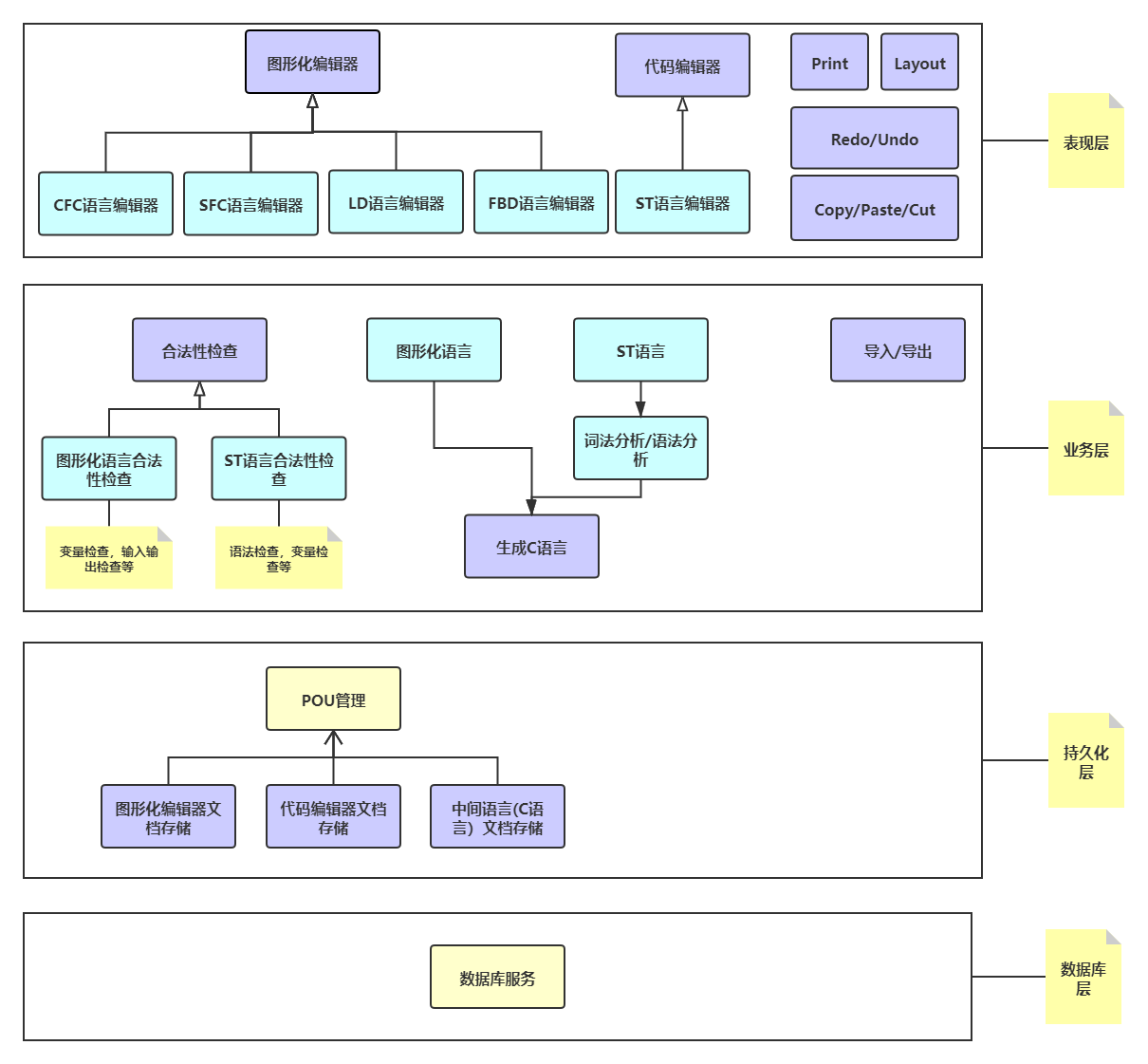
VIEC相关问题

# 概述

包含内容：

* ST
* CFC
* SFC
* LD
* FBD

# 分层架构图

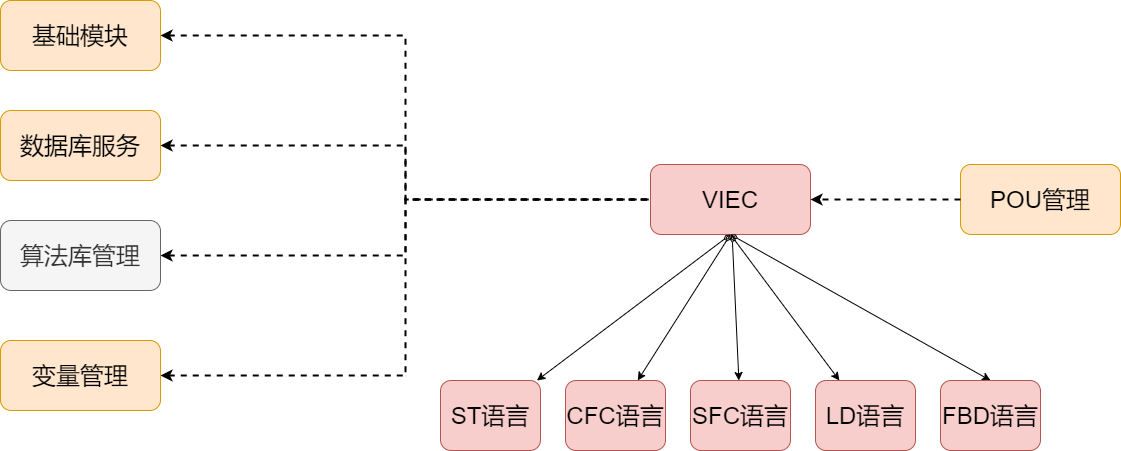


# 设计原则

**界面/业务/数据分离，在不需要界面层参与的情况下，可以进行单元测试和自动化测试**。

原有的代码中，POU中包含了POU相关的代码，也包含了HCC相关的代码；图形语言相关的元件中，既包含UI相关的代码，也包含数据相关的代码，更包含HCC相关的代码。

# 组件依赖关系



~~（在线模块是否被VIEC依赖？）~~

数据库服务：数据的增删改查

变量模块：新建POU/全局变量、变量的引用

算法库模块：IEC编译需要算法块提供编译依赖（obj、.h等）

POU管理：IEC引用，不依赖，如：全部POU下的变量引用

# 流程简述



## 打开POU

UI 双击POU名称，打开组态界面/代码编辑器界面

若为打开POU：

UI调VIEC加载数据，基于加载的数据，初始化为UI组态数据

（注：UI需要对VIEC加载的数据处理为界面显示数据），

## 新建POU

UI 双击POU名称，打开组态界面/代码编辑器界面

通过算法库管理等进行组态 或 编写代码，过程中涉及到业务相关调用VIEC接口（这个过程会频繁的调用VIEC的处理），编辑完成，保存编译或保存入库调用VIEC的功能

## 拖放元件

1. 循路算法，UI实现
2. 拖放元件，引脚能否关联，VIEC提供接口是否允许连接，若允许 ，由UI实现关联

## 删除元件

UI行为，VIEC不涉及，保存时，调用VIEC的保存

## 新建变量/变量引用

新建变量，设计UI和变量模块，VIEC不涉及（允许的变量类型是否涉及VIEC）

变量引用，同新建变量

# 相关问题讨论

## 业务相关问题

### 生成代码

1. 变量生成代码的原则

* LOCAL变量生成名称
