

MXLIMS

Global Phasing

MXCuBE/ISPyB Meeting, Lund, May 2024

Introduction

Overview and scope

- Scientific API for communication with LIMS and beamlines
 - Message specification and data model
 - Usable across multiple Synchrotrons, beamlines and LIMS systems
 - Detailed and flexible enough to handle all traffic
 - Extensible to allow for additional site- and program- specific data
 - Each site and program gets its own namespace to define its own extension schemas
- Scope: MX and related techniques
 - Extension to SSX contemplated

MXLIMS Implementation

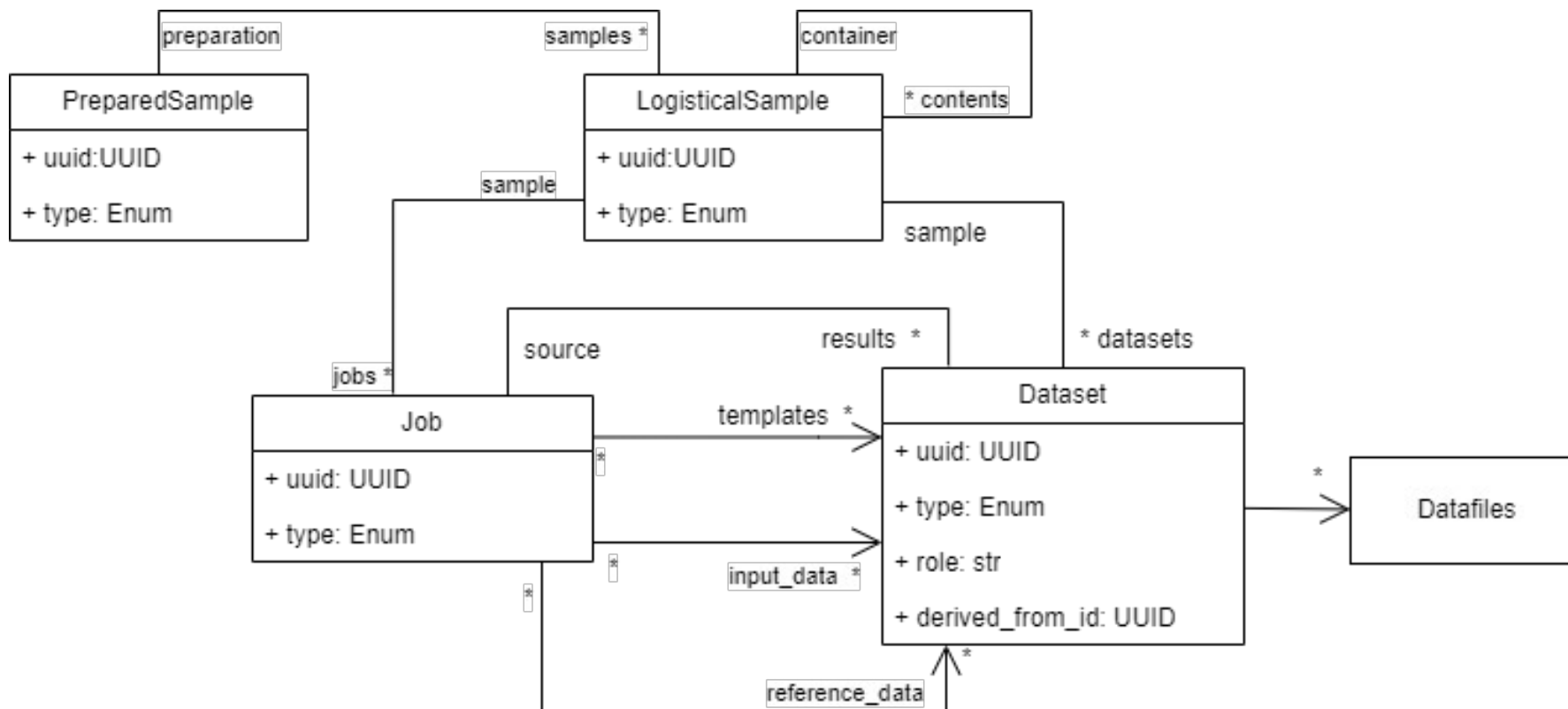
- Multiple messages defined as JSON schemas
- Simple core structure, to allow storage in Mongo-DB or data-catalogue type systems
 - Few core types, with most data as variable metadata / parameters
 - Parameters in JSON schemas that are specific to one kind of data; core model can be applied to multiple techniques
- No direct application to LIMS system implementation
 - MXLIMS defines data and the structure and level of complexity required
 - Data transfer to LIMS (or beamline control) systems via converters
 - MXLIMS does not limit implementations or program internal data models

