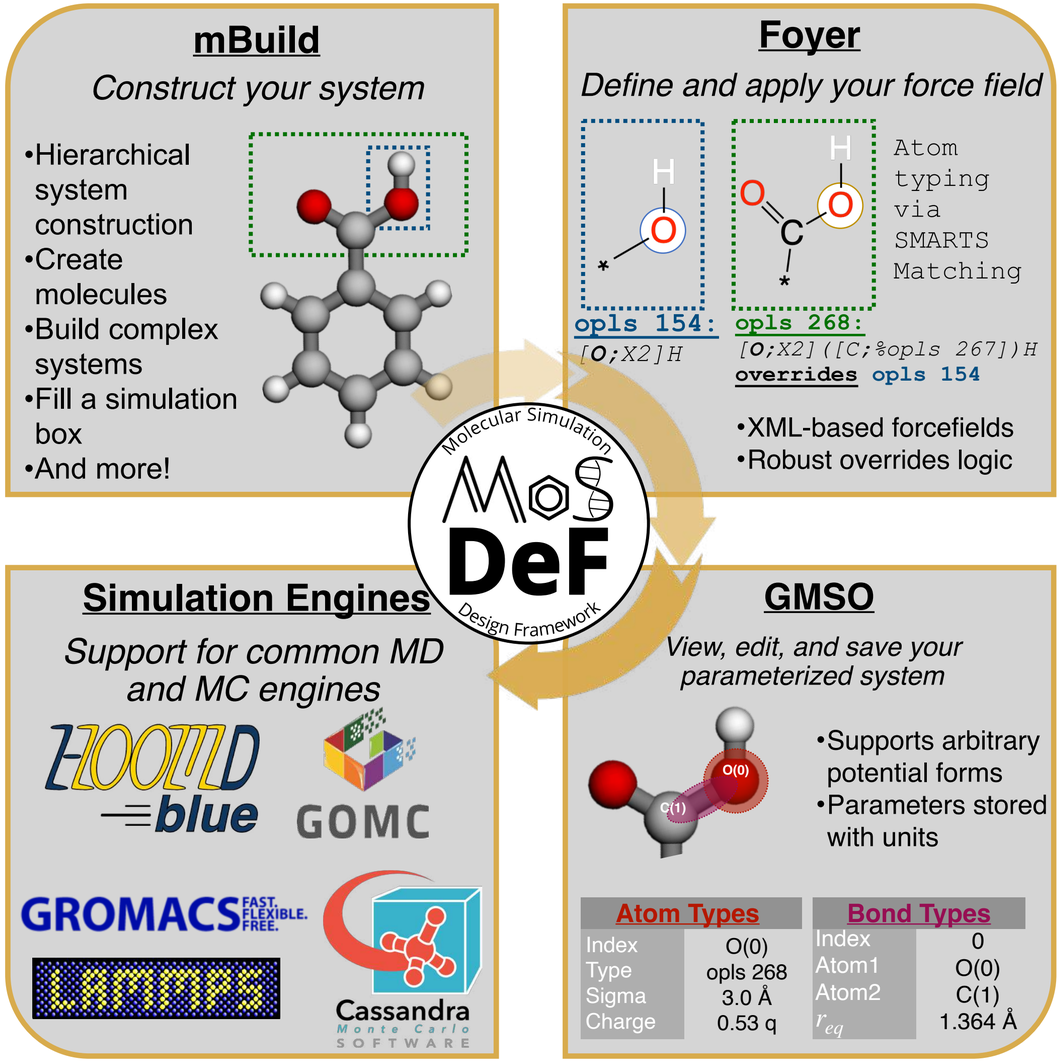
# The Molecular Simulation Design Framework (MoSDeF)

- MoSDeF is a collection of Python libraries to assist in the system initialization stage of molecular simulation in a reproducible manner
  - Developed with best practices of OSS dev
    - Test driven development, version control,
- **mBuild**[1] - A generic molecule builder
  - Build systems up from smaller re-usable components
  - Plug-in system for users to share and extend complex systems
- **Foyer**[2] - Apply interaction parameters in a simulation engine-agnostic manner
  - Extends OpenMM forcefield XMLs
  - Plug-in system for users to share forcefields
- **GMSO** - Molecular topology information
  - Replacing ParmEd
  - Interaction parameters and functional forms



