Unlocking Healthcare Data with Open Standards, Open Source Tools, and DuckDB

Dan Gottlieb

Principal, Central Square Solutions, LLC
Senior Technical Advisor, CHIP Program, Boston Children's Hospital

Dan@CentralSquareSolutions.com

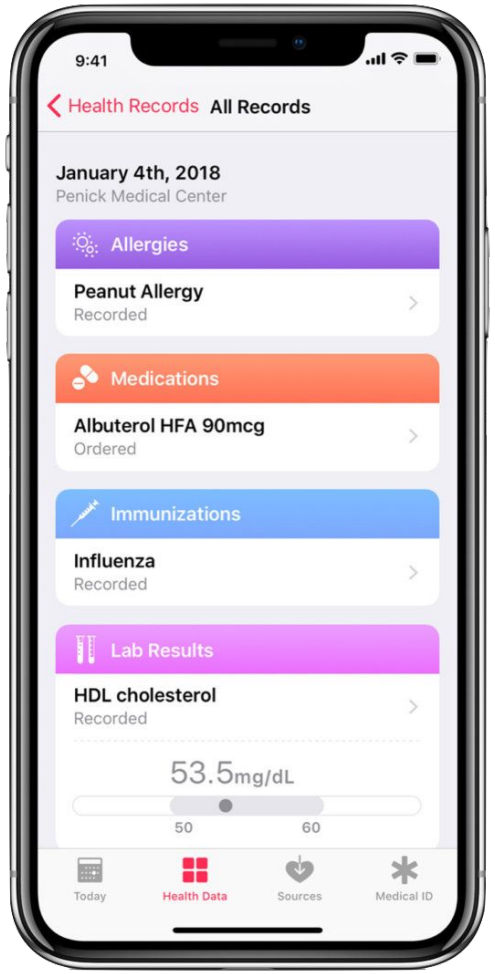
Christine is a 7 year old with Crohn's disease.

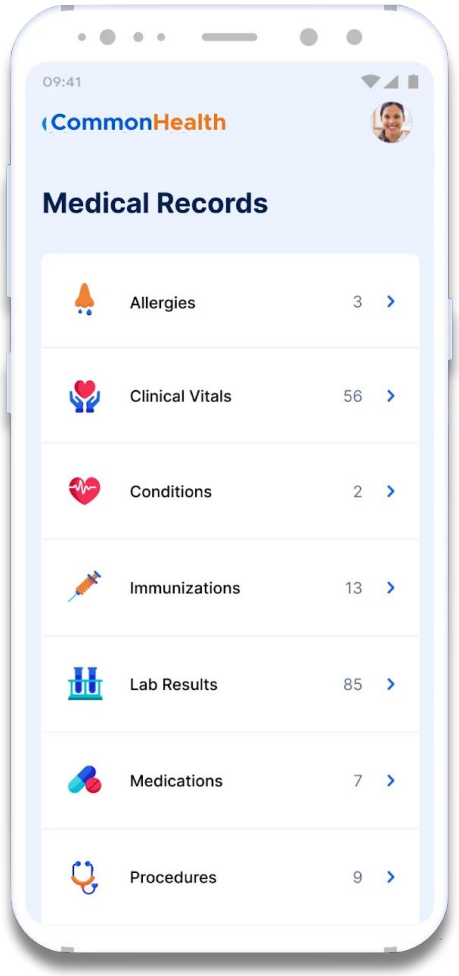
She is no longer doing well on infliximab (the first line treatment).

Which treatment should she try next?

FHIR (2012, but really 2018)

```
{
  "resourceType": "Observation", "id": "calprotectin-001",
  "status": "final",
  "code": {
    "coding": [{ "system": "http://loinc.org",
                 "code": "38445-3",
                 "display": "Calprotectin [Mass/mass]" }] },
  "subject": { "reference": "Patient/12345" },
  "effectiveDateTime": "2026-08-17T08:30:00Z",
  "valueQuantity": { "value": 285, "unit": "ug/g",
                     "system": "http://unitsofmeasure.org", "code": "ug/g" }
}
```


