**imageNeusoft Confidential**

File No. : ProjectNo.- D00-D01-T01-流水号

NeuSAR aCore

软件需求规格说明书

（Software requirement specification）

**(CM DDS)**

Neusoft Reach Automotive Technology Co., Ltd

Change Log

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Version** | **Contents Revised** | **Status** | **Reviser/**  **Date** | **Approve/Date** |
| 1 | 0.5 | 新建 |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Contents**

[1 引言（Introduction） 1](#_Toc98762463)

[1.1 目的（Goal） 1](#_Toc98762464)

[1.2 范围（Scope） 1](#_Toc98762465)

[1.3 参考文档（Reference） 1](#_Toc98762466)

[1.4 术语及缩略语（Terms And Abbreviations） 1](#_Toc98762467)

[2 软件系统概述（Software System Overview） 1](#_Toc98762468)

[2.1 软件系统背景（Software System Background） 1](#_Toc98762469)

[2.2 软件系统目标（Software System Goal） 1](#_Toc98762470)

[2.3 外部关联 （External Association） 1](#_Toc98762471)

[3 功能需求（Functional Requirement） 1](#_Toc98762472)

[3.1 功能需求1(Function Requirement No.1) 1](#_Toc98762473)

[3.1.1 [SWRD\_CM\_DDS\_00001] DDS网络绑定 1](#_Toc98762474)

[3.1.2 [SWRD\_CM\_DDS\_00002]通过域参与者Qos的OfferService函数映射 1](#_Toc98762475)

[3.1.3 [SWRD\_CM\_DDS\_00003] DDS绑定应该为服务实例赋予一个 DomainParticipant 1](#_Toc98762476)

[3.1.4 [SWRD\_CM\_DDS\_00004] 为服务接口的每一个event赋予一个topic和DataWriter 1](#_Toc98762477)

[3.1.5 [SWRD\_CM\_DDS\_00005] 为服务接口的每一个method赋予一个request和reply topic，和DataWriter和DataReader 1](#_Toc98762478)

[3.1.6 [SWRD\_CM\_DDS\_00006] 为服务接口的每一个hasNotify为true的field赋予一个topic和DataWriter 1](#_Toc98762479)

[3.1.7 [SWRD\_CM\_DDS\_00007] 为服务接口的所有field的 getter和setter赋予一个request， reply topic和DataWriter和DataReader 1](#_Toc98762480)

[3.1.8 [SWRD\_CM\_DDS\_00008]OfferService函数映射 1](#_Toc98762481)

[3.1.9 [SWRD\_CM\_DDS\_00009] 通过域参与者Qos的StopOfferService方法的映射 1](#_Toc98762482)

[3.1.10 [SWRD\_CM\_DDS\_00010] 在client端查找一个合适的DomainParticipant 1](#_Toc98762483)

[3.1.11 [SWRD\_CM\_DDS\_00011] 在client端创建一个合适的DomainParticipant 1](#_Toc98762484)

[3.1.12 [SWRD\_CM\_DDS\_00012] 通过创建或者查找到的DomainParticipants来发现remote service 1](#_Toc98762485)

[3.1.13 [SWRD\_CM\_DDS\_00013] StartFindService方法的映射 1](#_Toc98762486)

[3.1.14 [SWRD\_CM\_DDS\_00014] BuiltinParticipantListener的定义 1](#_Toc98762487)

[3.1.15 [SWRD\_CM\_DDS\_00015] 为DomainParticipant绑定一个BuiltinParticipantListener 1](#_Toc98762488)

[3.1.16 [SWRD\_CM\_DDS\_00016] StopFindService方法的映射 1](#_Toc98762489)

[3.1.17 [SWRD\_CM\_DDS\_00017] BuiltinParticipantListener和DomainParticipant解绑 1](#_Toc98762490)

[3.1.18 [SWRD\_CM\_DDS\_00018] event和topic的映射 1](#_Toc98762491)

[3.1.19 [SWRD\_CM\_DDS\_00019] event topic data type定义 1](#_Toc98762492)

[3.1.20 [SWRD\_CM\_DDS\_00020] event Subscribe方法的映射 1](#_Toc98762493)

[3.1.21 [SWRD\_CM\_DDS\_00021] 为event订阅创建DataReader 1](#_Toc98762494)

[3.1.22 [SWRD\_CM\_DDS\_00022] event DataReaderListener的定义 1](#_Toc98762495)

[3.1.23 [SWRD\_CM\_DDS\_00023] event GetSubscriptionState方法的映射 1](#_Toc98762496)

[3.1.24 [SWRD\_CM\_DDS\_00024] event GetNewSamples方法的映射 1](#_Toc98762497)

[3.1.25 [SWRD\_CM\_DDS\_00025] event GetFreeSampleCount方法的映射 1](#_Toc98762498)

[3.1.26 [SWRD\_CM\_DDS\_00026] event SetReceiveHandler方法的映射 1](#_Toc98762499)

[3.1.27 [SWRD\_CM\_DDS\_00027] event UnsetReceiveHandler方法的映射 1](#_Toc98762500)

[3.1.28 [SWRD\_CM\_DDS\_00028] event SetSubscriptionStateHandler方法的映射 1](#_Toc98762501)

[3.1.29 [SWRD\_CM\_DDS\_00029] UnsetSubscriptionStateHandler方法的映射 1](#_Toc98762502)

[3.1.30 [SWRD\_CM\_DDS\_00030] method到DDS method和topic的映射 1](#_Toc98762503)

[3.1.31 [SWRD\_CM\_DDS\_00031] Request Topic data type定义 1](#_Toc98762504)

[3.1.32 [SWRD\_CM\_DDS\_00032] Reply Topic data type定义 1](#_Toc98762505)

[3.1.33 [SWRD\_CM\_DDS\_00033] ApApplicationError中ara::core::ErrorCode的映射 1](#_Toc98762506)

[3.1.34 [SWRD\_CM\_DDS\_00034] 在client端创建DataWriter处理请求 1](#_Toc98762507)

[3.1.35 [SWRD\_CM\_DDS\_00035] 在client端创建DataReader处理method应答 1](#_Toc98762508)

[3.1.36 [SWRD\_CM\_DDS\_00036] 在server端创建一个DataReader处理method请求 1](#_Toc98762509)

[3.1.37 [SWRD\_CM\_DDS\_00037] 在server端创建DataWriter处理method应答 1](#_Toc98762510)

[3.1.38 [SWRD\_CM\_DDS\_00038] 在client端调用服务方法 1](#_Toc98762511)

[3.1.39 [SWRD\_CM\_DDS\_00039] 通知client method的响应并将method结果通知client应用程序。 1](#_Toc98762512)

[3.1.40 [SWRD\_CM\_DDS\_00040] 在server端处理以event driven处理方法调用 1](#_Toc98762513)

[3.1.41 [SWRD\_CM\_DDS\_00041] 在server端创建DataReaderListener处理异步请求 1](#_Toc98762514)

[3.1.42 [SWRD\_CM\_DDS\_00042] 在server端以polling处理一个方法 1](#_Toc98762515)

[3.1.43 [SWRD\_CM\_DDS\_00043] 在server端发送method应答 1](#_Toc98762516)

[3.1.44 [SWRD\_CM\_DDS\_00044] field的hasNotifier的映射 1](#_Toc98762517)

[3.1.45 [SWRD\_CM\_DDS\_00045] Field Notifier DDS Topic data type 的定义 1](#_Toc98762518)

[3.1.46 [SWRD\_CM\_DDS\_00046] Field Update方法的映射 1](#_Toc98762519)

[3.1.47 [SWRD\_CM\_DDS\_00047] field Subscribe方法的映射 1](#_Toc98762520)

[3.1.48 [SWRD\_CM\_DDS\_00048] 为field订阅创建DataReader 1](#_Toc98762521)

[3.1.49 [SWRD\_CM\_DDS\_00049] 为field订阅创建DataReaderListener 1](#_Toc98762522)

[3.1.50 [SWRD\_CM\_DDS\_00050]field Unsubscribe的映射 1](#_Toc98762523)

[3.1.51 [SWRD\_CM\_DDS\_00051]field GetSubscriptionState方法的映射 1](#_Toc98762524)

[3.1.52 [SWRD\_CM\_DDS\_00052]field GetNewSamples方法的映射 1](#_Toc98762525)

[3.1.53 [SWRD\_CM\_DDS\_00053]field GetFreeSampleCount方法的映射 1](#_Toc98762526)

[3.1.54 [SWRD\_CM\_DDS\_00054]field SetReceiveHandler方法的映射 1](#_Toc98762527)

[3.1.55 [SWRD\_CM\_DDS\_00055]field UnsetReceiveHandler方法的映射 1](#_Toc98762528)

[3.1.56 [SWRD\_CM\_DDS\_00056]field SetSubscriptionStateHandler方法映射 1](#_Toc98762529)

[3.1.57 [SWRD\_CM\_DDS\_00057]field UnsetSubscriptionStateHandler方法的映射 1](#_Toc98762530)

[3.1.58 [SWRD\_CM\_DDS\_00058] Field Get/Set方法到服务接口的服务方法和topic的映射 1](#_Toc98762531)

[3.1.59 [SWRD\_CM\_DDS\_00059] Field getter 和 setter中Request Topic data type 定义 1](#_Toc98762532)

[3.1.60 [SWRD\_CM\_DDS\_00060] 对于Field getter 和 setter Reply Topic data type 的定义 1](#_Toc98762533)

[3.1.61 [SWRD\_CM\_DDS\_00061] 为client端get和set请求创建DataWriter 1](#_Toc98762534)

[3.1.62 [SWRD\_CM\_DDS\_00062] 在client端为Field get/set的应答创建DataReader 1](#_Toc98762535)

[3.1.63 [SWRD\_CM\_DDS\_00063] 在server端为Field set/get请求创建DataReader 1](#_Toc98762536)

[3.1.64 [SWRD\_CM\_DDS\_00064] 在server端为Field get/set应答创建一个DataWriter 1](#_Toc98762537)

[3.1.65 [SWRD\_CM\_DDS\_00065] 在client端调用Field get/set方法 1](#_Toc98762538)

[3.1.66 [SWRD\_CM\_DDS\_00066] 当收到Field get/set方法的响应时通知client 1](#_Toc98762539)

[3.1.67 [SWRD\_CM\_DDS\_00067] 在server端处理field相关联的get/set方法 1](#_Toc98762540)

[3.1.68 [SWRD\_CM\_DDS\_00068] 在server端为Field getters和setters创建DataReaderListener用于处理异步请求。 1](#_Toc98762541)

[3.1.69 [SWRD\_CM\_DDS\_00069] 在server端以polling处理Field get/set方法 1](#_Toc98762542)

[3.1.70 [SWRD\_CM\_DDS\_00070] 在server端为Field get/set发送一个应答 1](#_Toc98762543)

[3.1.71 [SWRD\_CM\_DDS\_00071] DDS标准序列化 1](#_Toc98762544)

[3.1.72 [SWRD\_CM\_DDS\_00072] Strings的编码格式和大小端 1](#_Toc98762545)

[3.1.73 [SWRD\_CM\_DDS\_00073] VARIANT类型的序列化 1](#_Toc98762546)

[3.1.74 [SWRD\_CM\_DDS\_00074] 服务发现协议选择 1](#_Toc98762547)

[3.1.75 [SWRD\_CM\_DDS\_00075] 通过域参与者Qos做服务发现时topic命名规则 1](#_Toc98762548)

[3.1.76 [SWRD\_CM\_DDS\_00076] 为服务接口的没一个trigger赋予一个topic和DataWriter 1](#_Toc98762549)

[3.1.77 [SWRD\_CM\_DDS\_00077]通过topic的OfferService函数映射 1](#_Toc98762550)

[3.1.78 [SWRD\_CM\_DDS\_00078] 在ara.com://services/discovery主题上发布服务id、服务实例id和ServiceInterface contract version 1](#_Toc98762551)

[3.1.79 [SWRD\_CM\_DDS\_00079] 通过Topic的StopOfferService方法的映射 1](#_Toc98762552)

