APPLICATION INTERFACES FOR THE RECONFIGURABLE WET PLANT

Shreya Gautam, Richard Kram, Jonathan Liss (SubCom LLC)
Email: sgautam@subcom.com

SubCom LLC, 250 Industrial Way West, Eatontown, NJ 07724

Abstract: As the Subsea Optical Transmission Cable industry continues in the direction of a vendor-neutral Open Cable model, Software Defined Network (SDN) based management solutions allowing customers to interface to all components of a system at the data/control orchestration layer become valuable. These minimize the need for multiple, vendor-specific Graphical User Interfaces (GUIs) for element management. In this paper, we provide an overview of the architecture and methods used to address this need. Application Programming Interfaces (APIs) use web technology and web services to support SDN orchestration for alarms, inventory, performance data view and control operations in a secure environment. The APIs manage functions and monitor the undersea cables and repeaters. Smart Undersea Network Elements (NEs), such as Wavelength Selective Switched (WSS) Reconfigurable Optical Add Drop Multiplexers (ROADMs) and optical switching branching units adapt to the transport needs through API controls. These Representational State Transfer (ReST) based APIs together with WebSocket alarms and notifications allow Operations Support Systems (OSS) management of the broader network, replacing manual operations. The APIs include ReST operations such as PUT, GET, POST and DELETE for control of the reconfigurable wet plant. These capabilities are especially important in open cable systems, where shared cables and multiple land network elements and vendors need to be orchestrated on several continents and multiple countries. APIs are essential for abstracted management of complex submarine cables. The server-based APIs may be hosted on public or private clouds, on physical computers, or on shelf mounted circuit packs. Local Element Management System (EMS) functionality is achieved in addition to API controls via the data communications network and internet/VPN. Customer partitioning, policy enforcement and certificate security from external locations are critical elements to enabling the overall architecture and managing the undersea plant from cloud or the station via APIs, and will also be discussed in this paper.

1. INTRODUCTION

Undersea cable management systems traditionally have used GUIs for network management operations. The need to move to an SDN orchestrator-based model has become imperative with the transition to open cable systems, which provide better machine to machine communication and reduce the need for manual intervention. ReST based APIs, along with Websockets, have been utilized for such an SDN based network management solution. Since these systems utilize web technology, they can be hosted on public or private clouds or on shelf mounted embedded processor units.

The implementation utilizes microservices for improved availability. For a monolithic GUI based design, existing systems experiencing one failure can cause the entire application to crash and disrupt management functions. APIs provide health checks for each component of the service to facilitate the availability of all the components, circumventing the effects of a single failure. Customer GUI clients can easily be built using ReST APIs and Websockets, providing a flexible base and the ability to host such services on the web.

2. IMPLEMENTATION

2.1 Architecture

To provide an interface at the data level to the customers' Operations Support System (OSS)/Network Management System (NMS) client, two components are needed: ReST APIs and Websockets. Figure 1 shows the overall architecture of the system. The APIs follow a microservices architecture with each NE having its own service. An API Gateway sits on top of these services to provide seamless and abstracted operations to the client.

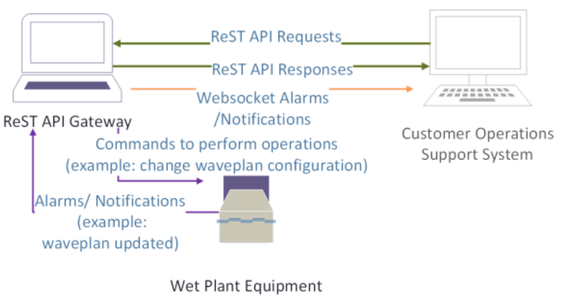

Figure 1: Architecture for API Control of Wet Plant

To reduce customer complexity, a single point of access is provided. Client applications send ReST requests to the API Gateway, which routes them to the appropriate underlying ReST Services based on runtime configurable routing logic. Clients can also connect to the Gateway to receive WebSocket messages, notifications and events from any of the underlying ReST Services. For systems equipped with multiple gateways for redundancy, a recommended priority list of Uniform Resource Locators (URLs) is provided.

