

# EDGE Process Automation Interface

## Overview

---

The purpose of this document is to provide an ICD (Interface Control Document) for the Ensign EDGE Automation Interface. The purpose is to communicate all possible inputs to and all potential outputs from the EDGE system for some potential or actual user of the system. This document describe the following

- The inputs and outputs of the EDGE Process Automation Interface.
- The interface between two systems or subsystems, e.g. "The Edge control system and, but not limited to, the Shell Auto Tuner"
- The complete interface protocol from the lowest physical elements (e.g., Profinet) to the highest logical levels (e.g., EDGE Process Automation Interface).

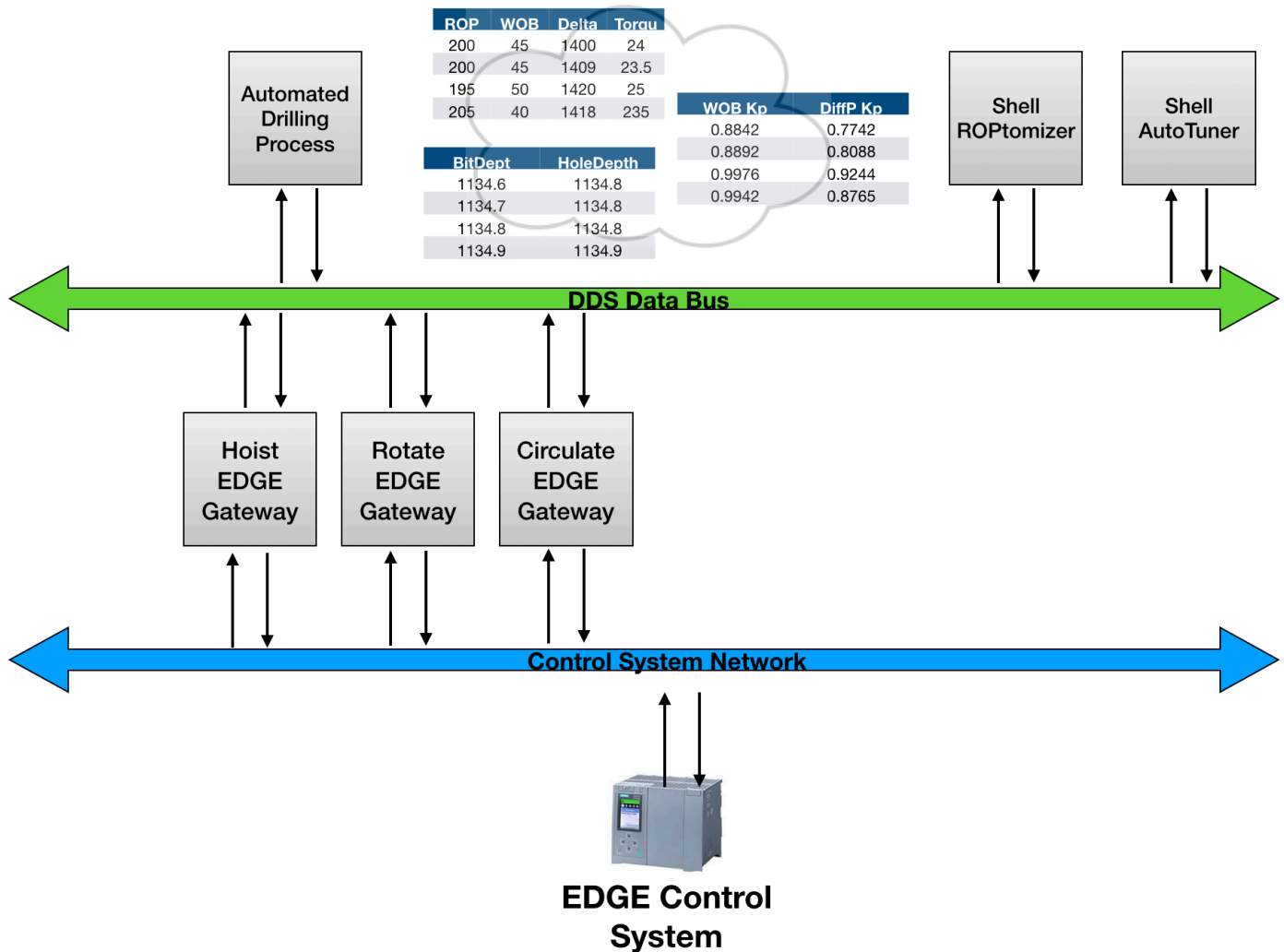
This ICD will only describe the interface itself, and not the characteristics of the systems which use it to connect. The function and logic of those systems will be described in their own design documents if required. In this way, independent teams can develop the connecting systems which use the interface specified, without regard to how other systems will react to data and signals which are sent over the interface. This ICD will include information about the type, description, and unit of what is measured by the data, but not any ultimate meaning of the data in its intended use by any user.

