

# MXLIMS

Global Phasing

MXCuBE/ISPyB Meeting, Lund, May 2024

---

# Introduction

Overview and scope

# MXLIMS purpose

---

- 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

# Main use cases

---

- 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
-

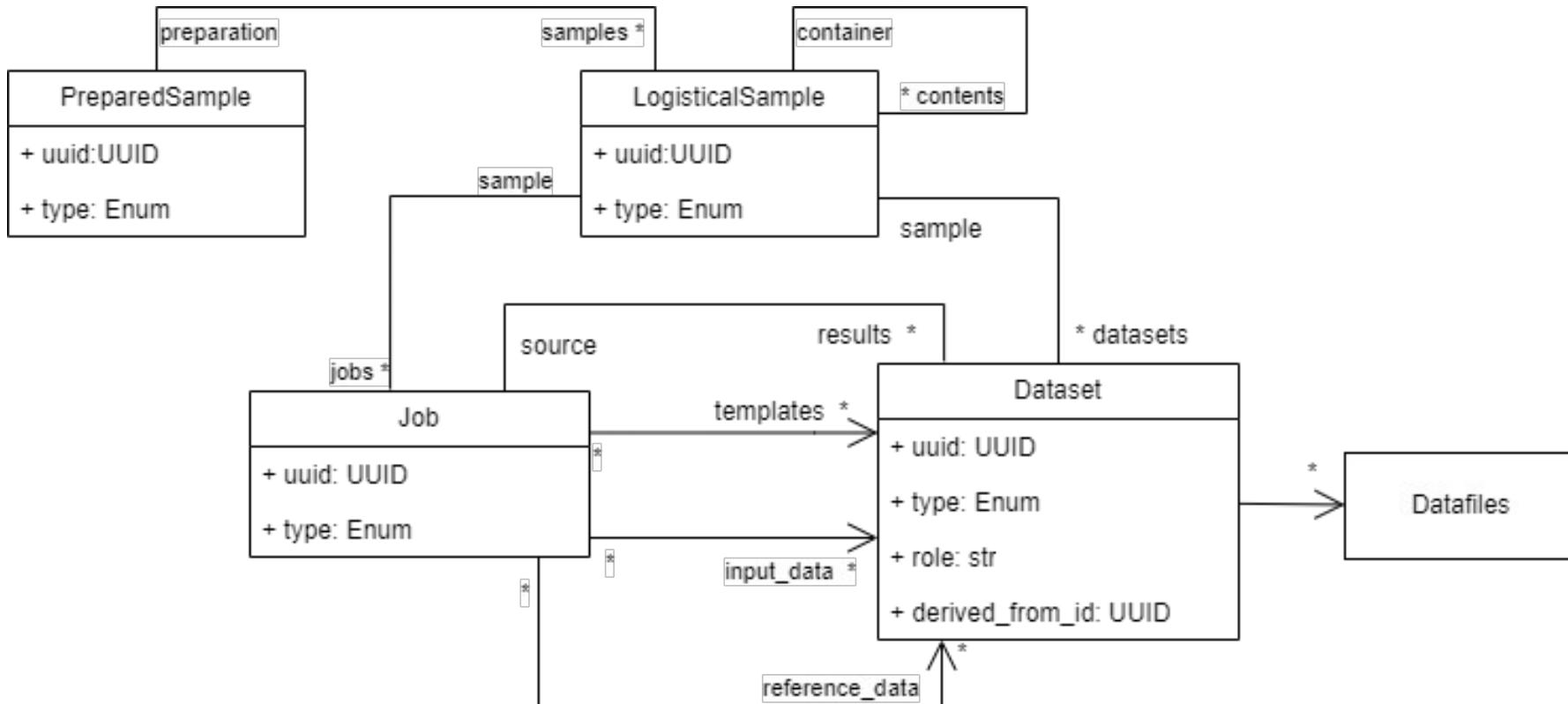
- Samples and shipping (Ed Daniel, with Karl Levik)
  - Detailed JSON schemas (not finalized)
    - MX sample holders and crystals  
Shipment, Dewar, Puck, Pin, Plate, Well, Drop, Crystal, annotated Crystal Image
    - Sample contents (Sample, Protein, Ligand) need filling in
- MX
  - Detailed draft, 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,