Additionally, the API Gateway handles contingency situations for a variety of failure modes. If the rate of ReST requests of a particular type reaches a configured threshold, the gateway is able to throttle the forwarding of requests to underlying ReST services, so as to not overwhelm them. When an underlying ReST service is not able to handle incoming requests, a "Circuit Breaker" functionality will stop forwarding requests to that service until it recovers. Client-side load-balancing capabilities in the API Gateway assess where to route a request to instances of a particular ReST service based on load feedback from those instances.

Figure 2 shows two instances of API Gateway for redundancy. Each instance can receive ReST requests from clients, apply routing logic and forward them to underlying NE services. Each Gateway registers with all underlying NE services to receive Webockets and forward them to the customer client applications that registered for receiving WebSocket messages.

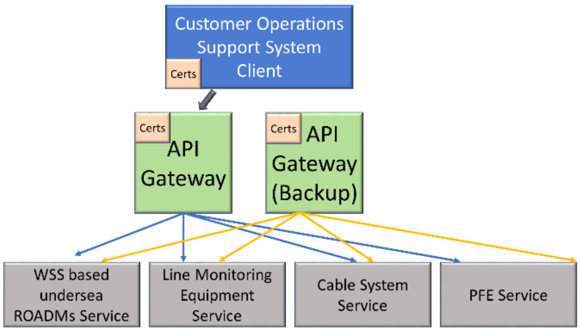

Figure 2: API Gateway Design

2.2 REST APIS

ReST APIs provide control of the reconfigurable wet plant, including wavelength provisioning, recovery, inventory management, performance data view and measurements.

ReST APIs can be divided into categories for better accessibility as per the example in Figure 3. For instance, they can be divided into wet plant, dry plant and cable system including fiber pair filtering, with further drill down and details as required. Such a division allows each part to be accessed by its specific URL.

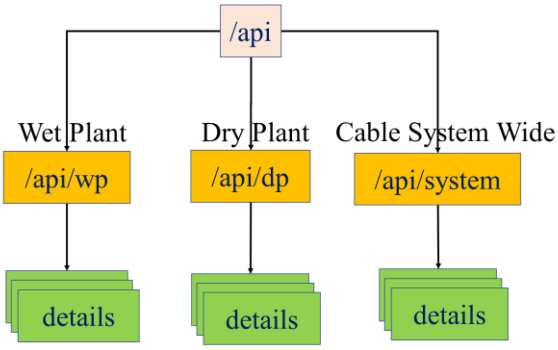

Figure 3: Division of ReST API Calls

2.3 WEBSOCKETS

Websockets provide support for real time alarms and event management. Websockets for each NE are divided into categories, each of which can be subscribed to. There are common categories like alarms and Attribute Value Change (AVC) events along with specific categories for individual NEs. To subscribe to the AVC events in the wet plant, the corresponding topic can be subscribed to after registering to the wet plant API URL. It is also possible to subscribe to all the Websockets.

2.4 BACKWARDS COMPATABILITY

Backwards compatibility is essential to minimize client disruption when updates to APIs are brought on line. API URLs can be specified with a version number or can default to the latest, thus maintaining client integrity. In this way, clients can access any API version at run time.

2.5 CLIENT-SIDE CODE EASE FROM DOCUMENTATION

The detailed documentation for each call in the APIs and all Websocket subscriptions is in industry standard Swagger. It allows for the creation of skeleton code from the requirements, making it easier for client-side code development. The specifications are also available via a ReST call for the specific NE.

