FHIR Python Client

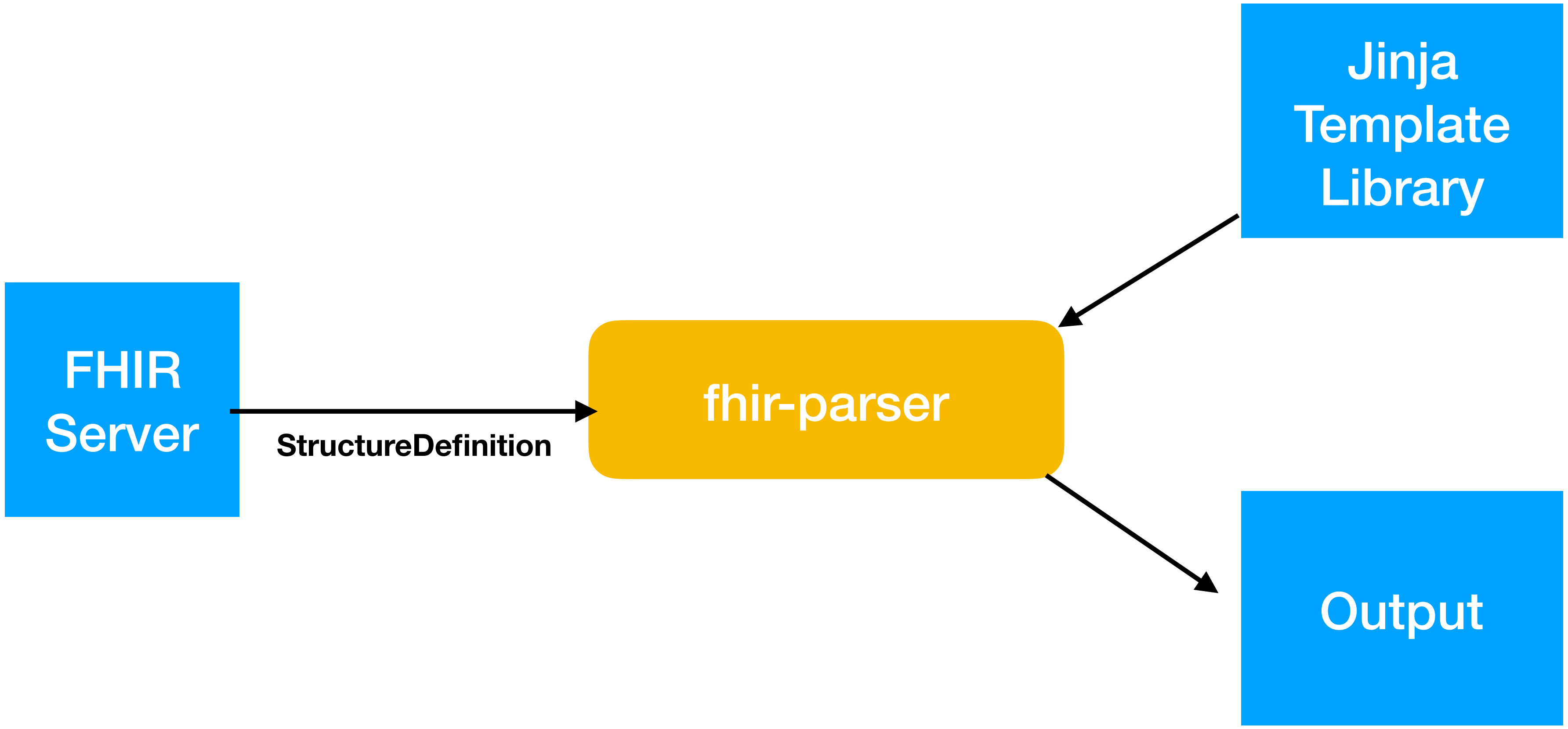
Background

Short meeting in Montreal

- General question: where is (are) the current FHIR Python implementations?
- Conclusion - the repository that appeared to be most current / complete was https://github.com/smart-on-fhir/fhir-parser/ and https://github.com/smart-on-fhir/client-py/
- Agreed to focus on this as **a** community supported implementation. (Others are welcome as well — this is just a place to start)
- Agreed to add it to FHIR wiki and other sites, improve documentation and enhance / improve