- Beamline-LIMS communication supporting multiple LIMS systems
 - Multi-synchrotron sample shipping, including diffraction and processing plans
 - Detailed specification of experiment and processing program input
 - Reporting of processing results to LIMS
 - Query, viewing and summary of experiments and processed data sets
-

- Samples and shipping (Ed Daniel, with Karl Levik)
 - Detailed JSON schemas (not finalized)
 - MX sample holders and crystals
Shipment, Dewar, Puck, Pin, Plate, Well, Drop, Crystal, annotated Crystal Image
 - Sample contents (Sample, Protein, Ligand) need filling in
 - MX
 - Detailed draft, no schemas yet
 - *Jobs*: CrystallographicExperiment and CrystallographicProcessing
 - *Datasets*: CollectionSweep and ReflectionSet
 - Data types: UnitCell, Scan, ReflectionStatistics, QualityFactor
-

Model contents

Core structure, Samples, MX Experiments, MX processing,

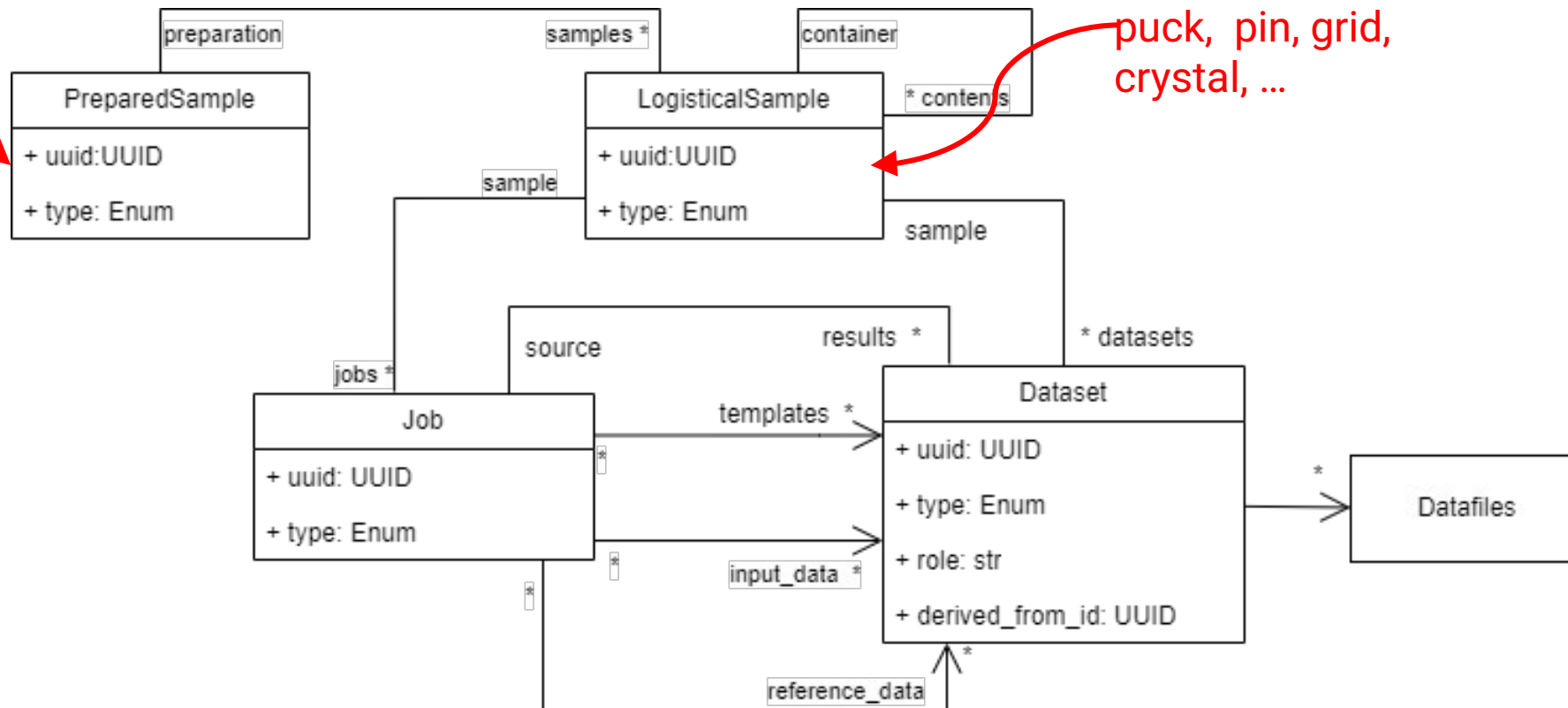


Four classes – with UUID so you can reference them elsewhere

Separate schemas for each specific kind of data

Core model - Samples and shipping

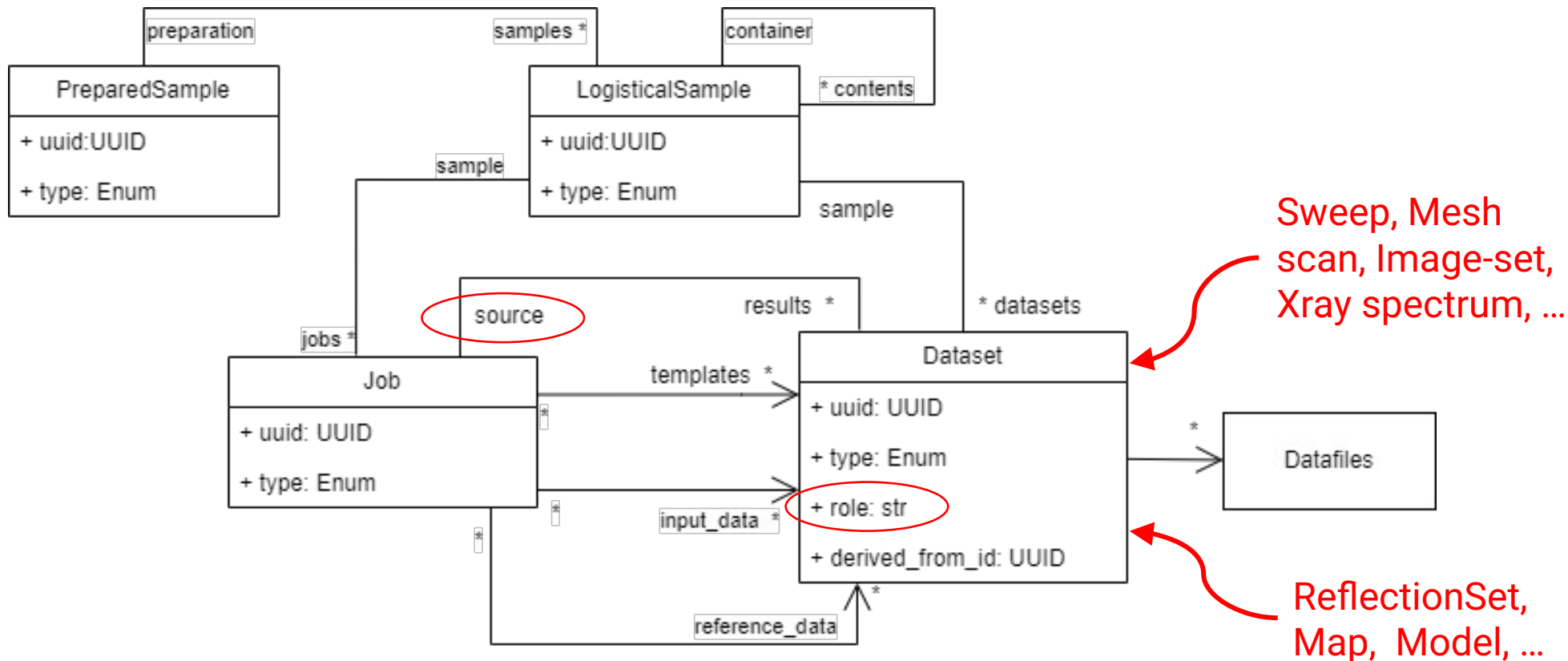
Protein, ligands,
crystal form,
preparation, ...