## Communication

---

All communication between the EDGE control system and the Shell optimization applications will be expressed as normalized topics which use the DDS (Data Distribution Service) data-centric protocol to provide asynchronous communication. Certain QoS (quality of service) attributes will be available for to the optimization applications to determine the quality of the data it is receiving as well as the state of the data it has requested. These QoS attributes will allow the applications to specify specific events (or timers) for notification of data arrival, deadlines, etc. A communication pattern known as Objective/State (similar to observer pattern) will be used to distribute objectives to the Rig Control System.

For safety reasons all process level requests to the system are merely suggestions and the EDGE control system will decide if the request is within the specified safety margins.



## Objective/State

The objective/state pattern utilizes two actors. One actor represents a requestor and the other represents an effector. There can be a many-to-many relationship between requestors and effectors. This allows the communication between the Rig Control System (the control layer) and the Edge Process Automation Interface (the process layer) to be data-centric. It also allows the process layer to remain agnostic about the underlying control system. Focusing on the functional aspects of the system rather than detailed knowledge of how the Rig Control system functions. Allows components to express intent and communicate about actions

- Allows additional observing components to leverage request, objective data
- Avoids tight application-to-application coupling of RPC-like (anti-)patterns

## Objective State Pattern: Use Cases

- How can one party request that another party do something?

- How can a producer know that observers(s) have acted on its data?
- How can a producer observe the “result” of a remote action?

## **Requestor**

A requestor is an actor which "requests" the resources of another actor. A requestor makes a "request" by stating its intentions or publishing the intended state to the system. The request is completely asynchronous which means that the requestor must observe the current state of the system to see if the request is being handled by the system. Requestors perform the following responsibilities:

- publish requests to the system
- observe current state of the system

## **Effector**

The effector is an actor which can satisfy the state of a requestor. Effectors perform the following responsibilities:

- observe intended state changes
- publish the current state of the effector

## **QoS**

The following QoS policies will be supported in the first release of the interface. These QoS policies are configurable at runtime. The policies should not be baked into the code but set by the consumer of the data.

### **Reliability Policies**

Reliability controls the guarantee of packet delivery as well as historical persistence.

For some use cases, such as the periodic update of sensor values to a GUI displaying the value to a person, "best effort" delivery is often good enough. It is certainly the fastest, most efficient, and least resource-intensive (CPU and network bandwidth) method of getting the newest/latest value for a topic from DataWriters to DataReaders. But there is no guarantee that the data sent will be received. It may be lost due to a variety of factors, including data loss by the physical transport such as wireless RF or even Ethernet. Packets received out of order are dropped and a `SAMPLE_LOST` Status is generated.

However, there are data streams (topics) in which you want an absolute guarantee that all data sent by a DataWriter is received reliably by DataReaders. This means that Connex DDS must check whether or not data was received, and repair any data that was lost by resending a copy of the data as many times as it takes for the DataReader to receive the data.

- *Reliability* - Specifies whether or not DDS will deliver data reliably.
- *History* - Specifies how much data must be stored by the middleware.

## Durability Policies

Because the publish-subscribe paradigm is connectionless, applications can create publications and subscriptions in any way they choose. As soon as a matching pair of DataWriters and DataReaders exist, then data published by the DataWriter will be delivered to the DataReader. However, a DataWriter may publish data before a DataReader has been created. For example, before you subscribe to a magazine, there have been past issues that were published.

This QoS Policy can be used to help ensure that DataReaders get all data that was sent by DataWriters, regardless of when it was sent. This QoS Policy can increase system tolerance to failure conditions. The following options are available for durability:

- *Volatile Durability* - Do not save or deliver old data samples.
- *Transient Local* - Save and deliver old data samples if the publishing application still exists.
- *Transient Durability* - Save and deliver old data samples using a memory-based service.
- *Persistence Durability* - Save and deliver old data samples using disk-based service.

