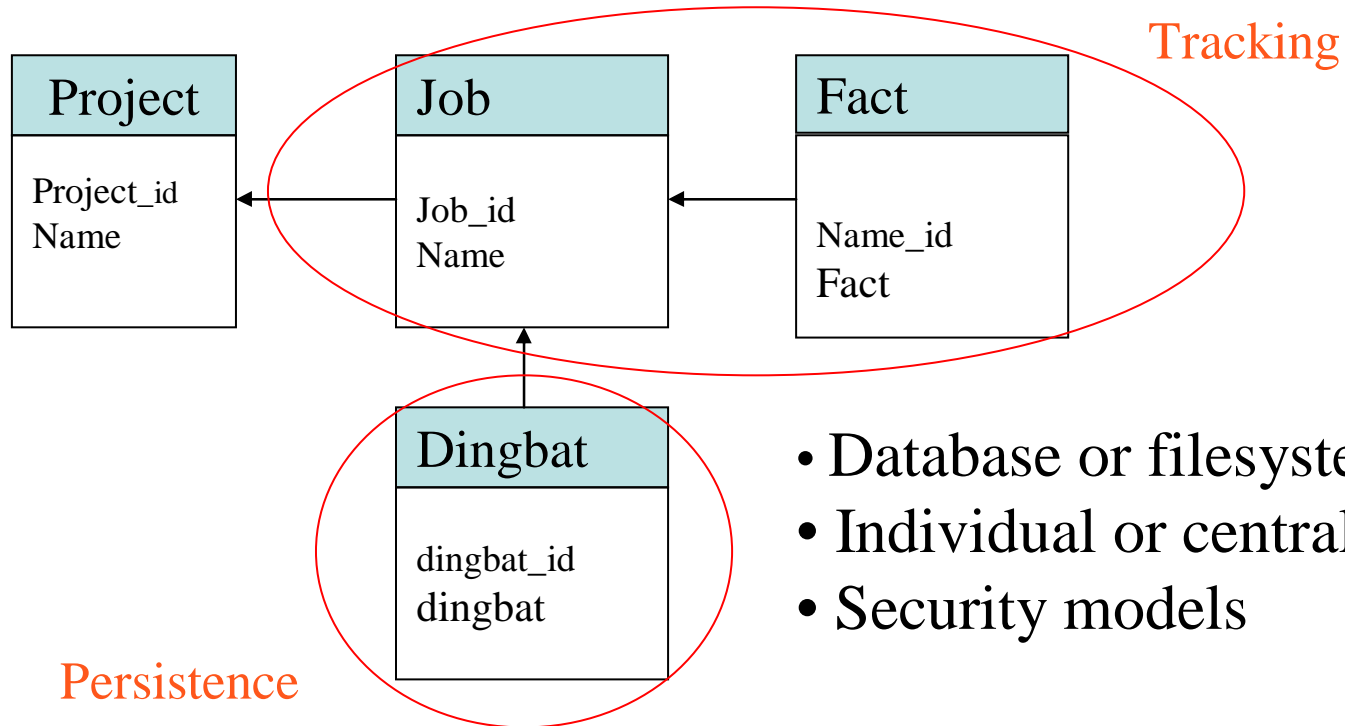


“Database” usage and requirements for CCP4 automation

Projects at Daresbury

- HAPPy - experimental phasing
 - DJR, PE, EJD, CCB
- XIA-DPA - dataprocessing (other)
 - GW (DNA)
- BMP - molecular replacement
 - RK, MDW (NDS)
- CCP4i
 - PJB
- Data storage
 - WY, PJB

Current state of affairs



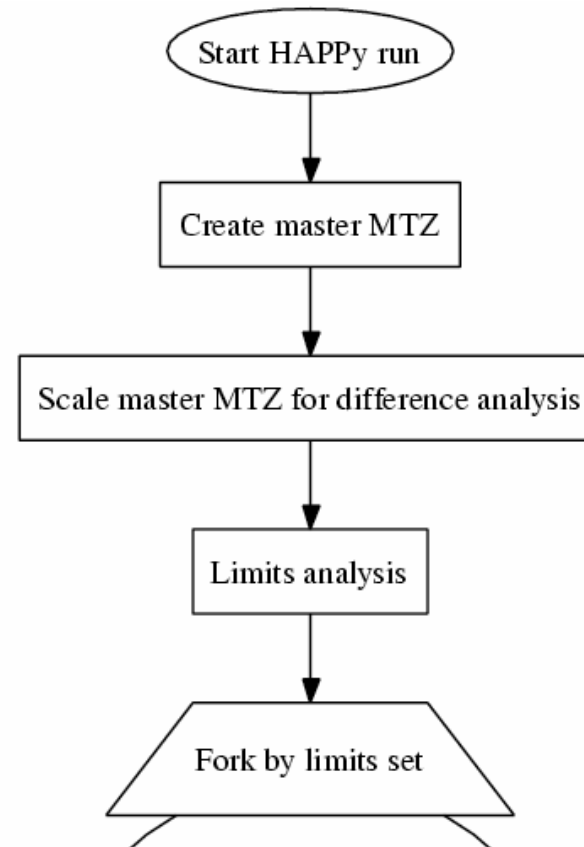
- Database or filesystem
- Individual or central
- Security models