9:41
Health Records
All Records
January 4th, 2018
Penick Medical Center
Allergies
Peanut Allergy
Recorded
Medications
Albuterol HFA 90mcg
Ordered
Immunizations
Influenza
Recorded
Lab Results
HDL cholesterol
Recorded
53.5mg/dL
50
60
Today
Health Data
Sources
Medical ID


09:41
CommonHealth
Medical Records
Allergies 3
Clinical Vitals 56
Conditions 2
Immunizations 13
Lab Results 85
Medications 7
Procedures 9

Bulk FHIR (2018, but really 2023)


```
{"resourceType":"Observation","code":{...},"valueQuantity":{"value":157.4,...}}
{"resourceType":"Observation","code":{...},"valueQuantity":{"value":158.1,...}}
```

Many Bulk FHIR implementations!

Open Source FHIR Servers

- Microsoft
- HAPI
- Medplum

Commercial FHIR Servers

- Azure API for FHIR
- CareEvolution
- Firely Server (SQL endpoint)

EHRs

- 269+ ONC Certified Implementations
- Including: Epic, Cerner and Meditech

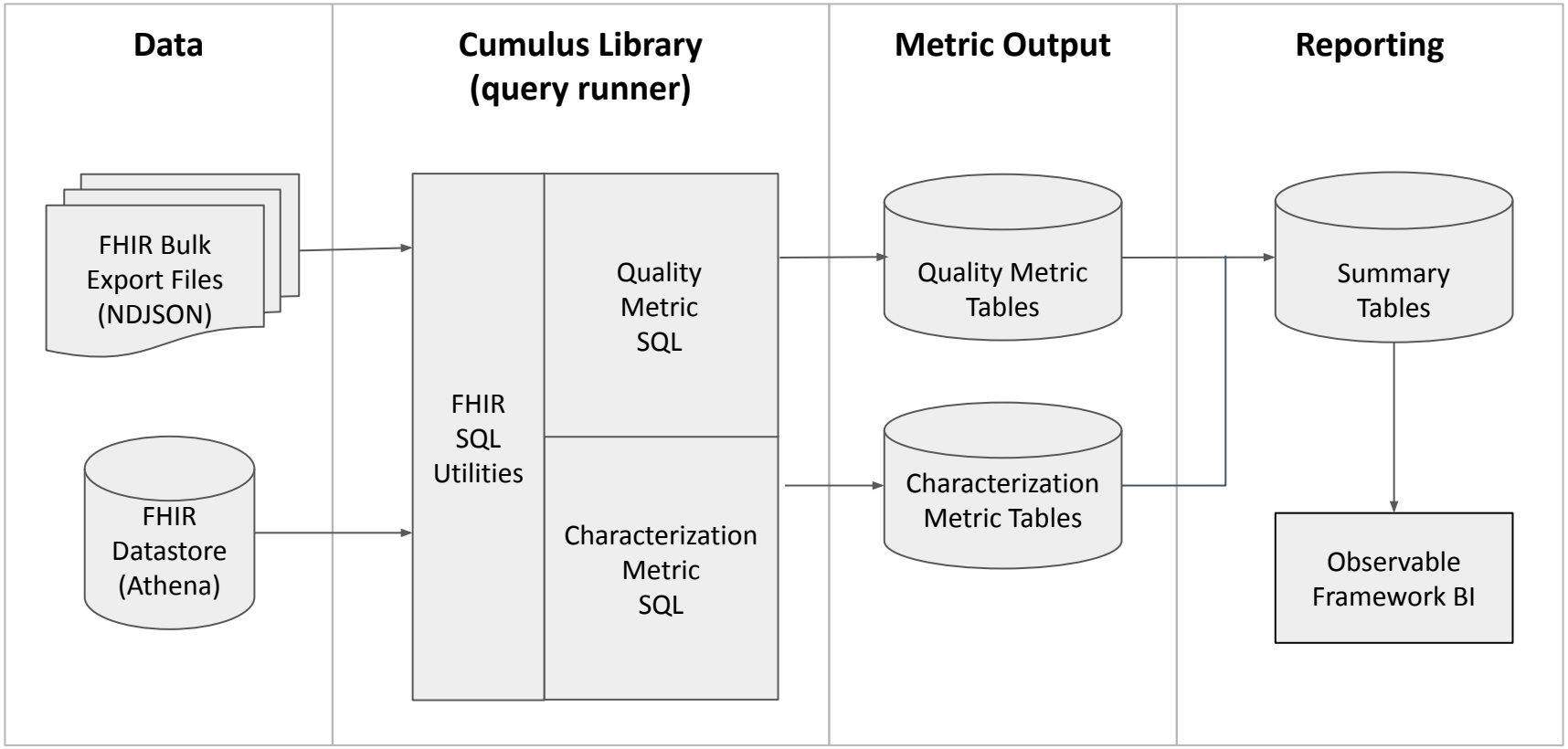
Payor Data Servers

- CMS ACO Beneficiary Claims Data
- CMS Claims to Part D Sponsors
- CMS Data at the Point of Care


Medical Record
System
Data Silo
FHIR
Bulk Data
Research DB

CumulusQ

https://smarthealthit.org/cumulusq/



Quality and Characterization Examples

(full set at https://github.com/sync-for-science/qualifier/blob/master/metrics.md)

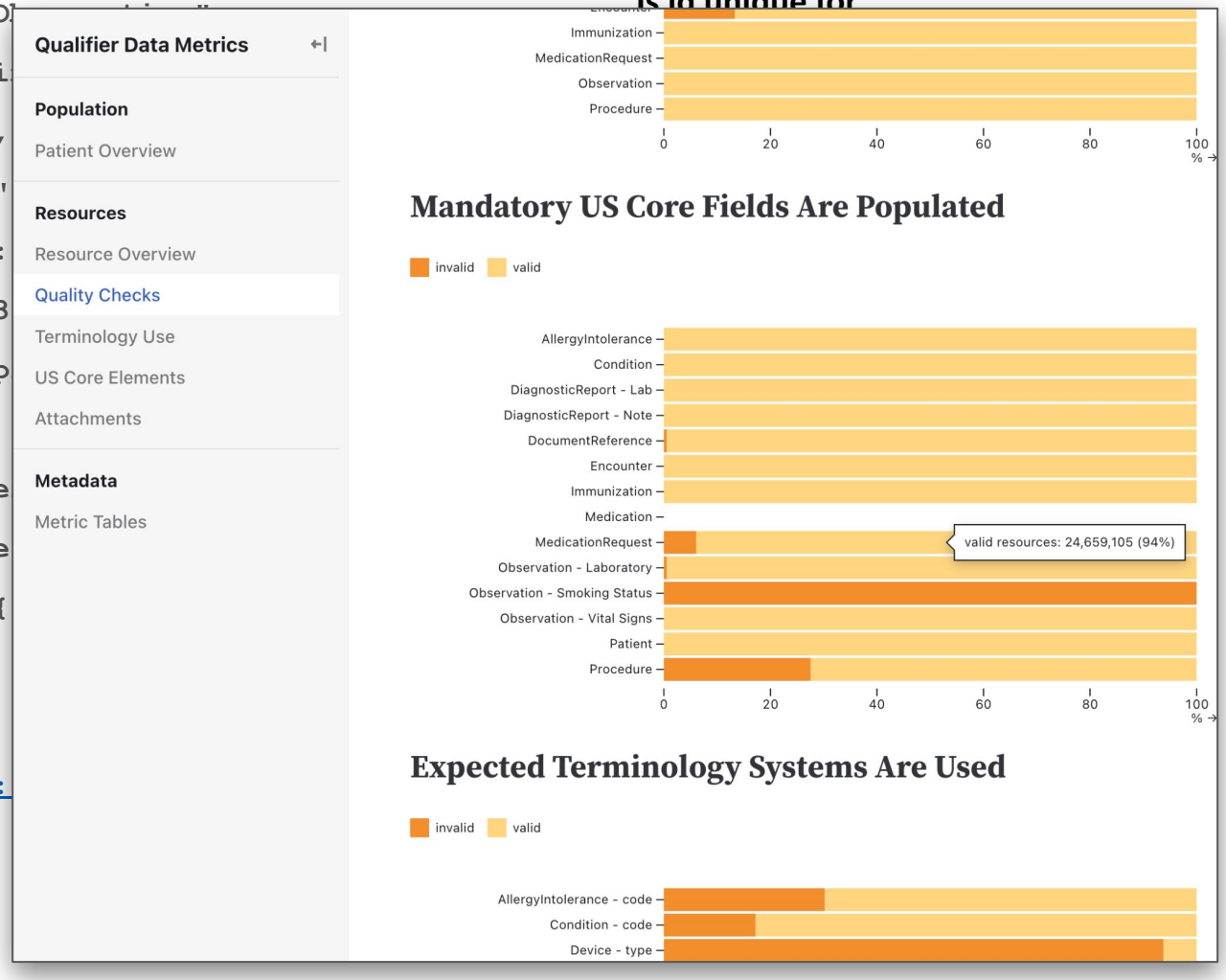
How many lab results are in this dataset
c_resource_count