[3.1.80 [SWRD\_CM\_DDS\_00080] FindService方法的映射 1](#_Toc98762553)

[3.1.81 [SWRD\_CM\_DDS\_00081] 通过ara.com://services/discovery topic来发现remote service 1](#_Toc98762554)

[3.1.82 [SWRD\_CM\_DDS\_00082] StartFindService方法的映射 1](#_Toc98762555)

[3.1.83 [SWRD\_CM\_DDS\_00083] StopFindService方法的映射 1](#_Toc98762556)

[3.1.84 [SWRD\_CM\_DDS\_00084] trigger和topic的映射 1](#_Toc98762557)

[3.1.85 [SWRD\_CM\_DDS\_00085] trigger topic data type定义 1](#_Toc98762558)

[3.1.86 [SWRD\_CM\_DDS\_00086] trigger Send 方法的映射 1](#_Toc98762559)

[3.1.87 [SWRD\_CM\_DDS\_00087] 为trigger订阅创建DataReader 1](#_Toc98762560)

[3.1.88 [SWRD\_CM\_DDS\_00088] trigger DataReaderListener的定义 1](#_Toc98762561)

[3.1.89 [SWRD\_CM\_DDS\_00089] trigger GetNewSamples方法的映射 1](#_Toc98762562)

[3.1.90 [SWRD\_CM\_DDS\_00090] trigger SetReceiveHandler方法的映射 1](#_Toc98762563)

[3.1.91 [SWRD\_CM\_DDS\_00091] trigger UnsetReceiveHandler方法的映射 1](#_Toc98762564)

[3.1.92 [SWRD\_CM\_DDS\_00092] 1911已实现的特性 1](#_Toc98762565)

[4 非功能需求(Non-Functional Requirements) 1](#_Toc98762566)

[5 接口说明（API） 1](#_Toc98762567)

[附录A- 信息定义 1](#_Toc98762568)

[附录B- 配置信息 1](#_Toc98762569)

# 引言（Introduction）

## 目的（Goal）

本文是对AUTOSAR ADAPTIVE CM模块DDS绑定是具体需求。

## 范围（Scope）

使用于CM软件开发，测试，管理，人员。

## 参考文档（Reference）

|  |  |  |
| --- | --- | --- |
| **序号（No.）** | **文档名（Document Name）** | **版本名（Revision）** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 术语及缩略语（Terms And Abbreviations）

|  |  |
| --- | --- |
| **术语**  **Term/Abbreviation** | **描述（Description）** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 软件系统概述（Software System Overview）

## 软件系统背景（Software System Background）

参考《NeuSar aCore\_Software Requirement Specification CM API》。

## 软件系统目标（Software System Goal）

参考《NeuSar aCore\_Software Requirement Specification CM API》。

## 外部关联 （External Association）

无

# 功能需求（Functional Requirement）

## 功能需求1(Function Requirement No.1)

### [SWRD\_CM\_DDS\_00001] DDS网络绑定

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00001 |
| **Type** | Valid |
| **Description** | DDS网络绑定应该符合DDS Minimum Profile，DDS Wire Interoperability protocol (RTPS) 和DDS-XTYPES Minimal Programming Interface和Network Interoperability Profiles |
| **Upstream ID** | SWS\_CM\_11000 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00002]通过域参与者Qos的OfferService函数映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00002 |
| **Type** | Valid |
| **Description** | OfferService函数映射，应该执行以下的操作：  [SWS\_CM\_11002] 应该为服务接口赋予一个DomainParticipant  [SWS\_CM\_11003] 应该为服务接口的每个event赋予一个topic和DataWriter  [SWS\_CM\_10550] 应该为服务接口的每个trigger赋予一个topic和DataWriter  [SWS\_CM\_11029] 应该为服务接口的所有方法提供一个request topic和reply topic，并创建相应的DataReader和DataWriter  [SWS\_CM\_11030]应该为服务接口的每一个有hasNotifier的field赋予一个topic和DataWriter  [SWS\_CM\_11031] 应该为服务接口的所有field的get和set赋予一个request topic和reply topic，并创建并创建相应的DataReader和DataWriter，用于调用配置了hasGetter或者hasSetter属性的field  [SWS\_CM\_09004] 应该为 DomainParticipant的USER\_DATA QoS添加Service ID, Service Instance IDs,和服务接口contract版本 |
| **Upstream ID** | SWS\_CM\_11001 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 变更 |

### [SWRD\_CM\_DDS\_00003] DDS绑定应该为服务实例赋予一个 DomainParticipant

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00003 |
| **Type** | Valid |
| **Description** | DomainParticipant的配置在TPS\_ManifestSpecification描述：  domain ID ：从DdsProvidedServiceInstance的清单继承而来。  QoS Profile：从DdsProvidedServiceInstance定义的QoS Profile继承而来。  在创建DomainParticipant前，DDS绑定应该根据配置规范查询当前进程已经存在的DomainParticipant，如果能查到，DDS绑定应该将该DomainParticipant赋值给该实例，否则，应该创建一个新的DomainParticipant。  一旦服务实例的DomainParticipant可用，DDS绑定应该创建Publisher和Subscriber，Publisher和Subscriber的Partition QoS应该包含以下的Partition名称：  "ara.com://services/<svcId>\_<svcInId>"  <svcId>：是从DdsServiceInterfaceDeployment获取的service id。  <svcInId>：是从DdsServiceInterfaceDeployment获取的serviceInstanceId。  Subscriber和Publisher可以被服务接口的其他event/资源重用，所以应该在DomainParticipant销毁的时候销毁Subscriber和Publisher。 |
| **Upstream ID** | SWS\_CM\_11002 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | DomainParticipant携带qos应该与配置的一致。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00004] 为服务接口的每一个event赋予一个topic和DataWriter

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00004 |
| **Type** | Valid |
| **Description** | DDS绑定对服务接口内的每一个event赋予一个topic和DataWriter，因为这些topic可能已经在其他服务接口使用了，应该先在DomainParticipant内查找符合的topic，如果没查到，才会创建一个新的topic。  一旦服务接口内的topic可用，DomainParticipant使用publiser创建DataWriter，DataWriter应根据DdsEventQosProps进行配置。  Topic对象可以在其他服务实例中使用，所以在销毁DomainParticipant时才会销毁topic。 |
| **Upstream ID** | SWS\_CM\_11003 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | Rtps相关联的event携带的qos信息是否与配置的一致。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00005] 为服务接口的每一个method赋予一个request和reply topic，和DataWriter和DataReader

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00005 |
| **Type** | Valid |
| **Description** | DDS绑定应该实例化一个服务处理服务接口的所有方法，这意味着为服务赋予一个request topic，一个reply topic，因为这些topic可能在其他服务实例中使用，应该在创建前查找是否有可用的topic，如果没有查到，才会创建新的request和reply topic，一旦request和reply topic可用，DomainParticipant会创建DataReader，用于处理来自Subscriber的请求。创建DataWriter，处理publisher所有的应答。  topic对象可以被其他实例重用，所以在销毁DomainParticipant时才会销毁topic。 |
| **Upstream ID** | SWS\_CM\_11029 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 如果服务接口配置了method，会发布/订阅相关的request和reply topic。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00006] 为服务接口的每一个hasNotify为true的field赋予一个topic和DataWriter

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00006 |
| **Type** | Valid |
| **Description** | 因为topic可能在其他服务实例中使用，所以在创建前应该查询该topic是否已经存在，如果不存在，才会创建。  一旦服务接口的所有topic可用，服务接口的DomainParticipant应该创建DataWriter，DataWriter应根据DdsFieldQosProps创建。  主题对象可以被其他服务实例使用，所以在销毁DomainParticipant时才会销毁topic。 |
| **Upstream ID** | SWS\_CM\_11030 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | Field相关联的writer会被创建，rtps携带的信息应该与配置的一致。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00007] 为服务接口的所有field的 getter和setter赋予一个request， reply topic和DataWriter和DataReader

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00007 |
| **Type** | Valid |
| **Description** | DDS绑定应该为所有hasGetter and/or hasSetter为true的field关联一个request， reply topic和DataWriter和DataReader。  因为topic可能已经被使用，所以在创建topic对象之前应先检查该topic是否已经存在，如果不存在，才会创建request， reply topic。  一旦所有request， reply topic可用，DomainParticipant会创建一个DataReader，用于处理来自Subscriber所有请求。一个DataWriter，用户Publisher回复的所有应答。  topic对象可以被其他实例重用，所以在销毁DomainParticipant时才会销毁topic。 |
| **Upstream ID** | SWS\_CM\_11031 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 创建DataWriter和DataReader。相应的RTPS报文会被发送。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00008]OfferService函数映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00008 |
| **Type** | Valid |
| **Description** | DDS绑定应该通过使用DDS发现机制为服务实例的DomainParticipant传播Service IDs, Service Instance IDs,和ServiceInterface Contract Versions。USER\_DATA QoS在服务发现报文中增加自定义消息实现。自定义消息可以被ara::com clients和DDS本地应用使用，用于识别服务实例。  Service IDs, Service Instance IDs, 和ServiceInterface Contract Versions在USER\_DATA QoS进行如下编码：  "ara.com://services/<svcId>\_<svcInId>-<svcMajVersion>.<svcMinVersion>[&<svcId>\_<svcInId>-<svcMajVersion>.<svcMinVersion>]\*"  <svcId> 是从DdsServiceInterfaceDeployment配置的服务ID.  <svcInId> 是从DdsProvidedServiceInstance配置的服务实例ID.  <svcMajVersion> 是服务接口的主版本号.  <svcMinVersion>服务接口的次版本号.  因为DomainParticipant可以关联多个服务实例，允许多个<svcId>\_<svcInId><svcMajVersion>.<svcMinVersion>对出现。  如果USER\_DATA QoS是空的，DDS绑定应该设置它为"ara.com://services/<svcId>\_<svcInId>-<svcMajVersion>.<svcMinVersion>"，如果USER\_DATA不为空，应该尾部增加 "&<svcId>\_<svcInId>-<svcMajVersion>.<svcMinVersion>" |
| **Upstream ID** | SWS\_CM\_09004 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 相关的DomainParticipant的USER\_DATA QoS应该与描述的一致。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00009] 通过域参与者Qos的StopOfferService方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00009 |
| **Type** | Valid |
| **Description** | 在DomainParticipant的USER\_DATA QoS删除相应Service id和Instance id。  删除所有event关联的DataWriter  删除所有trigger关联的DataWriter  删除所有method关联的DataWriter和DataReader  删除所有field（hasNotifier为true）关联的DataWriter  删除所有field（hasGetter and/or hasSetter）关联的DataWriter和DataReader。 |
| **Upstream ID** | SWS\_CM\_11005 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 删除相关的DataWriter和DataReader，相关报文会通过rtps发送出来。 |
| **Risk** | 无 |
| **Change Type** | 变更 |

### [SWRD\_CM\_DDS\_00010] 在client端查找一个合适的DomainParticipant

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00010 |
| **Type** | Valid |
| **Description** | DDS绑定应该提供一个DomainParticipant处理服务发现和与远端DomainParticipant进行通信。DomainParticipant的配置在TPS\_ManifestSpecification描述：  domain id：从DdsRequiredServiceInstance配置的domain id而来。  QoS Profile：从DdsRequiredServiceInstance配置的qosProfile而来。 |
| **Upstream ID** | SWS\_CM\_11007 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 如果已经有相关的DomainParticipant，不会创建新的DomainParticipant。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00011] 在client端创建一个合适的DomainParticipant

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00011 |
| **Type** | Valid |
| **Description** | DDS绑定应该提供一个DomainParticipant处理服务发现和与远端DomainParticipant进行通信。DomainParticipant的配置在TPS\_ManifestSpecification描述：  domain id：从DdsRequiredServiceInstance配置的domain id而来。  QoS Profile：从DdsRequiredServiceInstance配置的qosProfile而来。 |
| **Upstream ID** | SWS\_CM\_11008 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 创建DomainParticipant。配置为上位机配置的qos。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00012] 通过创建或者查找到的DomainParticipants来发现remote service

