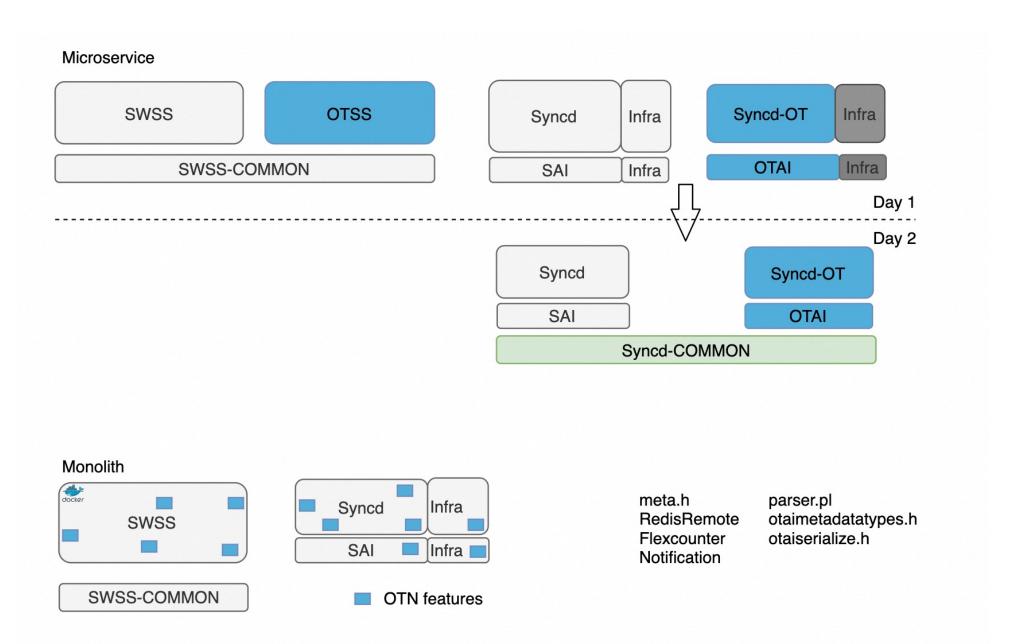
Syncd-ot Source code and architecture overview

Weitang Zheng@Alibaba

Syncd-ot software architectures, Microservice vs Monolithic



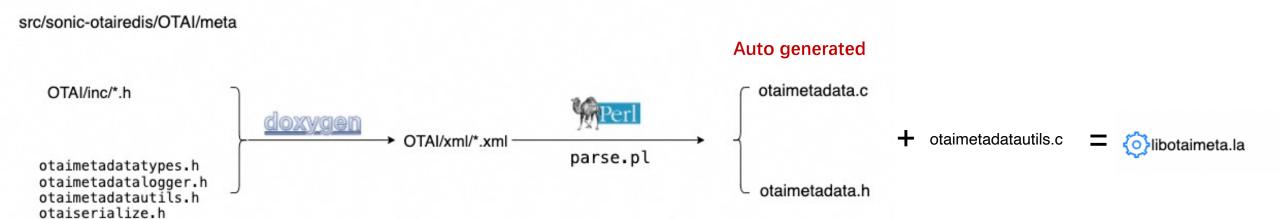
Compare src/otairedis and src/sairedis repository **LICENSE** LICENSE Makefile.am Makefile.am sai/otai interfaces definitions README.md OTAI SAI README.md autoclean.sh autogen.sh autogen.sh debian azsyslog.conf lib azure-pipelines.yml sai/otai redis implementation libtool configure.ac m4 debian meta < Igtm.yml syncd lib sai/otai meta interface implementation vslib m4 meta package.cfg proxylib pyext saiasiccmp Syncd for sai and otai saidiscovery saidump saiplayer saisdkdump stub.pl syncd tests vs implementation for sai and otai unittest vslib



src/otairedis/OTAI/meta

Generate and save OTAI metadata information

- Structure metadata
- Enums metadata
- Union metadata
- Notification metadata
- Apis metadata
- Serializer/ deserializer



libotaimeta.la

```
/**
* @brief Port attribute IDs
*/
typedef enum otai port attr t
/**
* @brief Admin state
* @type otai admin state t
* @flags CREATE AND SET
*/
OTAI PORT ATTR ADMIN STATE,
}
/**
* @brief Output power
* @type otai double t
* @unit dBm
* @precision precision2
* @iscounter false
*/
OTAI PORT STAT_OUTPUT_POWER,
```