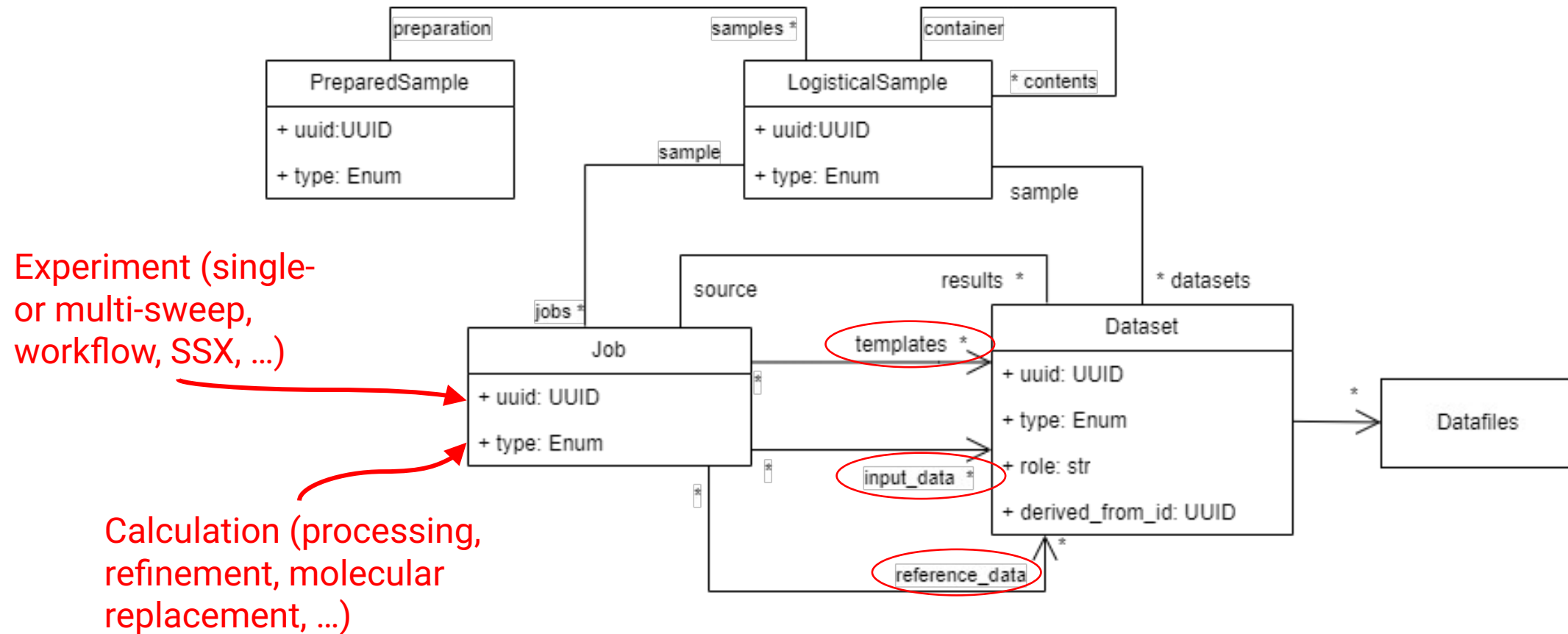
Shipment, plate,
puck, pin, grid,
crystal, ...