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00012 |
| **Type** | Valid |
| **Description** | 为了获取服务实例的列表，DDS绑定应该迭代发现的每个remote DomainParticipant，通过在DomainParticipant内建的关联topic为DCPSParticipant的DataReader调用read方法。DCPSParticipant是标准的topic，用于通知其他网络中DomainParticipants他们已经上线。DomainParticipant传播DomainParticipants的USER\_DATA QoS，这些消息包含了识别服务实例的所有必要信息。  DDS绑定应该分析remote DomainParticipants USER\_DATA QoS的内容，并检查相关服务实例是否满足：  如果requiredServiceInstanceId设置为ALL，应该返回一个新的handle包含所有的查找到的服务实例。USER\_DATA符合"ara.com://services/.\*<svcId>.\*"。  如果requiredServiceInstanceId设置不是ALL，只有满足如下的服务实例才会返回，"ara.com://services/.\*<svcId>\_<reqSvcInId>.\*"。  在任何情况，应该先评估服务实例的contract version，只有在下面条件都满足的情况下，才会返回一个新的handers：  1. 发现的服务实例的contract version与serviceInterfaceDeployment配置的接口兼容。  2. contract version不在DdsRequiredServiceInstance.blacklistedVersion黑名单中。  在返回handers之前，应该对于每个服务实例保证用于发现的DomainParticipants已经创建了一个Publisher和Subscriber。Publisher和Subscriber的Partition QoS应该包含如下的partition名称"ara.com://services/<svcId>\_<reqSvcInId>"。  如果绑定实现未找到具有上述要求的DDS publisher，则它将创建一个新的DDS publisher，并使用上面定义的分区名称配置publisher的Partition QoS。同样，如果找不到满足这些要求的DDS publisher，则应创建一个新的DDS publisher并进行相应的配置。  Subscriber和publisher对象可以在与远程服务实例相关联的代理之间重用；因此，在包含该域的DomainParticipant被销毁之前，不得删除它们。 |
| **Upstream ID** | SWS\_CM\_11009 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 如果没有合适的Subscriber和publisher，则创建相应的Subscriber和publisher。 |
| **Risk** | 依赖第三方dds，是否支持迭代查找。 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00013] StartFindService方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00013 |
| **Type** | Valid |
| **Description** | 当连续查找服务，应该执行以下操作：  查找已经存在的有能力查找远端服务的DomainParticipant，如果没有这样的DomainParticipant的存在，则创建一个。  定义一个BuiltinParticipantListener，用于在以下情况下调用FindServiceHandler：  1.发现新的匹配到的服务remote DomainParticipant  2.匹配到服务的remote DomainParticipant不在包含该服务。  3.匹配到服务的remote DomainParticipant不存在了  应该为DomainParticipant绑定一个BuiltinParticipantListener。 |
| **Upstream ID** | SWS\_CM\_11010 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码，并验证API功能。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00014] BuiltinParticipantListener的定义

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00014 |
| **Type** | Valid |
| **Description** | DDS绑定应该定义类BuiltinParticipantListener，当发现remote DomainParticipant时，处理事件通知，该类应该继承DataReaderListener类，定义采样类型为ParticipantBuiltinTopicData和内建的topic（DCPSParticipant），BuiltinParticipantListener需要实现如下的方法：  构造函数，两个输入参数（引用），FindServiceHandler和requiredServiceInstanceId，这些引用应该被成员变量的形式存储，以便可以在后续的on\_data\_available使用，当发现新的DomainParticipant会调用该方法。  on\_data\_available：在调用FindServiceHandler时将成员变量requiredServiceInstanceId作为参数，如果返回的ServiceHandleContainer包含多个元素，调用on\_data\_available是应该将container作为参数。否则，不执行任何操作。 |
| **Upstream ID** | SWS\_CM\_11011 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00015] 为DomainParticipant绑定一个BuiltinParticipantListener

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00015 |
| **Type** | Valid |
| **Description** | 绑定应该通过参数FindServiceHandler和requiredServiceInstanceId创建一个新的BuiltinParticipantListener对象，该服务后续会调用set\_listener()， StatusMask = DATA\_AVAILABLE\_STATUS，将该listener绑定到DomainParticipant，当DomainParticipant销毁时，应该删除BuiltinParticipantListener。 |
| **Upstream ID** | SWS\_CM\_11012 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码。 |
| **Risk** | 依赖第三方dds，是否支持BuiltinParticipantListener。 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00016] StopFindService方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00016 |
| **Type** | Valid |
| **Description** | 当停止一个通过StartFindService建立的连续的查找，应该执行以下的操作：  查找满足条件的DomainParticipant，如果没有查找到DomainParticipant，StopFindService应该直接返回。  解除BuiltinParticipantListener和DomainParticipant绑定。 |
| **Upstream ID** | SWS\_CM\_11013 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码。 |
| **Risk** | 依赖第三方dds，是否支持BuiltinParticipantListener。 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00017] BuiltinParticipantListener和DomainParticipant解绑

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00017 |
| **Type** | Valid |
| **Description** | 应该使用StatusMask = STATUS\_MASK\_NONE调用set\_listener() |
| **Upstream ID** | SWS\_CM\_11014 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00018] event和topic的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00018 |
| **Type** | Valid |
| **Description** | topic应满足：  如果identifier\_type属性设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION或SERVICE\_INSTANCE\_RESOURCE\_INSTANCE\_ID，则主题名称应设置为：ara.com://services/<InterfaceID>/<Major>.<Minor>/< TopicName >  如果identifier\_type属性设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION则该主题的样本应通过 DataWriters和DataReader发送和接收，各自的父 Publisher 和 Subscriber QoS策略需包含：ara.com://services/<InterfaceID>/<InstanceId>  如果identifier\_type属性设置为SERVICE\_INSTANCE\_TOPIC\_PREFIX主题名称应该设置为：ara.com://services/<InterfaceID>/<InstanceID>/<TopicName>  定义topic data type，应该以等效的data type名称注册。 |
| **Upstream ID** | SWS\_CM\_11015 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 会创建该topic实体。 |
| **Risk** | 无 |
| **Change Type** | 变更 |

### [SWRD\_CM\_DDS\_00019] event topic data type定义

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00019 |
| **Type** | Valid |
| **Description** | topic data types定义符合如下IDL定义：  1 struct <eventTypeName>EventType {  2 @key uint16 instanceIdentifier;  3 <eventTypeName> data;  4 };  eventTypeName 是event关联的数据类型  instanceIdentifier是这个类型的key，用于从所有的采样中识别相同instanceIdentifier的采样。  Data是这个event的真实类型，根据DDS序列化进行编码和解码。 |
| **Upstream ID** | SWS\_CM\_11016 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00020] event Subscribe方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00020 |
| **Type** | Valid |
| **Description** | DDS绑定应该用Subscriber创建一个DataReader。 |
| **Upstream ID** | SWS\_CM\_11018 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 相应的创建DataReader。相关报文会通过rtps发送出来。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00021] 为event订阅创建DataReader

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00021 |
| **Type** | Valid |
| **Description** | 为了保证proxy只能与绑定的Service Instance通信，应该使用Subscriber（partition name为"ara.com://services/<svcId>\_ <reqSvcInId>"，创建DataReader  DataReader应该如下配置：  DataReaderQos qosProfile应该为DdsEventQosProps指定的Qos，Cache Size为根据Subscribe方法调用的输入参数。在qosProfile中的HISTORY QoS应该向下面这样被改写：  history.kind = KEEP\_LAST\_HISTORY\_QOS  history.depth = <maxSampleCount>  Listener 为DataReaderListener的实例.  StatusMask应该为STATUS\_MASK\_NONE. |
| **Upstream ID** | SWS\_CM\_11019 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码，相应的创建DataReader报文将会发送。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00022] event DataReaderListener的定义

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00022 |
| **Type** | Valid |
| **Description** | DDS绑定应该定义DataReaderListener，处理当订阅状态改变或者有数据到达时的事件处理。该类继承DataReaderListener。  DataReaderListener应该实现以下的方法：  构造函数，参数为EventReceiveHandler和SubscriptionStateChangeHandler的引用。  on\_data\_available()：如果EventReceiveHandler被设置，当数据达到时，则调用EventReceiveHandler  on\_subscription\_matched()：如果SubscriptionStateChangeHandler被设置，调用GetSubscriptionState()，并将结果传递给SubscriptionStateChangeHandler。  set\_event\_receive\_handler()：入参为EventReceiveHandler的引用，用于更新EventReceiveHandler成员  set\_subscription\_state\_change\_handler()：入参为SubscriptionStateChangeHandler的引用，更新SubscriptionStateChangeHandler的成员。 |
| **Upstream ID** | SWS\_CM\_11020 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00023] event GetSubscriptionState方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00023 |
| **Type** | Valid |
| **Description** | DDS绑定应该检查相关的DataReader是否存在，  如果存在，应该调用DataReader的get\_subscription\_matched\_status()函数。  如果SubscriptionMatchedStatus的total\_count 大于0，GetSubscriptionState() 应该返回 SubscriptionState = kSubscribed  否则，返回SubscriptionState = kSubscriptionPending  如果DataReader不存在，意味着Subscribe()没有被调用过，或者Unsubscribe()之前被调用过，GetSubscriptionState () 应该返回 SubscriptionState = kNotSubscribed。 |
| **Upstream ID** | SWS\_CM\_11022 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码并按照API需求进行验证。 |
| **Risk** | 有 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00024] event GetNewSamples方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00024 |
| **Type** | Valid |
| **Description** | DDS绑定应该在DataReader上执行take(),方法如下：  如果指定了maxNumberOfSamples，take应该指定max\_samples = maxNumberOfSamples。  如果没有指定maxNumberOfSamples，take则不指定max\_samples的限制。  在调用take之后，在每个有效的data采样后，应该调用可调用对象 f。 |
| **Upstream ID** | SWS\_CM\_11023 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码，并按照API需求进行验证。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00025] event GetFreeSampleCount方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00025 |
| **Type** | Valid |
| **Description** | DDS绑定返回DataReader中空闲槽的数目。 |
| **Upstream ID** | SWS\_CM\_11024 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 按照API需求进行验证。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00026] event SetReceiveHandler方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00026 |
| **Type** | Valid |
| **Description** | DDS绑定实行一下操作：  通过DataReader的 get\_listener()方法获取listener  通过调用set\_event\_receive\_handler()，通知listener，当数据到达时调用EventReceiveHandler函数  通过set\_listener()方法更新listener，StatusMask 设置如下:  如果原始StatusMask是STATUS\_MASK\_NONE或者  DATA\_AVAILABLE\_STATUS, 则将它设置为DATA\_AVAILABLE\_STATUS.  如果原始StatusMask是 SUBSCRIPTION\_MATCHED\_STATUS, 则设置为  DATA\_AVAILABLE\_STATUS|SUBSCRIPTION\_MATCHED\_STATUS.  如果原始StatusMask是DATA\_AVAILABLE\_STATUS|SUBSCRIPTION\_MATCHED\_STATUS, 则设置为 DATA\_AVAILABLE\_STATUS|SUBSCRIPTION\_MATCHED\_STATUS. |
| **Upstream ID** | SWS\_CM\_11025 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码，并验证该函数功能。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00027] event UnsetReceiveHandler方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00027 |
| **Type** | Valid |
| **Description** | DDS应该执行一下操作：  通过 get\_listener()获取DataReader的 listener引用  使用set\_event\_receive\_handler()撤销EventReceiveHandler  调用 set\_listener()更新listener，StatusMask按照如下配置：  如果原始StatusMask 是 STATUS\_MASK\_NONE 或者 DATA\_AVAILABLE\_STATUS, 则设置为 STATUS\_MASK\_NONE.  如果原始StatusMask 是 UBSCRIPTION\_MATCHED\_STATUS, 则设置为 SUBSCRIPTION\_MATCHED\_STATUS.  如果原始StatusMask 是DATA\_AVAILABLE\_STATUS|SUBSCRIPTION\_MATCHED\_STATUS, 则设置为 SUBSCRIPTION\_MATCHED\_STATUS. |
| **Upstream ID** | SWS\_CM\_11026 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00028] event SetSubscriptionStateHandler方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00028 |
| **Type** | Valid |
| **Description** | DDS绑定应该执行一下操作：  通过get\_listener()获取DataReader的listener  使用set\_subscription\_state\_change\_handler()方法，以便在SubscriptionMatchedStatus发生变化时，能够使用新的SubscriptionStateChangeHandler  通过set\_listener()设置新的listener，StatusMask应按照如下配置：  如果原始StatusMask 为 STATUS\_MASK\_NONE 或者 SUBSCRIPTION\_MATCHED\_STATUS, 则设置为SUBSCRIPTION\_MATCHED\_STATUS.  如果原始StatusMask 为 DATA\_AVAILABLE\_STATUS, 则设置为 DATA\_AVAILABLE\_STATUS|SUBSCRIPTION\_MATCHED\_STATUS.  如果原始StatusMask 为 DATA\_AVAILABLE\_STATUS|SUBSCRIPTION\_MATCHED\_STATUS, 则设置为 DATA\_AVAILABLE\_STATUS|SUBSCRIPTION\_MATCHED\_STATUS. |
| **Upstream ID** | SWS\_CM\_11027 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00029] UnsetSubscriptionStateHandler方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00029 |
| **Type** | Valid |
| **Description** | DDS绑定应该执行一下操作：  通过get\_listener()获取获取DataReader的listener  通过set\_subscription\_state\_change\_handler()取消之前设置的SubscriptionStateChangeHandler  通过set\_listener()设置新的listener，StatusMask如下进行配置：  如果原始StatusMask 为 STATUS\_MASK\_NONE or SUBSCRIPTION\_MATCHED\_STATUS, 则设置为 STATUS\_MASK\_NONE.  如果原始StatusMask 为 DATA\_AVAILABLE\_STATUS, 则设置为 DATA\_AVAILABLE\_STATUS.  如果原始StatusMask 为 DATA\_AVAILABLE\_STATUS|SUBSCRIPTION\_MATCHED\_STATUS, 则设置为 DATA\_AVAILABLE\_STATUS. |
| **Upstream ID** | SWS\_CM\_11028 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00030] method到DDS method和topic的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00030 |
| **Type** | Valid |
| **Description** | 相应的DDS服务应该提供服务接口的所有方法。  DDS服务应该由 Basic Service Mapping Profile of the RPC over DDS构建，在每个DDS服务赋予两个topic：request topic和reply topic。所以在每个包含多个method的服务接口中应该触发相应topic的创建。  Request topic如下进行配置：  request topic名称应满足：  如果identifier\_type属性设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION或SERVICE\_INSTANCE\_RESOURCE\_INSTANCE\_ID，则主题名称应设置为：ara.com://services/<InterfaceID>/<Major>.<Minor>/< TopicName >  如果identifier\_type属性设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION则该主题的样本应通过 DataWriters和DataReader发送和接收，各自的父 Publisher 和 Subscriber QoS策略需包含：ara.com://services/<InterfaceID>/<InstanceId>  如果identifier\_type属性设置为SERVICE\_INSTANCE\_TOPIC\_PREFIX主题名称应该设置为：ara.com://services/<InterfaceID>/<InstanceID>/<TopicName>  Request Topic Data Type是由SWS\_CM\_11101定义，被注册进DDS中。  Reply topic如下进行配置：  Reply topic名称应该满足：  如果identifier\_type属性设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION或SERVICE\_INSTANCE\_RESOURCE\_INSTANCE\_ID，则主题名称应设置为：ara.com://services/<InterfaceID>/<Major>.<Minor>/< TopicName >  如果identifier\_type属性设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION则该主题的样本应通过 DataWriters和DataReader发送和接收，各自的父 Publisher 和 Subscriber QoS策略需包含：ara.com://services/<InterfaceID>/<InstanceId>  如果identifier\_type属性设置为SERVICE\_INSTANCE\_TOPIC\_PREFIX主题名称应该设置为：ara.com://services/<InterfaceID>/<InstanceID>/<TopicName>  Reply topic data type由SWS\_CM\_11102定义，并被注册进DDS中。 |
| **Upstream ID** | SWS\_CM\_11100 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 创建相关的topic。 |
| **Risk** | 无 |
| **Change Type** | 变更 |