attribute

statistics

```
const otai attr metadata t* const* const otai metadata attr by object type[] = {
otai_metadata_object_type_otai_linecard_attr_t,
otai_metadata_object_type_otai_port_attr_t,
};
const otai attr metadata t* const otai metadata object type otai port attr t[] = {
&otai_metadata_attr_OTAI_PORT_ATTR_ADMIN_STATE,
};
const otai_attr_metadata_t otai_metadata_attr_OTAI_PORT_ATTR_ADMIN_STATE = {
.objecttype = OTAI_OBJECT_TYPE_PORT,
.attrid = OTAI PORT ATTR ADMIN STATE,
.attridname = "OTAI_PORT_ATTR_ADMIN_STATE",
.flags = (otai_attr_flags_t)(OTAI_ATTR_FLAGS_CREATE_AND_SET),
.enummetadata = &otai metadata enum otai admin state t,
.ismandatoryoncreate = false,
.iscreateonly = false,
.iscreateandset = true,
.isreadonly = false,
};
const otai_enum_metadata_t otai_metadata_enum_otai_admin_state_t = {
.name = "otai admin state t",
.valuescount = 3,
.values = (const int*)otai_metadata_otai_admin_state_t_enum_values,
.valuesnames = otai metadata otai admin state t enum values names,
};
const otai_stat_metadata_t* const* const otai_metadata_stat_by_object_type[] = {
otai_metadata_stat_object_type_otai_port_stat_t,
```

src/otairedis/meta

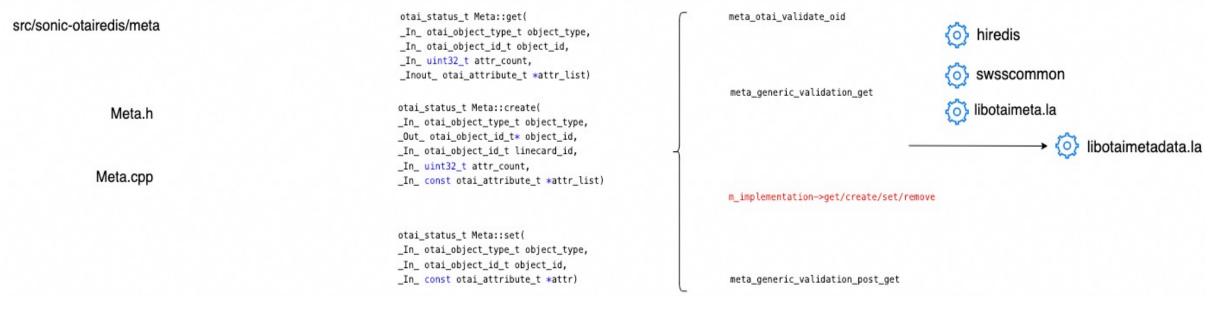
OTAI API interface with metadata validation.

Validate all the parameters and returned values(OID, attributes, enums

Validate notification data

Validate OTAI objects hierarchy (delete all children objects first)

Implement all OTAI APIs in RedisRemoteOtaiInterface

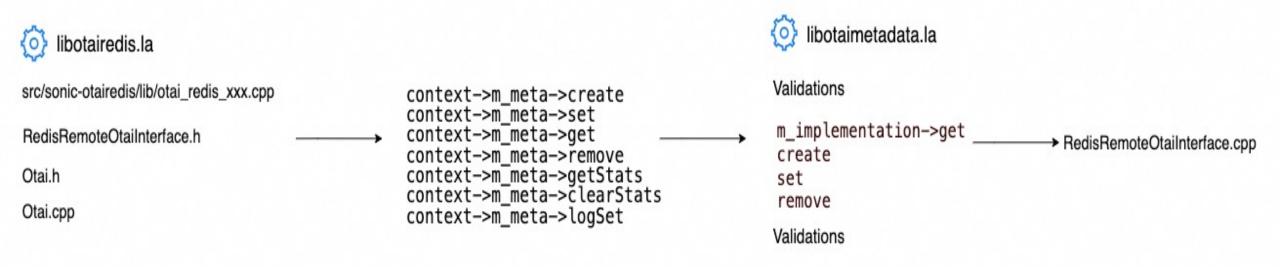




src/otairedis/lib

OTAI API implementation between OTSS and Syncd-OT.

