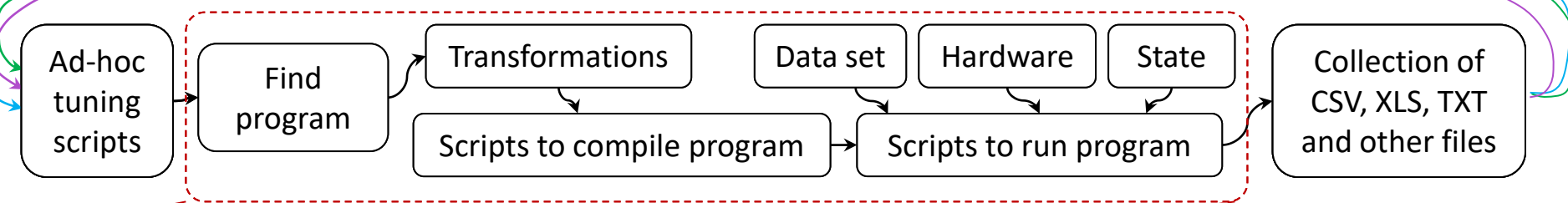
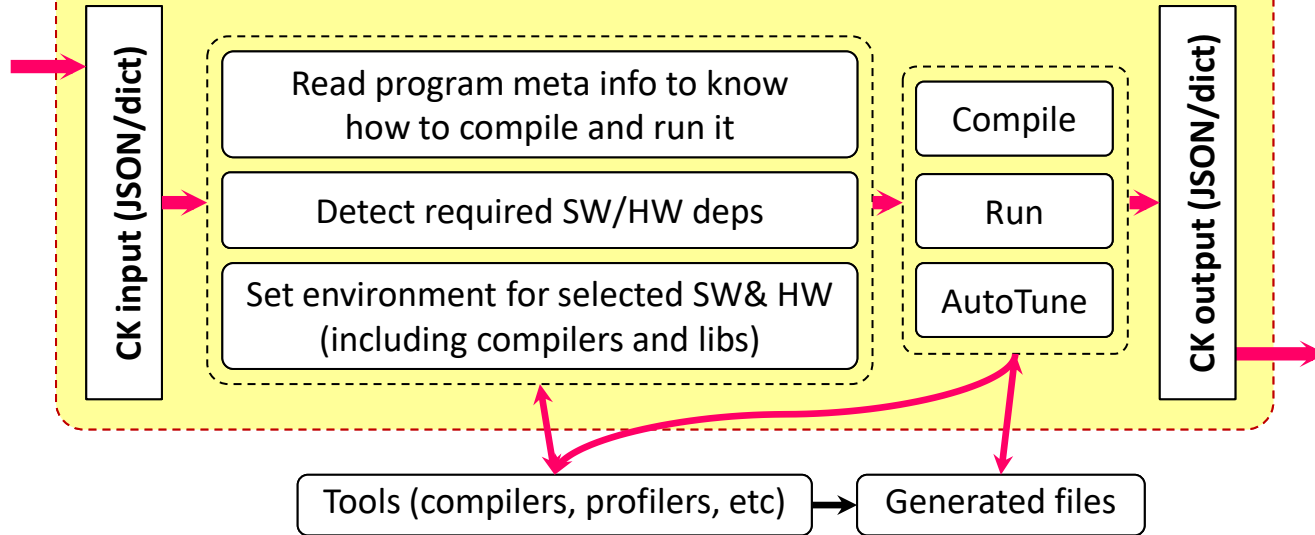


Typical experimental workflow



CK Python module (wrapper) with a unified JSON API



Example of CK API (**program module**)

```
import ck.kernel as ck
```

```
def compile(i)
    # Process CMD
    prog=i.get('data_uoa',"")
    flags=i.get('flags',"")
    speed=i.get('speed',"")
```

```
# Get program meta and path
r=ck.access({'action':'load',
             'module_uoa':'program',
             'data_uoa':prog})
if r['return']>0: return r
meta=r['dict']
path=r['path']
...
return {'return':0}
```

```
# ck – simple command line front-end to manage CK repositories,
#   call CK modules, abstract tools and unify input/output
ck compile program:cbench-automotive-susan --speed --flags=-O3 ...
ck run program:cbench-automotive-susan
ck autotune program:cbench-automotive-susan
```