---

# Core model

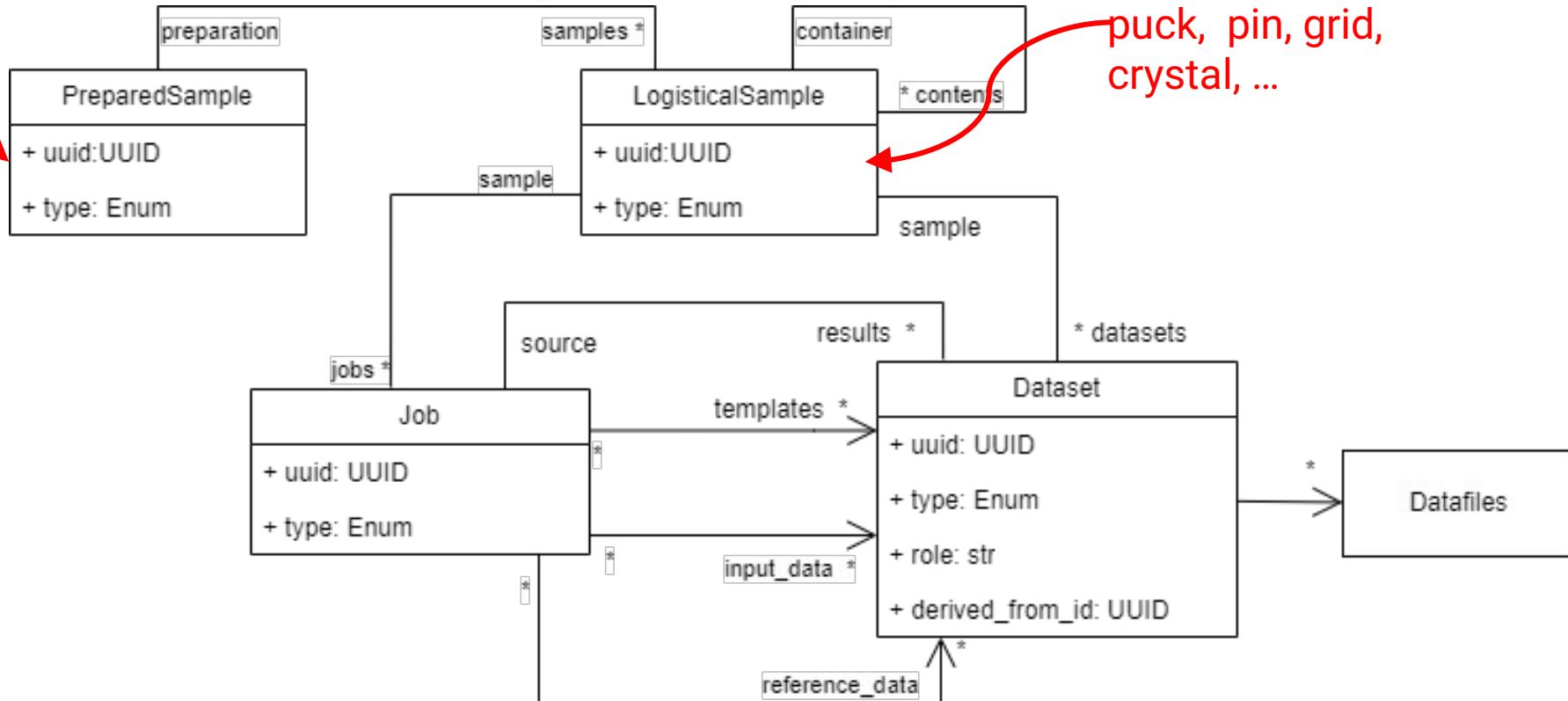


Four classes – with UUID so you can reference them elsewhere

Separate schemas for each specific kind of data

# Core model - Samples and shipping

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

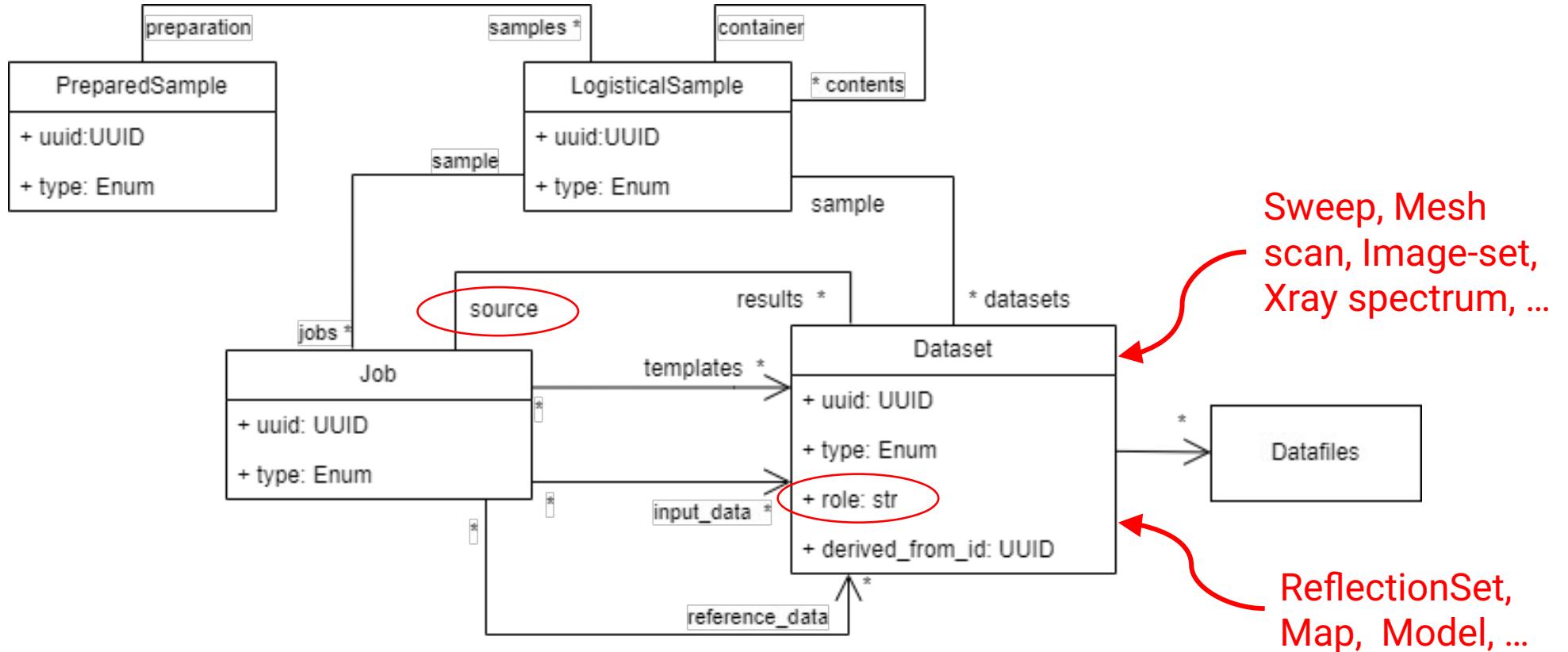


# Special cases - Samples

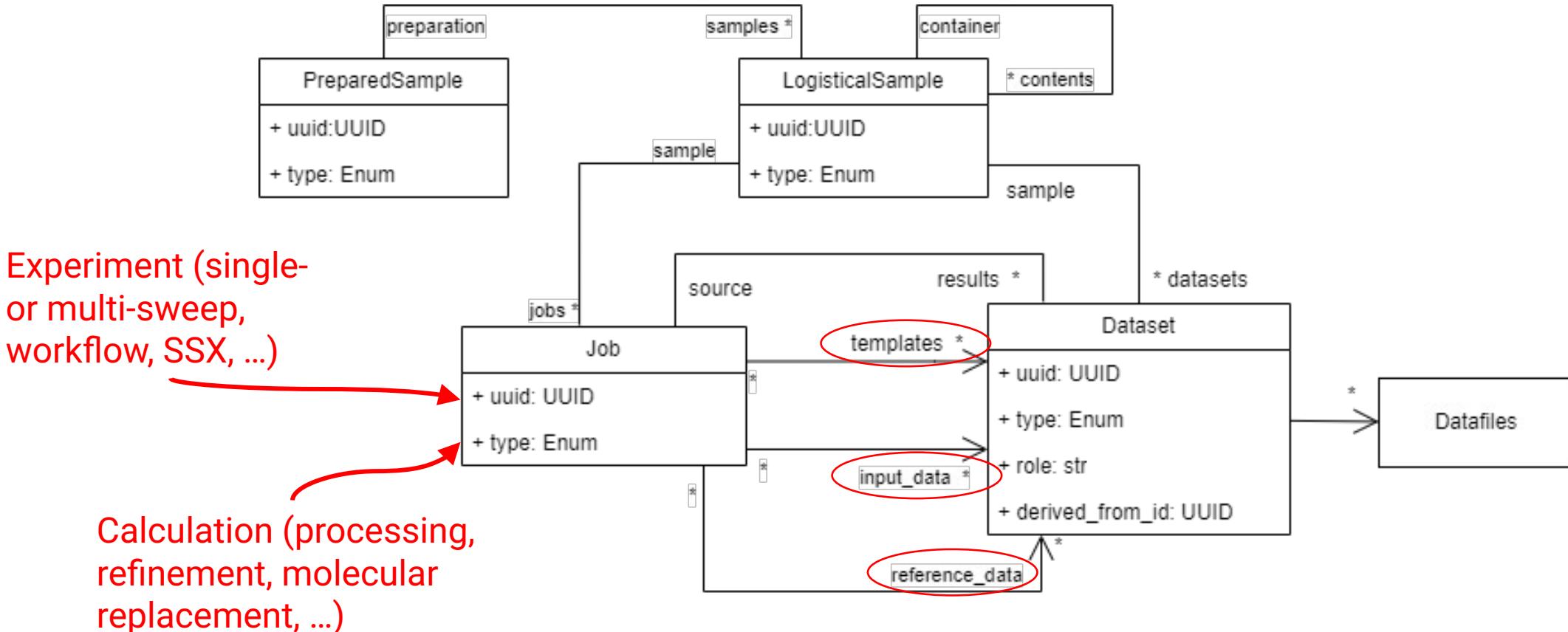
---