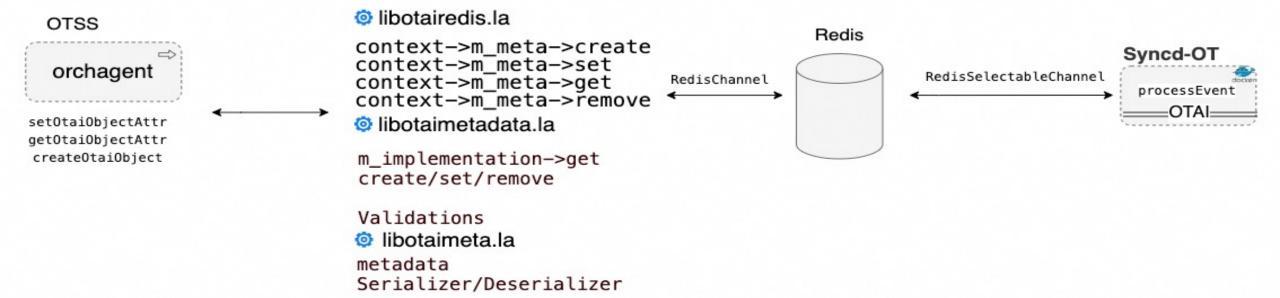
libOtairedis.la implements Otai APIs, Syncd-OT can receive these OTAI CRUD APIs



OTSS and Syncd Interactions

OTSS call libotairedis.la to send CRUD requests to Syncd

It involves libotaimetadata.la and libotaimeta.la for metadata validation and serializer/deserializer



Compare src/otairedis/OTAl and src/sairedis/SAl repository sonic-otairedis/OTAL sonic-sairedis/SAI inc LICENSE.txt sai.h Doxyfile Makefile LICENSE.txt saiacl.h Makefile README.md saiars.h README.md azure-pipelines.yml saiarsprofile.h meta saibfd.h inc data Doxyfile - TAM otai.h saibridge.h Doxyfile.compat debian saibuffer.h otaiaps.h Makefile doc otaiapsport.h saicounter.h acronyms.txt otaiassignment.h **ACL** saidebugcounter.h aspell.en.pws **ARS** otaiattenuator.h saidtel.h cap.pm **BFD** otaiethernet.h saifdb.h checkheaders.pl **DTEL** otaiinterface.h saigenericprogrammable.h checksymbols.pl otailinecard.h tunnel saihash.h aimetadatalogger.l xSTP saiswitch.h otailldp.h otaimetadatatest.c otailogicalchanne.h experimental saisystemport.h otaimetadatatypes.h otaimediachanne .h saiexperimentalbmtor.h otaimetadatautils.c saitunnel.h otaioa.h saiexperimentaldashacl.h otaimetadatautils.h otaiobject.h saitwamp.h otaiserialize.c saiex perimental dash direction look up.hotaioch.h saitypes.h otaiserialize.h saiexperimentaldasheni.h saiudf.h otaiocm.h parse.pl saiexperimentaldashha.h otaiosc.h saiversion.h sample.cap otaiotdr.h flexsai saivirtualrouter.h serialize.pm otaiotn.h saivlan.h spec.pm otaiphysicalchannel.h ptf saiwred.h style.pm otaiport.h LogConfig.py utils.pm otaistatus.h confia saisanity.py xml otaitransceiver.h data module saischeduler.py xmlutils.pm otaitypes.h docs saischedulergroup.py otaiwss.h saisrv6.pv

