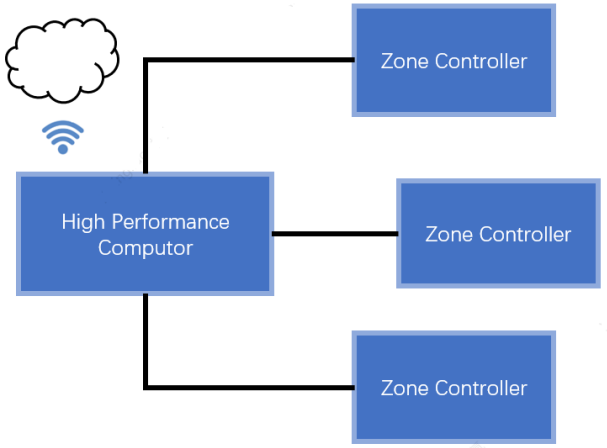
ZEEKR

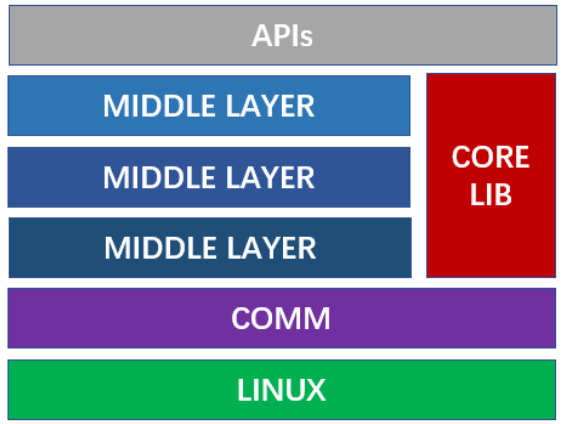
基于模型开发车载OS环境下的SOA应用软件

罗一鸣　极氪智能科技

MATLAB EXPO

背景



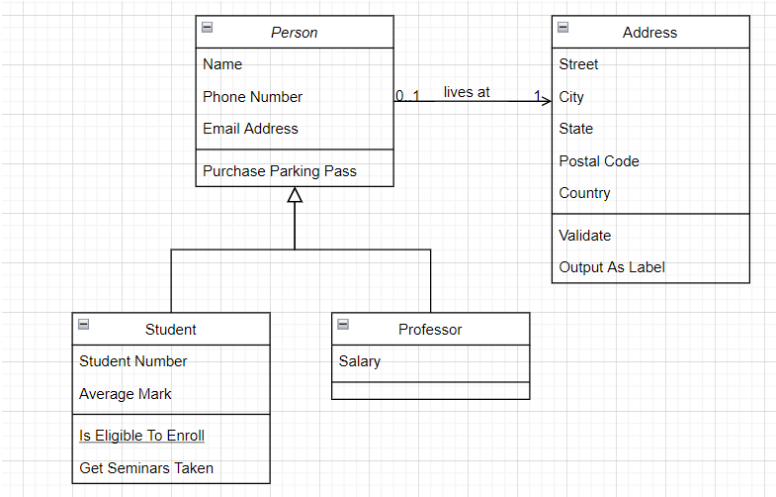


- SOA行业趋势
  - 软件定义汽车
  - 逐渐增长的功能需求
- 下一代电子电气架构
  - 域集成
  - 为软件持续增长提供硬件支持
- 车载OS
  - OEM自研的分布式操作系统
  - 对SOA提供了有力支持

挑战

- 代码复杂
  - 手写C++对开发人员的技能水平以及配套的工具链设施提出了很高的要求

```
55      *  derived from basic_streambuf to do the actual output.
56      */
57     template<typename _CharT, typename _Traits>
58       class basic_ostream : virtual public basic_ios<_CharT, _Traits>
59       {
60       public:
61         // Types (inherited from basic_ios):
62         typedef _CharT          char_type;
63         typedef typename _Traits::int_type    int_type;
64         typedef typename _Traits::pos_type    pos_type;
65         typedef typename _Traits::off_type    off_type;
66         typedef _Traits         traits_type;
67
68         // Non-standard Types:
69         typedef basic_streambuf<_CharT, _Traits>    __streambuf_type;
70         typedef basic_ios<_CharT, _Traits>     __ios_type;
71         typedef basic_ostream<_CharT, _Traits>     __ostream_type;
72         typedef num_put<_CharT, ostreambuf_iterator<_CharT, _Traits> >
```



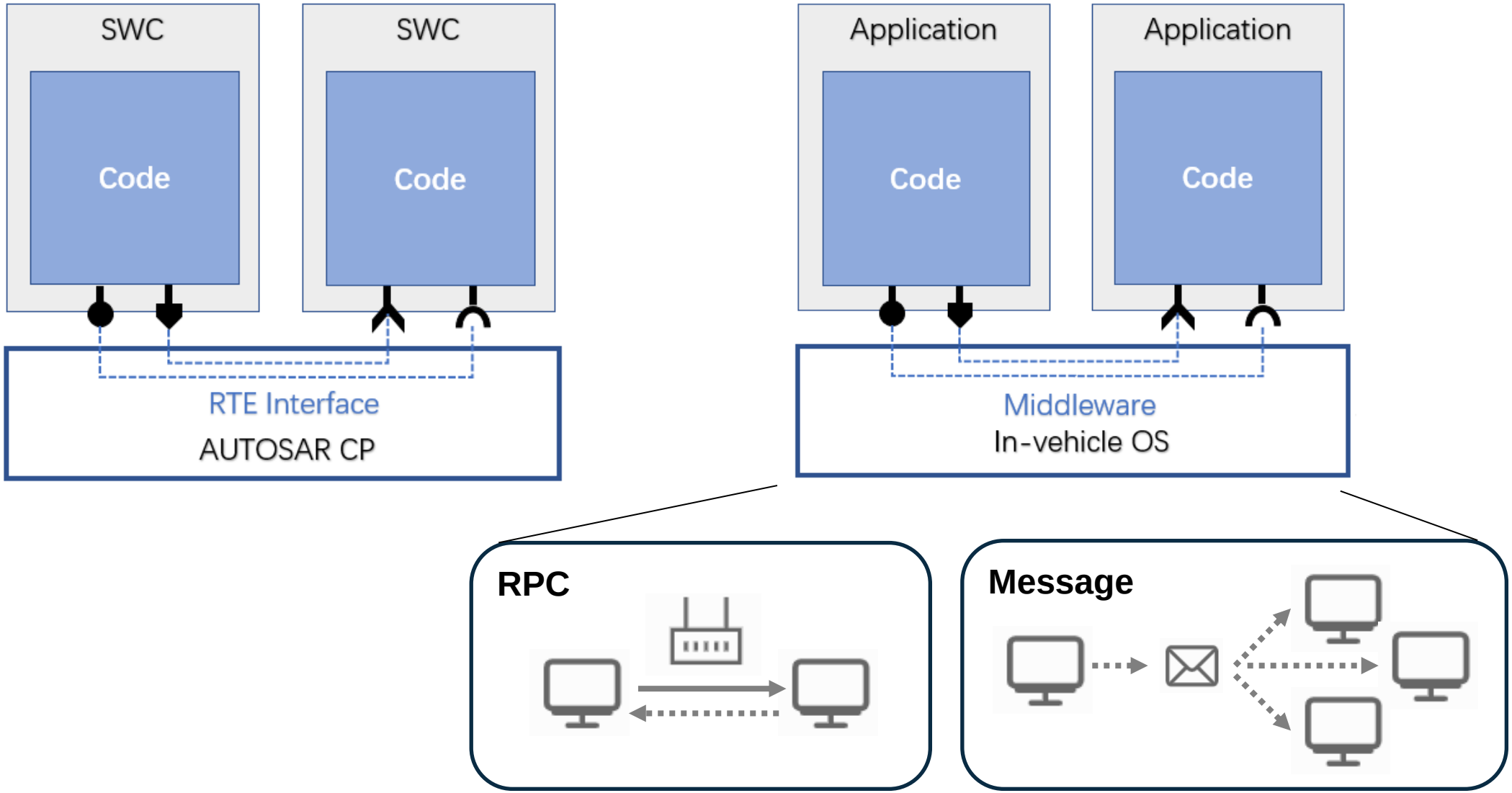
- 系统复杂
  - 目前市场上很多应用软件设计和管理工具并不兼容自研的非标准中间件

