gRPC support in ODIM

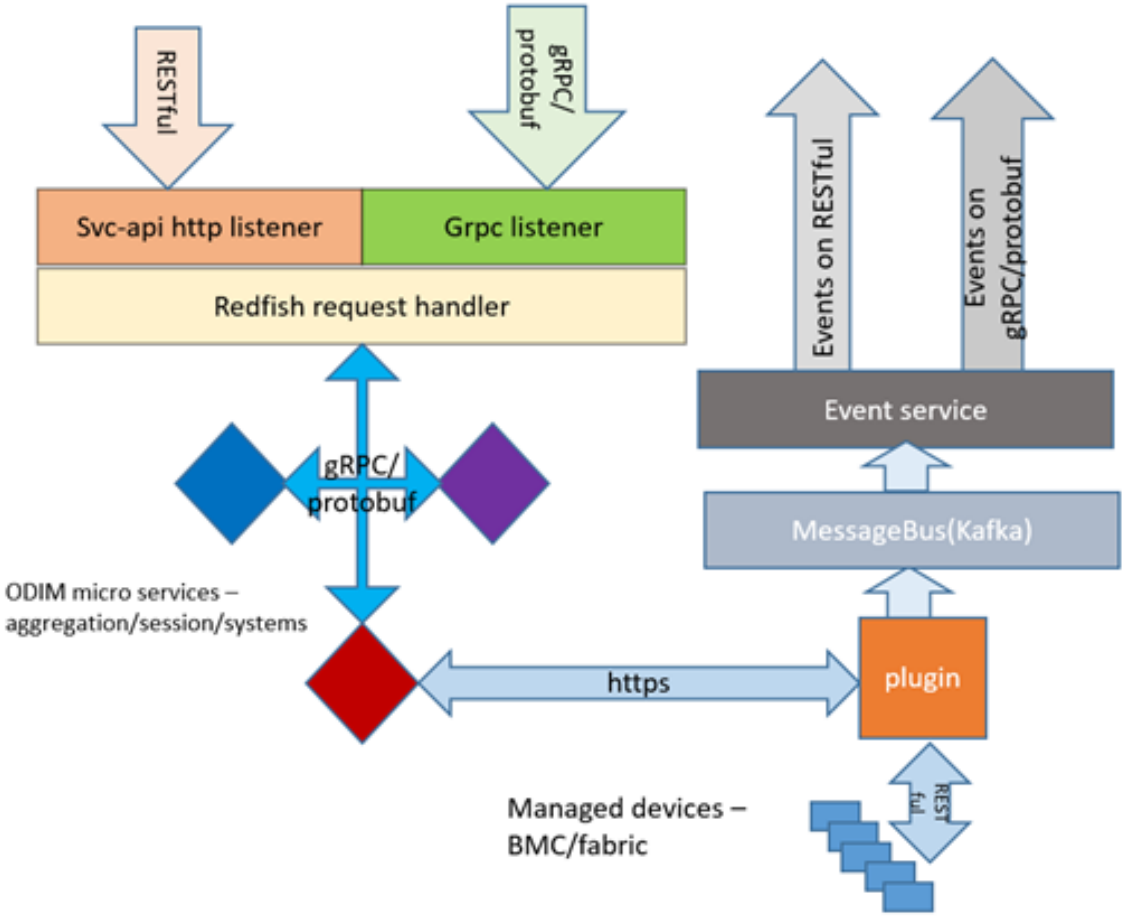
The Open Distributed Infrastructure Management (ODIM) framework provides a way to manage heterogeneous infrastructure resources using a single, standardized interface and data model. Infrastructure resources typically include stand-alone resources such as servers, storage and Ethernet networking equipment and composite resources such as racks, which in turn contain stand-alone resources. DMTF currently supports HTTPs to expose the redfish REST API to clients interacting with a redfish implementation. This proposal is to add gRPC as an alternative to HTTPs. Note here that gRPC is currently used by individual micro services in ODIMRA to communicate with each other.

gRPC is a modern open source high performance Remote Procedure Call (RPC) framework that can run in any environment. It can efficiently connect services in and across data centers with pluggable support for load balancing, tracing, health checking and authentication. It is also applicable in last mile of distributed computing to connect devices, mobile applications and browsers to backend services. gRPC can use protocol buffers as both its Interface Definition Language (**IDL**) and as its underlying message interchange format. By default, gRPC uses Protocol Buffers, Google's mature open source mechanism for serializing structured data (although it can be used with other data formats such as JSON). Salient features of gRPC include:

- Define your service using Protocol Buffers, a powerful binary serialization toolset and language.
- Automatically generate idiomatic client and server stubs for your service in a variety of languages and platforms.
- Install runtime and dev environments with a single line and also scale to millions of RPCs per second with the framework.
- Bi-directional streaming and fully integrated pluggable authentication with HTTP/2-based transport.

Refer to gRPC introduction for more details.

Adding support for gRPC will require implementing service for each of the redfish resource managed by ODIM. As there are a large number of resources exposed by redfish it is proposed to build a tool that shall generate proto files given the schema json files as input. A proto file will be generated for each redfish resource. The attributes in the request/response payloads will have to be modelled in the proto files along with the REST operations supported for each of these like GET/PUT/POST/PATCH/DELETE. Note the actions use POST payload and will be handled separately. Stubs have to be generated on client sides and on the server side implement code that will create redfish json payloads from the grpc request and invoke current redfish API handlers in the ODIM Resource Aggregator. The response sent back from ODIMRA will be in json format. These responses and error codes will be used to construct gRPC responses and sent back to the client by the gRPC server code. The interaction between the plugins and managed devices are kept out of this. The plugin implementers are free to use whatever protocol needed to communicate with the devices. However the core of ODIM will continue using redfish native model. The database will continue storing redfish payloads in json format.



Use cases

- A north bound client invokes redfish API for interacting with ODIMRA via gRPC instead of https

If a client invokes gRPC call the svc-api (ODIM API gateway) will that receive the request and handover the same to the appropriate micro service. The other components of ODIM and plugins for eg. micro services within ODIM RA, the plugins and the managed devices be they network/server/storage devices will not need changes.

1. The gRPC requests will terminate on the API service (svc-api) which will implement the interface defined for the service.
2. The svc-api will decode the service request parameters, create a context and add request parameters to a json payload
3. Forward the request to handlers which will forward the request to the right micro service using gRPC
4. Svc-api will receive the response from the micro services, recreate the grpc representation of the response and send back the response to the client.

- ODIM event service streams redfish events and telemetry reports using to clients subscribed to ODIM event service

gRPC is expected to have a much higher throughput of events hence it is proposed that ODIM provides event delivery to subscribers using gRPC as an option in addition to the https delivery. These events will be using the

1. The client shall subscribe to redfish events or telemetry reports using either of gRPC or https calls to the event service in ODIM
2. The event service will subscribe for the same with the managed devices.
3. The devices publish either of the event types to the plugin
4. Plugin publishes the same to the message bus.
5. the event service will pick up the events from the message bus, look up the subscribers for the event and forward the same to the subscriber(s) using gRPC

References

1. Redfish standards
2. gRPC
3. gRPC introduction
4. odim wiki