- 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)

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

# Special cases - MX

---

- 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)

Core

Input

Crystallographic Processing	
uuid	
sample	
templates	
input_data	
reference_data	
results	
start_time	
end_time	
job_status	
extensions	
ESRF_extensions	
unit_cell	
space_group_name	

Core

mmCIF

MRFANA

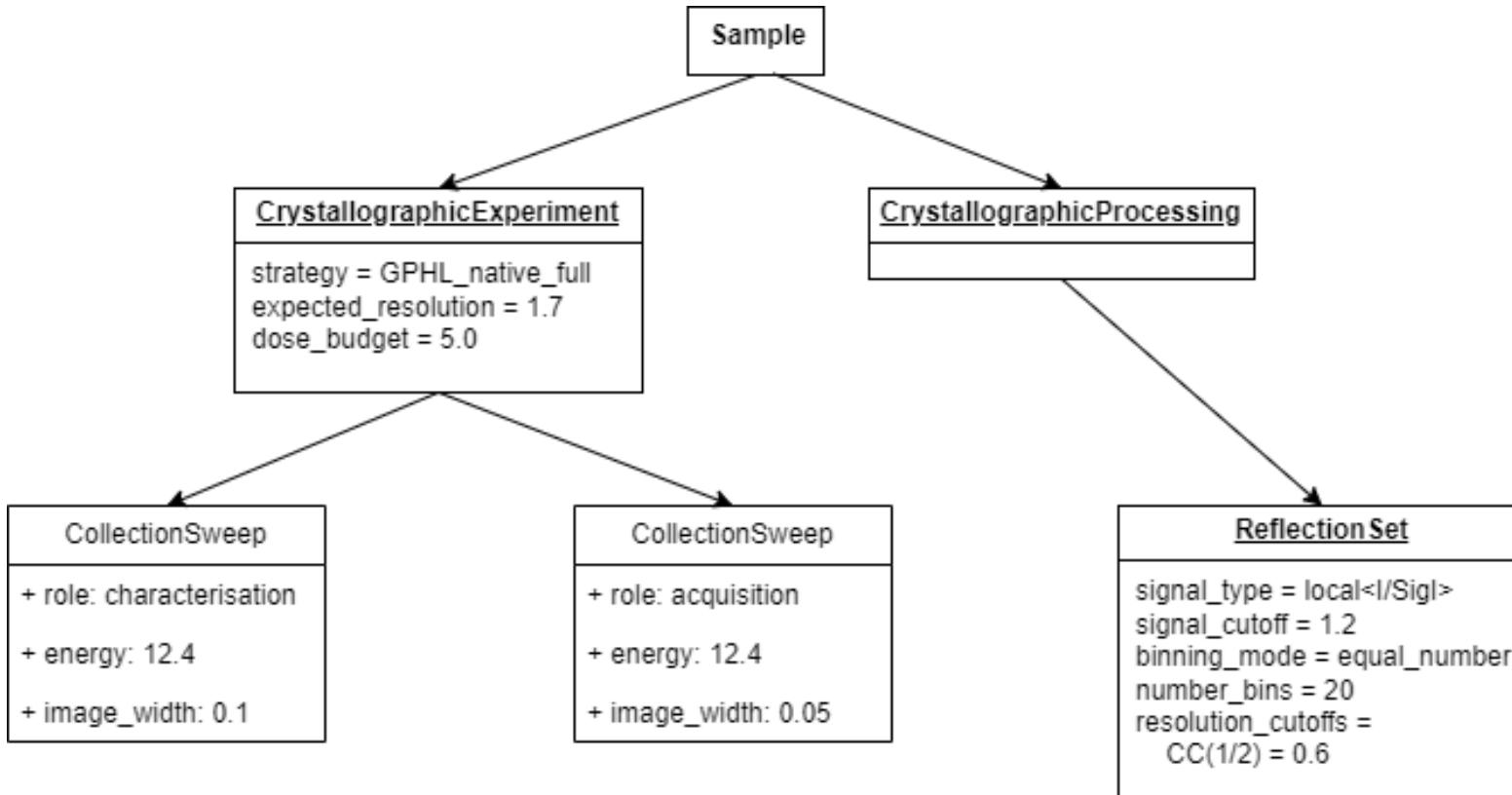
ReflectionSet
uuid
source
role
sample
extensions
unit_cell
space_group_name
operational_resolution
num_unique_reflections
diffraction_limits_estimated
overall_refl_statistics: ReflectionStatistics
refln_shells: List[ReflectionStatistics]
signal_type
signal_cutoff
resolution_cutoffs: List[QualityFactor]
binning_mode
number_bins
file parameters ...