解决方案

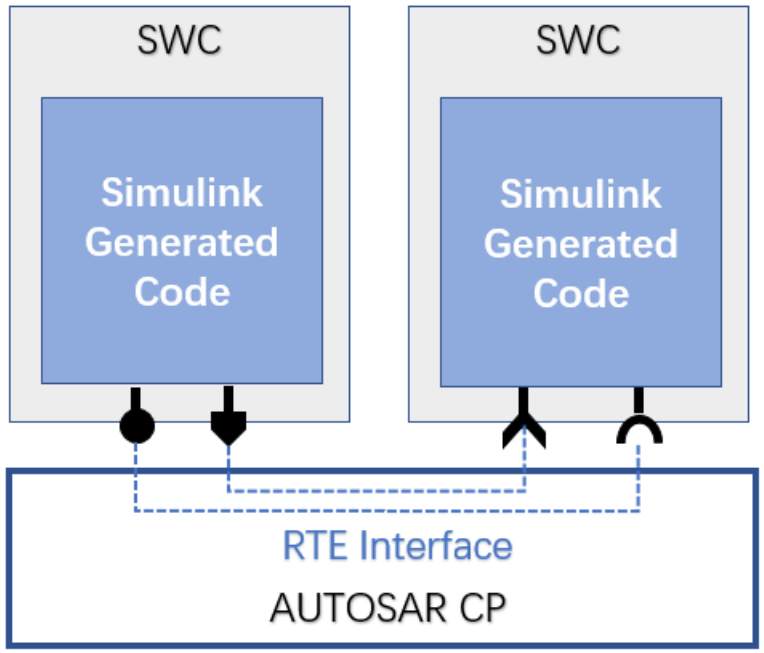
- 如何对SOA软件行为进行建模
  - SOA 专属的建模语义
  - Simulink 新特性
  - 包装代码生成器
- 如何维护SOA软件系统
  - 基于模型的系统工程
  - 基于System Composer深度定制

第1部分 如何对SOA软件行为进行建模

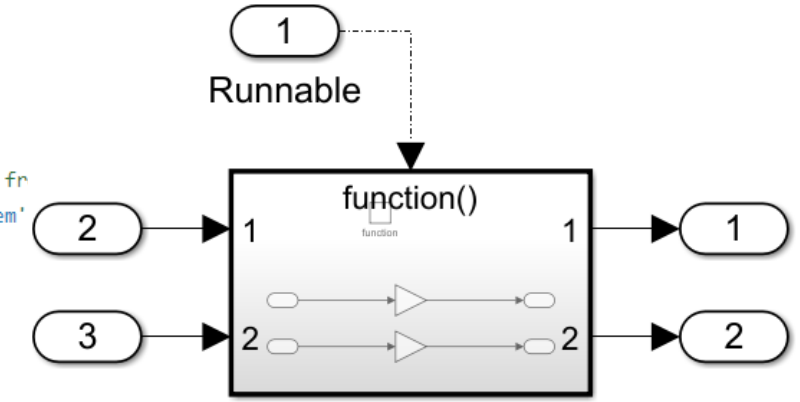
不同建模环境之间的比较



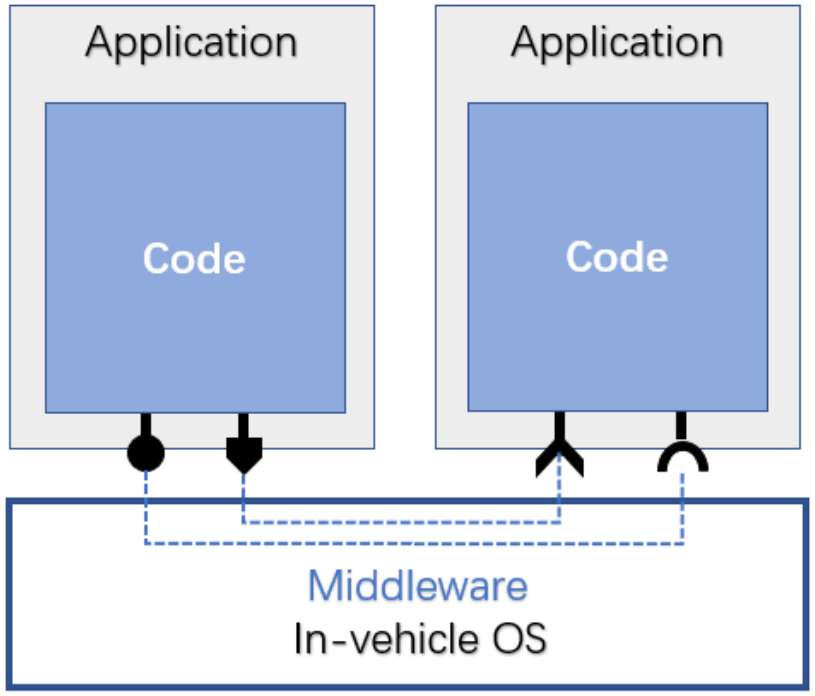
在AUTOSAR 环境下典型的基于模型开发过程

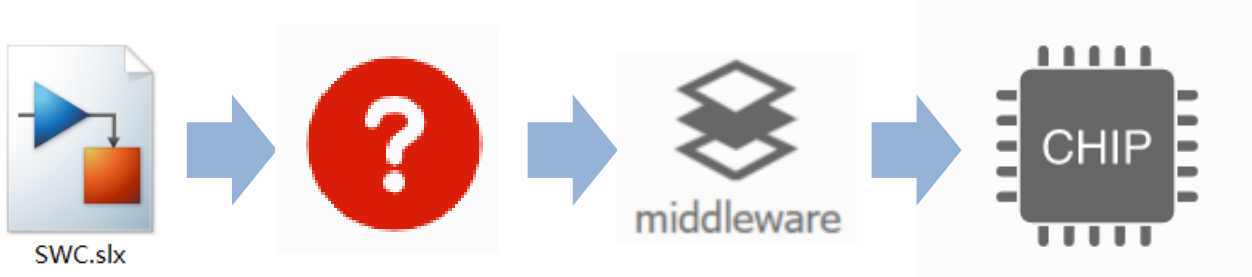



```
19  void Runnable(void)
20  {
21    /* RootInportFunctionCallGenerator generated fr
22     *  SubSystem: '<Root>/Function-Call Subsystem'
23     */
24    /* Outport: '<Root>/Out1' incorporates:
25     *  Gain: '<S1>/Gain'
26     *  Inport: '<Root>/In1'
27     */
28    Rte_IWrite_Runnable_Out1_Out1(2.0 * Rte_IRead_Runnable_In1_In1());
29
30    /* Outport: '<Root>/Out2' incorporates:
31     *  Gain: '<S1>/Gain1'
32     *  Inport: '<Root>/In2'
33     */
34    Rte_IWrite_Runnable_Out2_Out2(2.0 * Rte_IRead_Runnable_In2_In2());
35
36    /* End of Outputs for RootInportFunctionCallGenerator generated from: '<Root>/Ru
37  }
```

