Developer`s Overview of Sonic

Praveen Chaudhary

Sr. Software Engineer

LinkedIn

Developer`s Overview of Sonic

To Become a Sonic Developer

1.) https://github.com/Azure/SONiC/wiki
2.) User Guide
3.) Developer`s overview Session(s).
    ** New Sonic Wiki will have link to User Guide & OCP.
    ** https://sonicswitch.slack.com/messages (Join Slack)

Praveen Chaudhary
LinkedIn

pchaudhary@linkedin.com

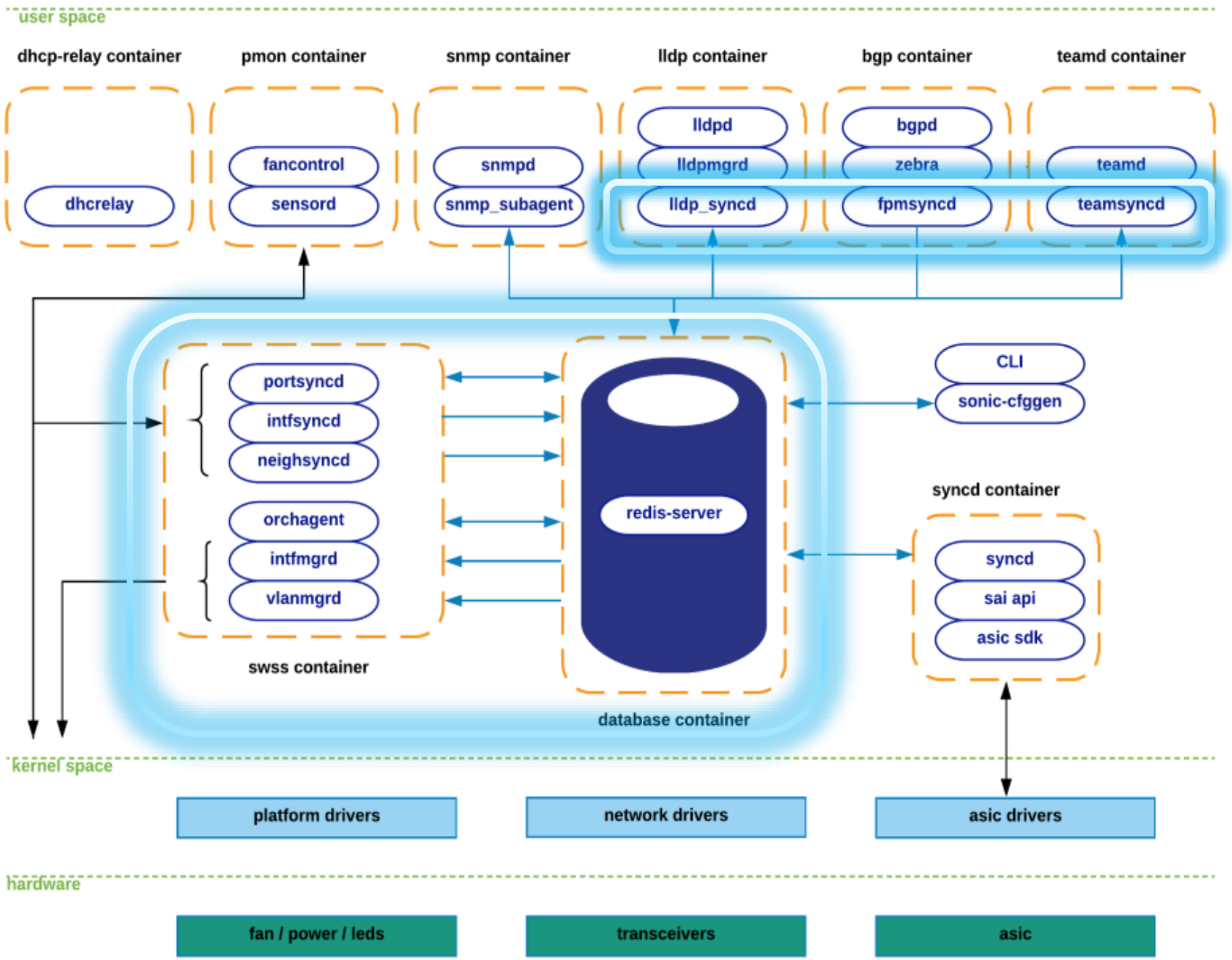
1. Redis-server, different DBs, view of key-value set, basic libraries to interact with Redis.

2. Components interaction.

3. *Syncd Processes. FPMSyncd, LLDP_Syncd, TeamSyncd, PortSyncd and NeighSyncd.

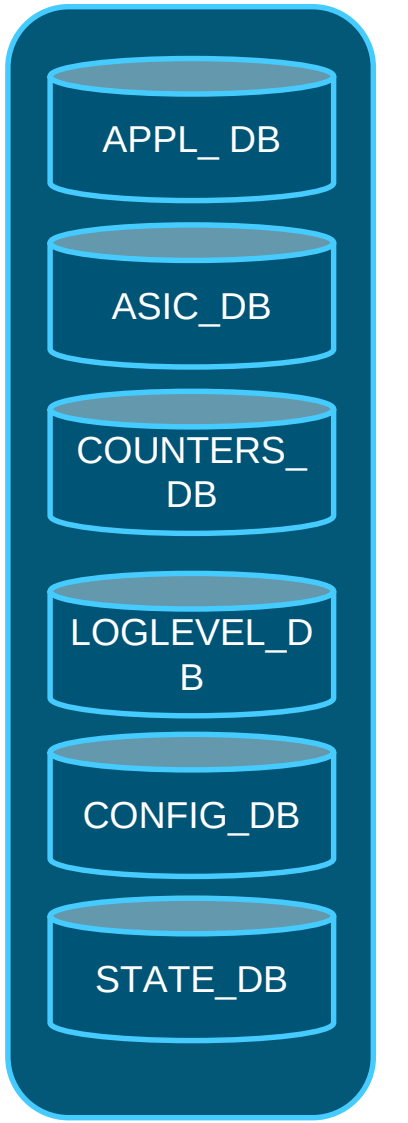
4. Orchagent [Swss Docker]