Can I rely on a this terminology being used for all labs?
c_system_use

Is date in the future or distant past?
q_date_recent

Is value realistic?
q_obs_value_range

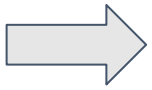
```
{
"resourceType": "O
"id": "calprotecti
"status": "final",
"code": { "coding"
  "system": "http:
  "code": "38445-3
  "display": "Calp
 }]},
"subject": { "refe
"effectiveDateTime
"valueQuantity": {
  "value": 285,
  "unit": "ug/g",
  "system": "http:
  "code": "ug/g"
}
}
```




Medical Record
System
Data Silo
FHIR
Bulk Data
Research DB

What do we need?


```
{
  "resourceType": "Observation",
  "id": "calprotectin-001",
  "status": "final",
  "code": { "coding": [{
    "system": "http://loinc.org",
    "code": "38445-3",
    "display": "Calprotectin [Mass/mass]"
   }]},
  "subject": { "reference": "Patient/12345" },
  "effectiveDateTime": "2026-08-17T08:30:00Z",
  "valueQuantity": {
    "value": 285,
    "unit": "ug/g",
    "system": "http://unitsofmeasure.org",
    "code": "ug/g"
  }
}
```

patient_id	effective_date_time	value
12345	2026-18-17…	285

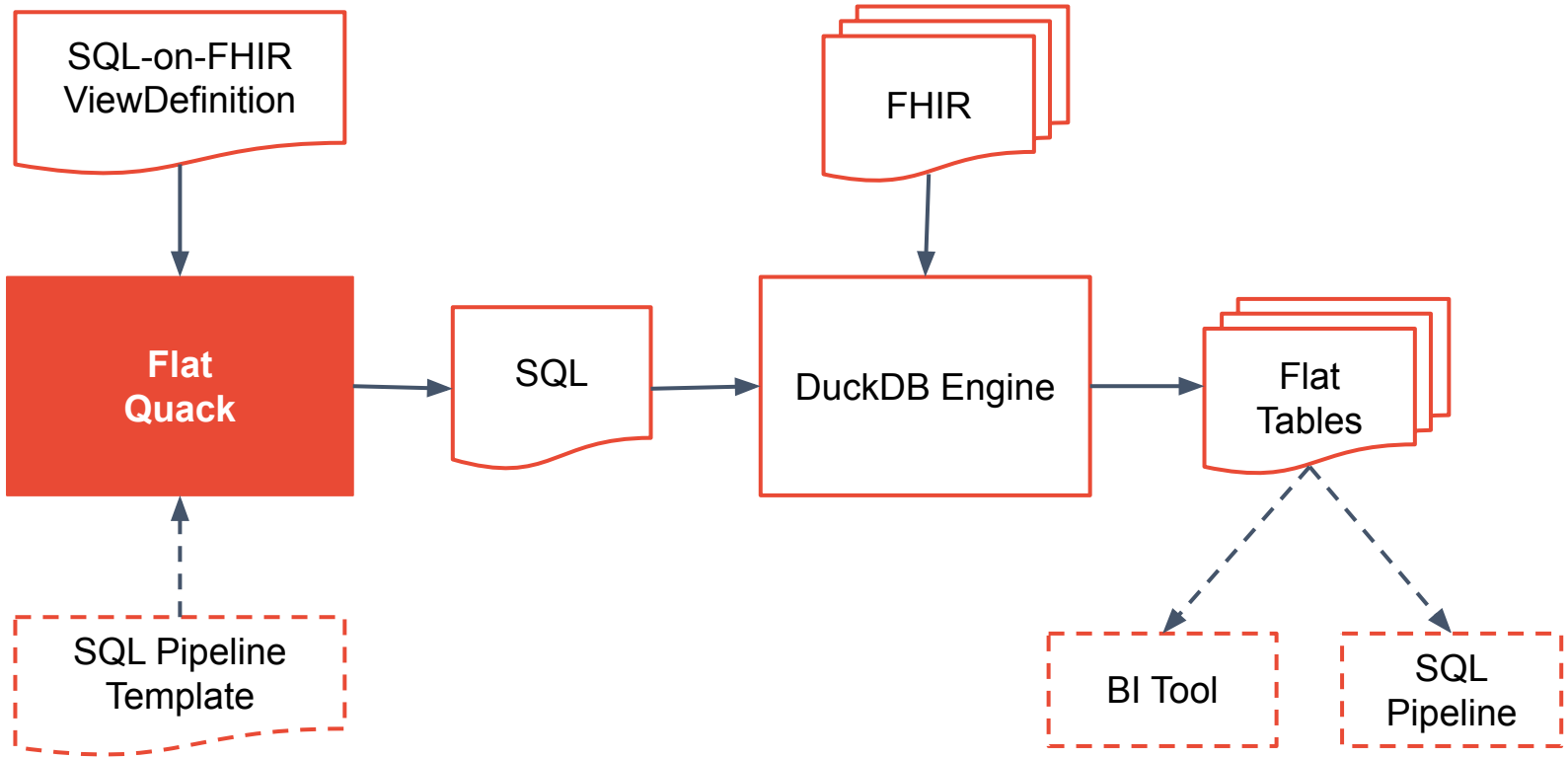
Calprotectin ViewDefinition

```
{
    "resource": "Observation",
    "name": "calprotectin",
    "select": [{
        "column": [{
            "name": "patient_id",