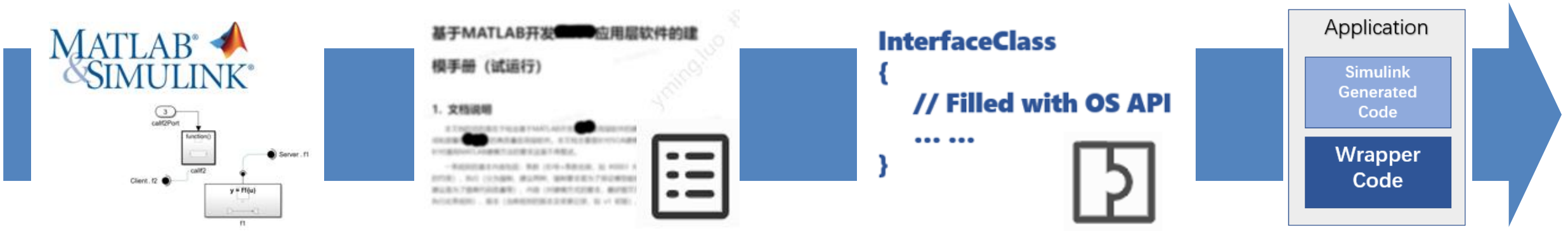


如何在车载OS环境下进行建模





如何在车载OS环境下进行建模



产品新特性

- 支持SOA的模型行为

全新建模规则

- 在实践中总结出一套符合SOA软件开发要求的建模规则

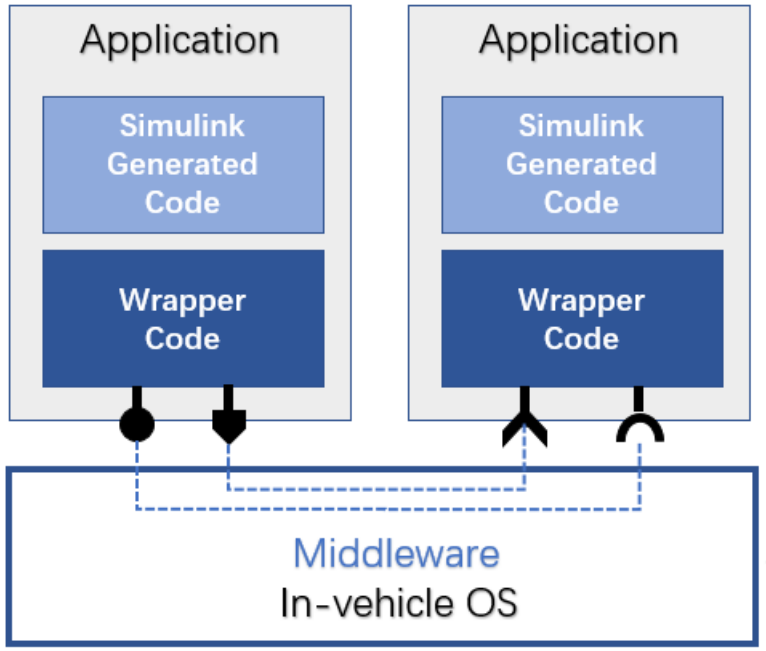
包装代码生成器

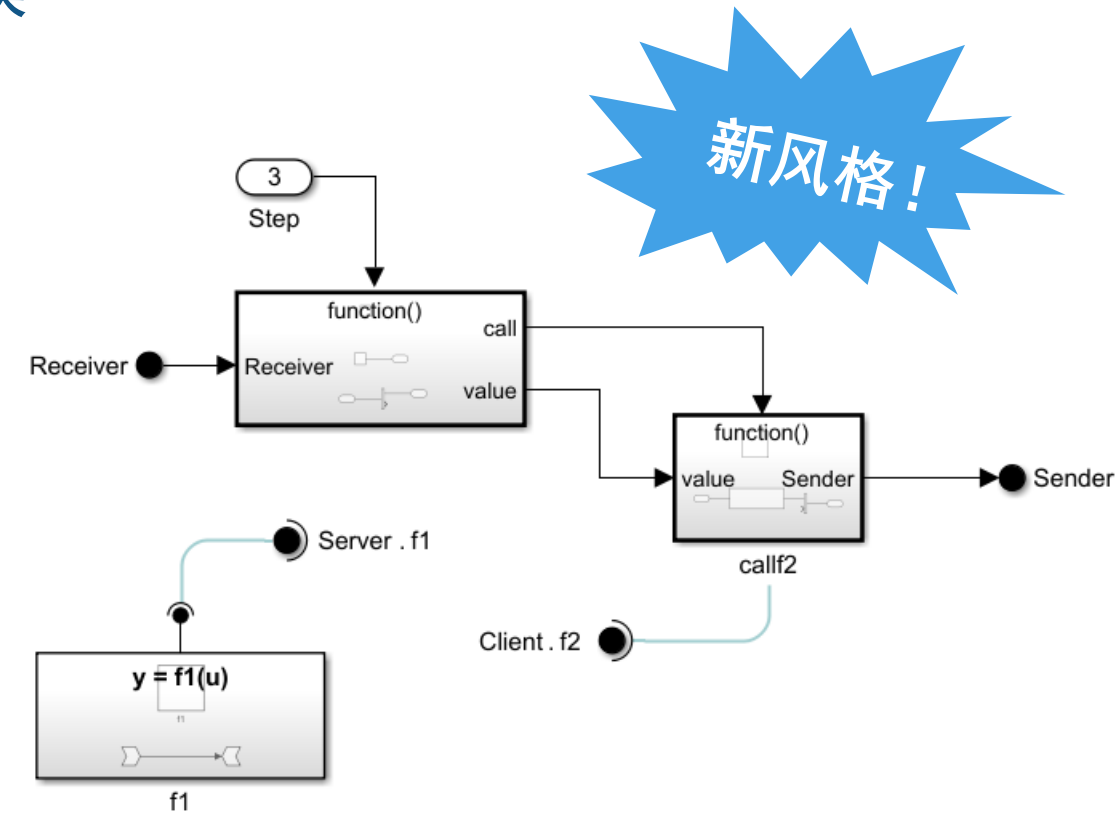
- 作为Simulink侧代码和OS接口代码的桥接

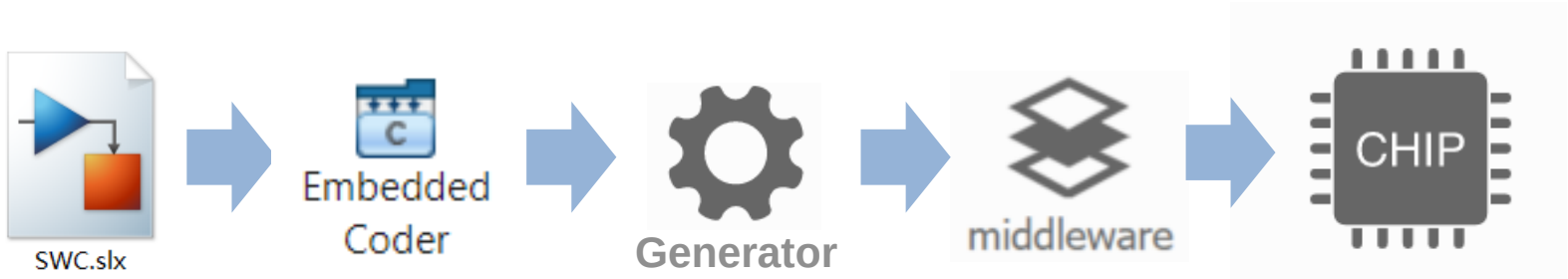
部署

- 可以直接部署到OS环境下，和其他OS应用无任何区别