## QoS Events

- *OnDataAvailable* - Mechanism to notify a subscriber when new data is available for reading.
- *OnLivelinessLost* - Mechanism to notify a subscriber that a producer of data is no longer valid. Does not tell us why the DataWriter is no longer available.
- *OnDeadlineMissed* - Mechanism to notify a subscriber that the expected deadline for a new data sample has expired.
- *OnDataDisposed* - Mechanism to notify a subscriber that the DataWriter has disposed of the topic instance. This instance will never be seen again
- *OnPublicationMatched* - Mechanism to notify a subscriber that a new publisher is available.

## OnDataAvailable

The *OnDataAvailable* QoS policy requests a specific mechanism to be notified when new samples of data arrive. The subscribing application may want to act immediately on new data samples. This mechanism allows the subscribing application to base arrival of data on events rather than traditional polling mechanisms. The `SAMPLE_INFO` structure will inform the subscribing application about the validity of the data sample.

## OnLivelinessLost

The *OnLivelinessLost* QoS policy requests a specific mechanism for the publishing application to maintain the

liveliness of all subscribing entities. The subscribing application may want to know that the publishing application is explicitly asserting the liveliness of the matching DataWriter rather than inferring its liveliness. The subscribers *leaseduration* specifies the maximum period at which matching Publishers must have their liveliness asserted. In addition, in the subscribing application DDS uses an internal thread that wakes up at the period set by the subscribers *leaseduration* to see if the publishers *lease\_duration* has been violated.

### **OnDeadlineMissed**

This *OnDeadlineMissed* QoS policy states the maximum period in which the application expects to receive new values for the Topic. The application may receive data faster than the rate set by this QoS policy. You can use this QoS policy during system integration to ensure that applications have been coded to meet design specifications. You can also use it during run time to detect when systems are performing outside of design specifications. Receiving applications can take appropriate actions to prevent total system failure when data is not received in time. For topics on which data is not expected to be periodic, the deadline period should be set to an infinite value.

### **OnPublicationMatched**

This *OnPublicationMatched* QoS policy informs the subscriber that a publisher is now available in the system. This QoS policy can be used to inform the application the state of the system. If there is a publisher available

## **Safe Operation**

---

To ensure safe operation of the control system the platform solution must ensure that there is only every a single owner of the process and/or tool. The platform must also ensure that any set-points presented to the system which fall outside the normal operating range will be disregarded by the system.

### **Single tool owner**

To ensure safe operations of the system there can only ever be a single owner of the rig control system. The driller MUST be presented a mechanism to easily switch between application and manual input.

### **Validation of set-points**

Validation of set-points will take place at multiple levels. First it is the responsibility of the optimization application to only send values within the constraints specified in the *State* interface. Second the networker will be responsible for validating any input it receives from the application. If a value is determined to be outside the safe operating limits then the networker will disregard the suggested set-points and not send them to the control system. Finally it will be the responsibility of the control system to vet all set points it receives to ensure the values are within safe tool limits.

# Interfaces

This section details the interfaces that are available in the EDGE Process Automation interface. Interfaces are broken into two categories *Request* and *State*. *Requests* are used to infer state to the environment while *State* interfaces are used to observe a stateful view of the environment.

*Request* and *State* always have the following framework data members *id*, *objective id*, *estimated duration* , and *timestamp*. The *id* is the unique id of the requester. The *objective id* is the unique id of the current rig state objective. The *id* is the primary key of the topic and is used to differentiate this topic from other topics of the same type. The *timestamp* is the time the data sample was published. At the time of delivery the framework will take the timestamp from the sample metadata that was produced by the publisher and store it in the sample. This is done to allow the consumer of the data the opportunity to make queries on the samples based on a specified time series. The *estimated duration* is the expected time the requestor will need the resource.

## Units

All units are specified in derived SI. The current SDK forces an explicit unit convresion to the proper type at compile time. This mechanism ensures that the data is always expressed in the proper unit when it is put on the data-bus. Specifying a standard base unit system allows the consumer of the data to easily and dynamically determine a unit without affecting the interface.

## Drilling Calibration Request Interface

The drilling calibration request interface is an ephemeral interface which issued by a requestor to modify the current state of the system. The state of the interface will be published only when the publishing applications requests.

Variable	Description	Type	SI Unit
wobProportional	Actual measured weight on bit.	double	N/A
wobIntegral	Actual measured weight on bit.	double	sec
differentialPressureProportional	Actual measured differential pressure.	double	N/A
differentialPressureIntegral	Actual measured differential pressure.	double	sec
torqueProportional	Actual measured surface torque.	double	N/A
torqueIntegral	Actual measured surface torque.	double	sec

## Drilling Calibration Request Topic

```
struct DrillingCalibrationRequest
{
    double    wobProportional;
    double    wobIntegral;
    double    differentialPressureProportional;
    double    differentialPressureIntegral;
    double    torqueProportional;
    double    torqueIntegral;
};
```

## Drilling Calibration Request QoS

The following QoS events are available for any observer of the Drilling State Calibration Topic.