5. Make file overview to build a new Feature with Sonic Base image or in a new docker.



https://github.com/Azure/sonic-swss-common/blob/master/common/schema.h

DB name	DB No.	Description	Additional Information
APPL_DB (Application DB)	0	ARP/NDP Entries, BGP Routes, LLDP entries, Next-Hop etc.	ROUTE_TABLE: INTF_TABLE: NEIGH_TABLE: VLAN_MEMBER_TABLE: PORT_TABLE: COPP_TABLE: LLDP_ENTRY_TABLE:
ASIC_DB	1	Running ASIC Configuration and ASIC State Data.	"ASIC_STATE:SAI_OBJECT_TYPE_ROUTE_ENTRY:{\"dest\":\"xxxx:f3g5:60:4::12b/128\",\"switch_id\":\"oid:0x2777000000000\",\"vr\":\"oid:0x3000000000042\"}" "ASIC_STATE:SAI_OBJECT_TYPE_NEXT_HOP_GROUP_MEMBER:oid:0x2d000000003363" "ASIC_STATE:SAI_OBJECT_TYPE_NEIGHBOR_ENTRY:{\"ip\":\"xxxx:f349:40:a794::2\",\"rif\":\"oid:0x6000000000add\",\"switch_id\":\"oid:0x21000000000000\"}"
COUNTERS_DB	2	Counter data for port, lag, queue, ACLs.	COUNTERS: CRM:ACL_STATS:INGRESS:LAG:



Redis Server

LOGLEVEL_DB	3	Log level control for Sonic subsystems	swssloglevel -p buffermgrd NOTICE fpmsyncd NOTICE intfmgrd NOTICE intfsyncd NOTICE neighsyncd NOTICE orchagent NOTICE portsyncd NOTICE syncd NOTICE teamsyncd NOTICE vlanmgrd NOTICE
CONFIG_DB	4	DB for Sonic Configuration	/etc/sonic/config_db.json $config reload\load [sonic-cfggen] VLAN_MEMBER PORT|Ethernet90 INTERFACE|Ethernet116|xxxx:f4x7:470:a750::2/126 ACL_RULE|NO-NSW-PACL-V6|Rule_500 BGP NEIGHBORS
STATE_DB	6	Operational state for objects in CONFIG_DB	PORT_TABLE|Ethernet106: VLAN_TABLE|Vlan567: VLAN_TABLE|Vlan234:

Augmented Backus-Naur Form (ABNF) RFC 5234

Redis DB: Key-Value data & Python libraries

"PORT|Ethernet4”

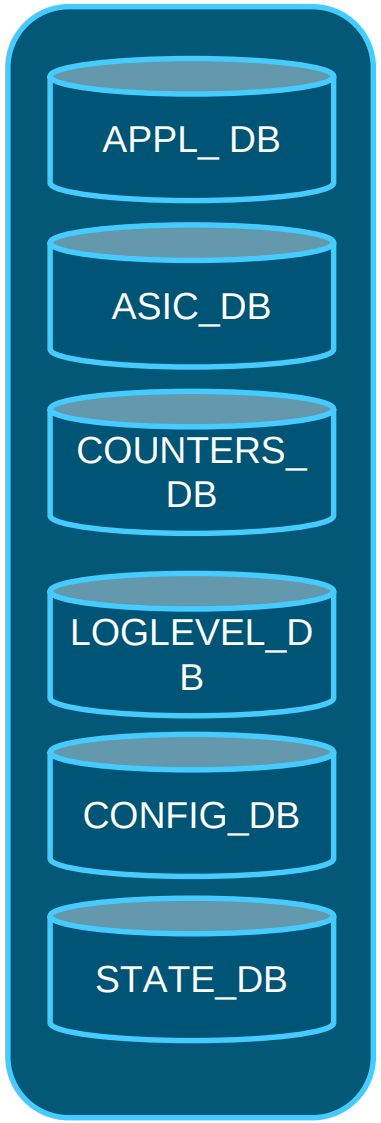
1) "alias"
2) "Eth2/1"
3) "lanes"
4) "69,70"
5) "description"
6) ”xxG|switch-in-dc.nw|Ethernet112"
7) "fec"
8) ”xx"

"ROUTE_TABLE:xxxx:f547:4551:21b::/64”

1) "ifname"
2)"Ethernet64,Ethernet66,Ethernet68,Ethernet70,Ethernet72,Ethernet74,Ethernet76,Ethernet78
3) "nexthop"
4)xxxx:f547:40:4027::1,xxxx:f547:40:4067::1,xxxx:f547:40:40a7::1,xxxx:f547:40:40e7::1,xxxx:f547:40:4127::1,xxxx:f547:40:4167::1,xxxx:f547:40:41a7::1,xxxx:f547:40:41e7::1"

"ASIC_STATE:SAI_OBJECT_TYPE_ROUTE_ENTRY:{\"dest\":\”xx.aaa.239.128/26\",\"switch_id\":\"oid:0x21000000000000\",\"vr\":\"oid:0x3000000000042\"}”

1) "SAI_ROUTE_ENTRY_ATTR_NEXT_HOP_ID"
2) "oid:0x500000000xxc2"



Redis Server

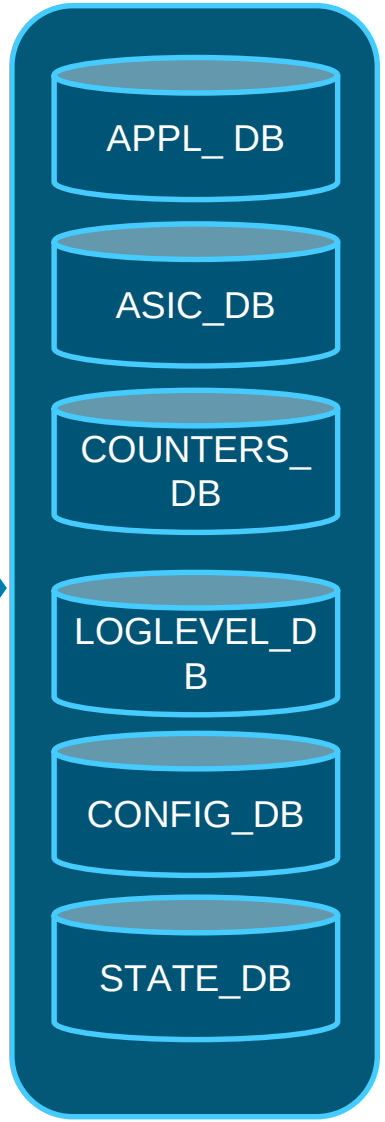
	Code Path for Python Libraries:
	https://github.com/Azure/sonic-py-swsssdk/blob/master/src/swsssdk/interface.py https://github.com/Azure/sonic-py-swsssdk/blob/master/src/swsssdk/configdb.py https://github.com/Azure/sonic-py-swsssdk/blob/master/src/swsssdk/dbconnector.py