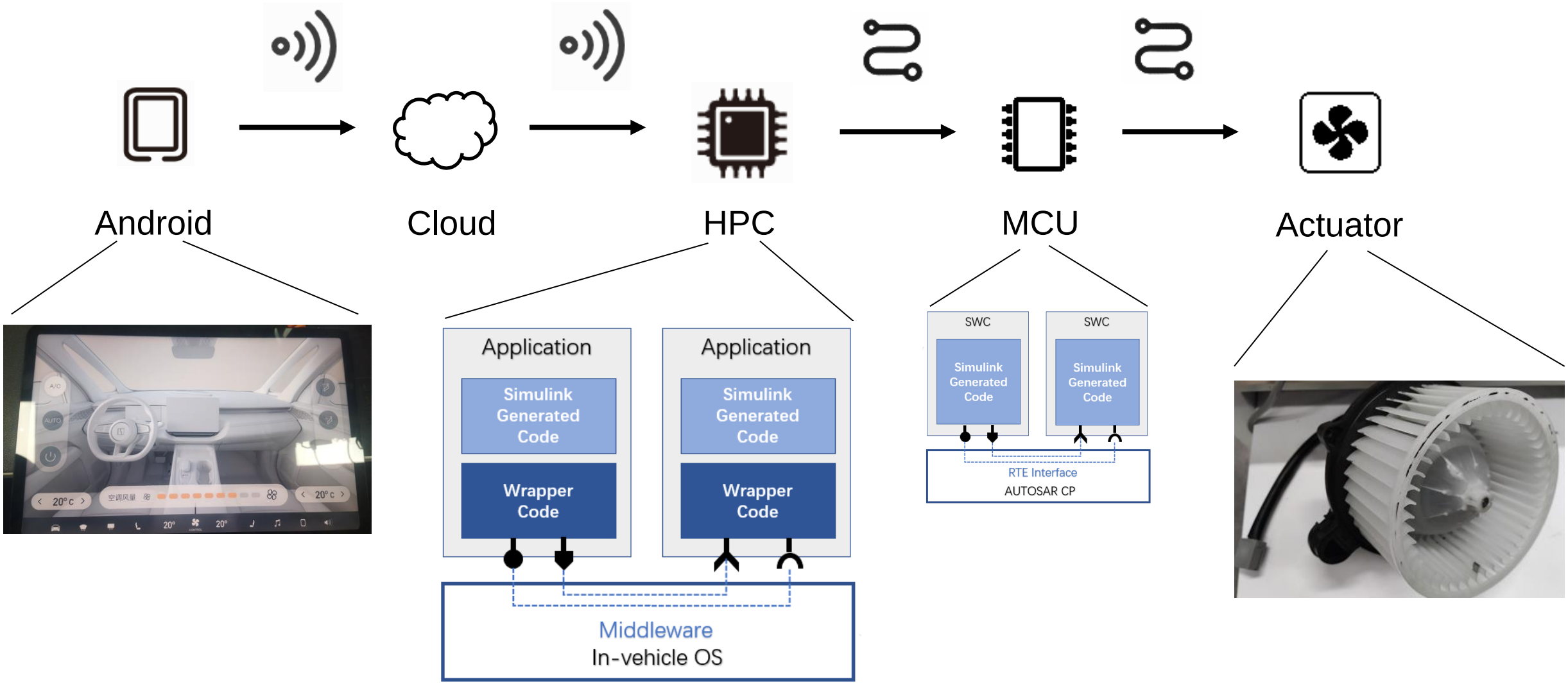
如何在车载OS环境下进行建模



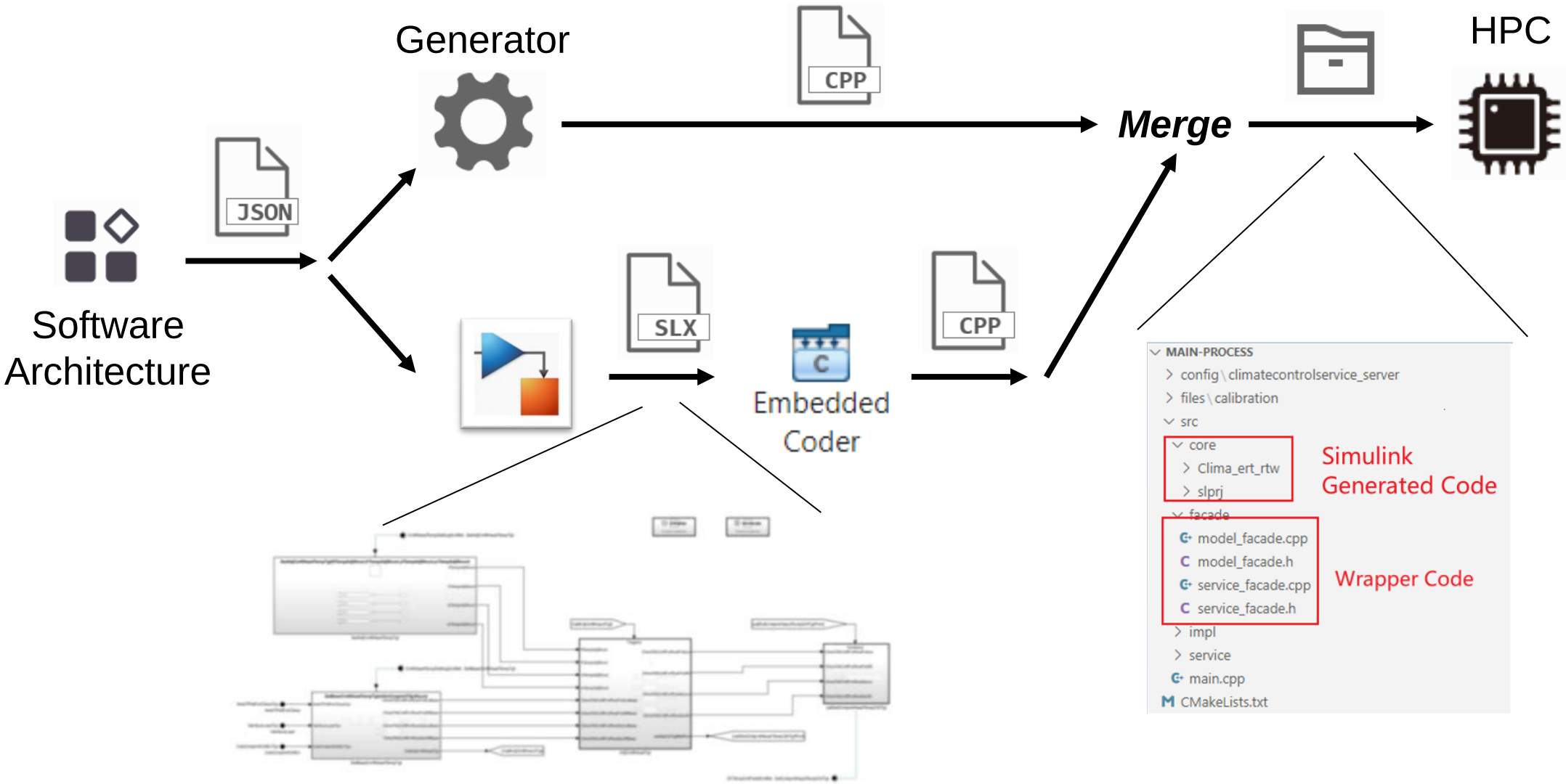




真实案例

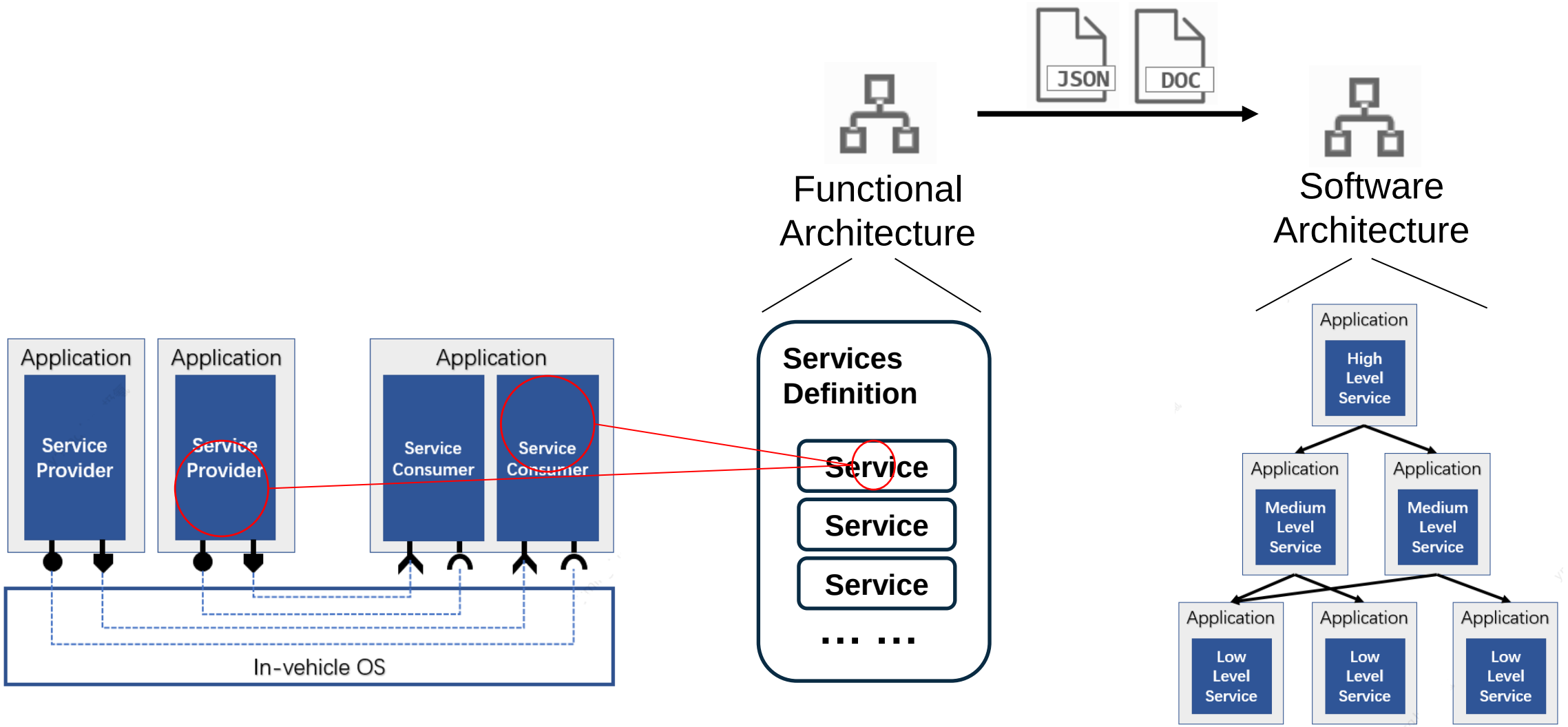


真实案例

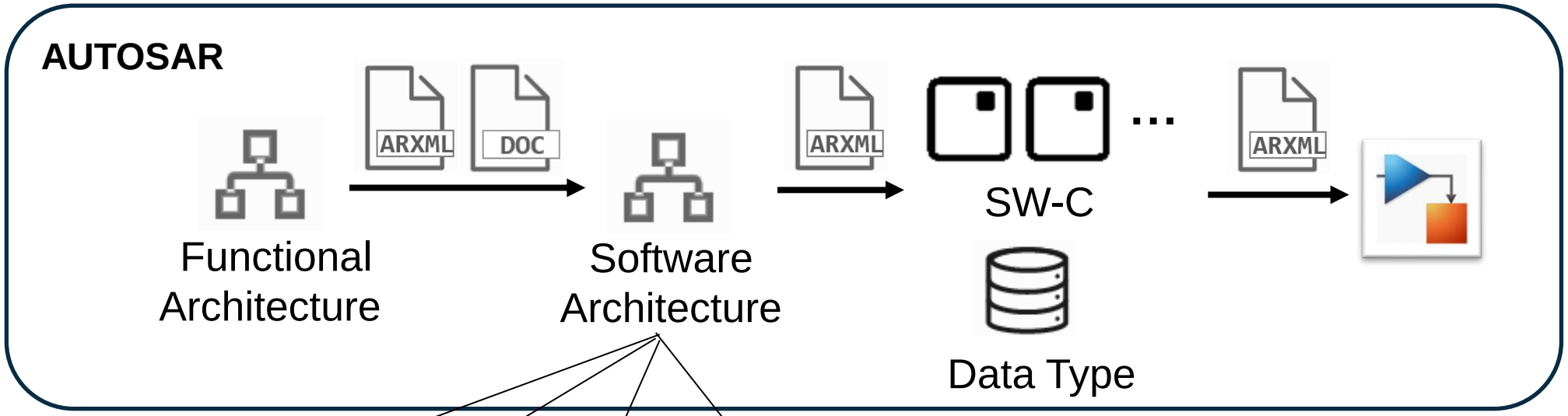

Generator
CPP
HPC
Merge
JSON
Software
Architecture
SLX
Embedded
Coder
CPP
MAIN-PROCESS
config \ climatecontrolservice_server
