Contents

1 Document Purpose

This document establishes a working definition of user needs, including the needs of the clinician and patient, associated with the OMNI Summit Microservice.

This document provides a working definition of the software architecture and detailed design elements of the OMNI Summit Microservice and associated support libraries.

2 Document Scope

This document addresses the architecture and design of the OMNI Summit Microservice only. Additional details may be found in software source files.

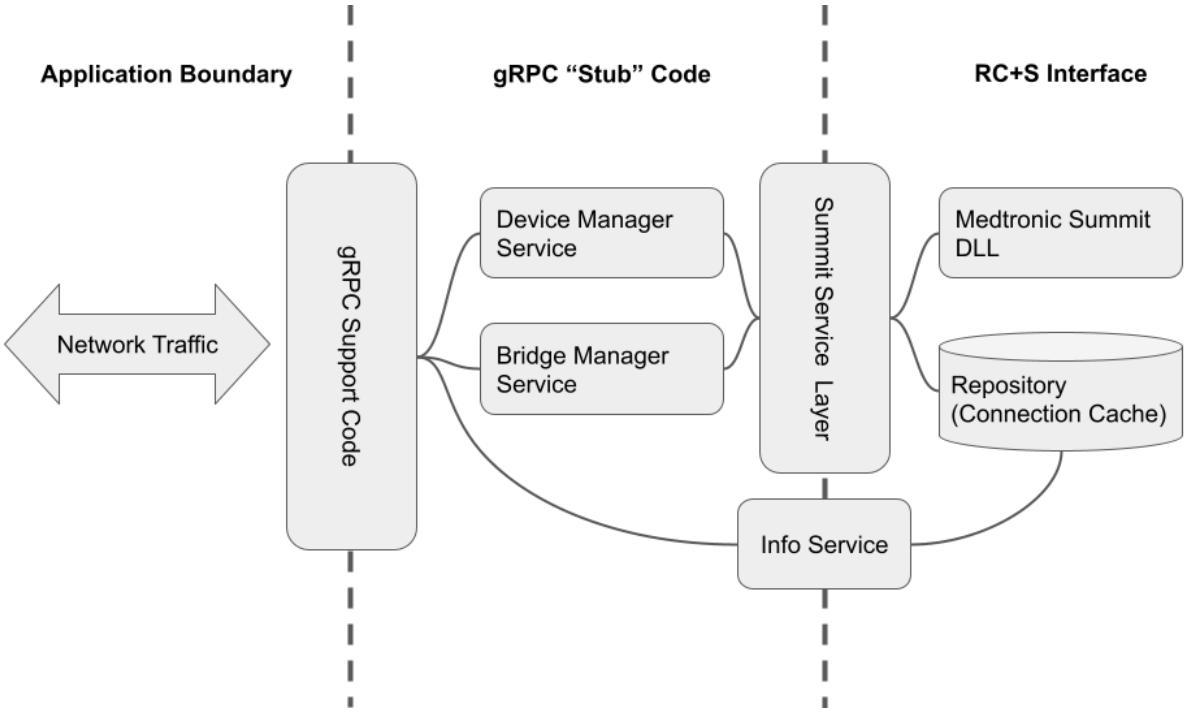
3 Background

Today, neural stimulation applications are written in such a way that little to no code-reuse is possible. The purpose of the OMNI Summit Microservice is to provide a hardware-agnostic API layer to manage connections between Summit CTM and Summit RC+S.

The OMNI Summit Microservice achieves this goal by leveraging gRPC, a remote procedure call library based on HTTP/2 protocol. OMNI Summit Microservice uses the Protobuf interface description language to define messages and their associated types.

4 Architecture

The OMNI Summit Microservice consists of three APIs: Bridge Manager API, Device Manager API and Info API. The Bridge Manager API controls connections to the Summit CTM, and provides robust connection management. The Device Manager API manages connections with the Summit RC+S, and streaming data from the Summit RC+S. The Info API provides some debug information about the OMNI Summit Microservice.



User input comes from the network via gRPC. The gRPC library for C# provides code for handling the network requests and parsing them. gRPC also automatically generates stub classes for each of the services defined in the Protobuf interface description documents. OMNI Summit Microservice overrides the methods on the autogenerated classes to inject the desired implementation. When a network request is received the gRPC library spawns a thread to handle the request. All exceptions in the network thread are caught and sent back to the client application. This ensures that if there's an error processing a request the OMNI Summit Microservice will not crash.

The Bridge and Device Manager services are supported by the Summit Service Layer. The Summit Service Layer orchestrates all calls to the Summit DLLs. The repository stores any active connections, as well as the connection information for both Bridges and Devices. The Info service provides debug data to the developer, such as the supported devices and version of OMNI Summit Microservice. The Info service can also see what active connections are stored in the Repository.

The gRPC library orchestrates the operational mode requirements (OMNI-SM-002). The service implementation classes are provided to the gRPC library through dependency injection (i.e. passed through to the constructor) of the gRPC server object.

