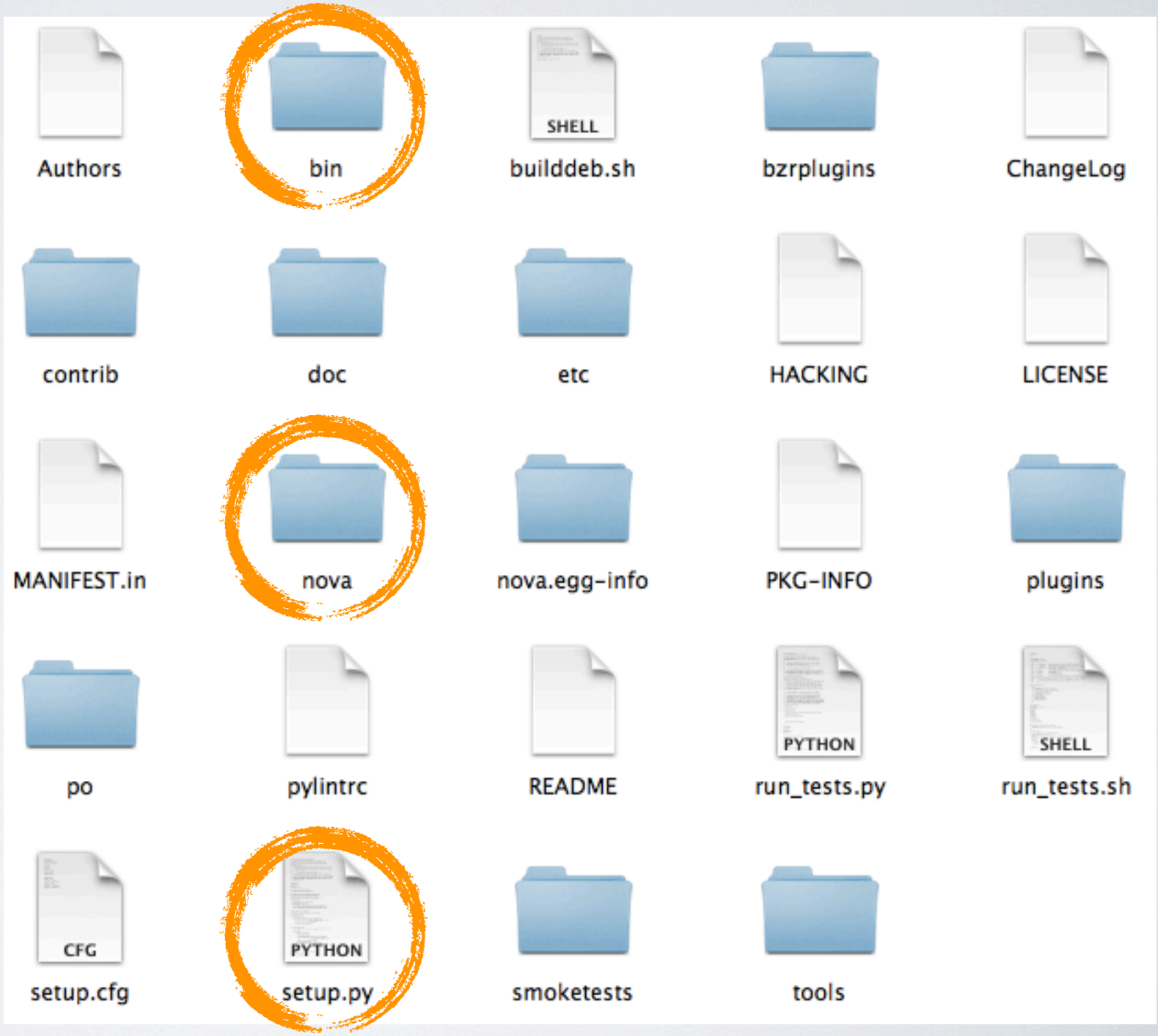
A FIRST STEP INTO NOVA

by can.