files \ calibration
src
core
Clima_ert_rtw
slprj
Simulink
Generated Code
facade
model_facade.cpp
model_facade.h
service_facade.cpp
service_facade.h
Wrapper Code
impl
service
main.cpp
CMakeLists.txt

第2部分 如何维护SOA软件系统

软件架构是如何起作用的

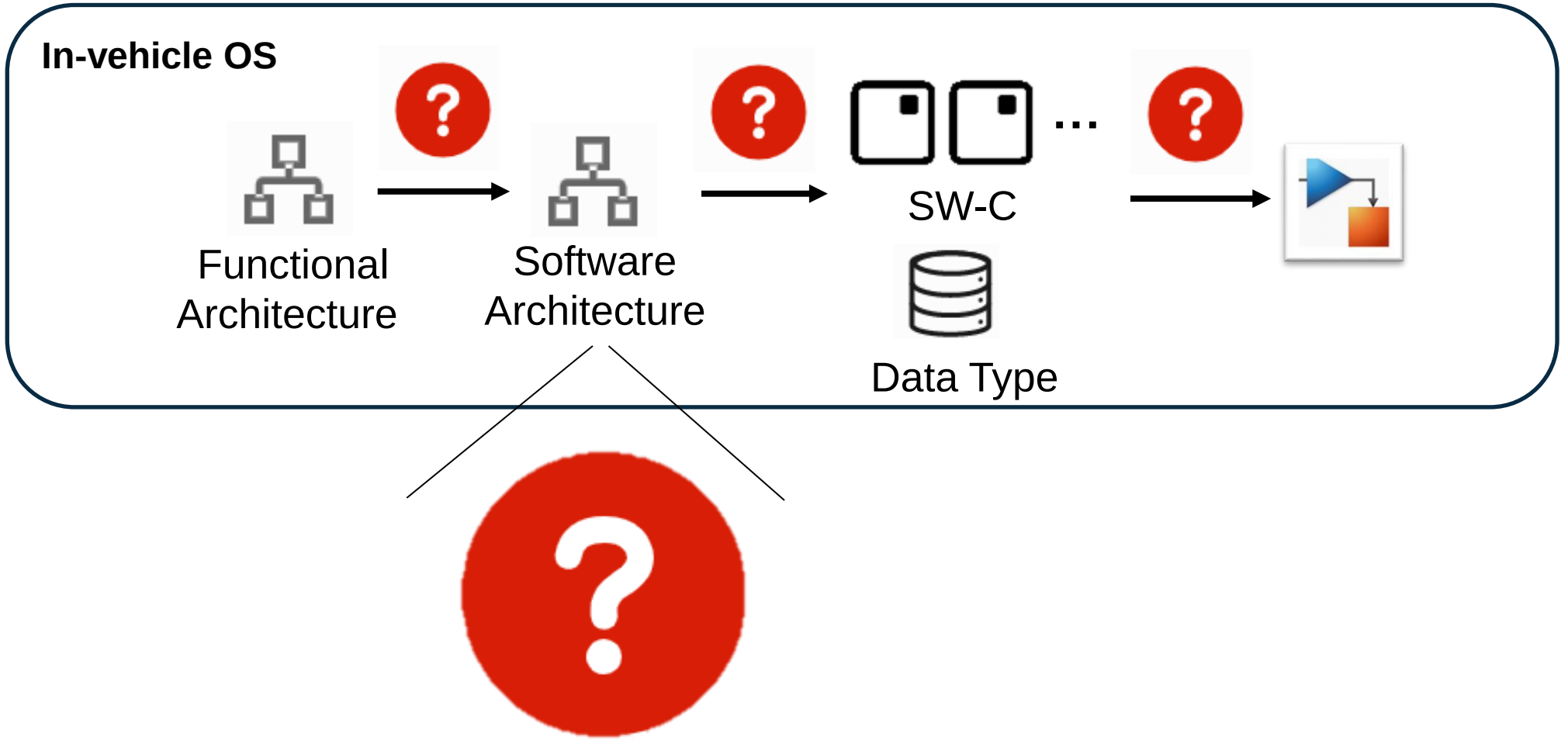


软件管理的窘况

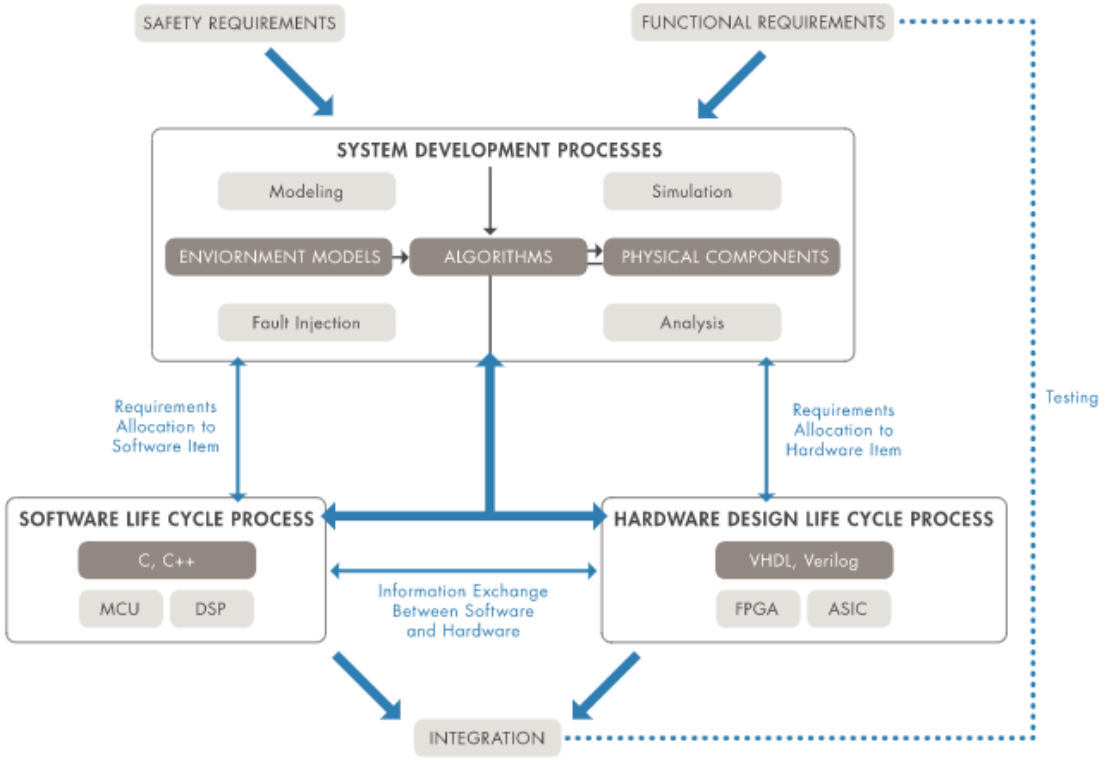


面向AUTOSAR!

