核心库包括以下几个方面

### 1、ORM框架

当前系统采用了ORM框架（对象关系映射）进行数据操作，实体映射部分（Song.Entities项目）是开源的，不影响二次开发。

如果数据库新增或修改表和字段后，用代码生成器（WeiSha.Data.Generete.exe）重新生成 Song.Entities 项目的实体类。

代码生成器（WeiSha.Data.Generete.exe）可以在"/Sourcecode/Lib"文件夹获取到。

### **2、IOC控制**

IOC全称是Inversion of Control（控制反转）。是常用的降低耦合的方法之一。接口与实现方法的对应关系可以在web.config中配置。

如果需要新增或调整接口，可以通过该配置处理。

### **3、模板引擎**

当前系统是由.Net MVC架构改造而成的MVVM架构，剔除了Razor（微软官方提供的模板引擎），采用了自主编写的模板引擎，在满足自身业务需求的同时，更轻量、运行更快。路由直接映射到模板库（/templates/文件夹）中的模板文件，而模板文件全部采用缓存处理，大幅减少IO操作。

前端文件都在模板库，采用Vue作为前端框架。后端接口全部在Song.ViewData项目（开源），二次开发不会涉及核心库的内容。

### **4、常用方法**

一些常用的类库和方法，包括加密、解密、文件压缩、图形处理、安全校验、地理信息计算、数据类型转换、日志管理、缓存管理等等；

### **5、授权管理**

用于识别商业版本的授权信息。

**综述**

像ORM、IOC、模板引擎等，一般都是被封装的DLL(动态链接库)。要么由官方提供，要么是第三方公司提供,通过NuGet引入。在我们的产品中，只不过这些东西我们自己提供罢了。因为涉及框架底层，封装成DLL是必要的。