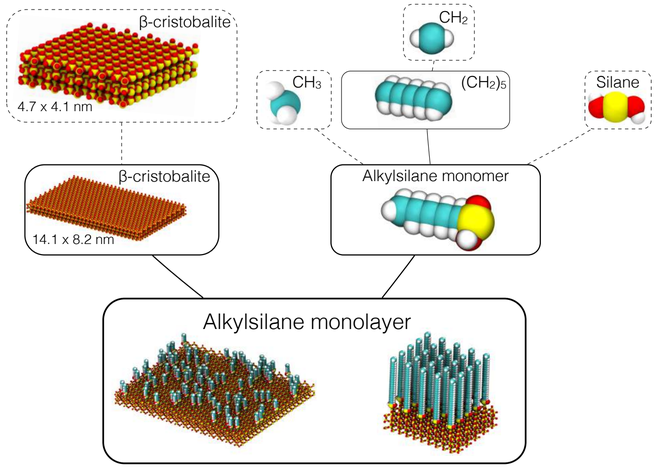


1. *Klein C, Sallai J, Jones TJ, Iacovella, CR, McCabe C, Cummings PT. Foundations of Molecular Modeling and Simulation. 2016*
2. *Klein C, Summers AZ, Thompson MW, Gilmer JB, McCabe C, Cummings PT, Sallai J, Iacovella CR. Computational Materials Science. 167, 2019*

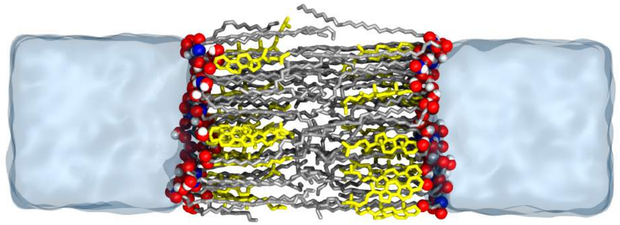
VANDERBILT

# mBuild: A Hierarchical, Component-based Molecule Builder

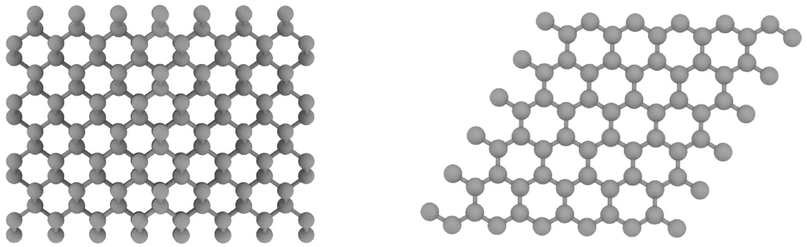
- ❑ Connect base components together like legos
  - o Import starting structures from a file
  - o Build from SMILES strings
- ❑ Design systems with TRUE in mind
  - o Expose tunable parameters
    - Polymer length, unit cell repeats, terminal groups, polymer tethers, etc.
  - o Enables further automation of system initialization
- ❑ Share as python scripts or installable plug-ins