Qos Policy	Description
OnLivelinessLost	Event to notify subscriber that producer of data is no longer valid.
Reliability	Best Effort
Durability	Volatile Durability
History	Only preserve the last sample

## Drilling State Calibration Interface

The drilling state calibration interface is a persistent interface which is guaranteed to be available by the system at all times regardless of the current system objective. The state of the interface will be published only when any of the interface contents change. This interface is available for consumption by any actor of the system interested in observing the current state calibration parameters of the EDGE autodriller. The interface is guaranteed to publish only at a frequency equal to or less than that of the underlying control system.

Variable	Description	Type	SI Unit
wobProportional	Weight on bit proportional.	double	N/A
minWobProportional	Minimum weight on bit proportional.	double	N/A
maxWobProportional	Maximum weight on bit proportional.	double	N/A
wobIntegral	Weight on bit integral.	double	sec
minWobIntegral	Minimum weight on bit integral.	double	sec
maxWobIntegral;	Maximum weight on bit integral.	double	sec
differentialPressureProportional	Differential pressure proportional.	double	N/A
minDifferentialPressureProportional	Minimum differential pressure proportional.	double	N/A
maxDifferentialPressureProportional	Maximum differential pressure proportional.	double	N/A
differentialPressureIntegral	Differential pressure integral.	double	sec
minDifferentialPressureIntegral	Minimum differential pressure integral.	double	sec
maxDifferentialPressureIntegral	Maximum differential pressure integral.	double	sec
torqueProportional	Torque proportional.	double	N/A
minTorqueProportional	Minimum torque proportional.	double	N/A
maxTorqueProportional	Maximim torque proportional.	double	N/A
torqueIntegral	Torque integral.	double	sec
minTorqueIntegral	Minimum torque integral.	double	sec
maxTorqueIntegral	Maximum torque integral.	double	sec

## Drilling State Calibration Topic



```

struct DrillingStateCalibration
{
    double    wobProportional;
    double    minWobProportional;
    double    maxWobProportional;
    double    wobIntegral;
    double    minWobIntegral;
    double    maxWobIntegral
    double    differentialPressureProportional;
    double    minDifferentialPressureProportional;
    double    maxDifferentialPressureProportional;
    double    differentialPressureIntegral;
    double    minDifferentialPressureIntegral;
    double    maxDifferentialPressureIntegral
    double    torqueProportional;
    double    minTorqueProportional;
    double    maxTorqueProportional
    double    torqueIntegral;
    double    minTorqueIntegral;
    double    maxTorqueIntegral;
};

```

## Drilling State Calibration QoS

The following QoS events are available for any observer of the Drilling State Calibration Topic.

Event	Description
OnDataAvailable	Event to notify subscriber new data is available for reading.
OnLivelinessLost	Event to notify subscriber that producer of data is no longer valid.
OnDeadlineMissed	Event to notify subscriber that expected deadline for new data sample has passed.
Reliability	Best Effort
Durability	Volatile Durability
History	Only preserve the last sample

## Drilling Request Interface

The rotate request interface is an ephemeral interface which issued by a requestor to modify the current state of the system. The state of the interface will be published only when the publishing algorithm requests.

Variable	Description	Type	SI Unit
ropLimit	Maximum configured rate of penetration.	double	m/sec
wobLimit	Maximum configured weight on bit.	double	N
differentialPressureLimit	Maximum configured differentail pressure.	double	Pa
torqueLimit	Maximum configured surface torque.	double	Nm
ropMode	Rate of Penetration mode enabled.	boolean	N/A
wobMode	Weight on bit mode enabled.	boolean	N/A
differentialPressureMode	Differential Pressure mode enabled.	boolean	N/A
torqueMode	Torque mode enabled.	boolean	N/A

**Drilling Request Topic**

```
struct RotateRequest
{
    double    ropLimit;
    double    wobLimit;
    double    differentialPressureLimit;
    double    torqueLimit;
    double    ropMode;
    double    wobMode;
    double    differentialPressureMode;
    double    torqueMode;
};
```

**Drilling Request QoS**

The following QoS events are available for any observer of the Drilling State Calibration Topic.

