

The python-ihm library

<https://github.com/ihmwwg/python-ihm>

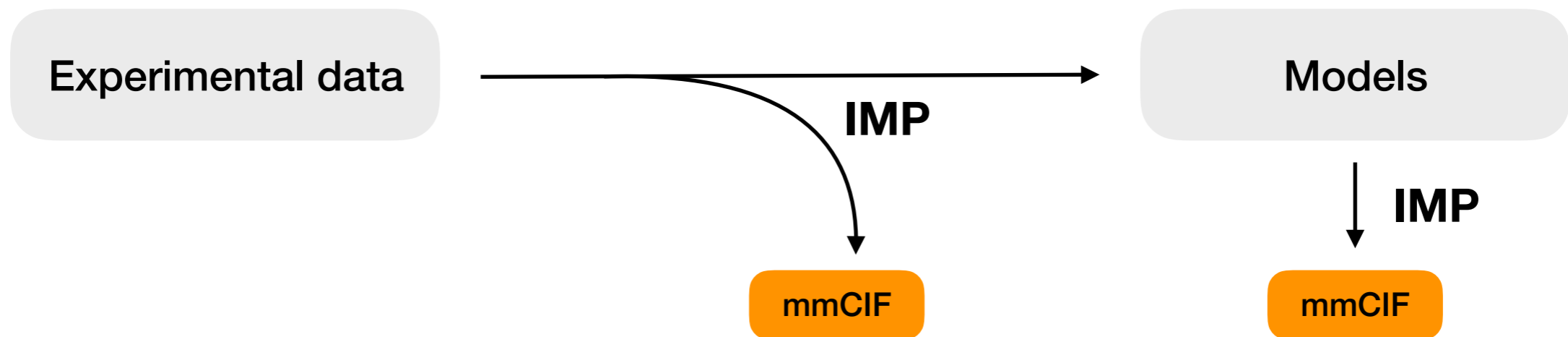
Objective

- Provide a mechanism to programmatically generate IHM mmCIF files
- Can't just write a "PDB to IHM mmCIF" converter since IHM is a *superset* of PDB (e.g. this would lose all experimental information, which defeats the point of IHM)
- Need to merge coordinates ("PDB files") with other input data (e.g. experiments), and information about how the modeling was done, into the complete IHM system

python-ihm overview

- Represents an IHM system as a set of Python classes
- Provides a mechanism to write these classes to an mmCIF file, or to read classes from mmCIF
- Intended to be used as a translation layer within modeling or visualization software, or standalone

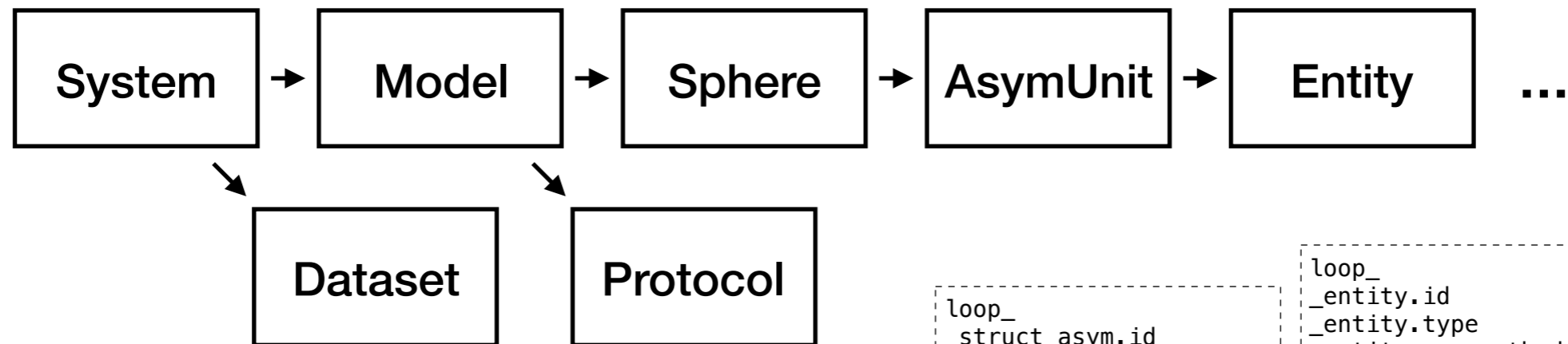
mmCIF output in the Sali lab



- IHM mmCIF files are generated either as a side effect of the modeling with IMP, or from output models
- More or less equivalent as IMP models already contain much of the input data and modeling protocol info
- But: two code paths, and IMP-specific code
- python-ihm is basically this code, cleaned up and with IMP dependencies removed

Data model

- python-ihm represents the system as a hierarchy of Python classes that roughly correspond to IHM/PDBx mmCIF categories, e.g.



```
system = ihm.System()
entityA = ihm.Entity('AAA', description='Subunit A')
asymA = ihm.AsymUnit(entityA, details='Subunit A')
```