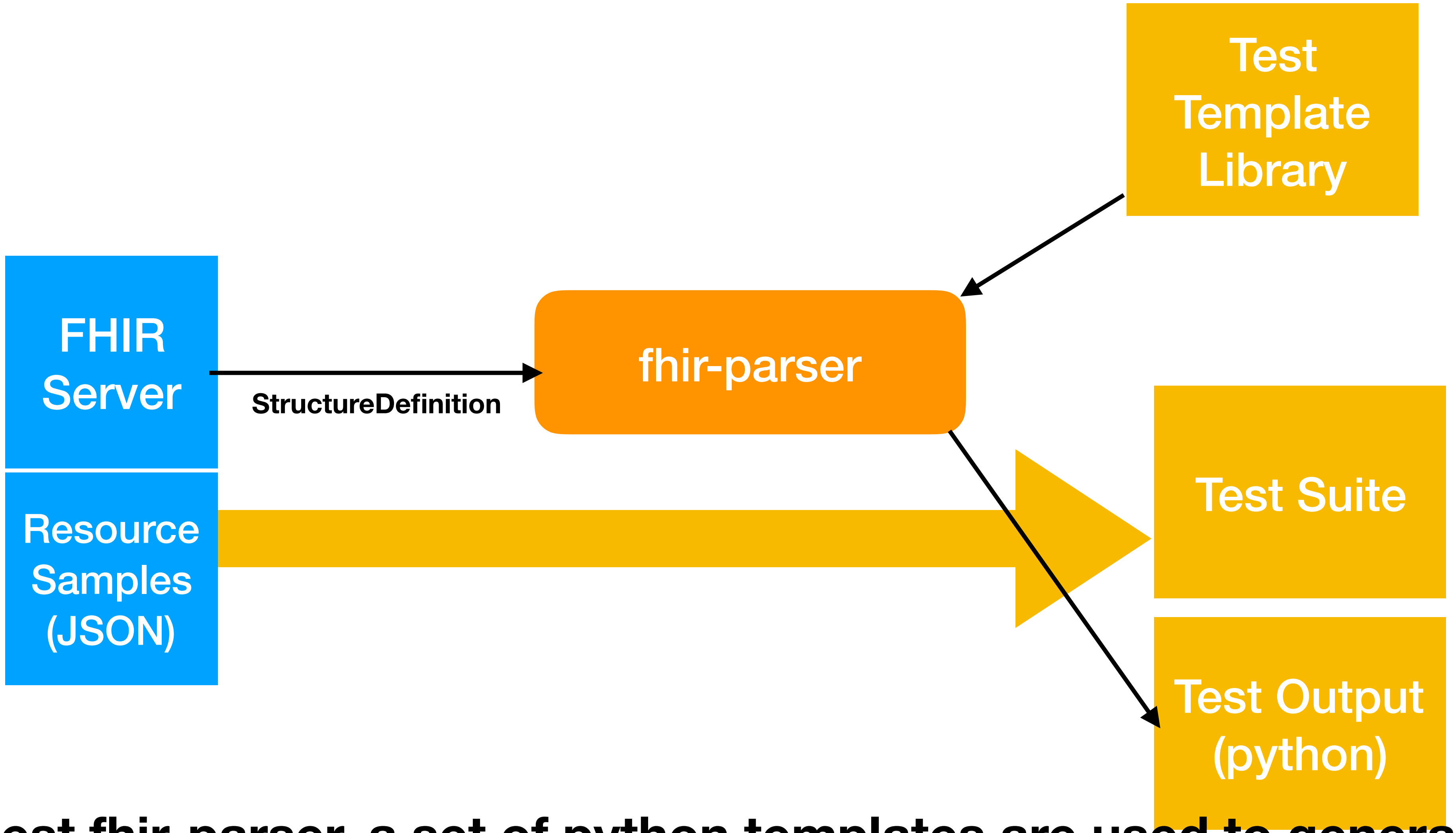
fhir-parser
vs.
client-py

fhir-parser



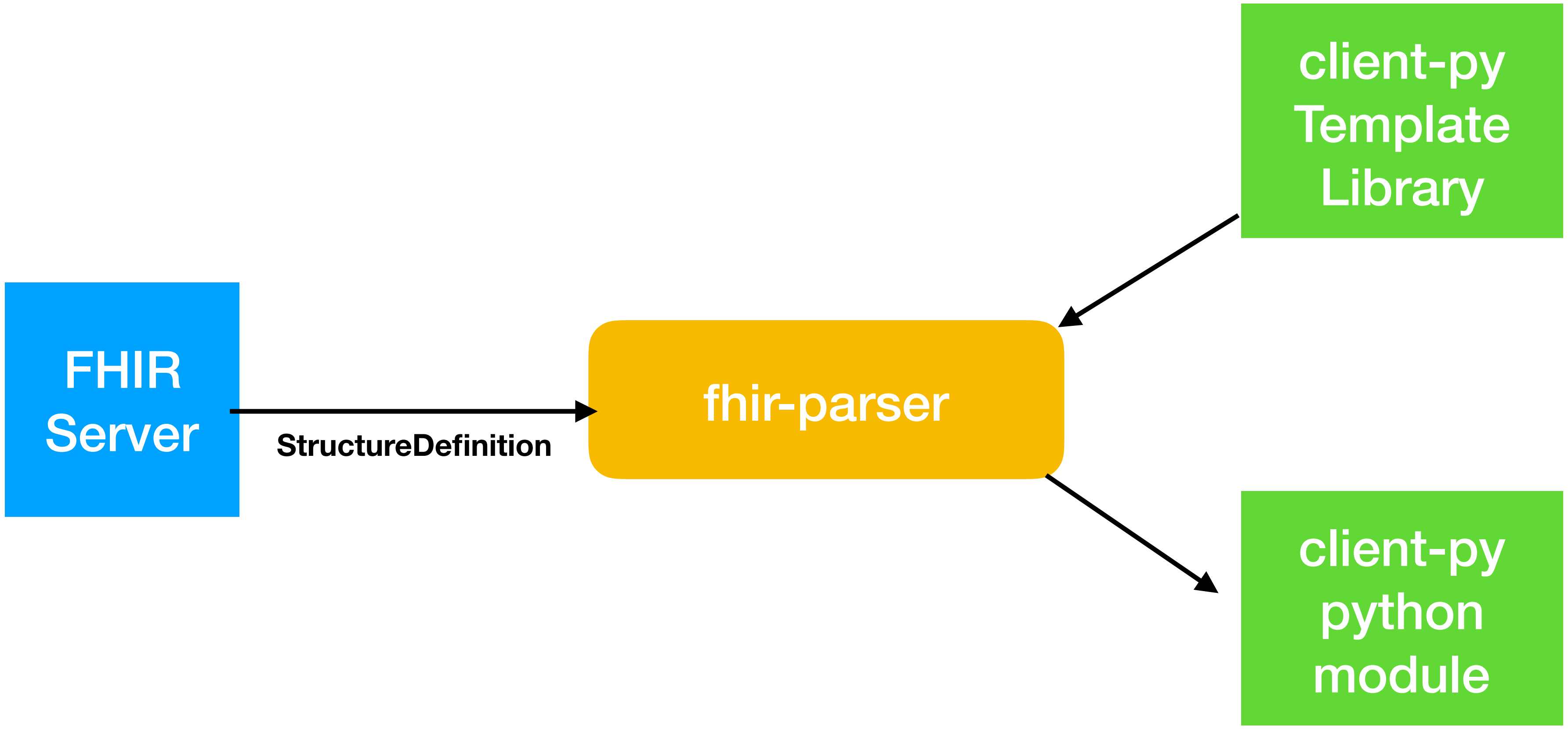
The purpose of fhir-parser is to feed a set (release) of FHIR resource definitions into a set of Jinja templates to produce output

