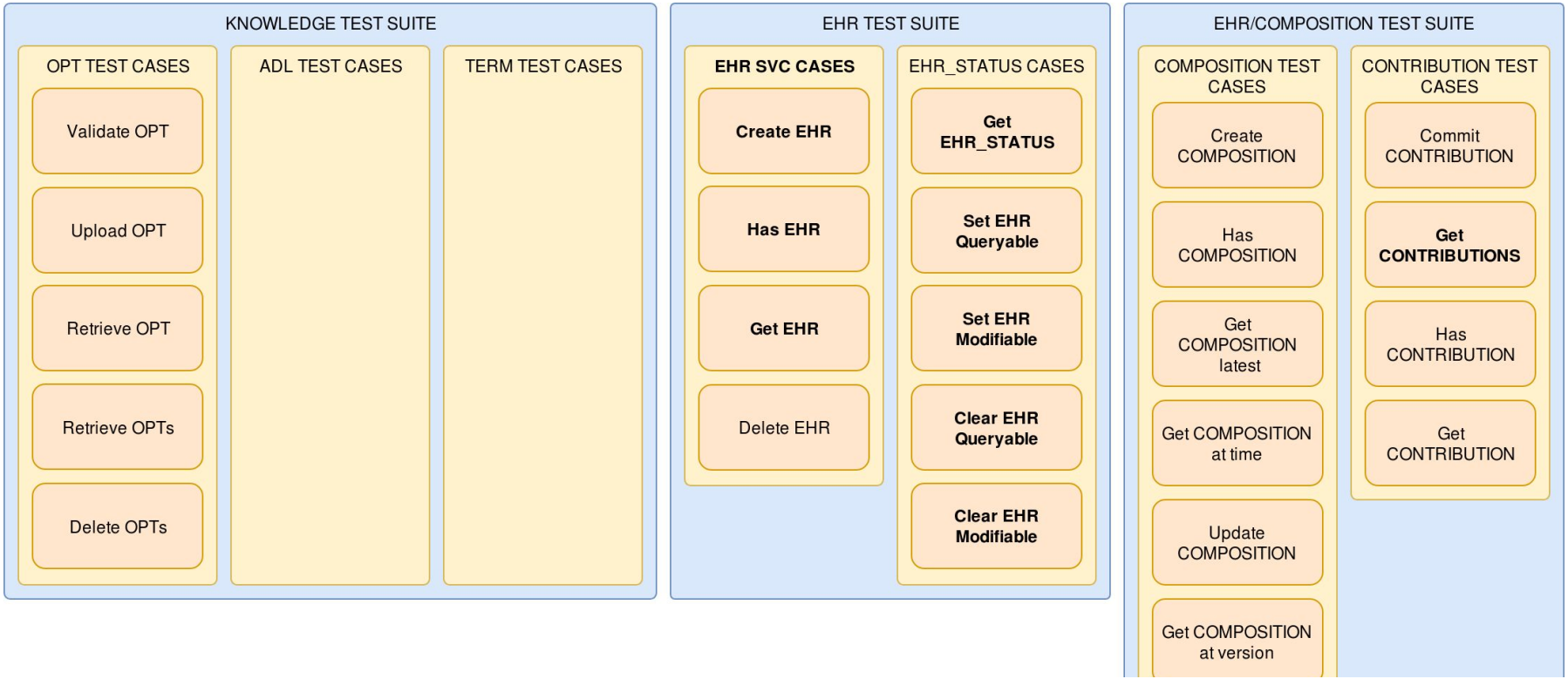
* test case spec considered current CNF and SM