Qos Policy	Description
OnLivelinessLost	Event to notify subscriber that producer of data is no longer valid.
Reliability	Best Effort
Durability	Volatile Durability
History	Only preserve the last sample

## Drilling State Interface

The drilling state interface is an ephemeral interface which is only published by the system when the AutoDrill functionality of the Edge control system is enabled. The state of the interface will be published only when any of the interface contents change. This interface is available for consumption by any actor of the system interested in observing the current state of the drilling process. The interface is guaranteed to publish only at a frequency equal to or less than that of the underlying control system.

An observer of the system will know that the drilling objective is not being processed when the this interface is not present in the system.

Variable	Description	Type	SI Unit
ropActual	Actual measured rate of penetration.	double	m/sec
wobActual	Actual measured weight on bit.	double	N
differentialPressureActual	Actual measured differential pressure.	double	Pa
torqueActual	Actual measured surface torque.	double	Nm
ropLimit	Maximum configured rate of penetration.	double	m/sec
wobLimit	Maximum configured weight on bit.	double	N
differentialPressureLimit	Maximum configured differential pressure.	double	Pa
torqueLimit	Maximum configured surface torque.	double	Nm
ropMode	Rate of Penetration mode enabled.	boolean	N/A
wobMode	Weight on bit mode enabled.	boolean	N/A
differentialPressureMode	Differential Pressure mode enabled.	boolean	N/A
torqueMode	Torque mode enabled.	boolean	N/A
ropTarget	Target rate of penetration.	double	m/sec
wobTarget	Target weight on bit.	double	N
differentialPressureTarget	Target differential pressure.	double	Pa
torqueTarget	Target surface torque.	double	Nm

## Drilling State Topic

```

struct DrillingState
{
    double    ropActual;
    double    wobActual;
    double    differentialPressureActual;
    double    torqueActual;
    double    ropLimit;
    double    wobLimit;
    double    differentialPressureLimit;

    double    torqueLimit;
    boolean    ropMode;
    boolean    wobMode;
    boolean    differentialPressureMode;
    boolean    torqueMode;
    boolean    bitOnBottom;
};

```

## Drilling State QoS

The following QoS events are available for any observer of the Drilling State Topic.

Event	Description
OnDataAvailable	Event to notify subscriber new data is available for reading.
OnLivelinessLost	Event to notify subscriber that producer of data is no longer valid.
OnDeadlineMissed	Event to notify subscriber that expected deadline for new data sample has passed.
OnPublicationMatched	Event to notify subscriber that a new product of data is available.
Reliability	Best Effort
Durability	Volatile Durability
History	Only preserve the last sample

## Wellbore Interface

The wellbore state interface is a persistent interface which is guaranteed to be available by the system at all times regardless of the current system objective. The state of the interface will be published only when any of

the interface contents change. This interface is available for consumption by any actor of the system interested in observing the current state of the bit in relation to the bottom of the wellbore. The interface is guaranteed to publish only at a frequency equal to or less than that of the underlying control system.

Variable	Description	Type	SI Unit
holeDepth	Current measured hole depth	double	m
bitDepth	Current measured bit depth	double	m

### Wellbore State Topic

```
struct WellboreState
{
    unsigned long holeDepth;
    unsigned long bitDepth;
}
```

### Wellbore State QoS

The following QoS events are available for any observer of the Drilling State Topic.

Event	Description
OnDataAvailable	Event to notify subscriber new data is available for reading.
OnLivelinessLost	Event to notify subscriber that producer of data is no longer valid.
OnDeadlineMissed	Event to notify subscriber that expected deadline for new data sample has passed.
Reliability	Best Effort
Durability	Volatile Durability
History	Only preserve the last sample

### Circulation State Interface

The circulation state interface is a persistent interface which is guaranteed to be available by the system at all times regardless of the current system objective. The state of the interface will be published only when any of the interface contents change. This interface is available for consumption by any actor of the system interested in observing the current process state of the circulation system. The interface is guaranteed to publish only at a

frequency equal to or less than that of the underlying control system.