### [SWRD\_CM\_DDS\_00031] Request Topic data type定义

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00031 |
| **Type** | Valid |
| **Description** | Request Topic data type是一个由Request Header和data组成的结构体。IDL定义如下  1 struct <svcId>Method\_Request {  2 dds::rpc::RequestHeader header;  3 <svcId>Method\_Call data;  4 };  <svcId> 是服务接口ID.  dds::rpc::RequestHeader 是标准的Request header  <svcId>Method\_Call一个联合体，包含相应方法的输入参数。DDS绑定应该设置用字符串表示的serviceInstanceId设置instanceName（RequestHeader的一个成员）  <svcId>Method\_Call按照如下方式构建：  联合体的名称为<svcId>Method\_Call  联合体的区分值为32bit有符号数  联合体对于不支持的操作有一个默认的case dds::rpc::UnknownOperation  联合体应该对于服务接口的每一个method有一个case标签  case标签的值应该为由method 名称hash出来的32bit的整形值，哈希值的算法为7.5.1.1.2规定。服务接口在IDL中定义一个32bit有符号常量，用来简化联合体的表示。  每个case标签的名称应该为method的名称。  每个case标签的类型应该为<svcId>Method\_<methodName>\_In（7.5.1.1.4指导如何创建）  联合体<svcId>Method\_Call IDL定义如下  1 union <svcId>Method\_Call switch(int32) {  2 default:  3 dds::rpc::UnknownOperation unknownOp;  4 case <svcId>Method\_<method0Name>\_Hash:  5 <svcId>Method\_<method0Name>\_In <method0Name>;  6 case <svcId>Method\_<method1Name>\_Hash:  7 <svcId>Method\_<method1Name>\_In <method1Name>;  8 // ...  9 case <svcId>Method\_<methodNName>\_Hash:  10 <svcId>Method\_<methodNName>\_In <methodNName>;  11 };  <svcId>Method\_<methodName>\_In结构体应该包含方法的method in inout参数，<svcId>Method\_<methodName>\_In的表示如下：  1 struct <svcId>Method\_<methodName>\_In { 2 <ArgumentDataPrototype[0]>;  3 <ArgumentDataPrototype[1]>;  4 // ...  5 <ArgumentDataPrototype[n]>;  6 };  如果method没有输入参数，<svcId>Method\_<methodName>\_In应该有一个类型为dds::rpc::UnusedMember名称为dummy成员。  Request Topic data type应该按照DDS序列化规则进行序列化。 |
| **Upstream ID** | SWS\_CM\_11101 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00032] Reply Topic data type定义

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00032 |
| **Type** | Valid |
| **Description** | Reply Topic data type是由 Reply Header和和返回的结构组成，IDL定义如下：  1 struct <svcId>Method\_Reply {  2 dds::rpc::ReplyHeader header;  3 <svcId>Method\_Return data;  4 };  <svcId> 为服务接口ID.  dds::rpc::ReplyHeader 为7.5.1.1.1定义应答头.  <svcId>Method\_Return为联合体，存有应答的返回值。  dds::rpc::ReplyHeader由7.5.1.1.1定义。  <svcId>Method\_Return由7.5.1.1.7定义：  联合体的名称为<svcId>Method\_Return。  联合体的区分值为32bit有符号类型。  联合体为了表示不支持的操作或者未知的操作有一个默认的case，类型为dds::rpc::UnknownOperation。  联合体为服务接口的每一个method都有一个case标签，标签的值为method名称哈希出来的32bit有符号值。hash值得算法由7.5.1.1.2指定，服务接口应该定义32bit的常量用来简化联合体的表示，case标签应该为method的名称，每个标签的类型为<svcId>Method\_<methodName>\_Result。<svcId>Method\_Return IDL表示如下:  1 union <svcId>Method\_Return switch(int32) {  2 default:  3 dds::rpc::UnknownOperation unknownOp;  4 case <svcId>Method\_<method0Name>\_Hash:  5 <svcId>Method\_<method0Name>\_Result <method0Name>;  6 case <svcId>Method\_<method1Name>\_Hash:  7 <svcId>Method\_<method1Name>\_Result <method1Name>;  8 // ...  9 case <svcId>Method\_<methodNName>\_Hash:  10 <svcId>Method\_<methodNName>\_Result <methodNName>  11 };  <svcId>Method\_<methodName>\_Result 按照如下方式构建：  联合体的区分值应该为32bit有符号数。  联合体有一个dds::RETCODE\_OK的标签，代表一个成功的返回。  RETCODE\_OK 的值为0x00，成功的分支下，应该一个类型为<svcId>Method\_<methodName>\_Out，名称为 result 的成员。  联合体应该有一个dds::RETCODE\_ERROR的标签，代表method返回的 ApApplicationError。  RETCODE\_ERROR的值为0x01；  在错误情况下，有一个类型为ara::core::ErrorCode，名称为error的成员。  <svcId>Method\_<methodName>\_Result IDL表示如下：  1 union <svcId>Method\_<methodName>\_Result switch(int32) {  2 case dds::RETCODE\_OK:  3 <svcId>Method\_<methodName>\_Out result;  4 case dds::RETCODE\_ERROR:  5 ara::core::ErrorCode error;  6 };  <svcId>Method\_<methodName>\_Out 构造如下：  应该包含每个method所有out inout的参数。  out inout参数的顺序应该与定义一致。  对于non-void方法，这个结构的最后一个成员名称为return\_,类型为method的返回类型。如果这个method有一个参数名为return\_，这个成员应该根据7.5.1.1.5重新命名，如果返回值的类型为 ara::core::Result<ValueType,ErrorType>，ValueType被认为是返回类型。  如果method没有返回值，没有out参数，没有intout参数，这个结构应该包含名称为dummy，类型为dds::rpc::UnusedMember 的成员。  <svcId>Method\_<methodName>\_Out的表示如下：  1 struct <svcId>Method\_<methodName>\_Out {  2 <ArgumentDataPrototype[0]>;  3 <ArgumentDataPrototype[1]>;  4 // ...  5 <ArgumentDataPrototype[n]>;  6 };  Reply Topic data type 应该按照DDS序列化。 |
| **Upstream ID** | SWS\_CM\_11102 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00033] ApApplicationError中ara::core::ErrorCode的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00033 |
| **Type** | Valid |
| **Description** | 应该表示成如下的IDL:  1 module dds {  2 module ara {  3 module core {  4  5 struct ErrorCode {  6 uint64 error\_domain\_value;  7 int32 error\_code;  8 };  error\_domain\_value为用于表示ApApplicationErrorDomain64位无符号整数。  error\_code是一个用于表示ApApplicationError的32bit有符号数。  ara::core::ErrorCode 应该根据DDS序列化规则进行序列化，因为IDL modules被转换为c++ namespace，顶层的dds阻止命名冲突。 |
| **Upstream ID** | SWS\_CM\_10431 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 如果server执行method出错，相应的错误结果会发送给client。报文按照序列化规则进行序列化。 |
| **Risk** | 无 |
| **Change Type** | 变更 |

### [SWRD\_CM\_DDS\_00034] 在client端创建DataWriter处理请求

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00034 |
| **Type** | Valid |
| **Description** | DDS绑定会为服务接口的代理实例Request Topic创建一个DataWriter。  如果identifier\_type 设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION，  为了保证代理只能与绑定的服务实例通信，DDS绑定只能使用SWS\_CM\_11009规定的Publiser，DataWriter应该按如下进行配置：  DataWriterQos应该使用DdsRequiredServiceInstance定义的qosProfile。 |
| **Upstream ID** | SWS\_CM\_11103 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 创建相关的DataWriter。相关报文会通过rtps发送出来。 |
| **Risk** | 无 |
| **Change Type** | 变更 |

### [SWRD\_CM\_DDS\_00035] 在client端创建DataReader处理method应答

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00035 |
| **Type** | Valid |
| **Description** | DDS绑定应该为服务接口Reply topic创建一个DataReader。  如果identifier\_type 设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION，  为了保证proxy只能与绑定的服务实例通信，应该使用之前创建的Subscriber创建DataReader。DataReader应该按照如下进行配置：  DataReaderQos应该使用DdsRequiredServiceInstance配置的qosProfile。 |
| **Upstream ID** | SWS\_CM\_11104 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 创建相关的DataReader。相关报文会通过rtps发送出来。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00036] 在server端创建一个DataReader处理method请求