The command subsystem of the OMNI Summit Microservice is divided between the Bridge Manager, Device Manager and Info services. Support commands are implemented in the Info service. Bridge query, management, configuration and status streaming are all implemented in the Bridge Manager service. Connection monitoring is implemented in the Bridge Manager service as well. Device query, management, configuration, and data streaming are all implemented in the Device Manager service. The number of users supported are a limitation of the implementation; the responsibility is shared between the Bridge and Device Manager services. The computing platform is specified by the Summit RC+S DLLs. Network listening address configuration is configured when the application first starts.

The service execution begins with the code entrypoint of the server object, which then instantiates the provided microservices embodied in the OmniServer class.

5 Programming Classes Description

Following the architecture defined above, we designed the following classes with associated functionality.

5.1 gRPC Server Object

The server class functions as the entrypoint for the OMNI Summit Microservice. The Server class starts the OmniServer and waits for network requests.

5.2 OmniServer Class

The OmniServer class wraps the gRPC server object and provides a few convenience functions for setting the network interface to listen on. The supported devices of the OMNI Summit Microservice are also defined in the OmniServer class. This class was devised to facilitate the automated unit and integration tests.

5.3 Repository

The Repository class stores information about active Summit RC+S connections, as well as connection information for Bridges and Devices. The Repository is passed to the OmniServer's constructor.

The Repository class is named after the repository design pattern.

5.4 SummitService

The SummitService class provides a translation layer between the incoming gRPC network requests and the Summit DLL calls. All methods on the SummitService class expect gRPC messages as input, and provide gRPC messages as output. All interactions with the Summit DLLs happen in this class.

The SummitService class is named after the service design pattern. Due to the architecture of the Summit RC+S DLLs, both CTM and INS share functionality, encapsulated in the SummitSystem class. In order to cleanly separate the bridge and device functionality, a service layer is used for all interactions with the Summit DLLs.

5.5 Services

The service classes are implementations of the autogenerated service classes the gRPC library provides. Each method in the service class corresponds to an endpoint in the service definition protobuf file.

5.5.1 BridgeManagerService

The BridgeManagerService class handles all the bridge related functionality. Each method calls a corresponding method in the SummitService class.

5.5.2 DeviceManagerService

The DeviceManagerService class handles all the device related functionality. Each method calls a corresponding method in the SummitService class.

5.5.3 InfoService

The InfoService class handles all debug information. In contrast to both the BridgeManagerService and DeviceManagerService classes, the InfoService does not make any calls to the SummitService class.

5.6 Testing Wrappers

The object implementation of C# uses interfaces to define a set of functionality. OMNI Summit Microservice leverages interfaces to mock functionality of the Summit DLL objects. Due design constraints in C#, a separate interface is required to mock out system functionality when running automated tests. This enables automated testing by mocking different behaviors of the SummitManager and SummitSystem.

5.6.1 ISummitManager

The ISummitManager interface exposes the methods used on the SummitManager class found in the Summit DLLs. This includes creating summits and getting different telemetry. The methods defined on the interface are a clone of the methods on the SummitManager object.

5.6.2 ISummitSystem

The ISummitSystem interface exposes the methods used on the SummitSystem class found in the Summit DLLs. This includes configuring the device and reading data from the device. The methods defined on the interface are a clone of the methods on the SummitSystem object.

5.6.3 OmniManager

The OmniManager is a wrapper class for the SummitManager class. Any method called on the OmniManager class is passed to the underlying SummitManager class. This concrete implementation is used in the microservice when not in testing mode. When in testing mode, OMNI Summit Microservice creates a mock SummitManager by using the ISummitManager interface.

5.6.4 OmniSystem

The OmniSystem is a wrapper class for the SummitSystem class. Any method called on the OmniSystem class is passed to the underlying SummitSystem class. This concrete implementation is used in the microservice when not in testing mode. When in testing mode, OMNI Summit Microservice creates a mock SummitSystem by using the ISummitSystem interface.

5.7 URINameHelpers

The URINameHelpers class contains a collection of static methods to help translate gRPC resource identifiers into the bridge, device, and full names.

6 Interfaces

OMNI Summit Microservice is a network only service. All interactions between the client and the OMNI Summit Microservice occur over gRPC. All gRPC services, endpoints, and messages are defined in the Protobuf interface description files.

7 SOUP

OMNI Summit Microservice considers the Summit RC+S DLLs software of unknown provenance.

8 Detailed Design

Autogenerated by Doxygen.

9 Reference Information

9.1 Definitions

Term	Definition
gRPC	gRPC Remote Procedure Call framework (grpc.io)

10 Approvals

Approver Role	Signature and Date
Project Lead	DocuSigned by: David Borton 18A589B8FA1F246E... 5/19/2022 | 3:21:38 PM EDT
Quality	DocuSigned by: Timothy Denison 08/05/2022 | 10:03 AM EDT
Lead Developer, Author	DocuSigned by: Jeffrey Herron 5/19/2022 | 10:26:55 AM PDT