NOVA 2011.3 DIABLO/

- setup.py：安装脚本
- bin：可执行程序，比如nova-manage等
- nova：核心文件



/SETUP.PY

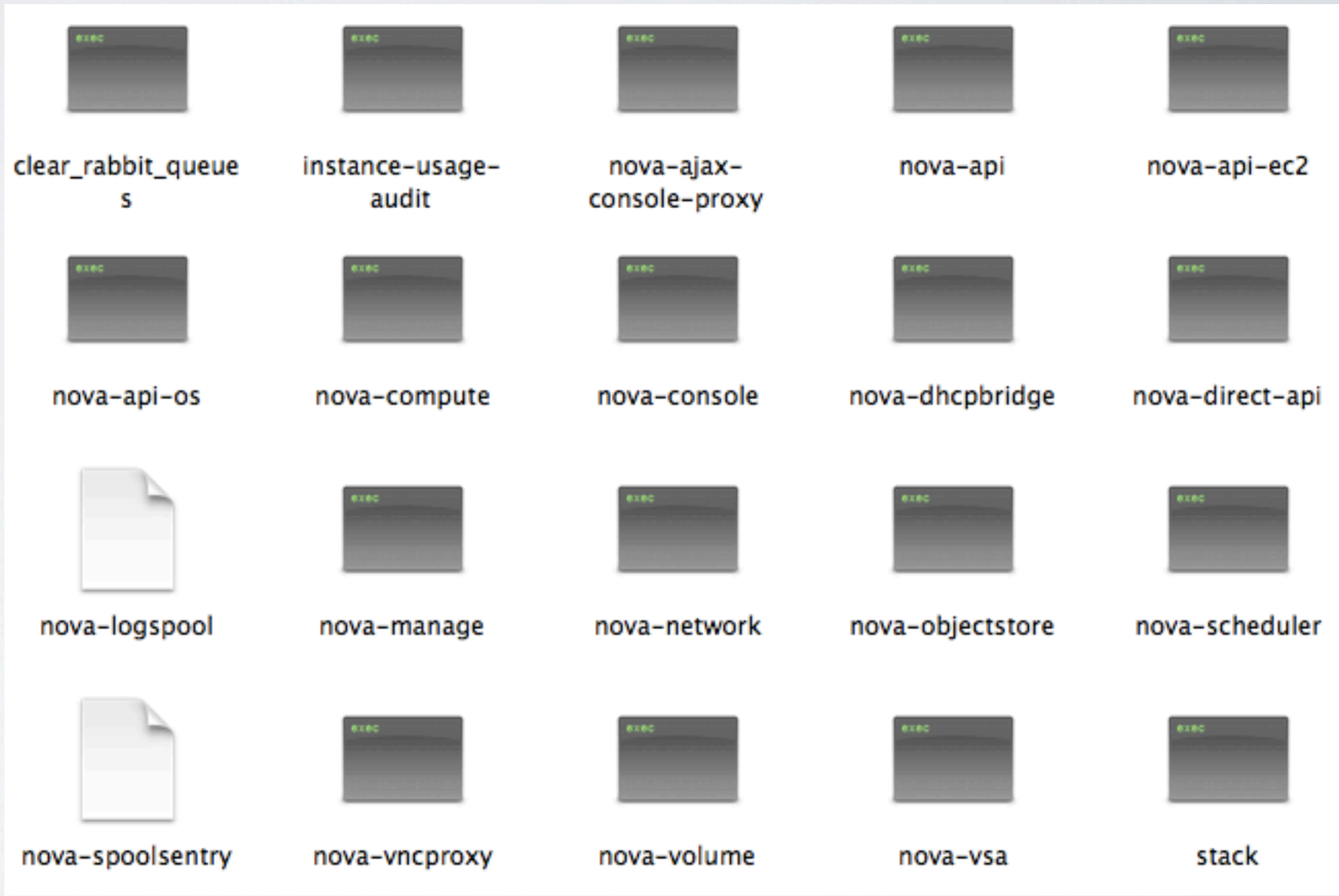
```
try:
    from DistUtilsExtra.auto import setup
except ImportError:
    from setuptools import setup
...

setup(name='nova',
      version=version.canonical_version_string(),
      description='cloud computing fabric controller',
      author='OpenStack',
      author_email='nova@lists.launchpad.net',
      url='http://www.openstack.org/',
      cmdclass=nova_cmdclass,
      packages=find_packages(exclude=['bin', 'smoketests']),
      include_package_data=True,
      test_suite='nose.collector',
      data_files=find_data_files('share/nova', 'tools'),
      scripts=['bin/nova-ajax-console-proxy',
               'bin/nova-api',
               'bin/nova-compute',
...
```