            "path": "subject.getReferenceKey()"
        },{
            "name": "effective_date_time",
            "path": "effectiveDateTime"
        },{
            "name": "value",
            "path": "value.ofType(Quantity).value"
        }]
    }],
    "where": [{
        "path": "code.coding.where(system='http://loinc.org' and code='38445-3').exists()"

    },{
        "path": "status = 'final' or status = 'corrected'"
    }]
}
```

FlatQuack.org

Fully open source tool to compile SQL-on-FHIR ViewDefinitions to DuckDB SQL



Calprotectin SQL

```
bunx flatquack -p calprotectin.vd.json -t @csv -m preview
```

```
SELECT {
    'patient_id': (subject.reference).parse_path ('/') [-1],
    'effective_date_time': effectiveDateTime,
    'value': (valueQuantity.value)
  } AS result
FROM
  read_json_auto(
    'Observation*.ndjson',
    columns = { subject: 'STRUCT(reference VARCHAR)', effectiveDateTime: 'VARCHAR',
                status: 'VARCHAR', valueQuantity: 'STRUCT(value DOUBLE)',
                code: 'STRUCT(coding STRUCT(system VARCHAR, code VARCHAR)[])'
              }
  )
WHERE
  (code.coding).list_filter (
      el -> ((el.system) = 'http://loinc.org')  and ((el.code) = '38445-3')
  ).list_filter(el -> el IS NOT NULL).ifnull2([]).len() > 0
```

What we learned

DuckDB macros are awesome!