软件管理的窘况



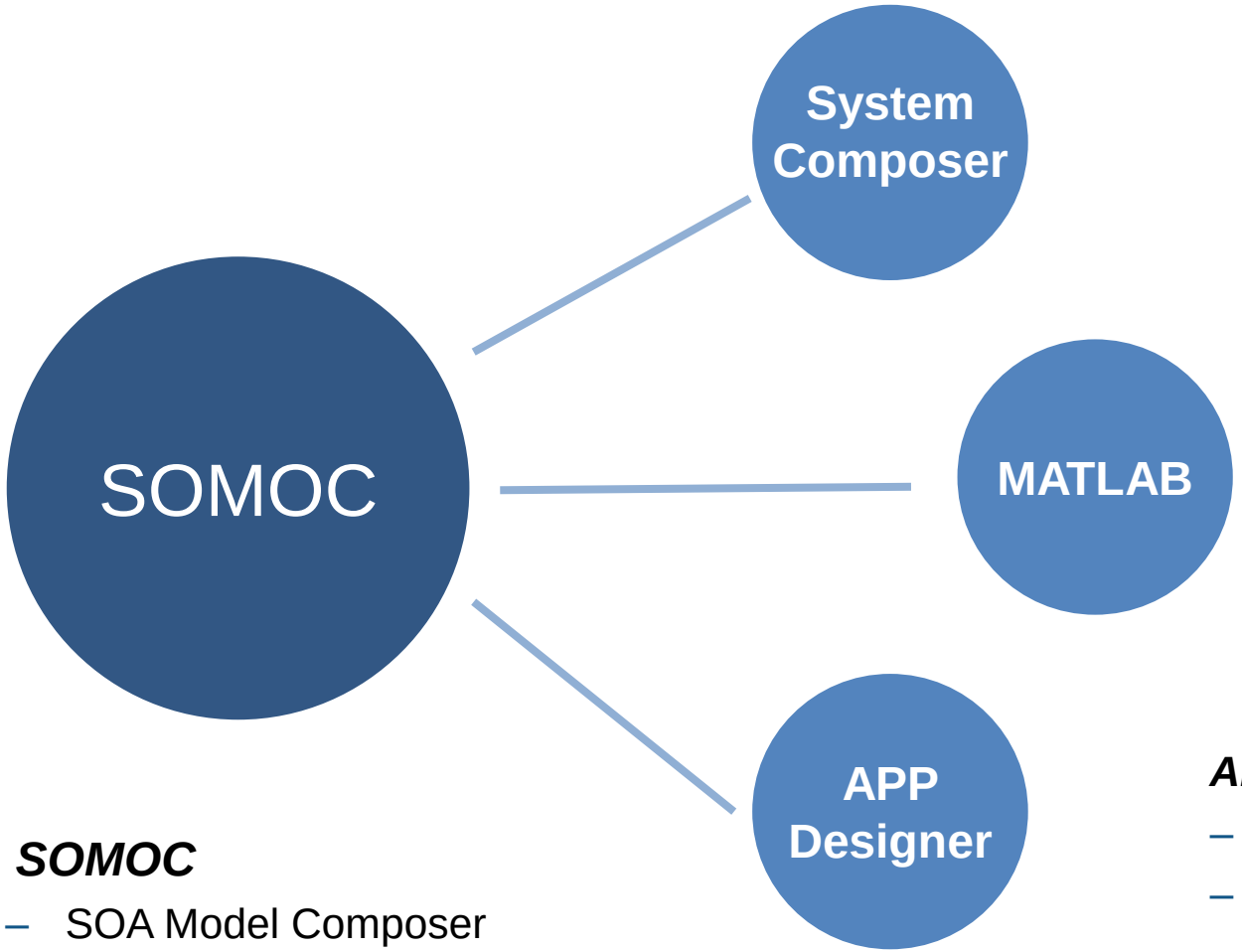
基于模型的系统工程



- 基于模型的系统工程（MBSE）
  - 工程师使用基于模型的系统工程(MBSE)来管理系统复杂性，改善通信，并产生优化的系统。
  - MATLAB®，Simulink®和System Composer™一起创建了一个用于创建描述性体系结构模型的单一环境，这些模型可以无缝地连接到详细的实现模型。