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00036 |
| **Type** | Valid |
| **Description** | 作为OfferService的一部分，DDS绑定应该为服务接口的Request topic创建一个DataReader，如果identifier\_type 设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION，应该使用partition 为"ara.com://services/<svcId>\_<svcInId>"的subscriber创建DataReader。  DataReader应该如下进行创建：  DataReaderQos应该使用DdsProvidedServiceInstance定义的DataReaderQos。  Listener 和 StatusMask根据MethodCallProcessingMode进行如下配置：  MethodCallProcessingMode = kEvent 或者 kEventSingleThread，listener应该为DataReaderListener的一个实例，StatusMask设置为DATA\_AVAILABLE\_STATUS。  如果MethodCallProcessingMode = kPoll，Lisener保持不变，StatusMask设置为STATUS\_MASK\_NONE。 |
| **Upstream ID** | SWS\_CM\_11105 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 创建相关的DataReader。相关报文会通过rtps发送出来。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00037] 在server端创建DataWriter处理method应答

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00037 |
| **Type** | Valid |
| **Description** | 作为OfferService的一部分，DDS绑定应该为服务接口的Reply topic创建一个DataWriter，如果identifier\_type 设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION，需要使用partition为 "ara.com://services/<svcId>\_<svcInId>创建DataWriter。  DataWriter应按照如下配置：  DataWriterQos应该使用DdsProvidedServiceInstance定义的DataWriterQos。 |
| **Upstream ID** | SWS\_CM\_11106 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 创建相关的DataWriter。相关报文会通过rtps发送出来。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00038] 在client端调用服务方法

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00038 |
| **Type** | Valid |
| **Description** | 当在client调用方法时，DDS绑定应该构造一个新的Topic data type定义的Request Topic sample。构造方法如下：  实例化RequestHeader对象， 根据RPC over DDS设置一个requestId和远端服务实例的实例id的instanceName。  为了初始化<svcId>Method\_Call 对象，首先选择合适的联合体类别，然后设置相应的包含in和inout参数的结构。  数据会作为参数传给write()，write将调用DDS序列化并发送数据。 |
| **Upstream ID** | SWS\_CM\_11107 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 服务方法指定的数据会按照序列化后的数据发送出来。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00039] 通知client method的响应并将method结果通知client应用程序。

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00039 |
| **Type** | Valid |
| **Description** | DDS绑定应该调用ara::core::Promise的set\_value()或者SetError()，将结果返回给ara::core::Future。  如果method响应的<svcId>Method\_<methodName>\_Result 联合体的区分值为dds::RETCODE\_OK，DDS绑定应该调用ara::core::Promise 的 set\_value()方法，参数为代表out和inout的成员<svcId>Method\_<methodName>\_Out。  如果method响应的<svcId>Method\_<methodName>\_Result 联合体的区分值不为dds::RETCODE\_OK，DDS绑定应该调用ara::core::Promise 的 SetError()方法，参数为基于相应的ApApplicationError的ara::core::ErrorCode。  在任何情况，接收到reply topic数据都应该执行set操作。DDS绑定应该调用DataReader的take方法接收数据，除此之外，为了关联响应和应答，应该比较原始的requestId比较接收到的header.relatedRequestId，如果没有匹配，则应答被直接丢弃。 |
| **Upstream ID** | SWS\_CM\_11108 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 验证method调用正确性。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00040] 在server端处理以event driven处理方法调用

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00040 |
| **Type** | Valid |
| **Description** | 如果 MethodCallProcessingMode以kEvent 或者 kEventSingleThread 构造ServiceSkeleton，DDS绑定应该创建DataReaderListener处理异步请求，并将该DataReaderListener实例关联到DataReader，listener负责识别该方法，处理该请求。 |
| **Upstream ID** | SWS\_CM\_11109 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码。并验证event driven功能。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00041] 在server端创建DataReaderListener处理异步请求

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00041 |
| **Type** | Valid |
| **Description** | MethodCallProcessingMode 为kEvent 或者 kEventSingleThread需要实例化DataReaderListener处理异步请求，listener继承于 Request Topic相关联的DataReaderListener。  DataReaderListener应该实现以下的方法：  on\_data\_available()，负责从DataReader使用take()读取相应的请求进行处理，识别ServiceSkeleton一个方法，发送给相应method处理。on\_data\_available()应该使用<svcId> Method\_Call的区分值，并在联合体类别中使用具体的方法。 |
| **Upstream ID** | SWS\_CM\_11110 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码。验证kEvent 或者 kEventSingleThread功能。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00042] 在server端以polling处理一个方法

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00042 |
| **Type** | Valid |
| **Description** | 在这种情况下，ProcessNextMethodCall负责在Request Topic对应的DataReader调用take()，ProcessNextMethodCal应该在DataReader取走第一个数据，根据<svcId>Method\_Call区分值，使用相关的参数，触发相应的方法调用。 |
| **Upstream ID** | SWS\_CM\_11111 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00043] 在server端发送method应答

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00043 |
| **Type** | Valid |
| **Description** | 为了发送响应，DDS绑定应该构建一个新的Reply Topic data type的数据，构造方式如下：  实例化ReplyHeader对象， header.requestId配置为relatedRequestId  实例化<svcId>Method\_Return对象，DDS绑定应该实现：  选择合适联合体类型  根据相应方法的的结果或者异常选择联合体的区分值。  如果返回正确的结果，应该设置dds::RETCODE\_OK并设置使用所有out和inout参数构建的<svcId>Method\_ <methodName>\_Out结构。  否则触发ApApplicationError异常，设置联合体的区分值为0x01，并构建相应的ara::core::ErrorCode。  之后数据将作为参数传给write()发送。 |
| **Upstream ID** | SWS\_CM\_11112 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | Server端在处理完一个method后，可能会发送应答，应答按照IDL描述的内容并按照dds序列化进行发送。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00044] field的hasNotifier的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00044 |
| **Type** | Valid |
| **Description** | DDS绑定应该对于服务接口定义的每个field（hasNotifier= true）映射为topic，topic如下进行配置：  topic名称应该满足：  如果identifier\_type属性设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION或SERVICE\_INSTANCE\_RESOURCE\_INSTANCE\_ID，则主题名称应设置为：ara.com://services/<InterfaceID>/<Major>.<Minor>/< TopicName >  如果identifier\_type属性设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION则该主题的样本应通过 DataWriters和DataReader发送和接收，各自的父 Publisher 和 Subscriber QoS策略需包含：ara.com://services/<InterfaceID>/<InstanceId>  如果identifier\_type属性设置为SERVICE\_INSTANCE\_TOPIC\_PREFIX主题名称应该设置为：ara.com://services/<InterfaceID>/<InstanceID>/<TopicName>  Topic Data Type被注册进DDS中。 |
| **Upstream ID** | SWS\_CM\_11130 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 变更 |

### [SWRD\_CM\_DDS\_00045] Field Notifier DDS Topic data type 的定义

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00045 |
| **Type** | Valid |
| **Description** | DDS绑定应该按照如下的IDL定义 Field Notifier。  1 struct <fieldTypeName>FieldNotifierType {  2 @key uint16 instance\_id;  3 <fieldTypeName> data;  4 };  <fieldTypeName>是field引用的数据类型。  instance\_id是类型的中key，用于在所有采样中识别该服务实例的采样。  Data是field真实的值，根据DDS序列化进行编码和解码。  @external是可选的 |
| **Upstream ID** | SWS\_CM\_11131 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 创建相关的DataWriter。 |
| **Risk** | 无 |
| **Change Type** | 变更 |

### [SWRD\_CM\_DDS\_00046] Field Update方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00046 |
| **Type** | Valid |
| **Description** | 当需要发送field的消息时，DDS绑定应该构造一个类型为DDS Topic data type的数据，像下面这样：  实例id为DdsProvidedServiceInstance定义的serviceInstanceId。  Data应该被当作参数传递给相应DataReader的write函数。这个函数负责将序列化并将数据发送出去。 |
| **Upstream ID** | SWS\_CM\_11132 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 检查代码，采样数据会按照序列化发送出来。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00047] field Subscribe方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00047 |
| **Type** | Valid |
| **Description** | DDS绑定应该创建DataReader用于订阅。 |
| **Upstream ID** | SWS\_CM\_11133 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 创建DataReader，相关报文会通过rtps发送出来。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00048] 为field订阅创建DataReader

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00048 |
| **Type** | Valid |
| **Description** | DDS绑定应该为field关联的topic创建DataReader。如果identifier\_type 设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION，确保根据[SWS\_CM\_11009] 创建的Subscriber来创建DataReader。  DataReader按照如下进行配置：  DataReaderQos为DdsFieldQosProps定义的qosProfile。DataReader的cache size为Subscribe的参数maxSampleCount。  HISTORY QoS会被重写：  history.kind = KEEP\_LAST\_HISTORY\_QOS  history.depth = <maxSampleCount>  除此之外，为了保证订阅后立刻能收到field的值，DURABILITY QoS应该被重写为：durability.kind = TRANSIENT\_LOCAL\_DURABILITY\_QOS。  durability.kind = TRANSIENT\_LOCAL\_DURABILITY\_QOS  RELIABILITY QoS应该被重写为：  reliability.kind = RELIABLE\_RELIABILITY\_QOS  Listener 应该为DataReaderListener的实例，StatusMask 应该为 STATUS\_MASK\_NONE. |
| **Upstream ID** | SWS\_CM\_11134 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 创建DataReader，相关报文会通过rtps发送出来。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00049] 为field订阅创建DataReaderListener

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00049 |
| **Type** | Valid |
| **Description** | DDS绑定应该创建DataReaderListener，当数据到达时处理field通知。 |
| **Upstream ID** | SWS\_CM\_11135 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00050]field Unsubscribe的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00050 |
| **Type** | Valid |
| **Description** | 当取消订阅field时，DDS绑定应该删除相关的DataReader。 |
| **Upstream ID** | SWS\_CM\_11136 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 删除DataReader，相关报文会通过rtps发送出来。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00051]field GetSubscriptionState方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00051 |
| **Type** | Valid |
| **Description** | 与event方法一致 |
| **Upstream ID** | SWS\_CM\_11137 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 有 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00052]field GetNewSamples方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00052 |
| **Type** | Valid |
| **Description** | 与event方法一致 |
| **Upstream ID** | SWS\_CM\_11138 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00053]field GetFreeSampleCount方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00053 |
| **Type** | Valid |
| **Description** | 与event方法一致 |
| **Upstream ID** | SWS\_CM\_11139 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00054]field SetReceiveHandler方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00054 |
| **Type** | Valid |
| **Description** | 与event方法一致 |
| **Upstream ID** | SWS\_CM\_11140 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00055]field UnsetReceiveHandler方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00055 |
| **Type** | Valid |
| **Description** | 与event方法一致 |
| **Upstream ID** | SWS\_CM\_11141 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00056]field SetSubscriptionStateHandler方法映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00056 |
| **Type** | Valid |
| **Description** | 与event方法一致 |
| **Upstream ID** | SWS\_CM\_11142 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00057]field UnsetSubscriptionStateHandler方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00057 |
| **Type** | Valid |
| **Description** | 与event方法一致 |
| **Upstream ID** | SWS\_CM\_11143 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00058] Field Get/Set方法到服务接口的服务方法和topic的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00058 |
| **Type** | Valid |
| **Description** | 为了client能够调用服务接口中配置了hasGetter和HasSetter的field的方法，需要将get/set映射为DDS服务和topic。DDS服务应该提供的所有field的getter和setter方法。DDS服务构建应该符合the Basic Service Mapping Profile of the RPC over DDS specification，所以服务接口中的getter/setter field都会触法一对topic的创建，request topic和reply topic。reqest topic应该按照如下进行构建：  reuest topic名称应该满足：  如果identifier\_type属性设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION或SERVICE\_INSTANCE\_RESOURCE\_INSTANCE\_ID，则主题名称应设置为：ara.com://services/<InterfaceID>/<Major>.<Minor>/< TopicName >  如果identifier\_type属性设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION则该主题的样本应通过 DataWriters和DataReader发送和接收，各自的父 Publisher 和 Subscriber QoS策略需包含：ara.com://services/<InterfaceID>/<InstanceId>  如果identifier\_type属性设置为SERVICE\_INSTANCE\_TOPIC\_PREFIX主题名称应该设置为：ara.com://services/<InterfaceID>/<InstanceID>/<TopicName>  Request Topic Data Type应该为SWS\_CM\_11145定义的一致。  reply topic名称应该满足：  如果identifier\_type属性设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION或SERVICE\_INSTANCE\_RESOURCE\_INSTANCE\_ID，则主题名称应设置为：ara.com://services/<InterfaceID>/<Major>.<Minor>/< TopicName >  如果identifier\_type属性设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION则该主题的样本应通过 DataWriters和DataReader发送和接收，各自的父 Publisher 和 Subscriber QoS策略需包含：ara.com://services/<InterfaceID>/<InstanceId>  如果identifier\_type属性设置为SERVICE\_INSTANCE\_TOPIC\_PREFIX主题名称应该设置为：ara.com://services/<InterfaceID>/<InstanceID>/<TopicName>  reply Topic Data Type应该为SWS\_CM\_11146定义的一致。 |
| **Upstream ID** | SWS\_CM\_11144 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 创建相关的topic。 |
| **Risk** | 无 |
| **Change Type** | 变更 |