* raised issues on both specs as we progress

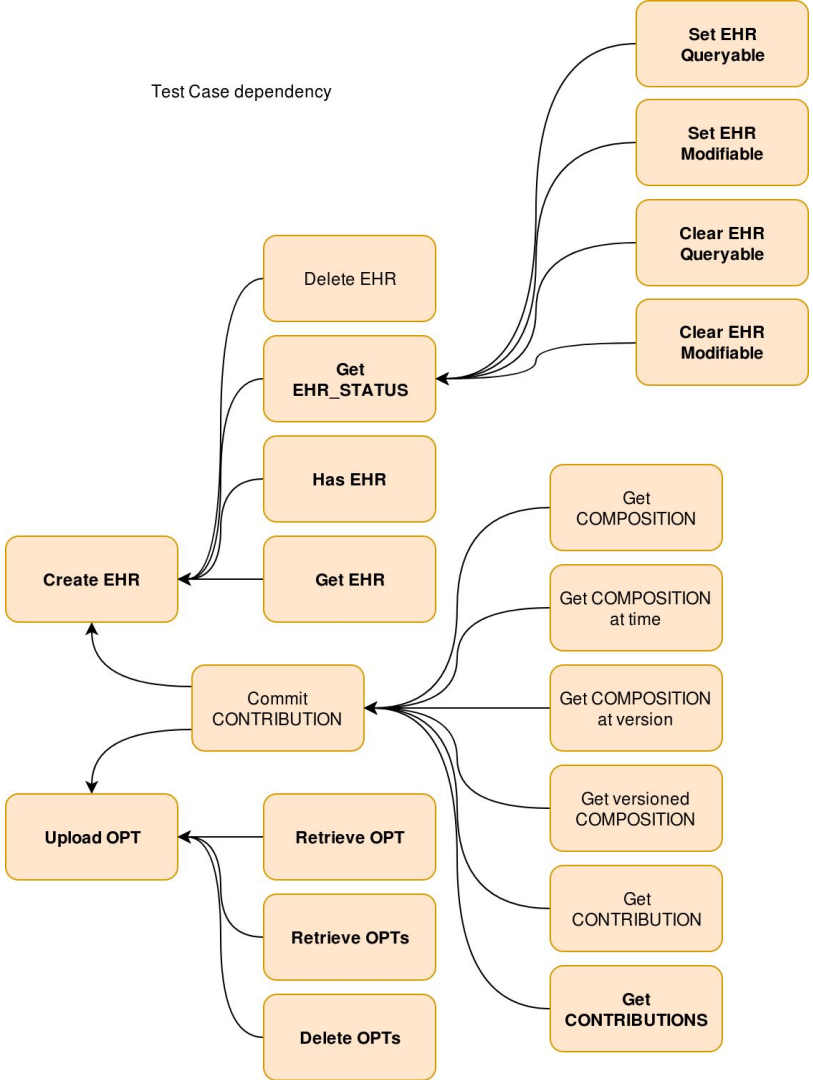
Conformance for HiGHmed (test case specs)

* one spec per SM component
  - EHR, COMPOSITION, CONTRIBUTION, FOLDER, DEFINITION, QUERY
  - TERMINOLOGY is still missing from the SM spec
* tried to define standard steps for each case
  - pre conditions, setup, actions, post conditions, and clean

Conformance for HiGHmed (analysis)



Conformance for HiGHmed (analysis)



Test cases for OPT (a simple one)

https://docs.google.com/document/d/1wdklJROuC2UQQK6x1cjPkF1I9Na2spfEIBjRbNVK-Q0/edit?usp=sharing

* Need to test different data sets (OPTs) and is difficult to find a minimal representative set, since OPTs can have any level of complexity, chosed:

- minimal OPTs (COMPO+1ENTRY+1DV)
- all datatypes (COMPO+ALL ENTRIES+ALL DVs)
- valid and invalid OPTs (syntactically)

Test cases for QUERY (a complex one)

https://docs.google.com/document/d/13TuxEX1T0ZBlguLBfMkulP-3iFUFFyBejbM2aUTxb1M/edit?usp=sharing

* Testing queries mean testing:

  - the REST API endpoints

  - the query syntax (there are millions of variations!)

  - the execution logic

  - the result sets (depend on the loaded data!)

  - is like testing a programming language and a compiler ¯_(ツ)_/¯

Test cases for QUERY (a complex one)

For testing the syntax, created a meaningful set with alternatives:

- SELECT e/ehr_id/value as uid FROM EHR e CONTAINS COMPOSITION c WHERE c/archetype_node_id **matches** {'openEHR-EHR-COMPOSITION.minimal.v1'}

- SELECT e/ehr_id/value as uid FROM EHR e CONTAINS COMPOSITION c WHERE c/archetype_node_id**=**'openEHR-EHR-COMPOSITION.minimal.v1'

- SELECT e/ehr_id/value as uid FROM EHR e CONTAINS COMPOSITION c **[openEHR-EHR-COMPOSITION.minimal.v1]**

- got 76 different alternatives until now for 24 test cases

Implementation

* Switched from Cucumber+Code to Robot framework (python, has Java integration)

* Robot is declarative and oriented to keywords https://robotframework.org/

* Really nice for simple tests, can get messy for complex tests and a lot of data sets and parameter alternatives, requires a lot of work to make tests readable, test execution reports are great!

* Personal preference, I would go with Spock (Groovy DSL for testing) http://spockframework.org/

Implementation in Robot

```
*** Test Cases ***

Main flow: successfully commit CONTRIBUTION with single valid VERSION<COMPOSITION>

   upload OPT    minimal/minimal_evaluation.opt

   create EHR

   commit CONTRIBUTION (JSON)  minimal/minimal_evaluation.contribution.json

upload OPT file

   ${resp}=            Post Request         ${SUT}    /definition/template/adl1.4
                       ...                  data=${file}    headers=${headers}
                       Set Test Variable    ${response}    ${resp}
                       # Log To Console      ${resp.content}
```

Open issues

* LIMIT/OFFSET for QUERY

* FOLDER operations for SM

  - e.g. discussed submitting the whole directory for each change vs. partial changes

  - also the ability to use the FOLDER uid for operations instead of the path param

* I_TERMINOLOGY is not yet defined in the SM

* are delete OPT / delete archetype going to be supported?

Open Source

Code and documents will be available soon!

With Apache 2 license