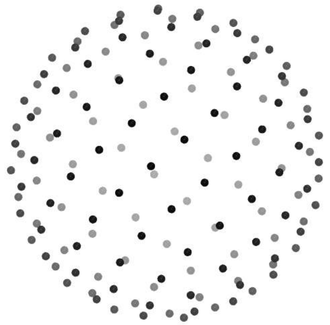

**Complex polymer coated monolayer systems**

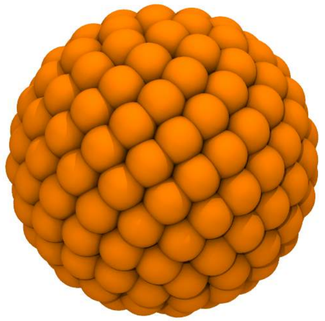


**Lipid Bilayers**

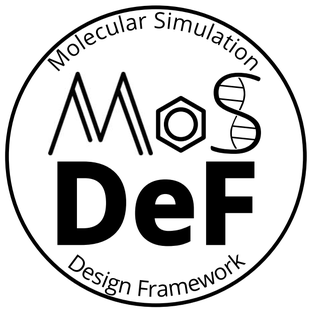


**Create crystalline systems manually or from CIF files**



**Nanoparticles**

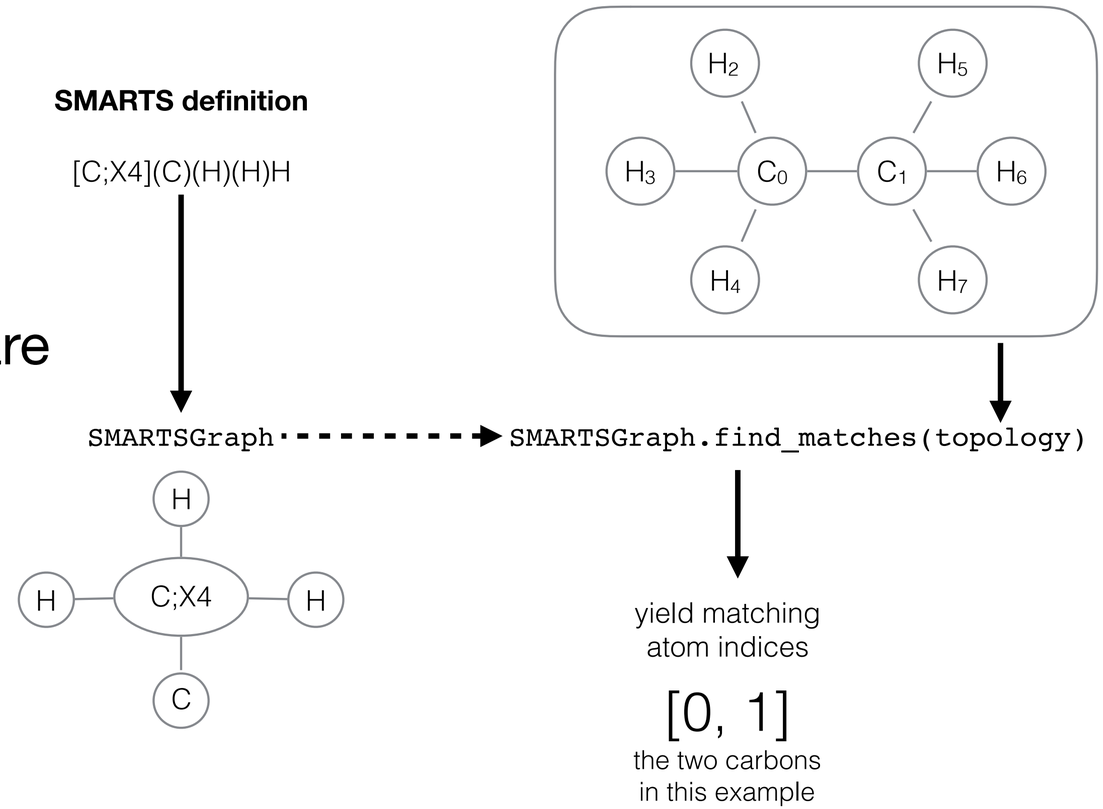



**mbuild.mosdef.org**

MUMS

# Foyer: A Package for Atomtyping and Disseminating Forcefields

- Operates on graph structures
  - Use SMARTS strings to define the atomtypes
- Currently supports traditional atom-type based forcefields
  - OPLS-AA, GAFF, TraPPE-UA (ongoing), etc.
- Users can create their own forcefield files and share with the community
  - As files or plug-ins




**Example of the SMARTS grammar and workflow that foyer uses to determine matches**

foyer.mosdef.org

*Klein C, Summers AZ, Thompson MW, Gilmer JB, McCabe C, Cummings PT, Sallai J, Iacovella CR. Computational Materials Science. 167, 2019*

VANDERBILT

# Foyer Forcefield XML

**SMARTS string defining the local chemical neighborhood of each atom type**

**DOI of parameter source, will provide bibtex file at end of parametrization. Listing what atom types came from each source**