Redis DB: interact using Python libraries

Main Classes:

```
-----------------------------------------------------------------------------
class DBInterface(object):
    REDIS_HOST = '127.0.0.1'
    REDIS_PORT = 6379
    REDIS_UNIX_SOCKET_PATH = "/var/run/redis/redis.sock"
    CONNECT_RETRY_WAIT_TIME = 10
    DATA_RETRIEVAL_WAIT_TIME = 3
    PUB_SUB_NOTIFICATION_TIMEOUT = 10.0  # seconds
    PUB_SUB_MAXIMUM_DATA_WAIT = 60.0  # seconds
    KEYSPACE_PATTERN = '__key*__:*'
    KEYSPACE_EVENTS = 'KEA'

    db_map = dict()
    def __init__(self, **kwargs):
    def db_list(self):
    def get_dbid(cls, db_name):
    def connect(self, db_name, retry_on=True):
    def _onetime_connect(self, db_name):
    def _persistent_connect(self, db_name):
    def _subscribe_keyspace_notification(self, db_name):
    def get_redis_client(self, db_name):
    def publish(self, db_name, channel, message):
   def exists(self, db_name, key):
    def keys(self, db_name, pattern='*'):
    def get(self, db_name, _hash, key):
    def get_all(self, db_name, _hash):
    def set(self, db_name, _hash, key, val):
-----------------------------------------------------------------------------

class SonicV2Connector(DBInterface):
class ConfigDBConnector(SonicV2Connector):
```

Py DB Intterface



Redis Server

Py DB Intterface

Example Code:

Example1: Fetch all keys from VLAN table from Config DB.

```
kwargs = {}
if redis_unix_socket_path:
    kwargs['unix_socket_path'] = redis_unix_socket_path

config_db = ConfigDBConnector(**kwargs)
config_db.connect(wait_for_init=False)

data = config_db.get_table('VLAN')
keys = data.keys()
```

Example2: Fetch set of key-value pair for a bvid from ASIC DB:

```
db = SonicV2Connector (**redis_kwargs)
db.connect('ASIC_DB')

vlan_obj = db.keys('ASIC_DB',
"ASIC_STATE:SAI_OBJECT_TYPE_VLAN:" + bvid)

vlan_entry = db.get_all('ASIC_DB', vlan_obj[0],
blocking=True)

vlan_id = vlan_entry[b"SAI_VLAN_ATTR_VLAN_ID"]

>>>>Sample Output:
 vlan_obj =
["ASIC_STATE:SAI_OBJECT_TYPE_VLAN:oid:0x26000
0000012a8"]

vlan_entry {
        "SAI_VLAN_ATTR_VLAN_ID": "555"
 } <<<<<<
```

Redis DB: interact with c++ libraries

C++ Libraries:

https://github.com/Azure/sonic-swss-common/blob/master/common/dbconnector.cpp
https://github.com/Azure/sonic-swss-common/blob/master/common/table.cpp

---

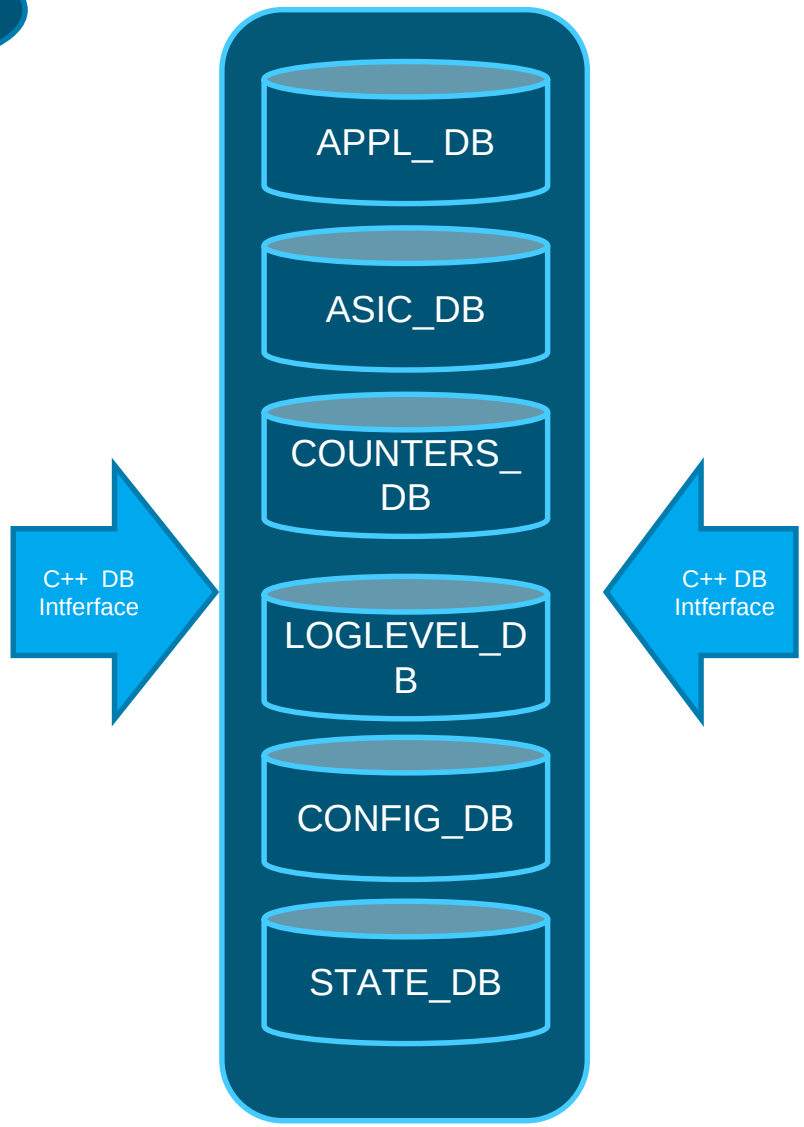
```
class DBConnector
{
public:
    static constexpr const char
*DEFAULT_UNIXSOCKET =
"/var/run/redis/redis.sock";

..........
}
```

---

https://github.com/Azure/sonic-swss-common/common/ *.h

(Global vars can be found here.)

```
#define CONFIG_DB       4
#define CFG_PORT_TABLE_NAME          "PORT"
#define CONFIGDB_TABLE_NAME_SEPARATOR
"|"
```