ReflectionStatistics
resolution_limits
number_unique_observations
quality_factors: List[QualityFactor]
completeness
chi_squared
redundancy
redundancy_anomalous

## Quality factors:

R(merge), R(meas), R(pim),  
I/SigI, CC(1/2), CC(ano),  
SigAno, Completeness,  
Redundancy

# Diffraction and Processing plan example



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

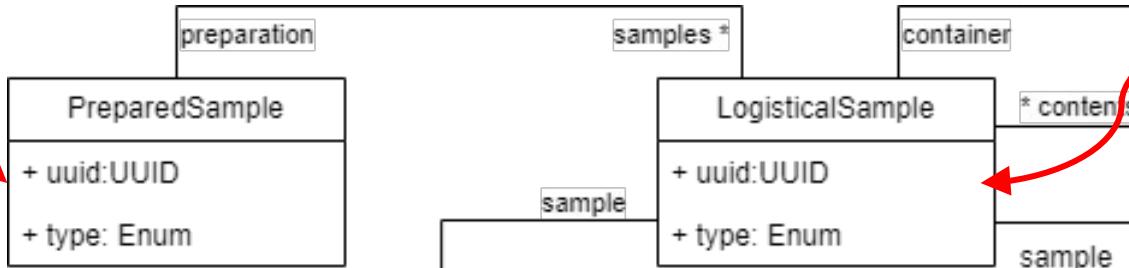
# Sessions, Users, and access control

---

- 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.
