OpenCADC - UWS

- Universal Worker Service: UWS
  - control and execution of asynchronous jobs
- cadcUWS library:
  - provides Job class and plugin architecture
  - provides servlet with UWS async behaviour
  - also provides servlet with sync behaviour
  - plugin architecture enables customisation
  - plugin configuration in web.xml
- java + restlet, log4j

OpenCADC - UWS

- JobManager
  - job control and negotiation
  - calls persistence
  - calls executor
  - provided impl: validation and limits
- JobPersistence
  - store and retrieve Job state
  - provided impl: in-memory map
  - working on: persist to database

OpenCADC - UWS

- JobExecutor
  - job execution
  - calls job runner
  - provided impl: ThreadExecutor runs every job in a new thread
  - working on: ThreadPoolExecutor
- JobRunner
  - code that actually runs the job
  - service must provide implementation

OpenCADC - TAP

- Table Access Protocol: TAP
  - asynchronous queries
  - synchronous queries
- cadcTAP library:
  - QueryRunner implements JobRunner
  - plugin architecture enables customisation
  - configuration of plugins TBD
- java + spring, jdom, jsqlparser, javacsv

OpenCADC - TAP

- TapSchema
  - library of java classes
  - DAO class to read from database
  - TAP_SCHEMA DDL statements (SQL)
  - self-describing content (SQL insert statements)
  - used by query parser(s) to validate table and column usage
  - used by TableWriter(s) to add metadata
  - working on: TapSchema -> XML for VOSI tables resource

OpenCADC - TAP

- UploadManager (plugin)
  - handles UPLOAD parameter
  - provided impl: UnsupportedUploadManager
    - throws UnsupportedOperationException if there are UPLOADs
  - working on: BasicUploadManager
    - download and parse VOTable, generate DDL, sanity check/parse table content, insert
    - returns table metadata (using TapSchema classes) so parser can validate table/column usage and modify table names in query

OpenCADC - TAP

- TapQuery interface
  - implemention for each value of LANG
  - parse query parameter(s)
  - map select-list to TapSchema
  - process query to local SQL
  - provided impl: SqlQuery (LANG=SQL)
  - provided impl: AdqlQuery (LANG=ADQL)
  - configuration TBD

OpenCADC - TAP

- SqlQuery implements TapQuery
  - configured to handle LANG=SQL
  - syntax validation via jsqlparser
  - re-usable TapSchema validation
  - fully navigates the query
    - including subqueries in the FROM, WHERE, and HAVING clauses
  - not all native SQL constructs will get past the jsqlparser

OpenCADC - TAP

- AdqlQuery implements TapQuery
  - configured to handle LANG=ADQL
  - syntax and TapSchema validation
  - multi-pass query processing using visitor pattern
    - convert TOP to LIMIT (for PostgreSQL)
    - find all ADQL region constructs
    - convert ADQL region constants to pgSphere constants, TODO: REGION(<stc-s>)
    - convert CONTAINS/INTERSECTS=0|1 into pgSphere operators
  - configuration of internals/dialect TBD

OpenCADC - TAP

- TableWriter (plugin)
  - TableWriterFactory validates FORMAT
    - instantiates a TableWriter
    - configuration TBD
  - provided: VOTableWriter
    - uses TapSchema and select-lists for metadata and to setup formatter(s) for each column
    - write ResultSet or Throwable
    - configuration of formatters TBD
  - provided: AsciiTableWriter (CSV and TSV)
    - write ResultSet

OpenCADC - TAP

- FileStore (plugin)
  - File getStorageDir()
  - URL put(File f)
  - simple implementation:
    - could use a work dir under the web server document root
    - would then generate a URL served by web server
  - CADC implementation:
    - put the file into our internal storage system
    - generate URL to our standard data access service

OpenCADC - TAP

- QueryRunner implements JobRunner
  - set job state (phase + result or error)
  - use FileStore to manage files/URLs
  - validate REQUEST, VERSION, LANG, MAXREC
  - find DataSource(s) via JNDI
  - read TapSchema
  - use UploadManager
  - use TapQuery
  - use TableWriter

OpenCADC - TAP

- right now: it works but not ready for primetime :)
- cadcUWS and cadcTAP are libraries
  - service implementor creates and deploys webapp
- TODO:
  - configuration of plugins/components
  - refactor ADQL/SQL parser to support re-use, other back-end DBs, custom extensions
- all code available at:

  http://code.google.com/p/opencadc/