fhir-parser
test harness



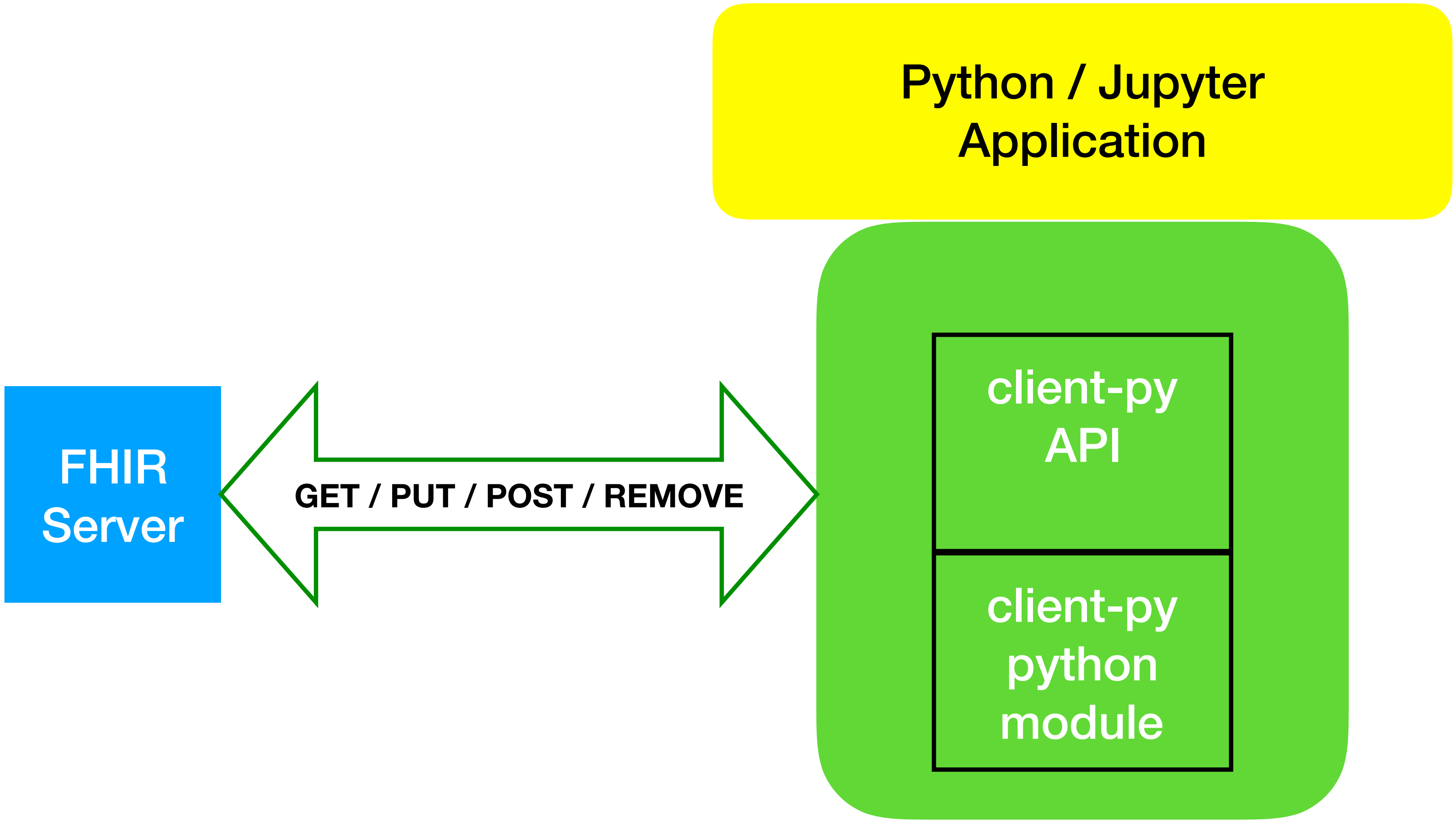
To test fhir-parser, a set of python templates are used to generate python loaders. A large collection of resource samples are then loaded using the test output and and the result is compared as the input, the thesis being that the parser is working correctly if all samples can be loselessly represented in the parser output.

client-py



client-py uses fhir-parser to transform a particular release of a FHIR server (STU3, R4, …) into a corresponding python library…

client-py



… the library is then used by the client-python API’s to create an access package that is used to read and write instance

Proposal for client-py enhancements (in no particular order)

1. Write permission or alternatives
2. Primitive data types — constraints, size, value ranges
3. Type hints using dataclass library
4. Shortcuts
5. pypi (pip and pipenv)
6. client-py.github.io documentation (if not already there)
7. Juypter notebook documentation and examples and binder
8. Split SMART from client-py?
9. Extensions and profiles?