- Pave over differences between DBs (e.g., Athena dateDiff vs. DuckDB date_sub)
- Fix persistent DB bugs (e.g., ifnull chaining)
- Add functionality (e.g., FHIRpath's nulls as empty list)

Just-in-time schema generation pattern is powerful

Modeling dates as ranges can avoid a lot of edge cases

Summary tables for visualization, detail tables for root cause analysis

Open Source FHIR Bulk Data Tools

[Sample FHIR Bulk Data] / [Synthetic Data Generator]

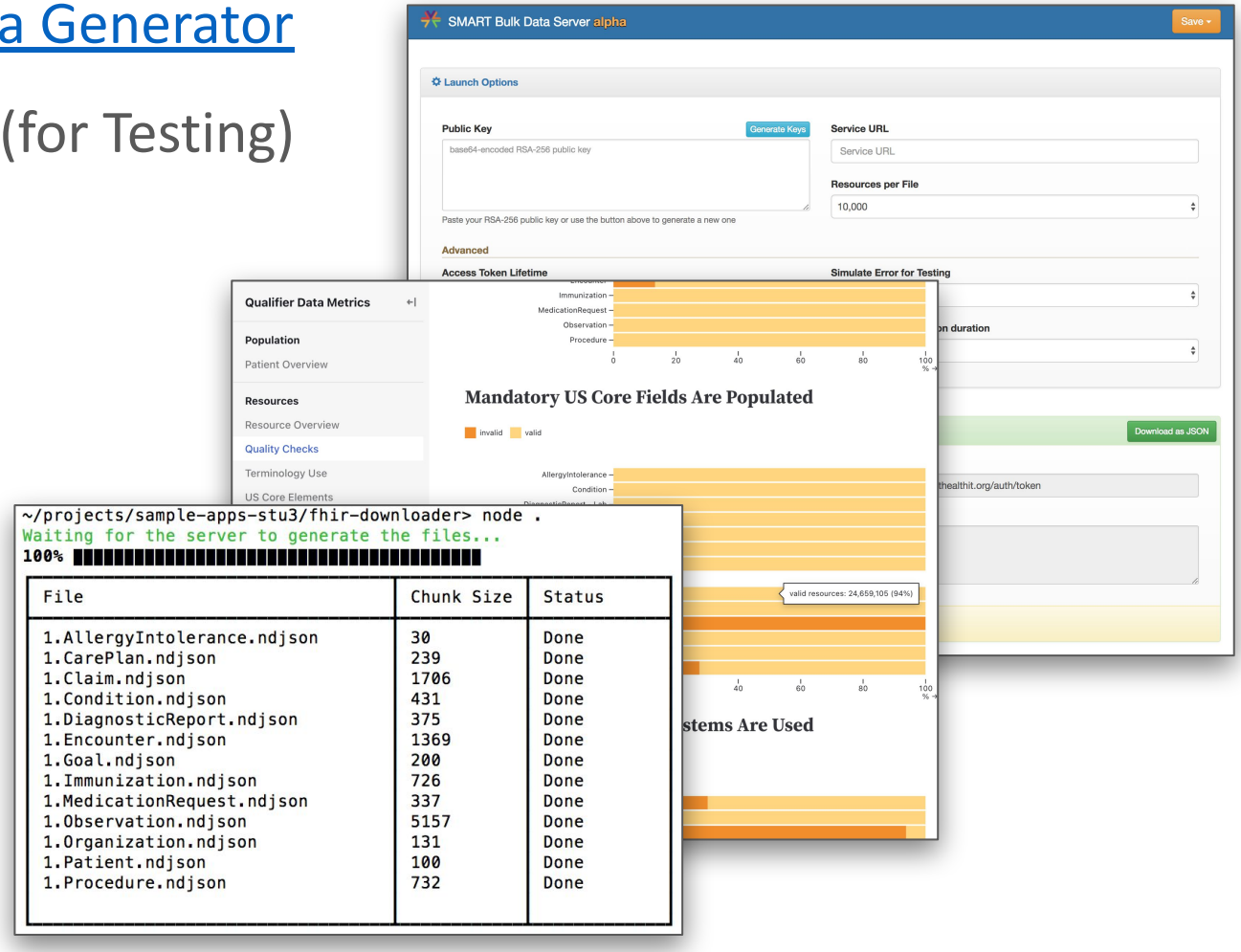
[API Server Reference Implementation] (for Testing)

[Client Reference Implementation]

[CumulusQ Project]

[FlatQuack Project]

[Microsoft Bulk FHIR De-Id Tool]



Slides

https://CentralSquareSolutions.com/duck-deck