WD-VOSI-1.1

- scalability improvement: VOSI-tables
- no changes to VOSI-availability
- no changes to VOSI-capabilities

WD-VOSI-1.1 - REST

- currently:
  - VOSI#tables (1.0) returns a document with all <table>s and <column>s
  - can be excessively large in some TAP services
  - note: we recommend that table name be a fully-qualified table name everywhere
    - e.g. mytable, ivoa.ObsCore, mydb..mytable, ''J/other/PZ/28.7/table1''
- goal:
  - define a scalable VOSI#tables REST binding

WD-VOSI-1.1 - REST

- option #1: REST resources mirror the VODataService model as-is
    - /tables/{schema}
        - e.g. /tables/tap_schema
    - /tables/{schema}/{fully-qualified-table}
        - e.g. /tables/tap_schema/tap_schema.columns
    - but: schema name doesn't have to be part of the fully qualified table name
        - e.g. /tables/public/user123.plan9

WD-VOSI-1.1 - REST

- option #2: flat listing of tables
  - /tables/{fully-qualified-table}
    - e.g. /tables/tap_schema.columns
  - since we recommend fully-qualified table names pretty much everywhere
    - de-emphasize the organisational {schema}
    - /tables contains: a list of tables!!

WD-VOSI-1.1 - REST

- /tables/{fully-qualified-table} returns full metadata for a single table:
  - \<table\> element as document root
  - simple addition to VOSITables xsd to allow
  - VOSI-1.0 clients won't ask, won't get

WD-VOSI-1.1 - scalable

- need a mechanism where /tables returns list of tables only (no column metadata)
  - client decides, service can refuse full dump?
  - service decides, client tries to figure it out?
  - no one decides?

WD-VOSI-1.1 - scalable

- service decides:

  - maybe /tables returns everything (1.0)

  - maybe /tables returns <table>s without <column>s

- client figures it out:

  - if <table> elements have no <column>(s)... make another request to /tables/{table}

- some services are compatible with 1.0, some not (but they don't work now)

WD-VOSI-1.1 - scalable

- client decides:
    - /tables returns everything (1.0)
    - /tables?detail=min returns <table>s without <column>s
- service (maybe) refuses:
    - /tables gives a 403 (forbidden)?
    - /tables gives a 302 (to /tables?detail=min)?

WD-VOSI-1.1 - scalable

- no choices:
  - /tables always returns <table>s without <column>s
- client has to make more requests
  - /tables/{fully-qualified-table}
- less compatible with 1.0:
  - but simple consistent behaviour

Summary/Proposal

- /tables can return full document (compat)

- /tables can redirect to /tables?detail=min

- /tables?detail=min returns <table>s without <column>s

- /tables/{fully-qualified-table} returns <table>
  - one-line addition to VOSITables xsd

- option: modest work but more incompatible to stop using <tableset> and <schema>