Write Permission or Alternatives

- Current web site is owned by SmartOnFHIR community

- May want to consider separating SMART specific components from client-py no matter what

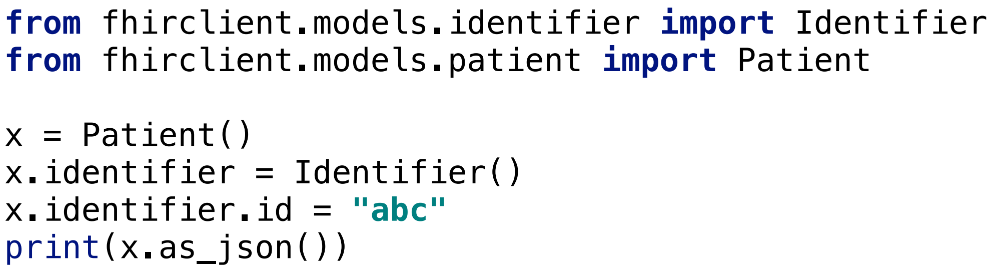
- **Todo**: Pascal will check with the SmartOnFHIR group to see whether they want to continue to maintain primary ownership. If they do, will ask for broader commit permissions. If not, we will consider an alternate organization (HAPI? Firely? HL7?)

Primitive Types

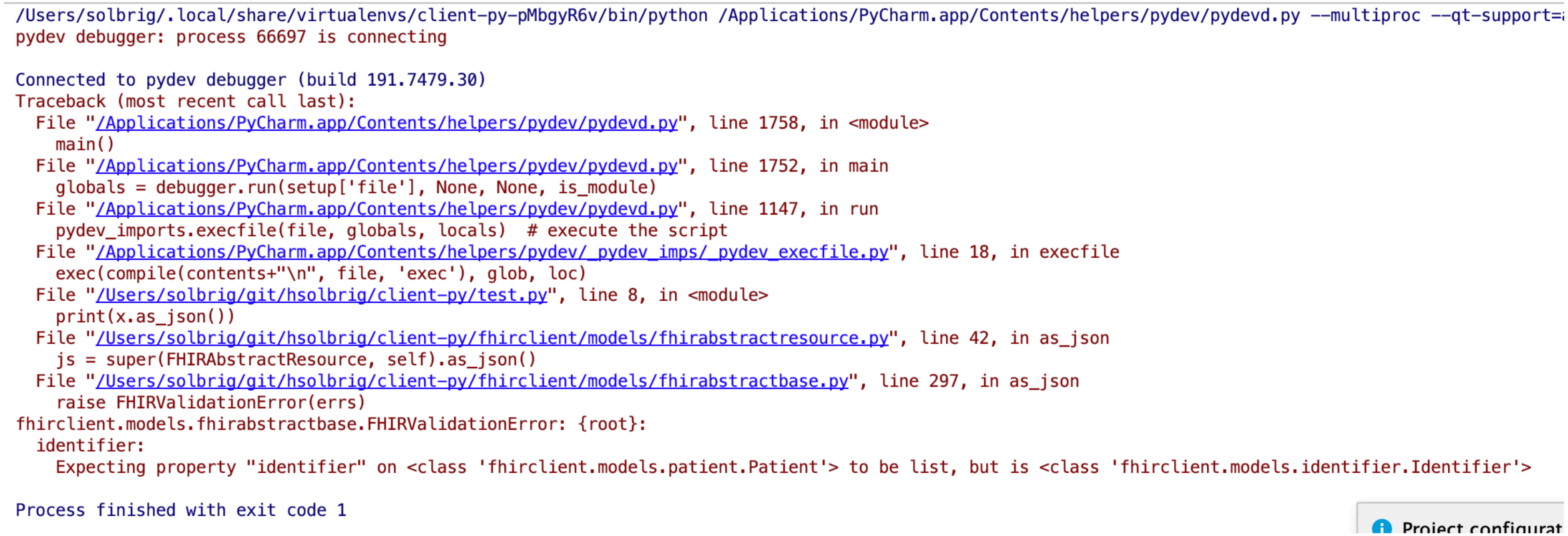
- See: datatypeextensions project for additional links and documentation
- String, Boolean, Integer, Id, …
- Transparent but extensible

```
x = Integer(17)
class SomeResource(…):
    def __init__(self, …):
            self.count: Integer = None

r = SomeResource(count=17)
r.count = 42         # == r.count.value = 42
```