### [SWRD\_CM\_DDS\_00059] Field getter 和 setter中Request Topic data type 定义

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00059 |
| **Type** | Valid |
| **Description** | Request Topic data type是一个组合的结构包含了一个Request Header和data的数据。IDL定义如下：  1 struct <svcId>Field\_Request {  2 dds::rpc::RequestHeader header;  3 <svcId>Field\_Call data;  4 };  <svcId>是对应的服务接口ID。  dds::rpc::RequestHeader是DDS规范定义的请求报文头。  <svcId>Field\_Call是一个联合体，存储着相应method的输入参数。  dds::rpc::RequestHeader应该根据DDS规范构建，DDS绑定应该设为instanceName（RequestHeader中的一个成员）为serviceInstanceId的字符串表示。  <svcId>Field\_Call按照如下方式构建：  联合体的名称为<svcId>Field\_Call；  联合体的区分值为32位有符号整形；  联合体应该对不支持的操作有一个默认的情形 dds::rpc::UnknownOperation；  联合体应该对于每一个 hasGetter 和 hasSetter 为 true的field有一个cast标签。这个标签的值为32bit的field的getter和setter名称的hash值。也就是 "Get<fieldName>" 和 "Set<fieldName>"， <fieldName> 是 Field的名称。DDS绑定应该根据DDS规范的hash算法计算。  Field\_Get<fieldName>\_Hash 或者 const int32 <svcId>Field\_Set <fieldName>\_Hash) 用来简化 union cases的表示。  case label的成员名对于getter应该为get<FieldName>，对于setter应该为 set<FieldName>。  每种case的类型名对于getter为<svcId>Field\_Get<fieldName>\_In，对于setter为<svcId>Field\_Set<fieldName>\_In  <svcId>Field\_Call 联合体IDL定有如下:  1 union <svcId>Field\_Call switch(int32) {  2 default:  3 dds::rpc::UnknownOperation unknownOp;  4 case <svcId>Field\_Get<Field0Name>\_Hash:  5 <svcId>Field\_Get<Field0Name>\_In get<Field0Name>;  6 case <svcId>Field\_Set<Field0Name>\_Hash:  7 <svcId>Field\_Set<Field0Name>\_In set<Field0Name>;  8 case <svcId>Field\_Get<Field1Name>\_Hash:  9 <svcId>Field\_Get<Field1Name>\_In get<Field1Name>;  10 case <svcId>Field\_Set<Field1Name>\_Hash:  11 <svcId>Field\_Set<Field1Name>\_In set<Field1Name>;  12 // ...  13 case <svcId>Field\_Get<FieldNName>\_Hash:  14 <svcId>Field\_Get<FieldNName>\_In get<FieldNName>;  15 case <svcId>Field\_Set<FieldNName>\_Hash:  16 <svcId>Field\_Set<FieldNName>\_In set<FieldNName>;  17 };  <svcId>Field\_Set<FieldName>\_In structures 应该包含设置的数据类型<svcId>Field\_Get<FieldName>\_In 应该包含获取的相关类型。 |
| **Upstream ID** | SWS\_CM\_11145 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00060] 对于Field getter 和 setter Reply Topic data type 的定义

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00060 |
| **Type** | Valid |
| **Description** | Reply Topic data type是一个组合类型，由 Reply Header和真实的数据类型构成。Reply Topic data type的IDL定义如下：  1 struct <svcId>Field\_Reply {  2 dds::rpc::ReplyHeader header;  3 <svcId>Field\_Return data;  4 }  <svcId> 为 serviceInterfaceId；  <svcId>Field\_Return是一个联合体，存储着相应应答的返回值。  dds::rpc::ReplyHeader有DDS规范定义；  <svcId>Field\_Return 按照如下定义：  联合体的名称为<svcId>Field\_Return；  联合体的区分值为32位有符号数；  联合体应该对于不支持的操作有一个默认的case，类型为 dds::rpc::UnknownOperation。  联合体对于每一个getter和setter都有一个case对应：  每一个标签对应一个Getter或者setter名称。"Get<FieldName>" 和 "Set<FieldName>"，<FieldName>为field的名称。DDS绑定应该根据DDS规范定义的hash算法计算出32bit的有符号数，并用Field\_Get<FieldName>\_Hash 或者 const int32 <svcId>Field\_Set <FieldName>\_Hash)常量表示。用于简化case的表示。  每一个case label名称对于getter为get<FieldName>， 对于setter为set<FieldName>。  每个case label的类型对于getter为<svcId>Field\_Get<FieldName>\_Result，对于setter为 <svcId>Field\_Set<FieldName>\_Result  <svcId>Field\_Return的IDL如下：  1 union <svcId>Field\_Return switch(int32) {  2 default:  3 dds::rpc::UnknownOperation unknownOp;  4 case <svcId>Field\_Get<Field0Name>\_Hash:  5 <svcId>Field\_Get<Field0Name>\_Result get<Field0Name>;  6 case <svcId>Field\_Set<Field0Name>\_Hash:  7 <svcId>Field\_Set<Field0Name>\_Result set<Field0Name>;  8 case <svcId>Field\_Get<Field1Name>\_Hash:  9 <svcId>Field\_Get<Field1Name>\_Result get<Field1Name>;  10 case <svcId>Field\_Set<Field1Name>\_Hash:  11 <svcId>Field\_Set<Field1Name>\_Result set<Field1Name>;  12 // ...  13 case <svcId>Field\_Get<FieldNName>\_Hash:  14 <svcId>Field\_Get<FieldNName>\_Result get<FieldNName>;  15 case <svcId>Field\_Set<FieldNName>\_Hash:  16 <svcId>Field\_Set<FieldNName>\_Result set<FieldNName>;  17 };  getters 和 setters 应该有相同的output parameter。 因此,<svcId>Field\_Get<FieldName>\_Result 和 <svcId> Field\_Set<FieldName>\_Result 构建如下:  联合体的区分值为 32-bit 有符号整形。  联合体有一个case 为 dds::RETCODE\_OK 用来表达成功的返回:  RETCODE\_OK 应该为 0  成功的case 应该有一个成员类型为 <svcId>Field\_Get<FieldName>\_Out 的result\_ ，用于存储getter的返回值,后者类型为<svcId>Field\_Set<FieldName>\_Out存储setter的返回值。  <svcId>Field\_Get<FieldName>\_Result IDL如下:  1 union <svcId>Field\_Get<FieldName>\_Result switch(int32) {  2 case dds::RETCODE\_OK:  3 <svcId>Field\_Get<FieldName>\_Out result\_;  4 };  否则, <svcId>Field\_Set<FieldName>\_Result IDL为:  1 union <svcId>Field\_Set<FieldName>\_Result switch(int32) {  2 case dds::RETCODE\_OK:  3 <svcId>Field\_Set<FieldName>\_Out result\_;  4 };  <svcId>Field\_Get<FieldName>\_Out 和Field\_Set<FieldName>\_Out 应该映射到return\_的成员上。  Reply Topic data type 应该根据 DDS serialization 规则序列化。 |
| **Upstream ID** | SWS\_CM\_11146 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00061] 为client端get和set请求创建DataWriter

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00061 |
| **Type** | Valid |
| **Description** | DDS绑定应该在proxy实例化时为field中的get/set关联的request topic创建DataWriter。  为了保证代理只与绑定的服务实例通信，需要由使用partition为ara.com://services/<svcId>\_<reqSvcInId>的publisher创建。  DataWriter如下进行配置：  DataReaderQos应该使用DdsFieldQosProps指定的qosProfile。 |
| **Upstream ID** | SWS\_CM\_11147 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 创建相应的DataWriter。相关报文会通过rtps发送出来。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00062] 在client端为Field get/set的应答创建DataReader

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00062 |
| **Type** | Valid |
| **Description** | DDS绑定应该在proxy实例化时，为服务接口中的field中的getter和setter关联的Reply Topic创建DataReader。  为了保证代理只与绑定的服务实例通信，需要由使用partition为ara.com://services/<svcId>\_<reqSvcInId>的publisher创建。  DataReader如下进行配置：  DataReaderQos应该使用DdsFieldQosProps指定的qosProfile。 |
| **Upstream ID** | SWS\_CM\_11148 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 创建相应的DataReader。相关报文会通过rtps发送出来。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00063] 在server端为Field set/get请求创建DataReader

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00063 |
| **Type** | Valid |
| **Description** | DDS绑定应该为关联的 getters 和 setters 的request topic创建 DataReader。应该使用partition为"ara.com://services/<svcId>\_<svcInId>") 的publisher创建 DataReader。  DataReader因按照如下配置：  DataReaderQos应该配置为DdsFieldQosProps定义的qosProfile。  Listener 和 StatusMask应该根据MethodCallProcessingMode设置：  对于 MethodCallProcessingMode = kEvent 或者 kEventSingleThread，Listener应该被设置为DataReaderListener的一个实例，StatusMask应该配置为DATA\_AVAILABLE\_STATUS；  对于MethodCallProcessingMode = kPoll，Listener应该保持不变，StatusMask 设置为 STATUS\_MASK\_NONE。 |
| **Upstream ID** | SWS\_CM\_11149 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00064] 在server端为Field get/set应答创建一个DataWriter

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00064 |
| **Type** | Valid |
| **Description** | DDS绑定应该为服务接口的getter/setter关联的Reply Topic创建一个DataWriter。  应该使用partition为"ara.com://services/<svcId>\_<svcInId>"的publisher创建DataWriter。  DataWriter应该按照如下配置：  DataWriterQos应该使用DdsFieldQosProps定义的qosProfile。 |
| **Upstream ID** | SWS\_CM\_11150 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 相应的partition和qos符合描述的内容。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00065] 在client端调用Field get/set方法

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00065 |
| **Type** | Valid |
| **Description** | DDS绑定应该构造类型为 Request Topic data type的实例，构造方法如下：  设置requestId（DDS规范指定）；  设置instanceName为远端服务实例的serviceInstanceId；  为了实例化<svcId>Field\_Call对象，DDS绑定应该首先选择一个合适联合体对象，然后，如果调用对应getter，DDS绑定应该保持<svcId>Field\_Get<FieldName>\_In的成员dummy不变。  如果调用对应setter，DDS绑定应该使用filed的新值设置<svcId>Field\_Set<FieldName>\_In中的唯一成员。  然后数据作为参数被传到write，数据将被序列化并发送。 |
| **Upstream ID** | SWS\_CM\_11151 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | Client调用get/set方法，相关数据将从网络发出。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00066] 当收到Field get/set方法的响应时通知client