**Overrides statement to set rule precedence**

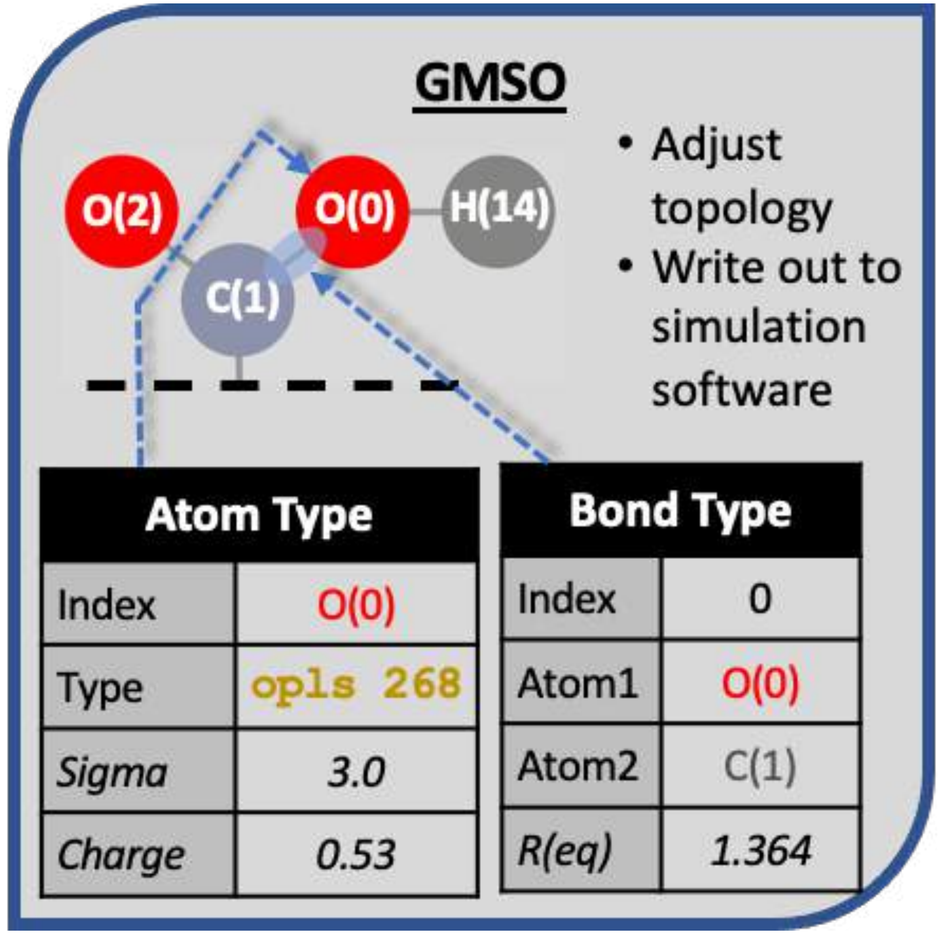
```
<ForceField>
 <AtomTypes>
 <Type name="opls_135" class="CT" element="C" mass="12.01100"
       def="[C;X4](C)(H)(H)H" desc="alkane CH3"
       doi="10.1021/ja9621760"/>
 <Type name="opls_148" class="CT" element="C" mass="12.01100"
       def="[C;X4]([C;%opls_145])(H)(H)H"
       desc="toluene CH3" overrides="opls_135"
       doi="10.1021/ja9621760"/>
 <Type name="opls_145" class="CA" element="C" mass="12.01100"
       def="[C;X3;r6]1[C;X3;r6][C;X3;r6][C;X3;r6][C;X3;r6][C;X3;r6]1"
       desc="aromatic C in 6-membered ring"
       overrides="opls_141,opls_142" doi="10.1021.ja9621760"/>
 </AtomTypes>
</ForceField>
```

MUMS

VANDERBILT

# GMSO: General Molecular Simulation Object

- Before GMSO, foyer and mBuild relied heavily on ParmEd
  - ParmEd has some limitations in functional form representation, etc.
    - Ex. only 12-6 LJ for non-bonded
- GMSO is an attempt to address some of these limitations
  - Let users define their functional forms using symbolic mathematics
  - If an engine supports that functional form, MoSDeF can support that engine
  - Add support for additional engines with minimal effort
- First phase of this transition already in place and being tested




Molecular Simulation MoSDeF Design Framework

gmso.mosdef.org

MUMS

VANDERBILT

# GMSO-Extended Forcefield XML

Default units when not listed below are defined here

Symbolic mathematics to represent functional forms for each type of interaction (non bonded, bonds, angles, etc.)