/BIN

- 其实都是python脚本

- nova-api/compute/network/scheduler等是服务的启动程序

- nova-manage是给管理员的命令行程序



/BIN/NOVA-API

```
...

    for api in flags.FLAGS.enabled_apis:
        servers.append(service.WSGIService(api))
    service.serve(*servers)
    service.wait()

...
```

/BIN/NOVA-COMPUTE

```
...

    server = service.Service.create(binary='nova-compute')
    service.serve(server)
    service.wait()

...
```

NOVA的SERVICE

- 分为两种，Service和WSGIService
- 前者供内部调用，通过RPC(remote procedure call)模块
- 后者对外提供API，通过http

神奇的FLAGS

- 整个项目统一采用flags模块处理可由用户定义的参数，包括配置文件、通过命令行传入的参数等

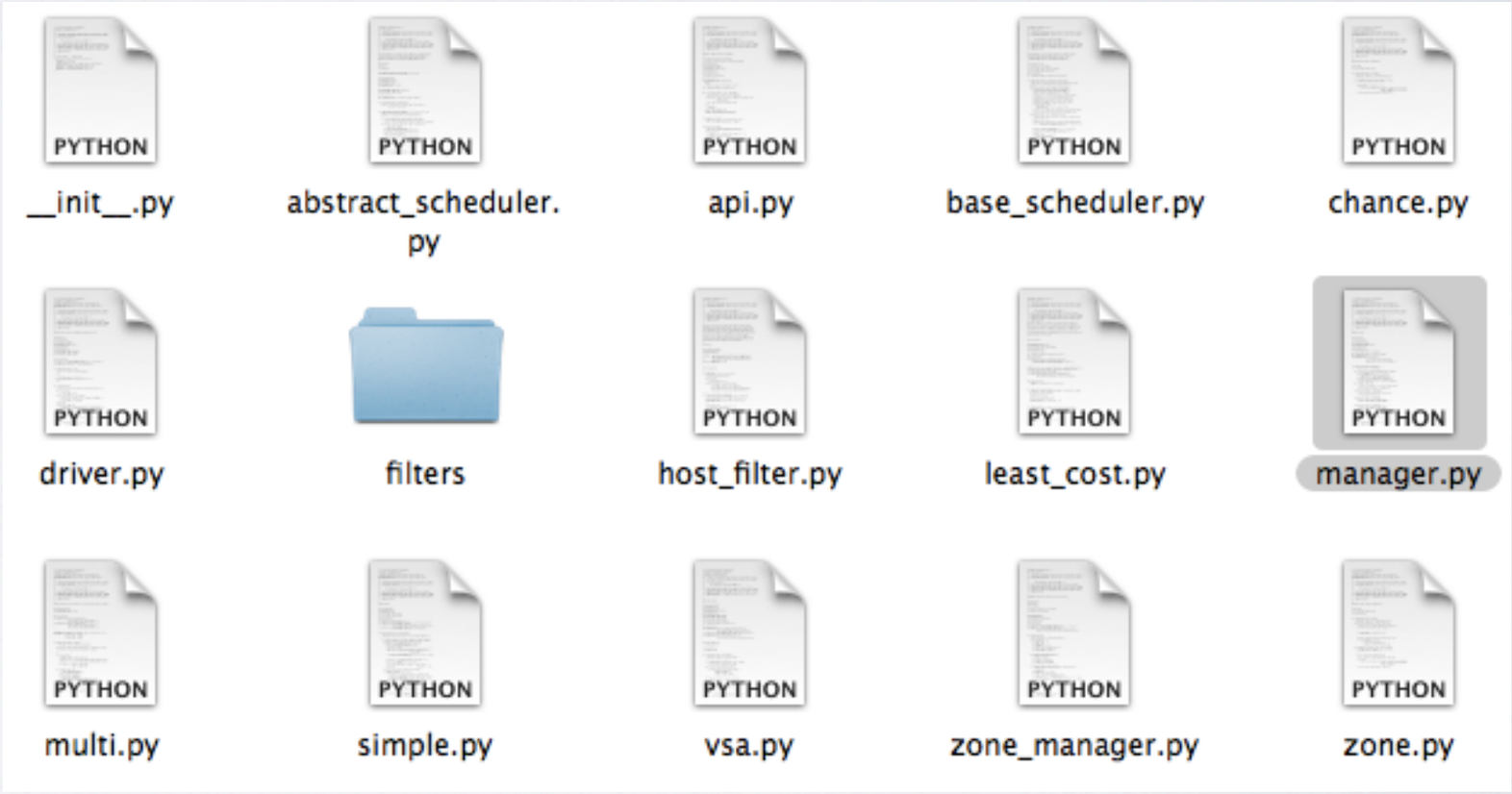
- 基于开源代码python-gflags

- 可以分布式地定义参数，即每个模块只需定义自己需要的，通过"import"引入其它模块定义的参数。有利于模块化

```
# from /nova/service.py
flags.DEFINE_string('ec2_listen', "0.0.0.0",
                    'IP address for EC2 API to listen')
flags.DEFINE_integer('ec2_listen_port', 8773, 'port for ec2 api to listen')
flags.DEFINE_string('osapi_listen', "0.0.0.0",
                    'IP address for OpenStack API to listen')
flags.DEFINE_integer('osapi_listen_port', 8774, 'port for os api to listen')
```