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00066 |
| **Type** | Valid |
| **Description** | 为了通知client get/set响应的结果，DDS绑定应该调用set\_value()，对于get，参数为<svcId>Field\_Get<FieldName>\_Result的result\_成员，对于set， 参数为<svcId> Field\_Set<FieldName>\_Out的result\_成员。  通过在DataReader读取数据来应该执行相应set操作，DDS应该用DataReader中的take()方法处理数据，为了匹配请求，DDS绑定通过比较header.relatedRequestsId，如果没有查找到匹配的发送者，响应会被丢弃。 |
| **Upstream ID** | SWS\_CM\_11152 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码，并验证API相关功能。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00067] 在server端处理field相关联的get/set方法

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00067 |
| **Type** | Valid |
| **Description** | 如果MethodCallProcessingMode 为kEvent 或者 kEventSingleThread，DDS绑定应该创建DataReaderListener实例处理异步请求，listener负责识别该方法并处理请求。 |
| **Upstream ID** | SWS\_CM\_11153 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码并验证该方法调用。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00068] 在server端为Field getters和setters创建DataReaderListener用于处理异步请求。

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00068 |
| **Type** | Valid |
| **Description** | 对于MethodCallProcessingMode 为 kEvent 或者 kEventSingleThread，需要实例化一个DataReaderListener处理异步请求，listener应该派生于标准的DataReaderListener，处理的数据类型为 Request Topic data type，DataReaderListener应该实现如下的函数：  on\_data\_available：负责处理从DataReader使用take()获取请求，分发到相应的注册的SetHandler或者GetHandler。为了识别ServiceSkeleton中的field，field中get/set，on\_data\_available()应该使用 <svcId>Field\_Call的区分值。在set操作的情况下，应该提供相应的SetHandler，参数为接收到的 <svcId>Field\_<FieldName>\_In的唯一成员。如果是get，绑定将分派给相应的GetHandler（如果已注册），或者分派给该字段的当前值的内部查找操作（如果未注册）。 |
| **Upstream ID** | SWS\_CM\_11154 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00069] 在server端以polling处理Field get/set方法

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00069 |
| **Type** | Valid |
| **Description** | 如果MethodCallProcessingMode 为 kPoll，ProcessNextMethodCall方法负责调用在DataReader调用take，处理第一个请求。分发到相应的SetHandler。  为了正确识别field和操作，应该使用<svcId>Field\_Call的区分值，如果为set操作，应该提供相应的SetHandler方法，参数为<svcId>Field\_<FieldName>\_In唯一成员。如果是get操作，绑定将分派给相应的GetHandler（如果已注册），或者分派给该字段的当前值的内部查找操作（如果未注册）。 |
| **Upstream ID** | SWS\_CM\_11155 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 每次调用ProcessNextMethodCall只会触发一次方法调用。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00070] 在server端为Field get/set发送一个应答

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00070 |
| **Type** | Valid |
| **Description** | DDS绑定应该在调用SetHandler/GetHandler之后发送一个应答。发送一个应答，DDS绑定应该构建 Reply Topic data type数据并发送。  为了初始化ReplyHeader对象，header.requestId 应该被设置为relatedRequestId。为了初始化<svcId>Field\_Return对象，应该实现：  选择合适的union case，getter/setter method的hash值为区分值，对于get操作，设置相应的<svcId>Field\_Get<FieldName>\_Result，对于set,设置相应的<svcId>Field\_Set<FieldName>\_Result。在两种情况下，应该选择dds::RETCODE\_OK，并设置相应的值。  构造的应答将会作为参数传递给DataWriter，并调用write发送。 |
| **Upstream ID** | SWS\_CM\_11156 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | Client端调用get/set方法，相应的server端注册的SetHandler/GetHandler回被调用执行。在执行之后，会发送应答，应答按照IDL进行序列化。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00071] DDS标准序列化

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00071 |
| **Type** | Valid |
| **Description** | 序列化遵守DDS规范 |
| **Upstream ID** | SWS\_CM\_11040 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | - |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00072] Strings的编码格式和大小端

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00072 |
| **Type** | Valid |
| **Description** | STRING\_TYPE的标准编码格式为: UTF-8，序列化不包括Byte Order Mark (BOM)。大小端已经在rtps报文体现。 |
| **Upstream ID** | SWS\_CM\_11046 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 按照xtype要求进行序列化。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00073] VARIANT类型的序列化

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00073 |
| **Type** | Valid |
| **Description** | 应该根据DDS UNION\_TYPE进行序列化。 |
| **Upstream ID** | SWS\_CM\_11050 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 按照xtype要求进行序列化。 |
| **Risk** | 无 |
| **Change Type** | 不变 |

### [SWRD\_CM\_DDS\_00074] 服务发现协议选择

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00074 |
| **Type** | Valid |
| **Description** | 通过设置discoveryType属性值确定服务发现的方式，Skeleton和Proxy,应该设置相同的服务发现方式  DomainParticipantUserDataQos ：通过域参与者的QoS 做服务发现  Topic：通过topic做服务发现 |
| **Upstream ID** | SWS\_CM\_90500 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00075] 通过域参与者Qos做服务发现时topic命名规则

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00075 |
| **Type** | Valid |
| **Description** | 主题命名规则应该按照如下格式：ara.com://services/<InterfaceID>/<Major>.<Minor>/< TopicName > |
| **Upstream ID** | SWS\_CM\_90501 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00076] 为服务接口的没一个trigger赋予一个topic和DataWriter

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00076 |
| **Type** | Valid |
| **Description** | DDS绑定对服务接口内的每一个trigger赋予一个topic和DataWriter，因为这些topic可能已经在其他服务接口使用了，应该先在DomainParticipant内查找符合的topic，如果没查到，才会创建一个新的topic。  一旦服务接口内的topic可用，DomainParticipant使用publiser创建DataWriter，DataWriter应根据DdsEventQosProps进行配置。  Topic对象可以在其他服务实例中使用，所以在销毁DomainParticipant时才会销毁topic。 |
| **Upstream ID** | SWS\_CM\_10550 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00077]通过topic的OfferService函数映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_000077 |
| **Type** | Valid |
| **Description** | OfferService函数映射，应该执行以下的操作：  [SWS\_CM\_90503] 应该为服务接口赋予一个DomainParticipant  [SWS\_CM\_90504] 应该为服务接口的每个event赋予一个topic和DataWriter  [SWS\_CM\_90505] 应该为服务接口的所有method提供一个request topic和reply topic，并创建相应的DataReader和DataWriter  [SWS\_CM\_90506]应该为服务接口的每一个有hasNotifier的field赋予一个topic和DataWriter  [SWS\_CM\_90507] 应该为服务接口的所有field的get和set赋予一个request topic和reply topic，并创建并创建相应的DataReader和DataWriter，用于调用配置了hasGetter或者hasSetter属性的field |
| **Upstream ID** | SWS\_CM\_90502 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00078] 在ara.com://services/discovery主题上发布服务id、服务实例id和ServiceInterface contract version

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_000078 |
| **Type** | Valid |
| Description | 应配置DDS Topic、Publisher和DataWriter对象，支持通过ara.com://services/discovery Topic发布公告消息，其类型为ServiceAnnouncementMessagemodule  dds {  module ara {  module com {  enum ServiceInstanceResourceIdentifierType {  SERVICE\_INSTANCE\_RESOURCE\_PARTITION,  SERVICE\_INSTANCE\_RESOURCE\_TOPIC\_PREFIX,  SERVICE\_INSTANCE\_RESOURCE\_INSTANCE\_ID  };  struct ServiceVersion {  uint32 major\_version;  uint32 minor\_version;  };  struct ServiceAnnouncementMessage {  @key string<256> interface\_id;  @key uint16 instance\_id;  ServiceVersion version;  ServiceInstanceResourceIdentifierType identifier\_type;  };  }; // module com  }; // module ara  }; // module dds  其中：  interface\_id是serviceInterfaceId  **instance\_id是**serviceInterfaceId  **version是**ServiceInterface 的majorVersion和minorVersion  需要设置如下Qos属性：  RELIABILITY 设置为 RELIABLE  HISTORY 设置为 KEEP\_LAST ，DEPTH 设置为 1  DURABILITY 设置为 TRANSIENT\_LOCAL  DataWriter 设置好后需要如下操作：   1. 实例化一个ServiceAnnouncementMessage 对象并使用register\_instance()返回的interface\_id和instance\_id更新它 2. 使用上一步返回的实例句柄，通过write()发布样本 3. 保存一份这个样本和实例句柄的副本在服务实例销毁的时候使用 |
| **Upstream ID** | SWS\_CM\_90508 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00079] 通过Topic的StopOfferService方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00079 |
| **Type** | Valid |
| **Description** | 使用公告期间保存的样本和实例句柄调用dispose()  删除所有event关联的DataWriter  删除所有trigger关联的DataWriter  删除所有method关联的DataWriter和DataReader  删除所有field（hasNotifier为true）关联的DataWriter  删除所有field（hasGetter and/or hasSetter）关联的DataWriter和DataReader。 |
| **Upstream ID** | SWS\_CM\_90509 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 删除相关的DataWriter和DataReader，相关报文会通过rtps发送出来。 |
| **Risk** | 无 |
| **Change Type** | 变更 |

### [SWRD\_CM\_DDS\_00080] FindService方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00080 |
| **Type** | Valid |
| **Description** | 查找当前存在能够用于查找服务的DomainParticipant，如果不存在，会创建一个新的DomainParticipant。  创建一个与topic和Qos匹配的DataReader，查看接收到的与服务实例相关联的所有样本（满足FinService的查找标准，有一个兼容的 contract version，不在DdsRequiredServiceInstance.blacklistedVersion黑名单中）  返回基于每个服务实例的HandleType：1.符合查找标准，2.有一个兼容的 contract version 3.不在DdsRequiredServiceInstance.blacklistedVersion黑名单中。HandleType应该包含了DomainParticipant和Publisher 和Subscriber的引用。 |
| **Upstream ID** | SWS\_CM\_90510 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 返回描述中符合的HandleType。 |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00081] 通过ara.com://services/discovery topic来发现remote service

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00081 |
| **Type** | Valid |
| **Description** | 通过DataReader 调用 read()来获取ServiceAnnouncementMessage 并检索服务实例的列表  如果requiredServiceInstanceId设置是ALL，应该为interface\_id等于serviceInterfaceId的每个实例返回一个新的句柄  如果requiredServiceInstanceId设置是ALL，应该为interface\_id等于serviceInterfaceId并且instance等于requiredServiceInstanceId的每个实例返回一个新的句柄  在任何情况，应该先评估ServiceAnnouncementMessage 中服务实例的contract version，只有在下面条件都满足的情况下，才会返回一个新的handers：  1. 发现的服务实例的contract version与serviceInterfaceDeployment配置的接口兼容。  2. contract version不在DdsRequiredServiceInstance.blacklistedVersion黑名单中。 |
| **Upstream ID** | SWS\_CM\_90513 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 如果没有合适的Subscriber和publisher，则创建相应的Subscriber和publisher。 |
| **Risk** | 依赖第三方dds，是否支持迭代查找。 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00082] StartFindService方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00082 |
| **Type** | Valid |
| **Description** | 当连续查找服务，应该执行以下操作：  查找已经存在的有能力查找远端服务的DomainParticipant，如果没有这样的DomainParticipant的存在，则创建一个。  通过ara.com://services/discovery 主题持续监控ServiceAnnouncementMessage 样本的到达，每当发现匹配的服务实例时调用FindServiceHandler |
| **Upstream ID** | SWS\_CM\_90514 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00083] StopFindService方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00083 |
| **Type** | Valid |
| **Description** | 当停止一个通过StartFindService建立的连续的查找，应该执行以下的操作：  查找满足条件的DomainParticipant，如果没有查找到DomainParticipant，StopFindService应该直接返回。  停止监听ara.com://services/discovery topic不再接收ServiceAnnouncementMessage 样本 |
| **Upstream ID** | SWS\_CM\_90515 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码。 |
| **Risk** | - |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00084] trigger和topic的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00084 |
| **Type** | Valid |
| **Description** | topic应满足：  如果identifier\_type属性设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION或SERVICE\_INSTANCE\_RESOURCE\_INSTANCE\_ID，则主题名称应设置为：ara.com://services/<InterfaceID>/<Major>.<Minor>/< TopicName >  如果identifier\_type属性设置为SERVICE\_INSTANCE\_RESOURCE\_PARTITION则该主题的样本应通过 DataWriters和DataReader发送和接收，各自的父 Publisher 和 Subscriber QoS策略需包含：ara.com://services/<InterfaceID>/<InstanceId>  如果identifier\_type属性设置为SERVICE\_INSTANCE\_TOPIC\_PREFIX主题名称应该设置为：ara.com://services/<InterfaceID>/<InstanceID>/<TopicName>  定义topic data type，应该以等效的data type名称注册。 |
| **Upstream ID** | SWS\_CM\_10524 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 会创建该topic实体。 |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00085] trigger topic data type定义