Type Hints
Dataclass Library

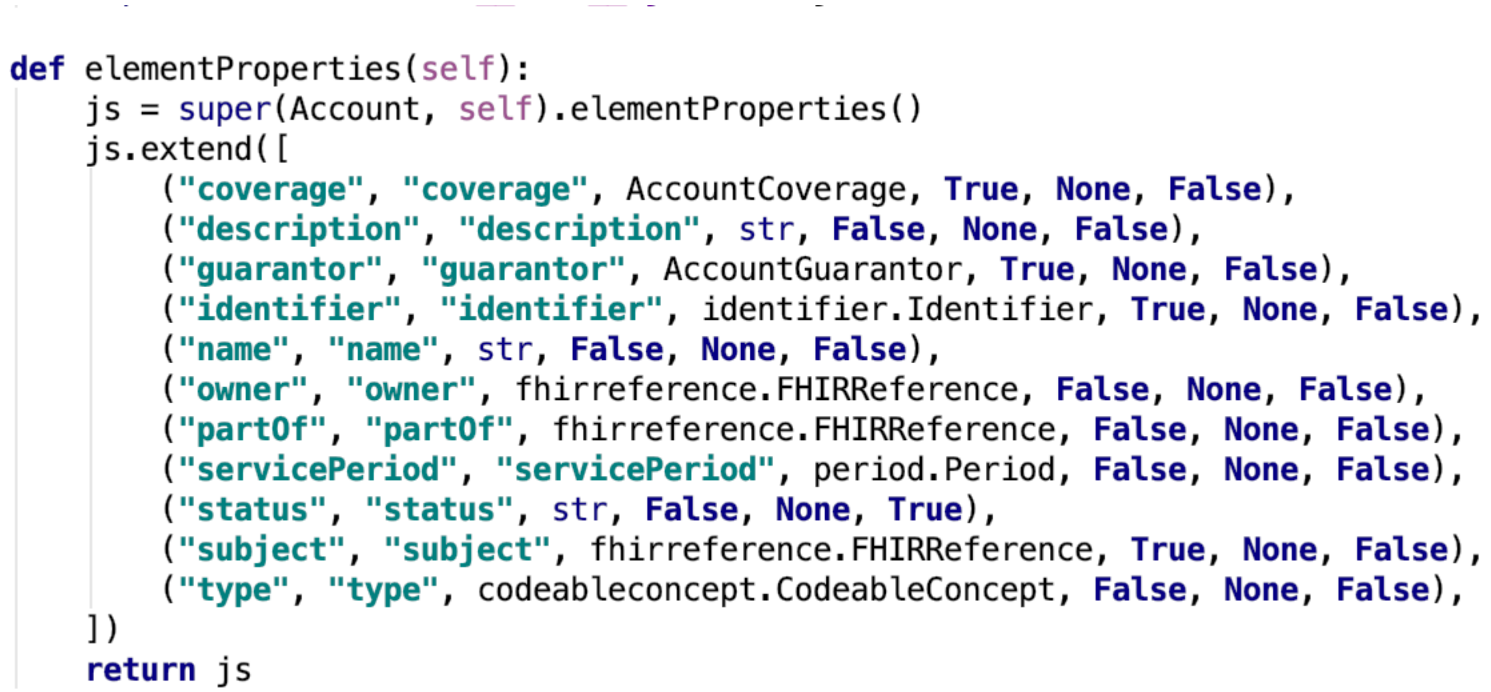


```python
from fhirclient.models.identifier import Identifier
from fhirclient.models.patient import Patient

x = Patient()
x.identifier = Identifier()
x.identifier.id = "abc"
print(x.as_json())
```