2.6 SECURITY CONSIDERATIONS

The ReST APIs and Websockets are secured with mutual authentication via certificates. JSON Web Token (JWT), where JSON stands for JavaScript Object Notation, is provided to the client after a successful authentication, which is used until the token expires. Multiple customers using a single ReST server could authenticate to the shared server with their own set of certificates. This is facilitated by the server providing individual port numbers to each customer secured with the corresponding set of certificates. This allows the customer to replace the default certificates provided with the system with those that are trusted for their domain.

2.7 PARTITIONING AND SESSION MANAGEMENT

After successful authentication, the JWT is used to perform secure ReST operations and to subscribe to Websockets. When such requests are received by the server, clients are only able to view/control the resources they are authorized to access. This is known as partitioning. Thus, the ReST APIs filter responses to provide a partitioned view, not only for different customers sharing the cable system, but also (if desired) for different users for each customer. This facility is provided by the ReST APIs and Websockets through interactions with a RADIUS authentication server containing user and domain information.

Session management is also important in a system with multiple users, with and without partitioning. This becomes even more imperative in a remote SDN environment, as simultaneous collisions can disrupt traffic leading to major downtimes. ReST APIs pose a greater challenge, as they are stateless and thus persistency or sessions cannot be maintained at the user end. Necessary checks validate the sessions held for individual NEs with corresponding JWTs, thus performing session management operations.

2.8 SYSTEM HEALTH

In addition, it is also essential that the customer is notified of the health of the system either periodically or on demand. Websockets provide notifications of a valid open connection through a keep-alive ping-pong message protocol. Additionally, ReST APIs provide the status of the health of each of the microservices running on the server, one per individual NE, when a call is made to /api/health.

3 API PROVISIONS

The microservices-based architecture allows provision of APIs accessing NEs and functions at any cable station without the need for a traditional computer server. This embedded solution also allows clients to add their own functionalities, if desired.

Alternatively, the APIs can be hosted on many platforms, like traditional computer servers or a cloud.

Some APIs/Websockets are NE specific and some are system related, such as Line Monitoring System (LMS) and Wet Plant access. The former provides direct control to the NEs while the latter requires proprietary topological algorithms or telemetry gateways to manage the functions. The microservices approach allows for an abstraction of some system-wide algorithms enabling an embedded solution and providing the clients with direct access to the hardware APIs.

4 SIMULATION AND TESTING

These API/Websocket solutions naturally enable a simulation environment with software-based tools. This “development environment,” or test bench, could be used during client testing phases, mitigating the need for access to “real systems” and “real hardware” during the client development phase. A simple plugin of NE software simulators creates instances of API/Websocket simulation environments that may be hosted in public clouds or delivered to client developer locations, thus reducing the client development lifecycle, costs and time to market.

5 BENEFITS

The move to SDN driven management with ReST based APIs and Websockets has many benefits.

APIs are platform independent since ReST APIs and Websockets are technology agnostic, as they are ubiquitous web services. They can be deployed anywhere from physical servers to private or public clouds, or even on shelf-mounted circuit packs, and do not have to be collocated with the actual NEs (like cable stations, for example.)

By utilizing central managers, APIs provide control of the entire cable system, eliminating the need for using local EMS.

For customers desiring element management, the NE-centric structure of the APIs provides individual NEs access as needed.

Microservices architecture allows all running services to be available independent from services that may be down. In monolithic applications, one failure can result in the loss of the entire application. In addition, microservices enable quicker installation and updates for the software, minimizing potential management down time.

APIs make it easy for customer OSS’ to control these services as they desire. Client-side agents can be incorporated as easily as building client-side GUIs.

In open cable systems, APIs provide an easy integration with other OSS’/NMS’, thus providing customers with a "single pane of glass" view.

The combination of ReST and Websockets allows a customer OSS to orchestrate network management on any platform without manual intervention and with significant flexibility.

6 CONCLUSION

This paper discussed a new flexible approach to providing the undersea cable industry with access to network management tools at the data layer secured with transport layer security mechanisms and mutual authentication with certificates. It discussed key components of the architecture along with benefits of moving to this approach, especially for open cable systems.