/NOVA/SCHEDULER

- 目录下面有”__init__.py”说明这是一个模块
- manager统一对外的接口
- 通过dirver兼容不同的底层结构



SCHEDULERS

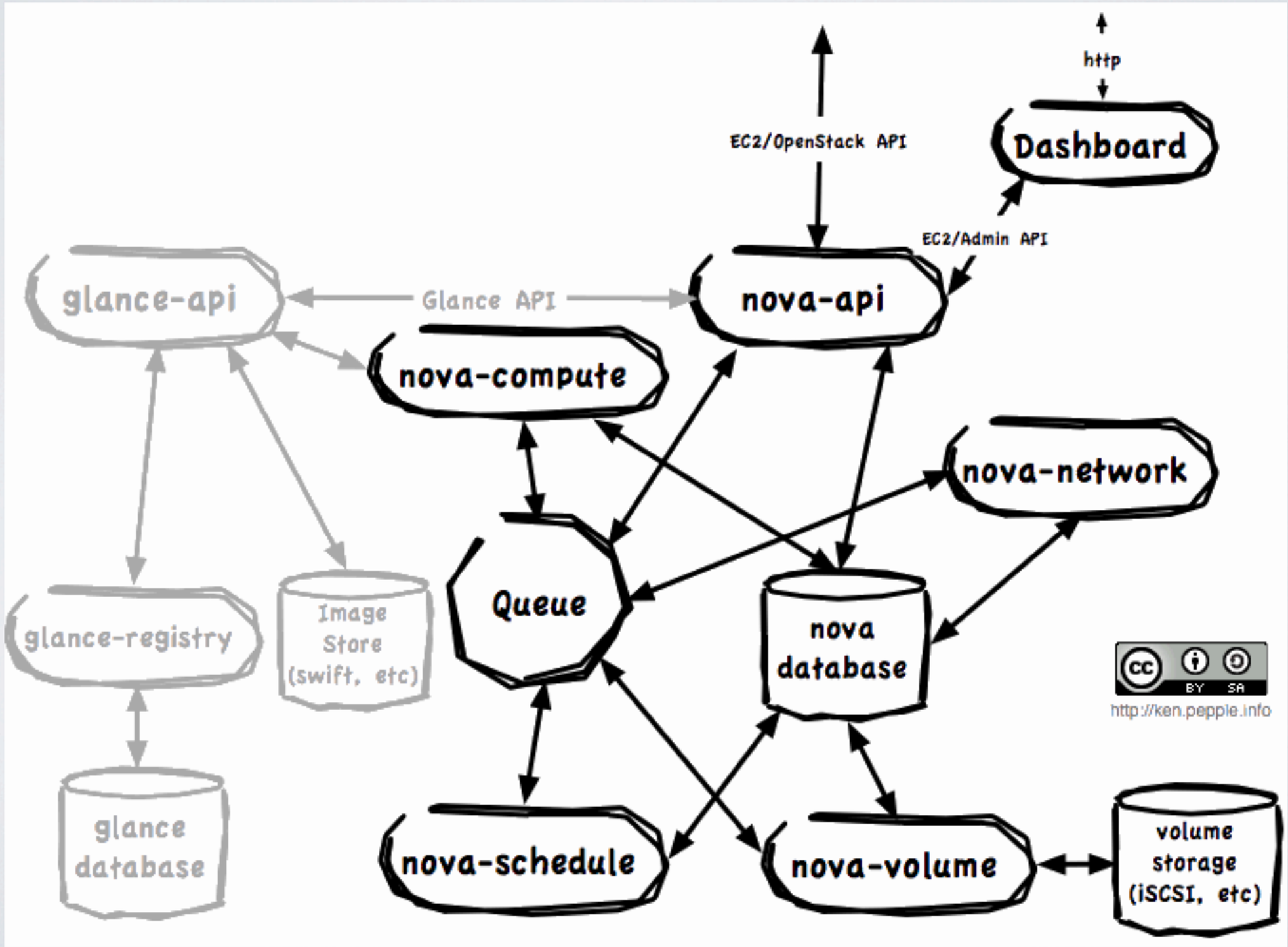
nova.scheduler.driver.Scheduler
ChanceScheduler
SimpleScheduler
VsaScheduler
VsaSchedulerLeastUsedHost
VsaSchedulerMostAvailCapacity
AbstractScheduler
BaseScheduler
LeastCostScheduler
MultiScheduler
ZoneScheduler