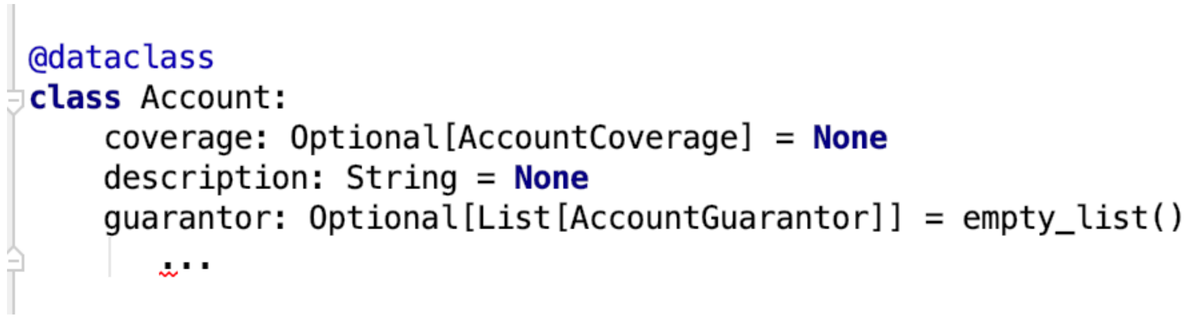
```
/Users/solbrig/.local/share/virtualenvs/client-py-pMbgyR6v/bin/python /Applications/PyCharm.app/Contents/helpers/pydev/pydevd.py --multiproc --qt-support=
pydev debugger: process 66697 is connecting

Connected to pydev debugger (build 191.7479.30)
Traceback (most recent call last):
  File "/Applications/PyCharm.app/Contents/helpers/pydev/pydevd.py", line 1758, in <module>
    main()
  File "/Applications/PyCharm.app/Contents/helpers/pydev/pydevd.py", line 1752, in main
    globals = debugger.run(setup['file'], None, None, is_module)
  File "/Applications/PyCharm.app/Contents/helpers/pydev/pydevd.py", line 1147, in run
    pydev_imports.execfile(file, globals, locals)  # execute the script
  File "/Applications/PyCharm.app/Contents/helpers/pydev/_pydev_imps/_pydev_execfile.py", line 18, in execfile
    exec(compile(contents+"\n", file, 'exec'), glob, loc)
  File "/Users/solbrig/git/hsolbrig/client-py/test.py", line 8, in <module>
    print(x.as_json())
  File "/Users/solbrig/git/hsolbrig/client-py/fhirclient/models/fhirabstractresource.py", line 42, in as_json
    js = super(FHIRAbstractResource, self).as_json()
  File "/Users/solbrig/git/hsolbrig/client-py/fhirclient/models/fhirabstractbase.py", line 297, in as_json
    raise FHIRValidationError(errs)
fhirclient.models.fhirabstractbase.FHIRValidationError: {root}:
  identifier:
    Expecting property "identifier" on <class 'fhirclient.models.patient.Patient'> to be list, but is <class 'fhirclient.models.identifier.Identifier'>

Process finished with exit code 1
```

Type Hints
Dataclass Library



```
def elementProperties(self):
    js = super(Account, self).elementProperties()
    js.extend([
        ("coverage", "coverage", AccountCoverage, True, None, False),
        ("description", "description", str, False, None, False),
        ("guarantor", "guarantor", AccountGuarantor, True, None, False),
        ("identifier", "identifier", identifier.Identifier, True, None, False),
        ("name", "name", str, False, None, False),
        ("owner", "owner", fhirreference.FHIRReference, False, None, False),
        ("partOf", "partOf", fhirreference.FHIRReference, False, None, False),
        ("servicePeriod", "servicePeriod", period.Period, False, None, False),
        ("status", "status", str, False, None, True),
        ("subject", "subject", fhirreference.FHIRReference, True, None, False),
        ("type", "type", codeableconcept.CodeableConcept, False, None, False),
    ])
    return js
```

Good idea…



```
@dataclass
class Account:
    coverage: Optional[AccountCoverage] = None
    description: String = None
    guarantor: Optional[List[AccountGuarantor]] = empty_list()
        ...
```

… and, core part of Python 3.6 on!

Shortcuts

- Make common idioms easy
- Instead of:

```
id1 = Identifier()
id1.value = “17”
p = Patient()
p.identifier = [id1]
```

- have

```
p = Patient()
p.identifier = “17”
```

or

```
p = Patient(identifier=“17”)
```

pypi

- setup.py / setup.cfg + requirements.txt
- Pipfile and Pipfile.lock   (pipenv)
- **Todo**: Set up for dual load (pip and pipenv)
- **Todo**: Determine on pipy account to use

Documentation

- https://docs.smarthealthit.org/clients/python/
- A good start but needs a lot of work
- **Todo**: Will decide where to go once repository and organization are settled see: Write Permission slide

Jupyter Notebook Documentation

- Some good examples have been posted on Zulip
- Also targeting AMIA Tutorial on Machine Learning w/ FHIR

Split Smart from client?

- Currently there is code for both
- Something to consider

Extensions and Profiles

- Future work to include:
  - Extensions as first class attributes
  - Constraints
  - Slicing
  - …