- Multi-crystal drops and pins
 - Multiple independent positions per pin
 - Multiple homologous crystals, e.g. drops, Crystal-Direct
 - Multi-crystal short sweeps in a single sample
- Crystals identified as part of the experiment
 - LCP sheets, SSX slurries, XFEL
- Homogenous many-crystal samples
 - SSX grids, slurries, XFEL

Core model - Datasets



Core model - Jobs

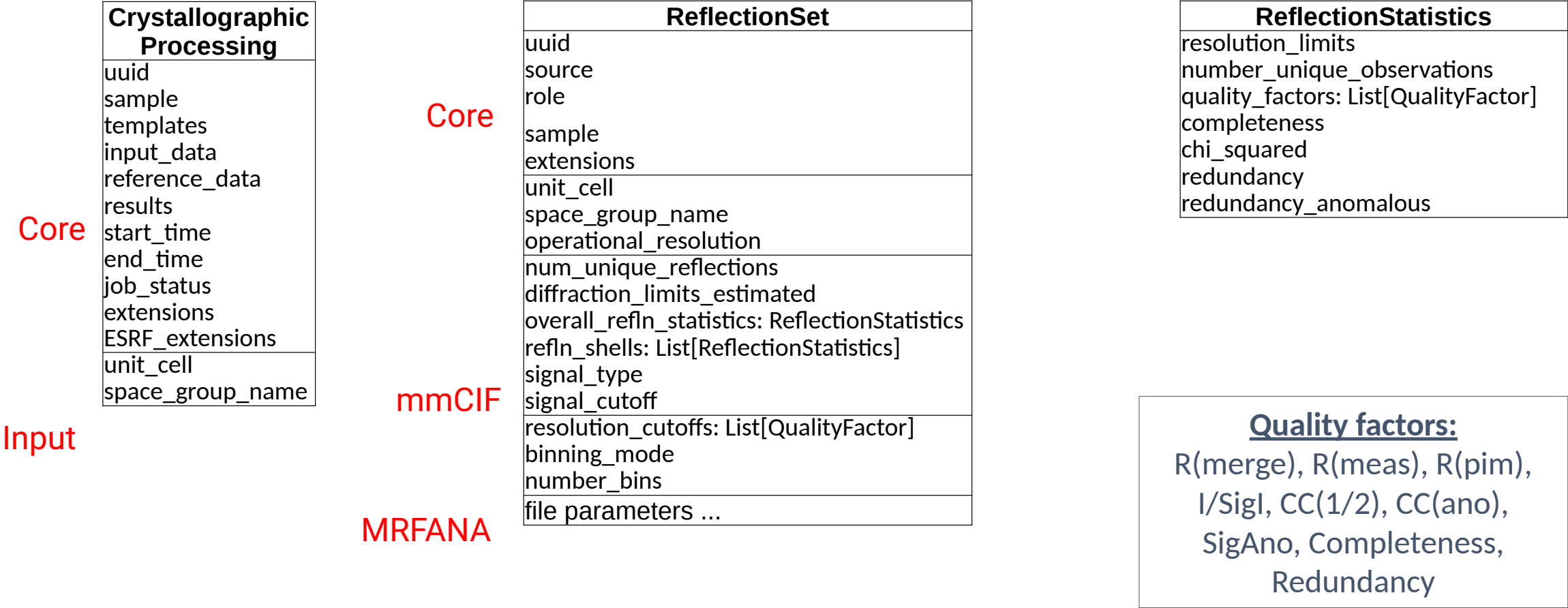


MX acquisition (selected fields)