Variable	Description	Type	SI Unit
actualFlowRate	Actual measured volumetric flow in.	double	m <sup>3</sup> /sec
actualStandpipePressure	Actual measured standpipe pressure.	double	Pa
minFlowRate	Minimum volumetric flow in.	double	m <sup>3</sup> /sec
maxFlowRate	Maximum volumetric flow in.	double	m <sup>3</sup> /sec
minStandpipePressure	Minimum allowable standpipe pressure.	double	Pa
maxStandpipePressure	Maximum allowable standpipe pressure.	double	Pa
targetFlowRate	Target volumetric flow in.	double	m <sup>3</sup> /sec
circulateStatus	<ul style="list-style-type: none"><li>• Good</li><li>• Fault</li></ul>	enumeration	N/A

Circulation State Topic

```
struct CirculationState
{
    double    acutalFlowRate;
    double    actualStandpipePressure;
    double    minFlowRate;
    double    maxFlowRate;
    double    minStandpipePressure;
    double    maxStandpipePressure;
    double    targetFlowRate;
    STATUS    circulateStatus;
}
```

Circulation State QoS

The following QoS events are available for any observer of the Drilling State Topic.

Event	Description
OnDataAvailable	Event to notify subscriber new data is available for reading.
OnLivelinessLost	Event to notify subscriber that producer of data is no longer valid.
OnDeadlineMissed	Event to notify subscriber that expected deadline for new data sample has passed.
Reliability	Best Effort
Durability	Volatile Durability
History	Only preserve the last sample

## Hoist State Interface

The hoist state interface is a persistent interface which is guaranteed to be available by the system at all times regardless of the current system objective. The state of the interface will be published only when any of the interface contents change. This interface is available for consumption by any actor of the system interested in observing the current process state of the hoisting system. The interface is guaranteed to publish only at a frequency equal to or less than that of the underlying control system.

*NOTE* : Please note the following semantics

**velocity** is always in reference to block speed

**position** is always in reference to elevator (*internal calculation of the control system may use both the elevator and the block to calculate position*).



Variable	Description	Type	SI Unit
actualVelocity	Actual block velocity(+ Hoisting, - Lowering)	double	m/sec
actualPosition	Actual measured quill tip position	double	m
maxHoistVelocity	Maximum hoist velocity of the block.	double	m/sec
maxLowerVelocity	Maximum lower velocity of the block.	double	m/sec
maxHoistPosition	Minimum distance the elevator will hoist.	double	m
maxLowerPosition	Maximum distance the elevator will lower.	double	m
targetVelocity	Target hoisting velocity, (+ Hoisting, - Lowering)	double	m/sec
targetPosition	Target measured quill position	double	m
hoistStatus	<ul style="list-style-type: none"><li>• Good</li><li>• Fault</li></ul>	enumeration	N/A

Hoist State Topic

```
struct HoistState
{
    double    actualVelocity;
    double    actualPosition;
    double    maxHoistVelocity;
    double    maxLowerVelocity;
    double    maxHoistVelocity;
    double    maxLowerVelocity;
    double    targetVelocity;
    double    targetPosition;
    STATUS    hoistStatus;
}
```

Hoist State QoS

Event	Description
OnDataAvailable	Event to notify subscriber new data is available for reading.
OnLivelinessLost	Event to notify subscriber that producer of data is no longer valid.
OnDeadlineMissed	Event to notify subscriber that expected deadline for new data sample has passed.
Reliability	Best Effort
Durability	Volatile Durability
History	Only preserve the last sample

## Rotate Request Interface

The rotate request interface is an ephemeral interface which issued by a requestor to modify the current state of the system. The state of the interface will be published only when the publishing algorithm requests.

Variable	Description	Type	SI Unit
targetRate	Requested target in rad/sec.	double	rad/sec

## Rotate Request Topic

```
struct RotateRequest
{
    double    targetRate;
};
```

## Rotate Request QoS

The following QoS events are available for any observer of the Drilling State Calibration Topic.

Qos Policy	Description
OnLivelinessLost	Event to notify subscriber that producer of data is no longer valid.
Reliability	Best Effort
Durability	Volatile Durability
History	Only preserve the last sample

# Rotate State Interface

The rotation state interface is a persistent interface which is guaranteed to be available by the system at all times regardless of the current system objective. The state of the interface will be published only when any of the interface contents change. This interface is available for consumption by any actor of the system interested in observing the current process state of the rotation system. The interface is guaranteed to publish only at a frequency equal to or less than that of the underlying control system.

Variable	Description	Type	SI Unit
actualRate	Current rate in rad/sec.	double	rad/sec
minRate	Minimum rate in rad/sec.	double	rad/sec
maxRate	Maximum rate in rad/sec.	double	rad/sec
targetRate	Current target in rad/sec.	double	rad/sec
rotateStatus	<ul style="list-style-type: none"><li>• Good</li><li>• Fault</li></ul>	enumeration	N/A

## Rotation State Topic

```
struct RotationState
{
    double    actualRate;
    double    minRate;
    double    maxRate;
    double    targetRate;
    STATUS    rotateStatus;
}
```

## Rotation State QoS

The following QoS events are available for any observer of the Drilling State Topic.