Compare src/otairedis/meta and src/sairedis/meta repository

sonic-otairedis/meta	sonic-sairedis/meta	├── NotificationSwitchAs
——— AttrKeyMap.cpp	AttrKeyMap.cpp	├── NotificationSwitchAs
——— AttrKeyMap.h	AttrKeyMap.h	NotificationSwitchSh
├── Globals.cpp	DummySaiInterface.cpp	├── NotificationSwitchSh
Globals.h	DummySaiInterface.h	├── NotificationSwitchSt
├── Makefile.am	Globals.cpp	├── NotificationSwitchSt
├── Meta.cpp	Globals.h	├── NotificationTwampS
├── Meta.h	├── Makefile.am	NotificationTwampS
—— MetaKeyHasher.cpp	├── Meta.cpp	NumberOidIndexGe
MetaKeyHasher.h	├── Meta.h	NumberOidIndexGe
Notification.cpp	├── MetaKeyHasher.cpp	OidRefCounter.cpp
Notification.h	├── MetaKeyHasher.h	OidRefCounter.h
├── NotificationFactory.cpp	├── MetaTestSaiInterface.cpp	PerformanceInterval
NotificationFactory.h	MetaTestSaiInterface.h	PerformanceInterval
├── NotificationLinecardStateChange.cpp	Notification.cpp	PortRelatedSet.cpp
├── NotificationLinecardStateChange.h	Notification.h	PortRelatedSet.h
├── NotificationOcmNotify.cpp	├── NotificationBfdSessionStateChange.cpp	RedisSelectableChar
NotificationOcmNotify.h	NotificationBfdSessionStateChange.h	RedisSelectableChar
OidRefCounter.cpp	NotificationFactory.cpp	SaiAttrWrapper.cpp
OidRefCounter.h	NotificationFactory.h	SaiAttrWrapper.h
OtaiAttrWrapper.cpp	NotificationFdbEvent.cpp	SaiAttributeList.cpp
OtaiAttrWrapper.h	NotificationFdbEvent.h	SaiAttributeList.h
OtaiAttributeList.cpp	NotificationNatEvent.cpp	Sailnterface.cpp
OtaiAttributeList.h	NotificationNatEvent.h	Sailnterface.h
—— OtaiInterface.cpp	├── NotificationPortHostTxReadyEvent.cpp	├── SaiObject.cpp
OtaiInterface.h	├── NotificationPortHostTxReadyEvent.h	├── SaiObject.h
—— OtaiObject.cpp	├── NotificationPortStateChange.cpp	SaiObjectCollection.
├── OtaiObject.h	NotificationPortStateChange.h	SaiObjectCollection.
OtaiObjectCollection.cpp	├── NotificationQueuePfcDeadlock.cpp	SaiSerialize.cpp
—— OtaiObjectCollection.h	├── NotificationQueuePfcDeadlock.h	SelectableChannel.c

Compare src/otairedis/lib and src/sairedis/lib repository

RemoteOtaiInterface.h sonic-sairedis sonic-sairedis/lib sonic-otairedis/lib SkipRecordAttrContainer.cpp ServerConfig.cpp Channel.cpp Channel.cpp SkipRecordAttrContainer.h ServerConfig.h Channel.h Channel.h Utils.cpp ClientConfig.cpp ServerSai.cpp Context.cpp Utils.h ServerSai.h ClientConfig.h Context.h SkipRecordAttrContaine VirtualObjectIdManager.cpp ClientSai.cpp ContextConfig.cpp VirtualObjectIdManager.h ClientSai.h SkipRecordAttrContaine ContextConfig.h context config.json ClientServerSai.cpp Switch.cpp ContextConfigContainer.cpp otai redis.h ClientServerSai.h Switch.h ContextConfigContainer.h SwitchConfig.cpp otai redis aps.cpp Context.cpp Linecard.cpp SwitchConfig.h otai redis apsport.cpp Context.h Linecard.h SwitchConfigContainer.c otai redis assignment.cpp ContextConfig.cpp LinecardConfig.cpp SwitchConfigContainer.h otai redis attenuator.cpp ContextConfia.h LinecardConfig.h SwitchContainer.cpp otai redis ethernet.cpp ContextConfigContainer.cpp LinecardConfigContainer.cpp otai redis interface.cpp SwitchContainer.h ContextConfigContainer.h LinecardConfigContainer.h Utils.cpp otai redis interfacequery.cpp Makefile.am LinecardContainer.cpp Utils.h otai redis linecard.cpp OidIndexGenerator.h LinecardContainer.h otai redis Ildp.cpp VirtualObjectIdManager Recorder.cpp Makefile.am otai redis logicalchannel.cpp Recorder.h VirtualObjectIdManager OidIndexGenerator.h ZeroMQChannel.cpp otai redis mediachannel.cpp RedisChannel.cpp Otai.cpp otai_redis_oa.cpp RedisChannel.h ZeroMQChannel.h Otai.h client config.json otai redis och.cpp RedisRemoteSaiInterface.cpp Otailnternal.h otai redis ocm.cpp RedisRemoteSaiInterface.h context config.json Recorder.cpp otai redis osc.cpp RedisVidIndexGenerator.cpp sairedis.h Recorder.h RedisVidIndexGenerator.h sairediscommon.h otai redis otdr.cpp RedisChannel.cpp otai redis otn.cpp RemoteSaiInterface.h tests.cpp RedisChannel.h otai redis physicalchannel.cpp Sai.cpp RedisRemoteOtaiInterface.cpp Sai.h otai_redis_port.cpp RedisRemoteOtaiInterface.h otai redis transceiver.cpp SaiInternal.h RedisVidIndexGenerator.cpp otai redis wss.cpp RedisVidIndexGenerator.h