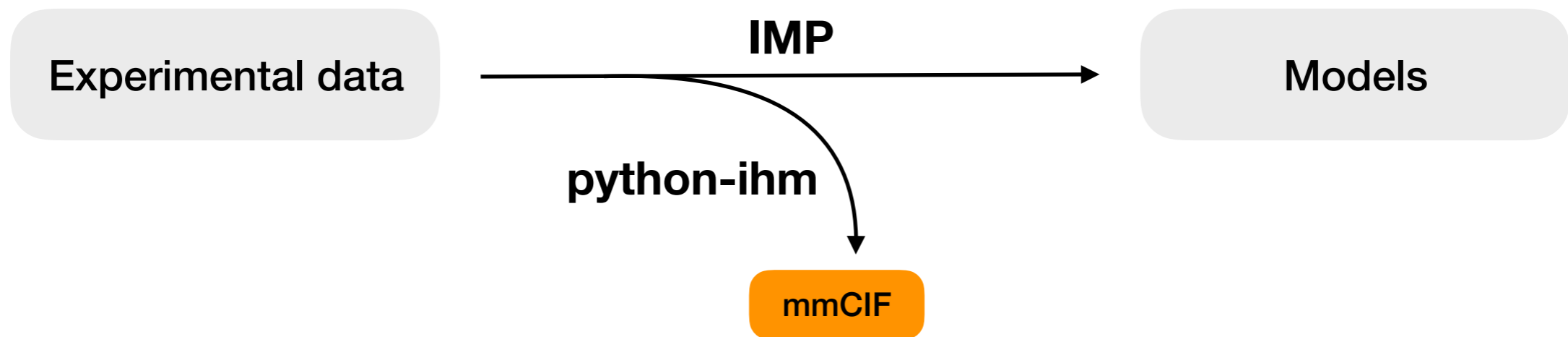
```
loop_
_struct_asym.id
_struct_asym.entity_id
_struct_asym.details
A 1 'Subunit A'
```

```
loop_
_entity.id
_entity.type
_entity.src_method
_entity.pdbx_description
_entity.formula_weight
_entity.pdbx_number_of_molecules
_entity.details
1 polymer man 'Subunit A' ? 1 .
```

- Python pointers (entityA, asymA) are used rather than IDs (1, A) throughout
- Lightweight: tries to avoid duplicating data where possible (instead passing through to native objects)

Usage

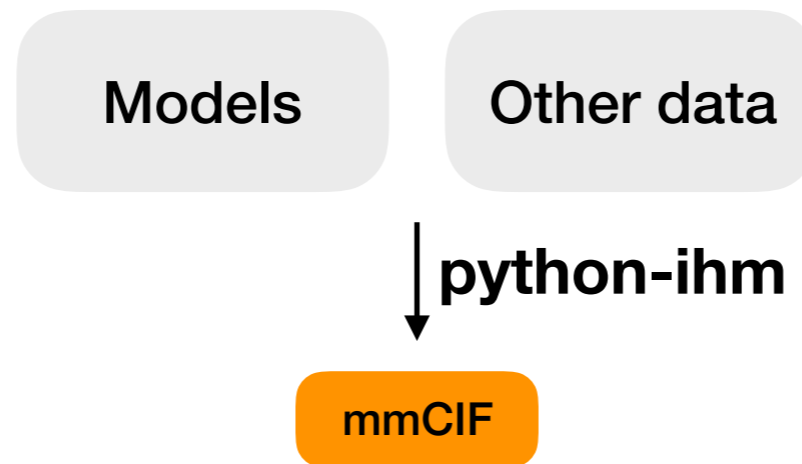
1. As part of existing modeling software



- IMP's internal data model is mapped to IHM's, generating an ihm.System object
- User can tweak things (e.g. add citations) before writing out mmCIF, using python-ihm's API
- See <https://salilab.org/npc2018> for a recently published system

Usage

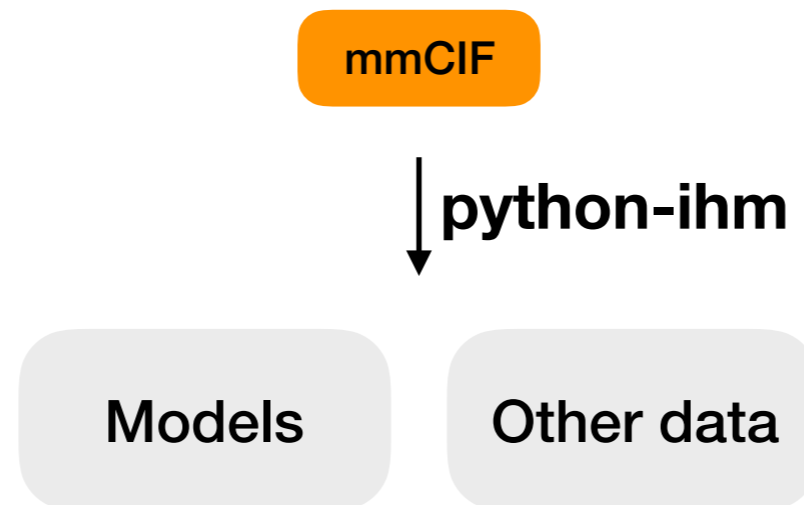
2. Standalone



- Simple Python script can be used to populate an IHM system (and thus an mmCIF file) from model coordinates plus other info (e.g. experimental data)
- See <https://salilab.org/nup133> for a recent published system, also linked from the `python-ihm` docs

Usage

3. Input



- Currently basic support for reading IHM mmCIF files and populating the data model (in development) - e.g. for visualization or using IHM models as input for further modeling

Availability

- Open source at GitHub:
<https://github.com/ihtmwg/python-ihm>
- Open source, permissive (MIT) license
- Pure Python, only requires Python standard library
- Python 2 and Python 3 compatible
- Contributions welcome