Event	Description
OnDataAvailable	Event to notify subscriber new data is available for reading.
OnLivelinessLost	Event to notify subscriber that producer of data is no longer valid.
OnDeadlineMissed	Event to notify subscriber that expected deadline for new data sample has passed.
Reliability	Best Effort
Durability	Volatile Durability
History	Only preserve the last sample

## Future Interfaces

The falling interfaces are to be provided in the future, but some aspect of the interface may be available in the current release.

### Formation Interface

The formation interface is a persistent interface which is guaranteed to be available by the system at all times regardless of the current system objective. The state of the interface will be published only when any of the interface contents change. This interface is available for consumption by any actor of the system interested in observing the historical state of the formations. The interface is guaranteed to publish only at a frequency equal to or less than that of the underlying control system.

Variable	Description	Type	SI Unit
name	Name of formation	string	N/A
topOfFormation	Top of formation expressed as measured depth	double	m
bottomOfFormation	Bottom of formation expressed as measured depth	double	m
porosity	Measure of its ability to hold a fluid	double	N/A
permeability	Measure of the resistance to the flow of a fluid through a rock	double	m <sup>2</sup>
strength			

## Formation Topic

```
struct Formation
{
    string name;
    double topOfFormation; // in measured depth
    double bottomOfFormation; // in measured depth
    double porosity;
    double permeability;
    double strength;
}
```

## Formation QoS

The following QoS events are available for any observer of the Drilling State Topic.

Event	Description
OnDataAvailable	Event to notify subscriber new data is available for reading.
OnLivelinessLost	Event to notify subscriber that producer of data is no longer valid.
Reliability	Reliable
Durability	Transient Local
History	Preserve all sample

## Pipe Tally Interface

The pipe tally interface is a persistent interface which is guaranteed to be available by the system at all times regardless of the current system objective. The state of the interface will be published only when any of the interface contents change. This interface is available for consumption by any actor of the system interested in observing the historical contents of the pipe tally. The interface is guaranteed to publish only at a frequency equal to or less than that of the underlying control system.

Variable	Description	Type	SI Unit
serialNumber	Serial number of tool joint.	string	N/A
jointNumber	Joint number in reference to string.	integer	N/A
standNumber	Stand number in reference to string.	integer	N/A
weight	Stand weight.	double	N
outerDiameter	Outer diameter of tool joint.	double	m
innerDiameter	Inner diameter of tool joint.	double	m
kellyDown	Depth at which Kelly Down will occur.	double	m
length	Strapped length of stand measured by the driller.	double	m
pipeLength	Sum of all active drill pipe excluding the BHA.	double	m
stringLength	Sum of all active elements in the drill string, including the BHA.	double	m
description	Driller notes.	string	N/A
pipeType	<ul style="list-style-type: none"> <li>• Heavy Weight drill pipe</li> <li>• Drill Collars</li> <li>• Standard Drill Pipe</li> </ul>	enumeration	N/A

## Pipe Tally Topic

```
struct PipeTally
{
    string          serialNumber;
    unsigned long   jointNumber;
    unsigned long   standNumber;
    double          weight;
    double          outerDiameter;
    double          innerDiameter;
    double          kellyDown;
    double          length;
    double          pipeLength;
    double          stringLength;
    string          description;
    PipeType        pipeType;
}
```

Pipe Tally QoS

Event	Description
OnDataAvailable	Event to notify subscriber new data is available for reading.
OnLivelinessLost	Event to notify subscriber that producer of data is no longer valid.
Reliability	Reliable
Durability	Transient Local
History	Preserve all sample