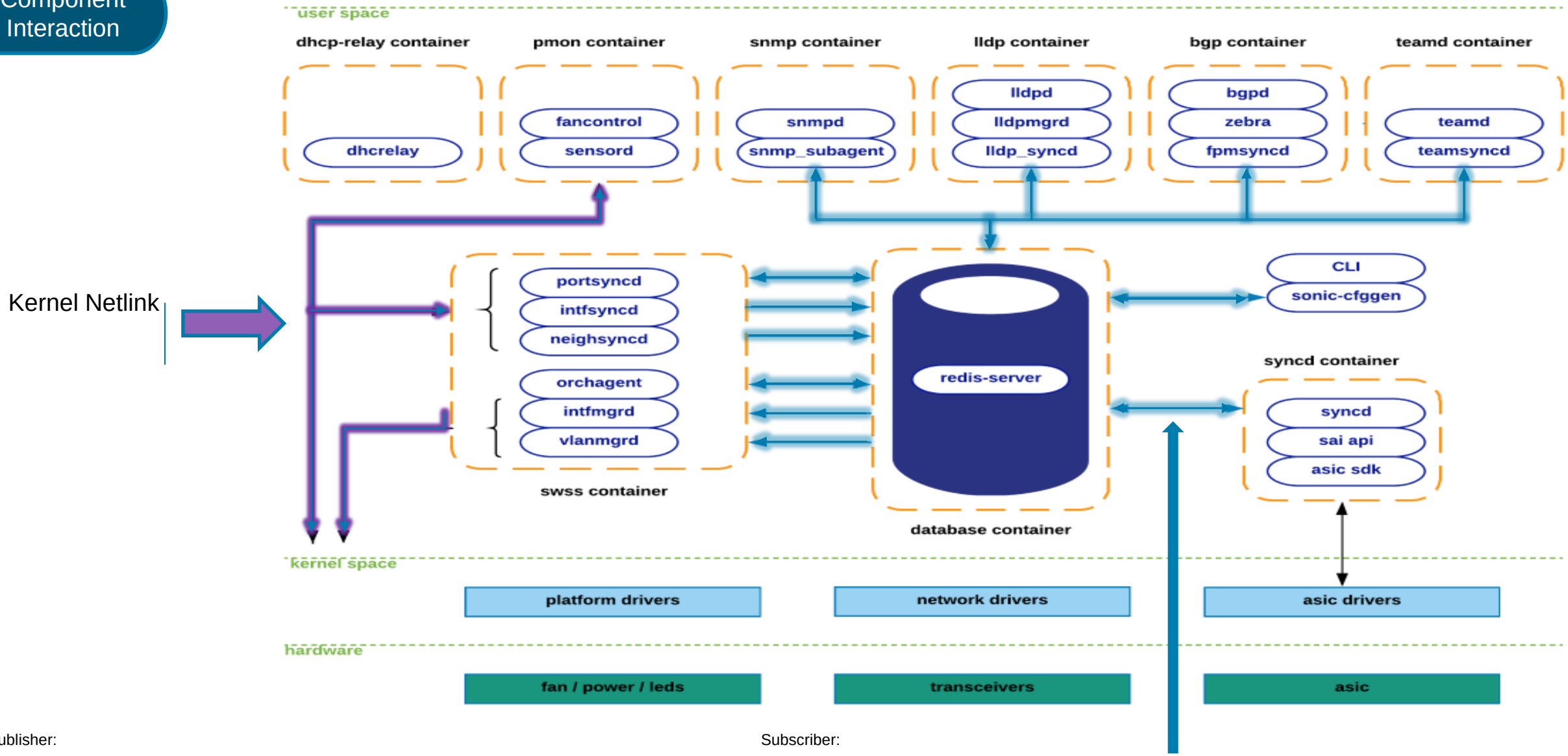
Redis Server

Example Code:

Access all <keys> and <key-value> pair from PORT_TABLE of CONFIG_DB using C++ libs:

```
----------
DBConnector cfgDb(CONFIG_DB,
DBConnector::DEFAULT_UNIXSOCKET, 0);

Table table(&cfgDb,
CFG_PORT_TABLE_NAME,
CONFIGDB_TABLE_NAME_SEPARATOR);

std::vector<FieldValueTuple> values;

std::vector<string> keys;table.getKeys(keys);

for ( auto &k : keys )
{
    table.get(k, ovalues);
    /----My Code ----/
}
```

Component Interaction



Publisher:

---

https://github.com/Azure/sonic-swss-common/blob/master/common/producerstatetable.cpp
https://github.com/Azure/sonic-swss-common/blob/master/common/producertable.cpp

Subscriber:

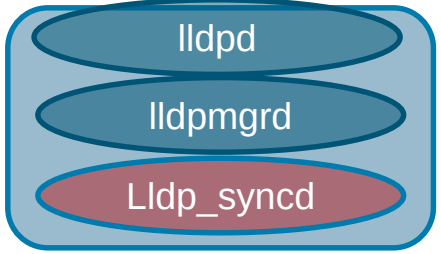
---

https://github.com/Azure/sonic-swss-common/blob/master/common/subscriberstatetable.cpp
https://github.com/Azure/sonic-swss-common/blob/master/common/consumertablebase.cpp
https://github.com/Azure/sonic-swss-common/blob/master/common/consumerstatetable.cpp

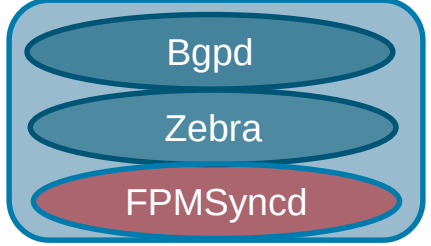
static constexpr **const char** *DEFAULT_UNIXSOCKET = **"/var/run/redis/redis.sock"**;

---

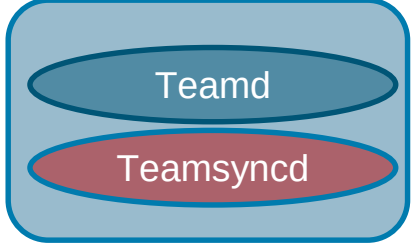
*_Syncd Processes:



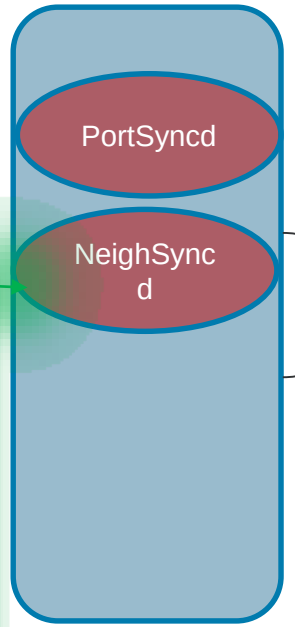
LLDP



FRR



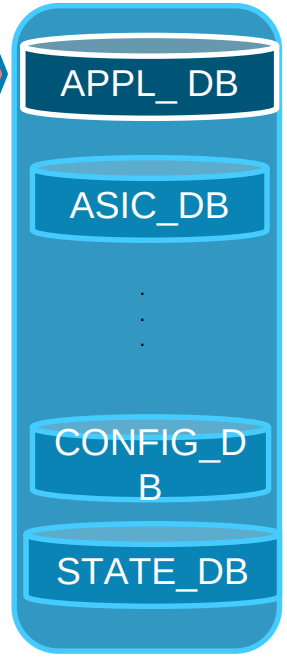
TeamD



SWSS Docker

Code Path:
https://github.com/Azure/sonic-swss/blob/master/neighsyncd
https://github.com/Azure/sonic-swss/blob/201803/ neighsyncd
(before warm reboot feature)

```
-----------------------------------------------------------------
Class NeighSync: public NetMsg
{
.......
producerStateTable m_neighTable;
}
-----------------------------------------------------------------
/* Main Function */
Main():
DBConnector db(APPL_DB, DBConnector::DEFAULT_UNIXSOCKET,
0);
NeighSync sync(&db);
-----------------------------------------------------------------
NeighSync::NeighSync(DBConnector *db) :
m_neighTable(db, APP_NEIGH_TABLE_NAME)
{}
-----------------------------------------------------------------
void NeighSync::onMsg(int nlmsg_type, struct nl_object *obj)
....
nl_addr2str(rtnl_neigh_get_dst(neigh), ipStr, MAX_ADDR_SIZE);
....
nl_addr2str(rtnl_neigh_get_lladdr(neigh), macStr, MAX_ADDR_SIZE);
.....
FieldValueTuple f("family", family);
FieldValueTuple nh("neigh", macStr);
fvVector.push_back(nh);
fvVector.push_back(f);
m_neighTable.set(key, fvVector);
}
-----------------------------------------------------------------
```



Code Path:
https://github.com/Azure/sonic-swss/blob/201803/fpmsyncd/fpmsyncd.cpp

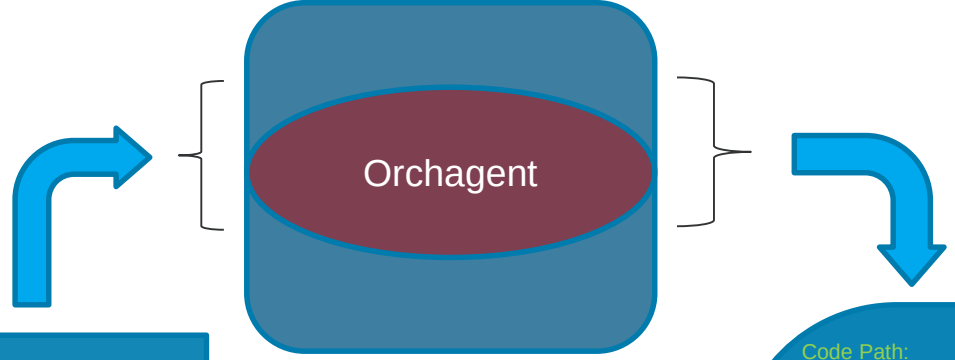
```
-----------------------------------------------------------------
DBConnector db(APPL_DB, DBConnector::DEFAULT_UNIXSOCKET, 0);
RedisPipeline pipeline(&db);
RouteSync sync(&pipeline);<<<<<<<<<<[Class to Process the netlink from zebra]

NetDispatcher::getInstance().registerMessageHandler(RTM_NEWROUTE, &sync);
NetDispatcher::getInstance().registerMessageHandler(RTM_DELROUTE, &sync);

/* Reading FPM messages forever (and calling "readMe" to read them) */
s.select(&temps, &tempfd);
pipeline.flush();
```

Code path for RouteSync is at:
https://github.com/Azure/sonic-swss/blob/201803/fpmsyncd/routesync.cpp

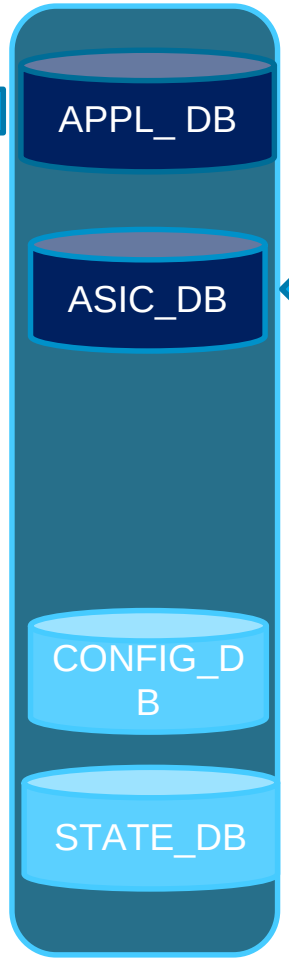
```
void RouteSync::onMsg(int nlmsg_type, struct nl_object *obj) {
.....
nl_addr2str(dip, destipprefix, MAX_ADDR_SIZE);

struct rtnl_nexthop *nexthop = rtnl_route_nexthop_n(route_obj, i);
.....
FieldValueTuple nh("nexthop", nexthops);
FieldValueTuple idx("ifname", ifnames);
fvVector.push_back(nh);
fvVector.push_back(idx);
m_routeTable.set(destipprefix, fvVector);
}
```

Kernel

Orchagent Processes:



SWSS



Code Path:
https://github.com/Azure/sonic-swss/blob/201803/orchagent/orchdaemon.cpp

---

```
gPortsOrch = new PortsOrch(m_applDb, ports_tables);
gFdbOrch = new FdbOrch(m_applDb, APP_FDB_TABLE_NAME, gPortsOrch);
IntfsOrch *intfs_orch = new IntfsOrch(m_applDb, APP_INTF_TABLE_NAME);
gNeighOrch = new NeighOrch(m_applDb, APP_NEIGH_TABLE_NAME, intfs_orch);
gRouteOrch = new RouteOrch(m_applDb, APP_ROUTE_TABLE_NAME, gNeighOrch);
```

***Orch Classes**

```
m_orchList = { switch_orch, gCrmOrch, gBufferOrch, gPortsOrch, intfs_orch,
gNeighOrch, gRouteOrch, copp_orch, tunnel_decap_orch, qos_orch, mirror_orch,
gAclOrch, gFdbOrch, vrf_orch };
```