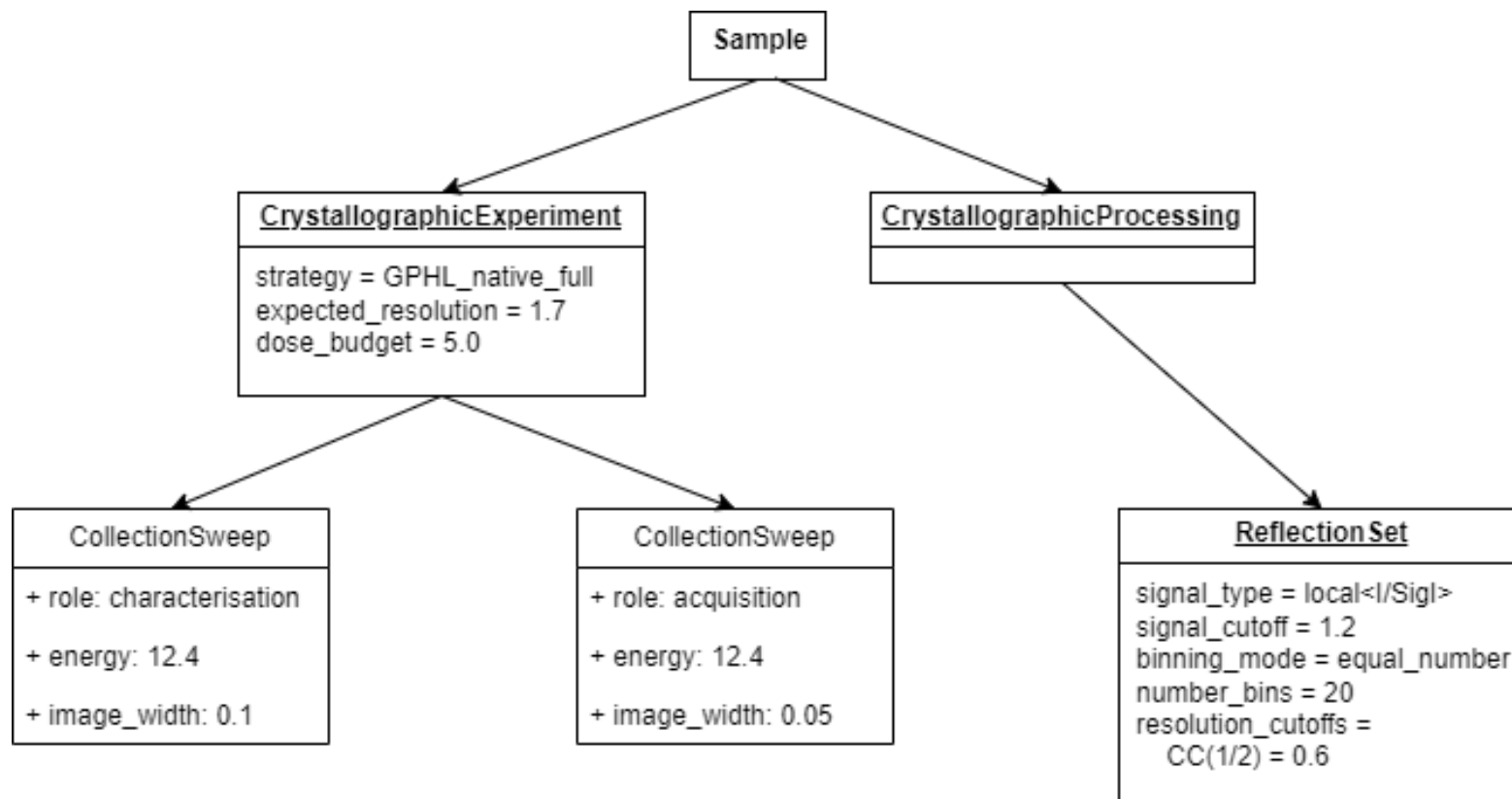
Core	Crystallographic Experiment	Core	Collection Sweep	Scan
	uuid		uuid	scan_position_start
	sample		source	first_image_no
	templates		role	num_images
	input_data		sample	ordinal
	reference_data		extensions	
	results		exposure_time	
	extensions		image_width	
	experiment_strategy		energy	
	expected_resolution		transmission	
Input	dose_budget	detector parameters ...		
	wedge_width	axis_positions_start		
	measured_flux	scan_axis		
	radiation_dose	axis_positions_end		
Output	unit_cell	Scans : List[Scan]		
	space_group_name	file parameters ...		

- Workflow and multi-sweep experiments
 - Multi-sweep experiment viewers must be organized by *Job* and not by Dataset
- Helical scan and other multi-motor scans
- Interleaved acquisition
 - Inverse beam, wavelength interleaving, ...
- Derived Datasets
 - E.g. for processing subset of a Sweep or multi-crystal image set

MX Processing (selected fields)



Diffraction and Processing plan example



The same schemas are used for diffraction plan, experiment and processing input, and experiment and processing result

Sessions, Users, and access control

- Administrative information: Necessary but not part of MXLIMS
 - Personal data make for legal problems with data protection
 - Different sites have different requirements, which may be hard to generalize reliably
- Relevant information should go into site-specific extensions
 - User IDs
 - Session IDs
 - Project codes
 - Contact persons
 - Etc.