Sequence Diagrams

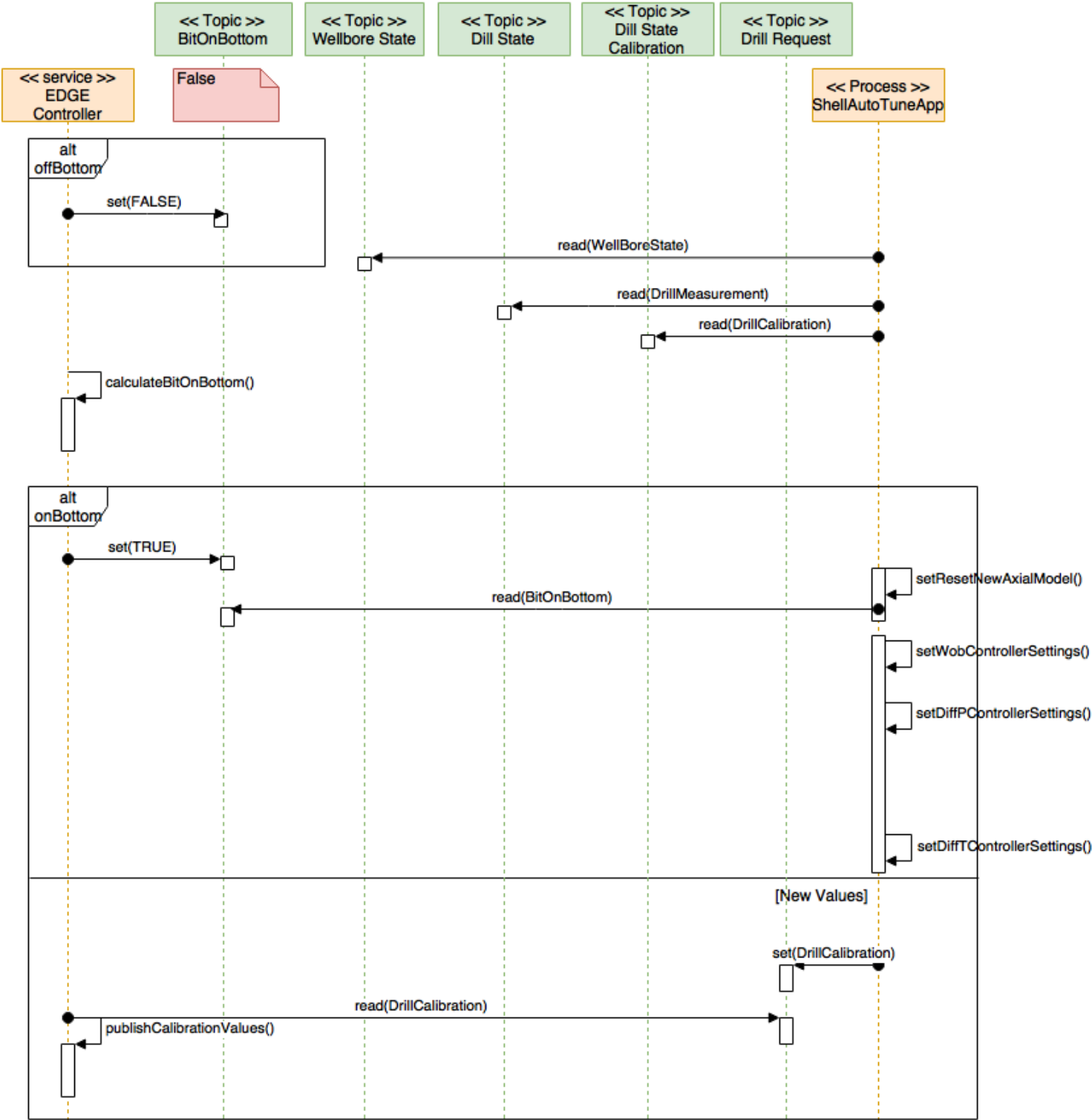
This section details the various interactions of the system.

Shell Auto Tuner - Cyclic Execution Cycle

This diagram depicts four state topics *BitOnBotton*, *WellboreState*, *DrillState*, *DrillCalibrationState* and one request topic *DrillRequest*. The four state topics are updated by the controller when new data samples arrive from the EDGE control system. The controller will monitor the state of the system and publish the *BitOnBottom* topic based on the state of the system. When the state of *BitOnBottom* changes the Shell Auto Tune app will be notified of the state change via an event. The application can then change its internal state from calculate to do not calculate. Event handlers on all topics will notify the application of all new data samples that are

published in the system.

Shell Auto Tuner Sequence Diagram - Cyclic Execution



Shell ROptimizer - Cyclic Execution Cycle



This diagram depicts four state topics *BitOnBottom*, *WellboreState*, *DrillState*, *DrillCalibrationState* and one request topic *DrillRequest*. The four state topics are updated by the controller when new data samples arrive from the EDGE control system. The controller will monitor the state of the system and publish the *BitOnBottom* topic based on the state of the system. When the state of *BitOnBottom* changes the Shell Auto Tune app will be notified of the state change via an event. The application can then change its internal state from calculate to do not calculate. Event handlers on all topics will notify the application of all new data samples that are published in the system.

# Shell ROptimzier Sequence Diagram - Cyclic Execution