atairadia h

Create Functions in Meta

```
static otai_status_t redis_create_linecard_uniq(
_Out_ otai_object_id_t *linecard_id,
_In_ uint32_t attr_count,
_In_ const otai_attribute_t *attr_list)
typedef int32 t otai status t;
                                                                 typedef INT32 sai_status_t;
typedef uint64 t otai object id t;
                                                                 typedef uint64 t sai object id t;
typedef struct _otai_attribute_t
otai attr_id_t id;
otai attribute value t value;
} otai attribute t;
typedef uint32 t otai attr id t;
typedef union otai attribute value t
/** @validonly meta->attrvaluetype == OTAI ATTR VALUE TYPE BOOL */
bool booldata;
/** @validonly meta->attrvaluetype == OTAI ATTR VALUE TYPE CHARDATA */
char chardata[512];
/** @validonly meta->attrvaluetype == OTAI ATTR VALUE TYPE UINT8 */
otai uint8 t u8;
```

Create Function in Meta

```
static otai_status_t redis_create_linecard_uniq(
_Out_ otai_object_id_t *linecard_id,
_In_ uint32_t attr_count,
_In_ const otai_attribute_t *attr_list)
       auto status = context->m_meta->create(
       objectType,
       objectId,
       linecardId,
       attr_count,
       attr_list);
```

Metadata class is type sensitive, and deeply coupled with SAI and OTAI

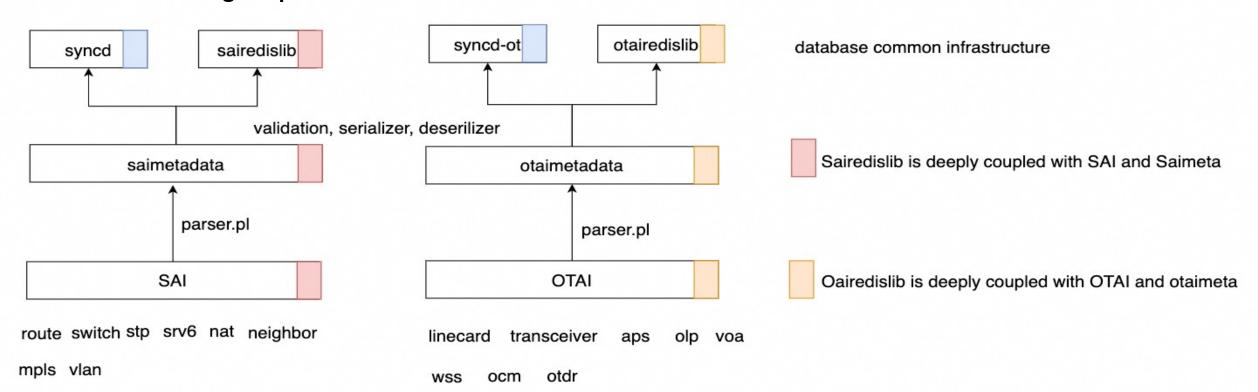
```
sai_metadata_all_object_type_infos
```

Option1: OTAI and Sai in different repository

- Move database common infrastructure to SWSS_COMMON
- Two parallel and independent APIs (SAI and OTAI)
 Meta/
 Lib/

Sai

OTN group maintenance the OTAI APIs and metadata



Option2: Merge SAI and OTAI

Move database common infrastructure to SWSS_COMMON

Embedded OTN APIs in SAI Meta/ Lib/ Sai

Different Syncd-ot module and OTSS container