https://www.mathworks.com/solutions/model-based-systems-engineering.html

自研软件架构工具



System Composer

- 基于模型的系统工程
- 支持软件架构元素设计
- 强大的可视化能力
- 支持自定义

MATLAB

- 强大的编程语言
- 面向编程特性
- 丰富的产品API

APP Designer

- 自定义桌面应用开发
- 简单易用

SOMOC

- SOA Model Composer
- 专门为车载OS软件架构开发而定制

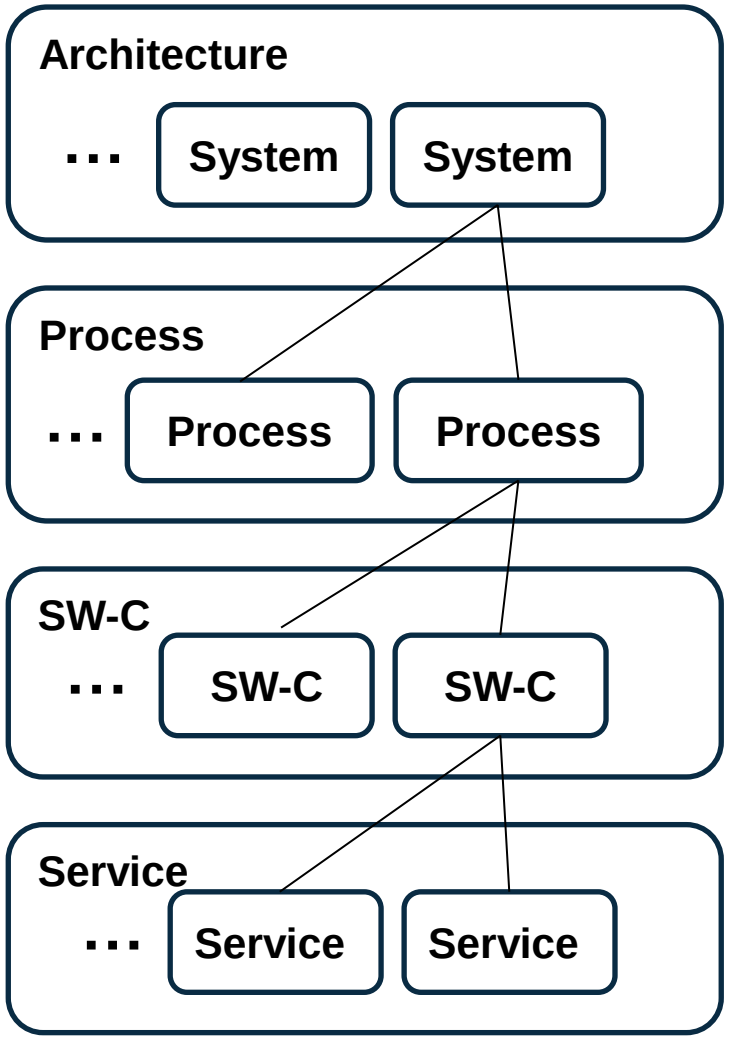
SOMOC中不同的架构层级

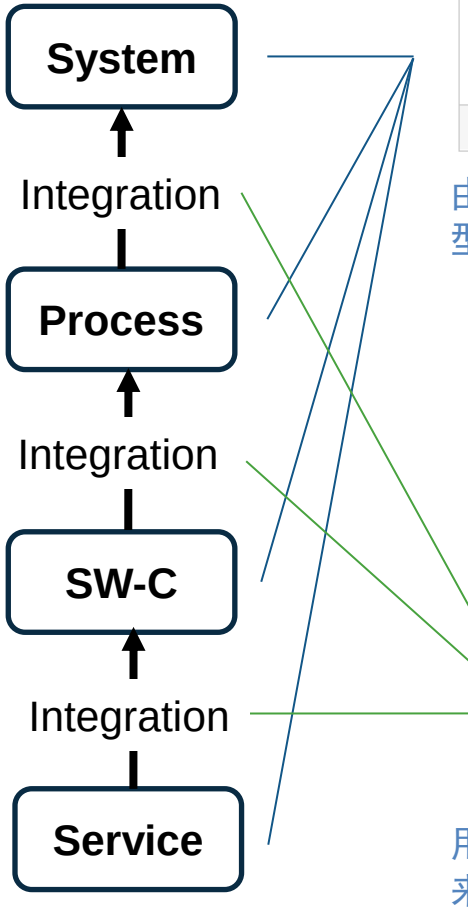
架构级
- 某个系统的全局视野

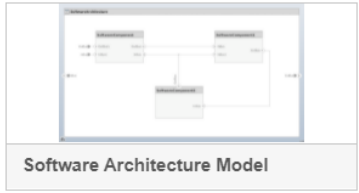
进程级
- 最小的集成单元
- 包含多个软件组件
- 相互独立的应用
- 通过中间件进行通信