Each parameter in the expression can have per-parameter values that override the default units set above.

SMARTS definitions and all other Foyer-XML features like doi's, overrides are supported.

Every interaction type has a similar format

```
<?xml version="1.0" encoding="UTF-8"?>
<ForceField name="ethylene_snippet" version="0.0.1">
  <FFMetaData>
    <Units energy="kcal/mol" mass="amu" charge="elementary_charge" distance="nm" />
  </FFMetaData>
  <AtomTypes expression="4 * epsilon * ((sigma/r)**12 - (sigma/r)**6)">
    <ParametersUnitDef parameter="epsilon" unit="kJ/mol" />
    <ParametersUnitDef parameter="sigma" unit="nm" />
    <AtomType name="opls_143" atomclass="CM" element="C" \\
      charge="-0.23" mass="12.01078" definition="[C;X3](C)(H)H">
      <Parameters>
        <Parameter name="epsilon" value="0.355" />
        <Parameter name="sigma" value="0.317984" />
      </Parameters>
    </AtomType>
    <AtomType name="opls_144" atomclass="HC" element="H" charge="0.115"\\
      mass="1.007947" definition="[H][C;X3]">
      <Parameters>
        <Parameter name="epsilon" value="0.242" />
        <Parameter name="sigma" value="0.12552" />
      </Parameters>
    </AtomType>
  </AtomTypes>
  <BondTypes expression="0.5 * k * (r - r_eq)**2">
    <ParametersUnitDef parameter="k" unit="kJ/mol/nm**2" />
```

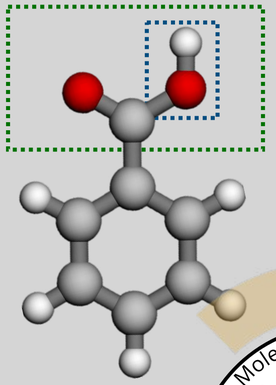
MUMS

VANDERBILT

## mBuild

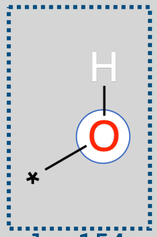
*Construct your system*

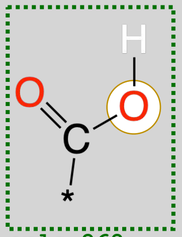
- Hierarchical system construction
- Create molecules
- Build complex systems
- Fill a simulation box
- And more!



## Foyer

*Define and apply your force field*




Atom typing via SMARTS Matching

```
opls 154:
[O;X2]H
```

```
opls 268:
[O;X2]([C;%opls 267])H
overrides opls 154
```

- XML-based forcefields
- Robust overrides logic

Molecular Simulation MoSDeF Design Framework

## Simulation Engines

*Support for common MD and MC engines*

HOOMD-blue

GOMC

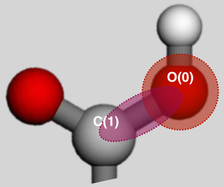
GROMACS FAST. FLEXIBLE. FREE.

LAMMPS

Cassandra Monte Carlo SOFTWARE

## GMSO

*View, edit, and save your parameterized system*




- Supports arbitrary potential forms
- Parameters stored with units

| Atom Types | |
|---|---|
| Index | O(0) |
| Type | opls 268 |
| Sigma | 3.0 Å |
| Charge | 0.53 q |

| Bond Types | |
|---|---|
| Index | 0 |
| Atom1 | O(0) |
| Atom2 | C(1) |
| $r_{eq}$ | 1.364 Å |

