Have an API

- Have a well-documented API aimed at people using your platform
- Ideally, have bindings to high-level languages (e.g., Python) to facilitate integration with campaign-scale tools in those ecosystems.
  - Running OpenMM: OPS just uses the OpenMM Python API
  - Running GROMACS: OPS launches a gmx subprocess and has to monitor as it writes to a file, and then adapt that into something that works with the simpler API approach.
- Be careful that your bindings give expected behavior in the target language

```
from [redacted] import Chem
mol = Chem.MolFromSmiles("CC")
atom = mol.GetAtoms()[0]
atom.GetOwningMol() == mol
# False
```

Enable object serialization

- Campaign-scale software will need to turn a description of your simulation into bytes, either to document how it was run or to be able to launch multiple
  - if we're using your CLI to launch, then your input files might work
  - if we're using some object component of yours, then we need it to be serializable
- Consider two types of objects:
  - Data objects: The data you're working with. Each object has a predetermined pattern, and there may be many instances of them.
  - Simulation objects: The objects that describe your simulation. These are varied in contents, but there are fewer of them.
- Both of these should be serializable