* GLOBAL变量需要使用地址
* ~~FUNCTION\_BLOCK生成对应的结构体~~**~~(ST语言的FUNCTION\_BLOCK和C语言的void function差别比较大，相当于结构体+函数，这点需要注意)~~**

1. C/C++，使用指针还是引用？
2. 生成代码时是否注意代码风格（**建议启用**）
3. ST语言相关类型/函数使用**运行库**进行封装
4. M7 ST语言支持范围同M6 ST语言，保持一致，ST语言的词法分析和语法分析沿用M6 ST语言
5. SFC语言需要最后实现（SFC语言的Action可以是其它三种语言，存在依赖性）

### 合法性检查

沿用M6的实现，返回的错误信息的内容可能需要进一步处理。

## 界面相关问题

### 之前提到的优化内容

1. 缩放后的界面，以选中/或者中心为中心点进行显示
2. 打印的页脚部分可以进行设计
3. 隐藏未连接引脚并且自动布局

### 其它优化内容

1. 实现的效果和目前一致，还是

### 新增需求

1. 调用树功能 – 按任务显示，且在多任务中的都需显示
2. 变量引用信息(需要定位到代码行，或者元件) – 基于任务查询（单个/所有POU的变量引用）
3. 变量高亮显示 – ~~代码编辑器中/~~当前POU中所有该变量高亮
4. POU添加安全版本和非安版本
5. ~~代码统计(ST语言)~~
6. ST/CFC/LD/SFC/FBD：添加POU规格信息，如：代码行数、变量信息、制约信息，

触发方式：右键菜单（POU信息） 显示区域：主窗口的信息输出窗口

1. CFC：窗格栅可调
2. 新增需求优先级

## 其它问题

### 存储格式的设计

1. IEC标准，IEC61131-3？
2. 自定义格式/标准格式？

格式1：现有文件数据格式，全部以数据方式存储数据库

格式2：类似HoneyWell格式，输入、BOX、输出等分别描述，POU主描述组态逻辑等

1. 同时支持数据库存储和文件存储

单元测试时，本地可以使用文件方式，发布版本使用数据库存储