+

signac

signac.io

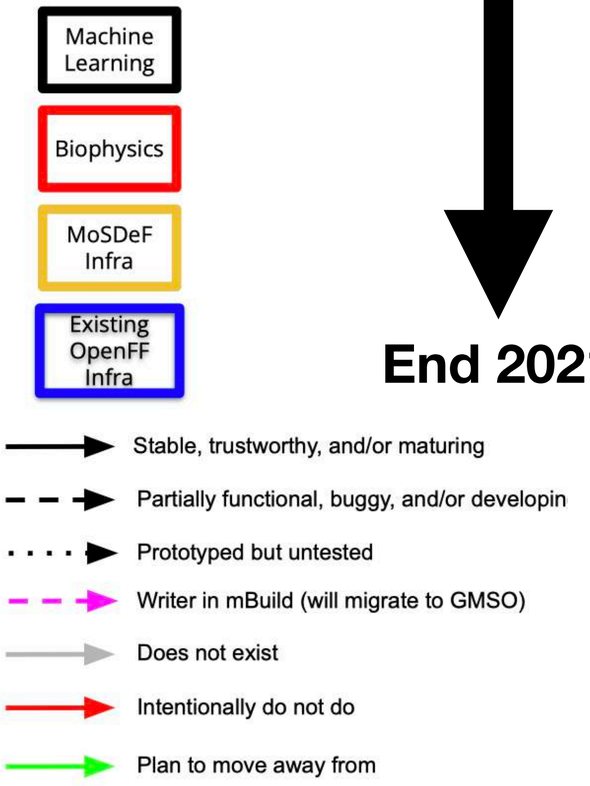
**Manage and automate this entire workflow using MoSDeF-affiliated Signac Framework**

MUMS

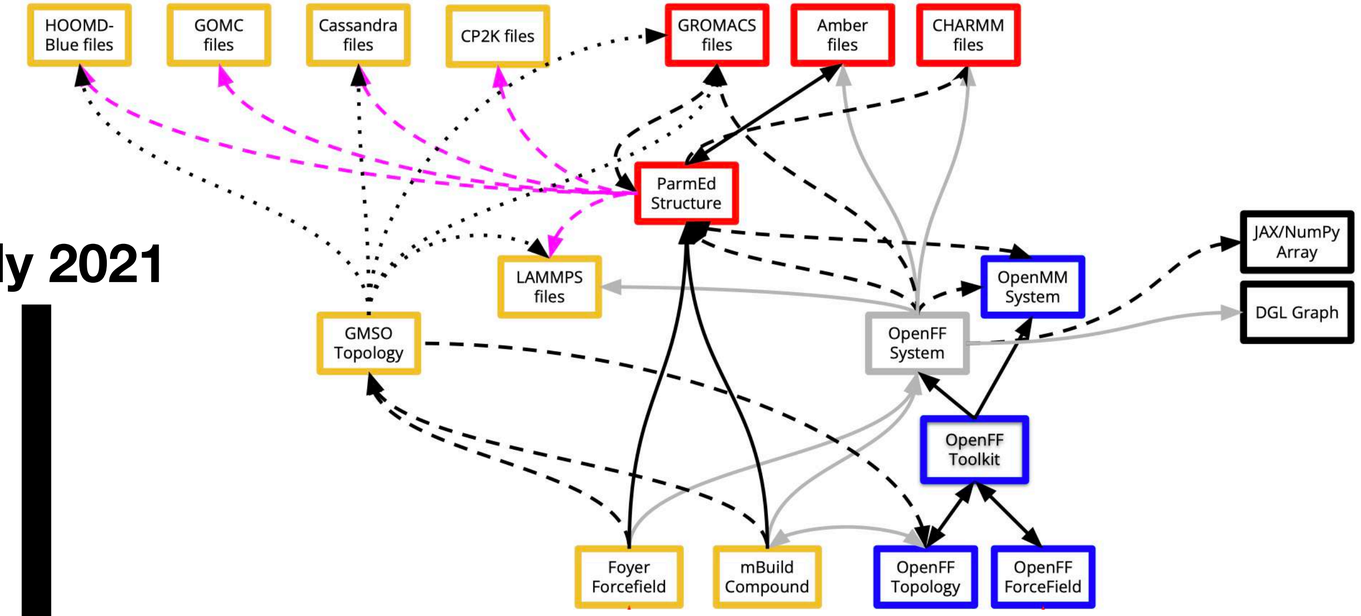
VANDERBILT

# MoSDeF + OpenFF Collaboration

- NIH Funding to provide interoperation between the OpenFF collaboration and MoSDeF
- Can use Foyer or OpenFF Parsley Forcefield to parametrize a system, write out to a simulation engine of choice
- Further integration underway




**Early 2021**




**End 2021**