自定义调度算法

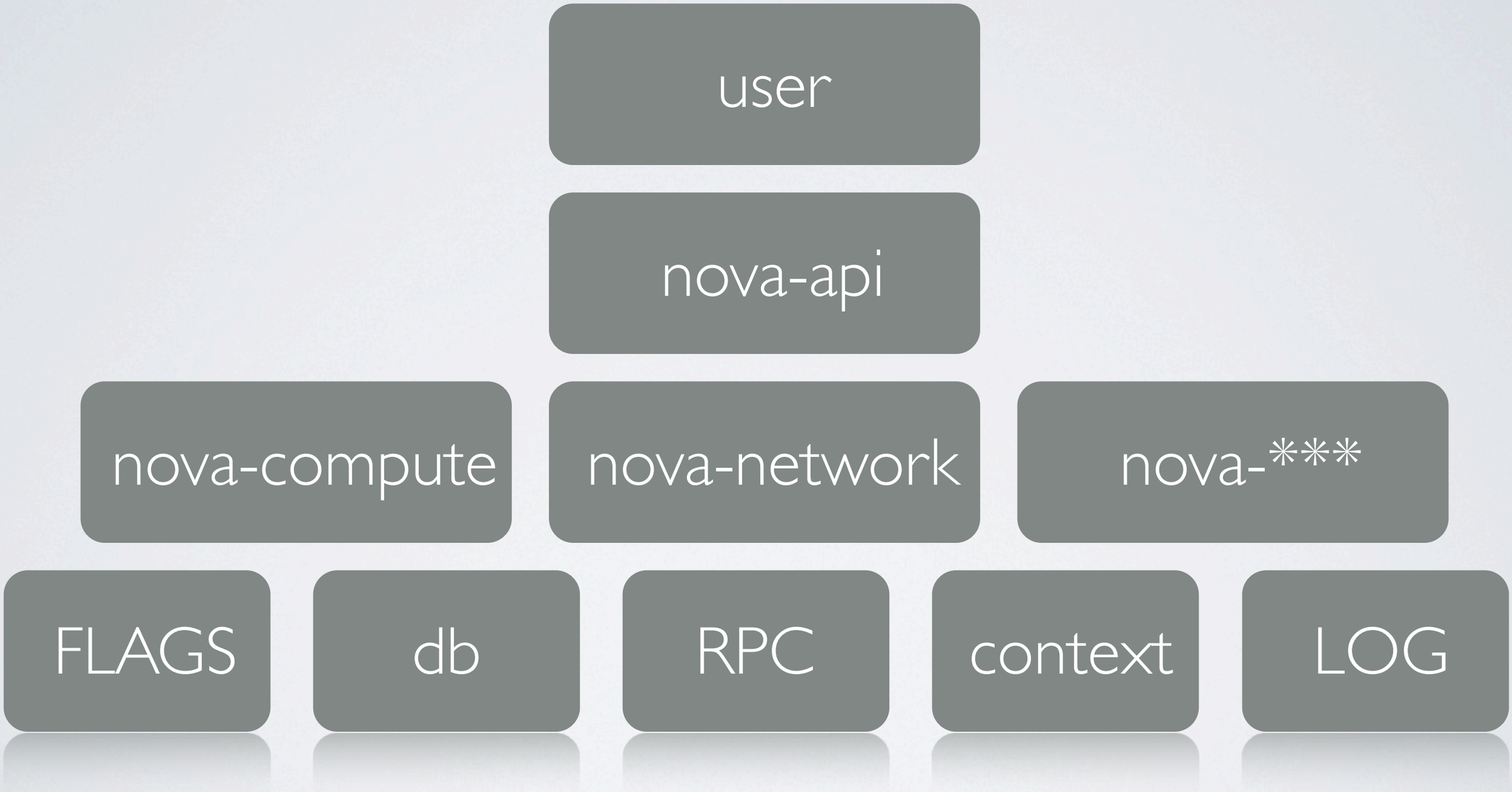
```
# from /nova/scheduler/chance.py
class ChanceScheduler(driver.Scheduler):
    """Implements Scheduler as a random node selector."""

    def schedule(self, context, topic, *_args, **_kwargs):
        """Picks a host that is up at random."""

        hosts = self.hosts_up(context, topic)
        if not hosts:
            raise driver.NoValidHost(_("Scheduler was unable to locate a host"
                                       " for this request. Is the appropriate"
                                       " service running?"))
        return hosts[int(random.random() * len(hosts))]
```

外界通过manager调用schedule_**()方法，若所需方法不存在，使用schedule()

逻辑结构


http
Dashboard
EC2/OpenStack API
EC2/Admin API
glance-api
Glance API
nova-api
nova-compute
nova-network
Queue
nova database
glance-registry
Image Store (swift, etc)
glance database
nova-schedule
nova-volume
volume storage (iSCSI, etc)
http://ken.pepple.info

“物理”结构?


user
nova-api
nova-compute
nova-network
nova-***
FLAGS
db
RPC
context
LOG