-

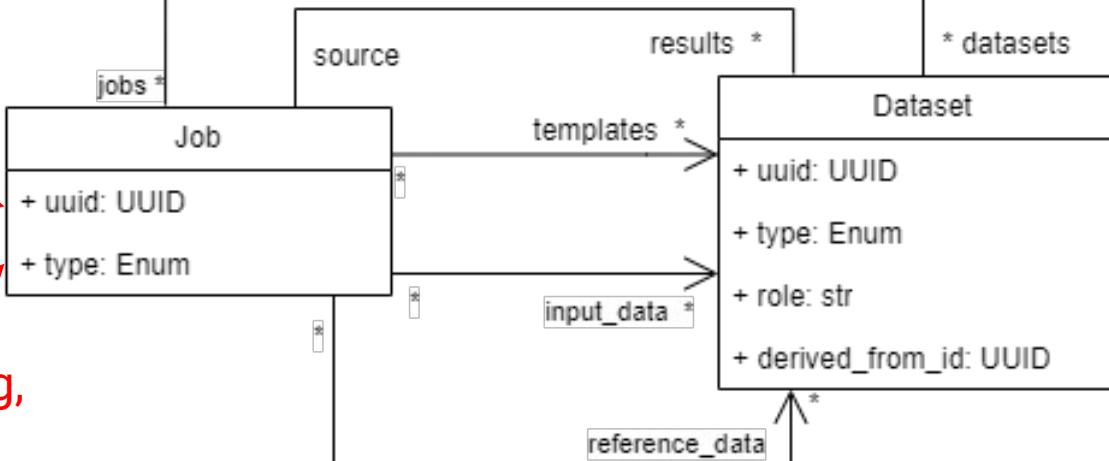
## Core model

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



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

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

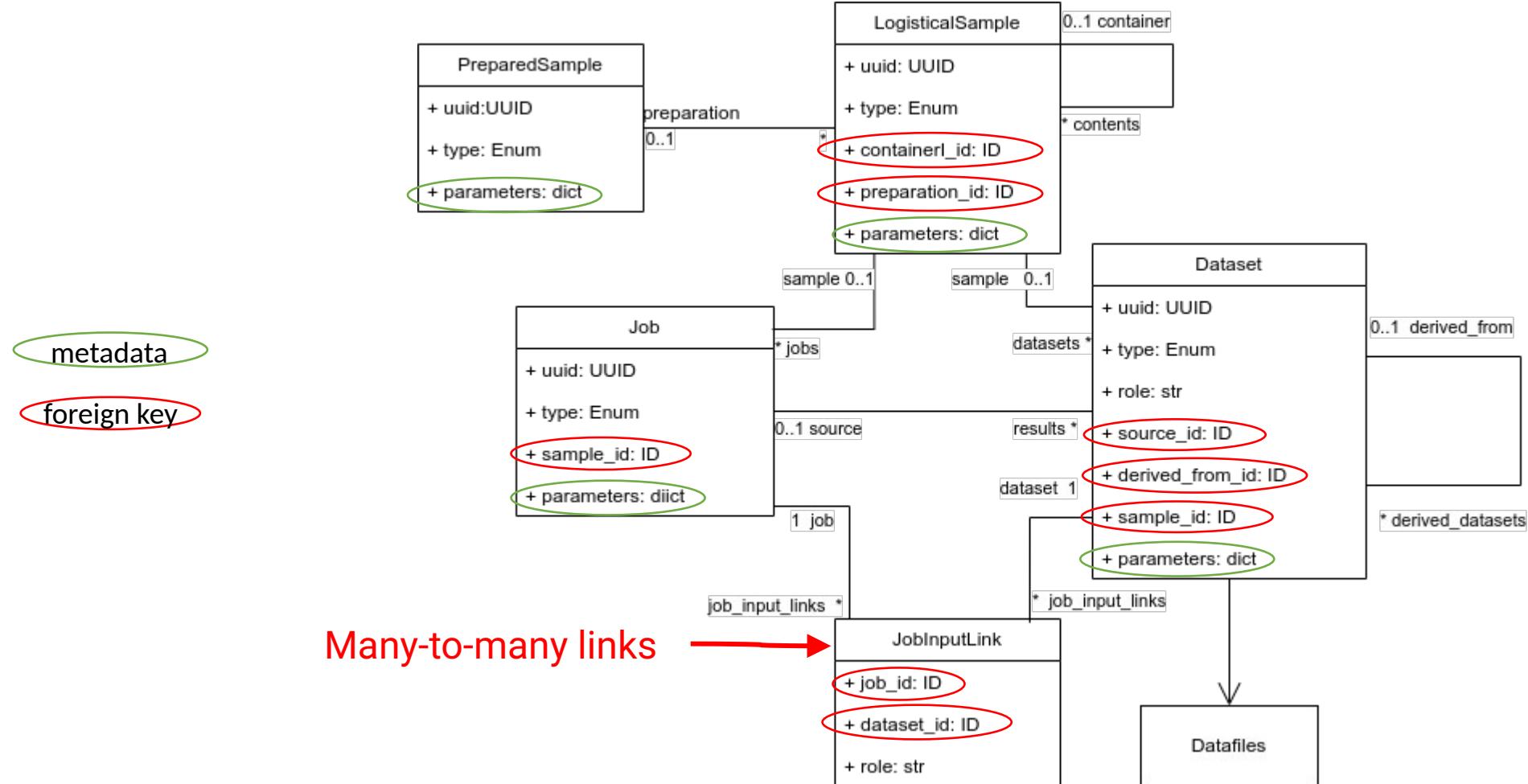


Sweep, Mesh  
scan, Image-set,  
Xray spectrum, ...

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

ReflectionSet,  
Map, Model, ...

## Table model



# 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

---

END

---