---

class NeighOrch : public Orch, public Subject

Code Path: https://github.com/Azure/sonic-swss/blob/201803/orchagent/orch.cpp

---

```
class Consumer : public Executor {
…
    void execute();
    void drain();
    SyncMap m_toSync;
};
```

```
void Consumer::execute() {
            m_toSync[key] = KeyOpFieldsValuesTuple(key, op, existing_values);
….
}
```

```
void Consumer::drain()
{
    if (!m_toSync.empty())
        m_orch->doTask(*this);
}
```

Code Path:
https://github.com/Azure/sonic-swss/blob/201803/orchagent/routeorch.cpp

---

```
void RouteOrch::doTask(Consumer& consumer)
{
  auto it = consumer.m_toSync.begin();
  while (it != consumer.m_toSync.end())
  {
     /* ---Process as per the role */
     Route_orch.<func>() [SAI call()]
    /* Erase it, if success.
    it = consumer.m_toSync.erase(it);
    continue;
  }
```

```
SAI: status = sai_route_api->create_route_entry(&unicast_route_entry, 1, &attr);
```

---

class RouteOrch : public Orch, public Subject

```
{
public:
    RouteOrch(DBConnector *db, string tableName, NeighOrch *neighOrch);
    bool hasNextHopGroup(const IpAddresses&) const;
    sai_object_id_t getNextHopGroupId(const IpAddresses&);
    void increaseNextHopRefCount(IpAddresses);
    void decreaseNextHopRefCount(IpAddresses);
    bool isRefCounterZero(const IpAddresses&) const;
    bool addNextHopGroup(IpAddresses);
    bool removeNextHopGroup(IpAddresses);
    bool validnexthopinNextHopGroup(const IpAddress &);
    bool invalidnexthopinNextHopGroup(const IpAddress &);
    void addTempRoute(IpPrefix, IpAddresses);
    bool addRoute(IpPrefix, IpAddresses);
    bool removeRoute(IpPrefix);

   void doTask(Consumer& consumer);
};
```

[Access Control List]
aclorch.cpp
aclorch.h

[Control Plane Policy]
copporch.cpp
copporch.h

[Forwarding DataBase]
fdborch.cpp
fdborch.h

[Interfacse]
intfsorch.cpp
intfsorch.h

[Mirror]
mirrororch.cpp
mirrororch.h

[Neighbor]
neighorch.cpp
neighorch.h

[Mirror]
mirrororch.cpp
mirrororch.h

[Neighbor]
neighorch.cpp
neighorch.h

[Base Orch Class]
orch.cpp
orch.h

[Priority Flow Control]
pfcwdorch.cpp
pfcwdorch.h

[Port]
portsorch.cpp
portsorch.h

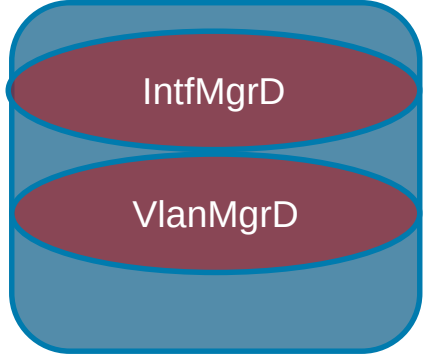
[Route]
routeorch.cpp
routeorch.h

[Tunnel Decap]
tunneldecaporch.cpp
tunneldecaporch.h

[Virtual Routing and Forwarding]
vrforch.cpp
vrforch.h

https://github.com/Azure/sonic-swss/tree/201803/orchagent

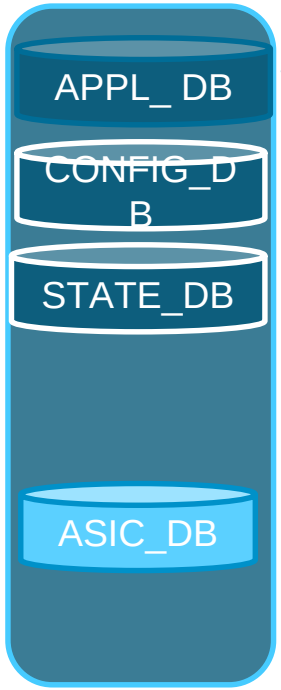
IntfMgrD & VlanMgrD Processes:



Code Path:
https://github.com/Azure/sonic-swss/blob/201803/cfgmgr/intfmgrd.cpp

```
class IntfMgr : public Orch
{
public:
    IntfMgr(DBConnector *cfgDb, DBConnector *appDb, DBConnector *stateDb, const vector<string> &tableNames);
    using Orch::doTask
private:
    ProducerStateTable m_appIntfTableProducer;
    Table m_cfgIntfTable, m_cfgVlanIntfTable;
    Table m_statePortTable, m_stateLagTable, m_stateVlanTable;
    bool setIntfIp(const string &alias, const string &opCmd, const string &ipPrefixStr);

    void doTask(Consumer &consumer);
    bool isIntfStateOk(const string &alias);
};
```

SWSS



Code Path:
https://github.com/Azure/sonic-swss/blob/201803/cfgmgr/vlanmgrd.cpp

```
class VlanMgr : public Orch
public:
 VlanMgr(DBConnector *cfgDb, DBConnector *appDb, DBConnector *stateDb, const vector<string> &tableNames);
    using Orch::doTask;
private:
    ProducerStateTable m_appVlanTableProducer, m_appVlanMemberTableProducer;
    Table m_cfgVlanTable, m_cfgVlanMemberTable;
    Table m_statePortTable, m_stateLagTable;
    Table m_stateVlanTable;
    std::set<std::string> m_vlans;

    void doTask(Consumer &consumer);

    void doVlanTask(Consumer &consumer);
    void doVlanMemberTask(Consumer &consumer);
    void processUntaggedVlanMembers(string vlan, const string &members);
    bool addHostVlan(int vlan_id);
    bool removeHostVlan(int vlan_id);
    bool setHostVlanAdminState(int vlan_id, const string &admin_status);
    bool setHostVlanMtu(int vlan_id, uint32_t mtu);
    bool addHostVlanMember(int vlan_id, const string &port_alias, const string& tagging_mode);
    bool removeHostVlanMember(int vlan_id, const string &port_alias);
    bool isMemberStateOk(const string &alias);
    bool isVlanStateOk(const string &alias);
    bool isVlanMacOk();
```

Kernel

List of open source code repo used in Sonic:

FRR & Zebra: https://github.com/FRRouting/frr

LLDP: https://github.com/vincentbernat/lldpd.git

LLDPMGRD: https://github.com/Azure/sonic-buildimage/blob/master/dockers/docker-lldp-sv2/lldpmgrd

SNMP: https://sourceforge.net/projects/net-snmp/files/net-snmp/5.7.3/

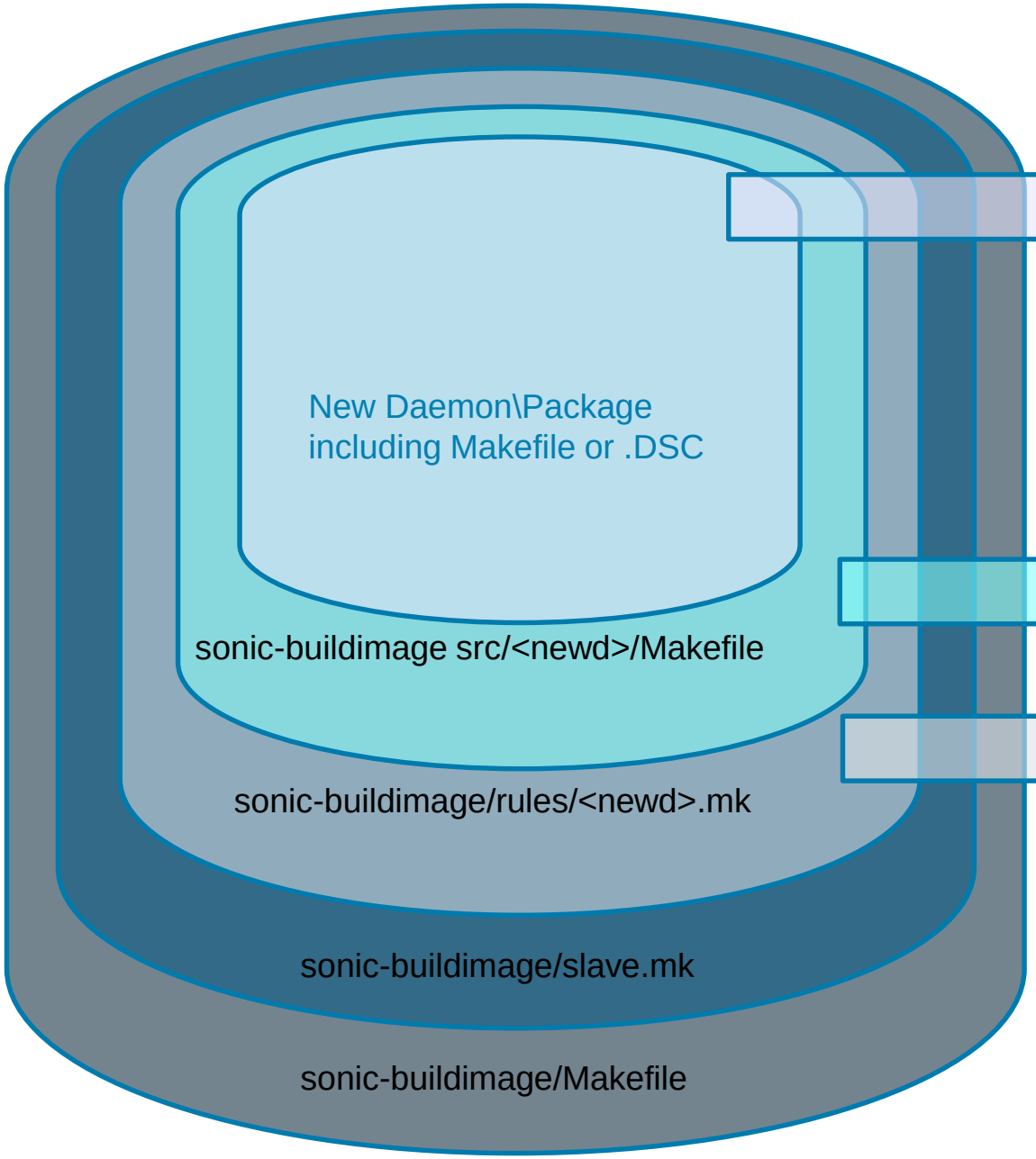
Sonic_snmp agent: https://github.com/Azure/sonic-snmpagent/tree/master/src

Teamd: https://salsa.debian.org/debian/libteam, https://github.com/jpirko/libteam.git

Dhcp_Relay: https://salsa.debian.org/berni/isc-dhcp.git

CLI: https://github.com/Azure/sonic-utilities

Sonic Makefile Overview:

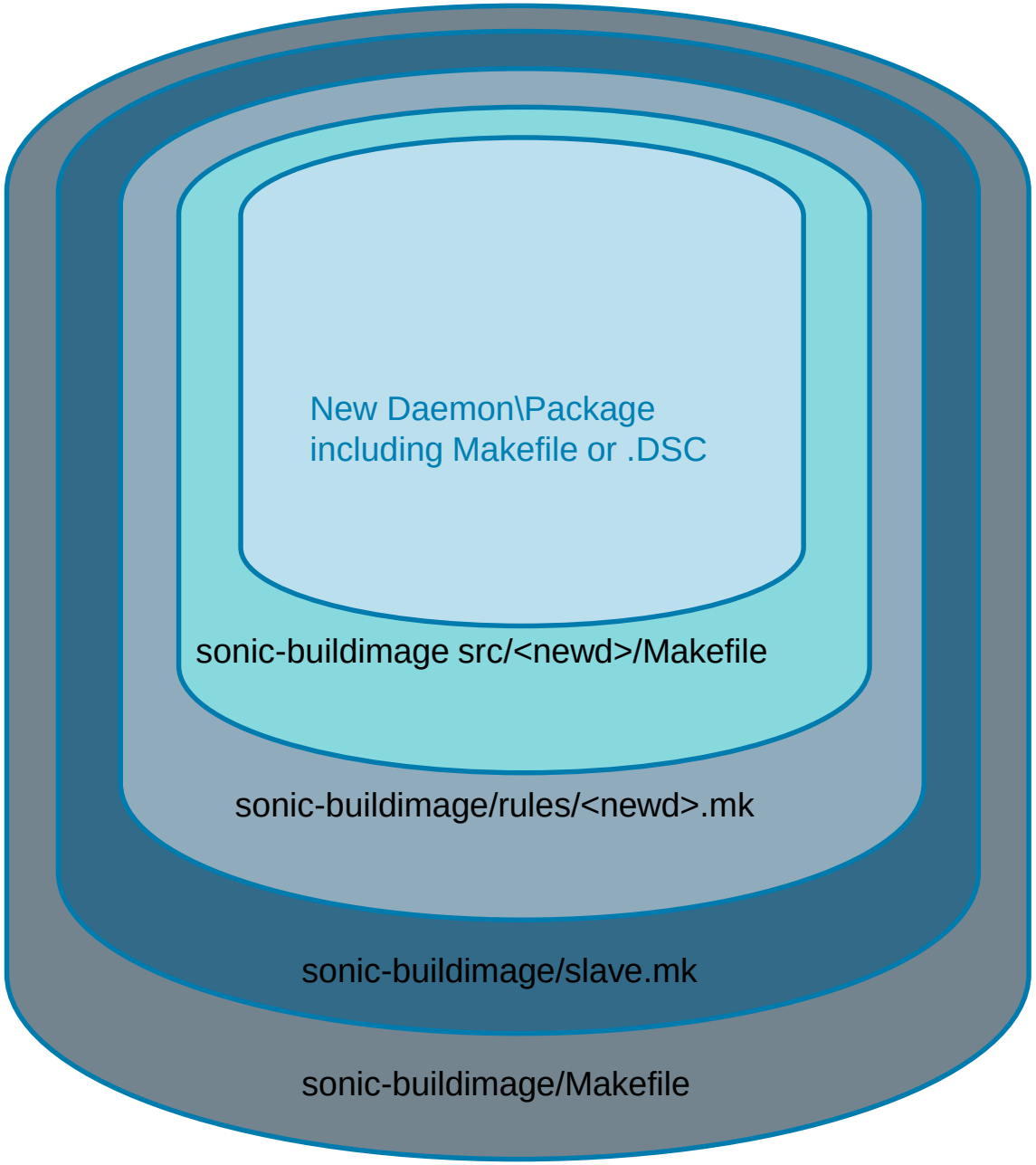


.gitignore	.gitignore: ignore vim swap files
.gitmodules	netlink: remove use of libnl3
.travis.yml	build: no fatal errors when compiling with embedded libevent on macOS
CONTRIBUTE.md	doc: add a CONTRIBUTE document
LICENSE	LICENSE: add title and copyright notice (#259)
Makefile.am	doc: add a license file
NEWS	daemon: fix creation of chroot directory
README.md	doc: update documentation for Android
autogen.sh	build: don't check for pkg-config several times
configure.ac	build: enable increased reliability of stack overflow detection
doxygen.am	Separate daemon and client code. Provide a client library.
doxygen.cfg	doc: fix doxygen warnings and don't parse include/linux
edit.am	daemon: don't use @mkdir_p@ substitution for systemd service file

```
MAIN_TARGET = $(NEWD)
DERIVED_TARGETS = $(LIBNEWD)

$(addprefix $(DEST)/, $(MAIN_TARGET)): $(DEST)/% :
    rm -rf ./newd

    # Clone newd repo
    git clone https://github.com/awesomeDev/newd.git
    pushd ./newd

    # Build source and Debian packages
    dpkg-buildpackage -rfakeroot -b -us -uc -j$(SONIC_CONFIG_MAKE_JOBS)
    popd

    # Move the newly-built .deb packages to the destination directory
    mv $* $(DERIVED_TARGETS) $(DEST)/

$(addprefix $(DEST)/, $(DERIVED_TARGETS)): $(DEST)/% : $(DEST)/$(MAIN_TARGET)
```

```
# newd package

NEWD_VERSION = 1.0.1

NEWD = newd_$(NEWD_VERSION)-0_amd64.deb
$(NEWD)_DEPENDS += $(LIBDEP)
$(NEWD)_RDEPENDS += $(LIBRDEP)
$(NEWD)_SRC_PATH = $(SRC_PATH)/newd
SONIC_MAKE_DEBS += $(NEWD)

LIBNEWD = libnewd_$(NEWD_VERSION)-0_amd64.deb
$(eval $(call add_derived_package,$(NEWD),$(LIBNEWD)))

# Export these variables so they can be used in a sub-make
export NEWD_VERSION
export NEWD
export LIBNEWD
```

Sonic Makefile Overview:




```
# Install new daemon version 1.0.1 with sonic base image.
sudo dpkg --root=$FILESYSTEM_ROOT -i target/debs/newd_*.deb || \
    sudo LANG=C DEBIAN_FRONTEND=noninteractive chroot $FILESYSTEM_ROOT apt-get -y install -f
```

files/build_templates/sonic_debian_extension.j2

Add Package with Host Image

```
{% elif 'docker-new-docker' in imagename %}
sudo LANG=C chroot $FILESYSTEM_ROOT docker tag {{imagename}}:latest {{imagename}}:{{new-docker-tag}}
```

files/build_templates/sonic_debian_extension.j2

```
# docker image for new daemon

DOCKER_NEW_DOCKER = docker-new-docker.gz
$(DOCKER_NEW_DOCKER)_PATH = $(DOCKERS_PATH)/docker-new-docker
$(DOCKER_NEW_DOCKER)_DEPENDS += $(NEWDD) $(LIBSWSSCOMMON) $(PYTHON_SWSSCOMMON)
$(DOCKER_NEW_DOCKER)_LOAD_DOCKERS += $(DOCKER_CONFIG_ENGINE)
SONIC_DOCKER_IMAGES += $(DOCKER_NEW_DOCKER)
SONIC_INSTALL_DOCKER_IMAGES += $(DOCKER_NEW_DOCKER)

$(DOCKER_NEW_DOCKER)_CONTAINER_NAME = newd
$(DOCKER_NEW_DOCKER)_RUN_OPT += --net=host --privileged -t
$(DOCKER_NEW_DOCKER)_RUN_OPT += -v /etc/sonic:/etc/sonic:ro
```

rules/docker-new-docker.mk

```
sonic-buildimage/dockers/docker-new-docker$ ls -l

total 12
-rwxrwxr-x 1 pchaudha pchaudha 1300 Mar  7 14:31 Dockerfile.j2
-rwxrwxr-x 1 pchaudha pchaudha 1870 Mar  7 14:31 start.sh
-rw-rw-r-- 1 pchaudha pchaudha  767 Mar  7 14:31 supervisord.conf
```

sonic-buildimage /dockers

Add Package in new docker

Summary:

To Become a Sonic Developer

1.) https://github.com/Azure/SONiC/wiki
2.) User Guide
3.) Developer`s overview Session(s).

Praveen Chaudhary
LinkedIn

pchaudhary@linkedin.com
https://github.com/praveen-li
https://sonicswitch.slack.com