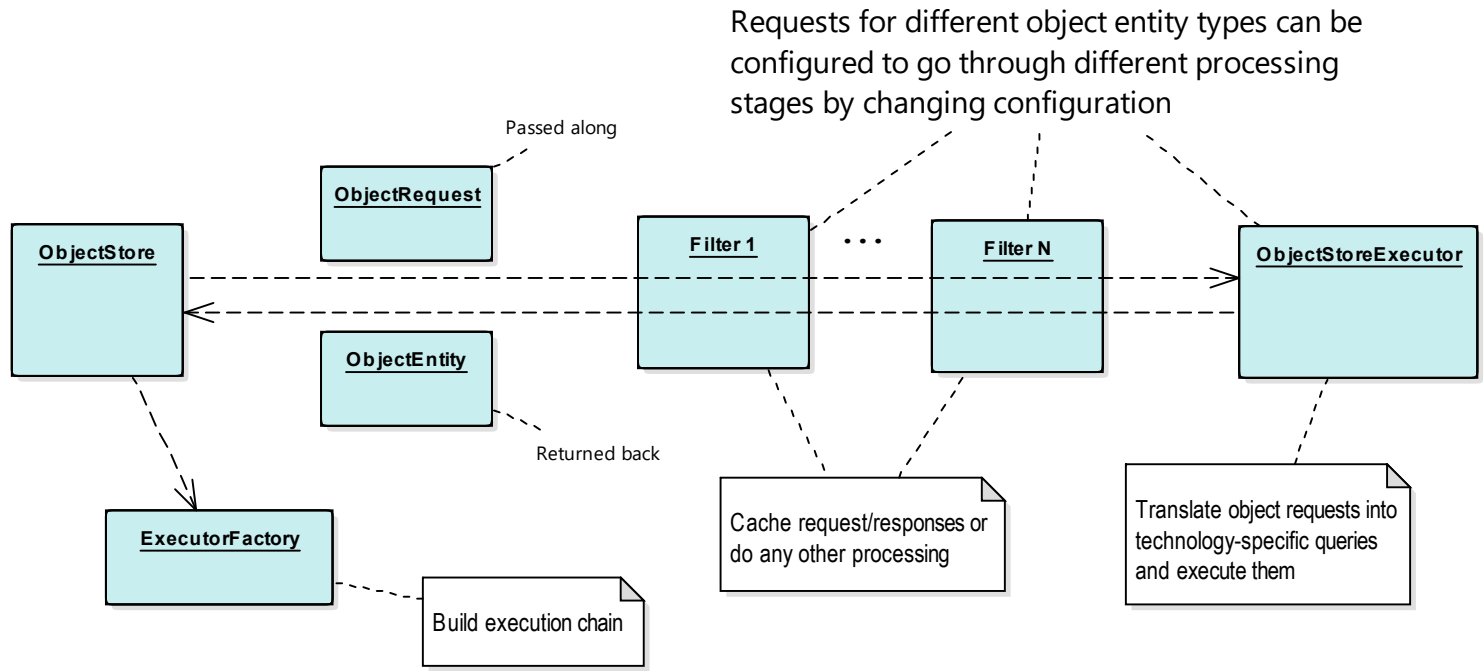


## ePricer object store interface design

The solution must be

- performant: may use caching under the hood
- user-friendly: simple interfaces and strongly typed results
- technology-neutral: can be backed by different data sources, ex. Solr, Redis, DB2, etc
- flexible: may use custom processing logic transparently, ex. multiple data sources



### Interface usage example

```
@Autowired
ObjectStore store;

Criteria criteria = Criteria
    .where("contenttype").is("CYctmporf.json")
    .and("orf_fk_tablecode").is("C");

List<OrfEntity> orfList = store.retrieve(OrfEntity.class).where(criteria).list();
```

### Object entity definition

```
class OrfEntity extends ObjectEntity {
    public String orf_activetransactionq;
    public String orf_ctmporfid;
    public String orf_description;
    public Nls[] nls_list;
}

class Nls extends ObjectEntity {
    public String nls_description;
    public String nls_languagecode;
}
```