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00085 |
| **Type** | Valid |
| **Description** | topic data types定义符合如下IDL定义：  struct TriggerType {  @key uint16 instanceIdentifier;  };eventTypeName 是event关联的数据类型  instanceIdentifier是这个类型的key，用于从所有的采样中识别相同instanceIdentifier的采样。。 |
| **Upstream ID** | SWS\_CM\_10525 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00086] trigger Send 方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00086 |
| **Type** | Valid |
| **Description** | DDS绑定应该按照如下方法构建一个新的采样：  Instance Id应该为DdsProvidedServiceInstance配置的serviceInstanceId  sample应该被传递给 DataWriter的write()。 |
| **Upstream ID** | SWS\_CM\_10526 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 应根据序列化规则对样本进行序列化，并通过 DDS 发布 |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00087] 为trigger订阅创建DataReader

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00087 |
| **Type** | Valid |
| **Description** | 为了保证proxy只能与绑定的Service Instance通信，应该使用Subscriber（partition name为"ara.com://services/<svcId>\_ <reqSvcInId>"，创建DataReader  DataReader应该如下配置：  DataReaderQos qosProfile应该为DdsEventQosProps指定的Qos，  Listener应该是DataReaderListener类的实例  StatusMask应设置为STATUS\_MASK\_NONE |
| **Upstream ID** | SWS\_CM\_10528 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码，相应的创建DataReader报文将会发送。 |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00088] trigger DataReaderListener的定义

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00088 |
| **Type** | Valid |
| **Description** | DDS绑定应该定义DataReaderListener，处理当订阅状态改变或者有数据到达时的事件处理。该类继承DataReaderListener，Topic data type 应该满足[SWS\_CM\_10525]规范。  DataReaderListener应该实现以下的方法：  构造函数，参数为TriggerReceiveHandler和SubscriptionStateChangeHandler的引用。  on\_data\_available()：如果TriggerReceiveHandler被设置，当数据达到时，则调用TriggerReceiveHandler  on\_subscription\_matched()：如果SubscriptionStateChangeHandler被设置，调用GetSubscriptionState()，并将SubscriptionState 传递给SubscriptionStateChangeHandler。  set\_ trigger \_receive\_handler()：入参为TriggerReceiveHandler的引用，用于更新TriggerReceiveHandler成员  set\_subscription\_state\_change\_handler()：入参为SubscriptionStateChangeHandler的引用，更新SubscriptionStateChangeHandler的成员。 |
| **Upstream ID** | SWS\_CM\_10529 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码。 |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00089] trigger GetNewSamples方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00089 |
| **Type** | Valid |
| **Description** | DDS绑定应该在DataReader上执行take()并且不指定max\_samples的限制。  在调用take()之后，应该按照take()返回的样本数量的比例增加内部触发器计数 |
| **Upstream ID** | SWS\_CM\_10532 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码，并按照API需求进行验证。 |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00090] trigger SetReceiveHandler方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00090 |
| **Type** | Valid |
| **Description** | DDS绑定实行一下操作：  通过DataReader的 get\_listener()方法获取listener  通过调用set\_trigger\_receive\_handler()，通知listener，当数据到达时调用EventReceiveHandler函数  通过set\_listener()方法更新listener，StatusMask 设置如下:  如果原始StatusMask是STATUS\_MASK\_NONE或者  DATA\_AVAILABLE\_STATUS, 则将它设置为DATA\_AVAILABLE\_STATUS.  如果原始StatusMask是 SUBSCRIPTION\_MATCHED\_STATUS, 则设置为  DATA\_AVAILABLE\_STATUS|SUBSCRIPTION\_MATCHED\_STATUS.  如果原始StatusMask是DATA\_AVAILABLE\_STATUS|SUBSCRIPTION\_MATCHED\_STATUS, 则设置为 DATA\_AVAILABLE\_STATUS|SUBSCRIPTION\_MATCHED\_STATUS. |
| **Upstream ID** | SWS\_CM\_10534 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码，并验证该函数功能。 |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00091] trigger UnsetReceiveHandler方法的映射

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00091 |
| **Type** | Valid |
| **Description** | DDS应该执行一下操作：  通过 get\_listener()获取DataReader的 listener引用  使用set\_trigger\_receive\_handler()撤销TriggerReceiveHandler  调用 set\_listener()更新listener，StatusMask按照如下配置：  如果原始StatusMask 是 STATUS\_MASK\_NONE 或者 DATA\_AVAILABLE\_STATUS, 则设置为 STATUS\_MASK\_NONE.  如果原始StatusMask 是 UBSCRIPTION\_MATCHED\_STATUS, 则设置为 SUBSCRIPTION\_MATCHED\_STATUS.  如果原始StatusMask 是DATA\_AVAILABLE\_STATUS|SUBSCRIPTION\_MATCHED\_STATUS, 则设置为 SUBSCRIPTION\_MATCHED\_STATUS. |
| **Upstream ID** | SWS\_CM\_10535 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | 查看代码。 |
| **Risk** | 无 |
| **Change Type** | 新增 |

### [SWRD\_CM\_DDS\_00092] 1911已实现的特性

|  |  |
| --- | --- |
| **SWRD\_ID** | SWRD\_CM\_DDS\_00092 |
| **Type** | Valid |
| **Description** | 1911已实现的SWS ID |
| **Upstream ID** | SWS\_CM\_11006、 SWS\_CM\_11017、SWS\_CM\_11021、 SWS\_CM\_11041、 SWS\_CM\_11042、SWS\_CM\_11043、 SWS\_CM\_11044、 SWS\_CM\_11047、SWS\_CM\_11048、 SWS\_CM\_11049、SWS\_CM\_90503、SWS\_CM\_90504、  SWS\_CM\_90505、SWS\_CM\_90506、SWS\_CM\_90507、  SWS\_CM\_90511、SWS\_CM\_90512、SWS\_CM\_90527、  SWS\_CM\_90530、SWS\_CM\_90531、SWS\_CM\_90536、  SWS\_CM\_90537 |
| **Dependencies** | - |
| **ASIL** | QM |
| **Status** | [In review] |
| **Priority** | M |
| **Verification Criteria** | - |
| **Risk** | 无 |
| **Change Type** | 新增 |

# 非功能需求(Non-Functional Requirements)

无

# 接口说明（API）

参考《软件需求规格说明书-CM API》

附录A- 信息定义

|  |  |  |
| --- | --- | --- |
| 类别 | 结构 | 备注 |
| 需求 | SWRD\_{需求类型}\_{功能简称}\_流水号  功能简称：参见下面功能简称列表  需求类型：功能需求为空，非功能需求为NF,接口为API  流水号：从00001开始的5位自然数 | *例：*  *SWRD\_Nvm\_00001*  *SWRD\_NF\_Nvm\_00001*  *SWRD\_API\_Nvm\_00001* |

|  |  |
| --- | --- |
| 功能简称列表（aCore） | 说明 |
| DM\_DEM | Diagnostics management模块的诊断事件管理 |
| DM\_DCM | Diagnostics management模块的诊断通信管理 |
| DM\_DCM\_DOIP | Diagnostics management模块的DO/IP相关功能 |
| CoreTypes | 核心数据类型 |
| CM\_SOMEIP | Communication management模块的SOME/IP相关功能 |
| CM\_DDS | Communication management模块的DDS相关功能 |
| CM\_CommunicationGroup | Communication management模块的通信组相关功能 |
| CM\_SHM | Communication management模块的共享内存相关功能 |
| CM\_IPC | Communication management模块的IPC相关功能 |
| CM\_Raw | Communication management模块的raw data streaming相关功能 |
| CM\_TLS | Communication management模块的TLS相关功能 |
| CM\_S2S | Communication management模块的S2S相关功能 |
| CM\_E2E | Communication management模块的E2E相关功能 |
| UCM\_Master | Update and config management模块的主站相关功能 |
| UCM\_Server | Update and config management模块的从站相关功能 |
| LT | Log and trace模块相关功能 |
| PHM | Platform health management模块相关功能 |
| Per | Persistency模块相关功能 |
| SM | State management模块相关功能 |
| Crypto | Cryptography模块相关功能 |
| EM | Execution mangement模块相关功能 |
| NM | Network management模块相关功能 |
| TS | Time synchronization模块相关功能 |

说明：根据项目情况可自己定义，增加功能简称

|  |  |  |
| --- | --- | --- |
| 安全等级(ASIL) | 解释说明 | 备注 |
| ASIL A | 根据S – Severity(严重度)  E – Exposure（暴露度） C – Controllability（可控性） 排定功能安全等级。详细理解可以参考26262标准文件。 | *如果有关于ASIL等级的特殊解释说明，请记录在此* |
| ASIL B |  |
| ASIL C |  |
| ASIL D |  |
| QM(A) | 从ASIL A到ASIL D 中拆分出来，拆分的标准，参考功能安全体系文件《功能安全需求分解指南\_FS.pdf》 |  |
| QM(B) |  |
| QM(C) |  |
| QM(D) |  |
| ASIL A(A) |  |
| ASIL A(B) |  |
| ASIL A(C) |  |
| ASIL A(D) |  |
| ASIL B(B) |  |
| ASIL B(C) |  |
| ASIL B(D) |  |
| ASIL C(C) |  |
| ASIL C(D） |  |
| ASIL D(D) |  |

|  |  |  |
| --- | --- | --- |
| 优先级（Priority） | 解释说明 | 备注 |
| H | 高优先级 | *例：被依赖的需求优先级设置为H级别* |
| M | 中优先级 | *例：* |
| L | 低优先级 | *例：其余功能均设置为L级别* |

|  |  |  |
| --- | --- | --- |
| 状态  （Status） | 状态说明 | 备注 |
| Draft | 草稿 | *例：表示新建* |
| In Review | 评审中 | *例：表示处于评审中* |
| Approved | 批准 | *例：表示通过评审* |
| Released | 发布 | *例：表示通过客户确认* |
| Modified | 修改 | *例：表示正在检讨修改中* |

|  |  |  |
| --- | --- | --- |
| 类型  （Type） | 状态说明 | 备注 |
| Valid | 有效 | *例：表示需要对应* |
| InValid | 不适用 | *例：表示不做对应* |
| TBD | 检讨中 | *例：表示正在检讨中* |

说明：根据项目情况可自己定义，但需要明确

|  |  |  |
| --- | --- | --- |
| 变更类型  (Change Type) | 解释说明 | 备注 |
| 新增 | 相对已建立的第一版需求基线（含Base项目的需求基线），如果是新增的需求，选择此项 | 如果有关于每个变更类型的特殊解释说明，请记录在此 |
| 修改 | 相对已建立的第一版需求基线（含Base项目的需求基线），发生了修改的需求 |  |
| 不变 | 相对已建立的base项目的需求基线，复用了base项目的需求，填此类型，如没有Base项目需求基线，不应填此类型。 |  |
| 删除 | 相对已建立的第一版需求基线（含Base项目的需求基线），如果是删除的需求，选择此项。 |  |

说明：根据项目情况可自己定义，但需要明确

附录B- 配置信息

|  |  |  |  |
| --- | --- | --- | --- |
| 配置信息 | 说明 | 范围 | 备注 |
| *API configuration class* |  | *1、2、3* |  |
|  |  |  |  |
|  |  |  |  |