Protein, ligands,
crystal form,
preparation, ...

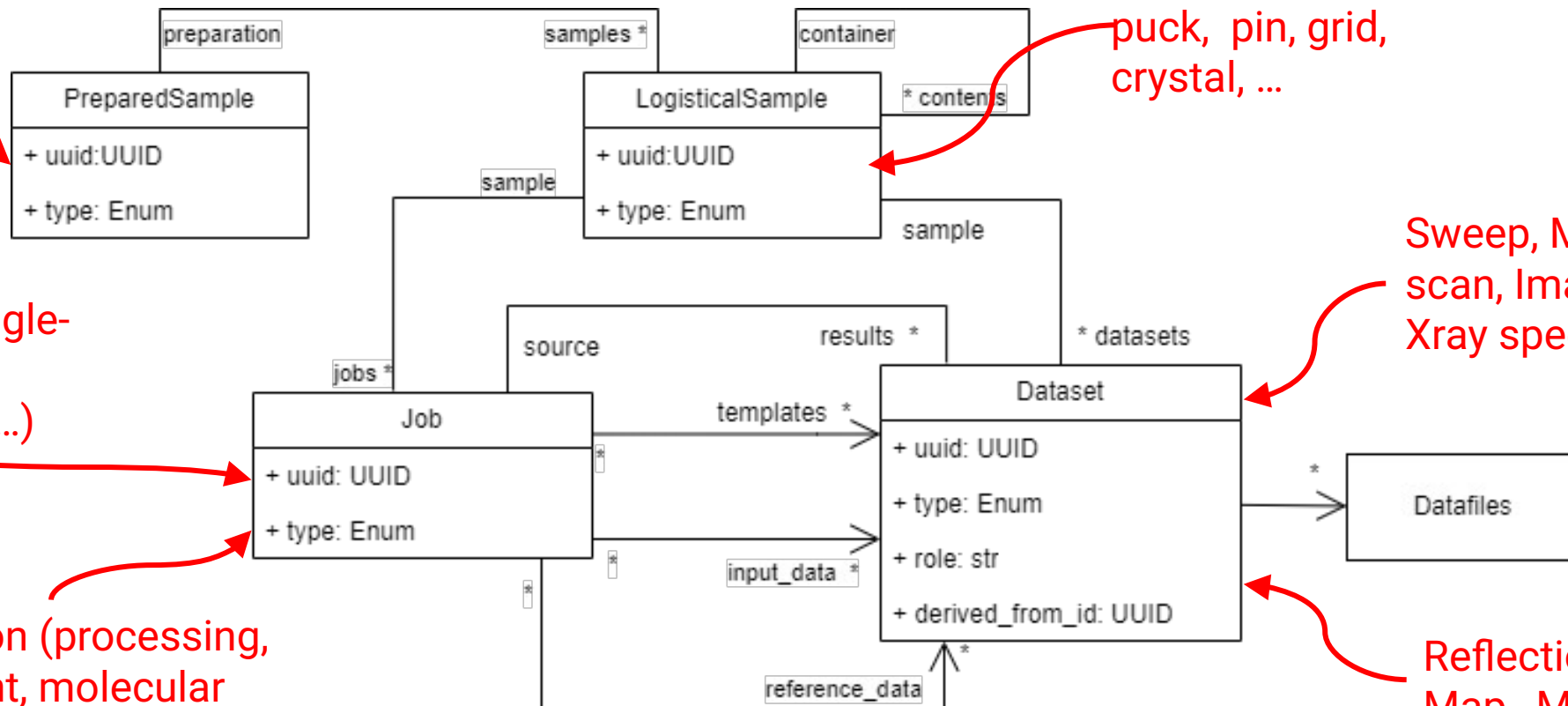
Experiment (single-
or multi-sweep,
workflow, SSX, ...)