软件组件级
- 最小的开发单元
- 包含多个服务的实现
- 包含具体的软件逻辑

服务级
- 包含对服务接口的描述
- 传递自上游的功能设计



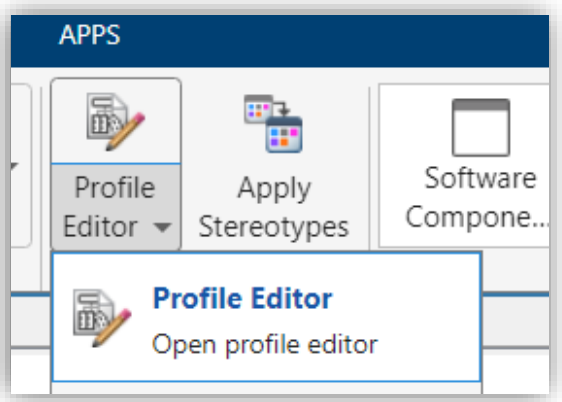


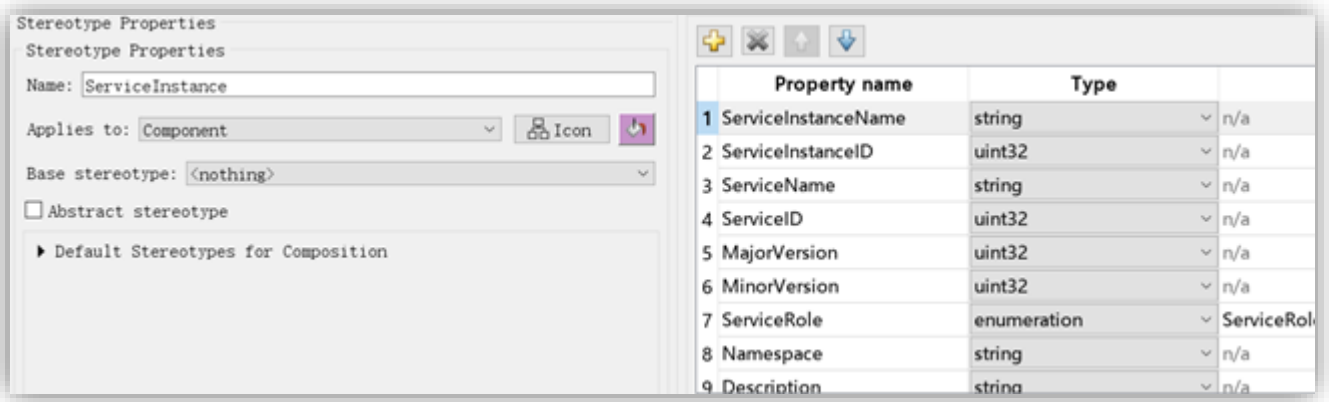


由System Composer模型来代表架构元素

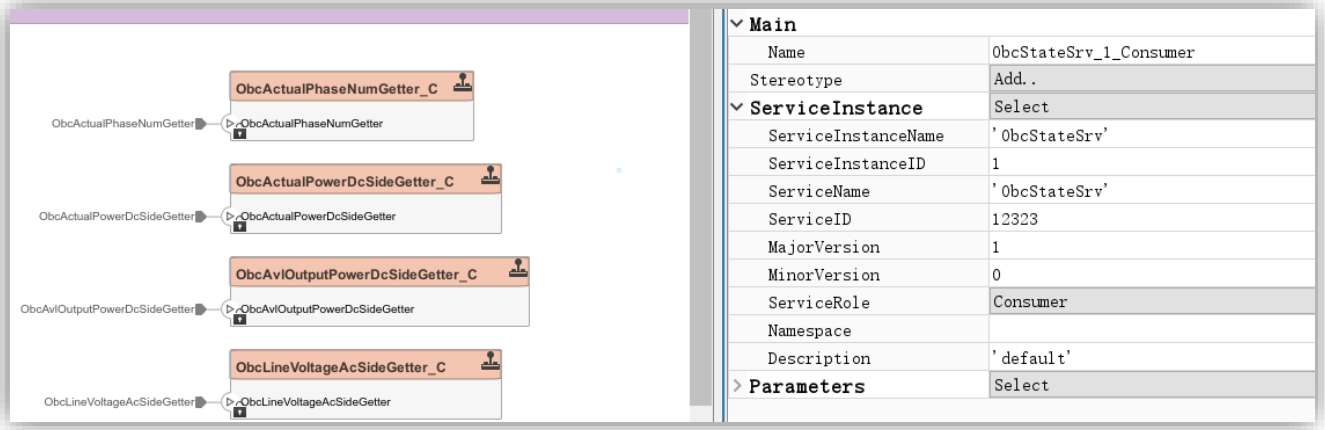
用Reference Component来反映架构元素之间的关联

使用Profile来增强System Composer模型

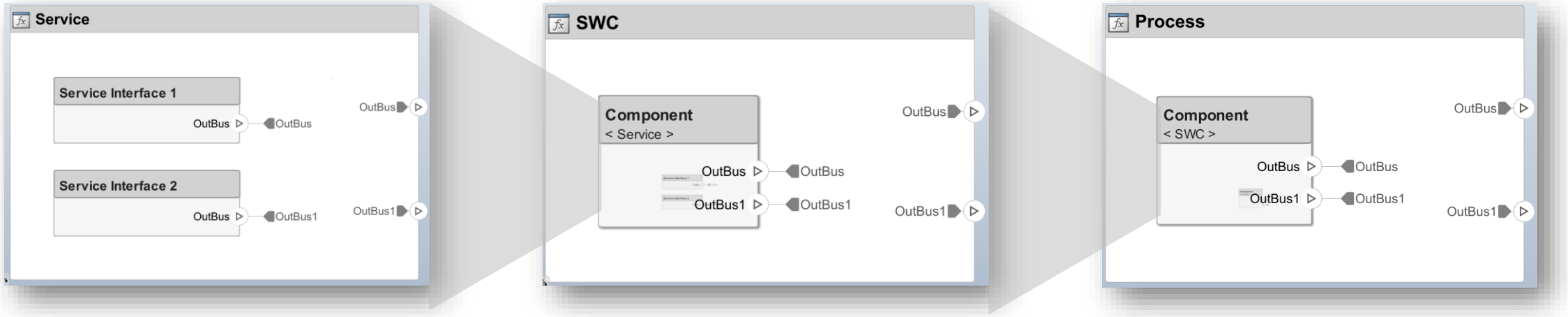




- Profile
  - 可以赋予模块自定义属性
  - 增强System Composer以满足特定系统的需求

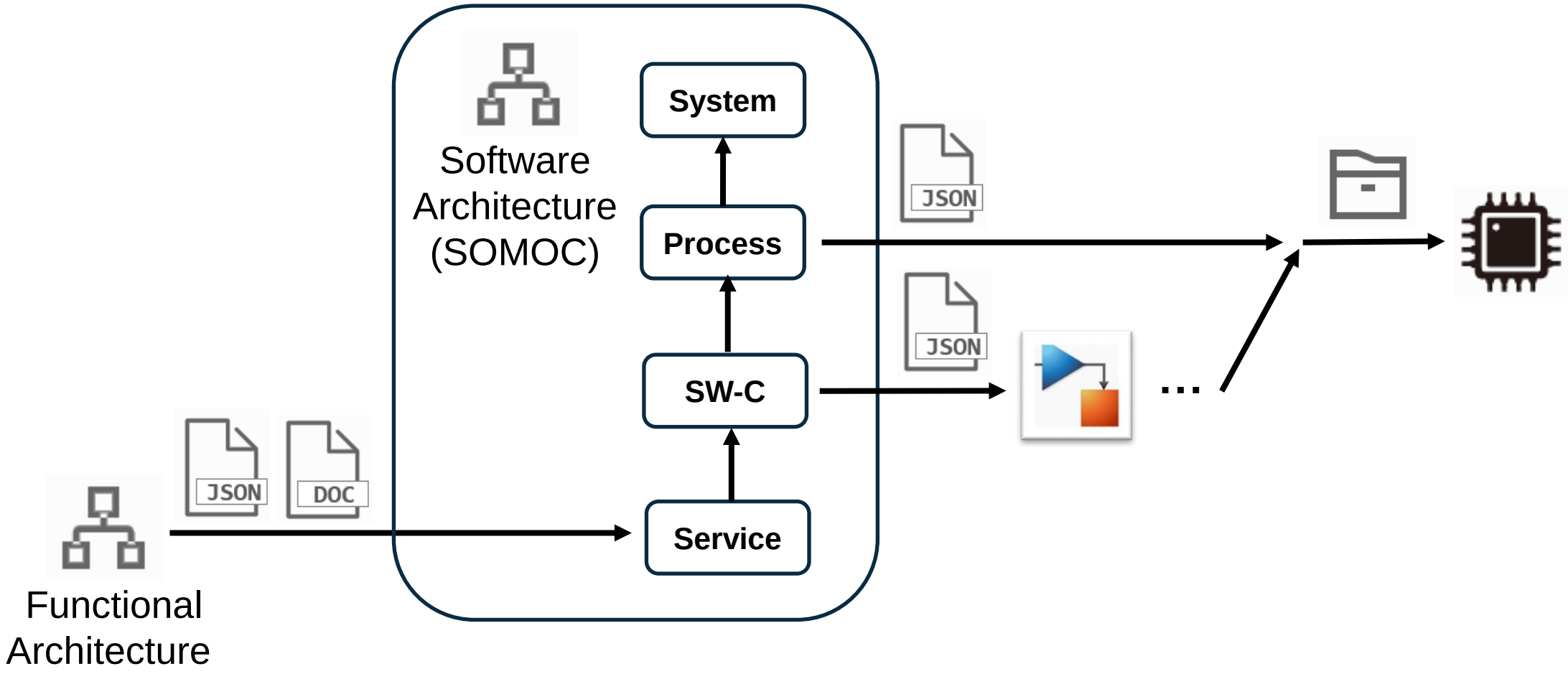


使用Reference Component来联系各设计元素

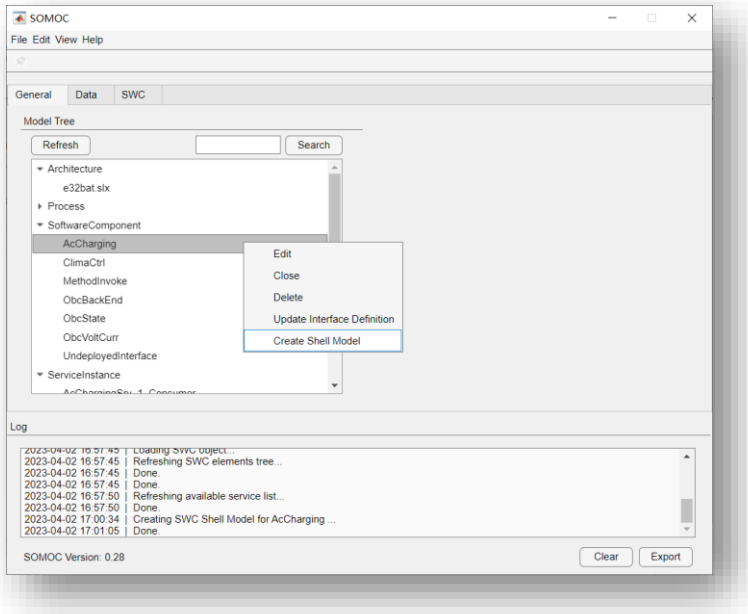


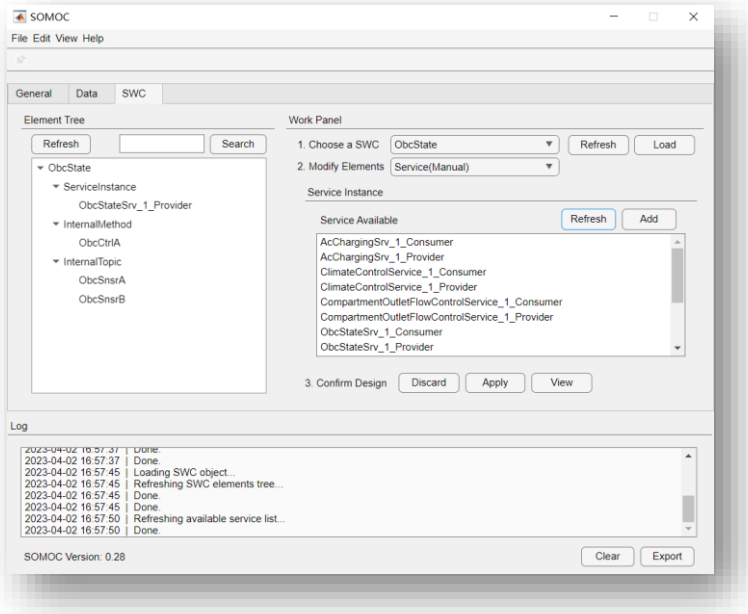
- Reference Component
  - 在当前System Composer模型中引用另一个模型
  - 用于模仿低层级元素集成到高层级元素的动作
  - 通过这种方式软件架构的各个元素可以被集中组织

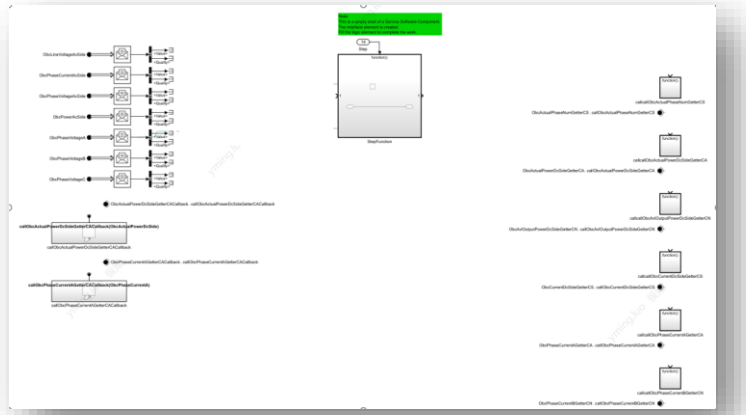
SOMOC 工作流


Software
Architecture
(SOMOC)
System
Process
SW-C
Service
JSON
DOC
Functional
Architecture
JSON
JSON
...

SOMOC GUI







架构元素树

- 不同级别架构元素的树形图
- 元素树节点都有可以操作的上下文菜单

元素设计界面

- 提供架构元素的设计界面
- 增加、移除或者修改某个架构元素
- 比直接操作System Composer模型更加简便

导出行为模型

- 根据架构元素接口定义导出相应的行为模型
- 传递给下游的模型开发工程师做进一步软件开发

总结

- 如何对SOA软件行为进行建模
  - SOA 专属的建模语义
  - Simulink 新特性
  - 包装代码生成器
- 如何维护SOA软件系统
  - 基于模型的系统工程
  - 基于System Composer深度定制

MATLAB EXPO

Thank you