(Loose) Storage Requirements

- Tracking (Jobs) - ccp4i, HAPPy
 - Data types
 - Knowledge base - ccp4i
 - Object persistence - GW
 - Templates and protocols - eHTPX
- } External view
- } Project
- } Control

Job Tracking

- Workflow view
- Database or filesystem
- Track progress
- ccp4i gives good example



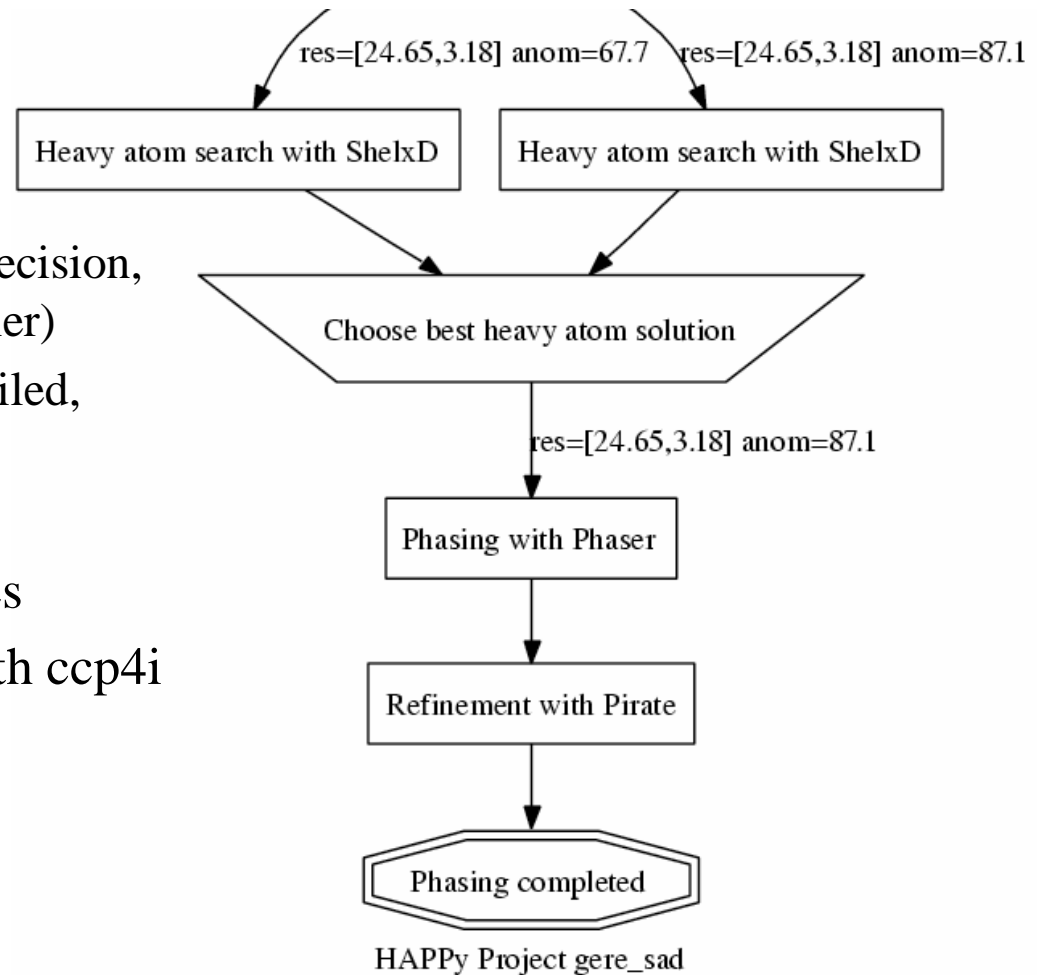
- HAPPy view

- Nodes (actions)

- Type: Job, Fork, Decision, Start, End (Container)
 - Status: Success, Failed, Killed, Running
 - Time stamps

- Human readable notes

- Designed to work with ccp4i database.



Data tracking

- Meta data (describe file)
 - URI
 - Wavelength
 - Cell
 - Column names
- Source and usage

Knowledge base

- Small amounts of data -> xml files, otherwise database
- Project and pipeline independent
- mmcif definitions at deposition end
- Datamodel (?)
 - HA positions and statistics
 - x, y, z, occ, anon_occ, b
 - Sequence
 - Solvent fraction
 - MR models

Persistent Objects (databucket)

- Project specific (unless common)
- Store local state
 - object orientated database (ZODB)
 - dingbats
 - .xml file
- HAPPy has HAPPy state object
 - always in memory
 - current state and history
 - persistence required for restart

Protocols and Templates

- User preferences
- Pathways / Workflow