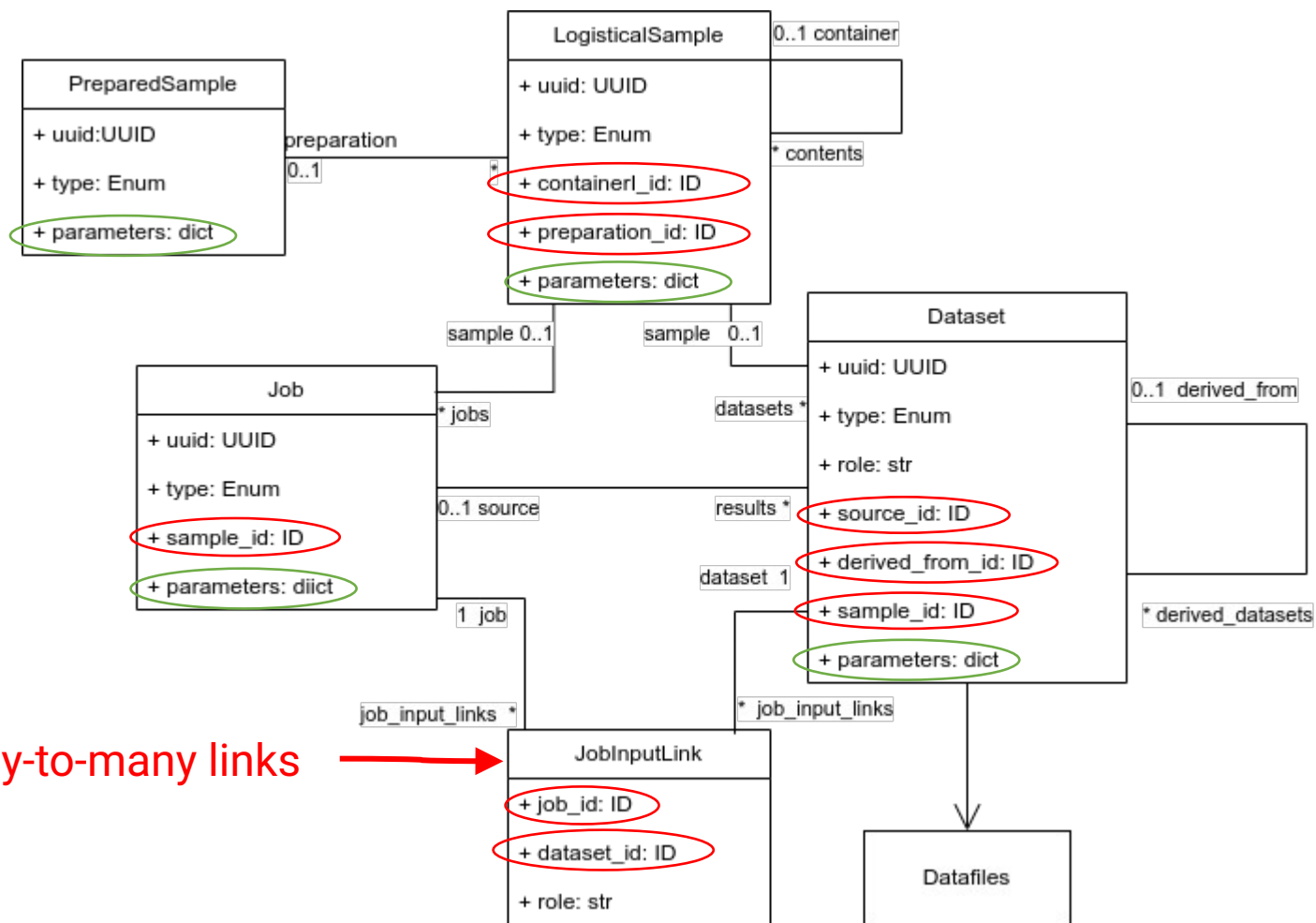
Calculation (processing,
refinement, molecular
replacement, ...)

Shipment, plate,
puck, pin, grid,
crystal, ...

Sweep, Mesh
scan, Image-set,
Xray spectrum, ...

ReflectionSet,
Map, Model, ...





Many-to-many links



Provenance and tracking

- Datasets have source Job (Experiment or calculation)
- Jobs have input Datasets (input_data, reference_data, and templates)
- Derived Datasets allow modifying Datasets for processing input
- LogisticalSample linked to either Job or Dataset
- PreparedSample can be linked to crystal or crystal holder (Drop, Grid, ...)

NOTE: API and messages do *not* require filling in all the links



Global Phasing Limited

END